Homework 5

Subhankar Ghosh

ghosh17

```
library(faraway)
library(ggplot2)
library(rpart)
library(ResourceSelection)
```

Question 1

(c)

Fit binary regression model

```
bin_modl = glm(Class ~ ., family = "binomial", data = wbca)
summary(bin_modl)
```

```
##
## Call:
## glm(formula = Class ~ ., family = "binomial", data = wbca)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.48282  -0.01179   0.04739   0.09678   3.06425
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 11.16678    1.41491   7.892 2.97e-15 ***
## Adhes       -0.39681    0.13384  -2.965  0.00303 **
## BNucl       -0.41478    0.10230  -4.055 5.02e-05 ***
## Chrom       -0.56456    0.18728  -3.014  0.00257 **
## Epith       -0.06440    0.16595  -0.388  0.69795
## Mitos       -0.65713    0.36764  -1.787  0.07387 .
## NNucl       -0.28659    0.12620  -2.271  0.02315 *
## Thick       -0.62675    0.15890  -3.944 8.01e-05 ***
## UShap       -0.28011    0.25235  -1.110  0.26699
## USize        0.05718    0.23271   0.246  0.80589
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 881.388  on 680  degrees of freedom
## Residual deviance:  89.464  on 671  degrees of freedom
## AIC: 109.46
##
## Number of Fisher Scoring iterations: 8
```

Residual deviance for Full model is 89.464 with a degree of freedom of 671

Using this data of residual deviance we can say if the model fits the data well but of course we cannot say if a better model/smaller model exists or not using just this data. We know according to approximation theory Deviance D is approximately χ^2 distributed with n-(p+1) degrees of freedom if the number of cases for each class in Y (response variable) is greater than at least 5, Since this data satisfies this assumption we can use Deviance to test whether the model fits well.

Performing lack of fit test using Deviance

```
pchisq(deviance(bin_modl), df.residual(bin_modl), lower = FALSE)
```

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

The p-value is 1 so we can say that it is a very good fit for the data.

(d)

Using step function to find the best subset of variables with AIC criterion

```
model.aic = step(bin_modl, direction = "backward", trace = 1)
```

```
## Start:  AIC=109.46
## Class ~ Adhes + BNucl + Chrom + Epith + Mitos + NNucl + Thick +
##     UShap + USize
##
##         Df Deviance    AIC
## - USize  1   89.523 107.52
## - Epith  1   89.613 107.61
## - UShap  1   90.627 108.63
## <none>       89.464 109.46
## - Mitos  1   93.551 111.55
## - NNucl  1   95.204 113.20
## - Adhes  1   98.844 116.84
## - Chrom  1   99.841 117.84
## - BNucl  1  109.000 127.00
## - Thick  1  110.239 128.24
##
## Step:  AIC=107.52
## Class ~ Adhes + BNucl + Chrom + Epith + Mitos + NNucl + Thick +
##     UShap
##
##         Df Deviance    AIC
## - Epith  1   89.662 105.66
## - UShap  1   91.355 107.36
## <none>       89.523 107.52
## - Mitos  1   93.552 109.55
## - NNucl  1   95.231 111.23
## - Adhes  1   99.042 115.04
## - Chrom  1  100.153 116.15
## - BNucl  1  109.064 125.06
## - Thick  1  110.465 126.47
##
## Step:  AIC=105.66
## Class ~ Adhes + BNucl + Chrom + Mitos + NNucl + Thick + UShap
```

```
##
##         Df Deviance    AIC
## <none>       89.662 105.66
## - UShap  1   91.884 105.88
## - Mitos  1   93.714 107.71
## - NNucl  1   95.853 109.85
## - Adhes  1  100.126 114.13
## - Chrom  1  100.844 114.84
## - BNucl  1  109.762 123.76
## - Thick  1  110.632 124.63
```

