Problem Set 2

Tom Hanna

10/15/2020

```r
rm(list = ls())

setwd("C:/R Studio Files/POLS6394-Machine-Learning/Problem Set 2")

# 5. a) QDA will perform better on the training set, but will overfit the test set.  LDA will perform b
# b) QDA, QDA
# c) Improve because with higher sample size QDA can take advantage of the separate covariance matrices
# d) False. If the boundary is linear, QDA can only improve on LDA by overfitting.

#6

#a

B_0 <- -6
B_1 <- 0.05
B_2 <- 1
X_1 <- 40
X_2 <- 3.5

prob <- exp(B_0 + (B_1*X_1) + (B_2*X_2))/(1 + exp(B_0 + (B_1*X_1) + (B_2*X_2)))
prob
```

```
## [1] 0.3775407
```

```r
#a) [1] 0.377540

# b)    .5 = exp(-6 + 0.05*X_1 + 3.5*1)/(1 + exp(-6 + .05*X_1 + 3.5*1)
#                                    .5*(1 + exp( .05*X_1 - 2.5)=  exp(.05*X_1 - 2.5)
#                                        .5 + .5*exp exp(.05*X_1 - 2.5))=  exp(.05*X_1 - 2.5)=
#                                        .5 = .5( exp(.05*X_1 - 2.5))
#                                    1 = exp(.05*X_1 - 2.5)=
#                                        log(1) = .05X_1 - 2.5
#                                    2.5 = .05 X_1
#                                    X_1 = 50
#                                    50 hours
#

#7

# pi-yes = .8
# pi-no = .2
```

```r
# mu-yes = 10
# mu-no = 0
# variance = 36
# P(Yes|4)  is  0.04033
# P(No|4) is 0.05324
# P(A) is .8 P(B) is .2
# P(A|4) = .8*.04033/(.8*.04033 + .2*0.05324)
# Probability of a dividend is [1] 0.7518643 or 75.2%


#10

#A - Volume increases with year

library(ISLR)

weekly <- as.data.frame(Weekly)

head(weekly)
```

```
##   Year   Lag1   Lag2   Lag3   Lag4   Lag5    Volume  Today Direction
## 1 1990  0.816  1.572 -3.936 -0.229 -3.484 0.1549760 -0.270      Down
## 2 1990 -0.270  0.816  1.572 -3.936 -0.229 0.1485740 -2.576      Down
## 3 1990 -2.576 -0.270  0.816  1.572 -3.936 0.1598375  3.514        Up
## 4 1990  3.514 -2.576 -0.270  0.816  1.572 0.1616300  0.712        Up
## 5 1990  0.712  3.514 -2.576 -0.270  0.816 0.1537280  1.178        Up
## 6 1990  1.178  0.712  3.514 -2.576 -0.270 0.1544440 -1.372      Down
```

