

Package ‘effectcheck’

May 8, 2026

Type Package

Title Statistical Consistency Checker for Published Research Results

Version 0.2.3

Description A conservative, assumption-aware statistical consistency checker for published research results. Parses test statistics, effect sizes, and confidence intervals from text, PDF, HTML, and Word documents across multiple citation styles including American Psychological Association (APA), Harvard, Frontiers, PLOS ONE, Scientific Reports, Nature Human Behaviour, PeerJ, eLife, PNAS, and others. Recomputes effect sizes using all plausible variants when design is ambiguous, and validates internal consistency. Supports t-tests, F-tests/ANOVA, correlations, chi-square, z-tests, regression, and nonparametric tests. Provides 'statcheck'-compatible API functions for batch processing of files and directories. Explicitly tracks all assumptions and uncertainty in output. Detects decision errors (significance reversals) similar to 'statcheck'. Note: this package is under active development and results should be independently verified. Use is at the sole responsibility of the user. Contributions and verification reports are welcome.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

URL <https://github.com/giladfeldman/escicheck>

BugReports <https://github.com/giladfeldman/escicheck/issues>

SystemRequirements poppler-utils (pdftotext) for PDF text extraction

Imports stringr, stringi, dplyr, purrr, tibble, xml2, rvest, glue,
logger, graphics, stats, utils

Suggests shiny, shinythemes, DT, knitr, rmarkdown, testthat (>=
3.0.0), MBESS, effectsize, jsonlite, tesseract, magick, qpdf,
statcheck

Depends R (>= 4.1.0)

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Gilad Feldman [aut, cre] (ORCID:
<https://orcid.org/0000-0003-2812-6599>)

Maintainer Gilad Feldman <giladfel@gmail.com>

Repository CRAN

Date/Publication 2026-03-25 10:50:02 UTC

Contents

.effectcheck_version	3
checkDOCXdir	3
checkHTML	4
checkHTMLdir	4
checkPDF	5
checkPDFdir	6
check_dir	7
check_file	8
check_files	9
check_text	9
compare_file_with_statcheck	11
compare_to_variants	12
compare_with_statcheck	12
count_by	13
ec_identify	13
export_csv	14
export_json	15
extract_pdf_comprehensive	15
filter_by_delta	16
filter_by_source	17
filter_by_test_type	17
filter_by_uncertainty	18
format_variants	18
generate_report	19
get_alternatives	20
get_decision_errors	20
get_effect_family	21
get_errors	21
get_same_type_variants	22
get_variants	22
get_variant_metadata	23
get_warnings	24
is.effectcheck	24
parse_text	25
plot.effectcheck	25
print.effectcheck	26
print.effectcheck_comparison	27
print.summary.effectcheck	27
rbind.effectcheck	28

`.effectcheck_version` 3

<code>read_any_text</code>	29
<code>render_report</code>	29
<code>summary.effectcheck</code>	30
<code>[.effectcheck</code>	31

Index 32

<code>.effectcheck_version</code>	<i>EffectCheck S3 Class Definition and Methods</i>
-----------------------------------	--

Description

This file defines the `effectcheck` S3 class and its associated methods for printing, summarizing, and plotting results.

Usage

```
.effectcheck_version()
```

<code>checkDOCXdir</code>	<i>Check Word documents in a directory</i>
---------------------------	--

Description

Scans a directory for `.docx` files and checks all detected statistics.

Usage

```
checkDOCXdir(dir, subdir = TRUE, messages = TRUE, ...)
```

Arguments

- | | |
|-----------------------|--|
| <code>dir</code> | Directory path to scan |
| <code>subdir</code> | Logical, recurse into subdirectories (default TRUE) |
| <code>messages</code> | Logical, show progress messages (default TRUE) |
| <code>...</code> | Additional arguments passed to <code>check_text()</code> |

Value

An `effectcheck` object with results

Examples

```
## Not run:  
# Requires a directory with DOCX files  
results <- checkDOCXdir("manuscripts/")  
summary(results)  
  
## End(Not run)
```

checkHTML	<i>Check HTML files for statistical consistency</i>
-----------	---

Description

Checks one or more HTML files for statistical consistency. This function is provided for compatibility with statcheck's checkHTML().

