DA5020 - Homework 5: Dates and Times

2019-10-13

Github

https://github.com/ajb7/R-workbooks/tree/master/workbook5 (https://github.com/ajb7/R-workbooks/tree/master/workbook5)

Read the data

```
# Installing dplyr library and importing dataset
# using read.csv to read csv data and read_excel to read data from xls file
#install.packages("dplyr")
library("dplyr")
library("readxl")
library("stringr")
library("lubridate")
#D:\UNIVERSITY\Assignments\DA 5020\Assegments\5
setwd("D:/UNIVERSITY/Assignments/DA 5020/Assegments/5")
usda <- read.csv("_farmers_market.csv_.csv", header = T, sep = ",")
```

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).

Answer: As days are seperated by semicolons, we count number of semicolons in each “Season1Days” column for each row sing “str_count()”.

```
usda$Season1Days <- str_count(usda$Season1Time, pattern=";")
usda[1:10,c("Season1Time", "Season1Days")]
```

	Season1Time <fctr>	**Season1Days** <int>
1	Wed: 9:00 AM-1:00 PM;	1
2	Sat: 9:00 AM-1:00 PM;	1
3		0
4	Wed: 3:00 PM-6:00 PM;Sat: 8:00 AM-1:00 PM;	2

	Season1Time	Season1Days
	<fctr>	<int>
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

1-10 of 10 rows

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

Answer: To see if the market is open, we parse “Season1Time” column of each row and look for “Sun” or “Sat” in them. If market is open on Saturday or Sunday, it is marked true as “WeekendOpen”

```
usda$WeekendOpen <- grepl("Sat|Sun", usda$Season1Time)
print(usda[1:10,c("Season1Time", "WeekendOpen")])
```

```
##                                      Season1Time WeekendOpen
## 1                         Wed: 9:00 AM-1:00 PM;       FALSE
## 2                         Sat: 9:00 AM-1:00 PM;        TRUE
## 3                                                     FALSE
## 4    Wed: 3:00 PM-6:00 PM;Sat: 8:00 AM-1:00 PM;        TRUE
## 5  Tue:8:00 am - 5:00 pm;Sat:8:00 am - 8:00 pm;        TRUE
## 6                         Tue: 3:30 PM-6:30 PM;       FALSE
## 7                        Tue: 10:00 AM-7:00 PM;       FALSE
## 8                        Fri: 8:00 AM-11:00 AM;       FALSE
## 9                         Sat: 9:00 AM-1:00 PM;        TRUE
## 10                        Sat: 9:00 AM-1:00 PM;        TRUE
```

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.

Method:

We parse “Season1Time” of each row. We first split each value in “Season1Time” by semicolon “;” , so that we have opening time for different days. For example, “Wed: 3:00 PM-6:00 PM;Sat: 8:00 AM-1:00 PM;” will be split into “Wed: 3:00 PM-6:00 PM” and “Sat: 8:00 AM-1:00 PM”.

In each such time, we split further by “-” and grab the second element to get closing time in each day. For example, “Wed: 3:00 PM-6:00 PM” will give “6:00 PM” as the output.

Next, we use strptime() to convert this format into 24 hour format and then into integer format. “6:00 PM” gives “18:00:00” coverted to “18”

Finally, we create a new column “closesBefore6” which has value “True” if above value is less than “18”, indicating that market closing time is before 6:00 PM.

We follow similar process to calculate average number of hours the market is open in week. We calculate the difference between start time and end time. For example “Wed: 3:00 PM-6:00 PM” will give us “15” and “18” after conversion to 24 hour format and then integer format. We subtract both to get the number of hours market is open.

