

# Package ‘pii’

May 9, 2026

**Title** Search Data Frames for Personally Identifiable Information

**Version** 1.3.0

**Maintainer** Jacob Patterson-Stein <jacobpstein@gmail.com>

## Description

Check a data frame for personal information, including names, location, disability status, and geo-coordinates.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Depends** R (>= 2.10), dplyr, stringr, uuid, utils

**RoxygenNote** 7.3.2

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**URL** <https://github.com/jacobpstein/pii>

**BugReports** <https://github.com/jacobpstein/pii/issues>

**NeedsCompilation** no

**Author** Jacob Patterson-Stein [aut, cre]

**Repository** CRAN

**Date/Publication** 2025-01-13 15:40:06 UTC

## Contents

check_PII . . . . .	2
split_PII_data . . . . .	2

<b>Index</b>	<b>4</b>
--------------	----------

---

`check_PII`*Search Data Frames for Personally Identifiable Information*

---

**Description**

Search Data Frames for Personally Identifiable Information

**Usage**

```
check_PII(df)
```

**Arguments**

`df` a data frame object

**Value**

Returns a data frame of columns that potentially contain PII

**Examples**

```
# create a data frame containing various personally identifiable information
pii_df <- data.frame(
  lat = c(40.7128, 34.0522, 41.8781),
  long = c(-74.0060, -118.2437, -87.6298),
  first_name = c("John", "Michael", "Linda"),
  phone = c("123-456-7890", "234-567-8901", "345-678-9012"),
  age = sample(30:60, 3, replace = TRUE),
  email = c("test@example.com", "contact@domain.com", "user@website.org"),
  disabled = c("No", "Yes", "No"),
  stringsAsFactors = FALSE
)

check_PII(pii_df)
```

---

`split_PII_data`*Split Data Into PII and Non-PII Columns*

---

**Description**

Split Data Into PII and Non-PII Columns

**Usage**

```
split_PII_data(df, exclude_columns = NULL)
```

**Arguments**

`df` a data frame object  
`exclude_columns` columns to exclude from the data frame

**Value**

Returns two data frames into the global environment: one containing the PII columns and one without the PII columns. A unique merge key is created to join them. The function then prints the columns that were flagged and split to the console.

**Examples**

```
# create a data frame containing various personally identifiable information
pii_df <- data.frame(
  lat = c(40.7128, 34.0522, 41.8781),
  long = c(-74.0060, -118.2437, -87.6298),
  first_name = c("John", "Michael", "Linda"),
  phone = c("123-456-7890", "234-567-8901", "345-678-9012"),
  age = sample(30:60, 3, replace = TRUE),
  email = c("test@example.com", "contact@domain.com", "user@website.org"),
  disabled = c("No", "Yes", "No"),
  stringsAsFactors = FALSE
)

split_PII_data(pii_df, exclude_columns = c("phone"))
```

# Index

`check_PII`, [2](#)

`split_PII_data`, [2](#)