Man Pan

Student ID: 914656278

Email: manpan@ucdavis.edu

1. Description

The report mainly talks about text mining. Firstly, I extract first line of each text as the title column in my final data set, then I extract the last 7 lines of each text as the 'data post', 'price', 'latitude', 'longitude','bedrooms','bathrooms' and 'size' columns in my final data set. Finally, I extract all remain text of each text as 'text' column in my final data set. In the following question, I can extract target information from 'title' or 'text' column again for future analysis, like title price, pet deposit, pet policy, apartment deposit, heat, ari conditioning, hide feature.....

2. Rental Price from the Title of Each Post and the User-Specified Prices

After extracting the rental price from the title of each Craigslist post, I find there are 180 titles don't have prices. And those 180 posts also have NA in the user-specified price. Then, I extract rows that rental price from the title of each post don't match the user-specified prices, I find there are 30 posts that have different expression forms for zero price, such as 0, 00, 000. It is abnormal that apartments price are zero, I check them in 'text' column, the price are not zero or absent. Thus, I think zero price apartments maybe wrong information. And I want mention, I find in some text, the price in text is not the same as the price in the bottom of the text.

3. Relationship between Rental Price and Deposit Amount

In order to get deposit amount of each post, I extract the sentence that has 'deposit' firstly, then I extract all price has '$' prefix from the sentence. I find there are some key deposit or pet deposit occur in the sentence, so I choose the maximum one as the Apartments deposit. Also, I find in some text, deposit information does not present as money, it says deposit is one month rent, or no deposit/ zero deposit. As for one month rent deposit, I should use the rent as the deposit; as for zero deposit, I should set deposit as 0. Finally, I combine the deposit from the two strategies. If I get two different deposit amounts from two strategies, I choose the deposit from the second strategy. If the deposit is smaller than $95 and larger than $0, I use NA to replace them. That's my idea to get deposit amount from posts. Form this, I get 13913 posts that have deposit amount.

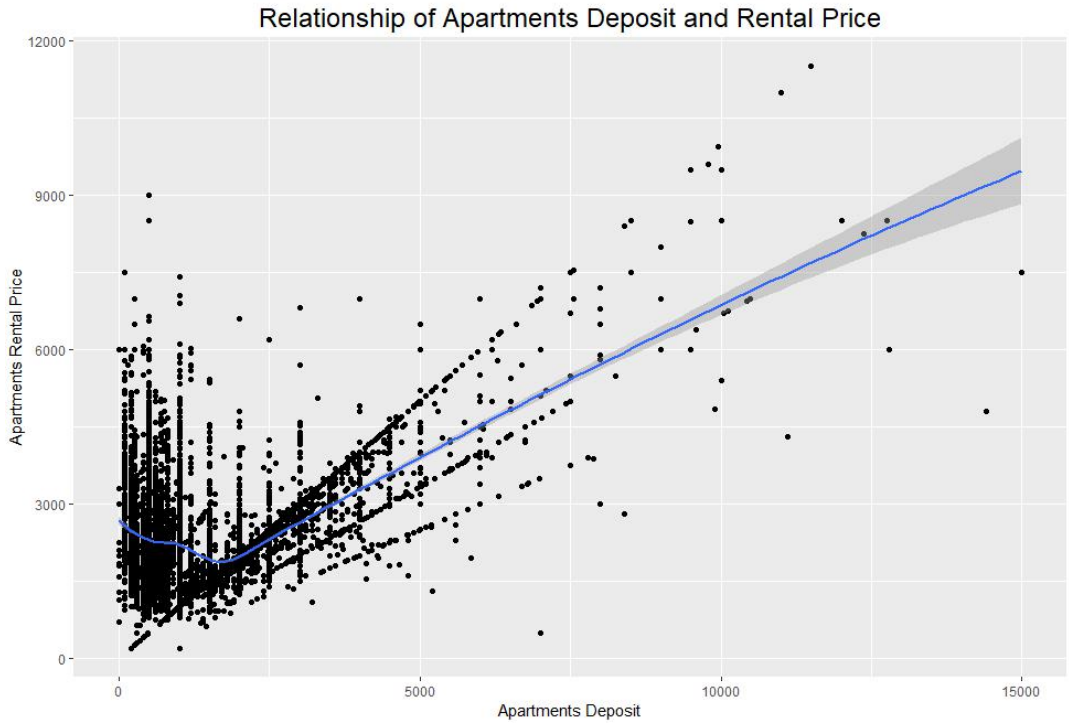

Figure 1

I plot scatter plot of deposit amount and rental price, after dealing with the outliers and errors in the two variables. I find some deposit that are larger than $15000 are not normal. Because the rent is not high when it has a so high deposit. I delete those abnormal deposit before analyzing the relationship. As the figure 1 shows: when the deposit is larger than $2000, there are an obvious positively relationship between rental price and deposit. The higher rent, a higher deposit. But when the deposit are from $95 - $2000, there are a negative relationship. Thus, I think, the relationship would likely change to different deposit amount.

