Assignment 3 - MATH 523_260677676

YUNHEUM DAN SEOL

2018-03-13

A8.

(a)

$$\exists\{y_{i1},\ y_{i2},\ ...,\ y_{i(ni-1)}, \forall i = 1,\ ...\ ,\ N$$

$$\text{if}\ \ Y_{ij} \sim \text{Bernoulli}(\pi_i)$$

i=1, ... , N and j=1, ..., n_i Then for each y_{ij} we have probability mass function $p(y_{ij};\pi_i)$ as below:

$$p(y_{ij};\pi_i) = \pi_i^{y_{ij}}(1-\pi_i)^{1-y_{ij}}$$

so

$$L(\beta) = \Pi_{i=1}^{N}\Pi_{j=1}^{n_i}\pi_i^{y_{ij}}(1-\pi_i)^{1-y_{ij}}$$

$$= \Pi_{i=1}^{N}\pi_i^{\sum_{j=1}^{N} y_{ij}}(1-\pi_i)^{n_i-\sum_{j=1}^{N} y_{ij}}$$

$$= \Pi_{i=1}^{N}\pi_i^{\sum_{j=1}^{N} y_{ij}}(1-\pi_i)^{n_i-\sum_{j=1}^{N} y_{ij}}$$

Now, define $Y_i = \frac{\sum_{j=1}^{n_i} Y_{ij}}{n_i}$ Then we have

$$n_iY_i \sim \text{Binomial}(n_i,\pi_i)\ \ \text{i=1,...,N}$$

with probability mass function

$$p(n_iy_i;n_i,\pi_i) = \begin{pmatrix} n_i \\ n_iy_i \end{pmatrix}\pi_i^{n_iy_i}(1-\pi_i)^{n_i-n_iy_i}$$

then

$$L(\beta) = \Pi_{i=1}^{N}\begin{pmatrix} n_i \\ n_iy_i \end{pmatrix}\pi_i^{n_iy_i}(1-\pi_i)^{n_i-n_iy_i}$$

They share the kernel part since

$$\Pi_{i=1}^{N}\pi_i^{n_iy_i}(1-\pi_i)^{n_i-n_iy_i} = \Pi_{i=1}^{N}\pi_i^{\sum_{j=1}^{N} y_{ij}}(1-\pi_i)^{n_i-\sum_{j=1}^{N} y_{ij}}$$

This implies that the parameter estimates for grouped and ungrouped data would be the same.

(b)

For a glm we have the likelihood equation (score function)

$$\frac{\partial l(\beta)}{\partial \beta_j} := \sum_{i=1}^{n} \frac{y_i - \mu_i}{Var(y_i)} \frac{\partial \mu_i}{\partial \eta_i} x_{ij} = 0$$

$$\text{if } \; Y_{ij} \sim \text{Bernoulli}(\pi_i)$$

i=1, . . . , N and j=1, . . . , n_i

$$\mathrm{E}[Y_{ij}] = \pi_i$$

$$\mathbb{VAR}[Y_{ij}] = \pi_i(1 - \pi_i)$$

so we have

$$\sum_{i=1}^{N} \sum_{j=1}^{n_i} \frac{y_i j - \pi_i}{\pi_i(1 - \pi_i)} \frac{\partial \pi_i}{\partial \eta_i} x_{ij} = 0$$

$$n_i Y_i \sim \text{Binomial}(n_i, \pi_i) \;\; \text{i=1,...,N}$$

$$\mathrm{E}[Y_i] = \pi_i$$

$$\sum_{i=1}^{N} \frac{n_i(y_i - \pi_i)}{\pi_i(1 - \pi_i)} \frac{\partial \pi_i}{\partial \eta_i} x_{ij} = 0$$

$$\mathbb{VAR}[Y_i] = \frac{\pi_i(1 - \pi_i)}{n_i}$$

$$\sum_{i=1}^{N} \sum_{j=1}^{n_i} \frac{y_i - \pi_i}{\pi_i(1 - \pi_i)} \frac{\partial \pi_i}{\partial \eta_i} x_{ij} = 0$$

It is known from class that For saturated models we have $y_i = \widehat{\mu_i}$ The difference of the number of parameters (for binomial responses we have N $\widehat{\mu_i} = y_i$'s, whereas for for bernoulli responses we have n = $\sum_{i=1}^{N} n_i$) results a difference in likelihood equations. ##(c)

```
#grouped data
x <- c(0,1,3)
trials <- c(3,4,5)
successes <-c(1,2,4)
failures <- trials-successes
p <- successes/trials
grouped <- data.frame(cbind(successes,x))
f0a <- glm(cbind(successes, failures)~1, family=binomial, data=grouped)
f1a <- glm(cbind(successes, failures)~x, family=binomial, data=grouped)

y_u <- c(1,0,0,1,1,0,0,1,1,1,1,0)
x_u <- c(0,0,0,1,1,1,1,3,3,3,3,3)
```

```r
ungrouped <- data.frame(cbind(y_u,x_u))
f0b <- glm(cbind(y_u, 1-y_u)~1, family=binomial, data=ungrouped)
f1b <- glm(cbind(y_u, 1-y_u)~x_u, family=binomial, data=ungrouped)
#incercept-only model:summary for the grouped data
summary(f0a)
```

```
##
## Call:
## glm(formula = cbind(successes, failures) ~ 1, family = binomial,
##     data = grouped)
##
## Deviance Residuals:
##       1        2        3
## -0.8722  -0.3357   1.0290
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.3365     0.5855   0.575    0.566
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1.9324  on 2  degrees of freedom
## Residual deviance: 1.9324  on 2  degrees of freedom
## AIC: 9.301
##
## Number of Fisher Scoring iterations: 4
```

```r
#intercept-only mode: summary for the ungrouped data
summary(f0b)
```

```
##
## Call:
## glm(formula = cbind(y_u, 1 - y_u) ~ 1, family = binomial, data = ungrouped)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.323  -1.323   1.038   1.038   1.038
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.3365     0.5855   0.575    0.566
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 16.301  on 11  degrees of freedom
## Residual deviance: 16.301  on 11  degrees of freedom
## AIC: 18.301
##
## Number of Fisher Scoring iterations: 4
```

```r
#model with x: summary for the grouped data
summary(f1a)
```

```
##
## Call:
```

```
## glm(formula = cbind(successes, failures) ~ x, family = binomial,
##     data = grouped)
##
## Deviance Residuals:
## 1  2  3
## 0  0  0
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.6931     0.9727  -0.713    0.476
## x             0.6931     0.5335   1.299    0.194
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1.9324  on 2  degrees of freedom
## Residual deviance: 0.0000  on 1  degrees of freedom
## AIC: 9.3687
##
## Number of Fisher Scoring iterations: 4
```

```
#model with x: summary for the ungrouped data
summary(f1b)
```

```
##
## Call:
## glm(formula = cbind(y_u, 1 - y_u) ~ x_u, family = binomial, data = ungrouped)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7941  -0.9697   0.6681   0.7954   1.4823
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.6931     0.9727  -0.713    0.476
## x_u           0.6931     0.5335   1.299    0.194
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 16.301  on 11  degrees of freedom
## Residual deviance: 14.368  on 10  degrees of freedom
## AIC: 18.368
##
## Number of Fisher Scoring iterations: 4
```

```
#comparing the fitted values for M1: they are the same
coefficients(f1a)
```

```
## (Intercept)           x
##  -0.6931472   0.6931472
```

```
coefficients(f1b)
```

```
## (Intercept)         x_u
##  -0.6931472   0.6931472
```

```
#comparing the deviances for M1: they are NOT the same!
deviance(f1a)
```

```
## [1] 0
```

```
deviance(f1b)
```

```
## [1] 14.36829
```

```
#comparing the differences between the M0 and M1:
deviance(f0a) - deviance(f1a)
```

```
## [1] 1.932352
```

```
deviance(f0b) - deviance(f1b)
```

```
## [1] 1.932352
```

