Project 1: Analyzing Housing Market through HomeZilla data

Yunqiu (Julie) Li, Kun Qiu, Yudan Ding, Jiahui Zhong

Fall 2018

1. Exploy data structure

```
library(readxl)
library(tidyverse)

# a.
## Read in the '62 Properties' page in Homezilla spreadsheet and name
it as properties.
properties <- read_excel("/Users/liyunqiu/Desktop/Fall 2018/big data
I/Homezilla.xlsx")

## readxl works best with a newer version of the tibble package.
## You currently have tibble v1.4.2.
## Falling back to column name repair from tibble <= v1.4.2.
## Message displays once per session.

## Read in the 'browsing Data' page in Homezilla spreadsheet and name
it as browsing
browsing <- read_excel("/Users/liyunqiu/Desktop/Fall 2018/big data
I/Homezilla.xlsx", sheet = "Browsing Data")

## Display the internal struture for properties
str(properties)

## Classes 'tbl_df', 'tbl' and 'data.frame':    62 obs. of  10
variables:
##  $ Web ID     : chr  "F1410261" "V1089633" "V1052961" "V1071997" ...
##  $ type       : chr  "house" "house" "house" "house" ...
##  $ subtype    : chr  "Single Family Detached" "Condo Apartment"
"Single Family Detached" "Single Family Detached" ...
##  $ sqfoot     : num  4115 990 4359 2769 1975 ...
##  $ bedrooms   : num  3 2 3 5 3 6 2 2 1 2 ...
##  $ bathrooms  : num  4 2 4 4 3 4 2 1 1 1 ...
##  $ half baths : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ price      : num  1050000 199900 1399000 2798000 598800 ...
##  $ status     : chr  "STACT" "STACT" "STINA" "STINA" ...
##  $ last update: chr  "2014-07-09 155732" "2014-10-10 152509" "2014-
09-15 133614" "2014-08-19 154230" ...

## Display the internal struture for browsing
str(browsing)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    29491 obs. of  15
variables:
##  $ Web ID     : chr  "F1410261" "F1410261" "F1410261" "F1410261" ...
##  $ Time Viewed: num  16 2 0 1 1 0 29 4 3 2 ...
##  $ Timestamp  : POSIXct, format: "2014-10-12 21:46:41" "2014-10-12
21:46:43" ...
##  $ Direction  : chr  NA "left" "left" "left" ...
##  $ Photo ID   : chr
"http://images.homezilla.ca/img/26/260938170_1.jpg"
"http://images.homezilla.ca/img/26/260938170_20.jpg"
"http://images.homezilla.ca/img/26/260938170_19.jpg"
"http://images.homezilla.ca/img/26/260938170_19.jpg" ...
##  $ Photo Tag 1: chr  "exterior" "exterior" "exterior" "exterior" ...
##  $ Photo Tag 2: chr  "waterfront" "other-exterior" "other-exterior"
"other-exterior" ...
##  $ Photo Tag 3: chr  NA "sunview" NA NA ...
##  $ Photo Tag 4: chr  NA NA NA NA ...
##  $ Photo Tag 5: chr  NA NA NA NA ...
##  $ Photo Tag 6: chr  NA NA NA NA ...
##  $ Photo Tag 7: chr  NA NA NA NA ...
##  $ Photo Tag 8: logi  NA NA NA NA NA NA ...
##  $ User Agent : chr  "Mozilla/5.0 (iPhone; CPU iPhone OS 8_0_2 like
Mac OS X) AppleWebKit/600.1.4 (KHTML like Gecko) Mobile/12A405 [F"|
__truncated__ "Mozilla/5.0 (iPhone; CPU iPhone OS 8_0_2 like Mac OS X)
AppleWebKit/600.1.4 (KHTML like Gecko) Mobile/12A405 [F"| __truncated__
"Mozilla/5.0 (iPhone; CPU iPhone OS 8_0_2 like Mac OS X)
AppleWebKit/600.1.4 (KHTML like Gecko) Mobile/12A405 [F"| __truncated__
"Mozilla/5.0 (iPhone; CPU iPhone OS 8_0_2 like Mac OS X)
AppleWebKit/600.1.4 (KHTML like Gecko) Mobile/12A405 [F"| __truncated__
...
##  $ Customer ID: num  239898 239898 239898 239898 239898 ...

## Keep properties that are houses in the data-frame properties
properties_clean <- properties %>%
  filter(type == "house")
```

There are 62 rows(oberservations) and 10 columns(variables) in the properties data sheet, and there are 29491 rows(oberservations) and 15 columns(variables) in the browsing data sheet.

