ggstatsplot Package

A.G.K.S. Dharmapriya, W.S.P. Dabarera

SE Statistics Visualization and More Using "R“
April 16th, 2024

Outline

- Why ggstatsplot?
- Setup For the Exercises
- Primary Functions
- Customizability of ggstatsplot
- Misconceptions & Limitations

What is ggstatsplot?

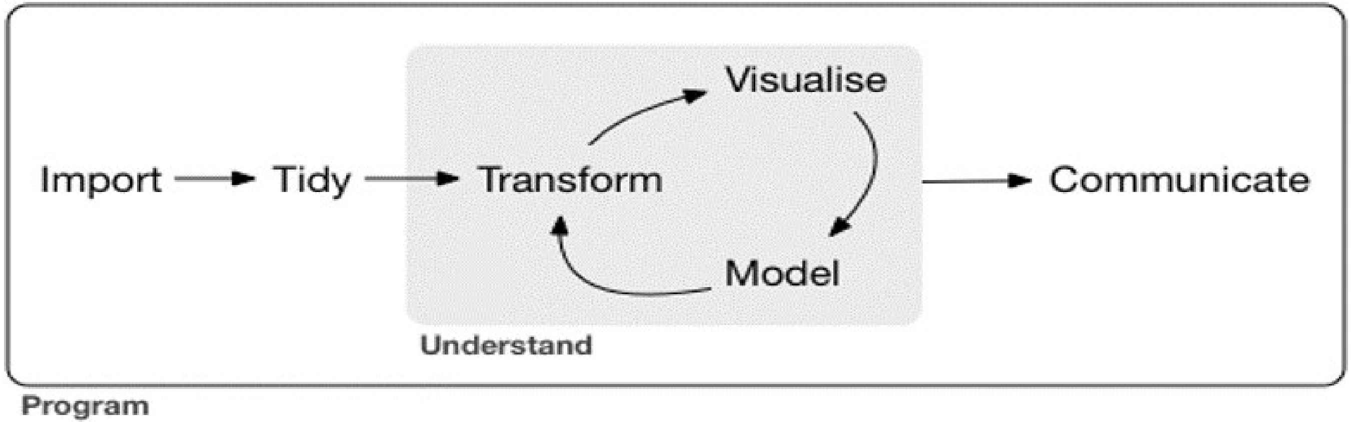
📊 information-rich plots with statistical details, which are📝 suitable for faster (exploratory) data analysis and scholarly reports

Extension of the `ggplot2`

Helpful in the Exploration Phase of the Data

Simpler/faster data analysis workflow

PARIS LODRON UNIVERSITÄT SALZBURG

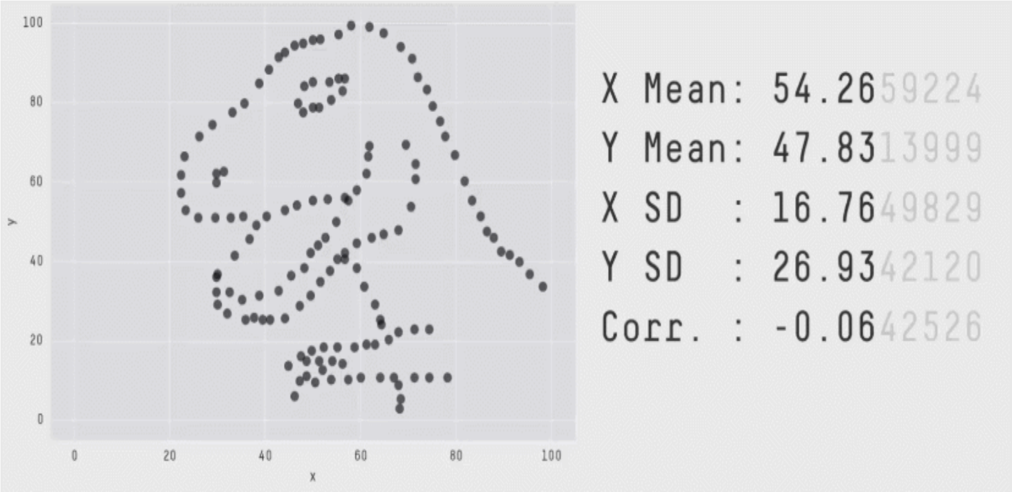


In a typical *exploratory* data analysis workflow, data visualization and statistical modeling are two different phases: visualization informs modeling, and modeling can suggest a different visualization, and so on and so forth.

💡 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



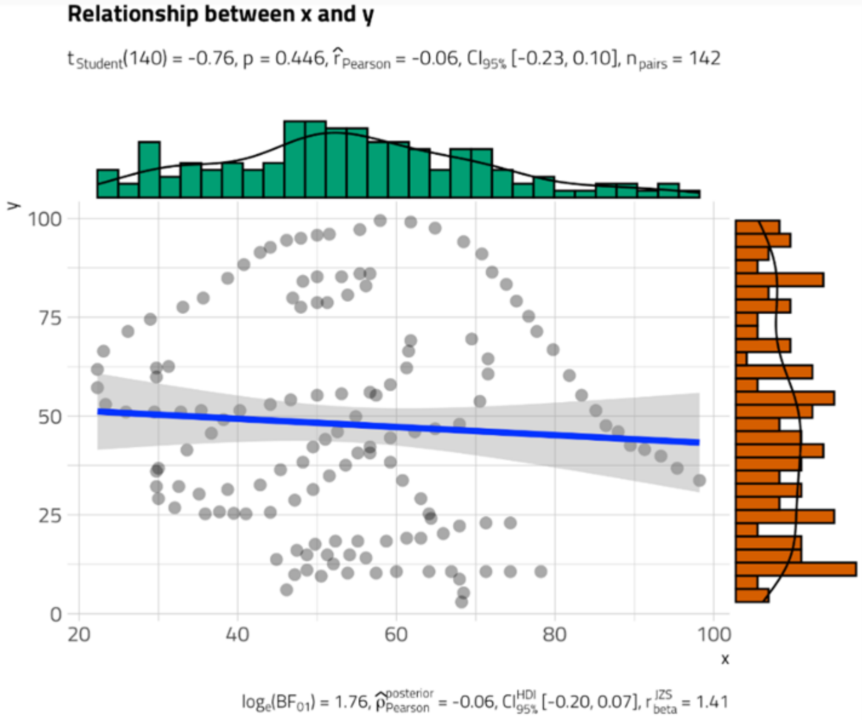
Graphical Summaries can reveal problems not visible from numerical statistics.

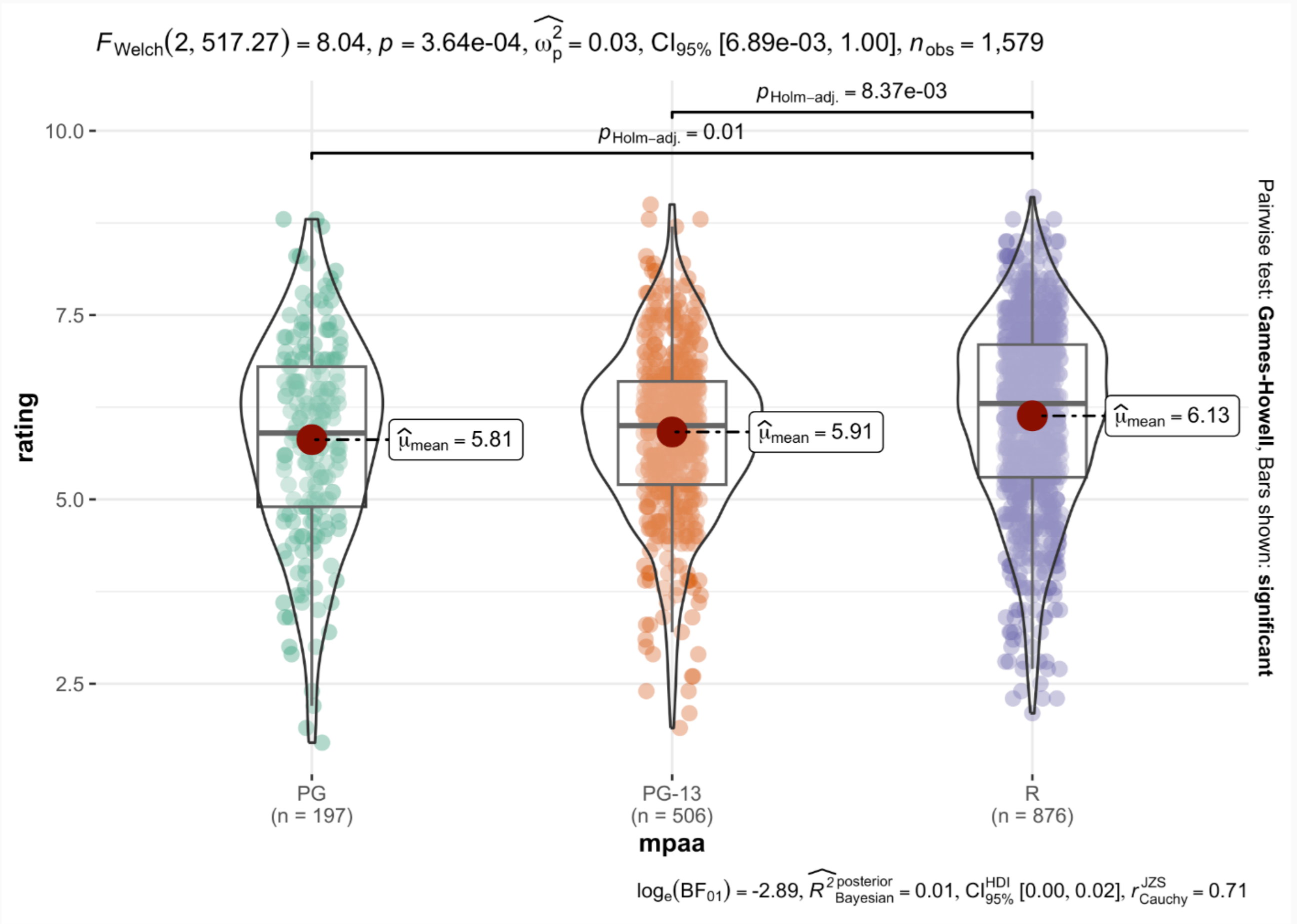
Matejka & Fitzmaurice, Autodesk Research 2017

Standard approach

`ggstatsplot` 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`).





PARIS LODRON UNIVERSITÄT SALZBURG

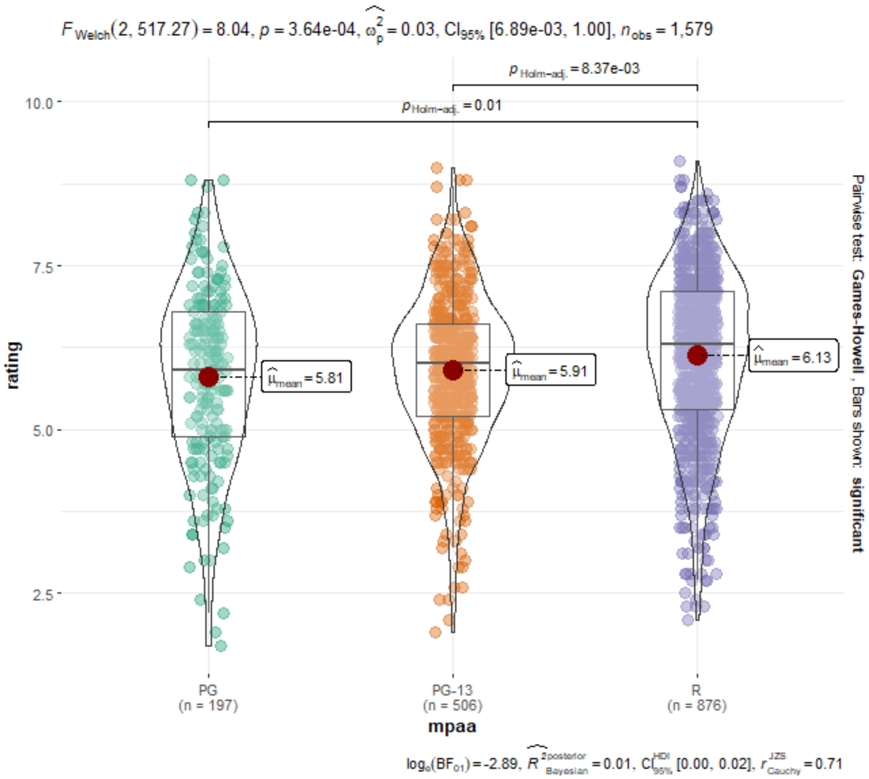
```
ggbetweenstats(
  data = movies_long,
  x = mpaa,
  y = rating
)
```

Function internally decides tests

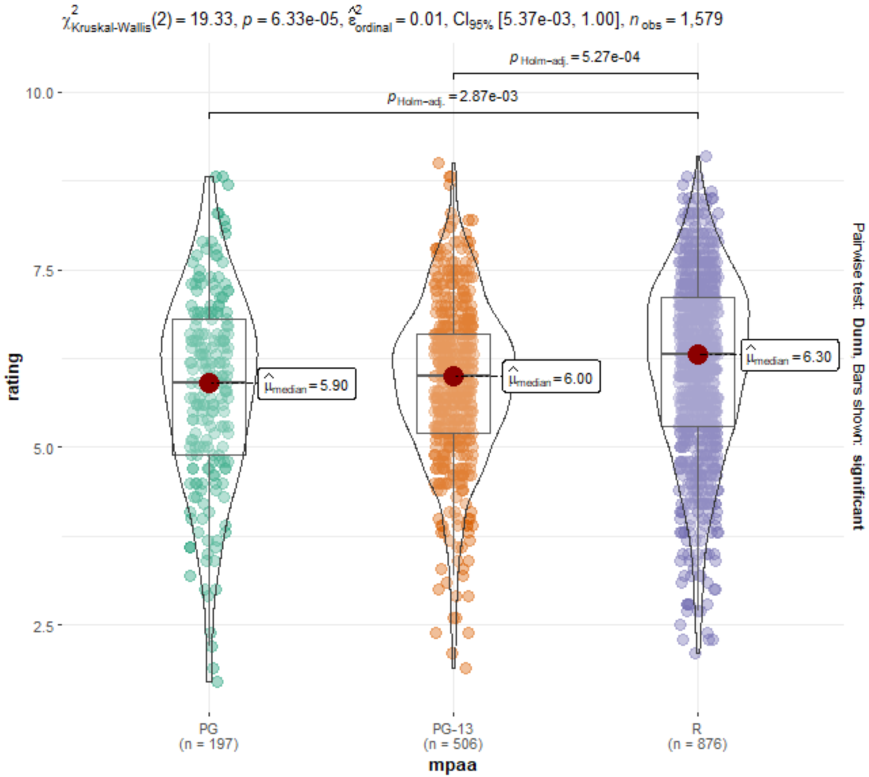
- *t*-test if `2` groups
- ANOVA if `> 2` groups

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

```r
ggbetweenstats(
  data = movies_long,
  x = mpaa,
  y = rating,
  type = "p"
)
```




```r
ggbetweenstats(
  data = movies_long,
  x = mpaa,
  y = rating,
  type = "np"
)
```



PARIS LODRON UNIVERSITÄT SALZBURG

Ready made Plot = No Customization

PARIS LODRON UNIVERSITÄT SALZBURG

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

Need Lot of Customization of Your Choosing

Nice to have a Ready Made Solution in the Exploration Phase

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

Consistent API = No Cognitive Fatigue

PARIS LODRON UNIVERSITÄT SALZBURG

```
stats::lm(formula = wt ~ mpg, data = mtcars)
```

```
stats::cor(x = mtcars$wt, y = mtcars$mpg)
```

```
stats::cor.test(formula = ~ wt + mpg, data = mtcars)
```

Functions in `ggstatsplot` -

✅ expect **dataframe**

✅ expect **tidy** data

✅ have consistent API (`foo(data, x, ...)`)

One Package to Access It All

PARIS LODRON UNIVERSITÄT SALZBURG

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?