Our response is a bernoulli random variable (i.e. $Y_i \sim$ Bernoulli(π_i)) thus discrete. The innate discreteness of response is, therefore, the reason why the residuals do not behave similar to normal random variables. This non-normal behavior of residual points explains the evident linear patterns in our both residual plots.

We can elaborate on this. From class we learned that when we have a binary response the deviance becomes just the function of β. Not only this gives us difficulty to use deviance as a measure for Goodness-of-fit test, but also is a reason of linear pattern. (b)

```
set.seed(329)
x1 <-runif(100, min=0, max=1)
```

A9.

(a)

```
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
```

```
set.seed(329)
x1 <-runif(100, min=0, max=1)
pi <- exp(-2+0.4*x1)/(exp(-2+0.4*x1)+1)
head(x1)
```

```
## [1] 0.6003120 0.5326102 0.5406586 0.5087419 0.9003864 0.2576369
```

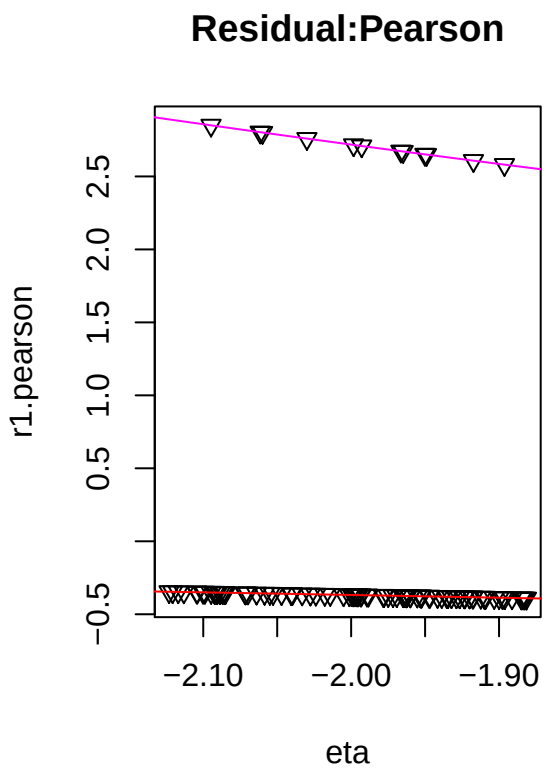
```
head(pi)
```

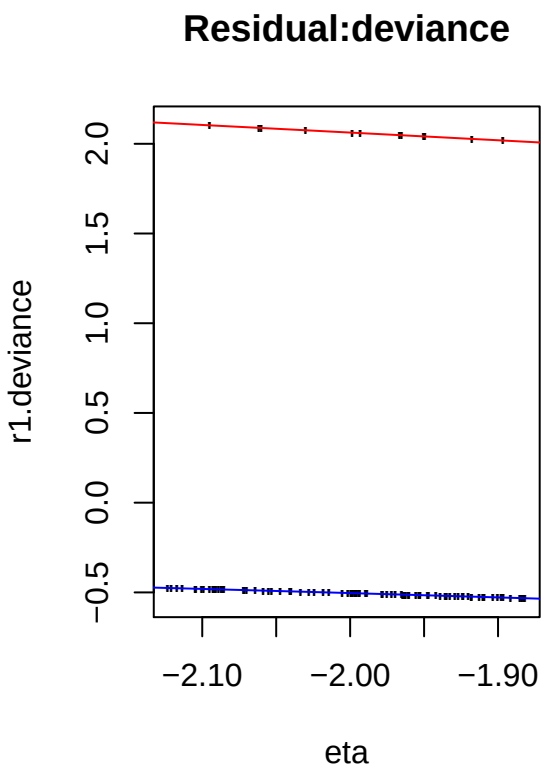
```
## [1] 0.1468060 0.1434463 0.1438424 0.1422773 0.1624861 0.1304546
```

```
y <- rbern(100, pi)
logistic1 <- glm(y~x1, family=binomial)
summary(logistic1)
```

```
##
## Call:
## glm(formula = y ~ x1, family = binomial)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.5326  -0.5165  -0.5045  -0.4832   2.1028
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.1284     0.6942  -3.066  0.00217 **
## x1            0.2500     1.1298   0.221  0.82487
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 73.385  on 99  degrees of freedom
## Residual deviance: 73.336  on 98  degrees of freedom
## AIC: 77.336
##
## Number of Fisher Scoring iterations: 4
```

