

# ISLR Notes and Exercises

*Nathaniel*

*Updated March 2018*

## Chapter 3: Linear Regression

### Prediction and Inference

From Cross Validated, Inference: Given a set of data you want to infer how the output is generated as a function of the data.

Prediction: Given a new measurement, you want to use an existing data set to build a model that reliably chooses the correct identifier from a set of outcomes."

I must say the definitions of inference, in statistics and econometrics, confuse me. ETM (2004): "If we are to interpret any given set of OLS parameter estimates, we need to know, at least approximately, how  $\hat{\beta}$  is actually distributed. For purposes of **inference**, the most important feature of the distribution of any vector of parameter estimates is the matrix of its central second moments.

It seems that inference according to ETM (2004), concerns about the distribution (1st and 2nd moments) of the estimator while, in ISLR (2013), inference, is about the change in response variable due to the change of an estimated parameters of a specified model.

### Potential Problems of Regression:

- Non-linearity of the response-predictor relationships.
- Correlation of error terms.
- Non-constant variance of error terms (heteroscedasticity).
- Outliers.
- High-leverage points.
- Collinearity.

### Frisch-Waugh-Lovell Theorem

1. The OLS estimates of  $\hat{\beta}$  from regressions

$$y = X_1\beta_1 + X_2\beta_2 + u$$

and

$$M_1y = M_1X_2\beta_2 + residuals$$

are numerically identical, where  $P_X = X(X^T X)^{-1}X^T$  and  $M_x = I - P_X = I - X(X^T X)^{-1}X^T$

2. The residuals from the one-step and two-step regressions are numerically identical (ETM P.69).

Think of the two-step model as regressing the residual of  $y$  on  $X_1$  onto the residual of  $X_1$  on  $X_2$ . The Theorem would shed important insight on the problem of collinearity.

For concepts such as hypothesis testing (t-test and F-test), SSE,  $R^2$ , variance inflation factor (VIF), Mallows'  $C_p$ , leverage point, influential point, AIC, BIC, forward selection, backward selection, mixed selection, see the ISLR book or the solution below for detail.

See ETM for the hat matrix and why “We say that observations for which  $h_t$  is large have high leverage or are leverage points. A leverage point is not necessarily influential, but it has the potential to be influential.”

See also the **broom** package for implementation of models and graphics.

## KNN regression vs KNN classification

### Question 2

KNN regression averages the closest observations to estimate prediction, KNN classifier assigns classification group based on majority of closest observations.

KNN regression: given a value for  $K$  and a prediction point  $x_0$ , KNN regression first identifies the  $K$  training observations that are closest to  $x_0$ , represented by  $N_0$ . It then estimates  $f(x_0)$  using the average of all the training responses in  $N_0$ . Then

$$\hat{f}(x_0) = \frac{1}{K} \sum_{x_i \in N_0} y_i$$

As usual, the greater the  $K$ , the more “smooth” the hyperplane, the lower the variance ( $Var(\hat{f}(x_0))$ ), the more likely of the result of  $Bias(\hat{f}(x_0))$ .  $K$  can be determined by Cross-Validation. Whether KNN regression outperforms linear regression model depends on the true model which is rarely known to modellers.

KNN classifier: given a positive K-nearest integer  $K$  and a test observation  $x_0$ , the KNN classifier first identifies the neighbors  $K$  points in the training data that are closest to  $x_0$ , represented by  $N_0$ . It then estimates the conditional probability for class  $j$  as the fraction of points in  $N_0$  whose response values equal  $j$ :

$$Pr(Y = j|X = x_0) = \frac{1}{K} \sum_{i \in N_0} I(y_i = j)$$

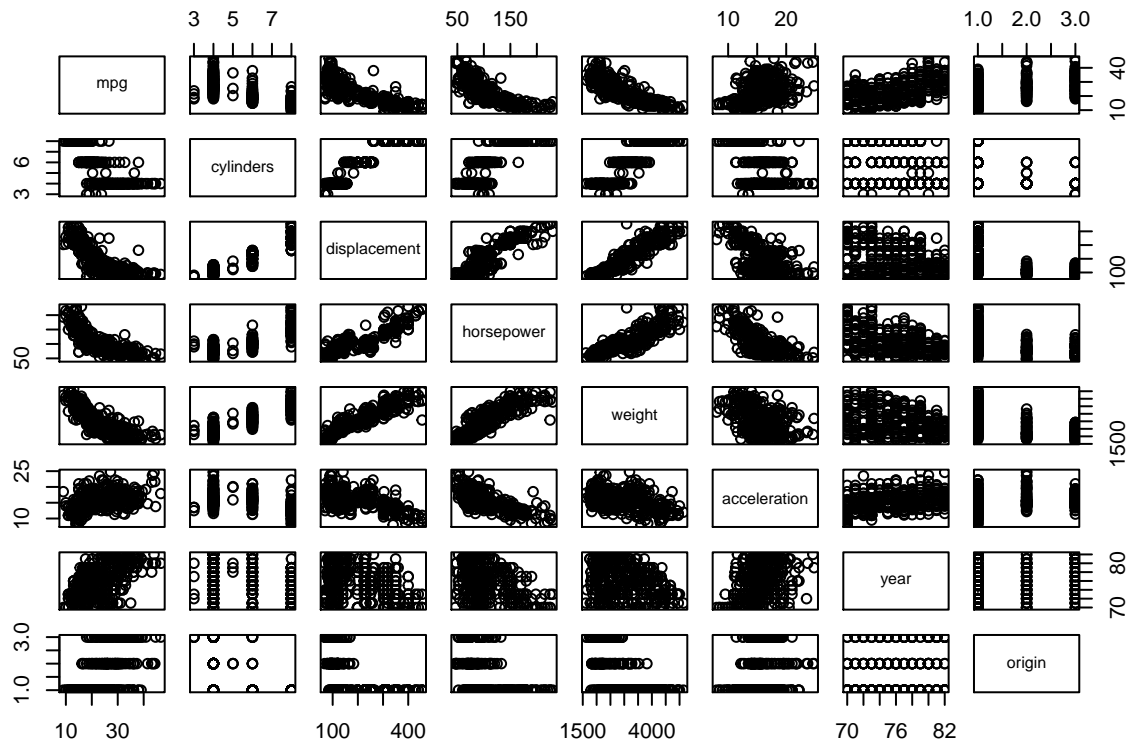
Finally, KNN classifier applies Bayes rule and classifies the test observation  $x_0$  to the class with the largest probability.