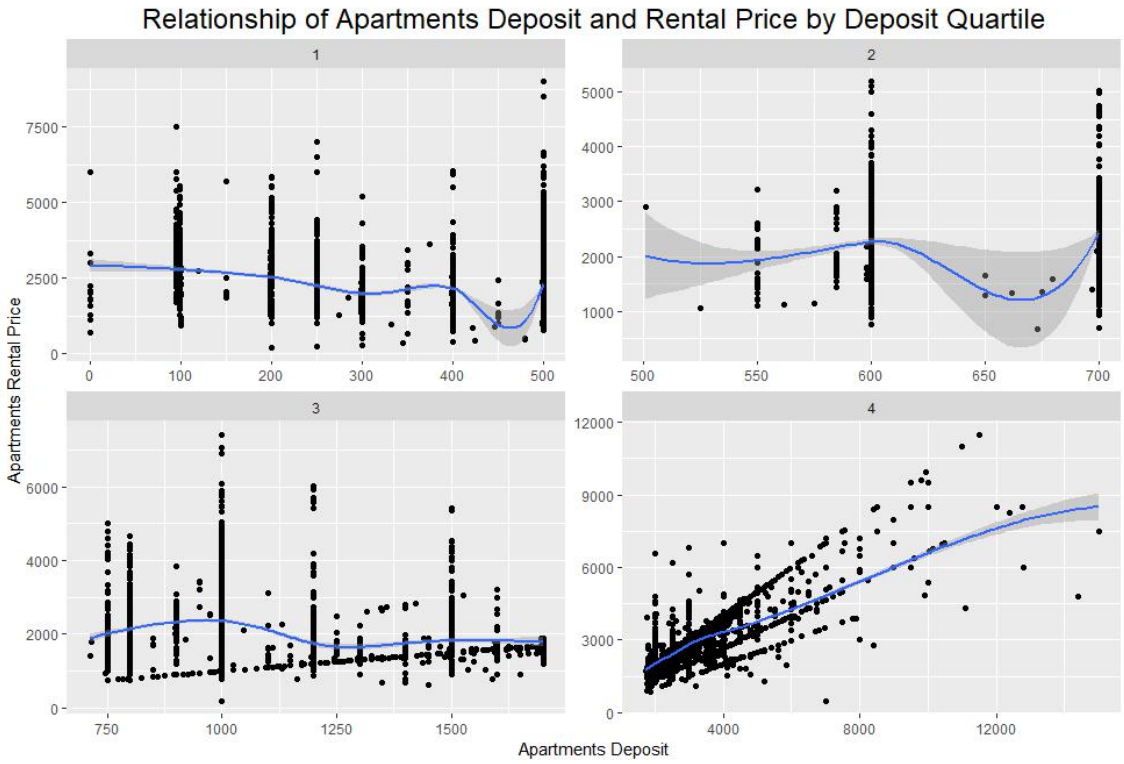

Figure 2

I split posts to 4 groups by deposit quartile. The 1st group contains deposit from $0 to $500, 2nd group contains deposit from $501 to $700, 3rd group contains deposit from $701 to $1700, 4th group contains deposit from $1701 to $15000. As the figure 2 above shows: in 4th group there are an obvious positive relationship. But for 1st, 2nd, 3rd group, no obvious relationship occurs. Deposits are always concentrated on hundreds or half hundreds, and price are scattered. And in 2nd group, deposits are concentrated on $600 and $700, price are also scattered. When deposits are between $600 and $700, price are relatively lower compared to other apartments in that group. Thus, I can conclude there is a positive relationship between rental price and deposit amount in 4th quartile group.

4. Pet policy in Apartments

Some apartments don't allow pets, some apartments allow dogs only, some allow cats only, others allow cats and dogs both. Thus, pet policy is a categorical feature: both, cat, dog, none. In order to get pet policy information, I combine text and title (two columns) firstly. Then, I category each group by locating different key words below.

Category	Key words
Both	Cats and dogs/ dogs and cats welcome/allowed/accepted/friendly/ok/are ok/okay/accepts/only, pets/animal welcome/allowed/accepted/friendly/ok/are ok/generally accepted/rent/deposit/upon approval, welcome cats and dogs, cats and small dogs friendly, small caged animals….
Cat	Cats welcome/allowed/accepted/friendly/ok/are ok/accepts/deposit, no dogs, dogs not allowed
Dog	Dogs welcome/allowed/accepted/friendly/ok/are ok/accepts/deposit, no cats, cats not allowed
None	No pets, no animals, no dogs or cats allowed

Table 1

Then I get 16357 apartments allow dogs and cats both, 6310 apartments don't allow pets, 188 apartments allow dogs only, 1176 apartments allow cats only.

both	cats	dogs	none	NA
16357	1176	188	6310	21814

Table 2

Also, I find some apartments allow some other kinds of animals like: common domesticated household birds, hamsters, gerbils, rabbits, guinea pigs, chinchillas and aquarium/terrarium animals including fish, hermit crabs, turtles, frogs, and small lizards.

5. Pet Deposits

In order to get pet deposit, I extract the sentence that has 'pet deposit' key words firstly, then I extract the money which has '$' in the front of numbers. Considering that there are pet rent deposit in the sentence, I choose the largest value in that sentence as pet deposit. Finally, I delete the pet deposit larger than $1000, because values larger than $1000 must be an error (they are security deposit). From this, I get 1852 apartments that have pet deposit.

