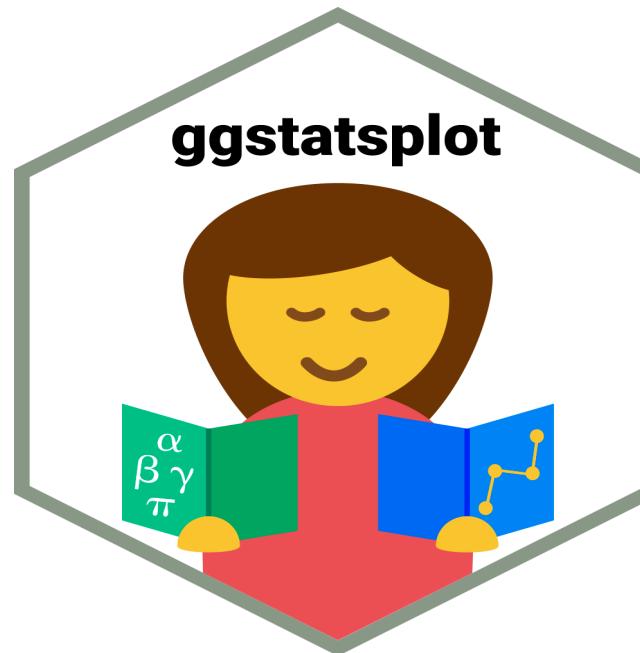


{ggstatsplot}: Informative Statistical Visualizations

Indrajeet Patil

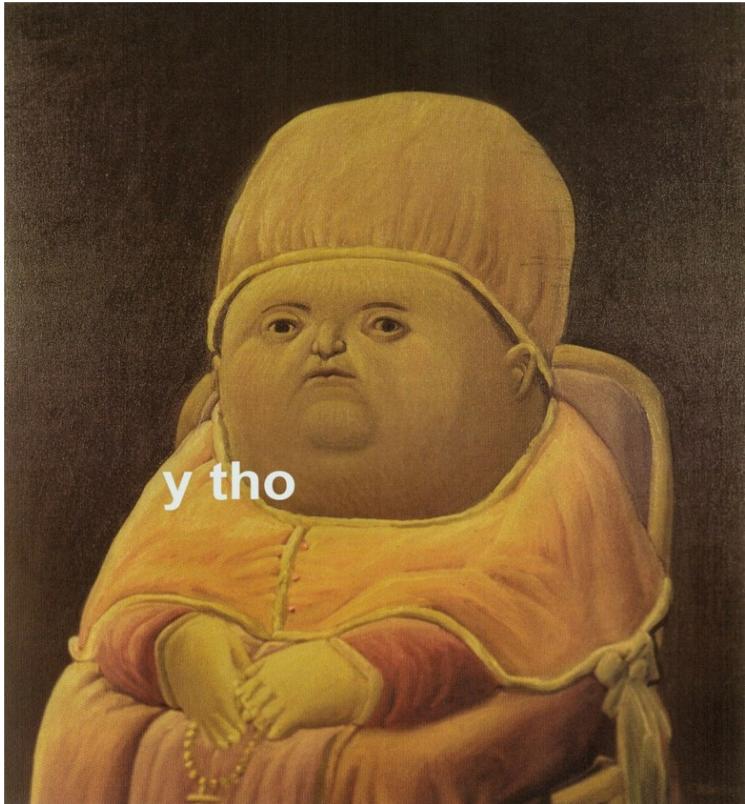


Plan

- Why `ggstatsplot`?
- Primary functions
- Customizability
- Benefits
- Misconceptions
- Limitations

Raison d'être

Current CRAN package count >23,000

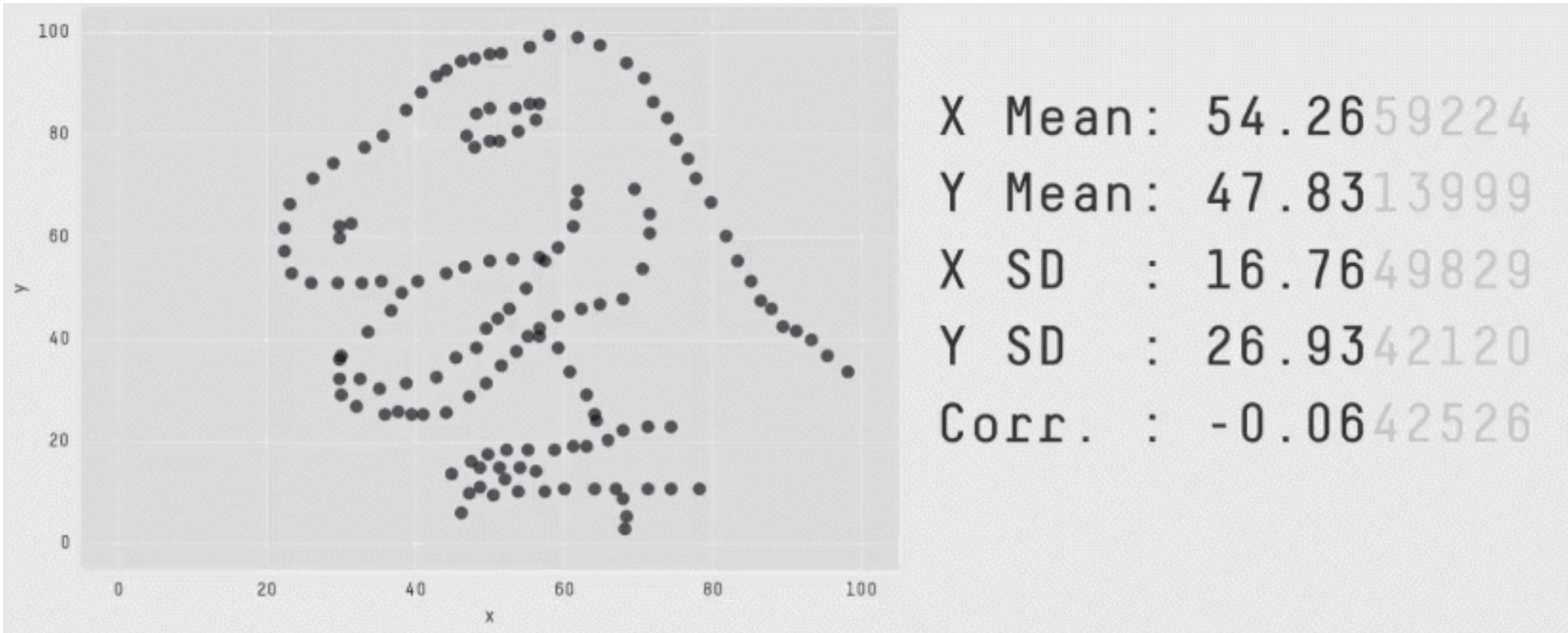


{ggstatsplot} provides

📊 information-rich plots with statistical details

✍️ suitable for faster (exploratory) data analysis and scholarly reports

Informative graphic = a thousand words



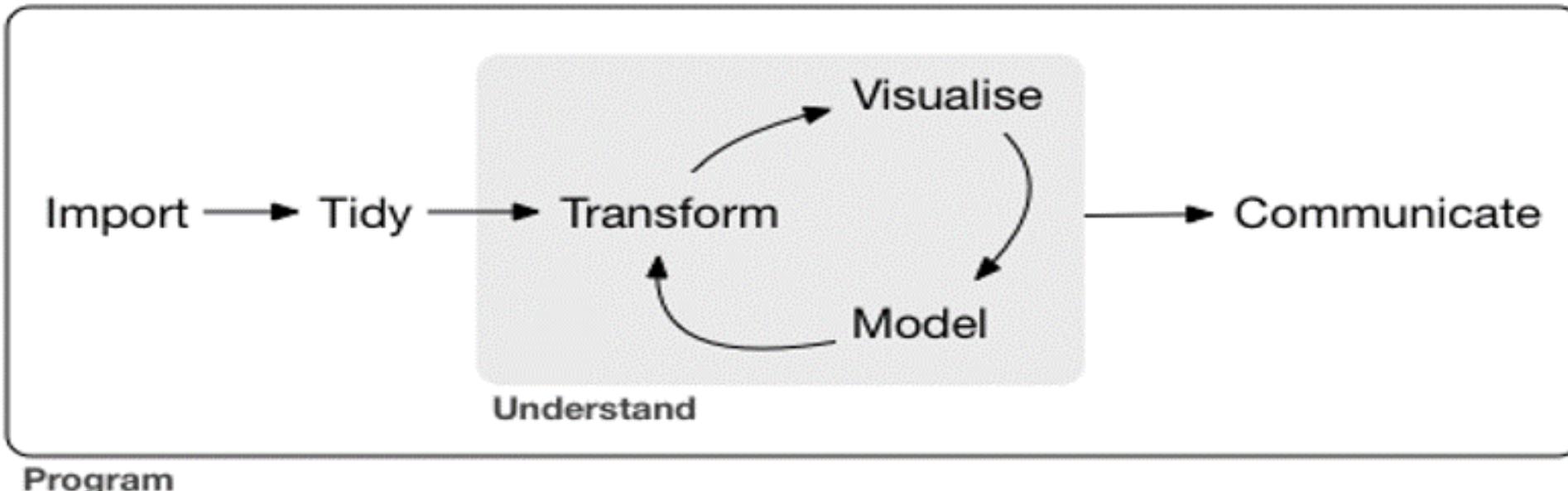
Graphical summaries can reveal problems not visible from numerical statistics.

