A3

Q8

a

```
library(purrr)
set.seed(2)
n=100
x = runif(n,0,100)
p = (exp(-2+0.04*x)/(1+exp(-2+0.04*x)))
y = rbernoulli(n,p)

g1 = glm(y~x, family = "binomial")
summary(g1)
```

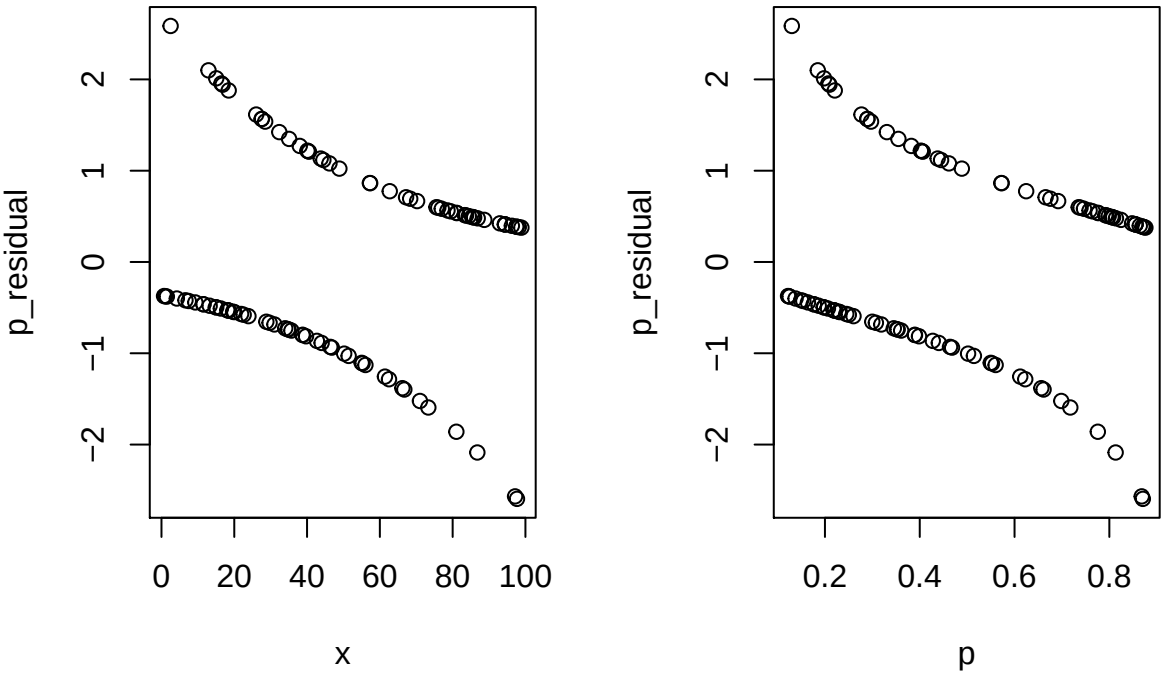
```
##
## Call:
## glm(formula = y ~ x, family = "binomial")
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.86727  -0.98711  -0.00853   0.91122   1.83225
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.551564   0.445537  -3.482 0.000497 ***
## x            0.031765   0.008024   3.959 7.53e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 138.63  on 99  degrees of freedom
## Residual deviance: 119.96  on 98  degrees of freedom
## AIC: 123.96
##
## Number of Fisher Scoring iterations: 4
```

We fit the synthetic data with the binomial GLM with the canonical link. The response is a binary variable 0 or 1. The predictor is the intercept term and a continuous variable x that ranges from 0 to 100. The coefficient for the intercept is -1.551564, and for x is 0.031765. And both have very low p-value, which means it necessary to contain these two terms in our model. The beta1 for x is 0.031765 which means based on our model, the expected increase in the odds of giving response 1 is increased by the factor exp{0.031765}=1.032275. X has positive effect on y.

b

```
p_residual = (y - p)/sqrt(p*(1-p))
```

```
par(mfrow=c(1,2))
plot(p_residual~x)
plot(p_residual~p)
```



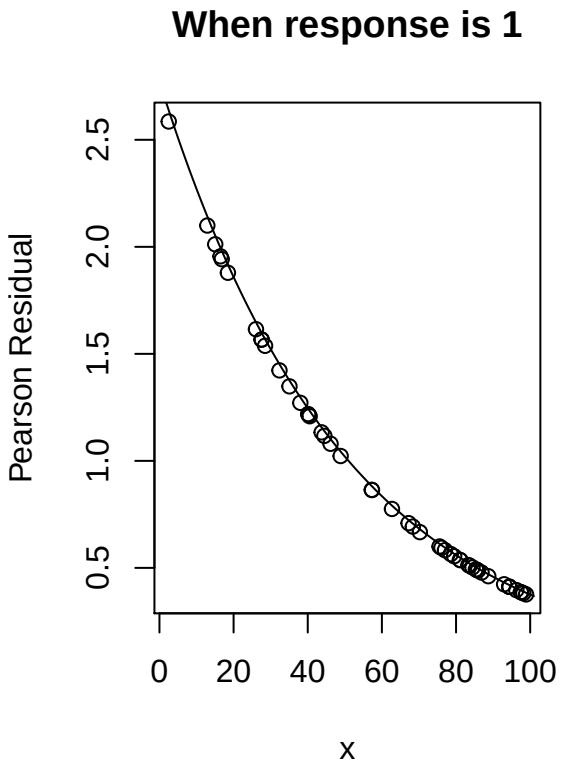
Pearson's Residual is not very imformative. The reason why that residuals lie on two lines is that $r_{p_i} = \frac{y_i - \hat{\pi}_i}{\sqrt{\hat{\pi}_i(1-\hat{\pi}_i)}}\sqrt{m_i}$

