Analysis

2023-10-22

data=read.csv('data.csv')

str(data)

## 'data.frame': 6830 obs. of 22 variables:  
## $ X : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ Suburb : chr "Abbotsford" "Abbotsford" "Abbotsford" "Abbotsford" ...  
## $ Address : chr "25 Bloomburg St" "5 Charles St" "55a Park St" "124 Yarra St" ...  
## $ Rooms : int 2 3 4 3 2 2 3 2 2 3 ...  
## $ Type : chr "h" "h" "h" "h" ...  
## $ Price : num 1035000 1465000 1600000 1876000 1636000 ...  
## $ Method : chr "S" "SP" "VB" "S" ...  
## $ SellerG : chr "Biggin" "Biggin" "Nelson" "Nelson" ...  
## $ Date : chr "4/2/2016" "4/3/2017" "4/6/2016" "7/5/2016" ...  
## $ Distance : num 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 ...  
## $ Postcode : int 3067 3067 3067 3067 3067 3067 3067 3067 3067 3067 ...  
## $ Bedroom2 : int 2 3 3 4 2 3 3 2 2 3 ...  
## $ Bathroom : int 1 2 1 2 1 1 2 2 1 2 ...  
## $ Car : int 0 0 2 0 2 2 2 1 2 1 ...  
## $ Landsize : int 156 134 120 245 256 220 214 0 238 113 ...  
## $ BuildingArea : num 79 150 142 210 107 75 190 94 97 110 ...  
## $ YearBuilt : int 1900 1900 2014 1910 1890 1900 2005 2009 1890 1880 ...  
## $ CouncilArea : chr "Yarra" "Yarra" "Yarra" "Yarra" ...  
## $ Lattitude : num -37.8 -37.8 -37.8 -37.8 -37.8 ...  
## $ Longtitude : num 145 145 145 145 145 ...  
## $ Regionname : chr "Northern Metropolitan" "Northern Metropolitan" "Northern Metropolitan" "Northern Metropolitan" ...  
## $ Propertycount: int 4019 4019 4019 4019 4019 4019 4019 4019 4019 4019 ...

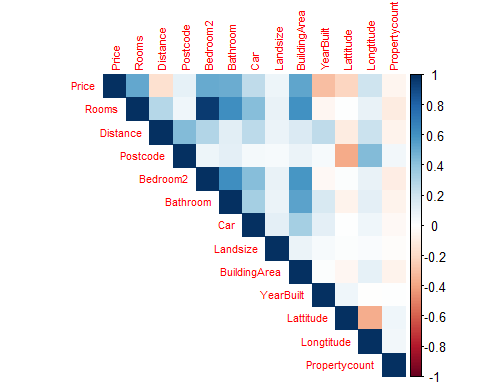
head(data)

## X Suburb Address Rooms Type Price Method SellerG Date  
## 1 1 Abbotsford 25 Bloomburg St 2 h 1035000 S Biggin 4/2/2016  
## 2 2 Abbotsford 5 Charles St 3 h 1465000 SP Biggin 4/3/2017  
## 3 3 Abbotsford 55a Park St 4 h 1600000 VB Nelson 4/6/2016  
## 4 4 Abbotsford 124 Yarra St 3 h 1876000 S Nelson 7/5/2016  
## 5 5 Abbotsford 98 Charles St 2 h 1636000 S Nelson 8/10/2016  
## 6 6 Abbotsford 10 Valiant St 2 h 1097000 S Biggin 8/10/2016  
## Distance Postcode Bedroom2 Bathroom Car Landsize BuildingArea YearBuilt  
## 1 2.5 3067 2 1 0 156 79 1900  
## 2 2.5 3067 3 2 0 134 150 1900  
## 3 2.5 3067 3 1 2 120 142 2014  
## 4 2.5 3067 4 2 0 245 210 1910  
## 5 2.5 3067 2 1 2 256 107 1890  
## 6 2.5 3067 3 1 2 220 75 1900  
## CouncilArea Lattitude Longtitude Regionname Propertycount  
## 1 Yarra -37.8079 144.9934 Northern Metropolitan 4019  
## 2 Yarra -37.8093 144.9944 Northern Metropolitan 4019  
## 3 Yarra -37.8072 144.9941 Northern Metropolitan 4019  
## 4 Yarra -37.8024 144.9993 Northern Metropolitan 4019  
## 5 Yarra -37.8060 144.9954 Northern Metropolitan 4019  
## 6 Yarra -37.8010 144.9989 Northern Metropolitan 4019