Installation

PARIS LODRON UNIVERSITÄT SALZBURG

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)
```

Exercises- Datasets

Starwars Dataset

```r
library(dplyr)
data("starwars")
```

	name	height	mass	hair_color	skin_color	eye_color	birth_year	sex	gender	homeworld	species
1	Luke Skywalker	172	77.0	blond	fair	blue	19.0	male	masculine	Tatooine	Human
2	C-3PO	167	75.0	NA	gold	yellow	112.0	none	masculine	Tatooine	Droid
3	R2-D2	96	32.0	NA	white, blue	red	33.0	none	masculine	Naboo	Droid
4	Darth Vader	202	136.0	none	white	yellow	41.9	male	masculine	Tatooine	Human
5	Leia Organa	150	49.0	brown	light	brown	19.0	female	feminine	Alderaan	Human
6	Owen Lars	178	120.0	brown, grey	light	blue	52.0	male	masculine	Tatooine	Human
7	Beru Whitesun Lars	165	75.0	brown	light	blue	47.0	female	feminine	Tatooine	Human
8	R5-D4	97	32.0	NA	white, red	red	NA	none	masculine	Tatooine	Droid
9	Biggs Darklighter	183	84.0	black	light	brown	24.0	male	masculine	Tatooine	Human
10	Obi-Wan Kenobi	182	77.0	auburn, white	fair	blue-gray	57.0	male	masculine	Stewjon	Human

name: Name of the character.

height: Height of the character in centimeters

mass: Weight of the character in kilograms

hair_color, skin_color, eye_color: Hair, skin, and eye colors of the character

birth_year: Year the character was born. BBY stands for Before Battle of Yavin

sex: The biological sex of the character, namely male, female, hermaphroditic, or none (as in the case for Droids).

gender: The gender role or gender identity of the character as determined by their personality or the way they were programmed.

homeworld: Name of the character's homeworld.

species: Name of the character's species.

films: List of films the character appeared in.

vehicles: List of vehicles the character has piloted.

starships: List of starships the character has piloted.

Where to get the Materials?

Primary Functions

PARIS LODRON UNIVERSITÄT SALZBURG

Hypothesis About Group Differences

📕 Multiple groups

- ggbetweenstats
- ggwithinstats

📕 Single group

- gghistostats
- ggdotplotstats

Hypothesis of composition of categorical variables

- ggpiestats
- ggbarstats

Hypothesis about correlation

📕 Two numeric variables

- ggscatterstats

📕 Multiple numeric variables

- ggcorrmat

Hypothesis about regression coefficients

- ggcoefstats

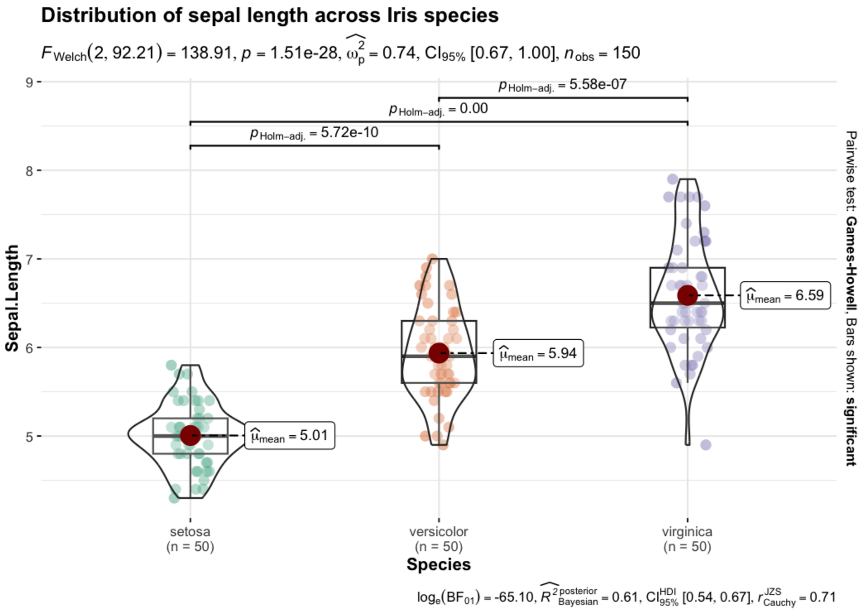
Primary functions

Hypothesis about group differences

📕Multiple groups - ggbetweenstats, ggwithinstats
📕Single group - gghistostats, ggdotplotstats

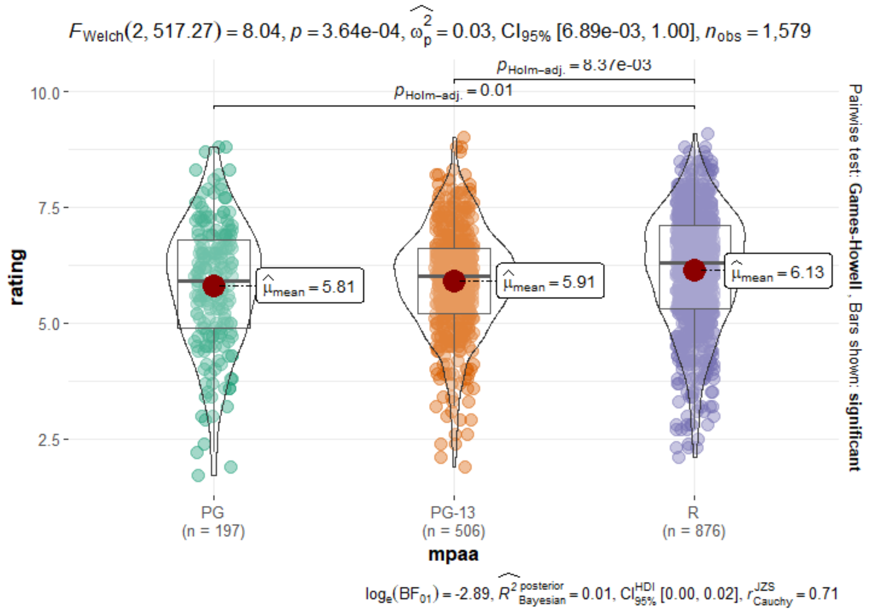
1. ggbetweenstats- for between group comparison

- Used to do data exploration.
- It used when we want to see the distribution of a numeric variable across different levels of a categorical variable.
- It provides a combination of a box plot and/or violin plot, along with the results of a statistical test.
- Statistical Details Included.
- Highly Customizable.
- Publication-Ready Plots.



1. ggbetweenstats- for between group comparison

	title	year	length	budget	rating	votes	mpaa	genre
	<chr>	<int>	<int>	<dbl>	<dbl>	<int>	<fct>	<fct>
1	Shawshank Redemption, The	1994	142	25	9.1	149494	R	Drama
2	Lord of the Rings: The Return of the King, The	2003	251	94	9	103631	PG-13	Action
3	Lord of the Rings: The Fellowship of the Ring, The	2001	208	93	8.8	157608	PG-13	Action
4	Lord of the Rings: The Two Towers, The	2002	223	94	8.8	114797	PG-13	Action
5	Pulp Fiction	1994	168	8	8.8	132745	R	Drama
6	Schindler's List	1993	195	25	8.8	97667	R	Drama
7	Star Wars	1977	125	11	8.8	134640	PG	Action
8	Star Wars: Episode V - The Empire Strikes Back	1980	129	18	8.8	103706	PG	Action
9	C'era una volta il West	1968	158	5	8.7	17241	PG-13	Drama
0	Cidade de Deus	2002	135	3.3	8.7	25964	R	Drama



Defaults return

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

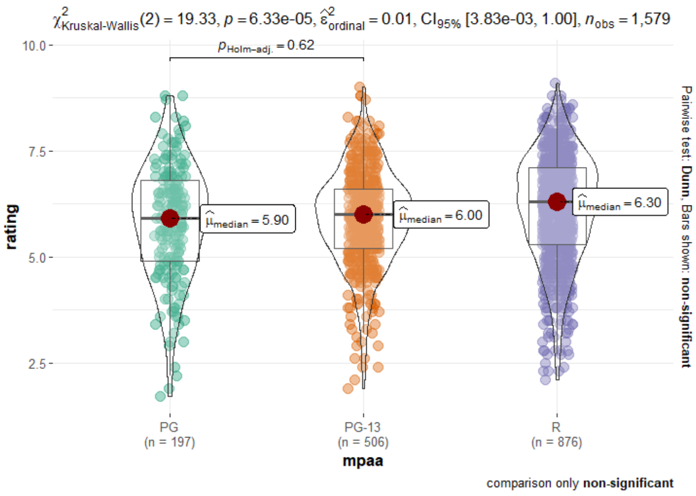
```
ggbetweenstats(
  data = movies_long,
  x = mpaa,
  y = rating
)
```

Function internally decides tests

- T-test if 2 groups
- ANOVA if >2 groups

1. ggbetweenstats- pairwise comparisons

	title	year	length	budget	rating	votes	mpaa	genre
	<chr>	<int>	<int>	<dbl>	<dbl>	<int>	<fct>	<fct>
1	Shawshank Redemption, The	1994	142	25	9.1	149494	R	Drama
2	Lord of the Rings: The Return of the King, The	2003	251	94	9	103631	PG-13	Action
3	Lord of the Rings: The Fellowship of the Ring, The	2001	208	93	8.8	157608	PG-13	Action
4	Lord of the Rings: The Two Towers, The	2002	223	94	8.8	114797	PG-13	Action
5	Pulp Fiction	1994	168	8	8.8	132745	R	Drama
6	Schindler's List	1993	195	25	8.8	97667	R	Drama
7	Star Wars	1977	125	11	8.8	134640	PG	Action
8	Star Wars: Episode V - The Empire Strikes Back	1980	129	18	8.8	103706	PG	Action
9	C'era una volta il West	1968	158	5	8.7	17241	PG-13	Drama
0	Cidade de Deus	2002	135	3.3	8.7	25964	R	Drama




```r
ggbetweenstats(
  data = movies_long,
  x = mpaa,
  y = rating,
  type = "np",
  pairwise.display = "ns",
  caption = expression(
    paste("comparison only ",
    bold("non-significant")))
)
```

Changing the type of test

- "p" → **Parametric** → μ_{mean}
- "np" → **non - parametric** → μ_{median}
- "r" → **robust** → $\mu_{trimmed}$
- "bf" → **Bayesian** → μ_{MAP}

Changing pairwise comparisons displayed

- "ns" → only **non-significant**
- "s" → only **significant**
- "all" → **all**

2. ggbetweenstats- Arguments

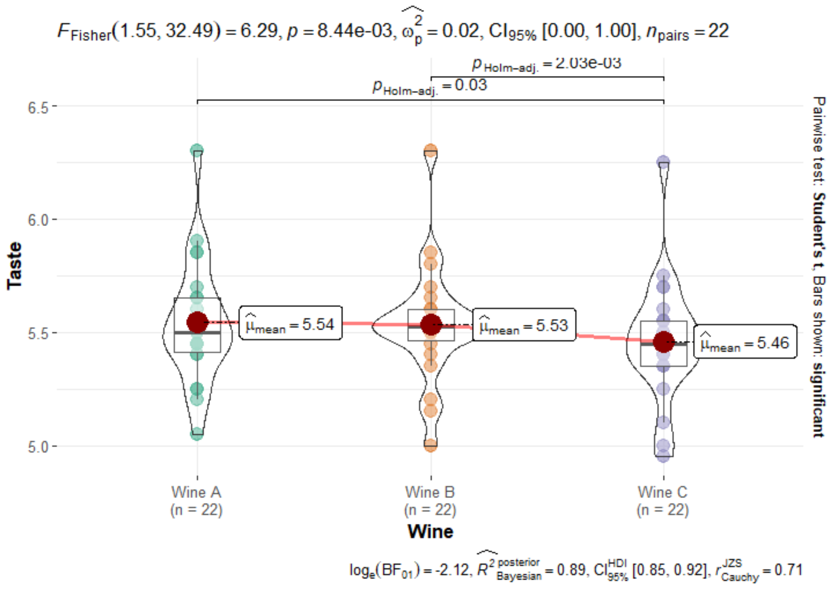
```
ggbetweenstats(
  data,
  x,
  y,
  type = "parametric",
  pairwise.display = "significant",
  p.adjust.method = "holm",
  effsize.type = "unbiased",
  bf.prior = 0.707,
  bf.message = TRUE,
  results.subtitle = TRUE,
  xlab = NULL,
  ylab = NULL,
  caption = NULL,
  title = NULL,
  subtitle = NULL,
  digits = 2L,
  var.equal = FALSE,
  conf.level = 0.95,
  nboot = 100L,
```