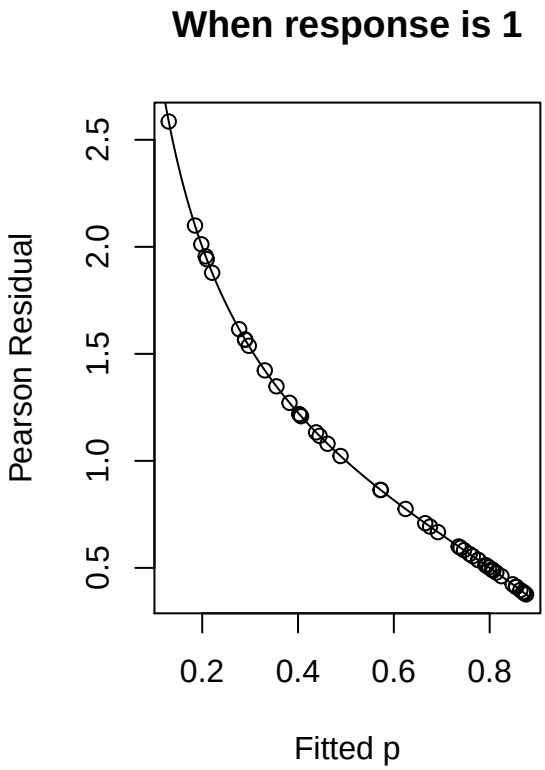
Here in our case, the data is in ungroupped form, so $y_i \in \{0,1\}$ $m_i = 1$, for our fixed estimate $\hat{\pi}_i$ the Pearson's Residual would either takes values for case $y_i = 1$ or $y_i = 0$, so it would lies on the top line or the bottom line respectively.

We can plot these dots seperately, and also we can generate data and add line that the dots lie on.(The line is obtained by create 0~100 x, and calculate the probability p, then force y to be all 1 or 0 and calculate the Pearson residual respectively. Then add line to our figure.)

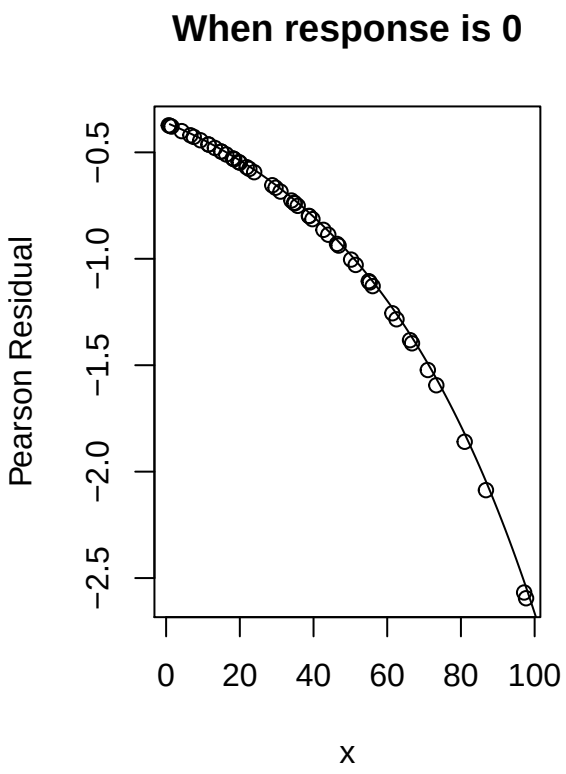
```
index1 = (y==1)
index0 = (y==0)

p_res1 = p_residual[index1]
p1 = p[index1]
x1 = x[index1]

p_res0 = p_residual[index0]
p0 = p[index0]
x0 = x[index0]

temp_x = seq(0,100)
temp_p = (exp(-2+0.04*temp_x)/(1+exp(-2+0.04*temp_x)))
temp_y1 = rep(1,101)
temp_y0 = rep(0,101)
r1 = (temp_y1-temp_p)/sqrt(temp_p*(1-temp_p))
r0= (temp_y0-temp_p)/sqrt(temp_p*(1-temp_p))

par(mfrow=c(1,2))
```

