STA303 Assignment 1

Jin Long Cao

7/24/2021

Question 1

Given $(y_1, y_2, \cdots, y_k) \sim \text{Multinomial}(n, \pi_1, \pi_2, \cdots, \pi_k)$.
The multinomial p.m.f. is $P(y_1, y_2, ..., y_k) = \frac{n!\pi_1^{y_1}\pi_2^{y_2}\cdots\pi_k^{y_k}}{y_1!y_2!\cdots y_k!}$

a)

$$\begin{aligned}
M_y(t_1, t_2, \cdots, t_k) &= E(e^{y_1t_1+y_2t_2+\cdots+y_kt_k}) \\
&= \sum_{y_1}\sum_{y_2}\cdots\sum_{y_k} e^{y_1t_1+y_2t_2+\cdots+y_kt_k}\frac{n!\pi_1^{y_1}\pi_2^{y_2}\cdots\pi_k^{y_k}}{y_1!y_2!\cdots y_k!} \\
&= \sum_{y_1}\sum_{y_2}\cdots\sum_{y_k} \frac{n!(\pi_1e^{t_1})^{y_1}(\pi_2e^{t_2})^{y_2}\cdots(\pi_ke^{t_k})^{y_k}}{y_1!y_2!\cdots y_k!} \\
&= (\pi_1e^{t_1} + \pi_2e^{t_2} + \cdots + \pi_ke^{t_k})^n
\end{aligned}$$

Therefore the moment-generating function (mgf) is $M_y(t) = (\pi_1e^{t_1} + \pi_2e^{t_2} + \cdots + \pi_ke^{t_k})^n = (\sum_{i=1}^k \pi_ie^{t_i})^n$

b)

The moment generating function of y_j can be calculated by setting all other variable equal to zero such that

$$\begin{aligned}
M_y(0, 0, \cdots, t_j, \cdots, 0) &= (\pi_1e^0 + \pi_2e^0 + \cdots + \pi_je^{t_j} + \cdots + \pi_ke^0)^n \\
&= ((1-\pi_j) + \pi_je^{t_j})^n \qquad (\text{Since } \sum_{i=1}^k \pi_i = 1)
\end{aligned}$$

i)

$$\begin{aligned}
E(y_j) &= [\frac{\partial}{\partial t_j}M_{y_j}(t_j)]_{t_j=0} \\
&= [n((1-\pi_j) + \pi_je^{t_j})^{n-1}\pi_je^{t_j}]_{t_j=0} \\
&= [n((1-\pi_j) + \pi_j)^{n-1}\pi_j] \\
E(y_j) &= n\pi_j
\end{aligned}$$

ii)

Similarly,

$$\begin{aligned}
E(y_j^2) &= [\frac{\partial^2}{\partial^2 t_j} M_{y_i}(t_j)]_{t_j=0} \\
&= [\frac{\partial}{\partial t_j} n((1-\pi_j)+\pi_j e^{t_j})^{n-1}\pi_j e^{t_j}]_{t_j=0} \\
&= n\pi_j[(n-1)((1-\pi_j)+\pi_j e^{t_j})^{n-2}\pi_j e^{2t_j} + ((1-\pi_j)+\pi_j e^{t_j})^{n-1}e^{t_j}]_{t_j=0} \\
&= n\pi_j[(n-1)\pi_j+1] \\
E(y_j^2) &= n^2\pi_j^2 - n\pi_j^2 + n\pi_j
\end{aligned}$$

Hence, $\text{var}(y_j) = E(y_j^2) - (E(y_j))^2 = n^2\pi_j^2 - n\pi_j^2 + n\pi_j - (n\pi_j)^2 = n\pi_j(1-\pi_j)$

iii)

The moment generating function of y_j and y_i can be calculated by setting all other variable equal to zero such that

$$\begin{aligned}
M_{y_i,y_j}(0,0,\cdots,t_i,\cdots,t_j,\cdots,0) &= (\pi_1 e^0 + \pi_2 e^0 + \cdots + \pi_i e^{t_i} + \cdots + \pi_j e^{t_j} + \cdots + \pi_k e^0)^n \\
&= ((1-\pi_j-\pi_i)+\pi_i e^{t_i}+\pi_j e^{t_j})^n \qquad (\text{Since } \sum_{i=1}^{k}\pi_i = 1)
\end{aligned}$$

