Dsouza_Clinton_Assignment#5

Clinton Dsouza

10/6/2019

GitHub Link: https://github.com/cdsouza94/R--Projects

Let's load the libraries required for this task

```
library(tidyverse)
```

```
## -- Attaching packages ----------------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   0.8.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

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

```
library(lubridate)
```

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

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

```
library(stringr)
```

Next, lets load the farmers market data

```
getwd()
```

```
## [1] "C:/Users/dsouz/OneDrive/Desktop/Assignment#5"
```

```
setwd("C:\\Users\\dsouz\\Desktop")
farmermkt <- read_csv("farmers_market.csv.csv", col_types = cols(Season4Date = col_character(), Season4
head(farmermkt,50)
```

```
## # A tibble: 50 x 59
##      FMID MarketName Website Facebook Twitter Youtube OtherMedia street
##     <dbl> <chr>      <chr>   <chr>    <chr>   <chr>   <chr>      <chr>
##  1 1.02e6 Caledonia~ https:~ https:/~ <NA>    <NA>    <NA>       <NA>
##  2 1.02e6 Stearns H~ http:/~ Stearns~ <NA>    <NA>    <NA>       6975 ~
```

```
##  3 1.01e6 106 S. Ma~ http:/~ <NA>     <NA>     <NA>     <NA>       106 S~
##  4 1.01e6 10th Stee~ <NA>    <NA>     <NA>     <NA>     http://ag~ 10th ~
##  5 1.00e6 112st Mad~ <NA>    <NA>     <NA>     <NA>     <NA>       112th~
##  6 1.01e6 12 South ~ http:/~ 12_Sout~ @12sou~ <NA>     @12southf~ 3000 ~
##  7 1.01e6 125th Str~ http:/~ https:/~ https:~ <NA>     Instagram~ 163 W~
##  8 1.01e6 12th & Br~ <NA>    https:/~ <NA>     <NA>     https://w~ 12th ~
##  9 1.01e6 14&U Farm~ <NA>    https:/~ https:~ <NA>     <NA>       1400 ~
## 10 1.01e6 14th & Ke~ <NA>    https:/~ 14KenFM <NA>     instagram~ 5500 ~
## # ... with 40 more rows, and 51 more variables: city <chr>, County <chr>,
## #   State <chr>, zip <chr>, Season1Date <chr>, Season1Time <chr>,
## #   Season2Date <chr>, Season2Time <chr>, Season3Date <chr>,
## #   Season3Time <chr>, Season4Date <chr>, Season4Time <chr>, x <dbl>,
## #   y <dbl>, Location <chr>, Credit <chr>, WIC <chr>, WICcash <chr>,
## #   SFMNP <chr>, SNAP <chr>, Organic <chr>, Bakedgoods <chr>,
## #   Cheese <chr>, Crafts <chr>, Flowers <chr>, Eggs <chr>, Seafood <chr>,
## #   Herbs <chr>, Vegetables <chr>, Honey <chr>, Jams <chr>, Maple <chr>,
## #   Meat <chr>, Nursery <chr>, Nuts <chr>, Plants <chr>, Poultry <chr>,
## #   Prepared <chr>, Soap <chr>, Trees <chr>, Wine <chr>, Coffee <chr>,
## #   Beans <chr>, Fruits <chr>, Grains <chr>, Juices <chr>,
## #   Mushrooms <chr>, PetFood <chr>, Tofu <chr>, WildHarvested <chr>,
## #   updateTime <chr>
```

Q1) Find the number of days the markets are open per week
`r DaysOpen <- gsub("[(AM|PM) a-z 0-9]","", farmermkt$Season1Time)` Using gsub here to remove everything except the initial of days.
`r farmermkt %>% transmute(Season1Time, Season1Days = (str_count(DaysOpen, "[A-Z a-z]")))`

```
## # A tibble: 8,788 x 2 ##    Season1Time
Season1Days ##    <chr>                                                <int> ##  1 Wed:
9:00 AM-1:00 PM;                                  1 ##  2 Sat: 9:00 AM-1:00 PM;
1 ##  3 <NA>                                                NA ##  4 Wed: 3:00
PM-6:00 PM;Sat: 8:00 AM-1:00 PM;            2 ##  5 Tue:8:00 am - 5:00 pm;Sat:8:00 am
- 8:00 pm;           2 ##  6 Tue: 3:30 PM-6:30 PM;                                1
##  7 Tue: 10:00 AM-7:00 PM;                                  1 ##  8 Fri: 8:00
AM-11:00 AM;                                  1 ##  9 Sat: 9:00 AM-1:00 PM;
1 ## 10 Sat: 9:00 AM-1:00 PM;                                   1 ## # ... with 8,778
more rows
```

Q2) Indicate if a market is open during the weekends

Considering weekend as a Saturday and Sunday

