Data and visualization

Prof. Maria Tackett

Click for PDF of slides

Exploratory data analysis

What is EDA?

- **Exploratory data analysis (EDA)** is an approach to analyzing data sets to summarize the main characteristics.

- Often, EDA is visual. That's what we're focusing on today.

- We can also calculate summary statistics and perform data wrangling/manipulation/transformation at (or before) this stage of the analysis.

Data visualization

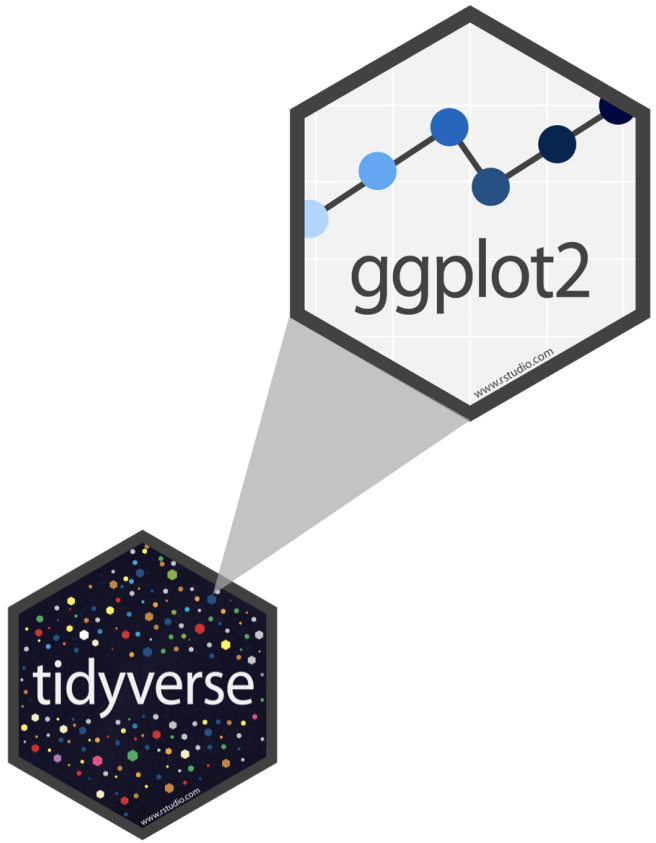
Data visualization

> *"The simple graph has brought more information to the data analyst's mind than any other device." — John Tukey*

- Data visualization is the creation and study of the visual representation of data.

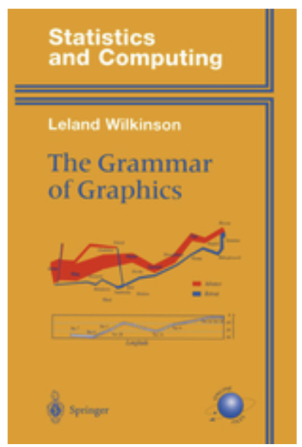
- There are many tools for visualizing data (R is one of them), and many approaches/systems within R for making data visualizations
  - We'll use **`ggplot2`**.

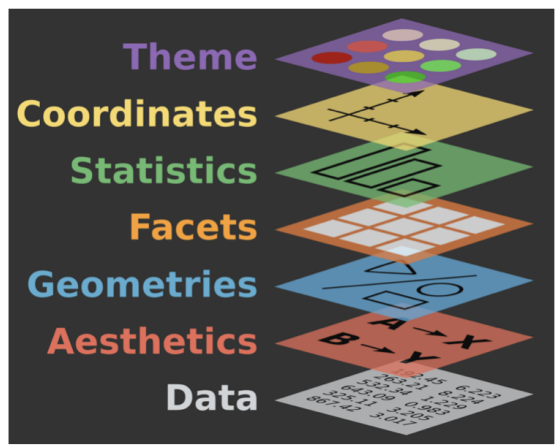
ggplot2 in tidyverse



Source: BloggoType

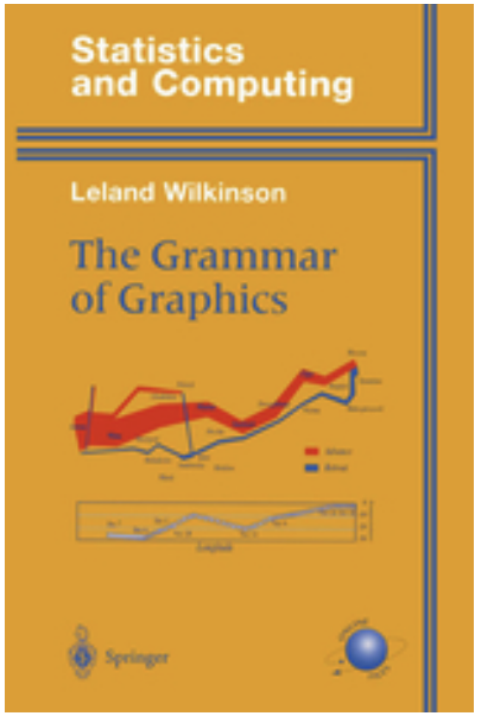
- **ggplot2** is tidyverse's data visualization package
- The **gg** in "ggplot2" stands for Grammar of Graphics
- It is inspired by the book **Grammar of Graphics** by Leland Wilkinson*

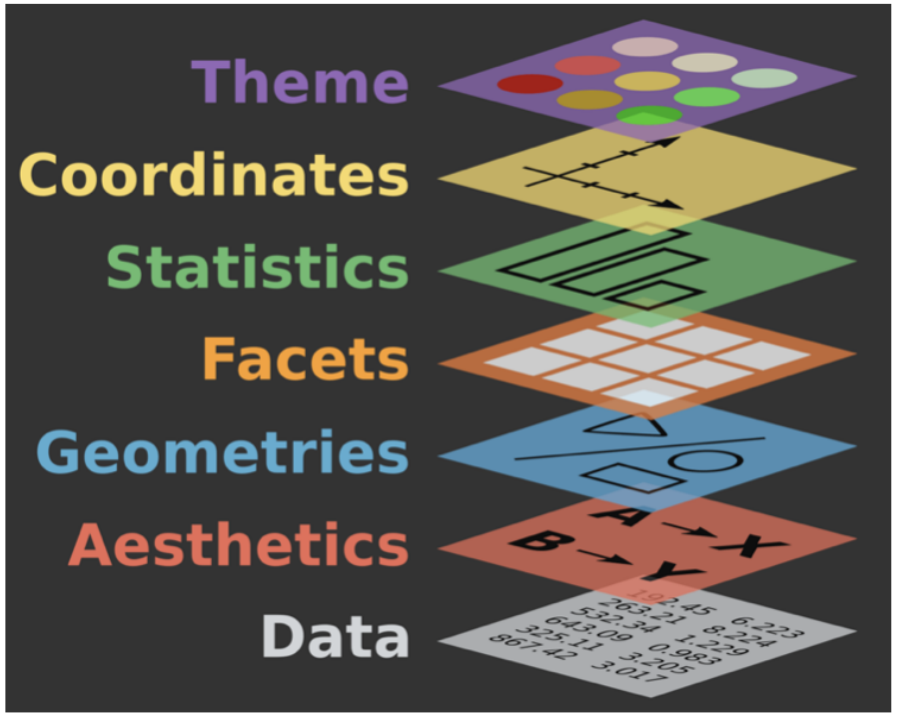




What is a Grammar of Graphics?

