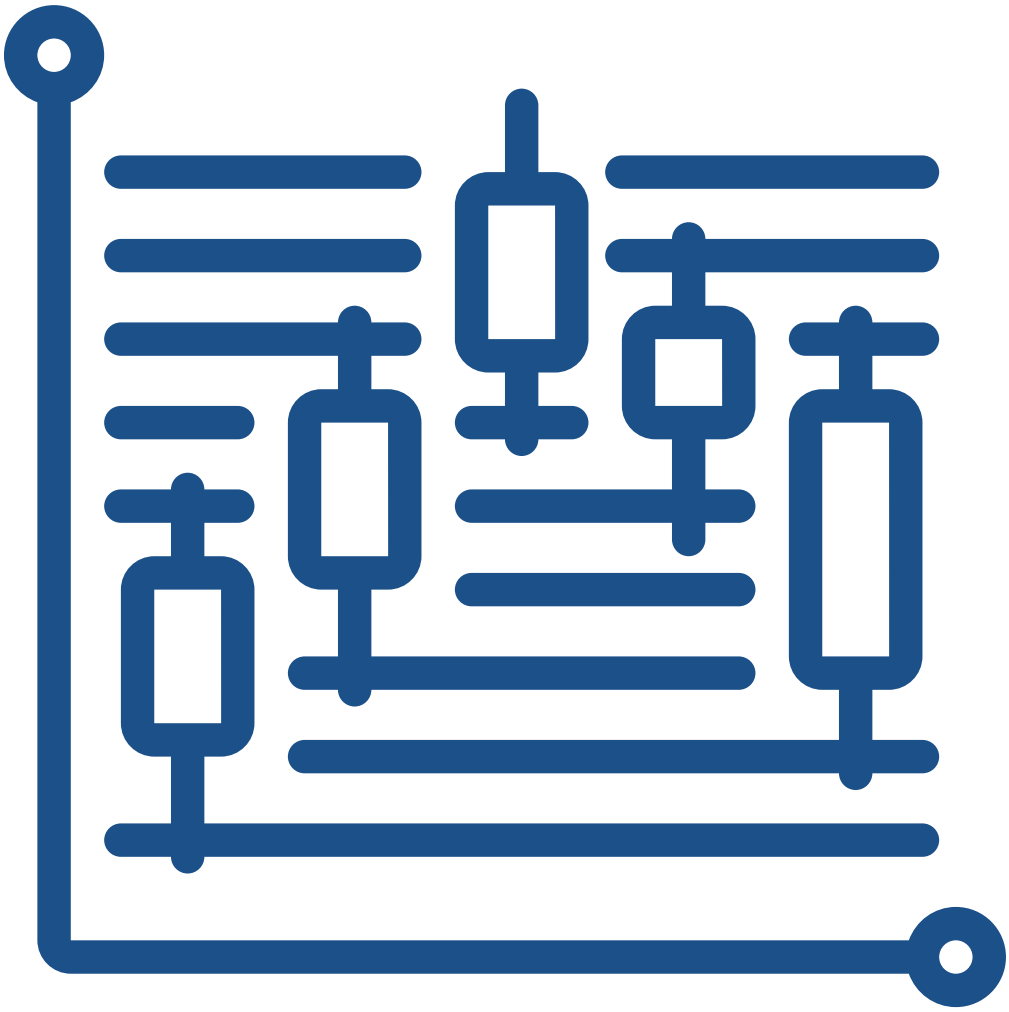
LEARN MORE 365



GGPLOT -Facets Lesson

Facets

Greg Martin

Using facets

What are facets

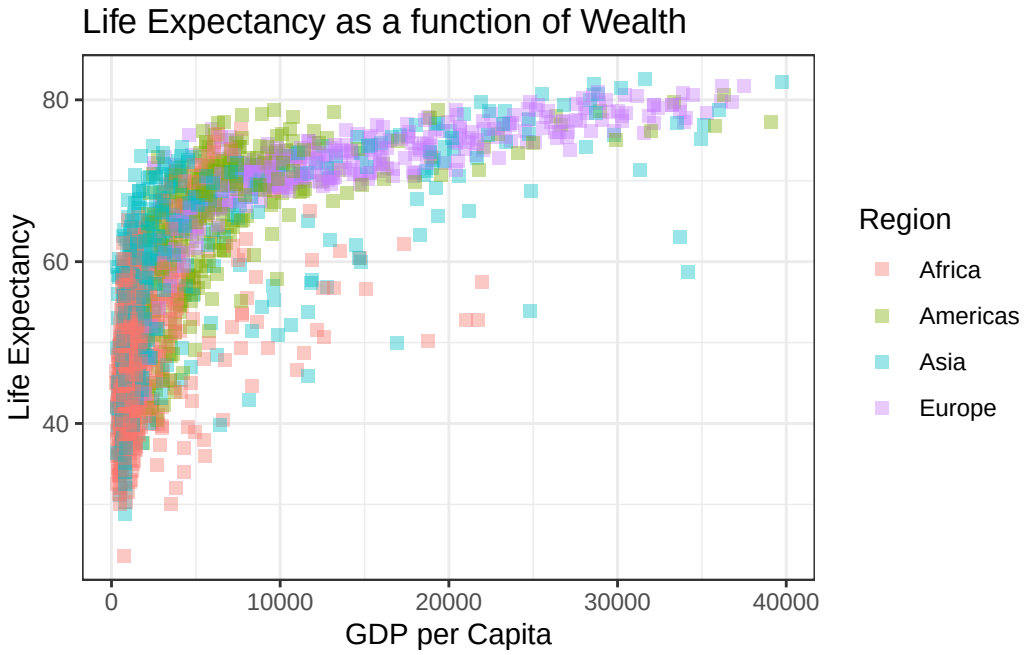
Facets are used to create multiple plots, each showing a subset of your data. These plots are arranged in a grid, with each plot showing a different subset of the data, based on the values of one or more variables. As the saying goes, a picture is worth a thousand words; so let's look at a a few examples of facets in action.

Let's take a look at the `gapminder` dataset (that you can access once you've installed the `gapminder` package. The variables that are of interest below are `gdpPercap` (a crude indicator of a countries wealth) and`lifeExp` (which is, as the name suggests, life expectancy).

We're interested in the relationship between these two variables and how that may vary in different regions of the world.

