Department of Statistics

COURSE STATS 330/762

Model answers for Assignment 4, 2011

1. *Read the data into R, and make a data frame containing the variables. Check for any obvious outliers. Print out the first 16 lines. [5 marks]*

This data has no variable names in the file, so you will either have to add these or read the data in and rename the variables. E.g.

```
pima.df = read.table(file.choose() , sep=",")
names(pima.df)=c("pregnant","glucose","diastolic","triceps","insulin",
"bmi","diabetes","age","test")
```

There were no deliberate corruptions of this data set. However, plotting the data indicates that several variables (e.g.bmi) have a lot of very unlikely zeroes, which should be omitted before we start fitting models. The following code draws histograms for each variable:

```
par(mfrow=c(4,2))
for(i in 1:8)boxplot(pima.no.0.df[,i]~pima.no.0.df[,9], main =
names(pima.df)[i])
```

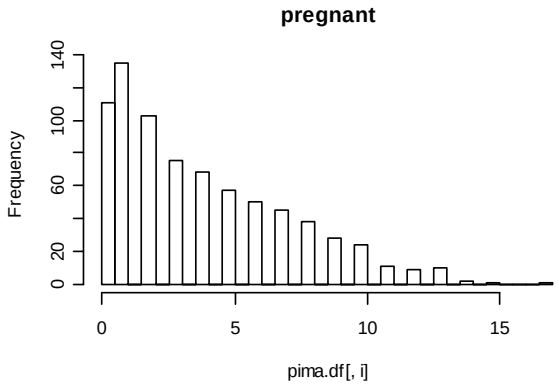
These are shown overleaf. The plots suggest that zeroes in the variables glucose, diastolic, triceps, insulin and bmi should be treated as missing values. We can delete these variables from the data using the following code:

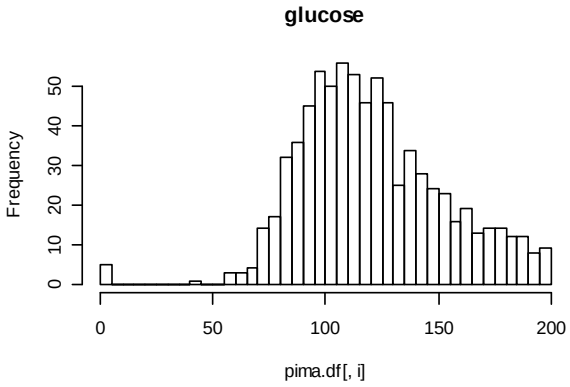
```
use = (pima.df$glucose>0)&(pima.df$diastolic>0)&(pima.df$triceps>0)&
(pima.df$insulin>0)&(pima.df$bmi>0)
pima.no.0.df = pima.df[use,]
```

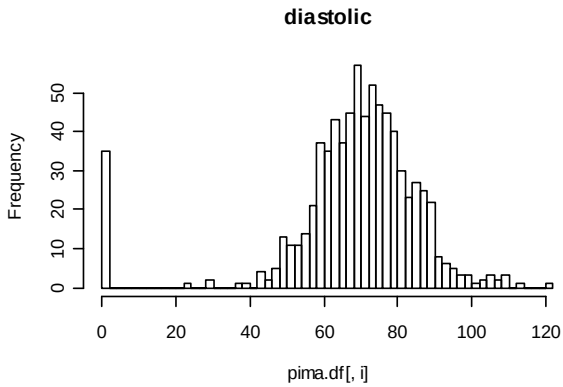
This discards quite a bit of data. However, the distribution of the responses is roughly the same for the data retained and the data excluded:

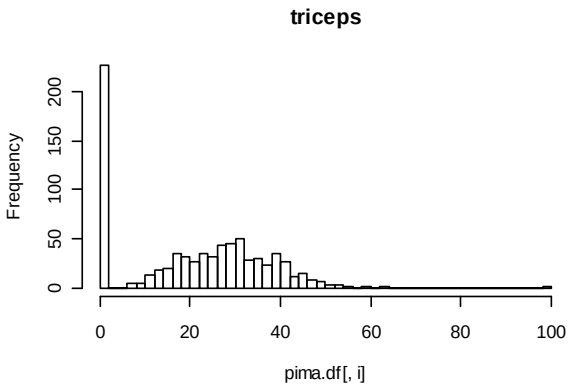
```
> table(use, pima.df$test)

use       0   1
  FALSE 145  91
  TRUE  355 177
> 91/45
[1] 2.022222
> 355/177
[1] 2.005650
```

