Package ‘DBERlibR’

December 6, 2022

Type Package

Version 0.1.3

Title Automated Assessment Data Analysis for Discipline-Based Education Research

Author Changsoo Song [aut, cre],
Resa Helikar [aut],
Wendy Smith [aut],
Tomas Helikar [aut]

Maintainer Changsoo Song <csong7@unl.edu>

Description Discipline-Based Education Research scientists repeatedly analyze assessment data to ensure question items’ reliability and examine the efficacy of a new educational intervention. Analyzing assessment data comprises multiple steps and statistical techniques that consume much of researchers’ time and are error-prone. While education research continues to grow across many disciplines of science, technology, engineering, and mathematics (STEM), the discipline-based education research community lacks tools to streamline education research data analysis. ‘DBERlibR’—an ‘R’ package to streamline and automate assessment data processing and analysis—fills this gap. The package reads user-provided assessment data, cleans them, merges multiple datasets (as necessary), checks assumption(s) for specific statistical techniques (as necessary), applies various statistical tests (e.g., one-way analysis of covariance, one-way repeated-measures analysis of variance), and presents and interprets the results all at once. By providing the most frequently used analytic techniques, this package will contribute to education research by facilitating the creation and widespread use of evidence-based knowledge and practices. The outputs contain a sample interpretation of the results for users’ convenience. User inputs are minimal; they only need to prepare the data files as instructed and type a function in the 'R' console to conduct a specific data analysis.\n For descriptions of the statistical methods employed in package, refer to the following Encyclopedia of Research Design, edited by Salkind, N. (2010) <doi:10.4135/9781412961288>.

Imports car, dplyr, emmeans, ggplot2, ggpubr, ggrepel, psych, readr, reshape, rstatix, tibble

License AGPL-3

Encoding UTF-8

RoxygenNote 7.2.2

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2022-12-06 13:52:31 UTC

R topics documented:

`demo_group_diff` *Demographic Group Differences*

Description

This function automatically combines demographic variables to a dataset, and runs the analysis of variance (ANOVA) with assumptions check to examine demographic sub-group differences all at once. Please make sure to name data files accurately and have them saved in the working directory.

Usage

```
demo_group_diff(score_csv_data, group_csv_data, m_cutoff = 0.15, group_name)
```

Arguments

`score_csv_data`	This function requires a csv data file. Its name (e.g., "data_treat_pre.csv") can be passed as an argument. Make sure to set the folder with the data file(s) as the working directory.
`group_csv_data`	This function requires a csv data file. Its name (e.g., "demographic_data.csv") can be passed as an argument. Make sure to set the folder with the data file(s) as the working directory.
`m_cutoff`	This package will treat skipped answers as incorrect. However, too many skipped answers may skew the results of the data analysis. User can can provide a cutoff for the proportion of skipped answers. For example, if the user enters 0.1, students who skipped more than 10 percent of the answers will be excluded from the data analysis to prevent skewed results. The default of 0.15 is commonly applied as a rule of thumb.
`group_name`	This function requires a group name as indicated in the csv data file (e.g., "gender", "grade")

Value

This function returns a tibble() including the following information:

- n_students_deleted: Number of students deleted from the data for analysis based on the percentage obtained via the argument of m_cutoff
- descriptive_statistics: Descriptive statistics
- boxplots: Boxplots - visual presentation of the descriptive statistics
- shapiro_wilk_test: Shapiro-Wilk test results to determine normality of residuals
- normal_qq_plot: The normal q-q plot to visually inspect the normality of residuals
- levene_test: Test homogeneity of variances
- one_way_anova: Results of the one-way anova with equal variances assumed
- one_way_anova_pwc: Pairwise t-test results for the one-way ANOVA with equal variances assumed
- welch_anova_test: Results of the one-way ANOVA with unequal variance
- games_howell_test: Pairwise t-test results for the one-way ANOVA with unequal variances assumed
- kruskal_wallis_test: Results of the Kruskal-Wallis test (non- parametric version of the one-way ANOVA)
- kruskal_wallis_test_pwc: Pairwise t-test results for the Kruskal-Wallis test