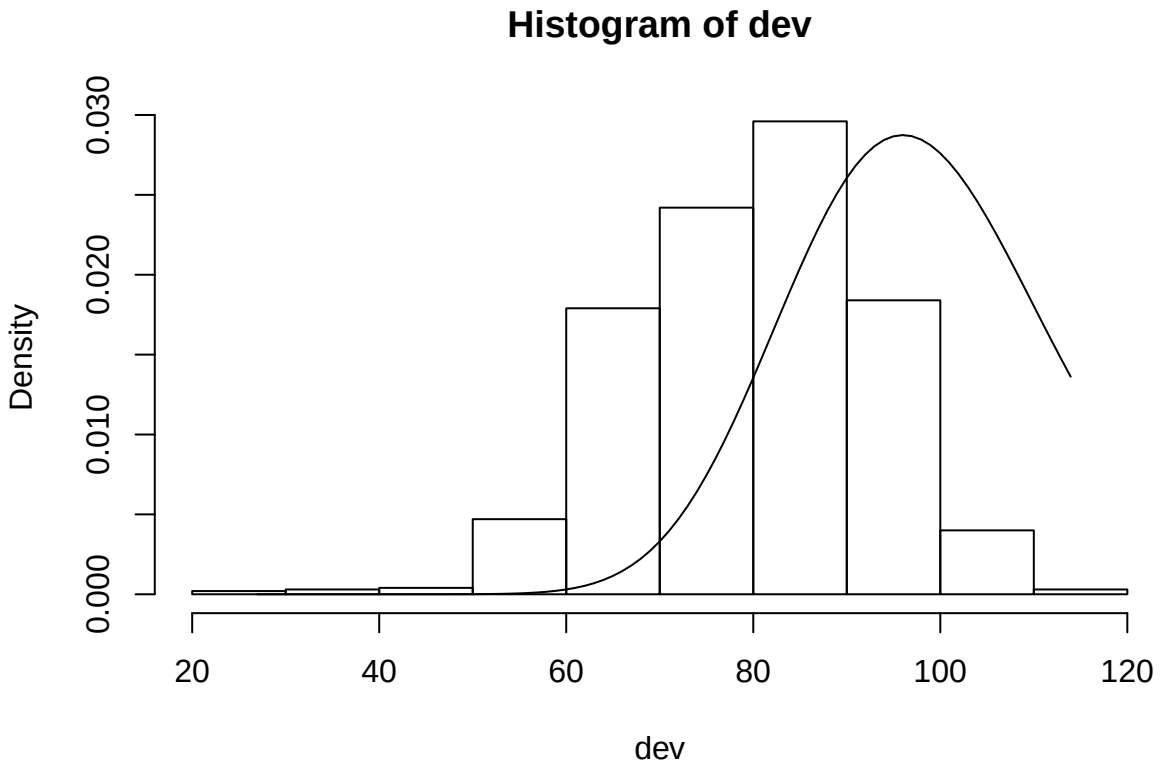
```
b1 <- -2
b2 <- 0.4


pi <- lapply(x1, function(i){exp(-2 + 0.4*i)/(1 + exp(-2 + 0.4*i))})
vec <- c()
for (i in 1:100){

  y<- c(vec,sample(0:1, 1, prob = c(1-unlist(pi)[i], unlist(pi)[i]))) }

pi <- unlist(pi)

r1.pearson <- residuals(logistic1,"pearson")

r1.deviance <- residuals(logistic1,"deviance")
eta <- predict(logistic1,type="link")
par(mfrow=c(1,2))
plot(r1.pearson~eta,main="Residual:Pearson", pch=25)
x2 = seq(-3.2,-0.8, length =201)
y2 = lapply(x2, FUN =function(x) exp(-x/2))
lines(x2, y2, lty="solid", lwd=1, col='magenta')
x3 = seq(-3.2,-0.8, length =201)
y3 = lapply(x3, FUN =function(x){ (-(exp(x))/sqrt(exp(x)))})
lines(x3,y3, lty="solid", lwd=1, col= 'red')
plot(r1.deviance~eta,main="Residual:deviance", pch=39)
xa = seq(-3.2,-0.8, length =201)
ya = lapply(xa, FUN =function(x) -sqrt(2*log(1+exp(x))))
lines(xa, ya, lty="solid", lwd=1, col='blue')
xb = seq(-3.2,-0.8, length =201)
yb = lapply(xb, FUN =function(x){sqrt(-2*log(exp(x)/(1+exp(x))))})
lines(xb,yb, lty="solid", lwd=1, col= 'red')
```





##(b)

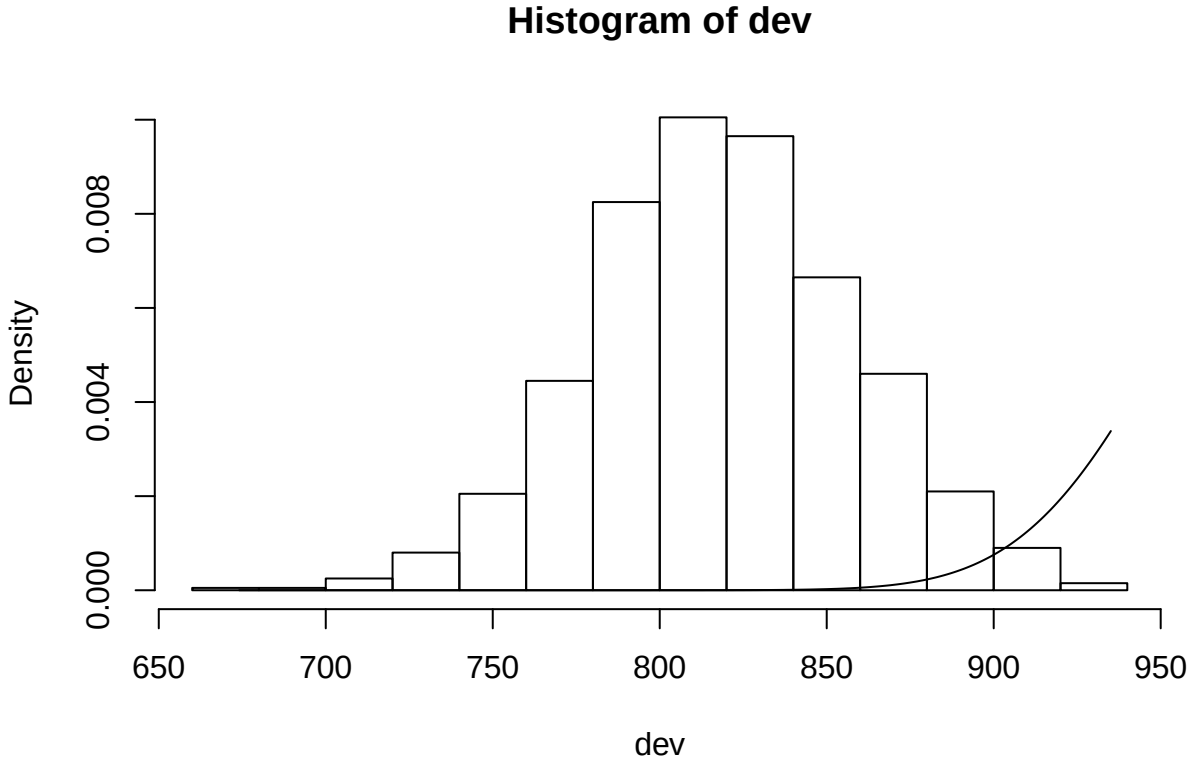
```
Duplicate <- function(m,N,a,b) {
x  <- runif(m,a,b)
pi = exp(-2+0.4*x)/(1+exp(-2+0.4*x))
dev <- rep(NA,N)
for (i in 1:N)
{
bern = rbern(m,pi)
model = glm(bern~x,family=binomial)

dev[i] = deviance(model) }
  #histogram  for 1000 samples of hundreds
hist(dev,prob=TRUE)
  #pdf of chi-squared distribution
lines(min(dev):max(dev), dchisq( min(dev):max(dev),df=m-2 ) ) }
Duplicate(100, 1000, 0, 1)
```