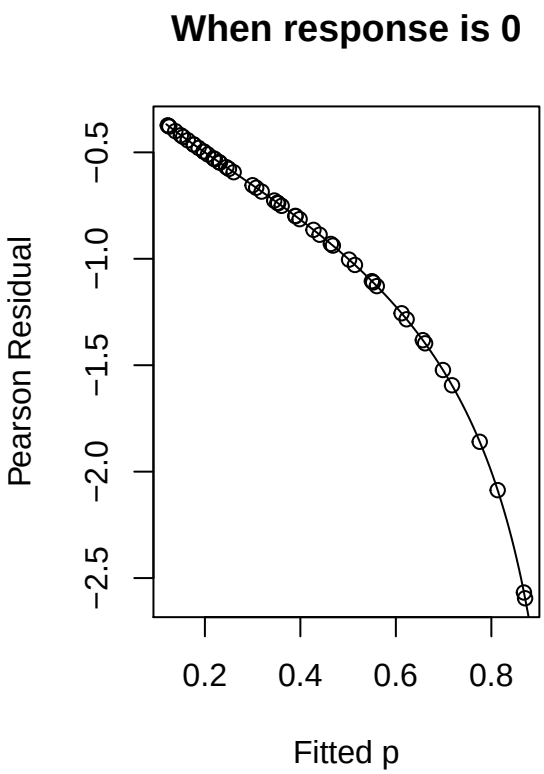
```
plot(p_res1~x1,xlab='x',ylab='Pearson Residual',main='When response is 1')
lines(r1)
plot(p_res1~p1,xlab='Fitted p',ylab='Pearson Residual',main='When response is 1')
lines(temp_p,r1)
```





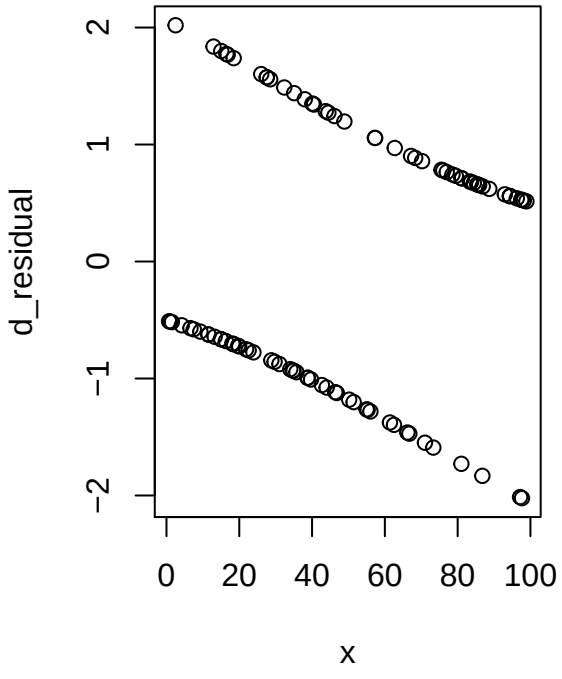

```
par(mfrow=c(1,2))
plot(p_res0~x0,xlab='x',ylab='Pearson Residual',main='When response is 0')
lines(r0)
plot(p_res0~p0,xlab='Fitted p',ylab='Pearson Residual',main='When response is 0')
lines(temp_p,r0)
```

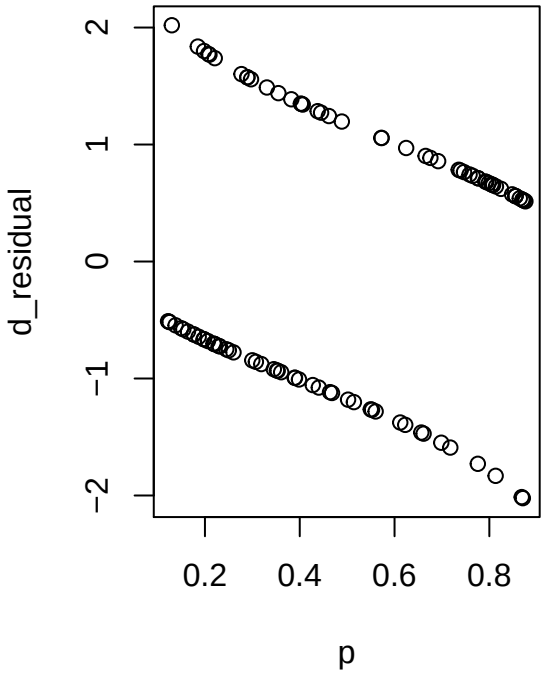




c

```
d_residual = sign(y - p)*sqrt(2*(-y*log(p)-(1-y)*log(1-p)))

par(mfrow=c(1,2))
plot(d_residual~x)
plot(d_residual~p)
```