Examples

```
# Run the following codes directly in the console panel. The plots
# generated through the link above may be displaced depending on the screen
# resolution.
demo_group_diff(score_csv_data =
        system.file("extdata", "data_treat_pre.csv", package = "DBERlibR"),
        group_csv_data =
        system.file("extdata", "demographic_data.csv", package = "DBERlibR"),
        m_cutoff = 0.15,
        group_name = "grade")
```

independent_samples *Independent Samples Data Analysis*

Description

This function automatically cleans the datasets (e.g., converting missing values to "0), binds treatment-control group datasets, check assumptions, and then runs the Independent Samples T-test (parametric) and Mann–Whitney U test (nonparametric) to help you examine the difference between the groups. R scripts and their outputs are as follows (just pay attention to the outputs since the codes are automatically run back-end by the function).

Usage

```
independent_samples(treat_csv_data, ctrl_csv_data, m_cutoff = 0.15)
```

Arguments

`treat_csv_data` This function requires a csv file with treatment group data. Its name (e.g., "data_treat_post.csv") can be passed as an argument. Make sure to set the folder with the data file(s) as the working directory.

`ctrl_csv_data` This function requires a csv file with control group data. Its name (e.g., "data_ctrl_post.csv") can be passed as an argument. Make sure to set the folder with the data file(s) as the working directory.

`m_cutoff` This package will treat skipped answers as incorrect. However, too many skipped answers may skew the results of the data analysis. User can can provide a cutoff for the proportion of skipped answers. For example, if the user enters 0.1, students who skipped more than 10 percent of the answers will be excluded from the data analysis to prevent skewed results. The default of 0.15 is commonly applied as a rule of thumb.

Value

This function returns a `tibble()` including the following information:

- `n_students_deleted`: Number of students deleted from the data for analysis based on the percentage obtained via the argument of m_cutoff
- `descriptive_statistics`: Descriptive statistics
- `boxplots`: Boxplots - visual presentation of the descriptive statistics
- `shapiro_wilk_test`: Shapiro-Wilk test results to determine normality
- `normal_qq_plot`: The normal q-q plot to visually inspect the normality
- `independent_samples_t_test_equal`: Results of the independent samples t-test with equal variances assumed
- `independent_samples_t_test_unequal`: Results of the independent samples t-test with unequal variances assumed
- `mann_whitney_u_test`: Results of the Mann-Whitney U test

Examples

```
# Run the following codes directly in the console panel. The plots
# generated through the link above may be displaced depending on the screen
# resolution.
independent_samples(treat_csv_data =
         system.file("extdata", "data_treat_post.csv", package = "DBERlibR"),
         ctrl_csv_data =
         system.file("extdata", "data_ctrl_post.csv", package = "DBERlibR"),
         m_cutoff = 0.15)
```

item_analysis *Item Analysis*

Description

This function automatically reads and cleans the data (e.g., converting missing values to "0"), and calculates difficulty and discriminant scores.

Usage

```
item_analysis(score_csv_data, m_cutoff = 0.15)
```

Arguments

`score_csv_data`	data This function requires a csv data file. Its name (e.g., "data_treat_pre.csv") can be passed as an argument. Make sure to set the folder with the data file(s) as the working directory.
`m_cutoff`	This package will treat skipped answers as incorrect. However, too many skipped answers may skew the results of the data analysis. User can can provide a cutoff for the proportion of skipped answers. For example, if the user enters 0.1, students who skipped more than 10 percent of the answers will be excluded from the data analysis to prevent skewed results. The default of 0.15 is commonly applied as a rule of thumb.

Value

This function returns a `tibble()` including the following information:

- `n_students_deleted`: Number of students deleted from the data for analysis based on the percentage obtained via the argument of m_cutoff
- `difficulty_index`: Calculated difficulty scores
- `difficulty_index_plot`: Plot of difficulty scores in the ascending order
- `too_difficulty_items`: List of items of which difficulty score is less than 0.2
- `discrimination_index`: Calculated discrimination scores
- `discrimination_index_plot`: Plot of discrimination scores in the ascending order
- `non_discrimination_items`: List of items of which discrimination score is less than 0.2

Examples

```
# Run the following codes directly in the console panel. The plots
# generated through the link above may be displaced depending on the screen
# resolution.
item_analysis(score_csv_data =
         system.file("extdata", "data_treat_pre.csv", package = "DBERlibR"),
         m_cutoff = 0.15)
```

one_way_ancova *One-way ANCOVA*

Description

This function automatically merges pre-post data sets, binds treatment-control data sets, checks assumptions of one-way ANCOVA, and then runs the main One-way ANCOVA all at once. Please make sure to name data files accurately (i.e., "data_treat_pre.csv", "data_treat_post.csv", "data_ctrl_pre.csv"," data_ctrl_post.csv") and have them saved in the working directory.

