

What is the difference between `facet_wrap()` and `facet_grid()` in R?

Authored by
stats writer

June 29, 2024

RECOMMENDED CITATION

stats writer (2024). *What is the difference between `facet_wrap()` and `facet_grid()` in R?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=158600>

`Facet_wrap()` and `facet_grid()` are both functions in the R programming language that allow for the creation of multiple plots within a single plot window. The main difference between the two is the way in which the plots are organized.

`Facet_wrap()` arranges the plots in a single row or column, depending on the specified direction, and wraps them onto subsequent rows or columns as needed. This results in a compact layout, with each plot sharing the same axes and scales.

On the other hand, `facet_grid()` creates a grid of plots, with each plot representing a unique combination of variables specified by the user. This allows for more flexibility in the layout and allows for the comparison of different variables across multiple plots.

In summary, the main difference between `facet_wrap()` and `facet_grid()` is the layout of the plots, with `facet_wrap()` creating a compact and `facet_grid()` creating a grid-like layout. The choice between the two functions depends on the desired visual representation and the variables being compared.

The Difference Between `facet_wrap()` and `facet_grid()` in R

The `facet_grid()` and `facet_wrap()` functions from the `ggplot2` package can both be used to produce a grid of plots.

Here's the main difference between the two functions:

The `facet_grid()` function will produce a grid of plots for each combination of variables that you specify, even if some plots are empty. The `facet_wrap()` function will only produce plots for the combinations of variables that have values, which means it won't produce any empty plots.

The following two examples illustrate the difference between these two functions, using the following data frame:

```
#create data frame
```

```
df <- data.frame(team=c('A', 'A', 'A', 'A', 'B', 'B', 'B', 'B'),  
position=c('G', 'G', 'F', 'F', 'G', 'G', 'G', 'G'),  
points=c(8, 14, 20, 22, 25, 29, 30, 31),  
assists=c(10, 5, 5, 3, 8, 6, 9, 12))
```

```
#view data frame
```

```
df
```

```
team position points assists
```

```
1 A G 8 10
```

```
2 A G 14 5
```

```
3 A F 20 5
```

```
4 A F 22 3
```

```
5 B G 25 8
```

```
6 B G 29 6
```

```
7 B G 30 9
```

