ASSIGNMENT 1

Directed Studies

Submitted to
Professor Dr. Mateen Shaikh

Submitted by
TARIQUE MAHMUD
T00685251

Logistic regression: Logistic regression is a part of generalized linear regression.

The likelihood for logistic regression:

$$P(Y|x) \sim \beta 1 + \beta 2 * x$$

Finding the liner function for the equation

$$ln[\frac{p(y|x)}{1-p(y|x)}] = \beta 1 + \beta 2 * x$$

To maximize the log-likelihood function for this equation, we need to calculate the value of $\beta 0$ and $\beta 1$.

$$\sum_{i=1}^{n} (y - ln(pi)) + (1-y) * ln(1-pi))$$

Here, pi is the sigmoid function as

$$P(Y|x) = \frac{e^{(\beta 1 + x\beta 2)}}{1+e^{(\beta 1 + x\beta 2)}}$$

The score function for logistic regression is a real-valued function for classification like yes-no. By thresholding the score, The score functions become a classifier.

If we consider a predictor variable X and the log odds of that event is Y = 1, the score function is used to estimate the probability of Y is equal to 1.

$$p(x) = \frac{1}{1+e^{-(\beta 1 + x\beta 2)}}$$

Results from the code:

The R code results have been exported when I compared the three approaches. The intercept beta 1 and the slope beta 2 are giving almost the nearest values.

	beta1	beta2	iter	Tol	Loglik
newton	**-15.2459**	**1.033589**	10	4.02E-09	-165.005
glm_default	**-15.2459**	**1.033589**	6	NA	165.0054
mle	**-15.2512**	**1.033933**	NA	########	1.033933

The code and results have been attached below in the R markdown.

Directed Studies - Assignment1

Taruque Mahmud - T00685251

9/26/2022

R Markdown

In this work, I have taken the dataset for breast cancer detection to develop a model of logistic regression and calculated the glm and mle. For this exploration, I have taken one variable which is the radius_mean as the dependent variable and the diagnosis as independent variable.

I have used optimx package to estimate the parameters for this dataset using Maximum Likelihood method.

#Loading packages and dataset

```
require(optimx)
```

```
## Loading required package: optimx
```

```
## Warning: package 'optimx' was built under R version 4.1.3
```

```
setwd("C:/All Files/TRU Study/2. Directed Studies/A1/")
data <- read.csv("data.csv")


class(data)
```

```
## [1] "data.frame"
```

```
df <- as.data.frame(data)
```

Here, In this dataset, the variables are as below: diagnosis: M = malignant, B = benign radius: distances from center to points on the perimeter

Now converting the diagnosis values to binary.

