

Package ‘nemsqar’

May 12, 2026

Title National Emergency Medical Service Quality Alliance Measure Calculations

Version 1.2.0

Description Designed to automate the calculation of Emergency Medical Service (EMS) quality metrics, 'nemsqar' implements measures defined by the National EMS Quality Alliance (NEMSQA). By providing reliable, evidence-based quality assessments, the package supports EMS agencies, healthcare providers, and researchers in evaluating and improving patient outcomes. Users can find details on all approved NEMSQA measures at <<https://www.nemsqa.org/measures>>. Full technical specifications, including documentation and pseudocode used to develop 'nemsqar', are available on the NEMSQA website after creating a user profile at <<https://www.nemsqa.org>>.

License MIT + file LICENSE

URL <https://github.com/bemts-hhs/nemsqar>,
<https://bemts-hhs.github.io/nemsqar/>

BugReports <https://github.com/bemts-hhs/nemsqar/issues>

Depends R (>= 4.1)

Imports cli, dplyr (>= 1.2.1), glue, hms (>= 1.1.4), lifecycle,
lubridate (>= 1.9.5), rlang, stats, tibble (>= 3.3.1),
tidyselect (>= 1.2.1), utils

Suggests knitr, rmarkdown, testthat (>= 3.0.0), tidytr

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

LazyData true

LazyDataCompression bzip2

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Nicolas Foss [aut, cre],
 Samuel Kordik [aut] (ORCID: <<https://orcid.org/0000-0003-4230-1154>>),
 Alyssa Green [ctb],
 Iowa Department of Health and Human Services [cph]

Maintainer Nicolas Foss <nicolas.foss@hhs.iowa.gov>

Repository CRAN

Date/Publication 2026-05-12 16:20:02 UTC

Contents

airway_01	3
airway_01_population	7
airway_05	11
airway_05_population	15
airway_18	18
airway_18_population	23
asthma_01	26
asthma_01_population	29
hypoglycemia_01	32
hypoglycemia_01_population	35
nemsqar_airway_table	39
nemsqar_arrest_table	40
nemsqar_disposition_table	42
nemsqar_exam_table	43
nemsqar_injury_table	44
nemsqar_medications_table	45
nemsqar_patient_scene_table	46
nemsqar_procedures_table	47
nemsqar_response_table	48
nemsqar_situation_table	49
nemsqar_vitals_table	51
nemsqa_binomial_confint	52
pediatrics_03b	53
pediatrics_03b_population	56
pretty_percent	59
respiratory_01	60
respiratory_01_population	63
respiratory_02	66
respiratory_02_population	69
safety_01	72
safety_01_population	74
safety_02	77
safety_02_population	79
safety_04	82
safety_04_population	85
seizure_02	89
seizure_02_population	92

stroke_01	95
stroke_01_population	97
syncope_01	100
syncope_01_population	103
tbi_01	106
tbi_01_population	110
trauma_01	113
trauma_01_population	116
trauma_03	120
trauma_03_population	123
trauma_04	127
trauma_04_population	132
trauma_08	138
trauma_08_population	141
trauma_14	144
trauma_14_population	149
ttr_01	154
ttr_01_population	157

Index**162**

airway_01

*Airway-01 Calculation***Description**

Calculates the NEMSQA Airway-01 measure.

Calculates the proportion of times when the first endotracheal intubation attempt is successful with no peri-intubation hypoxia or hypotension.

Usage

```
airway_01(
  df = NULL,
  patient_scene_table = NULL,
  response_table = NULL,
  arrest_table = NULL,
  procedures_table = NULL,
  vitals_table = NULL,
  erecord_01_col,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col,
  epatient_16_col,
  earrest_01_col,
  eresponse_05_col,
  evitals_01_col,
```

```

    evitals_06_col,
    evitals_12_col,
    eprocedures_01_col,
    eprocedures_02_col,
    eprocedures_03_col,
    eprocedures_05_col,
    eprocedures_06_col,
    confidence_interval = FALSE,
    method = c("wilson", "clopper-pearson"),
    conf.level = 0.95,
    correct = TRUE,
    ...
)

```

Arguments

<code>df</code>	A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.
<code>patient_scene_table</code>	A data.frame or tibble containing at least ePatient, and eScene as a fact table.
<code>response_table</code>	A data.frame or tibble containing at least the eResponse fields needed for this measure's calculations.
<code>arrest_table</code>	A data.frame or tibble containing at least the eArrest fields needed for this measure's calculations.
<code>procedures_table</code>	A dataframe or tibble containing at least the eProcedures fields needed.
<code>vitals_table</code>	A dataframe or tibble containing at least the eVitals fields needed.
<code>erecord_01_col</code>	The column representing the EMS record unique identifier.
<code>incident_date_col</code>	Column that contains the incident date. This defaults to NULL as it is optional in case not available due to PII restrictions.
<code>patient_DOB_col</code>	Column that contains the patient's date of birth. This defaults to NULL as it is optional in case not available due to PII restrictions.
<code>epatient_15_col</code>	Column representing the patient's numeric age agnostic of unit.
<code>epatient_16_col</code>	Column representing the patient's age unit ("Years", "Months", "Days", "Hours", or "Minutes").
<code>earrest_01_col</code>	Column representing whether or not the patient is in arrest.
<code>eresponse_05_col</code>	Column that contains eResponse.05 or the response type.
<code>evitals_01_col</code>	Date-time or POSIXct column containing vital signs date/time
<code>evitals_06_col</code>	Numeric column containing systolic blood pressure values.
<code>evitals_12_col</code>	Numeric column containing pulse oximetry values.

<code>eprocedures_01_col</code>	Date-time or POSIXct column for procedures
<code>eprocedures_02_col</code>	Column name for whether or not the procedure was performed prior to EMS care being provided.
<code>eprocedures_03_col</code>	Column containing procedure codes with or without procedure names.
<code>eprocedures_05_col</code>	Column containing a count for how many times procedure was attempted.
<code>eprocedures_06_col</code>	Column indicating whether or not procedure was successful.
<code>confidence_interval</code>	Logical. If TRUE, the function calculates a confidence interval for the proportion estimate.
<code>method</code>	Character. Specifies the method used to calculate confidence intervals. Options are "wilson" (Wilson score interval) and "clopper-pearson" (exact binomial interval). Partial matching is supported, so "w" and "c" can be used as shorthand.
<code>conf.level</code>	Numeric. The confidence level for the interval, expressed as a proportion (e.g., 0.95 for a 95% confidence interval). Defaults to 0.95.
<code>correct</code>	Logical. If TRUE, applies a continuity correction to the Wilson score interval when <code>method = "wilson"</code> . Defaults to TRUE.
<code>...</code>	optional additional arguments to pass onto <code>dplyr::summarize</code> .

Value

A data.frame summarizing results for two population groups (Adults and Peds) with the following columns:

- `pop`: Population type (Adults and Peds).
- `numerator`: Count of incidents meeting the measure.
- `denominator`: Total count of included incidents.
- `prop`: Proportion of incidents meeting the measure.
- `prop_label`: Proportion formatted as a percentage with a specified number of decimal places.
- `lower_ci`: Lower bound of the confidence interval for `prop` (if `confidence_interval = TRUE`).
- `upper_ci`: Upper bound of the confidence interval for `prop` (if `confidence_interval = TRUE`).

Author(s)

Samuel Kordik, BBA, BS, Nicolas Foss Ed.D., MS

Examples

```

# If you are sourcing your data from a SQL database connection
# or if you have your data in several different tables,
# you can pass table inputs versus a single data.frame or tibble

# create tables to test correct functioning

# patient table
patient_table <- tibble::tibble(

  erecord_01 = rep(c("R1", "R2", "R3", "R4", "R5"), 2),
  incident_date = rep(as.Date(c("2025-01-01", "2025-01-05", "2025-02-01",
    "2025-01-01", "2025-06-01")), 2),
  patient_dob = rep(as.Date(c("2000-01-01", "2020-01-01", "2023-02-01",
    "2023-01-01", "1970-06-01")), 2),
  epatient_15 = rep(c(25, 5, 2, 2, 55), 2), # Ages
  epatient_16 = rep(c("Years", "Years", "Years", "Years", "Years"), 2)

)

# response table
response_table <- tibble::tibble(

  erecord_01 = rep(c("R1", "R2", "R3", "R4", "R5"), 2),
  eresponse_05 = rep(2205001, 10)

)

# vitals table
vitals_table <- tibble::tibble(

  erecord_01 = rep(c("R1", "R2", "R3", "R4", "R5"), 2),
  evitals_01 = lubridate::as_datetime(c("2025-01-01 22:59:00",
    "2025-01-05 11:58:00", "2025-02-01 18:57:00", "2025-01-01 04:58:00",
    "2025-06-01 12:57:00", "2025-01-01 23:05:00", "2025-01-05 12:04:00",
    "2025-02-01 19:03:00", "2025-01-01 05:02:00", "2025-06-01 13:01:00")),
  evitals_06 = rep(c(90, 100, 102, 103, 104), 2),
  evitals_12 = rep(c(90, 91, 92, 93, 94), 2)

)

# arrest table
arrest_table <- tibble::tibble(

  erecord_01 = rep(c("R1", "R2", "R3", "R4", "R5"), 2),
  earrest_01 = rep("No", 10)

)

# procedures table
procedures_table <- tibble::tibble(

  erecord_01 = rep(c("R1", "R2", "R3", "R4", "R5"), 2),

```

```

    eprocedures_01 = rep(lubridate::as_datetime(c("2025-01-01 23:00:00",
"2025-01-05 12:00:00", "2025-02-01 19:00:00", "2025-01-01 05:00:00",
"2025-06-01 13:00:00")), 2),
    eprocedures_02 = rep("No", 10),
    eprocedures_03 = rep(c(16883004, 112798008, 78121007, 49077009,
        673005), 2),
    eprocedures_05 = rep(1, 10),
    eprocedures_06 = rep(9923003, 10)

)

# Run the function
# Return 95% confidence intervals using the Wilson method
airway_01(df = NULL,
    patient_scene_table = patient_table,
    procedures_table = procedures_table,
    vitals_table = vitals_table,
    arrest_table = arrest_table,
    response_table = response_table,
    erecord_01_col = erecord_01,
    incident_date_col = incident_date,
    patient_DOB_col = patient_dob,
    epatient_15_col = epatient_15,
    epatient_16_col = epatient_16,
    eresponse_05_col = eresponse_05,
    eprocedures_01_col = eprocedures_01,
    eprocedures_02_col = eprocedures_02,
    eprocedures_03_col = eprocedures_03,
    eprocedures_05_col = eprocedures_05,
    eprocedures_06_col = eprocedures_06,
    earrest_01_col = earrest_01,
    evitals_01_col = evitals_01,
    evitals_06_col = evitals_06,
    evitals_12_col = evitals_12,
    confidence_interval = TRUE
)

```

airway_01_population *Airway-01 Population*

Description

This function processes and analyzes the dataset to generate the populations of interest needed to perform calculations to obtain performance data.

Usage

```

airway_01_population(
  df = NULL,

```

```

patient_scene_table = NULL,
response_table = NULL,
arrest_table = NULL,
procedures_table = NULL,
vitals_table = NULL,
erecord_01_col,
incident_date_col = NULL,
patient_DOB_col = NULL,
epatient_15_col,
epatient_16_col,
earrest_01_col,
eresponse_05_col,
evitals_01_col,
evitals_06_col,
evitals_12_col,
eprocedures_01_col,
eprocedures_02_col,
eprocedures_03_col,
eprocedures_05_col,
eprocedures_06_col
)

```

Arguments

df A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.

patient_scene_table A data.frame or tibble containing at least ePatient, and eScene as a fact table.

response_table A data.frame or tibble containing at least the eResponse fields needed for this measure's calculations.

arrest_table A data.frame or tibble containing at least the eArrest fields needed for this measure's calculations.

procedures_table A dataframe or tibble containing at least the eProcedures fields needed.

vitals_table A dataframe or tibble containing at least the eVitals fields needed.

erecord_01_col The column representing the EMS record unique identifier.

incident_date_col Column that contains the incident date. This defaults to NULL as it is optional in case not available due to PII restrictions.

patient_DOB_col Column that contains the patient's date of birth. This defaults to NULL as it is optional in case not available due to PII restrictions.

epatient_15_col Column representing the patient's numeric age agnostic of unit.

epatient_16_col Column representing the patient's age unit ("Years", "Months", "Days", "Hours", or "Minutes").

earrest_01_col	Column representing whether or not the patient is in arrest.
eresponse_05_col	Column that contains eResponse.05 or the response type.
evitals_01_col	Date-time or POSIXct column containing vital signs date/time
evitals_06_col	Numeric column containing systolic blood pressure values.
evitals_12_col	Numeric column containing pulse oximetry values.
eprocedures_01_col	Date-time or POSIXct column for procedures
eprocedures_02_col	Column name for whether or not the procedure was performed prior to EMS care being provided.
eprocedures_03_col	Column containing procedure codes with or without procedure names.
eprocedures_05_col	Column containing a count for how many times procedure was attempted.
eprocedures_06_col	Column indicating whether or not procedure was successful.

Value

A list that contains the following:

- a tibble with counts for each filtering step,
- a tibble for each population of interest
- a tibble for the initial population
- a tibble for the total dataset with computations
- a tibble with a summary of missingness for each column in each table

Author(s)

Samuel Kordik, BBA, BS, Nicolas Foss Ed.D., MS

Examples

```
# If you are sourcing your data from a SQL database connection
# or if you have your data in several different tables,
# you can pass table inputs versus a single data.frame or tibble

# create tables to test correct functioning

# patient table
patient_table <- tibble::tibble(

  erecord_01 = rep(c("R1", "R2", "R3", "R4", "R5"), 2),
  incident_date = rep(as.Date(c("2025-01-01", "2025-01-05", "2025-02-01",
    "2025-01-01", "2025-06-01")), 2),
  patient_dob = rep(as.Date(c("2000-01-01", "2020-01-01", "2023-02-01",
```

```

                "2023-01-01", "1970-06-01")), 2),
epatient_15 = rep(c(25, 5, 2, 2, 55), 2), # Ages
epatient_16 = rep(c("Years", "Years", "Years", "Years", "Years"), 2)

)

# response table
response_table <- tibble::tibble(

  erecord_01 = rep(c("R1", "R2", "R3", "R4", "R5"), 2),
  eresponse_05 = rep(2205001, 10)

)

# vitals table
vitals_table <- tibble::tibble(

  erecord_01 = rep(c("R1", "R2", "R3", "R4", "R5"), 2),
  evitals_01 = lubridate::as_datetime(c("2025-01-01 22:59:00",
    "2025-01-05 11:58:00", "2025-02-01 18:57:00", "2025-01-01 04:58:00",
    "2025-06-01 12:57:00", "2025-01-01 23:05:00", "2025-01-05 12:04:00",
    "2025-02-01 19:03:00", "2025-01-01 05:02:00", "2025-06-01 13:01:00")),
  evitals_06 = rep(c(90, 100, 102, 103, 104), 2),
  evitals_12 = rep(c(90, 91, 92, 93, 94), 2)

)

# arrest table
arrest_table <- tibble::tibble(

  erecord_01 = rep(c("R1", "R2", "R3", "R4", "R5"), 2),
  earrest_01 = rep("No", 10)

)

# procedures table
procedures_table <- tibble::tibble(

  erecord_01 = rep(c("R1", "R2", "R3", "R4", "R5"), 2),
  eprocedures_01 = rep(lubridate::as_datetime(c("2025-01-01 23:00:00",
    "2025-01-05 12:00:00", "2025-02-01 19:00:00", "2025-01-01 05:00:00",
    "2025-06-01 13:00:00")), 2),
  eprocedures_02 = rep("No", 10),
  eprocedures_03 = rep(c(16883004, 112798008, 78121007, 49077009,
    673005), 2),
  eprocedures_05 = rep(1, 10),
  eprocedures_06 = rep(9923003, 10)

)

# Run the function
result <- airway_01_population(df = NULL,
  patient_scene_table = patient_table,
  procedures_table = procedures_table,

```

```

vitals_table = vitals_table,
arrest_table = arrest_table,
response_table = response_table,
erecord_01_col = erecord_01,
incident_date_col = incident_date,
patient_DOB_col = patient_dob,
epatient_15_col = epatient_15,
epatient_16_col = epatient_16,
eresponse_05_col = eresponse_05,
eprocedures_01_col = eprocedures_01,
eprocedures_02_col = eprocedures_02,
eprocedures_03_col = eprocedures_03,
eprocedures_05_col = eprocedures_05,
eprocedures_06_col = eprocedures_06,
earrest_01_col = earrest_01,
evitals_01_col = evitals_01,
evitals_06_col = evitals_06,
evitals_12_col = evitals_12
)

# show the results of filtering at each step
result$filter_process

```

airway_05

Airway-05 Calculation

Description

Calculates the NEMSQA Airway-05 measure.

Calculates the proportion of endotracheal intubation attempts with adequate oxygenation.

Usage

```

airway_05(
  df = NULL,
  patient_scene_table = NULL,
  response_table = NULL,
  arrest_table = NULL,
  procedures_table = NULL,
  vitals_table = NULL,
  erecord_01_col,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col,
  epatient_16_col,
  earrest_01_col,
  eresponse_05_col,

```

```

    evitals_01_col,
    evitals_12_col,
    eprocedures_01_col,
    eprocedures_02_col,
    eprocedures_03_col,
    confidence_interval = FALSE,
    method = c("wilson", "clopper-pearson"),
    conf.level = 0.95,
    correct = TRUE,
    ...
)

```

Arguments

df A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.

patient_scene_table A data.frame or tibble containing at least ePatient, and eScene as a fact table.

response_table A data.frame or tibble containing at least the eResponse fields needed for this measure's calculations.

arrest_table A data.frame or tibble containing at least the eArrest fields needed for this measure's calculations.

procedures_table A dataframe or tibble containing at least the eProcedures fields needed.

vitals_table A dataframe or tibble containing at least the eVitals fields needed.

erecord_01_col The column representing the EMS record unique identifier.

incident_date_col Column that contains the incident date. This defaults to NULL as it is optional in case not available due to PII restrictions.

patient_DOB_col Column that contains the patient's date of birth. This defaults to NULL as it is optional in case not available due to PII restrictions.

epatient_15_col Column representing the patient's numeric age agnostic of unit.

epatient_16_col Column representing the patient's age unit ("Years", "Months", "Days", "Hours", or "Minutes").

earrest_01_col Column representing whether or not the patient is in arrest.

eresponse_05_col Column that contains eResponse.05 or the response type.

evitals_01_col Date-time or POSIXct column containing vital signs date/time

evitals_12_col Numeric column containing pulse oximetry values.

eprocedures_01_col Date-time or POSIXct column for procedures

<code>eprocedures_02_col</code>	Column name for whether or not the procedure was performed prior to EMS care being provided.
<code>eprocedures_03_col</code>	Column containing procedure codes with or without procedure names.
<code>confidence_interval</code>	Logical. If TRUE, the function calculates a confidence interval for the proportion estimate.
<code>method</code>	Character. Specifies the method used to calculate confidence intervals. Options are "wilson" (Wilson score interval) and "clopper-pearson" (exact binomial interval). Partial matching is supported, so "w" and "c" can be used as shorthand.
<code>conf.level</code>	Numeric. The confidence level for the interval, expressed as a proportion (e.g., 0.95 for a 95% confidence interval). Defaults to 0.95.
<code>correct</code>	Logical. If TRUE, applies a continuity correction to the Wilson score interval when method = "wilson". Defaults to TRUE.
<code>...</code>	optional additional arguments to pass onto <code>dplyr::summarize</code> .

Value

A data.frame summarizing results for two population groups (Adults and Peds) with the following columns:

- `pop`: Population type (Adults and Peds).
- `numerator`: Count of incidents meeting the measure.
- `denominator`: Total count of included incidents.
- `prop`: Proportion of incidents meeting the measure.
- `prop_label`: Proportion formatted as a percentage with a specified number of decimal places.
- `lower_ci`: Lower bound of the confidence interval for prop (if `confidence_interval = TRUE`).
- `upper_ci`: Upper bound of the confidence interval for prop (if `confidence_interval = TRUE`).

Author(s)

Samuel Kordik, BBA, BS, Nicolas Foss Ed.D., MS

Examples

```
# If you are sourcing your data from a SQL database connection
# or if you have your data in several different tables,
# you can pass table inputs versus a single data.frame or tibble

# create tables to test correct functioning

# patient table
patient_table <- tibble::tibble(
```

```

erecord_01 = rep(c("R1", "R2", "R3", "R4", "R5"), 2),
incident_date = rep(as.Date(c("2025-01-01", "2025-01-05", "2025-02-01",
"2025-01-01", "2025-06-01")), 2),
patient_dob = rep(as.Date(c("2000-01-01", "2020-01-01", "2023-02-01",
"2023-01-01", "1970-06-01")), 2),
epatient_15 = rep(c(25, 5, 2, 2, 55), 2), # Ages
epatient_16 = rep(c("Years", "Years", "Years", "Years", "Years"), 2)

)

# response table
response_table <- tibble::tibble(

  erecord_01 = rep(c("R1", "R2", "R3", "R4", "R5"), 2),
  eresponse_05 = rep(2205001, 10)

)

# vitals table
vitals_table <- tibble::tibble(

  erecord_01 = rep(c("R1", "R2", "R3", "R4", "R5"), 2),
  evitals_01 = lubridate::as_datetime(c("2025-01-01 22:59:00",
"2025-01-05 11:58:00", "2025-02-01 18:57:00", "2025-01-01 04:58:00",
"2025-06-01 12:57:00", "2025-01-01 23:05:00", "2025-01-05 12:04:00",
"2025-02-01 19:03:00", "2025-01-01 05:02:00", "2025-06-01 13:01:00")),
  evitals_12 = rep(c(94, 95, 96, 97, 98), 2)

)

# arrest table
arrest_table <- tibble::tibble(

  erecord_01 = rep(c("R1", "R2", "R3", "R4", "R5"), 2),
  earrest_01 = rep("No", 10)

)

# procedures table
procedures_table <- tibble::tibble(

  erecord_01 = rep(c("R1", "R2", "R3", "R4", "R5"), 2),
  eprocedures_01 = rep(lubridate::as_datetime(c("2025-01-01 23:00:00",
"2025-01-05 12:00:00", "2025-02-01 19:00:00", "2025-01-01 05:00:00",
"2025-06-01 13:00:00")), 2),
  eprocedures_02 = rep("No", 10),
  eprocedures_03 = rep(c(16883004, 112798008, 78121007, 49077009,
673005), 2)

)

# Run the function
# Return 95% confidence intervals using the Wilson method

```

```
airway_05(df = NULL,
  patient_scene_table = patient_table,
  procedures_table = procedures_table,
  vitals_table = vitals_table,
  arrest_table = arrest_table,
  response_table = response_table,
  erecord_01_col = erecord_01,
  incident_date_col = incident_date,
  patient_DOB_col = patient_dob,
  epatient_15_col = epatient_15,
  epatient_16_col = epatient_16,
  eresponse_05_col = eresponse_05,
  eprocedures_01_col = eprocedures_01,
  eprocedures_02_col = eprocedures_02,
  eprocedures_03_col = eprocedures_03,
  earrest_01_col = earrest_01,
  evitals_01_col = evitals_01,
  evitals_12_col = evitals_12,
  confidence_interval = TRUE
)
```

airway_05_population *Airway-05 Population*

Description

This function processes and analyzes the dataset to generate the populations of interest needed to perform calculations to obtain performance data.

Usage

```
airway_05_population(
  df = NULL,
  patient_scene_table = NULL,
  response_table = NULL,
  arrest_table = NULL,
  procedures_table = NULL,
  vitals_table = NULL,
  erecord_01_col,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col,
  epatient_16_col,
  earrest_01_col,
  eresponse_05_col,
  evitals_01_col,
  evitals_12_col,
  eprocedures_01_col,
```

```

    eprocedures_02_col,
    eprocedures_03_col
  )

```

Arguments

df A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.

patient_scene_table A data.frame or tibble containing at least ePatient, and eScene as a fact table.

response_table A data.frame or tibble containing at least the eResponse fields needed for this measure's calculations.

arrest_table A data.frame or tibble containing at least the eArrest fields needed for this measure's calculations.

procedures_table A dataframe or tibble containing at least the eProcedures fields needed.

vitals_table A dataframe or tibble containing at least the eVitals fields needed.

erecord_01_col The column representing the EMS record unique identifier.

incident_date_col Column that contains the incident date. This defaults to NULL as it is optional in case not available due to PII restrictions.

patient_DOB_col Column that contains the patient's date of birth. This defaults to NULL as it is optional in case not available due to PII restrictions.

epatient_15_col Column representing the patient's numeric age agnostic of unit.

epatient_16_col Column representing the patient's age unit ("Years", "Months", "Days", "Hours", or "Minutes").

earrest_01_col Column representing whether or not the patient is in arrest.

eresponse_05_col Column that contains eResponse.05 or the response type.

evitals_01_col Date-time or POSIXct column containing vital signs date/time

evitals_12_col Numeric column containing pulse oximetry values.

eprocedures_01_col Date-time or POSIXct column for procedures

eprocedures_02_col Column name for whether or not the procedure was performed prior to EMS care being provided.

eprocedures_03_col Column containing procedure codes with or without procedure names.

Value

A list that contains the following:

- a tibble with counts for each filtering step,
- a tibble for each population of interest
- a tibble for the initial population
- a tibble for the total dataset with computations
- a tibble with a summary of missingness for each column in each table

Author(s)

Samuel Kordik, BBA, BS, Nicolas Foss Ed.D., MS

Examples

```
# If you are sourcing your data from a SQL database connection
# or if you have your data in several different tables,
# you can pass table inputs versus a single data.frame or tibble

# create tables to test correct functioning

# patient table
patient_table <- tibble::tibble(

  erecord_01 = rep(c("R1", "R2", "R3", "R4", "R5"), 2),
  incident_date = rep(as.Date(c("2025-01-01", "2025-01-05", "2025-02-01",
    "2025-01-01", "2025-06-01")), 2),
  patient_dob = rep(as.Date(c("2000-01-01", "2020-01-01", "2023-02-01",
    "2023-01-01", "1970-06-01")), 2),
  epatient_15 = rep(c(25, 5, 2, 2, 55), 2), # Ages
  epatient_16 = rep(c("Years", "Years", "Years", "Years", "Years"), 2)

)

# response table
response_table <- tibble::tibble(

  erecord_01 = rep(c("R1", "R2", "R3", "R4", "R5"), 2),
  eresponse_05 = rep(2205001, 10)

)

# vitals table
vitals_table <- tibble::tibble(

  erecord_01 = rep(c("R1", "R2", "R3", "R4", "R5"), 2),
  evitals_01 = lubridate::as_datetime(c("2025-01-01 22:59:00",
    "2025-01-05 11:58:00", "2025-02-01 18:57:00", "2025-01-01 04:58:00",
    "2025-06-01 12:57:00", "2025-01-01 23:05:00", "2025-01-05 12:04:00",
    "2025-02-01 19:03:00", "2025-01-01 05:02:00", "2025-06-01 13:01:00")),
  evitals_12 = rep(c(90, 91, 92, 93, 94), 2)
```

```

)

# arrest table
arrest_table <- tibble::tibble(

  erecord_01 = rep(c("R1", "R2", "R3", "R4", "R5"), 2),
  earrest_01 = rep("No", 10)
)

# procedures table
procedures_table <- tibble::tibble(

  erecord_01 = rep(c("R1", "R2", "R3", "R4", "R5"), 2),
  eprocedures_01 = rep(lubridate::as_datetime(c("2025-01-01 23:00:00",
"2025-01-05 12:00:00", "2025-02-01 19:00:00", "2025-01-01 05:00:00",
"2025-06-01 13:00:00")), 2),
  eprocedures_02 = rep("No", 10),
  eprocedures_03 = rep(c(16883004, 112798008, 78121007, 49077009,
673005), 2)

)

# Run the function
result <- airway_05_population(df = NULL,
  patient_scene_table = patient_table,
  procedures_table = procedures_table,
  vitals_table = vitals_table,
  arrest_table = arrest_table,
  response_table = response_table,
  erecord_01_col = erecord_01,
  incident_date_col = incident_date,
  patient_DOB_col = patient_dob,
  epatient_15_col = epatient_15,
  epatient_16_col = epatient_16,
  eresponse_05_col = eresponse_05,
  eprocedures_01_col = eprocedures_01,
  eprocedures_02_col = eprocedures_02,
  eprocedures_03_col = eprocedures_03,
  earrest_01_col = earrest_01,
  evitals_01_col = evitals_01,
  evitals_12_col = evitals_12
)

# show the results of filtering at each step
result$filter_process

```

Description

This function processes and analyzes the dataset to calculate the "Airway-18" NEMSQA metric. It includes cleaning and transforming several columns related to patient data, airway procedures, and vital signs, and it returns a cleaned dataset with the relevant calculations. The final calculation is an assessment of the successful last invasive airway procedures performed during an EMS response originating from a 911 request in which waveform capnography is used for tube placement confirmation.

Usage

```
airway_18(
  df = NULL,
  patient_scene_table = NULL,
  procedures_table = NULL,
  vitals_table = NULL,
  airway_table = NULL,
  response_table = NULL,
  erecord_01_col,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col,
  epatient_16_col,
  eresponse_05_col,
  eprocedures_01_col,
  eprocedures_02_col,
  eprocedures_03_col,
  eprocedures_06_col,
  eairway_02_col = NULL,
  eairway_04_col = NULL,
  evitals_01_col,
  evitals_16_col,
  confidence_interval = FALSE,
  method = c("wilson", "clopper-pearson"),
  conf.level = 0.95,
  correct = TRUE,
  ...
)
```

Arguments

<code>df</code>	A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.
<code>patient_scene_table</code>	A dataframe or tibble containing at least ePatient, and eScene as a fact table.
<code>procedures_table</code>	A dataframe or tibble containing at least the eProcedures fields needed.
<code>vitals_table</code>	A dataframe or tibble containing at least the eVitals fields needed.