Usage

```
one_way_ancova(
  treat_pre_csv_data,
  treat_post_csv_data,
  ctrl_pre_csv_data,
  ctrl_post_csv_data,
  m_cutoff = 0.15
)
```

Arguments

`treat_pre_csv_data`
: This function requires a csv file with treatment group's pre-test data. Its name (e.g., "data_treat_pre.csv") can be passed as an argument. Make sure to set the folder with the data file(s) as the working directory.

`treat_post_csv_data`
: This function requires a csv file with treatment group's post-test data. Its name (e.g., "data_treat_post.csv") can be passed as an argument. Make sure to set the folder with the data file(s) as the working directory.

`ctrl_pre_csv_data`
: This function requires a csv file with control group's pre-test. Its name (e.g., "data_ctrl_pre.csv") can be passed as an argument. Make sure to set the folder with the data file(s) as the working directory.

`ctrl_post_csv_data`
: This function requires a csv file with control group's post-test data. Its name (e.g., "data_ctrl_post.csv") can be passed as an argument. Make sure to set the folder with the data file(s) as the working directory.

`m_cutoff`
: This package will treat skipped answers as incorrect. However, too many skipped answers may skew the results of the data analysis. User can can provide a cutoff for the proportion of skipped answers. For example, if the user enters 0.1, students who skipped more than 10 percent of the answers will be excluded from the data analysis to prevent skewed results. The default of 0.15 is commonly applied as a rule of thumb.

Value

This function returns a `tibble()` including the following information:

- `n_students_deleted`: Number of students deleted from the data for analysis based on the percentage obtained via the argument of m_cutoff
- `pre_descriptive_statistics`: Pre-test data descriptive statistics
- `post_descriptive_statistics`: Post-test data descriptive statistics
- `boxplots`: Boxplots - visual presentation of the descriptive statistics
- `scatter_plot`: Scatter plots to test the linearity statistics
- `shapiro_wilk_test`: Shapiro-Wilk test results to determine normality of residuals
- `normal_qq_plot`: The normal q-q plot to visually inspect the normality of residuals
- `levene_test`: Test homogeneity of variances
- `independent_samples_t_test_equal`: Results of the independent samples t-test with equal variances assumed
- `one_way_ancova`: Results of the one-way ANCOVA
- `estimated_marginal_means`: Estimated marginal means

Examples

```
# Run the following codes directly in the console panel. The plots
# generated through the link above may be displaced depending on the screen
# resolution.
one_way_ancova(treat_pre_csv_data =
         system.file("extdata", "data_treat_pre.csv", package = "DBERlibR"),
         treat_post_csv_data =
         system.file("extdata", "data_treat_post.csv", package = "DBERlibR"),
         ctrl_pre_csv_data =
         system.file("extdata", "data_ctrl_pre.csv", package = "DBERlibR"),
         ctrl_post_csv_data =
         system.file("extdata", "data_ctrl_post.csv", package = "DBERlibR"),
         m_cutoff = 0.15)
```

`one_way_repeated_anova`

One-way Repeated Measures ANOVA

Description

This function automatically merges pre, post, and post2 datasets, and then runs the one-way repeated measures ANOVA with assumptions check all at once. Please make sure to name data files accurately (i.e., “data_treat_pre.csv”, “data_treat_post.csv”, and “data_treat_post2.csv”) and have them saved in the working directory.

Usage

```
one_way_repeated_anova(
  treat_pre_csv_data,
  treat_post_csv_data,
  treat_post2_csv_data,
  m_cutoff = 0.15
)
```

Arguments

`treat_pre_csv_data`
: This function requires a csv file with treatment group's pre-test data. Its name (e.g., "data_treat_pre.csv") can be passed as an argument. Make sure to set the folder with the data file(s) as the working directory.

