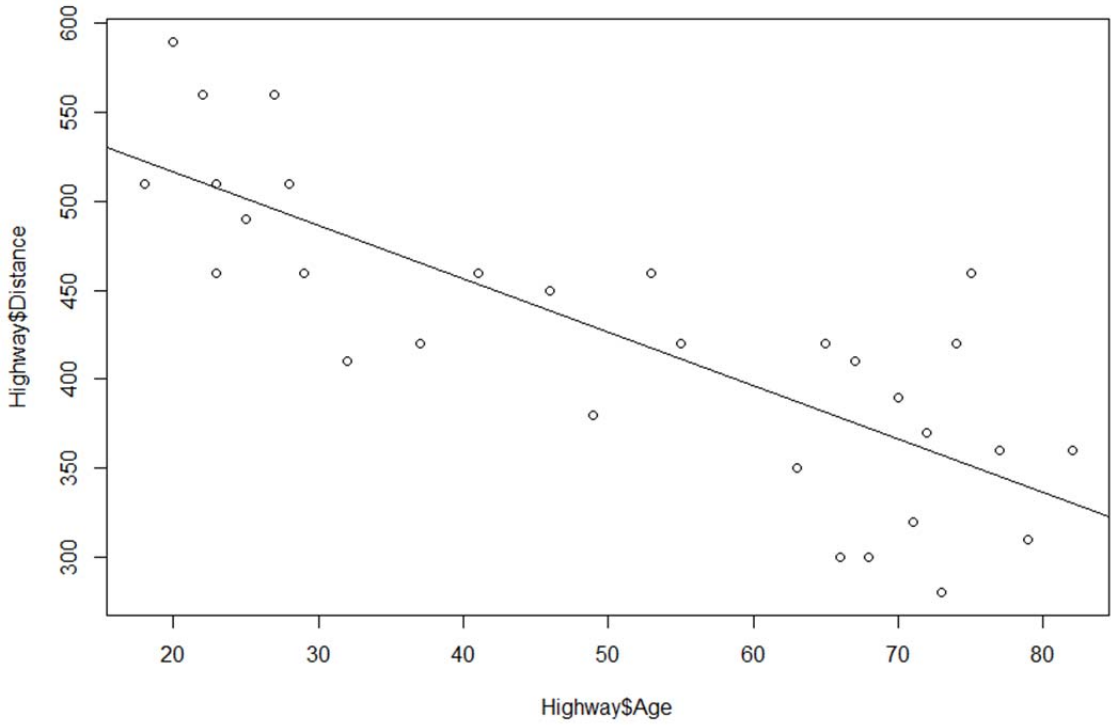
R Commands for Highway Sign Data to check conditions

