Mini-Project

BTMA 636

Tutorial Topic: Airbnb - From website scraping to visualization and analysis

Airbnb, an attractive renting business line, is gaining popularity among the people who own a property and want to rent it for short durations to earn money. This project is to help these people in quoting the right renting price for their property.

Secondly, the project would also be useful for the one who is planning for vacations in Canada and thinking about a stay at an Airbnb property. With this project, he/she can also guess whether a price for a property is fair or not.

In this tutorial, I will help them to find out what factors plays a vital role in predicting the renting price of Airbnb listings.

Firstly, I will scrape the data about properties listed on the www.airbnb.ca website. Then I will clean the data and bring it into the tabular format so that we can perform some visualization and analysis on that to predict the renting price for the property.

With the help of this project I will find the answers for the below mentioned 3 Questions:

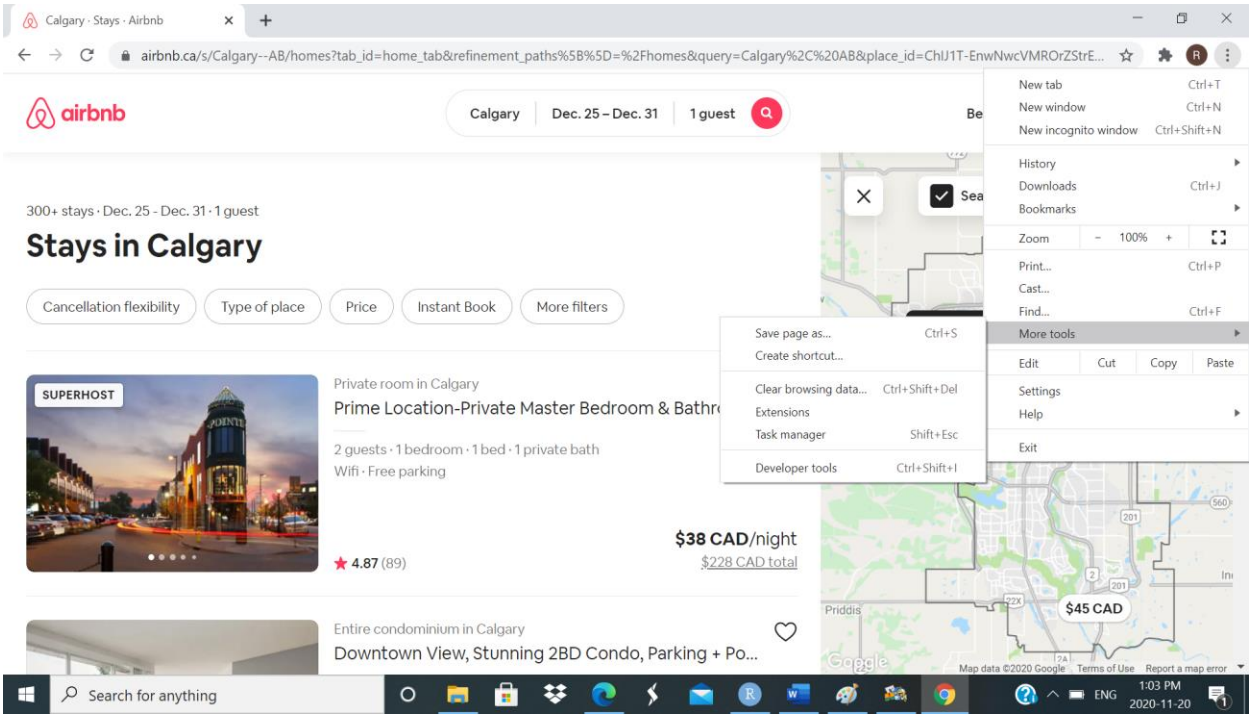
1. **What is the range of price in different cities of Canada?**
2. **What is the average price of each type of property (listed in Airbnb) in different cities?**
3. **What factors are significant in predicting the renting price of a property for Airbnb listing?**

Section-I

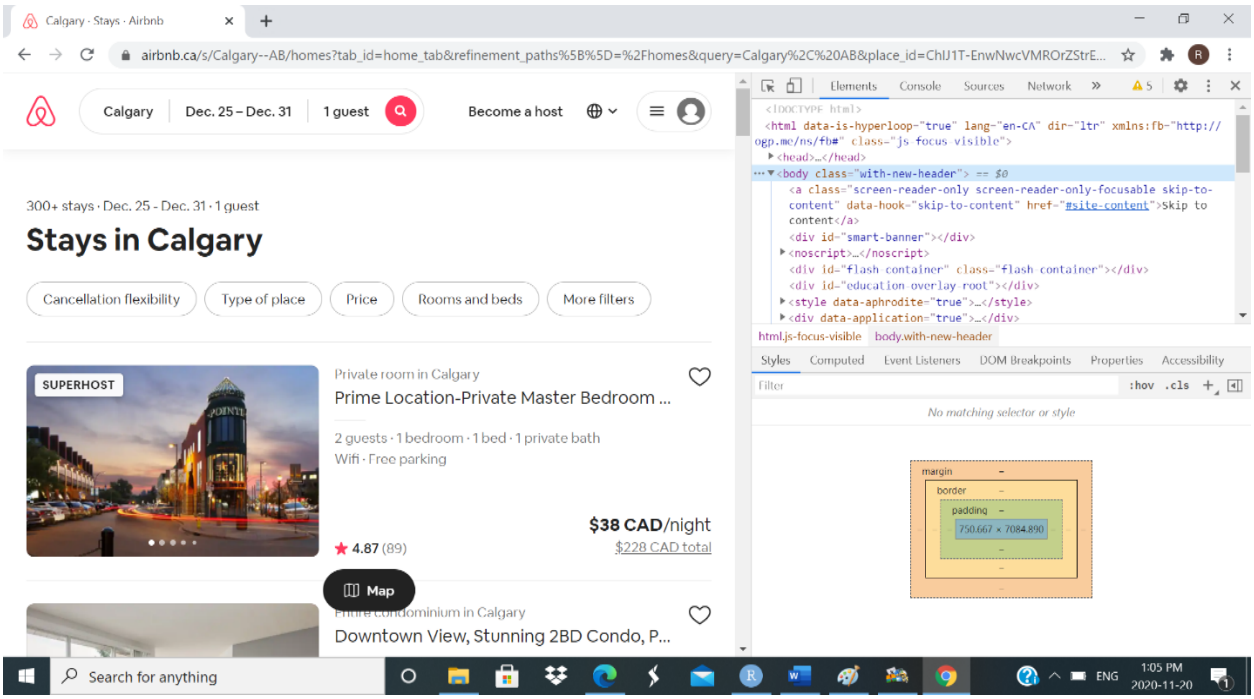
How to find out which class of data you need to scrape from a website