```
tr = 0.2,
centrality.plotting = TRUE,
centrality.type = type,
centrality.point.args = list(size = 5, color = "darkred"),
centrality.label.args = list(size = 3, nudge_x = 0.4, segment.linetype = 4,
  min.segment.length = 0),
point.args = list(position = ggplot2::position_jitterdodge(dodge.width = 0.6),
  0.4, size = 3, stroke = 0, na.rm = TRUE),
boxplot.args = list(width = 0.3, alpha = 0.2, na.rm = TRUE),
violin.args = list(width = 0.5, alpha = 0.2, na.rm = TRUE),
ggsignif.args = list(textsize = 3, tip_length = 0.01, na.rm = TRUE),
ggtheme = ggstatsplot::theme_ggstatsplot(),
package = "RColorBrewer",
palette = "Dark2",
ggplot.component = NULL,
```

https://indrajeetpatil.github.io/ggstatsplot/reference/ggbetweenstats.html

2. ggwithinstats - repeated measures equivalent

- Used to do data exploration.
- It is used when we want to see the distribution of a numeric variable across different levels of a categorical variable within the same subject.
- It provides a combination of a box plot and/or violin plot, along with the results of a statistical test.
- Statistical Details Included.
- Highly Customizable.
- Publication-Ready Plots.



2. ggwithinstats

Defaults return

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

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

Function internally decides tests

- T-test if 2 groups
- ANOVA if >2 groups

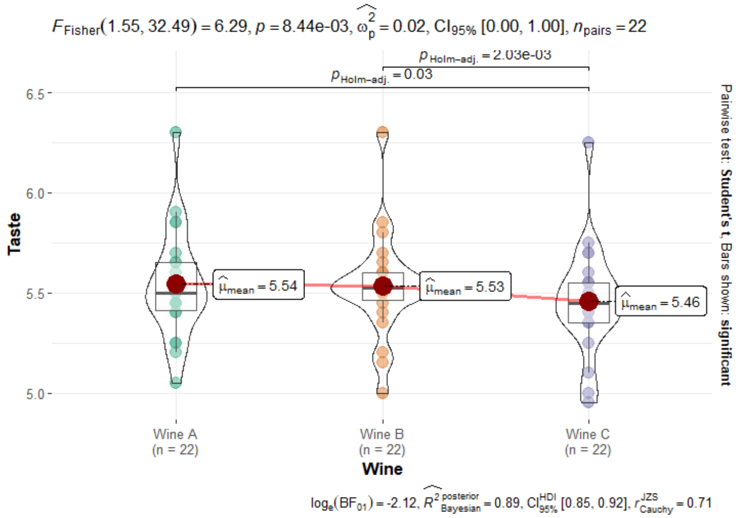
```
> WRS2::WineTasting
   Taste   Wine Taster
1   5.40 Wine A      1
2   5.50 Wine B      1
3   5.55 Wine C      1
4   5.85 Wine A      2
5   5.70 Wine B      2
6   5.75 Wine C      2
7   5.20 Wine A      3
8   5.60 Wine B      3
9   5.50 Wine C      3
10  5.55 Wine A      4
11  5.50 Wine B      4
```

Changing the type of test

- ✅ "p" → **Parametric** → μ_{mean}
- ✅ "np" → **non - parametric** → μ_{median}
- ✅ "r" → **robust** → $\mu_{trimmed}$
- ✅ "bf" → **Bayesian** → μ_{MAP}

Changing pairwise comparisons displayed

- ✅ "ns" → only **non-significant**
- ✅ "s" → only **significant**
- ✅ "all" → **all**



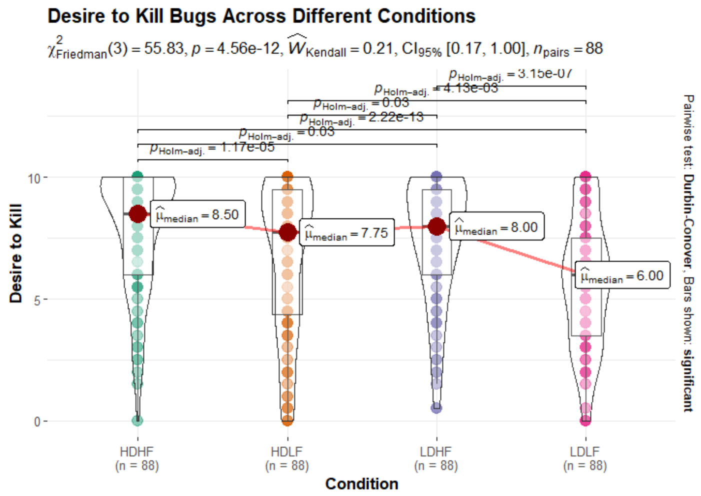

```
> dplyr::filter(WRS2::WineTasting, Taster == "1")
  Taste   Wine Taster
1  5.40 Wine A      1
2  5.50 Wine B      1
3  5.55 Wine C      1
> dplyr::filter(WRS2::WineTasting, Taster == "2")
  Taste   Wine Taster
1  5.85 Wine A      2
2  5.70 Wine B      2
3  5.75 Wine C      2
```

2. ggwithinstats - Example

Question: How does the desire to kill bugs vary across different conditions (LDLF, LDHF, HDLF, HDHF) within the same individual in the bugs_long dataset?

```
> bugs_long
# A tibble: 372 × 6
  subject gender region        education condition desire
    <int> <fct>  <fct>         <fct>     <chr>      <dbl>
1       1 Female North America some      LDLF           6
2       2 Female North America advance   LDLF          10
3       3 Female Europe        college   LDLF           5
4       4 Female North America college   LDLF           6
5       5 Female North America some      LDLF           3
6       6 Female Europe        some      LDLF           2
```

```
ggwithinstats(
  data = bugs_long, # data frame
  x = condition, # categorical variable(within-subjects factor)
  y = desire, # numeric variable
  title = "Desire to Kill Bugs Across Different Conditions",
  xlab = "Condition",
  ylab = "Desire to Kill",
  type = "np", # non-parametric test
  pairwise.comparisons = TRUE, # show pairwise comparisons
  pairwise.display = "significant", # only show significant comparisons
  p.adjust.method = "holm" # method for adjusting p values
)
```




```
> dplyr::filter(bugs_long, subject == 1)
# A tibble: 4 × 6
  subject gender region        education condition desire
    <int> <fct>  <fct>         <fct>     <chr>      <dbl>
1       1 Female North America some      LDLF           6
2       1 Female North America some      LDHF           6
3       1 Female North America some      HDLF           9
4       1 Female North America some      HDHF          10
```

2. ggwithinstats - Arguments

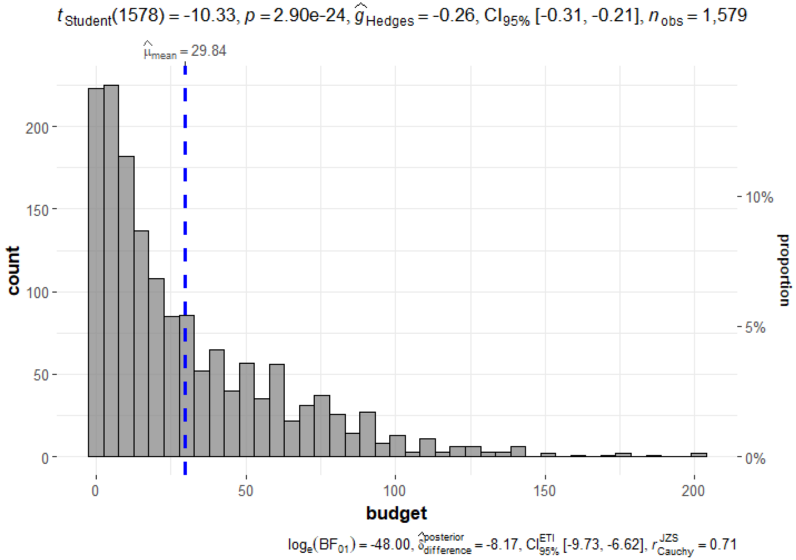
```
ggwithinstats(
  data,
  x,
  y,
  type = "parametric",
  pairwise.display = "significant",
  p.adjust.method = "holm",
  effsize.type = "unbiased",
  bf.prior = 0.707,
  bf.message = TRUE,
  results.subtitle = TRUE,
  xlab = NULL,
  ylab = NULL,
  caption = NULL,
  title = NULL,
  subtitle = NULL,
  digits = 2L,
  conf.level = 0.95,
  nboot = 100L,
  tr = 0.2,
  centrality.plotting = TRUE,
```

```
centrality.type = type,
centrality.point.args = list(size = 5, color = "darkred"),
centrality.label.args = list(size = 3, nudge_x = 0.4, segment.linetype = 4),
centrality.path = TRUE,
centrality.path.args = list(linewidth = 1, color = "red", alpha = 0.5),
point.args = list(size = 3, alpha = 0.5, na.rm = TRUE),
point.path = TRUE,
point.path.args = list(alpha = 0.5, linetype = "dashed"),
boxplot.args = list(width = 0.2, alpha = 0.5, na.rm = TRUE),
violin.args = list(width = 0.5, alpha = 0.2, na.rm = TRUE),
ggsignif.args = list(textsize = 3, tip_length = 0.01, na.rm = TRUE),
ggtheme = ggstatsplot::theme_ggstatsplot(),
package = "RColorBrewer",
palette = "Dark2",
ggplot.component = NULL,
```

https://indrajeetpatil.github.io/ggstatsplot/reference/ggwithinstats.html

3. gghistostats - Distribution of a numerical variable

- It is used for data exploration.
- It used to inspect distribution of a continuous variable.
- It can test its mean is significantly different from a specified value with a one-sample test.
- Publication-Ready histograms.



3. gghistostats - Distribution of a numerical variable

	title	year	length	budget	rating	votes	mpaa	genre
	<chr>	<int>	<int>	<dbl>	<dbl>	<int>	<fct>	<fct>
1	Shawshank Redemption, The	1994	142	25	9.1	149494	R	Drama
2	Lord of the Rings: The Return of the King, The	2003	251	94	9	103631	PG-13	Action
3	Lord of the Rings: The Fellowship of the Ring, The	2001	208	93	8.8	157608	PG-13	Action
4	Lord of the Rings: The Two Towers, The	2002	223	94	8.8	114797	PG-13	Action
5	Pulp Fiction	1994	168	8	8.8	132745	R	Drama
6	Schindler's List	1993	195	25	8.8	97667	R	Drama
7	Star Wars	1977	125	11	8.8	134640	PG	Action
8	Star Wars: Episode V - The Empire Strikes Back	1980	129	18	8.8	103706	PG	Action
9	C'era una volta il West	1968	158	5	8.7	17241	PG-13	Drama
10	Cidade de Deus	2002	135	3.3	8.7	25964	R	Drama

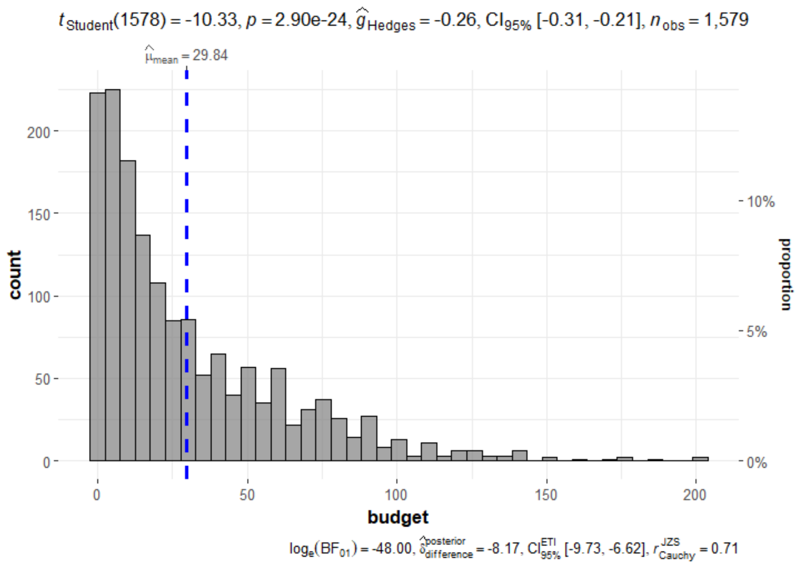
Defaults return

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

Centrality measures

- "p" → μ_{mean}
- "np" → μ_{median}
- "r" → $\mu_{trimmed}$
- "bf" → μ_{MAP}

```
gghistostats(
  data = movies_long,  # data frame
  x = budget,  #numerical variable create histogram
  test.value = 38  #the value that want to compare
                   # the mean of sample (budget).
)
```

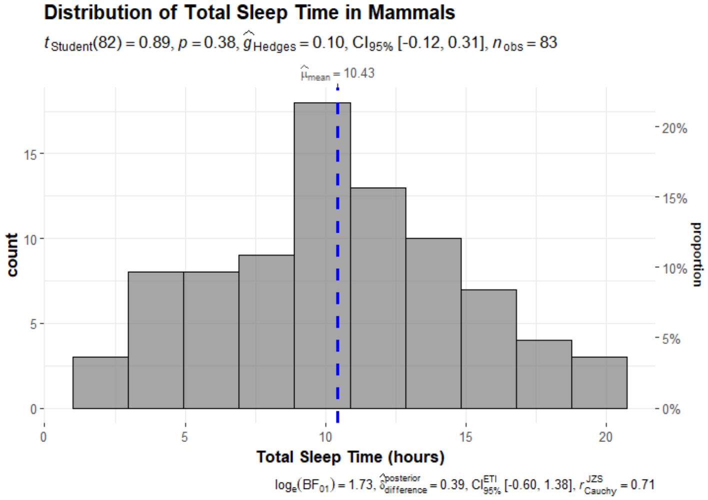


3. gghistostats - Example

Question: What is the distribution of total sleep time across all mammal species in the msleep dataset, and is the average total sleep time significantly different from 10 hours in msleep dataset?

