Package 'modeldata'

January 25, 2023

Title Data Sets Useful for Modeling Examples

Version 1.1.0

Description Data sets used for demonstrating or testing model-related packages are contained in this package.

License MIT + file LICENSE

URL https://modeldata.tidymodels.org, https://github.com/tidymodels/modeldata

BugReports https://github.com/tidymodels/modeldata/issues

Depends R (>= 3.4)

Imports dplyr, MASS, purrr, rlang, tibble

Suggests covr, testthat (>= 3.0.0), ggplot2

Config/Needs/website tidyverse/tidytemplate, tidymodels/tidymodels

Config/testthat/edition 3

Encoding UTF-8

LazyData true

LazyDataCompression xz

RoxygenNote 7.2.3

NeedsCompilation no

Author Max Kuhn [aut, cre], Posit Software, PBC [cph, fnd]

Maintainer Max Kuhn <max@posit.co>

Repository CRAN

Date/Publication 2023-01-25 08:20:02 UTC

R topics documented:

ad_data *Alzheimer's disease data*

Description

Alzheimer's disease data

Details

Craig-Schapiro et al. (2011) describe a clinical study of 333 patients, including some with mild (but well-characterized) cognitive impairment as well as healthy individuals. CSF samples were taken from all subjects. The goal of the study was to determine if subjects in the early states of impairment could be differentiated from cognitively healthy individuals. Data collected on each subject included:

- Demographic characteristics such as age and gender
- Apolipoprotein E genotype
- Protein measurements of Abeta, Tau, and a phosphorylated version of Tau (called pTau)
- Proteinmeasurements of 124 exploratory biomarkers, and
- Clinical dementia scores

For these analyses, we have converted the scores to two classes: impaired and healthy. The goal of this analysis is to create classification models using the demographic and assay data to predict which patients have early stages of disease.

Value

ad_data a tibble

Source

Kuhn, M., Johnson, K. (2013) *Applied Predictive Modeling*, Springer.

Craig-Schapiro R, Kuhn M, Xiong C, Pickering EH, Liu J, Misko TP, et al. (2011) Multiplexed Immunoassay Panel Identifies Novel CSF Biomarkers for Alzheimer's Disease Diagnosis and Prognosis. PLoS ONE 6(4): e18850.

Examples

```
data(ad_data)
str(ad_data)
```

ames *Ames Housing Data*

Description

A data set from De Cock (2011) has 82 fields were recorded for 2,930 properties in Ames IA. This version is copies from the `AmesHousing` package but does not include a few quality columns that appear to be outcomes rather than predictors.

Details

See this links for the sources below for more information as well as `?AmesHousing::make_ames`.

For these data, the training materials typically use:

```
library(tidymodels)

set.seed(4595)
data_split <- initial_split(ames, strata = "Sale_Price")
ames_train <- training(data_split)
ames_test  <- testing(data_split)

set.seed(2453)
ames_folds<- vfold_cv(ames_train)
```

Value

`ames` a tibble

Source

De Cock, D. (2011). "Ames, Iowa: Alternative to the Boston Housing Data as an End of Semester Regression Project," *Journal of Statistics Education*, Volume 19, Number 3.

http://jse.amstat.org/v19n3/decock/DataDocumentation.txt

http://jse.amstat.org/v19n3/decock.pdf

Examples

```
data(ames)
str(ames)
```

attrition *Job attrition*

Description

Job attrition

Details

These data are from the IBM Watson Analytics Lab. The website describes the data with "Uncover the factors that lead to employee attrition and explore important questions such as 'show me a breakdown of distance from home by job role and attrition' or 'compare average monthly income by education and attrition'. This is a fictional data set created by IBM data scientists.". There are 1470 rows.

Value

`attrition` a data frame

Source

The IBM Watson Analytics Lab website https://www.ibm.com/communities/analytics/watson-analytics-blog/hr-employee-attrition/

Examples

```
data(attrition)
str(attrition)
```

biomass *Biomass data*

Description

Ghugare et al (2014) contains a data set where different biomass fuels are characterized by the amount of certain molecules (carbon, hydrogen, oxygen, nitrogen, and sulfur) and the corresponding higher heating value (HHV). These data are from their Table S.2 of the Supplementary Materials

Value

`biomass` a data frame

Source

Ghugare, S. B., Tiwary, S., Elangovan, V., and Tambe, S. S. (2013). Prediction of Higher Heating Value of Solid Biomass Fuels Using Artificial Intelligence Formalisms. *BioEnergy Research*, 1-12.

