Package: konstats (via r-universe)

November 17, 2026

Title Course Materials for Statistics in Linguistics at the University of Konstanz

Version 0.1.0.9000

Description Companion package for the Statistics and Advanced Statistics courses taught by Massimiliano Canzi at the Department of Linguistics, University of Konstanz, written for their students and not meant for general use. Installing it brings in the packages of the Statistics course; install_course() adds those of Advanced Statistics, and check_setup() tests a computer for either course. lectures() lists the lectures with their data sets and slides, get_tutorial() copies the weekly exercise sheets into a project folder, and an R Markdown template starts the weekly assignments. The helper functions written in the Advanced Statistics slides are included: residual autocorrelation within trajectories, partial dependence, cross-validation by speaker and a 'ggplot2' theme for papers. The data sets of the courses are in the 'simling' package.

License MIT + file LICENSE

URL https://github.com/mcanzi/konstats, https://mcanzi.de/konstats/

BugReports https://github.com/mcanzi/konstats/issues

Depends R (>= 4.1.0)

Imports broom, cli, dplyr, effsize, ggplot2, janitor, lme4, lmerTest, palmerpenguins, purrr, rlang, simling, simr, stats, swirl, tibble, tidyverse, utils

Suggests cluster, GPArotation, ggeffects, ggrepel, ggridges, itsadug, lavaan, mclust, mgcv, patchwork, psych, ranger, rmarkdown, rpart, testthat (>= 3.0.0), tidygam

Remotes mcanzi/simling

Additional_repositories https://mcanzi.r-universe.dev

Config/testthat/edition 3

Encoding UTF-8

Language en-GB

Roxygen list(markdown = TRUE)

Config/roxygen2/version 8.1.1

Config/pak/sysreqs
cmake libfontconfig1-dev libfreetype6-dev libfribidi-dev make libharfbuzz-dev libicu-dev libjpeg-dev libpng-dev libtiff-dev libuv1-dev libwebp-dev libxml2-dev libssl-dev libx11-dev zlib1g-dev

Repository https://mcanzi.r-universe.dev

Date/Publication 2026-11-17 08:21:07 UTC

RemoteUrl https://github.com/mcanzi/konstats

RemoteRef HEAD

RemoteSha 6067e39ecb67810c016f7d695fe6f1031afaac86

Contents

check_setup 2
course_packages 3
cv_speakers 4
get_tutorial 5
install_course 6
lectures 7
partial_dependence 8
resid_acf 9
slides 10
theme_paper 10
tutorials 11

Index **12**

`check_setup` *Check that your computer is ready for a course*

Description

Reports your version of R and checks that every package the course uses is installed. It also checks that your version of simling has every data set the slides of the course use. Run it before the first session, and again whenever course code fails on your computer.

Usage

```
check_setup(course = "statistics")
```

Arguments

`course`	Which course: `"statistics"` or `"advanced"`.

Value

A tibble with one row per package and the columns `package`, `installed` (`TRUE` or `FALSE`) and `version`, invisibly. The function is called for the report it prints.

Examples

```
check_setup("statistics")
```

`course_packages` *The packages each course uses*

Description

Lists the packages used in the slides and tutorials of each course. The packages of the Statistics course are installed together with konstats; the extra packages of Advanced Statistics are installed with `install_course()`.

Usage

```
course_packages(course = NULL)
```

Arguments

`course`	Which course: `"statistics"` or `"advanced"`. `NULL`, the default, lists both.

Value

A tibble with one row per package and course, and the columns:

course `"statistics"` or `"advanced"`.

package Name of the package.

use What the package is used for in the course.

Examples

```
course_packages()
course_packages("advanced")
```

`cv_speakers` *Cross-validate a random forest by speaker*

Description

Splits the speakers into `k` folds, fits a random forest from ranger on all folds but one, and predicts the tokens of the speakers left out, so that every token is predicted by a forest that never saw its speaker. This estimates the accuracy for new speakers, which the out-of-bag accuracy overestimates when tokens of the same speaker are alike.

Usage

```
cv_speakers(
  formula,
  data,
  speaker = "speaker",
  k = 5,
  seed = 2,
  num_trees = 500
)
```

Arguments

`formula`	Formula of the forest, such as `l1 ~ vot + vowel_ratio + rate`.
`data`	A data frame with one row per token.
`speaker`	Name of the column with the speakers, as a string.
`k`	Number of folds.
`seed`	Seed for assigning speakers to folds and for breaking ties between the votes of the trees. The forest of fold `i` uses seed `i`. Your own random seed is left as it was.
`num_trees`	Number of trees in each forest.

