Simple Linear and Multiple Regression

2023-09-01

```
#Importing Data
library(ISLR2)
```

```
## Warning: package 'ISLR2' was built under R version 4.3.1
```

```
head(Auto)
```

```
##   mpg cylinders displacement horsepower weight acceleration year origin
## 1  18         8          307        130   3504         12.0   70      1
## 2  15         8          350        165   3693         11.5   70      1
## 3  18         8          318        150   3436         11.0   70      1
## 4  16         8          304        150   3433         12.0   70      1
## 5  17         8          302        140   3449         10.5   70      1
## 6  15         8          429        198   4341         10.0   70      1
##                       name
## 1 chevrolet chevelle malibu
## 2         buick skylark 320
## 3        plymouth satellite
## 4             amc rebel sst
## 5               ford torino
## 6          ford galaxie 500
```

```
summary(Auto)
```

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

```r
# Simple Linear Regression
attach(Auto)
lm_model<-lm(mpg ~ horsepower)
summary(lm_model)
```

```
##
## Call:
## lm(formula = mpg ~ horsepower)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5710  -3.2592  -0.3435   2.7630  16.9240
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 39.935861   0.717499   55.66   <2e-16 ***
## horsepower  -0.157845   0.006446  -24.49   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.906 on 390 degrees of freedom
## Multiple R-squared:  0.6059, Adjusted R-squared:  0.6049
## F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

Review of Simple Linear Regression

1) Looking at the summary from this analysis, it seems that horsepower has a negative, but significant impact on MPG.
2) The relationship has some strong elements, such as a low p value, high f-statistic and T score (albeit negative). The std. error is quite low, which is great. The R-square value is relatively strong, but there is room for improvement in terms of accounting for variance.
3) The effect of the relationship is negative.

```r
# Confidence intervals for model and prediction
predict(lm_model, data.frame(horsepower = (c(98))), interval = "confidence")
```

```
##        fit      lwr      upr
## 1 24.46708 23.97308 24.96108
```

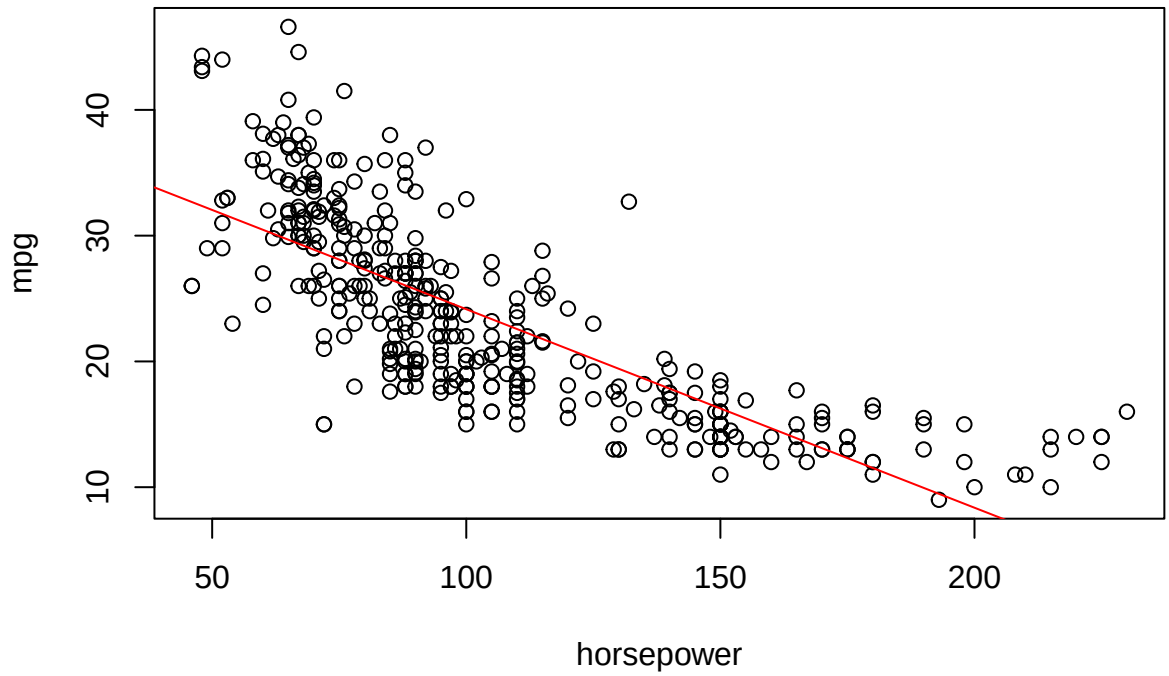
```r
predict(lm_model, data.frame(horsepower = (c(98))), interval = "prediction")
```

```
##        fit     lwr      upr
## 1 24.46708 14.8094 34.12476
```

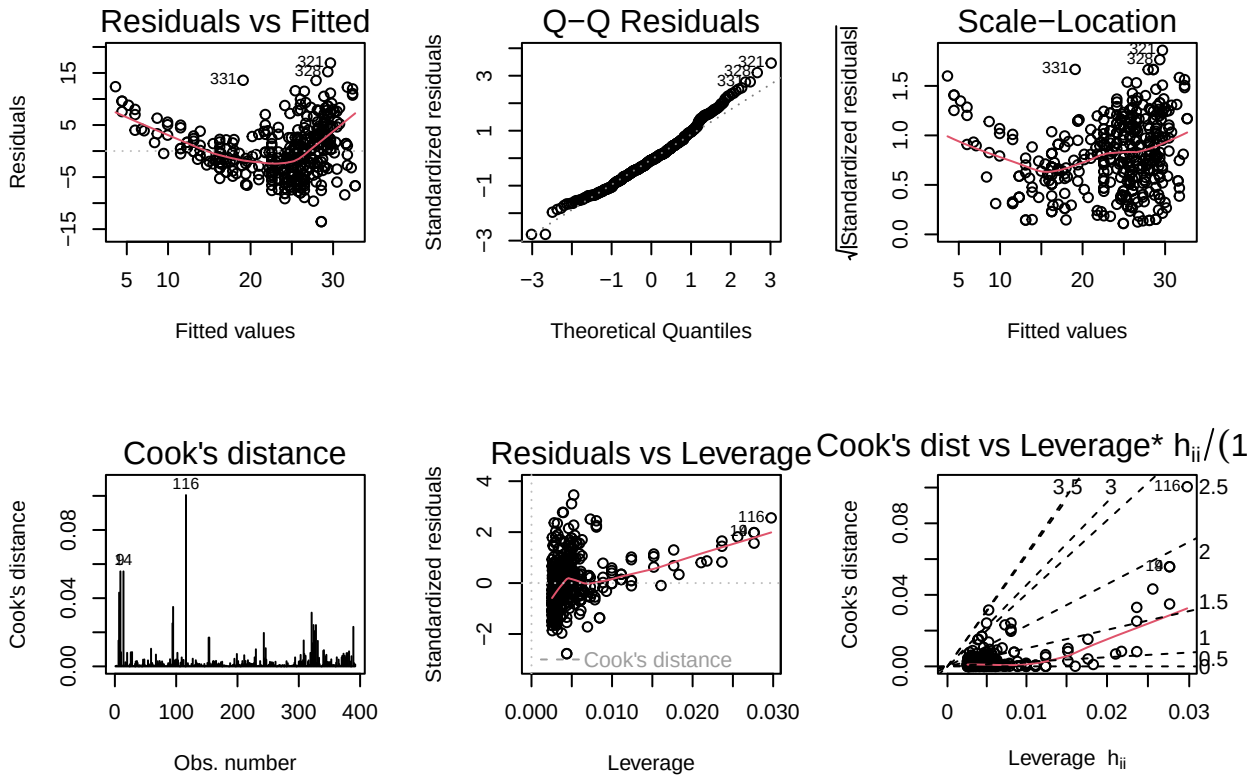
Prediction value at 98

24.46708

```r
# plotting simple linear regression
plot(horsepower, mpg)
abline(lm_model, col = "red")
```

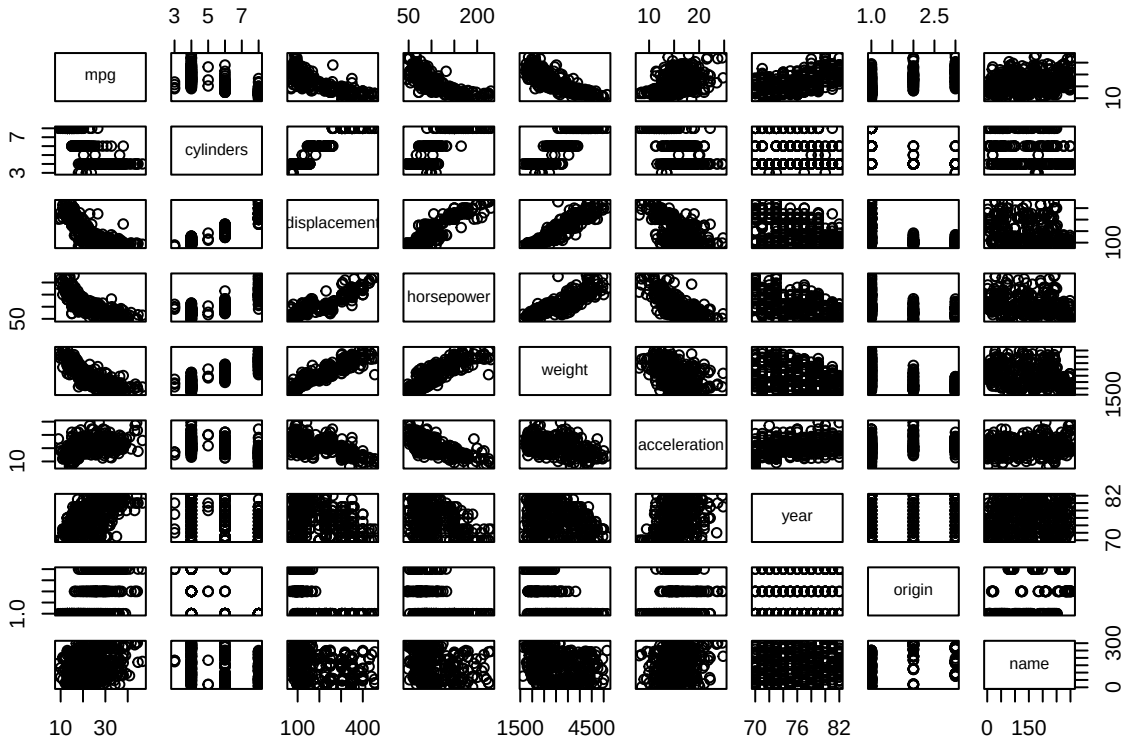



```r
par(mfrow = c(2, 3)); plot(lm_model, which = 1:6)
```



Diagnostics of simple linear regression 1) Residuals vs fitted shows heteroscedasticity, when it should show homoscedasticity. This violates the assumption of linearity. 2) The Q-Q residuals shows the points mostly falling close along the line, suggesting normality 3) The scale-location plot repeats a pattern of heteroscedasticity, violating the assumption of constant variance. 4) Cooks distance indicates observations 9, 14, and 116 as having an influence as outliers. 5) Reviewing

