hw02: Data Sphinx!

STAT 385, Fall 2018

Due: Friday, September 14th, 2018 at 6:00 PM

Warning! This document contains 18 failing tests.

Overview

Please see the homework policy for detailed instructions and some grading notes. Failure to follow instructions will result in point reductions. In particular, make sure to commit each exercise as you complete them.

> "Talk is cheap. Show me the code."
>
> — **Linus Torvalds**, on lkml August 25th, 2000

Objectives

The objectives behind this homework assignment are as follows:

- Reading in Data;
- Vectorization;
- Recycling;
- Writing Functions;
- Using R Packages

Grading

The rubric CAs will use to grade this assignment is:

Task	pts
Link to GitHub Repository	2
At least one commit per exercise (more is better!)	5
Commit messages that describe what changed	5
I *vant* to read your data	10
The Recycler	17
Scientists Rock!	8
Winter is Coming	10
Excellency at UIUC	14
Total	71

Note on Markdown

If you need help with markdown syntax, please go to the "Help" menu and select the *Markdown Quick Reference* guide. This will open in the **Help** tab on the *lower-right* corner of *RStudio*. For more examples, please see the literate programming slides and the in class examples of writing in *RMarkdown*.

In addition, the following two RStudio Cheatsheets will be helpful:

- R Markdown Cheat Sheet
- R Markdown Reference Guide

Both of these guides will be given to you during your time in the CBTF.

Package usage

For this homework assignment, you may only use the following *R* packages:

```
pkg_list = c("ggplot2", "rmarkdown", "readxl", "nasaweather")
```

Assignment

Collaborators

If you worked with any other student in preparing these answers, please make sure to list their full names and NetIDs (e.g. `FirstName LastName (NetID)`).

(12 Points) Exercise 0: Knowledge is Power

- [**2 Points**] **(a)** Place a link to your `hw02` GitHub repository here.
- [**5 Points**] **(b)** Commit every exercise as you finish them. There are *five* exercises (including this one).
- [**5 Points**] **(c)** Make each commit message *meaningful*.
  - The bare minimum for a "meaningful" commit is a length of 15 characters.
  - Inside the commit message, please make sure to appropriately describe what is happening. - Simply stating, "Exercise 3" or "Ex3" is not sufficient.
  - Provide detail on what *changed*.

(10 Points) Exercise 1: I *vant* to read your data!

[**5 Points**] **(a)** The Division of Management Information (DMI) is trying to use *R* to understand the Freshman Profile of Fall 2018. They are having difficult time importing the data into *R* and creating a visualization. Help them out by:

- [**3 Points**] translate the data given below into an *R* `data.frame` called `fa17_enrolled`;
- [**1 Point**] cleaning up the visualization code; and
- [**1 Point**] allow the visualization code chunk to execute.

Hint: Is `ggplot()` part of base R or is it an extension package to *R*?

```
Year  NEnrolled
1978    1
1990    1
1991    1
1993    2
1995    1
1996    7
1997    102
1998    2496
```

```
1999    4821
2000    84
2001    2
```

```
g = ggplot(<data-set-here>,
       aes(x = Year, y = NEnrolled)) +
  geom_col()

g
```

<span class="testrmd-badge label label-danger" data-toggle="collapse" data-target="#testrmd-chunk-766

```
target_df = readRDS("test/fa2018-enrolled.rds")
```

```
## Warning in gzfile(file, "rb"): cannot open compressed file 'test/fa2018-
## enrolled.rds', probable reason 'No such file or directory'
```

```
## Error in gzfile(file, "rb"): cannot open the connection
```

```
expect_true(exists("g"), info = "Verify graph was created.")
```

```
## Error: exists("g") isn't true.
## Verify graph was created.
```

```
expect_equal(fa17_enrolled, target_df, info = "Verify data frame was created correctly")
```

```
## Error in eval_bare(get_expr(quo), get_env(quo)): object 'fa17_enrolled' not found
```

[**5 Points**] **(b)** One of your international collaborators wanted to use the UIUC international student data at their school. However, the excel spreadsheet that was sent over contains multiple sheets. Your collaborator is only interested in the Spring 2017 enrollment information.

