```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```r
library(ggplot2)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```r
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
library(caTools)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```r
mydata <- read.csv('/Users/yashagarwal/Desktop/PROJECTS/PAS project/kc_house_data.csv')

head(mydata,10)
```

```
##            id       date   price bedrooms bathrooms sqft_living sqft_lot floors
## 1  7129300520 10/13/2014  221900        3      1.00        1180     5650      1
## 2  6414100192  12/9/2014  538000        3      2.25        2570     7242      2
## 3  5631500400  2/25/2015  180000        2      1.00         770    10000      1
## 4  2487200875  12/9/2014  604000        4      3.00        1960     5000      1
## 5  1954400510  2/18/2015  510000        3      2.00        1680     8080      1
## 6  7237550310  5/12/2014 1230000        4      4.50        5420   101930      1
## 7  1321400060  6/27/2014  257500        3      2.25        1715     6819      2
## 8  2008000270  1/15/2015  291850        3      1.50        1060     9711      1
## 9  2414600126  4/15/2015  229500        3      1.00        1780     7470      1
## 10 3793500160  3/12/2015  323000        3      2.50        1890     6560      2
##    waterfront view condition grade sqft_above sqft_basement yr_built
## 1           0    0         3     7       1180             0     1955
## 2           0    0         3     7       2170           400     1951
## 3           0    0         3     6        770             0     1933
## 4           0    0         5     7       1050           910     1965
## 5           0    0         3     8       1680             0     1987
## 6           0    0         3    11       3890          1530     2001
## 7           0    0         3     7       1715             0     1995
## 8           0    0         3     7       1060             0     1963
## 9           0    0         3     7       1050           730     1960
## 10          0    0         3     7       1890             0     2003
##    yr_renovated zipcode     lat     long sqft_living15 sqft_lot15
## 1             0   98178 47.5112 -122.257          1340       5650
## 2          1991   98125 47.7210 -122.319          1690       7639
## 3             0   98028 47.7379 -122.233          2720       8062
## 4             0   98136 47.5208 -122.393          1360       5000
## 5             0   98074 47.6168 -122.045          1800       7503
## 6             0   98053 47.6561 -122.005          4760     101930
## 7             0   98003 47.3097 -122.327          2238       6819
## 8             0   98198 47.4095 -122.315          1650       9711
## 9             0   98146 47.5123 -122.337          1780       8113
## 10            0   98038 47.3684 -122.031          2390       7570
```

```
str(mydata)
```

```
## 'data.frame':    21597 obs. of  21 variables:
##  $ id           : num  7.13e+09 6.41e+09 5.63e+09 2.49e+09 1.95e+09 ...
##  $ date         : chr  "10/13/2014" "12/9/2014" "2/25/2015" "12/9/2014" ...
##  $ price        : num  221900 538000 180000 604000 510000 ...
##  $ bedrooms     : int  3 3 2 4 3 4 3 3 3 3 ...
##  $ bathrooms    : num  1 2.25 1 3 2 4.5 2.25 1.5 1 2.5 ...
##  $ sqft_living  : int  1180 2570 770 1960 1680 5420 1715 1060 1780 1890 ...
##  $ sqft_lot     : int  5650 7242 10000 5000 8080 101930 6819 9711 7470 6560 ...
##  $ floors       : num  1 2 1 1 1 1 2 1 1 2 ...
##  $ waterfront   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ view         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ condition    : int  3 3 3 5 3 3 3 3 3 3 ...
##  $ grade        : int  7 7 6 7 8 11 7 7 7 7 ...
##  $ sqft_above   : int  1180 2170 770 1050 1680 3890 1715 1060 1050 1890 ...
##  $ sqft_basement: int  0 400 0 910 0 1530 0 0 730 0 ...
##  $ yr_built     : int  1955 1951 1933 1965 1987 2001 1995 1963 1960 2003 ...
##  $ yr_renovated : int  0 1991 0 0 0 0 0 0 0 0 ...
##  $ zipcode      : int  98178 98125 98028 98136 98074 98053 98003 98198 98146 98038 ...
##  $ lat          : num  47.5 47.7 47.7 47.5 47.6 ...
##  $ long         : num  -122 -122 -122 -122 -122 ...
##  $ sqft_living15: int  1340 1690 2720 1360 1800 4760 2238 1650 1780 2390 ...
##  $ sqft_lot15   : int  5650 7639 8062 5000 7503 101930 6819 9711 8113 7570 ...
```

```
summary(mydata)
```

```
##        id                date               price            bedrooms
##  Min.   :1.000e+06   Length:21597       Min.   :  78000   Min.   : 1.000
##  1st Qu.:2.123e+09   Class :character   1st Qu.: 322000   1st Qu.: 3.000
##  Median :3.905e+09   Mode  :character   Median : 450000   Median : 3.000
##  Mean   :4.580e+09                      Mean   : 540297   Mean   : 3.373
##  3rd Qu.:7.309e+09                      3rd Qu.: 645000   3rd Qu.: 4.000
##  Max.   :9.900e+09                      Max.   :7700000   Max.   :33.000  
##    bathrooms      sqft_living       sqft_lot           floors     
##  Min.   :0.500   Min.   :  370   Min.   :    520   Min.   :1.000  
##  1st Qu.:1.750   1st Qu.: 1430   1st Qu.:   5040   1st Qu.:1.000  
##  Median :2.250   Median : 1910   Median :   7618   Median :1.500  
##  Mean   :2.116   Mean   : 2080   Mean   :  15099   Mean   :1.494  
##  3rd Qu.:2.500   3rd Qu.: 2550   3rd Qu.:  10685   3rd Qu.:2.000  
##  Max.   :8.000   Max.   :13540   Max.   :1651359   Max.   :3.500  
##    waterfront            view          condition        grade       
##  Min.   :0.000000   Min.   :0.0000   Min.   :1.00   Min.   : 3.000  
##  1st Qu.:0.000000   1st Qu.:0.0000   1st Qu.:3.00   1st Qu.: 7.000  
##  Median :0.000000   Median :0.0000   Median :3.00   Median : 7.000  
##  Mean   :0.007547   Mean   :0.2343   Mean   :3.41   Mean   : 7.658  
##  3rd Qu.:0.000000   3rd Qu.:0.0000   3rd Qu.:4.00   3rd Qu.: 8.000  
##  Max.   :1.000000   Max.   :4.0000   Max.   :5.00   Max.   :13.000  
##    sqft_above   sqft_basement       yr_built     yr_renovated   
##  Min.   : 370   Min.   :   0.0   Min.   :1900   Min.   :   0.00  
##  1st Qu.:1190   1st Qu.:   0.0   1st Qu.:1951   1st Qu.:   0.00  
##  Median :1560   Median :   0.0   Median :1975   Median :   0.00  
##  Mean   :1789   Mean   : 291.7   Mean   :1971   Mean   :  84.46  
##  3rd Qu.:2210   3rd Qu.: 560.0   3rd Qu.:1997   3rd Qu.:   0.00  
##  Max.   :9410   Max.   :4820.0   Max.   :2015   Max.   :2015.00  
##     zipcode           lat             long        sqft_living15 
##  Min.   :98001   Min.   :47.16   Min.   :-122.5   Min.   : 399  
##  1st Qu.:98033   1st Qu.:47.47   1st Qu.:-122.3   1st Qu.:1490  
##  Median :98065   Median :47.57   Median :-122.2   Median :1840  
##  Mean   :98078   Mean   :47.56   Mean   :-122.2   Mean   :1987  
##  3rd Qu.:98118   3rd Qu.:47.68   3rd Qu.:-122.1   3rd Qu.:2360  
##  Max.   :98199   Max.   :47.78   Max.   :-121.3   Max.   :6210  
##    sqft_lot15    
##  Min.   :   651  
##  1st Qu.:  5100  
##  Median :  7620  
##  Mean   : 12758  
##  3rd Qu.: 10083  
##  Max.   :871200
```

