Final Project-Housing

Yu Chieh Cheng $ He Jin Chu

2023-04-28

Our dataset is from Kaggle website, called Housing Prices Dataset.
https://www.kaggle.com/datasets/yasserh/housing-prices-dataset

```
library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse
 2.0.0 ──
## ✔ dplyr     1.1.0     ✔ readr     2.1.4
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ ggplot2   3.4.1     ✔ tibble    3.2.1
## ✔ lubridate 1.9.2     ✔ tidyr     1.3.0
## ✔ purrr     1.0.1
## ── Conflicts ────────────────────────────────────────── tidyverse_confl
icts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force
 all conflicts to become errors

library(readr)
library(dplyr)
library(broom)
Housing <- read_csv("Housing.csv")

## Rows: 545 Columns: 13
## ── Column specification ───────────────────────────────────────────────
──────────
## Delimiter: ","
## chr (7): mainroad, guestroom, basement, hotwaterheating, airconditionin
g, pr...
## dbl (6): price, area, bedrooms, bathrooms, stories, parking
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet th
is message.

Housing

## # A tibble: 545 × 13
##      price  area bedro…¹ bathr…² stories mainr…³ guest…⁴ basem…⁵ hotwa…
⁶ airco…⁷
##      <dbl> <dbl>   <dbl>   <dbl>   <dbl> <chr>   <chr>   <chr>   <chr>
  <chr>
```

```
## 1  1.33e7  7420       4       2       3 yes     no      no      no
  yes
## 2  1.23e7  8960       4       4       4 yes     no      no      no
  yes
## 3  1.23e7  9960       3       2       2 yes     no      yes     no
  no
## 4  1.22e7  7500       4       2       2 yes     no      yes     no
  yes
## 5  1.14e7  7420       4       1       2 yes     yes     yes     no
  yes
## 6  1.08e7  7500       3       3       1 yes     no      yes     no
  yes
## 7  1.01e7  8580       4       3       4 yes     no      no      no
  yes
## 8  1.01e7 16200       5       3       2 yes     no      no      no
  no
## 9  9.87e6  8100       4       1       2 yes     yes     yes     no
  yes
## 10  9.8 e6  5750       3       2       4 yes     yes     no      no
  yes
## # … with 535 more rows, 3 more variables: parking <dbl>, prefarea <chr>,
## #   furnishingstatus <chr>, and abbreviated variable names ¹bedrooms,
## #   ²bathrooms, ³mainroad, ⁴guestroom, ⁵basement, ⁶hotwaterheating,
## #   ⁷airconditioning
```

#1. Offer a preliminary description of the data set. For example, indicate the size of the data source, describe the variables, and include any other data profile information that would be of interest.

-> There are 545 rows and 13 columns included in this data set.

