Report 5

Jie Gu

SID: 913953707

Part I~III: The Design of read_post and read_all_posts

In my read_all_posts function, I list files in a directory firstly, then I apply my read_post function to all the files in the directory. Next, I change the information I get into a data frame. Finally, I remove the column names and give them row names.

I have to make some design changes to read_post so that it works well with the read_all_posts function. More precisely, instead of reading the whole text together in read_post, I choose to read each character: title, text, date, price, latitude, longitude, bedrooms, bathrooms and sqft separately and return them together in read_post function. Also, as for the text in a single post, I combine them together by using paste function so that I can put it into a single unit of a data frame.

The columns in my data frames are title, text, date, price, latitude, longitude, bedrooms, bathrooms and sqft since these characters can be extracted easily at first and these characters are important when renting an apartment. The row in my data frame all the single posts in the directory. By doing that, I can observe each character in each post easily. My choice of rows and columns are convenient for further string process. For example, when trying to extract price from title, I can just focus on the titles in my data frame. Also, when trying to extract other information in the text, I can just focus on the text in my data frame.

Part IV: Comparison between rental price from titles and user-specified price

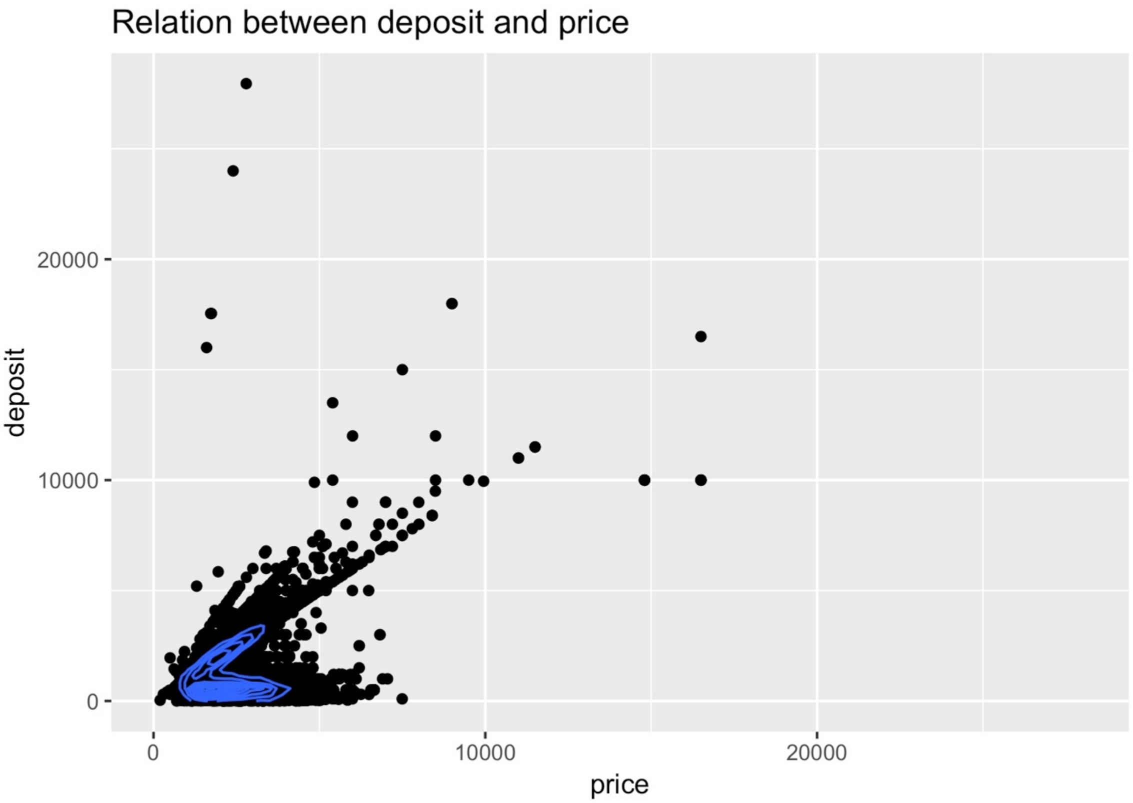
Overall Strategy: Since all the titles have price at the beginning of the title with a $ and a specific number, I can easily extract a $ with the following specific number and then remove the $. After that, I can make a difference between the rental price from titles and user-specified price.