```
> ggplot2::msleep
# A tibble: 83 × 11
   name                  genus vore  order conservation sleep_total sleep_rem sleep_cycle awake  brainwt  bodywt
   <chr>                 <chr> <chr> <chr> <chr>              <dbl>     <dbl>       <dbl> <dbl>    <dbl>   <dbl>
 1 Cheetah               Acin… carni Carn… lc                  12.1      NA        NA      11.9  NA        50
 2 Owl monkey            Aotus omni  Prim… NA                  17         1.8      NA       7     0.0155    0.48
 3 Mountain beaver       Aplo… herbi Rode… nt                  14.4       2.4      NA       9.6  NA         1.35
 4 Greater short-ta…     Blar… omni  Sori… lc                  14.9       2.3       0.133   9.1   0.00029   0.019
 5 Cow                   Bos   herbi Arti… domesticated         4         0.7       0.667  20     0.423   600
 6 Three-toed sloth      Brad… herbi Pilo… NA                  14.4       2.2       0.767   9.6  NA         3.85
 7 Northern fur seal     Call… carni Carn… vu                   8.7       1.4       0.383  15.3  NA        20.5
 8 Vesper mouse          Calo… NA    Rode… NA                   7        NA        NA      17    NA         0.045
 9 Dog                   Canis carni Carn… domesticated        10.1       2.9       0.333  13.9   0.07     14
10 Roe deer              Capr… herbi Arti… lc                   3        NA        NA      21     0.0982   14.8
```

```
gghistostats(
  data = msleep, # data frame
  x = sleep_total, # numeric variable
  test.value = 10, # test value
  type = "parametric", # type of statistical approach
  title = "Distribution of Total Sleep Time in Mammals", # title for the plot
  xlab = "Total Sleep Time (hours)" # x-axis label
)
```



3. gghistostats - Arguments

PARIS LODRON UNIVERSITÄT SALZBURG

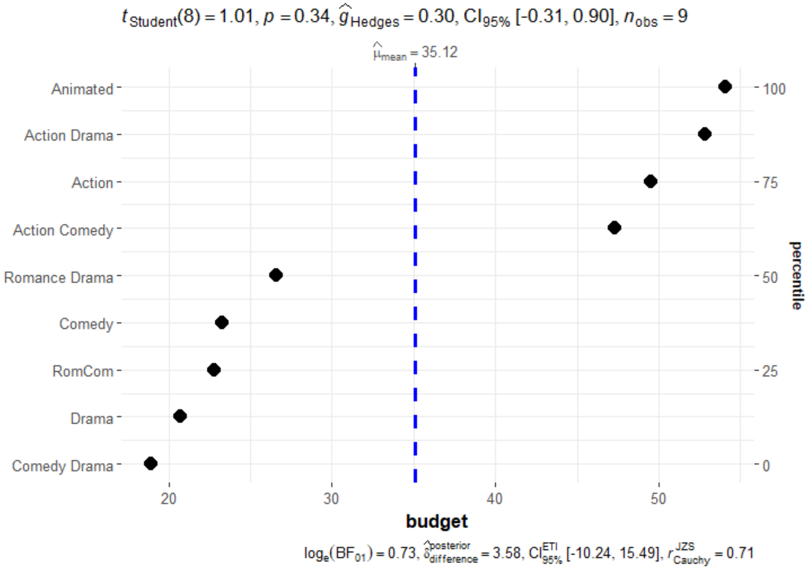
```
gghistostats(
  data,
  x,
  binwidth = NULL,
  xlab = NULL,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  type = "parametric",
  test.value = 0,
  bf.prior = 0.707,
  bf.message = TRUE,
  effsize.type = "g",
  conf.level = 0.95,
```

```
  tr = 0.2,
  digits = 2L,
  ggtheme = ggstatsplot::theme_ggstatsplot(),
  results.subtitle = TRUE,
  bin.args = list(color = "black", fill = "grey50", alpha = 0.7),
  centrality.plotting = TRUE,
  centrality.type = type,
  centrality.line.args = list(color = "blue", linewidth = 1, linetype = "dashed")
  normal.curve = FALSE,
  normal.curve.args = list(linewidth = 2),
  ggplot.component = NULL,
```

https://indrajeetpatil.github.io/ggstatsplot/reference/gghistostats.html

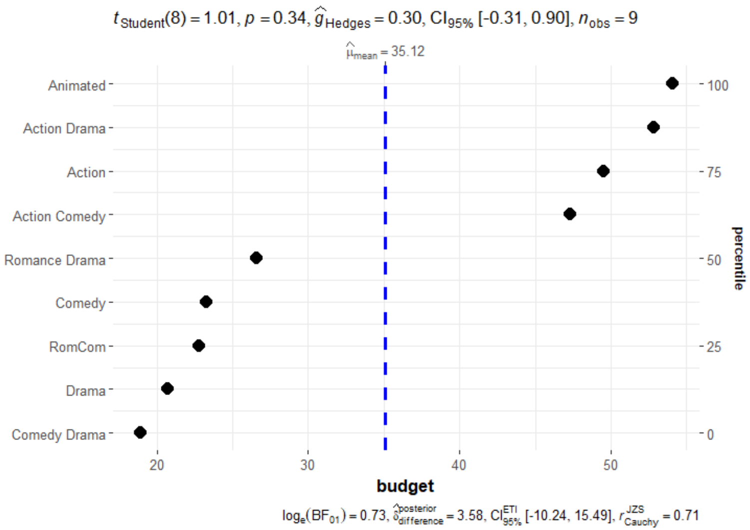
4. ggdotplotstats - Labeled Numerical Variable

- It is used for data exploration.
- A dot chart with statistical details from one-sample test.
- It used to inspect the distribution of a labeled numeric variable
- This function is a sister function of gghistostats with the difference being it expects a labeled numeric variable.
- Highly Customizable.
- Publication-Ready histograms.



4. ggdotplotstats - Labeled Numerical Variable

	title	year	length	budget	rating	votes	mpaa	genre
	<chr>	<int>	<int>	<dbl>	<dbl>	<int>	<fct>	<fct>
1	Shawshank Redemption, The	1994	142	25	9.1	149494	R	Drama
2	Lord of the Rings: The Return of the King, The	2003	251	94	9	103631	PG-13	Action
3	Lord of the Rings: The Fellowship of the Ring, The	2001	208	93	8.8	157608	PG-13	Action
4	Lord of the Rings: The Two Towers, The	2002	223	94	8.8	114797	PG-13	Action
5	Pulp Fiction	1994	168	8	8.8	132745	R	Drama
6	Schindler's List	1993	195	25	8.8	97667	R	Drama
7	Star Wars	1977	125	11	8.8	134640	PG	Action
8	Star Wars: Episode V - The Empire Strikes Back	1980	129	18	8.8	103706	PG	Action
9	C'era una volta il West	1968	158	5	8.7	17241	PG-13	Drama
0	Cidade de Deus	2002	135	3.3	8.7	25964	R	Drama



Defaults return

- ☑ descriptives (mean + sample size)
- ☑ inferential statistics
- ☑ effect size + CIs
- ☑ Bayesian hypothesis-testing
- ☑ Bayesian estimation

Centrality measures

- ☑ "p" → μ_{mean}
- ☑ "np" → μ_{median}
- ☑ "r" → $\mu_{trimmed}$
- ☑ "bf" → μ_{MAP}

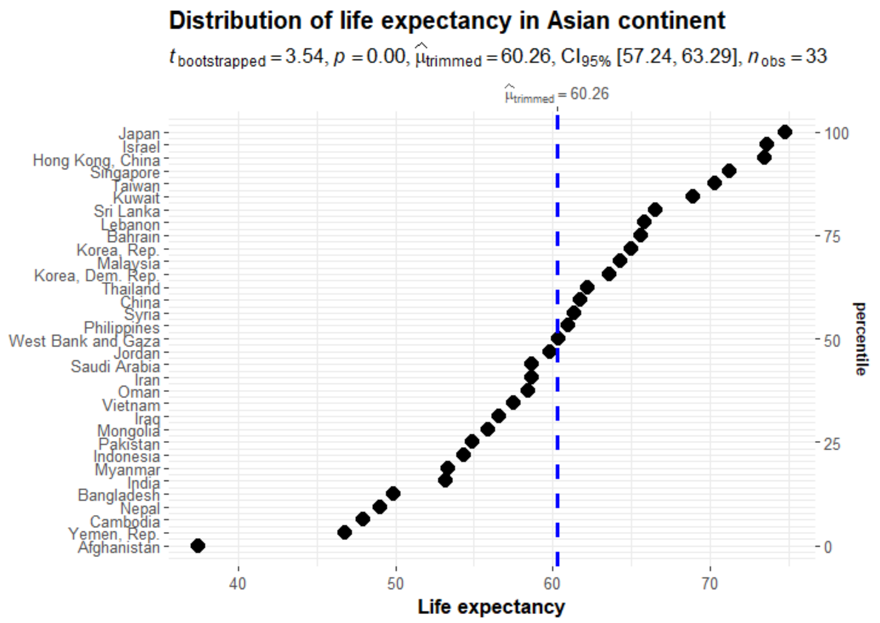
```
ggdotplotstats(
  data       = movies_long, # data frame
  x          = budget,  # numeric variable want to create dot plot
  y          = genre,  # the label
  test.value = 30  # the value test its mean is significantly
                   # different from a specified value with
                   # a one-sample test.
)
```

4. ggdotplotstats - Example

Question: What is the distribution of life expectancy across all Asian countries in the gapminder dataset, and is the average life expectancy significantly different from 55 years using a robust statistical approach?

```
> gapminder::gapminder
# A tibble: 1,704 × 6
   country     continent  year lifeExp      pop gdpPercap
   <fct>       <fct>     <int>   <dbl>    <int>     <dbl>
 1 Afghanistan Asia       1952    28.8  8425333      779.
 2 Afghanistan Asia       1957    30.3  9240934      821.
 3 Afghanistan Asia       1962    32.0 10267083      853.
 4 Afghanistan Asia       1967    34.0 11537966      836.
 5 Afghanistan Asia       1972    36.1 13079460      740.
 6 Afghanistan Asia       1977    38.4 14880372      786.
```

```
ggdotplotstats(
  data       = dplyr::filter(gapminder::gapminder, continent == "Asia"),
  y          = country,
  x          = lifeExp,
  test.value = 55,
  type       = "robust",
  title      = "Distribution of life expectancy in Asian continent",
  xlab       = "Life expectancy"
)
```



3. ggdotplotstats - Arguments

```
ggdotplotstats(
  data,
  x,
  y,
  xlab = NULL,
  ylab = NULL,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  type = "parametric",
  test.value = 0,
  bf.prior = 0.707,
  bf.message = TRUE,
  effsize.type = "g",
  conf.level = 0.95,
```

```
  tr = 0.2,
  digits = 2L,
  results.subtitle = TRUE,
  point.args = list(color = "black", size = 3, shape = 16),
  centrality.plotting = TRUE,
  centrality.type = type,
  centrality.line.args = list(color = "blue", linewidth = 1, linetype = "dashed"),
  ggplot.component = NULL,
  ggtheme = ggstatsplot::theme_ggstatsplot(),
```

https://indrajeetpatil.github.io/ggstatsplot/reference/ggdotplotstats.html

Exercise 1

Load the starwars dataset.

1.1 Plot the distribution of height height across gender (feminine, masculine) Star Wars characters, considering only those characters that have a weight greater than 50?

1.2 Plot the distribution of mass among Star Wars characters who appear in ‘The Empire Strikes Back’ film and have brown hair, and is the average mass significantly different from 50 in this subset of characters?

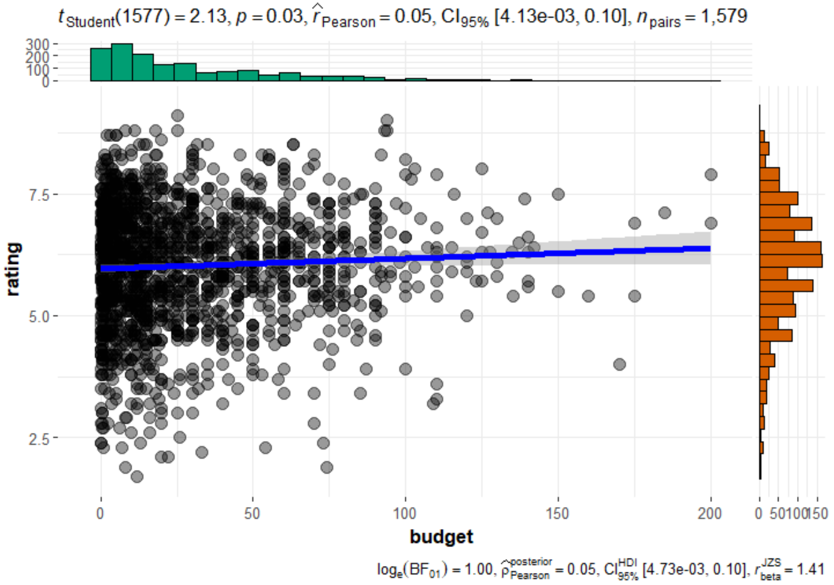
1.3 Plot the distribution of height among Star Wars characters for each eye color, and is the average height significantly different from 180 cm for each eye color group?

Hypothesis about correlation

📕Two numeric variables - ggscatterstats
📕Multiple numeric variables - ggcorrmat

5. ggscatterstats- Two numeric variables

- It is used for data exploration.
- It used to check linear association between two continuous variables.
- It used to check distribution of two continuous variables.
- Highly Customizable.
- Publication-Ready scatterplot with all statistical details included in the plot itself.



5. ggscatterstats- Two numeric variables