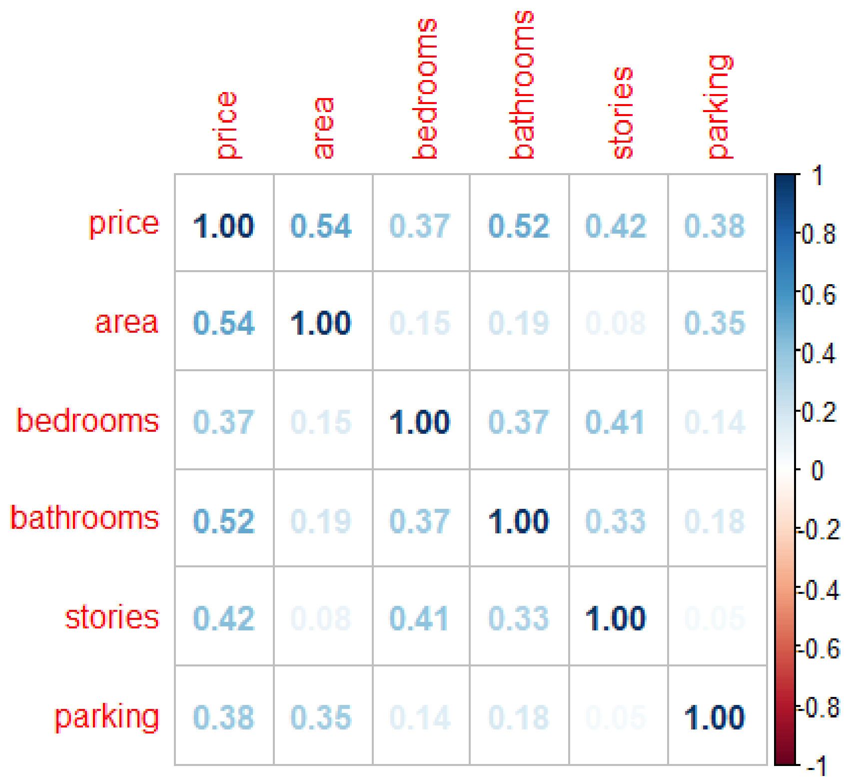
-> The dependent variable we want to look at is house price, and the independent variables we want to discuss is area, bedrooms, bathrooms, stories, hot water heating, airconditioning, parking and furnishing status. While hot water heating, airconditioning, and furnishing status are categorical variables, we change them into dummy variable to analyze these data.

```
dim(Housing)

## [1] 545  13

#choosing variables we would like to discuss
Housing%>%
  select(price,area,bedrooms,bathrooms,stories,hotwaterheating,airconditio
ning,parking,furnishingstatus)->Housing1
```

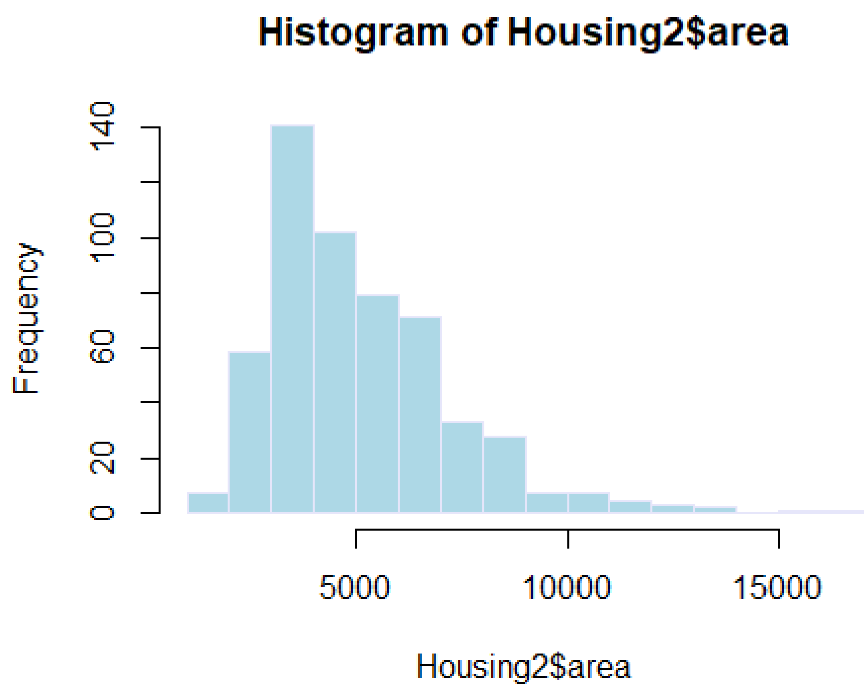
#2. Generate relevant data visual plots that explore multicollinearity for the quantitative variables and normality for the quantitative variables as well. Also, use R code to confirm the levels of the categorical variables.

-> The highest correlation is 0.54 between two independent variables, therefore there is no multicollinearity shown in our dataset.