```
# create correlation matrix of variables in Auto
pairs(Auto)
```



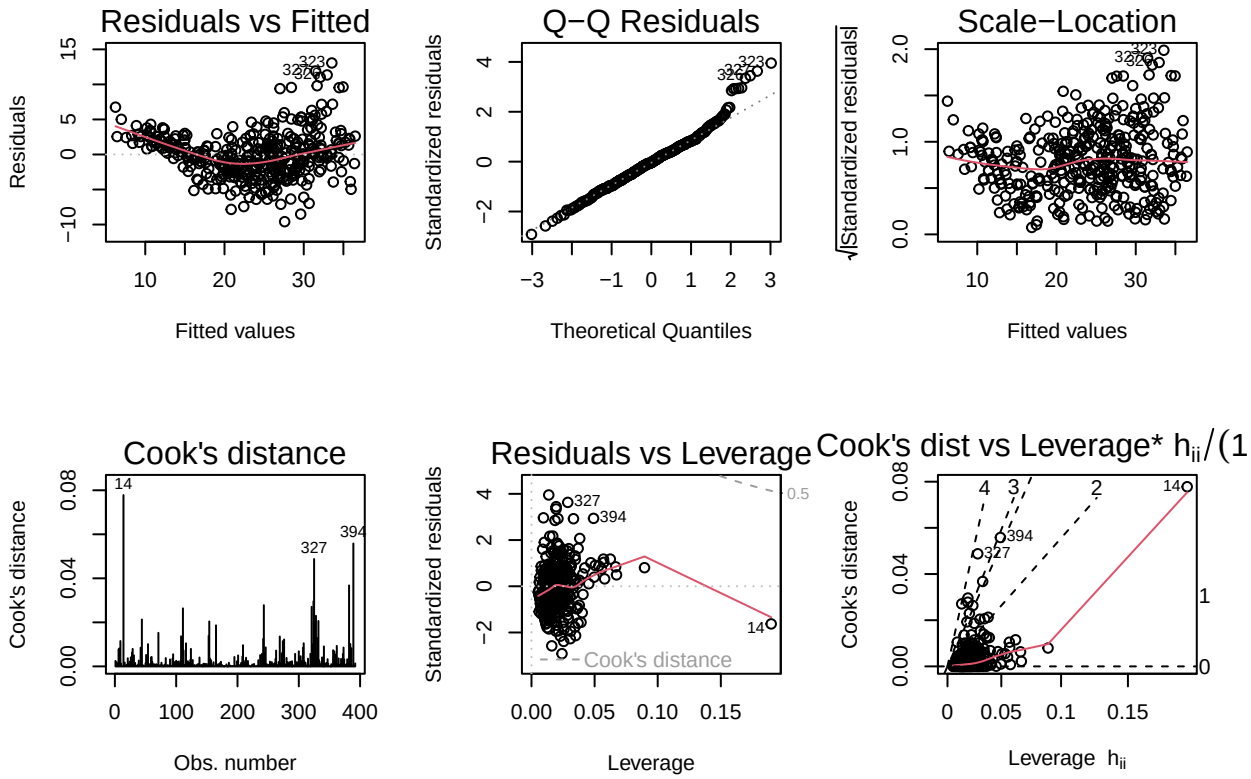

```
data_sub<-Auto[, -which(names(Auto) == "name")]
correlation_matrix<-cor(data_sub)
print(correlation_matrix)
```

```
##                     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
```

```
# create multiple linear regression
multi_lm<-lm(mpg ~. -name, data = Auto)
summary(multi_lm)
```

```
##
## Call:
## lm(formula = mpg ~ . - name, data = Auto)
##
## 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
```

```
par(mfrow = c(2, 3)); plot(multi_lm, which = 1:6)
```



Multi Linear Regression Analysis

The output from this analysis is interesting, showing that factors of displacement, weight, year and origin being candidates for significant predictors of MPG (some indicating a negative relationship). I think this makes sense, since heavier cars may have lower mpg, and newer cars may indeed have a higher mpg. The coefficient for the year indicates that for every one unit increase in year, the dependent variable increases by .75 units.

The diagnostic plots show some overlapping errors with the simple linear analysis, with the residuals vs fitting plot showing heteroscedasticity. The Q-Q residuals generally follows a linear path, but variance starts to emerge on this chart. The scale-location graph also shows heteroscadasticity.Cooks distace indicates some outlier potential for observations 14, 327, and 394. The final charts also indicate observation 14, especially, as having leverage as an outlier.

Some Further Models

The correlation charts show that there is some relationship between variables such as horsepower, weight, and displacement, most notably. Some of the other variables show some possible correlation, but not as strong. Further, the display from simple linear regression between mpg and horsepower seems it may be better fitted with a quadratic model. So, time to experiment.

```
# creating models to compare with interaction terms
lm1<- lm(mpg ~ year + origin + acceleration + displacement * horsepower, data = Auto)
summary(lm1)
```

```
##
## Call:
## lm(formula = mpg ~ year + origin + acceleration + displacement *
##     horsepower, data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5506 -1.7454 -0.2343  1.4139 13.5583
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              5.164e+00  4.229e+00   1.221  0.22281
## year                     7.015e-01  4.559e-02  15.386  < 2e-16 ***
## origin                   7.441e-01  2.570e-01   2.895  0.00401 **
## acceleration            -4.476e-01  7.858e-02  -5.696 2.44e-08 ***
## displacement            -8.930e-02  5.961e-03 -14.981  < 2e-16 ***
## horsepower              -2.502e-01  1.627e-02 -15.376  < 2e-16 ***
## displacement:horsepower  5.966e-04  4.092e-05  14.581  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.015 on 385 degrees of freedom
## Multiple R-squared:  0.8531, Adjusted R-squared:  0.8508
## F-statistic: 372.6 on 6 and 385 DF,  p-value: < 2.2e-16
```

```
lm2<-lm(mpg ~ year + origin + displacement:horsepower, data = Auto)
summary(lm2)
```

```
##
## Call:
## lm(formula = mpg ~ year + origin + displacement:horsepower, data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.1561  -2.7148  -0.4351   2.3139  13.7558
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -3.087e+01  5.058e+00  -6.103 2.52e-09 ***
## year                      7.162e-01  6.427e-02  11.145  < 2e-16 ***
## origin                    2.546e+00  3.108e-01   8.193 3.74e-15 ***
## displacement:horsepower -1.721e-04  1.243e-05 -13.843  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.273 on 388 degrees of freedom
## Multiple R-squared:  0.7026, Adjusted R-squared:  0.7003
## F-statistic: 305.5 on 3 and 388 DF,  p-value: < 2.2e-16
```

```
lm4<-lm(mpg ~ year + origin + horsepower * weight, data = Auto)
summary(lm4)
```

```
##
```

```
## Call:
## lm(formula = mpg ~ year + origin + horsepower * weight, data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.6051 -1.7722 -0.1304  1.5205 12.0369
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        8.145e-01  3.969e+00   0.205  0.83753
## year               7.677e-01  4.464e-02  17.195  < 2e-16 ***
## origin             7.224e-01  2.328e-01   3.103  0.00206 **
## horsepower        -2.160e-01  2.055e-02 -10.514  < 2e-16 ***
## weight            -1.106e-02  6.343e-04 -17.435  < 2e-16 ***
## horsepower:weight  5.501e-05  5.051e-06  10.891  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.931 on 386 degrees of freedom
## Multiple R-squared:  0.8608, Adjusted R-squared:  0.859
## F-statistic: 477.5 on 5 and 386 DF,  p-value: < 2.2e-16
```

```
lm6<-lm(mpg ~ year + origin + horsepower:weight, data = Auto)
summary(lm6)
```

```
##
## Call:
## lm(formula = mpg ~ year + origin + horsepower:weight, data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5528 -2.4873 -0.3992  2.1518 13.2096
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -2.579e+01  4.638e+00  -5.560 5.03e-08 ***
## year               6.856e-01  5.846e-02  11.728  < 2e-16 ***
## origin             2.320e+00  2.824e-01   8.218 3.14e-15 ***
## horsepower:weight -1.922e-05  1.105e-06 -17.400  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.914 on 388 degrees of freedom
## Multiple R-squared:  0.7504, Adjusted R-squared:  0.7485
## F-statistic: 388.9 on 3 and 388 DF,  p-value: < 2.2e-16
```

Interaction Outcomes

The best fitting model seems to include the multiplicative interaction between horsepower and weight, which has a relatively high F statistic and R-Squared value as compared to other models. The correlation plot shows some interesting trends between acceleration, weight, horsepower, and displacement. Further analyses might benefit from examining the relationship between these variables more closely.

Fitting and Modeling Quadratic Equation

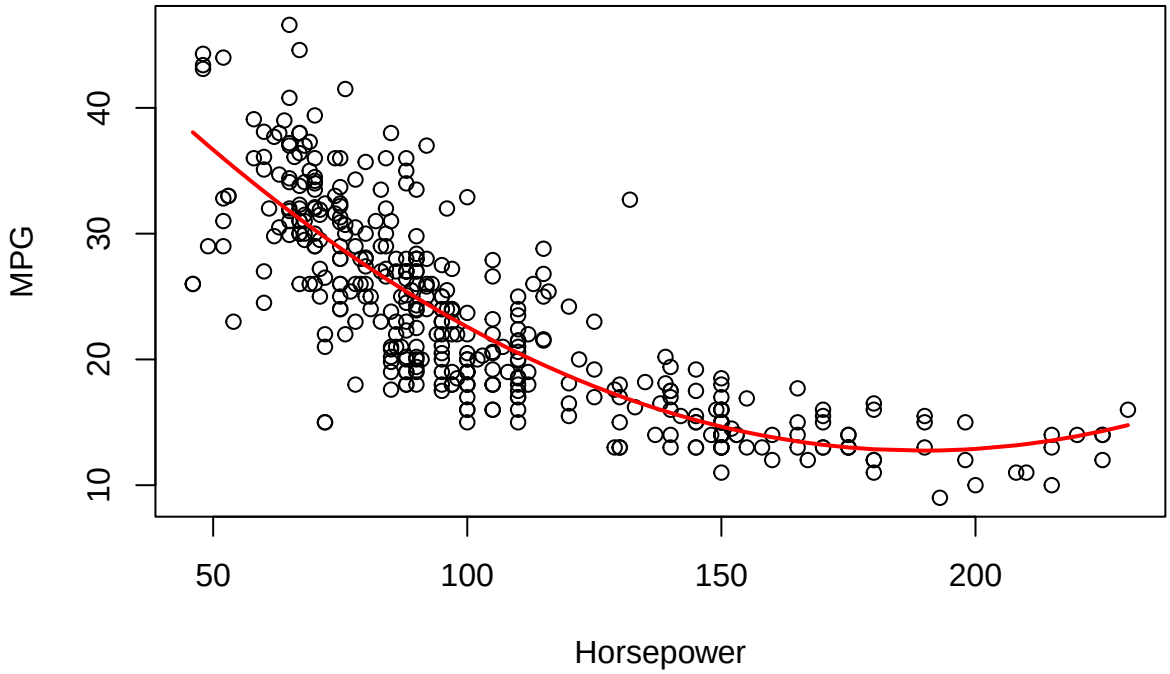
```
# log regression for horsepower
lm5<-lm(mpg ~ log(horsepower))
summary(lm5)
```

```
##
## Call:
## lm(formula = mpg ~ log(horsepower))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.2299  -2.7818  -0.2322   2.6661  15.4695
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     108.6997     3.0496   35.64   <2e-16 ***
## log(horsepower) -18.5822     0.6629  -28.03   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.501 on 390 degrees of freedom
## Multiple R-squared:  0.6683, Adjusted R-squared:  0.6675
## F-statistic: 785.9 on 1 and 390 DF,  p-value: < 2.2e-16
```

```
#visualizing quadratic analysis
sorted<-Auto[order(Auto$horsepower), ]
lm3<-lm(mpg ~ horsepower + I(horsepower^2), data = sorted)
summary(lm3)
```

```
##
## Call:
## lm(formula = mpg ~ horsepower + I(horsepower^2), data = sorted)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.7135  -2.5943  -0.0859   2.2868  15.8961
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     56.9000997  1.8004268   31.60   <2e-16 ***
## horsepower      -0.4661896  0.0311246  -14.98   <2e-16 ***
## I(horsepower^2)  0.0012305  0.0001221   10.08   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.374 on 389 degrees of freedom
## Multiple R-squared:  0.6876, Adjusted R-squared:  0.686
## F-statistic:   428 on 2 and 389 DF,  p-value: < 2.2e-16
```

```r
y_pred<-predict(lm3)
plot(sorted$horsepower, sorted$mpg, xlab = "Horsepower", ylab = "MPG")
lines(sorted$horsepower, y_pred, col = "red", lwd = 2)
```



Final thoughts The relationship between the predictors and MPG dont seem to be linear, and may be better suited in a quadratic model.