<code>airway_table</code>	A data frame or tibble containing only the eAirway fields needed for this measure's calculations. Default is NULL.
<code>response_table</code>	A data.frame or tibble containing at least the eResponse fields needed for this measure's calculations.
<code>erecord_01_col</code>	The column representing the EMS record unique identifier.
<code>incident_date_col</code>	Column that contains the incident date. This defaults to NULL as it is optional in case not available due to PII restrictions.
<code>patient_DOB_col</code>	Column that contains the patient's date of birth. This defaults to NULL as it is optional in case not available due to PII restrictions.
<code>epatient_15_col</code>	Column representing the patient's numeric age agnostic of unit.
<code>epatient_16_col</code>	Column representing the patient's age unit ("Years", "Months", "Days", "Hours", or "Minutes").
<code>eresponse_05_col</code>	Column that contains eResponse.05 or the response type.
<code>eprocedures_01_col</code>	Date-time or POSIXct column for procedures
<code>eprocedures_02_col</code>	Column name for whether or not the procedure was performed prior to EMS care being provided.
<code>eprocedures_03_col</code>	Column containing procedure codes with or without procedure names.
<code>eprocedures_06_col</code>	Column indicating whether or not procedure was successful.
<code>eairway_02_col</code>	Column name for date/time airway device placement confirmation. Default is NULL.
<code>eairway_04_col</code>	Column name for confirmation of airway device placement. Default is NULL.
<code>evitals_01_col</code>	Date-time or POSIXct column containing vital signs date/time
<code>evitals_16_col</code>	Column with numeric value of the patient's exhaled end tidal carbon dioxide (ETCO2) level measured as a unit of pressure in millimeters of mercury (mmHg), percentage or, kilopascal (kPa).
<code>confidence_interval</code>	Logical. If TRUE, the function calculates a confidence interval for the proportion estimate.
<code>method</code>	Character. Specifies the method used to calculate confidence intervals. Options are "wilson" (Wilson score interval) and "clopper-pearson" (exact binomial interval). Partial matching is supported, so "w" and "c" can be used as shorthand.
<code>conf.level</code>	Numeric. The confidence level for the interval, expressed as a proportion (e.g., 0.95 for a 95% confidence interval). Defaults to 0.95.
<code>correct</code>	Logical. If TRUE, applies a continuity correction to the Wilson score interval when method = "wilson". Defaults to TRUE.
<code>...</code>	optional additional arguments to pass onto <code>dplyr::summarize</code> .

Value

A data.frame summarizing results for two population groups (Adults and Peds) with the following columns:

- pop: Population type (Adults and Peds).
- numerator: Count of incidents meeting the measure.
- denominator: Total count of included incidents.
- prop: Proportion of incidents meeting the measure.
- prop_label: Proportion formatted as a percentage with a specified number of decimal places.
- lower_ci: Lower bound of the confidence interval for prop (if confidence_interval = TRUE).
- upper_ci: Upper bound of the confidence interval for prop (if confidence_interval = TRUE).

Author(s)

Nicolas Foss, Ed.D., MS, Samuel Kordik, BBA, BS

Examples

```
# If you are sourcing your data from a SQL database connection
# or if you have your data in several different tables,
# you can pass table inputs versus a single data.frame or tibble

# create tables to test correct functioning

# patient table
patient_table <- tibble::tibble(

  erecord_01 = rep(c("R1", "R2", "R3", "R4", "R5"), 2),
  incident_date = rep(as.Date(c("2025-01-01", "2025-01-05", "2025-02-01",
    "2025-01-01", "2025-06-01")), 2),
  patient_dob = rep(as.Date(c("2000-01-01", "2020-01-01", "2023-02-01",
    "2023-01-01", "1970-06-01")), 2),
  epatient_15 = rep(c(25, 5, 2, 2, 55), 2), # Ages
  epatient_16 = rep(c("Years", "Years", "Years", "Years", "Years"), 2)

)

# response table
response_table <- tibble::tibble(

  erecord_01 = rep(c("R1", "R2", "R3", "R4", "R5"), 2),
  eresponse_05 = rep(2205001, 10)

)

# vitals table
vitals_table <- tibble::tibble(
```

```

erecord_01 = rep(c("R1", "R2", "R3", "R4", "R5"), 2),
evitals_01 = lubridate::as_datetime(c("2025-01-01 23:02:00",
"2025-01-05 12:03:00", "2025-02-01 19:04:00", "2025-01-01 05:05:00",
"2025-06-01 13:01:00", "2025-01-01 23:02:00",
"2025-01-05 12:03:00", "2025-02-01 19:04:00", "2025-01-01 05:05:00",
"2025-06-01 13:06:00")),
evitals_16 = rep(c(5, 6, 7, 8, 9), 2)

)

# airway table
airway_table <- tibble::tibble(
erecord_01 = rep(c("R1", "R2", "R3", "R4", "R5"), 2),
eairway_02 = rep(lubridate::as_datetime(c("2025-01-01 23:05:00",
"2025-01-05 12:02:00", "2025-02-01 19:03:00", "2025-01-01 05:04:00",
"2025-06-01 13:06:00")), 2),
eairway_04 = rep(4004019, 10)
)

# procedures table
procedures_table <- tibble::tibble(

erecord_01 = rep(c("R1", "R2", "R3", "R4", "R5"), 2),
eprocedures_01 = rep(lubridate::as_datetime(c("2025-01-01 23:00:00",
"2025-01-05 12:00:00", "2025-02-01 19:00:00", "2025-01-01 05:00:00",
"2025-06-01 13:00:00")), 2),
eprocedures_02 = rep("No", 10),
eprocedures_03 = rep(c(16883004, 112798008, 78121007, 49077009,
673005), 2),
eprocedures_06 = rep(9923003, 10)

)

# Run the function
# Return 95% confidence intervals using the Wilson method
airway_18(df = NULL,
patient_scene_table = patient_table,
procedures_table = procedures_table,
vitals_table = vitals_table,
response_table = response_table,
airway_table = airway_table,
erecord_01_col = erecord_01,
incident_date_col = incident_date,
patient_DOB_col = patient_dob,
epatient_15_col = epatient_15,
epatient_16_col = epatient_16,
eresponse_05_col = eresponse_05,
eprocedures_01_col = eprocedures_01,
eprocedures_02_col = eprocedures_02,
eprocedures_03_col = eprocedures_03,
eprocedures_06_col = eprocedures_06,
evitals_01_col = evitals_01,

```

```
    evitals_16_col = evitals_16,  
    eairway_02_col = eairway_02,  
    eairway_04_col = eairway_04,  
    confidence_interval = TRUE  
  )
```

airway_18_population *Airway-18 Populations*

Description

This function processes and analyzes the dataset to generate the populations of interest needed to perform calculations to obtain performance data.

Usage

```
airway_18_population(  
  df = NULL,  
  patient_scene_table = NULL,  
  procedures_table = NULL,  
  vitals_table = NULL,  
  airway_table = NULL,  
  response_table = NULL,  
  erecord_01_col,  
  incident_date_col = NULL,  
  patient_DOB_col = NULL,  
  epatient_15_col,  
  epatient_16_col,  
  eresponse_05_col,  
  eprocedures_01_col,  
  eprocedures_02_col,  
  eprocedures_03_col,  
  eprocedures_06_col,  
  eairway_02_col = NULL,  
  eairway_04_col = NULL,  
  evitals_01_col,  
  evitals_16_col  
)
```

Arguments

df A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.

patient_scene_table A data.frame or tibble containing at least ePatient, and eScene as a fact table.

procedures_table A dataframe or tibble containing at least the eProcedures fields needed.

vitals_table	A dataframe or tibble containing at least the eVitals fields needed.
airway_table	A data frame or tibble containing only the eAirway fields needed for this measure's calculations. Default is NULL.
response_table	A data.frame or tibble containing at least the eResponse fields needed for this measure's calculations.
erecord_01_col	The column representing the EMS record unique identifier.
incident_date_col	Column that contains the incident date. This defaults to NULL as it is optional in case not available due to PII restrictions.
patient_DOB_col	Column that contains the patient's date of birth. This defaults to NULL as it is optional in case not available due to PII restrictions.
epatient_15_col	Column representing the patient's numeric age agnostic of unit.
epatient_16_col	Column representing the patient's age unit ("Years", "Months", "Days", "Hours", or "Minutes").
eresponse_05_col	Column that contains eResponse.05 or the response type.
eprocedures_01_col	Date-time or POSIXct column for procedures
eprocedures_02_col	Column name for whether or not the procedure was performed prior to EMS care being provided.
eprocedures_03_col	Column containing procedure codes with or without procedure names.
eprocedures_06_col	Column indicating whether or not procedure was successful.
eairway_02_col	Column name for date/time airway device placement confirmation. Default is NULL.
eairway_04_col	Column name for confirmation of airway device placement. Default is NULL.
evitals_01_col	Date-time or POSIXct column containing vital signs date/time
evitals_16_col	Column with numeric value of the patient's exhaled end tidal carbon dioxide (ETCO2) level measured as a unit of pressure in millimeters of mercury (mmHg), percentage or, kilopascal (kPa).

Value

A list that contains the following:

- a tibble with counts for each filtering step,
- a tibble for each population of interest
- a tibble for the initial population
- a tibble for the total dataset with computations
- a tibble with a summary of missingness for each column in each table

Author(s)

Nicolas Foss, Ed.D., MS, Samuel Kordik, BBA, BS

Examples

```
# If you are sourcing your data from a SQL database connection
# or if you have your data in several different tables,
# you can pass table inputs versus a single data.frame or tibble

# create tables to test correct functioning

# patient table
patient_table <- tibble::tibble(

  erecord_01 = rep(c("R1", "R2", "R3", "R4", "R5"), 2),
  incident_date = rep(as.Date(c("2025-01-01", "2025-01-05", "2025-02-01",
    "2025-01-01", "2025-06-01")), 2),
  patient_dob = rep(as.Date(c("2000-01-01", "2020-01-01", "2023-02-01",
    "2023-01-01", "1970-06-01")), 2),
  epatient_15 = rep(c(25, 5, 2, 2, 55), 2), # Ages
  epatient_16 = rep(c("Years", "Years", "Years", "Years", "Years"), 2)

)

# response table
response_table <- tibble::tibble(

  erecord_01 = rep(c("R1", "R2", "R3", "R4", "R5"), 2),
  eresponse_05 = rep(2205001, 10)

)

# vitals table
vitals_table <- tibble::tibble(

  erecord_01 = rep(c("R1", "R2", "R3", "R4", "R5"), 2),
  evitals_01 = lubridate::as_datetime(c("2025-01-01 23:02:00",
    "2025-01-05 12:03:00", "2025-02-01 19:04:00", "2025-01-01 05:05:00",
    "2025-06-01 13:01:00", "2025-01-01 23:02:00",
    "2025-01-05 12:03:00", "2025-02-01 19:04:00", "2025-01-01 05:05:00",
    "2025-06-01 13:06:00")),
  evitals_16 = rep(c(5, 6, 7, 8, 9), 2)

)

# airway table
airway_table <- tibble::tibble(
  erecord_01 = rep(c("R1", "R2", "R3", "R4", "R5"), 2),
  eairway_02 = rep(lubridate::as_datetime(c("2025-01-01 23:05:00",
    "2025-01-05 12:02:00", "2025-02-01 19:03:00", "2025-01-01 05:04:00",
    "2025-06-01 13:06:00")), 2),
  eairway_04 = rep(4004019, 10)
```

```

)

# procedures table
procedures_table <- tibble::tibble(

  erecord_01 = rep(c("R1", "R2", "R3", "R4", "R5"), 2),
  eprocedures_01 = rep(lubridate::as_datetime(c("2025-01-01 23:00:00",
"2025-01-05 12:00:00", "2025-02-01 19:00:00", "2025-01-01 05:00:00",
"2025-06-01 13:00:00")), 2),
  eprocedures_02 = rep("No", 10),
  eprocedures_03 = rep(c(16883004, 112798008, 78121007, 49077009,
673005), 2),
  eprocedures_06 = rep(9923003, 10)

)

# Run the function
result <- airway_18_population(df = NULL,
  patient_scene_table = patient_table,
  procedures_table = procedures_table,
  vitals_table = vitals_table,
  response_table = response_table,
  airway_table = airway_table,
  erecord_01_col = erecord_01,
  incident_date_col = incident_date,
  patient_DOB_col = patient_dob,
  epatient_15_col = epatient_15,
  epatient_16_col = epatient_16,
  eresponse_05_col = eresponse_05,
  eprocedures_01_col = eprocedures_01,
  eprocedures_02_col = eprocedures_02,
  eprocedures_03_col = eprocedures_03,
  eprocedures_06_col = eprocedures_06,
  evitals_01_col = evitals_01,
  evitals_16_col = evitals_16,
  eairway_02_col = eairway_02,
  eairway_04_col = eairway_04
)

# show the results of filtering at each step
result$filter_process

```

asthma_01

Asthma-01 Calculation

Description

Calculates the NEMSQA Asthma-01 measure.

Calculates key statistics related to asthma-related incidents in an EMS dataset, specifically focusing on cases where 911 was called for respiratory distress, and certain medications were administered.

This function segments the data by age into adult and pediatric populations, computing the proportion of cases that received beta-agonist treatment.

Usage

```
asthma_01(
  df = NULL,
  patient_scene_table = NULL,
  response_table = NULL,
  situation_table = NULL,
  medications_table = NULL,
  erecord_01_col,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col,
  epatient_16_col,
  eresponse_05_col,
  esituation_11_col,
  esituation_12_col,
  emedications_03_col,
  confidence_interval = FALSE,
  method = c("wilson", "clopper-pearson"),
  conf.level = 0.95,
  correct = TRUE,
  ...
)
```

Arguments

<code>df</code>	A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.
<code>patient_scene_table</code>	A data.frame or tibble containing at least ePatient, and eScene as a fact table.
<code>response_table</code>	A data.frame or tibble containing at least the eResponse fields needed for this measure's calculations.
<code>situation_table</code>	A data.frame or tibble containing at least the eSituation fields needed for this measure's calculations. Default is NULL.
<code>medications_table</code>	A data.frame or tibble containing at least the eMedications fields needed for this measure's calculations. Default is NULL.
<code>erecord_01_col</code>	The column representing the EMS record unique identifier.
<code>incident_date_col</code>	Column that contains the incident date. This defaults to NULL as it is optional in case not available due to PII restrictions.
<code>patient_DOB_col</code>	Column that contains the patient's date of birth. This defaults to NULL as it is optional in case not available due to PII restrictions.

<code>epatient_15_col</code>	Column representing the patient's numeric age agnostic of unit.
<code>epatient_16_col</code>	Column representing the patient's age unit ("Years", "Months", "Days", "Hours", or "Minutes").
<code>eresponse_05_col</code>	Column that contains eResponse.05 or the response type.
<code>esituation_11_col</code>	Column that contains eSituation.11 provider primary impression data.
<code>esituation_12_col</code>	Column that contains all eSituation.12 values as (possible a single comma-separated list), provider secondary impression data.
<code>emedications_03_col</code>	Column that contains all medication administered to the patient (eMedications.03) values as a single comma-separated list per distinct eRecord.01 ID.
<code>confidence_interval</code>	Logical. If TRUE, the function calculates a confidence interval for the proportion estimate.
<code>method</code>	Character. Specifies the method used to calculate confidence intervals. Options are "wilson" (Wilson score interval) and "clopper-pearson" (exact binomial interval). Partial matching is supported, so "w" and "c" can be used as shorthand.
<code>conf.level</code>	Numeric. The confidence level for the interval, expressed as a proportion (e.g., 0.95 for a 95% confidence interval). Defaults to 0.95.
<code>correct</code>	Logical. If TRUE, applies a continuity correction to the Wilson score interval when method = "wilson". Defaults to TRUE.
<code>...</code>	optional additional arguments to pass onto <code>dplyr::summarize</code> .

Value

A data.frame summarizing results for two population groups (All, Adults and Peds) with the following columns:

- `pop`: Population type (All, Adults, and Peds).
- `numerator`: Count of incidents meeting the measure.
- `denominator`: Total count of included incidents.
- `prop`: Proportion of incidents meeting the measure.
- `prop_label`: Proportion formatted as a percentage with a specified number of decimal places.
- `lower_ci`: Lower bound of the confidence interval for prop (if `confidence_interval = TRUE`).
- `upper_ci`: Upper bound of the confidence interval for prop (if `confidence_interval = TRUE`).

Author(s)

Nicolas Foss, Ed.D., MS

Examples

```

# Synthetic test data
test_data <- tibble::tibble(
  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  epatient_15 = c(34, 5, 45, 2, 60), # Ages
  epatient_16 = c("Years", "Years", "Years", "Months", "Years"),
  eresponse_05 = rep(2205001, 5),
  esituation_11 = c("Respiratory Distress", "Respiratory Distress",
    "Chest Pain", "Respiratory Distress", "Respiratory Distress"),
  esituation_12 = c("Asthma", "Asthma", "Other condition", "Asthma", "Asthma"),
  emedications_03 = c("Albuterol", "Albuterol", "Epinephrine", "None",
    "Albuterol")
)

# Run the function
# Return 95% confidence intervals using the Wilson method
asthma_01(
  df = test_data,
  erecord_01_col = erecord_01,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col = epatient_15,
  epatient_16_col = epatient_16,
  eresponse_05_col = eresponse_05,
  esituation_11_col = esituation_11,
  esituation_12_col = esituation_12,
  emedications_03_col = emedications_03,
  confidence_interval = TRUE
)

```

asthma_01_population *Asthma-01 Populations*

Description

Filters data down to the target populations for Asthma-01, and categorizes records to identify needed information for the calculations.

Identifies key categories related to asthma-related incidents in an EMS dataset, specifically focusing on cases where 911 was called for respiratory distress, and certain medications were administered. This function segments the data by age into adult and pediatric populations, computing the proportion of cases that received beta-agonist treatment.

Usage

```

asthma_01_population(
  df = NULL,
  patient_scene_table = NULL,
  response_table = NULL,

```

```

situation_table = NULL,
medications_table = NULL,
erecord_01_col,
incident_date_col = NULL,
patient_DOB_col = NULL,
epatient_15_col,
epatient_16_col,
eresponse_05_col,
esituation_11_col,
esituation_12_col,
emedications_03_col
)

```

Arguments

df A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.

patient_scene_table A data.frame or tibble containing at least ePatient, and eScene as a fact table.

response_table A data.frame or tibble containing at least the eResponse fields needed for this measure's calculations.

situation_table A data.frame or tibble containing at least the eSituation fields needed for this measure's calculations. Default is NULL.

medications_table A data.frame or tibble containing at least the eMedications fields needed for this measure's calculations. Default is NULL.

erecord_01_col The column representing the EMS record unique identifier.

incident_date_col Column that contains the incident date. This defaults to NULL as it is optional in case not available due to PII restrictions.

patient_DOB_col Column that contains the patient's date of birth. This defaults to NULL as it is optional in case not available due to PII restrictions.

epatient_15_col Column representing the patient's numeric age agnostic of unit.

epatient_16_col Column representing the patient's age unit ("Years", "Months", "Days", "Hours", or "Minutes").

eresponse_05_col Column that contains eResponse.05 or the response type.

esituation_11_col Column that contains eSituation.11 provider primary impression data.

esituation_12_col Column that contains all eSituation.12 values as (possible a single comma-separated list), provider secondary impression data.

emedications_03_col

Column that contains all medication administered to the patient (eMedications.03) values as a single comma-separated list per distinct eRecord.01 ID.

Value

A list that contains the following:

- a tibble with counts for each filtering step,
- a tibble for each population of interest
- a tibble for the initial population
- a tibble for the total dataset with computations
- a tibble with a summary of missingness for each column in each table

Author(s)

Nicolas Foss, Ed.D., MS

Examples

```
# If you are sourcing your data from a SQL database connection
# or if you have your data in several different tables,
# you can pass table inputs versus a single data.frame or tibble

# create tables to test correct functioning

# patient table
patient_table <- tibble::tibble(

  erecord_01 = 1:3,
  incident_date = as.Date(c("2025-01-01", "2025-01-05", "2025-02-01")),
  patient_dob = as.Date(c("2000-01-01", "2020-01-01", "2023-01-01")),
  epatient_15 = c(25, 5, 2),
  epatient_16 = c("years", "years", "months")

)

# response table
response_table <- tibble::tibble(

  erecord_01 = 1:3,
  eresponse_05 = c("2205001", "2205009", "2205003")

)

# situation table
situation_table <- tibble::tibble(

  erecord_01 = 1:3,
  esituation_11 = c("weakness", "asthma", "bronchospasm"),
  esituation_12 = c("asthma", "weakness", "weakness")

)
```

```

)

# medications table
medications_table <- tibble::tibble(

  erecord_01 = 1:3,
  emedications_03 = c("albuterol", "levalbuterol", "metaproterenol")

)

# test the success of the function
result <- asthma_01_population(patient_scene_table = patient_table,
                              response_table = response_table,
                              situation_table = situation_table,
                              medications_table = medications_table,
                              erecord_01_col = erecord_01,
                              incident_date_col = incident_date,
                              patient_DOB_col = patient_dob,
                              epatient_15_col = epatient_15,
                              epatient_16_col = epatient_16,
                              eresponse_05_col = eresponse_05,
                              esituation_11_col = esituation_11,
                              esituation_12_col = esituation_12,
                              emedications_03_col = emedications_03
                              )

# show the results of filtering at each step
result$filter_process

```

hypoglycemia_01

Hypoglycemia-01

Description

The hypoglycemia_01 function calculates the NEMSQA measure evaluating how often hypoglycemic patients with altered mental status receive hypoglycemia treatment.

Usage

```

hypoglycemia_01(
  df = NULL,
  patient_scene_table = NULL,
  response_table = NULL,
  situation_table = NULL,
  vitals_table = NULL,
  medications_table = NULL,
  procedures_table = NULL,
  erecord_01_col,

```

```

incident_date_col = NULL,
patient_DOB_col = NULL,
epatient_15_col,
epatient_16_col,
eresponse_05_col,
esituation_11_col,
esituation_12_col,
evitals_18_col,
evitals_23_col,
evitals_26_col,
emedications_03_col,
eprocedures_03_col,
confidence_interval = FALSE,
method = c("wilson", "clopper-pearson"),
conf.level = 0.95,
correct = TRUE,
...
)

```

Arguments

df A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.

patient_scene_table A data.frame or tibble containing at least ePatient, and eScene as a fact table.

response_table A data.frame or tibble containing at least the eResponse fields needed for this measure's calculations.

situation_table A data.frame or tibble containing at least the eSituation fields needed for this measure's calculations. Default is NULL.

vitals_table A dataframe or tibble containing at least the eVitals fields needed.

medications_table A data.frame or tibble containing at least the eMedications fields needed for this measure's calculations. Default is NULL.

procedures_table A dataframe or tibble containing at least the eProcedures fields needed.

erecord_01_col The column representing the EMS record unique identifier.

incident_date_col Column that contains the incident date. This defaults to NULL as it is optional in case not available due to PII restrictions.

patient_DOB_col Column that contains the patient's date of birth. This defaults to NULL as it is optional in case not available due to PII restrictions.

epatient_15_col Column representing the patient's numeric age agnostic of unit.

epatient_16_col Column representing the patient's age unit ("Years", "Months", "Days", "Hours", or "Minutes").

<code>eresponse_05_col</code>	Column that contains eResponse.05 or the response type.
<code>esituation_11_col</code>	Column that contains eSituation.11 provider primary impression data.
<code>esituation_12_col</code>	Column that contains all eSituation.12 values as (possible a single comma-separated list), provider secondary impression data.
<code>evitals_18_col</code>	Column for blood glucose levels.
<code>evitals_23_col</code>	Column for Glasgow Coma Scale (GCS) scores.
<code>evitals_26_col</code>	Column for AVPU alertness levels.
<code>emedications_03_col</code>	Column that contains all medication administered to the patient (eMedications.03) values as a single comma-separated list per distinct eRecord.01 ID.
<code>eprocedures_03_col</code>	Column containing procedure codes with or without procedure names.
<code>confidence_interval</code>	Logical. If TRUE, the function calculates a confidence interval for the proportion estimate.
<code>method</code>	Character. Specifies the method used to calculate confidence intervals. Options are "wilson" (Wilson score interval) and "clopper-pearson" (exact binomial interval). Partial matching is supported, so "w" and "c" can be used as shorthand.
<code>conf.level</code>	Numeric. The confidence level for the interval, expressed as a proportion (e.g., 0.95 for a 95% confidence interval). Defaults to 0.95.
<code>correct</code>	Logical. If TRUE, applies a continuity correction to the Wilson score interval when method = "wilson". Defaults to TRUE.
<code>...</code>	optional additional arguments to pass onto <code>dplyr::summarize</code> .

Value

A data.frame summarizing results for two population groups (All, Adults and Peds) with the following columns:

- `pop`: Population type (All, Adults, and Peds).
- `numerator`: Count of incidents meeting the measure.
- `denominator`: Total count of included incidents.
- `prop`: Proportion of incidents meeting the measure.
- `prop_label`: Proportion formatted as a percentage with a specified number of decimal places.
- `lower_ci`: Lower bound of the confidence interval for prop (if `confidence_interval = TRUE`).
- `upper_ci`: Upper bound of the confidence interval for prop (if `confidence_interval = TRUE`).

Author(s)

Nicolas Foss, Ed.D., MS

Examples

```

# Synthetic test data
test_data <- tibble::tibble(
  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  epatient_15 = c(34, 5, 45, 2, 60), # Ages
  epatient_16 = c("Years", "Years", "Years", "Months", "Years"),
  eresponse_05 = rep(2205001, 5),
  esituation_11 = c(rep("E13.64", 3), rep("E16.2", 2)),
  esituation_12 = c(rep("E13.64", 2), rep("E16.2", 3)),
  emedications_03 = c(372326, 376937,
                      377980, 4850,
                      4832),
  evitals_18 = c(60, 59, 58, 57, 56),
  evitals_23 = c(16, 15, 14, 13, 12),
  evitals_26 = c("Alert", "Painful", "Verbal", "Unresponsive", "Alert"),
  eprocedures_03 = rep("710925007", 5)
)

# Run the function
# Return 95% confidence intervals using the Wilson method
hypoglycemia_01(
  df = test_data,
  erecord_01_col = erecord_01,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col = epatient_15,
  epatient_16_col = epatient_16,
  eresponse_05_col = eresponse_05,
  esituation_11_col = esituation_11,
  esituation_12_col = esituation_12,
  emedications_03_col = emedications_03,
  evitals_18_col = evitals_18,
  evitals_23_col = evitals_23,
  evitals_26_col = evitals_26,
  eprocedures_03_col = eprocedures_03,
  confidence_interval = TRUE
)

```

hypoglycemia_01_population

Hypoglycemia-01 Populations

Description

Filters data down to the target populations for Hypoglycemia-01, and categorizes records to identify needed information for the calculations.

Identifies key categories related to diabetes/hypoglycemia incidents in an EMS dataset, specifically focusing on cases where 911 was called for diabetes/hypoglycemia distress, certain medications

were administered, and a weight is taken. This function segments the data into pediatric populations, computing the proportion of cases that have a documented weight.

Usage

```
hypoglycemia_01_population(
  df = NULL,
  patient_scene_table = NULL,
  response_table = NULL,
  situation_table = NULL,
  vitals_table = NULL,
  medications_table = NULL,
  procedures_table = NULL,
  erecord_01_col,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col,
  epatient_16_col,
  eresponse_05_col,
  esituation_11_col,
  esituation_12_col,
  evitals_18_col,
  evitals_23_col,
  evitals_26_col,
  emedications_03_col,
  eprocedures_03_col
)
```

Arguments

df A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.

patient_scene_table A data.frame or tibble containing at least ePatient, and eScene as a fact table.

response_table A data.frame or tibble containing at least the eResponse fields needed for this measure's calculations.

situation_table A data.frame or tibble containing at least the eSituation fields needed for this measure's calculations. Default is NULL.

vitals_table A dataframe or tibble containing at least the eVitals fields needed.

medications_table A data.frame or tibble containing at least the eMedications fields needed for this measure's calculations. Default is NULL.

procedures_table A dataframe or tibble containing at least the eProcedures fields needed.

erecord_01_col The column representing the EMS record unique identifier.

incident_date_col	Column that contains the incident date. This defaults to NULL as it is optional in case not available due to PII restrictions.
patient_DOB_col	Column that contains the patient's date of birth. This defaults to NULL as it is optional in case not available due to PII restrictions.
epatient_15_col	Column representing the patient's numeric age agnostic of unit.
epatient_16_col	Column representing the patient's age unit ("Years", "Months", "Days", "Hours", or "Minutes").
eresponse_05_col	Column that contains eResponse.05 or the response type.
esituation_11_col	Column that contains eSituation.11 provider primary impression data.
esituation_12_col	Column that contains all eSituation.12 values as (possible a single comma-separated list), provider secondary impression data.
evitals_18_col	Column for blood glucose levels.
evitals_23_col	Column for Glasgow Coma Scale (GCS) scores.
evitals_26_col	Column for AVPU alertness levels.
emedications_03_col	Column that contains all medication administered to the patient (eMedications.03) values as a single comma-separated list per distinct eRecord.01 ID.
eprocedures_03_col	Column containing procedure codes with or without procedure names.

Value

A list that contains the following:

- a tibble with counts for each filtering step,
- a tibble for each population of interest
- a tibble for the initial population
- a tibble for the total dataset with computations
- a tibble with a summary of missingness for each column in each table

Author(s)

Nicolas Foss, Ed.D., MS

Examples

```
# create tables to test correct functioning

# patient table
patient_table <- tibble::tibble(
```

```

erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
incident_date = as.Date(c("2025-01-01", "2025-01-05", "2025-02-01",
"2025-01-01", "2025-06-01")),
patient_dob = as.Date(c("2000-01-01", "2020-01-01", "2023-02-01",
"2023-01-01", "1970-06-01")),
epatient_15 = c(25, 5, 2, 2, 55), # Ages
epatient_16 = c("Years", "Years", "Years", "Years", "Years")

)

# response table
response_table <- tibble::tibble(

  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  eresponse_05 = rep(2205001, 5)

)

# situation table
situation_table <- tibble::tibble(

  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  esituation_11 = c(rep("E13.64", 3), rep("E16.2", 2)),
  esituation_12 = c(rep("E13.64", 2), rep("E16.2", 3))

)

# medications table
medications_table <- tibble::tibble(

  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  emedications_03 = c(372326, 376937, 377980, 4850, 4832),

)

# vitals table
vitals_table <- tibble::tibble(

  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  evitals_18 = c(60, 59, 58, 57, 56),
  evitals_23 = c(16, 15, 14, 13, 12),
  evitals_26 = c("Alert", "Painful", "Verbal", "Unresponsive", "Alert")

)

# procedures table
procedures_table <- tibble::tibble(

  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  eprocedures_03 = rep("710925007", 5)

)

```

```
# test the success of the function
result <- hypoglycemia_01_population(patient_scene_table = patient_table,
  response_table = response_table,
  situation_table = situation_table,
  medications_table = medications_table,
  vitals_table = vitals_table,
  procedures_table = procedures_table,
  erecord_01_col = erecord_01,
  incident_date_col = incident_date,
  patient_DOB_col = patient_dob,
  epatient_15_col = epatient_15,
  epatient_16_col = epatient_16,
  eresponse_05_col = eresponse_05,
  esituation_11_col = esituation_11,
  esituation_12_col = esituation_12,
  emedications_03_col = emedications_03,
  evitals_18_col = evitals_18,
  evitals_23_col = evitals_23,
  evitals_26_col = evitals_26,
  eprocedures_03_col = eprocedures_03
)

# show the results of filtering at each step
result$filter_process
```

nemsqar_airway_table *Synthetic Test Data for eAirway Fields in National EMS Information System*

Description

This dataset provides completely synthetic test data for the airway-related fields in the National EMS Information System (NEMSIS). It is not specific to any single function but can be used to test multiple functions in the nemsqar package. Users are encouraged to experiment with this dataset, but results should not be interpreted as meaningful. Some outputs may be nonsensical, which is expected since this data is only intended to demonstrate the expected structure of input data.