(Please Note: I am going according to google chrome browser settings)

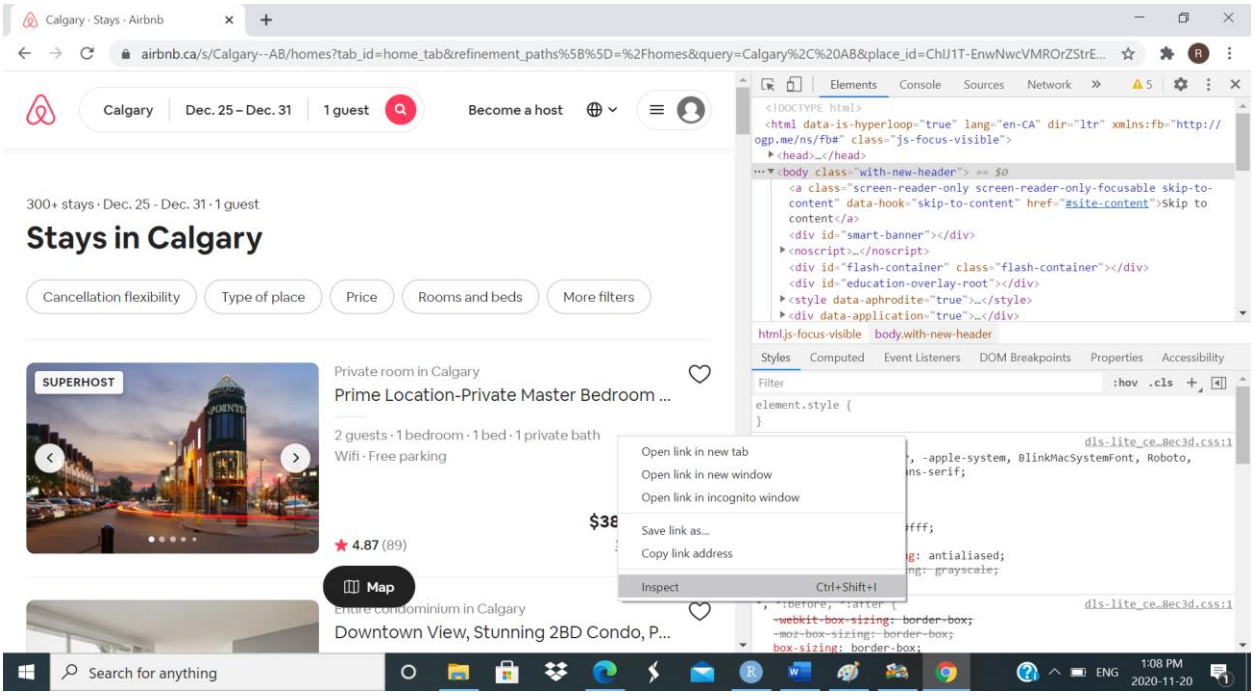
1. Go the website (www.airbnb.ca)
2. Input the location, check-in, check-out, guest information and click search
3. After reaching the listing page, click on the customize and control tab shown on the upper right corner of the browser with three vertical dots.
4. Select More tools and then select developer tools from the sub menu.