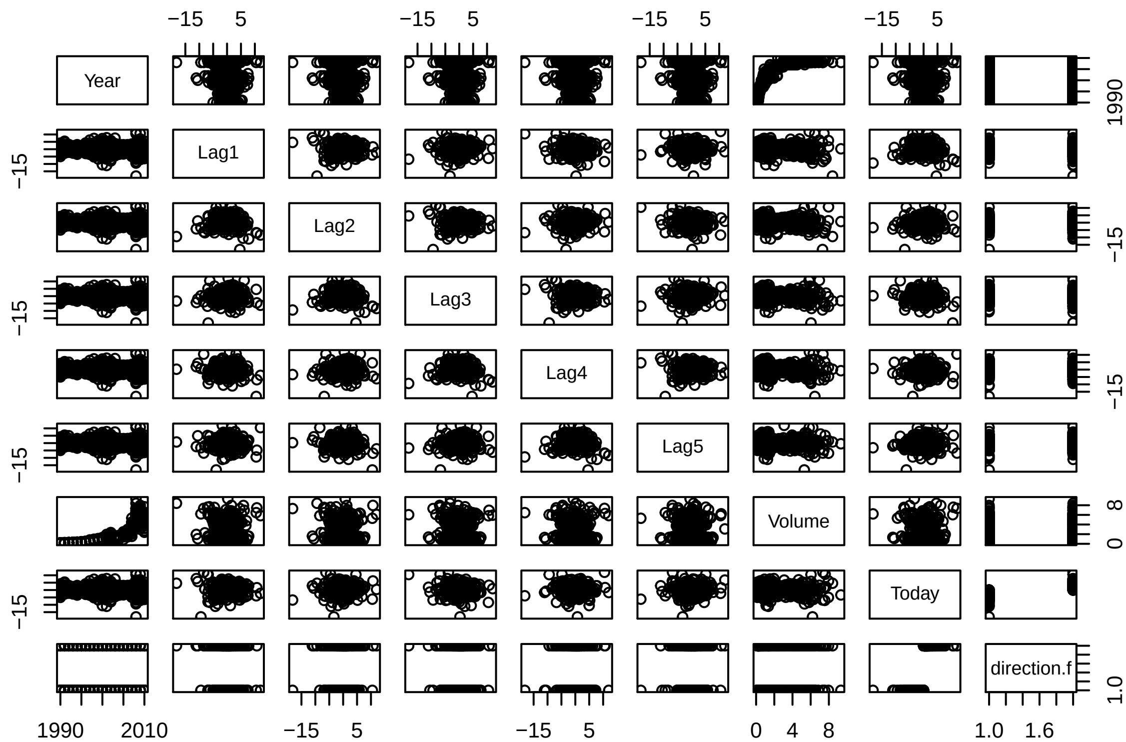
```r
#create factor with 0,1 values for Direction
#Down = 0, Up = 1

weekly$direction.f <- factor(weekly$Direction,levels = c("Down","Up"), labels = c(0,1))

#drop non-numeric Direction variable

weekly <- weekly[ -c(9)]

str(weekly)
```

```
## 'data.frame':    1089 obs. of  9 variables:
##  $ Year       : num  1990 1990 1990 1990 1990 1990 1990 1990 1990 1990 ...
##  $ Lag1       : num  0.816 -0.27 -2.576 3.514 0.712 ...
##  $ Lag2       : num  1.572 0.816 -0.27 -2.576 3.514 ...
##  $ Lag3       : num  -3.936 1.572 0.816 -0.27 -2.576 ...
##  $ Lag4       : num  -0.229 -3.936 1.572 0.816 -0.27 ...
##  $ Lag5       : num  -3.484 -0.229 -3.936 1.572 0.816 ...
##  $ Volume     : num  0.155 0.149 0.16 0.162 0.154 ...
##  $ Today      : num  -0.27 -2.576 3.514 0.712 1.178 ...
##  $ direction.f: Factor w/ 2 levels "0","1": 1 1 2 2 2 1 2 2 2 1 ...
```

```r
weekly$direction.f <- as.numeric(weekly$direction.f)

str(weekly)
```

```
## 'data.frame':    1089 obs. of  9 variables:
##  $ Year       : num  1990 1990 1990 1990 1990 1990 1990 1990 1990 1990 ...
##  $ Lag1       : num  0.816 -0.27 -2.576 3.514 0.712 ...
##  $ Lag2       : num  1.572 0.816 -0.27 -2.576 3.514 ...
##  $ Lag3       : num  -3.936 1.572 0.816 -0.27 -2.576 ...
##  $ Lag4       : num  -0.229 -3.936 1.572 0.816 -0.27 ...
##  $ Lag5       : num  -3.484 -0.229 -3.936 1.572 0.816 ...
##  $ Volume     : num  0.155 0.149 0.16 0.162 0.154 ...
##  $ Today      : num  -0.27 -2.576 3.514 0.712 1.178 ...
##  $ direction.f: num  1 1 2 2 2 1 2 2 2 1 ...
```

```r
pairs(weekly)
```




```r
summary(weekly)
```

```
##       Year           Lag1               Lag2               Lag3
##  Min.   :1990   Min.   :-18.1950   Min.   :-18.1950   Min.   :-18.1950
##  1st Qu.:1995   1st Qu.: -1.1540   1st Qu.: -1.1540   1st Qu.: -1.1580
##  Median :2000   Median :  0.2410   Median :  0.2410   Median :  0.2410
##  Mean   :2000   Mean   :  0.1506   Mean   :  0.1511   Mean   :  0.1472
##  3rd Qu.:2005   3rd Qu.:  1.4050   3rd Qu.:  1.4090   3rd Qu.:  1.4090
##  Max.   :2010   Max.   : 12.0260   Max.   : 12.0260   Max.   : 12.0260
##       Lag4               Lag5              Volume             Today
##  Min.   :-18.1950   Min.   :-18.1950   Min.   :0.08747   Min.   :-18.1950
##  1st Qu.: -1.1580   1st Qu.: -1.1660   1st Qu.:0.33202   1st Qu.: -1.1540
```

```
##  Median :  0.2380   Median :  0.2340   Median :1.00268   Median :  0.2410
##  Mean   :  0.1458   Mean   :  0.1399   Mean   :1.57462   Mean   :  0.1499
##  3rd Qu.:  1.4090   3rd Qu.:  1.4050   3rd Qu.:2.05373   3rd Qu.:  1.4050
##  Max.   : 12.0260   Max.   : 12.0260   Max.   :9.32821   Max.   : 12.0260
##   direction.f
##  Min.   :1.000
##  1st Qu.:1.000
##  Median :2.000
##  Mean   :1.556
##  3rd Qu.:2.000
##  Max.   :2.000
```

```r
cov(weekly)
```

```