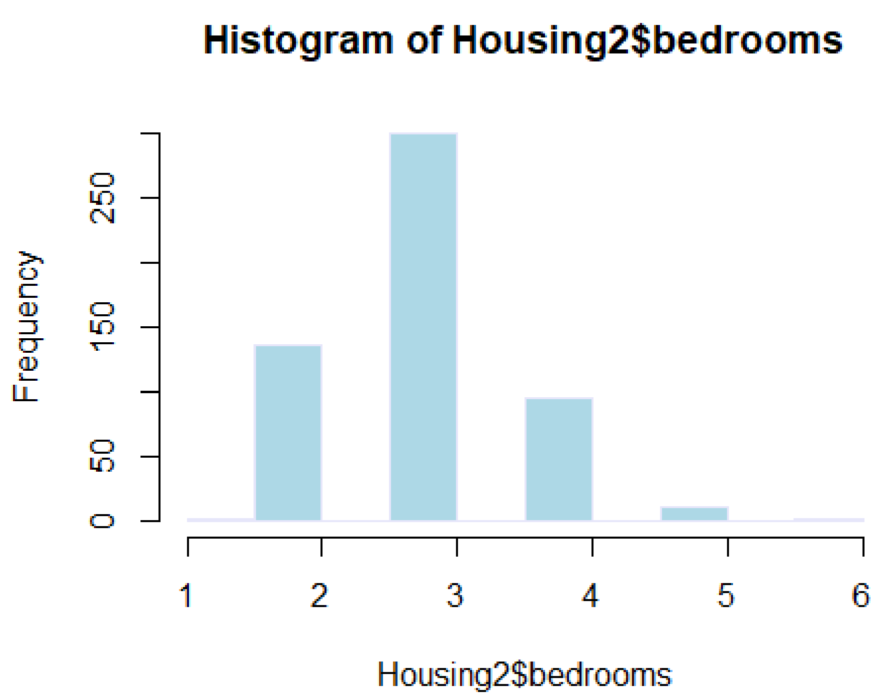
```
library(corrplot)

## corrplot 0.92 loaded

#choose the quantitative variables from dataset
Housing%>%
  select(price,area,bedrooms,bathrooms,stories,parking)->Housing2

#explore multicollinearity for the quantitative variables
corrplot(cor(Housing2), method = "number")
```



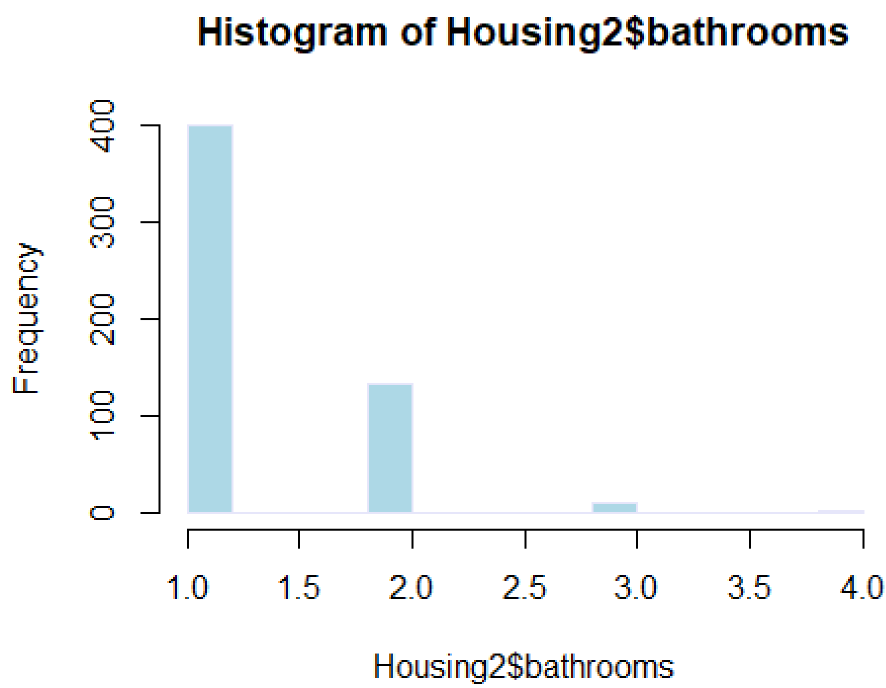

```
#normality for the quantitative variables
hist(Housing2$area, col = "lightblue", border = "lavender")
```



