Predicting Automobile Fuel Efficiency

Robin Phetsavong

Contents

Dataset

The dataset was obtained from the NHTSA. It consists of final paid transactions (purchases) during the CARS or "Cash for Clunkers" program that was in effect from 2009 to 2011. There are a total of 675,427 records and 42 total features originally. For the purpose of this study, the number of features was reduced to 6 after analysis and data mining. The variables are as follows:

1. Displacement – Factor with 33 levels
2. Cylinder – Factor with 5 levels
3. Transmission – Factor with 2 levels
4. Gears – Factor with 5 levels
5. Drivetrain – Factor with 3 levels
6. Mpg – Int

Goal

The goal of this study is to use exploratory statistics and machine learning algorithms to understand factors that contribute to a car's fuel efficiency as well as to accurately predict fuel efficiency given a set of parameters.

Techniques Used

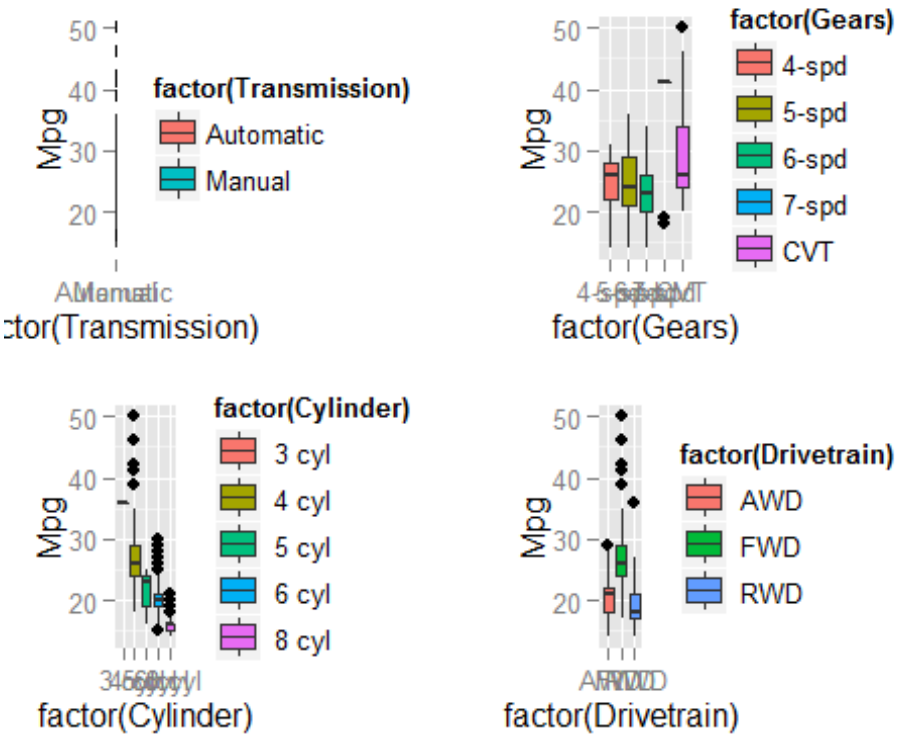
1. Correlation Matrix
2. Cross Validation
3. Confusion Matrix
4. Linear Regression Modeling

Feature Engineering and Component Analysis

Features were mined from the original field “New_vehicle_drive_train” which contained characteristics of the engine and transmission. This information was extracted to obtain all factors used in this study excluding MPG. This was done in excel which will not be shown in this report.

Data Exploration

Plots



The above plots were generated using ggplot2 in R. These boxplots reveal any outliers that may be present in the data (elaborate, I can’t see the charts)