##                    Year        Lag1        Lag2        Lag3        Lag4
## Year        36.39928361 -0.45916269 -0.47486414 -0.42733259 -0.44326136
## Lag1        -0.45916269  5.55550803 -0.41588937  0.32623326 -0.39651131
## Lag2        -0.47486414 -0.41588937  5.55664749 -0.42133412  0.32482186
## Lag3        -0.42733259  0.32623326 -0.42133412  5.57196961 -0.42006376
## Lag4        -0.44326136 -0.39651131  0.32482186 -0.42006376  5.57091625
## Lag5        -0.43477694 -0.04554365 -0.40354287  0.33809200 -0.42175889
## Volume       8.56741626 -0.25820895 -0.33998578 -0.27585576 -0.24313391
## Today       -0.46157229 -0.41682494  0.32872301 -0.39636625 -0.04353538
## direction.f -0.06658497 -0.05859181  0.08519046 -0.02688777 -0.02411213
##                    Lag5      Volume       Today direction.f
## Year        -0.43477694  8.56741626 -0.46157229 -0.06658497
## Lag1        -0.04554365 -0.25820895 -0.41682494 -0.05859181
## Lag2        -0.40354287 -0.33998578  0.32872301  0.08519046
## Lag3         0.33809200 -0.27585576 -0.39636625 -0.02688777
## Lag4        -0.42175889 -0.24313391 -0.04353538 -0.02411213
## Lag5         5.57566532 -0.23305313  0.06128981 -0.02132721
## Volume      -0.23305313  2.84474239 -0.13149343 -0.01508865
## Today        0.06128981 -0.13149343  5.55510671  0.84365635
## direction.f -0.02132721 -0.01508865  0.84365635  0.24714052
```

```r
cor(weekly)
```

```
##                    Year         Lag1        Lag2        Lag3         Lag4
## Year         1.00000000 -0.032289274 -0.03339001 -0.03000649 -0.031127923
## Lag1        -0.03228927  1.000000000 -0.07485305  0.05863568 -0.071273876
## Lag2        -0.03339001 -0.074853051  1.00000000 -0.07572091  0.058381535
## Lag3        -0.03000649  0.058635682 -0.07572091  1.00000000 -0.075395865
## Lag4        -0.03112792 -0.071273876  0.05838153 -0.07539587  1.000000000
## Lag5        -0.03051910 -0.008183096 -0.07249948  0.06065717 -0.075675027
## Volume       0.84194162 -0.064951313 -0.08551314 -0.06928771 -0.061074617
## Today       -0.03245989 -0.075031842  0.05916672 -0.07124364 -0.007825873
## direction.f -0.02220025 -0.050003804  0.07269634 -0.02291281 -0.020549456
##                     Lag5      Volume        Today direction.f
## Year        -0.030519101  0.84194162 -0.032459894 -0.02220025
## Lag1        -0.008183096 -0.06495131 -0.075031842 -0.05000380
## Lag2        -0.072499482 -0.08551314  0.059166717  0.07269634
## Lag3         0.060657175 -0.06928771 -0.071243639 -0.02291281
## Lag4        -0.075675027 -0.06107462 -0.007825873 -0.02054946
```

```
## Lag5          1.000000000 -0.05851741  0.011012698 -0.01816827
## Volume       -0.058517414  1.00000000 -0.033077783 -0.01799521
## Today         0.011012698 -0.03307778  1.000000000  0.72002470
## direction.f -0.018168272 -0.01799521  0.720024704  1.00000000
```

```
#a - volume increases with year

Weekly <- as.data.frame(Weekly)

model1 <- glm(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume, data = Weekly,
              family = binomial(link = "logit"))
summary(model1)
```