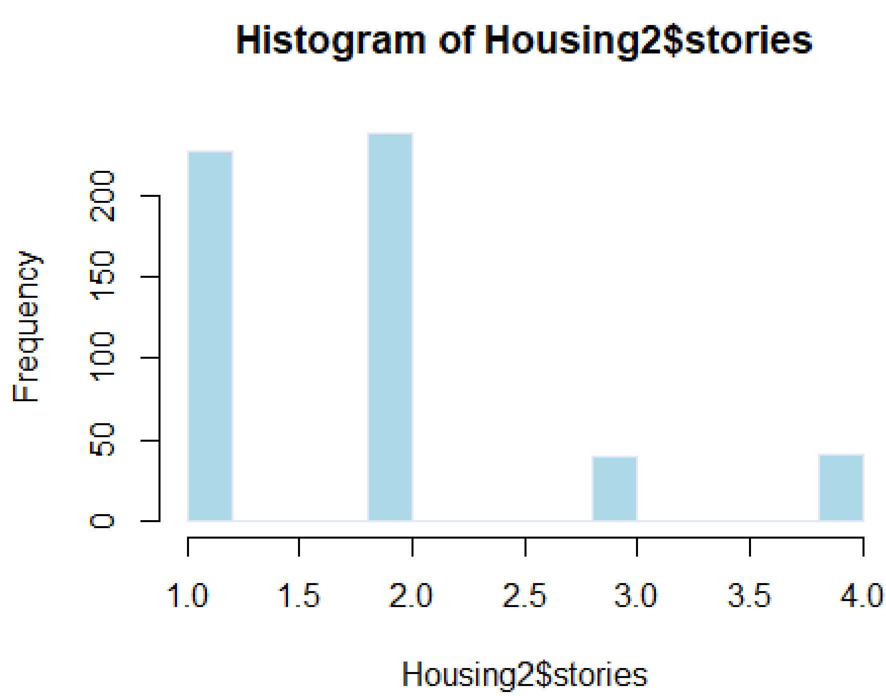

```
hist(Housing2$bedrooms, col = "lightblue", border = "lavender")
```



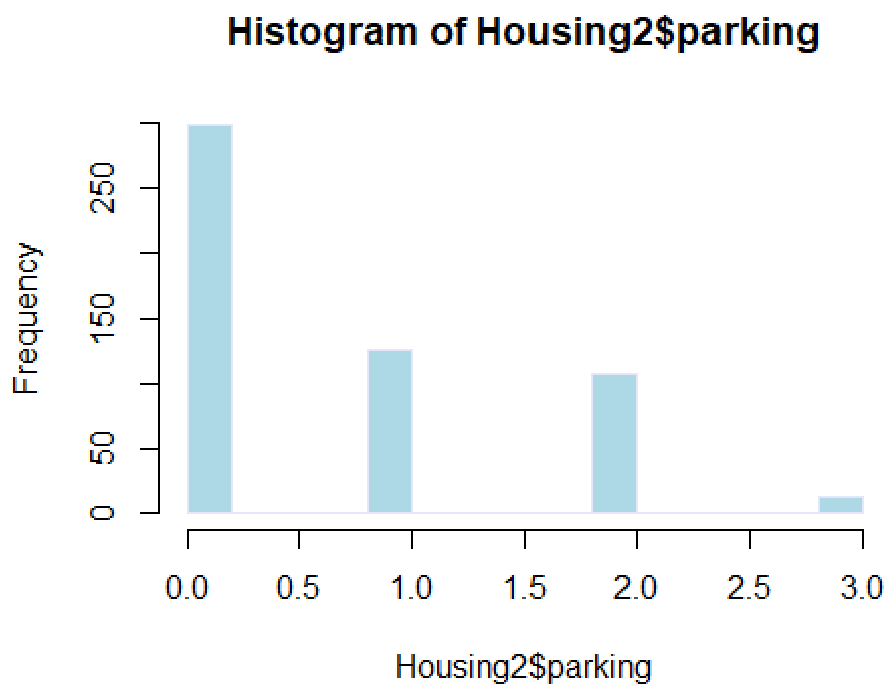

```
hist(Housing2$bathrooms, col = "lightblue", border = "lavender")
```




```
hist(Housing2$stories, col = "lightblue", border = "lavender")
```




```
hist(Housing2$parking, col = "lightblue", border = "lavender")
```




```
#the levels of the categorical variables
unique(Housing1$hotwaterheating)

## [1] "no"  "yes"

unique(Housing1$airconditioning)

## [1] "yes" "no"

unique(Housing1$furnishingstatus)

## [1] "furnished"      "semi-furnished" "unfurnished"
```

