Exploratory data analysis part II:
Extracting features with regular expression

Xiaozhu Zhang

December 1, 2018

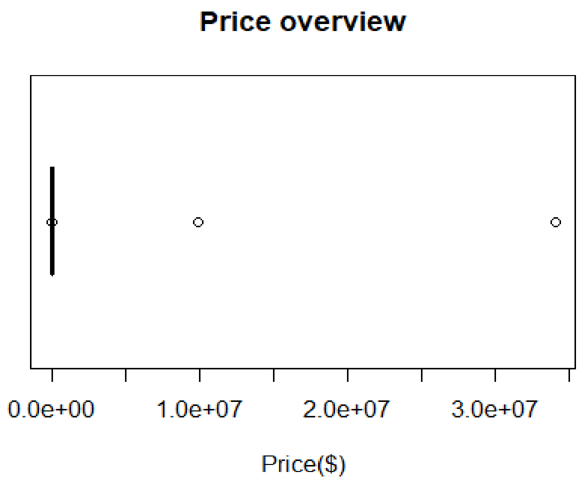
1. Extracting rental prices

Method:

By observing the title of each post in the title column, we can tell that each title seems to start with the prices of the advertising apartment(digits) followed by a single space.The method used here to obtain the prices is to chop the title into two part: digits of price,and rest of the title.

Result

- After comparing the prices in the title with the user specified prices with boolean expression "==", they seem to match each other very well(all true's).
- In addition,we should note that not all titles have prices. 180 of the post are without price attribute and the prices are also absent for these post in the title. The post that do not have a price attribute also do not have it in the title.The ones have prices attribute tend to have price in their title and they match.
- We should also note that according the boxplot below there are two unusually large observations of prices which turned out to be a price range but being put in the wrong format.



Extracting deposit amount

Method

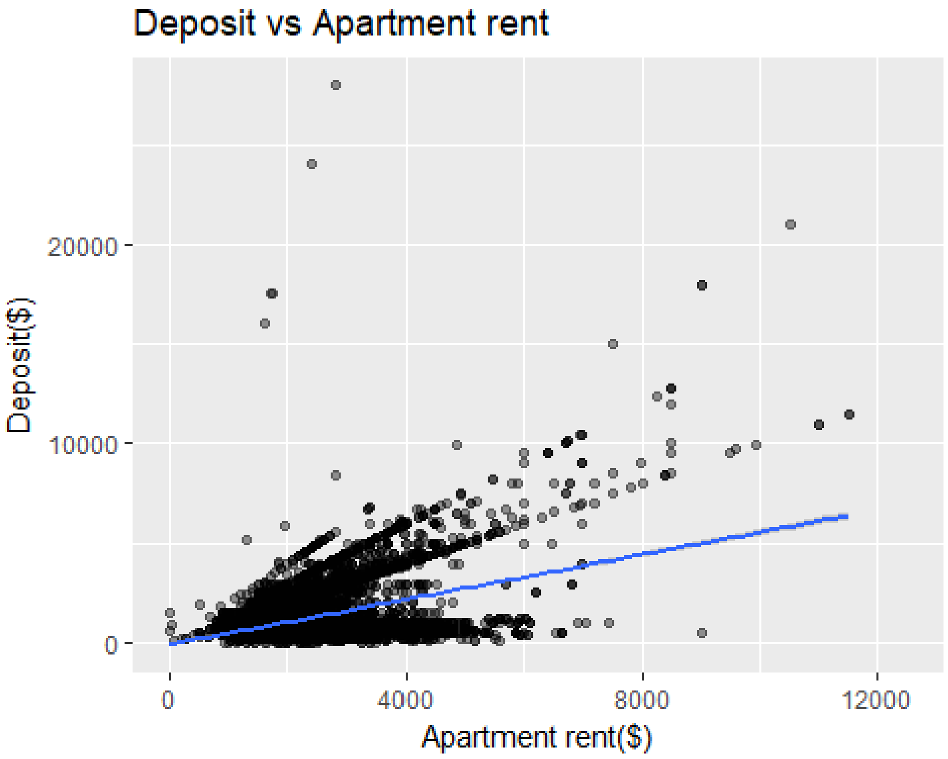
- For the purpose of easier(faster) handling, I first subset all the rows include the word "deposit".

