

STATS 250 Lab 02

Basics of Data with R

Nick Seewald
nseewald@umich.edu
Week of 09/4/2020

Reminders

Your tasks for the week running Friday 9/4 - Friday 9/11 (plus an extra):

Task	Due Date	Submission
Homework 1	Friday 9/11 8AM ET	course.work
Lab 1	Friday 9/11 8AM ET	Canvas
Lab 2	Friday 9/11 8AM ET	Canvas
Student Survey	Friday 9/18 8AM ET (1st wave)	Qualtrics Email*

* If you added the class after 8/29, you will be in a later "wave" of the student survey and will get an email soon. Your due date will also be extended appropriately.

This Week's Learning Objectives

Statistical Learning Objectives

1. Understand the structure of data (observations and variables)
2. Think about the scope of a data set: what questions can and cannot be answered with a particular data set?

R Learning Objectives

1. Learn how to "assign" information to "objects" in R
2. See how R "reads in" a data set from a file
3. Be able to identify the names of variables contained in a data set
4. Make a frequency table for one or two variables

R Markdown Tips



- Please don't modify the pre-written chunks of code!
- When copying and pasting code, only take the code
- Don't delete any blank lines
- Always have a blank line after a chunk

RStudio Cloud

<https://rstudio.cloud>

Log in with Google using your U-M Account

Search for the project called lab02-FA20 and click START

If you haven't created an account yet, do so at <https://bit.ly/250millerfa20rsc>

RStudio Desktop

Download materials from the Lab 2 Assignment on Canvas

Unzip the folder, then open lab02-FA20.Rproj to open up RStudio.

Let's get started!

Assignment

R can do a lot of stuff, and we usually want to "save" results to use later.

The way we tell R to remember something is to *assign* that thing a name.

```
x <- 36 / 6
```

The arrow is R's *assignment operator*: "**x** gets 6". R now remembers that **x** is equal to 6.

```
x
```

```
[1] 6
```

Important note: R is "case-sensitive"

R treats lower-case and upper-case letters as *different things*. Check it out:

```
# lower-case x; this is 6  
x
```

```
[1] 6
```

```
# upper-case X; this doesn't exist  
X
```

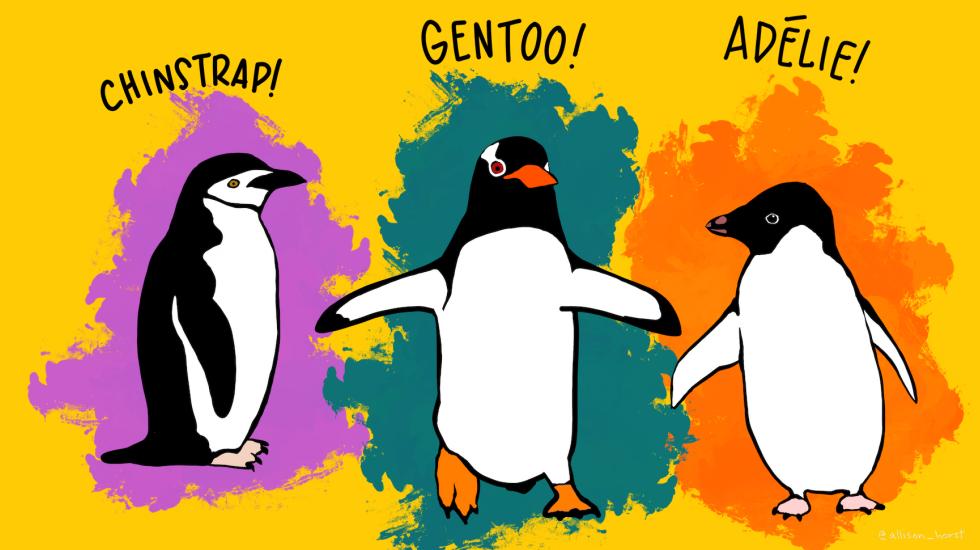
```
Error in eval(expr, envir, enclos): object 'X' not found
```

Palmer Penguins Data



We're going to learn about some basic R functions using a data set on 333 penguins living on 3 islands in the Palmer Archipelago in Antarctica.

Data were made available by [Dr. Kristen Gorman](#) and the [Palmer Station](#), [Antarctica Long Term Ecological Research area](#), a member of the Long Term Ecological Research Network. The data were prepared by [Dr. Allison Horst](#)



Reading in a CSV file

Remember this code?

```
penguins <- read.csv(url("https://raw.githubusercontent.com/STATS250SBI/palmerpenguins/master/inst/e
```

Notice the assignment operator? "**penguins** gets **read.csv** of [blah blah blah]"

Reading in a CSV file

Remember this code?

```
penguins <- read.csv(url("https://raw.githubusercontent.com/STATS250SBI/palmerpenguins/master/inst/e
```