- [**2 Points**] Help your colleague out by reading into *R* only the `Spring2017` sheet from `enroll_global.xlsx`. Call the imported data `sp17_enrolled`.
- [**1 Point**] Provide your colleague with a summary of the imported data. Save the summary into `overview_sp17`.
- [**2 Points**]Which variable (outside of "All") had the *largest* overall amount of students? How about the *smallest* **mean** number of students? (Outside of 0).

```
# your code here
```

<span class="testrmd-badge label label-danger" data-toggle="collapse" data-target="#testrmd-chunk-045

```
# Load secured data
target_df = readRDS("test/sp17-enrolled.rds")
```

```
## Warning in gzfile(file, "rb"): cannot open compressed file 'test/sp17-
## enrolled.rds', probable reason 'No such file or directory'
```

```
## Error in gzfile(file, "rb"): cannot open the connection
```

```
target_summary = readRDS("test/summary-sp17.rds")
```

```
## Warning in gzfile(file, "rb"): cannot open compressed file 'test/summary-
## sp17.rds', probable reason 'No such file or directory'
```

```
## Error in gzfile(file, "rb"): cannot open the connection
```

```
# Verify existence of objects
expect_true(exists("overview_sp17"), info = "Verify summary was created.")
```

```
## Error: exists("overview_sp17") isn't true.
```

```
## Verify summary was created.
```

```
expect_true(exists("sp17_enrolled"), info = "Verify graph was created.")
```

```
## Error: exists("sp17_enrolled") isn't true.
## Verify graph was created.
```

```
# Verify data is accurate
expect_equal(overview_sp17, target_summary, info = "Verify summary was obtained correctly.")
```

```
## Error in eval_bare(get_expr(quo), get_env(quo)): object 'overview_sp17' not found
```

```
expect_equal(sp17_enrolled, target_df, info = "Verify data frame was created correctly.")
```

```
## Error in eval_bare(get_expr(quo), get_env(quo)): object 'sp17_enrolled' not found
```

(17 Points) Exercise 2: The Recycler

To receive credit for answers to exercises listed here, you *must* use *R*'s recycling behavior.

Hint You may wish to consult help documentation for the `rep()` function and revisit the vectorization slides at the Transforming Data and accompanying code.

- **[2 Points] (a)** If we define `x = c(2, 10)` and `y = c(9, 2, 6, 10)`, what should the output of `z = x * y` be?

```
<span class="testrmd-badge label label-danger" data-toggle="collapse" data-target="#testrmd-chunk-230
```

```
target_vec = readRDS("test/z-result.rds")
```

```
## Warning in gzfile(file, "rb"): cannot open compressed file 'test/z-
## result.rds', probable reason 'No such file or directory'
```

```
## Error in gzfile(file, "rb"): cannot open the connection
```

```
expect_equal(z, target_vec)
```

```
## Error in eval_bare(get_expr(quo), get_env(quo)): object 'z' not found
```

- **[3 Points] (b)** Explain why the output is occuring by:
  - Creating an example using `rep()` that mimics how *R* is performing the operation.
  - Describe the underlying property *R* applies to vectors.
- **[6 Points] (c)** What happens if we update `x` to be `x = c(3, 5, 12, 18, 11)` and perform the operation again?
  - **[1 Points]** Recompute the result of `x*y` with the new `x`
  - **[4 Points]** Create and include a *diagram* that shows the process *R* goes through when evaluating this operation.
    - The included diagram can be a picture of a *hand* sketch, digitally drawn(MS Paint, PowerPoint, Photoshop, ...), or even done using `DiagrammeR`.
    - The choice is yours.
- **[3 Points] (d)** Create the `alt_vec` vector with length 31 such that successive values alternating between polarity (e.g. `1` and `-1`).
  - For example, the first *four* values of such a vector would be: `c(1, -1, 1, -1)`
  - *Hint:* Consider $(1)^2$, $(-1)^2$, $(1)^3$, and $(-1)^3$.
  - *Hint:* The colon (`:`) operator is useful to generate sequences.

```
<span class="testrmd-badge label label-danger" data-toggle="collapse" data-target="#testrmd-chunk-096
```

```
target_vec = readRDS("test/alternating-vec.rds")
```

```
## Warning in gzfile(file, "rb"): cannot open compressed file 'test/
## alternating-vec.rds', probable reason 'No such file or directory'
```

```
## Error in gzfile(file, "rb"): cannot open the connection
```

```
expect_equal(alt_vec, target_vec)
```

```
## Error in eval_bare(get_expr(quo), get_env(quo)): object 'alt_vec' not found
```