By observing the post in a text editor, we can see that the posts follows the following patterns (not comprehensive, in decreasing priority)

- Pattern 1: Deposit:digits
- Pattern 2: digits deposit
- Pattern 3: digits security deposit
- Pattern 4: security deposit…digits

Result

Based on the scatterplot of deposit vs the monthly apartment rent, it seems that the observations follow a somewhat upward linear trend(as the blue linear line shows). By fitting a simple linear model, we get a coefficient of 0.563 which suggest that the deposit seem to be increasing as the prices increase.



Extracting pet policy

Method:

To extract a feature on pet policy, I first subset all the observations with no pets allowed:

- Pattern 1: no pet(s?)
- Pattern 2: pet policy:No,None,Not allowed

After excluding the apartments that do not allow pets, I subset all rows with pet information included, the following extractions of allowed pet type and deposit will focus on this subset.

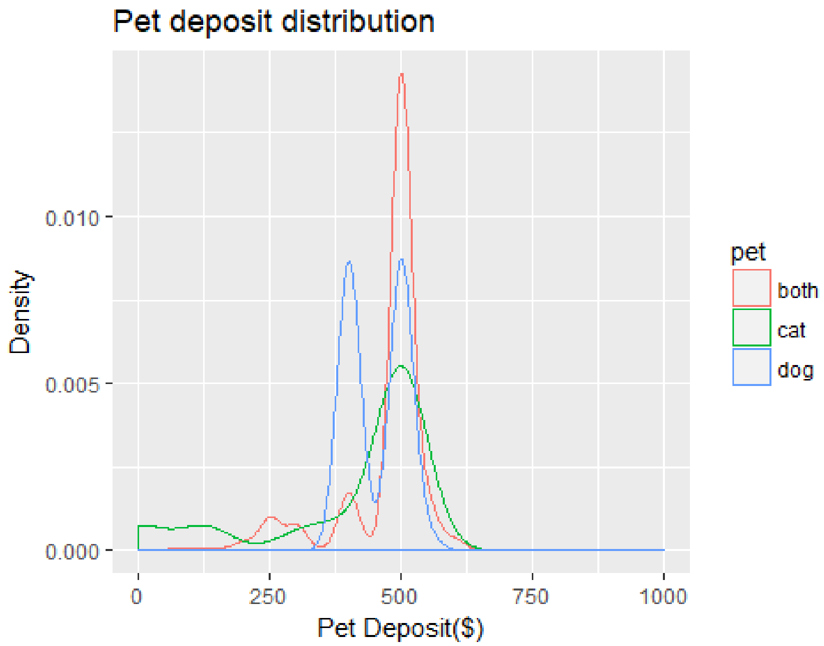
- The approach I took is first detect all rows contain “cat”, then detect all the rows contain dog.By combing the two logical expressions we get a data frame with two columns, afterwards, I added another column which is the sum of the existing two columns such that:2 indicates both, 1 in the first column and 0 in the second column indicates cat, 1 in the second column and 0 in the first column indicate dog.(by the fact that TRUE=1,FALSE=0)

To extract the pet deposit, the following patterns are used:

- additional or pet deposit: digits
- pet or additional deposit digits

Result

- In addition to cats and dogs, there are apartments that allow other kinds of pets. This assumption is made based on the observation that many apartments are claimed to be pet friendly but the person post the information do not specify which pets are/are not allowed. Furthermore, there are also apartment claim pet policy is negotiable.
- According to the distribution plot, we can see that the deposit for cat mostly center around $500 and the deposit for dog mostly center around $400 and $500.This suggests that $500 is a relatively common pet deposit regardless of the kind of pet and there is some chance that dogs have lower deposit than cats.