# Load the corrplot package  
library(corrplot)

## corrplot 0.92 loaded

# Select the variables of interest  
variables\_of\_interest <- data[, c('Price','Rooms' , 'Distance', 'Postcode', 'Bedroom2', 'Bathroom' ,'Car' ,'Landsize', 'BuildingArea', 'YearBuilt' , 'Lattitude', 'Longtitude' ,'Propertycount'  
 )]  
  
# Calculate the correlation matrix  
correlation\_matrix <- cor(variables\_of\_interest)  
  
# Create the heatmap  
corrplot(correlation\_matrix, method = "color", type = "upper", tl.cex = 0.7)



correlation matrix

# Select the variables of interest  
variables\_of\_interest <- data[,c('Price','Rooms' , 'Distance', 'Postcode', 'Bedroom2', 'Bathroom' ,'Car' ,'Landsize', 'BuildingArea', 'YearBuilt' , 'Lattitude', 'Longtitude' ,'Propertycount'  
 )]  
# Calculate the correlation matrix  
correlation\_matrix <- cor(variables\_of\_interest)  
  
# Print the correlation matrix as a table  
print(correlation\_matrix)

## Price Rooms Distance Postcode Bedroom2  
## Price 1.00000000 0.517717896 -0.16497466 0.10934259 0.50045197  
## Rooms 0.51771790 1.000000000 0.28976342 0.06867449 0.95585069  
## Distance -0.16497466 0.289763421 1.00000000 0.43827430 0.29610483  
## Postcode 0.10934259 0.068674489 0.43827430 1.00000000 0.07161117  
## Bedroom2 0.50045197 0.955850691 0.29610483 0.07161117 1.00000000  
## Bathroom 0.49248086 0.613284670 0.12404359 0.11277583 0.61683208  
## Car 0.25091590 0.420493152 0.26514249 0.04922636 0.42404101  
## Landsize 0.07353590 0.099030957 0.08244905 0.03999792 0.09806603  
## BuildingArea 0.52049158 0.603149536 0.15514796 0.08179100 0.58846256  
## YearBuilt -0.30734344 -0.049272047 0.25846163 0.03681922 -0.03812612  
## Lattitude -0.21691878 0.009005193 -0.10109179 -0.37490363 0.01388202  
## Longtitude 0.20978618 0.096665470 0.21559399 0.43057915 0.09553354  
## Propertycount -0.05333561 -0.100447456 -0.06143317 0.05854245 -0.09843843  
## Bathroom Car Landsize BuildingArea YearBuilt  
## Price 0.49248086 0.250915900 0.07353590 0.52049158 -0.307343435  
## Rooms 0.61328467 0.420493152 0.09903096 0.60314954 -0.049272047  
## Distance 0.12404359 0.265142492 0.08244905 0.15514796 0.258461626  
## Postcode 0.11277583 0.049226356 0.03999792 0.08179100 0.036819222  
## Bedroom2 0.61683208 0.424041014 0.09806603 0.58846256 -0.038126120  
## Bathroom 1.00000000 0.335331245 0.08167965 0.53971687 0.166412095  
## Car 0.33533124 1.000000000 0.11342687 0.33170216 0.114339508  
## Landsize 0.08167965 0.113426872 1.00000000 0.08281503 0.031474094  
## BuildingArea 0.53971687 0.331702156 0.08281503 1.00000000 0.017940312  
## YearBuilt 0.16641210 0.114339508 0.03147409 0.01794031 1.000000000  
## Lattitude -0.06698307 0.003349774 0.01313278 -0.04222951 0.064333166  
## Longtitude 0.11957345 0.061410347 0.02688183 0.10715291 -0.005064202  
## Propertycount -0.06212654 -0.033258135 -0.01490947 -0.06331520 0.005116166  
## Lattitude Longtitude Propertycount  
## Price -0.216918775 0.209786178 -0.053335615  
## Rooms 0.009005193 0.096665470 -0.100447456  
## Distance -0.101091792 0.215593995 -0.061433167  
## Postcode -0.374903626 0.430579152 0.058542450  
## Bedroom2 0.013882020 0.095533540 -0.098438434  
## Bathroom -0.066983074 0.119573445 -0.062126544  
## Car 0.003349774 0.061410347 -0.033258135  
## Landsize 0.013132783 0.026881829 -0.014909475  
## BuildingArea -0.042229514 0.107152907 -0.063315199  
## YearBuilt 0.064333166 -0.005064202 0.005116166  
## Lattitude 1.000000000 -0.362765439 0.060939704  
## Longtitude -0.362765439 1.000000000 0.052395218  
## Propertycount 0.060939704 0.052395218 1.000000000