Examples

```
data(biomass)
str(biomass)
```

`bivariate` *Example bivariate classification data*

Description

Example bivariate classification data

Details

These data are a simplified version of the segmentation data contained in `caret`. There are three columns: `A` and `B` are predictors and the column `Class` is a factor with levels "One" and "Two". There are three data sets: one for training (n = 1009), validation (n = 300), and testing (n = 710).

Value

`bivariate_train`, `bivariate_test`, `bivariate_val`
: tibbles

Examples

```
data(bivariate)
str(bivariate_train)
str(bivariate_val)
str(bivariate_test)
```

`car_prices` *Kelly Blue Book resale data for 2005 model year GM cars*

Description

Kuiper (2008) collected data on Kelly Blue Book resale data for 804 GM cars (2005 model year).

Value

`car_prices`
: data frame of the suggested retail price (column `Price`) and various characteristics of each car (columns `Mileage`, `Cylinder`, `Doors`, `Cruise`, `Sound`, `Leather`, `Buick`, `Cadillac`, `Chevy`, `Pontiac`, `Saab`, `Saturn`, `convertible`, `coupe`, `hatchback`, `sedan` and `wagon`)

Source

Kuiper, S. (2008). Introduction to Multiple Regression: How Much Is Your Car Worth?, *Journal of Statistics Education*, Vol. 16 http://jse.amstat.org/jse_archive.htm#2008.

Examples

```
data(car_prices)
str(car_prices)
```

`cells` *Cell body segmentation*

Description

Hill, LaPan, Li and Haney (2007) develop models to predict which cells in a high content screen were well segmented. The data consists of 119 imaging measurements on 2019. The original analysis used 1009 for training and 1010 as a test set (see the column called `case`).

Details

The outcome class is contained in a factor variable called `class` with levels "PS" for poorly segmented and "WS" for well segmented.

The raw data used in the paper can be found at the Biomedcentral website. The version contained in `cells` is modified. First, several discrete versions of some of the predictors (with the suffix "Status") were removed. Second, there are several skewed predictors with minimum values of zero (that would benefit from some transformation, such as the log). A constant value of 1 was added to these fields: `avg_inten_ch_2`, `fiber_align_2_ch_3`, `fiber_align_2_ch_4`, `spot_fiber_count_ch_4` and `total_inten_ch_2`.

Value

`cells` a tibble

Source

Hill, LaPan, Li and Haney (2007). Impact of image segmentation on high-content screening data quality for SK-BR-3 cells, *BMC Bioinformatics*, Vol. 8, pg. 340, https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-8-340.

Examples

```
data(cells)
str(cells)
```

`check_times` *Execution time data*

Description

These data were collected from the CRAN web page for 13,626 R packages. The time to complete the standard package checking routine was collected In some cases, the package checking process is stopped due to errors and these data are treated as censored. It is less than 1 percent.

Details

As predictors, the associated package source code were downloaded and parsed to create predictors, including

- `authors`: The number of authors in the author field.
- `imports`: The number of imported packages.
- `suggests`: The number of packages suggested.
- `depends`: The number of hard dependencies.
- `Roxygen`: a binary indicator for whether Roxygen was used for documentation.
- `gh`: a binary indicator for whether the URL field contained a GitHub link.
- `rforge`: a binary indicator for whether the URL field contained a link to R-forge.
- `descr`: The number of characters (or, in some cases, bytes) in the description field.
- `r_count`: The number of R files in the R directory.
- `r_size`: The total disk size of the R files.
- `ns_import`: Estimated number of imported functions or methods.
- `ns_export`: Estimated number of exported functions or methods.
- `s3_methods`: Estimated number of S3 methods.
- `s4_methods`: Estimated number of S4 methods.
- `doc_count`: How many Rmd or Rnw files in the vignettes directory.
- `doc_size`: The disk size of the Rmd or Rnw files.
- `src_count`: The number of files in the `src` directory.
- `src_size`: The size on disk of files in the `src` directory.
- `data_count` The number of files in the `data` directory.
- `data_size`: The size on disk of files in the `data` directory.
- `testthat_count`: The number of files in the `testthat` directory.
- `testthat_size`: The size on disk of files in the `testthat` directory.
- `check_time`: The time (in seconds) to run `R CMD check` using the "r-devel-windows-ix86+x86_64‘ flavor.
- `status`: An indicator for whether the tests completed.

Data were collected on 2019-01-20.

