Data Visualization and Analysis

Semester Project

Amrit Kaur
Pratik Kumar Agarwal
Mayuri Mendke
Nofel Mahmood

Dataset

https://www.kaggle.com/c/house-prices-advanced-regression-techniques

Motivation

The Dataset has 79 explanatory variables describing (almost) every aspect of residential homes in Ames, Iowa. The problem was to predict sale prices of the houses based on the variables. The problem was interesting to us because we had studied linear regression and other models to solve problems like this.

Approach

As there were a lot of variables we tried to first find out the ones which were influencing the house price on a higher level and then used them to build our linear model to predict house sale prices. After that we applied time series to forecast house prices for the next 10 years. In the end we applied clustering to group houses with respect to sale price (low, mid, high) and use inference tree to show sale prices with respect to years in which the particular house was built.

Exploration / Visualizations

Install Pacman

```
library(pacman)
```

Load all required packages

```
p_load(tidyverse, stringr,lubridate, ggplot2, tseries, forecast, scales, party)

house_training_data <- read.csv("./DataSet/train.csv")
house_test_data <- read.csv("./DataSet/test.csv")
```

Get the idea of Minimum and Maximum Price of the house along with mean and others.

```
summary(house_training_data$SalePrice)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   34900  129975  163000  180921  214000  755000
```

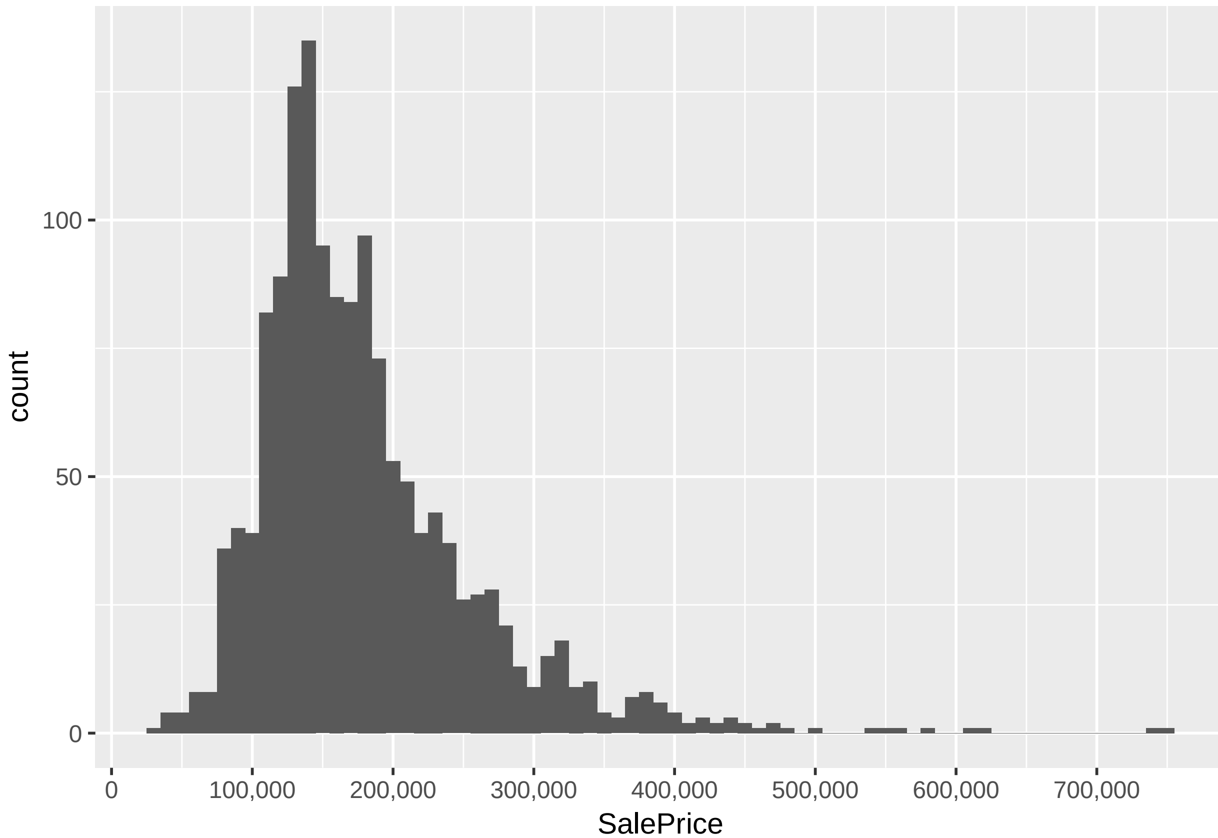
Add saleprice column to the test data. And assigned it to a new variable. Combine both the training and test data. It will be easier for analysis. From now on we will work on this dataset.

```r
house_test_data.SalePrice <-
  data.frame(SalePrice = rep(NA, nrow(house_test_data)), house_test_data[,])

house_test_data.SalePrice <-
  data.frame(SalePrice = rep(NA, nrow(house_test_data)), house_test_data[,])
house_combined <- rbind(house_training_data, house_test_data.SalePrice)

dim(house_combined)  #Dimention of the combined dataset.
```

```
## [1] 2919   81
```

With this plot we can say: Few people can afford very expensive houses. Majority of people bought houses in the range 1,00,000 to 2,50,000.

```r
training_data <- house_training_data[!is.na(house_combined$SalePrice),]
training_data %>% ggplot(aes(x=SalePrice)) +
                  geom_histogram(binwidth = 10000) +
                  scale_x_continuous(breaks= seq(0, 800000, by=100000), labels = comma)
```



