Package 'industRial'

July 22, 2025

Type Package

Date 2021-06-04

Title Data, Functions and Support Materials from the Book ``industRial Data Science"

Version 0.1.0

Description Companion package to the book ``industRial data science", J.Ramalho (2021) <https://j-ramalho.github.io/industRial/>. Provides data sets and functions to complete the case studies and contains the book original Rmd files and tutorials.

URL https://github.com/J-Ramalho/industRial

BugReports https://github.com/J-Ramalho/industRial/issues

License GPL (>= 3)

Encoding UTF-8

LazyData true

Imports ggplot2, stats, dplyr, tidyr, magrittr, rlang, lattice, SixSigma

Depends R (>= 3.5.0)

RoxygenNote 7.1.1

Suggests glue, tibble, stringr, scales, purrr, janitor, patchwork, forcats, broom, viridis, learnr, DoE.base, qcc, car, qicharts2, rsm, ggforce, ggraph, tidygraph, igraph, bookdown, rmarkdown, knitr, agricolae, RcmdrMisc, gt, skimr, ggtext

NeedsCompilation no

Author Joao Ramalho [aut, cre]

Maintainer Joao Ramalho <ramalho.joao@protonmail.com>

Repository CRAN

Date/Publication 2021-06-11 09:40:02 UTC

Contents

`battery_charging` *Charging time of a lithium-ion battery.*

Description

A data set with charging time in hours required to recharge a lithium-ion battery based on a full factorial design of experiment with four variables (A, B, C, D) coded as +/- 1. Design effects are coded as numerical variables in order to allow to build models without coding the contrasts and then to make predictions on a continuous range from -1 to +1.

A Variable A (numerical)

B Variable B (numerical)

C Variable B (numerical)

D Variable B (numerical)

Replicate The independent repeat of each unique factor combination.

charging_time Battery charging time [h]

Usage

```
battery_charging
```

Format

A tibble with 32 observations on 6 variables.

Source

Original data set.

References

For a complete case study application refer to https://j-ramalho.github.io/industRial/.

Examples

```
data(battery_charging)
head(battery_charging)

# Building a linear model:
battery_lm <- lm(
    formula = charging_time ~ A * B * C,
    data = battery_charging
)
summary(battery_lm)
```

---

chart_Cpk *Create a capability chart for statistical process control*

Description

Generate a histogram type chart from a set of consecutive measurements.

Usage

```
chart_Cpk(data)
```

Arguments

data	A dataset generated by the function process_stats

Details

This type of chart is typically applied in product manufacturing to monitor deviations from the target value over time. It is usually accompanied by the statistical process control time series chart_I and chart_IMR

Value

This function returns an object of class ggplot

References

For a complete case study application refer to https://j-ramalho.github.io/industRial/

`chart_I` *Create IMR chart for statistical process control*

Description

Generate a single point time series chart from a set of consecutive measurements.

Usage

```
chart_I(data)
```

Arguments

`data`	A dataset generated by the function `process_stats`

Details

This type of chart is typically applied in product manufacturing to monitor deviations from the target value over time. It is usually accompanied by the `chart_IMR`

Value

This function returns an object of class ggplot

References

For a complete case study application refer to https://j-ramalho.github.io/industRial/

chart_IMR *Create R MR chart for statistical process control*

Description

Generate a moving range chart chart from a set of consecutive measurements.

Usage

```
chart_IMR(data)
```

Arguments

data	A dataset generated by the function process_stats

Details

This type of chart is typically applied in product manufacturing to monitor deviations from the target value over time. It is usually accompanied by the chart_IMR

Value

This function returns an object of class ggplot

References

For a complete case study application refer to https://j-ramalho.github.io/industRial/

dial_control *Collection of visual defects on watch dial production.*

Description

This data set contains observations of visual defects present in watch dials such as indentations and scratches taken during production. It provides a practical case to establish pareto charts typically with a function like paretochart.

Operator The shop floor operator collecting the data

Date Data collection date

Defect Defect type ("Indent", "Scratch")

Location Position on the watch dial refered to as the hour (1h, 2h)

id Part unique id number

Usage

```
dial_control
```

Format

An object of class tibble with 58 observations on 4 variables.

Source

Original data set.

References

For a complete case study application refer to https://j-ramalho.github.io/industRial/.

Examples

```
head(dial_control)
```

`ebike_hardening` *Cycles to failure of ebikes frames after temperature treatment.*

Description

A data set with the results of aging tests on several groups of ebikes frames (g1, g2, ...). Each entry corresponds to the number of cycles to failure for each level of treatment temperature-

temperature Position of the part on the device

g1 group 1, remaining groups have names g2 to g5

Usage

```
ebike_hardening
```

Format

A tibble with 4 observations on 6 variables.