We were successful in finding a smaller model with 7 variables: Class ~ Adhes + BNucl + Chrom + Mitos + NNucl + Thick + UShap. Let us see the residual deviance

```
summary(model.aic)
```

```
##
## Call:
## glm(formula = Class ~ Adhes + BNucl + Chrom + Mitos + NNucl +
##     Thick + UShap, family = "binomial", data = wbca)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.44161  -0.01119   0.04962   0.09741   3.08205
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  11.0333     1.3632   8.094 5.79e-16 ***
## Adhes        -0.3984     0.1294  -3.080  0.00207 **
## BNucl        -0.4192     0.1020  -4.111 3.93e-05 ***
## Chrom        -0.5679     0.1840  -3.085  0.00203 **
## Mitos        -0.6456     0.3634  -1.777  0.07561 .
## NNucl        -0.2915     0.1236  -2.358  0.01837 *
## Thick        -0.6216     0.1579  -3.937 8.27e-05 ***
## UShap        -0.2541     0.1785  -1.423  0.15461
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 881.388  on 680  degrees of freedom
## Residual deviance:  89.662  on 673  degrees of freedom
## AIC: 105.66
##
## Number of Fisher Scoring iterations: 8
```

The residual deviance does not change much it is 89.662 with 673 degrees of freedom. Lets see the p-value for a chisquare test statistic.

```
pchisq(deviance(model.aic), df.residual(model.aic), lower = FALSE)
```

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

We get the same value, both the models fit the data very well.

(e)

Predict probabilities from reduced model and find misclassifications with 0.5 as cut-off.

```
reduced.predicted = predict(model.aic, newdata = wbca, type = "response")
pred = ifelse(reduced.predicted > 0.5, 1, 0)
sum(pred!=wbca$Class)
```

```
## [1] 20
```

There would be a total of 20 misclassified samples when 0.5 is chosen as the cut-off.

(f)

Predict probabilities from reduced model and find misclassifications with 0.9 as cut-off.

```
reduced.predicted2 = predict(model.aic, newdata = wbca, type = "response")
pred2 = ifelse(reduced.predicted2 > 0.9, 1, 0)
sum(pred2!=wbca$Class)
```

```
## [1] 17
```

There would be a total of 17 misclassified samples when 0.5 is chosen as the cut-off.

(h)

Create test set

```
testind = seq(3, nrow(wbca), by = 3)
```

Refit model

```
model.aic = update(model.aic, subset = -testind)
```

Misclassification based on 0.5 cut-off

```
test.prediction = predict(model.aic, newdata = wbca[testind,], type = "response")
pred = ifelse(test.prediction > 0.5, 1, 0)
sum(pred!=wbca$Class[testind])
```

```
## [1] 7
```

Misclassification based on 0.9 cut-off

```
pred2 = ifelse(test.prediction > 0.9, 1, 0)
sum(pred2!=wbca$Class[testind])
```

```
## [1] 5
```

With the test set the outcome for both the cut-offs 0.5 and 0.7 seem similar as the number of misclassifications are 7 and 5 respectively.