#3. Using R code, produce a full Regression Model that consists of quantitative and categorical variables. Make use of the R generated dummy variable matrices

-> Named each categorical variable into a column name and change each level into 0 or 1

```
#named each categorical variables into a column name and change each level
 into 0 or 1
hu <- model.matrix(~hotwaterheating-1, data=Housing1)
hwhyes <- hu[,"hotwaterheatingyes"]
hwhno <- hu[,"hotwaterheatingno"]

ac <- model.matrix(~airconditioning-1, data=Housing1)
acyes <- ac[,"airconditioningyes"]
acno <- ac[,"airconditioningno"]
```

```
fs <- model.matrix(~furnishingstatus-1, data=Housing1)
fsfurnished <- fs[,"furnishingstatusfurnished"]
fssemifurnished <- fs[,"furnishingstatussemi-furnished"]
fsunfurnished <- fs[,"furnishingstatusunfurnished"]

#add dummy variables into original dataset and create another table
Housing1%>%
  mutate(hotwaterheatingyes=hwhyes)%>%
  mutate(hotwaterheatingno=hwhno)%>%
  mutate(airconditioningyes=acyes)%>%
  mutate(airconditioninno=acno)%>%
  mutate(furnished=fsfurnished)%>%
  mutate(semifurnished=fssemifurnished)%>%
  mutate(unfurnished=fsunfurnished)%>%
  select(-hotwaterheating,-airconditioning,-furnishingstatus)-> Housing3
Housing3

## # A tibble: 545 × 13
##      price  area bedro…¹ bathr…² stories parking hotwa…³ hotwa…⁴ airco…
⁵ airco…⁶
##      <dbl> <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl
>   <dbl>
##  1  1.33e7  7420       4       2       3       2       0       1
1       0
##  2  1.23e7  8960       4       4       4       3       0       1
1       0
##  3  1.23e7  9960       3       2       2       2       0       1
0       1
##  4  1.22e7  7500       4       2       2       3       0       1
1       0
##  5  1.14e7  7420       4       1       2       2       0       1
1       0
##  6  1.08e7  7500       3       3       1       2       0       1
1       0
##  7  1.01e7  8580       4       3       4       2       0       1
1       0
##  8  1.01e7 16200       5       3       2       0       0       1
0       1
##  9  9.87e6  8100       4       1       2       2       0       1
1       0
## 10  9.8 e6  5750       3       2       4       1       0       1
1       0
## # … with 535 more rows, 3 more variables: furnished <dbl>, semifurnishe
d <dbl>,
## #   unfurnished <dbl>, and abbreviated variable names ¹bedrooms, ²bathr
ooms,
## #   ³hotwaterheatingyes, ⁴hotwaterheatingno, ⁵airconditioningyes,
## #   ⁶airconditioninno
```

-> Full regression model:

price = 463377.7+292.2*area+144630.4*bedrooms+1032224.5*bathrooms+408902.6*stories+773516.1*hotwaterheatingyes+948967.7*airconditioningyes+289463*parking-115599.7*semi-furnished-563049.7*unfurnished

```
#full regression model that consists of quantitative and categorical varia
bles
Housingmodel<-lm(price~.,data = Housing1)
Housingmodel

##
## Call:
## lm(formula = price ~ ., data = Housing1)
##
## Coefficients:
##                   (Intercept)                          area
##                      463377.7                         292.2
##                      bedrooms                     bathrooms
##                      144630.4                     1032224.5
##                       stories             hotwaterheatingyes
##                      408902.6                      773516.1
##            airconditioningyes                       parking
##                      948967.7                      289463.0
## furnishingstatussemi-furnished     furnishingstatusunfurnished
##                     -115599.7                     -563049.7
```

