Grant Humpries &
Anton Van de Putte

DATA VISULATISATION

museum
Operational Directorate Natural Environment
OD Nature | OD Natuur | DO Nature

BIODIVERSITY.AQ

HiDef
AERIAL SURVEYING

What is data Visualisation

THE VISUAL REPRESENTATION AND PRESENTATION OF DATA TO FACILITATE COMPREHENSION

Data Visualisation: A
Handbook for Data Driven
Design' (2nd edition, 2019)
Andy Kirk

3 PHASES OF UNDERSTANDING

Perceiving

What data is shown?

How is the data represented?

Interpreting

What features are interesting?

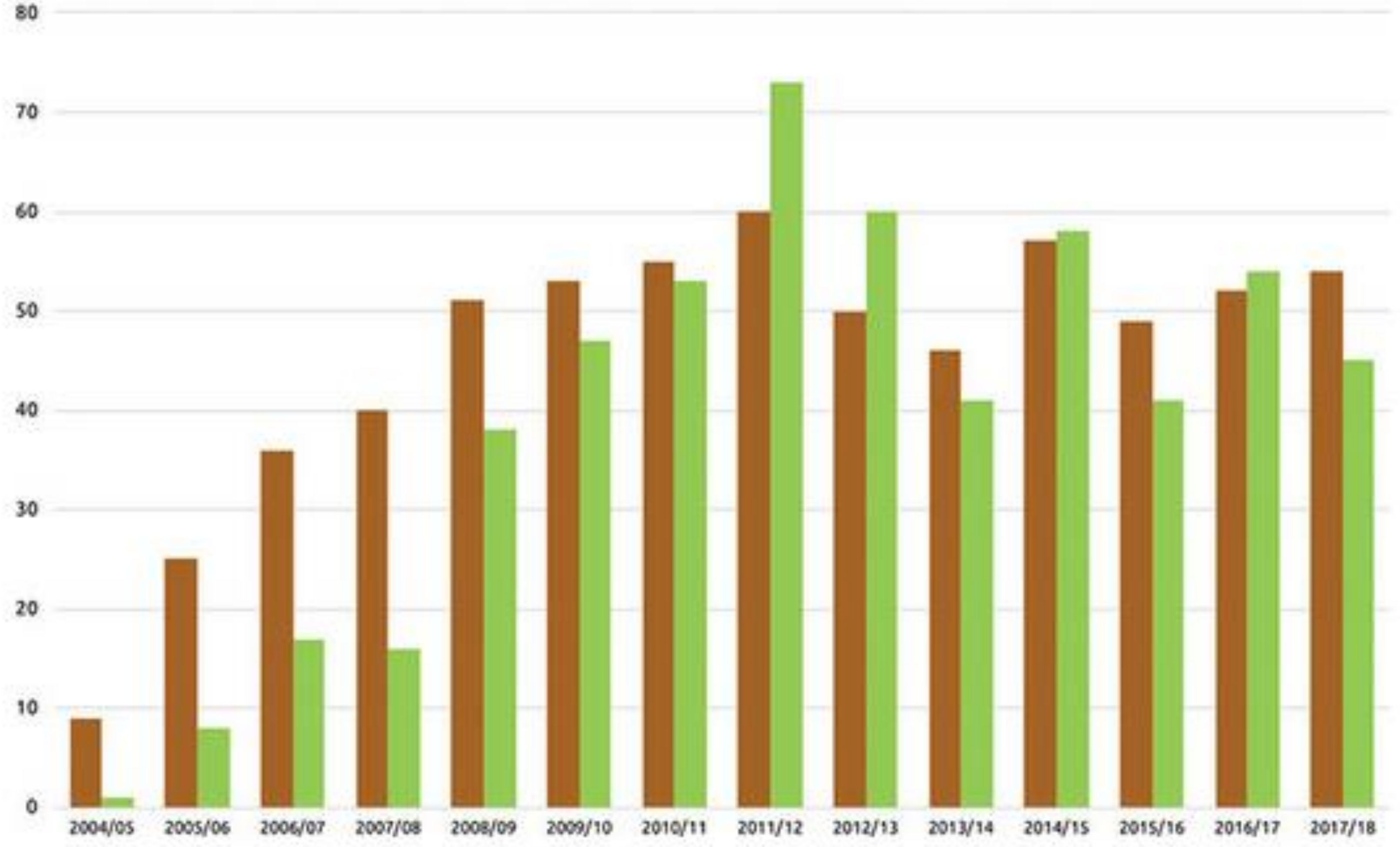
What features are important?

Comprehending

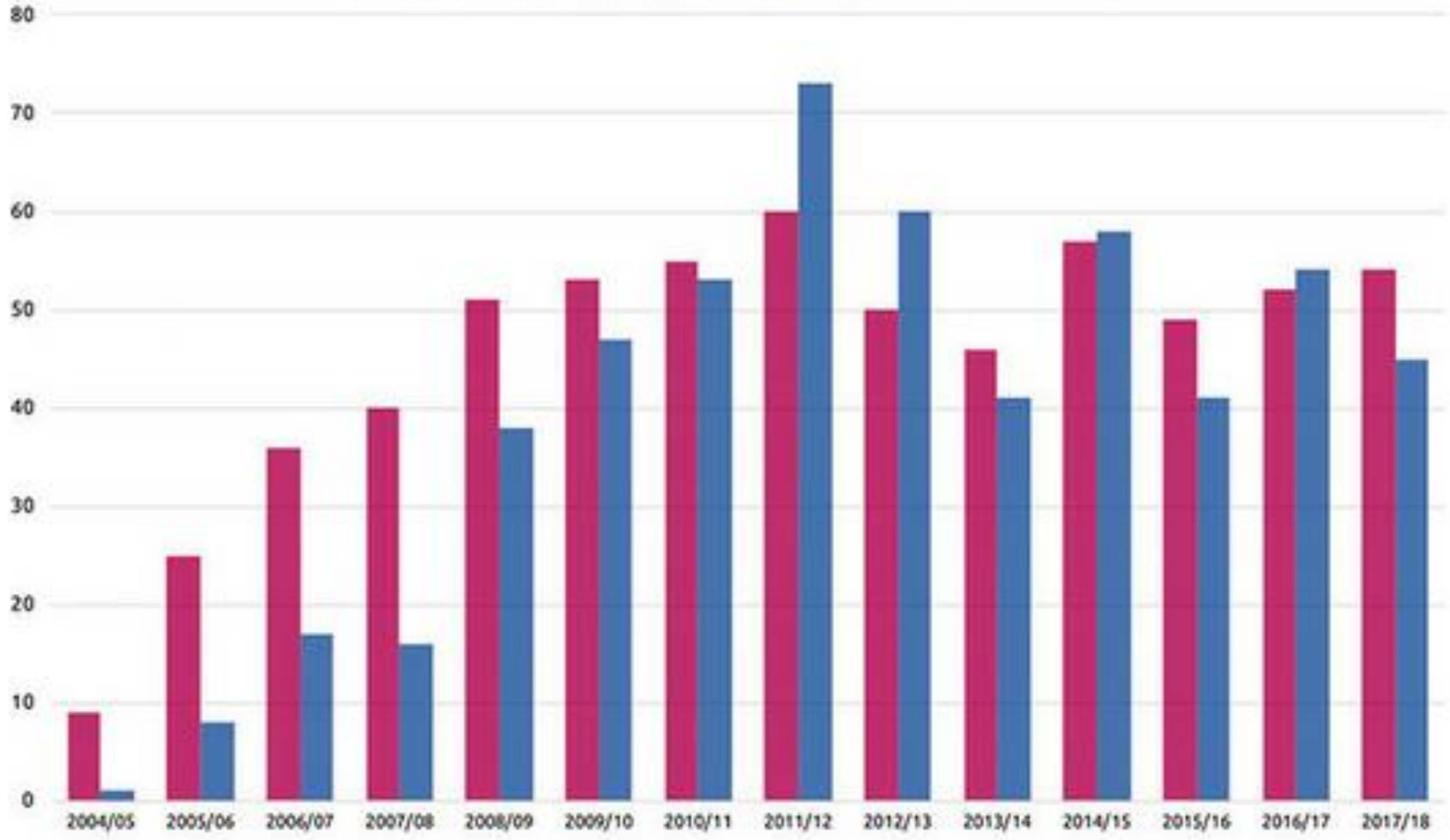
What have I learned?

What do I feel?

Total sightings of Winglets and Spungles



Lionel Messi: Games and Goals for FC Barcelona

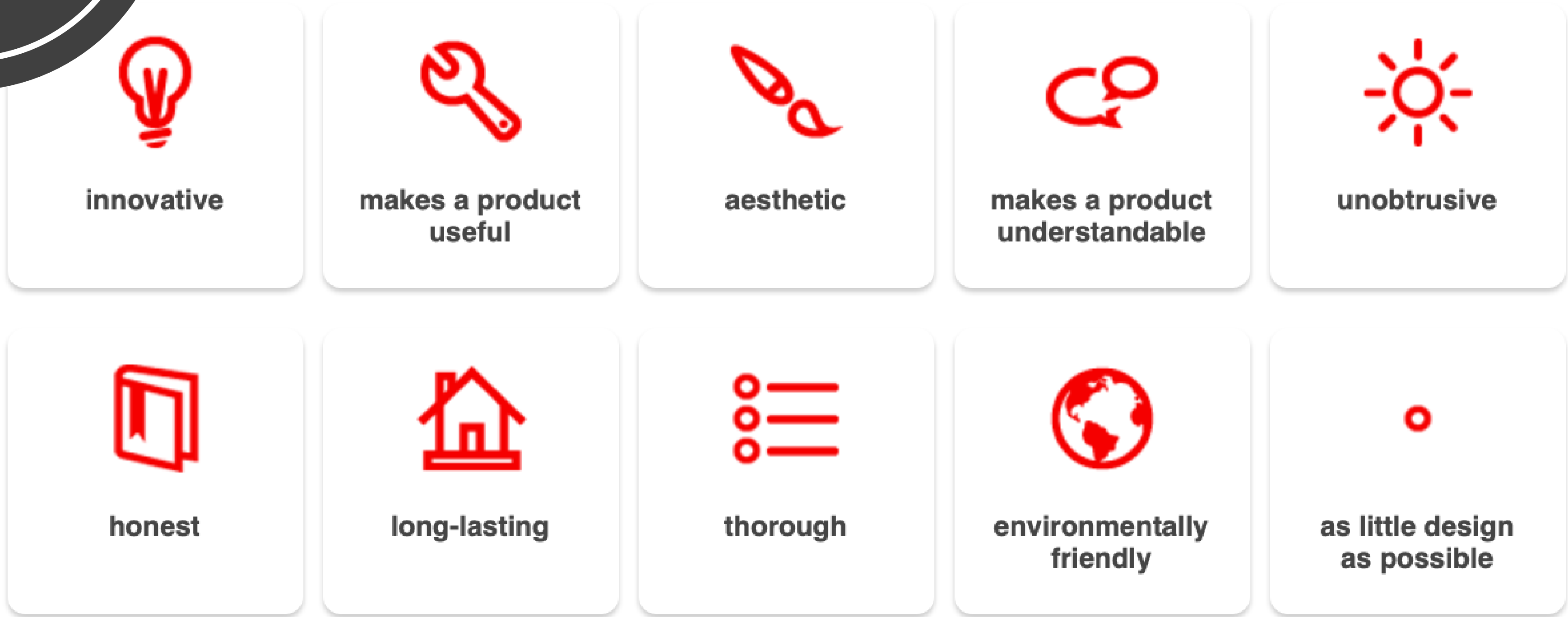


Source: Data from transfermarkt.com

GOOD DESIGN

- Innovative
- Makes a product Useful
- Aesthetic
- Makes a product understandable
- Unobtrusive
- Honest
- Long-lasting
- Thorough
- Environmental friendly
- As little design as possible

