

# Package: MSMU (via r-universe)

May 28, 2026

**Title** Descriptive Statistics Functions for Numeric Data

**Version** 0.1.2

**Description** Provides fundamental functions for descriptive statistics, including `MODE()`, `estimate_mode()`, `center_stats()`, `position_stats()`, `pct()`, `spread_stats()`, `kurt()`, `skew()`, and `shape_stats()`, which assist in summarizing the center, spread, and shape of numeric data. For more details, see McCurdy (2025), "Introduction to Data Science with R" <<https://jonmccurdy.github.io/Introduction-to-Data-Science/>>.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**Depends** R (>= 3.5)

**LazyData** true

**Suggests** roxygen2

**NeedsCompilation** no

**Author** Luke Papayoanou [aut], Jon McCurdy [aut, cre]

**Maintainer** Jon McCurdy <j.r.mccurdy@msmary.edu>

**Repository** <https://cran.r-universe.dev>

**Date/Publication** 2025-07-22 11:01:57 UTC

**RemoteUrl** <https://github.com/cran/MSMU>

**RemoteRef** HEAD

**RemoteSha** 08733e725092d67d40f1930b30bf1378ea891ad6

## Contents

baseball_teams	2
basketball	3
center_stats	4
christmas	5

class_demographics . . . . .	6
college_data . . . . .	6
county_data . . . . .	7
course_scores . . . . .	8
data_210_census . . . . .	9
election_2020 . . . . .	10
estimate_mode . . . . .	10
exam_data . . . . .	11
football . . . . .	12
heart . . . . .	13
housing_data . . . . .	14
income_data . . . . .	14
kurt . . . . .	15
ledger_data . . . . .	16
mlb_eda . . . . .	19
MODE . . . . .	20
mount_dorms . . . . .	21
MSMU . . . . .	21
pct . . . . .	23
position_stats . . . . .	23
reaction_time . . . . .	24
shape_stats . . . . .	25
skew . . . . .	26
soccer . . . . .	26
spread_stats . . . . .	27
<b>Index</b>	<b>28</b>

---

baseball_teams	<i>Professional baseball teams data</i>
----------------	---

---

### Description

This dataset contains historical performance and statistics for professional baseball teams across multiple seasons from 2000-2020.

### Usage

```
baseball_teams
```

### Format

A data frame with 630 rows and 12 columns:

**year** Year (integer)  
**team\_name** Team (character)  
**games\_played** Number of games played (integer)

**wins** Number of wins (integer)  
**losses** Number of losses (integer)  
**world\_series** World series winner that specific year (character)  
**runs\_scored** Number of total runs scored during season (integer)  
**hits** Number of total hits during season (integer)  
**homeruns** Number of total homeruns during season (integer)  
**earned\_run\_average** Team earned run average per 9 innings (numeric)  
**fielding\_percentage** Team fielding percentage (numeric)  
**home\_attendance** Average home game attendance (integer)

### Source

Data retrieved from [Lahmans Baseball Database](#) with alterations made for educational purposes

---

basketball	<i>College basketball data</i>
------------	--------------------------------

---

### Description

This dataset contains performance statistics for 363 men's college basketball teams from the 2022-23 season.

### Usage

basketball

### Format

A data frame with 363 rows and 18 columns:

**School** School (character)  
**State** State (character)  
**W** Wins (integer)  
**L** Loss's (integer)  
**W.L.** Win Loss percentage (numeric)  
**SRS** Simple Rating System (numeric)  
**SOS** Strength of Schedule (numeric)  
**Points.Scored** Points scored (integer)  
**Points.Allowed** Points allowed (integer)  
**FG.** Team field goal percentage (numeric)  
**X3P.** Three point percentage (numeric)  
**FT.** Free throw percentage (numeric)

**Rebounds** Number of rebounds (integer)  
**AST** Number of assists (integer)  
**STL** Number of steals (integer)  
**Blocks** Number of blocks (integer)  
**Turn.Overs** Number of turn overs (integer)  
**Fouls** Number of fouls (integer)

### Source

Data retrieved from [Sports Reference](#) with alterations made for educational purposes.

