DA5020 - Homework 5: Dates and Times

2017-10-07

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 (http://uc-r.github.io/control_statements) since they will come handy in solving some of the problems.

Questions

```
#setwd("C:/Users/Zhixiong Cheng/Desktop/DA5020 17561 CollectStoreRetrieve Data SEC 01 Fall 2017
 Semester Graduate [BOS-2-TR]/Week 4-Data wrangling data import and strings")
farmers_market <- read.csv("farmers_market.csv", header = TRUE, stringsAsFactors = FALSE, na.str
ings = "")
#str(farmers_market)
library(lubridate)
```

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)
library(stringr)
```

```
## Warning: package 'stringr' was built under R version 3.4.2
```

```
# first store the Season1Days into variable a:
a<-farmers_market$Season1Time %>%
  strsplit(split="\\;") %>% # after observation, the time is seperated by symbol ";"
  lapply(function(x) length(x[!is.na(x)])) %>%
  unlist()
# Then add it as the new column of farmers_market table
options(tibble.width = Inf)
options(tibble.print_max = 25, tibble.print_min = 20)
Q1 <- farmers_market %>%
  mutate(Season1Days =a) %>%
  as_tibble() %>%
  select(c("FMID", "Season1Time", "Season1Days"))
Q1
```

```
## # A tibble: 8,707 x 3
##       FMID                                                                              Seaso
n1Time Season1Days
##      <int>
<chr>       <int>
##  1 1018261                                                                    Wed: 9:00 AM-1:
00 PM;           1
##  2 1018318                                                                    Sat: 9:00 AM-1:
00 PM;           1
##  3 1009364
  <NA>           0
##  4 1010691                                               Wed: 3:00 PM-6:00 PM;Sat: 8:00 AM-1:
00 PM;           2
##  5 1002454                                             Tue:8:00 am - 5:00 pm;Sat:8:00 am - 8:
00 pm;           2
##  6 1011100                                                                    Tue: 3:30 PM-6:
30 PM;           1
##  7 1009845                                                                   Tue: 10:00 AM-7:
00 PM;           1
##  8 1005586                                                                   Fri: 8:00 AM-11:
00 AM;           1
##  9 1008071                                                                    Sat: 9:00 AM-1:
00 PM;           1
## 10 1012710                                                                    Sat: 9:00 AM-1:
00 PM;           1
## 11 1018792                                                                    Wed: 2:30 PM-6:
30 PM;           1
## 12 1016782                                                                    Tue: 8:00 AM-6:
00 PM;           1
## 13 1003877                                                                   Wed:3:00 pm - 7:
00 pm;           1
## 14 1016784                                                                    Thu: 3:00 PM-6:
30 PM;           1
## 15 1010968                                                                    Thu: 3:00 PM-7:
00 PM;           1
## 16 1009994                                                                   Sat: 8:00 AM-11:
00 AM;           1
## 17 1018365                                                                   Sat: 8:00 AM-12:
00 PM;           1
## 18 1012790                                                                    Sat: 8:30 AM-1:
00 PM;           1
## 19 1012158 Wed: 11:00 AM-6:00 PM;Thu: 11:00 AM-6:00 PM;Fri: 11:00 AM-6:00 PM;Sat: 11:00 AM-6:
00 PM;           4
## 20 1010873                                                                   Fri: 7:30 AM-12:
00 PM;           1
## # ... with 8,687 more rows
```

```
# mutate(sep = do.call(rbind, lapply(a, function(x) length(x[!is.na(x)]))))
```

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

```
# my rule is that if a market opens during weekends in `Season1`, then `WeekendOpen` = TRUE; or
 `WeekendOpen` = FALSE
a<-farmers_market$Season1Time %>%
  strsplit(split="\\;") # after observation, the time is seperated by symbol ";"
index <- grep("Sun|Sat", a, ignore.case = TRUE) # get the index of opens during weekends

# print out the result:
options(tibble.width = Inf)
options(tibble.print_max = 25, tibble.print_min = 20)
Q2 <- farmers_market %>%
  mutate( WeekendOpen = ifelse(c(1:length(farmers_market$Season1Time)) %in% index, TRUE, FALSE))
 %>%
  as_tibble() %>%
  select(c("FMID", "Season1Time","WeekendOpen"))
Q2
```

```
## # A tibble: 8,707 x 3
##       FMID                                                                              Seaso
n1Time WeekendOpen
##      <int>
<chr>       <lgl>
##  1 1018261                                                                    Wed: 9:00 AM-1:
00 PM;       FALSE
##  2 1018318                                                                    Sat: 9:00 AM-1:
00 PM;        TRUE
##  3 1009364
  <NA>       FALSE
##  4 1010691                                               Wed: 3:00 PM-6:00 PM;Sat: 8:00 AM-1:
00 PM;        TRUE
##  5 1002454                                             Tue:8:00 am - 5:00 pm;Sat:8:00 am - 8:
00 pm;        TRUE
##  6 1011100                                                                    Tue: 3:30 PM-6:
30 PM;       FALSE
##  7 1009845                                                                   Tue: 10:00 AM-7:
00 PM;       FALSE
##  8 1005586                                                                   Fri: 8:00 AM-11:
00 AM;       FALSE
##  9 1008071                                                                    Sat: 9:00 AM-1:
00 PM;        TRUE
## 10 1012710                                                                    Sat: 9:00 AM-1:
00 PM;        TRUE
## 11 1018792                                                                    Wed: 2:30 PM-6:
30 PM;       FALSE
## 12 1016782                                                                    Tue: 8:00 AM-6:
00 PM;       FALSE
## 13 1003877                                                                   Wed:3:00 pm - 7:
00 pm;       FALSE
## 14 1016784                                                                    Thu: 3:00 PM-6:
30 PM;       FALSE
## 15 1010968                                                                    Thu: 3:00 PM-7:
00 PM;       FALSE
## 16 1009994                                                                   Sat: 8:00 AM-11:
00 AM;        TRUE
## 17 1018365                                                                   Sat: 8:00 AM-12:
00 PM;        TRUE
## 18 1012790                                                                    Sat: 8:30 AM-1:
00 PM;        TRUE
## 19 1012158 Wed: 11:00 AM-6:00 PM;Thu: 11:00 AM-6:00 PM;Fri: 11:00 AM-6:00 PM;Sat: 11:00 AM-6:
00 PM;        TRUE
## 20 1010873                                                                   Fri: 7:30 AM-12:
00 PM;       FALSE
## # ... with 8,687 more rows
```

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.

