R Markdown Visualizations

Creating Graphs in R Markdown

by Martin Frigaard

Written: October 03 2022

Updated: December 15 2022

Created using the "λέξις" theme

Materials

The slides are in the `slides.pdf` file

The materials for this training are in the `worksheets` folder:

```
worksheets
├── import.Rmd
├── export.Rmd
├── objects.Rmd
├── rmd-basic.Rmd
├── rmd-tables.Rmd
└── rmd-visualizations.Rmd
```

Outline

1. Importing data
2. Common Data Objects
3. R Markdown
4. R Markdown Data Visualizations
5. R Markdown Tables
6. Exporting Data

R Markdown Data Visualizations

Open `rmd-visualizations.Rmd` to follow along

R Markdown Data Visualizations

The NHANES package comes with data from the 2014 American National Health and Nutrition Examination surveys. We will load a sample from it below:

```
library(NHANES)
SmallNhanes <- NHANES |>
  select(ID, Gender, Age, AgeDecade, Race1, HealthGen,
         Height, BMI, Weight, Pulse, BPSysAve)
```

Quick Tip: Column Names

Standardize names with `janitor::clean_names()`

```
SmallNhanes <- SmallNhanes |> janitor::clean_names()
glimpse(SmallNhanes)
```

```
Rows: 10,000
Columns: 11
$ id         <int> 51624, 51624, 51624, 51625, 51630, 51638, 51646, 51647, 51647, 51647, 51654, 51…
$ gender     <fct> male, male, male, male, female, male, male, female, female, female, male, male,…
$ age        <int> 34, 34, 34, 4, 49, 9, 8, 45, 45, 45, 66, 58, 54, 10, 58, 50, 9, 33, 60, 16, 56,…
$ age_decade <fct>  30-39,  30-39,  30-39,  0-9,  40-49,  0-9,  0-9,  40-49,  40-49,  40-49,  60-6…
$ race1      <fct> White, White, White, Other, White, White, White, White, White, White, White, Wh…
$ health_gen <fct> Good, Good, Good, NA, Good, NA, NA, Vgood, Vgood, Vgood, Vgood, Vgood, Fair, NA…
$ height     <dbl> 164.7, 164.7, 164.7, 105.4, 168.4, 133.1, 130.6, 166.7, 166.7, 166.7, 169.5, 18…
$ bmi        <dbl> 32.22, 32.22, 32.22, 15.30, 30.57, 16.82, 20.64, 27.24, 27.24, 27.24, 23.67, 23…
$ weight     <dbl> 87.4, 87.4, 87.4, 17.0, 86.7, 29.8, 35.2, 75.7, 75.7, 75.7, 68.0, 78.4, 74.7, 3…
$ pulse      <int> 70, 70, 70, NA, 86, 82, 72, 62, 62, 62, 60, 62, 76, 80, 94, 74, 92, 96, 84, 76,…
$ bp_sys_ave <int> 113, 113, 113, NA, 112, 86, 107, 118, 118, 118, 111, 104, 134, 104, 127, 142, 9…
```

Formating factors

We have a `health_gen` variable with the following levels:

Excellent, Vgood, Good, Fair, or Poor. These are ordered.

```
SmallNhanes <- SmallNhanes |>
    mutate(health_gen = factor(x = health_gen,
                               levels = c("Poor", "Fair",
                                          "Good", "Vgood",
                                          "Excellent"),
                                ordered = TRUE))
```

```
levels(SmallNhanes$health_gen)
```

```
[1] "Poor"      "Fair"      "Good"      "Vgood"     "Excellent"
```

ggplot2

The *Layered* grammar of graphics

How it works:

1) Graphs are *initialized* with `ggplot()`

2) Variables are *mapped* to aesthetics

3) Geoms are linked to *statistics*

What relationship do we expect to see between height and weight?

1) Use data with pipe to initialize graph

```
SmallNhanes |>
```

2) Map variables to aesthetics

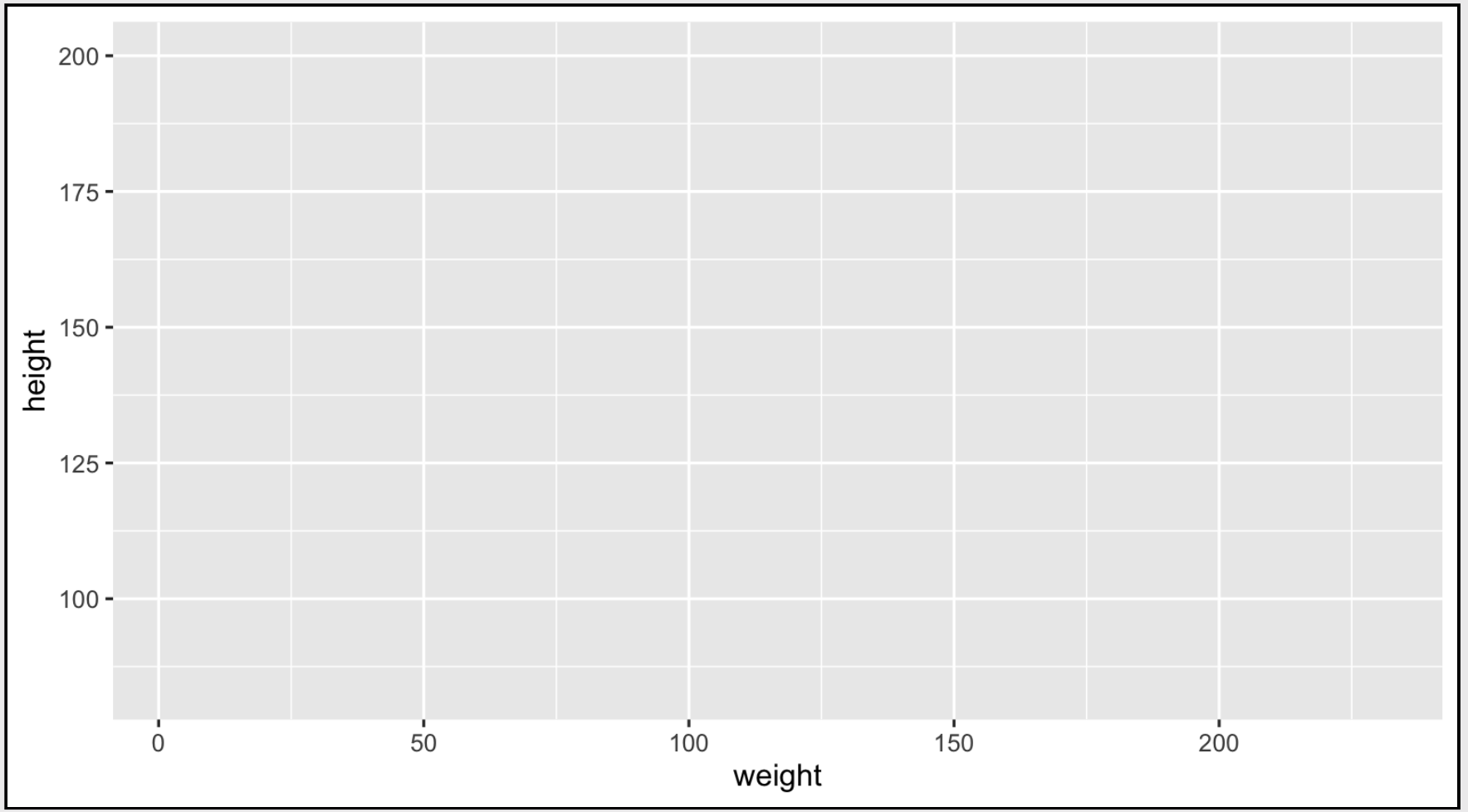
```
SmallNhanes |>
ggplot(mapping = aes(x = weight, y = height))
```

3) Add geoms and layers

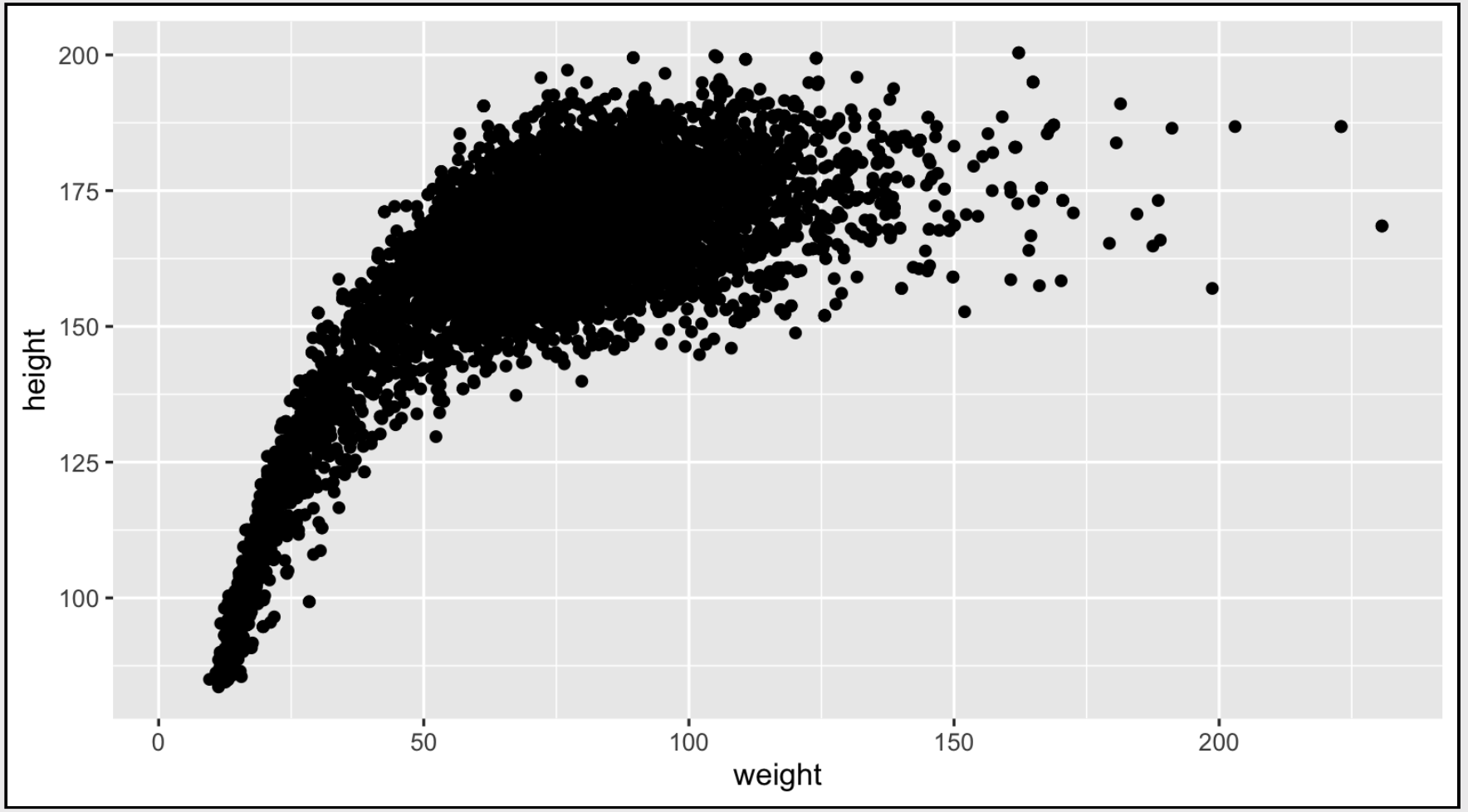
```
SmallNhanes |>
ggplot(mapping = aes(x = weight, y = height)) +
geom_point()
```

```
SmallNhanes %>%
  ggplot() # initialize
```

```
SmallNhanes %>%
  ggplot(mapping = aes(x = weight, y = height)) # map variables
```



```r
SmallNhanes %>%
  ggplot(mapping = aes(x = weight, y = height)) +
  geom_point() # add geoms
```



ggplot2 template

Initialize the plot the `ggplot()`, map the aesthetics, and add a `<GEOM_FUNCTION>`