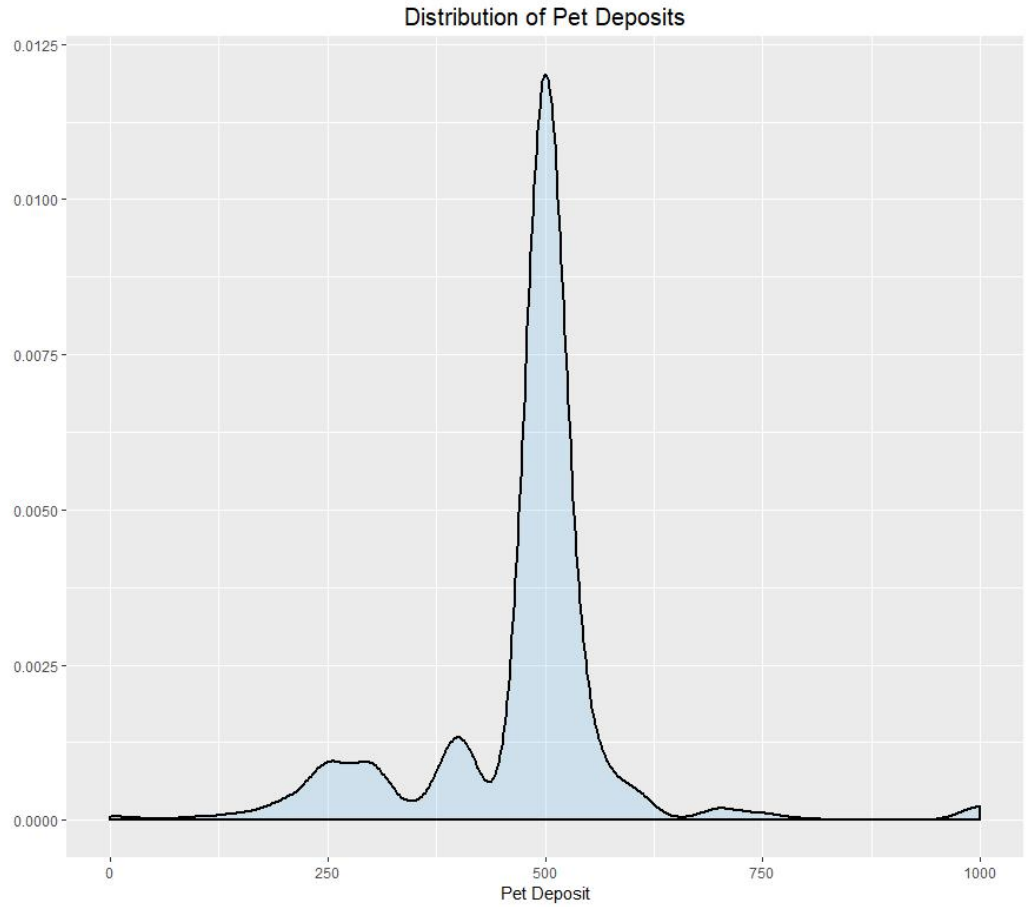

Figure 3

As the density plot of pet deposit shows: pet deposits are concentrated around $500, $250, $750, $1000. Most apartments that allow pets require $500 pet deposit.

6. Relationship between Apartment has Heating and Apartment has Air Conditioning

In order to find apartments have a heater, I locate key words: Heating, Heat and Heater in texts. Then, I use key words: fireplaces, stoves and burning to find texts that have a fireplace. I extract two kinds of heater separately. From two columns, I find that there are 2816 posts have heat and fireplace, there are 5044 posts that just have fireplace, there are 8906 posts that just have heat, there are 29079 posts do not have heat and fireplace either. Thus, 16766 posts have heating totally.

In order to find apartments have air conditioning, I locate key words: air conditioning, air, air conditioner, A/C. I find there are 14203 posts that have air conditioning.

Then, in the 16766 posts have heating, there are 9139 posts have air conditioning, 7627 posts don’t have air conditioning. They are 54.51% and 45.49% separately. And

in the 14203 posts have air conditioning, there are 1925 posts have heat and fireplaces both (13.55%), 5704 posts just have heat (40.16%), 1510 posts just have fireplaces (10.63%), 5064 posts don't have heat or fireplaces(35.65%). Thus, in posts have air conditioning, there are 64.35% posts have heat, 35.65% posts don't have heat.

For 14203 apartments that has air conditioning:

Heating	No Heating
9139 (64.35%)	5064 (35.65%)

Table 3

For 16766 apartments has heating:

Air Conditioning	No Air Conditioning
9139 (54.51%)	7627 (45.49%)

Table 4

Consequently, I think air conditioning isn't more common than heating. Apartments with air conditioning are more likely to have heat, while Apartments with heating don't have air conditioning typically.

7. Hide Email Addresses and Phone Numbers or Not?

After exploring texts in files, I find some texts have words "show contact info". From Craigslist, I find people use the hide feature if "show contact info" occurs in the text. So in the loop code, if texts have "show contact info", that means people use hide feature, then if texts don't have "show contact info" but have "email, text, phone, call or contact", that means people don't use hide feature, others mean no information about hide feature, indicates NA. By this strategy, I get 34832 posts have hide feature, 4140 posts don't have hide feature.

Then, I also find some texts have email addresses and phone number, that also means people don't use hide feature. After using regular expression, I get 73 emails and 99 phone numbers. Note: I extract phone number in two forms: (***) *** **** and ***-***-****.

