A Tour Around the Tidyverse World

L. Paloma Rojas-Saunero

2020-06-24

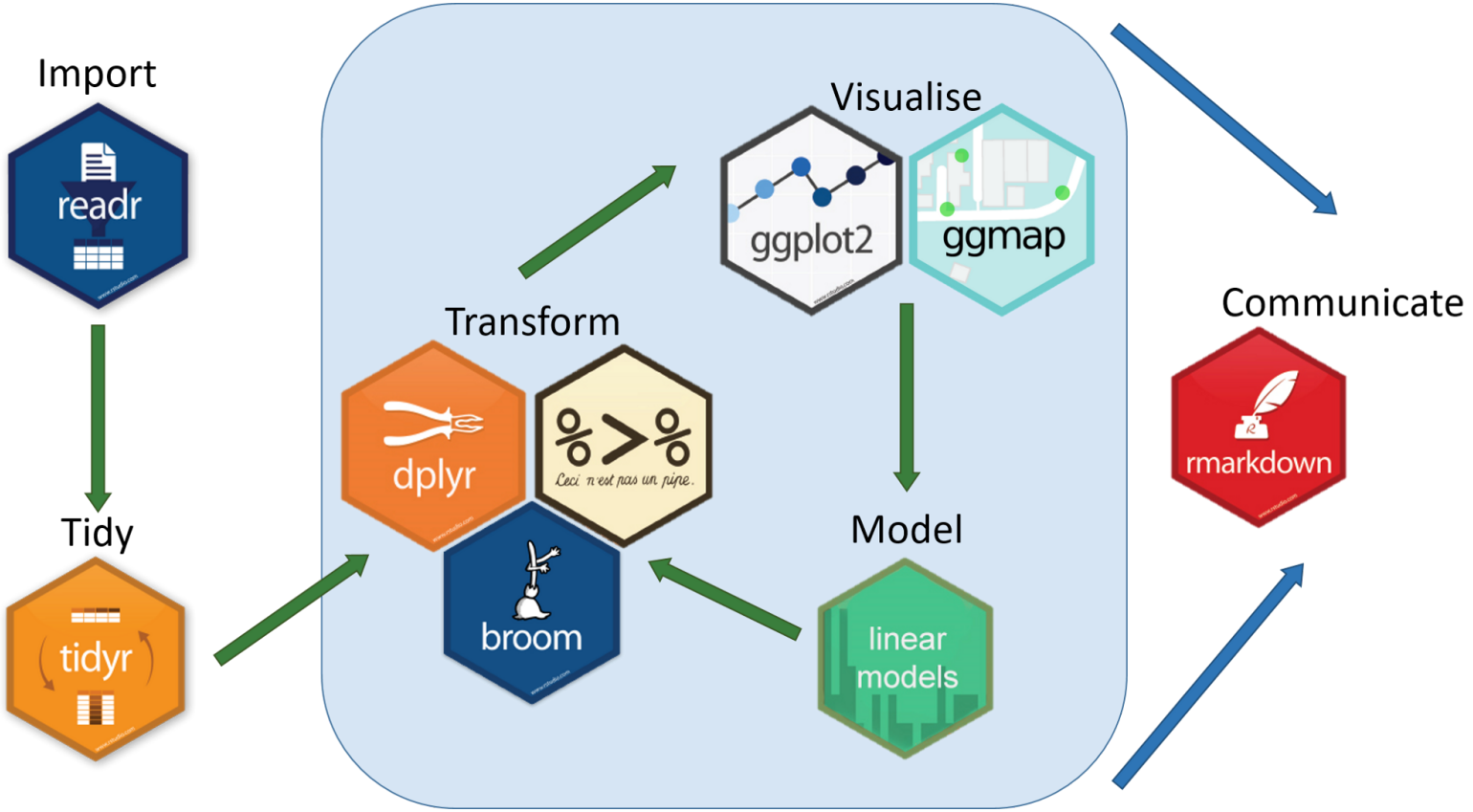
The workshop plan

During the session you will learn:

- How to use the pipe `%>%` operator
- What is the Tidyverse and differences with base R
- How to clean and transform your data with `dplyr`
- What is tidy data and how to make your raw data tidy ✨
- The ggplot grammar 📈📊

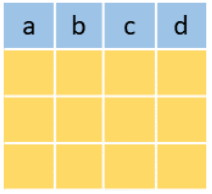
We will use this **shared doc** to communicate (no account needed)

First, what is tidyverse??

