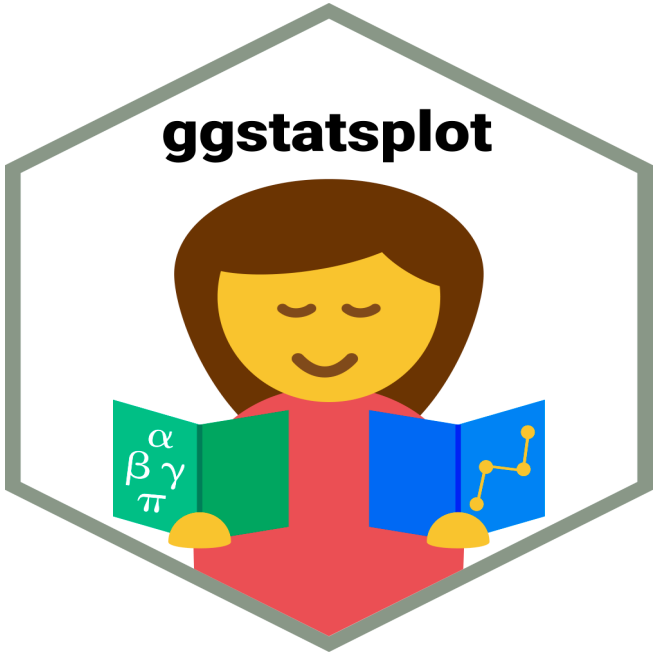
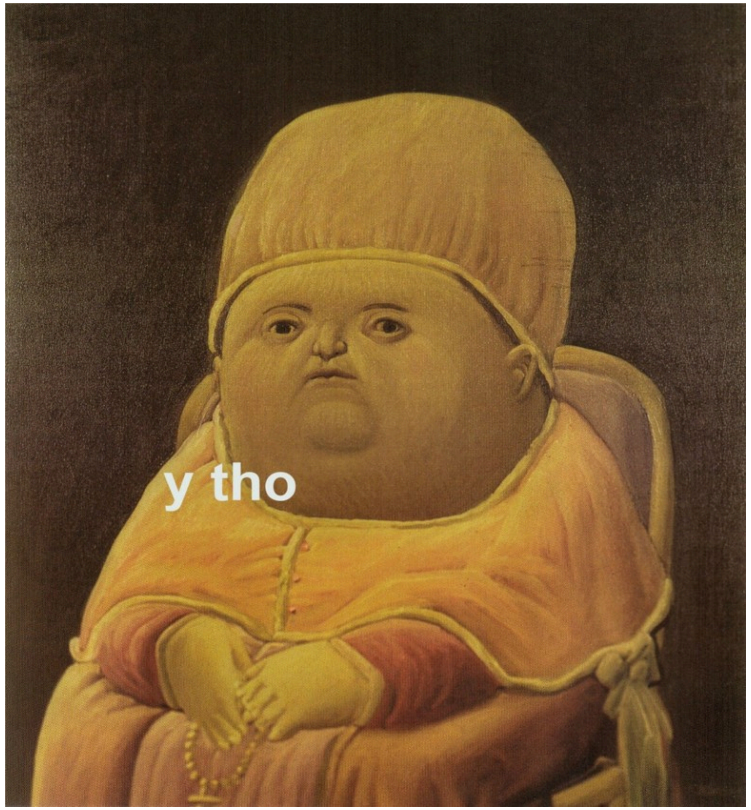
{ggstatsplot}: Informative Statistical Visualizations

Indrajeet Patil



Why another package?