```
weekend <- gsub("[a-z ^s 0-9 MTWTF]", "",farmermkt$Season1Time)
```

In the above step, I have eliminated all alphabets except lower and upper 's' and then count their occurences as below

```
farmermkt %>%
  transmute(Season1Time, WeekendOpen = str_count(weekend, "[(^S)|(^s)]"))
```

```
## # A tibble: 8,788 x 2
##    Season1Time                                  WeekendOpen
```

```
##    <chr>                                                  <int>
##  1 Wed: 9:00 AM-1:00 PM;                                      0
##  2 Sat: 9:00 AM-1:00 PM;                                      1
##  3 <NA>                                                      NA
##  4 Wed: 3:00 PM-6:00 PM;Sat: 8:00 AM-1:00 PM;                 1
##  5 Tue:8:00 am - 5:00 pm;Sat:8:00 am - 8:00 pm;               1
##  6 Tue: 3:30 PM-6:30 PM;                                      0
##  7 Tue: 10:00 AM-7:00 PM;                                     0
##  8 Fri: 8:00 AM-11:00 AM;                                     0
##  9 Sat: 9:00 AM-1:00 PM;                                      1
## 10 Sat: 9:00 AM-1:00 PM;                                      1
## # ... with 8,778 more rows
```

From the above result: 1 indicates that the market is open on a Saturday or a Sunday. 2 indicates its open on both days. 0 indicates it is not open on weekends. This can also be done using str_detect() which will give a TRUE/FALSE result.

Q3) First, I have created two different columns for the Start Time and End Time of every Market. Below are the steps taken to do that:
Following a regex pattern to remove all unnecessary words and keep only the time data for this analysis

```
r Times <-
gsub("(?i)(Mon)|(Tue)|(Wed)|(Thu)|(Fri)|(Sat)|(Sun)","",farmermkt$Season1Time)
```

Making a new column for start time of the market Creating a regex which will keep only the Start Time:

```
r StartTime <- gsub("\\-(.*)","", Times) StartTime <- gsub("^:?","", StartTime)
StartTime <- parse_time(StartTime)
## Warning: 5 parsing failures. ##  row col   expected actual ##   62  -- time like
3pm ## 4376  -- time like     TBD ## 8357  -- time like     2 ## 8773  -- time like
TBD ## 8774  -- time like     TBD
r head(StartTime, 20)
## 09:00:00 ## 09:00:00 ##       NA ## 15:00:00 ## 08:00:00 ## 15:30:00 ## 10:00:00 ##
08:00:00 ## 09:00:00 ## 09:00:00 ## 09:00:00 ## 14:30:00 ## 08:00:00 ## 15:00:00 ##
15:00:00 ## 15:00:00 ## 08:00:00 ## 08:00:00 ## 08:30:00 ## 11:00:00
```

Making a new column for end time of the market Creating a regex which will keep only the End Time:

```
r EndTime <- gsub("(\\;).*","",Times) EndTime <- gsub("(.*)\\-","",EndTime) EndTime <-
parse_time(EndTime)
## Warning: 5 parsing failures. ##  row col   expected actual ##   62  -- time like
6pm ## 4376  -- time like     TBD ## 8357  -- time like     6 ## 8773  -- time like
TBD ## 8774  -- time like     TBD
r head(EndTime, 20)
## 13:00:00 ## 13:00:00 ##       NA ## 18:00:00 ## 17:00:00 ## 18:30:00 ## 19:00:00 ##
11:00:00 ## 13:00:00 ## 13:00:00 ## 13:30:00 ## 18:30:00 ## 18:00:00 ## 19:00:00 ##
18:30:00 ## 19:00:00 ## 11:00:00 ## 12:00:00 ## 13:00:00 ## 18:00:00
```

Details of Markets which close before 6PM are as follows: I have used the filter() along with the hour() to pull out the exact times as needed from the column

```
r farmermkt %>% transmute(FMID, MarketName, Season1Time, StartTime, EndTime) %>%
filter(hour(EndTime) < 18)
```

