DA5020 - Homework 5: Dates and Times

2018-02-14

```
knitr::opts_chunk$set(error = TRUE)
```

Continue working with Farmers Market data from last week.

This week's assignment is not only about dates and times, but also what you learnt from past weeks: data transformation, strings, and more.

You may also need to go through a review on R control statesments since they will come handy in solving some of the problems.

Questions

1. (10 points) Add a new column Season1Days that contains the number of days a market is opened per week (for the dates it is open).

```
library(tidyverse)

## ── Attaching packages
─────────────────────────────────────────────────────── tidyverse 1.2.1 ──

## ✔ ggplot2 2.2.1     ✔ purrr   0.2.4
## ✔ tibble  1.4.1     ✔ dplyr   0.7.4
## ✔ tidyr   0.7.2     ✔ stringr 1.2.0
## ✔ readr   1.1.1     ✔ forcats 0.2.0

## ── Conflicts
────────────────────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()

library(lubridate)

##
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##     date

library(stringr)
farmers_market.csv <- read_csv("Export.csv")

## Parsed with column specification:
## cols(
```

```
##   .default = col_character(),
##   FMID = col_integer(),
##   x = col_double(),
##   y = col_double()
## )

## See spec(...) for full column specifications.

farmers_market.csv$Season1Days <-
str_count(farmers_market.csv$Season1Time, "Mon|Tue|Wed|Thu|Fri|Sat|Sun")
```

2. (10 points) Add a new column WeekendOpen indicating whether a market opens during weekends in Season1.

```
farmers_market.csv$WeekendOpen <-
str_count(farmers_market.csv$Season1Time, "Sat|Sun")
WO <- farmers_market.csv$WeekendOpen
WO <- gsub("0", "No", WO)
WO <- gsub("1|2", "Yes", WO)
farmers_market.csv$WeekendOpen <- WO
```

3. (20 points) Find out which markets close before 6PM, and which open only for fewer than 4 hours a day. For simplicity, consider only Season1Time. For markets with different open hours across a week, use the average length of open hours for the days they actually open.

```
#select FMID, season1time.
FMID_time <- farmers_market.csv %>% select(FMID,Season1Time)

#Replace days with space
Temp1 <-str_replace_all(FMID_time$Season1Time,"^[A-Za-z:]+[ ]|^[A-Za-z]+
[:]","")

#Remove ";" to clean the data
Temp1.1 <- str_replace(Temp1,";","")

#Extract opening time
Temp2 <- str_extract(Temp1,"^[0-9]+[:][0-9]+[ ][A-Za-z]+")

#Convert to 24 hour format
Temp3 <- parse_time (Temp2, "%I:%M %p")

## Warning: 1 parsing failure.
## row # A tibble: 1 x 4 col     row   col expected          actual  expected
<int> <int> <chr>           <chr>   actual 1  6943    NA time like %I:%M %p
0:00 pm

#Extract closing time
Temp4 <- str_extract(Temp1.1 , "[0-9]+[:][0-9]+[ ][A-Za-z]+$")
```

```
#Convert to 24 hour format
Temp5 <- parse_time ( Temp4, "%I:%M %p")

## Warning in rbind(names(probs), probs_f): number of columns of result is
not
## a multiple of vector length (arg 1)

## Warning: 2 parsing failures.
## row # A tibble: 2 x 4 col     row   col expected          actual  expected
<int> <int> <chr>             <chr>   actual 1   589    NA time like %I:%M %p
0:00 AM row 2  1315    NA time like %I:%M %p 0:00 pm

#Defined a new variable for the time 6:00 PM.
pt <- parse_time("6:00 PM")

#NDF is a new dataframe containing the opening and closing times.
NDF <-mutate(FMID_time , Open = Temp3, Close = Temp5)

#Filtered to get the markets which are open for less than 4 hours(14400
seconds)
Less_4 <- filter(NDF, abs(parse_time(Open)-parse_time(Close)) < 14400)

#Filtered by Closing time < 6 to get the markets which close before 6
hours_6 <- filter(NDF, parse_time(Close) < pt)
```

4. (40 Points) The seasons are not standardized and would make analysis difficult. Create four new columns for four seasons (Spring, Summer, Fall, Winter), indicating whether a market is available in that season. Also, create two additional columns HalfYear and YearRound to identify those who open across seasons. Define “half year” and “year round” on your own terms, but explain them before you write the code (or as comments in your code). (Hint: you may want to create even more auxiliary columns, Season1BeginDate and Season1EndDate for example.)

