Modern Data Mining - HW 2

Cheshta Dhingra

## Overview / Instructions

This is homework #2 of STAT 471/571/701. It will be **due on 19 Febrary, 2017 by 11:59 PM** on Canvas. You can directly edit this file to add your answers. Submit the Rmd file, a PDF or word or HTML version with only 1 submission per HW team.

## Problem 0

Review the code and concepts covered during lecture: multiple regression, model selection and penalized regression through elastic net.

## Problem 1:

Auto data from ISLR. The original data contains 408 observations about cars. It has some similarity as the data CARS that we use in our lectures. To get the data, first install the package ISLR. The data Auto should be loaded automatically. We use this case to go through methods learnt so far.

You can access the necessary data with the following code:

# check if you have ISLR package, if not, install it  
if(!requireNamespace('ISLR')) install.packages('ISLR')   
auto\_data <- ISLR::Auto

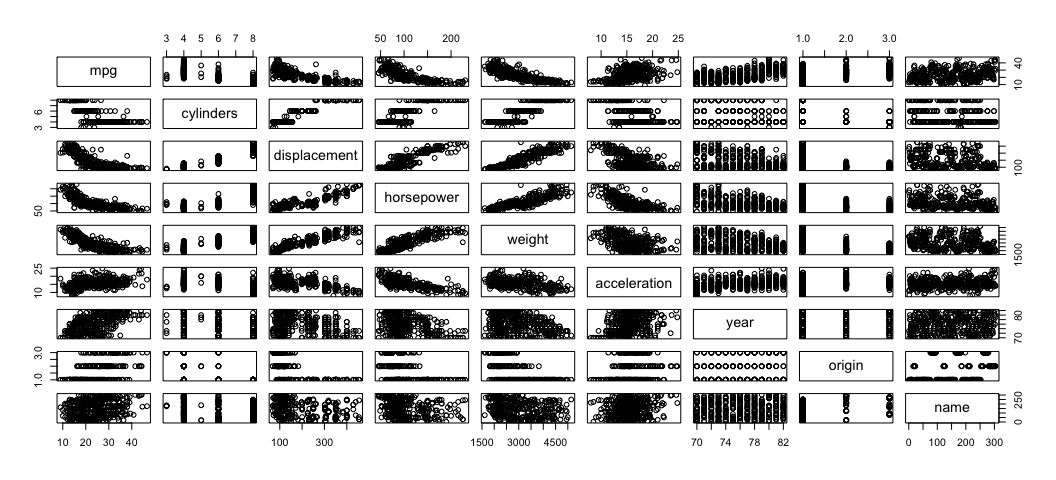
Get familiar with this dataset first. You can use ?ISLR::Auto to view a description of the dataset.

1. Explore the data, with particular focus on pairwise plots and summary statistics. Briefly summarize your findings and any peculiarities in the data.

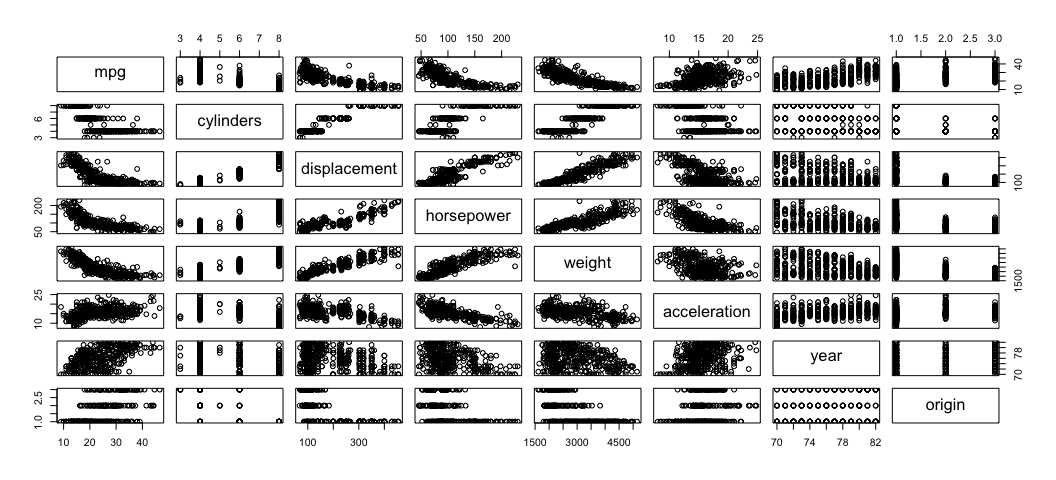
# Summarize Data   
summary(auto\_data)

## mpg cylinders displacement horsepower   
## Min. : 9.00 Min. :3.000 Min. : 68.0 Min. : 46.0   
## 1st Qu.:17.00 1st Qu.:4.000 1st Qu.:105.0 1st Qu.: 75.0   
## Median :22.75 Median :4.000 Median :151.0 Median : 93.5   
## Mean :23.45 Mean :5.472 Mean :194.4 Mean :104.5   
## 3rd Qu.:29.00 3rd Qu.:8.000 3rd Qu.:275.8 3rd Qu.:126.0   
## Max. :46.60 Max. :8.000 Max. :455.0 Max. :230.0   
##   
## weight acceleration year origin   
## Min. :1613 Min. : 8.00 Min. :70.00 Min. :1.000   
## 1st Qu.:2225 1st Qu.:13.78 1st Qu.:73.00 1st Qu.:1.000   
## Median :2804 Median :15.50 Median :76.00 Median :1.000   
## Mean :2978 Mean :15.54 Mean :75.98 Mean :1.577   
## 3rd Qu.:3615 3rd Qu.:17.02 3rd Qu.:79.00 3rd Qu.:2.000   
## Max. :5140 Max. :24.80 Max. :82.00 Max. :3.000   
##   
## name   
## amc matador : 5   
## ford pinto : 5   
## toyota corolla : 5   
## amc gremlin : 4   
## amc hornet : 4   
## chevrolet chevette: 4   
## (Other) :365