```
   title                                            year length budget rating  votes mpaa
   <chr>                                           <int>  <int>  <dbl>  <dbl>  <int> <fct>
 1 Shawshank Redemption, The                        1994    142     25    9.1 149494 R
 2 Lord of the Rings: The Return of the King, The   2003    251     94    9   103631 PG-13
 3 Lord of the Rings: The Fellowship of the Ring, The 2001  208     93    8.8 157608 PG-13
 4 Lord of the Rings: The Two Towers, The           2002    223     94    8.8 114797 PG-13
 5 Pulp Fiction                                     1994    168      8    8.8 132745 R
 6 Schindler's List                                 1993    195     25    8.8  97667 R
 7 Star Wars                                        1977    125     11    8.8 134640 PG
 8 Star Wars: Episode V - The Empire Strikes Back   1980    129     18    8.8 103706 PG
 9 C'era una volta il West                          1968    158      5    8.7  17241 PG-13
 0 Cidade de Deus                                   2002    135      3.3  8.7  25964 R
```

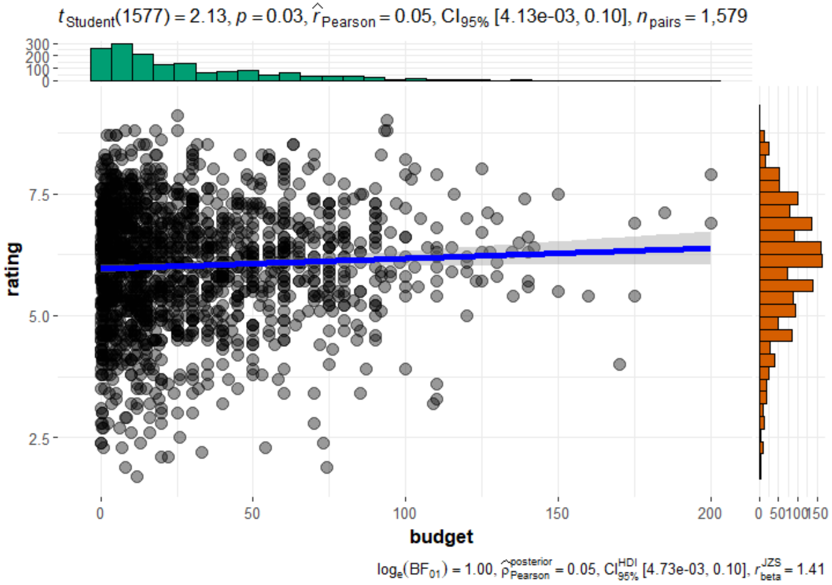
Defaults return

- raw data + distributions
- marginal distributions
- inferential statistics
- effect size + CIs
- Bayesian hypothesis-testing
- Bayesian estimation

Changing the type of test

- "p" → **Parametric** → μ_{mean}
- "np" → **non - parametric** → μ_{median}
- "r" → **robust** → $\mu_{trimmed}$
- "bf" → **Bayesian** → μ_{MAP}

```
ggscatterstats (
  data = movies_long,  # data frame
  x = budget,  # numerical variable
  y = rating  # numerical variable
)
```

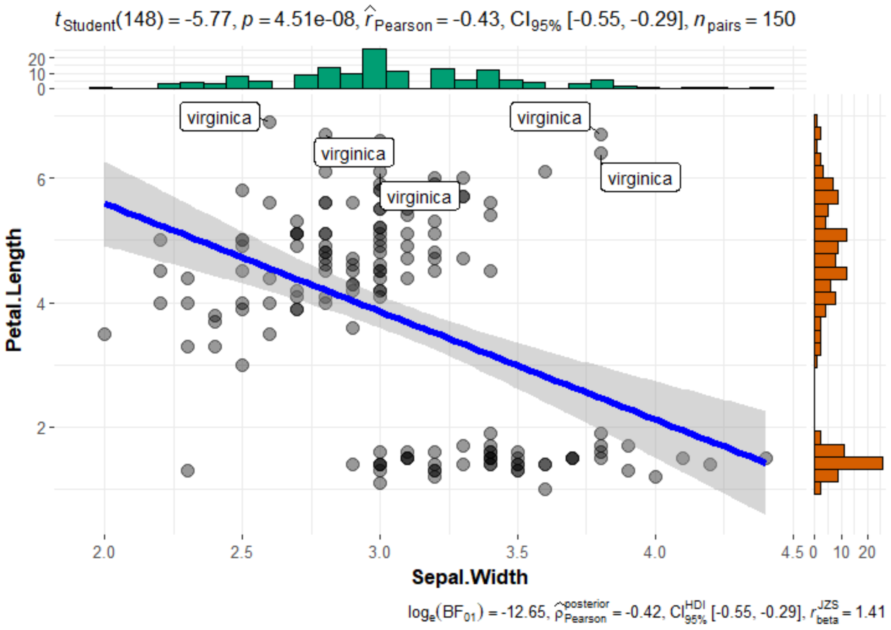


5. ggscatterstats - Example

Question: What is the relationship between Sepal Width and Petal Length in the iris dataset, and are there any specific species of iris where the Sepal Length is greater than 7.6?

```
> iris
  Sepal.Length Sepal.Width Petal.Length Petal.Width  Species
1          5.1         3.5          1.4         0.2   setosa
2          4.9         3.0          1.4         0.2   setosa
3          4.7         3.2          1.3         0.2   setosa
4          4.6         3.1          1.5         0.2   setosa
5          5.0         3.6          1.4         0.2   setosa
6          5.4         3.9          1.7         0.4   setosa
7          4.6         3.4          1.4         0.3   setosa
8          5.0         3.4          1.5         0.2   setosa
9          4.4         2.9          1.4         0.2   setosa
```

```
ggscatterstats(
  data = iris,  #data frame
  x = Sepal.Width, #numeric variable for the x-axis of the scatterplot
  y = Petal.Length,#numeric variable for the y-axis of the scatterplot
  label.var = Species,#variable used for labeling the points
                      #in the scatterplot.
  label.expression = Sepal.Length > 7.6 #an expression used to determine
                                 #which points to label in the scatterplot
)
```



3. ggscatterstats - Arguments

PARIS LODRON UNIVERSITÄT SALZBURG

```
ggscatterstats(
  data,
  x,
  y,
  type = "parametric",
  conf.level = 0.95,
  bf.prior = 0.707,
  bf.message = TRUE,
  tr = 0.2,
  digits = 2L,
  results.subtitle = TRUE,
  label.var = NULL,
  label.expression = NULL,
  marginal = TRUE,
  point.args = list(size = 3, alpha = 0.4, stroke = 0),
```

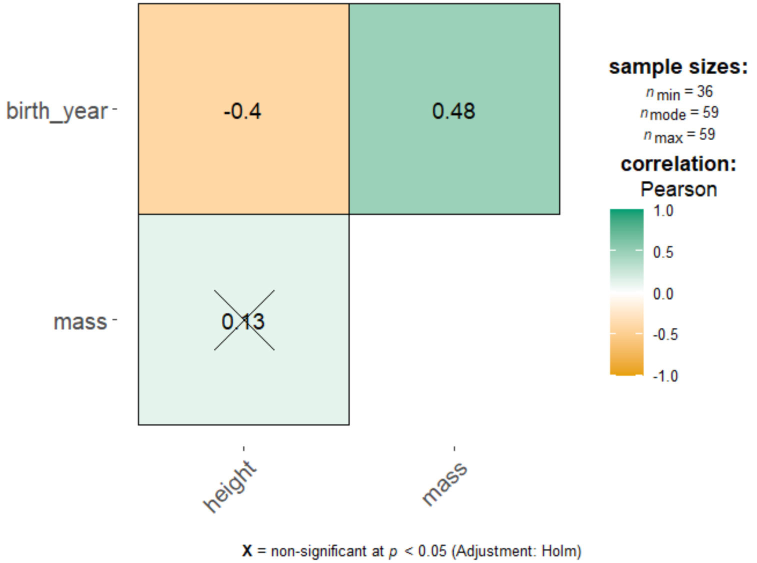
```
point.width.jitter = 0,
point.height.jitter = 0,
point.label.args = list(size = 3, max.overlaps = 1e+06),
smooth.line.args = list(linewidth = 1.5, color = "blue", method = "lm", formula
  x),
xsidehistogram.args = list(fill = "#009E73", color = "black", na.rm = TRUE),
ysidehistogram.args = list(fill = "#D55E00", color = "black", na.rm = TRUE),
xlab = NULL,
ylab = NULL,
title = NULL,
subtitle = NULL,
caption = NULL,
ggtheme = ggstatsplot::theme_ggstatsplot(),
ggplot.component = NULL,
```

https://indrajeetpatil.github.io/ggstatsplot/reference/ggscatterstats.html

6. ggcorrmat- Multiple Numeric Variables

PARIS LODRON UNIVERSITÄT SALZBURG

- It is used for data exploration.
- Can be used to get a correlation coefficient matrix or the associated p-value matrix
- Quickly explore correlation between (all) numeric variables in the dataset.
- Highly Customizable.
- Publication-Ready correlation matrix with all statistical details included in the plot itself.



6. ggcorrmat- Multiple Numeric Variables

	title	year	length	budget	rating	votes	mpaa	genre
	<chr>	<int>	<int>	<dbl>	<dbl>	<int>	<fct>	<fct>
1	Shawshank Redemption, The	1994	142	25	9.1	149494	R	Drama
2	Lord of the Rings: The Return of the King, The	2003	251	94	9	103631	PG-13	Action
3	Lord of the Rings: The Fellowship of the Ring, The	2001	208	93	8.8	157608	PG-13	Action
4	Lord of the Rings: The Two Towers, The	2002	223	94	8.8	114797	PG-13	Action
5	Pulp Fiction	1994	168	8	8.8	132745	R	Drama
6	Schindler's List	1993	195	25	8.8	97667	R	Drama
7	Star Wars	1977	125	11	8.8	134640	PG	Action
8	Star Wars: Episode V - The Empire Strikes Back	1980	129	18	8.8	103706	PG	Action
9	C'era una volta il West	1968	158	5	8.7	17241	PG-13	Drama
0	Cidade de Deus	2002	135	3.3	8.7	25964	R	Drama

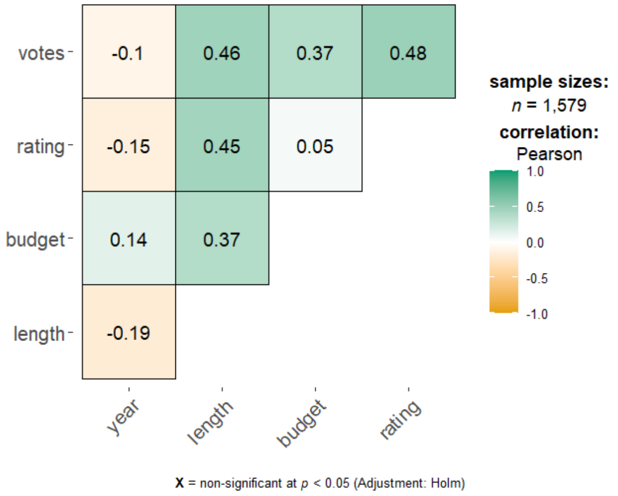
Defaults return

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

Changing the type of test

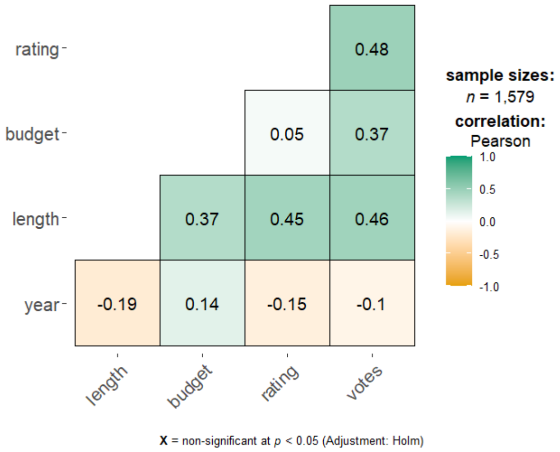
- ✅ "p" → **Parametric** → μ_{mean}
- ✅ "np" → **non - parametric** → μ_{median}
- ✅ "r" → **robust** → $\mu_{trimmed}$
- ✅ "bf" → **Bayesian** → μ_{MAP}

```
ggcorrmat(
  data = movies_long
)
```

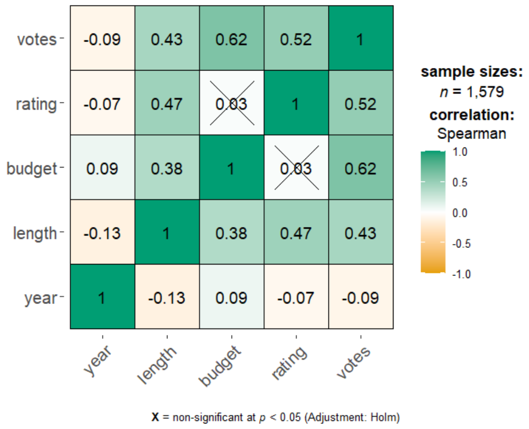


6. ggcorrmat - Multiple Numeric Variables

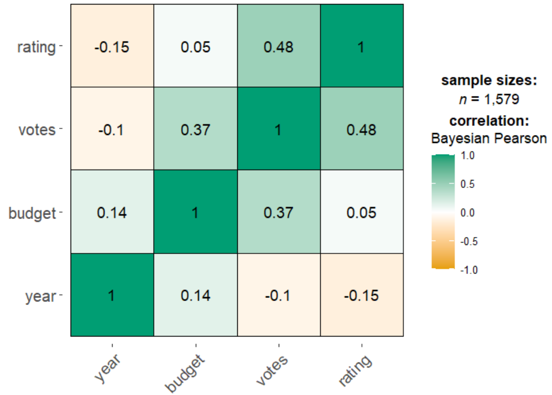
```
ggcorrmat(
  data = movies_long,
  matrix.type = "lower"
)
```




```
ggcorrmat(
  data = movies_long,
  matrix.type = "full",
  type = "np"
)
```




```
ggcorrmat(
  data = movies_long,
  matrix.type = "full",
  type = "bf",
  cor.vars = c(year, budget, votes, rating)
)
```



6. ggcorrmat - Example

Question: show the the correlation matrix between life expectancy and GDP per capita for all countries in the gapminder dataset for the year 2007?

