

# Week 3: *R*Markdown & Plotting

🏛️ EMSE 6035: Marketing Analytics for Design Decisions

👤 John Paul Helveston

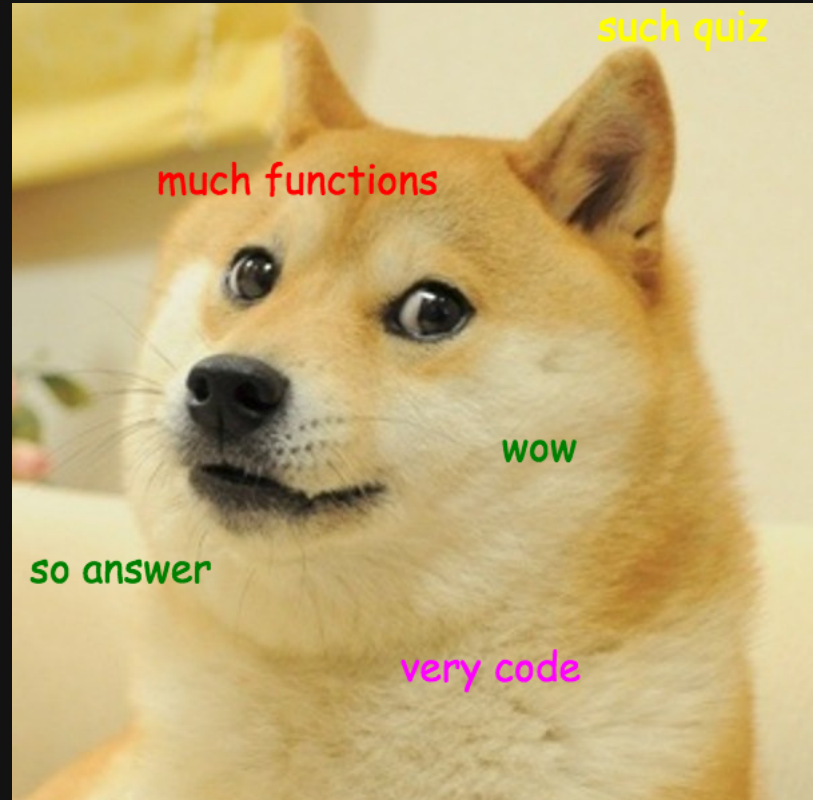
📅 September 15, 2021

# Quiz 1

Link is on the [schedule](#)

Make sure to download the zip file on the first page!

10:00



# Week 3: *R*Markdown & Plotting

1. Intro to RMarkdown

2. Intro to ggplot2

**BREAK**

3. Project attributes & levels

# Week 3: *R*Markdown & Plotting

1. Intro to RMarkdown

2. Intro to ggplot2

BREAK

3. Project attributes & levels

# "Literate programming"

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



Donald E. Knuth

# R markdown

TEXT. CODE. OUTPUT.  
(GET IT TOGETHER, PEOPLE.)



# Anatomy of a .Rmd file

YAML (**Y**et **A**nother **M**arkdown **L**anguage)

Markdown text

R code

## Quick demo

Open `rmd_demo.Rmd`, then click "knit"



# Define overall document options in YAML

## Basic html page

```
---  
title: Your title here  
author: Your name here  
output: html_document  
---
```

## Add table of contents, change theme

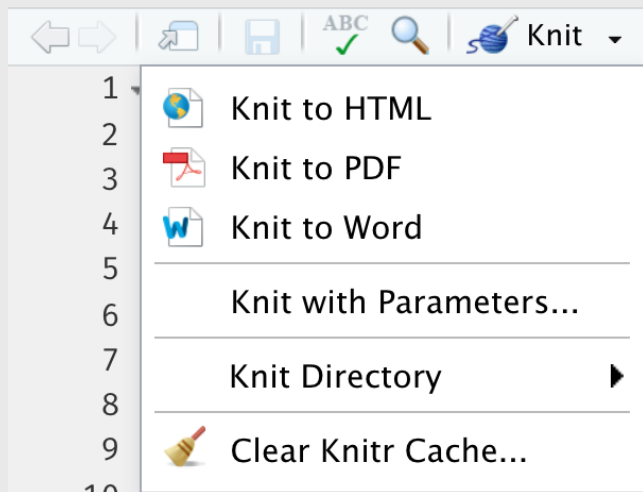
```
---  
title: Your title here  
author: Your name here  
output:  
  html_document:  
    toc: true  
    toc_float: true  
    theme: flatly  
---
```

Other themes at <https://bootswatch.com/>



# Knit to multiple outputs

```
rmarkdown::render("rmd_demo.Rmd", output_format = "all")
```



```
---  
title: Your title here  
author: Your name here  
output:  
  html_document:  
    toc: true  
    toc_float: true  
    theme: flatly  
  word_document: default  
  pdf_document: default  
---
```

# Anatomy of a .Rmd file

~~YAML (**Y**et **A**nother **M**arkdown **L**anguage)~~

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/)



Back to `rmd_demo.Rmd`

# Anatomy of a .Rmd file

~~YAML (**Y**et **A**nother **M**arkdown **L**anguage)~~

~~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}
bears %>%
  count(month)
```
```

...will produce this when compiled:

```
bears %>%
  count(month)
```

```
#> # A tibble: 12 × 2
#>   month     n
#>   <dbl> <int>
#> 1     1     3
#> 2     2     1
#> 3     3     1
#> 4     4     4
#> 5     5    18
#> 6     6    20
#> 7     7    27
#> 8     8    28
#> 9     9    25
#> 10    10    25
#> 11    11    12
```

# Chunk options

Control what chunks output using options inside `{r}`:

Example: `{r, echo=FALSE, message=FALSE}`

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

```
```${r, eval=FALSE}  
cat('hello world!')  
```
```

