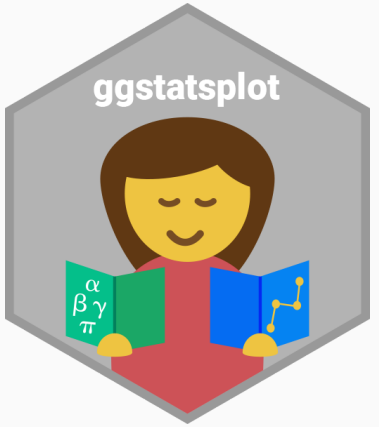
Introduction to `{ggstatsplot}`: `{ggplot2}` Plots with Statistics

Indrajeet Patil

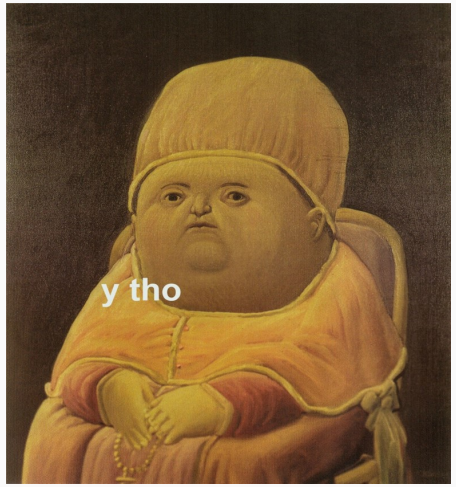


Plan

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

Why *ggstatsplot*?

Raison d'être



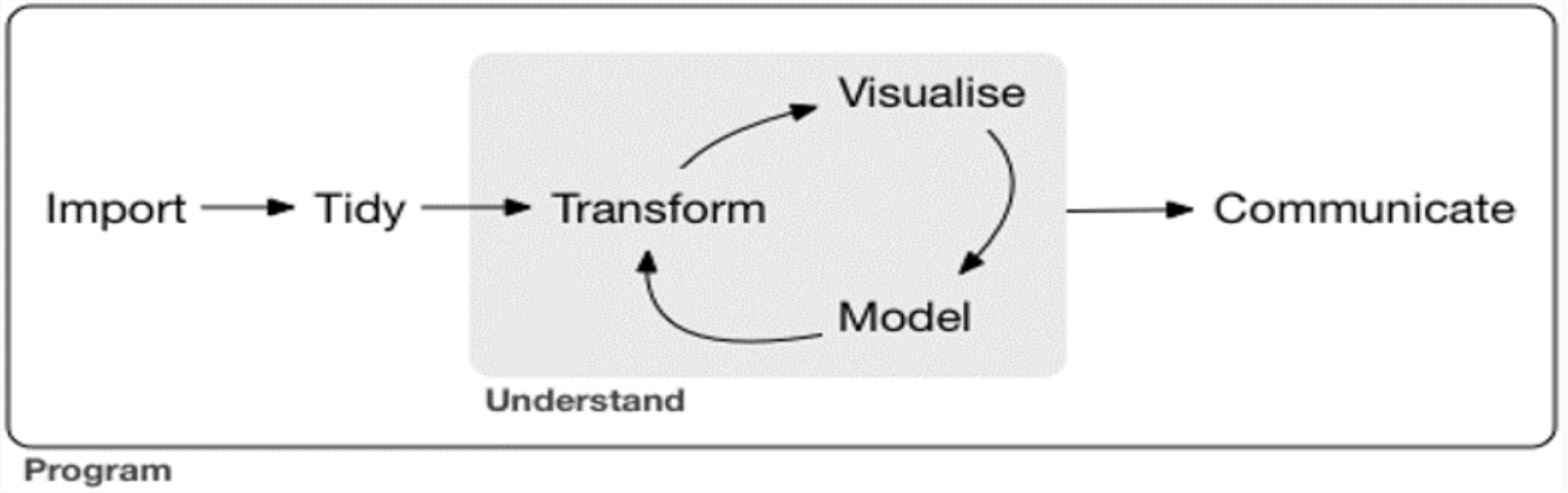
Current count of packages on the Comprehensive R Archive Network (`CRAN`) > 19,000

In short, `ggstatsplot` returns

📊 information-rich plots with statistical details, which are

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

Simpler/faster data analysis workflow

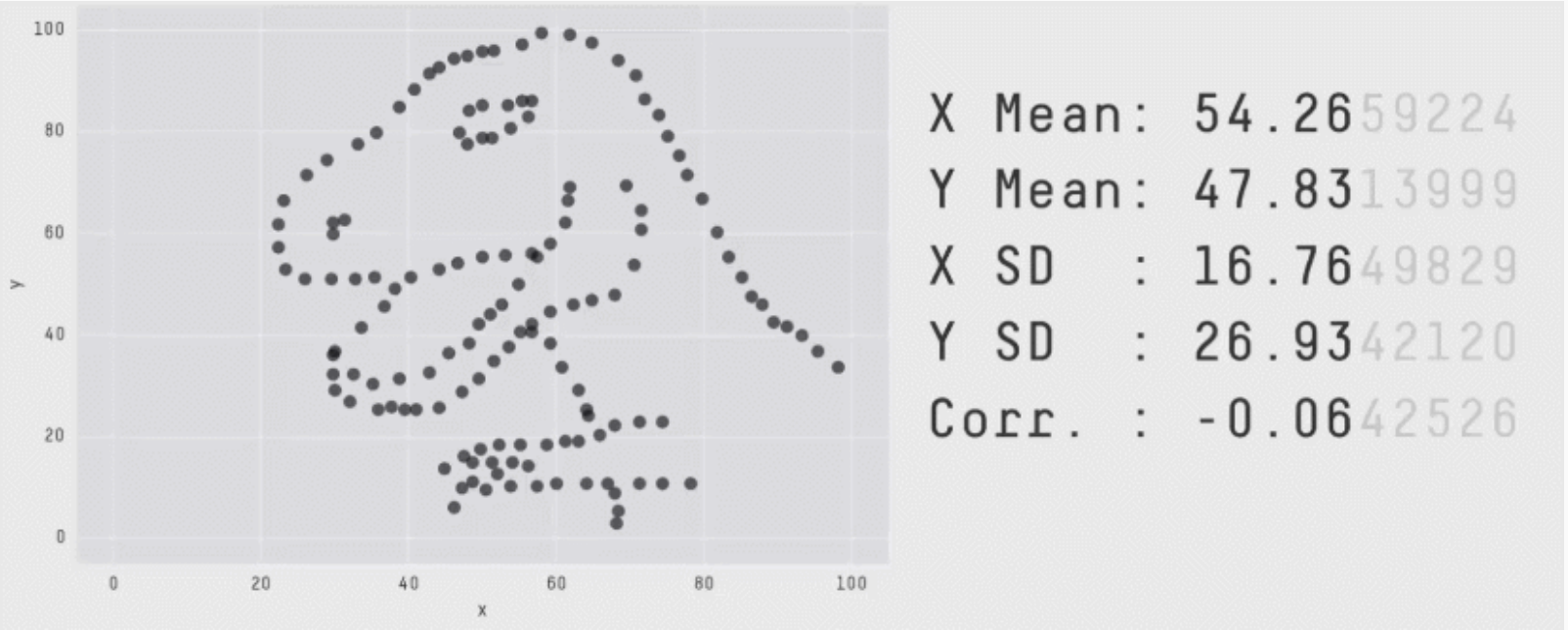


In a typical *exploratory* data analysis workflow, data visualization and statistical modeling are two different phases: visualization informs modeling, and modeling can suggest a different visualization, and so on and so forth.

💡 The central idea of `ggstatsplot` is simple: combine these two phases into one!

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

Information-rich graphic is worth 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...

$\sum_{time}$ (Needed time ↑ + Likelihood to graphical explore data ↓) = Avoidance habit

And a LOT more!

...but we will come back to that later 📌

Let's get started first!

Installation

Install the stable version of `ggstatsplot` from CRAN:

```
install.packages("ggstatsplot")
```

You can get the development version of the package from Github:

```
remotes::install_github("IndrajeetPatil/ggstatsplot")
```