GOOD DESIGN



innovative

makes a product useful

aesthetic

makes a product understandable

unobtrusive

honest

long-lasting

thorough

environmentally friendly

as little design as possible

https://startupsthisishowdesignworks.com

GOOD VISUAL DESIGN IS

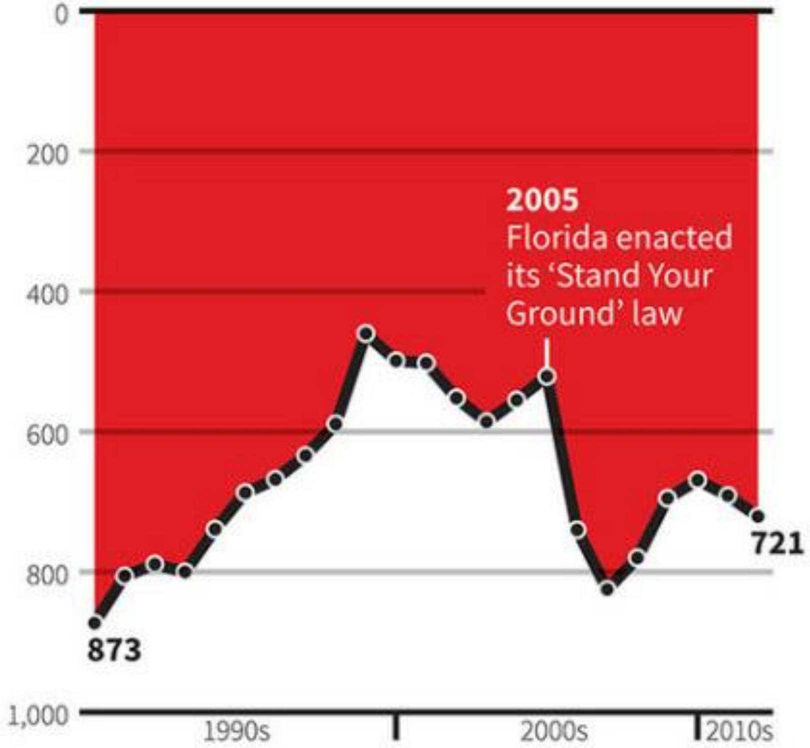
Trustworthy

Accessible

Elegant

Gun deaths in Florida

Number of murders committed using firearms



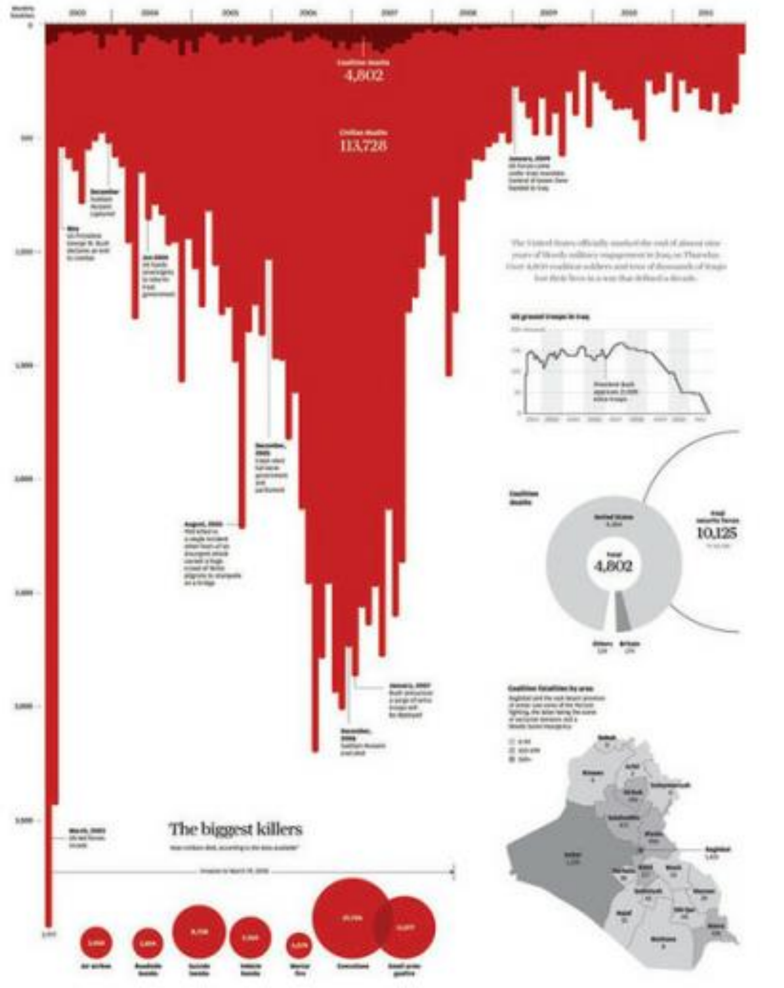
2005
Florida enacted
its 'Stand Your
Ground' law

Source: Florida Department of Law Enforcement

16/02/2014

REUTERS

Iraq's bloody toll



The United States officially marked the end of almost nine years of bloody military engagement in Iraq on Thursday. Over 4,800 coalition soldiers and tens of thousands of Iraqis lost their lives in a war that defined a decade.

VISUALISATION IN ACTION: GGPLOT

Feel free to follow and code along! If you already know ggplot, play around and be creative! We're going to use the starwars dataset again

ggplot2 package is included in the tidyverse!

```
library(tidyverse)
data(starwars)
```

GGPLOT2… your best friend and worst enemy, wrapped into a neat little r package…

- Powerful and effective once you figure out how to manipulate it
- Demands your time and energy
- Breaks your confidence when you do something wrong
- Never happy for your achievements
- Probably makes fun of you behind your back

How does GGPLOT work??

GGPLOT 2 uses similar tidyverse logic to what we’ve seen before!

However, ggplot was a late addition to the tidyverse, and so the logic connector isn’t a pipe (‘%>%’), it’s a plus symbol (‘+’).

```
ggplot()  # Creates the canvas

ggplot(data,aes(x=variable,y=variable)) #
Defines the data and aesthetics

ggplot(data,aes(x=variable,y=variable)) +
geom_point() #Creates the geometric
object to be plotted
```

KEY terms to remember!

Data: well.. This is self explanatory…

Canvas: The background that your points will be laid on

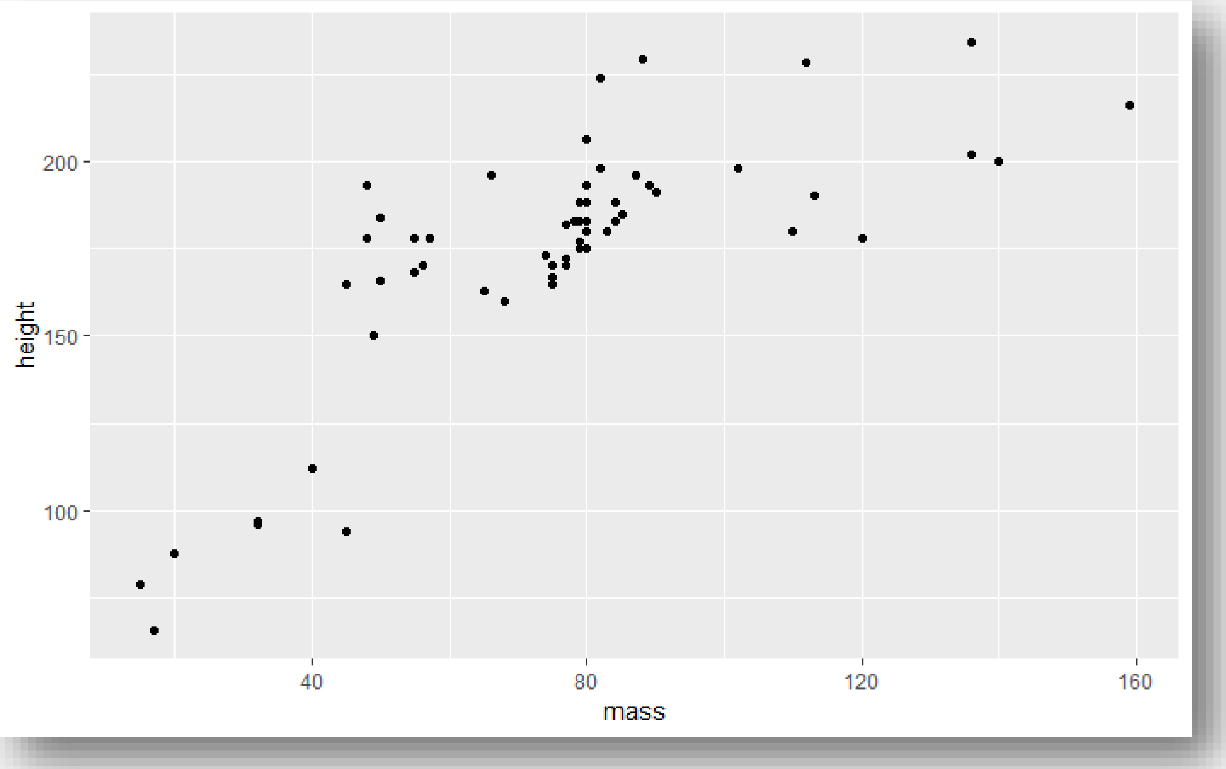
Aesthetics: WHAT you want to graph (x,y or z), and any ‘grouping’ parameters.