Notice the assignment operator? "**penguins** gets **read.csv** of [blah blah blah]"

read.csv reads in a .csv file (**comma separated values**) and creates a **data.frame**

```
"species","island","bill_length_mm","bill_depth_mm","flipper_length_mm","body_mass_g","sex","year"  
"Adelie","Torgersen",39.1,18.7,181,3750,"male",2007  
"Adelie","Torgersen",39.5,17.4,186,3800,"female",2007  
"Adelie","Torgersen",40.3,18,195,3250,"female",2007  
"Adelie","Torgersen",36.7,19.3,193,3450,"female",2007  
"Adelie","Torgersen",39.3,20.6,190,3650,"male",2007  
"Adelie","Torgersen",38.9,17.8,181,3625,"female",2007
```

Recall: Use head() to peek at the data

```
head(penguins)
```

```
species      island bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
1  Adelie Torgersen        39.1         18.7            181        3750
2  Adelie Torgersen        39.5         17.4            186        3800
3  Adelie Torgersen        40.3         18.0            195        3250
4  Adelie Torgersen        36.7         19.3            193        3450
5  Adelie Torgersen        39.3         20.6            190        3650
6  Adelie Torgersen        38.9         17.8            181        3625
   sex year
1  male 2007
2 female 2007
3 female 2007
4 female 2007
5  male 2007
6 female 2007
```

Names in a dataset

When working with data in R, it's important to know *exactly what things are called*.

What if we just wanted to know the names of the variables that are contained in **penguins**?

```
names(penguins)
```

```
[1] "species"           "island"            "bill_length_mm"  
[4] "bill_depth_mm"    "flipper_length_mm" "body_mass_g"  
[7] "sex"               "year"
```

Naming things in R

When giving things names in R, you can only use a combination of letters, numbers, periods, and underscores, and the names have to start with a letter or a period. People tend to use underscores or periods instead of spaces.

```
tik tok <- 12
```

```
Error: <text>:1:5: unexpected symbol  
1: tik tok  
      ^
```

```
_hi_mom <- 5^2
```

```
Error: <text>:1:1: unexpected input  
1: _  
      ^
```

```
4eva <- 4 * 2
```

```
Error: <text>:1:1: unexpected input  
1: 4ev  
      ^
```

```
tiktok <- 12  
forever <- 4 * 2  
dear_mother <- 5^2
```

Exploring Data

To see a data frame's *structure*, use the function `str()` (pronounced "stir"):

```
str(penguins)
```

```
'data.frame': 333 obs. of 8 variables:  
 $ species      : chr  "Adelie" "Adelie" "Adelie" "Adelie" ...  
 $ island        : chr  "Torgersen" "Torgersen" "Torgersen" "Torgersen" ...  
 $ bill_length_mm: num  39.1 39.5 40.3 36.7 39.3 38.9 39.2 41.1 38.6 34.6 ...  
 $ bill_depth_mm : num  18.7 17.4 18 19.3 20.6 17.8 19.6 17.6 21.2 21.1 ...  
 $ flipper_length_mm: int  181 186 195 193 190 181 195 182 191 198 ...  
 $ body_mass_g   : int  3750 3800 3250 3450 3650 3625 4675 3200 3800 4400 ...  
 $ sex           : chr  "male" "female" "female" "female" ...  
 $ year          : int  2007 2007 2007 2007 2007 2007 2007 2007 2007 2007 ...
```

Exploring Data

If you really only want the "dimension" of the data frame (i.e., how many rows and how many columns), you can use the `dim()` function:

```
dim(penguins)
```

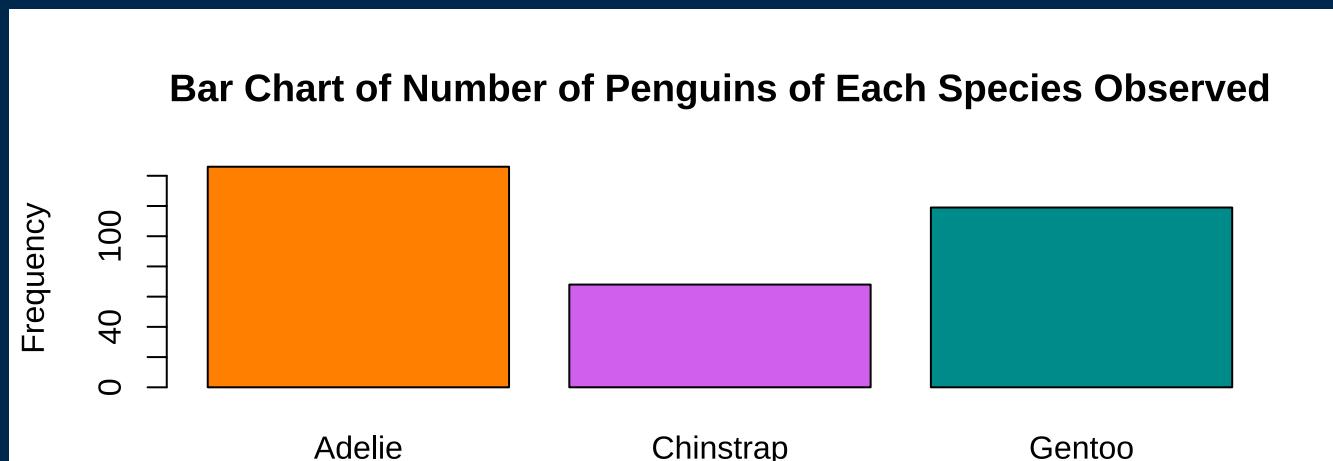
```
[1] 333    8
```

The results are given in the order "rows, columns" because data is **Really Cool** (rows, columns).