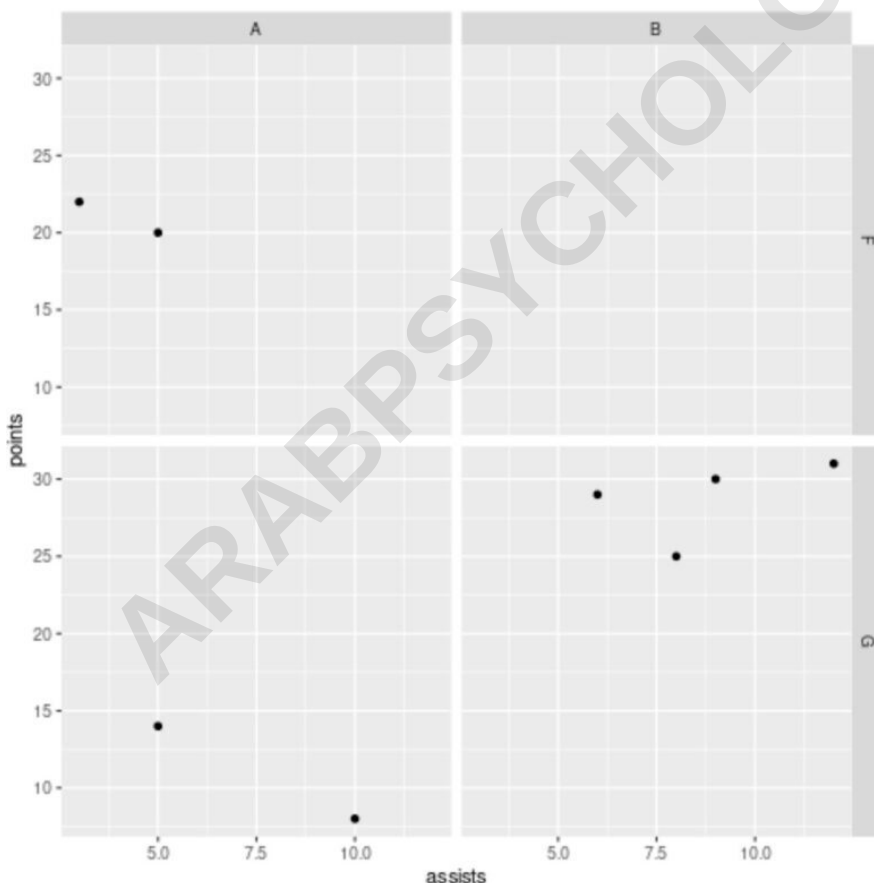
```
8 B G 31 12
```

Example 1: Use `facet_grid()` Function

The following code shows how to use `facet_grid()` to create a grid that displays a scatterplot of assists vs. points for each combination of team and position:

```
library(ggplot2)

ggplot(df, aes(assists, points)) +
  geom_point() +
  facet_grid(position~team)
```



Notice that a scatterplot is produced for every

combination of team and position, even though no values exist in the original data frame for a team value of B and a position value of F:



This is how `facet_grid()` works: It will produce a plot for every combination of the variables that you specify, even if some plots are empty.

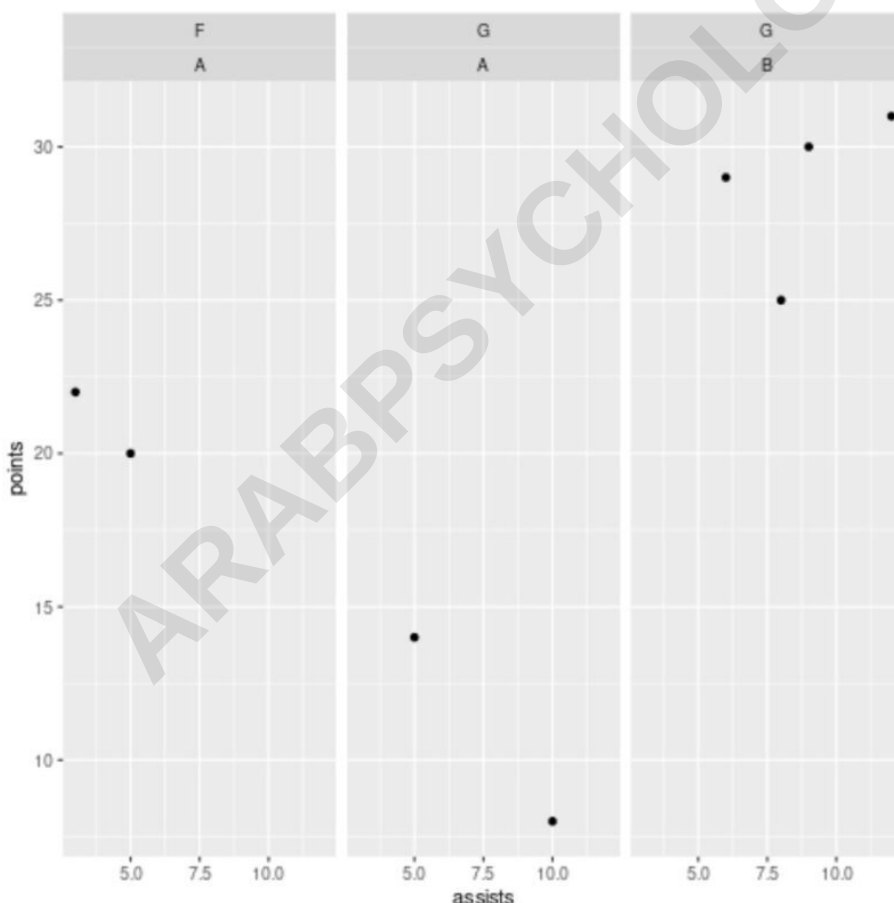
Example 2: Use `facet_wrap()` Function

The following code shows how to use `facet_wrap()` to

create a grid that displays a scatterplot of assists vs. points for each combination of team and position that exists:

```
library(ggplot2)
```

```
ggplot(df, aes(assists, points)) +  
geom_point() +  
facet_wrap(position~team)
```



Notice that a scatterplot is only produced for the

combinations of team and position that exist in the original data frame.

This is how `facet_wrap()` works: It will never produce an empty plot.

Note: Refer to the `ggplot2` documentation for a complete guide to the `facet` functions.

Additional Resources

The following tutorials explain how to perform other common tasks in `ggplot2`: