Analyzing Oregon Housing Data: A Spatial Analysis

Aparajita Sengupta

4/14/2022

Purpose of the report:

This project report is aimed at analyzing the housing data in Corvallis, Oregon, and predicting the price of housing in the area based on the following factors: 1. Total squarefeet 2. Year built 3. Acres 4. Class 5. Bedrooms 6. Full baths 7. Month of sale

Techniques used:

1. Visualizing Geospatial Data in R
2. ggmap
3. Drawing polygons
4. Choropleth maps
5. Exploratory Data Analysis (EDA)

Executive Summary:

Import libraries:

```
#Loading libraries-----
library(knitr)
library(tidyverse) #collection of R packages designed for data science
library(assertive) #An R package that provides readable check functions to ensure code integrit
y.
library(ggmap)
library(sp)
library(lubridate) #Date and time
library(corrr) #for exploring correlations
library(PerformanceAnalytics) # used to display a chart of a correlation matrix
library(skimr) #provide summary statistics about variables in data frames, tibbles, data tables
vectors
library(janitor) #for examining and cleaning dirty data.
library(here) #The here package creates paths relative to the top-level directory. The package
 displays the top-level of the current project on load
library(lm.beta) #Add Standardized Regression Coefficients to lm-Objects
library(magrittr) #decrease development time and improve readability and maintainability of cod
e.
library(caTools) #for splitting data for test and train sets
library(ggpubr) #correlation
#library(recipes)
library(lme4) #modern, efficient linear algebra methods as implemented in the Eigen package
methods(sigma)
library(future)
#install.packages(pkgs = "caret", dependencies = c("Depends", "Imports"))
library(mlbench)#Returns the decision of the (optimal) Bayes classifier for a given data set.
library(caret) #Classification And REgression Training
#https://www.pluralsight.com/guides/explore-r-libraries:-caret
library(ROSE)
library(ggcorrplot)
library(corrplot)
library(extrafontdb)
library(hrbrthemes)
library(Rttf2pt1)
library(sqldf)
library(rockchalk) #combining variables
library(readr)
library(forcats) #solves common problems with factors, including changing the order of levels o
r the values.
# to concatenate plots and dataset
library(glue)
library(patchwork)
library(lares) #for cross-correlation
library(patchwork)
library(broom)
library(purrr)
library(htmlTable)
library(base)
```

Descriptive analysis and cleaning the data:

Glimpse of the data

```
#head(sales_init)
glimpse(sales_init)
```

```
## Rows: 931
## Columns: 20
## $ lon                 <dbl> -123.2803, -123.2330, -123.2635, -123.2599, -123.2…
## $ lat                 <dbl> 44.57808, 44.59718, 44.56923, 44.59453, 44.53606, …
## $ price               <dbl> 267500, 255000, 295000, 5000, 13950, 233000, 24500…
## $ finished_squarefeet <int> 1520, 1665, 1440, 784, 1344, 1567, 1174, 912, 1404…
## $ year_built          <int> 1967, 1990, 1948, 1978, 1979, 2002, 1972, 1970, 20…
## $ date                <date> 2015-12-31, 2015-12-31, 2015-12-31, 2015-12-31, 2…
## $ address             <chr> "1112 NW 26TH ST", "1221 NE CONROY PL", "440 NW 7T…
## $ city                <chr> "CORVALLIS", "CORVALLIS", "CORVALLIS", "CORVALLIS"…
## $ state               <chr> "OR", "OR", "OR", "OR", "OR", "OR", "OR", "OR", "O…
## $ zip                 <chr> "97330-4331", "97330", "97330-6308", "97330-3654",…
## $ acres               <dbl> 0.18, 0.13, 0.12, 0.00, 0.00, 0.04, 0.23, 0.22, 0.…
## $ num_dwellings       <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,…
## $ class               <chr> "Dwelling", "Dwelling", "Dwelling", "Mobile Home",…
## $ condition           <chr> "AV", "AV", "G", "F", "AV", "AV", "G", "AV", "AV",…
## $ total_squarefeet    <int> 2192, 2193, 1860, 784, 1344, 2007, 1678, 1440, 160…
## $ bedrooms            <int> 5, 3, 3, 0, 0, 3, 3, 3, 3, 3, 3, 5, 0, 6, 0, 4, 5,…
## $ full_baths          <int> 2, 2, 2, 1, 2, 2, 1, 1, 2, 1, 2, 3, 2, 3, 1, 2, 2,…
## $ half_baths          <int> 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1,…
## $ month               <dbl> 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12…
## $ address_city        <chr> "1112 NW 26TH ST, CORVALLIS, OR", "1221 NE CONROY …
```

Summary of the numeric features

'year_built' column has been modified to create a new feature, 'age'. This will make it easier to visualize the change in price of the house (if any) with age.

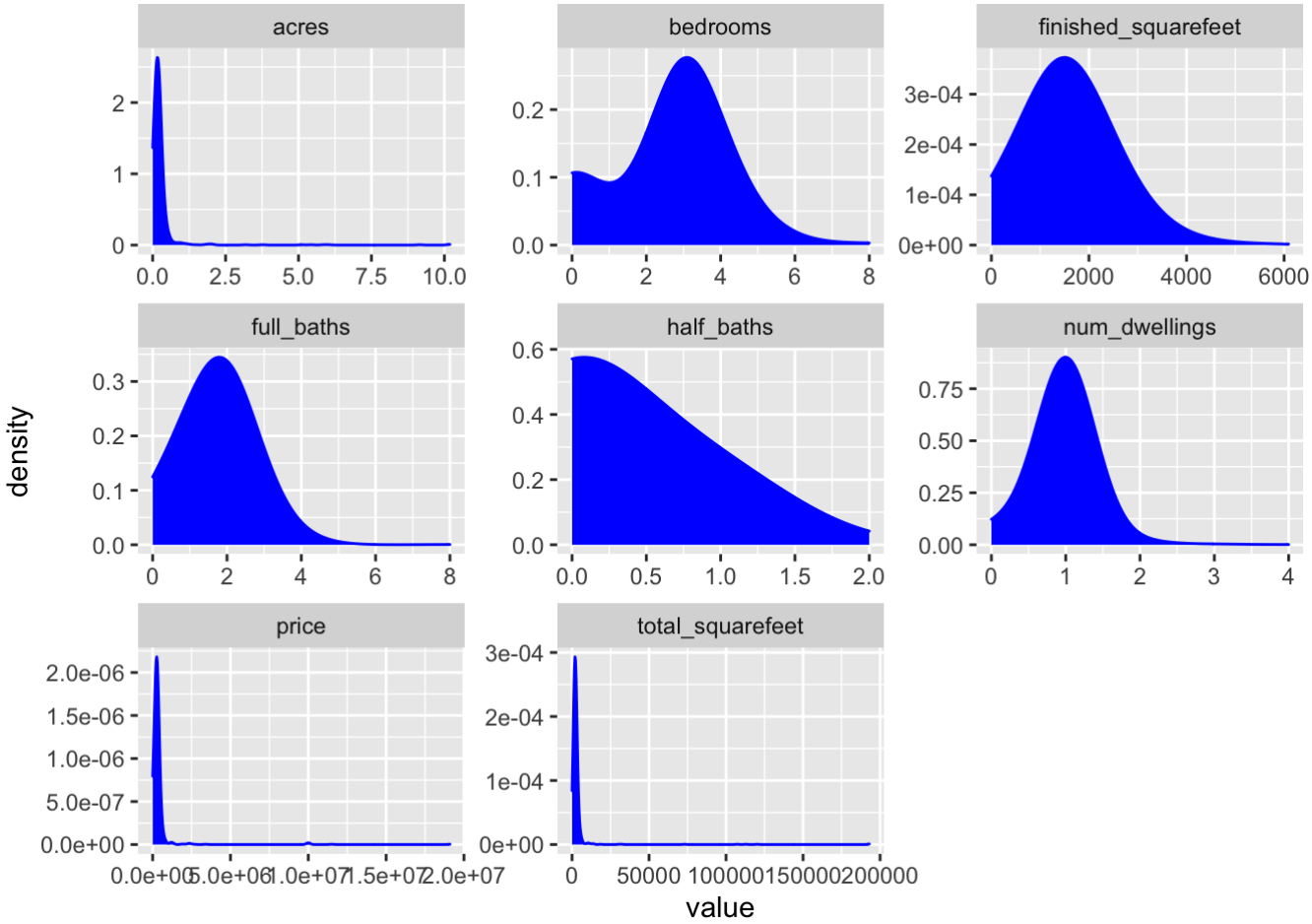
```
# *Summary of the numerical varables:*
sales_init_1 <- sales_init %>%
        keep(is.numeric) %>%   # Keep only numeric columns
        select(-lon, -lat,-year_built,-month)
summary(sales_init_1)
```

```
##      price          finished_squarefeet     acres         num_dwellings
##  Min.   :       1   Min.   :   0        Min.   : 0.0000   Min.   :0.0000
##  1st Qu.:  196250   1st Qu.:1090        1st Qu.: 0.1100   1st Qu.:1.0000
##  Median :  253400   Median :1453        Median : 0.1800   Median :1.0000
##  Mean   :  379911   Mean   :1531        Mean   : 0.2614   Mean   :0.9431
##  3rd Qu.:  341250   3rd Qu.:1975        3rd Qu.: 0.2300   3rd Qu.:1.0000
##  Max.   :19100000   Max.   :6093        Max.   :10.1900   Max.   :4.0000
##                                         NA's   :4
##  total_squarefeet    bedrooms       full_baths      half_baths
##  Min.   :     0   Min.   :0.000   Min.   :0.000   Min.   :0.0000
##  1st Qu.:  1562   1st Qu.:2.000   1st Qu.:1.000   1st Qu.:0.0000
##  Median :  2007   Median :3.000   Median :2.000   Median :0.0000
##  Mean   :  3170   Mean   :2.544   Mean   :1.654   Mean   :0.2868
##  3rd Qu.:  2668   3rd Qu.:3.000   3rd Qu.:2.000   3rd Qu.:1.0000
##  Max.   :192991   Max.   :8.000   Max.   :8.000   Max.   :2.0000
##
```

Univariate Analysis: Distribution (smoothed histogram) of the numeric features:

```
All_distribution <- sales_init_1 %>%
        #drop_na() %>%
        gather() %>%                             # Convert to key-value pairs
        ggplot(aes(value)) +                     # Plot the values
        facet_wrap(~ key, scales = "free") +   # In separate panels
        #geom_density(color="blue", adjust = 5)  +
        geom_density(color="blue", fill="blue",adjust = 5) #+  # as density


All_distribution
```

```
## Warning: Removed 4 rows containing non-finite values (stat_density).
```



Observations:

1. The sales data set has 931 data points, and 20 features. 7 features are characters, one feature is a date and rest of the features are numeric.
2. Acres, price and total squarefeet are highly right-skewed.

- Range of acres is (0.00, 10.19). However, 75% of the values lie below 0.23 acres.
- Range of price is (1, 19100000). However, 75% of the values lie below $341250.

- Range of total squarefeet is (0, 192991). However, 75% of the values lie below 2668 squarefeet.

3. Age is also right-skewed.

Selecting features of interest:

- Remove the city and sales column since it's the same city.
- All dates are in 2015 and hence can be removed.
- Zip has two parts to the data. The first five digits are the same. Hence the zip can be split, and only the last four digits can be used as a feature.
- I have also removed address, city, state from the data frame. City and state are the same for all rows.
- Let's create a new dataframe with the above conditions
- num_dwellings for almost all the data (~97%) is 1. The variation in this column is very low. Hence, we will remove this feature as well.

```
table(sales_init$num_dwellings)
```

```
##
##   0   1   2   3   4
##  78 833  16   3   1
```

```
sales_init_2 <- sales_init %>%
        #mutate(age= 2022 - as.numeric(year_built)) %>%
        select( -date, -zip, -address_city, -city, -state, num_dwellings)

glimpse(sales_init_2)
```

```
## Rows: 931
## Columns: 16
## $ lon                 <dbl> -123.2803, -123.2330, -123.2635, -123.2599, -123.2…
## $ lat                 <dbl> 44.57808, 44.59718, 44.56923, 44.59453, 44.53606, …
## $ price               <dbl> 267500, 255000, 295000, 5000, 13950, 233000, 24500…
## $ finished_squarefeet <int> 1520, 1665, 1440, 784, 1344, 1567, 1174, 912, 1404…
## $ year_built          <int> 1967, 1990, 1948, 1978, 1979, 2002, 1972, 1970, 20…
## $ address             <chr> "1112 NW 26TH ST", "1221 NE CONROY PL", "440 NW 7T…
## $ acres               <dbl> 0.18, 0.13, 0.12, 0.00, 0.00, 0.04, 0.23, 0.22, 0.…
## $ num_dwellings       <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,…
## $ class               <chr> "Dwelling", "Dwelling", "Dwelling", "Mobile Home",…
## $ condition           <chr> "AV", "AV", "G", "F", "AV", "AV", "G", "AV", "AV",…
## $ total_squarefeet    <int> 2192, 2193, 1860, 784, 1344, 2007, 1678, 1440, 160…
## $ bedrooms            <int> 5, 3, 3, 0, 0, 3, 3, 3, 3, 3, 3, 5, 0, 6, 0, 4, 5,…
## $ full_baths          <int> 2, 2, 2, 1, 2, 2, 1, 1, 2, 1, 2, 3, 2, 3, 1, 2, 2,…
## $ half_baths          <int> 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1,…
## $ month               <dbl> 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12…
## $ zip_split           <chr> "97330", "97330", "97330", "97330", "97333", "9733…
```

Taking account of missing data:

The following features have missing data: 1. year_built: 35 data points or 3.8% of the data are missing.