Details

The ebike_hardening2 dataset contains alternative data that gives non significant results in the analysis of variance study.

Source

Original data set.

References

For a complete case study application refer to https://j-ramalho.github.io/industRial/.

Examples

```
data(ebike_hardening)
```

`expand_formula` *Formula expansion*

Description

Takes a linear model formula and returns it expanded version.

Usage

```
expand_formula(formulae)
```

Arguments

`formulae`	Takes as input object of class formula, e.g.: Y ~ A * B, see ?formula for syntax details

Details

Supports verification and understanding of the creation of linear models syntax such as *,+ and other conventions.

Value

Returns a character vector such as A + B + A:B

References

For an example application refer to https://j-ramalho.github.io/industRial/

`industRial` *industRial: companion package to the book "industRial data science"*

Description

This package contains datasets and toy functions to run the examples from the book "industRial data science". It also contains all the book original Rmd files and the learnr Rmd original tutorial files.

Author(s)

João Ramalho

References

For complete case studies refer to https://j-ramalho.github.io/industRial/

`juice_drymatter` *Dry matter content of different juices obtained with two different measurement devices.*

Description

This data set contains laboratory measurements of the dry matter content of different fruit juices obtained with two different measurement devices. One of the devices is considered the reference (REF) and the other one is a new device (DRX) on which a linearity and bias study has to be performed.

product The juice base fruit ("Apple", "Beetroot")

drymatter_TGT Target drymatter content in [g]

speed Production line speed

particle_size Dry matter powder particle size [micrometers]

part Part number

drymatter_DRX Drymatter content measured with device DRX

drymatter_REF Drymatter content measured with reference device

Usage

```
juice_drymatter
```

Format

An object of class tibble with 108 observations on 7 variables.

Source

Adapted from a real gage bias and linearity study performed in 2021 on industrial beverages dry matter content measurement. The structure of the data corresponds to a full factorial design of 5 factors (3 with 3 levels and 2 with 2 levels).

References

For a complete case study application refer to https://j-ramalho.github.io/industRial/.

Examples

```
library(dplyr)
# Calculate the bias between the new device and the reference:
juice_drymatter <- juice_drymatter %>% dplyr::mutate(bias = drymatter_DRX - drymatter_REF)
# Establish the analysis of variance:
juice_drymatter_aov <- aov(
     bias ~ drymatter_TGT * speed * particle_size,
     data = juice_drymatter)
summary(juice_drymatter_aov)
```

`off_spec` *Calculate percentage of out of specification for Statistical Process Control*

Description

This function takes process variables and calculates the probability that parts are produced out of specification on the long run.

Usage

```
off_spec(UCL, LCL, mean, sd)
```

Arguments

`UCL`	the process upper control limit
`LCL`	the process lower control limit
`mean`	the process mean
`sd`	the process standard deviation

Value

This function returns an object of class numeric

References

For a complete case study application refer to https://j-ramalho.github.io/industRial/

Examples

```
off_spec(100, 0, 10, 3)
```

`perfume_experiment` *Correlation matrix of the input variables of an experiment design in perfume formulation.*

Description

The data set contains the expected correlation (expressed in 1 to 10) of an experiment anonymized input variables. The dataset consists in a double entry table with the same variables in row and column. It is coded as a tibble but subsequent utilization in network plots requires it to be converted to a matrix format.

Usage

```
perfume_experiment
```

Format

A tibble with 22 observations on 23 variables.

Source

Original data set.

References

For a complete case study application refer to https://j-ramalho.github.io/industRial/.

Examples

```
data(perfume_experiment)
```

`pet_delivery` *Tensile strength values on PET raw material for the clothing industry.*

Description

Measurements of tensile strength of two different deliveries of PET raw material used in the clothing industry. The two data sets follow approximately a normal distribution.

A Tensile strenght measurements for product A [Mpa] (numeric)

B Tensile strenght measurements for product B [Mpa] (numeric)

Usage

```
pet_delivery
```

Format

An object of class tibble with 28 observations on 2 variables.

Source

Original data set.

References

For a complete case study application refer to https://j-ramalho.github.io/industRial/.

Examples

```
data(pet_delivery)
```

pet_doe *A factorial design for the improvement of PET film tensile strength.*

Description

The data corresponds to full factorial design with two factors coded as +/- and 3 replicates for each combination.

A PET formulation A (factor)

B PET formulation B (factor)

replicate the measurement replicate I to III (factor)

yield the output variable measured on the PET, (numerical)

Usage

```
pet_doe
```

Format

An object of classes design and data.frame with 12 observations of 4 variables.

Source

Original data set generated with the function `fac.design` form the package DoE.base.