# Pairwise plots   
pairs(auto\_data)



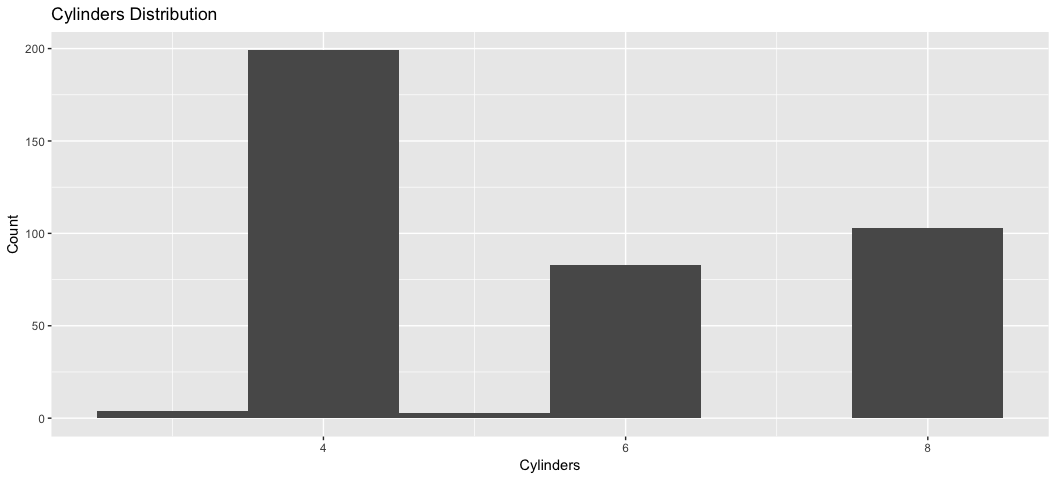
name.num <- sapply(auto\_data, is.numeric) # pulling out all the num. var's.  
pairs(auto\_data[name.num])



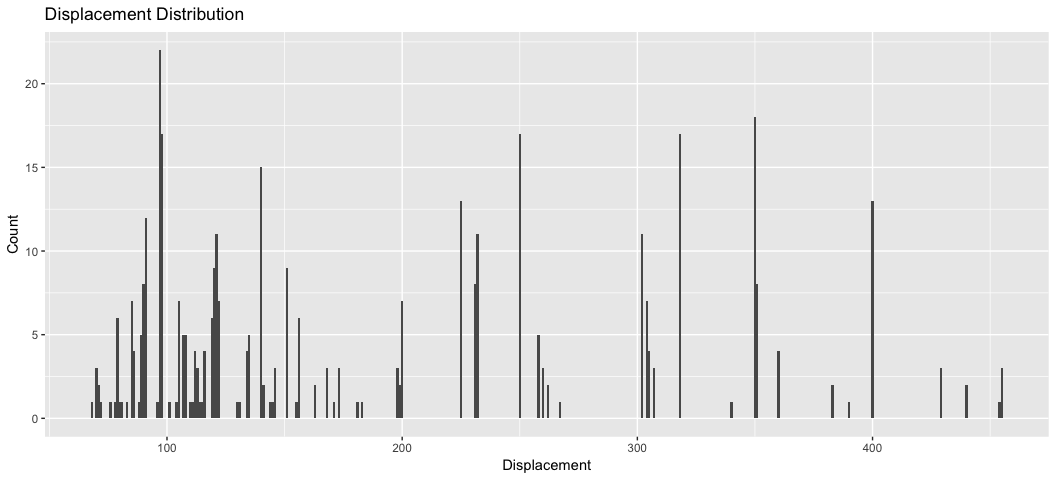
cor(auto\_data[name.num])

## mpg cylinders displacement horsepower weight  
## mpg 1.0000000 -0.7776175 -0.8051269 -0.7784268 -0.8322442  
## cylinders -0.7776175 1.0000000 0.9508233 0.8429834 0.8975273  
## displacement -0.8051269 0.9508233 1.0000000 0.8972570 0.9329944  
## horsepower -0.7784268 0.8429834 0.8972570 1.0000000 0.8645377  
## weight -0.8322442 0.8975273 0.9329944 0.8645377 1.0000000  
## acceleration 0.4233285 -0.5046834 -0.5438005 -0.6891955 -0.4168392  
## year 0.5805410 -0.3456474 -0.3698552 -0.4163615 -0.3091199  
## origin 0.5652088 -0.5689316 -0.6145351 -0.4551715 -0.5850054  
## acceleration year origin  
## mpg 0.4233285 0.5805410 0.5652088  
## cylinders -0.5046834 -0.3456474 -0.5689316  
## displacement -0.5438005 -0.3698552 -0.6145351  
## horsepower -0.6891955 -0.4163615 -0.4551715  
## weight -0.4168392 -0.3091199 -0.5850054  
## acceleration 1.0000000 0.2903161 0.2127458  
## year 0.2903161 1.0000000 0.1815277  
## origin 0.2127458 0.1815277 1.0000000

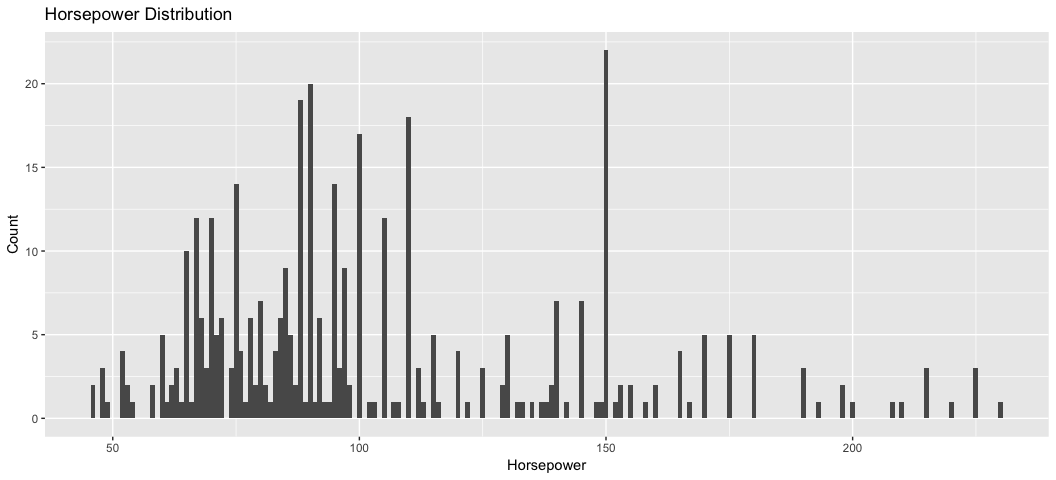
# Plot Cylinders Distribution  
ggplot(auto\_data, aes(x = auto\_data$cylinders)) + geom\_histogram(binwidth = 1) +labs(title = "Cylinders Distribution", x = "Cylinders", y = "Count")