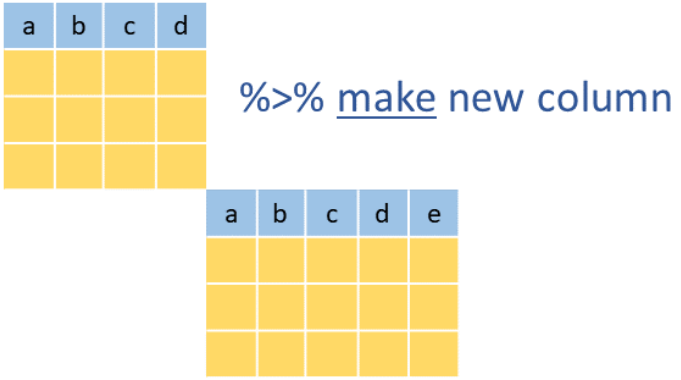


As a set of principles: Human-centered, Consistent, Composable, Inclusive

The Pipe operator %>% *(and then)*

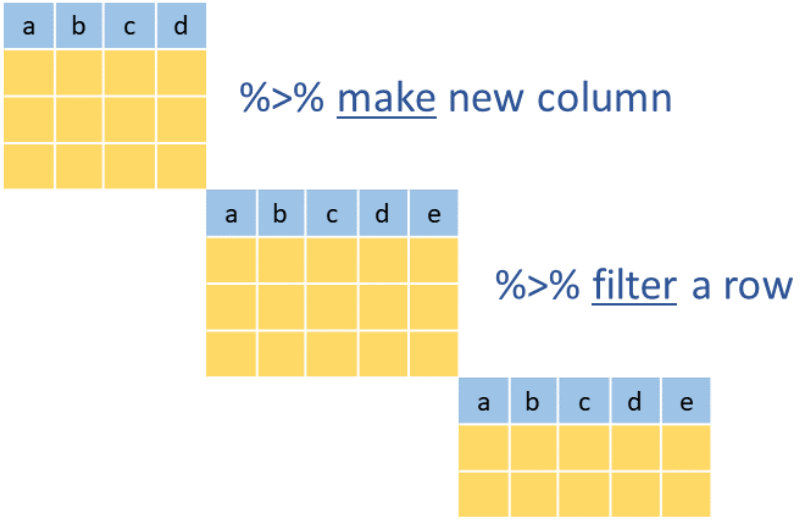


The Pipe operator %>% *(and then)*



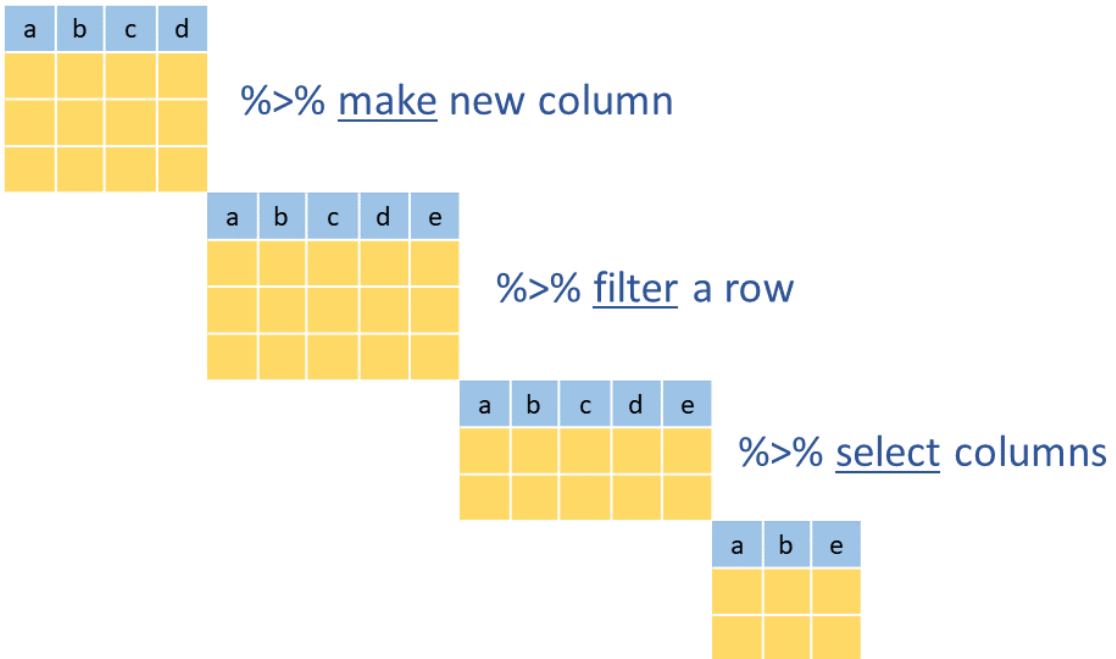
Shortcut: Control/Cmd + shift + m

The Pipe operator %>% *(and then)*



Shortcut: Control/Cmd + shift + m

The Pipe operator %>% *(and then)*



Shortcut: Control/Cmd + shift + m

R base vs. tidyverse

Base R:

```
starwars[starwars$height <200& starwars$gender == "male",]
```

Tidyverse:

```
starwars %>%

filter(height <200, gender == "male")
```

R base vs. tidyverse

Base R:

```
starwars$bmi <- starwars$mass/(starwars$height/100)^2)
```

Tidyverse:

```
starwars %>%

mutate(bmi = mass/((height/100)^2))
```

Hands on!

Best option:

- Sign to Rstudio cloud, and join the **project**.
- Make a **permanent copy** by clicking at right top corner.

Or

- Download the project folder and work on your Rstudio

In any case open: 02_R, workshop.Rmd

- Open the tidyverse library

```
library(tidyverse)
library(here)
```

The dataset

We will use 2 sets of data from the TV series Game of thrones:

1) `got_char.csv`: the total of minutes and seconds in TV per season for each character.

2) `got_houses.csv`: gender and the house each character represents.

#nospoilers

1) Source: benkahle/bayesianGameofThrones

2) Source: Preetish/GoT_screen_time

Import files

Consider type of files:

- We use `read_csv` for files delimited by commas
- Use `read_csv2` for files delimited by semicolon
- If you use data in SPSS, Stata, SAS, the best package is rio
    - It is a Swiss knife that wraps all packages
    - Only one function to import any kind of file: `import()`
    - Only one function to export any kind of file: `export()`

About reproducibility

- AVOID absolute paths or setting/clearing directory using R Projects

```
got_char <-
  read_csv("C:/Users/palol/Dropbox/github/
           tidyverse_workshop_oscr/01_data/got_char.csv")
```

Check this post to understand why Jenny Bryan will come and set your computer on 🔥 if your first line in your R scripts are `setwd("C:\Users\paloma\path\that\only\I\have")` or `rm(list = ls())`

About reproducibility

- Use relative paths

```
got_char <- read_csv("../01_data/got_char.csv")
```

- Even better, use `readr` + `here`

```
got_char <- read_csv(here("01_data", "got_char.csv"))
```

Check this post to understand why to use `here` inside projects

Import with `read_csv`

```
got_char <- read_csv(here("01_data", "got_char.csv"))

got_houses <- read_csv(here("01_data", "got_houses.csv"))
```

Exercise 1. Building the Top 10 within the first 3 seasons

TOP 10 Characters		
Character	Total acting time	House
Tyrion Lannister	167.75	House Lannister
Jon Snow	124.50	Night's Watch
Daenerys Targaryen	123.50	House Targaryen
Arya Stark	98.75	House Stark
Eddard 'Ned' Stark	92.50	House Stark
Sansa Stark	91.50	House Stark
Cersei Lannister	86.25	House Lannister
Catelyn Stark	82.75	House Stark
Theon Greyjoy	79.50	House Greyjoy
Robb Stark	77.75	House Stark

Dplyr package

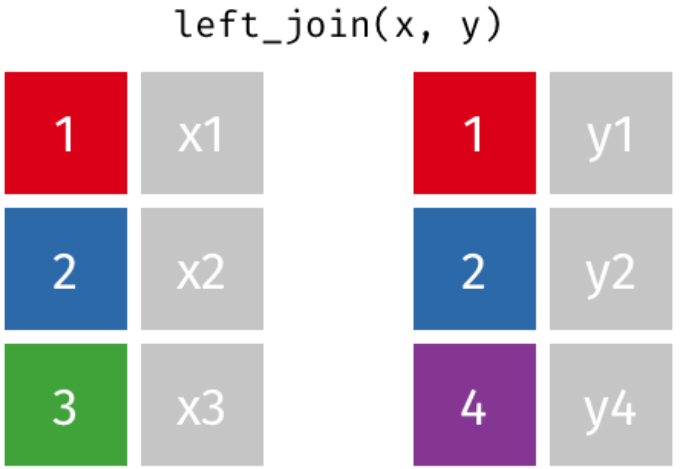
`Dplyr` is a package that provides a set of tools for efficiently manipulating datasets in R.

Today we will learn the following functions:

- mutate
- arrange
- select
- rename
- slice
- left_join (and family of joins)
- group_by
- summarise/summarize

1. Merge the two tables with `left_join`

1. Pick identifier/key variables on each datase
    - got_char = actor
    - got_houses = name
2. Choose how you want to merge



Animated joins by @gadenbuie

2. Make new columns with `mutate`

```
got_complete %>%
  mutate(total = season_1 + season_2 + season_3)
```

- Make new variables

  a) With a specific value

  b) Based on other variables

  c) Change an existing variable

Art by Allison Horst

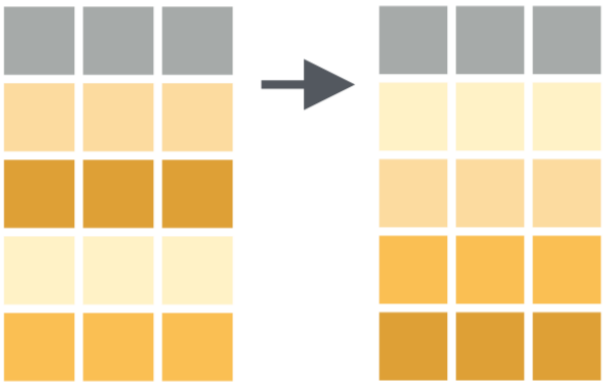
3. Reorganize your rows with `arrange`

Ascending

```
got_complete %>%
  arrange(total)
```

Descending

```
got_complete %>%
  arrange(desc(total))
```



4. Select/remove columns with `select`

```
got_complete %>%
  select(actor, total, house_a)
```

Helpful feats of `select`

a) starts_with("season")

b) contains("hous")

c) matches("_[:digit:]")

d) -actor

e) -c(actor, total)

f) everything()

5. `rename` variables

new name = old name

```
got_complete %>%
  rename(Character = actor,
         House = house_a,
         `Total acting time` = total)
```