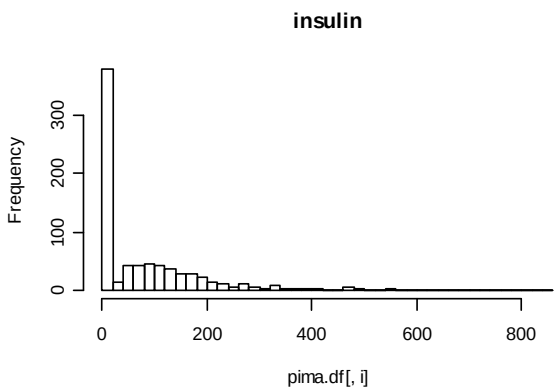
[5 marks: 1 for reading in, 1 for printing out, 3 for eliminating zeros.]

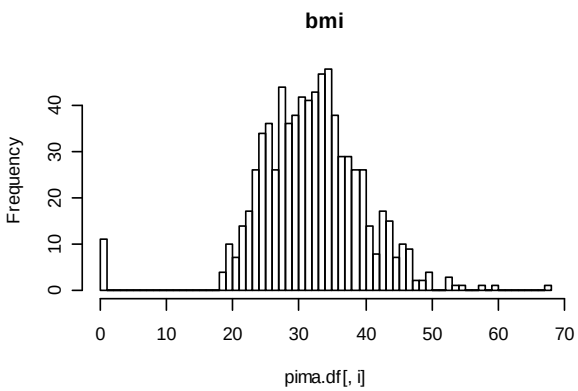

pregnant
Frequency
0
20
60
100
140
0
5
10
15
pima.df[, i]

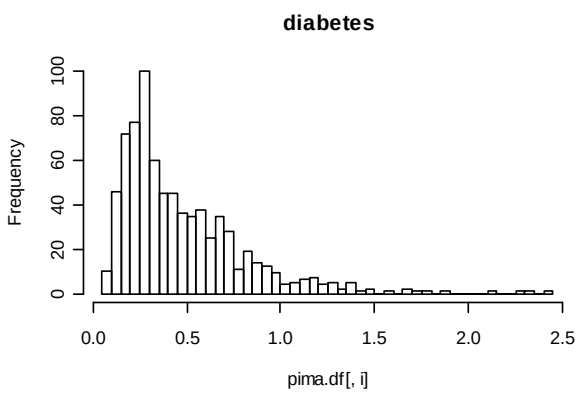

glucose
Frequency
0
10
20
30
40
50
0
50
100
150
200
pima.df[, i]

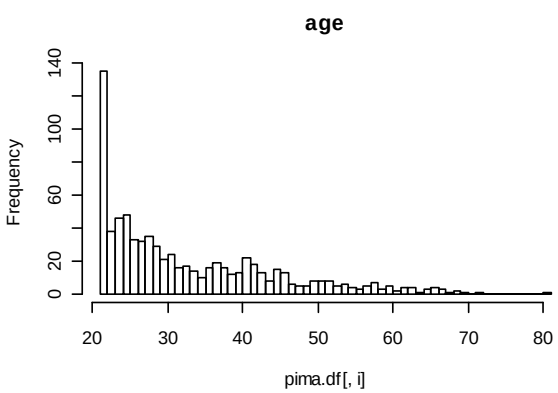

diastolic
Frequency
0
10
20
30
40
50
0
20
40
60
80
100
120
pima.df[, i]


triceps
Frequency
0
50
100
150
200
0
20
40
60
80
100
pima.df[, i]


insulin
Frequency
0
100
200
300
0
200
400
600
800
pima.df[, i]


bmi
Frequency
0
10
20
30
40
0
10
20
30
40
50
60
70
pima.df[, i]


diabetes
Frequency
0
20
40
60
80
100
0.0
0.5
1.0
1.5
2.0
2.5
pima.df[, i]