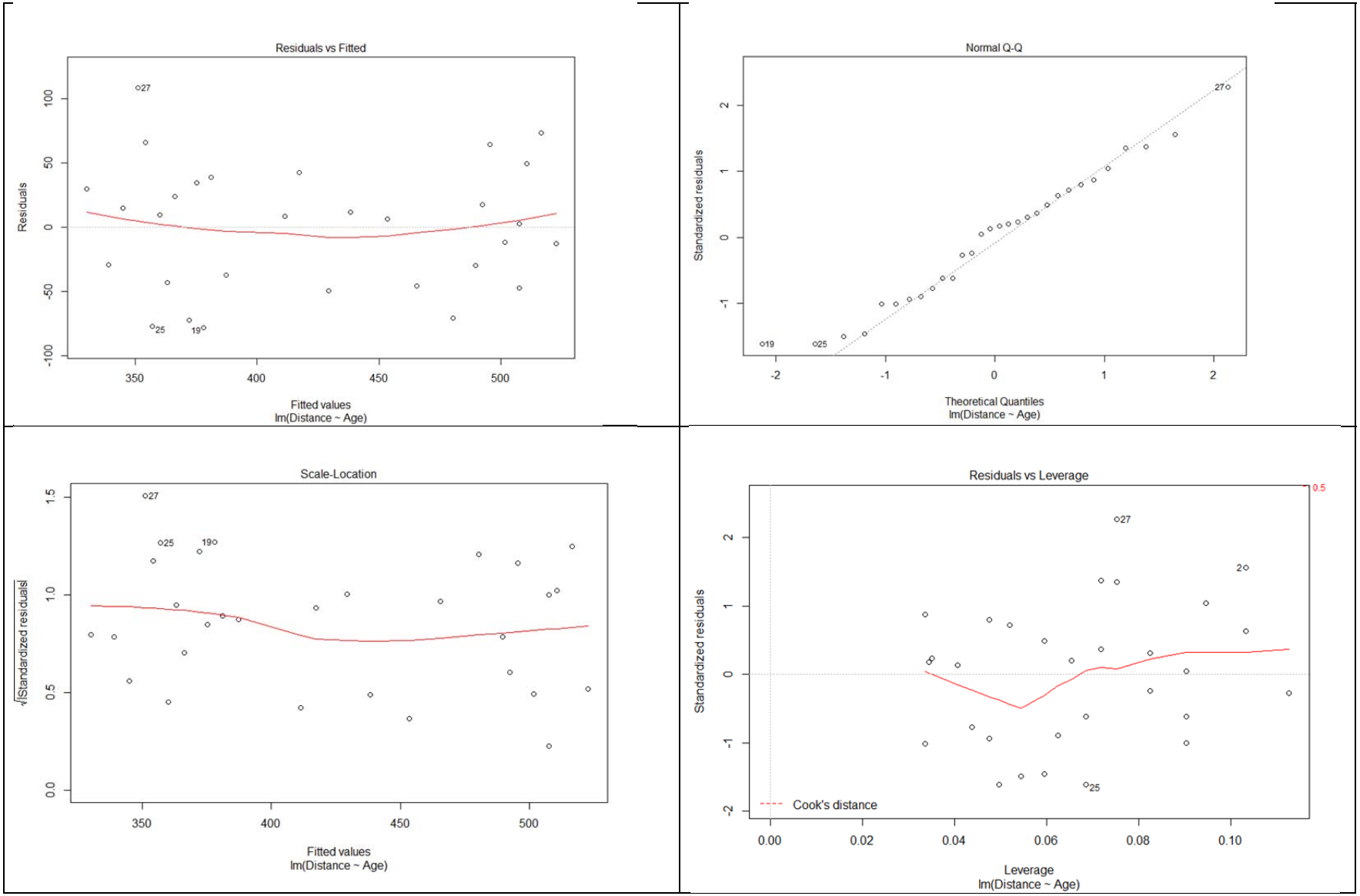
NOTE: This is the R session used to find the regression equation, and some plots for the Highway sign data. Download the file HighwaySign.txt from the webpage.

```
> #Read in the data.
> #sep="\t" shows that the columns are separated with a tab.
> #header=F says there is no beginning line with variable names.
> #col.names provides names for the two columns.
> Highway<-read.table("HighwaySign.txt", header=F, sep="\t", col.names=c("Age","Distance"))

> #Make sure it worked by printing out first 6 lines:
> head(Highway)
  Age Distance
1  18      510
2  20      590
3  22      560
4  23      510
5  23      460
6  25      490
> #Create the regression model. Call it "HWModel"
> HWModel<-lm(Distance~Age,data=Highway)
> #See a summary of the model, including coefficients, etc.
> summary(HWModel)
Call:
lm(formula = Distance ~ Age, data = Highway)

Residuals:
    Min      1Q  Median      3Q     Max
-78.231 -41.710   7.646  33.552 108.831

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 576.6819    23.4709  24.570  < 2e-16 ***
Age          -3.0068     0.4243  -7.086 1.04e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 49.76 on 28 degrees of freedom
Multiple R-squared: 0.642,      Adjusted R-squared: 0.6292
F-statistic: 50.21 on 1 and 28 DF,  p-value: 1.041e-07

> #Create semi-studentized residuals
> Highway$StandardResids <- rstandard(HWModel)
```