2. Understand housing markets

```
# a.
## Extract the number of distinct customers and number of distinct
photos by property Web ID from the browsing dataset
browsing $`Web ID` <- as.factor(browsing$`Web ID`)
browsing_new <-  browsing %>%
group_by(`Web ID`) %>%
summarize(Distinct_Customer=n_distinct(`Customer ID`),
          Distinct_Photo=n_distinct(`Photo ID`))
```

```r
# b.
## Merge the data extrated above with properties_clean
df_merge <- merge(properties_clean, browsing_new)

# c.

## Look at the supply of houses, number of customers and the average
price for each subtype
df_merge %>%
  group_by(subtype) %>%
  summarise(HouseSupply=n(),
            HousePrice=mean(price), Customer=sum(Distinct_Customer))

## # A tibble: 3 x 4
##   subtype                HouseSupply HousePrice Customer
##   <chr>                        <int>      <dbl>    <int>
## 1 Condo Apartment                 16    391881.      488
## 2 Single Family Detached          36    941233.      944
## 3 Townhouse                        9    403419.      255

## Look at supply of houses, number of customers and the average price
for each subtype and number of bedrooms
df_merge %>%
  group_by(subtype,bedrooms) %>%
  summarise(HouseSupply=n(),
            HousePrice=mean(price), Customer=sum(Distinct_Customer))

## # A tibble: 13 x 5
## # Groups:   subtype [?]
##    subtype                bedrooms HouseSupply HousePrice Customer
##    <chr>                     <dbl>       <int>      <dbl>    <int>
##  1 Condo Apartment               1           6    327083.      127
##  2 Condo Apartment               2           8    318587.      332
##  3 Condo Apartment               3           2    879450        29
##  4 Single Family Detached        2           2    143450        45
##  5 Single Family Detached        3          11    705327.      485
##  6 Single Family Detached        4           7    720200       100
##  7 Single Family Detached        5           8   1382837.      194
##  8 Single Family Detached        6           4   1101725        87
##  9 Single Family Detached        7           1   1498888         5
## 10 Single Family Detached        8           2   1340000.       24
## 11 Single Family Detached        9           1   1149000         4
## 12 Townhouse                     2           3    328362.      150
## 13 Townhouse                     3           6    440948       105
```

