

Assignment_5

Vinisha

October 3, 2017

- GitHub: <https://github.com/NUDA5020/homework-vinishag>

```
library(tidyverse)
```

```
## Loading tidyverse: ggplot2
## Loading tidyverse: tibble
## Loading tidyverse: tidyr
## Loading tidyverse: readr
## Loading tidyverse: purrr
## Loading tidyverse: dplyr
```

```
## Conflicts with tidy packages -----
```

```
## filter(): dplyr, stats
## lag():    dplyr, stats
```

```
library(lubridate)
```

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

```
library(stringr)
library(magrittr)
```

```
##
## Attaching package: 'magrittr'
## The following object is masked from 'package:purrr':
##
##     set_names
## The following object is masked from 'package:tidyr':
##
##     extract
```

```
farmer <- read.csv("farmer_market.csv")
```

```
#Part1
```

```
#creating a pattern to extract just the days of the week
colpattern <- "[:](.*)"
```

```
#split and replace it with blanks
lst <- strsplit(as.character(farmer$Season1Time),';',fixed=TRUE)
```

```
# craete a data frame with number of days
d1 <- data.frame(lapply(lst, `length<-`, max(lengths(lst))) )
colnames(d1) <- paste0('V', seq_along(d1))
```

```

y <- as.data.frame(lapply(d1, function(z) gsub(colpattern, "", z)))

# assign a new column with number of days
farmer$SeasonDays <- apply(y, 2, function(m)length(which(unique(m) != "")))

# display just a portion of data frame for readable output
head(data.frame(farmer$MarketName, farmer$Season1Time, farmer$SeasonDays),10)%>%
  knitr::kable()

```

farmer.MarketName	farmer.Season1Time	farmer.SeasonDays
Caledonia Farmers Market Association - Danville	Wed: 9:00 AM-1:00 PM;	1
Stearns Homestead Farmers' Market	Sat: 9:00 AM-1:00 PM;	1
106 S. Main Street Farmers Market		0
10th Steet Community Farmers Market	Wed: 3:00 PM-6:00 PM;Sat: 8:00 AM-1:00 PM;	2
112st Madison Avenue	Tue:8:00 am - 5:00 pm;Sat:8:00 am - 8:00 pm;	2
12 South Farmers Market	Tue: 3:30 PM-6:30 PM;	1
125th Street Fresh Connect Farmers' Market	Tue: 10:00 AM-7:00 PM;	1
12th & Brandywine Urban Farm Market	Fri: 8:00 AM-11:00 AM;	1
14&U Farmers' Market	Sat: 9:00 AM-1:00 PM;	1
14th & Kennedy Street Farmers Market	Sat: 9:00 AM-1:00 PM;	1

#Part2

```

# checking for string if it matches with Sat or sun depicting weekends
count_weekend_table <- apply(y, 2, function(m)length(which(unique(m) %in% c("Sat", "Sun"))))

# Assign a new column for giving a deciosion if the market is open on weekends or not
farmer$WeekendOpen <- ifelse(count_weekend_table > 0, TRUE, FALSE)

# making the output more readable
data.frame(farmer$MarketName, farmer$Season1Time, farmer$WeekendOpen) %>%
  head(10)%>% knitr::kable()

```

farmer.MarketName	farmer.Season1Time	farmer.WeekendOpen
Caledonia Farmers Market Association - Danville	Wed: 9:00 AM-1:00 PM;	FALSE
Stearns Homestead Farmers' Market	Sat: 9:00 AM-1:00 PM;	TRUE
106 S. Main Street Farmers Market		FALSE
10th Steet Community Farmers Market	Wed: 3:00 PM-6:00 PM;Sat: 8:00 AM-1:00 PM;	TRUE
112st Madison Avenue	Tue:8:00 am - 5:00 pm;Sat:8:00 am - 8:00 pm;	TRUE
12 South Farmers Market	Tue: 3:30 PM-6:30 PM;	FALSE
125th Street Fresh Connect Farmers' Market	Tue: 10:00 AM-7:00 PM;	FALSE
12th & Brandywine Urban Farm Market	Fri: 8:00 AM-11:00 AM;	FALSE
14&U Farmers' Market	Sat: 9:00 AM-1:00 PM;	TRUE
14th & Kennedy Street Farmers Market	Sat: 9:00 AM-1:00 PM;	TRUE

#Part3

```

farmer %>%
  group_by(Season1Time) %>%
  mutate(
    # extracting the opening time for markets

