STA 445 Exam #1

Angelica Alcala

November 09, 2023

Exam Questions

Question 1 [30 points]

You will be asked to produce two functions. The first (`vecsummary`) will create a summary output for a given numerical vector. The second (`mean_split`) will provide ABOVE/BELOW strings depending on if the element in the vector is above or below the mean of the vector. Please answer each sub question below.

a. For some data to analyze with your functions, load the data set `cars` from base R. The `cars` set contains two variables, `speed` and `dist`, of which there are 50 total observations.

```
data(cars)
mean(cars$speed)
```

```
## [1] 15.4
```

```
mean(cars$dist)
```

```
## [1] 42.98
```

b. Produce the function `vecsummary`. The function should accept as an input a numerical vector and return a data.frame with columns `Size`, `Min`, `Max`, `Min.Max`, `Mean`, in that order. The size here refers to the number of elements (`length`) in the vector. The element `Min.Max` is the ratio `Min / Max`. The function should return a data.frame with proper labels for each column, in the order specified above..

```
vecsummary <- function(x){
  Size <- length(x)
  Min.Max <- Min/Max
  Min <- min(data.frame$x)
  Max <- max(data.frame$x)
  Mean <- mean(x)
output <- data.frame('Size','Min','Max','Min.Max','Mean')
return(output)
}
```

c. Demonstrate that the `vecsummary` function works properly by displaying the results of your function when analyzing the variable `speed` within the `cars` object.

```
#vecsummary(cars$speed)
```

d. Produce the function `mean_split`. The function should accept as an input a numerical vector and return a vector of ABOVE/BELOW determined for each element being above or below the mean value of the vector.

```
mean_split <- function(x){
  output <- ifelse(x < mean(x), 'BELOW','ABOVE')
  return(output)
}
```

e. Demonstrate that the `mean_split` function works properly by displaying the results of your function when analyzing the variable `dist` within the `cars` object.

```
mean_split(cars$dist)
```

```
##  [1] "BELOW" "BELOW" "BELOW" "BELOW" "BELOW" "BELOW" "BELOW" "BELOW" "BELOW"
## [10] "BELOW" "BELOW" "BELOW" "BELOW" "BELOW" "BELOW" "BELOW" "BELOW" "BELOW"
## [19] "ABOVE" "BELOW" "BELOW" "ABOVE" "ABOVE" "BELOW" "BELOW" "ABOVE" "BELOW"
## [28] "BELOW" "BELOW" "BELOW" "ABOVE" "BELOW" "ABOVE" "ABOVE" "ABOVE" "BELOW"
## [37] "ABOVE" "ABOVE" "BELOW" "ABOVE" "ABOVE" "ABOVE" "ABOVE" "ABOVE" "ABOVE"
## [46] "ABOVE" "ABOVE" "ABOVE" "ABOVE" "ABOVE"
```

Question 2 [20 points]

In the area marked as `???` write a regular expression that accomplishes the requested task. Use the `strings` object to demonstrate your answer is correct by giving a minimum of two TRUE and two FALSE responses.

a. An expression that can determine if a string contains any of the following words: `Captain`, `Chief`, `Colonel`, `Cadet`, `Cook`.

```
#strings <- c("Captain", "Chief", "Colonel", "Cadet", "Cook", "hello","test", "Frank")
#data.frame(string = strings) %>%
  #mutate(result = str_detect(string, 'Captain', 'Chief', `Colonel`, `Cadet`, `Cook`) )
```

b. An expression that can determine if a string begins with 2 of any upper or lowercase letter followed by any digit 4 times.

```
### Change to eval=TRUE or remove eval option
#strings <- c('a5555', 'hello')
#data.frame( string = strings ) %>%
  #mutate( result = str_detect(string, 'D', 'd', '{4}') )
```

c. An expression that can determine if a string begins with `bar` or ends in `foo`.

```
### Change to eval=TRUE or remove eval option
strings <- c()
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '???') )
```