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

We can add more aesthetics *inside* geoms

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

ggplot2 template

Because ggplot2 is a language of layers, we can continue adding *more* geoms

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

Note the different syntax (%>% vs. +)

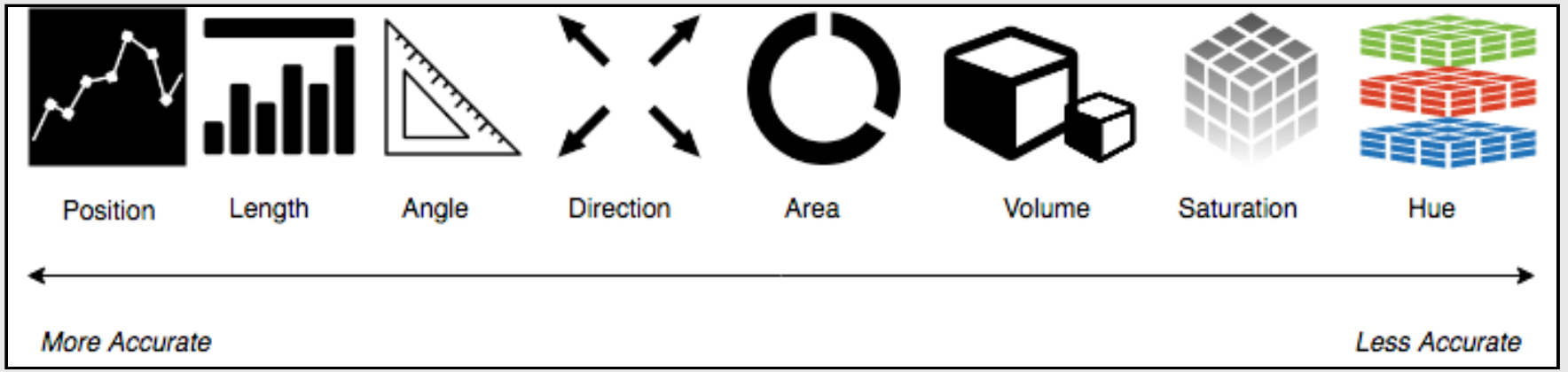
```
<DATA> %>% #<< pipe!
  ggplot(mapping = aes(<MAPPINGS>)) + #<< plus!
  <GEOM_FUNCTION>(mapping = aes(<MAPPINGS>))
```

Aesthetics

Is the relationship between `weight` and `height` the same for both `gender`s?

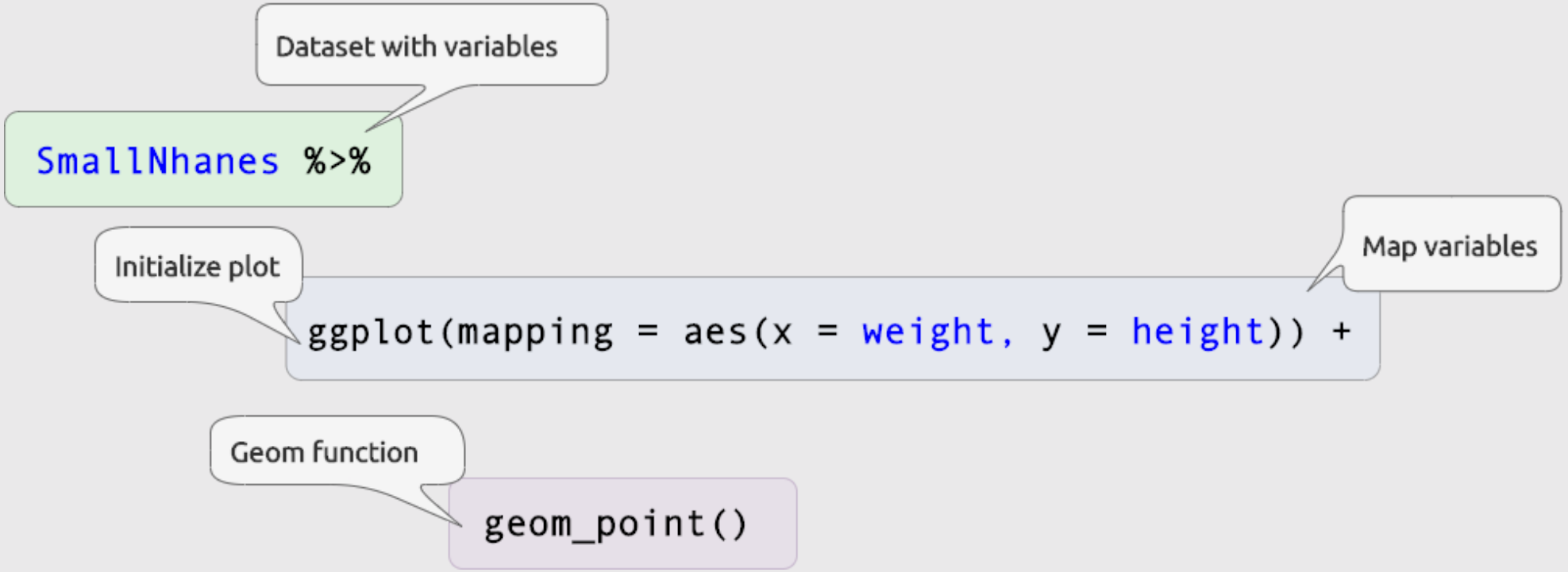
We can explore this by mapping the variables to different aesthetics

Aesthetics as graph elements (`color`, `size`, `shape`, and `alpha`)



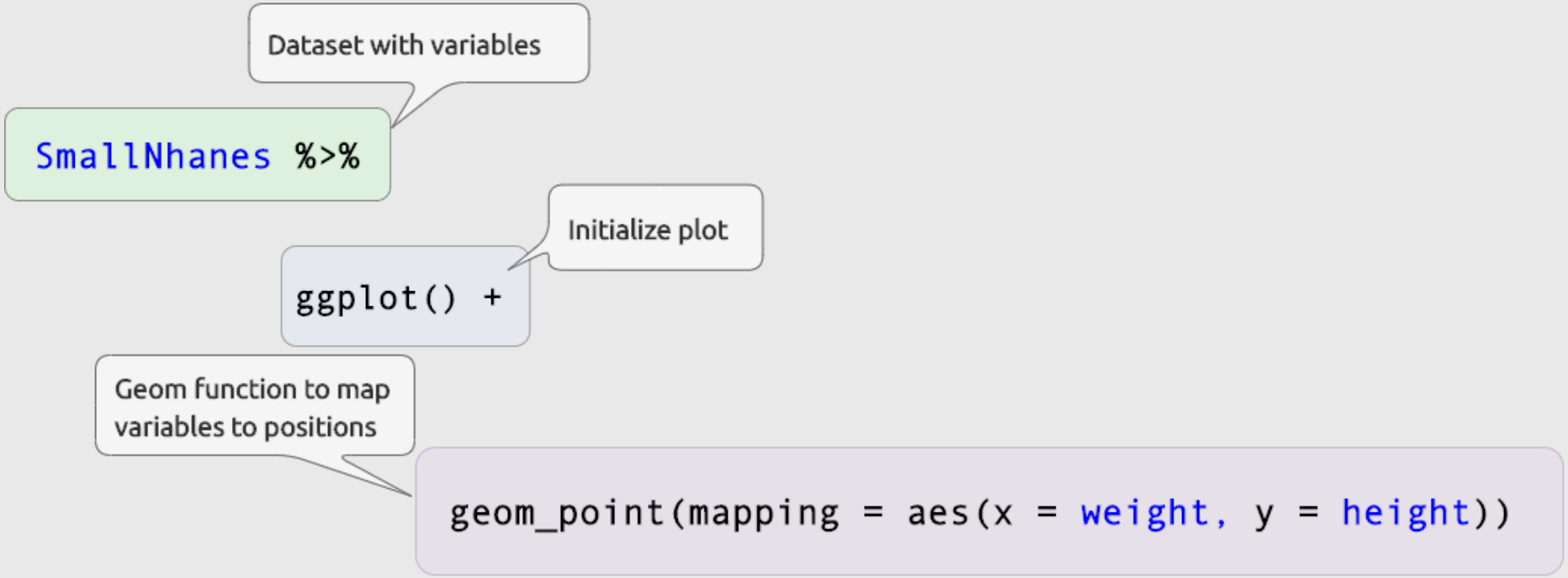
Global `ggplot2` mapping

inside the `ggplot()` function = setting variables ***globally***



Local `ggplot2` mapping

inside the `geom()` function = setting variables ***locally***



Your Turn

Set local vs. global aesthetic mappings

From here...

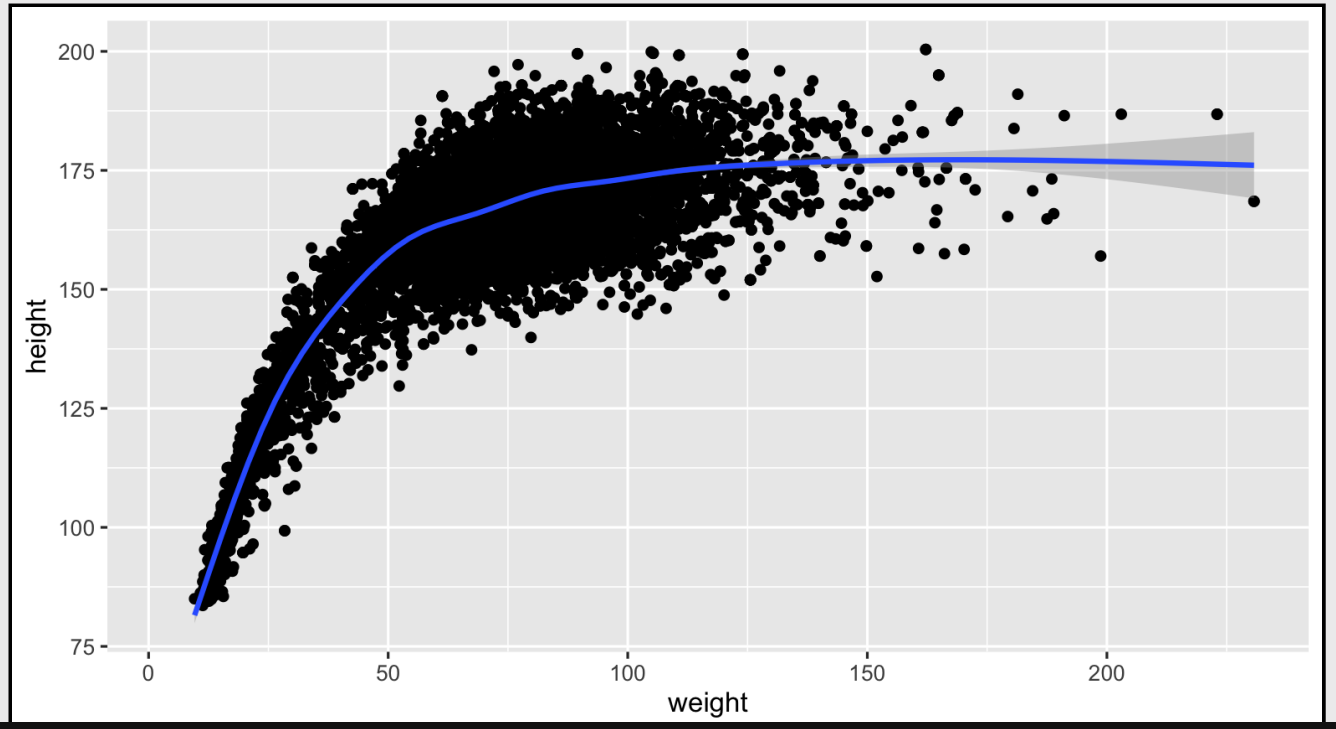
```
SmallNhanes %>%
  ggplot(
      mapping =
          aes(x = weight, y = height)) +
  geom_point() +
  geom_smooth()
```