```

```

opening_time = str_match(Season1Time, ("\\d+:\\d+ [A-Z a-z]+")),
# regex for extracting closing time
last_regex = str_c("(?:-)", "\\d+:\\d+ [A-Z]+", "|", "(?:- )", "\\d+:\\d+ [A-Z a-z]+"),
# extracting the closing time for markets
closing_time = str_match(Season1Time, last_regex),
# further cleaning up the closing time
closing_time = str_replace_all(closing_time, "-", ""),
# parsing opening and closing time into date,time variable
start = parse_date_time(opening_time, "%I:%M %p"),
end = parse_date_time(closing_time, "%I:%M %p")
) %>%
# removing rows with missing variable
filter(!is.na(closing_time)) %>%
mutate(
  # checking to see which shops open for less than 4 hours
  open_brief = (end - start) < 4,
  # checking which close before 6 pm
  early_shut = hour(end) < 18
) %>%
select(open_brief, early_shut, closing_time, opening_time, Season1Time, MarketName) %>%
# display 10 results
head(10) %>%
# making the output more readable
knitr::kable()

```

```

## Warning: All formats failed to parse. No formats found.
## Warning: All formats failed to parse. No formats found.
## Warning: All formats failed to parse. No formats found.
## Warning: All formats failed to parse. No formats found.
## Warning: All formats failed to parse. No formats found.
## Warning: All formats failed to parse. No formats found.
## Warning: All formats failed to parse. No formats found.
## Warning: All formats failed to parse. No formats found.
## Warning: All formats failed to parse. No formats found.
## Warning: All formats failed to parse. No formats found.
## Warning: All formats failed to parse. No formats found.
## Warning: All formats failed to parse. No formats found.
## Warning: All formats failed to parse. No formats found.
## Warning: All formats failed to parse. No formats found.
## Warning: All formats failed to parse. No formats found.
## Warning: All formats failed to parse. No formats found.

```


open_brief	early_shut	closing_time	opening_time	Season1Time	MarketName
FALSE	TRUE	1:00 PM	9:00 AM	Wed: 9:00 AM-1:00 PM;	Caledonia Far
FALSE	TRUE	1:00 PM	9:00 AM	Sat: 9:00 AM-1:00 PM;	Stearns Home
TRUE	FALSE	6:00 PM	3:00 PM	Wed: 3:00 PM-6:00 PM;Sat: 8:00 AM-1:00 PM;	10th Steet Co
FALSE	TRUE	5:00 pm	8:00 am	Tue:8:00 am - 5:00 pm;Sat:8:00 am - 8:00 pm;	112st Madison
TRUE	FALSE	6:30 PM	3:30 PM	Tue: 3:30 PM-6:30 PM;	12 South Farn
FALSE	FALSE	7:00 PM	10:00 AM	Tue: 10:00 AM-7:00 PM;	125th Street I
TRUE	TRUE	11:00 AM	8:00 AM	Fri: 8:00 AM-11:00 AM;	12th & Brand
FALSE	TRUE	1:00 PM	9:00 AM	Sat: 9:00 AM-1:00 PM;	14&U Farmer
FALSE	TRUE	1:00 PM	9:00 AM	Sat: 9:00 AM-1:00 PM;	14th & Kenne
FALSE	FALSE	6:30 PM	2:30 PM	Wed: 2:30 PM-6:30 PM;	170 Farm Sta

#Part4

```

# create date ranges for the four seasons as well
# as half year and year across
winter_start = as.Date("2016-12-01")
spring_start = as.Date("2016-03-01")
summer_start = as.Date("2016-06-01")
fall_start   = as.Date("2016-09-01")

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   <- interval(spring_start, fall_start)
year_round  <- interval(spring_start, spring_start)

Seasons <- farmer %>%
  filter(Season1Date != "") %>%
  mutate(
    # consolidating all dates to be of same year
    Season1Date = str_replace_all(Season1Date, "\\d+ ", "2016 "),
    Season1Date = str_replace_all(Season1Date, "\\d+$", "2016"),
    # extract the begin and end dates
    start = str_extract(Season1Date, "(\\d+\\\\\\\\\\d+\\\\\\\\\\d+)?"),
    finish = str_extract(Season1Date, "(\\d+\\\\\\\\\\d+\\\\\\\\\\d+)$"),
    # parsing the values into required format
    Season1BeginDate = mdy(start),
    Season1EndDate   = mdy(finish)
  ) %>%
  # cleaning up the missing rows
  filter(!is.na(Season1EndDate)) %>%
  mutate(
    Season1DateRange = interval(Season1BeginDate, Season1EndDate),
    # The column will now contain
    # TRUE/FALSE values returned by `int_overlap`
    Winter      = int_overlaps(Season1DateRange, winter),
    Spring      = int_overlaps(Season1DateRange, spring),
    Summer      = int_overlaps(Season1DateRange, summer),
    Fall        = int_overlaps(Season1DateRange, fall),
    Half_Year   = int_overlaps(Season1DateRange, half_year),
    Year_Round  = int_overlaps(Season1DateRange, year_round)
  )

```

```

) %>%
select(Winter, Spring, Summer, Fall, Half_Year, Year_Round,
       Season1Date) %>%
# displaying 10 random outputs
sample_n(10) %>%
# making the output more readable
knitr::kable()

```

Warning: 15 failed to parse.