Question 3 [15 points]

a. Traveling to `Europe/Rome`, you are told that your flight will arrive at 2:00 PM local time on June 15th, 2024. Create the object `Rome.Arrival` for this time stamp, be sure to include the timezone. Display the object to screen.

```
Rome.Arrival <- mdy_hm('Jun 15 2024, 2:00 PM', tz = 'Europe/Rome')
Rome.Arrival
```

```
## [1] "2024-06-15 14:00:00 CEST"
```

b. You want to make a call back to the USA when you arrive. What time will it be in `US/Arizona` when you arrive in Rome? Return/display the results of an R calculation here.

```
UStime <- Rome.Arrival %>%
  with_tz(tzone = 'US/Arizona')
UStime
```

```
## [1] "2024-06-15 05:00:00 MST"
```

c. You are set to return to the USA on August 10th, 2024 at 10:00 PM (Arizona). Assuming all of these dates/times are not delayed, how many **days** will you be traveling? (Return a calculated time object that displays days, hours, minutes, seconds).

```
USreturn <- mdy_hm('Aug 10 2024, 10:00 PM', tz = 'US/Arizona')
USreturn
```

```
## [1] "2024-08-10 22:00:00 MST"
```

```
Travel <- interval(Rome.Arrival, USreturn)
Traveltime <- as.period(Travel)
Traveltime
```

```
## [1] "1m 26d 17H 0M 0S"
```

Question 4 [25 points]

The `Exam1_Data_F23.xlsx` file contains data developed across multiple trips (`ID`) where information was collected for four different `Sites` at four different `Periods`. Two measurements were taken for each observations and were recorded as `X` and `Y`. There are potentially many observations of the same `Period` and `Site`. The data was collected across two trips that are available in the `Exam1_Data.xlsx` file on Sheets 2 and 3.

a. Read in the two different data sets on Sheets 2 and 3. Combine the data into a single data frame named `Q4data`. There should be 5 columns: `ID`, `Site`, `Period`, `X`, `Y`, totaling 100 observations. Demonstrate this by returning to screen the structure of the `Q4data` object. This can be done using the `str()` function. Do **not** display any of the data.frames in your final answer, only the structure of the combined data frame (a tibble is acceptable).

```
Sheet2 <- read_excel(path = 'Exam1_Data_F23.xlsx', sheet = 2, range = 'A1:E47')
Sheet3 <- read_excel(path = 'Exam1_Data_F23.xlsx', sheet = 3, range = 'A1:E55')
Q4data <- full_join(Sheet2,Sheet3)
```

```
## Joining with `by = join_by(ID, Site, Period, X, Y)`
```

```
str(Q4data)
```

```
## tibble [100 x 5] (S3: tbl_df/tbl/data.frame)
##  $ ID    : num [1:100] 1 1 1 1 1 1 1 1 1 1 ...
##  $ Site  : chr [1:100] "B" "C" "C" "B" ...
##  $ Period: chr [1:100] "S" "Q" "S" "Q" ...
##  $ X     : num [1:100] -1.45 1.36 0.966 -1.216 0.925 ...
##  $ Y     : num [1:100] -3.35 5.08 3.9 -2.65 3.78 ...
```

b. We are interested in counting the number of `Site` and `Period` combinations. Create a `wide` table that has the `Period` as columns, the `Site` as rows, and the counts for each pair within the table.

```
Q4data2 <- Q4data %>%
  pivot_wider(Site, names_from = ID, values_from = Period)
```

```
## Warning: Specifying the `id_cols` argument by position was deprecated in tidyr 1.3.0.
## i Please explicitly name `id_cols`, like `id_cols = Site`.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: Values from `Period` are not uniquely identified; output will contain
## list-cols.
## * Use `values_fn = list` to suppress this warning.
## * Use `values_fn = {summary_fun}` to summarise duplicates.
## * Use the following dplyr code to identify duplicates.
##   {data} %>%
```

```
##   dplyr::group_by(Site, ID) %>%
##   dplyr::summarise(n = dplyr::n(), .groups = "drop") %>%
##   dplyr::filter(n > 1L)
```

```
Q4data2
```

```
## # A tibble: 3 x 5
##   Site  `1`        `2`        `3`        `4`
##   <chr> <list>     <list>     <list>     <list>
## 1 B     <chr [10]> <chr [3]>  <chr [7]>  <chr [6]>
## 2 C     <chr [6]>  <chr [14]> <chr [9]>  <chr [10]>
## 3 A     <chr [4]>  <chr [9]>  <chr [13]> <chr [9]>
```

c. Finally, we want to make a visual assessment of the total `X` and total `Y` observed across the four different observation sets `ID`. Group the data by `Site` and `Period` and summarize the total `X` and total `Y` found for each combination. Save this as a new data.frame `plotme` and display the `head`. *Hint: Finding the total requires we take the sum.*

d. Use the data.frame `plotme` to view the total X against the total Y as a scatter graph, and color the scatter based on the `Site`.