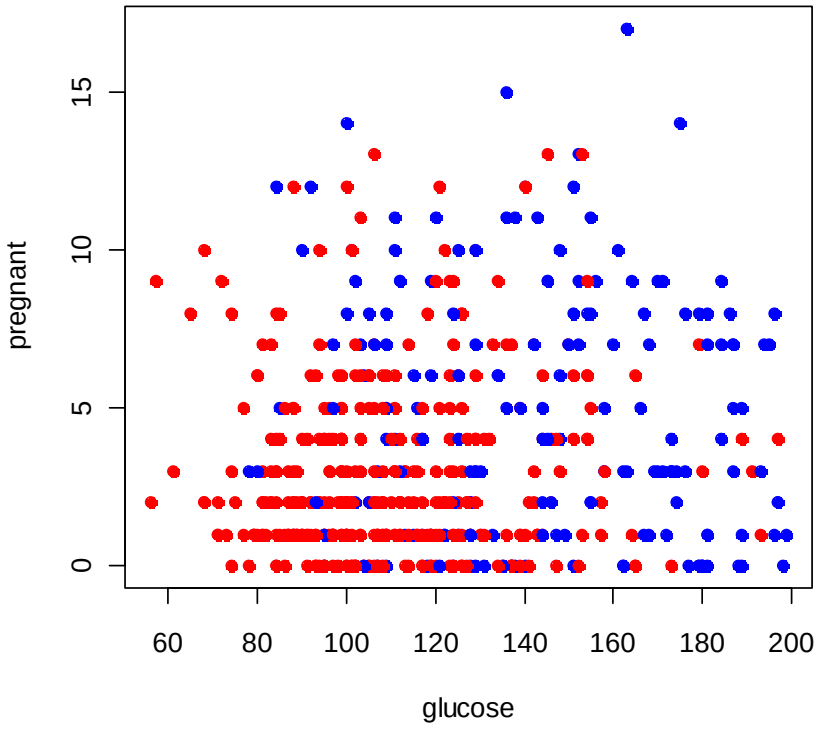

age
Frequency
0
20
60
100
140
20
30
40
50
60
70
80
pima.df[, i]

2. *Make some plots that will let you see how the various explanatory variables affect the response. What are these effects, if any? [5 marks]*

Suitable plots include (a) plotting two explanatory variables against eachother, with test indicated by a colour code, and (b) side-by -side boxplots of the explanatory variables , grouping by test. e.g.

```
colour = c("red","blue")
plot(pregnant~glucose, data=pima.no.0.df, type="n")
points(pima.no.0.df$glucose,pima.no.0.df$pregnant,col=colour[pima.
no.0.df$test+1], pch=19)
```

