TODAY'S CLASS

Part 1: Working in a Reproducible Environment:

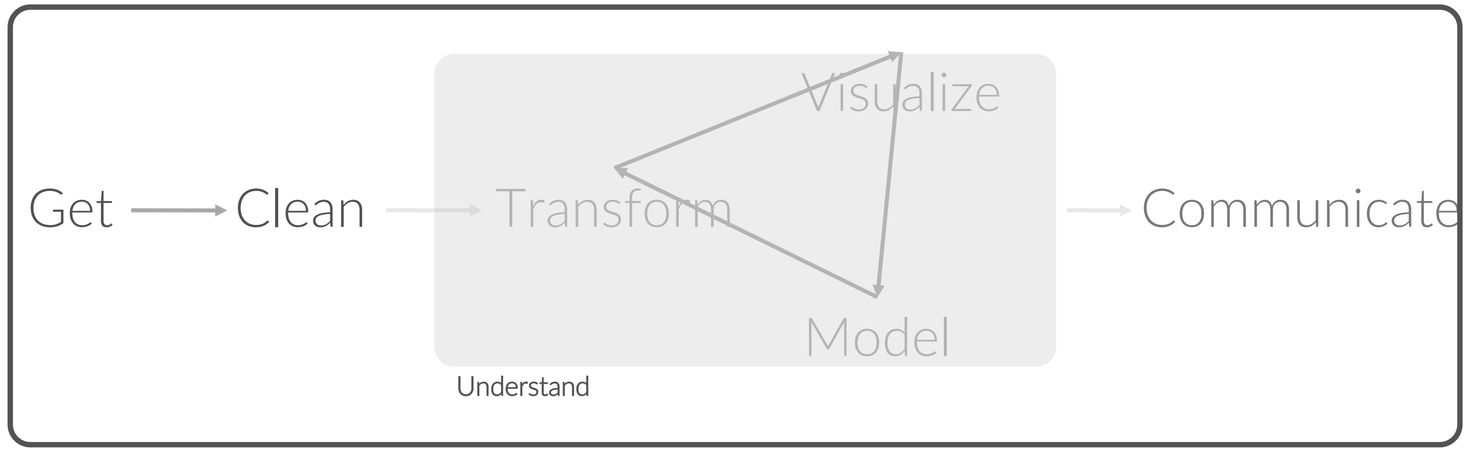
- ❑ R Projects, R Markdown, R Notebooks
- ❑ Creating and editing R Markdown files

Part 2: Importing Data with Base R and the Tidyverse

Part 3: Coding Exercises (importing data to begin your midterm project)

FIRST DATE GUIDELINES...