Ready-made plot = no customization

The grammar of graphics is a powerful framework ([Wilkinson, 2011](#)) and can help you make *any* graphics fitting your specific data visualization needs! But...



Simpler/faster data analysis workflow



In a typical *exploratory* data analysis workflow, **data visualization** and **statistical modeling** are two different phases: visualization informs modeling, and modeling can suggest a different visualization, and so on and so forth.



Central idea of `{ggstatsplot}`

Simple: combine these two phases into one!

And a LOT more!

...but we will come back to that later 

Let's get started first!

Package available for installation on CRAN and GitHub:

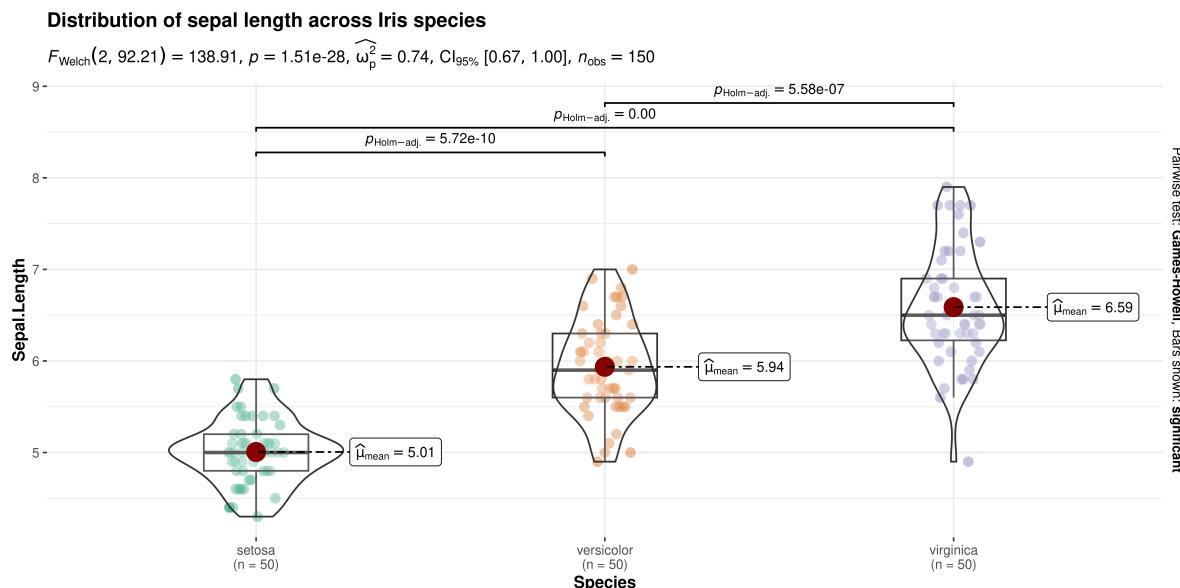
Type	Command
Release	<code>install.packages("ggstatsplot")</code>
Development	<code>pak::pak("IndrajeetPatil/ggstatsplot")</code>

Example function

ggbetweenstats()

For between-group comparisons

```
1 ggbetweenstats(  
2   data = iris,  
3   x     = Species,  
4   y     = Sepal.Length,  
5   title = "Distribution of sepal length across Iris species"  
6 )
```



! Important

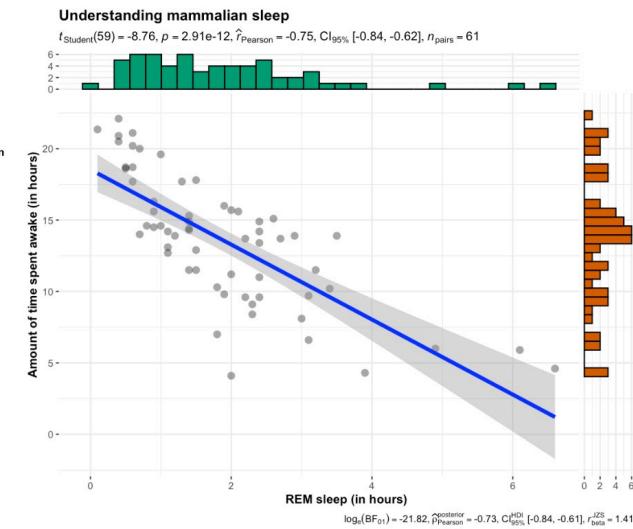
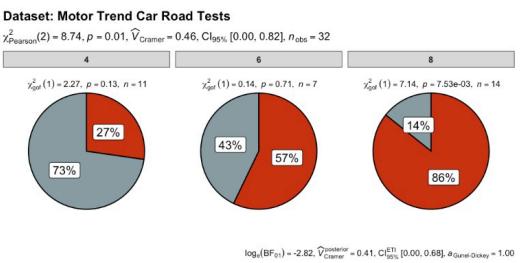
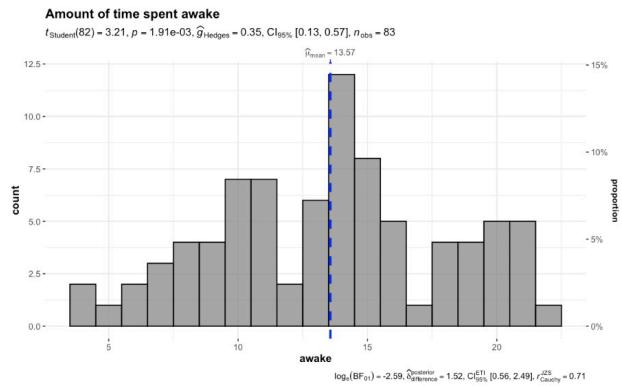
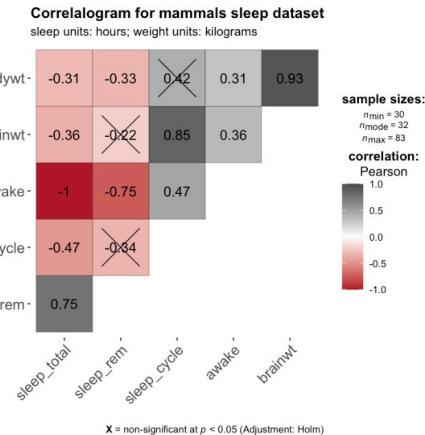
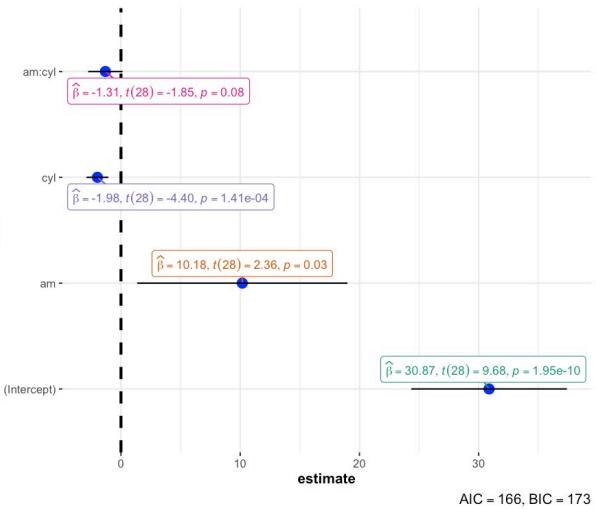
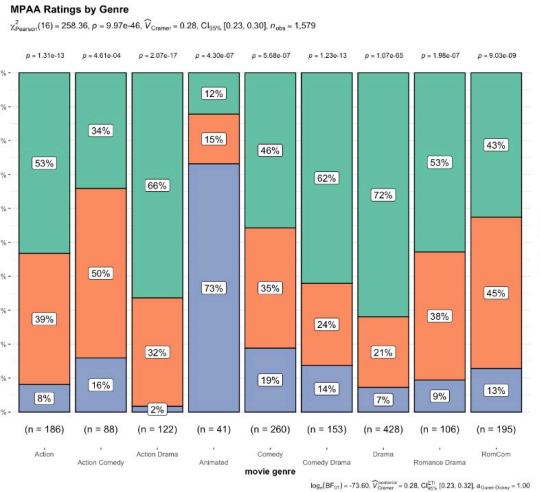
Defaults

- raw data + distributions
- descriptive statistics
- inferential statistics
- effect size + uncertainty
- pairwise comparisons
- Bayesian hypothesis-testing
- Bayesian estimation

Statistical approaches available

- parametric
- parametric
- robust
- Bayesian

Other functions



Why use `{ggstatsplot}` ?



