Teaching R in DASC 1104

R Part I

John Tipton

The University of Arkansas

2022/06/01 (updated: 2022-05-25)

Materials

- Available on gitHub at https://github.com/jtipton25/dasc-1104-teaching-2022

Guiding principles

- Make the initial burden to working with real data as minimal as possible
- Teach a **process** of working with data more than the actual programming
    - Focus on actions/principles rather than specific syntax
    - More than in other programming classes, focus on reproducibility
- Great resources for learning R/python/git/shell
    - https://software-carpentry.org/lessons/index.html
- Scaffold Learning objectives and provide many examples
    - Here is a dataset -- Generate a scatterplot of mpg vs. engine displacement (demo with `mtcars`)
    - Start with simple examples and show how to build
    - Encourage student participation by working examples alongside

Guiding principles

- Allow for student creativity and spontaneity
  - Different from typical programming classes
  - Here is a dataset -- Create two visualizations that tell me something interesting about it
- Focus on the high-level concepts before getting into the details
  - Some low level knowledge is ok -- variable types
  - Other low level knowledge not needed at this stage
    - Parsing files one line at a time, etc.

The Rstudio IDE

- I recommend the RStudio IDE, but you do not have to.
    - Allows for R and python code development
- Can be installed locally or on a server (I use a SSO server option)
- RStudio Cloud provides a remote-hosting option for teaching
    - Plan to use this resource going forward

Project-oriented workflows

- Especially useful for students with less comp-sci experience
- Makes file path management easier
- References here and here and here

What does this code do?

```
library(tidyverse)
setwd("~/Documents/John/hw-1")
dat <- read_csv("my-secrets.csv")
```

- What might be the issue with this code?

Hands-on experience project management in Rstudio

- Use of Rstudio projects (5-10 minutes)
- Use of the `here` library and the `here()` function
- Namespaces in `R`
- Generate "data file" in excel, load and process the data
- For more details, see Lecture 9 in unit 4

The Two Towers

- The R language has split into two dialects "base R" and the "tidyverse"
- Much has been written about this difference
- Having learned Base R first and `tidyverse` second -- I am a strong proponent of the `tidyverse` -- there is a strong movement to translate this syntax to python
    - Most of my students would agree that `tidyverse` is the way to go
    - Many python users also agree

The Rings of Power

```
library(tidyverse)
library(palmerpenguins)
glimpse(penguins)
```

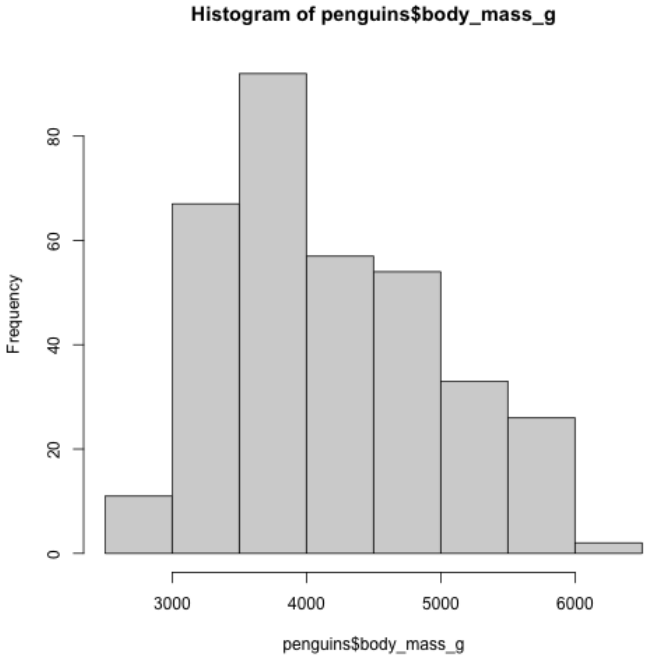
```
## Rows: 344
## Columns: 8
## $ species           <fct> Adelie, Adelie, Adelie, Adelie, Adelie, Adelie, Adelie, Adelie, Adelie
## $ island            <fct> Torgersen, Torgersen, Torgersen, Torgersen, Torgersen, Torgersen, Torg
## $ bill_length_mm    <dbl> 39.1, 39.5, 40.3, NA, 36.7, 39.3, 38.9, 39.2, 34.1, 42.0, 37.8, 37.8,
## $ bill_depth_mm     <dbl> 18.7, 17.4, 18.0, NA, 19.3, 20.6, 17.8, 19.6, 18.1, 20.2, 17.1, 17.3,
## $ flipper_length_mm <int> 181, 186, 195, NA, 193, 190, 181, 195, 193, 190, 186, 180, 182, 191, 1
## $ body_mass_g       <int> 3750, 3800, 3250, NA, 3450, 3650, 3625, 4675, 3475, 4250, 3300, 3700,
## $ sex               <fct> male, female, female, NA, female, male, female, male, NA, NA, NA, NA,
## $ year              <int> 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007
```

