Package 'mdsr'

August 19, 2024

Title Complement to 'Modern Data Science with R'

Version 0.2.8

Description A complement to all editions of *Modern Data Science with R*
(ISBN: 978-0367191498, publisher URL:
<https://www.routledge.com/Modern-Data-Science-with-R/Baumer-Kaplan-Horton/p/book/9780367191498>).
This package contains data and code to complete exercises and reproduce examples from the text. It also facilitates connections to the SQL database server used in the book. All editions of the book are supported by this package.

Depends R (>= 4.1.0)

License CC0

LazyData true

LazyDataCompression xz

Imports babynames, DBI, dbplyr, downloader, dplyr, fs, ggplot2, htmlwidgets, kableExtra, RMariaDB, skimr, stringr, tibble, webshot2

Suggests etl, knitr, Lahman, leaflet, lubridate, macleish, mosaic, mosaicData, nycflights13, nycflights23, sf, testthat, utf8

RoxygenNote 7.3.2

Encoding UTF-8

URL https://github.com/mdsr-book/mdsr

BugReports https://github.com/mdsr-book/mdsr/issues

NeedsCompilation no

Author Benjamin S. Baumer [aut, cre] (<https://orcid.org/0000-0002-3279-0516>), Nicholas Horton [aut] (<https://orcid.org/0000-0003-3332-4311>), Daniel Kaplan [aut]

Maintainer Benjamin S. Baumer <ben.baumer@gmail.com>

Repository CRAN

Date/Publication 2024-08-19 19:30:02 UTC

Contents

`Cherry` *Cherry Blossom runs*

Description

Cherry Blossom runs

Usage

```
Cherry
```

Format

An object of class tibble::tbl_df with 41,248 rows and 8 columns. Each row refers to an individual runner in one race of the Cherry Blossom Ten Miler. The data cover the years 1999 to 2008. All of the runners listed ran at least two of the races in that period, some ran many more than that.

name.yob a unique identifier for each runner composed of the runner's full name and year of birth.

age integer giving the runner's age in the race whose result is being reported.

gun the number of minutes elapsed from the starter's gun to the person crossing the finish line

net the number of minutes elapsed from the runner's crossing the start line to crossing the finish line.

sex the runner's sex

year the year of that race

previous integer specifying how many times previous to this race the runner had participated in the years 1999 to 2008.

nruns integer giving the total number of times that runner participated in the years from 1999 to 2008. The smallest is 2, the largest is 10.

nruns integer giving the total number of times that runner participated in the years from 1999 to 2008. The smallest is 2, the largest is 10.

Details

The Cherry Blossom 10 Mile Run is a road race held in Washington, D.C. in April each year. (The name comes from the famous cherry trees that are in bloom in April in Washington.) The results of this race are published at https://www.cherryblossom.org/post-race/race-results/.

Source

https://www.cherryblossom.org/post-race/race-results/.

See Also

Data Science in R, Nolan and Temple Lang (ISBN 978-1482234817), Ch. 2

Examples

```
if (require(dplyr)) {
  Cherry |>
    group_by(name.yob) |>
    count() |>
    group_by(n) |>
    count(name = "appearances")
}
```

CholeraDeaths *Deaths and Pumps from 1854 London cholera outbreak*

Description

Deaths and Pumps from 1854 London cholera outbreak

Usage

```
CholeraDeaths

CholeraPumps
```

Format

An object of class `sf::sf()` whose data attribute has 250 rows and 2 columns.

An object of class sf::sf.

Details

Both spatial objects are projected in EPSG:27700, aka the British National Grid.

Source

https://blog.rtwilson.com/john-snows-cholera-data-in-more-formats/

Examples

```
if (require(sf)) {
  plot(st_geometry(CholeraDeaths))
}
```

CIACountries *Several variables on countries from the CIA Factbook, 2014.*

Description

The CIA Factbook has geographic, demographic, and economic data on a country-by-country basis. In the description of the variables, the 4-digit number indicates the code used to specify that variable on the data and documentation web site.

