Assignment 2 (Part B)

2023-08-29

Assignment 2 Part B responses

Below are my responses for Assignment 2, part B.

Question 1

- Question: **Edit the command below so that you're specifying either male *or* first-year. Are more or fewer individuals selected compared to the command below? Give a brief explanation of why that's the case.**

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become er
```

```
ROOT = "/data/Week_2/assignment_b/Clean-KimData.csv"
Clean.KimData <- read.csv(ROOT)
KimData <- Clean.KimData
FreshAndMales <-KimData %>%
  filter(Semester < 2 & Gender == "M")
count(FreshAndMales)
```

```
##    n
## 1 43
```

- Response: The below code will change the outout to **male or first-year**.

```
library(tidyverse)
ROOT = "/data/Week_2/assignment_b/Clean-KimData.csv"
Clean.KimData <- read.csv(ROOT)
KimData <- Clean.KimData
FreshOrMales <-KimData %>%
  filter(Semester < 2 | Gender == "M")

count(FreshOrMales)
```

```
##     n
## 1 229
```

As we can see, the count for `FreshOrMales` is greater than or equal to `FreshAndMales`.

We know why the number of elements in variable `FreshOrMales` will always be greater than or equal to `FreshAndMales` from set theory. Denote the *cardinality* (number of elements) of a set S to be $n(S)$. We know 3 things about any finite sets S, T

1. $(S \cap T) \subseteq S$ and $(S \cap T) \subseteq T$.

2. $S \subseteq T \implies n(S) \leq n(T)$
3. $n(S \cup T) = n(S) + n(T) - n(S \cap T)$

Let $A = \{x | x \in \text{Male Students}\}$ and $B = \{x | x \in \text{Freshman Students}\}$. From (1) and (2) above, we know that $n(A \cap B) \leq n(A)$ and $n(A \cap B) \leq n(B)$ and thus

$$\begin{aligned} 2 * n(A \cap B) \leq n(A) + n(B) &\implies \\ n(A \cap B) \leq n(A) + n(B) - n(A \cap B) &\implies \\ n(A \cap B) \leq n(A \cup B) & \end{aligned}$$

Hence, we know that `FreshOrMales` will always be greater than or equal to `FreshAndMales`.

Question 2

- Question: **A person's "Body Mass Index" is calculated by taking mass divided by height squared. If you're measuring in metric (kg and m), you're done. If you're measuring in pounds and inches (as we're doing here), you then have to multiply by 703. In other words, BMI = 703*(Weight/Height^2). Use the `mutate` command twice to first create the `BMI` variable, and then create a variable `Obese` whose value is TRUE for anyone whose BMI is greater than or equal to 30. Print the first six observations using the function head().**
- Response: ***See the code below***

```
library(tidyverse)
KimDataBMI <- KimData %>%
  mutate(BMI = 703*(Weight/Height^2)) %>%
  mutate(Obese = BMI >= 30)
head(KimDataBMI[c("BMI","Obese")])
```

```
##        BMI Obese
## 1 27.19401 FALSE
## 2 32.09497  TRUE
## 3 22.14871 FALSE
## 4 33.12476  TRUE
## 5 23.87746 FALSE
## 6 25.38949 FALSE
```

Question 3

- Question: **We've heard of the "freshman 15," the weight that many college students gain after their first year of all-you-can-eat dorm food. Use `group_by` and `summarize` to group students by `Year` and calculate mean BMI for each group. Does BMI seem to be higher for students who have been here more years?**
- Response: ***See the code below***

```
sumBMI <- KimDataBMI %>%
  mutate(Year=round(Semester / 2)) %>%
  group_by(Year) %>%
  summarize(Average.BMI = mean(BMI, na.rm=TRUE))
sumBMI
```

```
## # A tibble: 6 x 2
##    Year Average.BMI
##   <dbl>       <dbl>
## 1     0        25.1
## 2     1        23.1
## 3     2        24.0
## 4     3        23.5
## 5     4        24.6
## 6     5        28.5
```

As we can see, the average BMI for freshmen is slightly higher than upper class students except for students who have been here beyond the standard 4 years.

Question 4

- Question: **Write the code to calculate the percentage of obese people in each year. Does this percentage show any clear trend?**
- Response: ***See the code below***

Below is another method where we make a new column, `Class.Name`, which will have the columns **Freshman, Sophomore, Junior, and Senior**. Note that we will denote **Senior** as anyone who has been here for 6 or more semesters. In fact, we will order them that way via the `factor` command. We will call the new column `Class.Name`.

