Data Science for Biological, Medical and Health Research: Notes for PQHS/CRSP/MPHP 431

Thomas E. Love

2021-08-19

Contents

Working with These Notes **5**
The 431 Course online . . . 5
What You'll Find Here . . . 5
Setting Up R . . . 6
Initial Setup of R Packages . . . 7
The `Love-boost.R` script . . . 7
Additional R Packages installed for this book . . . 8

1 Data Science **9**
1.1 Data Science Project Cycle . . . 10
1.2 Data Science and the 431 Course . . . 11
1.3 What The Course Is and Isn't . . . 11

Part A. Exploring Data **15**

2 Looking at the Palmer Penguins **15**
2.1 Package Loading, then Dealing with Missing Data . . . 15
2.2 Counting Things and Making Tables . . . 16
2.3 Visualizing the Data in a Graph (or a few…) . . . 17
2.4 Six Ways To "Improve" This Graph . . . 19
2.5 A Little Reflection . . . 20

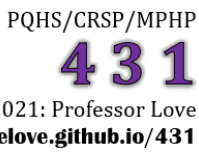
3 NHANES: Initial Exploring **21**
3.1 The NHANES data: A First Sample . . . 21
3.2 Age and Height . . . 22
3.3 Subset of Subjects with Known Age and Height . . . 24
3.4 The Distinction between `Gender` and `Sex` . . . 24
3.5 Age-Height and Sex? . . . 25

Working with These Notes

1. This document is broken down into multiple chapters. Use the table of contents on the left side of the screen to navigate, and use the hamburger icon (horizontal bars) at the top of the document to open or close the table of contents.
2. At the top of the document, you'll see additional icons which you can click to
   - search the document,
   - change the size, font or color scheme of the page, and
   - download a PDF or EPUB (Kindle-readable) version of the entire document.
3. The document will be updated (unpredictably) throughout the semester.

The 431 Course online

The **main web page** for the 431 course in Fall 2021 is https://thomaselove.github.io/431/. Go there for all information related to the course.



What You'll Find Here

These Notes provide a series of examples using R to work through issues that are likely to come up in PQHS/CRSP/MPHP 431. What you will mostly find are brief explanations of a key idea or summary, accompanied (most of the time) by R code and a demonstration of the results of applying that code.

While these Notes share some of the features of a textbook, they are neither comprehensive nor completely original. The main purpose is to give 431 students a set of common materials on which to draw during the course. In class, we will sometimes:

- reiterate points made in this document,
- amplify what is here,
- simplify the presentation of things done here,
- use new examples to show some of the same techniques,
- refer to issues not mentioned in this document,

but what we don't do is follow these notes very precisely. We assume instead that you will read the materials and try to learn from them, just as you will attend classes and try to learn from them. We welcome feedback of all kinds on this document or anything else.

Everything you see here is available to you as HTML or PDF. You will also have access to the R Markdown files, which contain the code which generates everything in the document, including all of the R results. We will demonstrate the use of R Markdown (this document is generated with the additional help of an R package called `bookdown`) and RStudio (the "program" we use to interface with the R language) in class.

All data and R code related to these notes are also available to you.

Setting Up R

These Notes make extensive use of

- the statistical software language R, and
- the development environment R Studio,

both of which are free, and you'll need to install them on your machine. Instructions for doing so are in found in the course syllabus.

If you need an even gentler introduction, or if you're just new to R and RStudio and need to learn about them, we encourage you to take a look at http://moderndive.com/, which provides an introduction to statistical and data sciences via R at Ismay and Kim (2021).

These notes were written using R Markdown. R Markdown, like R and R Studio, is free and open source.

R Markdown is described as an *authoring framework* for data science, which lets you

- save and execute R code
- generate high-quality reports that can be shared with an audience

This description comes from http://rmarkdown.rstudio.com/lesson-1.html which you can visit to get an overview and quick tour of what's possible with R Markdown.

Another excellent resource to learn more about R Markdown tools is the Communicate section (especially the R Markdown chapter) of Grolemund and Wickham (2021).

Initial Setup of R Packages

To start, I'll present a series of commands I run at the beginning of these Notes. These particular commands set up the output so it will look nice as either an HTML or PDF file, and also set up R to use several packages (libraries) of functions that expand its capabilities. A chunk of code like this will occur near the top of any R Markdown work.

```
knitr::opts_chunk$set(comment = NA)

library(knitr)
library(magrittr)
library(janitor)
library(NHANES)
library(palmerpenguins)
library(patchwork)
library(rms)
library(mosaic)
library(Epi)
library(naniar)
library(broom) # note: tidymodels includes the broom package
library(tidyverse) # note: tidyverse includes the dplyr and ggplot2 packages

theme_set(theme_bw())
```

I have deliberately set up this list of loaded packages to be relatively small, and will add some others later in these Notes. You only need to install a package once, but you need to reload it every time you start a new session.

The `Love-boost.R` script

Starting in October, we'll make use of a few scripts I've gathered for you.

```
source("data/Love-boost.R")
```

Additional R Packages installed for this book

Some packages need to be installed on the user's system, but do not need to be loaded by R in order to run the code presented in this set of notes until later. These additional packages include the following.

```
boot
car
GGally
gt
psych
modelsummary
naniar
visdat
```

Chapter 1

Data Science

The definition of **data science** can be a little slippery. One current view of data science, is exemplified by Steven Geringer's 2014 Venn diagram.