---

center_stats	<i>Summary of Central Tendency</i>
--------------	------------------------------------

---

### Description

Computes a variety of center statistics for a numeric vector, including: mean, median, trimmed means (10% and 25%), and estimated mode (via probability density function using `estimate_mode()`).

### Usage

```
center_stats(x)
```

### Arguments

`x` A numeric vector.

### Value

A named numeric vector with values for:

**mean** Arithmetic mean  
**median** Median  
**trim25** 25% trimmed mean  
**trim10** 10% trimmed mean  
**est\_mode** Estimated mode from `estimate_mode()`

### See Also

[estimate\\_mode](#)

**Examples**

```
# Center Stats of continuous random data
set.seed(123)
x <- rnorm(1000, mean=50, sd=10)
center_stats(x)

# Center Stats of Sepal Length in iris data set
data("iris")
center_stats(iris$Sepal.Length)
```

---

 christmas

*Christmas data*


---

**Description**

Santa's dataset, exploring if Santa gives children presents based a variety of variables!

**Usage**

```
christmas
```

**Format**

A data frame with 1000 rows and 15 columns:

**Gender** Gender (character)

**Toy\_Count** Number of toys (integer)

**Chores\_Completed** Number of Chores completed (numeric)

**Favorite\_Color** Childs Favorite color (character)

**Helping\_Hand** Childs helping hand number/score (integer)

**Complaints\_Received** Number of complaints child says (numeric)

**Tantrum\_Count** Number of Tantrums child has (integer)

**Rule\_Breaks** Number of rule breaking child does (numeric)

**Sharing\_Behavior** Childs willingness to share (numeric)

**Hours\_of\_Sleep** Childs average hours of sleep per night (numeric)

**Screen\_Time** Childs average hours of screen time (numeric)

**School\_Grade** Childs school grade (numeric)

**Parent\_Presence** Childs parent presence (numeric)

**Greed\_Score** Santas numeric system for labeling childrens greed (numeric)

**Outcome** Whether a child gets a present or coal (character)

**Source**

Santa

---

class_demographics	<i>Class demographics</i>
--------------------	---------------------------

---

**Description**

A sample dataset representing demographic and academic information for 50 college students.

**Usage**

class\_demographics

**Format**

A data frame with 50 rows and 6 columns:

**names** Persons name (character)

**ages** Persons age (int)

**state** Persons state (character)

**year** Persons year in college (character)

**majors** Persons major (character)

**sport** Binary Sport, 1(yes) or 0(no) (integer)

**Source**

Synthetic Data

---

college_data	<i>College data</i>
--------------	---------------------

---

**Description**

This dataset provides detailed information on 777 U.S. colleges and universities from 1995, covering aspects of admissions, academics, finances, and student demographics.

**Usage**

college\_data

**Format**

A data frame with 777 rows and 16 columns:

**Name** College name (character)

**Region** US region (character)

**Accept** Acceptance (integer)

**Enroll** Enrollment (integer)

**Top10perc** Percent of students that were top 10 in highschool class (integer)

**Top25perc** Percent of students that were top 25 in highschool class (integer)

**F.Undergrad** Full time undergrad (integer)

**P.Undergrad** Part time undergrad (integer)

**Outstate** Number of Out of state students (integer)

**Room.Board** Annual room and board price (integer)

**PhD** Percentage of Faculty with a PhD (integer)

**Terminal** Percentage of Faculty with a terminal degree (integer)

**S.F.Ratio** Student Faculty ratio (numeric)

**perc.alumni** Percent of alumni who donate to the college (integer)

**Expend** Instructional expenditure per student (integer)

**Grad.Rate** Graduation Rate (integer)

**Source**

This dataset was taken from the [StatLib](#) library which is maintained at Carnegie Mellon University. Adapted from the College data set in the ISLR library with alterations made for educational purposes.

---

county\_data

*County data*

---

**Description**

Data for 3142 counties in the United States containing demographic, educational, economic, and technological statistics.