## Applied Questions

### Question 9

This question focuses on standard regression procedures.

```
Auto = read.csv("~/Desktop/R_Notes/R_ISLR/ISLR_data/Auto.csv", header=T, na.strings="?")
Auto = na.omit(Auto)
pairs(Auto[, -9])
```



```
cor(subset(Auto, select=-name))
```

```
##           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
```

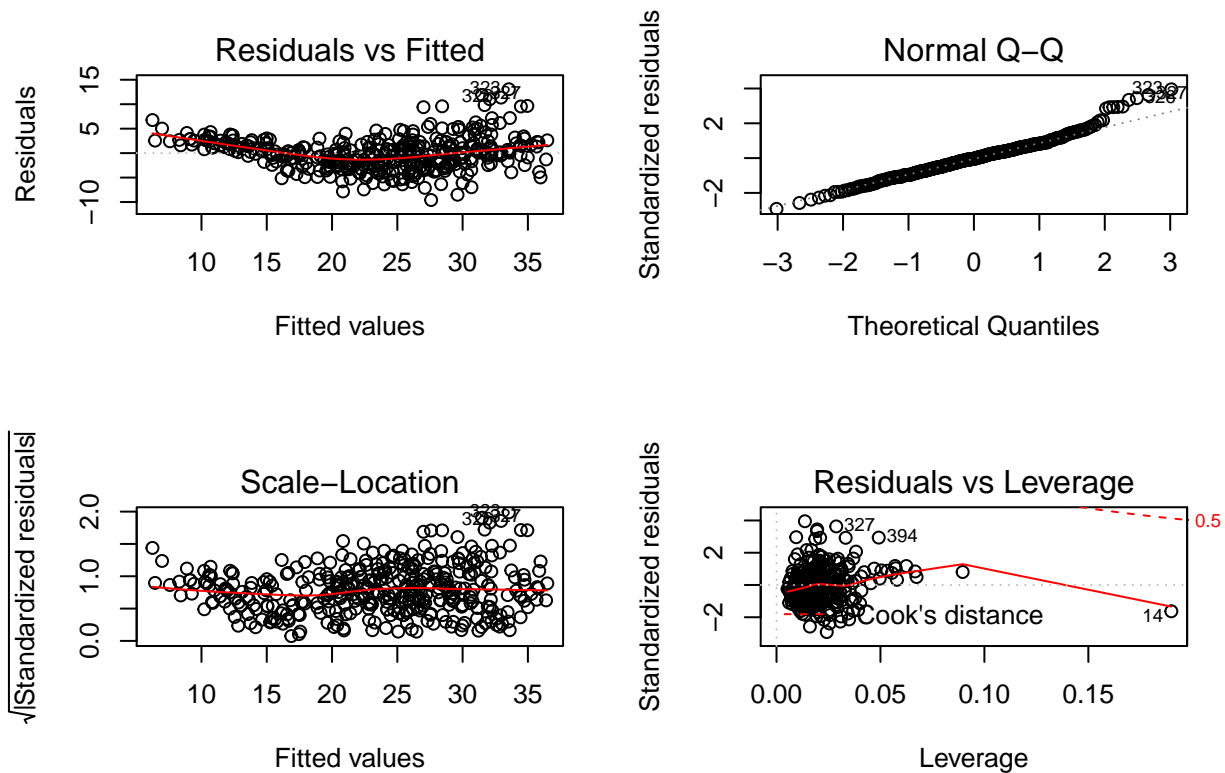
```
lm.fit0 <- lm(mpg ~ . -name, data=Auto)
summary(lm.fit0)
```

```
##
## 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
```

Overall, the model supports a relationship between predictors and the response, as suggested by the low p-value from the F test. Of the seven variables (excluding the intercept), **displacement**, **weight**, **year** and **origin** have statistically significant effects on **mpg** while **cylinders**, **horsepower**, and **acceleration** do not. The variable, **year**, indicates that, for every one year, **mpg** increases by a positive 0.7507727. In other words, cars become more fuel-efficient every year.

```
par(mfrow=c(2,2))
plot(lm.fit0)
```



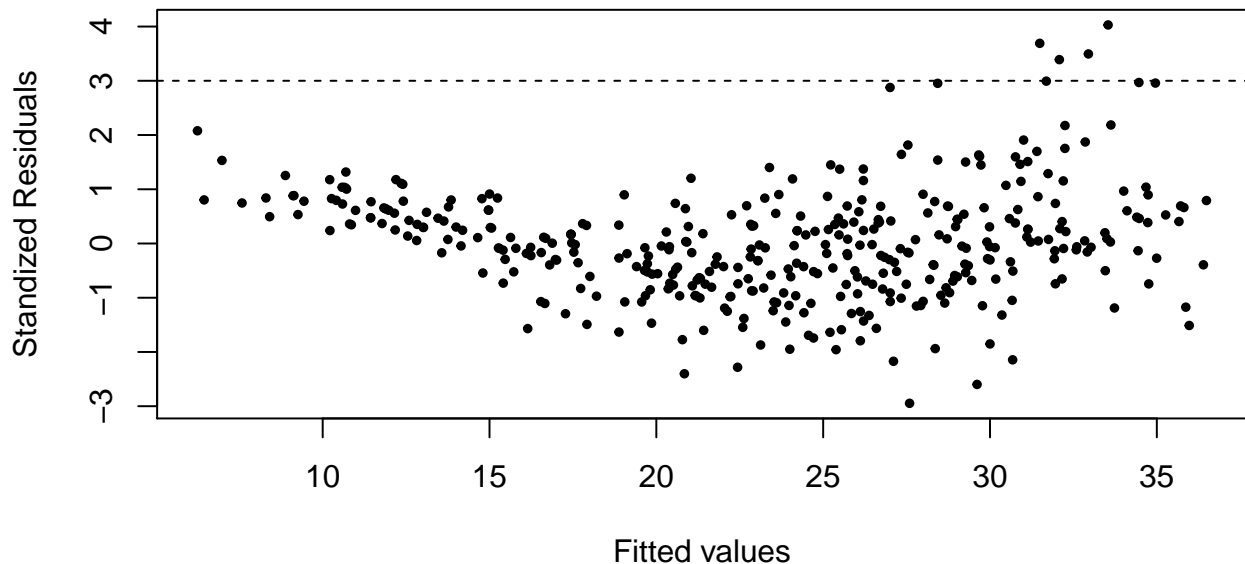
From the Residual vs Fitted plot, there seems to be a quadratic relationship between the residuals and the fitted values, suggesting that non-linearity between the predictors and response. Polynomial regression or non-linear transformation such as interaction of the variables may be needed.

