Package 'CleaningValidation'

January 20, 2025

Version 1.0

Title Cleaning Validation Functions for Pharmaceutical Cleaning Process

Description Provides essential Cleaning Validation functions for complying with pharmaceutical cleaning process regulatory standards. The package includes non-parametric methods to analyze drug active-ingredient residue (DAR), cleaning agent residue (CAR), and microbial colonies (Mic) for non-Poisson distributions. Additionally, Poisson methods are provided for Mic analysis when Mic data follow a Poisson distribution.

License GPL-3

Depends R (>= 4.3.0)

Imports dplyr (>= 1.0.0), rlang, ggplot2 (>= 3.3.3), cowplot (>= 1.1.1), dunn.test, boot, AER, lme4

Encoding UTF-8

RoxygenNote 7.3.1

Author Mohamed Chan [aut],
Wendy Lou [aut],
Xiande Yang [aut, cre]

Maintainer Xiande Yang <xyang1@apotex.com>

URL https://github.com/ChandlerXiandeYang/CleaningValidation

NeedsCompilation no

Repository CRAN

Date/Publication 2024-05-17 09:10:21 UTC

Contents

CleaningValidation   *Cleaning Validation Package*

Description

This package offers a comprehensive suite of functions for cleaning validation, a critical component of quality control in pharmaceutical manufacturing. The included functions assist in analyzing residue data, evaluating cleaning efficacy, and ensuring that cleaning processes meet regulatory standards.

Details

The functions primarily return data frames, streamlining data preprocessing, analysis, and the application of statistical methods for cleaning process evaluation. This toolset simplifies the workflow for cleaning validation professionals, providing resources for various tasks. Function cv01 cleans three data types. Functions cv02 to cv12 (excluding cv05) are designed for sequential DAR and CAR analysis. Functions cv13 and cv14 assess whether Mic follows a Poisson distribution. For Mic data that follows a Poisson distribution, function cv05 and functions cv15 to cv29 should be used in sequence. If Mic data do not follow a Poisson distribution, function cv05 and functions cv02 to cv12 (excluding cv06) are applicable. Function cv30 synthesizes the Process Performance Index (Ppu) for DAR, CAR, and Mic. Supplementary to its core capabilities, the package includes datasets—Eq_DAR, Eq_CAR, and Eq_Mic—for demonstrating functionality in practical contexts.

License

This package is free software; you may redistribute and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License or (at your option) any later version.

Examples

```
## Not run:
# Example code here to demonstrate package usage:
# This could include data loading, transforming, and cleaning validation analysis.

## End(Not run)
```

---

cv01_dfclean   *Clean and preprocess residue data for stability and capability analysis*

Description

This function ensures data type and no missing data in residue_col, cleaning_event_col, usl_col of data their type. Furthermore, it changes cleaning_event_col to time ordered factor. It cleans and pre-processes the residue data for stability and capability analysis, ensuring that it meets the necessary criteria for analysis.

Usage

```
cv01_dfclean(data, residue_col, cleaning_event_col, usl_col)
```

Arguments

data	A data frame containing one of drug active-ingredient residue (DAR), cleaning agent residue (CAR), or microbial bioburden residue (Mic) data.
residue_col	The name of the column containing the numeric residue data.
cleaning_event_col	
	The name of the column containing the Cleaning Event data.
usl_col	The name of the column containing the numeric upper specification limit (USL) data.

Value

A cleaned and pre-processed data frame such that all variables have no missing values, its CleaningEvent
is time-ordered categorical variable, and Residue and USL are numeric.

Author(s)

Chan, Mohamed, Lou, Wendy, Yang, Xiande [xiande.yang at gmail.com]

Examples

```
# Assume Eq_DAR, Eq_CAR, and Eq_Mic are loaded datasets

# Clean and preprocess residue data for Eq_DAR
Eq_DAR <- cv01_dfclean(data = Eq_DAR, residue_col = "DAR", usl_col = "USL",
cleaning_event_col = "CleaningEvent")

# Clean and preprocess residue data for Eq_CAR
Eq_CAR <- cv01_dfclean(data = Eq_CAR, residue_col = "CAR", usl_col = "USL",
cleaning_event_col = "CleaningEvent")

# Clean and preprocess residue data for Eq_Mic
Eq_Mic <- cv01_dfclean(data = Eq_Mic, residue_col = "Mic", usl_col = "USL",
cleaning_event_col = "CleaningEvent")
```

---

cv02_nonpro_oos_reswab

*Summarize Non-Process Related OOS and Reswab Data Which May
Not Be Included in the Analysis*

Description

This function processes three datasets to identify unique project IDs based on non-process-related
out-of-specification (OOS) and reswab cases, then summarizes this information into a dataframe. If
your data does not have reswab or OOS, you do not need to use this function.