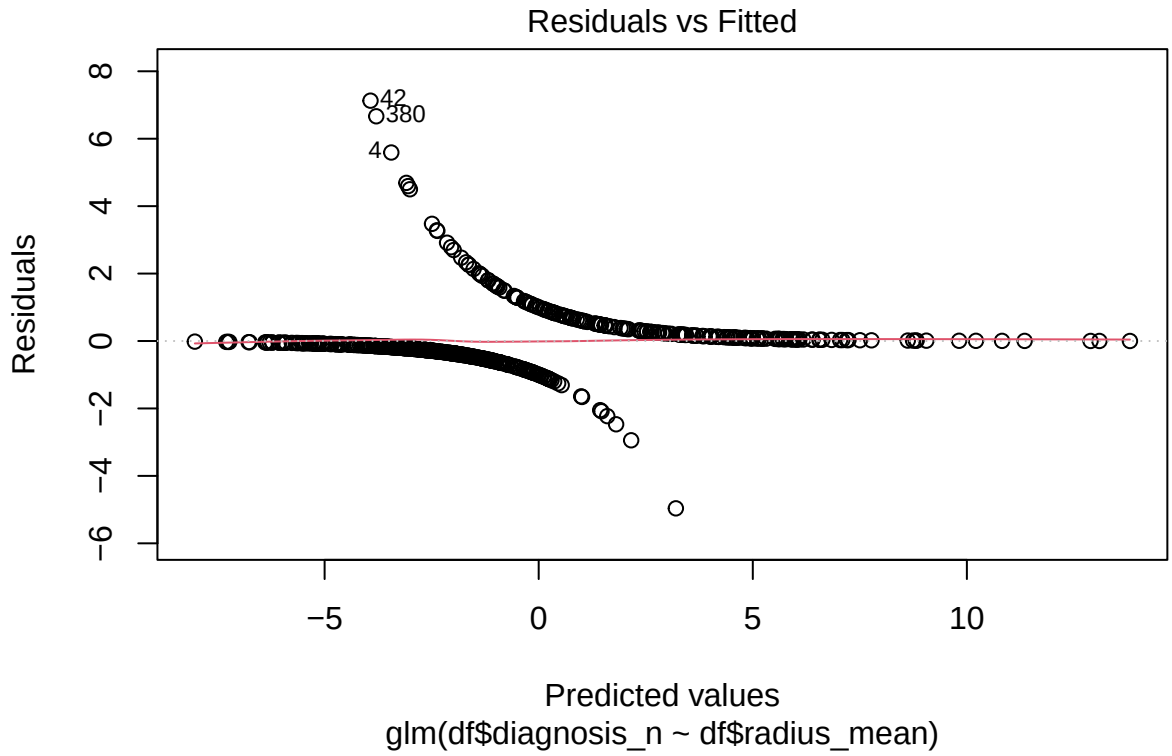
```
df$diagnosis_n <- ifelse(df$diagnosis == "M", 1, 0)
```