Value

check_times a data frame

Source

CRAN

Examples

```
data(check_times)
str(check_times)
```

Chicago *Chicago ridership data*

Description

Chicago ridership data

Details

These data are from Kuhn and Johnson (2020) and contain an *abbreviated* training set for modeling the number of people (in thousands) who enter the Clark and Lake L station.

The date column corresponds to the current date. The columns with station names (Austin through California) are a *sample* of the columns used in the original analysis (for file size reasons). These are 14 day lag variables (i.e. date - 14 days). There are columns related to weather and sports team schedules.

The station at 35th and Archer is contained in the column Archer_35th to make it a valid R column name.

Value

Chicago a tibble

stations a vector of station names

Source

Kuhn and Johnson (2020), *Feature Engineering and Selection*, Chapman and Hall/CRC . https://bookdown.org/max/FES/ and https://github.com/topepo/FES

Examples

```
data(Chicago)
str(Chicago)
stations
```

concrete *Compressive strength of concrete mixtures*

Description

Yeh (2006) describes an aggregated data set for experimental designs used to test the compressive strength of concrete mixtures. The data are used by Kuhn and Johnson (2013).

Value

`concrete` a tibble

Source

Yeh I (2006). "Analysis of Strength of Concrete Using Design of Experiments and Neural Networks." *Journal of Materials in Civil Engineering*, 18, 597-604.

Kuhn, M., Johnson, K. (2013) *Applied Predictive Modeling*, Springer.

Examples

```
data(concrete)
str(concrete)
```

covers *Raw cover type data*

Description

These data are raw data describing different types of forest cover-types from the UCI Machine Learning Database (see link below). There is one column in the data that has a few difference pieces of textual information (of variable lengths).

Value

`covers` a data frame

Source

https://archive.ics.uci.edu/ml/machine-learning-databases/covtype/covtype.info

Examples

```
data(covers)
str(covers)
```

credit_data *Credit data*

Description

These data are from the website of Dr. Lluís A. Belanche Muñoz by way of a github repository of Dr. Gaston Sanchez. One data point is a missing outcome was removed from the original data.

Value

`credit_data` a data frame

Source

https://github.com/gastonstat/CreditScoring, http://bit.ly/2kkBFrk

Examples

```
data(credit_data)
str(credit_data)
```

crickets *Rates of Cricket Chirps*

Description

These data are from from McDonald (2009), by way of Mangiafico (2015), on the relationship between the ambient temperature and the rate of cricket chirps per minute. Data were collected for two species: *O. exclamationis* and *O. niveus*. The data are contained in a data frame called `crickets` with a total of 31 data points.

Value

`crickets` a tibble

Source

Mangiafico, S. 2015. "An R Companion for the Handbook of Biological Statistics." https://rcompanion.org/handbook/.

McDonald, J. 2009. *Handbook of Biological Statistics*. Sparky House Publishing.

Examples

```
data(crickets)
str(crickets)
```

drinks *Sample time series data*

Description

Sample time series data

Details

Drink sales. The exact name of the series from FRED is: "Merchant Wholesalers, Except Manufacturers' Sales Branches and Offices Sales: Nondurable Goods: Beer, Wine, and Distilled Alcoholic Beverages Sales"

Value

`drinks` a tibble

Source

The Federal Reserve Bank of St. Louis website https://fred.stlouisfed.org/series/S4248SM144NCEN

Examples

```
data(drinks)
str(drinks)
```

grants *Grant acceptance data*

Description

A data set related to the success or failure of academic grants.

Details

The data are discussed in Kuhn and Johnson (2013):

"These data are from a 2011 Kaggle competition sponsored by the University of Melbourne where there was interest in predicting whether or not a grant application would be accepted. Since public funding of grants had decreased over time, triaging grant applications based on their likelihood of success could be important for estimating the amount of potential funding to the university. In addition to predicting grant success, the university sought to understand factors that were important in predicting success."

The data ranged from 2005 and 2008 and the data spending strategy was driven by the date of the grant. Kuhn and Johnson (2013) describe:

"The compromise taken here is to build models on the pre-2008 data and tune them by evaluating a random sample of 2,075 grants from 2008. Once the optimal parameters are determined, final model is built using these parameters and the entire training set (i.e., the data prior to 2008 and the additional 2,075 grants). A small holdout set of 518 grants from 2008 will be used to ensure that no gross methodology errors occur from repeatedly evaluating the 2008 data during model tuning. In the text, this set of samples is called the 2 0 0 8 holdout set. This small set of year 2008 grants will be referred to as the test set and will not be evaluated until set of candidate models are identified."

