Problem Set 1

Tom Hanna

9/28/2020

##Chapter 2

Answer 1:

```
a) A more flexible approach will give a better fit. With the larger sample size, there is less concern f
b) With the smaller sample size, which implies more noise, there is an expectation that using a large nu
c) Since the relationship is non-linear, a flexible approach is needed to better fit the data and it is
d) This is a classic case of a high noise to signal ratio, so a flexible approach will result in overfi
```

Answer 7:

```
a) obs 1: 3
   obs 2: 2
   obs 3: sqrt(1^2 + 3^2) = sqrt(10)
   obs 4: sqrt(1^2 + 2^2) = sqrt(5)
   obs 5: sqrt(-1^2 + 1^2) = sqrt(2)
   obs 6: sqrt(1^2 + 1^2 + 1^2) = sqrt(3)
```

b) The nearest neighbor with a distance sqrt(2) is observation 5, Green.

c) The three nearest neighbors with distance sqrt(2), 2, and sqrt(3) are observations 5, 2, and 6. Green, Red, and Red.
d) Small. A higher value for K would produce a less flexible, more linear boundary (p.40 in the text).

##Chapter 3

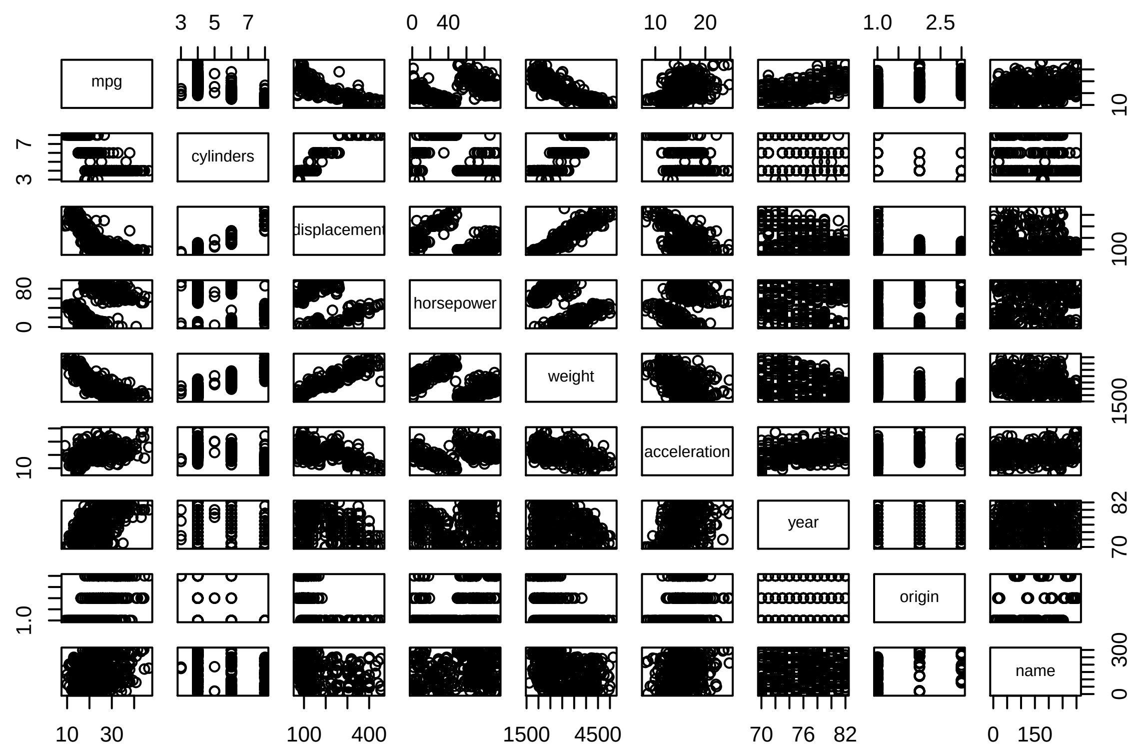
Answer 3:

```
a) iii - The coefficient for the interaction terms show that males earn more than females with the same
b) Salary = 50 + 20*4 + 35*1 + .07*110 + .01*110*4 + 10*4*1 = 50+80+35+7.7+4.4-40 = 137.1 or $137,100
c_ False. We would have to know the standard error, so we could compute significance. If the standard e
```

```r
gc()
```

```
##          used (Mb) gc trigger (Mb) max used (Mb)
## Ncells 398464 21.3     819144 43.8   638648 34.2
## Vcells 735048  5.7    8388608 64.0  1632854 12.5
```

```
rm(list=ls())
options(scipen = 999)


# Chapter 3 Lab: Linear Regression
setwd("C:/R Studio Files/POLS6394-Machine-Learning/Lab 3")
library(MASS)
library(ISLR)


Auto <- read.csv("C:/R Studio Files/POLS6394-Machine-Learning/datasets/Auto.csv")
View(Auto)

#I was getting an error in pairs because of the nonnumeric vectors horsepower
#and names, so I used the data.matrix command to convert everything to numeric.
#Note that as.numeric(horsepower) produces different results than horsepower,
#because there are three NA rows.

AutoMatrix <- data.matrix(Auto, rownames.force = NA)

#Problem 9

#a

pairs(AutoMatrix)
```



```r
#b

cor(AutoMatrix)
```

```
##                     mpg  cylinders displacement horsepower     weight
## mpg           1.0000000 -0.7762599   -0.8044430  0.4228227 -0.8317389
## cylinders    -0.7762599  1.0000000    0.9509199 -0.5466585  0.8970169
## displacement -0.8044430  0.9509199    1.0000000 -0.4820705  0.9331044
## horsepower    0.4228227 -0.5466585   -0.4820705  1.0000000 -0.4821507
## weight       -0.8317389  0.8970169    0.9331044 -0.4821507  1.0000000
## acceleration  0.4222974 -0.5040606   -0.5441618  0.2662877 -0.4195023
## year          0.5814695 -0.3467172   -0.3698041  0.1274167 -0.3079004
## origin        0.5636979 -0.5649716   -0.6106643  0.2973734 -0.5812652
## name          0.2745323 -0.2803461   -0.2946560  0.1600054 -0.2557389
##              acceleration        year     origin        name
## mpg             0.4222974  0.58146946  0.5636979  0.27453225
## cylinders      -0.5040606 -0.34671722 -0.5649716 -0.28034613
## displacement   -0.5441618 -0.36980409 -0.6106643 -0.29465598
## horsepower      0.2662877  0.12741665  0.2973734  0.16000542
## weight         -0.4195023 -0.30790041 -0.5812652 -0.25573888
## acceleration    1.0000000  0.28290089  0.2100836  0.13647687
## year            0.2829009  1.00000000  0.1843141  0.08185952
## origin          0.2100836  0.18431408  1.0000000  0.35854033
## name            0.1364769  0.08185952  0.3585403  1.00000000
```

```r
Auto$hp.num <- as.numeric(Auto$horsepower)
```

```
## Warning: NAs introduced by coercion
```

```r
names(Auto)
```