Geometric object: HOW you want your aesthetics to be plotted

Theme: The visual layout of the canvas and other aspects of the figure (fonts, etc…)

THE BASIC SCATTER PLOT

```
starwars <- starwars %>% filter(mass < 1000) # remove Jabba...

ggplot(starwars,aes(x=mass,y=height)) +
  geom_point()
```



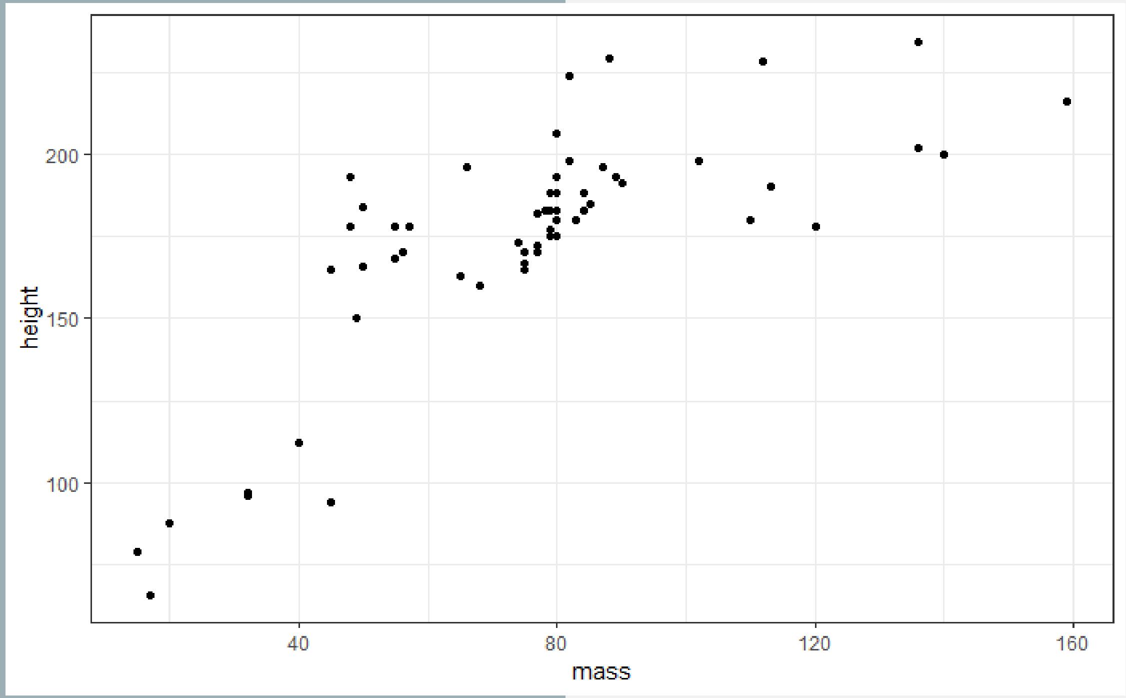
STYLING THE CANVAS

https://ggplot2.tidyverse.org/reference/ggtheme.html

ggplot themes are super handy, and can be easily applied! OR you can use a custom theme!

```
ggplot(starwars,aes(x=mass,y=height)) +
  geom_point() +
  theme_bw()
```

MY FAVOURITES:



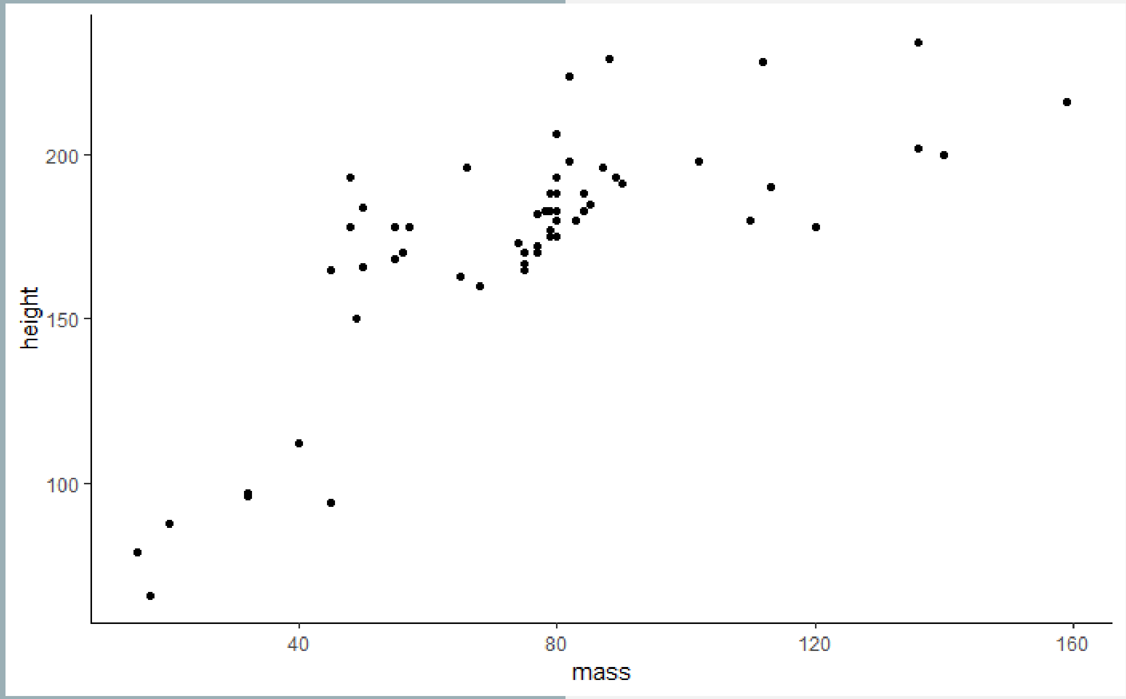
STYLING THE CANVAS

https://ggplot2.tidyverse.org/reference/ggtheme.html

ggplot themes are super handy, and can be easily applied! OR you can use a custom theme!

```
ggplot(starwars,aes(x=mass,y=height)) +
  geom_point() +
  theme_classic()
```

MY FAVOURITES:



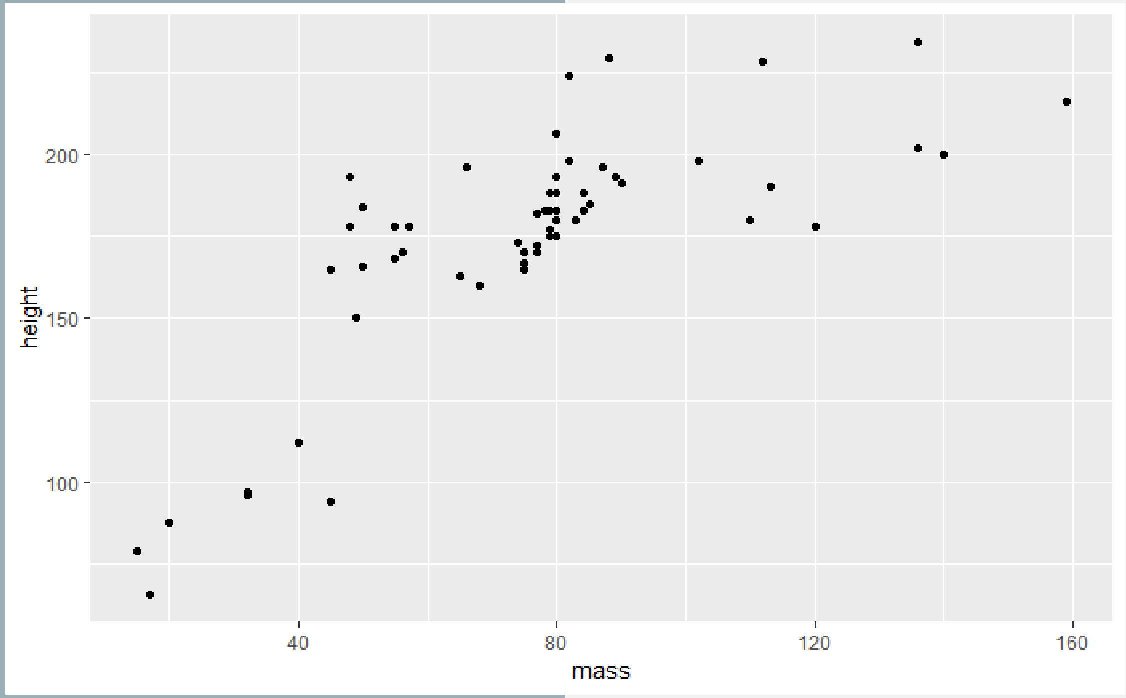
STYLING THE CANVAS

https://ggplot2.tidyverse.org/reference/ggtheme.html

ggplot themes are super handy, and can be easily applied! OR you can use a custom theme!

```
ggplot(starwars,aes(x=mass,y=height)) +
  geom_point() +
  theme_grey()
```

MY FAVOURITES:



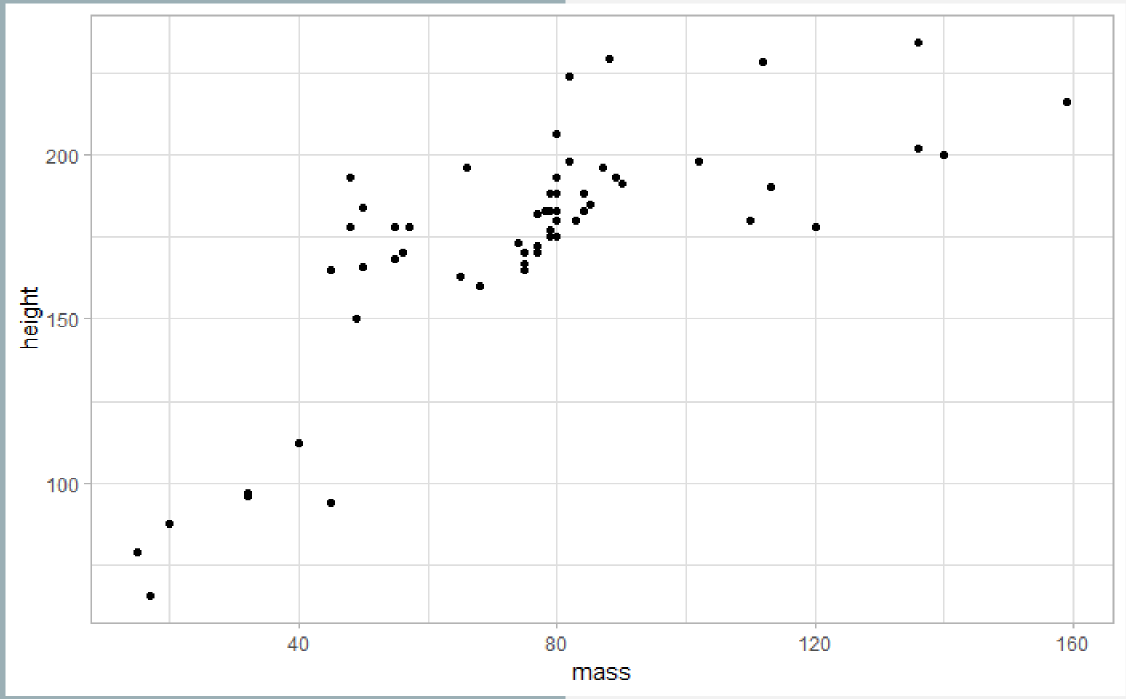
STYLING THE CANVAS

https://ggplot2.tidyverse.org/reference/ggtheme.html

```
ggplot(starwars,aes(x=mass,y=height)) +
  geom_point() +
  theme_light()
```

ggplot themes are super handy, and can be easily applied! OR you can use a custom theme!