The rental price from titles and user-specified price are all the same since all of their differences equal to 0.

Part V: The relationship between rental price and deposit amount

Overall Strategy: In the text, the deposit has forms like 'Deposit: $money', '$money Deposit', and 'The deposit is $Money' etc. So, what I do here is to extract a sentence with 'Deposit' or 'deposit' from the text at first. And since some text do not have period to split each sentence, I set a limit on that: in a long sentence without period, we only read 30 characters before deposit and 30 characters after deposit. We do need to worry too much about the pet deposit here since the deposit amount always appears in front of the pet deposit. Therefore, when we extract a sentence containing deposit, we are more likely to get deposit amount. After that, I extract $ with a following number in each sentence containing deposit and then remove the $.

To investigate the relationship between rental price and deposit amount, I adjust the price first like I did in the report 3 since some prices are extremely large and some prices are too small. I correct the price $9951095 as $995 and the price $34083724 as $3408 since they are the range in the text. Also, there are many prices smaller than $100. I delete them since most of them are repairing information and advertisements like improving bad credit. Now let’s see a dot plot of relationship between deposit and price.



From this dot plot, we can see that for the most points, with the increase of price, the deposit amount also increases. In other words, there may exists a linear relationship between the price and deposit, and the slope is positive. Also, by the density showed in the graph, most apartments have a price roughly lower than $4000 and a deposit amount roughly lower than $4000. There are also some points with a relatively low price but with a really high deposit. These points can be considered as outliers since it is almost impossible for an apartment to have a deposit much larger than its rental price.

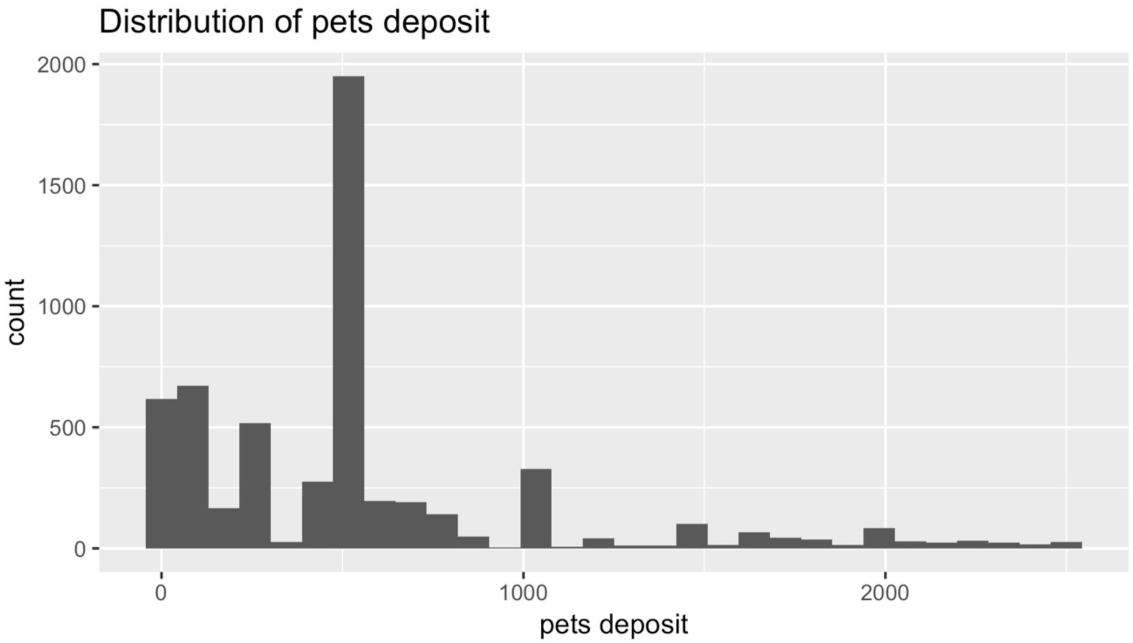
Part VI: Investigate on pets' policy and pets deposit

Overall Strategy for pets' policy: At first, I detect the pattern containing the 'cats'. The word 'cation' should be avoided in this part. Then, I detect the pattern that cats are not allowed. So, the patterns that cats are allowed should be the pattern containing the 'cats' but without the pattern that cats are not allowed. The method is similar for the patterns that dogs are allowed. Then we can get the situation that both cats and dogs are allowed by combing the situation that cats are allowed and dogs the situation that dogs are allowed. The method for the situation that none of them are allowed is similar. Next, since if an apartment does not mention the pets' policy, it usually does not have a limitation on pets. Therefore, we consider the unknown situation as the situation that both cats and dogs are allowed. By doing these things, we get the result: 27780 apartments allow both cats and dogs, 15928 apartments allow only cats, 1690 apartments allow dogs and 447 apartments allow none of them.

