Midterm Math 158

Joshua Jansen-Montoya

2022-11-05

First, we can load the dataset as follows and then format our response variable data as defined here,

```
  library(MASS)
  Y <- 100/(Cars93$MPG.city)
```

Introduction

In this report, we will be attempting to construct a linear model that best uses the different variables from the Cars93 dataset to describe the response variable "MPG.city". That is, we are trying to answer the question of which variables in the Cars93 dataset are most significant and can best predict the response variable, "MPG.city". In answering this question, we will first construct a mapping for our non-numeric possible response variables to make the variables viable for linear regression. We will be using the linear models generated by R, then we will be focusing on optimizing this linear model using a step wise testing approach using stepwise regression, then utilizing Box-Cox transformations on our response variable. Finally we will be using our different techniques of evaluating our linear model to see if it truly meets our assumptions for normality using different R functions as well as to test for strenuous points that may affect our linear model. Finally, we will use different robust linear regression techniques to further evaluate the performance of our linear model. ### Data Evaluation The data set that we are using consists of data from 93 different cars on sale in the U.S. and informs the reader as to different specs to the car which can be seen in the summary. We can look at the summary as a preliminary evaluation of our data by simply summarizing our data.

```
?Cars93
  summary(Cars93)
```

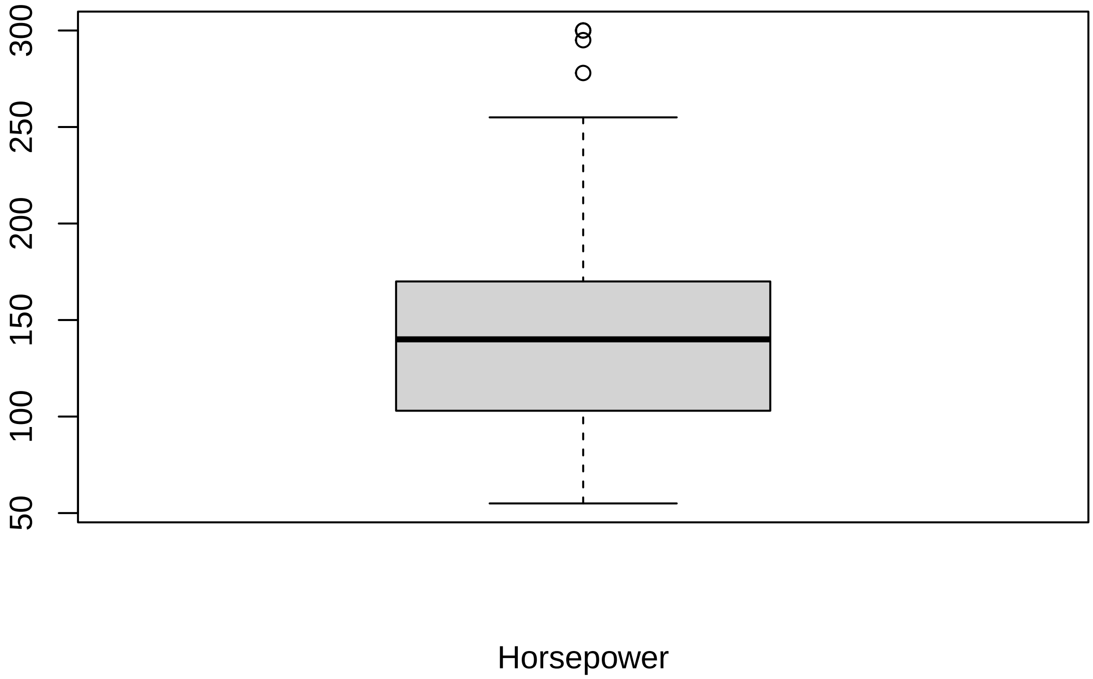
```
##     Manufacturer     Model         Type      Min.Price         Price
##  Chevrolet: 8    100    : 1   Compact:16   Min.   : 6.70   Min.   : 7.40
##  Ford     : 8    190E   : 1   Large  :11   1st Qu.:10.80   1st Qu.:12.20
##  Dodge    : 6    240    : 1   Midsize:22   Median :14.70   Median :17.70
##  Mazda    : 5    300E   : 1   Small  :21   Mean   :17.13   Mean   :19.51
##  Pontiac  : 5    323    : 1   Sporty :14   3rd Qu.:20.30   3rd Qu.:23.30
##  Buick    : 4    535i   : 1   Van    : 9   Max.   :45.40   Max.   :61.90
##  (Other)  :57    (Other):87
##    Max.Price       MPG.city      MPG.highway                  AirBags
##  Min.   : 7.9   Min.   :15.00   Min.   :20.00   Driver & Passenger:16
##  1st Qu.:14.7   1st Qu.:18.00   1st Qu.:26.00   Driver only       :43
##  Median :19.6   Median :21.00   Median :28.00   None              :34
##  Mean   :21.9   Mean   :22.37   Mean   :29.09
##  3rd Qu.:25.3   3rd Qu.:25.00   3rd Qu.:31.00
##  Max.   :80.0   Max.   :46.00   Max.   :50.00
##
##  DriveTrain  Cylinders    EngineSize    Horsepower           RPM
##  4WD  :10   3     : 3   Min.   :1.000   Min.   : 55.0   Min.   :3800
##  Front:67   4     :49   1st Qu.:1.800   1st Qu.:103.0   1st Qu.:4800
##  Rear :16   5     : 2   Median :2.400   Median :140.0   Median :5200
##             6     :31   Mean   :2.668   Mean   :143.8   Mean   :5281
##             8     : 7   3rd Qu.:3.300   3rd Qu.:170.0   3rd Qu.:5750
##             rotary: 1   Max.   :5.700   Max.   :300.0   Max.   :6500
##
##   Rev.per.mile  Man.trans.avail Fuel.tank.capacity   Passengers
##  Min.   :1320   No :32          Min.   : 9.20      Min.   :2.000
##  1st Qu.:1985   Yes:61          1st Qu.:14.50      1st Qu.:4.000
##  Median :2340                   Median :16.40      Median :5.000
##  Mean   :2332                   Mean   :16.66      Mean   :5.086
##  3rd Qu.:2565                   3rd Qu.:18.80      3rd Qu.:6.000
##  Max.   :3755                   Max.   :27.00      Max.   :8.000
##
##      Length        Wheelbase          Width        Turn.circle
##  Min.   :141.0   Min.   : 90.0   Min.   :60.00   Min.   :32.00
##  1st Qu.:174.0   1st Qu.: 98.0   1st Qu.:67.00   1st Qu.:37.00
##  Median :183.0   Median :103.0   Median :69.00   Median :39.00
##  Mean   :183.2   Mean   :103.9   Mean   :69.38   Mean   :38.96
##  3rd Qu.:192.0   3rd Qu.:110.0   3rd Qu.:72.00   3rd Qu.:41.00
##  Max.   :219.0   Max.   :119.0   Max.   :78.00   Max.   :45.00
##
##  Rear.seat.room   Luggage.room       Weight        Origin             Make
##  Min.   :19.00   Min.   : 6.00   Min.   :1695   USA    :48   Acura Integra: 1
##  1st Qu.:26.00   1st Qu.:12.00   1st Qu.:2620   non-USA:45   Acura Legend : 1
##  Median :27.50   Median :14.00   Median :3040                Audi 100     : 1
##  Mean   :27.83   Mean   :13.89   Mean   :3073                Audi 90      : 1
##  3rd Qu.:30.00   3rd Qu.:15.00   3rd Qu.:3525                BMW 535i     : 1
##  Max.   :36.00   Max.   :22.00   Max.   :4105                Buick Century: 1
##  NA's   :2       NA's   :11                                  (Other)      :87
```

Looking at our results, we can see that the Make category would likely not be useful for our exploits. Similarly, for the sake of our investigation, I do not expect that Airbags, Origin, Model would be useful for our model. We can see that we have a couple of different variables that are not numeric, and thus, we will have to assign a scale to them. For the category, type, let us assign the scale (0,5) for each of Compact, Small, Midsize, Large, Sporty, and Van respectively. Similarly, we can assign (0,1) for Man.trans.available, (1,3) for Drive.Train such that Front: 0, Back: 1, 4WD: 2, and Cylinder such that rotary: -1, 3: 0, 4: 1, 5: 2, 6: 3, 8: 4. We can do this with the following code,

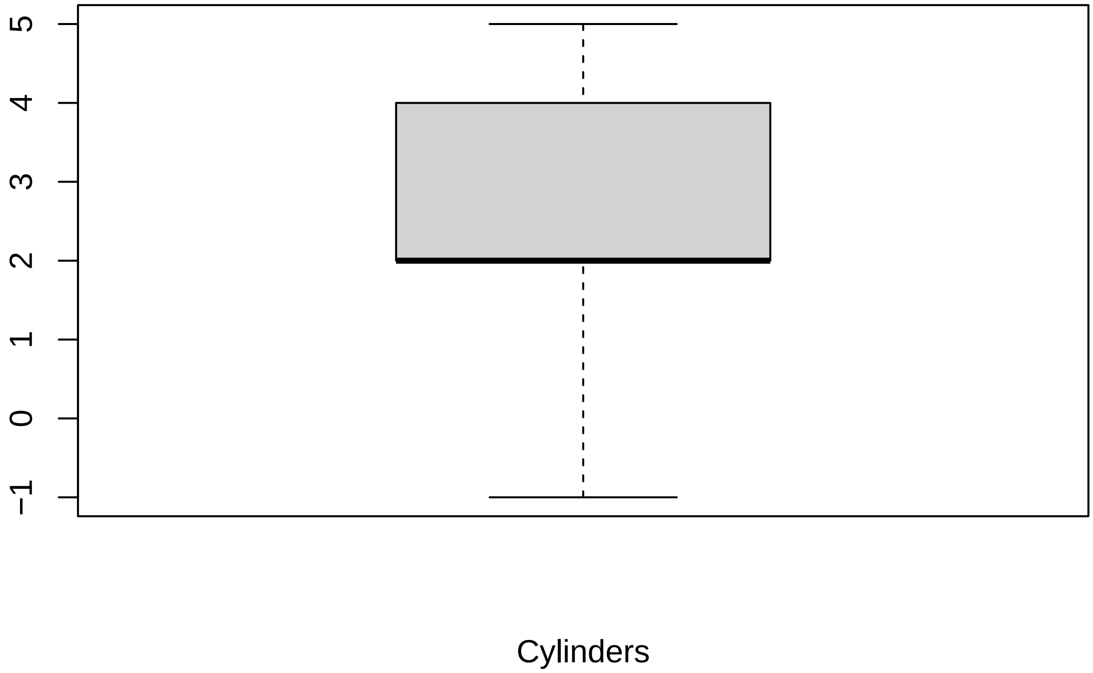
```
type <- ifelse(Cars93$Type == "Compact", 0, ifelse(Cars93$Type == " Small", 1, ifelse(Cars93$Type ==
trans <- ifelse(Cars93$Man.trans.avail == "Yes", 1, 0)
drive <- ifelse(Cars93$DriveTrain == "Front", 1, ifelse(Cars93$DriveTrain == "Back", 2, 3))
cylinders <- ifelse(Cars93$Cylinders == 3, 1, ifelse(Cars93$Cylinders == 4, 2, ifelse(Cars93$Cylinders
```