Question 2

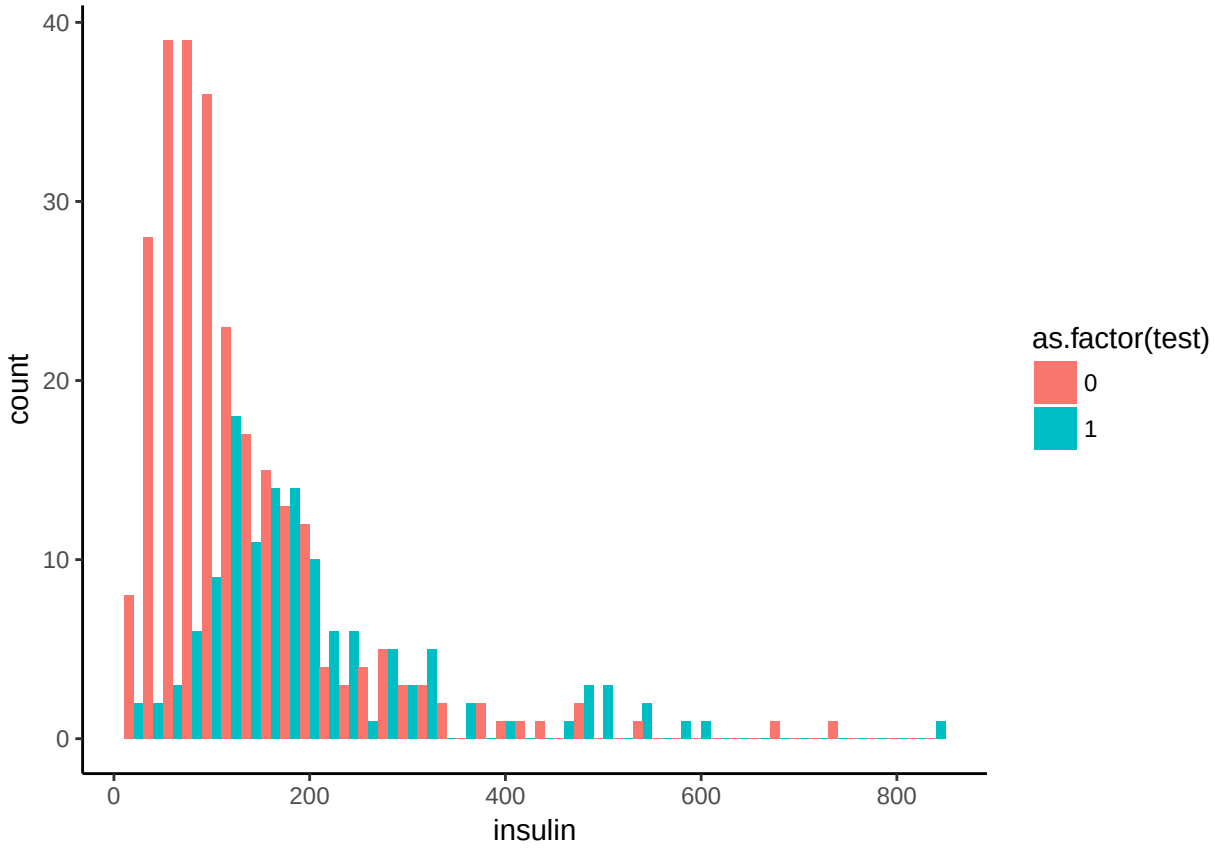
(b)

Plot histogram after setting values of insulin to NA in rows having insulin = 0

```
pima_data = pima
pima_data[pima_data$insulin == 0, "insulin"] = NA
```

```
p = ggplot(pima_data, aes(x = insulin, fill = as.factor(test)))
p = p + geom_histogram(binwidth = 20, position = "dodge")
p = p + theme(panel.background = element_blank(), axis.line = element_line(colour = "black"))
p
```

```
## Warning: Removed 374 rows containing non-finite values (stat_bin).
```



The plot we get after removal of 374 rows(done automatically by R).

- seems to be skewed distribution for test = 0. This distribution has very high frequencies of low values of insulin, thus showing that the test = 0 or negative when insulin levels are low. We also get very few cases where insulin levels are very high but the test value is 0, this leads to a skewed distribution.

- Distribution where test = 0 looks to have high frequencies in the middle range of insulin (between 100 to 250). This distribution looks like somewhat normal. Even this distribution has skewed values with very high values of insulin with test = 1.