The Scale-Location plot, also known as the Spread-Location plot, shows if residuals are spread equally along the ranges of predictors. Homoscedasticity is questionable here as indicated by the non-horizontal line—unequally (instead of randomly) spread points.

The QQ plot displays a steeper slope on the right tail, implying a positive skewness of the residuals.

The Residual vs Leverage plot suggests that `buick estate wagon (sw)` (observation 14) has high leverage, despite low magnitude residuals.

```
plot(predict(lm.fit0), rstudent(lm.fit0),
     pch = 20, cex = 0.8,
     xlab = 'Fitted values',
     ylab = 'Standardized Residuals')
abline(3, 0, lty=2)
```



There are possible outliers as seen in the plot of studentized residuals as suggested by the presence of datapoints with a value greater than 3.

```
# Interaction Terms
lm.fit0 <- lm(mpg ~ . - name, data=Auto)
lm.fit1 <- lm(mpg~cylinders+weight*cylinders+year+origin, data=Auto)
# lm.fit2 <- lm(mpg~acceleration+weight*acceleration+year+origin, data=Auto)
# lm.fit3 <- lm(mpg~horsepower+weight*horsepower+year+origin, data=Auto)
summary(lm.fit0)
```

```
##
## 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
```

```
summary(lm.fit1)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + weight * cylinders + year + origin,
##     data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.5277  -1.7587  -0.2015   1.5147  12.7885
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.8845201   4.5378720   0.856  0.3925
## cylinders      -4.4875789   0.5639369  -7.958 1.97e-14 ***
## weight        -0.0144268   0.0010804 -13.353  < 2e-16 ***
## year           0.8115738   0.0455113  17.832  < 2e-16 ***
## origin         0.5345940   0.2506425   2.133  0.0336 *
## cylinders:weight 0.0013967   0.0001637   8.533 3.30e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.078 on 386 degrees of freedom
## Multiple R-squared:  0.8464, Adjusted R-squared:  0.8444
## F-statistic: 425.5 on 5 and 386 DF,  p-value: < 2.2e-16
```

```
# summary(lm.fit2)
```

```
# summary(lm.fit3)
```

Insignificant variables's effect to mpg maybe captured by *synergy* or interaction terms. Interaction bewteen weight and cylinders (lm.fit1), bewteen weight and acceleration (lm.fit2), bewteen weight and and horsepower (lm.fit3) are all statistically significant.

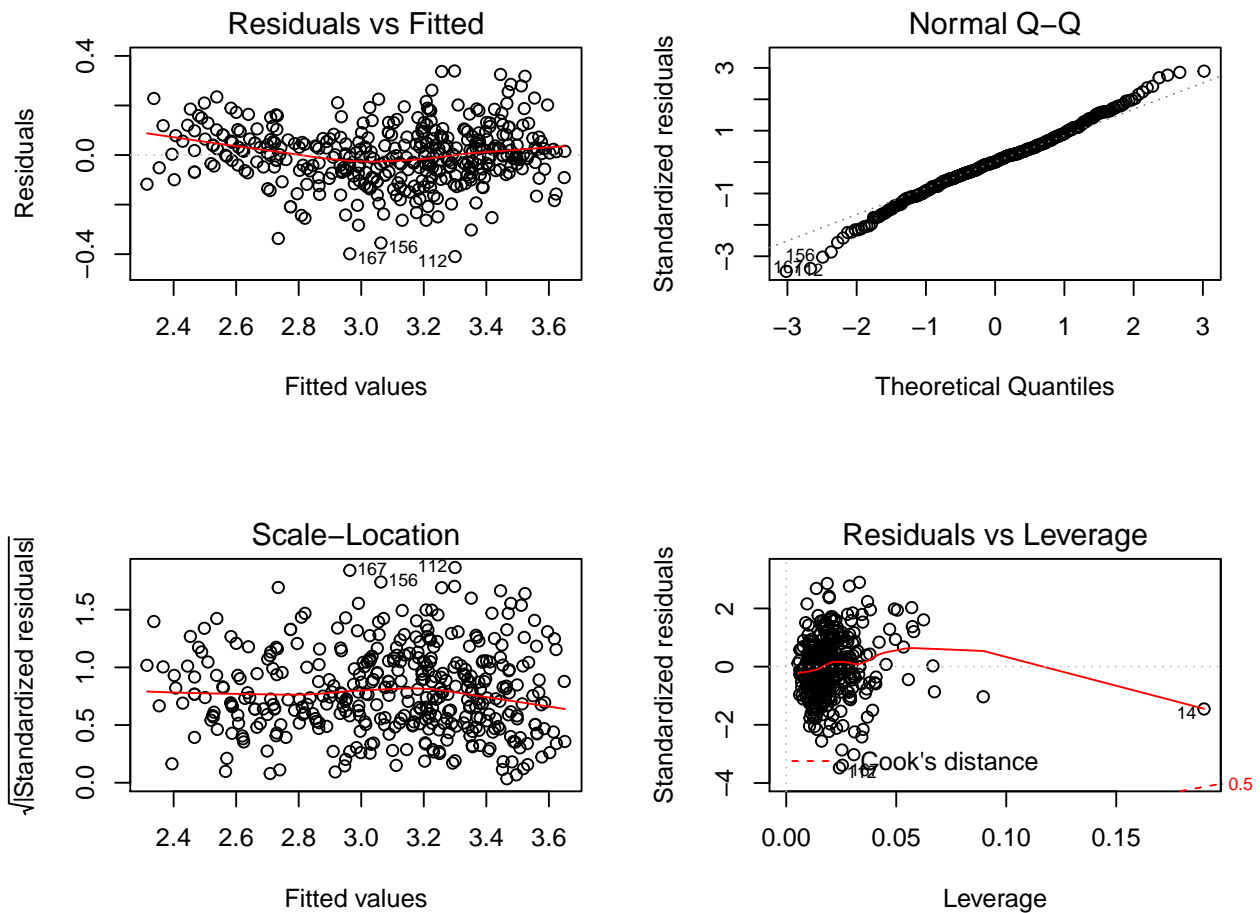
```
# Non-linear Transformations of the Predictors
```

```
lm.fit4 <- lm(log(mpg)~cylinders+displacement+horsepower+weight+acceleration+year+origin,data=Auto)
summary(lm.fit4)
```

```
##
## Call:
## lm(formula = log(mpg) ~ cylinders + displacement + horsepower +
##     weight + acceleration + year + origin, data = Auto)
##
## Residuals:
```

