R for data analysis and visualization

Andrés L. Parrado, Krishanu Chakraborty

February 5, 2019

Section 1: To start off

Disclaimer

- The J-PAL MIT Micromasters - 102x

Disclaimer

- The J-PAL MIT Micromasters - 102x
-

Disclaimer

- The J-PAL MIT Micromasters - 102x
- Creative Commons Attribution-NonCommercial-NoDerivs 3.0 License
- R for Data Science

Disclaimer

- The J-PAL MIT Micromasters - 102x
- Creative Commons Attribution-NonCommercial-NoDerivs 3.0 License
- R for Data Science
- The World Wide Web

Disclaimer

- The J-PAL MIT Micromasters - 102x
- Creative Commons Attribution-NonCommercial-NoDerivs 3.0 License
- R for Data Science
- The World Wide Web
- Harry Potter

Disclaimer

- Numerically literate - YES!

Disclaimer

- Numerically literate - YES!
- First time programmer - YES!

Disclaimer

- Numerically literate - YES!
- First time programmer - YES!
- Big data - NO!

Disclaimer

- Numerically literate - YES!
- First time programmer - YES!
- Big data - NO!
- Python, Julia, Scala and friends - NO!

Disclaimer

- Numerically literate - YES!
- First time programmer - YES!
- Big data - NO!
- Python, Julia, Scala and friends - NO!
- Non-rectangular data - NO!

Disclaimer

- Numerically literate - YES!
- First time programmer - YES!
- Big data - NO!
- Python, Julia, Scala and friends - NO!
- Non-rectangular data - NO!
- Not focusing on methods / speed, but just crude examples

A brief history of R (1/3)

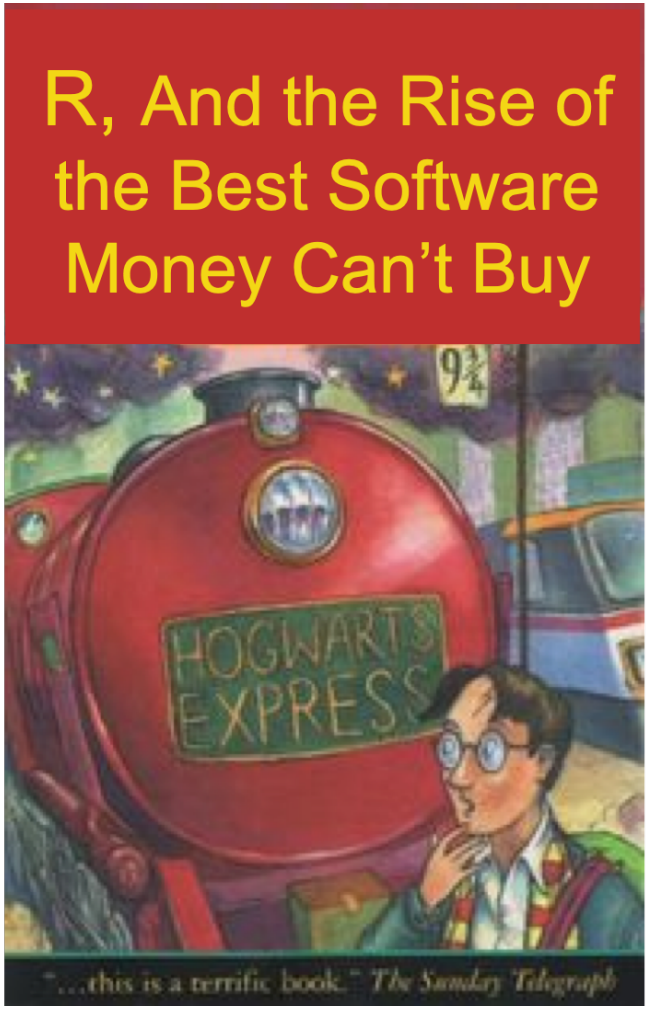


Figure 1: What is R

A brief history of R (2/3)

- S: language for data analysis developed at Bell Labs circa 1976

A brief history of R (2/3)

- S: language for data analysis developed at Bell Labs circa 1976
- Licensed by AT&T/Lucent to Insightful Corp. Product name: S-plus

A brief history of R (2/3)

- S: language for data analysis developed at Bell Labs circa 1976
- Licensed by AT&T/Lucent to Insightful Corp. Product name: S-plus
- R: initially written & released as an open source software by Ross Ihaka and Robert Gentleman at U Auckland during 90s

A brief history of R (2/3)

- S: language for data analysis developed at Bell Labs circa 1976
- Licensed by AT&T/Lucent to Insightful Corp. Product name: S-plus
- R: initially written & released as an open source software by Ross Ihaka and Robert Gentleman at U Auckland during 90s
- Since 1997: international R-core team 20 people & 1000s of code writers and statisticians happy to share their libraries! AWESOME!

A brief history of R (3/3)

- Provides full access to algorithms and their implementation

A brief history of R (3/3)

- Provides full access to algorithms and their implementation
- Gives you the ability to fix bugs and extend software

A brief history of R (3/3)

- Provides full access to algorithms and their implementation
- Gives you the ability to fix bugs and extend software
- Provides a forum allowing researchers to explore and expand the methods used to analyze data

A brief history of R (3/3)

- Provides full access to algorithms and their implementation
- Gives you the ability to fix bugs and extend software
- Provides a forum allowing researchers to explore and expand the methods used to analyze data
- Is the product of 1000s of leading experts in the fields they know best. It is CUTTING EDGE.

A brief history of R (3/3)

- Provides full access to algorithms and their implementation
- Gives you the ability to fix bugs and extend software
- Provides a forum allowing researchers to explore and expand the methods used to analyze data
- Is the product of 1000s of leading experts in the fields they know best. It is CUTTING EDGE.
- Ensures that everyone around the world - and not just ones in rich countries - are the co-owners to the software tools needed to carry out research

A brief history of R (3/3)

- Provides full access to algorithms and their implementation
- Gives you the ability to fix bugs and extend software
- Provides a forum allowing researchers to explore and expand the methods used to analyze data
- Is the product of 1000s of leading experts in the fields they know best. It is CUTTING EDGE.
- Ensures that everyone around the world - and not just ones in rich countries - are the co-owners to the software tools needed to carry out research
- Promotes reproducible research by providing open and accessible tools

A brief history of R (3/3)

- Provides full access to algorithms and their implementation
- Gives you the ability to fix bugs and extend software
- Provides a forum allowing researchers to explore and expand the methods used to analyze data
- Is the product of 1000s of leading experts in the fields they know best. It is CUTTING EDGE.
- Ensures that everyone around the world - and not just ones in rich countries - are the co-owners to the software tools needed to carry out research
- Promotes reproducible research by providing open and accessible tools
- Most of R is written in... R! This makes it quite easy to see what functions are actually doing.

How is R different from Stata (or SPSS/ SAS)

Figure 2: Stata = muggle

How is R different from Stata (or SPSS/ SAS)

- Stata / SAS / SPSS users are like muggles. They are limited in their ability to change their environment. They have to rely on algorithms that have been developed for them. The way they approach a problem is constrained by how Stata/SAS/SPSS employed programmers thought to approach them. And they have to pay money to use these constraining algorithms.

How is R different from Stata (or SPSS/ SAS)

Figure 3: R = wizard

How is R different from Stata (or SPSS/ SAS)

- R users are like wizards. They can rely on functions (spells) that have been developed for them by statistical researchers, but they can also create their own. They don't have to pay for the use of them, and once experienced enough (like Dumbledore), they are almost unlimited in their ability to change their environment.

OOPS I did it again!

- We can do *object oriented programming* in R. In fact, everything in R is an object

OOPS I did it again!

- We can do *object oriented programming* in R. In fact, everything in R is an object
- An object is a data structure having some attributes and methods which act on its attributes

OOPS I did it again!

- We can do *object oriented programming* in R. In fact, everything in R is an object
- An object is a data structure having some attributes and methods which act on its attributes
- Class is a blueprint for the object

OOPS I did it again!

- We can do *object oriented programming* in R. In fact, everything in R is an object
- An object is a data structure having some attributes and methods which act on its attributes
- Class is a blueprint for the object
- We can think of class like a sketch (prototype) of a house

OOPS I did it again!

- We can do *object oriented programming* in R. In fact, everything in R is an object
- An object is a data structure having some attributes and methods which act on its attributes
- Class is a blueprint for the object
- We can think of class like a sketch (prototype) of a house
- Class contains all the details about the floors, doors, windows etc - based on these descriptions we build the house

Learning R - the possibilities

- Making graphs like this

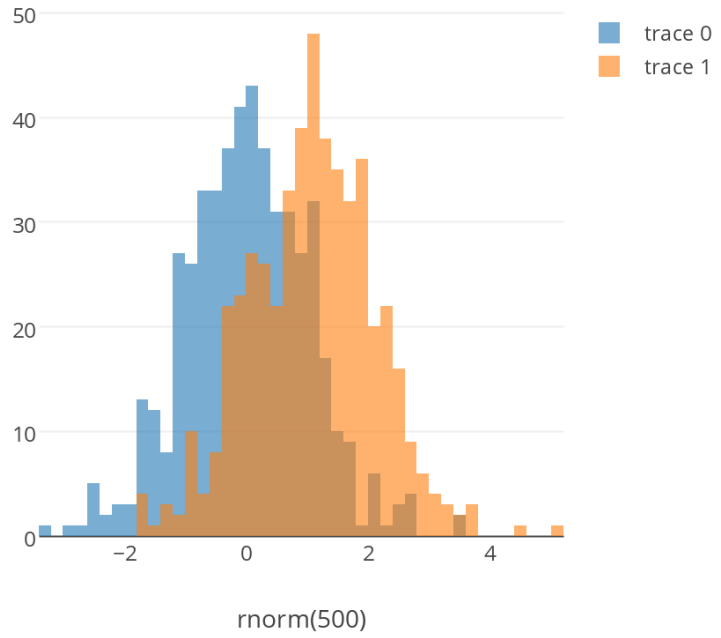


Figure 4: Data in: Graph out

Learning R - the possibilities

- Making a presentation like this!