Current CRAN package count **>23,000**

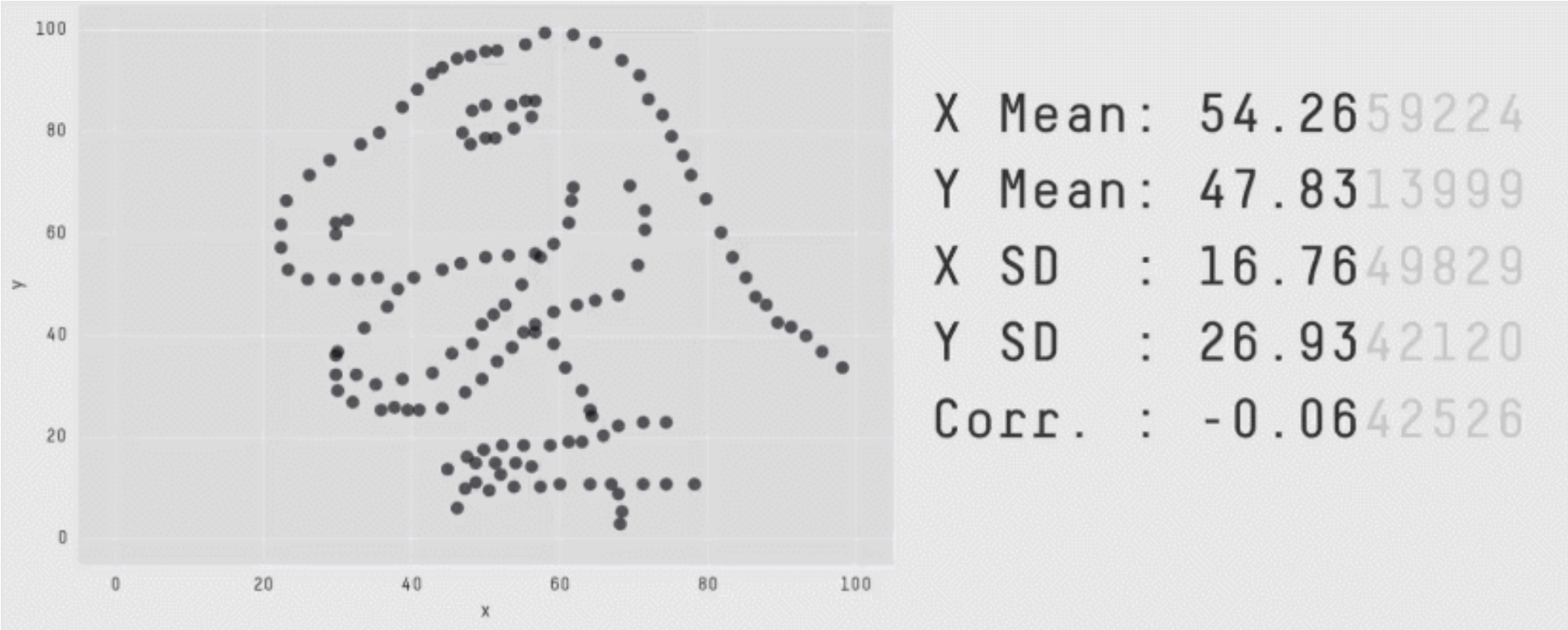


`{ggstatsplot}` provides-

📊 information-rich plots with statistical details

📝 suitable for faster (exploratory) data analysis and reporting

Informative graphic = a thousand words

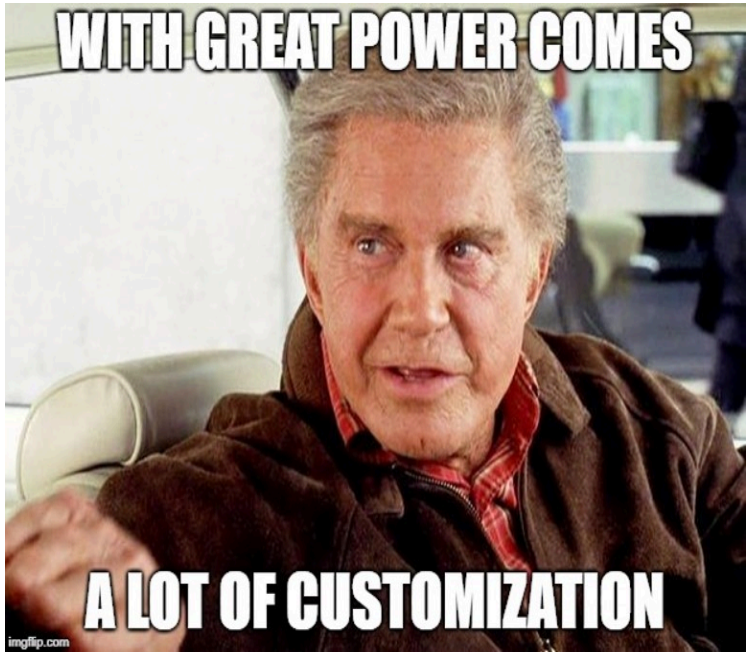


Graphical summaries can reveal problems not visible from **numerical** statistics.

(Matejka & Fitzmaurice, *Autodesk Research*,2017)

Ready-made plot = no customization

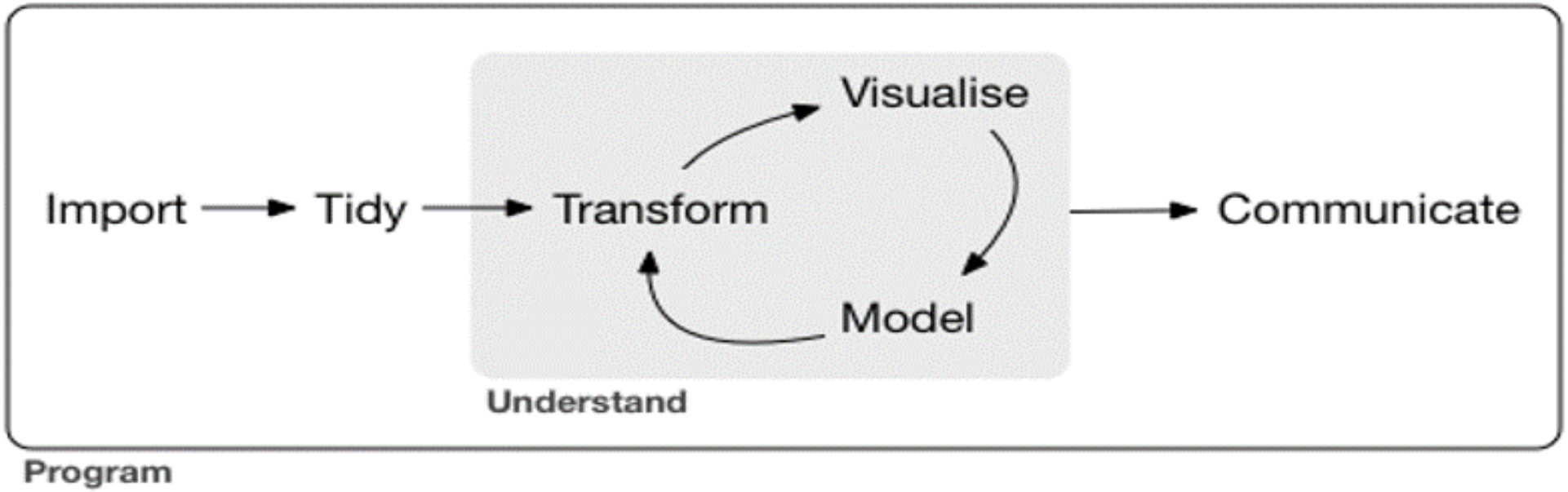
The **grammar of graphics** is a powerful framework (Wilkinson, 2011) and can help you make *any* graphics fitting your specific data visualization needs! But...



Quality of Life (QoL) improvements with `{ggstatsplot}`

Provide ready-made plots with defaults following the best practices in statistical reporting and data visualization.

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!

(Grolemund & Wickham, *R for Data Science*, 2017)

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:

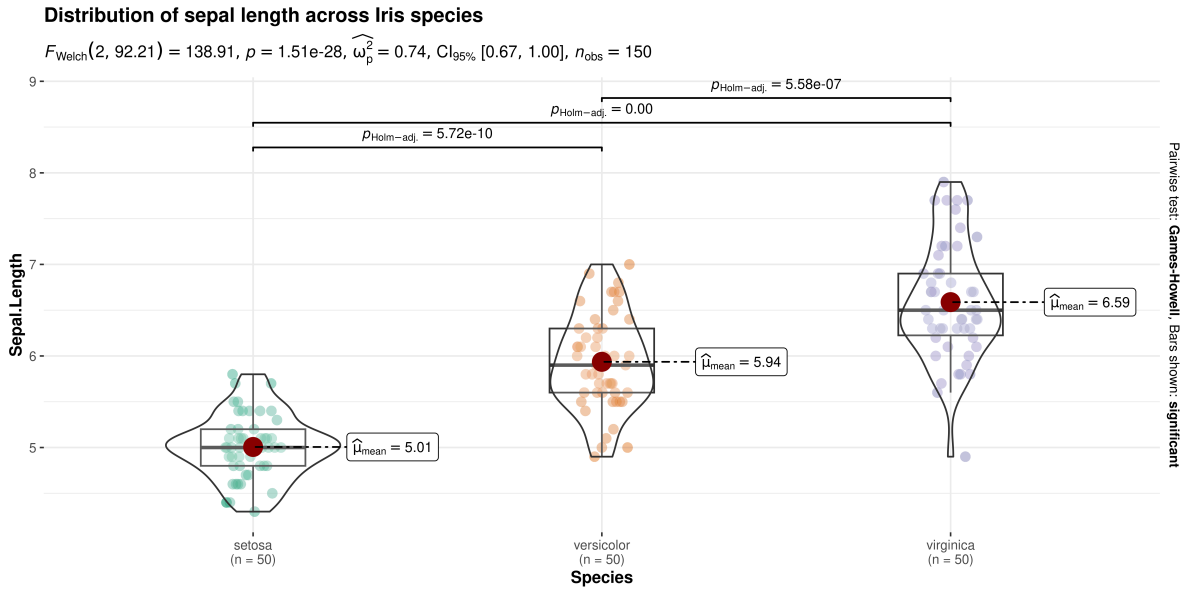
Type	Command
Release	`install.packages("ggstatsplot")`
Development	`pak::pak("IndrajeetPatil/ggstatsplot")`

Example function

`ggbetweenstats()`

For between-group comparisons

```
ggbetweenstats(
  data  = iris,
  x     = Species,
  y     = Sepal.Length,
  title = "Distribution of sepal length across Iris species"
)
```