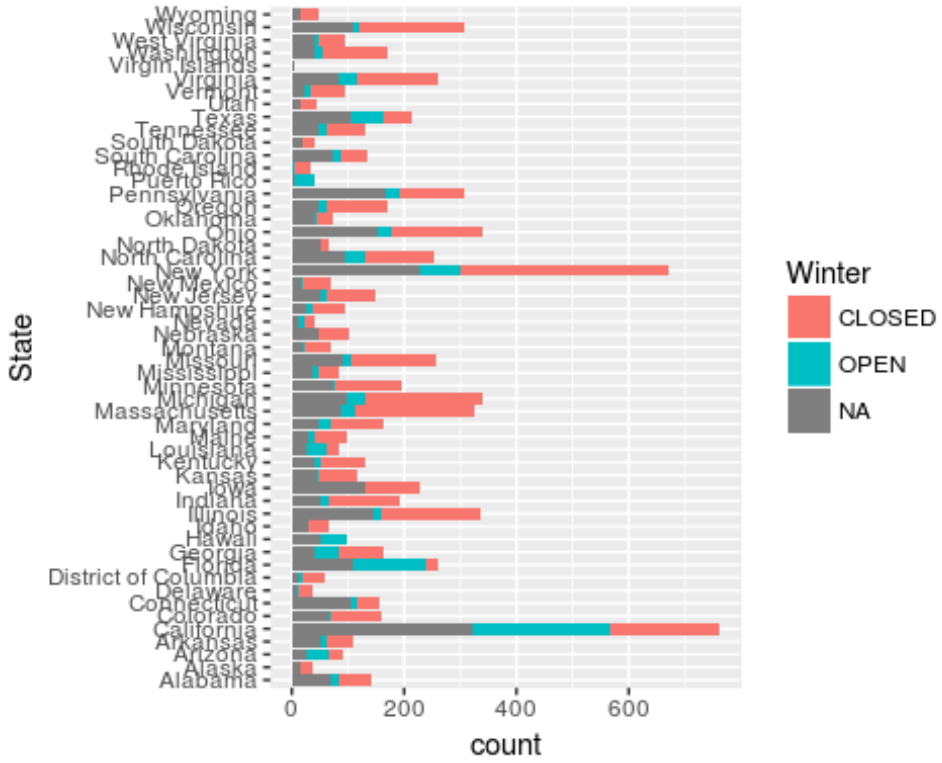
```
# Replace the month names in Season1Date with dates to make it easier to
calculate the duration
farmers_market.csv$Season1Date <-
 str_replace_all(
 farmers_market.csv$Season1Date,
 c(
 "January" = "01/01/2017",
 "February" = "02/01/2017",
 "March" ="03/01/2017",
 "April" = "04/01/2017",
 "May" = "05/01/2017",
 "July" = "07/01/2017",
 "June" = "06/01/2017",
 "August" = "08/01/2017",
 "September" ="09/01/2017",
 "October" = "10/01/2017",
```

```
 "November" ="11/01/2017",
 "December" = "12/01/2017"
 ))

# Extract the opening date of markets
season1begin <- str_extract(farmers_market.csv$Season1Date, "^[0-9]+[/]
[0-9]+[/][0-9]+")

# Extract the closing date of markets
season1end <- str_extract(farmers_market.csv$Season1Date, "[0-9]+[/][0-
9]+[/][0-9]+$")

# Change all the years to same year(2017) to make analysis easy.
s1b <- str_replace(season1begin, "[0-9]+$", "2017")
s1e <- str_replace(season1end, "[0-9]+$", "2017")

# Parse the date to the 24 hour format
s1b <- parse_date(s1b, "%m/%d/%Y")
s1e <- parse_date(s1e , "%m/%d/%Y")

## Warning: 16 parsing failures.
## row # A tibble: 5 x 4 col     row   col expected   actual     expected   <int>
<int> <chr>      <chr>      actual 1    78    NA valid date 11/31/2017 row 2
162    NA valid date 09/31/2017 col 3   682    NA valid date 09/31/2017
expected 4  1324    NA valid date 02/29/2017 actual 5  1803    NA valid date
09/31/2017
## ... ................. ... ..... ........ ..... ... ..... ... ..... ... ..... ........ .
..... ..... .....
## See problems(...) for more details.

# create date ranges for the four seasons
winter_start <- as.Date("2017-12-01") # Defining winter start date
spring_start <- as.Date("2017-03-01") #Defining spring start date.
summer_start <- as.Date("2017-06-01") #Defining summer start date./
fall_start <- as.Date("2017-09-01") # Defining fall start date.
winter <- interval(winter_start, spring_start + years(1))
spring <- interval(spring_start, summer_start)
summer <- interval(summer_start, fall_start)
fall <- interval(fall_start, winter_start)

# Half year are the markets that are open for 6 months
Halfyear <- interval(s1b, s1b + months(6))

#Added 3 months to make it a year
yearRound <- interval(winter_start, fall_start + months(3))

farmers_market.csv <- farmers_market.csv %>%
mutate(
Season1BeginDate = s1b,
```