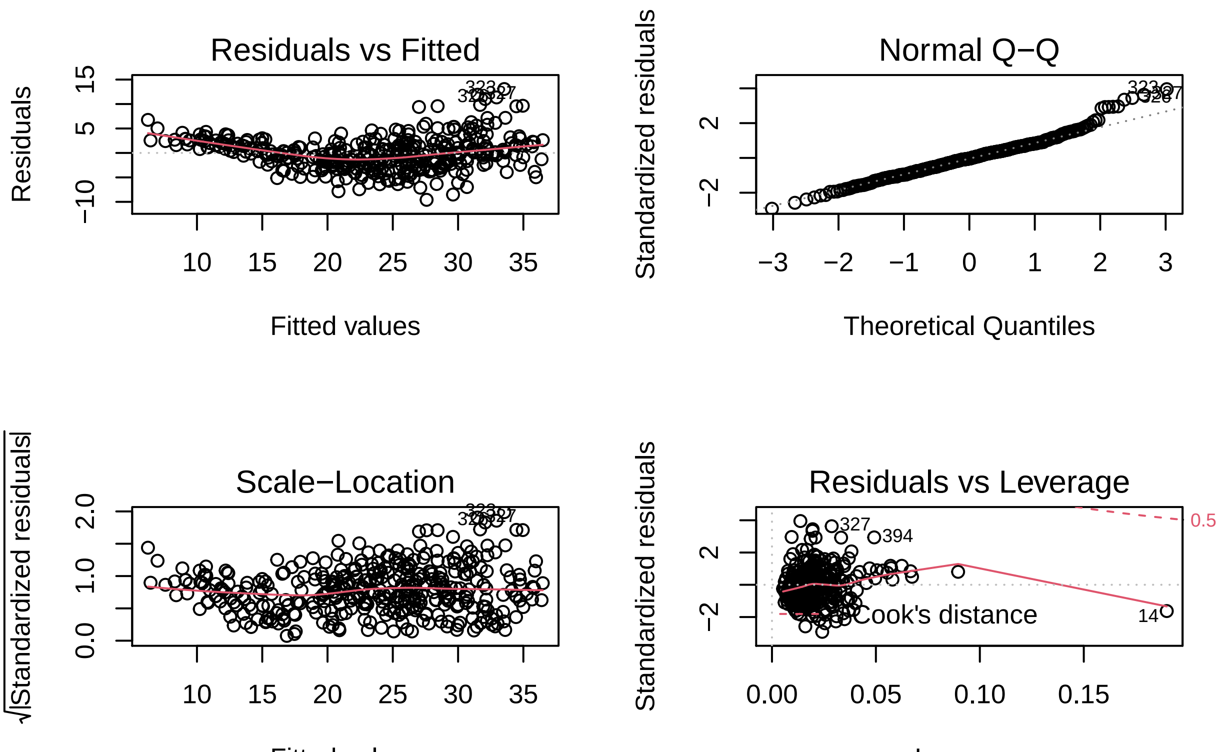
```
##  [1] "mpg"          "cylinders"    "displacement" "horsepower"   "weight"
##  [6] "acceleration" "year"         "origin"       "name"         "hp.num"
```

```r
#c

modelc <- lm(mpg ~ cylinders + displacement + hp.num + weight + acceleration + year + origin, data = Au
summary(modelc)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + hp.num + weight +
##     acceleration + year + origin, 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 **
## hp.num        -0.016951   0.013787  -1.230              0.21963
## weight        -0.006474   0.000652  -9.929 < 0.0000000000000002 ***
## acceleration   0.080576   0.098845   0.815              0.41548
## year           0.750773   0.050973  14.729 < 0.0000000000000002 ***
## origin         1.426141   0.278136   5.127           0.000000467 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.328 on 384 degrees of freedom
##   (5 observations deleted due to missingness)
## Multiple R-squared:  0.8215, Adjusted R-squared:  0.8182
## F-statistic: 252.4 on 7 and 384 DF,  p-value: < 0.00000000000000022
```

```
#Intercept, displacement, weight, year, and origin are statistically significant to the .05 level or hi

#d

par(mfrow=c(2,2))
plot(modelc)
```




```
#e

#Cylinders and displacement are related design factors, with larger engines typically having more cylin
```

```r
modelcyldisp <- lm(mpg ~ cylinders*displacement + hp.num + weight + acceleration + year + origin, data =
summary(modelcyldisp)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders * displacement + hp.num + weight +
##     acceleration + year + origin, data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.6081  -1.7833  -0.0465   1.6821  12.2617
##
## Coefficients:
##                         Estimate Std. Error t value             Pr(>|t|)
## (Intercept)           -2.7096590  4.6858582  -0.578             0.563426
## cylinders             -2.6962123  0.4094916  -6.584   0.0000000001509175 ***
## displacement          -0.0774797  0.0141535  -5.474   0.0000000796120535 ***
## hp.num                -0.0476026  0.0133736  -3.559             0.000418 ***
## weight                -0.0052339  0.0006253  -8.370   0.0000000000000011 ***
## acceleration           0.0597997  0.0918038   0.651             0.515188
## year                   0.7594500  0.0473354  16.044 < 0.0000000000000002 ***
## origin                 0.7087399  0.2736917   2.590             0.009976 **
## cylinders:displacement  0.0136081  0.0017209   7.907   0.0000000000000284 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.089 on 383 degrees of freedom
##   (5 observations deleted due to missingness)
## Multiple R-squared:  0.8465, Adjusted R-squared:  0.8433
## F-statistic: 264.1 on 8 and 383 DF,  p-value: < 0.00000000000000022
```

```r
#The interaction is statistically significant. Interestingly, without the interaction effect increasing

#Horsepower has an odd distribution in the scatterplot. I suspect this has to do with an interaction be

modelwthp <- lm(mpg ~ weight + hp.num + cylinders + displacement + acceleration + year + origin + weigh
summary(modelwthp)
```