- **[3 Points] (e)** Construct the following sequence `k` of length 15: $1, -2, 3, 4, -5, 6, 7, -8, 9, \ldots, 13, -14, 15$.
  - This sequence *must* be created using recycling...

```
  <span class="testrmd-badge label label-danger" data-toggle="collapse" data-target="#testrmd-chunk-971
```

```
target_vec = readRDS("test/reconstructed-seq.rds")
```

```
## Warning in gzfile(file, "rb"): cannot open compressed file 'test/
## reconstructed-seq.rds', probable reason 'No such file or directory'
```

```
## Error in gzfile(file, "rb"): cannot open the connection
```

```
# expect_equal(k, target_vec)
```

(8 Points) Exercise 3: Scientists Rock!

[3 Points] (a) Develop a function called `fahrenheit_to_celcius()` that converts fahrenheit (°F) to celcius (°C) given the definition of:

$$f(x) = \frac{5\,(x - 32)}{9}$$

```
#' @title Convert from Fahrenheit to Celcius
#' @description
#' Given a temperature in Fahrenheit convert it to the
#' the temperature in Celcius.
#'
#' @param degrees_c A `numeric` vector containing temperature in degrees Celcius.
#'
#' @return A `numeric` vector of temperatures in degrees Fahrenheit.
#' @details
#' The function implements the following formula:
#' \eqn{f(x) = \frac{{5\left( {x - 32} \right)}}{9}}{5*(x-32)/9}
#'
#' @examples
#' # Calculate a single value
#' fahrenheit_to_celcius(32.0)
#'
#' # Calculate multiple values
#' fahrenheit_to_celcius(c(69.8, 98.6, 212.0))
fahrenheit_to_celcius = function(degrees_c) {

  # Logic implemented here

}
```

```
  <span class="testrmd-badge label label-danger" data-toggle="collapse" data-target="#testrmd-chunk-841
```

```
# Check single value
expect_equal(
```

```
  fahrenheit_to_celcius(32),
  0,
  info = "Check single fahrenheit value"
)
```

```
## Error: fahrenheit_to_celcius(32) not equal to 0.
## target is NULL, current is numeric
## Check single fahrenheit value
```

```
# Check vectorization
expect_equal(
  fahrenheit_to_celcius(c(69.8, 98.6, 212.0)),
  c(21, 37, 100),
  info = "Check multiple fahrenheit values"
)
```

```
## Error: fahrenheit_to_celcius(c(69.8, 98.6, 212)) not equal to c(21, 37, 100).
## target is NULL, current is numeric
## Check multiple fahrenheit values
```

[**5 Points**] **(b)** Now, create a function called `celcius_to_fahrenheit()` that does the inverse of the previous one. That is, write $f^{-1}(x)$ which translates celcius (°C) to fahrenheit (°F).

- [**2 Points**] Derive the formula for the celcius to fahrenheit conversion and include it in the `@details` section of the function's documentation.
- [**3 Points**] Implement the derived formula in the *R* function below.

```
#' @title Convert from Celcius to Fahrenheit
#' @description
#' Given a temperature in Celcius convert it to the
#' the temperature in Fahrenheit.
#'
#' @param degrees_f A `numeric` vector containing temperature in degrees Fahrenheit.
#'
#' @return A `numeric` vector of temperatures in degrees Celcius.
#' @details
#' The function implements the following formula:
#' \eqn{LATEX-HERE}{REGULAR-SYMBOLS-HERE}
#' An example of this is given above...
#' @examples
#' # Calculate with a single value
#' celcius_to_fahrenheit(0)
#'
#' # Calculate multiple values
#' celcius_to_fahrenheit(c(21, 37, 100))
celcius_to_fahrenheit = function(degrees_f) {

  # Logic implemented here

}
```

```
  <span class="testrmd-badge label label-danger" data-toggle="collapse" data-target="#testrmd-chunk-766
```

```
# Check single value
expect_equal(
  celcius_to_fahrenheit(c(0)),
  32,
```

```
  info = "Check single celcius value"
)
```

```
## Error: celcius_to_fahrenheit(c(0)) not equal to 32.
## target is NULL, current is numeric
## Check single celcius value
```

```
# Check vectorization
expect_equal(
  celcius_to_fahrenheit(c(21, 37, 100)),
  c(69.8, 98.6, 212.0),
  info = "Check multiple celcius values"
)
```

```
## Error: celcius_to_fahrenheit(c(21, 37, 100)) not equal to c(69.8, 98.6, 212).
## target is NULL, current is numeric
## Check multiple celcius values
```