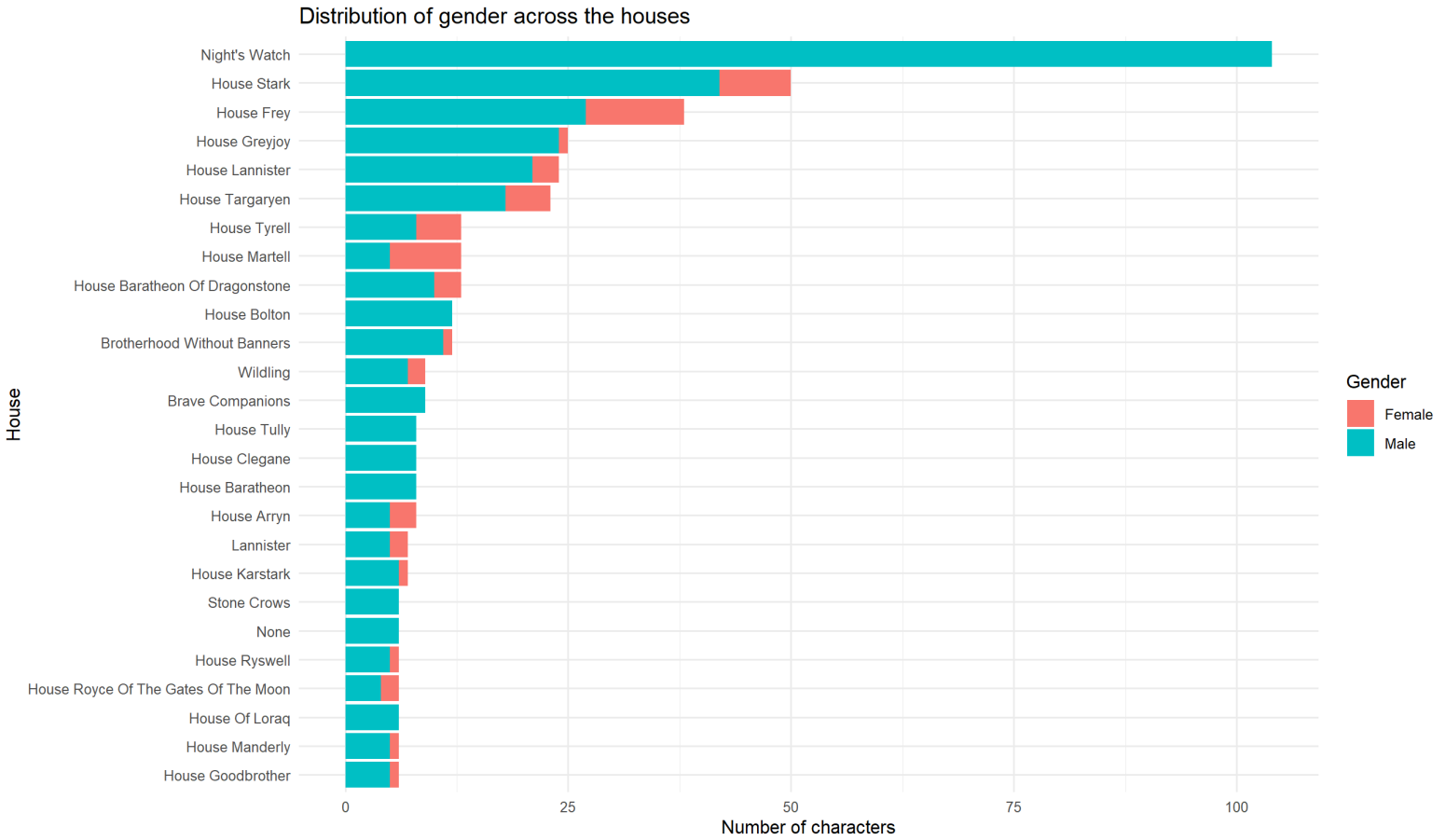
6. `slice` rows

```
got_complete %>%
  slice(1:10)
```

Pipe all steps and...ta da!

```
got_char %>%
  left_join(got_houses, by = c("actor" = "name")) %>%
  mutate(total = (season_1 + season_2 + season_3)) %>%
  arrange(desc(total)) %>%
  select(actor, total, house_a) %>%
  slice(1:10) %>%
  rename(Character = actor,
         House = house_a,
         `Total acting time` = total)
```

Exercise 2. How is the gender distribution across houses?



1. Explore the variables with `count`

```
got_houses %>%
  count(gender)
```

```
got_houses %>%
  count(house_a, sort = TRUE) %>%
  slice(1:4)
```

2. Drop missing observations (rows) with `drop_na`

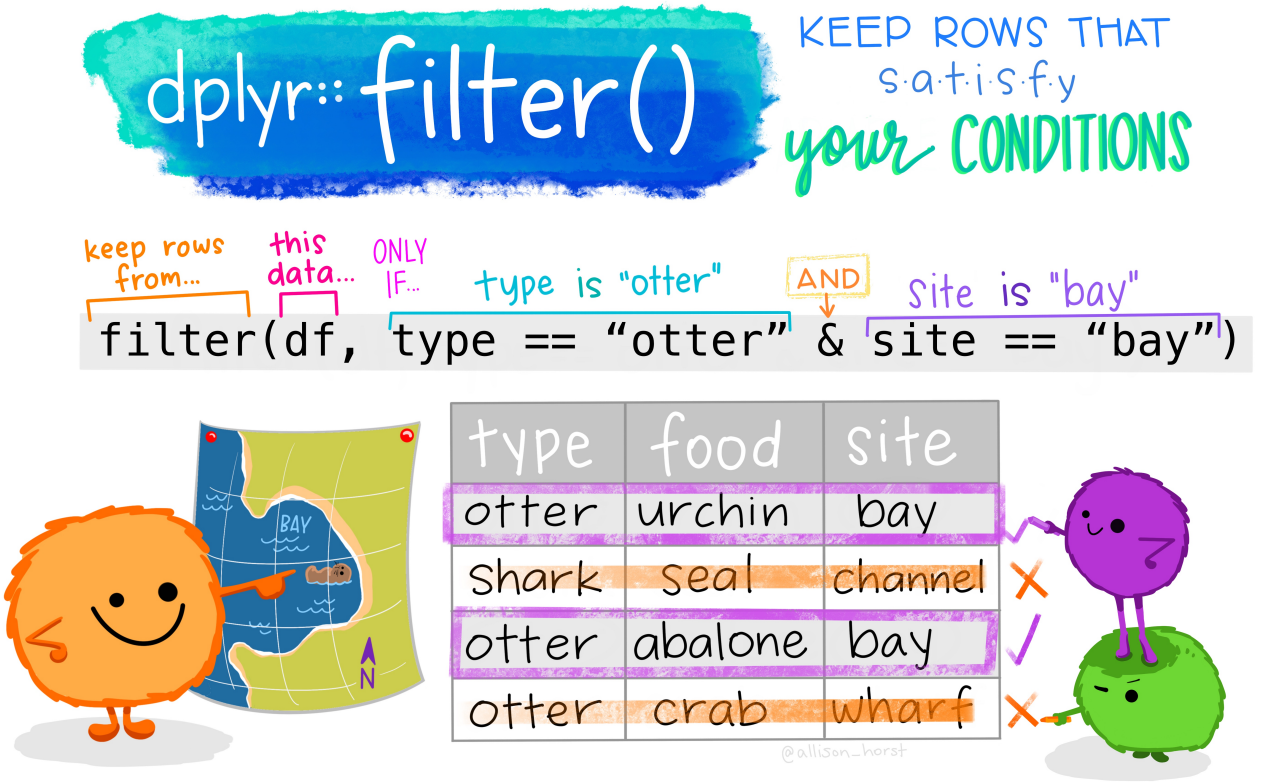
```
got_houses %>%
  drop_na(gender, house_a)
```

3. Do operations within categories of a variable with `group_by`

```
got_houses %>%
  group_by(house_a) %>%
  mutate(n = n()) %>%
  ungroup()
```

n() gives the current group size.