Usage

```
CIACountries
```

Format

A data frame with the following variables for each of the Countries in the World. (236 countries are given.)

country Name of the country

pop number of people, 2119

area area (sq km), 2147

oil_prod Crude oil - production (bbl/day), 2241

gdp Gross Domestic Product per capita ($/person), 2001

educ education spending (% of GDP), 2206

roadways Roadways per unit area (km/sq km), 2085

net_users Fraction of Internet users (% of population), 2153

Source

From the CIA World Factbook, https://www.cia.gov/the-world-factbook/

References

https://github.com/factbook/factbook/blob/master/CATEGORIES.md

See Also

mosaic::CIAdata

Examples

```
str(CIACountries)
```

DataSciencePapers *Data Science Papers from arXiv.org*

Description

Papers matching the search string "Data Science" on arXiv.org in August, 2020

Usage

```
DataSciencePapers
```

Format

A data frame with 1089 observations on the following 15 variables.

id unique arXiv.org identifier for the paper

submitted date submitted

updated date last updated

title title of the paper

abstract contents of the abstract

authors authors of the paper

affiliations affiliations of the authors

link_abstract direct link to the abstract

link_pdf direct link to the pdf

link_doi direct link to the digital object identifier (doi)

comment commentary

journal_ref reference to the journal (if published)

doi digital object identifier

primary_category arXiv.org primary category

categories arXiv.org categories

Source

https://arxiv.org/

Examples

```
data(DataSciencePapers)
str(DataSciencePapers)
```

Elections *Election Statistics from the 2013 Minneapolis Mayoral Election*

Description

Election Statistics from the 2013 Minneapolis Mayoral Election

Usage

```
Elections
```

Format

An object of class tibble::tbl_df with 117 rows and 13 columns.

Ward Number of the ward

Precinct Number of the precinct

Registered Voters at 7am Number of registered votes as of 7 am

Voters Registering at Polls Number of voters registering at the polls

Voters Registering by Absentee Number of voters registering by absentee

Total Registrations Total number of registered voters

Voters at Polls Number of voters at the polls

Absentee Voters Number of absentee voters

Total Ballots Cast Number of total ballots cast

Total Turnout Total number of voters turning out

Percentage Absentee Percentage of absentee voters

% Registered to Total (Election Day) Percentage of voters relative to total number of people

Spoiled Ballots Number of spolied ballots

Source

https://vote.minneapolismn.gov/results-data/election-results/2013/mayor/

`Emails_train` *Email Train*

Description

The training dataset includes a set of email subject lines used for classification of whether the message is spam (unsolicited commercial content) or not. Many subject lines include subject matter inappropriate for classroom use. Given the volume of headlines containing such language (especially for `spam == TRUE`), user discretion is advised. This dataset is a random sample of 80% of the emails data.

The testing dataset is a random sample of 20% of the emails data.

Usage

```
Emails_train

Emails_test
```

Format

A data frame with 5,526 rows and 3 variables:

ids an integer vector

subjectline a character vector

type a character vector

A data frame with 1,382 rows and 3 variables:

Source

Originally retrieved from https://www.stat.berkeley.edu/~nolan/data/spam/SpamAssassinMessages.zip

See Also

Data Science in R, Nolan and Temple Lang (ISBN 978-1482234817), Ch. 3

Examples

```
nrow(Emails_train)
nrow(Emails_test)
```

---

etl_NCI60 *Load the NCI60 data from GitHub*

Description

Load the NCI60 data from GitHub

Usage

```
etl_NCI60()
```

Value

A tibble::tbl_df

Examples

```
# The file is 5.0 MB
NCI60 <- etl_NCI60()
```

Headlines_train *Headlines_train*

Description

This data comes from Chakraborty et. al., which combines headlines from a variety of news and clickbait sources. Some headlines contain subject matter inappropriate for classroom use. Given the volume of headlines containing such language (especially for `clickbait == TRUE`), this filtering might not catch all problematic headlines. User discretion is advised. The training dataset is a random sample of approximately 80% of the observations from the original dataset.

The testing dataset is a random sample of the remaining 20% of the observations not found in the training set.

Usage

```
Headlines_train

Headlines_test
```

Format

A data frame with 18,360 rows and 3 variables:

title a character vector

clickbait a logical vector

ids an integer vector

A data frame with 4,589 rows and 3 variables:

Source

https://github.com/bhargaviparanjape/clickbait/

References

doi:10.1109/ASONAM.2016.7752207

Examples

```
nrow(Headlines_train)
nrow(Headlines_test)
```

Macbeth_raw *Text of Macbeth*

Description

The entire text of Macbeth, stored in a character vector of length 1.