Usage

```
checkHTML(files, messages = TRUE, ...)
```

Arguments

files	Character vector of HTML file paths
messages	Logical, show progress messages (default TRUE)
...	Additional arguments passed to check_text()

Value

An effectcheck object with results

Examples

```
tmp <- tempfile(fileext = ".html")
writeLines("<p>t(28) = 2.21, p = .035, d = 0.80</p>", tmp)
results <- checkHTML(tmp)
summary(results)
unlink(tmp)
```

checkHTMLdir	<i>Check a directory of HTML files</i>
--------------	--

Description

Scans a directory for HTML files and checks all detected statistics. This function is provided for compatibility with statcheck's checkHTMLdir().

Usage

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

Arguments

dir	Directory path to scan
subdir	Logical, recurse into subdirectories (default TRUE)
messages	Logical, show progress messages (default TRUE)
...	Additional arguments passed to check_text()

Value

An effectcheck object with results

Examples

```
## Not run:  
# Requires a directory with HTML files  
results <- checkHTMLdir("manuscripts/")  
summary(results)  
  
## End(Not run)
```

checkPDF

Check PDF files for statistical consistency

Description

Checks one or more PDF files for statistical consistency. This function is provided for compatibility with statcheck's checkPDF().

Usage

```
checkPDF(files, try_tables = TRUE, try_ocr = FALSE, messages = TRUE, ...)
```

Arguments

files	Character vector of PDF file paths
try_tables	Logical, attempt table extraction (default TRUE)
try_ocr	Logical, attempt OCR for scanned PDFs (default FALSE)
messages	Logical, show progress messages (default TRUE)
...	Additional arguments passed to check_text()

Value

An effectcheck object with results

Examples

```
## Not run:
# Requires PDF files
results <- checkPDF(c("paper1.pdf", "paper2.pdf"))
summary(results)

## End(Not run)
```

checkPDFdir	<i>Check a directory of PDF files</i>
-------------	---------------------------------------

Description

Scans a directory for PDF files and checks all detected statistics. This function is provided for compatibility with statcheck's checkPDFdir().

Usage

```
checkPDFdir(
  dir,
  subdir = TRUE,
  try_tables = TRUE,
  try_ocr = FALSE,
  messages = TRUE,
  ...
)
```

Arguments

dir	Directory path to scan
subdir	Logical, recurse into subdirectories (default TRUE)
try_tables	Logical, attempt table extraction (default TRUE)
try_ocr	Logical, attempt OCR for scanned PDFs (default FALSE)
messages	Logical, show progress messages (default TRUE)
...	Additional arguments passed to check_text()

Value

An effectcheck object with results

Examples

```
## Not run:
# Requires a directory with PDF files
results <- checkPDFdir("manuscripts/")
summary(results)

## End(Not run)
```

check_dir	<i>Check a directory for statistical consistency</i>
-----------	--

Description

Recursively scans a directory for supported files and checks all detected statistics for consistency.

Usage

```
check_dir(
  dir,
  subdir = TRUE,
  pattern = "\\.(pdf|html?|docx|txt)$",
  try_tables = TRUE,
  try_ocr = FALSE,
  messages = TRUE,
  allowed_base_dirs = NULL,
  ...
)
```

Arguments

dir	Directory path to scan
subdir	Logical, recurse into subdirectories (default TRUE)
pattern	File pattern regex (default matches .pdf, .html, .htm, .docx, .txt)
try_tables	Logical, attempt table extraction from PDFs (default TRUE)
try_ocr	Logical, attempt OCR for scanned PDFs (default FALSE)
messages	Logical, show progress messages (default TRUE)
allowed_base_dirs	Optional character vector of allowed base directories for security
...	Additional arguments passed to check_text()

Value

An effectcheck object with results from all files

Examples

```
d <- tempdir()
writeLines("t(28) = 2.21, p = .035, d = 0.80", file.path(d, "test.txt"))
results <- check_dir(d, pattern = "\\.(txt)$")
summary(results)
unlink(file.path(d, "test.txt"))
```

check_file	<i>EffectCheck API Functions</i>
------------	----------------------------------

Description

This file provides statcheck-compatible API functions for checking statistical consistency in APA-style results. Check a single file for statistical consistency

Usage

```
check_file(path, try_tables = TRUE, try_ocr = FALSE, ...)
```

Arguments

path	Path to the file (.pdf, .html, .docx, or .txt)
try_tables	Logical, attempt table extraction from PDFs (default TRUE)
try_ocr	Logical, attempt OCR for scanned PDFs (default FALSE)
...	Additional arguments passed to check_text()

Details

Reads a file and checks all detected statistics for consistency.