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

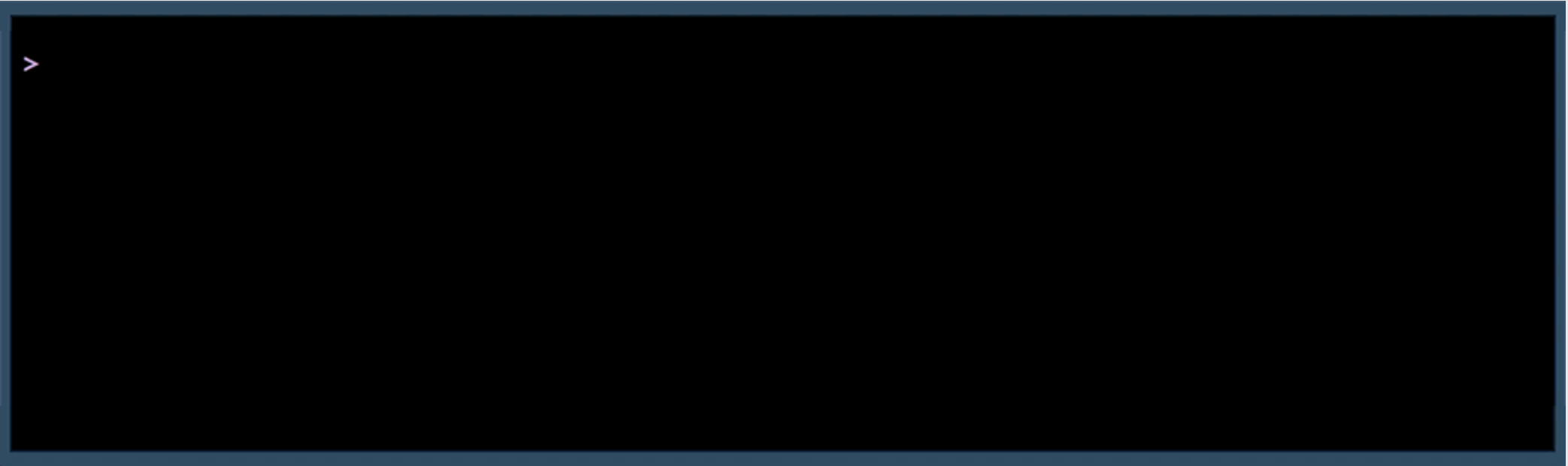
```
```${r, include=FALSE}  
cat('hello world!')  
```
```

Runs, but doesn't print  
anything

```
#> hello world!
```

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

# message / warning





# message / warning

Drop messages and warnings in chunk settings

```
```{r, message=FALSE, warning=FALSE}  
library(tidyverse)  
```
```

# A global `setup` chunk

One chunk to rule them all!

```
```${r} setup, include = FALSE}
knitr::opts_chunk$set(
  warning = FALSE,
  message = FALSE,
  comment = "#>",
  fig.retina = 3,
  fig.path = "figs/"
)
```
```

- A special chunk label: `setup`
- Typically the first chunk
- All following chunks will use these options (i.e., sets global chunk options)
- **Tip:** set `include=FALSE`
- You can (and should) use individual chunk options too

# Think Pair Share

- 1) Open the `bears.Rmd` 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 `rmd_demo.Rmd` 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`

# Week 3: *R*Markdown & Plotting

1. Intro to RMarkdown

2. Intro to ggplot2

BREAK

3. Project attributes & levels

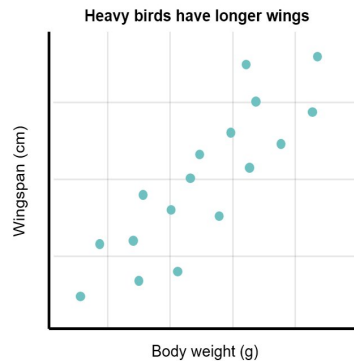
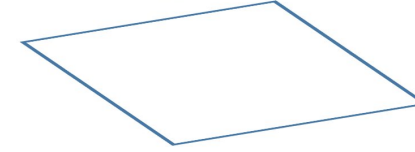
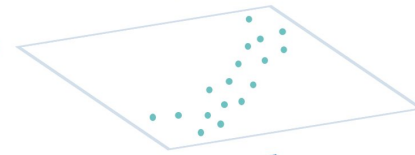
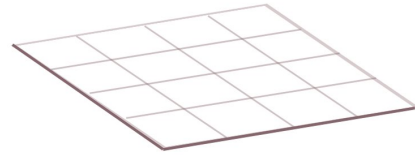
## MAKING A GRAPH WITH GGPLOT2

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

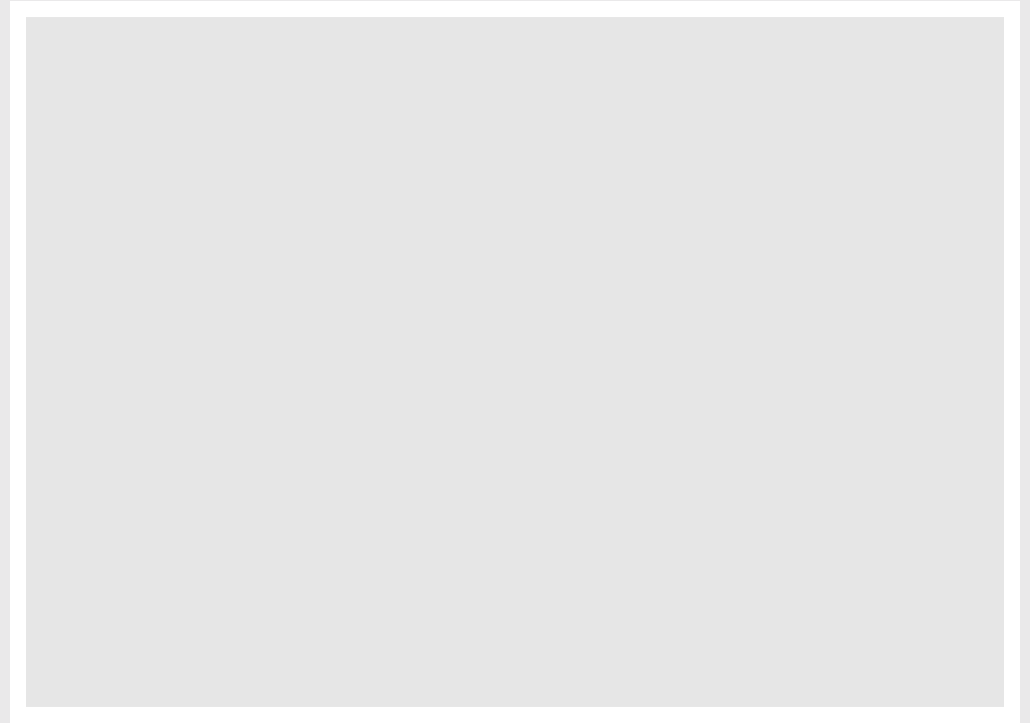
```
head(mpg)
```

```
#> # 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()
```