We do this for every row and store the result in “avgOpenTime” column.

```
po <- usda[, c("Season1Time")]

#fdf <- usda[1:20,]
eachDay <- str_split(po, ";")
i = 1
for(day in eachDay){
  timeInDay <- str_split(day, "-")
  for(eachTime in timeInDay){
    endTime24 <- as.integer(format(strptime(eachTime[2], "%I:%M %p"), format="%H"))
    usda[i, "closesBefore6"] <- ""
    if(is.na(endTime24)){
      break
    }
    if(endTime24 < 18){
      usda[i, "closesBefore6"] <- "True"
      break
    }

  }
  i=i+1
}

i = 1
for(day in eachDay){
  timeInDay <- str_split(day, "-")
  j <- 1
  timeAvg <- 0
  for(eachTime in timeInDay){
    startTime <- str_split(eachTime[1], "^[a-zA-Z]*:")
    endTime24 <- as.integer(format(strptime(eachTime[2], "%I:%M %p"), format="%H"))
    startTime24 <- as.integer(format(strptime(startTime[[1]][2], "%I:%M %p"), format
="%H"))
    if(is.na(endTime24)){
      next
    }
    timeDiff <- endTime24 - startTime24

    if(j>1){
      timeAvg <- (timeAvg + timeDiff)/2
    }else{
      timeAvg <- timeDiff
    }
    j = j+1
  }

  usda[i, "avgOpenTime"] <- timeAvg
  i=i+1
}
```

```
print(str_c("Number of markets open for less than 4 hours: ", length(usda$MarketName[u
sda$avgOpenTime <=4])))
```

```
## [1] "Number of markets open for less than 4 hours: 6928"
```

```
print(str_c("Number of markets closing before 6PM: ", length(usda$MarketName[usda$clos
esBefore6 == "True"])))
```

```
## [1] "Number of markets closing before 6PM: 4116"
```

3. (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.)

Answer: Let us say that Fall Season is September to December, Summer is from June to September, Spring is from March to June, Winter is from January to February.

For each row, we split the "season1Date", "season2Date", "season3Date", "season4Date" by "to" so that we have start date and end date for each column.

Next, we get the month using "month.name" function out of each start date and end date formats, in numeric format. For example, January refers to 1, February refers to 2 and so on.

For each of the Season columns, we store their begin and end month reference. For example, "Season1Date" begin and end month are stored in "Season1BeginMonth", "Season1EndMonth" and so on.

Now that we have month for each season, we create 4 new columns, Spring, Summer, Winter and Fall. Based on pre-defined assumption that Spring season is between March and June, we check for all the rows where month value in either begin or end month column is between March and June.

