Random Forest Car Price

Read data set and clean data

Which variables are significant in predicting the price of a car?

How well those variables describe the price of a car ?

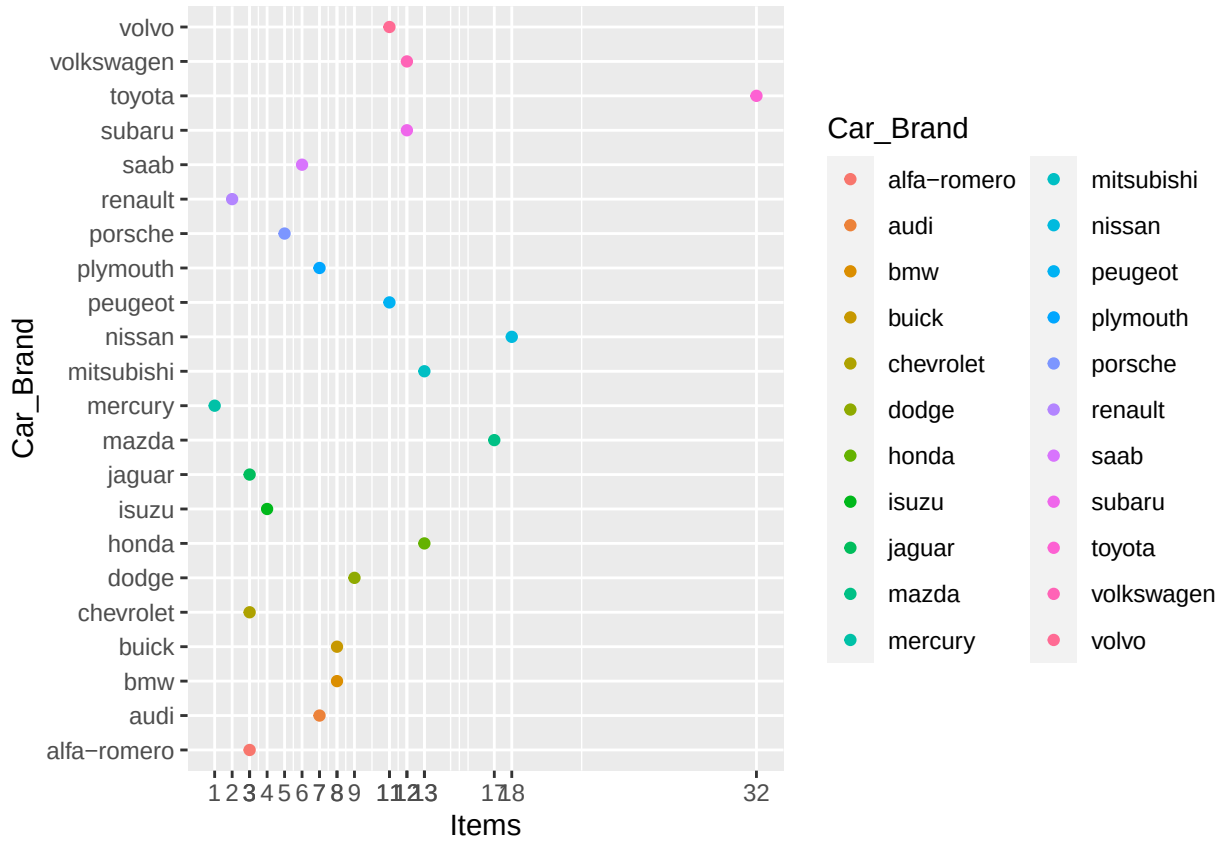
```
#Read data sets
car_data<-read.csv('Cars_Data.csv')
car_data2<-read.csv('Cars_Data.csv')

# Fix the car names
car_data$CarName<-gsub("maxda",'mazda',car_data$CarName)
car_data$CarName<-gsub("porcshce",'porsche',car_data$CarName)
car_data$CarName<-gsub("vokswagen",'volkswagen',car_data$CarName)
car_data$CarName<-gsub("vw",'volkswagen',car_data$CarName)
car_data$CarName<-gsub("toyouta",'toyota',car_data$CarName)
car_data$CarName<-gsub("Nissan",'nissan',car_data$CarName)

#add brand name column
brand<-car_data$CarName<-word(car_data$CarName,1)
car_data$brand<-brand

#How many cars of each brand?
car_count<-table (car_data$brand)
print(car_count)
```

```
##
## alfa-romero        audi         bmw       buick   chevrolet       dodge
##           3           7           8           8           3           9
##       honda       isuzu      jaguar       mazda     mercury  mitsubishi
##          13           4           3          17           1          13
##      nissan     peugeot    plymouth     porsche     renault        saab
##          18          11           7           5           2           6
##      subaru      toyota  volkswagen       volvo
##          12          32          12          11
```

```
car_count_df <- as.data.frame(car_count, check.names = FALSE)
names(car_count_df)[1]<-paste("Car_Brand")
names(car_count_df)[2]<-paste("Items")
car_count_df$Var1<-as.character(car_count_df$Car_Brand)
```