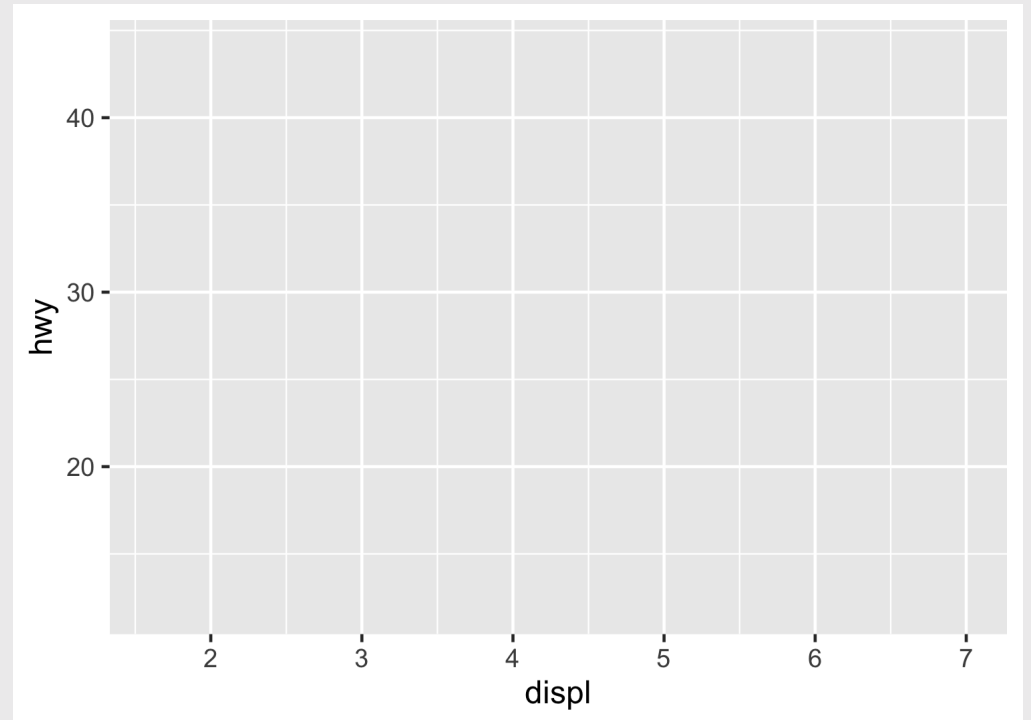




# Layer 2: The aesthetic mapping

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

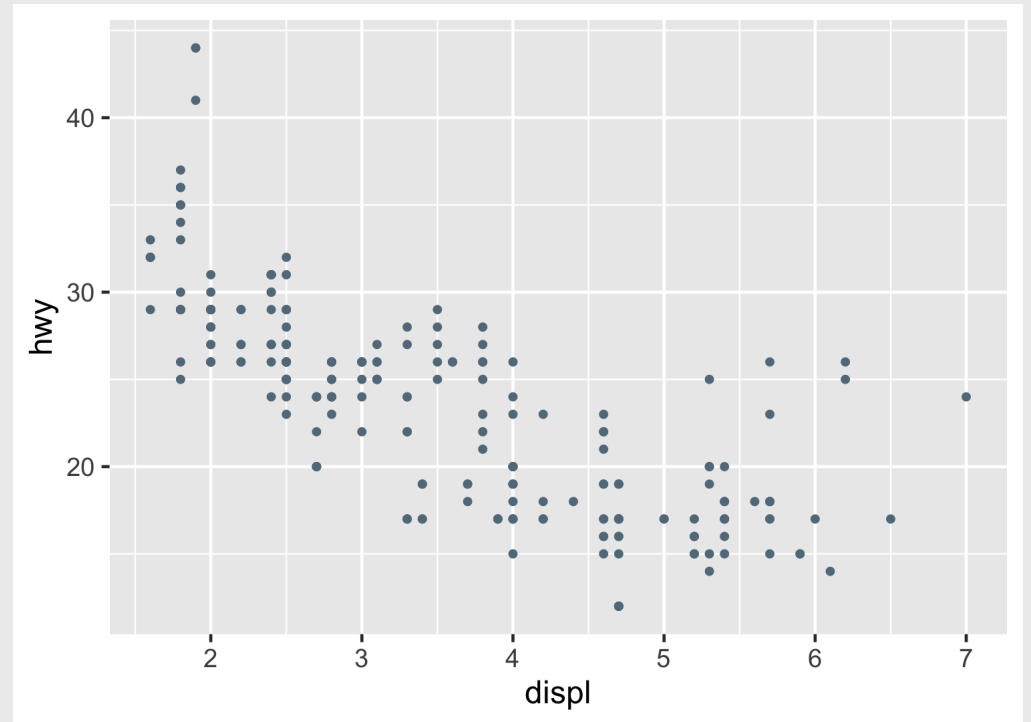
```
mpg %>%  
  ggplot(aes(x = displ, y = hwy))
```