```
> gapminder::gapminder
# A tibble: 1,704 x 6
   country     continent  year lifeExp      pop gdpPercap
   <fct>       <fct>     <int>   <dbl>    <int>     <dbl>
 1 Afghanistan Asia       1952    28.8  8425333      779.
 2 Afghanistan Asia       1957    30.3  9240934      821.
 3 Afghanistan Asia       1962    32.0 10267083      853.
 4 Afghanistan Asia       1967    34.0 11537966      836.
 5 Afghanistan Asia       1972    36.1 13079460      740.

## select data only from the year 2007
gapminder_2007 <- dplyr::filter(gapminder::gapminder, year == 2007)

## producing the correlation matrix
ggcorrmat(
  data = gapminder_2007, ## data from which variable is to be taken
  matrix.type = "full",
  cor.vars = c(lifeExp,gdpPercap) ## specifying correlation matrix variable:
)
```

3. ggcorrmat - Arguments

PARIS LODRON UNIVERSITÄT SALZBURG

```
ggcorrmat(
  data,
  cor.vars = NULL,
  cor.vars.names = NULL,
  matrix.type = "upper",
  type = "parametric",
  tr = 0.2,
  partial = FALSE,
  digits = 2L,
  sig.level = 0.05,
  conf.level = 0.95,
  bf.prior = 0.707,
  p.adjust.method = "holm",
  pch = "cross",
  ggcorrplot.args = list(method = "square", outline.color = "black", pch.cex = 14
  package = "RColorBrewer",
  palette = "Dark2",
  colors = c("#E69F00", "white", "#009E73"),
  ggtheme = ggstatsplot::theme_ggstatsplot(),
  ggplot.component = NULL,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
```

https://indrajeetpatil.github.io/ggstatsplot/reference/ggcorrmat.html

Exercise 2

Load the starwars dataset.

2.1 Plot to determine if the average weight for male human species has any relationship to their height?

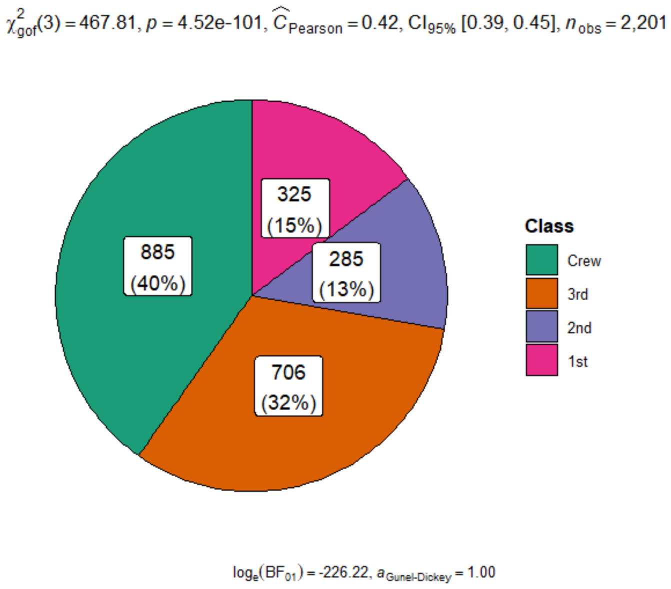
2.2 Plot the correlation matrix of numerical variables (height, mass, birth_year) for characters in the starwars separately for type argument "parametric" and "robust"

Hypothesis of composition of categorical variables

📕ggpiestats
📕ggbarstats

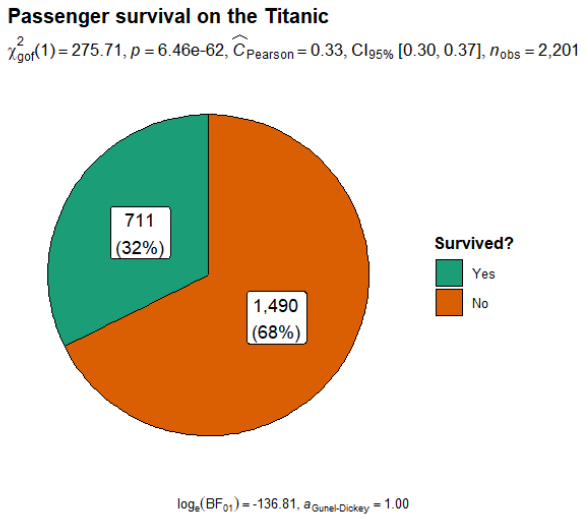
7. ggpiestats - association between categorical variables

- It is used for data exploration.
- Provide pie charts to summarize the statistical relationship(s) among one or more categorical variables.
- Can be used to check goodness of fit.
- To see if the frequency distribution of two categorical variables are independent of each other using the contingency table analysis
- To check if the proportion of observations at each level of a categorical variable is equal
- Highly Customizable.
- Publication-Ready pie charts.



7. ggpiestats - Goodness of Fit

```
> Titanic_full
# A tibble: 2,201 x 5
     id Class Sex   Age   Survived
  <dbl> <fct> <fct> <fct> <fct>
1     1 3rd   Male  Child No
2     2 3rd   Male  Child No
3     3 3rd   Male  Child No
4     4 3rd   Male  Child No
5     5 3rd   Male  Child No
6     6 3rd   Male  Child No
7     7 3rd   Male  Child No
8     8 3rd   Male  Child No
9     9 3rd   Male  Child No
```



Defaults return

- ☑ descriptives (frequency + %s)
- ☑ inferential statistics
- ☑ effect size + CIs
- ☑ Goodness-of-fit tests
- ☑ Bayesian hypothesis-testing
- ☑ Bayesian estimation

Changing the type of test

- ☑ "p" → **Parametric** → μ_{mean}
- ☑ "np" → **non - parametric** → μ_{median}
- ☑ "r" → **robust** → $\mu_{trimmed}$
- ☑ "bf" → **Bayesian** → μ_{MAP}

Test by design

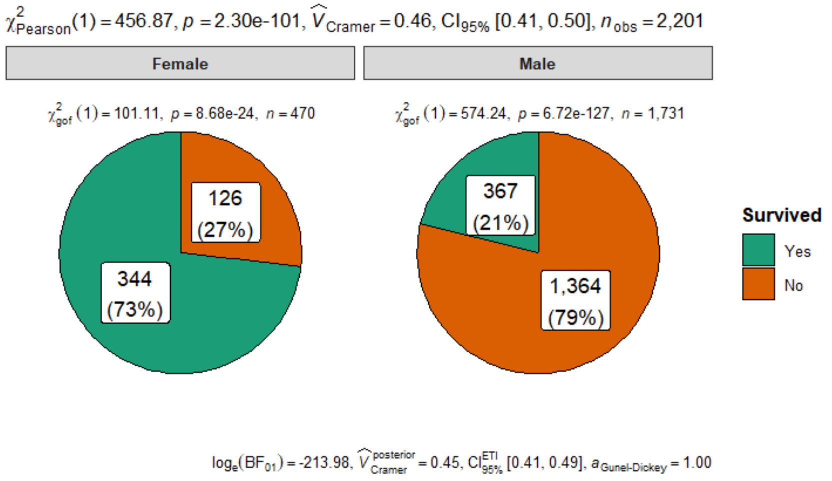
- Paired = FALSE → Pearson's χ^2
- Paired = TRUE → McNemar's χ^2

```
ggpiestats(
  data = Titanic_full, #data frame
  x = Survived, # different categories in the Survived
  title = "Passenger survival on the Titanic",
  caption = "Source: Titanic survival dataset",
  legend.title = "Survived?",
  label = "both"  # other values "percentage"
  #(default), "counts"
)
```

7. ggpiestats- association between categorical variables

```
> Titanic_full
# A tibble: 2,201 × 5
     id Class Sex   Age   Survived
  <dbl> <fct> <fct> <fct> <fct>
1     1 3rd   Male  Child No
2     2 3rd   Male  Child No
3     3 3rd   Male  Child No
4     4 3rd   Male  Child No
5     5 3rd   Male  Child No
6     6 3rd   Male  Child No
7     7 3rd   Male  Child No
8     8 3rd   Male  Child No
9     9 3rd   Male  Child No
```

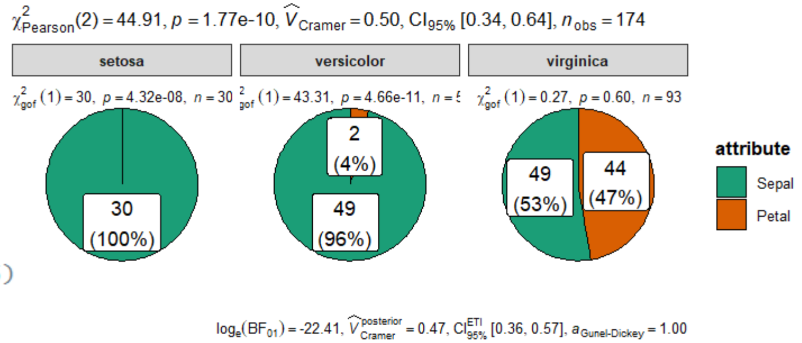
```
ggpiestats(
  data = Titanic_full, #data frame
  x = Survived, # different categories in the Survived
                # as rows in contigency table
  y = Sex, #sex of passanges as columns in contigency table
  label = "both"  # other values "percentage"
  #(default), "counts"
)
```



7. ggpiestats - Example

Question: Display a pie chart with the both percentages and counts to show attributes (Sepal or Petal) with a length of 5 or more among the different species in the Iris dataset?

```
> iris_long
# A tibble: 600 × 6
     id Species condition    attribute measure value
  <int> <fct>   <fct>        <fct>     <fct>   <dbl>
1     1 setosa  Sepal.Length Sepal     Length    5.1
2     2 setosa  Sepal.Length Sepal     Length    4.9
3     3 setosa  Sepal.Length Sepal     Length    4.7
4     4 setosa  Sepal.Length Sepal     Length    4.6
5     5 setosa  Sepal.Length Sepal     Length    5  
6     6 setosa  Sepal.Length Sepal     Length    5.4
7     7 setosa  Sepal.Length Sepal     Length    4.6
```




```
iris_filter <- dplyr::filter(iris_long, measure == "Length" & value >= 5)

# Use ggpiestats to create the pie chart
ggpiestats(
  data = iris_filter, # data frame
  x = attribute , # variable for rows in the contingency table
  y = Species , # variable for columns in the contingency table
  label = "both" # display both counts and percentages
)
```

3. ggpiestats - Arguments

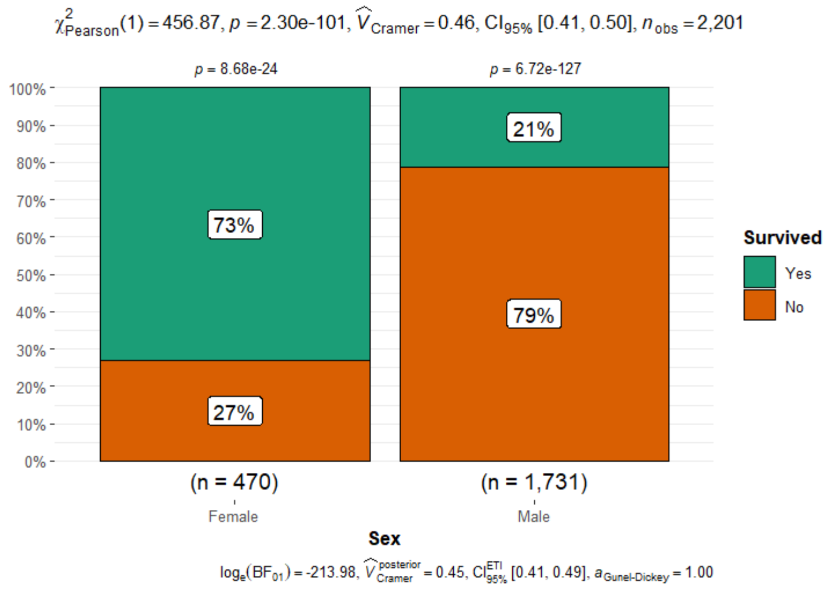
```
ggpiestats(
  data,
  x,
  y = NULL,
  counts = NULL,
  type = "parametric",
  paired = FALSE,
  results.subtitle = TRUE,
  label = "percentage",
  label.args = list(direction = "both"),
  label.repel = FALSE,
  digits = 2L,
  proportion.test = results.subtitle,
  digits.perc = 0L,
  bf.message = TRUE,
  ratio = NULL,
```

```
conf.level = 0.95,
sampling.plan = "indepMulti",
fixed.margin = "rows",
prior.concentration = 1,
title = NULL,
subtitle = NULL,
caption = NULL,
legend.title = NULL,
ggtheme = ggstatsplot::theme_ggstatsplot(),
package = "RColorBrewer",
palette = "Dark2",
ggplot.component = NULL,
```

https://indrajeetpatil.github.io/ggstatsplot/reference/ggpiestats.html

8. ggbarstats- association between categorical variables

- It is used for data exploration.
- Provide bar charts to summarize the statistical relationship(s) among one or more categorical variables.
- Can be used to check goodness of fit.
- To see if the frequency distribution of two categorical variables are independent of each other using the contingency table analysis
- To check if the proportion of observations at each level of a categorical variable is equal
- Highly Customizable.
- Publication-Ready bar charts.



8. ggbarstats- association between categorical variables

PARIS LODRON UNIVERSITÄT SALZBURG