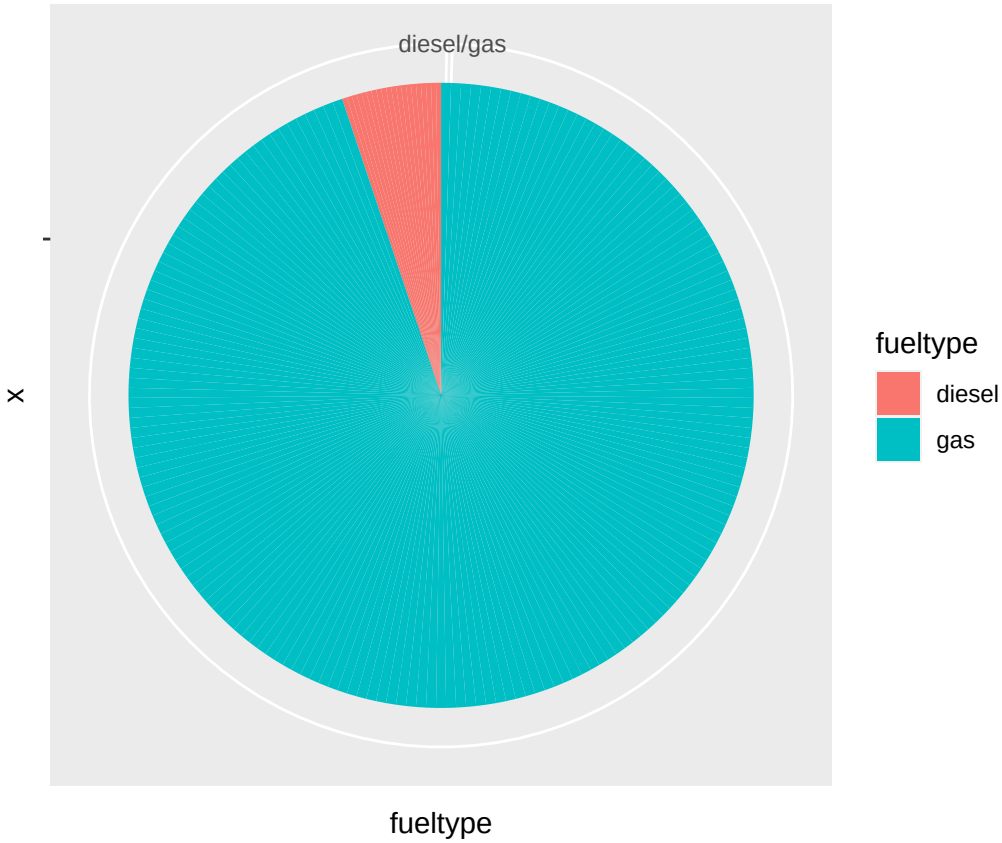
Plot of the number of cars of each brand

```
ggplot(car_count_df,aes(x=Items,y=Car_Brand))+geom_point(aes(color= Car_Brand))+
scale_x_continuous(breaks = car_count_df$Items)
```