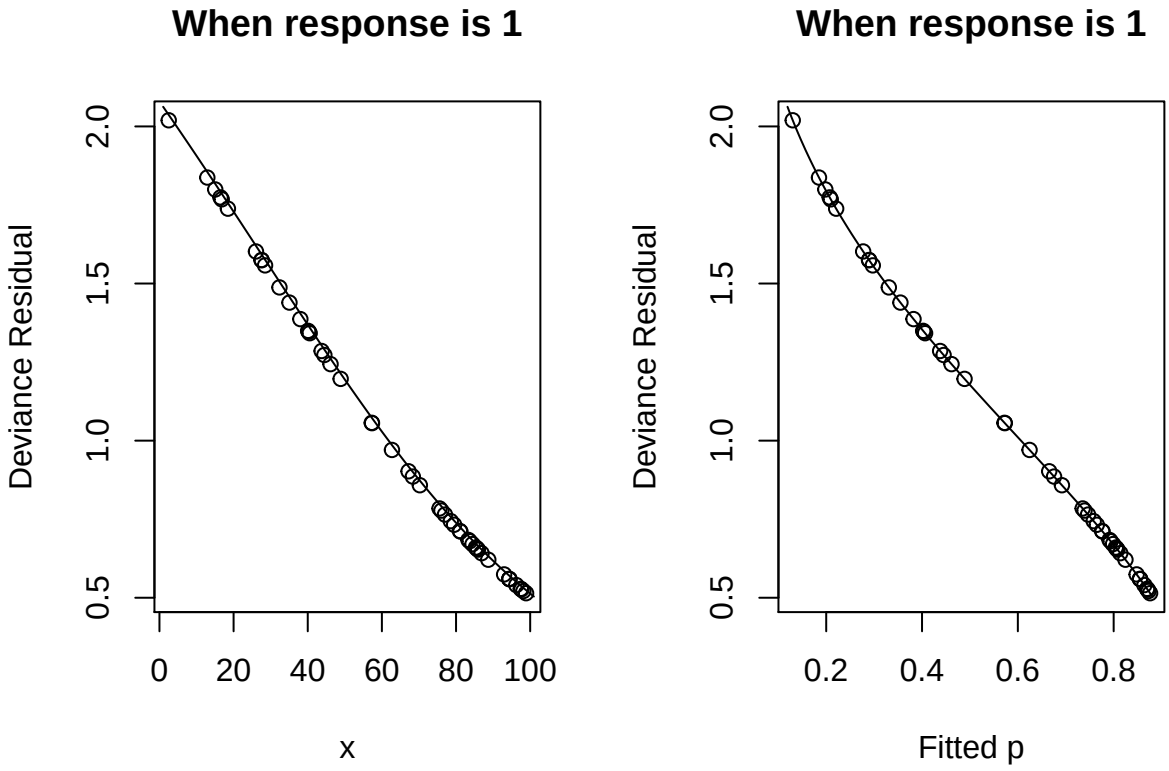


The

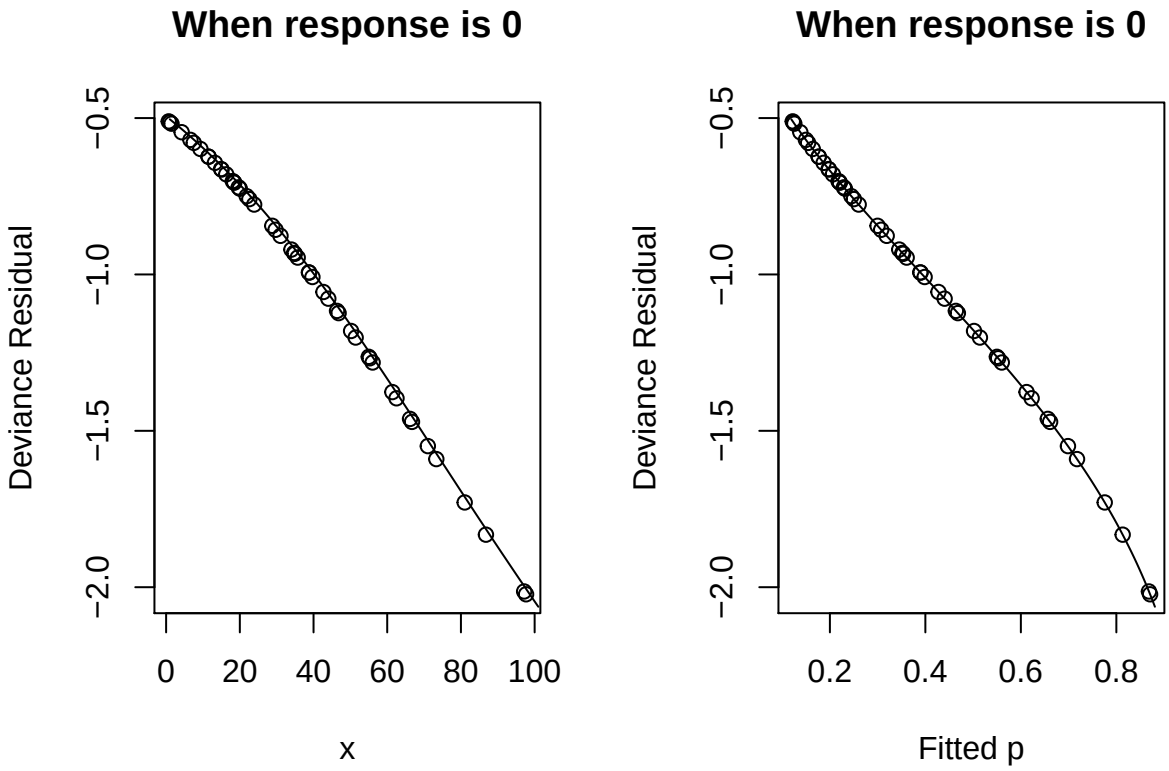
similar explanation applies here, other than the formula for deviance residual.
$r_{D_i} = sign(y - \hat{\pi}_i) * \sqrt{2 * (-y_i log\hat{\pi}_i - (1 - y_i) log(1 - \hat{\pi}_i))}$
When y = 1, $r_{D_i} = \sqrt{2 * (-log\hat{\pi})}$, when y = 0 $r_{D_i} = -\sqrt{2 * (-log(1 - \hat{\pi}_i))}$
That is why deviance residual lies on the top and the bottom line seperately.