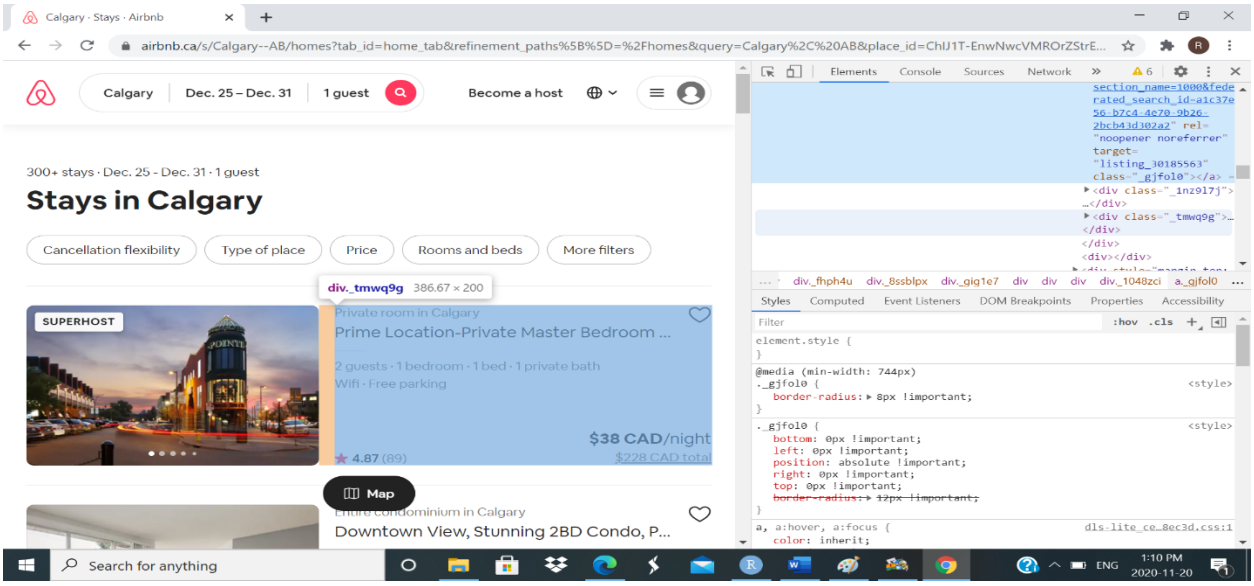
5. After clicking the developer tool, the screen will get divided into two sections, listing page on the left and the codes of the page on the right.



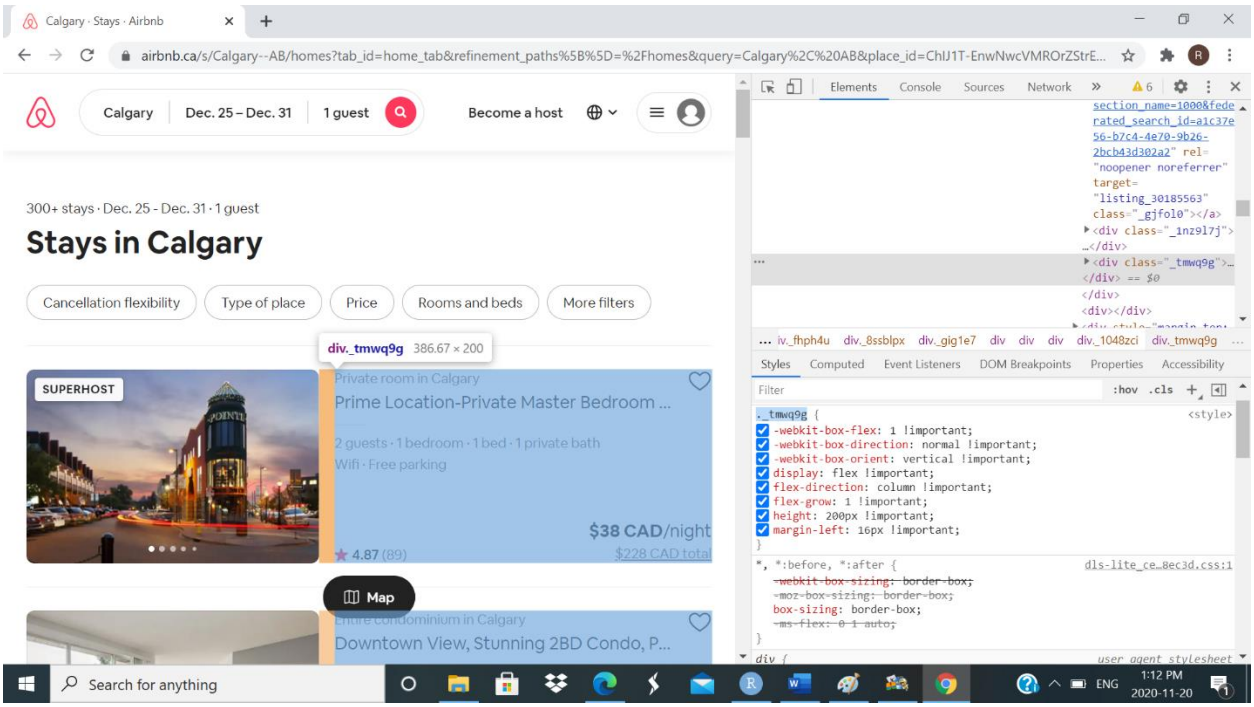
6. Now right click on the section you are interested in to scrape. And select inspect.



7. Now, hover the mouse up and down in the code window and find the class which contains the data you wants to scrape.



8. Now, copy the class name from there and paste it in your R-code at required location.



Section-II

Getting started with R codes

Step 1: Required R package

Before starting web scraping make sure that you have all the below mentioned packages in your R script. To install the package use install.packages("[package Name]").

```
library(robotstxt) # For checking website scraping permission
library(rvest) # For scraping the data
library(XML) # For scraping the data
library(stringr) # For cleaning the data
library(dplyr) # For filtering or reordering the data
library(ggplot2) # For creating charts
library(plotly) # For making the charts interactive
```

Step 2: Check for scraping permission

The function paths_allowed() is the function of robotstxt library. With the help of this function we can check for the permission of scraping the website. If we get "TRUE" in the console, means we can scrape the website as an anonymous scraper. If we get "FALSE", then for scraping the website you should provide some information on the website on the to the administrator of website.