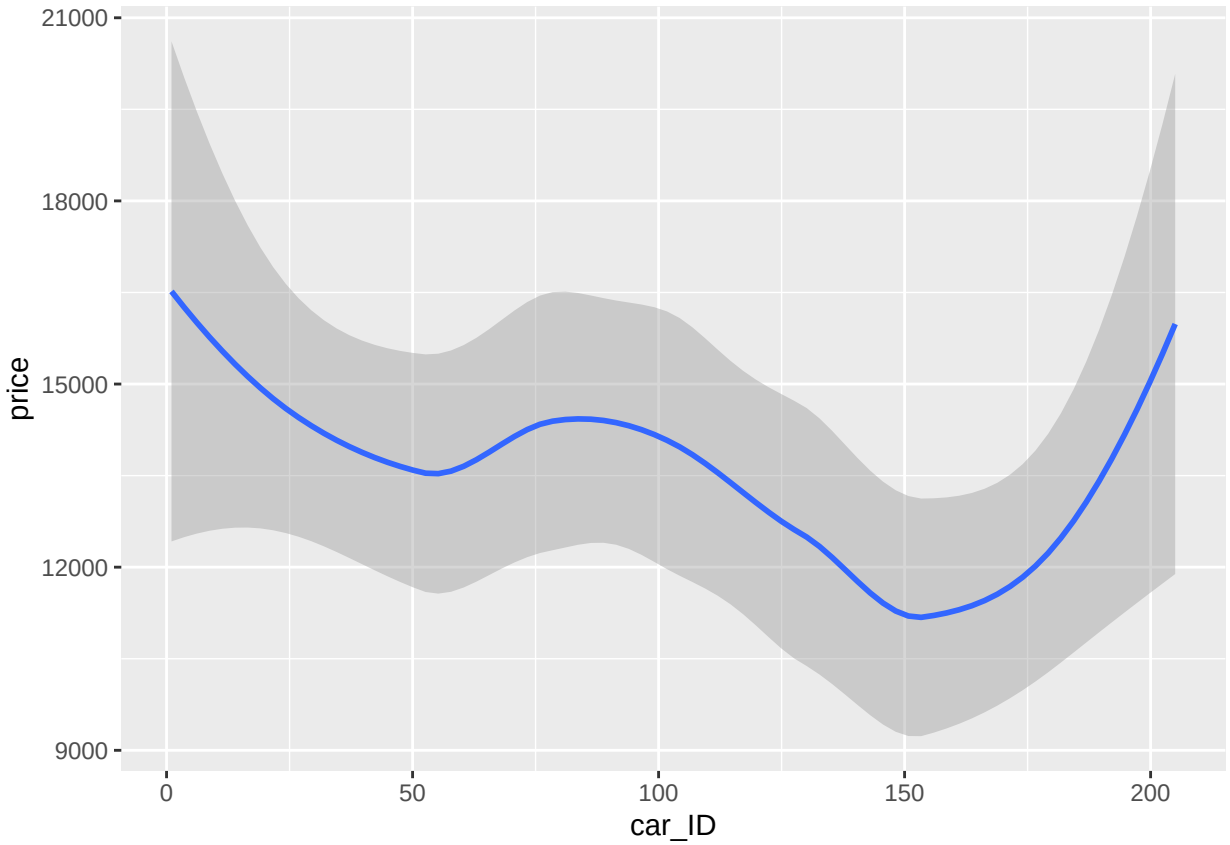
Plot fuel type between all cars

```
ggplot(car_data, aes(x = "", y = fueltype, fill = fueltype)) +
  geom_col() +coord_polar(theta = "y")
```