```
> Titanic_full
# A tibble: 2,201 × 5
     id Class Sex   Age   Survived
  <dbl> <fct> <fct> <fct> <fct>
1     1 3rd   Male  Child No
2     2 3rd   Male  Child No
3     3 3rd   Male  Child No
4     4 3rd   Male  Child No
5     5 3rd   Male  Child No
6     6 3rd   Male  Child No
7     7 3rd   Male  Child No
```

Defaults return

- ✅ descriptives (frequency + %s)
- ✅ inferential statistics
- ✅ effect size + CIs
- ✅ Goodness-of-fit tests
- ✅ Bayesian hypothesis-testing
- ✅ Bayesian estimation

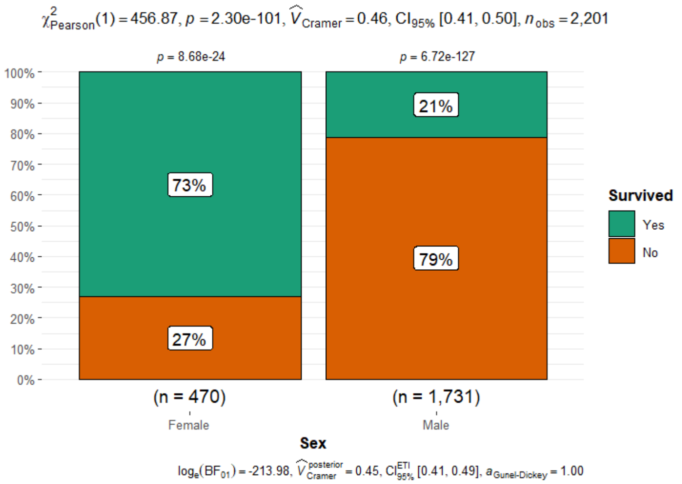
Test by design

- Paired = FALSE → Pearson's χ^2
- Paired = TRUE → McNemar's χ^2

Changing the type of test

- ✅ "p" → **Parametric** → μ_{mean}
- ✅ "np" → **non - parametric** → μ_{median}
- ✅ "r" → **robust** → $\mu_{trimmed}$
- ✅ "bf" → **Bayesian** → μ_{MAP}

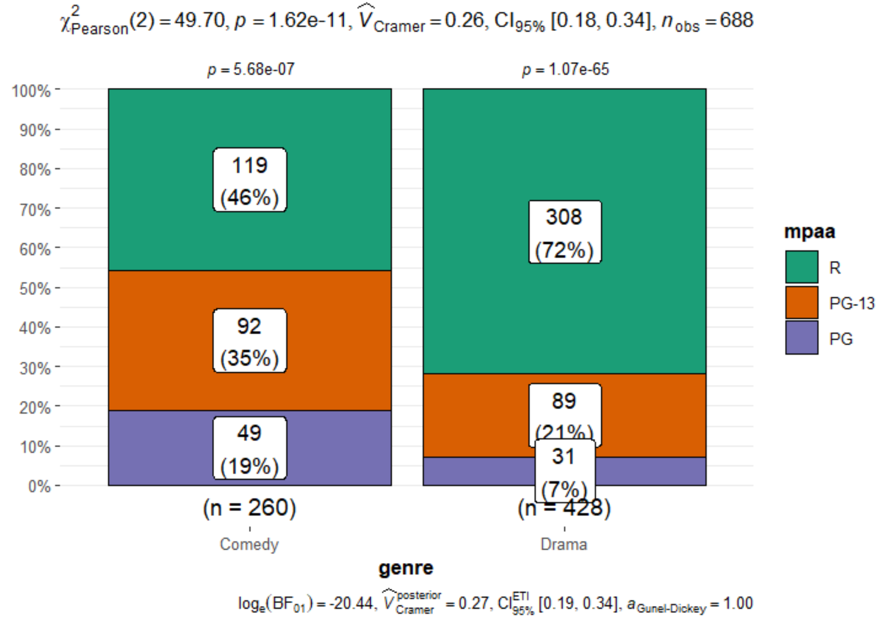
```
ggbarstats(
  data = Titanic_full,
  x = Survived,
  y = Sex
)
```



8. ggbarstats - Example

Question: Display a bar chart with the both percentages and counts to show attributes of MPAA ratings among Drama and Comedy genres in the movie dataset?

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



3. ggbarstats - Arguments

```
ggbarstats(
  data,
  x,
  y,
  counts = NULL,
  type = "parametric",
  paired = FALSE,
  results.subtitle = TRUE,
  label = "percentage",
  label.args = list(alpha = 1, fill = "white"),
  sample.size.label.args = list(size = 4),
  digits = 2L,
  proportion.test = results.subtitle,
  digits.perc = 0L,
  bf.message = TRUE,
  ratio = NULL,
  conf.level = 0.95,
```

```
  sampling.plan = "indepMulti",
  fixed.margin = "rows",
  prior.concentration = 1,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  legend.title = NULL,
  xlab = NULL,
  ylab = NULL,
  ggtheme = ggstatsplot::theme_ggstatsplot(),
  package = "RColorBrewer",
  palette = "Dark2",
  ggplot.component = NULL,
```

https://indrajeetpatil.github.io/ggstatsplot/reference/ggbarstats.html

Exercise 3

Load the starwars dataset.

3.1 Display a pie chart with the percentage distribution of genders among the characters in the Star Wars dataset?

3.2 Plot the pie-chart for character's gender percentage associated with, eye colour (blue, red, yellow and brown)

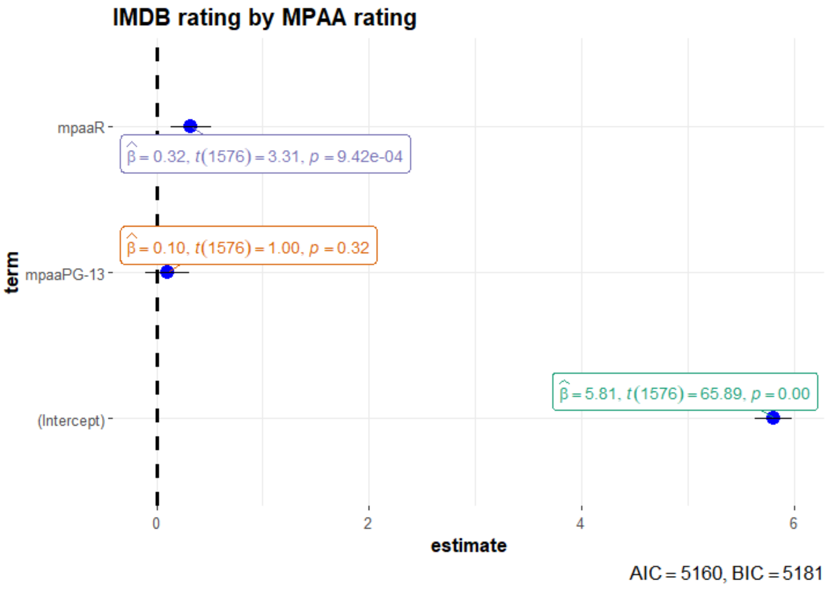
3.3 Display a bar chart to identify male species(Human, Wookiee, Zabrak) who has height greater than 70 was independent of or associated with hair_colour in (blond, brown and black)

Hypothesis about regression coefficients

📕ggcoefstats: Any regression model object

9. ggcoefstats

- It is used to generate **dot-and-whisker** plots for regression models saved in a tidy data frame.
- The tidy data frames are prepared using **parameters::model_parameters**.
- if available, the model summary indices are also extracted from **performance::model_performance**
- A dot representing the **estimate** and their **confidence intervals** (95% is the default).
- Estimate can be either effect size or regression coefficient.



9. ggcoefstats

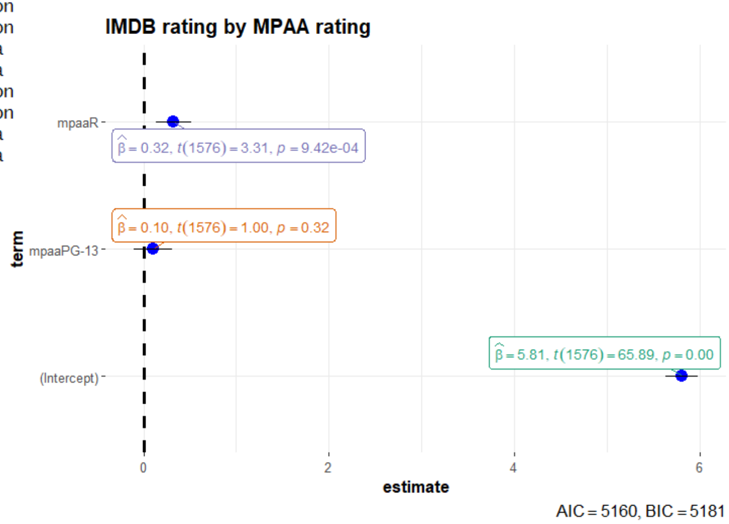
	title	year	length	budget	rating	votes	mpaa	genre
	<chr>	<int>	<int>	<dbl>	<dbl>	<int>	<fct>	<fct>
1	Shawshank Redemption, The	1994	142	25	9.1	149494	R	Drama
2	Lord of the Rings: The Return of the King, The	2003	251	94	9	103631	PG-13	Action
3	Lord of the Rings: The Fellowship of the Ring, The	2001	208	93	8.8	157608	PG-13	Action
4	Lord of the Rings: The Two Towers, The	2002	223	94	8.8	114797	PG-13	Action
5	Pulp Fiction	1994	168	8	8.8	132745	R	Drama
6	Schindler's List	1993	195	25	8.8	97667	R	Drama
7	Star Wars	1977	125	11	8.8	134640	PG	Action
8	Star Wars: Episode V - The Empire Strikes Back	1980	129	18	8.8	103706	PG	Action
9	C'era una volta il West	1968	158	5	8.7	17241	PG-13	Drama
0	Cidade de Deus	2002	135	3.3	8.7	25964	R	Drama

Defaults return

- ☑ inferential statistics
- ☑ estimate + CIs
- ☑ model summary (AIC and BIC)

```r
#model
mod <- stats::lm(
          formula = rating ~ mpaa,
          data = movies_long
)

ggcoefstats(
  x = mod, #model
  title = "IMDB rating by MPAA rating"
)
```



9. ggcoefstats- Supported models

```
  [1] "aareg"                 "afex_aov"
  [3] "AKP"                   "Anova.mlm"
  [5] "anova.rms"             "aov"
  [7] "aovlist"               "Arima"
  [9] "averaging"             "bamlss"
 [11] "bamlss.frame"          "bayesQR"
 [13] "bayesx"                "BBmm"
 [15] "BBreg"                 "bcplm"
 [17] "betamfx"               "betaor"
 [19] "betareg"               "BFBayesFacto
 [21] "bfsl"                  "BGGM"
 [23] "bife"                  "bifeAPEs"
 [25] "bigglm"                "biglm"
 [27] "blavaan"               "blrm"
 [29] "bracl"                 "brglm"
 [31] "brmsfit"               "brmultinom"
 [33] "btergm"                "censReg"
 [35] "cgam"                  "cgamm"
 [37] "cglm"                  "clm"
 [39] "clm2"                  "clmm"
 [41] "clmm2"                 "clogit"
 [43] "coeftest"              "complmrob"
 [45] "confusionMatrix"       "coxme"
 [47] "coxph"                 "coxph.penal"
 [49] "coxr"                  "cpglm"
 [51] "cpglmm"                "crch"
 [53] "crq"                   "crqs"
 [55] "crr"                   "dep.effect"
 [57] "DirichletRegModel"     "draws"
 [59] "drc"                   "eglm"
 [61] "elm"                   "epi.2by2"
 [63] "ergm"                  "feglm"
 [65] "feis"                  "felm"
 [67] "fitdistr"              "fixest"
 [69] "flac"                  "flexsurvreg"
 [71] "flic"                  "gam"
 [73] "Gam"                   "gamlss"
 [75] "gamm"                  "gamm4"
```

```
 [77] "garch"                 "gbm"
 [79] "gee"                   "geeglm"
 [81] "glht"                  "glimML"
 [83] "glm"                   "Glm"
 [85] "glmm"                  "glmmadmb"
 [87] "glmmPQL"               "glmmTMB"
 [89] "glmrob"                "glmRob"
 [91] "glmx"                  "gls"
 [93] "gmnl"                  "hglm"
 [95] "HLfit"                 "htest"
 [97] "hurdle"                "iv_robust"
 [99] "ivFixed"               "ivprobit"
[101] "ivreg"                 "lavaan"
[103] "lm"                    "lm_robust"
[105] "lme"                   "lmerMod"
[107] "lmerModLmerTest"       "lmodel2"
[109] "lmrob"                 "lmRob"
[111] "logistf"               "logitmfx"
[113] "logitor"               "logitr"
[115] "LORgee"                "lqm"
[117] "lqmm"                  "lrm"
[119] "manova"                "MANOVA"
[121] "marginaleffects"       "marginaleffects.summary"
[123] "margins"               "maxLik"
[125] "mblogit"               "mclogit"
[127] "mcmc"                  "mcmc.list"
[129] "MCMCglmm"              "mcp1"
[131] "mcp12"                 "mcp2"
[133] "med1way"               "mediate"
[135] "merMod"                "merModList"
[137] "meta_bma"              "meta_fixed"
[139] "meta_random"           "metaplus"
[141] "mhurdle"               "mipo"
[143] "mira"                  "mixed"
[145] "MixMod"                "mixor"
[147] "mjoint"                "mle"
[149] "mle2"                  "mlm"
[151] "mlogit"                "mmclogit"
[153] "mmlogit"               "mmrm"
[155] "mmrm_fit"              "mmrm_tmb"
```