paths_allowed("https://www.airbnb.ca/s/Calgary--AB/homes?tab_id=home_tab&refinement_paths%5B%5D=%2Fhomes&source=structured_search_input_header&search_type=filter_change&place_id=ChIJ1T-EnwNwcVMROrZStrE7bSY&checkin=2020-12-25&checkout=2020-12-31&adults=1")

Output in console

```
> paths_allowed("https://www.airbnb.ca/s/Calgary--AB/homes?tab_id=home_tab&refine
ment_paths%5B%5D=%2Fhomes&source=structured_search_input_header&search_type=filte
r_change&place_id=ChIJ1T-EnwNwcVMROrZStrE7bSY&checkin=2020-12-25&checkout=2020-12
-31&adults=1")
 www.airbnb.ca

[1] TRUE
>
```

Step 3: Spliting the Url

a) Go to the website www.airbnb.ca .
b) Fill the entries for "Location, Check in, Check out, guest" and click "search.
c) Go to page number 2 of property listing.
d) Copy the Url
e) Split the Url at the point highlighted below.

C_Url <- https://www.airbnb.ca/s/Calgary--AB/homes?tab_id=home_tab&refinement_paths%5B%5D=%2Fhomes&source=structured_search_input_header&search_type=pagination&checkin=2020-12-25&checkout=2020-12-31&adults=1&place_id=ChIJ1T-EnwNwcVMROrZStrE7bSY&federated_search_session_id=c8711d0b-d7a6-40e5-8ab7-510a7e454bf6&items_offset=

20

Last_Url <- "&section_offset=3"

Repeat Step three for all locations, for which you want to scrape.

Step 4: Web scraping

Create a vector for collecting the data. If you are scraping for more than one location create the vector for each location. (*Highlighted in pink*)

Since Airbnb lists 20 properties on each page and provides 15 pages listing for each location, the offset value “i” would go from 0 to 280 with a jump of 20 steps in for loop. (*Highlighted in skyblue*)

Create the url for each location using paste0() function as shown below. (*Highlighted in yellow*)

Use this url to scrape the data from www.Airbnb.ca website using the codes below. (*Highlighted in green*)

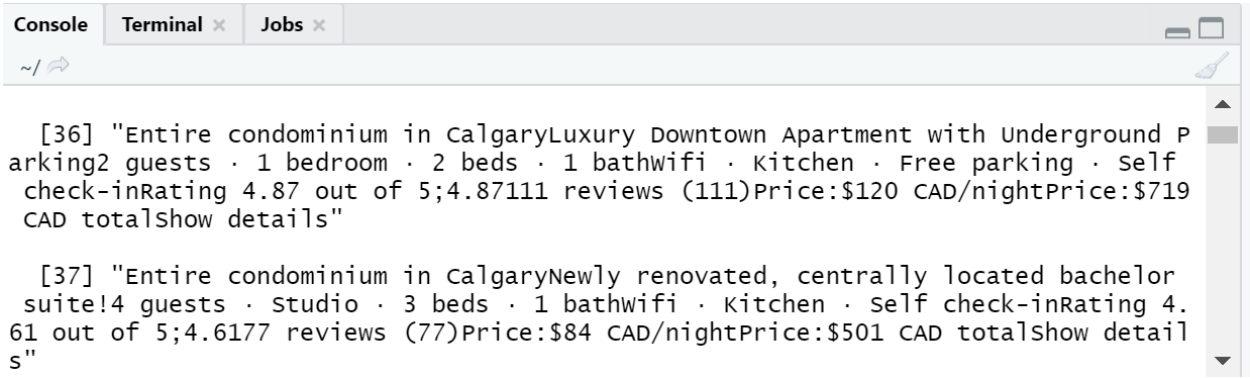
```
Calgary_Data <- vector()

for (i in seq(from=0,to=280,by=20))

{

  C_Url <- paste0(C_Url,i,Last_Url)

  Calgary_Data <- c(Calgary_Data,read_html(C_Url) %>%  html_nodes("._tmwq9g ") %>%  html_text())

}
```

Step 5: Combining the data

If you are scraping more than one location data, then merge the data into one vector as shown below. Otherwise, ignore these codes and move ahead.

```
Data<-vector()

Data <-
c(Calgary_Data,Toronto_Data,Vancouver_Data,Montreal_Data,Ottawa_Data,Quebec_Data,Saskatoon_Data)
```

Output in the console (of scraped data)



Step 6: Creating the vector for city

If you are scraping more than one location data, then create this vector to get the data about city in one vector as shown below. Otherwise, ignore these codes and move ahead.

In the line of codes below, I am considering that we have data for 7 cities (300 each) and based on that I am assigning city values to the vector.

```
# Creation of City vector
City <- character()
for(i in 1:2100)
{
  City[i]=case_when(
    i <= 300 ~ 'Calgary',
    i > 300 & i<=600 ~ 'Toronto',
    i > 600 & i<=900 ~ 'Vancouver',
    i > 900 & i<=1200  ~ 'Montreal',
    i > 1200 & i<=1500  ~ 'Ottawa',
    i > 1500 & i<=1800  ~ 'Quebec',
    i > 1800 & i<=2100  ~ 'Saskatoon' )
}
```