```r
options(warn = -1)
library(dplyr)
# first is to get the table of markets that is closed before 6PM:
## my rule for multiple close days is that if one market close before 6pm on Sunday, for example, and don't close before 6pm on Monday, we will treat it as the market that don't close before 6pm.
b<-farmers_market$Season1Time %>% # b is the date with string format, mode is list:
  str_extract_all( pattern = "\\-\\s?\\d\\d?\\:\\d{2}\\s?[a|A|p|P][m|M]") %>%
  str_extract_all( pattern = "\\d\\d?\\:\\d{2}\\s?[a|A|p|P][m|M]")
# Find the list of market that close before 6pm:
temp_list<-as.logical( c(1:length(farmers_market$Season1Time)))
for (ind in c(1:length(farmers_market$Season1Time))){
  temp_list[ind] <- (min(parse_time(unlist(b[ind]))) < parse_time("6:00 pm"))
}
# get the index that market close before 6pm: which(temp_list == TRUE)
# print out the result:
options(tibble.width = Inf)
options(tibble.print_max = 25, tibble.print_min = 20)
Q4_1 <- farmers_market %>%
  as_tibble() %>%
  mutate(FMID,
         Season1Time,
    close_before_six = temp_list) %>%
  select(c("FMID", "Season1Time", "close_before_six"))
Q4_1
```

```
## # A tibble: 8,707 x 3
##       FMID                                                                              Seaso
n1Time close_before_six
##      <int>
<chr>            <lgl>
##  1 1018261                                                                    Wed: 9:00 AM-1:
00 PM;             TRUE
##  2 1018318                                                                    Sat: 9:00 AM-1:
00 PM;             TRUE
##  3 1009364
  <NA>               NA
##  4 1010691                                               Wed: 3:00 PM-6:00 PM;Sat: 8:00 AM-1:
00 PM;             TRUE
##  5 1002454                                             Tue:8:00 am - 5:00 pm;Sat:8:00 am - 8:
00 pm;             TRUE
##  6 1011100                                                                    Tue: 3:30 PM-6:
30 PM;            FALSE
##  7 1009845                                                                   Tue: 10:00 AM-7:
00 PM;            FALSE
##  8 1005586                                                                   Fri: 8:00 AM-11:
00 AM;             TRUE
##  9 1008071                                                                    Sat: 9:00 AM-1:
00 PM;             TRUE
## 10 1012710                                                                    Sat: 9:00 AM-1:
00 PM;             TRUE
## 11 1018792                                                                    Wed: 2:30 PM-6:
30 PM;            FALSE
## 12 1016782                                                                    Tue: 8:00 AM-6:
00 PM;            FALSE
## 13 1003877                                                                   Wed:3:00 pm - 7:
00 pm;            FALSE
## 14 1016784                                                                    Thu: 3:00 PM-6:
30 PM;            FALSE
## 15 1010968                                                                    Thu: 3:00 PM-7:
00 PM;            FALSE
## 16 1009994                                                                   Sat: 8:00 AM-11:
00 AM;             TRUE
## 17 1018365                                                                   Sat: 8:00 AM-12:
00 PM;             TRUE
## 18 1012790                                                                    Sat: 8:30 AM-1:
00 PM;             TRUE
## 19 1012158 Wed: 11:00 AM-6:00 PM;Thu: 11:00 AM-6:00 PM;Fri: 11:00 AM-6:00 PM;Sat: 11:00 AM-6:
00 PM;            FALSE
## 20 1010873                                                                   Fri: 7:30 AM-12:
00 PM;             TRUE
## # ... with 8,687 more rows
```

```
# Second is to get the table of markets that open only for fewer than 4 hours a day
options(warn = -1)
library(dplyr)
b<-farmers_market$Season1Time %>% # b is the date with string format, mode is list:
  str_extract_all( pattern = "\\d\\d?\\:\\d{2}\\s?[a|A|p|P][m|M]")
# get the time interval, and compare it with 4 hours:
tp_avg <- vector(mode="logical", length=length(farmers_market$Season1Time))
for (ind in c(1:length(farmers_market$Season1Time))){
  i <- length(b[[ind]])/2
  # initialize a temporary vector:
  tp_list <- vector(mode="numeric", length=i)
  if(i >= 1){
  for (ii in seq(from = 1, to = length(b[[ind]]), by = 2)){
  tp_list[(ii+1)/2] <- parse_time(b[[ind]][ii+1]) - parse_time(b[[ind]][ii])
  }
  }else tp_list = NA
  # get avarage time difference:
  tp_avg[ind] <- mean(tp_list) < as.integer(dhours(4)) # tp_avg would be TRUE if average time di
ff is less than 4 hours; FALSE otherwise.

