Logistic Regression Using R

To understand Logistic Regression using R, we will use "menarche" data and "mtcars" data

In the built-in data set mtcars, (MASS, library) the data column am represents the transmission type of the automobile model (0 = automatic, 1 = manual). With the logistic regression equation, we can model the probability of a manual transmission in a vehicle based on its engine horsepower and weight data.

```
> attach(mtcars)
> head(mtcars)
                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
>
```

```
> am.glm = glm(formula=am ~ hp + wt, data=mtcars, family=binomial)
>
> ###We then wrap the test parameters inside a data frame newdata.
>
> summary(am.glm)

Call:
glm(formula = am ~ hp + wt, family = binomial, data = mtcars)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2537  -0.1568  -0.0168   0.1543   1.3449

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 18.86630    7.44356   2.535  0.01126 *
hp           0.03626    0.01773   2.044  0.04091 *
wt          -8.08348    3.06868  -2.634  0.00843 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 43.230  on 31  degrees of freedom
Residual deviance: 10.059  on 29  degrees of freedom
AIC: 16.059

Number of Fisher Scoring iterations: 8
```

```
>
> hp=c(100, 105, 110, 115, 120)
> wt=c(2.8, 2, 2.4, 2.2, 2.6)
> newdata = data.frame(hp, wt)
> pred=predict(am.glm, newdata, type="response")
> pred
        1         2         3         4         5 
0.4645913 0.9985081 0.9693524 0.9947905 0.9002437 
>
>
>
> out=cbind(hp, wt, pred)
> out
   hp  wt      pred
1 100 2.8 0.4645913
2 105 2.0 0.9985081
3 110 2.4 0.9693524
4 115 2.2 0.9947905
5 120 2.6 0.9002437
> 
```

```
> hp=mtcars$hp[1:10]
> wt=mtcars$wt[1:10]
> newdata = data.frame(hp, wt)
> pred=predict(am.glm, newdata, type="response")
> pred
          1           2           3           4           5 
0.842335537 0.404782533 0.970240822 0.041728035 0.069388122 
          6           7           8           9          10 
0.004988159 0.248041206 0.009265579 0.040998134 0.011190709 
> out=cbind(hp, wt, am, pred)
> out
    hp    wt am        pred
1  110 2.620  1 0.842335537
2  110 2.875  1 0.404782533
3   93 2.320  1 0.970240822
4  110 3.215  0 0.041728035
5  175 3.440  0 0.069388122
6  105 3.460  0 0.004988159
7  245 3.570  0 0.248041206
8   62 3.190  0 0.009265579
9   95 3.150  0 0.040998134
10 123 3.440  0 0.011190709
> 
```

```
>
> with(am.glm, null.deviance - deviance)
[1] 33.17062
> with(am.glm, df.null - df.residual)
[1] 2
> with(am.glm, pchisq(null.deviance - deviance, df.null - df.residual, lower.tail = FALSE))
[1] 6.267449e-08
>
>
```

R Console

```
> am.glm = glm(formula=am ~ hp + wt + disp + gear, data=mtcars, fami$
Warning messages:
1: glm.fit: algorithm did not converge
2: glm.fit: fitted probabilities numerically 0 or 1 occurred
> ###We then wrap the test parameters inside a data frame newdata.
> summary(am.glm)

Call:
glm(formula = am ~ hp + wt + disp + gear, family = binomial,
    data = mtcars)

Deviance Residuals:
       Min           1Q       Median           3Q          Max
-2.996e-05  -2.110e-08  -2.110e-08   2.110e-08   2.501e-05

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.248e+01  4.616e+05   0.000    1.000
hp           4.186e-01  2.075e+03   0.000    1.000
wt          -1.165e+02  1.479e+05  -0.001    0.999
disp         2.815e-01  1.674e+03   0.000    1.000
gear         7.717e+01  9.730e+04   0.001    0.999

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 4.3230e+01  on 31  degrees of freedom
Residual deviance: 2.2394e-09  on 27  degrees of freedom
AIC: 10
```

```
> hp=mtcars$hp[1:10]
> wt=mtcars$wt[1:10]
> disp=mtcars$disp[1:10]
> gear=mtcars$gear[1:10]
> am=mtcars$am[1:10]
> pred=predict(am.glm, newdata, type="response")
> pred
           1            2            3            4            5            6            7 
1.000000e+00 1.000000e+00 1.000000e+00 2.220446e-16 2.220446e-16 2.220446e-16 8.812427e-11 
           8            9           10 
2.220446e-16 4.488538e-10 2.220446e-16 
> out=cbind(hp, wt, disp, gear, am, pred)
> out
    hp    wt  disp gear am         pred
1  110 2.620 160.0    4  1 1.000000e+00
2  110 2.875 160.0    4  1 1.000000e+00
3   93 2.320 108.0    4  1 1.000000e+00
4  110 3.215 258.0    3  0 2.220446e-16
5  175 3.440 360.0    3  0 2.220446e-16
6  105 3.460 225.0    3  0 2.220446e-16
7  245 3.570 360.0    3  0 8.812427e-11
8   62 3.190 146.7    4  0 2.220446e-16
9   95 3.150 140.8    4  0 4.488538e-10
10 123 3.440 167.6    4  0 2.220446e-16
> 
```

```
>
> with(am.glm, null.deviance - deviance)
[1] 43.22973
> with(am.glm, df.null - df.residual)
[1] 4
> with(am.glm, pchisq(null.deviance - deviance, df.null - df.residual, lower.tail = FALSE))
[1] 9.27207e-09
>
>
```

```
>
> library(leaps)
> library(bestglm)
> X=data[,-9]
> Y=data[,9]
> Xy <- cbind(X, Y)
> bestglm(Xy, family = binomial)
Morgan-Tatar search since family is non-gaussian.
BIC
BICq equivalent for q in (0.00271755325461331, 0.849778894907533)
Best Model:
              Estimate Std. Error       z value  Pr(>|z|)
(Intercept)   24.97793  211732.15  0.0001179695 0.9999059
wt          -148.46570   84415.17 -0.0017587562 0.9985967
gear         105.57256   68256.49  0.0015467037 0.9987659
There were 50 or more warnings (use warnings() to see the first 50)
>
>
```

```
> wt=mtcars$wt[1:10]
> gear=mtcars$gear[1:10]
> am=mtcars$am[1:10]
> pred=predict(am.glm, newdata, type="response")
> pred
           1            2            3            4            5            6            7
1.000000e+00 1.000000e+00 1.000000e+00 2.220446e-16 2.220446e-16 2.220446e-16 8.812427e-11
           8            9           10
2.220446e-16 4.488538e-10 2.220446e-16
> out=cbind(hp, wt, am, pred)
> out
    hp    wt am         pred
1  110 2.620  1 1.000000e+00
2  110 2.875  1 1.000000e+00
3   93 2.320  1 1.000000e+00
4  110 3.215  0 2.220446e-16
5  175 3.440  0 2.220446e-16
6  105 3.460  0 2.220446e-16
7  245 3.570  0 8.812427e-11
8   62 3.190  0 2.220446e-16
9   95 3.150  0 4.488538e-10
10 123 3.440  0 2.220446e-16
> 
```

>fitted(am.glm)   ###Gives probabilities of Y=1 for all data###