A2Q4

Undergraduate Student

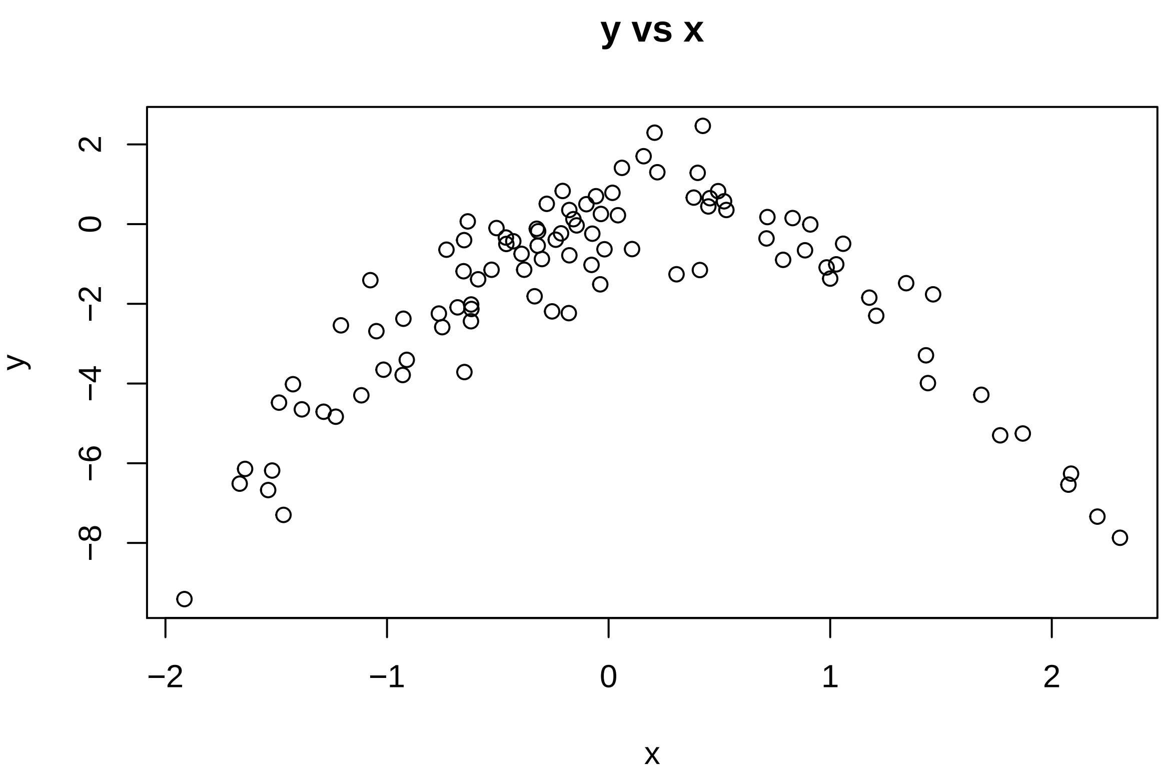
(a)

```
set.seed(1)
e = rnorm(100)
x = rnorm(100)
y = x-2*x^2+e
```

Comment: n = 100. p = 1. The equation is: y = x - $2x^2$+ϵ

(b)

```
plot(x, y, main = "y vs x")
```



Comment: The plot suggests a quadratic relationship between Y and X since we can see a clear parabolic curve.

(c)

(i)

```
library(boot)
set.seed(1)
data <- data.frame(x, y)
m1 <- glm(y~x, data = data)
summary(m1)
```

```
##
## Call:
## glm(formula = y ~ x, data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -7.1052  -1.0395   0.6767   1.7058   4.0770
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.7373     0.2456  -7.075 2.25e-10 ***
## x             0.2956     0.2575   1.148    0.254
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 6.020807)
##
##     Null deviance: 597.98  on 99  degrees of freedom
## Residual deviance: 590.04  on 98  degrees of freedom
## AIC: 467.29
##
## Number of Fisher Scoring iterations: 2
```

```
cv.glm(data = data, m1)$delta[1]
```

```
## [1] 6.351742
```

(ii)

```
m2 <- glm(y~poly(x, 2, raw = T), data = data)
summary(m2)
```

```
##
## Call:
## glm(formula = y ~ poly(x, 2, raw = T), data = data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.32473  -0.60440   0.00421   0.58388   2.29127
```

```
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)             0.11081    0.11730   0.945    0.347
## poly(x, 2, raw = T)1    0.99981    0.09932  10.066   <2e-16 ***
## poly(x, 2, raw = T)2   -2.00212    0.08043 -24.892   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.8233896)
##
##     Null deviance: 597.976  on 99  degrees of freedom
## Residual deviance:  79.869  on 97  degrees of freedom
## AIC: 269.31
##
## Number of Fisher Scoring iterations: 2
```

```
cv.glm(data = data, m2)$delta[1]
```

```
## [1] 0.8392001
```

(iii)

```
m3 <- glm(y~poly(x, 3, raw = T), data = data)
summary(m3)
```

```
##
## Call:
## glm(formula = y ~ poly(x, 3, raw = T), data = data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.25527  -0.62344   0.04973   0.51803   2.23692
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)             0.10256    0.11784   0.870    0.386
## poly(x, 3, raw = T)1    1.14372    0.19403   5.895 5.58e-08 ***
## poly(x, 3, raw = T)2   -1.96707    0.09019 -21.811  < 2e-16 ***
## poly(x, 3, raw = T)3   -0.06382    0.07389  -0.864    0.390
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.8255507)
##
##     Null deviance: 597.976  on 99  degrees of freedom
## Residual deviance:  79.253  on 96  degrees of freedom
## AIC: 270.54
##
## Number of Fisher Scoring iterations: 2
```

```r
cv.glm(data = data, m3)$delta[1]
```

```
## [1] 0.84757
```