```
## # A tibble: 3,848 x 5 ##       FMID MarketName              Season1Time
StartTime EndTime ##      <dbl> <chr>                     <chr>                    <time>
<time> ##  1 1018261 Caledonia Farmers Mark~ Wed: 9:00 AM-1:00 PM;  09:00     13:00 ##
2 1018318 Stearns Homestead Farm~ Sat: 9:00 AM-1:00 PM;  09:00     13:00 ##  3 1002454
112st Madison Avenue    Tue:8:00 am - 5:00 pm~ 08:00     17:00 ##  4 1005586 12th &
Brandywine Urba~ Fri: 8:00 AM-11:00 AM; 08:00     11:00 ##  5 1008071 14&U Farmers'
Market    Sat: 9:00 AM-1:00 PM;  09:00     13:00 ##  6 1012710 14th & Kennedy Street ~
Sat: 9:00 AM-1:00 PM;  09:00     13:00 ##  7 1019157 16th Ave Farmers Market Sun: 9:00
AM-1:30 PM;  09:00     13:30 ##  8 1009994 18th Street Farmer's M~ Sat: 8:00 AM-11:00
AM; 08:00     11:00 ##  9 1018365 18th Street Farmers Ma~ Sat: 8:00 AM-12:00 PM; 08:00
12:00 ## 10 1012790 19/27 Community Farmer~ Sat: 8:30 AM-1:00 PM;  08:30     13:00 ##
# ... with 3,838 more rows
```

Details of Markets which are open for less than 4 hours a day are as follows: Performing arithmetic operation on the 2 columns of data type: time

```
r farmermkt %>% transmute(FMID, MarketName, Season1Time, StartTime, EndTime) %>%
filter((EndTime - StartTime) < 14400)
## # A tibble: 1,797 x 5 ##       FMID MarketName           Season1Time
StartTime EndTime ##      <dbl> <chr>                <chr>                     <time>
<time> ##  1 1010691 10th Steet Communit~ Wed: 3:00 PM-6:00 PM;Sat~ 15:00     18:00 ##
2 1011100 12 South Farmers Ma~ Tue: 3:30 PM-6:30 PM;     15:30     18:30 ##  3 1005586
12th & Brandywine U~ Fri: 8:00 AM-11:00 AM;    08:00     11:00 ##  4 1016784 17th
Street Farmers~ Thu: 3:00 PM-6:30 PM;     15:00     18:30 ##  5 1009994 18th Street
Farmer'~ Sat: 8:00 AM-11:00 AM;    08:00     11:00 ##  6 1010994 2nd Street Farmers'~
Thu: 3:30 PM-6:30 PM;     15:30     18:30 ##  7 1018389 38th & Meridian Far~ Thu: 4:00
PM-6:30 PM;     16:00     18:30 ##  8 1008391 84 west farmers mar~ Thu: 3:00 PM-6:00
PM;     15:00     18:00 ##  9 1001961 8th Avenue City Far~ Tue:4:00 PM - 6:00 PM;Sa~
16:00     18:00 ## 10 1011989 9th and Grand Farme~ Tue: 4:00 PM-7:00 PM;     16:00
19:00 ## # ... with 1,787 more rows
```

Q4) First, I have created two different columns: Season1Begin and Season1Date which will state the Beginning and End Date of the Market.

This is done by using regex to only pick the first part of the dates before "to" and then use the parse function to convert all the rows in a standard format I have updated all the years to 2017 since there were many rows with only months mentioned. This will avoid confusion in our analysis.

```
Season1Begin <- gsub("(to).*","",farmermkt$Season1Date)
Season1Begin <- parse_date_time(Season1Begin, orders = c("m/d/Y", "m"))
```

```
## Warning: 6 failed to parse.
```

```
Season1Begin <- update(Season1Begin, year = 2017)
```

Similary for Season1End, I have used regex to only pick the last part of the dates after "to" and then use the parse function to convert all the rows in a standard format

```
Season1End <- gsub(".*(to)","",farmermkt$Season1Date)
Season1End <- parse_date_time(Season1End, orders = c("m/d/Y", "m"))
```

```
## Warning: 362 failed to parse.
```

```
Season1End <- update(Season1End, year = 2017)
```

Mutating to combine all required columns and using if_else to satisfy the condition in order to get a different column for Fall, Spring, Summer, Winter, HalfYear and Full Year. Each of the seasons will represent if the market is open during that season. Halfyear indicates if the market is open for 6-11 months and FullYear indicates if the market is open for 12 months along the yearly cycle.

Here assumption is made that Fall is from August to October; Summer is from May to July; Winter is from November to January and Spring is from February to April

```
Season <- farmermkt %>%
          transmute(Season1Date, Season1Begin, Season1End, BM = month(Season1Begin), EM = month(Season
                 Fall = if_else((BM == 8 | BM == 9 | BM == 10 | EM == 8 | EM == 9 | EM == 10 | (BM ==
                 Summer = if_else((BM ==  5| BM == 6 | BM == 7 | EM == 5 | EM == 6 | EM == 7 |(BM == 1
                 Winter = if_else((BM == 11 | BM == 12 | BM == 1 | EM == 11 | EM == 12 | EM == 1 | (BM
                 Spring = if_else((BM == 2 | BM == 3 | BM == 4 | EM == 2 | EM == 3 | EM == 4 | (BM ==
                 HalfYear = if_else(( (EM-BM) >= 5 & (EM-BM) < 12), "Half Year", "Less than Half Year"
                 FullYear = if_else(( (EM-BM) == 11 | (EM-BM) == 0), "Full Year", "Not Full Year"))
head(Season, 75, 11)
```