Usage

```
cv02_nonpro_oos_reswab(Eq_DAR, Eq_CAR, Eq_Mic)
```

Arguments

Eq_DAR	A dataframe containing equipment DAR data.
Eq_CAR	A dataframe containing equipment CAR data.
Eq_Mic	A dataframe containing equipment Mic data.

Value

A dataframe summarizing the non-process-related OOS and reswab data.

Author(s)

Chan, Mohamed, Lou, Wendy, Yang, Xiande [xiande.yang at gmail.com]

Examples

```
cv02_nonpro_oos_reswab(Eq_DAR, Eq_CAR, Eq_Mic)
```

---

cv03_usl_unification   *Unify USL Percentages for Specified Residue*

Description

This function takes a dataset and computes the percentage of residue over USL for each event, as well as mean and median of these percentages for each cleaning event and overall.

Usage

```
cv03_usl_unification(data, cleaning_event_col, residue_col, usl_col)
```

Arguments

data	A dataframe containing the relevant dataset.
cleaning_event_col	
	Name of the column in 'data' that contains the cleaning event identifiers as a string.
residue_col	Name of the column in 'data' that contains the residue measurements as a string.
usl_col	Name of the column in 'data' that contains the USL values as a string.

Value

A dataframe with original data and additional columns for residue percentages, and their mean and median values per cleaning event and overall.

Author(s)

Chan, Mohamed, Lou, Wendy, Yang, Xiande [xiande.yang at gmail.com]

Examples

```
cv03_usl_unification(data = Eq_DAR, cleaning_event_col = "CleaningEvent",
residue_col = "DAR", usl_col = "USL")
```

---

cv04_histogram_kde   *Plot Histogram with Kernel Density Estimate Curve*

Description

This function takes a dataset and a column representing the residue percentages and generates a histogram overlaid with a KDE (Kernel Density Estimate) curve. It calculates and marks quantiles P0.5, P0.8413, P0.9772, and the P0.99865, i.e., UCL (Upper Control Limit) on the plot.

Usage

```
cv04_histogram_kde(data, residue_pct_col)
```

Arguments

data   A dataframe containing the relevant dataset.

residue_pct_col

The name of the column in 'data' that contains the residue percentages.

Value

A ggplot object representing the histogram with KDE curve.

Author(s)

Chan, Mohamed, Lou, Wendy, Yang, Xiande [xiande.yang at gmail.com]

Examples

```
Eq_DAR <- cv03_usl_unification(data=Eq_DAR,"CleaningEvent", "DAR", usl_col="USL")
cv04_histogram_kde(data = Eq_DAR, residue_pct_col = "DAR_Pct")
```

---

cv05_sw_norm_test_1   *Perform Shapiro-Wilk Normality Test*

Description

This function performs the Shapiro-Wilk test for normality on a specified variable in a dataset. It returns a data frame with the variable name, the Shapiro-Wilk statistic, the p-value in scientific notation, and an indication of whether the p-value is less than 0.05.

Usage

```
cv05_sw_norm_test_1(data, residue_col)
```

Arguments

`data`	A data frame containing the dataset.
`residue_col`	The name of the column to test for normality.

Value

A data frame with the Shapiro-Wilk test results.

Author(s)

Chan, Mohamed, Lou, Wendy, Yang, Xiande [xiande.yang at gmail.com]

Examples

```
data(Eq_Mic)
cv05_sw_norm_test_1(data=Eq_Mic, residue_col="Mic")
```

---

cv06_sw_norm_test_2	*Perform Shapiro-Wilk Normality Test on Two Variables*

Description

This function performs the Shapiro-Wilk test for normality on two specified variables within a dataset. It returns a data frame with the variables' names, Shapiro-Wilk statistics, p-values in scientific notation, and indications of whether the p-values are less than 0.05.

Usage

```
cv06_sw_norm_test_2(data, residue_col, residue_pct_col)
```

Arguments

`data`	A data frame containing the dataset.
`residue_col`	The name of the first column to test for normality.
`residue_pct_col`	
	The name of the second column to test for normality.

Value

A data frame with Shapiro-Wilk test results for both variables.

Author(s)

Chan, Mohamed, Lou, Wendy, Yang, Xiande [xiande.yang at gmail.com]

Examples

```
# assuming Eq_DAR is a predefined dataset
Eq_DAR <- cv03_usl_unification(data = Eq_DAR,  "CleaningEvent",  "DAR",  "USL")
cv06_sw_norm_test_2(data=Eq_DAR, residue_col="DAR", residue_pct_col="DAR_Pct")
```

---

cv071_median_control_chart

Median Control Chart

Description

This function creates a control chart for the median residue percentages based on kernel density estimation. The in put residue_pct_meidan_col can be median of non-USL_unified variable such as Mic_Median, DAR_Median, or CAR_Median

Usage

```
cv071_median_control_chart(data, cleaning_event_col, residue_pct_median_col)
```

Arguments

data   A data frame containing the data to plot.

cleaning_event_col

                    The name of the column containing cleaning event identifiers.

residue_pct_median_col

                    The name of the column containing the calculated median residue percentages.