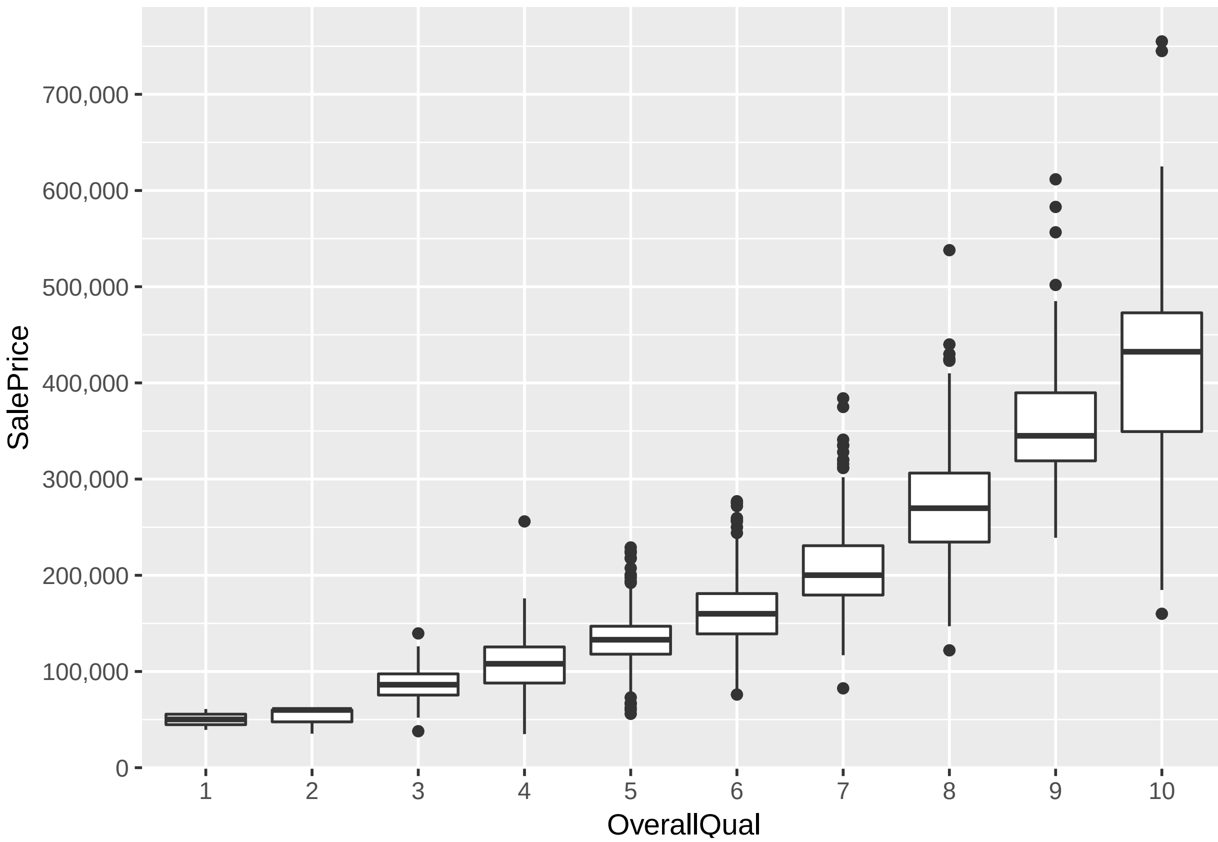
Now we have to find which attributes are more significant for SalePrice.

```r
#We don't need ID. So drop ID column from house_combined
house_training_data$Id <- NULL

#Here we have selected only those variables which has type numeric.
#Now we can check there correlation with SalePrice.
numeric.type.variables <- which(sapply(house_training_data, is.numeric))
numeric.type.name.variables <- names(numeric.type.variables)
```

```r
cor.numeric.variables <- cor(house_training_data[, numeric.type.variables],
                             use="pairwise.complete.obs")
#Lot of NA's .
#so we use="pairwise.complete.obs".

#sort the correlation with saleprice in decreasing order.
#So we will get the highly correlated variable at the top.
cor_sorted <- as.matrix(sort(cor.numeric.variables[,'SalePrice'], decreasing = TRUE))
colnames(cor_sorted)<- c("values")

#Select only high correlation
CorHigh <- names(which(apply(cor_sorted, 1, function(x) abs(x)>0.5)))
#So we got "OverallQual" as the highly significant variable for Saleprice and after that
#we "GrLivArea" and so on..

model_OverallQual<-lm(SalePrice~OverallQual, data = house_training_data)
summary(model_OverallQual)
```

```
##
## Call:
## lm(formula = SalePrice ~ OverallQual, data = house_training_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -198152  -29409   -1845   21463  396848
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -96206.1     5756.4  -16.71   <2e-16 ***
## OverallQual  45435.8      920.4   49.36   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 48620 on 1458 degrees of freedom
## Multiple R-squared:  0.6257, Adjusted R-squared:  0.6254
## F-statistic:  2437 on 1 and 1458 DF,  p-value: < 2.2e-16
```

```r
ggplot(house_training_data[!is.na(house_training_data$SalePrice),],
       aes(x= factor(OverallQual), y = SalePrice)) +
       geom_boxplot() + labs(x = "OverallQual", y = "SalePrice") +
       scale_y_continuous(breaks= seq(0, 800000, by=100000), labels = comma)
```




```
#We can Clearly see that increase in the overall quality of the house has increased the
#saleprice of the house.
```