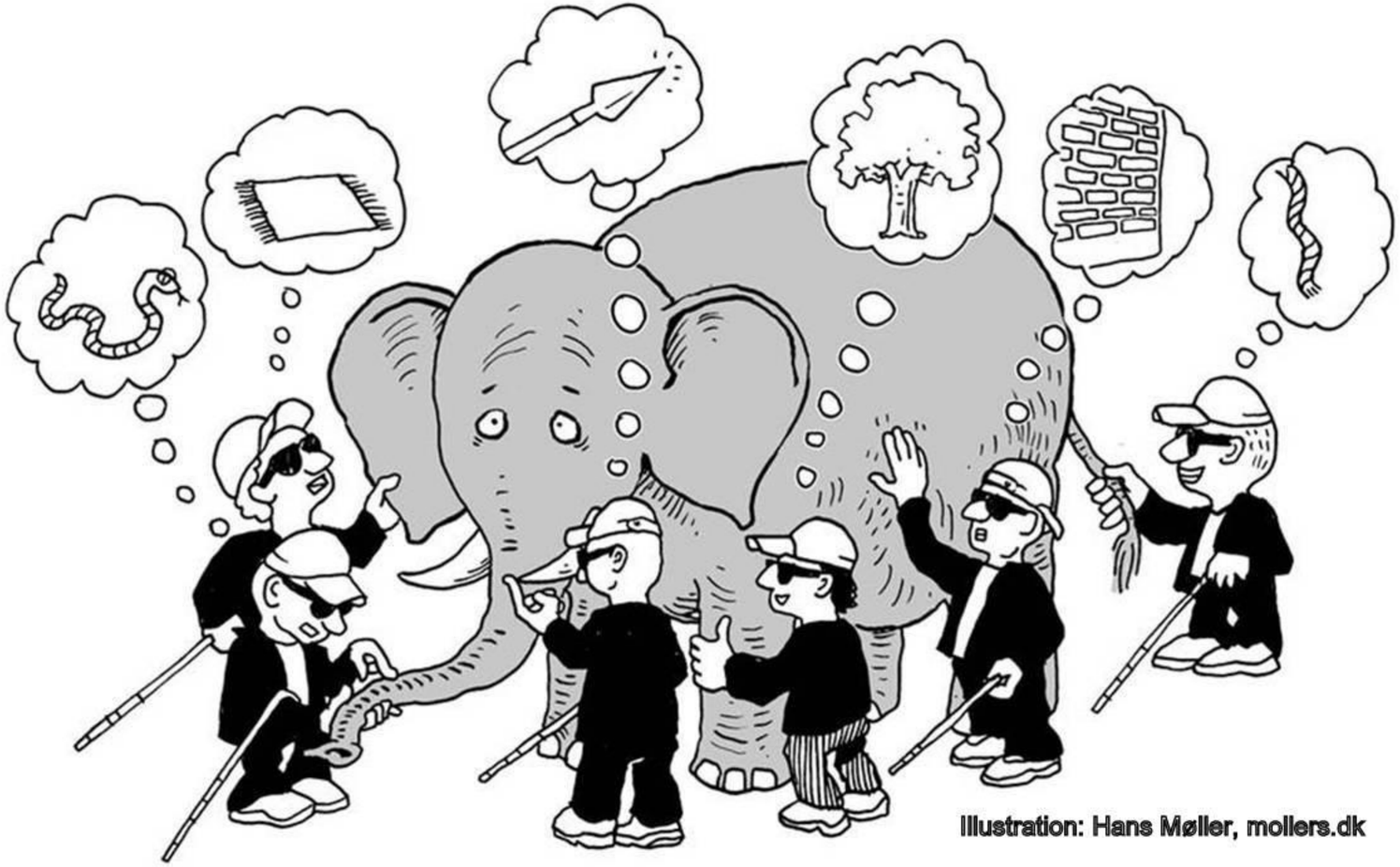
for data



†A modified version of Hadley Wickham's analytic process

“What we have is a data glut.”

- Vernon Vinge



“Unless the data is something I’ve analyzed a lot before, I usually feel like the blind men and the elephant.”

- Jeff Leek

FIRST THINGS TO DO

Don’t try to kiss your data on the first date; rather, you just want to get to know the data:

1. ?
2. ?
3. ?
4. ?

FIRST THINGS TO DO

Don’t try to kiss your data on the first date; rather, you just want to get to know the data:

1. Import the data
2. Review the codebook
3. Learn about the data
4. Quick visual understanding of the data

DATA IMPORT

FLAT FILES

What are the main functions to read in a **.csv** file?

Function	Package
?	?
?	?
?	?

FLAT FILES

What are the main functions to read in a **.csv** file?

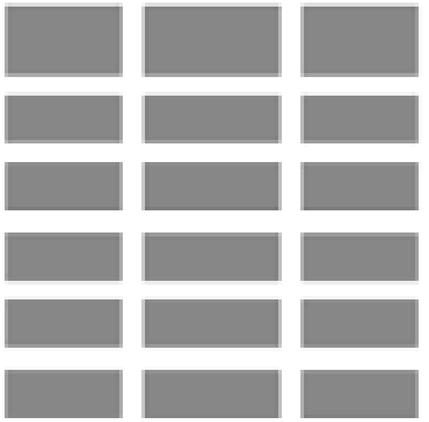
Function	Package
read.csv	utils (Base R)
read_csv	readr
fread	data.table

FLAT FILES

How are these functions the same?

csv

read.csv(“mydata.csv”)
read_csv(“mydata.csv”)
fread(“mydata.csv”)



data frame

FLAT FILES

How do these functions differ?

FLAT FILES

How do these functions differ?

1. Base R vs. Packages

```
# Base R
read.csv(“data/flights.csv”)

# Load package readr
library(readr)
readr::read_csv(“data/flights.csv”)

# Load package data.table
library(data.table)
fread(“data/flights.csv”)
```

FLAT FILES

How do these functions differ?

1. Base R vs. Packages
2. Speed

```
# Slowest
read.csv("data/flights.csv")
```

~4 seconds

```
# Faster
library(readr)
readr::read_csv("data/flights.csv")
```

~0.8 seconds

```
# Fastest
library(data.table)
fread("data/flights.csv")
```

~0.3 seconds

FLAT FILES

How do these functions differ?

1. Base R vs. Packages
2. Speed
3. Variable attributes

```
# read.csv defaults to saving character variables as factors
df <- read.csv("data/flights.csv")
str(df)
'data.frame': 336776 obs. of  19 variables:
 $ year          : int  2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 ...
 $ month         : int  1 1 1 1 1 1 1 1 1 1 ...
 $ day           : int  1 1 1 1 1 1 1 1 1 1 ...
 $ dep_time      : int  517 533 542 544 554 554 555 557 557 558 ...
 $ sched_dep_time: int  515 529 540 545 600 558 600 600 600 600 ...
 $ dep_delay     : num  2 4 2 -1 -6 -4 -5 -3 -3 -2 ...
 $ arr_time      : int  830 850 923 1004 812 740 913 709 838 753 ...
 $ sched_arr_time: int  819 830 850 1022 837 728 854 723 846 745 ...
 $ arr_delay     : num  11 20 33 -18 -25 12 19 -14 -8 8 ...
 $ carrier       : Factor w/ 16 levels "9E","AA","AS",..: 12 12 2 4 5 12 4 6 4
 $ flight        : int  1545 1714 1141 725 461 1696 507 5708 79 301 ...
 $ tailnum       : Factor w/ 4043 levels "D942DN","N0EGMQ",..: 180 524 2401
 $ origin        : Factor w/ 3 levels "EWR","JFK","LGA": 1 3 2 2 3 1 1 3 2 3 ...
```

FLAT FILES

How do these functions differ?