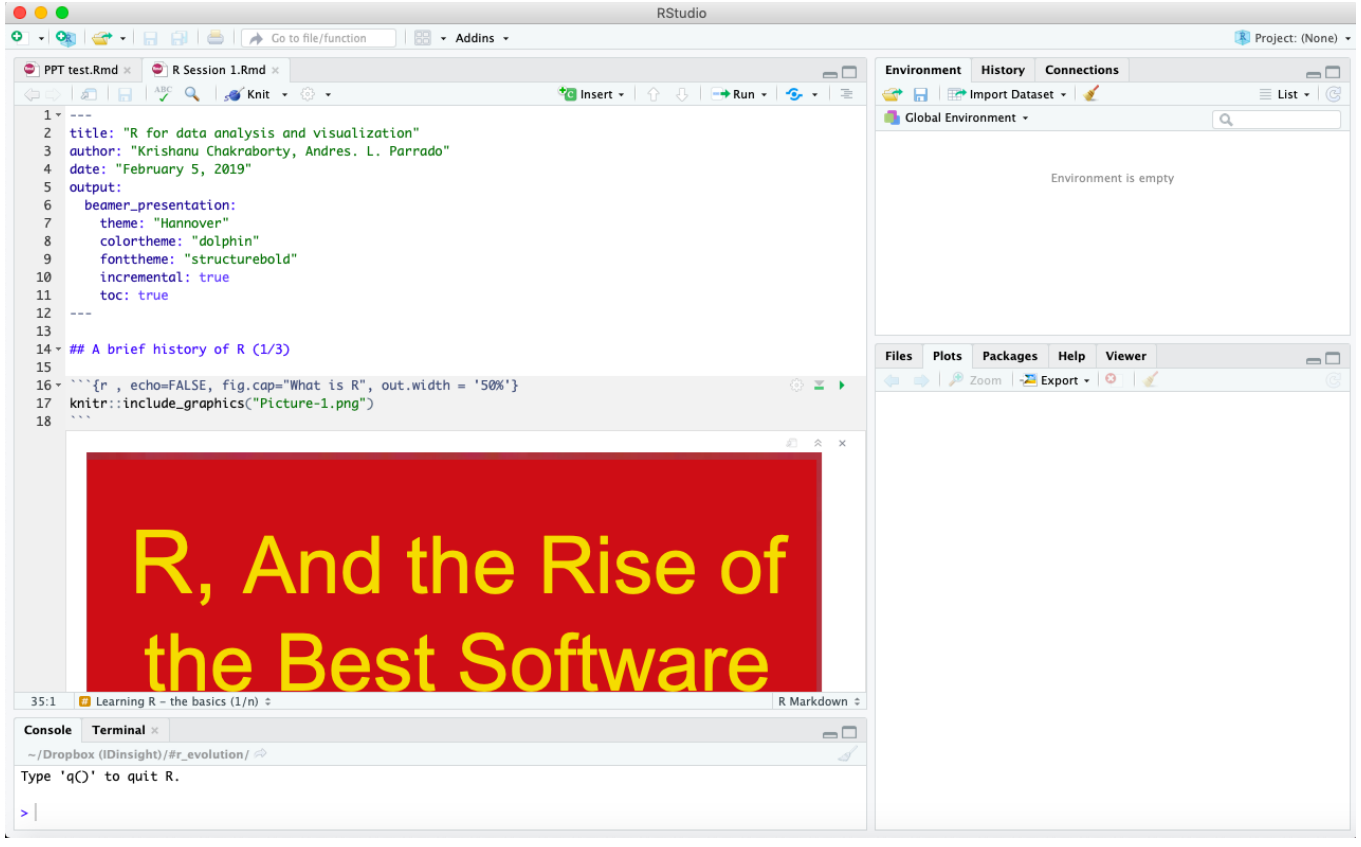


Figure 5: PPT!

Learning R - the possibilities

- Write a book and publish it on a website!

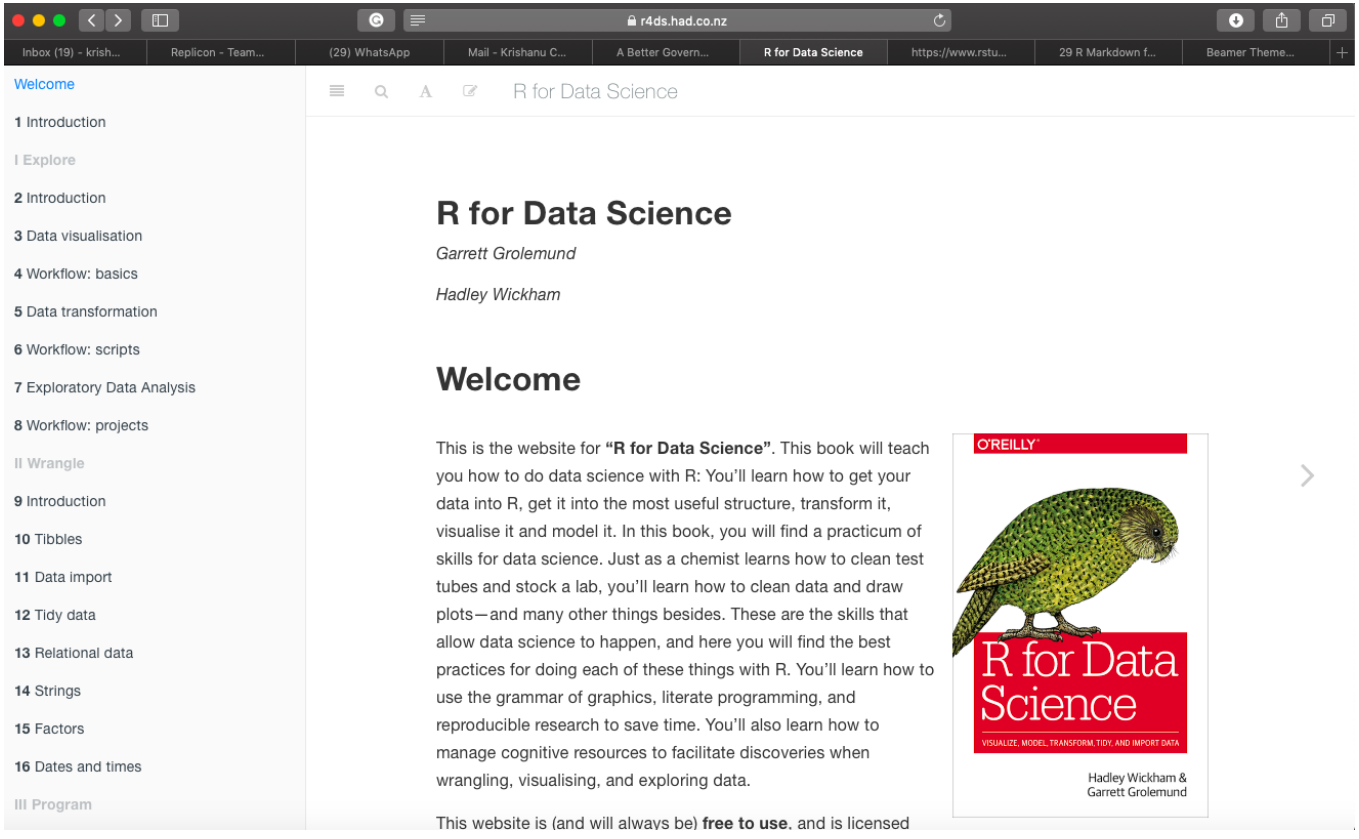


Figure 6: Publish!

Learning R - the possibilities

- Making apps like this

R in action - Here in IDinsight

- We have been using R in the FI project[1]

[1] There are other instances, we acknowledge the limitation of our knowledge

R in action - Here in IDinsight

- We have been using R in the FI project[1]
- Actions *do* speak louder than words