```r
head(Carseats)
```

```
##   Sales CompPrice Income Advertising Population Price ShelveLoc Age Education
## 1  9.50       138     73          11        276   120       Bad  42        17
## 2 11.22       111     48          16        260    83      Good  65        10
## 3 10.06       113     35          10        269    80    Medium  59        12
## 4  7.40       117    100           4        466    97    Medium  55        14
## 5  4.15       141     64           3        340   128       Bad  38        13
## 6 10.81       124    113          13        501    72       Bad  78        16
##   Urban  US
## 1   Yes Yes
## 2   Yes Yes
## 3   Yes Yes
## 4   Yes Yes
## 5   Yes  No
## 6    No Yes
```

```r
lm_car<-lm(Sales ~ Price + Urban + US, data = Carseats)
summary(lm_car)
```

```
##
## Call:
## lm(formula = Sales ~ Price + Urban + US, data = Carseats)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9206 -1.6220 -0.0564  1.5786  7.0581
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.043469   0.651012  20.036  < 2e-16 ***
## Price       -0.054459   0.005242 -10.389  < 2e-16 ***
## UrbanYes    -0.021916   0.271650  -0.081    0.936
## USYes        1.200573   0.259042   4.635 4.86e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.472 on 396 degrees of freedom
## Multiple R-squared:  0.2393, Adjusted R-squared:  0.2335
## F-statistic: 41.52 on 3 and 396 DF,  p-value: < 2.2e-16
```

Analysis of coefficients

Looking at this output, the coefficient for price indicates that for a unit increase in price, average sales is changed by -.055. Assuming the Urban and US columns are read in as binary, then sales is changed by -.02 for urbanyes and 1.2 for USyes. This is stated with the evaluation of soley the coefficients, not considering other relevant factors.

f(x) = 13.04 + (-.05)(price) + (-.02)(UrbanYes) + 1.2(USyes)

or assuming binary for urban and Us are No.

f(x) = 13.04 + (-.05)(price)

If you consider the respective T and P values, it looks like only the US and price predictors reject the null.

```r
better_model<-lm(Sales ~ Price + US, data = Carseats)
summary(better_model)
```

```
##
## Call:
## lm(formula = Sales ~ Price + US, data = Carseats)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9269 -1.6286 -0.0574  1.5766  7.0515
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.03079    0.63098  20.652  < 2e-16 ***
## Price       -0.05448    0.00523 -10.416  < 2e-16 ***
```

```
## USYes        1.19964    0.25846   4.641 4.71e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.469 on 397 degrees of freedom
## Multiple R-squared:  0.2393, Adjusted R-squared:  0.2354
## F-statistic: 62.43 on 2 and 397 DF,  p-value: < 2.2e-16
```

Analysis

This is a smaller model with low p values for the predictors, however, some of the other evaluation values have room for improvement. The second model seems to fit a bit better, overall. The diagnostic plots indicate 3 values as potential outliers. If evaluating using 2p/n, then the leverage of the outliers is of concern.

```
confint(better_model)
```

```
##                   2.5 %      97.5 %
## (Intercept) 11.79032020 14.27126531
## Price       -0.06475984 -0.04419543
## USYes        0.69151957  1.70776632
```

```
leverage<-hatvalues(better_model)
summary(leverage)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.003876 0.004543 0.007081 0.007500 0.008820 0.043338
```

```
num<-nobs(better_model)
num
```

```
## [1] 400
```

```
set.seed (1)
x <- rnorm(100)
y <- 2 * x + rnorm(100)
```

```
no_intercept<-lm(y ~ x + 0)
summary(no_intercept)
```

```
##
## Call:
## lm(formula = y ~ x + 0)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9154 -0.6472 -0.1771  0.5056  2.3109
##
## Coefficients:
##   Estimate Std. Error t value Pr(>|t|)
## x   1.9939     0.1065   18.73   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9586 on 99 degrees of freedom
## Multiple R-squared:  0.7798, Adjusted R-squared:  0.7776
## F-statistic: 350.7 on 1 and 99 DF,  p-value: < 2.2e-16
```

No intercept Results

This shows the estimated coefficient to be 1.99, Std Err. .1 t value of 18.73 and a p-value below .05 allowing us to reject the null.

```
no_intercept1<-lm(x ~ y + 0)
summary(no_intercept1)
```

```
##
## Call:
## lm(formula = x ~ y + 0)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.8699 -0.2368  0.1030  0.2858  0.8938
##
## Coefficients:
##   Estimate Std. Error t value Pr(>|t|)
## y  0.39111    0.02089   18.73   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4246 on 99 degrees of freedom
## Multiple R-squared:  0.7798, Adjusted R-squared:  0.7776
## F-statistic: 350.7 on 1 and 99 DF,  p-value: < 2.2e-16
```

Follow up

Reversing the variables shows an estimated coefficient of .37 with a standard error of .02, a t value of 17.62

t-equation (saving code)

$$t = \frac{\hat{\beta}}{SE(\hat{\beta})}$$

SE(B hat) Expanded

$$t = \frac{\beta^2}{\sqrt{\frac{\sum(y_i - x_i\beta)^2}{(n-1)\sum x_i^2}}}$$

Then

Sqaure bottom term, factor by (n-1) sum xi^2, expand (y1-x1b)^2. Factor in sum then.

$$t2 = \beta^2 \left(\frac{(n-1)(\sum x_i^2)^2}{\sum y_i^2 \sum x_i^2 - (\sum x_i y_i)^2} \right)$$

=

$$t2 = \left(\frac{(\sum x_i y_i)(\sum x_i^2)}{\sum x_i^2} \right)^2 \left(\frac{(n-1)(\sum x_i^2)^2}{\sum y_i^2 \sum x_i^2 - (\sum x_i y_i)^2} \right)$$

=

$$t2 = \frac{(n-1)(\sum x_i y_i)^2}{\sum y_i^2 \sum x_i^2 - (\sum x_i y_i)^2}$$

=

$$t = \frac{\sqrt{n-1} \sum x_i y_i}{\sqrt{\sum y_i^2 \sum x_i^2 - (\sum x_i y_i)^2}}$$

Now lets try With x and y reversals

```
n<-length(x)
t_v<-sqrt(n - 1)*(x %*% y)/sqrt(sum(x^2) * sum(y^2) - (x %*% y)^2)
print(t_v)
```

```
##          [,1]
## [1,] 18.72593
```

Replace x with y

```
t_v2<-sqrt(n - 1)*(y %*% x)/sqrt(sum(y^2) * sum(x^2) - (y %*% x)^2)
print(t_v2)
```

```
##          [,1]
## [1,] 18.72593
```

Evaluation with Intercept

```
lm_int<- lm(x ~ y)
summary(lm_int)
```

```
##
## Call:
## lm(formula = x ~ y)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.90848 -0.28101  0.06274  0.24570  0.85736
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.03880    0.04266    0.91    0.365
## y            0.38942    0.02099   18.56   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4249 on 98 degrees of freedom
## Multiple R-squared:  0.7784, Adjusted R-squared:  0.7762
## F-statistic: 344.3 on 1 and 98 DF,  p-value: < 2.2e-16
```

```
lm_int2<- lm(y ~ x)
summary(lm_int2)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8768 -0.6138 -0.1395  0.5394  2.3462
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.03769    0.09699  -0.389    0.698
## x            1.99894    0.10773  18.556   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9628 on 98 degrees of freedom
## Multiple R-squared:  0.7784, Adjusted R-squared:  0.7762
## F-statistic: 344.3 on 1 and 98 DF,  p-value: < 2.2e-16
```

The results with an intercept show reversing the variables sustains the same t value (18.5)

Coefficients without Intercepts

Assuming you have:

$$\hat{\beta} = \frac{\sum_i x_i y_i}{\sum_j x_j^2}$$

The bottom term would need to be

$$\sum_j x_j^2 = \sum_j y_j^2$$

```
set.seed(1)
x<-1:100
y<-2 * x +rnorm(100, sd=0.1)
lmx<- lm(x ~ y + 0)
lmy<- lm (y ~ x + 0)
coef(lmx)
```

```
##         y
## 0.4999619
```

```
coef(lmy)
```

```
##        x
## 2.000151
```

Lets try to make the coefficients the same

```
set.seed(42)
x<-1:100
y<-100:1
# regression models
lx<-lm(x ~ y + 0)
ly<-lm(y ~ x + 0)
coef(lx)
```

```
##         y
## 0.5074627
```

```
coef(ly)
```

```
##         x
## 0.5074627
```

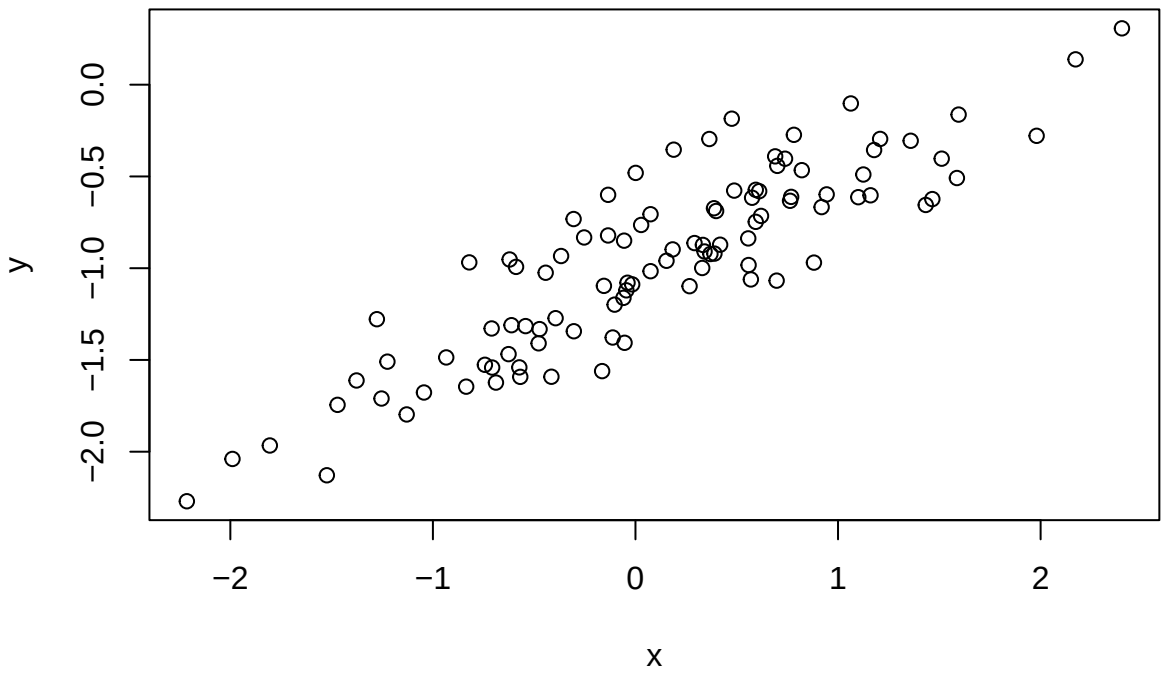
```
set.seed(1)
x<- rnorm(100)
esp<-rnorm(100, mean = 0, sd = 0.25)
y<- -1 + 0.5*x + esp
y2<- -1 + 0.5*x
length(y)
```