Value

The meidan control chart.

Author(s)

Chan, Mohamed, Lou, Wendy, Yang, Xiande [xiande.yang at gmail.com]

Examples

```
# Assuming 'Eq_DAR' is a data frame with appropriate columns:
Eq_DAR <- cv03_usl_unification(data = Eq_DAR,  "CleaningEvent",  "DAR",  "USL")
cv07_median_control_chart(data = Eq_DAR, cleaning_event_col = "CleaningEvent",
residue_pct_median_col="DAR_Pct_Median")
```

```
cv07_median_control_chart
```
Median Control Chart and Density Plot

Description

This function creates a control chart and a density plot for the median residue percentages based on kernel density estimation.

Usage

```
cv07_median_control_chart(data, cleaning_event_col, residue_pct_median_col)
```

Arguments

data   A data frame containing the data to plot.

cleaning_event_col

     The name of the column containing cleaning event identifiers.

residue_pct_median_col

     The name of the column containing the calculated median residue percentages.

Value

A cowplot object containing the combined control chart and density plot.

Author(s)

Chan, Mohamed, Lou, Wendy, Yang, Xiande [xiande.yang at gmail.com]

Examples

```
# Assuming 'Eq_DAR' is a data frame with appropriate columns:
Eq_DAR <- cv03_usl_unification(data = Eq_DAR, "CleaningEvent", "DAR", "USL")
cv07_median_control_chart(data = Eq_DAR, cleaning_event_col = "CleaningEvent",
residue_pct_median_col="DAR_Pct_Median")
```

```
cv08_variability_chart
```
Variability Chart for Cleaning Events

Description

This function generates a variability chart for cleaning events, showing data points, outliers, and overall statistics like the grand mean and median.

Usage

```
cv08_variability_chart(data, cleaning_event_col, residue_pct_col, usl_pct_col)
```

Arguments

data	A data frame containing the data to plot.
cleaning_event_col	
	Name of the column representing cleaning events (as a string).
residue_pct_col	
	Name of the column representing residue percentages (as a string).
usl_pct_col	Name of the column representing the upper specification limit percentages (as a string).

Value

A ggplot object representing the variability chart.

Author(s)

Chan, Mohamed, Lou, Wendy, Yang, Xiande [xiande.yang at gmail.com]

Examples

```
Eq_DAR <- cv01_dfclean(data=Eq_DAR, cleaning_event_col="CleaningEvent",
residue_col="DAR", usl_col="USL" )
Eq_DAR <- cv03_usl_unification(data=Eq_DAR, cleaning_event_col="CleaningEvent",
residue_col="DAR", usl_col="USL")
cv08_variability_chart(data=Eq_DAR, cleaning_event_col="CleaningEvent",
residue_pct_col="DAR_Pct", usl_pct_col="USL_Pct")
```

---

cv09_kw_test	*Kruskal-Wallis Test for Residue Percentages*

Description

Perform Kruskal-Wallis test for residue percentages based on cleaning events.

Usage

```
cv09_kw_test(data, residue_col, cleaning_event_col)
```

Arguments

data	A data frame containing the data.
residue_col	The name of the column containing residue percentages.
cleaning_event_col	
	The name of the column containing cleaning event identifiers.

Value

A data frame of Kruskal-Wallis test results.

Author(s)

Chan, Mohamed, Lou, Wendy, Yang, Xiande [xiande.yang at gmail.com]

Examples

```
# Assuming 'Eq_DAR' is the data frame, 'DAR_Pct' is the residue column,
# and 'CleaningEvent' is the cleaning event column.
Eq_DAR <- cv01_dfclean(data=Eq_DAR, cleaning_event_col="CleaningEvent",
residue_col="DAR", usl_col="USL" )
Eq_DAR <- cv03_usl_unification(data=Eq_DAR, cleaning_event_col="CleaningEvent",
residue_col="DAR", usl_col="USL")
kw_test_results <- cv09_kw_test(data = Eq_DAR, residue_col = "DAR_Pct",
 cleaning_event_col = "CleaningEvent")
```

```
cv10_dunn_test_vs_control
```
Dunn's Test for Residue

Description

Perform Dunn's test for residue based on cleaning events. Choose the control group as the cleaning event whose median is closest to the grand median. This function is for investigation purpose.

Usage

```
cv10_dunn_test_vs_control(data, residue_col, cleaning_event_col)
```

Arguments

data	A data frame containing the data.
residue_col	The name of the column containing residue.
cleaning_event_col	
	The name of the column containing cleaning event identifiers.

