STAT230 HW 7
University of California, Berkeley

Thibault Doutre, Student ID 26980469

March 14, 2016

1 Lab 11

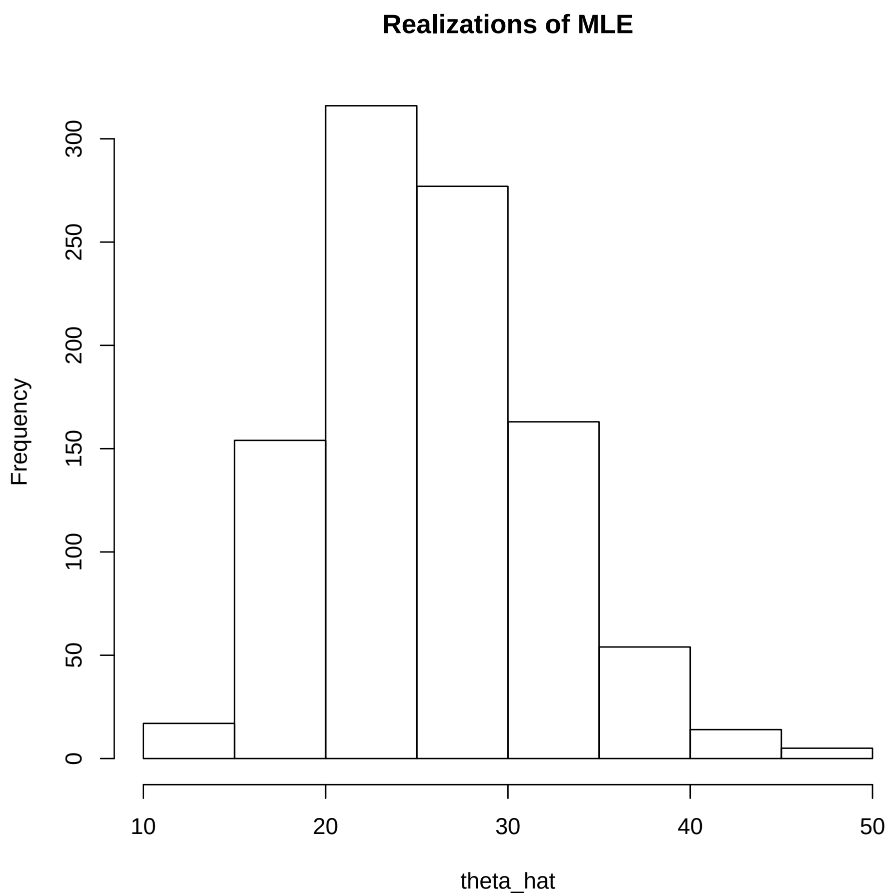
```
# Compute log likelihood
log_likelihood_aux = function(theta,X){
  n = lengthgth(X)
  n*log(theta) - 2*sum(log(theta+X))
}
log_likelihood = Vectorize(log_likelihood_aux)

# Change variable theta<-exp(phi)
log_likelihood_phi_aux = function(phi,X){
  n = length(X)
  n*phi - 2*sum(log(exp(phi)+X))
}

MLE = function (X){
  f = function(phi){-log_likelihood_phi_aux(phi,X)}
  opt = optim(0,f,
              method="Brent",
              lower = 0,
              upper = 10)
  theta_hat = exp(opt$par)
  theta_hat
}
```

1.1 Generate uniform RV

```
set.seed(1)
theta_hat_values = c()
for (i in 1:1000){
  # Generate data
  U = runif(50)
  theta = 25
  X = theta*sapply(U,function(x) x/(1-x))
  # Compute MLE
  theta_hat = MLE(X)
  theta_hat_values = c(theta_hat_values,theta_hat)
}
```

1.2 Plot histogram

```
hist(theta_hat_values,
     xlab = "theta_hat",
     main = "Realizations of MLE")
```



1.3 Mean/SD

```
mu = mean(theta_hat_values)
mu

## [1] 25.72292

sigma = sd(theta_hat_values)
sigma

## [1] 6.04452
```

```r
fisher_info = function(theta){
  1/(3*theta^2)
}
# Comparison
asympt_var = 1/sqrt(50*fisher_info(25))
sigma
```

```r
asympt_var-sigma
```

The asymptotic sd is equal to $1/\sqrt{50 I_\theta(25)}$. Therefore they shoud be equal for an infinite number of simulations, c.f. formula Example 4, Chapter 7.

1.4 Bonus

6.
The asymptotic sd should be more accurate to estimate SE because it is a limit of the sd of a n-sized sample.
7.
The asymptotic sd doubles and the fisher info is divided by 4, see formula. The standard deviation of the observed info will also double and will still tend to the asymptotic sd as n grows to infinity.