It is known that as n increases a binomial random variable (i.e. sum of independent Bernoulli random variables) the histogram (probability mass function) would look closer to a normal random variable by the Central Limit Theorem. Nevertheless, the responses are binary, so the deviance histogram would not look different from χ^2_{98} distribution.

(c)

```
Duplicate(1000,1000,0,1)
```



The histogram of deviance even looks closer to the normal distribution, yet the density of χ^2 distribution started deviating from looking closer to that of normal.

A10.

For a logistic regression, the model formulae would be

$$\pi_i = exp(\sum_{j=1}^{p} \beta_j x_{ij}) \iff ln(\frac{\pi_i}{1-\pi_i}) = \sum_{j=1}^{p} \beta_j x_{ij}$$

We have

$$x = \text{age}$$

$$Y = \begin{cases} 1 \text{ if using facebook} \\ 0 \text{ if not using facebook} \end{cases}$$

and

$$0.8 < P(Y=1|x=18) < 0.9 \implies ln(\frac{0.8}{0.2}) = ln(4) < logit(P(Y=1|x=18)) < ln(\frac{0.9}{0.1}) = ln(9)$$

Likewise, we have

$$0.2 < P(Y=1|x=65) < 0.3 \implies ln(\frac{0.2}{0.8}) = ln(\frac{1}{4}) < logit(P(Y=1|x=65)) < ln(\frac{0.3}{0.7})$$

It follows that

$$-ln(9) < -logit(P(Y=1|x=18)) < -ln(4)$$

$$ln(\frac{1}{4}) < logit(P(Y=1|x=65)) < ln(\frac{0.3}{0.7})$$

implying

$$ln(\frac{1}{4}) + ln(\frac{1}{9}) < logit(P(Y=1|x=65) - logit(P(Y=1|x=18)) < ln(\frac{3}{7}) + ln(\frac{1}{4})$$

Since we know

$$logit(P(Y=1|x=65) - logit(P(Y=1|x=18)) = (65-18)\beta_j = 47\beta_j$$

$$ln(\frac{1}{4}) + ln(\frac{1}{9}) < 47\beta_j < ln(\frac{3}{7}) + ln(\frac{1}{4})$$

```
round(c(log(1/4)+log(1/9), log(3/7)+log(1/4)),2)
```

```
## [1] -3.58 -2.23
```

$$\implies -3.58 < 47\beta_j < -2.23 \implies -3.58/47 < \beta_j < -2.23/47$$

```
round(c((log(1/4)+log(1/9))/47, (log(3/7)+log(1/4))/47),3)
```

```
## [1] -0.076 -0.048
```

so

$$= -0.076 < \beta_j < -0.048$$

#A11.

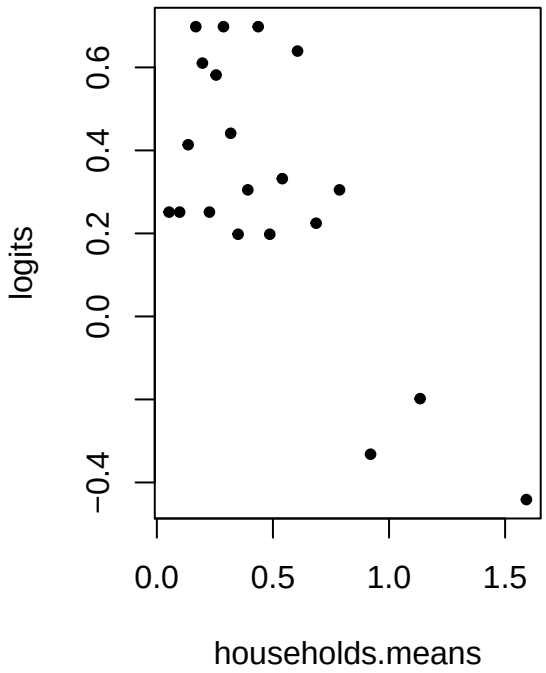
```
wells <- read.table("wells.dat")
attach(wells)
dist100 <- dist/100
head(wells)
```