For the other kinds of pets, we can consider the most common pets: fish, bird, and rabbit. The method to get the situation for these pets is similar to the method for dogs and cats. The result is that there are 559 apartments allow these kinds of pets.

Overall Strategy for pets' deposit: For the pets' deposit, there are three forms. The first is that '$money…...pets……deposit'; the second is that 'pets……$money……deposit'; the third is that 'pets……deposit……$money'. The methods for these three forms are similar. Firstly, I extract the patterns from the text. Then, I extract the $money from the patterns and remove the $. Finally, I put the money getting from each pattern into the data frame. Also, since pets deposit almost cannot be larger than $2500, I just delete the deposit over $2500.

Now, let's take a look at a histogram of the distribution of the pets' deposits.



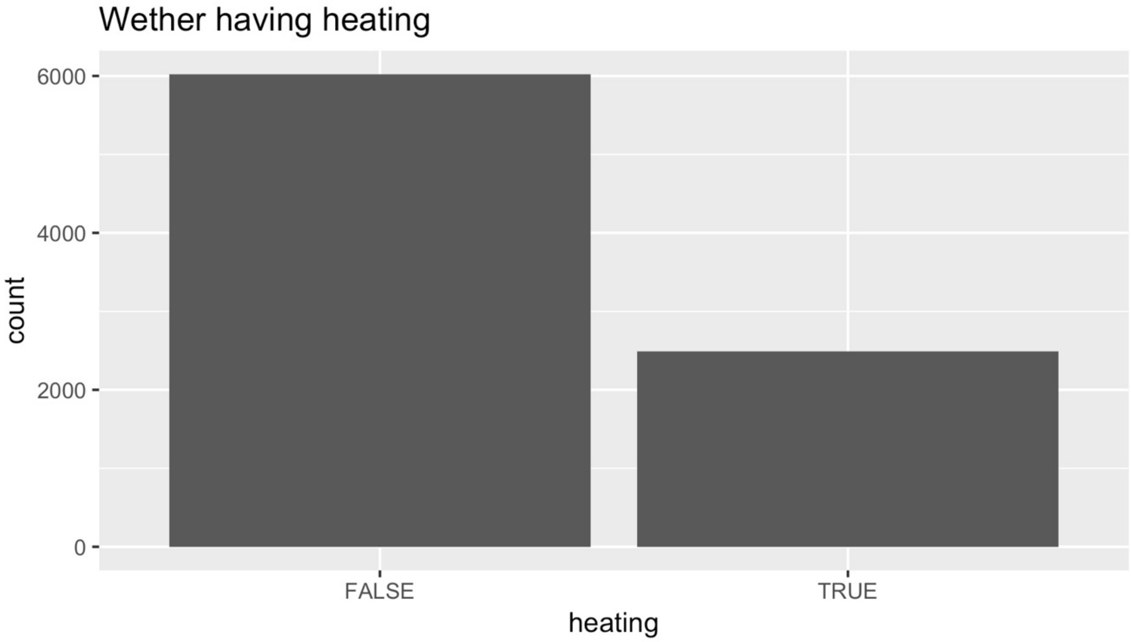
From this histogram, we can see that most pets' deposits are lower than $1000. In addition, there are only an extremely small percentage of apartments having pets' deposits over $2000. Also, the most common pets' deposits are around $500. In other words, a large percentage of the apartments have pets' deposits around $500.

Part VII: Heating and Air conditioning

Overall Strategy: I detect the word 'heater' in the text and set it as 'heater' in the data frame. Then, I detect word 'fireplace' and 'wood-burning stoves' in the text separately and combine them as 'fireplace in the data frame. For the texts that do not mention any of them, I make an assumption that those apartments do not have heating. The reason why I do this is that if an apartment has heating, the information is more likely to be written in the text since this can attract other people to rent the apartment.

