A7 Suppose that m_iY_i is binomial (m_i, π_i), where $g(\pi_i) = X_i\boldsymbol{\beta}$ and $i = 1, \dots, n$.

(a) Show that $\hat{\boldsymbol{\beta}}$ is the same when the data are entered in a grouped form and when they are entered in an ungrouped form.

According to the score function of grouped form :

$$\frac{\partial l}{\partial \beta_j} = \sum_{i=1}^{n} \frac{y_i - b'(\theta_i)}{a(\phi)\, b''(\theta_i)} \frac{1}{g'(\pi_i)} x_{ij}$$

$$= \sum_{i=1}^{n} \frac{y_i - \pi_i}{\pi_i(1-\pi_i)/m_i} \cdot \frac{1}{g'(\pi_i)} x_{ij} = 0$$

$$\sum_{i=1}^{n} \frac{m_i(y_i - \pi_i)}{\pi_i(1-\pi_i)} \cdot \frac{x_{ij}}{g'(\pi_i)} = 0$$

$$\sum_{i=1}^{n} \left(\frac{\sum_{k=1}^{m_i} Y_{ik}^*}{\pi_i(1-\pi_i)} - \frac{m_i\pi_i}{\pi_i(1-\pi_i)} \right) \frac{x_{ij}}{g'(\pi_i)} = 0$$

$$\circledast \quad \underbrace{\sum_{i=1}^{n} \sum_{k=1}^{m_i} \frac{y_{ik}^* - \pi_i}{\pi_i(1-\pi_i)} \frac{x_{ij}}{g'(\pi_i)} = 0}$$

Consider the case of ungrounded y_{ik}^*, then $y_{ik}^* \sim$ Bernoulli (π_i)

$$\Rightarrow \frac{\partial l}{\partial \beta_{jk}} = \sum_{i=1}^{n} \frac{y_{ik}^* - b'(\theta_{ik}^*)}{\pi_i(1-\pi_i)} \cdot \frac{1}{g'(\pi_i)} x_{ijk} = 0$$

where $x_{ijk} = x_{ij}$

and $\theta_{ik} = \log \frac{\pi_i}{1-\pi_i} = \theta_i$

$b'(\theta_{ik}) = \log(1+e^{\theta_{ik}}) = b'(\theta_i)$

$$\Rightarrow \underbrace{\frac{\partial l}{\partial \beta_{jk}} = \sum_{i=1}^{n} \frac{y_{ik}^* - b'(\theta_i)}{\pi_i(1-\pi_i)} \cdot \frac{1}{g'(\pi_i)} x_{ij} = 0} \quad (\ast\ast)$$

$$\Rightarrow \sum_{k=1}^{m_i} \sum_{i=1}^{n} \frac{y_{ik}^* - b'(\theta_i)}{\pi_i(1-\pi_i)} \cdot \frac{1}{g'(\pi_i)} x_{ij} = 0$$

$$\Rightarrow \sum_{i=1}^{n} \sum_{k=1}^{m_i} \frac{y_{ik}^* - b'(\theta_i)}{\pi_i(1-\pi_i)} \frac{1}{g'(\pi_i)} x_{ij} = 0 \quad \text{for } \forall j \in 1, \cdots p$$

$= \circledast$ where $\pi_i = x_i\hat{\beta}$

$(\ast\ast) \rightarrow \circledast$ and $\circledast \rightarrow (\ast\ast)$ since Binomial (m_i, π_i) is m_i trials of Bernoulli(π_i)

$$\sum_{i=1}^{n} \sum_{k=1}^{m_i} \frac{y_{ik}^* - x_i\hat{\beta}}{x_i\hat{\beta}(1-x_i\hat{\beta})} \frac{1}{g'(\theta_i)} x_{ij} = 0$$ for both group and ungrouped cases.

$\Rightarrow$ both cases have the same $\hat{\beta}$

(b) Consider the null model, for which $\pi_1 = \ldots = \pi_N$. Show that

$$\hat{\pi} = \frac{\sum_{i=1}^{n} m_i y_i}{\sum_{i=1}^{n} m_i}.$$

When $m_i = 1$ for all $i \in \{1, \ldots, n\}$, show that in this case, the Pearson X^2 statistic, which is defined as the sum of the squared Pearson residuals, equals n. Decide whether or not X^2 is useful for testing whether a Binomial GLM model fits the data well when the response is binary.

Consider the score equation for null model where

$$\sum_{i=1}^{n} \frac{y_i - b'(\theta_i)}{Var(y_i)} \frac{1}{g'(\mu_i)} x_{ij} = 0$$

$$\sum_{i=1}^{n} \frac{y_i - \pi_i}{\pi_i(1-\pi_i)/m_i} \frac{1}{g'(\mu_i)} x_{ij} = 0$$

$$\sum_{i=1}^{n} \frac{m_i(y_i - \pi)}{\pi(1-\pi)} \frac{1}{g'(\pi)} x_{ij} = 0$$

$$\sum_{i=1}^{n} m_i (y_i - \pi) x_{ij} = 0$$

$$\sum_{i=1}^{n} m_i y_i x_{ij} = \pi \sum_{i=1}^{n} m_i x_{ij}$$

Since null model contains only the intercept, so $x_{ij} = 0$ or 1

$$\Rightarrow \quad \sum_{i=1}^{n} m_i y_i = \pi \sum_{i=1}^{n} m_i$$

$$\Rightarrow \quad \hat{\pi} = \frac{\sum_{i=1}^{n} m_i y_i}{\sum_{i=1}^{n} m_i}$$