```
[157] "model_fit"             "multinom"
[159] "mvord"                 "negbinirr"
[161] "negbinmfx"             "nestedLogit"
[163] "ols"                   "onesampb"
[165] "orm"                   "pgmm"
[167] "phyloglm"              "phylolm"
[169] "plm"                   "PMCMR"
[171] "poissonirr"            "poissonmfx"
[173] "polr"                  "probitmfx"
[175] "psm"                   "Rchoice"
[177] "ridgelm"               "riskRegression"
[179] "rjags"                 "rlm"
[181] "rlmerMod"              "RM"
[183] "rma"                   "rma.uni"
[185] "robmixglm"             "robtab"
[187] "rq"                    "rqs"
[189] "rqss"                  "rvar"
[191] "Sarlm"                 "scam"
[193] "selection"             "sem"
[195] "SemiParBIV"            "semLm"
[197] "semLme"                "serp"
[199] "slm"                   "speedglm"
[201] "speedlm"               "stanfit"
[203] "stanmvreg"             "stanreg"
[205] "summary.lm"            "survfit"
[207] "survreg"               "svy_vglm"
[209] "svychisq"              "svyglm"
[211] "svyolr"                "t1way"
[213] "tobit"                 "trimcibt"
[215] "truncreg"              "vgam"
[217] "vglm"                  "wbgee"
[219] "wblm"                  "wbm"
[221] "wmcpAKP"               "yuen"
[223] "yuend"                 "zcpglm"
[225] "zeroinfl"              "zerotrunc"
```

3. ggcoefstats - Arguments

PARIS LODRON UNIVERSITÄT SALZBURG

```
ggcoefstats(
  x,
  statistic = NULL,
  conf.int = TRUE,
  conf.level = 0.95,
  digits = 2L,
  exclude.intercept = FALSE,
  effectsize.type = "eta",
  meta.analytic.effect = FALSE,
  meta.type = "parametric",
  bf.message = TRUE,
  sort = "none",
  xlab = NULL,
  ylab = NULL,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  only.significant = FALSE,
```

```
point.args = list(size = 3, color = "blue", na.rm = TRUE),
errorbar.args = list(height = 0, na.rm = TRUE),
vline = TRUE,
vline.args = list(linewidth = 1, linetype = "dashed"),
stats.labels = TRUE,
stats.label.color = NULL,
stats.label.args = list(size = 3, direction = "y", min.segment.length = 0, na.r
  TRUE),
package = "RColorBrewer",
palette = "Dark2",
ggtheme = ggstatsplot::theme_ggstatsplot(),
```

https://indrajeetpatil.github.io/ggstatsplot/reference/ggcoefstats.html

grouped_variants of all functions

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

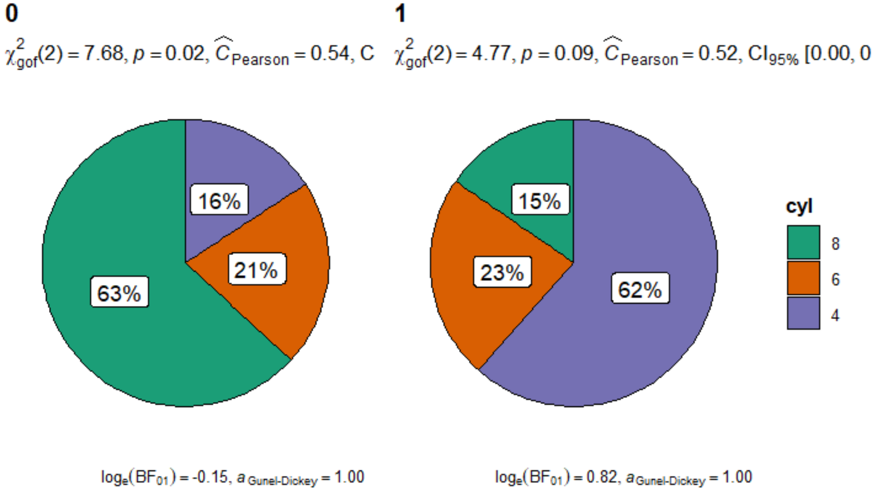
grouped_functions

```
> mtcars
                     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Mazda RX4           21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
```

Available grouped_ variants

- grouped_ggbetweenstats
- grouped_withinstats
- grouped_gghistostats
- grouped_ggdotplotstats
- grouped_ggscatterstats
- grouped_ggcorrmatt
- grouped_ggpiestats
- grouped_ggbarstats

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



Exercise 4

Load the starwars dataset.

4.1 Display a pie chart with the percentage distribution of genders among the characters in the Star Wars dataset, group by their eye colour in ("blue','red','yellow','brown").

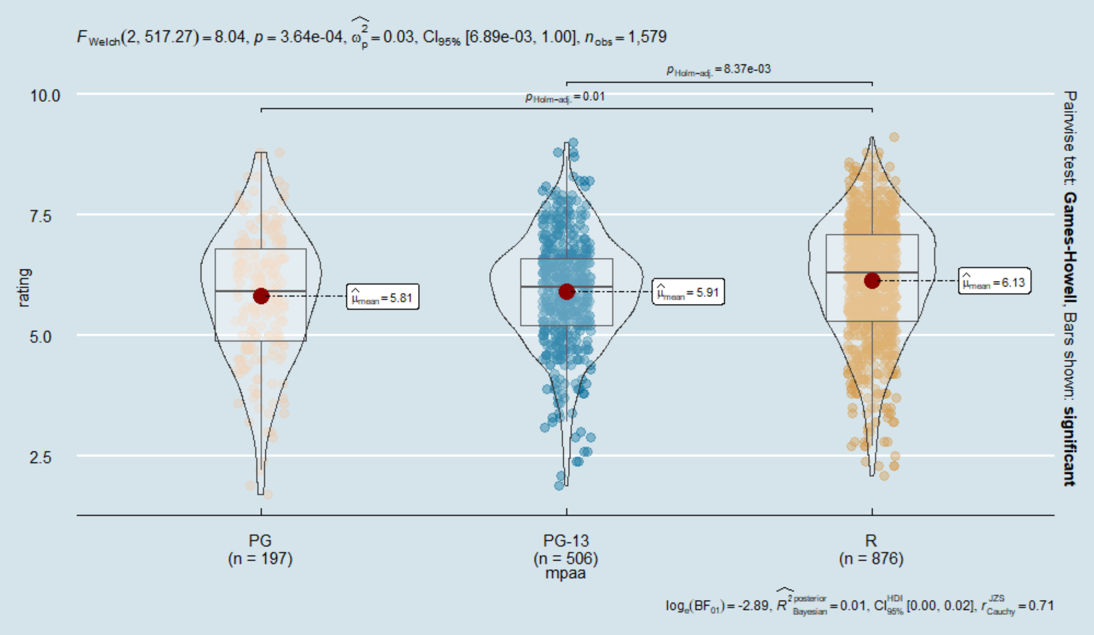
4.2 Plot the distribution of height across gender (feminine, masculine) of Star Wars characters, which grouping by their skin colour (fair, gold)

Customizability of ggstatsplot

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

Changing Themes and Color Palettes

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



The default palette is colorblind-friendly.

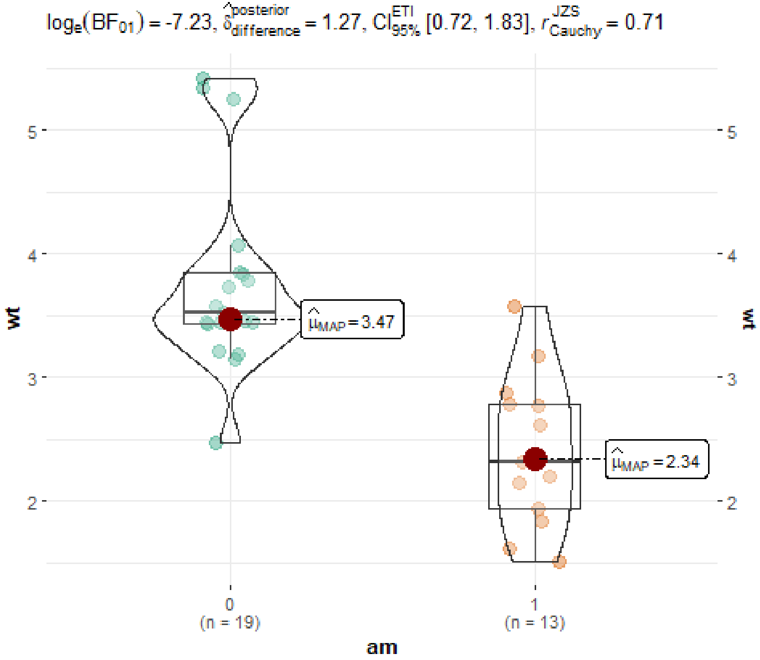
Install the ggthemes package

```
install.packages('ggthemes')
```

Further Modifications Using ggplot2

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

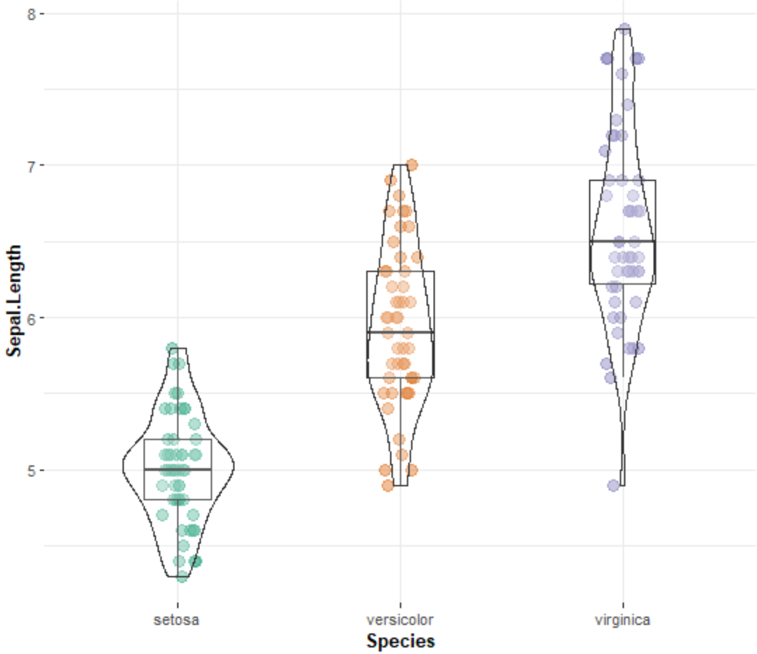
```r
ggbetweenstats(
  data = mtcars,
  x = am,
  y = wt,
  type = "bayes"
) +
  scale_y_continuous(sec.axis = dup_axis())
```



Show Only Plots

`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"
)
```

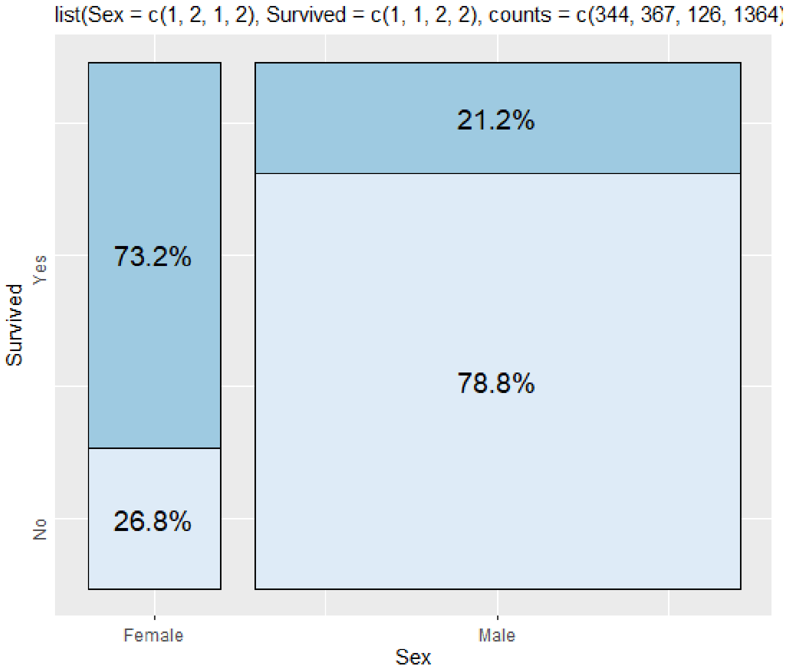


Get Only Expression

`ggstatsplot` can be used to get only expressions.

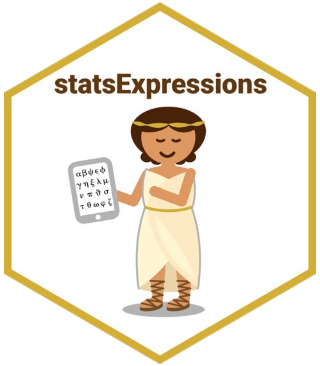
```
results <- ggpiestats(
  data = Titanic_full,
  x = Survived,
  y = Sex,
  output = "subtitle"
)

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



Get Output as a Table

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




```
library(statsExpressions)

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

mu	statistic	df.error	p.value	method	alternative	effectsize	estimate	conf.level	conf.low	conf.high
3	1.256009	31	0.2184965	One Sample t-test	two.sided	Hedges' g	0.2166103	0.95	-0.1273401	0.5571645

Exercise 5

Using the starwars dataset,

5.1: Construct a histogram for the ‘height’ variable. Ensure that there are no statistical annotations included in the plot.

5.2: Generate a scatterplot for the variables ‘Height’ and ‘Mass’. The plot should not contain any subtitle annotations.

- Identify if there are any outliers present in the scatterplot. If outliers are found, remove them and replot the data.
- Apply the Wall Street Journal Theme (theme_wsj) to the scatterplot.

Misconceptions

PARIS LODRON UNIVERSITÄT SALZBURG

❌ 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

PARIS LODRON UNIVERSITÄT SALZBURG

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.

Sources

Git Repository: https://indrajeetpatil.github.io/ggstatsplot/

Source Presentation: https://indrajeetpatil.github.io/ggstatsplot_slides/slides/ggstatsplot_presentation.html

Questions

Thank You