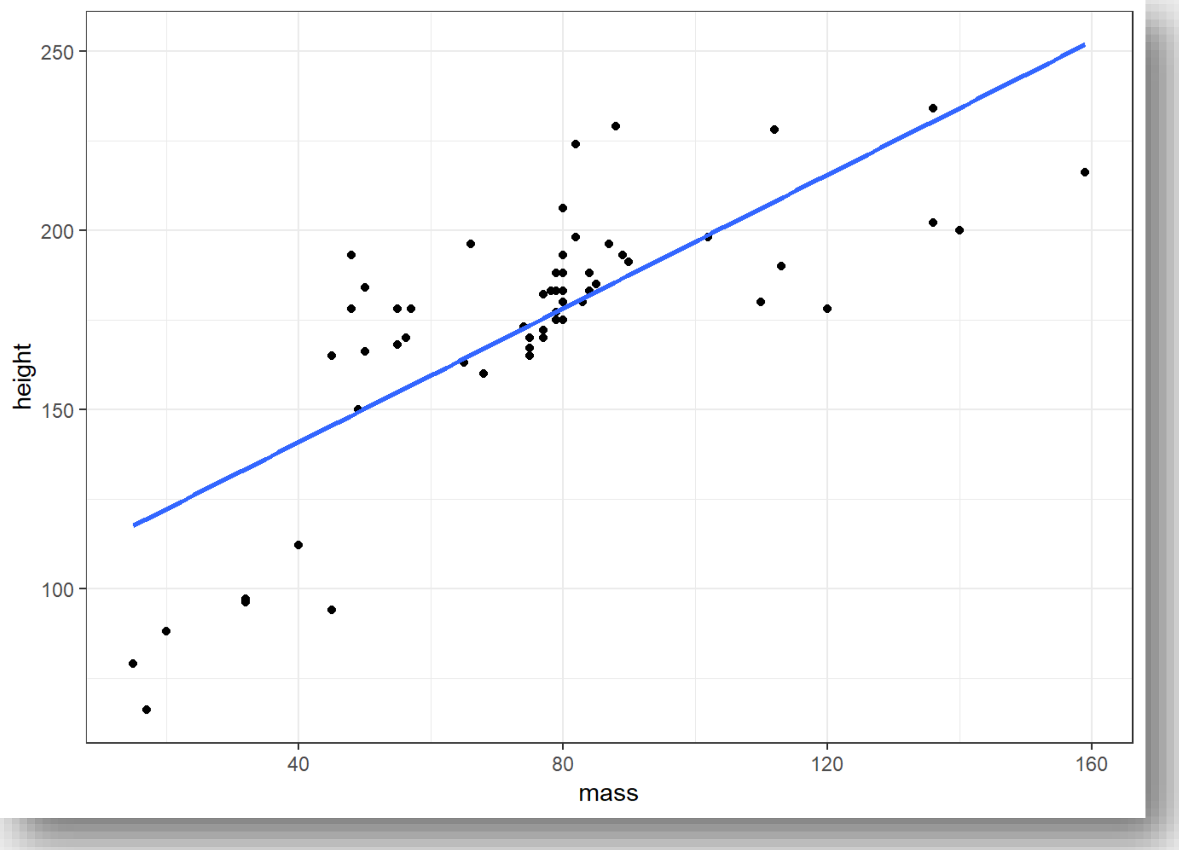
MY FAVOURITES:



ADDING OTHER ELEMENTS

A common task might be to take your points and run a trendline through them.
ggplot's logic allows you to simply stack **LAYERS** of geometric objects to your plot.

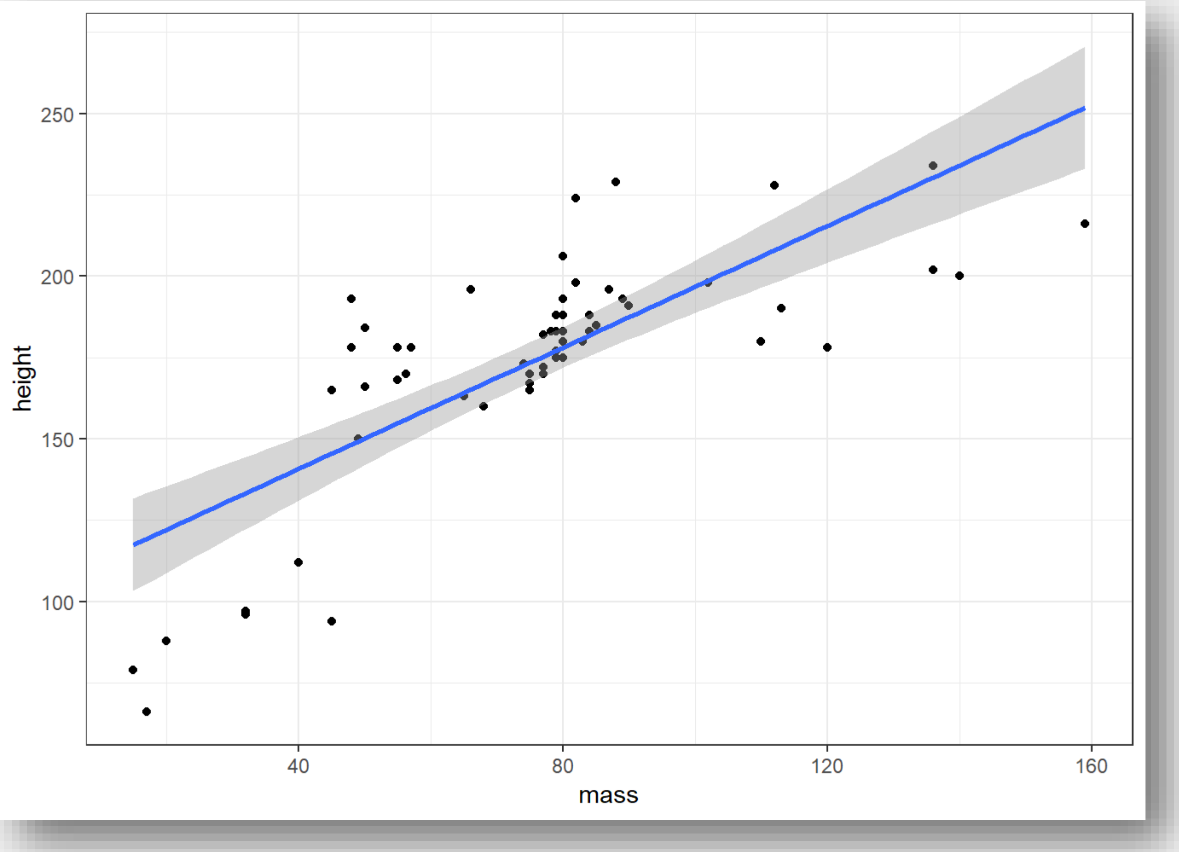
```
ggplot(starwars,aes(x=mass,y=height)) +
  geom_point() +
  geom_smooth(method=lm,se=FALSE) +
  theme_bw()
```



ADDING OTHER ELEMENTS

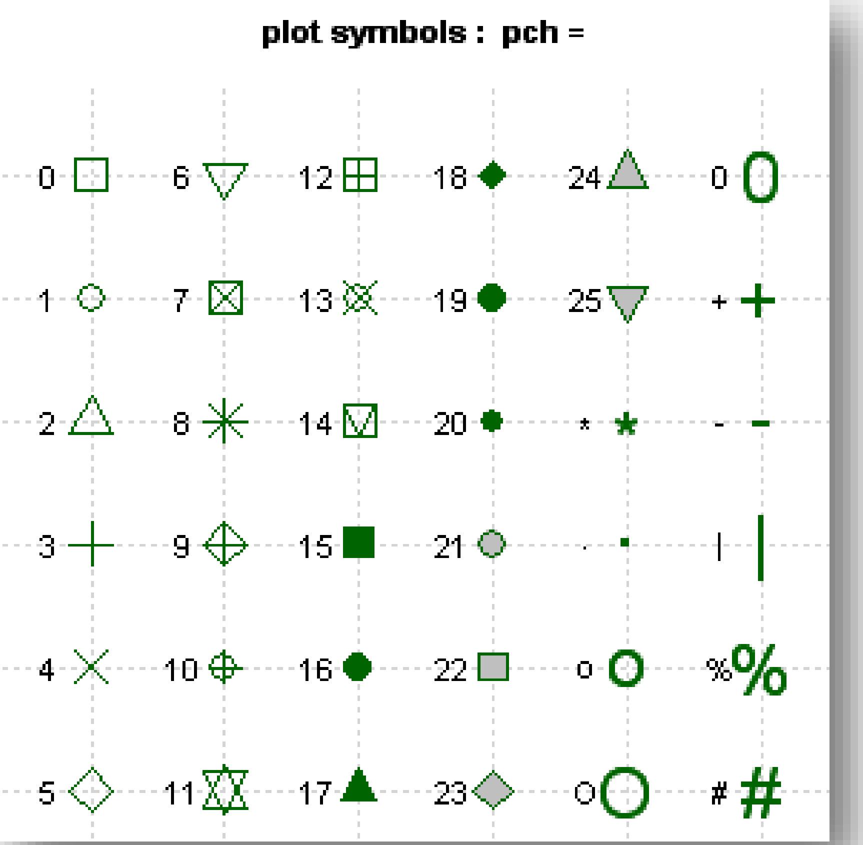
You can add uncertainty to your LM estimate by simply setting se=TRUE.

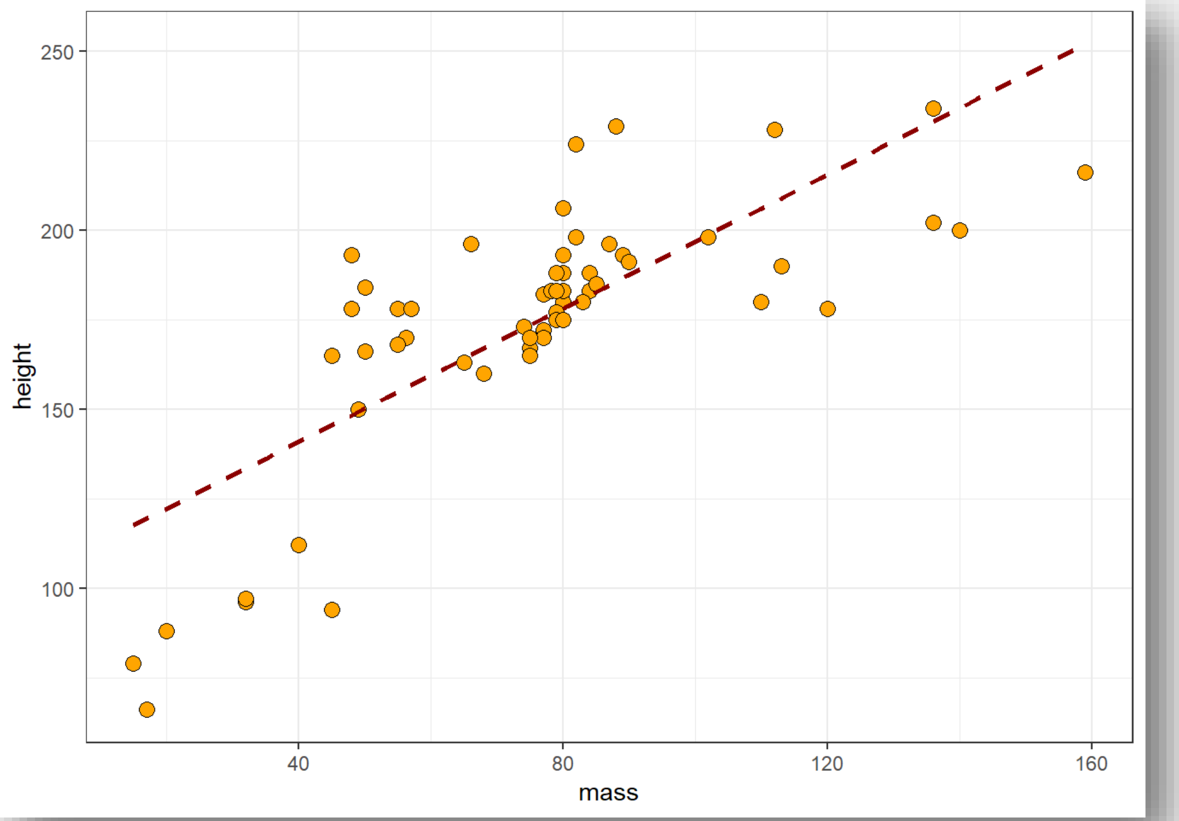
```
ggplot(starwars,aes(x=mass,y=height)) +
  geom_point() +
  geom_smooth(method=lm,se=TRUE) +
  theme_bw()
```