Usage

```
data(nemsqar_airway_table)
```

Format

A data frame with 10,000 rows and 8 variables:

Incident Patient Care Report Number - PCR (eRecord.01) Unique identifier for the incident report (character).

Incident Date Date of the incident (Date).

- Airway Indications For Invasive Management List (eAirway.01)** List of indications for invasive airway management (character).
- Airway Device Placement Confirmation Date Time (eAirway.02)** Timestamp of airway device placement confirmation (datetime).
- Airway Device Being Confirmed (eAirway.03)** Type of airway device being confirmed (character).
- Patient Airway Device Being Confirmed List (eAirway.03)** List of airway devices being confirmed (character).
- Airway Device Placement Confirmed Method (eAirway.04)** Primary method used to confirm airway device placement (character).
- Airway Device Placement Confirmed Method List (eAirway.04)** List of methods used to confirm airway device placement (character).

Examples

```
data(nemsqar_airway_table)
head(nemsqar_airway_table)
```

nemsqar_arrest_table *Synthetic Test Data for eArrest Fields in National EMS Information System*

Description

This dataset provides completely synthetic test data for the cardiac arrest-related fields in the National EMS Information System (NEMSIS). It is not specific to any single function but can be used to test multiple functions in the nemsqar package. Users are encouraged to experiment with this dataset, but results should not be interpreted as meaningful. Some outputs may be nonsensical, which is expected since this data is only intended to demonstrate the expected structure of input data.

Usage

```
data(nemsqar_arrest_table)
```

Format

A data frame with 10,000 rows and 28 variables:

- Incident Patient Care Report Number - PCR (eRecord.01)** Unique identifier for the incident report (character).
- Incident Date** Date of the incident (Date).
- Cardiac Arrest During EMS Event With Code (eArrest.01)** Indicates whether cardiac arrest occurred during the EMS event (character).

- Cardiac Arrest Etiology With Code (eArrest.02)** Suspected cause of the cardiac arrest (character).
- Cardiac Arrest Indications Resuscitation Attempted By EMS (eArrest.03)** Whether resuscitation was attempted by EMS (character).
- Cardiac Arrest Indications Resuscitation Attempted By EMS With Code List (eArrest.03)** Detailed reasons for resuscitation decisions (character).
- Cardiac Arrest Witnessed By (eArrest.04)** Who witnessed the cardiac arrest (character).
- Cardiac Arrest Witnessed By List (eArrest.04)** List of all witnesses to the cardiac arrest (character).
- Cardiac Arrest CPR Provided Prior To EMS Arrival (3.4=eArrest.05/3.5=itArrest.105)** Whether CPR was provided before EMS arrival (character).
- Cardiac Arrest Who Provided CPR Prior To EMS (3.4=eArrest.06/3.5=itArrest.106)** Who performed CPR before EMS arrival (character).
- Cardiac Arrest Who Provided CPR Prior To EMS Arrival With Code List (3.4=eArrest.06/3.5=itArrest.106)** List of responders who provided CPR before EMS arrival (character).
- Cardiac Arrest AED Use Prior To EMS Arrival (eArrest.07)** Whether an AED was used before EMS arrival (character).
- Cardiac Arrest Who Used AED Prior To EMS (3.4=eArrest.08/3.5=itArrest.108)** Who used the AED before EMS arrival (character).
- Cardiac Arrest Who Used AED Prior To EMS Arrival With Code List (3.4=eArrest.08/3.5=itArrest.108)** List of responders who used the AED before EMS arrival (character).
- Cardiac Arrest Type Of CPR Provided List (eArrest.09)** List of types of CPR performed (character).
- Cardiac Arrest First Monitored Arrest Rhythm Of Patient (eArrest.11)** First recorded cardiac rhythm during arrest (character).
- Cardiac Arrest First Monitored Arrest Rhythm Of Patient With Code (eArrest.11)** Coded representation of the first monitored cardiac rhythm (character).
- Cardiac Arrest Any Return Of Spontaneous Circulation (eArrest.12)** Whether the patient regained spontaneous circulation (character).
- Cardiac Arrest Any Return Of Spontaneous Circulation With Code List (eArrest.12)** List of codes indicating ROSC status (character).
- Cardiac Arrest Date Time (eArrest.14)** Timestamp of cardiac arrest occurrence (datetime).
- Cardiac Arrest Resuscitation Discontinued Date Time (eArrest.15)** Timestamp of resuscitation discontinuation (datetime).
- Cardiac Arrest Reason CPR Resuscitation Discontinued (eArrest.16)** Reason for discontinuing resuscitation (character).
- Cardiac Arrest Rhythm On Arrival At Destination List (eArrest.17)** Recorded cardiac rhythm upon arrival at the destination (character).
- Cardiac Arrest Patient Outcome At End Of EMS Event (eArrest.18)** Patient's condition at the end of the EMS event (character).
- Cardiac Arrest Initial CPR Date Time (eArrest.19)** Timestamp of initial CPR administration (datetime).

Cardiac Arrest Who Initiated CPR With Code (3.4=itArrest.008/3.5=eArrest.20) Who initiated CPR (character).

Cardiac Who First Applied The AED (3.4=itArrest.015/3.5=eArrest.21) Who first applied the AED (character).

Cardiac Who First Defibrillated The Patient (3.4=itArrest.013/3.5=eArrest.22) Who performed the first defibrillation (character).

Examples

```
data(nemsqar_arrest_table)
head(nemsqar_arrest_table)
```

```
nemsqar_disposition_table
```

Synthetic Test Data for eDisposition Fields in National EMS Information System

Description

This dataset provides completely synthetic test data for evaluating disposition-related functions in the nemsqar package. It includes key variables related to patient disposition, transport mode, hospital capabilities, and team pre-arrival alerts. The dataset is intended to assist users in testing the expected input structure for disposition-related measures.

Usage

```
data(nemsqar_disposition_table)
```

Format

A data frame with 10,000 rows and 13 variables:

Incident Patient Care Report Number - PCR (eRecord.01) Unique identifier for the incident report (character).

Incident Date Date of the incident (Date).

Disposition Position Of Patient During Transport List (eDisposition.14) Position of the patient during transport (character).

Disposition Additional Transport Mode Descriptor (eDisposition.18) Additional transport mode details (character).

Disposition Additional Transport Mode Descriptor List (eDisposition.18) List of additional transport mode descriptors (character).

Hospital Capability (3.4=itDisposition.105/3.5=eDisposition.23) Primary hospital capability associated with the incident (character).

Hospital Capability List (3.4=itDisposition.105/3.5=eDisposition.23) List of hospital capabilities at the destination facility (character).

- Disposition Team Pre Arrival Alert (eDisposition.24)** Indicates whether a pre-arrival alert was triggered for the team (character).
- Disposition Destination Team Pre-arrival Alert Activation (eDisposition.24)** Indicates the specific type of pre-arrival alert activated (character).
- Destination Trauma General Team Pre-arrival Activation (eDisposition.24)** Indicates whether a general trauma team was activated pre-arrival (logical).
- Patient Evaluation/Care (3.4=itDisposition.100/3.5=eDisposition.28)** Evaluation or care provided to the patient (character).
- Transport Disposition (3.4=itDisposition.102/3.5=eDisposition.30)** Final transport disposition of the patient (character).
- Disposition Incident Patient Disposition With Code (3.4=eDisposition.12/3.5=itDisposition.112)** Final disposition of the patient, including standardized codes (character).

Details

Users are encouraged to test these functions with this dataset, but results should not be interpreted as meaningful. Some outputs may be nonsensical, which is expected since this data is only intended to demonstrate the expected structure of input data.

Examples

```
data(nemsqar_disposition_table)
head(nemsqar_disposition_table)
```

nemsqar_exam_table	<i>Synthetic Test Data for eExam Fields in National EMS Information System</i>
--------------------	--

Description

This dataset provides completely synthetic test data for evaluating patient examination-related functions in the nemsqar package. It includes key variables related to patient weight, assessment findings, lung and chest examinations, and neurological assessments. The dataset is intended to assist users in testing the expected input structure for examination-related measures.

Usage

```
data(nemsqar_exam_table)
```

Format

A data frame with 10,000 rows and 11 variables:

- Incident Patient Care Report Number - PCR (eRecord.01)** Unique identifier for the incident report (character).

- Incident Date** Date of the incident (Date).
- Patient Weight In Kilograms (eExam.01)** Patient's weight in kilograms (numeric).
- Patient Length Based Color (eExam.02)** Color coding based on patient length for pediatric patients (character).
- Patient Assessment Date Time (eExam.03)** Timestamp for patient assessment (datetime).
- Patient Extremity Assessment Findings List (eExam.16)** List of findings related to extremity assessment (character).
- Patient Neurological Assessment Findings List (eExam.20)** List of neurological assessment findings (character).
- Patient Lung Assessment Finding Location (3.4=itExam.099/3.5=eExam.22)** Location of lung assessment findings (character).
- Patient Lung Assessment Findings List (3.4=itExam.100/3.5=eExam.23)** List of lung assessment findings (character).
- Patient ChestExclusive Assessment Finding Location (3.4=itExam.101/3.5=eExam.24)** Location of chest-exclusive assessment findings (character).
- Patient ChestExclusive Assessment Findings List (3.4=itExam.102/3.5=eExam.25)** List of chest-exclusive assessment findings (character).

Details

Users are encouraged to test these functions with this dataset, but results should not be interpreted as meaningful. Some outputs may be nonsensical, which is expected since this data is only intended to demonstrate the expected structure of input data.

Examples

```
data(nemsqar_exam_table)
head(nemsqar_exam_table)
```

nemsqar_injury_table *Synthetic Test Data for eInjury Fields in National EMS Information System*

Description

This dataset provides completely synthetic test data for evaluating injury-related functions in the nemsqar package. It includes key variables related to the cause of injury, trauma triage criteria, vehicular risk factors, and height of falls. The dataset is intended to assist users in testing the expected input structure for injury-related measures.

Usage

```
data(nemsqar_injury_table)
```

Format

A data frame with 10,000 rows and 8 variables:

Incident Patient Care Report Number - PCR (eRecord.01) Unique identifier for the incident report (character).

Incident Date Date of the incident (Date).

Injury Cause of Injury (eInjury.01) General description of the cause of injury (character).

Injury Cause Of Injury Description And Code List (eInjury.01) Detailed description and coding of injury causes (character).

Injury Trauma Center/Triage Criteria (Steps 1 and 2) List (eInjury.03) List of trauma triage criteria met in Steps 1 and 2 (character).

Injury Vehicular Pedestrian Or Other Injury Risk Factor/Triage Criteria (Steps 3 and 4) (eInjury.04) Primary vehicular or other risk factors for injury (character).

Injury Vehicular Pedestrian Or Other Injury Risk Factor/Triage Criteria (Steps 3 and 4) List (eInjury.04) Detailed list of vehicular or pedestrian injury risk factors (character).

Injury Height Of Fall In Feet (eInjury.09) Height of fall in feet when applicable (numeric).

Details

Users are encouraged to test these functions with this dataset, but results should not be interpreted as meaningful. Some outputs may be nonsensical, which is expected since this data is only intended to demonstrate the expected structure of input data.

Examples

```
data(nemsqar_injury_table)
head(nemsqar_injury_table)
```

nemsqar_medications_table

Synthetic Test Data for eMedications Fields in National EMS Information System

Description

This dataset provides completely synthetic test data for evaluating medication-related functions in the nemsqar package. It includes key variables related to medication administration, timing, route, and standardized coding. The dataset is designed to assist users in testing the expected input structure for medication-related measures.

Usage

```
data(nemsqar_medications_table)
```

Format

A data frame with 10,000 rows and 8 variables:

Incident Patient Care Report Number - PCR (eRecord.01) Unique identifier for the incident report (character).

Incident Date Date of the incident (Date).

Medication Administered Date Time (eMedications.01) Date and time the medication was administered (datetime).

Medication Administered Prior To EMS Unit Care (eMedications.02) Indicator of whether medication was administered before EMS arrival (character).

Medication Given or Administered Description And RXCUI Code (eMedications.03) Name of medication administered with its associated RXCUI code (character).

Patient Medication Given or Administered Description And RXCUI Codes List (eMedications.03) List of all medications administered with RXCUI codes (character).

Medication Administered Route (eMedications.04) Method by which the medication was administered (character).

Medication Administered Route Code (eMedications.04) Standardized code for medication administration route (character).

Details

Users are encouraged to test these functions with this dataset, but results should not be interpreted as meaningful. Some outputs may be nonsensical, which is expected since this data is only intended to demonstrate the expected structure of input data.

Examples

```
data(nemsqar_medications_table)
head(nemsqar_medications_table)
```

nemsqar_patient_scene_table

Synthetic ePatient Data from the National Emergency Medical Services Information System (NEMSIS)

Description

This dataset provides synthetic patient information from the ePatient section of the National Emergency Medical Services Information System (NEMSIS). It contains example patient demographics and scene-related details that can be used for testing various functions within the nemsqar package. Users are encouraged to test these functions with this dataset, but results should not be interpreted as meaningful. Some outputs may be nonsensical, which is expected since this data is only intended to demonstrate the expected structure of input data.

Usage

```
nemsqar_patient_scene_table
```

Format

A tibble with 10,000 rows and 6 variables:

Incident Patient Care Report Number - PCR (eRecord.01) (chr) Unique identifier for the patient care report.

Incident Date (date) The date of the EMS incident.

Patient Age (ePatient.15) (dbl) Patient's age at the time of the incident.

Patient Age Units (ePatient.16) (chr) The units of measurement for patient age (e.g., Years, Months, Hours, etc.).

Patient Date Of Birth (ePatient.17) (date) The patient's date of birth.

Patient Gender (ePatient.13) (chr) The patient's gender as recorded in the EMS report.

Details

The data in this table are entirely synthetic and intended solely for testing purposes. These data do not represent real patients, incidents, or outcomes and should not be used for research or operational decision-making.

Examples

```
data(nemsqar_patient_scene_table)
dplyr::glimpse(nemsqar_patient_scene_table)
```

```
nemsqar_procedures_table
```

Synthetic eProcedures Data from the National Emergency Medical Services Information System (NEMSIS)

Description

This dataset provides synthetic procedure-related information from the eProcedures section of the National Emergency Medical Services Information System (NEMSIS). It contains example procedure details that can be used for testing various functions within the nemsqar package. Users are encouraged to test these functions with this dataset, but results should not be interpreted as meaningful. Some outputs may be nonsensical, which is expected since this data is only intended to demonstrate the expected structure of input data.

Usage

```
nemsqar_procedures_table
```

Format

A tibble with 10,000 rows and 8 variables:

Incident Patient Care Report Number - PCR (eRecord.01) (chr) Unique identifier for the patient care report.

Incident Date (date) The date of the EMS incident.

Procedure Performed Date Time (eProcedures.01) (dtm) The date and time the procedure was performed.

Procedure Performed Prior To EMS Care (eProcedures.02) (chr) Indicates whether the procedure was performed before EMS arrival.

Procedure Performed Description And Code (eProcedures.03) (chr) Description and code of the performed procedure.

Patient Attempted Procedure Descriptions And Codes List (eProcedures.03) (chr) List of attempted procedures with descriptions and codes.

Procedure Number Of Attempts (eProcedures.05) (dbl) Number of attempts made to perform the procedure.

Procedure Successful (eProcedures.06) (chr) Indicates whether the procedure was successful.

Details

The data in this table are entirely synthetic and intended solely for testing purposes. These data do not represent real patients, incidents, or outcomes and should not be used for research or operational decision-making.

Examples

```
data(nemsqar_procedures_table)
dplyr::glimpse(nemsqar_procedures_table)
```

nemsqar_response_table

Synthetic eResponse Data from the National Emergency Medical Services Information System (NEMSIS)

Description

A synthetic dataset representing the eResponse section of the National Emergency Medical Services Information System (NEMSIS). This dataset is intended for testing purposes only and does not contain real patient or incident data. Users are encouraged to test functions with this dataset, but results should not be interpreted as meaningful. Some outputs may be nonsensical, which is expected since this data is only intended to demonstrate the expected structure of input data.

Usage

```
data(nemsqar_response_table)
```

Format

A tibble with 10,000 rows and 5 variables:

Incident Patient Care Report Number - PCR (eRecord.01) Character. Unique identifier for the patient care report.

Incident Date Date. The date of the EMS incident.

Response Type Of Service Requested With Code (eResponse.05) Character. The type of service requested, including a coded value.

Response Type Of Scene Delay List (eResponse.10) Character. A list of delay types encountered at the scene.

Response Additional Response Mode Descriptors List (eResponse.24) Character. A list of additional response mode descriptors.

nemsqar_situation_table

Synthetic Test Data for eSituation Fields in National EMS Information System

Description

This dataset provides completely synthetic test data for evaluating situation-related functions in the nemsqar package. It includes variables related to patient complaints, symptoms, injury status, and provider impressions. The dataset is designed to assist users in testing the expected input structure for situation-related measures.

Usage

```
data(nemsqar_situation_table)
```

Format

A data frame with 10,000 rows and 18 variables:

Incident Patient Care Report Number - PCR (eRecord.01) Unique identifier for the incident report (character).

Incident Date Date of the incident (Date).

Situation Symptom Onset Date Time (eSituation.01) Date and time when symptoms began (datetime).

Situation Possible Injury With Code (eSituation.02) Indicates whether an injury is possible, including coded response (character).

Situation Complaint Type (eSituation.03) Classification of the patient's complaint (character).

- Situation Complaint Statement (eSituation.04)** Primary complaint reported by the patient (character).
- Situation Primary Complaint Statement List (eSituation.04)** List of primary complaints (character).
- Situation Complaint Duration (eSituation.05)** Duration of the complaint (numeric).
- Situation Complaint Duration Time Units (eSituation.06)** Units of time associated with the complaint duration (character).
- Situation Chief Complaint Anatomic Location (eSituation.07)** Anatomic location of the primary complaint (character).
- Situation Chief Complaint Organ System (eSituation.08)** Organ system affected by the chief complaint (character).
- Situation Primary Symptom (eSituation.09)** Primary symptom reported by the patient, including ICD code (character).
- Situation Other Associated Symptom Description (eSituation.10)** Description of additional symptoms (character).
- Situation Other Associated Symptom ICD Code (eSituation.10)** ICD code for associated symptoms (character).
- Situation Other Associated Symptoms List (eSituation.10)** List of additional symptoms reported (character).
- Situation Provider Primary Impression Code And Description (eSituation.11)** Primary impression of the provider, including ICD code (character).
- Situation Provider Secondary Impression Description And Code (eSituation.12)** Secondary provider impression, including ICD code (character).
- Situation Provider Secondary Impression Description And Code List (eSituation.12)** List of secondary provider impressions (character).

Details

Users are encouraged to test these functions with this dataset, but results should not be interpreted as meaningful. Some outputs may be nonsensical, which is expected since this data is only intended to demonstrate the expected structure of input data.

Examples

```
data(nemsqar_situation_table)
head(nemsqar_situation_table)
```

nemsqar_vitals_table	<i>Synthetic eVitals Data for NEMSIS from the National Emergency Medical Services Information System (NEMSIS)</i>
----------------------	---

Description

This dataset contains synthetic data for the eVitals section of the National Emergency Medical Services Information System (NEMSIS). It is designed for testing functions within the nemsqar package. The data structure follows the expected format for eVitals fields but does not produce meaningful clinical results.

Usage

```
nemsqar_vitals_table
```

Format

A tibble with 10,000 rows and 19 variables:

Incident Patient Care Report Number - PCR (eRecord.01) Unique incident identifier (character).

Incident Date Date of the EMS incident (Date).

Vitals Signs Taken Date Time (eVitals.01) Timestamp of vital signs measurement (Datetime).

Vitals ECG Type (eVitals.04) ECG type recorded (character).

Vitals Systolic Blood Pressure SBP (eVitals.06) Systolic blood pressure (numeric).

Vitals Diastolic Blood Pressure DBP (eVitals.07) Diastolic blood pressure (numeric).

Vitals Heart Rate (eVitals.10) Heart rate in beats per minute (numeric).

Vitals Pulse Oximetry (eVitals.12) Oxygen saturation percentage (numeric).

Vitals Respiratory Rate (eVitals.14) Respiratory rate in breaths per minute (numeric).

Vitals Respiratory Effort (eVitals.15) Observed respiratory effort (character).

Vitals Carbon Dioxide CO2 (eVitals.16) End-tidal CO2 measurement (numeric).

Vitals Blood Glucose Level (eVitals.18) Blood glucose level in mg/dL (numeric).

Vitals Glasgow Coma Score GCS Motor (eVitals.21) GCS motor response score (character).

Vitals Total Glasgow Coma Score GCS (eVitals.23) Total Glasgow Coma Score (numeric).

Vitals Level Of Responsiveness AVPU (eVitals.26) AVPU scale assessment (character).

Vitals Pain Scale Score (eVitals.27) Pain scale score (numeric).

Vitals Pain Scale Score Range Sort Order (eVitals.27) Sort order for pain scale (numeric).

Vitals Stroke Scale Score (eVitals.29) Stroke scale assessment result (character).

Vitals Stroke Scale Type (eVitals.30) Type of stroke scale used (character).

Details

Users are encouraged to use this dataset to test functions, but the outputs may be nonsensical, as the data is solely intended to demonstrate the expected input structure.

Examples

```
data(nemsqar_vitals_table)
dplyr::glimpse(nemsqar_vitals_table)
```

```
nemsqa_binomial_confint
```

Wilson and Clopper-Pearson Confidence Intervals for Binomial Proportions

Description

Computes confidence intervals for binomial proportions using either the Wilson or Clopper-Pearson method. This function supports vectorized operations and allows optional correction for continuity. The Wilson interval is computed using `stats::prop.test()`, while the Clopper-Pearson interval is computed using `stats::binom.test()`.

Usage

```
nemsqa_binomial_confint(
  data = NULL,
  x,
  n,
  method = c("wilson", "clopper-pearson"),
  conf.level = 0.95,
  correct = TRUE
)
```

Arguments

<code>data</code>	An optional tibble or data.frame containing the variables x and n. If provided, x and n should be column names.
<code>x</code>	Numeric vector or column name (if data is provided) representing the number of successes.
<code>n</code>	Numeric vector or column name (if data is provided) representing the total number of trials.
<code>method</code>	Character string specifying the confidence interval method. Must be either "wilson" (default) or "clopper-pearson".
<code>conf.level</code>	Numeric value between 0 and 1 indicating the confidence level. Defaults to 0.95 (95% confidence interval).
<code>correct</code>	Logical, indicating whether to apply continuity correction for Wilson intervals. Defaults to TRUE.

Details

The Wilson confidence interval is calculated using `stats::prop.test()`, which provides an improved approximation to the binomial proportion confidence interval by avoiding the instability of the Wald interval (Wilson, 1927). The Clopper-Pearson interval, computed using `stats::binom.test()`, is an exact method based on the cumulative probabilities of the binomial distribution (Clopper & Pearson, 1934).

The use of `match.arg()` within `nemsqa::nemsqa_binomial_confint()` allows users to specify the method using partial matching, meaning they can enter just "w" instead of "wilson" or "c" instead of "clopper-pearson".

Value

A tibble containing the estimated proportion (`prop`), lower confidence interval (`lower_ci`), upper confidence interval (`upper_ci`), and a formatted proportion label (`prop_label`). If data is provided, these columns are appended to data via `dplyr::bind_cols()`.

References

Clopper, C. J. & Pearson, E. S. (1934). The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26, 404–413. doi:10.2307/2331986.

Wilson, E.B. (1927). Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22, 209–212. doi:10.2307/2276774.

Examples

```
# Example without a data frame
nemsqa_binomial_confint(data = NULL,
  x = c(5, 10, 20),
  n = c(50, 100, 200),
  method = "wilson"
)

# Example with a data.frame
data <- data.frame(successes = c(5, 10, 20), trials = c(50, 100, 200))
nemsqa_binomial_confint(data,
  x = successes,
  n = trials,
  method = "clopper-pearson"
)
```

Description

The function calculates a pediatric metric focused on EMS responses, specifically targeting responses that involve patients under 18 years of age, where certain weight-based medications were administered. This function filters EMS data to identify relevant 911 responses and further narrows down the dataset to cases involving children, calculating the proportion of cases with documented weight among those where weight-based medications were administered.

Usage

```
pediatrics_03b(
  df = NULL,
  patient_scene_table = NULL,
  response_table = NULL,
  exam_table = NULL,
  medications_table = NULL,
  erecord_01_col,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col,
  epatient_16_col,
  eresponse_05_col,
  eexam_01_col,
  eexam_02_col,
  emedications_03_col,
  emedications_04_col,
  confidence_interval = FALSE,
  method = c("wilson", "clopper-pearson"),
  conf.level = 0.95,
  correct = TRUE,
  ...
)
```

Arguments

<code>df</code>	A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.
<code>patient_scene_table</code>	A data.frame or tibble containing at least <code>ePatient</code> , and <code>eScene</code> as a fact table.
<code>response_table</code>	A data.frame or tibble containing at least the <code>eResponse</code> fields needed for this measure's calculations.
<code>exam_table</code>	A data.frame or tibble containing only the <code>eExam</code> fields needed for this measure's calculations. Default is <code>NULL</code> .
<code>medications_table</code>	A data.frame or tibble containing at least the <code>eMedications</code> fields needed for this measure's calculations. Default is <code>NULL</code> .
<code>erecord_01_col</code>	The column representing the EMS record unique identifier.

<code>incident_date_col</code>	Column that contains the incident date. This defaults to NULL as it is optional in case not available due to PII restrictions.
<code>patient_DOB_col</code>	Column that contains the patient's date of birth. This defaults to NULL as it is optional in case not available due to PII restrictions.
<code>epatient_15_col</code>	Column representing the patient's numeric age agnostic of unit.
<code>epatient_16_col</code>	Column representing the patient's age unit ("Years", "Months", "Days", "Hours", or "Minutes").
<code>eresponse_05_col</code>	Column that contains eResponse.05 or the response type.
<code>eexam_01_col</code>	Column containing estimated patient body weight in kilograms.
<code>eexam_02_col</code>	Column containing data on length based tape measure for patients.
<code>emedications_03_col</code>	Column that contains all medication administered to the patient (eMedications.03) values as a single comma-separated list per distinct eRecord.01 ID.
<code>emedications_04_col</code>	Column indicating route medication was administered to the patient.
<code>confidence_interval</code>	Logical. If TRUE, the function calculates a confidence interval for the proportion estimate.
<code>method</code>	Character. Specifies the method used to calculate confidence intervals. Options are "wilson" (Wilson score interval) and "clopper-pearson" (exact binomial interval). Partial matching is supported, so "w" and "c" can be used as shorthand.
<code>conf.level</code>	Numeric. The confidence level for the interval, expressed as a proportion (e.g., 0.95 for a 95% confidence interval). Defaults to 0.95.
<code>correct</code>	Logical. If TRUE, applies a continuity correction to the Wilson score interval when <code>method = "wilson"</code> . Defaults to TRUE.
<code>...</code>	optional additional arguments to pass onto <code>dplyr::summarize</code> .

Value

A data.frame summarizing results for two population groups (Peds) with the following columns:

- `pop`: Population type (Peds).
- `numerator`: Count of incidents meeting the measure.
- `denominator`: Total count of included incidents.
- `prop`: Proportion of incidents meeting the measure.
- `prop_label`: Proportion formatted as a percentage with a specified number of decimal places.
- `lower_ci`: Lower bound of the confidence interval for `prop` (if `confidence_interval = TRUE`).
- `upper_ci`: Upper bound of the confidence interval for `prop` (if `confidence_interval = TRUE`).

Author(s)

Nicolas Foss, Ed.D., MS

Examples

```
# Synthetic test data
test_data <- tibble::tibble(
  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  incident_date = as.Date(c("2025-01-01", "2025-01-05", "2025-02-01",
    "2025-06-01", "2025-12-15")),
  patient_dob = as.Date(c("2021-01-01", "2020-01-01", "2022-02-01",
    "2023-06-01", "2019-12-15")),
  epatient_15 = c(4, 5, 3, 2, 6), # Ages
  epatient_16 = c("Years", "Years", "Years", "Years", "Years"),
  eresponse_05 = rep(2205001, 5),
  emedications_03 = rep("stuff", 5),
  emedications_04 = c("Inhalation", "pill", "liquid", "pill", "liquid"),
  eexam_01 = c(60, 59, 58, 57, 56),
  eexam_02 = c("Red", "Purple", "Grey", "Yellow", "Orange")
)

# Run the function
# Return 95% confidence intervals using the Wilson method
pediatrics_03b(
  df = test_data,
  erecord_01_col = erecord_01,
  incident_date_col = incident_date,
  patient_DOB_col = patient_dob,
  epatient_15_col = epatient_15,
  epatient_16_col = epatient_16,
  eresponse_05_col = eresponse_05,
  emedications_03_col = emedications_03,
  emedications_04_col = emedications_04,
  eexam_01_col = eexam_01,
  eexam_02_col = eexam_02,
  confidence_interval = TRUE
)
```

pediatrics_03b_population

Pediatrics-03B Populations

Description

Filters data down to the target populations for Pediatrics-03B, and categorizes records to identify needed information for the calculations.

Identifies key categories related to diabetes/hypoglycemia incidents in an EMS dataset, specifically focusing on cases where 911 was called for diabetes/hypoglycemia distress, certain medications

were administered, and a weight is taken. This function segments the data into pediatric populations, computing the proportion of cases that have a documented weight.

