Package 'StatCharrms'

May 1, 2020

Version 0.90.95

Date 2020-4-30

Title Statistical Analysis of Chemistry, Histopathology, and
Reproduction Endpoints Including Repeated Measures and
Multi-Generation Studies

Depends R (>= 3.1.0)

Imports RGtk2, R2HTML, gWidgets2, gWidgets2RGtk2, multcomp, nlme,
lattice, cairoDevice, car, clinfun, survival, coxme, methods,
RSCABS

SystemRequirements GTK+ (>= 2.8.0)

ByteCompile no

LazyLoad yes

LazyData yes

Description A front end for the statistical analyses involved in the tier II endocrine
disruptor screening program. The analyses available to this package are:
Rao-Scott adjusted Cochran-Armitage test for trend By Slices (RSCABS),
a Standard Cochran-Armitage test for trend By Slices (SCABS),
mixed effects Cox proportional model, Jonckheere-Terpstra step down trend test
Dunn test, one way ANOVA, weighted ANOVA, mixed effects ANOVA, repeated
measures ANOVA, and Dunnett test.

License CC0

URL https://CRAN.R-project.org/package=StatCharrms

Author Joe Swintek [aut, cre],
Kevin Flynn [ctb],
Jon Haselman [ctb],
Maki Sakurai [ctb]

Maintainer Joe Swintek <swintek.joe@epa.gov>

RoxygenNote 7.1.0

NeedsCompilation no

Repository CRAN

Date/Publication 2020-04-30 22:50:02 UTC

R **topics documented:**

StatCharrms-package *StatCharrms*

Description

Statistical analysis of Chemistry, Histopathology, and Reproduction endpoints including Repeated measures and Multi-generation Studies (StatCharrms). Based on the SAS build of StatCharrms written by John Green, the R version of StatCharrms was developed for the statistical analyses needed for the tier II endocrine disruptor screening program. StatCharrms was funded by the USEPA under Contract EP-D-13-052.

Details

- **Package:** StatCharrms
- **Type:** Package \ Frontend
- **Version:** 0.96-00
- **Date:** 2016-06-20
- **License:** CC0

Author(s)

Joe Swintek <swintek.joe@epa.gov> as author, Kevin Flynn as a tester, Jon Haselman as a tester.

Maintainer: Joe Swintek <swintek.joe@epa.gov>

References

OECD Guidelines for the Testing of Chemicals, Section 2. Test No. 240: Medaka Extended One Generation Reproduction Test (MEOGRT)

Examples

```
## Not run:
Run.StatCharrms()
generateExamples()

## End(Not run)
```

---

addMultiSpec.std   *Add a data specification tab*

Description

This function adds a tab to the analysis main window where details about the data can be specified.

Usage

```
addMultiSpec.std(Notebook)
```

Arguments

Notebook   The notebook the specification tab is added to.

Value

These items are available from the start of the data specification tab.

`Treatment`	Used to specify the name of the treatment variable. This must be selected for any analysis to run.
`Replicate`	Used to specify the name of the replicate variable.
`Generation Variable`	
	Used to specify the name of the generation variable.
`Generation Value`	
	The generation the selected analysis is performed on.
`Age Variable`	Used to specify the name of the age variable.
`Age Value`	The age the selected analysis is performed on.
`Gender Variable`	
	Used to specify the name of the gender variable.
`Gender Value`	The gender the selected analysis is performed on.
`Anova Weights`	The name of the variable that contains weights for a weighted ANOVA.
`Test Direction`	The direction of the hypothesis test.
`Alpha Level`	The alpha level used in the Jonckheere-Terpstra test from the jonckheereTerpstraTest function.
`Select Endpoints To Test`	
	Brings up a window that allows for the selection of each endpoint to be tested.

These item are relevant to repeated measures taken on the same subject at different times throughout the study:

`Time Variable`	Used to specify the time variable.
`Time Format`	Used to specify the format the time variable is in.

Analysis Interval

> This is used to specify the length of time a response is averaged over for an analysis.

Graph Interval This is used to specify the length of time a response is averaged over for the purpose of graphing.

Select Excluded Times

> Used to select dates or times to exclude from the analysis.

Excluded Times Displays all the times selected to be excluded from the anylsis.

After the Select [**Endpoints To Test**] button is pressed a new frame for each selected endpoint will appear.

Test Type This will select the test type. If nothing is selected the test type defaults to 'Auto'. See `autoStdAnylsis` and `forceStdAnalysis` for further details.

Transformation

> This will select the type of transformation the specific endpoint in the data set undergoes. If nothing is selected the test type defaults to 'None'.

Author(s)

Joe Swintek

See Also

`autoStdAnylsis` and `forceStdAnalysis`.

---

addSpec.te *Add the specification widgets for time to event analysis*

Description

This handles all the GUI widgets needed to specify time to event data.

Usage

```
addSpec.te()
```

Value

Treatment Used to specify the name of the treatment variable. This must be selected for any analysis to run.

Replicate Used to specify the name of the replicate variable.

Gender Variable

> Used to specify the name of the gender variable.

Gender Value The gender the selected analysis is performed on.

Generation Variable

> Used to specify the name of the generation variable.

Generation Value

The generation the selected analysis is performed on.

Time Variable   Used to specify the name of the time variable. This must be selected for any analysis to run.

Status Variable

Used to specify the name of the event status variable. This must be selected for any analysis to run.

Event Value   Used to specify the name of the value used to signify the event the analysis is for. This must be selected for any analysis to run.