Results *in context* of the data



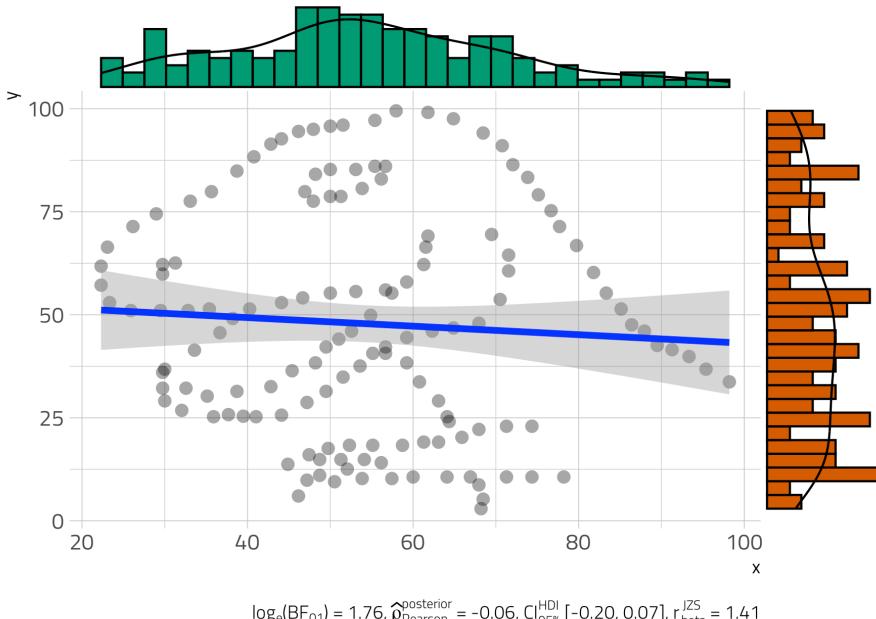
Standard approach

Pearson's correlation test revealed that, across 142 participants, variable x was negatively correlated with variable y : $t(140) = -0.76, p = .446$. The effect size ($r = -0.06, 95\%CI[-.23, .10]$) was small, as per Cohen's (1988) conventions. The Bayes Factor for the same analysis revealed that the data were 5.81 times more probable under the null hypothesis as compared to the alternative hypothesis. This can be considered moderate evidence (Jeffreys, 1961) in favor of the null hypothesis (absence of any correlation between x and y).

{ggstatsplot} approach

Relationship between x and y

$t_{\text{Student}}(140) = -0.76, p = 0.446, \hat{r}_{\text{Pearson}} = -0.06, \text{CI}_{95\%} [-0.23, 0.10], n_{\text{pairs}} = 142$



Toggling statistical approaches



Parametric

```
1 # anova
2 ggbetweenstats(
3   data = mtcars,
4   x = cyl,
5   y = wt,
6   type = "p"
7 )
8
9 # correlation analysis
10 ggscatterstats(
11   data = mtcars,
12   x = wt,
13   y = mpg,
14   type = "p"
15 )
16
17 # t-test
18 gghistostats(
19   data = mtcars,
20   x = wt,
21   test.value = 2,
22   type = "p"
23 )
```

Non-parametric

```
1 # anova
2 ggbetweenstats(
3   data = mtcars,
4   x = cyl,
5   y = wt,
6   type = "np"
7 )
8
9 # correlation analysis
10 ggscatterstats(
11   data = mtcars,
12   x = wt,
13   y = mpg,
14   type = "np"
15 )
16
17 # t-test
18 gghistostats(
19   data = mtcars,
20   x = wt,
21   test.value = 2,
22   type = "np"
23 )
```

Alternative to painful workflow

Load 'em up!

- 📦 for inferential statistics (e.g. `stats`)
- 📦 computing effect size + CIs (e.g. `effectsize`)
- 📦 for descriptives (e.g. `skimr`)
- 📦 pairwise comparisons (e.g. `multcomp`)
- 📦 Bayesian hypothesis testing (e.g. `BayesFactor`)
- 📦 Bayesian estimation (e.g. `bayestestR`)
- 📦 .



Things to worry about 🤔

- 🤔 accepts dataframe, vector, matrix?
- 🤔 long/wide format data?
- 🤔 works with `NAs`?
- 🤔 returns list, dataframe, arrays?
- 🤔 works with tibbles?
- 🤔 has all necessary details?
- 🤔 .



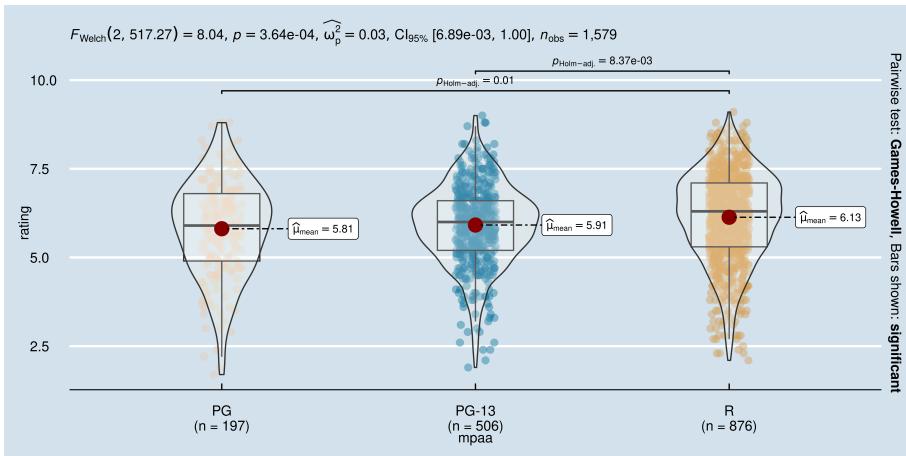
Plot Customizability

“What if I don’t like the default plots?” 

Changing aesthetics



```
1 ggbetweenstats(  
2   data = movies_long,  
3   x = mpaa,  
4   y = rating,  
5   ggtheme = ggthemes::theme_economist(),  
6   palette = "Darjeeling2",  
7   package = "wesanderson"  
8 )
```



Aesthetic preferences not an excuse to avoid `{ggstatsplot}`! 😺 Any `ggplot` theme or palette can be used.

