House Prices

CUNY SPS DATA 621

GROUP 2: William Aiken, Donald Butler, Michael Ippolito, Bharani Nittala, and Leticia Salazar

Abstract:

Use 250 words or less to summarize your problem, methodology, and major outcomes.

[insert text here]

Key words:

house prices, linear models, regression, home investment, real estate

Introduction:

Describe the background and motivation of your problem.

[insert text here]

Literature Review:

Discuss how other researchers have addressed similar problems, what their achievements are, and what the advantage and drawbacks of each reviewed approach are. Explain how your investigation is similar or different to the state-of-the- art. Please cite the relevant papers where appropriate.

[insert text here]

Methology:

Discuss the key aspects of your problem, data set and regression model(s). Given that you are working on real-world data, explain at a high-level your exploratory data analysis, how you prepared the data for regression modeling, your process for building regression models, and your model selection.

[insert text here]

Discussion of Findings:

Describe the specifics of what you did (data exploration, data preparation, model building, model selection, model evaluation, etc.), and what you found out (statistical analyses, interpretation and discussion of the results, etc.). Conclude your findings, limitations, and suggest areas for future work.

[insert text here]

Recommendations: [insert text here]

Limitations: [insert text here]

Conclusion:

[insert text here]

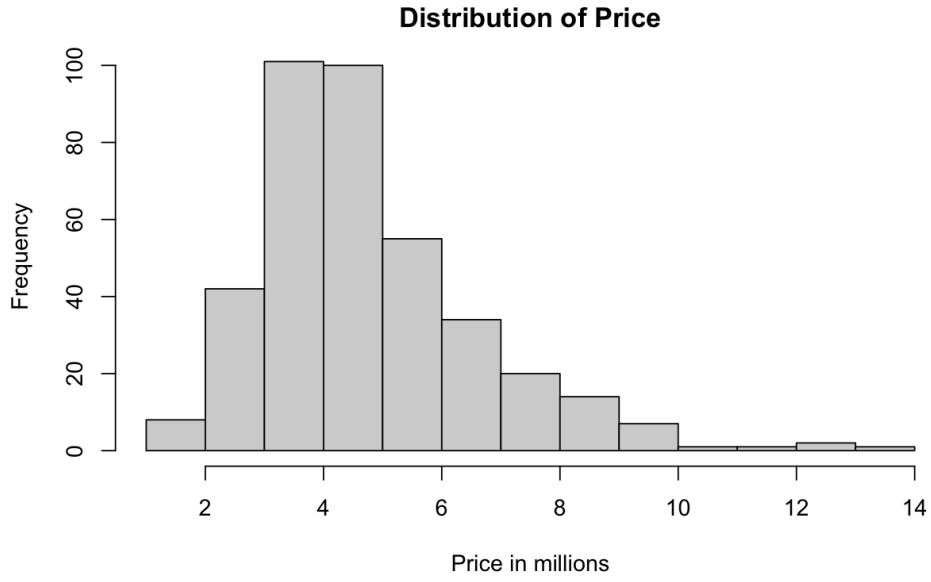
References / Bibliography:

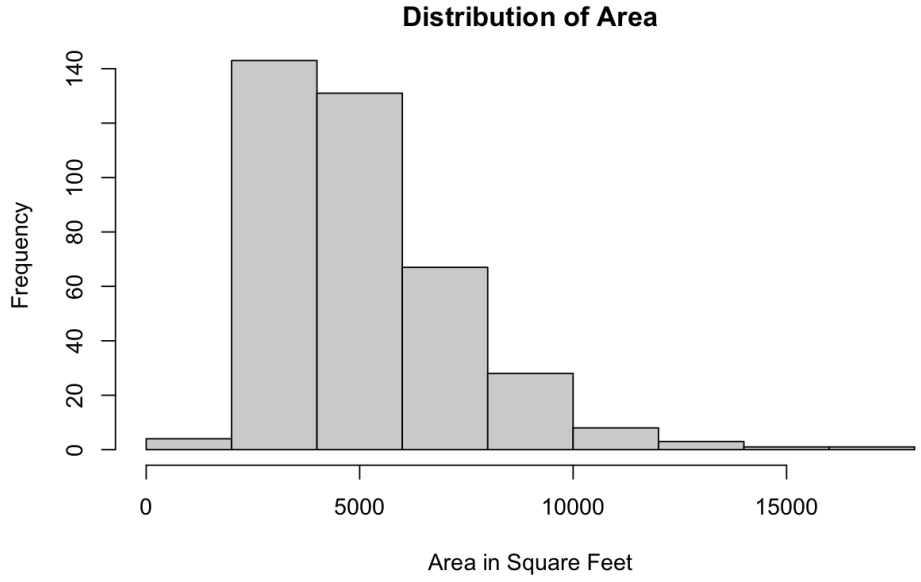
Be sure to cite all references used in the report (APA format).

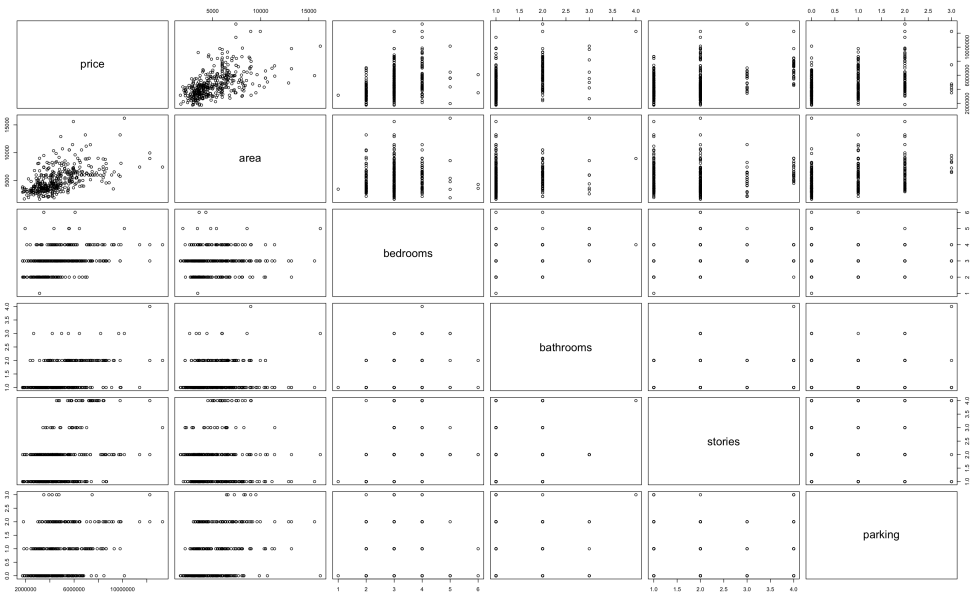
- H, M. Y. (2022, January 12). Housing prices dataset. Kaggle. Retrieved November 28, 2022, from https://www.kaggle.com/datasets/yasserh/housing-prices-dataset
- Schmidt, A. (2018, June). Linear Regression and the normality assumption. Retrieved December 5, 2022, from https://doi.org/10.1016/j.jclinepi.2017.12.006
- [insert text here]