A tool that allows for concisely describing the components of a graphic:

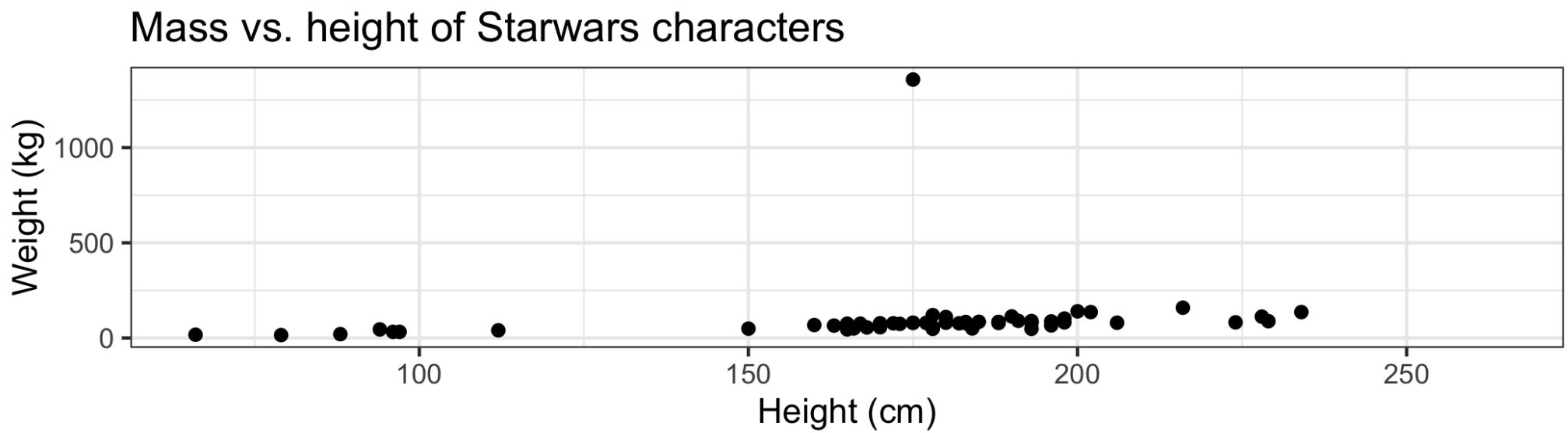




What function is doing the plotting?

```
ggplot(data = starwars, mapping = aes(x = height, y = mass)) +
  geom_point() +
  labs(title = "Mass vs. height of Starwars characters",
       x = "Height (cm)", y = "Weight (kg)")
```

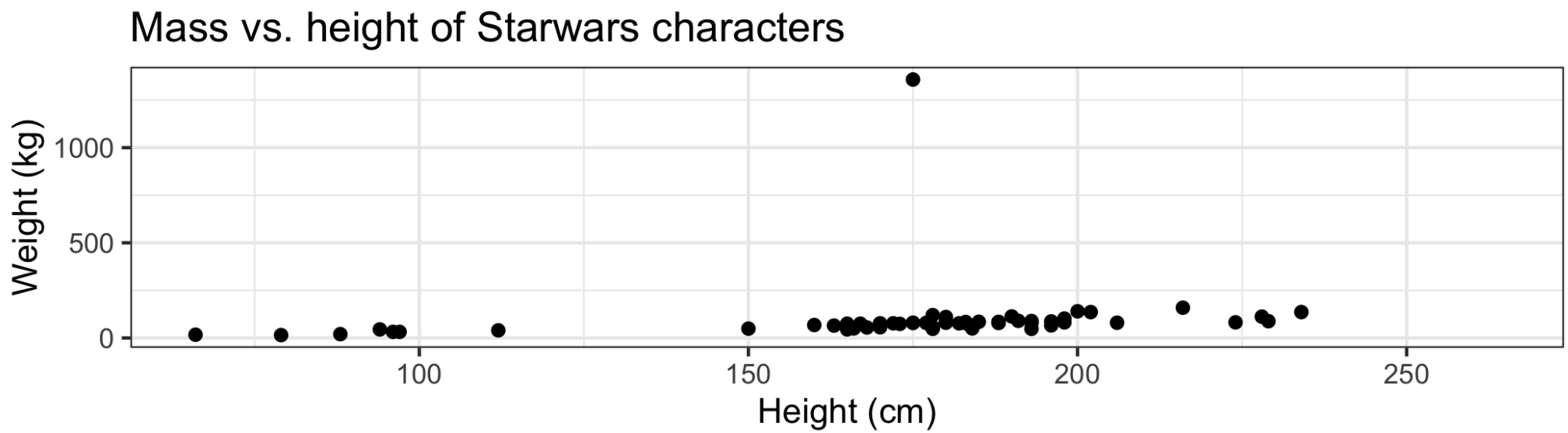
```
## Warning: Removed 28 rows containing missing values (geom_point).
```



What is the dataset being plotted?

```
ggplot(data = starwars, mapping = aes(x = height, y = mass)) +
  geom_point() +
  labs(title = "Mass vs. height of Starwars characters",
       x = "Height (cm)", y = "Weight (kg)")
```

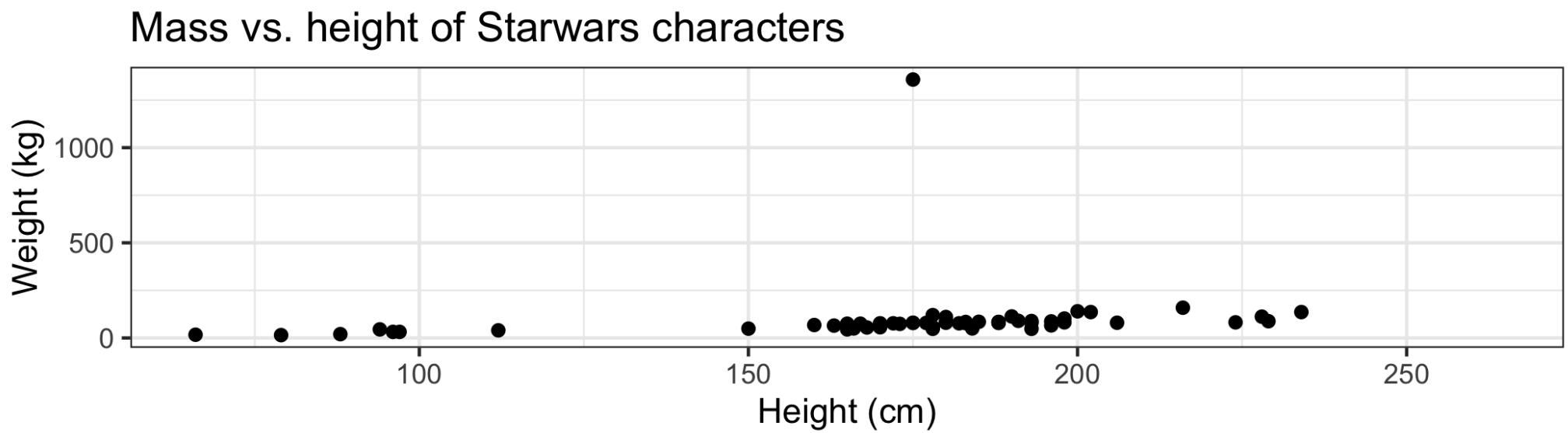
```
## Warning: Removed 28 rows containing missing values (geom_point).
```