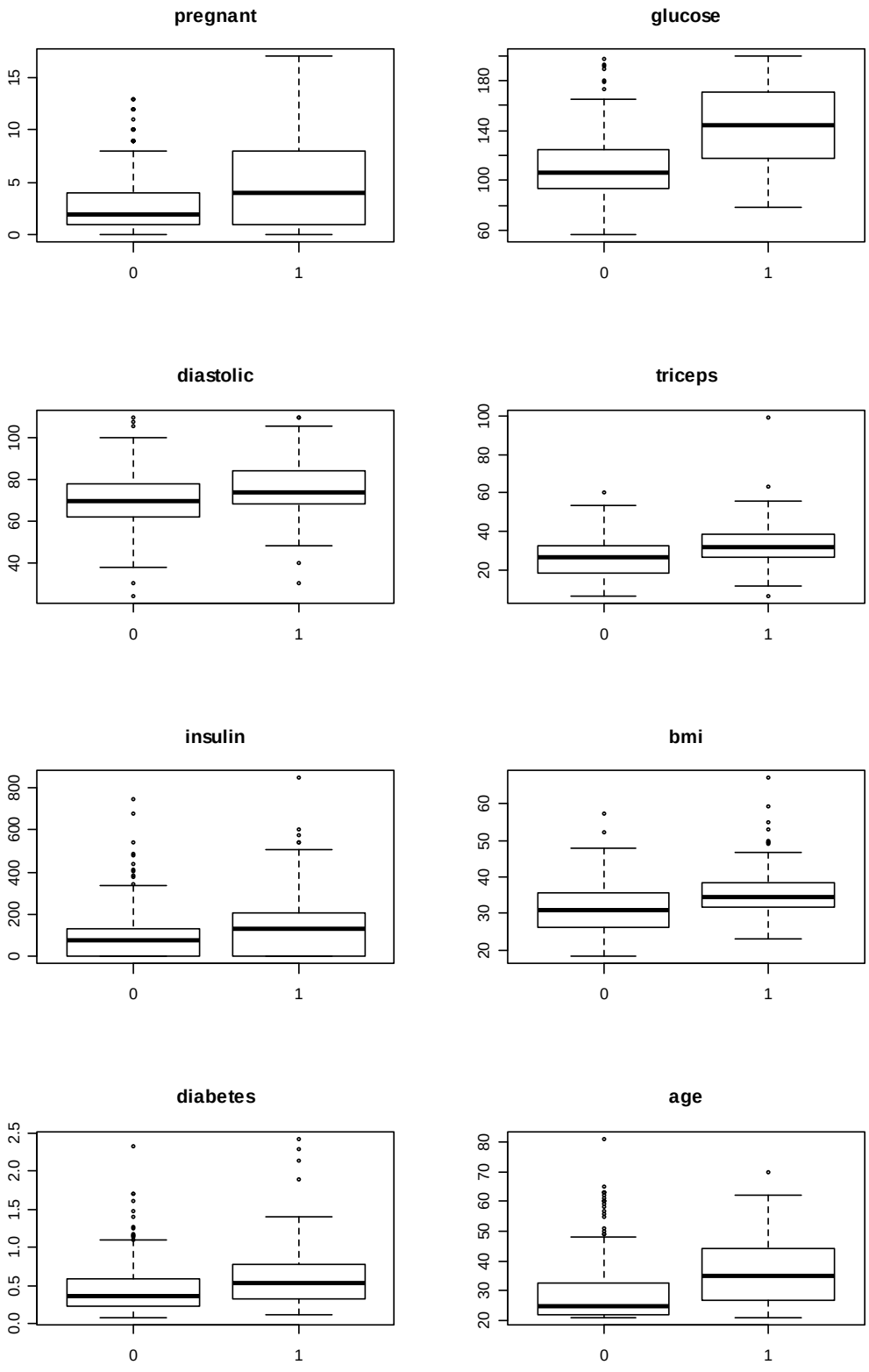
which produces



Side by side box-plots can be drawn using the code

```
par(mfrow=c(4,2))
for(i in 1:8)boxplot(pima.no.0.df[,i]~pima.no.0.df[,9], main =
names(pima.df)[i])
```

which gives


pregnant
glucose
diastolic
triceps
insulin
bmi
diabetes
age

From this it seems that all the variables have an effect on the response, as there are differences between the boxplots for the two groups. The variables pregnant, glucose, age seem to have the biggest effects.

3. *Fit a model to the data, discarding variables and making transformations as appropriate. [10 marks]*

Fitting the model can be done by the code

```
> pima.glm = glm(test~., family=binomial,data=pima.no.0.df )
> summary(pima.glm)

Call:
glm(formula = test ~ ., family = binomial, data = pima.no.0.df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.8627  -0.6639  -0.3672   0.6347   2.4942

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -9.677562   1.005400  -9.626  < 2e-16 ***
pregnant     0.121235   0.043926   2.760 0.005780 **
glucose      0.037439   0.004765   7.857 3.92e-15 ***
diastolic   -0.009316   0.010446  -0.892 0.372494
triceps      0.006341   0.014853   0.427 0.669426
insulin     -0.001053   0.001007  -1.046 0.295651
bmi          0.085992   0.023661   3.634 0.000279 ***
diabetes     1.335764   0.365771   3.652 0.000260 ***
age          0.026430   0.013962   1.893 0.058371 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 676.79  on 531  degrees of freedom
Residual deviance: 465.23  on 523  degrees of freedom
AIC: 483.23

Number of Fisher Scoring iterations: 5
```

From this it seems that insulin is not required in the model – this is useful as this has the most missing values. A stepwise regression produces

```
> null.glm = glm(test~1, family=binomial,data=pima.no.0.df )
> chosen = step(null.glm, formula(pima.glm), direction = "both",
trace=0)
> chosen
```

```
Call:  glm(formula = test ~ glucose + pregnant + bmi + diabetes +
age,
    family = binomial, data = pima.no.0.df)

Coefficients:
(Intercept)      glucose     pregnant          bmi     diabetes
age
   -9.87999      0.03503      0.12389      0.08512      1.32155
0.02384

Degrees of Freedom: 531 Total (i.e. Null);  526 Residual
Null Deviance:      676.8
Residual Deviance: 467.1        AIC: 479.1
```

From this we conclude that we can drop the variables insulin, diastolic and triceps.