```
KimDataYear <- KimDataBMI %>%
 mutate(Class.Name = case_when(Semester < 2 ~ "Freshman",
            Semester < 4 ~ "Sophomore",
            Semester < 6 ~ "Junior",
            Semester >= 6 ~ "Senior"))

### Now we set an imposed order on the column (Freshman < Sophomore < Junior < Senior)
KimDataYear$Class.Name <- factor(KimDataYear$Class.Name, c("Freshman","Sophomore","Junior","Senior")
#### Now we group by
```

```
KimDataYear %>% group_by(Class.Name) %>%
  summarize(n = n(), Obese.Percent = 100*mean(Obese==TRUE, na.rm=TRUE))
```

```
## # A tibble: 4 x 3
##   Class.Name     n Obese.Percent
##   <fct>      <int>         <dbl>
## 1 Freshman     110          18.6
## 2 Sophomore    164          10.6
## 3 Junior        59           3.57
## 4 Senior        44          11.6
```

It would appear that the percentage of **obese** students is higher for **Freshman** than other years.

Question 5

- Question: **Write two R commands using `dplyr`. One should calculate the average size of men and women's feet using all data points. The second should calculate the average size of men's and women's feet after removing the person with the giant feet. (Think about how you'll remove that person using a dplyr command.) How much of a difference in average shoe size did removing the outlier make**
- Response: ***See the code below***

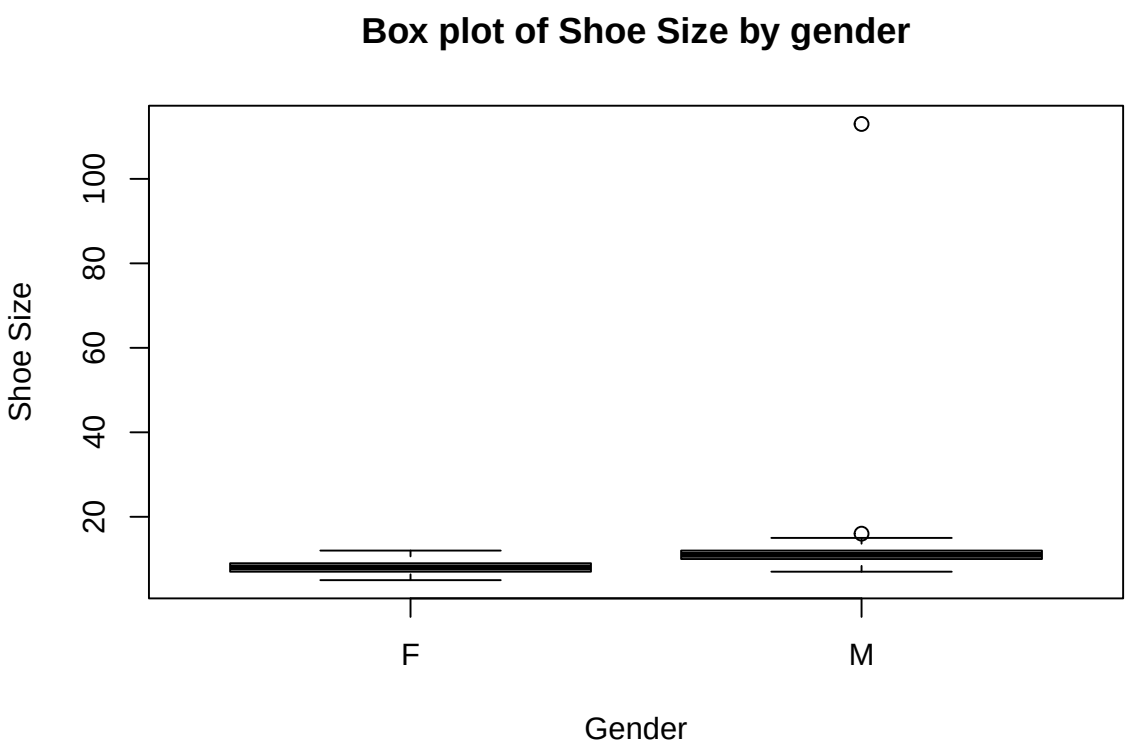
First we will compute the average foot size between male and female students, removing NAs or "other".

