


## Week 3: *Quarto & Plotting*

 EMSE 6035: Marketing Analytics for Design Decisions

 John Paul Helveston

 September 11, 2024

# Week 3: *Quarto & Plotting*

1. Intro to Quarto

QUIZ 1

2. Intro to ggplot2

3. Project attributes & levels

# Week 3: *Quarto & Plotting*

## 1. Intro to Quarto

### QUIZ 1

## 2. Intro to ggplot2

## 3. Project attributes & levels

# "Literate programming"

Treat programs as a  
"literature" understandable  
to **human beings**



Donald E. Knuth

# Quick demo

1. Open `quarto_demo.qmd`
2. Click "Render"



# Anatomy of a .qmd file

Header

Markdown text

R code

# Define overall document options in header

## Basic html page

```
---  
title: Your title  
author: Author name  
format: html  
---
```

## Add table of contents, change theme

```
---  
title: Your title  
author: Author name  
toc: true  
format:  
  html:  
    theme: united  
---
```

More on themes at

<https://quarto.org/docs/output-formats/html-themes.html>

# Render to multiple outputs

## PDF uses LaTeX

```
---  
title: Your title  
author: Author name  
format: pdf  
---
```

## Microsoft Word

```
---  
title: Your title  
author: Author name  
format: docx  
---
```

If you don't have LaTeX on your computer,  
install tinytex in R:

```
tinytex::install_tinytex()
```



# Anatomy of a .qmd file

~~Header~~

Markdown text

R code

Right now, bookmark this! 📌

<https://commonmark.org/help/>

---

(When you have 10 minutes, do this! 📌)

<https://commonmark.org/help/tutorial/>

# Headers

**# HEADER 1**

**## HEADER 2**

**### HEADER 3**

**#### HEADER 4**

**##### HEADER 5**

**##### HEADER 6**

HEADER 1

HEADER 2

HEADER 3

**HEADER 4**

HEADER 5

HEADER 6

# Basic Text Formatting

Type this...

- normal text
- *\_italic text\_*
- *\*italic text\**
- **\*\*bold text\*\***
- **\*\*\*bold italic text\*\*\***
- ~~~~strikethrough~~~~
- ``code text``

..to get this

- normal text
- *italic text*
- *italic text*
- **bold text**
- ***bold italic text***
- ~~strikethrough~~
- `code text`

# Lists

## Bullet list:

- first item
- second item
- third item

- first item
- second item
- third item

## Numbered list:

1. first item
2. second item
3. third item

1. first item
2. second item
3. third item

# Links

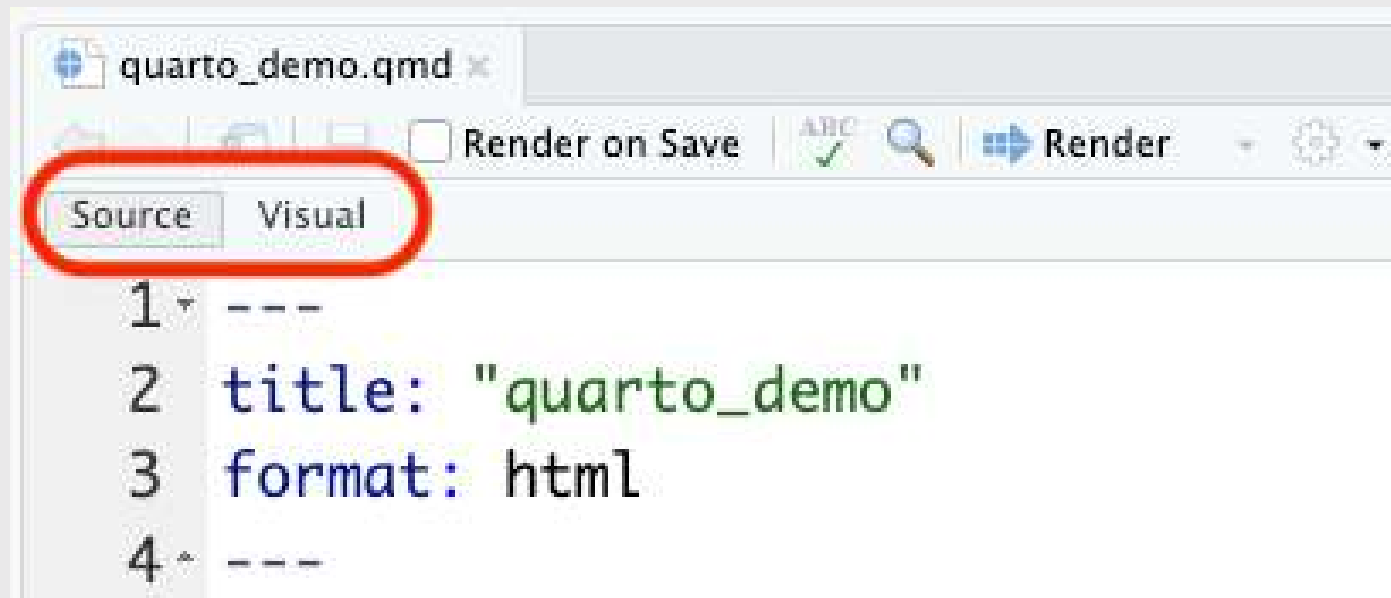
Simple **url link** to another site:

[Download R] (<http://www.r-project.org/>)

[Download R](http://www.r-project.org/)

# Don't want to use Markdown?

## Use Visual Mode!



# Anatomy of a .qmd file

~~Header (think of this as the "settings")~~

~~Markdown text~~

R code



# R Code

## Inline code

```
`r insert code here`
```

## Code chunks

```
```${r}  
insert code here  
insert more code here  
```
```

# Inline R code

```
The sum of 3 and 4 is `r 3 + 4`
```

Produces this:

The sum of 3 and 4 is 7

# R Code chunks

This code chunk...