STYLING YOUR FIGURES

```
ggplot(starwars,aes(x=mass,y=height)) +
  geom_point(pch=21,fill='orange',color='black',size=3) +
  geom_smooth(linetype='dashed',color='darkred',method=lm,se=FALSE) +
  theme_bw()
```



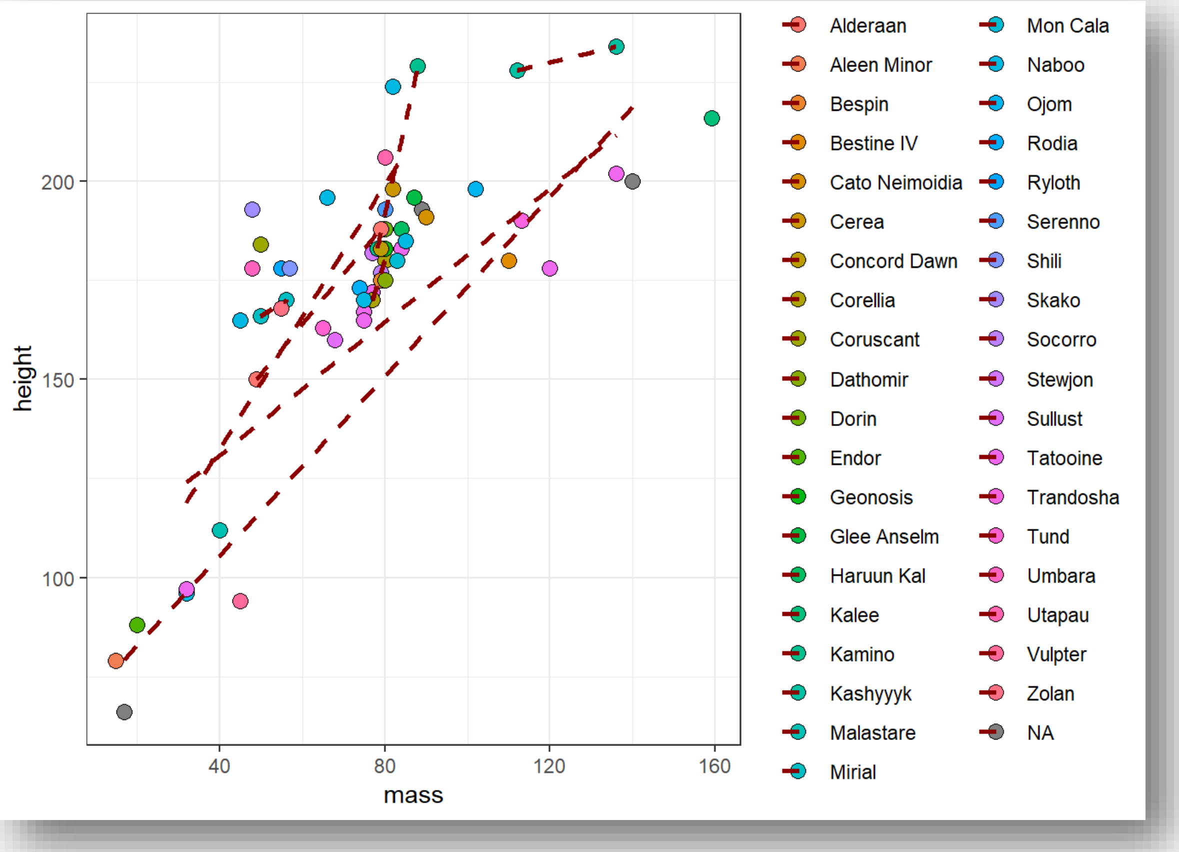


STYLING YOUR FIGURES

ggplot allows you to easily group your data and colour it based on those groups!

Remember to read through the ggplot code logically.

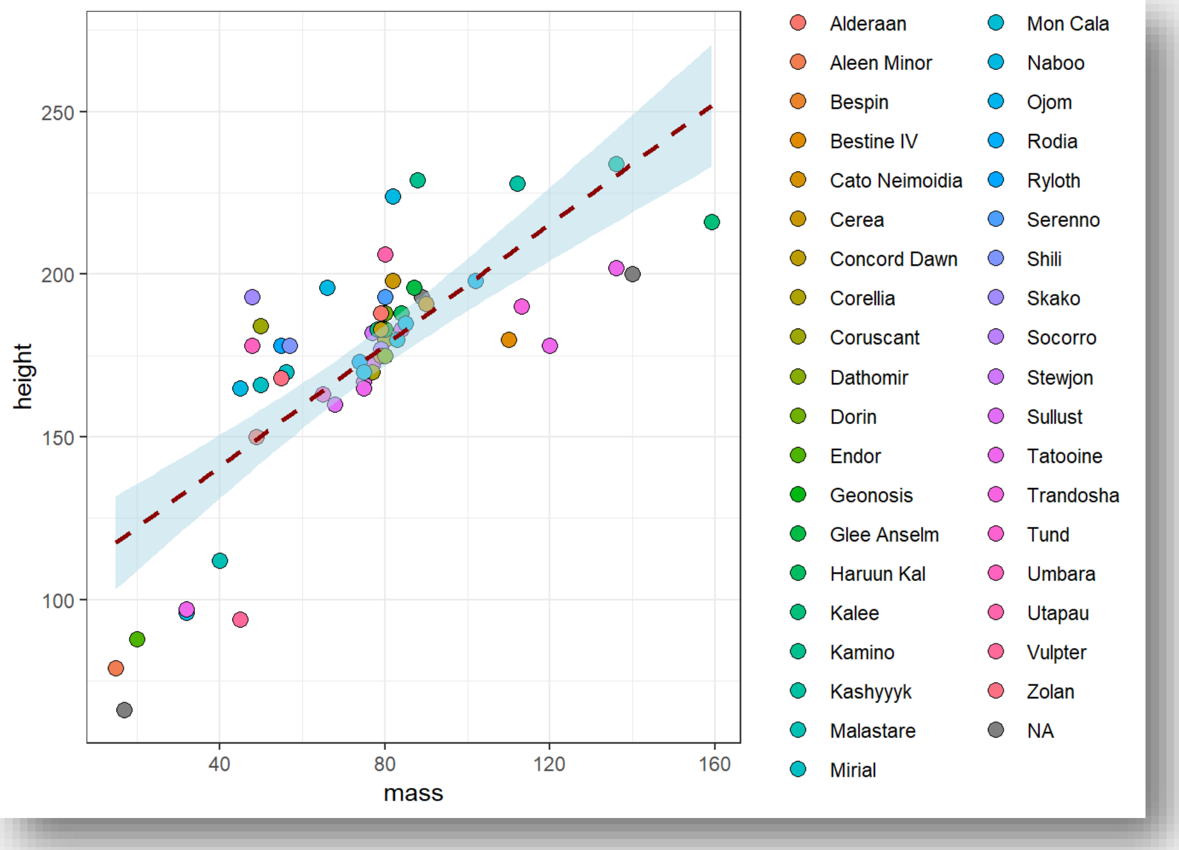
```
# A tibble: 58 x 3
   mass height homeworld
  <dbl>  <int> <chr>
1    77    172 Tatooine
2    75    167 Tatooine
3    32     96 Naboo
4   136    202 Tatooine
5    49    150 Alderaan
6   120    178 Tatooine
7    75    165 Tatooine
```



```
ggplot(starwars,aes(x=mass,y=height,fill=homeworld)) +
  geom_point(pch=21,color='black',size=3) +
  geom_smooth(linetype='dashed',color='darkred',method=lm,se=FALSE) +
  theme_bw()
```

STYLING YOUR FIGURES

But look at all those lines?!? What if you want to show just the trend-line for the whole dataset. All you have to do is move your grouping variable to the geometric object you want to group

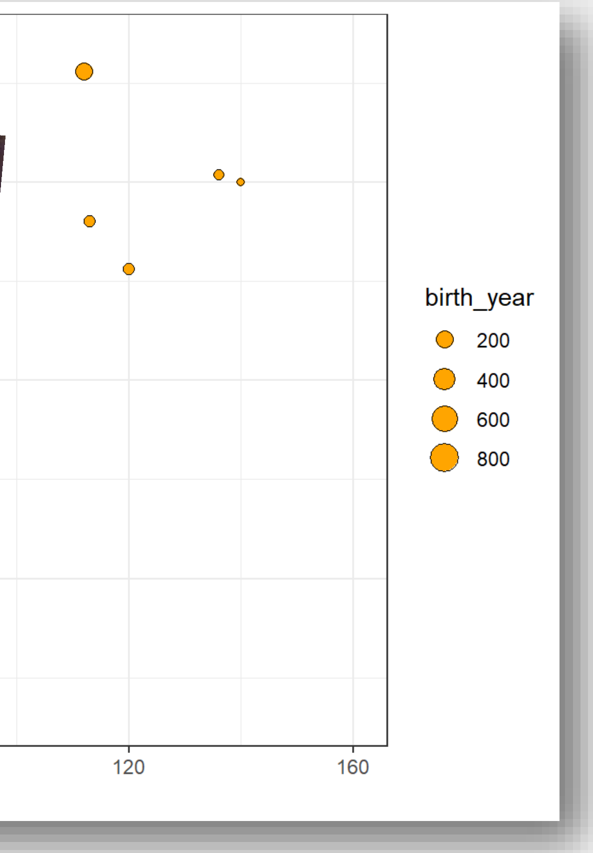


```
ggplot(starwars,aes(x=mass,y=height)) +
  geom_point(aes(fill=homeworld),pch=21,color='black',size=3) +
  geom_smooth(linetype='dashed',color='darkred',method=lm,se=TRUE,fill='lightblue',alpha=0.5) +
  theme_bw()
```

STYLING YOUR FIGURES

You can group by a continuous variable as well to style points by size.

```
# A tibble: 58 x 3
   mass height birth_year
  <dbl>  <int>      <dbl>
1    77    172         19
2    75    167        112
3    32     96         33
4   136    202       41.9
5    49    150         19
6   120    178         52
7    75    165         47
```