At this point we should add back in the cases that have zeroes on these 3 variables. We get more data on the other variables, but the zeroes in insulin, diastolic and triceps don't now matter as as these variables will not be used in the subsequent analysis. We now have 752 observations.

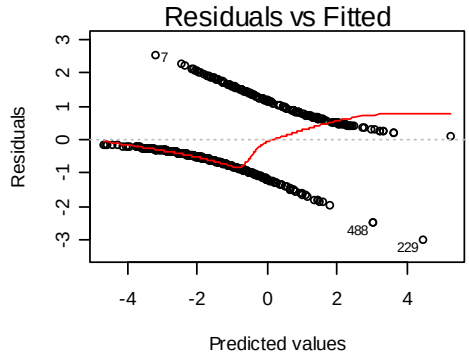
Running diagnostics shows no major problems. There are some outliers but these do not have much effect on the fitted coefficients and the p-values.

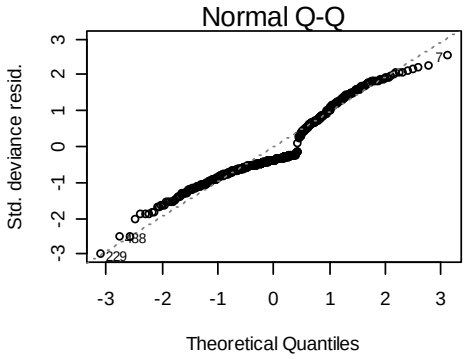
Diagnostic plots are shown overleaf.

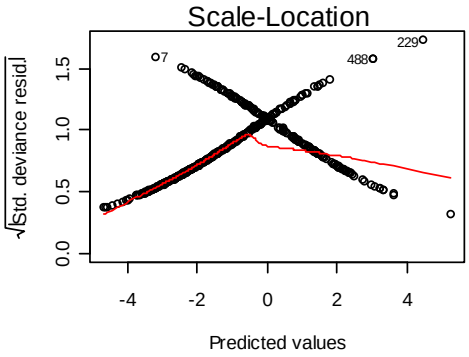
```
> plot(chosen)
> glm.diag.plots(chosen)
```

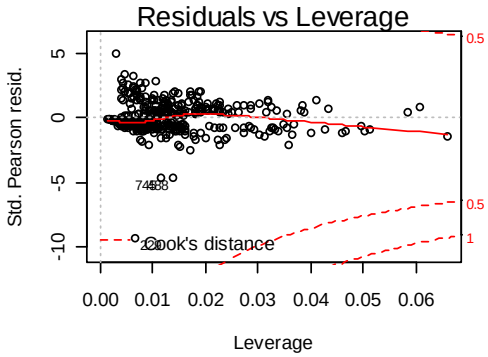
There are some high-influence/ large residual points – points 119, 318, 462, 154. We can explore the effect of these by fitting models and examining the effect on the coefficients. The following code does this, comparing the model will all points in, all out and the effect of deleting points one at a time

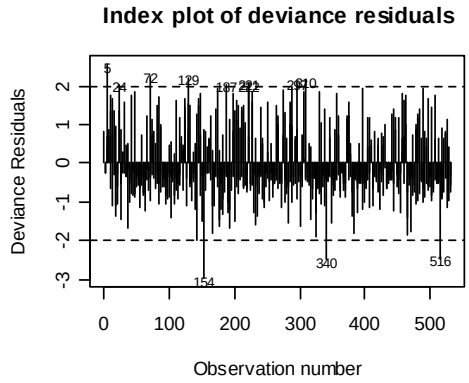
```
coefs1 = coefficients(chosen)
coefs2 = coefficients(glm(test ~ glucose + age + bmi + diabetes + pregnant
, family=binomial,data=pima.no.0.df, subset = -119 ))
coefs3 = coefficients(glm(test ~ glucose + age + bmi + diabetes + pregnant
, family=binomial,data=pima.no.0.df, subset = -154 ))
coefs4= coefficients(glm(test ~ glucose + age + bmi + diabetes + pregnant
, family=binomial,data=pima.no.0.df, subset = -318 ))
coefs5 = coefficients(glm(test ~ glucose + age + bmi + diabetes + pregnant
, family=binomial,data=pima.no.0.df, subset = -462 ))
coefs6 = coefficients(glm(test ~ glucose + age + bmi + diabetes + pregnant
, family=binomial,data=pima.no.0.df, subset = -c(119, 154, 318, 462) ))
coef.table = cbind(coefs1, coefs2, coefs3, coefs4, coefs5, coefs6)
```