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
##     Volume, family = binomial(link = "logit"), data = Weekly)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6949  -1.2565   0.9913   1.0849   1.4579
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.26686    0.08593   3.106   0.0019 **
## Lag1        -0.04127    0.02641  -1.563   0.1181
## Lag2         0.05844    0.02686   2.175   0.0296 *
## Lag3        -0.01606    0.02666  -0.602   0.5469
## Lag4        -0.02779    0.02646  -1.050   0.2937
## Lag5        -0.01447    0.02638  -0.549   0.5833
## Volume      -0.02274    0.03690  -0.616   0.5377
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1496.2  on 1088  degrees of freedom
## Residual deviance: 1486.4  on 1082  degrees of freedom
## AIC: 1500.4
##
## Number of Fisher Scoring iterations: 4
```

```
#(b) Lag2 is significant at the .05 level

p1 = predict(model1, type = "response")
p2 = rep("Down", length(p1))
p2[p1 > 0.5] = "Up"
table(p2, Weekly$Direction)
```

```
##
## p2     Down  Up
##   Down   54  48
##   Up    430 557
```

```r
#c - The percentage of correct predictions (Up/Up and Down/Down - the top left and bottom right cells)

(54+557)/(54 + 48 + 430 + 557)*100
```

```
## [1] 56.10652
```

```r
#Out of the 605 weeks the market was up, it predicted correctly 557 times for correct prediction

(557/605)*100
```

```
## [1] 92.06612
```

```r
#of the 484 down weeks, it predicted correctly 54 times for a percentage correct of only

(54/484)*100
```

```
## [1] 11.15702
```

```r
#the model overpredicts Up significantly overall.

train1 <- subset(Weekly, Year < 2009)
test1 <- subset(Weekly, Year > 2008)
model2 <- glm(Direction ~ Lag2, data = train1, family = binomial(link = "logit"))

p3 = predict(model2, test1, type = "response")
p4 = rep("Down", length(p3))
p4[p3 > 0.5] = "Up"
table(p4, test1$Direction)
```

```
##
## p4     Down Up
##   Down    9  5
##   Up     34 56
```

```r
#Correct predictions in test data is
(9+56)/(9 + 5 + 34 + 56)
```

```
## [1] 0.625
```

```r
#e

library(MASS)
lda.model1 <- lda(Direction ~ Lag2, data = train1)
lda.predict <- predict(lda.model1,test1)
table(lda.predict$class,test1$Direction)
```

```
##
##        Down Up
##   Down    9  5
##   Up     34 56
```

```r
#Correct prediction for LDA is
(9+56)/(9+5+34+56)
```

```
## [1] 0.625
```

```r
#f
qda.model1 <- qda(Direction ~ Lag2, data = train1)
qda.cl <- predict(qda.model1,test1)$class
table(qda.cl,test1$Direction)
```

```
##
## qda.cl Down Up
##   Down    0  0
##   Up     43 61
```

```r
#Correct predictions for QDA is
(61)/(43 + 61)
```

```
## [1] 0.5865385
```

```r
#g
library(class)
train.X <- as.matrix(train1$Lag2)
test.X <- as.matrix(test1$Lag2)
set.seed(1)
knn.p <- knn(train.X, test.X, train1$Direction, k = 1)
table(knn.p,test1$Direction)
```

```
##
## knn.p  Down Up
##   Down   21 30
##   Up     22 31
```

```r
#Correct predictions for KNN with k = 1 is
(21+31)/(21+30+31+22)
```

```
## [1] 0.5
```

```r
#h LDA and logit produce the best error rate with 0.625 correct predictions.

#i

qda.model3 <- qda(Direction ~ Lag2^2, data = train1)
qda.c2 <- predict(qda.model3,test1)$class
table(qda.c2,test1$Direction)
```

```
##
## qda.c2 Down Up
##   Down    0  0
##   Up     43 61
```

```r
qda.model4 <- qda(Direction ~ Lag2*Year, data = train1)
qda.c3 <- predict(qda.model4,test1)$class
table(qda.c3,test1$Direction)
```