1. Base R vs. Packages
2. Speed
3. Variable attributes

```
# read.csv defaults to saving character variables as factors
df <- read.csv("data/flights.csv", stringsAsFactors = FALSE)
str(df)
'data.frame': 336776 obs. of  19 variables:
 $ year          : int  2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 ...
 $ month         : int  1 1 1 1 1 1 1 1 1 1 ...
 $ day           : int  1 1 1 1 1 1 1 1 1 1 ...
 $ dep_time      : int  517 533 542 544 554 554 555 557 557 558 ...
 $ sched_dep_time: int  515 529 540 545 600 558 600 600 600 600 ...
 $ dep_delay     : num  2 4 2 -1 -6 -4 -5 -3 -3 -2 ...
 $ arr_time      : int  830 850 923 1004 812 740 913 709 838 753 ...
 $ sched_arr_time: int  819 830 850 1022 837 728 854 723 846 745 ...
 $ arr_delay     : num  11 20 33 -18 -25 12 19 -14 -8 8 ...
 $ carrier       : chr  "UA" "UA" "AA" "B6" ...
 $ flight        : int  1545 1714 1141 725 461 1696 507 5708 79 301 ...
 $ tailnum       : chr  "N14228" "N24211" "N619AA" "N804JB" ...
 $ origin        : chr  "EWR" "LGA" "JFK" "JFK" ...
```

FLAT FILES

How do these functions differ?

1. Base R vs. Packages
2. Speed
3. Variable attributes

Both **read_csv** & **fread** default to **stringsAsFactors = FALSE**

```
# read.csv defaults to saving character variables as factors
df <- read.csv("data/flights.csv", stringsAsFactors = FALSE)
str(df)
'data.frame': 336776 obs. of  19 variables:
 $ year          : int  2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 ...
 $ month         : int  1 1 1 1 1 1 1 1 1 1 ...
 $ day           : int  1 1 1 1 1 1 1 1 1 1 ...
 $ dep_time      : int  517 533 542 544 554 554 555 557 557 558 ...
 $ sched_dep_time: int  515 529 540 545 600 558 600 600 600 600 ...
 $ dep_delay     : num  2 4 2 -1 -6 -4 -5 -3 -3 -2 ...
 $ arr_time      : int  830 850 923 1004 812 740 913 709 838 753 ...
 $ sched_arr_time: int  819 830 850 1022 837 728 854 723 846 745 ...
 $ arr_delay     : num  11 20 33 -18 -25 12 19 -14 -8 8 ...
 $ carrier       : chr  "UA" "UA" "AA" "B6" ...
 $ flight        : int  1545 1714 1141 725 461 1696 507 5708 79 301 ...
 $ tailnum       : chr  "N14228" "N24211" "N619AA" "N804JB" ...
 $ origin        : chr  "EWR" "LGA" "JFK" "JFK" ...
```

WHAT ABOUT OTHER FLAT FILES?

Package	Function	.CSV	.TSV	.TXT	FIXED WIDTH	SPECIAL SEPARATOR
utils (Base R)	read.csv	x				
	read.delim		x			
	read.table			x		x
readr	read_csv	x				
	read_tsv		x			
	read_table			x	x	
	read_fwf				x	
	read_delim					x
data.table	fread	x	x	x	x	x

WHAT ABOUT EXCEL FILES?

readxl TO READ IN EXCEL FILES

Function	Description
?	preview sheet names in specified Excel file
?	reads in excel file
?	write to UTF-8 encoded .csv

readxl TO READ IN EXCEL FILES

Function	Description
excel_sheets	preview sheet names in specified Excel file
read_excel	reads in excel file
readr::write_excel_csv	write to UTF-8 encoded .csv
Other packages allow you to write directly to, and format, Excel files. You can learn these on your own time.	

READING IN A FILE

Excel is still the spreadsheet software of choice

You need to understand both the workbook *and* the sheet that you want to read in

```
# identify the sheet you want
excel_sheets(“data/mydata.xlsx")
[1] "PICK_ME_FIRST!" "Sheet2"        "extra_header"   "functions"
[5] "date_time"      “unique_NA”
```

READING IN A FILE

Excel is still the spreadsheet software of choice

You need to understand both the workbook *and* the sheet that you want to read in

```
# identify the sheet you want
excel_sheets(“data/mydata.xlsx")
[1] "PICK_ME_FIRST!" "Sheet2"        "extra_header"   "functions"
[5] "date_time"      “unique_NA"

# now read in the data
read_excel("data/mydata.xlsx", sheet = "PICK_ME_FIRST!")
# A tibble: 3 × 3
  `variable 1` `variable 2` `variable 3`
         <dbl>      <chr>         <dbl>
1          10       beer             1
2          25       wine             1
3           8     cheese             0
```

OTHER OPTIONS

GDATA

XLConnect

ROBC

RExcel

All require Java or Perl dependencies

CODEBOOK

THE CODEBOOK

- A codebook is a technical description of the data that was collected for a particular purpose
- Should be the first thing you review before any kind of analysis

Understanding the source data is crucial to any analysis.

WHAT

- Original source of the data
- How the data are generated
- Variable definitions
- Missing data encodings

WHY

- Organizational data often represent multiple truths
- Organizations are increasingly integrating internal and external data sources
- Analysts need to understand what assumptions or nuances are represented by the data

Understand what you are analyzing!

IMPORTANT

This aligns with project standards: 3.1 & 3.2

LEARN ABOUT YOUR DATA

LEARN ABOUT THE DATA

- So what are the first things we want to know about our data?
    - ?
    - ?
    - ?
    - ?