To emulate this, `grants_other` contains the training (pre-2008, n = 6,633) and holdout/validation data (2008, n = 1,557). `grants_test` has 518 grant samples from 2008. The object `grants_2008` is an integer vector that can be used to separate the modeling with the holdout/validation sets.

Value

`grants_other,grants_test,grants_2008`
: two tibbles and an integer vector of data points used for training

Source

Kuhn and Johnson (2013). *Applied Predictive Modeling*. Springer.

Examples

```
data(grants)
str(grants_other)
str(grants_test)
str(grants_2008)
```

---

hpc_cv *Class probability predictions*

Description

Class probability predictions

Details

This data frame contains the predicted classes and class probabilities for a linear discriminant analysis model fit to the HPC data set from Kuhn and Johnson (2013). These data are the assessment sets from a 10-fold cross-validation scheme. The data column columns for the true class (`obs`), the class prediction (`pred`) and columns for each class probability (columns `VF`, `F`, `M`, and `L`). Additionally, a column for the resample indicator is included.

Value

`hpc_cv` a data frame

Source

Kuhn, M., Johnson, K. (2013) *Applied Predictive Modeling*, Springer

Examples

```
data(hpc_cv)
str(hpc_cv)
```

`hpc_data` *High-performance computing system data*

Description

Kuhn and Johnson (2013) describe a data set where characteristics of unix jobs were used to classify there completion times as either very fast (1 min or less, `VF`), fast (1–50 min, `F`), moderate (5–30 min, `M`), or long (greater than 30 min, `L`).

Value

`hpc_data` a tibble

Source

Kuhn, M., Johnson, K. (2013) *Applied Predictive Modeling*, Springer.

Examples

```
data(hpc_data)
str(hpc_data)
```

`lending_club` *Loan data*

Description

Loan data

Details

These data were downloaded from the Lending Club access site (see below) and are from the first quarter of 2016. A subset of the rows and variables are included here. The outcome is in the variable `Class` and is either "good" (meaning that the loan was fully paid back or currently on-time) or "bad" (charged off, defaulted, of 21-120 days late). A data dictionary can be found on the source website.

Value

`lending_club` a data frame

Source

Lending Club Statistics https://www.lendingclub.com/info/download-data.action

Examples

```
data(lending_club)
str(lending_club)
```

`meats` *Fat, water and protein content of meat samples*

Description

"These data are recorded on a Tecator Infratec Food and Feed Analyzer working in the wavelength range 850 - 1050 nm by the Near Infrared Transmission (NIT) principle. Each sample contains finely chopped pure meat with different moisture, fat and protein contents.

Details

If results from these data are used in a publication we want you to mention the instrument and company name (Tecator) in the publication. In addition, please send a preprint of your article to

Karin Thente, Tecator AB, Box 70, S-263 21 Hoganas, Sweden

The data are available in the public domain with no responsibility from the original data source. The data can be redistributed as long as this permission note is attached."

"For each meat sample the data consists of a 100 channel spectrum of absorbances and the contents of moisture (water), fat and protein. The absorbance is -log10 of the transmittance measured by the spectrometer. The three contents, measured in percent, are determined by analytic chemistry."

Included here are the training, monitoring and test sets.

Value

`meats` a tibble

Examples

```
data(meats)
str(meats)
```

mlc_churn *Customer churn data*

Description

A data set from the MLC++ machine learning software for modeling customer churn. There are 19 predictors, mostly numeric: `state` (categorical), `account_length` `area_code` `international_plan` (yes/no), `voice_mail_plan` (yes/no), `number_vmail_messages` `total_day_minutes` `total_day_calls` `total_day_charge` `total_eve_minutes` `total_eve_calls` `total_eve_charge` `total_night_minutes` `total_night_calls` `total_night_charge` `total_intl_minutes` `total_intl_calls` `total_intl_charge`, and `number_customer_service_calls`.

Details

The outcome is contained in a column called `churn` (also yes/no). A note in one of the source files states that the data are "artificial based on claims similar to real world".

Value

`mlc_churn` a tibble

Source

Originally at `http://www.sgi.com/tech/mlc/`

Examples

```
data(mlc_churn)
str(mlc_churn)
```

oils *Fatty acid composition of commercial oils*

Description

Fatty acid concentrations of commercial oils were measured using gas chromatography. The data is used to predict the type of oil. Note that only the known oils are in the data set. Also, the authors state that there are 95 samples of known oils. However, we count 96 in Table 1 (pgs. 33-35).

