`ggplot2`: An Implementation of the Grammar of Graphics in R

Data science using R-Software

Marcus Nunes

December 6, 2021

Statistics Department - UFRN

What we will see today

What we will see today

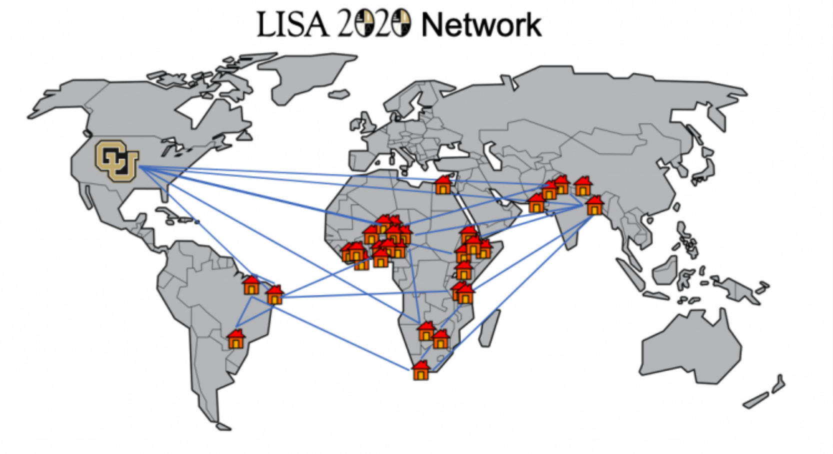
1. Who am I?
2. Motivation
3. Grammar of Graphics
4. `ggplot2`
5. Conclusions

Who am I?

Who am I?

- Marcus Nunes, Assistant Professor at Statistics Department, Federal University of Rio Grande do Norte
- PhD in Statistics - Pennsylvania State University (2013)
- Interested in Statistical Education, Machine Learning and Statistical Collaboration Projects
- Email: `marcus@marcusnunes.me`

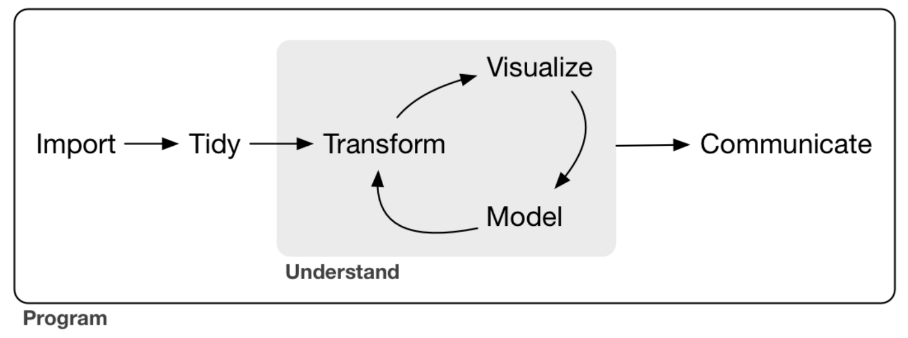
Who am I?



link

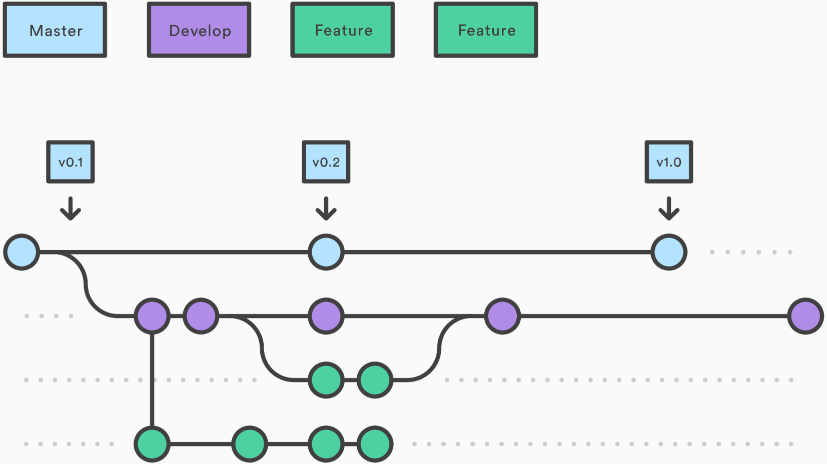
Motivation

Motivation



Source: Grolemund and Wickham (2017)

