homework 6

Problem 1 Exercise 12.4 from the TEXT.

a) For each of the leagues, fit the model.

```r
goal <- read.table("Table12.15.txt", header = T)
attach(goal)
goal$Ymat <- cbind(Success, Attempts - Success)
NFL <- goal[1:5,]
AFL <- goal[6:10,]
# model for NFL
glm.NFL <- glm(NFL$Ymat ~ NFL$Distance + I(NFL$Distance^2), family = binomial)
summary(glm.NFL)
```

```
##
## Call:
## glm(formula = NFL$Ymat ~ NFL$Distance + I(NFL$Distance^2), family = binomial)
##
## Deviance Residuals:
##        1         2         3         4         5
##  0.11628  -0.00048  -0.40173   0.64209  -0.91465
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)        2.490203   1.018620   2.445   0.0145 *
## NFL$Distance      -0.013167   0.065990  -0.200   0.8419
## I(NFL$Distance^2) -0.001513   0.001008  -1.500   0.1335
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 147.7816  on 4  degrees of freedom
## Residual deviance:   1.4238  on 2  degrees of freedom
## AIC: 28.89
##
## Number of Fisher Scoring iterations: 4
```

```r
# model for AFL
glm.AFL <- glm(AFL$Ymat ~ AFL$Distance + I(AFL$Distance^2), family = binomial)
summary(glm.AFL)
```

```
##
## Call:
## glm(formula = AFL$Ymat ~ AFL$Distance + I(AFL$Distance^2), family = binomial)
##
## Deviance Residuals:
##       1        2        3        4        5
##  0.3187  -0.6829   0.7721  -0.5231   0.2853
```

```
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)        4.892466   1.189274   4.114 3.89e-05 ***
## AFL$Distance      -0.197046   0.074348  -2.650  0.00804 **
## I(AFL$Distance^2)  0.001604   0.001098   1.461  0.14395
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 78.7794  on 4  degrees of freedom
## Residual deviance:  1.5192  on 2  degrees of freedom
## AIC: 28.443
##
## Number of Fisher Scoring iterations: 3
```

b) Fit a single model combining the data from both leagues by extending the model to include the indicator Z

```
glm.fit2 <- glm(goal$Ymat ~ goal$Distance + I(goal$Distance^2) + as.factor(goal$Z), family = binomial)
summary(glm.fit2)
```

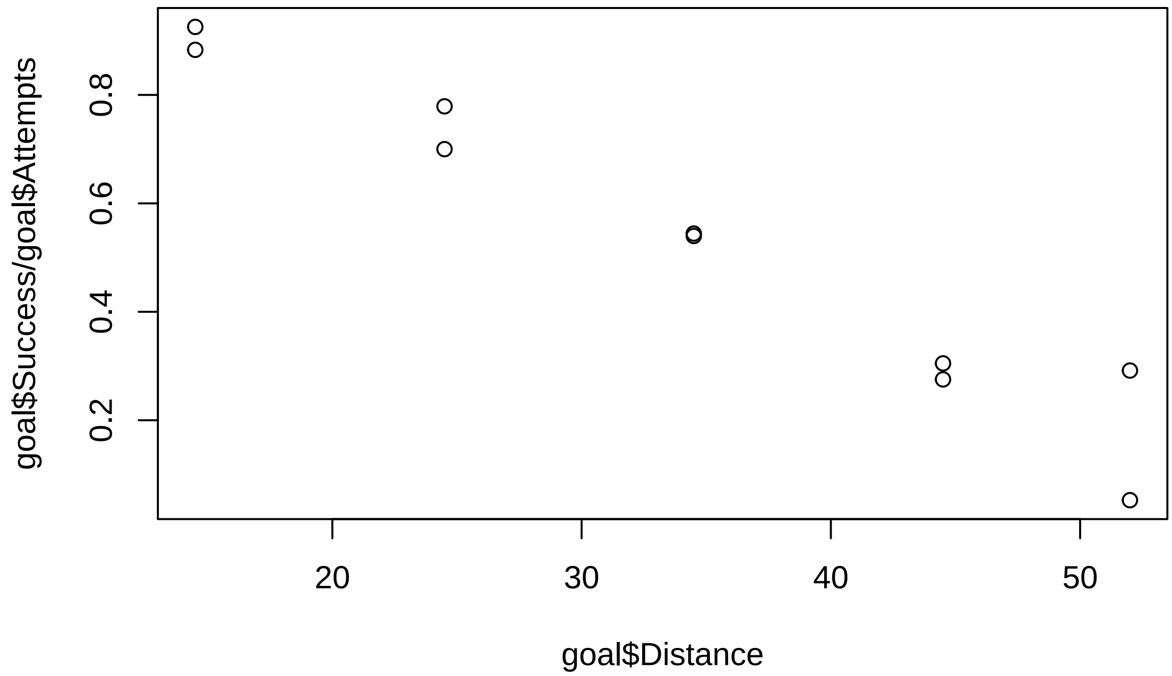
```
##
## Call:
## glm(formula = goal$Ymat ~ goal$Distance + I(goal$Distance^2) +
##     as.factor(goal$Z), family = binomial)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.86350  -0.20086   0.03301   0.55505   1.60112
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         3.5241844  0.7747832   4.549  5.4e-06 ***
## goal$Distance      -0.0958710  0.0490210  -1.956   0.0505 .
## I(goal$Distance^2) -0.0001086  0.0007365  -0.147   0.8828
## as.factor(goal$Z)1  0.1037533  0.1698311   0.611   0.5413
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 228.5180  on 9  degrees of freedom
## Residual deviance:   8.9776  on 6  degrees of freedom
## AIC: 59.367
##
## Number of Fisher Scoring iterations: 4
```