4. `Filter` rows with a criteria



Art by Allison Horst

4. Filter

How would you filter houses that have at least 10 characters?

a) `got_houses %>% filter(n >= 10)`

b) `got_houses %>% filter(n < 10)`

c) `got_houses %>% filter(houses_a >= 10)`

5. Make labels for gender

Using `ifelse`, only 2 conditions

ifelse(`condition`, **TRUE**, FALSE)

```
got_houses %>%
  mutate(gender = ifelse(gender == 0, "Female", "Male"))
```

Tip: Use **`case_when`** for more than two conditions

Data ready to be plotted!

```
got_houses_plot <- got_houses %>%
  drop_na(gender, house_a) %>%
  group_by(house_a) %>%
  mutate(n = n()) %>%
  ungroup() %>%
  filter(n > 10) %>%
  mutate(gender = ifelse(gender == 0, "Female", "Male"))
```

```
got_houses_plot %>%
  head(4)
```

```
## # A tibble: 4 x 4
##   house_a         gender name                                 n
##   <chr>           <chr>  <chr>                            <int>
## 1 House Frey      Male   Aegon Frey (Jinglebell)             38
## 2 House Targaryen Male   Aegon Targaryen                     23
## 3 House Greyjoy   Male   Adrack Humble                       25
## 4 Night's Watch   Male   Aemon Targaryen (son of Maekar I)   104
```

`ggplot` grammar

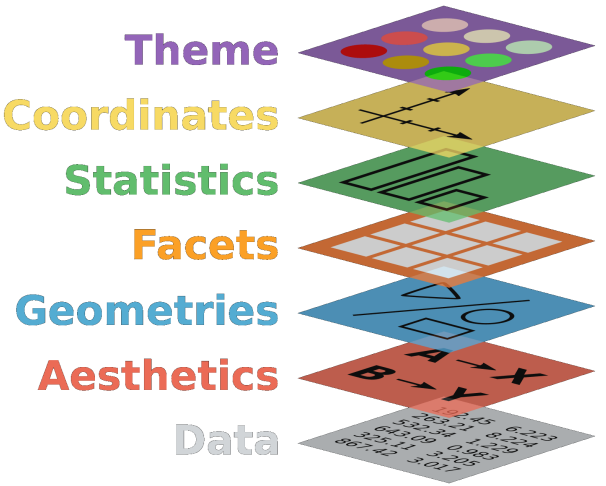
Data = tibble

Aesthetics = variables to be plotted

Geometries = Type of plot

Theme = colors and details

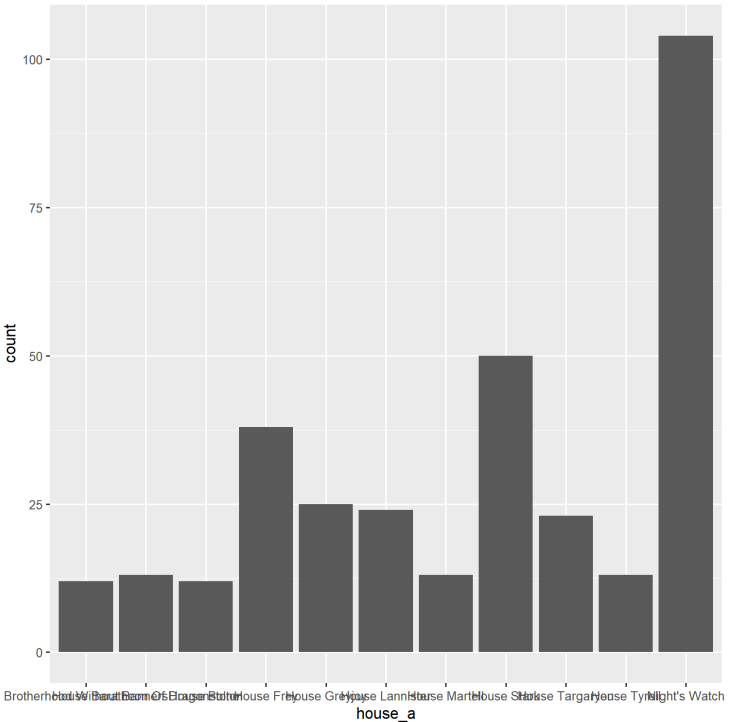
We go from `%>%` for a `+` for each layer



Adapted picture from @CedScherer

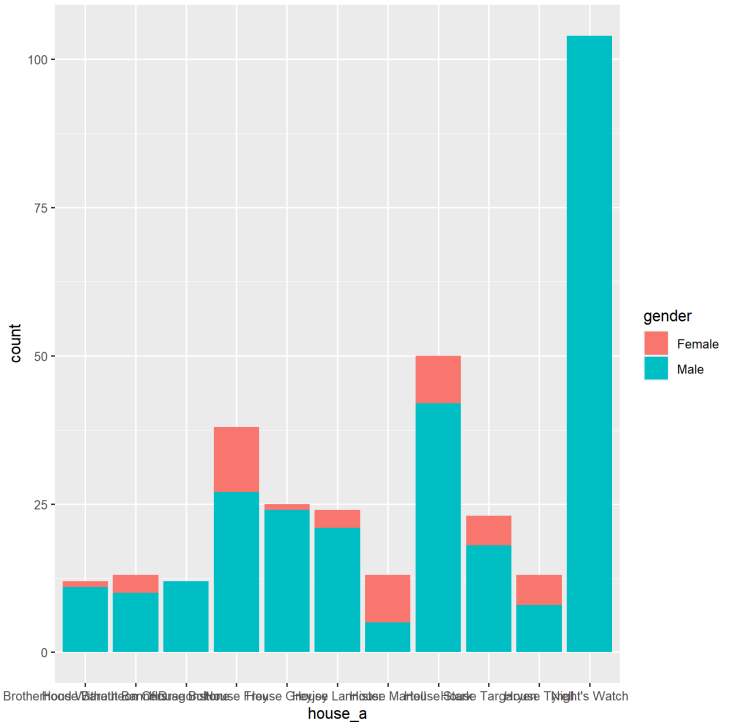
1. Let's start with a basic bar plot

```
got_houses_plot %>%
  ggplot(aes(house_a)) +
  geom_bar()
```



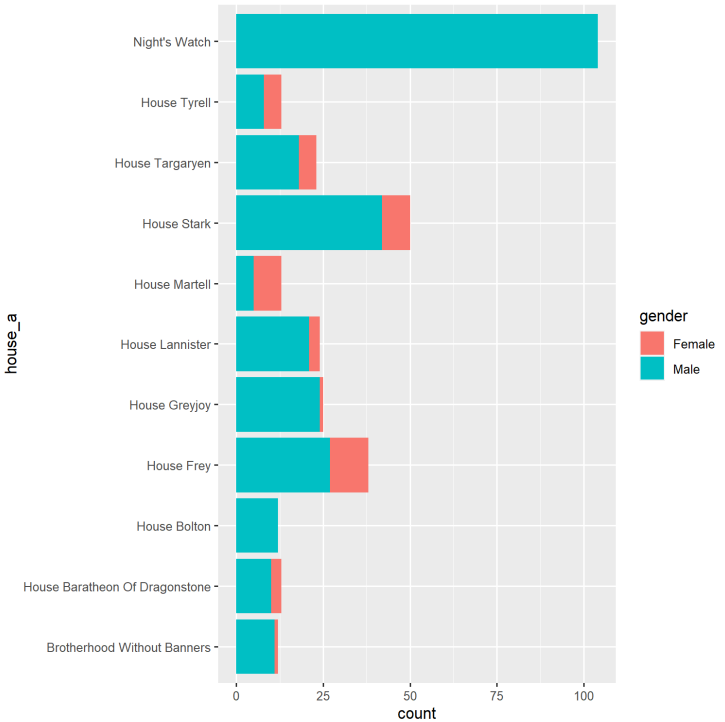
2. Now let's add add gender to aes()

```
got_houses_plot %>%
  ggplot(aes(house_a,
             fill = gender)) +
  geom_bar()
```