Usage

```
pediatrics_03b_population(
  df = NULL,
  patient_scene_table = NULL,
  response_table = NULL,
  exam_table = NULL,
  medications_table = NULL,
  erecord_01_col,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col,
  epatient_16_col,
  eresponse_05_col,
  eexam_01_col,
  eexam_02_col,
  emedications_03_col,
  emedications_04_col
)
```

Arguments

<code>df</code>	A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.
<code>patient_scene_table</code>	A data.frame or tibble containing at least ePatient, and eScene as a fact table.
<code>response_table</code>	A data.frame or tibble containing at least the eResponse fields needed for this measure's calculations.
<code>exam_table</code>	A data.frame or tibble containing only the eExam fields needed for this measure's calculations. Default is NULL.
<code>medications_table</code>	A data.frame or tibble containing at least the eMedications fields needed for this measure's calculations. Default is NULL.
<code>erecord_01_col</code>	The column representing the EMS record unique identifier.
<code>incident_date_col</code>	Column that contains the incident date. This defaults to NULL as it is optional in case not available due to PII restrictions.
<code>patient_DOB_col</code>	Column that contains the patient's date of birth. This defaults to NULL as it is optional in case not available due to PII restrictions.
<code>epatient_15_col</code>	Column representing the patient's numeric age agnostic of unit.
<code>epatient_16_col</code>	Column representing the patient's age unit ("Years", "Months", "Days", "Hours", or "Minutes").

<code>eresponse_05_col</code>	Column that contains eResponse.05 or the response type.
<code>eexam_01_col</code>	Column containing estimated patient body weight in kilograms.
<code>eexam_02_col</code>	Column containing data on length based tape measure for patients.
<code>emedications_03_col</code>	Column that contains all medication administered to the patient (eMedications.03) values as a single comma-separated list per distinct eRecord.01 ID.
<code>emedications_04_col</code>	Column indicating route medication was administered to the patient.

Value

A list that contains the following:

- a tibble with counts for each filtering step,
- a tibble for each population of interest
- a tibble for the initial population
- a tibble for the total dataset with computations
- a tibble with a summary of missingness for each column in each table

Author(s)

Nicolas Foss, Ed.D., MS

Examples

```
# create tables to test correct functioning
patient_table <- tibble::tibble(

  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  incident_date = as.Date(c("2025-01-01", "2025-01-05",
                           "2025-02-01", "2025-01-01",
                           "2025-06-01")
                        ),
  patient_dob = as.Date(c("2000-01-01", "2020-01-01",
                         "2023-02-01", "2023-01-01",
                         "1970-06-01")
                     ),
  epatient_15 = c(25, 5, 2, 2, 55), # Ages
  epatient_16 = c("Years", "Years", "Years", "Years", "Years")

)

response_table <- tibble::tibble(

  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  eresponse_05 = rep(2205001, 5)

)
```

```

exam_table <- tibble::tibble(
  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  eexam_01 = c(60, 59, 58, 57, 56),
  eexam_02 = c("Red", "Purple", "Grey", "Yellow", "Orange")
)

medications_table <- tibble::tibble(
  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  emedications_03 = rep("stuff", 5),
  emedications_04 = c("Inhalation", "pill", "liquid", "pill", "liquid"),
)

# test the success of the function

result <- pediatrics_03b_population(patient_scene_table = patient_table,
  response_table = response_table,
  exam_table = exam_table,
  medications_table = medications_table,
  erecord_01_col = erecord_01,
  incident_date_col = incident_date,
  patient_DOB_col = patient_dob,
  epatient_15_col = epatient_15,
  epatient_16_col = epatient_16,
  eresponse_05_col = eresponse_05,
  emedications_03_col = emedications_03,
  emedications_04_col = emedications_04,
  eexam_01_col = eexam_01,
  eexam_02_col = eexam_02
)

# show the results of filtering at each step
result$filter_process

```

pretty_percent

Format Numeric Variables as Percentages

Description

This function formats numeric variables as percentages with a specified number of decimal places. It refines the output by removing unnecessary trailing zeros after the decimal point and ensures the percentage sign is correctly applied without extraneous characters, resulting in a polished, human-readable percentage representation.

Usage

```
pretty_percent(variable, n_decimal = 1)
```

Arguments

variable	A numeric vector representing proportions to format as percentages. The values are on a scale from 0 to 1.
n_decimal	A numeric value specifying the number of decimal places. Defaults to 1.

Value

A character vector containing the formatted percentages.

Author(s)

Nicolas Foss, Ed.D., MS

Examples

```
# Example usage:
pretty_percent(0.12345) # Default decimal places
pretty_percent(0.12345, n_decimal = 2) # Two decimal places
pretty_percent(c(0.1, 0.25, 0.3333), n_decimal = 1) # Vector input
```

respiratory_01

Respiratory-01 Calculation

Description

The `respiratory_01` function filters and analyzes data related to emergency 911 respiratory distress incidents, providing summary statistics for adult and pediatric populations. This function uses specific data columns for 911 response codes, primary and secondary impressions, and vital signs to calculate the proportion of cases with complete vital signs recorded, stratified by age.

Usage

```
respiratory_01(  
  df = NULL,  
  patient_scene_table = NULL,  
  response_table = NULL,  
  situation_table = NULL,  
  vitals_table = NULL,  
  erecord_01_col,  
  incident_date_col = NULL,  
  patient_DOB_col = NULL,  
  epatient_15_col,
```

```

epatient_16_col,
eresponse_05_col,
esituation_11_col,
esituation_12_col,
evitals_12_col,
evitals_14_col,
confidence_interval = FALSE,
method = c("wilson", "clopper-pearson"),
conf.level = 0.95,
correct = TRUE,
...
)

```

Arguments

df A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.

patient_scene_table A data.frame or tibble containing at least ePatient, and eScene as a fact table.

response_table A data.frame or tibble containing at least the eResponse fields needed for this measure's calculations.

situation_table A data.frame or tibble containing at least the eSituation fields needed for this measure's calculations. Default is NULL.

vitals_table A dataframe or tibble containing at least the eVitals fields needed.

erecord_01_col The column representing the EMS record unique identifier.

incident_date_col Column that contains the incident date. This defaults to NULL as it is optional in case not available due to PII restrictions.

patient_DOB_col Column that contains the patient's date of birth. This defaults to NULL as it is optional in case not available due to PII restrictions.

epatient_15_col Column representing the patient's numeric age agnostic of unit.

epatient_16_col Column representing the patient's age unit ("Years", "Months", "Days", "Hours", or "Minutes").

eresponse_05_col Column that contains eResponse.05 or the response type.

esituation_11_col Column that contains eSituation.11 provider primary impression data.

esituation_12_col Column that contains all eSituation.12 values as (possible a single comma-separated list), provider secondary impression data.

evitals_12_col Numeric column containing pulse oximetry values.

evitals_14_col	Column containing data on patient's respiratory rate expressed as a number per minute.
confidence_interval	Logical. If TRUE, the function calculates a confidence interval for the proportion estimate.
method	Character. Specifies the method used to calculate confidence intervals. Options are "wilson" (Wilson score interval) and "clopper-pearson" (exact binomial interval). Partial matching is supported, so "w" and "c" can be used as shorthand.
conf.level	Numeric. The confidence level for the interval, expressed as a proportion (e.g., 0.95 for a 95% confidence interval). Defaults to 0.95.
correct	Logical. If TRUE, applies a continuity correction to the Wilson score interval when method = "wilson". Defaults to TRUE.
...	optional additional arguments to pass onto <code>dplyr::summarize</code> .

Value

A `data.frame` summarizing results for two population groups (All, Adults and Peds) with the following columns:

- `pop`: Population type (All, Adults, and Peds).
- `numerator`: Count of incidents meeting the measure.
- `denominator`: Total count of included incidents.
- `prop`: Proportion of incidents meeting the measure.
- `prop_label`: Proportion formatted as a percentage with a specified number of decimal places.
- `lower_ci`: Lower bound of the confidence interval for `prop` (if `confidence_interval = TRUE`).
- `upper_ci`: Upper bound of the confidence interval for `prop` (if `confidence_interval = TRUE`).

Author(s)

Nicolas Foss, Ed.D., MS

Examples

```
# Synthetic test data
test_data <- tibble::tibble(
  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  epatient_15 = c(34, 5, 45, 2, 60), # Ages
  epatient_16 = c("Years", "Years", "Years", "Months", "Years"),
  eresponse_05 = rep(2205001, 5),
  esituation_11 = c(rep("J80", 3), rep("I50.9", 2)),
  esituation_12 = c(rep("J80", 2), rep("I50.9", 3)),
  evitals_12 = c(60, 59, 58, 57, 56),
  evitals_14 = c(16, 15, 14, 13, 12)
)
```

```

# Run the function
# Return 95% confidence intervals using the Wilson method
respiratory_01(
  df = test_data,
  erecord_01_col = erecord_01,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col = epatient_15,
  epatient_16_col = epatient_16,
  eresponse_05_col = eresponse_05,
  esituation_11_col = esituation_11,
  esituation_12_col = esituation_12,
  evitals_12_col = evitals_12,
  evitals_14_col = evitals_14,
  confidence_interval = TRUE
)

```

```
respiratory_01_population
```

Respiratory-01 Populations

Description

The `respiratory_01_population` function filters and analyzes data related to emergency 911 respiratory distress incidents, providing the adult, pediatric, and initial populations. This function uses specific data columns for 911 response codes, primary and secondary impressions, and vital signs to filter a dataset down to the populations of interest.

Usage

```

respiratory_01_population(
  df = NULL,
  patient_scene_table = NULL,
  response_table = NULL,
  situation_table = NULL,
  vitals_table = NULL,
  erecord_01_col,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col,
  epatient_16_col,
  eresponse_05_col,
  esituation_11_col,
  esituation_12_col,
  evitals_12_col,
  evitals_14_col
)

```

Arguments

<code>df</code>	A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.
<code>patient_scene_table</code>	A data.frame or tibble containing at least ePatient, and eScene as a fact table.
<code>response_table</code>	A data.frame or tibble containing at least the eResponse fields needed for this measure's calculations.
<code>situation_table</code>	A data.frame or tibble containing at least the eSituation fields needed for this measure's calculations. Default is NULL.
<code>vitals_table</code>	A dataframe or tibble containing at least the eVitals fields needed.
<code>erecord_01_col</code>	The column representing the EMS record unique identifier.
<code>incident_date_col</code>	Column that contains the incident date. This defaults to NULL as it is optional in case not available due to PII restrictions.
<code>patient_DOB_col</code>	Column that contains the patient's date of birth. This defaults to NULL as it is optional in case not available due to PII restrictions.
<code>epatient_15_col</code>	Column representing the patient's numeric age agnostic of unit.
<code>epatient_16_col</code>	Column representing the patient's age unit ("Years", "Months", "Days", "Hours", or "Minutes").
<code>eresponse_05_col</code>	Column that contains eResponse.05 or the response type.
<code>esituation_11_col</code>	Column that contains eSituation.11 provider primary impression data.
<code>esituation_12_col</code>	Column that contains all eSituation.12 values as (possible a single comma-separated list), provider secondary impression data.
<code>evitals_12_col</code>	Numeric column containing pulse oximetry values.
<code>evitals_14_col</code>	Column containing data on patient's respiratory rate expressed as a number per minute.

Value

A list that contains the following:

- a tibble with counts for each filtering step,
- a tibble for each population of interest
- a tibble for the initial population
- a tibble for the total dataset with computations
- a tibble with a summary of missingness for each column in each table


```

        situation_table = situation_table,
        vitals_table = vitals_table,
        erecord_01_col = erecord_01,
        incident_date_col = incident_date,
        patient_DOB_col = patient_dob,
        epatient_15_col = epatient_15,
        epatient_16_col = epatient_16,
        eresponse_05_col = eresponse_05,
        esituation_11_col = esituation_11,
        esituation_12_col = esituation_12,
        evitals_12_col = evitals_12,
        evitals_14_col = evitals_14
    )

# show the results of filtering at each step
result$filter_process

```

respiratory_02

Respiratory-02 Calculation

Description

The `respiratory_02` function calculates metrics for pediatric and adult respiratory populations based on pre-defined criteria, such as low oxygen saturation and specific medication or procedure codes. It returns a summary table of the overall, pediatric, and adult populations, showing counts and proportions.

Usage

```

respiratory_02(
  df = NULL,
  patient_scene_table = NULL,
  response_table = NULL,
  vitals_table = NULL,
  medications_table = NULL,
  procedures_table = NULL,
  erecord_01_col,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col,
  epatient_16_col,
  eresponse_05_col,
  evitals_12_col,
  emedications_03_col,
  eprocedures_03_col,
  confidence_interval = FALSE,
  method = c("wilson", "clopper-pearson"),

```

```

    conf.level = 0.95,
    correct = TRUE,
    ...
)

```

Arguments

<code>df</code>	A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.
<code>patient_scene_table</code>	A data.frame or tibble containing at least ePatient, and eScene as a fact table.
<code>response_table</code>	A data.frame or tibble containing at least the eResponse fields needed for this measure's calculations.
<code>vitals_table</code>	A dataframe or tibble containing at least the eVitals fields needed.
<code>medications_table</code>	A data.frame or tibble containing at least the eMedications fields needed for this measure's calculations. Default is NULL.
<code>procedures_table</code>	A dataframe or tibble containing at least the eProcedures fields needed.
<code>erecord_01_col</code>	The column representing the EMS record unique identifier.
<code>incident_date_col</code>	Column that contains the incident date. This defaults to NULL as it is optional in case not available due to PII restrictions.
<code>patient_DOB_col</code>	Column that contains the patient's date of birth. This defaults to NULL as it is optional in case not available due to PII restrictions.
<code>epatient_15_col</code>	Column representing the patient's numeric age agnostic of unit.
<code>epatient_16_col</code>	Column representing the patient's age unit ("Years", "Months", "Days", "Hours", or "Minutes").
<code>eresponse_05_col</code>	Column that contains eResponse.05 or the response type.
<code>evitals_12_col</code>	Numeric column containing pulse oximetry values.
<code>emedications_03_col</code>	Column that contains all medication administered to the patient (eMedications.03) values as a single comma-separated list per distinct eRecord.01 ID.
<code>eprocedures_03_col</code>	Column containing procedure codes with or without procedure names.
<code>confidence_interval</code>	Logical. If TRUE, the function calculates a confidence interval for the proportion estimate.
<code>method</code>	Character. Specifies the method used to calculate confidence intervals. Options are "wilson" (Wilson score interval) and "clopper-pearson" (exact binomial interval). Partial matching is supported, so "w" and "c" can be used as shorthand.

<code>conf.level</code>	Numeric. The confidence level for the interval, expressed as a proportion (e.g., 0.95 for a 95% confidence interval). Defaults to 0.95.
<code>correct</code>	Logical. If TRUE, applies a continuity correction to the Wilson score interval when <code>method = "wilson"</code> . Defaults to TRUE.
<code>...</code>	optional additional arguments to pass onto <code>dplyr::summarize</code> .

Value

A `data.frame` summarizing results for two population groups (All, Adults and Peds) with the following columns:

- `pop`: Population type (All, Adults, and Peds).
- `numerator`: Count of incidents meeting the measure.
- `denominator`: Total count of included incidents.
- `prop`: Proportion of incidents meeting the measure.
- `prop_label`: Proportion formatted as a percentage with a specified number of decimal places.
- `lower_ci`: Lower bound of the confidence interval for `prop` (if `confidence_interval = TRUE`).
- `upper_ci`: Upper bound of the confidence interval for `prop` (if `confidence_interval = TRUE`).

Author(s)

Nicolas Foss, Ed.D., MS

Examples

```
# Synthetic test data
test_data <- tibble::tibble(
  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  epatient_15 = c(34, 5, 45, 2, 60), # Ages
  epatient_16 = c("Years", "Years", "Years", "Months", "Years"),
  eresponse_05 = rep(2205001, 5),
  emedications_03 = c("Oxygen", "Oxygen", "Oxygen", "Oxygen", "Oxygen"),
  evitals_12 = c(60, 59, 58, 57, 56),
  eprocedures_03 = rep("applicable thing", 5)
)

# Run the function
# Return 95% confidence intervals using the Wilson method
respiratory_02(
  df = test_data,
  erecord_01_col = erecord_01,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col = epatient_15,
  epatient_16_col = epatient_16,
  eresponse_05_col = eresponse_05,
  emedications_03_col = emedications_03,
```

```

    evitals_12_col = evitals_12,
    eprocedures_03_col = eprocedures_03,
    confidence_interval = TRUE
  )

```

respiratory_02_population

Respiratory-02 Populations

Description

The `respiratory_02_population` function calculates metrics for pediatric and adult respiratory populations based on pre-defined criteria, such as low oxygen saturation and specific medication or procedure codes. It returns a summary table of the overall, pediatric, and adult populations, showing counts and proportions.

Usage

```

respiratory_02_population(
  df = NULL,
  patient_scene_table = NULL,
  response_table = NULL,
  vitals_table = NULL,
  medications_table = NULL,
  procedures_table = NULL,
  erecord_01_col,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col,
  epatient_16_col,
  eresponse_05_col,
  evitals_12_col,
  emedications_03_col,
  eprocedures_03_col
)

```

Arguments

<code>df</code>	A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.
<code>patient_scene_table</code>	A data.frame or tibble containing at least <code>ePatient</code> , and <code>eScene</code> as a fact table.
<code>response_table</code>	A data.frame or tibble containing at least the <code>eResponse</code> fields needed for this measure's calculations.
<code>vitals_table</code>	A dataframe or tibble containing at least the <code>eVitals</code> fields needed.

<code>medications_table</code>	A data.frame or tibble containing at least the eMedications fields needed for this measure's calculations. Default is NULL.
<code>procedures_table</code>	A dataframe or tibble containing at least the eProcedures fields needed.
<code>erecord_01_col</code>	The column representing the EMS record unique identifier.
<code>incident_date_col</code>	Column that contains the incident date. This defaults to NULL as it is optional in case not available due to PII restrictions.
<code>patient_DOB_col</code>	Column that contains the patient's date of birth. This defaults to NULL as it is optional in case not available due to PII restrictions.
<code>epatient_15_col</code>	Column representing the patient's numeric age agnostic of unit.
<code>epatient_16_col</code>	Column representing the patient's age unit ("Years", "Months", "Days", "Hours", or "Minutes").
<code>eresponse_05_col</code>	Column that contains eResponse.05 or the response type.
<code>evitals_12_col</code>	Numeric column containing pulse oximetry values.
<code>emedications_03_col</code>	Column that contains all medication administered to the patient (eMedications.03) values as a single comma-separated list per distinct eRecord.01 ID.
<code>eprocedures_03_col</code>	Column containing procedure codes with or without procedure names.

Value

A list that contains the following:

- a tibble with counts for each filtering step,
- a tibble for each population of interest
- a tibble for the initial population
- a tibble for the total dataset with computations
- a tibble with a summary of missingness for each column in each table

Author(s)

Nicolas Foss, Ed.D., MS

Examples

```
# create tables to test correct functioning
# patient table
patient_table <- tibble::tibble(
  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
```



```

    patient_DOB_col = patient_dob,
    epatient_15_col = epatient_15,
    epatient_16_col = epatient_16,
    eresponse_05_col = eresponse_05,
    emedications_03_col = emedications_03,
    evitals_12_col = evitals_12,
    eprocedures_03_col = eprocedures_03
  )

# show the results of filtering at each step
result$filter_process

```

safety_01

Safety-01 Calculation

Description

The `safety_01` function calculates the proportion of 911 responses where "lights and sirens" were not used in an EMS dataset. It generates age-based population summaries, calculating the count and proportion of "lights and sirens" responses among all incidents, and within adult and pediatric groups.

Usage

```

safety_01(
  df = NULL,
  patient_scene_table = NULL,
  response_table = NULL,
  erecord_01_col,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col,
  epatient_16_col,
  eresponse_05_col,
  eresponse_24_col,
  confidence_interval = FALSE,
  method = c("wilson", "clopper-pearson"),
  conf.level = 0.95,
  correct = TRUE,
  ...
)

```

Arguments

`df` A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.

patient_scene_table	A data.frame or tibble containing at least ePatient, and eScene as a fact table.
response_table	A data.frame or tibble containing at least the eResponse fields needed for this measure's calculations.
erecord_01_col	The column representing the EMS record unique identifier.
incident_date_col	Column that contains the incident date. This defaults to NULL as it is optional in case not available due to PII restrictions.
patient_DOB_col	Column that contains the patient's date of birth. This defaults to NULL as it is optional in case not available due to PII restrictions.
epatient_15_col	Column representing the patient's numeric age agnostic of unit.
epatient_16_col	Column representing the patient's age unit ("Years", "Months", "Days", "Hours", or "Minutes").
eresponse_05_col	Column that contains eResponse.05 or the response type.
eresponse_24_col	Column detailing documentation of response mode techniques used for this EMS response.
confidence_interval	Logical. If TRUE, the function calculates a confidence interval for the proportion estimate.
method	Character. Specifies the method used to calculate confidence intervals. Options are "wilson" (Wilson score interval) and "clopper-pearson" (exact binomial interval). Partial matching is supported, so "w" and "c" can be used as shorthand.
conf.level	Numeric. The confidence level for the interval, expressed as a proportion (e.g., 0.95 for a 95% confidence interval). Defaults to 0.95.
correct	Logical. If TRUE, applies a continuity correction to the Wilson score interval when method = "wilson". Defaults to TRUE.
...	optional additional arguments to pass onto <code>dplyr::summarize</code> .

Value

A data.frame summarizing results for two population groups (All, Adults and Peds) with the following columns:

- pop: Population type (All, Adults, and Peds).
- numerator: Count of incidents meeting the measure.
- denominator: Total count of included incidents.
- prop: Proportion of incidents meeting the measure.
- prop_label: Proportion formatted as a percentage with a specified number of decimal places.

- lower_ci: Lower bound of the confidence interval for prop (if confidence_interval = TRUE).
- upper_ci: Upper bound of the confidence interval for prop (if confidence_interval = TRUE).

Author(s)

Nicolas Foss, Ed.D., MS

Examples

```
# Synthetic test data
test_data <- tibble::tibble(
  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  epatient_15 = c(34, 5, 45, 2, 60), # Ages
  epatient_16 = c("Years", "Years", "Years", "Months", "Years"),
  eresponse_05 = rep(2205001, 5),
  eresponse_24 = rep("No Lights or Sirens", 5)
)

# Run the function
# Return 95% confidence intervals using the Wilson method
safety_01(
  df = test_data,
  erecord_01_col = erecord_01,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col = epatient_15,
  epatient_16_col = epatient_16,
  eresponse_05_col = eresponse_05,
  eresponse_24_col = eresponse_24,
  confidence_interval = TRUE
)
```

safety_01_population *Safety-01 Populations*

Description

Filters data down to the target populations for Safety-01, and categorizes records to identify needed information for the calculations.

Identifies key categories related to 911 responses where "lights and sirens" were not used in an EMS dataset. This function segments the data by age into adult and pediatric populations.

Usage

```
safety_01_population(
  df = NULL,
  patient_scene_table = NULL,
  response_table = NULL,
  erecord_01_col,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col,
  epatient_16_col,
  eresponse_05_col,
  eresponse_24_col
)
```

Arguments

df A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.

patient_scene_table A data.frame or tibble containing at least ePatient, and eScene as a fact table.

response_table A data.frame or tibble containing at least the eResponse fields needed for this measure's calculations.

erecord_01_col The column representing the EMS record unique identifier.

incident_date_col Column that contains the incident date. This defaults to NULL as it is optional in case not available due to PII restrictions.

patient_DOB_col Column that contains the patient's date of birth. This defaults to NULL as it is optional in case not available due to PII restrictions.

epatient_15_col Column representing the patient's numeric age agnostic of unit.

epatient_16_col Column representing the patient's age unit ("Years", "Months", "Days", "Hours", or "Minutes").

eresponse_05_col Column that contains eResponse.05 or the response type.

eresponse_24_col Column detailing documentation of response mode techniques used for this EMS response.

Value

A list that contains the following:

- a tibble with counts for each filtering step,
- a tibble for each population of interest

- a tibble for the initial population
- a tibble for the total dataset with computations
- a tibble with a summary of missingness for each column in each table

Author(s)

Nicolas Foss, Ed.D., MS

Examples

```
# create tables to test correct functioning

# patient table
patient_table <- tibble::tibble(

  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  incident_date = as.Date(c("2025-01-01", "2025-01-05",
                           "2025-02-01", "2025-01-01",
                           "2025-06-01")
                        ),
  patient_dob = as.Date(c("2000-01-01", "2020-01-01",
                         "2023-02-01", "2023-01-01",
                         "1970-06-01")
                     ),
  epatient_15 = c(25, 5, 2, 2, 55), # Ages
  epatient_16 = c("Years", "Years", "Years", "Years", "Years")

)

# response table
response_table <- tibble::tibble(

  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  eresponse_05 = rep(2205001, 5),
  eresponse_24 = rep("No Lights or Sirens", 5)

)

# Run the function
result <- safety_01_population(patient_scene_table = patient_table,
                              response_table = response_table,
                              erecord_01_col = erecord_01,
                              incident_date_col = incident_date,
                              patient_DOB_col = patient_dob,
                              epatient_15_col = epatient_15,
                              epatient_16_col = epatient_16,
                              eresponse_05_col = eresponse_05,
                              eresponse_24_col = eresponse_24
                              )

# show the results of filtering at each step
result$filter_process
```

safety_02

*Safety-02 Calculation***Description**

The `safety_02` function calculates the Safety-02 metric, evaluating the proportion of emergency medical calls involving transport where no lights and sirens were used. This function categorizes the population into adult and pediatric groups based on their age, and summarizes results with a total population count as well.

Usage

```
safety_02(
  df = NULL,
  patient_scene_table = NULL,
  response_table = NULL,
  disposition_table = NULL,
  erecord_01_col,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col,
  epatient_16_col,
  eresponse_05_col,
  edisposition_18_col,
  edisposition_28_col,
  transport_disposition_col,
  transport_disposition_cols = lifecycle::deprecated(),
  confidence_interval = FALSE,
  method = c("wilson", "clopper-pearson"),
  conf.level = 0.95,
  correct = TRUE,
  ...
)
```

Arguments

`df` A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.

`patient_scene_table` A data.frame or tibble containing at least ePatient, and eScene as a fact table.

`response_table` A data.frame or tibble containing at least the eResponse fields needed for this measure's calculations.

`disposition_table` A data.frame or tibble containing only the edisposition fields needed for this measure's calculations.

<code>erecord_01_col</code>	The column representing the EMS record unique identifier.
<code>incident_date_col</code>	Column that contains the incident date. This defaults to NULL as it is optional in case not available due to PII restrictions.
<code>patient_DOB_col</code>	Column that contains the patient's date of birth. This defaults to NULL as it is optional in case not available due to PII restrictions.
<code>epatient_15_col</code>	Column representing the patient's numeric age agnostic of unit.
<code>epatient_16_col</code>	Column representing the patient's age unit ("Years", "Months", "Days", "Hours", or "Minutes").
<code>eresponse_05_col</code>	Column that contains eResponse.05 or the response type.
<code>edisposition_18_col</code>	Column giving documentation of transport mode techniques for this EMS response.
<code>edisposition_28_col</code>	Column giving patient disposition for an EMS event identifying whether a patient was evaluated and care or services were provided.
<code>transport_disposition_col</code>	One or more unquoted column names (such as <code>edisposition.12</code> , <code>edisposition.30</code>) containing transport disposition for an EMS event identifying whether a transport occurred and by which unit.
<code>transport_disposition_cols</code>	[Deprecated] Use <code>transport_disposition_col</code> instead.
<code>confidence_interval</code>	Logical. If TRUE, the function calculates a confidence interval for the proportion estimate.
<code>method</code>	Character. Specifies the method used to calculate confidence intervals. Options are "wilson" (Wilson score interval) and "clopper-pearson" (exact binomial interval). Partial matching is supported, so "w" and "c" can be used as shorthand.
<code>conf.level</code>	Numeric. The confidence level for the interval, expressed as a proportion (e.g., 0.95 for a 95% confidence interval). Defaults to 0.95.
<code>correct</code>	Logical. If TRUE, applies a continuity correction to the Wilson score interval when <code>method = "wilson"</code> . Defaults to TRUE.
<code>...</code>	optional additional arguments to pass onto <code>dplyr::summarize</code> .

Value

A data.frame summarizing results for two population groups (All, Adults and Peds) with the following columns:

- `pop`: Population type (All, Adults, and Peds).
- `numerator`: Count of incidents meeting the measure.

- denominator: Total count of included incidents.
- prop: Proportion of incidents meeting the measure.
- prop_label: Proportion formatted as a percentage with a specified number of decimal places.
- lower_ci: Lower bound of the confidence interval for prop (if confidence_interval = TRUE).
- upper_ci: Upper bound of the confidence interval for prop (if confidence_interval = TRUE).

Author(s)

Nicolas Foss, Ed.D., MS

Examples

```
# Synthetic test data
test_data <- tibble::tibble(
  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  epatient_15 = c(34, 5, 45, 2, 60), # Ages
  epatient_16 = c("Years", "Years", "Years", "Months", "Years"),
  eresponse_05 = rep(2205001, 5),
  edisposition_18 = rep(4218015, 5),
  edisposition_28 = rep(4228001, 5),
  edisposition_30 = rep(4230001, 5)
)

# Run the function
# Return 95% confidence intervals using the Wilson method
safety_02(
  df = test_data,
  erecord_01_col = erecord_01,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col = epatient_15,
  epatient_16_col = epatient_16,
  eresponse_05_col = eresponse_05,
  edisposition_18_col = edisposition_18,
  edisposition_28_col = edisposition_28,
  transport_disposition_col = edisposition_30,
  confidence_interval = TRUE
)
```

Description

Filters data down to the target populations for Safety-02, and categorizes records to identify needed information for the calculations.

Identifies key categories related to a 911 request during which lights and sirens were not used during patient transport. This function segments the data by age into adult and pediatric populations.