Value

A data frame of Dunn's test results with control group.

Author(s)

Chan, Mohamed, Lou, Wendy, Yang, Xiande [xiande.yang at gmail.com]

Examples

```
# 'Eq_DAR' is the  data frame, 'DAR_Pct' is the residue column, and
# 'CleaningEvent' is the cleaning event column.
Eq_DAR <- cv03_usl_unification(data = Eq_DAR, residue_col = "DAR",
cleaning_event_col = "CleaningEvent",  usl_col = "USL")
dunn_test_results_vs_control <- cv10_dunn_test_vs_control(data = Eq_DAR,
residue_col = "DAR_Pct", cleaning_event_col = "CleaningEvent")
```

---

cv11_vca_by_median   *Variability Components Analysis by Median with Bootstrap*

Description

Perform Variability Components Analysis (VCA) by median for residue percentages based on cleaning events with bootstrap for confidence intervals.

Usage

```
cv11_vca_by_median(data, residue_col, cleaning_event_col, n_bootstrap = 2000)
```

Arguments

data	A data frame containing the data.
residue_col	The name of the column containing residue percentages.
cleaning_event_col	
	The name of the column containing cleaning event identifiers.
n_bootstrap	The number of bootstrap iterations. Default is 2000.

Value

A data frame summarizing variability components analysis by median along with confidence intervals from bootstrap.

Author(s)

Chan, Mohamed, Lou, Wendy, Yang, Xiande [xiande.yang at gmail.com]

Examples

```
# Assuming 'Eq_DAR' is the data frame, 'DAR_Pct' is the residue column,
# and 'CleaningEvent' is the cleaning event column.
Eq_DAR <- cv01_dfclean(data=Eq_DAR, cleaning_event_col="CleaningEvent",
residue_col="DAR", usl_col="USL" )
Eq_DAR <- cv03_usl_unification(data=Eq_DAR, cleaning_event_col="CleaningEvent",
residue_col="DAR", usl_col="USL")
summary <- cv11_vca_by_median(data = Eq_DAR, residue_col = "DAR_Pct",
cleaning_event_col = "CleaningEvent", n_bootstrap = 2000)
```

---

cv12_kde_ppu   *Calculate PPU using KDE density estimation*

Description

Calculate PPU using KDE density estimation

Usage

```
cv12_kde_ppu(
  data,
  residue_col,
  cleaning_event_col,
  usl_col,
  n_bootstrap = 1000
)
```

Arguments

data	The dataset containing the columns specified in other parameters.
residue_col	The name of the column containing residue data.
cleaning_event_col	
	The name of the column containing cleaning event data (unused).
usl_col	The name of the column containing USL values.
n_bootstrap	The number of bootstrap samples to use.

Value

A dataframe with the estimated PPU and its 95

Author(s)

Chan, Mohamed, Lou, Wendy, Yang, Xiande [xiande.yang at gmail.com]

Examples

```
 Eq_DAR <- cv03_usl_unification(data = Eq_DAR, cleaning_event_col = "CleaningEvent",
residue_col = "DAR", usl_col = "USL")
cv12_kde_ppu(data = Eq_DAR, residue_col = "DAR_Pct", cleaning_event_col = "CleaningEvent",
 usl_col = "USL_Pct", n_bootstrap = 1000)
```

---

cv13_poisson_test   *Poisson Goodness-of-Fit Test*

Description

Conducts a goodness-of-fit test to evaluate if the Mic data follows a Poisson distribution.

Usage

```
cv13_poisson_test(data, residue_col)
```

Arguments

data   A dataframe containing the observed data.

residue_col   A string specifying the column in 'data' to be tested.

Value

A dataframe object representing the chi-squared statistic and the p-value from the goodness-of-fit test.

Author(s)

Chan, Mohamed, Lou, Wendy, Yang, Xiande [xiande.yang at gmail.com]

Examples

```
# Assuming Eq_Mic is your dataframe and Mic is the column to be tested
 cv13_poisson_test(data=Eq_Mic, residue_col="Mic")
```

---

cv14_dispersion_test   *Dispersion Test for Poisson Regression Models*

Description

Performs a dispersion test on a Poisson regression model to check for overdispersion. The function fits a Poisson regression model to the data using the specified columns, and then performs a dispersion test using the model.

Usage

```
cv14_dispersion_test(data, residue_col, cleaning_event_col)
```

Arguments

data   A dataframe containing the observed data.

residue_col   A string specifying the response column in the model.

cleaning_event_col

A string specifying the explanatory variable in the model.

Value

A dataframe object with the results of the overdispersion test, including

Author(s)

Chan, Mohamed, Lou, Wendy, Yang, Xiande [xiande.yang at gmail.com] the Z-value, P-value, and dispersion estimate.