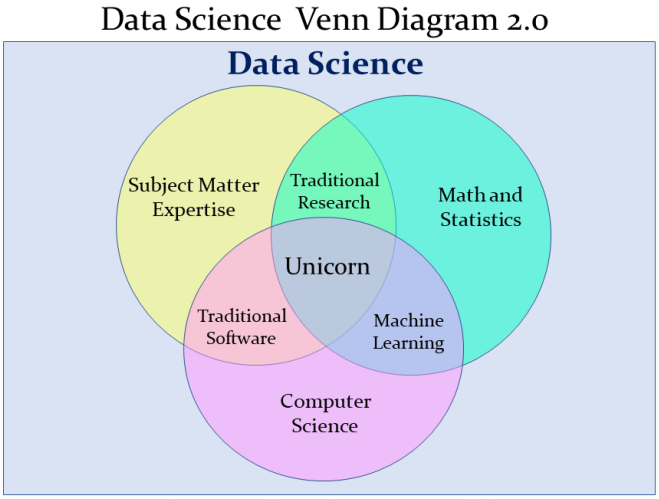


Figure 1.1: Data Science Venn Diagram from Steven Geringer

- The field encompasses ideas from mathematics and statistics and from computer science, but with a heavy reliance on subject-matter knowledge. In our case, this includes clinical, health-related, medical or biological knowledge.
- As Gelman and Nolan (2017) suggest, the experience and intuition necessary for good statistical practice are hard to obtain, and teaching data

science provides an excellent opportunity to reinforce statistical thinking skills across the full cycle of a data analysis project.
- The principal form in which computer science (coding/programming) play a role in this course is to provide a form of communication. You'll need to learn how to express your ideas not just orally and in writing, but also through your code.

Data Science is a **team** activity. Everyone working in data science brings some part of the necessary skillset, but no one person can cover all three areas alone for excellent projects.

> [The individual who is truly expert in all three key areas (mathematics/statistics, computer science and subject-matter knowledge) is] a mythical beast with magical powers who's rumored to exist but is never actually seen in the wild.
>
> http://www.kdnuggets.com/2016/10/battle-data-science-venn-diagrams.html

1.1 Data Science Project Cycle

A typical data science project can be modeled as follows, which comes from the introduction to the amazing book **R for Data Science**, by Garrett Grolemund and Hadley Wickham, which is a key text for this course (Grolemund and Wickham, 2021).

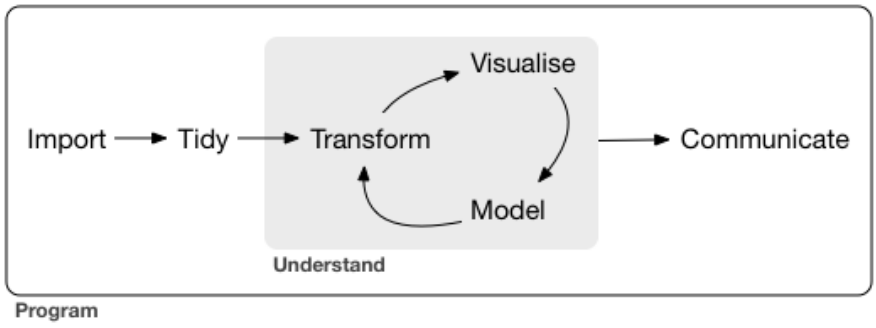


Figure 1.2: Source: R for Data Science: Introduction

This diagram is sometimes referred to as the Krebs Cycle of Data Science. For more on the steps of a data science project, we encourage you to read the Introduction of Grolemund and Wickham (2021).

1.2 Data Science and the 431 Course

We'll discuss each of these elements in the 431 course, focusing at the start on understanding our data through transformation, modeling and (especially in the early stages) visualization. In 431, we learn how to get things done.

- We get people working with R and R Studio and R Markdown, even if they are completely new to coding. A gentle introduction is provided at Ismay and Kim (2021)
- We learn how to use the `tidyverse` (http://www.tidyverse.org/), an array of tools in R (mostly developed by Hadley Wickham and his colleagues at R Studio) which share an underlying philosophy to make data science faster, easier, more reproducible and more fun. A critical text for understanding the tidyverse is Grolemund and Wickham (2021). Tidyverse tools facilitate:
  - **importing** data into R, which can be the source of intense pain for some things, but is really quite easy 95% of the time with the right tool.
  - **tidying** data, that is, storing it in a format that includes one row per observation and one column per variable. This is harder, and more important, than you might think.
  - **transforming** data, perhaps by identifying specific subgroups of interest, creating new variables based on existing ones, or calculating summaries.
  - **visualizing** data to generate actual knowledge and identify questions about the data - this is an area where R really shines, and we'll start with it in class.
  - **modeling** data, taking the approach that modeling is complementary to visualization, and allows us to answer questions that visualization helps us identify.
  - and last, but definitely not least, **communicating** results, models and visualizations to others, in a way that is reproducible and effective.
- Some programming/coding is an inevitable requirement to accomplish all of these aims. If you are leery of coding, you'll need to get past that, with the help of this course and our stellar teaching assistants. Getting started is always the most challenging part, but our experience is that most of the pain of developing these new skills evaporates by early October.

1.3 What The Course Is and Isn't

The 431 course is about **getting things done**. In developing this course, we adopt a modern approach that places data at the center of our work. Our goal is to teach you how to do truly reproducible research with modern tools. We

want you to be able to collect and use data effectively to address questions of interest.

The curriculum includes more on several topics than you might expect from a standard graduate introduction to biostatistics.

