# Package ‘dslabs’

July 14, 2019

**Title** Data Science Labs

**Version** 0.7.1

**Description** Datasets and functions that can be used for data analysis practice, homework and projects in data science courses and workshops. 26 datasets are available for case studies in data visualization, statistical inference, modeling, linear regression, data wrangling and machine learning.

**Author** Rafael A. Irizarry, Amy Gill

**Maintainer** Rafael A. Irizarry <rafa@jimmy.harvard.edu>

**Depends** R (>= 3.1.2)

**Imports** ggplot2

**License** Artistic-2.0

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.1.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2019-07-14 21:30:04 UTC

## R topics documented:

admissions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
brca . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
brexit_polls . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
death_prob . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
divorce_margarine . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
ds_theme_set . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
gapminder . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
greenhouse_gases . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
heights . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
historic_co2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
mnist_27 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10

movielens . . . . . . . . . . 11
murders . . . . . . . . . . 12
na_example . . . . . . . . . . 12
nyc_regents_scores . . . . . . . . . . 13
olive . . . . . . . . . . 14
outlier_example . . . . . . . . . . 15
polls_2008 . . . . . . . . . . 15
polls_us_election_2016 . . . . . . . . . . 16
read_mnist . . . . . . . . . . 17
reported_heights . . . . . . . . . . 18
research_funding_rates . . . . . . . . . . 19
rfalling_object . . . . . . . . . . 20
stars . . . . . . . . . . 21
take_poll . . . . . . . . . . 21
temp_carbon . . . . . . . . . . 22
tissue_gene_expression . . . . . . . . . . 23
trump_tweets . . . . . . . . . . 24
us_contagious_diseases . . . . . . . . . . 25

**Index** **26**

---

`admissions` *Gender bias among graduate school admissions to UC Berkeley.*

---

## Description

The admission data for six majors for the fall of 1973; often used as an example of Simpson's paradox

## Usage

```
data(admissions)
```

## Format

An object of class "`data.frame`".

## Details

- major. The major or university department.
- gender. Men or women.
- admitted. Percent of students admitted.
- applicants. Total number of applicants.

## Source

PJ Bickel, EA Hammel, and JW O'Connell. Science (1975)

### Examples

```
data(admissions)
admissions
```

## brca *Breast Cancer Wisconsin Diagnostic Dataset from UCI Machine Learning Repository*

### Description

Biopsy features for classification of 569 malignant (cancer) and benign (not cancer) breast masses.

### Usage

```
data(brca)
```

### Format

An object of class `list`.

### Details

Features were computationally extracted from digital images of fine needle aspirate biopsy slides. Features correspond to properties of cell nuclei, such as size, shape and regularity. The mean, standard error, and worst value of each of 10 nuclear parameters is reported for a total of 30 features.

This is a classic dataset for training and benchmarking machine learning algorithms.

- y. The outcomes. A factor with two levels denoting whether a mass is malignant ("M") or benign ("B").
- x. The predictors. A matrix with the mean, standard error and worst value of each of 10 nuclear measurements on the slide, for 30 total features per biopsy:
  - radius. Nucleus radius (mean of distances from center to points on perimeter).
  - texture. Nucleus texture (standard deviation of grayscale values).
  - perimeter. Nucleus perimeter.
  - area. Nucleus area.
  - smoothness. Nucleus smoothness (local variation in radius lengths).
  - compactness. Nucleus compactness (perimeter^2/area - 1).
  - concavity, Nucleus concavity (severity of concave portions of the contour).
  - concave_pts. Number of concave portions of the nucleus contour.
  - symmetry. Nucleus symmetry.
  - fractal_dim. Nucleus fractal dimension ("coastline approximation" -1).

### Source

UCI Machine Learning Repository

### Examples

```
data(brca)
table(brca$y)
dim(brca$x)
head(brca$x)
```

---

## `brexit_polls` *Brexit Poll Data*

---

### Description

Brexit (EU referendum) poll outcomes for 127 polls from January 2016 to the referendum date on June 23, 2016.

### Usage

```
data(brexit_polls)
```

### Format

An object of class `"data.frame"`.