Which variable is on the x-axis? On the y-axis?

```
ggplot(data = starwars, mapping = aes(x = height, y = mass)) +
  geom_point() +
  labs(title = "Mass vs. height of Starwars characters",
       x = "Height (cm)", y = "Weight (kg)")
```

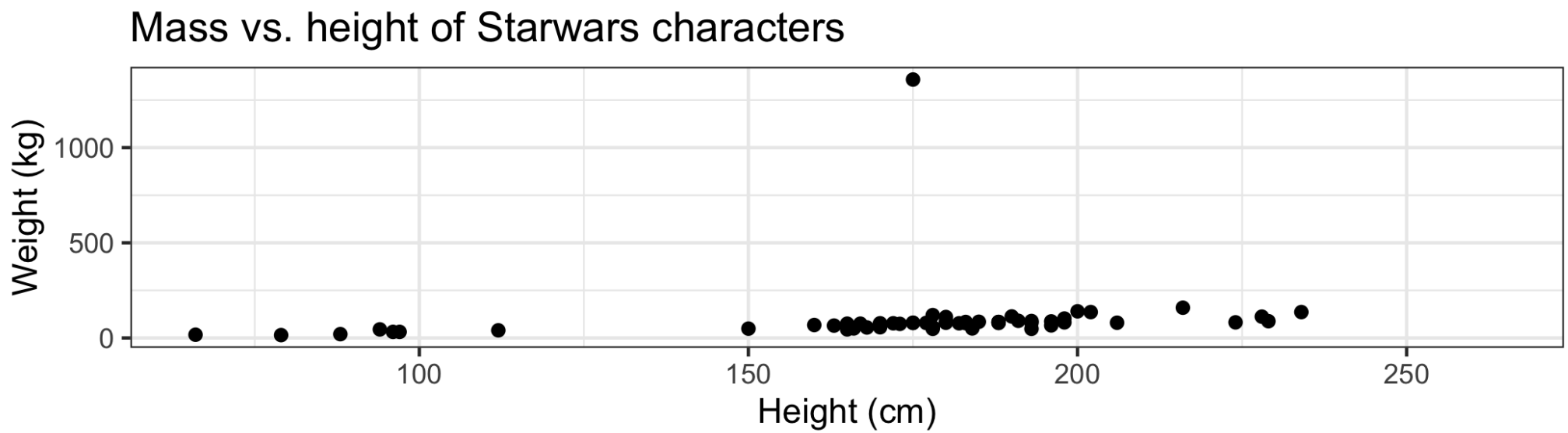
```
## Warning: Removed 28 rows containing missing values (geom_point).
```



What does the warning mean?

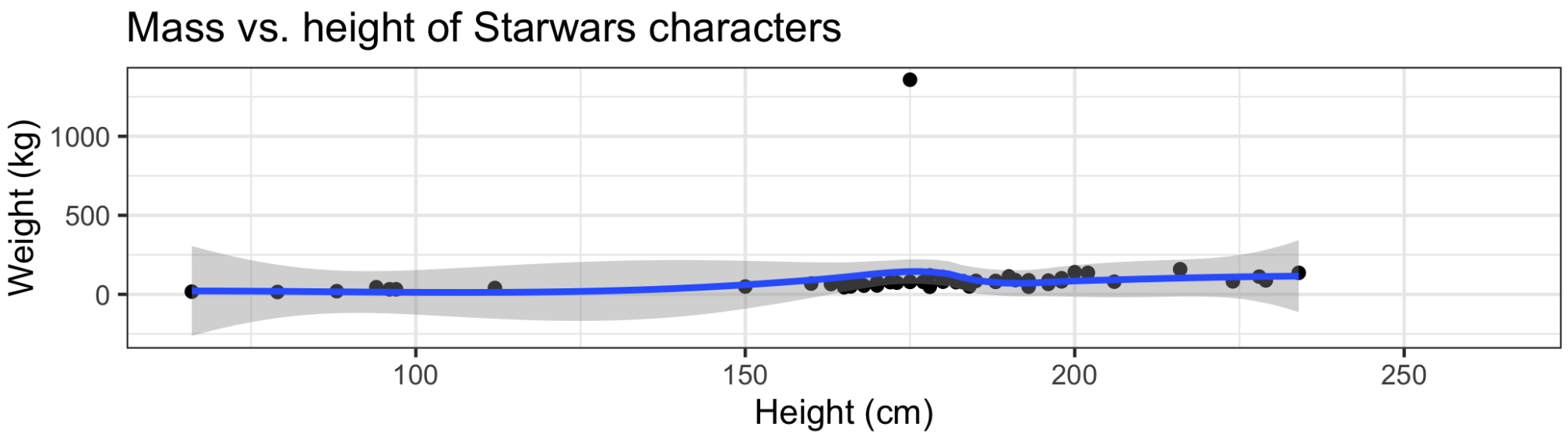
```
ggplot(data = starwars, mapping = aes(x = height, y = mass)) +
  geom_point() +
  labs(title = "Mass vs. height of Starwars characters",
       x = "Height (cm)", y = "Weight (kg)")
```

```
## Warning: Removed 28 rows containing missing values (geom_point).
```



What does `geom_smooth()` do?

```
ggplot(data = starwars, mapping = aes(x = height, y = mass)) +
  geom_point() +
  geom_smooth() +
  labs(title = "Mass vs. height of Starwars characters",
       x = "Height (cm)", y = "Weight (kg)")
```



Hello ggplot2!

- **`ggplot()`** is the main function in ggplot2 and plots are constructed in layers
- The structure of the code for plots can often be summarized as

```
ggplot +
  geom_xxx
```

Hello ggplot2!

- **`ggplot()`** is the main function in ggplot2 and plots are constructed in layers
- The structure of the code for plots can often be summarized as

```
ggplot +
  geom_xxx
```

or, more precisely

```
ggplot(data = [dataset], mapping = aes(x = [x-variable], y = [y-variable])) +
   geom_xxx() +
   other options
```

Hello ggplot2!

To use ggplot2 functions, first load tidyverse

```
library(tidyverse)
```

For help with the ggplot2, see ggplot2.tidyverse.org

Visualizing Star Wars

Dataset terminology

```
starwars
```

```
## # A tibble: 87 x 14
##    name  height  mass hair_color skin_color eye_color birth_year sex   gender
##    <chr>  <int> <dbl> <chr>      <chr>      <chr>           <dbl> <chr> <chr>
##  1 Luke…    172    77 other      fair       blue             19   male  mascu…
##  2 C-3PO    167    75 none       gold       yellow          112   none  mascu…
##  3 R2-D2     96    32 none       white, bl… red              33   none  mascu…
##  4 Dart…    202   136 none       white      yellow           41.9 male  mascu…
##  5 Leia…    150    49 brown      light      brown            19   fema… femin…
##  6 Owen…    178   120 brown      light      blue             52   male  mascu…
##  7 Beru…    165    75 brown      light      blue             47   fema… femin…
##  8 R5-D4     97    32 none       white, red red              NA   none  mascu…
##  9 Bigg…    183    84 black      light      brown            24   male  mascu…
## 10 Obi-…    182    77 other      fair       blue-gray        57   male  mascu…
## # … with 77 more rows, and 5 more variables: homeworld <chr>, species <chr>,
## #   films <list>, vehicles <list>, starships <list>
```

Each row is an observation. Each column is a variable

Luke Skywalker

```
eye_color = blue
hair_color = blond
skin_color = fair
gender = male
species = Human
height = 172 cm
birth_year = 19 BBY (Before Battle of Yavin)
films = c("Revenge of the Sith",
"Return of the Jedi",
"The Empire Strikes Back",
"A New Hope",
"The Force Awakens")
vehicles = c("Snowspeeder", "Imperial Speeder Bike")
starships = c("X-wing", "Imperial shuttle")
weight = 77 kg
```

What's in the Star Wars data?