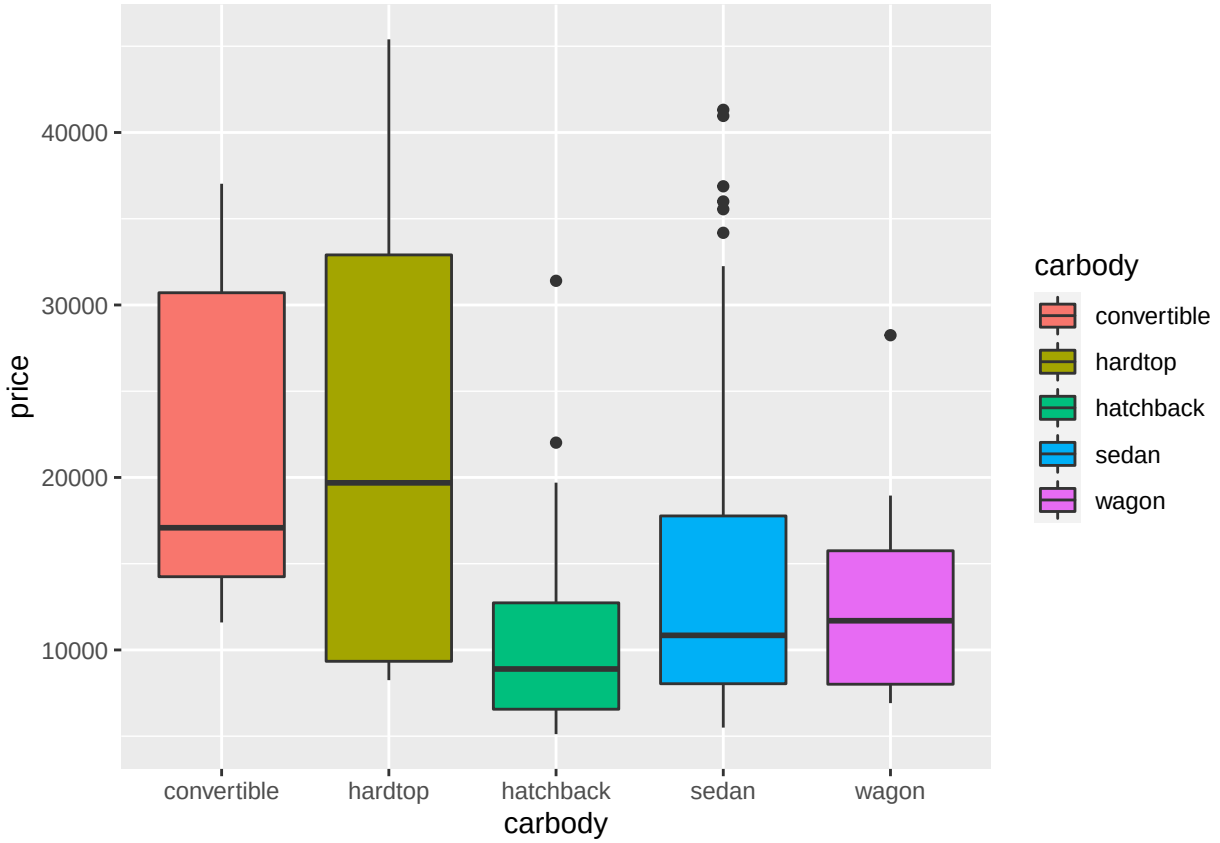
Plot the distribution of cars and price

```
ggplot(car_data2,aes(x=car_ID,y=price,fill=price))+geom_smooth()
```