Now, using reasoning, we can think that some covariates of interest would be Horsepower, Cylinders, Type, MPG.highway, Length, Width, Weight, and Passengers. Thus, we can find boxplots for these covariates with the following code,

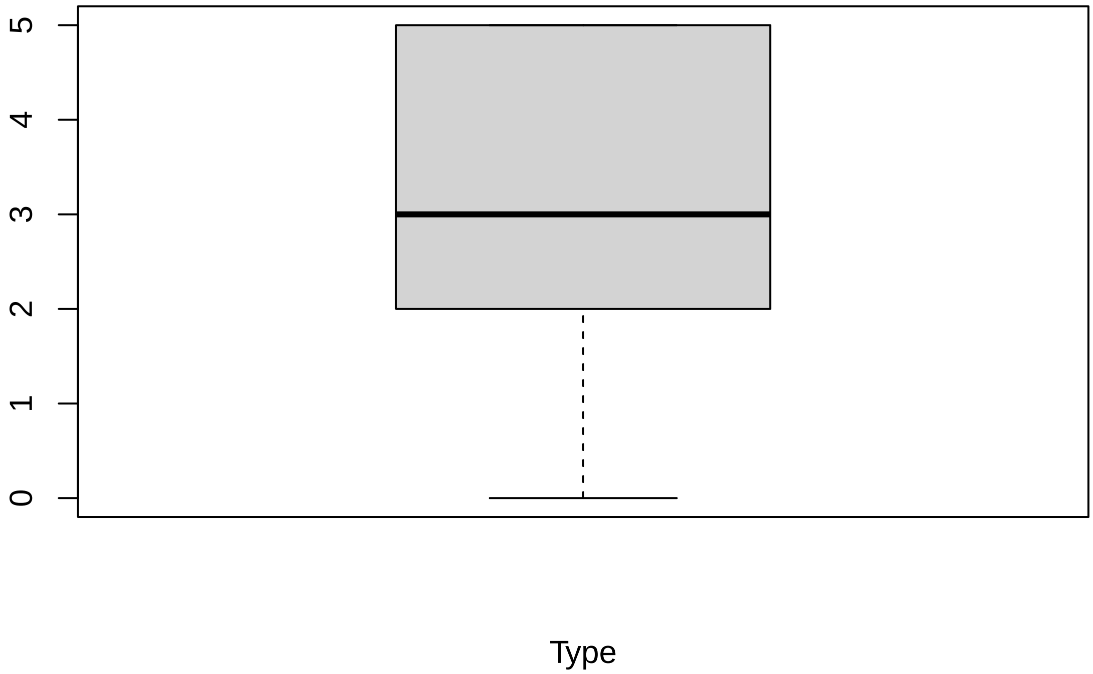
```
boxplot(Cars93$Horsepower, xlab = "Horsepower")
```



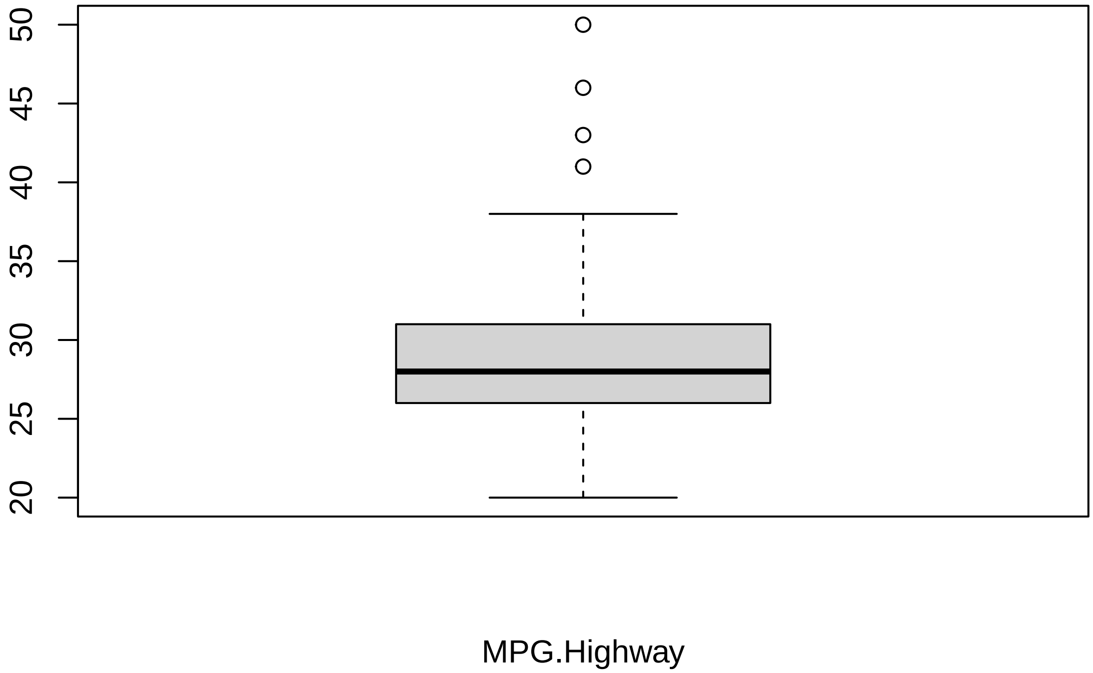
```
boxplot(cylinders, xlab = "Cylinders")
```



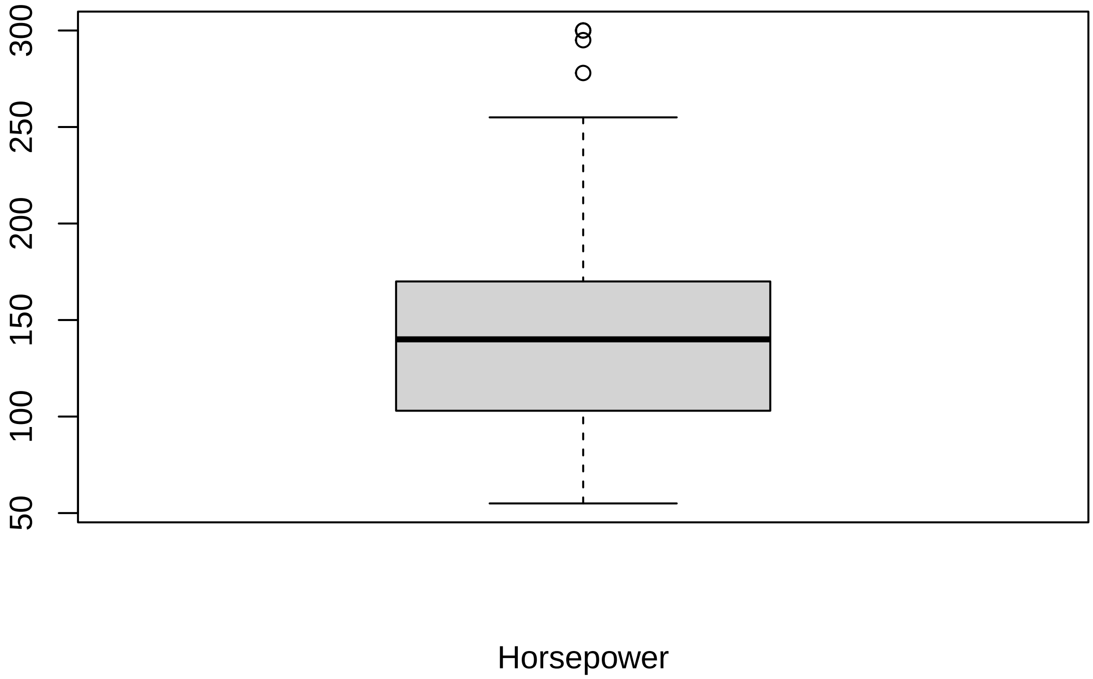
```
boxplot(type, xlab = "Type")
```



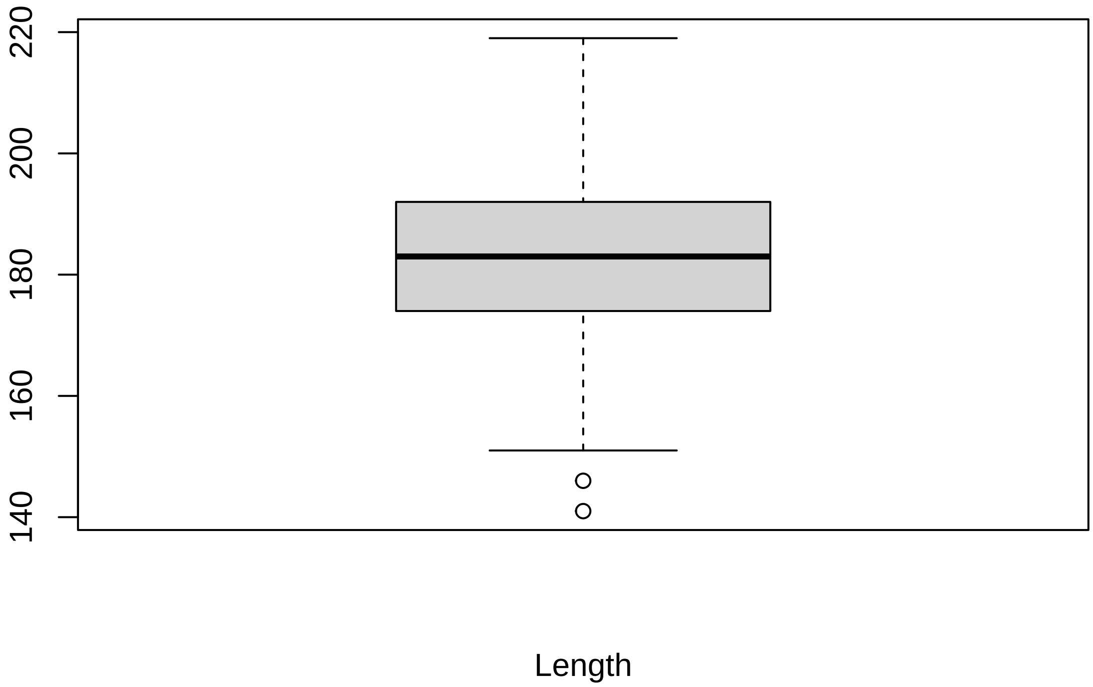
```r
boxplot(Cars93$MPG.highway, xlab = "MPG.Highway")
```