#4. Using only the quantitative variables as predictors, produce a model using matrix methods. Also use matrix methods to find the fitted values and the residuals

```
#matrix y
Ym<-matrix(Housing$price,ncol = 1,byrow = TRUE)
head(Ym)

##           [,1]
##   [1,] 13300000
##   [2,] 12250000
##   [3,] 12250000
##   [4,] 12215000
##   [5,] 11410000
##   [6,] 10850000

#matrix x
#choosing only quantitative variables in original dataset and set them int
o as.matrix, also assign all independant values into 1 to become x-matrix
Xm<-as.matrix(Housing2)
Xm[Xm>20000]<-1
head(Xm)
```

```
##        price  area bedrooms bathrooms stories parking
##   [1,]     1  7420        4         2       3       2
##   [2,]     1  8960        4         4       4       3
##   [3,]     1  9960        3         2       2       2
##   [4,]     1  7500        4         2       2       3
##   [5,]     1  7420        4         1       2       2
##   [6,]     1  7500        3         3       1       2
```

```
t(Xm) -> transposeX
transposeX%*%Xm -> Product
solve(Product)%*%transposeX%*%Ym->interpretandslopes
interpretandslopes
```

```
##                   [,1]
## price     -145734.4895
## area          331.1155
## bedrooms   167809.7881
## bathrooms 1133740.1627
## stories    547939.8095
## parking    377596.2887
```

```
#fitted values
Xm%*%interpretandslopes ->fittedvalue
head(fittedvalue)
```

```
##            [,1]
##   [1,]  7648874
##   [2,] 11351808
##   [3,]  7774158
##   [4,]  7505020
##   [5,]  5967194
##   [6,]  7545414
```

```
#residuals
Ym-fittedvalue ->residuals
head(residuals)
```

```
##                 [,1]
##   [1,]  5651126.0307
##   [2,]   898191.7444
##   [3,]  4475842.2702
##   [4,]  4709980.3119
##   [5,]  5442806.0029
##   [6,]  3304586.0355
```

#5. Produce an output summary table to be used to analyze and evaluate the full model (Adjusted R squared, Standard Error, Significance of Variables, ect…)

-> There are 62.98% of the variation in the dependent variable is explained by the independent variables in the model.

-> Independent variable area standard error is the lowest in the full regression model, which indicates that the predicted values are closer to the actual values.

-> Only bedrooms and semifurnished status are not significant, other independent variables are significant.

```
summary(Housingmodel)

##
## Call:
## lm(formula = price ~ ., data = Housing1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2693624  -728713   -83105   568026  5271131
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)

## (Intercept)                       463377.72  257068.00   1.803  0.07202 .

## area                                 292.17      24.95  11.712  < 2e-16 *
**
## bedrooms                          144630.41   76640.04   1.887  0.05968 .

## bathrooms                        1032224.53  110154.30   9.371  < 2e-16 *
**
## stories                           408902.62   65521.36   6.241 8.87e-10 *
**
## hotwaterheatingyes                773516.09  239402.67   3.231  0.00131 *
*
## airconditioningyes                948967.68  115731.31   8.200 1.80e-15 *
**
## parking                           289462.98   62376.15   4.641 4.37e-06 *
**
## furnishingstatussemi-furnished -115599.66  124939.33  -0.925  0.35525

## furnishingstatusunfurnished    -563049.74  134166.76  -4.197 3.17e-05 *
**
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1148000 on 535 degrees of freedom
## Multiple R-squared:  0.6298, Adjusted R-squared:  0.6236
## F-statistic: 101.1 on 9 and 535 DF,  p-value: < 2.2e-16
```