- data gathering
- data wrangling
- exploratory data analysis and visualization
- multivariate modeling
- communication

It also nearly completely avoids formalism and is extremely applied - this is absolutely **not** a course in theoretical or mathematical statistics, and these Notes reflect that approach.

There's very little of the mathematical underpinnings here:

$$f(x) = \frac{e^{-(x-\mu)^2/(2\sigma^2)}}{\sigma\sqrt{2\pi}}$$

Instead, these notes (and the course) focus on how we get R to do the things we want to do, and how we interpret the results of our work. Our next Chapter provides a first example.

Part A. Exploring Data

Chapter 2

Looking at the Palmer Penguins

The data in the `palmerpenguins` package in R include size measurements, clutch observations, and blood isotope ratios for adult foraging Adélie, Chinstrap, and Gentoo penguins observed on islands in the Palmer Archipelago near Palmer Station, Antarctica. The data were collected and made available by Dr. Kristen Gorman and the Palmer Station Long Term Ecological Research (LTER) Program.

For more on the `palmerpenguins` package, visit https://allisonhorst.github.io/palmerpenguins/.

2.1 Package Loading, then Dealing with Missing Data

To start, let's load up the necessary R packages to manage the data and summarize it in a small table, and a plot. We've actually done this previously, but we'll repeat the steps here, because it's worth seeing what R is doing.

In this case, we'll load up five packages.

```
library(palmerpenguins)  # source for the data set
library(janitor)         # some utilities for cleanup and simple tables
library(magrittr)        # provides us with the pipe %>% for code management
library(dplyr)           # part of the tidyverse: data management tools
library(ggplot2)         # part of the tidyverse: tools for plotting data
```

It's worth remembering that everything after the `#` on each line above is just a comment for the reader, and is ignored by R. We'll see later that the loading

of a single package (called `tidyverse`) gives us both the `dplyr` and `ggplot2` packages, as well as several other useful things.

Next, let's take the `penguins` data from the `palmerpenguins` package, and identify those observations which have complete data (so, no missing values) in four variables of interest. We'll store that result in a new data frame (think of this as a data set) called `new_penguins` and then take a look at that result using the following code.

```
new_penguins <- penguins %>%
    filter(complete.cases(flipper_length_mm, body_mass_g, species, sex))

new_penguins
```

```
# A tibble: 333 x 8
   species island    bill_length_mm bill_depth_mm
   <fct>   <fct>              <dbl>         <dbl>
 1 Adelie  Torgersen           39.1          18.7
 2 Adelie  Torgersen           39.5          17.4
 3 Adelie  Torgersen           40.3          18
 4 Adelie  Torgersen           36.7          19.3
 5 Adelie  Torgersen           39.3          20.6
 6 Adelie  Torgersen           38.9          17.8
 7 Adelie  Torgersen           39.2          19.6
 8 Adelie  Torgersen           41.1          17.6
 9 Adelie  Torgersen           38.6          21.2
10 Adelie  Torgersen           34.6          21.1
# ... with 323 more rows, and 4 more variables:
#   flipper_length_mm <int>, body_mass_g <int>, sex <fct>,
#   year <int>
```

2.2 Counting Things and Making Tables

So, how many penguins are in our `new_penguins` data? When we printed out the result, we got an answer, but (as with many things in R) there are many ways to get the same result.

```
nrow(new_penguins)
```

```
[1] 333
```

How do our `new_penguins` data break down by sex and species?

```
new_penguins %>%
    tabyl(sex, species) # tabyl comes from the janitor package
```

```
    sex Adelie Chinstrap Gentoo
 female     73        34     58
```

```
male     73        34     61
```

Note the strange spelling of `tabyl` here. The output is reasonably clear, but could we make that table a little prettier, and while we're at it, can we add the row and column totals to it?

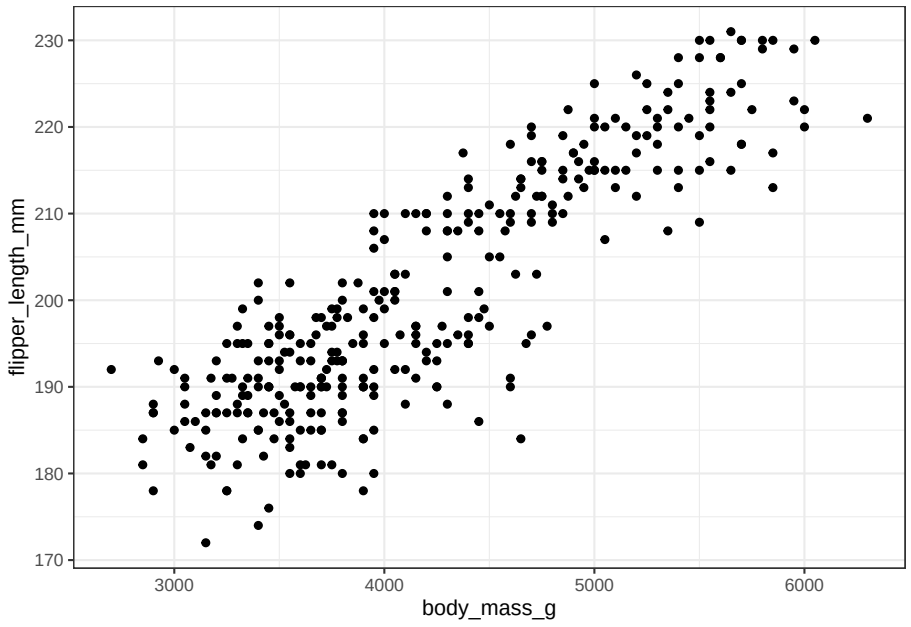
```
new_penguins %>%
    tabyl(sex, species) %>%
    adorn_totals(where = c("row", "col")) %>% # add row, column totals
    kable  # one convenient way to make the table prettier
```

sex	Adelie	Chinstrap	Gentoo	Total
female	73	34	58	165
male	73	34	61	168
Total	146	68	119	333

2.3 Visualizing the Data in a Graph (or a few…)

Now, let's look at the other two variables of interest. Let's create a graph showing the association of body mass with flipper length across the complete set of 333 penguins.