2 Lab 12

```r
data = read.table("pac01.dat")
head(data)
```

```
##    V1  V2  V3  V4  V5  V6  V7  V8  V9  V10  V11  V12  V13  V14
   V15  V16
## 1  23   1   4   8   4   2   3  37   1    2    2   40    0    1
    22    9
```

```
## 2 45   1   1   8   1   1   3  39   6    0    1   -1    1    0
     0    0
## 3 39   2   3   8   1   1   3  40   1    3    2   46    0    0
    18    2
## 4 16   1   3   8   7   0   3  35   4    8   11   -1    6    1
    24   15
## 5 53   2   1   8   5   0   1  41   1    1    2   54    0    0
    17    2
## 6 42   1   1   8   7   0   1  42   1    1    2   50    0    0
    10    1
##       V17    V18  V19  V20   V21  V22  V23  V24      V25
   V26  V27
## 1  15002  15002    1    0   445    1    1    0  216162
   330502   91
## 2   2200  28300    1    0  1451    1    1    2  259495
   324334   91
## 3  26100  28300    1    0  1451    1    2    1  205681
   324334   91
## 4      0  28300    1    0  1451    1    3    0  218787
   356936   91
## 5  23000  23000    1    0  4356    1    1    0  200660
   347911   91
## 6  44297  44297    1    0  4357    1    1    0  206279
   351372   91
```

```
names(data)=c("AGE",
              "SEX",
              "RACE",
              "ETHNICITY",
              "MARITAL",
              "NUMKIDS",
              "FAMPERS",
              "EDLEVEL",
              "LABSTAT",
              "CLASSWORK",
              "FULLPART",
              "HOURS",
              "WHYNOTWORK",
              "INSCHOOL",
```

```
             "INDUSTRY",
             "OCCUPATION",
             "PINCOME",
             "INCFAM",
             "CITIZEN",
             "IMMIGYR",
             "HHSEQNUM",
             "FSEQNUM",
             "PERSCODE",
             "SPOUCODE",
             "FINALWGT",
             "MARCHWGT",
             "STATE")

head(data)
```

```
##   AGE SEX RACE ETHNICITY MARITAL NUMKIDS FAMPERS
   EDLEVEL
## 1  23   1    4         8       4       2       3
        37
## 2  45   1    1         8       1       1       3
        39
## 3  39   2    3         8       1       1       3
        40
## 4  16   1    3         8       7       0       3
        35
## 5  53   2    1         8       5       0       1
        41
## 6  42   1    1         8       7       0       1
        42
##   LABSTAT CLASSWORK FULLPART HOURS WHYNOTWORK
   INSCHOOL
## 1       1         2        2    40          0
          1
## 2       6         0        1    -1          1
          0
## 3       1         3        2    46          0
          0
## 4       4         8       11    -1          6
```

```
          1
## 5       1         1        2    54          0
          0
## 6       1         1        2    50          0
          0
##   INDUSTRY OCCUPATION PINCOME INCFAM CITIZEN
   IMMIGYR
## 1       22          9   15002  15002        1
         0
## 2        0          0    2200  28300        1
         0
## 3       18          2   26100  28300        1
         0
## 4       24         15       0  28300        1
         0
## 5       17          2   23000  23000        1
         0
## 6       10          1   44297  44297        1
         0
##   HHSEQNUM FSEQNUM PERSCODE SPOUCODE FINALWGT
   MARCHWGT
## 1      445       1        1        0   216162
   330502
## 2     1451       1        1        2   259495
   324334
## 3     1451       1        2        1   205681
   324334
## 4     1451       1        3        0   218787
   356936
## 5     4356       1        1        0   200660
   347911
## 6     4357       1        1        0   206279
   351372
##   STATE
## 1    91
## 2    91
## 3    91
## 4    91
## 5    91
```

```
## 6     91
```

```
subset = data[,c("LABSTAT","AGE","SEX","RACE","EDLEVEL")]
head(subset)
```

```
##   LABSTAT AGE SEX RACE EDLEVEL
## 1       1  23   1    4      37
## 2       6  45   1    1      39
## 3       1  39   2    3      40
## 4       4  16   1    3      35
## 5       1  53   2    1      41
## 6       1  42   1    1      42
```

2.1

I chose to only define binary random variables in order to avoid putting more weight on some factors. The baseline individual in the model, choose a person who is male, non- white, age 1619, and did not graduate from high school. It corresponds to zero values variables in the model.

```
# Split age
split_age1 = rep(0,length(data$AGE))
split_age2 = rep(0,length(data$AGE))
split_age3 = rep(0,length(data$AGE))

split_age1[data$AGE>=20 & data$AGE<=39] = 1#"2039"
split_age2[data$AGE>=40 & data$AGE<=64] = 1#"40-64"
split_age3[data$AGE>=65] = 1#"65+"

# Split Race
split_race = rep(0,length(data$RACE))
split_race[data$RACE!=1] = 1#"white"

# Split Education level
split_edlevel1 = rep(0,length(data$EDLEVEL))
split_edlevel2 = rep(0,length(data$EDLEVEL))
split_edlevel1[data$EDLEVEL==39] = 1#"HS "
```

```
split_edlevel2[data$EDLEVEL>=40] = 1#"HS+"

# Split SEX
split_sex = rep(NA,length(data$SEX))
split_sex[data$SEX==1]=0#"M"
split_sex[data$SEX==2]=1#"F"

features = data.frame(LABSTAT = as.numeric(data$LABSTAT==1),
                      SEX = as.numeric(split_sex),
                      AGE1 = as.numeric(split_age1),
                      AGE2 = as.numeric(split_age2),
                      AGE3 = as.numeric(split_age3),
                      RACE = as.numeric(split_race),
                      EDLEVEL1 = as.numeric(split_edlevel1),
                      EDLEVEL2 = as.numeric(split_edlevel2))
head(features)
```

```
##   LABSTAT SEX AGE1 AGE2 AGE3 RACE EDLEVEL1
   EDLEVEL2
## 1       1   0    1    0    0    1        0
         0
## 2       0   0    0    1    0    0        1
         0
## 3       1   1    1    0    0    1        0
         1
## 4       0   0    0    0    0    1        0
         0
## 5       1   1    0    1    0    0        0
         1
## 6       1   0    0    1    0    0        0
         1
```

```
any(is.na(features))
```

