Purr (iteration) homework

(1 pt for each problem except p. 328-329 (21.5.3) #1 which is 2 pts). Be sure and show test results in your document.

p. 316 (21.2.1) #1,2:

1. Write for loops to:
   1. Compute the mean of every column in mtcars.
   2. Determine the type of each column in nycflights13::flights.
   3. Compute the number of unique values in each column of iris.
   4. Generate 10 random normals for each of μ = −10, 0, 10, and 100.

Think about the output, sequence, and body **before** you start writing the loop.

1. Eliminate the for loop in each of the following examples by taking advantage of an existing function that works with vectors:

out <- ""

for (x in letters) {

out <- stringr::**str\_c**(out, x)

}

x <- **sample**(100)

sd <- 0

for (i in **seq\_along**(x)) {

sd <- sd + (x[i] - **mean**(x)) ^ 2

}

sd <- **sqrt**(sd / (**length**(x) - 1))

x <- **runif**(100)

out <- **vector**("numeric", **length**(x))

out[1] <- x[1]

for (i in 2:**length**(x)) {

out[i] <- out[i - 1] + x[i]

}

p. 321 (21.3.5) #1, 3

1. Imagine you have a directory full of CSV files that you want to read in. You have their paths in a vector, files <- dir("data/", pattern = "\\.csv$", full.names = TRUE), and now want to read each one with read\_csv(). Write the for loop that will load them into a single data frame. You may assume that all csv files contain the same variables and formats. Test your code by downloading the stock price .csv files from the data folder on blackboard and then print out rows 50,000 to 50,015.

3. Write a function that prints the mean of each numeric column in a data frame, along with its name. For example, show\_mean(iris) would print:

**show\_mean**(iris)

*#> Sepal.Length: 5.84*

*#> Sepal.Width: 3.06*

*#> Petal.Length: 3.76*

*#> Petal.Width: 1.20*

p. (21.4.1) #2

2. Adapt col\_summary() so that it only applies to numeric columns . You may want to look at the purrr function keep. Test your function by computing the median on the flights dataframe.

p. 328-329 (21.5.3) #1,2

1. Write code that uses one of the map functions to:
   1. Compute the mean of every column in mtcars.
   2. Determine the type of each column in nycflights13::flights.
   3. Compute the number of unique values in each column of iris.
   4. Generate 10 random normals for each of μ = −10, 0, 10, and 100.
2. How can you create a single vector that for each column in a data frame indicates whether or not it’s a factor? Test on the diamonds dataset.

21.7 exercises:

1. Write code to generate 6 random numbers from a normal distribution with mean of 3 and standard deviation of 0.2, 10 random numbers from a uniform distribution from 2 to 5, and 12 random numbers from a poisson distribution with a lambda of 3.5. Set your seed to 613 at the beginning.

2. Combine the following datasets from Lahman: Master, Salaries, AwardsPlayers, Batting and BattingPost.

Extra Credit (5 pts)

Download the stock price .xlsx files from Blackboard to your data subfolder. Read in the "dividends" sheet from all .xlxs files in the data folder, add the company symbol from the filename as a new variable called Symbol and combine the data from the excel spreadsheets into one large tibble. Then tidy the data so the head looks like:

Dividends Symbol Event Date

<dbl> <chr> <chr> <dttm>

1 0.45 AFL Pay\_date 2017-12-01

2 0.45 AFL Record 2017-11-15

3 0.45 AFL Ex\_Div 2017-11-14

4 0.45 AFL Declared 2017-10-25

5 0.43 AFL Pay\_date 2017-09-01

6 0.43 AFL Record 2017-08-23

7 0.43 AFL Ex\_Div 2017-08-21

8 0.43 AFL Declared 2017-07-27

9 0.43 AFL Pay\_date 2017-06-01

10 0.43 AFL Record 2017-05-24