Package 'ailm'

March 10, 2025

Type Package

Title Foundations of linear modeling at the age of AI

Version 0.0.1

Author Xu Liu [aut,cre]

Maintainer Xu Liu <liu.xu@sufe.edu.cn>

Description Data sets used in the book ``Foundations of linear modeling at the age of AI".

License GPL (>=2)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Depends R (>= 3.5.0)

Imports Matrix, glmnet, glmtrans, rrpack, Renvlp, dplyr, gglasso, RidgeVar, hdtrd, rbs

Remotes xliusufe/RidgeVar, xliusufe/hdtrd, xliusufe/rbs

NeedsCompilation yes

Repository github

URL https://github.com/xliusufe/ailm

Contents

`ais` *Australian Institute of Sports (AIS) data*

Description

Physical measurements and blood measurements from high performance athletes at the AIS. The dataset contains 202 observations with 13 variables.

Usage

```
data(ais)
```

Arguments

`sex`	The sex of the athlete: F means female, and M means male.
`sport`	The sport of the athlete; one of BBall (basketball), Field, Gym (gymnastics), Netball, Rowing, Swim, T400m (track, further than 400m), Tennis, TSprnt (track sprint events), WPolo (waterpolo).
`lbm`	Lean body mass, in kg.
`ht`	Height, in cm.
`wt`	Weight, in kg.
`bmi`	Body mass index, in kg per metre-squared.
`ssf`	Sum of skin folds.
`rbc`	Red blood cell count, in 10^{12} per litre.
`wbc`	White blood cell count, in 10^{12} per litre.
`hct`	Hematocrit, in percent.
`hgb`	Hemoglobin concentration, in grams per decilitre.
`ferr`	Plasma ferritins, in ng per decilitre.
`pbf`	Percentage body fat.

Details

The data give measurements from high-performance athletes from the Australian Institute of Sport (AIS), for 202 athletes (102 males; 100 females) on 13 variables. Telford and Cunningham (1991) provide more information on how the data were collected.

Source

http://www.statsci.org/data/ or R package GLMsData

References

Telford, R. D., & Cunningham, R. B. (1991). Sex, sport, and body-size dependency of hematology in highly trained athletes. *Medicine and Science in Sports and Exercise*, **23**(7), 788-794.

Examples

```
data(ais)
model <- lm(hgb ~ lbm + bmi + pbf, data = ais)
summary(model)


library(Renvlp)
data(ais)
ais$sex = as.numeric(ais$sex)
ais$sport = as.numeric(ais$sport)
y_col = c("rbc", "wbc", "lbm", "bmi")
x_col = c("sex", "sport", "ht")
Y = as.matrix(ais[, y_col])
X = as.matrix(ais[, x_col])
Y = scale(Y)
X = scale(X)
set.seed(123)
u_hat <- u.env(X, Y)$u.bic
model <- env(X, Y, u_hat)
print(model)
```

---

babblers *Feeding rates of babblers*

Description

The daily individual feeding rates of chestnut-crowned babblers. The dataset contains 97 observations on 8 variables.

Usage

```
data(babblers)
```

Arguments

obstime	The length of observation (in decimal hours); a numeric vector.
sex	The sex of the bird; one of f (female) or m (male).
age	The age of non-breeding group members; one of adult or yearling.
relatedness	The pedigree-based relatedness to the brood; one of 0.5 (first-order relatives); 0.25 (second-order relatives) or 0 (more distant relatives).
chickage	The age of the brood, in days; a numeric vector.
broodsize	The size of the brood; a numeric vector.
unitsize	The number of individuals in the unit; a numeric vector.
feedingrate	The daily individual feeding rates, in feeds per hour; a numeric vector.

Details

The data relate to a population of colour-ringed population of chestnut-crowned babblers in an area of the University of New South Wales Arid Zone Research Station, (Fowlers Gap, western New South Wales, Australia). The study determined whether, where and how often non-breeding group members contributed to providing for nestlings by monitoring the visit rate of tagged birds during 2007 and 2008. These data are extracted from a larger data set, extracted so that there is one (randomly chosen) observation for each individual bird.

Source

R package GLMsData

References

Browning, L. E., Patrick, S. C., et al. (2012). Kin selection, not group augmentation, predicts helping in an obligate cooperatively breeding bird. *Proceedings of the Royal Society B: Biological Sciences*, **279**(1743), 3861-3869.

Examples

```
data(babblers)
model = lm(feedingrate ~ ., data = babblers)
summary(model)
```

---

birthweight *Birth weight data*

Description

The birthweight dataset contains measurements on infants' birth weights along with various maternal risk factors.

Usage

```
data(birthweight)
```

Arguments

low	A binary variable indicating whether the birth weight is low (i.e., below 2500 grams; 1 = low, 0 = normal).
age	Age of the mother in years.
lwt	Weight of the mother (in pounds) at the last menstrual period.
race	Race of the mother (1 = White, 2 = Black, 3 = Other).
smoke	Smoking status during pregnancy (1 = Yes, 0 = No).
ptl	Number of previous premature labors.
ht	History of hypertension (1 = Yes, 0 = No).
ui	Presence of uterine irritability (1 = Yes, 0 = No).
ftv	Number of physician visits during the first trimester.
bwt	Birth weight in grams (the response variable).

Details

This dataset originates from a study examining risk factors associated with low birth weight in a cohort of 189 infants.

Source

Venables, W. N., & Ripley, B. D. (2002). *Modern Applied Statistics with S*. Springer.

References

Hosmer, D. W., & Lemeshow, S. (1989). *Applied Logistic Regression*. John Wiley & Sons.