```r
NA_values=data.frame(no_of__values_=colSums(is.na(mydata)))
head(NA_values,21)
```

```
##                no_of__values_
## id                          0
## date                        0
## price                       0
## bedrooms                    0
## bathrooms                   0
## sqft_living                 0
## sqft_lot                    0
## floors                      0
## waterfront                  0
## view                        0
## condition                   0
## grade                       0
## sqft_above                  0
## sqft_basement               0
## yr_built                      0
## yr_renovated                  0
## zipcode                       0
## lat                           0
## long                          0
## sqft_living15                 0
## sqft_lot15                    0
```

```
set.seed(123)
sample=sample.split(mydata,SplitRatio = 0.8)

train_data=subset(mydata,sample==TRUE)
test_data=subset(mydata,sample==FALSE)

cor_data=data.frame(train_data[,3:21])
correlation=cor(cor_data)
par(mfrow=c(1,1))
corrplot(correlation,method = 'color')
```

PAS_PROJECT_files/figure-latex/unnamed-chunk-1-1.pdf

```
p1=ggplot(data = train_data, aes(x = bedrooms, y = price)) +
  geom_jitter() +  geom_smooth(method = "lm", se = FALSE)+labs(title="Scatter plot of Bedrooms and Pric
p2=ggplot(data = train_data, aes(x = bathrooms, y = price)) +
  geom_jitter() +  geom_smooth(method = "lm", se = FALSE)+labs(title="Scatter plot of Bathrooms and Pri
p3=ggplot(data = train_data, aes(x = sqft_living, y = price)) +
  geom_jitter() +  geom_smooth(method = "lm", se = FALSE)+labs(title="Scatter plot of Sqft_living and P
p4=ggplot(data = train_data, aes(x = sqft_above, y = price)) +
  geom_jitter() +  geom_smooth(method = "lm", se = FALSE)+labs(title="Scatter plot of Sqft_above and Pr
p5=ggplot(data = train_data, aes(x = sqft_basement, y = price)) +
  geom_jitter() +  geom_smooth(method = "lm", se = FALSE)+labs(title="Scatter plot of Sqft_basement and
p6=ggplot(data = train_data, aes(x = lat, y = price)) +
  geom_jitter() +  geom_smooth(method = "lm", se = FALSE)+labs(title="Scatter plot of Latitude and Pric
p7=ggplot(data = train_data, aes(x = sqft_living15, y = price)) +
  geom_jitter() +  geom_smooth(method = "lm", se = FALSE)+labs(title="Scatter plot of Sqft_living15 and
grid.arrange(p1,p2,p3,p4,p5,p6,p7,nrow=4)
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```

PAS_PROJECT_files/figure-latex/unnamed-chunk-1-2.pdf

```
par(mfrow=c(1, 2))
boxplot(price~view,data=train_data,main="Different boxplots", xlab="view",ylab="price",col="orange",bor
boxplot(price~grade,data=train_data,main="Different boxplots", xlab="grade",ylab="price",col="orange",b
```

PAS_PROJECT_files/figure-latex/unnamed-chunk-1-3.pdf

```
date_sale=mdy(train_data$date)
train_data$sale_date_year=as.integer(year(date_sale))
train_data$age=train_data$sale_date_year-train_data$yr_built

train_data$reno=ifelse(train_data$yr_renovated==0,0,1)
train_data$reno=as.factor(train_data$reno)

ggpairs(train_data, columns= c("price","bedrooms","bathrooms","view","grade","sqft_living","sqft_above"
```

PAS_PROJECT_files/figure-latex/unnamed-chunk-1-4.pdf

```
ggplot(data=train_data)+geom_boxplot(aes(x=bedrooms,y=price))
```

```