Censored Value   Used to specify the name of the value used to signify a censored event.

Author(s)

Joe Swintek

See Also

[Time2EventMain](#) and [analyseTime2Effect](#)

---

analyseTime2Effect   *Mixed effect time to event analysis*

Description

A wrapper function for the [coxme](#) function from the coxme package. Coxme is a mixed effects version of cox proportional models for comparison of time to event data taking into account censoring events.

Usage

```
analyseTime2Effect(Data, StatusVar, TimeVar, TreatmentVar, ReplicateVar)
```

Arguments

Data   A data set.

StatusVar   The name (as a string) of the status variable.

TimeVar   The name (as a string) of the time variable.

TreatmentVar   The name (as a string) of the treatment variable.

ReplicateVar   The name (as a string) of the replicate variable.

Value

Returns a list with the following values:

FitME	The results from [coxme]().
FitHP	The results from [coxph](). It is used in graphing.
FitS	The results from [survfit](). It is used in graphing.
EffectsTable	A dunnett table from [oneWayDunnettTest]().
MedianTable	A table showing the median time to effect along with the median's 95% confidence intervals. See [quantile.survfit]() for more details.

Author(s)

Joe Swintek

References

Ripatti and J Palmgren, Estimation of multivariate frailty models using penalized partial likelihood, *Biometrics* **56**:1016-1022, 2000.
T Therneau, P Grambsch and VS Pankratz, Penalized survival models and frailty, *J Computational and Graphical Statistics* **12**:156-175, 2003.

See Also

[coxme]()

Examples

```
#Data
data(eventTimeData)
#Run
analyseTime2Effect(Data=eventTimeData, StatusVar='Status',
TimeVar='Time', TreatmentVar='Trt', ReplicateVar='Rep')
```

---

autoStdAnylsis	*Run statistical tests according to OECD guidelines*

Description

Runs tests following the flow chart outlined in the OECD guidelines for the tier II tests for endocrine disruptors. The basic outline of the flow chart is as follows: 1) if the data is monotonic run a jonckheere Terpstra trend test, 2) is the data is not monotonic but is still normally distributed, run a Dunnett's test, 3) if the data is not monotonic and not normally distributed run a Dunns test.

Usage

```
autoStdAnylsis(Data, Response, TreatmentVar, Transform, WeightsVar, TimeVar,
TestDirection, ReplicateVar, AlphaLevel)
```

Arguments

`Data`	A data set.
`Response`	The name (as a string) of the response variable.
`TreatmentVar`	The name (as a string) of the treatment variable.
`Transform`	The name (as a string) of the transformation of the response variable. Can take on the values of c('Transformation', 'None', 'Log', log+1', 'Square_Root', 'Arcsin', 'Rank').
`WeightsVar`	The name (as a string) of the column containing the ANOVA weights.
`TimeVar`	The name (as a string) of the Time variable.
`TestDirection`	The direction of the hypothesis test. Can take on any of the values of: c('Decreasing', 'Increasing', 'Both').
`ReplicateVar`	The name (as a string) of the replicate variable.
`AlphaLevel`	The alpha level used in the Jonckheere-Terpstra test for trend. See ([jonckheereTerpstraTest](#)).

Value

See the following function for details on the analysis.

`Jonckheere`	Jonckheere-Terpstratest test see [jonckheereTerpstraTest](#).
`Dunns`	A Dunns test see [dunnsTest](#).
`Dunnet`	A Dunnet test see [oneWayDunnettTest](#).

Author(s)

Joe Swintek

References

OECD, 2015, Test No. 240: Medaka Extended One Generation Reproduction Test (MEOGRT), OECD Publishing, Paris. \ DOI: http://dx.doi.org/10.1787/9789264242258-en

---

basicAnova	*Perform a standard one way ANOVA.*

Description

A wrapper function for the [aov](#) function, which performs a standard ANOVA.

Usage

```
basicAnova(Data, Treatment, Response, WeightList)
```

Arguments

`Data`	A data set.
`Treatment`	The name (as a string) of the treatment variable.
`Response`	The name (as a string) of the response variable.
`WeightList`	The name (as a string) of the variables containing the weight for ANOVA. Set this value to NULL to indicate that weights are not being used.

Value

`Sum Sq`	The within treatment sum of squares.
`Df`	The degrees of freedom.
`F value`	The value of the F-statistic.
`Pr(>F)`	The p-value corresponding to the F statistic.
`Signif`	The significance flag for the where "." is a p-value > 0.05, "*" is a $0.01 <$ p-value $<= 0.05$, "**" for $0.001 <$ p-value $<= 0.01$, and "***" for p-value $<= 0.001$.

Author(s)

Joe Swintek

References

Chambers, J. M., Freeny, A and Heiberger, R. M. (1992) emphAnalysis of variance; designed experiments
Chapter 5 of Statistical Models in S eds J. M. Chambers and T. J. Hastie, Wadsworth & Brooks/Cole

See Also

[aov](#)

Examples

```
#Data
data(lengthWeightData)
#Subset the data
SubData<-lengthWeightData[lengthWeightData$Age=='16 week', ]
SubData<-SubData[SubData$Generation=='F1', ]
SubData<-SubData[SubData$SEX=='M', ]
#Run
basicAnova(Data=SubData, Treatment='Treatment',
Response='WEIGHT',WeightList=NULL)
```

---

buildResultsWindow *Build the results window for the analysis of numerical endpoints*

Description

This window is used to both display results from, and graph the data used in; the analysis of numerical endpoints.