We can see from the plot below that people living in richer countries do seem to live longer. It is also clear that that the relationship differs in the different regions but its difficult to see the details with all of the plot points overlapping each other.

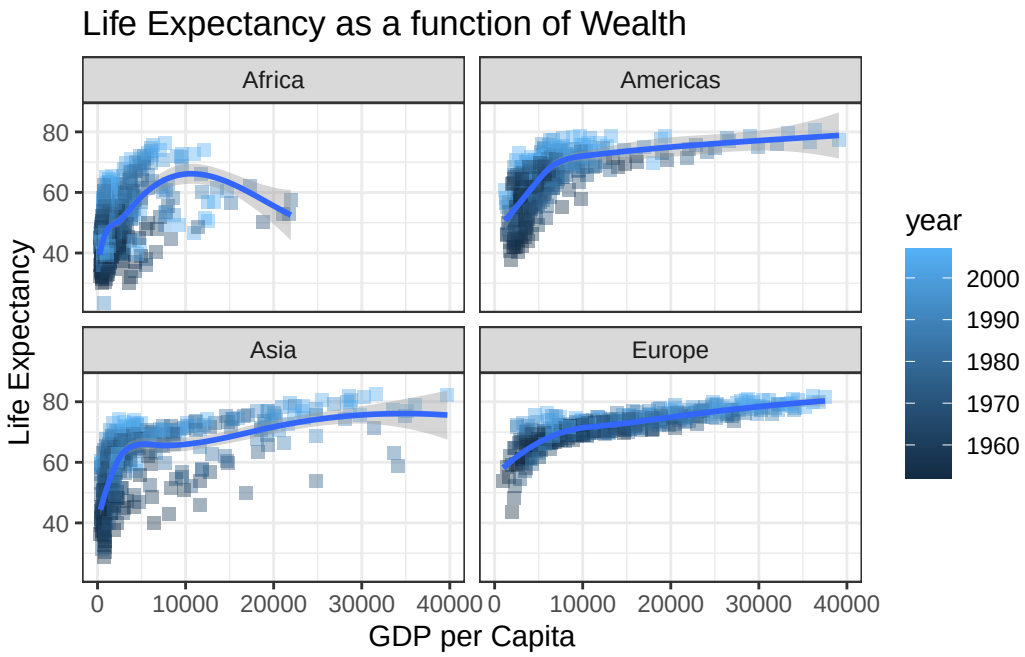
```r
# install.packages("gapminder")
library(gapminder)
view(gapminder)

gapminder %>%
  filter(gdpPercap < 40000 &
         continent != "Oceania") %>%
  ggplot(mapping = aes(x = gdpPercap,
                       y = lifeExp,
                       colour = continent))+
  geom_point(shape = "square",
             alpha = 0.4,
             size = 2)+
  labs(title = "Life Expectancy as a function of Wealth",
       x = "GDP per Capita",
       y = "Life Expectancy",
       colour = "Region")+
  theme_bw()
```