Examples

```
library(dplyr)
data(birthweight)
attach(birthweight)
cols_factor = c("race", "ptl", "ftv")
birthweight[cols_factor] = lapply(birthweight[cols_factor], as.factor)
birth_dummy = model.matrix(~ race + ptl + ftv - 1, birthweight)[, -1]
```

---

birthwt_augmented *Augmented Birth Weight Data*

Description

The birthwt_augmented dataset is an enhanced version of the original birthweight dataset. In this augmented dataset, polynomial transformations of continuous predictors and dummy coding of categorical predictors have been applied to capture potential nonlinear relationships and interactions.

Usage

```
data(birthwt_augmented)
```

Format

A data frame with 189 observations on 21 variables. The variables are:

age1 First-order polynomial term for mother's age.

age2 Second-order polynomial term for mother's age.

age3 Third-order polynomial term for mother's age.

lwt1 First-order polynomial term for mother's weight (at last menstrual period).

lwt2 Second-order polynomial term for mother's weight.

lwt3 Third-order polynomial term for mother's weight.

race2 Dummy variable for race (with race1 as the reference group).

race3 Dummy variable for race.

ptl1 Dummy variable for the number of previous premature labors.

ptl2 Dummy variable for the number of previous premature labors.

ptl3 Dummy variable for the number of previous premature labors.

ftv1 Dummy variable for the number of physician visits during the first trimester.

ftv2 Dummy variable for the number of physician visits during the first trimester.

ftv3 Dummy variable for the number of physician visits during the first trimester.

ftv4 Dummy variable for the number of physician visits during the first trimester.

ftv6 Dummy variable for the number of physician visits during the first trimester.

low Binary variable indicating low birth weight (1 if birth weight is below 2500 grams, 0 otherwise).

smoke Smoking status during pregnancy (1 = smoker, 0 = non-smoker).

ht History of hypertension (1 = yes, 0 = no).

ui Presence of uterine irritability (1 = yes, 0 = no).

bwt Birth weight in grams (response variable).

Details

This dataset is derived from the original birthweight dataset after applying data augmentation procedures. Specifically:

- The categorical variables `race`, `ptl`, and `ftv` were first converted to factors and then expanded into dummy variables using `model.matrix()`. Note that `race1` was set as the baseline group and omitted.
- A third-degree polynomial expansion was performed on the continuous variables `age` and `lwt`, generating three new features each (i.e., `age1`, `age2`, `age3` and `lwt1`, `lwt2`, `lwt3`).
- The resulting polynomial features, dummy variables, and the remaining variables (after removing the original `age`, `lwt`, and factor versions of `race`, `ptl`, and `ftv`) were combined to form the final augmented dataset.

The final dataset has dimensions of 189 observations and 21 variables.

Source

Venables, W. N., & Ripley, B. D. (2002). *Modern Applied Statistics with S*. Springer.

References

Hosmer, D. W., & Lemeshow, S. (1989). *Applied Logistic Regression*. John Wiley & Sons.

Examples

```
library(gglasso)
data(birthwt_augmented)
x = as.matrix(birthwt_augmented[, -21])
y = birthwt_augmented$bwt
group = c(rep(1:2, each = 3), rep(3, 2), rep(4, 3), rep(5, 5), 6:9)
set.seed(12345)
fit_gglasso = cv.gglasso(x, y, group, loss = "ls", pred.loss = "L2")
```

`boston` *Boston housing data*

Description

The dataset contains information on housing values in suburbs of Boston, including various attributes such as crime rate, property tax, and average number of rooms.

Usage

```
data(boston)
```

Arguments

`crim`	Per capita crime rate by town.
`zn`	Proportion of residential land zoned for large lots (over 25,000 square feet).
`indus`	Proportion of non-retail business acres per town.
`chas`	Charles River dummy variable (1 if tract bounds river; 0 otherwise).
`nox`	Nitrogen oxide concentration (parts per 10 million).
`rm`	Average number of rooms per dwelling.
`age`	Proportion of owner-occupied units built before 1940.
`dis`	Weighted distance to employment centers in Boston.
`rad`	Index of accessibility to radial highways.
`tax`	Full-value property tax rate per $10,000.
`ptratio`	Pupil-teacher ratio by town.
`b`	Proportion of residents of African American descent.
`lstat`	Percentage of lower status population.
`medv`	Median value of owner-occupied homes in $1000s.

Details

The dataset is derived from the Boston Housing dataset, originally from the UCI Machine Learning Repository. It contains data collected from 506 census tracts in Boston, providing a snapshot of various housing-related features, which can be used for regression and classification tasks in machine learning.

Source

https://lib.stat.cmu.edu/datasets/boston or R package MASS

References

Harrison, D., & Rubinfeld, D. L. (1978). Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, **5**(1), 81-102.

Examples

```
data(boston)
model = lm(medv ~ ., data = boston)
summary(model)
```

`breastcancer` *Breast cancer data*

Description

The dataset contains gene expression and gene copy number information from 89 subjects.

Usage

```
data(breastcancer)
```

Arguments

`dna`	Copy number variation (CNV) data representing genomic DNA amplification or deletion events in tumor samples.
`rna`	Gene expression profiles measured via RNA transcript levels (e.g., microarray or RNA-seq data).
`chrom`	Chromosome numbers (1-22, X, Y) corresponding to the genomic location of the measured genes.
`nuc`	Nucleotide positions (start/end coordinates) of genes or probes on the chromosome (e.g., hg18/hg19 reference).
`gene`	Unique gene identifiers (e.g., Entrez Gene IDs or probe IDs) linked to genomic features.
`genenames`	Official gene symbols or names (e.g., BRCA1, ERBB2) standardized by HUGO Gene Nomenclature Committee (HGNC).
`genechr`	Chromosomal mapping information for each gene (e.g., "chr17" for TP53).
`genedesc`	Brief functional descriptions of genes (e.g., "tumor protein p53" or "estrogen receptor 1").
`genepos`	Genomic coordinates of genes (e.g., cytoband or base-pair positions like "17q21.31").