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```

PAS_PROJECT_files/figure-latex/unnamed-chunk-1-5.pdf

```
outliers=boxplot(train_data$price,plot=FALSE)$out
outliers_data=train_data[which(train_data$price %in% outliers),]
train_data1= train_data[-which(train_data$price %in% outliers),]


par(mfrow=c(1, 2))
plot(train_data$bedrooms, train_data$price, main="With Outliers", xlab="bedrooms", ylab="price", pch="*
abline(lm(price ~ bedrooms, data=train_data), col="blue", lwd=3, lty=2)
plot(train_data1$bedrooms, train_data1$price, main="Outliers removed", xlab="bedrooms", ylab="price", p
abline(lm(price ~bedrooms, data=train_data1), col="blue", lwd=3, lty=2)
```

PAS_PROJECT_files/figure-latex/unnamed-chunk-1-6.pdf

```
model=lm(data=train_data,price~bedrooms+bathrooms+sqft_living+view+grade+sqft_above+sqft_basement+sqft_l
summary(model)
```

```
##
## Call:
## lm(formula = price ~ bedrooms + bathrooms + sqft_living + view +
##     grade + sqft_above + sqft_basement + sqft_living15, data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1245264  -123294   -19165    96345  4659105
##
## Coefficients: (1 not defined because of singularities)
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -5.041e+05  1.650e+04 -30.549  < 2e-16 ***
## bedrooms      -2.979e+04  2.476e+03 -12.031  < 2e-16 ***
## bathrooms     -1.835e+04  3.799e+03  -4.830 1.38e-06 ***
## sqft_living    2.231e+02  5.315e+00  41.982  < 2e-16 ***
## view           8.896e+04  2.597e+03  34.255  < 2e-16 ***
## grade          1.012e+05  2.732e+03  37.057  < 2e-16 ***
## sqft_above    -4.961e+01  4.961e+00  -9.999  < 2e-16 ***
## sqft_basement         NA         NA      NA       NA
## sqft_living15  6.026e+00  4.415e+00   1.365    0.172
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 235300 on 16447 degrees of freedom
## Multiple R-squared:  0.5828, Adjusted R-squared:  0.5826
## F-statistic:  3282 on 7 and 16447 DF,  p-value: < 2.2e-16
```

```
model5=lm(data=train_data,price~bedrooms+bathrooms+sqft_living+view+grade+sqft_lot+age+floors+waterfron
summary(model5)
```

```
##
## Call:
## lm(formula = price ~ bedrooms + bathrooms + sqft_living + view +
##     grade + sqft_lot + age + floors + waterfront, data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1301702  -110143    -9314    90592  4279283
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -9.578e+05  1.761e+04 -54.401  < 2e-16 ***
## bedrooms    -3.564e+04  2.284e+03 -15.607  < 2e-16 ***
## bathrooms    5.081e+04  3.886e+03  13.075  < 2e-16 ***
## sqft_living  1.627e+02  3.753e+00  43.358  < 2e-16 ***
## view         4.992e+04  2.561e+03  19.494  < 2e-16 ***
## grade        1.270e+05  2.442e+03  52.018  < 2e-16 ***
## sqft_lot    -2.555e-01  4.076e-02  -6.267 3.76e-10 ***
## age          3.713e+03  7.310e+01  50.795  < 2e-16 ***
## floors       1.798e+04  3.883e+03   4.630 3.69e-06 ***
## waterfront   4.984e+05  2.130e+04  23.395  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 215600 on 16445 degrees of freedom
## Multiple R-squared:  0.6498, Adjusted R-squared:  0.6496
## F-statistic:  3390 on 9 and 16445 DF,  p-value: < 2.2e-16
```

```
cooksd <- cooks.distance(model5)
mean(cooksd)
```

```
## [1] 0.0002512902
```

```
par(mfrow=c(1, 1))
plot(cooksd, main="Influential Obs by Cooks distance",xlim=c(0,25000),ylim=c(0,0.1))