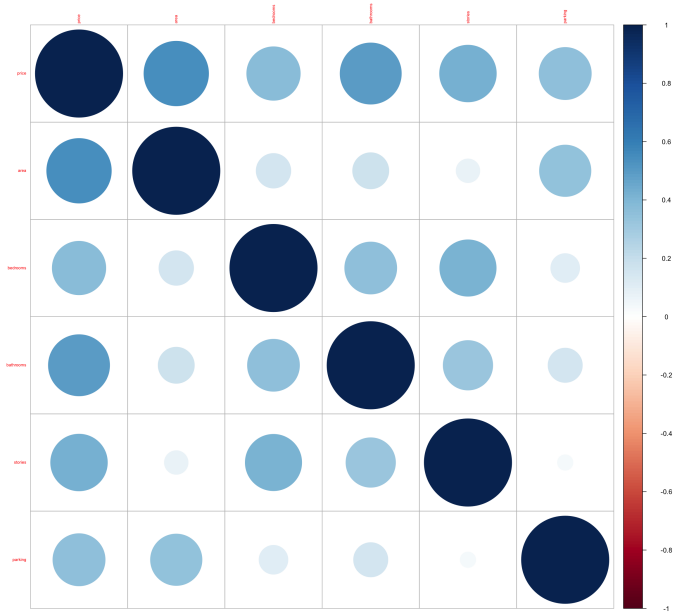
Appendices:

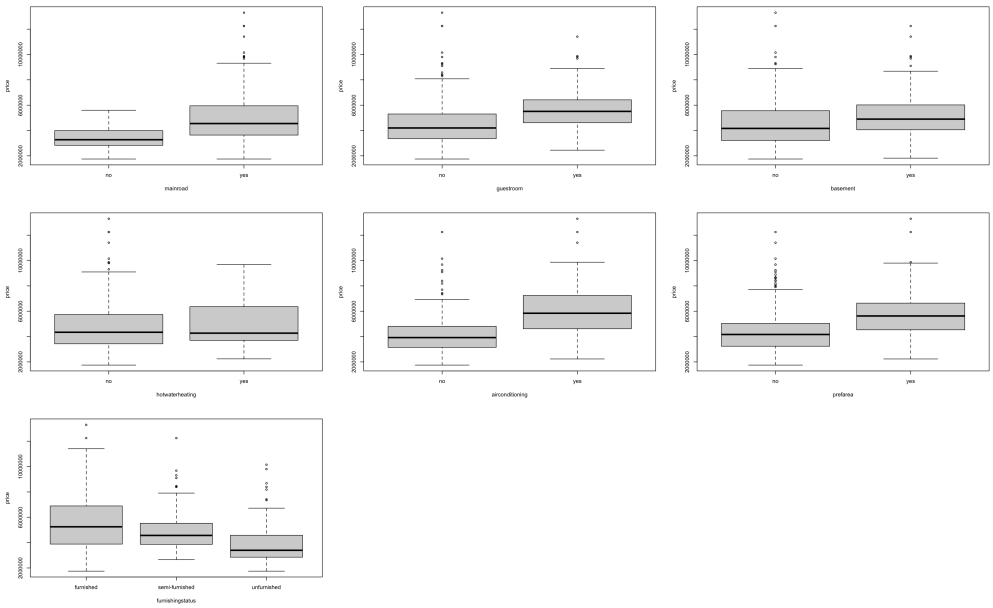
Appendix A. Figures:






price
area
bedrooms
bathrooms
stories
parking


price
area
bedrooms
bathrooms
stories
parking
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1


price
mainroad
guestroom
basement
hotwaterheating
airconditioning
prefarea
furnishingstatus
furnished
semi-furnished
unfurnished