#6. Use procedures and techniques explored in class to produce confidence intervals for the independent quantitative variables of your model. Choose at least two of the quantitative variables to find confidence intervals for.

```r
#confidence intervals for the independent quantitative variables
Housingmodel2<-lm(price~.,data=Housing2)
tidy(Housingmodel2, conf.int = TRUE)
```

```
## # A tibble: 6 × 7
##   term        estimate std.error statistic  p.value conf.low conf.high
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
## 1 (Intercept) -145734.   246634.    -0.591 5.55e- 1 -630217.   338748.
## 2 area            331.       26.6   12.4   1.92e-31      279.       383.
## 3 bedrooms     167810.    82933.     2.02  4.35e- 2     4899.   330721.
## 4 bathrooms   1133740.   118828.     9.54  4.86e-20   900317.  1367164.
## 5 stories      547940.    68894.     7.95  1.07e-14   412605.   683274.
## 6 parking      377596.    66804.     5.65  2.57e- 8   246368.   508825.
```

#7. Now produce a reduced model (removing variables of your choice with justification). Use R summary coding for both models and offer justification for choosing one model over the other.

-> According to p-value, we removed bedrooms and one dummy variable of each categorical variables then create new reduced model

-> New regression model:

price = 625984.83+294.78*area+1083853.31*bathrooms+448108.09*stories+296188.69*parking+239929.05*hotwaterheatingyes+949085.40*airconditioningyes+112015.45*furnished-460212.16*unfurnished

```r
#reduced model
Housing4<-Housing3%>%
  select(-bedrooms,-semifurnished,-hotwaterheatingno,-airconditioninno)
reducedmodel<-lm(price~.,data = Housing4)
summary(reducedmodel)
```

```
##
## Call:
## lm(formula = price ~ ., data = Housing4)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2723604  -729880   -82182   556837  5321232
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           625984.83  192389.58   3.254  0.00121 **
## area                     294.78      24.97  11.807  < 2e-16 ***
```

```
## bathrooms          1083853.31  106957.48  10.133  < 2e-16 ***
## stories             448108.09   62288.82   7.194 2.13e-12 ***
## parking             296188.69   62422.86   4.745 2.68e-06 ***
## hotwaterheatingyes  782258.14  239929.05   3.260  0.00118 **
## airconditioningyes  949085.40  116007.48   8.181 2.05e-15 ***
## furnished           112015.45  125223.01   0.895  0.37144
## unfurnished        -460212.16  116740.67  -3.942 9.15e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1150000 on 536 degrees of freedom
## Multiple R-squared:  0.6274, Adjusted R-squared:  0.6218
## F-statistic: 112.8 on 8 and 536 DF,  p-value: < 2.2e-16
```

#8. Research and apply a model analysis technique not discussed in class to your full model or reduced model. Fully explain the technique or procedure and how it is being applied to your specific model.

-> The new regression model we found that can analyze our data is random forest. Random forest can prevent overfitting problem with multiple decision trees, each tree draws a sample random data giving the random forest more randomness to produce much better accuracy.

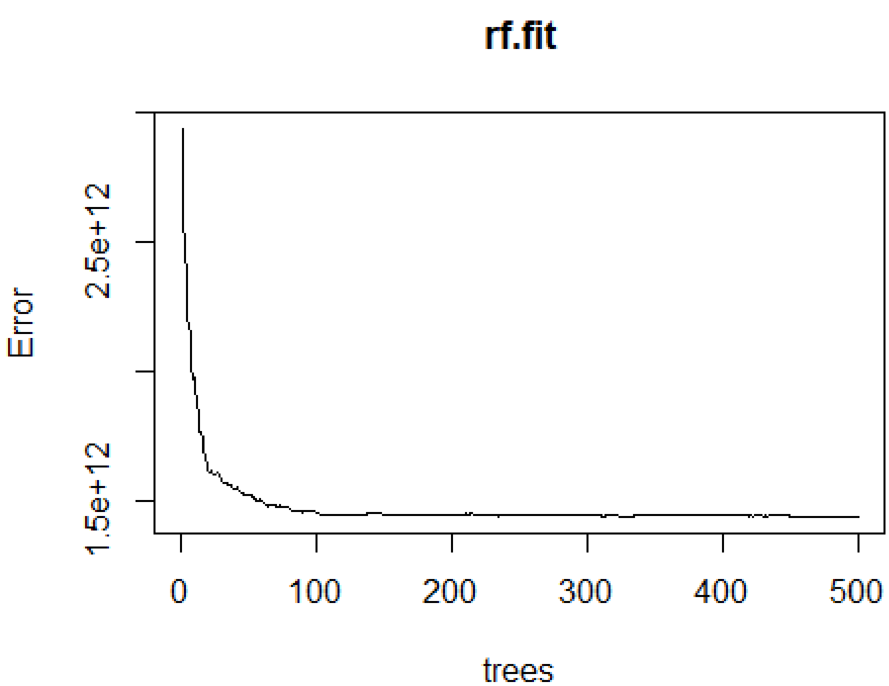
-> From regression model, there is 58.66% of the variation in the dependent variable is explained by the independent variables included in the model.