c) Does the quadratic term contribute significantly to the model

No, from the summary of model in part b), the quadratic term is not significant at 5% significant level.

d) Are the probabilities of scoring field goals from a given distance the same for each league?

```
plot(goal$Success/goal$Attempts ~ goal$Distance)
```



For Distance at 50, there is difference between NFL and AFL, for other distances, there's no difference.

Problem 2 Exercise 12.6 from the TEXT.

a) Show that there is no substantial improvement in fit, and the correct classification rate from a model using only IR and SSPG.

```
# data processing
diabete <- read.table("Table12.6-Table12.7(1).txt", header = T)
diabete$CC <- as.factor(diabete$CC)
levels(diabete$CC)
```

```
## [1] "1" "2" "3"
```

```
diabete$CC <- relevel(diabete$CC, ref = '3')
```

```
# model using IR and SSPG
multinom_model <- multinom(CC ~ IR + SSPG, data = diabete)
```

```
## # weights:  12 (6 variable)
## initial  value 159.298782
## iter  10 value 72.172679
## iter  20 value 72.028901
## final  value 72.028883
```

```
## converged
```

```r
summary(multinom_model)
```

```
## Call:
## multinom(formula = CC ~ IR + SSPG, data = diabete)
##
## Coefficients:
##   (Intercept)           IR       SSPG
## 1   -7.110590 -0.013427199 0.04259435
## 2   -4.548408  0.003257602 0.01951007
##
## Std. Errors:
##   (Intercept)          IR        SSPG
## 1   1.6882103 0.004651300 0.007973417
## 2   0.7714595 0.002292307 0.004451874
##
## Residual Deviance: 144.0578
## AIC: 156.0578
```

```r
diabete$CC_predict <- predict(multinom_model, diabete, "class")
tab <- table(diabete$CC, diabete$CC_predict)
# classification rate
round((sum(diag(tab))/sum(tab))*100,2)
```

```
## [1] 81.38
```

```r
# model using IR, SSPG and RW
multinom_model2 <- multinom(CC ~ IR + SSPG + RW, data = diabete)
```

```
## # weights:  15 (8 variable)
## initial  value 159.298782
## iter  10 value 69.027793
## iter  20 value 68.418245
## iter  30 value 68.414665
## final  value 68.414644
## converged
```

```r
summary(multinom_model2)
```

```
## Call:
## multinom(formula = CC ~ IR + SSPG + RW, data = diabete)
##
## Coefficients:
##   (Intercept)           IR       SSPG        RW
## 1   -1.845230 -0.013353688 0.04550552 -5.867196
## 2   -7.615261  0.003586749 0.01641449  3.472572
##
## Std. Errors:
##   (Intercept)          IR        SSPG       RW
## 1    3.463507 0.005019289 0.009241721 3.866580
## 2    2.335615 0.002349168 0.004981886 2.446151
##
## Residual Deviance: 136.8293
## AIC: 152.8293
```

```r
diabete$CC_predict2 <- predict(multinom_model2, diabete, "class")
tab2 <- table(diabete$CC, diabete$CC_predict2)
# classification rate
round((sum(diag(tab2))/sum(tab2))*100,2)
```

```
## [1] 82.76
```