facet_wrap

A simple solution is to plot each of the continents of interest on a separate but adjacent graph so that we can more clearly see the difference between them. To do this we use the `face_wrap()` function and specify which categorical variable to facet by using the ~ symbol.

```
gapminder %>%
  filter(gdpPercap < 40000 &
         continent != "Oceania") %>%
  ggplot(mapping = aes(x = gdpPercap,
                       y = lifeExp,
                       colour = year))+
  geom_jitter(shape = "square",
              alpha = 0.4,
              size = 2)+
  geom_smooth()+
  facet_wrap(~continent)+
  labs(title = "Life Expectancy as a function of Wealth",
       x = "GDP per Capita",
       y = "Life Expectancy")+
  theme_bw()
```

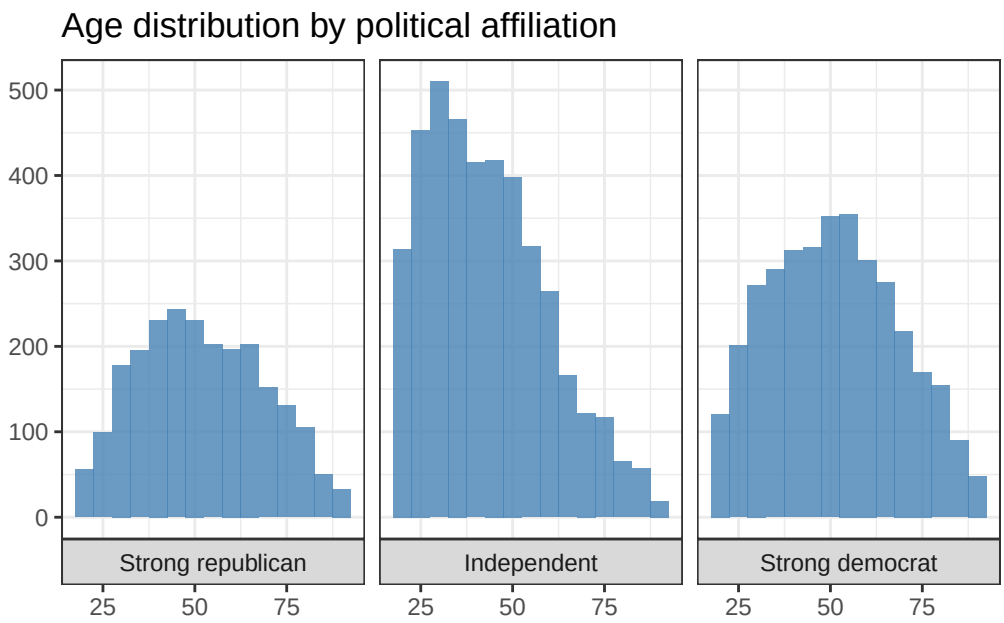


> **Note**
>
> You'll notice that because we no longer needed to use color to distinguish the continents, we can now map that aesthetic to another variables, in this case `year`.

The `facet_wrap` function has a number of arguments that can be used to tweak your plot. Run the command `?facet_wrap` to see a list of all of the arguments. The more commonly used arguments are `nrow` and `ncol` to specific the number of rows and number of columns respectively.