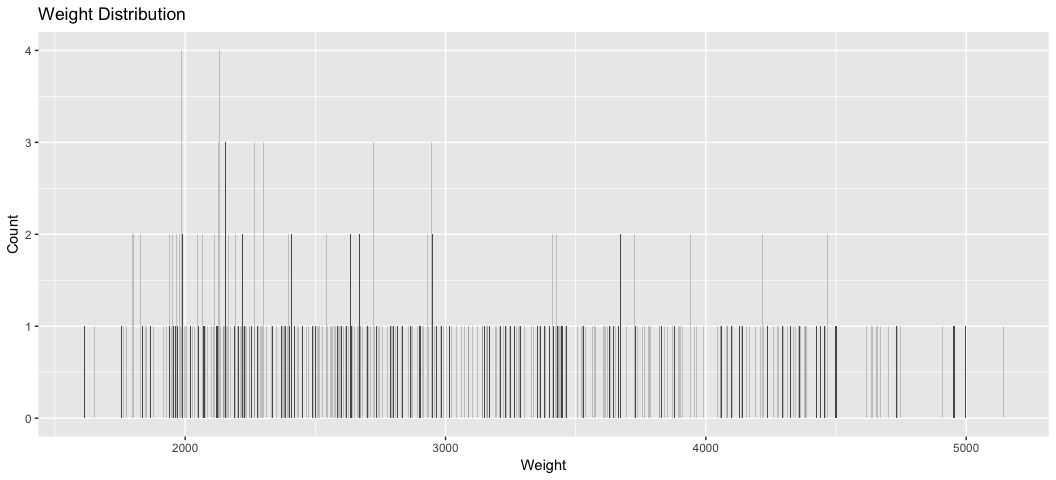
# Plot Displacement Distribution   
ggplot(auto\_data, aes(x = auto\_data$displacement)) + geom\_histogram(binwidth = 1) +labs(title = "Displacement Distribution", x = "Displacement", y = "Count")



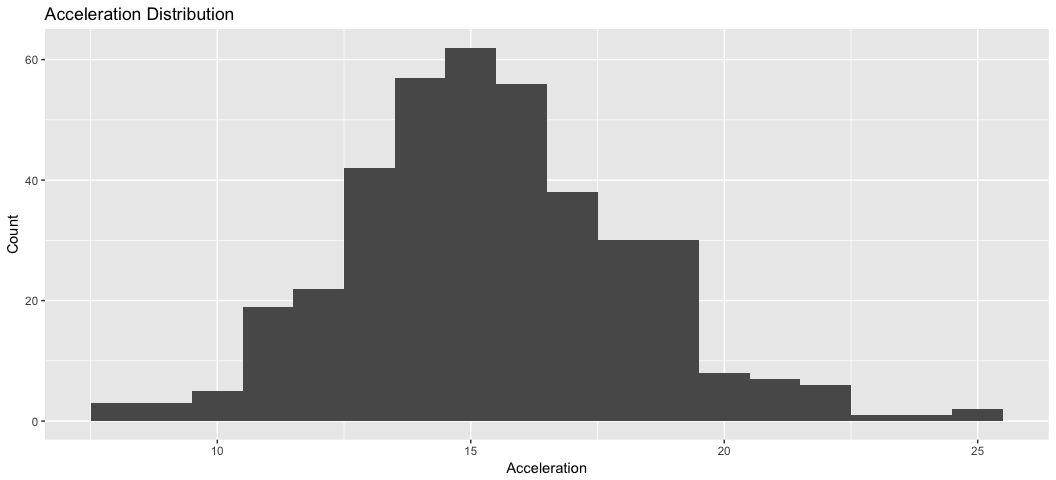
# Plot Horsepower Distribution  
ggplot(auto\_data, aes(x = auto\_data$horsepower)) + geom\_histogram(binwidth = 1) +labs(title = "Horsepower Distribution", x = "Horsepower", y = "Count")



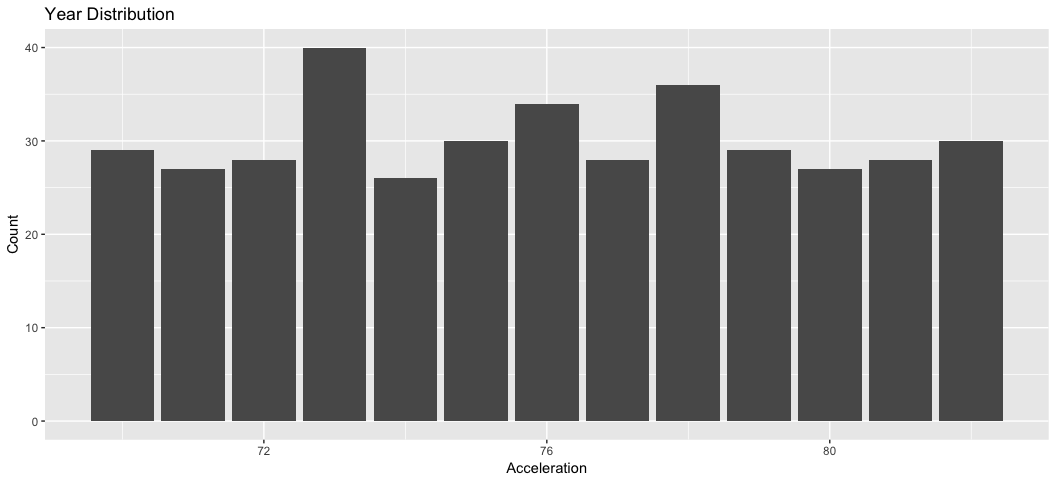
# Plot Weight Distribution  
ggplot(auto\_data, aes(x = auto\_data$weight)) + geom\_histogram(binwidth = 1) +labs(title = "Weight Distribution", x = "Weight", y = "Count")



# Plot Acceleration Distribution  
ggplot(auto\_data, aes(x = auto\_data$acceleration)) + geom\_histogram(binwidth = 1) +labs(title = "Acceleration Distribution", x = "Acceleration", y = "Count")



# Plot Year Distribution   
ggplot(auto\_data, aes(x = auto\_data$year)) + geom\_bar() +labs(title = "Year Distribution", x = "Acceleration", y = "Count")

 To summarize the data, we can see that the mean mpg of this sample = 23.45. The mean number of cylinders = 5.47, the mean displacement = 194.4, mean horsepower = 104.5, mean weight = 2978, mean acceleration = 15.54. Finally the mean year these cars were made = 76. We can see strong correlations by looking at the pairwise plots above. For example, mpg seems to be strongly negatively correlated with displacement, horsepower and weight, all of which seem to be positively correlated with each other. These three variables also appear to be negatively correlated with acceleration.