Linear Regression

data types

# Assuming 'data' is a data frame in R  
sapply(data, class)

## X Suburb Address Rooms Type   
## "integer" "character" "character" "integer" "character"   
## Price Method SellerG Date Distance   
## "numeric" "character" "character" "character" "numeric"   
## Postcode Bedroom2 Bathroom Car Landsize   
## "integer" "integer" "integer" "integer" "integer"   
## BuildingArea YearBuilt CouncilArea Lattitude Longtitude   
## "numeric" "integer" "character" "numeric" "numeric"   
## Regionname Propertycount   
## "character" "integer"

# Assuming 'data' is your data frame  
set.seed(123) # For reproducibility  
sample\_indices <- sample(nrow(data), nrow(data) \* 0.7) # Split data into 70% training and 30% testing  
train\_data <- data[sample\_indices, ]  
test\_data <- data[-sample\_indices, ]

# Perform stepwise regression for feature selection  
# Assuming you want to perform forward selection  
initial\_model <- lm(Price ~ 1, data = train\_data) # Start with an intercept-only model  
  
final\_model <- step(initial\_model, direction = "forward", scope = formula(~ .))

## Start: AIC=128790.4  
## Price ~ 1

# Train the final model with selected features  
final\_model <- lm(Price ~ Rooms + Distance + Postcode + Bedroom2 + Bathroom + Car + Landsize + BuildingArea + YearBuilt + Lattitude + Longtitude + Propertycount, data = train\_data)

# Make predictions on the test data  
predictions <- predict(final\_model, newdata = test\_data)  
  
# Assess the model's performance  
mse <- mean((test\_data$SALE.PRICE - predictions)^2)  
r\_squared <- 1 - (sum((test\_data$SALE.PRICE - predictions)^2) / sum((test\_data$SALE.PRICE - mean(test\_data$SALE.PRICE))^2))

## Warning in mean.default(test\_data$SALE.PRICE): argument is not numeric or  
## logical: returning NA

# View the model summary  
summary(final\_model)

##   
## Call:  
## lm(formula = Price ~ Rooms + Distance + Postcode + Bedroom2 +   
## Bathroom + Car + Landsize + BuildingArea + YearBuilt + Lattitude +   
## Longtitude + Propertycount, data = train\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3248086 -220983 -37509 157700 8140702   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.387e+08 9.908e+06 -13.999 < 2e-16 \*\*\*  
## Rooms 1.525e+05 2.239e+04 6.811 1.09e-11 \*\*\*  
## Distance -4.162e+04 1.352e+03 -30.777 < 2e-16 \*\*\*  
## Postcode 8.726e+02 9.081e+01 9.609 < 2e-16 \*\*\*  
## Bedroom2 1.423e+04 2.198e+04 0.647 0.51740   
## Bathroom 1.790e+05 1.256e+04 14.252 < 2e-16 \*\*\*  
## Car 6.886e+04 7.556e+03 9.114 < 2e-16 \*\*\*  
## Landsize 1.857e+01 7.083e+00 2.623 0.00876 \*\*   
## BuildingArea 2.691e+03 1.107e+02 24.303 < 2e-16 \*\*\*  
## YearBuilt -4.353e+03 1.839e+02 -23.666 < 2e-16 \*\*\*  
## Lattitude -1.187e+06 9.099e+04 -13.044 < 2e-16 \*\*\*  
## Longtitude 6.890e+05 7.082e+04 9.730 < 2e-16 \*\*\*  
## Propertycount -2.258e+00 1.487e+00 -1.519 0.12886   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 442000 on 4768 degrees of freedom  
## Multiple R-squared: 0.6103, Adjusted R-squared: 0.6093   
## F-statistic: 622.2 on 12 and 4768 DF, p-value: < 2.2e-16

# Fit the linear regression model  
final\_model <- lm(log(Price) ~ Rooms + Distance + Postcode + Bedroom2 + Bathroom + Car + Landsize + BuildingArea + YearBuilt + Lattitude + Longtitude + Propertycount, data = train\_data)  
  