(c)

```
pima_data[pima_data$glucose == 0, "glucose"] = NA
pima_data[pima_data$diastolic == 0, "diastolic"] = NA
pima_data[pima_data$triceps == 0, "triceps"] = NA
pima_data[pima_data$age == 0, "age"] = NA
```

```
diabetes.model = glm(test ~ ., family = "binomial", data = pima_data)
summary(diabetes.model)
```

```
##
## Call:
## glm(formula = test ~ ., family = "binomial", data = pima_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7742  -0.6611  -0.3622   0.6394   2.5617
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.007e+01  1.215e+00  -8.284  < 2e-16 ***
## pregnant     8.253e-02  5.543e-02   1.489  0.13651
## glucose      3.828e-02  5.770e-03   6.635 3.25e-11 ***
## diastolic   -1.561e-03  1.182e-02  -0.132  0.89491
## triceps      1.082e-02  1.702e-02   0.636  0.52485
## insulin     -8.296e-04  1.307e-03  -0.635  0.52553
## bmi          7.186e-02  2.687e-02   2.675  0.00748 **
## diabetes     1.129e+00  4.250e-01   2.657  0.00787 **
## age          3.415e-02  1.837e-02   1.859  0.06301 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 498.90  on 392  degrees of freedom
## Residual deviance: 344.14  on 384  degrees of freedom
##   (375 observations deleted due to missingness)
## AIC: 362.14
##
## Number of Fisher Scoring iterations: 5
```

392 observations were used to fit the model.

Only 392 observations were used to fit the model even though 768 observations were there in the dataset because 376 observations were removed by R because they contained missing values. These rows are the ones which we marked as NA for values corresponding to 0 in age, bmi, insulin, triceps, diastolic, glucose columns. These data seemed to be wrong data so we marked them as NA and R removed them

(d)

```
diabetes.model2 = glm(test ~ pregnant + glucose + diastolic + bmi + diabetes + age, family = "binomial"
summary(diabetes.model2)
```

```
##
```

```
## Call:
## glm(formula = test ~ pregnant + glucose + diastolic + bmi + diabetes +
##     age, family = "binomial", data = pima_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7816  -0.7259  -0.4070   0.7194   2.4866
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -8.849809   0.810974 -10.913  < 2e-16 ***
## pregnant     0.117828   0.033195   3.550 0.000386 ***
## glucose      0.035024   0.003587   9.764  < 2e-16 ***
## diastolic   -0.006645   0.008552  -0.777 0.437176
## bmi          0.083975   0.015203   5.524 3.32e-08 ***
## diabetes     0.940630   0.303294   3.101 0.001926 **
## age          0.016737   0.009759   1.715 0.086346 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 936.60  on 727  degrees of freedom
## Residual deviance: 679.58  on 721  degrees of freedom
##   (40 observations deleted due to missingness)
## AIC: 693.58
##
## Number of Fisher Scoring iterations: 5
```

This time 724 observations were used.

Comparing the two models using Chi-square p-value derived from the Deviance of each of the models.

```
pchisq(deviance(diabetes.model), df.residual(diabetes.model), lower = FALSE)
```

```
## [1] 0.9288084
```

```
pchisq(deviance(diabetes.model2), df.residual(diabetes.model2), lower = FALSE)
```

```
## [1] 0.8631721
```

Since the p-value for the first model (with all parameters) is larger than that of the second model (with insulin and triceps removed) we can conclude that the first model fits the data better. Also from the AIC score of the two models: First model AIC score = 362.02 and Second model AIC score = 686.86 we conclude the same result that the first model fits the data better.