In the same way, we generate dots and plot the lines.

```
index1 = (y==1)
index0 = (y==0)

d_res1 = d_residual[index1]
p1 = p[index1]
x1 = x[index1]

d_res0 = d_residual[index0]
p0 = p[index0]
x0 = x[index0]

temp_x = seq(0,100)
temp_p = (exp(-2+0.04*temp_x)/(1+exp(-2+0.04*temp_x)))
temp_y1 = rep(1,101)
temp_y0 = rep(0,101)
r1 = sign(temp_y1 - temp_p)*sqrt(2*(-temp_y1*log(temp_p)-(1-temp_y1)*log(1-temp_p)))
r0= sign(temp_y0 - temp_p)*sqrt(2*(-temp_y0*log(temp_p)-(1-temp_y0)*log(1-temp_p)))

par(mfrow=c(1,2))
plot(d_res1~x1,xlab='x',ylab='Deviance Residual',main='When response is 1')
lines(r1)
plot(d_res1~p1,xlab='Fitted p',ylab='Deviance Residual',main='When response is 1')
lines(temp_p,r1)
```



```
par(mfrow=c(1,2))
plot(d_res0~x0,xlab='x',ylab='Deviance Residual',main='When response is 0')
lines(r0)
plot(d_res0~p0,xlab='Fitted p',ylab='Deviance Residual',main='When response is 0')
lines(temp_p,r0)
```



Q9

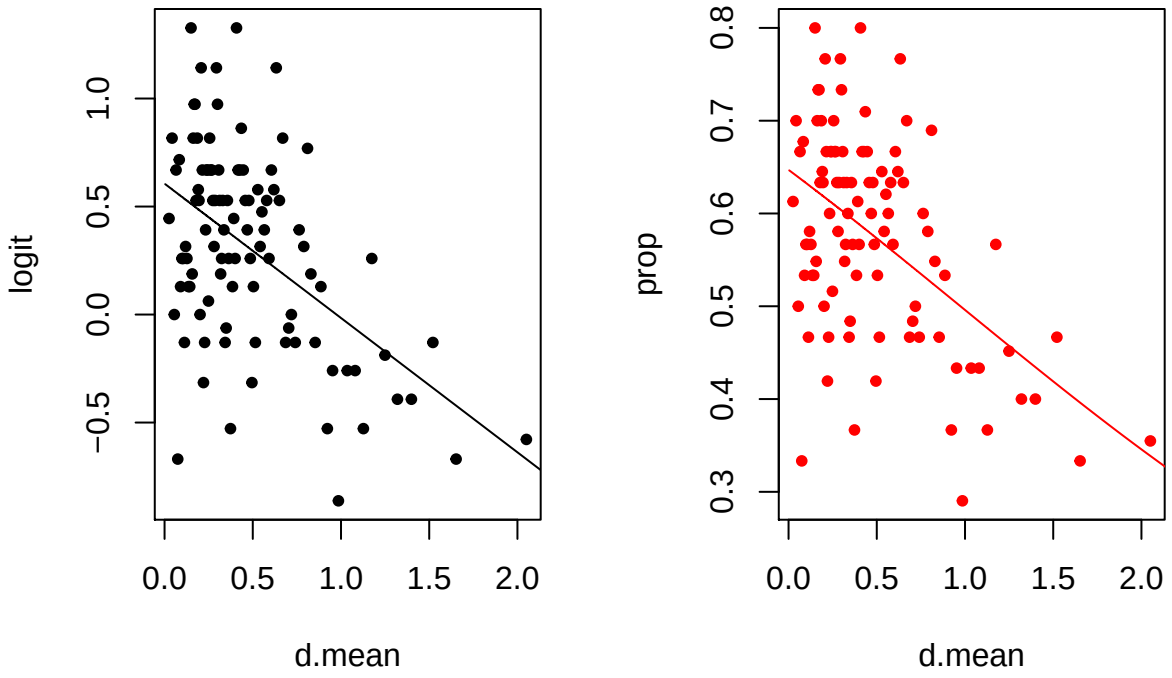
```
wells <- read.table("./wells.dat")
attach(wells)
dist100 <- dist/100
```