...to here.

```
SmallNhanes %>%
  ggplot() +
  geom_point(
      mapping =
          aes(x = weight, y = height)) +
  geom_smooth(
      mapping =
          aes(x = weight, y = height))
```

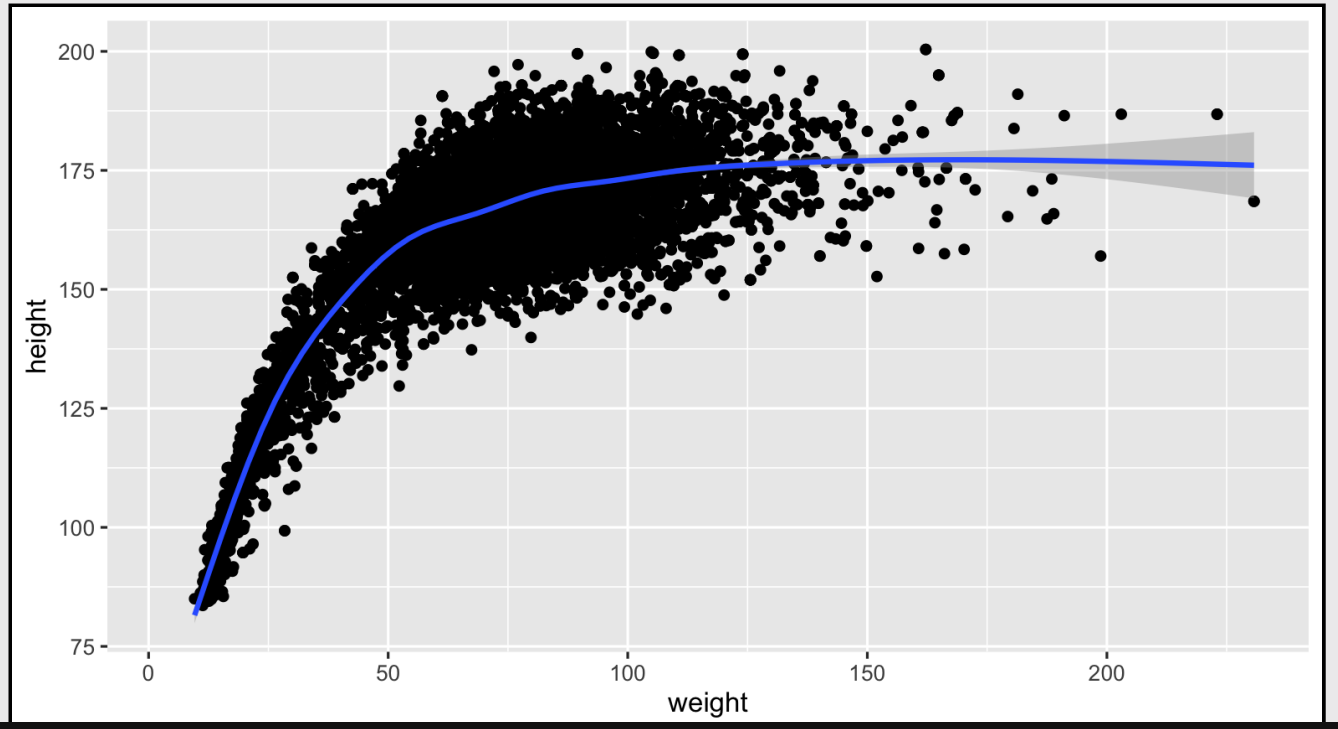
Your Turn (solution 1)

```
SmallNhanes %>%
  ggplot(mapping = aes(x = weight, y = height)) +
  geom_point() +
  geom_smooth()
```



Your Turn (solution 2)

```
SmallNhanes %>%
  ggplot() +
  geom_point(mapping = aes(x = weight, y = height)) +
  geom_smooth(mapping = aes(x = weight, y = height))
```



Variables, Aestheitcs, and Geoms

Variables, Aestheitcs, and Geoms (1)

Each graph needs a variable or value, an aesthetic, and geom (the accompanying graphic, geometry)

```
geom_point(mapping = aes(x = weight, y = height)) + # layer 1
geom_smooth(mapping = aes(x = weight, y = height)) # layer 2
```

variable	aesthetic	geom
`weight`	position = `x`	dots = `point`
`height`	position = `y`	dots = `point`
`weight`	position = `x`	line = `smooth`
`height`	position = `y`	line = `smooth`

These have the same aesthetics! What if we added a layer with a variable mapped to a different aesthetic?

Variables, Aestheitcs, and Geoms (2)

But we can add *more* variables, map them to *different* aesthetics, and *adding* another `geom` layer

Add another layer, coloring the points by `gender`

```
SmallNhanes %>%
  ggplot() +
  geom_point(mapping = aes(x = weight, y = height)) +
  geom_point(mapping = aes(color = gender))
```

variable	aesthetic	geom
`weight`	position = `x`	dots = `point`
`height`	position = `y`	dots = `point`
`gender`	color = `color`	dots = `point`

Variables, Aestheitcs, and Geoms (3)

ERROR!

```
SmallNhanes %>%
  ggplot() +
  geom_point(
      aes(x = weight, y = height)) +
  geom_point(
      aes(color = gender))
```

```
# Error: geom_point requires the following
missing aesthetics: x and y
```

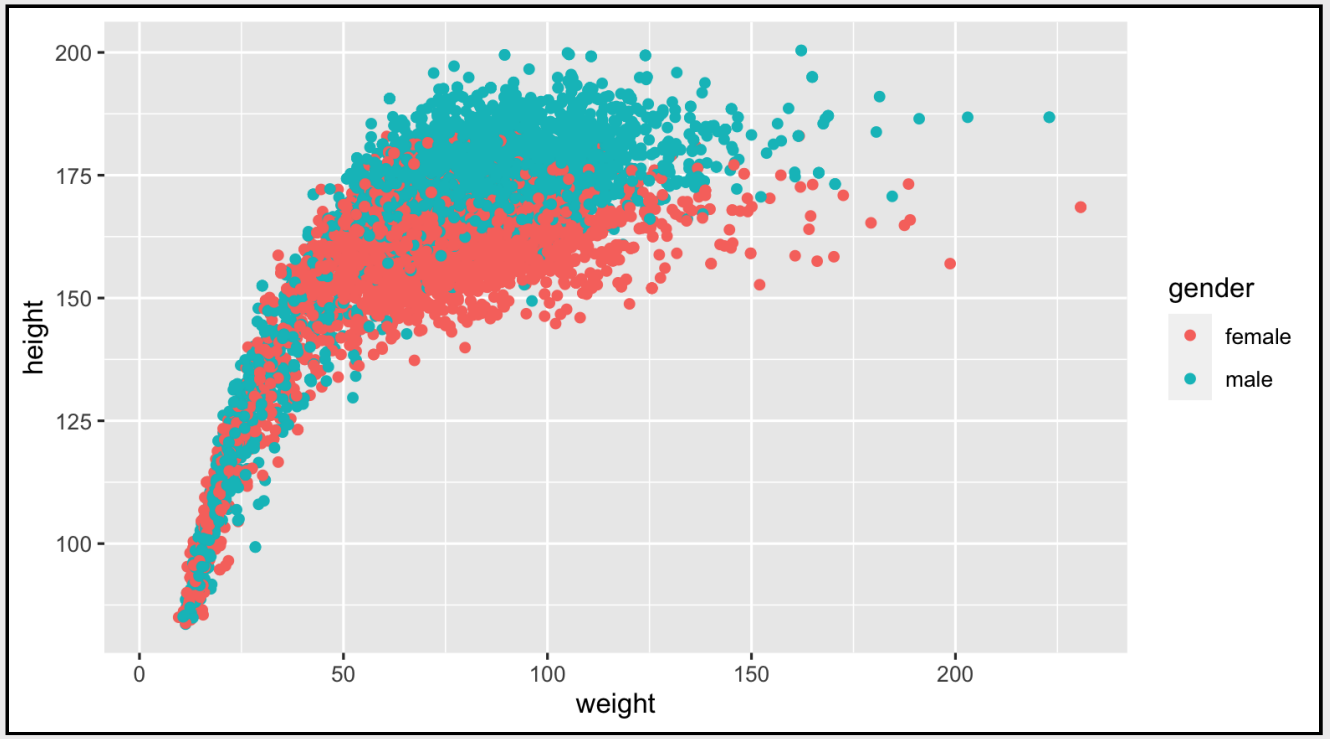
SOLUTION

All geoms have required aesthetics--map variables globally

```
SmallNhanes %>%
  ggplot(
    aes(x = weight, y = height)) +
    geom_point(aes(color = gender))
```

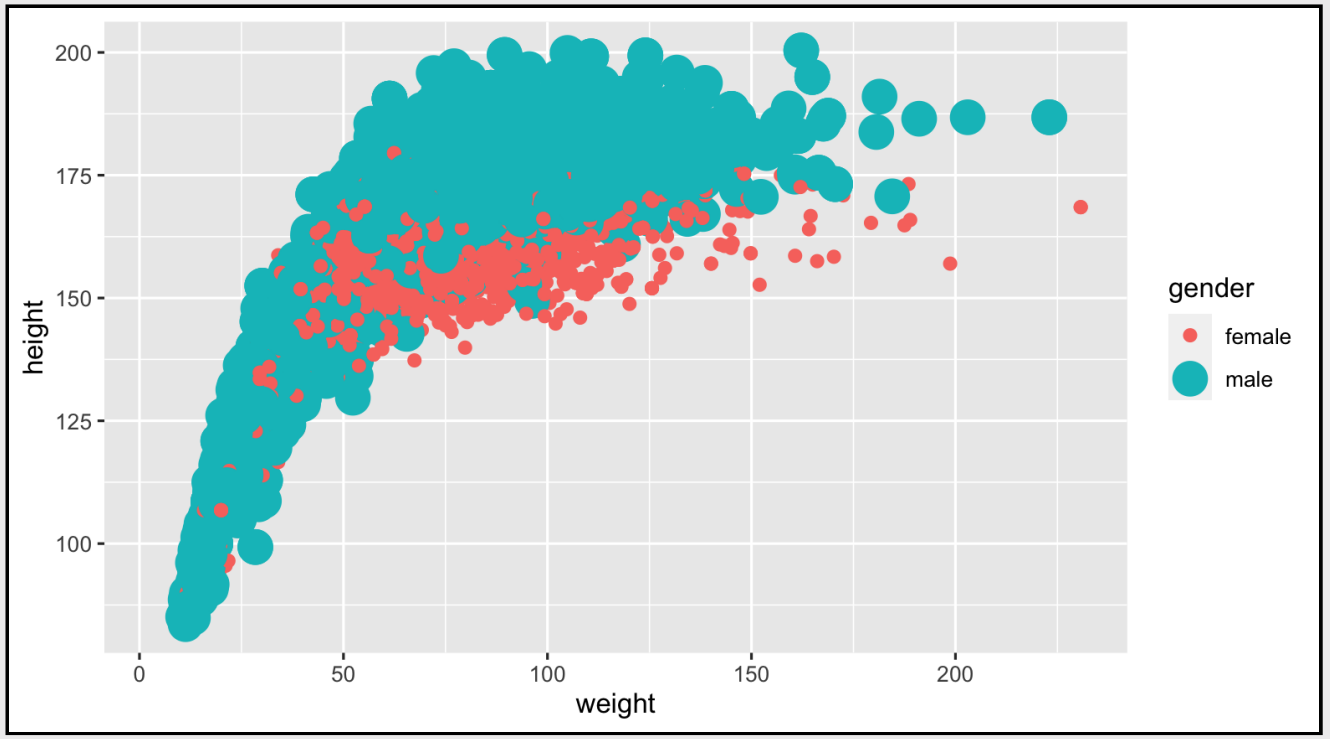
Aesthetics: color

```
SmallNhanes %>%
  ggplot(aes(x = weight, y = height)) +
    geom_point(aes(color = gender))
```