Value

An effectcheck object with results

Examples

```
tmp <- tempfile(fileext = ".txt")
writeLines("t(28) = 2.21, p = .035, d = 0.80", tmp)
result <- check_file(tmp)
print(result)
unlink(tmp)
```

check_files	<i>Check files for statistical consistency</i>
-------------	--

Description

Reads one or more files and checks all detected statistics for consistency.

Usage

```
check_files(paths, try_tables = TRUE, try_ocr = FALSE, messages = TRUE, ...)
```

Arguments

paths	Character vector of file paths (.pdf, .html, .docx, or .txt)
try_tables	Logical, attempt table extraction from PDFs (default TRUE)
try_ocr	Logical, attempt OCR for scanned PDFs (default FALSE)
messages	Logical, show progress messages (default TRUE)
...	Additional arguments passed to check_text()

Value

An effectcheck S3 object with consistency check results

Examples

```
tmp <- tempfile(fileext = ".txt")
writeLines("t(28) = 2.21, p = .035, d = 0.80", tmp)
results <- check_files(tmp)
print(results)
unlink(tmp)
```

check_text	<i>Check raw text for statistical consistency</i>
------------	---

Description

Parses APA-style statistical results from text and checks for consistency between reported and computed values. Uses type-matched comparison to ensure reported effect sizes are compared against the same type of computed values.

Usage

```

check_text(
  text,
  stats = c("t", "F", "r", "chisq", "z", "U", "W", "H", "regression"),
  ci_level = 0.95,
  alpha = 0.05,
  one_tailed = FALSE,
  paired_r_grid = seq(0.1, 0.9, by = 0.1),
  assume_equal_ns_when_missing = TRUE,
  ci_method_phi = "bonett_price",
  ci_method_V = "bonett_price",
  tol_effect = list(d = 0.02, r = 0.005, phi = 0.02, V = 0.02),
  tol_ci = 0.02,
  tol_p = 0.001,
  messages = FALSE,
  max_text_length = 10^7,
  max_stats_per_text = 10000,
  cross_type_action = "NOTE",
  ci_affects_status = TRUE,
  plausibility_filter = TRUE,
  sign_sensitive = FALSE
)

```

Arguments

text	Character vector of text to check
stats	Character vector of test types to check (default: all supported types)
ci_level	Default confidence interval level (default 0.95)
alpha	Significance threshold for decision error detection (default 0.05)
one_tailed	Logical, assume one-tailed tests (default FALSE)
paired_r_grid	Numeric vector of correlation values for paired t-test grid search
assume_equal_ns_when_missing	Logical, assume equal group sizes when missing (default TRUE)
ci_method_phi	CI method for phi coefficient (default "bonett_price")
ci_method_V	CI method for Cramer's V (default "bonett_price")
tol_effect	List of tolerances for effect sizes by type
tol_ci	Tolerance for CI bounds (default 0.02)
tol_p	Tolerance for p-values (default 0.001)
messages	Logical, show progress messages (default FALSE)
max_text_length	Maximum total text length in characters (default 10 ⁷)
max_stats_per_text	Maximum number of stats to process per text (default 10000)

cross_type_action Action when cross-type match found ("NOTE", "WARN", or "ERROR"; default "NOTE")
ci_affects_status Whether CI mismatches affect status (default TRUE)
plausibility_filter Whether to apply plausibility bounds filter (default TRUE)
sign_sensitive Whether sign differences affect status (default FALSE)

Value

An effectcheck S3 object with consistency check results

Examples

```
result <- check_text("t(28) = 2.21, p = .035, d = 0.80")
print(result)
summary(result)
```

compare_file_with_statcheck

Compare effectcheck and statcheck results on a file

Description

Compare effectcheck and statcheck results on a file

Usage

```
compare_file_with_statcheck(path, ...)
```

Arguments

path Path to a file (PDF, HTML, DOCX, or text)
... Additional arguments passed to check_file()

Value

A tibble with source column

Examples

```
tmp <- tempfile(fileext = ".txt")
writeLines("t(28) = 2.21, p = .035, d = 0.80", tmp)
comp <- compare_file_with_statcheck(tmp)
unlink(tmp)
```

compare_to_variants *Compare reported value to all variants*

Description

Creates a comparison table showing the reported value against all computed variants.