Usage

```
buildResultsWindow(Results)
```

Arguments

Results   The results data structure from autoStdAnylsis or forceStdAnalysis.

Details

The results window has the potential to display the results from every possible type of analysis. See the help section of each analysis for more details.

Value

Every analyzed endpoint will have a tab associated with it. Every tab will contain these 2 buttons:

Save All Results

Creates a folder and saves all the results from every endpoint in the folder as an HTML document.

Save All Results and Graphs

Creates a folder and saves all the results from every endpoint in the folder as an HTML document. Also saves all possible graphs for every endpoint as pdfs in the same folder.

Each tab produces a table for every possible analysis. See each individual function for details about each table.

Summary table   See makeSummaryTable.

Test For Monotonicity

See monotonicityTest.

Jonckheere-Terpstra table

See jonckheereTerpstraTest.

Wilks table   See wilksTest.

Levene table   See leveneTestSC.

Anova table   See basicAnova.

Dunnet table   See oneWayDunnettTest.

Dunns table   See dunnsTest.

Williams Table   The results from the Williams test `williamsTest`.

In addition to the two save buttons shared by every tab, the graph tab has the additional button of:

Save Graph   Saves the current graph as a pdf.

The graph tab will always have the following controls:

Choose Response

       This allows for the selection of the endpoint to be graphed. This is can be any of the endpoints used in the analysis.

Choose Graph Type

       This allows for the selection of graph type. It can be a boxplot from the `boxplot` function, a quantile-quantile plot from the `qqnorm` function, or a violin plot from the `bwplot` function in the lattice package. If a time variable was specified, "Interaction" will be included as an additional option. Selecting "Interaction" produces an interaction plot using the `interaction.plot` function.

Use Transformed Scale

       This controls the scale the data is graphed in. The scale can be unmodified which uses the original, unmodified data or the scale can be modified which use the data that has been transformed by the transformation selected in the analysis.

If the time variable is specified, the graph will have these additional controls:

Group Variable   This controls what the x-axis is. It can be either the treatment variable used in the analysis or the time variable.

Time Interval   This controls the amount of time the response variable is averaged over. For example if the data was recorded every day selecting an interval of "7" would yield an average weekly response.

Author(s)

Joe Swintek

See Also

`autoStdAnylsis` and `forceStdAnalysis`

---

    buildResultsWindow.te   *Display the results from time to event analyses*

Description

This function builds a window that displays the results from a time to event analysis.

Usage

```
buildResultsWindow.te(Results)
```

Arguments

Results   The results output from analyseTime2Effect.

Value

The window displays two objects; a graph which is a kaplan-meier plot generated from the plot.survfit function and Dunnett's table from the glht function in the multcomp package. The Dunnett's table contains the following items:

Comparison   The levels of the treatment variable that are being compared.

Estimate   The estimated difference between the levels.

Std. Error   The standard error.

z value   The z value from the standard normal distribution.

P Value   The p-value corresponding to the z value.

Sig   The significance flag where "." is a p-value > 0.05, "*" is a 0.01 < p-value <= 0.05, "**" for 0.001 < p-value <= 0.01, and "***" for p-value <= 0.001.

The buttons on the window are:

Save Everything
               Creates a folder where the graph is saved as a pdf. In addition, the data set and the Dunnett's table will also be contained in an HTLM file within the folder.

Save Results Saves the data set used and the Dunnett's table as a HTML file.
Save Graphs Only
               Saves the graph as a pdf.

Author(s)

Joe Swintek

---

checkTime   *Checks the format the date is in*

Description

Checks the date to see if it is in the selected format.

Usage

```
checkTime(TimeTemp)
```

Arguments

TimeTemp   A vector of time converted to numerics by date2Numeric.

Value

Returns FALSE if all of the times were not in the correct format. Returns TRUE if **any** of the times where in the correct format.

Author(s)

Joe Swintek

---

cleanString   *Cleans a string of illegal characters*

Description

An internal function that converts any characters in a string that can not be in a windows file name to an underscore ("_").

Usage

```
cleanString(String)
```

Arguments

String   A character string.

Value

String   A character string where all of the illegal characters are converted to "_".

Author(s)

Joe Swintek

---

date2Numeric   *Convert a date into a number*

Description

Converts a vector of dates into a vector of numbers, where the first date is considered to be at time 0.

Usage

```
date2Numeric(DateVec, Format)
```

Arguments

DateVec	A vector of dates.
Format	The R-coded format (as.Date) the dates are in.

Value

DateVec	A numeric vector representing the date.

Note

This always sets the earliest date to 0.

Author(s)

Joe Swintek

---

dunnsTest   *Perform a Dunns test*

Description

This function performs a Dunns (non-parametric dunnett) test for multiple comparisons. The code used in this function is a direct translation of the code in the SAS version of StatCharrms.

Usage

```
dunnsTest(Data, Treatment, Response, TestDirection)
```

Arguments

Data	A data set.
Treatment	The name (as a string) of the treatment variable.
Response	The name (as a string) of the response variable.
TestDirection	The direction the test statistic can be: c('Both', 'Descending', 'Ascending').

Value

Return a data frame containing:

Treatment	The level of the treatment variable.
Count	The Number of observation in that treatment.
Rank	The average rank of the response in the treatment level.
Difference	Difference in rank of the current level to the rank of the controls.
P Value	The p-value of the test statistic.
Signif	The significance flag where"." is a p-value > 0.05, "*" is a 0.01 < p-value <= 0.05, "**" for 0.001 < p-value <= 0.01, and "***" for p-value <= 0.001.