**Usage**

county\_data

**Format**

A data frame with 3142 rows and 17 columns:

**state** State (character)

**name** County name (character)

**fips** County level FIPS code (integer)

**pop** County population (integer)

**households** Number of households (integer)

**median\_age** Median age of people in county (numeric)

**age\_over\_18** Percent age of people over 18 (numeric)

**age\_over\_65** Percent age of people over 65 (numeric)

**hs\_grad** Percent of highschool grads (numeric)

**bachelors** Percent of people with bachelors degrees (numeric)

**white** Percent of population that is white (numeric)

**black** Percent of population that is black (numeric)

**hispanic** Percent of population that is hispanic (numeric)

**household\_has\_smartphone** Percent of households who have a smartphone (numeric)

**mean\_household\_income** Average household income (integer)

**median\_household\_income** Median household income (integer)

**unemployment\_rate** Unemployment rate (numeric)

**Source**

Adapted from the county\_complete data set in the [usdata library](#) with alterations made for educational purposes.

---

course\_scores

*Course scores data*

---

**Description**

This dataset contains academic performance records for 200 students across four years of high school, with scores or letter grades in English and Math.

**Usage**

course\_scores

**Format**

A data frame with 200 rows and 10 columns:

**student** Student ID (integer)

**type** Grade type (character)

**Freshman\_English** Freshman English Score/letter grade (character)

**Freshman\_Math** Freshman Math Score/letter grade (character)

**Sophomore\_English** Sophomore English Score/letter grade (character)

**Sophomore\_Math** Sophomore Math Score/letter grade (character)

**Junior\_English** Junior English Score/letter grade (character)

**Junior\_Math** Junior Math Score/letter grade (character)

**Senior\_English** Senior English Score/letter grade (character)

**Senior\_Math** Senior Math Score/letter grade (character)

**Source**

Synthetic Data

---

data_210_census	<i>Synthetic Census dataset</i>
-----------------	---------------------------------

---

**Description**

A synthetic dataset containing demographic and socioeconomic information for 1,000 individuals.

**Usage**

data\_210\_census

**Format**

A data frame with 1000 rows and 5 columns:

**age** Persons Age (integer)

**gender** Persons Gender (character)

**degree** Persons level of education (character)

**salary** Persons Yearly Salary (integer)

**height** Persons Height in inches (integer)

**Source**

Synthetic Data

---

election_2020	<i>2020 election data</i>
---------------	---------------------------

---

**Description**

Dataset providing detailed results from the 2020 U.S. presidential election at the county level.

**Usage**

```
election_2020
```

**Format**

A data frame with 32177 rows and 7 columns:

**state** State (character)

**state\_ev** State electoral votes (integer)

**county** County name (character)

**candidate** Candidate name (character)

**party** Candidate party (character)

**total\_votes** Total number of votes (integer)

**won** True or false for the candidate to win the county (logical)

**Source**

Data retrieved from [MIT Election Data and Science Lab](#), 2018, "County Presidential Election Returns 2000-2020" with alterations made for educational purposes.

---

estimate_mode	<i>Estimate Mode using Density function to find Mode of continuous data</i>
---------------	---

---

**Description**

Estimates the mode of a numeric vector by identifying the value corresponding to the peak of its estimated probability density function.

**Usage**

```
estimate_mode(x)
```

**Arguments**

x                    A numeric vector. Missing values (NA) are removed.

**Value**

A single numeric value representing the estimated mode.

**Examples**

```
# Estimate the mode of continuous random data
set.seed(123)
x <- rnorm(1000, mean=5, sd=2)
estimate_mode(x)

# Estimate the mode of miles-per-gallon (mpg) in the mtcars dataset
data("mtcars")
estimate_mode(mtcars$mpg)
```

---

exam\_data

*Exam data*

---

**Description**

Synthetic dataset containing academic performance and background information for 1,000 students.