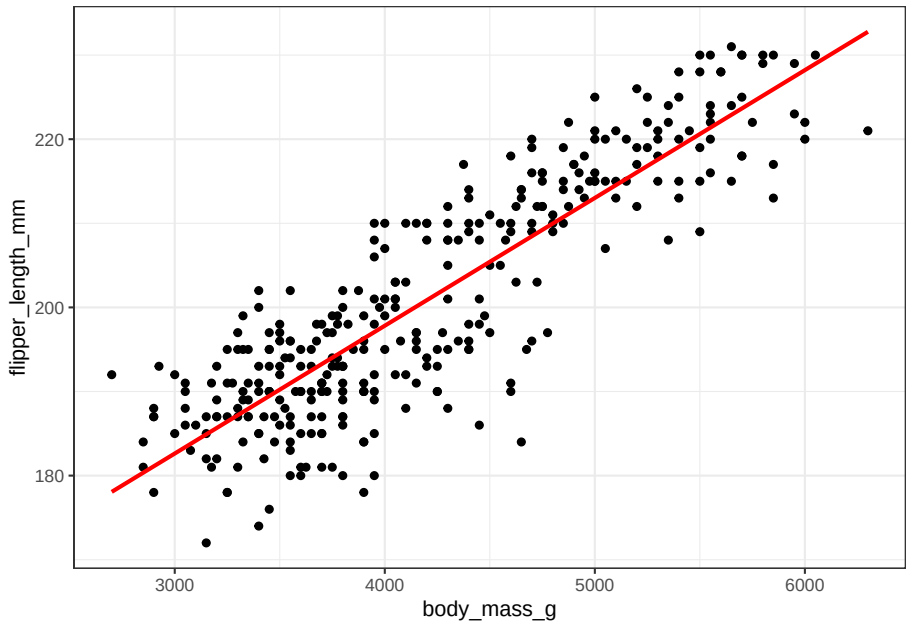
```
ggplot(new_penguins, aes(x = body_mass_g, y = flipper_length_mm)) +
    geom_point()
```



Some of you may want to include a straight-line model (fit by a classical linear

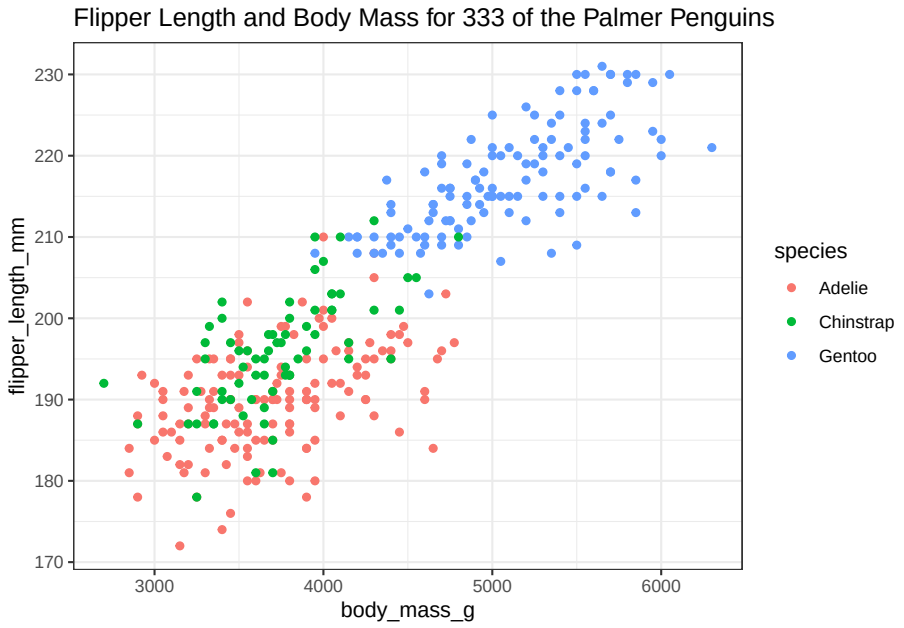
regression) to this plot. One way to do that in R involves the addition of a single line of code, like this:

```
ggplot(new_penguins, aes(x = body_mass_g, y = flipper_length_mm)) +
    geom_point() +
    geom_smooth(method = "lm", formula = y ~ x,
                col = "red", se = FALSE)
```



Whenever we build a graph for ourselves, these default choices may be sufficient. But I'd like to see a prettier version if I was going to show it to someone else. So, I might use a different color for each species, and I might neaten up the theme (to get rid of the default grey background) and add a title, like this.

```
ggplot(new_penguins, aes(x = body_mass_g, y = flipper_length_mm, col = species)) +
    geom_point() +
    theme_bw() +
    labs(title = "Flipper Length and Body Mass for 333 of the Palmer Penguins")
```



