

تکالیف درس روشهای چندمتغیری گسسته - دانشگاه اراک - نیم‌سال 001

تکلیف شماره 1

نام و نام خانوادگی

**محراب عتیقی**

شماره دانشجویی

**39712131125**

1. صورت سوال 1

***برای یک جدول رده‌بندی شده‌ی I\*2، مدل لوجیت زیر را درنظر گیرید:  
الف) اگر باشد، را چگونه برای ، برآورد می‌کنید. برآورد خودرا تفسیر کنید.***

***ب) تابع درستنمایی را درصورت استقلال X و Y بدست آورید و را برآورد کنید.***

پاسخ تشریحی سوال 1  
اگر قرار دهیم:  
احتمال اینکه، y=1 بشود، به شرط X=i  
*می‌دانیم که:  
در نتیجه داریم:  
اگر فرض بکنید که باشد.  
طبق رابطه‌ی ستاره و اینکه می‌دانیم که داریم:  
  
و برای زمانی که y=0 باشد داریم:*

*حال برای بدست آوردن برآورد باید اول تابع درستنمایی را نوشته و سپس برای راحتی کار از آن لگاریتم در مبنای عدد طبیعی گرفته و سپس از آن مشتق بگیریم.  
ابتدا باید فرض کنیم که هریک از خانه های جدول ما مرتبه تکرار داشته‌اند. یعنی بصورت زیر:*

|  |  |  |
| --- | --- | --- |
| *Y=1* | *Y=0* | *X Y* |
|  |  | *X=1* |
|  |  | *X=2* |
|  |  |  |
|  |  | *X=I* |

*حال با ln گیری از تابع درستنمایی داریم:  
  
از اینجا به بعد باید از تابع مشتق گرفته نسبت به پارامتر‌هایی که داریم و برابر 0 گذاشته و برآوردگرها را بدست آوریم و از آنجایی که بصورت دستی تقریب غیرممکن هستش محاسبه آن و بسیار وقت گیر است فرم کلی دستورات آنرا در نرم افزار R بیان می‌کنیم:*

rm(list=ls())  
#soal6:  
I=#number of Rows  
data<-matrix(c(N10,N11,N20,N21,...,NI0,NI1) ,nrow=I, byrow = TRUE )  
colnames(data)<-c("y=0" , "y=1")  
rownames(data)<-c("x=1" , "x=2", ... , "x=I")  
data  
betaI=0  
L<-function(alpha , c(beta1,beta2 , ... , betaI)){  
 (1 / (1 + exp(alpha-beta1))) ^ data[1,1] \*  
 (exp(alpha+beta1) / (1 + exp(alpha+beta1))) ^ data[1,2] \*  
 (1 / ( 1+ exp(alpha + beta2)))^ data[2,1] \*  
 (exp(alpha + beta2) /( 1 + exp(alpha + beta2))) ^ data[2,2]\*  
 ... \*  
 (1 / ( 1+ exp(alpha + betaI)))^ data[I,1] \*  
 (exp(alpha + betaI) /( 1 + exp(alpha + betaI))) ^ data[I,2]  
}  
  
l<-function(alpha,c(beta1,beta2 , ... , betaI)){  
 log(L(alpha,beta))  
}  
Negative.ll<-function(par){  
 alpha=par[1];beta1=par[2];...;betaI-1 =par[I-1]  
 -l(alpha,c(beta1,beta2 , ... , betaI))  
}  
  
(ml<-optim(par = c(1,1) ,Negative.ll ))  
ml.alpha=ml$par[1]  
ml.beta1=ml$par[2]  
...  
ml.betaI-1=ml$par[I-1]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| سطح صنور-> | 0.00 | 0.50 | 1.0 | 1.50 |
| تعداد آسیب دیده‌ها | 25 | 16 | 20 | 29 |
| تعداد کل | 41 | 27 | 24 | 38 |

صورت سوال 2 داده‌های جدول زیر مربوط به 130 درخت صنوبر است که در یک محیط مصنوعی برای مدت 8 سال نگهداری شده اند.

