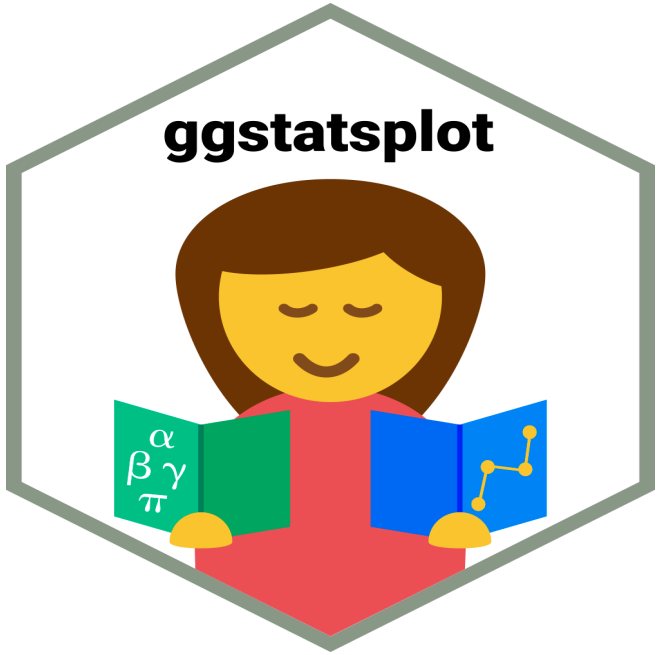
Statistical Visualizations with {ggstatsplot}: A Biography

Indrajeet Patil



Genesis

Why a new package?

Life in the trenches (c. 2017)

(as an academic researcher)

Large-scale problems

- **Reporting errors**:

> "half of all published psychology papers contained at least one *p*-value that was inconsistent"[1]

- **Interpretation errors**:

> "in 72% of cases, nonsignificant results were misinterpreted [to mean] that effect was absent"[2]

- **Replication crisis**:

> "39% of effects were subjectively rated to have replicated the original result"[3]

and more…

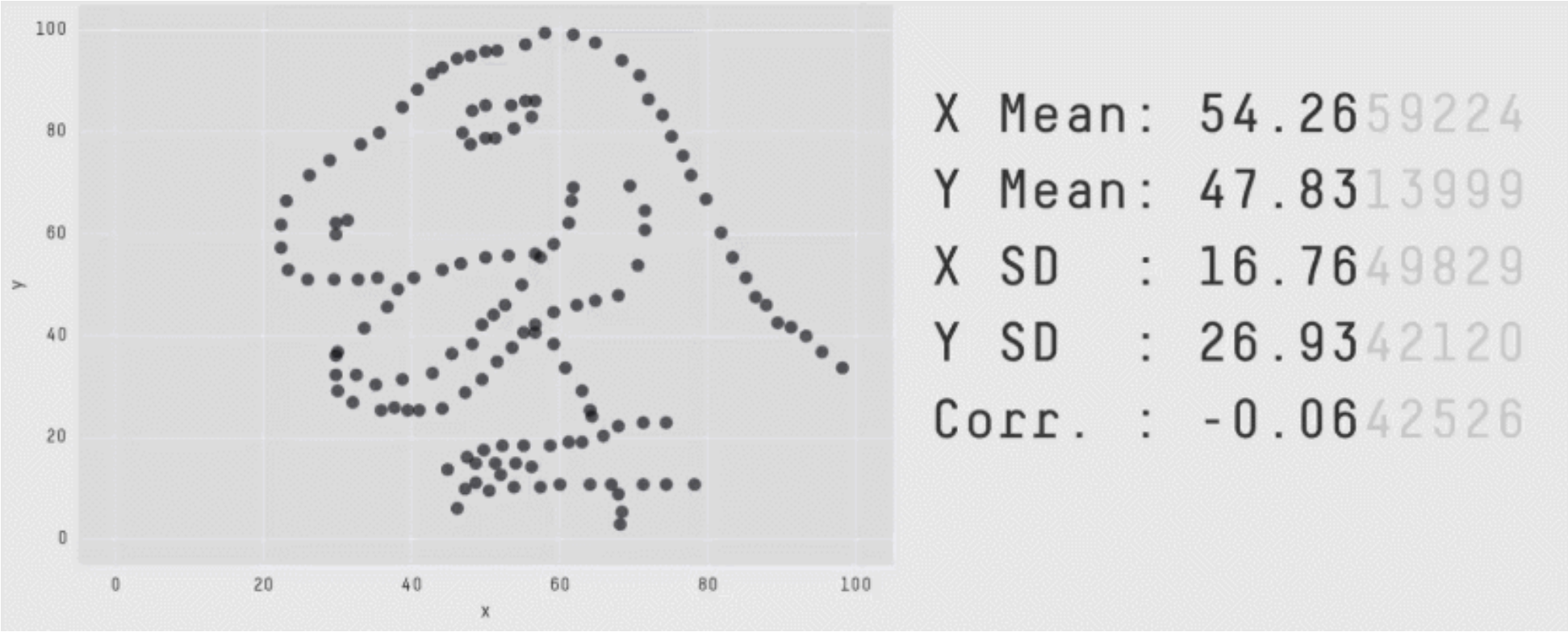
Personal challenges

- How to avoid reporting errors?
- How to make visualizing data effortless?
- How to emphasize the importance of the effect?
- How to interpret null results?
- How to easily assess validity of model assumptions?
- How to increase reproducibility?

Proposal

Information-rich, ready-made statistical visualizations

Defaulting to a rich graphic

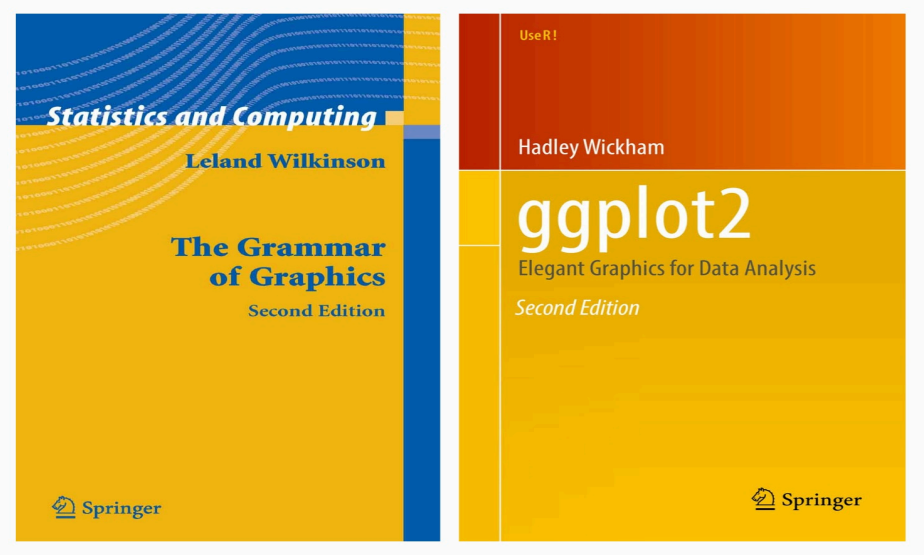


💡 **Visualizations** reveal problems not discernible from **descriptive** statistics!

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

One-line syntax for producing plots

The **grammar of graphics** framework can prepare **any** visualization! But building plots from scratch can be time-consuming.



💡 Using **ready-made plots** lowers the effort needed for visualizing data!

Action Plan

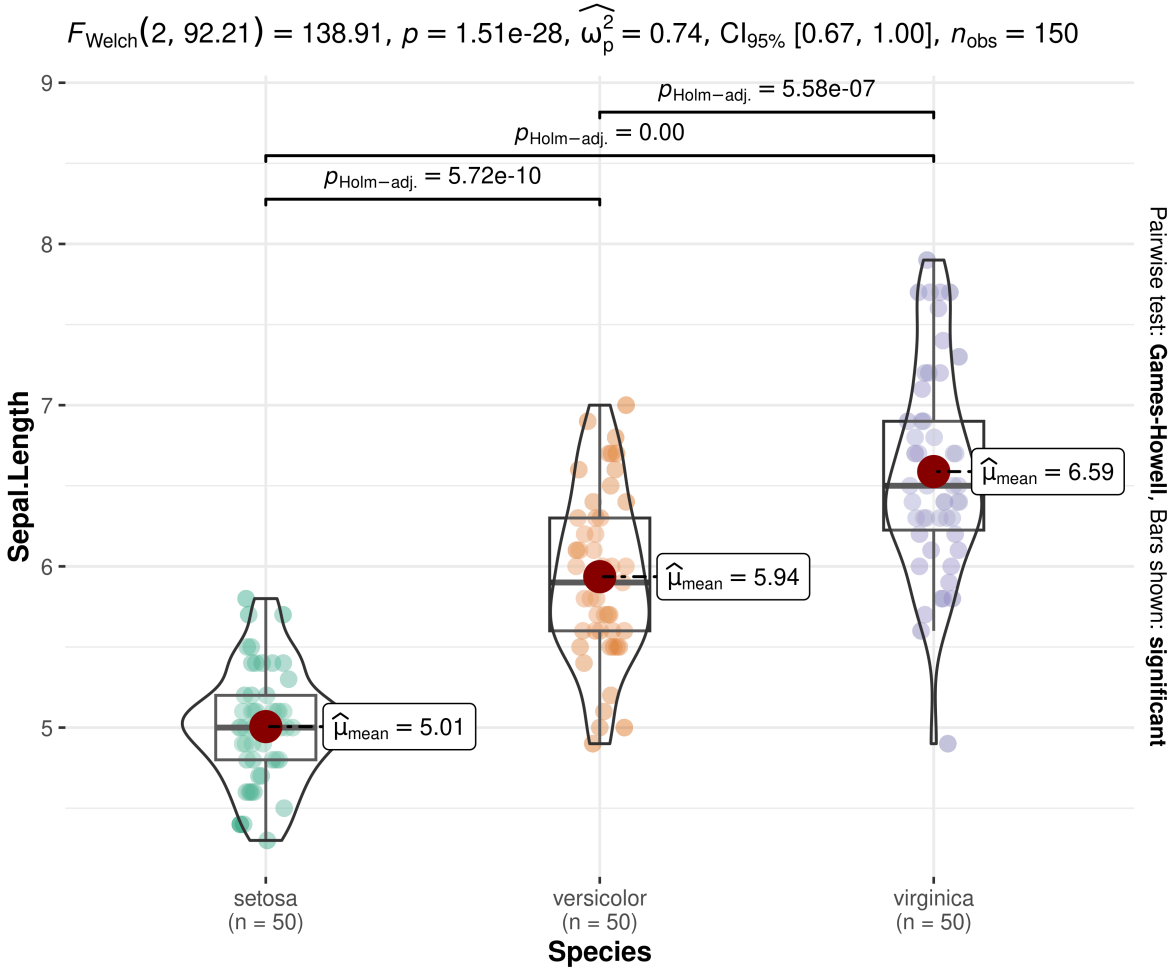
`{ggstatsplot}` was born!

(started out as loose scripts, which were later consolidated into a package)

Example function

E.g., for hypothesis about group differences

```
1  ggbetweenstats(iris, Species, Sepal.Length)
```



Important

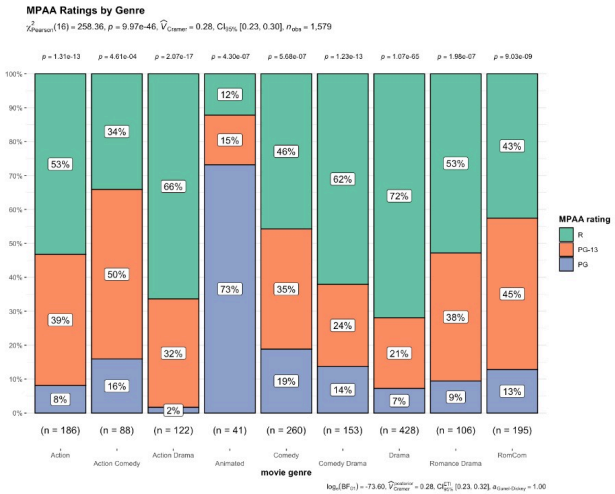
Information-rich defaults

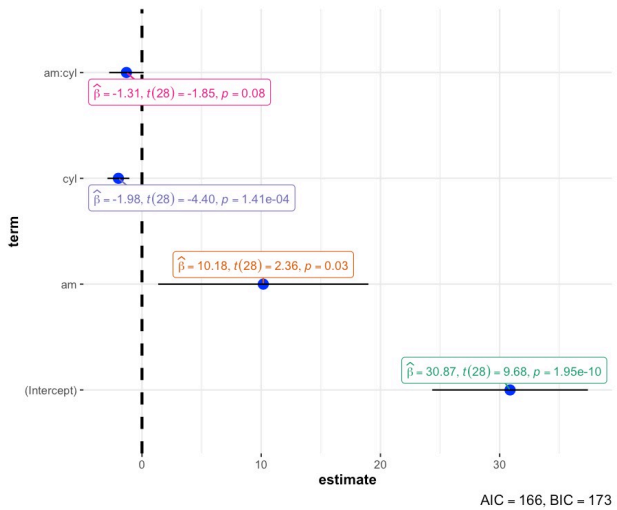
- raw data + distributions
- descriptive statistics
- inferential statistics
- effect size + uncertainty
- pairwise comparisons
- Bayesian hypothesis-testing
- Bayesian estimation

Statistical approaches available

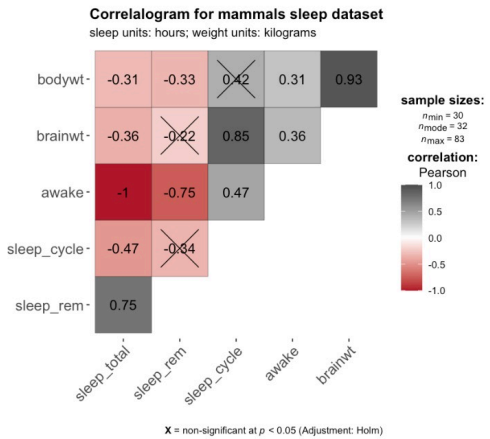
- parametric
- parametric
- robust
- Bayesian

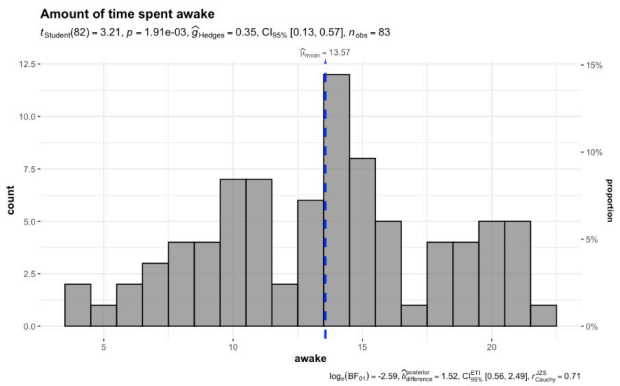
And there is more!