### Details

- startdate. Start date of poll.
- enddate. End date of poll.
- pollster. Pollster conducting the poll.
- poll_type. Online or telephone poll.
- samplesize. Sample size of poll.
- remain. Proportion voting Remain.
- leave. Proportion voting Leave.
- undecided. Proportion of undecided voters.
- spread. Spread calculated as remain - leave.

### Source

Wikipedia

### Examples

```
data(brexit_polls)
head(brexit_polls)
```

## `death_prob` *2015 US Period Life Table*

### Description

Probability of death within 1 year by age and sex in the United States in 2015.

### Usage

```
data(death_prob)
```

### Format

An object of class `"data.frame"`.

### Details

- age. Age strata, with each year a different stratum.
- sex. Male or Female.
- prob. Probability of death within 1 year given exact age and sex.

### Source

Social Security Administraton

### Examples

```
data(death_prob)
head(death_prob)
```

## `divorce_margarine` *Divorce rate and margarine consumption data*

### Description

Divorce rates in Maine and per capita consumption of margarine in US data

### Usage

```
data(divorce_margarine)
```

### Format

An object of class `"data.frame"`.

## Details

- divorce_rate_maine. Divorce per 1000 in Maine.
- margarine_consumption_per_capita. US per capita consumption of margarine in pounds.
- year. Year.

## Source

Spurious Correlations

## Examples

```
data(divorce_margarine)
with(divorce_margarine, plot(margarine_consumption_per_capita, divorce_rate_maine))
```

---

# `ds_theme_set` *dslabs theme set*

---

## Description

This function sets a ggplot2 theme used throughout the data science labs. It can be called without arguments.

## Usage

```
ds_theme_set(new = "theme_bw", args = NULL, base_size = 11,
  bold_title = TRUE, ...)
```

## Arguments

| | |
|---|---|
| `new` | a prebuilt ggplot2 theme. Defaults to `"theme_minimal"` |
| `args` | the arguments to be passed along to the ggplot2 theme function. Defaults to `"NULL"`. |
| `base_size` | if `"args"` is `"NULL"`, base_size is one of the arguments passed to the theme function. It defaults to 11. |
| `bold_title` | if TRUE, sets titles to be bold |
| `...` | additional arguments to be used by theme |

## Value

None

### Examples

```
library(ggplot2)
ds_theme_set()
qplot(hp, mpg, data=mtcars, color=am, facets=gear~cyl,
main="Scatterplots of MPG vs. Horsepower",
xlab="Horsepower", ylab="Miles per Gallon")
```

## gapminder *Gapminder Data*

### Description

Health and income outcomes for 184 countries from 1960 to 2016. Also includes two character vectors, `oecd` and `opec`, with the names of OECD and OPEC countries from 2016.

### Usage

```
data(gapminder)
```

### Format

An object of class `"data.frame"`.

### Details

- country.
- year.
- infant_mortality. Infant deaths per 1000.
- life_expectancy. Life expectancy in years.
- fertility. Average number of children per woman.
- population. Country population.
- gpd. GDP according to World Bankdev.
- continent.
- region. Geographical region.

### Examples

```
data(gapminder)
head(gapminder)
print(oecd)
print(opec)
```

## greenhouse_gases *Greenhouse gas concentrations over 2000 years*

### Description

Concentrations of the three main greenhouse gases carbon dioxide, methane and nitrous oxide. Measurements are from the Law Dome Ice Core in Antarctica. Selected measurements are provided every 20 years from 1-2000 CE.

### Usage

```
data(greenhouse_gases)
```

### Format

An object of class `"data.frame"`.

### Details

- year. Year (CE).
- gas. Gas being measured: carbon dioxide (‘CO2‘), methane (‘CH4‘) or nitrous oxide (‘N2O‘).
- concentration. Gas concentration in ppm by volume (‘CO2‘) or ppb by volume (‘CH4‘, ‘N2O‘).

### Source

MacFarling Meure et al. 2006 via NOAA.

### Examples

```
data(greenhouse_gases)
head(greenhouse_gases)
```

## heights *Self-Reported Heights*

### Description

Self-reported heights in inches for males and females.

### Usage

```
data(heights)
```

### Format

An object of class `"data.frame"`.

### Details

- sex. Male or Female.
- height. Height in inches.

