

# Package ‘sda123’

July 25, 2023

**Title** R-paket för kurserna Statistik och dataanalys I, II och III vid SU

**Version** 0.0.2

**Description** Funktioner för grundläggande statistik.

**License** MIT

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.2.3

**Imports** glmnet,  
mvtnorm,  
RColorBrewer,  
SUdatasets,  
cowplot,  
magrittr

**Remotes** StatisticsSU/SUdatasets

**Depends** R (>= 2.10),  
ggplot2

**LazyData** true

**Suggests** rmarkdown,  
knitr

**VignetteBuilder** knitr

## R topics documented:

arima_summary . . . . .	2
bike . . . . .	3
corr_matrix . . . . .	3
ebaycoins . . . . .	4
electricitycost . . . . .	5
ericsson . . . . .	6
lifespan . . . . .	6
logisticreg_simulate . . . . .	7
logisticreg_summary . . . . .	8

moving_average . . . . .	9
moving_average_manip . . . . .	9
moving_average_seasonal . . . . .	10
reg_crossval . . . . .	10
reg_predict . . . . .	11
reg_residuals . . . . .	12
reg_simulate . . . . .	12
reg_summary . . . . .	14
simAR1 . . . . .	15
titanic . . . . .	15
triss . . . . .	16
<b>Index</b>	<b>17</b>

---

arima_summary	<i>Summarize the estimates from an ARIMA(p,d,q) fit</i>
---------------	---

---

**Description**

Alternative to the usual summary function for arima fit.

**Usage**

```
arima_summary(arimafit)
```

**Arguments**

arimafit            an ARIMA fit from arima.

**Value**

data frame with estimates, std err, z-ratio etc

**Examples**

```
library(SUdatasets)
arimafit = arima(swedinfl$KPIF, order = c(2,0,2))
arimasumm = arima_summary(arimafit)
```

---

bike	<i>Number of daily rides for a bike share company in Washington D.C.</i>
------	--

---

**Description**

A dataset containing the number of rides per day and other attributes over the course of 2 years

**Usage**

```
bike
```

**Format**

A data frame with 731 rows and 12 variables:

**dteday** date in YYYY-MM-DD format

**season** categorical variable (1="winter", 2 = "spring", 3 = "summer", 4 = "fall")

**yr** year (0="2011", 1 = "2012")

**mnth** month from 1-12 where 1 = "January"

**holiday** binary variable for public holidays

**weekday** day of the week 0-6, 0 = "Sunday"

**workingday** binary variable for working days (=1)

**weathersit** categorical variable (1="clear", 2 = "mist", 3 = "light snow")

**temp** continuous temperature variable, normalized between 0,1

**hum** continuous humidity variable, normalized between 0,1

**windspeed** continuous windspeed variable, normalized between 0,1 ...

**Source**

<https://archive.ics.uci.edu/ml/datasets/bike+sharing+dataset>

---

corr_matrix	<i>Compute pair-wise correlations and hypothesis test</i>
-------------	---

---

**Description**

Computes pair-wise correlations between variables in a dataframe df Uses p-values to test:

H0:  $\rho = 0$

H1:  $\rho \neq 0$

**Usage**

```
corr_matrix(df)
```

**Arguments**

df                      dataframe

**Value**

list with two tables: corrs (correlations), pvals (p-values)

**Examples**

```
library(sda123)
corr_matrix(mtcars[,c("mpg", "hp", "drat", "wt")])
```

---

ebaycoins

*ebay coins auctions*

---

**Description**

The dataset contains the final price and number of bidders in 1000 eBay auctions of collectors coins (U.S. proof sets, i.e. specially packaged collectors' coins sold by the U.S. Mint) along with several auction-specific features carefully collected by a human by visual inspection of text and images. The data was collected for auctions in the time periods Nov 7 - Dec 19, 2007 and Dec 27, 2007 - Jan 22, 2008.

**Usage**

ebaycoins

**Format**

A data frame with 1000 rows and 10 variables:

**BookVal** The recommended value of the coin as reported by the Internet coin seller Golden Eagle Coins at [http:// www.goldeneaglecoin.com](http://www.goldeneaglecoin.com).

**MinorBlem** Binary, coded as 1 if the proof set had minor damage on the box or packaging according to a subjective assessment of the item using the seller's description and pictures of the auctioned object.