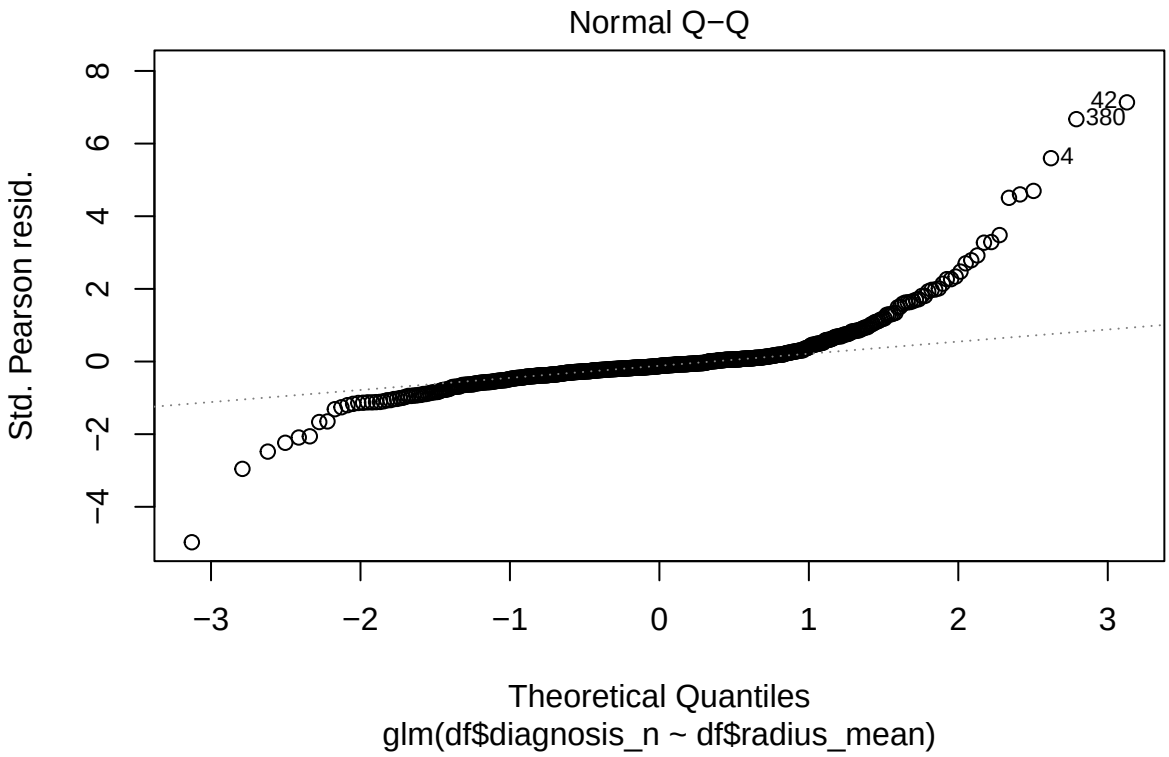
Generalized linear model

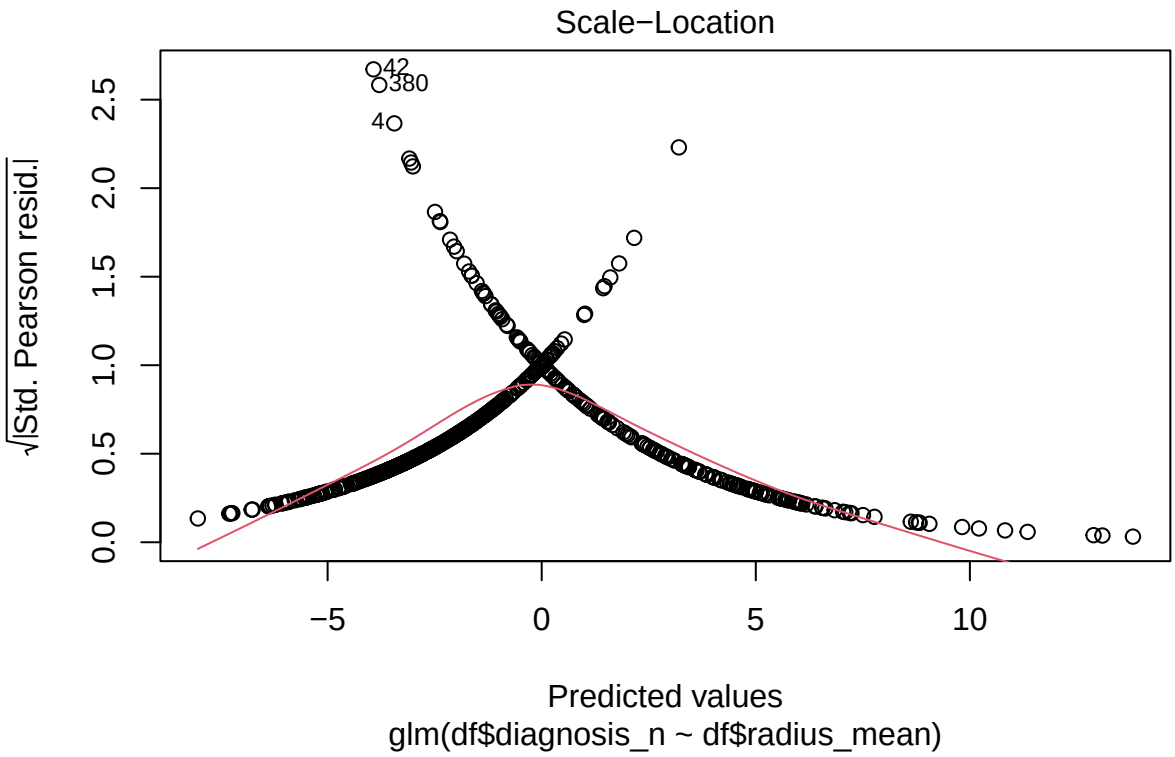
```
modelcp<- glm(df$diagnosis_n~df$radius_mean,family=binomial)
(result<-summary(modelcp))
```

