Package: statcheck (via r-universe)

August 25, 2024

Title Extract Statistics from Articles and Recompute P-Values

Version 1.6.0

Date 2024-07-26

Description A ``spellchecker'' for statistics. It checks whether your p-values match their accompanying test statistic and degrees of freedom. statcheck searches for null-hypothesis significance test (NHST) in APA style (e.g., t(28) = 2.2, p < .05). It recalculates the p-value using the reported test statistic and degrees of freedom. If the reported and computed p-values don't match, statcheck will flag the result as an error. If the reported p-value is statistically significant and the recomputed one is not, or vice versa, the result will be flagged as a decision error. You can use statcheck directly on a string of text, but you can also scan a PDF or HTML file, or even a folder of PDF and/or HTML files. Statcheck needs an external program to convert PDF to text: Xpdf. Instructions on where and how to download this program, how to install statcheck, and more details on what statcheck can and cannot do can be found in the online manual: <`https://rpubs.com/michelenuijten/statcheckmanual`>. You can find a point-and-click web interface to scan PDF or HTML or DOCX articles on <`http://statcheck.io`>.

License GPL-3

Depends R (>= 2.14.2)

Imports ggplot2, plyr, rlang, rmarkdown, stringi, tcltk, stringr, pdftools

Suggests testthat

ByteCompile yes

Encoding UTF-8

LazyLoad yes

RoxygenNote 7.3.1

URL `https://github.com/MicheleNuijten/statcheck`

BugReports `https://github.com/MicheleNuijten/statcheck/issues`

Repository https://michelenuijten.r-universe.dev

RemoteUrl https://github.com/michelenuijten/statcheck

RemoteRef HEAD

RemoteSha b288d9488c6019d748c4af1a08e6f1188a482d92

Contents

checkHTML 2
checkHTMLdir 3
identify.statcheck 3
plot.statcheck 4
statcheck 5
statcheckReport 8
summary.statcheck 9
trim 10

Index **11**

`checkHTML` *Extract statistics from PDF/HTML articles and recalculate p-values*

Description

These functions search for NHST results in PDF and/or HTML articles and send the extracted statistics to `statcheck`.

Usage

```
checkHTML(files, ...)

checkPDF(files, ...)
```

Arguments

`files`	Vector of strings containing file paths to HTML files to check.
`...`	Arguments sent to `statcheck`.

Value

A statcheck data frame with the extracted statistics. See `statcheck` for details.

`checkHTMLdir` *Extract statistics from folders with PDF/HTML articles and recalculate p-values*

Description

These functions search for NHST results in all PDF and/or HTML articles in a certain folder and send the extracted statistics to `statcheck`.

Usage

```
checkHTMLdir(dir, subdir = TRUE, extension = TRUE, ...)

checkPDFdir(dir, subdir = TRUE, ...)

checkdir(dir, subdir = TRUE, ...)
```

Arguments

`dir`	String indicating the directory to be used. If this is left empty, a window will pop up from which you can choose a directory.
`subdir`	Logical. Indicates whether you also want to check subfolders. Defaults to TRUE
`extension`	Logical. Indicates whether the HTML extension should be checked. Defaults to TRUE
`...`	Arguments sent to `statcheck`.

Value

A statcheck data frame with the extracted statistics. See `statcheck` for details.

`identify.statcheck` *Identify specific points in a statcheck plot.*

Description

With this function you can simply point and click on the datapoints in the plot to see the corresponding statcheck details, such as the paper from which the data came and the exact statistical results.

Usage

```
## S3 method for class 'statcheck'
identify(x, alpha = 0.05, ...)
```

Arguments

`x`	A statcheck object. See `statcheck`.
`alpha`	assumed level of significance in the scanned texts. Defaults to .05.
`...`	arguments to be passed to methods, such as graphical parameters (see `par`).

Examples

```
## Not run:

# First we need a statcheck object
# Here, we create one by running statcheck on some raw text

txt <- "This test is consistent t(28) = 0.2, p = .84, but this one is
inconsistent: F(2, 28) = 4.2, p = .01. This final test is even a
gross/decision inconsistency: z = 1.23, p = .03"

result <- statcheck(txt)

# Now, we can run identify.statcheck(), or shorter, simply identify():
identify(result)

# Further instructions:
# click on one or multiple points of interest
# press Esc
# a dataframe with information on the selected points will appear


## End(Not run)
```

`plot.statcheck` *Plot method for statcheck*

Description

Function for plotting of `statcheck` objects. Reported p values are plotted against recalculated p values, which allows the user to easily spot if articles contain miscalculations of statistical results.

Usage

```
## S3 method for class 'statcheck'
plot(x, alpha = 0.05, APAstyle = TRUE, group = NULL, ...)
```

Arguments

`x`	A statcheck object. See `statcheck`.
`alpha`	assumed level of significance in the scanned texts. Defaults to .05.