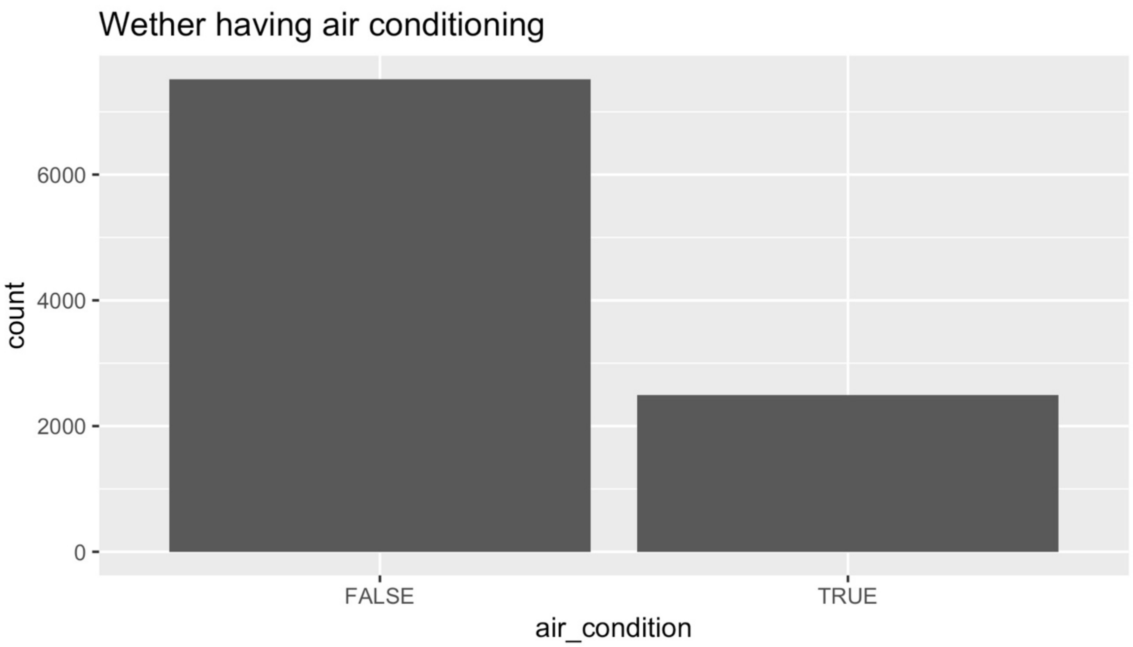
As for the air conditioning, it can be written as air conditioning or air-condition in the text, so I try to detect the word ‘air-condition’ or ‘air condition’ in the text. For the apartments do not mention air conditioning, I also make an assumption that they do not have the air conditioning and the reason is similar to that of heating.

The air conditioning is not more common than heating based on my assumption since the amount of heating is 10010 and the amount of air conditioning is 8514. Now, let’s take a look at a bar plot of whether apartments with air conditioning have heating.



In this bar plot, ‘FALSE’ means that the apartments with air conditioning do not have heating, ‘TRUE’ means that the apartments with air conditioning do have heating. We can easily see that the amount of ‘FALSE’ is larger than the amount of ‘TRUE’. Hence, the apartments with air conditioning do not typically have heating.

Now, let’s take a look at a bar plot of whether apartments with heating typically have air conditioning.