**MajorBlem** Binary, coded as 1 if at least one coin was missing in the package or if other major imperfections were present.

**PowerSeller** Binary, coded as 1 if the seller is ranked among the most successful sellers in terms of product sales and customer satisfaction on eBay.

**IDSeller** Binary, coded as 1 if the seller's identity has been established by cross-checking his contact information in consumer and business databases.

**Sealed** Binary, coded as 1 if the proof set is sealed in its original envelope.

**NegFeedback** Binary, coded as 1 if more than 1% of the seller's feedback scores from buyers have been negative.

**ReservePriceFrac** The seller's reservation price (lowest accepted sale price) as a fraction of the object's book value.

**NBidders** Counts. Number of bidders in the auction.

**FinalPrice** Final price of the auctioned coin including shipping costs. NaN if the item was not sold. ...

## Source

Wegmann, B. and Villani, M. (2011). Bayesian Inference in Structural Second-Price Common Value Auctions, *Journal of Business and Economic Statistics*. <https://doi.org/10.1198/jbes.2011.08289>

---

electricitycost	<i>Determinants of electricity cost for 1602 households from South Australia</i>
-----------------	--

---

## Description

Determinants of electricity cost for 1602 households from South Australia

## Usage

electricitycost

## Format

A data frame with 1602 rows and 3 variables:

**cost** annual cost of electricity for a household in Australian dollars.

**rooms** number of rooms in the house.

**people** number of usual residents in the house

**income** annual pretax household income in Australian dollars

**onlysecondary** indicator for electric secondary heating only

**waterheat** indicator for peak electric water heating

**cookel** indicator for electric cooking only

**poolfilt** indicator for pool filter

**airrev** indicator for reverse cycle air conditioning

**aircond** indicator for air conditioning

**microwave** indicator for microwave

**dish** indicator for dishwasher

**dryer** indicator for dryer ...

## Source

Bartels, R., Fiebig, D. and Plumb, M. (1996). Gas or electricity, which is cheaper? An econometric approach with application to Australian expenditure data, *The Energy Journal* 17(4): 33–58.

---

ericsson

*Daily percentage returns on Ericsson B stock*


---

### Description

This data set contains daily percentage returns on Ericsson B stock for all of year 2022

### Usage

```
ericsson
```

### Format

A data frame with 25 rows and 2 variables:

**datum** date in format YYYY-MM-DD

**avkastning** daily percentage returns  $100 * (\log(x_t) - \log(x_{t-1})) \dots$

### Source

Nasdaq Nordic [https://www.nasdaqomxnordic.com/index/historiska\\_kurser?languageId=3&Instrument=SSE101](https://www.nasdaqomxnordic.com/index/historiska_kurser?languageId=3&Instrument=SSE101).

---

lifespan

*Determinants of life expectancy in 30 countries.*


---

### Description

Determinants of life expectancy in 30 countries.

### Usage

```
lifespan
```

### Format

A data frame with 30 rows and 5 variables:

**country** Country name

**spending** Spending on health per capita in thousands of dollars per capita.

**lifespan** Life expectancy in years

**doctorvisits** average number of visits/consultations to the doctor

**gdp** gross domestic product per capita in thousands of dollars per capita. ...

**Source**

Gelman, Hill and Vehtari (2020). Regression and other stories, *Cambridge University Press*. <https://avehtari.github.io/ROS-Examples/>

OECD. <https://data.oecd.org/>

---

logisticreg\_simulate    *Simulate from a logistic regression model*

---

**Description**

Simulates a dataset with n observation from the logistic regression model

$$\Pr(y = 1|x) = \frac{1}{1 + \exp(-(\beta_0 + \beta_1 x_1 + \dots + \beta_k x_k))}$$

with covariates (x) simulated from a normal distribution with the same correlation rho\_x between all pairs of covariates. Covariate x\_j has standard deviation sigma\_x[j].

Alternatively the covariate can follow a uniform distribution.

**Usage**

```
logisticreg_simulate(
  n,
  betavect,
  intercept = TRUE,
  covdist = "normal",
  rho_x = 0,
  sigma_x = rep(1, length(betavect) - intercept)
)
```

**Arguments**

n	the number of observations in the simulated dataset.
betavect	a vector with regression coefficients c(beta_0,beta_1,...beta_k). First element is intercept if intercept = TRUE
intercept	if TRUE an intercept is added to the model.
covdist	distribution of the covariates. Options: 'normal' or 'uniform'.
rho_x	correlation among the covariates. Same for all covariate pairs.
sigma_x	vector with standard deviation of the covariates.