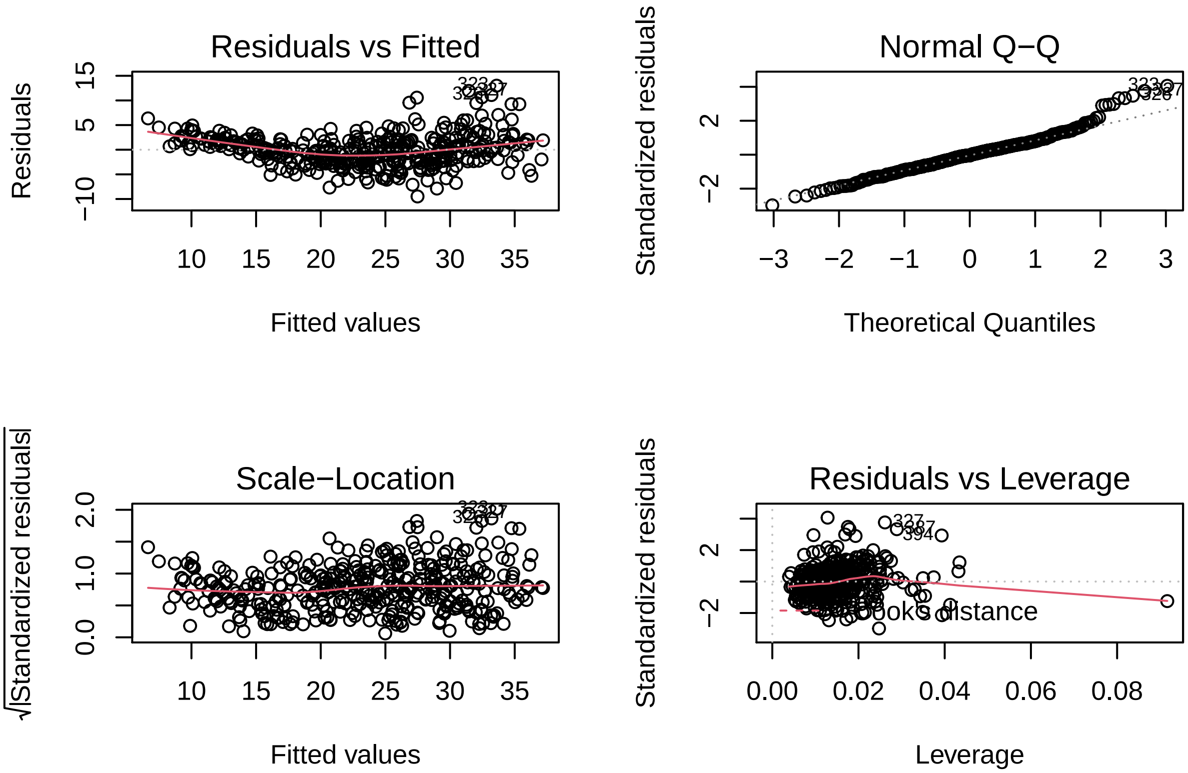
```
##
## Call:
## lm(formula = mpg ~ weight + hp.num + cylinders + displacement +
##     acceleration + year + origin + weight:hp.num, data = Auto)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.589 -1.617 -0.184  1.541 12.001
##
## Coefficients:
##                  Estimate   Std. Error t value             Pr(>|t|)
## (Intercept)    2.875748260  4.510615754   0.638             0.524147
## weight        -0.011214651  0.000728542 -15.393 < 0.0000000000000002 ***
## hp.num        -0.231326725  0.023627408  -9.791 < 0.0000000000000002 ***
```

```
## cylinders     -0.029551410  0.288128191  -0.103             0.918363
## displacement   0.005949890  0.006749875   0.881             0.378610
## acceleration  -0.090193021  0.088554342  -1.019             0.309081
## year           0.769461261  0.044935777  17.124 < 0.0000000000000002 ***
## origin         0.834401609  0.251309454   3.320             0.000986 ***
## weight:hp.num  0.000055289  0.000005227  10.577 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.931 on 383 degrees of freedom
##   (5 observations deleted due to missingness)
## Multiple R-squared:  0.8618, Adjusted R-squared:  0.859
## F-statistic: 298.6 on 8 and 383 DF,  p-value: < 0.00000000000000022
```

```
#The effect is significant. Increased weight and increased horsepower reduce gas mileage all else being

#f

modeltrans1 <- lm(mpg ~ sqrt(displacement) + sqrt(weight) + acceleration^2 + year + origin, data = Auto)
summary(modeltrans1)
```

```
##
## Call:
## lm(formula = mpg ~ sqrt(displacement) + sqrt(weight) + acceleration^2 +
##     year + origin, data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.4849 -2.0411 -0.1352  1.7735 12.9975
##
## Coefficients:
##                    Estimate Std. Error t value             Pr(>|t|)
## (Intercept)        -0.48653    4.29209  -0.113             0.909808
## sqrt(displacement)  0.16304    0.15717   1.037             0.300230
## sqrt(weight)       -0.73977    0.06480 -11.417 < 0.0000000000000002 ***
## acceleration        0.10882    0.07357   1.479             0.139874
## year                0.76833    0.04742  16.203 < 0.0000000000000002 ***
## origin              1.04786    0.26940   3.890             0.000118 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.221 on 391 degrees of freedom
## Multiple R-squared:  0.8327, Adjusted R-squared:  0.8306
## F-statistic: 389.3 on 5 and 391 DF,  p-value: < 0.00000000000000022
```

```
par(mfrow=c(2,2))
plot(modeltrans1)
```