```
##
## qda.c3 Down Up
##   Down   24 38
##   Up     19 23
```

```r
#correct prediction for QDA model 4 is
(24+23)/(24+38+19+23)
```

```
## [1] 0.4519231
```

```r
#The model did better always predicting "Up"

train.X2 <- as.matrix(train1$Lag2)
test.X2 <- as.matrix(test1$Lag2)
set.seed(1)
knn.p2 <- knn(train.X2, test.X2, train1$Direction, k = 20)
table(knn.p2,test1$Direction)
```

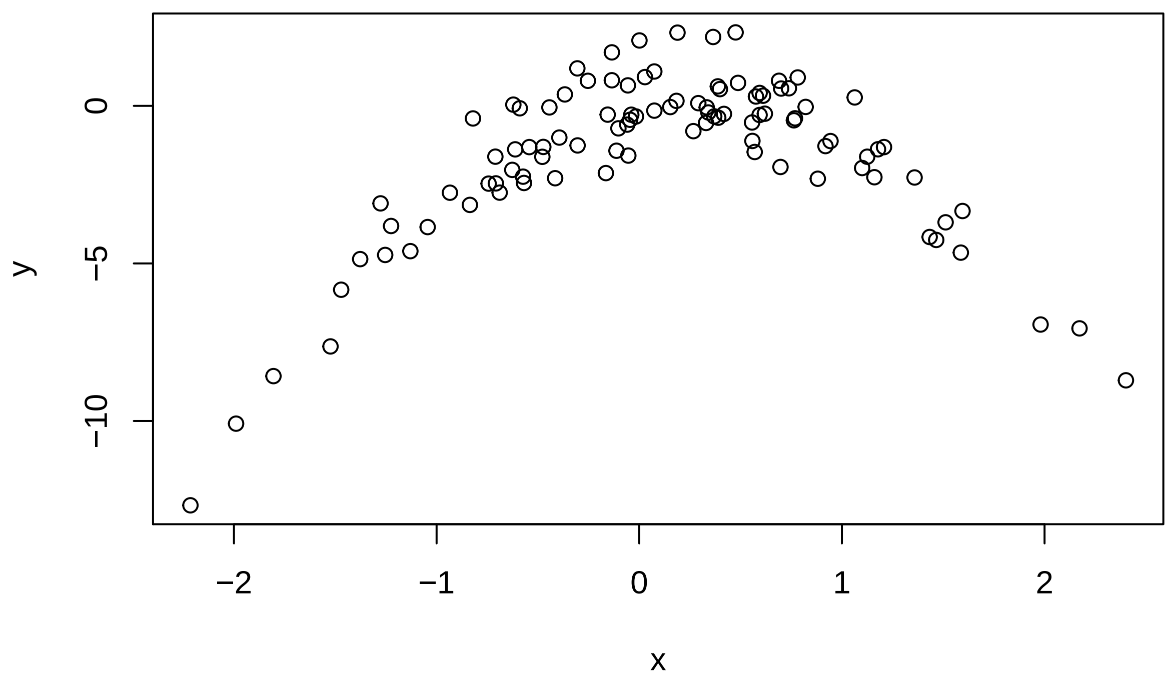
```
##
## knn.p2 Down Up
##   Down   21 21
##   Up     22 40
```

```r
#Correct prediction for KNN with k = 20 is
(21+40)/(21+21+22+40)
```

```
## [1] 0.5865385
```

```r
#which is an improvement over correct prediction of 0.5 for KNN with k = 1.

##Chapter 5

rm(list = ls())

setwd("C:/R Studio Files/POLS6394-Machine-Learning/Problem Set 2")

#3
#a  The data is split into k sets. Starting with the first set, the set is used as a validation set
# with the other k-1 sets used as a training set. The Mean Squared Error is evaluated, then the process
#is repeated for each of the k sets.

#b
#i Like LOOCV, k-fold has the advantage of always returning the same error rate, unlike the validation
#set approach where randomness can cause variation in the error rate.
#ii Since LOOCV is a special case of k-fold validation where k = n, LOOCV requires more computation.

set.seed(1)
```

```
x <- rnorm(100)
y = x - 2 * x^2 + rnorm(100)

#n = 100, p = 2
#Y = X - 2(x^2) + e

plot(x,y)
```




```
#The relationship is curvilinear and nonmonotonic, with the value of Y first rising, then
#falling as X increases.

library(boot)
MyData <- data.frame(x, y)
set.seed(1)
# i.
fit1 <- glm(y ~ x)
cv.glm(MyData, fit1)$delta
```