1. What effect does time have on MPG?
   1. Start with a simple regression of mpg vs. year and report R's summary output. Is year a significant variable at the .05 level? State what effect year has on mpg, if any, according to this model.

reg1 <- lm(auto\_data$mpg ~ auto\_data$year)  
summary(reg1)

##   
## Call:  
## lm(formula = auto\_data$mpg ~ auto\_data$year)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -12.0212 -5.4411 -0.4412 4.9739 18.2088   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -70.01167 6.64516 -10.54 <2e-16 \*\*\*  
## auto\_data$year 1.23004 0.08736 14.08 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 6.363 on 390 degrees of freedom  
## Multiple R-squared: 0.337, Adjusted R-squared: 0.3353   
## F-statistic: 198.3 on 1 and 390 DF, p-value: < 2.2e-16

Year is a significant variable at the 0.05 level (and even lower). We expect that for every additional year that passes, the MPG increases by 1.23 on average.

i. Add horsepower on top of the variable year. Is year still a significant variable at the .05 level? Give a precise interpretation of the year effect found here.

reg2 <- lm(auto\_data$mpg ~ auto\_data$year + auto\_data$horsepower)  
summary(reg2)

##   
## Call:  
## lm(formula = auto\_data$mpg ~ auto\_data$year + auto\_data$horsepower)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -12.0768 -3.0783 -0.4308 2.5884 15.3153   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -12.739166 5.349027 -2.382 0.0177 \*   
## auto\_data$year 0.657268 0.066262 9.919 <2e-16 \*\*\*  
## auto\_data$horsepower -0.131654 0.006341 -20.761 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 4.388 on 389 degrees of freedom  
## Multiple R-squared: 0.6855, Adjusted R-squared: 0.6839   
## F-statistic: 423.9 on 2 and 389 DF, p-value: < 2.2e-16

Yes, year is still significant at the 0.05 level. If horsepower is held constant, the effect of an additional year on MPG is an increase of 0.657 on average. If year is held constant, the effect of an additional unit of horsepower on MPG is a decrease of 0.132 on average.

i. The two 95% CI's for the coefficient of year differ among i) and ii). How would you explain the difference to a non-statistician?

The reason these two CI's are different is because the variables are defined differently in the two models. With the single regressor model, we don't take any other features of the car into account and only look at the effect of year on MPG on average. However, when we add Horsepower to the model, the coefficient for year stands for the impact of year on MPG *keeping horsepower constant*. i. Do a model with interaction by fitting lm(mpg ~ year \* horsepower). Is the interaction effect significant at .05 level? Explain the year effect (if any).

reg3 <- lm(auto\_data$mpg ~ auto\_data$year + auto\_data$horsepower + auto\_data$year \* auto\_data$horsepower)  
summary(reg3)

##   
## Call:  
## lm(formula = auto\_data$mpg ~ auto\_data$year + auto\_data$horsepower +   
## auto\_data$year \* auto\_data$horsepower)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -12.3492 -2.4509 -0.4557 2.4056 14.4437   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) -1.266e+02 1.212e+01 -10.449 <2e-16  
## auto\_data$year 2.192e+00 1.613e-01 13.585 <2e-16  
## auto\_data$horsepower 1.046e+00 1.154e-01 9.063 <2e-16  
## auto\_data$year:auto\_data$horsepower -1.596e-02 1.562e-03 -10.217 <2e-16  
##   
## (Intercept) \*\*\*  
## auto\_data$year \*\*\*  
## auto\_data$horsepower \*\*\*  
## auto\_data$year:auto\_data$horsepower \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.901 on 388 degrees of freedom  
## Multiple R-squared: 0.7522, Adjusted R-squared: 0.7503   
## F-statistic: 392.5 on 3 and 388 DF, p-value: < 2.2e-16

Yes, the interaction term is significant at the 0.05 level. This means that the effect of year on MPG is different for different values of horsepower, and vice versa. The net effect of year on the MPG is not limited to B1 but also includes B3 and the horsepower. Put another way, the slopes of the regression lines for the effect of year on MPG are different for different values of horsepower, and this difference is indicated by the interaction term coefficient, B3. Since this term is negative, the larger horsepower gets, the lower the effect of year on MPG gets. B1 by itself is now interpreted as the unique effect of year on MPG when it's horsepower = 0. Theoretically this value = 2.192 (though it's hard to interpret a situation where the car's horsepower = 0.)

1. Remember that the same variable can play different roles! Take a quick look at the variable cylinders, try to use this variable in the following analyses wisely. We all agree that larger number of cylinder will lower mpg. However, we can interpret cylinders as either a continuous (numeric) variable or a categorical variable.
   1. Fit a model, that treats cylinders as a continuous/numeric variable: lm(mpg ~ horsepower + cylinders, ISLR::Auto). Is cylinders significant at the 0.01 level? What effect does cylinders play in this model?

reg4 <- lm(auto\_data$mpg ~ auto\_data$cylinders)  
summary(reg4)

##   
## Call:  
## lm(formula = auto\_data$mpg ~ auto\_data$cylinders)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -14.2413 -3.1832 -0.6332 2.5491 17.9168   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 42.9155 0.8349 51.40 <2e-16 \*\*\*  
## auto\_data$cylinders -3.5581 0.1457 -24.43 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 4.914 on 390 degrees of freedom  
## Multiple R-squared: 0.6047, Adjusted R-squared: 0.6037   
## F-statistic: 596.6 on 1 and 390 DF, p-value: < 2.2e-16

Each additional cylinder corresponds to a decrease in MPG by 3.55 on average. This is significant at the 0.01 level.

ii. Fit a model that treats `cylinders` as a categorical/factor variable: `lm(mpg ~ horsepower + as.factor(cylinders), ISLR::Auto)`. Is `cylinders` significant at the .01 level? What is the effect of `cylinders` in this model? Use `anova(fit1, fit2)` and `Anova(fit2`)` to help gauge the effect. Explain the difference between `anova()` and `Anova`.

reg5 <- lm(auto\_data$mpg ~ auto\_data$horsepower + as.factor(auto\_data$cylinders))  
summary(reg5)

##   
## Call:  
## lm(formula = auto\_data$mpg ~ auto\_data$horsepower + as.factor(auto\_data$cylinders))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9.5917 -2.7067 -0.6102 1.9001 16.3258   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 30.77614 2.41283 12.755 < 2e-16 \*\*\*  
## auto\_data$horsepower -0.10303 0.01133 -9.095 < 2e-16 \*\*\*  
## as.factor(auto\_data$cylinders)4 6.57344 2.16921 3.030 0.00261 \*\*   
## as.factor(auto\_data$cylinders)5 5.07367 3.26661 1.553 0.12120   
## as.factor(auto\_data$cylinders)6 -0.34406 2.18580 -0.157 0.87501   
## as.factor(auto\_data$cylinders)8 0.49738 2.27639 0.218 0.82716   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 4.27 on 386 degrees of freedom  
## Multiple R-squared: 0.7046, Adjusted R-squared: 0.7008   
## F-statistic: 184.1 on 5 and 386 DF, p-value: < 2.2e-16

anova(reg4, reg5)