Take a **`glimpse`** of the data:

```
glimpse(starwars)
```

```
## Rows: 87
## Columns: 14
## $ name       <chr> "Luke Skywalker", "C-3PO", "R2-D2", "Darth Vader", "Lei
## $ height     <int> 172, 167, 96, 202, 150, 178, 165, 97, 183, 182, 188, 18
## $ mass       <dbl> 77.0, 75.0, 32.0, 136.0, 49.0, 120.0, 75.0, 32.0, 84.0,
## $ hair_color <chr> "other", "none", "none", "none", "brown", "brown", "bro
## $ skin_color <chr> "fair", "gold", "white, blue", "white", "light", "light
## $ eye_color  <chr> "blue", "yellow", "red", "yellow", "brown", "blue", "blu
## $ birth_year <dbl> 19.0, 112.0, 33.0, 41.9, 19.0, 52.0, 47.0, NA, 24.0, 57
## $ sex        <chr> "male", "none", "none", "male", "female", "male", "fema
## $ gender     <chr> "masculine", "masculine", "masculine", "masculine", "fe
## $ homeworld  <chr> "Tatooine", "Tatooine", "Naboo", "Tatooine", "Alderaan"
```

What's in the Star Wars data?

Run the following **in the Console** to view the help

```
?starwars
```

starwars {dplyr} R Documentation

Starwars characters

Description

This data comes from SWAPI, the Star Wars API, http://swapi.co/

Usage

```
starwars
```

Format

A tibble with 87 rows and 13 variables:

name

Name of the character

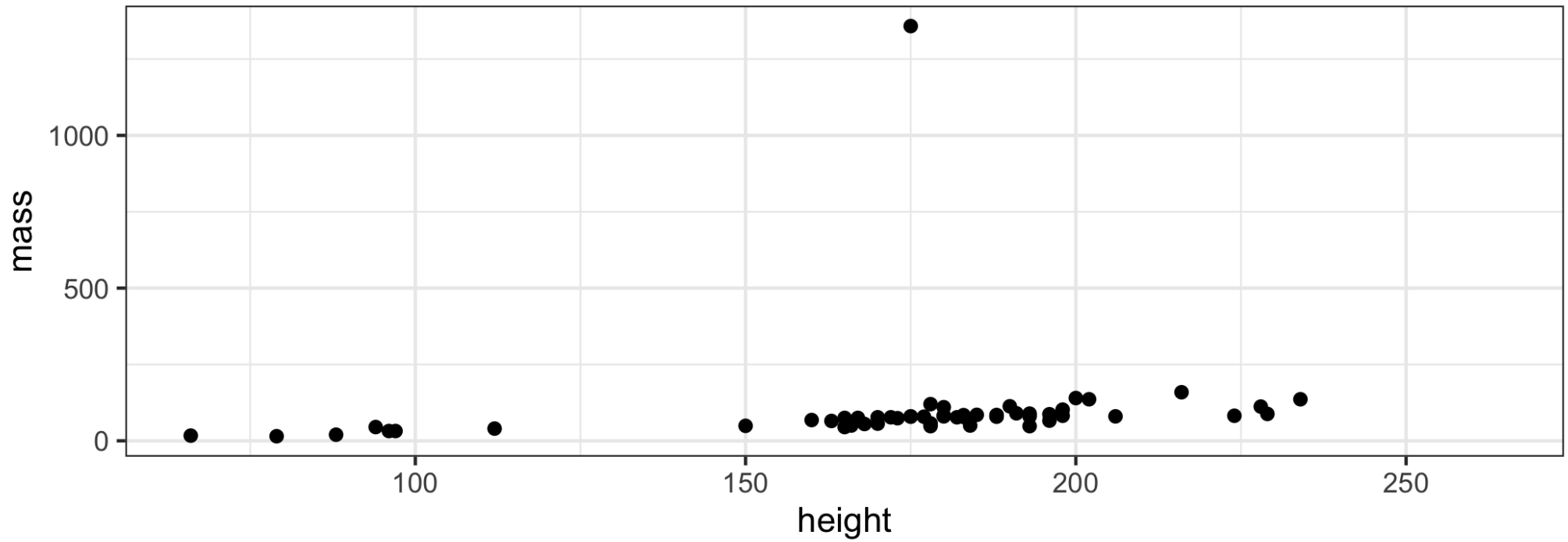
height

Height (cm)

Mass vs. height

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

```
## Warning: Removed 28 rows containing missing values (geom_point).
```



What's that warning?

- Not all characters have height and mass information (hence 28 of them not plotted)

```
## Warning: Removed 28 rows containing missing values (geom_point).
```

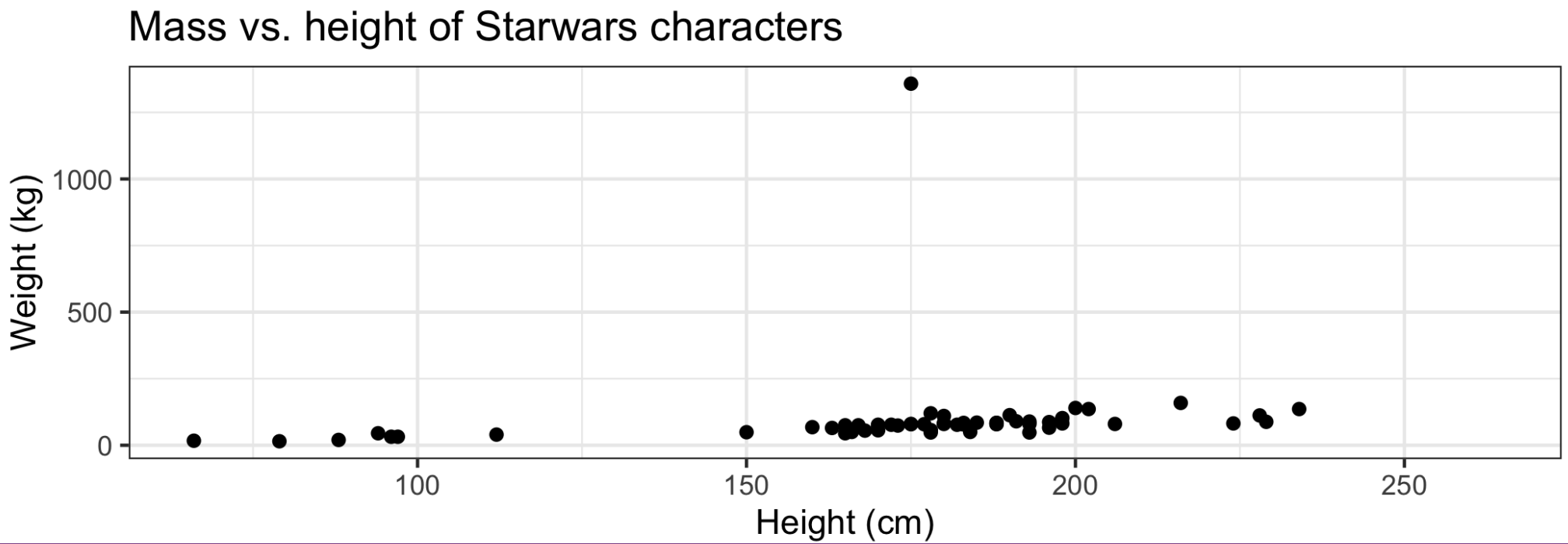
- We can suppress warnings to save space on the output documents, but it's important to note them