(iv)

```r
m4 <- glm(y~poly(x, 4, raw = T), data = data)
summary(m4)
```

```
##
## Call:
## glm(formula = y ~ poly(x, 4, raw = T), data = data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.19661  -0.61381  -0.01333   0.52359   2.15799
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            0.19108    0.13904   1.374    0.173
## poly(x, 4, raw = T)1  1.24405    0.21108   5.894 5.73e-08 ***
## poly(x, 4, raw = T)2 -2.24150    0.24704  -9.074 1.58e-14 ***
## poly(x, 4, raw = T)3 -0.13394    0.09429  -1.420    0.159
## poly(x, 4, raw = T)4  0.08358    0.07006   1.193    0.236
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.8219298)
##
##     Null deviance: 597.976  on 99  degrees of freedom
## Residual deviance:  78.083  on 95  degrees of freedom
## AIC: 271.05
##
## Number of Fisher Scoring iterations: 2
```

```r
cv.glm(data = data, m4)$delta[1]
```

```
## [1] 0.8737288
```

(d)

(i)

```r
set.seed(3)
e = rnorm(100)
x = rnorm(100)
y = x-2*x^2+e
data <- data.frame(x, y)
```

```r
m1 <- glm(y~x, data = data)
summary(m1)
```

```
##
## Call:
## glm(formula = y ~ x, data = data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -10.6288   -0.7015    1.0277    1.8149    4.1024
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.3671     0.3065  -7.723 9.85e-12 ***
## x             0.3547     0.2804   1.265    0.209
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 9.391978)
##
##     Null deviance: 935.44  on 99  degrees of freedom
## Residual deviance: 920.41  on 98  degrees of freedom
## AIC: 511.75
##
## Number of Fisher Scoring iterations: 2
```

```r
cv.glm(data = data, m1)$delta[1]
```

```
## [1] 10.05822
```

(ii)

```r
m2 <- glm(y~poly(x, 2, raw = T), data = data)
summary(m2)
```

```
##
## Call:
## glm(formula = y ~ poly(x, 2, raw = T), data = data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.19612  -0.74151   0.05864   0.74586   1.80209
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -0.04932    0.10980  -0.449    0.654
## poly(x, 2, raw = T)1  0.93001    0.08041  11.566   <2e-16 ***
## poly(x, 2, raw = T)2 -1.94839    0.05746 -33.911   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for gaussian family taken to be 0.7381454)
##
##     Null deviance: 935.44  on 99  degrees of freedom
## Residual deviance:  71.60  on 97  degrees of freedom
## AIC: 258.38
##
## Number of Fisher Scoring iterations: 2
```

```
cv.glm(data = data, m2)$delta[1]
```

```
## [1] 0.757718
```

(iii)

```
m3 <- glm(y~poly(x, 3, raw = T), data = data)
summary(m3)
```

```
##
## Call:
## glm(formula = y ~ poly(x, 3, raw = T), data = data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.22225  -0.70730   0.05345   0.72684   1.78529
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -0.06829    0.11206  -0.609    0.544
## poly(x, 3, raw = T)1  1.03624    0.14585   7.105 2.11e-10 ***
## poly(x, 3, raw = T)2 -1.92285    0.06453 -29.797  < 2e-16 ***
## poly(x, 3, raw = T)3 -0.03614    0.04137  -0.873    0.385
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.7399536)
##
##     Null deviance: 935.444  on 99  degrees of freedom
## Residual deviance:  71.036  on 96  degrees of freedom
## AIC: 259.59
##
## Number of Fisher Scoring iterations: 2
```

```
cv.glm(data = data, m3)$delta[1]
```

```
## [1] 0.7592085
```

(iv)

```r
m4 <- glm(y~poly(x, 4, raw = T), data = data)
summary(m4)
```

```
##
## Call:
## glm(formula = y ~ poly(x, 4, raw = T), data = data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.13706  -0.63797   0.02866   0.63651   1.88106
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -0.174582   0.132442  -1.318    0.191
## poly(x, 4, raw = T)1  0.994890   0.147604   6.740 1.22e-09 ***
## poly(x, 4, raw = T)2 -1.698835   0.164137 -10.350  < 2e-16 ***
## poly(x, 4, raw = T)3 -0.006862   0.045613  -0.150    0.881
## poly(x, 4, raw = T)4 -0.045550   0.030722  -1.483    0.141
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.7308313)
##
##     Null deviance: 935.444  on 99  degrees of freedom
## Residual deviance:  69.429  on 95  degrees of freedom
## AIC: 259.3
##
## Number of Fisher Scoring iterations: 2
```

```r
cv.glm(data = data, m4)$delta[1]
```

```
## [1] 0.7744529
```

Comments: The MSE in (c) and (d) are different because we set different seeds. However, LOOCV errors of models iii and iv increases in both (c) and (d) comparing to i and ii.

(e)

Since from the cubic term, the LOOCV errors begin to increase. We conclude that the second model in (c) has the smallest LOOCV error. This is expected because we can see a quadratic pattern in scatterplot. Naturally, we will prefer to use quadratic equation to fit the data.

(f)

Yes. From model i and model ii we can see that the p-values of linear and quadratic terms are significant smaller than 0.05. This means that they are statistically significant in the model. When we looking at model iii and iv, we found that both cubic term and fourth power term have p-value greater than 0.05. There is strong evidence that the null hypothesis is prefered. Therefore, cubic term and fourth power term can be eliminated in the model. The results exactly match the conclusion in (e).