Important

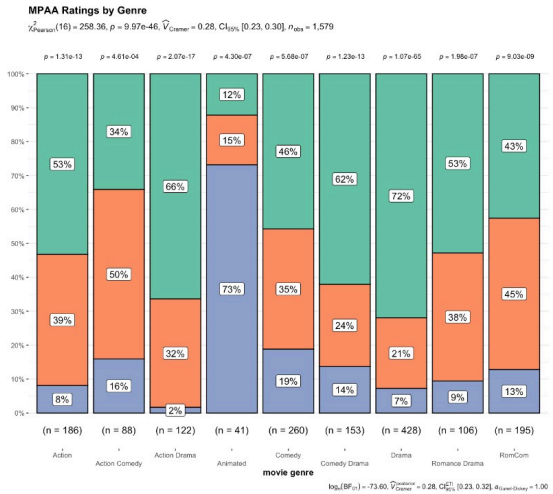
Defaults

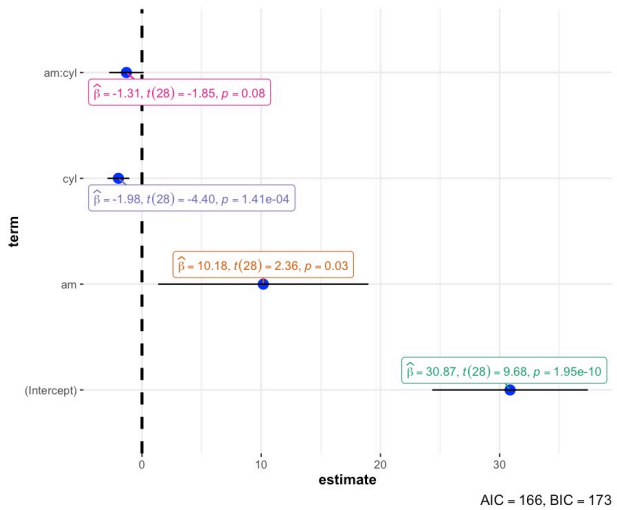
- raw data + distributions
- descriptive statistics
- inferential statistics
- effect size + uncertainty
- pairwise comparisons
- Bayesian hypothesis-testing
- Bayesian estimation

Statistical approaches available

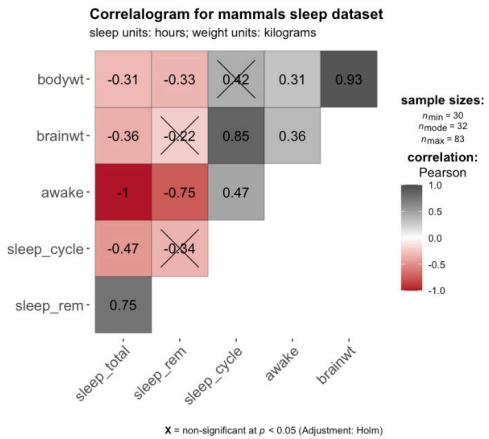
- parametric
- parametric
- robust
- Bayesian

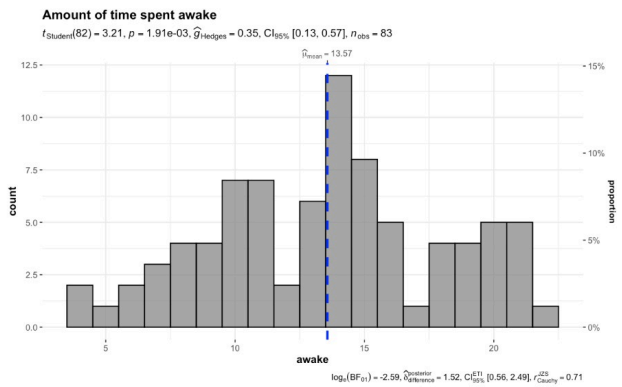
Other functions

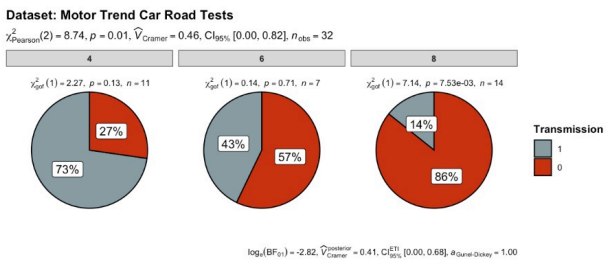


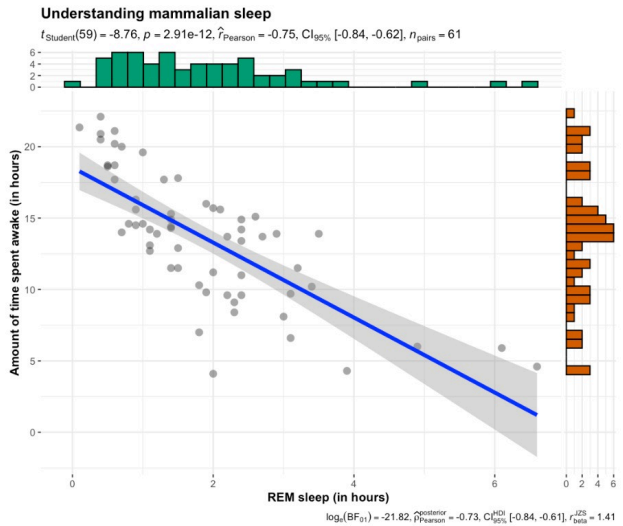


AIC = 166, BIC = 173









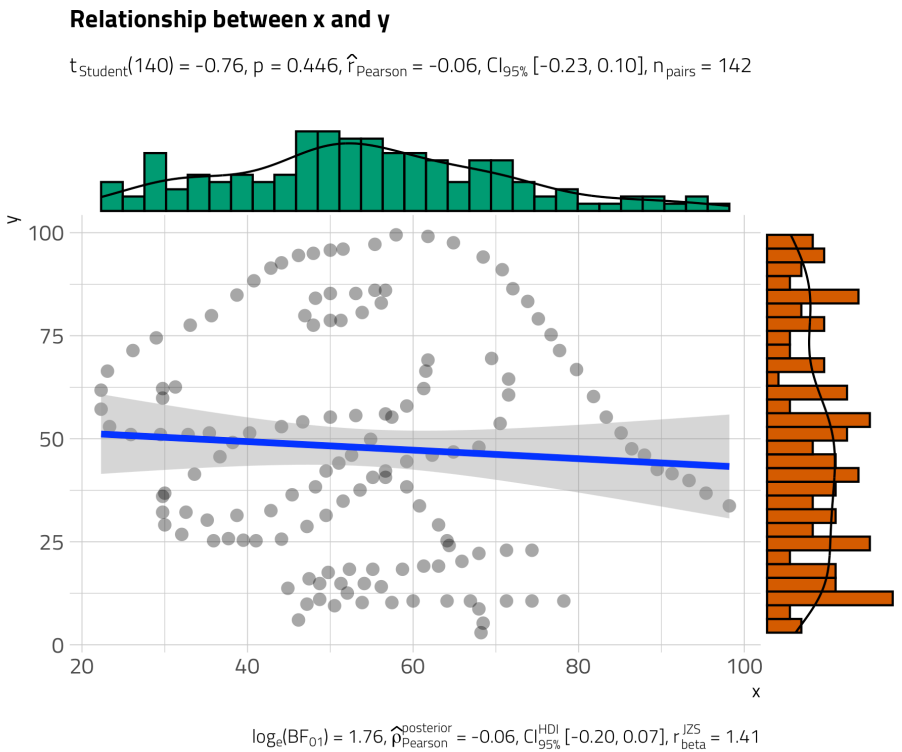
Benefits for Statistical Reporting

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



Toggling statistical approaches 🔀

Parametric

```
# anova
ggbetweenstats(
  data = mtcars,
  x = cyl,
  y = wt,
  type = "p"
)

# correlation analysis
ggscatterstats(
  data = mtcars,
  x = wt,
  y = mpg,
  type = "p"
)

# t-test
gghistostats(
  data = mtcars,
  x = wt,
  test.value = 2,
  type = "p"
)
```

Non-parametric

```
# anova
ggbetweenstats(
  data = mtcars,
  x = cyl,
  y = wt,
  type = "np"
)

# correlation analysis
ggscatterstats(
  data = mtcars,
  x = wt,
  y = mpg,
  type = "np"
)

# t-test
gghistostats(
  data = mtcars,
  x = wt,
  test.value = 2,
  type = "np"
)
```

Alternative: Pure Pain

🚧 Hunting for packages

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

🚧 Inconsistent APIs

- 🤔 accepts data frame, vector, matrix?
- 🤔 long/wide format data?
- 🤔 works with `NA`s?
- 🤔 returns data frame, vector, matrix?
- 🤔 works with tibbles?
- 🤔 has all necessary details?
- 🤔 ...