Step 7: Data cleaning

a) **Creating the Property Type vector from scraped data**

Using the line of codes shown below, I extracted the information of property type from the scraped data.

```
# Creation of Property_Type vector
A<-str_split(Data," ")
B<-data.frame(First=character(),Second=character(),Third=character())
P_T<-data.frame(Property_Type=character())
for(i in 1:2100)
{
  B[i,] <- A[[i]][1:3]
}
P_T <- paste(B$First,B$Second,B$Third,sep=" ")
Property_Type <- (str_remove_all(P_T," in"))
```

b) Creating the Rating vector from scraped data

Using the line of codes shown below, I extracted the information of Rating from the scraped data.

```
#Creating the Rating vector

R<-gsub(".*;", "", Data)

Rating<-as.numeric(substr(R,1,3))

Rating <- ifelse(is.na(Rating),2.5,Rating)
```

c) Creating the No_of_guests vector from scraped data

Using the line of codes shown below, I extracted the information of No_of_guests from the scraped data.

```
#Creating the No_of_guests vector

X<- str_extract(Data,'.*guest.')

No_of_guests <- sub("(\\d)[^0-9]+$", "\\1", X)

No_of_guests <- as.integer(substr(No_of_guests, nchar(No_of_guests), nchar(No_of_guests)))

No_of_guests <- ifelse(No_of_guests==0,10,No_of_guests)
```

d) Creating the Bedrooms vector from scraped data

Using the line of codes shown below, I extracted the information of Bedrooms from the scraped data.

```
#Creating the Bedrooms vector

Y<- str_extract(Data,'.*bedroom.')

Bedrooms <- sub("(\\d)[^0-9]+$", "\\1", Y)

Bedrooms <- as.integer(substr(Bedrooms, nchar(Bedrooms), nchar(Bedrooms)))

Bedrooms <- ifelse(is.na(Bedrooms),0,Bedrooms)
```

e) Creating the Beds vector from scraped data

Using the line of codes shown below, I extracted the information of Beds from the scraped data.

```
# Creating the Beds vector

Z<- str_extract(Data,'.*bed.')

Beds <- sub("(\\d)[^0-9]+$", "\\1", Z)

Beds <- as.integer(substr(Beds, nchar(Beds), nchar(Beds)))

Beds <- ifelse(is.na(Beds),1,Beds)
```

f) Creating the Baths vector from scraped data

Using the line of codes shown below, I extracted the information of Baths from the scraped data.

```
#Creating the Baths vector
W<- str_remove_all(str_remove_all(str_extract(Data,'.*bath.'),'shared'),'private')
W1 <- str_extract(W,'.{5}bath.')
Baths<-as.numeric(str_extract(W1, "\\d+\\.*\\d*"))
Baths<- ifelse(is.na(Baths),1,Baths)
```

g) Creating the Price vector from scraped data

Using the line of codes shown below, I extracted the information of Price from the scraped data.

```
#Creating the Price vector
V<- str_extract(Data,'.\\$...')
Price<-as.numeric(str_remove_all(str_extract(V, "\\d+"),"D"))
Price <- ifelse(is.na(Price),0,Price)
Price <- ifelse(Price==0,mean(Price),Price)
```

h) Creating the Wifi vector from scraped data

Using the line of codes shown below, I extracted the information of Wifi from the scraped data.

```
#Creating the Wifi vector
M<-str_detect(Data,"Wifi")
Wifi <- ifelse(M==TRUE,1,0)
```

i) Creating the Kitchen vector from scraped data

Using the line of codes shown below, I extracted the information of Kitchen from the scraped data.

```
#Creating the Kitchen vector
K<-str_detect(Data,"Kitchen")
Kitchen <- ifelse(K==TRUE,1,0)
```

j) Creating the Free_Parking vector from scraped data

Using the line of codes shown below, I extracted the information of Free parking from the scraped data.

```
#Creating the Free Parking vector
FP <-str_detect(Data,"Free parking")
```

```
Free_Parking <- ifelse(FP==TRUE,1,0)
```

k) Creating the Free_Cancelliation vector from scraped data

Using the line of codes shown below, I extracted the information of Free Cancellation from the scraped data.

```
#Creating the Free Cancellation vector

FC <-str_detect(Data,"Free cancellation")

Free_Cancellation <- ifelse(FC==TRUE,1,0)
```

l) Creating the Self_Check_In vector from scraped data

Using the line of codes shown below, I extracted the information of Self_Check_In from the scraped data.

```
#Creating the Self Check-in vector

SC <-str_detect(Data,"Self check-in")

Self_Check_In <- ifelse(SC==TRUE,1,0)
```

m) Creating the Heating vector from scraped data

Using the line of codes shown below, I extracted the information of Heating from the scraped data.

```
#Creating the Heating vector

H <-str_detect(Data,"Heating")

Heating <- ifelse(H==TRUE,1,0)
```

n) Creating the Washer vector from scraped data

Using the line of codes shown below, I extracted the information of Washer from the scraped data.

```
#Creating the Washer vector

Wa <-str_detect(Data,"Washer")

Washer <- ifelse(Wa==TRUE,1,0)
```

o) Creating the Dryer vector from scraped data

Using the line of codes shown below, I extracted the information of Dryer from the scraped data.

```
#Creating the Dryer vector

Dr <-str_detect(Data,"Dryer")

Dryer <- ifelse(Dr==TRUE,1,0)
```

p) Creating the Gym vector from scraped data

Using the line of codes shown below, I extracted the information of Gym from the scraped data.

```
#Creating the Gym vector

Gm <-str_detect(Data,"Gym")

Gym <- ifelse(Gm==TRUE,1,0)
```

q) Creating the Pool vector from scraped data

Using the line of codes shown below, I extracted the information of Pool from the scraped data.

