Assignment 3

Chang An Le Harry Jr

3/16/2022

Question 1 - Data Manipulation

```
# install.packages("lubridate")
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.4.1     v purrr   1.0.1
## v tibble  3.1.7     v dplyr   1.1.0
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x lubridate::as.difftime() masks base::as.difftime()
## x lubridate::date()        masks base::date()
## x dplyr::filter()          masks stats::filter()
## x lubridate::intersect()   masks base::intersect()
## x dplyr::lag()             masks stats::lag()
## x lubridate::setdiff()     masks base::setdiff()
## x lubridate::union()       masks base::union()
```

```
load("Data/assignment3_data.RDS")
```

1. We can say that the billboard data frame is untidy. Describe the tidy data definitions (on Page 60 of the lecture note Week5_slides.pdf) and explain why billboard violates them.

From the slides, it is known that
1. each variable forms a column,
2. each observation forms a row,
3. each type of observational unit forms a table.

However, the billboard dataframe violates the first condition in particular as the week variable does not form a single column on its own. Instead, creating columns for the rankings for each week (e.g wk1, wk2, wk3 etc) makes the data untidy overall.

2. Tidy up the structure of the billboard dataset and name it as df1. Store the weekly ranking in a variable named rank. Remove the rows from df1 where the rank column is NA. After this step, the first few lines of df1 should read:

```
df1 = billboard %>%
  gather("week","rank",4:79)

df1 = na.omit(df1) #observation count decreases from 24092 to 5307

head(df1)
```

```
## # A tibble: 6 x 5
##   artist              track                              date.entered week   rank
##   <chr>               <chr>                              <date>       <chr> <dbl>
## 1 Destiny's Child     Independent Women Part I           2000-09-23   wk1      78
## 2 Santana             Maria, Maria                       2000-02-12   wk1      15
## 3 Savage Garden       I Knew I Loved You                 1999-10-23   wk1      71
## 4 Madonna             Music                              2000-08-12   wk1      41
## 5 Aguilera, Christina Come On Over Baby (All I Want Is~ 2000-08-05   wk1      57
## 6 Janet               Doesn't Really Matter              2000-06-17   wk1      59
```

3. Convert the week variable to a number.

```
df1$week = gsub("wk","",df1$week)
df1$week = as.numeric(df1$week)

sapply(df1, class) #check if conversion is done correctly
```

```
##       artist        track date.entered         week         rank
##  "character"  "character"       "Date"    "numeric"    "numeric"
```

4. For this question, we will use the lubridate package. Read the package description by typing ?lubridate in your RStudio Console. The package provides tools that enables us to manipulate dates. Specifically, we would like to extract information from the date.entered variable of df1 and create the following variables:

- day: the day component of the date.entered variable
- month: month of the date
- year: year of the date
- day_of_week: the day of the week as an ordered factor variable
- After creating the new variables, remove the column date.entered and then overwrite the original df1 data frame.

```
?lubridate
```

```
df1a = df1 %>% #creating the different columns, storing into temp dataframe (df1a)
  mutate(day = mday(date.entered)) %>%
  mutate(month = month(date.entered)) %>%
  mutate(year = year(date.entered)) %>%
  mutate(day_of_week = wday(date.entered, label = TRUE, abbr = TRUE))

df1 = subset(df1a, select = -c(date.entered)) #remove date.entered column, overwrite df1
head(df1) #to check
```

```
## # A tibble: 6 x 8
##   artist              track             week  rank   day month  year day_of_week
##   <chr>               <chr>            <dbl> <dbl> <int> <dbl> <dbl> <ord>
## 1 Destiny's Child     Independent Wom~     1    78    23     9  2000 Sat
## 2 Santana             Maria, Maria         1    15    12     2  2000 Sat
## 3 Savage Garden       I Knew I Loved ~     1    71    23    10  1999 Sat
## 4 Madonna             Music                1    41    12     8  2000 Sat
## 5 Aguilera, Christina Come On Over Ba~     1    57     5     8  2000 Sat
## 6 Janet               Doesn't Really ~     1    59    17     6  2000 Sat
```

5. Below is a diagram for the variables in the df1 and songs tables. Identify the primary keys in the two tables, and check if they are able to uniquely identify the observations in the datasets. Illustrate (in words) the connections between the two tables.