در این جدول، تعداد درخت‌هایی را که آسیب دیده‌اند، به ازای سطوح مختلف سولفور نشان می‌دهد. اگر توزیع دوجمله‌ای را برای تعداد درختان آسیب دیده فرض کنیم و مدل لوجیت خطی را برای احتمال آسیب دیدن درخت در سطح x سولفور است، برآوردهای پارامترها عبارتند از  *آ) مقادیر برازش‌یافته تحت مدل را برای خانه‌های جدول بیابید.  
ب)مانده‌های پی‌یرسون را تحت مدل بیابید.  
پ) مدلی را که فرض می‌کند غلظت سولفور تاثیری روی احتمال آسیب دیدن ندارد، برازش دهید.مقادیر برازانده شده و مانده های پی‌یرسون آن‌ها را بیابید.  
ت)کیبش را بیابید و فرض را آزمون کنید.*

**پاسخ نرم‌افزاری سوال دوم:**

**ابتدا داده‌هایمان را شبیه سازی کرده و سپس به کمک دستور glm یک مدل لوژستیک میسازیم:**

#soal8:  
y=c(rep(1 ,90 ) ,rep(0 , 40 ))  
x= c(rep(0 , 25) ,rep(0.5 ,16) , rep(1 , 20) , rep(1.5 , 29) ,  
 rep(0,16) , rep(0.5 , 11) , rep(1 ,4) , rep(1.5 , 9))  
  
table(x,y)

## y  
## x 0 1  
## 0 16 25  
## 0.5 11 16  
## 1 4 20  
## 1.5 9 29

fit1<-glm(y~x,family = binomial(link = "logit") )

حال میخواهیم که خلاصه ای از این مدل ببینیم و مقدارهای برآورد آلفا و بتا را بدست آوریم و با رنگ نارنجی نمایان خواهیم کرد:

summary(fit1)

##   
## Call:  
## glm(formula = y ~ x, family = binomial(link = "logit"))  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.7612 -1.3473 0.6904 0.8987 1.0166   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.3909 0.2836 1.378 0.1681   
## x 0.6145 0.3250 1.891 0.0587 .  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 160.48 on 129 degrees of freedom  
## Residual deviance: 156.79 on 128 degrees of freedom  
## AIC: 160.79  
##   
## Number of Fisher Scoring iterations: 4

حال میخواهیم مقادیر باقی‌مانده پی‌یرسون مربوط به مدل را بدست آوریم:

residuals.glm(fit1 , type = "pearson")

## 1 2 3 4 5 6 7   
## 0.8224710 0.8224710 0.8224710 0.8224710 0.8224710 0.8224710 0.8224710   
## 8 9 10 11 12 13 14   
## 0.8224710 0.8224710 0.8224710 0.8224710 0.8224710 0.8224710 0.8224710   
## 15 16 17 18 19 20 21   
## 0.8224710 0.8224710 0.8224710 0.8224710 0.8224710 0.8224710 0.8224710   
## 22 23 24 25 26 27 28   
## 0.8224710 0.8224710 0.8224710 0.8224710 0.7053379 0.7053379 0.7053379   
## 29 30 31 32 33 34 35   
## 0.7053379 0.7053379 0.7053379 0.7053379 0.7053379 0.7053379 0.7053379   
## 36 37 38 39 40 41 42   
## 0.7053379 0.7053379 0.7053379 0.7053379 0.7053379 0.7053379 0.6048865   
## 43 44 45 46 47 48 49   
## 0.6048865 0.6048865 0.6048865 0.6048865 0.6048865 0.6048865 0.6048865   
## 50 51 52 53 54 55 56   
## 0.6048865 0.6048865 0.6048865 0.6048865 0.6048865 0.6048865 0.6048865   
## 57 58 59 60 61 62 63   
## 0.6048865 0.6048865 0.6048865 0.6048865 0.6048865 0.5187410 0.5187410   
## 64 65 66 67 68 69 70   
## 0.5187410 0.5187410 0.5187410 0.5187410 0.5187410 0.5187410 0.5187410   
## 71 72 73 74 75 76 77   
## 0.5187410 0.5187410 0.5187410 0.5187410 0.5187410 0.5187410 0.5187410   
## 78 79 80 81 82 83 84   
## 0.5187410 0.5187410 0.5187410 0.5187410 0.5187410 0.5187410 0.5187410   
## 85 86 87 88 89 90 91   
## 0.5187410 0.5187410 0.5187410 0.5187410 0.5187410 0.5187410 -1.2158483   
## 92 93 94 95 96 97 98   
## -1.2158483 -1.2158483 -1.2158483 -1.2158483 -1.2158483 -1.2158483 -1.2158483   
## 99 100 101 102 103 104 105   
## -1.2158483 -1.2158483 -1.2158483 -1.2158483 -1.2158483 -1.2158483 -1.2158483   
## 106 107 108 109 110 111 112   
## -1.2158483 -1.4177601 -1.4177601 -1.4177601 -1.4177601 -1.4177601 -1.4177601   
## 113 114 115 116 117 118 119   
## -1.4177601 -1.4177601 -1.4177601 -1.4177601 -1.4177601 -1.6532027 -1.6532027   
## 120 121 122 123 124 125 126   
## -1.6532027 -1.6532027 -1.9277443 -1.9277443 -1.9277443 -1.9277443 -1.9277443   
## 127 128 129 130   
## -1.9277443 -1.9277443 -1.9277443 -1.9277443