}
# get the index that market open only for fewer than 4 hours a day: which(tp_avg == TRUE)
# print out the result:

options(tibble.width = Inf)
options(tibble.print_max = 25, tibble.print_min = 20)
Q4_2 <- farmers_market %>%
  as_tibble() %>%
  mutate(FMID,
         Season1Time,
    open_fewer_four = tp_avg) %>%
  select(c("FMID", "Season1Time", "open_fewer_four"))
Q4_2
```

```
## # A tibble: 8,707 x 3
##       FMID                                                                              Seaso
n1Time open_fewer_four
##      <int>
<chr>           <lgl>
##  1 1018261                                                                    Wed: 9:00 AM-1:
00 PM;           FALSE
##  2 1018318                                                                    Sat: 9:00 AM-1:
00 PM;           FALSE
##  3 1009364
  <NA>              NA
##  4 1010691                                               Wed: 3:00 PM-6:00 PM;Sat: 8:00 AM-1:
00 PM;           FALSE
##  5 1002454                                             Tue:8:00 am - 5:00 pm;Sat:8:00 am - 8:
00 pm;           FALSE
##  6 1011100                                                                    Tue: 3:30 PM-6:
30 PM;            TRUE
##  7 1009845                                                                   Tue: 10:00 AM-7:
00 PM;           FALSE
##  8 1005586                                                                   Fri: 8:00 AM-11:
00 AM;            TRUE
##  9 1008071                                                                    Sat: 9:00 AM-1:
00 PM;           FALSE
## 10 1012710                                                                    Sat: 9:00 AM-1:
00 PM;           FALSE
## 11 1018792                                                                    Wed: 2:30 PM-6:
30 PM;           FALSE
## 12 1016782                                                                    Tue: 8:00 AM-6:
00 PM;           FALSE
## 13 1003877                                                                   Wed:3:00 pm - 7:
00 pm;           FALSE
## 14 1016784                                                                    Thu: 3:00 PM-6:
30 PM;            TRUE
## 15 1010968                                                                    Thu: 3:00 PM-7:
00 PM;           FALSE
## 16 1009994                                                                   Sat: 8:00 AM-11:
00 AM;            TRUE
## 17 1018365                                                                   Sat: 8:00 AM-12:
00 PM;           FALSE
## 18 1012790                                                                    Sat: 8:30 AM-1:
00 PM;           FALSE
## 19 1012158 Wed: 11:00 AM-6:00 PM;Thu: 11:00 AM-6:00 PM;Fri: 11:00 AM-6:00 PM;Sat: 11:00 AM-6:
00 PM;           FALSE
## 20 1010873                                                                   Fri: 7:30 AM-12:
00 PM;           FALSE
## # ... with 8,687 more rows
```

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.)

```
# First I check the exact date for 4 seasons in 2017 online:
# Spring: March 20th ~ June 19th
# Summer: June 20th ~ September 21st
# Fall: September 22nd ~ December 20th
# Winter:   December 21st ~ December 31st AND January 1st ~ March 19th
options(warn = -1)

# initialized the four columns:
Spring_col <- rep(NA, length=length(farmers_market$Season1Time))
Summer_col <- rep(NA, length=length(farmers_market$Season1Time))
Fall_col <- rep(NA, length=length(farmers_market$Season1Time))
Winter_col <- rep(NA, length=length(farmers_market$Season1Time))


a<- farmers_market$Season1Date %>%
  strsplit(split="\\s+to\\s+")

# add two auxiliary columns:
Season1BeginDate<-rep(NA, length=length(farmers_market$Season1Time))
Season1EndDate<-rep(NA, length=length(farmers_market$Season1Time))
for(n1 in c(1:length(farmers_market$Season1Time))){
Season1BeginDate[n1] <- a[[n1]][1]
Season1EndDate[n1] <- a[[n1]][2]
}

# get index of full date and only contain month
ind_no_number_begin <- grep(pattern = "^[a-zA-Z]+$", Season1BeginDate) # get index of only conta
in month
ind_no_number_end <- grep(pattern = "^[a-zA-Z]+$", Season1EndDate) # get index of only contain m
onth


# add days and year to the date that only contain month:
Season1BeginDate[ind_no_number_begin] <- paste(Season1BeginDate[ind_no_number_begin], "1, 2017")
Season1EndDate[ind_no_number_end] <- paste(Season1EndDate[ind_no_number_end], "1, 2017")


# get intervals for 4 seasons:
    in_sp <- as.interval( mdy("03/20/2017",tz = "UTC"), mdy("06/19/2017",tz = "UTC"))
    in_sm <- as.interval(mdy("06/20/2017",tz = "UTC"), mdy("09/21/2017",tz = "UTC"))
    in_f <- as.interval(mdy("09/22/2017",tz = "UTC"), mdy("10/20/2017",tz = "UTC"))
    in_w1 <- as.interval(mdy("12/21/2017",tz = "UTC"), mdy("12/31/2017",tz = "UTC"))
    in_w2 <- as.interval(mdy("01/01/2017",tz = "UTC"), mdy("03/19/2017",tz = "UTC"))