3. Flip the coords

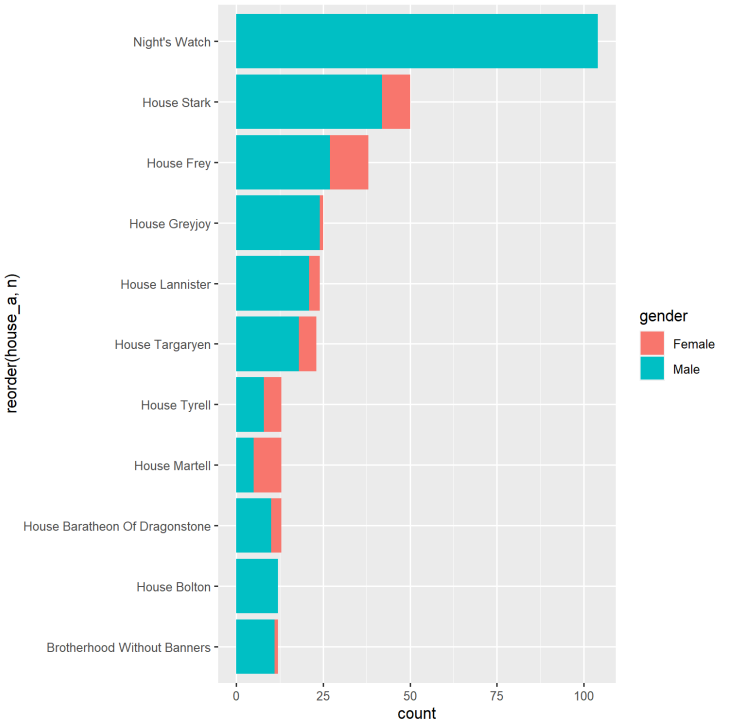
```
got_houses_plot %>%
  ggplot(aes(y = house_a,
             fill= gender)) +
  geom_bar()
```



Before `ggplot2 3.3.0` (last version) we needed to use `coord_flip()`

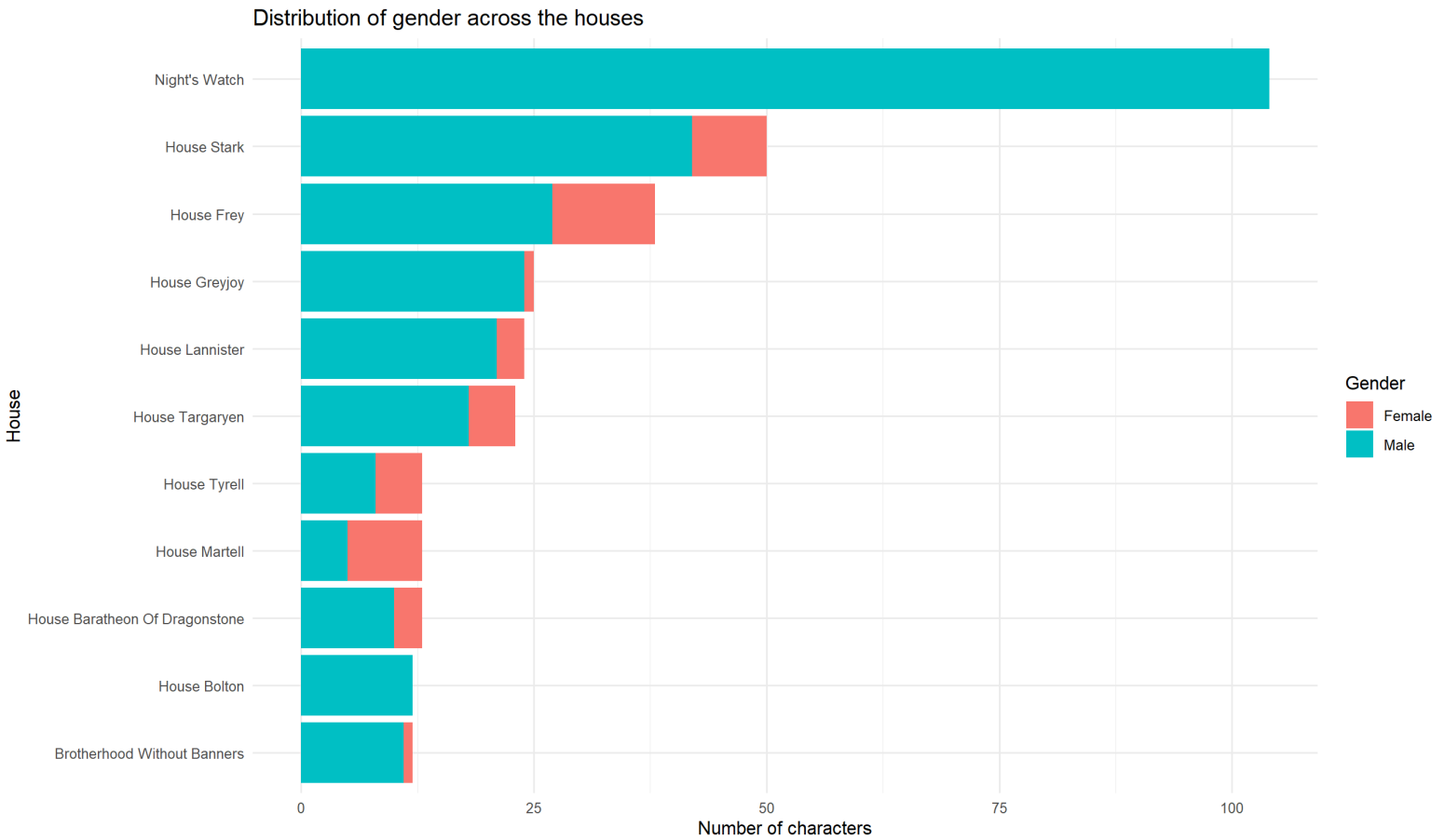
4. Sort by frequency

```
got_houses_plot %>%
  ggplot(aes(y = reorder(house_a,
                 fill= gender)) +
  geom_bar()
```