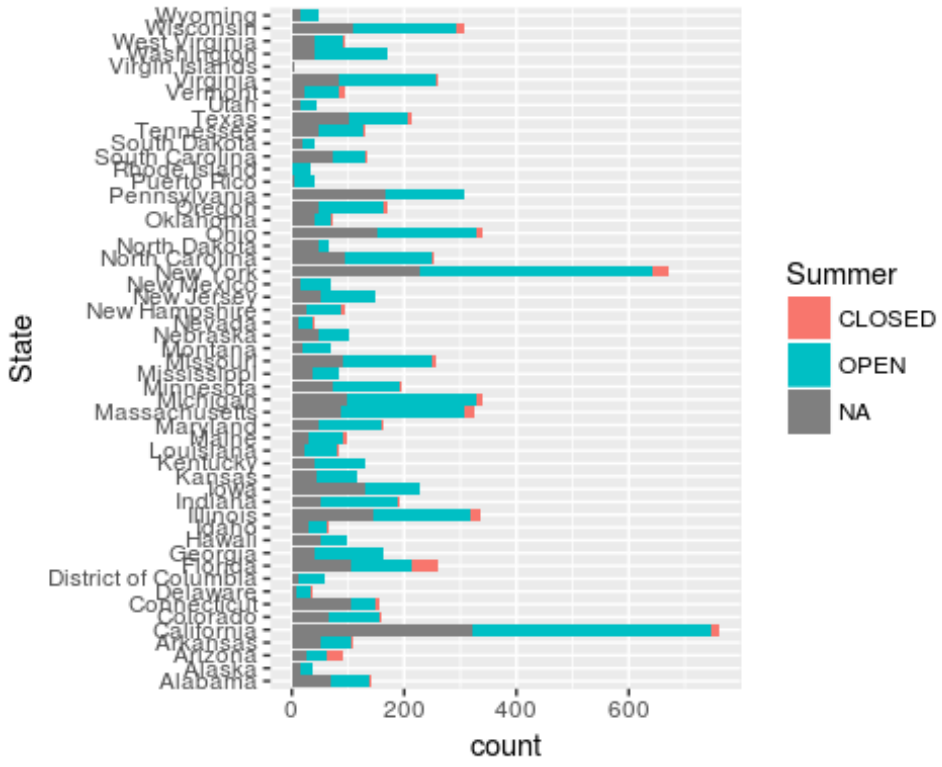
```
Season1EndDate = s1e ,
Season1EndDate = if_else(
Season1EndDate < Season1BeginDate,
Season1EndDate + years(1),
Season1EndDate
),
Season1DateRange = interval(Season1BeginDate, Season1EndDate),
Winter = int_overlaps(Season1DateRange, winter),
Spring = int_overlaps(Season1DateRange, spring),
Summer = int_overlaps(Season1DateRange, summer),
Fall = int_overlaps(Season1DateRange, fall),
YearRound = int_overlaps(Season1DateRange, yearRound),
Halfyear = int_overlaps(Season1DateRange, Halfyear)
)

farmers_market.csv$Summer <- gsub("TRUE","OPEN",
farmers_market.csv$Summer)
farmers_market.csv$Summer <- gsub("FALSE","CLOSED",
farmers_market.csv$Summer)
farmers_market.csv$Spring <- gsub("TRUE","OPEN",
farmers_market.csv$Spring)
farmers_market.csv$Spring <- gsub("FALSE","CLOSED",
farmers_market.csv$Spring)
farmers_market.csv$Winter <- gsub("TRUE","OPEN",
farmers_market.csv$Winter)
farmers_market.csv$Winter <- gsub("FALSE","CLOSED",
farmers_market.csv$Winter)
farmers_market.csv$Fall <- gsub("TRUE","OPEN", farmers_market.csv$Fall)
farmers_market.csv$Fall <- gsub("FALSE","CLOSED", farmers_market.csv$Fall)
farmers_market.csv$YearRound <- gsub("TRUE","OPEN",
farmers_market.csv$YearRound)
farmers_market.csv$YearRound <- gsub("FALSE","CLOSED",
farmers_market.csv$YearRound)
farmers_market.csv$Halfyear <- gsub("TRUE","OPEN",
farmers_market.csv$Halfyear)
farmers_market.csv$Halfyear <- gsub("FALSE","CLOSED",
farmers_market.csv$Halfyear)

head(farmers_market.csv)

## Note: method with signature 'Interval#ANY' chosen for function '-',
##  target signature 'Interval#Interval'.
##  "ANY#Interval" would also be valid

## Error in round_x - lhs: Arithmetic operators undefined for 'Interval' and
'Interval' classes:
##   convert one to numeric or a matching time-span class.
```