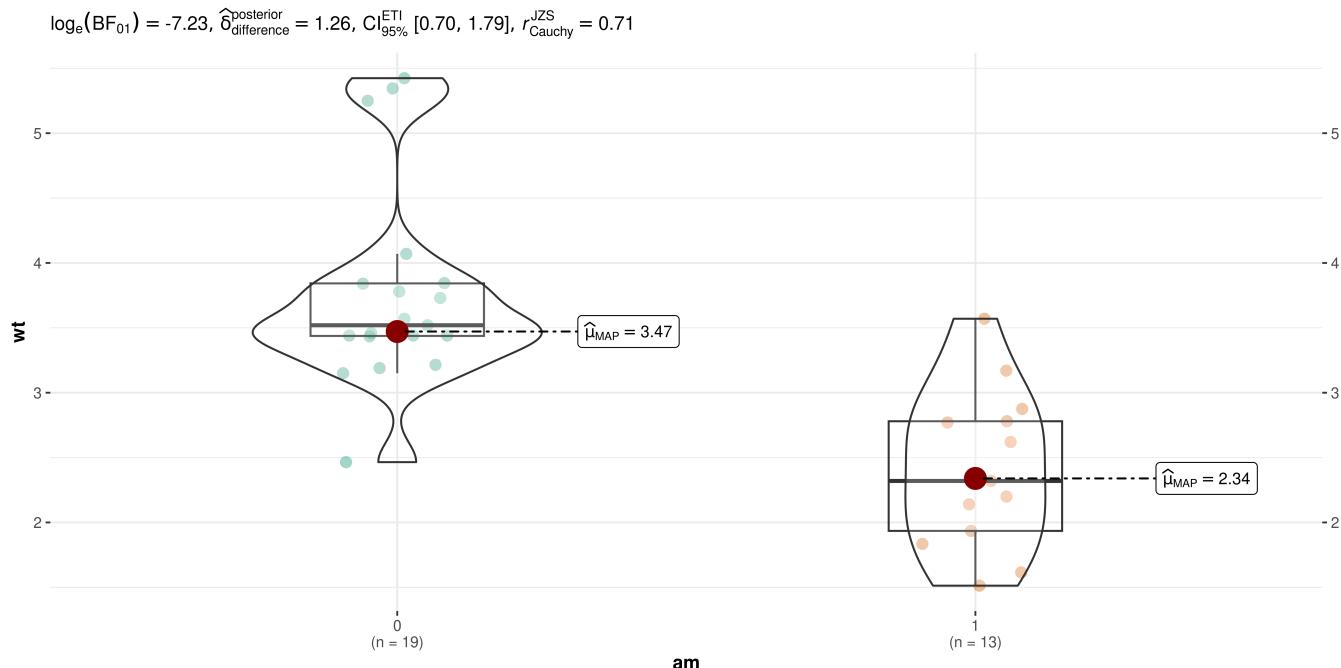
N.B. The default palette is **colorblind-friendly**.

Modification with

{ggplot2} 

You can modify `{ggstatsplot}` plots further using `{ggplot2}` functions. 

```
1 ggbetweenstats(  
2   data = mtcars,  
3   x = am,  
4   y = wt,  
5   type = "bayes"  
6 ) +  
7   scale_y_continuous(sec.axis = dup_axis())
```

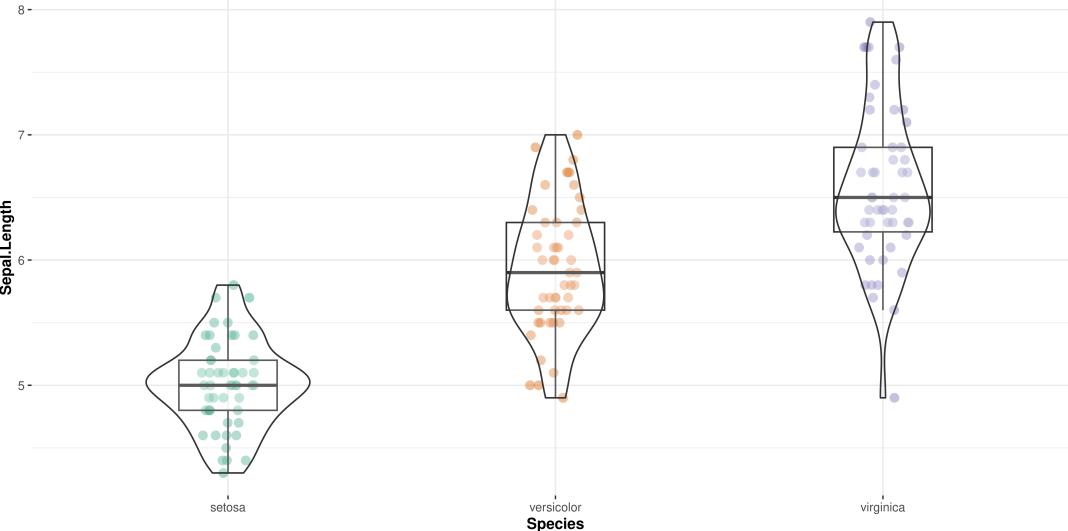


Too much information



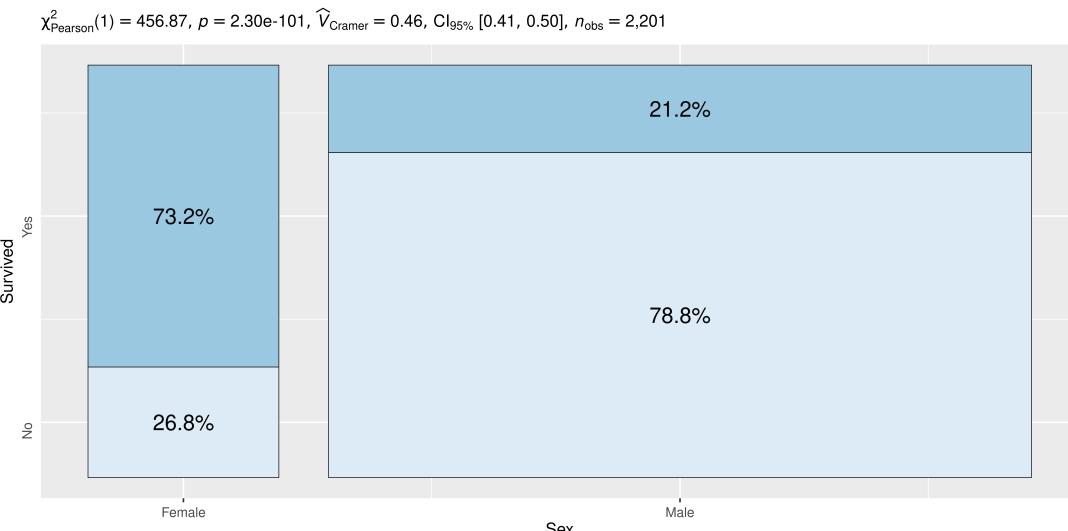
Get only plots:

```
1 ggbetweenstats(
2   data = iris,
3   x = Species,
4   y = Sepal.Length,
5   # turn off statistical analysis
6   centrality.plotting = FALSE,
7   results.subtitle = FALSE,
8   bf.message = FALSE,
9   # turn off pairwise comparisons
10 pairwise.display = "none"
11 )
```



Get only expressions:

```
1 stats_expr <- ggpiestats(
2   Titanic_full, Survived, Sex,
3 ) %>% extract_subtitle()
4
5 ggiraphExtra::ggSpine(
6   data = Titanic_full,
7   aes(x = Sex, fill = Survived)
8 ) +
9   labs(subtitle = stats_expr)
```

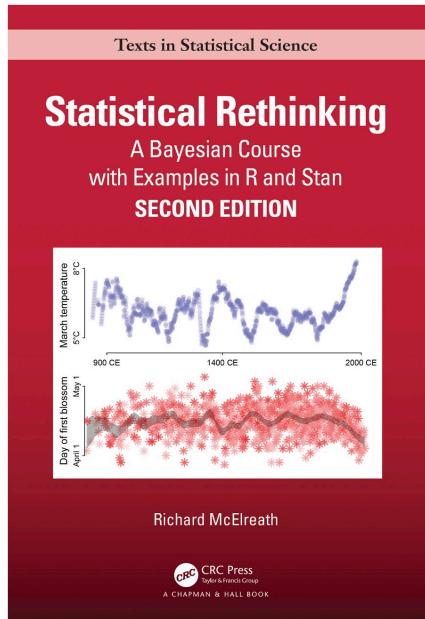


Negatives

“Golem of Prague” issue

Promotes mindless application of statistical tests.

Easy-to-use software can lead to misuse.



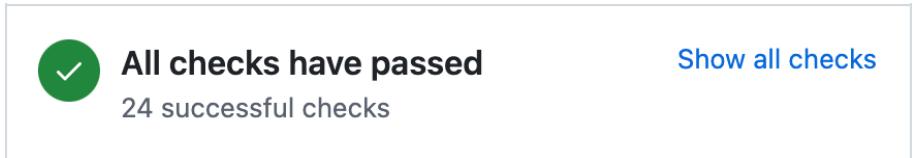
Clunky API

- Too many arguments to remember.
- Not a “real” ggplot2 extension.
- Limited number of functions.
- Expects non-trivial statistical proficiency.