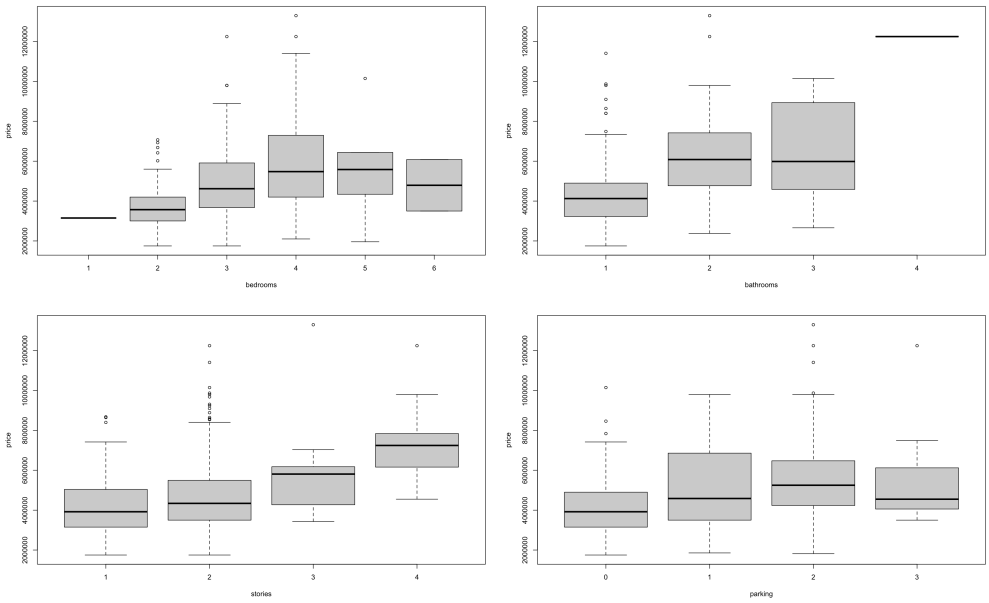
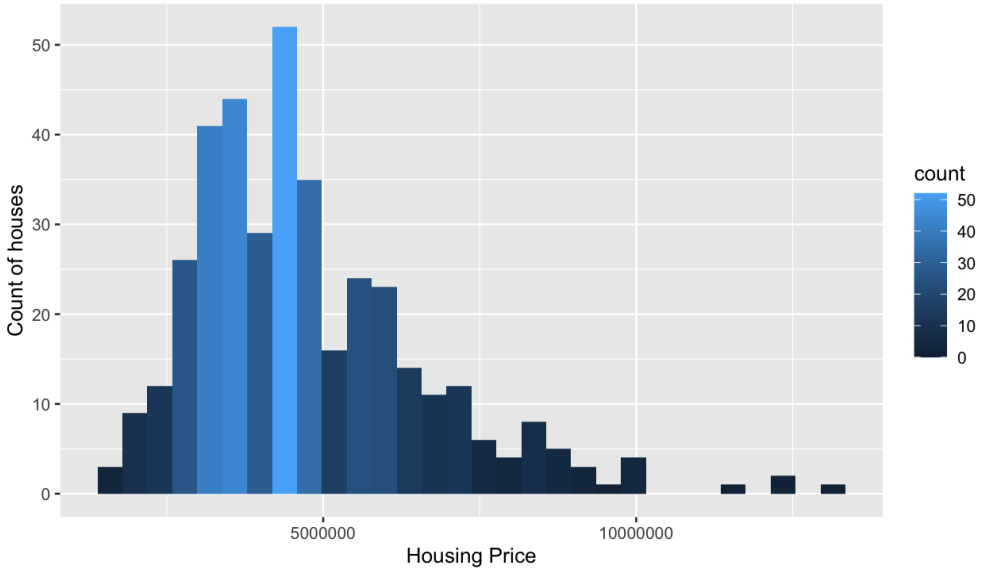
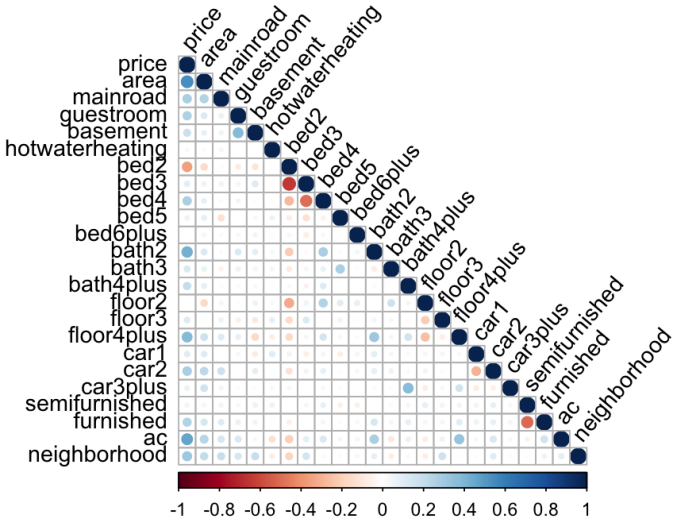
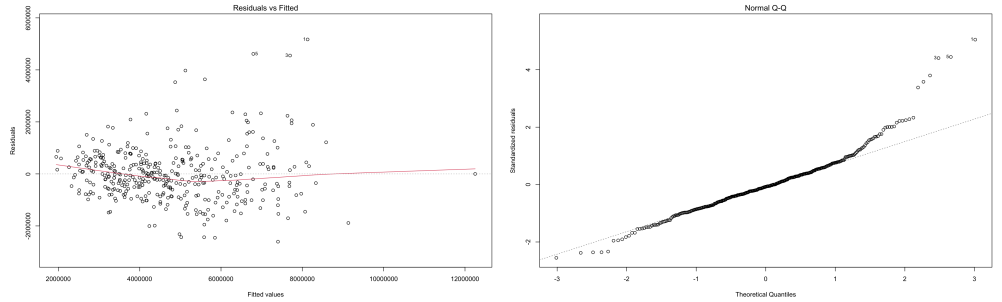
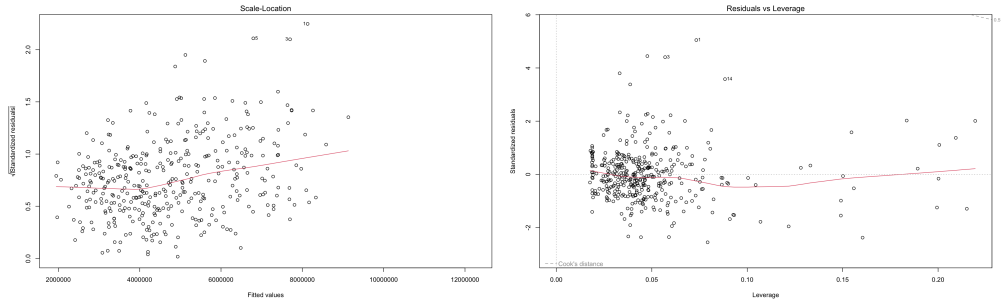