2. class: 23 data points or 2.5% of the data are missing.
3. acres: 4 data points or 0.4% of the data are missing.

```
skimmed_data <- skim(sales_init) %>%
        dplyr::filter(n_missing != 0)   %>%
        dplyr::arrange(desc(n_missing)) %>%
        dplyr::mutate(NA_percent = n_missing*100/nrow(sales_init))

#glimpse(skimmed_data)
skimmed_data$NA_percent
```

```
## [1] 3.7593985 2.4704619 0.4296455
```

```
#OR
#colnames(skimmed_data)


skim(sales_init) %>%
        dplyr::select(skim_type, skim_variable, n_missing)
```

Data summary

Name	sales_init
Number of rows	931
Number of columns	21
_____	
Column type frequency:	
character	8
Date	1
numeric	12
_____	
Group variables	None

Variable type: character

skim_variable	n_missing
address	0
city	0
state	0
zip	0
class	23
condition	0
address_city	0
zip_split	0

Variable type: Date

skim_variable	n_missing
date	0

Variable type: numeric

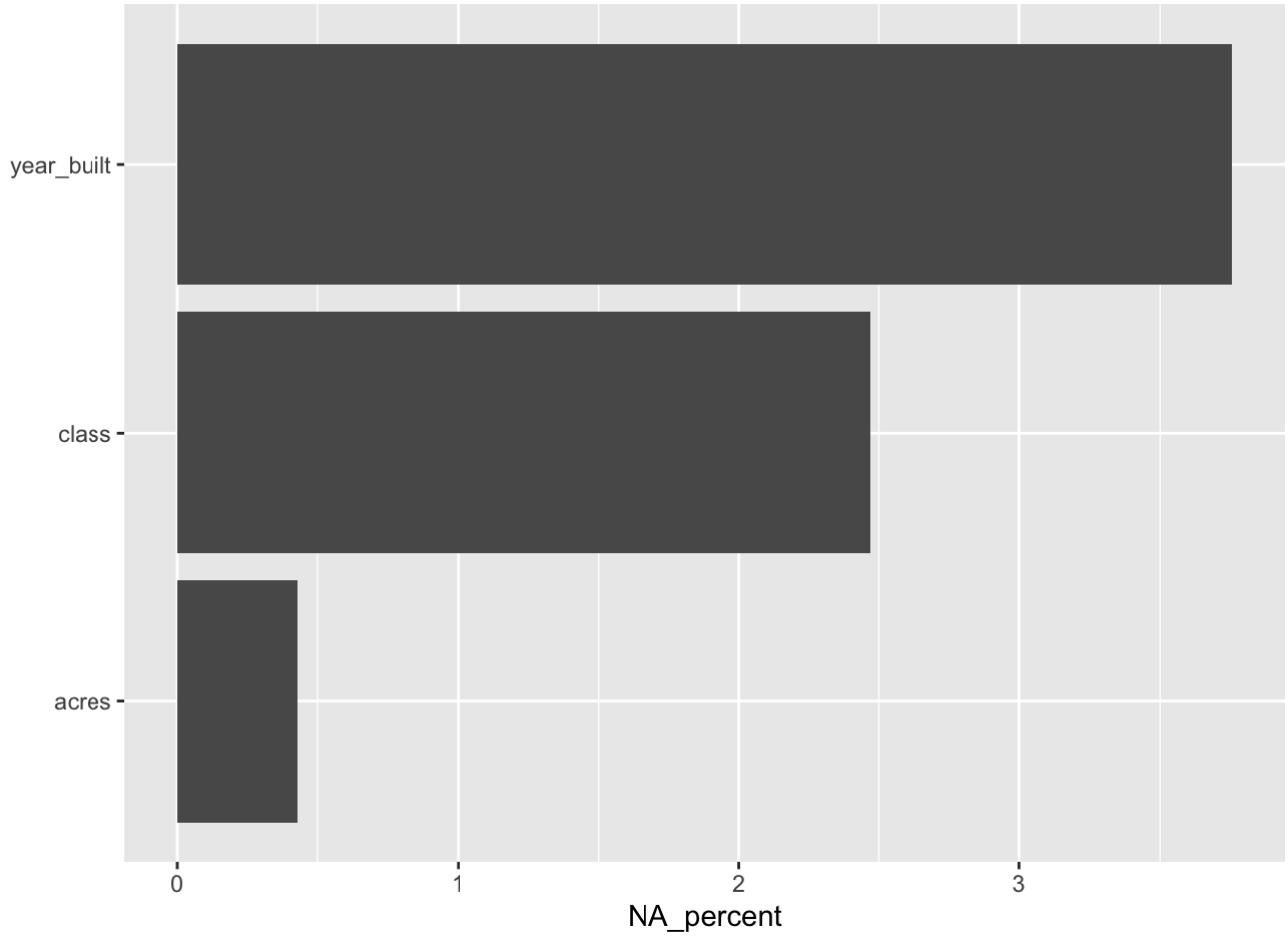
skim_variable	n_missing
lon	0
lat	0
price	0
finished_squarefeet	0
year_built	35
acres	4
num_dwellings	0
total_squarefeet	0
bedrooms	0
full_baths	0
half_baths	0

skim_variable	n_missing
month	0

```
na_features <- ggplot(data=skimmed_data, aes(x = reorder(skim_variable, NA_percent), y =NA_perc
ent)) +
        geom_bar(stat = "identity") +
        coord_flip()+  # as density
        xlab("") #+
        ylab(" ")
```

```
## $y
## [1] " "
##
## attr(,"class")
## [1] "labels"
```

```
na_features
```

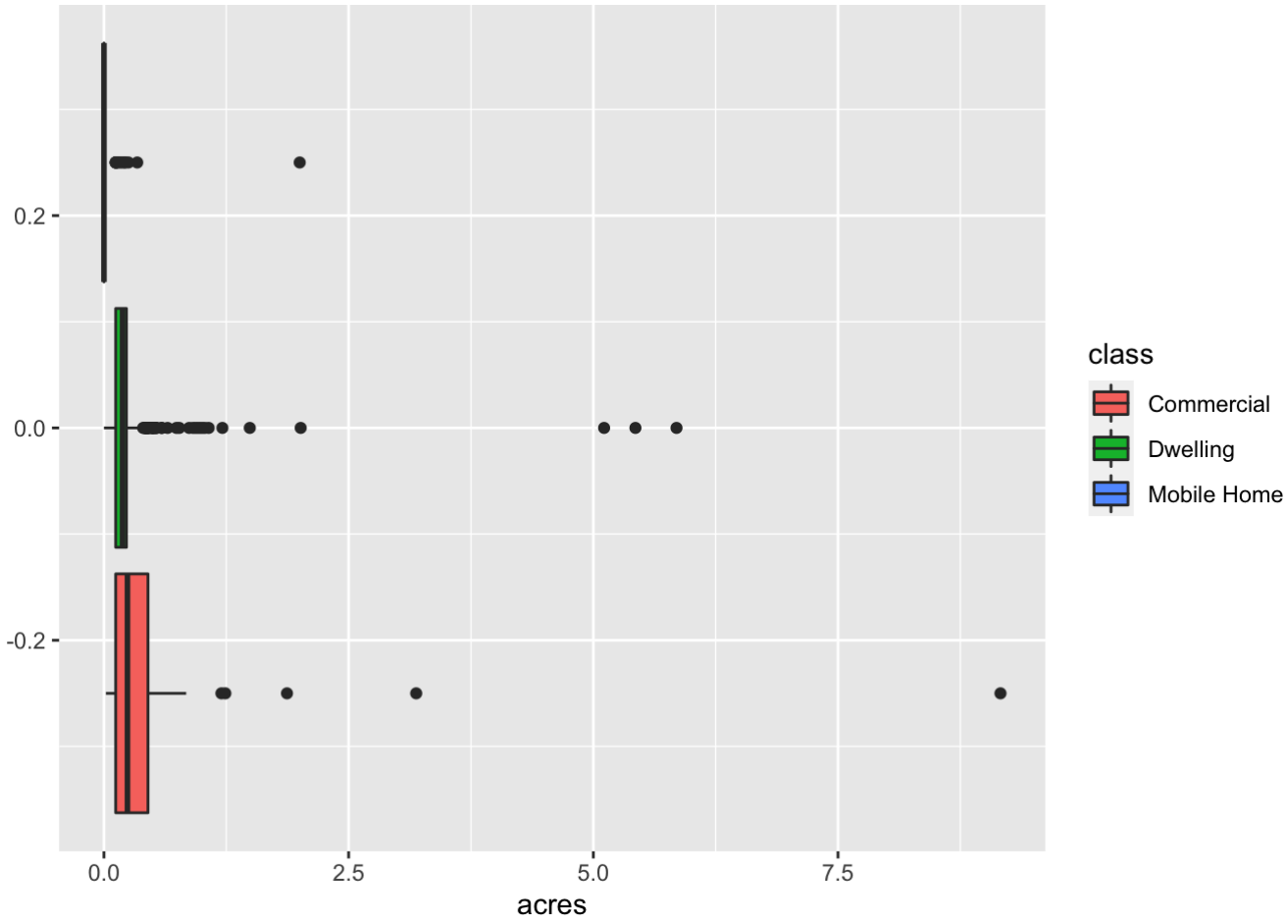


Cleaning data and converting year to age of the house

```
#Cleaning data and converting year to age of the house


sales <- sales_init_2 %>%
        clean_names() %>%
        remove_empty() %>%
        distinct() %>%
        mutate(across(where(is.character), as.factor)) %>%
        mutate(age= 2022 - as.numeric(year_built)) %>%
  filter(!is.na(class) & !class == "RES Feature") %>%
  dplyr::select(-year_built,-address) %>%
  mutate(month = as.character(month),class = as.character(class),bedrooms = as.character(bedroo
ms),full_baths = as.character(full_baths)) %>%
       dplyr::select(price, dplyr::everything()) %>%
  glimpse()

table(sales$class)
```