**Value**

dataframe with simulated data (y, X1, X2, ..., XK) (no intercept included).

## Examples

```
library(sda123)
simdata <- logisticreg_simulate(n = 500, betavect = c(1, -2, 1, 0))
glmfit <- glm(y ~ X1 + X2 + X3, data = simdata, family = binomial)
logisticreg_summary(glmfit, odds_ratio = F)
```

---

logisticreg_summary	<i>Summarize the results from a logistic regression analysis</i>
---------------------	--

---

## Description

Alternative to `summary.glm` to summarize a regression from `glm`. Prints a table similar to the one generated by SAS and Minitab.

## Usage

```
logisticreg_summary(glmobject, odds_ratio = T, param = T, conf_intervals = F)
```

## Arguments

<code>glmobject</code>	a fitted regression model from <code>glm</code> .
<code>odds_ratio</code>	TRUE if odds ratios for parameters is computed.
<code>param</code>	TRUE if parameter estimates, standard errors etc is computed.
<code>conf_intervals</code>	TRUE if confidence intervals for parameters.

## Value

list with two tables: `param`, `odds_ratio`

## Examples

```
library(sda123)
glmfit <- glm(survived ~ age + sex + firstclass, data = titanic, family = binomial)
logisticreg_summary(glmfit)
```



---

moving_average	<i>Centered moving average to smooth out a time series</i>
----------------	--

---

**Description**

Centered moving average to smooth out a time series

**Usage**

```
moving_average(y, r, plotfig = TRUE)
```

**Arguments**

y	a vector with time series data
r	the number of observations to the left of the center in the average, i.e. the function computes a 2r+1 point average.
plotfig	if TRUE then a figure is plotted with data and moving average.

**Value**

a vector of the same length as y with moving averages (NA at boundaries)

**Examples**

```
library(SUdatasets)
M = moving_average(globaltemp$temp, 2)
```

---

moving_average_manip	<i>Manipulate version of centered moving average to smooth out a time series</i>
----------------------	--

---

**Description**

Manipulate version of centered moving average to smooth out a time series

**Usage**

```
moving_average_manip(y)
```

**Arguments**

y	a vector with time series data
---	--------------------------------

**Examples**

```
library(SUdatasets)
moving_average_manip(globaltemp$temp)
```

---

moving\_average\_seasonal

*Moving average to smooth out seasonal time series*


---

### Description

Moving average to smooth out seasonal time series

### Usage

```
moving_average_seasonal(y, season, plotfig = TRUE)
```

### Arguments

y	a vector with time series data
season	season = 12 for montly, season = 4 for quarterly etc
plotfig	if TRUE then a figure is plotted with data and moving average.

### Value

a vector of the same length as y with moving averages (NA at boundaries)

### Examples

```
M = moving_average_seasonal(c(AirPassengers), season = 12)
```

---

reg\_crossval

*K-fold cross-validation of regression models estimated with lm()*


---

### Description

K-fold cross-validation of regression models estimated with lm()

### Usage

```
reg_crossval(formula, data, nfolds, obs_order = "random")
```

### Arguments

formula	an object of class "formula": a symbolic description of the model to be fitted.
data	a data frame with the data used for fitting the models.
nfolds	the number of folds in the cross-validation.
obs_order	order of the observations when splitting the data. obs_order = "random" gives a random order.

**Value**

RMSE Root mean squared prediction error on test data

**Examples**

```
library(sda123)
RMSE_CV = reg_crossval(mpg ~ hp, data = mtcars, nfolds = 4, obs_order = 1:32)
print(RMSE_CV)
```

---

reg\_predict

*Plot confidence and prediction intervals for simple linear regression*

---

**Description**

Plot confidence and prediction intervals for simple linear regression

**Usage**

```
reg_predict(formula, data, level = 0.95, conf_int_line = T, pred_interval = T)
```

**Arguments**

formula	an object of class "formula": a symbolic description of the model to be fitted.
data	a data frame with the data.
level	confidence level, default is level = 0.95
conf_int_line	if TRUE, then conf intervals for regression line are plotted.
pred_interval	if TRUE, then prediction intervals are plotted.