In the example below, we've used the `gss_cat` dataset that comes with the `forcats` package to illustrate the use of these arguments. From the `gss_cat` dataset we plot histograms of `age`, disaggregated by `partyid` (political affiliation). To ensure that all of the facets are next to each other on the same row, we stipulate that `nrow = 1`. We've also changed the position of the strip that labels the facet but stipulating `strip.position = "bottom"`:

```
library(forcats)
view(gss_cat)

gss_cat %>%
  filter(partyid %in% c("Strong democrat",
                        "Strong republican",
                        "Independent")) %>%
  ggplot(aes(age))+
  geom_histogram(binwidth = 5,
                 fill = "steelblue",
                 alpha = .8)+
    facet_wrap(~partyid,
               nrow = 1,
               ncol = 3,
               strip.position = "bottom")+
  labs(title = "Age distribution by political affiliation",
       x = "",
       y = "")+
  theme_bw()
```



---

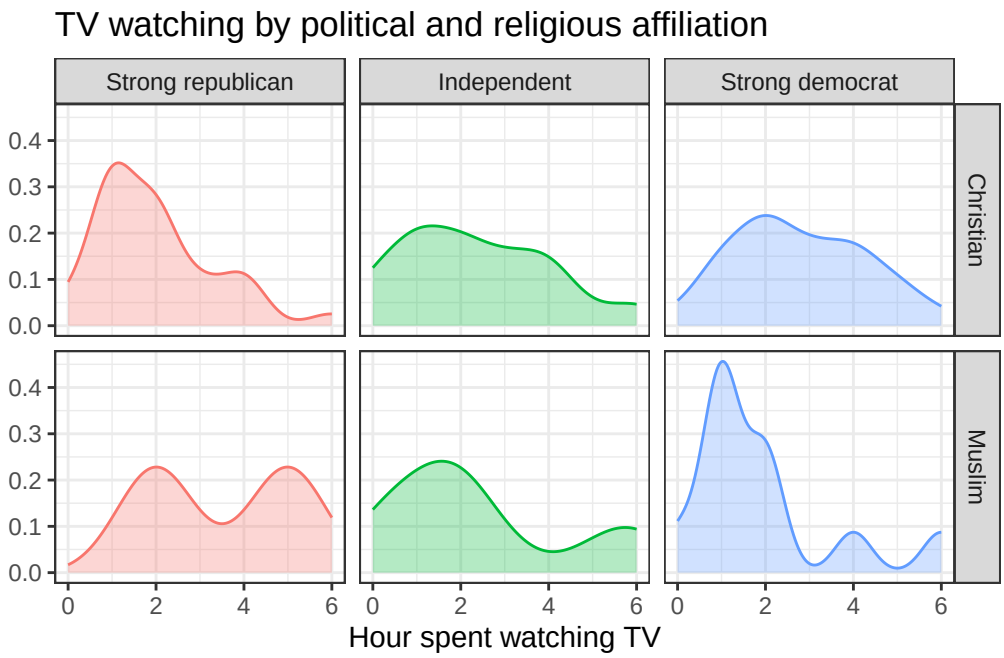
facet_grid

Can we facet in two directions? What does that even mean? The easiest way to describe a facet_grid is to create one and then take a look at what it is and how to interpret it. So here we go.

> **!** Data wrangling
>
> Oh, but they way. The `gss_cat` dataset that comes with the `forcats` package uses the very outdated spelling `Moslem` instead of the preferred `Muslim` in reference to people that adhere to the Islamic faith. So in the code below, the first thing we do is use the `mutate` and `recode` functions to fix that before creating the plot.

```
gss_cat %>%
  mutate(relig = recode(relig,
                        "Moslem/islam" = "Muslim")) %>%
    filter(relig %in% c("Christian",
                        "Muslim") &
           partyid %in% c("Strong democrat",
```

```
                      "Strong republican",
                      "Independent")&
       tvhours<10) %>%
ggplot(aes(x = tvhours,
           color = partyid,
           fill = partyid))+
geom_density(alpha = 0.3, show.legend = F)+
facet_grid(relig ~ partyid)+
labs(title = "TV watching by political and religious affiliation",
     x = "Hour spent watching TV",
     y = "")+
theme_bw()
```



Notice that in each facet, the data has been disaggregated by two factors (religion and political affiliation. By looking at the left column we can see difference between Christian and Muslim republicans, in terms of the distribution of their TV watching behavior. By looking at the first row, however, we can see the difference in the distribution of TV watching behavior across political affiliations.

Learn More 365

Share this cheat sheet with others

Unlock a wealth of FREE resources TODAY!

Click on the links below to enrich your knowledge in these topics:

- Statistics & Research Methods Resource Library
- R Programming & Data Visualization Resource Library
- Discover the Public Health & Epidemiology Resource Library

Explore video lessons on **Statistics & Research Methods, R Programming & Data Visualization Public Health & Epidemiology,** and more at Learn More 365!

Click **HERE** or scan the QR code below to start learning!