Load the needed packages-

```
library(ggstatsplot)
library(ggplot2)
```

Primary functions

Hypothesis about group differences

ggbetweenstats - For between group comparisons

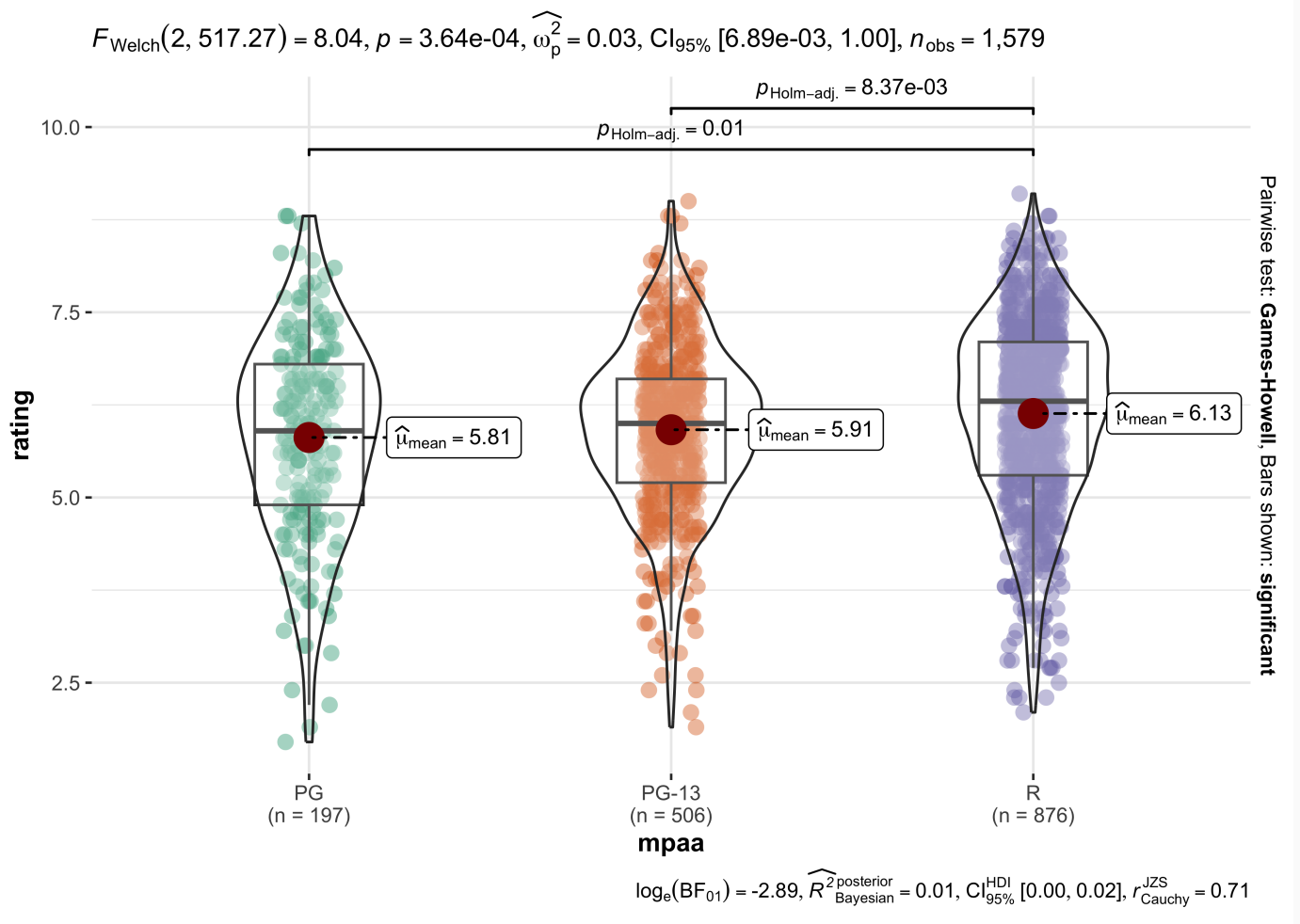
```
ggbetweenstats(
  data = movies_long,
  x = mpaa,
  y = rating
)
```

Function internally decides tests

- *t*-test if `2` groups
- ANOVA if `> 2` groups

✏️ Defaults return

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



ggwithinstats - repeated measures equivalent

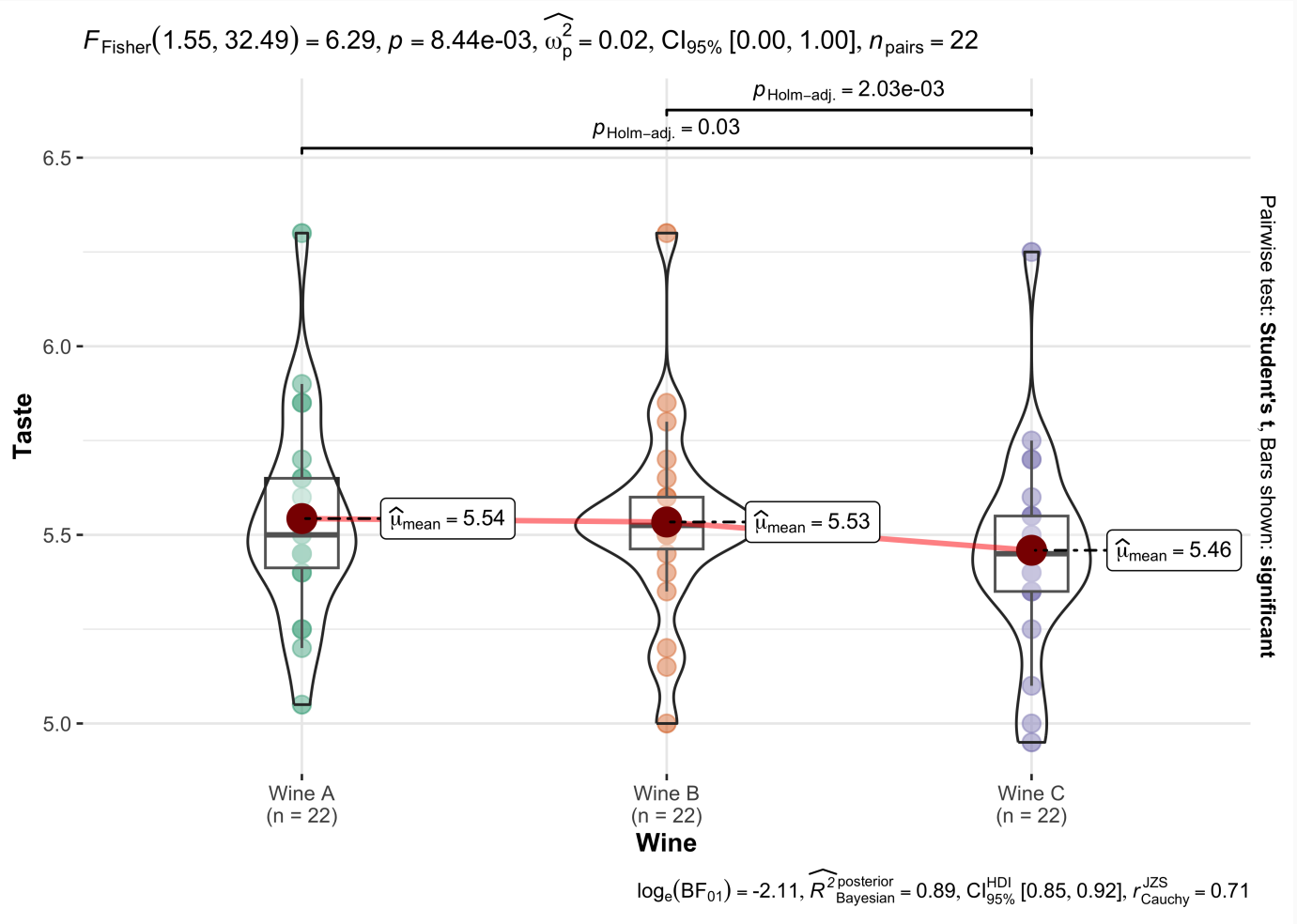
```
ggwithinstats(
  data = WRS2::WineTasting,
  x = Wine,
  y = Taste
)
```

✏️ Defaults return

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

Changing the `type` of test

- ✅ `"p"` → parametric
- ✅ `"np"` → non-parametric
- ✅ `"r"` → robust
- ✅ `"bf"` → Bayesian



gghistostats - Distribution of a numeric variable

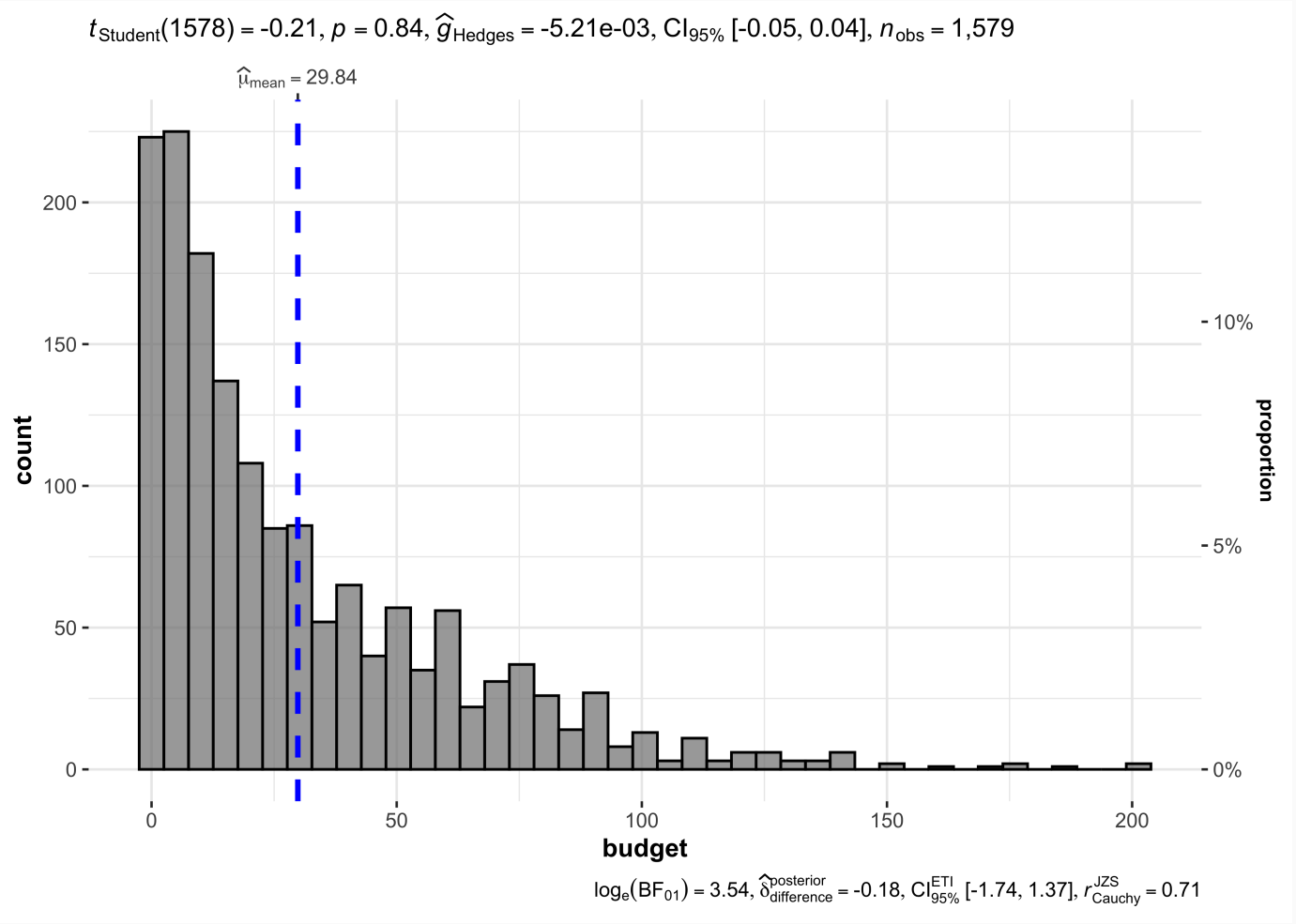
```
gghistostats(
  data = movies_long,
  x = budget,
  test.value = 30
)
```