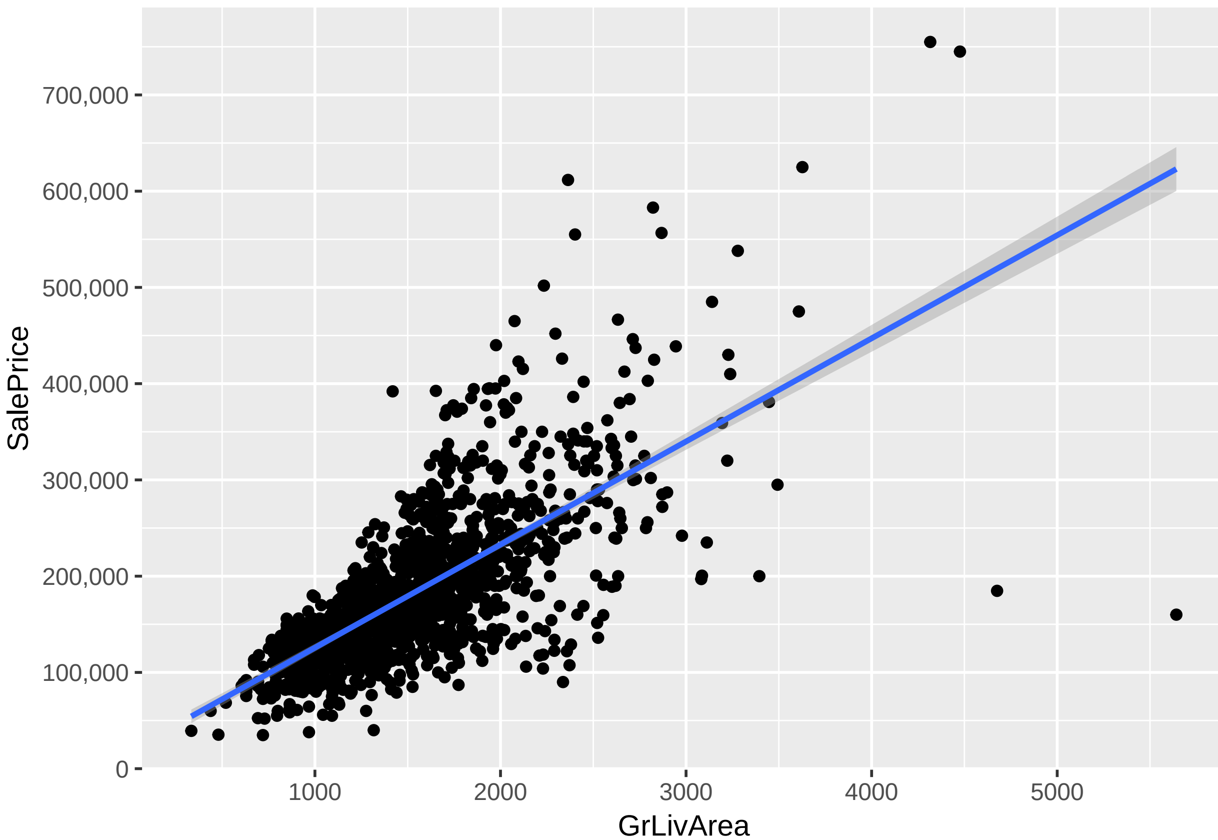
Our models

```
model_GrLiveArea<-lm(SalePrice~GrLivArea, data = house_training_data)
summary(model_GrLiveArea)
```

```
##
## Call:
## lm(formula = SalePrice ~ GrLivArea, data = house_training_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -462999  -29800   -1124   21957  339832
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 18569.026   4480.755   4.144 3.61e-05 ***
## GrLivArea     107.130      2.794  38.348  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 56070 on 1458 degrees of freedom
## Multiple R-squared:  0.5021, Adjusted R-squared:  0.5018
```

```
## F-statistic:  1471 on 1 and 1458 DF,  p-value: < 2.2e-16
```

```
ggplot(house_training_data[!is.na(house_training_data$SalePrice),],
       aes(x= GrLivArea, y = SalePrice)) + geom_point() +
       geom_smooth(method = "lm") +  labs(x = "GrLivArea", y = "SalePrice") +
       scale_y_continuous(breaks= seq(0, 800000, by=100000), labels = comma)
```




```
#Next highly correlated variable was "GrLivArea" i.e ground living area square feet.
#As the house with bigger living area will have high sale price.
#The two dots at the bottom right seems to be the outliers.

factor.type.variables.names<- which(sapply(house_training_data, is.factor))%>% names()
model_Street_Neighborhood <- lm(SalePrice ~ Street+Neighborhood+GarageCond+
                                  KitchenQual+MiscFeature,
                                data = house_training_data)
summary(model_Street_Neighborhood)
```

```
##
## Call:
## lm(formula = SalePrice ~ Street + Neighborhood + GarageCond +
##     KitchenQual + MiscFeature, data = house_training_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -47070 -17561      0  14025  94121
##
## Coefficients: (1 not defined because of singularities)
##                        Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)           253266      65834   3.847 0.000580 ***
## StreetPave           -134950      38966  -3.463 0.001628 **
## NeighborhoodClearCr   128389      34430   3.729 0.000799 ***
## NeighborhoodCollgCr   114820      31750   3.616 0.001083 **
## NeighborhoodCrawfor   156389      41890   3.733 0.000790 ***
## NeighborhoodEdwards    17132      29905   0.573 0.571006
## NeighborhoodGilbert    83296      28026   2.972 0.005781 **
## NeighborhoodIDOTRR   -156569      43968  -3.561 0.001255 **
## NeighborhoodMitchel    57250      25774   2.221 0.034028 *
## NeighborhoodNAmes      40727      21943   1.856 0.073294 .
## NeighborhoodNWAmes     88722      26059   3.405 0.001900 **
## NeighborhoodOldTown    67879      25612   2.650 0.012714 *
## NeighborhoodSawyer     34149      26383   1.294 0.205422
## NeighborhoodSawyerW    78889      41890   1.883 0.069397 .
## NeighborhoodTimber        NA         NA      NA       NA
## GarageCondFa           -9278      45071  -0.206 0.838298
## GarageCondTA            7227      35223   0.205 0.838814
## KitchenQualGd          23879      32403   0.737 0.466887
## KitchenQualTA           7768      29998   0.259 0.797427
## MiscFeatureOthr       -41039      42463  -0.966 0.341535
## MiscFeatureShed       -39312      26147  -1.503 0.143174
## MiscFeatureTenC        11855      48023   0.247 0.806694
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 33750 on 30 degrees of freedom
##   (1409 observations deleted due to missingness)
## Multiple R-squared:  0.7424, Adjusted R-squared:  0.5707
## F-statistic: 4.323 on 20 and 30 DF,  p-value: 0.000161
```