Usage

```
Macbeth_raw
```

Format

A character vector of length 1

Source

Project Gutenberg, https://www.gutenberg.org/ebooks/1129/

make_babynames_dist *Wrangle babynames data*

Description

Wrangle babynames data

Usage

```
make_babynames_dist()
```

Value

a tibble::tbl_df similar to babynames::babynames with a column for the estimated number of people alive in 2014.

Examples

```
BabynameDist <- make_babynames_dist()
if (require(dplyr)) {
  BabynameDist |>
    filter(name == "Benjamin")
}
```

mdsr_table *Custom table output*

Description

Custom table output

Usage

```
mdsr_table(x, ...)

mdsr_sql_explain_table(x, ...)

mdsr_sql_keys_table(x, ...)
```

Arguments

x	A data.frame
...	arguments passed to kableExtra::kbl()

Examples

```
mdsr_table(faithful)
```

MedicareCharges *Charges to and Payments from Medicare*

Description

These data for 2011, released in May 2013, describe how much hospitals charged Medicare for various inpatient procedures, how many were performed, and how much Medicare actually paid.

Usage

```
MedicareCharges
```

Format

A data frame with 5,025 observations on the following 4 variables.

drg Code for the Diagnosis Related Group: a character string that looks like a number.

stateProvider the state providing the care.

num_charges the total number of charges.

mean_charge the average charge for each drg across each state

Details

These data are part of a set with `DiagnosisRelatedGroup`, which gives a description of the medical procedure associated with each DRG, and MedicareProviders, which translates `idProvider` into a name, address, state, Zip, etc..

These data have been pre-aggregated by state.

Source

Data from the Centers for Medicare and Medicaid Services. See https://data.cms.gov/provider-summary-by-type-of-medicare-inpatient-hospitals/

See Also

MedicareProviders

Examples

```
data(MedicareCharges)
```

MedicareProviders *Medicare Providers*

Description

Name and location data for the medicare providers in the MedicareCharges data table.

Usage

```
MedicareProviders
```

Format

A data frame with 3337 observations on the following 7 variables.

idProvider a unique number assigned to each provider

nameProvider Name of the provider. (text string)

addressProvider Street address of the provider. (text string)

cityProvider The name of the city in which the provider is located. (factor)

stateProvider The two-letter postal code of the state in which the provider is located. (factor)

zipProvider The provider's ZIP code. (factor)

referralRegion An identifier for the region serviced by the provider.

Details

This data table is related to MedicareCharges data.

Source

Extracted from the highly repetitive table provided by the Centers for Medicare and Medicaid Services. See https://data.cms.gov/provider-summary-by-type-of-service/medicare-inpatient-hospitals/

See Also

MedicareCharges

Examples

```
data(MedicareProviders)
```

`Minneapolis2013` *Ballots in the 2013 Mayoral election in Minneapolis*

Description

The choices marked on each (valid) ballot for the election, which was run using a rank-choice, instant runoff system.

Usage

```
Minneapolis2013
```

Format

A data frame with 80,101 observations on the following 5 variables. All are stored as character strings.

Precinct Precincts are sub-divisions within Wards

First The voter's first choice

Second The voter's second choice

Third The voter's third choice

Ward The city is divided spatially into districts or 'wards'. These are further subdivided into precincts.

Details

Ballot information for the 2013 Minneapolis Mayoral election, which was run as a rank-choice election. In rank-choice, a voter can indicate first, second, and third choices. If a voter's first choice is eliminated (by being last in the count across voters), the second choice is promoted to that voter's first choice, and similarly third -> second. Eliminations are done successively until one candidate has a majority of the first-choice votes.

Source

Ballot data from the Minneapolis city government: https://vote.minneapolismn.gov/results-data/election-results/2013/mayor/

References

Description of ranked-choice voting: https://vote.minneapolismn.gov/ranked-choice-voting/

A Minnesota Public Radio story about the election ballot tallying process: https://www.mprnews.org/2013/11/22/politics/ranked-choice-vote-count-programmers/

The Wikipedia article about the election: https://en.wikipedia.org/wiki/2013_Minneapolis_mayoral_election

Examples

```
data(Minneapolis2013)
```

MLB_teams *Data about recent major league baseball teams*

Description

A dataset containing information about Major League Baseball teams from 2008-2014.

