DSO 545 Take Home Final

Na Li

December 3, 2017

I. Case 01 Icecream

1. Aggregate the dataset to find the average spending on icecream for all listed countries over the given years.

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

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

```r
icecream<-read.csv("icecream.csv")
##use group_by and summarize to aggregate spending data by countries
aggreg<-icecream %>%
  group_by(Country.or.Area) %>%
  summarize(spending=mean(USDinMillions))
head(aggreg)
```

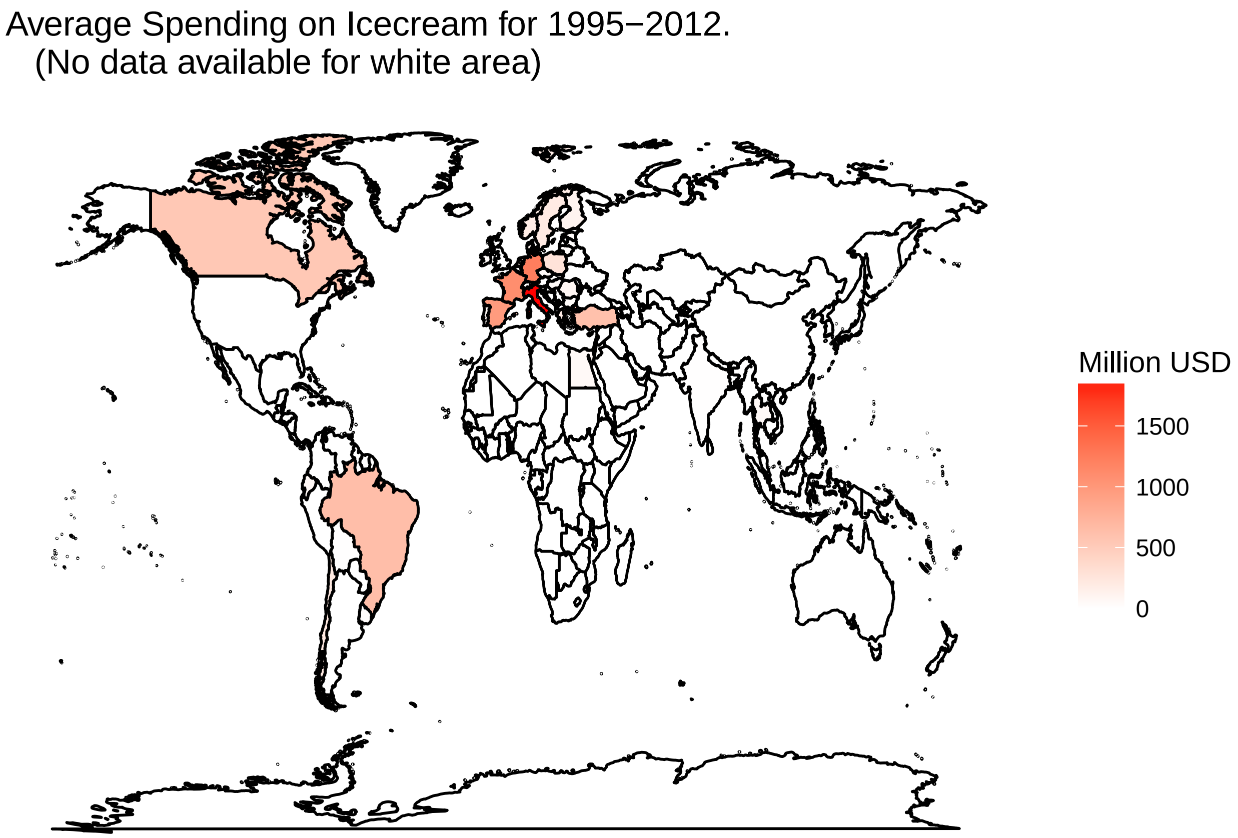
```
## # A tibble: 6 x 2
##   Country.or.Area   spending
##            <fctr>      <dbl>
## 1         Albania   3.225439
## 2         Bolivia   5.487864
## 3          Brazil 637.807535
## 4        Bulgaria  32.672548
## 5          Canada 545.672454
## 6           Chile 208.901264
```

2. Create a chloropleth map that shows the average spending on icecream for the listed countiresover the given years.

```r
library(ggplot2)
library(stringr)
##read in the world map data and merge it with icecream spending data by country
world_map=map_data("world")
aggreg$Country.or.Area<-as.character(aggreg$Country.or.Area)
icecream_world<-left_join(x=world_map, y=aggreg, by=c("region"="Country.or.Area"))

##is spending value is NA, convert the value to 0
icecream_world<-mutate(icecream_world, spending=ifelse(is.na(spending), 0,spending))
```

```
###create the map
ggplot(icecream_world, aes(x=long, y=lat, group=group, fill=spending))+
  geom_polygon(color="black")+
  scale_fill_gradient(low="white", high="red", name="Million USD")+
  labs(title="Average Spending on Icecream for 1995-2012. \n   (No data available for white area)", x="
  theme_void()
```


Average Spending on Icecream for 1995−2012.
(No data available for white area)
Million USD
1500
1000
500
0

II. Case 02

1. Use rvest R package to scrape the data table. Save it to players.

```
library(rvest)
```

```
## Loading required package: xml2
```

```
##save the page url
page<-"https://en.wikipedia.org/wiki/Designated_Player_Rule"
###scrape web table
players<-page %>%
  read_html() %>%
  html_node("table") %>%
  html_table()
```