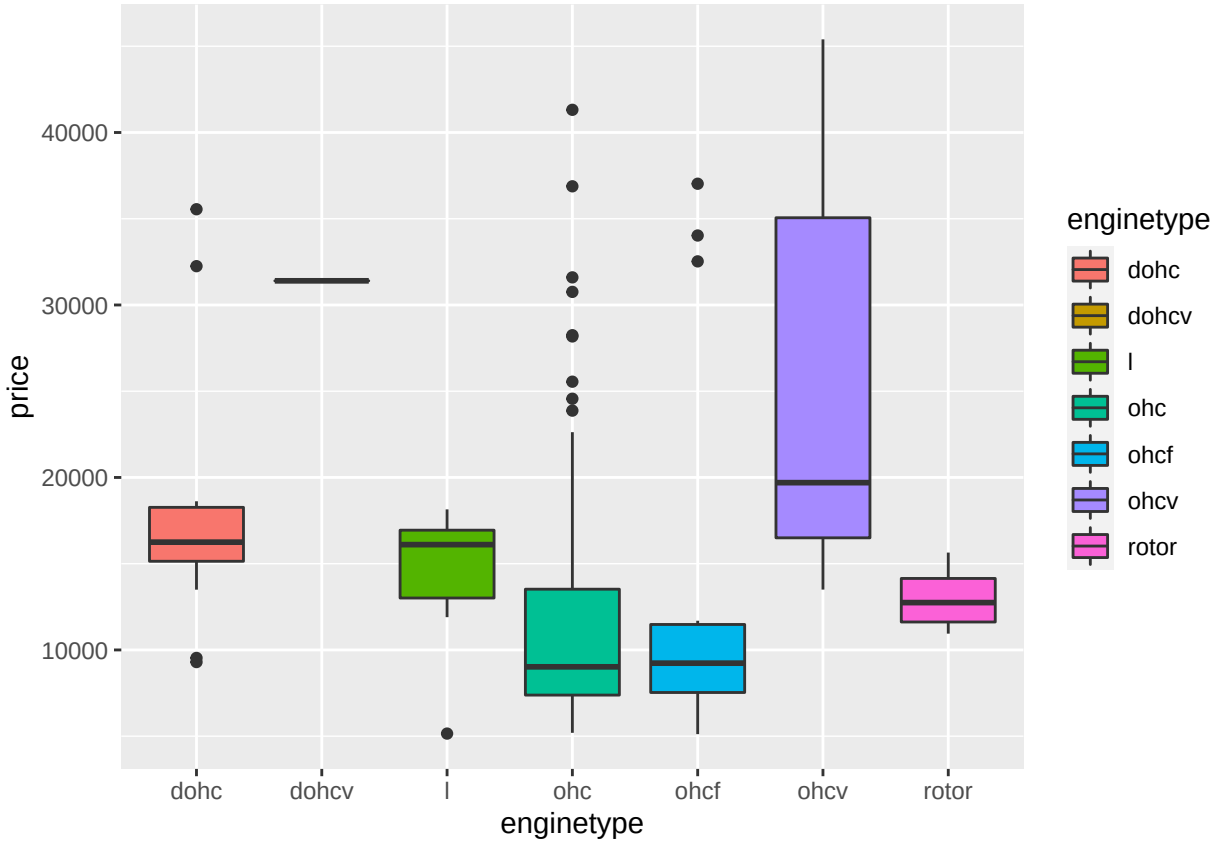
Plot the distribution of car body and price

```
ggplot(car_data,aes(x=carbody,y=price,fill=carbody))+geom_boxplot()
```



Plot the distribition of car body and price

```
ggplot(car_data,aes(x=enginetype,y=price,fill=enginetype))+geom_boxplot()
```



Linear multivariate Model of price

```
Car_Price_Regression<-lm(price~ symboling+fueltype+aspiration+doornumber+carbody+
                          drivewheel+enginelocation+enginelocation+wheelbase+carlength
                         +carwidth+carheight+curbweight+enginetype+cylindernumber+
                         enginesize+ fuelsystem+boreratio+stroke+compressionratio+
                         horsepower+compressionratio+peakrpm+citympg+highwaympg
                          ,data = car_data)

summary(Car_Price_Regression)
```