حال میخواهیم مقادیر پیش بینی شده یا برازیده شده توسط مدل را به ازای همان مقادیر جدول بدست آوریم:

fitted.values(fit1)

## 1 2 3 4 5 6 7 8   
## 0.5964955 0.5964955 0.5964955 0.5964955 0.5964955 0.5964955 0.5964955 0.5964955   
## 9 10 11 12 13 14 15 16   
## 0.5964955 0.5964955 0.5964955 0.5964955 0.5964955 0.5964955 0.5964955 0.5964955   
## 17 18 19 20 21 22 23 24   
## 0.5964955 0.5964955 0.5964955 0.5964955 0.5964955 0.5964955 0.5964955 0.5964955   
## 25 26 27 28 29 30 31 32   
## 0.5964955 0.6677789 0.6677789 0.6677789 0.6677789 0.6677789 0.6677789 0.6677789   
## 33 34 35 36 37 38 39 40   
## 0.6677789 0.6677789 0.6677789 0.6677789 0.6677789 0.6677789 0.6677789 0.6677789   
## 41 42 43 44 45 46 47 48   
## 0.6677789 0.7321246 0.7321246 0.7321246 0.7321246 0.7321246 0.7321246 0.7321246   
## 49 50 51 52 53 54 55 56   
## 0.7321246 0.7321246 0.7321246 0.7321246 0.7321246 0.7321246 0.7321246 0.7321246   
## 57 58 59 60 61 62 63 64   
## 0.7321246 0.7321246 0.7321246 0.7321246 0.7321246 0.7879648 0.7879648 0.7879648   
## 65 66 67 68 69 70 71 72   
## 0.7879648 0.7879648 0.7879648 0.7879648 0.7879648 0.7879648 0.7879648 0.7879648   
## 73 74 75 76 77 78 79 80   
## 0.7879648 0.7879648 0.7879648 0.7879648 0.7879648 0.7879648 0.7879648 0.7879648   
## 81 82 83 84 85 86 87 88   
## 0.7879648 0.7879648 0.7879648 0.7879648 0.7879648 0.7879648 0.7879648 0.7879648   
## 89 90 91 92 93 94 95 96   
## 0.7879648 0.7879648 0.5964955 0.5964955 0.5964955 0.5964955 0.5964955 0.5964955   
## 97 98 99 100 101 102 103 104   
## 0.5964955 0.5964955 0.5964955 0.5964955 0.5964955 0.5964955 0.5964955 0.5964955   
## 105 106 107 108 109 110 111 112   
## 0.5964955 0.5964955 0.6677789 0.6677789 0.6677789 0.6677789 0.6677789 0.6677789   
## 113 114 115 116 117 118 119 120   
## 0.6677789 0.6677789 0.6677789 0.6677789 0.6677789 0.7321246 0.7321246 0.7321246   
## 121 122 123 124 125 126 127 128   
## 0.7321246 0.7879648 0.7879648 0.7879648 0.7879648 0.7879648 0.7879648 0.7879648   
## 129 130   
## 0.7879648 0.7879648

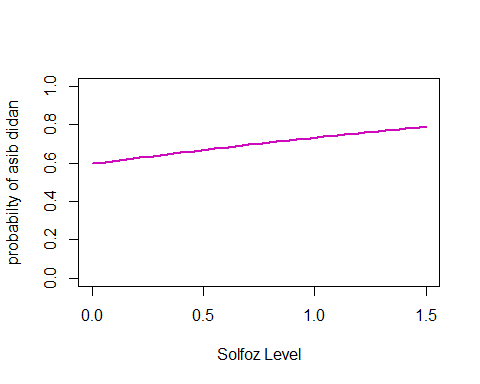
بدست آوردن همان مقادیر بالا به کمک دستور predict بصورت زیر است که باید نوع آنرا پاسخ قرار بدهیم.

predict.glm(fit1 , type = "response")