2. Clean the column with compensation information. Change the column type to numeric, and rename it Compensation.

```
library(stringr)
##use str_replace_all to remove $ and comma in the string and convert the variable
##into a numeric one
players<-players %>% mutate(Compensation=as.numeric(
  str_replace_all(`2017 Guaranteed compensation[14]`,
                  pattern="\\$|,",
                 replacement="")))
```

```
## Warning in evalq(as.numeric(str_replace_all(`2017 Guaranteed
## compensation[14]`, : NAs introduced by coercion
```

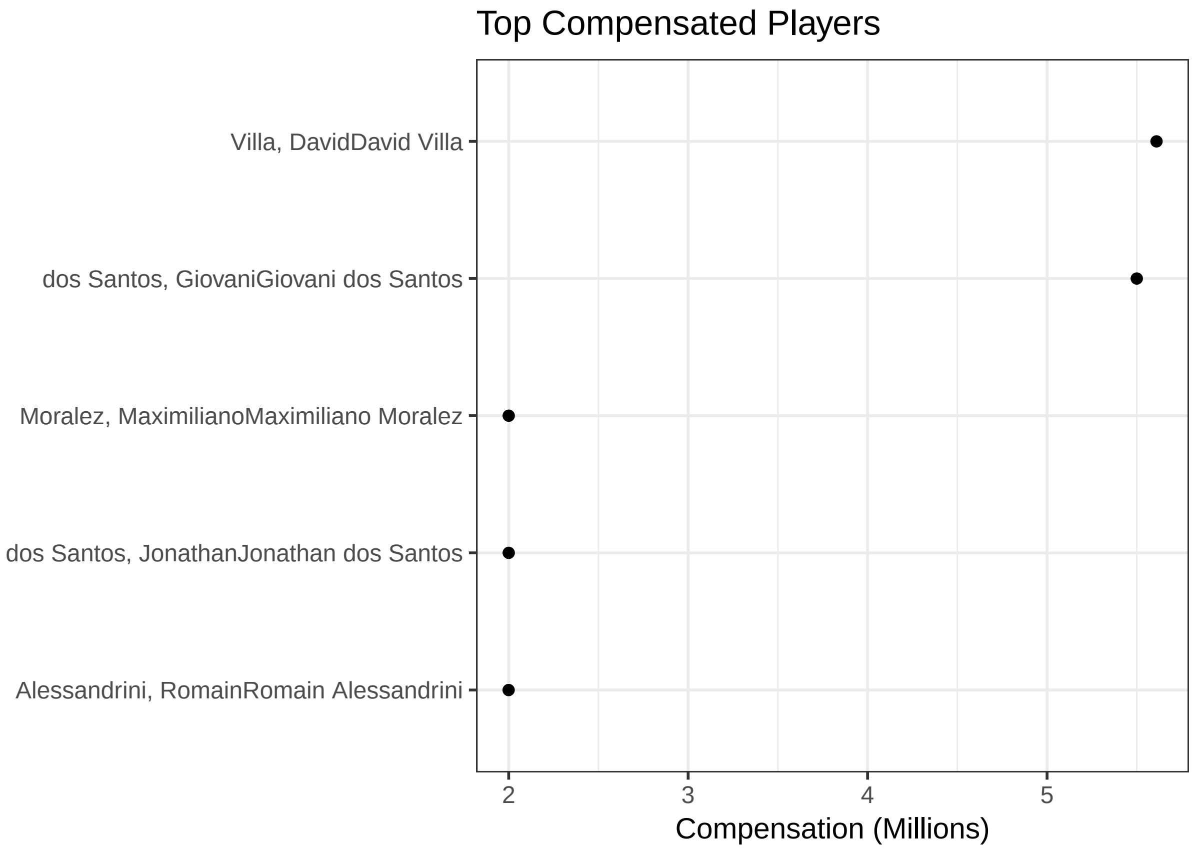
3. Create a subset of players called NYLAplayers, which only contains records of players currently playing for New York City FC or LA Galaxy, and order your subset by Compensation in decreasing order.

```
##change the column names of players dataset
colnames(players)=c("year", "player", "nation", "club", "compensationText", "Compensation")

##get the rows of players belonging to "New York City FC" and "LA Galaxy"
NYLAplayers<-players %>%
  filter(club%in% c("New York City FC", "LA Galaxy")) %>%
  select(c("year", "player", "club", "Compensation")) %>%
  arrange(desc(Compensation))
```

4. Visualize the NYLAplayers compensation

```
library(ggplot2)
ggplot(NYLAplayers, aes(x=Compensation, y=reorder(player, Compensation)))+
  geom_point()+
  ###the following code change the break labels
  scale_x_continuous(breaks=c(2e6, 3e6, 4e6, 5e6),labels=c("2", "3", "4", "5"))+
  labs(title="Top Compensated Players",
       x="Compensation (Millions)",
       y="")+
  theme_bw()
```



III. Case 03: How much does Joey Owe Chandler in Friends TV Show?

```r
library(stringr)
fileName<-"friends.txt"
text<-readChar(fileName,file.info(fileName)$size)
##create a parser using regular expression
parser="\\$[0-9]+"
###extract all the information related to money using the parser, convert the list to a vector
dollars<-unlist(str_extract_all(string = text, pattern=parser))
dollars<-as.numeric(str_replace_all(dollars, "\\$", ""))
##sum up the numbers
sum(dollars)
```

```
## [1] 91760
```

Answer: Joey owes Chandler $91760