Author(s)

Joe Swintek

References

Dunn, O. J. 1964. Multiple comparisons using rank sums. *Technometrics* **6**:241-252.

Examples

```
#Data
data(lengthWeightData)
#Subset the data
SubData<-lengthWeightData[lengthWeightData$Age=='16 week', ]
SubData<-SubData[SubData$Generation=='F1', ]
SubData<-SubData[SubData$SEX=='M', ]
#Run
dunnsTest(Data=SubData, Treatment='Treatment',
Response='WEIGHT', TestDirection='Both')
```

---

eventTimeData   *Time to event data*

Description

An anonymized data set used in the example analysis for time to event data. The data set is used during the call of generateExamples which is called from the [**Examples**] button in the StatCharrms main window.

Usage

```
data("eventTimeData")
```

Format

A data frame with 240 observations on the following 4 variables.

Time Time as a numeric vector

Status The status variable; 0 for a censored event, 1 for the the measured event.

Trt The treatment level.

Rep The replicate variable.

Examples

```
data(eventTimeData)
str(eventTimeData)
```

---

fecundityData *Example fecundity data*

Description

Example fecundity data taken in daily increments. The data set is used during the call of generateExamples which is called from the [**Examples**] button in the StatCharrms main window.

Usage

```
data("fecundityData")
```

Format

Date The date the observation took place.

Treatment The treatment level.

Rep The replicate the observation belongs to.

Fecundity The number of counted eggs for the pair.

Generation The generation of the fish.

References

TBD

Examples

```
data(eventTimeData)
str(eventTimeData)
```

FindFormat *Find the format the date is written in*

Description

An internal function that finds the R-code formats for dates.

Usage

```
FindFormat(string, CurrentDate)
```

Arguments

string	The current date in a format chosen through the GUI.
CurrentDate	A vector containing the current date in a variety of formats.

Value

string   The R-code for the format of the date.

Author(s)

Joe Swintek

---

forceStdAnalysis   *Run a chosen analysis*

Description

This function runs an analysis chosen through the GUI.

Usage

```
forceStdAnalysis(Data, Response, TreatmentVar, Transform, WeightsVar,
TimeVar, TestDirection, ReplicateVar, Test, AlphaLevel)
```

Arguments

Data	A data set.
Response	The name (as a string) of the response variable.
TreatmentVar	The name (as a string) of the treatment variable.
Transform	The Transformation for the response. Can be any of the following: c('Transformation', 'None', 'Log', log+1', 'Square_Root', 'Arcsin', 'Rank').
WeightsVar	The name (as a string) of the column containing the ANOVA weights.
TimeVar	The name (as a string) of the time variable.
TestDirection	The direction of the hypothesis test. Can take on any of the values of: c('Decreasing', 'Increasing', 'Both').
ReplicateVar	The name (as a string) of the replicate variable.
Test	The test being ran. Any of the following: c('RM ANOVA', 'ME ANOVA', 'Simple ANOVA', 'Weighted ANOVA', 'Jonckheere', 'Dunns', 'Dunnett','Williams') can be used.
AlphaLevel	The alpha level used in the Jonckheere-Terpstra trend test. See jonckheereTerpstraTest for more details.

Value

See the individual functions for information about each test.

Jonckheere	Jonckheere-Terpstra trend test see jonckheereTerpstraTest.
RM ANOVA	Repeated measures ANOVA see runMultiGen.
ME ANOVA	Mixed effects ANOVA see runMultiGen.

Simple ANOVA	One way ANOVA see `basicAnova`.
Weighted ANOVA	A weighted one way ANOVA see `basicAnova`.
Dunns	A Dunns test see `dunnsTest`.
Dunnet	A Dunnett's test see `oneWayDunnettTest`.
WilliamsTable	The results from the Williams test `williamsTest`.

Author(s)

Joe Swintek

---

generateExamples   *Generate examples*

Description

Creates a folder which is populated with examples of appropriate data sets and their corresponding analyses for every type of analysis StatCharrms is capable of producing.

Usage

```
generateExamples()
```

Details

Called by pressing the [**Generate Examples**] button from the main StatCharrms window.

Note

This will create and populate a folder on the computers hard drive.

Author(s)

Joe Swintek

---

getFileName | *Gets a file name*

Description

Calls a dialogue window where the name of a file to be saved can be typed in.

Usage

```
getFileName()
```

Details

This is called immediately before a data structure is saved.

Value

FileName   The name written into the dialogue window.

Author(s)

Joe Swintek

---

getLineContrast | *Generate contrasts*

Description

Generates contrasts for testing monotonicity based on the OECD guidance for statistics.

Usage

```
getLineContrast(Data, Treatment)
getQuadContrast(Data, Treatment)
```

Arguments

Data   A data set.

Treatment   The name (as a string) of the treatment variable.

Value

Returns a numeric vector of contrasts.

Author(s)

Joe Swintek

References

OECD SERIES ON TESTING AND ASSESSMENT Number 54

---

jonckheereTerpstraTest

Perform the Jonckheere-Terpstra trend test

Description

The Jonckheere-Terpstra is a non-parametric test for trend. This function is a wrapper for `jonckheere.test` from the clinfun package.