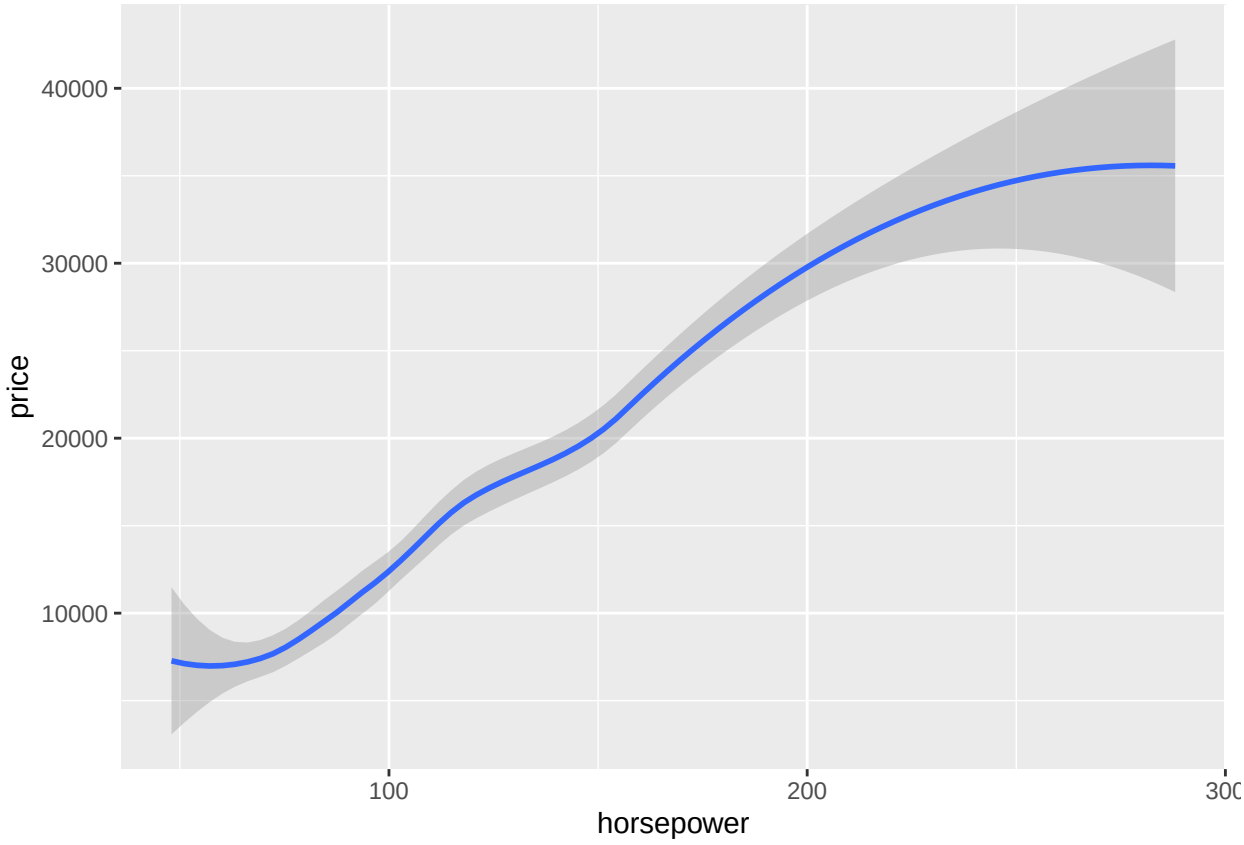
```
##
## Call:
## lm(formula = price ~ symboling + fueltype + aspiration + doornumber +
##     carbody + drivewheel + enginelocation + enginelocation +
##     wheelbase + carlength + carwidth + carheight + curbweight +
##     enginetype + cylindernumber + enginesize + fuelsystem + boreratio +
##     stroke + compressionratio + horsepower + compressionratio +
##     peakrpm + citympg + highwaympg, data = car_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5416.2 -1152.0   -35.8   830.8  9835.6
##
```

```
## Coefficients: (2 not defined because of singularities)
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -2.226e+04  1.652e+04  -1.347 0.179705
## symboling             7.388e+01  2.386e+02   0.310 0.757238
## fueltypegas          -1.178e+04  7.017e+03  -1.678 0.095232 .
## aspirationturbo       1.626e+03  8.856e+02   1.836 0.068172 .
## doornumbertwo         1.876e+02  5.854e+02   0.320 0.749028
## carbodyhardtop       -3.207e+03  1.376e+03  -2.331 0.020992 *
## carbodyhatchback     -3.281e+03  1.223e+03  -2.683 0.008055 **
## carbodysedan         -2.152e+03  1.332e+03  -1.615 0.108182
## carbodywagon         -3.266e+03  1.455e+03  -2.244 0.026191 *
## drivewheelfwd         7.405e+01  1.040e+03   0.071 0.943351
## drivewheelrwd         1.033e+03  1.205e+03   0.857 0.392688
## enginelocationrear    7.695e+03  2.536e+03   3.035 0.002802 **
## wheelbase             4.882e+01  9.675e+01   0.505 0.614563
## carlength            -6.130e+01  4.875e+01  -1.257 0.210410
## carwidth              6.936e+02  2.394e+02   2.897 0.004283 **
## carheight             8.943e+01  1.278e+02   0.700 0.485209
## curbweight            3.942e+00  1.715e+00   2.299 0.022781 *
## enginetypedohcv      -7.189e+03  4.674e+03  -1.538 0.125912
## enginetypel          -1.051e+03  1.608e+03  -0.654 0.514246
## enginetypeohc         3.126e+03  9.088e+02   3.439 0.000741 ***
## enginetypeohcf        1.234e+03  1.572e+03   0.785 0.433661
## enginetypeohcv       -5.605e+03  1.247e+03  -4.495 1.31e-05 ***
## enginetyperotor      -6.925e+01  4.505e+03  -0.015 0.987754
## cylindernumberfive   -9.280e+03  2.716e+03  -3.417 0.000800 ***
## cylindernumberfour   -9.879e+03  3.054e+03  -3.234 0.001476 **
## cylindernumbersix    -6.570e+03  2.192e+03  -2.997 0.003154 **
## cylindernumberthree  -4.629e+02  4.499e+03  -0.103 0.918173
## cylindernumbertwelve -1.024e+04  4.384e+03  -2.336 0.020707 *
## cylindernumbertwo            NA         NA      NA       NA
## enginesize            1.174e+02  2.600e+01   4.515 1.21e-05 ***
## fuelsystem2bbl       -3.907e+01  8.920e+02  -0.044 0.965118
## fuelsystem4bbl       -1.624e+03  2.775e+03  -0.585 0.559295
## fuelsystemidi                NA         NA      NA       NA
## fuelsystemmfi        -3.480e+03  2.590e+03  -1.344 0.180967
## fuelsystemmpfi       -2.444e+02  1.001e+03  -0.244 0.807415
## fuelsystemspdi       -3.027e+03  1.382e+03  -2.191 0.029883 *
## fuelsystemspfi       -6.187e+02  2.508e+03  -0.247 0.805484
## boreratio            -1.882e+03  1.598e+03  -1.178 0.240443
## stroke               -4.454e+03  9.009e+02  -4.944 1.89e-06 ***
## compressionratio     -8.003e+02  5.259e+02  -1.522 0.129981
## horsepower            9.791e+00  2.227e+01   0.440 0.660789
## peakrpm               2.202e+00  6.194e-01   3.555 0.000495 ***
## citympg              -1.477e+02  1.474e+02  -1.003 0.317569
## highwaympg            1.916e+02  1.347e+02   1.422 0.156916
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2197 on 163 degrees of freedom
## Multiple R-squared:  0.9395, Adjusted R-squared:  0.9243
## F-statistic: 61.79 on 41 and 163 DF,  p-value: < 2.2e-16
```