Motivation



Source: KDnuggets

Motivation

- Good statisticians and mathematicians who write code without optimization
- Good computer scientists who understand a little statistics and math
- Geoscientists with data expertise
- Managers who know how to make these people work together

Motivation

- A data scientist is someone who understands programming more than a traditional statistician
- Also, understands statistics more than a traditional Computer Scientist
- And, above all, it is someone who can find solutions to problems by joining these two areas of knowledge

Motivation

- Data science on a programming language makes reproducibility easier
- Why use R?
- According to IEEE, R is the 7th most popular programming language in the world
- R is built by statisticians to statisticians
- It is a natural choice for us

Motivation

- It is lightweight: almost any computer can run it
- Even if your computer cannot run R, you can use an internet browser
- It is the language of choice by many statisticians

Grammar of Graphics

Grammar of Graphics

- `ggplot2` is a data visualization package
- It was created by Hadley Wickham in 2005, based on the book *Grammar of Graphics*, by Leland Wilkinson, but its first version was only available to the public in 2007
- The main idea is to create graphics as if they were phrases in a language, where each graphic element is a word

Grammar of Graphics

- That is, let's work with the concept of **grammar of graphics** (hence the **gg** in `ggplot2`)
- This allows us to build graphics as complex as we want
- Graphics created with this tool are generally more beautiful than traditional `R` graphics or other similar tools.

Grammar of Graphics

- Each graph consists of seven layers: data, aesthetics, geometry, facets, statistics, coordinates and theme
- The first three are fundamental: every graphic will have them
    - data: consist of the base layer; from the data we will think which variables will be worked
    - aesthetics: consists of the variables selected for plotting, grouping, coloring, etc.
    - geometry: layer where we define the shapes of graphic elements, such as points, lines and intervals

Grammar of Graphics

- The following four are optional: they allow us to customize our views
  - facets: useful when we want to split chart information for better visualization, it can be used for group comparisons
  - statistics: it is the layer that represents the analysis of the data, if they are transformed
  - coordinates: informs where the graph will be built, whether in Cartesian or Polar coordinates, for example
  - theme: layer referring to the general view of the chart, changing background colors, axes format, font size and much more.

ggplot2

ggplot2

- Install R: https://cran.r-project.org/
- Install RStudio: https://www.rstudio.com/products/rstudio/download/
- Open RStudio and run the following code:

```
install.packages("tidyverse")
```

ggplot2

- Our first plot will be made from the `mpg` dataset

```
## # A tibble: 234 x 11
##    manufacturer model      displ  year   cyl trans drv     cty   hwy fl    class
##    <chr>        <chr>      <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>
##  1 audi         a4           1.8  1999     4 auto~ f        18    29 p     comp~
##  2 audi         a4           1.8  1999     4 manu~ f        21    29 p     comp~
##  3 audi         a4           2    2008     4 manu~ f        20    31 p     comp~
##  4 audi         a4           2    2008     4 auto~ f        21    30 p     comp~
##  5 audi         a4           2.8  1999     6 auto~ f        16    26 p     comp~
##  6 audi         a4           2.8  1999     6 manu~ f        18    26 p     comp~
##  7 audi         a4           3.1  2008     6 auto~ f        18    27 p     comp~
##  8 audi         a4 quattro   1.8  1999     4 manu~ 4        18    26 p     comp~
##  9 audi         a4 quattro   1.8  1999     4 auto~ 4        16    25 p     comp~
## 10 audi         a4 quattro   2    2008     4 manu~ 4        20    28 p     comp~
## # ... with 224 more rows
```