a

For Pearson Statistic and likelihood ratio test to be informative, we need to bin the data first.

```
m1 = glm(switch~dist100,family = 'binomial')
beta = m1$coefficients
beta = coef(m1)
ncat = 100
bins = cut(dist100,quantile(dist100,prob=c(0:ncat)/ncat),include.lowest=T)
lsat = split(switch,bins)

counts = lapply(lsat,FUN=function(x){as.numeric(x==1)})
par(mfrow=c(1,2))
logit  = as.numeric(lapply(counts,FUN=function(x){log((sum(x)+0.5)/(length(x)-sum(x)+0.5))}))
prop = as.numeric(lapply(counts,FUN=function(x){(sum(x))/(length(x))}))
d.mean = as.numeric(lapply(split(dist100,bins),FUN=function(x){mean(x)}))
```

```r
logit_ = beta[1]+beta[2]*sort(dist100)
prob_ = exp(sort(dist100)*beta[2]+beta[1])/(1+exp(sort(dist100)*beta[2]+beta[1]))

plot(d.mean,logit,pch=20)
lines(sort(dist100),beta[1]+beta[2]*sort(dist100))
plot(d.mean,prop,pch=20,col="red")
lines(sort(dist100),prob_,col='red')
```




```r
obs = lapply(counts, function(x){c(sum(x),length(x)-sum(x))})
obs = matrix(as.numeric(unlist(obs)),ncol=2,byrow=TRUE)
fit = lapply(split(dist100,bins),function(x){pi=exp(beta[1]+x*beta[2])/(1+exp(beta[1]+x*beta[2]));c(sum
fit <- matrix(as.numeric(unlist(fit)),ncol=2,byrow=TRUE)

X.2 = sum((obs-fit)^2/fit)
G.2 = 2*sum(obs*log(obs/fit))
pchisq(X.2,df=98,lower.tail=FALSE)
```

```
## [1] 0.09614898
```

```r
pchisq(G.2,df=98,lower.tail=FALSE)
```

```
## [1] 0.08295474
```

Both the Pearson X^2 and Likelihood ratio G2 statistics give the similar p values. Since both 0.09614898 and 0.08295474 $>$ 0.05, we cannot reject the null hypothesis that the model with only the intercept term is an adequate simplification of the model with the intercept and the distance as the predictor.

b

```r
m2 = glm(switch~dist100+arsenic, family = "binomial")
anova(m1, m2, test = "LRT")
```

```
## Analysis of Deviance Table
```

```
##
## Model 1: switch ~ dist100
## Model 2: switch ~ dist100 + arsenic
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1      3018     4076.2
## 2      3017     3930.7  1   145.57 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
m3 = glm(switch~dist100+arsenic+educ, family = "binomial")
anova(m2, m3, test = "LRT")
```

```
## Analysis of Deviance Table
##
## Model 1: switch ~ dist100 + arsenic
## Model 2: switch ~ dist100 + arsenic + educ
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1      3017     3930.7
## 2      3016     3910.4  1   20.235 6.848e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
m4 = glm(switch~dist100+arsenic+educ+assoc, family = "binomial")
anova(m3, m4, test = "LRT")
```

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

```
m5 = glm(switch~., family = "binomial",data = wells)
drop1(m5,test="LRT")
```

```
## Single term deletions
##
## Model:
## switch ~ arsenic + dist + assoc + educ
##         Df Deviance    AIC     LRT  Pr(>Chi)
## <none>       3907.8 3917.8
## arsenic  1   4056.1 4064.1 148.260 < 2.2e-16 ***
## dist     1   3985.2 3993.2  77.365 < 2.2e-16 ***
## assoc    1   3910.4 3918.4   2.607    0.1064
## educ     1   3927.7 3935.7  19.825 8.489e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Likelihood Ratio test makes use of the deviance, we try to add some predictors to our model and find assoc may not be very useful. Again it is testified by the drop out method. So currently, the model with all predictors except for assoc is our best choice. Then we try to find some meaningful interaction.

```
m6 =glm(switch~dist100+arsenic+educ+dist100:arsenic+dist100:educ+arsenic:educ, family = "binomial")
drop1(m6,test='LRT')
```

```
## Single term deletions
```

```
##
## Model:
## switch ~ dist100 + arsenic + educ + dist100:arsenic + dist100:educ +
##     arsenic:educ
##                 Df Deviance    AIC    LRT Pr(>Chi)
## <none>               3891.7 3905.7
## dist100:arsenic  1   3893.0 3905.0 1.2896 0.256125
## dist100:educ     1   3901.1 3913.1 9.3534 0.002226 **
## arsenic:educ     1   3894.5 3906.5 2.7334 0.098269 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
m7 =glm(switch~dist100+arsenic+educ+dist100:educ+arsenic:educ, family = "binomial")
drop1(m7,test='LRT')
```

```
## Single term deletions
##
## Model:
## switch ~ dist100 + arsenic + educ + dist100:educ + arsenic:educ
##              Df Deviance    AIC    LRT Pr(>Chi)
## <none>            3893.0 3905.0
## dist100:educ  1   3902.9 3912.9 9.8786 0.001672 **
## arsenic:educ  1   3896.2 3906.2 3.1170 0.077479 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(m7)
```