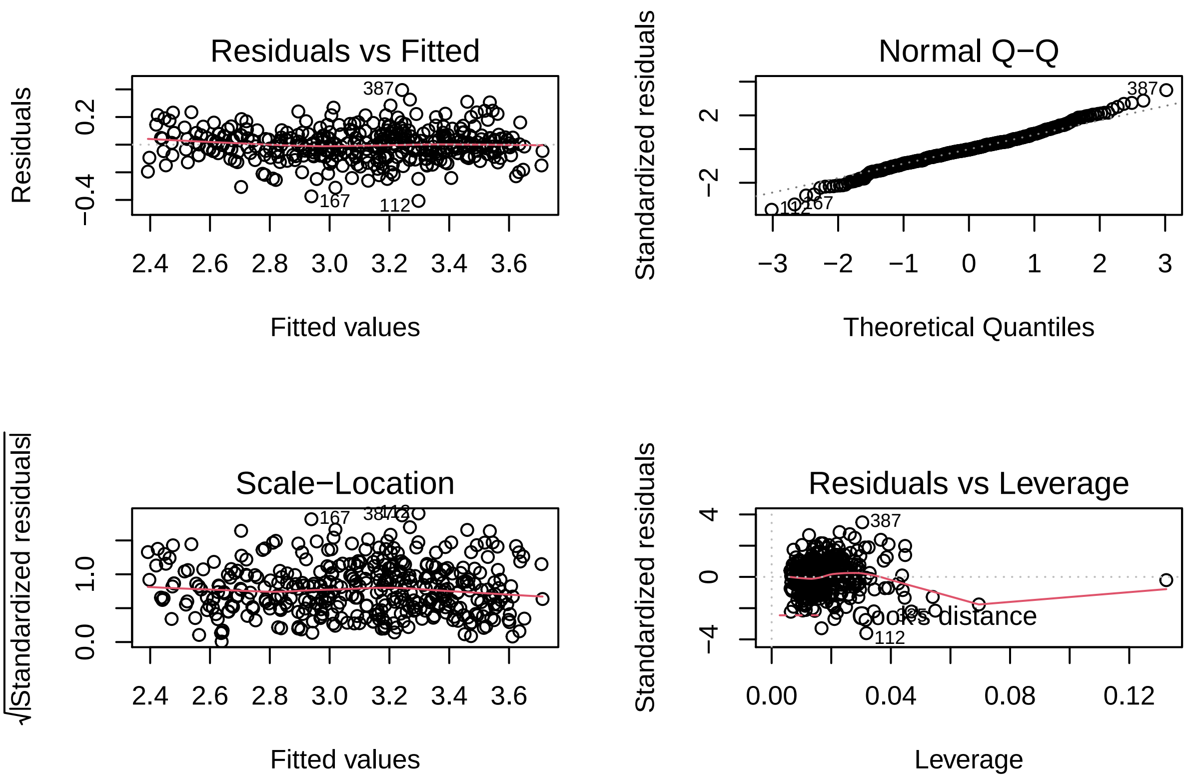



```r
#The transformation imporved R-squared marginally, but did not improve the residuals or leverage.

modeltrans2 <- lm(log(mpg) ~ sqrt(hp.num) + log(displacement) + log(weight) + acceleration^2 + year + o
summary(modeltrans2)
```

```
##
## Call:
## lm(formula = log(mpg) ~ sqrt(hp.num) + log(displacement) + log(weight) +
##     acceleration^2 + year + origin, data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.40695 -0.06689 -0.00362  0.06446  0.39517
##
## Coefficients:
##                    Estimate Std. Error t value             Pr(>|t|)
## (Intercept)        6.653933   0.406252  16.379 < 0.0000000000000002 ***
## sqrt(hp.num)      -0.039163   0.010589  -3.698             0.000248 ***
## log(displacement) -0.034423   0.039529  -0.871             0.384392
## log(weight)       -0.653994   0.077772  -8.409 0.000000000000000814 ***
## acceleration      -0.005451   0.003664  -1.488             0.137629
## year               0.029912   0.001765  16.952 < 0.0000000000000002 ***
## origin             0.020486   0.010214   2.006             0.045596 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.1148 on 385 degrees of freedom
##   (5 observations deleted due to missingness)
## Multiple R-squared:  0.8878, Adjusted R-squared:  0.886
## F-statistic: 507.5 on 6 and 385 DF,  p-value: < 0.00000000000000022
```

```
#This transformation improved R-squared further, made a small improvement to residuals, and made some i

par(mfrow=c(2,2))
plot(modeltrans2)
```




```
#9 - g

#These models run several hundred pages, so I am just providing the code and results


#model4 <- lm(mpg ~ cylinders*displacement*hp.num*weight*acceleration*year*orig#in + as.factor(name), d
#summary(model4)

#Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#Residual standard error: 0.466 on 1 degrees of freedom
#  (5 observations deleted due to missingness)
#Multiple R-squared:      1,    Adjusted R-squared:  0.9964
#F-statistic: 281.3 on 390 and 1 DF,  p-value: 0.04751
```

```r
#model4 used the "names" variable as.factor, which wasn't allowed in the other questions, but the quest

#model6 <- lm(mpg ~ cylinders*displacement*horsepower*weight*acceleration*year*origin, data = Auto)