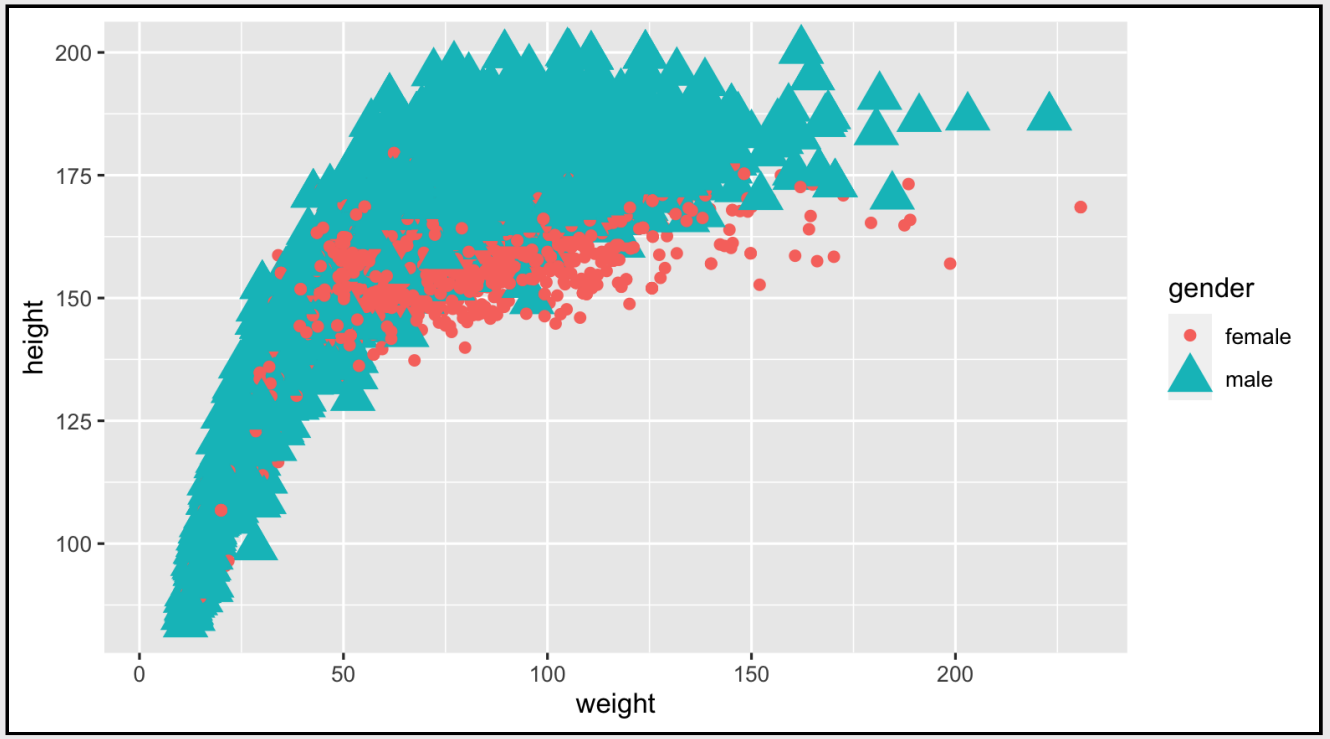
Aesthetics: size

```
SmallNhanes %>%
  ggplot(aes(x = weight, y = height)) +
    geom_point(aes(color = gender, size = gender))
```



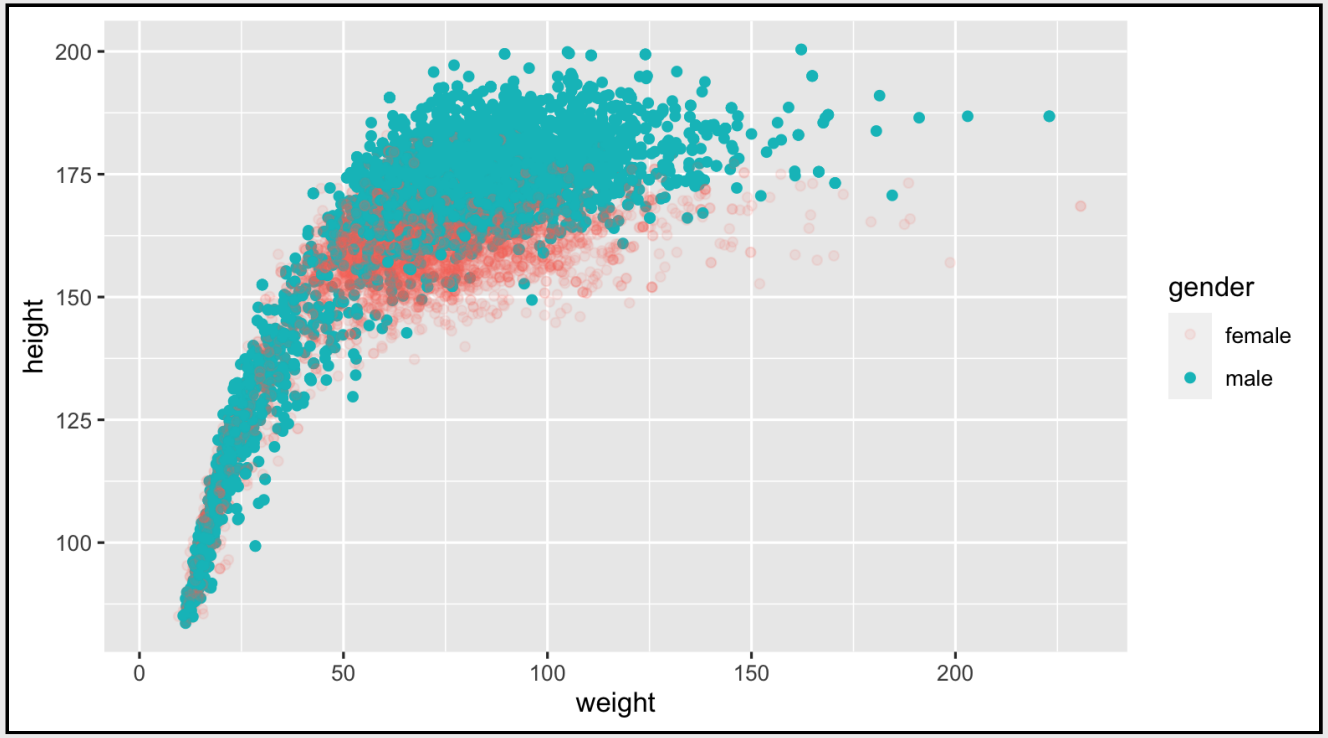
Aesthetics: shape

```
SmallNhanes %>%
  ggplot(aes(x = weight, y = height)) +
    geom_point(aes(color = gender, size = gender, shape = gender))
```



Aesthetics: alpha (opacity)

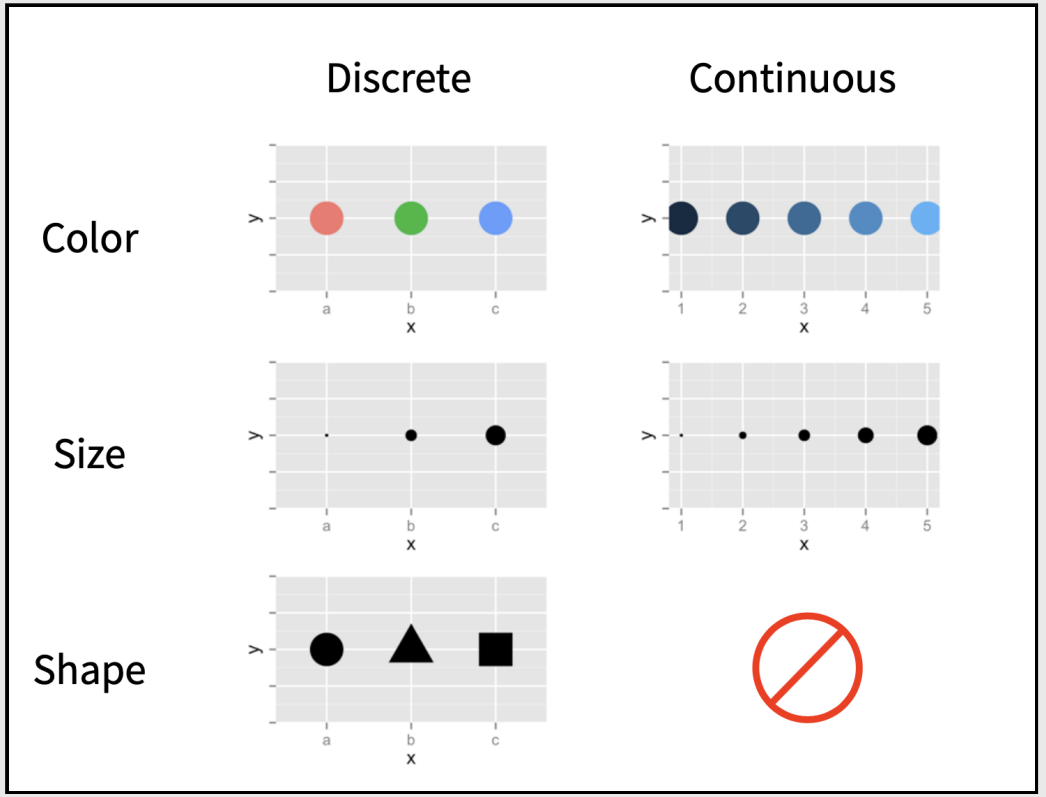
```
SmallNhanes %>%
  ggplot(aes(x = weight, y = height)) +
    geom_point(aes(color = gender, alpha = gender))
```