```
```{r}
library(palmerpenguins)

head(penguins)
```
```

...will produce this when compiled:

```
library(palmerpenguins)

head(penguins)
```

```
#> # A tibble: 6 × 8
#>   species island    bill_length_mm
bill_depth_mm flipper_length_mm
body_mass_g sex      year
#>   <fct>    <fct>          <dbl>
<dbl>          <int>      <int> <fct>
<int>
#> 1 Adelie  Torgersen          39.1
18.7              181      3750 male
2007
#> 2 Adelie  Torgersen          39.5
17.4              186      3800 female
2007
#> 3 Adelie  Torgersen          40.3
```

# Chunk options

Control what chunks output using options

All options [here](#)

| option     | default  | effect  |
|------------|----------|---|
| eval       | TRUE     | Whether to evaluate the code and include its results      |
| echo       | TRUE     | Whether to display code along with its results            |
| warning    | TRUE     | Whether to display warnings                               |
| error      | FALSE    | Whether to display errors                                 |
| message    | TRUE     | Whether to display messages                               |
| tidy       | FALSE    | Whether to reformat code in a tidy way when displaying it |
| results    | "markup" | "markup", "asis", "hold", or "hide"                       |
| cache      | FALSE    | Whether to cache results for future renders               |
| comment    | "##"     | Comment character to preface results with                 |
| fig.width  | 7        | Width in inches for plots created in chunk                |
| fig.height | 7        | Height in inches for plots created in chunk               |

# Chunk output options

By default, code chunks print **code** + **output**

```
```${r}  
#| echo: false  
  
cat('hello world!')  
```
```

Prints only **output**  
(doesn't show code)

```
#> hello world!
```

```
```${r}  
#| eval: false  
  
cat('hello world!')  
```
```

Prints only **code**  
(doesn't run the code)

```
cat('hello world!')
```

```
```${r}  
#| include: false  
  
cat('hello world!')  
```
```

Runs, but doesn't print  
anything

# message / warning



>

# message / warning

Drop messages and warnings in chunk settings

```
```${r}  
#| message: false  
#| warning: false  
  
library(tidyverse)  
```
```

# A global `setup` chunk 🌍

```
```\{r}  
#| label: setup  
#| include: false  
  
knitr::opts_chunk$set(  
  warning = FALSE,  
  message = FALSE,  
  fig.path = "figs/",  
  fig.width = 7.252,  
  fig.height = 4,  
  comment = "#>",  
  fig.retina = 3  
)  
```
```

- Typically the first chunk
- All following chunks will use these options (i.e., sets global chunk options)
- You can (and should) use individual chunk options too
- Often where I load libraries, etc.



# Your turn

15:00

- 1) Open the `bears.qmd` file, and title it *"Bears Analysis"*
- 2) Create a "setup" code chunk to read in the `bear_killings.csv` data file (HINT: You might want to look back at the `quarto_demo.qmd` file!).
- 3) Use text and code to find answers each of the following questions - show your code and results to justify each answer:
  - Which months have the highest frequency of bear killings?
  - Who has been killed more often by bears: hunters or hikers?
  - How do the the number of bear attacks on men vs women compare?

HINT: Use `bears %>% count(variable)` to count how many rows are in the data for each unique value of `variable`

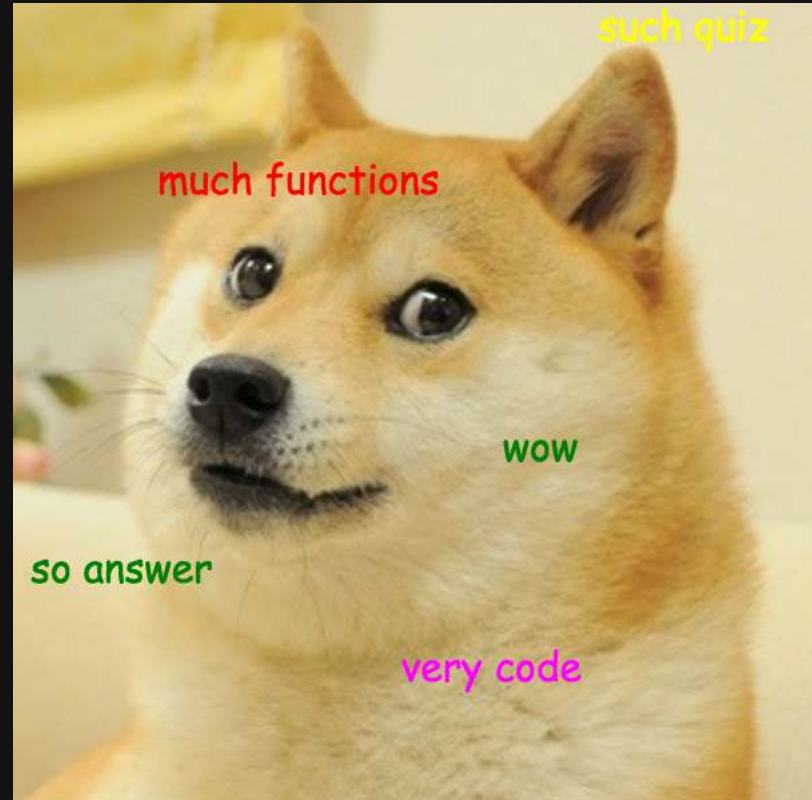
# Quiz 1

Download the template from the  
#class channel

Make sure you unzip it!

When done, submit your  
**quiz1.qmd** on Blackboard

10:00



# Week 3: *Quarto & Plotting*

1. Intro to Quarto

QUIZ 1

2. Intro to ggplot2

3. Project attributes & levels

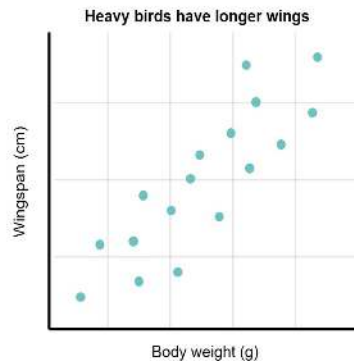
## MAKING A GRAPH WITH GGPLOT2

Customise the look of your plot with themes  
(pre-made or your own!):  
`+ theme_bw()`

Add labels and titles:  
`+ labs(x = "Body weight (g)", y = "Wingspan (cm)",  
title = "Heavy birds have longer wings")`

Specify the type of graph and the variables to use:  
`+ geom_point(aes(x = body.weight, y = wingspan))`

Plot the device containing your data:  
`ggplot(data = birds)`



# "Grammar of Graphics"

Concept developed by Leland Wilkinson  
(1999)

**ggplot2** package developed by Hadley  
Wickham (2005)

# Making plot layers with ggplot2

1. The data
2. The aesthetic mapping (what goes on the axes?)
3. The geometries (points? bars? etc.)
4. The annotations / labels
5. The theme

# Layer 1: The data

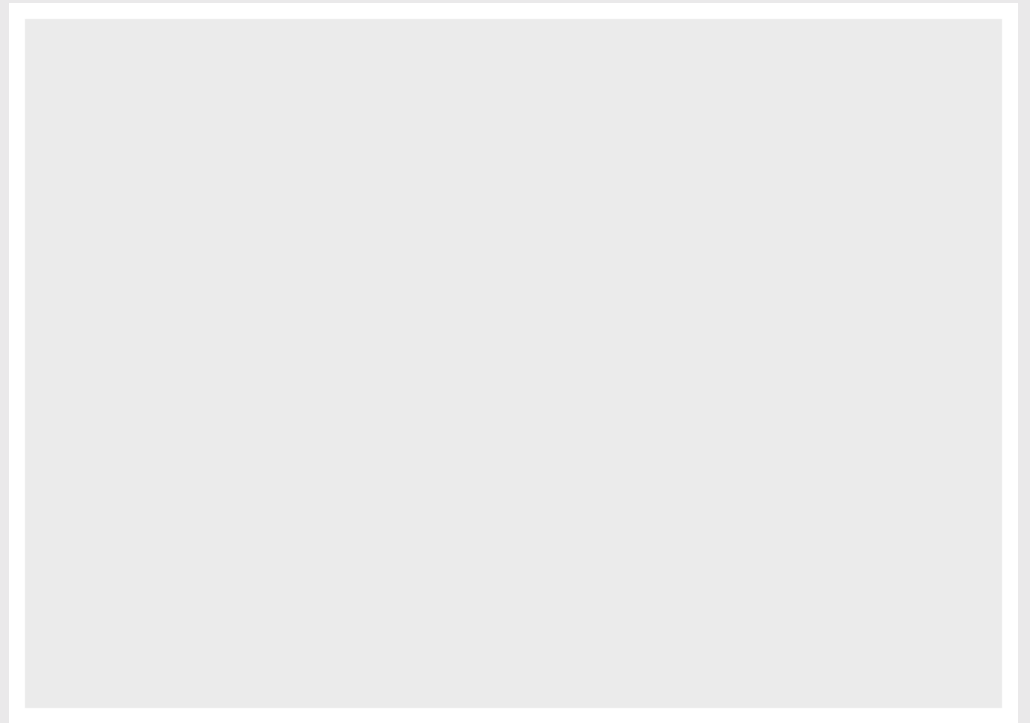
```
head(mpg)
```