## Analysis of Variance Table  
##   
## Model 1: auto\_data$mpg ~ auto\_data$cylinders  
## Model 2: auto\_data$mpg ~ auto\_data$horsepower + as.factor(auto\_data$cylinders)  
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 390 9415.9   
## 2 386 7036.7 4 2379.2 32.629 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

car::Anova(reg5)

## Anova Table (Type II tests)  
##   
## Response: auto\_data$mpg  
## Sum Sq Df F value Pr(>F)   
## auto\_data$horsepower 1507.8 1 82.712 < 2.2e-16 \*\*\*  
## as.factor(auto\_data$cylinders) 2349.2 4 32.217 < 2.2e-16 \*\*\*  
## Residuals 7036.7 386   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Cylinders is significant at the 0.01 level when comparing cars with 4 cylinders versus those with just 3. However, it is not significant at this level when comparing cars with 5, 6 or 8 cylinders to those with just 3. The Anova() function performs the F test for each variable. The anova() function may be used to perform a hypothesis test comparing a first model without a variable to a second model with an added variable.

iii. What are the fundamental differences between treating `cylinders` as a numeric and or a factor models?

When treating cylinders as a numeric entity, we find the average effect of an additional cylinder on MPG. However, when treating it as a factor we find the additional effect of 4, 5, 6 or 8 cylinders on MPG compared to the effect of 3 cylinders on MPG (which is our baseline and given by the intercept term). As.factor turns our cylinder variables into indictor variables.

1. Final modelling question: we want to explore the effects of each feature as best as possible. You may explore interactions, feature transformations, higher order terms, or other strategies within reason. The model(s) should be as parsimonious (simple) as possible unless the gain in accuracy is significant from your point of view. Use Mallow's Cp or BIC to select the model.

* Describe the final model and its accuracy. Include diagnostic plots with particular focus on the model residuals.
* Summarize the effects found.
* Predict the mpg of a car that is: built in 1983, in US, red, 180 inches long, 8 cylinders, 350 displacement, 260 as horsepower and weighs 4000 pounds. Give a 95% CI.

data <- auto\_data[1:8] # all variables except name   
fit.all <- lm(auto\_data$mpg ~., data)   
summary(fit.all)

##   
## Call:  
## lm(formula = auto\_data$mpg ~ ., data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9.5903 -2.1565 -0.1169 1.8690 13.0604   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -17.218435 4.644294 -3.707 0.00024 \*\*\*  
## cylinders -0.493376 0.323282 -1.526 0.12780   
## displacement 0.019896 0.007515 2.647 0.00844 \*\*   
## horsepower -0.016951 0.013787 -1.230 0.21963   
## weight -0.006474 0.000652 -9.929 < 2e-16 \*\*\*  
## acceleration 0.080576 0.098845 0.815 0.41548   
## year 0.750773 0.050973 14.729 < 2e-16 \*\*\*  
## origin 1.426141 0.278136 5.127 4.67e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.328 on 384 degrees of freedom  
## Multiple R-squared: 0.8215, Adjusted R-squared: 0.8182   
## F-statistic: 252.4 on 7 and 384 DF, p-value: < 2.2e-16

fit.exh <- regsubsets(auto\_data$mpg ~., data, nvmax=8, method="exhaustive")  
names(fit.exh)

## [1] "np" "nrbar" "d" "rbar" "thetab"   
## [6] "first" "last" "vorder" "tol" "rss"   
## [11] "bound" "nvmax" "ress" "ir" "nbest"   
## [16] "lopt" "il" "ier" "xnames" "method"   
## [21] "force.in" "force.out" "sserr" "intercept" "lindep"   
## [26] "nullrss" "nn" "call"

summary(fit.exh) # List the model with the smallest RSS among each size of the model

## Subset selection object  
## Call: regsubsets.formula(auto\_data$mpg ~ ., data, nvmax = 8, method = "exhaustive")  
## 7 Variables (and intercept)  
## Forced in Forced out  
## cylinders FALSE FALSE  
## displacement FALSE FALSE  
## horsepower FALSE FALSE  
## weight FALSE FALSE  
## acceleration FALSE FALSE  
## year FALSE FALSE  
## origin FALSE FALSE  
## 1 subsets of each size up to 7  
## Selection Algorithm: exhaustive  
## cylinders displacement horsepower weight acceleration year origin  
## 1 ( 1 ) " " " " " " "\*" " " " " " "   
## 2 ( 1 ) " " " " " " "\*" " " "\*" " "   
## 3 ( 1 ) " " " " " " "\*" " " "\*" "\*"   
## 4 ( 1 ) " " "\*" " " "\*" " " "\*" "\*"   
## 5 ( 1 ) " " "\*" "\*" "\*" " " "\*" "\*"   
## 6 ( 1 ) "\*" "\*" "\*" "\*" " " "\*" "\*"   
## 7 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" "\*"

f.e <- summary(fit.exh)  
names(f.e)

## [1] "which" "rsq" "rss" "adjr2" "cp" "bic" "outmat" "obj"

f.e$which

## (Intercept) cylinders displacement horsepower weight acceleration year  
## 1 TRUE FALSE FALSE FALSE TRUE FALSE FALSE  
## 2 TRUE FALSE FALSE FALSE TRUE FALSE TRUE  
## 3 TRUE FALSE FALSE FALSE TRUE FALSE TRUE  
## 4 TRUE FALSE TRUE FALSE TRUE FALSE TRUE  
## 5 TRUE FALSE TRUE TRUE TRUE FALSE TRUE  
## 6 TRUE TRUE TRUE TRUE TRUE FALSE TRUE  
## 7 TRUE TRUE TRUE TRUE TRUE TRUE TRUE  
## origin  
## 1 FALSE  
## 2 FALSE  
## 3 TRUE  
## 4 TRUE  
## 5 TRUE  
## 6 TRUE  
## 7 TRUE