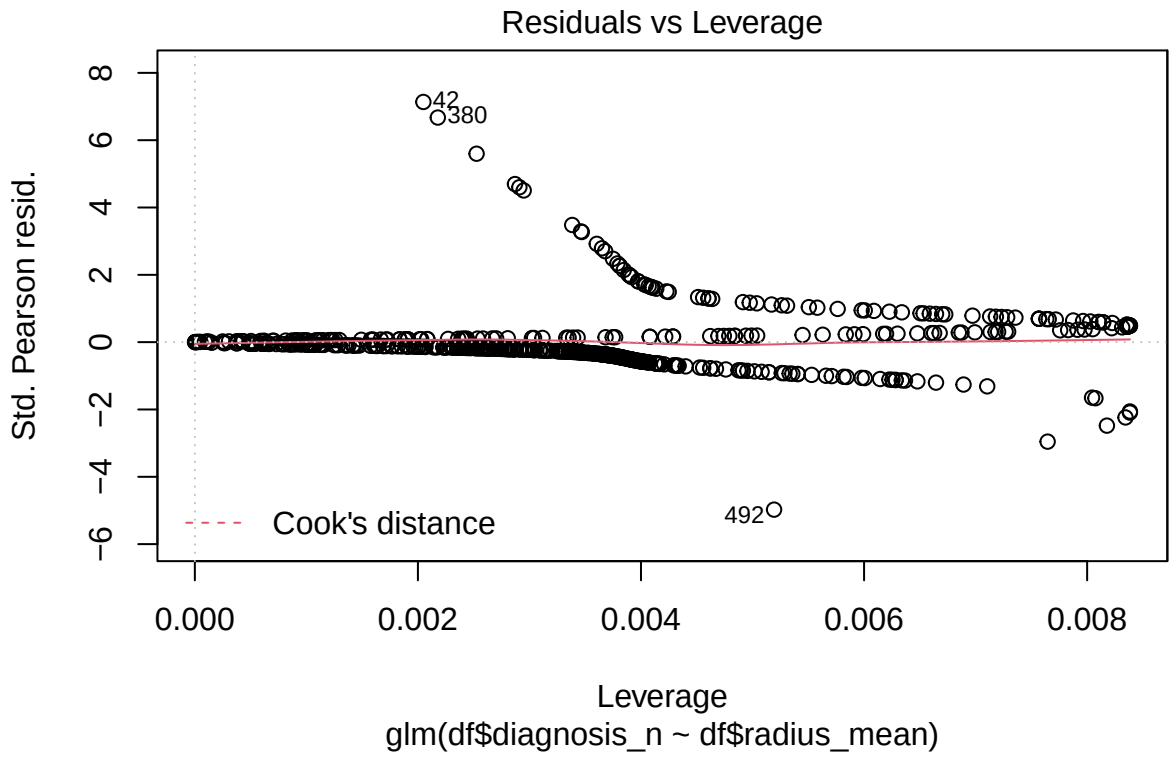
```
##
## Call:
## glm(formula = df$diagnosis_n ~ df$radius_mean, family = binomial)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5470  -0.4694  -0.1746   0.1513   2.8098
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -15.24587    1.32463  -11.51   <2e-16 ***
## df$radius_mean   1.03359    0.09311   11.10   <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: 751.44  on 568  degrees of freedom
## Residual deviance: 330.01  on 567  degrees of freedom
## AIC: 334.01
##
## Number of Fisher Scoring iterations: 6
```

```
plot(modelcp)
```




Normal Q−Q
Std. Pearson resid.
8
6
4
2
0
−2
−4
42
380
4
−3
−2
−1
0
1
2
3
Theoretical Quantiles
glm(df$diagnosis_n ~ df$radius_mean)


Scale−Location
42
380
4
√|Std. Pearson resid.|
0.0
0.5
1.0
1.5
2.0
2.5
−5
0
5
10
Predicted values
glm(df$diagnosis_n ~ df$radius_mean)