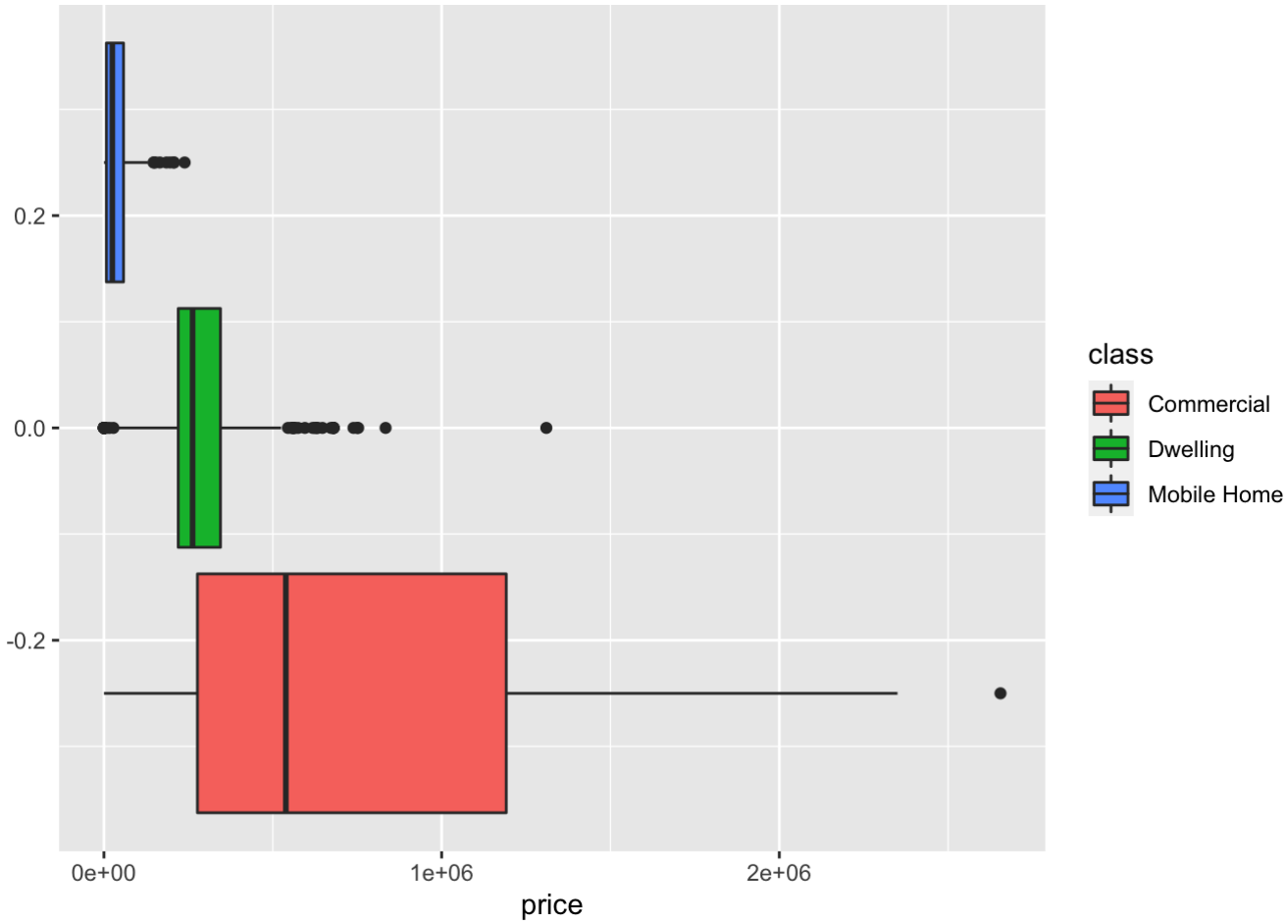
Exploring the categorical data:

```
#########################
#Acres
#########################

#box_acres <- boxplot(log10(sales$acres),
#  ylab = "acres",
#  main = "log of Boxplot of acres"
#)
#mtext(paste("Outliers: ", paste(out, collapse = ", ")))
sales_acres <- sales[!is.na(sales$acres) & sales$price < 0.3*10^7,]
box_acres <- sales_acres %>%
  ggplot(aes(x = acres, fill = class)) +
  geom_boxplot()

box_acres
```




```
#colnames(sales_acres)
```

```
#########################
#Prices
#########################


#box_price <- boxplot(log10(sales$price),
#  ylab = "price",
#  main = "log of Boxplot of price"
#)
#mtext(paste("Outliers: ", paste(out, collapse = ", ")))

sales_price <- sales[!is.na(sales$price) & sales$price < 0.3*10^7,]
box_price <- sales_price %>%
  ggplot(aes(x = price, fill = class)) +
  geom_boxplot()

box_price
```

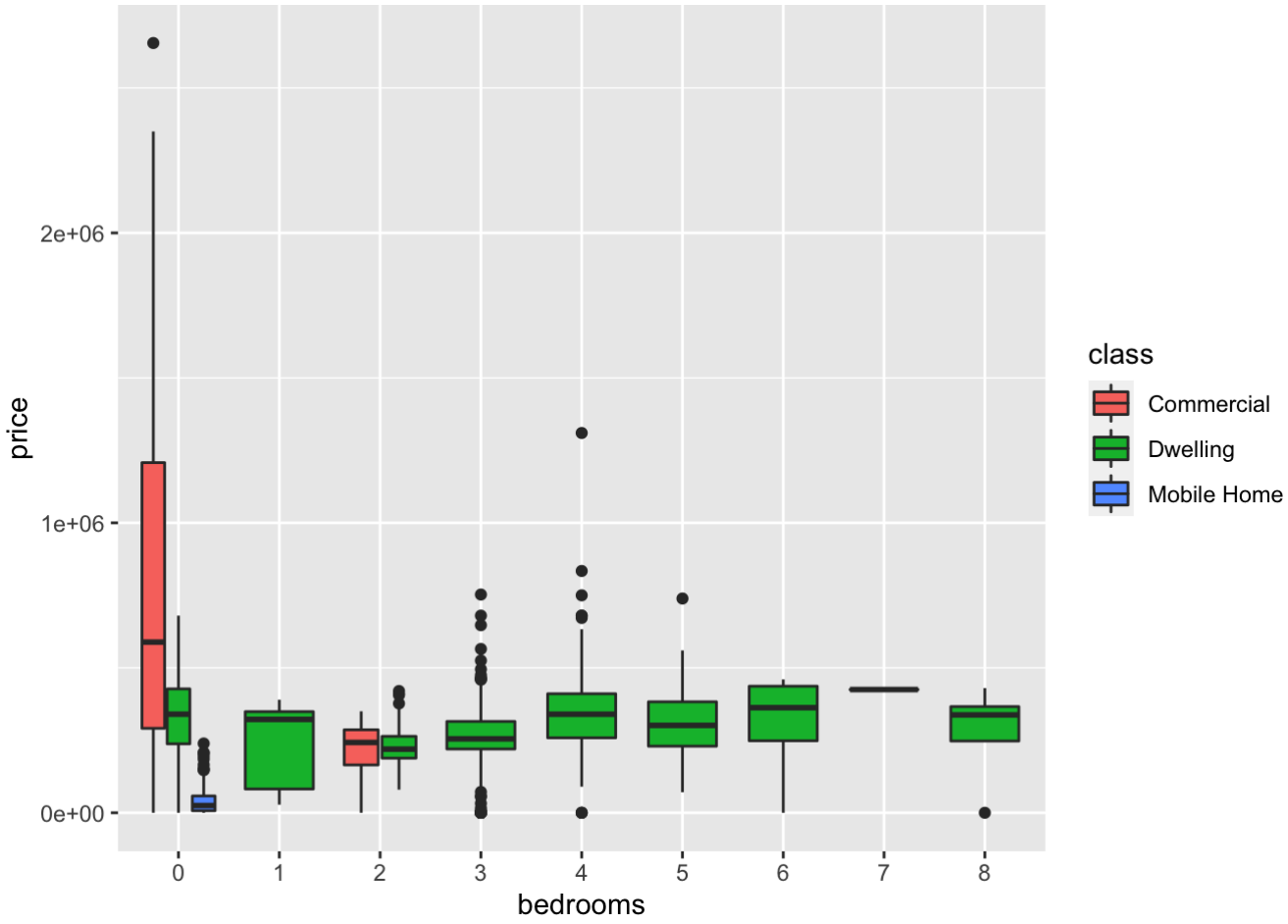



```
#month
table(sales$month)
```

```
##
##   1  10  11  12   2   3   4   5   6   7   8   9
##  48  74  61  54  43  72  74  82 104  93  95  92
```

```
#Patchwork
#theme_set(theme_light())
#p2 <- box_acres  /
#  box_price + plot_layout(guides = "collect") & theme(legend.position = "top")

#p2
```

```
sales_bedrooms <- sales[!is.na(sales$bedrooms) &  sales$price < 0.3*10^7 & sales$full_baths<8,]
box_bedrooms <- sales_bedrooms %>%
  ggplot(aes(x = bedrooms, y = price, fill = class)) +
  geom_boxplot()
box_bedrooms
```




```
commercial <- sales[sales$class=="Commercial",]
table(commercial$bedrooms)/49 *100
```

```
##
##         0         2
## 87.755102  8.163265
```

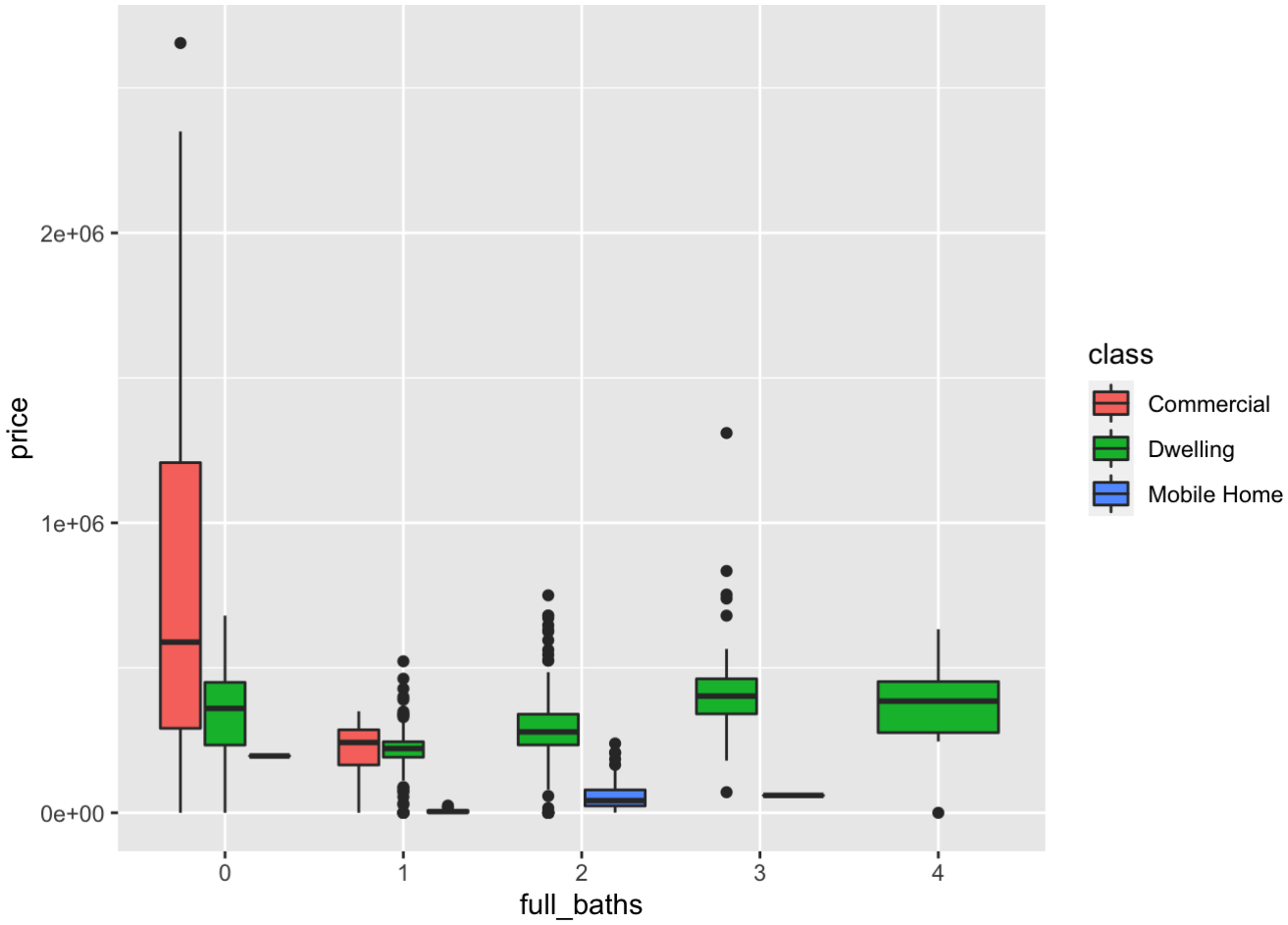
```
table(commercial$full_baths)/49 *100
```

```
##
##         0         1
## 87.755102  8.163265
```

```
dwelling <- sales[sales$class=="Dwelling",]
table(dwelling$full_baths)
```

```
##
##   0   1   2   3   4   8
##  19 192 445  74  12   1
```