Usage

```
MLB_teams
```

Format

A tibble::tbl_df object.

yearID season in which the team played

teamID the team's three character identifier

lgID the league in which the team played

W number of wins

L number of losses

WPct winning percentage

attendance number of fans in attendance

normAttend number of fans in attendance, relative to the team with the highest attendance in this sample (the 2008 New York Yankees)

payroll the sum of the salaries of the players on each team. Note that this number is only an estimate of the actual team payroll – and may not even be a very good one. Salaries are accumulated from Lahman::Salaries

metroPop the size of the team's home city's metropolitan population, according to Wikipedia and the 2010 US Census

name the full name of the team

Source

The Lahman::Teams table from Lahman::Lahman-package and https://en.wikipedia.org/wiki/List_of_Metropolitan_Statistical_Areas

See Also

Lahman::Teams

NCI60_tiny *Gene expression in cancer*

Description

The data come from a National Cancer Institute study of gene expression in cell lines drawn from various sorts of cancer.

Usage

```
NCI60_tiny

Cancer
```

Format

The expression data, NCI60_tiny is a dataframe of 41,078 gene probes (rows) and 60 cell lines (columns). The first column, Probe gives the name of the Agilent microarray probe. Each of the remaining columns is named for a cell line. The value is the log-2 expression associated with that probe for the cell line.

Probe the name of the Agilent microarray probe

For Cancer:

otherCellLine a character vector giving the name of one cell line

cellLine a character vector giving the name of another cell line

correlation the correlation between the two cell lines. See `stats::cor()`

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 1770 rows and 3 columns.

Details

Cancer gives information about each cell line.

References

- Staunton et al. (2001), *PNAS* (doi:10.1073/pnas.191368598)
- D.T. Ross et al. (2000) *Nature Genetics*, 24(3):227-234 (doi:10.1038/73432)
- CellMiner

See Also

Cancer

Examples

```
data(NCI60_tiny)
```

`ordway_birds` *Birds captured and released at Ordway, complete and uncleaned*

Description

The historical record of birds captured and released at the Katharine Ordway Natural History Study Area, a 278-acre preserve in Inver Grove Heights, Minnesota, owned and managed by Macalester College.

Usage

```
ordway_birds
```

Format

A data frame with 15,829 observations on the bird's species, size, date found, and band number.

bogus a character vector

Timestamp Timestamp indicates when the data were entered into an electronic record, not anything about the bird being described

Year a character vector

Day a character vector

Month a character vector

CaptureTime a character vector

SpeciesName a character vector

Sex a character vector

Age a character vector

BandNumber a character vector

TrapID a character vector

Weather a character vector

BandingReport a character vector

RecaptureYN a character vector

RecaptureMonth a character vector

RecaptureDay a character vector

Condition a character vector

Release a character vector

Comments a character vector

DataEntryPerson a character vector

Weight a character vector

WingChord a character vector

Temperature a character vector

RecaptureOriginal a character vector

RecapturePrevious a character vector

TailLength a character vector

Timestamp indicates when the data were entered into an electronic record, not anything about the bird being described.

Details

There are many extraneous levels of variables such as species. Part of the purpose of this data set is to teach about data cleaning.

Source

Jerald Dosch, Dept. of Biology, Macalester College: the manager of the Study Area.

References

https://www.macalester.edu/ordway/

Examples

```
ordway_birds
```

Rnw2Rmd *Convert Rnw to Rmd*

Description

Convert Rnw to Rmd

Usage

```
Rnw2Rmd(path, new_path = NULL)
```

Arguments

path	A character vector of one or more paths.
new_path	New file path. If new_path is existing directory, the file will be moved into that directory; otherwise it will be moved/renamed to the full path. Should either be the same length as path, or a single directory.

---

saratoga_houses *Saratoga Houses*

Description

Saratoga Houses

Usage

```
saratoga_houses

saratoga_codes
```

Format

A tibble with 1728 rows and 16 variables:

price ,
lot_size ,
waterfront ,
age ,
land_value ,
construction ,
air_cond ,
fuel ,
heat ,
sewer ,
living_area ,
pct_college ,
bedrooms ,
fireplaces ,
bathrooms ,
rooms

@examples saratoga_houses

An object of class spec_tbl_df (inherits from tbl_df, tbl, data.frame) with 13 rows and 3 columns.