f.e$rsq

## [1] 0.6926304 0.8081803 0.8174522 0.8180977 0.8200242 0.8211691 0.8214781

f.e$rss

## [1] 7321.234 4568.952 4348.105 4332.729 4286.842 4259.571 4252.213

f.e$bic

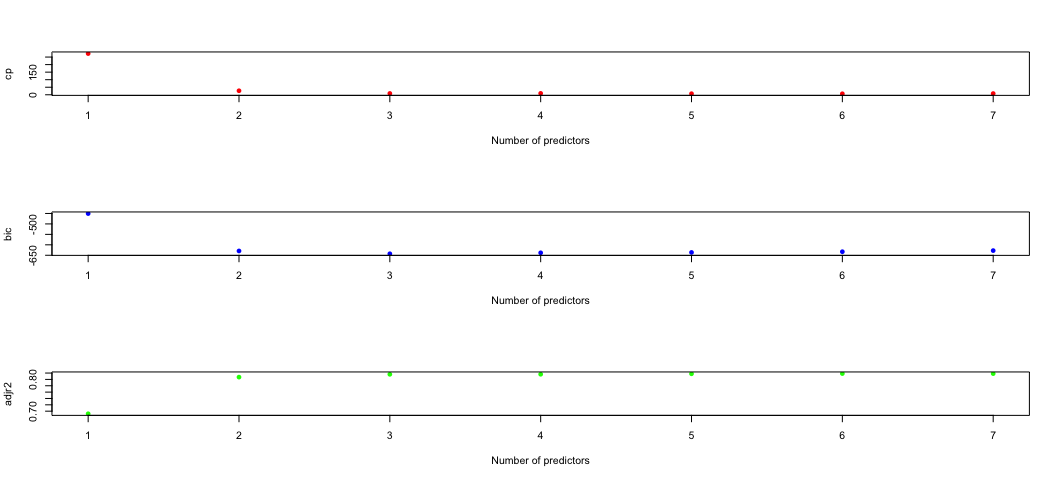
## [1] -450.5016 -629.3564 -642.8063 -638.2237 -636.4261 -632.9566 -627.6631

f.e$cp

## [1] 273.150806 26.603456 8.659680 9.271088 7.127265 6.664509  
## [7] 8.000000

Here are the plots of cp vs number of predictors. Similarly we have the plots of BIC v.s. number of the predictors

par(mfrow=c(3,1))  
plot(f.e$cp, xlab="Number of predictors",   
 ylab="cp", col="red", type="p", pch=16)  
plot(f.e$bic, xlab="Number of predictors",   
 ylab="bic", col="blue", type="p", pch=16)  
plot(f.e$adjr2, xlab="Number of predictors",   
 ylab="adjr2", col="green", type="p", pch=16)



par(mfrow=c(1,1))

In this case we may use 6 variables as our model size.

opt.size <- which.min(f.e$cp) # locate the optimal model size  
opt.size

## [1] 6

fit.exh.var <- f.e$which   
fit.exh.var[opt.size,] # this gives us the optimal variables selected

## (Intercept) cylinders displacement horsepower weight   
## TRUE TRUE TRUE TRUE TRUE   
## acceleration year origin   
## FALSE TRUE TRUE

fit.exh.var[6,]

## (Intercept) cylinders displacement horsepower weight   
## TRUE TRUE TRUE TRUE TRUE   
## acceleration year origin   
## FALSE TRUE TRUE

To pull out the final model:

fit.exh.6 <- lm(auto\_data$mpg ~., data[fit.exh.var[6,]])   
summary(fit.exh.6) # Note: there is no guarantee that all the var's in the final model are significant.

##   
## Call:  
## lm(formula = auto\_data$mpg ~ ., data = data[fit.exh.var[6, ]])  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9.7604 -2.1791 -0.1535 1.8524 13.1209   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.556e+01 4.175e+00 -3.728 0.000222 \*\*\*  
## cylinders -5.067e-01 3.227e-01 -1.570 0.117236   
## displacement 1.927e-02 7.472e-03 2.579 0.010287 \*   
## horsepower -2.389e-02 1.084e-02 -2.205 0.028031 \*   
## weight -6.218e-03 5.714e-04 -10.883 < 2e-16 \*\*\*  
## year 7.475e-01 5.079e-02 14.717 < 2e-16 \*\*\*  
## origin 1.428e+00 2.780e-01 5.138 4.43e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.326 on 385 degrees of freedom  
## Multiple R-squared: 0.8212, Adjusted R-squared: 0.8184   
## F-statistic: 294.6 on 6 and 385 DF, p-value: < 2.2e-16

library(car)

##   
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':  
##   
## recode

Anova(fit.exh.6) # Once again this gives us the test for each var at a time.

## Anova Table (Type II tests)  
##   
## Response: auto\_data$mpg  
## Sum Sq Df F value Pr(>F)   
## cylinders 27.3 1 2.4649 0.11724   
## displacement 73.6 1 6.6498 0.01029 \*   
## horsepower 53.8 1 4.8629 0.02803 \*   
## weight 1310.4 1 118.4429 < 2.2e-16 \*\*\*  
## year 2396.2 1 216.5774 < 2.2e-16 \*\*\*  
## origin 292.0 1 26.3940 4.434e-07 \*\*\*  
## Residuals 4259.6 385   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Forward Selection

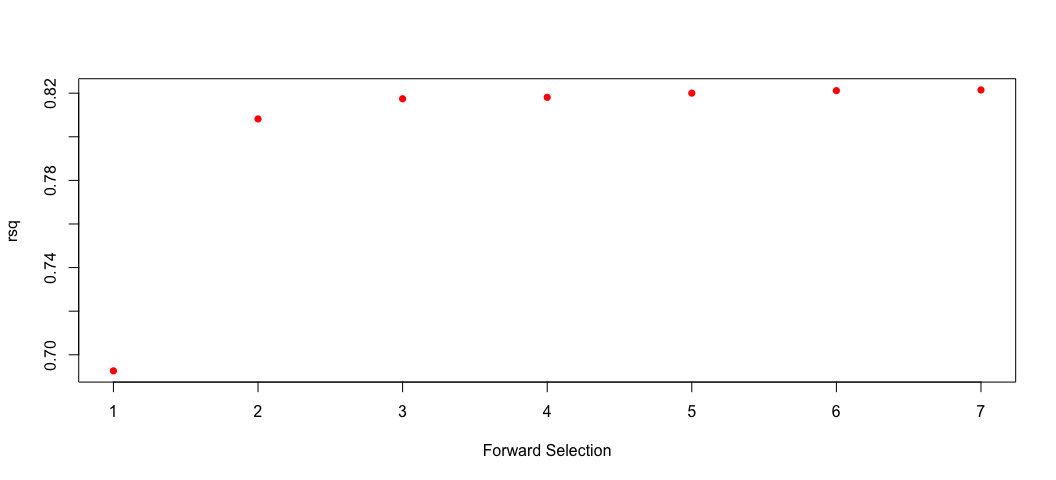
fit.forward <- regsubsets(auto\_data$mpg ~., data, nvmax=8, method="forward")  
fit.forward

## Subset selection object  
## Call: regsubsets.formula(auto\_data$mpg ~ ., data, nvmax = 8, method = "forward")  
## 7 Variables (and intercept)  
## Forced in Forced out  
## cylinders FALSE FALSE  
## displacement FALSE FALSE  
## horsepower FALSE FALSE  
## weight FALSE FALSE  
## acceleration FALSE FALSE  
## year FALSE FALSE  
## origin FALSE FALSE  
## 1 subsets of each size up to 7  
## Selection Algorithm: forward