Usage

```
safety_02_population(
  df = NULL,
  patient_scene_table = NULL,
  response_table = NULL,
  disposition_table = NULL,
  erecord_01_col,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col,
  epatient_16_col,
  eresponse_05_col,
  edisposition_18_col,
  edisposition_28_col,
  transport_disposition_col,
  transport_disposition_cols = lifecycle::deprecated()
)
```

Arguments

<code>df</code>	A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.
<code>patient_scene_table</code>	A data.frame or tibble containing at least ePatient, and eScene as a fact table.
<code>response_table</code>	A data.frame or tibble containing at least the eResponse fields needed for this measure's calculations.
<code>disposition_table</code>	A data.frame or tibble containing only the edisposition fields needed for this measure's calculations.
<code>erecord_01_col</code>	The column representing the EMS record unique identifier.
<code>incident_date_col</code>	Column that contains the incident date. This defaults to NULL as it is optional in case not available due to PII restrictions.
<code>patient_DOB_col</code>	Column that contains the patient's date of birth. This defaults to NULL as it is optional in case not available due to PII restrictions.
<code>epatient_15_col</code>	Column representing the patient's numeric age agnostic of unit.

epatient_16_col	Column representing the patient's age unit ("Years", "Months", "Days", "Hours", or "Minutes").
eresponse_05_col	Column that contains eResponse.05 or the response type.
edisposition_18_col	Column giving documentation of transport mode techniques for this EMS response.
edisposition_28_col	Column giving patient disposition for an EMS event identifying whether a patient was evaluated and care or services were provided.
transport_disposition_col	One or more unquoted column names (such as edisposition.12, edisposition.30) containing transport disposition for an EMS event identifying whether a transport occurred and by which unit.
transport_disposition_cols	[Deprecated] Use transport_disposition_col instead.

Value

A list that contains the following:

- a tibble with counts for each filtering step,
- a tibble for each population of interest
- a tibble for the initial population
- a tibble for the total dataset with computations
- a tibble with a summary of missingness for each column in each table

Author(s)

Nicolas Foss, Ed.D., MS

Examples

```
# create tables to test correct functioning

# patient table
patient_table <- tibble::tibble(

  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  incident_date = as.Date(c("2025-01-01", "2025-01-05",
                           "2025-02-01", "2025-01-01",
                           "2025-06-01")
                        ),
  patient_dob = as.Date(c("2000-01-01", "2020-01-01",
                          "2023-02-01", "2023-01-01",
                          "1970-06-01")
                      ),
  epatient_15 = c(25, 5, 2, 2, 55), # Ages
```

```

    epatient_16 = c("Years", "Years", "Years", "Years", "Years")
  )

  # response table
  response_table <- tibble::tibble(

    erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
    eresponse_05 = rep(2205001, 5)

  )

  # disposition table
  disposition_table <- tibble::tibble(
    erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
    edisposition_18 = rep(4218015, 5),
    edisposition_28 = rep(4228001, 5),
    edisposition_30 = rep(4230001, 5)
  )

  # test the success of the function
  result <- safety_02_population(patient_scene_table = patient_table,
                                response_table = response_table,
                                disposition_table = disposition_table,
                                erecord_01_col = erecord_01,
                                incident_date_col = incident_date,
                                patient_DOB_col = patient_dob,
                                epatient_15_col = epatient_15,
                                epatient_16_col = epatient_16,
                                eresponse_05_col = eresponse_05,
                                edisposition_18_col = edisposition_18,
                                edisposition_28_col = edisposition_28,
                                transport_disposition_col = edisposition_30
  )

  # show the results of filtering at each step
  result$filter_process

```

safety_04

Safety-04 Calculation

Description

The `safety_04` function processes EMS incident data for specific safety and transport criteria, filtering by patient age and incident type to identify cases that meet specified exclusion or inclusion criteria. This function accommodates data with various EMS-specific codes, age descriptors, and procedure identifiers.

Usage

```
safety_04(
  df = NULL,
  patient_scene_table = NULL,
  response_table = NULL,
  arrest_table = NULL,
  injury_table = NULL,
  procedures_table = NULL,
  disposition_table = NULL,
  erecord_01_col,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col,
  epatient_16_col,
  eresponse_05_col,
  earrest_01_col,
  einjury_03_col,
  eprocedures_03_col,
  edisposition_14_col,
  transport_disposition_col,
  confidence_interval = FALSE,
  method = c("wilson", "clopper-pearson"),
  conf.level = 0.95,
  correct = TRUE,
  ...
)
```

Arguments

<code>df</code>	A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.
<code>patient_scene_table</code>	A data.frame or tibble containing at least ePatient, and eScene as a fact table.
<code>response_table</code>	A data.frame or tibble containing at least the eResponse fields needed for this measure's calculations.
<code>arrest_table</code>	A data.frame or tibble containing at least the eArrest fields needed for this measure's calculations.
<code>injury_table</code>	A data frame or tibble containing fields from eInjury needed for this measure's calculations.
<code>procedures_table</code>	A dataframe or tibble containing at least the eProcedures fields needed.
<code>disposition_table</code>	A data.frame or tibble containing only the edisposition fields needed for this measure's calculations.
<code>erecord_01_col</code>	The column representing the EMS record unique identifier.

<code>incident_date_col</code>	Column that contains the incident date. This defaults to NULL as it is optional in case not available due to PII restrictions.
<code>patient_DOB_col</code>	Column that contains the patient's date of birth. This defaults to NULL as it is optional in case not available due to PII restrictions.
<code>epatient_15_col</code>	Column representing the patient's numeric age agnostic of unit.
<code>epatient_16_col</code>	Column representing the patient's age unit ("Years", "Months", "Days", "Hours", or "Minutes").
<code>eresponse_05_col</code>	Column that contains eResponse.05 or the response type.
<code>earrest_01_col</code>	Column representing whether or not the patient is in arrest.
<code>einjury_03_col</code>	Column describing Trauma triage criteria for the red boxes (Injury Patterns and Mental Status and Vital Signs) in the 2021 ACS National Guideline for the Field Triage of Injured Patients.
<code>eprocedures_03_col</code>	Column containing procedure codes with or without procedure names.
<code>edisposition_14_col</code>	Column giving the position of the patient during transport from the scene.
<code>transport_disposition_col</code>	One or more unquoted column names (such as <code>edisposition.12</code> , <code>edisposition.30</code>) containing transport disposition for an EMS event identifying whether a transport occurred and by which unit.
<code>confidence_interval</code>	Logical. If TRUE, the function calculates a confidence interval for the proportion estimate.
<code>method</code>	Character. Specifies the method used to calculate confidence intervals. Options are "wilson" (Wilson score interval) and "clopper-pearson" (exact binomial interval). Partial matching is supported, so "w" and "c" can be used as shorthand.
<code>conf.level</code>	Numeric. The confidence level for the interval, expressed as a proportion (e.g., 0.95 for a 95% confidence interval). Defaults to 0.95.
<code>correct</code>	Logical. If TRUE, applies a continuity correction to the Wilson score interval when <code>method = "wilson"</code> . Defaults to TRUE.
<code>...</code>	optional additional arguments to pass onto <code>dplyr::summarize</code> .

Value

A `data.frame` summarizing results for two population groups (All, Adults and Peds) with the following columns:

- `pop`: Population type (All, Adults, and Peds).
- `numerator`: Count of incidents meeting the measure.
- `denominator`: Total count of included incidents.

- prop: Proportion of incidents meeting the measure.
- prop_label: Proportion formatted as a percentage with a specified number of decimal places.
- lower_ci: Lower bound of the confidence interval for prop (if confidence_interval = TRUE).
- upper_ci: Upper bound of the confidence interval for prop (if confidence_interval = TRUE).

Author(s)

Nicolas Foss, Ed.D., MS

Examples

```
# Synthetic test data
test_data <- tibble::tibble(
  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  epatient_15 = c(34, 5, 45, 2, 60), # Ages
  epatient_16 = c("Years", "Years", "Years", "Months", "Years"),
  eresponse_05 = rep(2205001, 5),
  earrest_01 = rep("No", 5),
  einjury_03 = rep("non-injury", 5),
  edisposition_14 = rep(4214001, 5),
  edisposition_30 = rep(4230001, 5),
  eprocedures_03 = rep("other response", 5)
)

# Run the function
# Return 95% confidence intervals using the Wilson method
safety_04(
  df = test_data,
  erecord_01_col = erecord_01,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col = epatient_15,
  epatient_16_col = epatient_16,
  eresponse_05_col = eresponse_05,
  earrest_01_col = earrest_01,
  einjury_03_col = einjury_03,
  edisposition_14_col = edisposition_14,
  transport_disposition_col = edisposition_30,
  eprocedures_03_col = eprocedures_03,
  confidence_interval = TRUE
)
```

Description

Filters data down to the target populations for Safety-04, and categorizes records to identify needed information for the calculations.

Identifies key categories related to a 911 request or interfacility request for patients less than 8 years of age during which patients are transported using a pediatric restraint device. This function segments the data by age into adult and pediatric populations.

Usage

```
safety_04_population(
  df = NULL,
  patient_scene_table = NULL,
  response_table = NULL,
  arrest_table = NULL,
  injury_table = NULL,
  procedures_table = NULL,
  disposition_table = NULL,
  erecord_01_col,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col,
  epatient_16_col,
  eresponse_05_col,
  earrest_01_col,
  einjury_03_col,
  eprocedures_03_col,
  edisposition_14_col,
  transport_disposition_col
)
```

Arguments

<code>df</code>	A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.
<code>patient_scene_table</code>	A data.frame or tibble containing at least ePatient, and eScene as a fact table.
<code>response_table</code>	A data.frame or tibble containing at least the eResponse fields needed for this measure's calculations.
<code>arrest_table</code>	A data.frame or tibble containing at least the eArrest fields needed for this measure's calculations.
<code>injury_table</code>	A data frame or tibble containing fields from eInjury needed for this measure's calculations.
<code>procedures_table</code>	A dataframe or tibble containing at least the eProcedures fields needed.
<code>disposition_table</code>	A data.frame or tibble containing only the edisposition fields needed for this measure's calculations.

<code>erecord_01_col</code>	The column representing the EMS record unique identifier.
<code>incident_date_col</code>	Column that contains the incident date. This defaults to NULL as it is optional in case not available due to PII restrictions.
<code>patient_DOB_col</code>	Column that contains the patient's date of birth. This defaults to NULL as it is optional in case not available due to PII restrictions.
<code>epatient_15_col</code>	Column representing the patient's numeric age agnostic of unit.
<code>epatient_16_col</code>	Column representing the patient's age unit ("Years", "Months", "Days", "Hours", or "Minutes").
<code>eresponse_05_col</code>	Column that contains eResponse.05 or the response type.
<code>earrest_01_col</code>	Column representing whether or not the patient is in arrest.
<code>einjury_03_col</code>	Column describing Trauma triage criteria for the red boxes (Injury Patterns and Mental Status and Vital Signs) in the 2021 ACS National Guideline for the Field Triage of Injured Patients.
<code>eprocedures_03_col</code>	Column containing procedure codes with or without procedure names.
<code>edisposition_14_col</code>	Column giving the position of the patient during transport from the scene.
<code>transport_disposition_col</code>	One or more unquoted column names (such as <code>edisposition.12</code> , <code>edisposition.30</code>) containing transport disposition for an EMS event identifying whether a transport occurred and by which unit.

Value

A list that contains the following:

- a tibble with counts for each filtering step,
- a tibble for each population of interest
- a tibble for the initial population
- a tibble for the total dataset with computations
- a tibble with a summary of missingness for each column in each table

Author(s)

Nicolas Foss, Ed.D., MS

Examples

```
# create tables to test correct functioning

# patient table
patient_table <- tibble::tibble(
```



```

procedures_table = procedures_table,
disposition_table = disposition_table,
erecord_01_col = erecord_01,
incident_date_col = incident_date,
patient_DOB_col = patient_dob,
epatient_15_col = epatient_15,
epatient_16_col = epatient_16,
eresponse_05_col = eresponse_05,
earrest_01_col = earrest_01,
einjury_03_col = einjury_03,
edisposition_14_col = edisposition_14,
transport_disposition_col = edisposition_30,
eprocedures_03_col = eprocedures_03
)

# show the results of filtering at each step
result$filter_process

```

seizure_02

Seizure-02 Calculation

Description

Calculates the NEMSQA Seizure-02 Measure.

Calculates age-based seizure metrics for a dataset. This function filters data for patients based on incident information, diagnoses, and administered medications to assess adherence to Seizure-02 metrics.

Usage

```

seizure_02(
  df = NULL,
  patient_scene_table = NULL,
  response_table = NULL,
  situation_table = NULL,
  medications_table = NULL,
  erecord_01_col,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col,
  epatient_16_col,
  eresponse_05_col,
  esituation_11_col,
  esituation_12_col,
  emedications_03_col,
  confidence_interval = FALSE,
  method = c("wilson", "clopper-pearson"),

```

```

    conf.level = 0.95,
    correct = TRUE,
    ...
)

```

Arguments

df A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.

patient_scene_table A data.frame or tibble containing at least ePatient, and eScene as a fact table.

response_table A data.frame or tibble containing at least the eResponse fields needed for this measure's calculations.

situation_table A data.frame or tibble containing at least the eSituation fields needed for this measure's calculations. Default is NULL.

medications_table A data.frame or tibble containing at least the eMedications fields needed for this measure's calculations. Default is NULL.

erecord_01_col The column representing the EMS record unique identifier.

incident_date_col Column that contains the incident date. This defaults to NULL as it is optional in case not available due to PII restrictions.

patient_DOB_col Column that contains the patient's date of birth. This defaults to NULL as it is optional in case not available due to PII restrictions.

epatient_15_col Column representing the patient's numeric age agnostic of unit.

epatient_16_col Column representing the patient's age unit ("Years", "Months", "Days", "Hours", or "Minutes").

eresponse_05_col Column that contains eResponse.05 or the response type.

esituation_11_col Column that contains eSituation.11 provider primary impression data.

esituation_12_col Column that contains all eSituation.12 values as (possible a single comma-separated list), provider secondary impression data.

emedications_03_col Column that contains all medication administered to the patient (eMedications.03) values as a single comma-separated list per distinct eRecord.01 ID.

confidence_interval Logical. If TRUE, the function calculates a confidence interval for the proportion estimate.

method	Character. Specifies the method used to calculate confidence intervals. Options are "wilson" (Wilson score interval) and "clopper-pearson" (exact binomial interval). Partial matching is supported, so "w" and "c" can be used as shorthand.
conf.level	Numeric. The confidence level for the interval, expressed as a proportion (e.g., 0.95 for a 95% confidence interval). Defaults to 0.95.
correct	Logical. If TRUE, applies a continuity correction to the Wilson score interval when method = "wilson". Defaults to TRUE.
...	optional additional arguments to pass onto <code>dplyr::summarize</code> .

Value

A data.frame summarizing results for two population groups (All, Adults and Peds) with the following columns:

- pop: Population type (All, Adults, and Peds).
- numerator: Count of incidents meeting the measure.
- denominator: Total count of included incidents.
- prop: Proportion of incidents meeting the measure.
- prop_label: Proportion formatted as a percentage with a specified number of decimal places.
- lower_ci: Lower bound of the confidence interval for prop (if confidence_interval = TRUE).
- upper_ci: Upper bound of the confidence interval for prop (if confidence_interval = TRUE).

Author(s)

Nicolas Foss, Ed.D., MS

Examples

```
# Synthetic test data
test_data <- tibble::tibble(
  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  epatient_15 = c(34, 5, 45, 2, 60), # Ages
  epatient_16 = c("Years", "Years", "Years", "Months", "Years"),
  eresponse_05 = rep(2205001, 5),
  esituation_11 = rep("G40", 5),
  esituation_12 = rep("r56", 5),
  emedications_03 = rep(3322, 5)
)

# Run the function
# Return 95% confidence intervals using the Wilson method
seizure_02(
  df = test_data,
  erecord_01_col = erecord_01,
  incident_date_col = NULL,
```

```

    patient_DOB_col = NULL,
    epatient_15_col = epatient_15,
    epatient_16_col = epatient_16,
    eresponse_05_col = eresponse_05,
    esituation_11_col = esituation_11,
    esituation_12_col = esituation_12,
    emedications_03_col = emedications_03,
    confidence_interval = TRUE
  )

```

seizure_02_population *Seizure-02 Populations*

Description

Filters data down to the target populations for Seizure-02, and categorizes records to identify needed information for the calculations.

Identifies key categories related to asthma-related incidents in an EMS dataset, specifically focusing on cases where 911 was called for respiratory distress, and certain medications were administered. This function segments the data by age into adult and pediatric populations.

Usage

```

seizure_02_population(
  df = NULL,
  patient_scene_table = NULL,
  response_table = NULL,
  situation_table = NULL,
  medications_table = NULL,
  erecord_01_col,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col,
  epatient_16_col,
  eresponse_05_col,
  esituation_11_col,
  esituation_12_col,
  emedications_03_col
)

```

Arguments

df A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.

patient_scene_table A data.frame or tibble containing at least ePatient, and eScene as a fact table.

response_table	A data.frame or tibble containing at least the eResponse fields needed for this measure's calculations.
situation_table	A data.frame or tibble containing at least the eSituation fields needed for this measure's calculations. Default is NULL.
medications_table	A data.frame or tibble containing at least the eMedications fields needed for this measure's calculations. Default is NULL.
erecord_01_col	The column representing the EMS record unique identifier.
incident_date_col	Column that contains the incident date. This defaults to NULL as it is optional in case not available due to PII restrictions.
patient_DOB_col	Column that contains the patient's date of birth. This defaults to NULL as it is optional in case not available due to PII restrictions.
epatient_15_col	Column representing the patient's numeric age agnostic of unit.
epatient_16_col	Column representing the patient's age unit ("Years", "Months", "Days", "Hours", or "Minutes").
eresponse_05_col	Column that contains eResponse.05 or the response type.
esituation_11_col	Column that contains eSituation.11 provider primary impression data.
esituation_12_col	Column that contains all eSituation.12 values as (possible a single comma-separated list), provider secondary impression data.
emedications_03_col	Column that contains all medication administered to the patient (eMedications.03) values as a single comma-separated list per distinct eRecord.01 ID.

Value

A list that contains the following:

- a tibble with counts for each filtering step,
- a tibble for each population of interest
- a tibble for the initial population
- a tibble for the total dataset with computations
- a tibble with a summary of missingness for each column in each table

Author(s)

Nicolas Foss, Ed.D., MS


```

    epatient_15_col = epatient_15,
    epatient_16_col = epatient_16,
    eresponse_05_col = eresponse_05,
    esituation_11_col = esituation_11,
    esituation_12_col = esituation_12,
    emedications_03_col = emedications_03
  )

# show the results of filtering at each step
result$filter_process

```

stroke_01

Stroke-01 Calculation

Description

The `stroke_01` function processes EMS dataset to identify potential stroke cases based on specific criteria and calculates the stroke scale measures. It filters the data for 911 response calls, identifies stroke-related impressions and scales, and aggregates results by unique patient encounters.

Usage

```

stroke_01(
  df = NULL,
  patient_scene_table = NULL,
  response_table = NULL,
  situation_table = NULL,
  vitals_table = NULL,
  erecord_01_col,
  eresponse_05_col,
  esituation_11_col,
  esituation_12_col,
  evitals_23_col,
  evitals_26_col,
  evitals_29_col,
  evitals_30_col,
  confidence_interval = FALSE,
  method = c("wilson", "clopper-pearson"),
  conf.level = 0.95,
  correct = TRUE,
  ...
)

```

Arguments

`df` A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.

patient_scene_table	A data.frame or tibble containing at least ePatient, and eScene as a fact table.
response_table	A data.frame or tibble containing at least the eResponse fields needed for this measure's calculations.
situation_table	A data.frame or tibble containing at least the eSituation fields needed for this measure's calculations. Default is NULL.
vitals_table	A dataframe or tibble containing at least the eVitals fields needed.
erecord_01_col	The column representing the EMS record unique identifier.
eresponse_05_col	Column that contains eResponse.05 or the response type.
esituation_11_col	Column that contains eSituation.11 provider primary impression data.
esituation_12_col	Column that contains all eSituation.12 values as (possible a single comma-separated list), provider secondary impression data.
evitals_23_col	Column for Glasgow Coma Scale (GCS) scores.
evitals_26_col	Column for AVPU alertness levels.
evitals_29_col	The column containing the stroke scale score achieved during assessment.
evitals_30_col	The column containing stroke scale type descriptors (e.g., FAST, NIH, etc.).
confidence_interval	Logical. If TRUE, the function calculates a confidence interval for the proportion estimate.
method	Character. Specifies the method used to calculate confidence intervals. Options are "wilson" (Wilson score interval) and "clopper-pearson" (exact binomial interval). Partial matching is supported, so "w" and "c" can be used as shorthand.
conf.level	Numeric. The confidence level for the interval, expressed as a proportion (e.g., 0.95 for a 95% confidence interval). Defaults to 0.95.
correct	Logical. If TRUE, applies a continuity correction to the Wilson score interval when method = "wilson". Defaults to TRUE.
...	optional additional arguments to pass onto <code>dplyr::summarize</code> .

Value

A data.frame summarizing results for two population groups (All, Adults and Peds) with the following columns:

- pop: Population type (All, Adults, and Peds).
- numerator: Count of incidents meeting the measure.
- denominator: Total count of included incidents.
- prop: Proportion of incidents meeting the measure.
- prop_label: Proportion formatted as a percentage with a specified number of decimal places.

- lower_ci: Lower bound of the confidence interval for prop (if confidence_interval = TRUE).
- upper_ci: Upper bound of the confidence interval for prop (if confidence_interval = TRUE).

Author(s)

Nicolas Foss, Ed.D., MS

Examples

```
# Synthetic test data
test_data <- tibble::tibble(
  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  epatient_15 = c(34, 5, 45, 2, 60), # Ages
  epatient_16 = c("Years", "Years", "Years", "Months", "Years"),
  eresponse_05 = rep(2205001, 5),
  esituation_11 = c(rep("I60", 3), rep("I61", 2)),
  esituation_12 = c(rep("I63", 2), rep("I64", 3)),
  evitals_23 = c(16, 15, 14, 13, 12),
  evitals_26 = c("Alert", "Painful", "Verbal", "Unresponsive", "Alert"),
  evitals_29 = rep("positive", 5),
  evitals_30 = rep("a pain scale", 5)
)

# Run the function
# Return 95% confidence intervals using the Wilson method
stroke_01(
  df = test_data,
  erecord_01_col = erecord_01,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  eresponse_05_col = eresponse_05,
  esituation_11_col = esituation_11,
  esituation_12_col = esituation_12,
  evitals_23_col = evitals_23,
  evitals_26_col = evitals_26,
  evitals_29_col = evitals_29,
  evitals_30_col = evitals_30,
  confidence_interval = TRUE
)
```

Description

Filters data down to the target populations for Stroke-01, and categorizes records to identify needed information for the calculations.

Identifies key categories related to stroke-related incidents in an EMS dataset, specifically focusing on cases where 911 was called for stroke, and a stroke scale was administered. .

Usage

```
stroke_01_population(
  df = NULL,
  patient_scene_table = NULL,
  response_table = NULL,
  situation_table = NULL,
  vitals_table = NULL,
  erecord_01_col,
  eresponse_05_col,
  esituation_11_col,
  esituation_12_col,
  evitals_23_col,
  evitals_26_col,
  evitals_29_col,
  evitals_30_col
)
```

Arguments

<code>df</code>	A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.
<code>patient_scene_table</code>	A data.frame or tibble containing at least ePatient, and eScene as a fact table.
<code>response_table</code>	A data.frame or tibble containing at least the eResponse fields needed for this measure's calculations.
<code>situation_table</code>	A data.frame or tibble containing at least the eSituation fields needed for this measure's calculations. Default is NULL.
<code>vitals_table</code>	A dataframe or tibble containing at least the eVitals fields needed.
<code>erecord_01_col</code>	The column representing the EMS record unique identifier.
<code>eresponse_05_col</code>	Column that contains eResponse.05 or the response type.
<code>esituation_11_col</code>	Column that contains eSituation.11 provider primary impression data.
<code>esituation_12_col</code>	Column that contains all eSituation.12 values as (possible a single comma-separated list), provider secondary impression data.
<code>evitals_23_col</code>	Column for Glasgow Coma Scale (GCS) scores.
<code>evitals_26_col</code>	Column for AVPU alertness levels.

evitals_29_col The column containing the stroke scale score achieved during assessment.
 evitals_30_col The column containing stroke scale type descriptors (e.g., FAST, NIH, etc.).

Value

A list that contains the following:

- a tibble with counts for each filtering step,
- a tibble for each population of interest
- a tibble for the initial population
- a tibble for the total dataset with computations
- a tibble with a summary of missingness for each column in each table

Author(s)

Nicolas Foss, Ed.D., MS

Examples

```
# create tables to test correct functioning

# patient table
patient_table <- tibble::tibble(

  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  incident_date = as.Date(c("2025-01-01", "2025-01-05",
                           "2025-02-01", "2025-01-01",
                           "2025-06-01")
                          ),
  patient_dob = as.Date(c("2000-01-01", "2020-01-01",
                         "2023-02-01", "2023-01-01",
                         "1970-06-01")
                       ),
  epatient_15 = c(25, 5, 2, 2, 55), # Ages
  epatient_16 = c("Years", "Years", "Years", "Years", "Years")

)

# response table
response_table <- tibble::tibble(

  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  eresponse_05 = rep(2205001, 5)

)

# situation table
situation_table <- tibble::tibble(

  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  esituation_11 = c(rep("I60", 3), rep("I61", 2)),
```

```

    esituation_12 = c(rep("I63", 2), rep("I64", 3)),
  )

# vitals table
vitals_table <- tibble::tibble(

  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  evitals_23 = c(16, 15, 14, 13, 12),
  evitals_26 = c("Alert", "Painful", "Verbal", "Unresponsive", "Alert"),
  evitals_29 = rep("positive", 5),
  evitals_30 = rep("a pain scale", 5)
)

# test the success of the function
result <- stroke_01_population(patient_scene_table = patient_table,
                              response_table = response_table,
                              situation_table = situation_table,
                              vitals_table = vitals_table,
                              erecord_01_col = erecord_01,
                              eresponse_05_col = eresponse_05,
                              esituation_11_col = esituation_11,
                              esituation_12_col = esituation_12,
                              evitals_29_col = evitals_29,
                              evitals_23_col = evitals_23,
                              evitals_26_col = evitals_26,
                              evitals_30_col = evitals_30
                              )

# show the results of filtering at each step
result$filter_process

```

 syncope_01

Syncope-01 Calculation

Description

The `syncope_01` function processes EMS dataset to identify potential syncope (fainting) cases based on specific criteria and calculates related ECG measures. This function `dplyr::filters` data for 911 response calls, assesses primary and associated symptoms for syncope, determines age-based populations (adult and pediatric), and aggregates results by unique patient encounters.

Usage

```

syncope_01(
  df = NULL,
  patient_scene_table = NULL,
  response_table = NULL,
  situation_table = NULL,

```

```

vitals_table = NULL,
erecord_01_col,
incident_date_col = NULL,
patient_DOB_col = NULL,
epatient_15_col,
epatient_16_col,
eresponse_05_col,
esituation_09_col,
esituation_10_col,
esituation_11_col,
esituation_12_col,
evitals_04_col,
confidence_interval = FALSE,
method = c("wilson", "clopper-pearson"),
conf.level = 0.95,
correct = TRUE,
...
)

```

Arguments

df A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.

patient_scene_table A data.frame or tibble containing at least ePatient, and eScene as a fact table.

response_table A data.frame or tibble containing at least the eResponse fields needed for this measure's calculations.

situation_table A data.frame or tibble containing at least the eSituation fields needed for this measure's calculations. Default is NULL.

vitals_table A dataframe or tibble containing at least the eVitals fields needed.

erecord_01_col The column representing the EMS record unique identifier.

incident_date_col Column that contains the incident date. This defaults to NULL as it is optional in case not available due to PII restrictions.

patient_DOB_col Column that contains the patient's date of birth. This defaults to NULL as it is optional in case not available due to PII restrictions.

epatient_15_col Column representing the patient's numeric age agnostic of unit.

epatient_16_col Column representing the patient's age unit ("Years", "Months", "Days", "Hours", or "Minutes").

eresponse_05_col Column that contains eResponse.05 or the response type.

esituation_09_col Column with primary sign and symptom present in the patient or observed by EMS personnel.

<code>esituation_10_col</code>	Column with other symptoms identified by the patient or observed by EMS personnel.
<code>esituation_11_col</code>	Column that contains eSituation.11 provider primary impression data.
<code>esituation_12_col</code>	Column that contains all eSituation.12 values as (possible a single comma-separated list), provider secondary impression data.
<code>evitals_04_col</code>	Column with type of ECG associated with the cardiac rhythm if available.
<code>confidence_interval</code>	Logical. If TRUE, the function calculates a confidence interval for the proportion estimate.
<code>method</code>	Character. Specifies the method used to calculate confidence intervals. Options are "wilson" (Wilson score interval) and "clopper-pearson" (exact binomial interval). Partial matching is supported, so "w" and "c" can be used as shorthand.
<code>conf.level</code>	Numeric. The confidence level for the interval, expressed as a proportion (e.g., 0.95 for a 95% confidence interval). Defaults to 0.95.
<code>correct</code>	Logical. If TRUE, applies a continuity correction to the Wilson score interval when <code>method = "wilson"</code> . Defaults to TRUE.
<code>...</code>	optional additional arguments to pass onto <code>dplyr::summarize</code> .

Value

A `data.frame` summarizing results for two population groups (All, Adults and Peds) with the following columns:

- `pop`: Population type (All, Adults, and Peds).
- `numerator`: Count of incidents meeting the measure.
- `denominator`: Total count of included incidents.
- `prop`: Proportion of incidents meeting the measure.
- `prop_label`: Proportion formatted as a percentage with a specified number of decimal places.
- `lower_ci`: Lower bound of the confidence interval for `prop` (if `confidence_interval = TRUE`).
- `upper_ci`: Upper bound of the confidence interval for `prop` (if `confidence_interval = TRUE`).

Author(s)

Nicolas Foss, Ed.D., MS

Examples

```

# Synthetic test data
test_data <- tibble::tibble(
  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  epatient_15 = c(34, 5, 45, 2, 60), # Ages
  epatient_16 = c("Years", "Years", "Years", "Months", "Years"),
  eresponse_05 = rep(2205001, 5),
  esituation_09 = c(rep("R55", 3), rep("R40.4", 2)),
  esituation_10 = c(rep("R40.4", 2), rep("R55", 3)),
  esituation_11 = c(rep("R55", 3), rep("R40.4", 2)),
  esituation_12 = c(rep("R40.4", 2), rep("R55", 3)),
  evitals_04 = rep("15 Lead", 5)
)

# Run the function
# Return 95% confidence intervals using the Wilson method
syncope_01(
  df = test_data,
  erecord_01_col = erecord_01,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col = epatient_15,
  epatient_16_col = epatient_16,
  eresponse_05_col = eresponse_05,
  esituation_09_col = esituation_09,
  esituation_10_col = esituation_10,
  esituation_11_col = esituation_11,
  esituation_12_col = esituation_12,
  evitals_04_col = evitals_04,
  confidence_interval = TRUE
)

```

syncope_01_population *Syncope-01 Populations*

Description

Filters data down to the target populations for Syncope-01, and categorizes records to identify needed information for the calculations.