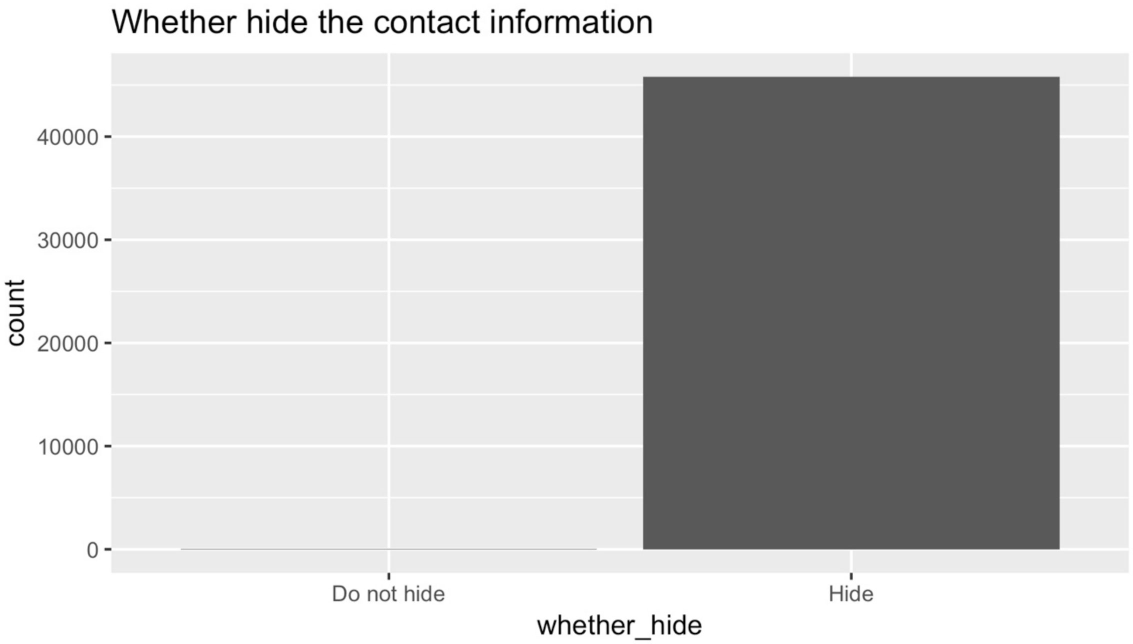


In this bar plot, ‘FALSE’ means that the apartments with heating do not have air conditioning, ‘TRUE’ means that the apartments with heating do have air conditioning. From the plot, it is easy to see that the amount of ‘FALSE’ is larger than the amount of ‘TRUE’. Therefore, we can conclude that the apartments with heating do not typically have air condition.

Part VIII: Do people use an optional feature to hide contact information?

Overall Strategy: Since phone always have a form ‘xxx-xxx-xxxx’, ‘(xxx)xxx-xxxx’ or ‘(xxx)xxx xxxx’, I detect this pattern in the text. Since email always have a form ‘xxx@xxx.xxx’ or ‘xxx@xxx.xx’, I detect this pattern in the text. If either of them can be detected, then they will be defined as ‘Do not hide’ in data frame; otherwise, they will be defined as ‘Hide’.

Now, the following is a bar plot of whether people hide their contact information.



From the bar plot, we can see that there are almost no people that do not hide the contact information. The majority of people hide their contact information. Actually, only 25 people do not hide their phone or e-mail. And the rest 45820 people hide their phone or e-mail.

Therefore, my conclusion is that nearly 100% of people use an optional feature to hide their email addresses and phone numbers from web scrapers like the one that scraped this data set.