Then,

$$\begin{aligned}
E(y_i y_j) &= [\frac{\partial^2}{\partial t_i \partial t_j} M_{y_i,y_j}(t_i,t_j)]_{t_i=0,t_j=0} \\
&= [\frac{\partial}{\partial t_i} n[(1-\pi_i-\pi_j)+\pi_i e^{t_i}+\pi_j e^{t_j}]^{n-1}\pi_j e^{t_j}]_{t_i=0,t_j=0} \\
&= n(n-1)[(1-\pi_i-\pi_j)+\pi_i e^{t_i}+\pi_j e^{t_j}]^{n-2}\pi_i e^{t_i}\pi_j e^{t_j}|_{t_i=0,t_j=0} \\
&= n(n-1)\pi_i\pi_j \\
E(y_i y_j) &= n^2\pi_i\pi_j - n\pi_i\pi_j
\end{aligned}$$

and, $\text{cov}(y_i y_j) = E(y_i y_j) - E(y_i)E(y_j) = n^2\pi_i\pi_j - n\pi_i\pi_j - (n\pi_i)(n\pi_j) = -n\pi_i\pi_j$

c)

$\text{Cor}(Y_i,Y_j) = \frac{\text{Cov}(Y_i,Y_j)}{\sqrt{Var(Y_i)Var(Y_j)}} = \frac{-n\pi_i\pi_j}{\sqrt{n\pi_i(1-\pi_i)n\pi_j(1-\pi_j)}} = \frac{-\pi_i\pi_j}{\sqrt{\pi_i(1-\pi_i)\pi_j(1-\pi_j)}}$

d)

If C=2, then $\pi_i + \pi_j = 1 \Rightarrow \pi_i = 1 - \pi_j$ and $\pi_j = 1 - \pi_i$. Hence,

$$\begin{aligned}
\text{Cor}(Y_1, Y_2) &= \frac{\text{Cov}(Y_1, Y_2)}{\sqrt{Var(Y_1)Var(Y_2)}} \\
&= \frac{-\pi_1\pi_2}{\sqrt{\pi_1(1-\pi_1)\pi_2(1-\pi_2)}} \\
&= \frac{-\pi_1(1-\pi_1)}{\sqrt{\pi_1(1-\pi_1)(1-\pi_1)(1-(1-\pi_1))}} \\
&= \frac{-\pi_1(1-\pi_1)}{\sqrt{\pi_1^2(1-\pi_1)^2}} \\
\text{Cor}(Y_1, Y_2) &= -1
\end{aligned}$$

Question 2

Let Y_1 and Y_2 be independent Poisson random variables with parameters μ_1 and μ_2, respectively.

$$\begin{aligned}
P(Y_1 = k | Y_1 + Y_2 = n) &= \frac{P(Y_1 = k, Y_1 + Y_2 = n)}{P(Y_1 + Y_2 = n)} \\
&= \frac{P(Y_1 = k)P(Y_1 + Y_2 = n - k)}{P(Y_1 + Y_2 = n)} \\
&= \frac{\frac{\mu_1^k}{k!}e^{-\mu_1} \cdot \frac{(\mu_1+\mu_2)^{n-k}}{(n-k)!}e^{-(\mu_1+\mu_2)}}{\frac{(\mu_1+\mu_2)^n}{n!}e^{-(\mu_1+\mu_2)}} \\
&= \frac{\mu_1^k(\mu_1+\mu_2)^{n-k}n!}{k!(n-k)!(\mu_1+\mu_2)^n} \\
&= \binom{n}{k}\left(\frac{\mu_1}{\mu_1+\mu_2}\right)^k\left(\frac{\mu_2}{\mu_1+\mu_2}\right)^{n-k}
\end{aligned}$$