### Examples

```
data(heights)
head(heights)
```

## `historic_co2` *Atmospheric carbon dioxide concentration over 800,000 years*

### Description

Concentration of carbon dioxide in ppm by volume from direct measurements at Mauna Loa (1959-2018 CE) and indirect measurements from a series of Antarctic ice cores (approx. -800,000-2001 CE).

### Usage

```
data(historic_co2)
```

### Format

An object of class `"data.frame"`.

### Details

- year. Year (CE).
- co2. Carbon dioxide concentration in ppm by volume.
- source. Source of carbon dioxide measurement: direct CO2 annual mean concentrations from Mauna Loa (‘Mauna Loa‘) or indirect CO2 concentrations from air trapped in ice cores (‘Ice Cores‘).

### Source

Mauna Loa data from NOAA. Ice core data from Bereiter et al. 2015 via NOAA.

### Examples

```
data(historic_co2)
head(historic_co2)
```

mnist_27 *Useful example for illustrating machine learning algorithms based on MNIST data*

## Description

We only include a randomly selected set of 2s and 7s along with the two predictors based on the proportion of dark pixels in the upper left and lower right quadrants respectively. The dataset is divided into training and test sets.

## Usage

```
data(mnist_27)
```

## Format

An object of class `list`.

## Details

- train. A data frame containing training data: labels and predictors.
- test. A data frame containing test data: labels and predictors.
- index_train. The index of the original mnist training data used for the training set.
- index_test. The index of the original mnist test data used for the test set.
- true_p. A `data.frame` containing the two predictors `x_1` and `x_2` and the conditional probability of being a 7 for `x_1`, `x_2`.

## Source

http://yann.lecun.com/exdb/mnist/

## Examples

```
data(mnist_27)
with(mnist_27$train, plot(x_1, x_2, col = as.numeric(y)))
```

`movielens` *Movie ratings*

## Description

MovieLens Latest Dataset (Small)

## Usage

```
data(movielens)
```

## Format

Two object of class `data.frame`.

## Details

- movieId. Unique ID for the movie.
- title. Movie title (not unique).
- year. Year the movie was released.
- genres. Genres associated with the movie.
- userId. Unique ID for the user.
- rating. A rating between 0 and 5 for the movie.
- timestamp. Date and time the rating was given.

## Source

http://files.grouplens.org/datasets/movielens/ml-latest-small.zip

## References

F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4, Article 19 (December 2015), 19 pages. DOI=http://dx.doi.org/10.1145/2827872

## Examples

```
data(movielens)
head(movielens)
```

## murders *US gun murders by state for 2010*

### Description

Gun murder data from FBI reports. Also contains the population of each state.

### Usage

```
data(murders)
```

### Format

An object of class "data.frame".

### Details

- state. US state
- abb. Abbreviation of US state
- region. Geographical US region
- population. State population (2010)
- total. Number of gun murders in state (2010)

### Source

Wikipedia

### Examples

```
data(murders)
print(murders)
```

## na_example *Count data with some missing values*

### Description

This dataset was randomly generated.

### Usage

```
data(na_example)
```

### Format

An object of class "integer".

### Examples

```
data(na_example)
print(sum(is.na(na_example)))
```

## nyc_regents_scores *NYC Regents exams scores 2010*

### Description

Distribution of scores for New York City Regents algebra, global history, biology, English, and U.S. history exams. These data were used to make this New York Times plot.

### Usage

```
data(nyc_regents_scores)
```

### Format

An object of class "data.frame".

### Details

- score. Test score from 0 to 100.
- integrated_algebra. Score frequency on Algebra exam.
- global_history. Score frequency on global history exam.
- living_environment. Score frequency on biology exam.
- english. Score frequency on English exam.
- us_history. Score frequency on U.S. history exam.

### Source

New York City Department of Education via Amanda Cox.

### Examples

```
data(nyc_regents_scores)
print(nyc_regents_scores)
```

## `olive` *Italian olive*

### Description

Composition in percentage of eight fatty acids found in the lipid fraction of 572 Italian olive oils

### Usage

```
data(olive)
```

### Format

An object of class `"data.frame"`.