```
##
## Call:
## glm(formula = switch ~ dist100 + arsenic + educ + dist100:educ +
##     arsenic:educ, family = "binomial")
##
## 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 ***
## arsenic       0.39748    0.06210   6.400 1.55e-10 ***
## educ         -0.02481    0.01997  -1.243  0.21402
## dist100:educ  0.08278    0.02662   3.109  0.00187 **
## arsenic:educ  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
```

Agian, using drop one out, we find the model above (with dist100, arsenic, educ, dist100*educ, arsenic*educ) with lowest AIC score. The model shows that one unit increase in dist100, arsenic, educ will result in the odds of switch to safe places times exp{-1.31799+0.08278*educ}, exp{0.39748+0.01912*educ}, exp{-0.02481} respectively, and it means that for more time in receiving education, the effect of one unit increase in dist100 and arsenic will result in more extra positive effect on odds of switching to safe wells.

c

```
edu= as.numeric(educ>=9)
m8 =glm(switch~dist100+arsenic+edu+dist100:edu+arsenic:edu, family = "binomial")
summary(m7)
```

```
##
## Call:
## glm(formula = switch ~ dist100 + arsenic + educ + dist100:educ +
##     arsenic:educ, family = "binomial")
##
## 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 ***
## arsenic       0.39748    0.06210   6.400 1.55e-10 ***
## educ         -0.02481    0.01997  -1.243  0.21402
## dist100:educ  0.08278    0.02662   3.109  0.00187 **
## arsenic:educ  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
```

```
summary(m8)
```

```
##
## Call:
## glm(formula = switch ~ dist100 + arsenic + edu + dist100:edu +
##     arsenic:edu, family = "binomial")
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.601  -1.197   0.723   1.058   1.862
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.01601    0.08833   0.181    0.856
## dist100     -1.13445    0.12130  -9.352  < 2e-16 ***
```

```
## arsenic      0.45235    0.04504  10.044  < 2e-16 ***
## edu         -0.17650    0.21578  -0.818    0.413
## dist100:edu  1.12280    0.27150   4.136 3.54e-05 ***
## arsenic:edu  0.18625    0.12583   1.480    0.139
## ---
## 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: 3866.6  on 3014  degrees of freedom
## AIC: 3878.6
##
## Number of Fisher Scoring iterations: 4
```

According to the performance based on deviance and AIC score, we see the model 8 with having education as binary predictor is obviously better than m7.

```
drop1(m8,test='LRT')
```

```
## Single term deletions
##
## Model:
## switch ~ dist100 + arsenic + edu + dist100:edu + arsenic:edu
##             Df Deviance    AIC     LRT  Pr(>Chi)
## <none>           3866.6 3878.6
## dist100:edu  1   3884.6 3894.6 17.9346 2.286e-05 ***
## arsenic:edu  1   3869.0 3879.0  2.3019    0.1292
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
m9 = glm(switch~dist100+arsenic+edu+dist100:edu, family = "binomial")
summary(m9)
```

```
##
## Call:
## glm(formula = switch ~ dist100 + arsenic + edu + dist100:edu,
##     family = "binomial")
##
## 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 ***
## edu          0.04607    0.15625   0.295    0.768
## dist100:edu  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
```

By dropping one out, we fail to get a better model due to increasing deviance and AIC score.

Now back to model 8, compared to its original version, we find the coefficient of education and other 2 interaction terms about education becomes much more larger, while the sign of all terms remain unchanged and the coefficient of the other terms doesn't change much. The effect of education in binary case is much more obvious.

d

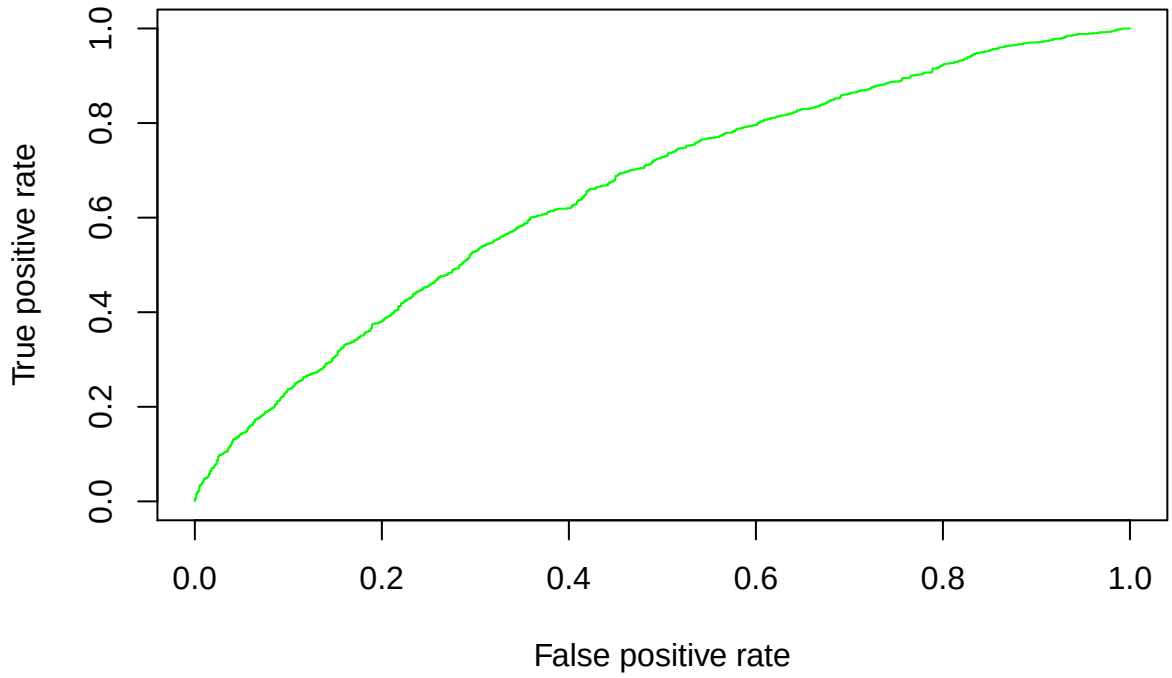
```
library(ROCR)
```

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

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

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