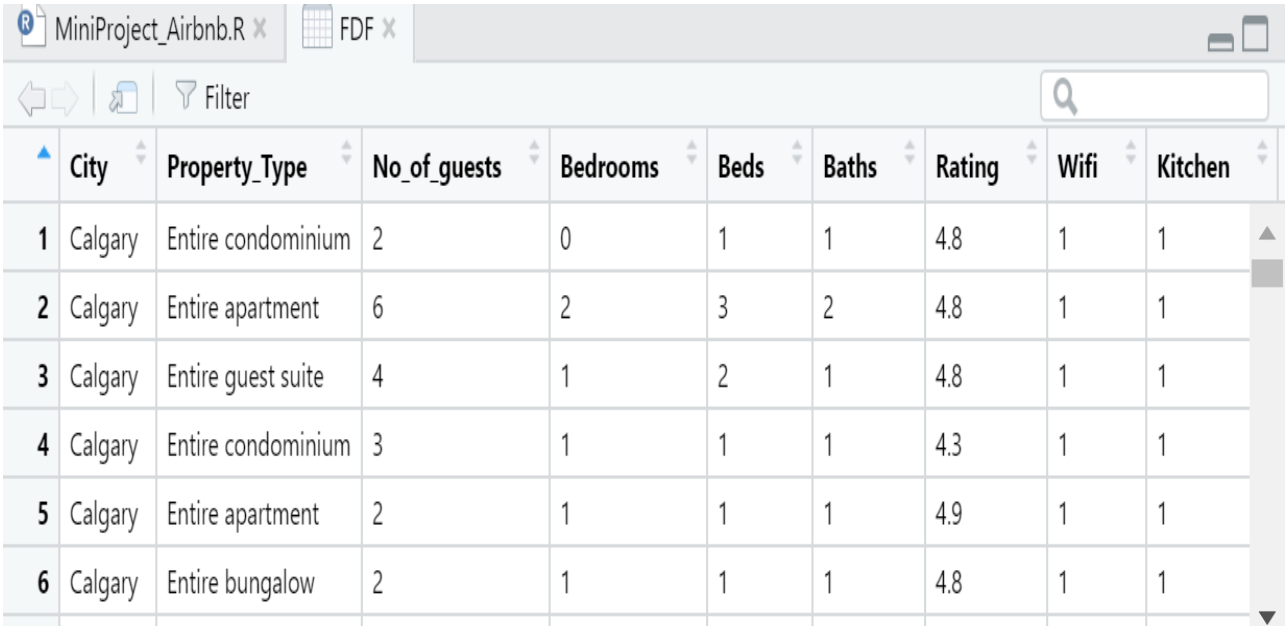
```
#Creating the Pool vector

Pl <-str_detect(Data,"Pool")

Pool <- ifelse(Pl==TRUE,1,0)
```

Step 8: Creating a data-frame

At this step, I combined all the vectors as columns and created a data-frame named FDF.

```
FDF <- data.frame(cbind(City,Property_Type,No_of_guests,Bedrooms,Beds,Baths, Rating,
Wifi,Kitchen,Free_Parking,Free_Cancellation,Self_Check_In,Heating,Washer,Dryer,Pool,Gym,Price))
```

Dataframe created as output at this point



	City	Property_Type	No_of_guests	Bedrooms	Beds	Baths	Rating	Wifi	Kitchen
1	Calgary	Entire condominium	2	0	1	1	4.8	1	1
2	Calgary	Entire apartment	6	2	3	2	4.8	1	1
3	Calgary	Entire guest suite	4	1	2	1	4.8	1	1
4	Calgary	Entire condominium	3	1	1	1	4.3	1	1
5	Calgary	Entire apartment	2	1	1	1	4.9	1	1
6	Calgary	Entire bungalow	2	1	1	1	4.8	1	1