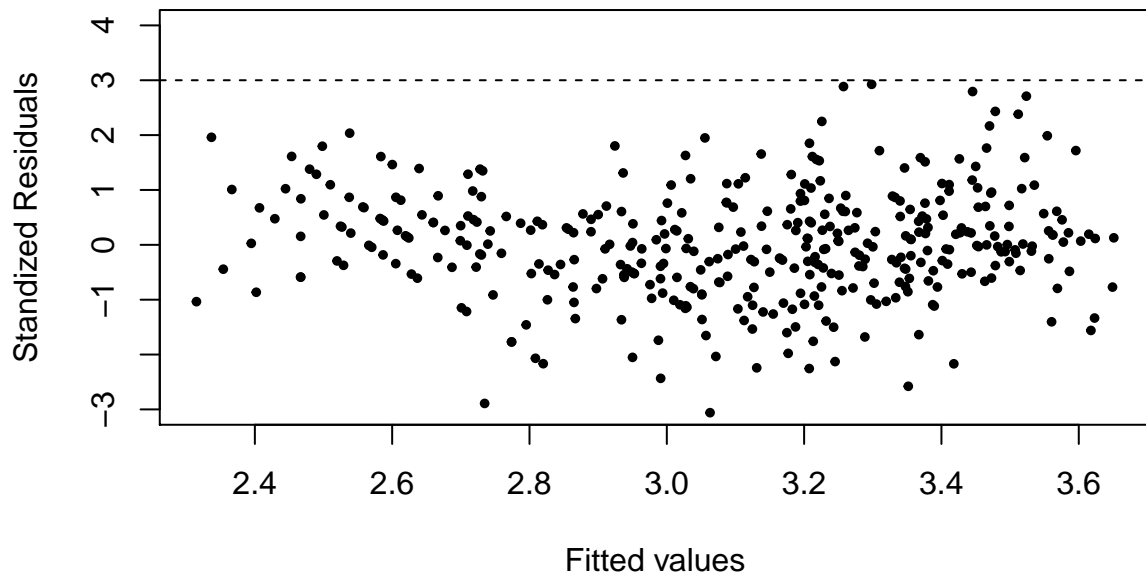
```
##      Min      1Q   Median      3Q      Max
## -0.40955 -0.06533  0.00079  0.06785  0.33925
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.751e+00  1.662e-01  10.533 < 2e-16 ***
## cylinders    -2.795e-02  1.157e-02  -2.415  0.01619 *
## displacement  6.362e-04  2.690e-04   2.365  0.01852 *
## horsepower   -1.475e-03  4.935e-04  -2.989  0.00298 **
## weight       -2.551e-04  2.334e-05 -10.931 < 2e-16 ***
## acceleration -1.348e-03  3.538e-03  -0.381  0.70339
## year         2.958e-02  1.824e-03  16.211 < 2e-16 ***
## origin       4.071e-02  9.955e-03   4.089  5.28e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1191 on 384 degrees of freedom
## Multiple R-squared:  0.8795, Adjusted R-squared:  0.8773
## F-statistic: 400.4 on 7 and 384 DF, p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(lm.fit4)
```



```
par(mfrow=c(1,1))
plot(predict(lm.fit4), rstudent(lm.fit4), pch = 20, cex = 0.8,
      xlab = 'Fitted values',
```

```
ylab = 'Standized Residuals',
ylim = c(-3, 4))
abline(3, 0, lty=2)
```



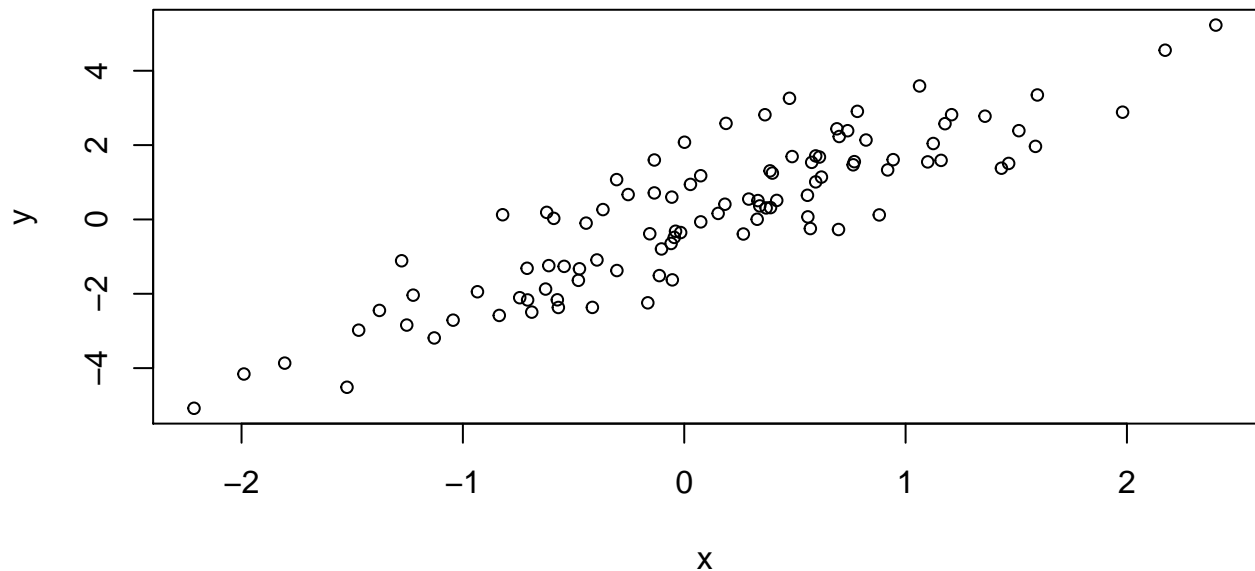
As indicated by the Residual vs Fitted plot, QQ plot and the Scale-Location plot, heteroskedasticity appears to be a feature in the previous model with linear predictors. Also in the scatter matrix, `displacement`, `horsepower` and `weight` show a similar nonlinear pattern against response `mpg`. Non-linear transformations of the predictors may be appropriate. Using `log(mpg)` as the response variable, the outputs show that log transform of `mpg` yield a higher  $R^2$  and residuals more normally distributed.