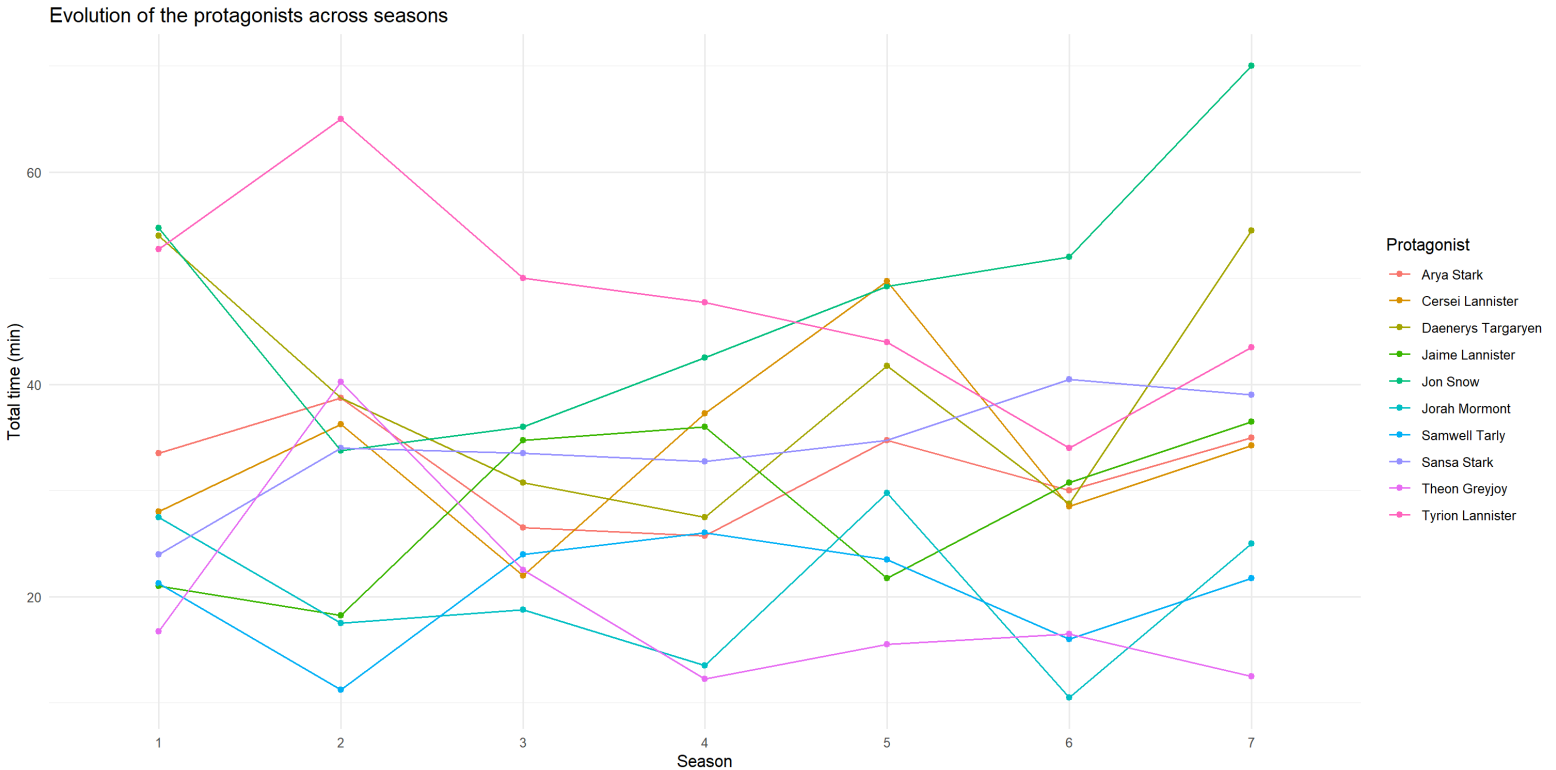


5. Details count 

```
got_houses_plot %>%
  ggplot(aes(y = reorder(house_a, n),
             fill= gender)) +
  geom_bar() +
  labs(title = "Distribution of gender across the houses",
       x = "Number of characters",
       y = "House",
       fill = "Gender") +
  theme_minimal()
```


Distribution of gender across the houses
Night's Watch
House Stark
House Frey
House Greyjoy
House Lannister
House Targaryen
House Tyrell
House Martell
House Baratheon Of Dragonstone
House Bolton
Brotherhood Without Banners
House
0
25
50
75
100
Number of characters
Gender
Female
Male

Exercise 3. How was the evolution of the protagonists across seasons?

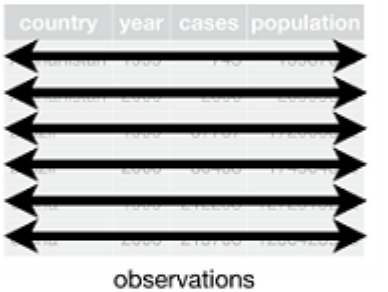


What variables do we need to plot this graph?

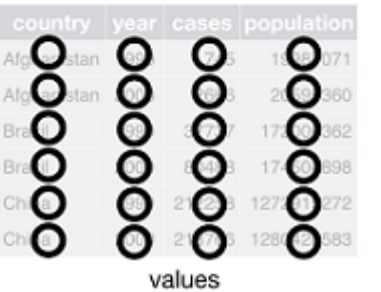
What is tidy data

- **Rule 1**: Each ***variable*** must have its own ***column***.

- **Rule 2**: Each ***observation*** must have its own ***row***.

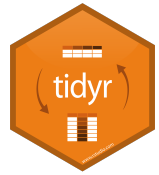


- **Rule 3**: Each ***value*** must have its own ***cell***.



https://r4ds.had.co.nz/

Go from wide to long with `pivot_longer`



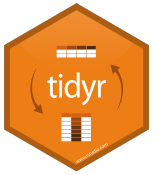
Wide

actor	season_1	season_2
Arya Stark	33.50	38.75
Jhon Snow	54.75	33.75
Sansa Stark	24.00	34.00

Long

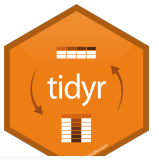
country	season	time
Arya Stark	1	33.50
Arya Stark	2	38.75
Jhon Snow	1	54.75
Jhon Snow	2	33.75
Sansa Stark	1	24.00
Sansa Stark	2	34.00

Go from wide to long with `pivot_longer`



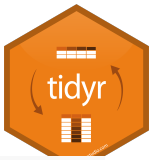

```
got_complete %>%
  pivot_longer(
    cols = season_1:season_7,
    names_to = "season",
    values_to = "time",
    names_prefix = "season_")

head(got_long)
```

Go from wide to long with `pivot_longer`



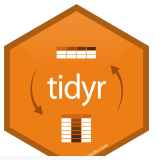

```
got_complete %>%
  pivot_longer(
    cols = season_1:season_7,
    names_to = "season",
    values_to = "time",
    names_prefix = "season_")
```

Go from wide to long with `pivot_longer`



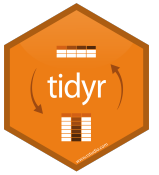

```
got_complete %>%
  pivot_longer(
    cols = season_1:season_7,
    names_to = "season",
    values_to = "time",
    names_prefix = "season_")
```

Go from wide to long with `pivot_longer`




```
got_complete %>%
  pivot_longer(
    cols = season_1:season_7,
    names_to = "season",
    values_to = "time",
    names_prefix = "season_")
```

Go from wide to long with `pivot_longer`




```
got_long <- got_complete %>%
  pivot_longer(
    cols = season_1:season_7,
    names_to = "season",
    values_to = "time",
    names_prefix = "season_")
```

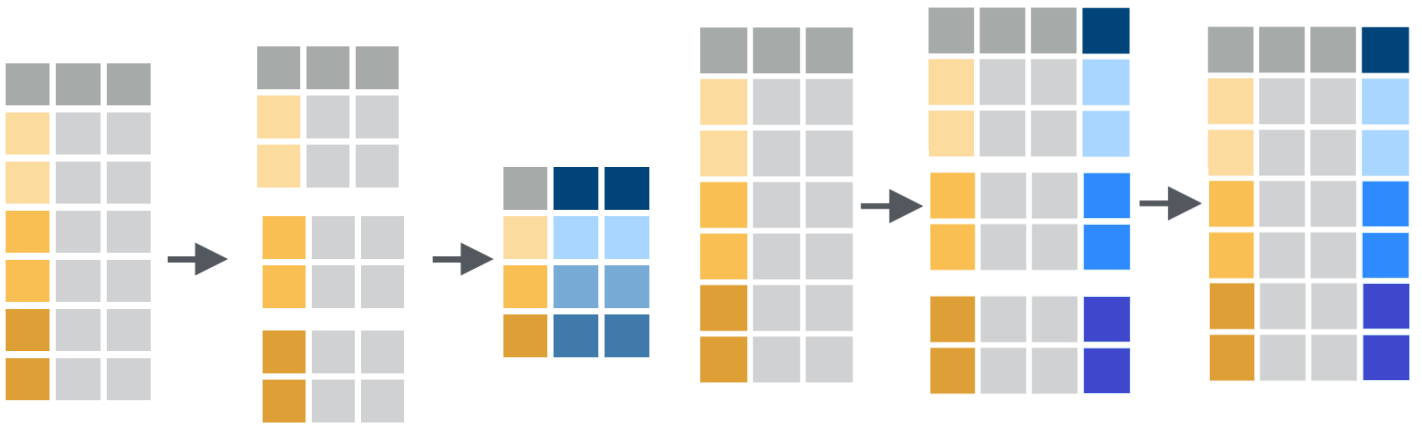
Create a total sum of time by character

group_by() + summarise()

