HW#4: Strings and Factors (DA5020)

Milad Tatari

10/3/2019

Preparation

Download US Farmers Markert Directory data from the website of USDA (click on "Export to Excel"). Rename the file as *farmers_market.csv*.

Download the Know Your Farmer, Know Your Food Projects dataset and name it as *kyfprojects.xls*. Put it into the same folder.

Read the data:

Warm Up

This dataset stores city and state in different columns, what if you want to print out city and state in the format "City, State"?

Questions

Please edit this file and add your own solutions to these questions. Make your output as readable as possible. Next time you would need to create this file on your own. Feel free to try out other templates (e.g. Tufte Handout) if your are familiar with LaTex. But for whatever template you choose, you should always include a link to your GitHub repo at the first page of your PDF.

1. (20 points) Cleanup the `Facebook` and `Twitter` column to let them contain only the facebook username or twitter handle name. I.e., replace "https://www.facebook.com/pages/Cameron-Park-Farmers-Market/97634216535?ref=hl" with "Cameron-Park-Farmers-Market", "https://twitter.com/FarmMarket125th" with "FarmMarket125th", and "@21acres" with "21acres".

```
na.vals <- c("", "NA", "n/a", "N/A", "none")
fmarkets <- read_csv("farmers_market.csv", na = na.vals)
kyfp <- read_xls("kyfprojects.xls", na = na.vals)
```

```
# Identify patterns

#For Facebook, we want to extract facebook username components, but also retain fixable irregular data

#- a page name "Glendale Farmers Market"
#- a group page "/group/xxx/"

#There are many variants of free-format URLs, regular expressions are so powerful that you can match th



re_facebook <- str_c(
  "(?i)",          # regex flag: case incensitive,
                   # check help doc: ?stri_opts_regex
  "^(?:.*(?:f?acebook|fb)(?:\\.com)?[/ ]?",  # the domain
  "(?:#\\!/)?(?:pages/)?)?",     # extraneous parts "#!/pages/"
  "(?:@)?",        # some FB records also contains "@"
  "([^#]*?)",      # the actual username we want
```

```
                  # (or page name, or fb group url)
  "/?",           # the extraneous slash at the end
  "(?:\\?.*)?$"   # the query strings, eg. ?ref=hl
                  # `$` is a must here, otherwise lazy quantifier
                  # will prevent the pattern to match all the way
                  # till the end of the string
)

#For twitter, the solution is much simpler. There are only limited variants of twitter urls:

#-  `https://twitter.com/@xxxx`
#-  `ColoradoFreshMarkets@COfreshmarkets`
#-  `https://twitter.com/#!/GreendaleRec`

#Therefor a twitter handle can be matched as the last occurence of alphanumeric characters, including p
#right after the last appearance of an "/" or "@".

#We use a greedy match `".*"` to match anything that's before part we actually want.


re_twitter <- str_c(
  "(?i)",
  "(?:.*[\\@/])?([a-z0-9_]+)"
)

# Execute the regex


fmarkets <- fmarkets %>%
  mutate(
    Facebook.clean = Facebook %>%
      str_replace(re_facebook, "\\1"),
    # Above two lines are equivalent to:
    #    Facebook.clean = str_replace(Facebook, re_facebook, "\\1"),
    # using pipes makes the code more readable.

    # Change empty strings to NA
    Facebook.clean = ifelse(Facebook.clean == "", NA, Facebook.clean),

    Twitter.clean = Twitter %>%
      str_replace(re_twitter, "\\1"),

    Twitter.clean = ifelse(Twitter.clean == "", NA, Twitter.clean)
  )

#`Facebook` is the original column name. `Facebook.clean` is the name of the new column we want to add.
# You can access column names directly inside dplyr pipes, without putting them in quotes or use the do

#"`\\1`" is a "back reference", meaning the first captured matching group.
#Normally every pair of parentheses `(...)` will create a matching group,
#but since we used "non-capturing group" `(?:...)` for most of them, the first
#captured group is limited to contain only the part we want.
```

```
#We directly modified the original data frame and added new columns.
#In practice, if you are sure your cleaning will not lose important
#information, you can just override the existing column.
```

2. (20 points) Clean up the `city` and `street` column. Remove state and county names from the `city` column and consolidate address spellings to be more consistent (e.g. "St.", "ST.", "Street" all become "St"; "and" changes to "&", etc...).

```
na.vals <- c("", "NA", "n/a", "N/A", "none")
fmarkets <- read_csv("farmers_market.csv", na = na.vals)
kyfp <- read_xls("kyfprojects.xls", na = na.vals)
```

```
#To clean the data, we will remove whatever comes after "," in the city column since I
#saw some states and counties are added #to city names. Some cities have the fisrs letter
#small, it is #chaning to capital as well.
fmarkets <- fmarkets %>%
  mutate(cityclean = str_replace(fmarkets$city,",(.*)",""))
  fmarkets$cityclean<-str_replace(fmarkets$cityclean,"^[a-z]", "\\U")
# ,(.*) matches anything after and replace it with empty. It was possible to do it with str_remove as w
# ^[a-z] matches any small starting character and  gets replaced by capital by \\U.

#Street column is not very organized. What I do is to change any st,ST,sT,St words to
#St to be more organized. Furtheremore, I will change and to &.

# . matches any character except line break
# [...] matches characters in the bracket
fmarkets <- fmarkets %>%
  mutate(streetclean = str_replace(fmarkets$street,"[sS][tT](.*)([ .]*)"," St"))
  fmarkets$streetclean<-str_replace(fmarkets$streetclean,"[aA][nN][dD]", "&")
```