`APAstyle`	If TRUE, prints plot in APA style.
`group`	Indicate grouping variable to facet plot. Only works when `APAstyle==TRUE`
`...`	arguments to be passed to methods, such as graphical parameters (see `par`).

Details

If APAstyle = FALSE, inconsistencies between the reported and the recalculated p value are indicated with an orange dot. Recalculations of the p value that render a previously non significant result (p >= .5) as significant (p < .05), and vice versa, are considered decision errors, and are indicated with a red dot. Exactly reported p values (i.e. p = ..., as opposed to p < ... or p > ...) are indicated with a diamond.

Acknowledgements

Many thanks to John Sakaluk who adapted the plot code to create graphs in APA style.

See Also

`statcheck`

Examples

```
# First we need a statcheck object
# Here, we create one by running statcheck on some raw text

txt <- "This test is consistent t(28) = 0.2, p = .84, but this one is
inconsistent: F(2, 28) = 4.2, p = .01. This final test is even a
gross/decision inconsistency: z = 1.23, p = .03"

result <- statcheck(txt)

# We can then plot the statcheck object 'result' by simply calling plot() on
# "result". R will know what kind of plot to make, because "result" is of
# class "statcheck"
plot(result)
```

`statcheck` *Extract statistics and recompute p-values*

Description

`statcheck` extracts Null Hypothesis Significance (NHST) results from strings and returns the extracted values, reported p-values and recomputed p-values.

Usage

```
statcheck(
  texts,
  stat = c("t", "F", "cor", "chisq", "Z", "Q"),
  OneTailedTests = FALSE,
  alpha = 0.05,
  pEqualAlphaSig = TRUE,
  pZeroError = TRUE,
  OneTailedTxt = FALSE,
  AllPValues = FALSE,
  messages = TRUE
)
```

Arguments

`texts`	A vector of strings.
`stat`	Specify which test types you want to extract. "t" to extract t-values, "F" to extract F-values, "cor" to extract correlations, "chisq"to extract $\chi 2$ values, "Z" to extract Z-values, and "Q" to extract Q-values. Using `c()` you can specify multiple tests. Defaults to all tests.
`OneTailedTests`	Logical. Do you want to assume that all reported tests are one-tailed (TRUE) or two-tailed (FALSE, default)?
`alpha`	Assumed level of significance in the scanned texts. Defaults to .05.
`pEqualAlphaSig`	Logical. If TRUE, statcheck counts p <= alpha as significant (default), if FALSE, statcheck counts p < alpha as significant.
`pZeroError`	Logical. If TRUE, statcheck counts p = .000 as an error (because a p-value is never exactly zero, and should be reported as < .001), if FALSE, statcheck does not count p = .000 automatically as an error.
`OneTailedTxt`	Logical. If TRUE, statcheck searches the text for "one-sided", "one-tailed", and "directional" to identify the possible use of one-sided tests. If one or more of these strings is found in the text AND the result would have been correct if it was a one-sided test, the result is assumed to be indeed one-sided and is counted as correct.
`AllPValues`	Logical. If TRUE, the output will consist of a dataframe with all detected p values, also the ones that were not part of the full results in APA format.
`messages`	Logical. If TRUE, statcheck will print a progress bar while it's extracting statistics from text.

Details

`statcheck` roughly works in three steps.

1. Scan text for statistical results

`statcheck` uses regular expressions to recognizes statistical results from t-tests, F-tests, $\chi 2$-tests, Z-tests, Q-tests, and correlations. statcheck can only recognize these results if the results are reported exactly according to the APA guidelines:

- t(df) = value, p = value
- F(df1, df2) = value, p = value
- r(df) = value, p = value
- $\chi 2$ (df, N = value) = value, p = value (N is optional)
- Z = value, p = value
- Q(df) = value, p = value (statcheck can distinguish between Q, Qw / Q-within, and Qb / Q-between)

`statcheck` takes into account that test statistics and p values may be exactly (=) or inexactly (< or >) reported. Different spacing has also been taken into account.

2. Recompute p-value

`statcheck` uses the reported test statistic and degrees of freedom to recompute the p-value. By default, the recomputed p-value is two-sided

3. Compare reported and recomputed p-value

This comparison takes into account how the results were reported, e.g., p < .05 is treated differently than p = .05. Incongruent p values are marked as an `error`. If the reported result is significant and the recomputed result is not, or vice versa, the result is marked as a `decision_error`.

Correct rounding is taken into account. For instance, a reported t-value of 2.35 could correspond to an actual value of 2.345 to 2.354 with a range of p-values that can slightly deviate from the recomputed p-value. `statcheck` will not count cases like this as errors.