- To suppress warning:

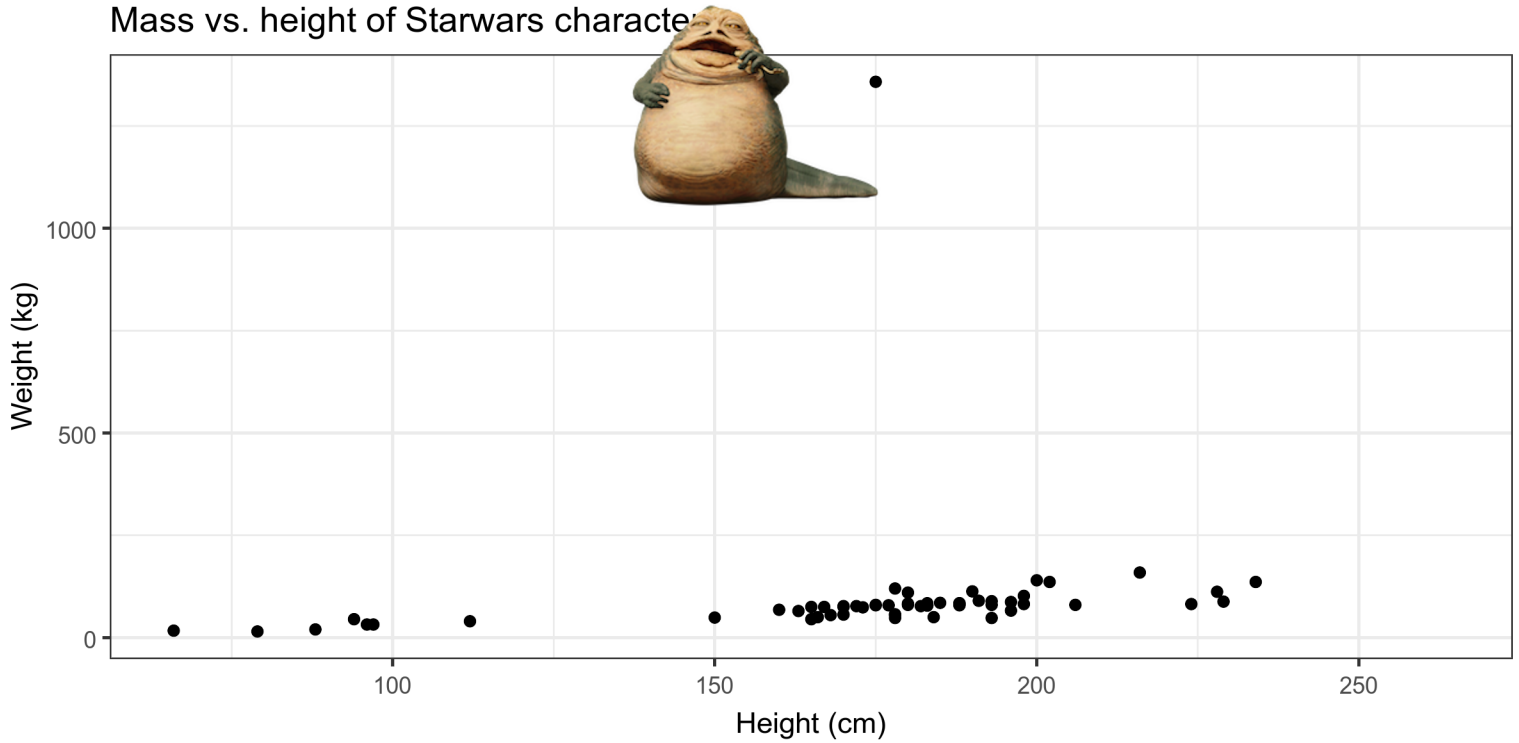
```
{r code-chunk-label, warning=FALSE}
```

Mass vs. height

How would you describe this **relationship**? Who is the not so tall but really heavy character?



Jabba!


Mass vs. height of Starwars characte
1000
500
0
Weight (kg)
100
150
200
250
Height (cm)

Additional variables

We can map additional variables to various features of the plot:

- **aesthetics**
  - shape
  - color
  - size
  - alpha (transparency)
- **faceting**: small multiples displaying different subsets

Aesthetics

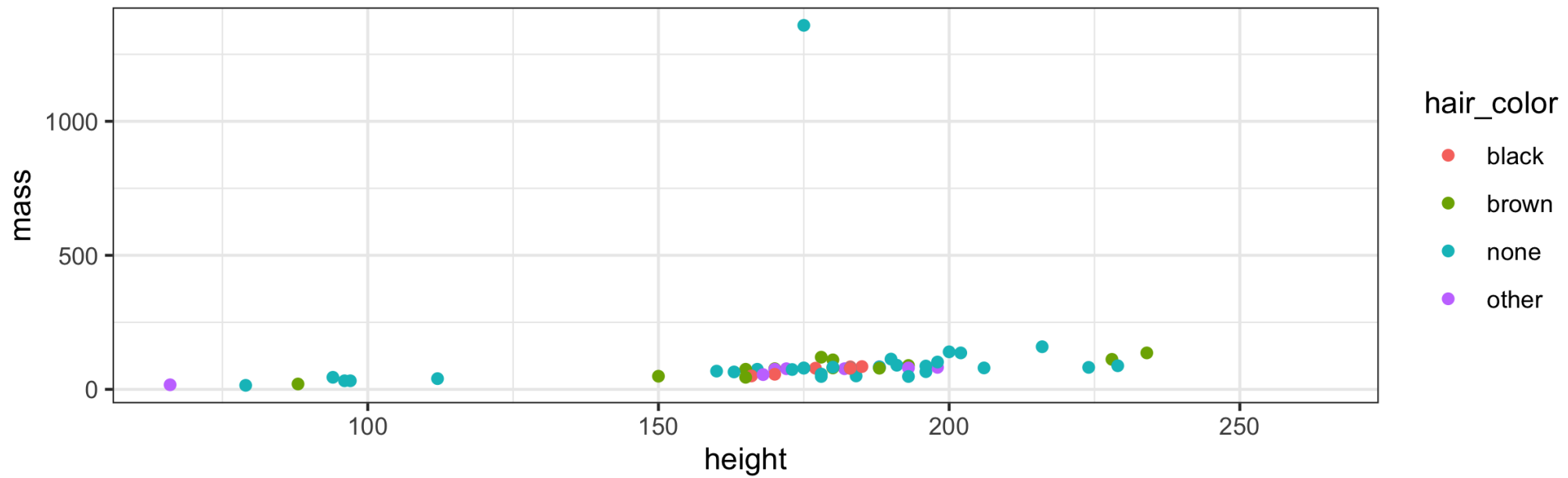
Aesthetics options

Visual characteristics of plotting characters that can be **mapped to a specific variable** in the data are

- **`color`**
- **`size`**
- **`shape`**
- **`alpha`** (transparency)