Examples

```
cv14_dispersion_test(data = Eq_Mic, residue_col = "Mic", cleaning_event_col = "CleaningEvent")
```

---

cv15_mic_mutate   *Calculate Mic Statistics*

Description

Calculate Mic Statistics

Usage

```
cv15_mic_mutate(data, cleaning_event_col, residue_col)
```

Arguments

data   A dataframe containing the data.

cleaning_event_col

The name of the column that identifies the cleaning event.

residue_col   The name of the column containing residue measurements.

Value

A dataframe with new columns for mean, median, grand mean, and grand median of Mic values.

Author(s)

Chan, Mohamed, Lou, Wendy, Yang, Xiande [xiande.yang at gmail.com]

Examples

```
cv15_mic_mutate(data=Eq_Mic, cleaning_event_col="CleaningEvent", residue_col="Mic")
```

---

cv16_u_chart   *Create a u-Chart for Poisson-distributed Data*

Description

This function generates a u-chart for visualizing the stability and capability of a process based on a Poisson distribution.

Usage

```
cv16_u_chart(data, residue_col, cleaning_event_col)
```

Arguments

data   A data frame containing the data for plotting.

residue_col   The name of the column representing residue data (numeric).

cleaning_event_col

                     The name of the column representing cleaning events (factor or character).

Value

A ggplot object representing the u-chart.

Author(s)

Chan, Mohamed, Lou, Wendy, Yang, Xiande [xiande.yang at gmail.com]

Examples

```
cv16_u_chart(data = Eq_Mic, residue_col = "Mic", cleaning_event_col = "CleaningEvent")
```

---

cv17_cusum   *Create a CUSUM Chart for Poisson-distributed Data*

Description

This function computes the cumulative sum (CUSUM) for the mean values of a specified residue column aggregated by a cleaning event column. It then generates a CUSUM chart for visualizing the stability of a process based on a Poisson distribution. The reference value 'k' can be provided; if not, it defaults to half of the process average lambda.

Usage

```
cv17_cusum(data, residue_col, cleaning_event_col, k = NULL)
```

Arguments

`data`	A data frame containing the dataset for analysis.
`residue_col`	The name of the column representing residue data.
`cleaning_event_col`	
	The name of the column representing cleaning events.
`k`	The reference value used in calculating CUSUM, by default it is set to half of lambda.

Value

A ggplot object representing the CUSUM chart.

Author(s)

Chan, Mohamed, Lou, Wendy, Yang, Xiande [xiande.yang at gmail.com]

Examples

```
# To create a CUSUM chart with default k value
cv17_cusum(data = Eq_Mic, residue_col = "Mic", cleaning_event_col = "CleaningEvent")

# To create a CUSUM chart with a specified k value
cv17_cusum(data = Eq_Mic, residue_col = "Mic", cleaning_event_col = "CleaningEvent", k = 0.75)
```

---

cv18_ewma	*Exponentially Weighted Moving Average (EWMA) Chart*

Description

Generates an EWMA chart for a specified residue column grouped by cleaning events.

Usage

```
cv18_ewma(data, residue_col, cleaning_event_col, alpha = 0.2)
```

Arguments

`data`	A data frame containing the data set for analysis.
`residue_col`	The name of the column representing residue data.
`cleaning_event_col`	
	The name of the column representing cleaning events.
`alpha`	The smoothing parameter for the EWMA calculation, default is 0.2.

Value

A ggplot object representing the EWMA chart.

Author(s)

Chan, Mohamed, Lou, Wendy, Yang, Xiande [xiande.yang at gmail.com]

Examples

```
# Assuming 'Eq_Mic' is a data frame, 'Mic' is the residue column of interest,
# and 'CleaningEvent' is the column representing cleaning events.
ewma_plot <- cv18_ewma(data = Eq_Mic, residue_col = "Mic", cleaning_event_col = "CleaningEvent")
print(ewma_plot)
```

---

cv19_poisson_simple   *Poisson Fixed Effect Model Summary*

Description

Fits a simple Poisson model to the data and returns a data frame containing the model's term, estimate, standard error, z value, and p-value, formatted to a fixed number of decimal places.

Usage

```
cv19_poisson_simple(data, residue_col)
```

Arguments

data   A data frame containing the data set for analysis.

residue_col   The name of the column representing residue data.

Value

A data frame with the formatted summary of the Poisson regression model.

Author(s)

Chan, Mohamed, Lou, Wendy, Yang, Xiande [xiande.yang at gmail.com]

Examples

```
# Assuming 'Eq_Mic' is a data frame and 'Mic' is the residue column of interest.
cv19_poisson_simple(data = Eq_Mic, residue_col = "Mic")
```

cv20_poisson_fixed *Poisson Fixed Effect Model*

Description

Fits a fixed effects Poisson model and returns a data frame with the summary. If the p-value is significant, then the corresponding cleaning event is significantly different from other cleaning events. For a stable cleaning process, we wish all p-values are not significant.