price
bedrooms
bathrooms
stories
parking

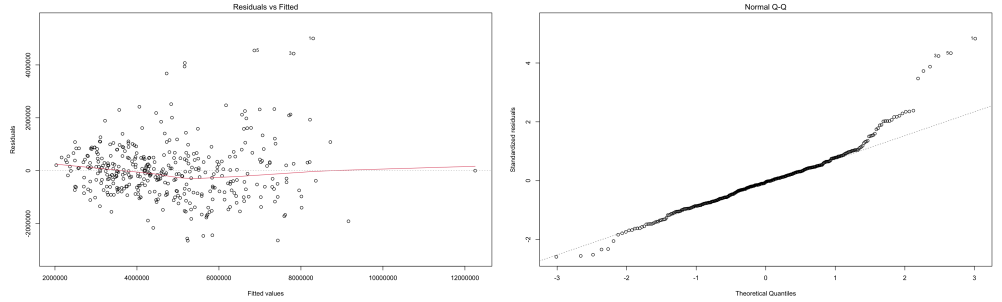
Figure 1 Histogram of Price

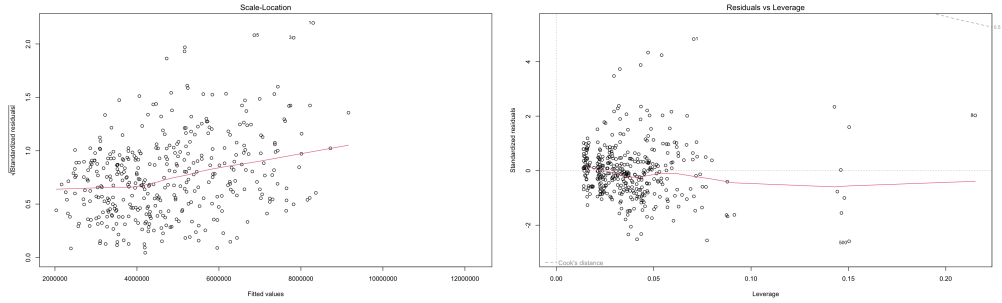


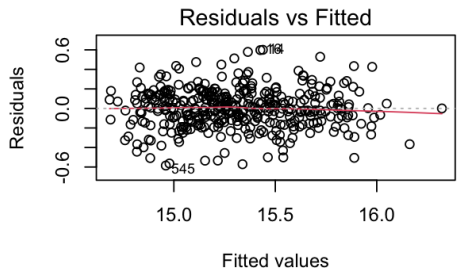


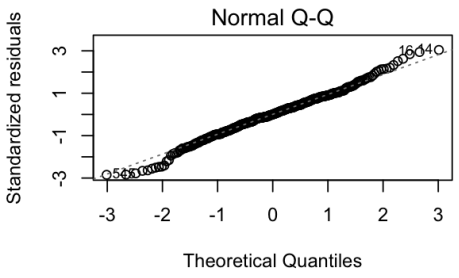

Residuals vs Fitted
Residuals
Fitted values
Normal Q-Q
Standardized residuals
Theoretical Quantiles

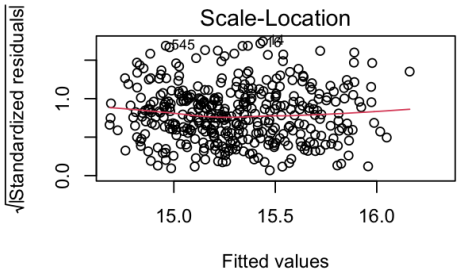

Scale-Location
√|Standardized residuals|
Fitted values
Residuals vs Leverage
Standardized residuals
Leverage
Cook's distance

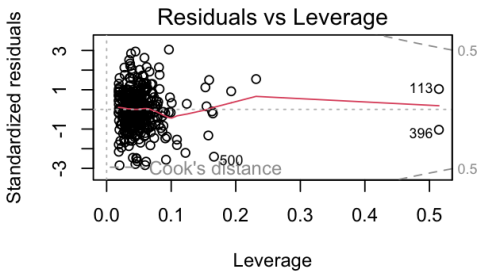

Residuals vs Fitted
Residuals
Fitted values
Normal Q-Q
Standardized residuals
Theoretical Quantiles

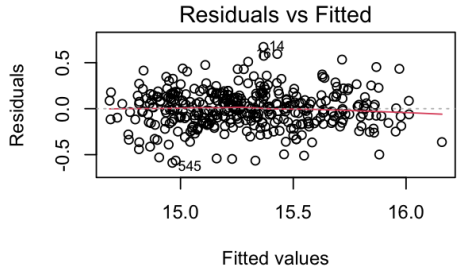

Scale-Location
√|Standardized residuals|
Fitted values
Residuals vs Leverage
Standardized residuals
Leverage
Cook's distance

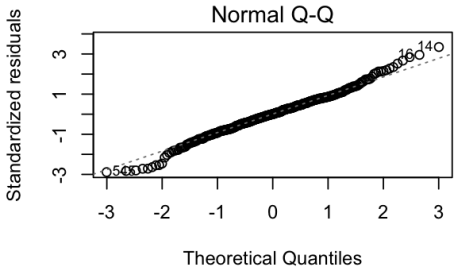

Residuals vs Fitted
Residuals
Fitted values
15.0
15.5
16.0
-0.6
0.0
0.6
545

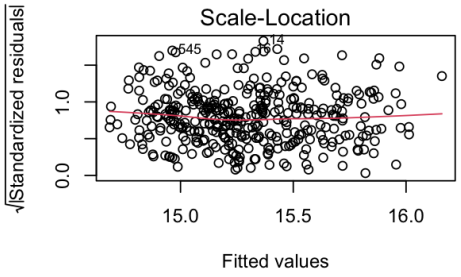

Normal Q-Q
Standardized residuals
Theoretical Quantiles
-3
-2
-1
0
1
2
3
16
14
545

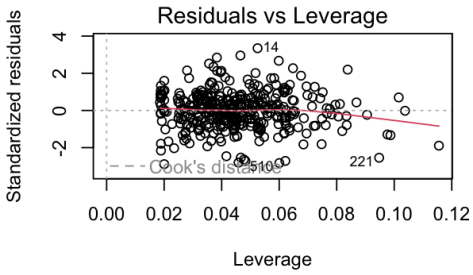

Scale-Location
√|Standardized residuals|
Fitted values
15.0
15.5
16.0
0.0
1.0
545