Finally we have "True" values for each row in Spring column, if market is open on Spring based on data in "Season1Date". Same goes for Winter, Fall and Summer columns.

```r
getMonth <- function(monthName){
  mnth <- str_trim(monthName)
  if(is.na(mdy(mnth))){
    mnth <- match(mnth, month.name)
  }else{
    mnth <- month(mdy(mnth))
  }

  return(mnth)
}

for (row in 1:nrow(usda)) {

  season1 <- str_split(usda[row, "Season1Date"], "to")
  season2 <- str_split(usda[row, "Season2Date"], "to")
  season3 <- str_split(usda[row, "Season3Date"], "to")
  season4 <- str_split(usda[row, "Season4Date"], "to")



  if(!is.na(season1)){
    season1BeginMonth <- getMonth(season1[[1]][1])
    season1EndMonth <- getMonth(season1[[1]][2])
  }

  if(!is.na(season2)){
    season2BeginMonth <- getMonth(season2[[1]][1])
    season2EndMonth <- getMonth(season2[[1]][2])
  }

  if(!is.na(season3)){
    season3BeginMonth <- getMonth(season3[[1]][1])
    season3EndMonth <- getMonth(season3[[1]][2])
  }

  if(!is.na(season4)){
    season4BeginMonth <- getMonth(season4[[1]][1])
    season4EndMonth <- getMonth(season4[[1]][2])
  }

  usda[row, "season1BeginMonth"] <- season1BeginMonth
  usda[row, "season1EndMonth"] <- season1EndMonth

  usda[row, "season2BeginMonth"] <- season2BeginMonth
  usda[row, "season2EndMonth"] <- season2EndMonth

  usda[row, "season3BeginMonth"] <- season3BeginMonth
  usda[row, "season3EndMonth"] <- season3EndMonth
  usda[row, "season4BeginMonth"] <- season4BeginMonth
  usda[row, "season4EndMonth"] <- season4EndMonth
}

usda$Spring <- ""
usda$Fall <- ""
usda$Winter <- ""
usda$Summer <- ""

usda$Spring[!is.na(usda$season1BeginMonth) & (usda$season1BeginMonth > 2 & usda$season
1BeginMonth < 6)  ] <- "True"
usda$Spring[!is.na(usda$season2BeginMonth) & (usda$season2BeginMonth > 2 & usda$season
2BeginMonth < 6)  ] <- "True"
usda$Spring[!is.na(usda$season3BeginMonth) & (usda$season3BeginMonth > 2 & usda$season
3BeginMonth < 6)  ] <- "True"
usda$Spring[!is.na(usda$season4BeginMonth) & (usda$season4BeginMonth > 2 & usda$season
4BeginMonth < 6)  ] <- "True"

usda$Fall[!is.na(usda$season1BeginMonth) & (usda$season1BeginMonth > 8 & usda$season1B
eginMonth <= 12)  ] <- "True"
usda$Fall[!is.na(usda$season2BeginMonth) & (usda$season2BeginMonth > 8 & usda$season2B
eginMonth <= 12)  ] <- "True"
usda$Fall[!is.na(usda$season3BeginMonth) & (usda$season3BeginMonth > 8 & usda$season3B
eginMonth <= 12)  ] <- "True"
usda$Fall[!is.na(usda$season4BeginMonth) & (usda$season4BeginMonth > 8 & usda$season4B
eginMonth <= 12)  ] <- "True"

usda$Winter[!is.na(usda$season1BeginMonth) & (usda$season1BeginMonth >= 1 & usda$seaso
n1BeginMonth < 3)  ] <- "True"
usda$Winter[!is.na(usda$season2BeginMonth) & (usda$season2BeginMonth >= 1 & usda$seaso
n2BeginMonth < 3)  ] <- "True"
usda$Winter[!is.na(usda$season3BeginMonth) & (usda$season3BeginMonth >= 1 & usda$seaso
n3BeginMonth < 3)  ] <- "True"
usda$Winter[!is.na(usda$season4BeginMonth) & (usda$season4BeginMonth >= 1 & usda$seaso
n4BeginMonth < 3)  ] <- "True"

usda$Summer[!is.na(usda$season1BeginMonth) & (usda$season1BeginMonth >= 6 & usda$seaso
n1BeginMonth < 9)  ] <- "True"
usda$Summer[!is.na(usda$season2BeginMonth) & (usda$season2BeginMonth >= 6 & usda$seaso
n2BeginMonth < 9)  ] <- "True"
usda$Summer[!is.na(usda$season3BeginMonth) & (usda$season3BeginMonth >= 6 & usda$seaso
n3BeginMonth < 9)  ] <- "True"
usda$Summer[!is.na(usda$season4BeginMonth) & (usda$season4BeginMonth >= 6 & usda$seaso
n4BeginMonth < 9)  ] <- "True"
```

4. (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?

Answer:

Let us see which counties have most markets open in Fall.

```
viz1 <- filter(usda, Fall == "True") %>% group_by(County) %>% summarize(count=n()) %>%
arrange(desc(count))
viz1 %>% top_n(8)
```

County	**count**
<fctr>	<int>
Maricopa	25
Middlesex	9
Washington	9
Lee	8
Pima	8
Cook	7
Jackson	7
Los Angeles	7

8 rows

We see tat Maricopa, Middlsex, Washington have most markets open in Fall. Let us see, what Maricopa, Middlsex and Washington sell in Fall. We compare between Fruits, Wine, Seafood, Vegetables and Grains.

```r
viz2 <- filter(usda, County == "Maricopa" | County == "Middlesex" | County == "Washington", Fruits == "Y") %>% group_by(County) %>% summarize(count=n()) %>% arrange(desc(count))

viz3 <- filter(usda, County == "Maricopa" | County == "Middlesex" | County == "Washington", Wine == "Y") %>% group_by(County) %>% summarize(count=n()) %>% arrange(desc(count))

viz4 <- filter(usda, County == "Maricopa" | County == "Middlesex" | County == "Washington", Seafood == "Y") %>% group_by(County) %>% summarize(count=n()) %>% arrange(desc(count))

viz5 <- filter(usda, County == "Maricopa" | County == "Middlesex" | County == "Washington", Vegetables == "Y") %>% group_by(County) %>% summarize(count=n()) %>% arrange(desc(count))

viz6 <- filter(usda, County == "Maricopa" | County == "Middlesex" | County == "Washington", Grains == "Y") %>% group_by(County) %>% summarize(count=n()) %>% arrange(desc(count))

viz2
```