```
## # A tibble: 75 x 11
##    Season1Date Season1Begin        Season1End             BM    EM Fall
##    <chr>       <dttm>              <dttm>              <dbl> <dbl> <chr>
##  1 06/14/2017~ 2017-06-14 00:00:00 2017-08-30 00:00:00     6     8 Fall
##  2 06/24/2017~ 2017-06-24 00:00:00 2017-09-30 00:00:00     6     9 Fall
##  3 <NA>        NA                  NA                     NA    NA <NA>
##  4 04/02/2014~ 2017-04-02 00:00:00 2017-11-30 00:00:00     4    11 Not ~
##  5 July to No~ 2017-07-01 00:00:00 2017-11-01 00:00:00     7    11 Not ~
##  6 05/05/2015~ 2017-05-05 00:00:00 2017-10-27 00:00:00     5    10 Fall
##  7 06/10/2014~ 2017-06-10 00:00:00 2017-11-25 00:00:00     6    11 Not ~
##  8 05/16/2014~ 2017-05-16 00:00:00 2017-10-17 00:00:00     5    10 Fall
##  9 05/03/2014~ 2017-05-03 00:00:00 2017-11-22 00:00:00     5    11 Not ~
## 10 04/09/2016~ 2017-04-09 00:00:00 2017-11-19 00:00:00     4    11 Not ~
## # ... with 65 more rows, and 5 more variables: Summer <chr>, Winter <chr>,
## #   Spring <chr>, HalfYear <chr>, FullYear <chr>
```

Once this is run, we can see that if the market is open for any two/three seasons, then it also shows that it is "HalfYear". If the market is open for any one season, it shows that it is "Less Than Half Year" and "Not Full Year"

Q5) Extracting only the columns we need for this analysis/visualization

```
Extract <- farmermkt %>%
           transmute(city, County, State, Fall = Season$Fall, Summer = Season$Summer,
                     Spring = Season$Spring, Winter = Season$Winter, Half_Year = Season$HalfYear, Full
head(Extract,75)
```

```
## # A tibble: 75 x 9
##    city   County   State   Fall   Summer Spring Winter Half_Year  Full_Year
##    <chr>  <chr>    <chr>   <chr>  <chr>  <chr>  <chr>  <chr>      <chr>
```