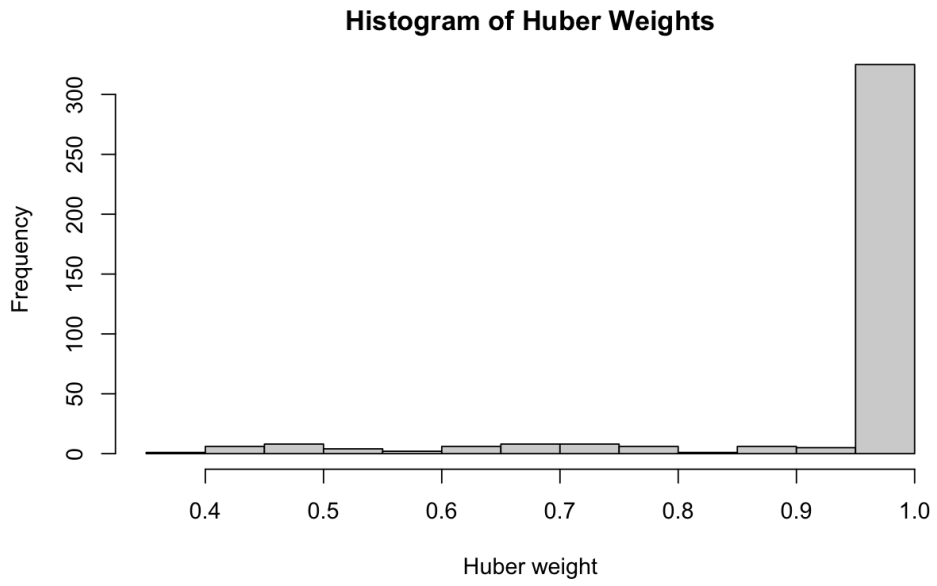

Residuals vs Leverage
Standardized residuals
Leverage
Cook's distance
0.0
0.1
0.2
0.3
0.4
0.5
0.5
0.5
113
396
500