# for loop to get 4 seasons' columns:
for(ind in c(1:length(farmers_market$Season1Date))){

    if(!is.na(Season1BeginDate[ind])){
    bg <- make_datetime(2017, month(mdy(Season1BeginDate[ind])), day(mdy(Season1BeginDate[ind]
)))
    } else bg<-as.Date(NA) # bg is the begining date
    if(!is.na(Season1EndDate[ind])){
```

```
    ed <- make_datetime(2017, month(mdy(Season1EndDate[ind])), day(mdy(Season1EndDate[ind] )))
    } else ed<-as.Date(NA) # ed is the closing date

    # need to get the across seasons situations:
    over_3_2 <- ((bg %within% in_f) & (ed %within% in_sm)) # it means market start from fall(i.
e.3), and close in summer next year(i.e.2)
    over_4_2 <- ((((bg %within% in_w1)|(bg %within% in_w2))) & (ed %within% in_sm))
    over_4_3 <- ((((bg %within% in_w1)|(bg %within% in_w2))) & (ed %within% in_f))
    over_1_3 <- ((bg %within% in_sp) & (ed %within% in_f))
    over_1_4 <- ((bg %within% in_sp) & (((ed %within% in_w1)|(ed %within% in_w2))))
    over_2_4 <- ((bg %within% in_sm) & (((ed %within% in_w1)|(ed %within% in_w2))))
    over_2_1 <- ((bg %within% in_sm) & (ed %within% in_sp))
    over_3_1 <- ((bg %within% in_f) & (ed %within% in_sp))

      if(!is.na((bg %within% in_sp)|(ed %within% in_sp)| over_3_2 |over_4_2|over_4_3)){
      Spring_col[ind] <- (bg %within% in_sp)|(ed %within% in_sp)| over_3_2 |over_4_2|over_4_3
      }#Spring_col could be TRUE, FALSE, NA(i.e.unknown)

      if(!is.na((bg %within% in_sm)|(ed %within% in_sm)|over_4_3|over_1_3|over_1_4)){
      Summer_col[ind] <- (bg %within% in_sm)|(ed %within% in_sm)|over_4_3|over_1_3|over_1_4
      }

      if(!is.na((bg %within% in_f)|(ed %within% in_f)|over_1_4|over_2_4|over_2_1)){
      Fall_col[ind] <- (bg %within% in_f)|(ed %within% in_f)|over_1_4|over_2_4|over_2_1
      }

      if(!is.na(((bg %within% in_w1)|(bg %within% in_w2))|((ed %within% in_w1)|(ed %within% in_w
2))|over_2_1|over_3_1|over_3_2)){
      Winter_col[ind] <- ((bg %within% in_w1)|(bg %within% in_w2))|((ed %within% in_w1)|(ed %wit
hin% in_w2))|over_2_1|over_3_1|over_3_2}


}

# year round means the market open for all seasons per year
# convert logical elements to numbers, TRUE = 1, FALSE = 0, NA = 0:
Spring_col_num <- as.numeric(Spring_col)
Spring_col_num[is.na(Spring_col_num[1:8707])] <- 0

Summer_col_num <- as.numeric(Summer_col)
Summer_col_num[is.na(Summer_col_num[1:8707])] <- 0

Fall_col_num <- as.numeric(Fall_col)
Fall_col_num[is.na(Fall_col_num[1:8707])] <- 0

Winter_col_num <- as.numeric(Winter_col)
Winter_col_num[is.na(Winter_col_num[1:8707])] <- 0

year_round <- Spring_col_num + Summer_col_num + Fall_col_num + Winter_col_num
year_round[year_round!=4] <- 0
year_round[year_round==4] <- 1
year_round<- as.logical(year_round)

# half year means the market open for exact 2 seasons per year, no matter what the sequence is:
 e.g. only open at summer and winter:
half_year <- Spring_col_num + Summer_col_num + Fall_col_num + Winter_col_num
half_year[half_year!=2] <- 0
half_year[half_year==2] <- 1
half_year<- as.logical(half_year)

options(tibble.width = Inf)
options(tibble.print_max = 30, tibble.print_min = 20)
Q5 <- farmers_market$Season1Date %>%
  as_tibble() %>%
  mutate(
         Season1BeginDate = Season1BeginDate,
         Season1EndDate = Season1EndDate,
         Spring_col = Spring_col,
         Summer_col = Summer_col,
         Fall_col = Fall_col,
         Winter_col = Winter_col,
         year_round = year_round,
         half_year = half_year
         ) %>%
  select(c( "Season1BeginDate", "Season1EndDate", "Spring_col", "Summer_col", "Fall_col", "Winter_col", "year_round", "half_year"))