5. (20 points) *Open question*: explore the new variables you just created. Aggregate them at different geographic levels, or some other categorical variable. What can you discover?

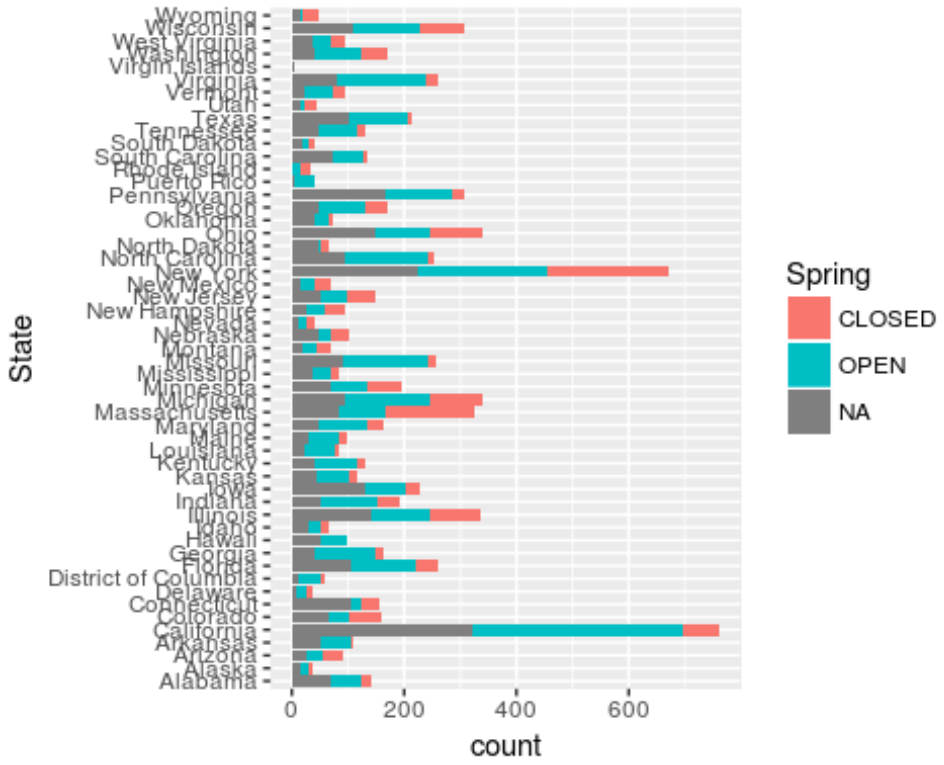
```
ggplot(data = farmers_market.csv)+
  geom_bar(mapping = aes(x = State,fill = Winter)) +
  coord_flip()
```