(10 Points) Exercise 4: Winter is Coming

[**2 Points**] **(a)** Install and load the `nasaweather` package. **Comment** out the installation command in your `.Rmd` file. (If you do not comment installation commands out, then they will be run every time you knit your `.Rmd` file.)

[**2 Points**] **(b)** Open up the help documentation for `glaciers` (the data set), find where the variables for the data set are listed and write in your *RMarkdown* document *what* the `country` variable contains.

[**6 Points**] **(c)** Provide summary information on the `glaciers` by:

- writing a sentence that *dynamically* describes the dimensions of the data;
  - A *dynamic* description means that sentences should use inline *R* code. Examples can be found in Literate Programming
- showing the last ten observations in the data set; and,
- providing a summary overview of the data to understand its contents.

(14 Points) Exercise 5: Excellency at UIUC

Under this exercise, we will explore the "Teachers Ranked As Excellent" data at UIUC from Fall 1993 to Spring 2018 as compiled by Wade Fagen-Ulmschneider. Please download the data from:

https://raw.githubusercontent.com/wadefagen/datasets/master/teachers-ranked-as-excellent/uiuc-tre-dataset.csv

To download the data, go to the webpage and press `Cntrl/Cmd + S` to save the file to your local computer. Name the file `uiuc-tre-dataset.csv`. *Upload it* onto RStudio Cloud. For help, see **Page 20** of **Reading 0**.

This data has a file extension of **CSV** form. Contained in the data are the following variables:

- `term`: Two letter semester code (`sp`, `su`, `fa`, or `wi`) followed by a four digit year.
  - Examples: `sp2017`, `fa2013`, `su2012`.
- `unit`: The CITL-supplied headers for the unit teaching the course.
  - Examples: `ACCOUNTANCY`, `SOCIAL WORK`, `LINGUISTICS`, `NUCLEAR, PLASMA & RAD. ENGR.`
- `lname`: The last name of the teacher.
  - Examples: `FAGEN-ULMSCHNEIDER`, `FLANAGAN`, `FLECK`
- `fname`: The first letter of the first name of the teacher.
  - Examples: `W`, `K`, `M`
- `role`: `Instructor` or `TA`

- `ranking`: `Excellent` or `Outstanding`
- `course`: The course the teacher was ranked as excellent. If no course is given, the `course` field is set to `?` (this includes cases when the raw data lists the course as `0`, `000`, or `999`).
  - Examples: `199`, `225`, `560`, `?`

[2 points] (a) Import into *R* the data in `uiuc-tre-dataset.csv` with the variable name `teaching_list_data`. As `course` contains a value that is *not* `NA`, which is how *R* represents missing values, you must use the parameter `na.strings = c("?", "NA")` in the appropriate `read.*()` function. If you need help, please see the appropriate help documentation via `?`.

NB The `*` in the `read.*()` statement is a placeholder for the type of file you want to read in.

[2 points] (b) Retrieve the dimensional information for this data using only one function.

[4 points] (c) Perform a summary of the `teaching_list_data`. Within the summary output, what variable output is different from the rest? What might have caused this? *Hint* Consider looking at the structure of `teaching_list_data`.

[2 points] (d) Who was your favorite teacher at UIUC? Search their last name in all capitals.

```
teaching_list_data[teaching_list_data$lname == "YOUR TEACHERS LAST NAME HERE IN CAPITALS", ]
```

[4 points] (e) Fix the following code so that it:

1. doesn't error;
2. produces a graph; and
3. hides the code.

Hint for the last two requirements look at different code chunk options.

```
ggplot(teaching_list_data, aes(fname)) +
  geom_bar() +
  facet_wrap( ~role) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(y = "Frequency",
       x = "First Name Letter")
```