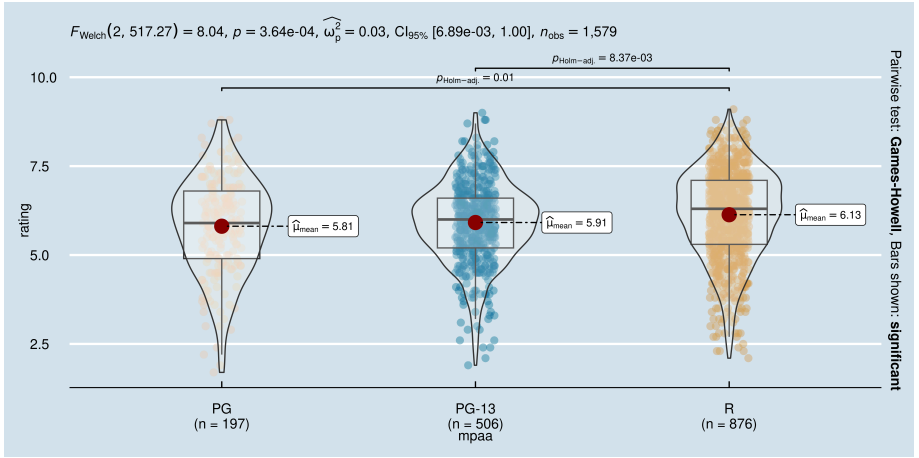
Customizability

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

Modify the look 🎨

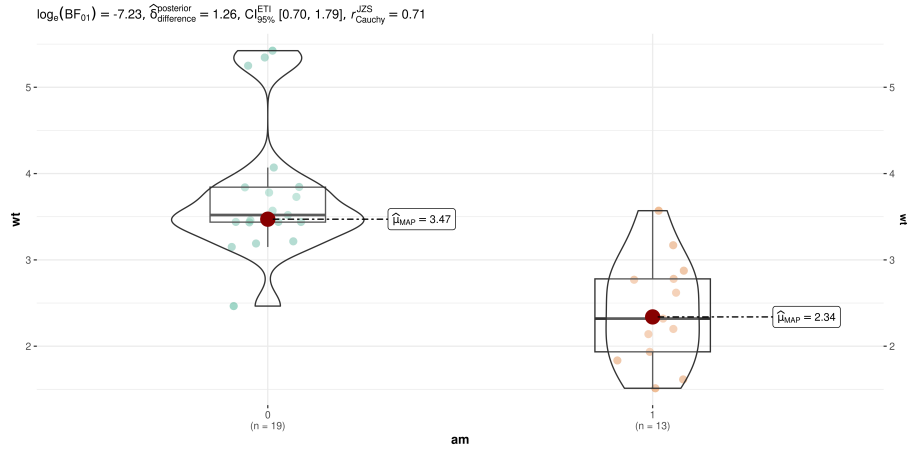
By changing **theme** and **palette**

```
ggbetweenstats(
  data = movies_long,
  x = mpaa,
  y = rating,
  ggtheme = ggthemes::theme_economist(),
  palette = "Darjeeling2",
  package = "wesanderson"
)
```



By using {ggplot2} functions

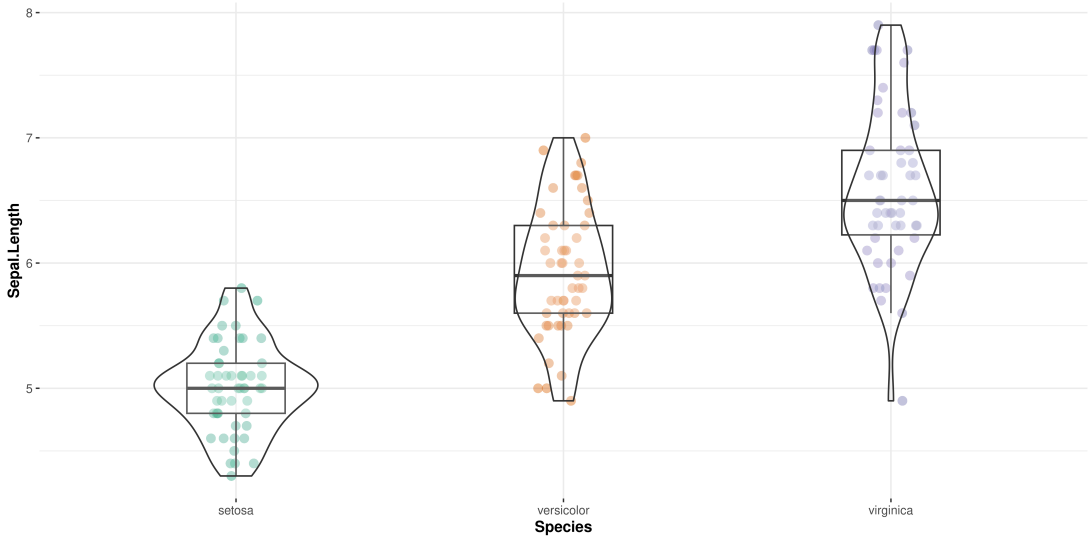
```
ggbetweenstats(
  data = mtcars,
  x = am,
  y = wt,
  type = "bayes"
) +
  scale_y_continuous(sec.axis = dup_axis())
```



Too much information 🙈

Get **only plots:**

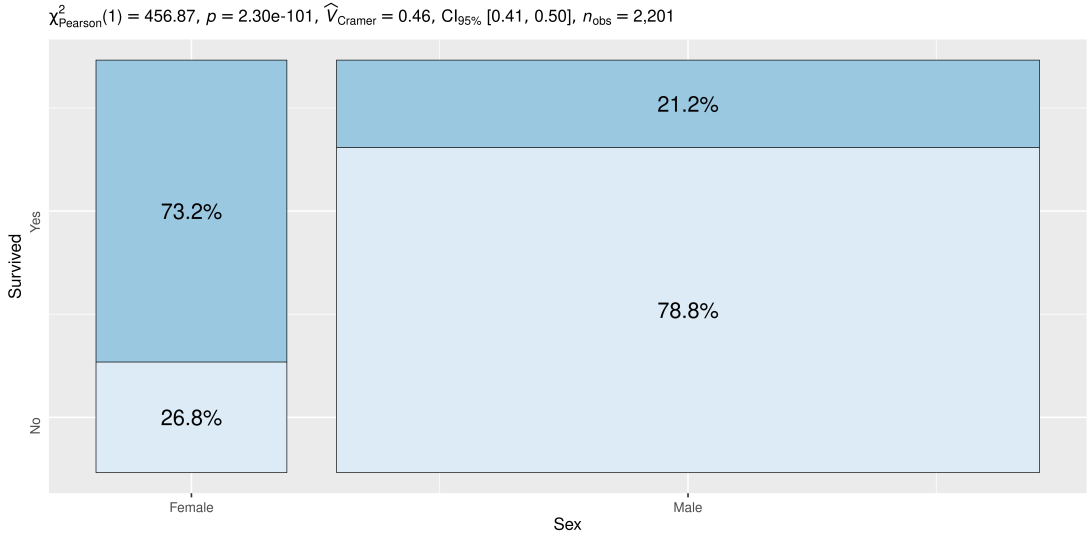
```
ggbetweenstats(
  data = iris,
  x = Species,
  y = Sepal.Length,
  # turn off statistical analysis
  centrality.plotting = FALSE,
  results.subtitle = FALSE,
  bf.message = FALSE,
  # turn off pairwise comparisons
  pairwise.display = "none"
)
```