Aesthetic mappings

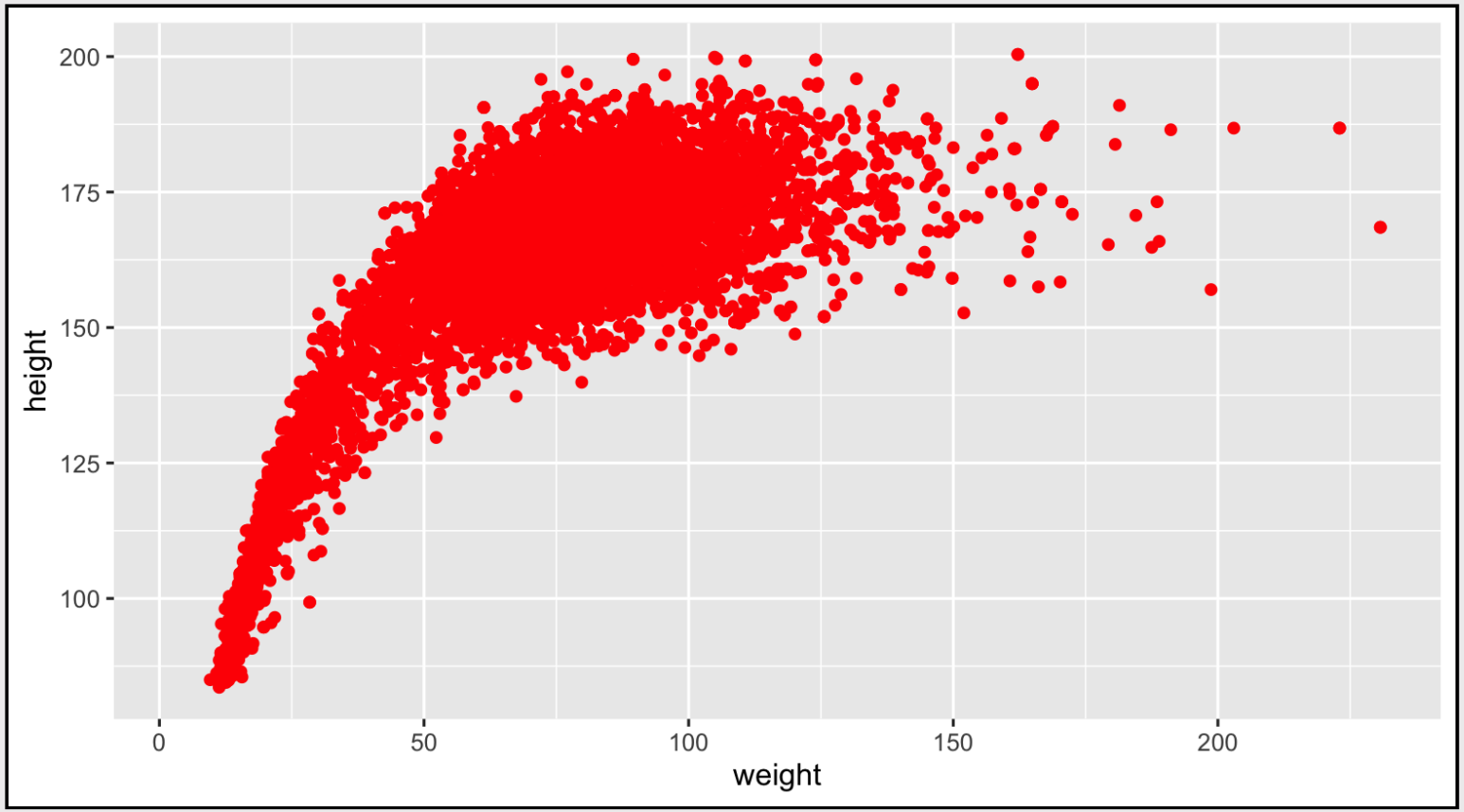
Legend is automatically included

Continuous variables best with `size`



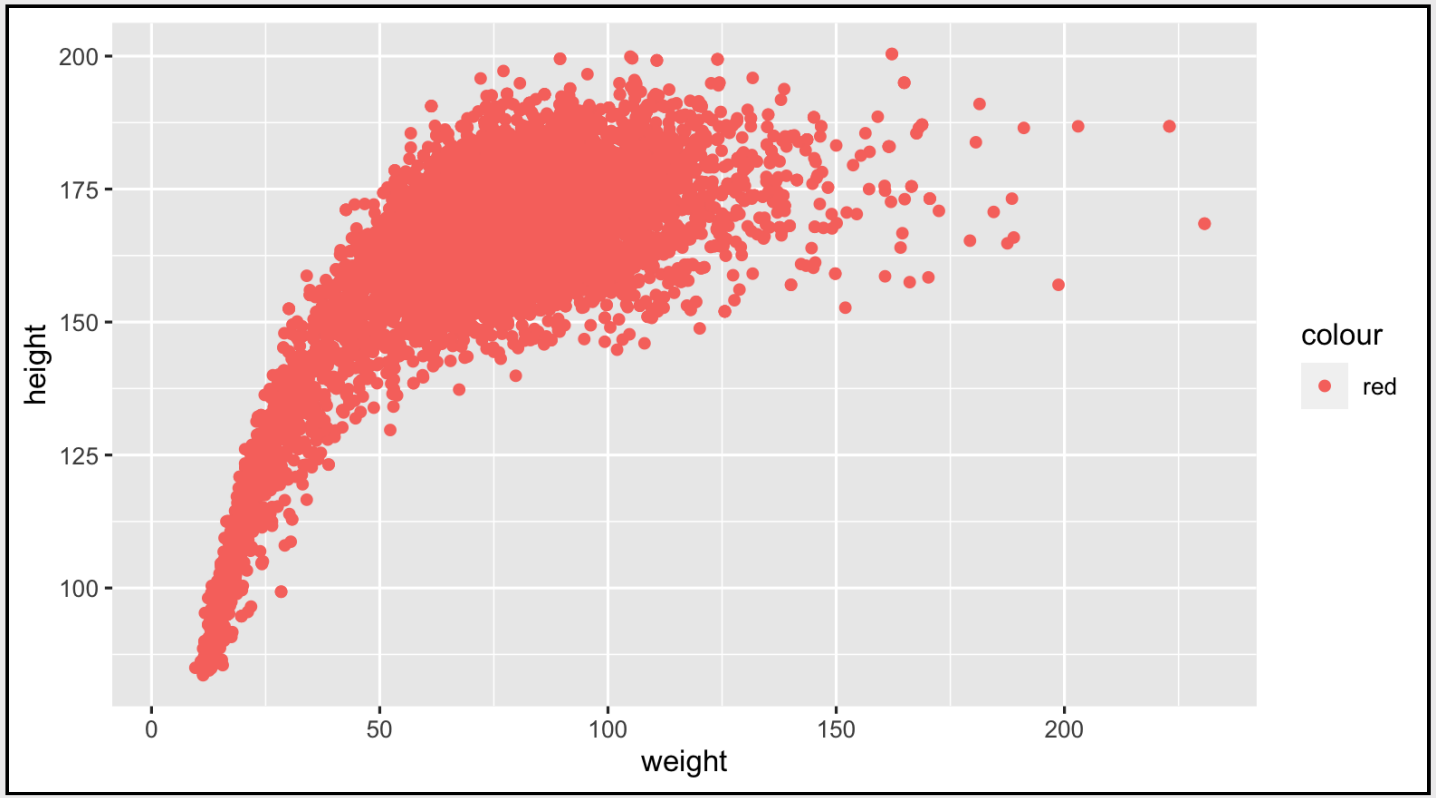
Setting values vs. mapping variables

How can we create this plot?



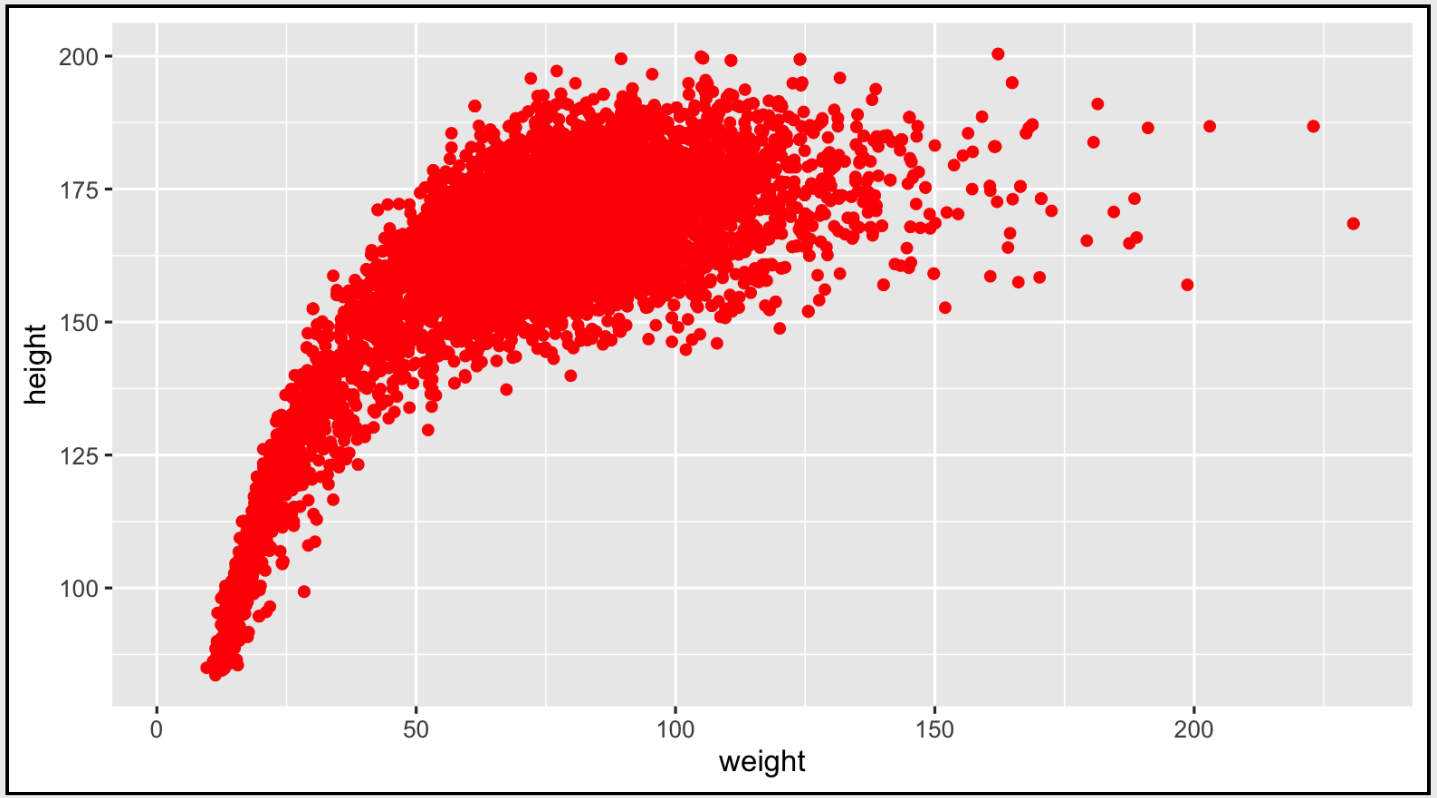
Inside `aes()`

```
SmallNhanes %>%
  ggplot(aes(x = weight, y = height)) +
    geom_point(aes(color = "red")) # inside aes
```