✏️ Defaults return

- ✅ counts + proportion for bins
- ✅ descriptive statistics
- ✅ inferential statistics
- ✅ effect size + CIs
- ✅ Bayesian hypothesis-testing
- ✅ Bayesian estimation

Changing the `type` of test

- ✅ `"p"` → parametric
- ✅ `"np"` → non-parametric
- ✅ `"r"` → robust
- ✅ `"bf"` → Bayesian



ggdotplotstats - Labeled numeric variable

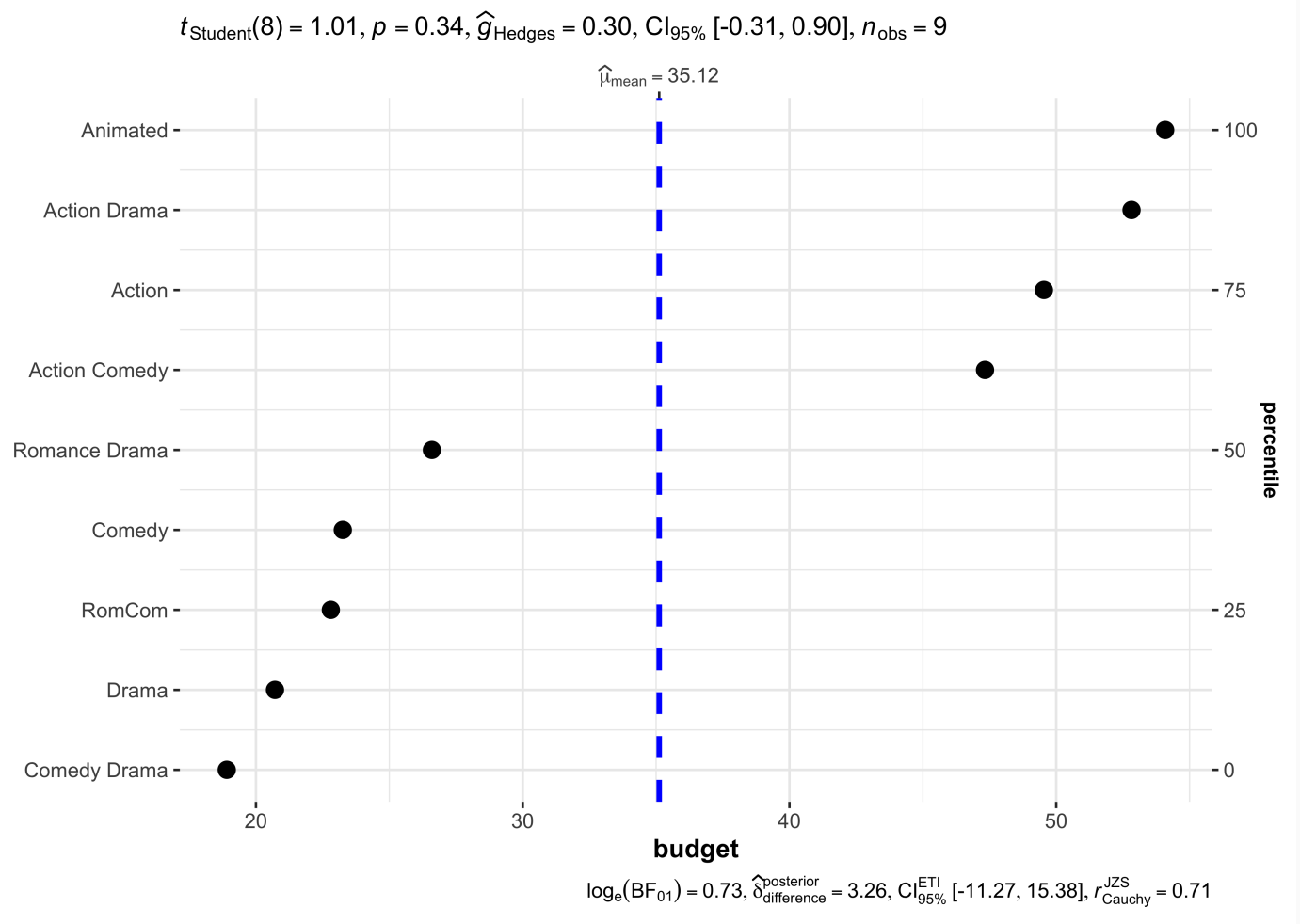
```
ggdotplotstats(
  data = movies_long,
  x = budget,
  y = genre,
  test.value = 30
)
```

✏️ Defaults return

- ✅ descriptive statistics
- ✅ inferential statistics
- ✅ effect size + CIs
- ✅ Bayesian hypothesis-testing
- ✅ Bayesian estimation

Changing the `type` of test

- ✅ `"p"` → parametric
- ✅ `"np"` → non-parametric
- ✅ `"r"` → robust
- ✅ `"bf"` → Bayesian



Hypothesis about correlation

ggscatterstats - Two numeric variables

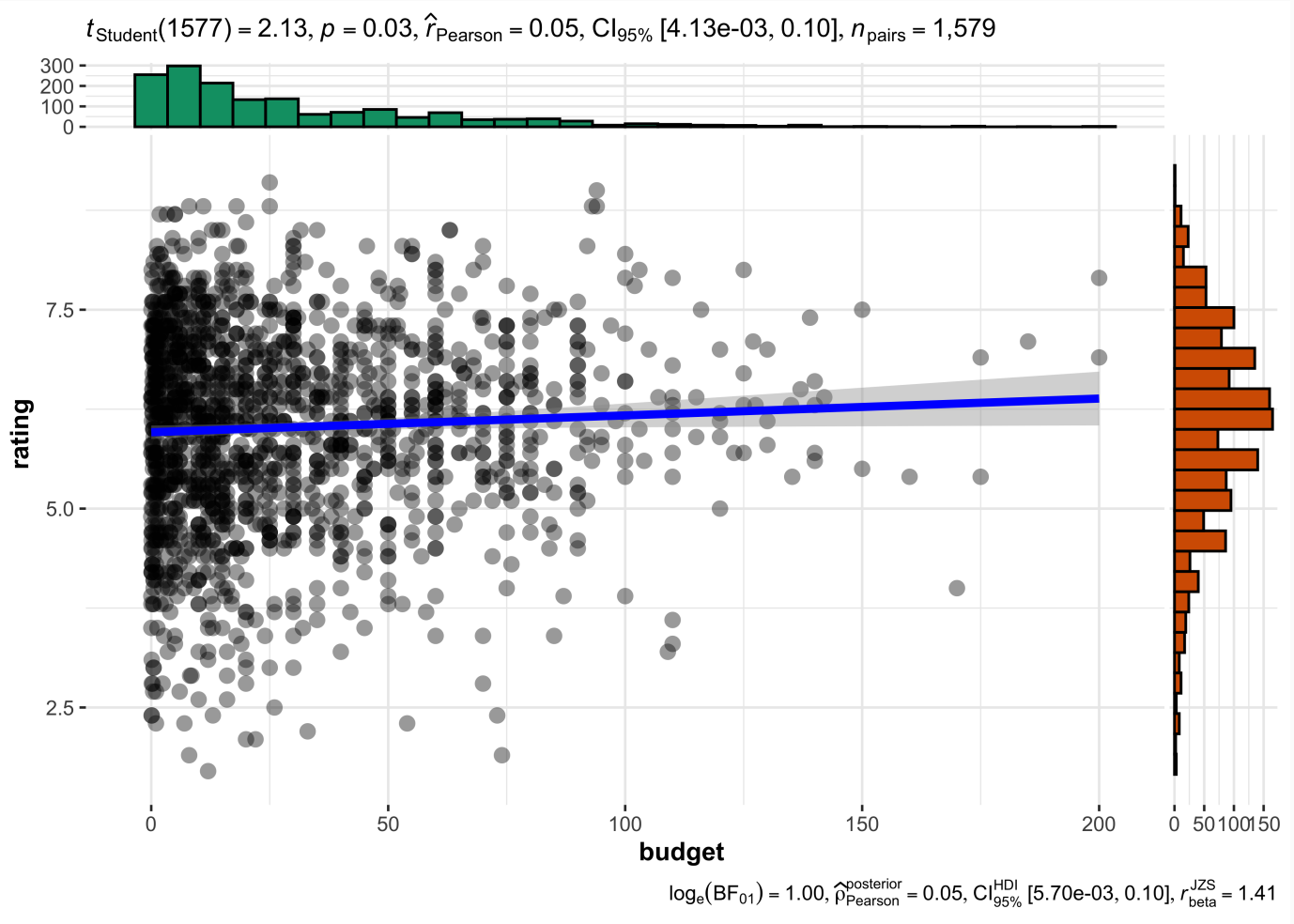
```
ggscatterstats(
  data = movies_long,
  x = budget,
  y = rating
)
```

✏️ Defaults return

- ✅ joint distribution
- ✅ marginal distributions
- ✅ inferential statistics
- ✅ effect size + CIs
- ✅ Bayesian hypothesis-testing
- ✅ Bayesian estimation

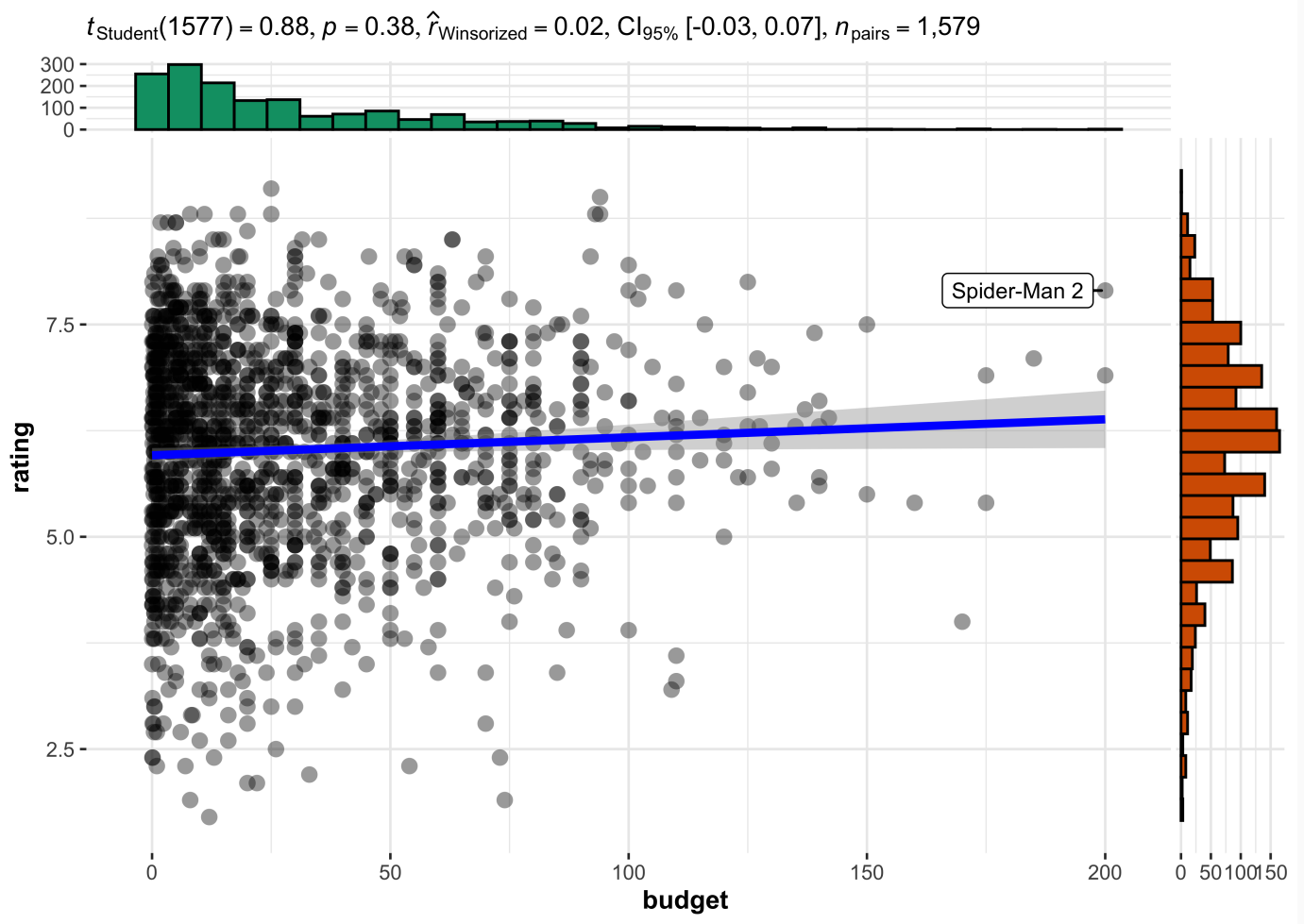
Changing the `type` of test

- ✅ `"p"` → parametric
- ✅ `"np"` → non-parametric
- ✅ `"r"` → robust
- ✅ `"bf"` → Bayesian



ggscatterstats - conditional point tagging

```
ggscatterstats(
  data = movies_long,
  x = budget,
  y = rating,
  type = "r",
  label.var = title,
  label.expression = budget > 150
  & rating > 7.5
)
```



ggcorrmat - multiple numeric variables

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

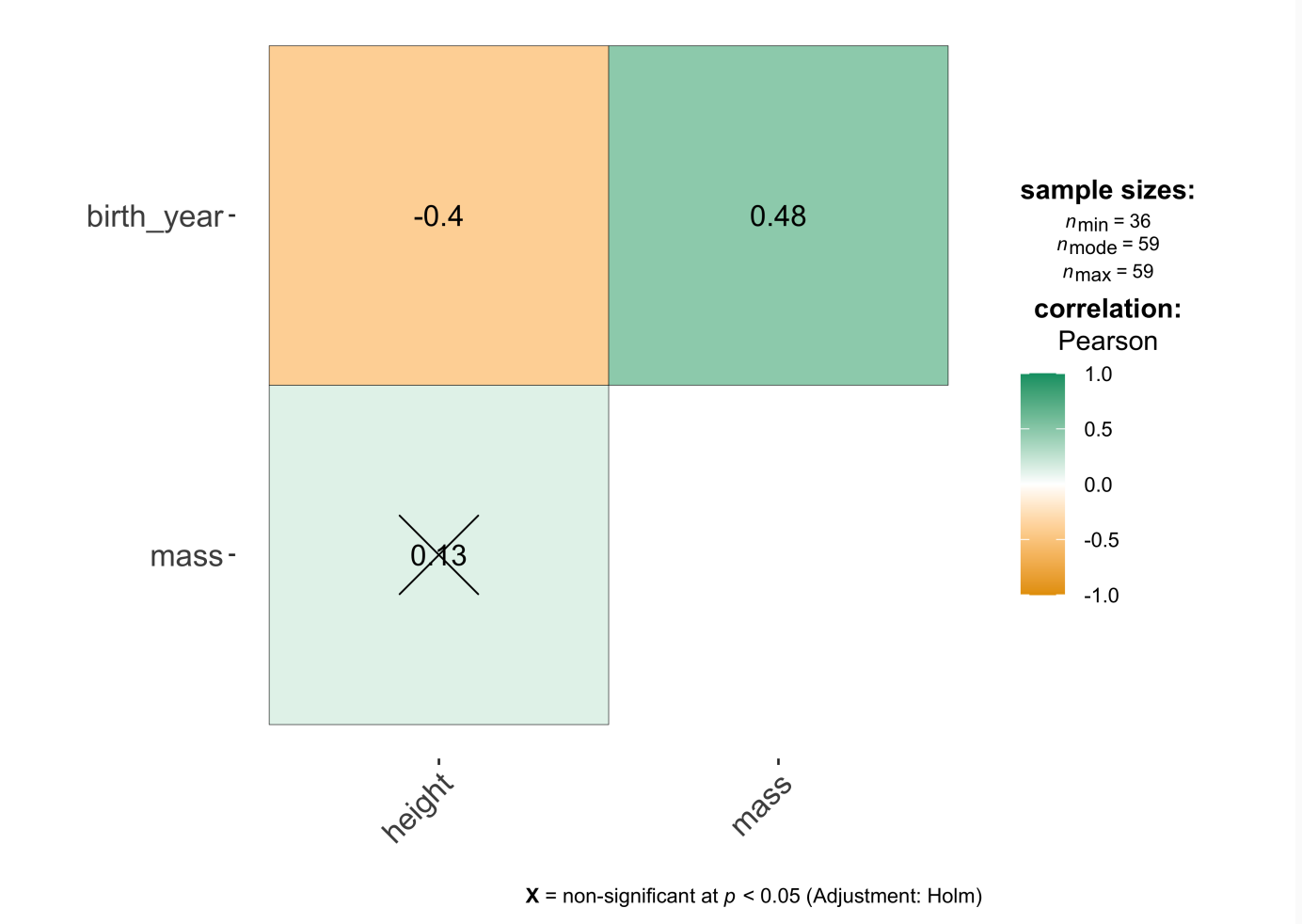
✏️ Defaults return

✅ effect size + significance
✅ careful handling of `NA`s

Changing the `type` of test

✅ `"p"` → parametric
✅ `"np"` → non-parametric
✅ `"r"` → robust
✅ `"bf"` → Bayesian

Partial correlations are also supported! Just set `partial=TRUE`.