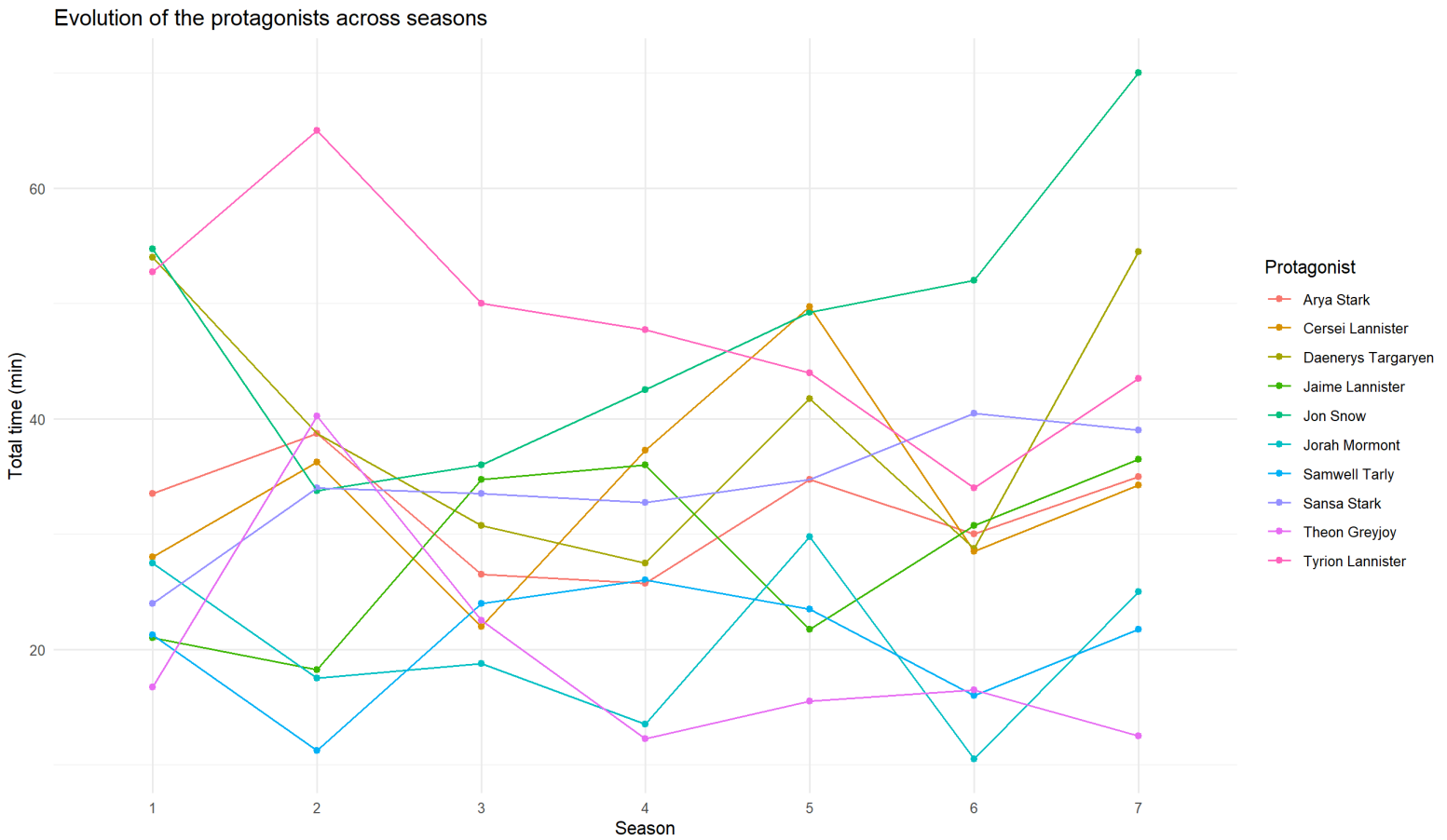
```
got_long %>%
  group_by(actor) %>%
  summarise (total = sum(time)) %
  ungroup ()
```

group_by() + mutate()

```
got_long %>%
  group_by(actor) %>%
  mutate(total = sum(time)) %>%
  ungroup()
```

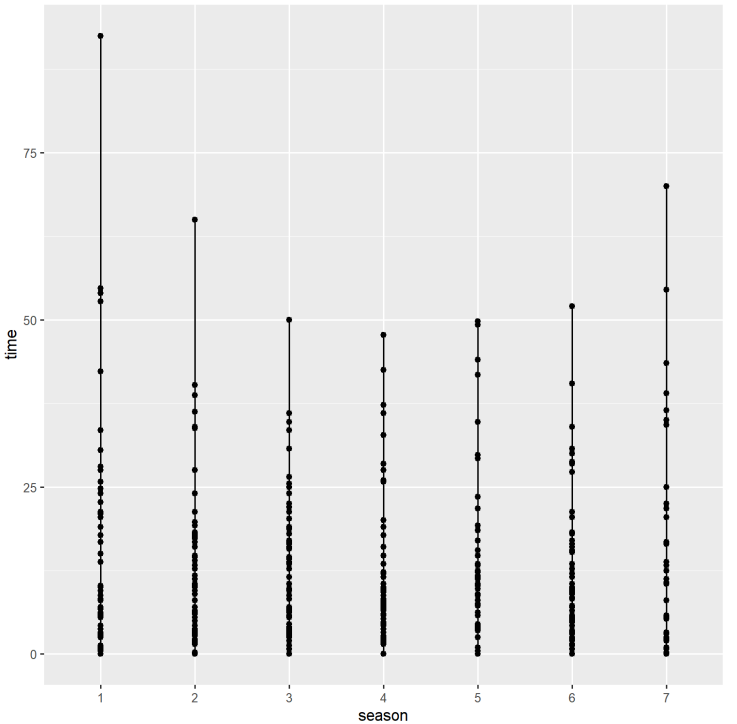


Back to the graph



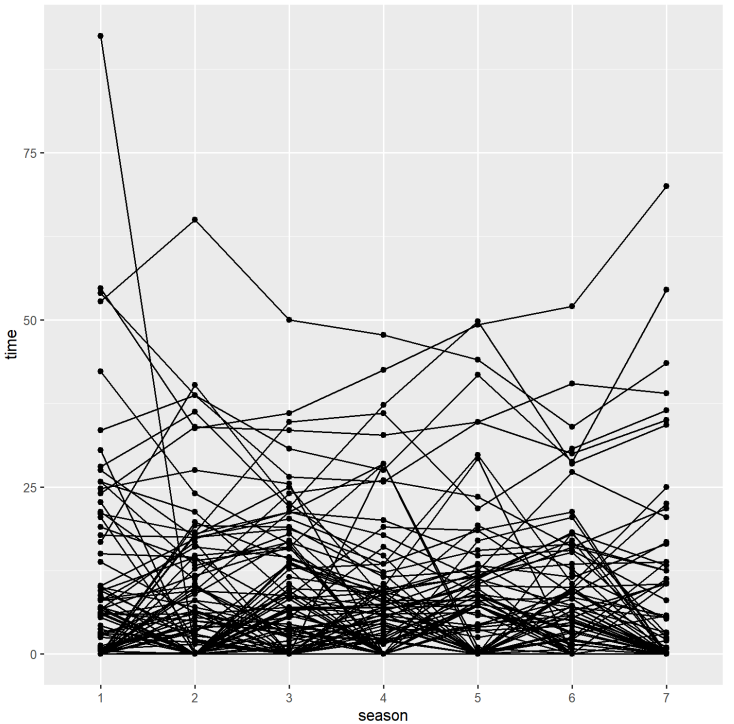
1. Add the aesthetics and geoms

```
got_long %>%
  ggplot(aes(season, time))+
  geom_point() +
  geom_line()
```



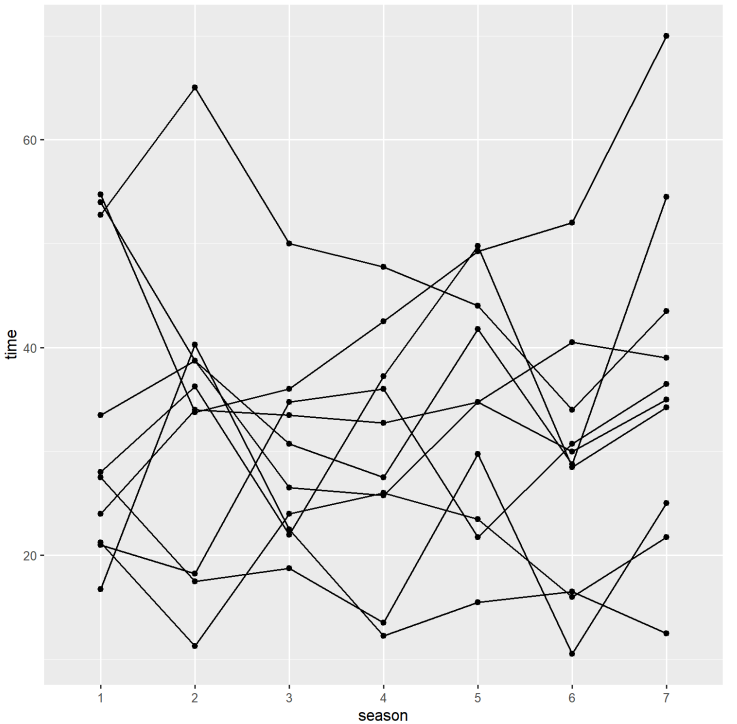
2. Add actors to aes()

```
got_long %>%
  ggplot(aes(season, time,
             group = actor)) +
  geom_point() +
  geom_line()
```