Outside `aes()`

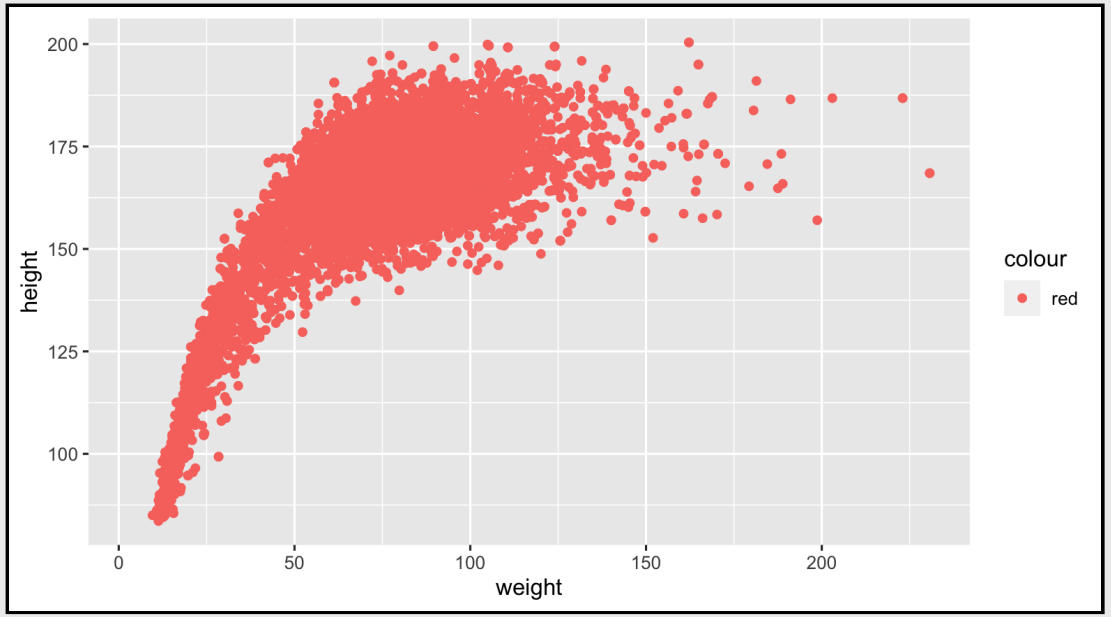
```
SmallNhanes %>%
  ggplot(aes(x = weight, y = height)) +
  geom_point(color = "red") # outside aes
```



What happened?

`aes()` expected a variable, not a value (`"red"`).

```
SmallNhanes %>%
  ggplot(aes(x = weight, y = height)) +
  geom_point(aes(color = "red")) # "value" in aes
```



Geoms (geometric objects)

Geoms

These are visual elements used to represent the data of the graph

Examples include:

- `geom_boxplot`
- `geom_col`
- `geom_line`
- `geom_smooth`

See the cheatsheet for more examples:

https://bit.ly/ggplot2-cheat

Your Turn

How does BMI vary across levels of self-reported general health?

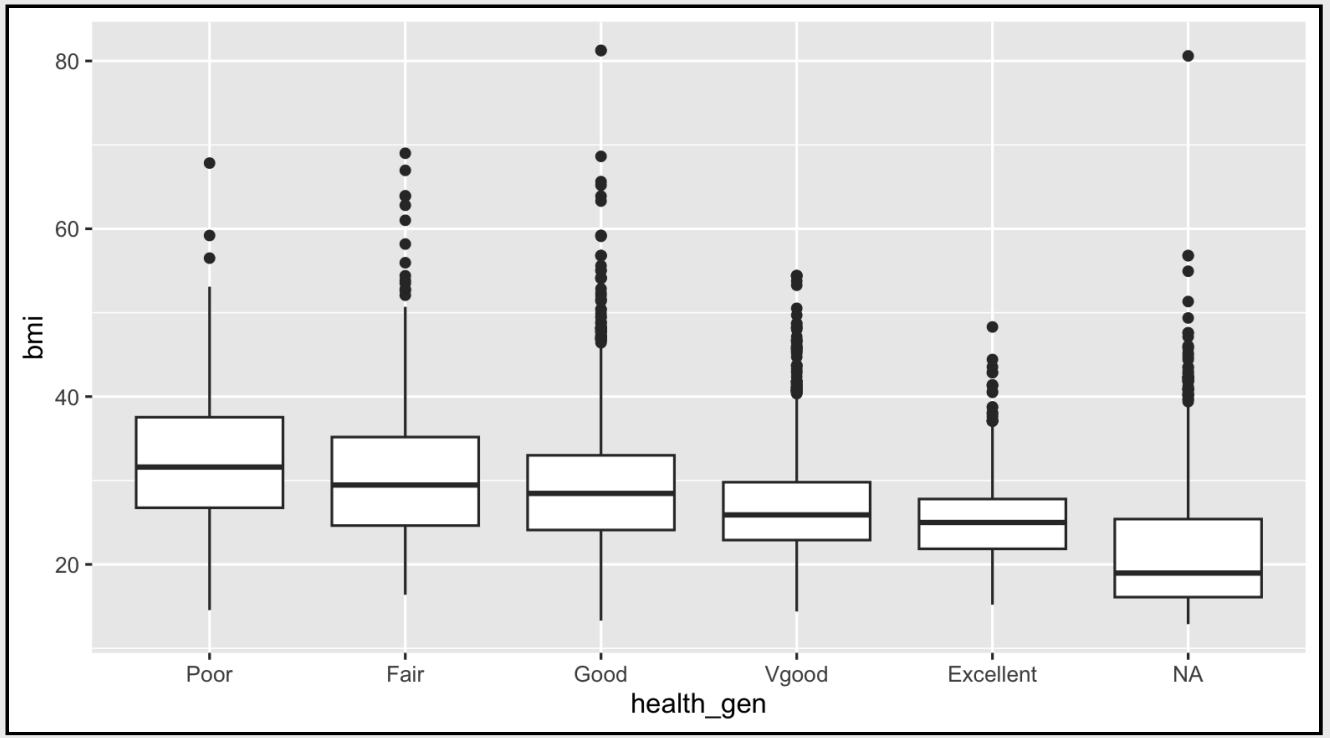
Complete the code below:

Map the variables locally inside the `geom_boxplot()` function

```
SmallNhanes %>%
  ggplot() %>%
  geom_boxplot(mapping = aes(x = ___________, y = ___))
```

```
SmallNhanes %>%
  ggplot() +
  geom_boxplot(mapping = aes(x = health_gen, y = bmi))
```

Box-plots are great for seeing how a continuous variable varies across a categorical variable



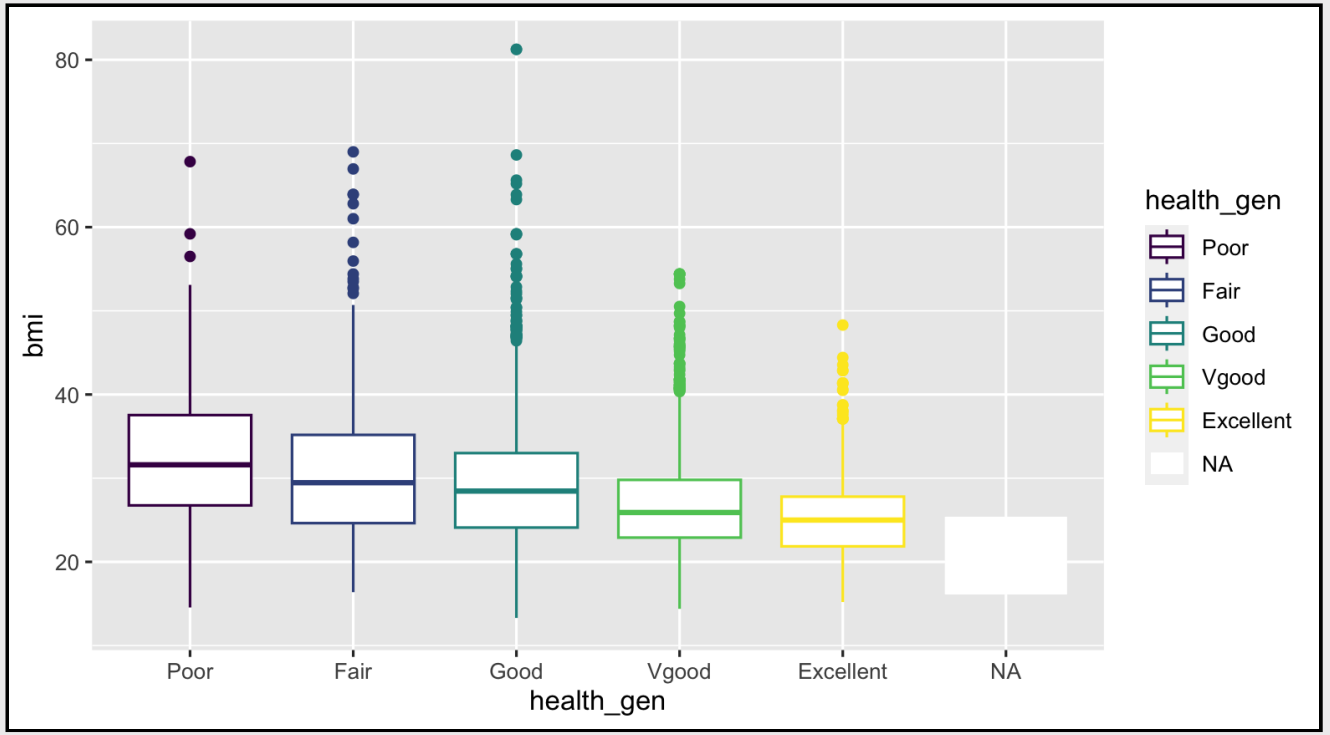
Your Turn

Fill in the code below to change the colors in the boxplot for each level of `health_gen`

```
SmallNhanes %>%
  ggplot() +
  geom_boxplot(
    aes(x = health_gen, y = bmi, _____ = health_gen))
```

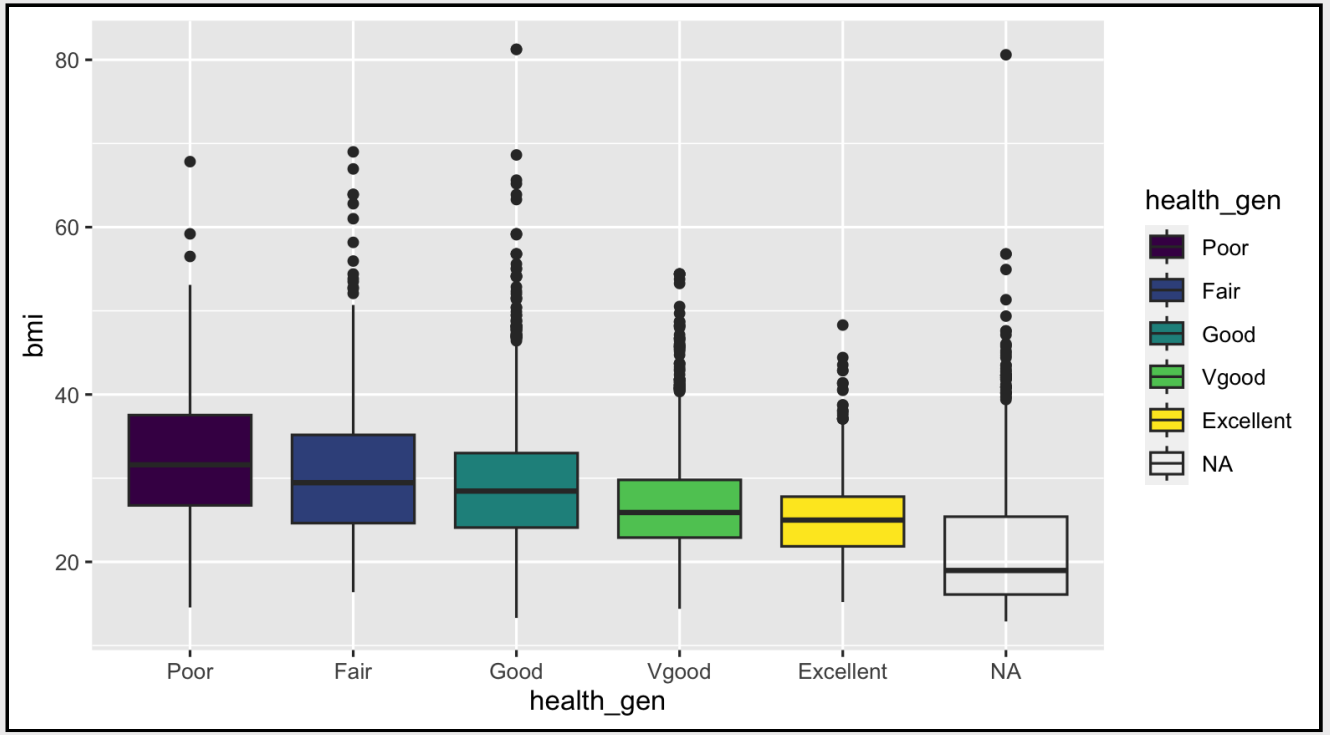
```
SmallNhanes %>%
  ggplot() +
  geom_boxplot(
    aes(x = health_gen, y = bmi, color = health_gen))
```