**Usage**

```
exam_data
```

**Format**

A data frame with 1000 rows and 8 columns:

**gender** Students gender (character)

**race.ethnicity** Students race/ethnicity (character)

**parental.level.of.education** Parents level of education (character)

**lunch** Students lunch plan (character)

**test.preparation.course** Student test prep level (character)

**math.score** Students math score (integer)

**reading.score** Students reading score (integer)

**writing.score** Students writing score (integer)

**Source**

Data retrieved from [roycekimmons generated data](#)

---

football

*Football/Quarterback data*

---

### Description

Dataset containing performance statistics for 106 football players who attempted a pass in the NFL for the 2022 season.

### Usage

football

### Format

A data frame with 106 rows and 17 columns:

**Player** Players name (character)

**Tm** Players team (character)

**Age** Players Age (integer)

**Pos** Players position (character)

**G** Number of games (integer)

**GS** Number of games starting (integer)

**Wins** Number of wins (integer)

**Cmp** Number of completions (integer)

**Att** Number of throwing attempts (integer)

**Cmp.** Completion percentage (numeric)

**Yds** Number of yards thrown (integer)

**TD** Number of touchdowns (integer)

**Int** Number of interceptions thrown (integer)

**Y.A** Yards per Attempt (numeric)

**Y.G** Yards per Game (numeric)

**Rate** Passer rating (numeric)

**QBR** Total Quarterback Rating (numeric)

### Source

Data retrieved from [Pro Football Reference](#) with alterations made for educational purposes.

---

heart	<i>Heart data</i>
-------	-------------------

---

**Description**

Dataset containing medical and diagnostic information for 303 patients, used to study the presence of Atherosclerotic Heart Disease (AHD).

**Usage**

heart

**Format**

A data frame with 303 rows and 14 columns:

**Age** Patients age (integer)

**Sex** Patients Sex (1 = Male, 0 = Female) (integer)

**ChestPain** Chest pain type (character)

**RestBP** Resting blood pressure (in mm Hg on admission to the hospital) (integer)

**Chol** Serum cholesterol in mg/dl (integer)

**Fbs** fasting blood sugar > 120 mg/dl (1 = true; 0 = false) (integer)

**RestECG** Resting electrocardiographic results (integer)

**MaxHR** Maximum heart rate achieved (integer)

**ExAng** Exercise induced angina (1 = yes; 0 = no) (integer)

**Oldpeak** ST depression induced by exercise relative to rest (numeric)

**Slope** The slope of the peak exercise ST segment (integer)

**Ca** Number of major vessels (0-3) colored by fluoroscopy (integer)

**Thal** Thal condition (character)

**AHD** Atherosclerosis Heart Disease condition (character)

**Source**

Data retrieved from [UC Irvine Machine Learning Repository](#)

---

`housing_data`*Housing data*

---

**Description**

Data on houses that were recently sold in the Duke Forest neighborhood of Durham, NC in November 2020.

**Usage**`housing_data`**Format**

A data frame with 98 rows and 6 columns:

**price** Home price (numeric)

**bed** Number of bedrooms (integer)

**bath** Number of bathrooms (numeric)

**area** Square footage (integer)

**year\_built** Date house was built (integer)

**lot** lot size (numeric)

**Source**

Adapted from the `duke_forest` dataset in the `openintro` library with alterations made for educational purposes.

---

`income_data`*Income data*

---

**Description**

Dataset containing basic demographic and financial information for 20 individuals.

**Usage**`income_data`

**Format**

A data frame with 20 rows and 5 columns:

**ID** ID (integer)

**Ages** age (integer)

**Years\_til\_Retirement.65** Years until retirement at 65 (integer)

**Salary** Salary (integer)

**Birth\_weight** Birth weight (integer)

**Source**

Synthetic Data

---

kurt

*Compute Sample Kurtosis*

---

**Description**

Calculates the kurtosis of a numeric vector. A value near 0 suggests normal kurtosis (mesokurtic), positive values indicate heavier tails (leptokurtic), and negative values indicate lighter tails (platykurtic).

**Usage**

kurt(x)

**Arguments**

x                    A numeric vector.

**Details**

The z-scores are computed as:

$$z_i = \frac{x_i - \bar{x}}{sd}$$

The kurtosis is then calculated as:

$$\text{Kurtosis} = \frac{1}{n} \sum_{i=1}^n z_i^4 - 3$$

Where:

- $\bar{x}$  is the mean of  $x$ ,
- $sd$  is the standard deviation of  $x$ ,
- and  $n$  is the number of observations.