Usage

```
cv20_poisson_fixed(data, residue_col, cleaning_event_col)
```

Arguments

data   Data frame containing the data.

residue_col   The name of the residue column.

cleaning_event_col

                    The name of the cleaning event column.

Value

A data frame output with the fixed effect summary.

Author(s)

Chan, Mohamed, Lou, Wendy, Yang, Xiande [xiande.yang at gmail.com]

Examples

```
fixed_effect_summary <- cv20_poisson_fixed(data = Eq_Mic, residue_col = "Mic",
cleaning_event_col = "CleaningEvent")
```

cv21_poisson_mixed *Poisson Mixed Effect Model Summary*

Description

Fits a mixed-effects Poisson model to the data and returns a data frame containing the fixed effect part estimates, standard errors, z-values, and p-values.

Usage

```
cv21_poisson_mixed(data, residue_col, cleaning_event_col)
```

Arguments

`data`	A data frame containing the data set for analysis.
`residue_col`	A string specifying the column in 'data' that contains residue data.
`cleaning_event_col`	
	A string specifying the column in 'data' for random effects grouping.

Value

A data frame with the fixed effect summary of the mixed-effects Poisson regression model.

Author(s)

Chan, Mohamed, Lou, Wendy, Yang, Xiande [xiande.yang at gmail.com]

Examples

```
# Assuming 'Eq_Mic' is a data frame, 'Mic' is the residue column of interest,
# and 'CleaningEvent' is the column for random effects grouping.
mixed_effect_summary <- cv21_poisson_mixed(data = Eq_Mic, residue_col = "Mic",
cleaning_event_col = "CleaningEvent")
print(mixed_effect_summary)
```

---

cv22_var_random_effect

Extract Variance of Random Effects

Description

This function fits a Poisson mixed-effects model with a specified random effect and extracts the variances and standard deviations of the random effects.

Usage

```
cv22_var_random_effect(data, residue_col, cleaning_event_col)
```

Arguments

`data`	A data frame containing the data.
`residue_col`	The name of the residue column.
`cleaning_event_col`	
	The name of the column used for random effects grouping.

Value

A data frame with the variances and standard deviations of the random effects.

Author(s)

Chan, Mohamed, Lou, Wendy, Yang, Xiande [xiande.yang at gmail.com]

Examples

```
ra_table <- cv22_var_random_effect(data=Eq_Mic, residue_col="Mic",
cleaning_event_col="CleaningEvent")
```

---

cv23_random_effect_coef

Extract Random Effect Coefficients

Description

This function fits a Poisson mixed-effects model with a specified random effect and extracts the random effect coefficients and their standard deviations.

Usage

```
cv23_random_effect_coef(data, residue_col, cleaning_event_col)
```

Arguments

data   A data frame containing the data.

residue_col   The name of the residue column.

cleaning_event_col

                   The name of the column used for random effects grouping.

Value

A data frame with the random effect coefficients and standard deviations.

Author(s)

Chan, Mohamed, Lou, Wendy, Yang, Xiande [xiande.yang at gmail.com]

Examples

```
re_coefs <- cv23_random_effect_coef(data=Eq_Mic, residue_col="Mic",
cleaning_event_col="CleaningEvent")
```

`cv24_vca_mic` *Variance Component Analysis for Microbial Counts*

Description

This function performs a variance component analysis using a mixed-effects model with a Poisson distribution to estimate within-group and between-group variance for microbial counts data. Assumes data is grouped by cleaning events and evaluates the residue or microbial counts within these groups.

Usage

```
cv24_vca_mic(data, residue_col, cleaning_event_col)
```

Arguments

data   A data frame containing the dataset.

residue_col   The name of the column in 'data' that contains the residue or microbial counts.

cleaning_event_col

                The name of the column in 'data' that contains the grouping factor for cleaning events.

Value

A data frame summarizing the variance components, including within-group variance, between-group variance, and total variance, along with their percentages and standard deviations.

Author(s)

Chan, Mohamed, Lou, Wendy, Yang, Xiande [xiande.yang at gmail.com]

Examples

```
# Assuming `Eq_Mic` is your dataframe, `Mic` is the microbial counts column,
# and `CleaningEvent` is the cleaning event column:
cv24_vca_mic(Eq_Mic, "Mic", "CleaningEvent")
```

---

cv25_qbinom_ppu   *Binomial Process Performance Calculation*

Description

Performs a process performance calculation using binomial distribution. This includes a bootstrap procedure to estimate the confidence interval of the Process Performance Index (Ppu).