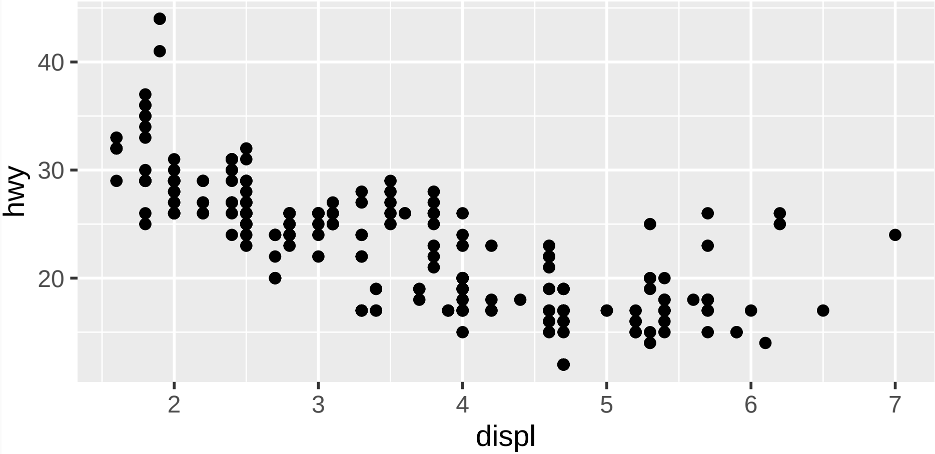
ggplot2

- Think about the variables `hwy` (consumption in miles per gallon on the road) and `displ` (size of car engine, in liters)
- What happens to the consumption of the car when the engine size in liters increases?
- Does this make sense according to your intuition?

ggplot2

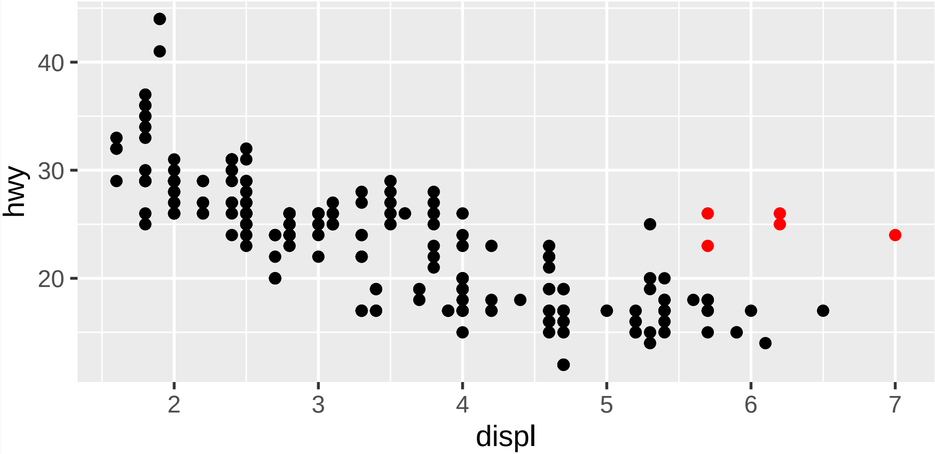
```r
# loading the ggplot2 package

library(ggplot2)

# hwy x displ scatter plot

ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point()
```

ggplot2



ggplot2

- See that there are some dots highlighted in red in the image below. They seem to run away from the linear behavior of the other points in the dataset.

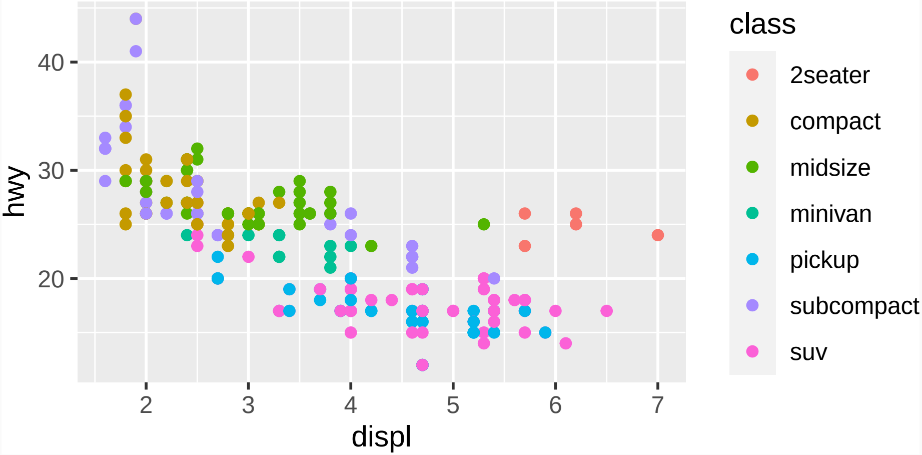


ggplot2

- Note that we are able to generate a more sophisticated graph if we change the code we’ve used so far

```
# hwy x displ scatter plot with caption

ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(aes(colour = class))
```