Positives

Quality Assurance

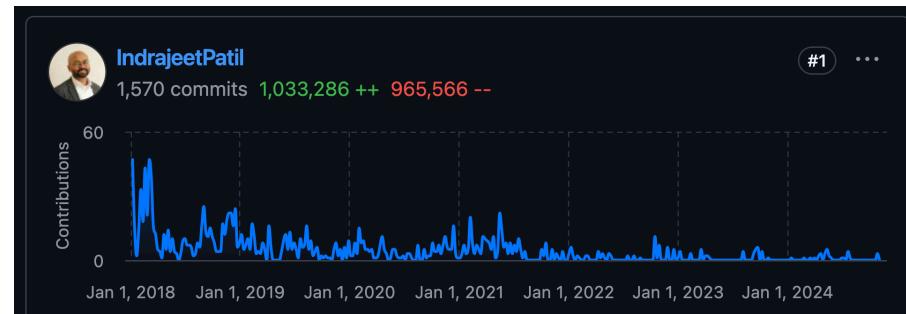
Each commit must pass many QA checks:



(i) **CI Checks (GitHub Actions)**

- Unit tests (random-order)
- Code coverage (100%)
- Linting (0 lints)
- Formatting (0 issues)
- Documentation (website, link rot, examples)
- CRAN checks (0 E, 0 W, 0 N)
- Pre-commit hooks (0 issues)
- Portability (Linux, macOS, Windows)
- Robustness (dependencies, R versions)

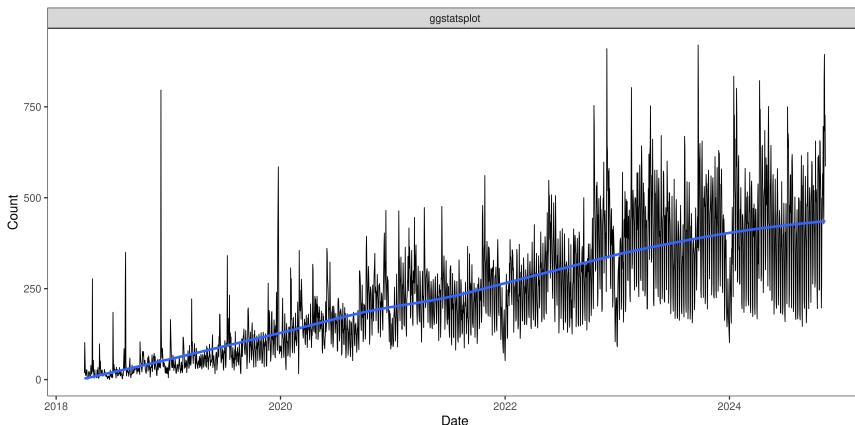
Healthy code base:



Reception

! Total downloads > 500K (97 percentile)

```
1 library(packageRank)
2 plot(
3   cranDownloads("ggstatsplot", from = "2018-04-03", to = Sys.Date()),
4   graphics = "ggplot2", smooth = TRUE
5 )
```



Total citations > 1000



The Journal of Open Source Software

Visualizations with statistical details: The 'ggstatsplot' approach

Indrajeet Patil¹

¹ Center for Humans and Machines, Max Planck Institute for Human Development, Berlin, Germany

DOI: [10.21105/joss.03167](https://doi.org/10.21105/joss.03167)

Software

- [Review](#)
- [Repository](#)
- [Archive](#)

Editor: Charlotte Soneson

Reviewers:

- @njtierney
- @kevinrue

Submitted: 30 March 2021

Published: 25 May 2021

Summary

Graphical displays can reveal problems in a statistical model that might not be apparent from purely numerical summaries. Such visualizations can also be helpful for the reader to evaluate the validity of a model if it is reported in a scholarly publication or report. But, given the onerous costs involved, researchers often avoid preparing information-rich graphics and exploring several statistical approaches or tests available. The *ggstatsplot* package in the R programming language (R Core Team, 2021) provides a one-line syntax to enrich *ggplot2*-based visualizations with the results from statistical analysis embedded in the visualization itself. In doing so, the package helps researchers adopt a rigorous, reliable, and robust data exploratory and reporting workflow.



Enhancing {ggplot2} plots with statistical analysis 📈🔊

● R ⭐ 2k ⚙ 190

Visualizations with statistical details: The 'ggstatsplot' approach

I Patil

Journal of Open Source Software 6 (61), 3167, 2021

1277

*

Conclusion

Exploring data made easy! 😊

In summary, the `ggstatsplot` approach-

- (a) avoids errors in statistical reporting,
 - (b) highlights the importance of the effect by providing effect size measures by default,
 - (c) provides an easy way to evaluate *absence* of an effect using Bayesian framework,
 - (d) demands to evaluate statistical analysis in the context of the underlying data,
- and is (e) easy and (f) simple enough that somebody with little coding experience can use it without making an error.

For more

Source code for these slides can be found [on GitHub](#).

If you are interested in good programming and software development practices, check out my other [slide decks](#).

Find me at...

 Twitter

 LinkedIn

 GitHub

 Website

 E-mail

Thank You 😊

Session information

```
1 sessioninfo::session_info(include_base = TRUE)

- Session info -----
  setting  value
  version   R version 4.4.2 (2024-10-31)
  os        Ubuntu 22.04.5 LTS
  system    x86_64, linux-gnu
  hostname  fv-az1456-909
  ui         X11
  language  (EN)
  collate   C.UTF-8
  ctype     C.UTF-8
  tz         UTC
  date      2024-11-10
  pandoc   3.5 @ /opt/hostedtoolcache/pandoc/3.5/x64/ (via rmarkdown)
  quarto    1.6.33 @ /usr/local/bin/quarto
```

```
- Packages -----
  package      * version    date (UTC) lib source
  base          * 4.4.2      2024-10-31 [3] local
  BayesFactor    0.9.12-4.7 2024-01-24 [1] RSPM
  bayestestR     0.15.0     2024-10-17 [1] RSPM
  . . . . .
```

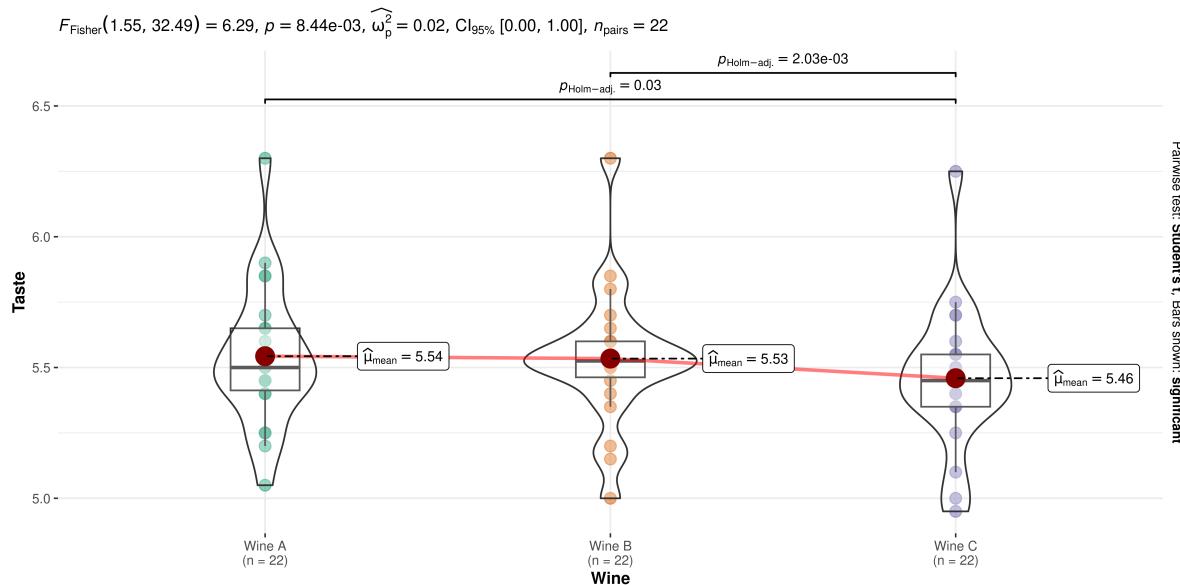
Appendix

Examples of other functions

ggwithinstats()

Hypothesis about group differences: repeated measures design

```
1 ggwithinstats(  
2   data = WRS2::WineTasting,  
3   x = Wine,  
4   y = Taste  
5 )
```



! Important

Defaults

- raw data + distributions
- descriptive statistics
- inferential statistics
- effect size + uncertainty
- pairwise comparisons
- Bayesian hypothesis-testing
- Bayesian estimation