Note that when `statcheck` flags an `error` or `decision_error`, it implicitly assumes that the p-value is the inconsistent value, but it could just as well be the case that the test statistic or degrees of freedom contain a reporting error. `statcheck` merely detects wether a set of numbers is consistent with each other.

Value

A data frame containing for each extracted statistic:

source	Name of the file of which the statistic is extracted
test_type	Character indicating the statistic that is extracted
df1	First degree of freedom (if applicable)
df2	Second degree of freedom
test_comp	Reported comparison of the test statistic, when importing from pdf this will often not be converted properly
test_value	Reported value of the statistic
p_comp	Reported comparison, when importing from pdf this might not be converted properly
reported_p	The reported p-value, or NA if the reported value was n.s.
computed_p	The recomputed p-value
raw	Raw string of the statistical reference that is extracted
error	The computed p value is not congruent with the reported p-value
decision_error	The reported result is significant whereas the recomputed result is not, or vice versa.
one_tailed_in_txt	Logical. Does the text contain the string "sided", "tailed", and/or "directional"?
apa_factor	What proportion of all detected p-values was part of a fully APA reported result?

See Also

For more details, see the online manual.

Examples

```
txt <- "blablabla the effect was very significant (t(100)=1, p < 0.001)"
statcheck(txt)
```

`statcheckReport` *Generate HTML report for statcheck output*

Description

This function uses R Markdown to generate a nicely formatted HTML report of `statcheck` output.

Usage

```
statcheckReport(statcheckOutput, outputFileName, outputDir)
```

Arguments

`statcheckOutput`	statcheck output of one of the following functions: `statcheck`, `checkPDFdir`, `checkPDF`, `checkHTMLdir`, `checkHTML`, or `checkdir`.
`outputFileName`	String specifying the file name under which you want to save the generated HTML report. The extension ".html" is automatically added, so doesn't need to be specified in this argument.
`outputDir`	String specifying the directory in which you want to save the generated HTML report.

Details

This function temporarily saves the inserted `statcheck` output as an .RData file in the "output" folder in the statcheck package directory. This file is then called by the .Rmd template that is saved in the folder "rmd", also in the statcheck package directory. After the HTML report is generated, the .RData file is removed again.

Value

An HTML report, saved in the directory specified in the argument "outputDir".

Examples

```
## Not run:

# first generate statcheck output, for instance by using the statcheck()
function

txt <- "blablabla the effect was very significant (t(100)=1, p < 0.001)"
stat <- statcheck(txt)

# next, use this output to generate a nice HTML report of the results
statcheckReport(stat, outputFileName="statcheckHTMLReport",
                outputDir="C:/mydocuments/results")

# you can now find your HTML report in the folder
# "C:/mydocuments/results" under the name "statcheckHTMLReport.html".


## End(Not run)
```

`summary.statcheck` *Summary method for statcheck*

Description

Gives the summaries for a `statcheck` object.

Usage

```
## S3 method for class 'statcheck'
summary(object, ...)
```

Arguments

`object`	a `statcheck` object.
`...`	additional arguments affecting the summary produced.

Value

A data frame containing for each source of statistics:

source Name of the file/origin of which the statistics are extracted

nr_p_values The number of extracted reported p values per article

nr_errors The number of errors per article

nr_decision_errors The number of decision errors per article

Examples

```
txt <- "blablabla the effect was very significant (t(100)=1, p < 0.001)"
stat <- statcheck(txt)
summary(stat)
```

`trim` *Trimming method for statcheck output*

Description

Returns a subset of columns of a `statcheck` object.

Usage

```
trim(object, ...)
```

Arguments

`object`	a `statcheck` object.
`...`	additional arguments affecting the trimmed output.

Value

A data frame containing for each source of statistics:

source Name of the file/origin of which the statistics are extracted

raw Raw string of the statistical reference that is extracted

computed_p The recomputed p-value

error The computed p value is not congruent with the reported p-value

decision_error The reported result is significant whereas the recomputed result is not, or vice versa.

Examples

```
txt <- "blablabla the effect was very significant (t(100)=1, p < 0.001)"
stat <- statcheck(txt)
trim(stat)
```

Index

`checkdir`, *8*
`checkdir` *(*`checkHTMLdir`*)*, 3
`checkdirs` *(*`checkHTMLdir`*)*, 3
`checkfiles` *(*`checkHTML`*)*, 2
`checkHTML`, 2, *8*
`checkHTMLdir`, 3, *8*
`checkPDF`, *8*
`checkPDF` *(*`checkHTML`*)*, 2
`checkPDFdir`, *8*
`checkPDFdir` *(*`checkHTMLdir`*)*, 3

`identify.statcheck`, 3

`par`, *4, 5*
`plot.statcheck`, 4

`statcheck`, *2–5*, 5, *8*
`statcheckReport`, 8
`summary.statcheck`, 9

`trim`, 10