Usage

```
cv25_qbinom_ppu(data, residue_col, cleaning_event_col, usl_col)
```

Arguments

data	A data frame containing the dataset.
residue_col	Name of the column in 'data' that contains the residue or defect counts.
cleaning_event_col	
	Name of the column in 'data' that groups data by cleaning event.
usl_col	Name of the column in 'data' that contains the Upper Specification Limit (USL) for each group.

Value

A data frame with the calculated Ppu and its 95 along with the method used ("Q-Binomial").

Author(s)

Chan, Mohamed, Lou, Wendy, Yang, Xiande [xiande.yang at gmail.com]

Examples

```
# Assuming `data` is the dataframe with columns "Residue", "CleaningEvent", and "USL":
cv25_qbinom_ppu(Eq_Mic, "Mic", "CleaningEvent", "USL")
```

---

cv26_qpoisson_ppu   *Calculate Process Performance Index using Poisson Distribution*

Description

This function calculates the Process Performance Index (Ppu) for data assumed to follow a Poisson distribution. It includes a bootstrap method for estimating the confidence interval of the Ppu.

Usage

```
cv26_qpoisson_ppu(data, residue_col, cleaning_event_col, usl_col)
```

Arguments

`data`	A data frame containing the dataset.
`residue_col`	Name of the column in 'data' containing residue counts.
`cleaning_event_col`	
	Name of the column in 'data' used to group data by cleaning event.
`usl_col`	Name of the column in 'data' that contains the Upper Specification Limit (USL) for each group.

Value

A data frame with columns Method, Ppu, CI_Lower, and CI_Upper.

Author(s)

Chan, Mohamed, Lou, Wendy, Yang, Xiande [xiande.yang at gmail.com]

Examples

```
cv26_qpoisson_ppu(Eq_Mic, "Mic", "CleaningEvent", "USL")
```

`cv27_anscombe_ppu`	*Process Performance Calculation using Anscombe's Transformation*

Description

Calculates the Process Performance Index (Ppu) using Anscombe's transformation. This function also performs a bootstrap to estimate the confidence interval of Ppu.

Usage

```
cv27_anscombe_ppu(data, residue_col, cleaning_event_col, usl_col)
```

Arguments

`data`	A data frame containing the dataset.
`residue_col`	Name of the column in 'data' containing residue or defect counts.
`cleaning_event_col`	
	Name of the column in 'data' used for grouping by cleaning event.
`usl_col`	Name of the column in 'data' that contains the Upper Specification Limit (USL).

Value

A data frame with columns for the Method, Ppu, CI_Lower, and CI_Upper.

Author(s)

Chan, Mohamed, Lou, Wendy, Yang, Xiande [xiande.yang at gmail.com]

Examples

```
cv27_anscombe_ppu(Eq_Mic, "Mic", "CleaningEvent", "USL")
```

---

cv28_freeman_ppu *Calculate Ppu using Freeman's Transformation*

Description

This function calculates the Process Performance Index (Ppu) using Freeman's transformation, including a bootstrap method to estimate the confidence interval of Ppu.

Usage

```
cv28_freeman_ppu(data, residue_col, cleaning_event_col, usl_col)
```

Arguments

`data`	A data frame containing the dataset.
`residue_col`	The name of the column in 'data' containing residue or defect counts.
`cleaning_event_col`	
	The name of the column in 'data' used for grouping data by cleaning event.
`usl_col`	The name of the column in 'data' that contains the Upper Specification Limit (USL).

Value

A data frame with columns for the Method, Ppu, CI_Lower, and CI_Upper.

Author(s)

Chan, Mohamed, Lou, Wendy, Yang, Xiande [xiande.yang at gmail.com]

Examples

```
cv28_freeman_ppu(Eq_Mic, "Mic", "CleaningEvent", "USL")
```

cv29_mic_ppu	*Calculate Mic Ppu with Five Methods*

Description

This function calculates the process performance index (Ppu) for Mic using five different methods, including Q-Binomial, Q-Poisson, Anscombe, Freeman, and KDE. It returns a dataframe with the Ppu values, lower and upper confidence intervals for each method, and appends a row for the method with the minimum Ppu value.

Usage

```
cv29_mic_ppu(data, residue_col, cleaning_event_col, usl_col)
```

Arguments

data	A dataframe containing the dataset.
residue_col	The name of the column in 'data' that contains the residue values.
cleaning_event_col	
	The name of the column in 'data' that contains the cleaning event identifiers.
usl_col	The name of the column in 'data' that contains the Upper Specification Limit values.

Value

A dataframe with the Ppu calculations for each method and the minimum Ppu method.