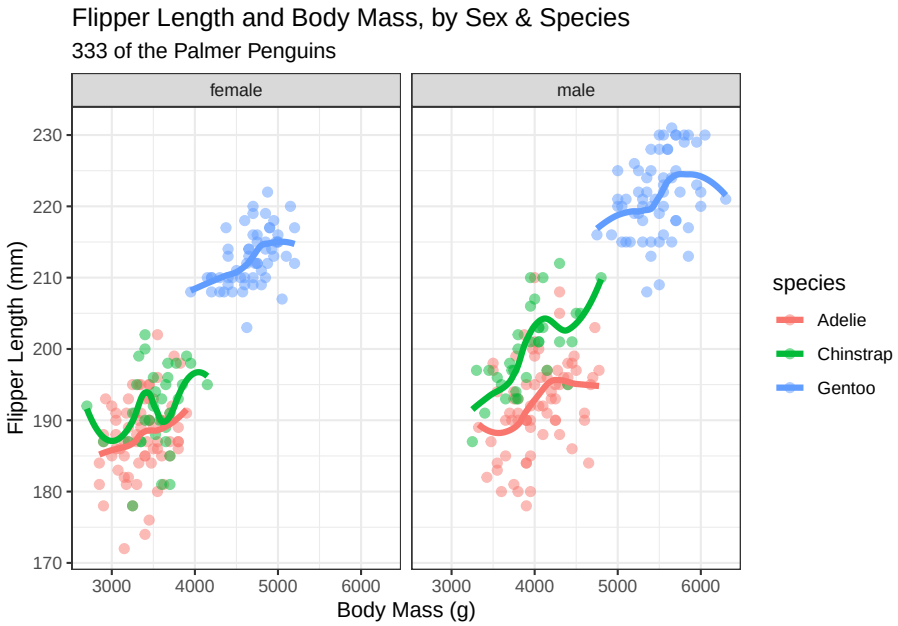
2.4 Six Ways To "Improve" This Graph

Now, let's build a new graph. Here, I want to:

1. plot the relationship between body mass and flipper length in light of both Sex and Species
2. increase the size of the points and add a little transparency so we can see if points overlap,
3. add some smooth curves to summarize the relationships between the two quantities (body mass and flipper length) within each combination of species and sex,
4. split the graph into two "facets" (one for each sex),
5. improve the axis labels,
6. improve the titles by adding a subtitle, and also adding in some code to count the penguins (rather than hard-coding in the total number.)

```
ggplot(new_penguins, aes(x = body_mass_g, y = flipper_length_mm,
                         col = species)) +
    geom_point(size = 2, alpha = 0.5) +
    geom_smooth(method = "loess", formula = y ~ x,
                se = FALSE, size = 1.5) +
    facet_grid(~ sex) +
    theme_bw() +
    labs(title = "Flipper Length and Body Mass, by Sex & Species",
```

```
        subtitle = paste0(nrow(new_penguins), " of the Palmer Penguins"),
        x = "Body Mass (g)",
        y = "Flipper Length (mm)")
```



2.5 A Little Reflection

What can we learn from these plots and their construction? In particular,

- What do these plots suggest about the center of the distribution of each quantity (body mass and flipper length) overall, and within each combination of Sex and Species?
- What does the final plot suggest about the spread of the distribution of each of those quantities in each combination of Sex and Species?
- What do the plots suggest about the association of body mass and flipper length across the complete set of penguins?
- How does the shape and nature of this body mass - flipper length relationship change based on Sex and Species?
- Do you think it would be helpful to plot a straight-line relationship (rather than a smooth curve) within each combination of Sex and Species in the final plot? Why or why not? (Also, what would we have to do to the code to accomplish this?)
- How was the R code for the plot revised to accomplish each of the six "wants" specified above?

Chapter 3

NHANES: Initial Exploring

Next, we'll explore some data from the US National Health and Nutrition Examination Survey, or NHANES.

We'll display R code as we go, but we'll return to all of the key coding ideas involved later in the Notes.

3.1 The NHANES data: A First Sample

The `NHANES` package provides a sample of 10,000 NHANES responses from the 2009-10 and 2011-12 administrations, in a data frame also called `NHANES`. We can obtain the dimensions of this data frame (think of it as a rectangle of data) with the `dim()` function.

```
dim(NHANES)
```

```
[1] 10000    76
```

We see that we have 10000 rows and 76 columns in the `NHANES` data frame.

For the moment, let's gather a random sample of 1,000 responses from the 10000 rows listed in the `NHANES` data frame, and then identify several variables of interest about those subjects[1]. Some of the motivation for this example came from a Figure in Baumer et al. (2017).

```
# library(NHANES) # already loaded NHANES package/library of functions, data

set.seed(431001)
# use set.seed to ensure that we all get the same random sample
# of 1,000 NHANES subjects in our nh_data collection
```

[1] For more on the NHANES data available in the NHANES package, type ?NHANES in the Console in R Studio.

```
nh_dat1 <-
    slice_sample(NHANES, n = 1000, replace = FALSE) %>%
    select(ID, SurveyYr, Gender, Age, Height)

nh_dat1
```

```
# A tibble: 1,000 x 5
      ID SurveyYr Gender   Age Height
   <int> <fct>    <fct>  <int>  <dbl>
 1 69638 2011_12  female     5   106.
 2 70782 2011_12  male      64   176.
 3 52408 2009_10  female    54   162.
 4 59031 2009_10  female    15   155.
 5 64530 2011_12  male      53   185.
 6 71040 2011_12  male      63   169.
 7 55186 2009_10  female    30   168.
 8 60211 2009_10  male       5   103.
 9 55730 2009_10  male      66   161.
10 68229 2011_12  female    36   170.
# ... with 990 more rows
```