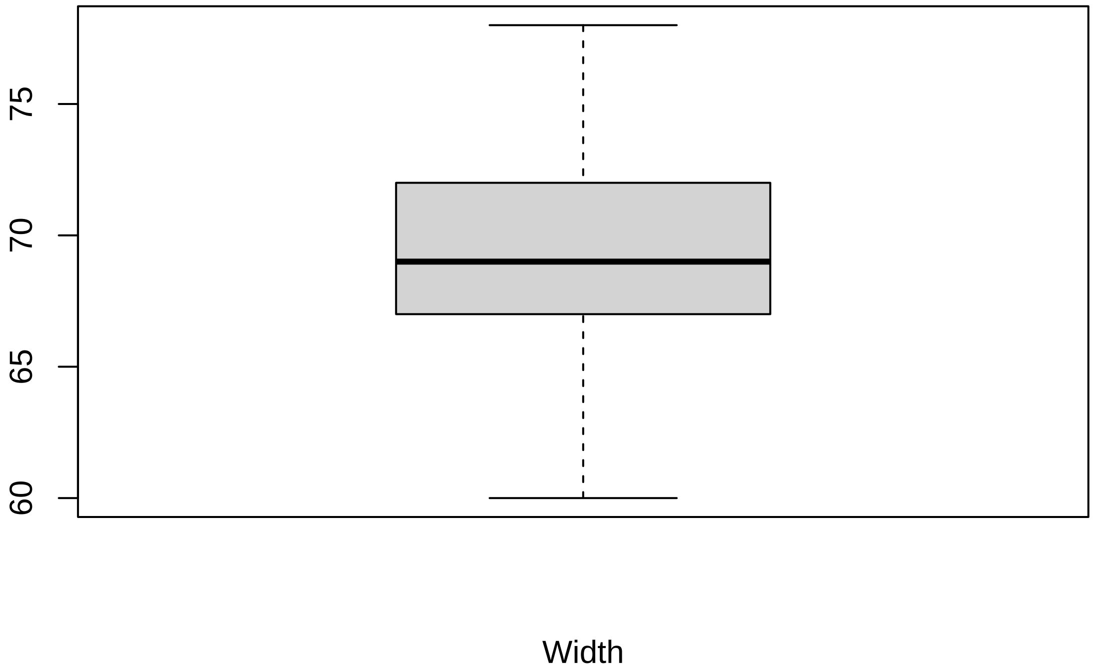
```r
boxplot(Cars93$Horsepower, xlab ="Horsepower")
```



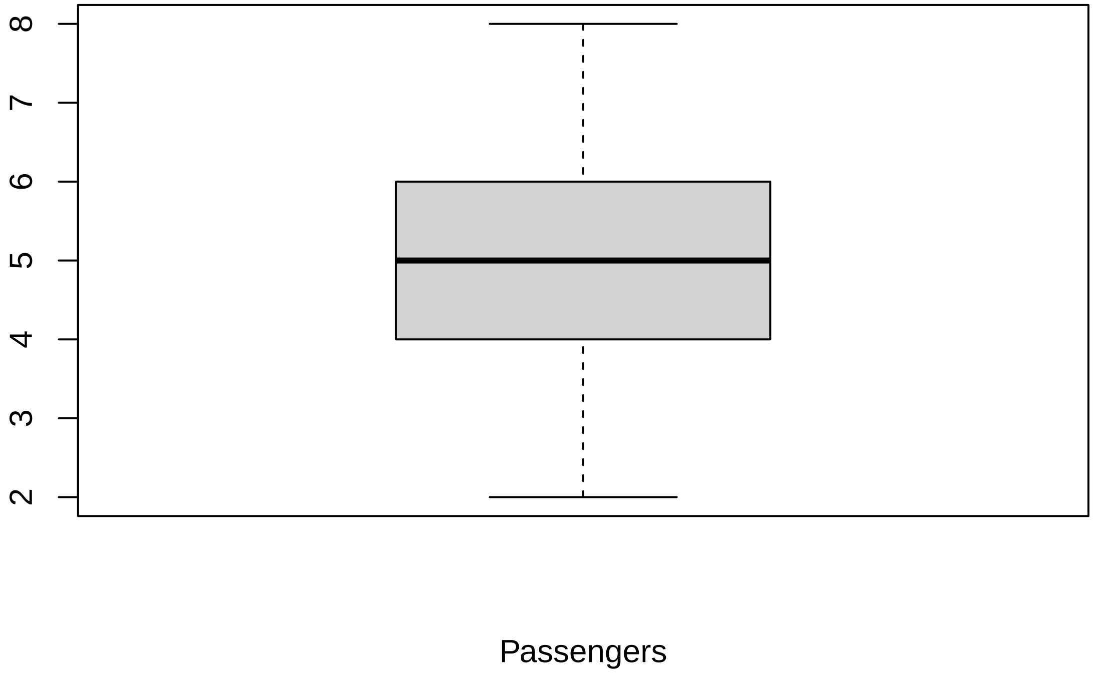
```r
boxplot(Cars93$Length, xlab = "Length")
```



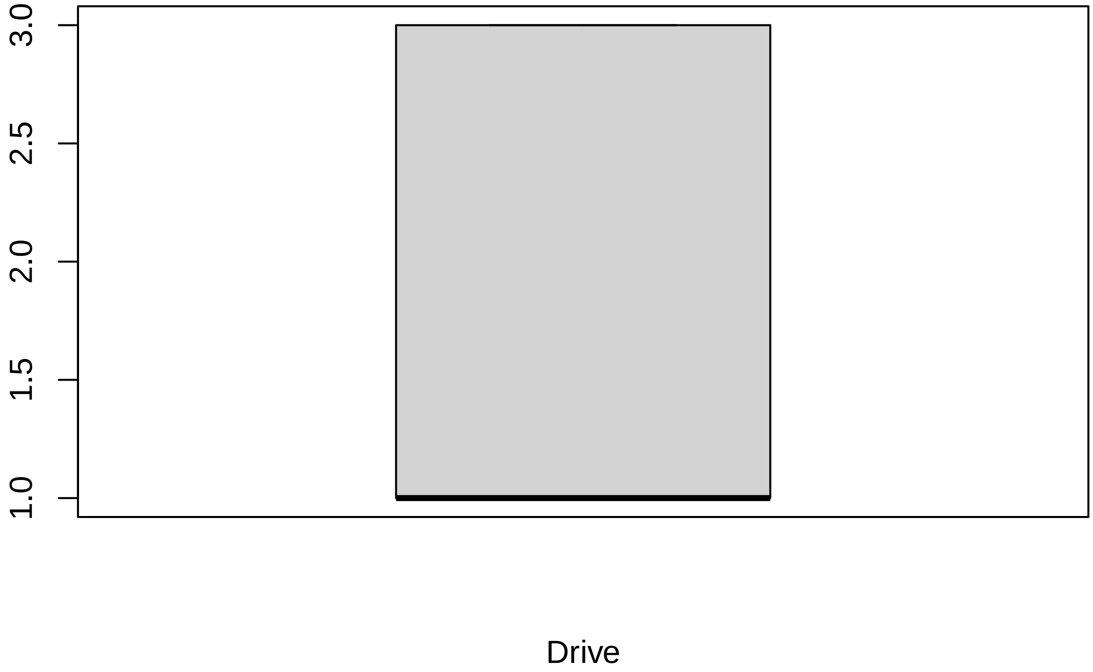
```r
boxplot(Cars93$Width, xlab = "Width")
```



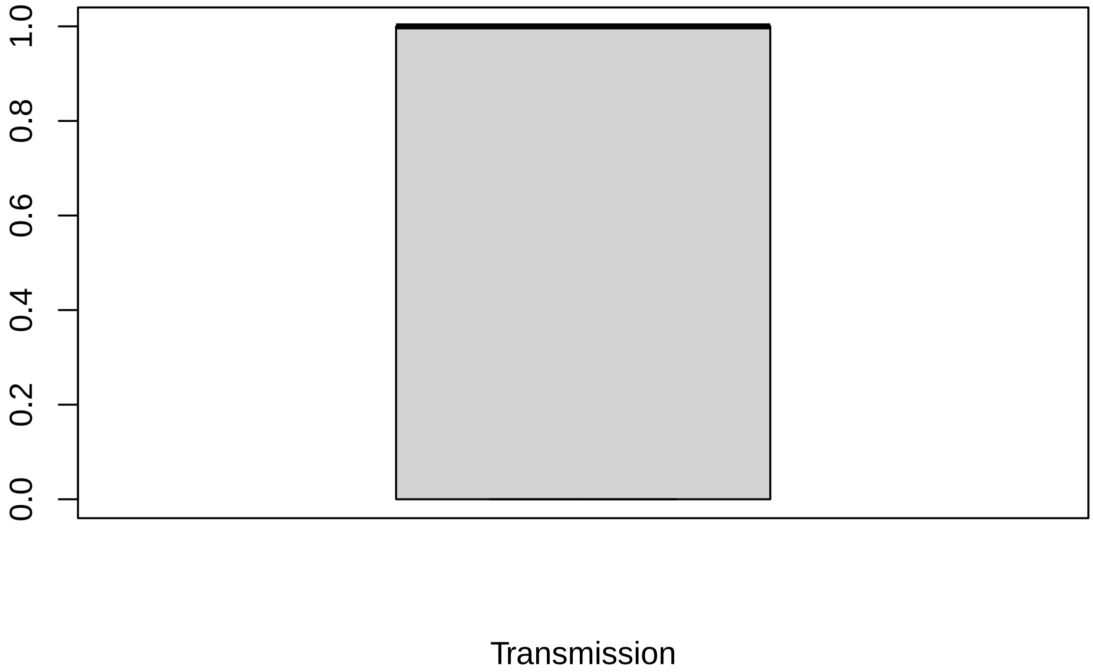
```
boxplot(Cars93$Passengers, xlab = "Passengers")
```



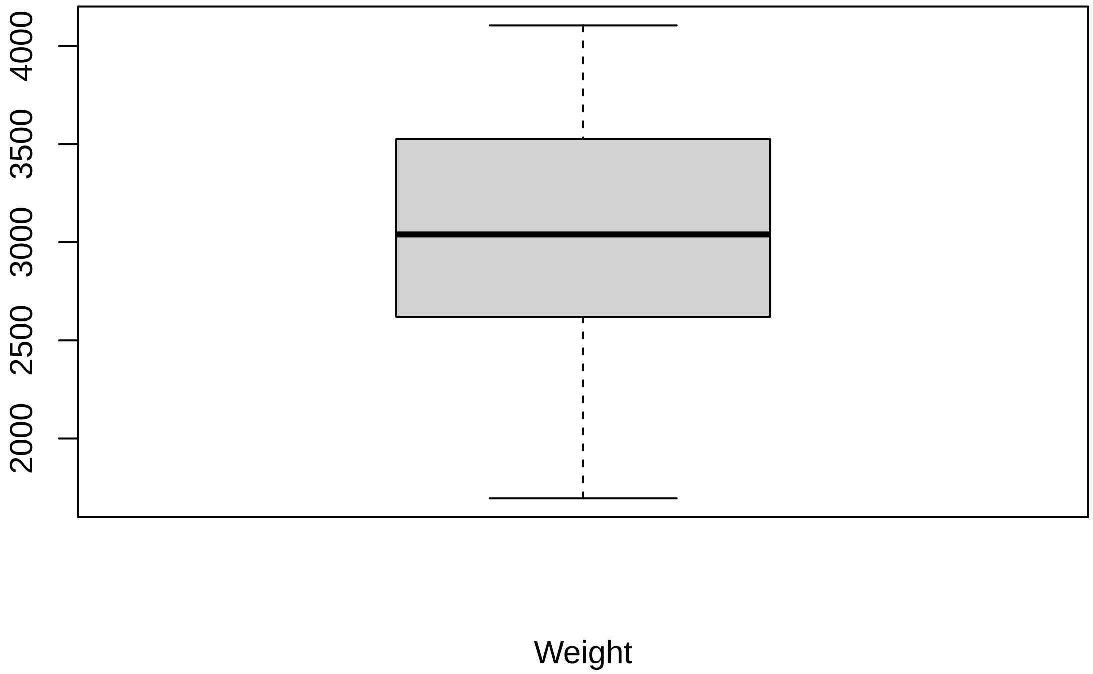
```
boxplot(drive, xlab = "Drive")
```



```r
boxplot(trans, xlab = "Transmission")
```