Using hide feature	Not using hide feature	NA
34832	4140	6873

Table 5

As the table above shows: I get 34832 posts use hide feature, 4312 posts don't use hide feature. Delete the NA group, there are 89.38% posts using hide features, 10.62% posts not using hide feature. Thus, I conclude that most people would likely to use hide Email and Phone Number features.

8. Explanation of Two Main Function

For read_post function, I use one parameters：file path to control which file is loaded.

For read_all_posts function, I design 9 functions firstly, which extract columns information in my final data set. They are ‘title’, ‘text’, ‘latitude’, ‘longitude’, ‘data posted’, ‘price’, ‘sqft’, ‘bedrooms’, ‘bathrooms’ separately. ‘Title’ is the first row in text file; ‘latitude’, ‘longitude’, ‘data posted’, ‘price’, ‘sqft’, ‘bedrooms’, ‘bathrooms’ are the last 7 rows in the text file, other rows are all included in ‘text’ file. In order to use read_post function in read_all_post, it would be better to change the design of read_post function a little. I just use one parameter: text name to design that function, because I need extract text in each file separately. Also I can use list.files function to get all text names in one file, then use sapply function combined with read_post function to read all text files. After using read_posts and read_all_posts function, I get my data set that rows are observations of posts, columns are ‘title’, ‘text’, ‘latitude’, ‘longitude’, ‘data posted’, ‘price’, ‘sqft’, ‘bedrooms’, ‘bathrooms’ separately.

9. Limitation

By using key words to locate target information of sentence, I cannot extract all information I need, even extract some error items. Also, not comprehensive key words lead to uncertainty. For example, when I extract apartments deposit, I would get pet deposit, key deposit or pet rent sometimes. When I extract pets deposit, I would get security deposit or pets rent sometimes. It is hard to distinguish the price’s category. When I extract fireplace, because I locate the key word: stoves, I would get some gas stove. Also, the email and phone number are hard to extract, so I just limit the form of phone number/Email to improve the accuracy of text mining.

10. Conclusion

In 13913 posts that have deposit amount, I find that there is a positive relationship between rental price and deposit amount in 4^{th} quartile deposit group.

Apartments have a higher probability to allow pets, and they are prefer to cats, compared to dogs. Also, some apartments allow some other small and non-toxic animals. Pets deposit are concentrated on $500.

Apartments with air conditioning are more likely to have heat, while Apartments with heating don’t have air conditioning typically.

Most people would likely to use hide Email and Phone Number features.