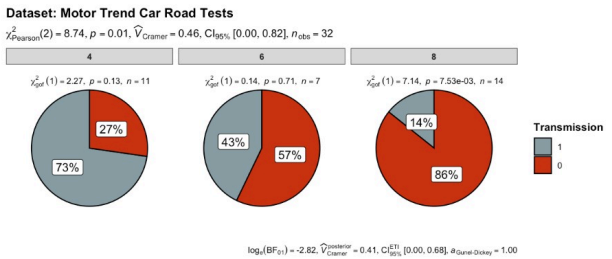


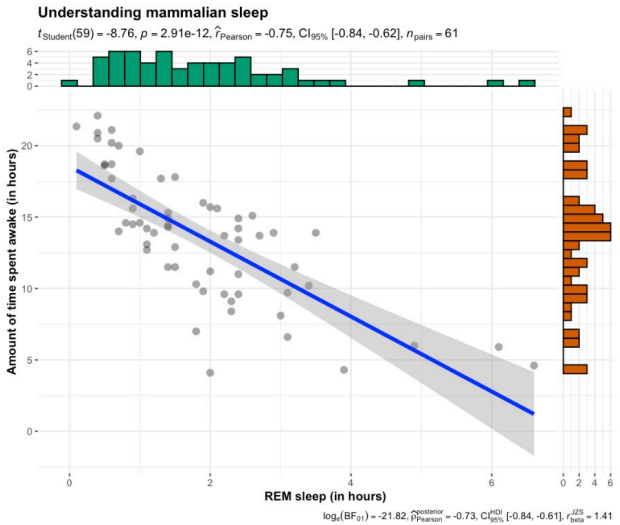


AIC = 166, BIC = 173









Appendix provides more details.

Promised Land

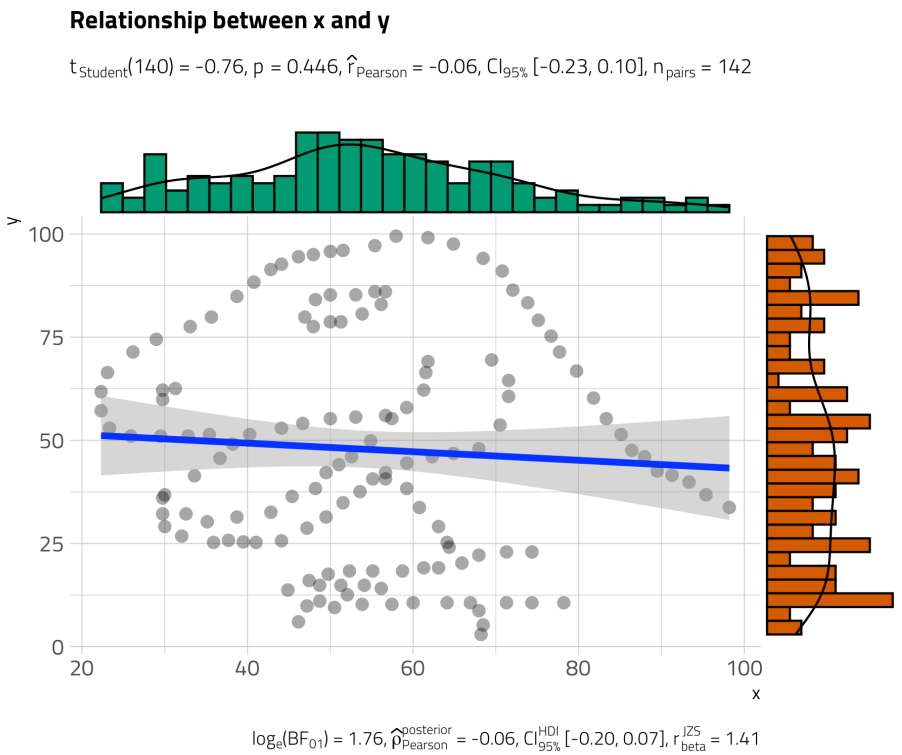
Does it deliver?

Show, don't tell

Without `{ggstatsplot}`

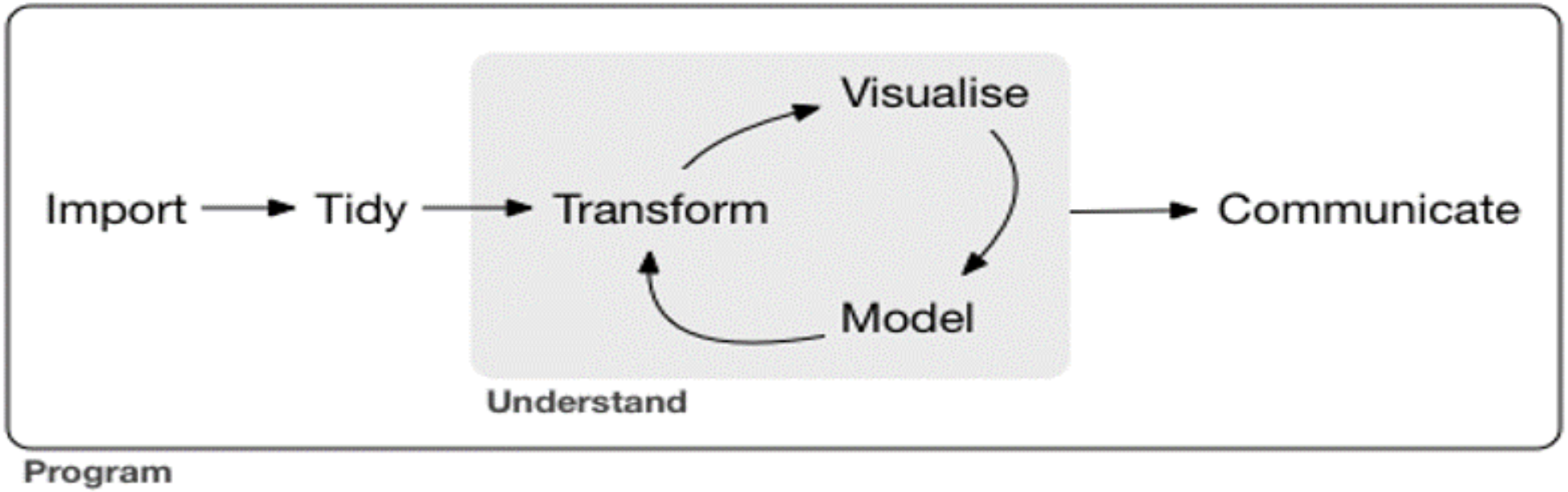
Pearson's correlation test revealed that, across 142 participants, variable `x` was negatively correlated with variable `y`: $t(140) = -0.76, p = 0.446$. The effect size $(r = -0.06, 95\%CI[-0.23, 0.10])$ was small, as per Cohen's (1988) conventions. The Bayes Factor for the same analysis revealed that 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`).

With `{ggstatsplot}`



✅ No need to worry about reporting or interpretation errors!