Appendix

```
library(stringr)
library(ggplot2)
#1.Write a function read_post that reads
#a single Craigslist post from a text file.
#===========================================

#In each function to read title,text, price etc.
#I extract them by seeing which line they are located in
#since these information in all the post have the same
location.

#function to read title
read_title <- function(post){
  title <- post[1]
  return(title)
}

#function to read text
read_text <- function(post){
  text <- post[3:(length(post)-8)]
  text <- paste(text, collapse = "\n")#put them together so
that
  #they can in a single box of data frame
  return(text)
}

#function to read date
read_date <- function(post){
  date <- post[length(post)-6]
  date <- gsub('Date Posted|\\: ','',date)
  return(date)
}


#function to read price
read_price <- function(post){
  price <- post[length(post)-5]
```

```r
  price<-gsub('Price|\\: |\\$','',price)
  return(price)
}

#function to read latitude
read_latitude <- function(post){
  latitude <- post[length(post)-4]
  latitude <- gsub('Latitude|\\: ','',latitude)
  return(latitude)
}

#function to read longitude
read_longitude <- function(post){
  longitude <- post[length(post)-3]
  longitude <- gsub('Longitude|\\: ','',longitude)
  return(longitude)
}

#function to read bedrooms
read_bedrooms <- function(post){
  bedrooms <- post[length(post)-2]
  bedrooms <- gsub('Bedrooms|\\: ','',bedrooms)
  return(bedrooms)
}

#function to read bathrooms
read_bathrooms <- function(post){
  bathrooms <- post[length(post)-1]
  bathrooms <- gsub('Bathrooms|\\: ','',bathrooms)
  return(bathrooms)
}

#function to read sqft
read_sqft <- function(post){
  sqft <- post[length(post)]
  sqft <- gsub('Sqft|\\: ','',sqft)
  return(sqft)
}
```

```r
#function to read a single post
read_post <- function(file){
  post <- readLines(file)
  title <- read_title(post)
  text <- read_text(post)
  date <- read_date(post)
  price <- read_price(post)
  latitude <- read_latitude(post)
  longitude <- read_longitude(post)
  bedrooms <- read_bedrooms(post)
  bathrooms <- read_bathrooms(post)
  sqft <- read_sqft(post)

return(c(title,text,date,price,latitude,longitude,bedrooms,
bathrooms,sqft))
}
```

```r
#2.Write a function read_all_posts that uses read_post
(from Question 1) to read
#all information from all posts in a directory and return
them in a single data frame.
#=====================================================
read_all_posts <- function(directory){
  files = list.files(directory,full.names = TRUE,recursive =
TRUE)
  all_files = sapply(files,read_post)
  post_df <- data.frame(t(all_files))
  names(post_df)[1:9]<-
c('title','text','date','price','latitude','longitude','bedrooms','bathrooms','sqft')
  rownames(post_df)<-NULL
  return(post_df)
}

#read all the post
post = read_all_posts('messy')
```

```
#4.Extract the rental price from the title of each Craigslist post.
#===================================================

#extract price from the titles
title_price = str_extract(post$title,'\\$[0-9]+')
#remove the $ from the price
title_price = gsub('\\$','',title_price)
#put it into the data frame
post$title_price = title_price


#check the type of two variables
typeof(post$price)
typeof(post$title_price)
#change two variables into numerical.
post$price = as.character(post$price)
post$price = as.numeric(post$price)
post$title_price = as.numeric(post$title_price)
```

```r
#How do these prices compare to the user-specified prices?
#check the difference between between them
difference = post$price - post$title_price
difference = difference[!is.na(difference)]
table(difference == 0)#all the price are the same.
```

```r
#5.Extract the deposit amount from the text of each
Craigslist post.

#The reason why I use str_extract here is that I find
#that the deposit appears befor the pet deposit in most
#text. So we only need to extract the first deposit.

#Draw a sentence with Deposit or deposit.
#Since some text do not have period to split each sentence,
#I set a limit on that:
#in a long sentence without period,
#we only read 30 characters before deposit and 30
characters after deposit.
deposit_sentence =
str_extract(post$text,'[^[.-]]{0,30}[Dd]eposits?[^.]{0,30}'
)
tail(deposit_sentence)
head(deposit_sentence)


#remove the , from deposit, extract$and number from that,
#and then remove the $
```

```
deposit = gsub('\\,','',deposit_sentence)