(e)

```
na.removed = na.omit(pima_data)
full_model = glm(test ~ ., family = "binomial", data = na.removed)
model.aic2 = step(full_model, direction = "backward", trace = 1)
```

```
## Start:  AIC=362.14
```

```
## test ~ pregnant + glucose + diastolic + triceps + insulin + bmi +
##     diabetes + age
##
##             Df Deviance    AIC
## - diastolic  1   344.16 360.16
## - insulin    1   344.54 360.54
## - triceps    1   344.55 360.55
## <none>           344.14 362.14
## - pregnant   1   346.38 362.38
## - age        1   347.72 363.72
## - diabetes   1   351.61 367.61
## - bmi        1   351.76 367.76
## - glucose    1   397.07 413.07
##
## Step:  AIC=360.16
## test ~ pregnant + glucose + triceps + insulin + bmi + diabetes +
##     age
##
##            Df Deviance    AIC
## - insulin   1   344.55 358.55
## - triceps   1   344.56 358.56
## <none>          344.16 360.16
## - pregnant  1   346.39 360.39
## - age       1   347.76 361.76
## - diabetes  1   351.70 365.70
## - bmi       1   352.15 366.15
## - glucose   1   397.42 411.42
##
## Step:  AIC=358.55
## test ~ pregnant + glucose + triceps + bmi + diabetes + age
##
##            Df Deviance    AIC
## - triceps   1   344.98 356.98
## <none>          344.55 358.55
## - pregnant  1   346.89 358.89
## - age       1   348.03 360.03
## - diabetes  1   351.94 363.94
## - bmi       1   352.17 364.17
## - glucose   1   411.21 423.21
##
## Step:  AIC=356.98
## test ~ pregnant + glucose + bmi + diabetes + age
##
##            Df Deviance    AIC
## <none>          344.98 356.98
## - pregnant  1   347.35 357.35
## - age       1   348.84 358.84
## - diabetes  1   352.74 362.74
## - bmi       1   361.57 371.57
## - glucose   1   411.92 421.92
```

```
#392
summary(model.aic2)
```

```
##
```

```
## Call:
## glm(formula = test ~ pregnant + glucose + bmi + diabetes + age,
##     family = "binomial", data = na.removed)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8770  -0.6528  -0.3680   0.6498   2.5818
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -10.015785   1.083499  -9.244  < 2e-16 ***
## pregnant      0.084227   0.055035   1.530 0.125915
## glucose       0.036458   0.004979   7.323 2.43e-13 ***
## bmi           0.078844   0.020398   3.865 0.000111 ***
## diabetes      1.141160   0.422032   2.704 0.006852 **
## age           0.034447   0.017809   1.934 0.053081 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 498.90  on 392  degrees of freedom
## Residual deviance: 344.98  on 387  degrees of freedom
## AIC: 356.98
##
## Number of Fisher Scoring iterations: 5
```

The final model that was selected has 5 predictor variables: test ~ pregnant + glucose + bmi + diabetes + age with an AIC of 356.89

Since all missing valued rows were removed so 392 rows were used to train the model.

(f)

Creating column that will have value 1 if missing values is present in that row else 0

```
na.ind = unique(which(is.na(pima_data), arr.ind = TRUE)[, 1])
```

```
pima_data$isna = 0
pima_data[na.ind, "isna"] = 1
```

Fitting model with this newly created variable "isna" as predictor to predict "test"

```
model.with.na = glm(test ~ isna, family = "binomial", data = pima_data)
summary(model.with.na)
```

```
##
## Call:
## glm(formula = test ~ isna, family = "binomial", data = pima_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9580  -0.9580  -0.8963   1.4140   1.4875
##
```

```
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.7046     0.1072  -6.572 4.96e-11 ***
## isna          0.1638     0.1515   1.081     0.28
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 993.48  on 767  degrees of freedom
## Residual deviance: 992.31  on 766  degrees of freedom
## AIC: 996.31
##
## Number of Fisher Scoring iterations: 4
```

We can conclude from the p-value = 0.304 that the missingness of the data is not associated with the response (test result).

```
refit.aic = glm(formula = test ~ pregnant + glucose + bmi + diabetes + age,
    family = "binomial", data = pima_data)
summary(refit.aic)
```

```
##
## Call:
## glm(formula = test ~ pregnant + glucose + bmi + diabetes + age,
##     family = "binomial", data = pima_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7828  -0.7191  -0.4117   0.7052   2.4779
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -8.905130   0.703597 -12.657  < 2e-16 ***
## pregnant     0.120949   0.032059   3.773 0.000162 ***
## glucose      0.035921   0.003538  10.152  < 2e-16 ***
## bmi          0.076312   0.013896   5.492 3.98e-08 ***
## diabetes     0.891054   0.296215   3.008 0.002629 **
## age          0.010328   0.009201   1.123 0.261631
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 986.70  on 762  degrees of freedom
## Residual deviance: 716.07  on 757  degrees of freedom
##   (5 observations deleted due to missingness)
## AIC: 728.07
##
## Number of Fisher Scoring iterations: 5
```

Now we have fitted the same model selected by AIC criteria with the entire dataset and only 16 observations were removed thus using 752 observations.