```r
#Here, we can see that street and Neighbourhood are significant variables effecting the
#SalesPrice of an house. The dummy Variables StreetPave,NeighborhoodCollgCr,
#NeighborhoodCrawfor are most significant.
#The model is very good because it has a high R value.
#Similarly we checked for other variables.

#So in our final model we are using Neighborhood, OverallQual, GrLiveArea,
#GarageCars, BsmtCond, TotalBsmtSF for prediction on our test data.

model_trained <- lm(SalePrice~Neighborhood+BsmtQual+OverallQual+GrLivArea+
                      GarageCars+TotalBsmtSF, data = house_training_data)
#Our model is trained . Now we will predict SalePrice on test dataset
summary(model_trained)
```

```
##
## Call:
## lm(formula = SalePrice ~ Neighborhood + BsmtQual + OverallQual +
##     GrLivArea + GarageCars + TotalBsmtSF, data = house_training_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -393727  -13988     386   13727  243446
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)          11829.852  12680.590   0.933 0.351028
## NeighborhoodBlueste -15224.189  25421.562  -0.599 0.549358
## NeighborhoodBrDale  -20263.602  12325.510  -1.644 0.100394
## NeighborhoodBrkSide   1190.652   9975.274   0.119 0.905007
## NeighborhoodClearCr  33718.601  10759.627   3.134 0.001762 **
## NeighborhoodCollgCr  19040.659   8716.814   2.184 0.029102 *
## NeighborhoodCrawfor  29941.376   9839.189   3.043 0.002386 **
## NeighborhoodEdwards  -7562.292   9464.814  -0.799 0.424433
## NeighborhoodGilbert  13694.632   9219.655   1.485 0.137671
## NeighborhoodIDOTRR  -12619.749  10622.504  -1.188 0.235028
## NeighborhoodMeadowV  -3292.073  12091.226  -0.272 0.785455
## NeighborhoodMitchel    715.539   9822.991   0.073 0.941941
## NeighborhoodNAmes     4474.325   9052.653   0.494 0.621204
## NeighborhoodNPkVill  -9460.267  14040.506  -0.674 0.500561
## NeighborhoodNWAmes    5227.460   9335.331   0.560 0.575593
## NeighborhoodNoRidge  71869.309  10034.889   7.162 1.28e-12 ***
## NeighborhoodNridgHt  50313.722   9308.709   5.405 7.62e-08 ***
## NeighborhoodOldTown -15860.852   9475.541  -1.674 0.094380 .
## NeighborhoodSWISU   -12498.425  11334.034  -1.103 0.270333
## NeighborhoodSawyer    7636.545   9643.275   0.792 0.428552
## NeighborhoodSawyerW  14209.211   9496.121   1.496 0.134798
## NeighborhoodSomerst  23196.852   9031.725   2.568 0.010321 *
## NeighborhoodStoneBr  64829.838  10734.032   6.040 1.98e-09 ***
## NeighborhoodTimber   24093.687   9947.805   2.422 0.015562 *
## NeighborhoodVeenker  49842.348  13140.223   3.793 0.000155 ***
## BsmtQualFa          -50991.507   7873.263  -6.477 1.30e-10 ***
## BsmtQualGd          -46898.758   4109.605 -11.412  < 2e-16 ***
## BsmtQualTA          -47116.719   5002.058  -9.419  < 2e-16 ***
## OverallQual          14458.709   1183.002  12.222  < 2e-16 ***
## GrLivArea               45.833      2.453  18.681  < 2e-16 ***
## GarageCars           12245.754   1695.136   7.224 8.28e-13 ***
## TotalBsmtSF             19.933      2.949   6.759 2.04e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 33880 on 1391 degrees of freedom
##   (37 observations deleted due to missingness)
## Multiple R-squared:  0.8218, Adjusted R-squared:  0.8178
## F-statistic: 206.9 on 31 and 1391 DF,  p-value: < 2.2e-16
```

```
pred_lm <- predict.lm(model_trained, house_test_data.SalePrice)
house_test_data_with_predictions <- house_test_data.SalePrice %>%
  mutate(predictedSalePrice = pred_lm)
```