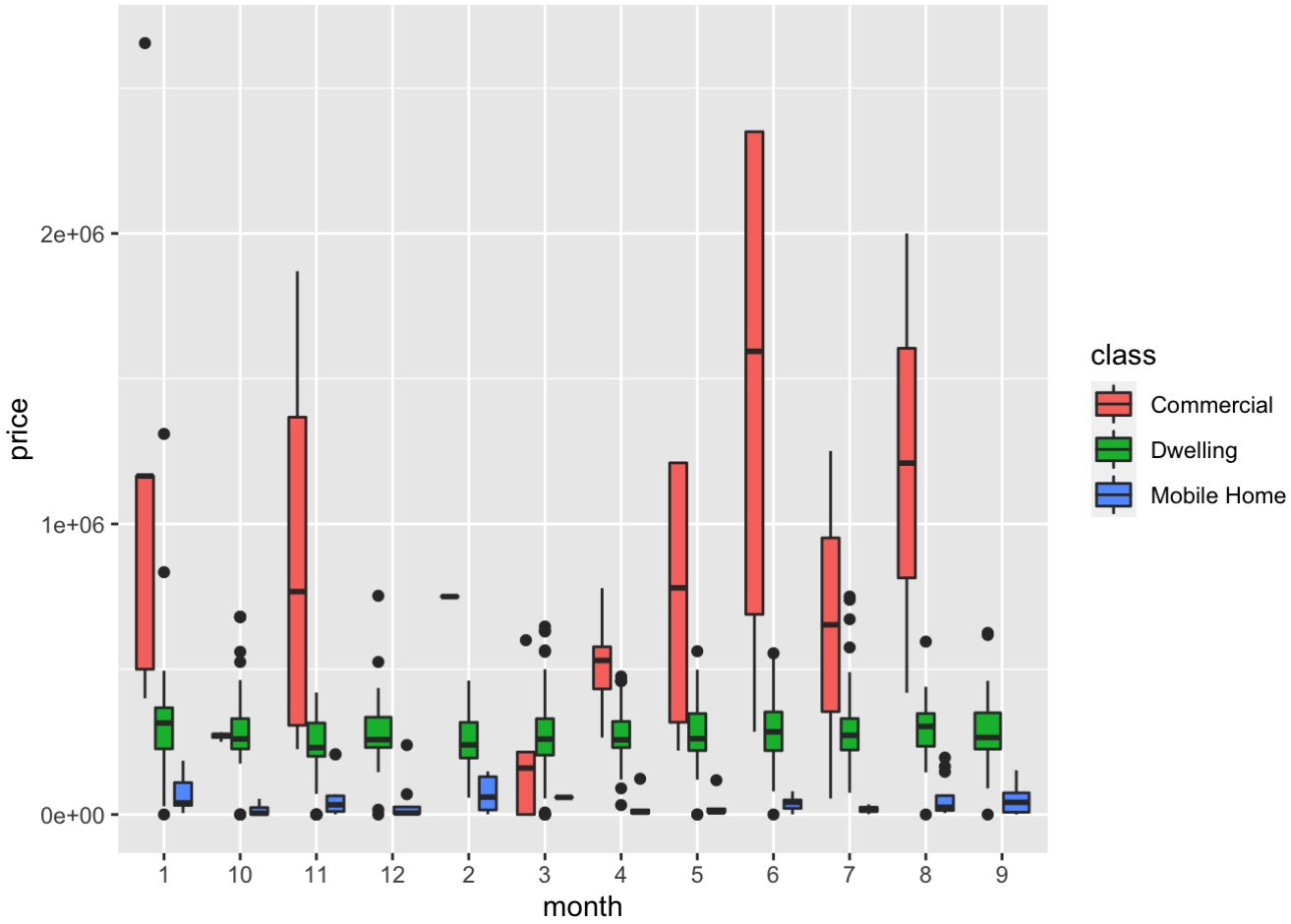
```
sales_full_baths <- sales[!is.na(sales$full_baths) &  sales$price < 0.3*10^7 & sales$full_baths
<8,]
box_full_baths <- sales_full_baths %>%
  ggplot(aes(x = full_baths, y = price, fill = class)) +
  geom_boxplot()
box_full_baths
```




```
sales_month <- sales[!is.na(sales$month) &  sales$price < 0.3*10^7 & sales$full_baths<8,]
table(sales_month$month)
```

```
##
##   1  10  11  12   2   3   4   5   6   7   8   9
##  46  74  61  52  43  72  74  82 104  93  94  91
```

```
#sales_month$month <- factor(sales_month$month, levels=c('1', '2', '3', '4', '5', '6','7', '8',
'9', '10', '11', '12'))
#sales_month <- sales_month[order(levels(sales_month$month)),]
box_month <- sales_month %>%
  ggplot(aes(x = month, y = price, fill = class)) +
  geom_boxplot()
box_month
```



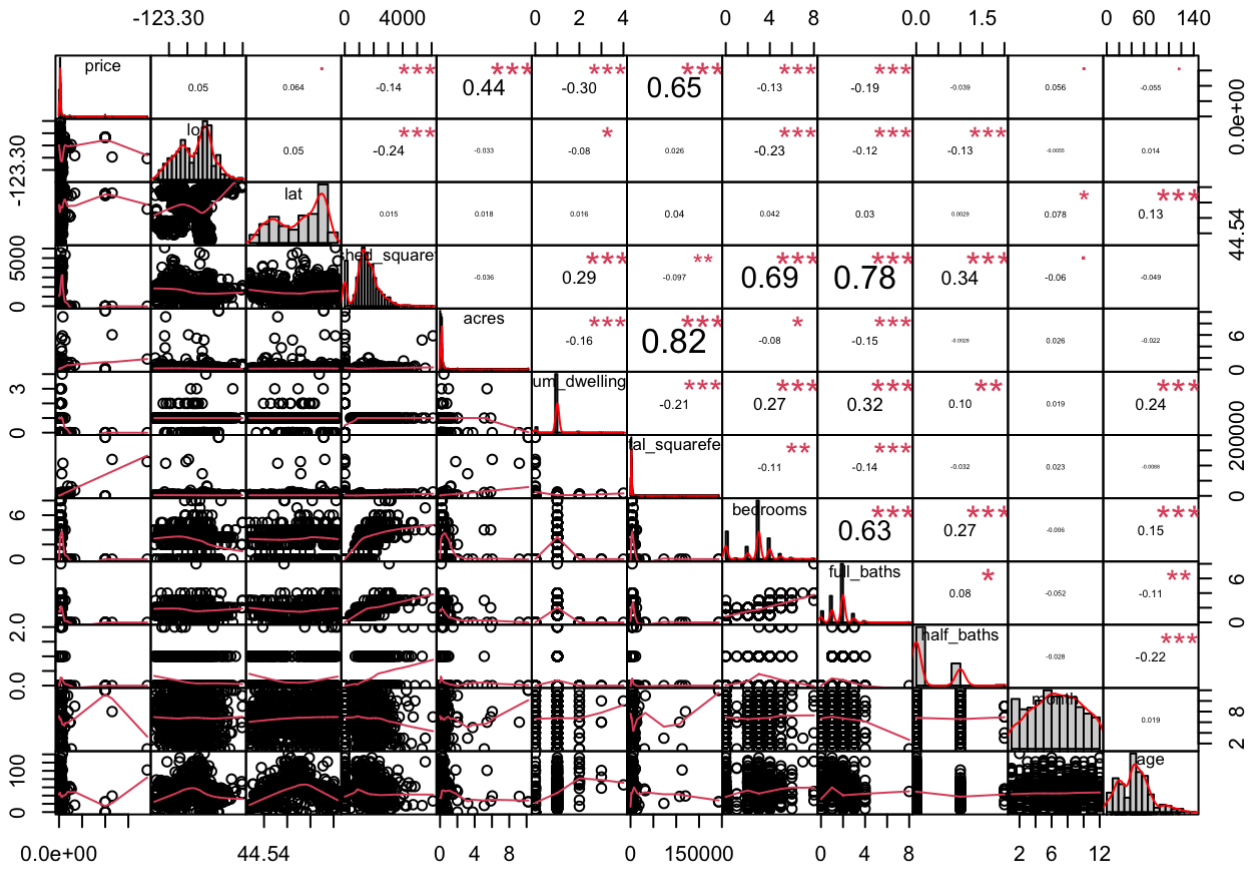
Observations:

-Median acres and price of commercial housing is the highest, followed by dwelling and mobile homes. - Dwellings have between 0-8 bedrooms. About 92% of the commercial buildings and all the mobile homes don't have any bedrooms or bathrooms. - For buildings without bedrooms and bathrooms, commercial buildings have the highest price, followed by dwellings and mobile homes. - However, there is no clear correlation between the number of bedrooms and full bathrooms, and the price of the building.

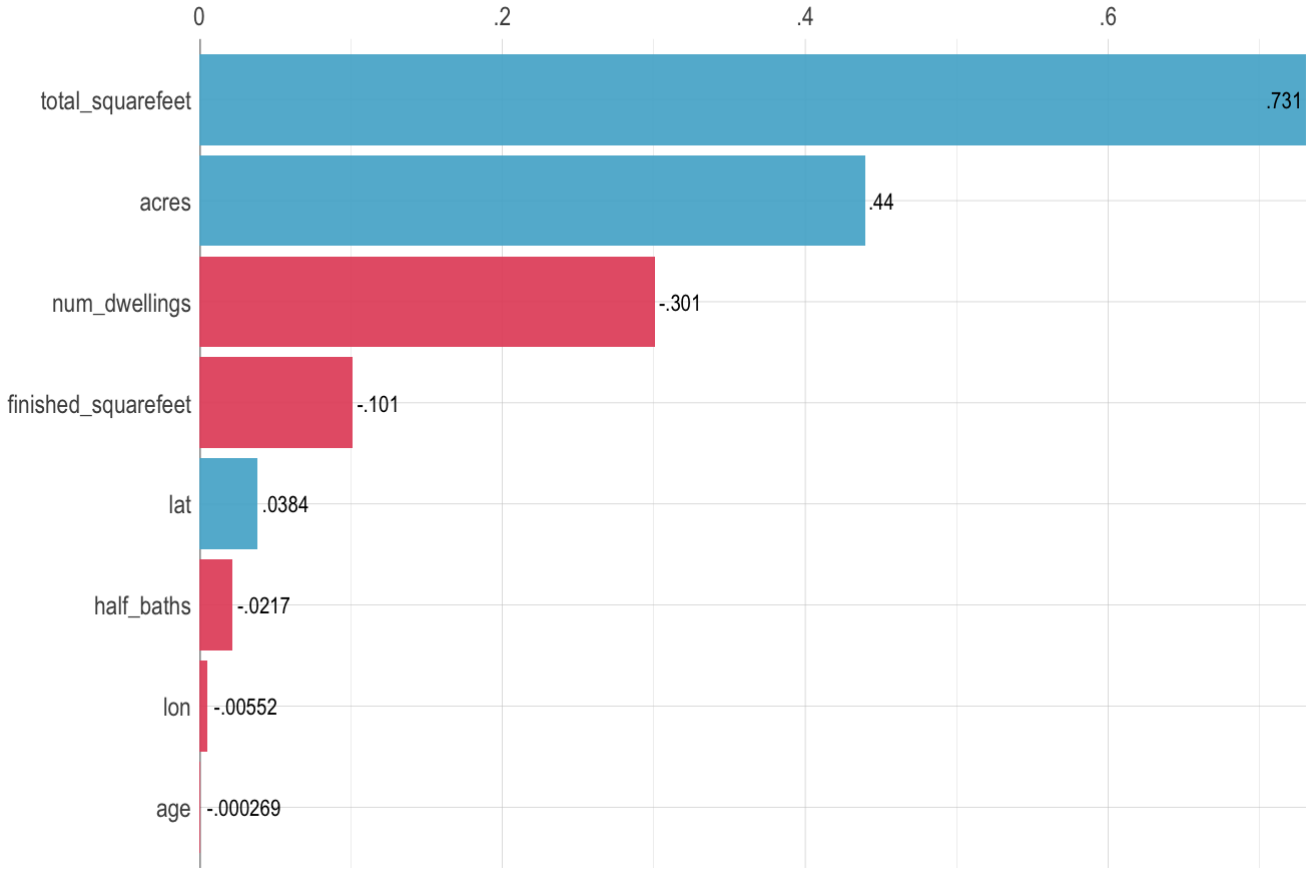
- The price of commercial buildings appear to have a seasonal trend.

Correlation plots:

Price is strongly correlated with acres and total squarefoot of the property.