3. (20 points) Create a new data frame (tibble) that explains the online presence of each state's farmers market. I.e., how many percentages of them have a facebook account? A twitter account? Or either of the accounts? (Hint: use the `is.na()` function)

```
library(tibble)
#First we need to know how many markets have facebook or twitter accounts by removing NA
#values as F1 and T1. F2 and T2 perentage of those who have the accounts out of all market.
#FTor is the percentage of those who have either Facbook or Twitter (at least one of them).
#FTand is the numbers that have both accounts.
F1<-!is.na(fmarkets$Facebook)
T1<-!is.na(fmarkets$Twitter)
F2<-length(F1[F1==TRUE])/length(fmarkets$Facebook)*100# percentage calculation
T2<-length(T1[T1==TRUE])/length(fmarkets$Twitter)*100
FT= !is.na(fmarkets$Facebook) | !is.na(fmarkets$Twitter)
FTor<-length(FT[FT==TRUE])/length(fmarkets$Facebook)*100
FT1= !is.na(fmarkets$Facebook) & !is.na(fmarkets$Twitter)
FTand<-length(FT1[FT1==TRUE])/length(fmarkets$Facebook)*100

# Now let's create a dataframe table showing the above information accordingly based on
#Markets Social Media (MSM)
MSM <- data_frame(Markets.Social.Media = c("Facebook Account Holders", "Twitter Acount Holders", "Faceb
                     Percentages = c(F2, T2, FTor,FTand))
```

```
## Warning: `data_frame()` is deprecated, use `tibble()`.
## This warning is displayed once per session.
```

```
MSM
```

```
## # A tibble: 4 x 2
##   Markets.Social.Media                Percentages
##   <chr>                                     <dbl>
## 1 Facebook Account Holders                   47.4
## 2 Twitter Acount Holders                     11.7
## 3 Facebook or Twitter Account holders        47.6
## 4 Facebook & Twitter Account holders         11.5
```

4. (20 points) Some of the farmer market names are quite long. Can you make them shorter by using the `forcats::fct_recode` function? Create a plot that demonstrates the number of farmers markets per location type. The locations should be ordered in descending order where the top of the graph will have the one with the highest number of markets.