```
## [1] 100
```

Parameters of this model

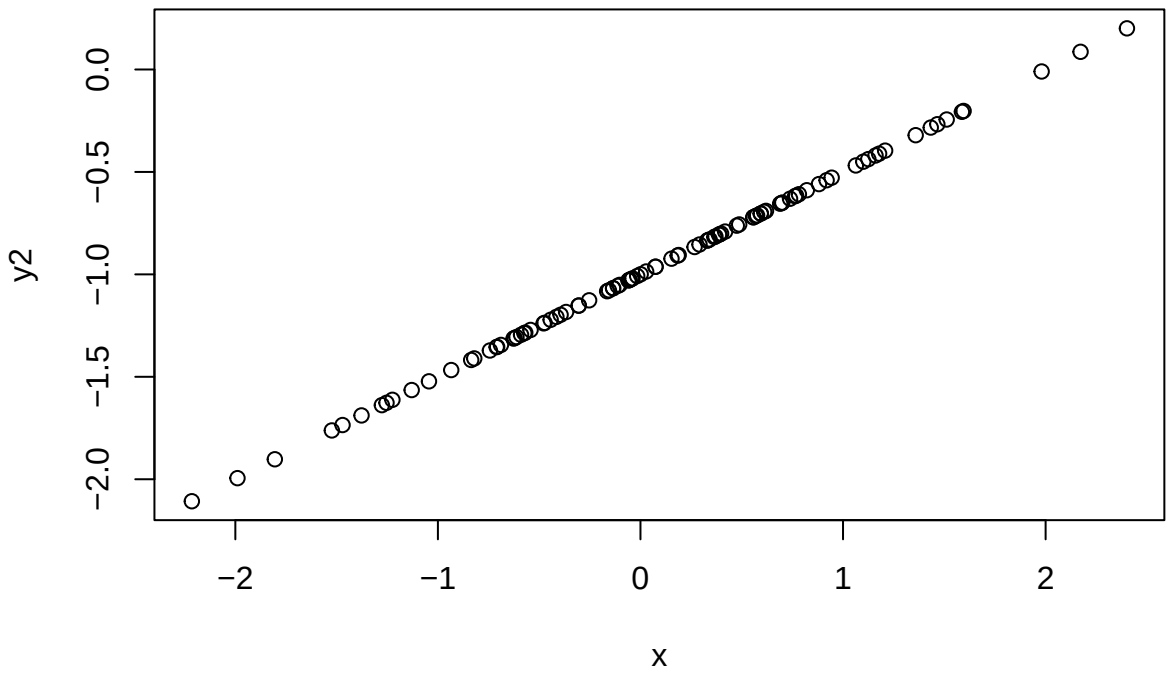
$$\beta_0 = -1, \beta_1 = 0.5$$

```
plot(x, y)
```



Plot Evaluation The values from the plot demonstrate a positive trend with variance (e). For comparision:

```
plot(x, y2)
```



A relationship between x and y assuming no var(e)

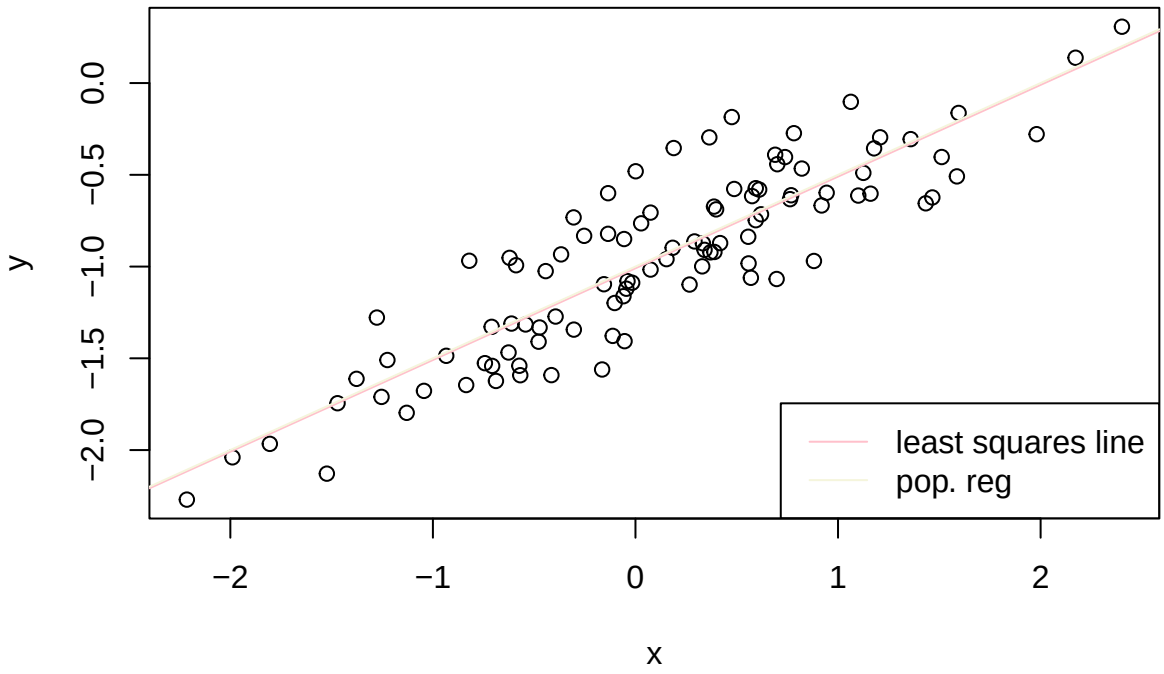
```
fit2<-lm(y ~ x)
summary(fit2)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.46921 -0.15344 -0.03487  0.13485  0.58654
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.00942    0.02425  -41.63   <2e-16 ***
## x            0.49973    0.02693   18.56   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2407 on 98 degrees of freedom
## Multiple R-squared:  0.7784, Adjusted R-squared:  0.7762
## F-statistic: 344.3 on 1 and 98 DF,  p-value: < 2.2e-16
```

Coefficient Approximation

The coefficient estimates are pretty close to the true coefficients

```
plot(x, y)
abline(fit2, col = "pink")
abline(-1, 0.5, col = "beige")
legend("bottomright", c("least squares line", "pop. reg"), col = c("pink", "beige"), lty = c(1,1))
```




```
# Trying quadratic
lm_quad<-lm(y ~ x + I(x^2))
summary(lm_quad)
```

```
##
## Call:
## lm(formula = y ~ x + I(x^2))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.4913 -0.1563 -0.0322  0.1451  0.5675
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.98582    0.02941 -33.516   <2e-16 ***
## x            0.50429    0.02700  18.680   <2e-16 ***
```

```
## I(x^2)      -0.02973    0.02119  -1.403    0.164
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2395 on 97 degrees of freedom
## Multiple R-squared:  0.7828, Adjusted R-squared:  0.7784
## F-statistic: 174.8 on 2 and 97 DF,  p-value: < 2.2e-16
```

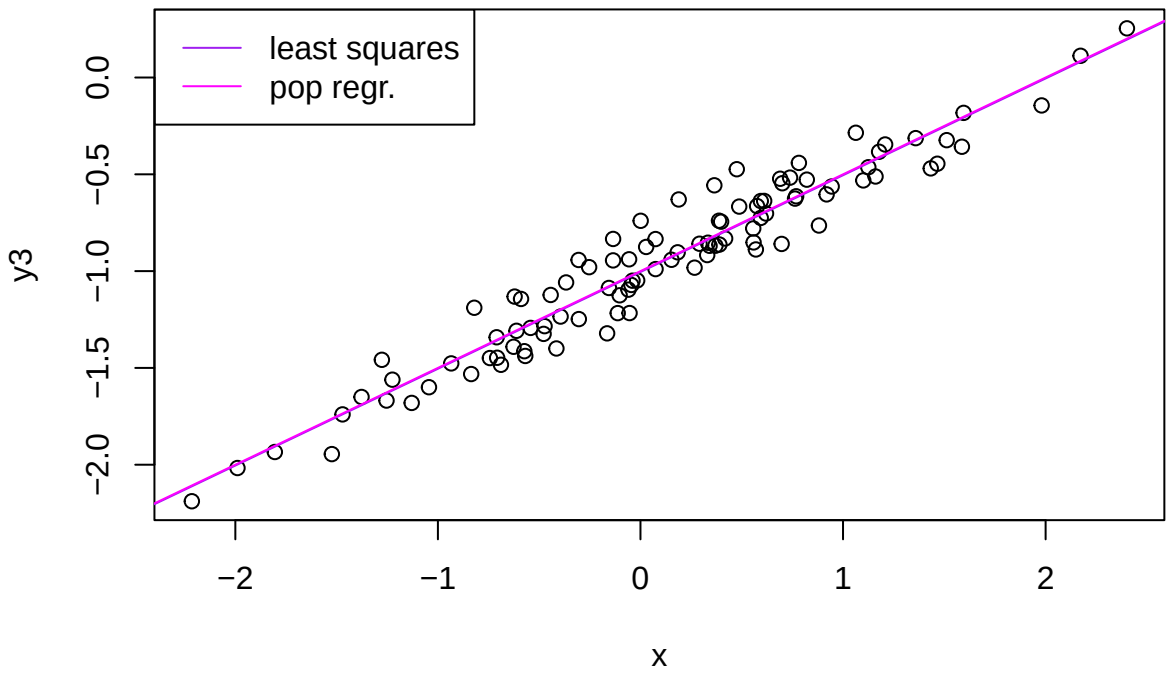
Results from quad terms

This appears to be a weaker model, with a significantly lower t value and F statistic, while the p value for x^2 fails to reject the null. Let examine these variables with reducing the noise and impact of the error term.

```
set.seed(1)
x<-rnorm(100)
esp2<-rnorm(100, sd = 0.125)
y3<- -1 + 0.5*x + esp2
lm_clean<-lm(y3 ~ x)
summary(lm_clean)
```

```
##
## Call:
## lm(formula = y3 ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.23461 -0.07672 -0.01744  0.06742  0.29327
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.00471    0.01212  -82.87   <2e-16 ***
## x            0.49987    0.01347   37.12   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1203 on 98 degrees of freedom
## Multiple R-squared:  0.9336, Adjusted R-squared:  0.9329
## F-statistic:  1378 on 1 and 98 DF,  p-value: < 2.2e-16
```

```
plot(x, y3)
abline(lm_clean, col = "purple")
abline(-1, 0.5, col = "magenta")
legend("topleft", c("least squares", "pop regr."), col = c("purple", "magenta"), lty = c(1, 1))
```




