Exercise 4: Base R vs. Tidyverse

Matthew Kim

11/4/2021

Base R tasks

Thank you Dillon for assistance with question 06 for part 1 in base R, and to Veronica for assistance with several sections in part 1 and in Tidyverse for part 2.

1. Download the food_coded.csv file

```
food <- read.csv("food_coded.csv")
```

2. Load the CSV file into your R environment.

```
food <- read.csv("food_coded.csv")
```

3. Extract the first 95 rows.

```
food1  <- food[1:95,]
```

4. Look at the following variables using both name and column index/number.

    - GPA
    - calories_chicken
    - drink
    - fav_cuisine
    - father_profession
    - mother_profession

```
food2 <- food1[  , c("GPA", "calories_chicken", "drink", "fav_cuisine", "father_profession", "mother_pr
```

5. Create a new variable for how healthy each person feels but convert the scale from 1 to 10 to 1 to 100.

```
library(scales)
food$healthyfeeling2 <- rescale(food$health_feeling, to = c(1, 100))
```

6. Filter to students who are female and have GPAs that are above 3.0.

```r
food$GPAnew <- as.numeric(as.character(food$GPA))

food[74, 63] <- 3.79

GPAfilter <- subset(food, Gender == "1" & GPAnew > "3")
```

7. Find the mean and standard deviation for the following variables, and summarize them in a data frame.

    - chicken_calories
    - tortilla_calories
    - turkey_calories
    - waffle_calories

```r
calories <- food[  , c("calories_chicken", "tortilla_calories", "turkey_calories", "waffle_calories")]

calories$chickenM <- mean(calories$calories_chicken)
calories$chickenSD <- sd(calories$calories_chicken)

calories$tortillaM <- mean(calories$tortilla_calories)
calories$tortillaSD <- sd(calories$tortilla_calories)

calories$turkeyM <- mean(calories$turkey_calories)
calories$turkeySD <- sd(calories$turkey_calories)

calories$waffleM <- mean(calories$waffle_calories)
calories$waffleSD <- sd(calories$waffle_calories)

head(calories)
```

8. Summarize GPA and weight within the gender and cuisine variables.

```r
food$weight <- as.numeric((food$weight))

food$weight[4] <- 240

food$weight[68] <- 144

female <- subset(food, Gender == 2)
male <- subset(food, Gender == 1)

## Mean of GPA (femaleG) and Weight (W) in cuisine variable ##

femaleG <- tapply(female$GPAnew, female$cuisine, mean, na.rm = T)

maleG <- tapply(male$GPAnew, male$cuisine, mean, na.rm = T)

femaleW <- tapply(female$GPAnew, female$weight, mean, na.rm = T)

maleW <- tapply(male$GPAnew, male$weight, mean, na.rm = T)

## Standard Deviation of GPA (femaleG) and Weight (W) in cuisine variable ##
```

```
femaleGSD <- tapply(female$GPAnew, female$cuisine, sd, na.rm = T)

maleSD <- tapply(male$GPAnew, male$cuisine, sd, na.rm = T)

femaleW <- tapply(female$GPAnew, female$weight, sd, na.rm = T)

maleW <- tapply(male$GPAnew, male$weight, sd, na.rm = T)
```

Tidyverse tasks

```
library(tidyverse)
```

1. Download the facebook-fact-check.csv

```
facebook <- read.csv("facebook-fact-check.csv")
```

2. Load the CSV file into your R environment.
3. Extract the last 500 rows.

   Hint: Check out the top_n() page to figure out how to extract the last 500 rows instead of the first 500 rows.
4. Look at the even-numbered column indices only. Identify them by name.

```
facebook %>% select(2,4,6,8,10,12)
```

The even numbered columns are called “post_id”, “Page”, “Date.Published”, “Rating”, “share_count” and “comment_count”

5. Using `mutate`, create a new variable called `post_type_coded` that renames each post type to the following:

   - link = 1
   - photo = 2
   - text = 3
   - video = 4

   Hint: look up case_when within tidyverse. You can also use if_else

```
facebook2 <-
    facebook %>% mutate(post_type_coded = case_when(
        facebook$Post.Type == "link" ~ 1,
        facebook$Post.Type == "photo" ~ 2,
        facebook$Post.Type == "text" ~ 3,
        facebook$Post.Type == "video" ~ 4,
           ))
head(facebook2)
```

6. Arrange page names in reverse order.

```
facebook %>% arrange(desc(Page))
```

7. Find the mean and standard deviation for the following variables, and summarize them.

    - share_count
    - reaction_count
    - comment_count

```
facebook %>%
    summarise(across(c(share_count, reaction_count, comment_count), list(sd = sd, mean = mean), na.rm =
```

8. Summarize the mean and standard deviations in Question 7 with the "mainstream" values in the `category` variable.

```
facebook %>%
    filter(Category == "mainstream") %>% summarise(across(c(share_count, reaction_count, comment_count)
```

Submit

Email me (laaker@wisc.edu) the link to your `ps811-exercises` repository when you are done.