#summary(model6)

#Residual standard error: 1.042 on 6 degrees of freedom
#Multiple R-squared:  0.9997,	Adjusted R-squared:  0.9823
#F-statistic: 57.25 on 390 and 6 DF,  p-value: 0.0000234

# Chapter 3 Lab: Linear Regression
setwd("C:/R Studio Files/POLS6394-Machine-Learning/Lab 3")
library(MASS)
library(ISLR)


##Problem 13

rm(list=ls())
options(scipen = 999)

set.seed(1735)

#a

x <- rnorm(100)

#b

eps <- rnorm(100,mean = 0,sd = sqrt(0.25))

#c

y <- -1 + 0.5*x + eps

#The vector length is 100. B_0 = -1 and B_1 = 0.5

#d

plot(x,y)

#There is a positive linear relationship between x and y, with what appears to be a normal distribution

#e

model13e <- lm(y ~ x)
summary(model13e)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -1.0409 -0.2836 -0.0099  0.3015  1.0016
##
## Coefficients:
##             Estimate Std. Error t value             Pr(>|t|)
## (Intercept) -1.05788    0.04464 -23.700 < 0.0000000000000002 ***
## x            0.39800    0.04663   8.534    0.000000000000181 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.442 on 98 degrees of freedom
## Multiple R-squared:  0.4264, Adjusted R-squared:  0.4205
## F-statistic: 72.84 on 1 and 98 DF,  p-value: 0.0000000000001813
```

```r
#The intercept, B_0, is fairly close to the original equation at -1.06. The B_1 coefficient, 0.4, is di

#f


plot(x, y)
abline(model13e, lwd=5, col=2)
abline(-1, 0.5, lwd=5, col=1)
legend(-1, legend = c("Model 13e", "Original equation"), col=2:1, lwd=5)

#g
x2 <- x^2

model13g <- lm(y ~ x + x2)
summary(model13g)
```

```
##
## Call:
## lm(formula = y ~ x + x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.03885 -0.27946 -0.01711  0.30906  1.02391
##
## Coefficients:
##             Estimate Std. Error t value             Pr(>|t|)
## (Intercept) -1.03729    0.05425 -19.119 < 0.0000000000000002 ***
## x            0.38933    0.04851   8.025     0.00000000000237 ***
## x2          -0.02373    0.03534  -0.671                0.504
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4433 on 97 degrees of freedom
## Multiple R-squared:  0.429,  Adjusted R-squared:  0.4172
## F-statistic: 36.44 on 2 and 97 DF,  p-value: 0.000000000001573
```

```r
plot(x, y)
abline(model13e, lwd=5, col=3)
abline(model13g, lwd=5, col=2)
```