axis(1, at=seq(0, 25000, 5000))
axis(2, at=seq(0, 0.1, 0.0001))
abline(h = 4*mean(cooksd, na.rm=T), col="green")
text(x=1:length(cooksd)+1,y=cooksd,labels=ifelse(cooksd>4*mean(cooksd,na.rm=T),names(cooksd),""), col="
```

PAS_PROJECT_files/figure-latex/unnamed-chunk-1-7.pdf

```
influential <- as.numeric(names(cooksd)[(cooksd > 4*mean(cooksd, na.rm=T))])  # influential row numbers
head(train_data[influential, ])
```

```
##              id       date   price bedrooms bathrooms sqft_living sqft_lot
## 28  3303700376  12/1/2014  667000        3      1.00        1400     1581
## 202 2222059065 11/12/2014  297000        3      2.50        1940    14952
## 303 2747100024  6/19/2014  576000        3      2.50        1940     9000
## 315 4139480200  12/9/2014 1400000        4      3.25        4290    12103
## 348 4048400070  12/5/2014  320000        2      1.00        1070    32633
## 354 3363900111  12/3/2014  437500        2      1.00         990     3120
##     floors waterfront view condition grade sqft_above sqft_basement yr_built
## 28     1.5          0    0         5     8       1400             0     1909
## 202    2.0          0    0         3     8       1940             0     1994
## 303    1.0          0    0         4     7        970           970     1948
## 315    1.0          0    3         3    11       2690          1600     1997
## 348    1.0          0    0         4     6       1070             0     1930
## 354    1.0          0    2         5     7        790           200     1907
##     yr_renovated zipcode     lat     long sqft_living15 sqft_lot15
## 28             0   98112 47.6221 -122.314          1860       3861
## 202            0   98042 47.3777 -122.165          2030      10450
## 303            0   98117 47.6933 -122.393          2190       7310
## 315            0   98006 47.5503 -122.102          3860      11244
## 348            0   98059 47.4716 -122.078          1360      32156
## 354            0   98103 47.6800 -122.353          1930       3120
##     sale_date_year age reno
## 28            2014 105    0
## 202           2014  20    0
## 303           2014  66    0
## 315           2014  17    0
## 348           2014  84    0
## 354           2014 107    0
```

```
influential_data=train_data[influential, ]
influencial_outliers=inner_join(outliers_data,influential_data)
```

```
## Joining, by = c("id", "date", "price", "bedrooms", "bathrooms", "sqft_living",
## "sqft_lot", "floors", "waterfront", "view", "condition", "grade", "sqft_above",
## "sqft_basement", "yr_built", "yr_renovated", "zipcode", "lat", "long",
## "sqft_living15", "sqft_lot15", "sale_date_year", "age", "reno")
```

```
train_data2=rbind(train_data1,influencial_outliers)

model6=lm(data=train_data2,price~bedrooms+bathrooms+sqft_living+view+grade+sqft_lot+age+floors+waterfron
summary(model6)
```

```
##
## Call:
## lm(formula = price ~ bedrooms + bathrooms + sqft_living + view +
##     grade + sqft_lot + age + floors + waterfront, data = train_data2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -717735  -94075   -7246   82346 2479259
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.592e+05  1.239e+04 -53.220  < 2e-16 ***