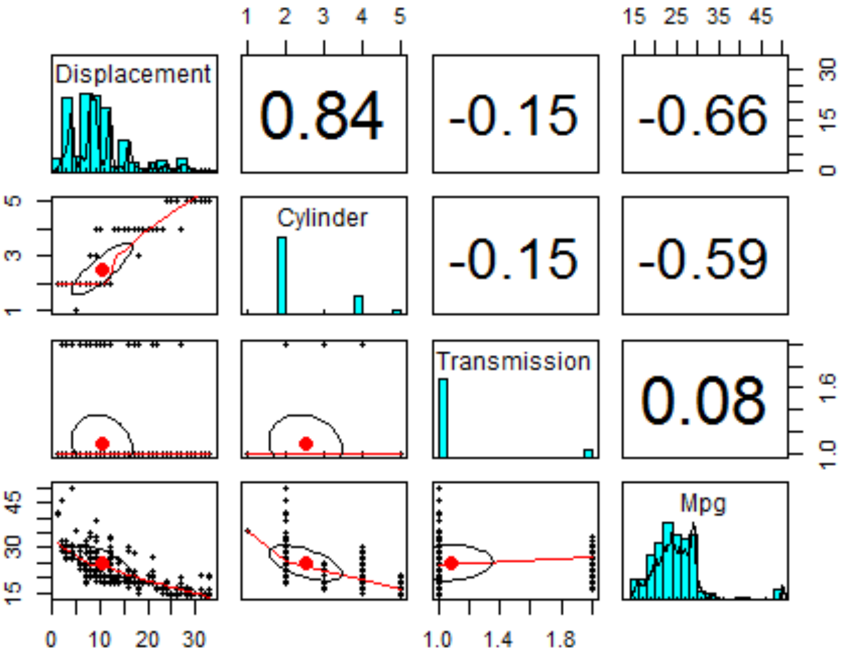
Correlation Matrix

The following is a correlation matrix of all 4 factors using the psych package in R

```
#Correlation Matrix
library(psych)

## Warning: package 'psych' was built under R version 3.2.3
```

```
##
## Attaching package: 'psych'

## The following object is masked from 'package:ggplot2':
##
##     %+%
```

```
cols <- c(1:3,6)
pairs.panels(auto_mpg[,cols])
```



A value closer to 1 or -1 means that two factors are highly correlated (as one changes so does the other) and a value closer to 0 means that they are weakly or not correlated at all.

We can see that Displacement and Cylinder are negatively correlated with MPG (given the negative correlation coefficient). This would make sense as the size of the engine goes up, fuel efficiency tends to drop and vice versa. Transmission has a very weak relationship with MPG and its effect on the model may prove to be marginal.

However we see that Displacement and Cylinder are very positively correlated with a coefficient of 0.84. Since we will be using linear regression to model our data for prediction, this presents a problem. Multicollinearity (or two or more predictor variables that are correlated) violates the assumptions of linear regression: No multicollinearity. To address this issue I will be removing the factor Cylinder as it has a marginally weaker coefficient of correlation with MPG than Displacement. However one can also choose to center one of the two factors at fault to potentially work around this without removing either variable.