Appendix

```
library(stringr)

library(stringi)

library(lubridate)

# Q1

read_post = function(file){ # file is the path of a text

   text = readLines(file, encoding = 'UTF-8')

   return(text)

}

# Q2

# in order to get a data frame contains all posts,

# I have to get all columns information from the posts firstly.

# in the txt, the first line is the title

title = function(txt){

   temp = txt[[1]] # get the first line

   return(temp)

}

# in the txt, the last line is the apartments size, I split the strings by " : "

size = function(txt){

   temp1 = txt[[length(txt)]] # get the last line

   size = sub(".*:", "", temp1) # split the strings by " : "

   size = str_replace_all(size, fixed(" "), "") # remove whitespace

   return(size)

}



# in the txt, the next to last line is the apartments bathroom, I split the strings by " : "

bathr = function(txt){

   temp1 = txt[[length(txt)-1]] # get the next to last line
```

```
    bathr = sub(".*:", "", temp1)
    bathr = str_replace_all(bathr, fixed(" "), "")
    return(bathr)
}

# in the txt, the third to last line is the apartments bedroom, I split the strings by " : "
bedr = function(txt){
    temp1 = txt[[length(txt)-2]] # get the third to last line
    bedr = sub(".*:", "", temp1)
    bedr = str_replace_all(bedr, fixed(" "), "")
    return(bedr)
}

# in the txt, the 4th to last line is the apartments longitude, I split the strings by " : "
longi = function(txt){
    temp1 = txt[[length(txt)-3]] # get the 4th to last line
    longi = sub(".*:", "", temp1)
    longi = str_replace_all(longi, fixed(" "), "")
    return(longi)
}

# in the txt, the 5th to last line is the apartments latitude, I split the strings by " : "
lati = function(txt){
    temp1 = txt[[length(txt)-4]] # get the 5th to last line
    lati = sub(".*:", "", temp1)
    lati = str_replace_all(lati, fixed(" "), "")
    return(lati)
}
```

```
# in the txt, the 6th to last line is the apartments bathroom, I split the strings by " : "
price = function(txt){
   temp1 = txt[[length(txt)-5]] # get the 6th to last line
   price = sub(".*:", "", temp1)
   price = str_replace_all(price, fixed(" "), "")
   price = strsplit(price,split='$', fixed=TRUE)[[1]][2] # remove "$"
   return(price)
}

# in the txt, the 7th to last line is the apartments date posts, I split the strings by " Date Posted: "
# note: there are two ":" in the string
Date_post = function(txt){
   temp1 = txt[[length(txt)-6]]
   Date_post = sub("Date Posted:*", "", temp1)
    Date_post= mdy_hm(Date_post)    # change form of time
   Date_post=str_extract(Date_post, "[^A-Z]+")
   return(Date_post)
}

# in the txt, the second line to the 8th to last line is the apartments all information
text = function(txt){
   temp1 = txt[2:(length(txt)-7)] # get the list from 2nd to 8th to last line
   text = do.call(paste, c(as.list(temp1), sep = "\n")) # combind all lists
   return(text)
}

# combine all column functions
```

```
all_col = function(txt){
    title = title(txt)
    text = text(txt)
    latitude = lati(txt)
    longitude = longi(txt)
    date_posted = Date_post(txt)
    price = price(txt)
    sqft = size(txt)
    bedrooms = bedr(txt)
    bathrooms = bathr(txt)
    temp = c(title, text, latitude, longitude,
                date_posted, price, sqft, bedrooms, bathrooms)
    return(temp)
}

read_all_posts = function(file_name){
    files = list.files(paste('messy_cl/messy/',file_name,sep=""), full.names = TRUE)
    desc = sapply(files, read_post)
    all = sapply(desc, all_col)
    data_frame = as.data.frame(t(all))
    colnames(data_frame) = c('title', 'text', 'latitude', 'longitude',
                                        'date_posted', 'price', 'sqft', 'bedrooms', 'bathrooms')
    rownames(data_frame) = NULL
    # change Rtypes
    data_frame$title = as.character(data_frame$title)
    data_frame$text = as.character(data_frame$text)
    data_frame$latitude = as.numeric(as.character(data_frame$latitude))
    data_frame$longitude = as.numeric(as.character(data_frame$longitude))
```

```
  data_frame$date_posted = as.character(data_frame$date_posted)
  data_frame$price = as.numeric(as.character(data_frame$price))
  data_frame$sqft = as.numeric(as.character(data_frame$sqft))
  data_frame$bedrooms = as.numeric(as.character(data_frame$bedrooms))
  data_frame$bathrooms = as.numeric(as.character(data_frame$bathrooms))
  return(data_frame)
}

losangeles = read_all_posts('losangeles')
sacramento = read_all_posts('sacramento')
sandiego = read_all_posts('sandiego')
sfbay = read_all_posts('sfbay')
sfbay_eby = read_all_posts('sfbay_eby')
sfbay_nby = read_all_posts('sfbay_nby')
sfbay_pen = read_all_posts('sfbay_pen')
sfbay_sby = read_all_posts('sfbay_sby')
sfbay_sfc = read_all_posts('sfbay_sfc')

craigslist = rep('losangeles',nrow(losangeles))
losangeles = cbind(losangeles,craigslist)