```
#> # A tibble: 6 × 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(l5)  f      18    29 p    compact
#> 2 audi         a4      1.8  1999     4 manual(m5) f      21    29 p    compact
#> 3 audi         a4      2    2008     4 manual(m6) f      20    31 p    compact
#> 4 audi         a4      2    2008     4 auto(av)   f      21    30 p    compact
#> 5 audi         a4      2.8  1999     6 auto(l5)  f      16    26 p    compact
#> 6 audi         a4      2.8  1999     6 manual(m5) f      18    26 p    compact
```

# Layer 1: The data

The `ggplot()` function initializes the plot with whatever data you're using

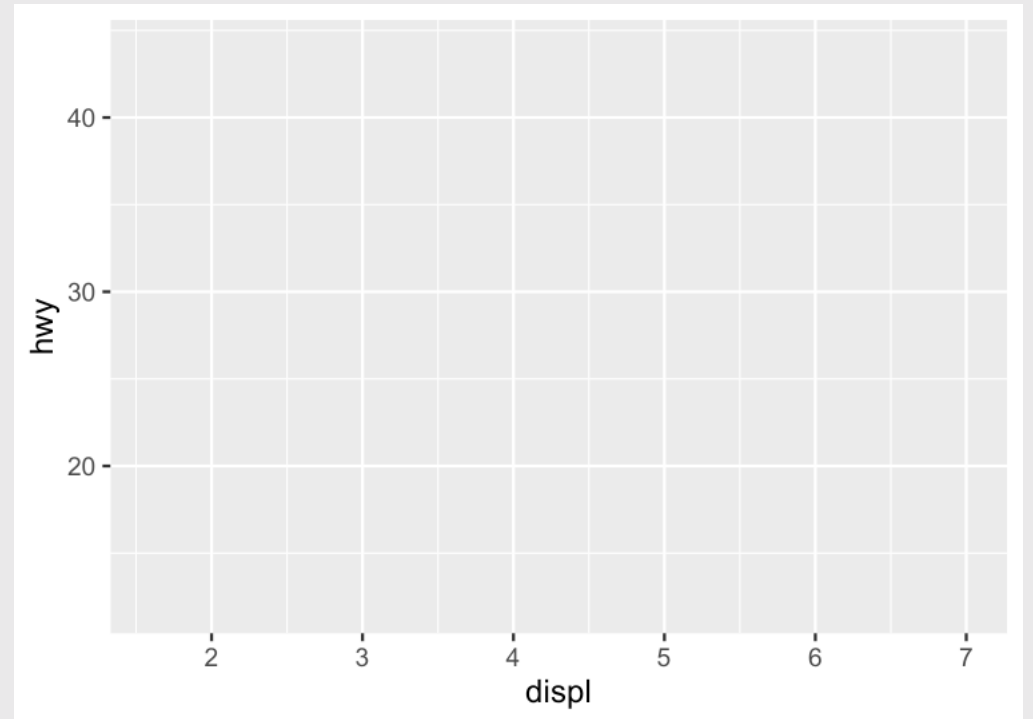
```
mpg %>%  
  ggplot()
```



# Layer 2: The aesthetic mapping

The `aes()` function determines which variables will be *mapped* to the geometries (e.g. the axes)