```
summary(lm_clean)
```

```
##
## Call:
## lm(formula = y3 ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.23461 -0.07672 -0.01744  0.06742  0.29327
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.00471    0.01212  -82.87   <2e-16 ***
## x            0.49987    0.01347   37.12   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1203 on 98 degrees of freedom
## Multiple R-squared:  0.9336, Adjusted R-squared:  0.9329
## F-statistic:  1378 on 1 and 98 DF,  p-value: < 2.2e-16
```

Reduced Error Analysis

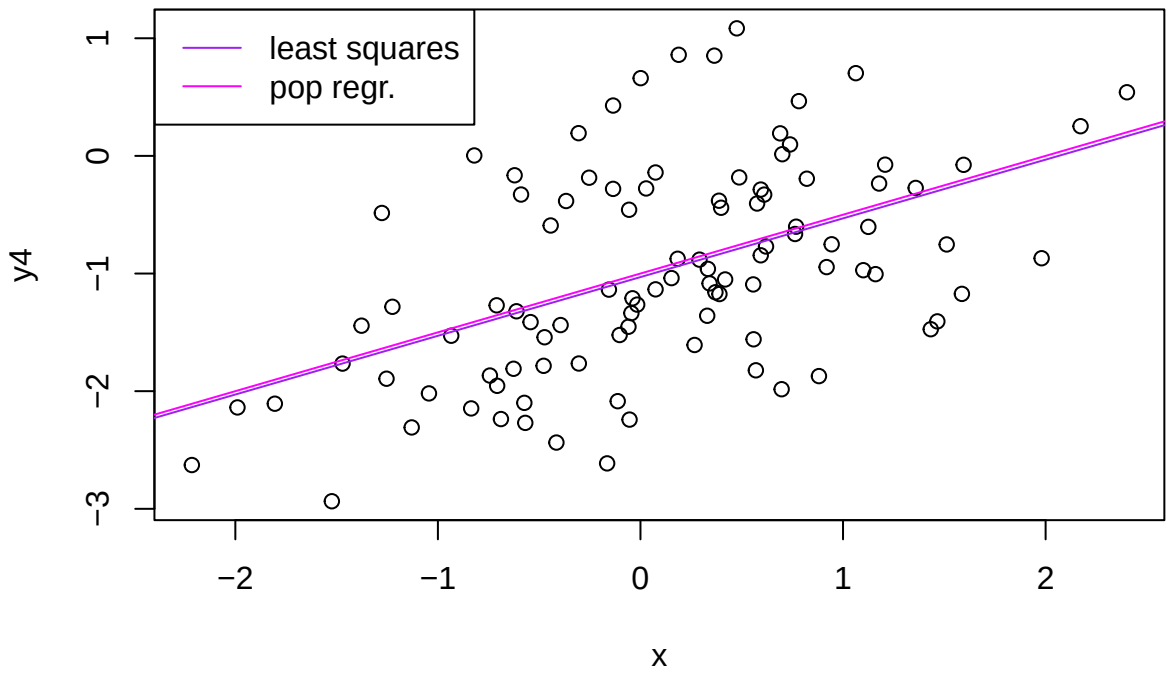
As expected, less noise results in less error. The f statistic is large, t value, and R^2 values are high. Low p value, this is a great model.

More noise???

```
set.seed(1)
x<-rnorm(100)
esp2<-rnorm(100, sd = 0.8)
y4<- -1 + 0.5*x + esp2
lm_noise<-lm(y4 ~ x)
summary(lm_noise)
```

```
##
## Call:
## lm(formula = y4 ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5015 -0.4910 -0.1116  0.4315  1.8769
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.03015    0.07759 -13.277  < 2e-16 ***
## x            0.49915    0.08618   5.792 8.42e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7702 on 98 degrees of freedom
## Multiple R-squared:  0.255,  Adjusted R-squared:  0.2474
## F-statistic: 33.55 on 1 and 98 DF,  p-value: 8.421e-08
```

```
plot(x, y4)
abline(lm_noise, col = "purple")
abline(-1, 0.5, col = "magenta")
legend("topleft", c("least squares", "pop regr."), col = c("purple", "magenta"), lty = c(1, 1))
```




```
summary(lm_noise)
```

```
##
## Call:
## lm(formula = y4 ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5015 -0.4910 -0.1116  0.4315  1.8769
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.03015    0.07759 -13.277  < 2e-16 ***
## x            0.49915    0.08618   5.792 8.42e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7702 on 98 degrees of freedom
## Multiple R-squared:  0.255,  Adjusted R-squared:  0.2474
## F-statistic: 33.55 on 1 and 98 DF,  p-value: 8.421e-08
```

Noise Analysis

It shows that increased variance reduces the accuracy of the model. The t value, f statistic, and R2 have decreased substantially.

```r
confint(lm_clean)
```

```
##                  2.5 %     97.5 %
## (Intercept) -1.0287701 -0.9806531
## x            0.4731449  0.5265901
```

```r
confint(lm_noise)
```

```
##                  2.5 %     97.5 %
## (Intercept) -1.1841286 -0.8761796
## x            0.3281271  0.6701763
```

```r
confint(fit2)
```

```
##                  2.5 %     97.5 %
## (Intercept) -1.0575402 -0.9613061
## x            0.4462897  0.5531801
```

Analysis of Confidence Intervals

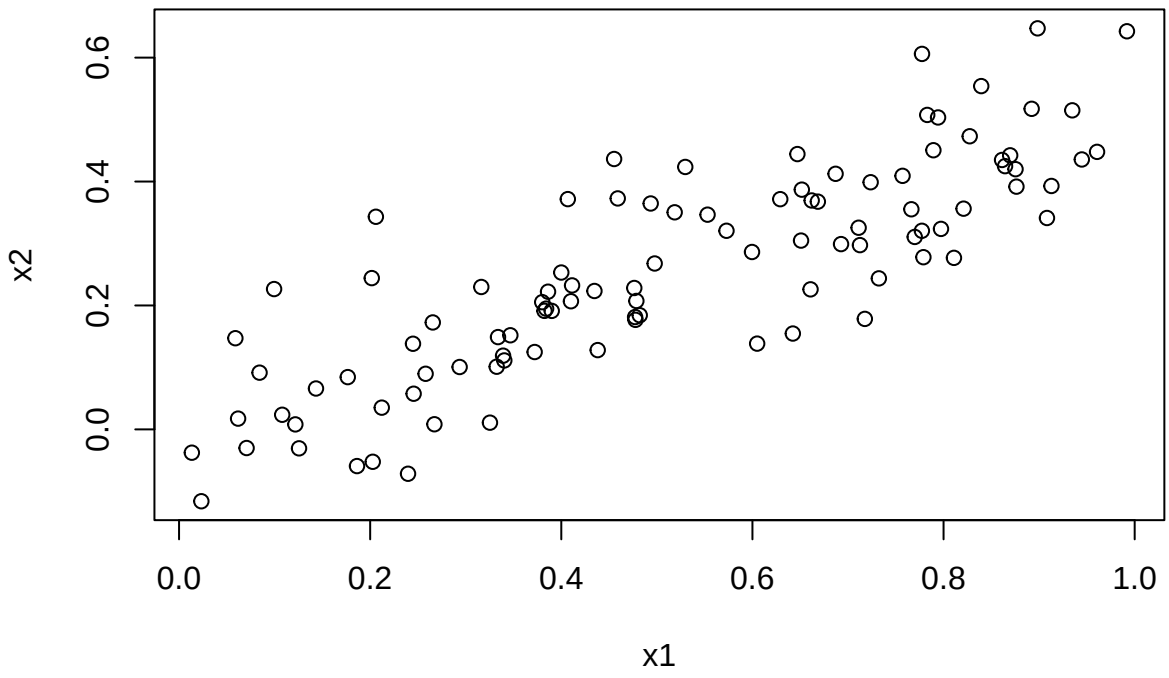
It looks like the ranges center around .5. The fit model and less noisey model are closer to this value, with the noisy model having a wider range.

```r
set.seed (1)
x1 <- runif (100)
x2 <- 0.5 * x1 + rnorm (100) / 10
y <- 2 + 2 * x1 + 0.3 * x2 + rnorm (100)
```

$$y = 2 + 2 \cdot x_1 + 0.3 \cdot x_2 + \varepsilon$$

$$\beta_0 = 2, \beta_1 = 2, \beta_3 = 0.3$$

```r
plot(x1, x2)
```



Positive correlation

```
fit4<-lm(y ~ x1 + x2)
summary(fit4)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8311 -0.7273 -0.0537  0.6338  2.3359
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.1305     0.2319   9.188 7.61e-15 ***
## x1            1.4396     0.7212   1.996   0.0487 *
## x2            1.0097     1.1337   0.891   0.3754
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.056 on 97 degrees of freedom
## Multiple R-squared:  0.2088, Adjusted R-squared:  0.1925
## F-statistic:  12.8 on 2 and 97 DF,  p-value: 1.164e-05
```

Coefficients are 2.1, 1.4, 1.0, with B0 hat being the only coefficient close to BO.

Fit 4 Evaluation

Following the trend of evaluation indicators we have been using throughout this project, this appears to be a weak model. Both predictors have an insignificant (or close) p value, low r2 and f stat, low t value. Lets unpack these terms.

```
fit5<-lm(y ~ x1)
fit6<-lm(y ~ x2)
summary(fit5)
```

```
##
## Call:
## lm(formula = y ~ x1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.89495 -0.66874 -0.07785  0.59221  2.45560
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.1124     0.2307   9.155 8.27e-15 ***
## x1            1.9759     0.3963   4.986 2.66e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.055 on 98 degrees of freedom
## Multiple R-squared:  0.2024, Adjusted R-squared:  0.1942
## F-statistic: 24.86 on 1 and 98 DF,  p-value: 2.661e-06
```

```
summary(fit6)
```

```
##
## Call:
## lm(formula = y ~ x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.62687 -0.75156 -0.03598  0.72383  2.44890
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.3899     0.1949   12.26  < 2e-16 ***
## x2            2.8996     0.6330    4.58 1.37e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.072 on 98 degrees of freedom
## Multiple R-squared:  0.1763, Adjusted R-squared:  0.1679
## F-statistic: 20.98 on 1 and 98 DF,  p-value: 1.366e-05
```

Individual Predictors Analysis

Looking at the outcomes form this assessment, the predictors show higher significance, though the other evaluation indicators are not outstanding. These results do contradict outcomes from a multiple linear regression, as a significant relation is assumed between the individual predictors.

```
x1 <- c(x1 , 0.1)
x2 <- c(x2 , 0.8)
y <- c(y, 6)
```

```
fit8<-lm(y ~ x1 + x2)
summary(fit8)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.73348 -0.69318 -0.05263  0.66385  2.30619
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.2267     0.2314   9.624 7.91e-16 ***
## x1            0.5394     0.5922   0.911  0.36458
## x2            2.5146     0.8977   2.801  0.00614 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.075 on 98 degrees of freedom
## Multiple R-squared:  0.2188, Adjusted R-squared:  0.2029
## F-statistic: 13.72 on 2 and 98 DF,  p-value: 5.564e-06
```