This was necessary because when we were doing variable selection we had started with a full model (all variables included) and therefore we had to prune our dataset of all the rows con-

taining any missing values so our model was trained on only 392 observations. This happened because we had removed rows which contain NA on predictors triceps and insulin - predictors not included in our above model.

(g)

We will start with the interpretation of the coefficients of the model

Estimate coefficient for bmi is 0.087529 abd the standard error for bmi is 0.014722

Let us find the difference in the 75percentile and 25th percentile in bmi values

```
diff.bmi = with(pima_data, quantile(bmi, 0.75, na.rm = TRUE) - quantile(bmi, 0.25, na.rm = TRUE))
diff.bmi
```

```
## 75%
## 9.3
```

Difference in odds of testing positive for diabetes in women with BMI at first quartile compared to that in third quartile is given by th following. It equals 2.218

```
exp(0.087529*diff.bmi)
```

```
##      75%
## 2.256962
```

Create a confidence interval for this difference

```
diff.logodds = 0.087529*diff.bmi
se.diff.logodds = 0.014722*diff.bmi
CI = exp(c(diff.logodds - 1.96*se.diff.logodds, diff.logodds, diff.logodds + 1.96*se.diff.logodds))
CI
```

```
##      75%      75%      75%
## 1.725759 2.256962 2.951674
```

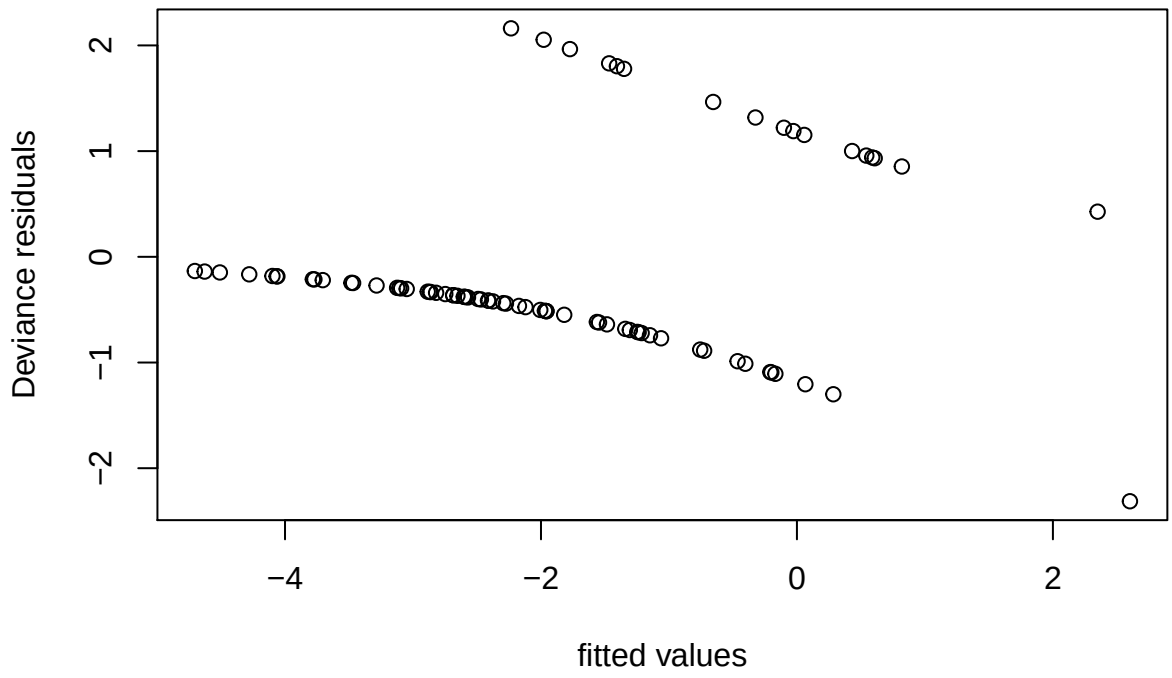
Predicted difference in odds between prediction of diabetes with BMI in first and third quartile is 2.217796