From the two tables, the primary keys would be the track title and the artist name. To verify this, we can do the following:

```
songs %>%
  count(artistname, track) %>% filter(n >1) #primary keys can uniquely identify observations
```

```
## # A tibble: 0 x 3
## # ... with 3 variables: artistname <chr>, track <chr>, n <int>
```

```
df1 %>%
  count(artist, track) %>% filter(n >1) #primary keys unable to uniquely identify observations
```

```
## # A tibble: 313 x 3
##    artist           track                                                 n
##    <chr>            <chr>                                             <int>
##  1 "2 Pac"          Baby Don't Cry (Keep Ya Head Up II)                   7
##  2 "2Ge+her"        The Hardest Part Of Breaking Up (Is Getting Back Your~     3
##  3 "3 Doors Down"   Kryptonite                                           53
##  4 "3 Doors Down"   Loser                                                20
##  5 "504 Boyz"       Wobble Wobble                                        18
##  6 "98\xa1"         Give Me Just One Night (Una Noche)                   20
##  7 "A*Teens"        Dancing Queen                                         5
##  8 "Aaliyah"        I Don't Wanna                                        20
##  9 "Aaliyah"        Try Again                                            32
## 10 "Adams, Yolanda" Open My Heart                                        20
## # ... with 303 more rows
```

As seen above, the identified primary keys can uniquely identify the observations in the songs dataset, but not in the df1 dataset. This is because in the df1 dataset, the number of observations for each unique combination of artist name and track title represents the number of weeks that each song has remained on the Billboard Top 100 List.

Also, to illustrate the connection between the two datasets, the songs dataset represents a subset of the full list of songs that appear in the df1 dataset, in which the latter represents the 317 songs that entered the Billboard Top 100 List.

6. Use an inner join to join df1 to the songs data frame and name the new object as df2. Describe (in words) what df2 contains. Display the summary statistics of the new object using the summary() function.

```
df2 = df1 %>%
  inner_join(songs, by = c("artist" = "artistname", "track" = "track"))
```

```
summary(df2)
```

```
##     artist             track                week            rank
##  Length:2175        Length:2175        Min.   : 1.00   Min.   :  1.00
##  Class :character   Class :character   1st Qu.: 5.00   1st Qu.: 21.50
##  Mode  :character   Mode  :character   Median :11.00   Median : 43.00
##                                        Mean   :13.11   Mean   : 46.43
##                                        3rd Qu.:18.00   3rd Qu.: 71.00
##                                        Max.   :65.00   Max.   :100.00
##
##       day             month             year      day_of_week     time
##  Min.   : 1.00   Min.   : 1.000   Min.   :1999   Sun:   0    Length:2175
##  1st Qu.: 8.00   1st Qu.: 4.000   1st Qu.:2000   Mon:   0    Class :character
##  Median :15.00   Median : 6.000   Median :2000   Tue:   0    Mode  :character
##  Mean   :15.19   Mean   : 6.173   Mean   :2000   Wed:   0
##  3rd Qu.:22.00   3rd Qu.: 9.000   3rd Qu.:2000   Thu:   0
##  Max.   :30.00   Max.   :12.000   Max.   :2000   Fri:   0
##                                                  Sat:2175
##      mode               key             popularity
##  Length:2175        Length:2175        Min.   : 9.00
```

```
##  Class :character   Class :character   1st Qu.:47.00
##  Mode  :character   Mode  :character   Median :55.00
##                                        Mean   :55.65
##                                        3rd Qu.:65.00
##                                        Max.   :82.00
##
```

df2 contains the list of songs that appear in both the songs dataset and the Billboard Top 100 List. It includes the weekly rankings of the 117 songs from the songs dataset that entered the weekly Billboard Top 100 List.

Besides the obvious features such as track duration, artist name, track title, the rank column represent each song's weekly ranking in the Billboard Top 100 List each week while the popularity column was extracted based on the internal ranking amongst the 117 songs in the songs dataset (based on popularity).

Question 2 - Visualization Write-Up

Major League Soccer: Team Performance vs Salary

As a sports fan, I've always wondered how much sport athletes would earn for a living, and whether their individual and/or team performance affects their pay. Thus, this visualisation on Major League Soccer piqued my interest. It aims to compare the 2018 season rankings of the 23 soccer clubs, as well as their salary rankings for the year. Lines are drawn between the two columns of season ranking and total base salary ranking, in an attempt to investigate the relationship between the two variables amongst the 23 clubs.

In my opinion, I generally dislike the way that the visualization was presented. Firstly, the intersecting lines may make it confusing overall to take note the exact rankings of both variables for each club, especially when there are so many clubs in the league. I would say that this form of visualization is still appropriate for 10 clubs or less, but looking at this for 23 clubs complicates interpretations in my opinion. Also, there seems to be no explanation on why the lines have varying thicknesses. Nonetheless, one merit that can be pointed out is the color-coding of the lines drawn, with red representing a club having a season ranking less than or equal to its total salary ranking, while blue represents the opposite.

The visualization being discussed is in the form of a bump chart, which is mainly used to represent changes of one type of rank over time. However, since this is not a time series plot, it would be more appropriate to represent the data using a scatterplot instead, where each point can effectively distinguish the identity of each club. This, in my opinion, would be a more effective way of comparing the 2 different variables when visually representing this relationship.