```
mpg %>%  
  ggplot(aes(x = displ, y = hwy))
```

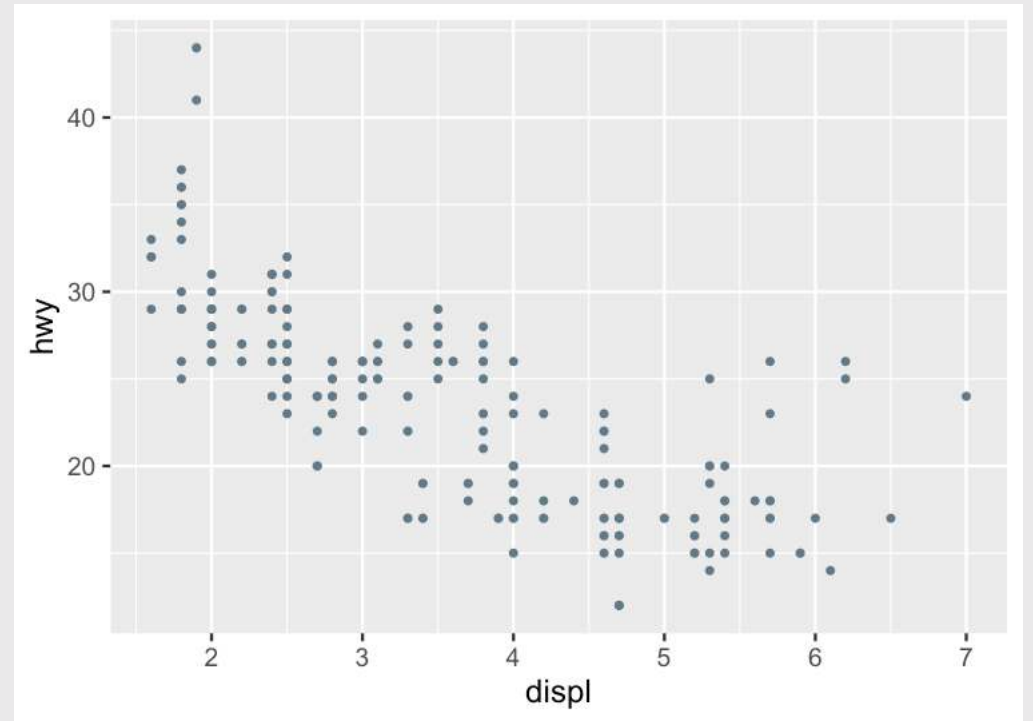




# Layer 3: The geometries

Use `+` to add geometries, e.g. `geom_points()` for points

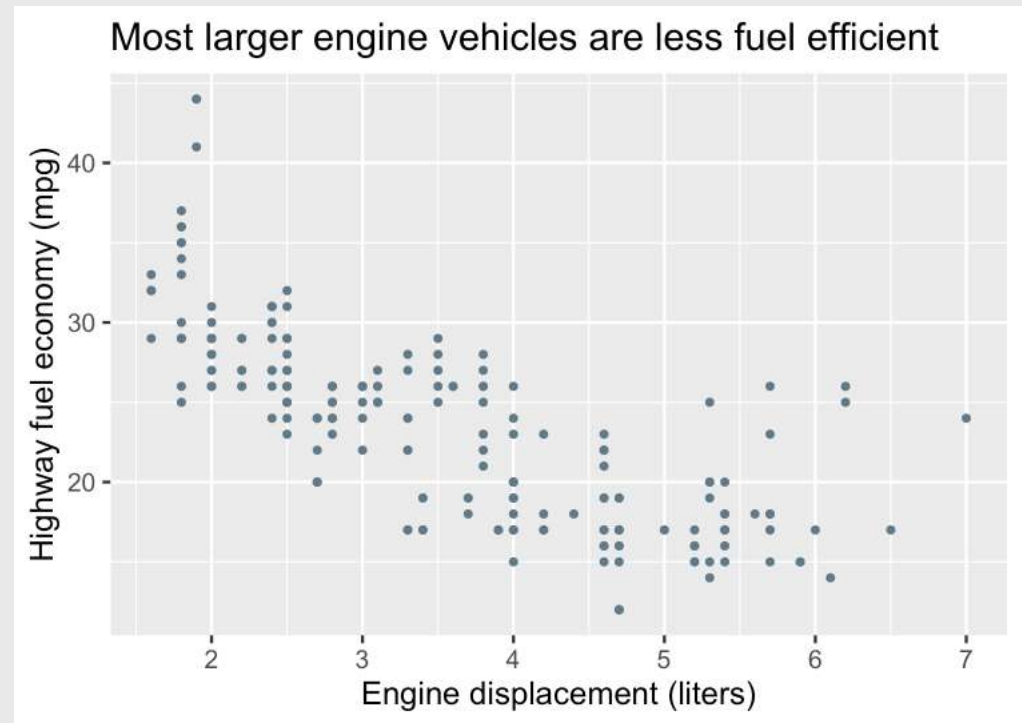
```
mpg %>%  
  ggplot(aes(x = displ, y = hwy)) +  
  geom_point()
```



# Layer 4: The annotations / labels

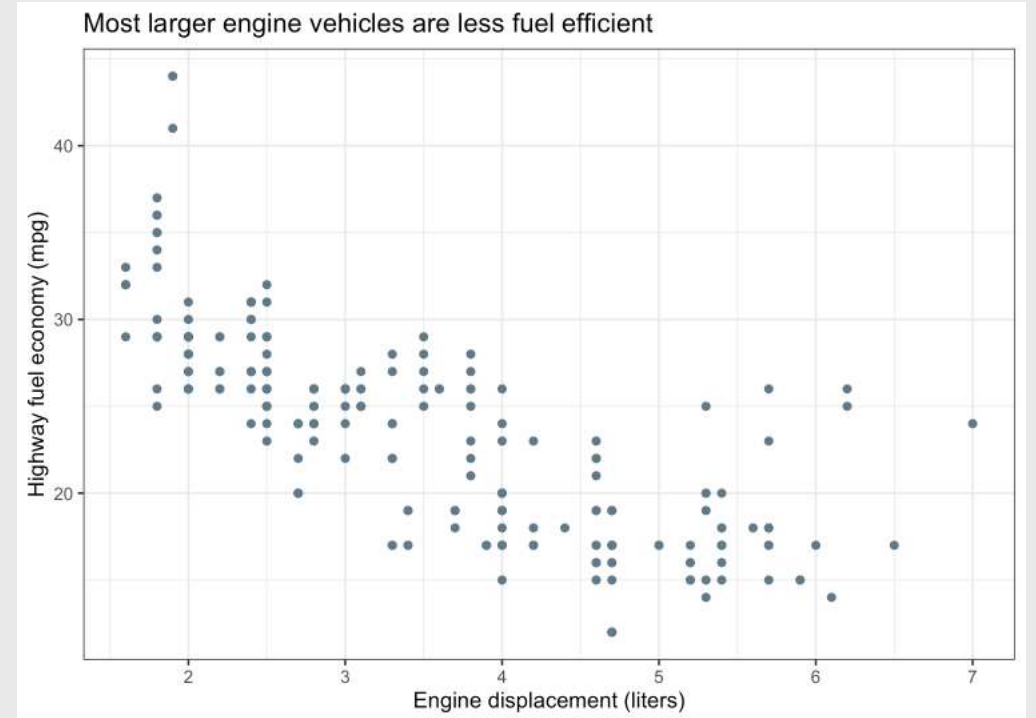
Use `labs()` to modify most labels