Seasons

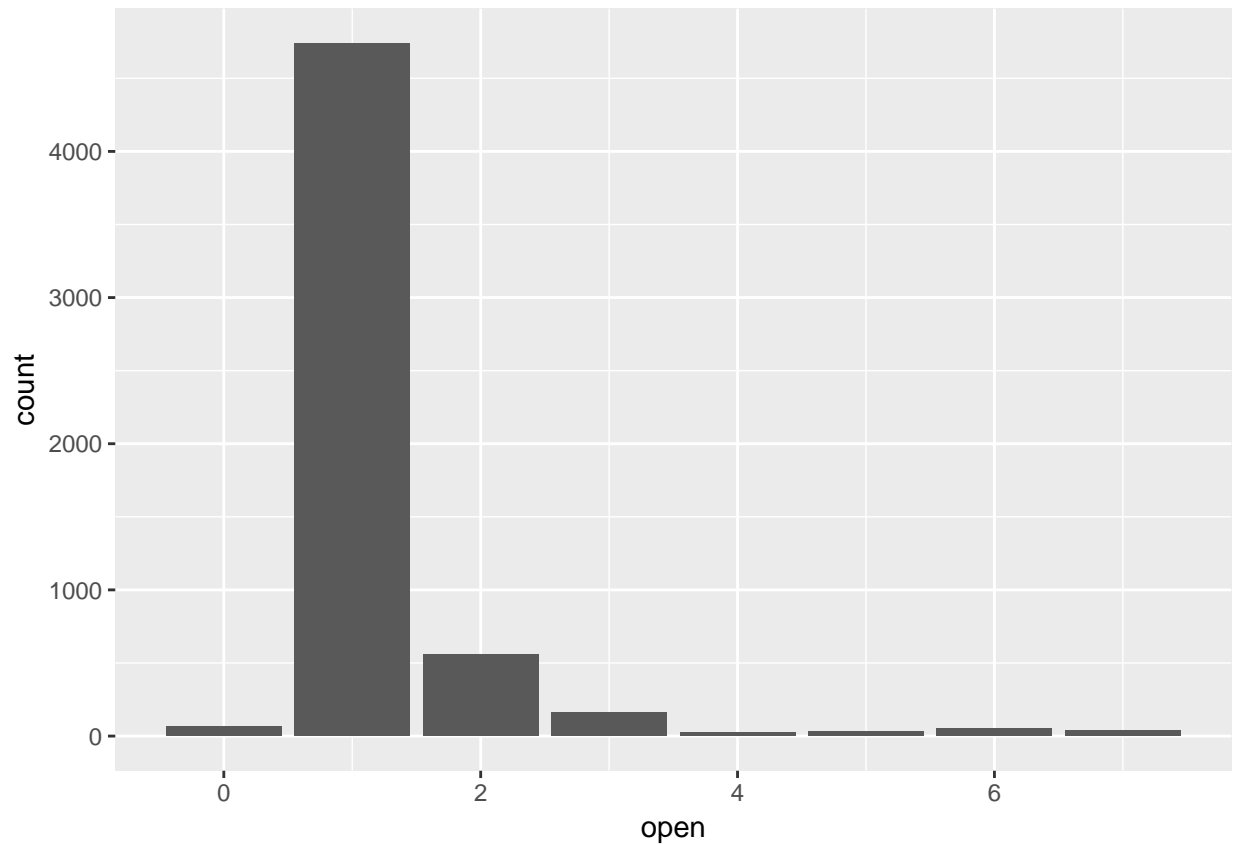
	Winter	Spring	Summer	Fall	Half_Year	Year_Round	Season1Date
1482	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	05/01/2016 to 10/01/2016
1365	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	04/05/2016 to 09/27/2016
1600	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	06/11/2016 to 09/03/2016
4117	FALSE	FALSE	TRUE	FALSE	TRUE	FALSE	06/13/2016 to 08/29/2016
2199	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	05/03/2016 to 10/25/2016
2652	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	06/13/2016 to 09/05/2016
1611	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	06/07/2016 to 09/27/2016
634	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	07/05/2016 to 10/25/2016
1855	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	06/13/2016 to 10/03/2016
2394	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	06/18/2016 to 09/24/2016

#Part5

```

Season1Days <- farmer %>%
  filter(Season1Time != "") %>%
  mutate(
    open = str_count(Season1Time, ("Mon|Tue|Wed|Thu|Fri|Sat|Sun"))
  ) %>%
  select(MarketName, Season1Time, open)
# plotting the number of shops open vs
# number of open days
ggplot(Season1Days, aes(open)) +
  geom_bar()

```



```
WeekendOpen <- farmer %>%  
  filter(Season1Time != "") %>%  
  mutate(  
    weekend = str_detect(Season1Time, ("Sat|Sun"))  
  ) %>%  
  select(MarketName, Season1Time, weekend)  
# confirming that more shops are open on  
# weekends than the number that are closed  
ggplot(WeekendOpen, aes(weekend, fill = weekend)) +  
  geom_bar()
```