# Layer 3: The geometries

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

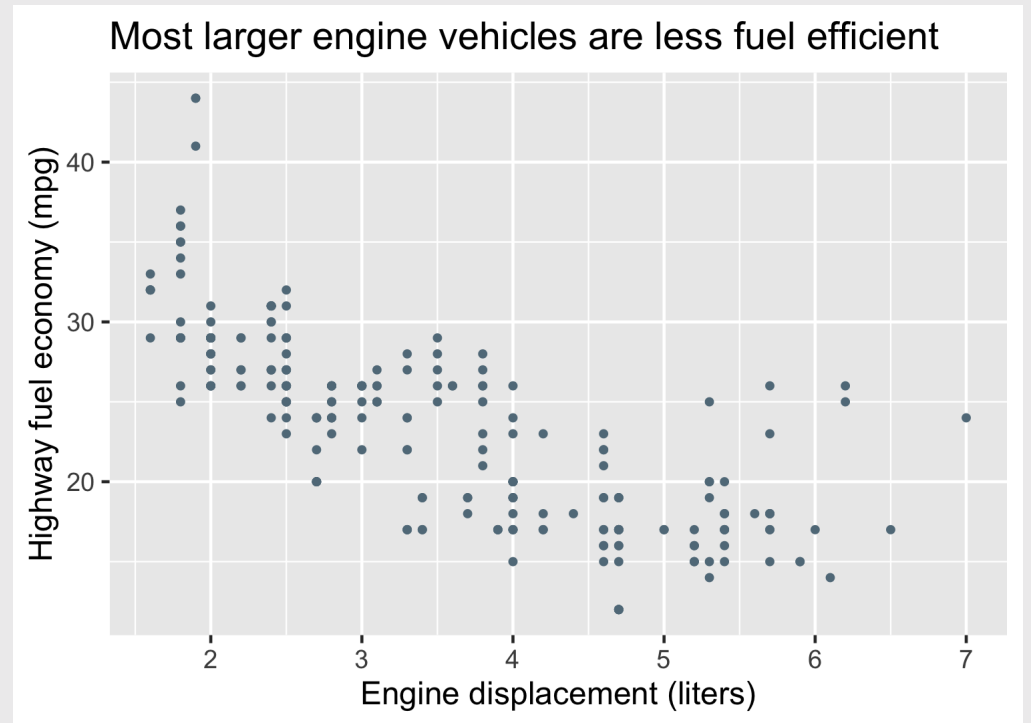
```
mpg %>%  
  ggplot(aes(x = displ, y = hwy)) +  
  geom_point()
```