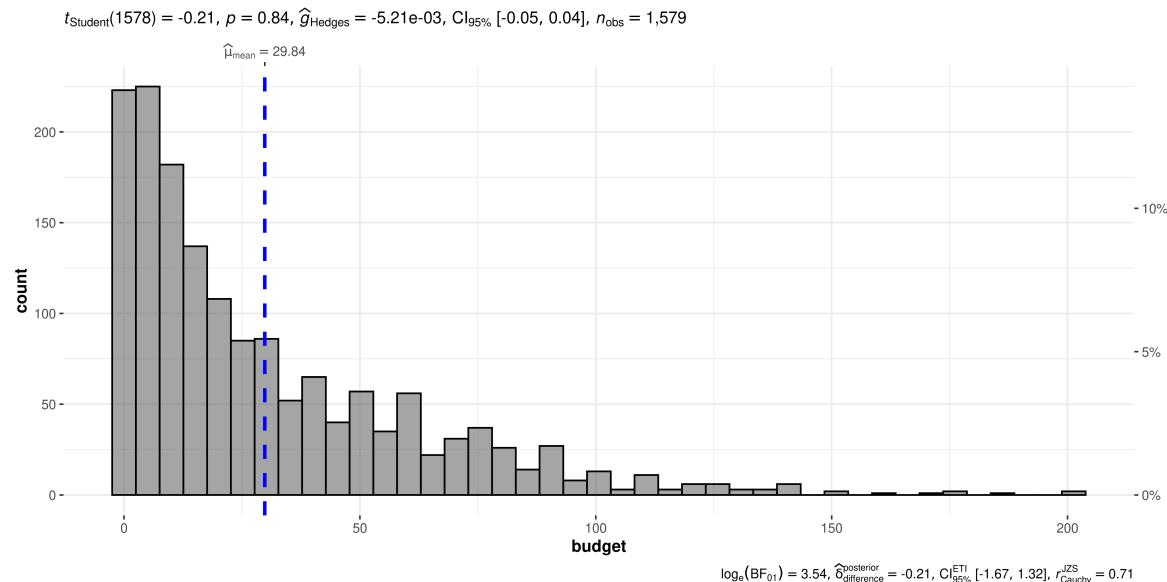
Statistical approaches available

- parametric
- parametric
- robust
- Bayesian

gghistostats()

Distribution of a numeric variable

```
1 gghistostats(  
2   data = movies_long,  
3   x = budget,  
4   test.value = 30  
5 )
```



! Important

💡 Defaults

- counts + proportion for bins
- descriptive statistics
- inferential statistics
- effect size + uncertainty
- pairwise comparisons
- Bayesian hypothesis-testing
- Bayesian estimation

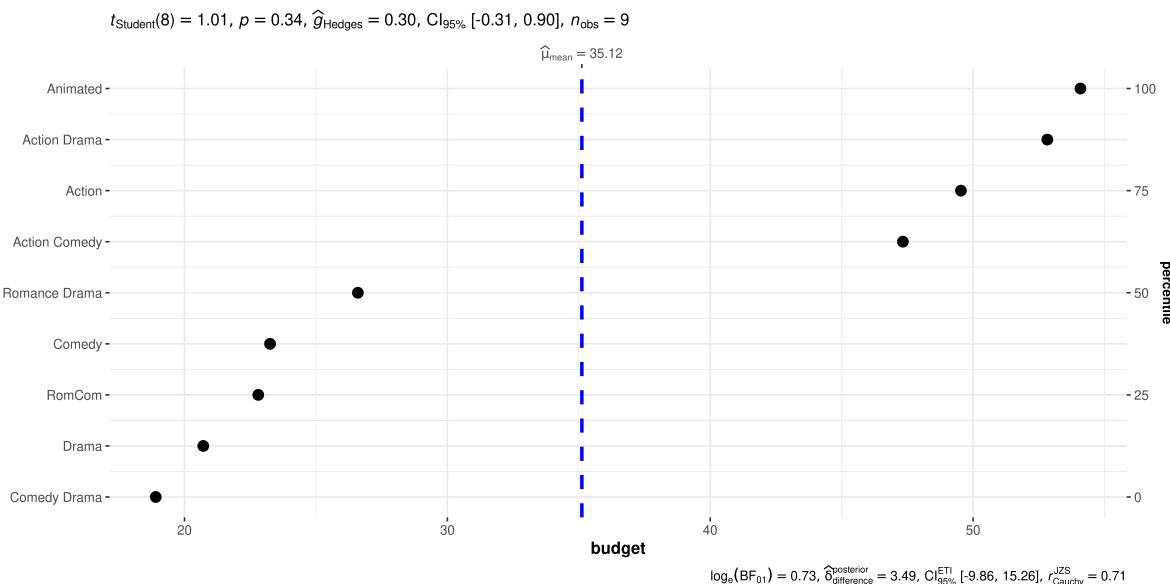
Statistical approaches available

- parametric
- parametric
- robust
- Bayesian

ggdotplotstats()

Labeled numeric variable

```
1 ggdotplotstats(  
2   data = movies_long,  
3   x = budget,  
4   y = genre,  
5   test.value = 30  
6 )
```



! Important

Defaults

- descriptive statistics
- inferential statistics
- effect size + uncertainty
- pairwise comparisons
- Bayesian hypothesis-testing
- Bayesian estimation

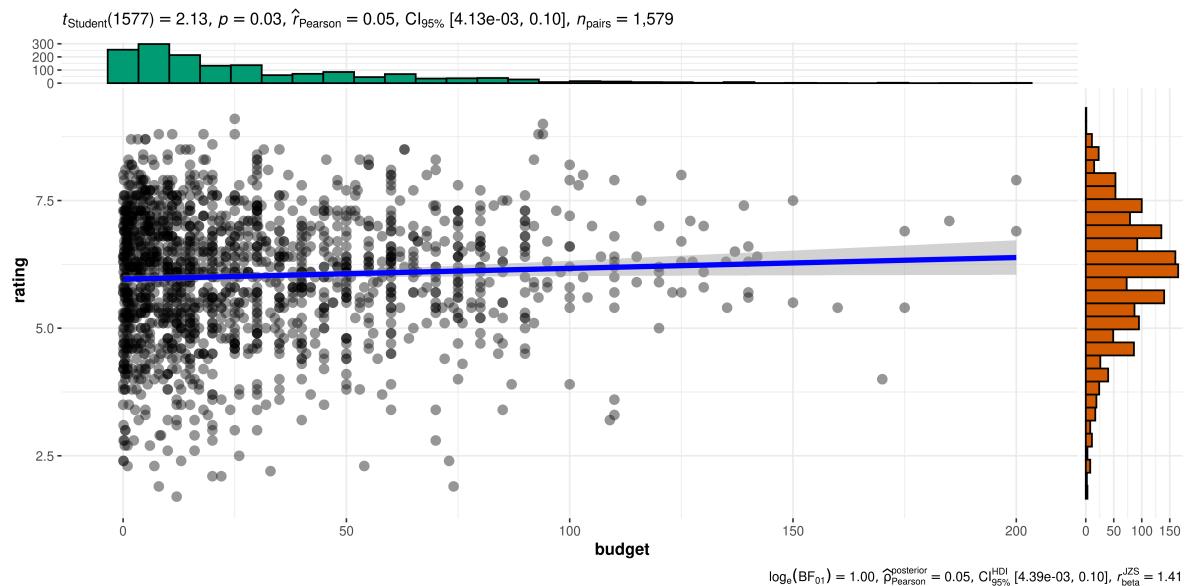
Statistical approaches available

- parametric
- parametric
- robust
- Bayesian

ggscatterstats()

Hypothesis about correlation: Two numeric variables

```
1 ggscatterstats(  
2   data = movies_long,  
3   x = budget,  
4   y = rating  
5 )
```



! Important

Defaults

- joint distribution
- marginal distribution
- effect size + uncertainty
- pairwise comparisons
- Bayesian hypothesis-testing
- Bayesian estimation