Question 3

$$P(\hat{\pi} - z_{\frac{\alpha}{2}}\sqrt{\frac{\hat{\pi}(1-\hat{\pi})}{n}} \leq \pi \leq \hat{\pi} + z_{\frac{\alpha}{2}}\sqrt{\frac{\hat{\pi}(1-\hat{\pi})}{n}}) = 0.95 \qquad \text{is the Wald Confidence Interval}$$

a)

$\hat{\pi} = \frac{27}{30} = 0.9$
$z_{\frac{\alpha}{2}} = 1.96$
$n = 30$
Then the 95% Wald Confidence Interval is $(\hat{\pi} - z_{\frac{\alpha}{2}}\sqrt{\frac{\hat{\pi}(1-\hat{\pi})}{n}}, \hat{\pi} + z_{\frac{\alpha}{2}}\sqrt{\frac{\hat{\pi}(1-\hat{\pi})}{n}}) = (0.79264, 1.00735)$

```
nprime <- 30 + (qnorm(1-0.05/2))^2
w1 <- 30/nprime
w2 <- ((qnorm(1-0.05/2))^2)/nprime
midpoint <- 0.9*w1+0.5*w2

L_score <- midpoint - qnorm(1-0.05/2)*sqrt((1/nprime)*(0.9*(1-0.9)*w1+0.25*w2))
```

```r
U_score <- midpoint + qnorm(1-0.05/2)*sqrt((1/nprime)*(0.9*(1-0.9)*w1+0.25*w2))

L_score
```

```
## [1] 0.7437892
```

```r
U_score
```

```
## [1] 0.9654001
```

Therefore, the 95% Score Confidence Interval is $(0.74378, 0.96540)$

b)

```r
set.seed(1005411252) # set seed for productivity
N <- 100000
pie <- 0.9 ## pi is already defined in R thus we use "pie"/ the probability
n <- 15
alph <- 0.05 # For 95% Confidence Interval
y <- rbinom(N, n, pie) # Simulating 100,000 observation of Y
pihat <- y/n


## Wald Confidence Interval ##
L.wald <- pihat - qnorm(1-alph/2)*sqrt((pihat*(1-pihat))/n)
U.wald <- pihat + qnorm(1-alph/2)*sqrt((pihat*(1-pihat))/n)
pi_in_wald_CI <- (pie > L.wald)*(pie < U.wald)
observedConfLevel_WaldCI <- mean(pi_in_wald_CI)
observedConfLevel_WaldCI
```

```
## [1] 0.79122
```

```r
## Score Confidence Interval ##
nprime <- n + (qnorm(1-alph/2))^2
w1 <- n/nprime
w2 <- ((qnorm(1-alph/2))^2)/nprime
midpoint <- pihat*w1+0.5*w2
L.score <- midpoint - qnorm(1-alph/2)*sqrt((1/nprime)*(pihat*(1-pihat)*w1+0.25*w2))
U.score <- midpoint + qnorm(1-alph/2)*sqrt((1/nprime)*(pihat*(1-pihat)*w1+0.25*w2))
pi_in_score_CI <- (pie >  L.score)*(pie <  U.score)
observedConfLevel_scoreCI <- mean(pi_in_score_CI)
observedConfLevel_scoreCI
```

```
## [1] 0.9444
```

The proportion of the 95% Wald and Score confidence interval (with 100,000 observation and 0.9 probability of success of each trial) is 0.79122 and 0.9444, respectively.

This value means that out of 100,000 simulated observations, 79.122% of the observation was within the 95% **Wald** confidence interval. A noted disadvantage for the Wald CI is that it perform poorly when the sample size is small. A sample size of 100,000 is not small but a larger sample size may lead to a proportion closer to 95%. But this proportion value would unlikely be above 95%. Therefore, having a 94% **score** CI is quite impressive. Hence, the score CI is more reliable.