Modeling and Predicting

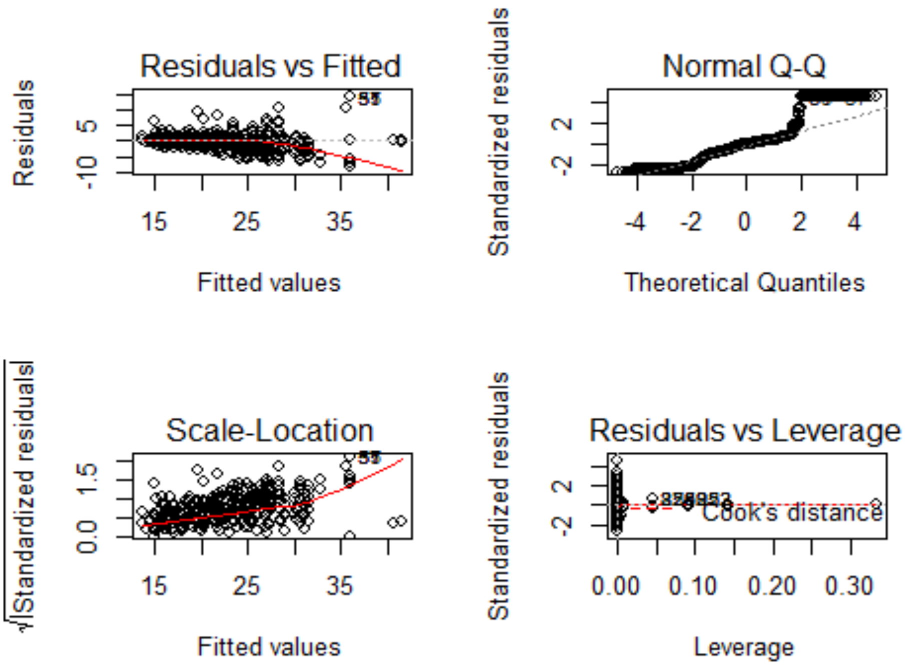
After exploring the data and addressing the issue of multicollinearity, the model was fitted using the lm function in R.

```
#Linear Model

lm_model <- lm(Mpg ~Displacement+Transmission+Gears+Drivetrain, data= auto_mp
g)
summary(lm_model)

##
## Call:
## lm(formula = Mpg ~ Displacement + Transmission + Gears + Drivetrain,
##     data = auto_mpg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.0253 -1.8039  0.1926  1.1926 13.9747
##
## Coefficients:
##                     Estimate Std. Error  t value Pr(>|t|)
## (Intercept)         34.47751    0.08977  384.058  < 2e-16 ***
## Displacement1.5L    -5.85144    0.09314  -62.823  < 2e-16 ***
## Displacement1.6L    -7.90365    0.09139  -86.479  < 2e-16 ***
## Displacement1.8L    -5.47480    0.08837  -61.951  < 2e-16 ***
## Displacement1L       0.67046    0.15687    4.274 1.92e-05 ***
## Displacement2.2L    -9.70581    0.09223 -105.230  < 2e-16 ***
## Displacement2.3L   -13.91705    0.09653 -144.167  < 2e-16 ***
## Displacement2.4L   -13.07931    0.08851 -147.780  < 2e-16 ***
## Displacement2.5L   -13.07726    0.08851 -147.753  < 2e-16 ***
## Displacement2.7L   -15.01865    0.09403 -159.726  < 2e-16 ***
## Displacement2.9L   -15.33099    0.10862 -141.138  < 2e-16 ***
## Displacement2L     -10.90452    0.08838 -123.387  < 2e-16 ***
## Displacement3.2L   -16.54123    0.25166  -65.728  < 2e-16 ***
## Displacement3.3L   -16.02882    0.09853 -162.687  < 2e-16 ***
## Displacement3.4L   -15.77182    0.10980 -143.643  < 2e-16 ***
## Displacement3.5L   -16.22305    0.08991 -180.445  < 2e-16 ***
## Displacement3.6L   -17.96713    0.09884 -181.788  < 2e-16 ***
## Displacement3.7L   -17.50184    0.10507 -166.576  < 2e-16 ***
## Displacement3.8L   -19.47338    0.11373 -171.232  < 2e-16 ***
## Displacement3.9L   -17.37337    0.67228  -25.842  < 2e-16 ***
## Displacement3L     -16.32442    0.09574 -170.513  < 2e-16 ***
## Displacement4.2L   -19.42620    0.94679  -20.518  < 2e-16 ***
## Displacement4.3L   -18.36071    0.10538 -174.232  < 2e-16 ***
## Displacement4.6L   -19.56186    0.09977 -196.066  < 2e-16 ***
## Displacement4.7L   -19.94200    0.13604 -146.589  < 2e-16 ***
## Displacement4.8L   -19.37138    0.11728 -165.177  < 2e-16 ***
## Displacement4L     -17.40223    0.09812 -177.356  < 2e-16 ***
## Displacement5.3L   -19.17479    0.09876 -194.157  < 2e-16 ***
## Displacement5.4L   -20.79999    0.11397 -182.511  < 2e-16 ***
## Displacement5.6L   -20.32954    1.80725  -11.249  < 2e-16 ***
## Displacement5.7L   -19.63050    0.11651 -168.492  < 2e-16 ***
## Displacement6.2L   -21.74216    1.18510  -18.346  < 2e-16 ***
## Displacement6L     -20.03564    0.27198  -73.667  < 2e-16 ***
## TransmissionManual   0.38767    0.02234   17.355  < 2e-16 ***
## Gears5-spd           0.04508    0.01622    2.779  0.00545 **
## Gears6-spd           1.37497    0.01770   77.676  < 2e-16 ***
## Gears7-spd           4.15013    0.14043   29.552  < 2e-16 ***
## GearsCVT             4.99032    0.01731  288.259  < 2e-16 ***
## Drivetrain FWD       2.03223    0.01649  123.218  < 2e-16 ***
## Drivetrain RWD       0.80695    0.02921   27.621  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.126 on 373799 degrees of freedom
## Multiple R-squared:  0.7122, Adjusted R-squared:  0.7121
## F-statistic: 2.372e+04 on 39 and 373799 DF,  p-value: < 2.2e-16
```