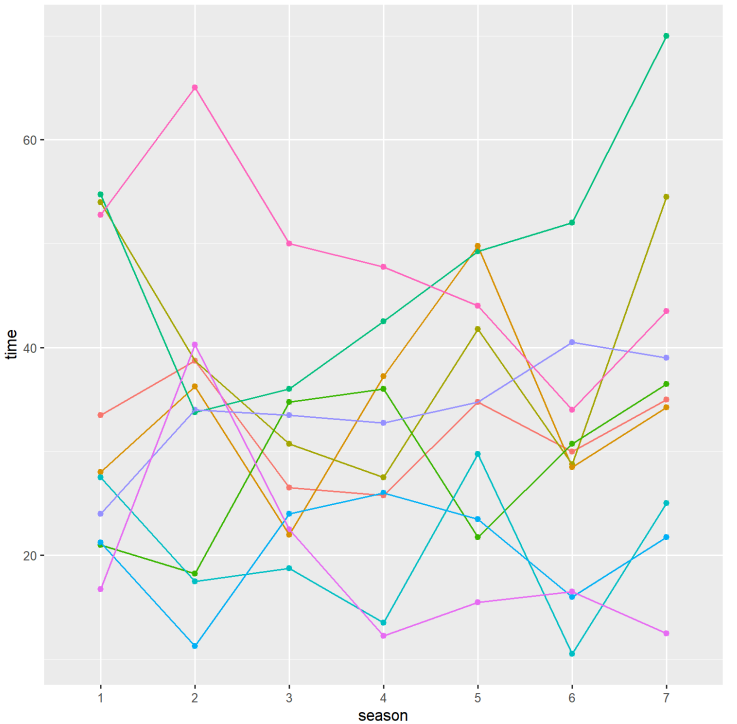
3. Filter the top ten (>130min)

```
got_long %>%
  filter(total >130) %>%
  ggplot(aes(season, time,
             group = actor))+
  geom_point() +
  geom_line()
```



4. Add a color for each character

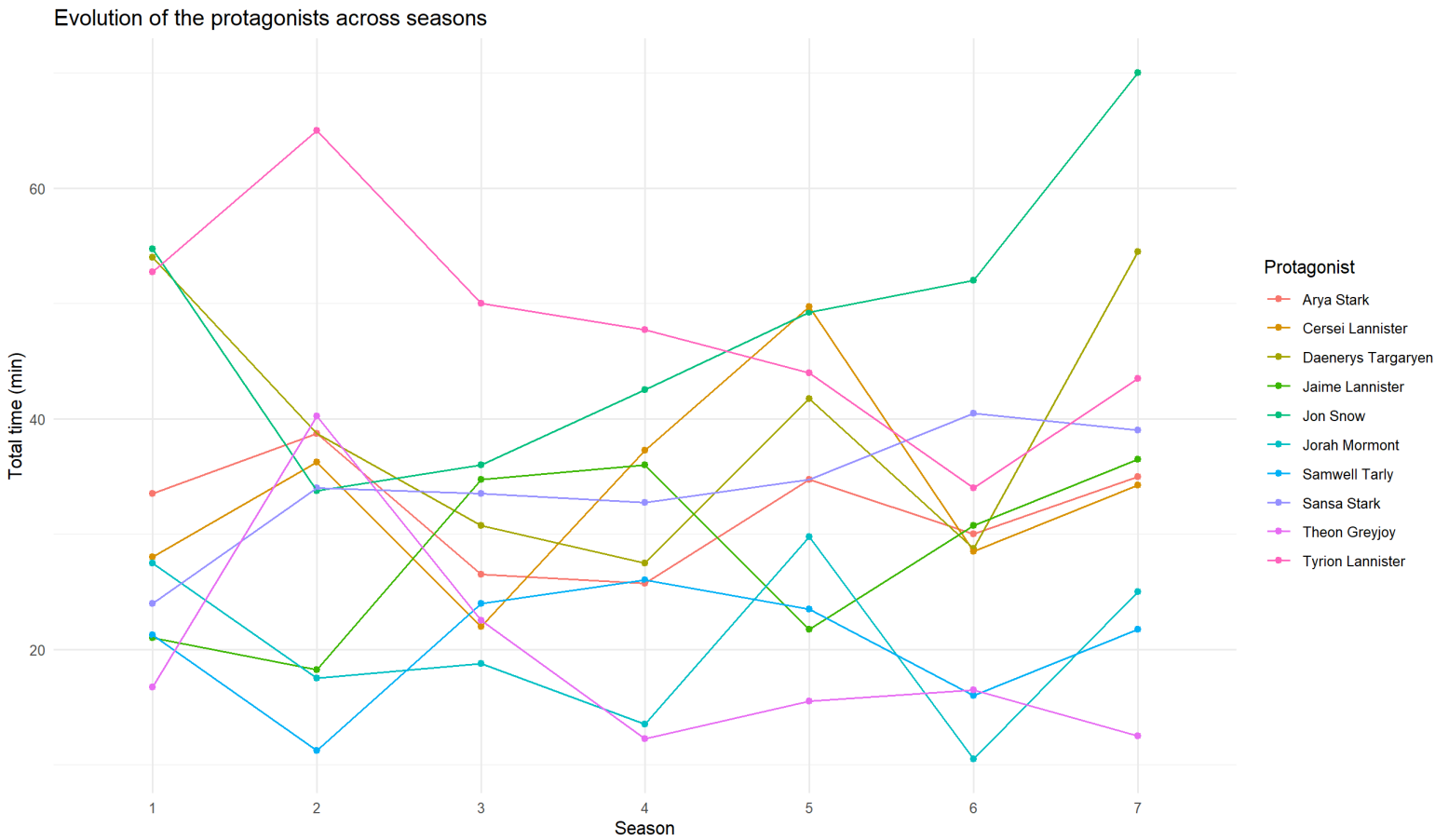
```
got_long %>%
  filter(total >130) %>%
  ggplot(aes(season, time,
             group = actor,
             color = actor)) +
  geom_point() +
  geom_line()
```



5. Details

```
got_long %>%
  filter(total >130) %>%
  ggplot(aes(season, time, group = actor, color = actor)) +
  geom_point() +
  geom_line() +
  theme_minimal() +
  labs(title = "Evolution of the protagonists across seasons",
       x = "Season",
       y = "Total time (min)",
       color = "Protagonist") +
  theme_minimal()
```

Final graph


Evolution of the protagonists across seasons
Total time (min)
60
40
20
1
2
3
4
5
6
7
Season
Protagonist
Arya Stark
Cersei Lannister
Daenerys Targaryen
Jaime Lannister
Jon Snow
Jorah Mormont
Samwell Tarly
Sansa Stark
Theon Greyjoy
Tyrion Lannister

We did it!

Useful resources

- Learn tidyverse with the R4DS book (free online): https://r4ds.had.co.nz/
- Practice your skills with the primers: https://rstudio.cloud/learn/primers
- Join the R4DS and Tidy-tuesday community in twitter

Thank you!!!

Keep in touch!

l.rojassaunero@erasmusmc.nl

@palolili23

@palolili23