## 1 2 3 4 5 6 7 8   
## 0.5964955 0.5964955 0.5964955 0.5964955 0.5964955 0.5964955 0.5964955 0.5964955   
## 9 10 11 12 13 14 15 16   
## 0.5964955 0.5964955 0.5964955 0.5964955 0.5964955 0.5964955 0.5964955 0.5964955   
## 17 18 19 20 21 22 23 24   
## 0.5964955 0.5964955 0.5964955 0.5964955 0.5964955 0.5964955 0.5964955 0.5964955   
## 25 26 27 28 29 30 31 32   
## 0.5964955 0.6677789 0.6677789 0.6677789 0.6677789 0.6677789 0.6677789 0.6677789   
## 33 34 35 36 37 38 39 40   
## 0.6677789 0.6677789 0.6677789 0.6677789 0.6677789 0.6677789 0.6677789 0.6677789   
## 41 42 43 44 45 46 47 48   
## 0.6677789 0.7321246 0.7321246 0.7321246 0.7321246 0.7321246 0.7321246 0.7321246   
## 49 50 51 52 53 54 55 56   
## 0.7321246 0.7321246 0.7321246 0.7321246 0.7321246 0.7321246 0.7321246 0.7321246   
## 57 58 59 60 61 62 63 64   
## 0.7321246 0.7321246 0.7321246 0.7321246 0.7321246 0.7879648 0.7879648 0.7879648   
## 65 66 67 68 69 70 71 72   
## 0.7879648 0.7879648 0.7879648 0.7879648 0.7879648 0.7879648 0.7879648 0.7879648   
## 73 74 75 76 77 78 79 80   
## 0.7879648 0.7879648 0.7879648 0.7879648 0.7879648 0.7879648 0.7879648 0.7879648   
## 81 82 83 84 85 86 87 88   
## 0.7879648 0.7879648 0.7879648 0.7879648 0.7879648 0.7879648 0.7879648 0.7879648   
## 89 90 91 92 93 94 95 96   
## 0.7879648 0.7879648 0.5964955 0.5964955 0.5964955 0.5964955 0.5964955 0.5964955   
## 97 98 99 100 101 102 103 104   
## 0.5964955 0.5964955 0.5964955 0.5964955 0.5964955 0.5964955 0.5964955 0.5964955   
## 105 106 107 108 109 110 111 112   
## 0.5964955 0.5964955 0.6677789 0.6677789 0.6677789 0.6677789 0.6677789 0.6677789   
## 113 114 115 116 117 118 119 120   
## 0.6677789 0.6677789 0.6677789 0.6677789 0.6677789 0.7321246 0.7321246 0.7321246   
## 121 122 123 124 125 126 127 128   
## 0.7321246 0.7879648 0.7879648 0.7879648 0.7879648 0.7879648 0.7879648 0.7879648   
## 129 130   
## 0.7879648 0.7879648

seq<-data.frame(x=seq(min(x),max(x),len=50))  
seq$y=predict(fit1,newdata= seq,type="response")  
plot(x=c(0 , 1.5),y=c(0,1),type="n" ,col= c(1:50),pch=19,cex=1.1,  
 ylab="probabilty of asib didan" , xlab="Solfoz Level")  
lines(y~x, data = seq,col=6,lwd= 2)

حال نمودار مربوط به احتمال آسیب دیدن یک درخت را با توجه به مقادیر مختلف سولفور مشاهده میکنیم.



حال در زیر برای مشاهده کردن مقادیر باقی مانده های اصلی و باقی مانده های استاندارد شده و استیودنت شده داریم:

residuals(fit1)