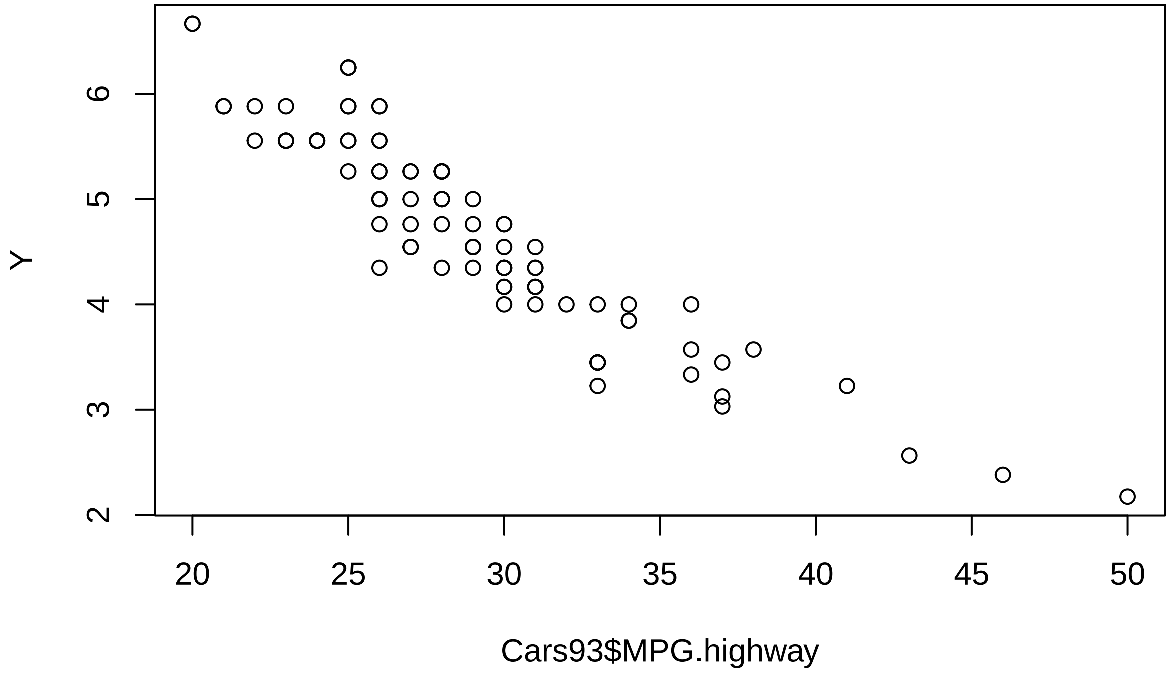


```r
boxplot(Cars93$Weight, xlab = "Weight")
```

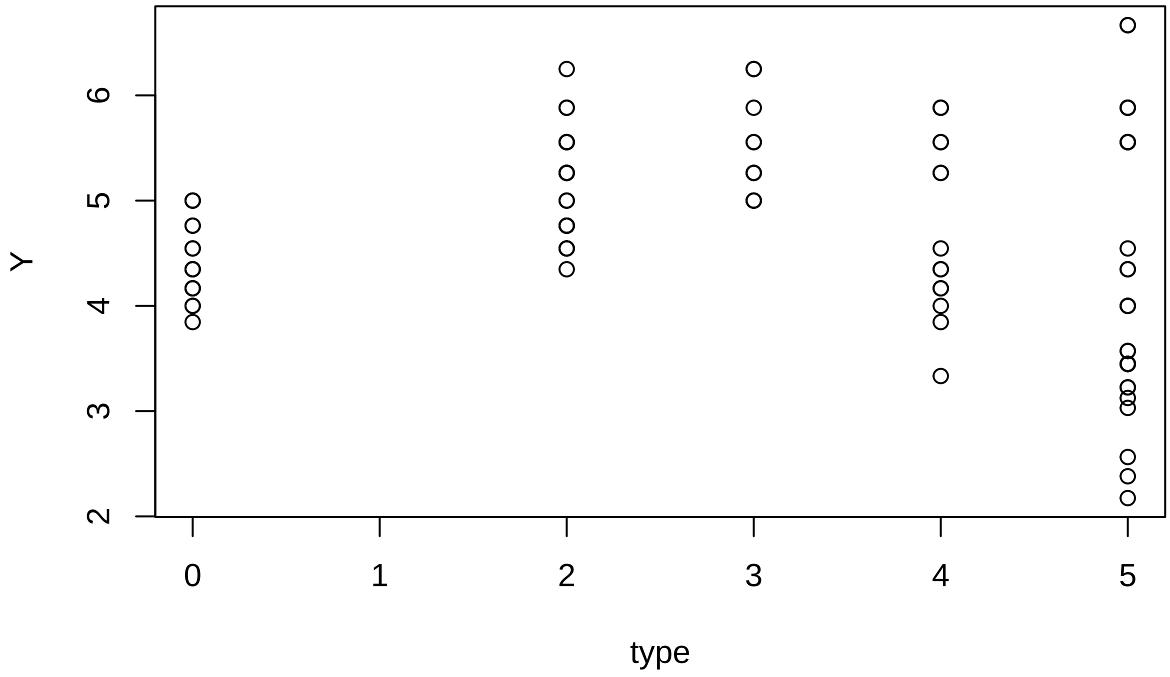


From this, we can see that our values are reasonably well distributed and have fairly few outliers. Now, we can graph a few of our variables of interest versus our desired response variable as follows,