### Question 11

Questions 5, 11 and 12 ask about simple linear regression without an intercept.

```
set.seed(1)
x=rnorm(100)
y=2*x+rnorm(100)
plot(x,y, cex=0.8)
```





*# Regress y on x. Result is highly significant*

```
lm.fit0 <- lm(y~x+0)
summary(lm.fit0)
```

```
##
## 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
```

*# Regress x on y. Result is highly significant*

```
lm.fit1 <- lm(x~y+0)
summary(lm.fit1)
```

```
##
## 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
```

First, the multiple  $R^2$ , adjusted  $R^2$ , t-statistics, and F-statistics are the same in the two models.

Second, since  $\hat{x} = \hat{\beta}_x y$  versus  $\hat{y} = \hat{\beta}_y x$ , so the betas should be inverse of each other ( $\hat{\beta}_x = \frac{1}{\hat{\beta}_y}$ ) but they are somewhat off here ( $\frac{1}{0.39111} = 2.557 \neq 1.994$ ).

```
lm.fit = lm(y~x)
lm.fit2 = lm(x~y)
summary(lm.fit)
```

```
##
## 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
```

```
summary(lm.fit2)
```

```
##
## 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
```

The t-statistics are the same.

## Question 12

Generate an example in R with  $n = 100$  observations in which the coefficient estimate for the regression of  $X$  onto  $Y$  is different from the coefficient estimate for the regression of  $Y$  onto  $X$ .

*# Question 11a is the example in point.*

Focus on the denominator in equation 3.38. If  $\sum(x_i^2) = \sum(y_i^2)$ ,  $\hat{\beta}$  of regressing  $y$  on  $x$  will be equal to that of regressing  $x$  on  $y$ . To illustrate, see

```
set.seed(1)
x <- rnorm(100)
# Generate random sample (i.e. y ) from x without replacement
y <- -sample(x, 100)
# suh that:
sum(x^2)==sum(y^2)

## [1] TRUE

lm.fit_x <- lm(y~x+0)
lm.fit_y <- lm(x~y+0)
summary(lm.fit_x)

##
## Call:
## lm(formula = y ~ x + 0)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.3926 -0.6877 -0.1027  0.5124  2.2315
##
## Coefficients:
##      Estimate Std. Error t value Pr(>|t|)
## x -0.02148     0.10048  -0.214   0.831
##
## Residual standard error: 0.9046 on 99 degrees of freedom
## Multiple R-squared:  0.0004614, Adjusted R-squared: -0.009635
## F-statistic: 0.0457 on 1 and 99 DF, p-value: 0.8312

summary(lm.fit_y)

##
## Call:
## lm(formula = x ~ y + 0)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.2400 -0.5154  0.1213  0.6788  2.3959
##
## Coefficients:
##      Estimate Std. Error t value Pr(>|t|)
## y -0.02148     0.10048  -0.214   0.831
##
## Residual standard error: 0.9046 on 99 degrees of freedom
## Multiple R-squared:  0.0004614, Adjusted R-squared: -0.009635
## F-statistic: 0.0457 on 1 and 99 DF, p-value: 0.8312
```

## Question 14

This problem focuses on collinearity.

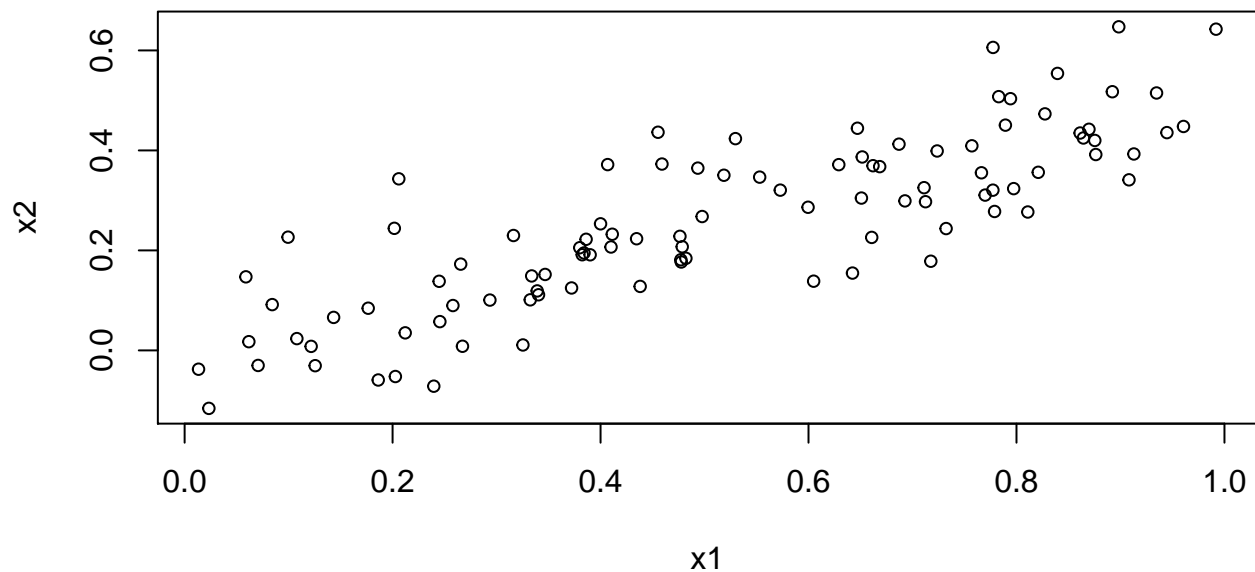
```
set.seed(1)
x1=runif(100)
x2 =0.5*x1+rnorm(100)/10
y=2+2*x1 +0.3*x2+rnorm(100)
```