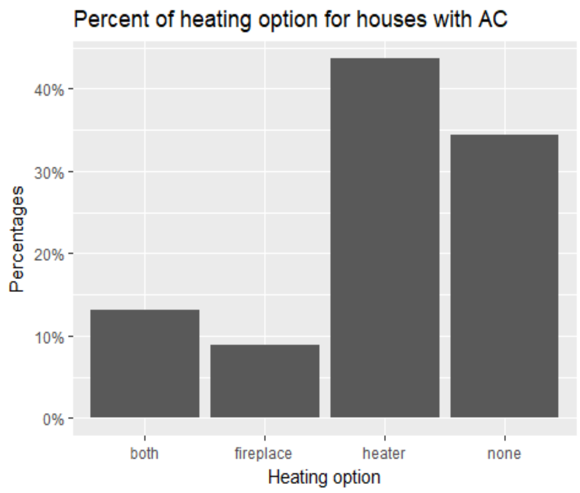
Extracting heating/cooling option

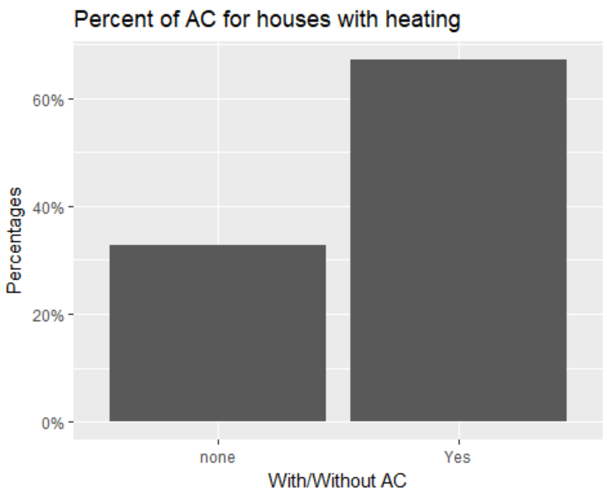
Method

- The first step I took to approach this question is to detect the presence of air conditioning. The patterns I found include:air(|-)condition(ing|er),AC,A/C,central air,thermostates,air.
- The patterns I use to detect heater option are: heater,heating, central heat, and the patterns I use to detect fireplace are fireplace, wood(-|)burning.
- Then I construct a data frame with three columns, the first column measures if the apartments have air conditioning(True of false), the second measures if there is heating(True or false), the third column is the sum of the first and the second column. 2 indicates both, 1 in the first column and 0 in the second indicates air conditioning, 0 in the first column and 1 in the second indicates heating.

Result

Based on the barplot, we can see that 67% of the houses with heating have air conditioning and that 66% of houses with air conditioning have heating. Since the percentages are very close, it seems that neither one of them is more common than another.





Extracting users' usage of a feature(hide email and phone number)

Method:

If the user uses this feature, then the text "show contact info" will appear in the text part of the post. Otherwise we won't be able to detect such text. We can calculate proportion of each situation to see how frequent this feature is used.

Result

Based on the proportion, we can see that over two third(majority) of the users uses this feature.