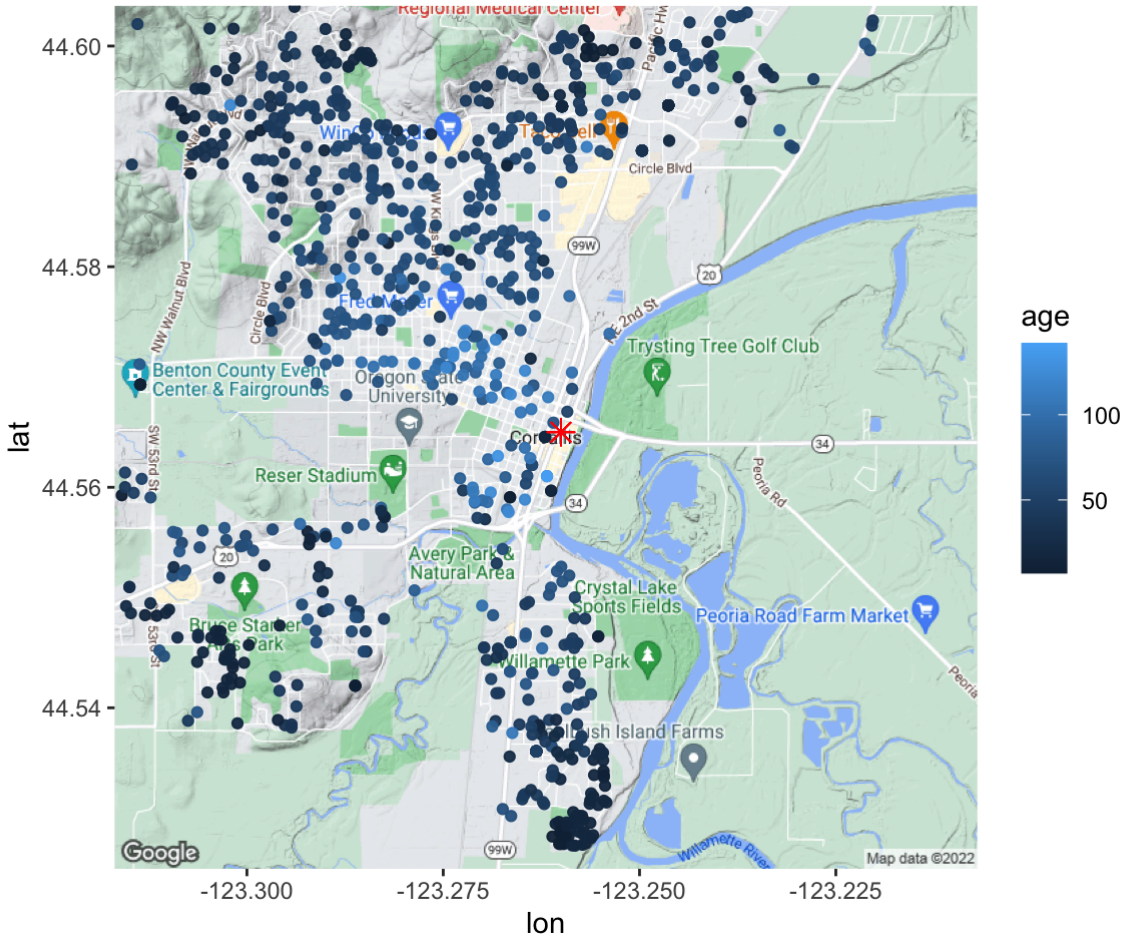
Correlations of price



Observation:

-Price is strongly correlated with acres and total squarefoot of the property.

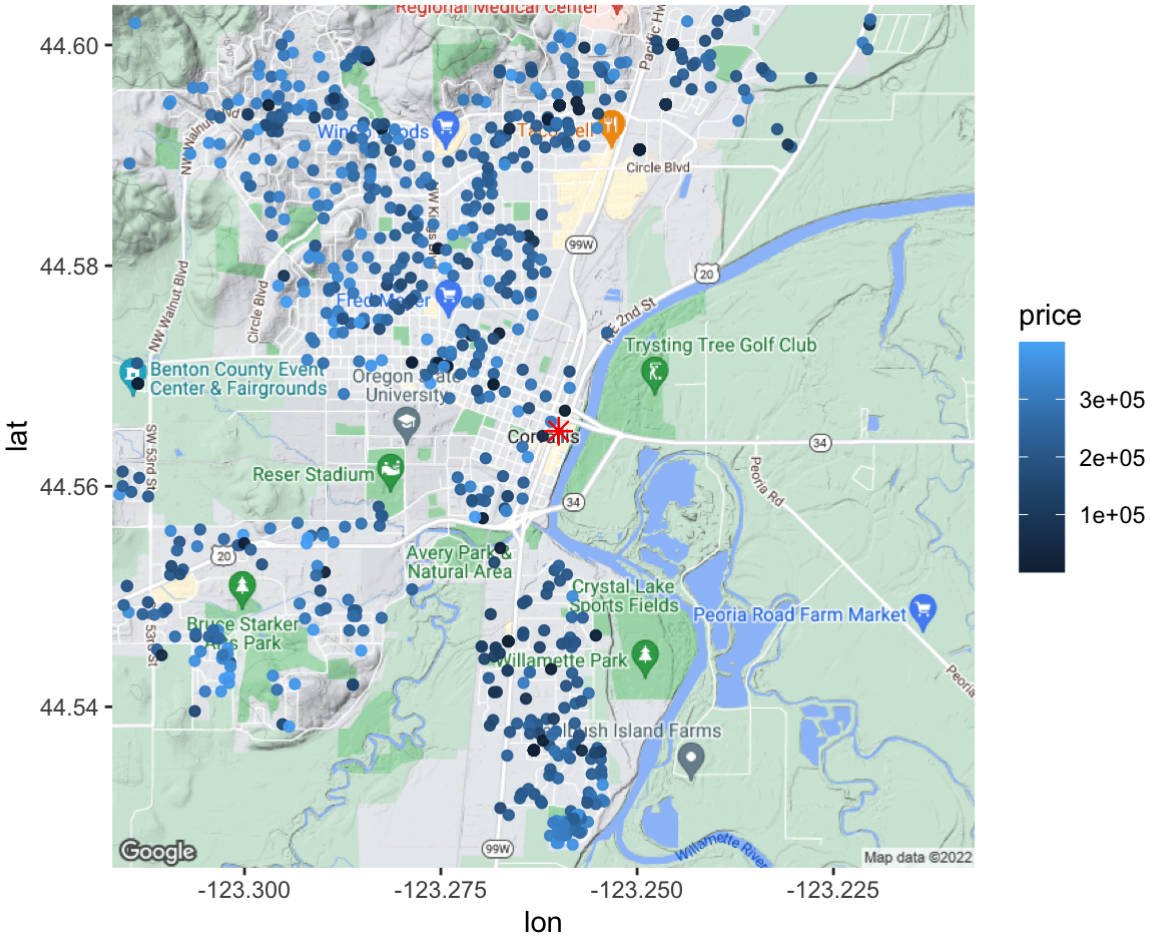
Houses color coded by acres



Observations:

The acres of land for the houses near the city center are smaller near the center, and increases in general as one goes outwards.

Houses color coded by price of the house



Observation: - The house prices tend to be higher in the north and west dirstions as compared to the southern part of the city.

Houses color coded by age of the house on a toner map:

```
# Add source and maptype to get toner map from Stamen Maps
corvallis_map_bw <- get_map(corvallis, zoom = 13,source = "stamen",maptype="toner")
```

```
## Source : https://maps.googleapis.com/maps/api/staticmap?center=44.5646,-123.262&zoom=13&size
=640x640&scale=2&maptype=terrain&key=xxx
```

```
## Source : http://tile.stamen.com/toner/13/1289/2959.png
```

```
## Source : http://tile.stamen.com/toner/13/1290/2959.png
```

```
## Source : http://tile.stamen.com/toner/13/1291/2959.png
```

```
## Source : http://tile.stamen.com/toner/13/1292/2959.png
```

```
## Source : http://tile.stamen.com/toner/13/1289/2960.png
```

```
## Source : http://tile.stamen.com/toner/13/1290/2960.png
```

```
## Source : http://tile.stamen.com/toner/13/1291/2960.png
```

```
## Source : http://tile.stamen.com/toner/13/1292/2960.png
```

```
## Source : http://tile.stamen.com/toner/13/1289/2961.png
```

```
## Source : http://tile.stamen.com/toner/13/1290/2961.png
```

```
## Source : http://tile.stamen.com/toner/13/1291/2961.png
```

```
## Source : http://tile.stamen.com/toner/13/1292/2961.png
```

```
## Source : http://tile.stamen.com/toner/13/1289/2962.png
```

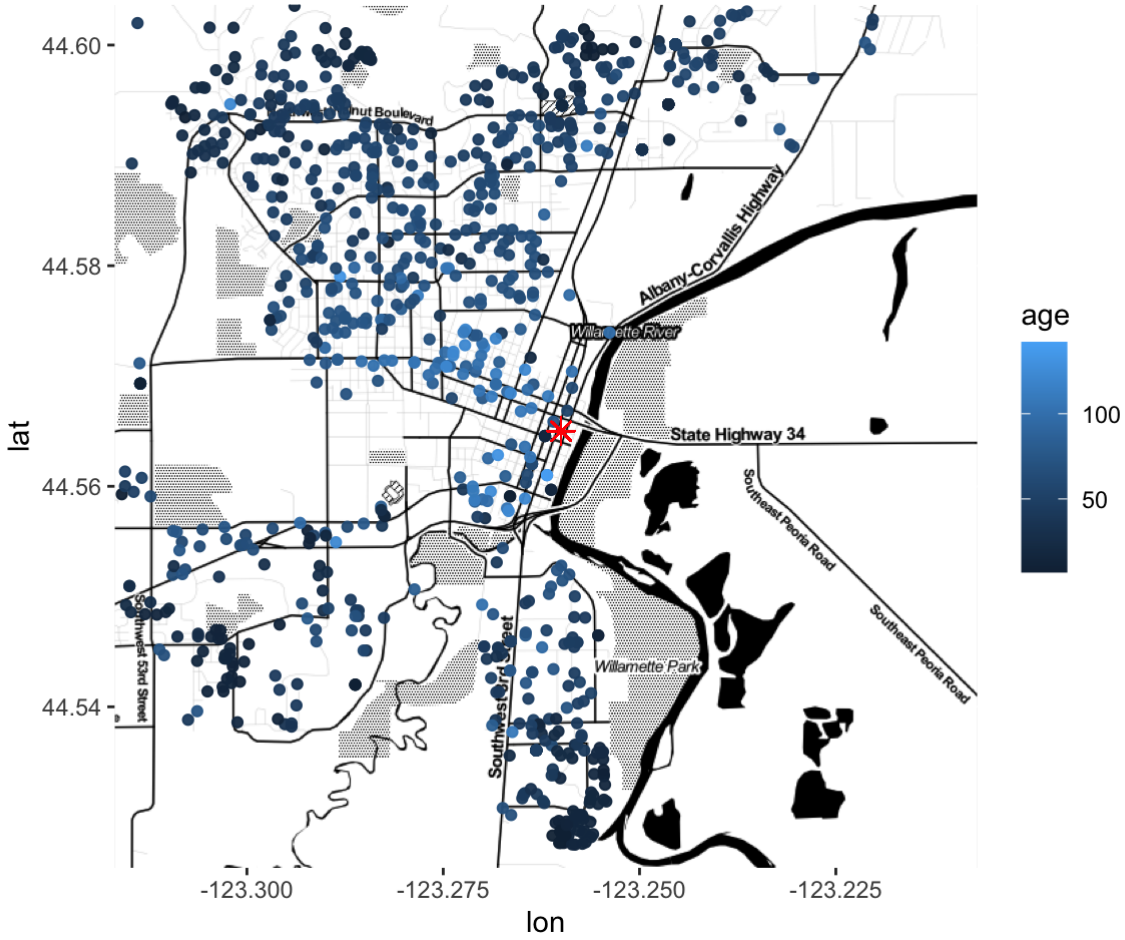
```
## Source : http://tile.stamen.com/toner/13/1290/2962.png
```

```
## Source : http://tile.stamen.com/toner/13/1291/2962.png
```

```
## Source : http://tile.stamen.com/toner/13/1292/2962.png
```

```
# Edit to display toner map
ggmap(corvallis_map_bw,
      base_layer = ggplot(sales, aes(lon, lat))) +
        geom_point(aes( color = age))+
        geom_point(aes(x= -123.260, y= 44.565),color= "red", shape = 8, size = 3)
```

```
## Warning in missing(base_layer) || base_layer == "auto": 'length(x) = 9 > 1' in
## coercion to 'logical(1)'
```



Observation:

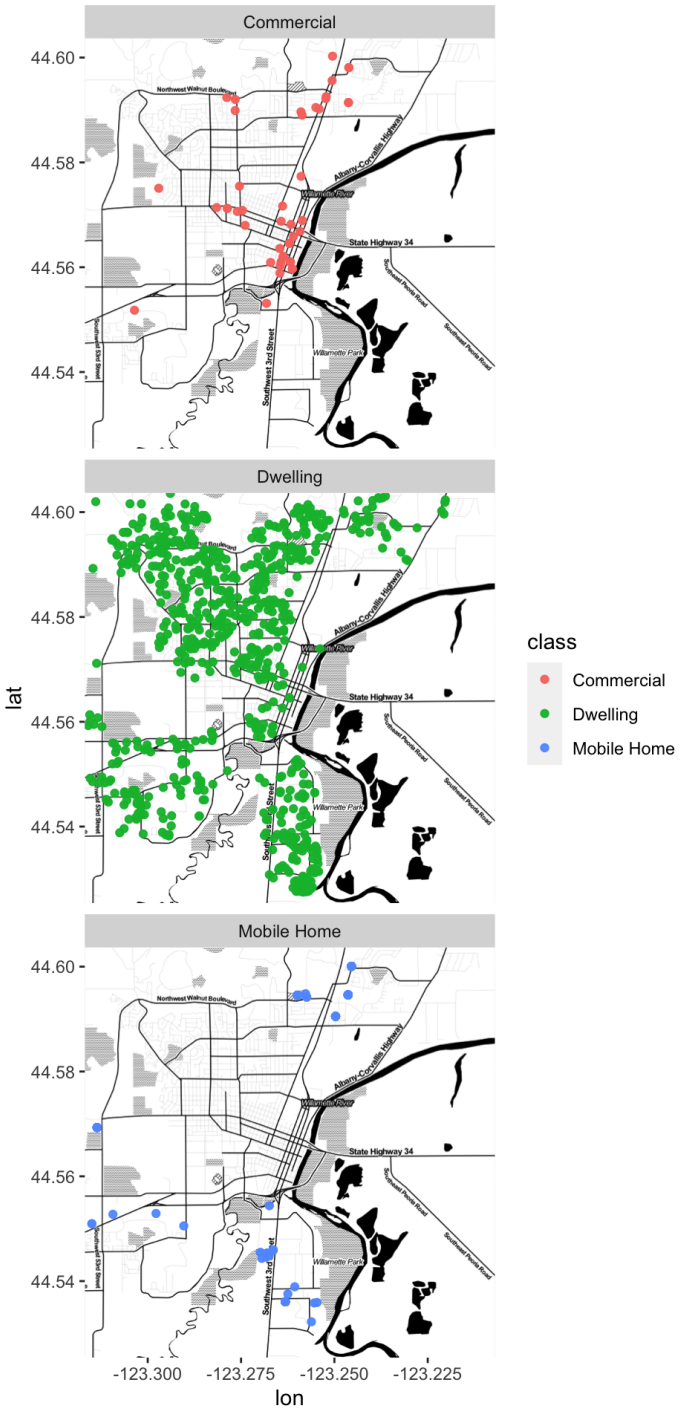
-The approximate center of the city has been marked by a red star. The oldest buildings in the city are near the city center. The city has grown ourwards in the past 142 years.