The form of the linear model is  $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \varepsilon$  where  $\beta_0 = 2$ ,  $\beta_1 = 2$  and  $\beta_2 = 0.3$ .

```
cor(x1,x2)
```

```
## [1] 0.8351212
```

```
plot(x1,x2, xlab = 'x1', ylab = 'x2', cex=0.8)
```



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

```
##
## 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
```

Estimated beta coefficients are  $\hat{\beta}_0 = 2.13$ ,  $\hat{\beta}_1 = 1.44$  and  $\hat{\beta}_2 = 1.01$ . Coefficient for  $x_1$  is statistically significant but the coefficient for  $x_2$  is not. Null hypothesis for  $x_1$ ,  $H_0 : \beta_1 = 0$ , is rejected at 0.01 significant level while that of  $x_2$ ,  $H_0 : \beta_2 = 0$ , is retained.

```
par(mfrow=c(2,1), mar=c(2, 3, 2, 1), mgp=c(2, 0.8, 0))
lm.fit1 <- lm(y~x1)
summary(lm.fit1)
```

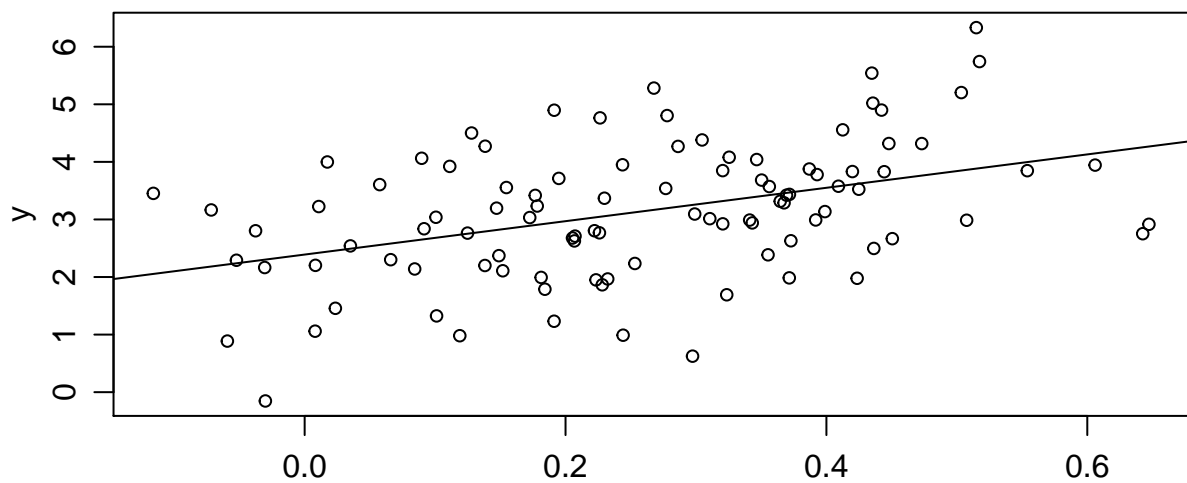
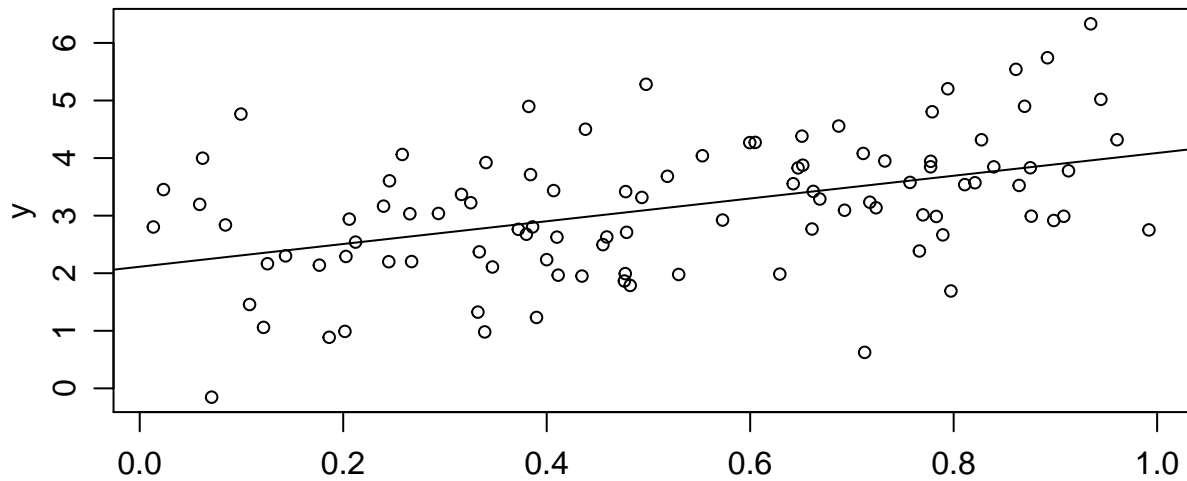
```
##
## Call:
## lm(formula = y ~ x1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.89495 -0.66874 -0.07785  0.59221  2.45560
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.1124     0.2307   9.155 8.27e-15 ***
## x1             1.9759     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
```

```
plot(x1,y, cex=0.8)
abline(lm.fit1)
```

```
lm.fit2 <- lm(y~x2)
summary(lm.fit2)
```

```
##
## 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
```

```
plot(x2,y, cex=0.8)
abline(lm.fit2)
```



Individually, both  $x_1$  and  $x_2$  enter the simple regression model with highly significant statistical levels.

There is no contradiction. The problem lies in collinearity. It is hard to distinguish their individual effects from the combined effects when regressed upon together.