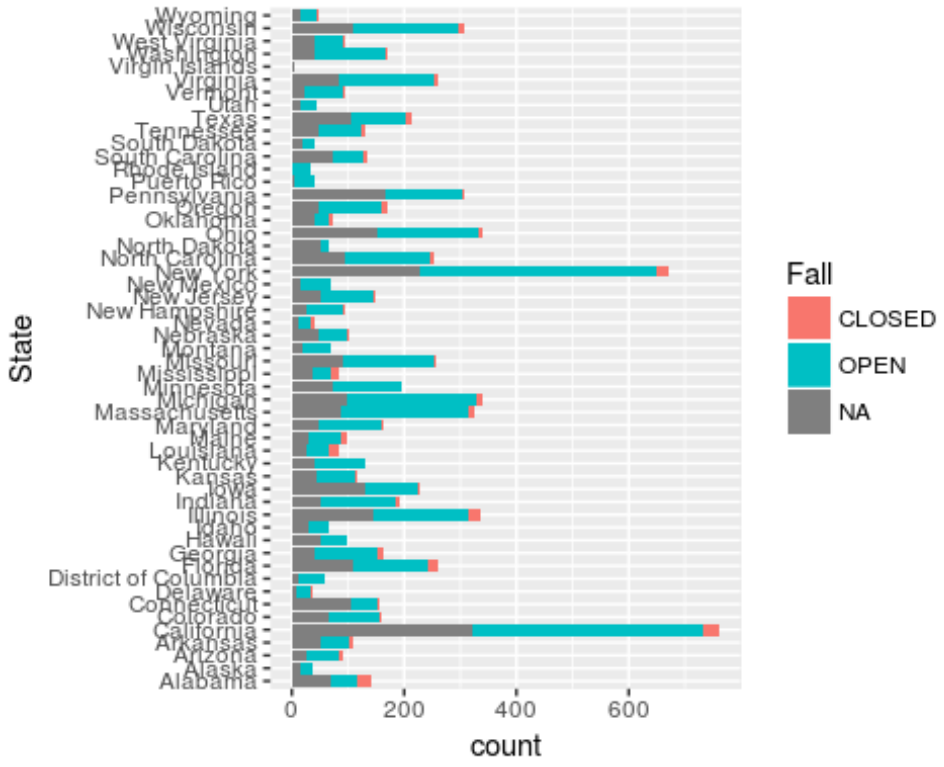

```
#We can see that the proportion of markets which are open in winter are
much lesser than those which are closed.
#This could be because not a lot of farm products are obtained in winter and
also because the snow and low temperature is not conducive for the
optimum yield.
#However a big proportion of markets are open in California. This is probably
because of the temperature is not too low in the winter in that state.

ggplot(data = farmers_market.csv)+
  geom_bar(mapping = aes(x = State,fill = Summer)) +
  coord_flip()
```



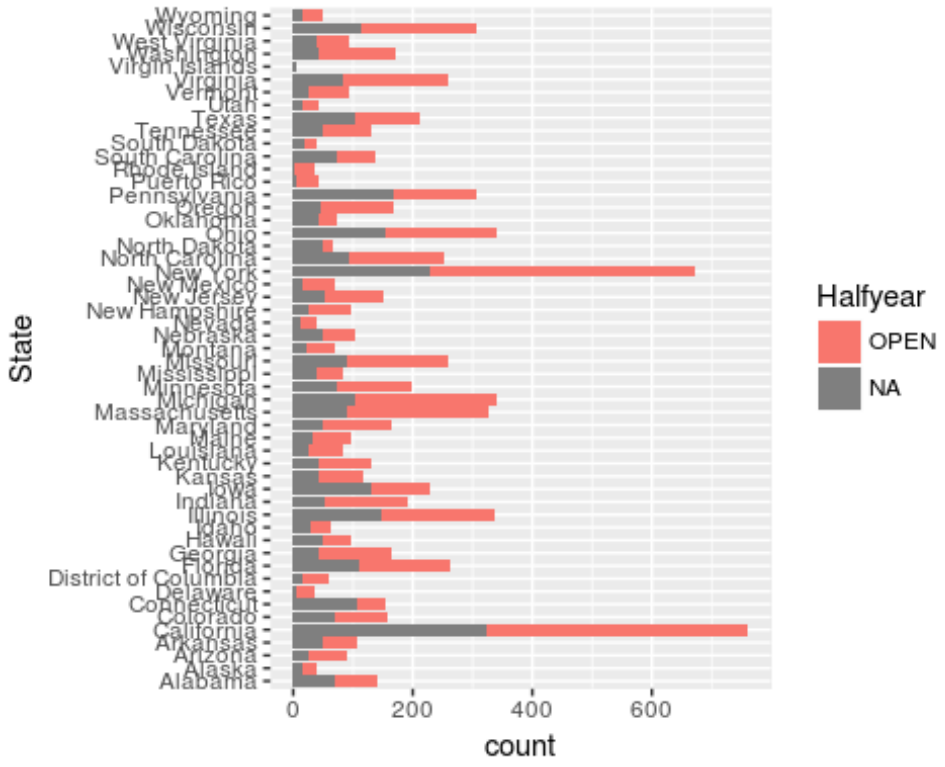

```
# Most of the markets are open in summer in all the states.
# This shows that a lot of produce is available in summer.

ggplot(data = farmers_market.csv)+
  geom_bar(mapping = aes(x = State,fill = Spring)) +
  coord_flip()
```