Mass vs. height + hair color

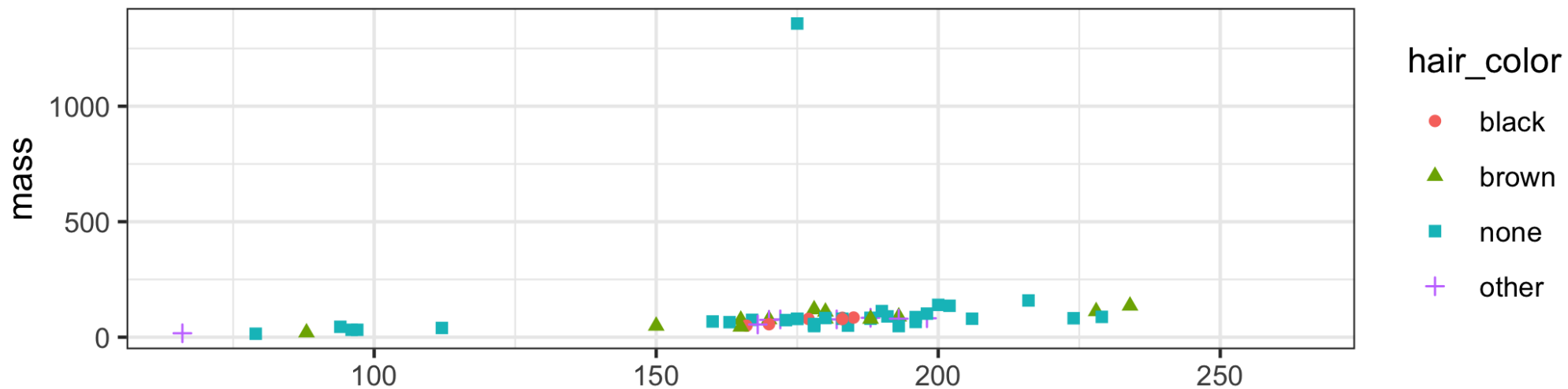
```
ggplot(data = starwars, mapping = aes(x = height, y = mass,
                                      color = hair_color)) +
  geom_point()
```



Mass vs. height + hair color

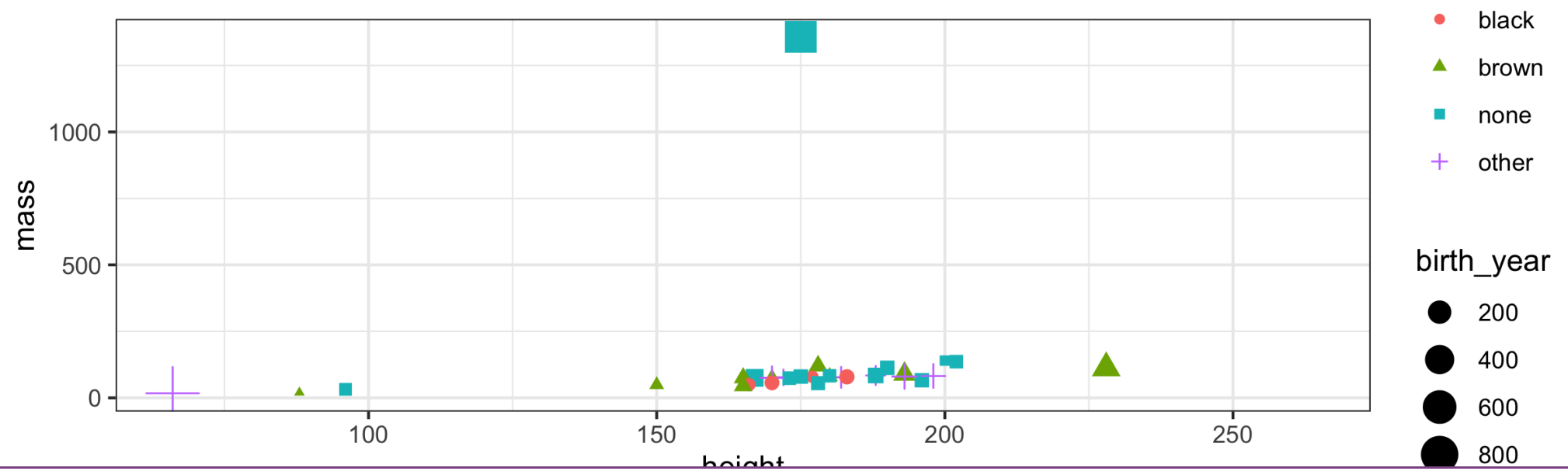
Let's map **shape** and **color** to **hair_color**

```
ggplot(data = starwars,
       mapping = aes(x = height, y = mass, color = hair_color,
                                           shape = hair_color
                                           )) +
  geom_point()
```



Mass vs. height + hair_color + birth year

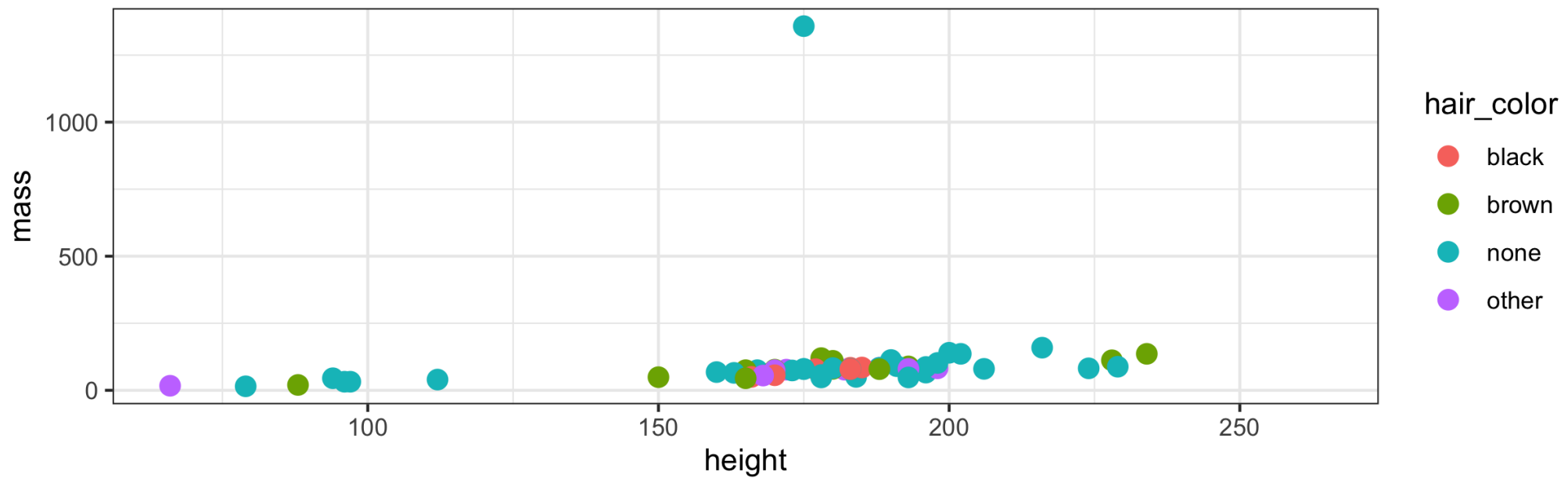
```
ggplot(data = starwars, mapping = aes(x = height, y = mass,
                                      color = hair_color, shape = hair
                                      size = birth_year
                                      )) +
  geom_point()
```



Mass vs. height + hair color

Let's increase the size of all points across the board:

```
ggplot(data = starwars, mapping = aes(x = height, y = mass,
                                      color = hair_color)) +
  geom_point(size = 3)
```