Usage

```
compare_to_variants(x, row_index = 1)
```

Arguments

x	An effectcheck object
row_index	The row index

Value

A data frame with variant comparisons

Examples

```
res <- check_text("t(28) = 2.21, p = .035, d = 0.80")
compare_to_variants(res, 1)
```

compare_with_statcheck *Compare effectcheck results with statcheck*

Description

Runs both effectcheck and statcheck on the same text and returns a merged comparison tibble.

Usage

```
compare_with_statcheck(text, ...)
```

Arguments

text	Character string containing APA-formatted statistics
...	Additional arguments passed to check_text()

Value

A tibble with source column ("both", "effectcheck_only", "statcheck_only")

Examples

```
comp <- compare_with_statcheck("t(28) = 2.21, p = .035, d = 0.80")
print(comp)
```

count_by	<i>Count statistics by category</i>
----------	-------------------------------------

Description

Provides counts of statistics grouped by various categories.

Usage

```
count_by(x, by = c("status", "test_type", "uncertainty", "design", "source"))
```

Arguments

x	An effectcheck object
by	Character, grouping variable: "status", "test_type", "uncertainty", "design", or "source"

Value

A data frame with counts

Examples

```
results <- check_text("t(28) = 2.21, p = .035. F(1, 50) = 4.03, p = .049")
count_by(results, "status")
count_by(results, "test_type")
```

ec_identify	<i>Identify and Filter EffectCheck Results</i>
-------------	--

Description

Functions for filtering and identifying problematic results in effectcheck output. Identify problematic results

Usage

```
ec_identify(
  x,
  what = c("errors", "warnings", "decision_errors", "high_uncertainty", "insufficient",
    "all_problems"),
  ...
)
```

Arguments

x	An effectcheck object
what	Character vector specifying what to identify: <ul style="list-style-type: none"> • "errors": Results with ERROR status • "warnings": Results with WARN status • "decision_errors": Results with significance reversal • "high_uncertainty": Results with high uncertainty level • "insufficient": Results with insufficient data • "all_problems": All of the above
...	Additional arguments (ignored)

Details

Filters effectcheck results to show only problematic cases based on specified criteria.

Value

An effectcheck object containing only the identified results

Examples

```
results <- check_text("t(28) = 2.21, p = .035, d = 0.80")
errors <- ec_identify(results, "errors")
```

export_csv

Export results to CSV

Description

Exports check results to CSV format with proper handling of special characters and NA values.

Usage

```
export_csv(res, out, na = "", row.names = FALSE)
```

Arguments

res	tibble returned by check_text() / check_files()
out	output file path (csv)
na	string to use for NA values (default: "")
row.names	logical, include row names (default: FALSE)

Value

Invisible path to the generated CSV file.

Examples

```
res <- check_text("t(28) = 2.21, p = .035, d = 0.80")
export_csv(res, out = tempfile(fileext = ".csv"))
```

`export_json`*Export results to JSON*

Description

Exports check results to JSON format with structured metadata.

Usage

```
export_json(res, out, pretty = TRUE)
```

Arguments

<code>res</code>	tibble returned by <code>check_text()</code> / <code>check_files()</code>
<code>out</code>	output file path (json)
<code>pretty</code>	logical, pretty-print JSON (default: TRUE)

Value

Invisible path to the generated JSON file.

Examples

```
res <- check_text("t(28) = 2.21, p = .035, d = 0.80")
export_json(res, out = tempfile(fileext = ".json"))
```

`extract_pdf_comprehensive`*Comprehensive PDF text extraction with fallbacks*

Description

Tries multiple methods: regular text extraction, table extraction, and OCR.

Usage

```
extract_pdf_comprehensive(  
  pdf_path,  
  try_tables = TRUE,  
  try_ocr = TRUE,  
  min_text_length = 100  
)
```

Arguments

pdf_path	Path to PDF file
try_tables	Logical, attempt table extraction if regular text fails (default TRUE)
try_ocr	Logical, attempt OCR if text extraction yields little content (default TRUE)
min_text_length	Minimum text length to consider extraction successful (default 100)

Value

List with 'text' (main text), 'tables' (table text), 'ocr' (OCR text), 'method' (method used)

Examples

```
## Not run:
# Requires a PDF file and poppler-utils (pdftotext)
result <- extract_pdf_comprehensive("paper.pdf")
cat(result$text)

## End(Not run)
```

filter_by_delta	<i>Filter results by effect size delta</i>
-----------------	--

Description

Filters effectcheck results by the magnitude of effect size discrepancy.