When $m_i = 1 \Rightarrow \hat{\pi} = \frac{1}{n} \sum_{i=1}^{n} y_i = \bar{y}$

$$r_{p_i} = \frac{y_i - \hat{\mu}_i}{\sqrt{V(\hat{\mu}_i)}} \sqrt{W_i}$$

$$= \frac{y_i - \hat{\pi}}{\sqrt{b''(\theta_i)}} \cdot \sqrt{1}$$

$$= \frac{y_i - \hat{\pi}}{\sqrt{\hat{\pi}(1-\hat{\pi})}}$$

then

$$\sum_{i=1}^{n} r_{p_i}^2 = \sum_{i=1}^{n} \frac{(y_i - \hat{\pi})^2}{\hat{\pi}(1-\hat{\pi})} = \frac{1}{\hat{\pi}(1-\hat{\pi})} \sum_{i=1}^{n} y_i^2 + \hat{\pi}^2 - 2\hat{\pi} y_i$$

$$= \frac{1}{\bar{y}(1-\bar{y})} \sum_{i=1}^{n} y_i^2 + \bar{y}^2 - 2\bar{y} y_i$$

$$\circledast \quad = \frac{1}{\bar{y}(1-\bar{y})} (n \overline{y^2} + n\bar{y}^2 - 2n\bar{y}^2)$$

Since $\overline{y^2} = \sum_{i=1}^{n} y_i^2 = \sum_{i=1}^{n} y_i$ then

$$\circledast = \frac{1}{\bar{y}(1-\bar{y})}(n\bar{y} - n\bar{y}^2)$$

$$= n$$

It's not a good fit to test whether the binomial GLM can fit the data well, Since the Pearson X^2 statistic depends on the sample size n.

A8 & A9

A8 R excercise. Suppose that the logistic model holds in which x is uniformly distributed between 0 and 100, and logit(π_i)=-2.0+0.04x_i

(a) Set the seed to 2, viz. set.seed(2) in R (find out what set.seed does if you don't know what this means) and randomly generate n = 100 independent observations from this model. This can be done as follows:

(i) Generate n independent variables from the (continuous) uniform distribution on (0, 100). Store these in a vector x.

```
# random number generator
set.seed(2)
x<-runif(100,0,100)
```

(ii)

```
p<-exp(-2+0.04*x)+1
p=1-1/p
p
```

```
##   [1] 0.2208928 0.6920021 0.5728051 0.2095248 0.8551226 0.8549420 0.1849198
##   [8] 0.7914677 0.4680621 0.5498179 0.5524801 0.2602979 0.7392460 0.2181091
##  [15] 0.4063992 0.8044266 0.8705233 0.2503591 0.4450323 0.1544545 0.6564683
##  [22] 0.3894079 0.7937299 0.1981346 0.3518514 0.4887751 0.1973385 0.3608332
##  [29] 0.8641947 0.1868648 0.1236466 0.2072748 0.7756978 0.8138835 0.5142779
##  [36] 0.6245216 0.7986239 0.2972312 0.6612574 0.1981145 0.8729072 0.3074736
##  [43] 0.1765842 0.2063291 0.8552230 0.7648499 0.8697502 0.3535099 0.5019699
##  [50] 0.7758405 0.1222210 0.1255134 0.6755978 0.8479846 0.2893804 0.7768562
##  [57] 0.7583245 0.8760569 0.6120201 0.6986217 0.7465151 0.8246155 0.6225738
##  [64] 0.2771185 0.8078799 0.4378117 0.3899741 0.4615770 0.2450297 0.1497889
##  [71] 0.2896271 0.3189773 0.1380837 0.2207491 0.2198558 0.7353327 0.2999027
##  [78] 0.8132668 0.4038548 0.5721773 0.3549320 0.6655209 0.1301313 0.4023709
##  [85] 0.2314585 0.8062929 0.8683058 0.3306802 0.7176361 0.3453082 0.8706841
##  [92] 0.3984484 0.3822511 0.5600957 0.4638713 0.2291896 0.4274589 0.1641183
##  [99] 0.1767151 0.4403176
```

(iii) For each i, draw a Bernoulli observation with probability π_i. Fit the logistic model with the intercept and x as predictors and report the parameter estimates.

```
library(Rlab)
```

```
## Rlab 2.15.1 attached.
```

```
##
## Attaching package: 'Rlab'
```

```
## The following objects are masked from 'package:stats':
##
##     dexp, dgamma, dweibull, pexp, pgamma, pweibull, qexp, qgamma,
##     qweibull, rexp, rgamma, rweibull
## The following object is masked from 'package:datasets':
##
##     precip
```