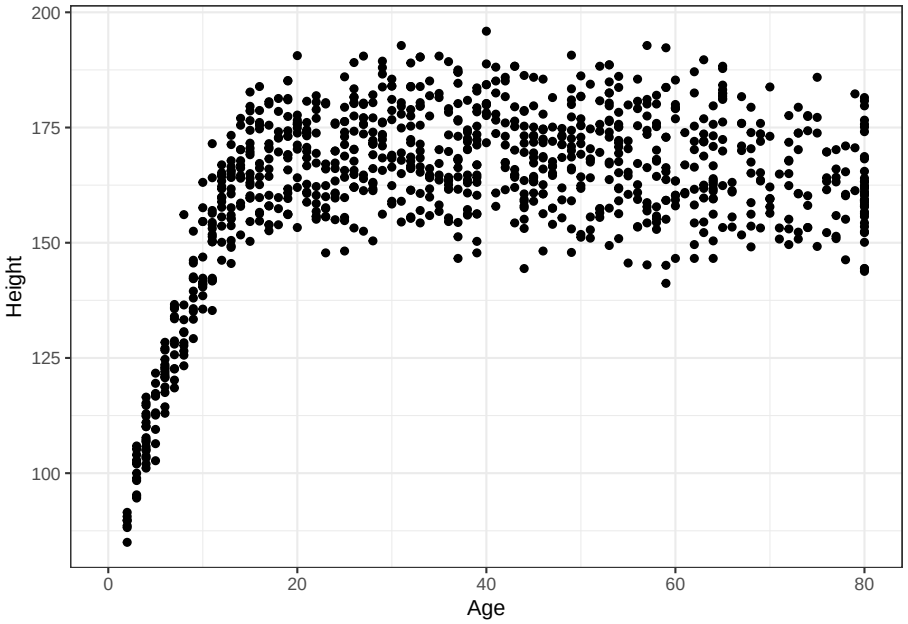
We have 1000 rows (observations) and 5 columns (variables) that describe the responses listed in the rows.

3.2 Age and Height

Suppose we want to visualize the relationship of Height and Age in our 1,000 NHANES observations. The best choice is likely to be a scatterplot.

```
ggplot(data = nh_dat1, aes(x = Age, y = Height)) +
    geom_point()
```

```
Warning: Removed 37 rows containing missing values
(geom_point).
```



We note several interesting results here.

1. As a warning, R tells us that it has "Removed 37 rows containing missing values (geom_point)." Only 963 subjects plotted here, because the remaining 37 people have missing (NA) values for either Height, Age or both.
2. Unsurprisingly, the measured Heights of subjects grow from Age 0 to Age 20 or so, and we see that a typical Height increases rapidly across these Ages. The middle of the distribution at later Ages is pretty consistent at at a Height somewhere between 150 and 175. The units aren't specified, but we expect they must be centimeters. The Ages are clearly reported in Years.
3. No Age is reported over 80, and it appears that there is a large cluster of Ages at 80. This may be due to a requirement that Ages 80 and above be reported at 80 so as to help mask the identity of those individuals.[2]

As in this case, we're going to build most of our visualizations using tools from the `ggplot2` package, which is part of the `tidyverse` series of packages. You'll see similar coding structures throughout this Chapter, most of which are covered as well in Chapter 3 of Grolemund and Wickham (2021).

[2]If you visit the NHANES help file with ?NHANES, you will see that subjects 80 years or older were indeed recorded as 80.

3.3 Subset of Subjects with Known Age and Height

Before we move on, let's manipulate the data set a bit, to focus on only those subjects who have complete data on both Age and Height. This will help us avoid that warning message.

```
nh_dat2 <- nh_dat1 %>%
    filter(complete.cases(Age, Height))

summary(nh_dat2)
```

```
       ID           SurveyYr      Gender         Age
 Min.   :51624   2009_10:487   female:484   Min.   : 2.00
 1st Qu.:57034   2011_12:476   male  :479   1st Qu.:19.00
 Median :62056                              Median :37.00
 Mean   :61967                              Mean   :38.29
 3rd Qu.:67269                              3rd Qu.:56.00
 Max.   :71875                              Max.   :80.00
     Height
 Min.   : 85.0
 1st Qu.:156.2
 Median :165.0
 Mean   :162.3
 3rd Qu.:174.5
 Max.   :195.9
```

Note that the units and explanations for these variables are contained in the NHANES help file, available via typing `?NHANES` in the Console of R Studio, or by typing `NHANES` into the Search bar in R Studio's Help window.

3.4 The Distinction between `Gender` and `Sex`

The `Gender` variable here is mis-named. These data refer to the biological status of these subjects, which is their `Sex`, and not the social construct of `Gender` which can be quite different. In our effort to avoid further confusion, we'll rename the variable `Gender` to `Sex` so as to more accurately describe what is actually measured here.

To do this, we can use this approach…

```
nh_dat2 <- nh_dat1 %>%
    rename(Sex = Gender) %>%
    filter(complete.cases(Age, Height))

summary(nh_dat2)
```

```
       ID           SurveyYr       Sex           Age       
 Min.   :51624   2009_10:487   female:484   Min.   : 2.00  
 1st Qu.:57034   2011_12:476   male  :479   1st Qu.:19.00  
 Median :62056                              Median :37.00  
 Mean   :61967                              Mean   :38.29  
 3rd Qu.:67269                              3rd Qu.:56.00  
 Max.   :71875                              Max.   :80.00  
     Height     
 Min.   : 85.0  
 1st Qu.:156.2  
 Median :165.0  
 Mean   :162.3  
 3rd Qu.:174.5  
 Max.   :195.9  
```

That's better. How many observations do we have now? We could use `dim` to find out the number of rows and columns in this new data set.

```
dim(nh_dat2)
```

```
[1] 963   5
```

Or, we could simply list the data set and read off the result.