Q5
```

```
## # A tibble: 8,707 x 8
##    Season1BeginDate    Season1EndDate Spring_col Summer_col Fall_col Winter_col year_round ha
lf_year
##               <chr>             <chr>      <lgl>      <lgl>    <lgl>      <lgl>      <lgl>
  <lgl>
##  1       06/14/2017        08/30/2017       TRUE       TRUE    FALSE      FALSE      FALSE
   TRUE
##  2       06/24/2017        09/30/2017      FALSE       TRUE     TRUE      FALSE      FALSE
   TRUE
##  3             <NA>              <NA>         NA         NA       NA         NA      FALSE
  FALSE
##  4       04/02/2014        11/30/2014       TRUE      FALSE    FALSE      FALSE      FALSE
  FALSE
##  5     July 1, 2017  November 1, 2017      FALSE       TRUE    FALSE      FALSE      FALSE
  FALSE
##  6       05/05/2015        10/27/2015       TRUE      FALSE    FALSE      FALSE      FALSE
  FALSE
##  7       06/10/2014        11/25/2014       TRUE      FALSE    FALSE      FALSE      FALSE
  FALSE
##  8       05/16/2014        10/17/2014       TRUE       TRUE     TRUE      FALSE      FALSE
  FALSE
##  9       05/03/2014        11/22/2014       TRUE      FALSE    FALSE      FALSE      FALSE
  FALSE
## 10       04/09/2016        11/19/2016       TRUE      FALSE    FALSE      FALSE      FALSE
  FALSE
## 11       07/05/2017        11/22/2017      FALSE       TRUE    FALSE      FALSE      FALSE
  FALSE
## 12       06/29/2017        11/22/2017      FALSE       TRUE    FALSE      FALSE      FALSE
  FALSE
## 13     June 1, 2017 September 1, 2017       TRUE       TRUE    FALSE      FALSE      FALSE
   TRUE
## 14       06/08/2017              <NA>       TRUE         NA       NA         NA      FALSE
  FALSE
## 15       06/01/2015        11/15/2015       TRUE      FALSE    FALSE      FALSE      FALSE
  FALSE
## 16       06/07/2014        09/27/2014       TRUE       TRUE     TRUE      FALSE      FALSE
  FALSE
## 17       05/20/2017        09/30/2017       TRUE       TRUE     TRUE      FALSE      FALSE
  FALSE
## 18       09/03/2016        06/24/2017      FALSE       TRUE    FALSE      FALSE      FALSE
  FALSE
## 19       01/01/2016        12/31/2016      FALSE      FALSE    FALSE       TRUE      FALSE
  FALSE
## 20       06/05/2015        10/30/2015       TRUE      FALSE    FALSE      FALSE      FALSE
  FALSE
## # ... with 8,687 more rows
```

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?

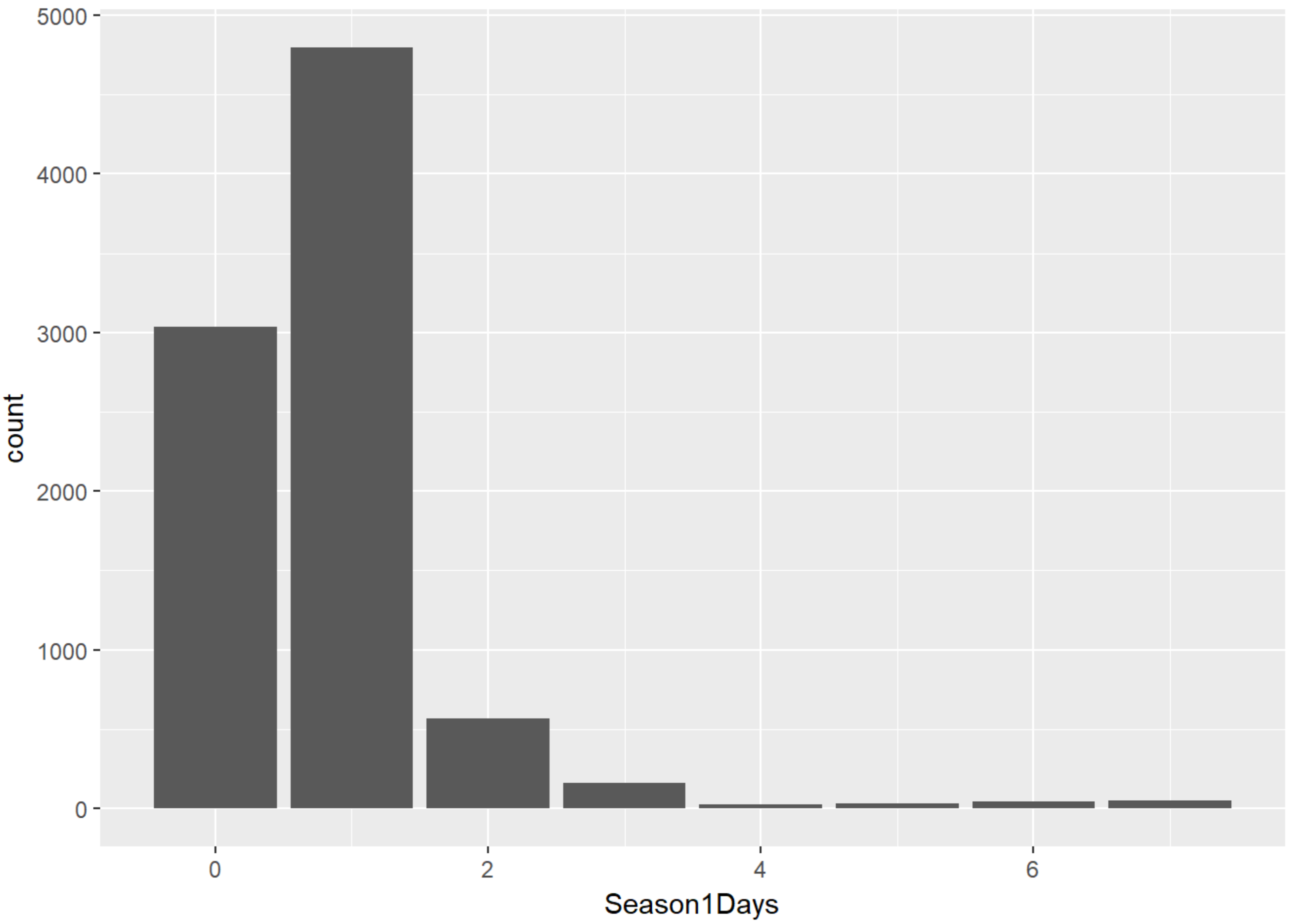
```
#`Season1Days` means the number of days a market is opened per week: Q1$Season1Days
# `WeekendOpen` indicating whether a market opens during weekends in `Season1`: Q2$WeekendOpen
# in Season1, which markets close before 6PM, and which open only for fewer than 4 hours a day:
 Q4_1$close_before_six; Q4_2$open_fewer_four
# 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`: Q5