Recall: Bar Charts in R

In order to make this plot, we had to give `barplot()` a "frequency table" of the variable **species**.

```
barplot(table(penguins$species),  
       xlab = "Species",  
       ylab = "Frequency",  
       main = "Bar Chart of Number of Penguins of Each Species Observed",  
       col = c("darkorange1", "mediumorchid2", "darkcyan"))
```



Frequency Tables

A *frequency table* is a way to count the number of observations (rows) in the data that correspond to each level of a categorical variable.

To make a frequency table, use the `table()` function

```
table(penguins$species)
```

Adelie	Chinstrap	Gentoo
146	68	119



The dollar sign (\$) is how we tell R where to look for a particular variable.

```
table(penguins$species)
```

Inside **table()**, we need to tell R to look for the variable **species** inside the data.frame called **penguins**. If we don't include **penguins\$**, watch what happens:

```
table(species)
```

Error in table(species): object 'species' not found

Two-Way Frequency Tables

We can make "two-way" frequency tables (sometimes called "contingency tables") to summarize counts for two categorical variables:

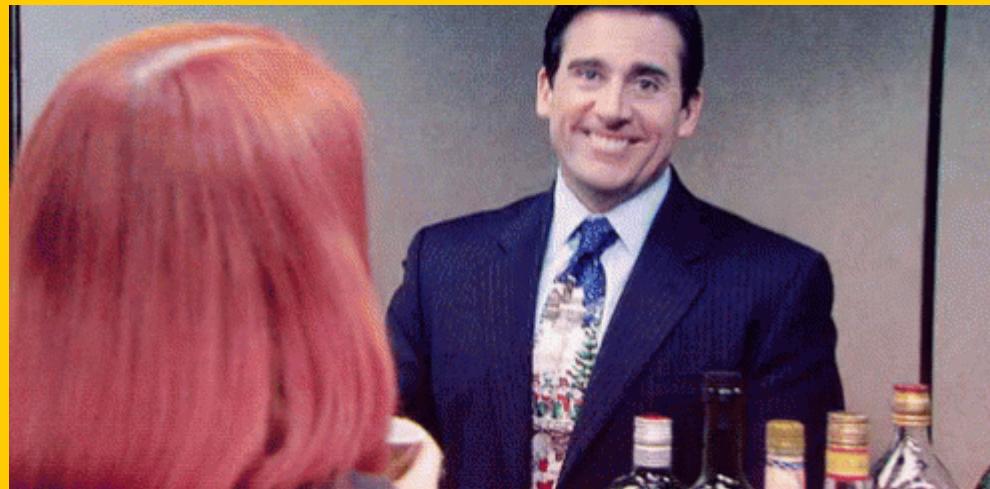
```
table(penguins$species, penguins$island)
```

	Biscoe	Dream	Torgersen
Adelie	44	55	47
Chinstrap	0	68	0
Gentoo	119	0	0

Remember that data is **really cool**, so the first variable you give to `table()` is in the **rows** of the table, and the second is in the **columns**.

Notice that we separated the two variables inside of `table()` with a comma.

Now it's your turn!



Lab Project



Your tasks

- Complete the "Try It!" and "Discussion" portions of the lab assignment by copy/pasting and modifying appropriate code from earlier in the document.

How to get help

- Use the "labs" section of Piazza to ask questions and work with your peers.
- If you use Piazza, please note that in the "Collaborators" list at the top of the Try It section.
- If you're really stuck, email your lab instructor!



Lab Submission: Finding Your Report

Hit the Knit button one last time, then:

RStudio Cloud

1. In the Files pane, check the box next to `lab02report.html`
2. Click More > Export...
3. Click Download and save the file on your computer in a folder you'll remember and be able to find later.

RStudio Desktop (local)

1. Locate the `lab02report.html` file on your computer. The file will be saved in the location indicated at the top of the files pane.



Lab Submission: Canvas (Due 9/11 8a ET)

1. Click the "Assignments" panel on the left side of the page. Scroll to find "Lab 2", and open the assignment. Click "Submit Assignment".
2. Towards the bottom of the page, you'll be able to choose `lab02report.html` from the folder you saved it in from RStudio Cloud or noted if you're using RStudio Desktop. **You will only be able to upload a .html file -- do not upload any other file type.**
3. Click "Submit Assignment". You're done!