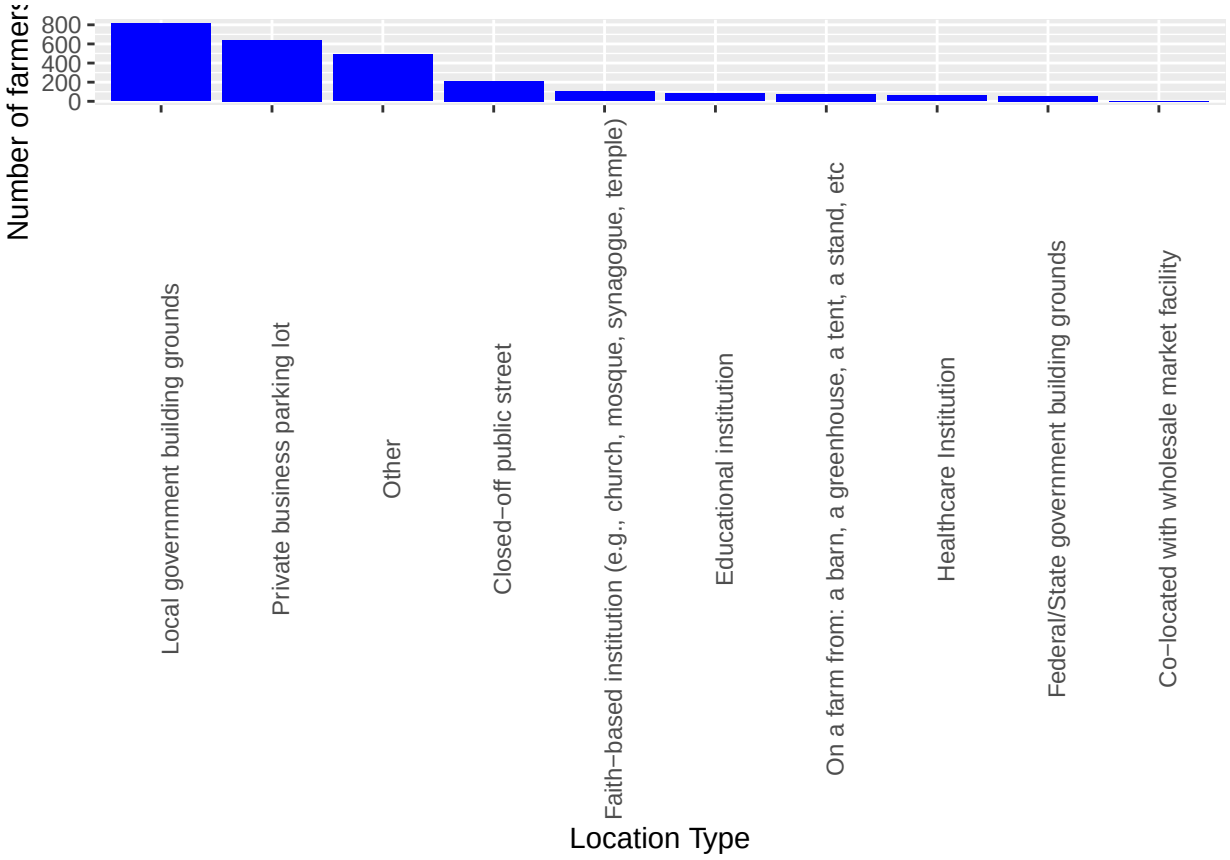
```
library(forcats)
library(dplyr)
# First, we want to shorten some markets name as they are too long. To do this we use package
#"forcats" and function of fct_recode. It gets  exact long names as appear below and will
#replace them accordingly by assigned short names. I have used this command to show that I
#have learnt how to use it. In the next part I am using a general method to shorten tha names.
fmarkets$MarketName<-as.factor(fmarkets$MarketName)
p<-fmarkets %>%
  mutate(MarketNamenew = fct_recode(MarketName,
                                    "Main Market" = "106 S. Main Street Farmers Market",
                                    "Business Market"  ="22nd Annual Highlands Business Partnership Farmers
                                    "HOPE FARM" = "All Natural & Certified Organic Farmer Market (HOPE FARM
                                    "Festival Market" = "American National Bank Downtown Farmers Market Fes
                                    "Newport Market" = "Appalachian (Newport) Farmers Market, Cocke County
                                    "Arcadia Community Market" = "Arcadia's Mobile Market -- Community of M
                                    "Harmony Market" = "Berkshire Harmony Downtown Pittsfield Farmers Marke
                                    "Sierra Market" = "Bishop Farmers' Market - Eastern Sierra Certified Fa

                                     ))
fmarkets.shortened<-as.data.frame(p)
# Now, I want to add a new column to make the abbreviation in general pattern in all observations
#like changing farmers market (also farmers' market) to F.M. Actually it is a more general
#way to shorten the names of all rows (markets)
fmarkets.shortened.revised<-fmarkets.shortened %>%
  mutate(name.abb.revised=str_replace(fmarkets.shortened$MarketNamenew,"[fF][aA][rR][mM][eE][rR][sS]('*


#part b
#we want to categorize market based on location types and then sort them. Finally we want
#to plot the as a descending order using ggplot.
Categ.Loc<-fmarkets[!is.na(fmarkets$Location), ]# Getting rid of those NA locations
Categ.Loc$Location<-as.factor(Categ.Loc$Location)

Categ.Loc<-Categ.Loc %>%
  mutate(Location = fct_infreq(Location))%>%# placing a new column as categorical locations
```

```
  count(Location)# counting the location types
# plotting using ggplot, set the axis legend, rotating axis text by 90 to fit in.
ggplot(Categ.Loc,aes(Location,n)) +
geom_col(fill="blue")+
  theme(axis.text.x = element_text(angle = 90))+
  labs(x="Location Type",y="Number of farmers markets")
```



5. (20 points) Write code to sanity check the `kyfprojects` data. For example, does `Program Abbreviation` always match `Program Name` for all the rows? (Try thinking of your own rules, too.)

```
# For this problem, first, I will delete all the sapce and small letters such that I can
#have only the capital letters left. Then, I will compare it with abbreviation names as
#boolean data type as the output. Then I will count the number of true and fulse.
#The false number is all the mismatch cases which is : 149
kyfp<- kyfp %>%
  mutate(
    correct.ABB= gsub("[:a-z:]","",kyfp$`Program Name`))#removing small letters
kyfp$correct.ABB<-gsub(" |-|//.|[0-9]","",kyfp$correct.ABB)#removing spaces and - and numbers
Sanity.check<-kyfp$`Program Abbreviation`==kyfp$correct.ABB#matching two columns to see when they have
table(Sanity.check)#counting the number of flase and true variables
```

```
## Sanity.check
## FALSE  TRUE
##   149  2230
```

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.