```
average.foot <- Clean.KimData %>%
     filter(!(is.na(Gender)) & Gender!="other") %>%
  group_by(Gender) %>%
  summarize(Average.Shoe = mean(Shoe.Size, na.rm = TRUE))

average.foot
```

```
## # A tibble: 2 x 2
##   Gender Average.Shoe
##   <chr>         <dbl>
## 1 F              7.99
## 2 M             11.5
```

Let's take a look a the boxblot of foot sizes.

```
gendered_data <- Clean.KimData %>%
      filter(!(is.na(Gender)) & Gender!="other")

boxplot(gendered_data$Shoe.Size ~ gendered_data$Gender,
        xlab = "Gender", ylab = "Shoe Size",main = "Box plot of Shoe Size by gender")
```



As we can see, we have a shoe size that appears to be well beyond what is possible. Let's look at the order of values for shoe size in reverse order:

```
male.shoesize <- gendered_data %>%
  filter(Gender == "M") %>%
  arrange(desc(Shoe.Size))

head(male.shoesize$Shoe.Size)
```

```
## [1] 113  16  15  14  14  14
```

As we can see, the largest shoe size is 113 which is clearly an outlier. As stated in the instructions, we don't necessarily want remove outliers from the dataset unless we can't find a good reason for the data point or cannot correct it. Since a shoe size of 113 is likely impossible and we have no method of correcting the datapoint, we will simply remove the value and recompute the average foot size.

```
average.fix <- Clean.KimData %>%
     filter(!(is.na(Gender)) & Gender!="other") %>%
  filter(Shoe.Size <= 20) %>% # Remove giant feet
  group_by(Gender) %>%
  summarize(Average.Shoe = mean(Shoe.Size, na.rm = TRUE))
average.fix
```

```
## # A tibble: 2 x 2
##   Gender Average.Shoe
##   <chr>         <dbl>
## 1 F              7.99
```

```
## 2 M             10.9
```

Let's look at the new boxplot without the outlier

```
cleaned.vals <- Clean.KimData %>%
     filter(!(is.na(Gender)) & Gender!="other") %>%
  filter(Shoe.Size <= 20) #Remove giant feet
# Draw boxplot
boxplot(cleaned.vals$Shoe.Size ~ cleaned.vals$Gender,
        xlab = "Gender", ylab = "Shoe Size",main = "Box plot of Shoe Size by gender fixed")
```

Note that the average for male shoe size dropped from 11.5 to 10.9, over half a shoe size. Remember that the average value is not very robust against outliers, especially significant outliers.

Let's do what we did last week and see if there is a statisically significant difference between shoe sizes of males and females. We will conduct a **Student T-Test** on the data.

```
m.shoesize <- cleaned.vals %>%
  filter(Gender == "M")
f.shoesize <- cleaned.vals %>%
  filter(Gender=="F")
t.test(x = m.shoesize$Shoe.Size, y = f.shoesize$Shoe.Size, alternative = "g", conf.level = 0.99)
```

```
##
##  Welch Two Sample t-test
##
## data:  m.shoesize$Shoe.Size and f.shoesize$Shoe.Size
## t = 19.256, df = 317.35, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 99 percent confidence interval:
##  2.536178      Inf
## sample estimates:
## mean of x mean of y
## 10.879687  7.992991
```

Our null hypothesis H_0 is that there is no difference between average male and female shoe sizes. The alternative hypothesis H_1 is that male shoe sizes are greater on average than female shoe sizes. Since the resultant p-value is less than 0.01, we reject the null hypothesis and accept H_1 that the average shoe sizes for males are statistically larger than average female shoe sizes.