ggplot2

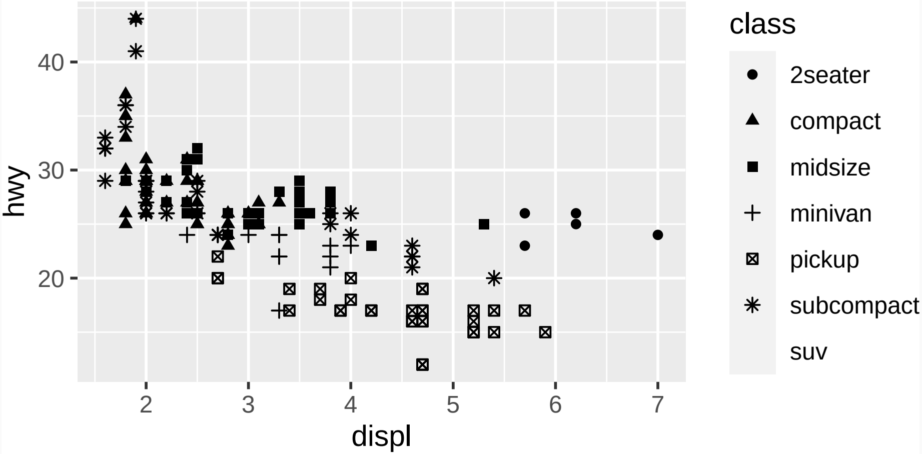

hwy
40
30
20
2
3
4
5
6
7
displ
class
2seater
compact
midsize
minivan
pickup
subcompact
suv

ggplot2

- We are not limited to using only colors to identify the different types of cars
- Shapes are a good option too, specially if you plan to print you plot in black and white

```r
# hwy x displ scatter plot with caption

ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(aes(shape = class))
```

ggplot2


hwy
40
30
20
2
3
4
5
6
7
displ
class
2seater
compact
midsize
minivan
pickup
subcompact
suv

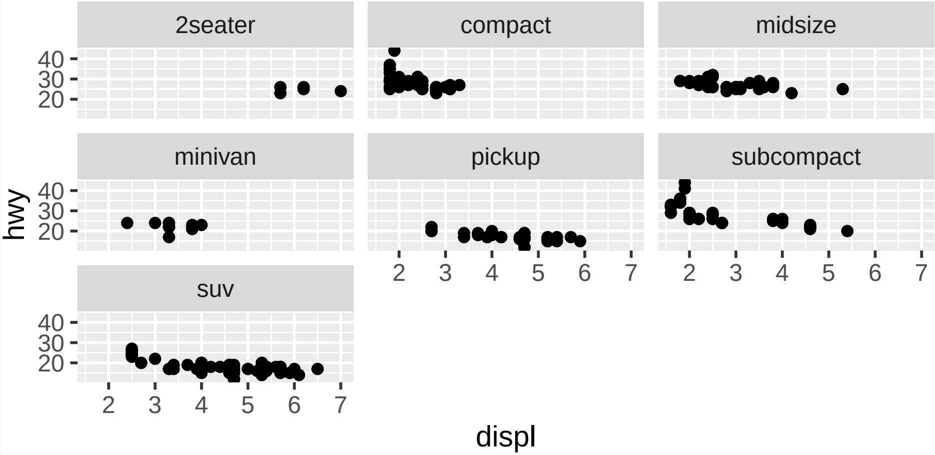
ggplot2

- From the list of the seven main components of each graph, we’ve already seen how to work with data, aesthetics and geometry
- We still have to see how to add facets, statistics, coordinates and theme to our product
- We’ll start with the facets
- See the next chart, divided into panels

ggplot2

```r
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point() +
  facet_wrap(~class)
```