# Layer 4: The annotations / labels

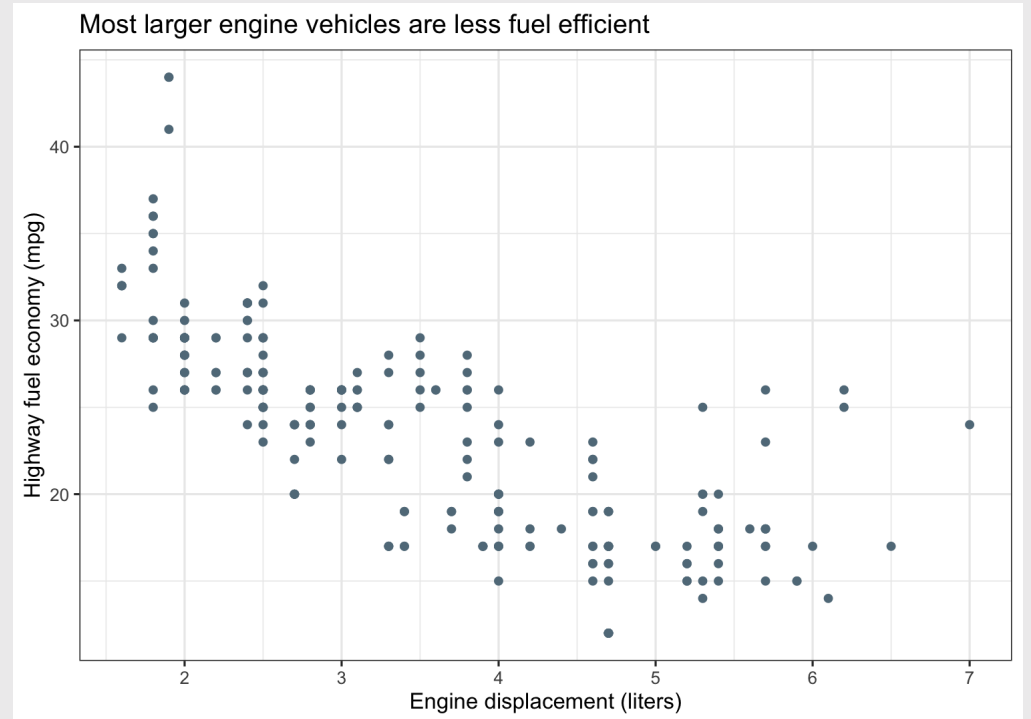
Use `labs()` to modify most labels

```
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

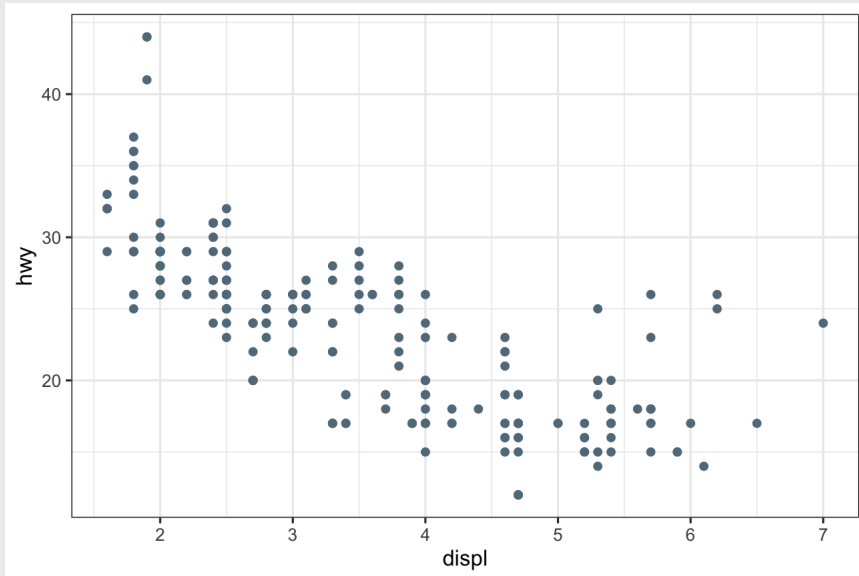
```
mpg %>%  
  ggplot(aes(x = displ, y = hwy)) +  
  geom_point() +  
  labs(  
    x = "Engine displacement (liters)",  
    y = "Highway fuel economy (mpg)",  
    title = "Most larger engine vehicles  
are less fuel efficient"  
  ) +  
  theme_bw()
```



# Common themes

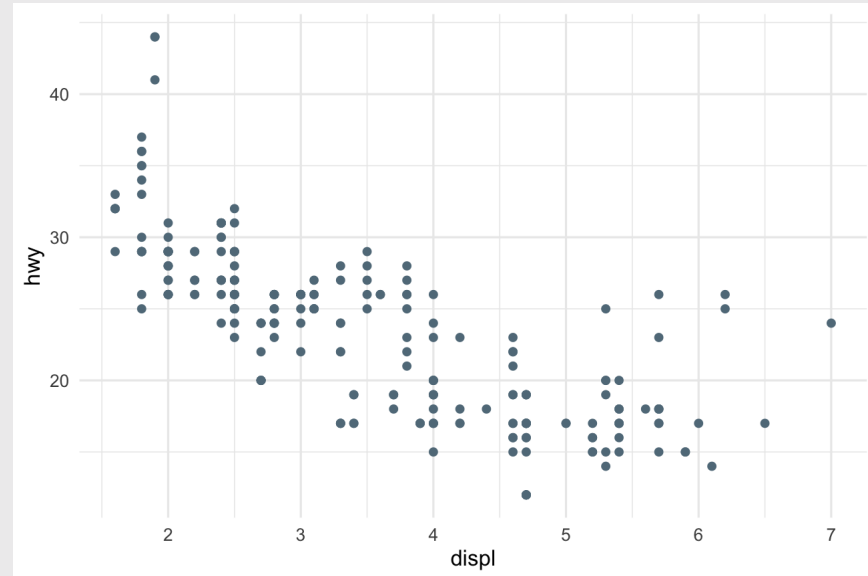
## theme\_bw()

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



## theme\_minimal()

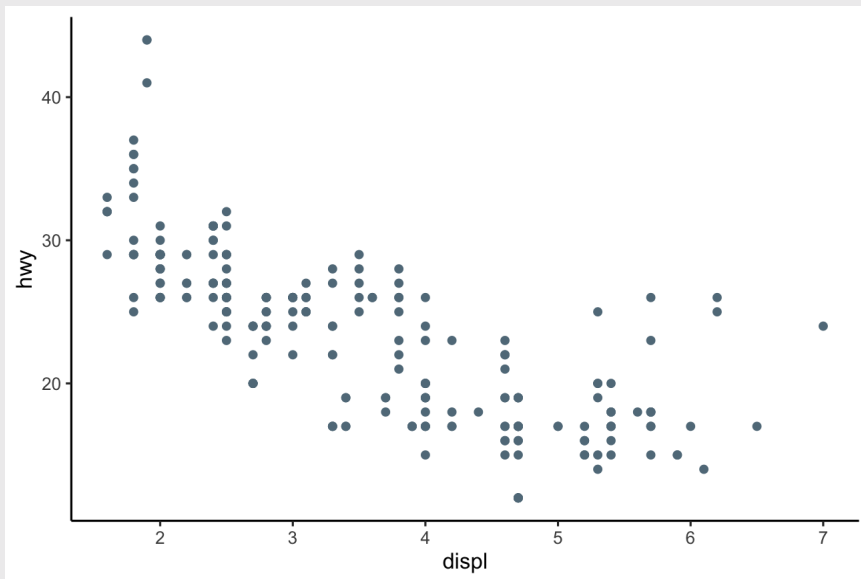
```
mpg %>%  
  ggplot(aes(x = displ, y = hwy)) +  