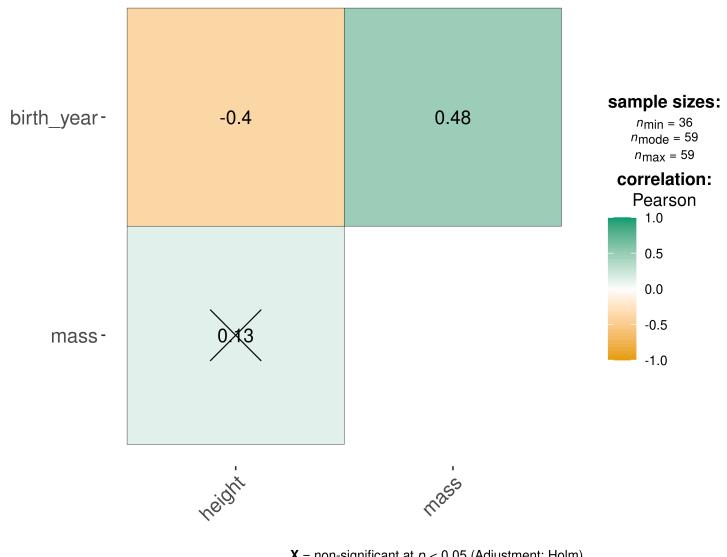
Statistical approaches available

- parametric
- parametric
- robust
- Bayesian

ggcorrmat()

Hypothesis about correlation: Multiple numeric variables

```
1 ggcorrmat(dplyr::starwars)
```



! Important

Defaults

- inferential statistics
- effect size + uncertainty
- careful handling of NAs
- partial correlations

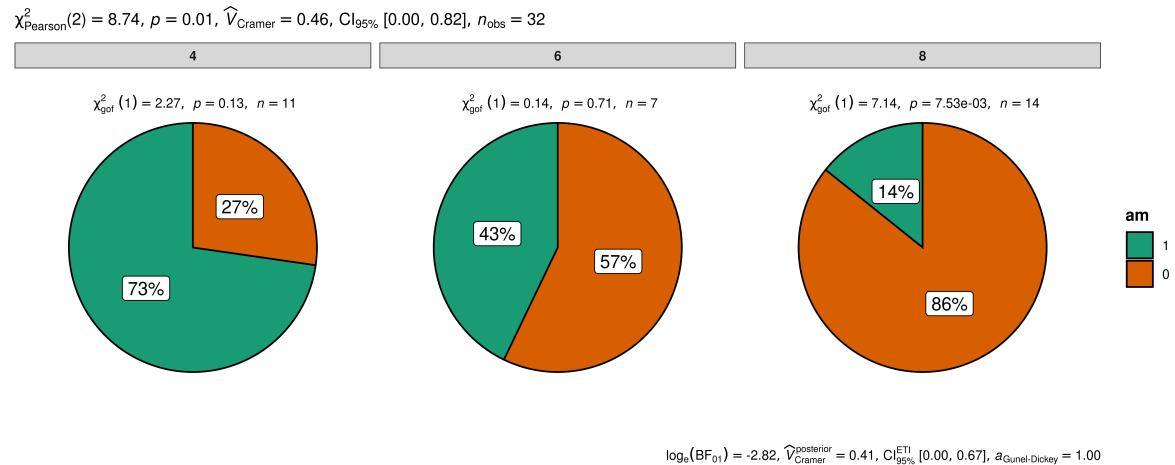
Statistical approaches available

- parametric
- parametric
- robust
- Bayesian

ggpiestats()

Hypothesis about composition of categorical variables

```
1 ggpiestats(  
2   data = mtcars,  
3   x = am,  
4   y = cyl  
5 )
```



! Important

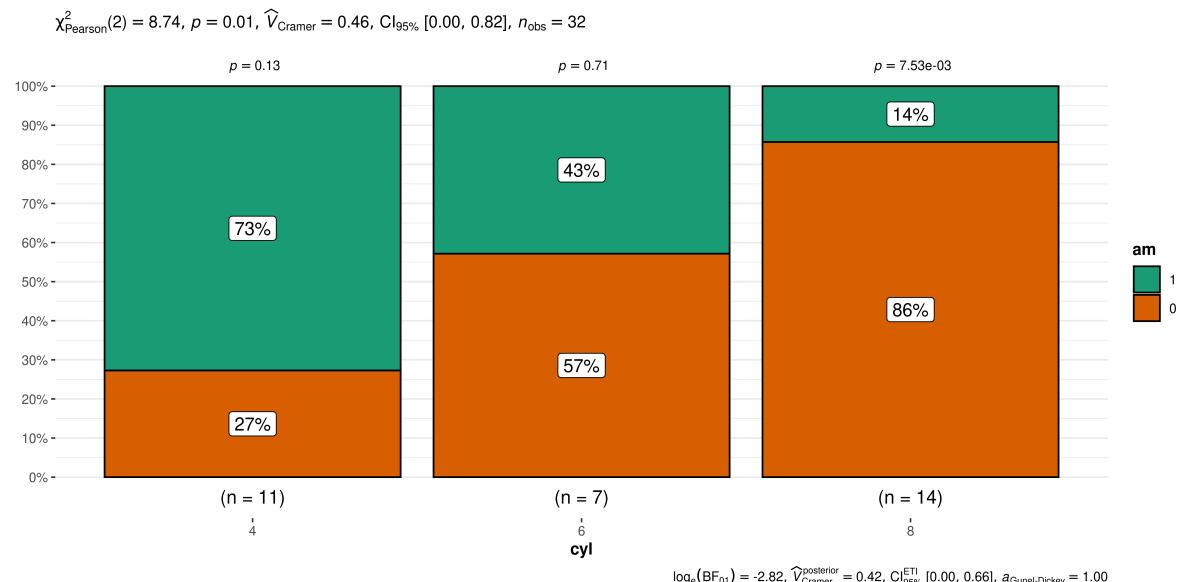
Defaults

- descriptive statistics
- inferential statistics
- effect size + uncertainty
- goodness-of-fit tests
- Bayesian hypothesis-testing
- Bayesian estimation

ggbarstats ()

Hypothesis about composition of categorical variables

```
1 ggbarstats(  
2   data = mtcars,  
3   x = am,  
4   y = cyl  
5 )
```



Important

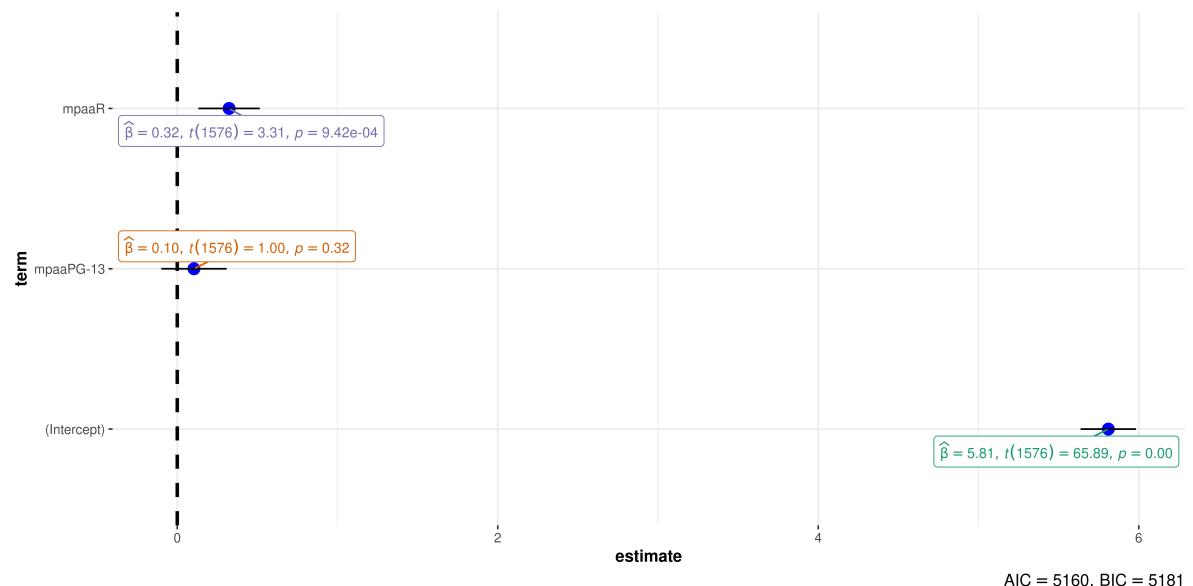
Defaults

- descriptive statistics
 - inferential statistics
 - effect size + uncertainty
 - goodness-of-fit tests
 - Bayesian hypothesis-testing
 - Bayesian estimation

ggcoefstats()

Hypothesis about regression coefficients

```
1 mod <- lm(  
2   formula = rating ~ mpaa,  
3   data = movies_long  
4 )  
5  
6 ggcoefstats(mod)
```



! Important

Defaults

- estimate + uncertainty
- inferential statistics (t, z, F, χ^2)
- model fit indices (AIC + BIC)

Supports all regression models supported in `{easystats}` ecosystem.