```
##  1 Danvi~ Caledon~ Vermont Fall   Summer Not S~ Not W~ Less than~ Not Full~
##  2 Parma  Cuyahoga Ohio    Fall   Summer Not S~ Not W~ Less than~ Not Full~
##  3 Six M~ <NA>     South ~ <NA>   <NA>   <NA>   <NA>   <NA>       <NA>
##  4 Lamar  Barton   Missou~ Not F~ Not S~ Spring Winter Half Year  Not Full~
##  5 New Y~ New York New Yo~ Not F~ Summer Not S~ Winter Less than~ Not Full~
##  6 Nashv~ Davidson Tennes~ Fall   Summer Not S~ Not W~ Half Year  Not Full~
##  7 New Y~ New York New Yo~ Not F~ Summer Not S~ Winter Half Year  Not Full~
##  8 Wilmi~ New Cas~ Delawa~ Fall   Summer Not S~ Not W~ Half Year  Not Full~
##  9 Washi~ Distric~ Distri~ Not F~ Summer Not S~ Winter Half Year  Not Full~
## 10 Washi~ Distric~ Distri~ Not F~ Not S~ Spring Winter Half Year  Not Full~
## # ... with 65 more rows
```

In this exercise I have tried to analyze which season shows more number of Open Market Days. Visualization is as follows:

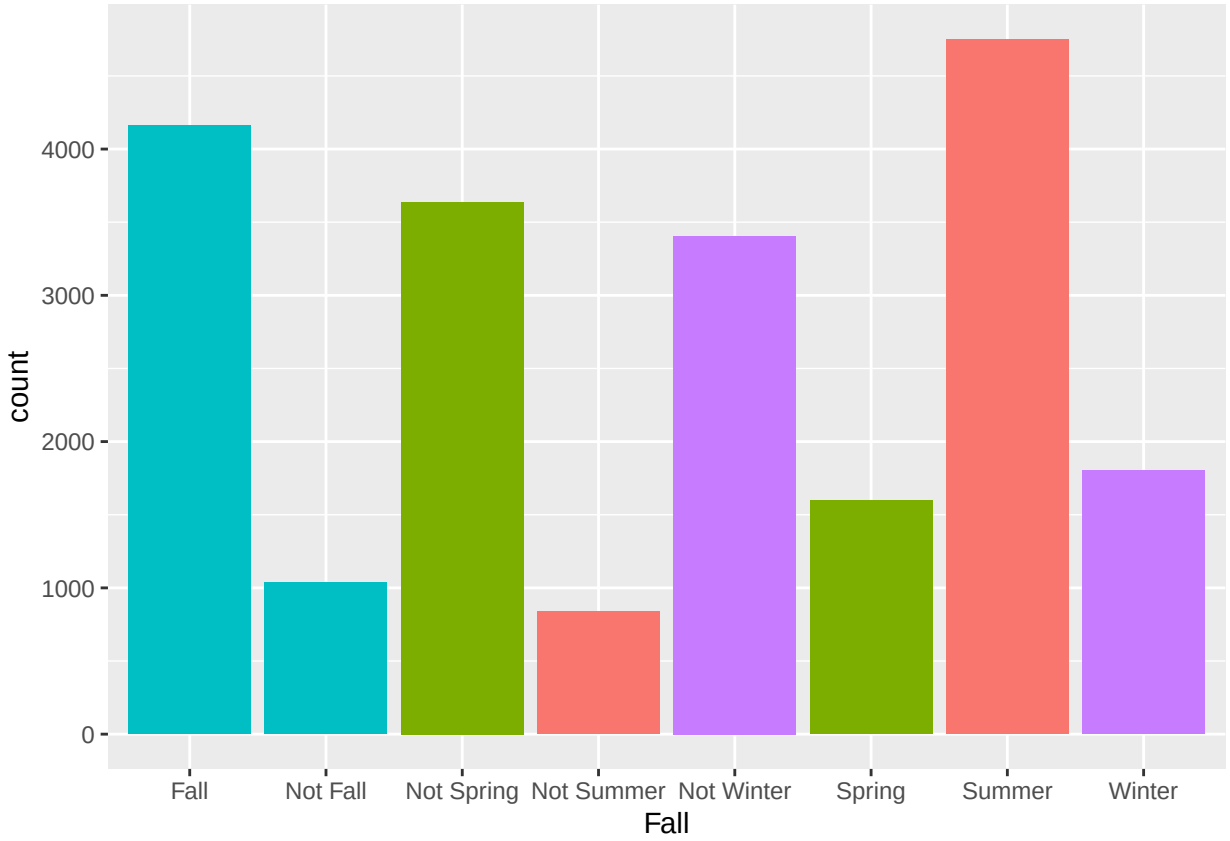
```
ggplot(Extract) + geom_bar(mapping = aes(x = Fall, fill = "Red")) +
  geom_bar(mapping = aes(x = Spring, fill = "Green")) +
  geom_bar(mapping = aes(x = Summer, fill = "Cyan")) +
  geom_bar(mapping = aes(x = Winter, fill = "Yellow")) + theme(legend.position = "none")
```

```
## Warning: Removed 3587 rows containing non-finite values (stat_count).