  geom_point() +  
  theme_minimal()
```



# Common themes

## theme\_classic()

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



## theme\_void()

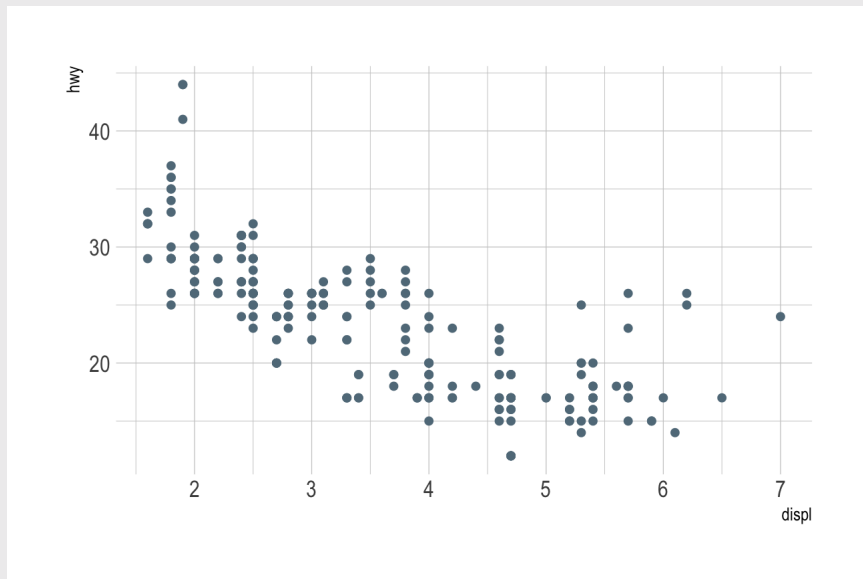
```
mpg %>%  
  ggplot(aes(x = displ, y = hwy)) +  
  geom_point() +  
  theme_void()
```



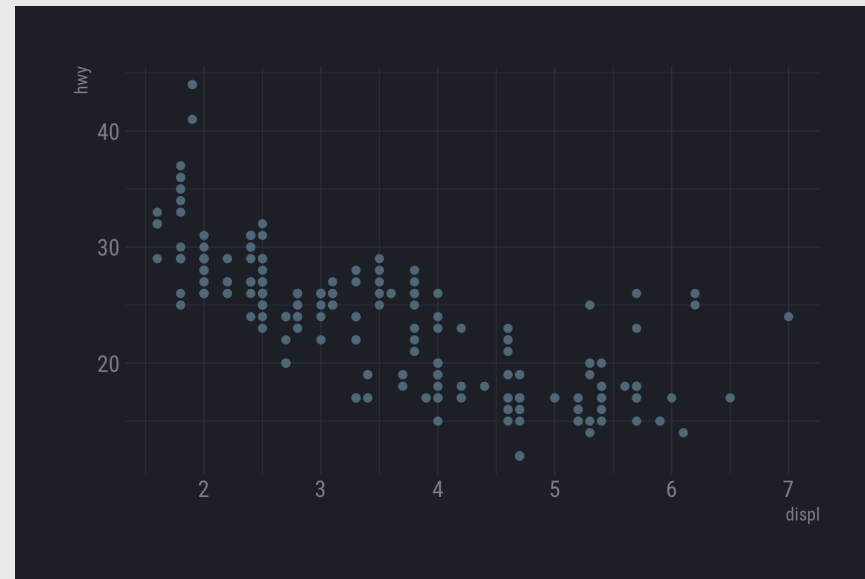
## Other themes: [hrbrthemes](#)

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

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



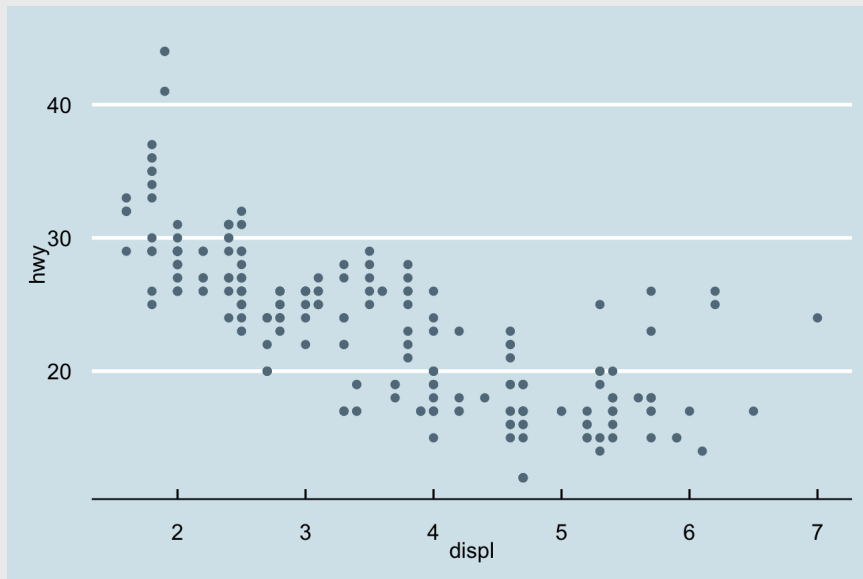
```
mpg %>%  
  ggplot(aes(x = displ, y = hwy)) +  
  geom_point() +  
  hrbrthemes::theme_ft_rc()
```



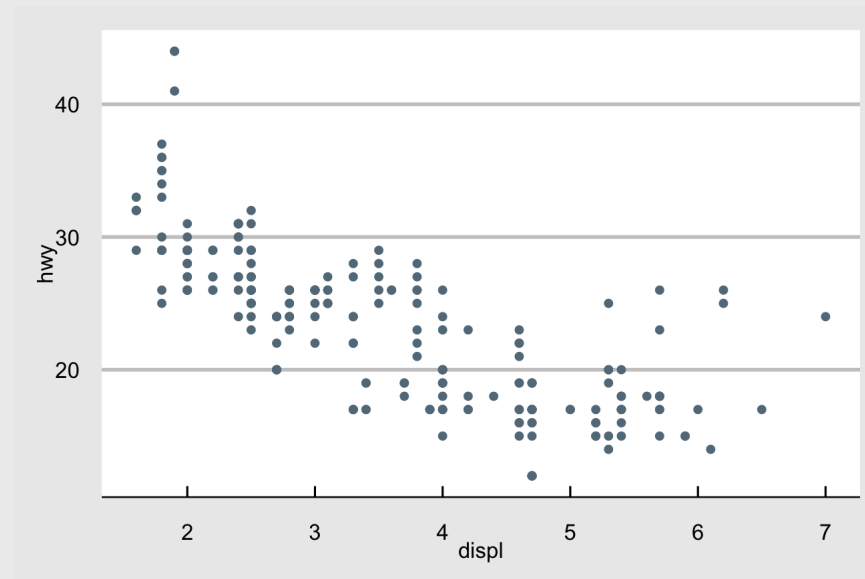
# Other themes: `ggthemes`