SAT_2010 *State SAT scores from 2010*

Description

SAT results by state for 2010

Usage

```
SAT_2010
```

Format

A data.frame with 50 rows and 9 variables.

state a factor with levels for each state

expenditure average expenditure per student (in each state)

pupil_teacher_ratio pupil to teacher ratio in that state

salary teacher salary (in 2010 US $)

read state average Reading SAT score

math state average Math SAT score

write state average Writing SAT score

total state average Total SAT score

sat_pct percent of students taking SAT in that state

Details

See also the earlier mosaicData::SAT dataset.

See Also

mosaicData::SAT

save_webshot *Embedded webshot of leaflet map*

Description

Embedded webshot of leaflet map

Usage

```
save_webshot(
  map,
  path_to_img,
  overwrite = FALSE,
  vwidth = 800,
  vheight = 600,
  cliprect = "viewport",
  ...
)
```

Arguments

map	A leaflet map object
path_to_img	A path to the image file to save
overwrite	Do you want to clobber any existing file?
vwidth	Viewport width. This is the width of the browser "window".
vheight	Viewport height This is the height of the browser "window".
cliprect	Clipping rectangle. If `cliprect` and `selector` are both unspecified, the clipping rectangle will contain the entire page. This can be the string `"viewport"`, in which case the clipping rectangle matches the viewport size, or it can be a four-element numeric vector specifying the left, top, width, and height. (Note that the order of left and top is reversed from the original webshot package.) When taking screenshots of multiple URLs, this parameter can also be a list with same length as `url` with each element of the list being "viewport" or a four-elements numeric vector. This option is not compatible with `selector`.
...	arguments passed to `webshot2::webshot()`

Value

a path to a PNG file

Examples

```
## Not run:
if (require(leaflet)) {
  map <- leaflet() |>
    addTiles() |>
```

```
    addMarkers(lng = 174.768, lat = -36.852, popup = "The birthplace of R")
  save_webshot(map, tempfile())
}

## End(Not run)
```

`skim` *Custom skimmer*

Description

Custom skimmer

Usage

```
skim(data, ...)
```

Arguments

`data`	A tibble, or an object that can be coerced into a tibble.
`...`	Columns to select for skimming. When none are provided, the default is to skim all columns.

Examples

```
skim(faithful)
```

`src_scidb` *src_scidb*

Description

Connect to the scidb server on Amazon Web Services.

Usage

```
src_scidb(dbname, ...)

dbConnect_scidb(dbname, ...)

mysql_scidb(dbname, ...)
```

Arguments

`dbname`	the name of the database to which you want to connect
`...`	arguments passed to `dbplyr::src_dbi()` or `DBI::dbConnect()`

Details

This is a public, read-only account. Any abuse will be considered a hostile act.

The MariaDB server accessible via these functions is a db.t3.micro RDS instance hosted by Amazon Web Services. It is NOT a powerful server, having only 2 CPUs, 1 GB of RAM, and 20 GB of disk space. It is useful for quick, efficient and no-stress setup, but not useful for any kind of serious computing.

The `airlines` database on the server contains complete flight records for the three years between 2013 and 2015, which contains about 6 million rows annually. Thus, the `flights` table contains approximately 18 million rows. The `flights` table has several indexes, including an indices on `year`, `origin`, `dest`, `carrier`, and `tailnum`. There is also a composite index on the date (across `year`, `month`, and `day`). Please use these indexes to improve query response times.

There are two databases on this server:

- `airlines`: The structure of the database is similar to what you find in the `nycflights13` and `nycflights23` packages. See their documentation at nycflights13::flights and nycflights23::airports, for example.
- `imdb`: These data were retrieved from an old dump of the Internet Movie Database, circa 2016. Please see this ER diagram for relationships between the tables.

Value

For `src_scidb()`, a dbplyr::src_dbi object

For `dbConnect_scidb()`, a RMariaDB::MariaDBConnection object

For `mysql_scidb()`, a character vector of length 1 to be used as an `engine.ops` argument, or on the command line.