Usage

```
jonckheereTerpstraTest(Data, Treatment, Response, TestDirection, AlphaLevel)
```

Arguments

Data	A data set.
Treatment	The name (as a string) of the treatment variable.
Response	The name (as a string) of the response variable.
TestDirection	The direction of the test statistic can be any of: c('Both', 'Descending', 'Ascending').
AlphaLevel	The alpha level each step needs to pass before moving on to test a lower treatment level.

Value

JT Statistic	The JT test statistic.
Decreasing Trend P-Value	
	The p-value for a decreasing trend.
D.Sig	The significance flag for the decreasing tend where "." is a p-value > 0.05, "*" is a $0.01 <$ p-value $<= 0.05$, "**" for $0.001 <$ p-value $<= 0.01$, and "***" for p-value $<= 0.001$.
Increasing Trend P-Value	
	The p-value for a increasing trend.
I.Sig	The significance flag for the increasing tend where "." is a p-value > 0.05, "*" is a $0.01 <$ p-value $<= 0.05$, "**" for $0.001 <$ p-value $<= 0.01$, and "***" for p-value $<= 0.001$.
Max Level of Treatment	
	The treatment level the test statistic corresponds to.

Note

A note when using the JT. The JT estimates p-values by a random process thus after an initial run, subsequent runs of the JT will produce slightly different p-values. To produce exactly the same p-values between each run you will need to use the `set.seed` function in the R console before each run.

Author(s)

Joe Swintek

References

Jonckheere, A. R. (1954). A distribution-free k-sample test against ordered alternatives. *Biometrika* **41**:133-145.
Terpstra, T. J. (1952). The asymptotic normality and consistency of Kendall's test against trend, when ties are present in one ranking. emphIndagationes Mathematicae **14**:327-333.

See Also

`jonckheere.test`

Examples

```
## Not run:
#Data
data(lengthWeightData)
#Subset the data
SubData<-lengthWeightData[lengthWeightData$Age=='16 week', ]
SubData<-SubData[SubData$Generation=='F1', ]
SubData<-SubData[SubData$SEX=='M', ]
#Not Run
## Not run:
jonckheereTerpstraTest(Data=SubData, Treatment='Treatment',
Response='WEIGHT', TestDirection='Both', AlphaLevel=0.05)

## End(Not run)
## End(Not run)
```

---

lengthWeightData   *Example length and weight data*

Description

Example length and weight data used in StatCharrms when `generateExamples` is called from the [**Examples**] button in the StatCharrms main window.

Usage

```
data("lengthWeightData")
```

Format

Generation Generation of the fish.

Treatment Treatment level for an observation.

Replicate Replicate the observation belongs to.

Number The row number in the dataset.

SEX The gender of the fish.

WEIGHT The weight of the fish.

LENGTH The length of the fish.

Concentration The concentration of the chemical the fish were exposed to.

Age The age of the fish.

References

TBA

Examples

```
data(lengthWeightData)
str(lengthWeightData)
```

---

leveneTestSC   *Levene test for homogeneity of variance*

Description

Test residuals from an analysis for homogeneity of variance. This function is a wrapper function for leveneTest from the car package.

Usage

```
leveneTestSC(Data, Treatment, Residuals)
```

Arguments

Data	A data set.
Treatment	The name (as a string) of the treatment variable.
Residuals	Residuals from a model fit.

Value

Treatment	The name of the treatment variable.
Df	The degrees of freedom.
F value	The value of the F-statistic.
Pr(>F)	The p-value corresponding to the F-statistic.

Author(s)

Joe Swintek

References

Levene, Howard (1960) Robust tests for equality of variances In Ingram Olkin, Harold Hotelling, et al. Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling. Stanford University Press. pp. 278-292.

See Also

leveneTest

Examples

```
#Data
data(lengthWeightData)
#Subset the data
SubData<-lengthWeightData[lengthWeightData$Age=='16 week', ]
SubData<-SubData[SubData$Generation=='F1', ]
SubData<-SubData[SubData$SEX=='M', ]
#Run
Residuals<-aov(WEIGHT~Treatment,SubData)$residuals
wilksTest(Residuals)
```

---

makeSummaryTable	*Generate a summary table of a data set.*

Description

Generates a table of standard summary statistical values for each treatment level of a response from a data set. This function will always be called when standard numerical endpoints are analysed.

Usage

```
makeSummaryTable(Data, Treatment, Response,
 alpha = 0.05, Replicate = NULL)
```

Arguments

Data	A data set.
Treatment	The name (as a string) of the treatment variable.
Response	The name (as a string) of the endpoint.
alpha	The alpha level used to calculate confidence intervals.
Replicate	The name (as a string) of the replicate variable.

Value

A data set containing the mean, median, standard deviation, standard error, and confidence intervals for each treatment level of the response. If a replicate variable is supplied, the output will also contain the mean of the replicate means and the number of replicates.

Author(s)

Joe Swintek

Examples

```
#Data
data(lengthWeightData)
#Subset the data
SubData<-lengthWeightData[lengthWeightData$Age=='16 week', ]
SubData<-SubData[SubData$Generation=='F1', ]
SubData<-SubData[SubData$SEX=='M', ]
#Run
makeSummaryTable(Data=SubData, Treatment='Treatment',
Response='WEIGHT')
```

---

medianData *Calculate the data median*

Description

Calculates the mean or median value of a response within a replicate for every replicate within a treatment.

Usage

```
medianData(Data, Treatment, Response, Replicate)
averageData(Data, Treatment, Response, Replicate)
```

Arguments

Data	A data set.
Treatment	The name (as a string) of the treatment variable.
Response	The name (as a string) of the response variable.
Replicate	The name (as a string) of the replicate variable.