# Make predictions  
predictions <- predict(final\_model, newdata = test\_data) # Replace test\_data with your test dataset  
  
# Apply the inverse log transformation to get back to the original scale  
predicted\_prices <- exp(predictions)  
  
summary(final\_model)

##   
## Call:  
## lm(formula = log(Price) ~ Rooms + Distance + Postcode + Bedroom2 +   
## Bathroom + Car + Landsize + BuildingArea + YearBuilt + Lattitude +   
## Longtitude + Propertycount, data = train\_data)  
##

DATA.CSV

**Regression Analysis: log\_price versus Rooms, Distance, Postcode, Bedroom2, Bathroom, Car, Landsize, BuildingArea, YearBuilt, Lattitude, Longtitude, Propertycount**

**Backward Elimination of Terms**

α to remove = 0.1

**Regression Equation**

|  |  |  |
| --- | --- | --- |
| log\_price | = | -139.32 + 0.1809 Rooms - 0.033772 Distance + 0.000464 Postcode + 0.0459 Bedroom2 + 0.11663 Bathroom + 0.05188 Car + 0.000011 Landsize + 0.001203 BuildingArea - 0.003850 YearBuilt - 1.0153 Lattitude + 0.8280 Longtitude - 0.000006 Propertycount |

**Coefficients**

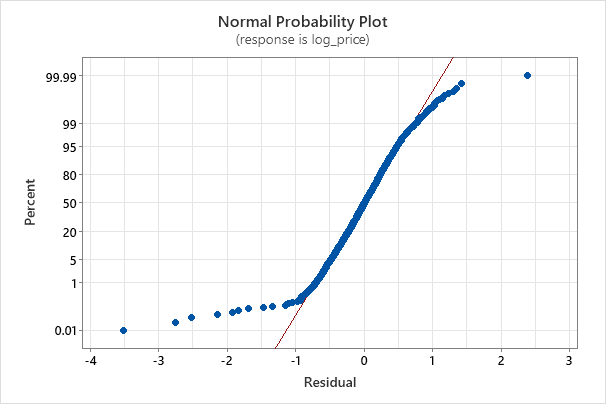
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Term** | **Coef** | **SE Coef** | **T-Value** | **P-Value** | **VIF** |
| Constant | -139.32 | 5.67 | -24.57 | 0.000 |  |
| Rooms | 0.1809 | 0.0133 | 13.65 | 0.000 | 12.11 |
| Distance | -0.033772 | 0.000785 | -43.03 | 0.000 | 1.62 |
| Postcode | 0.000464 | 0.000052 | 8.98 | 0.000 | 1.63 |
| Bedroom2 | 0.0459 | 0.0132 | 3.49 | 0.000 | 11.94 |
| Bathroom | 0.11663 | 0.00729 | 16.01 | 0.000 | 1.98 |
| Car | 0.05188 | 0.00446 | 11.62 | 0.000 | 1.30 |
| Landsize | 0.000011 | 0.000004 | 2.78 | 0.005 | 1.02 |
| BuildingArea | 0.001203 | 0.000054 | 22.36 | 0.000 | 1.71 |
| YearBuilt | -0.003850 | 0.000109 | -35.43 | 0.000 | 1.23 |
| Lattitude | -1.0153 | 0.0521 | -19.47 | 0.000 | 1.28 |
| Longtitude | 0.8280 | 0.0405 | 20.45 | 0.000 | 1.32 |
| Propertycount | -0.000006 | 0.000001 | -6.37 | 0.000 | 1.04 |