[1] There are other instances, we acknowledge the limitation of our knowledge

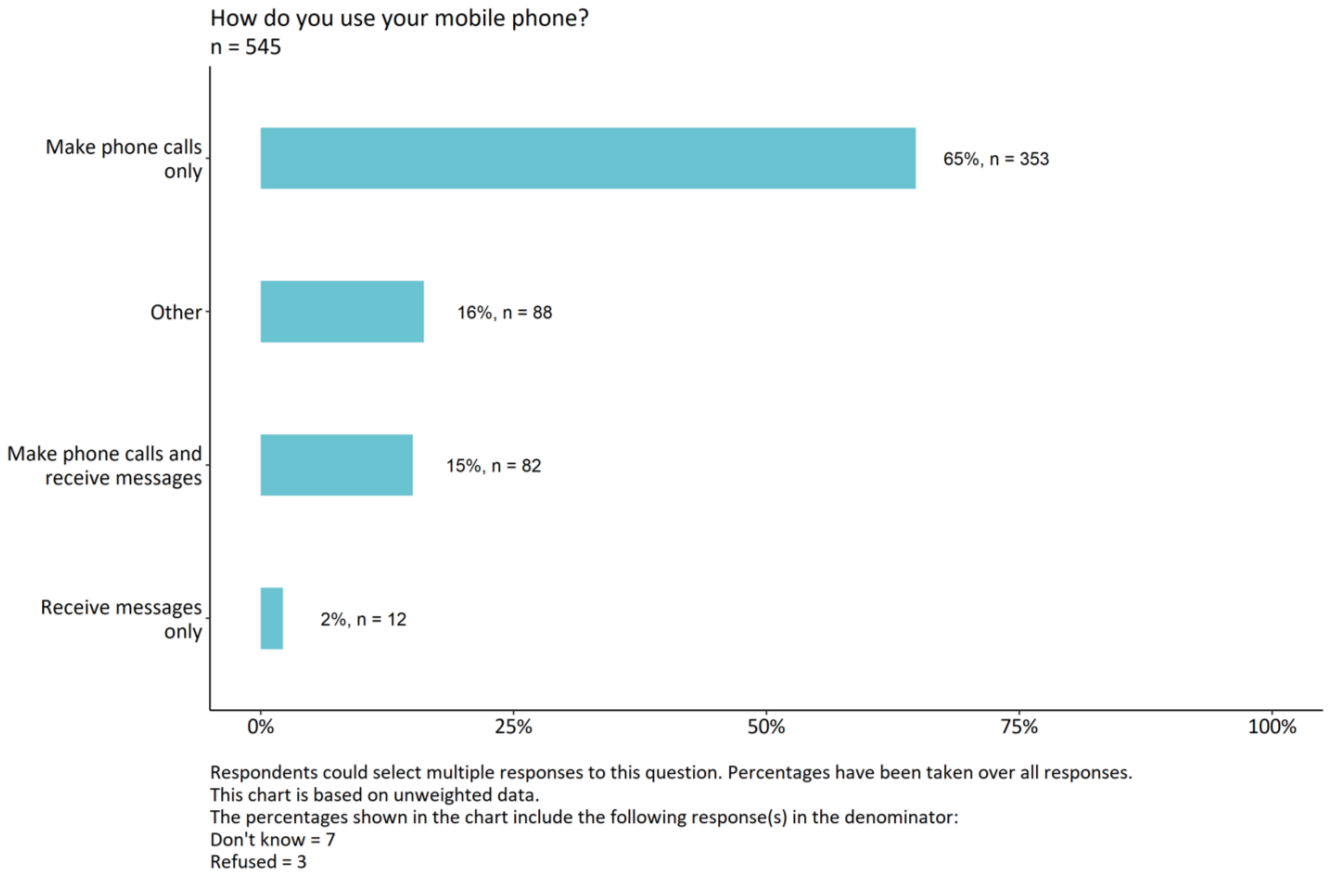
R in action - Here in IDinsight

- We have been using R in the FI project[1]
- Actions *do* speak louder than words
- Interesting experiences

[1] There are other instances, we acknowledge the limitation of our knowledge

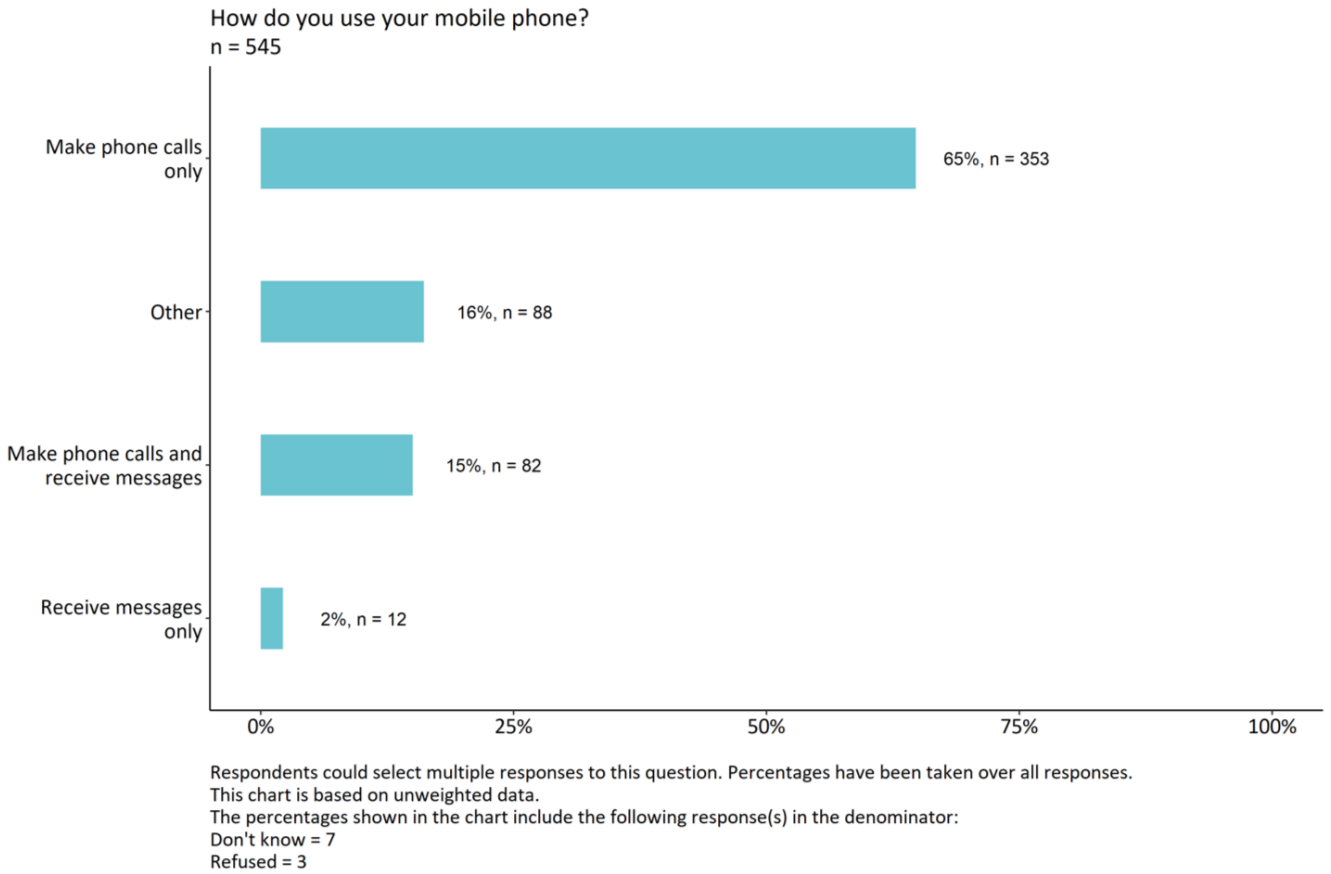
Ishita and a brief stint with the grammar of graphics

- Used Stata to create the chart data and R for the graphs



Ishita and a brief stint with the grammar of graphics

- Used Stata to create the chart data and R for the graphs
- She finds Stata easier for simple tasks but R better for complicated work



Pradyot and his journey to cloud 9

- Most analysis deals with tabular or rectangular data- mostly numeric; Missing from this: Unstructured and text-heavy data

Pradyot and his journey to cloud 9

- Most analysis deals with tabular or rectangular data- mostly numeric; Missing from this: Unstructured and text-heavy data
- Solution: `Tidytext` developed by Silge and Robinson in 2016

Pradyot and his journey to cloud 9

- Most analysis deals with tabular or rectangular data- mostly numeric; Missing from this: Unstructured and text-heavy data
- Solution: `Tidytext` developed by Silge and Robinson in 2016
- You can use existing tools to manipulate, summarise and visualise the characteristics of unstructured and text-heavy data

Pradyot and his journey to cloud 9

- Most analysis deals with tabular or rectangular data- mostly numeric; Missing from this: Unstructured and text-heavy data
- Solution: `Tidytext` developed by Silge and Robinson in 2016
- You can use existing tools to manipulate, summarise and visualise the characteristics of unstructured and text-heavy data
- The tidy text format is defined as a table with one-token-per-row

Pradyot and his journey to cloud 9

- Most analysis deals with tabular or rectangular data- mostly numeric; Missing from this: Unstructured and text-heavy data
- Solution: `Tidytext` developed by Silge and Robinson in 2016
- You can use existing tools to manipulate, summarise and visualise the characteristics of unstructured and text-heavy data
- The tidy text format is defined as a table with one-token-per-row
- A token is a meaningful unit of text, such as a word, that we are interested in using for analysis, and tokenization is the process of splitting text into tokens

Pradyot and his journey to cloud 9

- Most analysis deals with tabular or rectangular data- mostly numeric; Missing from this: Unstructured and text-heavy data
- Solution: `Tidytext` developed by Silge and Robinson in 2016
- You can use existing tools to manipulate, summarise and visualise the characteristics of unstructured and text-heavy data
- The tidy text format is defined as a table with one-token-per-row
- A token is a meaningful unit of text, such as a word, that we are interested in using for analysis, and tokenization is the process of splitting text into tokens
- This one-token-per-row structure is in contrast to the ways text is often stored in current analyses, perhaps as strings or in a document-term matrix

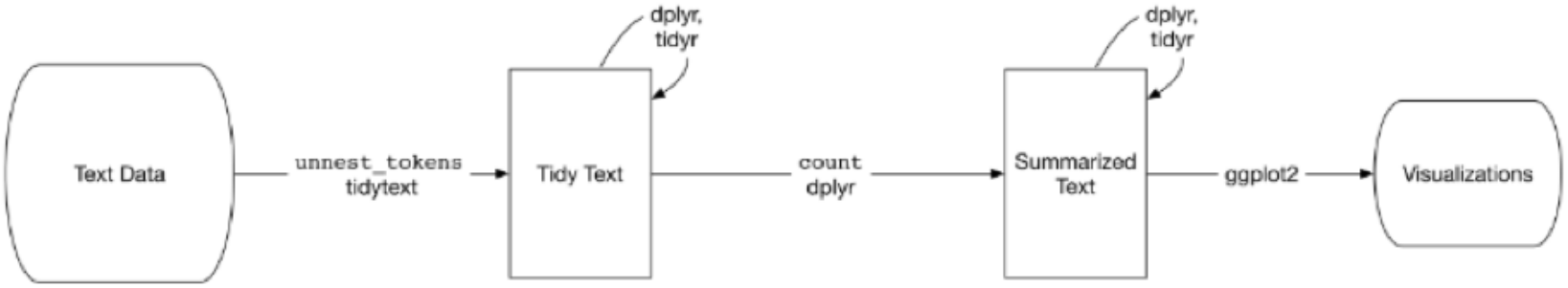


Figure 7: A flowchart of a typical text analysis using tidy data principles.