Time Series

```
#timeseries object for Sales Price
actual_preds <- data.frame(cbind(actuals=house_test_data.SalePrice$SalePrice,
predicteds = pred_lm))
salePricets<-ts(actual_preds$predicteds,start=c(2001,1),end=c(2010,12),frequency = 4);
```

```r
#timeseries object for Sales Price and Selling Year
yrSoldts<-ts(house_test_data_with_predictions$YrSold,start=c(2001,1),
             end=c(2010,12),frequency = 4);
salePricets<-ts(house_test_data_with_predictions$predictedSalePrice,
                start=c(2001,1),end=c(2010,12),frequency = 4);

#Checking for frequency data has been collected.
frequency(salePricets);
```

```
## [1] 4
```

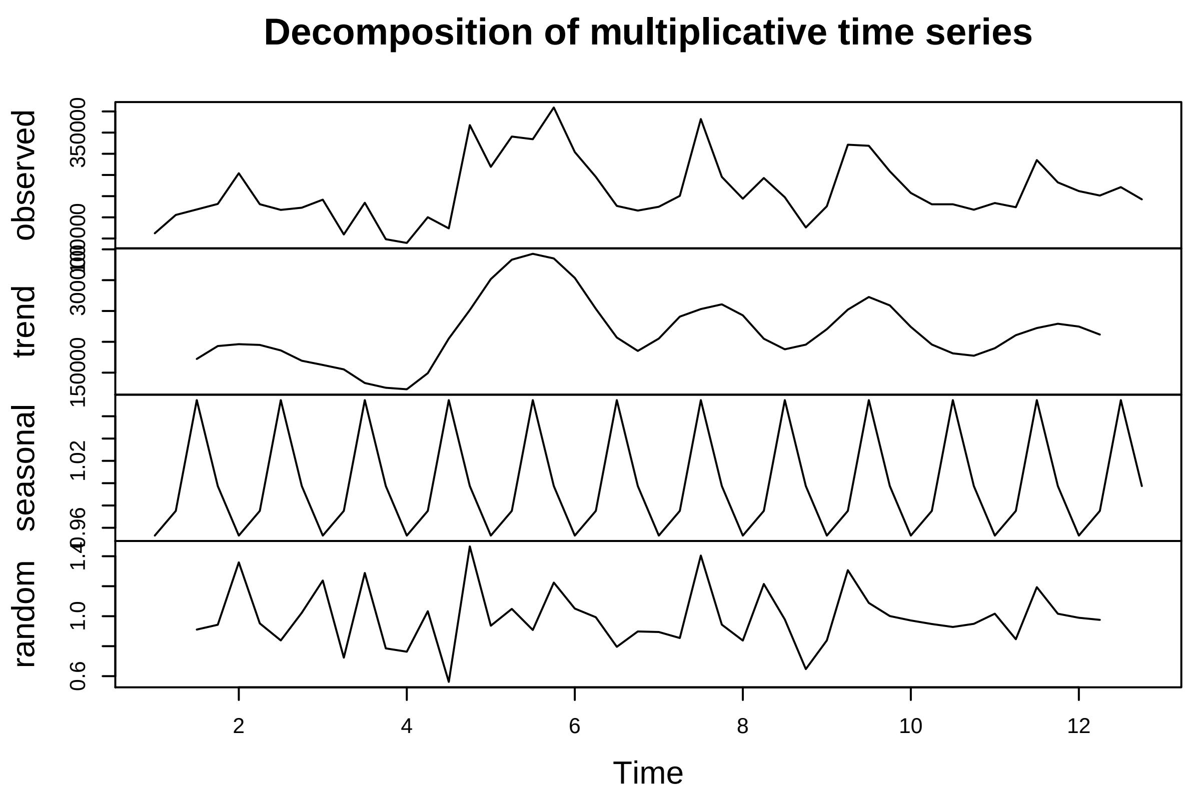
```r
#checking for missing values
sum(is.na(salePricets))
```

```
## [1] 0
```

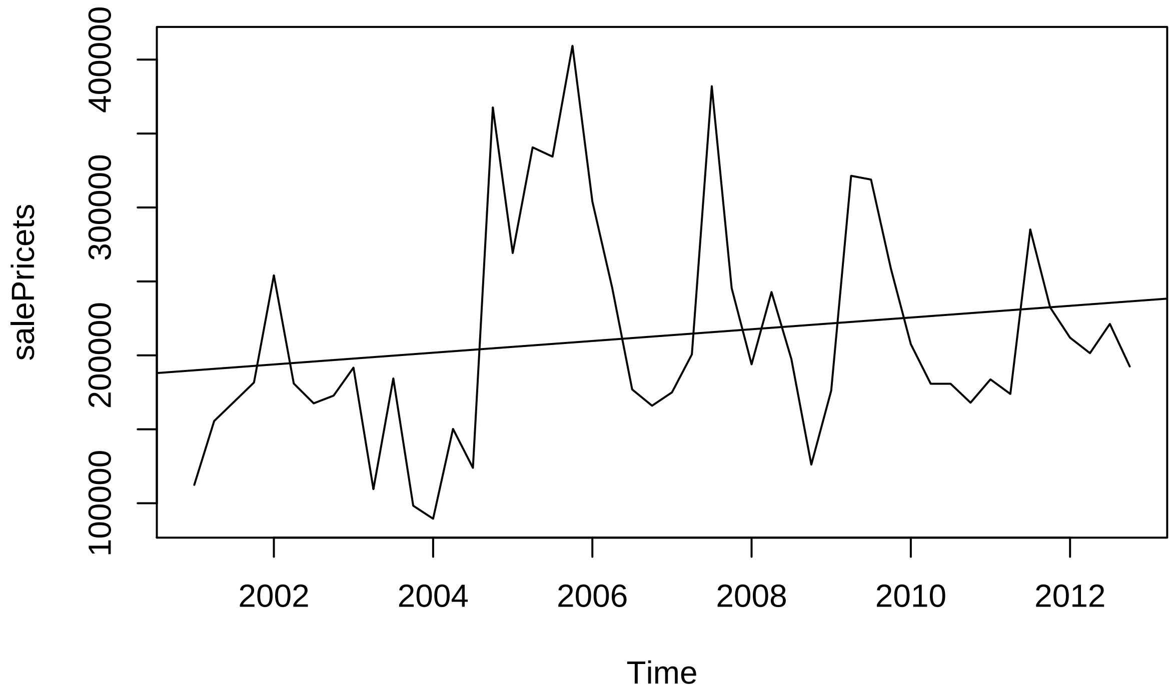
```r
#summary of the data
summary(salePricets)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   89503  171699  192087  213181  247655  409304
```

```r
#decomposing the data into trend, seasonal, regular and random components
tsdata<-ts(salePricets,frequency = 4)
ddata<-decompose(tsdata,"multiplicative")
plot(ddata)
```