Question 4

a)

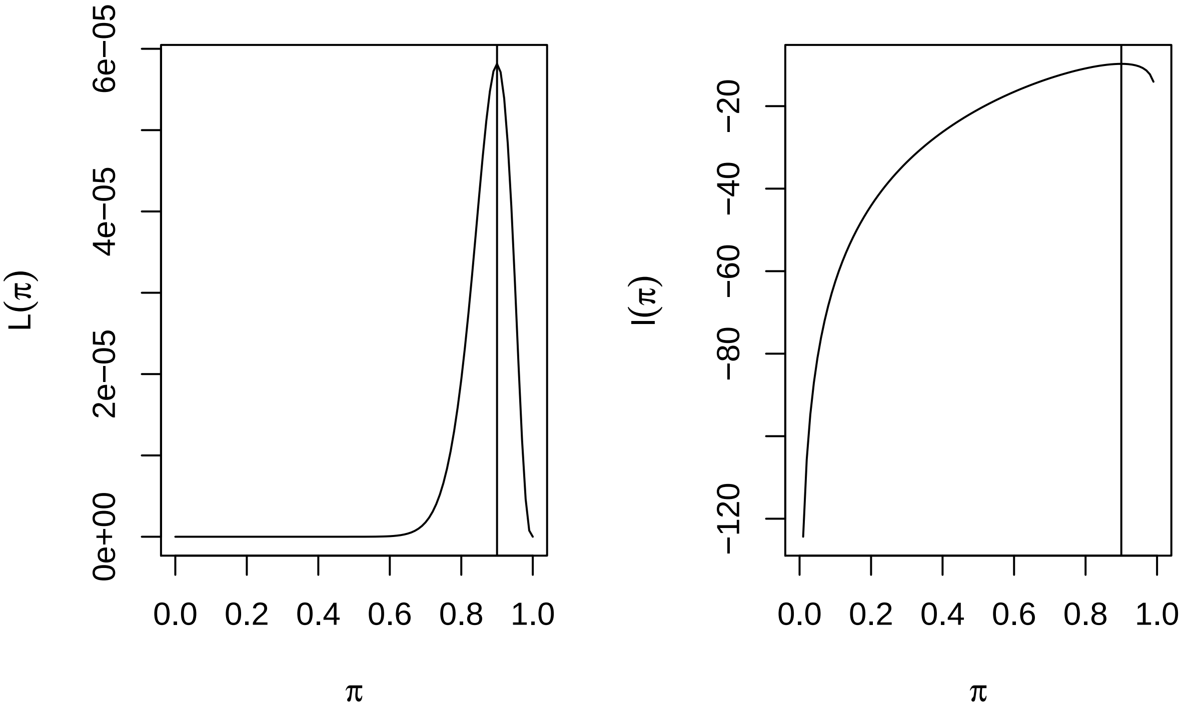
The likelihood function for the binomial is $L(\pi) = \pi^y(1-\pi)^{n-y} = \pi^{27}(1-\pi)^3$

The log-likelihood function for binomial is $log(L(\pi)) = log(\pi^{27}(1-\pi)^3) = 27log(\pi) + 3log(1-\pi)$

b)

```
par(mfrow = c(1,2)) # display two plots in the same screen/ together
# defining likelihood as a function in R
likelihood <- function(pi) { (pi^(27))*((1-pi)^3) }
# optimizing likelihood
opt.lik <- optimize(likelihood,
                    interval=c(0, 1), maximum=TRUE)
#plotting curve
curve(likelihood, from=0, to=1, xlab=expression(pi), ylab=expression(L(
  pi)))
# plotting vertical line at maximum
abline(v = opt.lik$maximum)

## same thing with log-likelihood ##
log_likelihood <- function(pi) { 27*log(pi) + 3*log(1-pi) }
curve(log_likelihood,
      from=0, to=1, xlab=expression(pi), ylab=expression(l(pi)))
opt.log_lik <- optimize(log_likelihood,
                        interval=c(0, 1), maximum=TRUE)
abline(v = opt.log_lik$maximum)
```



The maximum likelihood estimate of π is 0.899994

```
opt.log_lik$maximum
```

```
## [1] 0.8999994
```

c)