Value

`oils` a tibble

Source

Brodnjak-Voncina et al. (2005). Multivariate data analysis in classification of vegetable oils characterized by the content of fatty acids, *Chemometrics and Intelligent Laboratory Systems*, Vol. 75:31-45.

Examples

```
data(oils)
str(oils)
```

`parabolic` *Parabolic class boundary data*

Description

Parabolic class boundary data

Details

These data were simulated. There are two correlated predictors and two classes in the factor outcome.

Value

`parabolic` a data frame

Examples

```
data(parabolic)
str(parabolic)
```

`pathology` *Liver pathology data*

Description

Liver pathology data

Details

These data have the results of a *x*-ray examination to determine whether liver is abnormal or not (in the `scan` column) versus the more extensive pathology results that approximate the truth (in `pathology`).

Value

`pathology` a data frame

Source

Altman, D.G., Bland, J.M. (1994) "Diagnostic tests 1: sensitivity and specificity," *British Medical Journal*, vol 308, 1552.

Examples

```
data(pathology)
str(pathology)
```

---

pd_speech *Parkinson's disease speech classification data set*

Description

Parkinson's disease speech classification data set

Details

From the UCI ML archive, the description is "The data used in this study were gathered from 188 patients with PD (107 men and 81 women) with ages ranging from 33 to 87 (65.1 p/m 10.9) at the Department of Neurology in Cerrahpaşa Faculty of Medicine, Istanbul University. The control group consists of 64 healthy individuals (23 men and 41 women) with ages varying between 41 and 82 (61.1 p/m 8.9). During the data collection process, the microphone is set to 44.1 KHz and following the physician's examination, the sustained phonation of the vowel /a/ was collected from each subject with three repetitions."

The data here are averaged over the replicates.

Value

pd_speech a data frame

Source

UCI ML repository (data) https://archive.ics.uci.edu/ml/datasets/Parkinson%27s+Disease+Classification#,

Sakar et al (2019), "A comparative analysis of speech signal processing algorithms for Parkinson's disease classification and the use of the tunable Q-factor wavelet transform", *Applied Soft Computing*, V74, pg 255-263.

Examples

```
data(pd_speech)
str(pd_speech)
```

penguins *Palmer Station penguin data*

Description

A data set from Gorman, Williams, and Fraser (2014) containing measurements from different types of penguins. This version of the data was retrieved from Allison Horst's `palmerpenguins` package on 2020-06-22.

Value

`penguins` a tibble

Source

Gorman KB, Williams TD, Fraser WR (2014) Ecological Sexual Dimorphism and Environmental Variability within a Community of Antarctic Penguins (*Genus Pygoscelis*). PLoS ONE 9(3): e90081. doi:10.1371/journal.pone.0090081

https://github.com/allisonhorst/palmerpenguins

Examples

```
data(penguins)
str(penguins)
```

Sacramento *Sacramento CA home prices*

Description

This data frame contains house and sale price data for 932 homes in Sacramento CA. The original data were obtained from the website for the SpatialKey software. From their website: "The Sacramento real estate transactions file is a list of 985 real estate transactions in the Sacramento area reported over a five-day period, as reported by the Sacramento Bee." Google was used to fill in missing/incorrect data.

Value

`Sacramento` a tibble

Source

SpatialKey website: https://support.spatialkey.com/spatialkey-sample-csv-data/

Examples

```
data(Sacramento)
str(Sacramento)
```

scat *Morphometric data on scat*

Description

Reid (2015) collected data on animal feses in coastal California. The data consist of DNA verified species designations as well as fields related to the time and place of the collection and the scat itself. The data are on the three main species.

Value

scat a tibble

Source

Reid, R. E. B. (2015). A morphometric modeling approach to distinguishing among bobcat, coyote and gray fox scats. *Wildlife Biology*, 21(5), 254-262

Examples

```
data(scat)
str(scat)
```

sim_classification *Simulate datasets*

Description

These functions can be used to generate simulated data for supervised (classification and regression) and unsupervised modeling applications.