Usage

```
filter_by_delta(x, min_delta = 0, max_delta = Inf)
```

Arguments

x	An effectcheck object
min_delta	Minimum absolute delta to include (default 0)
max_delta	Maximum absolute delta to include (default Inf)

Value

An effectcheck object containing only results within the delta range

Examples

```
results <- check_text("t(28) = 2.21, p = .035, d = 0.80")
filter_by_delta(results, min_delta = 0.1)
```

filter_by_source *Filter results by source file*

Description

Filters effectcheck results to show only results from specific files.

Usage

```
filter_by_source(x, files, pattern = FALSE)
```

Arguments

x	An effectcheck object
files	Character vector of file names or patterns to include
pattern	Logical, if TRUE treat files as regex patterns (default FALSE)

Value

An effectcheck object containing only results from specified files

Examples

```
results <- check_text("t(28) = 2.21, p = .035, d = 0.80")
filter_by_source(results, "text_input")
```

filter_by_test_type *Filter results by test type*

Description

Filters effectcheck results to show only specific test types.

Usage

```
filter_by_test_type(x, types)
```

Arguments

x	An effectcheck object
types	Character vector of test types to include (e.g., "t", "F", "r", "chisq", "z")

Value

An effectcheck object containing only the specified test types

Examples

```
results <- check_text("t(28) = 2.21, p = .035. F(1, 50) = 4.03, p = .049")
filter_by_test_type(results, "t")
```

filter_by_uncertainty *Filter results by uncertainty level*

Description

Filters effectcheck results by uncertainty level.

Usage

```
filter_by_uncertainty(x, levels)
```

Arguments

x	An effectcheck object
levels	Character vector of uncertainty levels to include ("low", "medium", "high")

Value

An effectcheck object containing only the specified uncertainty levels

Examples

```
results <- check_text("t(28) = 2.21, p = .035, d = 0.80")
filter_by_uncertainty(results, "high")
```

format_variants *Format variants for display*

Description

Creates a formatted string representation of variants for a row.

Usage

```
format_variants(x, row_index = 1, include_alternatives = TRUE)
```

Arguments

x	An effectcheck object
row_index	The row index
include_alternatives	Whether to include alternative suggestions

Value

A character string with formatted variant information

Examples

```
res <- check_text("t(28) = 2.21, p = .035, d = 0.80")
cat(format_variants(res, 1))
```

generate_report	<i>Generate a submission-ready EffectCheck report</i>
-----------------	---

Description

Creates a self-contained HTML report with executive summary, color-coded results table, expandable details, reproducible R code, and footer stamp.

Usage

```
generate_report(
  res,
  out,
  format = "html",
  title = "EffectCheck Report",
  author = NULL,
  source_name = NULL,
  include_repro_code = TRUE,
  style = "beginner"
)
```

Arguments

res	tibble returned by check_text() / check_files()
out	output file path (html)
format	Output format: "html" (default) or "pdf" (requires rmarkdown)
title	Report title (default: "EffectCheck Report")
author	Author name (optional)
source_name	Source file name (optional)
include_repro_code	Logical, include reproducible R code section (default TRUE)
style	Report style: "beginner" for plain English narrative (default), "expert" for the traditional technical table format

Value

Invisible path to the generated report file

Examples

```
res <- check_text("t(28) = 2.21, p = .035, d = 0.80")
generate_report(res, out = tempfile(fileext = ".html"))
```

get_alternatives *Get alternative suggestions for a row*

Description

Get alternative suggestions for a row

Usage

```
get_alternatives(x, row_index = 1)
```

Arguments

x	An effectcheck object
row_index	The row index

Value

A list of alternative effect size suggestions

Examples

```
res <- check_text("t(28) = 2.21, p = .035, d = 0.80")
get_alternatives(res, 1)
```

get_decision_errors *Get decision errors from effectcheck results*

Description

Extracts results where the significance decision would be reversed (i.e., reported as significant when computed is not, or vice versa).

Usage

```
get_decision_errors(x)
```

Arguments

x	An effectcheck object
---	-----------------------

Value

An effectcheck object containing only decision errors

Examples

```
results <- check_text("t(28) = 2.21, p = .035, d = 0.80")
get_decision_errors(results)
```

get_effect_family *Get effect size family information*

Description

Returns information about an effect size family and its variants.