The likelihood ratio is given by

$$\frac{L(\pi_0)}{L(\hat{\pi})} = (\frac{\pi_0}{\hat{\pi}})^y(\frac{1-\pi_0}{1-\hat{\pi}})^{n-y} = (\frac{\pi_0}{\hat{\pi}})^{27}(\frac{1-\pi_0}{1-\hat{\pi}})^3$$

Then, the test statistic is given by

$$-2log(\frac{L(\pi_0)}{L(\hat{\pi})}) = 2[27log(\frac{27}{27 \cdot 0.5}) + (30-27)log(\frac{30-27}{30-30(0.5)})] = 12.06177 > \chi^2(0.05) = 3.84$$

Since the test statistic (chi-square calculated value) is greater than the chi-square critical value, we can reject our null hypothesis at $\alpha = 0.05$

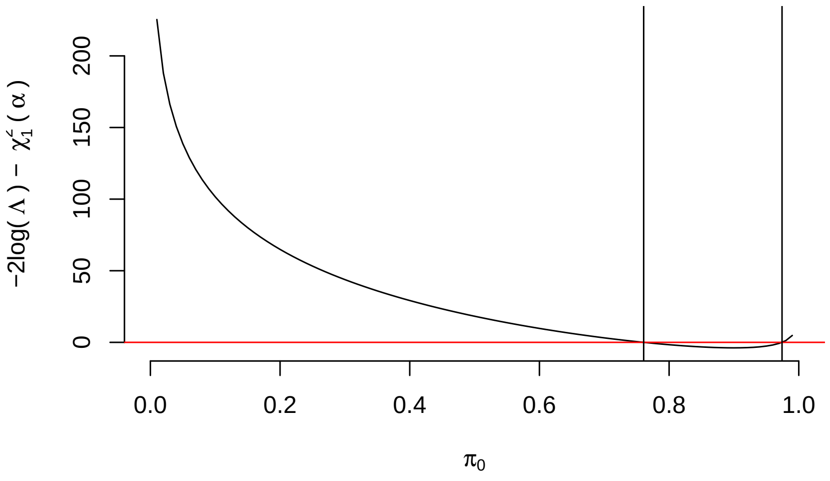
d)

The likelihood-ratio confidence interval can be calculated by

$$f(\pi_0) = 2[27log(\frac{27}{30 \cdot \pi_0}) + (3)log(\frac{3}{3-(3 \cdot \pi_0)})] \leq 3.84 = \chi^2(0.05)$$

, Where the two roots would result in the confidence interval.

```
#install.packages("rootSolve")
library(rootSolve)
n <- 30
y <- 27
phat <- y/n
alpha <- 0.05
f1 <- function(pi0) {
  2*(y*log(y/(n*pi0)) + (n-y)*log((n-y)/(n-n*pi0))) - qchisq(1-alpha,df=1)
}
par(oma = c(1,1,0,1), bty = "n")
curve(f1, from=0, to=1, xlab=expression(pi[0]),
      ylab=expression(paste("-2log(" ) ~ Lambda ~ paste( ") - ") ~ chi[1]^2 ~
                                 paste( "(" ) ~ alpha ~ paste( ")" ) ))
abline(h=0, col="red")
abline(v = uniroot.all(f=f1, interval=c(0.000001,0.999999)))
```



They intersect at $\pi_0 = (0.7607957, 0.9741571)$. Hence this is the likelihood ratio confidence interval.

```
f1 <- function(pi0) {
  2*(y*log(y/(n*pi0)) + (n-y)*log((n-y)/(n-n*pi0))) - qchisq(1-alpha,df=1)
}
uniroot.all(f=f1, interval=c(0.000001,0.999999))
```

```
## [1] 0.7607957 0.9741571
```