LEARN ABOUT THE DATA

- So what are the first things we want to know about our data?
    - dimensions
    - data types (i.e. character, integer, factor, etc.)
    - missing values
    - summary statistics

What are some functions to extract this information?

LEARN ABOUT THE DATA

- So what are the first things we want to know about our data?
    - dimensions: **dim(), ncol(), nrow(), names()**
    - data types: **str(), class(), is., as.**
    - missing values: **is.na(), sum(is.na()), colSums(is.na())**
    - summary statistics: **summary(), quantile(), var(), sd(), table()**

REPORTING

- Reporting this information in your data dictionary/preparation section is important:
    - How many missing values did you remove or impute
    - The range and most likely values for each variable
    - Example values seen in each variable
    - It's good practice to show at least the first few lines of your data so the reader can get a feel for the data: (i.e. **View(), head()**)

IMPORTANT

This aligns with project standards: 3.3-3.5

GET TO KNOW VISUALLY

LEARN ABOUT THE DATA

For quick data exploration, base R plotting functions can provide an expeditious and straightforward approach to understanding your data

What are some functions to extract this information?

QUICK PLOTS

Function	Description
?	scatter plot
?	line chart
?	bar chart
?	histogram
?	box plot
?	stem & leaf plot

QUICK PLOTS

Function	Description
plot(x, y)	scatter plot
plot(x, y, type = “l”)	line chart
barplot(table(x))	bar chart
hist(x)	histogram
boxplot(y ~ x, data)	box plot
stem(x)	stem & leaf plot

IMPORTANT

This somewhat aligns to standard 4.2; however, in week 5 you will learn to create much more meaningful visualizations.

TIME TO FLEX OUR NEW KNOWLEDGE MUSCLES!

Let’s review and apply what you’ve learned a use-case

GROUP CHALLENGE

EXCEL FILES

xxxx.xlsx

IMPORT

1. What spreadsheets are contained in the aircraft.xlsx file?
2. Read in the Trainers worksheet data without the header information.

SOLUTION

```
# 1. What spreadsheets are contained in the aircraft.xlsx file?

library(readxl)
excel_sheets("data/aircraft.xlsx")
[1] "Bombers"              "Fighters"             "Trainers"
[4] "UAV_Drones"           "Tankers_Transporters"
```

SOLUTION

```
# 2. Read in the Trainers worksheet data without the header information.

aircraft <- read_excel("data/aircraft.xlsx", sheet = "Trainers", skip = 3)
aircraft
# A tibble: 187 x 6
     Type    MD    FY    FH  Gallons     Cost
    <chr> <chr> <dbl> <dbl>    <dbl>    <dbl>
 1 Trainer AT-38  1996 12517  6681614  5641569
 2 Trainer AT-38  1997 11656  7707001  6506680
 3 Trainer AT-38  1998 12619  9749881  9526089
 4 Trainer AT-38  1999 13132 10534024  9343636
 5 Trainer AT-38  2000 14400 10769237  7242603
 6 Trainer AT-38  2001 12674  9680191 10533477
 7 Trainer AT-38  2002  1143  4107850  4813961
 8 Trainer AT-38  2003 -1561  2793119  2953701
 9 Trainer AT-38  2004 -1972  2579471  2995067
10 Trainer AT-38  2005 -1713  2727529  4996524
```

NUMERIC UNDERSTANDING

3. Look at the summary information for this data. Anything seem odd?
4. Which aircraft **MD**s are represented?
5. Are there any missing years between 1996-2014 in this data?
6. 90% of flying hours fall under what value?
7. What is the spread of the range of costs?

SOLUTION

3. Look at the summary information for this data. Anything seem odd??

```
summary(aircraft)
    Type               MD                 FY           FH
 Length:187         Length:187         Min.   :1996   Min.   :-62827
 Class :character   Class :character   1st Qu.:2001   1st Qu.:   679
 Mode  :character   Mode  :character   Median :2006   Median :  1788
                                       Mean   :2006   Mean   : 26929
                                       3rd Qu.:2011   3rd Qu.: 33090
                                       Max.   :2014   Max.   :147928
   Gallons             Cost
 Min.   : -373147   Min.   :     567
 1st Qu.:   11450   1st Qu.:  106344
 Median :  352201   Median :  1616098
 Mean   : 8646004   Mean   : 15554466
 3rd Qu.:10526094   3rd Qu.: 11166048
 Max.   :62154004   Max.   :158058171
```

SOLUTION

4. Which aircraft MDs are represented?

```
table(aircraft$MD)
 AT-38   T-1   T-38   T-41   T-51   T-53    T-6 TC-130 TC-135   TH-1   TU-2
    19    19    19    19    15     4    15    13    19     7    19
 UV-18
    19
```

SOLUTION

5. Are there any missing years between 1996-2014 in this data?

unique(aircraft$FY)

```
 [1] 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010
[16] 2011 2012 2013 2014
```

table(aircraft$FY)

```
1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011
   7    7    8    8    9    9   10    9    9   10   10   10   11   11   11   12
2012 2013 2014
  12   12   12
```

SOLUTION

```
# 6. 90% of flying hours fall under what value?

quantile(aircraft$FH, prob = .9)
    90%
109828.8
```