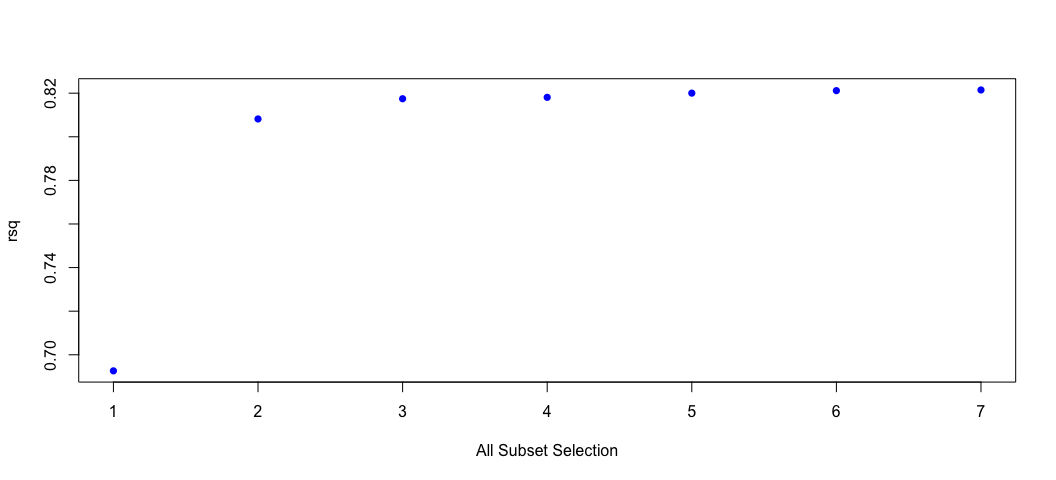
f.f <- summary(fit.forward)  
f.f

## Subset selection object  
## Call: regsubsets.formula(auto\_data$mpg ~ ., data, nvmax = 8, method = "forward")  
## 7 Variables (and intercept)  
## Forced in Forced out  
## cylinders FALSE FALSE  
## displacement FALSE FALSE  
## horsepower FALSE FALSE  
## weight FALSE FALSE  
## acceleration FALSE FALSE  
## year FALSE FALSE  
## origin FALSE FALSE  
## 1 subsets of each size up to 7  
## Selection Algorithm: forward  
## cylinders displacement horsepower weight acceleration year origin  
## 1 ( 1 ) " " " " " " "\*" " " " " " "   
## 2 ( 1 ) " " " " " " "\*" " " "\*" " "   
## 3 ( 1 ) " " " " " " "\*" " " "\*" "\*"   
## 4 ( 1 ) " " "\*" " " "\*" " " "\*" "\*"   
## 5 ( 1 ) " " "\*" "\*" "\*" " " "\*" "\*"   
## 6 ( 1 ) "\*" "\*" "\*" "\*" " " "\*" "\*"   
## 7 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" "\*"

plot(f.f$rsq, ylab="rsq", col="red", type="p", pch=16,  
 xlab="Forward Selection")



plot(f.e$rsq, ylab="rsq", col="blue", type="p", pch=16,  
 xlab="All Subset Selection")



par(mfrow=c(1,1))  
coef(fit.forward, 6)

## (Intercept) cylinders displacement horsepower weight   
## -15.563492306 -0.506685137 0.019269286 -0.023895029 -0.006218311   
## year origin   
## 0.747515952 1.428241885

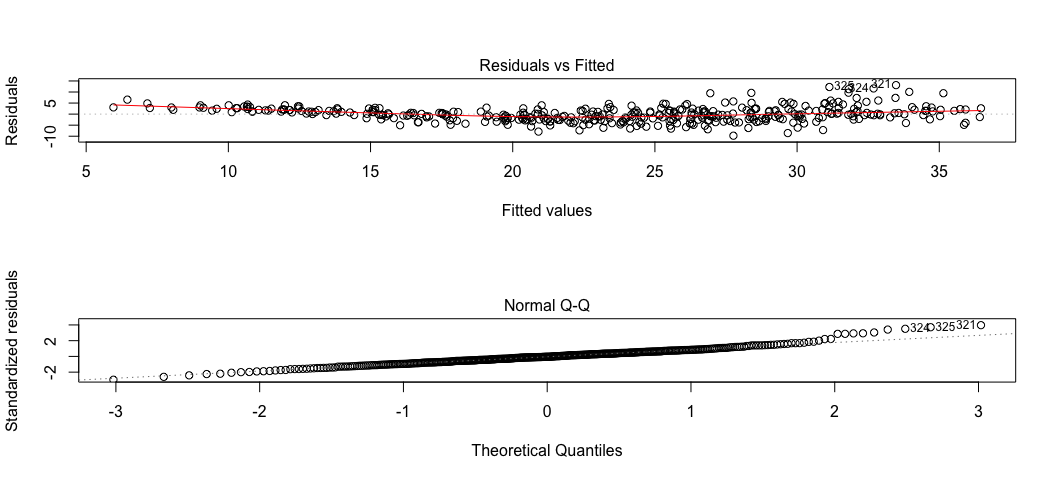
Final Model:

fit.final <- lm(auto\_data$mpg ~ auto\_data$cylinders + auto\_data$displacement + auto\_data$horsepower + auto\_data$weight + auto\_data$year + auto\_data$origin)  
summary(fit.final)

##   
## Call:  
## lm(formula = auto\_data$mpg ~ auto\_data$cylinders + auto\_data$displacement +   
## auto\_data$horsepower + auto\_data$weight + auto\_data$year +   
## auto\_data$origin)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9.7604 -2.1791 -0.1535 1.8524 13.1209   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.556e+01 4.175e+00 -3.728 0.000222 \*\*\*  
## auto\_data$cylinders -5.067e-01 3.227e-01 -1.570 0.117236   
## auto\_data$displacement 1.927e-02 7.472e-03 2.579 0.010287 \*   
## auto\_data$horsepower -2.389e-02 1.084e-02 -2.205 0.028031 \*   
## auto\_data$weight -6.218e-03 5.714e-04 -10.883 < 2e-16 \*\*\*  
## auto\_data$year 7.475e-01 5.079e-02 14.717 < 2e-16 \*\*\*  
## auto\_data$origin 1.428e+00 2.780e-01 5.138 4.43e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.326 on 385 degrees of freedom  
## Multiple R-squared: 0.8212, Adjusted R-squared: 0.8184   
## F-statistic: 294.6 on 6 and 385 DF, p-value: < 2.2e-16