Usage

```
get_effect_family(effect_type)
```

Arguments

effect_type The effect size type (e.g., "d", "eta2", "r")

Value

A list with family, variants, alternatives, and description

Examples

```
get_effect_family("d")
```

get_errors *Get errors from effectcheck results*

Description

Convenience function to extract only ERROR status results.

Usage

```
get_errors(x)
```

Arguments

x An effectcheck object

Value

An effectcheck object containing only errors

Examples

```
results <- check_text("t(28) = 2.21, p = .035, d = 0.80")
get_errors(results)
```

get_same_type_variants

Get same-type variants for a row

Description

Get same-type variants for a row

Usage

```
get_same_type_variants(x, row_index = 1)
```

Arguments

x	An effectcheck object
row_index	The row index

Value

A list of same-type variants with their values and metadata

Examples

```
res <- check_text("t(28) = 2.21, p = .035, d = 0.80")
get_same_type_variants(res, 1)
```

get_variants

Get all variants for a specific row

Description

Extracts and parses the all_variants JSON structure for a given row.

Usage

```
get_variants(x, row_index = 1)
```

Arguments

- x An effectcheck object
- row_index The row index to extract variants from

Value

A list with same_type and alternatives sublists

Examples

```
res <- check_text("t(28) = 2.21, p = .035, d = 0.80")  
get_variants(res, 1)
```

get_variant_metadata *Get variant metadata*

Description

Returns metadata for a specific effect size variant type.

Usage

```
get_variant_metadata(variant_name)
```

Arguments

- variant_name The name of the variant (e.g., "d_ind", "dz", "eta2")

Value

A list with name, assumptions, when_to_use, and formula

Examples

```
get_variant_metadata("d_ind")
```

get_warnings *Get warnings from effectcheck results*

Description

Convenience function to extract only WARN status results.

Usage

```
get_warnings(x)
```

Arguments

x An effectcheck object

Value

An effectcheck object containing only warnings

Examples

```
results <- check_text("t(28) = 2.21, p = .035, d = 0.80")
get_warnings(results)
```

is.effectcheck *Test if object is an effectcheck object*

Description

Test if object is an effectcheck object

Usage

```
is.effectcheck(x)
```

Arguments

x Object to test

Value

Logical

Examples

```
res <- check_text("t(28) = 2.21, p = .035, d = 0.80")
is.effectcheck(res)
```

parse_text	<i>Parse APA-style stats and effects from text</i>
------------	--

Description

Extracts test statistics, effect sizes, confidence intervals, and sample sizes from APA-style text. Includes context window extraction for design inference.

Usage

```
parse_text(text, context_window_size = 2)
```

Arguments

text	Character vector of text to parse
context_window_size	Number of sentences before/after to capture (default 2)

Value

Tibble with parsed elements including context windows

Examples

```
parsed <- parse_text("t(28) = 2.21, p = .035, d = 0.80")
parsed$test_type
parsed$stat_value
```

plot.effectcheck	<i>Plot method for effectcheck objects</i>
------------------	--

Description

Creates visualizations of effectcheck results.

Usage

```
## S3 method for class 'effectcheck'
plot(x, type = c("status", "uncertainty", "test_type", "delta", "all"), ...)
```

Arguments

x	An effectcheck object
type	Type of plot: "status", "uncertainty", "test_type", "delta", or "all"
...	Additional arguments passed to plotting functions

Value

Invisibly returns x.

Examples

```
res <- check_text("t(28) = 2.21, p = .035, d = 0.80")
plot(res, type = "status")
```

`print.effectcheck` *Print method for effectcheck objects*

Description

Displays a formatted summary of effectcheck results.

Usage

```
## S3 method for class 'effectcheck'
print(x, short = TRUE, n = 10, ...)
```

Arguments

x	An effectcheck object
short	Logical, if TRUE show abbreviated output (default TRUE)
n	Maximum number of rows to display (default 10)
...	Additional arguments (ignored)

Value

Invisibly returns x.

Examples

```
res <- check_text("t(28) = 2.21, p = .035, d = 0.80")
print(res)
```

```
print.effectcheck_comparison
```

Print method for effectcheck comparison

Description

Print method for effectcheck comparison

Usage

```
## S3 method for class 'effectcheck_comparison'  
print(x, ...)
```

Arguments

x	An effectcheck_comparison object
...	Additional arguments (ignored)

Value

Invisibly returns x.