Details

The dataset is derived from molecular bioinformatics data obtained from breast cancer tissue samples treated according to the standard of care between 1989 and 1997. It primarily consists of gene expression profiles and copy number variation data across 22 chromosomal pairs in tumor tissue samples from 89 breast cancer patients. For a detailed explanation of this dataset, please refer to Chin et al. (2006).

Source

http://icbp.lbl.gov/breastcancer/ or R package PMA

References

Chin, K., DeVries, S., et al. (2006). Genomic and transcriptional aberrations linked to breast cancer pathophysiologies. *Cancer cell*, **10**(6), 529-541.

Examples

```
library(glmnet)
data(breastcancer)
dna = breastcancer$dna[breastcancer$chrom==21,]
rna = breastcancer$rna[which(breastcancer$genechr==21),]
y = dna[1,]
x = t(rna)
set.seed(100)
fit_ridge = cv.glmnet(x,y,alpha = 0)
coef(fit_ridge, s = "lambda.min")
fit_lasso = cv.glmnet(x,y,alpha = 1)
coef(fit_lasso, s = "lambda.min")


library(rrpack)
data(breastcancer)
X = t(breastcancer$dna[breastcancer$chrom==21,])
Y = t(breastcancer$rna[which(breastcancer$genechr==21),])
set.seed(123)
model <- rssvd(Y = Y, X = X, nrank = 1, ic.type = "BIC")
summary(model)
U <- model$U
V <- model$V
D <- model$D
B_approx <- D * U
```

CHARLS *China Health and Retirement Longitudinal Study (CHARLS) data of Hebei, Shandong, and Fujian provinces.*

Description

The dataset contains the CHARLS data collected in Hebei, Shandong, and Fujian provinces.

Usage

```
data(CHARLS)
```

Arguments

hebei	The CHARLS data of Hebei province. A matrix with 257 rows and 50 columns including the response y and the 49 covariates v1,...,v49.
shandong	The CHARLS data of Shandong province. A matrix with 413 rows and 50 columns including the response y and the 49 covariates v1,...,v49.
fujian	The CHARLS data of Fujian province. A matrix with 167 rows and 50 columns including the response y and the 49 covariates v1,...,v49.

Details

The response y is the annual support income of elderly people, and the covariates v1,...,v49 denote the covariates "gender", "age", "marital status", "live alone", "live with a spouse", "live with children", "live with other members such as parents", "health status", "pension income", "whether to

receive a pension", "the number of surviving children", "wage income per household", "net operating income per household", "net transfer income per household", "the number of children with college degree or above", "the number of children earning over 10000 CNY each year", "emotional comfort", "the number of household members", "the number of deceased biological children", "the number of surviving adopted children", "the number of surviving sons", "financial support for parents", "financial support for other relatives", "net financial support received from other relatives", "the number of types of disability", "have a chronic illness", "whether to receive a retirement pension", "retirement pension income", "new rural pension income", "all other pension income", "pension income of elderly households", "the total financial assets of the elderly and their spouses", "the wage income of the main members of the household", "government subsidies for individual families in the past year", "government subsidies for the main members of the family", "the wage income of family's other members in the past year", "government subsidies for other members of the family", "total government subsidies for families", "government transfer income for households", "net household income excludes private transfer income", "net household income", "net household income per capita", "other net private transfer income of the elderly", "the family shared income received by the elderly", "whether to complete high school education", "annual net income from other sources", "financial support for children". For a detailed explanation of this dataset, please refer to Ren et al. (2006).

Source

https://charls.charlsdata.com/pages/Data/2015-charls-wave4/zh-cn.html

References

Ren, P., Liu, X., Zhang, X., Zhan, P., & Qiu, T. (2024). Integrative analysis of high-dimensional quantile regression with contrasted penalization. *Journal of Applied Statistics*, 1-17.

Examples

```
library(glmnet)
data(CHARLS)
data_hebei = CHARLS$hebei
y = data_hebei$y
x = data_hebei[,-1]
x = matrix(unlist(x),nrow = nrow(x))
fit_lasso = cv.glmnet(x,y,alpha = 1)
coef(fit_lasso,s = "lambda.min")
```

---

coloncancer *Colon cancer data*

Description

The dataset contains 62 observations of 2000 gene expressions and a one-dimensional response. 62 samples (40 tumor samples, 22 normal samples) from colon-cancer patients were analyzed with an Affymetrix oligonucleotide array.

Usage

```
data(coloncancer)
```

Arguments

x	A matrix with 62 rows and 2000 columns, where each column represents expression values of a gene.
y	Tissue identity, a categorical variable with two levels: "n" for normal tissue and "t" for tumor tissue.

Details

The dataset originates from the microarray experiment conducted by Alon et al. (1999), which measured the gene expression levels of 6500 human genes in 40 tumor samples and 22 normal colon tissue samples. Out of 6500 genes, 2000 were selected based on the confidence in the measured expression levels (for details, see Alon et al. (1999)).

Source

http://microarray.princeton.edu/oncology/affydata/index.html or

https://github.com/ramhiser/datamicroarray/blob/master/data/alon.RData

References

Alon, U., Barkai, N., et al. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, **96**(12), 6745-6750.

Examples

```
data(coloncancer)
y <- coloncancer$y
x <- coloncancer$x
x_scaled <- scale(x)
pca_result <- prcomp(x_scaled, center = TRUE, scale = TRUE)
print(pca_result)
```

---

diabetes *diabetes data*

Description

The dataset records information for 422 diabetic patients. This dataset includes various health metrics that may be used to predict the progression of diabetes in patients.