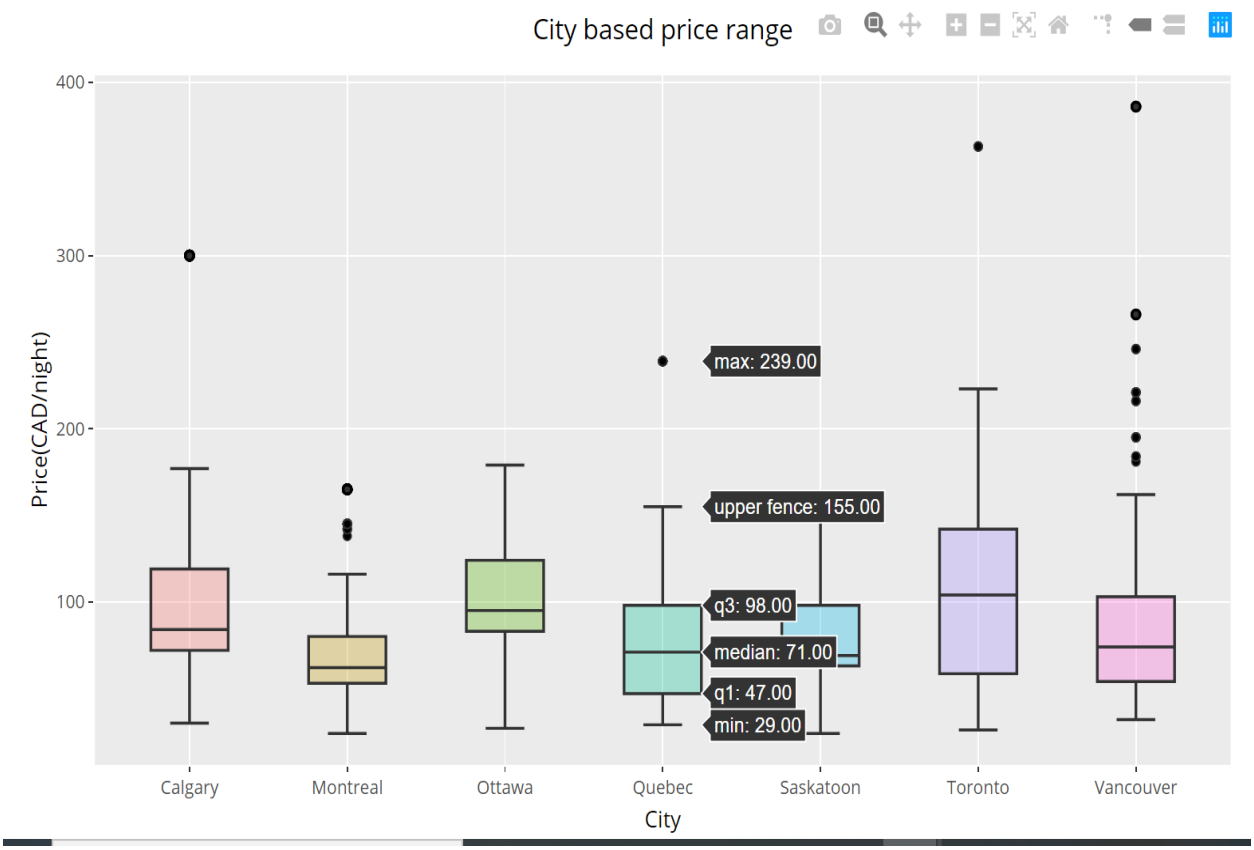
Step 9: Data Visualization

With the lines of code below, I generated the chart which answers my first question-

1. **What is the range of price in different cities of Canada?**

```
#Question : What is the price range in each city?

col_names <- c("Calgary", "Montreal", "Ottawa","Quebec","Saskatoon","Toronto","Vancouver")

Summary_chart <- FDF %>% group_by(City,Property_Type)%>% summarise(Price = as.numeric(Price) , .groups="drop") %>%  ggplot(aes(x=City, y=Price, fill=City)) +  geom_boxplot(alpha=0.3)  + scale_x_discrete(labels= col_names)+  xlab("City")+ylab("Price(CAD/night)")+ggtitle("City based price range")+ theme(plot.title = element_text(hjust = 0.5))+theme(legend.position="none")

ggplotly(Summary_chart)
```

With the help of the chart below we will get to know what the price range is for per night stay in Airbnb property in each city, if you are considering more than one city then you will get more boxplots like mine in your chart.

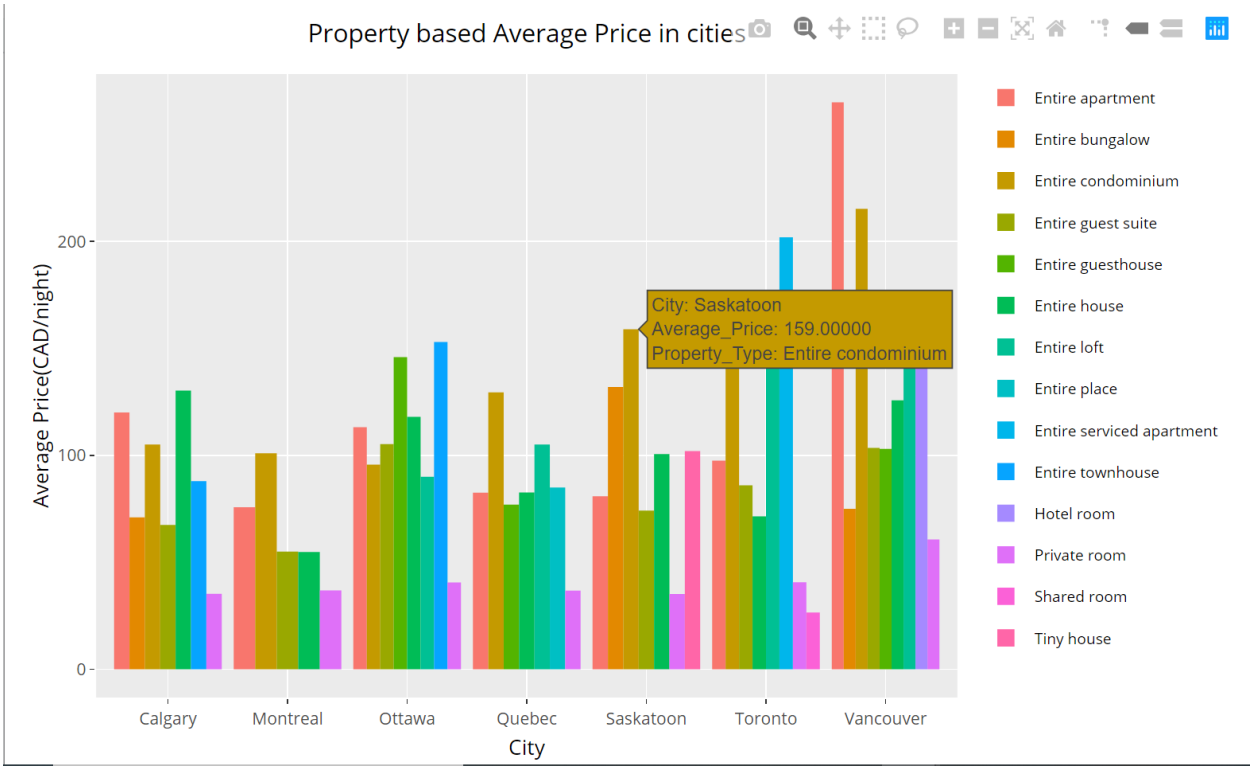


With the lines of code below, I generated the chart which answers my second question-

