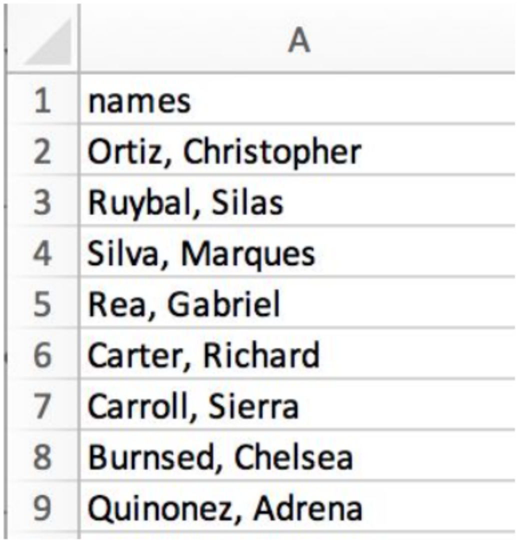
Assignment 1

Jordan Lian

2/9/2021

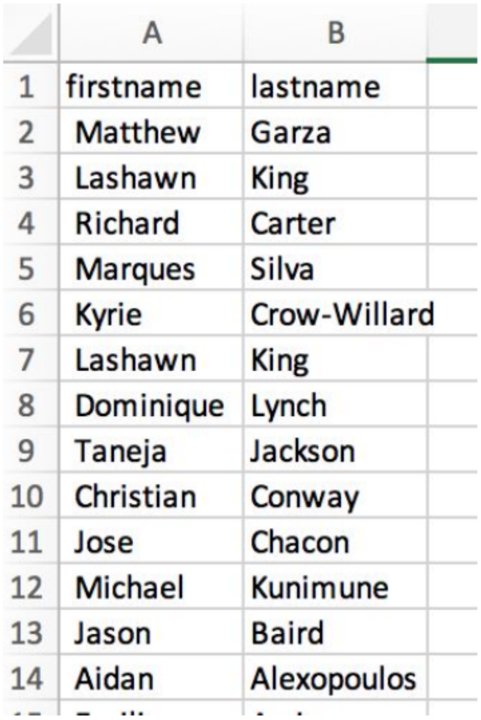
Problem 1 (40 Points)

50 students registered for a mechanical design course. Their names are given in the course roster that can be accessed by the professor. The details of their names are given in the file titled “roster.csv”. The names are printed as “last name, first name” format. A snapshot of the roster data is given below.



	A
1	names
2	Ortiz, Christopher
3	Ruybal, Silas
4	Silva, Marques
5	Rea, Gabriel
6	Carter, Richard
7	Carroll, Sierra
8	Burnsed, Chelsea
9	Quinonez, Adrena

When the students attend a lecture they enter the attendance for each class using an online form. The form records their first name and last name. The attendance data is given in the file titled “attendance.csv”. A snapshot of the student attendance data is given below.



	A	B
1	firstname	lastname
2	Matthew	Garza
3	Lashawn	King
4	Richard	Carter
5	Marques	Silva
6	Kyrie	Crow-Willard
7	Lashawn	King
8	Dominique	Lynch
9	Taneja	Jackson
10	Christian	Conway
11	Jose	Chacon
12	Michael	Kunimune
13	Jason	Baird
14	Aidan	Alexopoulos

Tasks

1. Read the two data files in R studio

```
# Libraries
library(tidyverse)
```

```
## -- Attaching packages ------------------------------------------------------------------------------
```

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## -- Conflicts ---------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
# Original Files
origin_roster <- read_csv("roster.csv")
```

```
## Parsed with column specification:
## cols(
##   names = col_character()
## )
```

```r
origin_attendance <- read_csv("attendance.csv")
```

```
## Parsed with column specification:
## cols(
##   firstname = col_character(),
##   lastname = col_character()
## )
```

```r
# Copy Data Frames to Modify
roster <- origin_roster
attendance <- origin_attendance
```

2. Write a code to aggregate attendance, i.e., how many lectures were attended by each student. If a student has not attended any lectures, set the value as zero. The code should be effective in case of multiple instances of the same first name or last name (or both). These values must be updated in the roster.