**Value**

A single numeric value representing the kurtosis

**Examples**

```
# Kurtosis of mpg in mtcars
data("mtcars")
kurt(mtcars$mpg)
```

---

ledger\_data

*Ledger data*

---

**Description**

Dataset mimicking a ledger showing the price an item was bought and sold for, the date it occurred, and the color of the product.

**Usage**

```
ledger_data
```

**Format**

A data frame with 4 rows and 104 columns:

**color** colors (character)

**type** age (integer)

**Jan\_08** Price on date (numeric)

**Jan\_15** Price on date (numeric)

**Jan\_16** Price on date (numeric)

**Jan\_31** Price on date (numeric)

**Feb\_02** Price on date (numeric)

**Feb\_03** Price on date (numeric)

**Feb\_04** Price on date (numeric)

**Feb\_14** Price on date (numeric)

**Feb\_20** Price on date (numeric)

**Feb\_22** Price on date (numeric)

**Feb\_25** Price on date (numeric)

**Feb\_27** Price on date (numeric)

**Feb\_28** Price on date (numeric)

**Mar\_01** Price on date (numeric)

**Mar\_05** Price on date (numeric)  
**Mar\_09** Price on date (numeric)  
**Mar\_12** Price on date (numeric)  
**Mar\_16** Price on date (numeric)  
**Mar\_20** Price on date (numeric)  
**Mar\_21** Price on date (numeric)  
**Mar\_22** Price on date (numeric)  
**Mar\_24** Price on date (numeric)  
**Mar\_27** Price on date (numeric)  
**Mar\_28** Price on date (numeric)  
**Mar\_31** Price on date (numeric)  
**Apr\_06** Price on date (numeric)  
**Apr\_08** Price on date (numeric)  
**Apr\_10** Price on date (numeric)  
**Apr\_18** Price on date (numeric)  
**Apr\_19** Price on date (numeric)  
**Apr\_24** Price on date (numeric)  
**Apr\_26** Price on date (numeric)  
**Apr\_29** Price on date (numeric)  
**May\_01** Price on date (numeric)  
**May\_04** Price on date (numeric)  
**May\_12** Price on date (numeric)  
**May\_17** Price on date (numeric)  
**May\_24** Price on date (numeric)  
**May\_25** Price on date (numeric)  
**May\_28** Price on date (numeric)  
**Jun\_01** Price on date (numeric)  
**Jun\_04** Price on date (numeric)  
**Jun\_11** Price on date (numeric)  
**Jun\_16** Price on date (numeric)  
**Jun\_25** Price on date (numeric)  
**Jun\_28** Price on date (numeric)  
**Jul\_03** Price on date (numeric)  
**Jul\_04** Price on date (numeric)  
**Jul\_08** Price on date (numeric)  
**Jul\_10** Price on date (numeric)  
**Jul\_11** Price on date (numeric)

**Jul\_13** Price on date (numeric)  
**Jul\_18** Price on date (numeric)  
**Jul\_23** Price on date (numeric)  
**Jul\_25** Price on date (numeric)  
**Aug\_05** Price on date (numeric)  
**Aug\_12** Price on date (numeric)  
**Aug\_13** Price on date (numeric)  
**Aug\_24** Price on date (numeric)  
**Aug\_26** Price on date (numeric)  
**Sep\_02** Price on date (numeric)  
**Sep\_06** Price on date (numeric)  
**Sep\_07** Price on date (numeric)  
**Sep\_08** Price on date (numeric)  
**Sep\_16** Price on date (numeric)  
**Sep\_21** Price on date (numeric)  
**Sep\_22** Price on date (numeric)  
**Sep\_23** Price on date (numeric)  
**Sep\_27** Price on date (numeric)  
**Oct\_07** Price on date (numeric)  
**Oct\_09** Price on date (numeric)  
**Oct\_10** Price on date (numeric)  
**Oct\_15** Price on date (numeric)  
**Oct\_16** Price on date (numeric)  
**Oct\_17** Price on date (numeric)  
**Oct\_19** Price on date (numeric)  
**Oct\_20** Price on date (numeric)  
**Oct\_21** Price on date (numeric)  
**Oct\_22** Price on date (numeric)  
**Oct\_29** Price on date (numeric)  
**Oct\_30** Price on date (numeric)  
**Oct\_31** Price on date (numeric)  
**Nov\_03** Price on date (numeric)  
**Nov\_04** Price on date (numeric)  
**Nov\_12** Price on date (numeric)  
**Nov\_13** Price on date (numeric)  
**Nov\_14** Price on date (numeric)  
**Nov\_16** Price on date (numeric)