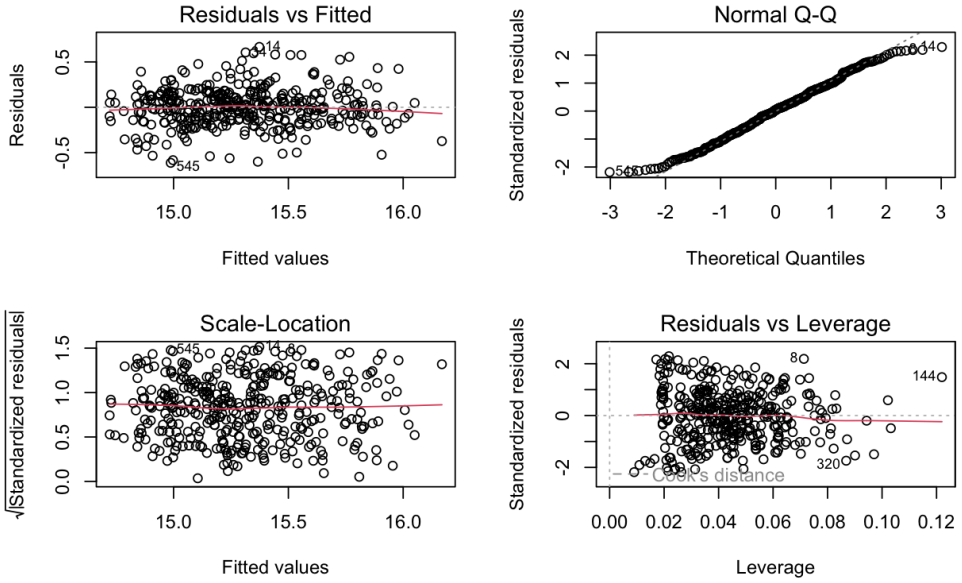

Residuals vs Fitted
Residuals
Fitted values
15.0
15.5
16.0
-0.5
0.0
0.5
14
545

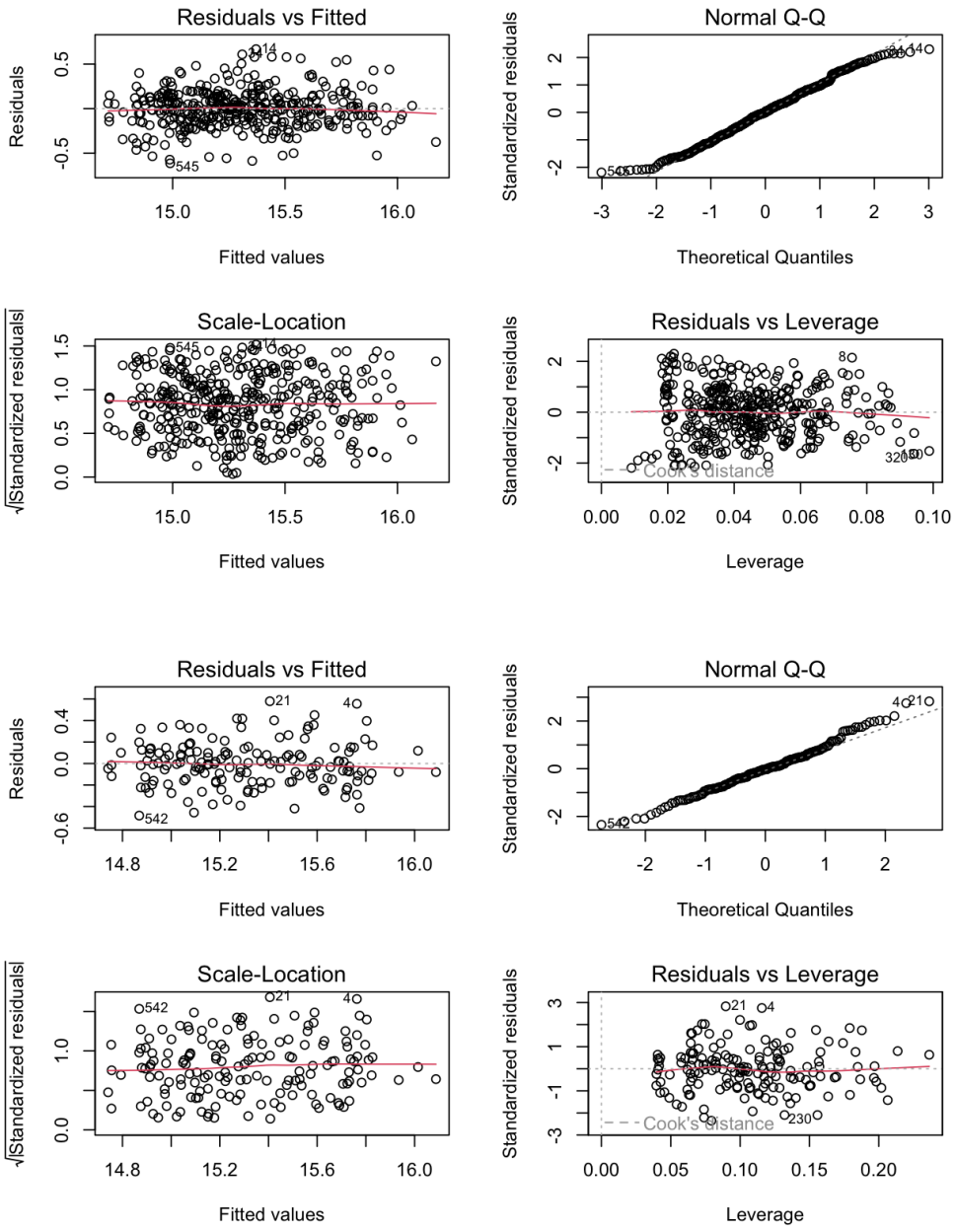

Normal Q-Q
Standardized residuals
Theoretical Quantiles
-3
-2
-1
0
1
2
3
16
14
545


Scale-Location
√|Standardized residuals|
Fitted values
15.0
15.5
16.0
0.0
1.0
14
545


Residuals vs Leverage
Standardized residuals
Leverage
Cook's distance
0.00
0.02
0.04
0.06
0.08
0.10
0.12
14
221
510


Histogram of Huber Weights
Frequency
0
50
100
150
200
250
300
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Huber weight


Residuals vs Fitted
Residuals
-0.5
0.5
15.0
15.5
16.0
Fitted values
Normal Q-Q
Standardized residuals
-2
0
1
2
-3
-2
-1
0
1
2
3
Theoretical Quantiles
Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
15.0
15.5
16.0
Fitted values
Residuals vs Leverage
Standardized residuals
-2
0
2
0.00
0.02
0.04
0.06
0.08
0.10
0.12
Leverage
Cook's distance



Appendix B. Tables:

#	Train/Validation	Linear/Robust	Full/Step-reduced	Log Transform	Outliers Removed	Huber-Weighted	Adj R-Sqr
<int>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<dbl>
1	Train	Linear	Full				0.675
2	Train	Linear	Step				0.677
3	Train	Linear	Step				0.670
4	Train	Linear	Full	Yes			0.691
5	Train	Linear	Step	Yes			0.691
6	Train	Linear	Step	Yes	Yes		0.684
7	Train	Linear	Step	Yes	Yes		0.684
8	Train	Robust	Step	Yes			*NA*
9	Train	Linear	Step	Yes		Yes	0.710
10	Train	Linear	Step	Yes	Yes	Yes	0.710