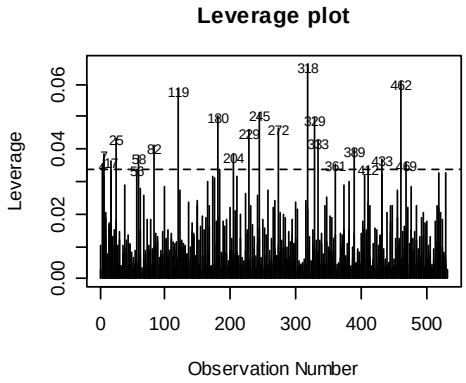

Residuals vs Fitted
Residuals
Predicted values

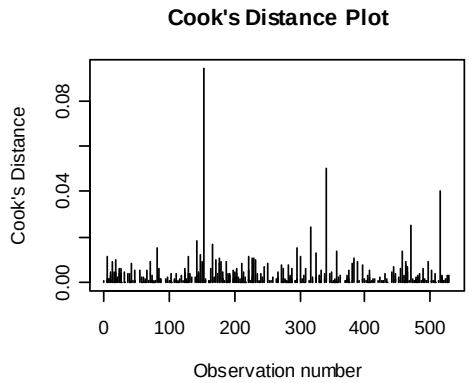

Normal Q-Q
Std. deviance resid.
Theoretical Quantiles

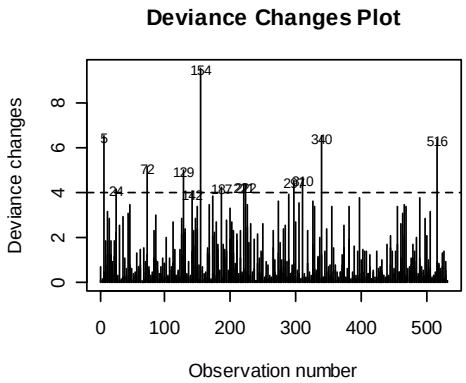

Scale-Location
√|Std. deviance resid.|
Predicted values


Residuals vs Leverage
Std. Pearson resid.
Cook's distance
Leverage


Index plot of deviance residuals
Deviance Residuals
Observation number


Leverage plot
Leverage
Observation Number


Cook's Distance Plot
Cook's Distance
Observation number


Deviance Changes Plot
Deviance changes
Observation number

```
colnames(coef.table) = c("None", 119, 154, 318, 462, "All")
> round(coef.table,3)
              None    119    154    318    462    All
(Intercept) -9.880 -9.324 -9.317 -9.312 -9.317 -9.303
glucose      0.035  0.036  0.036  0.036  0.036  0.036
pregnant     0.124  0.012  0.011  0.011  0.011  0.012
bmi          0.085  0.087  0.087  0.088  0.088  0.087
diabetes     1.322  0.912  0.919  0.915  0.915  0.899
age          0.024  0.113  0.115  0.115  0.114  0.112
```

The effect of these points is minimal, so we will leave them in. A gam plot indicates that bmi could be transformed by a cubic, so our final model is

```
test ~ glucose + age + poly(bmi,3) + diabetes + pregnant
```

[2 marks for fitting, 3 for model selection, 3 for discussion of outliers, 2 for transforming.]

4. *From your model, develop a rule that will allow you to predict if a female Pima Indian over 21 will be have diabetes. [10 marks]*

The prediction rule is

(a) Evaluate the fitted log-odds using the model test ~ glucose + age + poly(bmi,3) + diabetes + pregnant
(b) If the result is greater than 0.5, predict a "success" (i.e.predict diabetes, otherwise predict no diabetes.