**Nov\_18** Price on date (numeric)  
**Nov\_23** Price on date (numeric)  
**Nov\_24** Price on date (numeric)  
**Dec\_02** Price on date (numeric)  
**Dec\_03** Price on date (numeric)  
**Dec\_06** Price on date (numeric)  
**Dec\_11** Price on date (numeric)  
**Dec\_12** Price on date (numeric)  
**Dec\_13** Price on date (numeric)  
**Dec\_16** Price on date (numeric)  
**Dec\_17** Price on date (numeric)  
**Dec\_18** Price on date (numeric)  
**Dec\_19** Price on date (numeric)  
**Dec\_26** Price on date (numeric)

### Source

Synthetic Data

---

mlb\_eda

*MLB data*

---

### Description

Batter statistics for 2018 Major League Baseball season

### Usage

mlb\_eda

### Format

A data frame with 1270 rows and 13 columns:

**name** Players name (character)  
**team** Players team (character)  
**position** Players position (character)  
**games** Number of games (integer)  
**AB** Number of at bats (integer)  
**R** Number of runs (integer)  
**H** Number of hits (integer)  
**doubles** Number of doubles (integer)

- HR** Number of Home runs (integer)
- RBI** Number of Runs Batted In (integer)
- AVG** Players batting average (numeric)
- SLG** Players Slugging percentage (numeric)
- OPS** Players On-base Plus Slugging (numeric)

### Source

Data retrieved from [MLB](#), with alterations made for educational purposes.

---

MODE

*Find the Mode of a Numeric Vector*

---

### Description

Calculates the mode (most frequent value) of a numeric vector. If there is a tie, returns all values that share the highest frequency.

### Usage

```
MODE(x)
```

### Arguments

x                    A numeric vector.

### Value

A numeric value (or vector) representing the mode(s) of x.

### Examples

```
# Mode of a Numeric Vector
MODE(c(1,2,3,3,3,4,5,5,3,8))

# Mode of the number of cylinders in mtcars dataset
data("mtcars")
MODE(mtcars$cyl)
```

---

`mount_dorms`*Mount St.Mary's dorm data*

---

**Description**

Dataset summarizing the distribution of male and female students across various dormitories at Mount College, categorized by academic year.

**Usage**`mount_dorms`**Format**

A data frame with 4 rows and 11 columns:

**year** Students year (character)

**m\_Pangborn** Males living in Pangborn (integer)

**m\_Sheridan** Males living in Sheridan (integer)

**m\_Terrace** Males living in Terrace (integer)

**m\_Powell** Males living in Powell (integer)

**m\_Towers** Males living in the Towers (integer)

**f\_Pangborn** Females living in Pangborn (integer)

**f\_Sheridan** Females living in Sheridan (integer)

**f\_Terrace** Females living in Terrace (integer)

**f\_Powell** Females living in Powell (integer)

**f\_Towers** Females living in the Towers (integer)

**Source**

Synthetic Data

---

`MSMU`*MSMU: Fundamental Data Functions Package*

---

**Description**

The MSMU package provides core functions for descriptive statistics and exploratory data analysis. It includes functions for computing central tendency, spread, shape, and position statistics, along with utility functions for estimating modes and standardized ranges. The package contains

**Functions**

- `MODE()`
- `estimate_mode()`
- `center_stats()`
- `position_stats()`
- `pct()`
- `spread_stats()`
- `skew()`
- `kurt()`
- `shape_stats()`