Houses color coded by the 'class' of the house on a toner map:

-Each class is shown is a separate map in different color.

```
# Use base_layer argument to ggmap() and add facet_wrap()

sales_class <- sales[!is.na(sales$class) & !sales$class == "RES Feature",]
ggmap(corvallis_map_bw,
      base_layer = ggplot(sales_class, aes(lon, lat))) +
        geom_point(aes( color = class))+
        facet_wrap(~class, ,  dir = "v")
```

```
## Warning in missing(base_layer) || base_layer == "auto": 'length(x) = 9 > 1' in
## coercion to 'logical(1)'
```



Observaions: -The dwellings are everywhere. -Mobile homes are mostly away from the city center. -Commercial building cluster around the city center and the northern side of the city.

House sales on each month in the year of 2015

```
glimpse(sales)
```

```
## Rows: 892
## Columns: 15
## $ price               <dbl> 267500, 255000, 295000, 5000, 13950, 233000, 24500…
## $ lon                 <dbl> -123.2803, -123.2330, -123.2635, -123.2599, -123.2…
## $ lat                 <dbl> 44.57808, 44.59718, 44.56923, 44.59453, 44.53606, …
## $ finished_squarefeet <int> 1520, 1665, 1440, 784, 1344, 1567, 1174, 912, 1404…
## $ acres               <dbl> 0.18, 0.13, 0.12, 0.00, 0.00, 0.04, 0.23, 0.22, 0.…
## $ num_dwellings       <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,…
## $ class               <chr> "Dwelling", "Dwelling", "Dwelling", "Mobile Home",…
## $ condition           <fct> AV, AV, G, F, AV, AV, G, AV, AV, G, AV, G, AV, G, …
## $ total_squarefeet    <int> 2192, 2193, 1860, 784, 1344, 2007, 1678, 1440, 160…
## $ bedrooms            <chr> "5", "3", "3", "0", "0", "3", "3", "3", "3", "3", …
## $ full_baths          <chr> "2", "2", "2", "1", "2", "2", "1", "1", "2", "1", …
## $ half_baths          <int> 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1,…
## $ month               <chr> "12", "12", "12", "12", "12", "12", "12", "12", "1…
## $ zip_split           <fct> 97330, 97330, 97330, 97330, 97333, 97330, 97330, 9…
## $ age                 <dbl> 55, 32, 74, 44, 43, 20, 50, 52, 20, 63, 24, 51, 24…
```

```
table(sales$bedrooms)
```

```
##
##   0   1   2   3   4   5   6   7   8
## 165   7  90 425 151  38  10   1   5
```

```
sales_bedrooms6 <- sales[sales$bedrooms < 6 & !is.na(sales$class) & !sales$class =="RES Featur
e",]
table(sales_bedrooms6$class)
```

```
##
##  Commercial    Dwelling Mobile Home
##          47         727         102
```

```
# Plot house sales using qmplot()
qmplot(lon, lat, data = sales,
       geom = "point", color = month) +
        facet_wrap(~ month)
```

```
## Using zoom = 13...
```

```
## Source : http://tile.stamen.com/terrain/13/1289/2959.png
```

```
## Source : http://tile.stamen.com/terrain/13/1290/2959.png
```

```
## Source : http://tile.stamen.com/terrain/13/1291/2959.png
```

```
## Source : http://tile.stamen.com/terrain/13/1292/2959.png
```

```
## Source : http://tile.stamen.com/terrain/13/1289/2960.png
```

```
## Source : http://tile.stamen.com/terrain/13/1290/2960.png
```

```
## Source : http://tile.stamen.com/terrain/13/1291/2960.png
```

```
## Source : http://tile.stamen.com/terrain/13/1292/2960.png
```

```
## Source : http://tile.stamen.com/terrain/13/1289/2961.png
```

```
## Source : http://tile.stamen.com/terrain/13/1290/2961.png
```

```
## Source : http://tile.stamen.com/terrain/13/1291/2961.png
```

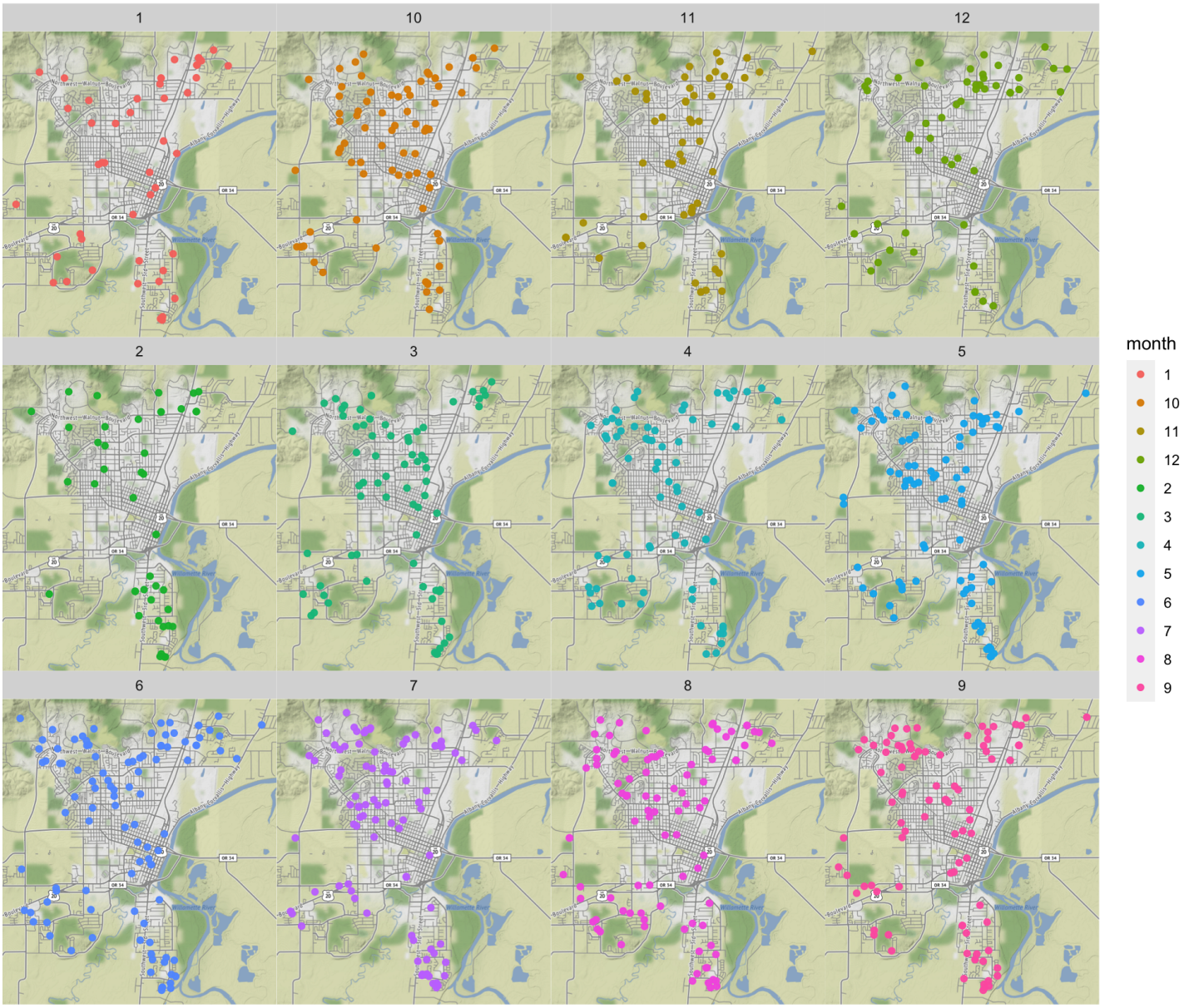
```
## Source : http://tile.stamen.com/terrain/13/1292/2961.png
```

```
## Source : http://tile.stamen.com/terrain/13/1289/2962.png
```

```
## Source : http://tile.stamen.com/terrain/13/1290/2962.png
```

```
## Source : http://tile.stamen.com/terrain/13/1291/2962.png
```

```
## Source : http://tile.stamen.com/terrain/13/1292/2962.png
```




```
table(sales$month)
```

```
##
##   1  10  11  12   2   3   4   5   6   7   8   9
##  48  74  61  54  43  72  74  82 104  93  95  92
```

Observation:

-House sales are low in the month of December (12), January (1) and February (2). -House sales are highest in the month of June, followed by August and July.

Add a point layer with color mapped to ward

```
# Add a point layer with color mapped to ward

head(ward_sales)
```

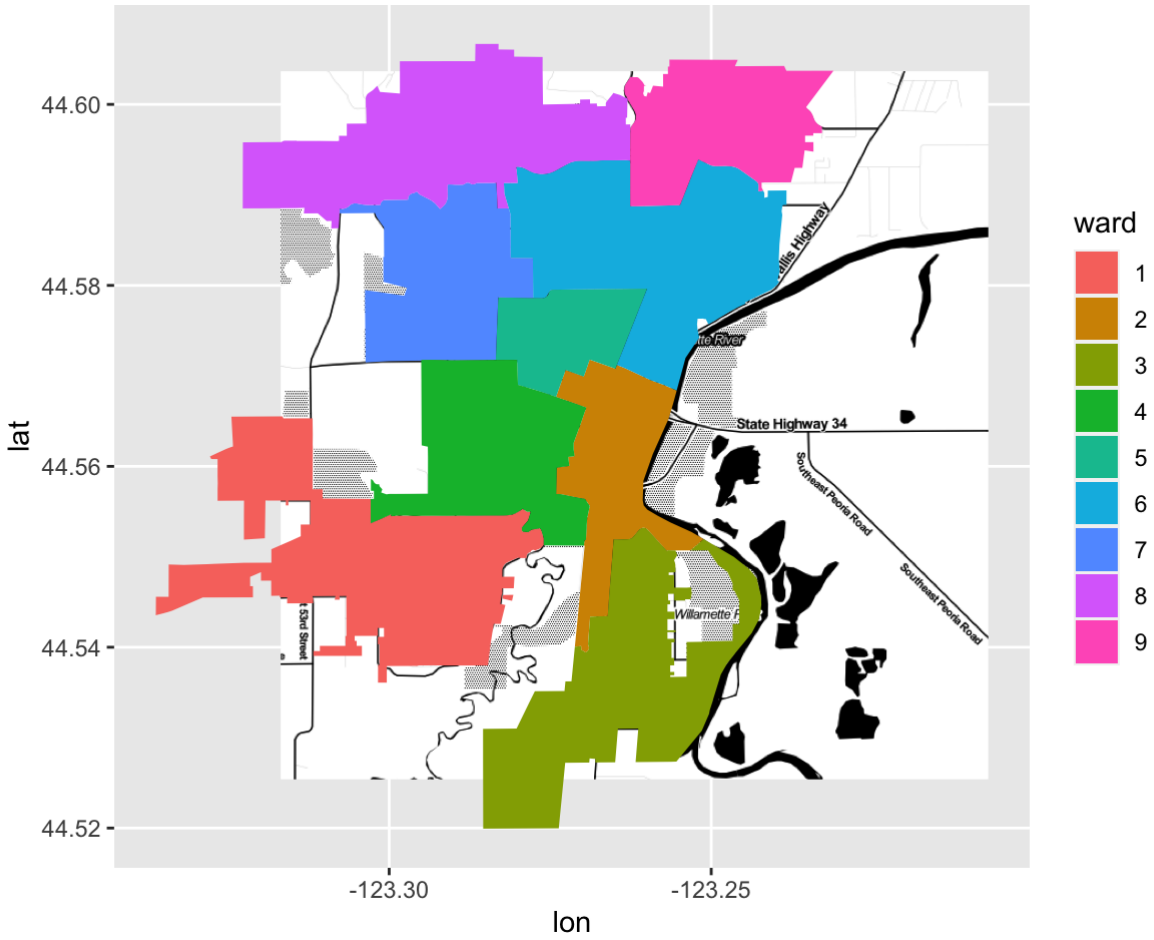
```
##   ward       lon      lat group order num_sales avg_price
## 1    1 -123.3128 44.56531   0.1     1       159  311626.9
## 2    1 -123.3122 44.56531   0.1     2       159  311626.9
## 3    1 -123.3121 44.56531   0.1     3       159  311626.9
## 4    1 -123.3119 44.56531   0.1     4       159  311626.9
## 5    1 -123.3119 44.56485   0.1     5       159  311626.9
## 6    1 -123.3119 44.56430   0.1     6       159  311626.9
##   avg_finished_squarefeet
## 1                1609.226
## 2                1609.226
## 3                1609.226
## 4                1609.226
## 5                1609.226
## 6                1609.226
```

```
levels(ward_sales$ward)
```

```
## [1] "1" "2" "3" "4" "5" "6" "7" "8" "9"
```

```
# Add a polygon layer with fill mapped to ward, and group to group
# Fix the polygon cropping

ggmap(corvallis_map_bw,
      base_layer = ggplot(ward_sales, aes(lon, lat)),extent = "normal", maprange = FALSE) +
        geom_polygon(aes(group = group, fill = ward))
```

```
## Warning in missing(base_layer) || base_layer == "auto": 'length(x) = 9 > 1' in
## coercion to 'logical(1)'
```