Identifies key categories to identify potential syncope (fainting) cases based on specific criteria and calculates related ECG measures. This function segments the data by age into adult and pediatric populations.

Usage

```

syncope_01_population(
  df = NULL,
  patient_scene_table = NULL,

```

```

response_table = NULL,
situation_table = NULL,
vitals_table = NULL,
erecord_01_col,
incident_date_col = NULL,
patient_DOB_col = NULL,
epatient_15_col,
epatient_16_col,
eresponse_05_col,
esituation_09_col,
esituation_10_col,
esituation_11_col,
esituation_12_col,
evitals_04_col
)

```

Arguments

df A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.

patient_scene_table A data.frame or tibble containing at least ePatient, and eScene as a fact table.

response_table A data.frame or tibble containing at least the eResponse fields needed for this measure's calculations.

situation_table A data.frame or tibble containing at least the eSituation fields needed for this measure's calculations. Default is NULL.

vitals_table A dataframe or tibble containing at least the eVitals fields needed.

erecord_01_col The column representing the EMS record unique identifier.

incident_date_col Column that contains the incident date. This defaults to NULL as it is optional in case not available due to PII restrictions.

patient_DOB_col Column that contains the patient's date of birth. This defaults to NULL as it is optional in case not available due to PII restrictions.

epatient_15_col Column representing the patient's numeric age agnostic of unit.

epatient_16_col Column representing the patient's age unit ("Years", "Months", "Days", "Hours", or "Minutes").

eresponse_05_col Column that contains eResponse.05 or the response type.

esituation_09_col Column with primary sign and symptom present in the patient or observed by EMS personnel.

esituation_10_col	Column with other symptoms identified by the patient or observed by EMS personnel.
esituation_11_col	Column that contains eSituation.11 provider primary impression data.
esituation_12_col	Column that contains all eSituation.12 values as (possible a single comma-separated list), provider secondary impression data.
evitals_04_col	Column with type of ECG associated with the cardiac rhythm if available.

Value

A list that contains the following:

- a tibble with counts for each filtering step,
- a tibble for each population of interest
- a tibble for the initial population
- a tibble for the total dataset with computations
- a tibble with a summary of missingness for each column in each table

Author(s)

Nicolas Foss, Ed.D., MS

Examples

```
# create tables to test correct functioning

# patient table
patient_table <- tibble::tibble(

  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  incident_date = as.Date(c("2025-01-01", "2025-01-05",
                           "2025-02-01", "2025-01-01",
                           "2025-06-01")
                        ),
  patient_dob = as.Date(c("2000-01-01", "2020-01-01",
                         "2023-02-01", "2023-01-01",
                         "1970-06-01")
                     ),
  epatient_15 = c(25, 5, 2, 2, 55), # Ages
  epatient_16 = c("Years", "Years", "Years", "Years", "Years")

)

# response table
response_table <- tibble::tibble(

  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  eresponse_05 = rep(2205001, 5)
```

```

)

# situation table
situation_table <- tibble::tibble(

  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  esituation_09 = c(rep("R55", 3), rep("R40.4", 2)),
  esituation_10 = c(rep("R40.4", 2), rep("R55", 3)),
  esituation_11 = c(rep("R55", 3), rep("R40.4", 2)),
  esituation_12 = c(rep("R40.4", 2), rep("R55", 3)),
)

# vitals table
vitals_table <- tibble::tibble(

  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  evitals_04 = rep("15 Lead", 5)

)

# test the success of the function
result <- syncope_01_population(patient_scene_table = patient_table,
                               response_table = response_table,
                               situation_table = situation_table,
                               vitals_table = vitals_table,
                               erecord_01_col = erecord_01,
                               incident_date_col = NULL,
                               patient_DOB_col = NULL,
                               epatient_15_col = epatient_15,
                               epatient_16_col = epatient_16,
                               eresponse_05_col = eresponse_05,
                               esituation_09_col = esituation_09,
                               esituation_10_col = esituation_10,
                               esituation_11_col = esituation_11,
                               esituation_12_col = esituation_12,
                               evitals_04_col = evitals_04
                               )

# show the results of filtering at each step
result$filter_process

```

tbi_01

TBI-01 Calculation

Description

This function screens for potential traumatic brain injury (TBI) cases based on specific criteria in a patient dataset. It produces a subset of the data with calculated variables for TBI identification.

Usage

```
tbi_01(
  df = NULL,
  patient_scene_table = NULL,
  response_table = NULL,
  situation_table = NULL,
  disposition_table = NULL,
  vitals_table = NULL,
  erecord_01_col,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col,
  epatient_16_col,
  eresponse_05_col,
  esituation_11_col,
  esituation_12_col,
  transport_disposition_col,
  evitals_06_col,
  evitals_12_col,
  evitals_16_col,
  evitals_23_col,
  evitals_26_col,
  confidence_interval = FALSE,
  method = c("wilson", "clopper-pearson"),
  conf.level = 0.95,
  correct = TRUE,
  ...
)
```

Arguments

df A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.

patient_scene_table A data.frame or tibble containing at least ePatient, and eScene as a fact table.

response_table A data.frame or tibble containing at least the eResponse fields needed for this measure's calculations.

situation_table A data.frame or tibble containing at least the eSituation fields needed for this measure's calculations. Default is NULL.

disposition_table A data.frame or tibble containing only the edisposition fields needed for this measure's calculations.

vitals_table A dataframe or tibble containing at least the eVitals fields needed.

erecord_01_col The column representing the EMS record unique identifier.

incident_date_col Column that contains the incident date. This defaults to NULL as it is optional in case not available due to PII restrictions.

patient_DOB_col	Column that contains the patient's date of birth. This defaults to NULL as it is optional in case not available due to PII restrictions.
epatient_15_col	Column representing the patient's numeric age agnostic of unit.
epatient_16_col	Column representing the patient's age unit ("Years", "Months", "Days", "Hours", or "Minutes").
eresponse_05_col	Column that contains eResponse.05 or the response type.
esituation_11_col	Column that contains eSituation.11 provider primary impression data.
esituation_12_col	Column that contains all eSituation.12 values as (possible a single comma-separated list), provider secondary impression data.
transport_disposition_col	One or more unquoted column names (such as edisposition.12, edisposition.30) containing transport disposition for an EMS event identifying whether a transport occurred and by which unit.
evitals_06_col	Numeric column containing systolic blood pressure values.
evitals_12_col	Numeric column containing pulse oximetry values.
evitals_16_col	Column with numeric value of the patient's exhaled end tidal carbon dioxide (ETCO2) level measured as a unit of pressure in millimeters of mercury (mmHg), percentage or, kilopascal (kPa).
evitals_23_col	Column for Glasgow Coma Scale (GCS) scores.
evitals_26_col	Column for AVPU alertness levels.
confidence_interval	Logical. If TRUE, the function calculates a confidence interval for the proportion estimate.
method	Character. Specifies the method used to calculate confidence intervals. Options are "wilson" (Wilson score interval) and "clopper-pearson" (exact binomial interval). Partial matching is supported, so "w" and "c" can be used as shorthand.
conf.level	Numeric. The confidence level for the interval, expressed as a proportion (e.g., 0.95 for a 95% confidence interval). Defaults to 0.95.
correct	Logical. If TRUE, applies a continuity correction to the Wilson score interval when method = "wilson". Defaults to TRUE.
...	optional additional arguments to pass onto <code>dplyr::summarize</code> .

Value

A `data.frame` summarizing results for two population groups (All, Adults and Peds) with the following columns:

- pop: Population type (All, Adults, and Peds).

- numerator: Count of incidents meeting the measure.
- denominator: Total count of included incidents.
- prop: Proportion of incidents meeting the measure.
- prop_label: Proportion formatted as a percentage with a specified number of decimal places.
- lower_ci: Lower bound of the confidence interval for prop (if confidence_interval = TRUE).
- upper_ci: Upper bound of the confidence interval for prop (if confidence_interval = TRUE).

Author(s)

Nicolas Foss, Ed.D., MS

Examples

```
# Synthetic test data
test_data <- tibble::tibble(
  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  epatient_15 = c(34, 5, 45, 2, 60), # Ages
  epatient_16 = c("Years", "Years", "Years", "Months", "Years"),
  eresponse_05 = rep(2205001, 5),
  esituation_11 = c(rep("S02", 3), rep("S06", 2)),
  esituation_12 = c(rep("S09.90", 2), rep("S06.0X9", 3)),
  evitals_06 = c(85, 80, 100, 90, 82),
  evitals_12 = c(95, 96, 97, 98, 99),
  evitals_16 = c(35, 36, 37, 38, 39),
  evitals_23 = rep(8, 5),
  evitals_26 = c("Verbal", "Painful", "Unresponsive", "Verbal", "Painful"),
  edisposition_30 = c(4230001, 4230003, 4230001, 4230007, 4230007)
)

# Run the function
# Return 95% confidence intervals using the Wilson method
tbi_01(
  df = test_data,
  erecord_01_col = erecord_01,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col = epatient_15,
  epatient_16_col = epatient_16,
  eresponse_05_col = eresponse_05,
  esituation_11_col = esituation_11,
  esituation_12_col = esituation_12,
  evitals_06_col = evitals_06,
  evitals_12_col = evitals_12,
  evitals_16_col = evitals_16,
  evitals_23_col = evitals_23,
  evitals_26_col = evitals_26,
  transport_disposition_col = edisposition_30,
  confidence_interval = TRUE
)
```

)

tbi_01_population	<i>TBI-01 Populations</i>
-------------------	---------------------------

Description

This function screens for potential traumatic brain injury (TBI) cases based on specific criteria in a patient dataset. It produces a subset of the data with calculated variables for TBI identification.

Usage

```
tbi_01_population(
  df = NULL,
  patient_scene_table = NULL,
  response_table = NULL,
  situation_table = NULL,
  disposition_table = NULL,
  vitals_table = NULL,
  erecord_01_col,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col,
  epatient_16_col,
  eresponse_05_col,
  esituation_11_col,
  esituation_12_col,
  transport_disposition_col,
  evitals_06_col,
  evitals_12_col,
  evitals_16_col,
  evitals_23_col,
  evitals_26_col
)
```

Arguments

df	A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.
patient_scene_table	A data.frame or tibble containing at least ePatient, and eScene as a fact table.
response_table	A data.frame or tibble containing at least the eResponse fields needed for this measure's calculations.
situation_table	A data.frame or tibble containing at least the eSituation fields needed for this measure's calculations. Default is NULL.

disposition_table	A data.frame or tibble containing only the edisposition fields needed for this measure's calculations.
vitals_table	A dataframe or tibble containing at least the eVitals fields needed.
erecord_01_col	The column representing the EMS record unique identifier.
incident_date_col	Column that contains the incident date. This defaults to NULL as it is optional in case not available due to PII restrictions.
patient_DOB_col	Column that contains the patient's date of birth. This defaults to NULL as it is optional in case not available due to PII restrictions.
epatient_15_col	Column representing the patient's numeric age agnostic of unit.
epatient_16_col	Column representing the patient's age unit ("Years", "Months", "Days", "Hours", or "Minutes").
eresponse_05_col	Column that contains eResponse.05 or the response type.
esituation_11_col	Column that contains eSituation.11 provider primary impression data.
esituation_12_col	Column that contains all eSituation.12 values as (possible a single comma-separated list), provider secondary impression data.
transport_disposition_col	One or more unquoted column names (such as edisposition.12, edisposition.30) containing transport disposition for an EMS event identifying whether a transport occurred and by which unit.
evitals_06_col	Numeric column containing systolic blood pressure values.
evitals_12_col	Numeric column containing pulse oximetry values.
evitals_16_col	Column with numeric value of the patient's exhaled end tidal carbon dioxide (ETCO2) level measured as a unit of pressure in millimeters of mercury (mmHg), percentage or, kilopascal (kPa).
evitals_23_col	Column for Glasgow Coma Scale (GCS) scores.
evitals_26_col	Column for AVPU alertness levels.

Value

A list that contains the following:

- a tibble with counts for each filtering step,
- a tibble for each population of interest
- a tibble for the initial population
- a tibble for the total dataset with computations
- a tibble with a summary of missingness for each column in each table

Author(s)

Nicolas Foss, Ed.D., MS

Examples

```

# create tables to test correct functioning

# patient table
patient_table <- tibble::tibble(

  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  incident_date = as.Date(c("2025-01-01", "2025-01-05",
                           "2025-02-01", "2025-01-01",
                           "2025-06-01"))
  ),
  patient_dob = as.Date(c("2000-01-01", "2020-01-01",
                        "2023-02-01", "2023-01-01",
                        "1970-06-01"))
  ),
  epatient_15 = c(25, 5, 2, 2, 55), # Ages
  epatient_16 = c("Years", "Years", "Years", "Years", "Years")

)

# response table
response_table <- tibble::tibble(

  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  eresponse_05 = rep(2205001, 5)

)

# situation table
situation_table <- tibble::tibble(

  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  esituation_11 = c(rep("S02", 3), rep("S06", 2)),
  esituation_12 = c(rep("S09.90", 2), rep("S06.0X9", 3)),

)

# vitals table
vitals_table <- tibble::tibble(

  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  evitals_06 = c(85, 80, 100, 90, 82),
  evitals_12 = c(95, 96, 97, 98, 99),
  evitals_16 = c(35, 36, 37, 38, 39),
  evitals_23 = rep(8, 5),
  evitals_26 = c("Verbal", "Painful", "Unresponsive", "Verbal", "Painful")

)

```

```

# disposition table
disposition_table <- tibble::tibble(
  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  edisposition_30 = c(4230001, 4230003, 4230001, 4230007, 4230007)
)

# test the success of the function
result <- tbi_01_population(patient_scene_table = patient_table,
  response_table = response_table,
  situation_table = situation_table,
  vitals_table = vitals_table,
  disposition_table = disposition_table,
  erecord_01_col = erecord_01,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col = epatient_15,
  epatient_16_col = epatient_16,
  eresponse_05_col = eresponse_05,
  esituation_11_col = esituation_11,
  esituation_12_col = esituation_12,
  evitals_06_col = evitals_06,
  evitals_12_col = evitals_12,
  evitals_16_col = evitals_16,
  evitals_23_col = evitals_23,
  evitals_26_col = evitals_26,
  transport_disposition_col = edisposition_30
)

# show the results of filtering at each step
result$filter_process

```

trauma_01

Trauma-01 Calculation

Description

This function processes EMS data to calculate the Trauma-01 performance measure, which evaluates the percentage of trauma patients assessed for pain using a numeric scale. The function filters and summarizes the data based on specified inclusion criteria.

Usage

```

trauma_01(
  df = NULL,
  patient_scene_table = NULL,
  response_table = NULL,
  situation_table = NULL,
  disposition_table = NULL,
  vitals_table = NULL,

```

```

erecord_01_col,
incident_date_col = NULL,
patient_DOB_col = NULL,
epatient_15_col,
epatient_16_col,
esituation_02_col,
eresponse_05_col,
evitals_23_col,
evitals_26_col,
evitals_27_col,
edisposition_28_col,
transport_disposition_col,
confidence_interval = FALSE,
method = c("wilson", "clopper-pearson"),
conf.level = 0.95,
correct = TRUE,
...
)

```

Arguments

df A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.

patient_scene_table A data.frame or tibble containing at least ePatient, and eScene as a fact table.

response_table A data.frame or tibble containing at least the eResponse fields needed for this measure's calculations.

situation_table A data.frame or tibble containing at least the eSituation fields needed for this measure's calculations. Default is NULL.

disposition_table A data.frame or tibble containing only the edisposition fields needed for this measure's calculations.

vitals_table A dataframe or tibble containing at least the eVitals fields needed.

erecord_01_col The column representing the EMS record unique identifier.

incident_date_col Column that contains the incident date. This defaults to NULL as it is optional in case not available due to PII restrictions.

patient_DOB_col Column that contains the patient's date of birth. This defaults to NULL as it is optional in case not available due to PII restrictions.

epatient_15_col Column representing the patient's numeric age agnostic of unit.

epatient_16_col Column representing the patient's age unit ("Years", "Months", "Days", "Hours", or "Minutes").

<code>esituation_02_col</code>	Column indicating whether or not there was an injury.
<code>eresponse_05_col</code>	Column that contains <code>eResponse.05</code> or the response type.
<code>evitals_23_col</code>	Column for Glasgow Coma Scale (GCS) scores.
<code>evitals_26_col</code>	Column for AVPU alertness levels.
<code>evitals_27_col</code>	Column giving the patient's indication of pain from a scale of 0-10.
<code>edisposition_28_col</code>	Column giving patient disposition for an EMS event identifying whether a patient was evaluated and care or services were provided.
<code>transport_disposition_col</code>	One or more unquoted column names (such as <code>edisposition.12</code> , <code>edisposition.30</code>) containing transport disposition for an EMS event identifying whether a transport occurred and by which unit.
<code>confidence_interval</code>	Logical. If TRUE, the function calculates a confidence interval for the proportion estimate.
<code>method</code>	Character. Specifies the method used to calculate confidence intervals. Options are "wilson" (Wilson score interval) and "clopper-pearson" (exact binomial interval). Partial matching is supported, so "w" and "c" can be used as shorthand.
<code>conf.level</code>	Numeric. The confidence level for the interval, expressed as a proportion (e.g., 0.95 for a 95% confidence interval). Defaults to 0.95.
<code>correct</code>	Logical. If TRUE, applies a continuity correction to the Wilson score interval when <code>method = "wilson"</code> . Defaults to TRUE.
<code>...</code>	optional additional arguments to pass onto <code>dplyr::summarize</code> .

Value

A `data.frame` summarizing results for two population groups (All, Adults and Peds) with the following columns:

- `pop`: Population type (All, Adults, and Peds).
- `numerator`: Count of incidents meeting the measure.
- `denominator`: Total count of included incidents.
- `prop`: Proportion of incidents meeting the measure.
- `prop_label`: Proportion formatted as a percentage with a specified number of decimal places.
- `lower_ci`: Lower bound of the confidence interval for `prop` (if `confidence_interval = TRUE`).
- `upper_ci`: Upper bound of the confidence interval for `prop` (if `confidence_interval = TRUE`).

Author(s)

Nicolas Foss, Ed.D., MS

Examples

```

# Synthetic test data
test_data <- tibble::tibble(
  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  epatient_15 = c(34, 5, 45, 2, 60), # Ages
  epatient_16 = c("Years", "Years", "Years", "Months", "Years"),
  eresponse_05 = rep(2205001, 5),
  esituation_02 = rep("Yes", 5),
  evitals_23 = rep(15, 5),
  evitals_26 = rep("Alert", 5),
  evitals_27 = c(0, 2, 4, 6, 8),
  edisposition_28 = rep(4228001, 5),
  edisposition_30 = c(4230001, 4230003, 4230001, 4230007, 4230007)
)

# Run the function
# Return 95% confidence intervals using the Wilson method
trauma_01(
  df = test_data,
  erecord_01_col = erecord_01,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col = epatient_15,
  epatient_16_col = epatient_16,
  eresponse_05_col = eresponse_05,
  esituation_02_col = esituation_02,
  evitals_23_col = evitals_23,
  evitals_26_col = evitals_26,
  evitals_27_col = evitals_27,
  edisposition_28_col = edisposition_28,
  transport_disposition_col = edisposition_30,
  confidence_interval = TRUE
)

```

trauma_01_population *Trauma-01 Population*

Description

Filters data down to the target populations for Trauma-08, and categorizes records to identify needed information for the calculations.

Identifies key categories to records that are 911 requests for patients with injury who were assessed for pain based on specific criteria and calculates related ECG measures. This function segments the data by age into adult and pediatric populations.

Usage

```

trauma_01_population(
  df = NULL,
  patient_scene_table = NULL,
  response_table = NULL,
  situation_table = NULL,
  disposition_table = NULL,
  vitals_table = NULL,
  erecord_01_col,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col,
  epatient_16_col,
  esituation_02_col,
  eresponse_05_col,
  evitals_23_col,
  evitals_26_col,
  evitals_27_col,
  edisposition_28_col,
  transport_disposition_col
)

```

Arguments

df A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.

patient_scene_table A data.frame or tibble containing at least ePatient, and eScene as a fact table.

response_table A data.frame or tibble containing at least the eResponse fields needed for this measure's calculations.

situation_table A data.frame or tibble containing at least the eSituation fields needed for this measure's calculations. Default is NULL.

disposition_table A data.frame or tibble containing only the edisposition fields needed for this measure's calculations.

vitals_table A dataframe or tibble containing at least the eVitals fields needed.

erecord_01_col The column representing the EMS record unique identifier.

incident_date_col Column that contains the incident date. This defaults to NULL as it is optional in case not available due to PII restrictions.

patient_DOB_col Column that contains the patient's date of birth. This defaults to NULL as it is optional in case not available due to PII restrictions.

epatient_15_col Column representing the patient's numeric age agnostic of unit.

epatient_16_col	Column representing the patient's age unit ("Years", "Months", "Days", "Hours", or "Minutes").
esituation_02_col	Column indicating whether or not there was an injury.
eresponse_05_col	Column that contains eResponse.05 or the response type.
evitals_23_col	Column for Glasgow Coma Scale (GCS) scores.
evitals_26_col	Column for AVPU alertness levels.
evitals_27_col	Column giving the patient's indication of pain from a scale of 0-10.
edisposition_28_col	Column name for patient care disposition details.
transport_disposition_col	One or more unquoted column names (such as edisposition.12, edisposition.30) containing transport disposition for an EMS event identifying whether a transport occurred and by which unit.

Value

A list that contains the following:

- a tibble with counts for each filtering step,
- a tibble for each population of interest
- a tibble for the initial population
- a tibble for the total dataset with computations
- a tibble with a summary of missingness for each column in each table

Author(s)

Nicolas Foss, Ed.D., MS

Examples

```
# create tables to test correct functioning

# patient table
patient_table <- tibble::tibble(

  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  incident_date = as.Date(c("2025-01-01", "2025-01-05",
                           "2025-02-01", "2025-01-01",
                           "2025-06-01")
                        ),
  patient_dob = as.Date(c("2000-01-01", "2020-01-01",
                        "2023-02-01", "2023-01-01",
                        "1970-06-01")
                     ),
  epatient_15 = c(25, 5, 2, 2, 55), # Ages
```

```

    epatient_16 = c("Years", "Years", "Years", "Years", "Years")
  )

  # response table
  response_table <- tibble::tibble(

    erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
    eresponse_05 = rep(2205001, 5)

  )

  # situation table
  situation_table <- tibble::tibble(

    erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
    esituation_02 = rep("Yes", 5),

  )

  # vitals table
  vitals_table <- tibble::tibble(

    erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
    evitals_23 = rep(15, 5),
    evitals_26 = rep("Alert", 5),
    evitals_27 = c(0, 2, 4, 6, 8)

  )

  # disposition table
  disposition_table <- tibble::tibble(
    erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
    edisposition_28 = rep(4228001, 5),
    edisposition_30 = c(4230001, 4230003, 4230001, 4230007, 4230007)
  )

  # test the success of the function
  result <- trauma_01_population(patient_scene_table = patient_table,
                                response_table = response_table,
                                situation_table = situation_table,
                                vitals_table = vitals_table,
                                disposition_table = disposition_table,
                                erecord_01_col = erecord_01,
                                incident_date_col = incident_date,
                                patient_DOB_col = patient_dob,
                                epatient_15_col = epatient_15,
                                epatient_16_col = epatient_16,
                                eresponse_05_col = eresponse_05,
                                esituation_02_col = esituation_02,
                                evitals_23_col = evitals_23,
                                evitals_26_col = evitals_26,
                                evitals_27_col = evitals_27,
                                edisposition_28_col = edisposition_28,
                                transport_disposition_col = edisposition_30
  )

```

```

    )

# show the results of filtering at each step
result$filter_process

```

trauma_03

Trauma-03 Calculation

Description

This function calculates the "Trauma-03" measure, which evaluates pain scale reassessment for trauma patients, using a comprehensive data frame with EMS records. The function processes input data to create both fact and dimension tables, identifies eligible patients, and summarizes results for adult and pediatric populations.

Usage

```

trauma_03(
  df = NULL,
  patient_scene_table = NULL,
  response_table = NULL,
  situation_table = NULL,
  disposition_table = NULL,
  vitals_table = NULL,
  erecord_01_col,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col,
  epatient_16_col,
  esituation_02_col,
  eresponse_05_col,
  edisposition_28_col,
  transport_disposition_col,
  evitals_01_col,
  evitals_27_col = NULL,
  evitals_27_initial_col = NULL,
  evitals_27_last_col = NULL,
  confidence_interval = FALSE,
  method = c("wilson", "clopper-pearson"),
  conf.level = 0.95,
  correct = TRUE,
  ...
)

```

Arguments

<code>df</code>	A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.
<code>patient_scene_table</code>	A data.frame or tibble containing at least ePatient, and eScene as a fact table.
<code>response_table</code>	A data.frame or tibble containing at least the eResponse fields needed for this measure's calculations.
<code>situation_table</code>	A data.frame or tibble containing at least the eSituation fields needed for this measure's calculations. Default is NULL.
<code>disposition_table</code>	A data.frame or tibble containing only the edisposition fields needed for this measure's calculations.
<code>vitals_table</code>	A dataframe or tibble containing at least the eVitals fields needed.
<code>erecord_01_col</code>	The column representing the EMS record unique identifier.
<code>incident_date_col</code>	Column that contains the incident date. This defaults to NULL as it is optional in case not available due to PII restrictions.
<code>patient_DOB_col</code>	Column that contains the patient's date of birth. This defaults to NULL as it is optional in case not available due to PII restrictions.
<code>epatient_15_col</code>	Column representing the patient's numeric age agnostic of unit.
<code>epatient_16_col</code>	Column representing the patient's age unit ("Years", "Months", "Days", "Hours", or "Minutes").
<code>esituation_02_col</code>	Column indicating whether or not there was an injury.
<code>eresponse_05_col</code>	Column that contains eResponse.05 or the response type.
<code>edisposition_28_col</code>	Column name for patient care disposition details.
<code>transport_disposition_col</code>	One or more unquoted column names (such as edisposition.12, edisposition.30) containing transport disposition for an EMS event identifying whether a transport occurred and by which unit.
<code>evitals_01_col</code>	Date-time or POSIXct column containing vital signs date/time
<code>evitals_27_col</code>	Column giving the patient's indication of pain from a scale of 0-10.
<code>evitals_27_initial_col</code>	The column for the initial pain scale score. Default is NULL.
<code>evitals_27_last_col</code>	The column for the last pain scale score. Default is NULL.
<code>confidence_interval</code>	Logical. If TRUE, the function calculates a confidence interval for the proportion estimate.

method	Character. Specifies the method used to calculate confidence intervals. Options are "wilson" (Wilson score interval) and "clopper-pearson" (exact binomial interval). Partial matching is supported, so "w" and "c" can be used as shorthand.
conf.level	Numeric. The confidence level for the interval, expressed as a proportion (e.g., 0.95 for a 95% confidence interval). Defaults to 0.95.
correct	Logical. If TRUE, applies a continuity correction to the Wilson score interval when method = "wilson". Defaults to TRUE.
...	optional additional arguments to pass onto <code>dplyr::summarize</code> .

Value

A data.frame summarizing results for two population groups (All, Adults and Peds) with the following columns:

- pop: Population type (All, Adults, and Peds).
- numerator: Count of incidents meeting the measure.
- denominator: Total count of included incidents.
- prop: Proportion of incidents meeting the measure.
- prop_label: Proportion formatted as a percentage with a specified number of decimal places.
- lower_ci: Lower bound of the confidence interval for prop (if confidence_interval = TRUE).
- upper_ci: Upper bound of the confidence interval for prop (if confidence_interval = TRUE).

Author(s)

Nicolas Foss, Ed.D., MS

Examples

```
# Synthetic test data
# for testing a single pain scale column
test_data2 <- tibble::tibble(
  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  epatient_15 = c(34, 5, 45, 2, 60), # Ages
  epatient_16 = c("Years", "Years", "Years", "Months", "Years"),
  eresponse_05 = rep(2205001, 5),
  esituation_02 = rep("Yes", 5),
  evitals_01 = lubridate::as_datetime(c("2025-01-01 12:00:00", "2025-01-05
18:00:00", "2025-02-01 06:00:00", "2025-01-01 01:00:00", "2025-06-01
14:00:00")),
  edisposition_28 = rep(4228001, 5),
  edisposition_30 = c(4230001, 4230003, 4230001, 4230007, 4230007)
)

# Expand data so each erecord_01 has 2 rows (one for each pain score)
test_data_expanded2 <- test_data2 |>
```

```

tidyr::uncount(weights = 2) |> # Duplicate each row twice
# Assign pain scores
dplyr::mutate(evitals_27 = c(0, 0, 2, 1, 4, 3, 6, 5, 8, 7)) |>
dplyr::group_by(erecord_01) |>
dplyr::mutate(
# Lower score = later time
  time_offset = dplyr::if_else(dplyr::row_number() == 1, -5, 0),
  evitals_01 = evitals_01 + lubridate::dminutes(time_offset)
) |>
dplyr::ungroup() |>
dplyr::select(-time_offset) # Remove temporary column

# Run function with the single pain score column
# Return 95% confidence intervals using the Wilson method
trauma_03(
  df = test_data_expanded2,
  erecord_01_col = erecord_01,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col = epatient_15,
  epatient_16_col = epatient_16,
  eresponse_05_col = eresponse_05,
  esituation_02_col = esituation_02,
  evitals_01_col = evitals_01,
  evitals_27_initial_col = NULL,
  evitals_27_last_col = NULL,
  evitals_27_col = evitals_27,
  edisposition_28_col = edisposition_28,
  transport_disposition_col = edisposition_30,
  confidence_interval = TRUE
)

```

trauma_03_population *Trauma-03 Populations*

Description

Filters data down to the target populations for Trauma-08, and categorizes records to identify needed information for the calculations.

Identifies key categories to records that are 911 request for patients whose pain score was lowered during the EMS encounter. based on specific criteria and calculates related ECG measures. This function segments the data by age into adult and pediatric populations.