**Datasets**

- `data_210_census`
- `class_demographics`
- `mlb_eda`
- `housing_data`
- `football`
- `college_data`
- `basketball`
- `mount_dorms`
- `baseball_teams`
- `christmas`
- `heart`
- `county_data`
- `reaction_time`
- `election_2020`
- `soccer`
- `exam_data`
- `course_scores`
- `income_data`
- `ledger_data`

**Author(s)**

Luke Papayoanou, Jon McCurdy

---

pct *Percent Within N Standard Deviations of the Mean*

---

**Description**

Calculates the percentage of values in a numeric vector that fall within n standard deviations of the mean.

**Usage**

```
pct(x, n)
```

**Arguments**

x A numeric vector.  
n A positive numeric value indicating how many standard deviations from the mean to use as bounds.

**Value**

A single numeric value representing the percentage (0–100) of values within the specified range.

**Examples**

```
# Percentage of values that fall within 2 sds of the mean in random normal data
set.seed(123)
x <- rnorm(1000)
pct(x,2)

# Percentage of values that fall within 2 sds of the mean in iris Sepal Lengths
data("iris")
pct(iris$Sepal.Length, 2)
```

---

position\_stats *Computes Position Statics, Quintiles and Quartiles*

---

**Description**

Calculates the quintiles, including quartiles(data is split in 4 equal parts) and quintiles(data is split in 5 equal parts) of a numeric vector using the 'quantile()' function. NA's are removed.

**Usage**

```
position_stats(x)
```

**Arguments**

x                    A numeric vector.

**Details**

**Percentiles** are values that divide a dataset into 100 equal parts, each representing 1% of the distribution. For example, the 25th percentile is the value below which 25% of the data fall.

**Quartiles** are special percentiles that divide the data into four equal groups: Q1 (25th percentile), Q2 (50th percentile or median), Q3 (75th percentile).

**Quintiles** divide data into five equal groups, each representing 20% of the distribution: 20th percentile, 40th, 60th, 80th percentiles split the data into quintiles.

**Value**

A list with two elements:

**quint** Numeric vector of quintiles (0%, 20%, 40%, ..., 100%)

**quart** Numeric vector of quartiles (0%, 25%, 50%, 75%, 100%)

**Examples**

```
# Position stats of random data
set.seed(123)
x <- rnorm(1000)
position_stats(x)

# Position stats of MPG in mtcars data set
data("mtcars")
position_stats(mtcars$mpg)
```

---

reaction_time	<i>Reaction Data</i>
---------------	----------------------

---

**Description**

This dataset contains synthetic reaction time measurements for 100 individuals under different conditions.

**Usage**

```
reaction_time
```

**Format**

A data frame with 100 rows and 6 columns:

**person** Person id (integer)

**color** color (character)

**left** left (numeric)

**right** right (numeric)

**age** Person age (numeric)

**gender** Person gender (character)

**Source**

Synthetic Data

---

shape\_stats

*Computes Sample Skew and Kurtosis*

---

**Description**

Calculates the skewness of a numeric vector (via `skew()`). A positive value indicates right skew (long right tail), while a negative value indicates left skew (long left tail). A zero value represents symmetry. Calculates the kurtosis of a numeric vector (via `kurt()`). A value near 0 suggests normal kurtosis (mesokurtic), positive values indicate heavier tails (leptokurtic), and negative values indicate lighter tails (platykurtic).

**Usage**

```
shape_stats(x)
```

**Arguments**

`x` A numeric vector.

**Value**

A list with two elements:

**skew** Skew of Data from `skew()`

**kurt** Kurtosis of Data from `kurt()`

**Examples**

```
# Shape stats of mpg in mtcars
data("mtcars")
shape_stats(mtcars$mpg)
```

---

skew

*Compute Sample Skewness*

---

### Description

Calculates the skewness of a numeric vector. A positive value indicates right skew (long right tail), while a negative value indicates left skew (long left tail). A zero value represents symmetry

### Usage

```
skew(x)
```

### Arguments

x                    A numeric vector.