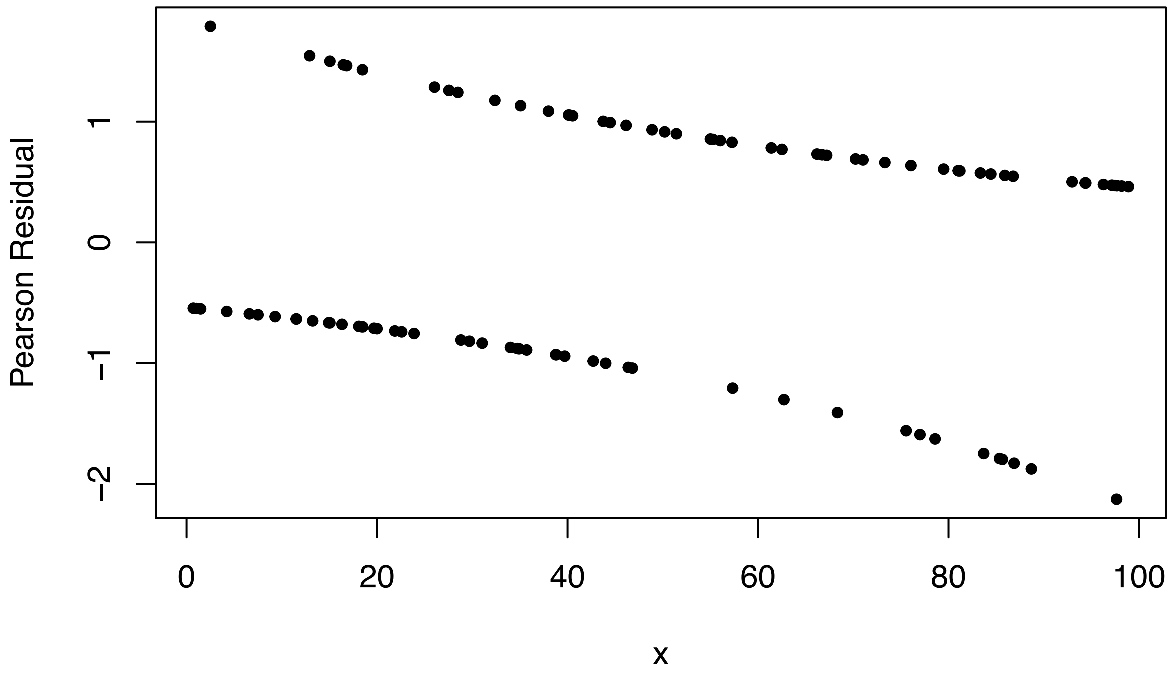
```
y<-0
for(i in 1:100){
  y[i]<-rbern(1,p[i])
}
f0<-glm(y~x,family=binomial(link=logit))
summary(f0)
```

```
##
## Call:
## glm(formula = y ~ x, family = binomial(link = logit))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8491  -1.0057   0.6324   1.0262   1.6945
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.234223   0.425755  -2.899 0.003745 **
## x            0.028105   0.007794   3.606 0.000311 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 138.27  on 99  degrees of freedom
## Residual deviance: 123.27  on 98  degrees of freedom
## AIC: 127.27
##
## Number of Fisher Scoring iterations: 4
```

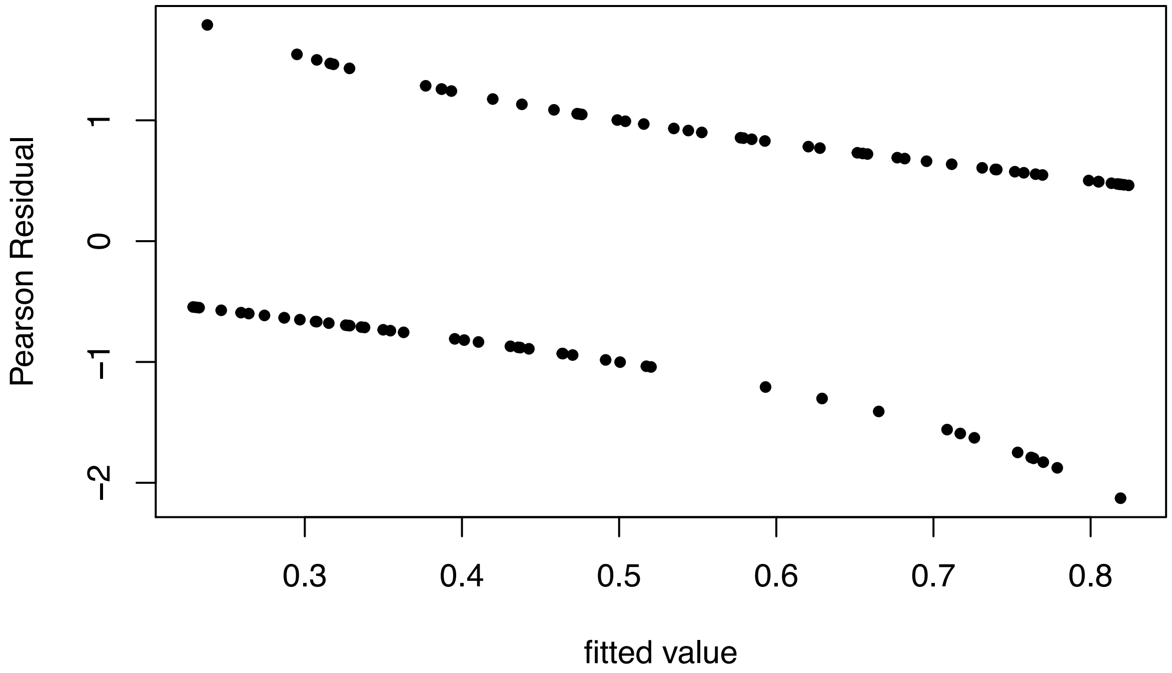
It shows that the estimated parameters is 0.028 and the intercept is -1.23

(b) For the sample generated in part (a), plot the Pearson residuals agains x and against the fitted values. Why do the residuals have this appearance? The points in the residual plots seem to lie on lines – find out what these lines are and display them on each residual plot.