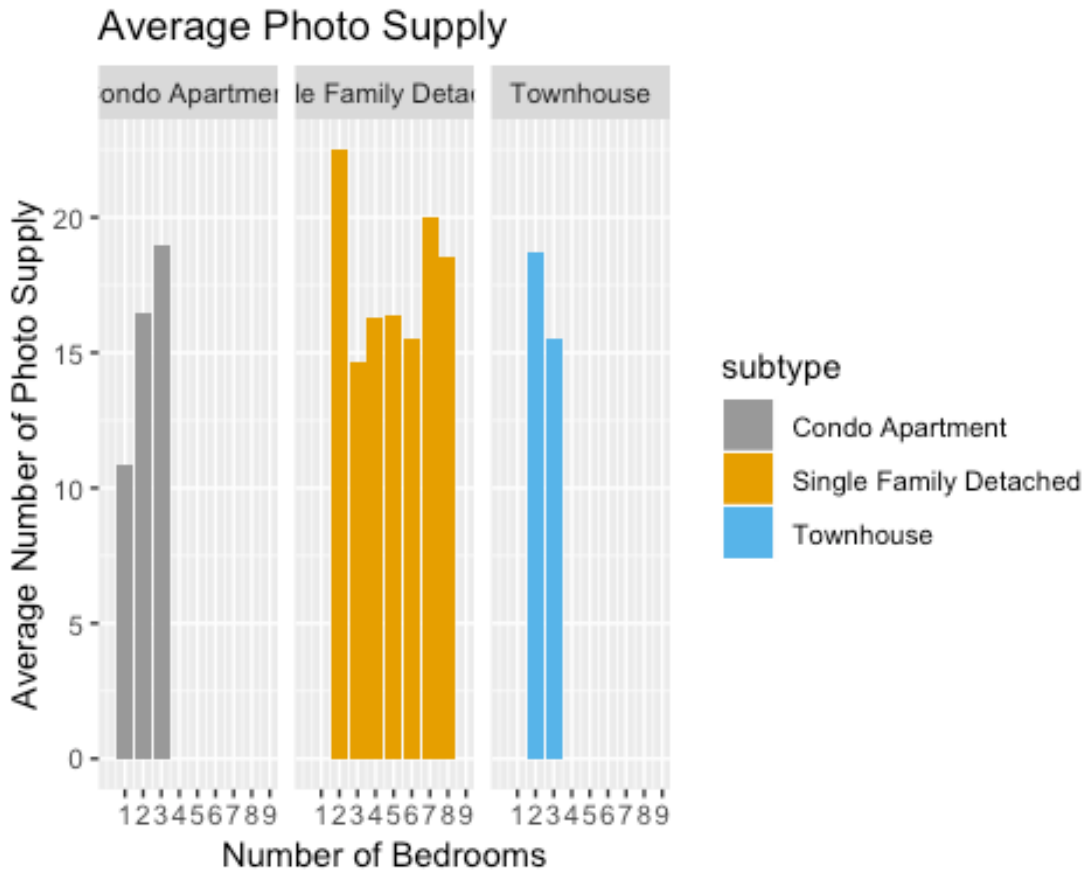
Based on the first table output, the housing market consists mostly of Single Family Detached houses, with the higest average price (941232.9) among subtypes. The number of customers(944) in this subtype is also significantly higher than that of Condo Apartment or Townhouse. When considering it with the number of bethroom for each subtype(seond table output), we can see Single Family Detached houses

with 3 bedrooms are most popular among all. We also find Single Family Detached with 5 bedrooms is the only kind of house whose customers are over 100 with price beyond 1 million. HomeZilla should definitely put the most focus on Family Detached houses since they're the most popular ones among customers and can sell at the highest prices among all. Within this subtype, they should pay more attention to Single Family Detached houses with 3 and 5 bedrooms.

3. Explore supply of pictures

```
## Look at supply of houses, number of customers and the average price
for each subtype and number of bedrooms
plot <- df_merge %>%
  group_by(subtype,bedrooms) %>%
  summarise(HouseSupply=n(),
            HousePrice=mean(price),
            Customer=sum(Distinct_Customer),
            AveragePhotoNumber = (sum(Distinct_Photo)/n_distinct(`Web
ID`))) %>%   arrange(AveragePhotoNumber)
# display "plot"
plot

## # A tibble: 13 x 6
## # Groups:   subtype [3]
##    subtype        bedrooms HouseSupply HousePrice Customer
AveragePhotoNum…
##    <chr>             <dbl>       <int>      <dbl>    <int>
<dbl>
##  1 Condo Apartme…        1           6    327083.      127
10.8
##  2 Single Family…        3          11    705327.      485
14.6
##  3 Single Family…        6           4   1101725        87
15.5
##  4 Townhouse             3           6    440948       105
15.5
##  5 Single Family…        4           7    720200       100
16.3
##  6 Single Family…        5           8   1382837.      194
16.4
##  7 Condo Apartme…        2           8    318587.      332
16.5
##  8 Single Family…        8           2   1340000.       24
18.5
##  9 Townhouse             2           3    328362.      150
18.7
## 10 Condo Apartme…        3           2    879450        29
19
## 11 Single Family…        7           1   1498888         5
20
## 12 Single Family…        9           1   1149000         4
```

```
20
## 13 Single Family…        2           2    143450        45
22.5

# Plot bar graph of average number of photo supply by number of
bedrooms, faceting by house subtype
ggplot(plot, aes(x = bedrooms, y = AveragePhotoNumber, fill = subtype))
+
  geom_bar(stat = 'identity') +
  facet_wrap(~subtype, scale = 'free_x') +
  scale_x_continuous(name = 'Number of Bedrooms', limits = c (0,9),
breaks=c(1,2,3,4,5,6,7,8,9)) +
  scale_y_continuous(name = 'Average Number of Photo Supply') +
  scale_fill_manual(values=c("#999999", "#E69F00", "#56B4E9")) +
  labs(title = "Average Photo Supply") +
  theme_update(plot.title = element_text(hjust = 0.5))
```