Simplified data analysis workflow

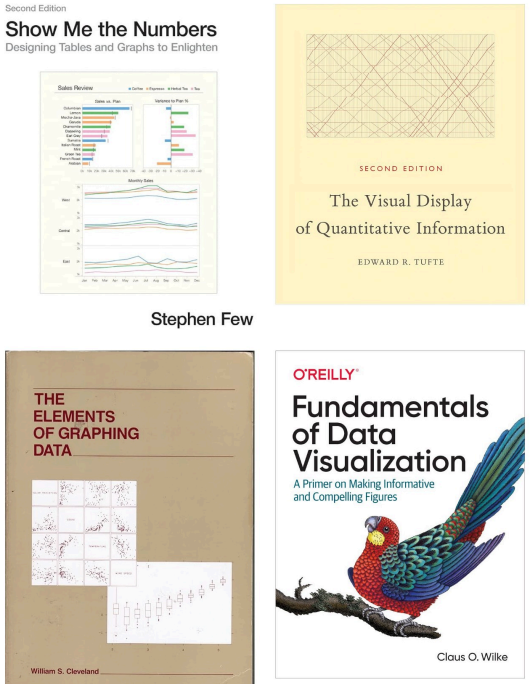


✅ Quick insight into data by combining visualization and modeling!

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

Thoughtful Defaults

Data Visualization



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

(Doorn et al., 2020; APA Manual)

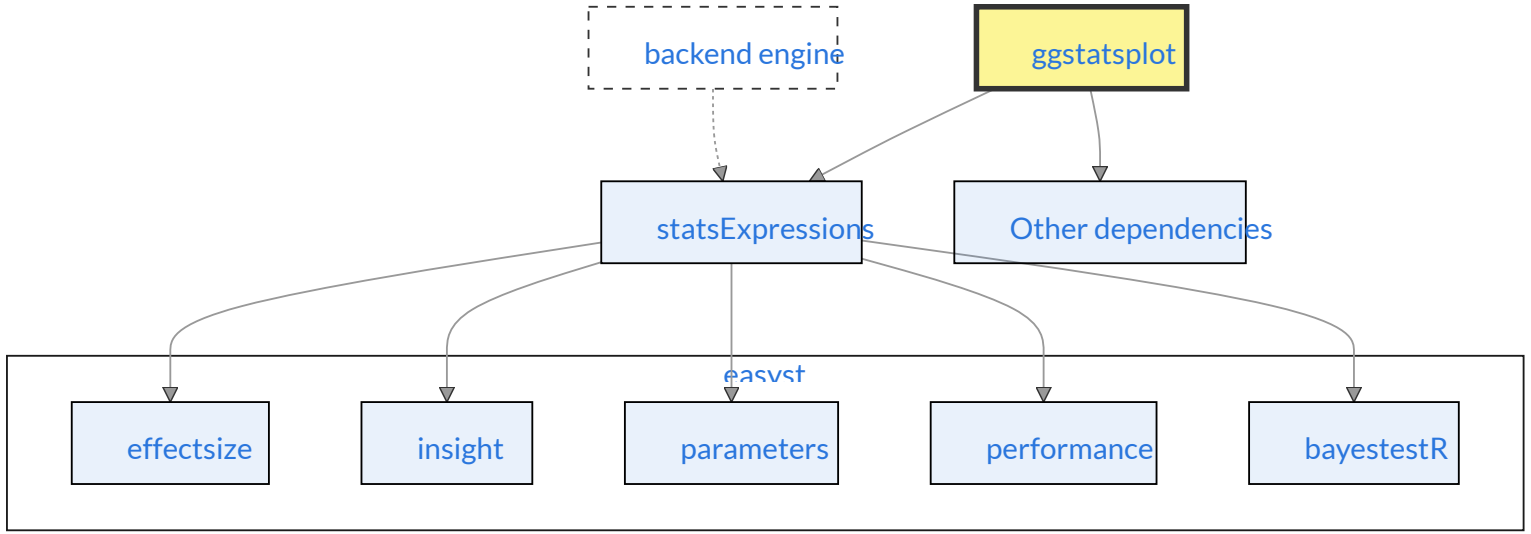
✅ Encourages best practices in data visualization and statistical reporting!

Pleasant Side Effects

Maybe the real treasure was the technical skills we picked up along the way!

Software Architecture

Breaking down the monolith: $20K_{(2017)} \rightarrow 1K_{(2024)}$ lines of code



Working with open-source teams

While re-architecting `{ggstatsplot}`, I started contributing upstream:

- joined the core `{easystats}` team (a collection of ten component packages)
- became a co-maintainer of `{ggsignif}`
- contributed to `{WRS2}` and `{ggcorrplot}`

Quality Assurance

CI Checks (GitHub Actions)

- Unit tests (random-order)
- Code coverage (100%)
- Linting (0 lints)
- Formatting (0 issues)
- Documentation (website, no link rot, many 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

Working with open-source teams

While improving the QA tooling for `{ggstatsplot}`, I started contributing upstream:

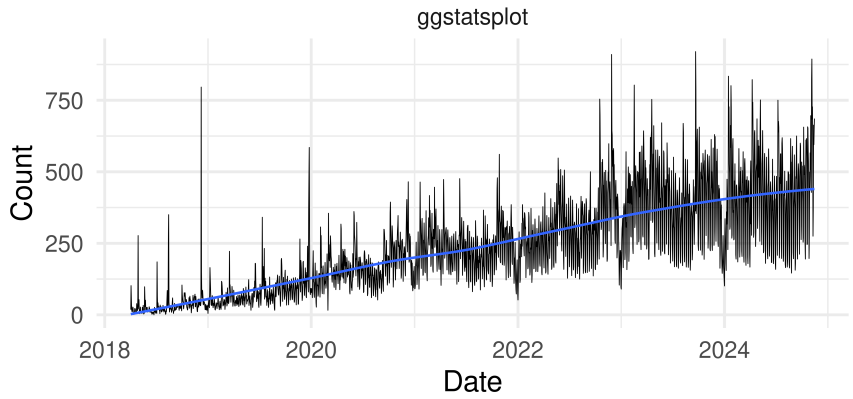
- became co-author of `{lintr}` (linter) and `{styler}` (formatter)

Impact

I can haz users?!

User Love

Total downloads > 500K (97 percentile)



Second most starred `{ggplot2}`-extension!

ggstatsplot Public

Enhancing {ggplot2} plots with statistical analysis 📊📣

R ☆ 2k 190

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.

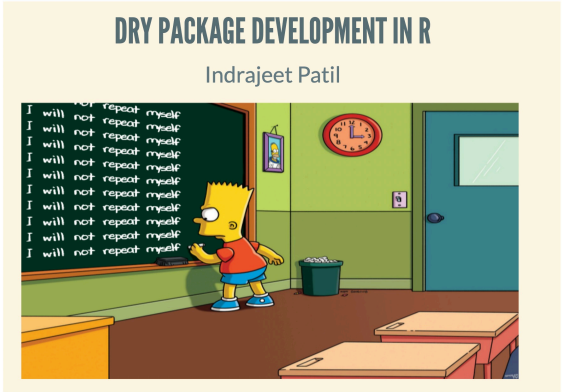
Visualizations with statistical details: The 'ggstatsplot' approach
I Patil
Journal of Open Source Software 6 (61), 3167, 2021

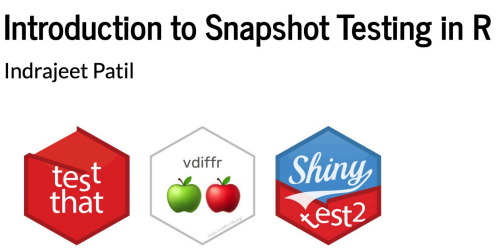
1277
*

Community Involvement

Communication

Multiple talks on best practices in software development to help community contribute to `{ggstatsplot}` (or its dependency).