Usage

```
data(diabetes)
```

Arguments

x	A matrix with 10 columns, including variables "age", "sex", "bmi" (body mass index), "map" (average blood pressure), "tc" (total serum cholesterol), "ldl" (low-density lipoproteins), "hdl" (high-density lipoproteins), "tch" (total cholesterol/HDL), "ltg" (possibly log of serum triglycerides level), "glu" (blood sugar level).
y	A numeric vector, which is a quantitative measure of disease progression one year after baseline.
x2	A matrix with 64 columns. This matrix consists of x plus certain interactions.

Details

The diabetes dataset is used to explore how various factors such as BMI and blood pressure can be used to predict diabetes progression. The dataset is derived from a study by , which is available in the "lars" package.

Source

R package lars

References

Efron, B., Hastie, T., Johnstone, I., & Tibshirani, R. (2004). Least angle regression. *Annals of Statistics*, **32**(2), 407-499.

Examples

```
library(glmnet)
data(diabetes)
fit = glmnet(diabetes$x,diabetes$y)
coef(fit, s = 1)
```

---

energy *Energy expenditure data*

Description

The energy expenditure for 104 females at rest for a 24 hour period.

Usage

```
data(energy)
```

Arguments

energy	The energy expenditure (units not given); a numeric vector.
fat	The mass of fat tissue (units not given); a numeric vector.
nonfat	The mass of fat-free tissue (units not given); a numeric vector.

Details

The data give the energy expenditure for 104 females at rest over a 24 hour period; the mass of fat and fat-free tissue was also recorded.

Note that the total mass of each subject is the sum of the fat and fat-free tissue masses.

Source

R package GLMsData

References

Jørgensen, B. (1992). Exponential dispersion models and extensions: A review. *International Statistical Review*, **60**(1), 5-20.

Examples

```
data(energy)
model <- lm(energy ~ fat, data = energy)
summary(model)
```

---

gdp *GDP growth rate data*

Description

The dataset contains GDP growth data compiled by Barro Lee. It includes 90 observations with 61 covariates.

Usage

```
data(gdp)
```

Arguments

`outcome`	Dependent variable: national growth rates in GDP per capital for the periods 1965-1975 and 1975-1985.
`x`	A list includes 61 covariates that could affect growth.

Details

The dataset is a subset of the Barro-Lee panel data, which covers 138 countries from 1950 to 2010. It includes 90 complete cases with 61 covariates, focusing on two growth periods: 1965-1975 (41 observations) and 1975-1985 (49 observations). Growth rates are calculated using the log-difference method.

Source

This version of dataset is maintained in the R package hdm.

The full data set and further details can be found at http://www.barrolee.com/ and,

https://www.bristol.ac.uk//Depts//Economics//Growth//barlee.htm.

References

Barro, R. J., & Lee, J. W. (1994). Data set for a panel of 138 countries.

Barro, R. J., & Lee, J. W. (2013). A new data set of educational attainment in the world, 1950-2010. *Journal of Development Economics*, **104**, 184-198.

Barro, R. J., & Sala-i-Martin, X. (1995). *Economic Growth*. McGraw-Hill, New York.

Examples

```
data(gdp)
mean_growth <- mean(gdp$outcome, na.rm = TRUE)
cat("Average GDP growth rate:", round(mean_growth, 3), "\n")
model <- lm(outcome ~ x$gdpsh465 + x$freeop + x$p65, data = gdp)
summary(model)


library(RidgeVar)
data(gdp)
subset <- 1:41
y <- gdp$outcome[subset]
x <- as.matrix(gdp$x[subset, ])
fit_rr <- VAR_RR(y, x)
sigma2_RR <- fit_rr$sigma2
print(sigma2_RR)
```

`glmtransbinomialdemo` *GLM trans demo data: logistic regression model*

Description

The dataset contains demo data for glmtrans, which is a simulated dataset for a logistic regression model.

Usage

```
data(glmtransbinomialdemo)
```

Arguments

`D.training`	Contains both the target and source data.
`D.training$target`	Target data, including both independent variables and the response variable.
`D.training$source`	Source data, including both independent variables and the response variable.
`D.test`	Contains the target test data.

Details

The dataset is used to demonstrate the glmtrans method, which applies transfer learning in the context of high-dimensional generalized linear models.

Source

Tian, Y., & Feng, Y. (2023). Transfer learning under high-dimensional generalized linear models. *Journal of the American Statistical Association*, **118**(544), 2684-2697.

References

Tian, Y., & Feng, Y. (2023). Transfer learning under high-dimensional generalized linear models. *Journal of the American Statistical Association*, **118**(544), 2684-2697.

Examples

```
library(glmtrans)
data(glmtransbinomialdemo)
str(glmtransbinomialdemo$D.training)
str(glmtransbinomialdemo$D.training$target)

D.training <-glmtransbinomialdemo$D.training
D.test <- glmtransbinomialdemo$D.test
fit.binomial <- glmtrans(D.training$target, D.training$source, family = "binomial")
summary(fit.binomial)
y.pred.glmtrans <- predict(fit.binomial, D.test$target$x)
```

---

glmtranslineardemo *GLM trans demo data: linear regression model*

Description

The dataset contains demo data for glmtrans, which is a simulated dataset for a linear regression model.

Usage

```
data(glmtranslineardemo)
```

Arguments

`D.training`	Contains both the target and source data.
`D.training$target`	Target data, including both independent variables and the response variable.
`D.training$source`	Source data, including both independent variables and the response variable.
`D.test`	Contains the target test data.