ggplot2

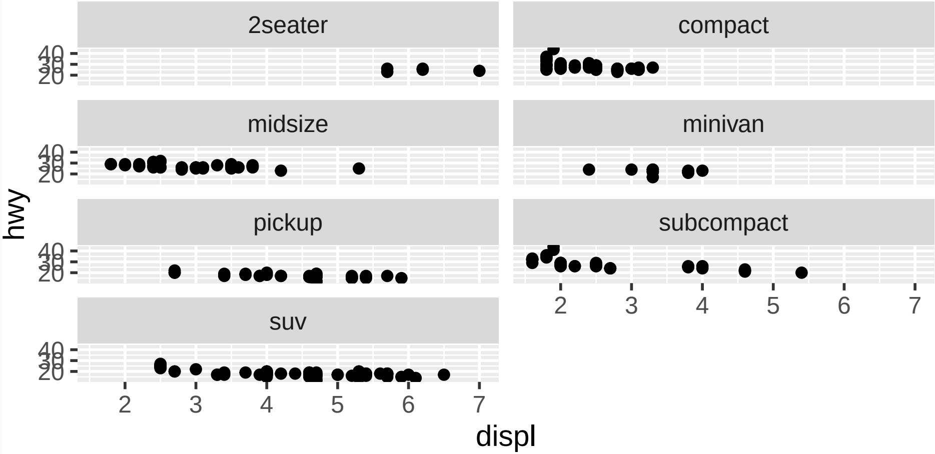

2seater
compact
midsize
minivan
pickup
subcompact
suv
40
30
20
2
3
4
5
6
7
hwy
displ

ggplot2

- Note that we managed to improve the visualization of our dataset compared to what we had before (at least it’s easier to visualize each level of the `class` variable)
- It is possible to change the organization of the panels very easily

```r
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point() +
  facet_wrap(~ class, ncol = 2)
```

ggplot2

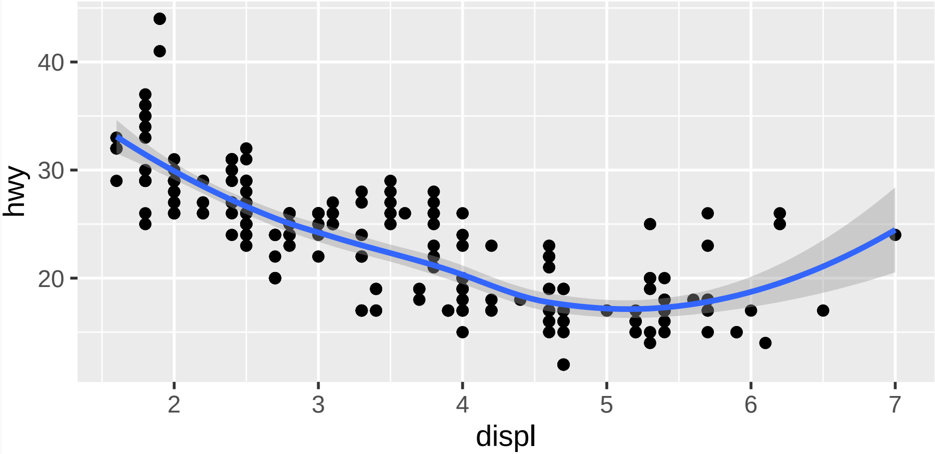

2seater
compact
midsize
minivan
pickup
subcompact
suv
40
30
20
2
3
4
5
6
7
hwy
displ

ggplot2

- How to explain what is happening on the chart below?

```r
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point() +
  geom_smooth()
```

ggplot2



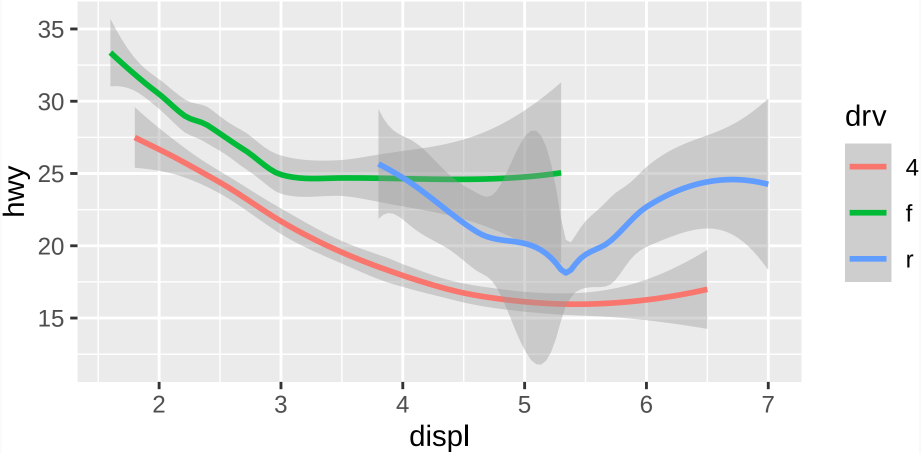
ggplot2

- The graph uses a curve to describe the behavior of the data
- With it, it’s easier to assess trends
- Understand what happens if we want to separate the cars according to their type of vehicle