### Details

- region. General region of Italy.
- area. Area of Italy.
- palmitic. Percent palmitic acid of sample.
- palmitoleic. Percent palmitoleic of sample.
- stearic. Percent stearic acid of sample.
- oleic. Percent oleic acid of sample.
- linoleic. Percent linoleic acid of sample.
- linolenic. Percent linolenic acid of sample.
- arachidic. Percent arachidic acid of sample.
- eicosenoic. Percent eicosenoic acid of sample.

### Source

J. Zupan, and J. Gasteiger. Neural Networks in Chemistry and Drug Design.

### Examples

```
data(olive)
head(olive)
```

## `outlier_example` *Adult male heights in feet with outliers*

### Description

This dataset was randomly generated with a normal distribution (average: 5 feet 9 inches, standard deviation: 3 inches). One value was changed to be mistakenly reported in centimeters rather than feet.

### Usage

```
data(outlier_example)
```

### Format

An object of class `"numeric"`.

### Examples

```
data(outlier_example)
mean(outlier_example)
median(outlier_example)
```

## `polls_2008` *Poll data for popular vote in 2008 presidential election*

### Description

Data from different pollsters for the popular vote between Obama and McCain in the 2008 presidential election.

### Usage

```
data(polls_2008)
```

### Format

An object of class `data.frame`.

### Details

- day. Days until election day. Negative numbers are reported so that days can increase up to 0, which is election day.
- margin. Average difference between Obama and McCain for that day.

### Source

https://web.archive.org/web/20161108190914/http://www.pollster.com/08USPresGEMvO-2.html

### Examples

```
data(polls_2008)
with(polls_2008, plot(day, margin))
```

---

polls_us_election_2016

## *Fivethirtyeight 2016 Poll Data*

---

### Description

Poll results from US 2016 presidential elections aggregated from HuffPost Pollster, RealClearPolitics, polling firms and news reports. The original csv file is here: `http://projects.fivethirtyeight.com/general-model/president_general_polls_2016.csv`. The dataset also includes election results (popular vote) and electoral college votes in results_us_election_2016.

### Usage

```
data(polls_us_election_2016)
```

### Format

An object of class `"data.frame"`.

### Details

- state. State in which poll was taken. 'U.S' is for national polls.
- startdate. Poll's start date.
- enddate. Poll's end date.
- pollster. Pollster conducting the poll.
- grade. Grade assigned by fivethirtyeight to pollster.
- samplesize. Sample size.
- population. Type of population being polled.
- rawpoll_clinton. Percentage for Hillary Clinton.
- rawpoll_trump. Percentage for Donald Trump
- rawpoll_johnson. Percentage for Gary Johnson
- rawpoll_mcmullin. Percentage for Evan McMullin.
- adjpoll_clinton. Fivethirtyeight adjusted percentage for Hillary Clinton.
- ajdpoll_trump. Fivethirtyeight adjusted percentage for Donald Trump
- adjpoll_johnson. Fivethirtyeight adjusted percentage for Gary Johnson
- adjpoll_mcmullin. Fivethirtyeight adjusted percentage for Evan McMullin.

### Source

Ballotpedia

### Examples

```
data(polls_us_election_2016)
head(polls_us_election_2016)
```

---

## read_mnist *Download and read the mnist dataset*

---

### Description

This function downloads the mnist training and test data from http://yann.lecun.com/exdb/mnist/

### Usage

```
read_mnist()
```

### Value

A list with two components: train and test. Each of these is a list with two components: images and labels. The images component is a matrix with each column representing one of the 28*28 = 784 pixels. The values are integers between 0 and 255 representing grey scale. The labels components is a vector representing the digit shown in the image.

Note that the data is over 200MB, so the download may take several seconds depending on internet speed.

### Author(s)

Samuela Pollack, `<spollack@jimmy.harvard.edu>`

### Source

http://yann.lecun.com/exdb/mnist/

### References

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition." Proceedings of the IEEE, 86(11):2278-2324, November 1998.

## Examples

```
# this can take several seconds, depending on internet speed.
## Not run:
mnist <- read_mnist()
i <- 5
image(1:28, 1:28, matrix(mnist$test$images[i,], nrow=28)[ , 28:1],
    col = gray(seq(0, 1, 0.05)), xlab = "", ylab="")
## the labels for this image is:
mnist$test$labels[i]

## End(Not run)
```

# reported_heights *Self-reported Heights*

## Description

Students were asked to report their height (in inches) and sex in an online form. This table includes the results from four courses.