Get **only expressions:**

```
stats_expr <- ggpiestats(
  Titanic_full, Survived, Sex,
) %>%
  extract_subtitle()

ggiraphExtra::ggSpine(
  data = Titanic_full,
  aes(x = Sex, fill = Survived)
) +
  labs(subtitle = stats_expr)
```



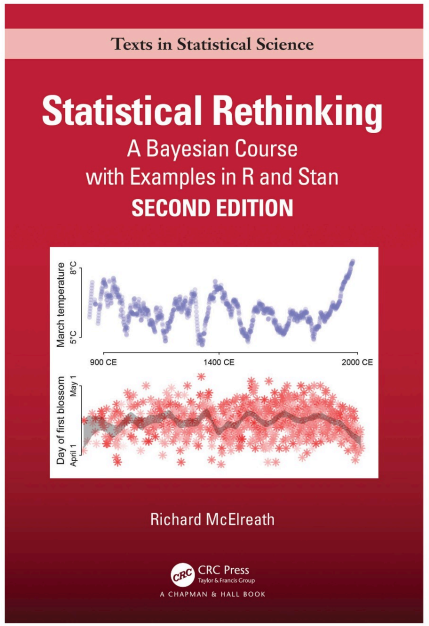
Critical Evaluation

Things to be wary of

The "Golem of Prague" problem

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.
- Statistical proficiency needed.

Attractive Qualities

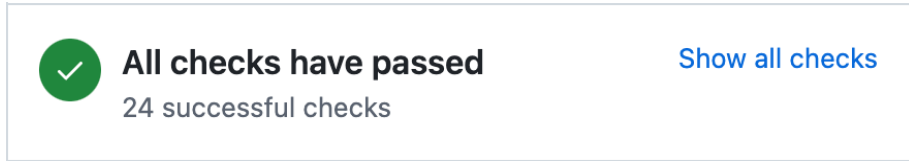
Things that will pull you in

Quality Assurance

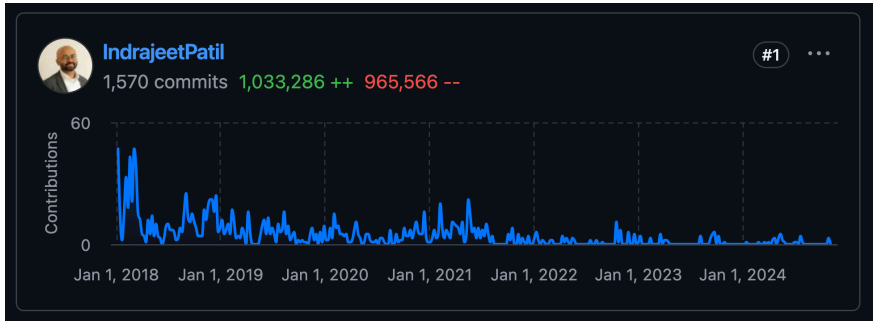
Each commit must pass many QA checks:

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)



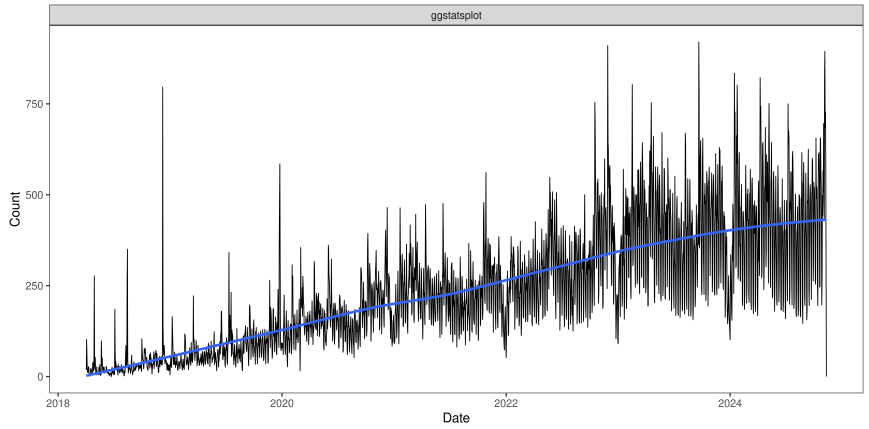
Healthy and active code base



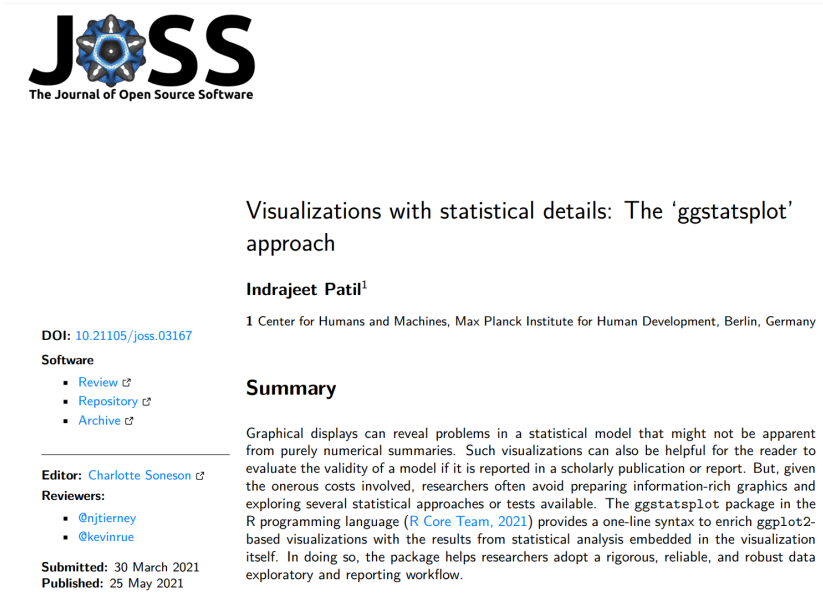
User Love

Total downloads > 500K (97 percentile)

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



Total citations > 1000



JOSS
The Journal of Open Source Software

Visualizations with statistical details: The 'ggstatsplot' approach

Indrajeet Patil[1]

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

DOI: 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.

Visualizations with statistical details: The 'ggstatsplot' approach
I Patil
Journal of Open Source Software 6 (61), 3167, 2021

1277
*

Conclusion

Benefits of the `{ggstatsplot}` approach

`{ggstatsplot}` combines data visualization and statistical analysis in a single step.