```r
# Combine first and last name to match up with roster, create new column
attendance$fullname <- paste(attendance$lastname, attendance$firstname, sep=", ")

# Create new data frame using a table which gets the counts of the names
attendance_count <- as.data.frame(table(attendance$fullname))
colnames(attendance_count) <- c("names", "freq")

# Put values into roster, use match() to match the names from the two datasets
roster$Count <- attendance_count$freq[match(roster$names, attendance_count$names)]

# Where values are NA, mark attendance as 0
roster$Count[is.na(roster$Count)] <- 0

# Separate first and last names, reorder the columns to print first name before last name
roster <- roster %>% separate(names, c("Last Name", "First Name"), sep = ", ")
roster <- roster[, c(2, 1, 3)]
```

3. The output of the code should generate a roster data frame as given below.

First Name	Last Name	Count
Bob	Ross	4
Ron	Swanson	0

```r
roster
```

```
## # A tibble: 50 x 3
##    `First Name` `Last Name` Count
##    <chr>        <chr>       <dbl>
##  1 Christopher  Ortiz           5
##  2 Silas        Ruybal          1
##  3 Marques      Silva           2
##  4 Gabriel      Rea             6
##  5 Richard      Carter          9
##  6 Sierra       Carroll         2
##  7 Chelsea      Burnsed         4
##  8 Adrena       Quinonez        4
##  9 Cyra         Morris          3
## 10 Monica       Caster          3
## # ... with 40 more rows
```

Problem 2

From the "wine_data.csv", answer the following questions using data wrangling functions from relevant packages

```r
wine_data <- read_csv("wine_data.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   X1 = col_double(),
##   country = col_character(),
##   description = col_character(),
##   designation = col_character(),
##   points = col_double(),
##   price = col_double(),
##   province = col_character(),
##   region_1 = col_character(),
##   region_2 = col_character(),
##   variety = col_character(),
##   winery = col_character()
## )
```

1. Write a code to calculate the frequency count of "variety" variable from the dataset. Display top 10 variety by count (**10 points**)

```r
# Create table, change column names
var_freq <- as.data.frame(table(wine_data$variety))
colnames(var_freq) <- c("variety", "freq")

# Get top 10 values, and arrange in descending order
var_freq %>%
  top_n(10, freq) %>%
  arrange(desc(freq))
```

```
##                     variety  freq
## 1                Chardonnay 14482
## 2                Pinot Noir 14291
## 3        Cabernet Sauvignon 12800
## 4                 Red Blend 10062
## 5  Bordeaux-style Red Blend  7347
## 6           Sauvignon Blanc  6320
## 7                     Syrah  5825
## 8                  Riesling  5524
## 9                    Merlot  5070
## 10                Zinfandel  3799
```

2. Write a code to calculate the average points by country (**10 points**)

```
wine_data %>%
  group_by(country) %>%
  summarise(avg_points = mean(points))
```

```
## ‘summarise()‘ ungrouping output (override with ‘.groups‘ argument)
```

```
## # A tibble: 49 x 2
##    country                avg_points
##    <chr>                       <dbl>
##  1 Albania                      88  
##  2 Argentina                    86.0
##  3 Australia                    87.9
##  4 Austria                      89.3
##  5 Bosnia and Herzegovina       84.8
##  6 Brazil                       83.2
##  7 Bulgaria                     85.5
##  8 Canada                       88.2
##  9 Chile                        86.3
## 10 China                        82  
## # ... with 39 more rows
```

3. Which province has the highest average price? (**10 points**)

```
# Modify data frame to get average price based off province
price <- wine_data %>%
  group_by(province) %>%
  summarise(avg_price = mean(price, na.rm = T))
```

```
## ‘summarise()‘ ungrouping output (override with ‘.groups‘ argument)
```

```
# Print province with highest average price
price %>% top_n(1, avg_price)
```

```
## # A tibble: 1 x 2
##   province   avg_price
##   <chr>          <dbl>
## 1 Santa Cruz      96.2
```

4. Which province in the US has the highest average price? (**10 points**)

```
# Modify data frame, where the country must be the US
US_price <- wine_data[wine_data$country == "US",] %>%
  group_by(province) %>%
  summarise(avg_price = mean(price, na.rm = T))
```

```
## ‘summarise()‘ ungrouping output (override with ‘.groups‘ argument)
```

```
# Print US province with highest average price
US_price %>% top_n(1, avg_price)
```

```
## # A tibble: 1 x 2
##   province avg_price
##   <chr>        <dbl>
## 1 Nevada          75
```

5. From the “designation” variable calculate the number of 20 year old wine (**20 points**)

```
# Use filter() and grepl() to get wines with 20 years in their description
wine20 <- wine_data %>%
  filter(grepl("20-Year|20-year|20-Years|20 Anos|20 Year|20 Years|20-Year-Old|20 yr.
               |20 Yr.|20th Anniversary|20 Anni", wine_data$designation))

# Count total rows
count(wine20)
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1    86
```