Used	Did not use
67%	33%

R APPENDEX

```r
#load library
library(ggplot2);library(scales);library(lubridate);library(stringr)
#Question 1

#Input: The read_post function takes the file path provided by the users
#Output: It reads a single post and return a single post as a single line.

read_post=function(file){
  post=readLines(file)
  post=str_c(post, collapse = "\n")
}
#Question 2

#Input: The read_all_post function takes the path of a directory
#Output: This function returns a data frame with 10 variables of all the post
 in the specified directory.

read_all_post=function(directory){
  full_paths=list.files(path=directory,full.names = TRUE)
  all_post=lapply(full_paths,read_post)
  all_post=unlist(all_post)
  result=as.data.frame(str_split_fixed(all_post, "\n",2))
  names(result)=c("title","remaining")
  result2 = as.data.frame(str_split_fixed(result$remaining, "\nDate Posted:",
 2))
  result$text=result2$V1
  result$remaining=result2$V2
  result$site=basename(directory)
  #get 7 attribute and turn into a data.frame
  attrs=data.frame(str_split_fixed(result$remaining,"\n",7))
  names_col=c("date_posted","price","latitude","longitude","bedrooms","bathro
om","sqft")
  names(attrs)=names_col
  attrs[,2:7]=lapply(attrs[,2:7],str_remove,"^[A-z ]+: ")
  attrs$price=str_remove(attrs$price,"\\$")
  result_final=cbind(result,attrs)
  result_final=result_final[,-2]

  #convert data type
  times=result_final$date_posted
  result_final$date_posted=mdy_hm(times)
  result_final$title=as.character(result_final$title)
  result_final$text=as.character(result_final$text)
  result_final[,5:10]=sapply(result_final[,5:10],as.numeric)
  result_final
}
```

```
#Apply the function read_all_post by creating another function:load_all_post
that reads all the posts in all directories.

load_all_post=function(data_location){
  all_directory=list.files(data_location,full.names = TRUE)
  total=lapply(all_directory,read_all_post)
}


##data_location="C:/Users/zxiaozhu/Desktop/messy_cl/messy"
##total=load_all_post(data_location)
##total_posts= do.call(rbind, total)
##saveRDS(total_posts, "not_messy_posts.rds")
#read the data
cl_data=readRDS("C:/Users/zhenguo/Desktop/STA141A/not_messy_posts.rds")

#Question4

#method
price_in_title=str_extract(cl_data$title,"\\$[0-9]+\\s")
price_in_title=str_remove_all(price_in_title,"\\$")
price_in_title=str_remove_all(price_in_title," ")
price_in_title=as.numeric(price_in_title)
is_equal=price_in_title==cl_data$price
boxplot(cl_data$price,main="Price overview",horizontal = T,xlab="Price($)")
# table(is_equal) #they all match
# table(is.na(cl_data$price)) #we see some post dont have a price attribute,
do they have it in the title?
no_price=cl_data[is.na(cl_data$price),]#the post lack price attribute also do
nt have price in the title
poss_range=cl_data[which(cl_data$price>100000),]#get those observations and t
hey are a range


#Question 5

#subset all rows contain the word "deposit"
deposit_str=str_detect(cl_data$text,regex("deposit",ignore_case = TRUE))
have_depo=cl_data[deposit_str,]

#Pattern 1
re1=regex("deposit[: ]*.[1-9,0-9]{2,}",ignore_case = TRUE)
m1=str_match(have_depo$text,re1)
m1=str_squish(str_remove_all(m1,"^.*\\$|^.*\ "))

#Pattern 2