SOLUTION

```
# 7. What is the spread of the range of costs?

# range
range(aircraft$Cost)
[1]       567 158058171

# spread
max(aircraft$Cost) - min(aircraft$Cost)
[1] 158057604

# or...

diff(range(aircraft$Cost))
[1] 158057604
```

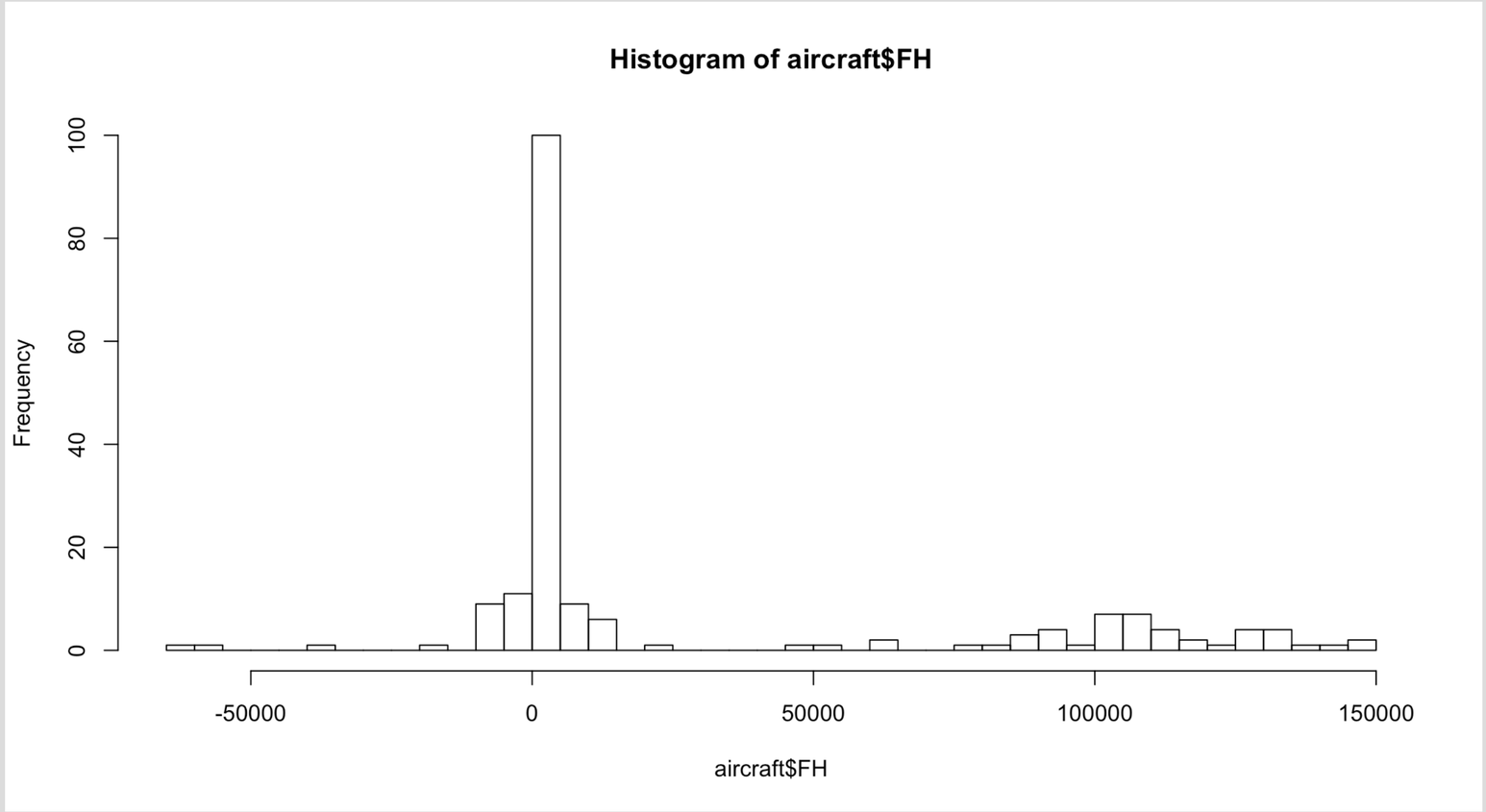
VISUAL UNDERSTANDING

8. How would you describe the distribution of flying hours?
9. If we wanted to focus on only the trainers with the largest variance in flying hours, which MDs would we select?
10. Are all FYs equally represented?