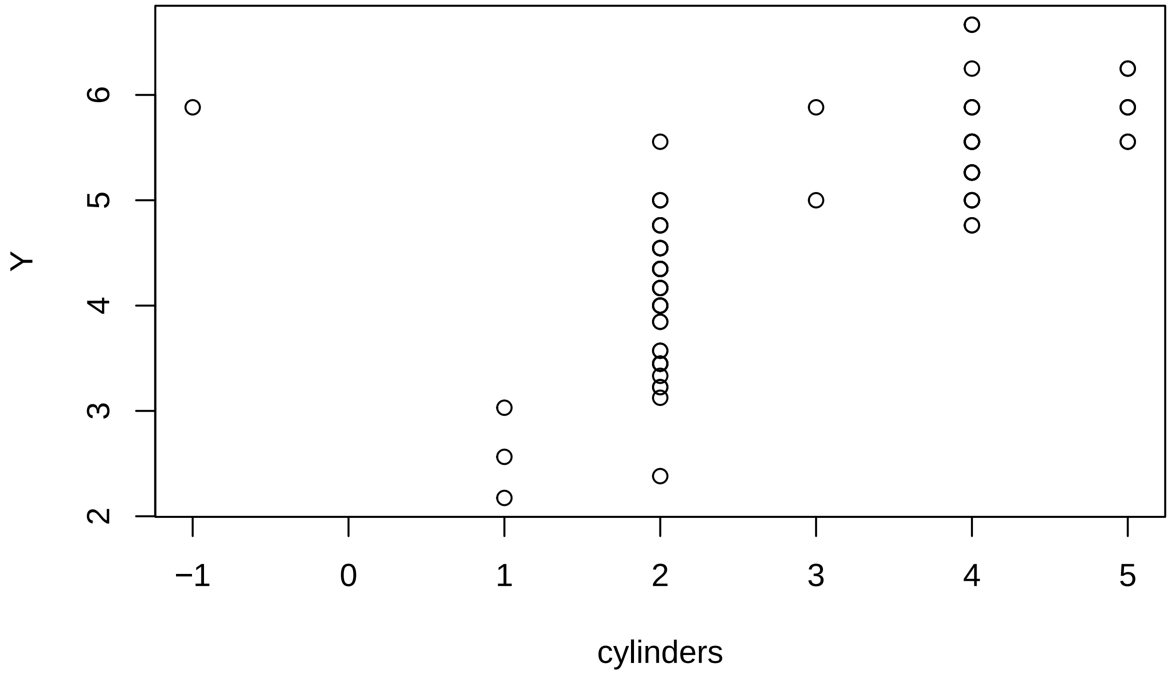
```
plot(Cars93$MPG.highway, Y)
```



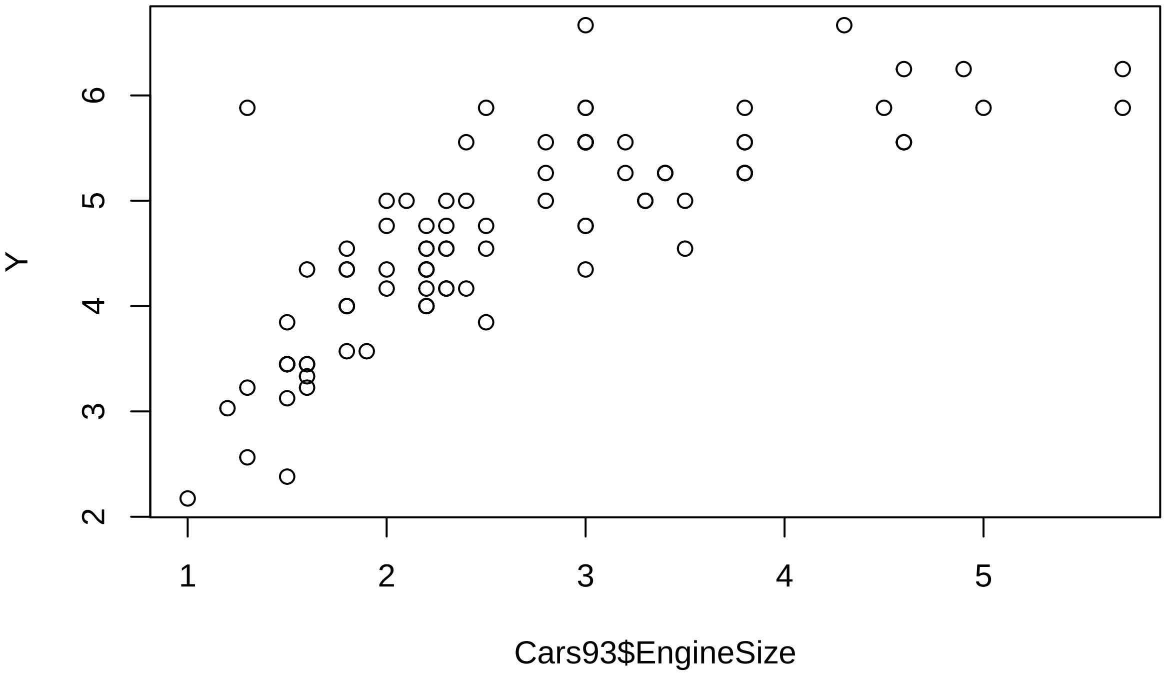
```
plot(type, Y)
```



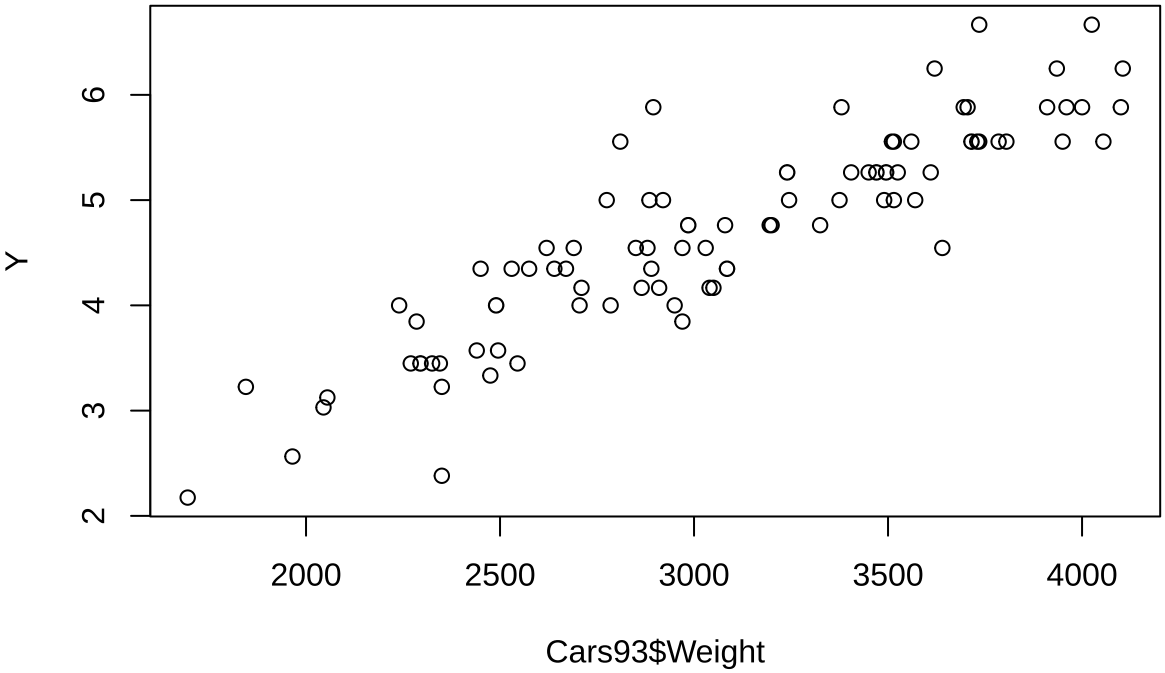
```
plot(cylinders, Y)
```



```
plot(Cars93$EngineSize, Y)
```



```
plot(Cars93$Weight, Y)
```



We can note that there seem to be linear relationships between $\frac{100}{CityMPG}$ and MPG.Highway, Weight and Engine size, although our other two variables are a bit more difficult to decipher, so we anticipate that these values may not be present in our final linear model. Using these plots, it seems as though it would be appropriate to use a linear model to describe our response variable using our predictor variables as desired.

Methodology

For our analysis, we will first construct a standard linear model using our data set given variables that we have deemed to be significant through our exploratory analysis. Then, we will use the step() function for our model selection. From this, we will be able to determine the best possible set of predictor variables for our linear model. After generating such a linear model, we can then apply the Box-Cox test to see whether or not we should apply some kind of transformation for our response variable to further improve our model. Once we have our model, we will check our predictor variables for normality using the Wilkes-Shapiro Test and QQ-Plot, and use VIF() and calculate the condition numbers to check for collinearity in our predictor variables to test our assumptions for our linear model to determine whether or not our linear model is appropriate, or if a different linear model would be appropriate. Finally, we will then fit a linear model using the Huber method, Least absolution deviations, and Least trimmed squares to determine if our robust regression techniques deviate greatly from our standard linear model.

Analysis

First, we will construct the linear model with all of the predictor variables in the dataset that we deemed to possibly be significant in our exporatory analysis,

```
HighwayMPG <-Cars93$MPG.highway
Price <- Cars93$Price
EngineSize <-  Cars93$EngineSize
HorsePower <- Cars93$Horsepower
RPM <- Cars93$RPM
REV <- Cars93$Rev.per.mile
Passengers <- Cars93$Passengers
Length <- Cars93$Length
Wheelbase <- Cars93$Wheelbase
Width <- Cars93$Width
TurnCircle <- Cars93$Turn.circle
RearSeatRoom <- Cars93$Rear.seat.room
Luggage <- Cars93$Luggage.room
Weight <- Cars93$Weight
FuelTank <- Cars93$Fuel.tank.capacity
dataFrame <- data.frame(Y, HighwayMPG, Price, EngineSize, HorsePower, RPM, REV, Passengers, Length, Wh
dataFrame <- na.omit(dataFrame)
lmod <- lm(Y ~ HighwayMPG+ Price+ EngineSize+ HorsePower+ RPM+ REV+ Passengers+ Length+ Wheelbase+ Wid
summary(lmod)
```

```
##
## Call:
## lm(formula = Y ~ HighwayMPG + Price + EngineSize + HorsePower +
##     RPM + REV + Passengers + Length + Wheelbase + Width + TurnCircle +
##     RearSeatRoom + Luggage + Weight + FuelTank, data = dataFrame)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.40499 -0.14904  0.00662  0.15346  0.50417
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.609e+00  1.354e+00   4.144 9.92e-05 ***
## HighwayMPG   -9.791e-02  1.031e-02  -9.495 5.79e-14 ***
## Price         9.715e-03  5.198e-03   1.869    0.066 .
## EngineSize    1.844e-01  1.112e-01   1.658    0.102
## HorsePower    1.325e-03  2.107e-03   0.629    0.532
## RPM          -6.738e-05  1.022e-04  -0.659    0.512
## REV          -1.161e-04  1.066e-04  -1.089    0.280
## Passengers   -7.480e-02  6.725e-02  -1.112    0.270
## Length        5.326e-03  5.587e-03   0.953    0.344
## Wheelbase    -1.440e-02  1.366e-02  -1.055    0.295
## Width         2.829e-02  2.241e-02   1.262    0.211
## TurnCircle   -2.536e-03  1.656e-02  -0.153    0.879
## RearSeatRoom  2.285e-02  1.719e-02   1.329    0.188
## Luggage       5.509e-03  1.627e-02   0.339    0.736
## Weight       -1.922e-04  2.558e-04  -0.751    0.455
## FuelTank      3.385e-02  2.282e-02   1.484    0.143
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2375 on 66 degrees of freedom
## Multiple R-squared:  0.9434, Adjusted R-squared:  0.9305
## F-statistic:  73.3 on 15 and 66 DF,  p-value: < 2.2e-16
```