Reception

- Improving Psychological Science Award (2020)
- 11 contributors
- 3 reverse dependencies
- Widely covered in YouTube videos and social media posts
- Almost 100% resolution rate on StackOverflow
- Over 100 daily visitors on GitHub repo
- Usage in a wide range of fields: psychology, biology, medicine, economics, etc.
- Usage in data science training programs

Conclusion

The `{ggstatsplot}` package provides accessible interface to create rich statistical visualizations, and helps users adopt a rigorous, reliable, and robust data exploratory and reporting workflow.

Thank You 

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

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-az1117-343
 ui       X11
 language (EN)
 collate  C.UTF-8
 ctype    C.UTF-8
 tz       UTC
 date     2024-11-15
 pandoc   3.5 @ /opt/hostedtoolcache/pandoc/3.5/x64/ (via rmarkdown)
 quarto   1.6.35 @ /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
```

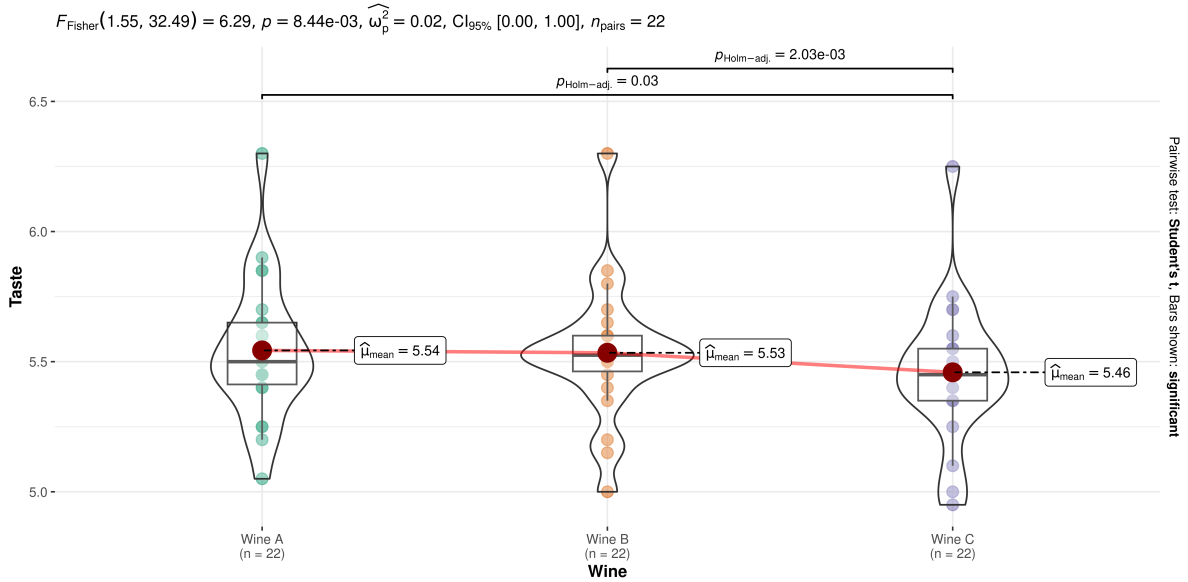
Appendix

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

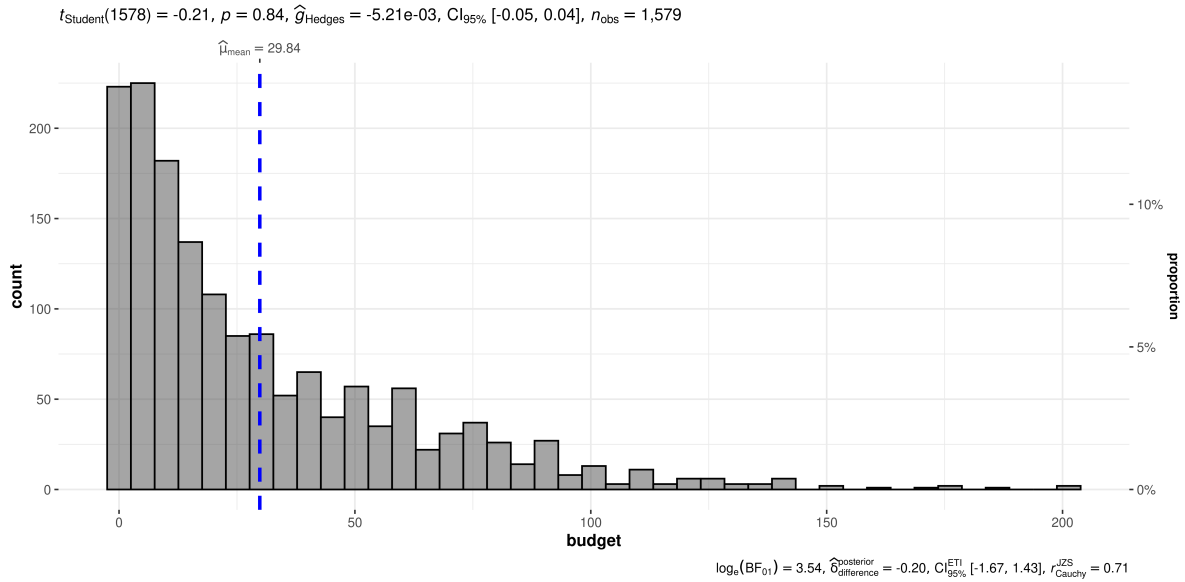
Statistical approaches available

- parametric
- parametric
- robust
- Bayesian

`gghistostats()`

Distribution of a numeric variable

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

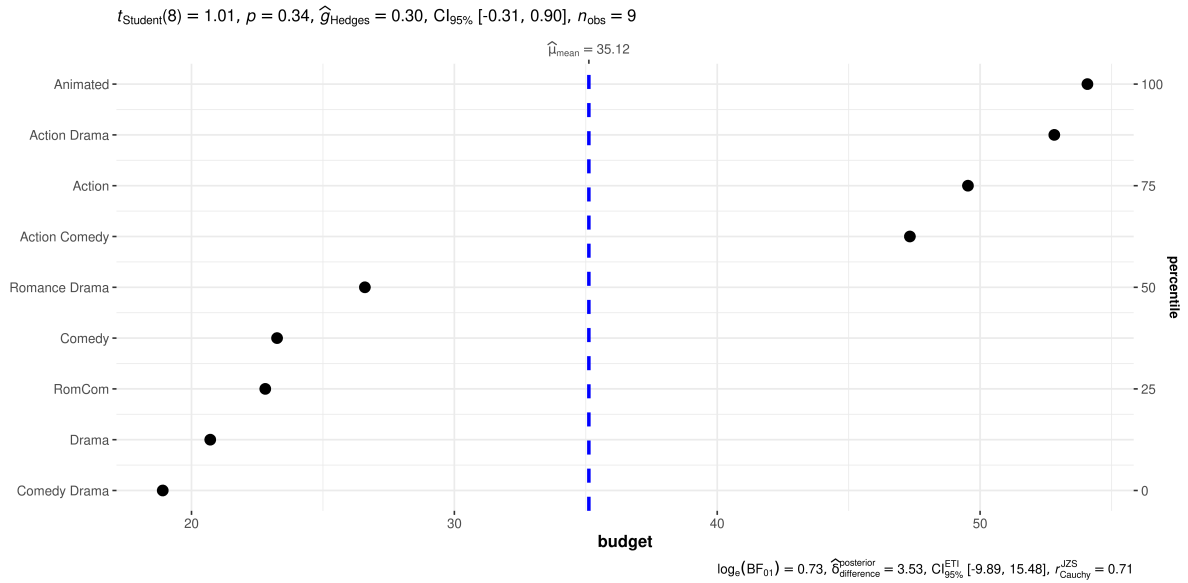


> **Important**
>
> ✏️ **Defaults**
>
> - counts + proportion for bins
> - descriptive statistics