```
fit9<-lm(y ~ x1)
summary(fit9)
```

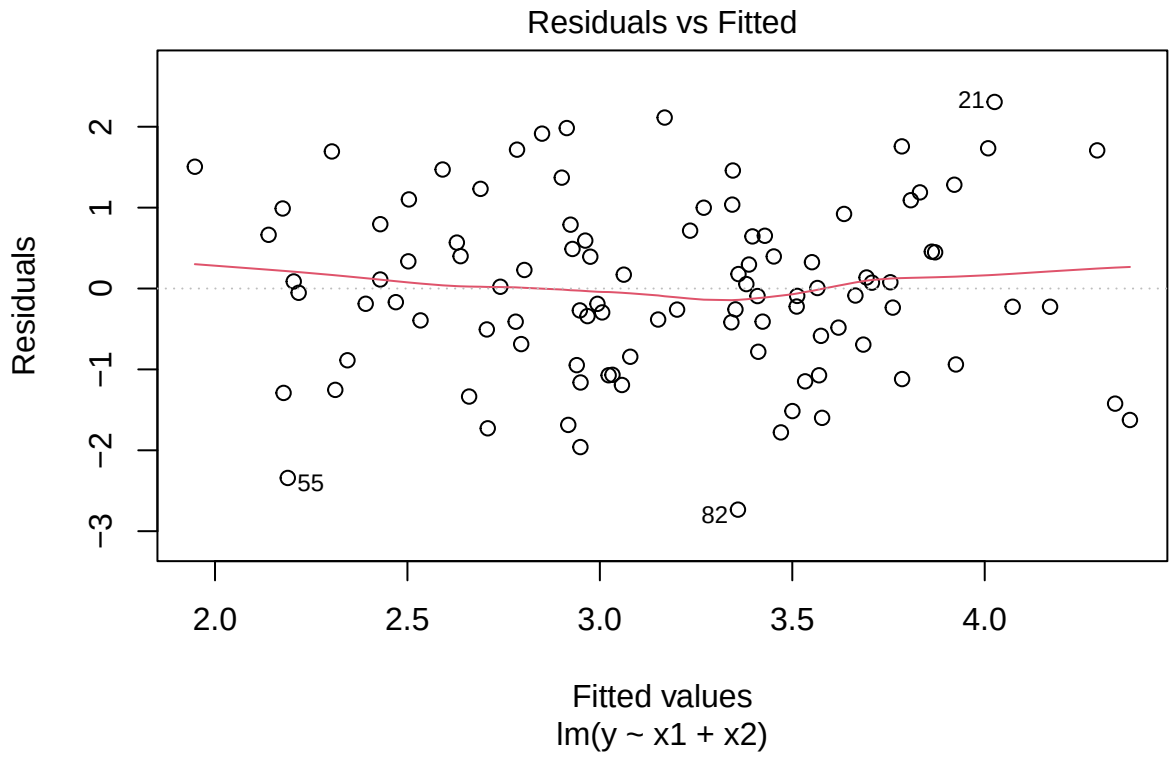
```
##
## Call:
## lm(formula = y ~ x1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8897 -0.6556 -0.0909  0.5682  3.5665
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.2569     0.2390   9.445 1.78e-15 ***
## x1            1.7657     0.4124   4.282 4.29e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.111 on 99 degrees of freedom
```

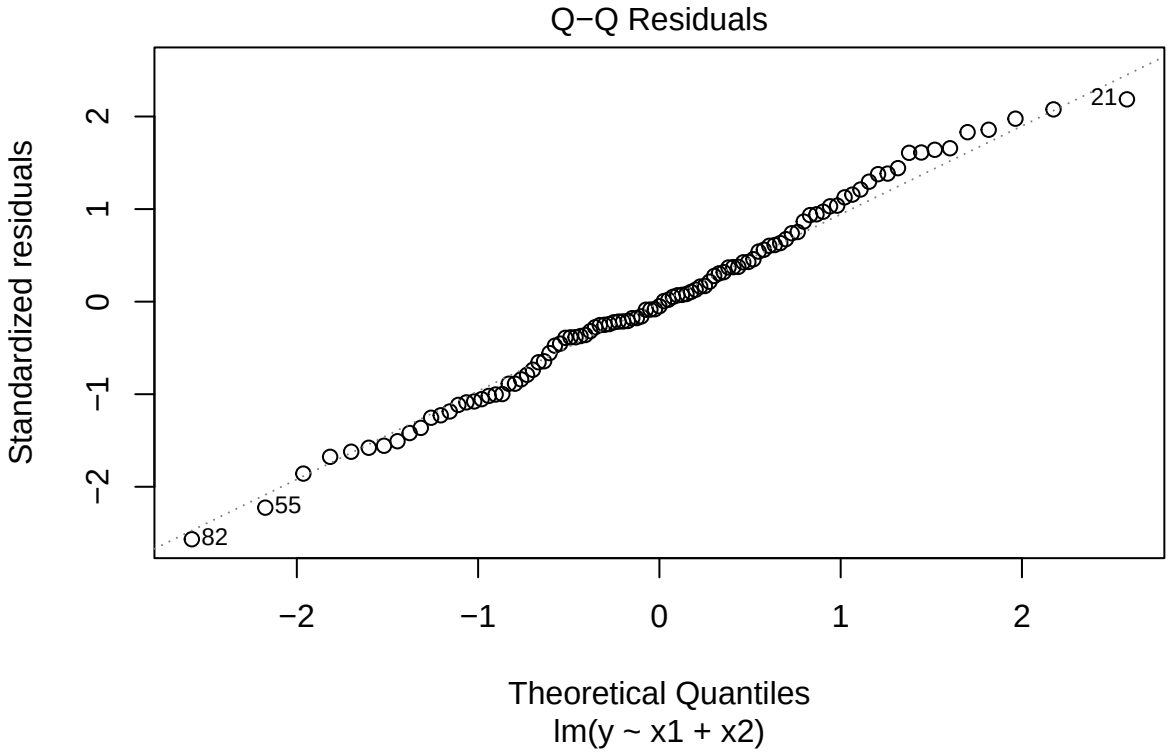
```
## Multiple R-squared:  0.1562, Adjusted R-squared:  0.1477
## F-statistic: 18.33 on 1 and 99 DF,  p-value: 4.295e-05
```

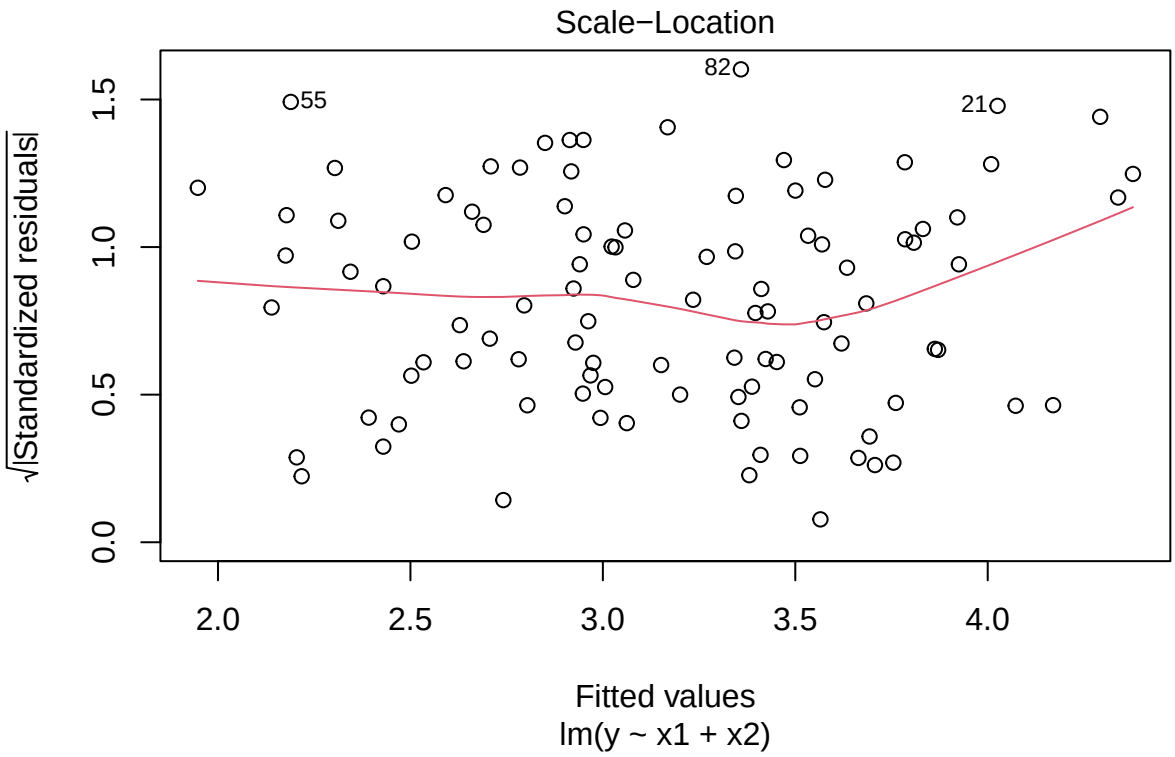
```
fit10<-lm(y ~ x2)
summary(fit10)
```

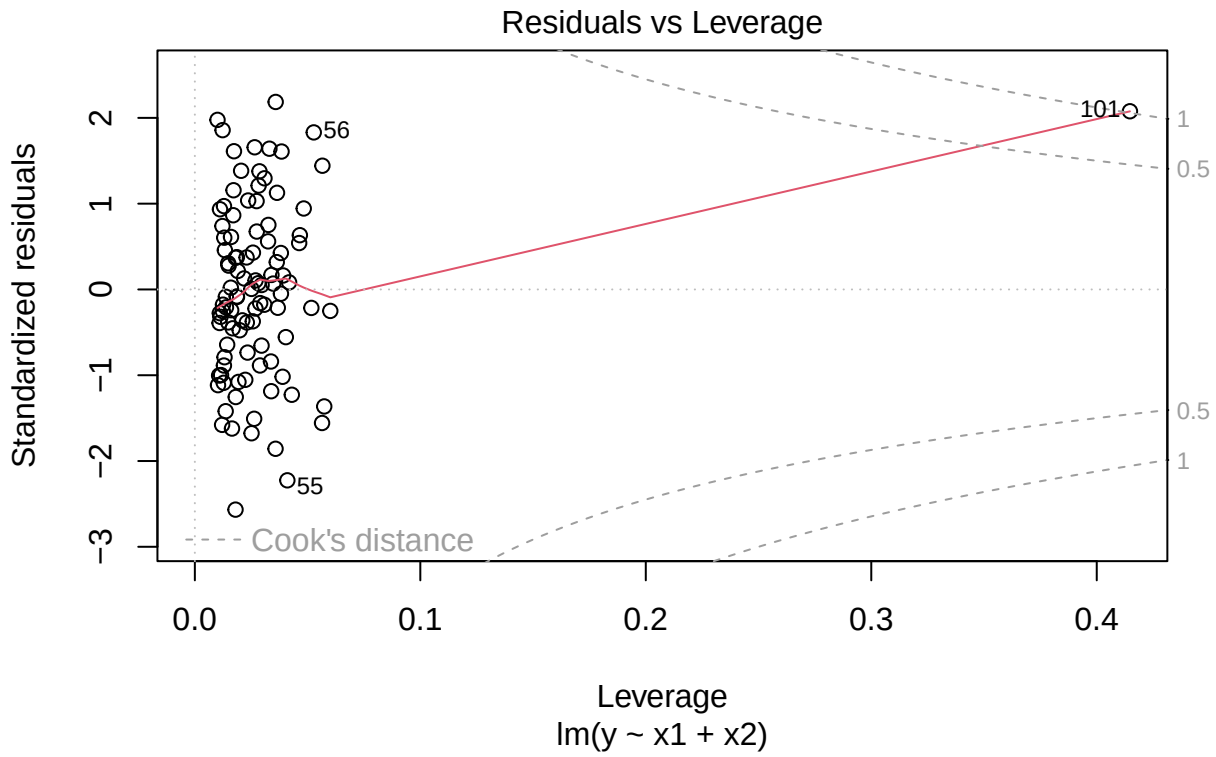
```
##
## Call:
## lm(formula = y ~ x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.64729 -0.71021 -0.06899  0.72699  2.38074
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.3451     0.1912  12.264  < 2e-16 ***
## x2            3.1190     0.6040   5.164 1.25e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.074 on 99 degrees of freedom
## Multiple R-squared:  0.2122, Adjusted R-squared:  0.2042
## F-statistic: 26.66 on 1 and 99 DF,  p-value: 1.253e-06
```