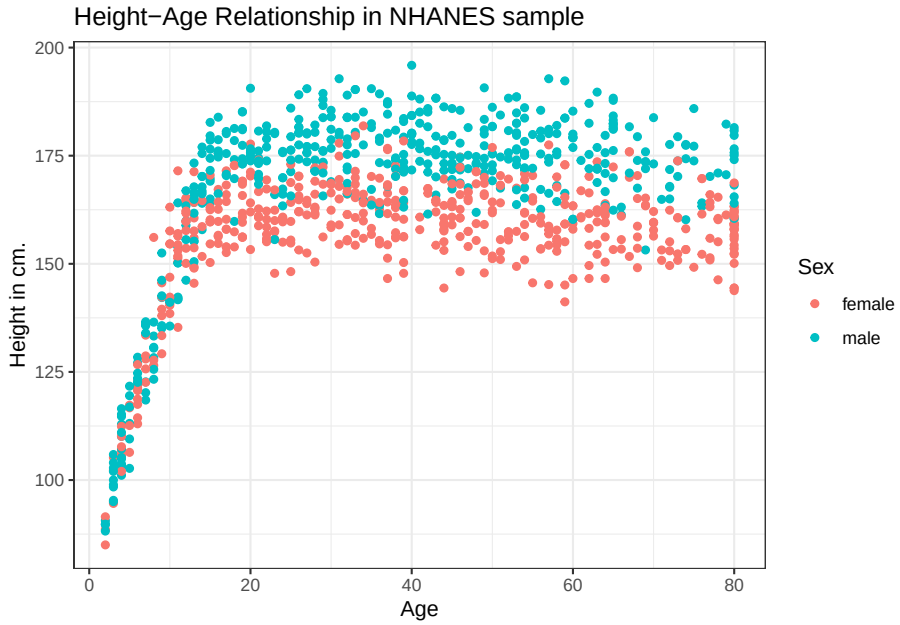
```
nh_dat2
```

```
# A tibble: 963 x 5
      ID SurveyYr Sex      Age Height
   <int> <fct>    <fct>  <int>  <dbl>
 1 69638 2011_12  female     5   106.
 2 70782 2011_12  male      64   176.
 3 52408 2009_10  female    54   162.
 4 59031 2009_10  female    15   155.
 5 64530 2011_12  male      53   185.
 6 71040 2011_12  male      63   169.
 7 55186 2009_10  female    30   168.
 8 60211 2009_10  male       5   103.
 9 55730 2009_10  male      66   161.
10 68229 2011_12  female    36   170.
# ... with 953 more rows
```

3.5 Age-Height and Sex?

Let's add Sex to the plot using color, and also adjust the y axis label to incorporate the units of measurement.

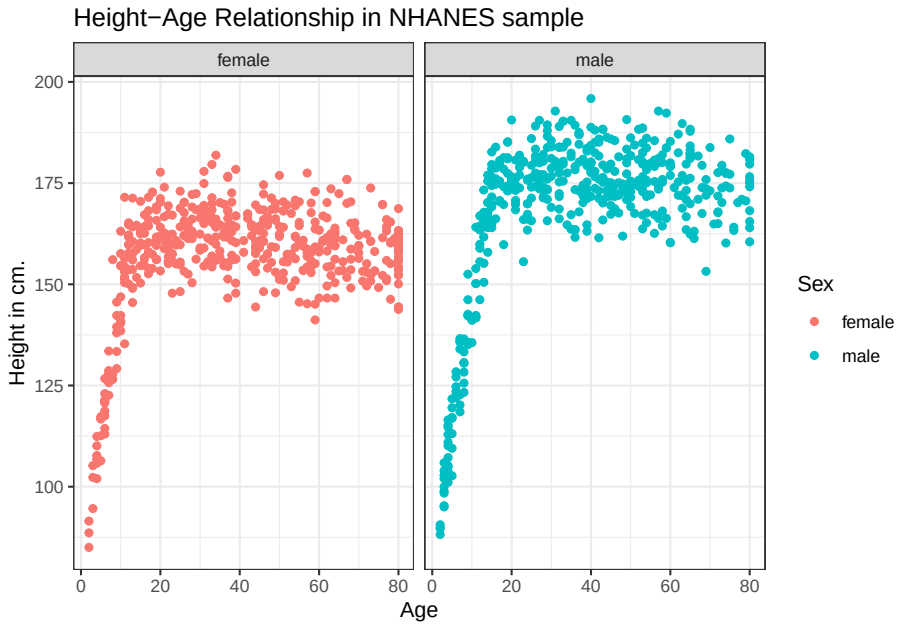
```
ggplot(data = nh_dat2, aes(x = Age, y = Height, color = Sex)) +
    geom_point() +
    labs(title = "Height-Age Relationship in NHANES sample",
         y = "Height in cm.")
```



3.5.1 Can we show the Female and Male relationships in separate panels?

Sure.