> - inferential statistics
> - effect size + uncertainty
> - pairwise comparisons
> - Bayesian hypothesis-testing
> - Bayesian estimation
>
> **Statistical approaches available**
>
> - parametric
> - parametric
> - 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

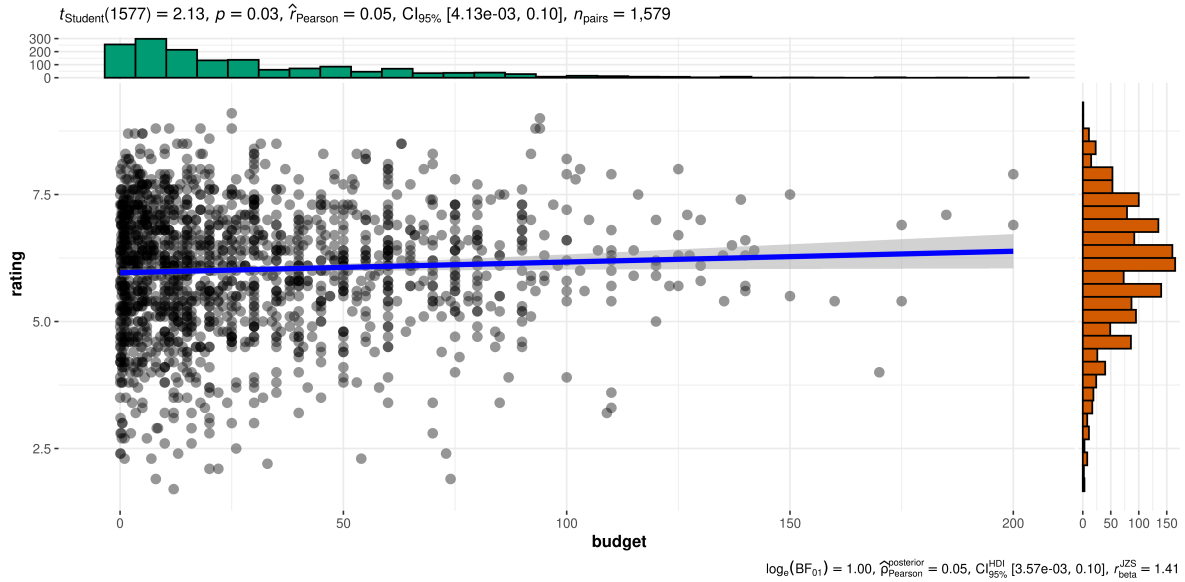
Statistical approaches available

- 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

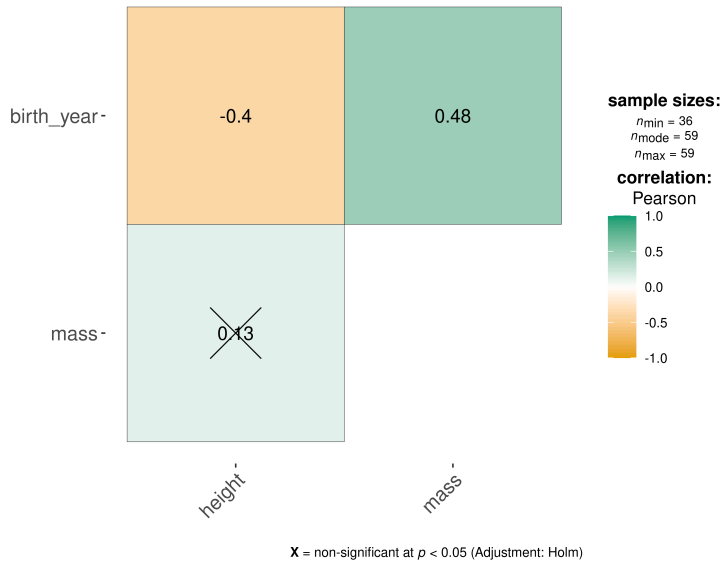
Statistical approaches available

- parametric
- parametric
- robust
- Bayesian

`ggcorrmat()`

Hypothesis about correlation: Multiple numeric variables

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

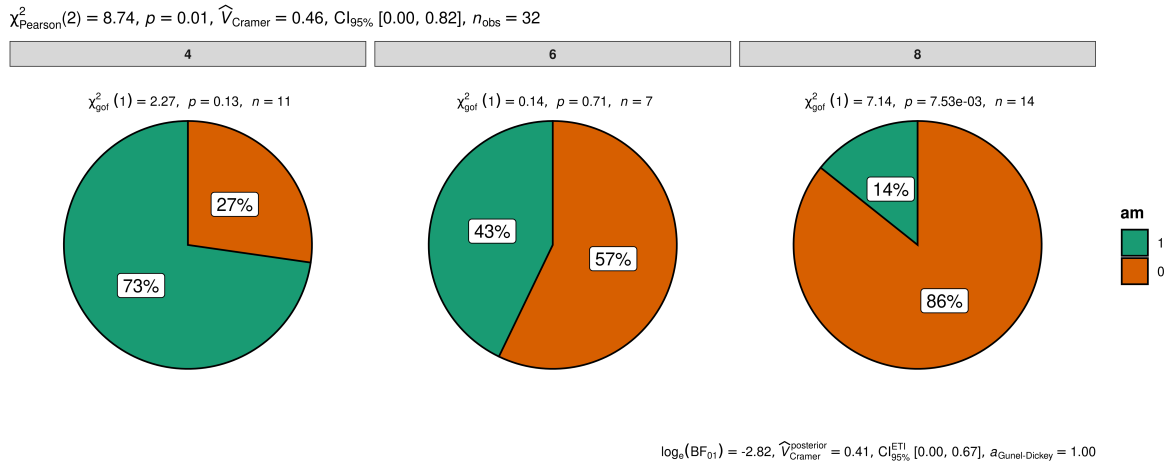


> **Important**
>
> **✏️ Defaults**
>
> - inferential statistics
> - effect size + uncertainty
> - careful handling of `NA`s
> - partial correlations
>
> **Statistical approaches available**
>
> - parametric
> - parametric
> - robust
> - Bayesian

`ggpiestats()`

Hypothesis about composition of categorical variables

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



Important

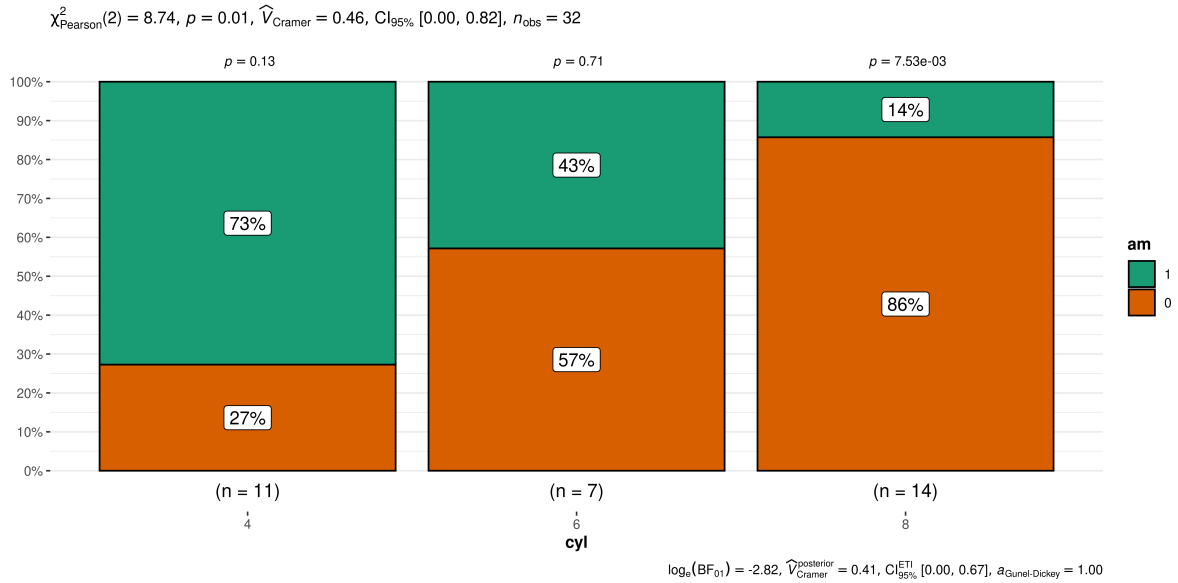
✏️ **Defaults**

- 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**

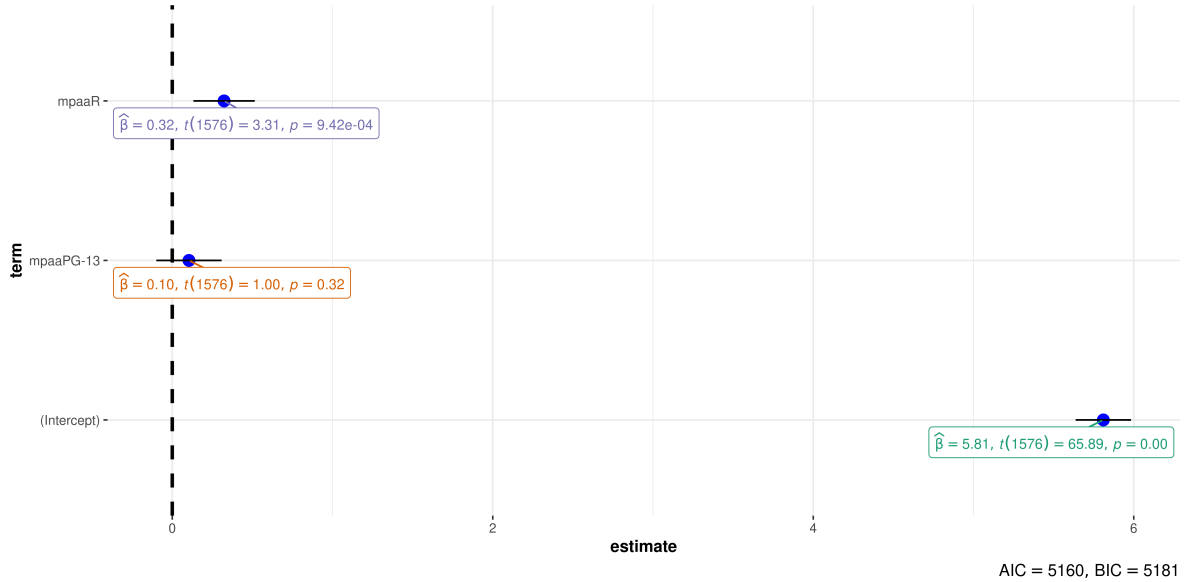
- descriptive statistics
- inferential statistics
- effect size + uncertainty
- goodness-of-fit tests
- Bayesian hypothesis-testing
- Bayesian estimation

`ggcoefstats()`

Hypothesis about regression coefficients