Plot the biggest factor in car price

Horse power was the biggest positive factor according to our model

```
ggplot(car_data,aes(x=horsepower,y=price))+geom_smooth()
```



Random Forest Model

```
#Delete variables for more predicted accuracy
car_data$car_ID<-NULL
car_data$symboling<-NULL
car_data$brand<-NULL


#test and training data cars data
sample_data<-sample(c(TRUE,FALSE),nrow(car_data),replace=TRUE,prob =c(0.7,0.3))
train_data <- car_data[sample_data,]
test_data <- car_data[!sample_data,]

#Random forest model
random_forest_model= randomForest(price~.,data = train_data)
print(random_forest_model)
```

```
##
## Call:
##  randomForest(formula = price ~ ., data = train_data)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 7
##
##           Mean of squared residuals: 4737657
##                     % Var explained: 93.06
```

```
#Predict using our test data with our trained model
predict_price= predict(random_forest_model,test_data,interval='prediction')

#add price predcition variable to test data data frame
test_data$price_predict=predict_price
```

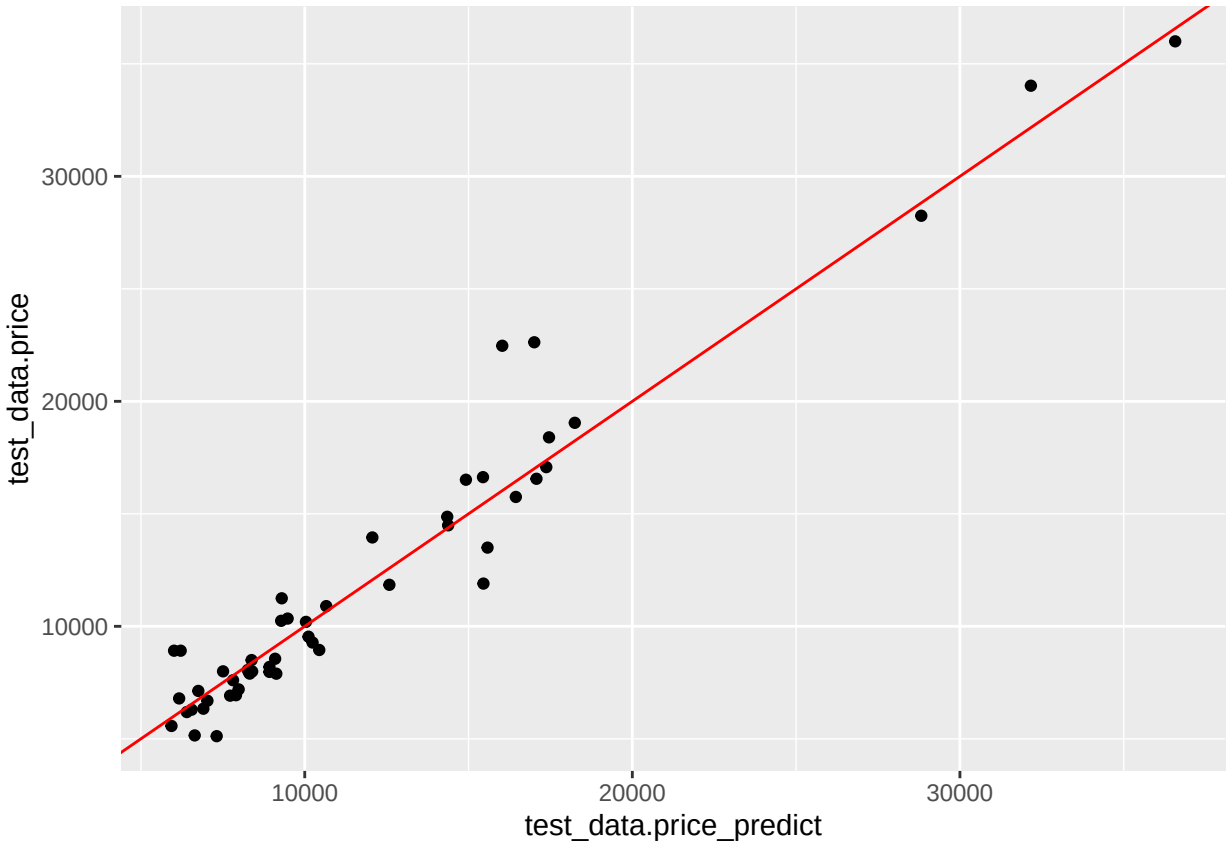
The accuracy of the Random Forest Model

```
#create data frame with price and predicted price
show_prediction<-data.frame(test_data$price,test_data$price_predict)


#compare values
all.equal(show_prediction$test_data.price,show_prediction$test_data.price_predict)
```

```
## [1] "Mean relative difference: 0.09453859"
```

plot the final results

```
ggplot(show_prediction,
       aes(x = test_data.price_predict,
           y = test_data.price
)) +
  geom_point() + geom_abline(color='red')
```


30000
20000
10000
test_data.price
10000
20000
30000
test_data.price_predict