5. *Evaluate the performance of your rule. [10 marks]*

We can evaluate the sensitivity and specificity on the training set by

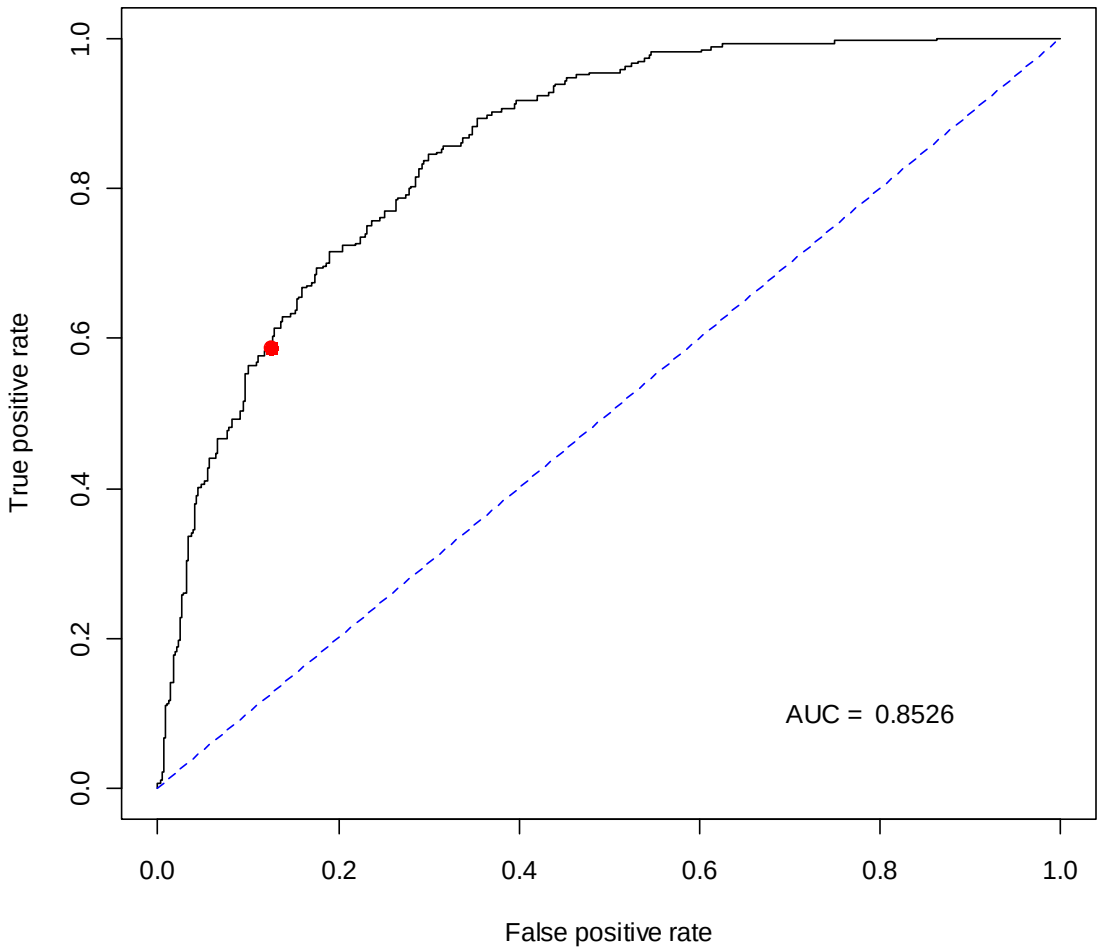
```
> poly.fit = glm(test ~ glucose + age + poly(bmi,3) + diabetes +
pregnant
+ , family=binomial,data=pima.no.0.df )
>
> pred = predict(poly.fit)
> table(pima.no.0.df$test==1,pred>0)

       FALSE TRUE
  FALSE   427   61
  TRUE    109  155
> 155/(155+109)
[1] 0.5871212
> 427/(427+61)
[1] 0.875
```

Cross-validated estimates are given by

```
> cross.val.glm(test ~ glucose + age + poly(bmi,3) + diabetes +
pregnant, data=pima.no.0.df)
Mean Specificity =  0.8732512
Mean Sensitivity =  0.58435
Mean Correctly classified =  0.7709333
```

The ROC curve is

```
> ROC.curve(test ~ glucose + age + poly(bmi,3) + diabetes + pregnant,
data=pima.no.0.df)
Area under ROC curve =  0.8526
```



Thus the predictor is reasonable. [3 mark for table, 3 for cross-val, 3 for ROC curve, 1 for comment]