From there, we can now use the step function to determine the optimal linear model,

```
step(lmod)
```

```
## Start:  AIC=-221.58
## Y ~ HighwayMPG + Price + EngineSize + HorsePower + RPM + REV +
##     Passengers + Length + Wheelbase + Width + TurnCircle + RearSeatRoom +
##     Luggage + Weight + FuelTank
##
##                Df Sum of Sq    RSS     AIC
## - TurnCircle    1    0.0013 3.7235 -223.55
## - Luggage       1    0.0065 3.7286 -223.44
## - HorsePower    1    0.0223 3.7444 -223.09
## - RPM           1    0.0245 3.7467 -223.04
## - Weight        1    0.0318 3.7540 -222.88
## - Length        1    0.0513 3.7734 -222.46
## - Wheelbase     1    0.0627 3.7849 -222.21
## - REV           1    0.0669 3.7891 -222.12
## - Passengers    1    0.0698 3.7919 -222.06
## - Width         1    0.0899 3.8120 -221.62
## <none>                     3.7222 -221.58
## - RearSeatRoom  1    0.0997 3.8219 -221.41
## - FuelTank      1    0.1242 3.8463 -220.89
## - EngineSize    1    0.1551 3.8772 -220.23
## - Price         1    0.1970 3.9192 -219.35
## - HighwayMPG    1    5.0847 8.8068 -152.96
##
## Step:  AIC=-223.55
## Y ~ HighwayMPG + Price + EngineSize + HorsePower + RPM + REV +
##     Passengers + Length + Wheelbase + Width + RearSeatRoom +
##     Luggage + Weight + FuelTank
##
##                Df Sum of Sq    RSS     AIC
## - Luggage       1    0.0070 3.7305 -225.39
## - HorsePower    1    0.0214 3.7449 -225.08
## - RPM           1    0.0232 3.7467 -225.04
## - Weight        1    0.0337 3.7572 -224.81
## - Length        1    0.0503 3.7738 -224.45
## - Wheelbase     1    0.0615 3.7850 -224.20
## - REV           1    0.0661 3.7896 -224.11
## - Passengers    1    0.0689 3.7924 -224.05
## - Width         1    0.0890 3.8125 -223.61
## <none>                      3.7235 -223.55
## - RearSeatRoom  1    0.0987 3.8222 -223.40
## - FuelTank      1    0.1254 3.8489 -222.83
## - EngineSize    1    0.1576 3.8811 -222.15
## - Price         1    0.2074 3.9309 -221.10
## - HighwayMPG    1    5.0868 8.8103 -154.93
##
## Step:  AIC=-225.39
## Y ~ HighwayMPG + Price + EngineSize + HorsePower + RPM + REV +
##     Passengers + Length + Wheelbase + Width + RearSeatRoom +
##     Weight + FuelTank
##
##                Df Sum of Sq    RSS     AIC
## - HorsePower    1    0.0188 3.7493 -226.98
## - RPM           1    0.0271 3.7576 -226.80
## - Weight        1    0.0318 3.7623 -226.70
## - Length        1    0.0491 3.7797 -226.32
## - Wheelbase     1    0.0609 3.7914 -226.07
## - Passengers    1    0.0633 3.7938 -226.01
## - REV           1    0.0691 3.7997 -225.89
## <none>                      3.7305 -225.39
## - Width         1    0.0962 3.8267 -225.31
## - RearSeatRoom  1    0.1178 3.8483 -224.84
## - FuelTank      1    0.1520 3.8825 -224.12
## - EngineSize    1    0.1579 3.8884 -224.00
## - Price         1    0.2117 3.9422 -222.87
## - HighwayMPG    1    5.2571 8.9876 -155.29
##
## Step:  AIC=-226.98
## Y ~ HighwayMPG + Price + EngineSize + RPM + REV + Passengers +
##     Length + Wheelbase + Width + RearSeatRoom + Weight + FuelTank
##
##                Df Sum of Sq    RSS     AIC
## - RPM           1    0.0086 3.7579 -228.79
## - Weight        1    0.0157 3.7650 -228.64
## - Length        1    0.0390 3.7883 -228.13
## - REV           1    0.0627 3.8120 -227.62
## - Passengers    1    0.0755 3.8248 -227.35
## - Wheelbase     1    0.0814 3.8307 -227.22
## <none>                      3.7493 -226.98
## - RearSeatRoom  1    0.1101 3.8594 -226.61
## - Width         1    0.1165 3.8658 -226.47
## - FuelTank      1    0.1419 3.8912 -225.94
## - Price         1    0.2808 4.0301 -223.06
## - EngineSize    1    0.3217 4.0710 -222.23
## - HighwayMPG    1    5.2671 9.0163 -157.03
##
## Step:  AIC=-228.79
## Y ~ HighwayMPG + Price + EngineSize + REV + Passengers + Length +
##     Wheelbase + Width + RearSeatRoom + Weight + FuelTank
##
##                Df Sum of Sq    RSS     AIC
## - Weight        1    0.0211 3.7790 -230.34
## - Length        1    0.0418 3.7997 -229.89
## - REV           1    0.0660 3.8239 -229.37
## - Passengers    1    0.0710 3.8289 -229.26
## - Wheelbase     1    0.0794 3.8373 -229.08
## <none>                      3.7579 -228.79
## - RearSeatRoom  1    0.1056 3.8635 -228.52
## - Width         1    0.1168 3.8746 -228.29
## - FuelTank      1    0.1348 3.8927 -227.91
## - Price         1    0.2731 4.0310 -225.04
## - EngineSize    1    0.4609 4.2188 -221.31
## - HighwayMPG    1    5.3467 9.1046 -158.23
##
## Step:  AIC=-230.34
## Y ~ HighwayMPG + Price + EngineSize + REV + Passengers + Length +
##     Wheelbase + Width + RearSeatRoom + FuelTank
##
##                Df Sum of Sq    RSS     AIC
## - Length        1    0.0319 3.8109 -231.65
## - REV           1    0.0545 3.8335 -231.16
## - Passengers    1    0.0609 3.8399 -231.02
## <none>                      3.7790 -230.34
## - Wheelbase     1    0.1027 3.8816 -230.14
## - Width         1    0.1030 3.8820 -230.13
## - RearSeatRoom  1    0.1064 3.8854 -230.06
## - FuelTank      1    0.1139 3.8928 -229.90
## - Price         1    0.2536 4.0326 -227.01
## - EngineSize    1    0.4402 4.2192 -223.30
## - HighwayMPG    1    6.0757 9.8546 -153.74
##
## Step:  AIC=-231.65
## Y ~ HighwayMPG + Price + EngineSize + REV + Passengers + Wheelbase +
##     Width + RearSeatRoom + FuelTank
##
##                Df Sum of Sq     RSS     AIC
## - Passengers    1    0.0478  3.8587 -232.62
## - REV           1    0.0577  3.8686 -232.41
## - Wheelbase     1    0.0730  3.8839 -232.09
## <none>                       3.8109 -231.65
## - RearSeatRoom  1    0.1103  3.9212 -231.31
## - FuelTank      1    0.1342  3.9451 -230.81
## - Width         1    0.1718  3.9827 -230.03
## - Price         1    0.2416  4.0525 -228.61
## - EngineSize    1    0.4322  4.2431 -224.84
## - HighwayMPG    1    6.2329 10.0438 -154.18
##
## Step:  AIC=-232.62
## Y ~ HighwayMPG + Price + EngineSize + REV + Wheelbase + Width +
##     RearSeatRoom + FuelTank
##
##                Df Sum of Sq     RSS     AIC
## - REV           1    0.0601  3.9188 -233.36
## - RearSeatRoom  1    0.0641  3.9227 -233.27
## <none>                       3.8587 -232.62
## - Wheelbase     1    0.1041  3.9627 -232.44
## - Width         1    0.1569  4.0156 -231.36
## - FuelTank      1    0.1585  4.0172 -231.32
## - Price         1    0.3016  4.1603 -228.45
## - EngineSize    1    0.4549  4.3135 -225.49
## - HighwayMPG    1    6.5479 10.4066 -153.27
##
## Step:  AIC=-233.36
## Y ~ HighwayMPG + Price + EngineSize + Wheelbase + Width + RearSeatRoom +
##     FuelTank
##
##                Df Sum of Sq     RSS     AIC
## - RearSeatRoom  1    0.0599  3.9787 -234.11
## - Wheelbase     1    0.0860  4.0047 -233.58
## <none>                       3.9188 -233.36
## - FuelTank      1    0.1223  4.0410 -232.84
## - Width         1    0.1874  4.1061 -231.53
## - Price         1    0.2622  4.1810 -230.05
## - EngineSize    1    0.7408  4.6596 -221.16
## - HighwayMPG    1    7.6505 11.5693 -146.59
##
## Step:  AIC=-234.11
## Y ~ HighwayMPG + Price + EngineSize + Wheelbase + Width + FuelTank
##
##              Df Sum of Sq     RSS     AIC
## - Wheelbase   1    0.0345  4.0132 -235.41
## <none>                     3.9787 -234.11
## - FuelTank    1    0.1208  4.0994 -233.66
## - Width       1    0.1364  4.1150 -233.35
## - Price       1    0.2269  4.2056 -231.56
## - EngineSize  1    0.8443  4.8229 -220.33
## - HighwayMPG  1    7.6048 11.5835 -148.49
##
## Step:  AIC=-235.4
## Y ~ HighwayMPG + Price + EngineSize + Width + FuelTank
##
##              Df Sum of Sq     RSS     AIC
## <none>                     4.0132 -235.41
## - FuelTank    1    0.1000  4.1131 -235.39
## - Width       1    0.1041  4.1173 -235.31
## - Price       1    0.1976  4.2108 -233.46
## - EngineSize  1    0.8113  4.8245 -222.31
## - HighwayMPG  1    8.0807 12.0939 -146.95
```

```
##
## Call:
## lm(formula = Y ~ HighwayMPG + Price + EngineSize + Width + FuelTank,
##     data = dataFrame)
##
## Coefficients:
## (Intercept)   HighwayMPG        Price   EngineSize        Width     FuelTank
##    4.643163    -0.096673     0.007594     0.262639     0.022692     0.025004
```

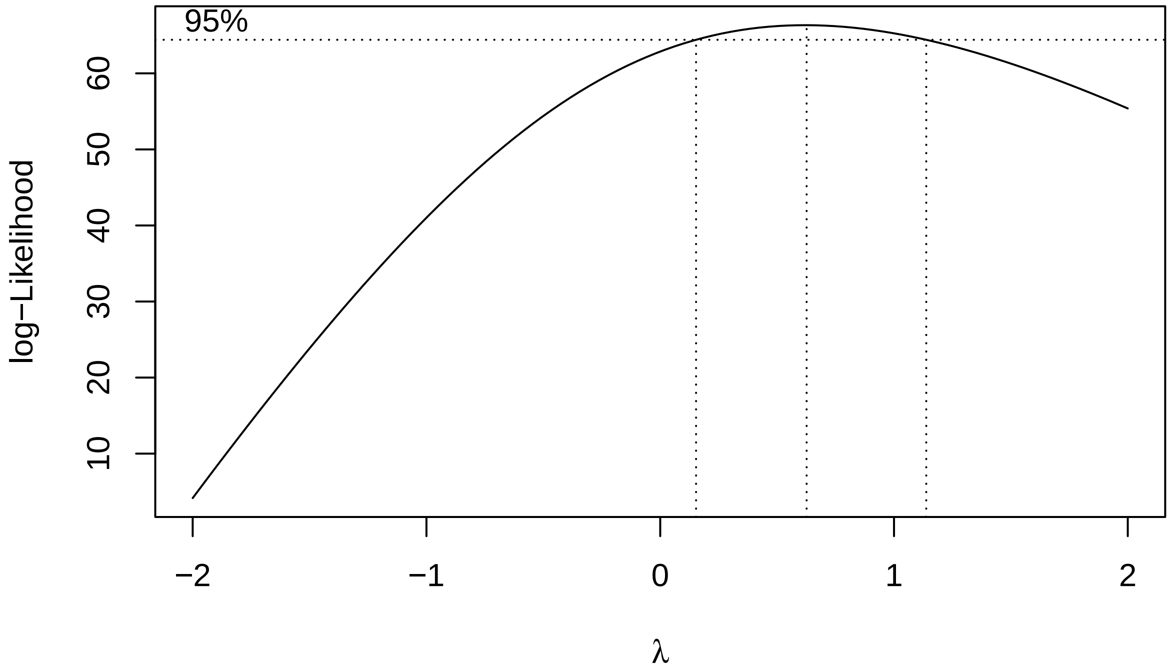
Now, we can see that as a result of our step function, we find that the following linear model is our ideal model which has HighwayMPG, Price, EngineSize, FuelTankCapacity, and Width as our predictor variables. Thus, we can construct such a linear model as follows,

```r
lmod <- lm(formula = Y ~ HighwayMPG + Price + EngineSize + Width + FuelTank, data = dataFrame)
summary(lmod)
```

```
##
## Call:
## lm(formula = Y ~ HighwayMPG + Price + EngineSize + Width + FuelTank,
##     data = dataFrame)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.4395 -0.1804 -0.0123  0.1819  0.4566
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.643163   1.004079   4.624 1.51e-05 ***
## HighwayMPG  -0.096673   0.007815 -12.370  < 2e-16 ***
## Price        0.007594   0.003925   1.935 0.056769 .
## EngineSize   0.262639   0.067006   3.920 0.000193 ***
## Width        0.022692   0.016163   1.404 0.164410
## FuelTank     0.025004   0.018174   1.376 0.172924
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2298 on 76 degrees of freedom
## Multiple R-squared:  0.9389, Adjusted R-squared:  0.9349
## F-statistic: 233.8 on 5 and 76 DF,  p-value: < 2.2e-16
```

Now, looking at these results, we can see that we have a high R^2, as well as a statsitically signficant p-value which make us feel as though we there is some legitimacy to this linear model. We can also note that while our R^2 did go down from our original model, we can see that our adjusted R^2 actually decreased which thus indicates to us that we are better off with fewer predictor variables since we our calculations indicated that we did not need so many variables. Now, we can implement our Box-Cox test to determine whether or not we should apply a transformation to our predictor variable.

```r
bc <- boxcox(lmod)
```



```
bc$x[which.max(bc$y)]
```

```
## [1] 0.6262626
```

From this, we can see that a transformation of $x^{\frac{2}{3}}$ would be an ideal tranformation (rounding to a clean rational number). Thus applying that, we obtain, that,

```
lmodBoxCox <- lm(Y^(2/3) ~ HighwayMPG + Price + EngineSize + Width + FuelTank, data = dataFrame)
summary(lmodBoxCox)
```

```
##
## Call:
## lm(formula = Y^(2/3) ~ HighwayMPG + Price + EngineSize + Width +
##     FuelTank, data = dataFrame)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.167765 -0.074210 -0.007431  0.070430  0.186944
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.835534   0.402076   7.052    7e-10 ***
## HighwayMPG  -0.042771   0.003129 -13.667  < 2e-16 ***
## Price        0.002471   0.001572   1.572 0.120097
## EngineSize   0.092179   0.026832   3.435 0.000962 ***
## Width        0.010282   0.006472   1.589 0.116319
## FuelTank     0.011185   0.007278   1.537 0.128482
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09202 on 76 degrees of freedom
## Multiple R-squared:  0.9421, Adjusted R-squared:  0.9383
## F-statistic: 247.4 on 5 and 76 DF,  p-value: < 2.2e-16
```

Which we can makes Width and FuelTank more significant, but Price less signficant. We can also see that both our adjusted R^2 and our regular R^2 both increased which gives further legitimacy to the transformation that we made. Looking at our parameters, we can see that this model gives us a linear model of the form of $Y = 2.835 + -0.0428x_1 + 0.00247x_2 + 0.0922x_3 + 0.0103x_4 + 0.01185x_5$, which indicates to us that if all other variables were held constant, if we have a 1MPG increase in the HighwayMPG, we would expect to be able to consume 0.0428 gallons less of gasoline over 100 miles within the city, that if we pay one thousand dollars more for our car, we would expect to consume 0.002471 gallons more of gas per 100 miles in the city, if we have an 1 Liter increase in the engine size, we can expect to consume 0.0922 gallons more of gas per 100 miles in the city, if we have a 1 inch increase in the width of our car, then we would expect to consume 0.0103 gallons more of gas per 100 miles in the city, and if we have an increase in the capacity of our fuel tank of 1 gallon, we would expect to have an increase of 0.0112 gallons of gas consumed per 100 miles in the city.