It...

- provides ready-made plots with information-rich defaults
- minimizes the chances of making errors in statistical reporting
- follows best practices in data visualization and statistical reporting
- helps evaluate statistical analysis in the context of the underlying data
- highlights the importance of the effect by providing effect size measures
- provides an easy way to evaluate *absence* of an effect using Bayesian framework
- extremely beginner-friendly

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
- LikedIn
- GitHub
- Website
- E-mail

Thank You 

Session information

```
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-az1117-453
 ui       X11
 language (EN)
 collate  C.UTF-8
 ctype    C.UTF-8
 tz       UTC
 date     2024-11-11
 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
```

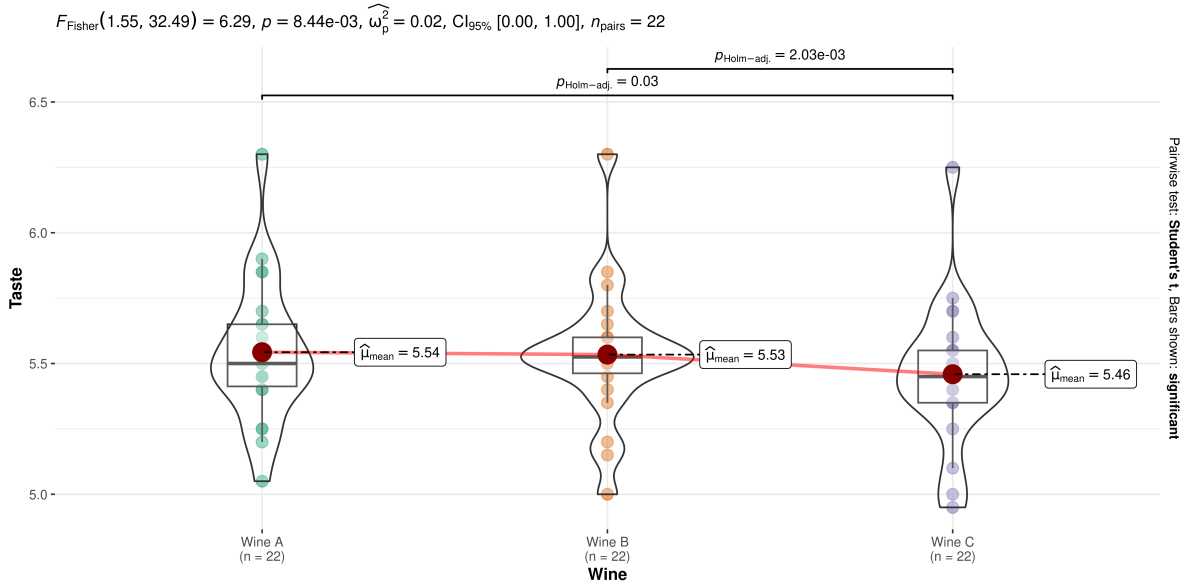
Appendix

Examples of other functions

`ggwithinstats()`

Hypothesis about group differences: repeated measures design

```
ggwithinstats(
  data = WRS2::WineTasting,
  x = Wine,
  y = Taste
)
```



Important

✏️ Defaults

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

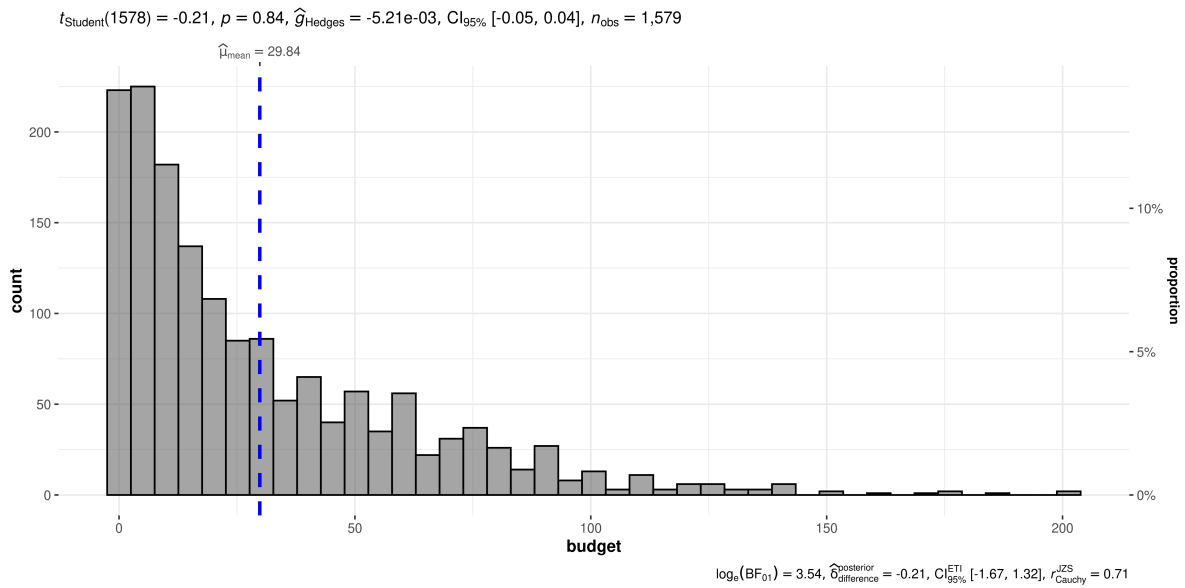
Statistical approaches available

- parametric
- parametric
- robust
- Bayesian

`gghistostats()`

Distribution of a numeric variable

```
gghistostats(
  data = movies_long,
  x = budget,
  test.value = 30
)
```



Important

✏️ **Defaults**

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

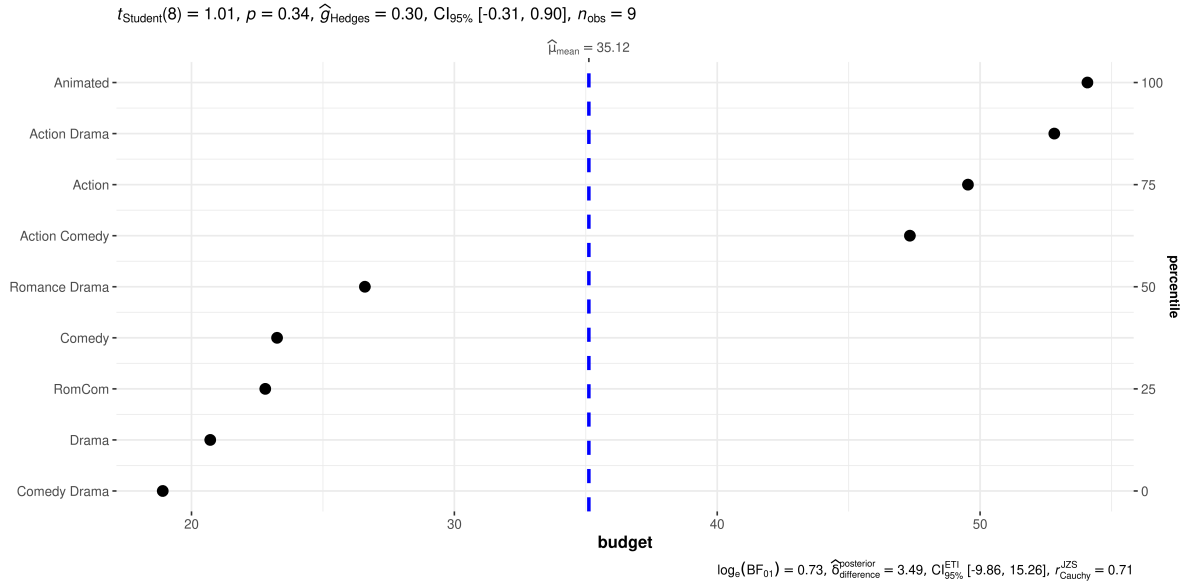
Statistical approaches available

- parametric
- parametric
- robust
- Bayesian

`ggdotplotstats()`

Labeled numeric variable

```
ggdotplotstats(
  data = movies_long,
  x = budget,
  y = genre,
  test.value = 30
)
```