Details

The dataset is used to demonstrate the glmtrans method, which applies transfer learning in the context of high-dimensional generalized linear models.

Source

Tian, Y., & Feng, Y. (2023). Transfer learning under high-dimensional generalized linear models. *Journal of the American Statistical Association*, **118**(544), 2684-2697.

References

Tian, Y., & Feng, Y. (2023). Transfer learning under high-dimensional generalized linear models. *Journal of the American Statistical Association*, **118**(544), 2684-2697.

Examples

```
library(glmtrans)
data(glmtranslineardemo)
str(glmtranslineardemo$D.training)
str(glmtranslineardemo$D.training$target)

D.training <-glmtranslineardemo$D.training
D.test <- glmtranslineardemo$D.test
fit.gaussian <- glmtrans(D.training$target, D.training$source)
summary(fit.gaussian)
y.pred.glmtrans <- predict(fit.gaussian, D.test$target$x)
```

`gtexbrain` *Gtex brain data*

Description

The dataset contains gene expression data from the GTEx (Genotype-Tissue Expression) project, specifically focusing on brain tissue samples. It includes gene sequencing results from 48 different tissue types, with detailed information about gene expression levels across various tissues, including the brain and other organs.

Usage

```
data(gtexbrain)
```

Arguments

The data set includes the following tissue types:

`Adipose_Subcutaneous`
: The data of tissues named 'Adipose_Subcutaneous'.

`Adipose_Visceral_Omentum`
: The data of tissues named 'Adipose_Visceral_Omentum'.

`Adrenal_Gland`
: The data of tissues named 'Adrenal_Gland'.

`Artery_Aorta`
: The data of tissues named 'Artery_Aorta'.

`Artery_Coronary`
: The data of tissues named 'Artery_Coronary'.

`Artery_Tibial`
: The data of tissues named 'Artery_Tibial'.

`Brain_Amygdala`
: The data of tissues named 'Brain_Amygdala'.

`Brain_Anterior_cingulate_cortex_BA24`
: The data of tissues named 'Brain_Anterior_cingulate_cortex_BA24'.

`Brain_Caudate_basal_ganglia`
: The data of tissues named 'Brain_Caudate_basal_ganglia'.

`Brain_Cerebellar_Hemisphere`
: The data of tissues named 'Brain_Cerebellar_Hemisphere'.

`Brain_Cerebellum`
: The data of tissues named 'Brain_Cerebellum'.

`Brain_Cortex`
: The data of tissues named 'Brain_Cortex'.

`Brain_Frontal_Cortex_BA9`
: The data of tissues named 'Brain_Frontal_Cortex_BA9'.

`Brain_Hippocampus`
: The data of tissues named 'Brain_Hippocampus'.

`Brain_Hypothalamus`
: The data of tissues named 'Brain_Hypothalamus'.

`Brain_Nucleus_accumbens_basal_ganglia`
: The data of tissues named 'Brain_Nucleus_accumbens_basal_ganglia'.

`Brain_Putamen_basal_ganglia`
: The data of tissues named 'Brain_Putamen_basal_ganglia'.

`Brain_Spinal_cord_cervical_c-1`
: The data of tissues named 'Brain_Spinal_cord_cervical_c-1'.

`Brain_Substantia_nigra`
: The data of tissues named 'Brain_Substantia_nigra'.

`Breast_Mammary_Tissue`
: The data of tissues named 'Breast_Mammary_Tissue'.

`Cells_EBV-transformed_lymphocytes`
: The data of tissues named 'Cells_EBV-transformed_lymphocytes'.

`Cells_Transformed_fibroblasts`
: The data of tissues named 'Cells_Transformed_fibroblasts'.

`Colon_Sigmoid`
: The data of tissues named 'Colon_Sigmoid'.

`Colon_Transverse`
: The data of tissues named 'Colon_Transverse'.

`Esophagus_Gastroesophageal_Junction`
: The data of tissues named 'Esophagus_Gastroesophageal_Junction'.

`Esophagus_Mucosa`
: The data of tissues named 'Esophagus_Mucosa'.

`Esophagus_Muscularis`
: The data of tissues named 'Esophagus_Muscularis'.

`Heart_Atrial_Appendage`
: The data of tissues named 'Heart_Atrial_Appendage'.

`Heart_Left_Ventricle`
: The data of tissues named 'Heart_Left_Ventricle'.

`Liver`
: The data of tissues named 'Liver'.

`Lung`
: The data of tissues named 'Lung'.

`Minor_Salivary_Gland`
: The data of tissues named 'Minor_Salivary_Gland'.

`Muscle_Skeletal`
: The data of tissues named 'Muscle_Skeletal'.

`Nerve_Tibial`
: The data of tissues named 'Nerve_Tibial'.

`Ovary`
: The data of tissues named 'Ovary'.

`Pancreas`
: The data of tissues named 'Pancreas'.

`Pituitary` The data of tissues named 'Pituitary'.

`Prostate` The data of tissues named 'Prostate'.

`Skin_Not_Sun_Exposed_Suprapubic`
The data of tissues named 'Skin_Not_Sun_Exposed_Suprapubic'.

`Skin_Sun_Exposed_Lower_leg`
The data of tissues named 'Skin_Sun_Exposed_Lower_leg'.

`Small_Intestine_Terminal_Ileum`
The data of tissues named 'Small_Intestine_Terminal_Ileum'.

`Spleen` The data of tissues named 'Spleen'.

`Stomach` The data of tissues named 'Stomach'.

`Testis` The data of tissues named 'Testis'.

`Thyroid` The data of tissues named 'Thyroid'.

`Uterus` The data of tissues named 'Uterus'.

`Vagina` The data of tissues named 'Vagina'.