## Usage

```
data(reported_heights)
```

## Format

An object of class "data.frame".

## Details

- time_stamp. Time and date of the entry.
- sex. Sex of the student.
- height. Height as reported by student by filling in a text free box.

## Examples

```
data(reported_heights)
head(reported_heights)
```

research_funding_rates
*Gender bias in research funding in the Netherlands*

## Description

Table S1 from paper title "Gender contributes to personal research funding success in The Netherlands"

## Usage

```
data(research_funding_rates)
```

## Format

An object of class "data.frame".

## Details

- discipline. Research area discipline.
- applications_total. Total applications.
- applications_men. Total applications by men.
- applications_women. Total applications by women.
- awards_total. Total awards.
- awards_men. Total awards received by men.
- awards_women. Total awards received by women.
- success_rates_total. Overall success rate.
- success_rates_men. Success rate for men.
- success_rates_women. Success rate for women.

## Source

van der Lee and Ellemers (2015) PNAS http://www.pnas.org/content/112/40/12349.abstract

## Examples

```
data(research_funding_rates)
research_funding_rates
# The raw data for this table is available from
data(raw_data_research_funding_rates)
```

## `rfalling_object` *Simulate falling object data*

### Description

The function simulates a falling object's position. Default parameters are for dropping a weight from the tower of Pisa.

### Usage

```
rfalling_object(n = 14, d_0 = 55.86, v_0 = 0, g = -9.8,
  scale = 1, time = seq(0, 3.25, length.out = n),
  error_distribution = c("rnorm", "rt"), df = 3)
```

### Arguments

| | |
|---|---|
| `n` | Sample size |
| `d_0` | Height from which object will fall in meters. |
| `v_0` | Initial velocity with which object will fall in meters per second. |
| `g` | Gravitational constant, 9.8 meters per second per seonnd |
| `scale` | The measurement errors will be multiplied by this constant. |
| `time` | Numeric vector of times, in seconds, at which measurements were taken. |
| `error_distribution` | Character. Either `rnorm` for normal or `rt` for t-distribution. |
| `df` | If using t-distribution, the degrees of freedom. |

### Value

A `data.frame` with the time, the distance travelled, and the observed distance.

### Examples

```
dat <- rfalling_object()
with(dat, plot(time, observed_distance))
with(dat, lines(time, distance, col = "blue"))
```

## `stars` *Physical Properties of Stars*

### Description

Physical properties of selected stars, including luminosity, temperature, and spectral class.

### Usage

```
data(stars)
```

### Format

An object of class `"data.frame"`.

### Details

- star. Name of star.
- magnitude. Absolute magnitude of the star, which is a function of the star's luminosity and distance to the star.
- temp. Surface temperature in degrees Kelvin (K).
- type. Spectral class of star in the OBAFGKM system.

### Source

Compiled from multiple open-access references on VizieR.

### Examples

```
data(stars)
head(stars)
```

## `take_poll` *Models results from taking a poll*

### Description

The function shows a plot of a random sample drawn from an urn with blue and red beads. The sample is taken with replacement. The proportion of blue beads is not shown so that students can try to estimate it.

### Usage

```
take_poll(n, ...)
```

### Arguments

| | |
|---|---|
| `n` | Sample size |
| `...` | additional arguments to be used by the function `sample`. |

### Value

None

### Examples

```
take_poll(25)
```

## `temp_carbon` *Global temperature anomaly and carbon emissions, 1751-2018*

### Description

Annual mean global temperature anomaly on land, sea and combined, 1880-2018. Annual global carbon emissions, 1751-2014.

### Usage

```
data(temp_carbon)
```

### Format

An object of class `"data.frame"`.

### Details

- year. Year (CE).
- temp_anomaly. Global annual mean temperature anomaly in degrees Celsius relative to the 20th century mean temperature. 1880-2018.
- land_anomaly. Annual mean temperature anomaly on land in degrees Celsius relative to the 20th century mean temperature. 1880-2018.
- ocean_anomaly. Annual mean temperature anomaly over ocean in degrees Celsius relative to the 20th century mean temperature. 1880-2018.
- carbon_emissions. Annual carbon emissions in millions of metric tons of carbon. 1751-2014.