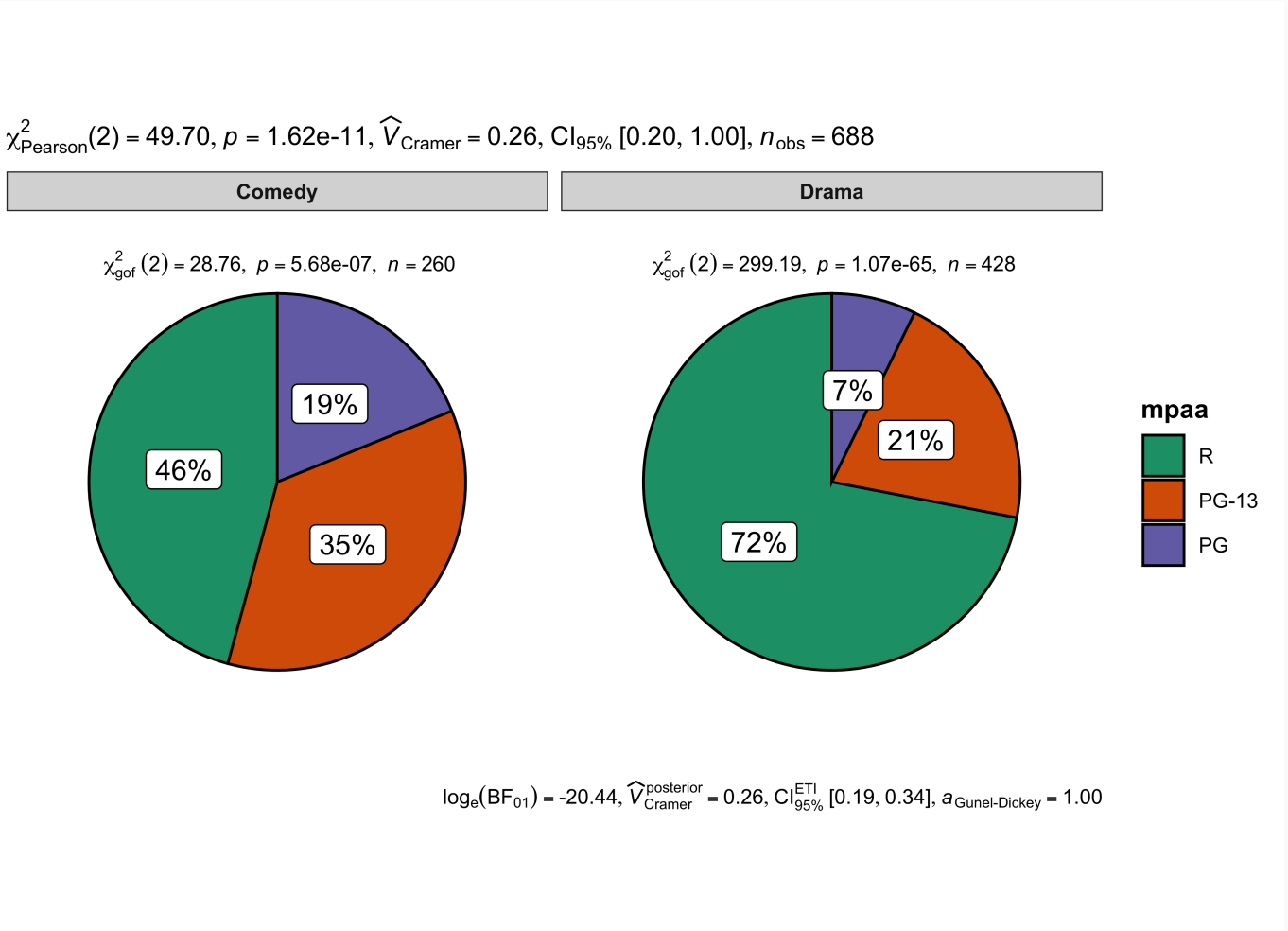
Hypothesis of composition of categorical variables

ggpiestats - association between categorical variables

```
ggpiestats(
  data = dplyr::filter(
    movies_long,
    genre %in% c("Drama", "Comedy")
  ),
  x = mpaa,
  y = genre
)
```

✏️ Defaults return

- ✅ descriptive statistics
- ✅ inferential statistics
- ✅ effect size + CIs
- ✅ Goodness-of-fit tests
- ✅ Bayesian hypothesis-testing
- ✅ Bayesian estimation

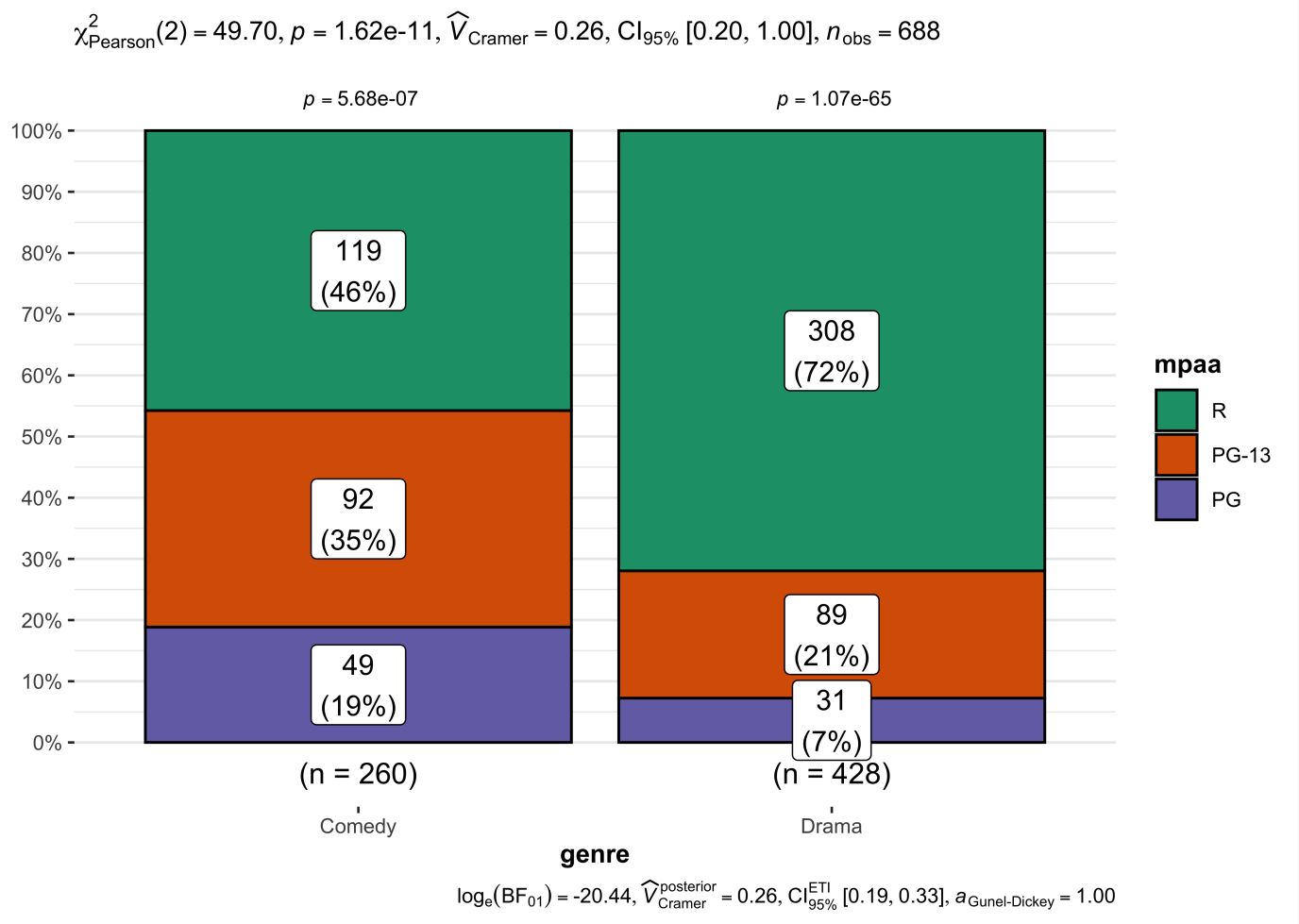


ggbarstats - association between categorical variables

```
ggbarstats(
  data = dplyr::filter(
    movies_long,
    genre %in% c("Drama", "Comedy")
  ),
  x = mpaa,
  y = genre,
  label = "both"
)
```

✏️ Defaults return

- ✅ descriptive statistics
- ✅ inferential statistics
- ✅ effect size + CIs
- ✅ Goodness-of-fit tests
- ✅ Bayesian hypothesis-testing
- ✅ Bayesian estimation



Hypothesis about regression coefficients

ggcoefstats

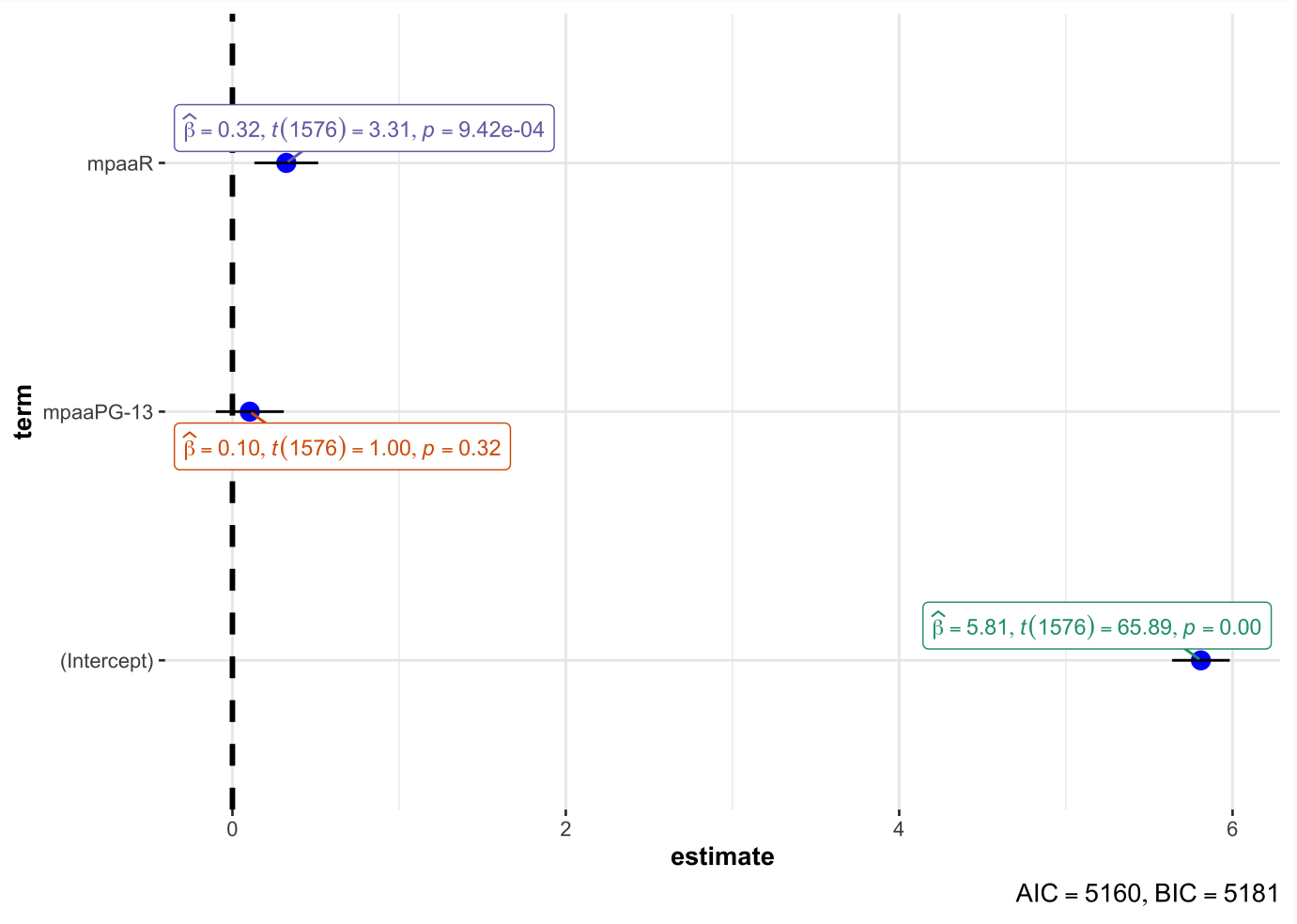
```
# model
mod ← lm(
  formula = rating ~ mpaa,
  data = movies_long
)

# plot
ggcoefstats(mod)
```