```
mpg %>%  
  ggplot(aes(x = displ, y = hwy)) +  
  geom_point() +  
  labs(  
    x = "Engine displacement (liters)",  
    y = "Highway fuel economy (mpg)",  
    title = "Most larger engine vehicles are  
less fuel efficient"  
  )
```



# Layer 5: The theme

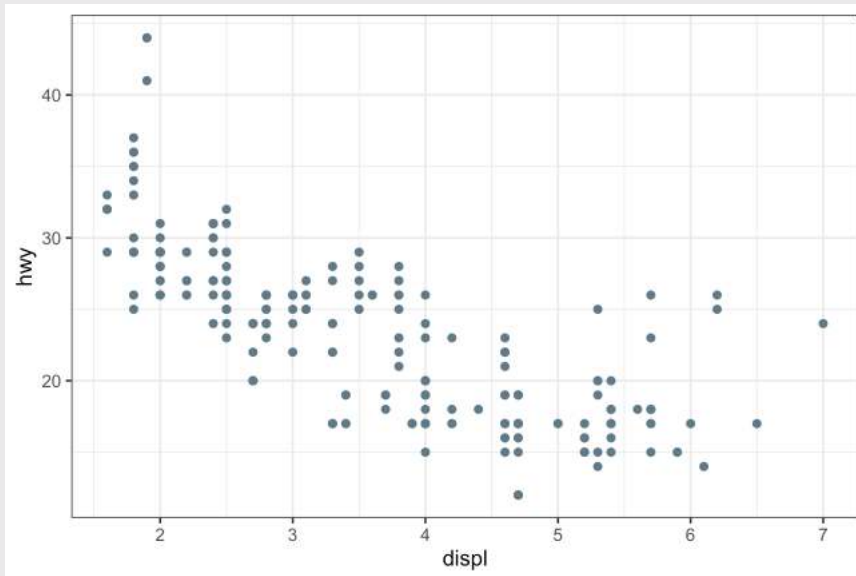
```
mpg %>%  
  ggplot(aes(x = displ, y = hwy)) +  
  geom_point() +  
  labs(  
    x = "Engine displacement (liters)",  
    y = "Highway fuel economy (mpg)",  
    title = "Most larger engine vehicles  
are less fuel efficient"  
  ) +  
  theme_bw()
```



# Common themes

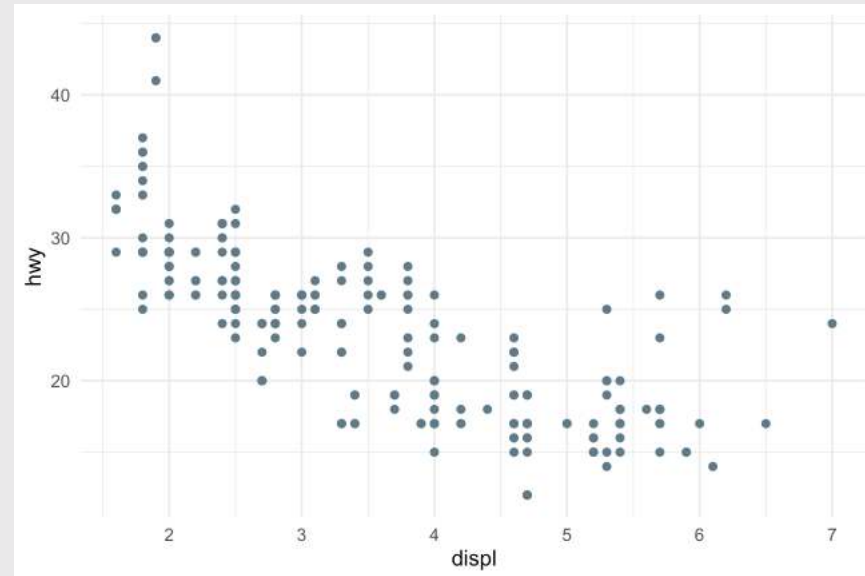
## theme\_bw()

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



## theme\_minimal()

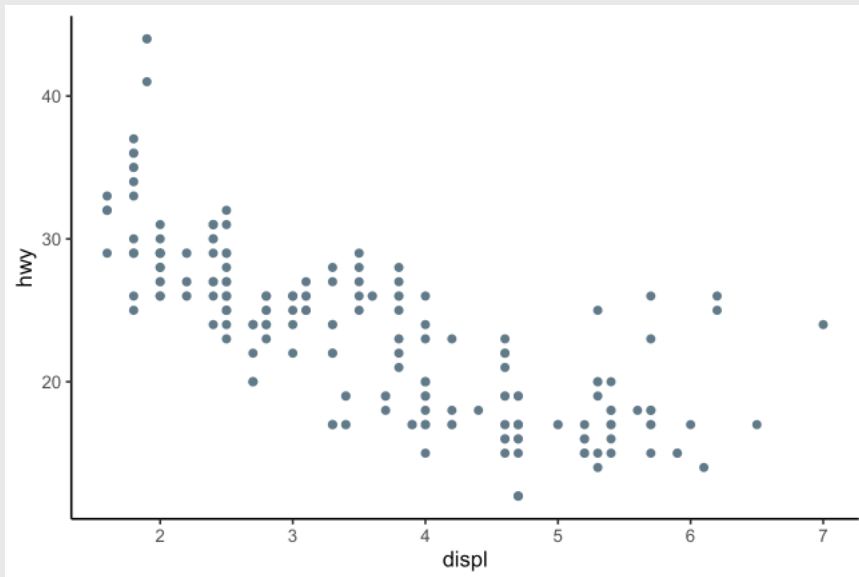
```
mpg %>%  
  ggplot(aes(x = displ, y = hwy)) +  
  geom_point() +  
  theme_minimal()
```



# Common themes

## theme\_classic()

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



## theme\_void()

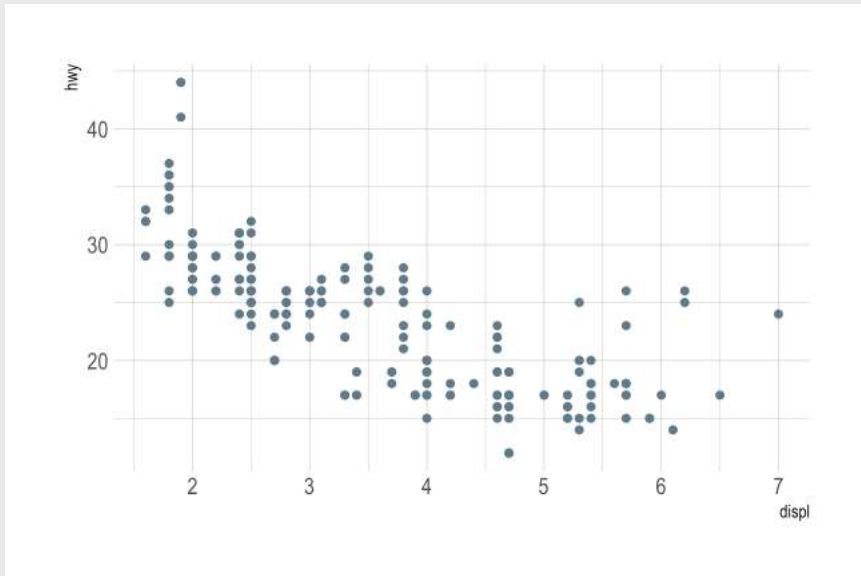
```
mpg %>%  
  ggplot(aes(x = displ, y = hwy)) +  
  geom_point() +  
  theme_void()
```



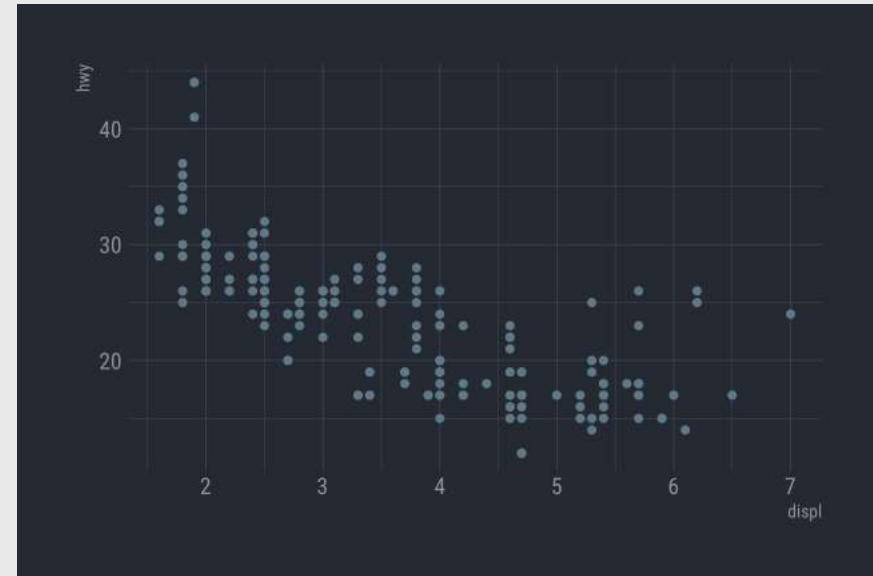
## Other themes: [hrbrthemes](#)