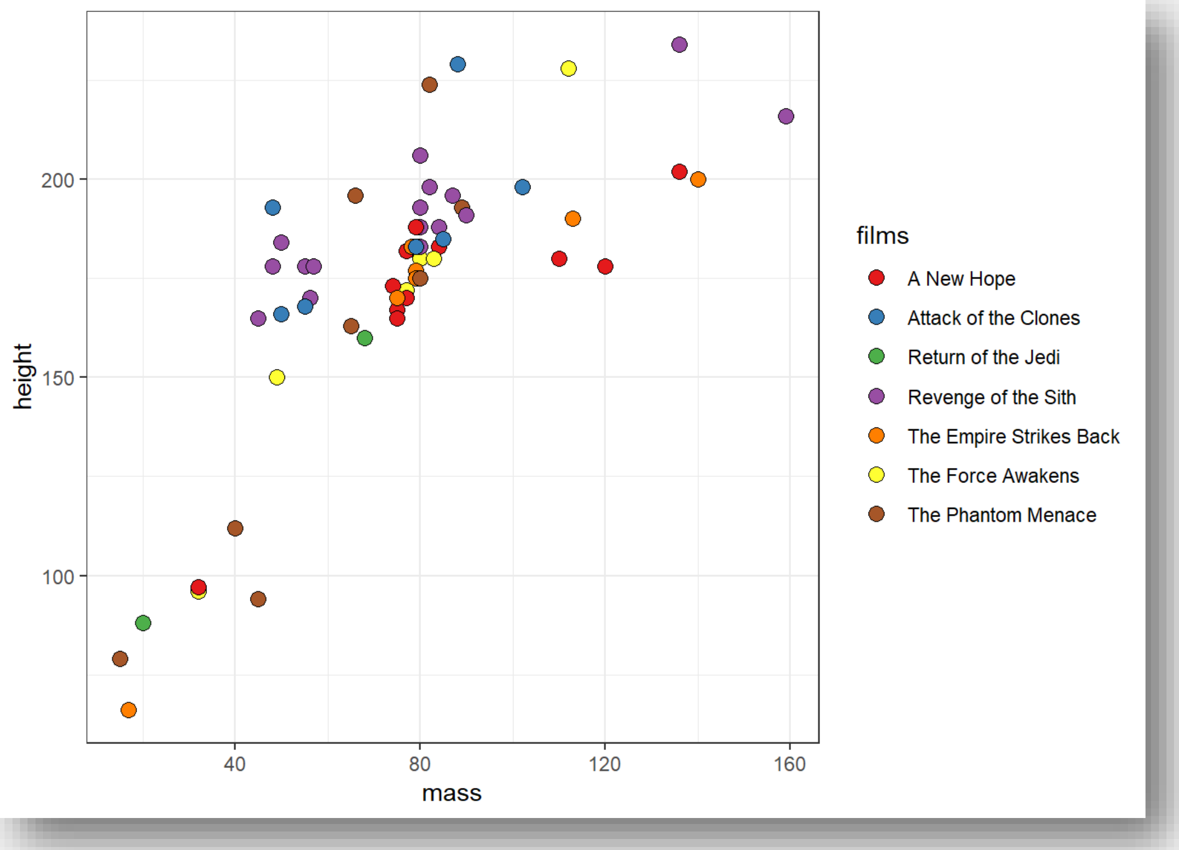


```
ggplot(starwars,a
  geom_point(pch=21,color='black',
theme_bw()
```

STYLING YOUR FIGURES

Let’s try a different grouping… say, by film instead! But remember, film is a nested column! So we have to use some tidyverse logic here.

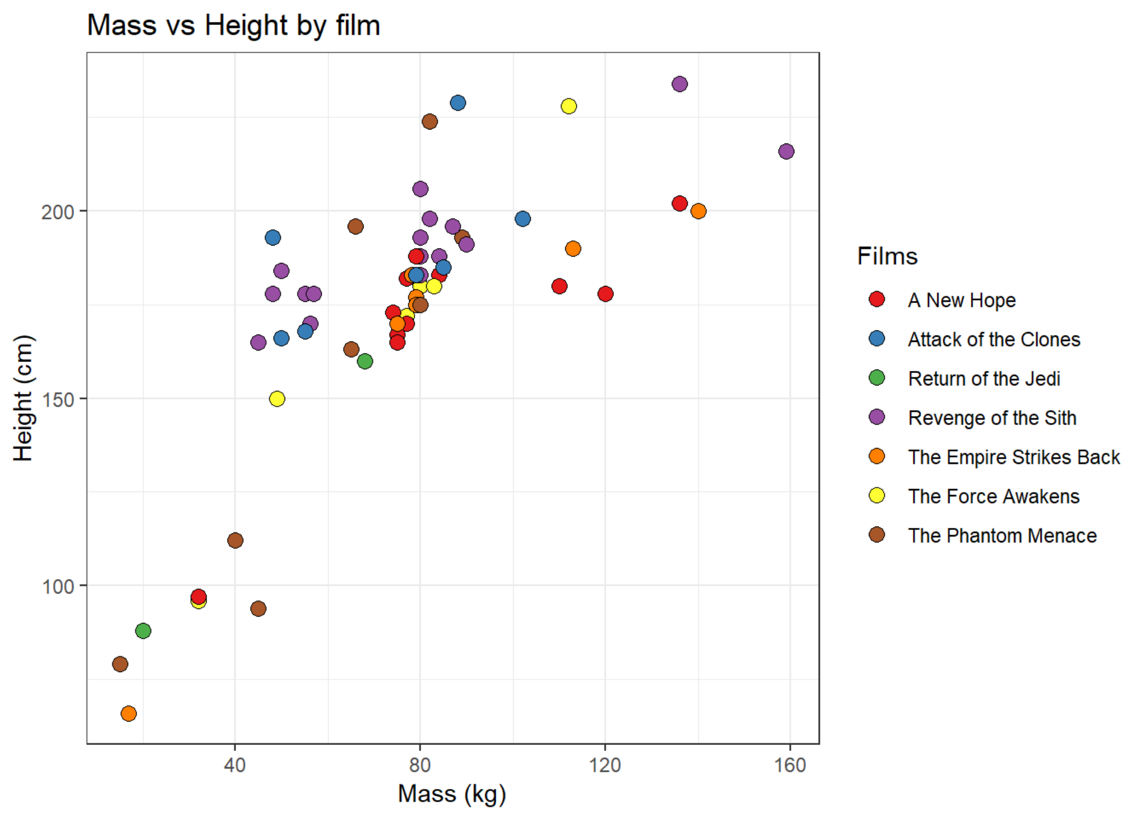
What tidyverse function do we need to apply here?



```
starwars %>% unnest(films) %>%
  ggplot(aes(x=mass,y=height)) +
    geom_point(aes(fill=films), pch=21,size=3) +
    scale_fill_brewer(palette='Set1’)+
    theme_bw()
```

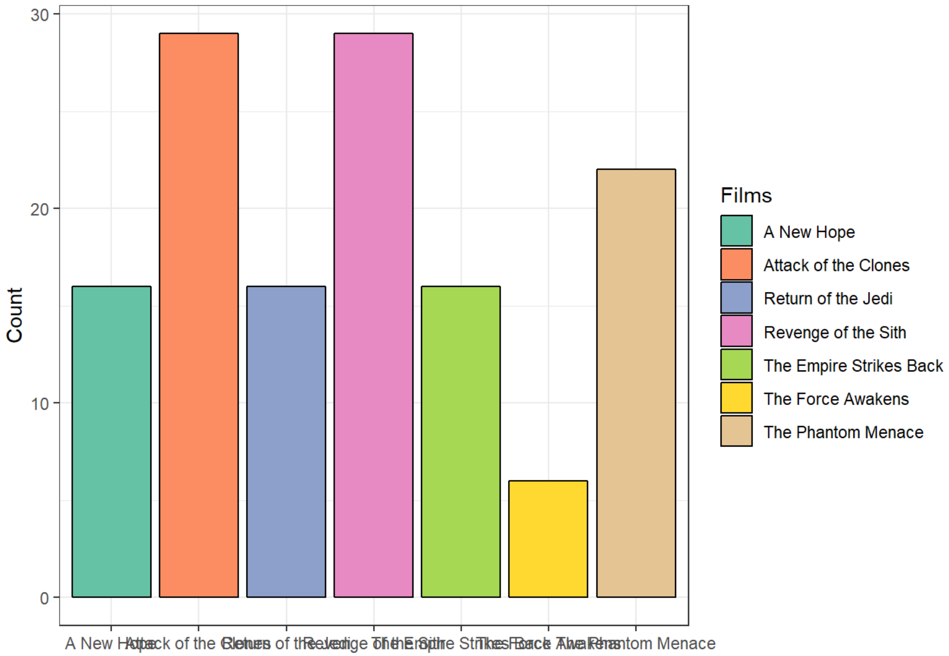
STYLING YOUR FIGURES

```
starwars %>% unnest(films) %>%
  ggplot(aes(x=mass,y=height)) +
    geom_point(aes(fill=films), pch=21,size=3) +
    scale_fill_brewer("Films",palette='Set1')+
    theme_bw() +
    xlab('Mass (kg)')+
    ylab('Height (cm)')+
  ggtitle("Mass vs Height by film")
```



Don’t worry about the messy axis labels! We can fix those with a few arguments.

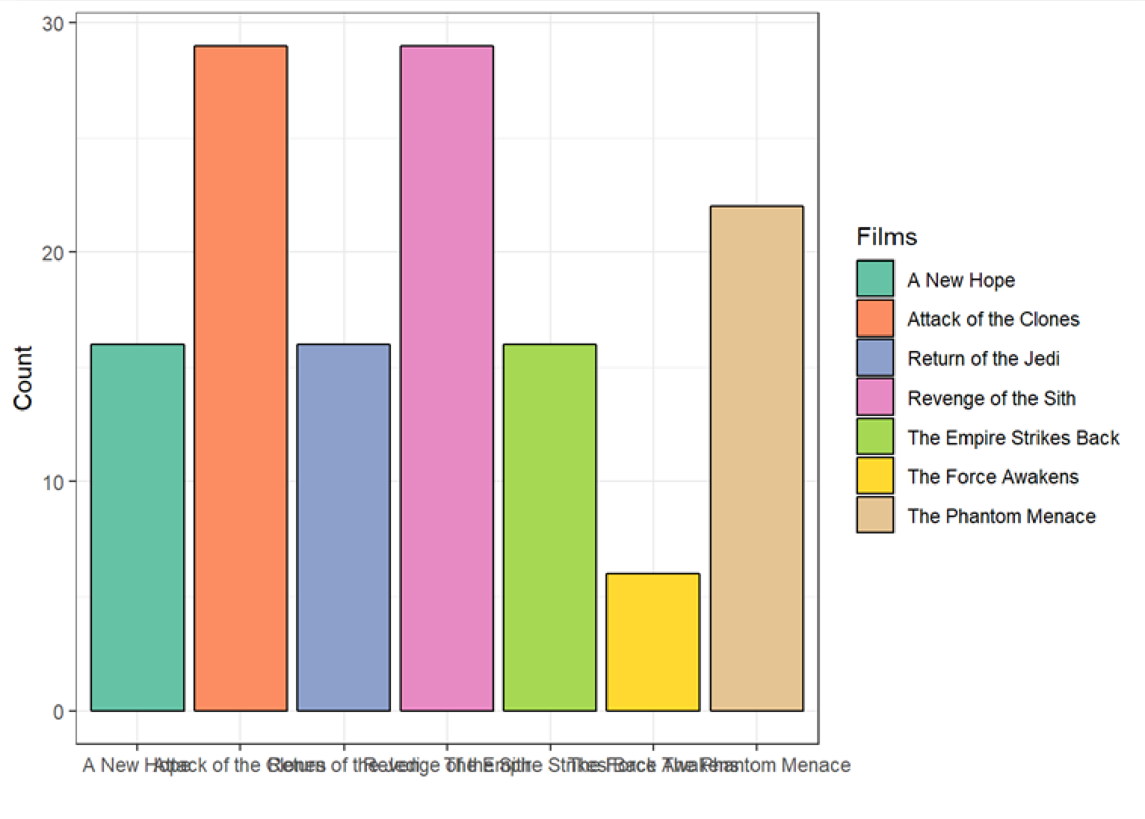
NOTE the use of scale_fill_brewer()!!!