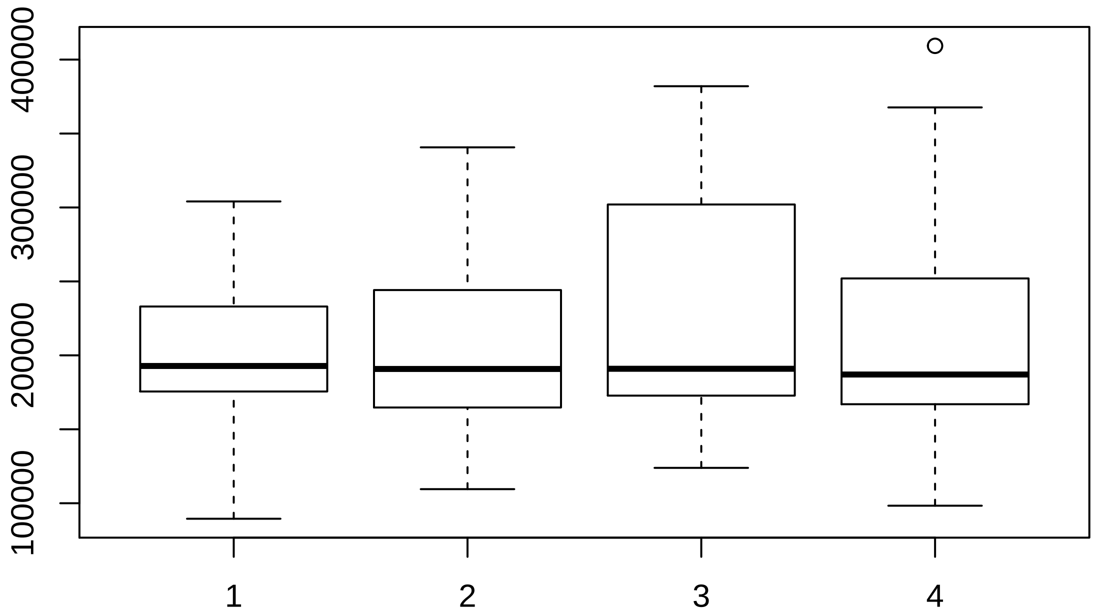

```r
#checking the original trend in data while performing linear regression.
plot(salePricets)
abline(reg=lm(salePricets~time(salePricets)))
```




```
cycle(salePricets)
```

```
##      Qtr1 Qtr2 Qtr3 Qtr4
## 2001    1    2    3    4
## 2002    1    2    3    4
## 2003    1    2    3    4
## 2004    1    2    3    4
## 2005    1    2    3    4
## 2006    1    2    3    4
## 2007    1    2    3    4
## 2008    1    2    3    4
## 2009    1    2    3    4
## 2010    1    2    3    4
## 2011    1    2    3    4
## 2012    1    2    3    4
```

```
#boxplot for quaterly data to analyse in which quater sales price is going up
boxplot(salePricets ~cycle(salePricets, xlab="Date"))
```




```r
#checking for the best model
priceModel<-auto.arima(salePricets)
priceModel
```

```
## Series: salePricets
## ARIMA(1,0,0) with non-zero mean
##
## Coefficients:
##          ar1       mean
##       0.5183  210572.87
## s.e.  0.1237   18628.69
##
## sigma^2 estimated as 4.196e+09:  log likelihood=-599.02
## AIC=1204.04   AICc=1204.59   BIC=1209.66
```