Details

When the response is a factor with one value per speaker, such as the L1, each fold gets the same number of speakers from each group. With the default seed, the folds and forests are the ones in the slides of Advanced Statistics, session 6.

Value

A tibble with one row per token and the columns `speaker`, `fold`, `observed` and `predicted`. The accuracy for new speakers is `mean(cv$predicted == cv$observed)`.

Examples

```
if (requireNamespace("ranger", quietly = TRUE)) {
  library(simling)
  cv <- cv_speakers(l1 ~ vot + vowel_ratio + rate, accent, num_trees = 100)
  mean(cv$predicted == cv$observed)
}
```

`get_tutorial` *Copy exercise sheets into your project*

Description

Copies the exercise sheet of a tutorial into a folder, where you can open it and write your answers in the empty chunks. If the sheet has a weekly assignment, it comes at the end, and you hand in the sheet itself (the `.Rmd` file).

Usage

```
get_tutorial(
  number,
  dir,
  course = "statistics",
  overwrite = FALSE,
  open = interactive()
)
```

Arguments

`number`	Number of the tutorial, the same as its lecture, such as `4`. Several numbers, such as `2:4`, or `"all"` copy several sheets at once. See `tutorials()` for the sheets available.
`dir`	Folder to copy the sheets into, such as `"."` for the current folder of your R project.
`course`	Which course: `"statistics"` or `"advanced"`.
`overwrite`	If `TRUE`, replace sheets of the same name that are already in `dir`, answers and all. By default a sheet that is already there is kept: with one number the copy stops with an error, with several numbers or `"all"` that sheet is skipped and the others are copied.
`open`	If `TRUE`, the default in an interactive session, open the sheet after copying it. Only used when you copy one sheet.

Value

The paths of the sheets in `dir`, invisibly.

Examples

```
dir <- tempfile()
dir.create(dir)
get_tutorial(4, dir, open = FALSE)
get_tutorial("all", dir)
list.files(dir)
```

`install_course` *Install the packages of a course*

Description

Installs the packages a course uses that are not installed yet, from CRAN. The packages of the Statistics course come with konstats, so `install_course("advanced")` is the one you need: run it once before the first session of Advanced Statistics.

Usage

```
install_course(course = "advanced", update = FALSE)
```

Arguments

`course`	Which course: `"statistics"` or `"advanced"`.
`update`	If `TRUE`, also reinstall the packages that are already installed, to get their latest versions.

Value

The names of the packages it installed, invisibly.

Examples

```
## Not run:
install_course("advanced")

## End(Not run)
```

`lectures` *The lectures of the courses*

Description

Lists the lectures of Statistics and the sessions of Advanced Statistics, with the chapters of Winter (2019) they follow, the simling data sets they use and the address of their slides.

Usage

```
lectures(course = NULL)
```

Arguments

`course`	Which course: `"statistics"` or `"advanced"`. `NULL`, the default, lists both.

Value

A tibble with one row per lecture and the columns:

course `"statistics"` or `"advanced"`.

number Number of the lecture.

topic Topic of the lecture.

winter Chapters of Winter (2019), *Statistics for Linguists: An Introduction Using R*, that the lecture follows. `NA` for Advanced Statistics.

datasets simling data sets used in the slides, separated by commas. `penguins` comes from palmerpenguins.

tutorial `TRUE` if the lecture has an exercise sheet, see `tutorials()`.

slides Address of the slides. `NA` for sessions still in preparation.

Examples

```
lectures()
lectures("advanced")

# lectures that use a given data set
lecs <- lectures()
lecs[grepl("megastudy", lecs$datasets), c("course", "number", "topic")]
```

`partial_dependence` *Partial dependence of a model on one predictor*

Description

Sets one predictor to the same value in every row of the data, predicts, averages the predictions, and repeats for each value (Friedman, 2001). Plotted against the values, the result shows how the predictions change with the predictor, averaged over the other predictors. Used in Advanced Statistics for random forests from ranger, but it works with any model that has a `predict()` method.

Usage

```
partial_dependence(model, data, var, values, class = NULL, ...)
```

Arguments

`model`	A fitted model. For a classification forest from ranger, fit it with `probability = TRUE`.
`data`	The data to predict for, usually the data the model was fitted to.
`var`	Name of the predictor, as a string.
`values`	Values of the predictor to set, such as `seq(18, 80, by = 2)`. For a factor, levels of the factor.
`class`	For models that predict a probability for each class: the class whose probability to average, such as `"deleted"`.
`...`	Passed on to `predict()`. For a logistic regression from `glm()`, use `type = "response"` to average probabilities instead of log-odds.