Andrés and his (in)discretion

- Used to design and analyse discrete choice experiments for preference elicitation

Andrés and his (in)discretion

- Used to design and analyse discrete choice experiments for preference elicitation
- End-to-end, reproducible data analysis, like this one!

Section 2: Wake up Sid!

Learning R - the basics (1/2)

Figure 8: Learning is an adventure

Learning R - the basics (2/2)

- A variable is a quantity, quality, or property that you can measure.

Learning R - the basics (2/2)

- A variable is a quantity, quality, or property that you can measure.
- A value is the state of a variable when you measure it. The value of a variable may change from measurement to measurement.

Learning R - the basics (2/2)

- A variable is a quantity, quality, or property that you can measure.
- A value is the state of a variable when you measure it. The value of a variable may change from measurement to measurement.
- An observation is a set of measurements made under similar conditions (you usually make all of the measurements in an observation at the same time and on the same object). An observation will contain several values, each associated with a different variable. I'll sometimes refer to an observation as a data point.

Learning R - the basics (2/2)

- A variable is a quantity, quality, or property that you can measure.
- A value is the state of a variable when you measure it. The value of a variable may change from measurement to measurement.
- An observation is a set of measurements made under similar conditions (you usually make all of the measurements in an observation at the same time and on the same object). An observation will contain several values, each associated with a different variable. I'll sometimes refer to an observation as a data point.
- Tabular data is a set of values, each associated with a variable and an observation. Tabular data is tidy if each value is placed in its own "cell", each variable in its own column, and each observation in its own row.

Hands on with R

- Step 1: Open up Rstudio

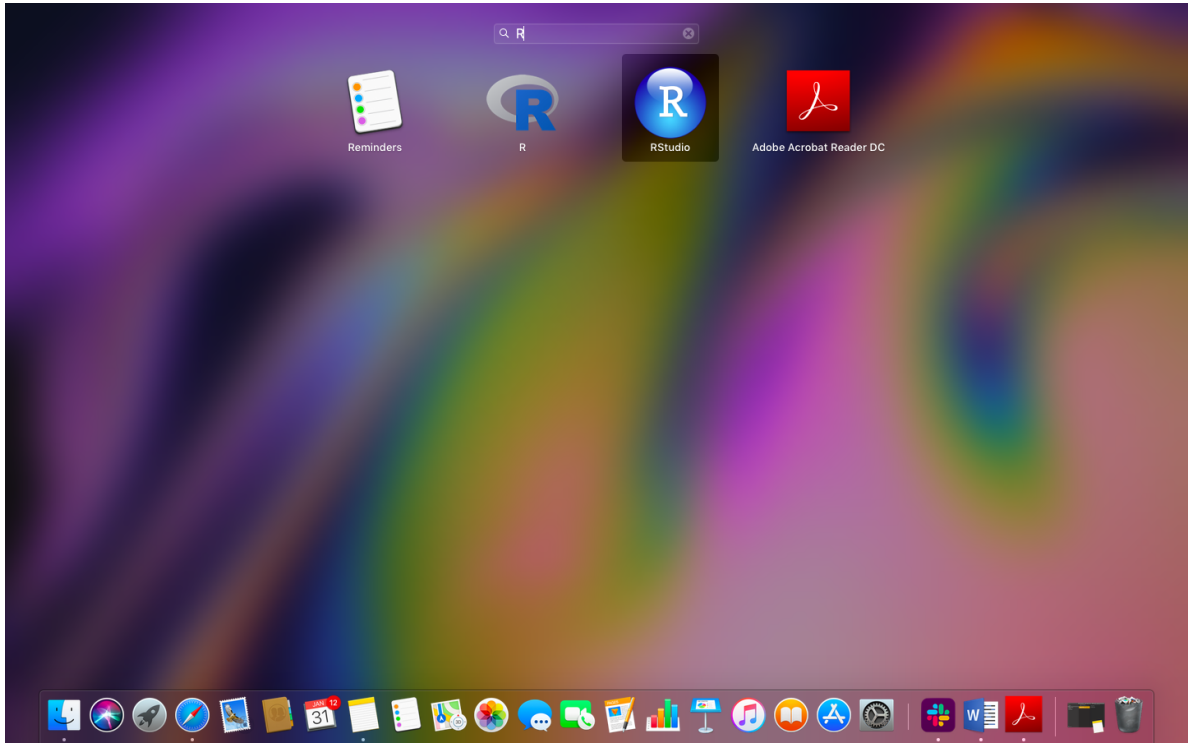


Figure 9: R vs RStudio

Hands on with R

- Step 2: Do you see this?

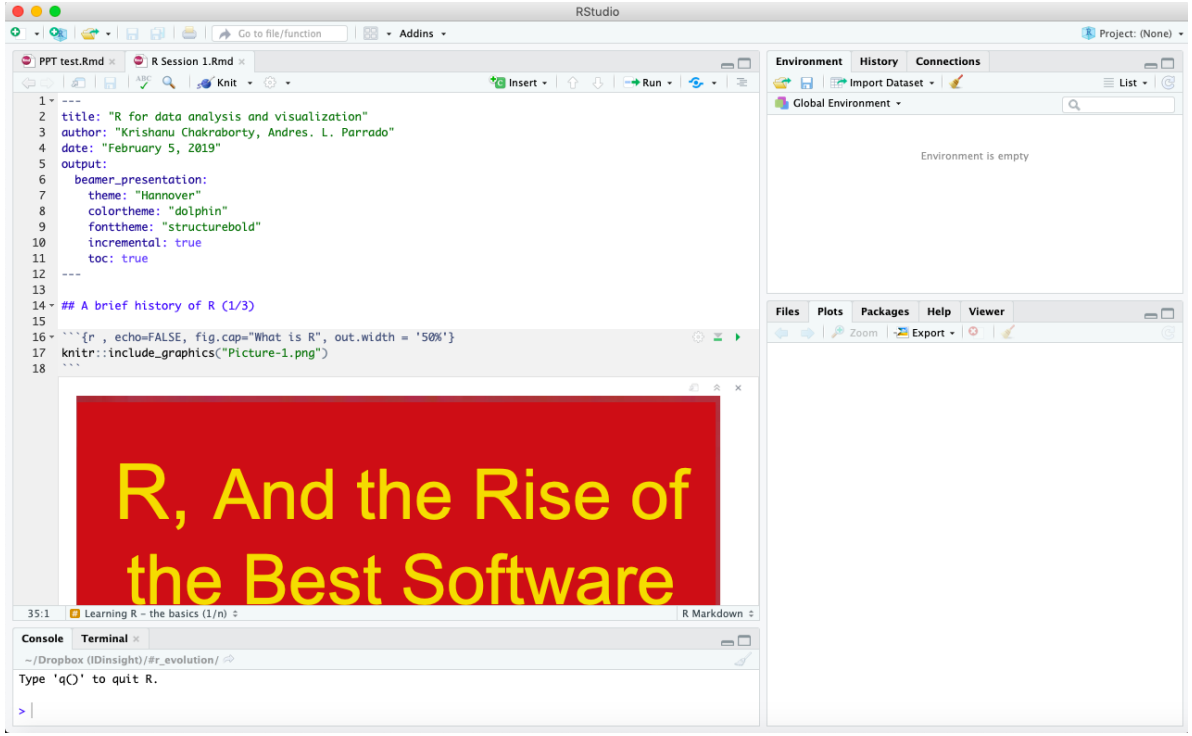


Figure 10: The interface to die for

Baby steps

- Addition

```
40 + 2
```

```
## [1] 42
```

Baby steps

- Multiplication

```
21 * 2
```

```
## [1] 42
```

Baby steps

- Built in functions

```
sqrt(1764)
```

```
## [1] 42
```

Baby steps

- Assignment

```
x <- 42
x
```

```
## [1] 42
```

Baby steps

- Wait! What is `x`?

```
typeof(x)
```

```
## [1] "double"
```

Baby steps

- Wait! What is `typeof()`?

```
help(typeof)
```

Serious business

- An R package is a collection of functions, data, and documentation that extends the capabilities of base R

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

Serious business

- To use a package, use the `library()` function

```
library("tidyverse")
```

```
## -- Attaching packages ------------------------------------------------------------------ tidyverse 1.2.1
```

```
## v ggplot2 3.1.0     v purrr   0.3.0
## v tibble  2.0.1     v dplyr   0.7.8
## v tidyr   0.8.2     v stringr 1.3.1
## v readr   1.3.1     v forcats 0.3.0
```

```
## -- Conflicts --------------------------------------------------------------------- tidyverse_conflicts()
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Serious business

- Please download the package `nycflghts13`

Serious business

- Please download the package `nycflghts13`
- `flights`: information on all 336,776 flights

Serious business

- Please download the package `nycflghts13`
- `flights`: information on all 336,776 flights
- `airlines`: translation between two letter IATA carrier codes and names (16 in total)

Serious business

- Please download the package `nycflghts13`
- `flights`: information on all 336,776 flights
- `airlines`: translation between two letter IATA carrier codes and names (16 in total)
- `planes`: construction information about each of 3,322 planes used

Serious business

- Please download the package `nycflghts13`
- `flights`: information on all 336,776 flights
- `airlines`: translation between two letter IATA carrier codes and names (16 in total)
- `planes`: construction information about each of 3,322 planes used
- `weather`: hourly meteorological data (about 8705 observations) for each of the three NYC airports

Serious business

- Please download the package `nycflghts13`
- `flights`: information on all 336,776 flights
- `airlines`: translation between two letter IATA carrier codes and names (16 in total)
- `planes`: construction information about each of 3,322 planes used
- `weather`: hourly meteorological data (about 8705 observations) for each of the three NYC airports
- `airports`: airport names and locations

Serious business

- Datatset to play with