✏️ Defaults return

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

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



*grouped_* variants of all functions

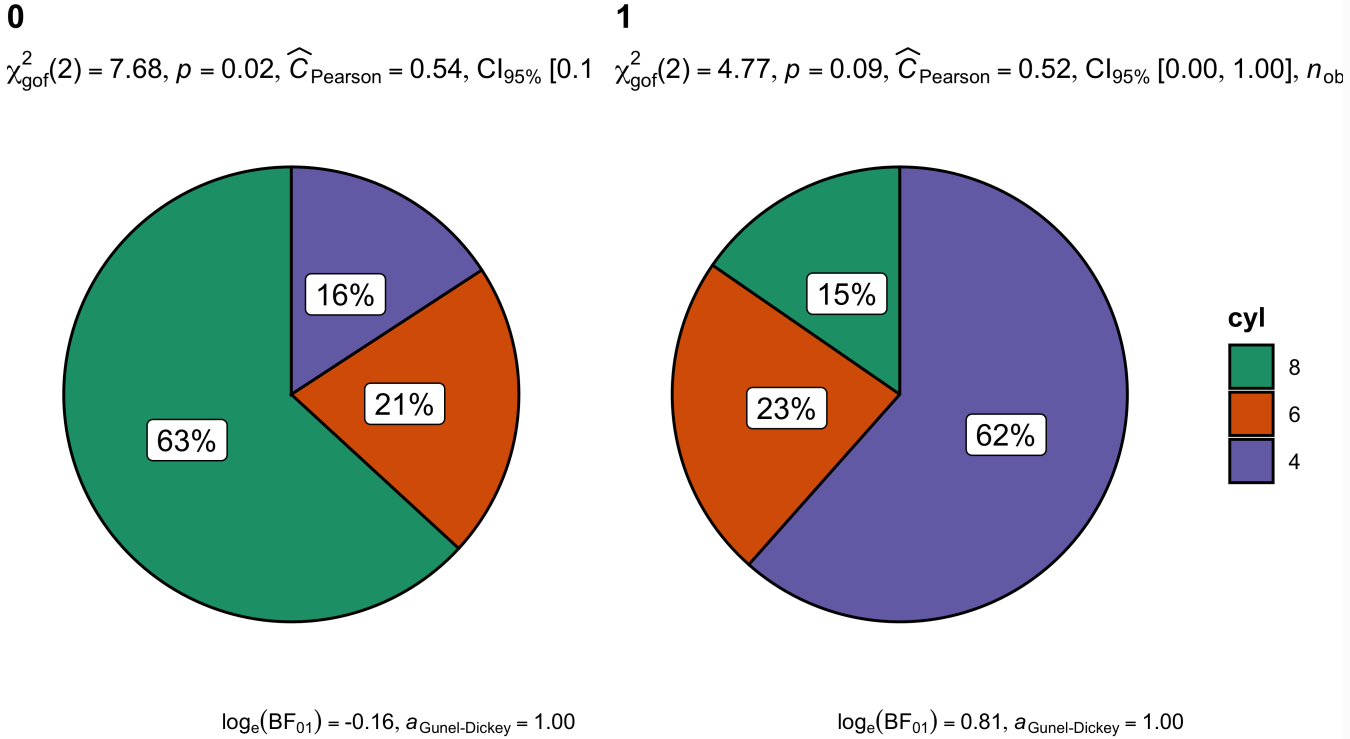
Running the same function for all levels of a single grouping variable

*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 of *ggstatsplot*

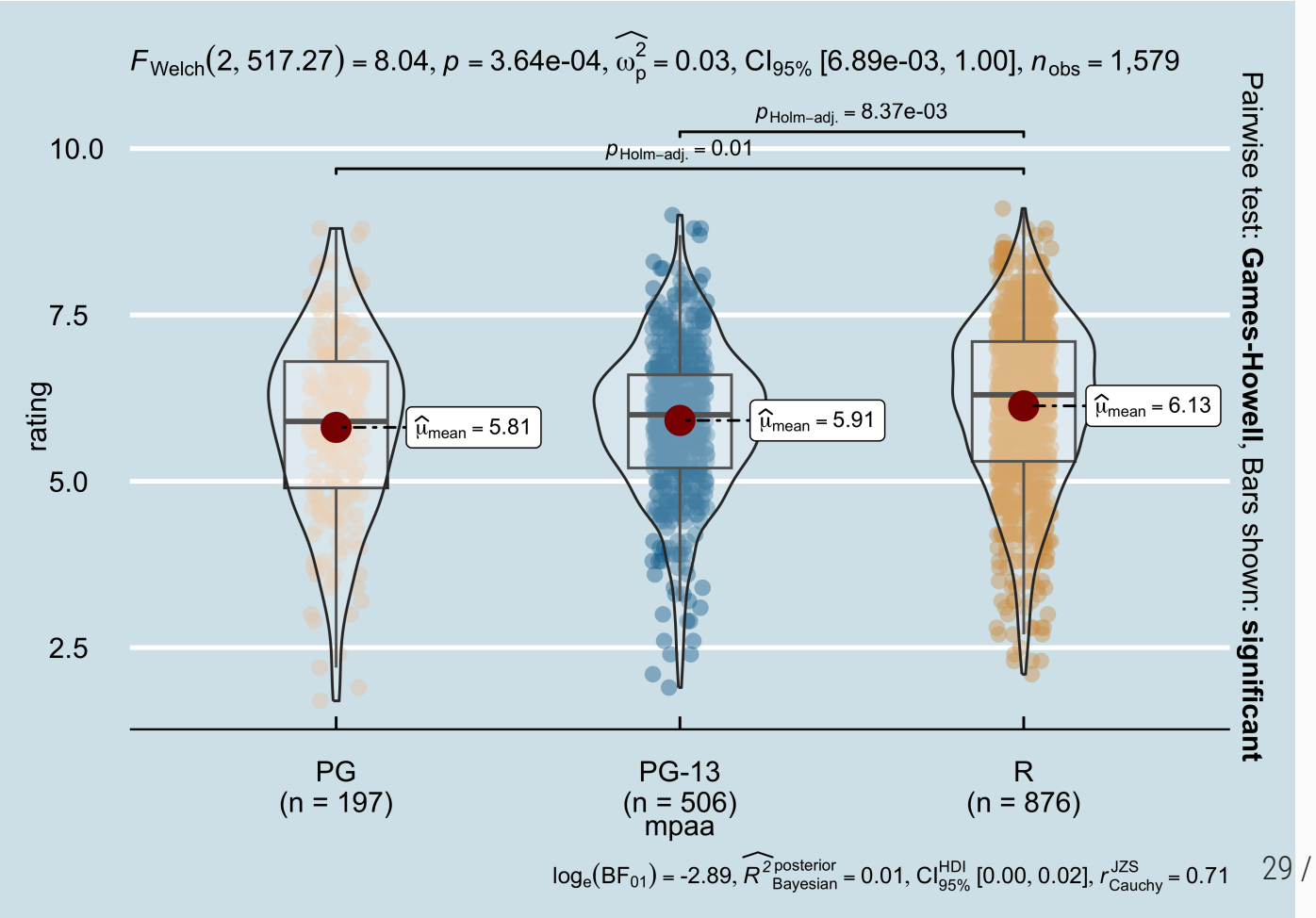
"What if I don't like the default plots?" 🤔