Important

✏️ **Defaults**

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

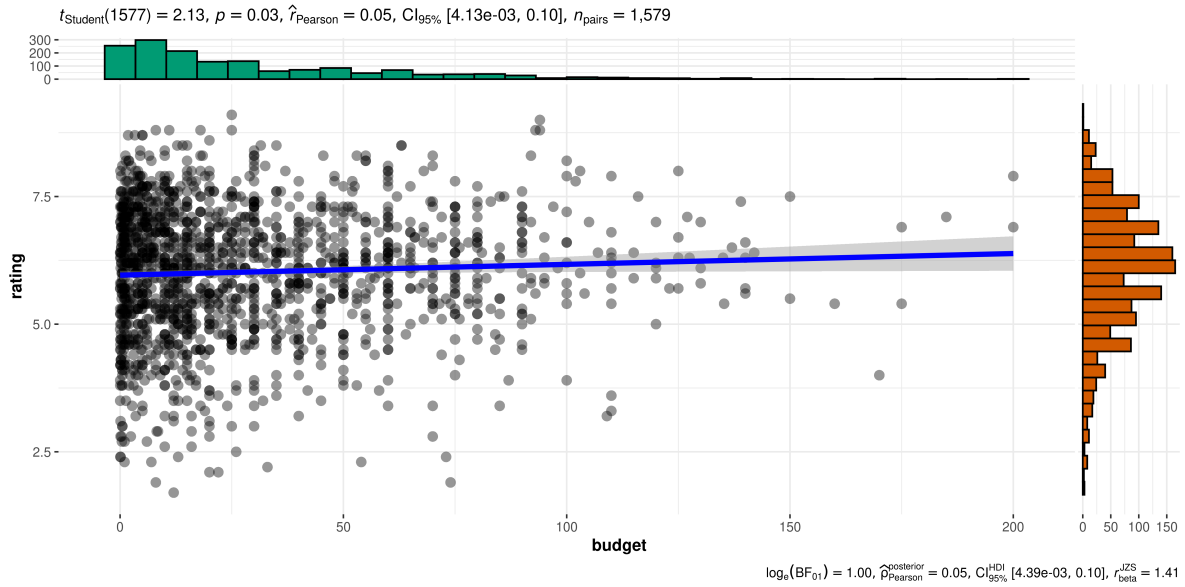
Statistical approaches available

- parametric
- parametric
- robust
- Bayesian

`ggscatterstats()`

Hypothesis about correlation: Two numeric variables

```
ggscatterstats(
  data = movies_long,
  x = budget,
  y = rating
)
```



Important

✏️ Defaults

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

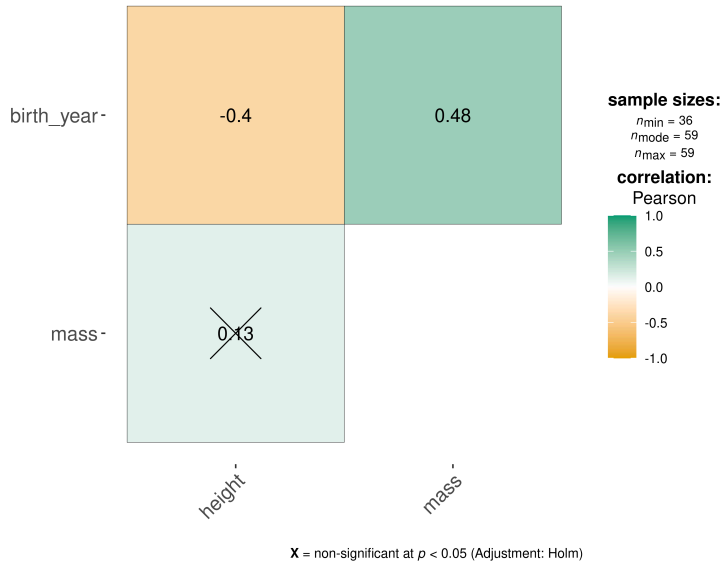
Statistical approaches available

- 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 `NA`s
- partial correlations

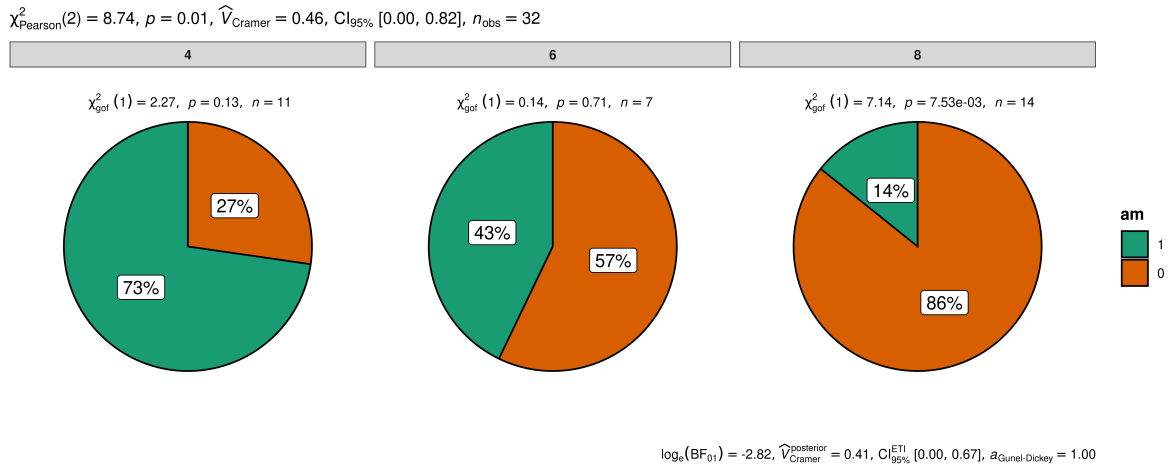
Statistical approaches available

- parametric
- parametric
- robust
- Bayesian

`ggpiestats()`

Hypothesis about composition of categorical variables

```
ggpiestats(
  data = mtcars,
  x = am,
  y = cyl
)
```