## 1 2 3 4 5 6 7   
## 1.0165467 1.0165467 1.0165467 1.0165467 1.0165467 1.0165467 1.0165467   
## 8 9 10 11 12 13 14   
## 1.0165467 1.0165467 1.0165467 1.0165467 1.0165467 1.0165467 1.0165467   
## 15 16 17 18 19 20 21   
## 1.0165467 1.0165467 1.0165467 1.0165467 1.0165467 1.0165467 1.0165467   
## 22 23 24 25 26 27 28   
## 1.0165467 1.0165467 1.0165467 1.0165467 0.8986636 0.8986636 0.8986636   
## 29 30 31 32 33 34 35   
## 0.8986636 0.8986636 0.8986636 0.8986636 0.8986636 0.8986636 0.8986636   
## 36 37 38 39 40 41 42   
## 0.8986636 0.8986636 0.8986636 0.8986636 0.8986636 0.8986636 0.7896892   
## 43 44 45 46 47 48 49   
## 0.7896892 0.7896892 0.7896892 0.7896892 0.7896892 0.7896892 0.7896892   
## 50 51 52 53 54 55 56   
## 0.7896892 0.7896892 0.7896892 0.7896892 0.7896892 0.7896892 0.7896892   
## 57 58 59 60 61 62 63   
## 0.7896892 0.7896892 0.7896892 0.7896892 0.7896892 0.6903649 0.6903649   
## 64 65 66 67 68 69 70   
## 0.6903649 0.6903649 0.6903649 0.6903649 0.6903649 0.6903649 0.6903649   
## 71 72 73 74 75 76 77   
## 0.6903649 0.6903649 0.6903649 0.6903649 0.6903649 0.6903649 0.6903649   
## 78 79 80 81 82 83 84   
## 0.6903649 0.6903649 0.6903649 0.6903649 0.6903649 0.6903649 0.6903649   
## 85 86 87 88 89 90 91   
## 0.6903649 0.6903649 0.6903649 0.6903649 0.6903649 0.6903649 -1.3472696   
## 92 93 94 95 96 97 98   
## -1.3472696 -1.3472696 -1.3472696 -1.3472696 -1.3472696 -1.3472696 -1.3472696   
## 99 100 101 102 103 104 105   
## -1.3472696 -1.3472696 -1.3472696 -1.3472696 -1.3472696 -1.3472696 -1.3472696   
## 106 107 108 109 110 111 112   
## -1.3472696 -1.4845569 -1.4845569 -1.4845569 -1.4845569 -1.4845569 -1.4845569   
## 113 114 115 116 117 118 119   
## -1.4845569 -1.4845569 -1.4845569 -1.4845569 -1.4845569 -1.6231040 -1.6231040   
## 120 121 122 123 124 125 126   
## -1.6231040 -1.6231040 -1.7612513 -1.7612513 -1.7612513 -1.7612513 -1.7612513   
## 127 128 129 130   
## -1.7612513 -1.7612513 -1.7612513 -1.7612513

rstudent(fit1)

## 1 2 3 4 5 6 7   
## 1.0230939 1.0230939 1.0230939 1.0230939 1.0230939 1.0230939 1.0230939   
## 8 9 10 11 12 13 14   
## 1.0230939 1.0230939 1.0230939 1.0230939 1.0230939 1.0230939 1.0230939   
## 15 16 17 18 19 20 21   
## 1.0230939 1.0230939 1.0230939 1.0230939 1.0230939 1.0230939 1.0230939   
## 22 23 24 25 26 27 28   
## 1.0230939 1.0230939 1.0230939 1.0230939 0.9010903 0.9010903 0.9010903   
## 29 30 31 32 33 34 35   
## 0.9010903 0.9010903 0.9010903 0.9010903 0.9010903 0.9010903 0.9010903   
## 36 37 38 39 40 41 42   
## 0.9010903 0.9010903 0.9010903 0.9010903 0.9010903 0.9010903 0.7920193   
## 43 44 45 46 47 48 49   
## 0.7920193 0.7920193 0.7920193 0.7920193 0.7920193 0.7920193 0.7920193   
## 50 51 52 53 54 55 56   
## 0.7920193 0.7920193 0.7920193 0.7920193 0.7920193 0.7920193 0.7920193   
## 57 58 59 60 61 62 63   
## 0.7920193 0.7920193 0.7920193 0.7920193 0.7920193 0.6941822 0.6941822   
## 64 65 66 67 68 69 70   
## 0.6941822 0.6941822 0.6941822 0.6941822 0.6941822 0.6941822 0.6941822   
## 71 72 73 74 75 76 77   
## 0.6941822 0.6941822 0.6941822 0.6941822 0.6941822 0.6941822 0.6941822   
## 78 79 80 81 82 83 84   
## 0.6941822 0.6941822 0.6941822 0.6941822 0.6941822 0.6941822 0.6941822   
## 85 86 87 88 89 90 91   
## 0.6941822 0.6941822 0.6941822 0.6941822 0.6941822 0.6941822 -1.3580569   
## 92 93 94 95 96 97 98   
## -1.3580569 -1.3580569 -1.3580569 -1.3580569 -1.3580569 -1.3580569 -1.3580569   
## 99 100 101 102 103 104 105   
## -1.3580569 -1.3580569 -1.3580569 -1.3580569 -1.3580569 -1.3580569 -1.3580569   
## 106 107 108 109 110 111 112   
## -1.3580569 -1.4904882 -1.4904882 -1.4904882 -1.4904882 -1.4904882 -1.4904882   
## 113 114 115 116 117 118 119   
## -1.4904882 -1.4904882 -1.4904882 -1.4904882 -1.4904882 -1.6315623 -1.6315623   
## 120 121 122 123 124 125 126   
## -1.6315623 -1.6315623 -1.7818520 -1.7818520 -1.7818520 -1.7818520 -1.7818520   
## 127 128 129 130   
## -1.7818520 -1.7818520 -1.7818520 -1.7818520