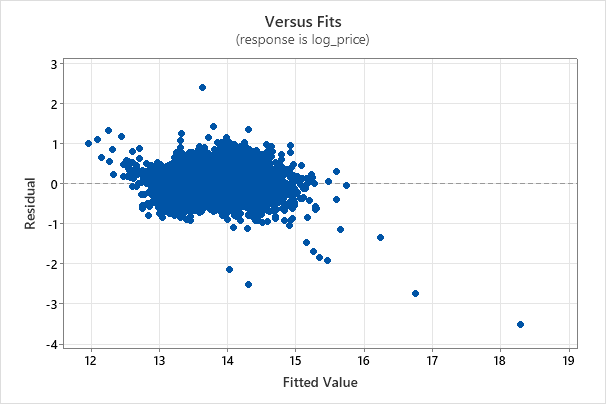
**Model Summary**

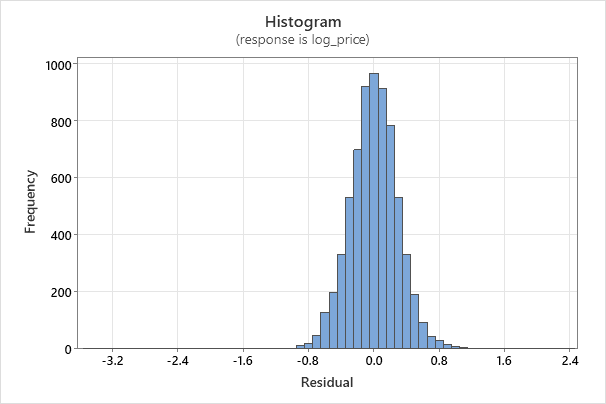
|  |  |  |  |
| --- | --- | --- | --- |
| **S** | **R-sq** | **R-sq(adj)** | **R-sq(pred)** |
| 0.305436 | 68.34% | 68.28% | 67.55% |

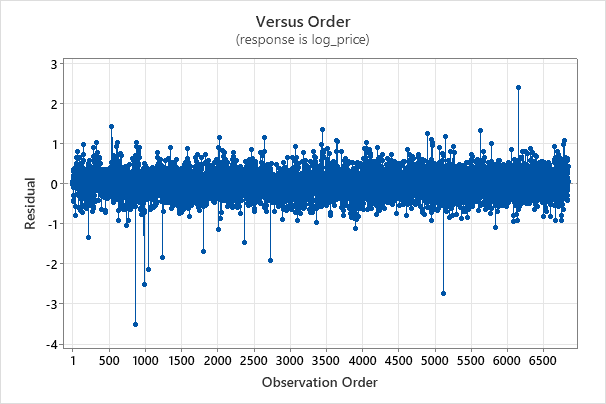
**Analysis of Variance**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Source** | **DF** | **Adj SS** | **Adj MS** | **F-Value** | **P-Value** |
| Regression | 12 | 1372.53 | 114.378 | 1226.03 | 0.000 |
| Rooms | 1 | 17.39 | 17.389 | 186.39 | 0.000 |
| Distance | 1 | 172.70 | 172.700 | 1851.20 | 0.000 |
| Postcode | 1 | 7.53 | 7.529 | 80.71 | 0.000 |
| Bedroom2 | 1 | 1.14 | 1.138 | 12.19 | 0.000 |
| Bathroom | 1 | 23.91 | 23.910 | 256.29 | 0.000 |
| Car | 1 | 12.61 | 12.607 | 135.14 | 0.000 |
| Landsize | 1 | 0.72 | 0.723 | 7.75 | 0.005 |
| BuildingArea | 1 | 46.66 | 46.660 | 500.15 | 0.000 |
| YearBuilt | 1 | 117.11 | 117.110 | 1255.32 | 0.000 |
| Lattitude | 1 | 35.36 | 35.359 | 379.01 | 0.000 |
| Longtitude | 1 | 39.01 | 39.006 | 418.11 | 0.000 |
| Propertycount | 1 | 3.79 | 3.785 | 40.58 | 0.000 |
| Error | 6817 | 635.97 | 0.093 |  |  |
| Lack-of-Fit | 6774 | 635.85 | 0.094 | 33.27 | 0.000 |
| Pure Error | 43 | 0.12 | 0.003 |  |  |
| Total | 6829 | 2008.50 |  |  |  |