Usage

```

trauma_03_population(
  df = NULL,
  patient_scene_table = NULL,

```

```

response_table = NULL,
situation_table = NULL,
disposition_table = NULL,
vitals_table = NULL,
erecord_01_col,
incident_date_col = NULL,
patient_DOB_col = NULL,
epatient_15_col,
epatient_16_col,
esituation_02_col,
eresponse_05_col,
evitals_01_col,
evitals_27_col = NULL,
evitals_27_initial_col = NULL,
evitals_27_last_col = NULL,
edisposition_28_col,
transport_disposition_col
)

```

Arguments

df A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.

patient_scene_table A data.frame or tibble containing at least ePatient, and eScene as a fact table.

response_table A data.frame or tibble containing at least the eResponse fields needed for this measure's calculations.

situation_table A data.frame or tibble containing at least the eSituation fields needed for this measure's calculations. Default is NULL.

disposition_table A data.frame or tibble containing only the edisposition fields needed for this measure's calculations.

vitals_table A dataframe or tibble containing at least the eVitals fields needed.

erecord_01_col The column representing the EMS record unique identifier.

incident_date_col Column that contains the incident date. This defaults to NULL as it is optional in case not available due to PII restrictions.

patient_DOB_col Column that contains the patient's date of birth. This defaults to NULL as it is optional in case not available due to PII restrictions.

epatient_15_col Column representing the patient's numeric age agnostic of unit.

epatient_16_col Column representing the patient's age unit ("Years", "Months", "Days", "Hours", or "Minutes").

esituation_02_col	Column indicating whether or not there was an injury.
eresponse_05_col	Column that contains eResponse.05 or the response type.
evitals_01_col	The column for the date/time vital signs were taken on the patient.
evitals_27_col	Column giving the patient's indication of pain from a scale of 0-10.
evitals_27_initial_col	The column for the initial pain scale score. Default is NULL.
evitals_27_last_col	The column for the last pain scale score. Default is NULL.
edisposition_28_col	Column name for patient care disposition details.
transport_disposition_col	One or more unquoted column names (such as edisposition.12, edisposition.30) containing transport disposition for an EMS event identifying whether a transport occurred and by which unit.

Value

A list that contains the following:

- a tibble with counts for each filtering step,
- a tibble for each population of interest
- a tibble for the initial population
- a tibble for the total dataset with computations
- a tibble with a summary of missingness for each column in each table

Author(s)

Nicolas Foss, Ed.D., MS

Examples

```
# create tables to test correct functioning

# patient table
patient_table <- tibble::tibble(

  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  incident_date = as.Date(c("2025-01-01", "2025-01-05",
                           "2025-02-01", "2025-01-01",
                           "2025-06-01")
                        ),
  patient_dob = as.Date(c("2000-01-01", "2020-01-01",
                         "2023-02-01", "2023-01-01",
                         "1970-06-01")
                     ),
  epatient_15 = c(25, 5, 2, 2, 55), # Ages
```

```

    epatient_16 = c("Years", "Years", "Years", "Years", "Years")
  )

  # response table
  response_table <- tibble::tibble(

    erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
    eresponse_05 = rep(2205001, 5)

  )

  # situation table
  situation_table <- tibble::tibble(

    erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
    esituation_02 = rep("Yes", 5),

  )

  # vitals table for a single pain scale column
  vitals_table <- tibble::tibble(

    erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
    evitals_01 = lubridate::as_datetime(c("2025-01-01 12:00:00", "2025-01-05
18:00:00", "2025-02-01 06:00:00", "2025-01-01 01:00:00", "2025-06-01
14:00:00"))

  ) |>
  tidyr::uncount(weights = 2) |> # Duplicate each row twice
  # Assign pain scores
  dplyr::mutate(evitals_27 = c(0, 0, 2, 1, 4, 3, 6, 5, 8, 7)) |>
  dplyr::group_by(erecord_01) |>
  dplyr::mutate(
    # Lower score = later time
    time_offset = dplyr::if_else(dplyr::row_number() == 1, -5, 0),
    evitals_01 = evitals_01 + lubridate::dminutes(time_offset)
  ) |>
  dplyr::ungroup() |>
  dplyr::select(-time_offset) # Remove temporary column

  # disposition table
  disposition_table <- tibble::tibble(
    erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
    edisposition_28 = rep(4228001, 5),
    edisposition_30 = c(4230001, 4230003, 4230001, 4230007, 4230007)
  )

  # test the success of the function
  # use the single pain scale column
  result <- trauma_03_population(patient_scene_table = patient_table,
                                response_table = response_table,
                                situation_table = situation_table,
                                vitals_table = vitals_table,
                                disposition_table = disposition_table,

```

```

        erecord_01_col = erecord_01,
        incident_date_col = incident_date,
        patient_DOB_col = patient_dob,
        epatient_15_col = epatient_15,
        epatient_16_col = epatient_16,
        eresponse_05_col = eresponse_05,
        esituation_02_col = esituation_02,
        evitals_01_col = evitals_01,
        evitals_27_initial_col = NULL,
        evitals_27_last_col = NULL,
        evitals_27_col = evitals_27,
        edisposition_28_col = edisposition_28,
        transport_disposition_col = edisposition_30
    )

# show the results of filtering at each step
result$filter_process

```

trauma_04

Trauma-04 Calculations

Description

This function processes EMS data to generate a set of binary variables indicating whether specific trauma triage criteria are met. The output is a data frame enriched with these indicators for further analysis. The final outcome is whether or not the EMS record documents the use of a verified trauma center levels 1-5 in the hospital capability documentation.

Usage

```

trauma_04(
  df = NULL,
  patient_scene_table = NULL,
  response_table = NULL,
  situation_table = NULL,
  vitals_table = NULL,
  exam_table = NULL,
  procedures_table = NULL,
  injury_table = NULL,
  disposition_table = NULL,
  erecord_01_col,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col,
  epatient_16_col,
  esituation_02_col,
  eresponse_05_col,

```

```

eresponse_10_col,
transport_disposition_col,
edisposition_23_col = lifecycle::deprecated(),
edisposition_02_col,
trauma_center_facility_IDs,
evitals_06_col,
evitals_10_col,
evitals_12_col,
evitals_14_col,
evitals_15_col,
evitals_21_col,
eexam_16_col,
eexam_20_col,
eexam_23_col,
eexam_25_col,
eprocedures_03_col,
einjury_01_col,
einjury_03_col,
einjury_04_col,
einjury_09_col,
confidence_interval = FALSE,
method = c("wilson", "clopper-pearson"),
conf.level = 0.95,
correct = TRUE,
...
)

```

Arguments

df A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.

patient_scene_table A data.frame or tibble containing at least ePatient, and eScene as a fact table.

response_table A data.frame or tibble containing at least the eResponse fields needed for this measure's calculations.

situation_table A data.frame or tibble containing at least the eSituation fields needed for this measure's calculations. Default is NULL.

vitals_table A dataframe or tibble containing at least the eVitals fields needed.

exam_table A data.frame or tibble containing only the eExam fields needed for this measure's calculations. Default is NULL.

procedures_table A dataframe or tibble containing at least the eProcedures fields needed.

injury_table A data frame or tibble containing fields from eInjury needed for this measure's calculations.

disposition_table A data.frame or tibble containing only the edisposition fields needed for this measure's calculations.

erecord_01_col	The column representing the EMS record unique identifier.
incident_date_col	Column that contains the incident date. This defaults to NULL as it is optional in case not available due to PII restrictions.
patient_DOB_col	Column that contains the patient's date of birth. This defaults to NULL as it is optional in case not available due to PII restrictions.
epatient_15_col	Column representing the patient's numeric age agnostic of unit.
epatient_16_col	Column representing the patient's age unit ("Years", "Months", "Days", "Hours", or "Minutes").
esituation_02_col	Column indicating whether or not there was an injury.
eresponse_05_col	Column that contains eResponse.05 or the response type.
eresponse_10_col	Column name containing informatin about scene delays, if any, of the EMS unit associated with the EMS event.
transport_disposition_col	One or more unquoted column names (such as edisposition.12, edisposition.30) containing transport disposition for an EMS event identifying whether a transport occurred and by which unit.
edisposition_23_col	[Deprecated] Use edisposition_02_col instead. You must also pass a character vector of trauma center facility IDs to trauma_center_facility_IDs to ensure that destination facility IDs passed via edisposition_02_col are correctly identified as trauma centers as applicable.
edisposition_02_col	Column name containing the code of the destination the patient was delivered or transferred to.
trauma_center_facility_IDs	A character vector of trauma center facility IDs that will allow destination facilities documented in edisposition_02_col to be classified correctly as trauma centers when applicable.
evitals_06_col	Numeric column containing systolic blood pressure values.
evitals_10_col	Column name containing the patient's heart rate expressed as a number per minute.
evitals_12_col	Numeric column containing pulse oximetry values.
evitals_14_col	Column name containing the patient's respiratory rate expressed as a number per minute.
evitals_15_col	Column name containing the patient's respiratory effort.
evitals_21_col	Column name containing the patient's Glasgow Coma Score Motor response.
eexam_16_col	Column name containing the assessment findings associated with the patient's extremities.

eexam_20_col	Column name containing the assessment findings of the patient's neurological examination.
eexam_23_col	Column name containing the assessment findings associated with the patient's lungs.
eexam_25_col	Column name containing the assessment findings associated with the patient's chest.
eprocedures_03_col	Column containing procedure codes with or without procedure names.
einjury_01_col	Column name containing the category of the reported/suspected external cause of the injury.
einjury_03_col	Column describing Trauma triage criteria for the red boxes (Injury Patterns and Mental Status and Vital Signs) in the 2021 ACS National Guideline for the Field Triage of Injured Patients.
einjury_04_col	Column name containing Trauma triage criteria for the yellow boxes (Mechanism of Injury and EMS Judgment) in the current ACS National Guideline for the Field Triage of Injured Patients.
einjury_09_col	Column name containing the distance in feet the patient fell, measured from the lowest point of the patient to the ground.
confidence_interval	Logical. If TRUE, the function calculates a confidence interval for the proportion estimate.
method	Character. Specifies the method used to calculate confidence intervals. Options are "wilson" (Wilson score interval) and "clopper-pearson" (exact binomial interval). Partial matching is supported, so "w" and "c" can be used as shorthand.
conf.level	Numeric. The confidence level for the interval, expressed as a proportion (e.g., 0.95 for a 95% confidence interval). Defaults to 0.95.
correct	Logical. If TRUE, applies a continuity correction to the Wilson score interval when method = "wilson". Defaults to TRUE.
...	optional additional arguments to pass onto <code>dplyr::summarize</code> .

Value

A `data.frame` summarizing results for two population groups (All, Adults and Peds) with the following columns:

- `pop`: Population type (All, Adults, and Peds).
- `numerator`: Count of incidents meeting the measure.
- `denominator`: Total count of included incidents.
- `prop`: Proportion of incidents meeting the measure.
- `prop_label`: Proportion formatted as a percentage with a specified number of decimal places.
- `lower_ci`: Lower bound of the confidence interval for `prop` (if `confidence_interval = TRUE`).
- `upper_ci`: Upper bound of the confidence interval for `prop` (if `confidence_interval = TRUE`).

Author(s)

Nicolas Foss, Ed.D., MS

Examples

```

# Synthetic test data
test_data <- tibble::tibble(
  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  epatient_15 = c(34, 5, 45, 2, 60), # Ages
  epatient_16 = c("Years", "Years", "Years", "Months", "Years"),
  eresponse_05 = rep(2205001, 5),
  eresponse_10 = rep(2210011, 5),
  esituation_02 = rep("Yes", 5),
  evitals_06 = c(100, 90, 80, 70, 85),
  evitals_10 = c(110, 89, 88, 71, 85),
  evitals_12 = c(50, 60, 70, 80, 75),
  evitals_14 = c(30, 9, 8, 7, 31),
  evitals_15 = c("apneic", "labored", "rapid", "shallow", "weak/agonal"),
  evitals_21 = c(5, 4, 3, 2, 1),
  eexam_16 = c(3516043, 3516067, 3516043, 3516067, 3516067),
  eexam_20 = c(3520045, 3520043, 3520019, 3520017, 3520017),
  eexam_23 = c(3523011, 3523003, 3523001, 3523011, 3523003),
  eexam_25 = c(3525039, 3525023, 3525005, 3525039, 3525023),
  edisposition_02 = c(9908029, 9908027, 9908025, 9908023, 9876543),
  edisposition_30 = c(4230001, 4230003, 4230001, 4230007, 4230007),
  eprocedures_03 = c(424979004, 427753009, 429705000, 47545007, 243142003),
  einjury_01 = c("V20", "V36", "V86", "V39", "V32"),
  einjury_03 = c(2903011, 2903009, 2903005, 3903003, 2903001),
  einjury_04 = c(2904013, 2904011, 2904009, 2904007, 2904001),
  einjury_09 = c(11, 12, 13, 14, 15)
)

# Run function with the first and last pain score columns
# Return 95% confidence intervals using the Wilson method
# test the success of the function
result <- trauma_04_population(
  df = test_data,
  erecord_01_col = erecord_01,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col = epatient_15,
  epatient_16_col = epatient_16,
  eresponse_05_col = eresponse_05,
  eresponse_10_col = eresponse_10,
  esituation_02_col = esituation_02,
  evitals_06_col = evitals_06,
  evitals_10_col = evitals_10,
  evitals_12_col = evitals_12,
  evitals_14_col = evitals_14,
  evitals_15_col = evitals_15,
  evitals_21_col = evitals_21,
  eexam_16_col = eexam_16,

```

```

eexam_20_col = eexam_20,
eexam_23_col = eexam_23,
eexam_25_col = eexam_25,
edisposition_02_col = edisposition_02,
trauma_center_facility_IDs = as.character(c(
  9908029,
  9908027,
  9908025,
  9908023,
  9908021
)),
transport_disposition_col = edisposition_30,
eprocedures_03_col = eprocedures_03,
einjury_01_col = einjury_01,
einjury_03_col = einjury_03,
einjury_04_col = einjury_04,
einjury_09_col = einjury_09
)

```

trauma_04_population *Trauma-04 Populations*

Description

This function processes EMS data to generate the population needed to calculate the Trauma-04 NEMSQA measure.

Usage

```

trauma_04_population(
  df = NULL,
  patient_scene_table = NULL,
  response_table = NULL,
  situation_table = NULL,
  vitals_table = NULL,
  exam_table = NULL,
  procedures_table = NULL,
  injury_table = NULL,
  disposition_table = NULL,
  erecord_01_col,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col,
  epatient_16_col,
  esituation_02_col,
  eresponse_05_col,
  eresponse_10_col,

```

```

transport_disposition_col,
edisposition_23_col = lifecycle::deprecated(),
edisposition_02_col,
trauma_center_facility_IDs,
evitals_06_col,
evitals_10_col,
evitals_12_col,
evitals_14_col,
evitals_15_col,
evitals_21_col,
eexam_16_col,
eexam_20_col,
eexam_23_col,
eexam_25_col,
eprocedures_03_col,
einjury_01_col,
einjury_03_col,
einjury_04_col,
einjury_09_col
)

```

Arguments

df A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.

patient_scene_table A data.frame or tibble containing at least ePatient, and eScene as a fact table.

response_table A data.frame or tibble containing at least the eResponse fields needed for this measure's calculations.

situation_table A data.frame or tibble containing at least the eSituation fields needed for this measure's calculations. Default is NULL.

vitals_table A dataframe or tibble containing at least the eVitals fields needed.

exam_table A data.frame or tibble containing only the eExam fields needed for this measure's calculations. Default is NULL.

procedures_table A dataframe or tibble containing at least the eProcedures fields needed.

injury_table A data frame or tibble containing fields from eInjury needed for this measure's calculations.

disposition_table A data.frame or tibble containing only the edisposition fields needed for this measure's calculations.

erecord_01_col The column representing the EMS record unique identifier.

incident_date_col Column that contains the incident date. This defaults to NULL as it is optional in case not available due to PII restrictions.

patient_DOB_col	Column that contains the patient's date of birth. This defaults to NULL as it is optional in case not available due to PII restrictions.
epatient_15_col	Column representing the patient's numeric age agnostic of unit.
epatient_16_col	Column representing the patient's age unit ("Years", "Months", "Days", "Hours", or "Minutes").
esituation_02_col	Column indicating whether or not there was an injury.
eresponse_05_col	Column that contains eResponse.05 or the response type.
eresponse_10_col	Column name containing informatin about scene delays, if any, of the EMS unit associated with the EMS event.
transport_disposition_col	One or more unquoted column names (such as edisposition.12, edisposition.30) containing transport disposition for an EMS event identifying whether a transport occurred and by which unit.
edisposition_23_col	[Deprecated] Use edisposition_02_col instead. You must also pass a character vector of trauma center facility IDs to trauma_center_facility_IDs to ensure that destination facility IDs passed via edisposition_02_col are correctly identified as trauma centers as applicable.
edisposition_02_col	Column name containing the code of the destination the patient was delivered or transferred to.
trauma_center_facility_IDs	A character vector of trauma center facility IDs that will allow destination facilities documented in edisposition_02_col to be classified correctly as trauma centers when applicable.
evitals_06_col	Numeric column containing systolic blood pressure values.
evitals_10_col	Column name containing the patient's heart rate expressed as a number per minute.
evitals_12_col	Numeric column containing pulse oximetry values.
evitals_14_col	Column name containing the patient's respiratory rate expressed as a number per minute.
evitals_15_col	Column name containing the patient's respiratory effort.
evitals_21_col	Column name containing the patient's Glasgow Coma Score Motor response.
eexam_16_col	Column name containing the assessment findings associated with the patient's extremities.
eexam_20_col	Column name containing the assessment findings of the patient's neurological examination.
eexam_23_col	Column name containing the assessment findings associated with the patient's lungs.

eexam_25_col	Column name containing the assessment findings associated with the patient's chest.
eprocedures_03_col	Column containing procedure codes with or without procedure names.
einjury_01_col	Column name containing the category of the reported/suspected external cause of the injury.
einjury_03_col	Column describing Trauma triage criteria for the red boxes (Injury Patterns and Mental Status and Vital Signs) in the 2021 ACS National Guideline for the Field Triage of Injured Patients.
einjury_04_col	Column name containing Trauma triage criteria for the yellow boxes (Mechanism of Injury and EMS Judgment) in the current ACS National Guideline for the Field Triage of Injured Patients.
einjury_09_col	Column name containing the distance in feet the patient fell, measured from the lowest point of the patient to the ground.

Value

A list that contains the following:

- a tibble with counts for each filtering step,
- a tibble for each population of interest
- a tibble for the initial population
- a tibble for the total dataset with computations
- a tibble with a summary of missingness for each column in each table

Author(s)

Nicolas Foss, Ed.D., MS

Examples

```
# create tables to test correct functioning

# patient table
patient_table <- tibble::tibble(

  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  incident_date = as.Date(c("2025-01-01", "2025-01-05",
                           "2025-02-01", "2025-01-01",
                           "2025-06-01")
  ),
  patient_dob = as.Date(c("2000-01-01", "2020-01-01",
                          "2023-02-01", "2023-01-01",
                          "1970-06-01")
  ),
  epatient_15 = c(25, 5, 2, 2, 55), # Ages
  epatient_16 = c("Years", "Years", "Years", "Years", "Years")
```

```

)

# response table
response_table <- tibble::tibble(

  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  eresponse_05 = rep(2205001, 5),
  eresponse_10 = rep(2210011, 5)
)

# situation table
situation_table <- tibble::tibble(

  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  esituation_02 = rep("Yes", 5),
)

# vitals table
vitals_table <- tibble::tibble(

  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  evitals_06 = c(100, 90, 80, 70, 85),
  evitals_10 = c(110, 89, 88, 71, 85),
  evitals_12 = c(50, 60, 70, 80, 75),
  evitals_14 = c(30, 9, 8, 7, 31),
  evitals_15 = c("apneic", "labored", "rapid", "shallow", "weak/agonal"),
  evitals_21 = c(5, 4, 3, 2, 1)
)

# disposition table
disposition_table <- tibble::tibble(
  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  edisposition_02 = as.character(c(
    9908029,
    9908027,
    9908025,
    9908023,
    9876543
  )),
  edisposition_30 = c(4230001, 4230003, 4230001, 4230007, 4230007)
)

# injury table
injury_table <- tibble::tibble(
  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  einjury_01 = c("V20", "V36", "V86", "V39", "V32"),
  einjury_03 = c(2903011, 2903009, 2903005, 3903003, 2903001),
  einjury_04 = c(2904013, 2904011, 2904009, 2904007, 2904001),
  einjury_09 = c(11, 12, 13, 14, 15)
)

# exam table
exam_table <- tibble::tibble(

```

```

erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
eexam_16 = c(3516043, 3516067, 3516043, 3516067, 3516067),
eexam_20 = c(3520045, 3520043, 3520019, 3520017, 3520017),
eexam_23 = c(3523011, 3523003, 3523001, 3523011, 3523003),
eexam_25 = c(3525039, 3525023, 3525005, 3525039, 3525023)
)

# procedures table
procedures_table <- tibble::tibble(
  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  eprocedures_03 = c(424979004, 427753009, 429705000, 47545007, 243142003)
)

# test the success of the function
result <- trauma_04_population(
  patient_scene_table = patient_table,
  response_table = response_table,
  situation_table = situation_table,
  vitals_table = vitals_table,
  disposition_table = disposition_table,
  exam_table = exam_table,
  injury_table = injury_table,
  procedures_table = procedures_table,
  erecord_01_col = erecord_01,
  incident_date_col = incident_date,
  patient_DOB_col = patient_dob,
  epatient_15_col = epatient_15,
  epatient_16_col = epatient_16,
  eresponse_05_col = eresponse_05,
  eresponse_10_col = eresponse_10,
  esituation_02_col = esituation_02,
  evitals_06_col = evitals_06,
  evitals_10_col = evitals_10,
  evitals_12_col = evitals_12,
  evitals_14_col = evitals_14,
  evitals_15_col = evitals_15,
  evitals_21_col = evitals_21,
  eexam_16_col = eexam_16,
  eexam_20_col = eexam_20,
  eexam_23_col = eexam_23,
  eexam_25_col = eexam_25,
  edisposition_02_col = edisposition_02,
  trauma_center_facility_IDs = as.character(c(
    9908029,
    9908027,
    9908025,
    9908023,
    9908021
  )),
  transport_disposition_col = edisposition_30,
  eprocedures_03_col = eprocedures_03,
  einjury_01_col = einjury_01,
  einjury_03_col = einjury_03,

```

```

    einjury_04_col = einjury_04,
    einjury_09_col = einjury_09
  )

  # show the results of filtering at each step
  result$filter_process

```

 trauma_08

Trauma-08 Calculation

Description

This function calculates the Trauma-08 measure, which evaluates the completeness of pain scale documentation for patients experiencing traumatic injury. It determines the total population, adult population, and pediatric population meeting the criteria for the Trauma-08 measure.

Usage

```

trauma_08(
  df = NULL,
  patient_scene_table = NULL,
  response_table = NULL,
  situation_table = NULL,
  disposition_table = NULL,
  vitals_table = NULL,
  erecord_01_col,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col,
  epatient_16_col,
  esituation_02_col,
  eresponse_05_col,
  transport_disposition_col,
  evitals_06_col,
  evitals_14_col,
  evitals_23_col,
  confidence_interval = FALSE,
  method = c("wilson", "clopper-pearson"),
  conf.level = 0.95,
  correct = TRUE,
  ...
)

```

Arguments

df A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.

patient_scene_table	A data.frame or tibble containing at least ePatient, and eScene as a fact table.
response_table	A data.frame or tibble containing at least the eResponse fields needed for this measure's calculations.
situation_table	A data.frame or tibble containing at least the eSituation fields needed for this measure's calculations. Default is NULL.
disposition_table	A data.frame or tibble containing only the edisposition fields needed for this measure's calculations.
vitals_table	A dataframe or tibble containing at least the eVitals fields needed.
erecord_01_col	The column representing the EMS record unique identifier.
incident_date_col	Column that contains the incident date. This defaults to NULL as it is optional in case not available due to PII restrictions.
patient_DOB_col	Column that contains the patient's date of birth. This defaults to NULL as it is optional in case not available due to PII restrictions.
epatient_15_col	Column representing the patient's numeric age agnostic of unit.
epatient_16_col	Column representing the patient's age unit ("Years", "Months", "Days", "Hours", or "Minutes").
esituation_02_col	Column indicating whether or not there was an injury.
eresponse_05_col	Column that contains eResponse.05 or the response type.
transport_disposition_col	One or more unquoted column names (such as edisposition.12, edisposition.30) containing transport disposition for an EMS event identifying whether a transport occurred and by which unit.
evitals_06_col	Numeric column containing systolic blood pressure values.
evitals_14_col	Column containing data on patient's respiratory rate expressed as a number per minute.
evitals_23_col	Column for Glasgow Coma Scale (GCS) scores.
confidence_interval	Logical. If TRUE, the function calculates a confidence interval for the proportion estimate.
method	Character. Specifies the method used to calculate confidence intervals. Options are "wilson" (Wilson score interval) and "clopper-pearson" (exact binomial interval). Partial matching is supported, so "w" and "c" can be used as shorthand.
conf.level	Numeric. The confidence level for the interval, expressed as a proportion (e.g., 0.95 for a 95% confidence interval). Defaults to 0.95.
correct	Logical. If TRUE, applies a continuity correction to the Wilson score interval when method = "wilson". Defaults to TRUE.
...	optional additional arguments to pass onto <code>dplyr::summarize</code> .

Value

A data.frame summarizing results for two population groups (All, Adults and Peds) with the following columns:

- pop: Population type (All, Adults, and Peds).
- numerator: Count of incidents meeting the measure.
- denominator: Total count of included incidents.
- prop: Proportion of incidents meeting the measure.
- prop_label: Proportion formatted as a percentage with a specified number of decimal places.
- lower_ci: Lower bound of the confidence interval for prop (if confidence_interval = TRUE).
- upper_ci: Upper bound of the confidence interval for prop (if confidence_interval = TRUE).

Author(s)

Nicolas Foss, Ed.D., MS

Examples

```
# Synthetic test data
test_data <- tibble::tibble(
  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  epatient_15 = c(34, 5, 45, 2, 60), # Ages
  epatient_16 = c("Years", "Years", "Years", "Months", "Years"),
  eresponse_05 = rep(2205001, 5),
  esituation_02 = rep("Yes", 5),
  evitals_06 = c(100, 90, 80, 70, 85),
  evitals_14 = c(30, 9, 8, 7, 31),
  evitals_23 = c(6, 7, 8, 8, 7),
  edisposition_30 = c(4230001, 4230003, 4230001, 4230007, 4230007)
)

# Run function with the first and last pain score columns
# Return 95% confidence intervals using the Wilson method
trauma_08(
  df = test_data,
  erecord_01_col = erecord_01,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col = epatient_15,
  epatient_16_col = epatient_16,
  eresponse_05_col = eresponse_05,
  esituation_02_col = esituation_02,
  evitals_06_col = evitals_06,
  evitals_14_col = evitals_14,
  evitals_23_col = evitals_23,
  transport_disposition_col = edisposition_30,
  confidence_interval = TRUE
)
```

```
)
```

```
trauma_08_population  Trauma-08 Populations
```

Description

Filters data down to the target populations for Trauma-08, and categorizes records to identify needed information for the calculations.

Identifies key categories to records that are 911 requests for patients with trauma during which GCS, systolic blood pressure, and respiratory rate are documented based on specific criteria and calculates related ECG measures. This function segments the data by age into adult and pediatric populations.

Usage

```
trauma_08_population(
  df = NULL,
  patient_scene_table = NULL,
  response_table = NULL,
  situation_table = NULL,
  disposition_table = NULL,
  vitals_table = NULL,
  erecord_01_col,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col,
  epatient_16_col,
  esituation_02_col,
  eresponse_05_col,
  transport_disposition_col,
  evitals_06_col,
  evitals_14_col,
  evitals_23_col
)
```

Arguments

`df` A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.

`patient_scene_table` A data.frame or tibble containing at least ePatient, and eScene as a fact table.

`response_table` A data.frame or tibble containing at least the eResponse fields needed for this measure's calculations.

situation_table	A data.frame or tibble containing at least the eSituation fields needed for this measure's calculations. Default is NULL.
disposition_table	A data.frame or tibble containing only the edisposition fields needed for this measure's calculations.
vitals_table	A dataframe or tibble containing at least the eVitals fields needed.
erecord_01_col	The column representing the EMS record unique identifier.
incident_date_col	Column that contains the incident date. This defaults to NULL as it is optional in case not available due to PII restrictions.
patient_DOB_col	Column that contains the patient's date of birth. This defaults to NULL as it is optional in case not available due to PII restrictions.
epatient_15_col	Column representing the patient's numeric age agnostic of unit.
epatient_16_col	Column representing the patient's age unit ("Years", "Months", "Days", "Hours", or "Minutes").
esituation_02_col	Column indicating whether or not there was an injury.
eresponse_05_col	Column that contains eResponse.05 or the response type.
transport_disposition_col	One or more unquoted column names (such as edisposition.12, edisposition.30) containing transport disposition for an EMS event identifying whether a transport occurred and by which unit.
evitals_06_col	Numeric column containing systolic blood pressure values.
evitals_14_col	Column containing data on patient's respiratory rate expressed as a number per minute.
evitals_23_col	Column for Glasgow Coma Scale (GCS) scores.

Value

A list that contains the following:

- a tibble with counts for each filtering step,
- a tibble for each population of interest
- a tibble for the initial population
- a tibble for the total dataset with computations
- a tibble with a summary of missingness for each column in each table

Author(s)

Nicolas Foss, Ed.D., MS

Examples

```
# create tables to test correct functioning

# patient table
patient_table <- tibble::tibble(

  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  incident_date = as.Date(c("2025-01-01", "2025-01-05", "2025-02-01",
    "2025-01-01", "2025-06-01")),
  patient_dob = as.Date(c("2000-01-01", "2020-01-01", "2023-02-01",
    "2023-01-01", "1970-06-01")),
  epatient_15 = c(25, 5, 2, 2, 55), # Ages
  epatient_16 = c("Years", "Years", "Years", "Years", "Years")

)

# response table
response_table <- tibble::tibble(

  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  eresponse_05 = rep(2205001, 5)

)

# situation table
situation_table <- tibble::tibble(

  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  esituation_02 = rep("Yes", 5)

)

# vitals table
vitals_table <- tibble::tibble(

  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  evitals_06 = c(100, 90, 80, 70, 85),
  evitals_14 = c(30, 9, 8, 7, 31),
  evitals_23 = c(6, 7, 8, 8, 7),

)

# disposition table
disposition_table <- tibble::tibble(
  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  edisposition_30 = c(4230001, 4230003, 4230001, 4230007, 4230007)
)

# test the success of the function
result <- trauma_08_population(patient_scene_table = patient_table,
  response_table = response_table,
  situation_table = situation_table,
  vitals_table = vitals_table,
  disposition_table = disposition_table,
  erecord_01_col = erecord_01,
```

```

incident_date_col = incident_date,
patient_DOB_col = patient_dob,
epatient_15_col = epatient_15,
epatient_16_col = epatient_16,
eresponse_05_col = eresponse_05,
esituation_02_col = esituation_02,
evitals_06_col = evitals_06,
evitals_14_col = evitals_14,
evitals_23_col = evitals_23,
transport_disposition_col = edisposition_30
)

# show the results of filtering at each step
result$filter_process

```

trauma_14

Trauma-14 Calculation

Description

This function processes EMS data to generate a set of binary variables indicating whether specific trauma triage criteria are met. The output #' is a data frame enriched with these indicators for further analysis. The final outcome is whether or not the EMS record documents the use of #' a pre-hospital trauma activation.