Usage

```
sim_classification(
  num_samples = 100,
  method = "caret",
  intercept = -5,
  num_linear = 10,
  keep_truth = FALSE
)

sim_regression(
  num_samples = 100,
  method = "sapp_2014_1",
  std_dev = NULL,
  factors = FALSE,
```

```
  keep_truth = FALSE
)

sim_noise(
  num_samples,
  num_vars,
  cov_type = "exchangeable",
  outcome = "none",
  num_classes = 2,
  cov_param = 0
)

sim_logistic(num_samples, eqn, correlation = 0, keep_truth = FALSE)

sim_multinomial(
  num_samples,
  eqn_1,
  eqn_2,
  eqn_3,
  correlation = 0,
  keep_truth = FALSE
)
```

Arguments

num_samples	Number of data points to simulate.
method	A character string for the simulation method. For classification, the single current option is "caret". For regression, values can be "sapp_2014_1", "sapp_2014_2", "van_der_laan_2007_1", or "van_der_laan_2007_2". See Details below.
intercept	The intercept for the linear predictor.
num_linear	Number of diminishing linear effects.
keep_truth	A logical: should the true outcome value be retained for the data? If so, the column name is `.truth`.
std_dev	Gaussian distribution standard deviation for residuals. Default values are shown below in Details.
factors	A single logical for whether the binary indicators should be encoded as factors or not.
num_vars	Number of noise predictors to create.
cov_type	The multivariate normal correlation structure of the predictors. Possible values are "exchangeable" and "toeplitz".
outcome	A single character string for what type of independent outcome should be simulated (if any). The default value of "none" produces no extra columns. Using "classification" will generate a `class` column with `num_classes` values, equally distributed. A value of "regression" results in an `outcome` column that contains independent standard normal values.
num_classes	When `outcome = "classification"`, the number of classes to simulate.

`cov_param` A single numeric value for the exchangeable correlation value or the base of the Toeplitz structure. See Details below.

`eqn, eqn_1, eqn_2, eqn_3` An R expression or (one sided) formula that only involves variables `A` and `B` that is used to compute the linear predictor. External objects should not be used as symbols; see the examples below on how to use external objects in the equations.

`correlation` A single numeric value for the correlation between variables `A` and `B`.

Details

Specific Regression and Classification methods:

These functions provide several supervised simulation methods (and one unsupervised). Learn more by `method`:

`method = "caret"`:
This is a simulated classification problem with two classes, originally implemented in `caret::twoClassSim()` with all numeric predictors. The predictors are simulated in different sets. First, two multivariate normal predictors (denoted here as `two_factor_1` and `two_factor_2`) are created with a correlation of about 0.65. They change the log-odds using main effects and an interaction:

```
intercept - 4 * two_factor_1 + 4 * two_factor_2 + 2 * two_factor_1 * two_factor_2
```

The intercept is a parameter for the simulation and can be used to control the amount of class imbalance.
The second set of effects are linear with coefficients that alternate signs and have a sequence of values between 2.5 and 0.25. For example, if there were four predictors in this set, their contribution to the log-odds would be

```
-2.5 * linear_1 + 1.75 * linear_2 -1.00 * linear_3 + 0.25 * linear_4
```

(Note that these column names may change based on the value of `num_linear`).
The third set is a nonlinear function of a single predictor ranging between `[0, 1]` called `non_linear_1` here:

```
(non_linear_1^3) + 2 * exp(-6 * (non_linear_1 - 0.3)^2)
```

The fourth set of informative predictors are copied from one of Friedman's systems and use two more predictors (`non_linear_2` and `non_linear_3`):

```
2 * sin(non_linear_2 * non_linear_3)
```

All of these effects are added up to model the log-odds.

`method = "sapp_2014_1"`:
This regression simulation is from Sapp et al. (2014). There are 20 independent Gaussian random predictors with mean zero and a variance of 9. The prediction equation is:

```
predictor_01 + sin(predictor_02) + log(abs(predictor_03)) +
 predictor_04^2 + predictor_05 * predictor_06 +
 ifelse(predictor_07 * predictor_08 * predictor_09 < 0, 1, 0) +
ifelse(predictor_10 > 0, 1, 0) + predictor_11 * ifelse(predictor_11 > 0, 1, 0) +
sqrt(abs(predictor_12)) + cos(predictor_13) + 2 * predictor_14 + abs(predictor_15) +
ifelse(predictor_16 < -1, 1, 0) + predictor_17 * ifelse(predictor_17 < -1, 1, 0) -
 2 * predictor_18 - predictor_19 * predictor_20
```

The error is Gaussian with mean zero and variance 9.

`method = "sapp_2014_2"`:
This regression simulation is also from Sapp et al. (2014). There are 200 independent Gaussian predictors with mean zero and variance 16. The prediction equation has an intercept of one and identical linear effects of `log(abs(predictor))`.
The error is Gaussian with mean zero and variance 25.