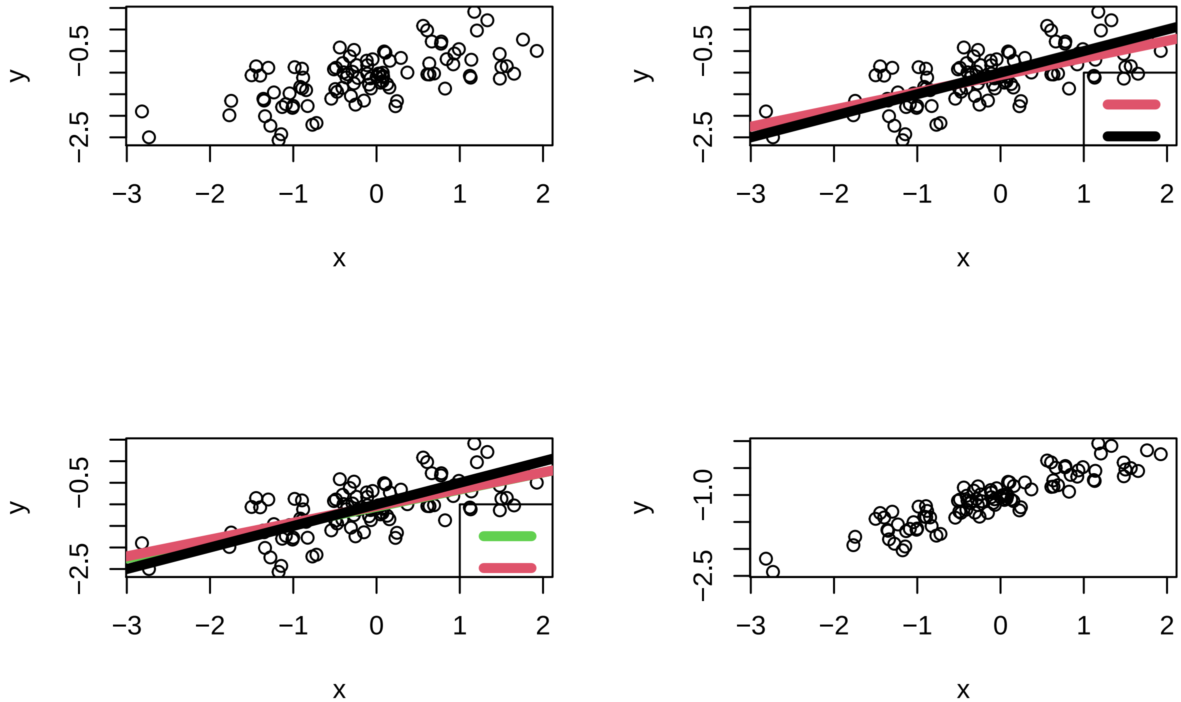
```
## Warning in abline(model13g, lwd = 5, col = 2): only using the first two of 3
```

```
## regression coefficients
```

```r
abline(-1, 0.5, lwd=5, col=1)
legend(-1, legend = c("Model 13e","Model 13g", "Original equation"), col=3:1, lwd=5)



#The R-squared, regular and adjusted, for the polynomial model is slightly lower than the simpler model
#The fitted lines are similar. The effect of X^2 is small in magnitude and not significant.

confint(model13e)
```

```
##                  2.5 %     97.5 %
## (Intercept) -1.1464580 -0.9692968
## x            0.3054538  0.4905429
```

```r
#h

rm(list=ls())
options(scipen = 999)

set.seed(1735)

#a

x <- rnorm(100)

#b

eps <- rnorm(100,mean = 0,sd = sqrt(0.05))

#c

y <- -1 + 0.5*x + eps

#The vector length is 100. B_0 = -1 and B_1 = 0.5

#d

plot(x,y)
```




```r
#There is a positive linear relationship between x and y, with what appears to be a normal distribution

#e

model13he <- lm(y ~ x)
summary(model13he)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.46550 -0.12684 -0.00443  0.13482  0.44793
##
## Coefficients:
##             Estimate Std. Error t value            Pr(>|t|)
## (Intercept) -1.02588    0.01996  -51.39 <0.0000000000000002 ***
## x            0.45438    0.02086   21.79 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1977 on 98 degrees of freedom
## Multiple R-squared:  0.8289, Adjusted R-squared:  0.8271
## F-statistic: 474.7 on 1 and 98 DF,  p-value: < 0.00000000000000022
```

```r
#The intercept, B_0, is closer to the original equation at -1.03. The B_1 coefficient, 0.45, is differe

#f


plot(x, y)
abline(model13he, lwd=5, col=2)
abline(-1, 0.5, lwd=5, col=1)
legend(-1, legend = c("Model 13e", "Original equation"), col=2:1, lwd=5)

#g
x2 <- x^2

model13hg <- lm(y ~ x + x2)
summary(model13hg)
```

```
##
## Call:
## lm(formula = y ~ x + x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.46459 -0.12498 -0.00765  0.13821  0.45791
##
## Coefficients:
##             Estimate Std. Error t value            Pr(>|t|)
## (Intercept) -1.01668    0.02426 -41.902 <0.0000000000000002 ***
## x            0.45051    0.02170  20.764 <0.0000000000000002 ***
## x2          -0.01061    0.01580  -0.671               0.504
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1982 on 97 degrees of freedom
## Multiple R-squared:  0.8297, Adjusted R-squared:  0.8262
## F-statistic: 236.2 on 2 and 97 DF,  p-value: < 0.00000000000000022
```

```r
plot(x, y)
abline(model13he, lwd=1, col=3)
abline(model13hg, lwd=1, col=2)
```

```
## Warning in abline(model13hg, lwd = 1, col = 2): only using the first two of 3
## regression coefficients
```