Confidence interval for the difference in odds between prediction of diabetes with BMI in first and third quartile is 1.705626 to 2.883761

Question 3

(b)

```
kyp.model = glm(Kyphosis ~ ., family = "binomial", data = kyphosis)
res = residuals(kyp.model, type = "deviance")
plot(predict(kyp.model), res, xlab="fitted values", ylab="Deviance residuals")
```



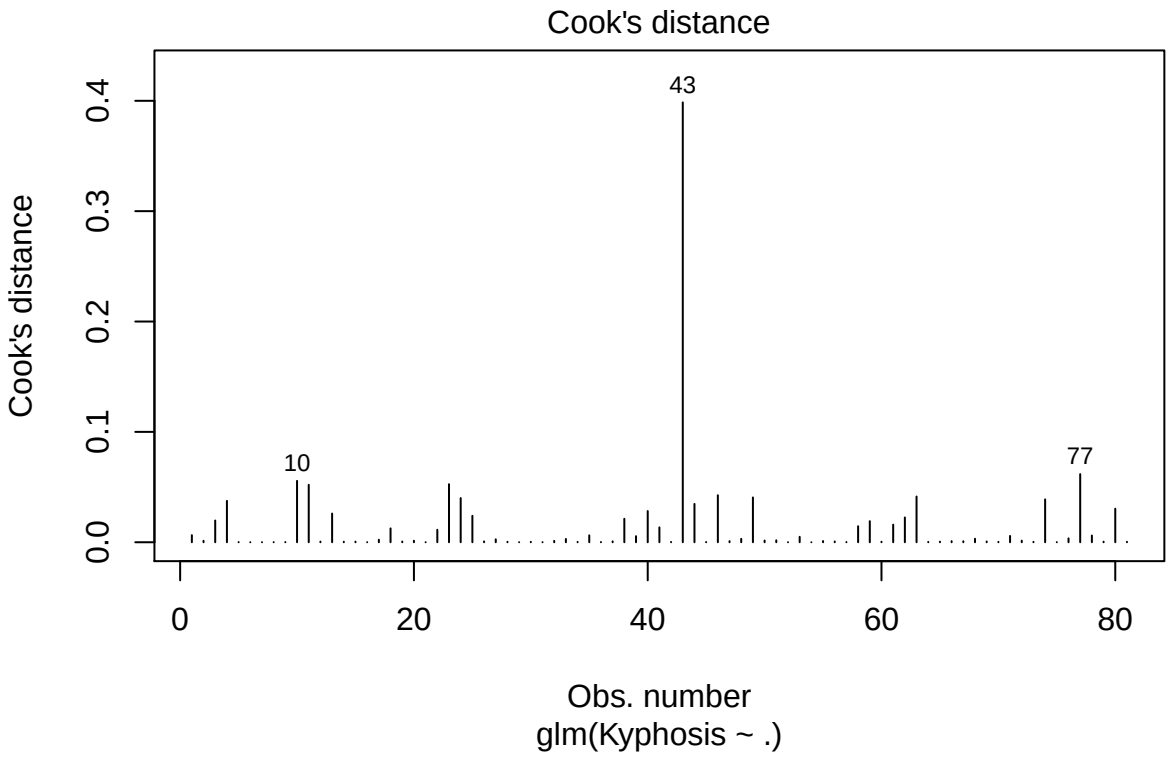
We can see a some non linearity in the Deviance residuals as plotted against the fitted values. The negative residuals especially form a curve as value of fitted values increase.

We can also see that variance of the deviance residuals are not constant across fitted values.