References

For a complete case study application refer to https://j-ramalho.github.io/industRial/

Examples

```
data(pet_doe)
contrasts(pet_doe$A)
```

process_Cpk *Calculate process capability index for Statistical Process Control*

Description

This function takes process variables and calculates the Cpk index which is a measure of the process centering and variability against specification.

Usage

```
process_Cpk(UCL, LCL, mean, sd)
```

Arguments

UCL	the process upper control limit
LCL	the process lower control limit
mean	the process mean
sd	the process standard deviation

Value

This function returns an object of class numeric

References

For a complete case study application refer to https://j-ramalho.github.io/industRial/

Examples

```
process_Cpk(100, 0, 10, 3)
```

---

process_stats *Calculate summary statistics for Statistical Process Control*

Description

This function takes process variables and calculates summary statistics and presents them in a easy readable table format.

Usage

```
process_stats(data, part_spec_percent)
```

Arguments

data	This function takes the dataset tablet_thickness cleaned with the clean_names function from the janitor package
part_spec_percent	the process tolerance in percentage.

Value

This function returns an object with class tibble (tbl_df)

References

For a complete case study application refer to https://j-ramalho.github.io/industRial/

process_stats_table *Summary statistics table outputs for Statistical Process Control*

Description

This function takes summary statistics and presents them in a easy readable table format.

Usage

```
process_stats_table(data)
```

Arguments

data	A data set generated by the function process_stats

Value

This function returns an object with classes gt_tbl and list

References

For a complete case study application refer to https://j-ramalho.github.io/industRial/

solarcell_fill *Yearly outputs and fills factor of solarcells of different types.*

Description

A dataset with the energy output resulting from tests on solarcells made of three different configurations. The fill factor provides an indication of the cell quality and is a non controlled variable that can be taken into consideration in an analysis of covariance to better assess the output variation from material to material.

material The solar cell material (character)

output he yearly energy output (numberic)

fillfactor The fill factor measured for each cell (numberic)

Usage

```
solarcell_fill
```

Format

A tibble with 15 observations of 3 variables.

Source

Original data set.

References

For a complete case study application refer to https://j-ramalho.github.io/industRial/.

Examples

```
hist(solarcell_fill$output)
```

`solarcell_output` *Yearly outputs of solarcells of different types.*

Description

A dataset with the energy output resulting from tests on solarcells made of three different raw materials / configurations.

material The solar cell type (character)

run The test run (numberic)

T-10 The yearly output for the test result at temperature of 10°C

T20 The yearly output for the test result at temperature of 20°C

T50 The yearly output for the test result at temperature of 50°C

Usage

```
solarcell_output
```

Format

A tibble with 12 observations of 5 variables.

Source

Original data set.

References

For a complete case study application refer to https://j-ramalho.github.io/industRial/.

Examples

```
data(solarcell_output)
```

`ss.rr.plots` *Gage R & R plots*

Description

Extracts stand alone plots from the ss.rr function of the SixSigma package.

Usage

```
ss.rr.plots(
  var,
  part,
  appr,
  lsl = NA,
  usl = NA,
  sigma = 6,
  data,
  main = "Six Sigma Gage R&R Study",
  sub = "",
  alphaLim = 0.05,
  errorTerm = "interaction",
  digits = 4
)
```

Arguments

`var`	Measured variable
`part`	Factor for parts
`appr`	Factor for appraisers (operators, machines, ...)
`lsl`	Numeric value of lower specification limit used with USL to calculate Study Variation as %Tolerance
`usl`	Numeric value of upper specification limit used with LSL to calculate Study Variation as %Tolerance
`sigma`	Numeric value for number of std deviations to use in calculating Study Variation
`data`	Data frame containing the variables
`main`	Main title for the graphic output
`sub`	Subtitle for the graphic output (recommended the name of the project)
`alphaLim`	Limit to take into account interaction
`errorTerm`	Which term of the model should be used as error term (for the model with interation)
`digits`	Number of decimal digits for output

Details

This is a modified version of the function ss.rr from the SixSigma package that allows to extract the individual plots from the output report. The input arguments of the function are the same as the original function. See the original function help with ?ss.rr for full documentation.

Value

Generates a list output that can be assigned to a user created variable. The plots can then be accessed with the syntax variable$plot1 to plot6.

References

For an example application refer to https://j-ramalho.github.io/industRial/

---

syringe_diameter *Production measurements of the inner diameter of syringes barrels.*

Description

This dataset contains process control measurements of the barrel diameters of pharmaceutical syringes. The sampling rate is hourly and the sample size is 6 syringes.

Hour The sampling hour expressed as Hour1, Hour2 (character)

Sample1 Syringe diameter of sample 1 (numerical)

Sample2 Syringe diameter of sample 2 (numerical)

Usage

```
syringe_diameter
```

Format

A tibble with 25 observations on 7 variables.