Aesthetics summary

- Continuous variable are measured on a continuous scale
- Discrete variables are measured (or often counted) on a discrete scale

aesthetics	discrete	continuous
color	rainbow of colors	gradient
size	discrete steps	linear mapping between radius and value
shape	different shape for each	shouldn't (and doesn't) work

Use aesthetics (**`aes`**) for mapping features of a plot to a variable, define the features in the **`geom_xxx`** for customization **not** mapped to a variable

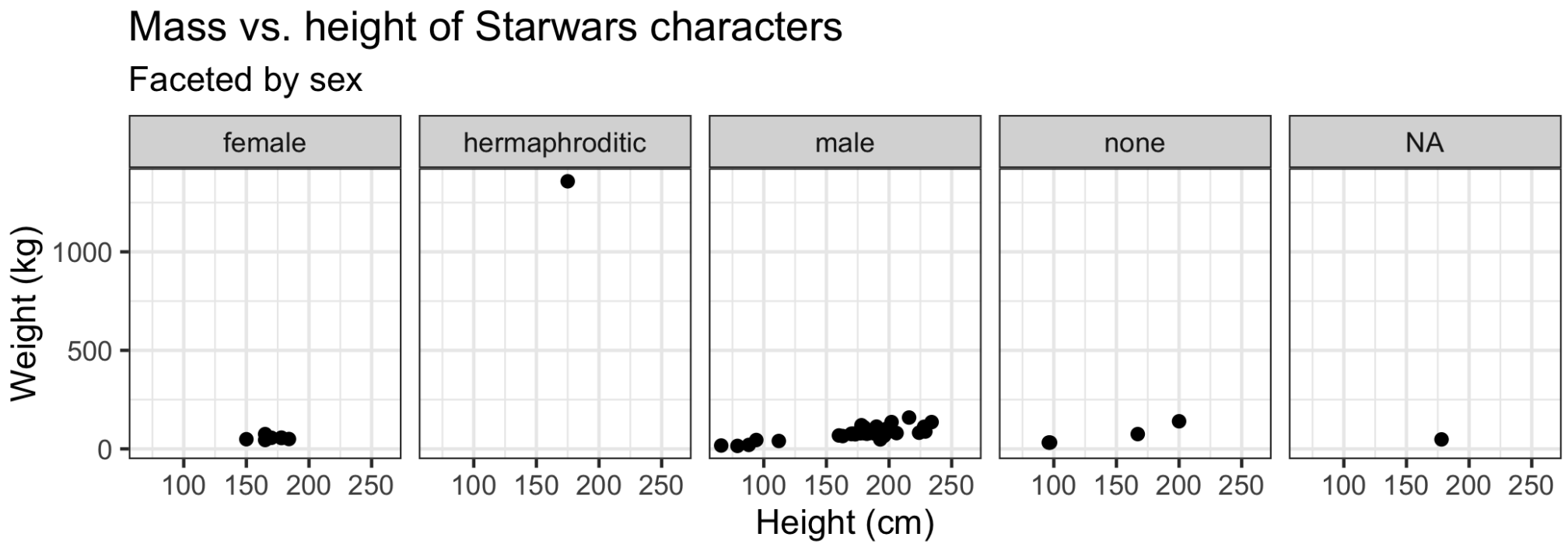
Faceting

Faceting options

- Smaller plots that display different subsets of the data
- Useful for exploring conditional relationships and large data

```
ggplot(data = starwars, mapping = aes(x = height, y = mass)) +
  facet_grid(. ~ sex) +
  geom_point() +
  labs(title = "Mass vs. height of Starwars characters",
       subtitle = "Faceted by sex",
       x = "Height (cm)", y = "Weight (kg)")
```

```
ggplot(data = starwars, mapping = aes(x = height, y = mass)) +
  facet_grid(. ~ sex) +
  geom_point() +
  labs(title = "Mass vs. height of Starwars characters",
       subtitle = "Faceted by sex",
       x = "Height (cm)", y = "Weight (kg)")
```



Dive further...

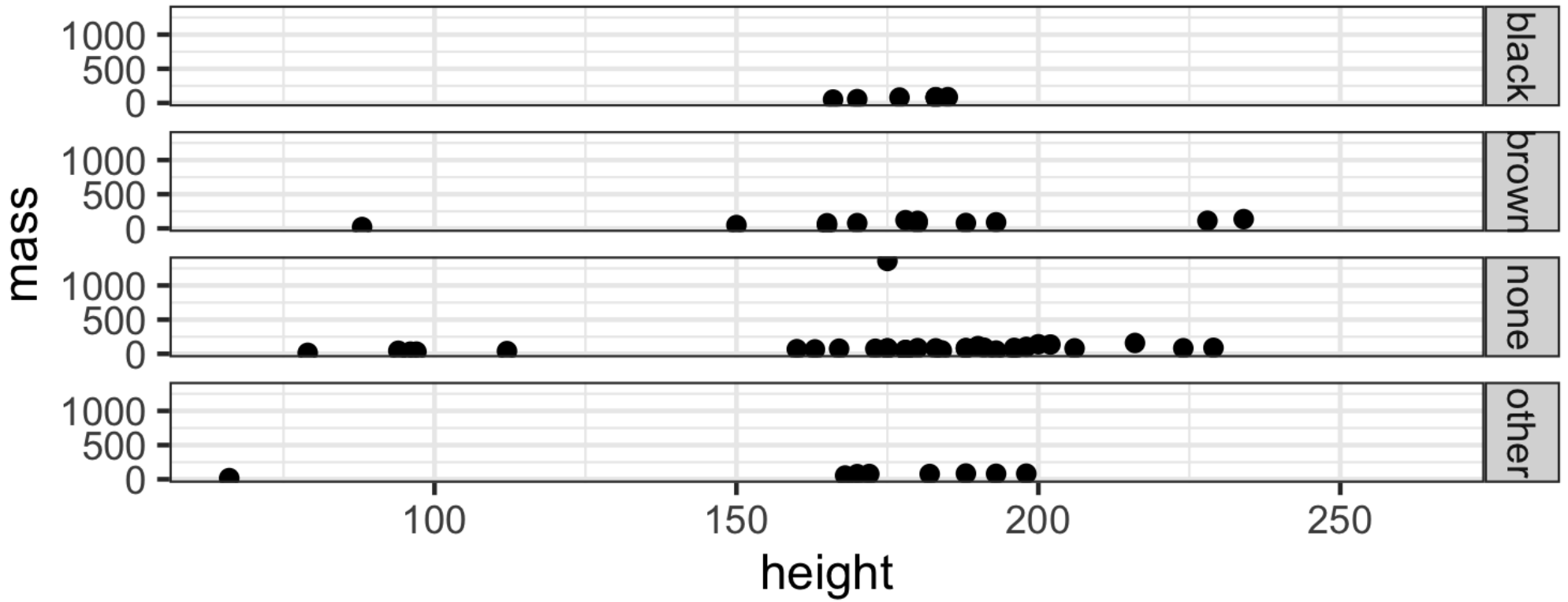
In the next few slides describe what each plot displays. Think about how the code relates to the output.

Dive further...