## Warning: Removed 3553 rows containing non-finite values (stat_count).

## Warning: Removed 3201 rows containing non-finite values (stat_count).

## Warning: Removed 3580 rows containing non-finite values (stat_count).
```

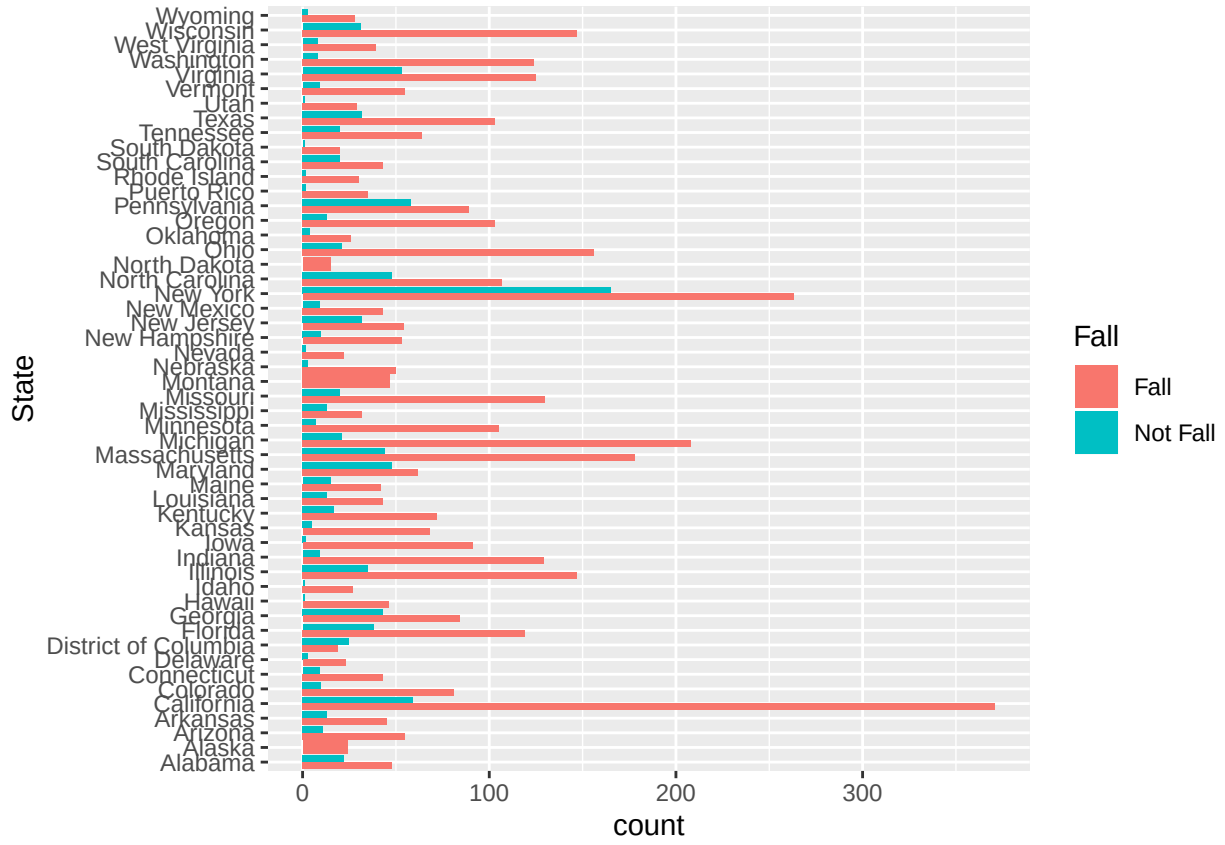


This shows us that there are more number of open markets during Fall and Summer season

To dig deeper inside, I have analyzed which States show much number of Open Markets Fall and Summer respectively:

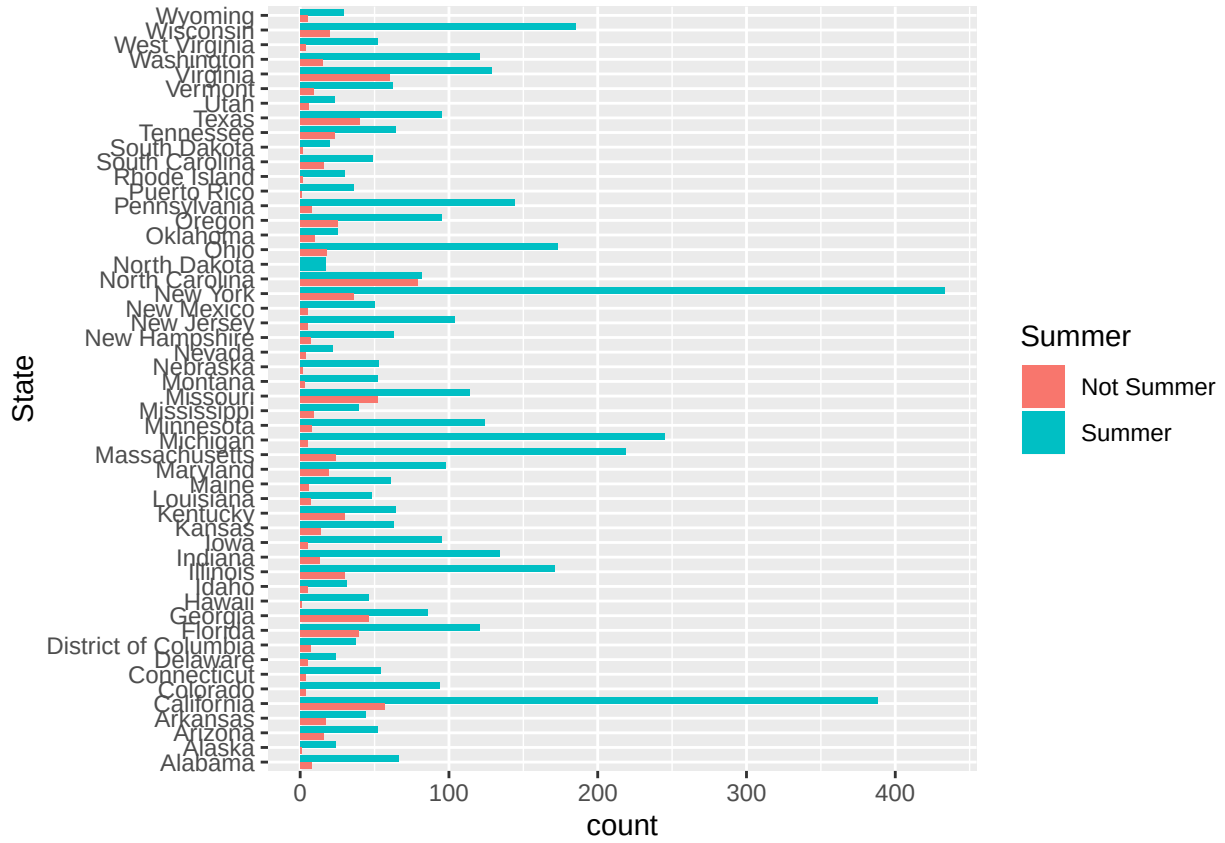
For Fall Season:

```
Extract %>%
  filter(!is.na(Fall)) %>%
  ggplot() + geom_bar(aes(State, fill = Fall), position = "dodge") + coord_flip()
```



For Summer Season:

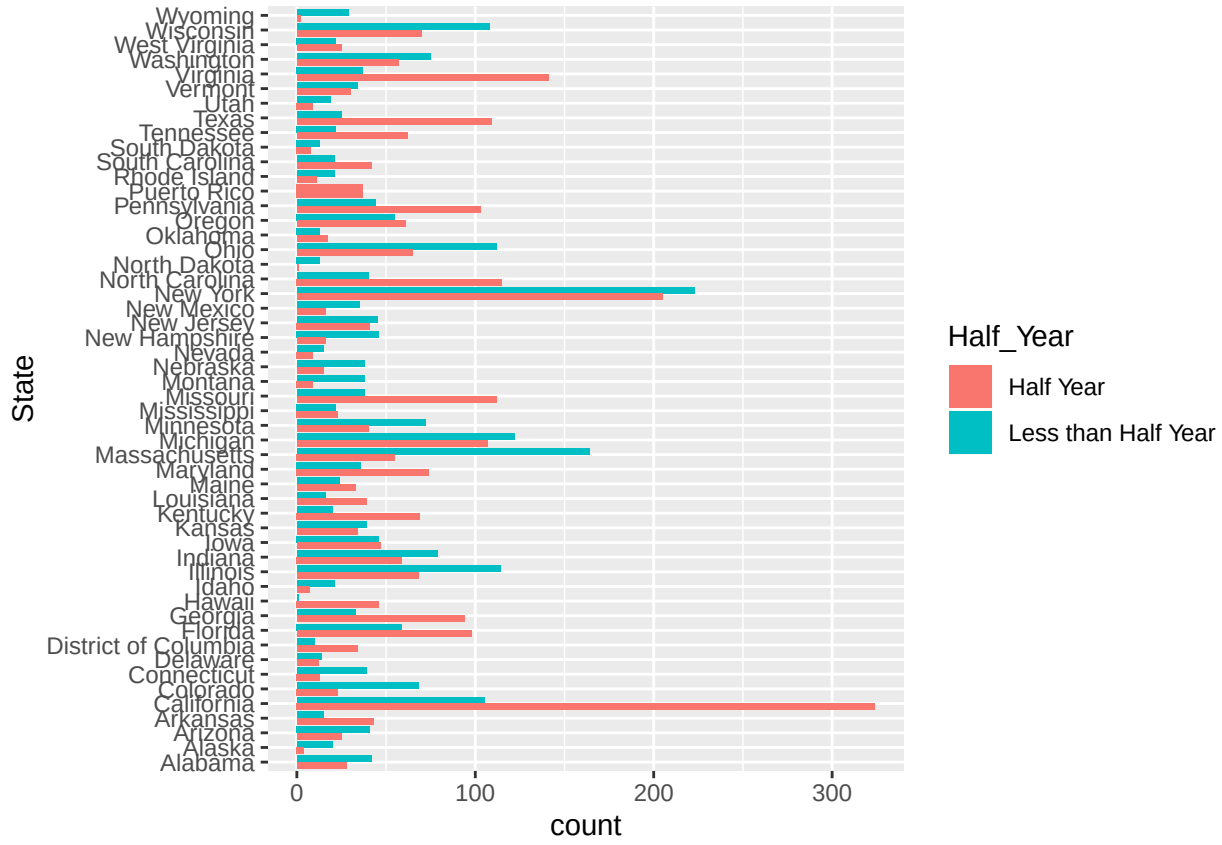
```
Extract %>%
  filter(!is.na(Summer)) %>%
  ggplot() + geom_bar(aes(State, fill = Summer), position = "dodge") + coord_flip()
```



From the above two graphs, we can see all States in the US have more number of markets opened during the Fall and Summer which is an interesting find. California followed by New York shows the top two number of open market states in Fall and Summer. California has more number of open markets in the Fall whereas New York has more number of open markets in the Summer. This makes sense since weather plays a big difference to decide if the market should be opened.

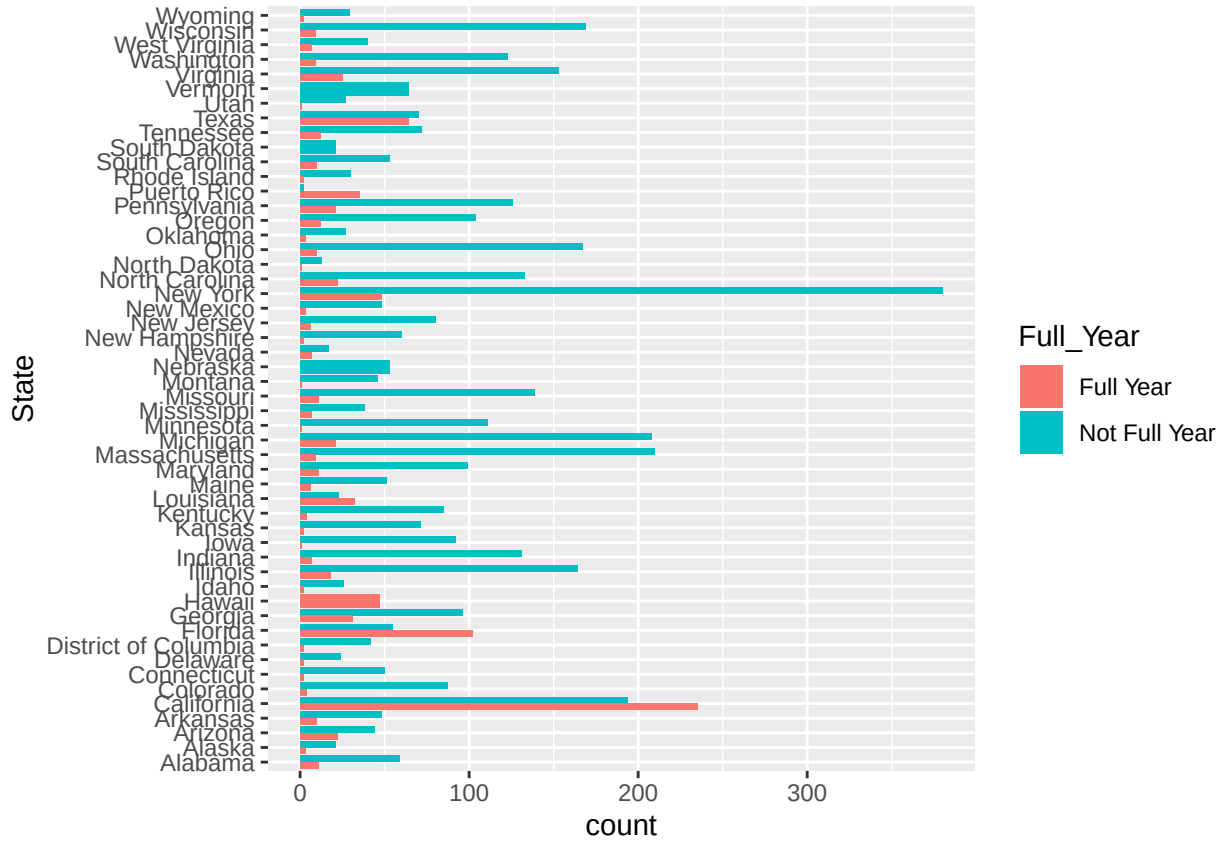
Additionally a bar graph of proportion of how many markets are open for Half Year in each State is done:

```
Extract %>%
  filter(!is.na(Half_Year)) %>%
  ggplot() + geom_bar(aes(State, fill = Half_Year), position = "dodge") + coord_flip()
```



Also, proportion of how many markets are open for Full year in each State is as follows:

```
Extract %>%
  filter(!is.na(Full_Year)) %>%
  ggplot() + geom_bar(aes(State, fill = Full_Year), position = "dodge") + coord_flip()
```



Conclusion: The above graphs show that there are more number of markets which are opened for Half Year rather that Full Year. Almost all states have more number of open markets for Half Year. This shows us that the States of have almost % of markets open year round and the States of have almost % of markets open half yearly.Only California is the state where there are highest number of markets opened for Full Year.