`method = "van_der_laan_2007_1"`:
This is a regression simulation from van der Laan et al. (2007) with ten random Bernoulli variables that have a 40% probability of being a value of one. The true regression equation is:

```
2 * predictor_01 * predictor_10 + 4 * predictor_02 * predictor_07 +
  3 * predictor_04 * predictor_05 - 5 * predictor_06 * predictor_10 +
3 * predictor_08 * predictor_09 + predictor_01 * predictor_02 * predictor_04 -
  2 * predictor_07 * (1 - predictor_06) * predictor_02 * predictor_09 -
  4 * (1 - predictor_10) * predictor_01 * (1 - predictor_04)
```

The error term is standard normal.

`method = "van_der_laan_2007_2"`:
This is another regression simulation from van der Laan et al. (2007) with twenty Gaussians with mean zero and variance 16. The prediction equation is:

```
predictor_01 * predictor_02 + predictor_10^2 - predictor_03 * predictor_17 -
  predictor_15 * predictor_04 + predictor_09 * predictor_05 + predictor_19 -
   predictor_20^2 + predictor_09 * predictor_08
```

The error term is also Gaussian with mean zero and variance 16.

`method = "hooker_2004"`:
Hooker (2004) and Sorokina *at al* (2008) used the following:

```
   pi ^ (predictor_01 * predictor_02) * sqrt( 2 * predictor_03 ) -
   asin(predictor_04) + log(predictor_05  + predictor_05) -
  (predictor_09 / predictor_10) * sqrt (predictor_07 / predictor_08) -
   predictor_02 * predictor_07
```

Predictors 1, 2, 3, 6, 7, and 9 are standard uniform while the others are uniform on `[0.6, 1.0]`. The errors are normal with mean zero and default standard deviation of 0.25.

`sim_noise()`:

This function simulates a number of random normal variables with mean zero. The values can be independent if `cov_param = 0`. Otherwise the values are multivariate normal with non-diagonal covariance matrices. For `cov_type = "exchangeable"`, the structure has unit variances and covariances of `cov_param`. With `cov_type = "toeplitz"`, the covariances have an exponential pattern (see example below).

Logistic simulation:

`sim_logistic()` provides a flexible interface to simulating a logistic regression model with two multivariate normal variables `A` and `B` (with zero mean, unit variances and correlation determined by the `correlation` argument).

For example, using `eqn = A + B` would specify that the true probability of the event was

```
prob = 1 / (1 + exp(A + B))
```

The class levels for the outcome column are `"one"` and `"two"`.

Multinomial simulation:

`sim_multinomial()` can generate data with classes `"one"`, `"two"`, and `"three"` based on the values in arguments `eqn_1`, `eqn_2`, and `eqn_3`, respectfully. Like `sim_logistic()` these equations use predictors `A` and `B`.

The individual equations are evaluated and exponentiated. After this, their values are, for each row of data, normalized to add up to one. These probabilities are them passed to `stats::rmultinom()` to generate the outcome values.

References

Van der Laan, M. J., Polley, E. C., & Hubbard, A. E. (2007). Super learner. *Statistical applications in genetics and molecular biology*, 6(1). DOI: 10.2202/1544-6115.1309.

Sapp, S., van der Laan, M. J., & Canny, J. (2014). Subsemble: an ensemble method for combining subset-specific algorithm fits. *Journal of applied statistics*, 41(6), 1247-1259. DOI: 10.1080/02664763.2013.864263

Hooker, G. (2004, August). Discovering additive structure in black box functions. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 575-580). DOI: 10.1145/1014052.1014122

Sorokina, D., Caruana, R., Riedewald, M., & Fink, D. (2008, July). Detecting statistical interactions with additive groves of trees. In *Proceedings of the 25th international conference on Machine learning* (pp. 1000-1007). DOI: 10.1145/1390156.1390282