As we can see from above results, classification rate for model using IR and SSPG is 81.38%, and model for including RW is 82.76%, so no substantial improment in prediction rate.

b) Fit an ordinal logistic model using RW, IR, and SSPG to explain CC. Show that there is no substantial improvement in fit, and the correct classification rate from a model using only IR and SSPG.

```r
ordinal_model <- polr(CC ~ IR + SSPG, data = diabete, Hess = T)
summary(ordinal_model)
```

```
## Call:
## polr(formula = CC ~ IR + SSPG, data = diabete, Hess = T)
##
## Coefficients:
##         Value Std. Error t value
## IR   0.006911   0.001669   4.141
## SSPG 0.010986   0.001760   6.241
##
## Intercepts:
##     Value  Std. Error t value
## 3|1 3.4820 0.5562     6.2598
## 1|2 4.9309 0.6384     7.7233
##
## Residual Deviance: 236.0811
## AIC: 244.0811
```

```r
diabete$CC_ordinal_predict = predict(ordinal_model, diabete)
tab3 <- table(diabete$CC, diabete$CC_ordinal_predict)
round((sum(diag(tab3))/sum(tab3))*100,2)
```

```
## [1] 62.76
```

```r
ordinal_model2 <- polr(CC ~ IR + SSPG + RW, data = diabete, Hess = T)
summary(ordinal_model2)
```

```
## Call:
## polr(formula = CC ~ IR + SSPG + RW, data = diabete, Hess = T)
##
## Coefficients:
##         Value Std. Error t value
## IR   0.006141   0.001678   3.659
## SSPG 0.010084   0.001833   5.501
## RW   2.754902   1.543805   1.784
##
## Intercepts:
##     Value  Std. Error t value
## 3|1 5.8654 1.4816     3.9588
## 1|2 7.3459 1.5354     4.7843
##
```

```
## Residual Deviance: 232.856
## AIC: 242.856
```

```
diabete$CC_ordinal_predict2 = predict(ordinal_model2, diabete)
tab4 <- table(diabete$CC, diabete$CC_ordinal_predict2)
round((sum(diag(tab4))/sum(tab4))*100,2)
```

```
## [1] 64.83
```

From the results above, we can see for ordinal logistic model using only IR and SSPG, model AIC is 244.0811, correct classification rate is 62.76%, by adding RW term, AIC is 242.856, and correct classification rate is 64.83%, in which AIC only decreases by a little and correct classification rate does not increase by a lot, thus, there's no substantial improvement in terms of model fit and classification rate.

Problem 3 Exercise 13.1 from the TEXT.