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

```
## # 1.
design = features[,-1]
design$Intercept = as.numeric(1)
head(design)
```

```
##    SEX  AGE1  AGE2  AGE3  RACE  EDLEVEL1  EDLEVEL2
   Intercept
## 1    0     1     0     0     1         0         0
           1
## 2    0     0     1     0     0         1         0
           1
## 3    1     1     0     0     1         0         1
           1
## 4    0     0     0     0     1         0         0
           1
## 5    1     0     1     0     0         0         1
           1
## 6    0     0     1     0     0         0         1
           1
```

```
names_features = names(design)
design = as.matrix(design)


# Size of design matrix
dim(design)
```

```
## [1] 13803      8
```

```
Y=features[,1]
```

2.2

I use pracma library in order to find the maximum likelihood estimator.

```
library(pracma)

Eta = function(x){
  sapply(x,function(x) 1/(1+exp(-x)))
}

# Negative log likelihood
loss = function(beta){
```

```
  x = design %*% beta
  -(sum(Y*log(Eta(x))+(1-Y)*log(1-Eta(x))))
}

beta0 = as.matrix(rep(0,dim(design)[2]))
loss(beta0)
```

```
## [1] 9567.511
```

```
mini = fminsearch(loss,beta0)
beta_hat = mini$xval
names(beta_hat) <- names_features
beta_hat
```

```
##         SEX        AGE1        AGE2        AGE3
        RACE
## -0.7176359  1.3289470  1.2365739 -1.6667350
   -0.1826282
##   EDLEVEL1   EDLEVEL2  Intercept
##  0.7369803  0.9931386 -0.5787865
```

We can also fit the logit with the glm function directly:

```
## Alternative
glm.fit <- glm(LABSTAT ~. , data = features, family = "binomial")
summary(glm.fit)
```

```
##
## Call:
## glm(formula = LABSTAT ~ ., family = "binomial",
   data = features)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9517  -0.8754   0.5925   0.8144   2.5251
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept) -0.57880    0.06953  -8.324  < 2e-16
   ***
## SEX         -0.71753    0.04088 -17.551  < 2e-16
   ***
## AGE1         1.32888    0.07487  17.750  < 2e-16
   ***
## AGE2         1.23652    0.07563  16.350  < 2e-16
   ***
## AGE3        -1.66690    0.10143 -16.434  < 2e-16
   ***
## RACE        -0.18265    0.05034  -3.628 0.000285
   ***
## EDLEVEL1     0.73705    0.05657  13.029  < 2e-16
   ***
## EDLEVEL2     0.99318    0.05067  19.601  < 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: 18319  on 13802  degrees of
   freedom
## Residual deviance: 14879  on 13795  degrees of
   freedom
## AIC: 14895
##
## Number of Fisher Scoring iterations: 4
```

2.3

```
# Standard errors
summary(glm.fit)$coefficients[,2]
```

```
## (Intercept)          SEX         AGE1         AGE2 
        AGE3 
##   0.06953246   0.04088345   0.07486672   0.07562729 
   0.10143142 
##         RACE     EDLEVEL1     EDLEVEL2 
##   0.05034075   0.05657165   0.05066913
```

2.4

When looking at the sign of the coefficients we can first say that employment is positively correlated with having been to high school or above, since the baseline is no high school and the coefficients for EDLEVEL1 and EDLEVEL2 are positive. Similarly, we can also conclude that beign either a woman or non-white has a bad impact on employment. Moreover, it is important to notice that the p-values are all very small, which shows the importance of all the features used to predict the outcome.

2.5

First of all there is no way to correctly quantify the education, so we cannot use a real valued variable and perform regression. As a matter of fact we have categorical variables. We use dummy variables in order to avoid giving more weight to some variables: the way of assigning factors matters in the regression in the sense that the linear model gives more importance to categories with a bigger factor.

2.6

The fact that most of women give birth may impact their employment since the employers know that they might not be able to work for a while. Moreover, the SEX variable might be correlated with the edication level for example. It is known that women have less access to education that men for some reasons and it might impact their employment. LABSTAT codes more than 4 are relevant because women aremore likely not to work than men do. Therefore they might not be looking for a job, which is generally not the case for men.