ggplot2

```
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_smooth(aes(colour = drv))
```

ggplot2

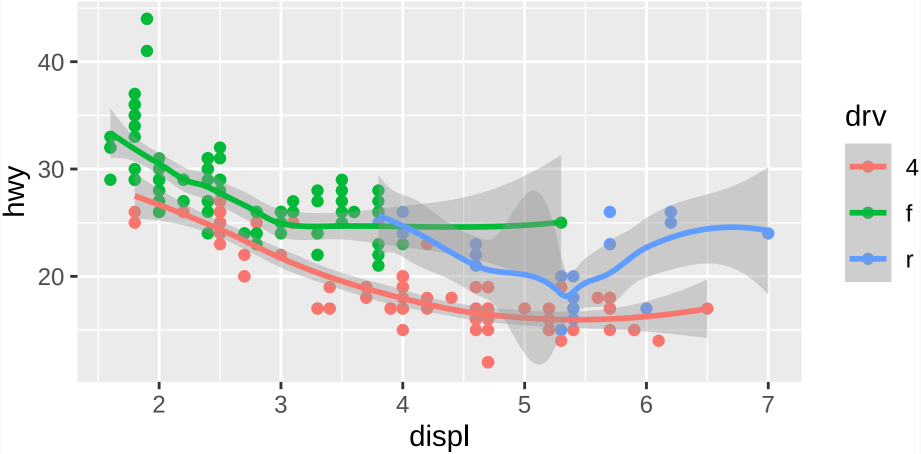


ggplot2

- Note that ggplot2 allows us to combine different geometries on the same graph:

```
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(aes(colour = drv)) +
  geom_smooth(aes(colour = drv))
```

ggplot2

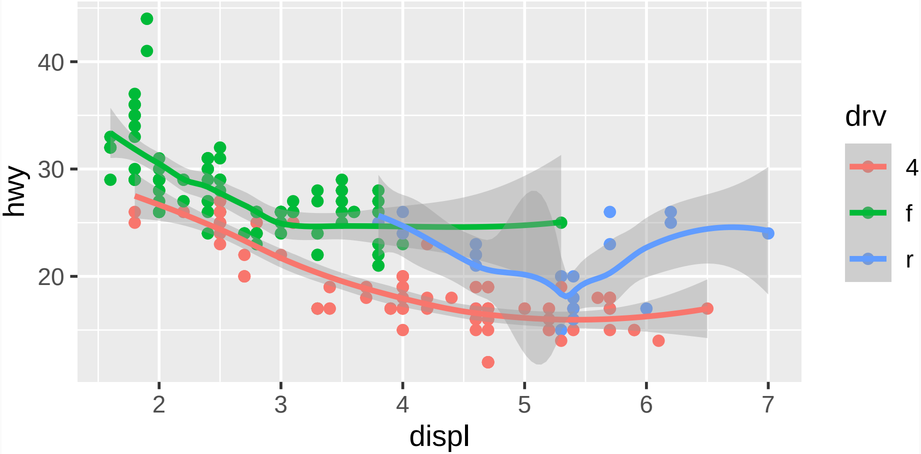


ggplot2

- It is possible to simplify the code above by placing a global declaration for the colors of both the points and the trend curves:

```
ggplot(mpg, aes(x = displ, y = hwy, color = drv)) +
  geom_point() +
  geom_smooth()
```

ggplot2



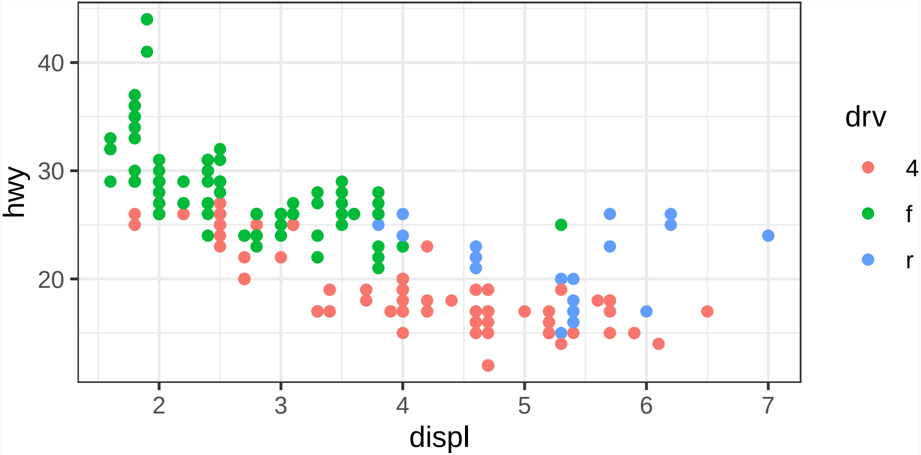
ggplot2

- Let’s assume that the aesthetic result obtained with the graphics we have obtained so far is not to your liking
- For example, suppose you don’t like the gray background
- It’s very easy to change this by applying themes to our graphics