craigslist = rep('sacramento',nrow(sacramento))
sacramento = cbind(sacramento,craigslist)
craigslist = rep('sandiego',nrow(sandiego))
sandiego = cbind(sandiego,craigslist)
craigslist = rep('sfbay',nrow(sfbay))
sfbay = cbind(sfbay,craigslist)
craigslist = rep('sfbay_eby',nrow(sfbay_eby))
sfbay_eby = cbind(sfbay_eby,craigslist)
```

```
craigslist = rep('sfbay_nby',nrow(sfbay_nby))

sfbay_nby = cbind(sfbay_nby,craigslist)

craigslist = rep('sfbay_pen',nrow(sfbay_pen))

sfbay_pen = cbind(sfbay_pen,craigslist)

craigslist = rep('sfbay_sby',nrow(sfbay_sby))

sfbay_sby = cbind(sfbay_sby,craigslist)

craigslist = rep('sfbay_sfc',nrow(sfbay_sfc))

sfbay_sfc = cbind(sfbay_sfc,craigslist)


# combine all sub- Craigslist    posts

final_data = rbind(losangeles, sacramento,sandiego,sfbay,sfbay_eby,

                           sfbay_nby,sfbay_pen,sfbay_sby,sfbay_sfc)


# Q4

# get apartments price from title

final_data$title_price = rep(0,nrow(final_data))


for (i in (1:nrow(final_data))){

   temp1 = strsplit(final_data$title[i],split='/', fixed=TRUE)[[1]][1] # get the string in the front of /

   temp1 = str_trim(temp1) # get rid of space

   temp2 = strsplit(temp1,split='$', fixed=TRUE)[[1]][2] # remove "$"

   temp2 = strsplit(temp2,split=' ', fixed=TRUE)[[1]][1] # get the first string

   final_data$title_price[i] = temp2

}

# get two columns from final_data

temp1 = final_data[,c('price','title_price')]

temp1 = na.omit(temp1)

not_equal = temp1[!temp1$price==temp1$title_price,]
```

```
# Q5

# get the posts that have deposit information

# p    = final_data[grep(pattern = "Deposit",final_data[,2]),]



# get the money started with $, choose the max as deposit

final_data$deposit_2 = rep(0,nrow(final_data))

for (i in (1:nrow(final_data))){

   temp1 = final_data$text[i]

   # extract lines have 'deposite' ,    ignore upper/lower character

   temp2 = str_remove_all(temp1, '\\.[0]+')

   temp2 = str_extract_all(temp2, regex('[\n.!-][^.\n!-]*deposit[^.\n!-]*[\n.!-]?',

                                                ignore_case = TRUE), simplify = T)

   temp3 = unlist(strsplit(temp2, '\\s'))

   temp4 = str_subset(temp3, "\\$[0-9]")

   temp5 = str_replace_all(temp4, fixed(","), "") # replace ,

   temp6 = str_extract(temp5, "\\$[0-9]+") # remove $ and -

   temp6 = str_extract(temp6,"[0-9]+")

   if (identical(temp6,character(0))) {

      temp6 = NA

   }

   else{temp6 = max(as.numeric(temp6), na.rm = T)}

   final_data$deposit_2[i] = temp6

}

# replace deposit from 1 to 95 to NA (those are pet deposit)

final_data$deposit_2[final_data$deposit_2 > 0 & final_data$deposit_2 < 95] = NA



# get no deposit = 0, one month deposit = rent price
```

```
final_data$deposit = rep(0,nrow(final_data))

for (i in (1:nrow(final_data))){

   temp1 = final_data$text[i]

   # extract lines have 'deposite' ,    ignore upper/lower character

   temp2 = str_extract_all(temp1, regex('[\n.!-][^.\n!-]*deposit[^.\n!-]*[\n.!-]?',

                                                ignore_case = TRUE), simplify = T)

   temp3 = str_match(temp2,'(one month rent|1 month rent|One Month Rent|1 Month Rent)')[,1][1]

   temp4 = str_match(temp2,'(no deposit|zero deposit|not deposit|Zero Deposit|Not Deposit|No Deposit)')[,1][1]



   if (!is.na(temp4) ) { temp5 = 0}

   else if (!is.na(temp3)){temp5 = final_data$price[i]}

   else{ temp5 = NA }

   final_data$deposit[i] = temp5

}



# combine two deposit columns

for (i in (1:nrow(final_data))){

   if (is.na(final_data$deposit[i])){final_data$deposit[i] = final_data$deposit_2[i]}

}



# correct some errors

final_data$deposit[23765] = 2900

final_data$deposit[23766] = 2995

final_data$deposit[37857] = 2350

final_data$deposit[20699] = 3000

final_data = subset( final_data, select = -c(deposit_2) )

saveRDS(final_data, "final_data.rds")
```

```
# relation of rental price and deposit

final_data = readRDS('final_data.rds')

# correct errors and limit outliers

final_data$price[final_data$price >= 30000000] = 3408

final_data$price[final_data$price >= 9900000] = 995

final_data$price[final_data$price < 200 ] = NA

final_data$price[final_data$price > 20000 ] = NA

final_data$deposit[final_data$deposit > 15000 ] = NA


library(ggplot2)

ggplot(final_data,aes(y = price)) + geom_boxplot()

# overal plot

data_1 = final_data[,c('price','deposit')]

data_1 = na.omit(data_1)

ggplot(data_1, aes(deposit,price)) + geom_point() + geom_smooth(method='loess') +

  labs(title="Relationship of Apartments Deposit and Rental Price",x="Apartments Deposit",y="Apartments Rental Price")+

  theme(plot.title=element_text(hjust=0.5,size = 18))


# plot by deposit quartile

data_1$deposit_quartile = rep(0, nrow(data_1))

data_1$deposit_quartile = as.integer(cut(data_1$deposit, quantile(data_1$deposit, probs=0:4/4), include.lowest=TRUE, na.rm = TRUE))