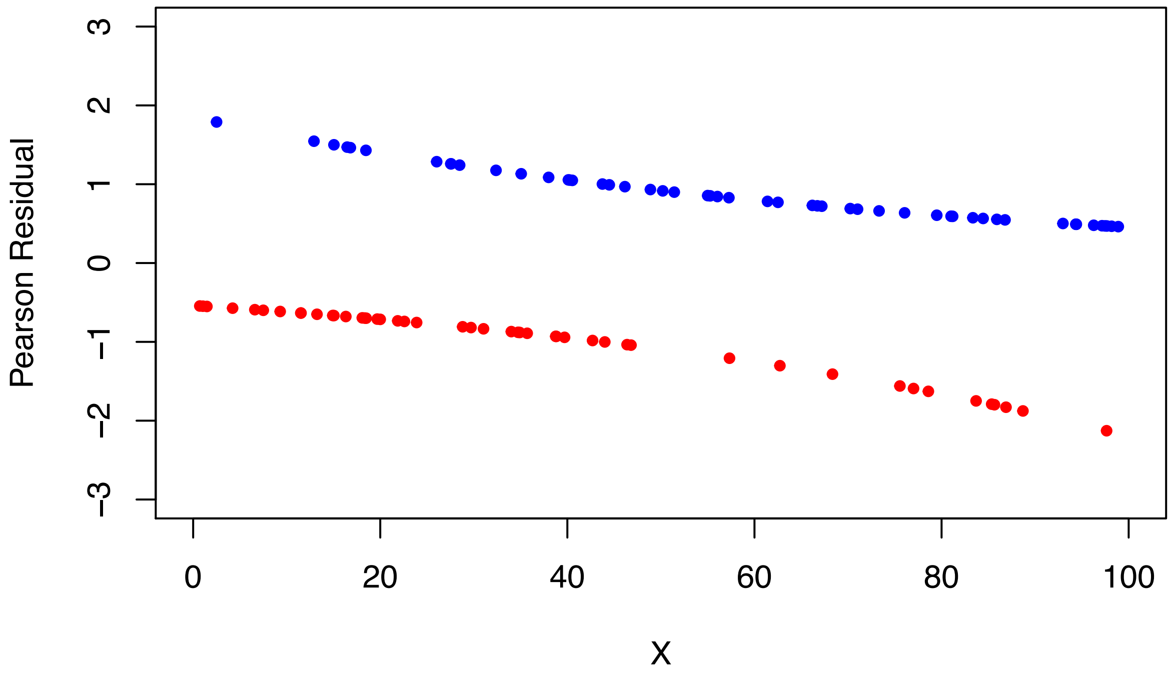
```
rp0<-residuals(f0,"pearson")
matplot(x,rp0,xlab="x",ylab="Pearson Residual",pch=20)
```



```
fitted<-f0$fitted.values
matplot(fitted,rp0,xlab="fitted value",ylab="Pearson Residual",pch=20)
```



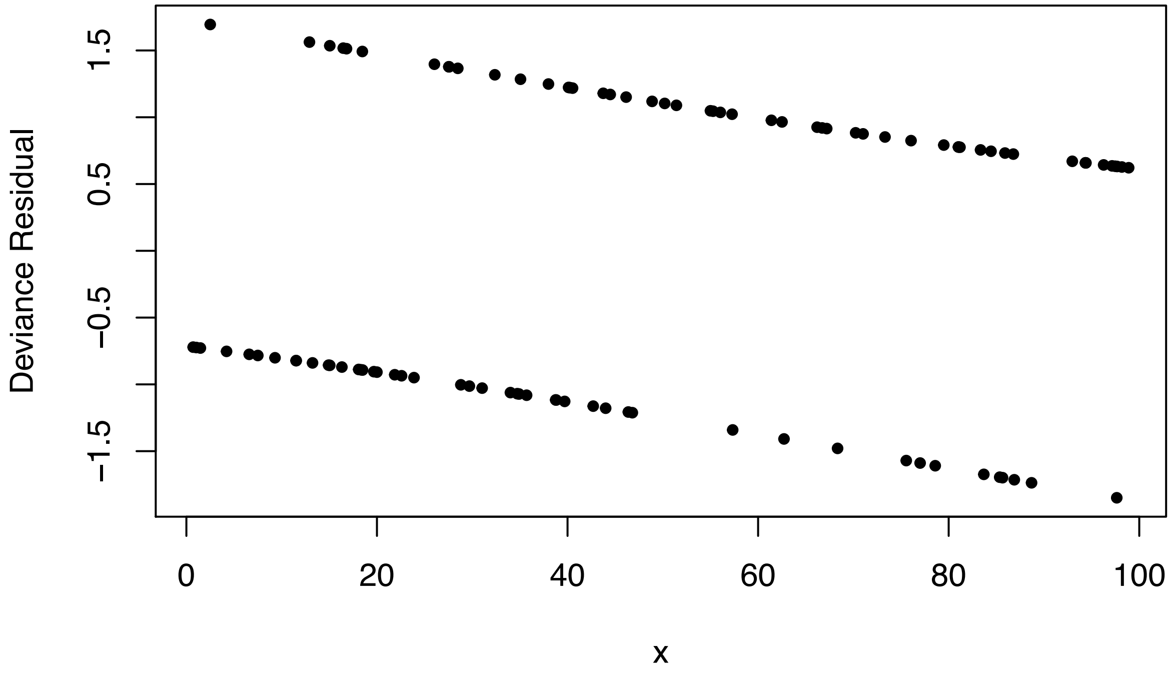
```
#investigate them
I1<-y==0
matplot(x[I1],rp0[I1],xlab="X",ylab="Pearson Residual",pch=20,col="red",xlim=c(0,100),ylim=c(-3,3))

I2<-y==1
points(x[I2],rp0[I2],xlab="X",ylab="Pearson Residual",pch=20,col="blue")
```