R^2 is the statistic that determines how “well” the model fits our data. That is to say that the combination of all the factors we put into the model can accurately account for 71% of the variation in the data set. I say “well” in quotations as the purpose of this study is to predict, not just fit a model to the dataset. In this case R^2 came out to be 0.71 which is a decent fit.

We further validate the model by viewing plots of the residuals.

```
plot(lm_model)
```



The residual plots show that the model is a good fit and that the residuals (or standard errors) are approximately normally distributed (this can further be checked with a Goodness of fit test).

As mentioned early however, our goal is to predict the MPG of cars, so we must cross validate our model by testing it on an independent dataset. In this case, before applying the model, I split the original 675,427 records in half and fitted the linear regression model on the first half (the training set) and then used the model to predict the second half (the test set).

```r
#Model can predict with an accuracy of 86% according to the correlation coeff
icient Using Cross Validation


#Predict Test Model
auto_mpg_test <- read.csv("testcar.csv")

predicttest <-predict.lm(lm_model,auto_mpg_test)

cor(auto_mpg_test$Mpg,predicttest)

## [1] 0.8624023
```

The result after computing the correlation coefficient between the model and the test set was an accuracy of 86%, a very respectable accuracy for our predictor model.

Conclusion

The CARS dataset was mined to gather information on the drivetrain of new cars purchased during the program from 2009 to 2011. This was done to analyze what factors contributed to the mpg ratings of these cars. Using basic statistics and exploratory analysis, we were able to find that the displacement and number of cylinders in an engine play a significant role in determining how fuel efficient a car is. We then split the data into training and test data sets and built a linear regression model on the training set with a R^2 of 0.71 after adjusting the model due to multicollinearity. The model was then used to predict the test set with an accuracy of 86%.