Color is not the setting we want here...



```
SmallNhanes %>%
  ggplot() +
  geom_boxplot(
    aes(x = health_gen, y = bmi, fill = health_gen))
```

Fill is better

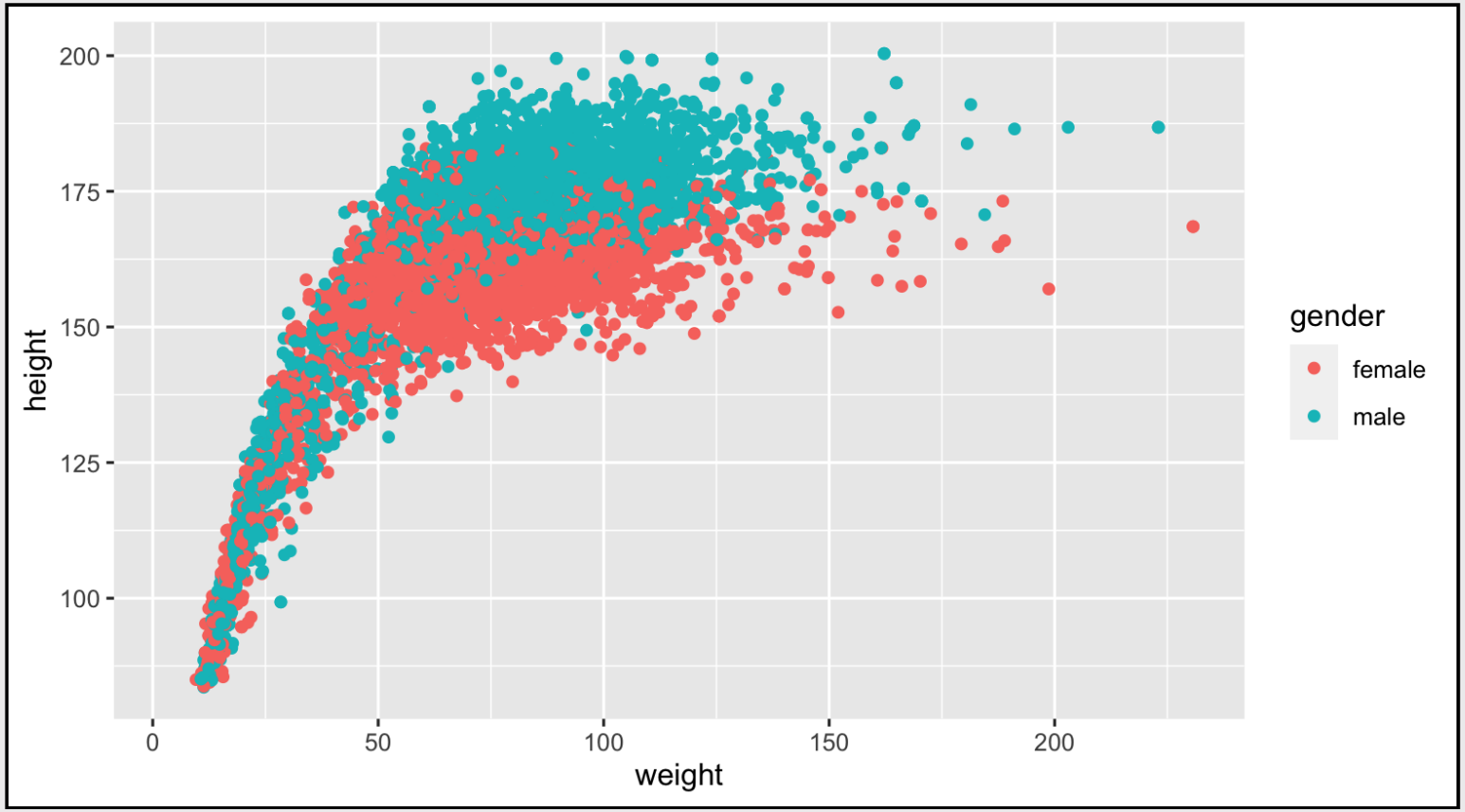


Adding layers

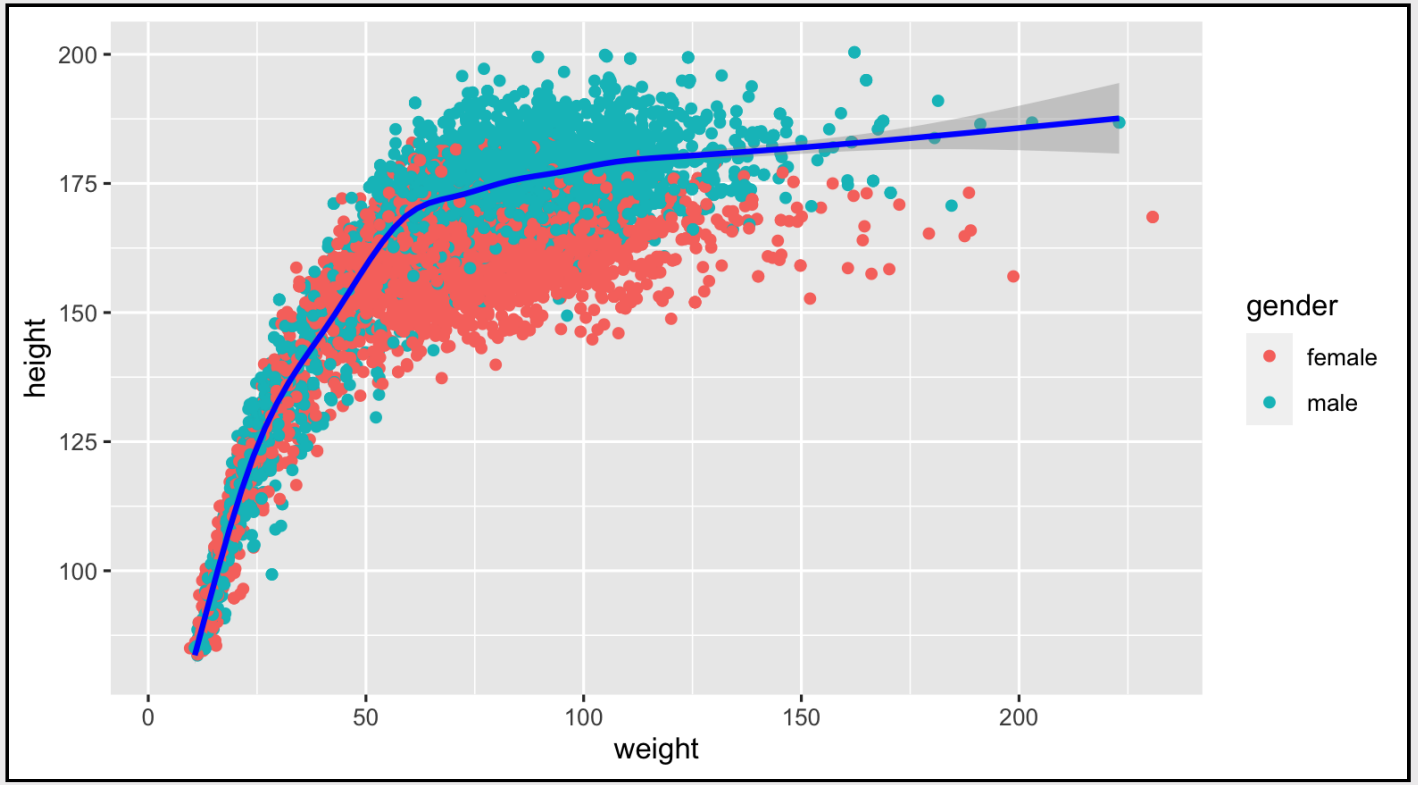
The 'infinitely extensible' part of `ggplot2` is where we start to really see it's power

Consider the relationship between `height` and `weight` again

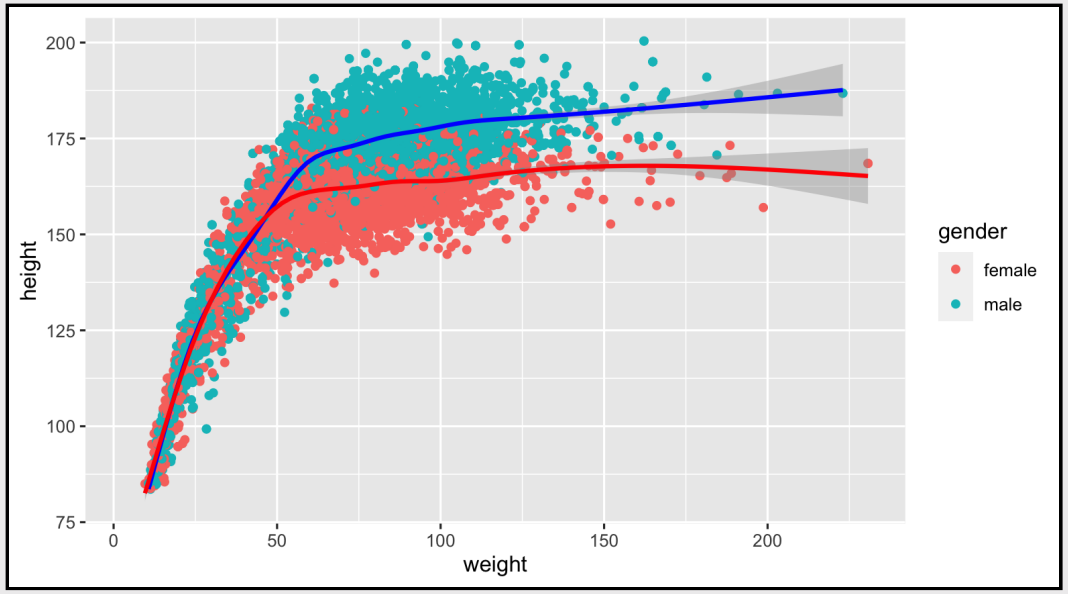
```
SmallNhanes %>%
  ggplot(aes(x = weight, y = height)) + # global
  geom_point(aes(color = gender))
```



```r
SmallNhanes %>%
  ggplot(aes(x = weight, y = height)) +
  geom_point(aes(color = gender)) +
  geom_smooth(data = # data 2
                filter(SmallNhanes, gender == "male"), # layer 2
              aes(x = weight, y = height),
              color = "blue")
```



```r
SmallNhanes %>%
  ggplot(aes(x = weight, y = height)) +
  geom_point(aes(color = gender)) +
  geom_smooth(data =
                filter(SmallNhanes, gender == "male"),
              aes(x = weight, y = height),
              color = "blue") +
    geom_smooth(data = # data 3
                  filter(SmallNhanes, gender == "female"), # layer 3
              aes(x = weight, y = height),
              color = "red")
```



