2.3: Plotting with ggplot2

Get to Know ggplot2

So far, we've used the base R plotting syntax. While quick plots in base R can still be really useful ways to do preliminary data exploration and visualization, we often want plots that go beyond the basics without too much additional effort. This is where ggplot2 comes in and really shines!

Example

Before we get into the nitty-gritty of how ggplot2 works, Let's run an example using the data about our sick crew members from earlier.

First, we need to load in both the tidyverse package and our data. We can remind oursevles what the data look like using the head() function.

```
# Load package
  library(tidyverse)
  # Load data
  sick <- read csv("data/sick data.csv")</pre>
  # View first few rows
  head(sick)
# A tibble: 6 x 10
 last
          first sex
                         age height_cm weight_kg specialties perc_fish perc_plant
                                            <dbl> <chr>
  <chr>
          <chr> <chr> <dbl>
                                 <dbl>
                                                                    <dbl>
                                                                               <dbl>
1 Gonzal~ Ange~ M
                                  169.
                                             51.4 Hydrology
                                                                   0.994
                                                                             0.00620
                          35
2 Navrat~ John M
                                             96.3 Genetics
                          19
                                  112.
                                                                   0.297
                                                                             0.703
3 Duff
          Josh~ M
                          26
                                  133.
                                             52.1 Horticultu~
                                                                   0.514
                                                                             0.486
4 Dottson Juli~ M
                          36
                                  140.
                                             52.6 Climatology
                                                                   0.686
                                                                             0.314
```

5	al-	Sul~	Mune~	М	26		194.	52.2	Geology	0.292	(0.708
6	Gal	leg~	Rich~	М	29	:	153.	98.1	Climatology	0.329	(0.671
#	i 1	more	e varia	able:	doctor_	trips	<dbl></dbl>					

Here is code to make a scatter plot of the relationship between percent fish in diets and how many trips to the doctor.

```
# Scatterplot of % fish diet and # of doctor visits
ggplot(sick, aes(x = perc_fish, y = doctor_trips)) +
geom_point() +
labs(x = "Percent Fish in Diet",
        y = "Number of Trips to the Doctor") +
theme_light()
```



Nice, right? In the next few classes, we will really start to see the power of ggplot. For now, though, let's focus on how this works.

ggplot2

The package ggplot2 is part of the tidyverse.

Here are some resources you might find helpful now or in the future:

- ggplot2 Book
- UC Business Analytics ggplot2 intro
- R for Data Science Data Visualization chapter

The gg in ggplot2 stands for "Grammar of Graphics." The "grammar" part is based on an idea that all statistical plots have the same fundamental features: data and mapping (and specific components of mapping).

The design is that you work iteratively, building up layer upon layer until you have your final plot.

The typical structure looks like this:

```
# ggplot(data = <DATA>, mapping = aes(<MAPPINGS>)) +
# <GEOM_FUNCTION>()
```

A few things to note:

- We always start with the ggplot() function
- We specify the dataset we want to use
- We specify the mappings (x- and y-axes and some other bits) with the aes() function
- We use a + to add layers
- We specify the type of plot, or geom using one of many possible geom functions
- We use the labs() function to clean up the labels
- We add a theme function to make it more visually readable

Let's iteratively build up to the plot we have made above:

1) Specify the data

ggplot(data = sick)



2) Specify the x-axis (horizontal) and the y-axis (vertical) in the $\verb"aes"()$ function.

ggplot(data = sick, mapping = aes(x = perc_fish, y = doctor_trips))



3) Add the type of plot we want using a geom function.

ggplot(data = sick, mapping = aes(x = perc_fish, y = doctor_trips)) +
geom_point()



4) Clean up the axis labels with the lab() function so they are more easily interpreted.

```
ggplot(data = sick, mapping = aes(x = perc_fish, y = doctor_trips)) +
geom_point() +
labs(x = "Percent Fish in Diet",
    y = "Number of Trips to the Doctor")
```



5) Choose a theme function to make the plot more aesthetically pleasing. My favorites are theme_bw(), theme_classic(), and theme_light().

```
ggplot(sick, aes(x = perc_fish, y = doctor_trips)) +
geom_point() +
labs(x = "Percent Fish in Diet",
    y = "Number of Trips to the Doctor") +
theme_light()
```