(f)

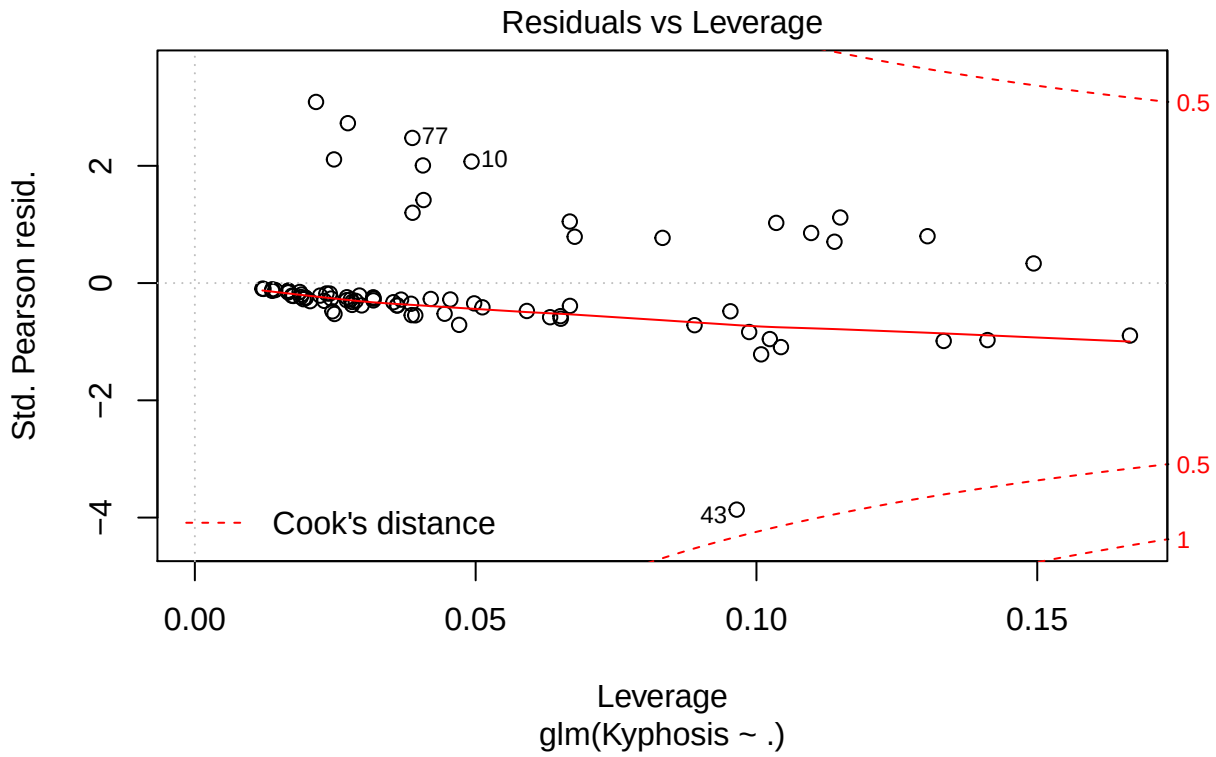
Looking at leverage plots

```
plot(kyp.model, which=c(4))
```



From the Cook's distance vs Observation plot we can clearly see 43rd observation having high Cook's distance thus high leverage. We should also investigate 77 and 10th observation.

```
plot(kyp.model, which=c(5))
```



From the standard Pearson residual vs leverage plot we can clearly note that 43 is an outlier. It has both high leverage as well as high residual thus it is definitely a high influence point. 77 and 10 are closely plotted in the plot they are high potential points to be influencial but not as influential as 43.

(h)

```
predprob = predict(kyp.model, type = "response")
predictions = ifelse(predprob > 0.5, "Present", "Absent")
table(predictions, kyphosis$Kyphosis)
```

```
##
## predictions absent present
##     Absent      61      10
##     Present      3       7
```

From the table we can see that when the observation is actually present the number of correct prediction is 7 out of 10. SO the probability that the model would predict kyphosis present when the kyphosis is actually present is 0.7. This metric is called *Recall*