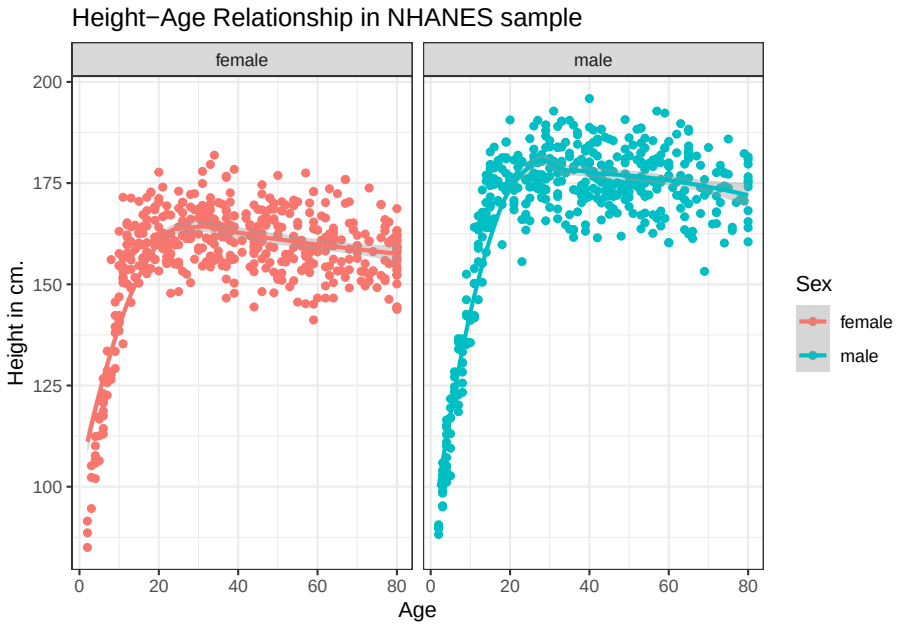
```
ggplot(data = nh_dat2, aes(x = Age, y = Height, color = Sex)) +
    geom_point() +
    labs(title = "Height-Age Relationship in NHANES sample",
         y = "Height in cm.") +
    facet_wrap(~ Sex)
```



3.5.2 Can we add a smooth curve to show the relationship in each plot?

Yep, and let's change the theme of the graph to remove the gray background, too.

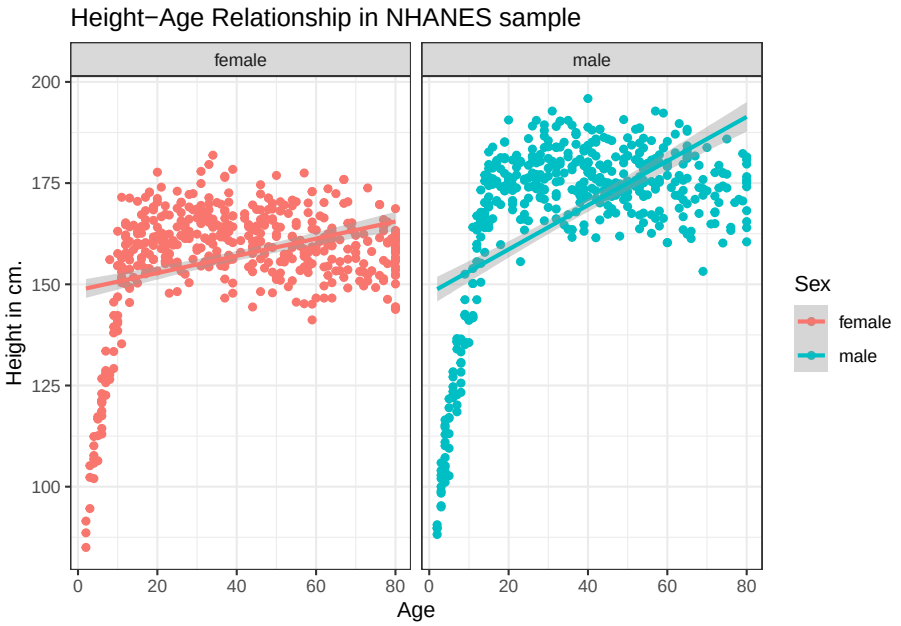
```
ggplot(data = nh_dat2, aes(x = Age, y = Height, color = Sex)) +
    geom_point() +
    geom_smooth(method = "loess", formula = y ~ x) +
    labs(title = "Height-Age Relationship in NHANES sample",
         y = "Height in cm.") +
    theme_bw() +
    facet_wrap(~ Sex)
```



3.5.3 What if we want to assume straight line relationships?

We could look at a linear model in the plot. Does this make sense here?

```
ggplot(data = nh_dat2, aes(x = Age, y = Height, color = Sex)) +
    geom_point() +
    geom_smooth(method = "lm", formula = y ~ x) +
    labs(title = "Height-Age Relationship in NHANES sample",
         y = "Height in cm.") +
    theme_bw() +
    facet_wrap(~ Sex)
```



I hope it seems clear from the graphs that a more complex relationship is going on here than just a straight line.

In the next Section of these Notes, we'll take a more carefully selected sample of NHANES respondents, and study those subjects in greater detail.

Bibliography

Baumer, B. S., Kaplan, D. T., and Horton, N. J. (2017). *Modern Data Science with R*. CRC Press, Boca Raton, FL.

Gelman, A. and Nolan, D. (2017). *Teaching Statistics: A Bag of Tricks*. Oxford University Press, Oxford, UK, second edition.

Grolemund, G. and Wickham, H. (2021). *R for Data Science*. O'Reilly.

Ismay, C. and Kim, A. Y. (2021). *ModernDive: Statistical Inference via Data Science*.