Important

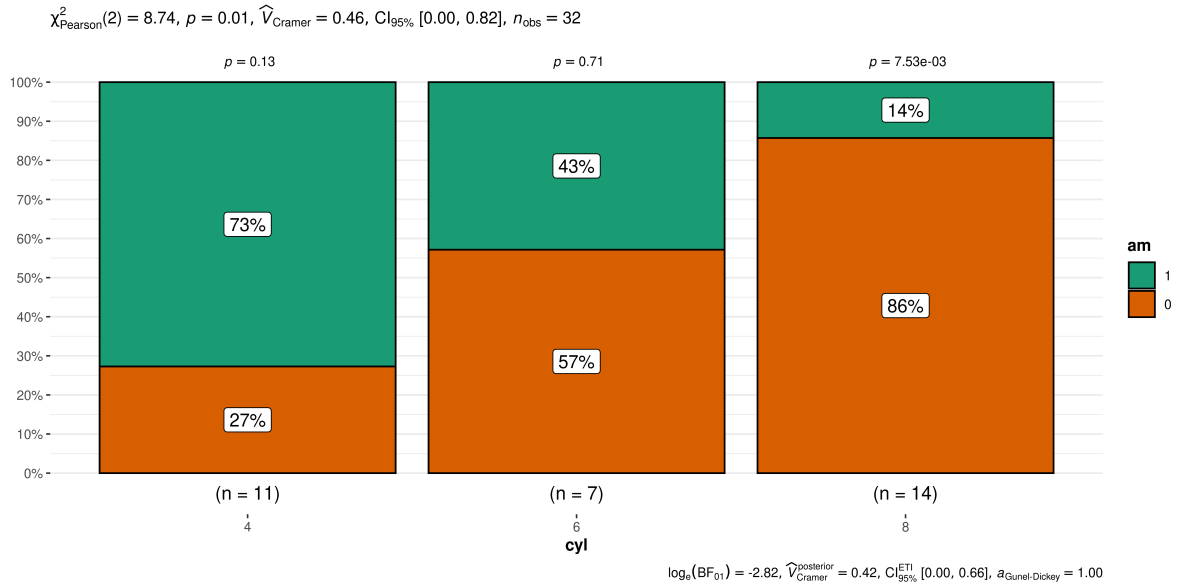
✏️ Defaults

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

`ggbarstats()`

Hypothesis about composition of categorical variables

```
ggbarstats(
  data = mtcars,
  x = am,
  y = cyl
)
```



Important

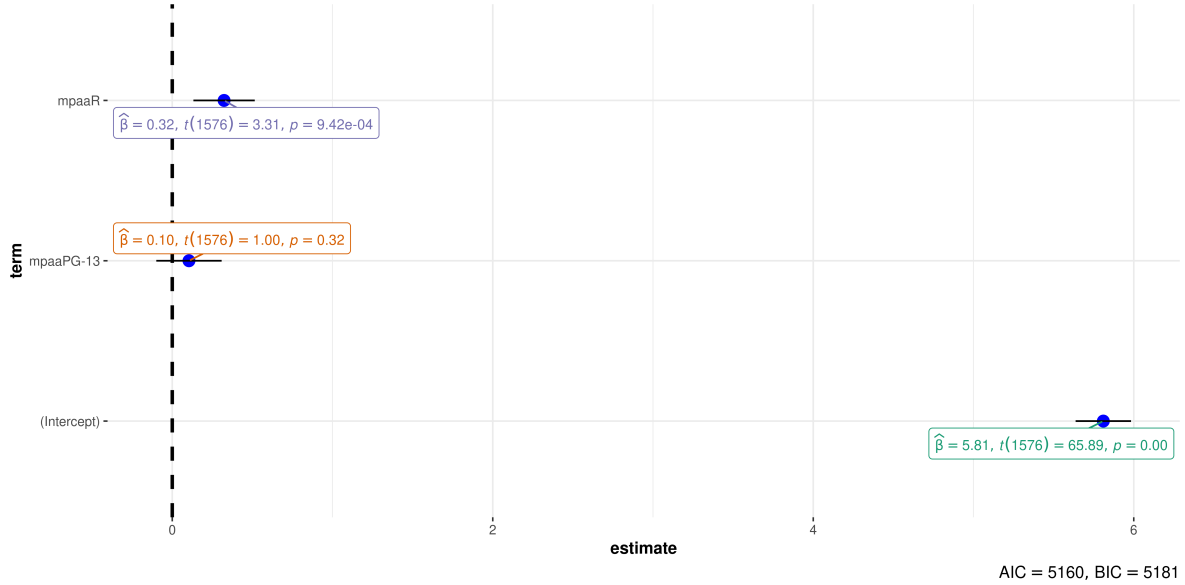
Defaults

- 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.

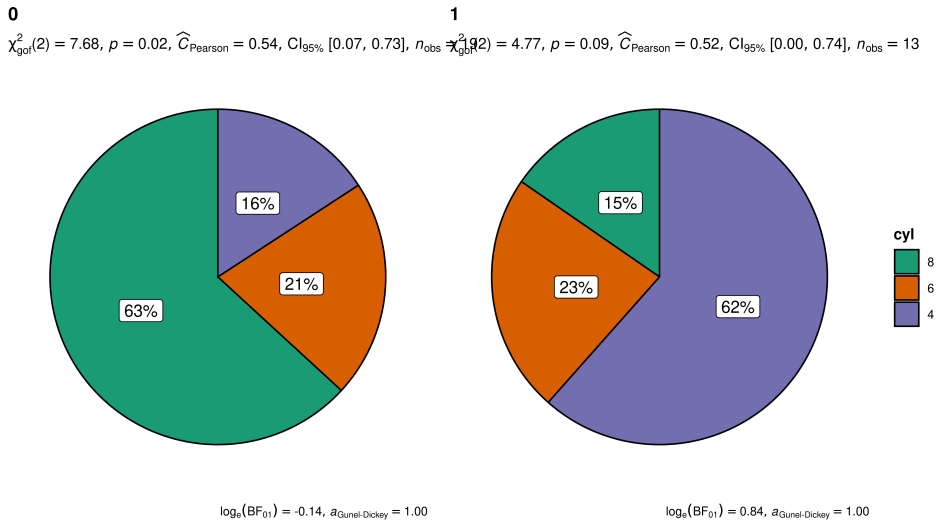
Meta-analysis is also supported!

*grouped_* variants

Iterating over a grouping variable

*grouped_* functions

```
grouped_ggpiestats(
  data = mtcars,
  x = cyl,
  grouping.var = am
)
```



Available `grouped_` variants:

- `grouped_ggbetweenstats()`
- `grouped_ggwithinstats()`
- `grouped_gghistostats()`
- `grouped_ggdotplotstats()`
- `grouped_ggscatterstats()`
- `grouped_ggcorrmat()`
- `grouped_ggpiestats()`
- `grouped_ggbarstats()`

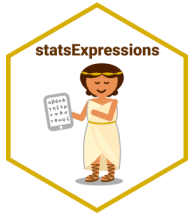
More `{ggstatsplot}` benefits

Supports different statistical approaches

Note

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

Best practices in statistical reporting 🏆



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

Template for Frequentist analysis

test | parameter | statistic | significance | effect size type + estimate + confidence intervals | number of observations

$$t_{\text{Welch}}(281.95) = -10.75,\ p = 8.31\text{e-}23,\ \hat{g}_{\text{Hedges}} = -1.27,\ \text{CI}_{99\%}[-1.61, -0.94],\ n_{\text{obs}} = 284$$

Template for Bayesian analysis

evidence in favor of null over alternative hypothesis | natural logarithm of Bayes Factor | posterior type + estimate + credible intervals | prior type and value

$$\log_e(\text{BF}_{01}) = -6.20,\ \delta^{\text{posterior}}_{\text{difference}} = -5.06,\ \text{CI}^{\text{HDI}}_{95\%}\ [-6.75, -3.53],\ r^{\text{JZS}}_{\text{cauchy}} = 0.71$$

<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 (`H0`) 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.

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)