Value

Returns a data set containing mean or median values for every replicate and treatment.

Author(s)

Joe Swintek

Examples

```
#Data
data(lengthWeightData)
#Subset the data
SubData<-lengthWeightData[lengthWeightData$Age=='8 week', ]
SubData<-SubData[SubData$Generation=='F1', ]
SubData<-SubData[SubData$SEX=='M', ]
#Run
medianData(Data=SubData, Treatment='Treatment', Response='WEIGHT',
Replicate='Replicate')
averageData(Data=SubData, Treatment='Treatment', Response='WEIGHT',
Replicate='Replicate')
```

---

monotonicityTest   *A test for a monotonic trend*

Description

Performs the test for monotonicity as per the the OECD guidance on statistics.

Usage

```
monotonicityTest(Data, Treatment, Response)
```

Arguments

Data	A data set.
Treatment	The name (as a string) of the treatment variable.
Response	The name (as a string) of the response variable.

Details

calls both [getLineContrast](#) and [getQuadContrast](#) to attain the contrasts used in the monotonicity test.

Value

A table of test statistics for both the linear and quadratic trends. If the quadratic trend is significant and the linear trend is not, then the response is **not** monotonic, otherwise it is consider to be monotonic.

Author(s)

Joe Swintek

References

Current Approaches in the Statistical Analysis of Ecotoxicity Data A guidance to application DOI: 10.1787/9789264085275-en

Examples

```
#Data
data(lengthWeightData)
#Subset the data
SubData<-lengthWeightData[lengthWeightData$Age=='8 week', ]
SubData<-SubData[SubData$Generation=='F1', ]
SubData<-SubData[SubData$SEX=='M', ]
#Run
monotonicityTest(Data=SubData, Treatment='Treatment',
Response='WEIGHT')
```

---

oneWayDunnettTest   *One way Dunnett's test.*

Description

Performs the Dunnett's test for multiple comparisons. This is a wrapper function for glht from the multcomp package.

Usage

```
oneWayDunnettTest(Data, Treatment, Response,
WeightList = NULL, TestDirection = "Decreasing"
,alpha = 0.05)
```

Arguments

Data	A data set.
Treatment	The name (as a string) of the treatment variable.
Response	The name (as a string) of the response variable.
WeightList	A list of weights for a weighted ANOVA.
TestDirection	The direction the test statistic can be: c('Both', 'Decreasing', 'Increasing').
alpha	The alpha level used to calculate confidence intervals.

Value

Return a data frame containing:

Treatment	The name of the treatment variable.
Levels	The levels of the treatment variable that are being compared.
Estimate	The estimated difference between the levels.
Std. Error	The standard error.
Df	The degrees of freedom.

Upper CI	The upper 1-alpha adjusted confidence interval of the estimated difference between the levels. See `confint.glht` for more information.
Lower CI	The lower 1-alpha adjusted confidence interval of the estimated difference between the levels. See `confint.glht` for more information.
t.value	The value of the test statistic.
p.value	The p-value of the test statistic.
Signif	The significance flag where "." is a p-value > 0.05, "*" is a 0.01 < p-value <= 0.05, "**" for 0.001 < p-value <= 0.01, and "***" for p-value <= 0.001.

Author(s)

Joe Swintek

References

Dunnett C. W. (1955) A multiple comparison procedure for comparing several treatments with a control, *Journal of the American Statistical Association*, **50**:1096-1121.

Examples

```
#Data
data(lengthWeightData)
#Subset the data
SubData<-lengthWeightData[lengthWeightData$Age=='8 week', ]
SubData<-SubData[SubData$Generation=='F1', ]
SubData<-SubData[SubData$SEX=='M', ]
#Run
oneWayDunnettTest(Data=SubData, Treatment='Treatment', Response='WEIGHT',
WeightList = NULL, TestDirection = "Decreasing")
```

---

rankTransform	*Rank transforms a response*

Description

Transforms a response by rank into critical values of the standard normal distribution. In the case of ties this function will use the mean of the transformed response.

Usage

```
rankTransform(Data, VecName)
```

Arguments

Data	A data set.
VecName	The name (as a string) of the response to be transformed.

Value

Data   The original data set with a new variable called 'TransformedResponse' which
                    is the rank transform of the response.

Author(s)

Joe Swintek

Examples

```
#Data
data(lengthWeightData)
#Subset the data
SubData<-lengthWeightData[lengthWeightData$Age=='16 week', ]
SubData<-SubData[SubData$Generation=='F1', ]
SubData<-SubData[SubData$SEX=='M', ]
#Run
RankData<-rankTransform(Data=SubData, VecName='WEIGHT')
head(RankData)
```

ReadMeFile   *Calls the read me file*

Description

Call a file containing the authors, citation, and changes of StatCharrms.

Usage

```
ReadMeFile()
```

Value

Changes button   Calls ShowChangeLog which displays the changes between versions.

References button

                    Calls Reference which displays the references for each package StatCharrms
                    uses.

Author(s)

Joe Swintek

Reference   *Display the references for StatCharrms*

Description

This is used to display the citations for all the packages StatCharrms uses.

Usage

```
Reference()
```

Details

Called by the [**References**] button in the [ReadMeFile](#) window.

Author(s)

Joe Swintek

---

responseTransform   *Transforms a response*

Description

Transforms a response based on a selection from the GUI input.