deposit = str_extract(deposit,'\\$[0-9]+')
deposit = gsub('\\$','',deposit)
deposit = as.numeric(deposit)


#set the deposit to data frame
post$deposit = deposit
#check how many deposits do we get.
table(!is.na(deposit))

#Adjust the price like report 3.
#sometimes the values are not acctualy that large.
#There was just an issue in parsing the data
#If we read the posting for the highest priced apartment
#Then we see the price range is $3408-3742
#the price in this data is 34083742
post$price[post$price>=30000000] <- 3408
post$price[post$price==9951095] <- 995

#smaller than 100 price delete
post$price[post$price <= 100] <- NA

ggplot(post,aes(x = price,y = deposit)) + geom_point()+
  labs(title = 'Relation between deposit and price') +
  geom_density2d()
```

```
#6. Extract a categorical feature from each Craigslist post
# (any part) that measures whether the apartment allows
```

```
#pets: cats, dogs, both, or none.
#=================================================


#Find Cats in text
#Since there are many words like cation in the
#text, when detect cat, we should aviod i behind the cat.
cats = str_detect(post$text,'[Cc]ats?[^i]')

#detect the situation that cats are not allowed.
no_cats1 =
str_detect(post$text,'[Cc]ats?[^i][^.]{0,10}[Nn]ot[^.]{0,5}
[Aa]llowed')
no_cats2 =
str_detect(post$text,'[Nn]o[^.]{0,10}[Cc]ats?[^i]')
no_cats = no_cats1 | no_cats2

#The pattern for the cats
cats = cats&!no_cats

#Find Dogs in text, similar to cats
dogs = str_detect(post$text,'[Dd]ogs?')

no_dogs1 = str_detect(post$text,'[Dd]ogs?
[^.]{0,10}[Nn]ot[^.]{0,5}[Aa]llowed')
no_dogs2 = str_detect(post$text,'[Nn]o[^.]{0,10} [Dd]ogs?')
no_dogs = no_dogs1|no_dogs2
dogs = dogs&!no_dogs

#both allow cats and dogs
both = cats&dogs

#do not allow cats or dogs
none = no_cats&no_dogs

#put pets situation into data frame
post$pets = 'unknown'
post$pets[cats] = 'cats'
post$pets[dogs] = 'dogs'
post$pets[both] = 'both'
post$pets[none] = 'none'
```

```r
#Before finish the pets' policy problem, let's extract the
#post that mentioned pets from the original post. Since
some
#changes will be made on post$pets if we continue on pets'
policy
#To get pet deposit from the post, we can extract pet
deposit
#from the text that mention pets
pets_post <- subset(post,pets != 'none' & pets !=
'unknown')


#Now let's continue on
#whether the apartment allows pets: cats, dogs, both, or
none.
#For this problem, we have many unknown situation.
#Usually, If a post does not mention that, we assume
#the apartment allow both cats and dogs
post$pets[post$pets == 'unknown'] ='both'

table(post$pets)

#As for other kinds of pets, we can consider bird, fish and
rabbit
#since these kinds of pets are most common.

#The strategy here is similar with dogs and cats
other_pets =
str_detect(post$text,"([Bb]ird|[Ff]ish|[Rr]abbit)")
no_other_pets1 =
str_detect(post$text,'([Bb]ird|[Ff]ish|[Rr]abbit)[^.]*[Nn]o
t[^.]*[Aa]llowed')
no_other_pets2 = str_detect(post$text,'[Nn]o[^.]*[Dd]ogs?')
no_other_pets =  no_other_pets1|no_other_pets2
other_pets = other_pets&!no_other_pets
table(other_pets)#see how many apartments allow other pets
```

```
#Now, let's investigate in pets deposit, we should extract
#the pets deposit at first

#Situation1:$xxx pets xxx deposits
#Extract this pattern first.
pets_deposit_sentence = str_extract(pets_post$text,
                              '\\$[0-
9]+[^.]*[Pp]ets?[^.]*[Dd]eposits?')
#extract the $xxx from the sentence and remove$
pet_deposit1 = gsub('\\,','',pets_deposit_sentence)
pet_deposit1 = str_extract(pet_deposit1,'\\$[0-9]+')
pet_deposit1 = gsub('\\$','',pet_deposit1)
pet_deposit1 = as.numeric(pet_deposit1)
pet_deposit1 = pet_deposit1[!is.na(pet_deposit1)]

#find which texts have these patterns and put deposit
#into the data frame
pets_deposit_pattern1 = str_detect(pets_post$text,
                             '\\$[0-
9]+[^.]*[Pp]ets?[^.]*[Dd]eposits?')

pets_post$pets_deposit = 'none'
pets_post$pets_deposit[pets_deposit_pattern1] =
pet_deposit1




#Situation2: pets xxx$xxx deposits
#The way I do this is similar to situation1
pets_deposit_sentence2 = str_extract(pets_post$text,
                               '[Dd]eposits?[^.]*\\$[0-
9]+[^.]*[Pp]ets?')
pet_deposit2 = gsub('\\,','',pets_deposit_sentence2)
pet_deposit2 = str_extract(pet_deposit2,'\\$[0-9]+')
pet_deposit2 = gsub('\\$','',pet_deposit2)
pet_deposit2 = as.numeric(pet_deposit2)
pet_deposit2 = pet_deposit2[!is.na(pet_deposit2)]