`Whole_Blood` The data of tissues named 'Whole_Blood'.

Details

This dataset contains gene expression profiles and genomic data derived from the Genotype-Tissue Expression (GTEx) project. It includes gene sequencing data for 48 tissue types, including various brain regions. The GTEx project aims to provide comprehensive data to better understand gene expression variability across tissues and how it relates to genetic variation. This resource is often used in genomics and biomedical research, helping to identify tissue-specific gene regulation and its potential implications for diseases like cancer and neurological disorders.

Source

Genotype-Tissue Expression (GTEx) project, available at: https://www.gtexportal.org/home/

References

Li, S., Cai, T. T., & Li, H. (2022). Transfer learning for high-dimensional linear regression: Prediction, estimation and minimax optimality. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, **84**(1), 149-173.

Examples

```
library(hdtrd)
data(gtexbrain)

amygdala_data <- gtexbrain[["Brain_Amygdala"]]
jam2_col <- which(colnames(amygdala_data) == "JAM2")

if (length(jam2_col) == 0) {
 stop("JAM2 gene not found in Brain_Amygdala data.")
}

target_Y <- amygdala_data[, jam2_col]
target_X <- amygdala_data[, -jam2_col]

source_data_list <- lapply(setdiff(names(gtexbrain), "Brain_Amygdala"), function(tissue) {
 tissue_data <- gtexbrain[[tissue]]
```

```
 jam2_col <- which(colnames(tissue_data) == "JAM2")

 if (length(jam2_col) == 0) {
 warning(paste("JAM2 gene not found in", tissue, "data. Skipping this tissue."))
 return(NULL)
 }
 Y <- tissue_data[, jam2_col]
 X <- tissue_data[, -jam2_col]
 list(Y = Y, X = X)
})

fit_translasso <- translasso(
 target = list(Y = target_Y, X = target_X),
 source = source_data_list,
 idtrans = seq_along(source_data_list)
)

print(fit_translasso$beta)
```

---

hcrabs *Males attached to female horseshoe crabs*

Description

The number of male crabs attached to female horseshoe crabs. The dataset contains 173 observations with 5 variables.

Usage

```
data(hcrabs)
```

Arguments

col	The color of the female; a factor with levels LM (light medium), M (medium), DM (dark medium) or D (dark).
spine	The spine condition; a factor with levels BothOK, OneOK or NoneOK.
width	The carapace width of the female crab in cm; a numeric vector.
wt	The weight of the female crab in grams; a numeric vector.
sat	The number of male crabs attached to the female; a numeric vector.

Details

The data come from an observational study of nesting horseshoe crabs (Brockmann, 1996; p. 4).

Source

R package GLMsData

References

Brockmann, H. J. (1996). Satellite male groups in horseshoe crabs, Limulus polyphemus. *Ethology*, **102**(1), 1-21.

Examples

```
data(hcrabs)
hcrabs$col <- as.integer(hcrabs$col)
hcrabs$spine <- as.integer(hcrabs$spine)
df <- scale(hcrabs, center = FALSE)
y <- as.matrix(df[,5])
x <- df[,1:4]
model <- lm(y ~ x)
summary(model)
```

---

leukemia *Leukemia Gene Expression Data*

Description

The leukemia dataset contains gene expression measurements of 7129 genes collected from a total of 72 leukemia patients. In the full dataset, 47 patients have Acute Lymphocytic Leukemia (ALL) and 25 patients have Acute Myelogenous Leukemia (AML). For practical model training and validation, the data is split into two parts: a training set and a test set.

Usage

```
data(leukemia)
```

Arguments

`train`	A data frame or matrix containing the training data. Each row corresponds to a sample (with 38 observations), and the first 7129 columns represent gene expression levels, with the last column indicating the class label.
`test`	A data frame or matrix containing the test data. This set consists of 34 samples with the same structure as the training data.

Details

The training set (`leukemia.train`) comprises 38 samples, while the test set (`leukemia.test`) contains 34 samples. Each dataset includes 7129 gene expression values along with a response variable indicating the leukemia subtype.

Source

The dataset is included in the `SIS` package.

References

Fan, J., & Lv, J. (2008). Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(5), 849-911.

Examples

```
library(rbs)
set.seed(123)
data(leukemia)
y = leukemia$train[, dim(leukemia$train)[2]]
x = as.matrix(leukemia$train[, -dim(leukemia$train)[2]])
model_dcsis = sisdc(y, x, d = 2, ntop = 10)
model_dcsis
```

`lime` *Small-leaved lime trees data*

Description

The data is from small-leaved lime trees grown in Russia and contains 385 observations with 4 variables.

Usage

```
data(lime)
```

Arguments

`foliage`	The foliage biomass, in kg (oven dried matter).
`dbh`	The tree diameter, at breast height, in cm.
`age`	The age of the tree, in years.
`origin`	The origin of the tree; one of Coppice, Natural, Planted.

Details

The data give measurements from small-leaved lime trees (Tilia cordata) growing in Russia.

Source

https://doi.pangaea.de/10.1594/PANGAEA.871491 or R package GLMsData

References

Schepaschenko, D., Shvidenko, A., et al. (2017). A dataset of forest biomass structure for Eurasia. *Scientific Data*, **4**(1), 1-11.

Examples

```
data(lime)
lime$origin <- as.integer(lime$origin)
df <- scale(lime, center = FALSE)
y <- as.matrix(df[,1])
x <- df[,2:4]
model <- lm(y ~ x)
summary(model)
```

`lungcap` *Lung capacity and smoking in youth*

Description

The health and smoking habits of 654 youth. The dataset contains 654 observations on 5 variables.