```
plot(fit8)
```

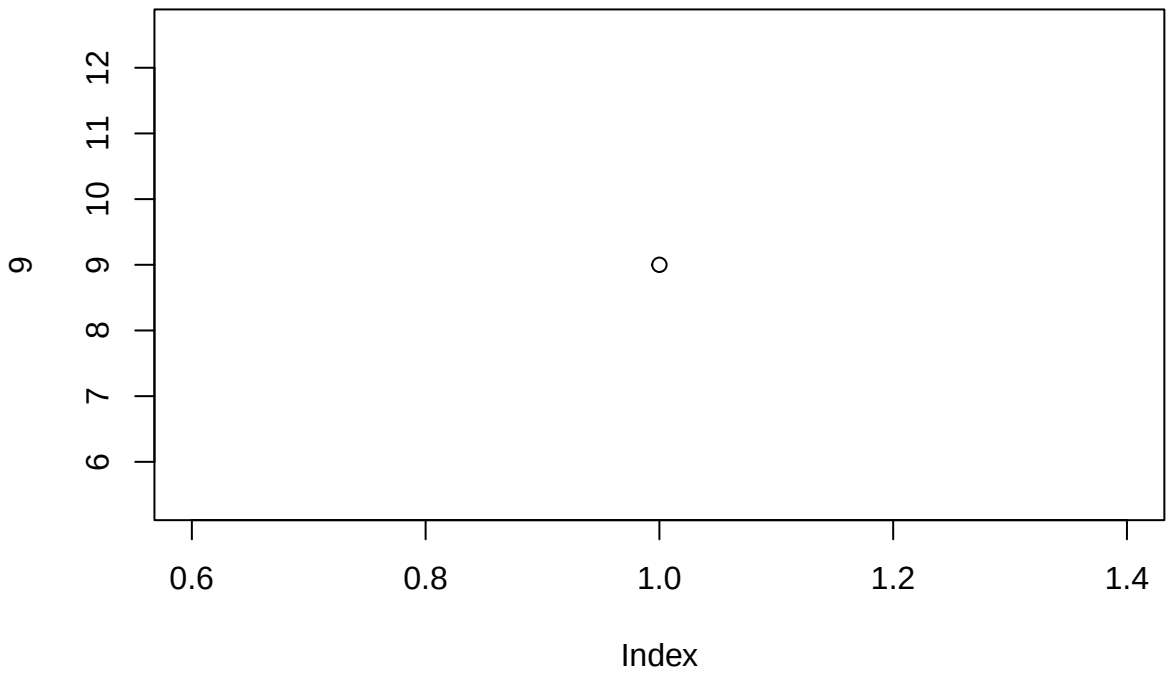



Q−Q Residuals
Standardized residuals
−2
−1
0
1
2
82
55
21
−2
−1
0
1
2
Theoretical Quantiles
lm(y ~ x1 + x2)

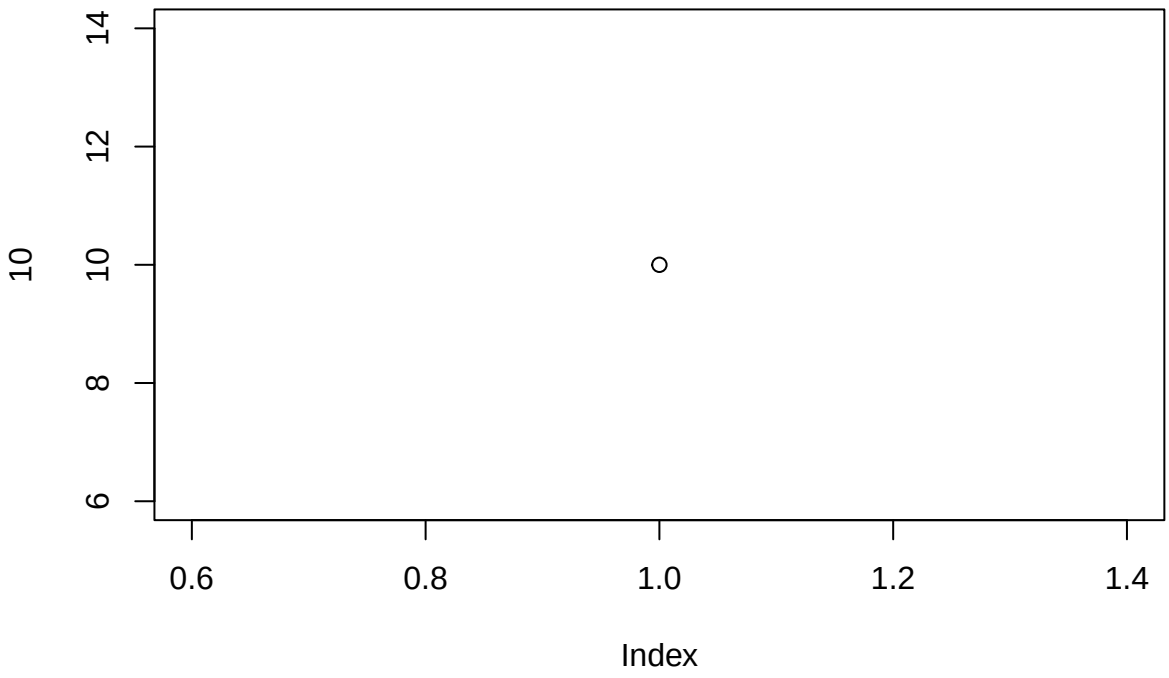

Scale−Location
82
55
21
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Fitted values
lm(y ~ x1 + x2)


Residuals vs Leverage
Standardized residuals
Leverage
lm(y ~ x1 + x2)
Cook's distance
101
56
55
1
0.5
0.5
1
0.0
0.1
0.2
0.3
0.4
−3
−2
−1
0
1
2

```
plot(9)
```




```
plot(10)
```



The last point presents as either and outlier or having a high leverage point across models.

```
names(Boston)
```

```
##  [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"      "age"
##  [8] "dis"     "rad"     "tax"     "ptratio" "lstat"   "medv"
```

```
library(nlme)
predictors<-c("zn", "indus", "chas", "nox", "rm", "age", "dis", "rad", "tax", "ptratio", "lstat", "medv
mod<- lapply(predictors, function(predictor) {
  formula<- as.formula(paste("crim", "~", predictor))
  lm(formula, data = Boston)
})

for (i in seq_along(mod)) {
  cat("Summary:", predictors[1], "\n")
  print(summary(mod[[i]]))
  cat("\n")
}
```

```
## Summary: zn
##
## Call:
## lm(formula = formula, data = Boston)
##
```

```
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.429 -4.222 -2.620  1.250 84.523
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.45369    0.41722  10.675  < 2e-16 ***
## zn          -0.07393    0.01609  -4.594 5.51e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.435 on 504 degrees of freedom
## Multiple R-squared:  0.04019,    Adjusted R-squared:  0.03828
## F-statistic:  21.1 on 1 and 504 DF,  p-value: 5.506e-06
##
##
## Summary: zn
##
## Call:
## lm(formula = formula, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.972  -2.698  -0.736   0.712  81.813
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.06374    0.66723  -3.093  0.00209 **
## indus        0.50978    0.05102   9.991  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.866 on 504 degrees of freedom
## Multiple R-squared:  0.1653, Adjusted R-squared:  0.1637
## F-statistic: 99.82 on 1 and 504 DF,  p-value: < 2.2e-16
##
##
## Summary: zn
##
## Call:
## lm(formula = formula, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.738 -3.661 -3.435  0.018 85.232
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.7444     0.3961   9.453   <2e-16 ***
## chas         -1.8928     1.5061  -1.257    0.209
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.597 on 504 degrees of freedom
```

```
## Multiple R-squared:  0.003124,   Adjusted R-squared:  0.001146
## F-statistic: 1.579 on 1 and 504 DF,  p-value: 0.2094
##
##
## Summary: zn
##
## Call:
## lm(formula = formula, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.371  -2.738  -0.974   0.559  81.728
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -13.720      1.699  -8.073 5.08e-15 ***
## nox           31.249      2.999  10.419  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.81 on 504 degrees of freedom
## Multiple R-squared:  0.1772, Adjusted R-squared:  0.1756
## F-statistic: 108.6 on 1 and 504 DF,  p-value: < 2.2e-16
##
##
## Summary: zn
##
## Call:
## lm(formula = formula, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.604 -3.952 -2.654  0.989 87.197
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   20.482      3.365   6.088 2.27e-09 ***
## rm            -2.684      0.532  -5.045 6.35e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.401 on 504 degrees of freedom
## Multiple R-squared:  0.04807,    Adjusted R-squared:  0.04618
## F-statistic: 25.45 on 1 and 504 DF,  p-value: 6.347e-07
##
##
## Summary: zn
##
## Call:
## lm(formula = formula, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.789 -4.257 -1.230  1.527 82.849
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.77791    0.94398  -4.002 7.22e-05 ***
## age          0.10779    0.01274   8.463 2.85e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.057 on 504 degrees of freedom
## Multiple R-squared:  0.1244, Adjusted R-squared:  0.1227
## F-statistic: 71.62 on 1 and 504 DF,  p-value: 2.855e-16
##
##
## Summary: zn
##
## Call:
## lm(formula = formula, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.708 -4.134 -1.527  1.516 81.674
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9.4993     0.7304  13.006   <2e-16 ***
## dis          -1.5509     0.1683  -9.213   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.965 on 504 degrees of freedom
## Multiple R-squared:  0.1441, Adjusted R-squared:  0.1425
## F-statistic: 84.89 on 1 and 504 DF,  p-value: < 2.2e-16
##
##
## Summary: zn
##
## Call:
## lm(formula = formula, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.164  -1.381  -0.141   0.660  76.433
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.28716    0.44348  -5.157 3.61e-07 ***
## rad          0.61791    0.03433  17.998  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.718 on 504 degrees of freedom
## Multiple R-squared:  0.3913, Adjusted R-squared:   0.39
## F-statistic: 323.9 on 1 and 504 DF,  p-value: < 2.2e-16
##
```

```
##
## Summary: zn
##
## Call:
## lm(formula = formula, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.513  -2.738  -0.194   1.065  77.696
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.528369   0.815809  -10.45   <2e-16 ***
## tax          0.029742   0.001847   16.10   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.997 on 504 degrees of freedom
## Multiple R-squared:  0.3396, Adjusted R-squared:  0.3383
## F-statistic: 259.2 on 1 and 504 DF,  p-value: < 2.2e-16
##
##
## Summary: zn
##
## Call:
## lm(formula = formula, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.654 -3.985 -1.912  1.825 83.353
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.6469     3.1473  -5.607 3.40e-08 ***
## ptratio       1.1520     0.1694   6.801 2.94e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.24 on 504 degrees of freedom
## Multiple R-squared:  0.08407,    Adjusted R-squared:  0.08225
## F-statistic: 46.26 on 1 and 504 DF,  p-value: 2.943e-11
##
##
## Summary: zn
##
## Call:
## lm(formula = formula, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.925  -2.822  -0.664   1.079  82.862
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) -3.33054    0.69376  -4.801 2.09e-06 ***
## lstat        0.54880    0.04776  11.491  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.664 on 504 degrees of freedom
## Multiple R-squared:  0.2076, Adjusted R-squared:  0.206
## F-statistic:   132 on 1 and 504 DF,  p-value: < 2.2e-16
##
##
## Summary: zn
##
## Call:
## lm(formula = formula, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.071 -4.022 -2.343  1.298 80.957
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.79654    0.93419   12.63   <2e-16 ***
## medv        -0.36316    0.03839   -9.46   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.934 on 504 degrees of freedom
## Multiple R-squared:  0.1508, Adjusted R-squared:  0.1491
## F-statistic: 89.49 on 1 and 504 DF,  p-value: < 2.2e-16
```

Chas is the only variable that fails to reject the null.

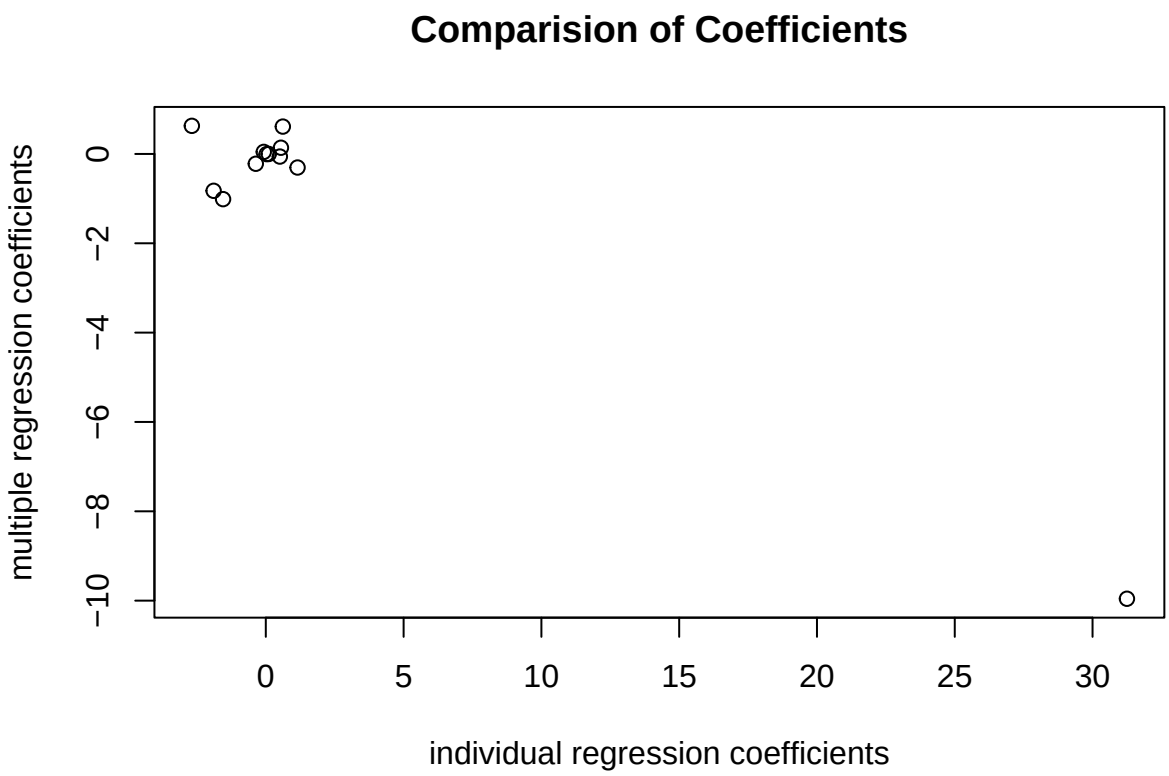
```
fit.sum <- lm(crim ~ ., data = Boston)
summary(fit.sum)
```