Examples

```
set.seed(1)
sim_regression(100)
sim_classification(100)

# Flexible logistic regression simulation
if (rlang::is_installed("ggplot2")) {
  library(dplyr)
  library(ggplot2)

  sim_logistic(1000, ~ .1 + 2 * A - 3 * B + 1 * A *B, corr = .7) %>%
    ggplot(aes(A, B, col = class)) +
    geom_point(alpha = 1/2) +
    coord_equal()

  f_xor <- ~ 10 * xor(A > 0, B < 0)
  # or
  f_xor <- rlang::expr(10 * xor(A > 0, B < 0))

  sim_logistic(1000, f_xor, keep_truth = TRUE) %>%
    ggplot(aes(A, B, col = class)) +
    geom_point(alpha = 1/2) +
    coord_equal() +
    theme_bw()
}

## How to use external symbols:
```

```
a_coef <- 2
# splice the value in using rlang's !! operator
lp_eqn <- rlang::expr(!!a_coef * A+B)
lp_eqn
sim_logistic(5, lp_eqn)

# Flexible multinomial regression simulation
if (rlang::is_installed("ggplot2")) {

}
```

`small_fine_foods` *Fine foods example data*

Description

Fine foods example data

Details

These data are from Amazon, who describe it as "This dataset consists of reviews of fine foods from amazon. The data span a period of more than 10 years, including all ~500,000 reviews up to October 2012. Reviews include product and user information, ratings, and a plaintext review."

A subset of the data are contained here and are split into a training and test set. The training set sampled 10 products and retained all of their individual reviews. Since the reviews within these products are correlated, we recommend resampling the data using a leave-one-product-out approach. The test set sampled 500 products that were not included in the training set and selected a single review at random for each.

There is a column for the product, a column for the text of the review, and a factor column for a class variable. The outcome is whether the reviewer gave the product a 5-star rating or not.

Value

`training_data,testing_data`
: tibbles

Source

https://snap.stanford.edu/data/web-FineFoods.html

Examples

```
data(small_fine_foods)
str(training_data)
str(testing_data)
```

Smithsonian *Smithsonian museums*

Description

Geocodes for the Smithsonian museums (circa 2018).

Value

`Smithsonian` a tibble

Source

https://en.wikipedia.org/wiki/List_of_Smithsonian_museums

Examples

```
data(Smithsonian)
str(Smithsonian)
```

solubility_test *Solubility predictions from MARS model*

Description

Solubility predictions from MARS model

Details

For the solubility data in Kuhn and Johnson (2013), these data are the test set results for the MARS model. The observed solubility (in column `solubility`) and the model results (`prediction`) are contained in the data.

Value

`solubility_test`
a data frame

Source

Kuhn, M., Johnson, K. (2013) *Applied Predictive Modeling*, Springer

Examples

```
data(solubility_test)
str(solubility_test)
```

stackoverflow *Annual Stack Overflow Developer Survey Data*

Description

Annual Stack Overflow Developer Survey Data

Details

These data are a collection of 5,594 data points collected on developers. These data could be used to try to predict who works remotely (as used in the source listed below).

Value

`stackoverflow` a tibble

Source

Julia Silge, *Supervised Machine Learning Case Studies in R*

https://supervised-ml-course.netlify.com/chapter2

Raw data: https://insights.stackoverflow.com/survey/

Examples

```
data(stackoverflow)
str(stackoverflow)
```

tate_text *Tate Gallery modern artwork metadata*

Description

Metadata such as artist, title, and year created for recent artworks owned by the Tate Gallery. Only artworks created during or after 1990 are included, and the metadata source was last updated in 2014. The Tate Gallery provides these data but requests users to be respectful of their guidelines for use.

Value

`tate_text` a tibble

Source

- https://github.com/tategallery/collection
- https://www.tate.org.uk/

Examples

```
data(tate_text)
str(tate_text)
```

`two_class_dat` *Two class data*

Description

Two class data

Details

There are artifical data with two predictors (`A` and `B`) and a factor outcome variable (`Class`).

Value

`two_class_dat` a data frame

Examples

```
data(two_class_dat)
str(two_class_dat)
```

`two_class_example` *Two class predictions*

Description

Two class predictions

Details

These data are a test set form a model built for two classes ("Class1" and "Class2"). There are columns for the true and predicted classes and column for the probabilities for each class.

Value

`two_class_example`
a data frame

Examples

```
data(two_class_example)
str(two_class_example)
```

wa_churn *Watson churn data*

Description

Watson churn data

Details

These data were downloaded from the IBM Watson site (see below) in September 2018. The data contain a factor for whether a customer churned or not. Alternatively, the `tenure` column presumably contains information on how long the customer has had an account. A survival analysis can be done on this column using the `churn` outcome as the censoring information. A data dictionary can be found on the source website.

Value

`wa_churn` a data frame

Source

IBM Watson Analytics https://ibm.co/2sOvyvy

Examples

```
data(wa_churn)
str(wa_churn)
```

Index