Changing aesthetics (themes + palettes) 🎨

Aesthetic preferences not an excuse to avoid `ggstatsplot`! 😻

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

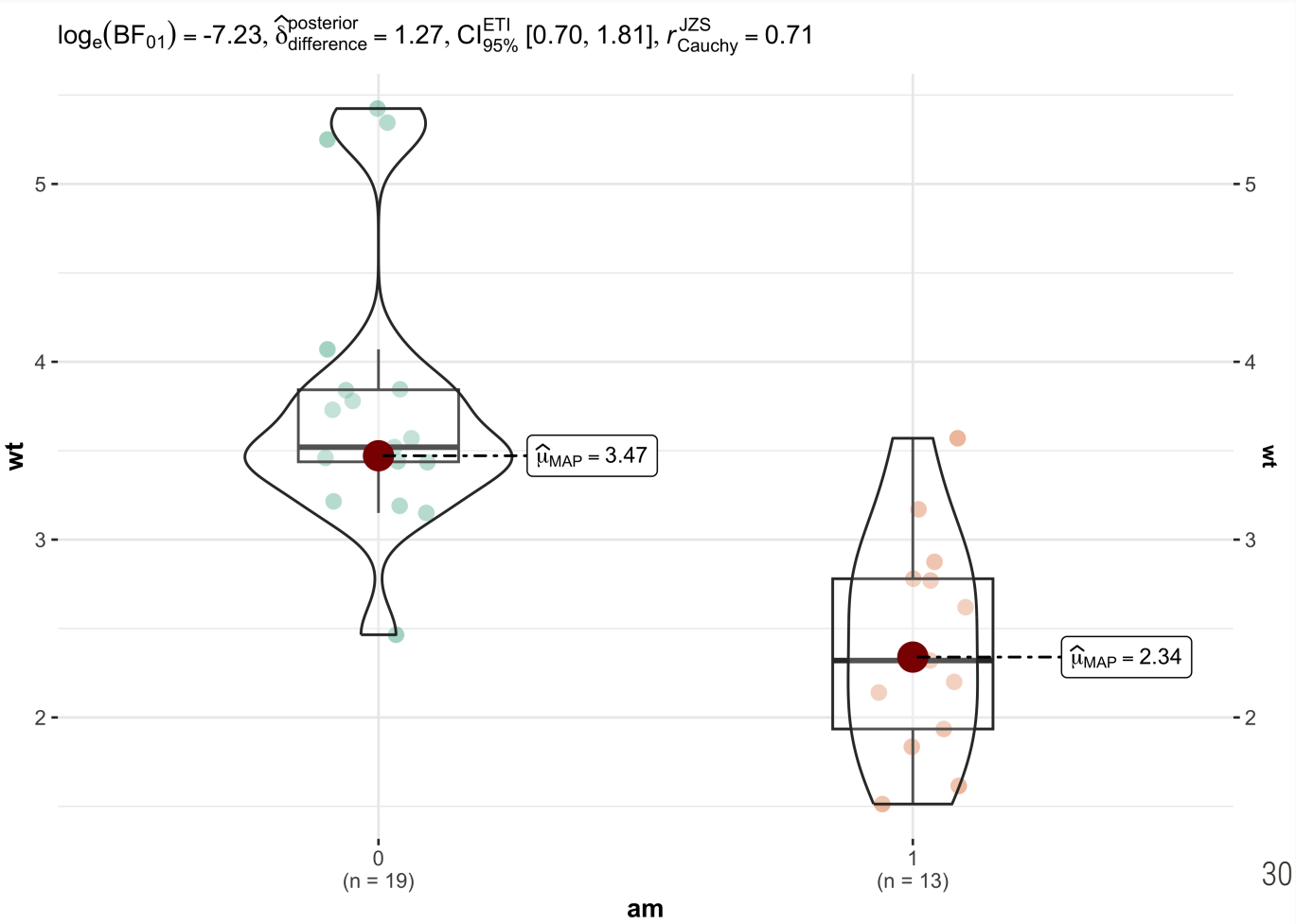
The default palette is colorblind-friendly.



Further modification with *ggplot2* 🛠️

You can modify `ggstatsplot` plots further using `ggplot2` functions. 🎉

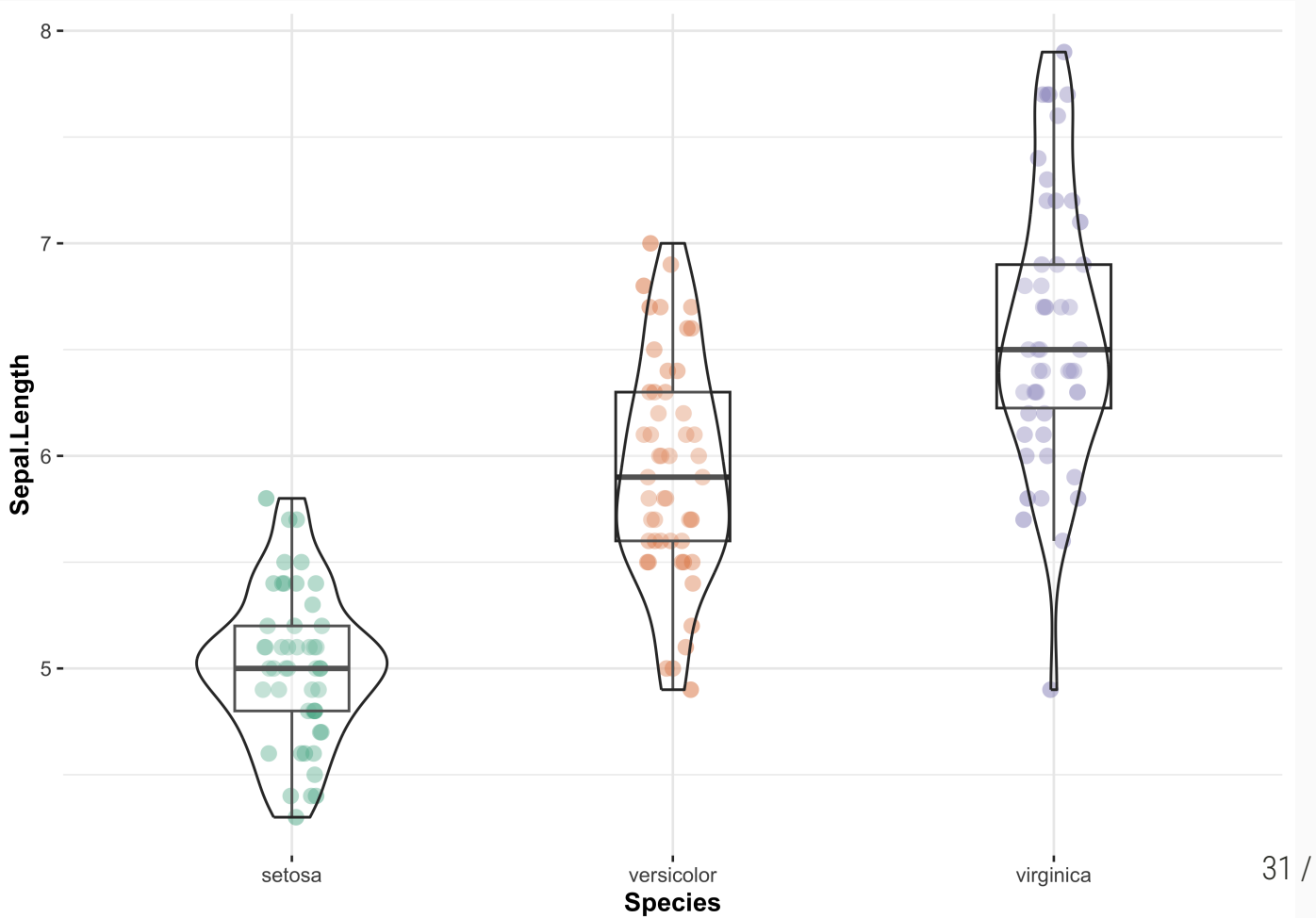
```
ggbetweenstats(
  data = mtcars,
  x = am,
  y = wt,
  type = "bayes"
) +
  scale_y_continuous(sec.axis = dup_axis())
```



Too much information 🙈

`ggstatsplot` can be used to get only plots.

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

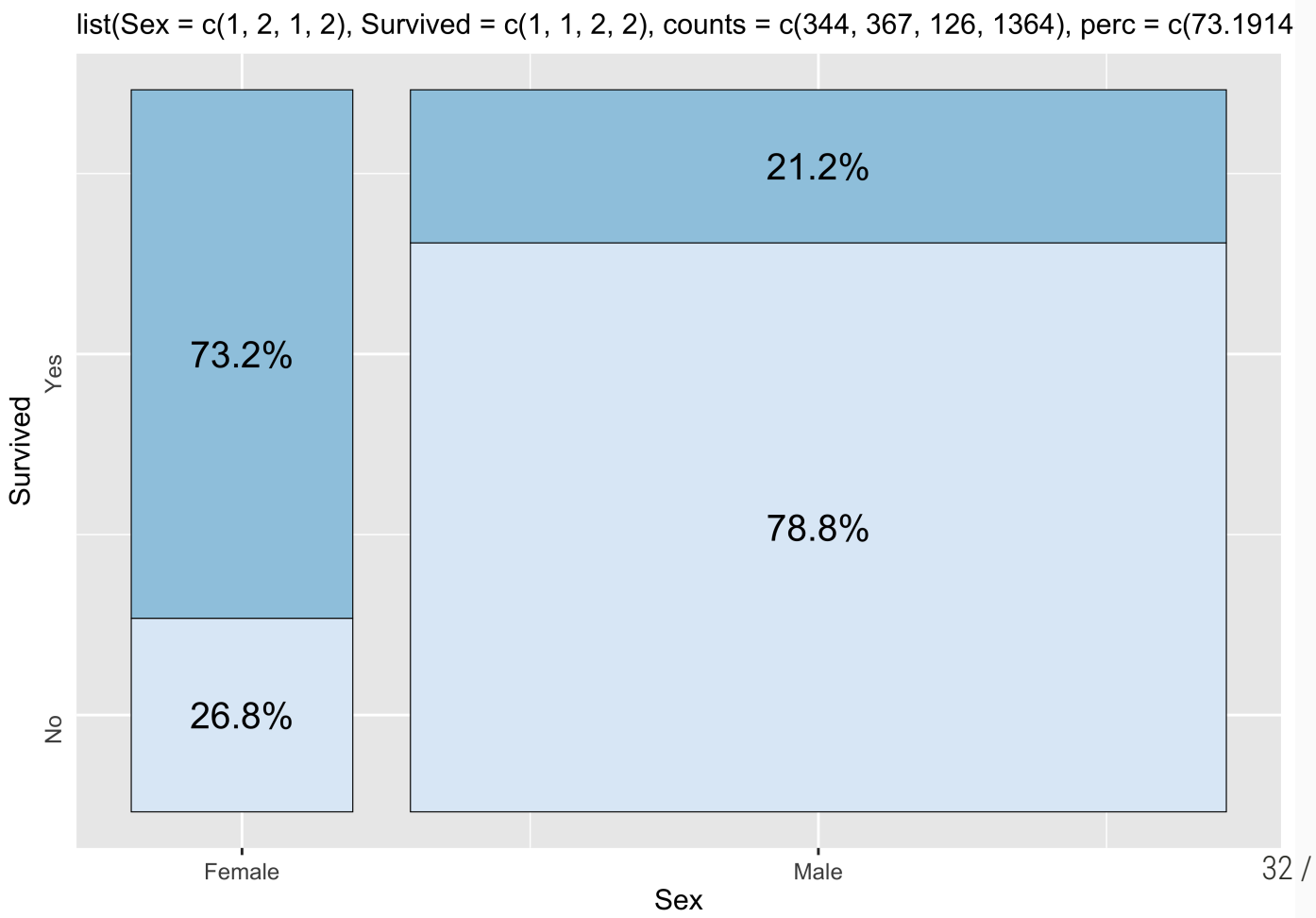


Expressions for custom plots 🀄

`ggstatsplot` can be used to get only expressions.