1–10 of 11 rows

Previous 1 2 Next

Appendix C. R Code:

```
# load libraries
library(tidyverse)
library(dplyr)
library(corrplot)
library(MASS)
library(dvmisc)
library(car)
library(lmtest)
library(olsrr)
library(caret)

# load data
house_prices <- read.csv("https://raw.githubusercontent.com/letisalba/Data_621/master/Final_Project/csv/

# make this example reproducible
set.seed(1)

# use 70% of data set as training set and 30% as evaluation set
sample <- sample(c(TRUE, FALSE), nrow(house_prices), replace=TRUE, prob=c(0.7,0.3))
dftrain  <- house_prices[sample, ]
dfeval   <- house_prices[!sample, ]

head(dftrain)

# explore data
str(dftrain)
str(dfeval)

summary(dftrain)
summary(dfeval)

# distribution of the price ranges in our data
options(scipen=5)
hist(dftrain$price / 10^6, main = 'Distribution of Price', xlab = 'Price in millions')

# distribution of area
hist(dftrain$area, main = 'Distribution of Area', xlab = 'Area in Square Feet')

# looking at all the variables separately
dftrain %>% count(bedrooms)
dftrain %>% count(bathrooms)
dftrain %>% count(stories)
dftrain %>% count(parking)
dftrain %>% count(furnishingstatus)
dftrain %>% count(mainroad)
dftrain %>% count(guestroom)
dftrain %>% count(basement)
dftrain %>% count(hotwaterheating)
dftrain %>% count(airconditioning)
dftrain %>% count(prefarea)
```

```r
# clean function to create dummy variables to replace our categorical variables
fun_clean_df <- function(df) {
  df <- df %>%
    mutate(bed2 = ifelse(bedrooms == 2,1,0)) %>%
    mutate(bed3 = ifelse(bedrooms == 3,1,0)) %>%
    mutate(bed4 = ifelse(bedrooms == 4,1,0)) %>%
    mutate(bed5 = ifelse(bedrooms == 5,1,0)) %>%
    mutate(bed6plus = ifelse(bedrooms >= 6,1,0)) %>% dplyr::select(-bedrooms) %>%
    mutate(bath2 = ifelse(bathrooms == 2,1,0)) %>%
    mutate(bath3 = ifelse(bathrooms == 3,1,0)) %>%
    mutate(bath4plus = ifelse(bathrooms >= 4,1,0)) %>% dplyr::select(-bathrooms) %>%
    mutate(floor2 = ifelse(stories == 2,1,0)) %>%
    mutate(floor3 = ifelse(stories == 3,1,0)) %>%
    mutate(floor4plus = ifelse(stories >= 4,1,0)) %>% dplyr::select(-stories) %>%
    mutate(car1 = ifelse(parking == 1,1,0)) %>%
    mutate(car2 = ifelse(parking == 2,1,0)) %>%
    mutate(car3plus = ifelse(parking >= 3,1,0)) %>% dplyr::select(-parking) %>%
    mutate(semifurnished = ifelse(furnishingstatus == 'semi-furnished',1,0)) %>%
    mutate(furnished = ifelse(furnishingstatus == 'furnished',1,0)) %>% dplyr::select(-furnishingstatus
    mutate(mainroad = ifelse(mainroad == 'yes',1,0)) %>%
    mutate(guestroom = ifelse(guestroom == 'yes',1,0)) %>%
    mutate(basement = ifelse(basement == 'yes',1,0)) %>%
    mutate(hotwaterheating = ifelse(hotwaterheating == 'yes',1,0)) %>%
    mutate(ac = ifelse(airconditioning == 'yes',1,0)) %>% dplyr::select(-airconditioning) %>%
    mutate(neighborhood = ifelse(prefarea == 'yes',1,0)) %>% dplyr::select(-prefarea)
}

dftrain_clean <- fun_clean_df(dftrain)
dfeval_clean <- fun_clean_df(dfeval)

# Compare quantitative variables
pairs(dftrain[, c('price', 'area', 'bedrooms', 'bathrooms', 'stories', 'parking')])
corrplot(cor(dftrain[, c('price', 'area', 'bedrooms', 'bathrooms', 'stories', 'parking')], use = 'compl

# Compare nominal categorical variables to price
par(mfrow=c(3, 3))
boxplot(price ~ mainroad, data=dftrain)
boxplot(price ~ guestroom, data=dftrain)
boxplot(price ~ basement, data=dftrain)
boxplot(price ~ hotwaterheating, data=dftrain)
boxplot(price ~ airconditioning, data=dftrain)
boxplot(price ~ prefarea, data=dftrain)
boxplot(price ~ furnishingstatus, data=dftrain)

# Compare ordinal categorical variables
par(mfrow=c(2, 2))
boxplot(price ~ bedrooms, data=dftrain)
boxplot(price ~ bathrooms, data=dftrain)
boxplot(price ~ stories, data=dftrain)
boxplot(price ~ parking, data=dftrain)

# Draw a histogram to figure out the distribution of Sale Price
options(scipen=10000)
```

```r
ggplot(dftrain_clean, aes(x = price, fill = ..count..)) +
  geom_histogram() +
  ggtitle("Figure 1 Histogram of Price") +
  ylab("Count of houses") +
  xlab("Housing Price") +
  theme(plot.title = element_text(hjust = 0.9))

# corrplot
cor_res <- cor(dftrain_clean)
corrplot(cor_res,
         type = "lower",
         order = "original",
         tl.col = "black",
         tl.srt = 50,
         tl.cex = 1)

# start of model building

#partition data
#set.seed(10000)
#train.index <- sample(c(1:dim(dftrain_clean)[1]), dim(dftrain_clean)[1]*0.8)
#model_lin_train = dftrain_clean[train.index,]
#model_lin_valid <- dftrain_clean[-train.index,]
model_lin_train <- dftrain_clean

# mlr 1
lm_mod1 <- lm(price ~., data = model_lin_train)
aic_lm_mod1 = AIC(lm_mod1)
summary(lm_mod1)

# mlr 2
lm_mod2 <- stepAIC(lm_mod1, trace = F)
aic_lm_mod2 = AIC(lm_mod2)
summary(lm_mod2)

# mlr 3
# reduce collinearity, and remove low values
lm_mod3 <- lm(price ~ area + guestroom + basement + bath2 + bath3 +
    bath4plus + floor2 + floor3 + floor4plus + car1 + car2 +
    car3plus + semifurnished + furnished + ac + neighborhood
    - car3plus - bed2,
    data = model_lin_train)
summary(lm_mod3)

# model selection
# Define function to calculate mean squared error
calc_mse <- function(lmod) {
  return(mean((summary(lmod))$residuals ^ 2))
}

# Define function to aid in model analysis
ModelAnalysis <- function(lmod) {
```

```r
# Plot residuals
print('-----------------------------------------------------')
print(lmod$call)
par(mfrow=c(2,2))
plot(lmod)
print('')

# Shapiro test to determine normality of residuals
# Null hypothesis: the residuals are normal.
# If the p-value is small, reject the null, i.e., consider the residuals *not* normally distributed.
if (length(lmod$fitted.values) > 3 & length(lmod$fitted.values) < 5000) {
    st <- shapiro.test(lmod$residuals)
    if (st$p.value <= 0.05) {
      print(paste0("Shapiro test for normality: The p-value of ", st$p.value, " is <= 0.05, so reject
    } else {