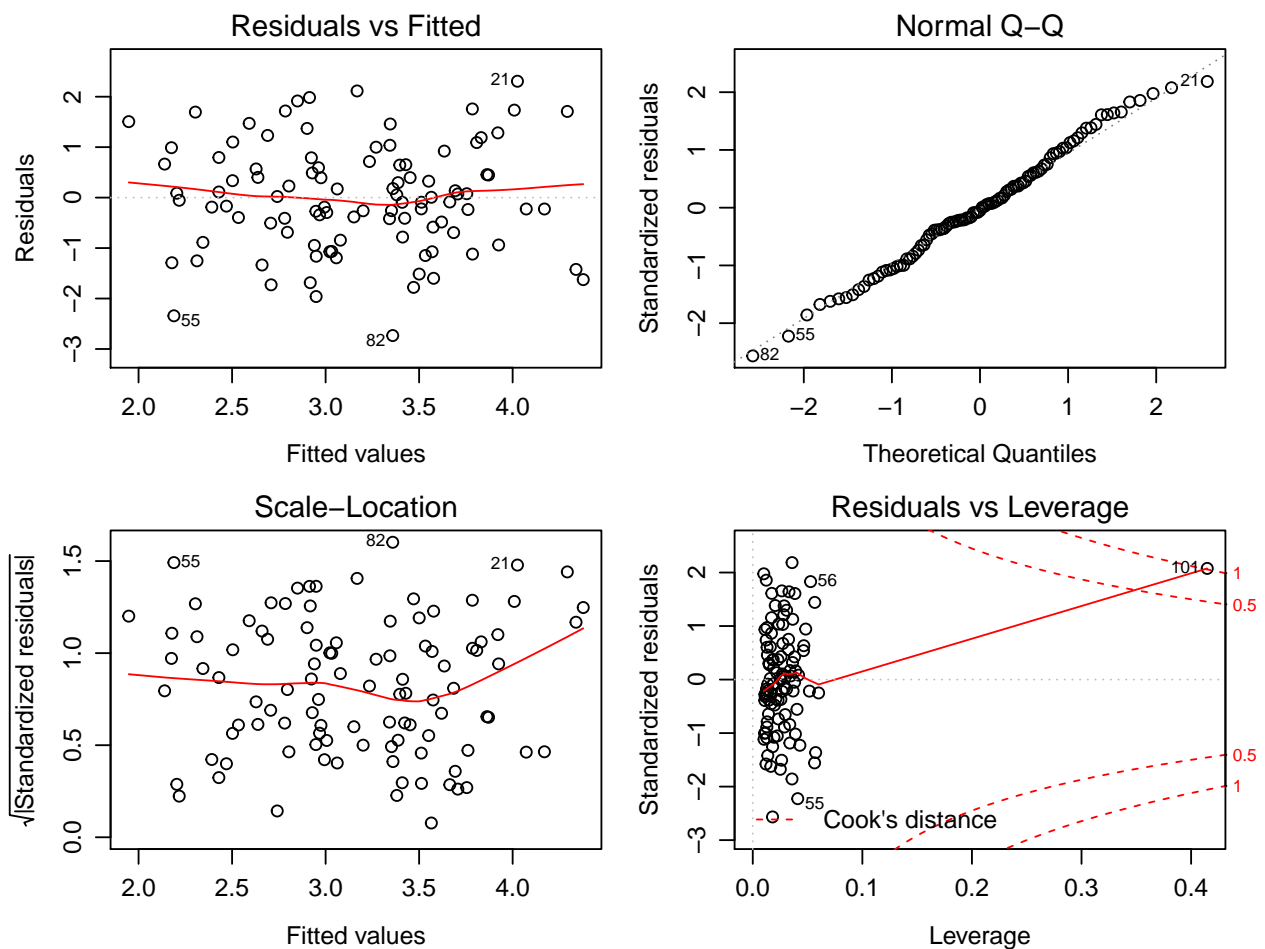
```
x1=c(x1, 0.1)
x2=c(x2, 0.8)
y=c(y,6)
par(mfrow=c(2,2), mar=c(3.5, 3.5, 2, 1), mgp=c(2.4, 0.8, 0))
# regression with both x1 and x2
fit.lm <- lm(y~x1+x2)
summary(fit.lm)
```

```
##
## 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
```

```
plot(fit.lm)
```

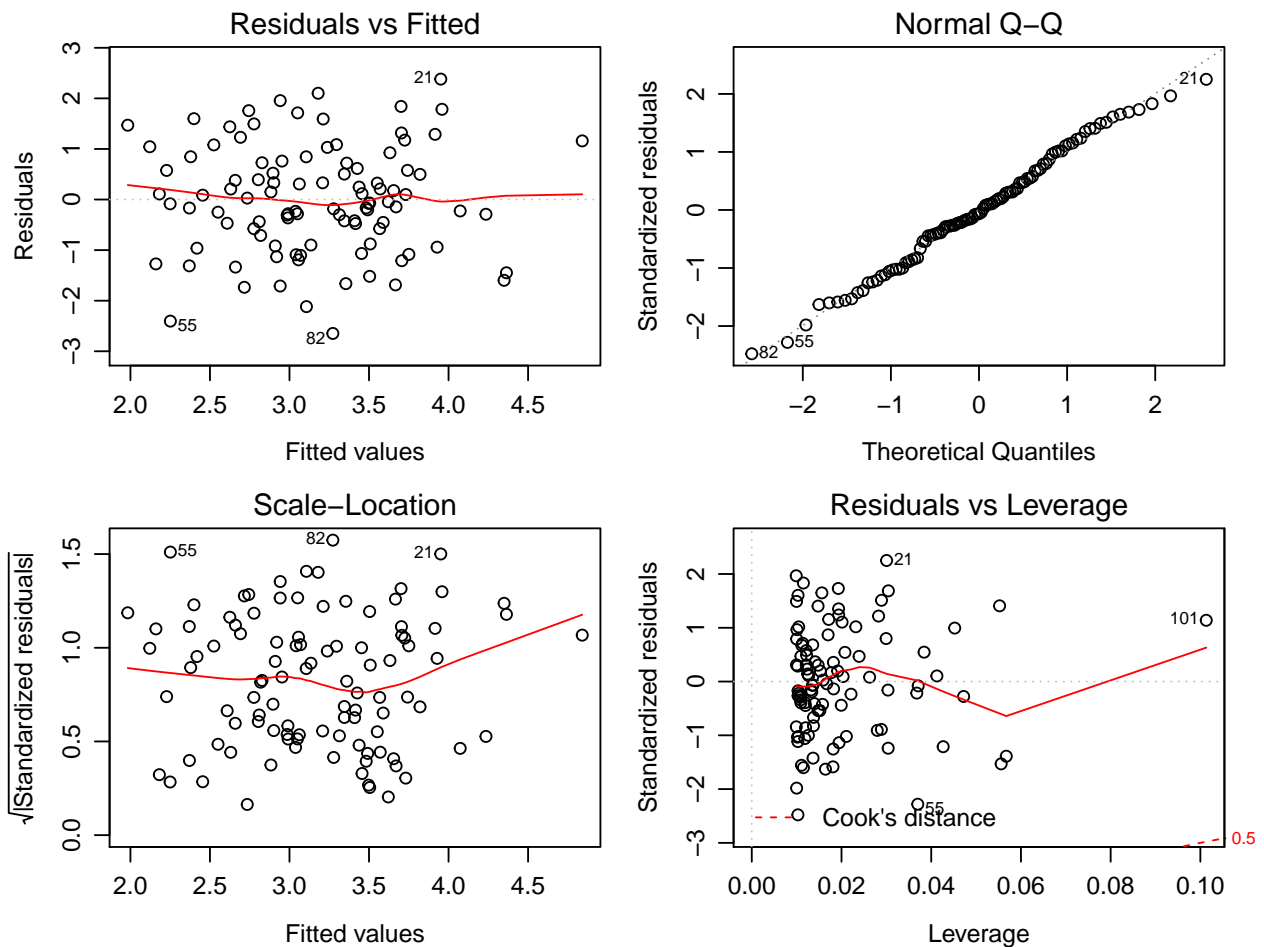


```
# regression with x1 only
fit.lm1 <- lm(y~x2)
summary(fit.lm1)
```

```
##
## 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(fit.lm1)
```