optim function MLE

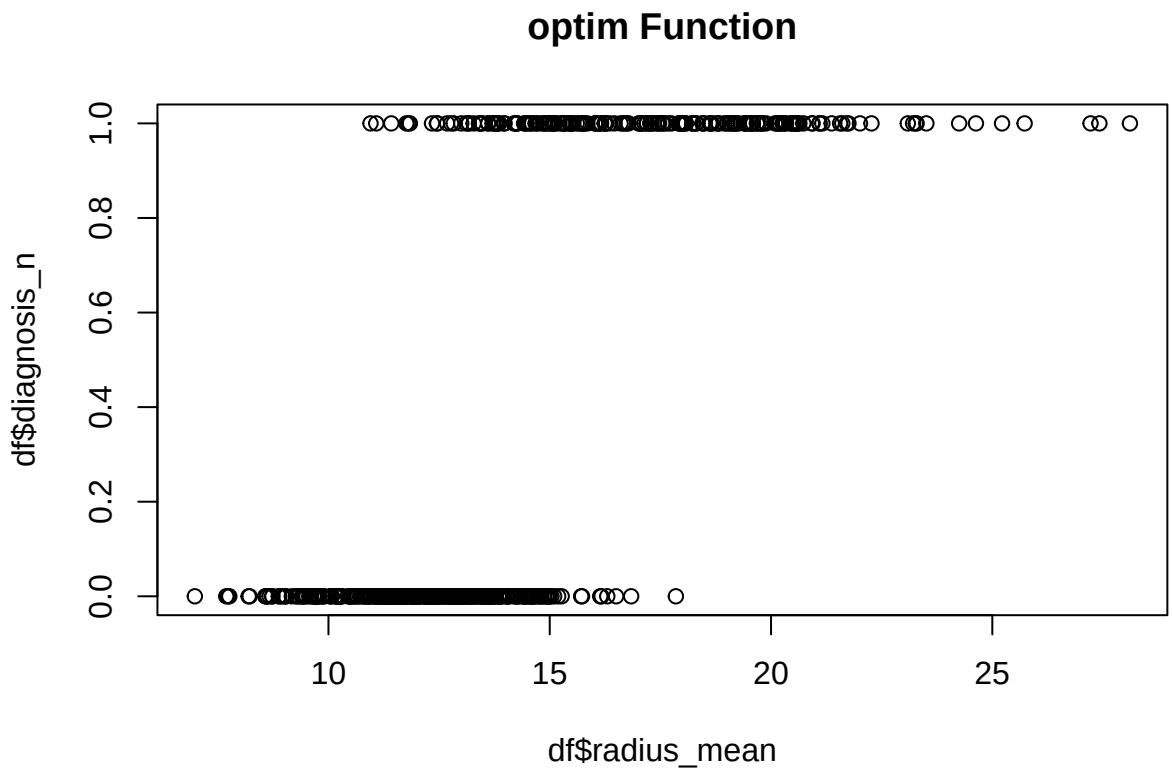
```
f1<-function(para){
  eta<-para[1]+para[2]*df$radius_mean
  p<-1/(1+exp(-eta))
  -sum(log(choose(1,df$diagnosis_n))+df$diagnosis_n*log(p)+(1-df$diagnosis_n)*log(1-p),na.rm=TRUE)
}
f2<-(optim1<-optim(c(1,1),fn=f1,hessian=TRUE))
f2
```

```
## $par
## [1] -15.251199   1.033933
##
## $value
## [1] 165.0054
##
## $counts
## function gradient
##      253       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
##
```