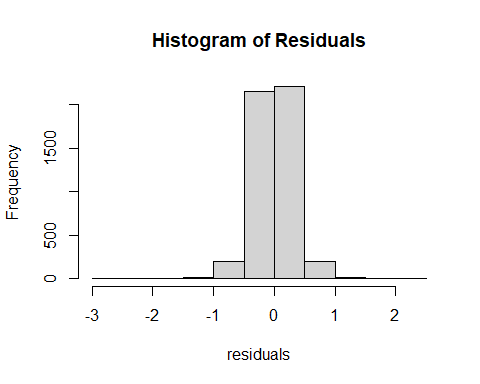




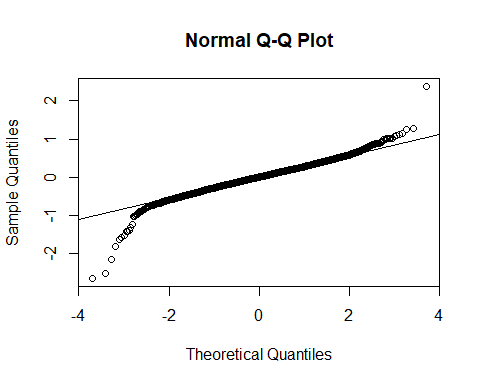
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.64419 -0.18519 0.00438 0.18996 2.37656   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.399e+02 6.836e+00 -20.462 < 2e-16 \*\*\*  
## Rooms 1.704e-01 1.545e-02 11.027 < 2e-16 \*\*\*  
## Distance -3.406e-02 9.331e-04 -36.496 < 2e-16 \*\*\*  
## Postcode 4.702e-04 6.266e-05 7.503 7.38e-14 \*\*\*  
## Bedroom2 3.750e-02 1.517e-02 2.472 0.0135 \*   
## Bathroom 9.560e-02 8.666e-03 11.031 < 2e-16 \*\*\*  
## Car 5.338e-02 5.214e-03 10.238 < 2e-16 \*\*\*  
## Landsize 7.147e-06 4.887e-06 1.462 0.1437   
## BuildingArea 1.787e-03 7.639e-05 23.388 < 2e-16 \*\*\*  
## YearBuilt -3.707e-03 1.269e-04 -29.213 < 2e-16 \*\*\*  
## Lattitude -1.055e+00 6.279e-02 -16.810 < 2e-16 \*\*\*  
## Longtitude 8.195e-01 4.887e-02 16.770 < 2e-16 \*\*\*  
## Propertycount -5.380e-06 1.026e-06 -5.244 1.64e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.305 on 4768 degrees of freedom  
## Multiple R-squared: 0.6952, Adjusted R-squared: 0.6944   
## F-statistic: 906.2 on 12 and 4768 DF, p-value: < 2.2e-16

Histogram and QQ Plot for Residuals:

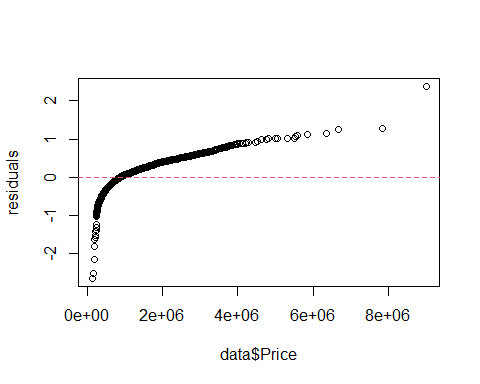
# Obtain the residuals  
residuals <- residuals(final\_model)  
  
# Create a histogram  
hist(residuals, main = "Histogram of Residuals")



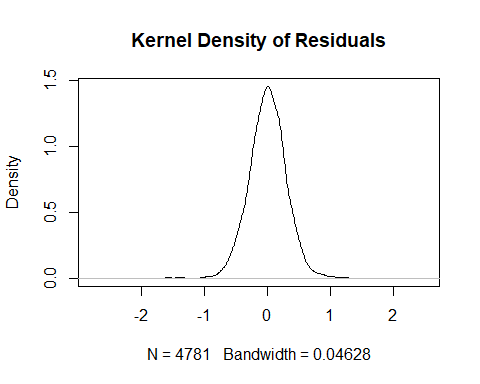
# Create a QQ plot  
qqnorm(residuals)  
qqline(residuals)



# Normal Probability Plot  
qqplot(data$Price,residuals)  
abline(h = 0, lty = 2, col = 2)



#Kernel Density Plot for Residuals:  
  
# Kernel Density Plot  
plot(density(residuals), main = "Kernel Density of Residuals")



#Shapiro-Wilk Normality Test:  
#You can perform a formal Shapiro-Wilk test for normality:  
  
  
shapiro.test(residuals)

##   
## Shapiro-Wilk normality test  
##   
## data: residuals  
## W = 0.97149, p-value < 2.2e-16

These plots and the Shapiro-Wilk test will help you assess the normality of the residuals. If the residuals closely follow a straight line in the QQ plot, have a symmetric histogram, and the p-value from the Shapiro-Wilk test is not significant (p > 0.05), it suggests that the assumption of normality is reasonable.