Evaluation of Model Now that we have produced our model, we can now perform our regression diagnostics. We will first apply the Wilkes-Shapiro test on each of our predictor variables,

```
?shapiro.test
shapiro.test(HighwayMPG)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  HighwayMPG
## W = 0.92443, p-value = 4.576e-05
```

```
shapiro.test(Price)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Price
## W = 0.88051, p-value = 4.235e-07
```

```
shapiro.test(EngineSize)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  EngineSize
## W = 0.9361, p-value = 0.000199
```

```
shapiro.test(Width)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Width
## W = 0.97154, p-value = 0.0397
```

```
shapiro.test(FuelTank)
```

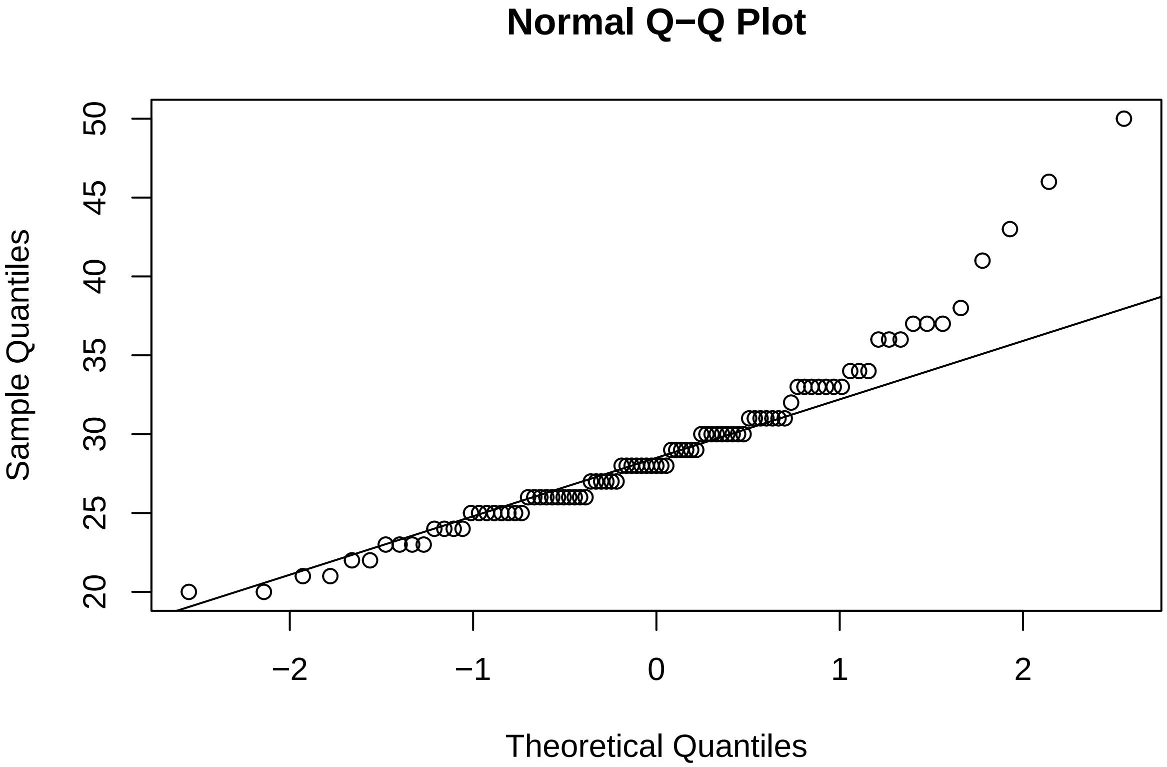
```
##
##  Shapiro-Wilk normality test
##
## data:  FuelTank
## W = 0.98341, p-value = 0.287
```

```
shapiro.test(Y)
```

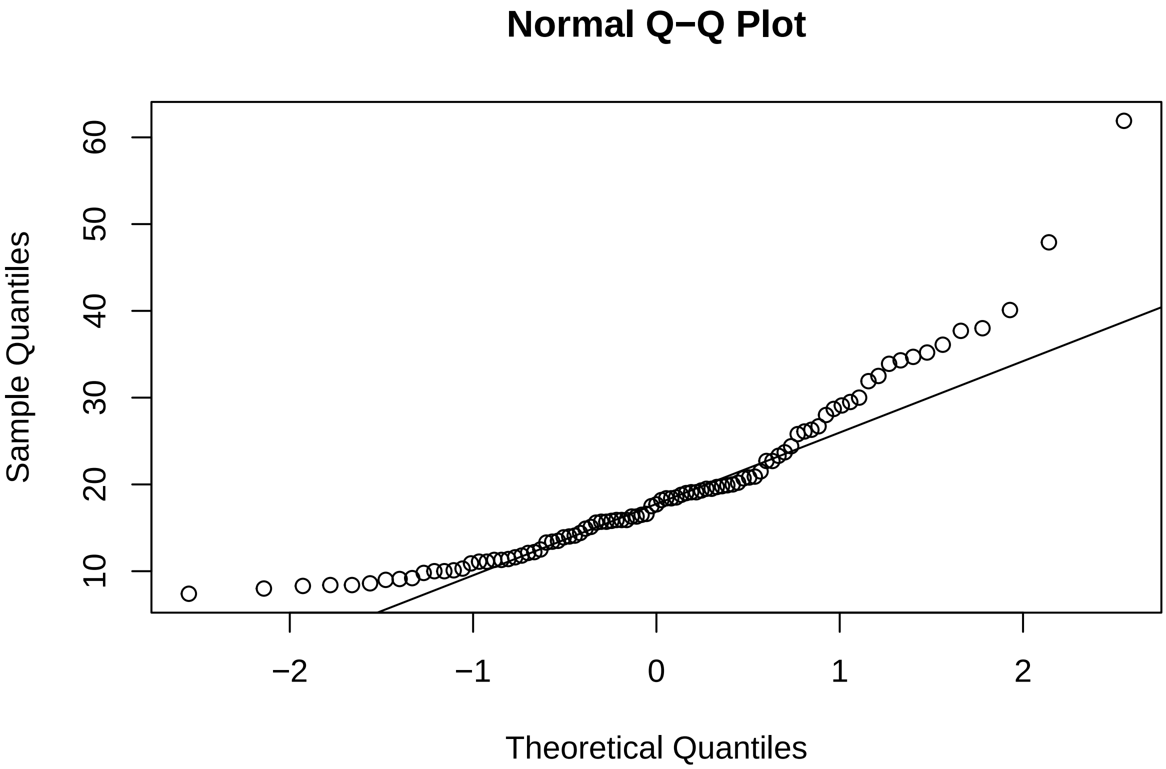
```
##
##  Shapiro-Wilk normality test
##
## data:  Y
## W = 0.97974, p-value = 0.1579
```

Looking at our results, we can see that for a value of $\alpha = 0.05$, the only one of our predictor variables that does not differ from the normal distribution in a statstically significant way is our FuelTank predictor. Similarly, our response variable does meeting our assumptions of normality. Visually, we can see from the qqplots of our non-normally distributed predictor variables that,

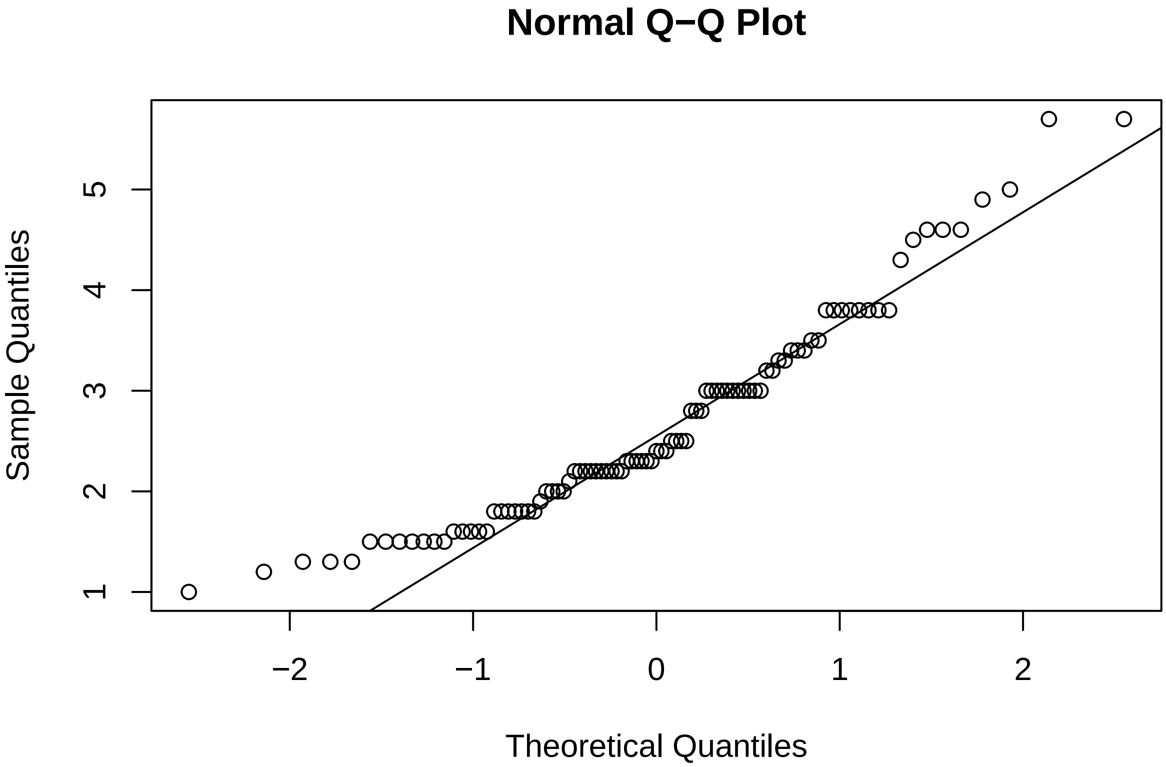
```
qqnorm(HighwayMPG)
qqline(HighwayMPG)
```