SOLUTION

8. How would you describe the distribution of flying hours?

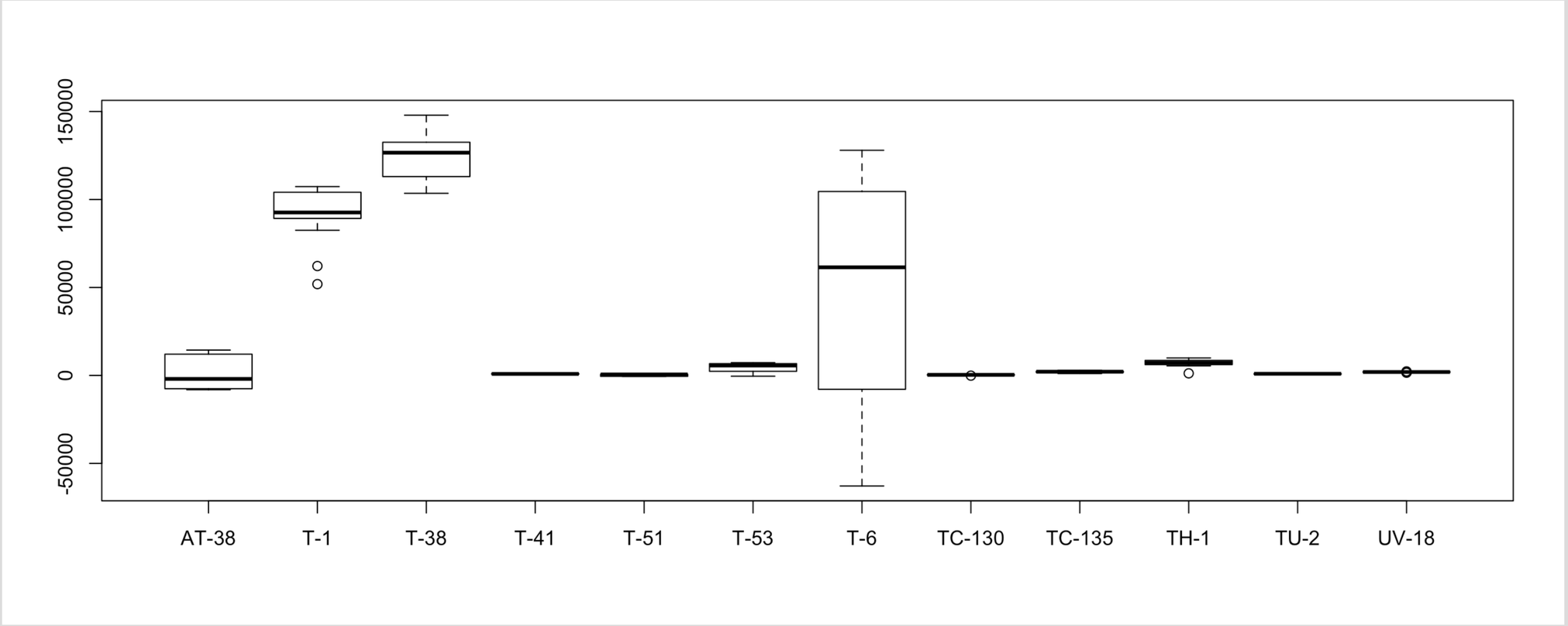
hist(aircraft$FH, breaks = 50)



SOLUTION

9. If we wanted to focus on only the trainers with the largest variance in
flying hours, which MDs would we select?

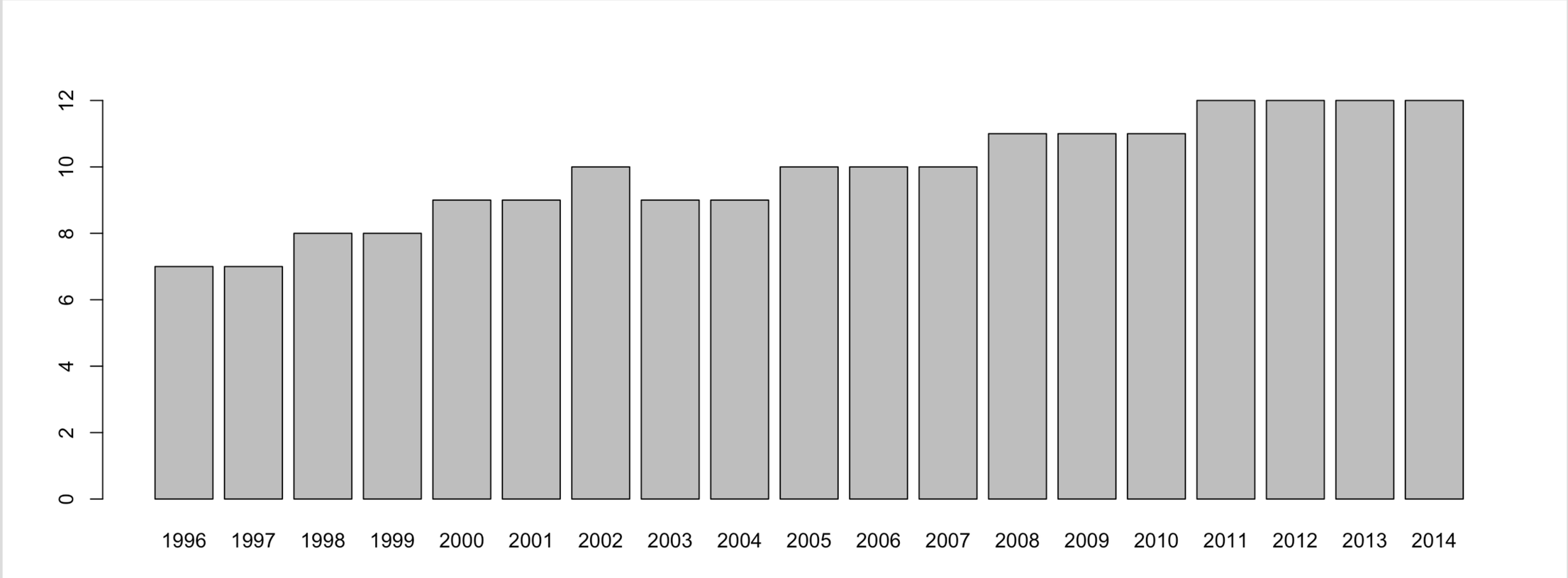
boxplot(FH ~ MD, data = aircraft)



SOLUTION

10. Are all FYs equally represented?

barplot(table(aircraft$FY))



ONLINE FILE

www.something.com

TASKS

1. Scrape the Cincinnati weather data: http://academic.udayton.edu/kissock/http/Weather/gsod95-current/OHCINCIN.txt
2. What is the average temperature for Cincinnati?
3. What has been the hottest and coldest temperature? Does this seem odd?
4. Recode the missing values (-99) to NA. Now what is the max, min, and mean temperature?
5. Look at the distribution of temperatures by month. What kind of pattern do you see?