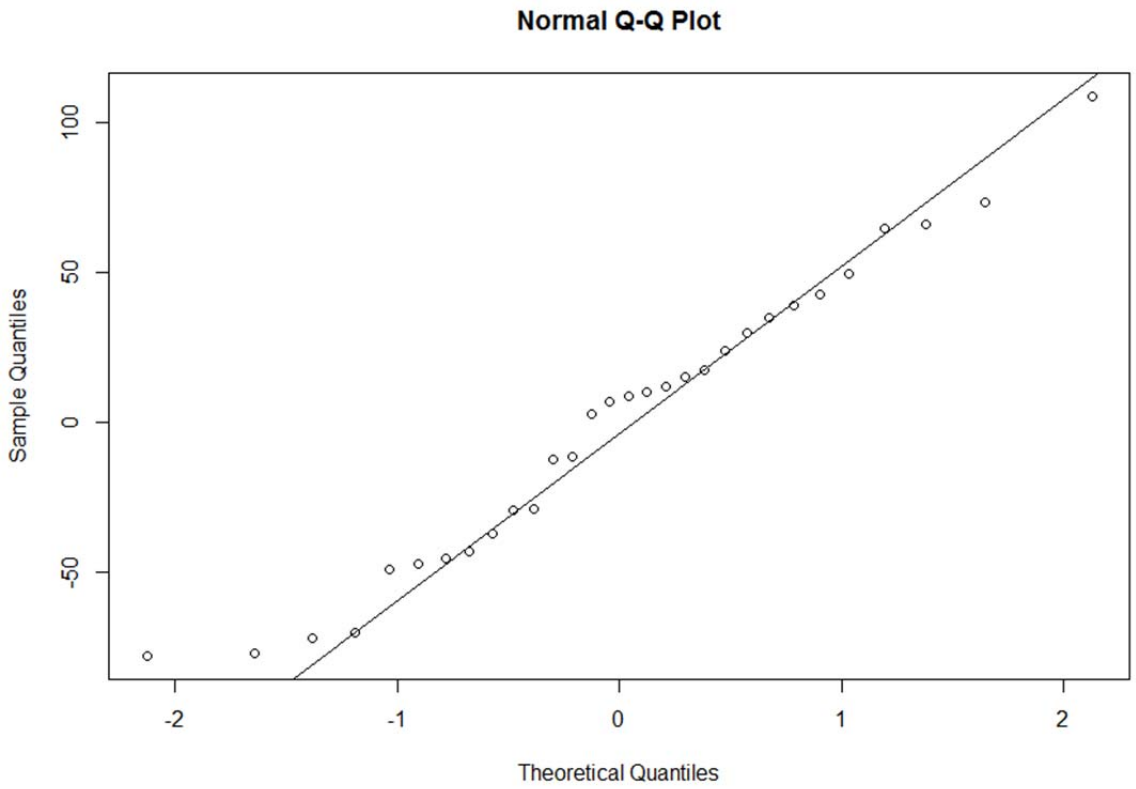
```
> #Plot Age versus distance; add least squares line
> plot(Highway$Age,Highway$Distance)
> abline(HWModel)
```




```
> #Get four residual plots all together (“leverage” explained later in course)
> plot(HWModel)
```


Residuals vs Fitted
Residuals
Fitted values
lm(Distance ~ Age)
Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(Distance ~ Age)
Scale-Location
√|Standardized residuals|
Fitted values
lm(Distance ~ Age)
Residuals vs Leverage
Standardized residuals
Cook's distance
Leverage
lm(Distance ~ Age)

```
> #Get a normal probability plot of the residuals and add a line
> qqnorm(HWModel$resid)
> qqline(HWModel$resid)
```



Transformations in R:

If you want to **transform** the response variable *Y* into some new variable *Y* ', you can add a new column to the data table consisting of the new variable.
For the data table named *Data*, to square the response variable *GPA* and add it to the data table, type:

```
> Data <- cbind(Data, Data$GPA^2)
```

To take its *square root*, type:

```
> Data <- cbind(Data, sqrt(Data$GPA) )
```

To take its *natural logarithm*, type:

```
> Data <- cbind(Data, log(Data$GPA) )
```

To take its *common logarithm (base 10),* type:

```
> Data <- cbind(Data, log10(Data$GPA) )
```

To take its *reciprocal*, type:

```
> Data <- cbind(Data, 1/Data$GPA )
```

To take its *reciprocal square root*, type:

```
> Data <- cbind(Data, 1/sqrt(Data$GPA) )
```

And so on. You will want to give the new column in the data table an appropriate name. You can then run a linear model using the transformed response variable and the original predictor.