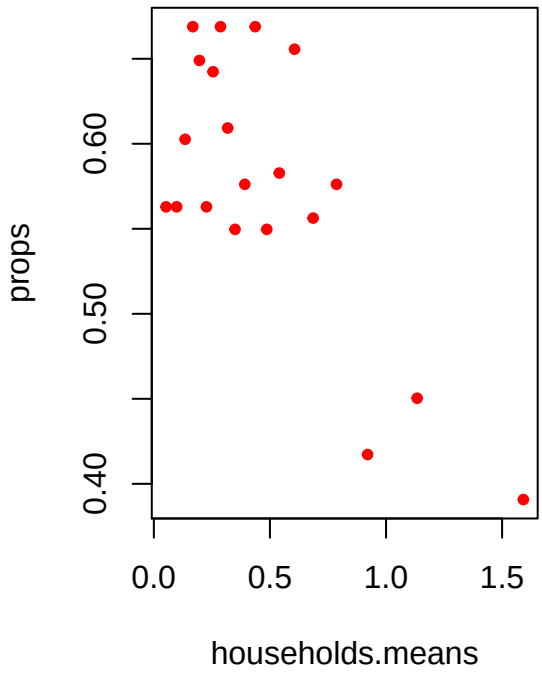
```
##   switch arsenic   dist assoc educ
## 1      1    2.36 16.826     0    0
## 2      1    0.71 47.322     0    0
## 3      0    2.07 20.967     0   10
## 4      1    1.15 21.486     0   12
## 5      1    1.10 40.874     1   14
## 6      1    3.90 69.518     1    9
```

```
dim(wells)
```

```
## [1] 3020    5
```

```
#binning dist/100(distance in 100m units to the closest known safe well) into 20 categories
#and computing the proportion of households that switched the well
ncat <- 20
bins <- cut(dist100, quantile(dist100, prob=c(0:ncat)/ncat), include.lowest=TRUE)
households <- split(switch, bins)
par(mfrow=c(1,2))
logits <- as.numeric(lapply(households,FUN=function(x){log((sum(x)+0.5)/(length(x)-sum(x)+0.5))}))
props <- as.numeric(lapply(households,FUN=function(x){sum(x)/length(x)}))
households.means <- as.numeric(lapply(split(dist/100, bins),mean))
plot(households.means, logits, pch=20)
plot(households.means, props, pch=20, col="red")
```






```
fit1 <- glm(cbind(switch, 1-switch)~dist100, family='binomial')
summary(fit1)
```

```
##
## Call:
## glm(formula = cbind(switch, 1 - switch) ~ dist100, family = "binomial")
##
## 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
```

```
anova(fit1)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: cbind(switch, 1 - switch)
##
```

```
## Terms added sequentially (first to last)
##
##
##          Df Deviance Resid. Df Resid. Dev
## NULL                     3019     4118.1
## dist100  1   41.861      3018     4076.2
```

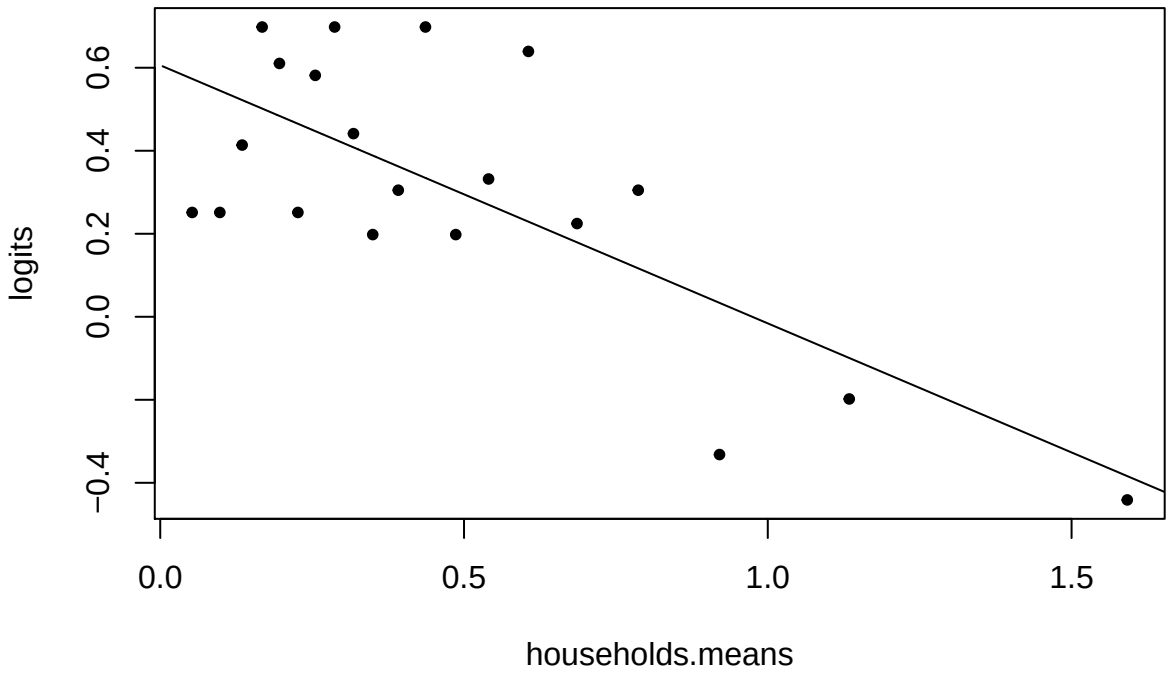
```
sum(residuals(fit1, types = "pearson")^2)
```

```
## [1] 4076.238
```