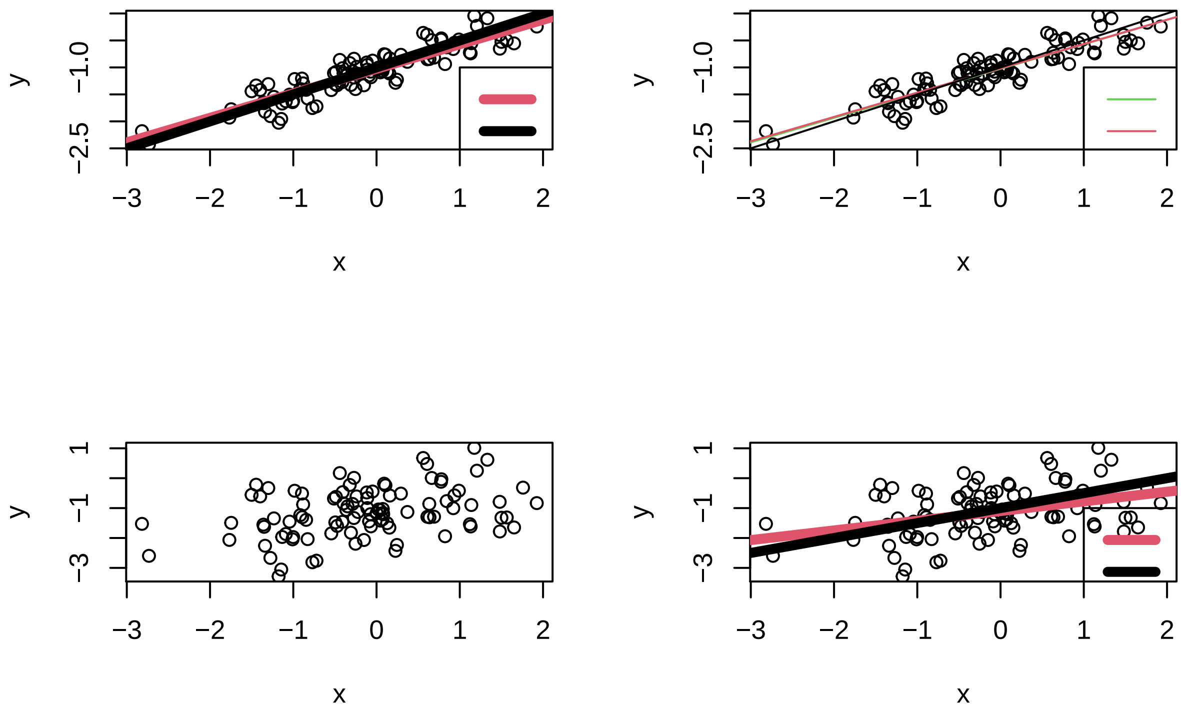
```r
abline(-1, 0.5, lwd=1, col=1)
legend(-1, legend = c("Model 13e","Model 13g", "Original equation"), col=3:1, lwd=1)



#The multiple R-squared is higher, but the adjusted R-squared is lower in the polynomial model. The Res
#The fitted lines are nearly indistinguishable.

confint(model13he)
```

```
##                   2.5 %     97.5 %
## (Intercept) -1.0654980 -0.9862691
## x            0.4129963  0.4957707
```

```r
#i

rm(list=ls())
options(scipen = 999)

set.seed(1735)

#a

x <- rnorm(100)

#b

eps <- rnorm(100,mean = 0,sd = sqrt(0.75))

#c

y <- -1 + 0.5*x + eps

#The vector length is 100. B_0 = -1 and B_1 = 0.5

#d

plot(x,y)

#There is a positive linear relationship between x and y. There are no outliers, but the data is not we

#e

model13ie <- lm(y ~ x)
summary(model13ie)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.80286 -0.49124 -0.01715  0.52216  1.73483
##
## Coefficients:
##             Estimate Std. Error t value             Pr(>|t|)
## (Intercept) -1.10025    0.07731 -14.231 < 0.0000000000000002 ***
## x            0.32333    0.08077   4.003             0.000122 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7656 on 98 degrees of freedom
## Multiple R-squared:  0.1405, Adjusted R-squared:  0.1318
## F-statistic: 16.02 on 1 and 98 DF,  p-value: 0.0001217
```

```r
#The intercept, B_0, at -1.10 is different than the original equation. The B_1 coefficient, 0.32, is mu

#f


plot(x, y)
abline(model13ie, lwd=5, col=2)
abline(-1, 0.5, lwd=5, col=1)
legend(-1, legend = c("Model 13e", "Original equation"), col=2:1, lwd=5)
```




```r
#g
x2 <- x^2

model13ig <- lm(y ~ x + x2)
summary(model13ig)
```

```
##
## Call:
## lm(formula = y ~ x + x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.79934 -0.48405 -0.02963  0.53530  1.77346
##
## Coefficients:
```

```
##             Estimate Std. Error t value             Pr(>|t|)
## (Intercept) -1.06460    0.09397 -11.329 < 0.0000000000000002 ***
## x            0.30832    0.08403   3.669             0.000398 ***
## x2          -0.04109    0.06120  -0.671             0.503543
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7678 on 97 degrees of freedom
## Multiple R-squared:  0.1445, Adjusted R-squared:  0.1269
## F-statistic: 8.192 on 2 and 97 DF,  p-value: 0.000516
```