```
mod <- lm(
  formula = rating ~ mpaa,
  data = movies_long
)

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.

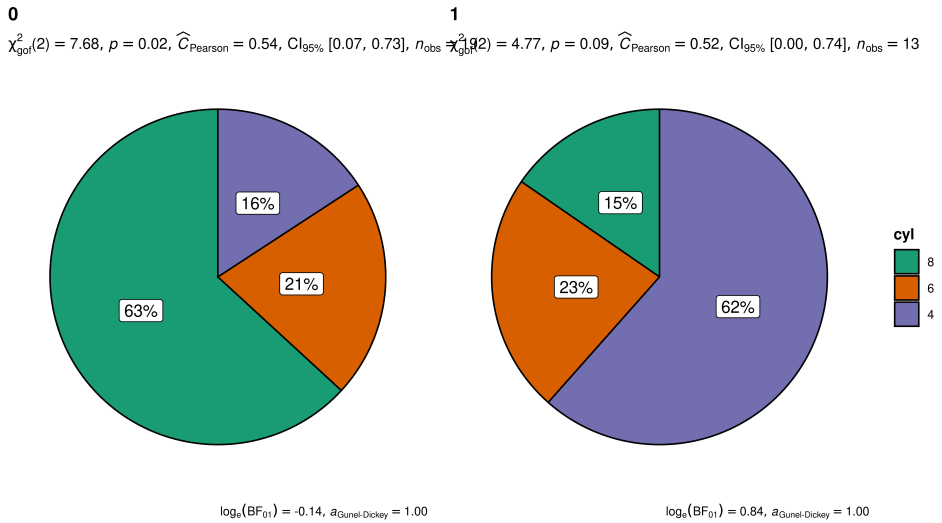
Meta-analysis is also supported!