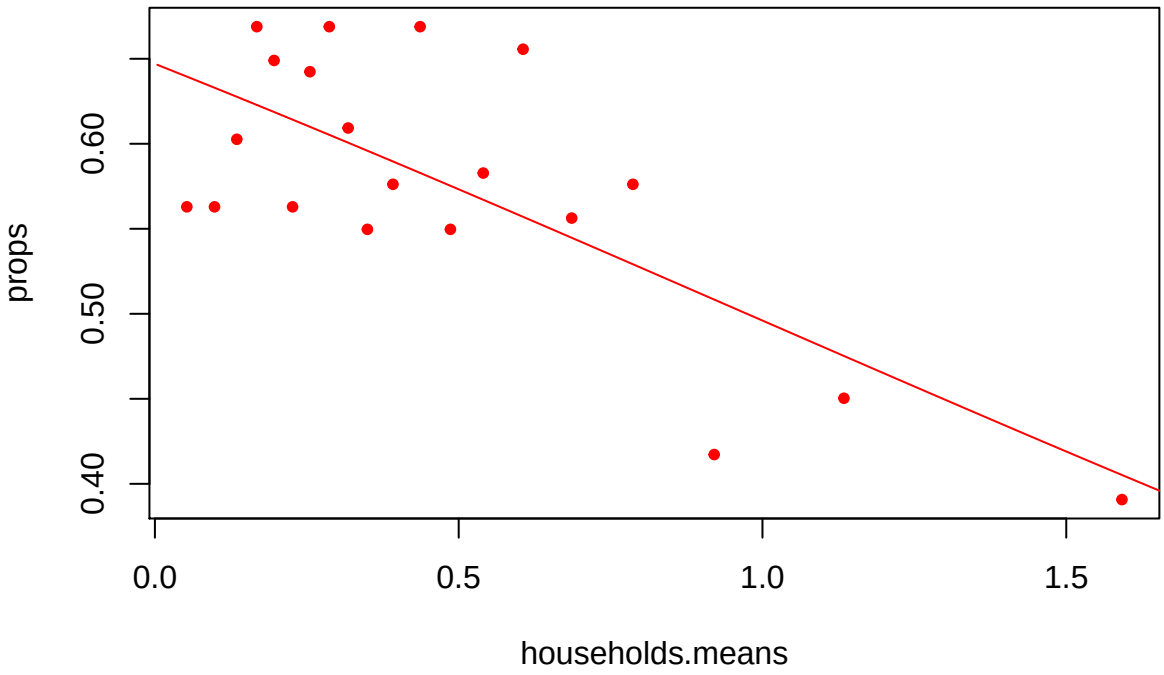
```
sum(residuals(fit1, types="deviance")^2)
```

```
## [1] 4076.238
```

```
beta <- coefficients(fit1)
prob <- exp(beta[1]+sort(dist100)*beta[2])/(1+exp(beta[1]+sort(dist100)*beta[2]))
plot(households.means, logits, pch=20)
lines(sort(dist100),beta[1]+beta[2]*sort(dist100))
```




```
plot(households.means, props, pch=20, col="red")
lines(sort(dist100),prob,col="red")
```




```
# Checking the fit of m1 by means of formal goodness of fit tests

observed <- lapply(households,FUN=function(x){c(sum(x),length(x)-sum(x))})

observed <- matrix(as.numeric(unlist(observed)),ncol=2,byrow=TRUE)

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)

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


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

pchisq(X.2, df=18,lower.tail=FALSE)
```

```
## [1] 0.01310562
```

```
pchisq(G.2, df=18,lower.tail=FALSE)
```

```
## [1] 0.01252289
```

We can conclude at $\alpha = 0.05$ level that there is significant evidence against the null that that there is a significant difference between the expected frequencies and observed frequencies in at least one category