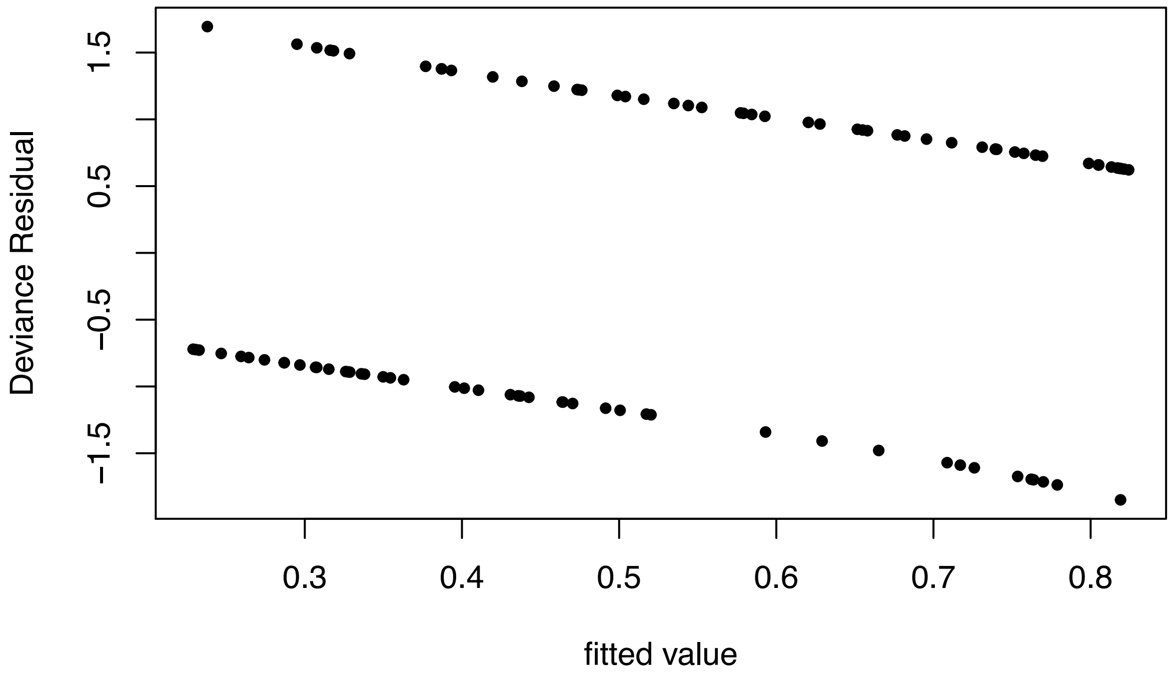
The above line showing that the data is really discrete and the above blue line representing the the case of y=1 and the below line show the case when y=0.

(c) Redo part (b) for deviance residuals.

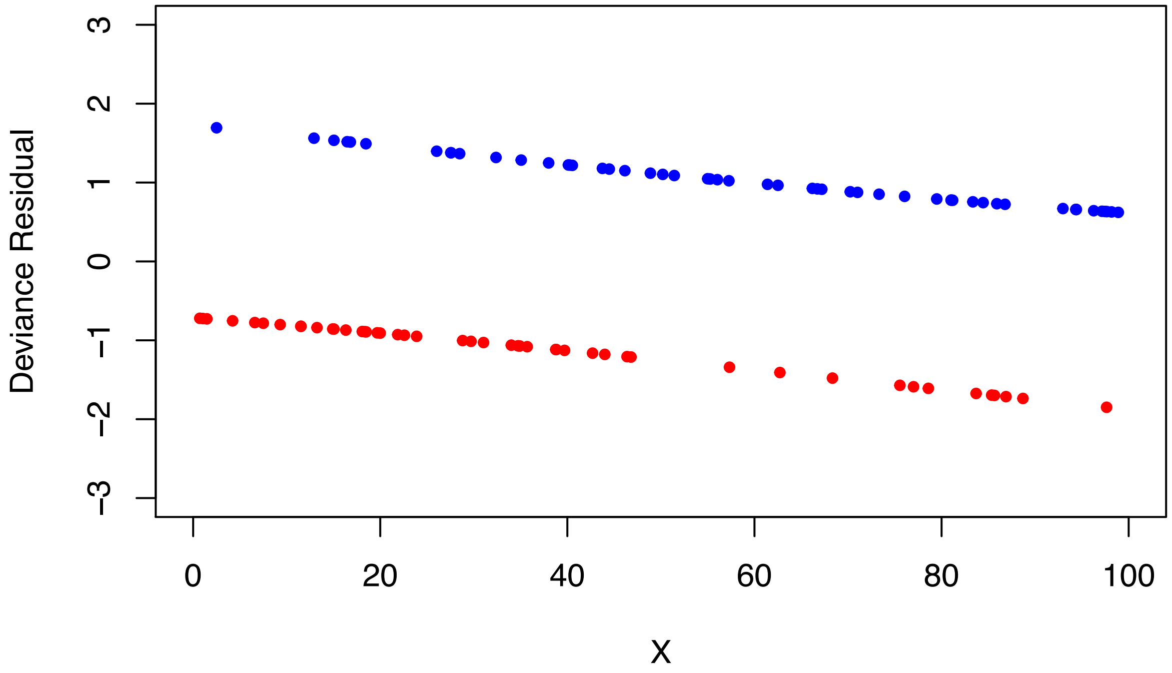
```
rp1<-residuals(f0,"deviance")
matplot(x,rp1,xlab="x",ylab="Deviance Residual",pch=20)
```



```
fitted<-f0$fitted.values
matplot(fitted,rp1,xlab="fitted value",ylab="Deviance Residual",pch=20)
```



```
#investigate them
I1<-y==0
matplot(x[I1],rp1[I1],xlab="X",ylab="Deviance Residual",pch=20,col="red",xlim=c(0,100),ylim=c(-3,3))
I2<-y==1
points(x[I2],rp1[I2],xlab="X",ylab="Deviance Residual",pch=20,col="blue")
```



The above line result of Deviance Residuals showing that the data is really discrete and the above line representing the the case of response when y=0 and y=1.