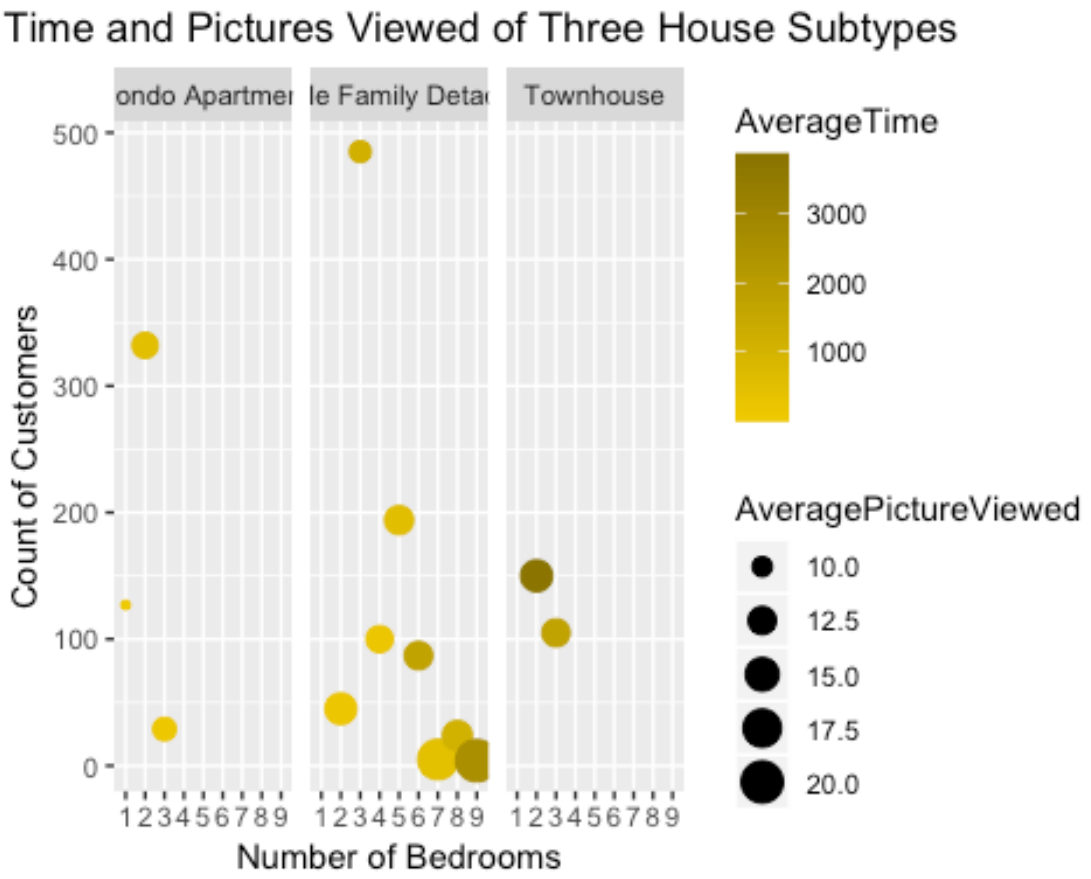
We choose average number of photos because the total number of photos could not provide a precise answer. As the number of properties for each subtype is different, the total number would be biased simply because a given property subtype has a larger supply. For example, if we generate the total photo number for each house subtype, single family detached will have the highest number because the supply of single family detached property(36) is way larger than the other two subtypes(16

and 9). From the above table output, we can see the average numbers of picture mainly concentrate between 15-20. Single family detached houses with 2 bedrooms has the highest average number of pictures(22.5), which is not corresponding to our previous finding. We would recommend HomeZilla to provide more pictures for single family detached with 3 and 5 bedrooms because these two categories have the largest number of customers with relatively high price, but currently they do not have the most pictures.