(b)

```
fit_a <- glm(cbind(switch, 1-switch)~(dist100+arsenic+educ+assoc)^2, family = binomial)
summary(fit_a)
```

```
##
## Call:
## glm(formula = cbind(switch, 1 - switch) ~ (dist100 + arsenic +
##     educ + assoc)^2, family = binomial)
```

```
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.747  -1.195   0.725   1.069   1.929
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -0.05775    0.17648  -0.327  0.74351
## dist100         -1.21607    0.27927  -4.354 1.33e-05 ***
## arsenic          0.53369    0.09446   5.650 1.61e-08 ***
## educ            -0.01324    0.02142  -0.618  0.53650
## assoc            0.13889    0.18922   0.734  0.46293
## dist100:arsenic -0.11005    0.10320  -1.066  0.28624
## dist100:educ     0.08385    0.02682   3.126  0.00177 **
## dist100:assoc    0.21882    0.21480   1.019  0.30834
## arsenic:educ     0.01744    0.01101   1.584  0.11315
## arsenic:assoc   -0.16256    0.08424  -1.930  0.05364 .
## educ:assoc      -0.02757    0.01981  -1.391  0.16408
## ---
## 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: 3883.4  on 3009  degrees of freedom
## AIC: 3905.4
##
## Number of Fisher Scoring iterations: 4
```

```
anova(fit_a)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: cbind(switch, 1 - switch)
##
## Terms added sequentially (first to last)
##
##
##                 Df Deviance Resid. Df Resid. Dev
## NULL                             3019     4118.1
## dist100          1   41.861      3018     4076.2
## arsenic          1  145.570      3017     3930.7
## educ             1   20.235      3016     3910.4
## assoc            1    2.607      3015     3907.8
## dist100:arsenic  1    2.475      3014     3905.4
## dist100:educ     1   13.851      3013     3891.5
## dist100:assoc    1    0.247      3012     3891.3
## arsenic:educ     1    2.643      3011     3888.6
## arsenic:assoc    1    3.278      3010     3885.3
## educ:assoc       1    1.936      3009     3883.4
```

```r
anova(fit1, fit_a)
```

```
## Analysis of Deviance Table
##
## Model 1: cbind(switch, 1 - switch) ~ dist100
## Model 2: cbind(switch, 1 - switch) ~ (dist100 + arsenic + educ + assoc)^2
##   Resid. Df Resid. Dev Df Deviance
## 1      3018     4076.2
## 2      3009     3883.4  9   192.84
```

```r
drop1(fit_a, test="Chisq") #deviance
```

```
## Single term deletions
##
## Model:
## cbind(switch, 1 - switch) ~ (dist100 + arsenic + educ + assoc)^2
##                 Df Deviance    AIC     LRT Pr(>Chi)
## <none>               3883.4 3905.4
## dist100:arsenic  1   3884.5 3904.5  1.1357 0.286564
## dist100:educ     1   3893.4 3913.4 10.0028 0.001563 **
## dist100:assoc    1   3884.4 3904.4  1.0365 0.308645
## arsenic:educ     1   3885.9 3905.9  2.5310 0.111628
## arsenic:assoc    1   3887.1 3907.1  3.7029 0.054319 .
## educ:assoc       1   3885.3 3905.3  1.9357 0.164134
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
drop1(fit_a, test="LRT") #LRT for comparison
```

```
## Single term deletions
##
## Model:
## cbind(switch, 1 - switch) ~ (dist100 + arsenic + educ + assoc)^2
##                 Df Deviance    AIC     LRT Pr(>Chi)
## <none>               3883.4 3905.4
## dist100:arsenic  1   3884.5 3904.5  1.1357 0.286564
## dist100:educ     1   3893.4 3913.4 10.0028 0.001563 **
## dist100:assoc    1   3884.4 3904.4  1.0365 0.308645
## arsenic:educ     1   3885.9 3905.9  2.5310 0.111628
## arsenic:assoc    1   3887.1 3907.1  3.7029 0.054319 .
## educ:assoc       1   3885.3 3905.3  1.9357 0.164134
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
fit_b <- update(fit_a, ~.-dist100:arsenic-dist100:assoc-arsenic:educ-arsenic:assoc-educ:assoc)
```

```r
summary(fit_b)
```

```
##
## Call:
## glm(formula = cbind(switch, 1 - switch) ~ dist100 + arsenic +
##     educ + assoc + dist100:educ, family = binomial)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6797  -1.2109   0.7511   1.0561   1.9229
```

```
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.065686   0.115872   0.567 0.570792
## dist100      -1.397341   0.171860  -8.131 4.27e-16 ***
## arsenic       0.479254   0.041993  11.413  < 2e-16 ***
## educ         -0.003208   0.015287  -0.210 0.833793
## assoc        -0.134331   0.077201  -1.740 0.081858 .
## dist100:educ  0.097081   0.025710   3.776 0.000159 ***
## ---
## 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: 3893.1  on 3014  degrees of freedom
## AIC: 3905.1
##
## Number of Fisher Scoring iterations: 4
```

```
anova(fit_b)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: cbind(switch, 1 - switch)
##
## Terms added sequentially (first to last)
##
##
##              Df Deviance Resid. Df Resid. Dev
## NULL                         3019     4118.1
## dist100       1   41.861      3018     4076.2
## arsenic       1  145.570      3017     3930.7
## educ          1   20.235      3016     3910.4
## assoc         1    2.607      3015     3907.8
## dist100:educ  1   14.702      3014     3893.1
```

```
drop1(fit_b, test="Chisq")
```

```
## Single term deletions
##
## Model:
## cbind(switch, 1 - switch) ~ dist100 + arsenic + educ + assoc +
##     dist100:educ
##              Df Deviance    AIC     LRT  Pr(>Chi)
## <none>            3893.1 3905.1
## arsenic       1   4047.1 4057.1 153.959 < 2.2e-16 ***
## assoc         1   3896.2 3906.2   3.027 0.0819086 .
## dist100:educ  1   3907.8 3917.8  14.702 0.0001259 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
drop1(fit_b, test="LRT")
```

```
## Single term deletions
##
## Model:
## cbind(switch, 1 - switch) ~ dist100 + arsenic + educ + assoc +
##     dist100:educ
##              Df Deviance    AIC     LRT  Pr(>Chi)
## <none>            3893.1 3905.1
## arsenic       1   4047.1 4057.1 153.959 < 2.2e-16 ***
## assoc         1   3896.2 3906.2   3.027 0.0819086 .
## dist100:educ  1   3907.8 3917.8  14.702 0.0001259 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
fit_c <-update(fit_b, ~.-assoc)
```

```
summary(fit_c)
```

```
##
## Call:
## glm(formula = cbind(switch, 1 - switch) ~ dist100 + arsenic +
##     educ + dist100:educ, family = binomial)
##
## 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 ***
## arsenic       0.4805993  0.0419866  11.446  < 2e-16 ***
## educ         -0.0020771  0.0152548  -0.136 0.891693
## 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
```

```
anova(fit_c)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: cbind(switch, 1 - switch)
##
## Terms added sequentially (first to last)
##
##
##              Df Deviance Resid. Df Resid. Dev
```

```
## NULL                         3019     4118.1
## dist100       1   41.861      3018     4076.2
## arsenic       1  145.570      3017     3930.7
## educ          1   20.235      3016     3910.4
## dist100:educ  1   14.283      3015     3896.2
```

```
drop1(fit_c, test="Chisq")
```

```
## Single term deletions
##
## Model:
## cbind(switch, 1 - switch) ~ dist100 + arsenic + educ + dist100:educ
##              Df Deviance    AIC     LRT  Pr(>Chi)
## <none>            3896.2 3906.2
## arsenic       1   4051.1 4059.1 154.948 < 2.2e-16 ***
## dist100:educ  1   3910.4 3918.4  14.283 0.0001573 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
drop1(fit_c, test="LRT")
```

```
## Single term deletions
##
## Model:
## cbind(switch, 1 - switch) ~ dist100 + arsenic + educ + dist100:educ
##              Df Deviance    AIC     LRT  Pr(>Chi)
## <none>            3896.2 3906.2
## arsenic       1   4051.1 4059.1 154.948 < 2.2e-16 ***
## dist100:educ  1   3910.4 3918.4  14.283 0.0001573 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
fit_d <- update(fit_c, ~.-educ-dist100:educ)
```

```
summary(fit_d)
```

```
##
## Call:
## glm(formula = cbind(switch, 1 - switch) ~ dist100 + arsenic,
##     family = binomial)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6351  -1.2139   0.7786   1.0702   1.7085
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.002749   0.079448   0.035    0.972
## dist100     -0.896644   0.104347  -8.593   <2e-16 ***
## arsenic      0.460775   0.041385  11.134   <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: 3930.7  on 3017  degrees of freedom
## AIC: 3936.7
##
## Number of Fisher Scoring iterations: 4
```

```
anova(fit_d)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: cbind(switch, 1 - switch)
##
## Terms added sequentially (first to last)
##
##
##         Df Deviance Resid. Df Resid. Dev
## NULL                     3019     4118.1
## dist100  1   41.861      3018     4076.2
## arsenic  1  145.570      3017     3930.7
```

It seems that the logistic model with dist in 100 m units and level of arsenic contamination explains the variation in response (switch) most adequately. The distance has a predicted coefficient less than 0 whereas the level of arsenic contamination returns a positive coefficient, so we can say that one unit arsenic level increase brings approximately a multiplicative $e^{0.461}$ level of increase in odds and every 100m the well gets further the odds of switching the well increase by a multiplicative factor of $e^{-0.89}$.

It follows that the probability of switching well decreases once the well gets further but it increases as the well gets to have a higher level of arsenic. We can notice that since $|-0.89| > |0.46|$ distance plays a bigger role for people when it comes to deciding whether to switch the well or not.