The Rings of Power

- Create a histogram of `body_mass_g`

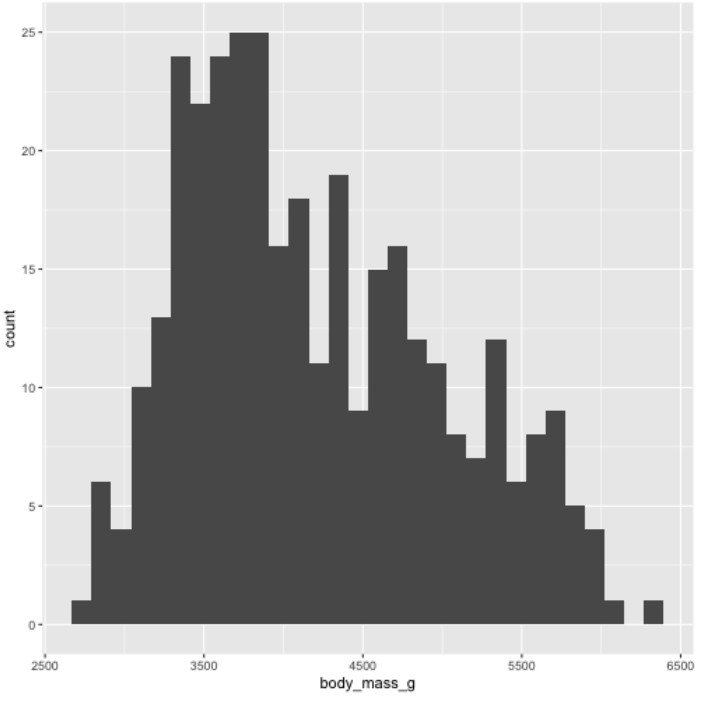
- Using base R

```
hist(penguins$body_mass_g)
```



- Using `tidyverse`

```
ggplot(penguins, aes(x = body_mass_g)) +
  geom_histogram()
```