re2=regex("[1-9]{1}(,?)[0-9]{2,}(?= deposit)",ignore_case = TRUE)
m2=str_squish(str_match(have_depo$text,re2)[,1])
result=cbind(m1,m2)
deposit=result[,1]
deposit[is.na(deposit)]=result[is.na(deposit),2]
```

```
#Pettern 3
re3=regex("[1-9]{1}(,?)[0-9]{2,}(?= security deposit)",ignore_case = TRUE)
m3=str_match(have_depo$text,re3)
result=cbind(result,m3[,1])
deposit[is.na(deposit)]=result[is.na(deposit),3]

#Pattern 4
re4=regex("(security deposit).*?[1-9]{1}(,?)[0-9]{2,}",ignore_case = TRUE)
m4=str_match(have_depo$text,re4)
m4=m4[,1]
m4=str_squish(str_remove_all(m4,"^.*\\$"))
result=cbind(result,m4)
deposit[is.na(deposit)]=result[is.na(deposit),4]
#clean up data and convert it into numerics
deposit=str_squish(str_remove_all(deposit,","))
deposit=as.numeric(deposit)
have_depo$deposit=deposit#add column
#delete the extreme observations(for proper scaling of graph)
have_depo$deposit[566:567]=NA
have_depo$price[2883]=NA
#add column to cl_data
cl_data$deposit=NA
cl_data[deposit_str,]$deposit=have_depo$deposit
#relationship?

ggplot(have_depo)+aes(price,deposit)+geom_point(alpha=0.4)+
  xlim(0,12500)+geom_smooth(method='lm')+
  ggtitle("Deposit vs Apartment rent")+
  xlab("Apartment rent($)")+ylab("Deposit($)")

model=lm(deposit~price,data=have_depo)
#model$coefficients


#Question 6

#I exclude all rows with no pets allowed
Reg1=regex("\\bno pet(s?)\\b|pet policy[:](\\bNo\\b|\\bNone\\b|\\bNot allowed
\\b)",ignore_case = T)
no_pet_in_text=str_detect(cl_data$text,Reg1)
cl_data2=cl_data[!no_pet_in_text,]#data with no-pets rows removed
no_pets=cl_data[no_pet_in_text,]#no pets allowed rows

no_pet_rows=as.numeric(rownames(no_pets))
cl_data$pet=NA
cl_data[no_pet_rows,]$pet="none"
```

```
#lOOK IN TITLE
#get all the rows with pet(s),dog(s),cat(s) in the title.
Reg2=regex("\\bpet(s?)\\b|\\bdog(s?)\\b|\\bcat(s?)\\b",ignore_case = T)
Mat1=str_detect(cl_data2$title,Reg2)
pets_in_title=cl_data2[Mat1,]

d=rownames(pets_in_title)
d=as.numeric(d)
cl_data3=cl_data2[-d,]#without pet_in_title to avoid redundency

#LOOK IN TEXT
Reg3=regex("\\bpet(s?) deposit\\b|\\bpet(s?) rent(s?)\\b|\\bcat(s?)\\b|\\bdog
(s?)\\b|\\bpet(s?)\\b|\\bpet(.)friendly\\b",ignore_case = TRUE)
Mat3=str_detect(cl_data3$text,Reg3)
pet_in_text=cl_data3[Mat3,]
pet_total=rbind(pet_in_text,pets_in_title)
cat1=regex("\\bcat(s?)\\b",ignore_case = T)
cat1_d=str_detect(pet_total$text,cat1)

dog1=regex("\\bdog(s?)\\b",ignore_case = T)
dog1_d=str_detect(pet_total$text,dog1)

result_pet=data.frame(cbind(cat1_d,dog1_d))
result_pet$total=NA
result_pet$total=result_pet$cat1_d+result_pet$dog1_d
both=result_pet$total==2

pet_total$pet=NA
pet_total[both,]$pet="both"
cat_only=result_pet$cat1_d==TRUE&result_pet$dog1_d==FALSE
dog_only=result_pet$cat1_d==FALSE&result_pet$dog1_d==TRUE
pet_total[cat_only,]$pet="cat"
pet_total[dog_only,]$pet="dog"

remaining3=pet_total[result_pet$total==0,]
Reg6=regex("\\bpet(s?)(.)(friendly|(.*)ok|allowed)\\b",ignore_case = T)
pet_friendly=str_detect(remaining3$text,Reg6)
pet_total[pet_friendly,]$pet="both"

pet_policy=pet_total$pet
pet_rows=as.vector(as.numeric(rownames(pet_total)))
cl_data$pet=NA
pets=as.data.frame(cl_data$pet)
pets[pet_rows,]=pet_policy
pets=pets[1:45845,]
cl_data$pet=pets