```
## [1] 7.288162 7.284744
```

```
#ii

fit2 <- glm(y ~ poly(x, 2))
cv.glm(MyData, fit2)$delta
```

```
## [1] 0.9374236 0.9371789
```

```
#iii
fit3 <- glm(y ~ (poly(x,3)))
cv.glm(MyData,fit3)$delta
```

```
## [1] 0.9566218 0.9562538
```

```
#iv
fit4 <- glm(y ~ (poly(x,4)))
cv.glm(MyData,fit4)$delta
```

```
## [1] 0.9539049 0.9534453
```

```
#d

set.seed(95)
MyData <- data.frame(x, y)
set.seed(1)
# i.
fit1 <- glm(y ~ x)
cv.glm(MyData, fit1)$delta
```

```
## [1] 7.288162 7.284744
```

```
#ii

fit2 <- glm(y ~ poly(x, 2))
cv.glm(MyData, fit2)$delta
```

```
## [1] 0.9374236 0.9371789
```

```
#iii
fit3 <- glm(y ~ (poly(x,3)))
cv.glm(MyData,fit3)$delta
```

```
## [1] 0.9566218 0.9562538
```

```
#iv
fit4 <- glm(y ~ (poly(x,4)))
cv.glm(MyData,fit4)$delta
```

```
## [1] 0.9539049 0.9534453
```

```
#The results are the same because it evaluates k = n test sets that do not change randomly.

#e - Model fit2, the 2nd degree polynomial, had the smallest error. The scatterplot matched a 2nd order
#polynomial, or quadratic, transformation, so this is not surprising.

#f

summary(fit1)
```

```
##
## Call:
## glm(formula = y ~ x)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -9.5161  -0.6800   0.6812   1.5491   3.8183
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.6254     0.2619  -6.205 1.31e-08 ***
## x             0.6925     0.2909   2.380   0.0192 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 6.760719)
##
##     Null deviance: 700.85  on 99  degrees of freedom
## Residual deviance: 662.55  on 98  degrees of freedom
## AIC: 478.88
##
## Number of Fisher Scoring iterations: 2
```

```r
summary(fit2)
```

```
##
## Call:
## glm(formula = y ~ poly(x, 2))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9650  -0.6254  -0.1288   0.5803   2.2700
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.5500     0.0958  -16.18  < 2e-16 ***
## poly(x, 2)1   6.1888     0.9580    6.46 4.18e-09 ***
## poly(x, 2)2 -23.9483     0.9580  -25.00  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.9178258)
##
##     Null deviance: 700.852  on 99  degrees of freedom
## Residual deviance:  89.029  on 97  degrees of freedom
## AIC: 280.17
##
## Number of Fisher Scoring iterations: 2
```

```r
summary(fit3)
```

```
##
## Call:
```

```
## glm(formula = y ~ (poly(x, 3)))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9765  -0.6302  -0.1227   0.5545   2.2843
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.55002    0.09626 -16.102  < 2e-16 ***
## poly(x, 3)1   6.18883    0.96263   6.429 4.97e-09 ***
## poly(x, 3)2 -23.94830    0.96263 -24.878  < 2e-16 ***
## poly(x, 3)3   0.26411    0.96263   0.274    0.784
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.9266599)
##
##     Null deviance: 700.852  on 99  degrees of freedom
## Residual deviance:  88.959  on 96  degrees of freedom
## AIC: 282.09
##
## Number of Fisher Scoring iterations: 2
```

```r
summary(fit4)
```

```
##
## Call:
## glm(formula = y ~ (poly(x, 4)))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0550  -0.6212  -0.1567   0.5952   2.2267
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.55002    0.09591 -16.162  < 2e-16 ***
## poly(x, 4)1   6.18883    0.95905   6.453 4.59e-09 ***
## poly(x, 4)2 -23.94830    0.95905 -24.971  < 2e-16 ***
## poly(x, 4)3   0.26411    0.95905   0.275    0.784
## poly(x, 4)4   1.25710    0.95905   1.311    0.193
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.9197797)
##
##     Null deviance: 700.852  on 99  degrees of freedom
## Residual deviance:  87.379  on 95  degrees of freedom
## AIC: 282.3
##
## Number of Fisher Scoring iterations: 2
```

```r
#Adding the squared term improved model fit and caused the x term significance to move above .01.
#Adding third and fourth degrees to the polynomial increased the AIC, indicating worse fit,
```

```
#and the 3rd and 4th degrees were not significant, though the first two degrees were still
#significant.
```