**Value**

plot of data with overlayed intervals

**Examples**

```
library(sda123)
reg_predict(mpg ~ hp, data = mtcars)
```

reg\_residuals

*Residual analysis mimicing the 4-in-1 plots from Minitab***Description**

Plots:

1. Normal QQ-plot
2. Residuals vs fitted values
3. Histogram and normal density fit
4. Residuals vs order.

**Usage**

reg\_residuals(lm\_object)

**Arguments**

lm\_object      a fitted regression model from lm.

**Examples**

```
library(sda123)
fit = lm(mpg ~ hp, data = mtcars)
reg_residuals(fit)
```

reg\_simulate

*Simulate from a linear regression model***Description**

Simulates a dataset with n observation from the linear regression model

$$y = \beta_0 + \beta_1 x_1 + \dots + \beta_k x_k + \epsilon$$

where the errors  $\epsilon$  have zero mean and standard deviation  $\sigma_\epsilon$ , but can follow either normal or student-t distribution. The variance can be homoscedastic or heteroscedastic with standard deviation function  $\sigma_\epsilon(x_1\gamma_1 + \dots + x_k\gamma_k)$ , where the  $(\gamma_1, \dots, \gamma_k)$  vector of variance function parameters are given by the argument heteroparams. The  $\epsilon$  can also have an AR(1) autocorrelation structure with coefficient on first lag given by the argument ar1phi. The covariates (x) are simulated from a normal distribution with the same correlation rho\_x between all pairs of covariates, and covariate  $x_j$  has standard deviation sigma\_x[j]. Alternatively the covariate can follow a uniform distribution.

**Usage**

```
reg_simulate(
  n,
  betavect,
  sigma_eps,
  intercept = TRUE,
  responsedist = "normal",
  heteroparams = NA,
  studentdf = NA,
  ar1phi = NA,
  covdist = "normal",
  rho_x = 0,
  sigma_x = rep(1, length(betavect) - intercept)
)
```

**Arguments**

n	the number of observations in the simulated dataset.
betavect	a vector with regression coefficients $c(\beta_0, \beta_1, \dots, \beta_k)$ . First element is intercept if <code>intercept = TRUE</code>
sigma_eps	stdev of epsilon (homo) or a variance function <code>sigma_eps(X %*% heteroparams)</code> with parameters <code>heteroparams</code> .
intercept	if TRUE an intercept is added to the model.
responsedist	options: 'normal' or 'student'
heteroparams	parameters in the heteroscedastic variance function
studentdf	degrees of freedom in the student-t errors
ar1phi	AR(1) coefficient on first lag for autocorrelated errors
covdist	distribution of the covariates. Options: 'normal' or 'uniform'.
rho_x	correlation among the covariates. Same for all covariate pairs.
sigma_x	vector with standard deviation of the covariates.

**Value**

dataframe with simulated data (y, X1, X2, ..., XK) (no intercept included).

**Examples**

```
library(sda123)
simdata <- reg_simulate(n = 500, betavect = c(1, -2, 1, 0), sigma_eps = 2)
lmfit <- lm(y ~ X1 + X2 + X3, data = simdata)
reg_summary(lmfit, anova = F)

# Simulate from a heteroscedastic student-t regression and detect problems with residuals
simdata <- reg_simulate(n = 500, betavect = c(1, -2, 1, 0), sigma_eps = exp, heteroparam = c(0,1,0,0), responsedist = "student")
lmfit <- lm(y ~ X1 + X2 + X3, data = simdata)
reg_residuals(lmfit)
```

```
#' # Simulate from a homoscedastic student-t regression with autocorrelated errors.
simdata <- reg_simulate(n = 500, betavect = c(1, -2, 1, 0), sigma_eps = 2, respondedist = 'student', studentdf = 4, a
lmfit <- lm(y ~ X1 + X2 + X3, data = simdata)
reg_residuals(lmfit)
```

---

reg\_summary

*Summarize the results from a regression analysis*


---

## Description

Alternative to `summary.lm` to summarize a regression from `lm`. Prints a table similar to the one generated by SAS and Minitab.