4. Explore demand of pictures

```
# a.
## Extract the total time spent on viewing pictures and the number of
pictures viewed for every customer and property from the browsing
dataset
photocustomer <- browsing %>%
  group_by(`Web ID`,`Customer ID`) %>%
  summarise(TotalBrowsingTime = sum(`Time Viewed`),PicturesViewed =
n_distinct(`Photo ID`)) %>%
  summarise(TotalBrowsingTime = sum(TotalBrowsingTime) /
n_distinct(`Customer ID`),PictureViewed = sum(PicturesViewed) /
n_distinct(`Customer ID`))
photocustomer

## # A tibble: 62 x 3
##    `Web ID` TotalBrowsingTime PictureViewed
##    <fct>                <dbl>         <dbl>
##  1 357518               258.          21.0
##  2 377268                36.7          3.66
##  3 377688               100            8
##  4 F1410261             164.          11.1
##  5 F1411209              33           11
##  6 F1415029           10158.           9.88
##  7 F1415727            1771.          11.7
##  8 F1417624            2580.          20
##  9 F1418582             304.           4.88
## 10 F1421451             141.          16.5
## # ... with 52 more rows

# b.
properties_new <- merge(df_merge,photocustomer)

# c.
## Look at the number of customers, average pictures viewed and average
time spent on viewing pictures for each subtype and number of bedrooms
plot4 <- properties_new %>%
  group_by(subtype,bedrooms) %>%
  summarise(Customer=sum(Distinct_Customer),
            AverageTime = (sum(TotalBrowsingTime)/n()),
AveragePictureViewed = sum(PictureViewed)/n())
plot4
```

```
## # A tibble: 13 x 5
## # Groups:   subtype [?]
##    subtype                bedrooms Customer AverageTime
AveragePictureView…
##    <chr>                     <dbl>    <int>       <dbl>
<dbl>
##  1 Condo Apartment               1      127        79.7
8.76
##  2 Condo Apartment               2      332       501.
11.7
##  3 Condo Apartment               3       29       130.
10.9
##  4 Single Family Detached        2       45       162.
14.0
##  5 Single Family Detached        3      485      1132.
10.4
##  6 Single Family Detached        4      100       204.
12.0
##  7 Single Family Detached        5      194       558.
12.8
##  8 Single Family Detached        6       87      1725.
12.5
##  9 Single Family Detached        7        5       477.
18.8
## 10 Single Family Detached        8       24      1080.
13.1
## 11 Single Family Detached        9        4      2580.
20
## 12 Townhouse                     2      150      3785.
14.1
## 13 Townhouse                     3      105      1770.
12.4

AveragePictureViewed = plot4$AveragePictureViewed
AverageTime = plot4$AverageTime

ggplot(plot4,aes(x=factor(bedrooms), y = Customer,size =
AveragePictureViewed,color = AverageTime))+
  geom_point(stat = "identity")+
  facet_wrap(~subtype)+
  scale_color_gradient(low = "#EEC900",high = "#8B7500")+
  labs(x = 'Number of Bedrooms', y = 'Count of Customers', title =
"Average Time and Pictures Viewed of Three House Subtypes")
```