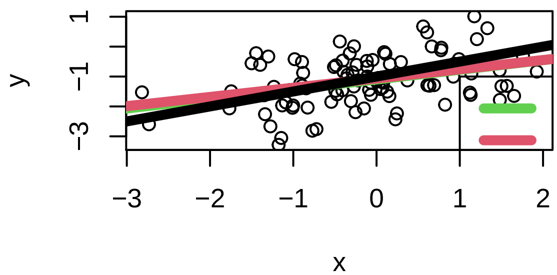
```r
plot(x, y)
abline(model13ie, lwd=5, col=3)
abline(model13ig, lwd=5, col=2)
```

```
## Warning in abline(model13ig, lwd = 5, col = 2): only using the first two of 3
## regression coefficients
```

```r
abline(-1, 0.5, lwd=5, col=1)
legend(-1, legend = c("Model 13e","Model 13g", "Original equation"), col=3:1, lwd=5)


#The multiple R-squared is better, but the adjusted R-squared is worse for the polynomial model. The RS

confint(model13ie)
```

```
##                 2.5 %     97.5 %
## (Intercept) -1.253673 -0.9468206
## x            0.163036  0.4836198
```

```r
#j

#confint(model13e)
#                  2.5 %     97.5 %
#   (Intercept) -1.1464580 -0.9692968
#    x            0.3054538  0.4905429

#confint(model13he)
#                     2.5 %     97.5 %
#    (Intercept) -1.0654980 -0.9862691
#    x            0.4129963  0.4957707

#  confint(model13ie)
#               2.5 %     97.5 %
#(Intercept) -1.253673 -0.9468206
#x            0.163036  0.4836198

#The fit is best on the model with the least noise and widest on the model with the most noise.
```



1
−1
−3
y
−3
−2
−1
0
1
2
x