Appendix

auto_mpg.R

Robin Phetsavong

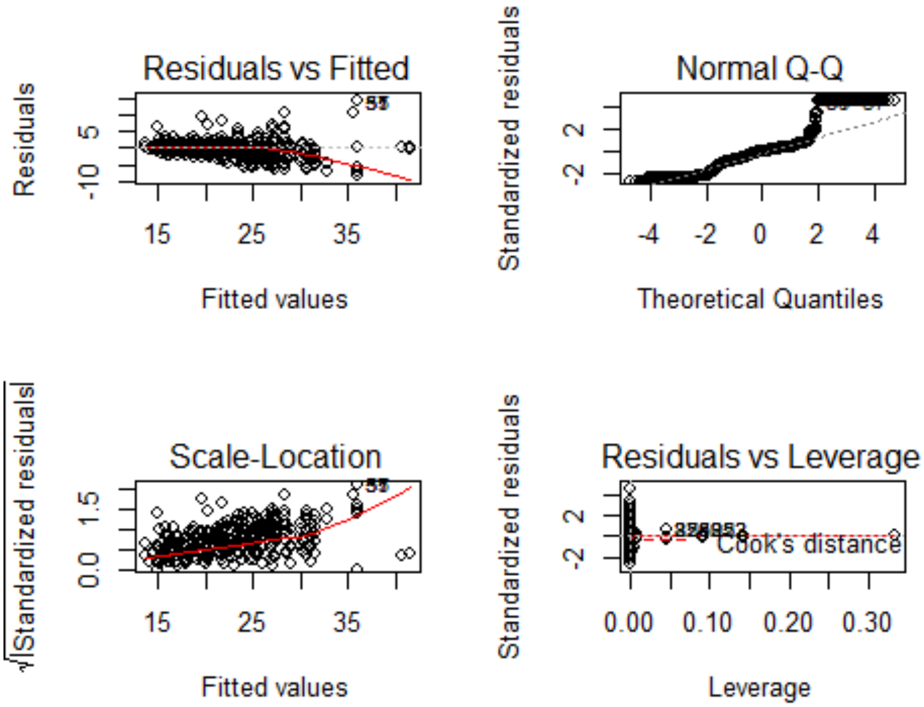
Wed Jan 20 19:54:50 2016

```
auto_mpg <- read.csv("traincar.csv")
str(auto_mpg)

## 'data.frame':    373839 obs. of  6 variables:
##  $ Displacement: Factor w/ 33 levels "1.3L","1.5L",..: 12 2 8 4 12 9 9 4 7
8 ...
##  $ Cylinder    : Factor w/ 5 levels " 3 cyl"," 4 cyl",..: 2 2 2 2 2 2 2 2
2 2 ...
##  $ Transmission: Factor w/ 2 levels "Automatic","Manual": 1 1 2 1 1 1 1 1
2 1 ...
##  $ Gears       : Factor w/ 5 levels "4-spd","5-spd",..: 1 2 2 2 1 1 3 2 2
2 ...
##  $ Drivetrain  : Factor w/ 3 levels " AWD"," FWD",..: 2 2 1 2 2 1 1 2 3 2
...
##  $ Mpg         : int  27 30 25 29 28 24 21 29 23 25 ...

par(mfrow=c(2,2))
library(ggplot2)


####### Function courtesey of Knitr and Jekyll @ Cookbook-r
multiplot <- function(..., plotlist=NULL, file, cols=1, layout=NULL) {
  library(grid)

  # Make a list from the ... arguments and plotlist
  plots <- c(list(...), plotlist)

  numPlots = length(plots)

  # If layout is NULL, then use 'cols' to determine layout
  if (is.null(layout)) {
    # Make the panel
    # ncol: Number of columns of plots
    # nrow: Number of rows needed, calculated from # of cols
    layout <- matrix(seq(1, cols * ceiling(numPlots/cols)),
                     ncol = cols, nrow = ceiling(numPlots/cols))
  }

  if (numPlots==1) {
    print(plots[[1]])

  } else {
    # Set up the page
    grid.newpage()
    pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))

    # Make each plot, in the correct location
    for (i in 1:numPlots) {
      # Get the i,j matrix positions of the regions that contain this subplot
      matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))

      print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,
                                      layout.pos.col = matchidx$col))
    }
  }
}
######


p1<-ggplot(auto_mpg, aes(factor(Transmission), Mpg)) + geom_boxplot(aes(fill=
factor(Transmission)))
p2<-ggplot(auto_mpg, aes(factor(Cylinder), Mpg)) + geom_boxplot(aes(fill=fact
or(Cylinder)))
p3<-ggplot(auto_mpg, aes(factor(Gears), Mpg)) + geom_boxplot(aes(fill=factor(
Gears)))
p4<-ggplot(auto_mpg, aes(factor(Drivetrain), Mpg)) + geom_boxplot(aes(fill=fa
ctor(Drivetrain)))

multiplot(p1,p2,p3,p4, cols=2)
```

```
#Correlation Matrix
library(psych)

## Warning: package 'psych' was built under R version 3.2.3

##
## Attaching package: 'psych'

## The following object is masked from 'package:ggplot2':
##
##     %+%

cols <- c(1:3,6)
pairs.panels(auto_mpg[,cols])
```

```
#Because of Collinearity, we have to center either the Displacement or Cylind
er factor to adhere to the assumptions of Linear Regression


#We have to deal with the categorical variables first

auto_mpg$Cylinder <- as.factor(auto_mpg$Cylinder)
auto_mpg$Transmission <- as.factor(auto_mpg$Transmission)

#Linear Model

lm_model <- lm(Mpg ~Displacement+Transmission+Gears+Drivetrain, data= auto_mp
g)
summary(lm_model)