Model c (fit_c) with education and and the interaction between the distance and the education gives a better fit actually, but I decided to choose model d thinking of the intepretation's sake (What significant effect would a college degree have on motivating people to walk more to choose the well for their households? If you were to get educated enough in such a town, would it not mean the household is wealthy enough to purchase water? Furthermore, if education indeed a predictor that has an effect on $\beta_{dist100}$, why would eduation itself would be not significant predictor?)

(c) We use the functions that were provided from class:

```
# Computing sensitivity and specificity
pred1 <- predict(fit1, type = "response")
pred_d <- predict(fit_d, type = "response")
pred_c <- predict(fit_c, type = "response")#For comparison

sens.spec <- function(y,pred,p=0.5){
    y.hat <- as.numeric(pred>p)
    tmp <- cbind(y,y.hat)
    I1 <- as.numeric(y==1)
    I2 <- as.numeric(y.hat==1)
    a <- sum(I1*I2)
    b <- sum(I1*(1-I2))
    c <- sum((1-I1)*I2)
    d <- sum((1-I1)*(1-I2))
    sens <- a/(a+b)
    spec <- d/(c+d)
    observed <- factor(c("Y=1","Y=1","Y=0","Y=0"),levels=c("Y=1","Y=0"))
```

```
    fitted <- factor(c("Y.hat=1","Y.hat=0","Y.hat=1","Y.hat=0"),levels=c("Y.hat=1","Y.hat=0"))
    print(xtabs(c(a,b,c,d)~observed+fitted))
    return(list(c(sensitivity=sens,specificity=spec)))
}

sens.spec(switch,pred1)
```

```
##         fitted
## observed Y.hat=1 Y.hat=0
##      Y=1    1604     133
##      Y=0    1089     194
```

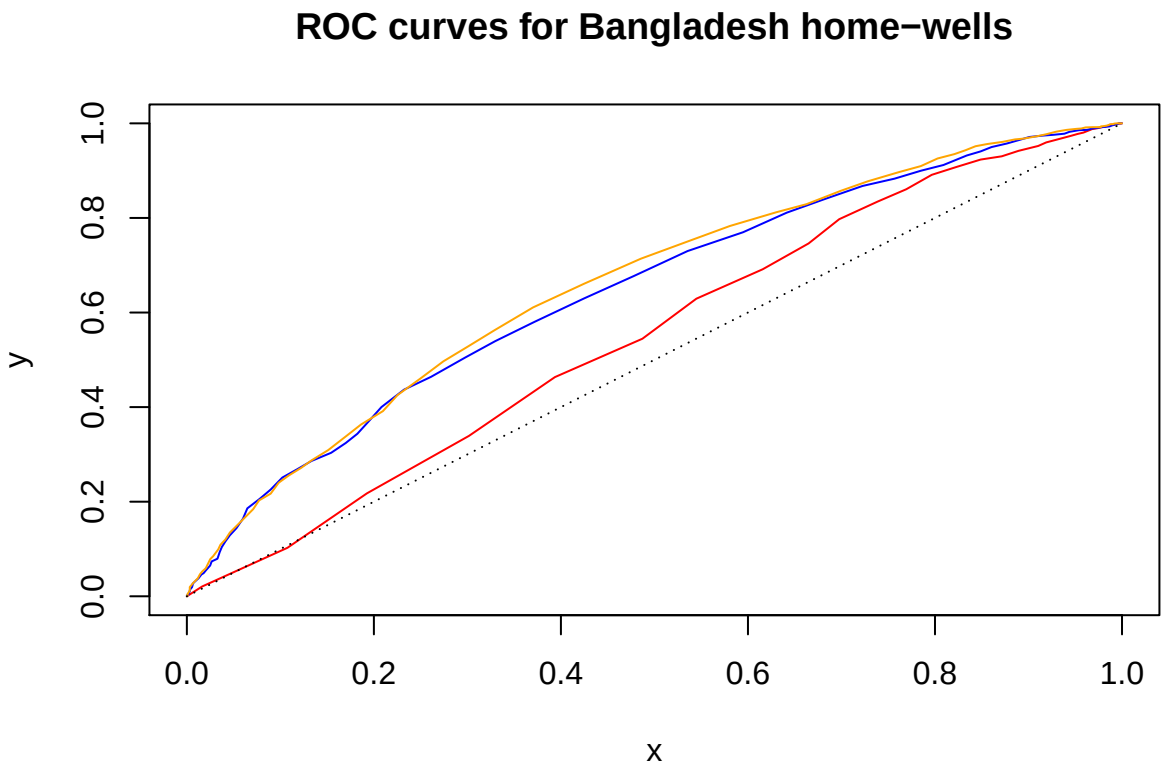
```
## [[1]]
## sensitivity specificity
##   0.9234312   0.1512081
```

```
sens.spec(switch,pred_d)
```

```
##         fitted
## observed Y.hat=1 Y.hat=0
##      Y=1    1456     281
##      Y=0     872     411
```

```
## [[1]]
## sensitivity specificity
##   0.8382268   0.3203429
```

```
# Computing the ROC curve

roc.curve <- function(y,pred){
    p <- seq(from=0,to=1,by=0.01)
    out <- matrix(ncol=2,nrow=length(p))
    for(i in 1:length(p)){
    y.hat <- as.numeric(pred>p[i])
    tmp <- cbind(y,y.hat)
    I1 <- as.numeric(y==1)
    I2 <- as.numeric(y.hat==1)
    a <- sum(I1*I2)
    b <- sum(I1*(1-I2))
    c <- sum((1-I1)*I2)
    d <- sum((1-I1)*(1-I2))
    sens <- a/(a+b)
    spec <- d/(c+d)
    out[i,1] <- 1-spec
    out[i,2] <- sens
    }
    out
}
roc.f1 <- roc.curve(switch,pred1)
roc.f_d <- roc.curve(switch,pred_d)
roc.f_c <- roc.curve(switch, pred_c)

plot(roc.f1,type="l",xlab="x",ylab="y",main="ROC curves for Bangladesh home-wells",col="red")
lines(roc.f_d,col="blue")
lines(roc.f_c,col = "orange")
lines(c(0,1),c(0,1),lty=3)
```



We can conclude that the model with arsenic level and distance is a certainly better one.