Author(s)

Chan, Mohamed, Lou, Wendy, Yang, Xiande [xiande.yang at gmail.com]

Examples

```
MicPPU <- cv29_mic_ppu(Eq_Mic, "Mic", "CleaningEvent", "USL")
```

---

cv30_dar_car_mic_ppu *Calculate DAR, CAR, and Mic Ppu Values and Identify the Overall Minimum*

Description

This function calculates Ppu values for DAR, CAR, and Mic using the KDE method provided by the 'cv12_kde_ppu' function. It then uses the 'cv29_mic_ppu' function to calculate combined Ppu for Mic and extract the method with the minimum Ppu value. The function assumes the availability of the datasets 'Eq_DAR', 'Eq_CAR', and 'Eq_Mic' that conform to expected column naming conventions and data structures. It is reliant on the results of the 'cv12_kde_ppu' and 'cv29_mic_ppu' functions being consistent and correctly formatted.

Usage

```
cv30_dar_car_mic_ppu(
  dar_data,
  dar_residue_col,
  dar_cleaning_event_col,
  dar_usl_col,
  car_data,
  car_residue_col,
  car_cleaning_event_col,
  car_usl_col,
  mic_data,
  mic_residue_col,
  mic_cleaning_event_col,
  mic_usl_col
)
```

Arguments

dar_data	A dataframe containing DAR data.
dar_residue_col	
	The name of the DAR residue column.
dar_cleaning_event_col	
	The name of the DAR cleaning event identifier column.
dar_usl_col	The name of the DAR Upper Specification Limit column.
car_data	A dataframe containing CAR data.
car_residue_col	
	The name of the CAR residue column.
car_cleaning_event_col	
	The name of the CAR cleaning event identifier column.
car_usl_col	The name of the CAR Upper Specification Limit column.
mic_data	A dataframe containing Mic data.

```
mic_residue_col
                The name of the Mic residue column.
mic_cleaning_event_col
                The name of the Mic cleaning event identifier column.
mic_usl_col      The name of the Mic Upper Specification Limit column.
```

Value

A dataframe with Ppu values for DAR, CAR, and Mic, along with the Overall Minimum Ppu.

Author(s)

Chan, Mohamed, Lou, Wendy, Yang, Xiande [xiande.yang at gmail.com]

Examples

```
Eq_DAR <- cv03_usl_unification(data = Eq_DAR, cleaning_event_col = "CleaningEvent",
 residue_col = "DAR", usl_col = "USL")
 Eq_CAR <- cv03_usl_unification(data = Eq_CAR, cleaning_event_col = "CleaningEvent",
 residue_col = "CAR", usl_col = "USL")
  df1 <- cv30_dar_car_mic_ppu(
    dar_data = Eq_DAR,
    dar_residue_col = "DAR_Pct",
    dar_cleaning_event_col = "CleaningEvent",
    dar_usl_col = "USL_Pct",
    car_data = Eq_CAR,
    car_residue_col = "CAR_Pct",
    car_cleaning_event_col = "CleaningEvent",
    car_usl_col = "USL_Pct",
    mic_data = Eq_Mic,
    mic_residue_col = "Mic",
    mic_cleaning_event_col = "CleaningEvent",
    mic_usl_col = "USL")
```

Eq_CAR   *Equipment Cleaning Data for CAR*

Description

A dataset containing cleaning validation data for equipment CAR.

Usage

```
Eq_CAR
```

Format

A data frame with 30 rows and 3 variables:

CAR Numeric vector with CAR measurements.

USL Numeric vector with Upper Specification Limits for CAR.

CleaningEvent Factor vector with Cleaning Event identifiers.

Classification Character vector with the deviation status for each cleaning event. Defaults to "normal".

LIMSProjectID Integer or character vector with unique project IDs assigned to each row.

Source

Details about the data source.

Eq_DAR   *Equipment Cleaning Data for DAR*

Description

A dataset containing cleaning validation data for equipment DAR.

Usage

Eq_DAR

Format

A data frame with 60 rows and 3 variables:

DAR Numeric vector with DAR measurements.

USL Numeric vector with Upper Specification Limits for DAR.

CleaningEvent Factor vector with Cleaning Event identifiers.

Classification Character vector with the deviation status for each cleaning event. Defaults to "normal".

LIMSProjectID Integer or character vector with unique project IDs assigned to each row.

Source

Details about the data source.

Eq_Mic   *Equipment Cleaning Data for Microbial Bioburden*

Description

A dataset containing cleaning validation data for microbial bioburden (Mic).

Usage

```
Eq_Mic
```

Format

A data frame with 20 rows and 3 variables:

Mic Numeric vector with Mic measurements.

USL Numeric vector with Upper Specification Limits for Mic.

CleaningEvent Factor vector with Cleaning Event identifiers.

Classification Character vector with the deviation status for each cleaning event. Defaults to "normal".

LIMSProjectID Integer or character vector with unique project IDs assigned to each row.

Source

Details about the data source.

Index