Question 9: R Excercise

Consider the following data on home-well contamination in 3020 households in Ara- hazar upazila, Bangladesh. The response variable is switch (binary variable whether or not the household switched to another well from an unsafe well). Other variables collected for each household were arsenic (the level of arsenic contamination in the household's original well, in hundreds of micrograms per liter), dist100 (distance in 100-meter units to the closest known safe well), educ (years of education of the head of the household) and assoc (whether or not any members of the household participated in any community organizations: no or yes). The data is available in MyCourses under Datasets. Load the data and compute dist100 as follows.

```r
wells <- read.table("wells.dat")
attach(wells)
dist100 <- dist/100
```

(a) Fit a logistic regression model with the intercept and dist100. Interpret the model. Test the adequacy of this model using the Pearson X^2 and the likelihood ratio G^2 statistics. Conclude at the 5% level.

```r
y<-as.factor(wells$switch)
logitmod<-glm(y~dist100,family=binomial(link="logit"))
summary(logitmod)
```

```
##
## Call:
## glm(formula = y ~ dist100, family = binomial(link = "logit"))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4406  -1.3058   0.9669   1.0308   1.6603
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.60596    0.06031  10.047  < 2e-16 ***
## dist100     -0.62188    0.09743  -6.383 1.74e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4118.1  on 3019  degrees of freedom
## Residual deviance: 4076.2  on 3018  degrees of freedom
## AIC: 4080.2
##
## Number of Fisher Scoring iterations: 4
```

Since g(x)=$\frac{1}{1-x}$, so we can get that $log\frac{\pi}{1-\pi} = 0.61 - 0.62 * dist100$. The paramter of dist100 is -0.62 which implies that as dist100 increase, the estimiated response will also increase, it is much more possible to switch the well. For the Pearson Residual:

```r
ncat <- 10

bins <- cut(dist100,quantile(dist100,prob=c(0:ncat)/ncat),include.lowest=T)
```

```
lswi <- split(switch,bins)
counts <- lapply(lswi,FUN=function(x){as.numeric(x > 0)})
beta <- coefficients(logitmod)

observed <- lapply(counts,FUN=function(x){c(sum(x),length(x)-sum(x))})
observed <- matrix(as.numeric(unlist(observed)),ncol=2,byrow=TRUE)

#fitted number of success and failure
fitted <- lapply(split(dist100,bins),FUN=function(x){pi <- exp(beta[1]+x*beta[2])/(1+exp(beta[1]+x*beta
fitted <- matrix(as.numeric(unlist(fitted)),ncol=2,byrow=TRUE)
cbind(observed,fitted)
```

```
##       [,1] [,2]     [,3]     [,4]
##  [1,]  170  132 192.1434 109.8566
##  [2,]  192  110 188.8229 113.1771
##  [3,]  183  119 186.1654 115.8346
##  [4,]  198  104 183.5034 118.4966
##  [5,]  175  127 180.6797 121.3203
##  [6,]  188  114 177.0390 124.9610
##  [7,]  171  131 172.4951 129.5049
##  [8,]  183  119 166.3623 135.6377
##  [9,]  150  152 156.6636 145.3364
## [10,]  127  175 133.1252 168.8748
```

```
X.2  <- sum(((observed-fitted)^2)/fitted)

observed[10,2] <- 0.5  # just to avoid a log of zero in the likelihood ratio statistics

G.2 <- 2*sum(observed*log(observed/fitted))

pchisq(X.2, df=8,lower.tail=FALSE)
```

```
## [1] 0.02879032
```

For the G^2:

```
pchisq(G.2, df=8,lower.tail=FALSE)
```

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

with a confidence interval

```
#library(arm)
# se.coef extract the standard error
#se <- se.coef(logitmod)
#c(exp(beta[2]-1.96*se[2]),exp(beta[2]+1.96*se[2]))
```

According to the p value of dist100 and intercept, they are all less than 5%. We can conclude that the parameter has siginificant level of 5%, we reject the null hypothesis.

(b) Find the most appropriate logistic regression model for the data. Use the deviance, but also consider practical significance by looking at the AIC and the size of the effect of the predictors. Interpret the final model.

