

# Package ‘mmcards’

May 9, 2026

**Title** Playing Cards Utility Functions

**Version** 0.1.1

**Description** Early insights in probability theory were largely influenced by questions about gambling and games of chance, as noted by Blitzstein and Hwang (2019, ISBN:978-1138369917). In modern times, playing cards continue to serve as an effective teaching tool for probability, statistics, and even 'R' programming, as demonstrated by Golemund (2014, ISBN:978-1449359010). The 'mmcards' package offers a collection of utility functions designed to aid in the creation, manipulation, and utilization of playing card decks in multiple formats. These include a standard 52-card deck, as well as alternative decks such as decks defined by custom anonymous functions and custom interleaved decks. Optimized for the development of educational 'shiny' applications, the package is particularly useful for teaching statistics and probability through card-based games. Functions include `shuffle_deck()`, which creates either a shuffled standard deck or a shuffled custom alternative deck; `deal_card()`, which takes a deck and returns a list object containing both the dealt card and the updated deck; and `i_deck()`, which adds image paths to card objects, further enriching the package's utility in the development of interactive 'shiny' application card games.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**URL** <https://github.com/mightymetrika/mmcards>

**BugReports** <https://github.com/mightymetrika/mmcards/issues>

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

**Config/testthat/edition** 3

**Depends** R (>= 2.10)

**LazyData** true

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Mackson Ncube [aut, cre],  
mightymetrika, LLC [cph, fnd]

**Maintainer** Mackson Ncube <macksonncube.stats@gmail.com>

**Repository** CRAN

**Date/Publication** 2023-09-06 16:50:02 UTC

## Contents

deal_card . . . . .	2
i_deck . . . . .	3
shuffle_deck . . . . .	4
standard_deck . . . . .	5
vector_playing_cards . . . . .	6

<b>Index</b>	<b>7</b>
--------------	----------

---

deal_card	<i>Deal a Card from the Deck</i>
-----------	----------------------------------

---

## Description

This function deals the top card from a given deck and returns the dealt card along with the updated deck.

## Usage

```
deal_card(current_deck)
```

## Arguments

`current_deck` A data frame representing the current deck of cards. This can either be a standard deck, an anonymous deck, or an interleaved deck. The function also accepts an object of class "UpDeck" which contains an updated deck and the last dealt card.

## Value

A list containing two elements: `dealt_card`, a data frame representing the card that was dealt, and `updated_deck`, a data frame representing the remaining cards in the deck. The list has the class attribute "UpDeck".

## Examples

```
# Using a standard deck
std_deck <- standard_deck()
result <- deal_card(std_deck)
result$dealt_card
result$updated_deck

# Using an "UpDeck" object
result2 <- deal_card(result)
```

```
result2$dealt_card
result2$updated_deck
```

---

i\_deck

---

*Add image paths to a deck of cards*


---

## Description

This function takes a deck of cards and adds image paths to each card. It produces a new deck that includes the original information along with the image paths. The function is designed to work with various types of decks and allows for customization of image paths.

## Usage

```
i_deck(
  deck,
  i_path,
  cards = c("2C", "2D", "2H", "2S", "3C", "3D", "3H", "3S", "4C", "4D", "4H", "4S", "5C",
    "5D", "5H", "5S", "6C", "6D", "6H", "6S", "7C", "7D", "7H", "7S", "8C", "8D", "8H",
    "8S", "9C", "9D", "9H", "9S", "10C", "10D", "10H", "10S", "JC", "JD", "JH", "JS",
    "QC", "QD", "QH", "QS", "KC", "KD", "KH", "KS", "AC", "AD", "AH", "AS"),
  i_names = vector_playing_cards,
  i_type = "png"
)
```

## Arguments

deck	A data frame representing a deck of cards. Default is <code>standard_deck()</code> , but the function is also optimized to work with <code>shuffled_deck()</code> .
i_path	The file path to the folder containing the card images.
cards	A vector of card names corresponding to the images. Default includes all cards based on a deck created with <code>standard_deck()</code>
i_names	A vector of image file names corresponding to the cards. Should be in the same order as <code>cards</code> the default is <code>vector_playing_cards</code> which works when <code>deck</code> is set to <code>standard_deck()</code> and <code>cards</code> uses the default vector.
i_type	The file extension of the image files (e.g., "png", "jpg").