# get a big table:
options(tibble.width = Inf)
options(tibble.print_max = 30, tibble.print_min = 20)
Q_all <- farmers_market %>%
  as_tibble() %>%
  mutate(FMID,
         city,
         State,
         Season1Date,
         Season1Time,
         Season1Days = Q1$Season1Days,
         WeekendOpen = Q2$WeekendOpen,
         close_before_six = Q4_1$close_before_six,
         open_fewer_four = Q4_2$open_fewer_four,
         Spring_col = Q5$Spring_col,
         Summer_col = Q5$Summer_col,
         Fall_col= Q5$Fall_col,
         Winter_col= Q5$Winter_col,
         year_round = Q5$year_round,
         half_year = Q5$half_year) %>%
  select(c("FMID", "city", "State","Season1Date", "Season1Time", "Season1Days","WeekendOpen","cl
ose_before_six", "open_fewer_four",  "Spring_col", "Summer_col", "Fall_col", "Winter_col", "year
_round", "half_year"))
Q_all
```

```
## # A tibble: 8,707 x 15
##       FMID         city                State              Season1Date
                                                      Season1Time Season1Days WeekendOpen close_
before_six open_fewer_four Spring_col Summer_col Fall_col Winter_col year_round half_year
##      <int>        <chr>               <chr>                    <chr>
                                                            <chr>       <int>       <lgl>
     <lgl>           <lgl>      <lgl>      <lgl>    <lgl>      <lgl>      <lgl>     <lgl>
##  1 1018261     Danville              Vermont 06/14/2017 to 08/30/2017
                                            Wed: 9:00 AM-1:00 PM;           1       FALSE
      TRUE           FALSE       TRUE       TRUE    FALSE      FALSE      FALSE      TRUE
##  2 1018318        Parma                 Ohio 06/24/2017 to 09/30/2017
                                            Sat: 9:00 AM-1:00 PM;           1        TRUE
      TRUE           FALSE      FALSE       TRUE     TRUE      FALSE      FALSE      TRUE
##  3 1009364     Six Mile       South Carolina                     <NA>
                                                                 <NA>           0       FALSE
        NA              NA         NA         NA       NA         NA      FALSE     FALSE
##  4 1010691        Lamar             Missouri 04/02/2014 to 11/30/2014
                       Wed: 3:00 PM-6:00 PM;Sat: 8:00 AM-1:00 PM;           2        TRUE
      TRUE           FALSE       TRUE      FALSE    FALSE      FALSE      FALSE     FALSE
##  5 1002454     New York             New York         July to November
                     Tue:8:00 am - 5:00 pm;Sat:8:00 am - 8:00 pm;           2        TRUE
      TRUE           FALSE      FALSE       TRUE    FALSE      FALSE      FALSE     FALSE
##  6 1011100    Nashville            Tennessee 05/05/2015 to 10/27/2015
                                            Tue: 3:30 PM-6:30 PM;           1       FALSE
     FALSE            TRUE       TRUE      FALSE    FALSE      FALSE      FALSE     FALSE
##  7 1009845     New York             New York 06/10/2014 to 11/25/2014
                                           Tue: 10:00 AM-7:00 PM;           1       FALSE
     FALSE           FALSE       TRUE      FALSE    FALSE      FALSE      FALSE     FALSE
##  8 1005586   Wilmington             Delaware 05/16/2014 to 10/17/2014
                                           Fri: 8:00 AM-11:00 AM;           1       FALSE
      TRUE            TRUE       TRUE       TRUE     TRUE      FALSE      FALSE     FALSE