pets_deposit_pattern2 = str_detect(pets_post$text,
                             '[Dd]eposits?[^.]*\\$[0-
9]+[^.]*[Pp]ets?')

pets_post$pets_deposit[pets_deposit_pattern2] =
pet_deposit2





#Situation3: pets xxx deposits xxx$xxx
#what I do here is similar to situation1
pets_deposit_sentence3 = str_extract(pets_post$text,

'[Pp]ets?[^.]*[Dd]eposits?[^.]*\\$[0-9]+')

pet_deposit3 = gsub('\\,','',pets_deposit_sentence3)
pet_deposit3 = str_extract(pet_deposit3,'\\$[0-9]+')
pet_deposit3 = gsub('\\$','',pet_deposit3)
pet_deposit3 = as.numeric(pet_deposit3)
pet_deposit3 = pet_deposit3[!is.na(pet_deposit3)]

pets_deposit_pattern3 = str_detect(pets_post$text,

'[Pp]ets?[^.]*[Dd]eposits?[^.]*\\$[0-9]+')

pets_post$pets_deposit[pets_deposit_pattern3] =
pet_deposit3


#Make a graphic that shows how pet deposits are distributed
#set none as NA
pets_post$pets_deposit[pets_post$pets_deposit == 'none'] <-
NA


#since the pet deposit usually will not
#be to high, for the pet deposit over $2500, maybe there
are
#some problems with that when we extract that. So we set
#pets deposit over $2500 as NA.
```

```
pets_post$pets_deposit = as.numeric(pets_post$pets_deposit)
pets_post$pets_deposit[pets_post$pets_deposit > 2500] <- NA
ggplot(pets_post,aes(x = pets_deposit)) + geom_histogram()
+
  labs(title = 'Distribution of pets deposit',x = 'pets
deposit')
```

```
#7.Extract a categorical feature from each Craigslist post
that measures
#whether each apartment has some kind of heating: a heater,
a fireplace
#(including wood-burning stoves), both, or neither of
these.
#================================================
#Detect heater from text
heater = str_detect(tolower(post$text),'heater')
#Detct fireplace from text
fireplace = str_detect(tolower(post$text),'fireplace')
wood_burning_stoves =
str_detect(tolower(post$text),'wood[\\- ]?burning stove')
fireplace = fireplace | wood_burning_stoves

both = heater&fireplace

#put the heating into the data frame
post$heating = 'none'
post$heating[heater] = 'heater'
post$heating[fireplace] = 'fireplace'
post$heating[both] = 'both'
#for the post that does not mention any of these, we assume
#that this apartment has neither of these
post$heating[post$heating == 'none'] = 'neither'
```

```
#Detect air conditioning in the post
#it can be air conditioning or air-condition.
air_conditioning =
str_detect(tolower(post$text),'air[\\- ]?condition')

#put air conditioning it the post
#Usually, if an apartment do not have air conditioner,
#they will not say in their post, so we can initilaize
#post$air_conditioner as none.
post$air_condition = 'none'
post$air_condition[air_conditioning] = 'TRUE'

#check the number of heating and air conditioning
table(post$heating != 'neither')
heater_amount = 10010
table(post$air_condition == 'TRUE')
air_condition_amount = 8514


#Do apartments with air conditioning typically have
heating?
post_air_conditioning = subset(post,air_condition ==
'TRUE')
post_air_conditioning$heating[post_air_conditioning$heating
!= 'neither'] <- 'TRUE'
post_air_conditioning$heating[post_air_conditioning$heating
== 'neither'] <- 'FALSE'
#Draw a barplot to reflect amount
ggplot(post_air_conditioning,aes(heating)) + geom_bar() +
  labs(title = 'Wether having heating')
table(post_air_conditioning$heating)


#Do apartments with heating typically have air
conditioning?
post_heating = subset(post,post$heating != 'neither')
post_heating$air_condition[post_heating$air_condition ==
'none'] <- 'FALSE'
```

```
#Draw a barplot to reflect amount
ggplot(post_heating,aes(air_condition)) + geom_bar() +
  labs(title = 'Wether having air conditioning')
```

```
#8. Craigslist has an optional feature to hide email
addresses
#and phone numbers from web scrapers like the one that
scraped this data set.
#==========================================================

#phone has the form xxx-xxx-xxxx or (xxx) xxx-xxx etc.
phone_number = str_detect(post$text,'[(]?[0-
9]{3}[)]?[- ]?[0-9]{3}[- ][0-9]{4}')
#email has the form xxx@xxxx.xxx.
email_addresses =
str_detect(post$text,'[^ ]+\\@[^ ]*?\\.[A-z]{2,3}')

do_not_hide = phone_number | email_addresses

#Put the situation into the data frame
post$whether_hide = 'Hide'
post$whether_hide[do_not_hide] = 'Do not hide'

table(post$whether_hide)#only 25 do not hide
#draw a bar plot to reflect the ammount
ggplot(post,aes(whether_hide)) + geom_bar() +
  labs(title = 'Whether hide the contact information')
```