```
remotes::install_github("hrbrmstr/hrbrthemes")
```

```
mpg %>%  
  ggplot(aes(x = displ, y = hwy)) +  
  geom_point() +  
  hrbrthemes::theme_ipsum()
```



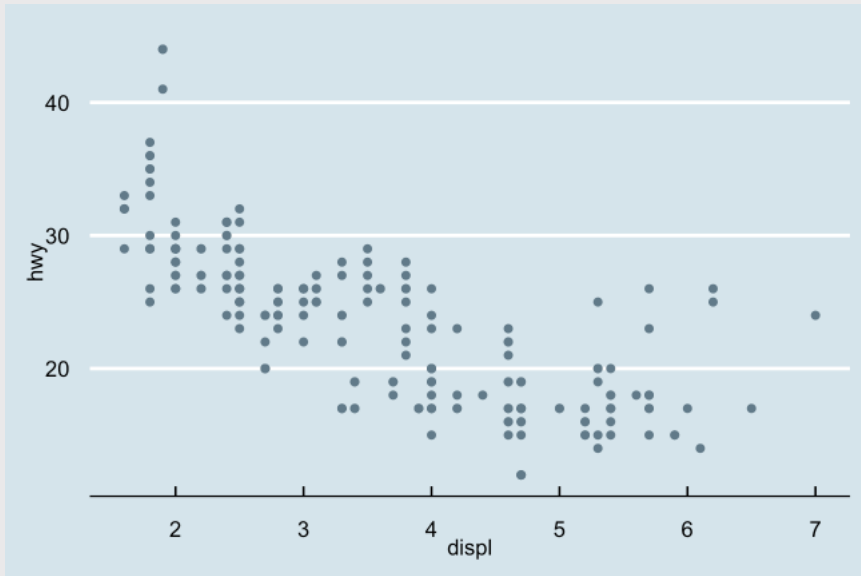
```
mpg %>%  
  ggplot(aes(x = displ, y = hwy)) +  
  geom_point() +  
  hrbrthemes::theme_ft_rc()
```



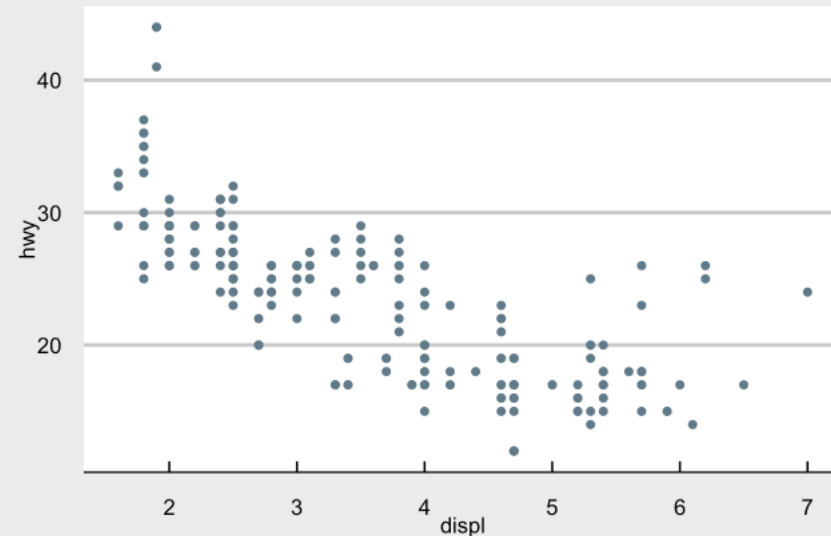
## Other themes: `ggthemes`