SOLUTION

```
# 1. Scrape the Cincinnati weather data:
# http://academic.udayton.edu/kissock/http/Weather/gsod95-current/OHCINCIN.txt

library(readr)
url <- "http://academic.udayton.edu/kissock/http/Weather/gsod95-current/OHCINCIN.txt"
weather <- read_table(url, col_names = c("Month", "Day", "Year", "Temp"))
weather
# A tibble: 8,327 x 4
   Month   Day  Year  Temp
   <int> <int> <int> <dbl>
 1     1     1  1995  41.1
 2     1     2  1995  22.2
 3     1     3  1995  22.8
 4     1     4  1995  14.9
 5     1     5  1995   9.5
 6     1     6  1995  23.8
 7     1     7  1995  31.1
```

SOLUTION

```
# 2. What is the average temperature for Cincinnati?

mean(weather$Temp)
[1] 54.35007
```

SOLUTION

3. What has been the hottest and coldest temperature? Does this seem odd?

```
summary(weather$Temp)
 Min.   1st Qu.  Median    Mean 3rd Qu.    Max.
-99.00   39.90   56.70   54.35   70.80   89.20
```

SOLUTION

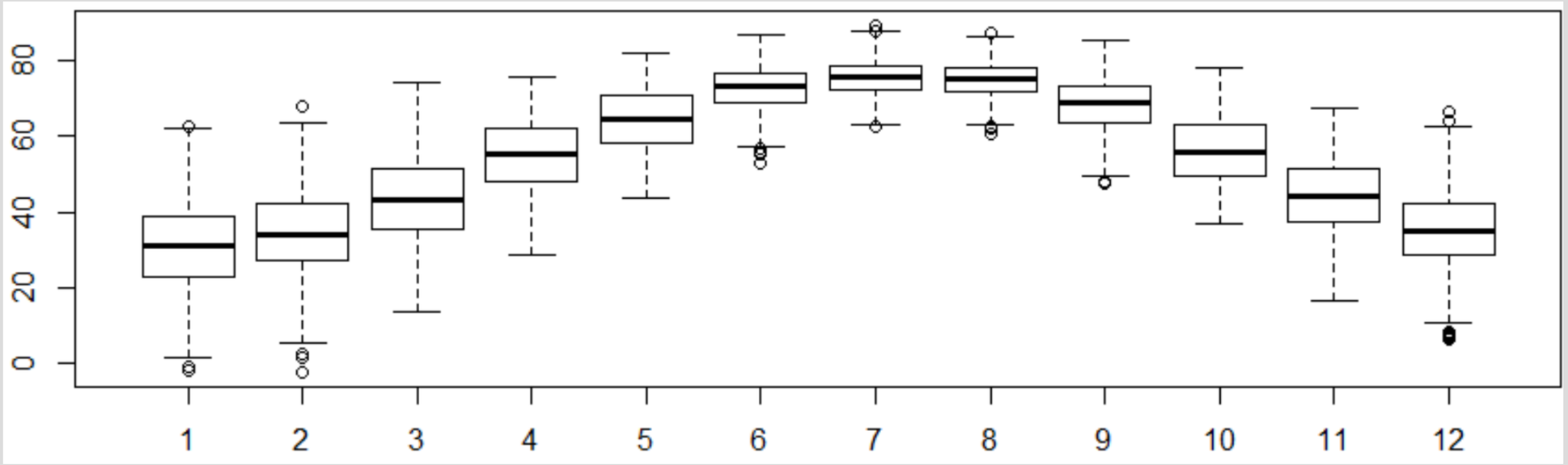
```
# 4. Recode the missing values (-99) to NA. Now what is the max, min, and mean
#    temperature?


weather$Temp[weather$Temp == -99] <- NA
summary(weather$Temp)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
  -2.20   40.00   56.80   54.59   70.80   89.20      14
```