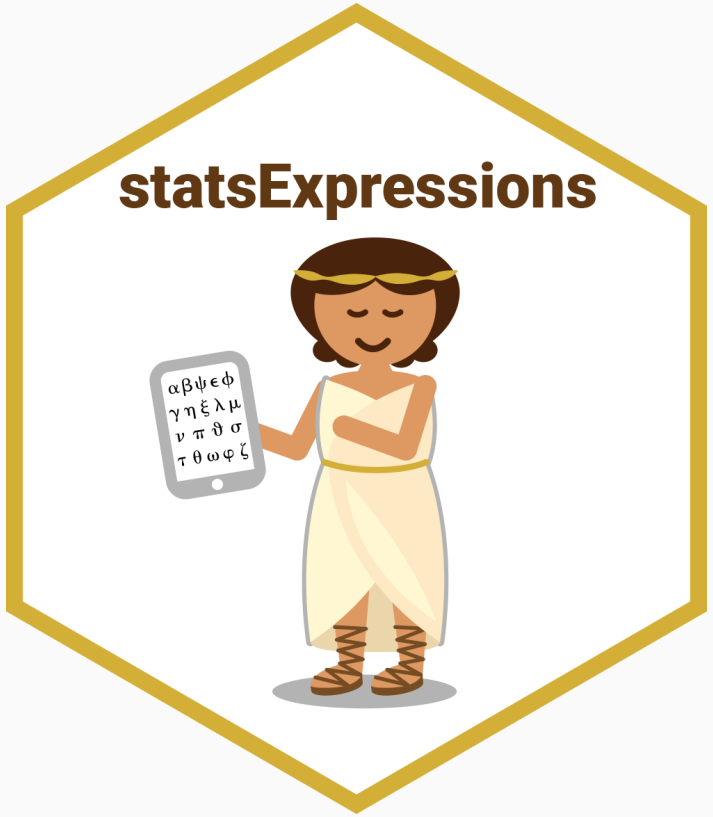
```
results ← ggpiestats(
  data = Titanic_full,
  x = Survived,
  y = Sex,
  output = "subtitle"
)

ggiraphExtra::ggSpine(
  data = Titanic_full,
  aes(x = Sex, fill = Survived),
  addlabel = TRUE,
  interactive = FALSE
) +
  labs(subtitle = results)
```



Dataframes

`statsExpressions`, statistical processing backend for `ggstatsplot`, can provide dataframes.




```
library(statsExpressions)

# for example
one_sample_test(
  data = mtcars,
  x = wt,
  test.value = 3
)
```

Why use *ggstatsplot*? 👍

Supports different statistical approaches

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 between 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 workflow to the following

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 😳

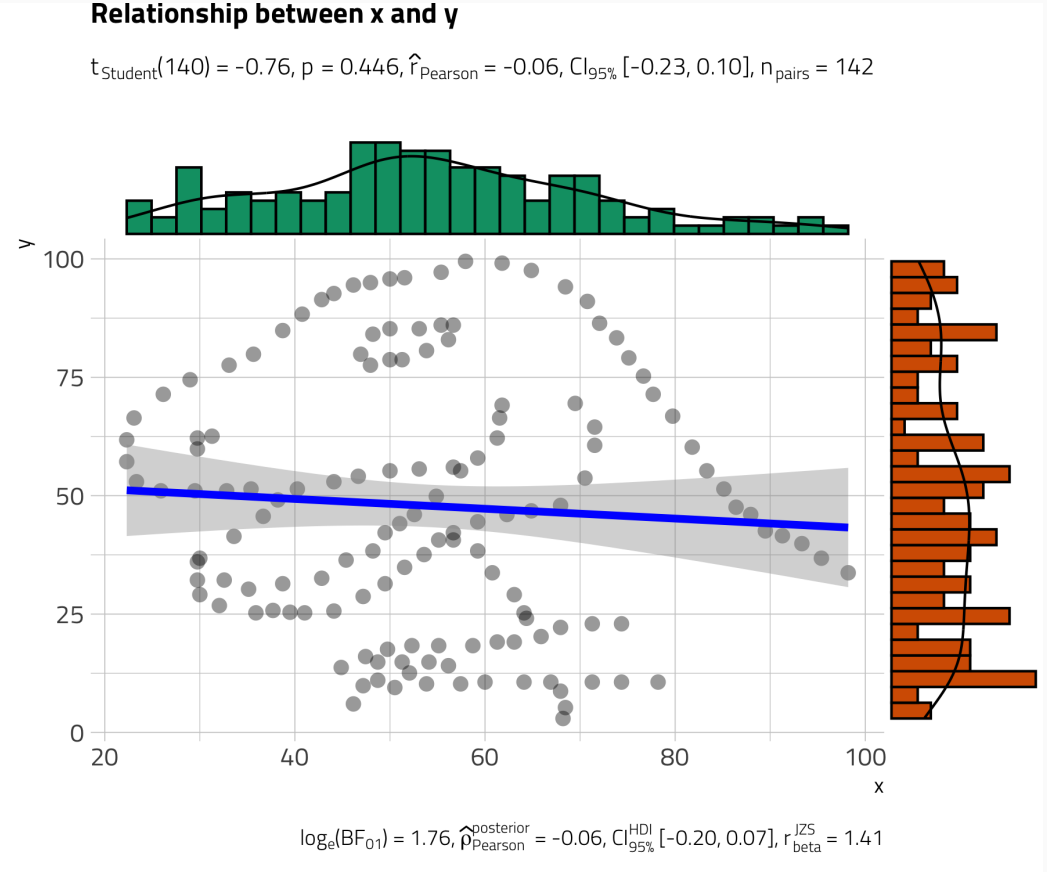
🤔 accepts dataframe, vectors, matrix?
🤔 long/wide format data?
🤔 works with `NA`s?
🤔 returns list, dataframe, arrays?
🤔 works with tibbles?
🤔 has all necessary details?
🤔 .

Results *in context* of the underlying data 🕵️

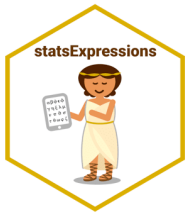
Standard approach

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



Best practices in statistical reporting 🏆



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

Template for Frequentist analysis

test: t_{Welch} (281.95) = -10.75, parameter: p = 8.31e-23, statistic: $\hat{g}_{\text{Hedges}}$ = -1.27, significance: $\text{CI}_{99\%}$ [-1.61, -0.94], effect size type + estimate + confidence intervals: n_{obs} = 284, number of observations

Template for Bayesian analysis

$\log_e(\text{BF}_{01})$ = -6.20, evidence in favor of null over alternative hypothesis: $\delta^{\text{posterior}}_{\text{difference}}$ = -5.06, natural logarithm of Bayes Factor: $\text{CI}^{\text{HDI}}_{95\%}$ [-6.75, -3.53], posterior type + estimate + credible intervals: $r^{\text{JZS}}_{\text{cauchy}}$ = 0.71, prior type and value

<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.

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. ✍️

No more excuses not to explore data! 😉

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.

Misconceptions and limitations

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 softwares: JASP and jamovi)

Limitations of *ggstatsplot* 👎

Limited no. of plots and statistical tests available. This will always be the case. 🤷

Expects a non-trivial level of statistical proficiency (but plots without statistics can still be useful).

Faceting does not work (since there are no corresponding `geom_` s). For the same reason, plots are not `{gganimate}`-friendly.

Overcoming these limitations 👥

Contributions (big or small) welcome!

Ways in which you can contribute

- Star on GitHub (increases visibility) ⭐
- Cite if used in a publication 📝
- Proof-read the documentation 📖
- Raise issues about bugs/features 🐞
- Review code 🕵
- Add new functionality 👩‍💻

Acknowledgments

Developer friends 🙌

Daniel Lüdecke, Dominique Makowski, Mattan S. Ben-Shachar, Brenton M. Wiernik

Support 💰

Mina Cikara, Fiery Cushman, Iyad Rahwan

Community 🙏

Contributors to *ggstatsplot* & *rstats* users and developers

More documentation

 Publication

 Website

 Yury Zablotski's YouTube playlist on *ggstatsplot*

For more

If you are interested in good programming and software development practices, check out my other slide decks.

Find me at...

🔭 @patilindrajeets

💻 @IndrajeetPatil

🔗 https://sites.google.com/site/indrajeetspatilmorality/

📧 patilindrajeet.science@gmail.com

The End 👋

To access code for these slides, see-

https://github.com/IndrajeetPatil/ggstatsplot_slides/