## Value

A data frame that includes the original deck along with image paths for each card. Inherits the class "ImgDeck" in addition to the original deck's classes.

## Examples

```
#The following example shows how to add image paths to the 'mmcards' default
#StandardDeck when the images referenced in ?vector_playing_cards are stored
#in the working directory.
```

```
image_deck <- i_deck(deck = standard_deck(),
                    i_path = getwd())
```

```
head(image_deck)
```

```
#See the README for an example of how to use i_deck to incorporate images
#into a 'shiny' application.
```

---

shuffle\_deck

*Shuffle a Deck of Cards*

---

## Description

This function shuffles a deck of cards and returns the shuffled deck. The function can handle standard decks, anonymous decks, and interleaved decks. For interleaved decks, an option to pair shuffle is also available.

## Usage

```
shuffle_deck(
  deck_of_cards = function(x) {
    standard_deck()
  },
  seed = NULL,
  paired = FALSE
)
```

## Arguments

deck_of_cards	An anonymous function that returns a deck of cards as either a data frame or a list of two numeric vectors for interleaved decks. Default is <code>function(x){standard_deck()}</code> .
seed	An optional seed for reproducibility. Default is <code>NULL</code> .
paired	Logical flag to indicate if the interleaved deck should be pair shuffled. Default is <code>FALSE</code> .

## Value

A data frame representing the shuffled deck of cards. The data frame inherits various classes based on its type. All shuffled decks will have the classes "ShuffledDeck" and "data.frame". Additional class inheritance depends on the `deck_of_cards` parameter:

- "StandardDeck" if `deck_of_cards` returns a standard deck (default)

- "AnonymousDeck" if deck\_of\_cards returns a single vector
- "InterleavedDeck" if deck\_of\_cards returns a list of two vectors. If the paired parameter is set to TRUE, an interleaved deck will also inherit the class "PairedDeck".

## Examples

```
# Standard deck
std_deck <- shuffle_deck()
head(std_deck)

# Anonymous deck
anon_deck <- shuffle_deck(deck_of_cards = function(x){runif(52, 1, 10)})
head(anon_deck)

# Interleaved deck
interleaved_deck <- shuffle_deck(
  deck_of_cards = function(x){list(runif(26, 1, 5),
                                   runif(26, 6, 10))})

head(interleaved_deck)

# Paired interleaved deck
paired_deck <- shuffle_deck(
  deck_of_cards = function(x){list(runif(26, 1, 5),
                                   runif(26, 6, 10))},
  paired = TRUE)

head(paired_deck)
```

---

standard\_deck

*Generate a Standard Deck of Playing Cards*

---

## Description

This function creates a standard deck of playing cards represented as a data frame. The deck includes suits, ranks, and values for each card.

## Usage

```
standard_deck(
  suits = c("C", "D", "H", "S"),
  ranks = c("2", "3", "4", "5", "6", "7", "8", "9", "10", "J", "Q", "K", "A"),
  values = seq(2, 14.75, by = 0.25)
)
```

## Arguments

**suits** A character vector specifying the suits in the deck. Default is `c('C', 'D', 'H', 'S')` for Clubs, Diamonds, Hearts, and Spades.

ranks	A character vector specifying the ranks in the deck. Default is <code>c('2', '3', '4', '5', '6', '7', '8', '9', '10', 'J', 'Q', 'K', 'A')</code> for ranks 2 to Ace.
values	A numeric vector specifying the values assigned to each card in the deck. Default is a sequence from 2 to 14.75 incremented by 0.25.

**Value**

A data frame representing the deck of cards. The data frame has four columns: rank, suit, card, and value. The data frame also has class attributes "StandardDeck" and "data.frame".

**Examples**

```
deck <- standard_deck()
head(deck)
tail(deck)
```

---

vector\_playing\_cards *Vector Playing Cards Image Names*

---

**Description**

Image names for vector playing cards

**Usage**

```
vector_playing_cards
```

**Format**

```
vector_playing_cards:
A vector of length 52
vector_playing_cards Names of cards for a 52 card deck
```

**Source**

<https://code.google.com/archive/p/vector-playing-cards/>

# Index

## \* datasets

vector\_playing\_cards, [6](#)

deal\_card, [2](#)

i\_deck, [3](#)

shuffle\_deck, [4](#)

standard\_deck, [5](#)

vector\_playing\_cards, [6](#)