-> The output assumes 500 trees in random forest regression, where at each split, only two variables are considered. The tree with the 474th split was found to have the minimum mean squared error.

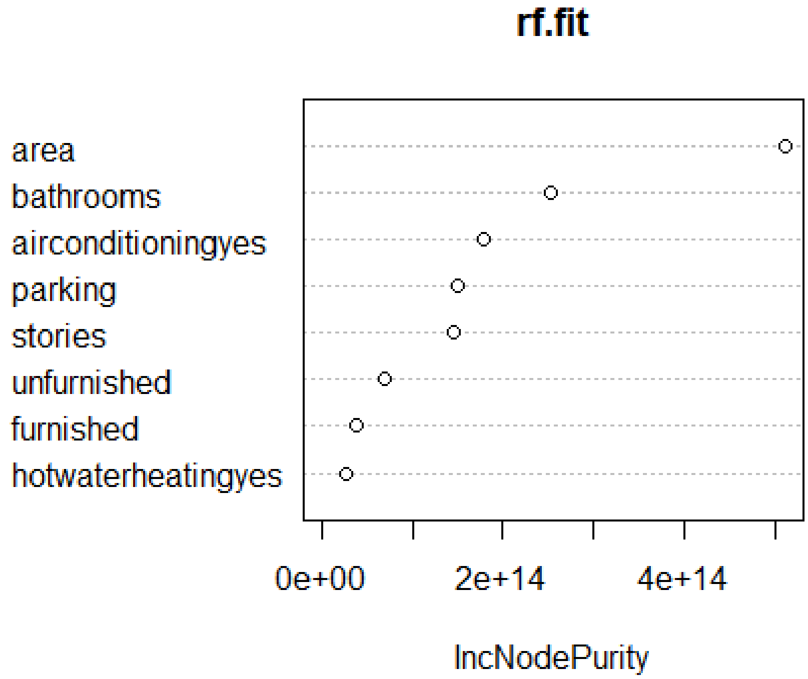
-> The square root of the mean squared error for the 474th split is 1200670, indicating the average deviation of the predicted values from the actual values.

```
library(randomForest)

## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

##'package:dplyr':
##
##     combine

## 'package:ggplot2':
##
##     margin

set.seed(4543)
rf.fit <- randomForest(price ~ ., data=Housing4)
rf.fit
```

```
##
## Call:
##  randomForest(formula = price ~ ., data = Housing4)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##           Mean of squared residuals: 1.443564e+12
##                     % Var explained: 58.66

which.min(rf.fit$mse)

## [1] 474

sqrt(rf.fit$mse[which.min(rf.fit$mse)])

## [1] 1200670

plot(rf.fit)
```



```
#shows the importance of each variable in a regression model in decreasing
 order of importance
varImpPlot(rf.fit)
```



#9. Offer final summary perspectives about the data and the models that you produce, suggesting how your models or model analysis enhanced your understanding of the data.

-> Initially, a full regression model was built, followed by the removal of insignificant variables based on their statistical significance. The reduced model resulted in a multiple R-squared value of 0.6274, indicating that 62.74% of the variation in housing prices can be explained by this model. Additionally, the analysis revealed that the most influential three predictors of housing prices were area, number of bathrooms, and availability of air conditioning. Interestingly, we expected the numbers of bedrooms would be one of the important factors but it was not found to be significant based on the analysis.