## bedrooms    -1.207e+04  1.565e+03  -7.713  1.3e-14 ***
## bathrooms    3.039e+04  2.668e+03  11.390  < 2e-16 ***
## sqft_living  8.559e+01  2.718e+00  31.496  < 2e-16 ***
## view         2.708e+04  1.916e+03  14.137  < 2e-16 ***
## grade        1.025e+05  1.704e+03  60.151  < 2e-16 ***
## sqft_lot    -1.051e-02  2.801e-02  -0.375 0.707512
## age          2.739e+03  5.080e+01  53.925  < 2e-16 ***
## floors       3.384e+04  2.636e+03  12.837  < 2e-16 ***
## waterfront   7.744e+04  2.084e+04   3.716 0.000203 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 140800 on 15586 degrees of freedom
## Multiple R-squared:  0.5624, Adjusted R-squared:  0.5621
## F-statistic:  2225 on 9 and 15586 DF,  p-value: < 2.2e-16
```

```
model12=lm(data=train_data2,price~bedrooms+bathrooms+sqft_living+view+grade+age+waterfront+long+lat+zipc
summary(model12)
```

```
##
## Call:
## lm(formula = price ~ bedrooms + bathrooms + sqft_living + view +
##     grade + age + waterfront + long + lat + zipcode + condition +
##     sqft_above + sqft_living15 + reno, data = train_data2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -540111  -75304   -7817   63497 2463513
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -1.330e+07  1.983e+06  -6.708 2.04e-11 ***
## bedrooms      -1.004e+04  1.328e+03  -7.561 4.24e-14 ***
## bathrooms      3.089e+04  2.245e+03  13.758  < 2e-16 ***
## sqft_living    6.137e+01  3.134e+00  19.583  < 2e-16 ***
## view           3.259e+04  1.667e+03  19.557  < 2e-16 ***
## grade          7.661e+04  1.553e+03  49.337  < 2e-16 ***
## age            1.652e+03  5.076e+01  32.555  < 2e-16 ***
## waterfront     1.351e+05  1.773e+04   7.619 2.71e-14 ***
## long          -5.634e+04  8.913e+03  -6.321 2.66e-10 ***
## lat            5.563e+05  7.320e+03  75.997  < 2e-16 ***
## zipcode       -2.097e+02  2.310e+01  -9.081  < 2e-16 ***
## condition      2.562e+04  1.634e+03  15.684  < 2e-16 ***
## sqft_above     1.810e+01  2.880e+00   6.285 3.36e-10 ***
## sqft_living15  4.347e+01  2.589e+00  16.788  < 2e-16 ***
## reno1          2.812e+04  5.449e+03   5.161 2.49e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 119400 on 15581 degrees of freedom
## Multiple R-squared:  0.6856, Adjusted R-squared:  0.6853
## F-statistic:  2427 on 14 and 15581 DF,  p-value: < 2.2e-16
```

```r
#accuracy on train data
pred=model12$fitted.values

tally_table=data.frame(actual=train_data2$price, predicted=pred)

mape=mean(abs(tally_table$actual-tally_table$predicted)/tally_table$actual)
accuracy=1-mape
accuracy
```

```
## [1] 0.7946321
```

```r
cat("THE ACCURACY IS: ",accuracy)
```

```
## THE ACCURACY IS:  0.7946321
```

```r
date_sale1=mdy(test_data$date)
test_data$sale_date_year=as.integer(year(date_sale1))
test_data$age=test_data$sale_date_year-test_data$yr_built

test_data$reno=ifelse(test_data$yr_renovated==0,0,1)
test_data$reno=as.factor(test_data$reno)

test_data_1=test_data[,c(4,5,6,10,9,12,23,24,17,18,19,11,13,20)]

pred_test=predict(newdata=test_data_1,model12)
```

```
#accuracy on test data
tally_table_1=data.frame(actual=test_data$price, predicted=pred_test)

mape_test=mean(abs(tally_table_1$actual-tally_table_1$predicted)/tally_table_1$actual)
accuracy_test=1-mape_test
accuracy_test
```

```
## [1] 0.789063
```

```
cat("Thus our model can predict price with an accuracy of: ",accuracy_test)
```

```
## Thus our model can predict price with an accuracy of:  0.789063
```