Model diagnostics

par(mfrow=c(2,1))  
plot(fit.final,1) # Everything seems to be fine. I only use the first two plots.  
plot(fit.final,2)



par(mfrow=c(1,1))

I decided to use the exhausitive model with 6 predictors as my final model. These predictors are "cylinders", "displacement", "horsepower", "weight", "year" and "origin". I used Mallow's Cp to generate this model. Using this model, we find the LS coefficients for each predictor holding all others constant, as shown above. An additional cylinder decreases MPG by -5.067e-01 (this is not significant). An additional unit of displacement increases MPG by 1.927e-02. An additional unit of horsepower decreases MPG by -2.389e-02. An additional pound decreases MPG by -6.218e-03. With each year the MPG increases by 7.475e-01. And the effect of origin on MPG is 1.428e+00.

newcar <- data[1,] # Create a new row with same structure as in auto\_data  
newcar[1] <- "NA"  
newcar[2] <- 8 # Assign features for the new car  
newcar[3] <- 350  
newcar[4] <- 260  
newcar[5] <- 4000  
newcar[6] <- "NA"  
newcar[7] <- 83  
newcar[8] <- 1  
newcar

## mpg cylinders displacement horsepower weight acceleration year origin  
## 1 NA 8 350 260 4000 NA 83 1

Get a 95% CI of the salary for the player

newcar.m <- predict(fit.final, newcar, interval="confidence", se.fit=TRUE)   
newcar.m # in log scale  
exp(newcar.m$fit)  
  
newcar.p <- predict(fit.final, newcar, interval="prediction", se.fit=TRUE)   
newcar.p # in log scale  
exp(newcar.p$fit)

## Problem 2

Do ISLR, page 262, problem 8, and write up the answer here. This question is designed to help us understanding model selections through simulations. You may want to do this first before answering question 4 in Problem 1.

1. Use the rnorm() function to generate a predictor X of length n = 100, as well as a noise vector ε of length n = 100.

x <- rnorm(100)  
ep <- rnorm(100)

1. Generate a response vector Y of length n = 100 according to the model Y = β0 +β1X +β2X2 +β3X3 +ε, where β0, β1, β2,and β3 are constants of your choice

y = 2 + 3\*x + 4\*x^2+ 2\*x^3+ 5\*x^4 + 4\*x^5 + 3\*x^6 + 6\*x^7 + 2\*x^8 + 3\*x^9 + 9\*x^10 + ep  
data3 <- data.frame(x, x^2, x^3, x^4, x^5, x^6, x^7, x^8, x^9, x^10, y)

fit.exh2 <- regsubsets(y~., data = data3, nvmax=10, method="exhaustive")  
names(fit.exh2)

## [1] "np" "nrbar" "d" "rbar" "thetab"   
## [6] "first" "last" "vorder" "tol" "rss"   
## [11] "bound" "nvmax" "ress" "ir" "nbest"   
## [16] "lopt" "il" "ier" "xnames" "method"   
## [21] "force.in" "force.out" "sserr" "intercept" "lindep"   
## [26] "nullrss" "nn" "call"

summary(fit.exh2) # List the model with the smallest RSS among each size of the model

## Subset selection object  
## Call: regsubsets.formula(y ~ ., data = data3, nvmax = 10, method = "exhaustive")  
## 10 Variables (and intercept)  
## Forced in Forced out  
## x FALSE FALSE  
## x.2 FALSE FALSE  
## x.3 FALSE FALSE  
## x.4 FALSE FALSE  
## x.5 FALSE FALSE  
## x.6 FALSE FALSE  
## x.7 FALSE FALSE  
## x.8 FALSE FALSE  
## x.9 FALSE FALSE  
## x.10 FALSE FALSE  
## 1 subsets of each size up to 10  
## Selection Algorithm: exhaustive  
## x x.2 x.3 x.4 x.5 x.6 x.7 x.8 x.9 x.10  
## 1 ( 1 ) " " " " " " " " " " " " " " " " " " "\*"   
## 2 ( 1 ) " " " " " " " " " " " " " " " " "\*" "\*"   
## 3 ( 1 ) " " " " " " " " "\*" " " " " " " "\*" "\*"   
## 4 ( 1 ) " " " " " " " " " " "\*" "\*" " " "\*" "\*"   
## 5 ( 1 ) " " " " " " " " "\*" "\*" "\*" " " "\*" "\*"   
## 6 ( 1 ) " " " " "\*" "\*" " " " " "\*" "\*" "\*" "\*"   
## 7 ( 1 ) "\*" " " "\*" "\*" " " " " "\*" "\*" "\*" "\*"   
## 8 ( 1 ) "\*" "\*" " " " " "\*" "\*" "\*" "\*" "\*" "\*"   
## 9 ( 1 ) "\*" "\*" " " "\*" "\*" "\*" "\*" "\*" "\*" "\*"   
## 10 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*"

f.e <- summary(fit.exh2)  
names(f.e)

## [1] "which" "rsq" "rss" "adjr2" "cp" "bic" "outmat" "obj"

f.e$which

## (Intercept) x x.2 x.3 x.4 x.5 x.6 x.7 x.8 x.9 x.10  
## 1 TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE  
## 2 TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE TRUE  
## 3 TRUE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE TRUE TRUE  
## 4 TRUE FALSE FALSE FALSE FALSE FALSE TRUE TRUE FALSE TRUE TRUE  
## 5 TRUE FALSE FALSE FALSE FALSE TRUE TRUE TRUE FALSE TRUE TRUE  
## 6 TRUE FALSE FALSE TRUE TRUE FALSE FALSE TRUE TRUE TRUE TRUE  
## 7 TRUE TRUE FALSE TRUE TRUE FALSE FALSE TRUE TRUE TRUE TRUE  
## 8 TRUE TRUE TRUE FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE  
## 9 TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE  
## 10 TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE

f.e$rsq

## [1] 0.9728965 0.9999697 0.9999805 0.9999998 1.0000000 1.0000000 1.0000000  
## [8] 1.0000000 1.0000000 1.0000000

f.e$rss

## [1] 1.228961e+09 1.374128e+06 8.823252e+05 8.486564e+03 8.189546e+02  
## [6] 1.418715e+02 1.320304e+02 1.139956e+02 1.110048e+02 1.065303e+02

f.e$bic

## [1] -351.5988 -1026.6041 -1066.3003 -1526.1028 -1755.3187 -1926.0243  
## [7] -1928.6080 -1938.6901 -1936.7436 -1936.2528

f.e$cp

## [1] 1.026727e+09 1.147912e+06 7.370407e+05 7.000043e+03 5.961901e+02  
## [6] 3.252557e+01 2.630393e+01 1.323687e+01 1.273817e+01 1.100000e+01