Examples

```
comp <- compare_with_statcheck("t(28) = 2.21, p = .035, d = 0.80")  
print(comp)
```

```
print.summary.effectcheck
```

Print method for summary.effectcheck objects

Description

Print method for summary.effectcheck objects

Usage

```
## S3 method for class 'summary.effectcheck'  
print(x, ...)
```

Arguments

x	A summary.effectcheck object
...	Additional arguments (ignored)

Value

Invisibly returns x.

Examples

```
res <- check_text("t(28) = 2.21, p = .035, d = 0.80")
s <- summary(res)
print(s)
```

rbind.effectcheck	<i>Combine effectcheck objects</i>
-------------------	------------------------------------

Description

Combine effectcheck objects

Usage

```
## S3 method for class 'effectcheck'
rbind(...)
```

Arguments

... effectcheck objects to combine

Value

Combined effectcheck object

Examples

```
res1 <- check_text("t(28) = 2.21, p = .035")
res2 <- check_text("F(1, 50) = 4.03, p = .049")
combined <- rbind(res1, res2)
```

read_any_text	<i>Read text from .docx, .html, .txt, or .pdf</i>
---------------	---

Description

Read text from .docx, .html, .txt, or .pdf

Usage

```
read_any_text(path, try_tables = TRUE, try_ocr = FALSE)
```

Arguments

path	File path
try_tables	Logical, attempt table extraction from PDFs (default TRUE)
try_ocr	Logical, attempt OCR for scanned PDFs (default FALSE)

Value

character vector of full text

Examples

```
tmp <- tempfile(fileext = ".txt")
writeLines("t(28) = 2.21, p = .035, d = 0.80", tmp)
text <- read_any_text(tmp)
unlink(tmp)
```

render_report	<i>Render an enhanced HTML report</i>
---------------	---------------------------------------

Description

Creates an HTML report with summary statistics, expandable sections, and uncertainty visualization.

Usage

```
render_report(res, out)
```

Arguments

res	tibble returned by check_text() / check_files()
out	output file path (html)

Value

Invisible path to the generated HTML report file.

Examples

```
res <- check_text("t(28) = 2.21, p = .035, d = 0.80")
render_report(res, out = tempfile(fileext = ".html"))
```

summary.effectcheck *Summary method for effectcheck objects*

Description

Provides comprehensive summary statistics for effectcheck results.

Usage

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

Arguments

object	An effectcheck object
...	Additional arguments (ignored)

Value

A list of class "summary.effectcheck" containing summary statistics

Examples

```
res <- check_text("t(28) = 2.21, p = .035, d = 0.80")
summary(res)
```

[.effectcheck	<i>Subset method for effectcheck objects</i>
---------------	--

Description

Preserves effectcheck class when subsetting.

Usage

```
## S3 method for class 'effectcheck'  
x[...]
```

Arguments

x	An effectcheck object
...	Subsetting arguments

Value

An effectcheck object

Examples

```
res <- check_text("t(28) = 2.21, p = .035. F(1, 50) = 4.03, p = .049")  
res[1, ]
```

Index

.effectcheck_version, 3
[.effectcheck, 31

check_dir, 7
check_file, 8
check_files, 9
check_text, 9
checkDOCXdir, 3
checkHTML, 4
checkHTMLdir, 4
checkPDF, 5
checkPDFdir, 6
compare_file_with_statcheck, 11
compare_to_variants, 12
compare_with_statcheck, 12
count_by, 13

ec_identify, 13
export_csv, 14
export_json, 15
extract_pdf_comprehensive, 15

filter_by_delta, 16
filter_by_source, 17
filter_by_test_type, 17
filter_by_uncertainty, 18
format_variants, 18

generate_report, 19
get_alternatives, 20
get_decision_errors, 20
get_effect_family, 21
get_errors, 21
get_same_type_variants, 22
get_variant_metadata, 23
get_variants, 22
get_warnings, 24

is.effectcheck, 24

parse_text, 25

plot.effectcheck, 25
print.effectcheck, 26
print.effectcheck_comparison, 27
print.summary.effectcheck, 27

rbind.effectcheck, 28
read_any_text, 29
render_report, 29

summary.effectcheck, 30