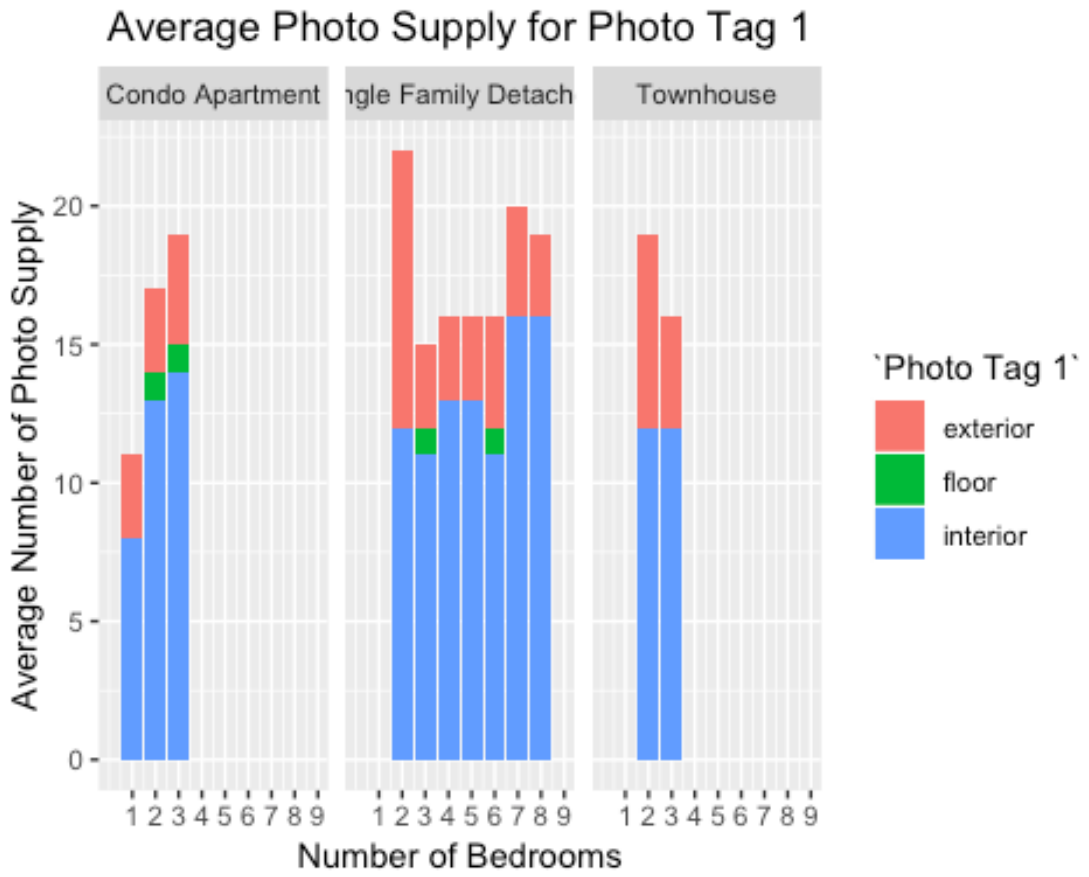
While the average numbers of picture mainly concentrate between 15-20, we can find that the average number of pictures viewed focus between 10-14. There are 6 categories from different house subtypes whose average view time is beyond 1000 (Single family detached houses with 3,6,8 and 9 bedrooms and Townhouses with 2 and 3 bedrooms). Based on the length of view time, We can infer that pictures of these houses are of good quality. The average view time of Condo Apartment is far below average, so we recommend Homezilla improve the quality of their pictures.

Combined with our findings before, we recommend Homezilla provides more good quality pictures for Single Family Detached houses with 3 bedrooms and improve the quality of pictures for Single Family Detached houses with 5 bedrooms.