Usage

```

trauma_14(
  df = NULL,
  patient_scene_table = NULL,
  response_table = NULL,
  situation_table = NULL,
  vitals_table = NULL,
  exam_table = NULL,
  procedures_table = NULL,
  injury_table = NULL,
  disposition_table = NULL,
  erecord_01_col,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col,
  epatient_16_col,
  esituation_02_col,
  eresponse_05_col,
  eresponse_10_col,
  transport_disposition_col,
  edisposition_24_col,

```

```

    evitals_06_col,
    evitals_10_col,
    evitals_12_col,
    evitals_14_col,
    evitals_15_col,
    evitals_21_col,
    eexam_16_col,
    eexam_20_col,
    eexam_23_col,
    eexam_25_col,
    eprocedures_03_col,
    einjury_01_col,
    einjury_03_col,
    einjury_04_col,
    einjury_09_col,
    confidence_interval = FALSE,
    method = c("wilson", "clopper-pearson"),
    conf.level = 0.95,
    correct = TRUE,
    ...
)

```

Arguments

df A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.

patient_scene_table A data.frame or tibble containing at least ePatient, and eScene as a fact table.

response_table A data.frame or tibble containing at least the eResponse fields needed for this measure's calculations.

situation_table A data.frame or tibble containing at least the eSituation fields needed for this measure's calculations. Default is NULL.

vitals_table A dataframe or tibble containing at least the eVitals fields needed.

exam_table A data.frame or tibble containing only the eExam fields needed for this measure's calculations. Default is NULL.

procedures_table A dataframe or tibble containing at least the eProcedures fields needed.

injury_table A data frame or tibble containing fields from eInjury needed for this measure's calculations.

disposition_table A data.frame or tibble containing only the edisposition fields needed for this measure's calculations.

erecord_01_col The column representing the EMS record unique identifier.

incident_date_col Column that contains the incident date. This defaults to NULL as it is optional in case not available due to PII restrictions.

patient_DOB_col	Column that contains the patient's date of birth. This defaults to NULL as it is optional in case not available due to PII restrictions.
epatient_15_col	Column representing the patient's numeric age agnostic of unit.
epatient_16_col	Column representing the patient's age unit ("Years", "Months", "Days", "Hours", or "Minutes").
esituation_02_col	Column indicating whether or not there was an injury.
eresponse_05_col	Column that contains eResponse.05 or the response type.
eresponse_10_col	Column name containing informatin about scene delays, if any, of the EMS unit associated with the EMS event.
transport_disposition_col	One or more unquoted column names (such as edisposition.12, edisposition.30) containing transport disposition for an EMS event identifying whether a transport occurred and by which unit.
edisposition_24_col	Column name containing the indication that an alert (or activation) was called by EMS to the appropriate destination healthcare facility team. The alert (or activation) should occur prior to the EMS Unit arrival at the destination with the patient.
evitals_06_col	Numeric column containing systolic blood pressure values.
evitals_10_col	Column name containing the patient's heart rate expressed as a number per minute.
evitals_12_col	Numeric column containing pulse oximetry values.
evitals_14_col	Column containing data on patient's respiratory rate expressed as a number per minute.
evitals_15_col	Column name containing the patient's respiratory effort.
evitals_21_col	Column name containing the patient's Glasgow Coma Score Motor response.
eexam_16_col	Column name containing the assessment findings associated with the patient's extremities.
eexam_20_col	Column name containing the assessment findings of the patient's neurological examination.
eexam_23_col	Column name containing the assessment findings associated with the patient's lungs.
eexam_25_col	Column name containing the assessment findings associated with the patient's chest.
eprocedures_03_col	Column containing procedure codes with or without procedure names.
einjury_01_col	Column name containing the category of the reported/suspected external cause of the injury.

<code>einjury_03_col</code>	Column describing Trauma triage criteria for the red boxes (Injury Patterns and Mental Status and Vital Signs) in the 2021 ACS National Guideline for the Field Triage of Injured Patients.
<code>einjury_04_col</code>	Column name containing Trauma triage criteria for the yellow boxes (Mechanism of Injury and EMS Judgment) in the current ACS National Guideline for the Field Triage of Injured Patients.
<code>einjury_09_col</code>	Column name containing the distance in feet the patient fell, measured from the lowest point of the patient to the ground.
<code>confidence_interval</code>	Logical. If TRUE, the function calculates a confidence interval for the proportion estimate.
<code>method</code>	Character. Specifies the method used to calculate confidence intervals. Options are "wilson" (Wilson score interval) and "clopper-pearson" (exact binomial interval). Partial matching is supported, so "w" and "c" can be used as shorthand.
<code>conf.level</code>	Numeric. The confidence level for the interval, expressed as a proportion (e.g., 0.95 for a 95% confidence interval). Defaults to 0.95.
<code>correct</code>	Logical. If TRUE, applies a continuity correction to the Wilson score interval when <code>method = "wilson"</code> . Defaults to TRUE.
<code>...</code>	optional additional arguments to pass onto <code>dplyr::summarize</code> .

Value

A `data.frame` summarizing results for two population groups (All, Adults and Peds) with the following columns:

- `pop`: Population type (All, Adults, and Peds).
- `numerator`: Count of incidents meeting the measure.
- `denominator`: Total count of included incidents.
- `prop`: Proportion of incidents meeting the measure.
- `prop_label`: Proportion formatted as a percentage with a specified number of decimal places.
- `lower_ci`: Lower bound of the confidence interval for `prop` (if `confidence_interval = TRUE`).
- `upper_ci`: Upper bound of the confidence interval for `prop` (if `confidence_interval = TRUE`).

Author(s)

Nicolas Foss, Ed.D., MS

Examples

```
# Synthetic test data
test_data <- tibble::tibble(
  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  epatient_15 = c(34, 5, 45, 2, 60), # Ages
```

```

epatient_16 = c("Years", "Years", "Years", "Months", "Years"),
eresponse_05 = rep(2205001, 5),
eresponse_10 = rep(2210011, 5),
esituation_02 = rep("Yes", 5),
evitals_06 = c(100, 90, 80, 70, 85),
evitals_10 = c(110, 89, 88, 71, 85),
evitals_12 = c(50, 60, 70, 80, 75),
evitals_14 = c(30, 9, 8, 7, 31),
evitals_15 = c("apneic", "labored", "rapid", "shallow", "weak/agonal"),
evitals_21 = c(5, 4, 3, 2, 1),
eexam_16 = c(3516043, 3516067, 3516043, 3516067, 3516067),
eexam_20 = c(3520045, 3520043, 3520019, 3520017, 3520017),
eexam_23 = c(3523011, 3523003, 3523001, 3523011, 3523003),
eexam_25 = c(3525039, 3525023, 3525005, 3525039, 3525023),
edisposition_24 = c(4224017, 4224003, 4224017, 4224003, 4224017),
edisposition_30 = c(4230001, 4230003, 4230001, 4230007, 4230007),
eprocedures_03 = c(424979004, 427753009, 429705000, 47545007, 243142003),
einjury_01 = c("V20", "V36", "V86", "V39", "V32"),
einjury_03 = c(2903011, 2903009, 2903005, 3903003, 2903001),
einjury_04 = c(2904013, 2904011, 2904009, 2904007, 2904001),
einjury_09 = c(11, 12, 13, 14, 15)
)

# Run function with the first and last pain score columns
# Return 95% confidence intervals using the Wilson method
trauma_14(
  df = test_data,
  erecord_01_col = erecord_01,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col = epatient_15,
  epatient_16_col = epatient_16,
  eresponse_05_col = eresponse_05,
  eresponse_10_col = eresponse_10,
  esituation_02_col = esituation_02,
  evitals_06_col = evitals_06,
  evitals_10_col = evitals_10,
  evitals_12_col = evitals_12,
  evitals_14_col = evitals_14,
  evitals_15_col = evitals_15,
  evitals_21_col = evitals_21,
  eexam_16_col = eexam_16,
  eexam_20_col = eexam_20,
  eexam_23_col = eexam_23,
  eexam_25_col = eexam_25,
  edisposition_24_col = edisposition_24,
  transport_disposition_col = edisposition_30,
  eprocedures_03_col = eprocedures_03,
  einjury_01_col = einjury_01,
  einjury_03_col = einjury_03,
  einjury_04_col = einjury_04,
  einjury_09_col = einjury_09,
  confidence_interval = TRUE
)

```

```
)
```

trauma_14_population *Trauma-14 Population*

Description

This function processes EMS data to generate the population needed to calculate the Trauma-14 NEMSQA measure.

Usage

```
trauma_14_population(  
  df = NULL,  
  patient_scene_table = NULL,  
  situation_table = NULL,  
  response_table = NULL,  
  disposition_table = NULL,  
  vitals_table = NULL,  
  exam_table = NULL,  
  procedures_table = NULL,  
  injury_table = NULL,  
  erecord_01_col,  
  incident_date_col = NULL,  
  patient_DOB_col = NULL,  
  epatient_15_col,  
  epatient_16_col,  
  esituation_02_col,  
  eresponse_05_col,  
  eresponse_10_col,  
  transport_disposition_col,  
  edisposition_24_col,  
  evitals_06_col,  
  evitals_10_col,  
  evitals_12_col,  
  evitals_14_col,  
  evitals_15_col,  
  evitals_21_col,  
  eexam_16_col,  
  eexam_20_col,  
  eexam_23_col,  
  eexam_25_col,  
  eprocedures_03_col,  
  einjury_01_col,  
  einjury_03_col,  
  einjury_04_col,
```

```
    einjury_09_col
  )
```

Arguments

df A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.

patient_scene_table A data.frame or tibble containing at least ePatient, and eScene as a fact table.

situation_table A data.frame or tibble containing at least the eSituation fields needed for this measure's calculations. Default is NULL.

response_table A data.frame or tibble containing at least the eResponse fields needed for this measure's calculations.

disposition_table A data.frame or tibble containing only the edisposition fields needed for this measure's calculations.

vitals_table A dataframe or tibble containing at least the eVitals fields needed.

exam_table A data.frame or tibble containing only the eExam fields needed for this measure's calculations. Default is NULL.

procedures_table A dataframe or tibble containing at least the eProcedures fields needed.

injury_table A data frame or tibble containing fields from eInjury needed for this measure's calculations.

erecord_01_col The column representing the EMS record unique identifier.

incident_date_col Column that contains the incident date. This defaults to NULL as it is optional in case not available due to PII restrictions.

patient_DOB_col Column that contains the patient's date of birth. This defaults to NULL as it is optional in case not available due to PII restrictions.

epatient_15_col Column representing the patient's numeric age agnostic of unit.

epatient_16_col Column representing the patient's age unit ("Years", "Months", "Days", "Hours", or "Minutes").

esituation_02_col Column indicating whether or not there was an injury.

eresponse_05_col Column that contains eResponse.05 or the response type.

eresponse_10_col Column name containing informatin about scene delays, if any, of the EMS unit associated with the EMS event.

transport_disposition_col One or more unquoted column names (such as edisposition.12, edisposition.30) containing transport disposition for an EMS event identifying whether a transport occurred and by which unit.

edisposition_24_col	Column name containing the indication that an alert (or activation) was called by EMS to the appropriate destination healthcare facility team. The alert (or activation) should occur prior to the EMS Unit arrival at the destination with the patient.
evitals_06_col	Numeric column containing systolic blood pressure values.
evitals_10_col	Column name containing the patient's heart rate expressed as a number per minute.
evitals_12_col	Numeric column containing pulse oximetry values.
evitals_14_col	Column containing data on patient's respiratory rate expressed as a number per minute.
evitals_15_col	Column name containing the patient's respiratory effort.
evitals_21_col	Column name containing the patient's Glasgow Coma Score Motor response.
eexam_16_col	Column name containing the assessment findings associated with the patient's extremities.
eexam_20_col	Column name containing the assessment findings of the patient's neurological examination.
eexam_23_col	Column name containing the assessment findings associated with the patient's lungs.
eexam_25_col	Column name containing the assessment findings associated with the patient's chest.
eprocedures_03_col	Column containing procedure codes with or without procedure names.
einjury_01_col	Column name containing the category of the reported/suspected external cause of the injury.
einjury_03_col	Column describing Trauma triage criteria for the red boxes (Injury Patterns and Mental Status and Vital Signs) in the 2021 ACS National Guideline for the Field Triage of Injured Patients.
einjury_04_col	Column name containing Trauma triage criteria for the yellow boxes (Mechanism of Injury and EMS Judgment) in the current ACS National Guideline for the Field Triage of Injured Patients.
einjury_09_col	Column name containing the distance in feet the patient fell, measured from the lowest point of the patient to the ground.

Value

A list that contains the following:

- a tibble with counts for each filtering step,
- a tibble for each population of interest
- a tibble for the initial population
- a tibble for the total dataset with computations
- a tibble with a summary of missingness for each column in each table

Author(s)

Nicolas Foss, Ed.D., MS

Examples

```
# create tables to test correct functioning

# patient table
patient_table <- tibble::tibble(

  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  incident_date = as.Date(c("2025-01-01", "2025-01-05", "2025-02-01",
    "2025-01-01", "2025-06-01")),
  patient_dob = as.Date(c("2000-01-01", "2020-01-01", "2023-02-01",
    "2023-01-01", "1970-06-01")),
  epatient_15 = c(25, 5, 2, 2, 55), # Ages
  epatient_16 = c("Years", "Years", "Years", "Years", "Years")

)

# response table
response_table <- tibble::tibble(

  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  eresponse_05 = rep(2205001, 5),
  eresponse_10 = rep(2210011, 5)

)

# situation table
situation_table <- tibble::tibble(

  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  esituation_02 = rep("Yes", 5),

)

# vitals table
vitals_table <- tibble::tibble(

  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  evitals_06 = c(100, 90, 80, 70, 85),
  evitals_10 = c(110, 89, 88, 71, 85),
  evitals_12 = c(50, 60, 70, 80, 75),
  evitals_14 = c(30, 9, 8, 7, 31),
  evitals_15 = c("apneic", "labored", "rapid", "shallow", "weak/agonal"),
  evitals_21 = c(5, 4, 3, 2, 1)

)

# disposition table
disposition_table <- tibble::tibble(
  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  edisposition_24 = c(4224017, 4224003, 4224017, 4224003, 4224017),
  edisposition_30 = c(4230001, 4230003, 4230001, 4230007, 4230007)
)
```

```
)

# injury table
injury_table <- tibble::tibble(
  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  einjury_01 = c("V20", "V36", "V86", "V39", "V32"),
  einjury_03 = c(2903011, 2903009, 2903005, 3903003, 2903001),
  einjury_04 = c(2904013, 2904011, 2904009, 2904007, 2904001),
  einjury_09 = c(11, 12, 13, 14, 15)
)

# exam table
exam_table <- tibble::tibble(
  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  eexam_16 = c(3516043, 3516067, 3516043, 3516067, 3516067),
  eexam_20 = c(3520045, 3520043, 3520019, 3520017, 3520017),
  eexam_23 = c(3523011, 3523003, 3523001, 3523011, 3523003),
  eexam_25 = c(3525039, 3525023, 3525005, 3525039, 3525023)
)

# procedures table
procedures_table <- tibble::tibble(
  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  eprocedures_03 = c(424979004, 427753009, 429705000, 47545007, 243142003)
)

# test the success of the function
result <- trauma_14_population(patient_scene_table = patient_table,
  response_table = response_table,
  situation_table = situation_table,
  vitals_table = vitals_table,
  disposition_table = disposition_table,
  exam_table = exam_table,
  injury_table = injury_table,
  procedures_table = procedures_table,
  erecord_01_col = erecord_01,
  incident_date_col = incident_date,
  patient_DOB_col = patient_dob,
  epatient_15_col = epatient_15,
  epatient_16_col = epatient_16,
  eresponse_05_col = eresponse_05,
  eresponse_10_col = eresponse_10,
  esituation_02_col = esituation_02,
  evitals_06_col = evitals_06,
  evitals_10_col = evitals_10,
  evitals_12_col = evitals_12,
  evitals_14_col = evitals_14,
  evitals_15_col = evitals_15,
  evitals_21_col = evitals_21,
  eexam_16_col = eexam_16,
  eexam_20_col = eexam_20,
  eexam_23_col = eexam_23,
  eexam_25_col = eexam_25,
```

```

        edisposition_24_col = edisposition_24,
        transport_disposition_col = edisposition_30,
        eprocedures_03_col = eprocedures_03,
        einjury_01_col = einjury_01,
        einjury_03_col = einjury_03,
        einjury_04_col = einjury_04,
        einjury_09_col = einjury_09
    )

# show the results of filtering at each step
result$filter_process

```

ttr_01

TTR-01 Calculation

Description

This function calculates the TTR-01 measure, which evaluates the completeness of vitals documentation for patients not experiencing cardiac arrest who were also not transported during a 911 response. It determines the total population, adult population, and pediatric population meeting the criteria for the TTR_01 measure.

Usage

```

ttr_01(
  df = NULL,
  patient_scene_table = NULL,
  response_table = NULL,
  disposition_table = NULL,
  vitals_table = NULL,
  arrest_table = NULL,
  erecord_01_col,
  incident_date_col = NULL,
  patient_DOB_col = NULL,
  epatient_15_col,
  epatient_16_col,
  eresponse_05_col,
  transport_disposition_col,
  earrest_01_col,
  evitals_06_col,
  evitals_07_col,
  evitals_10_col,
  evitals_12_col,
  evitals_14_col,
  evitals_23_col,
  evitals_26_col,
  confidence_interval = FALSE,

```

```

method = c("wilson", "clopper-pearson"),
conf.level = 0.95,
correct = TRUE,
...
)

```

Arguments

<code>df</code>	A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.
<code>patient_scene_table</code>	A data.frame or tibble containing at least ePatient, and eScene as a fact table.
<code>response_table</code>	A data.frame or tibble containing at least the eResponse fields needed for this measure's calculations.
<code>disposition_table</code>	A data.frame or tibble containing only the edisposition fields needed for this measure's calculations.
<code>vitals_table</code>	A dataframe or tibble containing at least the eVitals fields needed.
<code>arrest_table</code>	A data.frame or tibble containing at least the eArrest fields needed for this measure's calculations.
<code>erecord_01_col</code>	The column representing the EMS record unique identifier.
<code>incident_date_col</code>	Column that contains the incident date. This defaults to NULL as it is optional in case not available due to PII restrictions.
<code>patient_DOB_col</code>	Column that contains the patient's date of birth. This defaults to NULL as it is optional in case not available due to PII restrictions.
<code>epatient_15_col</code>	Column representing the patient's numeric age agnostic of unit.
<code>epatient_16_col</code>	Column representing the patient's age unit ("Years", "Months", "Days", "Hours", or "Minutes").
<code>eresponse_05_col</code>	Column that contains eResponse.05 or the response type.
<code>transport_disposition_col</code>	One or more unquoted column names (such as edisposition.12, edisposition.30) containing transport disposition for an EMS event identifying whether a transport occurred and by which unit.
<code>earrest_01_col</code>	Column representing whether or not the patient is in arrest.
<code>evitals_06_col</code>	Numeric column containing systolic blood pressure values.
<code>evitals_07_col</code>	A column containing the patient's diastolic blood pressure.
<code>evitals_10_col</code>	Column name containing the patient's heart rate expressed as a number per minute.
<code>evitals_12_col</code>	Numeric column containing pulse oximetry values.

evitals_14_col	Column name containing the patient's respiratory rate expressed as a number per minute.
evitals_23_col	Column for Glasgow Coma Scale (GCS) scores.
evitals_26_col	Column for AVPU alertness levels.
confidence_interval	Logical. If TRUE, the function calculates a confidence interval for the proportion estimate.
method	Character. Specifies the method used to calculate confidence intervals. Options are "wilson" (Wilson score interval) and "clopper-pearson" (exact binomial interval). Partial matching is supported, so "w" and "c" can be used as shorthand.
conf.level	Numeric. The confidence level for the interval, expressed as a proportion (e.g., 0.95 for a 95% confidence interval). Defaults to 0.95.
correct	Logical. If TRUE, applies a continuity correction to the Wilson score interval when method = "wilson". Defaults to TRUE.
...	optional additional arguments to pass onto <code>dplyr::summarize</code> .

Value

A data.frame summarizing results for two population groups (All, Adults and Peds) with the following columns:

- pop: Population type (All, Adults, and Peds).
- numerator: Count of incidents meeting the measure.
- denominator: Total count of included incidents.
- prop: Proportion of incidents meeting the measure.
- prop_label: Proportion formatted as a percentage with a specified number of decimal places.
- lower_ci: Lower bound of the confidence interval for prop (if confidence_interval = TRUE).
- upper_ci: Upper bound of the confidence interval for prop (if confidence_interval = TRUE).

Author(s)

Nicolas Foss, Ed.D., MS

Examples

```
# Synthetic test data
test_data <- tibble::tibble(
  erecoord_01 = c("R1", "R2", "R3", "R4", "R5"),
  incident_date = as.Date(c("2025-01-01", "2025-01-05", "2025-02-01",
    "2025-01-01", "2025-06-01")),
  patient_dob = as.Date(c("2000-01-01", "2020-01-01", "2023-02-01",
    "2023-01-01", "1970-06-01")),
  epatient_15 = c(34, 5, 45, 2, 60), # Ages
```

```

epatient_16 = c("Years", "Years", "Years", "Months", "Years"),
eresponse_05 = rep(2205001, 5),
earrest_01 = rep("No", 5),
evitals_06 = c(100, 90, 80, 70, 85),
evitals_07 = c(80, 90, 50, 60, 87),
evitals_10 = c(110, 89, 88, 71, 85),
evitals_12 = c(50, 60, 70, 80, 75),
evitals_14 = c(30, 9, 8, 7, 31),
evitals_23 = c(6, 7, 8, 9, 10),
evitals_26 = c(3326007, 3326005, 3326003, 3326001, 3326007),
edisposition_30 = c(4230013, 4230009, 4230013, 4230009, 4230013)
)

# Run function with the first and last pain score columns
# Return 95% confidence intervals using the Wilson method
ttr_01(
  df = test_data,
  erecord_01_col = erecord_01,
  incident_date_col = incident_date,
  patient_DOB_col = patient_dob,
  epatient_15_col = epatient_15,
  epatient_16_col = epatient_16,
  eresponse_05_col = eresponse_05,
  earrest_01_col = earrest_01,
  evitals_06_col = evitals_06,
  evitals_07_col = evitals_07,
  evitals_10_col = evitals_10,
  evitals_12_col = evitals_12,
  evitals_14_col = evitals_14,
  evitals_23_col = evitals_23,
  evitals_26_col = evitals_26,
  transport_disposition_col = edisposition_30
)

```

ttr_01_population	<i>TTR-01 Populations</i>
-------------------	---------------------------

Description

Filters data down to the target populations for TTR-01, and categorizes records to identify needed information for the calculations.

Identifies key categories to records that are 911 requests for patients not transported by EMS during which a basic set of vital signs is documented based on specific criteria and calculates related ECG measures. This function segments the data by age into adult and pediatric populations.

Usage

```
ttr_01_population(
```

```

df = NULL,
patient_scene_table = NULL,
response_table = NULL,
disposition_table = NULL,
vitals_table = NULL,
arrest_table = NULL,
erecord_01_col,
incident_date_col = NULL,
patient_DOB_col = NULL,
epatient_15_col,
epatient_16_col,
eresponse_05_col,
transport_disposition_col,
earrest_01_col,
evitals_06_col,
evitals_07_col,
evitals_10_col,
evitals_12_col,
evitals_14_col,
evitals_23_col,
evitals_26_col
)

```

Arguments

df A dataframe or tibble containing EMS data where each row represents an observation and columns represent features.

patient_scene_table A data.frame or tibble containing at least ePatient, and eScene as a fact table.

response_table A data.frame or tibble containing at least the eResponse fields needed for this measure's calculations.

disposition_table A data.frame or tibble containing only the edisposition fields needed for this measure's calculations.

vitals_table A dataframe or tibble containing at least the eVitals fields needed.

arrest_table A data.frame or tibble containing at least the eArrest fields needed for this measure's calculations.

erecord_01_col The column representing the EMS record unique identifier.

incident_date_col Column that contains the incident date. This defaults to NULL as it is optional in case not available due to PII restrictions.

patient_DOB_col Column that contains the patient's date of birth. This defaults to NULL as it is optional in case not available due to PII restrictions.

epatient_15_col Column representing the patient's numeric age agnostic of unit.

epatient_16_col	Column representing the patient's age unit ("Years", "Months", "Days", "Hours", or "Minutes").
eresponse_05_col	Column that contains eResponse.05 or the response type.
transport_disposition_col	One or more unquoted column names (such as edisposition.12, edisposition.30) containing transport disposition for an EMS event identifying whether a transport occurred and by which unit.
earrest_01_col	Column representing whether or not the patient is in arrest.
evitals_06_col	Numeric column containing systolic blood pressure values.
evitals_07_col	A column containing the patient's diastolic blood pressure.
evitals_10_col	Column name containing the patient's heart rate expressed as a number per minute.
evitals_12_col	Numeric column containing pulse oximetry values.
evitals_14_col	Column name containing the patient's respiratory rate expressed as a number per minute.
evitals_23_col	Column for Glasgow Coma Scale (GCS) scores.
evitals_26_col	Column for AVPU alertness levels.

Value

A list that contains the following:

- a tibble with counts for each filtering step,
- a tibble for each population of interest
- a tibble for the initial population
- a tibble for the total dataset with computations
- a tibble with a summary of missingness for each column in each table

Author(s)

Nicolas Foss, Ed.D., MS

Examples

```
# create tables to test correct functioning

# patient table
patient_table <- tibble::tibble(

  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  incident_date = as.Date(c("2025-01-01", "2025-01-05", "2025-02-01",
    "2025-01-01", "2025-06-01")),
  patient_dob = as.Date(c("2000-01-01", "2020-01-01", "2023-02-01",
    "2023-01-01", "1970-06-01")),
```

```

epatient_15 = c(25, 5, 2, 2, 55), # Ages
epatient_16 = c("Years", "Years", "Years", "Years", "Years")

)

# response table
response_table <- tibble::tibble(

  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  eresponse_05 = rep(2205001, 5),
)

# arrest table
arrest_table <- tibble::tibble(

  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  earrest_01 = rep("No", 5)
)

# vitals table
vitals_table <- tibble::tibble(

  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  evitals_06 = c(100, 90, 80, 70, 85),
  evitals_07 = c(80, 90, 50, 60, 87),
  evitals_10 = c(110, 89, 88, 71, 85),
  evitals_12 = c(50, 60, 70, 80, 75),
  evitals_14 = c(30, 9, 8, 7, 31),
  evitals_23 = c(6, 7, 8, 9, 10),
  evitals_26 = c(3326007, 3326005, 3326003, 3326001, 3326007),
)

# disposition table
disposition_table <- tibble::tibble(
  erecord_01 = c("R1", "R2", "R3", "R4", "R5"),
  edisposition_30 = c(4230013, 4230009, 4230013, 4230009, 4230013)
)

# test the success of the function
result <- ttr_01_population(patient_scene_table = patient_table,
  response_table = response_table,
  arrest_table = arrest_table,
  vitals_table = vitals_table,
  disposition_table = disposition_table,
  erecord_01_col = erecord_01,
  incident_date_col = incident_date,
  patient_DOB_col = patient_dob,
  epatient_15_col = epatient_15,
  epatient_16_col = epatient_16,
  eresponse_05_col = eresponse_05,
  earrest_01_col = earrest_01,
  evitals_06_col = evitals_06,
  evitals_07_col = evitals_07,

```

```
        evitals_10_col = evitals_10,  
        evitals_12_col = evitals_12,  
        evitals_14_col = evitals_14,  
        evitals_23_col = evitals_23,  
        evitals_26_col = evitals_26,  
        transport_disposition_col = edisposition_30  
    )  
  
# show the results of filtering at each step  
result$filter_process
```

Index

* datasets

- nemsqar_airway_table, 39
 - nemsqar_arrest_table, 40
 - nemsqar_disposition_table, 42
 - nemsqar_exam_table, 43
 - nemsqar_injury_table, 44
 - nemsqar_medications_table, 45
 - nemsqar_patient_scene_table, 46
 - nemsqar_procedures_table, 47
 - nemsqar_response_table, 48
 - nemsqar_situation_table, 49
 - nemsqar_vitals_table, 51
- airway_01, 3
- airway_01_population, 7
- airway_05, 11
- airway_05_population, 15
- airway_18, 18
- airway_18_population, 23
- asthma_01, 26
- asthma_01_population, 29
- hypoglycemia_01, 32
- hypoglycemia_01_population, 35
- nemsqa_binomial_confint, 52
- nemsqar_airway_table, 39
- nemsqar_arrest_table, 40
- nemsqar_disposition_table, 42
- nemsqar_exam_table, 43
- nemsqar_injury_table, 44
- nemsqar_medications_table, 45
- nemsqar_patient_scene_table, 46
- nemsqar_procedures_table, 47
- nemsqar_response_table, 48
- nemsqar_situation_table, 49
- nemsqar_vitals_table, 51
- pediatrics_03b, 53
- pediatrics_03b_population, 56
- pretty_percent, 59
- respiratory_01, 60
- respiratory_01_population, 63
- respiratory_02, 66
- respiratory_02_population, 69
- safety_01, 72
- safety_01_population, 74
- safety_02, 77
- safety_02_population, 79
- safety_04, 82
- safety_04_population, 85
- seizure_02, 89
- seizure_02_population, 92
- stroke_01, 95
- stroke_01_population, 97
- syncope_01, 100
- syncope_01_population, 103
- tbi_01, 106
- tbi_01_population, 110
- trauma_01, 113
- trauma_01_population, 116
- trauma_03, 120
- trauma_03_population, 123
- trauma_04, 127
- trauma_04_population, 132
- trauma_08, 138
- trauma_08_population, 141
- trauma_14, 144
- trauma_14_population, 149
- ttr_01, 154
- ttr_01_population, 157