data_1$deposit_quartile = as.factor(data_1$deposit_quartile)

# 0   500   700  1700 15000


ggplot(data_1, aes(deposit,price)) + geom_point() + geom_smooth(method='loess') +

  labs(title="Relationship of Apartments Deposit and Rental Price by Deposit Quartile",x="Apartments Deposit",y="Apartments Rental Price")+
```

```
    theme(plot.title=element_text(hjust=0.5,size = 18)) + facet_wrap(~deposit_quartile, scales = "free")


# Q6 pets

data_3 = final_data[,c('title','text')]

# combine title and text

data_3$comb = rep(0,nrow(data_3))

for (i in (1:nrow(data_3))){

  data_3$comb[i] =paste(data_3$title[i],data_3$text[i], sep = "\n")

}


get_pets = function(comb){

  # extract both pet (cats and dogs)

  temp1_1 = str_detect(comb, regex('welcome (cats? (and|&) dogs?|dogs? (and|&) cats?)', ignore_case = T))

  temp1_2 = str_detect(comb, regex('(cats? (and|&) dogs?|dogs? (and|&) cats?)[ ]?(welcome|allowed|accepted|-friendly|friendly|ok|okay|are ok|only|accepts)', ignore_case = T))

  temp1_3 = str_detect(comb, regex('pets?[ ]?(welcome|allowed|accepted|-friendly|friendly|ok|okay|are ok|generally accepted|accepts|rent|deposit|upon approval)', ignore_case = T))

  temp1_4 = str_detect(comb, regex('(welcome|love|accept) (pets?|(cats? (and|&) dogs?|dogs? (and|&) cats?))', ignore_case = T))

  temp1_5 = str_detect(comb, regex('accept (cats? (and|&) dogs?|dogs? (and|&) cats?)', ignore_case = T))

  temp1_6 = str_detect(comb, regex('allow (cats? (and|&) dogs?|dogs? (and|&) cats?)', ignore_case = T))

  temp1_7 = str_detect(comb, regex('cats? (and|&) small dogs? friendly', ignore_case = T))

  temp1_8 = str_detect(comb, regex('small caged animals', ignore_case = T))

  # extract dogs only and cats only

  temp2_1 = str_detect(comb, regex('no dogs?|dogs? not allowed', ignore_case = T))

  temp2_2 = str_detect(comb, regex('dogs? (welcome|allowed|accepted|-friendly|friendly|ok|okay|are ok|only|accepts|deposit)', ignore_case = T))

  temp3_1 = str_detect(comb, regex('no cats?|cats? not allowed', ignore_case = T))

  temp3_2 = str_detect(comb, regex('cats? (welcome|allowed|accepted|-friendly|friendly|ok|okay|are ok|only|accepts|deposit)', ignore_case = T))
```

```
    # extract no pets
    temp4 = str_detect(comb, regex('no (pets?|animals?)|no (dogs? or cats?|cats? or dogs?) allowed', ignore_case
= T))

    if(temp4){temp5 = 'none'}
    else if(temp1_1){temp5 = 'both'}
    else if(temp1_2){temp5 = 'both'}
    else if(temp1_3){temp5 = 'both'}
    else if(temp1_4){temp5 = 'both'}
    else if(temp1_5){temp5 = 'both'}
    else if(temp1_6){temp5 = 'both'}
    else if(temp1_7){temp5 = 'both'}
    else if(temp1_8){temp5 = 'both'}

    else if(temp2_1 & temp3_1){temp5 = 'none'}
    else if(temp2_1 & temp3_2){temp5 = 'cat'}
    else if(temp2_2 & temp3_1){temp5 = 'dog'}
    else if(temp2_2 & temp3_2){temp5 = 'both'}

    else if(temp2_2){temp5 = 'dog'}
    else if(temp3_2){temp5 = 'cat'}

    else {temp5 = NA}
    return(temp5)
  }
  data_3$pet = rep(0,nrow(data_3))
  data_3$pet = sapply(data_3$comb, get_pets)
```

```
length(which(data_3$pet == 'both')) # 16357

length(which(data_3$pet == 'none')) # 6310

length(which(data_3$pet == 'dog')) # 188

length(which(data_3$pet == 'cat')) # 1176


###pet deposit

data_3$pet_deposit = rep(0,nrow(data_3))

for (i in (1:nrow(data_3))){

   temp1 = final_data$text[i]

   # extract lines have 'deposite' ,    ignore upper/lower character

   temp2 = str_remove_all(temp1, '\\.[0]+')

   temp2 = str_extract_all(temp2, regex('[\n.!-][^.\n!-]*(pets?|cats?|dogs?) deposit[^.\n!-]*[\n.!-]?',

                                                ignore_case = TRUE), simplify = T)

   temp3 = unlist(strsplit(temp2, '\\s'))

   temp4 = str_subset(temp3, "\\$[0-9]")

   temp5 = str_replace_all(temp4, fixed(","), "") # replace ,

   temp6 = str_extract(temp5, "\\$[0-9]+") # remove $ and -

   temp6 = str_extract(temp6,"[0-9]+")

   if (identical(temp6,character(0))) {

      temp6 = NA

   }

   else{temp6 = max(as.numeric(temp6), na.rm = T)}

   data_3$pet_deposit[i] = temp6

}


data_3$pet_deposit[data_3$pet_deposit > 1200] = NA

sum(!is.na(data_3$pet_deposit))    # 1852
```

```
library(ggplot2)
ggplot(data = data_3, aes(x=pet_deposit))+
  geom_density( alpha = 0.2,fill="#56B4E9",size = 0.8) +
  # Change the fill colour to differentiate it
  labs(title = "Distribution of Pet Deposits") + theme(plot.title = element_text(hjust = 0.5,size = 15))+
  labs(y="")+labs(x="Pet Deposit")


##### other pets
data_3$other_pet = rep(0,nrow(data_3))
for (i in (1:nrow(data_3))){
  temp1 = data_3$text[i]
  temp2 = str_extract_all(temp1, regex('[\n.!-][^.\n!-]*small caged animals[^.\n!-]*[\n.!-]?',
                                       ignore_case = TRUE), simplify = T)[1,]
  if (identical(temp2,character(0))) {
    temp3 = NA
  }
  else{temp3 = temp2}
  data_3$other_pet[i] = temp3
}
```