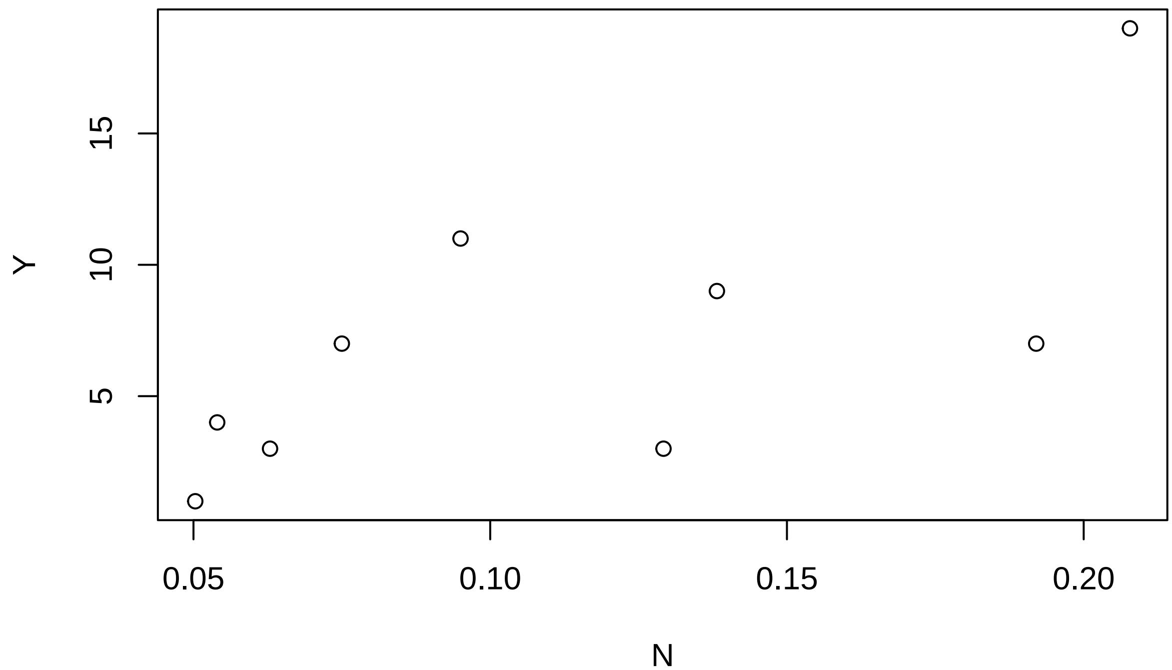
```
injury <- read.table("Table6.6(1).txt", header = T)
# least squares
lm.fit.ls <- lm(Y ~ N, data = injury)
summary(lm.fit.ls)
```

```
##
## Call:
## lm(formula = Y ~ N, data = injury)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.3351 -2.1281  0.1605  2.2670  5.6382
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.1402     3.1412  -0.045   0.9657
## N            64.9755    25.1959   2.579   0.0365 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.201 on 7 degrees of freedom
## Multiple R-squared:  0.4872, Adjusted R-squared:  0.4139
## F-statistic:  6.65 on 1 and 7 DF,  p-value: 0.03654
```

```
AIC(lm.fit.ls)
```

```
## [1] 55.11424
```

```
plot(Y ~ N, data = injury)
```




```r
# transformed least squares
lm.fit.tls <- lm(Y ~ I(N^2), data = injury)
summary(lm.fit.tls)
```

```
##
## Call:
## lm(formula = Y ~ I(N^2), data = injury)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.5729 -2.7775  0.1236  2.4296  5.5587
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.129      2.043   1.532   0.1694
## I(N^2)       256.170     96.776   2.647   0.0331 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.147 on 7 degrees of freedom
## Multiple R-squared:  0.5002, Adjusted R-squared:  0.4288
## F-statistic: 7.007 on 1 and 7 DF,  p-value: 0.03308
```

```r
AIC(lm.fit.tls)
```

```
## [1] 54.88222
```

```r
# Poisson
glm.fit.poisson <- glm(Y ~ N, data = injury, family = poisson)
summary(glm.fit.poisson)
```

```
##
## Call:
## glm(formula = Y ~ N, family = poisson, data = injury)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.81894  -1.69082   0.06495   1.02407   2.06811
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.8945     0.3265   2.739  0.00615 **
## N             8.5018     2.1575   3.941 8.13e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 31.859  on 8  degrees of freedom
## Residual deviance: 16.291  on 7  degrees of freedom
## AIC: 52.251
##
## Number of Fisher Scoring iterations: 5
```

We can see from results above, AIC for least squares model is 55.11424, AIC for transformed least squares model is 54.88222, and AIC for poisson regression model is 52.251, poisson regression model has the lowest AIC, which means it provides the best description of the data.

Problem 4 Exercise 13.4 from the TEXT.

```r
ad <- read.csv("table6.17.csv")
lm.ad <- lm(log(R) ~ log(P), data = ad)
summary(lm.ad)
```