##
## Call:
## lm(formula = Mpg ~ Displacement + Transmission + Gears + Drivetrain,
##     data = auto_mpg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.0253 -1.8039  0.1926  1.1926 13.9747
##
## Coefficients:
##                        Estimate Std. Error  t value Pr(>|t|)
## (Intercept)           34.47751    0.08977  384.058  < 2e-16 ***
## Displacement1.5L      -5.85144    0.09314  -62.823  < 2e-16 ***
## Displacement1.6L      -7.90365    0.09139  -86.479  < 2e-16 ***
## Displacement1.8L      -5.47480    0.08837  -61.951  < 2e-16 ***
## Displacement1L         0.67046    0.15687    4.274 1.92e-05 ***
## Displacement2.2L      -9.70581    0.09223 -105.230  < 2e-16 ***
## Displacement2.3L     -13.91705    0.09653 -144.167  < 2e-16 ***
## Displacement2.4L     -13.07931    0.08851 -147.780  < 2e-16 ***
## Displacement2.5L     -13.07726    0.08851 -147.753  < 2e-16 ***
## Displacement2.7L     -15.01865    0.09403 -159.726  < 2e-16 ***
## Displacement2.9L     -15.33099    0.10862 -141.138  < 2e-16 ***
## Displacement2L       -10.90452    0.08838 -123.387  < 2e-16 ***
## Displacement3.2L     -16.54123    0.25166  -65.728  < 2e-16 ***
## Displacement3.3L     -16.02882    0.09853 -162.687  < 2e-16 ***
## Displacement3.4L     -15.77182    0.10980 -143.643  < 2e-16 ***
## Displacement3.5L     -16.22305    0.08991 -180.445  < 2e-16 ***
## Displacement3.6L     -17.96713    0.09884 -181.788  < 2e-16 ***
## Displacement3.7L     -17.50184    0.10507 -166.576  < 2e-16 ***
## Displacement3.8L     -19.47338    0.11373 -171.232  < 2e-16 ***
## Displacement3.9L     -17.37337    0.67228  -25.842  < 2e-16 ***
## Displacement3L       -16.32442    0.09574 -170.513  < 2e-16 ***
## Displacement4.2L   -19.42620    0.94679  -20.518  < 2e-16 ***
## Displacement4.3L   -18.36071    0.10538 -174.232  < 2e-16 ***
## Displacement4.6L   -19.56186    0.09977 -196.066  < 2e-16 ***
## Displacement4.7L   -19.94200    0.13604 -146.589  < 2e-16 ***
## Displacement4.8L   -19.37138    0.11728 -165.177  < 2e-16 ***
## Displacement4L     -17.40223    0.09812 -177.356  < 2e-16 ***
## Displacement5.3L   -19.17479    0.09876 -194.157  < 2e-16 ***
## Displacement5.4L   -20.79999    0.11397 -182.511  < 2e-16 ***
## Displacement5.6L   -20.32954    1.80725  -11.249  < 2e-16 ***
## Displacement5.7L   -19.63050    0.11651 -168.492  < 2e-16 ***
## Displacement6.2L   -21.74216    1.18510  -18.346  < 2e-16 ***
## Displacement6L     -20.03564    0.27198  -73.667  < 2e-16 ***
## TransmissionManual   0.38767    0.02234   17.355  < 2e-16 ***
## Gears5-spd           0.04508    0.01622    2.779  0.00545 **
## Gears6-spd           1.37497    0.01770   77.676  < 2e-16 ***
## Gears7-spd           4.15013    0.14043   29.552  < 2e-16 ***
## GearsCVT             4.99032    0.01731  288.259  < 2e-16 ***
## Drivetrain FWD       2.03223    0.01649  123.218  < 2e-16 ***
## Drivetrain RWD       0.80695    0.02921   27.621  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.126 on 373799 degrees of freedom
## Multiple R-squared:  0.7122, Adjusted R-squared:  0.7121
## F-statistic: 2.372e+04 on 39 and 373799 DF,  p-value: < 2.2e-16
```

```
plot(lm_model)
```



```
#Predict
predicted <- predict.lm(lm_model,auto_mpg)

#Model can predict with an accuracy of 86% according to the correlation coeff
icient Using Cross Validation


#Predict Test Model
auto_mpg_test <- read.csv("testcar.csv")

predicttest <-predict.lm(lm_model,auto_mpg_test)

cor(auto_mpg_test$Mpg,predicttest)

## [1] 0.8624023
```