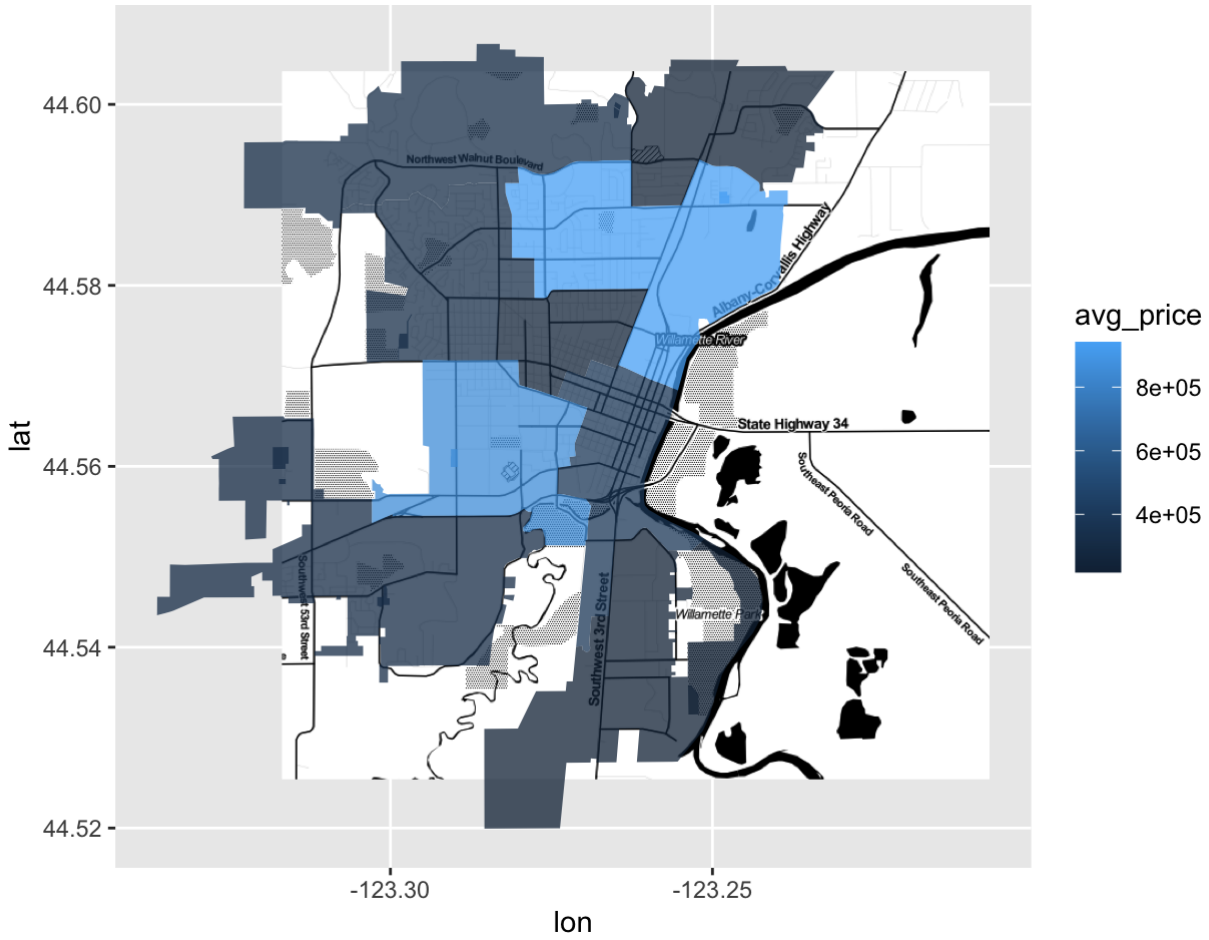


Observation: Each ward is shown in a different color. The average prices in each ward has been plotted below.

Average prices in each ward

```
# Repeat again, but map fill to avg_price
ggmap(corvallis_map_bw,
      base_layer = ggplot(ward_sales, aes(lon, lat)),
      extent = "normal", maprange = FALSE) +
        geom_polygon(aes(group = group, fill = avg_price), alpha= 0.75)
```

```
## Warning in missing(base_layer) || base_layer == "auto": 'length(x) = 9 > 1' in
## coercion to 'logical(1)'
```



Observations: -Ward 6 and 4 (shown in light blue) have much higher average prices thant the rest of the city.

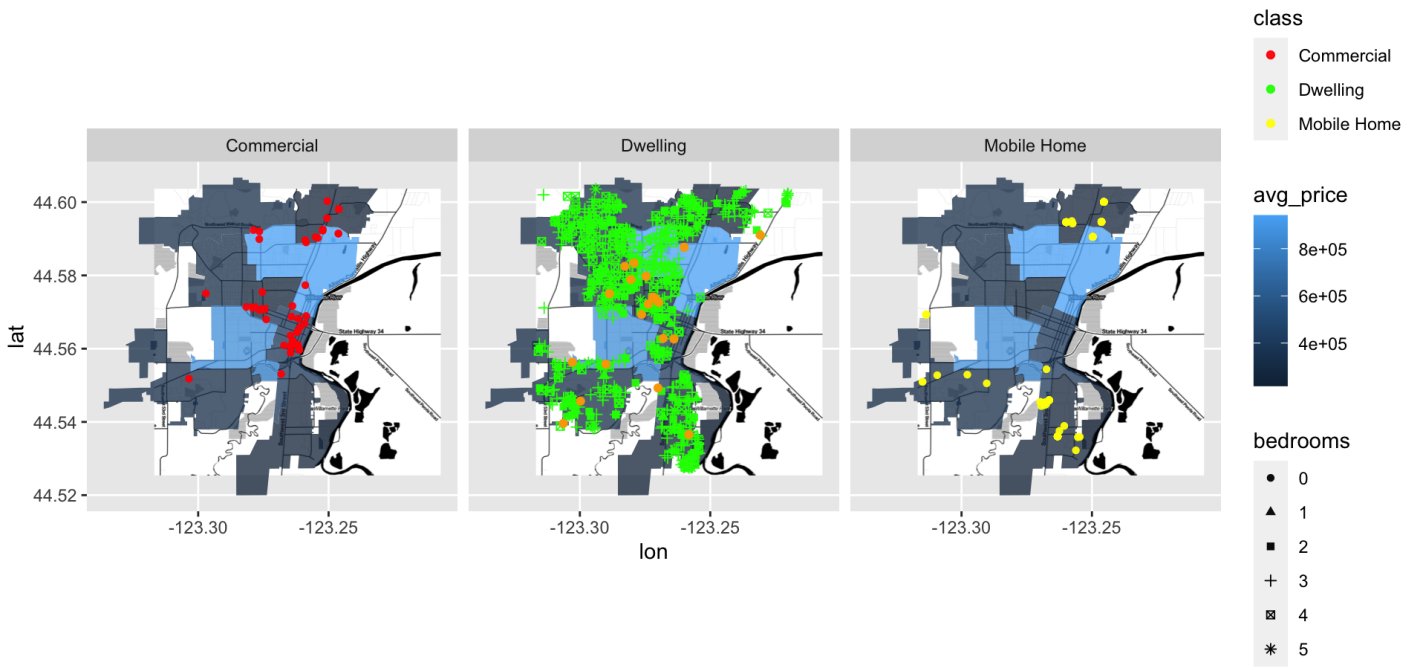
Overlay of the average price of each ward with different types of buildings:

```
sales_bedrooms0 <- sales[sales$bedrooms == 0 & sales$class == "Dwelling" & !is.na(sales$clas
s),]
table(sales_bedrooms6$class)
```

```
##
##  Commercial    Dwelling Mobile Home
##          47         727         102
```

```
ggmap(corvallis_map_bw,
      base_layer = ggplot(ward_sales, aes(lon, lat)),
      extent = "normal", maprange = FALSE) +
        geom_polygon(aes(group = group, fill = avg_price), alpha= 0.75)+
  geom_point(data= sales_bedrooms6 ,aes(color = class, shape=bedrooms))+
  scale_color_manual(values=c("red","green","yellow"))+
        facet_wrap(~class)+
   geom_point(data= sales_bedrooms0 ,color = "orange")
```

```
## Warning in missing(base_layer) || base_layer == "auto": 'length(x) = 9 > 1' in
## coercion to 'logical(1)'
```

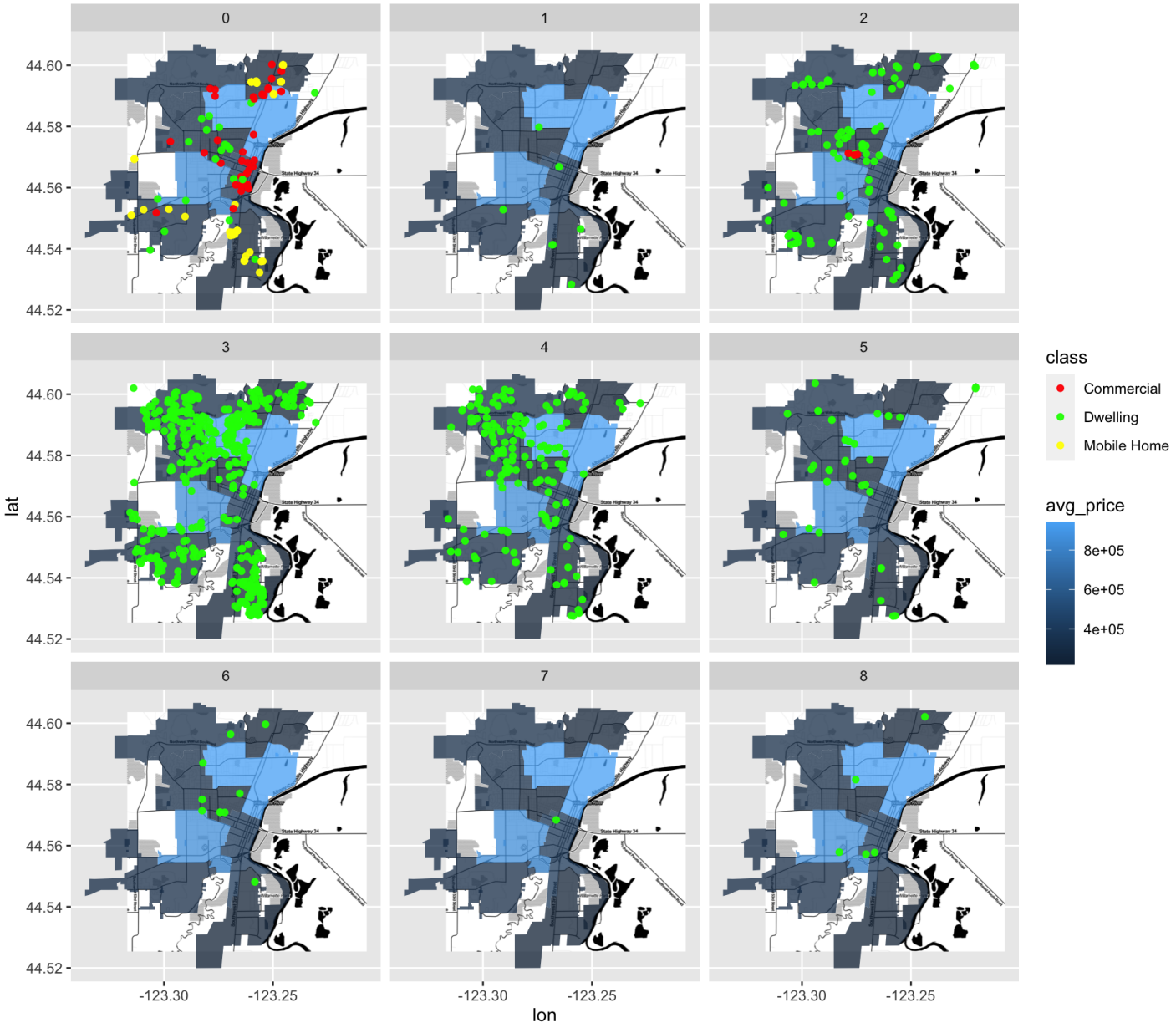


Overlay of the average price of each ward with different types of buildings and number of bedrooms:

-The shades of blue show the average price of the house/ward. -The colors of the dots show different types of buildings. -The different plots show the number of bedrooms.

```
# Repeat again, but map fill to avg_price
ggmap(corvallis_map_bw,
      base_layer = ggplot(ward_sales, aes(lon, lat)),
      extent = "normal", maprange = FALSE) +
        geom_polygon(aes(group = group, fill = avg_price), alpha= 0.75)+
  geom_point(data= sales,aes(color = class))+
   scale_color_manual(values=c("red", "green", "yellow"))+
        facet_wrap(~bedrooms)
```

```
## Warning in missing(base_layer) || base_layer == "auto": 'length(x) = 9 > 1' in
## coercion to 'logical(1)'
```



Observations: -Most of the buildings of all casses are in wards with lower housing price. -Most of the mobiles homes and commercial buildings have no bedroom as expected. -Most of the dwellings have 3 bedrooms, followed by 4.

##Linear regression We will perform a quick linear regression to get a sense of the data. This is by no means comprehensive.

```
sales_lm <- sales %>%
  filter(!is.na(age)) %>%
  filter(!is.na(acres))

lmPrice <- lm(price ~ lon + lat + acres + total_squarefeet +age +factor(class) +factor(bedroom
s)+factor(full_baths) +factor(month), data = sales_lm)
summary(lmPrice)
```

```
##
## Call:
## lm(formula = price ~ lon + lat + acres + total_squarefeet + age +
##     factor(class) + factor(bedrooms) + factor(full_baths) + factor(month),
##     data = sales_lm)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4790500   -78658    -3156    64083  8667458
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -1.500e+08  1.148e+08  -1.307  0.19161
## lon                      -1.091e+06  8.867e+05  -1.231  0.21879
## lat                       3.608e+05  7.061e+05   0.511  0.60946
## acres                    -5.658e+05  4.288e+04 -13.195  < 2e-16 ***
## total_squarefeet          9.487e+01  2.857e+00  33.202  < 2e-16 ***
## age                       1.069e+03  7.215e+02   1.482  0.13879
## factor(class)Dwelling    -3.650e+04  1.192e+05  -0.306  0.75948
## factor(class)Mobile Home -3.652e+05  3.484e+05  -1.048  0.29477
## factor(bedrooms)1        -1.591e+05  3.862e+05  -0.412  0.68039
## factor(bedrooms)2        -1.349e+05  3.450e+05  -0.391  0.69588
## factor(bedrooms)3        -1.008e+05  3.427e+05  -0.294  0.76878
## factor(bedrooms)4        -6.933e+04  3.445e+05  -0.201  0.84054
## factor(bedrooms)5        -1.590e+05  3.511e+05  -0.453  0.65079
## factor(bedrooms)6        -2.393e+05  3.774e+05  -0.634  0.52616
## factor(bedrooms)7        -3.344e+05  6.137e+05  -0.545  0.58592
## factor(bedrooms)8        -3.029e+05  4.357e+05  -0.695  0.48718
## factor(full_baths)1      -1.283e+05  3.423e+05  -0.375  0.70795
## factor(full_baths)2      -1.026e+05  3.411e+05  -0.301  0.76352
## factor(full_baths)3      -2.280e+04  3.468e+05  -0.066  0.94759
## factor(full_baths)4      -1.655e+04  3.838e+05  -0.043  0.96561
## factor(full_baths)8      -1.966e+05  6.471e+05  -0.304  0.76138
## factor(month)10          -2.848e+05  8.991e+04  -3.168  0.00159 **
## factor(month)11          -2.780e+05  9.319e+04  -2.984  0.00293 **
## factor(month)12          -3.046e+05  9.644e+04  -3.158  0.00164 **
## factor(month)2           -2.854e+05  1.009e+05  -2.829  0.00478 **
## factor(month)3           -3.702e+05  9.052e+04  -4.089 4.73e-05 ***
## factor(month)4           -2.838e+05  8.972e+04  -3.164  0.00161 **
## factor(month)5           -2.643e+05  8.923e+04  -2.961  0.00315 **
## factor(month)6           -2.164e+05  8.421e+04  -2.569  0.01035 *
## factor(month)7           -2.303e+05  8.630e+04  -2.668  0.00777 **
## factor(month)8           -1.981e+05  8.599e+04  -2.304  0.02148 *
## factor(month)9           -2.454e+05  8.636e+04  -2.842  0.00460 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 471800 on 855 degrees of freedom
## Multiple R-squared:  0.6997, Adjusted R-squared:  0.6888
## F-statistic: 64.26 on 31 and 855 DF,  p-value: < 2.2e-16
```

Observations: -Adjusted R-squared is 0.69. Hence about 69% of the data can be explained by this model. Acres, total squarefeet and months have P(t) < 0.05. Hence they have significant contribution towards the housing price.

```
fit <- lm(sales_lm$price ~ ., data = sales_lm)
require(broom)
glance(fit)
```

```
## # A tibble: 1 × 12
##   r.squared adj.r…¹  sigma stati…²   p.value    df  logLik    AIC    BIC devia…³
##       <dbl>   <dbl>  <dbl>   <dbl>     <dbl> <dbl>   <dbl>  <dbl>  <dbl>   <dbl>
## 1     0.708   0.694 4.68e5    51.2 1.31e-196    40 -12818. 25720. 25921. 1.85e14
## # … with 2 more variables: df.residual <int>, nobs <int>, and abbreviated
## #   variable names ¹adj.r.squared, ²statistic, ³deviance
## # ℹ Use `colnames()` to see all variable names
```

##List of the R2 for each simple (one-predictor) regression

```
sales %>%
  dplyr::select(-price) %>%  # exclude outcome, leave only predictors
  purrr::map(~lm(sales$price ~ .x, data = sales)) %>%
  purrr::map(summary) %>%
        purrr::map_dbl("r.squared") %>%
  tidy %>%  #Tidy the result of a test into a summary data.frame (under broom library)
  dplyr::arrange(desc(x)) %>%
  rename(r.squared = x) -> r2s
```

```
## Warning: 'tidy.numeric' is deprecated.
## See help("Deprecated")
```

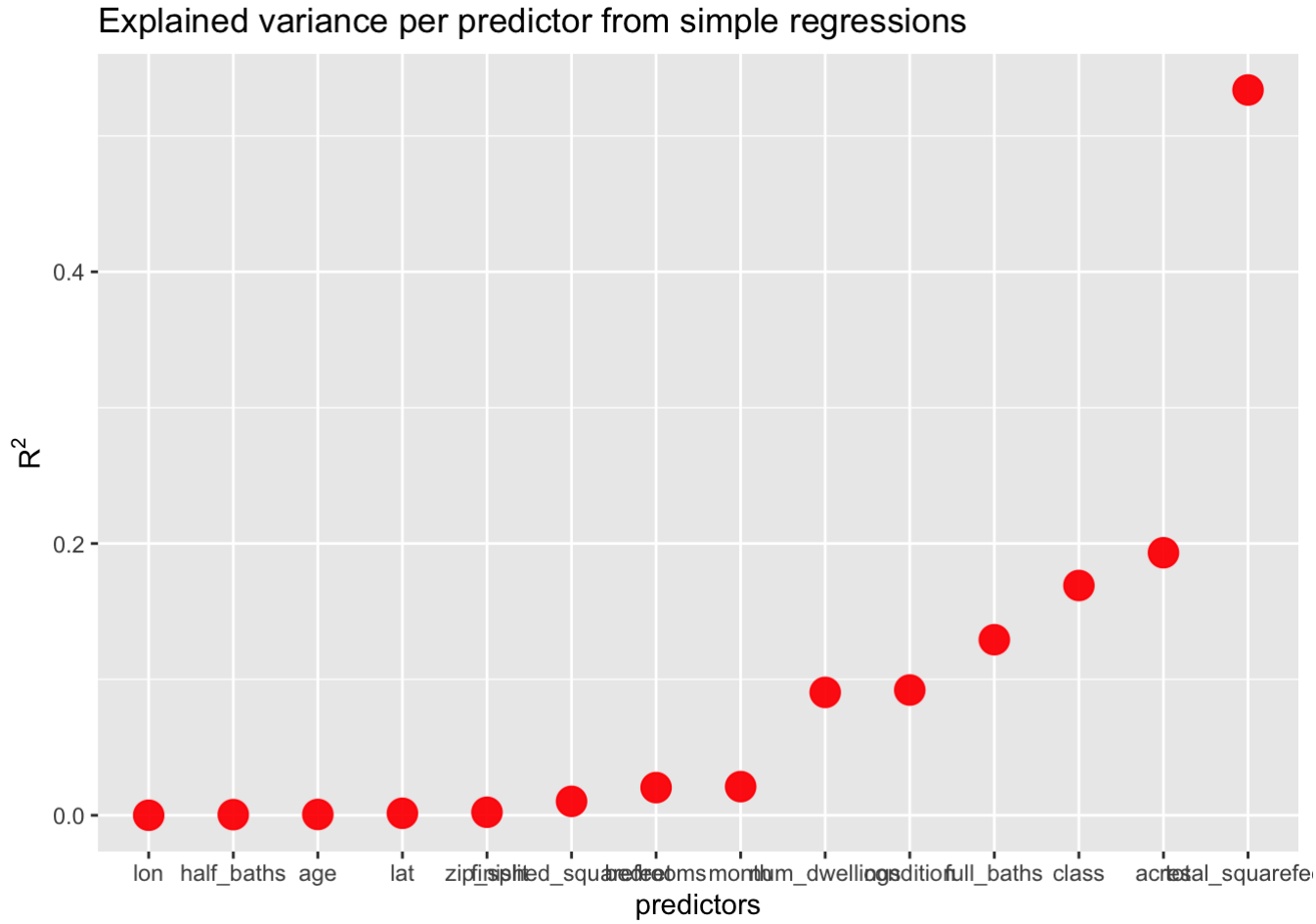
```
kable(r2s)
```

names	r.squared
total_squarefeet	0.5337959
acres	0.1932078
class	0.1691042
full_baths	0.1292119
condition	0.0921813
num_dwellings	0.0904421
month	0.0210803
bedrooms	0.0203717
finished_squarefeet	0.0102524
zip_split	0.0022259

names	r.squared
lat	0.0014768
age	0.0006117
half_baths	0.0004715
lon	0.0000305

##let's plot the resulting values (and sort the predictors by descending values).

```
ggplot(r2s, aes(x = reorder(names, r.squared), y = r.squared)) +
  geom_point(size = 5, color = "red") +
  ylab(expression(R^{2})) +
  xlab("predictors") +
  ggtitle("Explained variance per predictor from simple regressions")
```



Observation: Total squarefeet is the most important factor in explaining the price of the housing in the city, followed by acres and class.

Summary and recommendations:

We have graphically and statistically explored the various factors affecting the price of the housing in the city of Coravallis. -Adjusted R-squared is 0.69. Hence about 69% of the data can be explained by this model. Acres, total squarefeet and months have P(t) < 0.05. Hence they have significant contribution towards the housing price. -Further modifications must be built to fine-tune the model. -We must remove the multicollinearity between the features, check for VIF values, perform feature engineering (if necessary) etc. -We must also then check the assumptions of the linear model: No Heteroscedasticity, Mean of residuals should be 0, Linearity of variables, Normality of error terms. -For a predictive model, we must also set aside a test set separately so that we can check our model on this untouched test set.