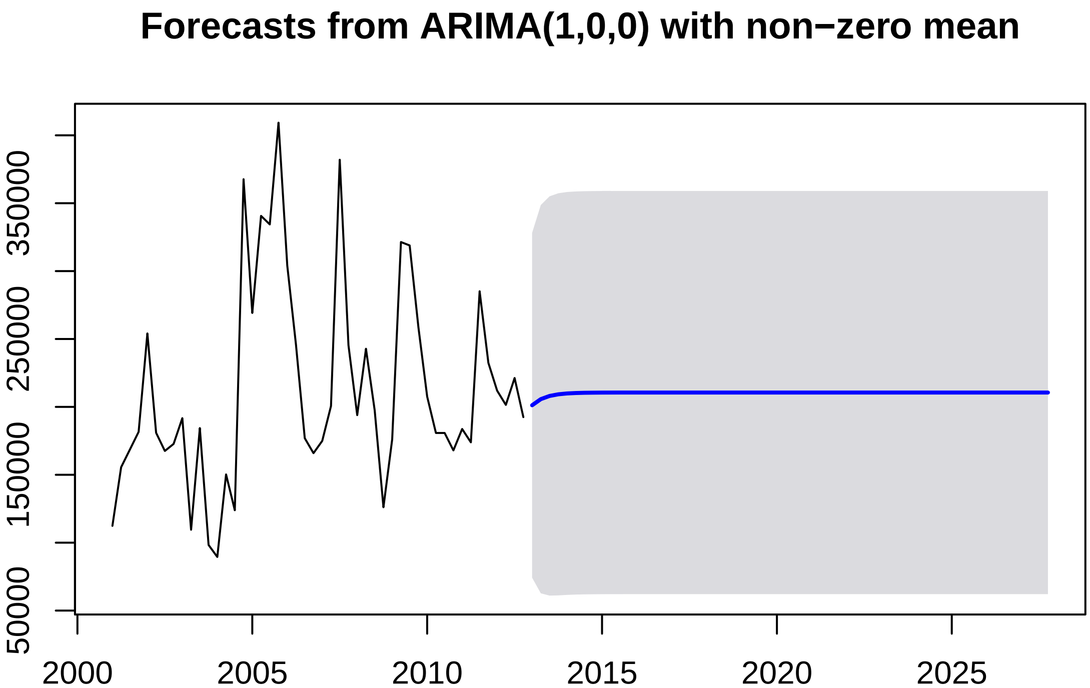
```r
#running with trace to compare the information criterion
auto.arima(salePricets,ic="aic",trace= TRUE)
```

```
##
##  ARIMA(2,0,2)(1,0,1)[4] with non-zero mean : Inf
##  ARIMA(0,0,0)           with non-zero mean : 1216.753
##  ARIMA(1,0,0)(1,0,0)[4] with non-zero mean : 1205.981
##  ARIMA(0,0,1)(0,0,1)[4] with non-zero mean : 1210.273
##  ARIMA(0,0,0)           with zero mean     : 1321.611
##  ARIMA(1,0,0)           with non-zero mean : 1204.044
##  ARIMA(1,0,0)(0,0,1)[4] with non-zero mean : 1205.977
##  ARIMA(1,0,0)(1,0,1)[4] with non-zero mean : Inf
##  ARIMA(2,0,0)           with non-zero mean : 1205.96
##  ARIMA(1,0,1)           with non-zero mean : 1205.992
##  ARIMA(2,0,1)           with non-zero mean : 1207.638
##  ARIMA(1,0,0)           with zero mean     : 1215.511
##
##  Best model: ARIMA(1,0,0)           with non-zero mean
```

```
## Series: salePricets
## ARIMA(1,0,0) with non-zero mean
```

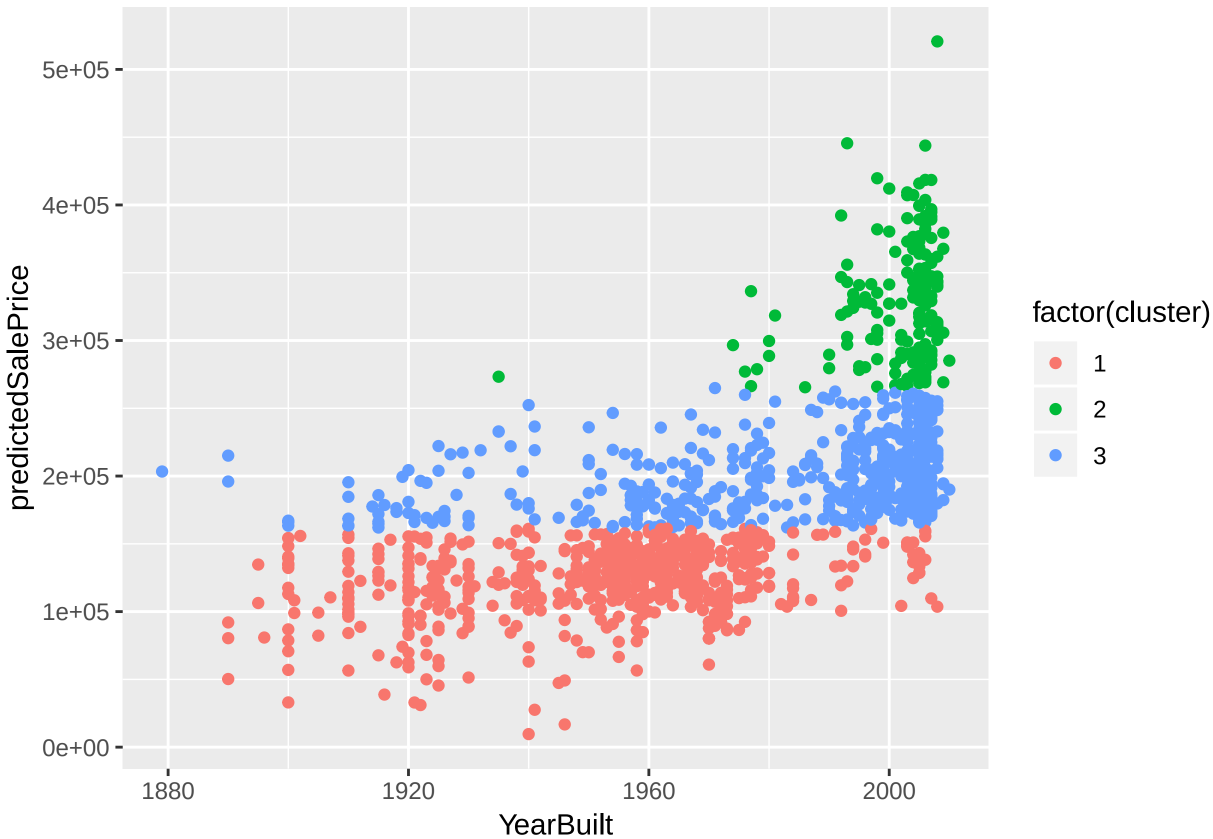
```
##
## Coefficients:
##          ar1       mean
##       0.5183  210572.87
## s.e.  0.1237   18628.69
##
## sigma^2 estimated as 4.196e+09:  log likelihood=-599.02
## AIC=1204.04   AICc=1204.59   BIC=1209.66
```

```r
#Using the model to forecast for next 5 years with 95% accuracy
priceForecast<-forecast(priceModel,level=c(95),h=5*12)
plot(priceForecast)
```