Source

Original data set.

References

For a complete case study application refer to https://j-ramalho.github.io/industRial/.

Examples

```
data(syringe_diameter)
```

`tablet_thickness` *Thickness measurements of pharmaceutical tablets*

Description

This data set contains physical measurements of pharmaceutical tablets (pills) including measurement room conditions. The data and the insights it provides are typical of an industrial context with high production throughput and stringent dimensional requirements.

Usage

```
tablet_thickness
```

Format

An object of class tibble with 675 observations on 11 variables

Details

The data set contains other variables not used in the text book related with to the measurement room conditions (not listed).

Position Position of the part on the measurement device

Size Size class (L, M, S)

Tablet Part number (L001, L002, ...)

Replicate Measurement replicate, a sequential numbers

Day Measurement Day, a sequential numbers

Date [DD.MM.YYYY] Measurement date (POSIXct)

Operator Operator name (ficticious)

Thickness [micron] Tablet thickness (micrometers)

Temperature [°C] Room temperature

Source

Based on a gage r&R (gage reproducibility and repeatability) study performed in 2020 on a physical measurement of parts coming out of a high throughput industrial equipment.

References

For a complete case study application refer to https://j-ramalho.github.io/industRial/

Examples

```
data(tablet_thickness)
```

`tablet_weight` *Weight measurements of pharmaceutical tablets*

Description

This data set contains weight measurements of pharmaceutical tablets (pills). The data and the #' insights it provides are typical of an industrial context with high production throughput and stringent dimensional requirements.

Usage

```
tablet_weight
```

Format

An object of class tibble with 137 observations on 3 variables

Details

The data set contains other variables not used in the text book related with to the measurement room conditions (not listed).

part_id Unique sequencial identifier given during production (numeric)

Weight Target Value Tablet weight target specification value in [mg] (numeric

Weight Value Tablet weight measured value [m] (numeric)

Source

Anonymized data based on statistical process control data obtained in a high volume production setup.

References

For a complete case study application refer to https://j-ramalho.github.io/industRial/

Examples

```
hist(tablet_weight$`Weight value`)
```

`theme_industRial` *Custom theme "industRial" for the book industRial Data Science plots*

Description

This theme aims at optimal balance between readability and precision. It has adapted from the package cowplot by Claus O.Wilke and reflects the principles of his book Fundamentals of Data Visualization

Usage

```
theme_industRial(
  font_size = 14,
  font_family = "",
  line_size = 0.5,
  rel_small = 12/14,
  rel_tiny = 11/14,
  rel_large = 16/14,
  base_size = font_size,
  base_family = font_family
)
```

Arguments

`font_size`	defaults to 14
`font_family`	defaults to ""
`line_size`	defaults to 0.5
`rel_small`	defaults to 12/14
`rel_tiny`	defaults to 11/14
`rel_large`	defaults to 16/14
`base_size`	internal arguments, defaults to font_size
`base_family`	internal arguments, defaults to font_family

Details

Apply this theme by adding it at the end of the code of any `ggplot` chart. It basically combines the half open theme with a grid background from cowplot

Value

This function returns an object of classes theme and gg from the ggplot2 package

References

For a complete case study application refer to `https://j-ramalho.github.io/industRial/`

Examples

```
library(dplyr)
library(ggplot2)

pet_delivery %>%
   ggplot(aes(x = A)) +
   geom_histogram(color = "grey", fill = "grey90") +
   labs(title = "PET clothing case study",
      subtitle = "Raw data plot",
      x = "Treatment",
      y = "Tensile strength [MPa]") +
      theme_industRial()
```

---

theme_qcc *Custom theme "qcc" for the book industRial Data Science plots*

Description

This theme provides a similar look and feel to the package qcc statistical process control charts (SPC) which have themselves a resemblance with Minitab charts. This theme aims at providing a layout that is familiar to readers of Minitab chart to help in reducing transition to R build reports and charts.

Usage

```
theme_qcc(base_size = 12, base_family = "")
```

Arguments

base_size	font size, defaults to 12
base_family	font family defaults to ""

Details

Apply this theme by adding it at the end of the code of any ggplot chart. It #' basically provides a grey background and some highlights to help reading key process statistics such as the population mean.

Value

This function returns an object of classes theme and gg from the ggplot2 package

References

For a complete case study application refer to https://j-ramalho.github.io/industRial/

Examples

```
library(dplyr)
library(ggplot2)

pet_delivery %>%
   ggplot(aes(x = A)) +
   geom_histogram(color = "grey", fill = "grey90") +
   labs(title = "PET clothing case study",
      subtitle = "Raw data plot",
      x = "Treatment",
      y = "Tensile strength [MPa]") +
      theme_qcc()
```

Index