```
install.packages('ggthemes', dependencies = TRUE)
```

```
mpg %>%  
  ggplot(aes(x = displ, y = hwy)) +  
  geom_point() +  
  ggthemes::theme_economist()
```



```
mpg %>%  
  ggplot(aes(x = displ, y = hwy)) +  
  geom_point() +  
  ggthemes::theme_economist_white()
```



More practice

Open `ggplot2.qmd`

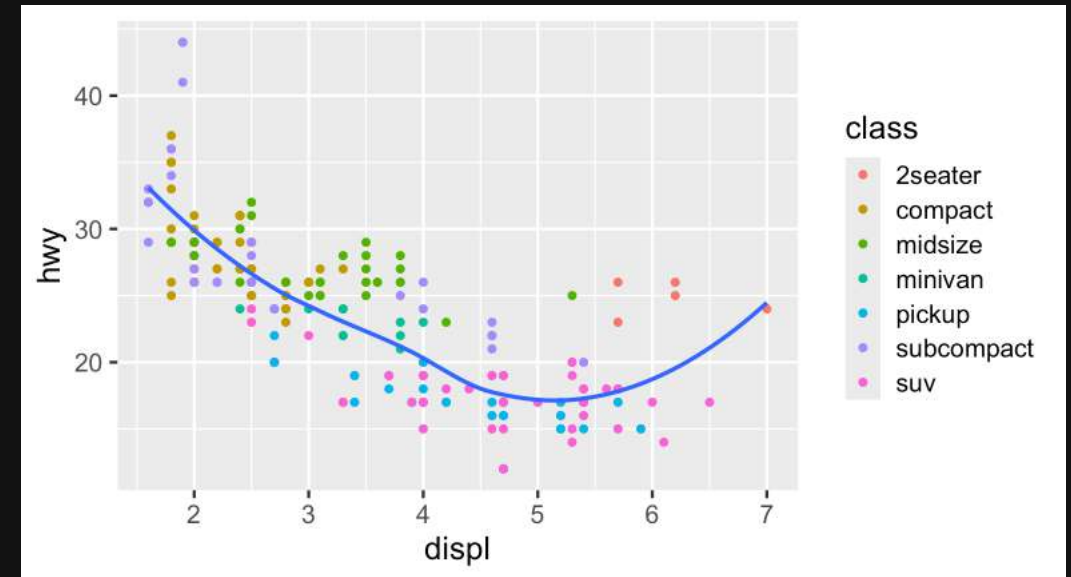
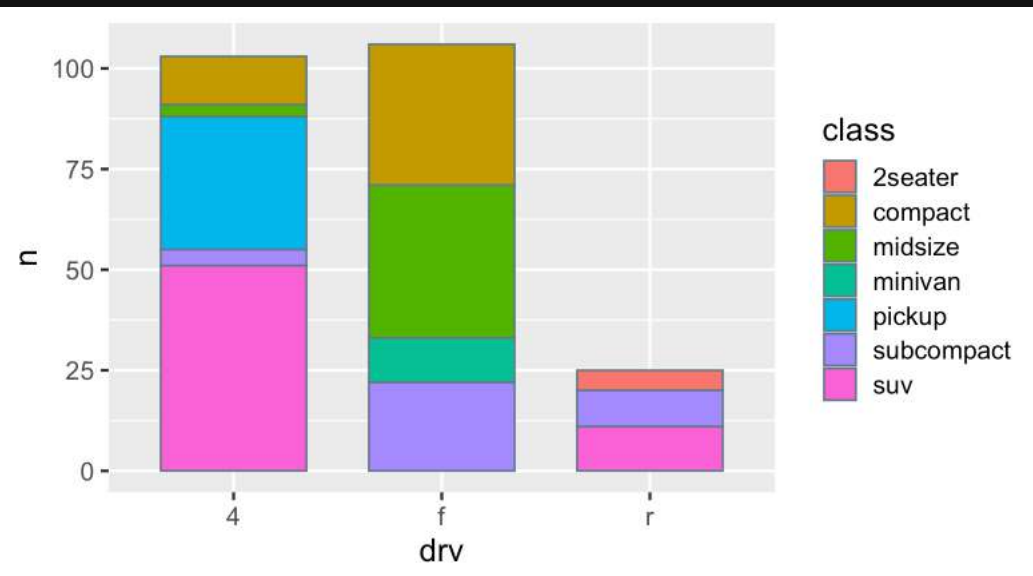
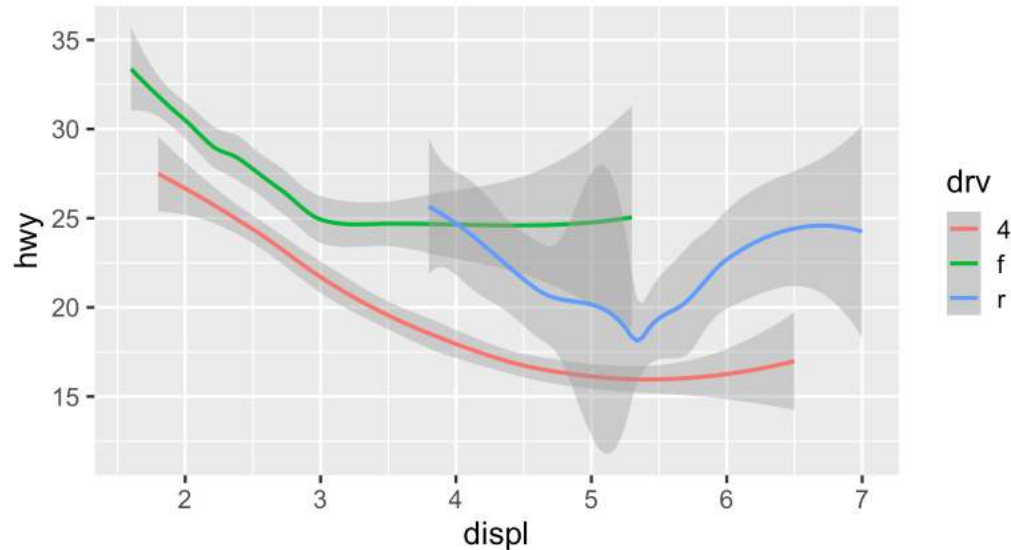


15:00

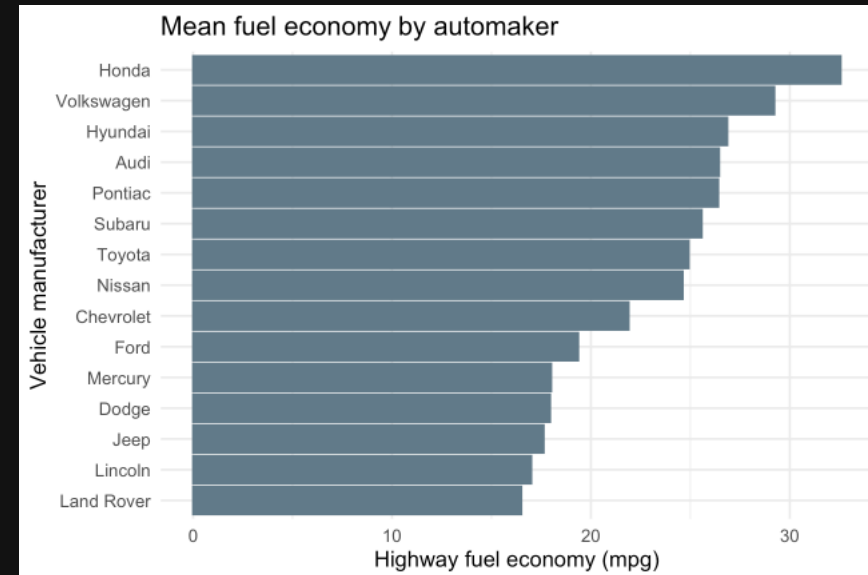
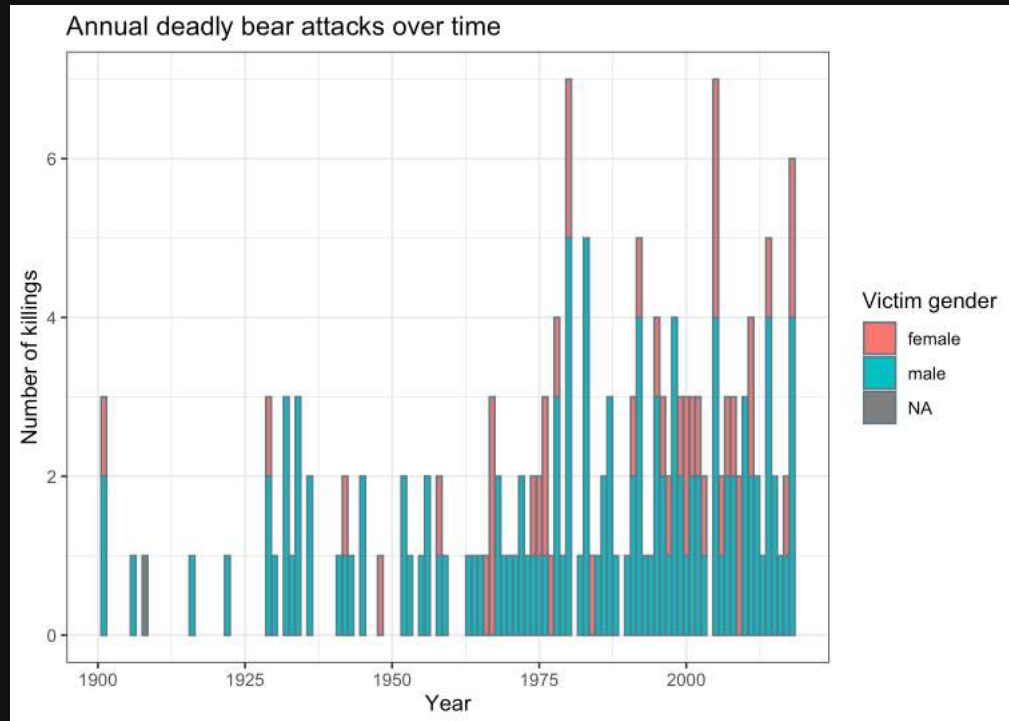
## Your turn

Open `practice.qmd`

Use the `mpg` data frame and ggplot to create these charts



# Extra practice



# Week 3: *Quarto & Plotting*

1. Intro to Quarto

QUIZ 1

2. Intro to ggplot2

3. Project attributes & levels

# Model Relationships Table (example)

|                           |              | <i>Features</i> |          |        | <i>Competitors</i> |                      |
|---------------------------|--------------|-----------------|----------|--------|--------------------|----------------------|
|                           |              | Range           | Units    | Demand | Aims Solar Panel   | SUAOKI Solar Charger |
| <i>Product Attributes</i> | Price        | \$60 - \$225    | USD / kW | -      | 225                | 160                  |
|                           | Weight       | 1 - 3           | kg       | -      | 2.6                | 2.06                 |
|                           | Power Output | 100 - 500       | W        | +      | 120                | 60                   |
|                           | Durability   | 12 - 60         | months   | +      | 60                 | 12                   |
|                           | Portability  | 200 - 2800      | cm^3     | +      | 20.6"x11"x 1.2"    | 11.5"x7.1"x2.9"      |

Start defining attribute *levels*

# Defining attribute levels

## Continuous

- **Price:** 1, 2, 3, 4, 5 (\$)
- **Power Output:** 60, 80, 120 (Watts)

- Look at competitors
- Search web for values that cover the full set of values available today (and maybe some into the future)

## Discrete

- **Color:** Red, Blue, Yellow
- **Material:** Plastic, Aluminum, Glass