#Small caged animals that are allowed are common domesticated household birds, hamsters, gerbils, rabbits, guinea pigs,

#chinchillas and aquarium/terrarium animals including fish, hermit crabs, turtles, frogs, and small lizards.

#Animals such as birds of prey, iguanas, ferrets, snakes, rats, mice, insects, arachnids (including spiders and scorpions),

#livestock, or any other exotic animals are not allowed.

```
# Q7
# find air conditioning
final_data$Air_conditioning = rep(0,nrow(final_data))
```

```
for (i in (1:nrow(final_data))){
   temp1 = final_data$text[i]
   temp2 = str_detect(temp1, regex('air conditioning| A/C| air| air conditioner', ignore_case = TRUE))
   if (temp2 ) { temp3 = 'Yes'}
   else {temp3 = 'None'}
   final_data$Air_conditioning[i] = temp3
}

# find heating
data_2 = final_data[,c('text','Air_conditioning')]
data_2$Heat = rep(0,nrow(data_2))
for (i in (1:nrow(data_2))){
   temp1 = data_2$text[i]
   temp2 = str_detect(temp1, regex(' Heating| Heat| Heater', ignore_case = TRUE))
   if (temp2 ) { temp3 = 1}
   else {temp3 = 0}
   data_2$Heat[i] = temp3
}

# find fireplace
data_2$Fireplace = rep(0,nrow(data_2))
for (i in (1:nrow(data_2))){
   temp1 = data_2$text[i]
   temp2 = str_detect(temp1, regex('Fireplaces?| fire places?| burning', ignore_case = TRUE))
   if (temp2 ) { temp3 = 1}
   else {temp3 = 0}
   data_2$Fireplace[i] = temp3
}
```

```
length(which(data_2$Air_conditioning == 'Yes')) # air conditioner 14203

data_2$Heat_Fireplace = rep(0,nrow(data_2))

for (i in (1:nrow(data_2))){

   if(data_2$Heat[i] == 1 & data_2$Fireplace[i] == 1){data_2$Heat_Fireplace[i]='both'}

   else if (data_2$Heat[i] == 1 & data_2$Fireplace[i] == 0){data_2$Heat_Fireplace[i]='Heat'}

   else if (data_2$Heat[i] == 0 & data_2$Fireplace[i] == 1){data_2$Heat_Fireplace[i]='Fireplace'}

   else {data_2$Heat_Fireplace[i]='None'}

}

length(which(data_2$Heat_Fireplace == 'both')) # 2844

length(which(data_2$Heat_Fireplace == 'Fireplace')) # 4728

length(which(data_2$Heat_Fireplace == 'Heat')) # 8878

length(which(data_2$Heat_Fireplace == 'None'))    # 29395

# fireplace & heat    16450


fire_heat = data_2[data_2[,'Heat_Fireplace']%in%c('both','Fireplace','Heat'),] #16766

length(which(fire_heat$Air_conditioning == 'Yes')) # 9139


Air_con = data_2[data_2$Air_conditioning =='Yes',] # 14203

length(which(Air_con$Heat_Fireplace == 'both')) # 1925

length(which(Air_con$Heat_Fireplace == 'Fireplace')) # 1510

length(which(Air_con$Heat_Fireplace == 'Heat')) # 5704

length(which(Air_con$Heat_Fireplace == 'None'))    # 5064


# Q8

# find email

data_2$Email = rep(0,nrow(data_2))

get_email = function(txt){
```

```
  temp1 = str_extract(txt,"[A-z0-9-]+\\@[A-z0-9-]+\\.[A-z0-9-]{0,5}")
  if (is.na(temp1)){temp2 = NA}
  else {temp2 = 'Email'}
  return(temp2)
}
data_2$Email = rep(0,nrow(data_2))
data_2$Email = sapply(data_2$text,get_email)
length(which(data_2$Email == 'Email')) # 73

# find phone
# str_extract(temp1,"\\b1?-?\\(?[0-9]{3}\\)?-?[0-9]{3}-?[0-9]{4}\\b")
# extract phone number like (123) 456 6789
data_2$Phone = rep(0,nrow(data_2))
for (i in (1:nrow(data_2))){
  temp1 = data_2$text[i]
  # extract email
  temp2 = str_match(temp1, regex("\\((\\d{3})\\)\\s(\\d{3})", comments = TRUE))[,1]
  if (is.na(temp2)){data_2$Phone[i] = NA}
  else {data_2$Phone[i] = 'phone'}
}
length(which(data_2$Phone == 'phone')) # 68

#extract phone number like 123-456-7890
data_2$Phone2 = rep(0,nrow(data_2))
for (i in (1:nrow(data_2))){
  temp1 = data_2$text[i]
  # extract email
  temp2 = str_match(temp1, regex("(\\d{3})\\-(\\d{3})\\-(\\d{3,4})", comments = TRUE))[,1]
```

```
    if (is.na(temp2)){data_2$Phone2[i] = NA}
    else {data_2$Phone2[i] = 'phone'}
  }
  length(which(data_2$Phone2 == 'phone')) # 34
  # extract phone number 99

  # regex("\\(?       # optional opening parens
  #                   (\\d{3}) # area code
  #                   [)- ]?     # optional closing parens, dash, or space
  #                   (\\d{3}) # another three numbers
  #                   [ -]?      # optional space or dash
  #                   (\\d{4}) # four more numbers
  #                   ", comments = TRUE)

  # find hide information posts
  data_2$contact = rep(0,nrow(data_2))
  for (i in (1:nrow(data_2))){
    temp1 = data_2$text[i]
    temp2 = str_detect(temp1, regex('show contact info', ignore_case = TRUE))
    if (temp2 ) { data_2$contact[i] = 'hide'}
    else if(str_detect(temp1, regex(' email| text| phone| call| contact', ignore_case = TRUE))) {data_2$contact[i] =
'not hide'}
    else {data_2$contact[i] = NA}
  }
  length(which(data_2$contact == 'hide'))   #34832
  length(which(data_2$contact == 'not hide'))   # 4140

  # combine email , phone numbers
```

```
for (i in (1:nrow(data_2))){
   if (is.na(data_2$Phone[i])){data_2$Phone[i] = data_2$Phone2[i]}
}
for (i in (1:nrow(data_2))){
   if (is.na(data_2$Phone[i])){data_2$Phone[i] = data_2$Email[i]}
}
```