ggplot2

```r
ggplot(mpg, aes(x = displ, y = hwy, colour = drv)) +
  geom_point() +
  theme_bw()
```

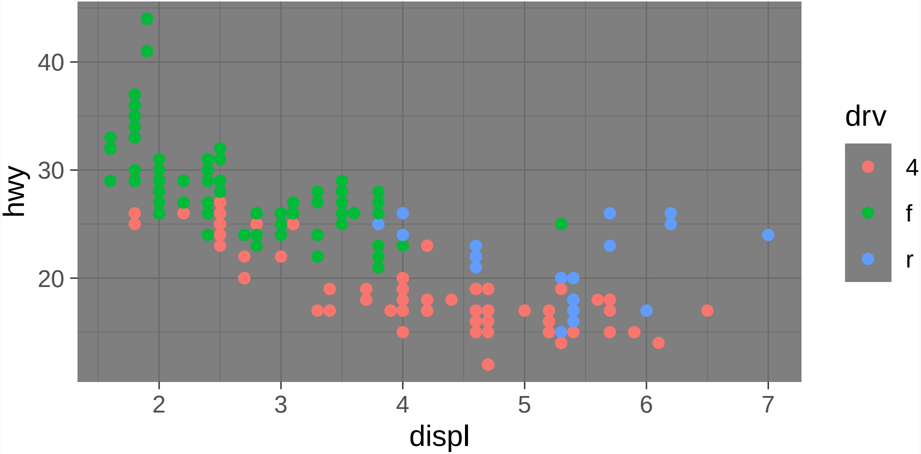
ggplot2



ggplot2

```r
ggplot(mpg, aes(x = displ, y = hwy, colour = drv)) +
  geom_point() +
  theme_dark()
```

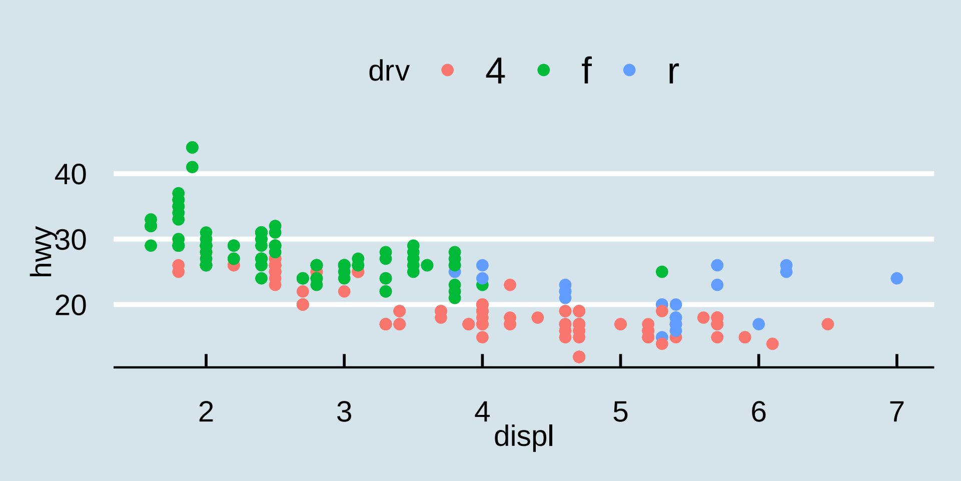
ggplot2



ggplot2

```r
library(ggthemes)

ggplot(mpg, aes(x = displ, y = hwy, colour = drv)) +
  geom_point() +
  theme_economist()
```

ggplot2



Conclusions

Conclusions

- I hope this was a nice introduction to `ggplot` and R language
- With a proper instructor and material, learn R is simpler than it seems

Contact

Contact

- Marcus Nunes
- Email: `marcus@marcusnunes.me`