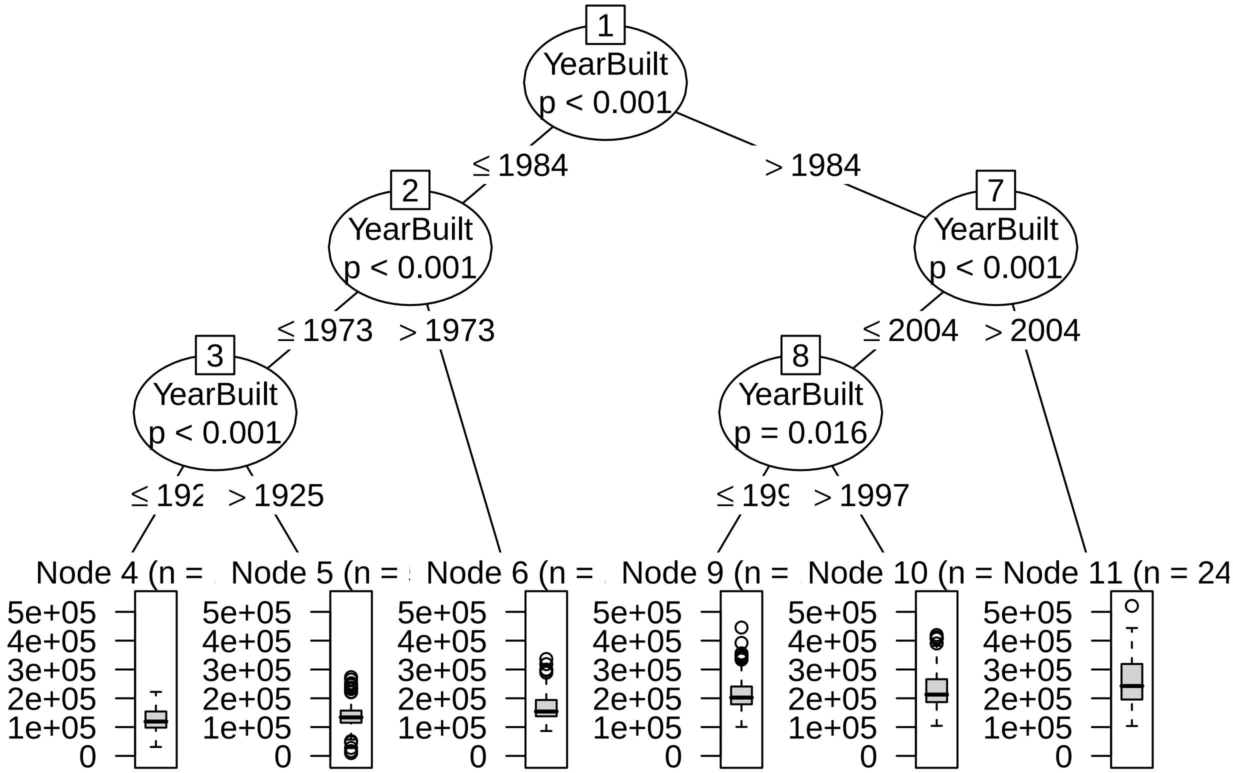


Clustering

```r
# Get Predicated Sale Price with Year Built
sale_price_with_built_year <- house_test_data_with_predictions %>%
  select(YearBuilt, predictedSalePrice) %>% na.omit()

cluster <- kmeans(sale_price_with_built_year, 3)$cluster
cbind(sale_price_with_built_year, cluster) %>%
  ggplot((aes(x = YearBuilt, y = predictedSalePrice, color = factor(cluster)))) +
  geom_point()
```




```
tree <- ctree(predictedSalePrice ~ ., data = sale_price_with_built_year,
controls = ctree_control(minbucket = 100))
plot(tree)
```



Conclusion

We saw that the variables which we used to build our linear model were effecting the sale price on a higher level such as Neighborhood, BsmtQual, OverallQual, GrLivArea, GarageCars, TotalBsmtSF. Then we used Time Series to forecast sale prices for the next 10 years. In the end we saw by applying k-means clustering that house prices with respect to the year they were built in can be clustered into high, low and mid sale prices. We can see that the most expensive houses can be found after 1980(year built) (approx).