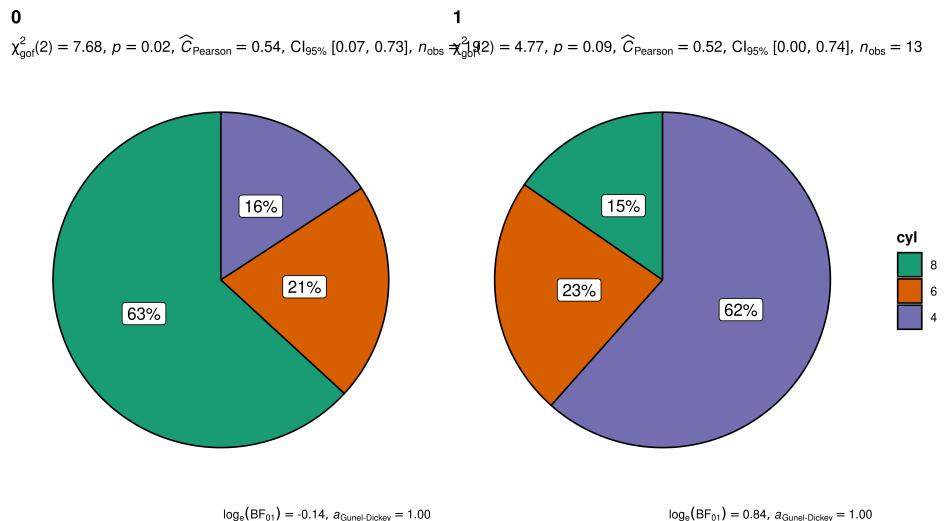
Meta-analysis is also supported!

grouped_ variants of all functions

Iterating over a grouping variable

grouped_ functions

```
1 grouped_ggpiestats(  
2   data = mtcars,  
3   x = cyl,  
4   grouping.var = am  
5 )
```



Available [grouped_](#) variants:

- [grouped_ggbetweenstats\(\)](#)
- [grouped_ggwithinstats\(\)](#)
- [grouped_gghistostats\(\)](#)
- [grouped_ggdotplotstats\(\)](#)
- [grouped_ggscatterstats\(\)](#)
- [grouped_ggcorrmat\(\)](#)
- [grouped_ggpiestats\(\)](#)
- [grouped_ggbarstats\(\)](#)

More `{ggstatsplot}` benefits

Supports different statistical approaches

Note

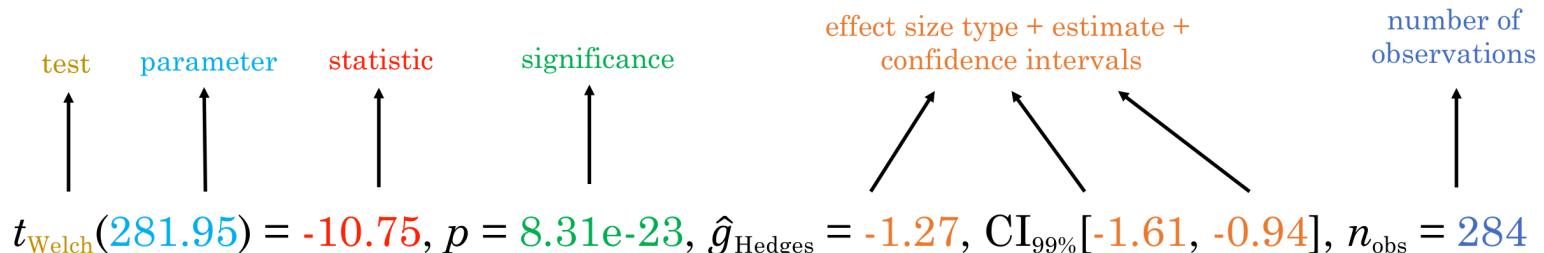
Functions	Description	Parametric	Non-parametric	Robust	Bayesian
<code>ggbetweenstats()</code>	Between group comparisons	✓	✓	✓	✓
<code>ggwithinstats()</code>	Within group comparisons	✓	✓	✓	✓
<code>gghistostats()</code> , <code>gddotplotstats()</code>	Distribution of a numeric variable	✓	✓	✓	✓
<code>ggcorrrmat()</code>	Correlation matrix	✓	✓	✓	✓
<code>ggscatterstats()</code>	Correlation between two variables	✓	✓	✓	✓
<code>ggpiestats()</code> , <code>ggbarstats()</code>	Association between categorical variables	✓	NA	NA	✓
<code>ggpiestats()</code> , <code>ggbarstats()</code>	Equal proportions for categorical variable levels	✓	NA	NA	✓
<code>ggcoefstats()</code>	Regression modeling	✓	✓	✓	✓
<code>ggcoefstats()</code>	Random-effects meta-analysis	✓	NA	✓	✓

Best practices in statistical reporting

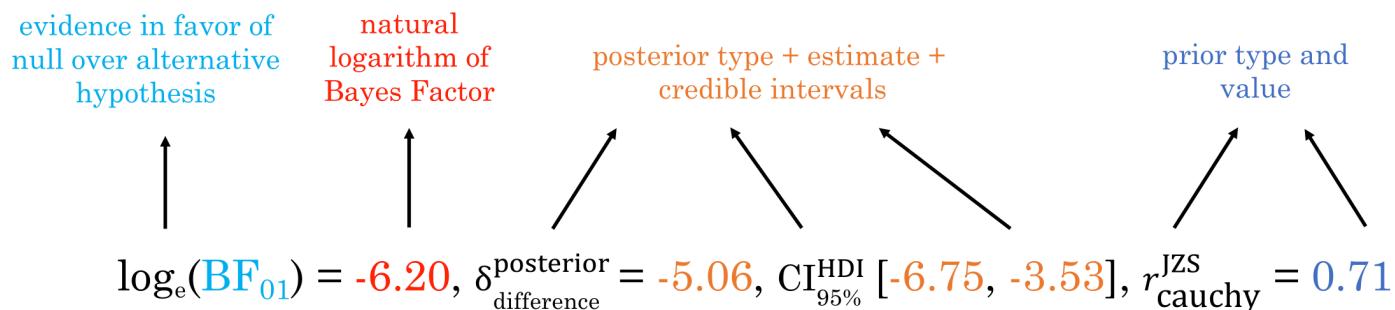


Results from Welch's t-test with {statsExpressions}

Template for Frequentist analysis



Template for Bayesian analysis



<<https://indrajeetpatil.github.io/statsExpressions/>>

src: @patilindrajeets

Avoiding reporting errors

“half of all published psychology papers that use NHST contained at least one p -value that was inconsistent with its test statistic and degrees of freedom. One in eight papers contained a grossly inconsistent p -value that may have affected the statistical conclusion”

(Nuijten et al., *Behavior Research Methods*, 2016)

Since the plot and the statistical analysis are yoked together, the chances of making an error in reporting the results are minimized.

No need to worry about updating figures and statistical details **separately.** 

Making sense of null results

$p > 0.05$: The null hypothesis (H_0) can't be rejected

But can it be **accepted**?! Null Hypothesis Significance Testing 😬

“In 72% of cases, nonsignificant results were misinterpreted, in that the authors inferred that the effect was absent. A Bayesian reanalysis revealed that fewer than 5% of the nonsignificant findings provided strong evidence (i.e., $BF_{01} > 10$) in favor of the null hypothesis over the alternative hypothesis.”

(Aczel et al., AMPPS, 2018)

Juxtaposing frequentist and Bayesian statistics for the same analysis helps to properly interpret the null results.

A few other benefits

Minimal code needed (`data`, `x`, `y`): minimizes chances of error + tidy scripts. 🎉

Disembodied figures stand on their own and are easy to evaluate. 😊

More breathing room for theoretical discussion and other text. ✍️

Misconceptions: This package is...

-  an alternative to learning `ggplot2`
-  the more you know `ggplot2`, the better you can modify the defaults to your liking)

-  meant to be used in talks/presentations
-  defaults too complicated for effectively communicating results in time-constrained presentation settings, e.g. conference talks)

-  only relevant when used in publications
-  not necessary; can also be useful *only* during exploratory phase

-  the only game in town
-  excellent GUI open-source software: `JASP` and `jamovi`)