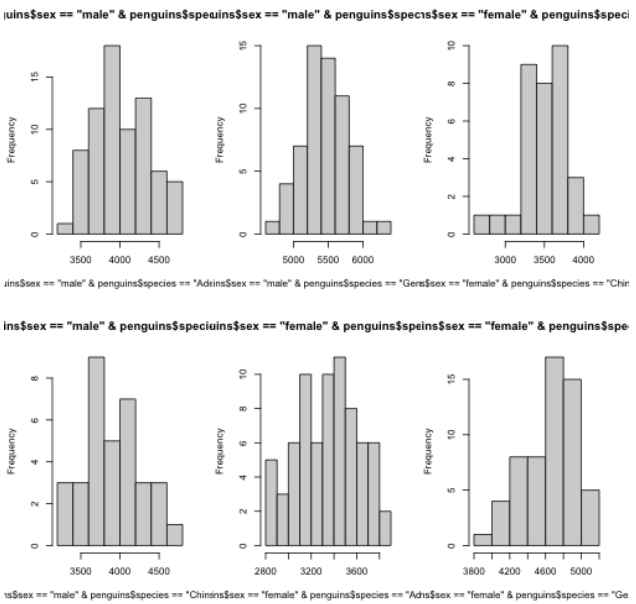


But they were all of them decieved

- Create a histogram of `body_mass_g` for each `species` and `sex`

- Using base R

```
layout(matrix(1:6, 2, 3))
hist(penguins[penguins$sex == "male" & penguins$species =
hist(penguins[penguins$sex == "male" & penguins$species =
hist(penguins[penguins$sex == "male" & penguins$species =
hist(penguins[penguins$sex == "female" & penguins$species
hist(penguins[penguins$sex == "female" & penguins$species
hist(penguins[penguins$sex == "female" & penguins$species
```



- Using `tidyverse`

```
ggplot(penguins, aes(x = body_mass_g)) +
  geom_histogram() +
  facet_grid(sex ~ species)
```

For in secret the Dark Lord Sauron forged in secret a Master Ring, to control all others.

- How many penguins are in the dataset of each `sex` and `species`?

```
with(penguins, table(sex, species))
```

```
##         species
## sex      Adelie Chinstrap Gentoo
##   female     73        34     58
##   male       73        34     61
```

```
penguins %>%
  count(sex, species)
```

```
## # A tibble: 8 × 3
##   sex    species       n
##   <fct>  <fct>     <int>
## 1 female Adelie       73
## 2 female Chinstrap    34
## 3 female Gentoo       58
## 4 male   Adelie       73
## 5 male   Chinstrap    34
## 6 male   Gentoo       61
## 7 <NA>   Adelie        6
## 8 <NA>   Gentoo        5
```

- Where did the `NA` values go?

One ring to rule them all!

- What does this code do?

```
aggregate(airquality[, "Ozone"],
          list(Month = airquality[, "Month"]),
          mean, na.rm = TRUE)
```

```
##   Month        x
## 1     5 23.61538
## 2     6 29.44444
## 3     7 59.11538
## 4     8 59.96154
## 5     9 31.44828
```

```
airquality %>%
  group_by(Month) %>%
    summarize(mean_o3 = mean(Ozone, na.rm = TRUE))
```

```
## # A tibble: 5 × 2
##   Month mean_o3
##   <int>   <dbl>
## 1     5    23.6
## 2     6    29.4
## 3     7    59.1
## 4     8    60.0
## 5     9    31.4
```

- Which example is more **human readable?**

Working with RMarkdown

- Demo with RMarkdown examples (15-20 minutes)

- Code chunks
- Running code (ctrl-enter, run all above)
- installing vs. loading packages
- Restarting the R environment and clearing variables
    - Do this often
- **Settings**--don't save your workspace
- Plain text, math equations, and communication
- Code chunk options
- File paths and loading data
- Managing the project environment
- Formatting and troubleshooting
- The tinytex/latex problem
    - `install.packages('tinytex')
    - `tinytex::install_tinytex()`
- python code in RMarkdown
- sourcing in scripts
- live preview (I'm learning this in real time)
- Getting help with `?`

What to teach

- Programming concepts are important
  - loops, data structures, functions
- But, I **don't** teach R as if it were a programming language
  - R is a programming language for **data science**
  - Many "real" programmers don't like R because it is not a "progamming first" language
    - R is a **data-science** first language, don't get mad when it isn't what you want it to be
  - Show how to do data science using R as a tool
- Lead with data visualization, follow up with data manipulation

Explore for meaningful datasets to use

1) Tidy Tuesday

2) FiveThirtyEight

3) BuzzFeed News

4) Kaggle

5) UCI Machine Learning Repository

Focus on basic syntax and datatypes (demo)

Typically in class, I include code in the slides, but not here

- vectors
- matrices
- arrays
- lists
- factors
- data.frames

Atomic types (demo)

- logical/boolean
- numeric
- integer
- complex (not covered)
- character/string

Working with data

- what are data.frames?
- accessing rows/columns with `[`
- accessing variables with `[`
- accessing variables with `$`