Facets

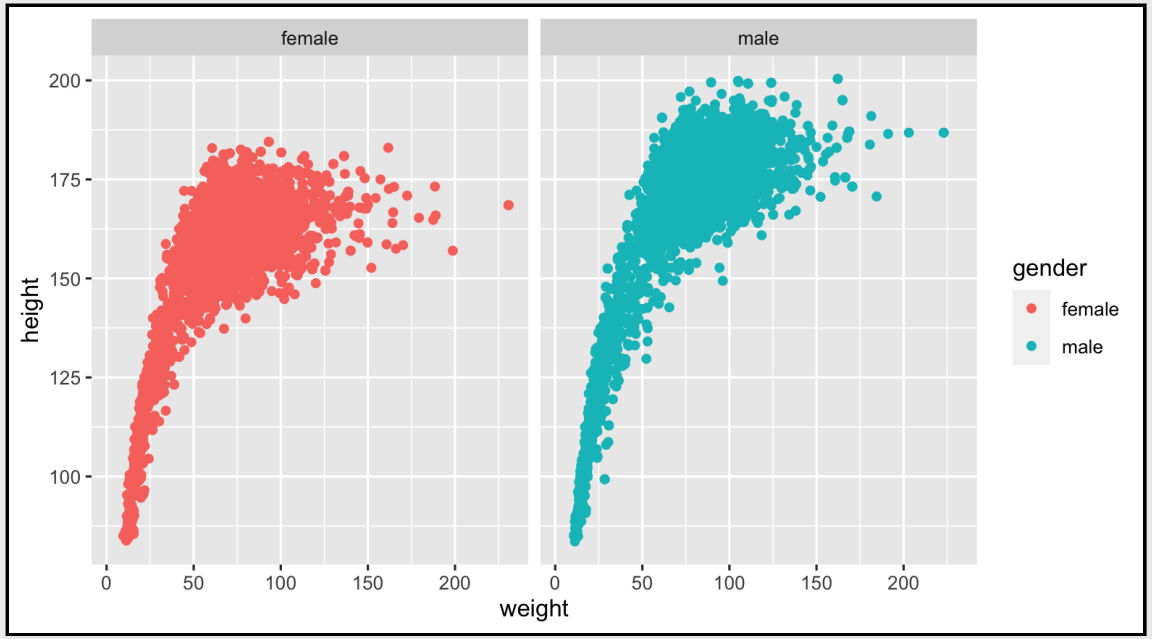
Faceting

Facet layers display subplots for levels of categorical variables

Facet layer	Display
`facet_wrap(. ~ gender)`	Plot for each level of `gender`
`facet_wrap(race1 ~ gender)`	Plot for each level of `gender` and `race`
`facet_wrap(. ~ gender, ncol = 1)`	Specify the number of columns
`facet_wrap(. ~ gender, nrow = 1)`	Specify the number of rows

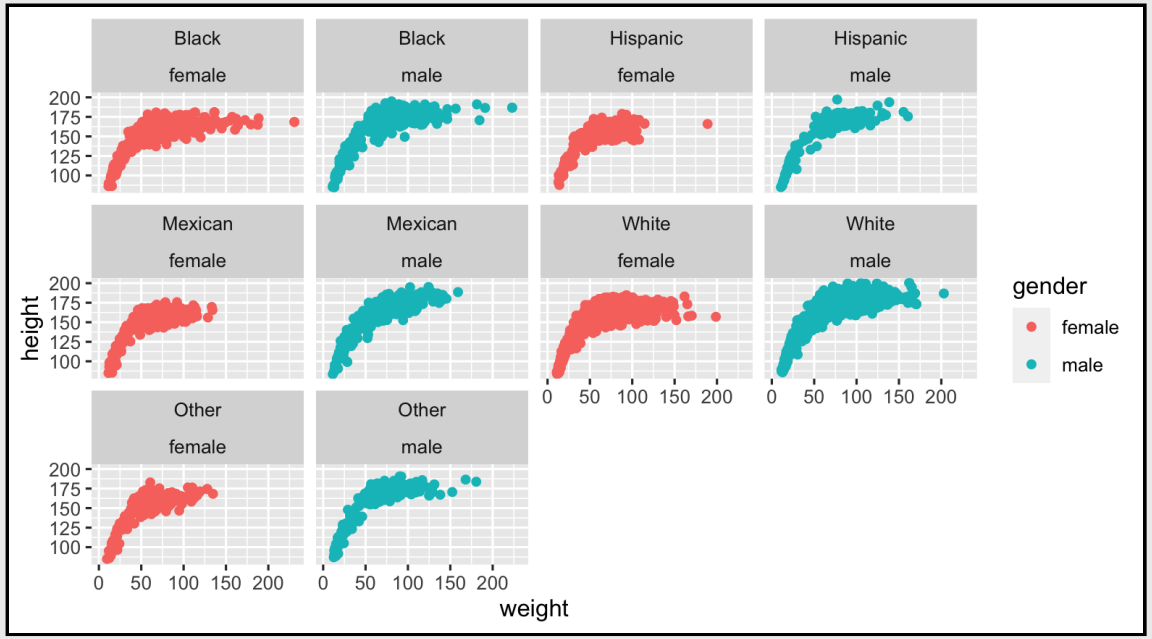
Facet Single Variable

```
SmallNhanes %>%
  ggplot(aes(x = weight, y = height)) +
  geom_point(aes(color = gender)) +
  facet_wrap(. ~ gender)
```



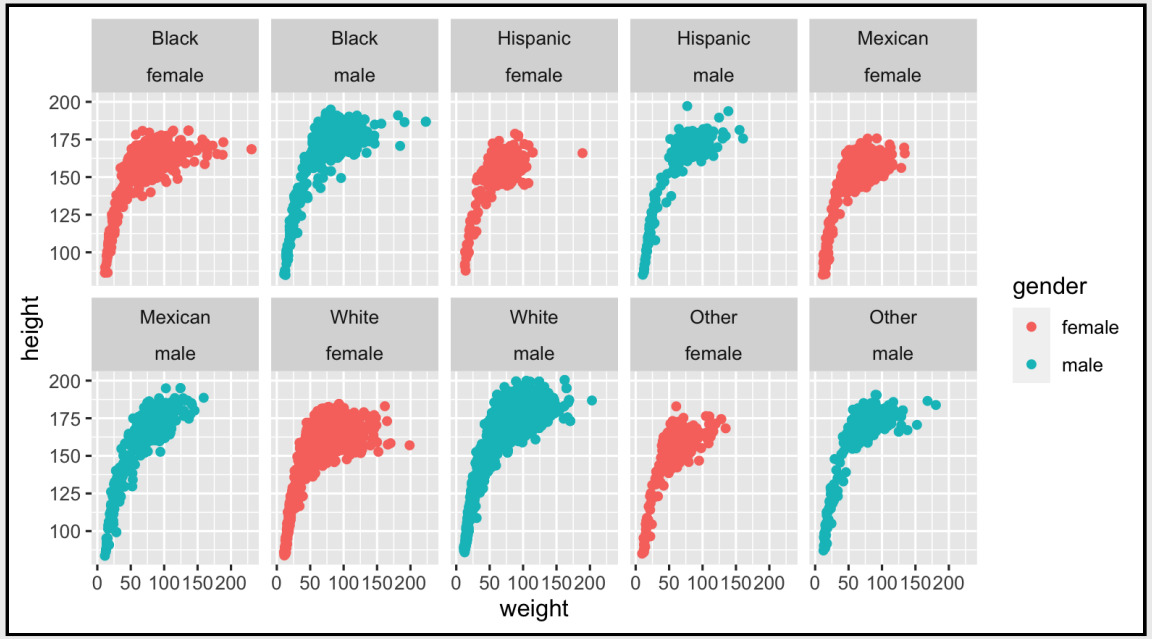
Facet Two Variables

```
SmallNhanes %>%
  ggplot(aes(x = weight, y = height)) +
  geom_point(aes(color = gender)) +
  facet_wrap(race1 ~ gender)
```



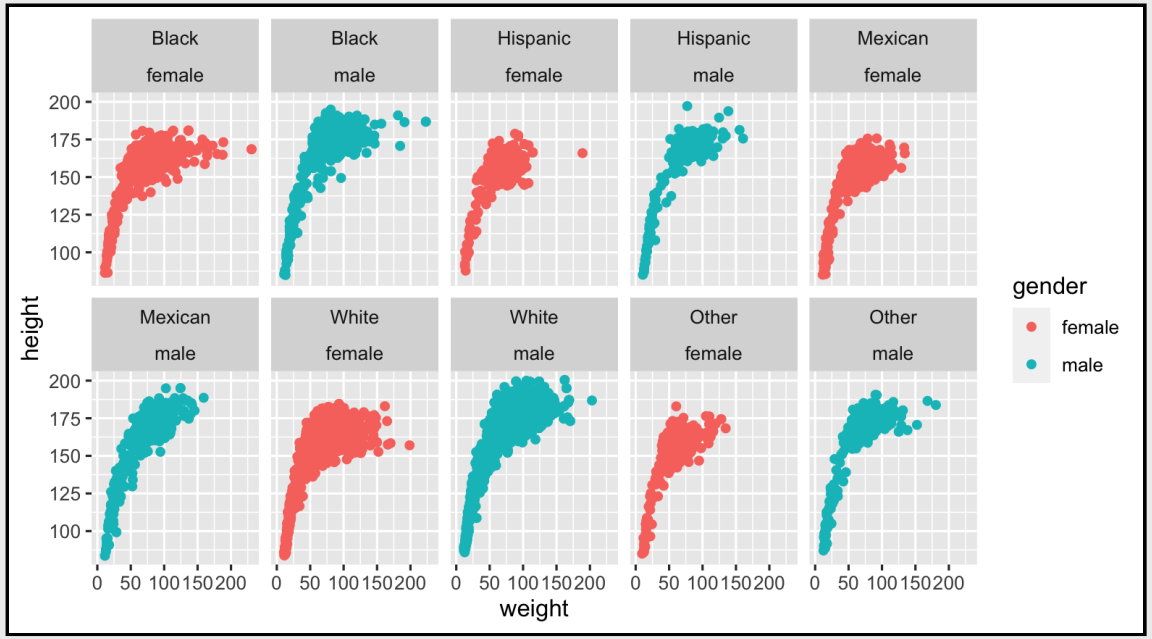
Facet: Set Columns

```
SmallNhanes %>%
  ggplot(aes(x = weight, y = height)) +
  geom_point(aes(color = gender)) +
  facet_wrap(race1 ~ gender, ncol = 5)
```



Facet: Set Rows

```
SmallNhanes %>%
  ggplot(aes(x = weight, y = height)) +
  geom_point(aes(color = gender)) +
  facet_wrap(race1 ~ gender, nrow = 2)
```



Recap

1) Introduction the grammar of graphics syntax

2) Identifying graph aesthetics (position, color, shape, opacity, etc.)

3) Recognizing and using `geoms` (`geom_point`, `geom_smooth`, etc.)

4) Facetting graphs (`facet_wrap` with 1 or two variables)