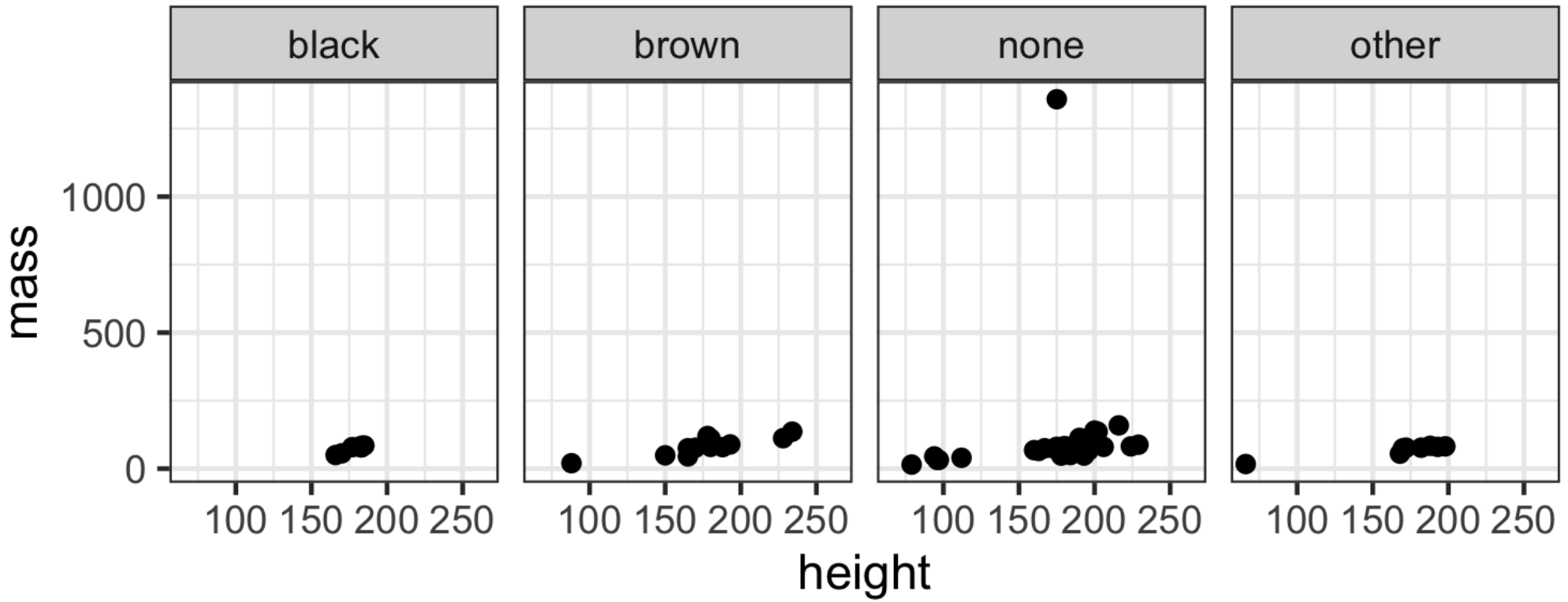
In the next few slides describe what each plot displays. Think about how the code relates to the output.

The plots in the next few slides do not have proper titles, axis labels, etc, so you can more easily focus on what's happening in the plots. But you should always label your plots!

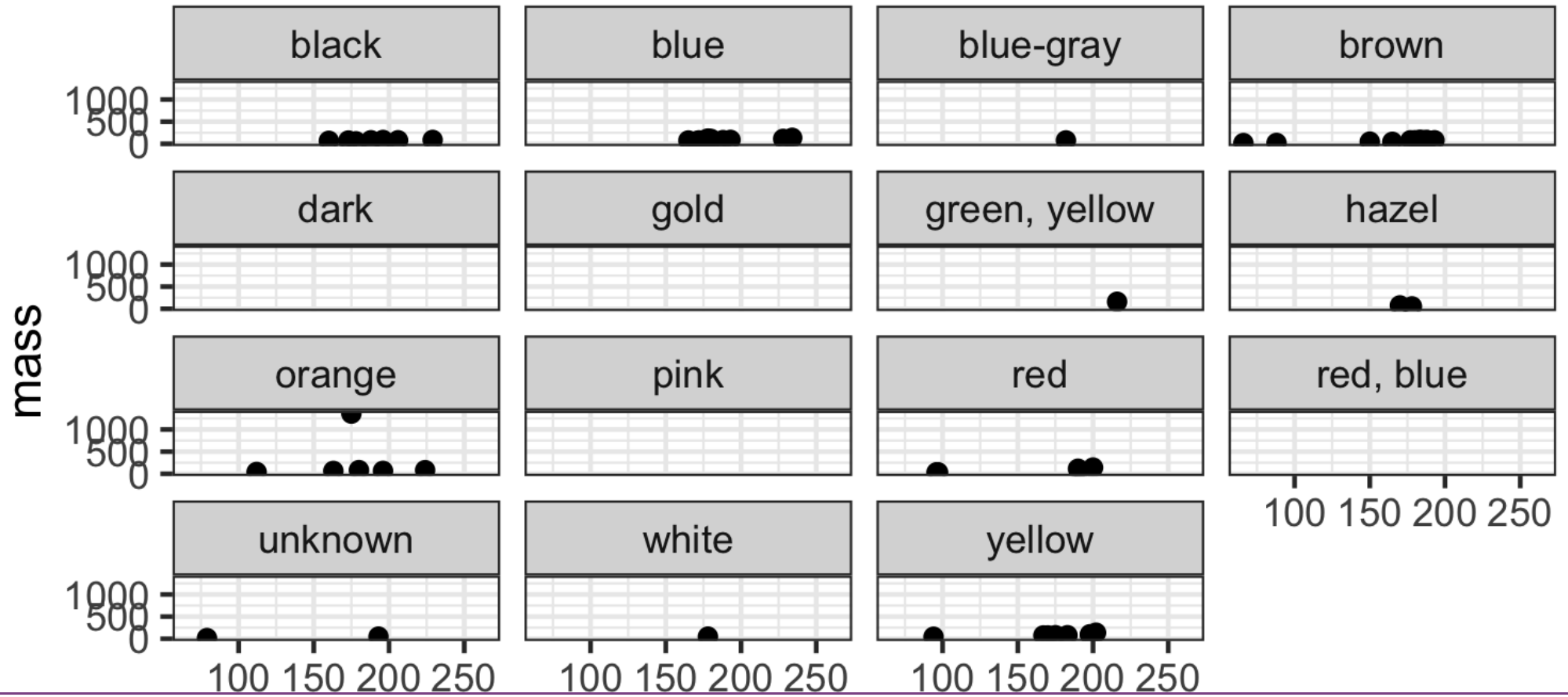
```
ggplot(data = starwars, mapping = aes(x = height, y = mass)) +
  geom_point() +
  facet_grid(hair_color ~ .)
```



```
ggplot(data = starwars, mapping = aes(x = height, y = mass)) +
  geom_point() +
  facet_grid(. ~ hair_color)
```



```
ggplot(data = starwars, mapping = aes(x = height, y = mass)) +
  geom_point() +
  facet_wrap(~ eye_color)
```



Facet summary

- `facet_grid()`:
  - 2d grid
  - `rows ~ cols`
  - use `.` for no split

Facet summary

- **`facet_grid()`**:
  - 2d grid
  - **`rows ~ cols`**
  - use **`.`** for no split
- **`facet_wrap()`**: 1d ribbon wrapped into 2d

`ggplot2` supplementary resources

1. ggplot2.tidyverse.org
2. **`ggplot2`** cheat sheet
3. STA 523 **`ggplot2`** slides
4. Top 50 **`ggplot2`** visualizations
5. How the BBC uses **`ggplot2`**
6. ggplot2: Elegant Graphics for Data Analysis