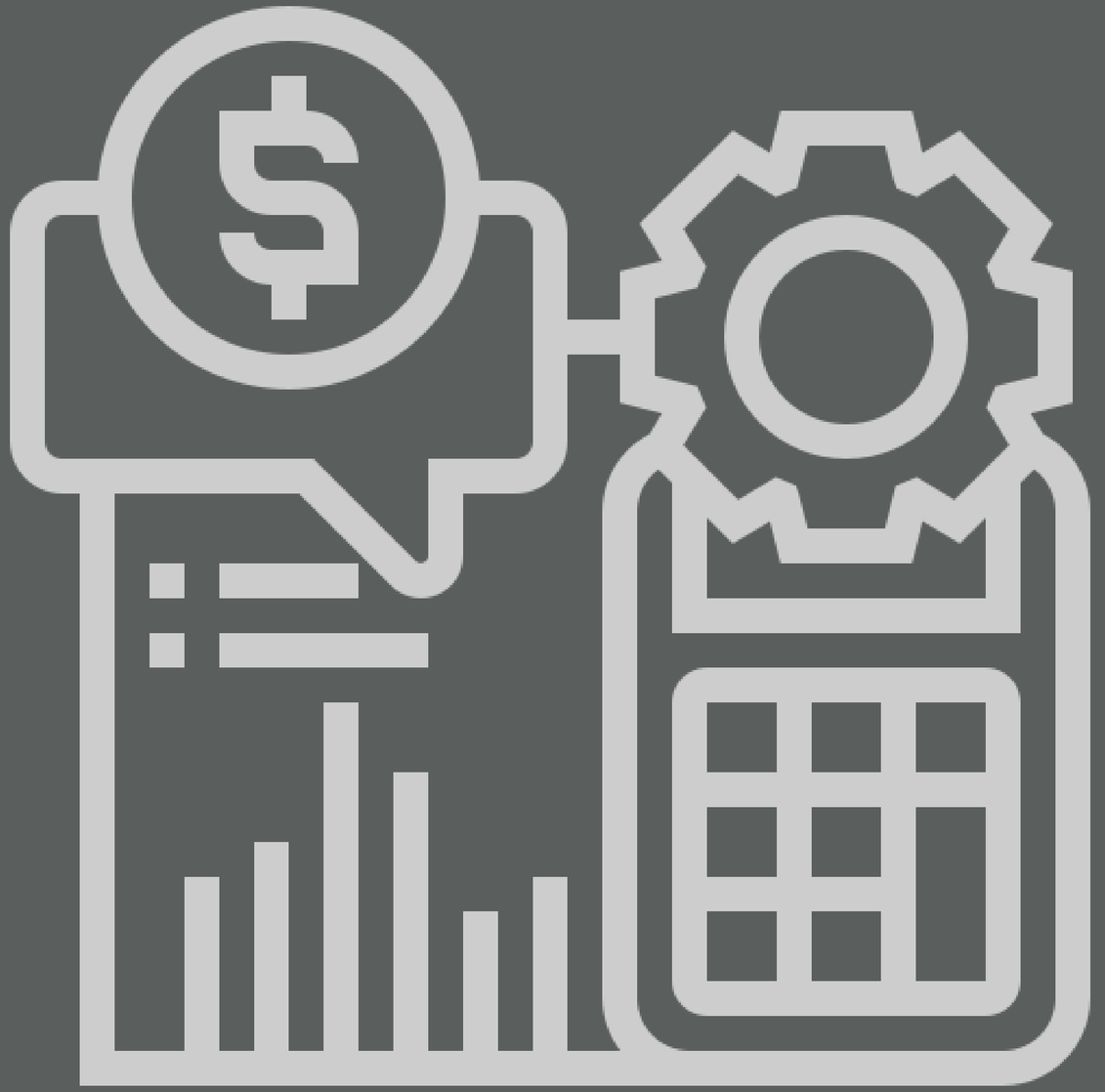
SOLUTION

```
# 5. Look at the distribution of temperatures by month. What kind of pattern do
#    you see?

boxplot(Temp ~ Month, data = weather)
```



MIDTERM PROJECT



MIDTERM PROJECT: HIGHLIGHTS

- Work by yourself or with one other person
  - If working as a pair, the other person does not need to come from your homework group (i.e., you can pick anyone else)
  - You will determine your individual/pair status in your homework.
  - Can't switch/change individual/pair status after you declare this week

- Purpose of midterm
  - Select data set, clean it, state your plan of action to deliver insights from your data that will turn into actions
  - Halfway checkpoint to prepare for the final project (the finished product)

- Purpose of final
  - Tell the story that you've planned from the midterm!
  - What actions can your audience take as a results of your insights?

- You submit your midterm as a .Rmd file in Slack.

MIDTERM PROJECT: FAQs

- Where are the data sets?
  - On the course website!

- How many data sets will I select?
  - One data set! Use the "Download here" link instead of downloading directly from GitHub.

- What is an insight?
  - Knowledge and wisdom gained from analyzing data (visualizations, summaries, tables, quick counts)
  - Never underestimate the power of a simple statistic or visualization. These are almost always easier to take action on than machine learning models.

- Am I required to do any machine learning?
  - NO! Machine learning is OPTIONAL. You will complete all EDA first before even *thinking* about machine learning.

- Should I include this final project in my portfolio of work after this semester?
  - YES!

FOR NEXT WEEK

READING AND HOMEWORK

- Check the course website (Week 2 page) for readings to complete before next week's class.
  - ❑ "Because reading is what? FUNDAMENTAL." -- *RuPaul*

- Homework #2 is FAST!
  - ➢ Submit it in Canvas.
  - ➢ Read the assignment for details.
  - ➢ You (and your partner, if applicable) should start your midterm project ASAP. There is a lot to complete.
    - ✓ Expect to spend several hours cleaning, preparing, and storyboarding for your midterm.

CODING EXERCISES

GROUP CODING EXERCISES

To get you comfortable with the procedures an analyst takes to get to know their data, spend the next 60 minutes working through the tasks in the *“Coding exercises”* PDF in the class download folder.