```
##
## Call:
## lm(formula = crim ~ ., data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.534 -2.248 -0.348  1.087 73.923
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.7783938  7.0818258   1.946 0.052271 .
## zn           0.0457100  0.0187903   2.433 0.015344 *
## indus       -0.0583501  0.0836351  -0.698 0.485709
## chas        -0.8253776  1.1833963  -0.697 0.485841
## nox         -9.9575865  5.2898242  -1.882 0.060370 .
## rm           0.6289107  0.6070924   1.036 0.300738
## age         -0.0008483  0.0179482  -0.047 0.962323
## dis         -1.0122467  0.2824676  -3.584 0.000373 ***
```

```
## rad          0.6124653  0.0875358   6.997 8.59e-12 ***
## tax         -0.0037756  0.0051723  -0.730 0.465757
## ptratio     -0.3040728  0.1863598  -1.632 0.103393
## lstat        0.1388006  0.0757213   1.833 0.067398 .
## medv        -0.2200564  0.0598240  -3.678 0.000261 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.46 on 493 degrees of freedom
## Multiple R-squared:  0.4493, Adjusted R-squared:  0.4359
## F-statistic: 33.52 on 12 and 493 DF,  p-value: < 2.2e-16
```

Factors medv, rad, zn are significant predictors.

```
ind_coef<- sapply(mod, function(model) coef(model)[2])
mul_coef<-coef(fit.sum)[-1]
plot(ind_coef, mul_coef,
     xlab = "individual regression coefficients",
     ylab = "multiple regression coefficients",
     main = "Comparision of Coefficients")
```



There is a difference between the coefficients between the models, individual and multiple regression. This is likely due to the inclusion of effect from the presence of other predictors in the multiple regression, which a simple linear regression will ignore.

```r
# chas removed as qualitative
predictors<-c("zn", "indus", "nox", "rm", "age", "dis", "rad", "tax", "ptratio", "lstat", "medv")

degree<- 3

mod<- lapply(predictors, function(predictor) {
  formula<- as.formula(paste("crim", "~ poly(", predictor, ",", degree, ")"))
  lm(formula, data = Boston)
})

for (i in seq_along(mod)) {
  predictor<-predictors[i]
  cat("Summary:", predictors, "\n")
  print(summary(mod[[i]]))
  cat("\n")
}
```

```
## Summary: zn indus nox rm age dis rad tax ptratio lstat medv
##
## Call:
## lm(formula = formula, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.821 -4.614 -1.294  0.473 84.130
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.6135     0.3722   9.709  < 2e-16 ***
## poly(zn, 3)1 -38.7498     8.3722  -4.628  4.7e-06 ***
## poly(zn, 3)2  23.9398     8.3722   2.859  0.00442 **
## poly(zn, 3)3 -10.0719     8.3722  -1.203  0.22954
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.372 on 502 degrees of freedom
## Multiple R-squared:  0.05824,    Adjusted R-squared:  0.05261
## F-statistic: 10.35 on 3 and 502 DF,  p-value: 1.281e-06
##
##
## Summary: zn indus nox rm age dis rad tax ptratio lstat medv
##
## Call:
## lm(formula = formula, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.278 -2.514  0.054  0.764 79.713
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)        3.614      0.330  10.950  < 2e-16 ***
## poly(indus, 3)1   78.591      7.423  10.587  < 2e-16 ***
```

```
## poly(indus, 3)2  -24.395      7.423  -3.286  0.00109 **
## poly(indus, 3)3  -54.130      7.423  -7.292  1.2e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.423 on 502 degrees of freedom
## Multiple R-squared:  0.2597, Adjusted R-squared:  0.2552
## F-statistic: 58.69 on 3 and 502 DF,  p-value: < 2.2e-16
##
##
## Summary: zn indus nox rm age dis rad tax ptratio lstat medv
##
## Call:
## lm(formula = formula, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.110 -2.068 -0.255  0.739 78.302
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.6135     0.3216  11.237  < 2e-16 ***
## poly(nox, 3)1  81.3720     7.2336  11.249  < 2e-16 ***
## poly(nox, 3)2 -28.8286     7.2336  -3.985 7.74e-05 ***
## poly(nox, 3)3 -60.3619     7.2336  -8.345 6.96e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.234 on 502 degrees of freedom
## Multiple R-squared:  0.297,  Adjusted R-squared:  0.2928
## F-statistic: 70.69 on 3 and 502 DF,  p-value: < 2.2e-16
##
##
## Summary: zn indus nox rm age dis rad tax ptratio lstat medv
##
## Call:
## lm(formula = formula, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.485  -3.468  -2.221  -0.015  87.219
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.6135     0.3703   9.758  < 2e-16 ***
## poly(rm, 3)1 -42.3794     8.3297  -5.088 5.13e-07 ***
## poly(rm, 3)2  26.5768     8.3297   3.191  0.00151 **
## poly(rm, 3)3  -5.5103     8.3297  -0.662  0.50858
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.33 on 502 degrees of freedom
## Multiple R-squared:  0.06779,    Adjusted R-squared:  0.06222
## F-statistic: 12.17 on 3 and 502 DF,  p-value: 1.067e-07
```

```
##
##
## Summary: zn indus nox rm age dis rad tax ptratio lstat medv
##
## Call:
## lm(formula = formula, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.762 -2.673 -0.516  0.019 82.842
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.6135     0.3485  10.368  < 2e-16 ***
## poly(age, 3)1  68.1820     7.8397   8.697  < 2e-16 ***
## poly(age, 3)2  37.4845     7.8397   4.781 2.29e-06 ***
## poly(age, 3)3  21.3532     7.8397   2.724  0.00668 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.84 on 502 degrees of freedom
## Multiple R-squared:  0.1742, Adjusted R-squared:  0.1693
## F-statistic: 35.31 on 3 and 502 DF,  p-value: < 2.2e-16
##
##
## Summary: zn indus nox rm age dis rad tax ptratio lstat medv
##
## Call:
## lm(formula = formula, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.757  -2.588   0.031   1.267  76.378
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.6135     0.3259  11.087  < 2e-16 ***
## poly(dis, 3)1 -73.3886     7.3315 -10.010  < 2e-16 ***
## poly(dis, 3)2  56.3730     7.3315   7.689 7.87e-14 ***
## poly(dis, 3)3 -42.6219     7.3315  -5.814 1.09e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.331 on 502 degrees of freedom
## Multiple R-squared:  0.2778, Adjusted R-squared:  0.2735
## F-statistic: 64.37 on 3 and 502 DF,  p-value: < 2.2e-16
##
##
## Summary: zn indus nox rm age dis rad tax ptratio lstat medv
##
## Call:
## lm(formula = formula, data = Boston)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -10.381  -0.412  -0.269   0.179  76.217
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.6135     0.2971  12.164  < 2e-16 ***
## poly(rad, 3)1 120.9074     6.6824  18.093  < 2e-16 ***
## poly(rad, 3)2  17.4923     6.6824   2.618  0.00912 **
## poly(rad, 3)3   4.6985     6.6824   0.703  0.48231
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.682 on 502 degrees of freedom
## Multiple R-squared:    0.4,  Adjusted R-squared:  0.3965
## F-statistic: 111.6 on 3 and 502 DF,  p-value: < 2.2e-16
##
##
## Summary: zn indus nox rm age dis rad tax ptratio lstat medv
##
## Call:
## lm(formula = formula, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.273  -1.389   0.046   0.536  76.950
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.6135     0.3047  11.860  < 2e-16 ***
## poly(tax, 3)1 112.6458     6.8537  16.436  < 2e-16 ***
## poly(tax, 3)2  32.0873     6.8537   4.682 3.67e-06 ***
## poly(tax, 3)3  -7.9968     6.8537  -1.167    0.244
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.854 on 502 degrees of freedom
## Multiple R-squared:  0.3689, Adjusted R-squared:  0.3651
## F-statistic:  97.8 on 3 and 502 DF,  p-value: < 2.2e-16
##
##
## Summary: zn indus nox rm age dis rad tax ptratio lstat medv
##
## Call:
## lm(formula = formula, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.833 -4.146 -1.655  1.408 82.697
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)          3.614      0.361  10.008  < 2e-16 ***
## poly(ptratio, 3)1   56.045      8.122   6.901 1.57e-11 ***
## poly(ptratio, 3)2   24.775      8.122   3.050  0.00241 **
```

```
## poly(ptratio, 3)3  -22.280      8.122  -2.743  0.00630 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.122 on 502 degrees of freedom
## Multiple R-squared:  0.1138, Adjusted R-squared:  0.1085
## F-statistic: 21.48 on 3 and 502 DF,  p-value: 4.171e-13
##
##
## Summary: zn indus nox rm age dis rad tax ptratio lstat medv
##
## Call:
## lm(formula = formula, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.234  -2.151  -0.486   0.066  83.353
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.6135     0.3392  10.654   <2e-16 ***
## poly(lstat, 3)1  88.0697     7.6294  11.543   <2e-16 ***
## poly(lstat, 3)2  15.8882     7.6294   2.082   0.0378 *
## poly(lstat, 3)3 -11.5740     7.6294  -1.517   0.1299
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.629 on 502 degrees of freedom
## Multiple R-squared:  0.2179, Adjusted R-squared:  0.2133
## F-statistic: 46.63 on 3 and 502 DF,  p-value: < 2.2e-16
##
##
## Summary: zn indus nox rm age dis rad tax ptratio lstat medv
##
## Call:
## lm(formula = formula, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -24.427  -1.976  -0.437   0.439  73.655
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.614      0.292  12.374  < 2e-16 ***
## poly(medv, 3)1  -75.058      6.569 -11.426  < 2e-16 ***
## poly(medv, 3)2   88.086      6.569  13.409  < 2e-16 ***
## poly(medv, 3)3  -48.033      6.569  -7.312 1.05e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.569 on 502 degrees of freedom
## Multiple R-squared:  0.4202, Adjusted R-squared:  0.4167
## F-statistic: 121.3 on 3 and 502 DF,  p-value: < 2.2e-16
```

Some models show a significant P value for the cubic term, others do not.