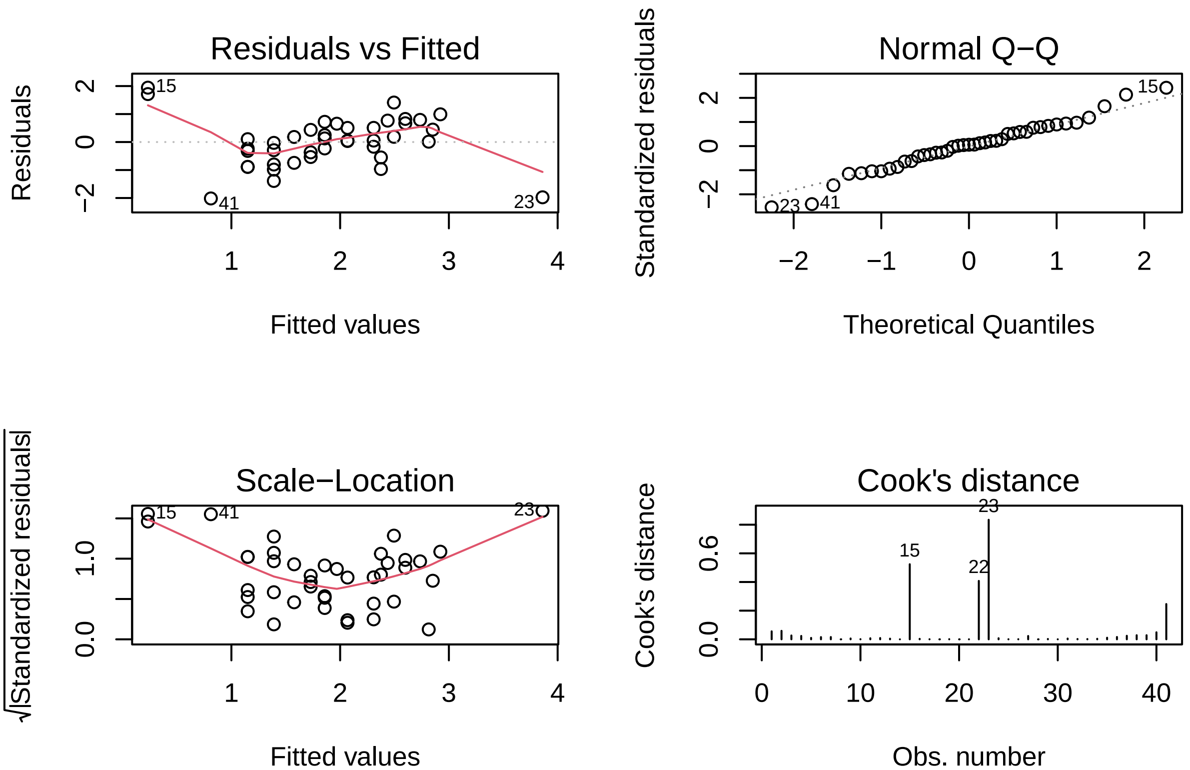
```
##
## Call:
## lm(formula = log(R) ~ log(P), data = ad)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.01534 -0.53524  0.04836  0.50718  1.94245
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.2323     0.3398   0.684    0.498
## log(P)        0.8354     0.1571   5.318 4.57e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8719 on 39 degrees of freedom
## Multiple R-squared:  0.4203, Adjusted R-squared:  0.4055
```

```
## F-statistic: 28.28 on 1 and 39 DF,  p-value: 4.575e-06
```

```r
influence.measures(lm.ad)
```

```
## Influence measures of
##   lm(formula = log(R) ~ log(P), data = ad) :
##
##      dfb.1_  dfb.l.P.    dffit cov.r   cook.d    hat inf
## 1  -0.17468  0.274904  0.33568 1.058 5.58e-02 0.0741
## 2  -0.09904  0.226175  0.35220 0.950 5.92e-02 0.0415
## 3  -0.10077  0.175671  0.23112 1.068 2.68e-02 0.0578
## 4  -0.07742  0.152330  0.21739 1.054 2.37e-02 0.0479
## 5  -0.06919  0.112478  0.14069 1.114 1.01e-02 0.0676
## 6  -0.06264  0.123252  0.17589 1.071 1.56e-02 0.0479
## 7  -0.04182  0.107860  0.17853 1.051 1.60e-02 0.0384
## 8  -0.00185  0.003072  0.00390 1.126 7.80e-06 0.0643
## 9  -0.01220  0.054035  0.10766 1.069 5.89e-03 0.0326
## 10 -0.01262  0.028819  0.04488 1.096 1.03e-03 0.0415
## 11  0.03561  0.013467  0.12042 1.048 7.33e-03 0.0247
## 12  0.05796 -0.005484  0.13221 1.041 8.81e-03 0.0244
## 13  0.01593  0.022859  0.09481 1.062 4.57e-03 0.0259
## 14 -0.00126  0.005581  0.01112 1.089 6.34e-05 0.0326
## 15  1.09618 -1.004332  1.09618 0.897 5.24e-01 0.1519   *
## 16  0.04755 -0.017337  0.08091 1.067 3.34e-03 0.0256
## 17  0.00403 -0.017845 -0.03555 1.087 6.48e-04 0.0326
## 18  0.00152  0.002181  0.00904 1.081 4.20e-05 0.0259
## 19  0.01960 -0.001854  0.04470 1.075 1.02e-03 0.0244
## 20  0.00114  0.001634  0.00678 1.081 2.36e-05 0.0259
## 21  0.01033 -0.000978  0.02357 1.078 2.85e-04 0.0244
## 22  0.94865 -0.869161  0.94865 0.969 4.08e-01 0.1519   *
## 23  1.00789 -1.310377 -1.39583 0.923 8.34e-01 0.2055   *
## 24  0.02187 -0.068245 -0.12227 1.069 7.59e-03 0.0354
## 25  0.02619 -0.014169  0.03589 1.082 6.60e-04 0.0289
## 26 -0.01820  0.001722 -0.04152 1.076 8.83e-04 0.0244
## 27  0.03874 -0.120855 -0.21653 1.022 2.33e-02 0.0354
## 28 -0.00557  0.003712 -0.00656 1.092 2.20e-05 0.0359
## 29 -0.04029  0.014689 -0.06856 1.071 2.40e-03 0.0256
## 30  0.02549 -0.019468  0.02728 1.108 3.82e-04 0.0497
## 31 -0.05873  0.021410 -0.09992 1.060 5.07e-03 0.0256
## 32 -0.05522  0.036829 -0.06503 1.086 2.16e-03 0.0359
## 33 -0.05810  0.044373 -0.06217 1.104 1.98e-03 0.0497
## 34 -0.07889  0.060247 -0.08441 1.101 3.64e-03 0.0497
## 35 -0.10865  0.058770 -0.14886 1.043 1.12e-02 0.0289
## 36 -0.15339  0.102306 -0.18065 1.044 1.64e-02 0.0359
## 37 -0.18934  0.126283 -0.22299 1.020 2.46e-02 0.0359
## 38 -0.22354  0.170718 -0.23919 1.047 2.85e-02 0.0497
## 39 -0.22354  0.170718 -0.23919 1.047 2.85e-02 0.0497
## 40 -0.27207  0.181461 -0.32042 0.950 4.91e-02 0.0359
## 41 -0.73865  0.622960 -0.75075 0.828 2.46e-01 0.0783   *
```

```r
par(mfrow=c(2,2))
plot(lm.ad,which=1:4)
```




```r
# It shows that these points have high standardized residuals and
#high cook distance, meaning they are outliers with high influence.
```

```r
# robust fit deleting four outliers
lm.robust <- lm(log(R) ~ log(P), data = ad[-c(15,22,23,41),])
summary(lm.robust)
```

```
##
## Call:
## lm(formula = log(R) ~ log(P), data = ad[-c(15, 22, 23, 41), ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.2759 -0.3202  0.1032  0.3523  1.0137
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.9941     0.2845  -3.494  0.00131 **
## log(P)        1.4351     0.1318  10.891 8.67e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5185 on 35 degrees of freedom
## Multiple R-squared:  0.7722, Adjusted R-squared:  0.7657
## F-statistic: 118.6 on 1 and 35 DF,  p-value: 8.667e-13
```