Usage

```
responseTransform(Data, ResponVar, Trans)
```

Arguments

Data	A data set.
ResponVar	The name (as a string) of the response variable.
Trans	The name (as a string) of the response variable. Can take on the values of c('Transformation', 'None', 'Log','Log1', 'Square_Root', 'Arcsin', 'Rank').

Value

OutData	The original data set with a new variable called 'TransformedResponse' which is the rank transform of the response.

Note

The log transformation is the common log (base 10) transformation, while 'Arcsin' is the arcsin of the square root of the response. The 'Rank' transformation uses [rankTransform](#).

Author(s)

Joe Swintek

Examples

```
#Data
data(lengthWeightData)
#Subset the data
SubData<-lengthWeightData[lengthWeightData$Age=='16 week', ]
SubData<-SubData[SubData$Generation=='F1', ]
SubData<-SubData[SubData$SEX=='M', ]
#Run
RankData<-responseTransform(Data=SubData, ResponVar='WEIGHT', Trans='Log')
head(RankData)
```

Run.StatCharrms   *Main function call for StatCharrms*

Description

Call this function to start StatCharrms.

Usage

```
Run.StatCharrms()
```

Details

This is the GUI front end for (StatCharrms).

Value

Read Me Button   Calls [ReadMeFile](#) which brings up the read me file.

Examples Button

                Calls [generateExamples](#), which populates a folder with example datasets and analyses.

Histology Analysis Button

                Calls [Histopath](#) which will run RSCABS.

Time to Event Analysis Button

                Calls [Time2EventMain](#) which allows for a time to event analysis.

Analysis of Other End Points Button

                Calls [StdEndMain](#) which allows for statistical tests on numerical endpoints.

Quit Button   Closes StatCharrms.

Author(s)

Joe Swintek

References

use [Reference](#) to see all the references.

---

runMultiGen   *Runs a mixed effects ANOVA*

Description

This function runs a mixed effect ANOVA.

Usage

```
runMultiGen(Data, TreatVar = ", ResponVar = ",
RepVar = "Not Used", TimeVar = "Not Used",
Path, TestDirection = "Descending",alpha=0.05)
```

Arguments

Data	A data set.
TreatVar	Name (as a string) of the treatment variable.
ResponVar	Name (as a string) of the response variable.
RepVar	Name (as a string) of the replicate variable.
TimeVar	Name (as a string) of the time variable.
Path	Path indicates the type of mixed effects analysis performed. Use Path = 2 for Time (Repeated Measures) and Path = 3 Group - Replicate (Mixed Effects).
TestDirection	Indicates the test direction. This can be c('Both', 'Descending', 'Ascending').
alpha	Alpha level used in calculating confidence intervals.

Details

This is mostly a wrapper function for [nlme](#) from the nlme package and [glht](#) from the multcomp package.

Value

Returns a list with the following items:

Anova.Table	An ANOVA table see [Anova](#).
MainEffects	A table of main effect, see [oneWayDunnettTest](#) for more details.
FreqTable	A table of frequencies of occurrences for every sub group in the data set.
Lmm	A nlme class data structure from [<nlme>](#) from the nlme package.
ShapiroTest	The results from a Shapiro-Wilks test. See [wilksTest](#) for more information.

Author(s)

Joe Swintek

References

Pinheiro, J.C., and Bates, D.M. (2000) Mixed-Effects Models in S and S-PLUS, Springer

See Also

[nlme](), [glht]()

Examples

```
#Data
data(lengthWeightData)
#Subset the data
SubData<-lengthWeightData[lengthWeightData$Age=='8 week', ]
SubData<-SubData[SubData$Generation=='F1', ]
SubData<-SubData[SubData$SEX=='M', ]
#Run

runMultiGen(Data=SubData, TreatVar = "Treatment",
ResponVar = "WEIGHT", RepVar = 'Replicate', TimeVar = "Not Used",
Path=3, TestDirection = "Descending")
```

ShowChangeLog   *Display the change logs for StatCharms*

Description

Displays the changes that have occurred between the versions of StatCharrms. Called from the [**changes**] button from the "Read me" window.

Usage

```
ShowChangeLog()
```

Author(s)

Joe Swintek

See Also

Also see [ReadMeFile]()

StdEndMain *GUI for Analysis of numerical endpoints*

Description

Main function call used to produce the GUI for analysis of standard numerical endpoints. Example numerical endpoints are length, weight, and fecundity.

Usage

```
StdEndMain()
```

Details

This is set up to follow the protocols for data analysis outlined in MEOGRT.

Value

Load Data Button

Load a data set from an csv file.

Specify Data Button

Calls addMultiSpec.std. This will produce a "Data specification" tab where the data can be specified.

Run Analysis Button

Calls autoStdAnylsis or forceStdAnalysis. Both functions will perform the analyses specified in the "Data specification" tab. After the analyses are ran the buildResultsWindow function will be called which will bring up a new window containing the results.

Author(s)

Joe Swintek

References

OECD, 2015, Test No. 240: Medaka Extended One Generation Reproduction Test (MEOGRT), OECD Publishing, Paris. DOI: http://dx.doi.org/10.1787/9789264242258-en

---

Time2EventMain   *GUI for time to event analysis*

Description

Main function call that produces the GUI for a time to event analysis.

Usage

```
Time2EventMain()
```

Details

Used as the GUI front-end to perform the time to event analysis as outline in the LAGDA guidelines.

Value

Load Data Button

Load a data set from an csv file.

Specify Data Button

Calls addSpec.te. This will produce the widgets needed to specify the data.