### Value

A single numeric value representing the skewness of the distribution.

### Examples

```
# Skew of Sepal Lengths in iris
data("iris")
skew(iris$Sepal.Length)
```

---

soccer

*Historic soccer data*

---

### Description

This dataset contains historical match results from various international soccer games between different countries for the years 1872-2024.

### Usage

```
soccer
```

### Format

A data frame with 13750 rows and 5 columns:

**date** Date of match (character)

**home\_team** Home team name (character)

**away\_team** Away team name (character)

**home\_score** Home teams goal count (integer)

**away\_score** Away teams goal count (integer)

**Source**

Data retrieved from [Kaggle International football results dataset](#) with alterations made for educational purposes.

---

spread_stats	<i>Summary of Spread Statistics</i>
--------------	-------------------------------------

---

**Description**

Computes a variety of spread statistics for a numeric vector, including: standard deviation, iqr, the normalized minimum, maximum, and range as well as the percentage of data within 1, 2, and 3 standard deviations (via `pct()`)

**Usage**

```
spread_stats(x)
```

**Arguments**

`x` A numeric vector

**Value**

**sd** Standard Deviation

**iqr** Inter Quartile Range

**minz** Normalized Minimum

**maxz** Normalized Maximum

**diffz** Normalized Range

**pct1** Percent of data within 1 standard deviation from `pct()`

**pct2** Percent of data within 2 standard deviation from `pct()`

**pct3** Percent of data within 3 standard deviation from `pct()`

**See Also**

[pct](#)

**Examples**

```
# Spread stats of random normal data
set.seed(123)
x <- rnorm(1000)
spread_stats(x)

# Spread stats of mpg in mtcars
data("mtcars")
spread_stats(mtcars$mpg)
```

# Index

## \* datasets

- baseball\_teams, [2](#)
  - basketball, [3](#)
  - christmas, [5](#)
  - class\_demographics, [6](#)
  - college\_data, [6](#)
  - county\_data, [7](#)
  - course\_scores, [8](#)
  - data\_210\_census, [9](#)
  - election\_2020, [10](#)
  - exam\_data, [11](#)
  - football, [12](#)
  - heart, [13](#)
  - housing\_data, [14](#)
  - income\_data, [14](#)
  - ledger\_data, [16](#)
  - mlb\_eda, [19](#)
  - mount\_dorms, [21](#)
  - reaction\_time, [24](#)
  - soccer, [26](#)
- 
- baseball\_teams, [2](#), [22](#)
  - basketball, [3](#), [22](#)
- 
- center\_stats, [4](#)
  - center\_stats(), [22](#)
  - christmas, [5](#), [22](#)
  - class\_demographics, [6](#), [22](#)
  - college\_data, [6](#), [22](#)
  - county\_data, [7](#), [22](#)
  - course\_scores, [8](#), [22](#)
- 
- data\_210\_census, [9](#), [22](#)
- 
- election\_2020, [10](#), [22](#)
  - estimate\_mode, [4](#), [10](#)
  - estimate\_mode(), [22](#)
  - exam\_data, [11](#), [22](#)
- 
- football, [12](#), [22](#)
- 
- heart, [13](#), [22](#)
  - housing\_data, [14](#), [22](#)
- 
- income\_data, [14](#), [22](#)
- 
- kurt, [15](#)
  - kurt(), [22](#)
- 
- ledger\_data, [16](#), [22](#)
- 
- mlb\_eda, [19](#), [22](#)
  - MODE, [20](#)
  - MODE(), [22](#)
  - mount\_dorms, [21](#), [22](#)
  - MSMU, [21](#)
  - MSMU-package (MSMU), [21](#)
- 
- pct, [23](#), [27](#)
  - pct(), [22](#)
  - position\_stats, [23](#)
  - position\_stats(), [22](#)
- 
- reaction\_time, [22](#), [24](#)
- 
- shape\_stats, [25](#)
  - shape\_stats(), [22](#)
  - skew, [26](#)
  - skew(), [22](#)
  - soccer, [22](#), [26](#)
  - spread\_stats, [27](#)
  - spread\_stats(), [22](#)