Question 5

```
set.seed(1005411252) # For productivity

rm(list = ls())
### Logistic Regression Simulation ##

## Simulate the data ##
n<-500 #Generating 500 random values
X1<-runif(n,-10,10)  #X1 is a normal distribution from -10 to 10
X2 <- rnorm(n, 0, 4) #X2 is a normal distribution with mean 0 and variance 4
X3 <- rbinom(n, 1, 0.7) #X3 is a bernoulli distribution with probability 0.7

Xmat <- cbind(rep(1,n), X1, X2, X3)
b0 <- -0.8
b1 <- 0.1
b2 <- 0.2
b3 <- 0.3
eta0<-b0+b1*X1+b2*X2+b3*X3
lambda = exp(eta0)
Y <- rpois(n, lambda)
```

a)

Here we simulated $Y_i \sim Poisson(\mu_i)$, Where $\mu_i = exp(\sum_j x_{ij}\beta_j)$

```
Y
```

```
##   [1]  0  0  5  0  0  1  4  1  0  5  0  0  0  0  3  0  0  0  1  5  1  0  0  3  1
##  [26]  0  0  2  1  2  0  1  3  1  1  2  1  1  0  0  0  3  0  1  1  0  4  2  3  4
##  [51]  0  2  0  1  1  0  0  3  0  2  0  0  0  0  0  0  3  1  1  0  1  0  0  0  0
##  [76]  5  2  1  0  0  6  0  0  0  0  1  4  0  0  0  1  0  2  3  0  0  0  2  1  3
## [101]  1  0  0  2  1  0  0  0  0  0  1  1  3  1  0  3  0  0  1  0  1  2  1  2  1
## [126]  0  0  0  0  0  1  1  2  1 11  1  6  1  0  3  1  2  2  1  3  0  0  0  0  0
## [151]  3  0  0  0  2  0  0  4  0  1  0  2  0  0  2  0  1  0  0  0  1  0  2  7  0
## [176]  0  0  3  0  3  0  0  1  0  5  0  0  1  0  0  0  1  0  0  0  0  2  0  0  6
## [201]  0  1  2  3  1  0  0  0  0  1  1  0  0  1  0  3  0  3  0  1  1  0  1  1  0
## [226]  0  1  0  1  1  1  0  0  2  0  0  0  0  0  0  0  1  2  0  0  1  0  1  0  2
## [251]  0  1  0  2  1 10  0  1  1  3  0  0  0  3  0  0  0  1  1  3  0  1  0  1  1
## [276]  8  0  0  1  0  3  0  0  1  0  0  0  1  1  1  0  3  2  3  0  0  0  2  0  0
## [301]  0  0  1  1  0  2  1  1  1  0  1  0  1  0  1  7  6  0  0  0  0  1  0  0  0
## [326]  0  1  0  0  0  0 10  0  1  0  0  0  0  2  0  0  1  0  0  0  0  1  0  2  2
## [351]  1  0  0  0  0  1  1  0  2  1  6  2  1  2  0  2  0  2  0  2  1  0  4  0  0
## [376]  0  0  1  1  0  0  1  0  3  0  0  0  0  1  0  1  0  0  0  0  0  0  0  0  0
## [401]  0  0  2  1  0  0  1  0  0  3  0  0  1  2  0  2  1  0  0  0  1  0  0  0  2
## [426]  0  0  1  2  0  2  3  0  0  0  0  0  1  0  3  2  2  1  0  4  0  1  1  5  0
## [451]  0  1  4  0  6  1  0  0  1  2  0  0  1  0  0  3  0  1  1  1  0  1  1  0  0
## [476]  0  0  1  0  1  0  3  0  1  1  0  3  0  0  0  0  3  5  0  0  1  0  1  1  2
```

b)

Since Y is a Poisson distribution,

$$\begin{aligned} f_Y(y;\lambda) &= \frac{e^{-\lambda}\lambda^y}{y!} \\ &= exp\{ylog\lambda - \lambda - logy!\} \\ &= exp\{y\theta - exp(\theta) - logy!\} \end{aligned}$$

$$b(\theta) = exp(\theta), \qquad b'(\theta) = exp(\theta) = \lambda, \qquad b''(\theta) = exp(\theta) = \lambda$$

Iteratively Reweighted Least Squares (IRLS) algorithm:

1. Set all $\hat{\beta_j}^{(0)} = 0$ for all j = 0, 1, 2, ..., q
2. Calculate $\hat{\mu_i}^{(0)} = \hat{\lambda_i}^{(0)} = exp(\sum_j x_{ij}\beta_j)$
3. $W^{(1)} = \text{diag}(exp(\lambda_1), exp(\lambda_2), ..., exp(\lambda_n))$.Here $\lambda_i = \lambda_i^{(0)}$
4. For i = 1, 2,..., n, set $z_i^{(1)} = \sum_j x_{ij}\beta_j + (W^{(0)})^{-1}(y_i - \hat{\pi_i}^{(0)})$
5. Estimate $\hat{\beta}^{(1)} = (X^T W^{(1)} X)^{-1} X^T W^{(1)} z^{(1)}$
6. Repeat steps 2-5 to get $W^{(2)}, z_i^{(2)}, \hat{\beta}^{(2)}$
7. Repeat 2-6 until $\hat{\pi}_i^{(t)}$ satisfy $|\hat{\pi}_i^{(t)} - \hat{\pi}_i^{(t-1)}| < \epsilon_\pi$

```
## WE NOW CREATE A FUNCTION FOR IRLS ##
glm.utsg <- function(Y, Xmat, tol = 1e-8){
  #Initialization
  beta <-rep(0, ncol(Xmat))
  eta <- Xmat %*% beta
  pivec <- exp(eta)
  W <- diag(as.vector(exp(eta)))
  Z <- eta + exp(-eta)*(Y-exp(eta))
  eqns<-sum(t(Xmat) %*% (Y-pivec))
  istep <- 0

  while(eqns^2 > tol){
    beta<-solve(t(Xmat) %*% W %*% Xmat) %*% t(Xmat) %*% W %*% Z
    eta <- Xmat %*% beta
    pivec <- exp(eta)
    W <- diag(as.vector(exp(eta)))
    Z <- eta + exp(-eta)*(Y-exp(eta))
    eqns<-sum(t(Xmat) %*% (Y-pivec))
    istep <- istep +1
  }
  SE <- sqrt(diag(solve(t(Xmat) %*% W %*% Xmat)))
  z = beta/SE
  res.mat <- data.frame(Estimates = beta, OR = c(1, exp(beta)[-1]),
                        SE = SE,
                        z = z,
                        p_value = ifelse(z < 0, 2*pnorm(z), 2*(1 - pnorm(z)))
  )
  rownames(res.mat)[1] <- "Intercept"
  results <- list(Table = res.mat, Iteration = istep)
  return(results)
}

## Now let’s run the function and compare it to R’s glm() ##
glm.utsg(Y = Y, Xmat = Xmat)
```

```
## $Table
##             Estimates       OR          SE          z      p_value
## Intercept -0.63110929 1.000000 0.100061543 -6.3072113 2.841079e-10
## X1         0.11086112 1.117240 0.008982496 12.3419057 0.000000e+00
## X2         0.21004167 1.233729 0.012762045 16.4583079 0.000000e+00
## X3         0.08816917 1.092173 0.105191655  0.8381765 4.019316e-01
##
## $Iteration
## [1] 5
```

```
## Run the glm() code in R ##
glm.p<-glm(Y ~ X1+X2+X3,family=poisson)
summary(glm.p)
```

```
##
## Call:
## glm(formula = Y ~ X1 + X2 + X3, family = poisson)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1028  -0.9001  -0.5077   0.4957   2.8164
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.631109   0.100059  -6.307 2.84e-10 ***
## X1           0.110861   0.008982  12.342  < 2e-16 ***
## X2           0.210042   0.012762  16.459  < 2e-16 ***
## X3           0.088169   0.105190   0.838    0.402
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 963.36  on 499  degrees of freedom
## Residual deviance: 483.05  on 496  degrees of freedom
## AIC: 1057.2
##
## Number of Fisher Scoring iterations: 5
```

The W matrix is a diagonal matrix with diagonal elements

$$w_{ii} = \frac{\lambda}{Var(Y_i)}$$