```
pred = prediction(fitted(m7),switch)
perf = performance(pred,'tpr','fpr')
plot(perf, col='Green')
```




```
performance(pred,'auc')
```

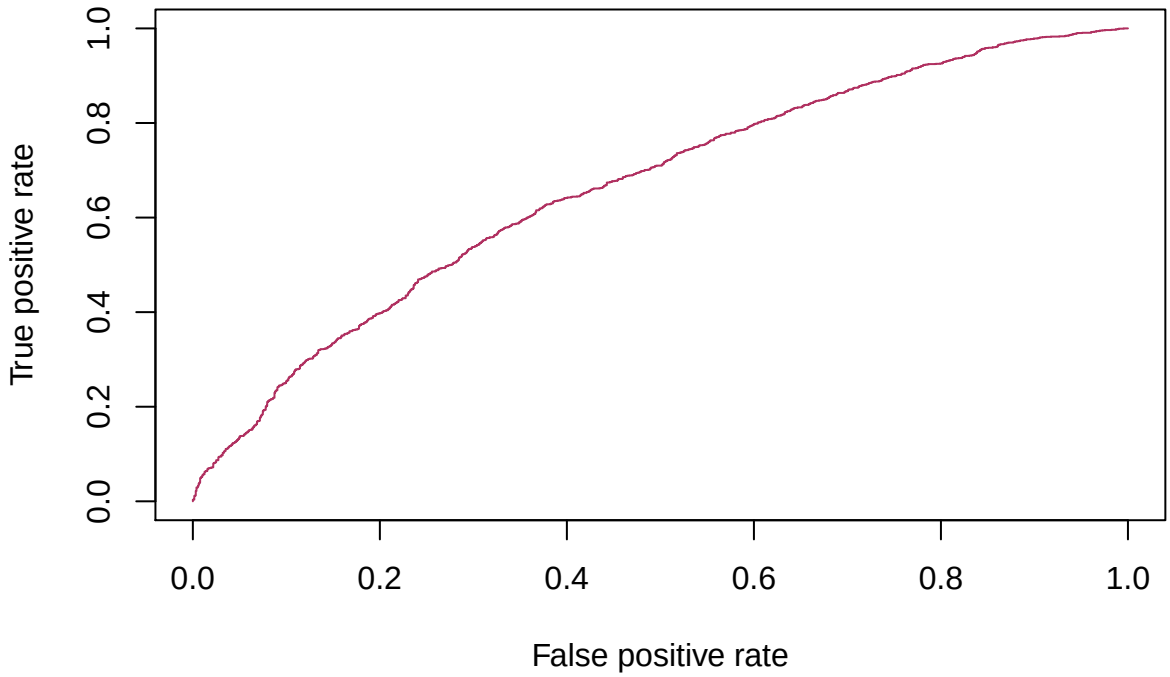
```
## An object of class "performance"
## Slot "x.name":
## [1] "None"
##
```

```
## Slot "y.name":
## [1] "Area under the ROC curve"
##
## Slot "alpha.name":
## [1] "none"
##
## Slot "x.values":
## list()
##
## Slot "y.values":
## [[1]]
## [1] 0.6582097
##
##
## Slot "alpha.values":
## list()
```

```
AIC(m7)
```

```
## [1] 3905.033
```

```
pred = prediction(fitted(m8),switch)
perf = performance(pred,'tpr','fpr')
plot(perf,col='Maroon')
```




```
performance(pred,'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] 0.6641491
##
##
## Slot "alpha.values":
## list()
```

```
AIC(m8)
```

```
## [1] 3878.648
```

Model 8 has both smaller AIC score and bigger the area under the curve 0.6641491 compared to model 7 0.6582097. Therefore, model 8 is our best choice so far.