##  9 1008071   Washington District of Columbia 05/03/2014 to 11/22/2014
                                            Sat: 9:00 AM-1:00 PM;           1        TRUE
      TRUE           FALSE       TRUE      FALSE    FALSE      FALSE      FALSE     FALSE
## 10 1012710  Washington  District of Columbia 04/09/2016 to 11/19/2016
                                            Sat: 9:00 AM-1:00 PM;           1        TRUE
      TRUE           FALSE       TRUE      FALSE    FALSE      FALSE      FALSE     FALSE
## 11 1018792        Bronx             New York 07/05/2017 to 11/22/2017
                                            Wed: 2:30 PM-6:30 PM;           1       FALSE
     FALSE           FALSE      FALSE       TRUE    FALSE      FALSE      FALSE     FALSE
## 12 1016782    New York              New York 06/29/2017 to 11/22/2017
                                            Tue: 8:00 AM-6:00 PM;           1       FALSE
     FALSE           FALSE      FALSE       TRUE    FALSE      FALSE      FALSE     FALSE
## 13 1003877  Minneapolis            Minnesota        June to September
                                           Wed:3:00 pm - 7:00 pm;           1       FALSE
     FALSE           FALSE       TRUE       TRUE    FALSE      FALSE      FALSE      TRUE
## 14 1016784    Richmond              Virginia           06/08/2017 to 
                                            Thu: 3:00 PM-6:30 PM;           1       FALSE
     FALSE            TRUE       TRUE         NA       NA         NA      FALSE     FALSE
## 15 1010968 Philadelphia         Pennsylvania 06/01/2015 to 11/15/2015
                                            Thu: 3:00 PM-7:00 PM;           1       FALSE
     FALSE           FALSE       TRUE      FALSE    FALSE      FALSE      FALSE     FALSE
## 16 1009994  Scottsbluff             Nebraska 06/07/2014 to 09/27/2014
```

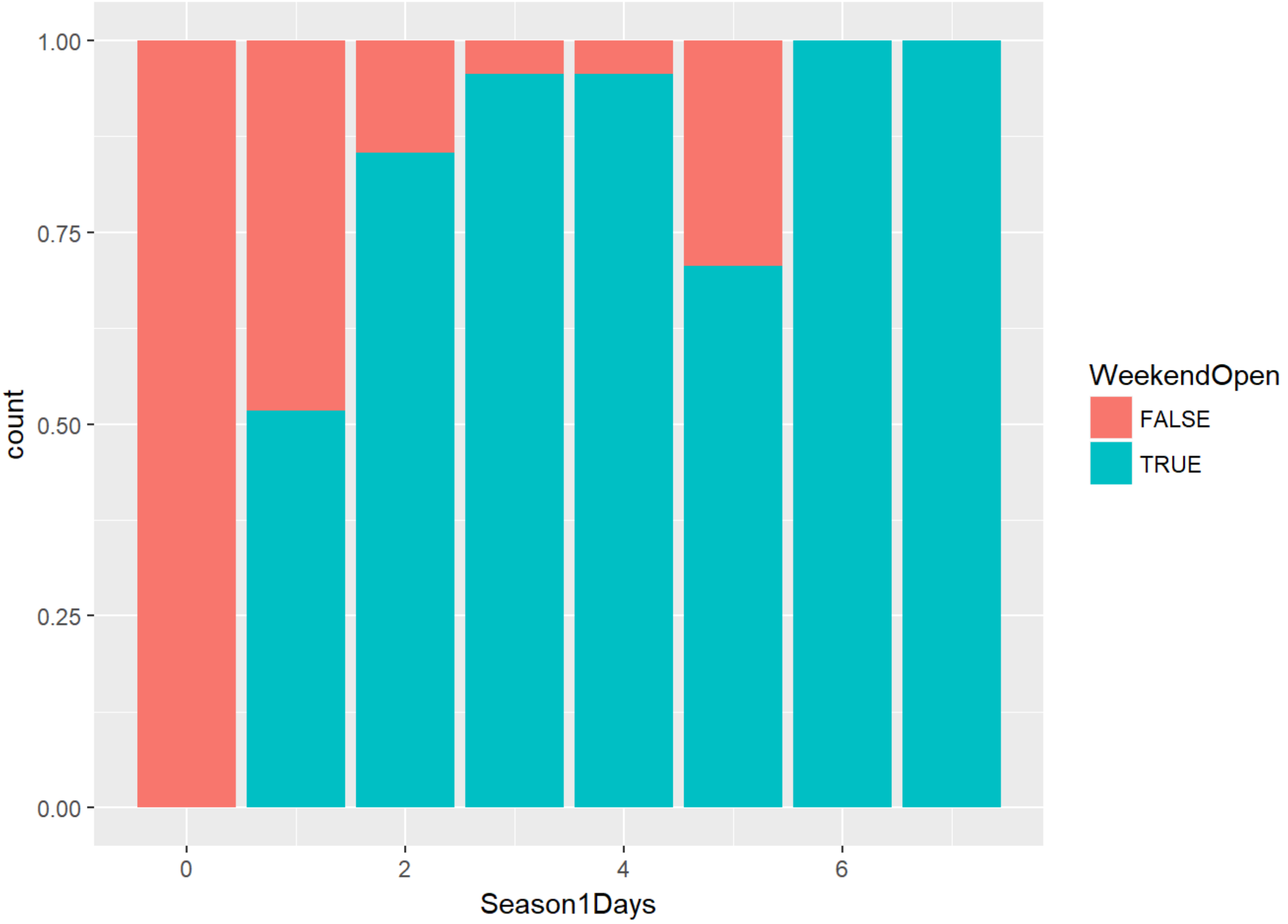
```
                                           Sat: 8:00 AM-11:00 AM;           1        TRUE
      TRUE            TRUE       TRUE       TRUE     TRUE      FALSE      FALSE     FALSE
## 17 1018365  Charleston              Illinois 05/20/2017 to 09/30/2017
                                           Sat: 8:00 AM-12:00 PM;           1        TRUE
      TRUE           FALSE       TRUE       TRUE     TRUE      FALSE      FALSE     FALSE
## 18 1012790    Chiefland              Florida 09/03/2016 to 06/24/2017
                                            Sat: 8:30 AM-1:00 PM;           1        TRUE
      TRUE           FALSE      FALSE       TRUE    FALSE      FALSE      FALSE     FALSE
## 19 1012158  Woodinville           Washington 01/01/2016 to 12/31/2016 Wed: 11:00 AM-6:00 PM;T
hu: 11:00 AM-6:00 PM;Fri: 11:00 AM-6:00 PM;Sat: 11:00 AM-6:00 PM;           4        TRUE
     FALSE           FALSE      FALSE      FALSE    FALSE       TRUE      FALSE     FALSE
## 20 1010873       Topeka               Kansas 06/05/2015 to 10/30/2015
                                           Fri: 7:30 AM-12:00 PM;           1       FALSE
      TRUE           FALSE       TRUE      FALSE    FALSE      FALSE      FALSE     FALSE
## # ... with 8,687 more rows
```

```
# First to see the count of variable Season1Days:
Q_all %>%
  ggplot(aes(Season1Days)) +
    geom_bar()
```



```
## Conclustion: we can see that most of the markets only open 0 or 1 day per week in Season1.