Usage

```
data(lungcap)
```

Arguments

`age`	The age of the subject in completed years; a numeric vector.
`fev`	The forced expiratory volume in litres, a measure of lung capacity; a numeric vector.
`ht`	The height in inches; a numeric vector.
`gender`	The gender of the subjects: a numeric vector with females coded as 0 and males as 1.
`smoke`	The smoking status of the subject: a numeric vector with non-smokers coded as 0 and smokers as 1.

Details

The data give information on the health and smoking habits of a sample of 654 youths, aged 3 to 19, in the area of East Boston during middle to late 1970s.

Source

R package GLMsData

References

Kahn, M. (2003). Data Sleuth. *STATS*, **37**, 24.

Kahn, M. (2005). An exhalent problem for teaching statistics. *The Journal of Statistical Education*, **13**(2).

Tager, I. B., Weiss, S. T., et al. (1983). Longitudinal study of the effects of maternal smoking on pulmonary function in children. *New England Journal of Medicine*, **309**(12), 699-703.

Examples

```
data(lungcap)
model = lm(fev ~ ., data = lungcap)
summary(model)
```

riboflavin *Riboflavin data set*

Description

The dataset of riboflavin production by Bacillus subtilis contains 71 observations of 4088 predictors (gene expressions) and a one-dimensional response (riboflavin production).

Usage

```
data(riboflavin)
```

Arguments

y	Log-transformed riboflavin production rate.
x	variables measuring the logarithm of the expression level of 4088 genes.

Details

This dataset was made publicly by Bühlmann et al. (2014).

Source

R package hdi

References

Bühlmann, P., Kalisch, M., & Meier, L. (2014). High-dimensional statistics with a view towards applications in biology. *Annual Review of Statistics and its Applications*, **1**, 255-278.

Examples

```
library(glmnet)
data(riboflavin)
y = riboflavin$y
x = as.matrix(riboflavin$x)
set.seed(100)
fit_lasso = cv.glmnet(x, y, alpha = 1)
coef(fit_lasso, s = "lambda.min")
```

skcm *Skin Cutaneous Melanoma (SKCM) data*

Description

The dataset contains clinical outcome measurements and high-dimensional gene expression profiles from 361 subjects with skin cutaneous melanoma.

Usage

```
data(skcm)
```

Arguments

y	A numeric vector of length 361 representing Breslow's thickness, a clinicopathologic feature of cutaneous melanoma.
gexp	A data frame with 361 rows and 2000 columns, where each column represents expression values of a gene. Gene names are provided as column names (e.g., SLC8A1, DPYD).
adj_matrix	A 2000 x 2000 adjacency matrix representing gene-gene interaction relationships in gexp. Entries are defined as: • 1: If two genes are directly interacting in the KEGG melanoma pathway (hsa05218) • 0: Otherwise (no interaction or gene not in the pathway) We obtain melanoma-related pathway data through KEGG portal: 1. Access https://www.kegg.jp/ and search "melanoma" 2. Identify pathway ID: hsa05218 (Skin Cutaneous Melanoma) 3. Contains 73 genes with curated regulatory relationships Key construction steps see Details.

Details

The dataset includes 361 samples with outcomes (Breslow's thickness measurements) and expression levels of the top 2000 most variable genes. It is derived from The Cancer Genome Atlas (TCGA) for Skin Cutaneous Melanoma (SKCM), one of the most aggressive cancer types.

adj_matrix construction follows these key steps:

1. **Pathway Data Retrieval**: Download KGML file from KEGG via:

```
download.file("https://rest.kegg.jp/get/hsa05218/kgml", "melanoma_kgml.xml")
```

2. **Graph Parsing**: Convert KGML to adjacency matrix:

```
if (!require("BiocManager", quietly = TRUE))
install.packages("BiocManager")
BiocManager::install("KEGGgraph")

library(KEGGgraph)
melanoma_kgml <- "melanoma_kgml.xml"
kegg_graph <- parseKGML2Graph(melanoma_kgml, genesOnly = TRUE)
adjacent <- as(kegg_graph, "matrix")
adj_matrix_KEGG <- adjacent + t(adjacent)
```

3. **Gene ID Conversion**: Map KEGG IDs to Gene Symbols:

```
BiocManager::install("KEGGREST")
BiocManager::install("org.Hs.eg.db")

library(KEGGREST)
library(org.Hs.eg.db)
kegg_genes <- keggLink("hsa", "hsa05218")
kegg_ids <- gsub("hsa:", "", kegg_genes)
gene_symbols <- mapIds(org.Hs.eg.db,
                    keys = kegg_ids,
                    column = "SYMBOL",
```

```
                        keytype = "ENTREZID")  # Get Symbol mapping
rownames(adj_matrix_KEGG) <- gene_symbols[match(rownames(adj_matrix_KEGG), kegg_genes)]
colnames(adj_matrix_KEGG) <- rownames(adj_matrix_KEGG)
```

4. **Matrix Embedding**: Create final adjacency matrix:

```
gene_skcm <- colnames(skcm$gexp)
p <- length(gene_skcm)
adj_matrix_skcm <- matrix(0, nrow = p, ncol = p, dimnames = list(gene_skcm, gene_skcm))
common_genes <- intersect(gene_skcm, gene_symbols)
adj_matrix_skcm[common_genes, common_genes] <- adj_matrix_KEGG[common_genes, common_genes]
```

Source

The Cancer Genome Atlas (TCGA) portal: https://tcga-data.nci.nih.gov

References

The Cancer Genome Atlas Consortium. (2015). Genomic classification of cutaneous melanoma. *Cell*, **161**(7), 1681-1696.

Tan, X., Zhang, X., Cui, Y., & Liu, X. (2024). Uncertainty quantification in high-dimensional linear models incorporating graphical structures with applications to gene set analysis. *Bioinformatics*, **40**(9), btae541.