`treat_post_csv_data`
: This function requires a csv file with treatment group's post-test data. Its name (e.g., "data_treat_post.csv") can be passed as an argument. Make sure to set the folder with the data file(s) as the working directory.

`treat_post2_csv_data`
: This function requires a csv file with treatment group's post2-test. Its name (e.g., "data_treat_post2.csv") can be passed as an argument. Make sure to set the folder with the data file(s) as the working directory.

`m_cutoff`
: This package will treat skipped answers as incorrect. However, too many skipped answers may skew the results of the data analysis. User can can provide a cutoff for the proportion of skipped answers. For example, if the user enters 0.1, students who skipped more than 10 percent of the answers will be excluded from the data analysis to prevent skewed results. The default of 0.15 is commonly applied as a rule of thumb.

Value

This function returns a `tibble()` including the following information:

- `n_students_deleted`: Number of students deleted from the data for analysis based on the percentage obtained via the argument of m_cutoff
- `descriptive_statistics`: Descriptive statistics
- `boxplots`: Boxplots - visual presentation of the descriptive statistics
- `shapiro_wilk_test`: Shapiro-Wilk test results to determine normality of residuals
- `normal_qq_plot`: The normal q-q plot to visually inspect the normality of residuals
- `one_way_repeated_anova`: Results of the one-way repeated measures ANOVA
- `one_way_repeated_anova_pwc`: Pairwise t-test results for the one-way repeated measures ANOVA
- `friedman_test`: Results of the friedman test
- `friedman_pwc`: Pairwise t-test results for the Friedman test

Examples

```
# Run the following codes directly in the console panel. The plots
# generated through the link above may be displaced depending on the screen
# resolution.
one_way_repeated_anova(treat_pre_csv_data =
       system.file("extdata", "data_treat_pre.csv", package = "DBERlibR"),
       treat_post_csv_data =
       system.file("extdata", "data_treat_post.csv", package = "DBERlibR"),
       treat_post2_csv_data =
       system.file("extdata", "data_treat_post2.csv", package = "DBERlibR"),
       m_cutoff = 0.15)
```

paired_samples *Paired Samples Data Analysis*

Description

This function automatically cleans the datasets (e.g., converting missing values to "0), merges pre-post datasets, checks assumptions, and then runs the (parametric) Paired Samples T-test and (non-parametric) Wilcoxon Signed-Rank test to help you examine the difference between pre-post scores.

Usage

```
paired_samples(pre_csv_data, post_csv_data, m_cutoff = 0.15)
```

Arguments

`pre_csv_data` This function requires a csv file with pre-test data. Its name (e.g., "data_treat_pre.csv") can be passed as an argument. Make sure to set the folder with the data file(s) as the working directory.

`post_csv_data` This function requires a csv file with post-test data. Its name (e.g., "data_treat_post.csv") can be passed as an argument. Make sure to set the folder with the data file(s) as the working directory.

`m_cutoff` This package will treat skipped answers as incorrect. However, too many skipped answers may skew the results of the data analysis. User can can provide a cutoff for the proportion of skipped answers. For example, if the user enters 0.1, students who skipped more than 10 percent of the answers will be excluded from the data analysis to prevent skewed results. The default of 0.15 is commonly applied as a rule of thumb.

Value

This function returns a `tibble()` including the following information:

- `n_students_deleted`: Number of students deleted from the data for analysis based on the percentage obtained via the argument of m_cutoff

- shapiro_wilk_test: Shapiro-Wilk test results to determine normality
- normal_qq_plot: The normal q-q plot to visually inspect the normality
- descriptive_statistics: Descriptive statistics
- boxplots: Boxplots - visual presentation of the descriptive statistics
- paired_samples_t_test: Paired samples t-test results
- wilcoxon_signed_rank_test: Wilcoxon signed rank test results

Examples

```
# Run the following codes directly in the console panel. The plots
# generated through the link above may be displaced depending on the screen
# resolution.
paired_samples(pre_csv_data =
         system.file("extdata", "data_treat_pre.csv", package = "DBERlibR"),
         post_csv_data =
         system.file("extdata", "data_treat_post.csv", package = "DBERlibR"),
         m_cutoff = 0.15)
```

Index