Source

- `airlines`: https://www.transtats.bts.gov/Fields.asp?gnoyr_VQ=FGJ
- `imdb`: https://developer.imdb.com/non-commercial-datasets/

See Also

`dbplyr::src_dbi()`, nycflights13::flights, nycflights23::airlines

RMariaDB::MariaDBConnection

`knitr::opts_chunk()`

Examples

```
# Connect to the database instance via `dplyr`
db_air <- src_scidb("airlines")
db_air


# Connect to the database instance via `DBI` (recommended)
db_air <- dbConnect_scidb("airlines")
db_air
```

```
# Get more information...
if (require(DBI)) {

  # About the database instance
  dbGetInfo(db_air)

  # About the available tables
  dbListTables(db_air)

  # About the variables in a particular table
  dbListFields(db_air, "flights")

  # About the indexes (using raw SQL)
  dbGetQuery(db_air, "SHOW KEYS FROM flights")
}


if (require(knitr)) {
  opts_chunk$set(engine.opts = mysql_scidb("airlines"))
}
```

`theme_mdsr` *MDSR themes*

Description

Graphical themes used in MDSR book

Usage

```
theme_mdsr(base_size = 12, base_family = "Bookman")
```

Arguments

`base_size`	base font size, given in pts.
`base_family`	base font family

Examples

```
if (require(ggplot2)) {
  p <- ggplot(mtcars, aes(x = hp, y = mpg, color = factor(cyl))) +
    geom_point() + facet_wrap(~ am) + geom_smooth()
  p + theme_grey()
  p + theme_mdsr()
 }
```

`Violations` *NYC Restaurant Health Violations*

Description

NYC Restaurant Health Violations

Usage

```
Violations

ViolationCodes

Cuisines
```

Format

A data frame with 480,621 observations on the following 16 variables.

camis unique identifier
dba full name doing business as
boro borough of New York
building building name
street street address
zipcode zipcode
phone phone number
inspection_date inspection date
action action taken
violation_code violation code, see ViolationCodes
score inspection score
grade inspection grade
grade_date grade date
record_date recording date
inspection_type inspect type
cuisine_code cuisine code, see Cuisines

A data frame with 174 observations on the following 3 variables.

violation_code a factor with many levels
critical_flag is violation critical: a factor with levels `N`, `Y`
violation_description violation description

A data frame with 84 observations on the following 2 variables.

cuisine_code a character vector
cuisine_description a character vector

Source

NYC Open Data

See Also

ViolationCodes, Cuisines

Examples

```
data(Violations)
if (require(dplyr)) {
  Violations |>
    inner_join(Cuisines, by = "cuisine_code") |>
    filter(cuisine_description == "American") |>
    arrange(grade_date) |>
    head()
 }
```

`Votes` *Votes from Scottish Parliament*

Description

Votes recorded on each ballot by each member of the Scottish Parliament in 2008 along with information about party affiliation.

Usage

```
Votes

Parties
```

Format

Votes is a data.frame with 103582 rows and 3 variables.

bill an identifier for the bill

name the name of the member of parliament

vote 1 means a vote for, -1 a vote against. 0 is an abstention.

Parties is a data.frame with 134 rows, one for each member of parliament, and 2 variables.

party the name of the political party the member belongs to

name the name of the member of parliament

An object of class `data.frame` with 134 rows and 2 columns.

Details

Almost all of the members of parliament belongs to a political party. This table identifies that party. These data were provided by Caroline Ettinger and form part of her senior honor's project at Macalester College. Prof. Andrew Beveridge supervised the thesis. Ms. Ettinger used the vote data to explore how to extract the party association of members purely from voting records. The Parties data was used to evaluate the success of methods.

`world_cities` *Cities and their populations*

Description

A list of cities

Usage

```
world_cities
```

Format

A data frame with 4,428 observations on the following 10 variables.

geoname_id integer id of record in geonames database

name name of geographical point in plain ascii characters

latitude latitude in decimal degrees (wgs84)

longitude longitude in decimal degrees (wgs84)

country ISO-3166 2-letter country code

country_region fipscode

population Population

timezone the iana timezone id

modification_date date of last modification

Source

GeoNames: http://download.geonames.org/export/dump/

Examples

```
world_cities
```

Index