rstandard(fit1)

## 1 2 3 4 5 6 7   
## 1.0265315 1.0265315 1.0265315 1.0265315 1.0265315 1.0265315 1.0265315   
## 8 9 10 11 12 13 14   
## 1.0265315 1.0265315 1.0265315 1.0265315 1.0265315 1.0265315 1.0265315   
## 15 16 17 18 19 20 21   
## 1.0265315 1.0265315 1.0265315 1.0265315 1.0265315 1.0265315 1.0265315   
## 22 23 24 25 26 27 28   
## 1.0265315 1.0265315 1.0265315 1.0265315 0.9025996 0.9025996 0.9025996   
## 29 30 31 32 33 34 35   
## 0.9025996 0.9025996 0.9025996 0.9025996 0.9025996 0.9025996 0.9025996   
## 36 37 38 39 40 41 42   
## 0.9025996 0.9025996 0.9025996 0.9025996 0.9025996 0.9025996 0.7936563   
## 43 44 45 46 47 48 49   
## 0.7936563 0.7936563 0.7936563 0.7936563 0.7936563 0.7936563 0.7936563   
## 50 51 52 53 54 55 56   
## 0.7936563 0.7936563 0.7936563 0.7936563 0.7936563 0.7936563 0.7936563   
## 57 58 59 60 61 62 63   
## 0.7936563 0.7936563 0.7936563 0.7936563 0.7936563 0.6971117 0.6971117   
## 64 65 66 67 68 69 70   
## 0.6971117 0.6971117 0.6971117 0.6971117 0.6971117 0.6971117 0.6971117   
## 71 72 73 74 75 76 77   
## 0.6971117 0.6971117 0.6971117 0.6971117 0.6971117 0.6971117 0.6971117   
## 78 79 80 81 82 83 84   
## 0.6971117 0.6971117 0.6971117 0.6971117 0.6971117 0.6971117 0.6971117   
## 85 86 87 88 89 90 91   
## 0.6971117 0.6971117 0.6971117 0.6971117 0.6971117 0.6971117 -1.3605029   
## 92 93 94 95 96 97 98   
## -1.3605029 -1.3605029 -1.3605029 -1.3605029 -1.3605029 -1.3605029 -1.3605029   
## 99 100 101 102 103 104 105   
## -1.3605029 -1.3605029 -1.3605029 -1.3605029 -1.3605029 -1.3605029 -1.3605029   
## 106 107 108 109 110 111 112   
## -1.3605029 -1.4910590 -1.4910590 -1.4910590 -1.4910590 -1.4910590 -1.4910590   
## 113 114 115 116 117 118 119   
## -1.4910590 -1.4910590 -1.4910590 -1.4910590 -1.4910590 -1.6312579 -1.6312579   
## 120 121 122 123 124 125 126   
## -1.6312579 -1.6312579 -1.7784637 -1.7784637 -1.7784637 -1.7784637 -1.7784637   
## 127 128 129 130   
## -1.7784637 -1.7784637 -1.7784637 -1.7784637

حال در زیر مقدار کیبش را میتوانیم مشاهده بکنیم که 3.69 یا همان میزان اختلاف شده است. و با نارنجی نمایش داده شده است.

anova(fit1)

## Analysis of Deviance Table  
##   
## Model: binomial, link: logit  
##   
## Response: y  
##   
## Terms added sequentially (first to last)  
##   
##   
## Df Deviance Resid. Df Resid. Dev  
## NULL 129 160.48  
## x 1 3.6924 128 156.79

fit1$null.deviance

## [1] 160.4829

fit1$deviance

## [1] 156.7905

حال در زیر نمودار چندک چندک باقی مانده ها را رسم میکنیم و فرض نرمال بودن آنها کامل رد میشود.

qqnorm(residuals(fit1))