5. Explore types of pictures

```
# classify photo-types
PhotoLoc <- browsing %>% group_by(`Web ID`,`Photo Tag 1`) %>%
  summarise(PhotoSupply = n_distinct(`Photo ID`))

df_Photoloc <- merge(properties_clean, PhotoLoc)

# Understand supply of pictures of each type by property subtype and
number of bedrooms
plot2 <- df_Photoloc %>% group_by(subtype, bedrooms, `Photo Tag 1`) %>%
```

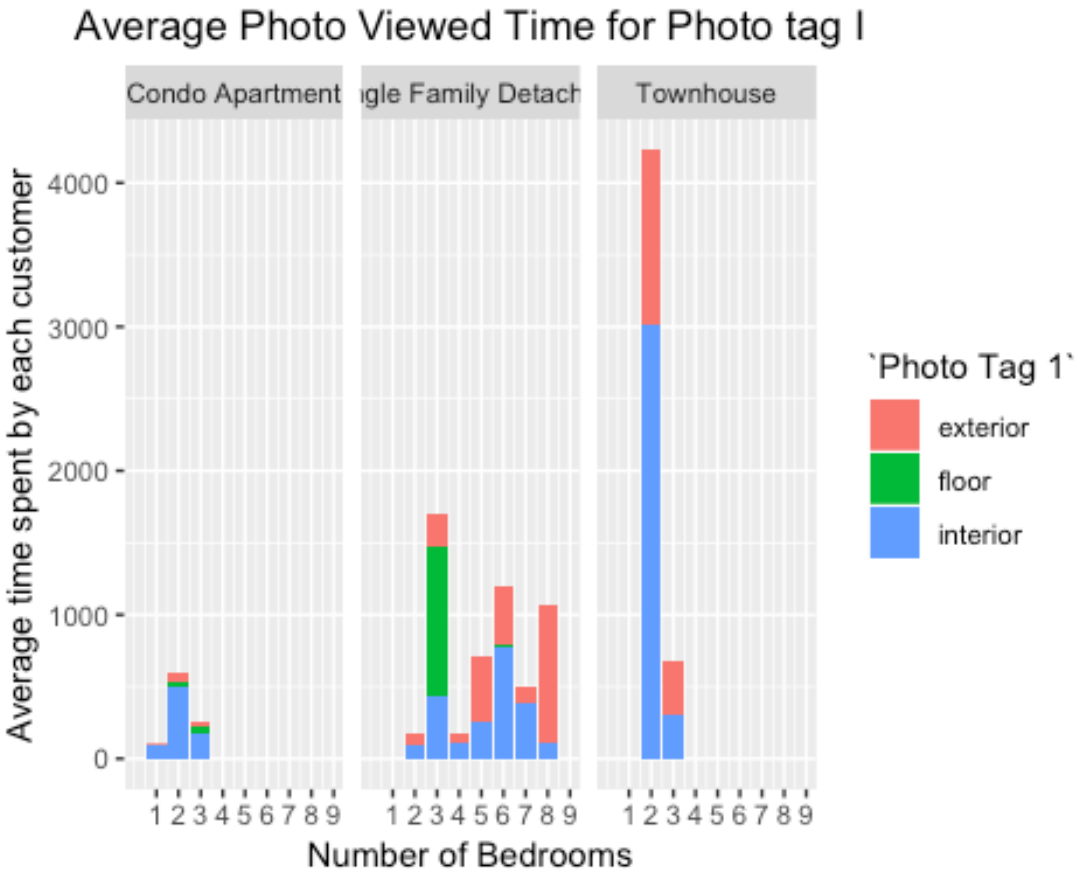
```
  summarise( AvgPhotoSupply = round((sum(PhotoSupply)/n_distinct(`Web
ID`))))

# Plot bar graph of photo supply for different tag under Photo tag I
category by number of bedrooms, faceting by house subtype
ggplot(plot2, aes(x = bedrooms, y = AvgPhotoSupply, fill = `Photo Tag
1`)) +
  geom_bar(stat = 'identity') +
  facet_grid(cols = vars(subtype)) +
  scale_x_continuous(name = 'Number of Bedrooms', limits = c (0,9),
breaks=c(1,2,3,4,5,6,7,8,9)) +
  scale_y_continuous(name = 'Average Number of Photo Supply')+
  labs(title = "Average Photo Supply for Photo Tag 1")
```