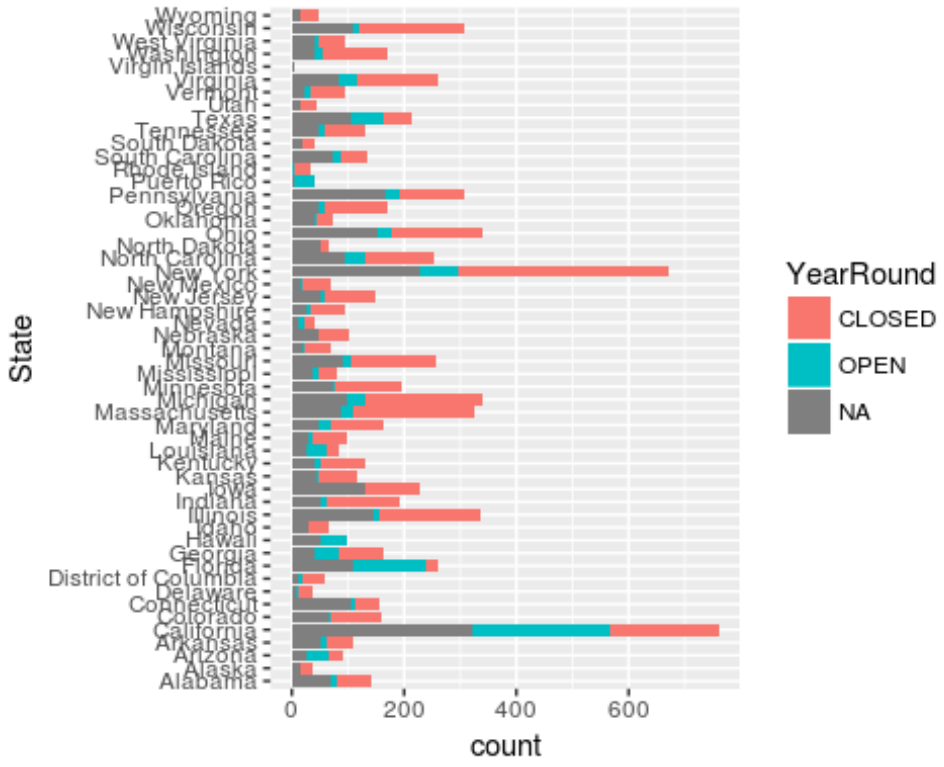

```
#Slightly less proportion of markets are open as compared to summer but
they are still a lot more than winter

ggplot(data = farmers_market.csv)+
  geom_bar(mapping = aes(x = State,fill = Fall)) +
  coord_flip()
```




```
#This clearly shows that in all the states across US, the fall is the season in which almost all the markets are open
#There is a minuscle number of makets which are closed and they are uniform across all the states

ggplot(data = farmers_market.csv)+
  geom_bar(mapping = aes(x = State,fill = Halfyear)) +
  coord_flip()
```




```
#All the farmers market are open for half a year with the most number of markets in New York and California

ggplot(data = farmers_market.csv)+
  geom_bar(mapping = aes(x = State,fill = YearRound)) +
  coord_flip()
```




```
# A high proportion of markets are not open the year round.
# Out of all the states, California has the most number of markets open throughtout the year
```

Submission

You need to submit an .Rmd extension file as well as the generated pdf file. Be sure to state all the assumptions and give explanations as comments in the .Rmd file wherever needed to help us assess your submission. Please name the submission file LAST_FirstInitial_1.Rmd for example for John Smith's 1st assignment, the file should be named Smith_J_1.Rmd.