### Source

NOAA and Boden, T.A., G. Marland, and R.J. Andres (2017) via CDIAC

### Examples

```
data(temp_carbon)
head(temp_carbon)
```

## tissue_gene_expression

*Gene expression profiles for 189 biological samples taken from seven different tissue types.*

### Description

This is a subset of the data provided by the `tissuesGeneExpression` package available from the `genomicsclass` GitHub repository. The predictors are gene expression measurements from 500 genes that are a random subset of the original 22,215.

### Usage

```
data(tissue_gene_expression)
```

### Format

An object of class `list`.

### Details

The example dataset is recommended for illustrating clustering and machine learning techniques.

- x. The predictors composed of 500 genes. Each row is a gene expression profile and each column is different gene. The column names are the gene symbols.
- y. The outcomes. A character vector representing the tissue. One of seven tissue types.

### Source

https://github.com/genomicsclass/tissuesGeneExpression

### Examples

```
data(tissue_gene_expression)
table(tissue_gene_expression$y)
dim(tissue_gene_expression$x)
```

## `trump_tweets` *Trump Tweets from2009 to 2017*

### Description

All tweets from Donald Trump's twitter account from 2009 to 2017

### Usage

```
data(trump_tweets)
```

### Format

An object of class `"data.frame"`.

### Details

- source. Device or service used to compose tweet.
- id_str. Tweet ID.
- text. Tweet.
- created_at. Data and time tweet was tweeted.
- retweet_count. How many times tweet had been retweeted at time dataset was created.
- in_reply_to_user_id_str. If a reply, the user id of person being replied to.
- favorite_count. Number of times tweet had been favored at time dataset was created.
- is_retweet. A logical telling us if it is a retweet or not.

### Source

The Trump Twitter Archive: http://www.trumptwitterarchive.com

### Examples

```
data(trump_tweets)
head(trump_tweets)
```

us_contagious_diseases

*Contagious disease data for US states*

## Description

Yearly counts for Hepatitis A, Measles, Mumps, Pertussis, Polio, Rubella, and Smallpox for US states. Original data courtesy of Tycho Project (http://www.tycho.pitt.edu/).

## Usage

```
data(us_contagious_diseases)
```

## Format

An object of class `"data.frame"`.

## Details

- disease. A factor containing disease names.
- state. A factor containing state names.
- year.
- weeks_reporting. Number of weeks counts were reported that year.
- count. Total number of reported cases.
- population. State population, interpolated for non-census years.

## Source

Tycho Project

## Examples

```
data(us_contagious_diseases)
head(us_contagious_diseases)
```

# Index

∗Topic **datasets**
- admissions, 2
- brca, 3
- brexit_polls, 4
- death_prob, 5
- divorce_margarine, 5
- gapminder, 7
- greenhouse_gases, 8
- heights, 8
- historic_co2, 9
- mnist_27, 10
- movielens, 11
- murders, 12
- na_example, 12
- nyc_regents_scores, 13
- olive, 14
- outlier_example, 15
- polls_2008, 15
- polls_us_election_2016, 16
- reported_heights, 18
- research_funding_rates, 19
- stars, 21
- temp_carbon, 22
- tissue_gene_expression, 23
- trump_tweets, 24
- us_contagious_diseases, 25

admissions, 2

brca, 3
brexit_polls, 4

death_prob, 5
divorce_margarine, 5
ds_theme_set, 6

gapminder, 7
greenhouse_gases, 8

heights, 8
historic_co2, 9

mnist_27, 10
movielens, 11
murders, 12

na_example, 12
nyc_regents_scores, 13

oecd *(gapminder)*, 7
olive, 14
opec *(gapminder)*, 7
outlier_example, 15

polls_2008, 15
polls_us_election_2016, 16

raw_data_research_funding_rates *(research_funding_rates)*, 19
read_mnist, 17
reported_heights, 18
research_funding_rates, 19
results_us_election_2016 *(polls_us_election_2016)*, 16
rfalling_object, 20

stars, 21

take_poll, 21
temp_carbon, 22
tissue_gene_expression, 23
trump_tweets, 24

us_contagious_diseases, 25