Package 'FormulR'

January 20, 2025

Title Comprehensive Tools for Drug Formulation Analysis and
Visualization

Version 1.0.0

Description This presents a comprehensive set of tools for the analysis and visualization of drug formulation data. It includes functions for statistical analysis, regression modeling, hypothesis testing, and comparative analysis to assess the impact of formulation parameters on drug release and other critical attributes. Additionally, the package offers a variety of data visualization functions, such as scatterplots, histograms, and boxplots, to facilitate the interpretation of formulation data. With its focus on usability and efficiency, this package aims to streamline the drug formulation process and aid researchers in making informed decisions during formulation design and optimization.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.2.3

Depends dplyr, ggplot2

Imports utils, knitr

VignetteBuilder knitr

NeedsCompilation no

Author Oche Ambrose George [aut, cre]
(<<https://orcid.org/0000-0002-3979-6232>>)

Maintainer Oche Ambrose George <ocheab1@gmail.com>

Repository CRAN

Date/Publication 2024-03-20 09:50:05 UTC

Suggests rmarkdown

Contents

---

anova_analysis   *Perform ANOVA analysis*

Description

This function conducts analysis of variance (ANOVA) to assess the impact of formulation parameters on key response variables.

Usage

```
anova_analysis(formulation_data)
```

Arguments

```
formulation_data
```
             A data frame containing the formulation data.

Value

A summary of the ANOVA analysis results.

Examples

```
formulation_data <- data.frame(
  Excipient_Concentration = runif(100, min = 0, max = 1),
  Drug_Release = rnorm(100, mean = 50, sd = 10),
  Particle_Size = rnorm(100, mean = 100, sd = 20)
)
anova_analysis(formulation_data)
```

---

batch_variability *Assess batch-to-batch variability*

Description

This function calculates the batch-to-batch variability of a specified parameter.

This function calculates the batch-to-batch variability of a specified parameter.

Usage

```
batch_variability(formulation_data, parameter)

batch_variability(formulation_data, parameter)
```

Arguments

formulation_data
 A data frame containing formulation data.

parameter The parameter for which batch-to-batch variability is calculated.

Value

The batch-to-batch variability of the specified parameter.

The batch-to-batch variability of the specified parameter.

---

boxplot *Generate boxplot*

Description

This function generates a boxplot to compare the distribution of a variable across different groups.

Usage

```
boxplot(formulation_data, x, y)
```

Arguments

formulation_data
 A data frame containing the formulation data.

x The name of the grouping variable.

y The name of the variable.

Value

A boxplot.

Examples

```
formulation_data <- data.frame(
  Formulation_Type = sample(c("Type A", "Type B"), 100, replace = TRUE),
  Drug_Release = rnorm(100, mean = 50, sd = 10)
)
boxplot(formulation_data, "Formulation_Type", "Drug_Release")
```

---

compare_distributions   *Compare distributions across groups*

Description

This function compares the distributions of a response variable across groups specified by group_var.

This function compares the distributions of a response variable across groups specified by group_var.

Usage

```
compare_distributions(formulation_data, group_var, response_var)

compare_distributions(formulation_data, group_var, response_var)
```

Arguments

formulation_data

A data frame containing formulation data.

group_var   The variable defining the groups for comparison.

response_var   The response variable to compare across groups.

Value

A boxplot comparing the distributions across groups.

A boxplot comparing the distributions across groups.

---

compare_means	*Compare means across groups*

Description

This function compares the means of a response variable across groups specified by group_var.

This function compares the means of a response variable across groups specified by group_var.

Usage

```
compare_means(formulation_data, group_var, response_var)

compare_means(formulation_data, group_var, response_var)
```

Arguments

formulation_data

> A data frame containing formulation data.

group_var   The variable defining the groups for comparison.

response_var   The response variable to compare across groups.

Value

Results of the t-test comparing means across groups.

Results of the t-test comparing means across groups.

---

confidence_intervals	*Confidence intervals of drug release*

Description

This function computes confidence intervals for drug release based on the provided formulation data.

This function computes confidence intervals for drug release based on the provided formulation data.

Usage

```
confidence_intervals(formulation_data)

confidence_intervals(formulation_data)
```

Arguments

`formulation_data`

A data frame containing formulation data.

Value

Confidence intervals for drug release.

Confidence intervals for drug release.

---

control_chart *Control chart for quality control*

Description

This function generates a control chart for monitoring the quality control parameter over time.

This function generates a control chart for monitoring the quality control parameter over time.

Usage

```
control_chart(formulation_data, parameter)

control_chart(formulation_data, parameter)
```

Arguments

`formulation_data`

A data frame containing formulation data.

`parameter`   The quality control parameter to monitor.

Value

A control chart for the specified quality control parameter.

A control chart for the specified quality control parameter.

---

histogram *Generate histogram*

Description

This function generates a histogram to visualize the distribution of a variable.

Usage

```
histogram(formulation_data, x, bins = 20)
```

Arguments

formulation_data

A data frame containing the formulation data.

x The name of the variable.

bins The number of bins for the histogram.

Value

A histogram.

Examples

```
formulation_data <- data.frame(
  Drug_Release = rnorm(100, mean = 50, sd = 10)
)
histogram(formulation_data, "Drug_Release")
```

---

hypothesis_testing *Perform hypothesis testing*

Description

This function conducts hypothesis testing to compare means between different formulation groups.

Usage

```
hypothesis_testing(formulation_data)
```

Arguments

formulation_data

A data frame containing the formulation data.

Value

The results of the hypothesis testing.

Examples

```
formulation_data <- data.frame(
  Formulation_Type = sample(c("Type A", "Type B"), 100, replace = TRUE),
  Drug_Release = rnorm(100, mean = 50, sd = 10)
)
hypothesis_testing(formulation_data)
```

---

regression_analysis   *Perform regression analysis*

Description

This function conducts regression analysis to model relationships between formulation parameters and response variables.

Usage

```
regression_analysis(formulation_data)
```

Arguments

```
formulation_data
```
A data frame containing the formulation data.

Value

A summary of the regression analysis results.

Examples

```
formulation_data <- data.frame(
  Excipient_Concentration = runif(100, min = 0, max = 1),
  Drug_Release = rnorm(100, mean = 50, sd = 10),
  Particle_Size = rnorm(100, mean = 100, sd = 20)
)
regression_analysis(formulation_data)
```

---

scatterplot *Generate scatterplot*

Description

This function generates a scatterplot to visualize the relationship between two variables.

Usage

```
scatterplot(formulation_data, x, y)
```

Arguments

```
formulation_data
```
A data frame containing the formulation data.

x   The name of the x-variable.

y   The name of the y-variable.

Value

A scatterplot.

Examples

```
formulation_data <- data.frame(
  Excipient_Concentration = runif(100, min = 0, max = 1),
  Drug_Release = rnorm(100, mean = 50, sd = 10)
)
scatterplot(formulation_data, "Excipient_Concentration", "Drug_Release")
```

---

summary_statistics *Summary statistics of formulation data*

Description

This function calculates summary statistics of the provided formulation data.

This function calculates summary statistics of the provided formulation data.

Usage

```
summary_statistics(formulation_data)
```

```
summary_statistics(formulation_data)
```

Arguments

`formulation_data`

A data frame containing formulation data.

Value

Summary statistics of the formulation data.

Summary statistics of the formulation data.

Index