Firstly, try all the predictor:

```r
lm<-glm(y~(dist100+educ+assoc+arsenic),family=binomial(link="logit"))
summary(lm)
```

```
##
## Call:
## glm(formula = y ~ (dist100 + educ + assoc + arsenic), family = binomial(link = "logit"))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5942  -1.1976   0.7541   1.0632   1.6739
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.156712   0.099601  -1.573    0.116
## dist100     -0.896110   0.104576  -8.569  < 2e-16 ***
## educ         0.042447   0.009588   4.427 9.55e-06 ***
## assoc       -0.124300   0.076966  -1.615    0.106
## arsenic      0.467022   0.041602  11.226  < 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: 4118.1  on 3019  degrees of freedom
## Residual deviance: 3907.8  on 3015  degrees of freedom
## AIC: 3917.8
##
## Number of Fisher Scoring iterations: 4
```

```r
lm0<-glm(y~(dist100+educ+arsenic),family=binomial(link="logit"))
anova(lm,lm0,test='Chisq')
```

```
## Analysis of Deviance Table
##
## Model 1: y ~ (dist100 + educ + assoc + arsenic)
## Model 2: y ~ (dist100 + educ + arsenic)
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      3015     3907.8
## 2      3016     3910.4 -1  -2.6072   0.1064
```

```r
lm1<-glm(y~(dist100+educ+arsenic)^2,family=binomial(link="logit"))
summary(lm1)
```

```
##
## Call:
## glm(formula = y ~ (dist100 + educ + arsenic)^2, family = binomial(link = "logit"))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5706  -1.1964   0.7314   1.0724   1.8712
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -0.01228    0.15507  -0.079  0.93686
## dist100         -1.09741    0.25998  -4.221 2.43e-05 ***
## educ            -0.02266    0.02001  -1.133  0.25737    
## arsenic          0.46466    0.08636   5.380 7.44e-08 ***
## dist100:educ     0.08067    0.02666   3.026  0.00247 ** 
## dist100:arsenic -0.11768    0.10353  -1.137  0.25569    
## educ:arsenic     0.01806    0.01097   1.647  0.09965 .  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 4118.1  on 3019  degrees of freedom
## Residual deviance: 3891.7  on 3013  degrees of freedom
## AIC: 3905.7
## 
## Number of Fisher Scoring iterations: 4
```

```
step(lm1)
```

```
## Start:  AIC=3905.74
## y ~ (dist100 + educ + arsenic)^2
## 
##                   Df Deviance    AIC
## - dist100:arsenic  1   3893.0 3905.0
## <none>                 3891.7 3905.7
## - educ:arsenic     1   3894.5 3906.5
## - dist100:educ     1   3901.1 3913.1
## 
## Step:  AIC=3905.03
## y ~ dist100 + educ + arsenic + dist100:educ + educ:arsenic
## 
##                Df Deviance    AIC
## <none>              3893.0 3905.0
## - educ:arsenic  1   3896.2 3906.2
## - dist100:educ  1   3902.9 3912.9
```

```
## 
## Call:  glm(formula = y ~ dist100 + educ + arsenic + dist100:educ + educ:arsenic, 
##     family = binomial(link = "logit"))
## 
## Coefficients:
##  (Intercept)       dist100          educ       arsenic  dist100:educ  
##      0.09656      -1.31799      -0.02481       0.39748       0.08278  
## educ:arsenic  
##      0.01912  
## 
## Degrees of Freedom: 3019 Total (i.e. Null);  3014 Residual
## Null Deviance:       4118 
## Residual Deviance: 3893  AIC: 3905
```

```
lm2<-glm(y~dist100+educ+arsenic+ dist100:educ +
    educ:arsenic,family=binomial(link="logit"))
summary(lm2)
```

```
## 
## Call:
## glm(formula = y ~ dist100 + educ + arsenic + dist100:educ + educ:arsenic, 
##     family = binomial(link = "logit"))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4576  -1.2035   0.7324   1.0669   1.9018
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.09656    0.12196   0.792  0.42852
## dist100      -1.31799    0.17497  -7.533 4.97e-14 ***
## educ         -0.02481    0.01997  -1.243  0.21402
## arsenic       0.39748    0.06210   6.400 1.55e-10 ***
## dist100:educ  0.08278    0.02662   3.109  0.00187 **
## educ:arsenic  0.01912    0.01088   1.757  0.07893 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4118.1  on 3019  degrees of freedom
## Residual deviance: 3893.0  on 3014  degrees of freedom
## AIC: 3905
##
## Number of Fisher Scoring iterations: 4
```

```
#anova(lm2,lm1,test='Chisq')
```

```
#sicne educ:arsenic has a high p value
lm3<- update(lm2,.~.-educ:arsenic,family=binomial(link="logit"))
anova(lm3,lm2,test='Chisq')
```

```
## Analysis of Deviance Table
##
## Model 1: y ~ dist100 + educ + arsenic + dist100:educ
## Model 2: y ~ dist100 + educ + arsenic + dist100:educ + educ:arsenic
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      3015     3896.2
## 2      3014     3893.0  1    3.117  0.07748 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
lm5<-update(lm3,.~.-eudc:arsenic,family=binomial(link="logit"))
summary(lm5)
```

```
##
## Call:
## glm(formula = y ~ dist100 + educ + arsenic + dist100:educ, family = binomial(link = "logit"))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6603  -1.2085   0.7535   1.0613   1.9448
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.0004956  0.1096145   0.005 0.996392
## dist100      -1.3898523  0.1718840  -8.086 6.17e-16 ***
## educ         -0.0020771  0.0152548  -0.136 0.891693
## arsenic       0.4805993  0.0419866  11.446  < 2e-16 ***
## dist100:educ  0.0956362  0.0256798   3.724 0.000196 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4118.1  on 3019  degrees of freedom
## Residual deviance: 3896.2  on 3015  degrees of freedom
## AIC: 3906.2
##
## Number of Fisher Scoring iterations: 4
```

```r
step(lm5)
```

```
## Start:  AIC=3906.15
## y ~ dist100 + educ + arsenic + dist100:educ
##
##                Df Deviance    AIC
## <none>               3896.2 3906.2
## - dist100:educ  1   3910.4 3918.4
## - arsenic       1   4051.1 4059.1

##
## Call:  glm(formula = y ~ dist100 + educ + arsenic + dist100:educ, family = binomial(link = "logit"))
##
## Coefficients:
##  (Intercept)      dist100          educ       arsenic  dist100:educ
##    0.0004956   -1.3898523    -0.0020771     0.4805993     0.0956362
##
## Degrees of Freedom: 3019 Total (i.e. Null);  3015 Residual
## Null Deviance:       4118
## Residual Deviance: 3896  AIC: 3906
```

When calling the step funtion on lm5, we can see that the the AIC value cannot decrese more.In the model lm5. we can see that almost all the paramter has a p value less than 5%.As arsenic increase by one unit, the switch will increased by 0.48. Considering dist100 and dist100:educ, they dist100:educ has a positive effect on the result and dist100 has a negative effect on the result. For fixed dist100, by increase eudc by one unit, will increase the response by 0.09*dist100-0.002 while for fixed educ, by increase one unit of dist100 will affect the result by -1.38+0.09educ. Therefore,for small educ value, it will cause the switch to decrease as dist100 increase and large educ value will cuase switch increasing as dist100 increase.