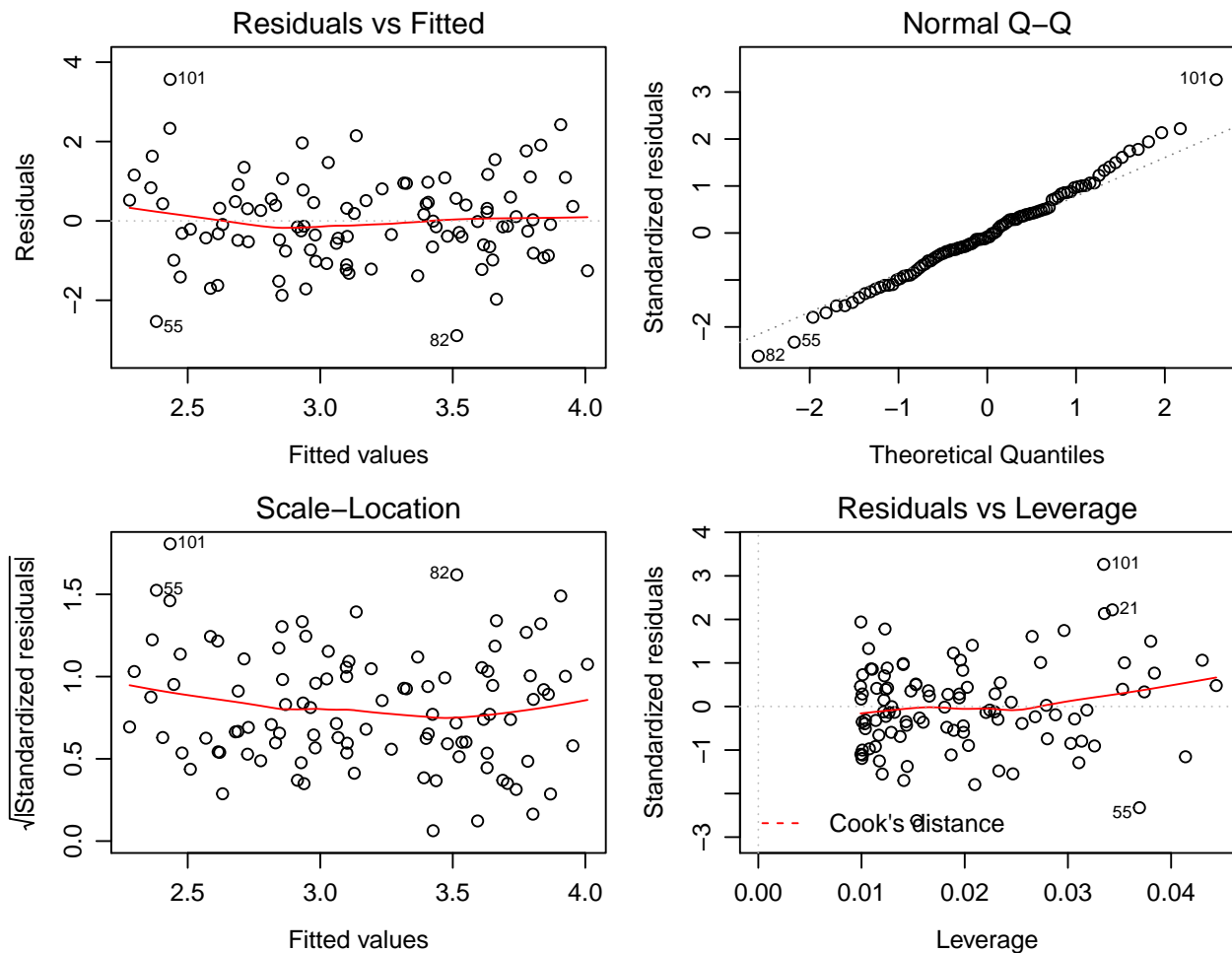


```
# regression with x2 only
fit.lm2 <- lm(y~x1)
summary(fit.lm2)
```

```
##
## 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
plot(fit.lm2)
```



The new observation ( $[y, x_1, x_2] = [6, 0.1, 0.8]$ ) is an outlier for  $x_2$  and has high leverage for both  $x_1$  and  $x_2$ . From the residuals vs leverage plot, observation 101 falls on the right hand side in all three models. In particular, it stands out as the red line is extensively tilted relative to the dotted black line indicating high leverage (Cook's Distance) for the model in which  $x_1$  and  $x_2$  are the predictors of  $y$ .