2. **What is the average price of each type of property (listed in Airbnb) in different cities?**

```
#Question : What is the average price of each type of property(listed in Airbnb) in different cities?

col_names <- c("Calgary", "Montreal", "Ottawa","Quebec","Saskatoon","Toronto","Vancouver")

FDF1 <- FDF %>% group_by(City,Property_Type)%>% summarise(Average_Price = mean(as.numeric(Price)),
.groups="drop")%>% ggplot( aes(x=City, y=Average_Price, fill=Property_Type))+

 geom_bar(stat="identity", position = "dodge")+scale_x_discrete(labels= col_names)+

 xlab("City")+ylab("Average Price(CAD/night)")+ggtitle("Property based Average Price in cities")+

 theme(legend.text=element_text(size=8),legend.margin = margin(1, 1, 1, 1))+ guides(fill =
guide_legend(title=NULL))+ theme(plot.title = element_text(hjust = 0.5))

 ggplotly(FDF1)
```

With the help of the chart below we will get to know what the price range is (for per night stay) in Airbnb property in each city, if you are considering more than one city your chart would look like mine.



Step 10: Data analysis

With the help of below mentioned codes, I did regression analysis on the data and figured out which variables are statistically significant in predicting the price of a property for Airbnb listing. This way we got the answer for our actual question with which we started doing the project.

3. **What factors are significant in predicting the renting price of a property for Airbnb listing?**

```
#Question: Which factors are significant in predicting the renting price of a property for Airbnb listing?

City<-as.factor(City)

Property_Type <- as.factor(Property_Type)

Reg_FDF <-
data.frame(cbind(City,Property_Type,No_of_guests,Bedrooms,Beds,Baths,Rating,Wifi,Kitchen,Free_Parking,Free_Cancellation,Self_Check_In,Heating,Washer,Dryer,Pool,Gym,Price))

fit <- lm(Price~.,Reg_FDF)

summary(fit)
```

The output we obtained with this code is shown below.

```
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)        -4.6624     7.7177  -0.604  0.54583
City                7.6645     0.5755  13.318  < 2e-16 ***
Property_Type      -4.3377     0.2415 -17.964  < 2e-16 ***
No_of_guests       16.2445     1.0800  15.041  < 2e-16 ***
Bedrooms           -7.6619     1.7827  -4.298 1.80e-05 ***
Beds               -3.8001     1.6048  -2.368  0.01798 *
Baths              11.8915     3.9324   3.024  0.00253 **
Rating              8.2715     1.0783   7.671 2.60e-14 ***
Wifi                    NA         NA      NA       NA
Kitchen             5.5721     3.1918   1.746  0.08100 .
Free_Parking       -0.2465     1.7908  -0.138  0.89054
Free_Cancellation  -0.7407     1.8559  -0.399  0.68986
Self_Check_In      25.1192     4.7896   5.245 1.73e-07 ***
Heating            -3.2578     3.0309  -1.075  0.28256
Washer              3.3681     2.6355   1.278  0.20140
Dryer                   NA         NA      NA       NA
Pool              -17.6192    16.0358  -1.099  0.27201
Gym                30.2068    10.0825   2.996  0.00277 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 36.9 on 2084 degrees of freedom
Multiple R-squared:  0.3602,    Adjusted R-squared:  0.3556
F-statistic: 78.23 on 15 and 2084 DF,  p-value: < 2.2e-16
```

In the above output you can see there are 2 variables with NA, two remove those two variables I used the codes shown below.

```
#Removing the NA columns
```

```
fit <- lm(Price~.-Dryer-Wifi,Reg_FDF)
```

```
summary(fit)
```

The final output is as shown below.

```
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)        -4.6624     7.7177  -0.604  0.54583
City                7.6645     0.5755  13.318  < 2e-16 ***
Property_Type      -4.3377     0.2415 -17.964  < 2e-16 ***
No_of_guests       16.2445     1.0800  15.041  < 2e-16 ***
Bedrooms           -7.6619     1.7827  -4.298 1.80e-05 ***
Beds               -3.8001     1.6048  -2.368  0.01798 *
Baths              11.8915     3.9324   3.024  0.00253 **
Rating              8.2715     1.0783   7.671 2.60e-14 ***
Kitchen             5.5721     3.1918   1.746  0.08100 .
Free_Parking       -0.2465     1.7908  -0.138  0.89054
Free_Cancellation  -0.7407     1.8559  -0.399  0.68986
Self_Check_In      25.1192     4.7896   5.245 1.73e-07 ***
Heating            -3.2578     3.0309  -1.075  0.28256
Washer              3.3681     2.6355   1.278  0.20140
Pool              -17.6192    16.0358  -1.099  0.27201
Gym                30.2068    10.0825   2.996  0.00277 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 36.9 on 2084 degrees of freedom
Multiple R-squared:  0.3602,    Adjusted R-squared:  0.3556
F-statistic: 78.23 on 15 and 2084 DF,  p-value: < 2.2e-16
```

Insight

City, Property_Type, No_of _guests, Bedrooms, Rating and Self_Check_in are the most statistically significant factors in predicting the renting price of a property for Airbnb Listing.