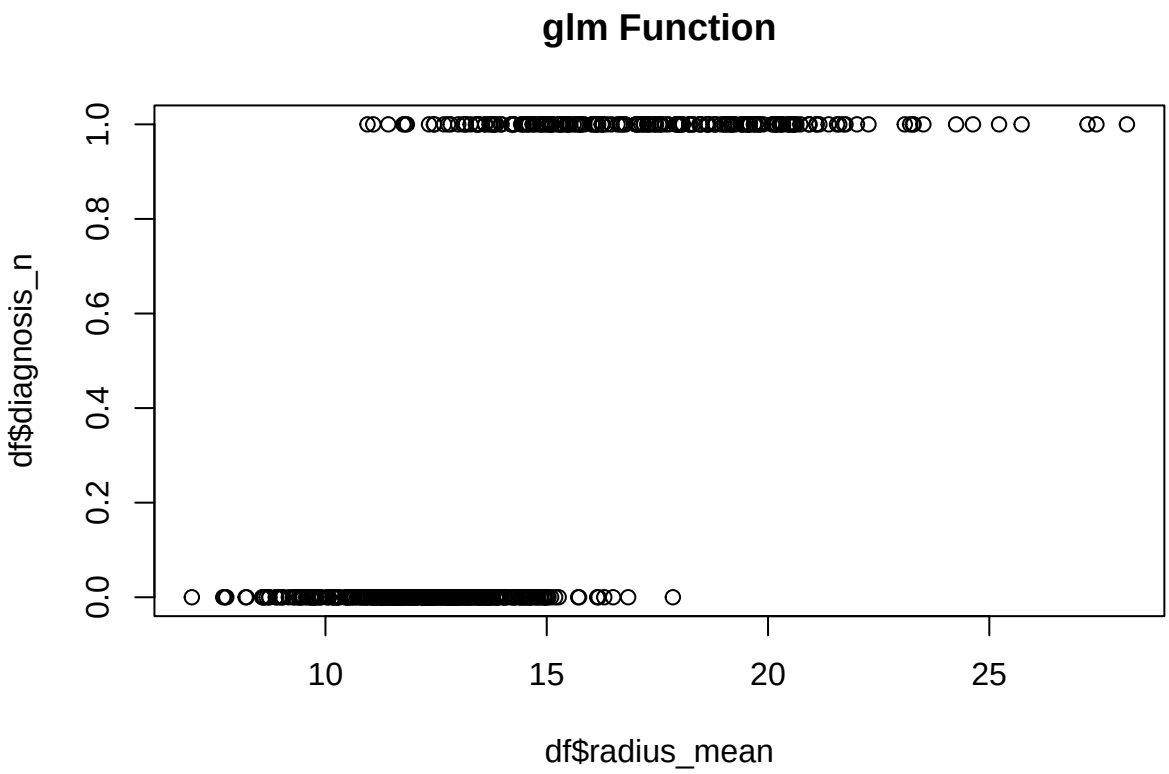
```
## $hessian
##           [,1]       [,2]
## [1,]  51.02612   721.9026
## [2,] 721.90261 10328.5118
```

Plotting the functions

```
#plot optim
plot(df$radius_mean,
     df$diagnosis_n,
     main = "optim Function")
```




```
#plot glm
plot(df$radius_mean,
     df$diagnosis_n,
     main = "glm Function")
```



newton-raphson method(NRM):

Equation:

```
X = model.matrix(modelcp)
y = modelcp$y
```

Creating function for the Newton Raphson method

```
newton <- function(
  X,
  y,
  tol  = 1e-12,
  iter = 100,
  stepsize = .5
) {

  int     = log(mean(y) / (1 - mean(y)))         # intercept
  beta    = c(int, rep(0, ncol(X) - 1))
  currtol = 1
  it = 0
  ll = 0

  while (currtol > tol && it < iter) {
    it = it +1
    ll_old = ll
```

```
    mu = plogis(X %*% beta)[,1]
    g  = crossprod(X, mu-y)               # gradient
    S  = diag(mu*(1-mu))
    H  = t(X) %*% S %*% X                 # hessian
    beta = beta - stepsize * solve(H) %*% g

    ll = sum(dbinom(y, prob = mu, size = 1, log = TRUE))
    currtol = abs(ll - ll_old)
  }

  list(
    beta = beta,
    iter = it,
    tol  = currtol,
    loglik = ll
  )
}
```

NRM results:

```
newton_result = newton(
  X = X,
  y = y,
  stepsize = .9,
  tol = 1e-8
)
```

Comparing the results with glm and newton raphson method

```
rbind(
  newton = unlist(newton_result),
  glm_default = c(
    beta = coef(modelcp),
    modelcp$iter,
    tol = NA,
    loglik = -logLik(modelcp)
  ),
  mle = c(
    beta = f2$par,
    f2$iter,
    tol = NA)
)
```

```
## Warning in rbind(newton = unlist(newton_result), glm_default = c(beta =
## coef(modelcp), : number of columns of result is not a multiple of vector length
## (arg 3)
```

```
##                  beta1    beta2 iter           tol      loglik
## newton      -15.24587 1.033589   10  4.016727e-09 -165.005422
## glm_default -15.24587 1.033589    6            NA  165.005422
## mle         -15.25120 1.033933   NA -1.525120e+01    1.033933
```

The resutl looks similar after 10 iteration in NRM.