```
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()
```





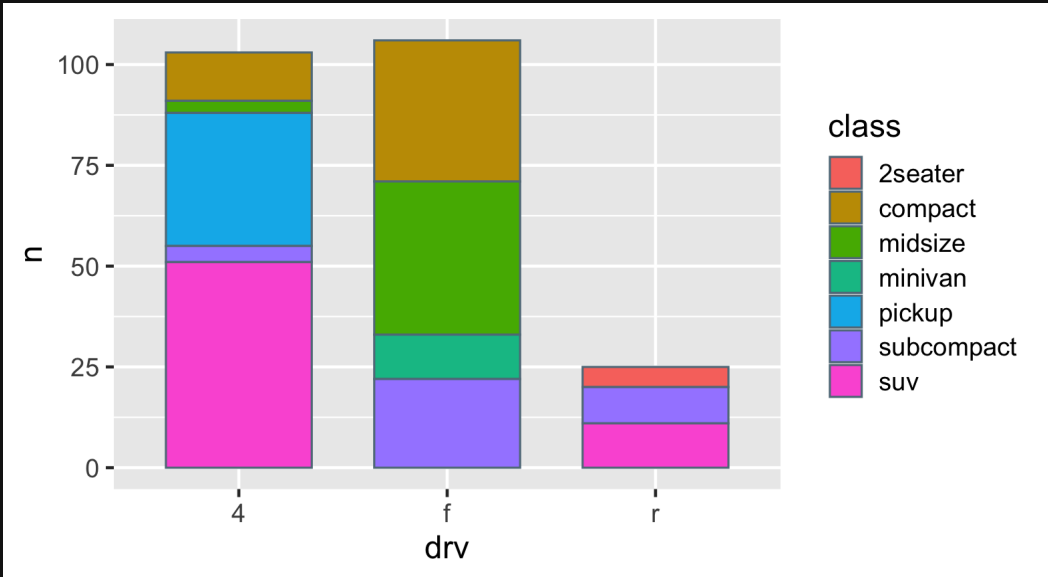
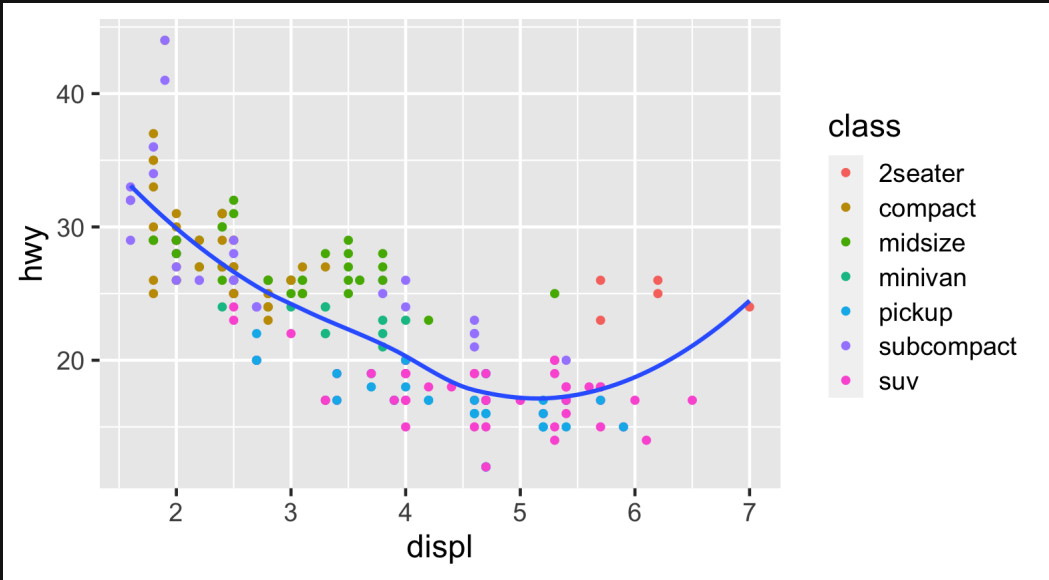
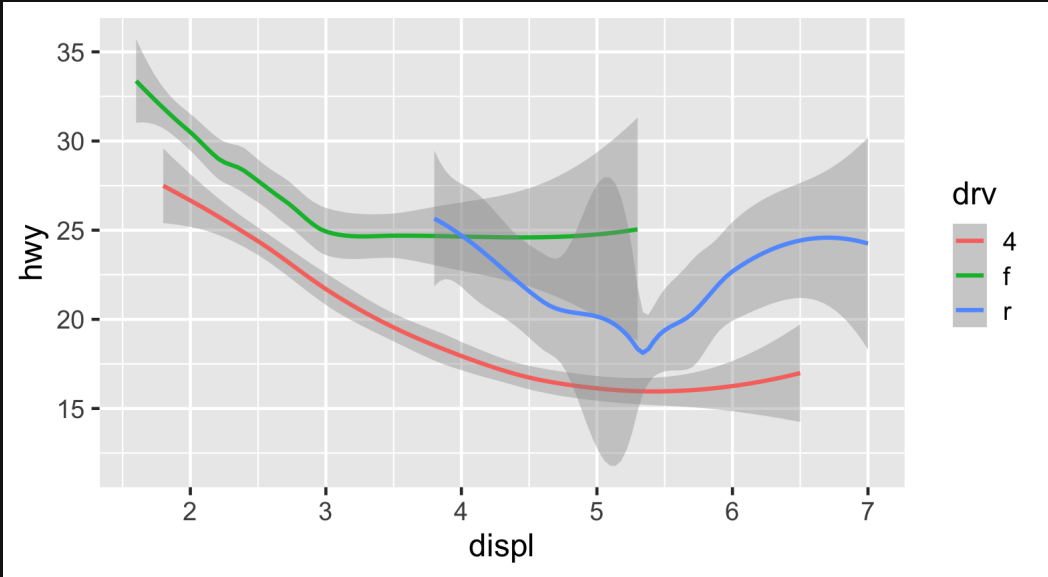
More practice

Open `ggplot2.Rmd`

15:00

# Think Pair Share

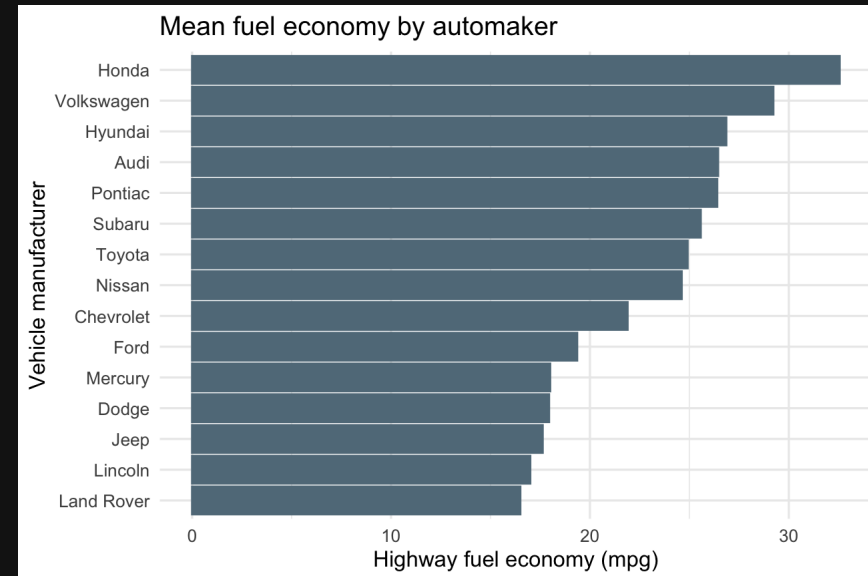
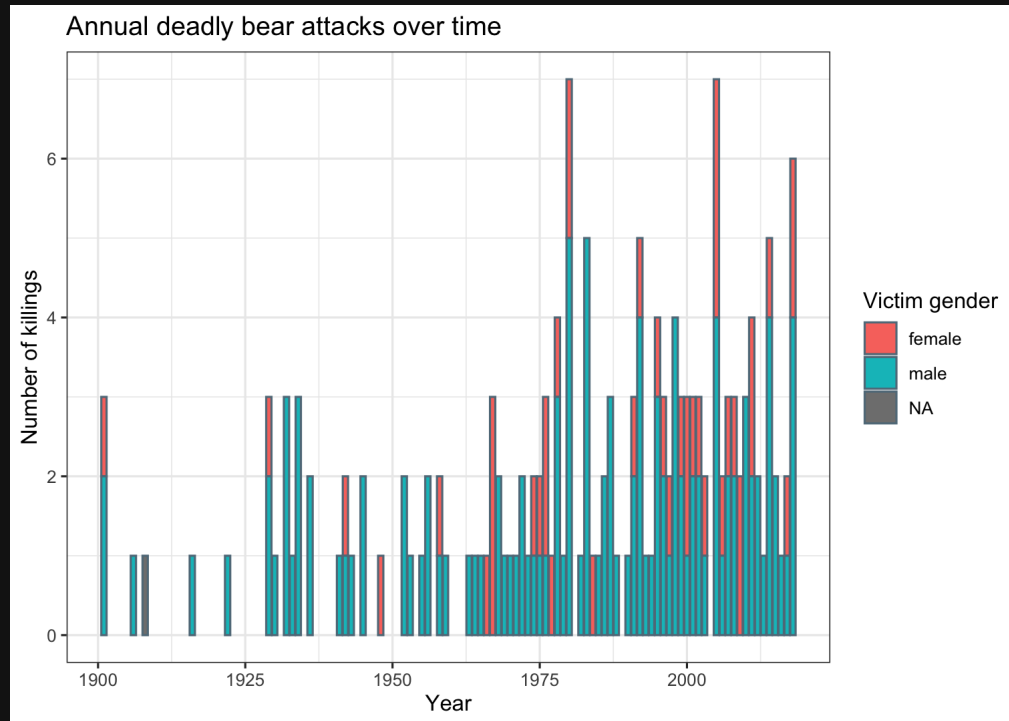
Use the `mpg` data frame and `ggplot` to create these charts



*Break*

05:00

# Extra practice



# Week 3: *R*Markdown & Plotting

1. Intro to RMarkdown

2. Intro to ggplot2

BREAK

3. Project attributes & levels

# Model Relationships Table (example)

|                           |              | <i>Decision Variables</i> |                  |                | <i>Demand</i> | <i>Competitors</i> |                      |          |
|---------------------------|--------------|---------------------------|------------------|----------------|---------------|--------------------|----------------------|----------|
|                           |              | Power Density             | Degradation Rate | Packing Design |               | Aims Solar Panel   | SUAOKI Solar Charger | Units    |
| <i>Product Attributes</i> | Price        | -                         | -                | +              | -             | 225                | 160                  | USD      |
|                           | Weight       | -                         | -                | +              | -             | 2.6                | 2.06                 | kg       |
|                           | Power Output | +                         | +                | +              | +             | 120                | 60                   | W        |
|                           | Durability   | -                         | +                | -              | +             | 60                 | 12                   | Months   |
|                           | Portability  | -                         | -/+              | +              | +             | 20.6"x11"x 1.2"    | 11.5"x7.1"x2.9"      | L"xW"xH" |
|                           | Domain       | [2.5, 60]                 | [24,1000]        | [200, 2800]    |               |                    |                      |          |
|                           | Units        | W/kg                      | Hours            | cm^3           |               |                    |                      |          |

Start defining attribute *levels*

# Defining attribute levels

## Continuous

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

## Discrete

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

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