County <fctr>	**count** <int>
Middlesex	55
Washington	53
Maricopa	30

3 rows

```r
viz3
```

County <fctr>	**count** <int>
Middlesex	26
Washington	18
Maricopa	3

3 rows

```r
viz4
```

County	**count**
<fctr>	<int>
Middlesex	37
Washington	17
Maricopa	16

3 rows

```
viz5
```

County	**count**
<fctr>	<int>
Washington	62
Middlesex	60
Maricopa	30

3 rows

```
viz6
```

County	**count**
<fctr>	<int>
Washington	14
Maricopa	11
Middlesex	9

3 rows

We see that as expected, not enough Grains are farmed, but lots of Fruits and Vegetables are grown. Let us see if the trend is same in Summer too.

```
viz7 <- filter(usda, Summer == "True") %>% group_by(County) %>% summarize(count=n()) %
>% arrange(desc(count))
viz7 %>% top_n(8)
```

County	**count**
<fctr>	<int>
Middlesex	44
Worcester	40

County	count
<fctr>	<int>
Cook	33
Essex	32
Bronx	26
New York	26
Wayne	26
Philadelphia	24

8 rows

We observe that Middlesex still appears in the top list of markets in summers, but the other counties are not seen. Let us see, which counties have most markets open all through the year.

```
viz8 <- filter(usda, Fall == "True", Summer == "True", Winter == "True", Spring == "Tr
ue") %>% group_by(County) %>% summarize(count=n(), avg_open_time = mean(avgOpenTime, n
a.rm=TRUE)) %>% arrange(desc(count))
viz8 %>% top_n(8)
```

County	count	avg_open_time
<fctr>	<int>	<dbl>
Fulton	1	3

1 row

Only Fulton is such county which has market open all through the year with average time of 3 hours everyday. Let us see how many markets are open half year, i.e; in Fall and Summer.

```
viz9 <- filter(usda, Fall == "True", Summer == "True") %>% group_by(County) %>% summar
ize(count=n()) %>% arrange(desc(count))
viz9 %>% top_n(8)
```

County	count
<fctr>	<int>
Maricopa	4
Middlesex	3
District of Columbia	2
Franklin	2
Milwaukee	2

County	count
<fctr>	<int>
Monroe	2
Pinellas	2
Plymouth	2

8 rows

Middlesex and Maricopa top the list of markets open in Fall and Summer both seasons. Lets see counties with most organic markets.

```
viz10 <- filter(usda, Organic == "Y") %>% group_by(County) %>% summarize(count=n()) %
>% arrange(desc(count))
viz10 %>% top_n(8)
```

County	count
<fctr>	<int>
Los Angeles	62
King	37
Santa Clara	32
Orange	30
San Diego	30
District of Columbia	29
Middlesex	28
Alameda	25
Washington	25

9 rows

Finally, lets see states where markets are open on average for most number of hours in the day.

```
viz11 <- filter(usda) %>% group_by(County) %>% summarize(count=n(), avg_open_time = me
an(avgOpenTime, na.rm=TRUE)) %>% arrange(desc(avg_open_time))
viz11 %>% top_n(8)
```

County	count	avg_open_time
<fctr>	<int>	<dbl>
Sanilac	1	16.0

County	**count**	**avg_open_time**
<fctr>	<int>	<dbl>
Amelia	1	15.0
Bartow	1	15.0
Pamlico	1	15.0
San Sebastian	2	13.5
Emporia	1	13.0
Tishomingo	1	12.0
Oregon	1	10.0
Rhea	1	10.0
Tate	1	10.0

1-10 of 11 rows Previous **1** 2 Next

We see that Sanilac and Amelia have markets open for on an average 16 hours everyday. This probably means, it is easier to shop for customers and competitive to sell for farmers.