## Usage

```
reg_summary(
  lmobject,
  anova = T,
  fit_measures = T,
  param = T,
  conf_intervals = F,
  vif_factors = F
)
```

## Arguments

<code>lmobject</code>	a fitted regression model from <code>lm</code> .
<code>anova</code>	TRUE if an ANOVA table is computed.
<code>fit_measures</code>	TRUE if measures of fit ( $R^2$ etc) is computed.
<code>param</code>	TRUE if parameter estimates, standard errors etc is computed.
<code>conf_intervals</code>	TRUE if confidence intervals for parameters.
<code>vif_factors</code>	TRUE if variance inflation factors are to be printed.

## Value

list with three tables: `param`, `anova` and `fit_measures`

## Examples

```
library(sda123)
lmfit = lm(nRides ~ temp + hum + windspeed, data = bike)
regsumm = reg_summary(lmfit, anova = T, conf_intervals = T, vif_factors = T)
regsumm$param
regsumm$anova
regsumm$fit_measures
```

---

simAR1	<i>Simulate from an AR(1) process</i>
--------	---------------------------------------

---

**Description**

Simulates n observations from

$$x_t = \mu + \phi(x_{t-1} - \mu) + \epsilon, \epsilon \sim N(0, \sigma_\epsilon)$$

**Usage**

```
simAR1(n, phi = 0, mu = 0, sigma_eps = 1, epsilons = NA)
```

**Examples**

```
library(sda123)
simdata = simAR1(n = 100, phi = 0.7, sigma_eps = 1)
plot(simdata)
```

---

titanic	<i>Survival of passengers on the Titanic</i>
---------	--

---

**Description**

This data set provides information on the fate of passengers on the fatal maiden voyage of the ocean liner ‘Titanic’, summarized according to economic status (class), sex, age and survival.

NOTE: this is not the same as the dataset Titanic (note capital T) which has more observations, but also missing values.

**Usage**

```
titanic
```

**Format**

A data frame with 887 rows and 8 variables:

**name** passenger name

**survived** 0 = no, 1 = yes

**sex** male/female

**age** age of passenger

**fare** ticket cost

**firstclass** first class ticket ...

## Details

The sinking of the Titanic is a famous event, and new books are still being published about it. Many well-known facts—from the proportions of first-class passengers to the ‘women and children first’ policy, and the fact that that policy was not entirely successful in saving the women and children in the third class—are reflected in the survival rates for various classes of passenger.

These data were originally collected by the British Board of Trade in their investigation of the sinking. Note that there is not complete agreement among primary sources as to the exact numbers on board, rescued, or lost.

Due in particular to the very successful film ‘Titanic’, the last years saw a rise in public interest in the Titanic. Very detailed data about the passengers is now available on the Internet, at sites such as Encyclopedia Titanica (<https://www.encyclopedia-titanica.org/>).

## Source

Dawson, Robert J. MacG. (1995), The ‘Unusual Episode’ Data Revisited. *Journal of Statistics Education*, 3. doi: 10.1080/10691898.1995.11910499.

---

triss

---

*Winnings in the Swedish Triss lottery*


---

## Description

This data set list the number of possible winning amounts and the number of tickets in each winning class for the Swedish Triss lottery.

## Usage

triss

## Format

A data frame with 25 rows and 2 variables:

**vinst** amount in each winning class

**antal** number of tickets in each winning class

**probs** probability for each winning class ...

## Source

Svenska spel <https://www.svenskaspel.se/triss/spelguide/triss-30>.



# Index

- \* **datasets**
  - bike, [3](#)
  - ebaycoins, [4](#)
  - electricitycost, [5](#)
  - ericsson, [6](#)
  - lifespan, [6](#)
  - titanic, [15](#)
  - triss, [16](#)
- [0, 1, 3](#)
- arma\_summary, [2](#)
- bike, [3](#)
- corr\_matrix, [3](#)
- ebaycoins, [4](#)
- electricitycost, [5](#)
- ericsson, [6](#)
- lifespan, [6](#)
- logisticreg\_simulate, [7](#)
- logisticreg\_summary, [8](#)
- moving\_average, [9](#)
- moving\_average\_manip, [9](#)
- moving\_average\_seasonal, [10](#)
- reg\_crossval, [10](#)
- reg\_predict, [11](#)
- reg\_residuals, [12](#)
- reg\_simulate, [12](#)
- reg\_summary, [14](#)
- simAR1, [15](#)
- titanic, [15](#)
- triss, [16](#)