Examples

```
data(skcm)
hist(skcm$y, main = "Distribution of Clinical Outcomes", xlab = "Outcome Value")
pca_result <- prcomp(skcm$gexp[, 1:100], scale = TRUE) # Run PCA on the top 100 genes
plot(pca_result$x[, 1:2], main = "PCA of Gene Expression Data")
```

---

translassodemo *Trans Lasso Demo Data*

Description

This dataset serves as a demo for the Trans Lasso (Transfer Lasso) method, which is designed for high-dimensional linear regression problems where data is sourced from multiple domains or datasets. The dataset includes both target and source data, as well as test data for validation.

Usage

```
data(translassodemo)
```

Arguments

`X`	The independent variables (features) in the target and source data.
`y`	The dependent variable (label or outcome) in the target and source data.
`X_test`	The independent variables (features) in the test dataset.
`y_test`	The dependent variable (label or outcome) in the test dataset.
`n.vec`	A vector indicating the sample size of each dataset (target and source data).
`beta0`	The true regression coefficients in the simulated data.
`size.A0`	The number of transferable sets in the data.

Details

This dataset is a demonstration of the Trans Lasso method, which aims to combine knowledge from multiple datasets (source and target) to improve regression models. The dataset includes both features and outcome variables from different domains, along with a simulated test set for performance evaluation. It is useful for illustrating the application of transfer learning techniques to high-dimensional regression tasks.

Source

Code adapted from Li, S., Cai, T. T., & Li, H. (2022). Transfer learning for high-dimensional linear regression: Prediction, estimation, and minimax optimality. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, **84**(1), 149-173.

References

Li, S., Cai, T. T., & Li, H. (2022). Transfer learning for high-dimensional linear regression: Prediction, estimation and minimax optimality. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, **84**(1), 149-173.

Examples

```
library(glmnet)
data(translassodemo)
y = translassodemo$y
X = translassodemo$X
set.seed(100)
#prop.re1 <- Trans.lasso(X, y, n.vec, I.til = 1:50, l1 = l1)
#print(prop.re1$beta.hat)
```

---

translassodemo2 *Trans Lasso Demo Data*

Description

This dataset serves as a demo for the Trans Lasso (Transfer Lasso) method, which is designed for high-dimensional linear regression problems where data is sourced from multiple domains or datasets. The dataset includes both target and source data, as well as test data for validation.

Usage

```
data(translassodemo2)
```

Arguments

`translassodemo2[[1]]`
: The target data including independent variables (features), the dependent variable (label or outcome) and lambda

`translassodemo2[[2]]`
: The source data including independent variables (features) and the dependent variable (label or outcome)

`translassodemo2[[3]]`
: The source data including independent variables (features) and the dependent variable (label or outcome)

`translassodemo2[[4]]`
: The source data including independent variables (features) and the dependent variable (label or outcome)

`translassodemo2[[5]]`
: The source data including independent variables (features) and the dependent variable (label or outcome)

`translassodemo2[[6]]`
: The source data including independent variables (features) and the dependent variable (label or outcome)

`translassodemo2[[7]]`
: The source data including independent variables (features) and the dependent variable (label or outcome)

Details

This dataset is a demonstration of the Trans Lasso method, which aims to combine knowledge from multiple datasets (source and target) to improve regression models. The dataset includes both features and outcome variables from different domains, along with a simulated test set for performance evaluation. It is useful for illustrating the application of transfer learning techniques to high-dimensional regression tasks.

Source

Code adapted from Li, S., Cai, T. T., & Li, H. (2022). Transfer learning for high-dimensional linear regression: Prediction, estimation, and minimax optimality. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, **84**(1), 149-173.

References

Li, S., Cai, T. T., & Li, H. (2022). Transfer learning for high-dimensional linear regression: Prediction, estimation and minimax optimality. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, **84**(1), 149-173.

Examples

```
library(glmnet)
library(hdtrd)
data(translassodemo2)
fit <- translasso(target = translassodemo2[[1]], source = translassodemo2[-1], idtrans = seq(5))
fit$beta[1:10]
```

---

uselection2020 *2020 U.S. Election Data*

Description

The data set contains election results for the 2020 U.S. presidential election, organized by state.

Usage

```
data(uselection2020)
```

Arguments

The data set includes the following states:

`Arkansas`	The election data for Arkansas.
`Georgia`	The election data for Georgia.
`Illinois`	The election data for Illinois.
`Michigan`	The election data for Michigan.
`Minnesota`	The election data for Minnesota.
`Mississippi`	The election data for Mississippi.
`North Carolina`	The election data for North Carolina.
`Virginia`	The election data for Virginia.

Details

A list of length 8, where each element is a list containing detailed election results for a specific state. Each state list has two elements: - `target`: A list of length 2 containing target data. - `source`: A list of length 47 containing source data.

Source

https://github.com/tonmcg/US_County_Level_Election_Results_08-24 and

https://www.kaggle.com/benhamner/2016-us-election.

References

Tian, Y., & Feng, Y. (2023). Transfer learning under high-dimensional generalized linear models. *Journal of the American Statistical Association*, **118**(544), 2684-2697.

Examples

```
library(glmtrans)
data(uselection2020)
str(uselection2020[['Arkansas']])
str(uselection2020[['Arkansas']]$target)
data_train <- uselection2020[['Arkansas']]
fit.binomial <- glmtrans(data_train$target, data_train$source, family = "binomial")
summary(fit.binomial)

data_train <- uselection2020[['Georgia']]
fit.binomial <- glmtrans(data_train$target, data_train$source, family = "binomial")
summary(fit.binomial)
```

Index