*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()`

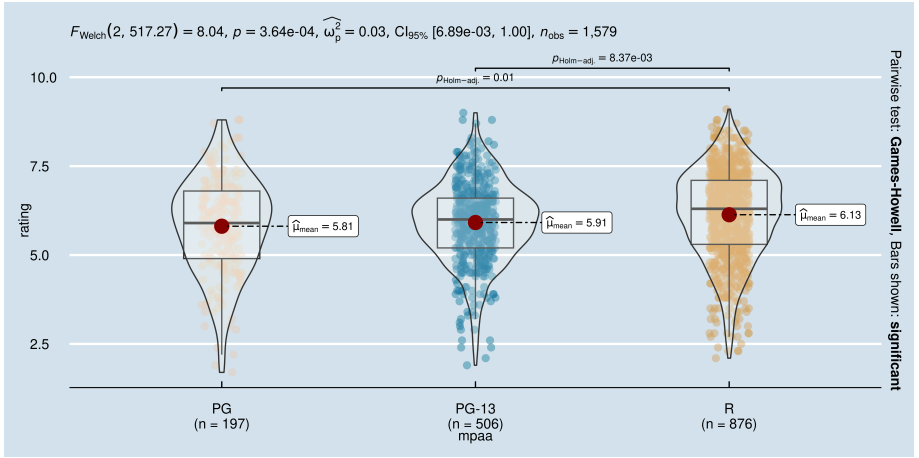
Customizability

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

Modify the look 🎨

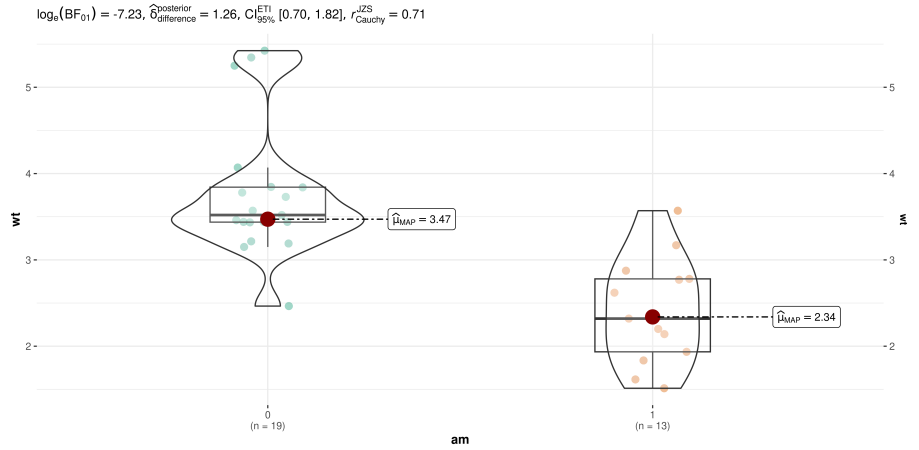
By changing **theme** and **palette**

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



By using {ggplot2} functions

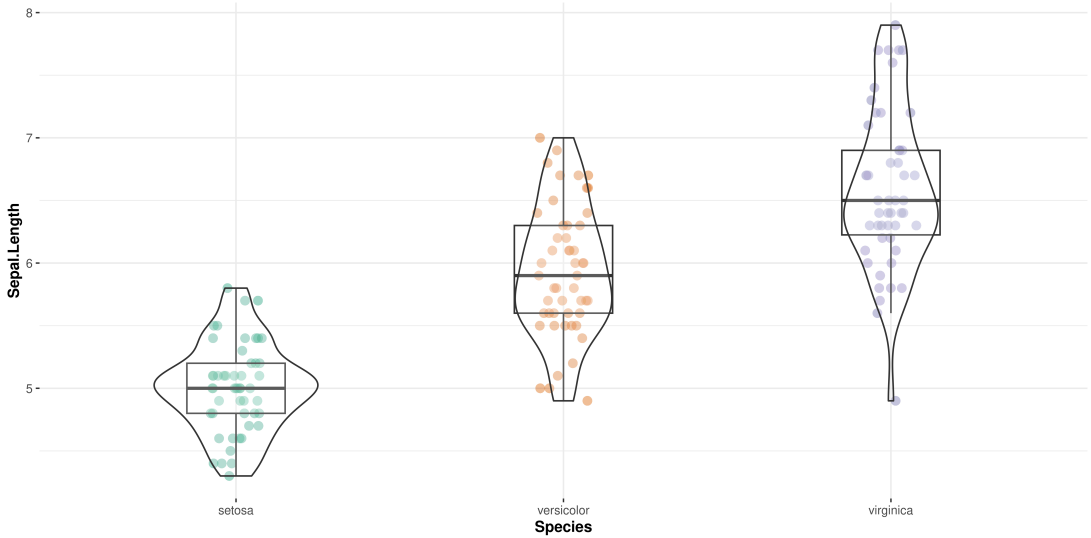
```
ggbetweenstats(
  data = mtcars,
  x = am,
  y = wt,
  type = "bayes"
) +
  scale_y_continuous(sec.axis = dup_axis())
```



Too much information 🙈

Get **only plots:**

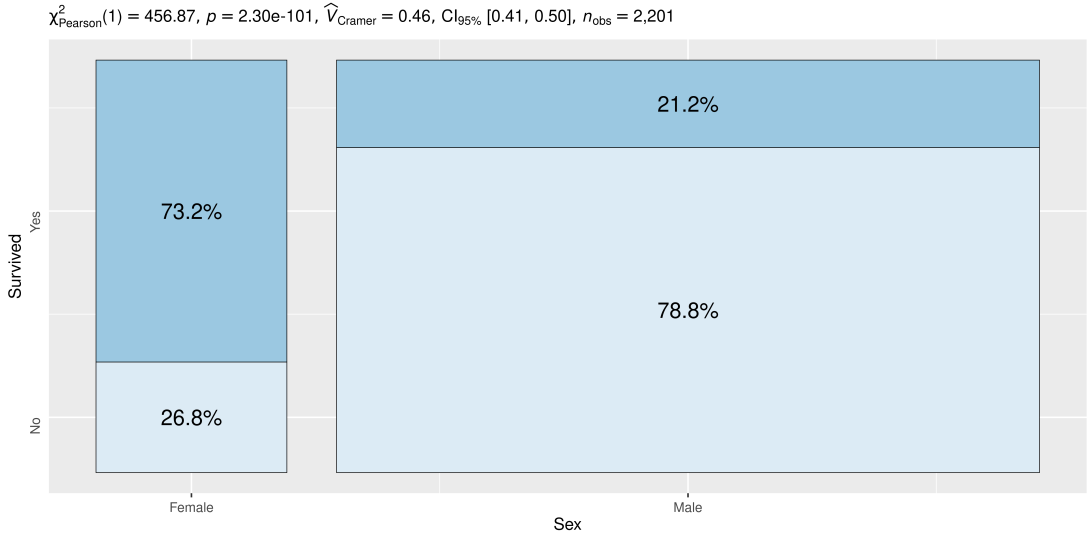
```
ggbetweenstats(
  data = iris,
  x = Species,
  y = Sepal.Length,
  # turn off statistical analysis
  centrality.plotting = FALSE,
  results.subtitle = FALSE,
  bf.message = FALSE,
  # turn off pairwise comparisons
  pairwise.display = "none"
)
```



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)
```



`{ggstatsplot}`: Details about statistical reporting

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	✅	✅

Toggling statistical approaches 🔀

Parametric

```r
# 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

```r
# 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?
- 🤔 ...

Benefits in details

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

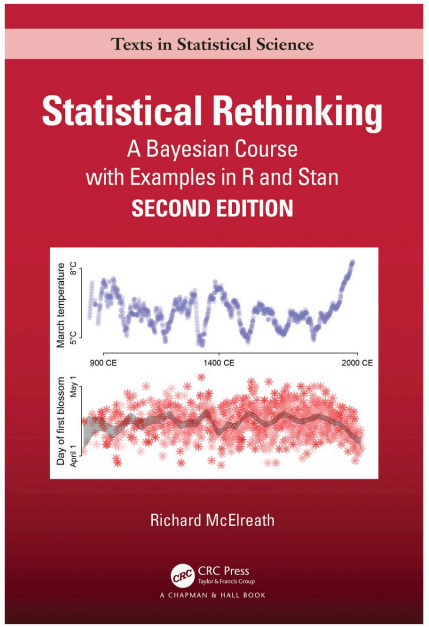
It...

- 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

A grain of salt

The "Golem of Prague" problem

❌ Promotes mindless application of statistical tests.



No stable release yet.

Footnotes

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

2. (Aczel et al., *AMPPS*, 2018)

3. Open Science Collaboration, *Science*, 2015