# Then, we discover relationship between Season1Days and WeekendOpen:
Q_all %>%
  ggplot() +
    geom_bar(aes(x= Season1Days, fill = WeekendOpen), position = "fill")
```



```
## Conclustion: in terms of proportion, the market tend to open at weekend when they open more t
han 3 days per week; when the market only open 1 day per week, 50% of the chance they would open
 exactly on weekend.

# 3rd, let's group by Season1Days, close_before_six and open_fewer_four to see the behaviour of
 the market:
Q_all %>%
  filter(!is.na(close_before_six), !is.na(open_fewer_four)) %>%
  group_by(Season1Days,close_before_six,open_fewer_four) %>%
  summarise(n= n())
```

```
## # A tibble: 24 x 4
## # Groups:   Season1Days, close_before_six [?]
##    Season1Days close_before_six open_fewer_four     n
##          <int>            <lgl>           <lgl> <int>
##  1           1            FALSE           FALSE   703
##  2           1            FALSE            TRUE   832
##  3           1             TRUE           FALSE  2543
##  4           1             TRUE            TRUE   702
##  5           2            FALSE           FALSE    21
##  6           2            FALSE            TRUE    21
##  7           2             TRUE           FALSE   373
##  8           2             TRUE            TRUE   144
##  9           3            FALSE           FALSE     8
## 10           3            FALSE            TRUE     3
## 11           3             TRUE           FALSE   119
## 12           3             TRUE            TRUE    22
## 13           4            FALSE           FALSE     4
## 14           4             TRUE           FALSE    17
## 15           4             TRUE            TRUE     1
## 16           5            FALSE           FALSE     5
## 17           5            FALSE            TRUE     1
## 18           5             TRUE           FALSE    27
## 19           5             TRUE            TRUE     1
## 20           6            FALSE           FALSE    14
## 21           6            FALSE            TRUE     1
## 22           6             TRUE           FALSE    29
## 23           7            FALSE           FALSE    21
## 24           7             TRUE           FALSE    26
```

```
## Conclusion: for the market that only open 1 day per week, the most possible case would be tha
t it close before 6pm and open more than 4 hours a day; this is also TRUE for the market open 2,
 3 and 4 day per week.

# 4th, geographic levels: group by state and Season1Days
Q_all %>%
  group_by(State,Season1Days) %>%
  summarise(n= n()) %>%
  arrange(desc(Season1Days), desc(n))
```

```
## # A tibble: 279 x 3
## # Groups:   State [53]
##             State Season1Days     n
##             <chr>       <int> <int>
##  1       Virginia           7     6
##  2          Texas           7     5
##  3        Florida           7     4
##  4       Michigan           7     4
##  5 North Carolina           7     3
##  6        Georgia           7     2
##  7       Illinois           7     2
##  8        Indiana           7     2
##  9       Kentucky           7     2
## 10  Massachusetts           7     2
## 11       Missouri           7     2
## 12       New York           7     2
## 13           Ohio           7     2
## 14 South Carolina           7     2
## 15      Wisconsin           7     2
## 16       Delaware           7     1
## 17         Hawaii           7     1
## 18       Maryland           7     1
## 19      Minnesota           7     1
## 20     New Jersey           7     1
## # ... with 259 more rows
```

```
## Conclusion: we can find Virginia has the most markets that open all days per week, Texas is next.

# 5th, let's find year_round city:
Q_all %>%
  filter(!is.na(year_round)) %>%
  group_by(State,city,year_round) %>%
  summarise(n= n()) %>%
  arrange(desc(year_round),desc(n))
```

```
## # A tibble: 6,316 x 4
## # Groups:   State, city [6,292]
##                   State              city year_round     n
##                   <chr>             <chr>      <lgl> <int>
##  1             New York          New York       TRUE     2
##  2             New York            Queens       TRUE     2
##  3              Alabama            Ariton       TRUE     1
##  4              Alabama       Gulf Shores       TRUE     1
##  5              Arizona     Goodyear, AZ        TRUE     1
##  6              Arizona           Peoria        TRUE     1
##  7             Arkansas        Perryville       TRUE     1
##  8             Arkansas         Van Buren       TRUE     1
##  9           California         Camarillo       TRUE     1
## 10           California    Half Moon Bay        TRUE     1
## 11           California       Los Angeles       TRUE     1
## 12           California   Manhattan Beach       TRUE     1
## 13           California         Pacifica        TRUE     1
## 14           California         San Diego       TRUE     1
## 15           California           Visalia       TRUE     1
## 16 District of Columbia        Washington       TRUE     1
## 17              Florida Fort Myers Beach        TRUE     1
## 18              Georgia           Roswell       TRUE     1
## 19             Illinois           Chicago       TRUE     1
## 20             Illinois       Tunnel Hill       TRUE     1
## # ... with 6,296 more rows
```

```
## Conclusion: we can find New York city and Queens city has the most markets that open all seasons
```

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.