```
library(nycflights13)
```

Serious business

- The seer

```
View(flights)
glimpse(flights)
```

Serious business

- The special seer

```
nycflights13::airlines$name
```

```
##  [1] "Endeavor Air Inc."           "American Airlines Inc."
##  [3] "Alaska Airlines Inc."        "JetBlue Airways"
##  [5] "Delta Air Lines Inc."        "ExpressJet Airlines Inc."
##  [7] "Frontier Airlines Inc."      "AirTran Airways Corporation"
##  [9] "Hawaiian Airlines Inc."      "Envoy Air"
## [11] "SkyWest Airlines Inc."       "United Air Lines Inc."
## [13] "US Airways Inc."             "Virgin America"
## [15] "Southwest Airlines Co."      "Mesa Airlines Inc."
```

Serious business

- The pipe operator can be read as "then". The %>% operator allows us to go from one step in to the next easily so we can, for example:

```
library(nycflights13)
portland_flights <- flights %>% filter(dest == "PDX")
```

Serious business

- The pipe operator can be read as "then". The %>% operator allows us to go from one step in to the next easily so we can, for example:
- `filter` our data frame to only focus on a few rows then

```
library(nycflights13)
portland_flights <- flights %>% filter(dest == "PDX")
```

Serious business

- The pipe operator can be read as "then". The %>% operator allows us to go from one step in to the next easily so we can, for example:
- `filter` our data frame to only focus on a few rows then
- `group_by` another variable to create groups then

```
library(nycflights13)
portland_flights <- flights %>% filter(dest == "PDX")
```

Serious business

- The pipe operator can be read as "then". The %>% operator allows us to go from one step in to the next easily so we can, for example:
- `filter` our data frame to only focus on a few rows then
- `group_by` another variable to create groups then
- `summarize` this grouped data to calculate the mean for each level of the group.

```
library(nycflights13)
portland_flights <- flights %>% filter(dest == "PDX")
```

Serious business

- Let's filter some more!

```
btv_sea_flights_fall <- flights %>% filter(origin ==
    "JFK", (dest == "BTV" | dest == "SEA"), month >=
    10)
```

Serious business

- Playing around

```
summary_temp <- weather %>% summarize(mean = mean(temp),
    std_dev = sd(temp))
summary_temp
```

```
## # A tibble: 1 x 2
##    mean std_dev
##   <dbl>   <dbl>
## 1    NA      NA
```

Serious business

- Still playing around

```
summary_temp <- weather %>% summarize(mean = mean(temp,
    na.rm = TRUE), std_dev = sd(temp, na.rm = TRUE))
summary_temp
```

```
## # A tibble: 1 x 2
##    mean std_dev
##   <dbl>   <dbl>
## 1  55.3    17.8
```

Serious business

- Further playing around

```
summary_monthly_temp <- weather %>% group_by(month) %>%
    summarize(mean = mean(temp, na.rm = TRUE), std_dev = sd(temp,
        na.rm = TRUE))
summary_monthly_temp
```

```
## # A tibble: 12 x 3
##    month  mean std_dev
##    <dbl> <dbl>   <dbl>
##  1     1  35.6   10.2
##  2     2  34.3    6.98
##  3     3  39.9    6.25
##  4     4  51.7    8.79
##  5     5  61.8    9.68
##  6     6  72.2    7.55
##  7     7  80.1    7.12
##  8     8  74.5    5.19
##  9     9  67.4    8.47
## 10    10  60.1    8.85
## 11    11  45.0   10.4
## 12    12  38.4    9.98
```

Serious business

- Regression!

```
## badly written code
install.packages("stargazer", repos = "http://cran.us.r-project.org")
library(stargazer)

mydata <- mtcars
mydata$fast <- as.numeric((mydata$mpg > 20.1))  #Creating a dummy variable 1 = fast car
m1 <- lm(mpg ~ hp, data = mydata)

stargazer(m1, type = "text", dep.var.labels = c("Miles/(US) gallon",
    "Fast car (=1)"), covariate.labels = c("Gross horsepower",
    "Rear axle ratio", "Four foward gears", "Five forward gears",
    "Type of transmission (manual=1)"), out = "models.txt")
```

Serious business

Let's use our first graph to answer a question: Do cars with big engines use more fuel than cars with small engines? You probably already have an answer, but try to make your answer precise. What does the relationship between engine size and fuel efficiency look like? Is it positive? Negative? Linear? Nonlinear?

- The `mpg` data frame

```
head(mpg)
```

```
## # A tibble: 6 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 manua~ f        21    29 p     comp~
## 3 audi         a4      2    2008     4 manua~ 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 manua~ f        18    26 p     comp~
```

Serious business

Among the variables in `mpg` are:

- `displ`, a car's engine size, in litres.

Serious business

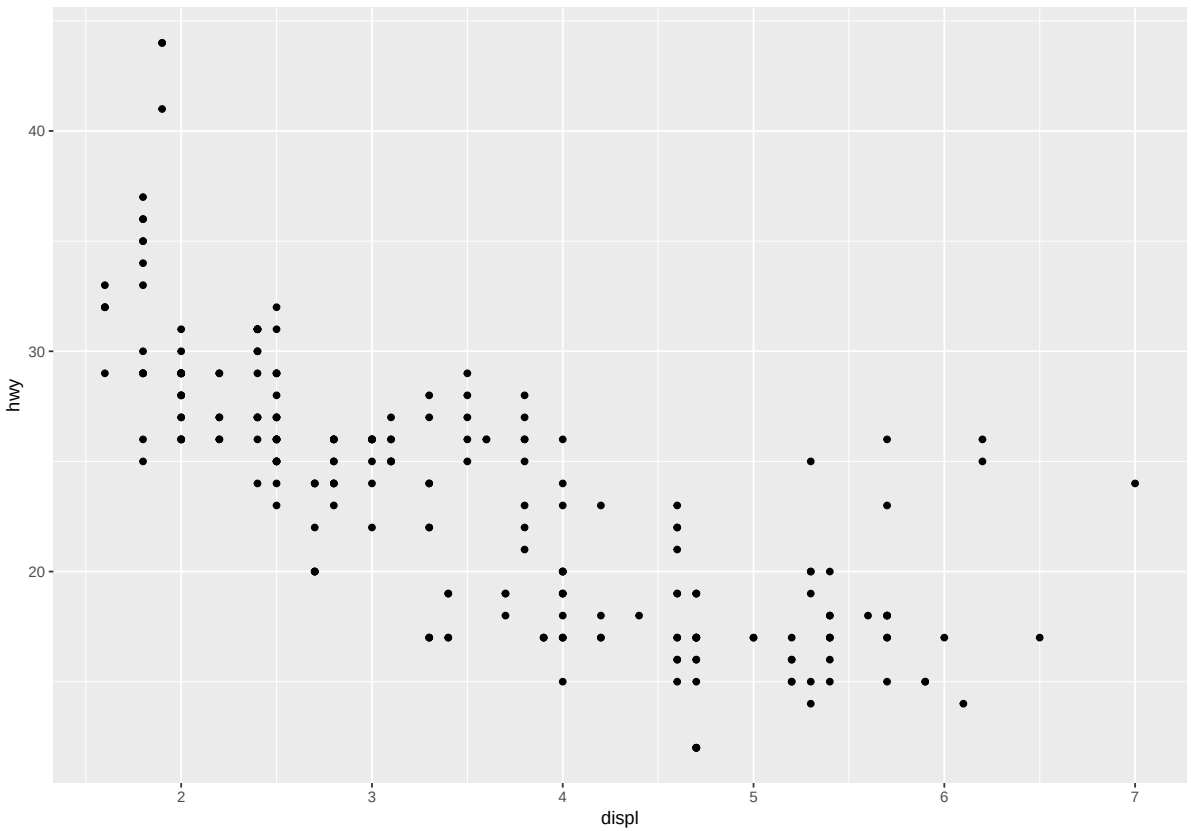
Among the variables in `mpg` are:

- `displ`, a car's engine size, in litres.
- `hwy`, a car's fuel efficiency on the highway, in miles per gallon (mpg). A car with a low fuel efficiency consumes more fuel than a car with a high fuel efficiency when they travel the same distance.

Serious business

To plot `mpg`, run this code to put `displ` on the x-axis and `hwy` on the y-axis:

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



Serious business

- To make a graph, replace the bracketed sections in the code below with a dataset, a geom function, or a collection of mappings.