After deleting four outliers, the robust regression has adjusted R-squared of 76.57%, comparing to previous of

40.55%, which is great improvement in terms of model fitness.

Problem 5 Exercise 13.5 from the TEXT.

a) Fit a Poisson regression model to relate Success with the Distance from which the kick is taken. Use Attempts as offset.

```
glm.goal.Poisson <- glm(Success ~ Distance, offset = log(Attempts), data = goal, family = poisson)
summary(glm.goal.Poisson)
```

```
##
## Call:
## glm(formula = Success ~ Distance, family = poisson, data = goal,
##     offset = log(Attempts))
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.82704  -0.79951   0.01202   0.90292   1.16179
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.553215   0.123134   4.493 7.03e-06 ***
## Distance    -0.038326   0.004133  -9.274  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 103.831  on 9  degrees of freedom
## Residual deviance:  14.219  on 8  degrees of freedom
## AIC: 70.643
##
## Number of Fisher Scoring iterations: 4
```

b) Fit a logistic model relating the probability of a successful kick to the distance from which the kick is taken.

```
glm.goal.logistic <- glm(Ymat ~ Distance, data = goal, family = binomial)
summary(glm.goal.logistic)
```

```
##
## Call:
## glm(formula = Ymat ~ Distance, family = binomial, data = goal)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9931  -0.4015   0.2430   0.4288   1.6741
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.678716   0.293068   12.55   <2e-16 ***
## Distance    -0.103212   0.008098  -12.74   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 228.5180  on 9  degrees of freedom
## Residual deviance:   9.3713  on 8  degrees of freedom
## AIC: 55.761
##
## Number of Fisher Scoring iterations: 4
```

c) Show that the logistic model gives a better fit than the Poisson regression model.

```
AIC(glm.goal.Poisson)
```

```
## [1] 70.64296
```

```
AIC(glm.goal.logistic)
```

```
## [1] 55.76081
```

In terms of AIC, logistic regression model has lower AIC than Poisson model, thus, it has better model fit.