Value

A tibble with one row per value and the columns `value` and `prediction`, the mean prediction (or probability of `class`).

References

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics, 29*(5), 1189–1232.

Examples

```
library(simling)
m <- lm(rt ~ log_freq * length, data = megastudy)
partial_dependence(m, megastudy, "length", 3:12)

if (requireNamespace("ranger", quietly = TRUE)) {
  rf <- ranger::ranger(deleted ~ following + morph + rate + age, data = td_deletion,
                       num.trees = 200, probability = TRUE, seed = 1)
  partial_dependence(rf, td_deletion, "age", seq(20, 80, by = 10), class = "deleted")
```

```
}

# a logistic regression: average the probabilities, not the log-odds
m <- glm(deleted ~ age + rate, data = td_deletion, family = binomial)
partial_dependence(m, td_deletion, "age", c(20, 50, 80), type = "response")
```

`resid_acf` *Autocorrelation of residuals within groups*

Description

Computes the autocorrelation of model residuals within each trajectory, time series or other group, for lags from 0 to `max_lag`. At lag 1 it is the correlation between each residual and the residual one measurement earlier in the same group; lag 2 skips one measurement, and so on. Residuals are never paired across groups. Used in Advanced Statistics to check GAMMs for autocorrelated residuals; `itsadug::acf_resid()` does the same with base R graphics.

Usage

```
resid_acf(res, group, max_lag = 8)
```

Arguments

`res`	Residuals of a model, such as `residuals(m)`.
`group`	The trajectory or series each residual belongs to, such as `data$traj`.
`max_lag`	The largest lag, in measurements.

Details

The residuals must be in the order of the data, sorted by time within each group, and `res` and `group` must have the same length: if the model dropped rows with missing values, drop them from `group` too.

Value

A tibble with one row per lag and the columns `lag` and `acf`. The autocorrelation at lag 0 is always 1. It is `NA` at lags as long as the groups, where there are too few pairs of residuals left.

Examples

```
library(simling)
# a straight line through the PRICE trajectories: the residuals of
# neighbouring measurements are strongly correlated
m <- lm(f2 ~ time, data = price_f2)
resid_acf(residuals(m), price_f2$traj)
```

slides *Open the slides of a lecture*

Description

Opens the slides of a lecture in your web browser.

Usage

```
slides(lecture, course = "statistics", open = interactive())
```

Arguments

lecture	Number of the lecture, or part of its topic, such as "mixed" or "random forests". A topic must match one lecture only.
course	Which course: "statistics" or "advanced".
open	If TRUE, the default in an interactive session, open the slides in the browser.

Value

The address of the slides, invisibly.

Examples

```
slides(4, open = FALSE)
slides("random forests in practice", course = "advanced", open = FALSE)
```

theme_paper *A ggplot2 theme for papers*

Description

A plain theme for figures in papers and theses: white background, light grid without minor lines, bold left-aligned facet labels, a title that starts at the left edge of the figure and the legend at the bottom. Built in Advanced Statistics, session 11. `ggplot2::theme_set(theme_paper())` makes it the default.

Usage

```
theme_paper(base_size = 11, base_family = "")
```

Arguments

base_size	Base font size, in points.
base_family	Base font family.

Value

A ggplot2 theme.

Examples

```
library(ggplot2)
library(simling)
ggplot(vowel_duration, aes(x = voicing, y = duration)) +
  geom_boxplot() +
  labs(x = "Following consonant", y = "Vowel duration (ms)") +
  theme_paper()
```

---

tutorials *The weekly exercise sheets*

Description

Lists the exercise sheets that come with konstats. Each sheet has the exercises of one tutorial and an empty code chunk for each answer: an R Markdown file (`.Rmd`) in Statistics, a Quarto document (`.qmd`) in Advanced Statistics. RStudio opens and renders both. Some sheets end with a weekly assignment, which you complete at home. Tutorial 1 of Statistics is a guided tutorial, and its sheet has all the code. Copy one into your project folder with `get_tutorial()`, or all of them with `get_tutorial("all", dir)`.

Usage

```
tutorials(course = NULL)
```

Arguments

`course`	Which course: `"statistics"` or `"advanced"`. `NULL`, the default, lists both.

Value

A tibble with one row per sheet and the columns `course`, `number` (the number of the lecture it belongs to), `topic` and `file`.

Examples

```
tutorials()
```

Index

check_setup, 2
course_packages, 3
cv_speakers, 4

get_tutorial, 5
get_tutorial(), *11*

install_course, 6
install_course(), *3*

lectures, 7

partial_dependence, 8

resid_acf, 9

slides, 10

theme_paper, 10
tutorials, 11
tutorials(), *5*, *7*