#extract pet deposit
reg_depo=regex("(additional|pet) deposit[: ] ?\\$[0-9]{2,}",ignore_case = TRU
E)
mat_pet1=str_match(pet_total$text,reg_depo)
pet_depo1=as.numeric(str_squish(str_remove_all(mat_pet1[,1],"^.*\\$")))

reg_depo2=regex("[.0-9]{2,}(?= (pet|additional) deposit)",ignore_case = TRUE)
mat_pet2=str_match(pet_total$text,reg_depo2)
pet_depo2=as.numeric(str_squish(mat_pet2[,1]))

result_pet_depo=cbind(pet_depo1,pet_depo2)
#tidy up the data and plot distribution

pet_depo1[is.na(pet_depo1)]=pet_depo2[is.na(pet_depo1)]
pet_total$pet_deposit=pet_depo1

pets_row=as.numeric(rownames(pet_total))
pet_total$pet=as.factor(cl_data[pets_row,]$pet)

#add column to cl_data
cl_data$pet_deposit=NA
deposit_rows=as.numeric(rownames(pet_total))
cl_data[deposit_rows,]$pet_deposit=pet_total$pet_deposit
cl_data=cl_data[1:45845,]

pet_total_no_na=pet_total[!is.na(pet_total$pet),]
ggplot(pet_total_no_na,aes(pet_deposit,color=pet))+geom_density()+
  ggtitle("Pet deposit distribution")+
  xlab("Pet Deposit($)")+ylab("Density")
# the deposite for cat and dog both center at 500. with some variations for d
ogs.


#Question 7

#extract AC
cl_data$air_condition="none"
ac=cl_data$air_condition

#search 1
AC_reg1=regex("\\bair( |-)condition(ing|er)\\b|\\bAC\\b|\\bA\\/C\\b|\\bcentra
l.*?air\\b|\\bthermostats\\b|\\bair\\b",ignore_case = TRUE)
AC1=str_detect(cl_data$text,AC_reg1)
ac[AC1]="Yes"

cl_data$air_condition=ac




#extract heating
cl_data$heating="none"
```

```
heat=cl_data$heating

#heting,heater, central heat
heatreg1=regex("\\bheat(er)?\\b|\\bheating\\b|\\bcentral heat\\b",ignore_case
 = TRUE)
heat_1_text=str_detect(cl_data$text,heatreg1)


#fireplace
heatreg2=regex("\\bfireplace\\b|\\bwood(-| )burning\\b",ignore_case = TRUE)
heat_2_text=str_detect(cl_data$text,heatreg2)

heat_result=cbind(heat_1_text,heat_2_text)
hr=data.frame(heat_result)
total=heat_1_text+heat_2_text
hr$total=total

heat_both=hr$total==2
cl_data[heat_both,]$heating="both"

heater=hr$heat_1_text==1&hr$heat_2_text==0
cl_data[heater,]$heating="heater"

fireplace=hr$heat_1_text==0&hr$heat_2_text==1
cl_data[fireplace,]$heating="fireplace"


#comparison
library(scales)

have_ac=cl_data[cl_data$air_condition=="Yes",]
ggplot(have_ac,aes(x=as.factor(heating),y=..prop..,group=1),stat='count')+
  geom_bar()+scale_y_continuous(labels = percent)+
  ggtitle("Percent of heating option for houses with AC")+
  xlab("Heating option")+ylab("Percentages")

tb2=table(have_ac$heating)
round(prop.table(tb2),digits = 2)
#66% of houses with Ac have heating


have_heating=cl_data[cl_data$heating!="none",]
ggplot(have_heating,aes(x=as.factor(air_condition),y=..prop..,group=1),stat='
count')+
  geom_bar()+scale_y_continuous(labels = percent)+
  ggtitle("Percent of AC for houses with heating")+
  xlab("With/Without AC")+ylab("Percentages")

tb1=table(have_heating$air_condition)
```

```
round(prop.table(tb1),digits = 2)

#67% of the houses wit heating have AC
#66% of houses with Ac have heating

#Question 8
feature_use=regex("\\bshow contact info\\b",ignore_case = TRUE)
feature_usage=str_detect(cl_data$text,feature_use)
tb=table(feature_usage)
round(prop.table(tb),digits = 2)
#most people seem to use this feature, "show contact info"
```