```
ggplot(data = <DATA>) +
  <GEOM_FUNCTION>(mapping = aes(<MAPPINGS>))
```

Serious business

You can add a third variable, like class, to a two dimensional scatterplot by mapping it to an aesthetic. An aesthetic is a visual property of the objects in your plot. Aesthetics include things like the size, the shape, or the color of your points. You can display a point in different ways by changing the values of its aesthetic properties. Since we already use the word "value" to describe data, let's use the word "level" to describe aesthetic properties.

Serious business

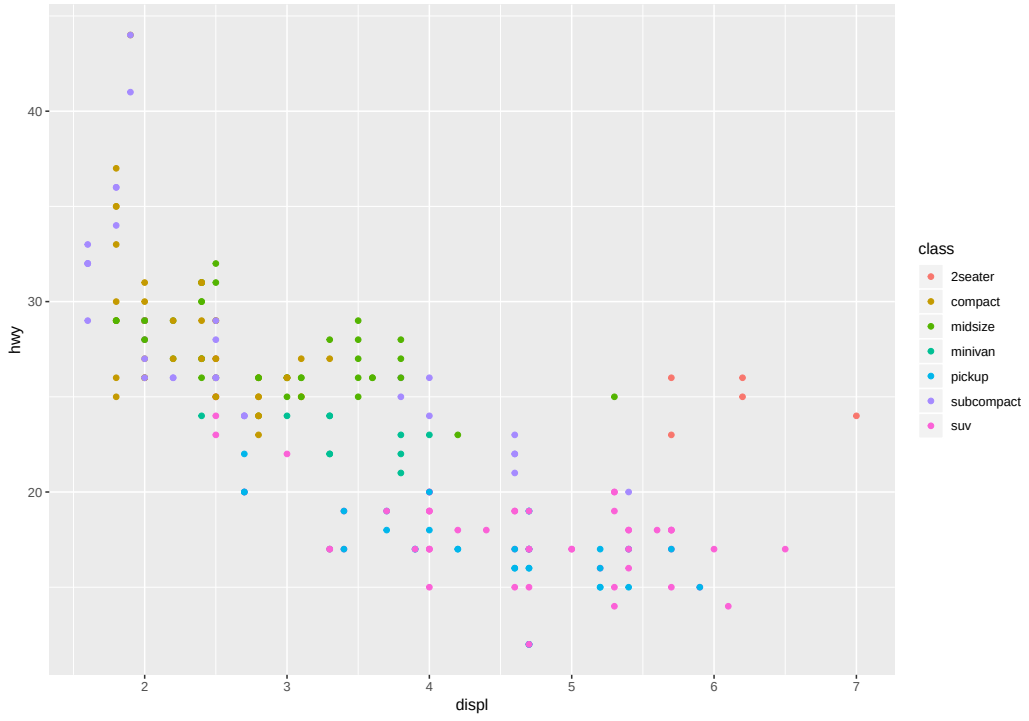
Here we change the levels of a point's size, shape, and color to make the point small, triangular, or blue:

Figure 11: Colors and levels

Serious business

You can map the colors of your points to the `class` variable to reveal the class of each car.

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

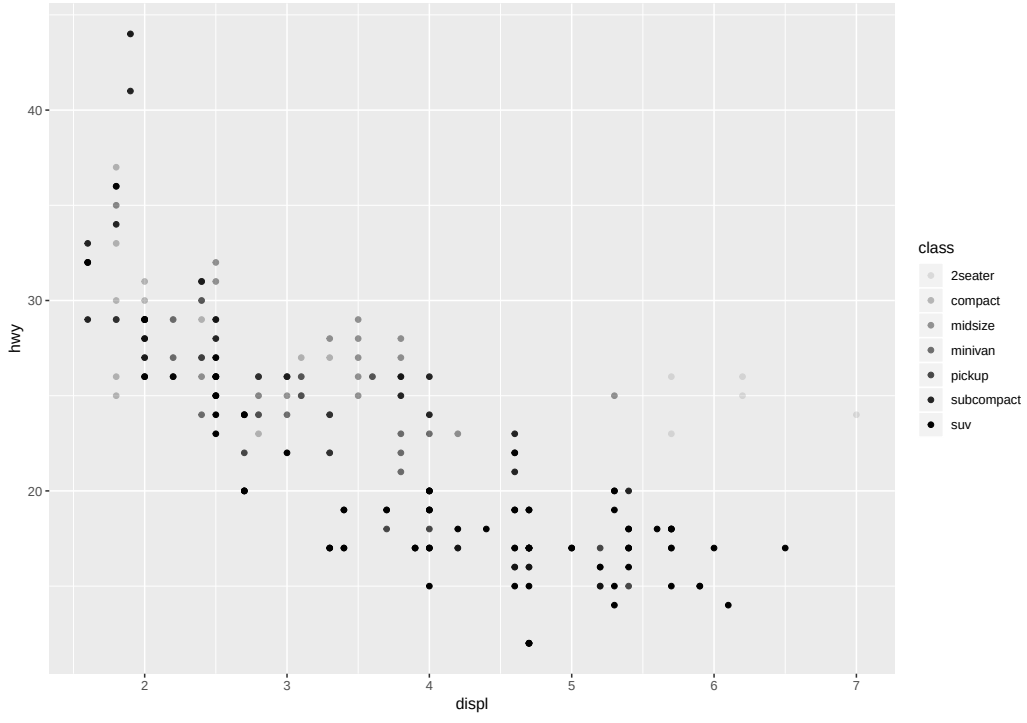


Serious business

We get a warning here, because mapping an unordered variable (`class`) to an ordered aesthetic (`size`) is not a good idea.

```
ggplot(data = mpg) + geom_point(mapping = aes(x = displ,
    y = hwy, alpha = class))
```

```
## Warning: Using alpha for a discrete variable is not advised.
```



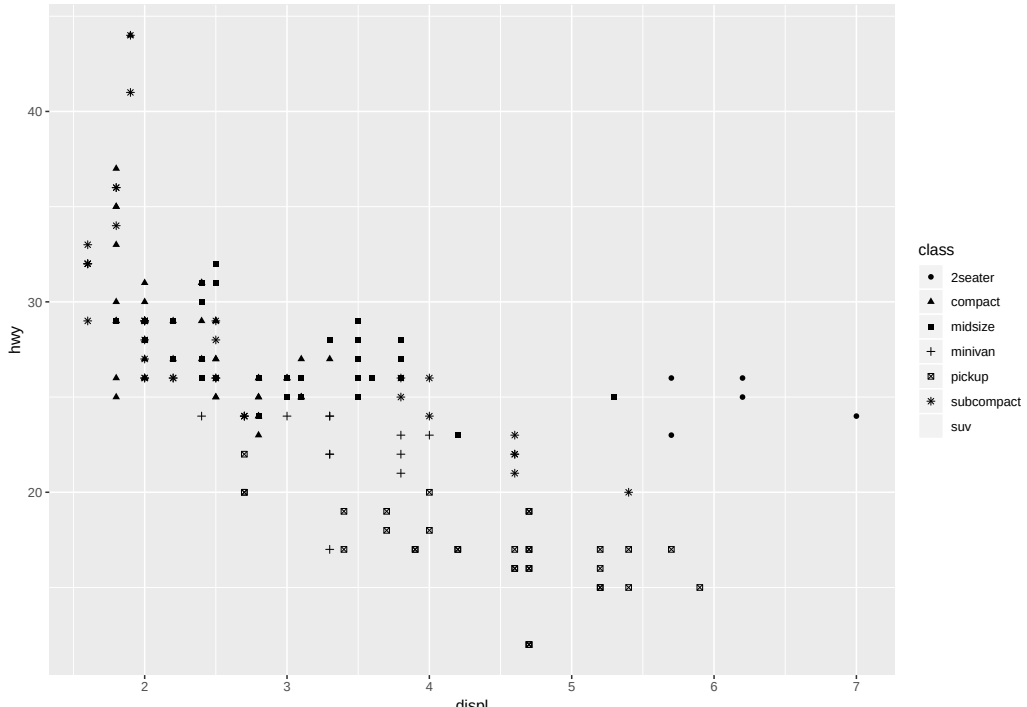
Serious business

What happened to the SUVs?

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

```
## Warning: The shape palette can deal with a maximum of 6 discrete values
## because more than 6 becomes difficult to discriminate; you have 7.
## Consider specifying shapes manually if you must have them.
```

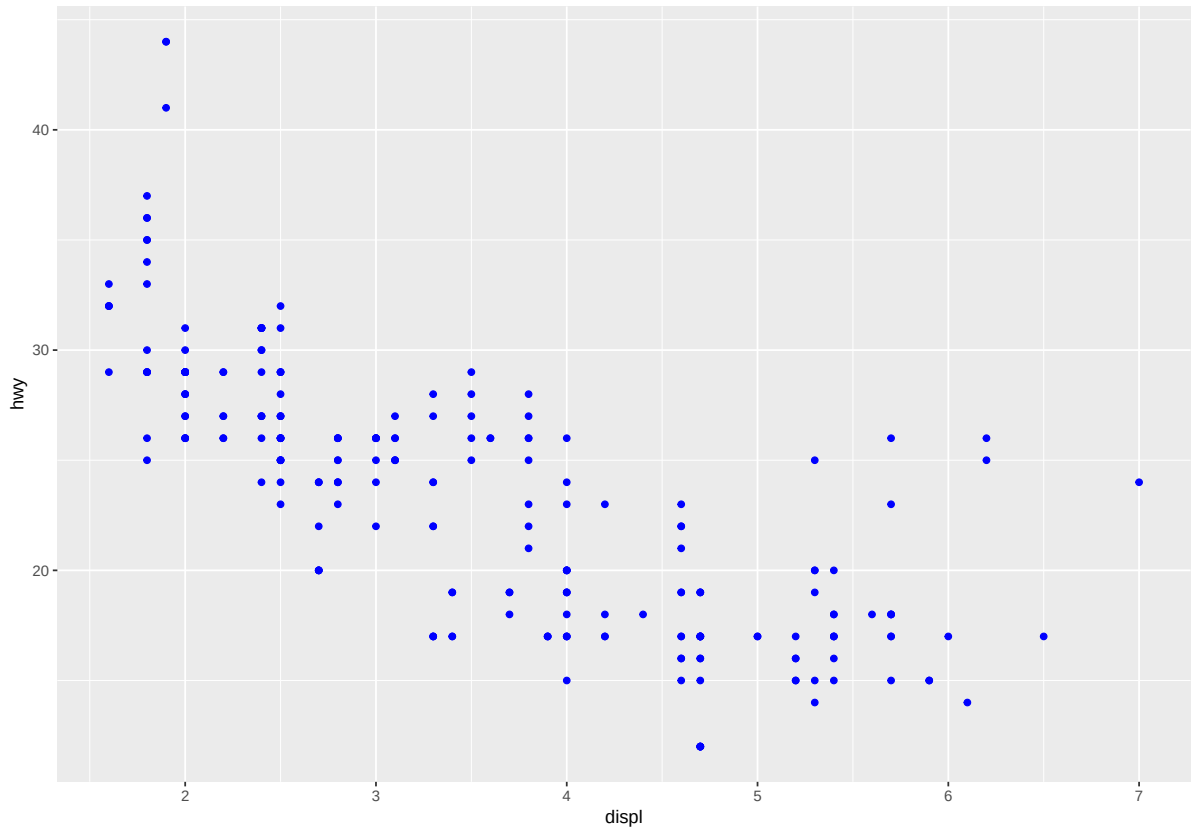
```
## Warning: Removed 62 rows containing missing values (geom_point).
```



Serious business

You can also set the aesthetic properties of your geom manually. For example, we can make all of the points in our plot blue:

```
ggplot(data = mpg) + geom_point(mapping = aes(x = displ,
    y = hwy), colour = "blue")
```

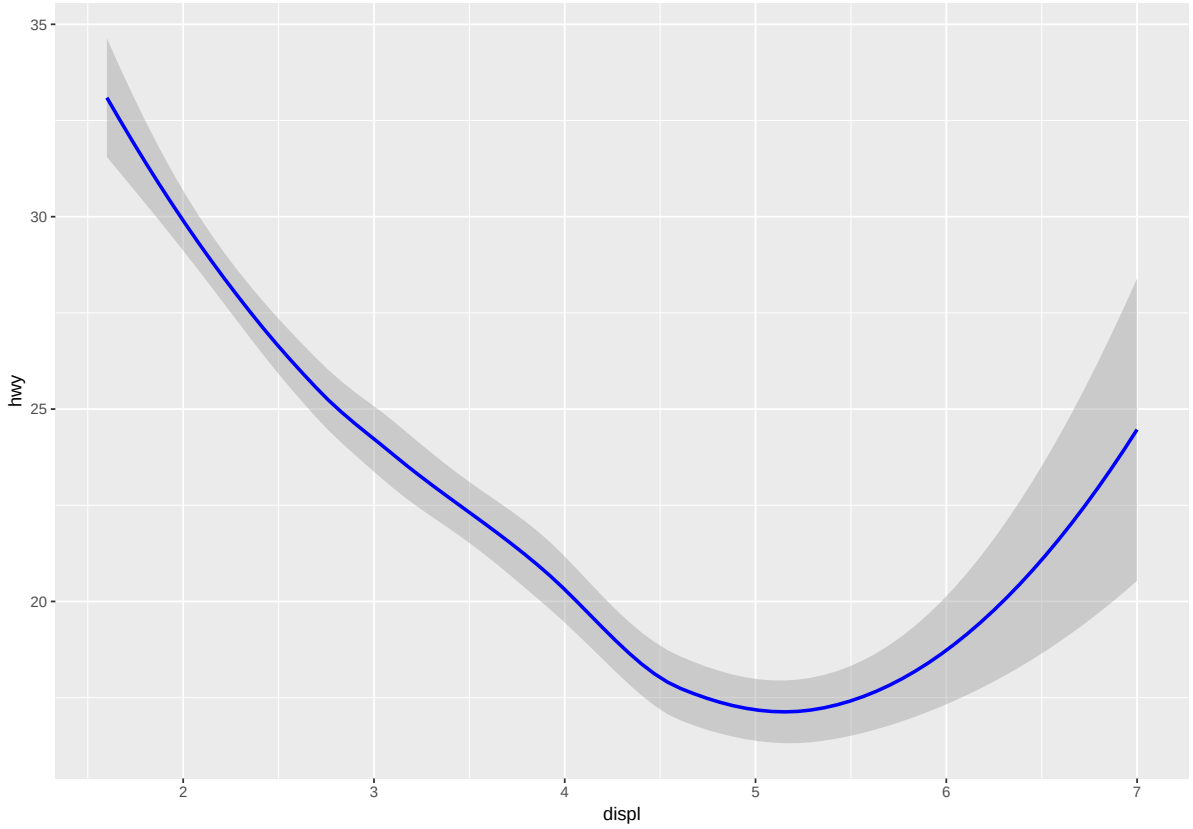


Serious business

Is this plot similar to the previous one?

```
ggplot(data = mpg) + geom_smooth(mapping = aes(x = displ,
    y = hwy), colour = "blue")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

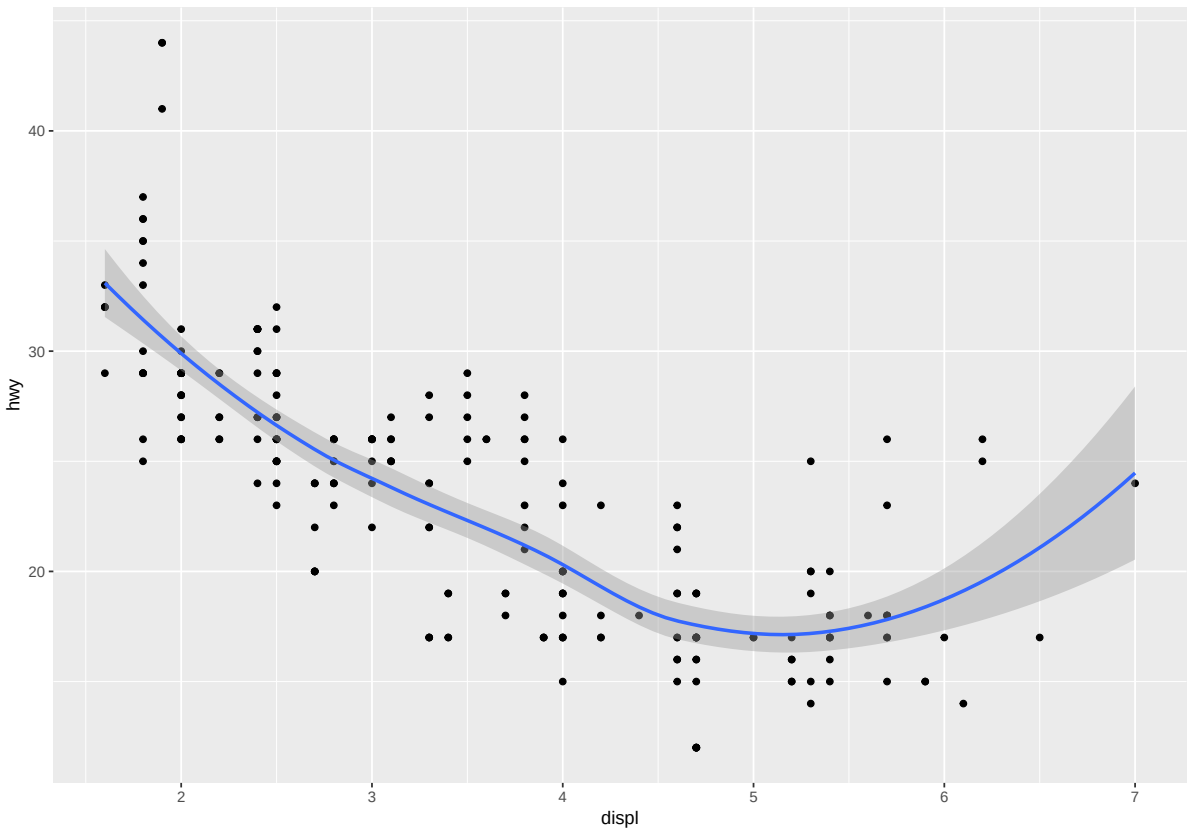


Serious business

What are *geoms*?

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

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

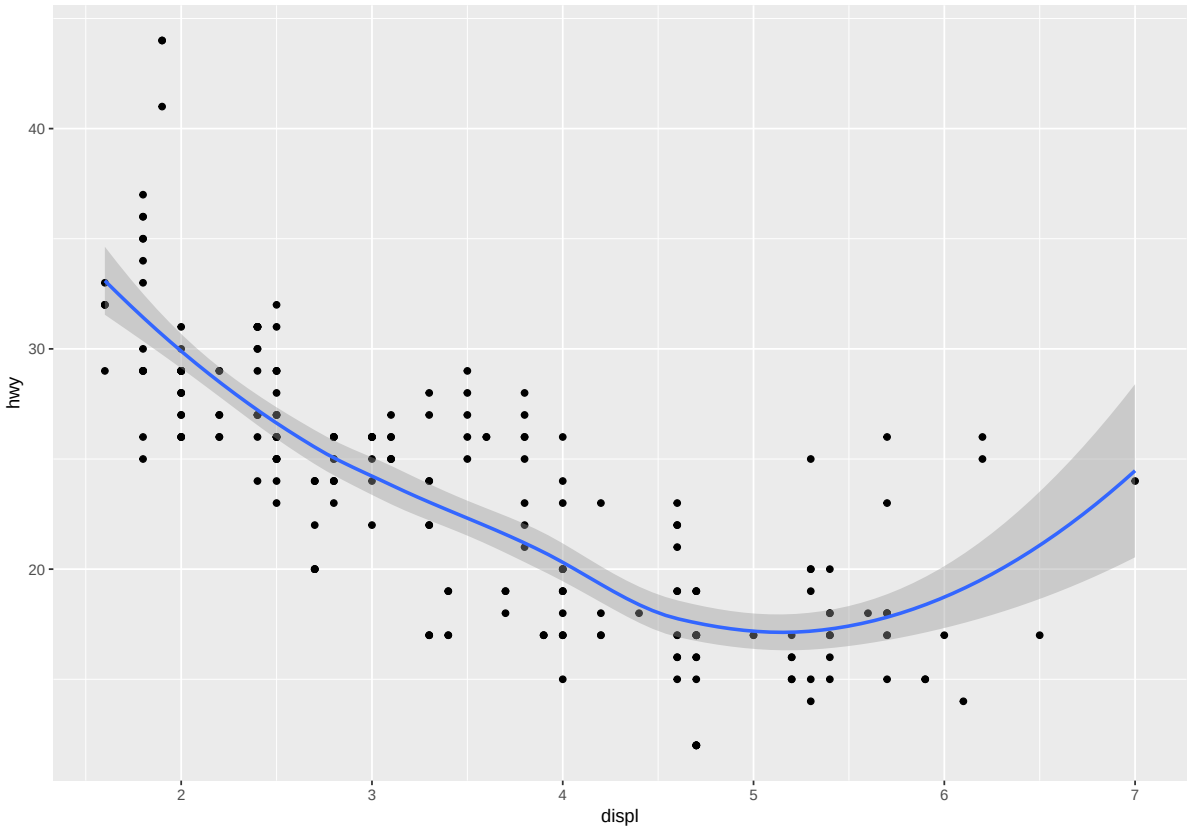


Serious business

Reduced form?