THE BAR PLOT

Another common way to explore and display your data is the ever-helpful bar plot

THE BAR PLOT

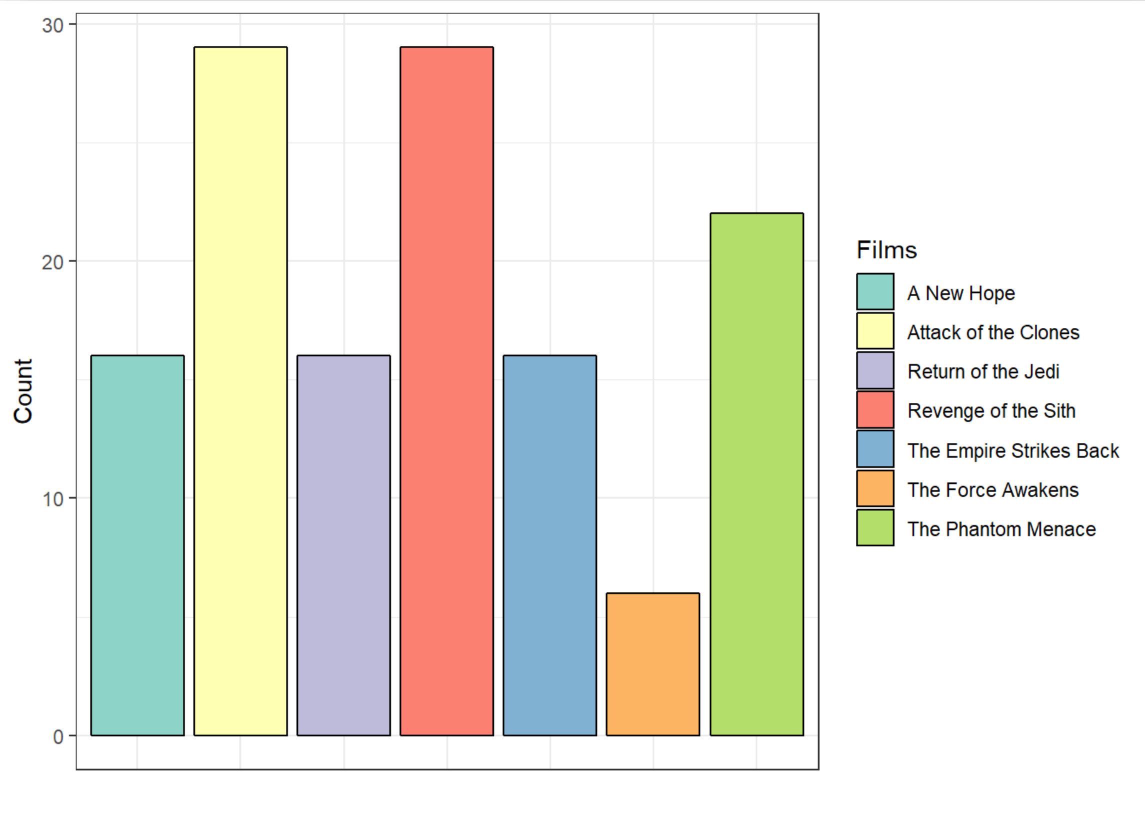


```
starwars %>% unnest(films) %>%
  ggplot(aes(x=films)) +
    geom_bar(aes(fill=films),color='black') +
    scale_fill_brewer("Films",palette='Set2')+
    theme_bw()+
    xlab("")+
    ylab("Count")
```

```
geom_bar(aes(fill=films),color='black’)
```

The geom_bar() function automatically counts the grouping element and then plots this if you don’t use the ‘stat’ argument

THE BAR PLOT

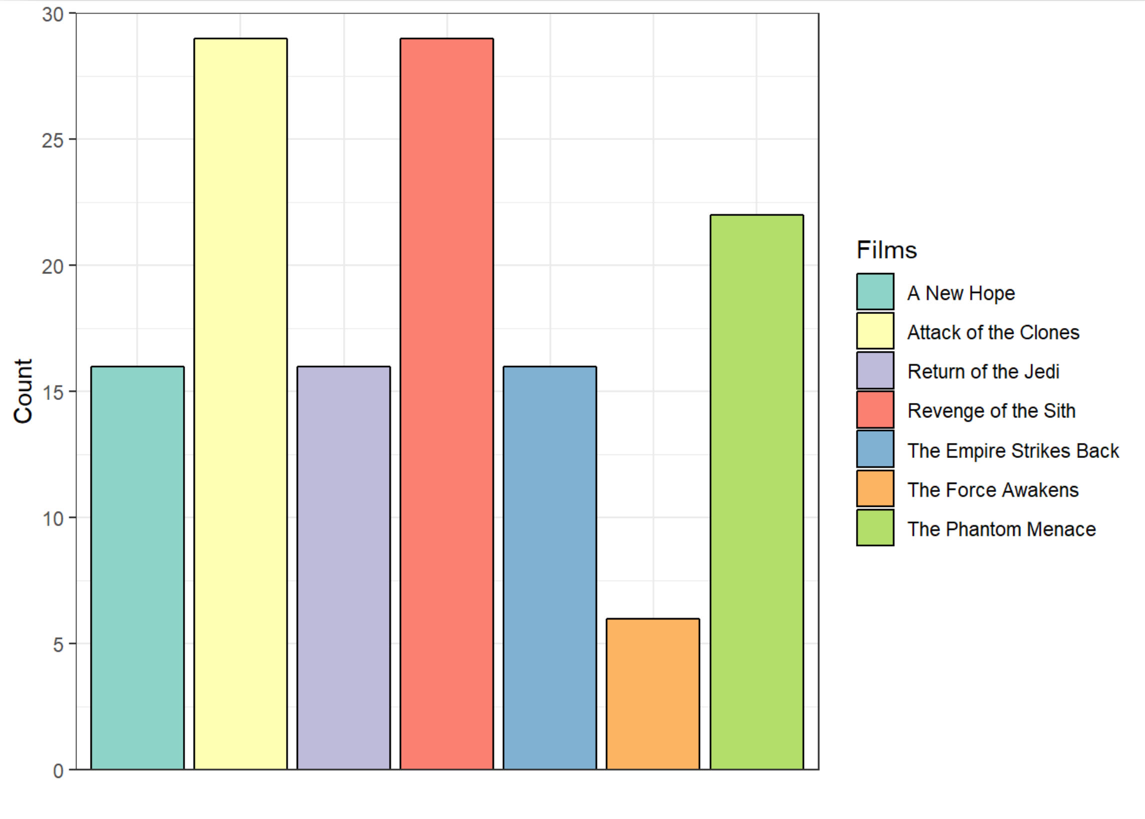


```
starwars %>% unnest(films) %>%
  ggplot(aes(x=films)) +
    geom_bar(aes(fill=films),color='black') +
    scale_fill_brewer("Films",palette='Set3')+
    theme_bw()+
    xlab("")+
    ylab("Count")+
  theme(axis.text.x = element_blank(),
              axis.ticks.x = element_blank())
```

theme() elements can over-write the theme_bw() elements so we can fix those messy x axes.

But now we have another funny problem – our bars are floating in space!!!

THE BAR PLOT

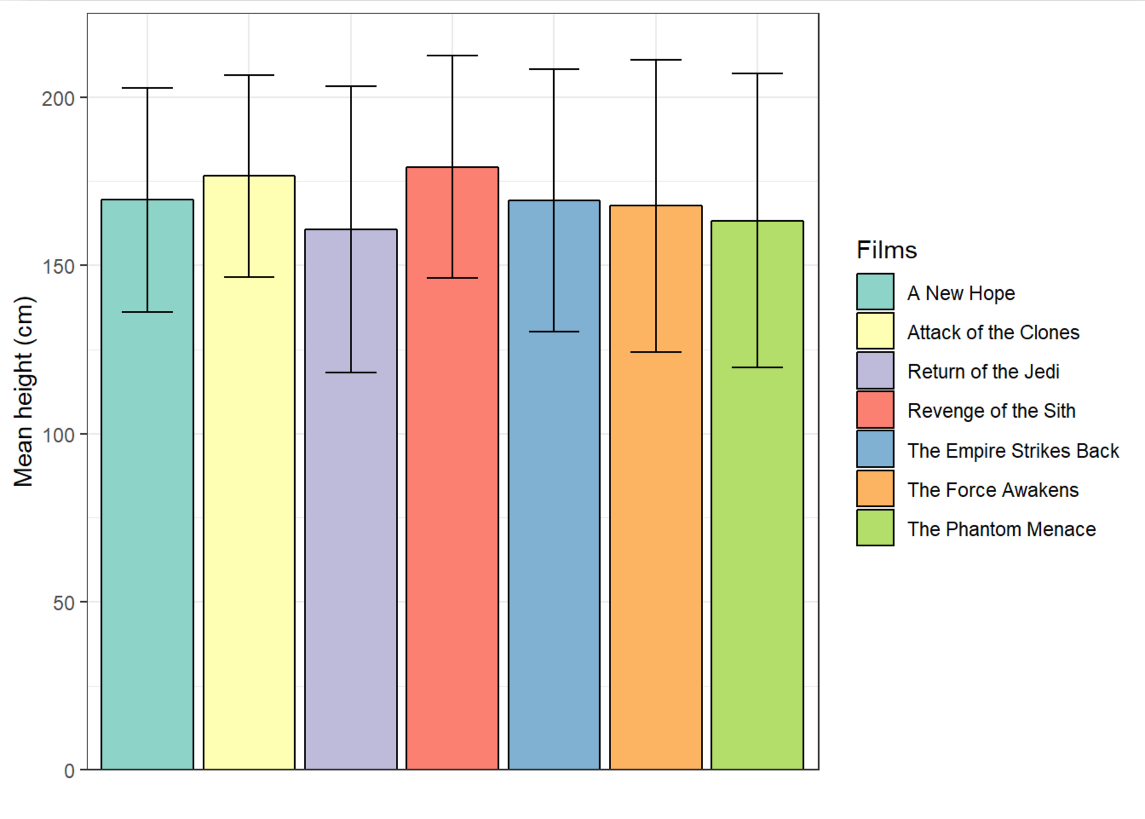


```
starwars %>% unnest(films) %>%
  ggplot(aes(x=films)) +
    geom_bar(aes(fill=films),color='black') +
    scale_fill_brewer("Films",palette='Set3')+
    theme_bw()+
    xlab("")+
    ylab("Count")+

scale_y_continuous(expand=c(0,0),breaks=seq(0,30,5)
,limits=c(0,30))+
  theme(axis.text.x = element_blank(),
             axis.ticks.x = element_blank())
```

scale_y_continuous() is used to alter the scaling of the axis!