As shown in the above table, HomeZilla provides more pictures of house interiors to users. The company may want to stock more pictures of floor plans, which is not currently available for townhouses. It may also consider adding more pictures of single family detached with 3 bedrooms and 5 bedrooms, as mentioned above.

```
# Understand which properties do consumers spend more time looking at
PhotoDemand <- browsing %>% group_by(`Customer ID`,`Web ID`, `Photo Tag
1`) %>% summarise(PhotoViewed = n_distinct(`Photo ID`), TimeSpent =
sum(`Time Viewed`))
```

```r
df_photodemand <- merge(PhotoDemand, properties_clean)

plot3 <- df_photodemand %>% group_by(subtype, bedrooms, `Photo Tag 1`)
%>%
  summarise(AvgPhotoViewed = sum(PhotoViewed)/n(), AvgTimeSpent =
sum(TimeSpent)/n())

# Plot bar graph of average time spent by each customer for different
tag under Photo tag I category by number of bedrooms, faceting by house
subtype
ggplot(plot3, aes(x = bedrooms, y = AvgTimeSpent, fill = `Photo Tag
1`)) +
  geom_bar(stat = 'identity') +
  facet_wrap(~subtype, scales = "free_x") +
  scale_x_continuous(name = 'Number of Bedrooms', limits = c (0,9),
breaks=c(1,2,3,4,5,6,7,8,9)) +
  scale_y_continuous(name = 'Average time spent by each customer') +
  labs(title = "Average Photo Viewed Time for Photo tag I")
```



As shown in the above table, customers generally spent more time in viewing pictures of house interiors(blue) than exteriors(red) or floor plans(green). However, for single family detached with 3 bedrooms users tend to spend more

time in viewing floor plans. Similar to the pattern we noticed in Question 4, customers who were searching for townhouses with 2 bedrooms on average spent more time in viewing pictures. However, we cannot conclude that users who are searching for townhouses are more interested in examining pictures as users may try to get a sense of the floor plan by viewing pictures of interiors.

6. Final Reflections

Discoveries regarding cleaning, analyzing and visualizing data:

We find that the first key step to clean data would be removing the columns and rows that do not have a corresponding value in it, because furthur analysis will all implement based on the cleaned dataset. Before implement fuctions or plot graph based on specific varaibles, it would be worthwhile to check their conceptual data type first by using the str() function. After that, we will decide if factorize a variable is necessary or which specific type of graph we shall use according to thd data type. Also, to get a broader picture of the customer behaviorial pattern, merging cleaned datasets together would be very beneficial as we can conduct analysis based on more variables to get a tailored result. When filter, group or summarize data, it would be better to start from thinking what would be the result we are trying to generate, so we can implement a given function more precisely. We also feel that "group by" and "summarise" would be two of the most helpful functions to figure out the patterns for dataset. These two allow us to group dataset into different sub-categories and implement aggregate function based on those sub-categories.

Discoveries regarding Homezilla:

Overall, properties that have bedrooms in the middle of range would be most popular among three subtypes. For example, condo apartment with two bedrooms have the most customers among all condo apartment. The overall range of number of bedrooms for condo apartment is 1-3. Similary, single family detached with 3-5 bedrooms have relative more customers among all single family detached. The overall range of number of bedrooms for single family detached is 2-9. Therefore, Homezila should focus more on those properties through increasing house supply, adding tailored marketing campign, increasing the number of photos, etc. Through closely following customer demand, Homezilla can generate an increasing number of potential sales. Homezilla may also want to improve its process of obtaining the 'Time Viewed' information. As mentioned in the case there is no timestamp for the last-viewed photo in each access and thus Homezilla puts 0 for the Time Viewed cells for all last-viewed photos. This may potenetially distort data and affect the validity of our conclusion. We also spotted that there are some 'Time Viewed' values beyond 100,000, and the highest value is up to 332,909. It can be practical meaningful because it may indicate customers really like these houses and have seen these pictures over and over again. We cannot decide what standard to use to judge a certain value an outlier since we know little about typical user behaviours in Homezilla websites. Qualitative research should also be applied to gain infomation about user behaviours, which can be used to set up a more reasonable standard.

Overall, we do concern about the effects of outliers, and strongly suggest Homezilla to improve its process of acquiring the 'Time Viewed' information and establish clear rules for data cleaning.