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

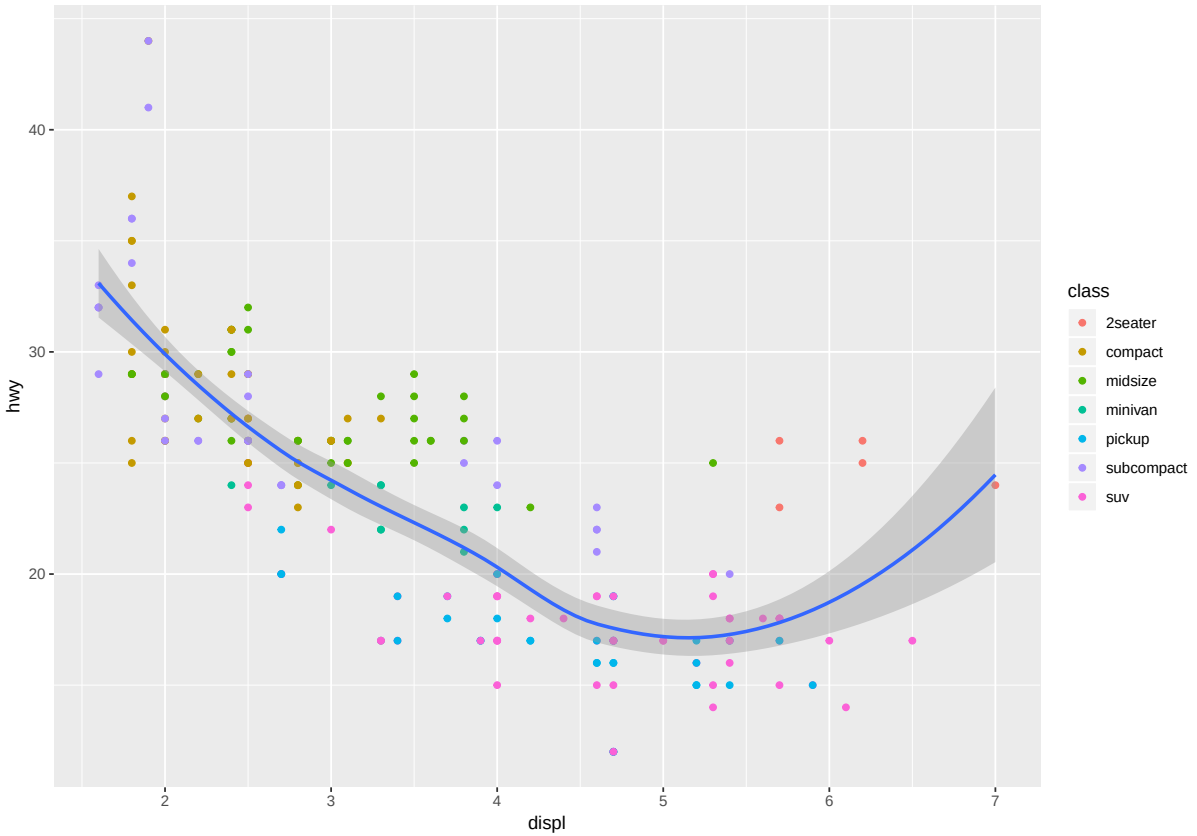
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Serious business

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

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Serious business

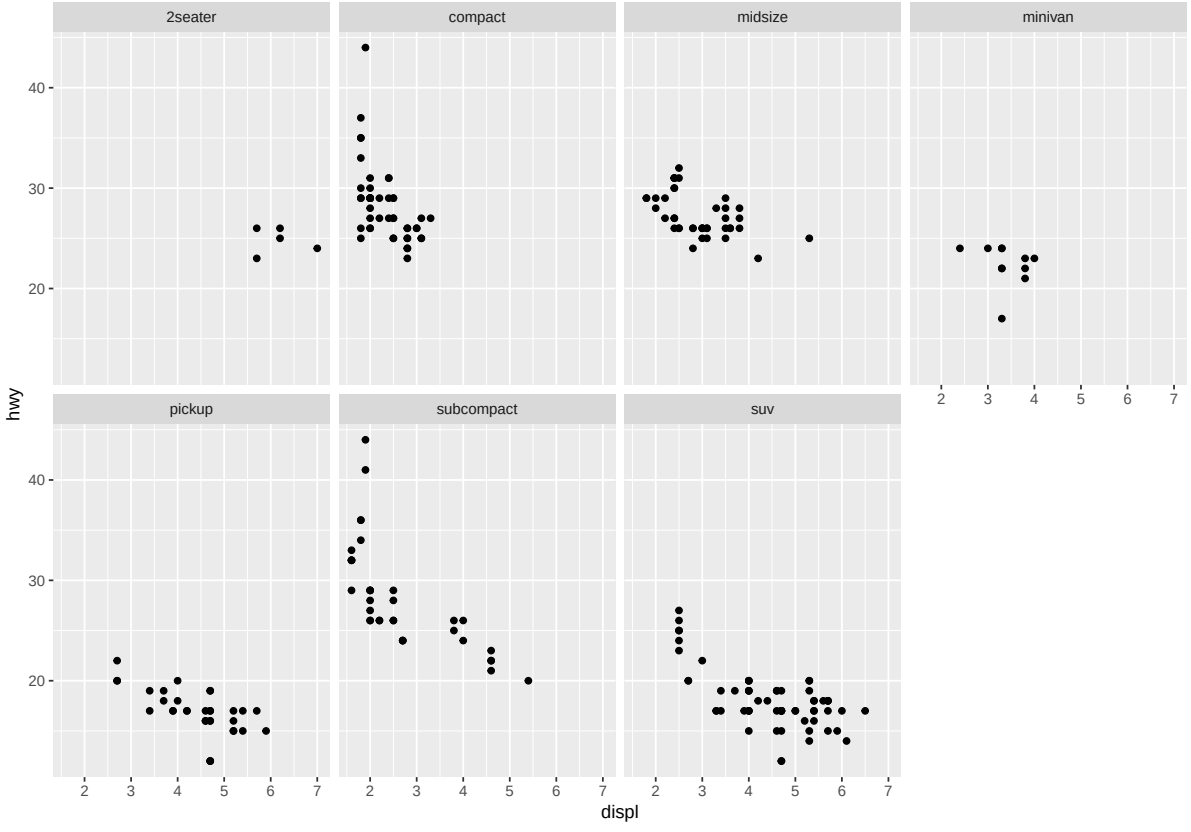
So what is the *grammar of graphics*?

```
ggplot(data = <DATA>) +
  <GEOM_FUNCTION>(
     mapping = aes(<MAPPINGS>),
     stat = <STAT>,
     position = <POSITION>
  ) +
  <COORDINATE_FUNCTION> +
  <FACET_FUNCTION>
```

Serious business

Bonus!

```
ggplot(data = mpg) + geom_point(mapping = aes(x = displ,
    y = hwy)) + facet_wrap(~class, nrow = 2)
```



Section 3: Never say goodbye

What is next?

- Please `swirl`[2]

[2] Meta as hell. 'install.packages("swirl")

What is next?

- Please `swirl`[2]
- Style guide - Google R Style Guide & Hadley Wickham

[2] Meta as hell. 'install.packages("swirl")

What is next?

- Please `swirl`[2]
- Style guide - Google R Style Guide & Hadley Wickham
- If there is need, we can have more sessions like this - focusing on one topic at a time

[2] Meta as hell. 'install.packages("swirl")

What is next?

- Please `swirl`[2]
- Style guide - Google R Style Guide & Hadley Wickham
- If there is need, we can have more sessions like this - focusing on one topic at a time
- If you want some practice on data management and analysis, materials will be shared

[2] Meta as hell. 'install.packages("swirl")

What is next?

- Please `swirl`[2]
- Style guide - Google R Style Guide & Hadley Wickham
- If there is need, we can have more sessions like this - focusing on one topic at a time
- If you want some practice on data management and analysis, materials will be shared
- `#rstats` on twitter

[2] Meta as hell. 'install.packages("swirl")

What is next?

- Please `swirl`[2]
- Style guide - Google R Style Guide & Hadley Wickham
- If there is need, we can have more sessions like this - focusing on one topic at a time
- If you want some practice on data management and analysis, materials will be shared
- `#rstats` on twitter
- Reach out for anything - Remember the name - #r_evolution on Slack

[2] Meta as hell. 'install.packages("swirl")

What is next?

- Please `swirl`[2]
- Style guide - Google R Style Guide & Hadley Wickham
- If there is need, we can have more sessions like this - focusing on one topic at a time
- If you want some practice on data management and analysis, materials will be shared
- `#rstats` on twitter
- Reach out for anything - Remember the name - #r_evolution on Slack
- Want to use R for your project? Reach out![3]

[2] Meta as hell. 'install.packages("swirl")

[3] We charge by the hour

Resources

- Everything is splattered across this Google Drive folder