You can also use scale_x_continuous() or scale_x/y_discrete()

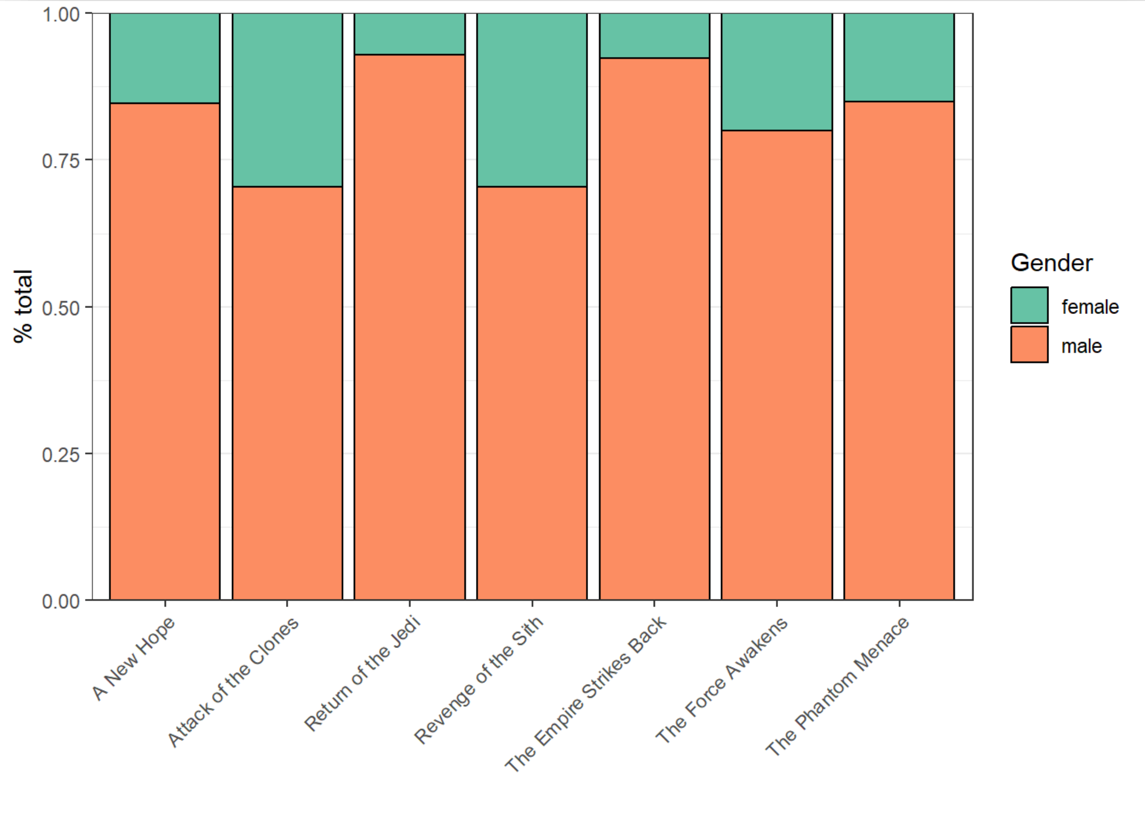
THE BAR PLOT



Let’s get crazy! Now we want to summarise some information and then add error bars around the mean!

```
starwars %>% unnest(films) %>%

  group_by(films) %>%

  summarise(meanheight =
mean(height,na.rm=T),sdheight=sd(height,na.rm=T))%>%


  ggplot(aes(x=films,y=meanheight)) +

    geom_bar(aes(fill=films),color='black',stat='identity')+

    geom_errorbar(aes(ymin=meanheight-
sdheight,ymax=meanheight+sdheight),width=0.5)+

    scale_fill_brewer("Films",palette='Set3')+

    theme_bw()+

    xlab("")+

    ylab("Mean height (cm)")+


scale_y_continuous(expand=c(0,0),breaks=seq(0,250,50),limits
=c(0,225))+

  theme(axis.text.x = element_blank(),axis.ticks.x =
element_blank())
```

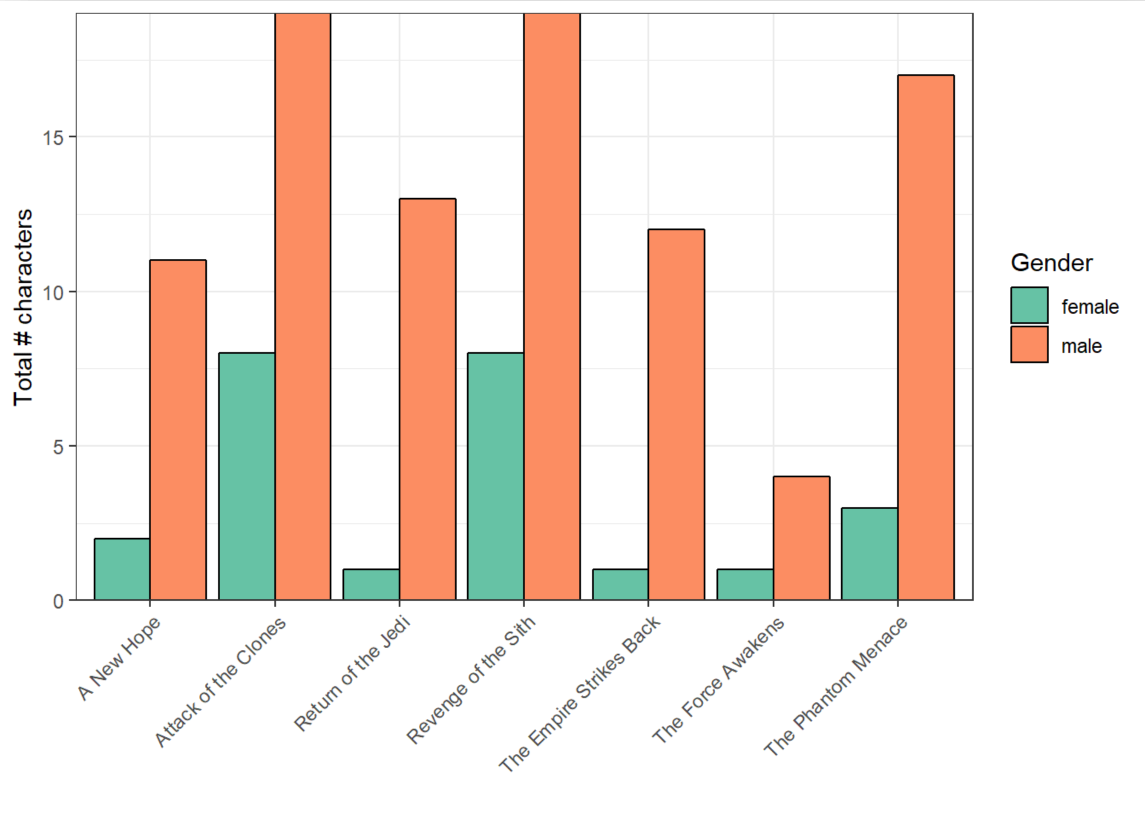
THE BAR PLOT



```
starwars %>% unnest(films) %>%
filter(!is.na(gender),gender!='none')%>%

  ggplot(aes(x=films,fill=gender)) +

  geom_bar(position='fill',color='black')+

  scale_fill_brewer("Gender",palette='Set2')+

    theme_bw()+

    xlab("")+

    ylab("% total")+

  scale_y_continuous(expand=c(0,0))+


  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

Bar plots can be stacked as well if you've got two groups you want to look at.   Using position='fill' in the geom_bar() will calculate % total of each class

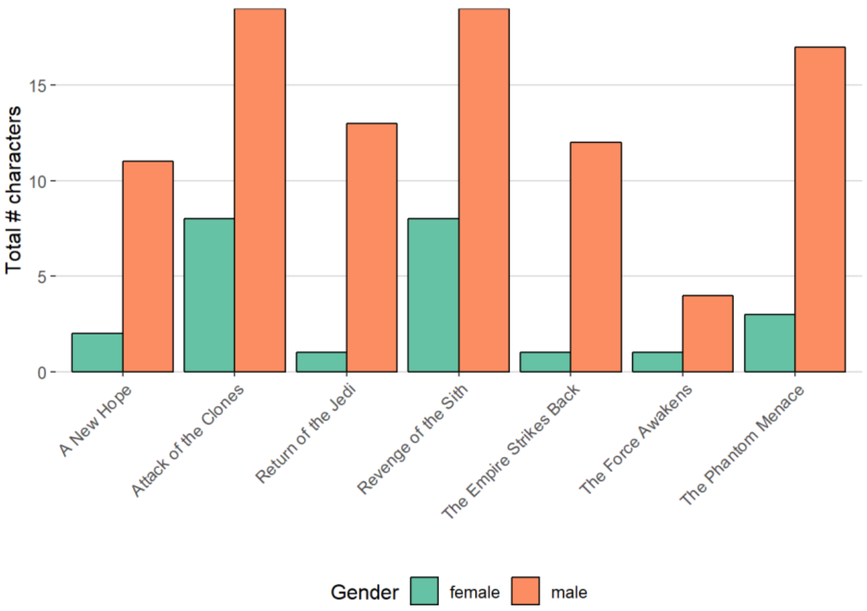
THE BAR PLOT



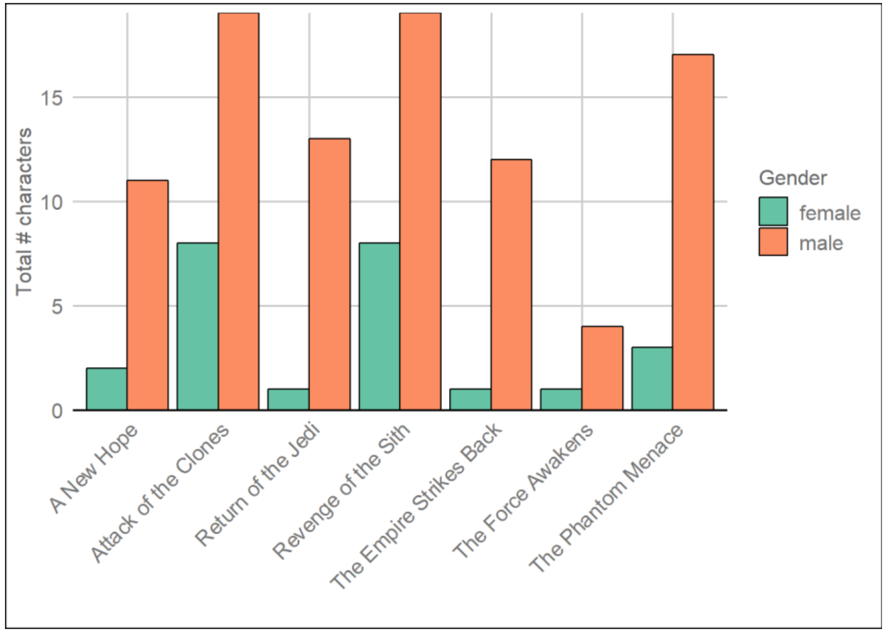
```
starwars %>% unnest(films) %>%
filter(!is.na(gender),gender!='none')%>%

  ggplot(aes(x=films,fill=gender)) +

  geom_bar(position='dodge',color='black')+

  scale_fill_brewer("Gender",palette='Set2')+

    theme_bw()+

    xlab("")+

    ylab("Total # characters")+

  scale_y_continuous(expand=c(0,0))+


  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

If you change to position=‘dodge’ then geom_bar will calculate total count and place the categories side by side

GGTHEMES PACKAGE



theme_hw()



theme_gdocs()

MAY THE
FORCE
BE WITH
YOU