      print(paste0("Shapiro test for normality: The p-value of ", st$p.value, " is > 0.05, so do not r
    }
    print('')
} else {
    print("Shapiro test for normality of residuals cannot be performed; sample length must be between
}

# Breusch-Pagan test to determine homoschedasticity of residuals
# Null hypothesis: the residuals are homoschedastic.
# If the p-value is small, reject the null, i.e., consider the residuals heteroschedastic.
bp <- bptest(lmod)
if (bp$p.value > 0.05 & bp$statistic < 10) {
    print(paste0("Breusch-Pagan test for homoschedasticity: The p-value of ", bp$p.value, " is > 0.05
        " is < 10, so don't reject the null; i.e., the residuals are HOMOSCHEDASTIC."))
} else if (bp$p.value <= 0.05) {
    print(paste0("Breusch-Pagan test for homoschedasticity: The p-value of ", bp$p.value, " is <= 0.05
        ", so reject the null; i.e., the residuals are HETEROSCHEDASTIC."))
} else {
    print(paste0("Breusch-Pagan test for homoschedasticity: The p-value of ", bp$p.value, " and test s
        " are inconclusive, so homoschedasticity can't be determined using this test. But since the p-
        "it is reasonable to conclude that the residuals are HOMOSCHEDASTIC."))
}
print('')

# Visually look for colinearity - dont do this for large models
#pairs(model.matrix(lmod))

# Variance inflation factor (VIF)
print('Variance inflation factor (VIF)')
print('<=1: not correlated, 1-5: moderately correlated, >5: strongly correlated')
print(sort(vif(lmod), decreasing=T))
print('')

# Standardized residual plots (look for points outside of 2 or 3 stdev)
p <- length(summary(lmod)$coeff[,1] - 1)  # number of model parameters
stanres <- rstandard(lmod)
for (i in seq(1, ceiling(p / 4))) {
  par(mfrow=c(2,2))
```

```
    starti <- ((i - 1) * 4) + 1
    for (j in seq(starti, starti + 3)) {
      if (j + 1 <= ncol(model.matrix(lmod))) {
        # Skip these plots since we're pretty sure that a linear model isn't valid here
        #plot(model.matrix(lmod)[, j + 1], stanres, xlab=colnames(model.matrix(lmod))[j + 1], ylab='Sta
        #abline(h=c(-2, 2), lt=3, col='blue')
        #abline(h=c(-3, 3), lt=2, col='red')
      }
    }
  }

  # Model scores
  print('Model scores:')
  print(paste0('    adjusted R-squared: ', round(summary(lmod)$adj.r.squared, 3)))
  print(paste0('    AIC: ', round(AIC(lmod, k=2), 3)))
  print(paste0('    BIC: ', round(BIC(lmod), 3)))
  print(paste0('    Mallow\'s Cp: ', round(ols_mallows_cp(lmod, fullmodel=lmod), 3)))
  print(paste0('    mean squared error: ', round(calc_mse(lmod), 3)))
  print('')

  # Find leverage point cutoff
  n <- length(lmod$residuals)
  cutoff <- 2 * (p + 1) / n
  print(paste0('Leverage point cutoff: ', cutoff))
  print('')

  # Show points of influence
  print('First 10 points of influence:')
  poi <- lm.influence(lmod)$hat
  len_poi <- length(poi)
  ct <- 0
  for (i in seq(1, length(poi))) {
    if (poi[i] > cutoff) {
      ct <- ct + 1
      print(paste0('    case #', i, ': ', round(poi[i], 3)))
    }
        if (ct > 10) {
            break
        }
  }
  print('')

}

# Analysis on the two step-reduced models
ModelAnalysis(lm_mod2)
ModelAnalysis(lm_mod3)

# Box-Cox transform on price
bc1 <- powerTransform(price ~ ., data=model_lin_train)
bc1

# Box-Cox result suggests doing a log transform on price
```

```
lm_mod4 <- lm(log(price) ~ ., data=model_lin_train)
summary(lm_mod4)
lm_mod5 <- stepAIC(lm_mod4, trace=F)
summary(lm_mod5)
ModelAnalysis(lm_mod5)

# Investigate top outliers
model_lin_train[c(2, 5, 9, 25, 49, 62, 77, 103, 110, 136, 210),]

# Log transform on price with outliers removed
lm_mod6 <- lm(formula(lm_mod5),
    data = model_lin_train[c(-2, -5, -9, -25, -49, -62, -77, -103, -110, -136, -210),])
summary(lm_mod6)
lm_mod7 <- stepAIC(lm_mod6, trace=F)
summary(lm_mod7)
ModelAnalysis(lm_mod7)

# Huber robust linear regression
lm_mod8 <- rlm(formula(lm_mod7), data=model_lin_train)
dftmp <- data.frame(cbind(price=model_lin_train$price, huber_weight=lm_mod8$w))
dftmp <- dftmp %>% arrange(huber_weight, ascending=F)
hist(dftmp$huber_weight, xlab='Huber weight', main='Histogram of Huber Weights')

# New linear model using Huber weights
lm_mod9 <- lm(formula(lm_mod7), weights=lm_mod8$w, data=model_lin_train)
summary(lm_mod9)
ModelAnalysis(lm_mod9)

# Remove most significant outliers
lm_mod10 <- lm(formula(lm_mod7), weights=lm_mod8$w[c(-53, -68, -87, -90, -103)],
    data=model_lin_train[c(-53, -68, -87, -90, -103),])
summary(lm_mod10)
ModelAnalysis(lm_mod10)

# Five-fold cross validation
set.seed(777)
tc <- trainControl(method = "cv", number = 5)
lmcv <- train(formula(lm_mod8), weights=lm_mod8$w, data=model_lin_train, method="lm", trControl=tc)
print(lmcv)
summary(lmcv)

# Huber robust linear regression
lm_valid1 <- lm(formula(lm_mod7), data=dfeval_clean)

# New linear model using Huber weights
lm_valid2 <- lm(formula(lm_mod7), weights=lm_valid1$w, data=dfeval_clean)
summary(lm_valid2)
ModelAnalysis(lm_valid2)

# Model comparison
hdr <- c('#', 'Train/Validation', 'Linear/Robust', 'Full/Step-reduced', 'Log Transform',
    'Outliers Removed', 'Huber-Weighted', 'Adj R-Sqr')
f1 <- c(seq(1:11))
```

```
f2 <- c(rep('Train', 10), 'Validation')
f3 <- c(rep('Linear', 7), 'Robust', rep('Linear', 3))
f4 <- c('Full', 'Step', 'Step', 'Full', rep('Step', 7))
f5 <- c(rep('', 3), rep('Yes', 8))
f6 <- c(rep('', 5), 'Yes', 'Yes', '', '', 'Yes', '')
f7 <- c(rep('', 8), rep('Yes', 3))
f8 <- round(c(
    summary(lm_mod1)$adj.r.squared,
    summary(lm_mod2)$adj.r.squared,
    summary(lm_mod3)$adj.r.squared,
    summary(lm_mod4)$adj.r.squared,
    summary(lm_mod5)$adj.r.squared,
    summary(lm_mod6)$adj.r.squared,
    summary(lm_mod7)$adj.r.squared,
    NA,
    summary(lm_mod9)$adj.r.squared,
    summary(lm_mod10)$adj.r.squared,
    summary(lm_valid2)$adj.r.squared), 3)
dfresult <- data.frame(f1, f2, f3, f4, f5, f6, f7, f8)
colnames(dfresult) <- hdr
dfresult
```