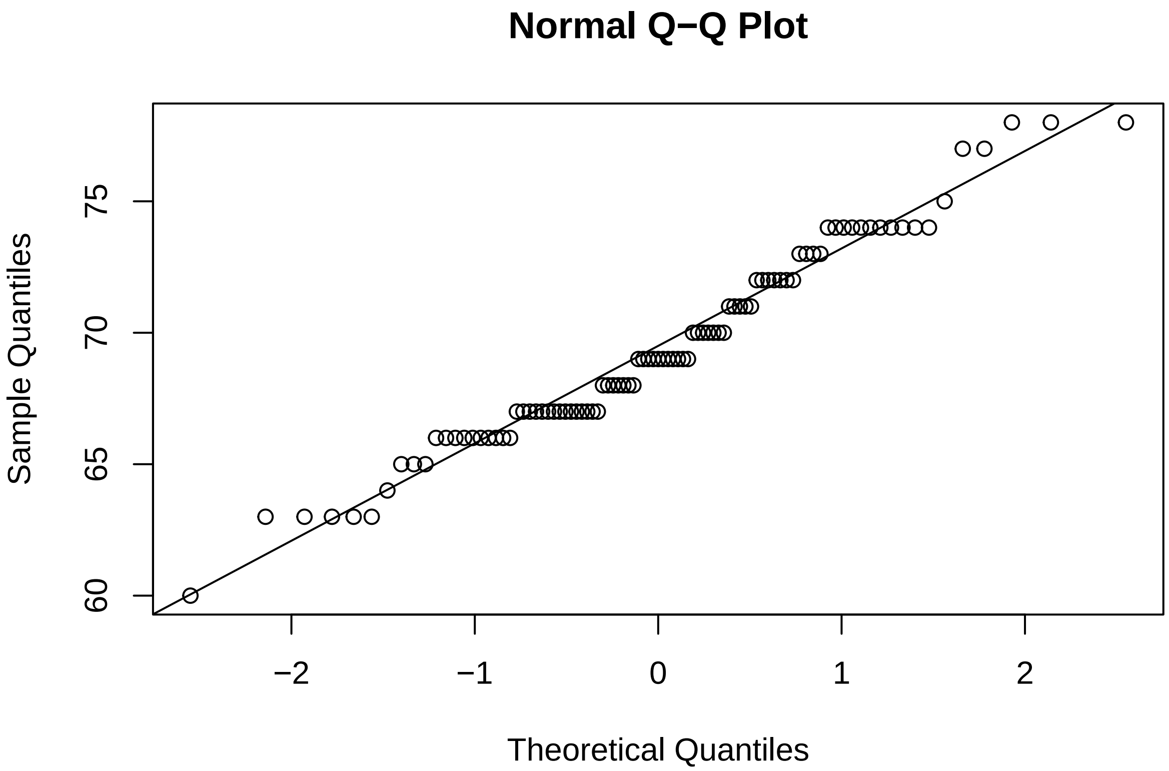
```
qqnorm(Price)
qqline(Price)
```



```
qqnorm(EngineSize)
qqline(EngineSize)
```



```
qqnorm(Width)
qqline(Width)
```



We can see that these qqplots corroborate with our worries about the normality of our predictor variables. Thus, we can see that our assumption of normality may not be met for our linear model. Now, looking for leverage points, we can conduct our search for these points as follows,

```
hatvalues(lmodBoxCox) > 2*mean(hatvalues(lmodBoxCox))
```

```
##     1     2     3     4     5     6     7     8     9    10    11    12    13
## FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE
##    14    15    18    20    21    22    23    24    25    27    28    29    30
## FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    31    32    33    34    35    37    38    39    40    41    42    43    44
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE
##    45    46    47    48    49    50    51    52    53    54    55    58    59
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
##    60    61    62    63    64    65    67    68    69    71    72    73    74
##  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    75    76    77    78    79    80    81    82    83    84    85    86    88
## FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    90    91    92    93
## FALSE FALSE FALSE FALSE
```

Thus, we can see that we find high leverage points at indices 5, 8, 18, 39, 42, 59, 60, and 80. Now, checking for outliers, we find that,

```
rstandard(lmodBoxCox)[abs(rstandard(lmodBoxCox))>2]
```

```
##       58
## 2.125537
```

Which gives us our point 58 as an outlier. Finally, we can look for highly influential points using the following code,

```
cooks.distance(lmodBoxCox)>4/length(cooks.distance(lmodBoxCox))
```

```
##     1     2     3     4     5     6     7     8     9    10    11    12    13 
## FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE 
##    14    15    18    20    21    22    23    24    25    27    28    29    30 
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE 
##    31    32    33    34    35    37    38    39    40    41    42    43    44 
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE 
##    45    46    47    48    49    50    51    52    53    54    55    58    59 
## FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE 
##    60    61    62    63    64    65    67    68    69    71    72    73    74 
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE 
##    75    76    77    78    79    80    81    82    83    84    85    86    88 
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE 
##    90    91    92    93 
## FALSE  TRUE FALSE FALSE
```

Which we can see gives us points 5, 10, 39, 48, 58, 59, and 91 all as highly influential points. Thus, we can now refit our model removing point 58 and then consider our resulting model as follows,

```
removedDataFrame <- dataFrame[-c(58), ]
lmodRemovedPoint <- lm(Y^(2/3) ~ HighwayMPG + Price + EngineSize + Width + FuelTank, data = dataFrame
summary(lmodRemovedPoint)
```

```
## 
## Call:
## lm(formula = Y^(2/3) ~ HighwayMPG + Price + EngineSize + Width + 
##     FuelTank, data = dataFrame)
## 
## Residuals:
##       Min        1Q    Median        3Q       Max 
## -0.167765 -0.074210 -0.007431  0.070430  0.186944 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  2.835534   0.402076   7.052    7e-10 ***
## HighwayMPG  -0.042771   0.003129 -13.667  < 2e-16 ***
## Price        0.002471   0.001572   1.572 0.120097    
## EngineSize   0.092179   0.026832   3.435 0.000962 ***
## Width        0.010282   0.006472   1.589 0.116319    
## FuelTank     0.011185   0.007278   1.537 0.128482    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.09202 on 76 degrees of freedom
## Multiple R-squared:  0.9421, Adjusted R-squared:  0.9383 
## F-statistic: 247.4 on 5 and 76 DF,  p-value: < 2.2e-16
```

Which we can see does not do anything to the estimated coefficients, nor to our R^2 values or F-statistic. Thus, we can conclude that our analysis does not change really when we remove this outlier point. Now, checking for collinearity with the VIFs and condition numbers for our model, we find that,

```
X <- model.matrix(lmodBoxCox)[,-1]
e <- eigen(t(X)%*%X)
sqrt(e$val[1]/e$val)
```

```
## [1]   1.000000   7.616721  19.190843  59.224532 152.887606
```

Looking at these values, we can see that there appears to be a high level of collinearity within our set of predictor variables as indicated by our large condition numbers. We can now check the VIF as well,

```
require(faraway)
```

```
## Loading required package: faraway
```

```
vif(X)
```

```
## HighwayMPG      Price EngineSize      Width   FuelTank 
##   2.352338   2.344298   6.943287   5.468079   4.591072
```

Specifically, we can see that there appears to be a high level of collinearity specifically wiht our EngineSize, Width, and FuelTank variables. We can check for this as follows,

```
cor(EngineSize, Width)
```

```
## [1] 0.8671102
```

```
cor(EngineSize, FuelTank)
```

```
## [1] 0.7593062
```

```
cor(FuelTank, Width)
```

```
## [1] 0.798719
```

Which we can see an especially high level of correlation between these three variables. Finally, to quell some of these issues, we can now fit a model using the Huber method, least absolute deviations, and the least trimmed squares methods. Computing those here,

```
huberMethod <- rlm(Y^(2/3) ~ HighwayMPG + Price + EngineSize + Width + FuelTank, data = dataFrame)
require(quantreg)
```

```
## Loading required package: quantreg

## Loading required package: SparseM

## 
## Attaching package: 'SparseM'

## The following object is masked from 'package:base':
## 
##     backsolve
```

```
l1Mod <- rq(Y^(2/3) ~ HighwayMPG + Price + EngineSize + Width + FuelTank, data = dataFrame)
ltsmod <- ltsreg(Y^(2/3) ~ HighwayMPG + Price + EngineSize + Width + FuelTank, data = dataFrame)
print("huber method")
```

```
## [1] "huber method"
```

```
summary(huberMethod)
```

```
## 
## Call: rlm(formula = Y^(2/3) ~ HighwayMPG + Price + EngineSize + Width + 
##     FuelTank, data = dataFrame)
## Residuals:
##       Min        1Q    Median        3Q       Max 
## -0.168245 -0.073778 -0.006493  0.071143  0.190059 
## 
## Coefficients:
```

```
##             Value    Std. Error t value 
## (Intercept)   2.8329   0.4253     6.6605
## HighwayMPG   -0.0427   0.0033   -12.8841
## Price         0.0023   0.0017     1.3982
## EngineSize    0.0934   0.0284     3.2901
## Width         0.0102   0.0068     1.4838
## FuelTank      0.0116   0.0077     1.5083
## 
## Residual standard error: 0.1101 on 76 degrees of freedom
```

```
print("L1Mod")
```

```
## [1] "L1Mod"
```

```
summary(l1Mod)
```

```
## 
## Call: rq(formula = Y^(2/3) ~ HighwayMPG + Price + EngineSize + Width + 
##     FuelTank, data = dataFrame)
## 
## tau: [1] 0.5
## 
## Coefficients:
##             coefficients lower bd upper bd
## (Intercept)  2.43529      1.09945  3.68473
## HighwayMPG  -0.04183     -0.05162 -0.03584
## Price        0.00397     -0.00079  0.00982
## EngineSize   0.07985      0.00972  0.14620
## Width        0.01721     -0.00489  0.03638
## FuelTank     0.00487     -0.00916  0.03706
```

```
print("ltsMod")
```

```
## [1] "ltsMod"
```

```
coef(ltsmod)
```

```
## (Intercept)  HighwayMPG       Price  EngineSize       Width    FuelTank 
##  1.46122900 -0.04475214  0.01088246 -0.02462338  0.04262178 -0.02850880
```

Looking first at the Huber method, we can see that our results were pretty similar withour original linear model, as indicated by the similar coefficients and approximately similar t-values calculated for each of our predictor variables. We can see that this corroborates well also with our least absolute deviations model and our original model, as each of calculated coefficients for our original linear model lies within the bounds of our LAD linear model. However, we can note some discrepencies between our least trimmed squares method, specifically with the discrpencies between the coefficients for our intercept, Price, EngineSize, and Width.

Conclusion

From our analysis, we were able to fit an ordinary linear model explaining our desired response variable, the number of gallons that a car consumes to cover 100 miles within the city. Using the step function, we found that the most relevant predictive variables were HighwayMPG, Price, EngineSize, Width, and FuelTank Capacity. However, in our evaluation of the model, we did find that some of our assumptions for our linear model, such as normality, were not met by all of our variables which casts doubt into whether or not we can use our linear model. However, we can note that when we applied our robust regression techniques, we generally found consensus between our ordinary linear model and our robust techniques (Huber and LAD). Therefore, there is some indication that we can trust our linear model, or that we can simply use our other robust techniques, as we did find that the outlier points did not affect our model too intensely as indicated by the fact that when we removed the point, we did not have much change to our model. It is important to be weary of this model however and further investigation is likely necessary due to our the failure of normality for our predictor variables which is a key assumption for utilizing a linear model. It is also interesting to consider that a lot of the variables that we initially believed to likely be interesting and important to our analysis were not such (e.g. Weight and Type), and that the EngineSize was one of the most important predicting variables when it came to predicting the fuel efficiency.