Run Analysis Button

Calls analyseTime2Effect and performs the time to event analysis. After which call the buildResultsWindow.te function which displays the results from the analysis.

Author(s)

Joe Swintek

References

OECD, 2015, OECD Guidelines for the Testing of Chemicals, Section 2. Test No. 241: The Larval Amphibian Growth and Development Assay (LAGDA) OECD Publishing, Paris. DOI:http://dx.doi.org/10.1787/9789264242 en

---

tranformTime   *Transforms the time variable*

Description

Transforms a variable that contains information about when an observation was taken into a numeric vector, then averages the response variable across a specified interval of time.

Usage

```
tranformTime(Data, TimeVar, Time, RepVar, GroupVar, ResponVar)
```

Arguments

`Data`	A data set.
`TimeVar`	The name (as a string) of the time variable.
`Time`	The interval time the response is averaged across.
`RepVar`	The name (as a string) of the replicate variable.
`GroupVar`	The name (as a string) of the treatment variable.
`ResponVar`	The name (as a string) of response variable.

Details

This function just averages across time.

Value

Returns a data set where the time variable has been changed to a numeric time and the response variable has been averaged across time.

Author(s)

Joe Swintek

---

transformationWarning   *Gives warnings about a transformation*

Description

An internal function that warns the user when a transformation can not be done on a set of numbers.

Usage

```
transformationWarning(Data, Trans, VecName)
```

Arguments

`Data`	A data set.
`Trans`	The transformation being tested. Gives warning for the values of: c('Log', Log1', Square_Root', 'Arcsin').
`VecName`	The name of the variable to be transformed.

Details

This function is called by responseTransform.

Value

Will display a warning message to the user, when appropriate.

Author(s)

Joe Swintek

---

wilksTest   *Performs the Shapiro-Wilks test.*

Description

Performs the Shapiro-Wilks test for normality on a set of residuals from an analysis. This is a wrapper function for the `shapiro.test` function with additional evaluation statistics.

Usage

```
wilksTest(Residuals)
```

Arguments

Residuals   Residuals from an analysis.

Value

WilksTest, a data frame containing:

OBS	Total number of observation.
STD	The standard error.
SKEW	The skew of the data set.
KURT	The measure of kurtosis (how heavy tailed the distribution is) of the data set.
SW_STAT	The Shapiro-Wilks test statistic.
P_VALUE	The p-Value for the test statistic.
Signif	The flag for p-values less then 0.01.

Author(s)

Joe Swintek

References

Patrick Royston (1982) An extension of Shapiro and Wilk's *W* test for normality to large samples. *Applied Statistics*, **31**: 115-124.

Patrick Royston (1982) Algorithm AS 181: The *W* test for Normality. *Applied Statistics*, **31**: 176-180.

Patrick Royston (1995) Remark AS R94: A remark on Algorithm AS 181: The *W* test for normality. *Applied Statistics*, **44**: 547-551.

Johnson, NL, Kotz, S, Balakrishnan N (1994) *Continuous Univariate Distributions, Vol 1, 2nd Edition* Wiley ISBN 0-471-58495-9.

Examples

```
#Data
data(lengthWeightData)
#Subset the data
SubData<-lengthWeightData[lengthWeightData$Age=='16 week', ]
SubData<-SubData[SubData$Generation=='F1', ]
SubData<-SubData[SubData$SEX=='M', ]
#Run
Residuals<-aov(WEIGHT~Treatment,SubData)$residuals
wilksTest(Residuals)
```

---

williamsTest   *Performs Williams Test*

Description

The Williams test is a parametric test for trend. It is used to test for a trend when normality assumption is met.

Usage

```
williamsTest(df,resp,trt,direction='decreasing',SeIn=NULL)
```

Arguments

df	A data set.
resp	The name (as a string) of the response variable.
trt	The name (as a string) of the treatment variable.
direction	The direction of the test statistic which can either be 'increasing', 'decreasing').
SeIn	This is used to specify a different standard error then the one calculated within the WilliamsTest function.

Value

trt	The treatment level the test statistic corresponds to.
Y.Tilde	The amalgamated averages for the treatment level.
Se Diff	The standard error.
DF	The degrees of freedom.
Will	The value of the Williams test statistic.
TCrit	The critical value of the Williams test statistic, corresponding to a p-value of 0.05.

Author(s)

Joe Swintek

References

Williams D.A. (1971). A test for differences between treatment means when several dose levels are compared with a zero dose control. *Biometrics* **27**(1):103-117.
Green J., Springer T., Holbeck H. *Statistical Analysis of Ecotoxicology Data* (Wiley in press)

Examples

```
#Data
data(lengthWeightData)
#Subset the data
SubData<-lengthWeightData[lengthWeightData$Age=='16 week', ]
SubData<-SubData[SubData$Generation=='F1', ]
SubData<-SubData[SubData$SEX=='M', ]
williamsTest(df=SubData, trt='Treatment',
resp='WEIGHT', direction='decreasing',SeIn=NULL)
```

---

williamsTestLookUpTable

Look up table for the critical values used in the Williams test.

Description

This is the look up table for the critical values provided in Green (2018). This table is automaticly called durring the the exicution of [williamsTest](#) function.

Usage

```
data("williamsTestLookUpTable")
```

Format

df The degree of freedom for the test.

Q

B

References

Green J., Springer T., Holbeck H. *Statistical Analysis of Ecotoxicology Data* (Wiley in press)

Examples

```
data(williamsTestLookUpTable)
str(williamsTestLookUpTable)
```

Index