(c) Try to simplify the model you found in part (b) by replacing educ by a binary factor predictor feduc, constructed as follows:This predictor feduc records whether the person has a primary education (i.e. 1–8 years) or secondary education and above (i.e. more than 9 years). Interpret the final model.

```r
feduc <- numeric(3020)
for(i in 1:3020){
  if(educ[i] < 9){feduc[i] <- 0}
  if(educ[i] > 8){feduc[i] <- 1}
}
```

Now the model becomes:

```r
lm6<-glm(y~dist100+arsenic++feduc+dist100:feduc,family=binomial(link="logit"))
summary(lm6)
```

```
##
## Call:
## glm(formula = y ~ dist100 + arsenic + +feduc + dist100:feduc,
##     family = binomial(link = "logit"))
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.663  -1.190   0.740   1.044   1.882
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -0.01489    0.08606  -0.173    0.863
## dist100       -1.15433    0.12112  -9.530  < 2e-16 ***
## arsenic        0.47840    0.04205  11.378  < 2e-16 ***
## feduc          0.04607    0.15625   0.295    0.768
## dist100:feduc  1.20655    0.26380   4.574 4.79e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4118.1  on 3019  degrees of freedom
## Residual deviance: 3869.0  on 3015  degrees of freedom
## AIC: 3879
##
## Number of Fisher Scoring iterations: 4
```

Compare to the previous model, we can see that there's a decrease in the AIC value and the p value of dist100:feduc has decreased.The parameter of dist100:feduc becomes larger, it means that dist100:feduc now has more effect on the result value: switch. If feduc=0 and increase dist100 by 1 unit, the response switch decrease by -1.15 and when feduc=1 and increase dist100 by 1 unit, the response will increase by 1.2-1.15=0.05 unit.As arsenic increase by 1 unit, the output will incresed by 0.47.

(d) Compare the final model in parts (b) and (c) using AIC and ROC curves. Which one do you prefer and why?

Firstly, model in part(c) has a smaller AIC value and the and the p value of dist100:feduc has decreased and The parameter of dist100:feduc becomes larger. It means that dist100:feduc become much important in the prediction model.

```r
library(ROCR)
```

```
## Loading required package: gplots
```

```
##
## Attaching package: 'gplots'
```

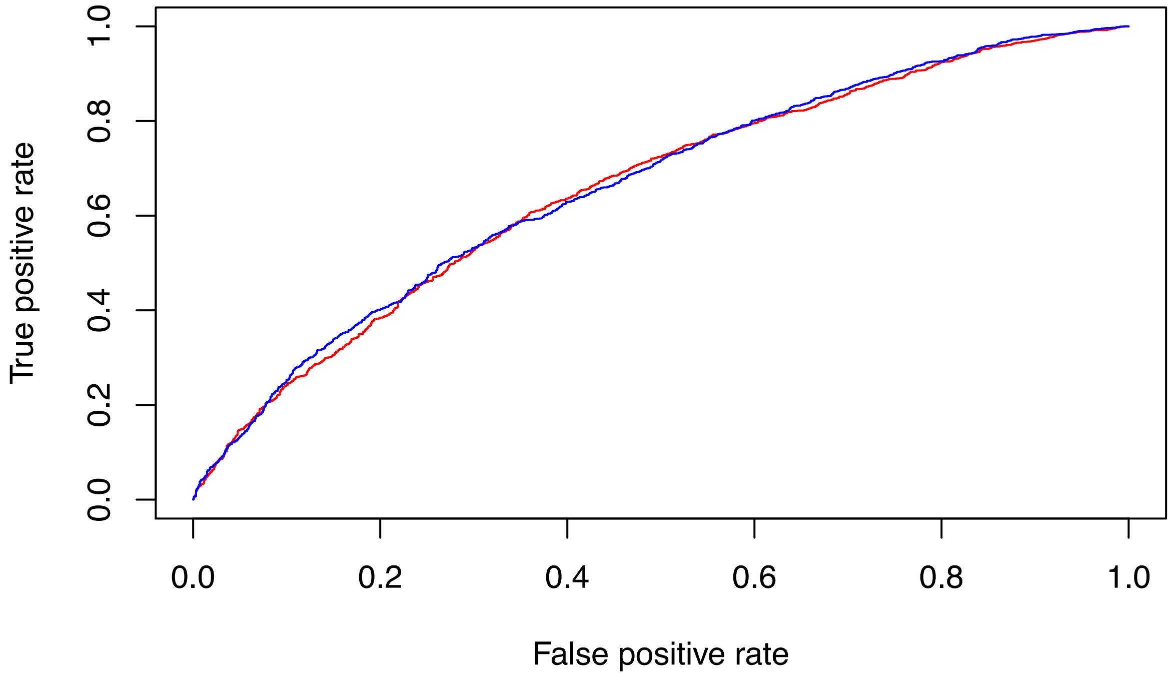
```
## The following object is masked from 'package:stats':
##
##     lowess
```

```r
pred5<-prediction(fitted(lm5),switch)
pref5<-performance(pred5,"tpr","fpr")
```

```
pred6<-prediction(fitted(lm6),switch)
pref6<-performance(pred6,"tpr","fpr")


plot(pref5,col="red")
performance(pred5,"auc")
```

```
## An object of class "performance"
## Slot "x.name":
## [1] "None"
##
## Slot "y.name":
## [1] "Area under the ROC curve"
##
## Slot "alpha.name":
## [1] "none"
##
## Slot "x.values":
## list()
##
## Slot "y.values":
## [[1]]
## [1] 0.6585808
##
##
## Slot "alpha.values":
## list()
```

```
plot(pref6,add=TRUE,col="blue")
```



```
performance(pred6,"auc")
```

```
## An object of class "performance"
## Slot "x.name":
## [1] "None"
##
## Slot "y.name":
## [1] "Area under the ROC curve"
##
## Slot "alpha.name":
## [1] "none"
##
## Slot "x.values":
## list()
##
## Slot "y.values":
## [[1]]
## [1] 0.6626791
##
##
## Slot "alpha.values":
## list()
```

From the output of performance we can see that 0.663> 0.658, so the model in part(c) is better.