hw5

Sohaib Syed

2022-11-18

Problem 1

a

```r
performance_ability_original<-read.table('CH14PR09.txt',col.names = c('y','x'))
fit<-glm(y~x,data=performance_ability_original,family=binomial(link="logit"))
summary(fit)
```

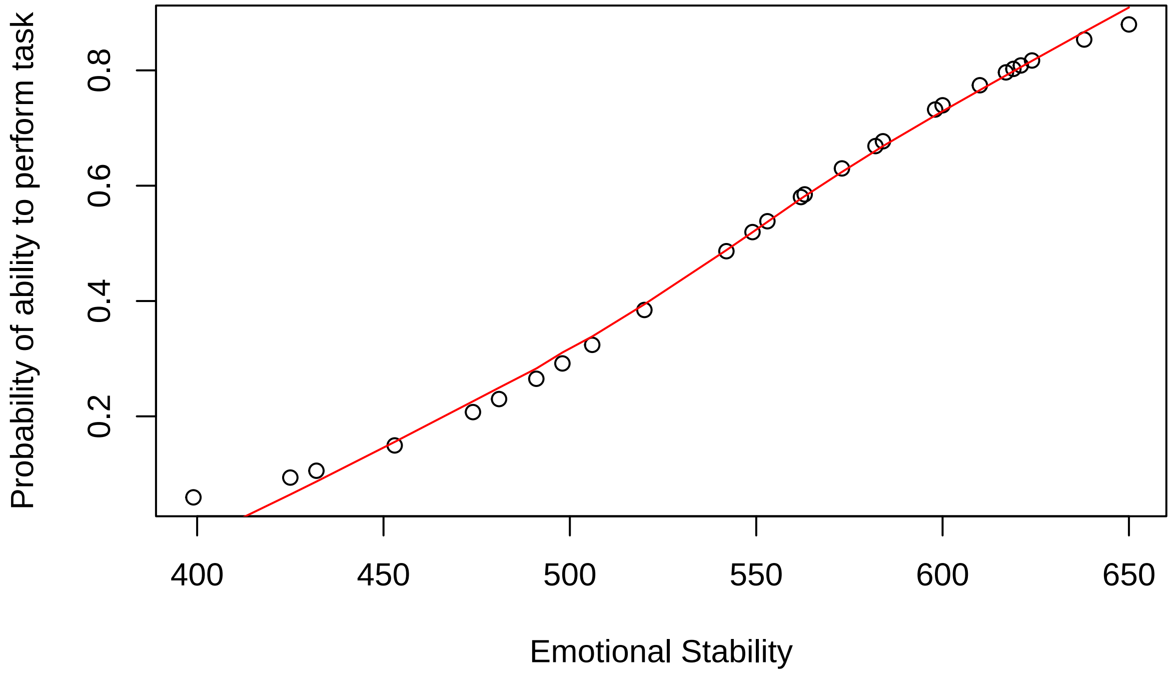
```
##
## Call:
## glm(formula = y ~ x, family = binomial(link = "logit"), data = performance_ability_original)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7845  -0.8350   0.5065   0.8371   1.7145
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -10.308925   4.376997  -2.355   0.0185 *
## x             0.018920   0.007877   2.402   0.0163 *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 37.393  on 26  degrees of freedom
## Residual deviance: 29.242  on 25  degrees of freedom
## AIC: 33.242
##
## Number of Fisher Scoring iterations: 4
```

B0=-10.308925 B1=0.01892

fitted equation= pi_hat= [1+exp(10.3089-0.1892X)]^-1

b

```
plot(performance_ability_original$x,fitted(fit),xlab='Emotional Stability' , ylab= ' Probability of abil
lines(lowess(performance_ability_original$x,fitted(fit)),col='red')
```



Yes, the logistic response function fits very well.

c

```
exp(0.01892)
```

```
## [1] 1.0191
```

This is the odd-ratio. Since b0 and B1 are on log scale we scale by exponent. This number is close to one thus an increase of 1 of the predictor is not likely to increase the odds of an event occurring. Specific to our data, an increase of 1 on the test score is unlikely to increase the chances of a person being able to perform.

d

```
pred<-predict(fit,newdata=data.frame(x=550),type='response')
pred
```

```
##         1
## 0.5242263
```

e

We have to manipulate the pi function from part a of this problem to solve for x:

.7=[1+exp(10.3089-0.01892X)])]^-1 1/.7=1+exp(10.3089-0.01892X) (1/.7)-1=exp(10.3089-0.01892X) ln((1/.7)-1)=10.3089-.01892x ln((1/.7)-1)-10.3089=-.01892x (ln((1/.7)-1)-10.3089)/-.01892=x =589.65 so test score is 589.65

14.10(a)

```
fitprobit<-glm(y~x,data=performance_ability_original,family=binomial(link="probit"))
summary(fitprobit)
```

```
##
## Call:
## glm(formula = y ~ x, family = binomial(link = "probit"), data = performance_ability_original)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7940  -0.8336   0.4824   0.8380   1.7223
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.374398   2.464111  -2.587  0.00968 **
## x            0.011695   0.004437   2.636  0.00839 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 37.393  on 26  degrees of freedom
## Residual deviance: 29.102  on 25  degrees of freedom
## AIC: 33.102
##
## Number of Fisher Scoring iterations: 5
```

B0=-6.374398 and B1=0.011695

probit = pi_hat= phi(B0+B1x)

pi_hat= phi(-6.374398+0.011695x)

The fits look similar; a negative B0 and very small non-zero B1.

I can conclude that since the response is binary both probit and logit response functions can fit data.

Problem 2

a

```
car_purchase_original<-read.table('CH14PR13.txt',col.names = c('y','x1','x2'))
car_fit_full<-glm(y~x1+x2,data=car_purchase_original,family=binomial(link="logit"))
summary(car_fit_full)
```

```
##
## Call:
## glm(formula = y ~ x1 + x2, family = binomial(link = "logit"),
##     data = car_purchase_original)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6189  -0.8949  -0.5880   0.9653   2.0846
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.73931    2.10195  -2.255   0.0242 *
## x1           0.06773    0.02806   2.414   0.0158 *
## x2           0.59863    0.39007   1.535   0.1249
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 44.987  on 32  degrees of freedom
## Residual deviance: 36.690  on 30  degrees of freedom
## AIC: 42.69
##
## Number of Fisher Scoring iterations: 4
```

B0= -4.73931, B1= 0.06773, B2= 0.59863

fitted response function = pi_hat= [1+exp(4.73931-0.06773X1-0.59863X2)]^-1

b

```
exp(car_fit_full$coefficients[2])
```

```
##       x1
## 1.070079
```

```
exp(car_fit_full$coefficients[3])
```

```
##       x2
## 1.819627
```

These are the odds ratios for the two predictors. For B1, I interpret it as when the annual family income increases by 1 (thousands), the probability of the family buying a new car increase by 1.070079. For B2, an increase of the age of the oldest car in the family by 1 year, the chances of a buying a new car increases by 1.81927.

c

```
pred_car<-predict(car_fit_full,newdata=data.frame(x1=50,x2=3),type='response')
pred_car
```

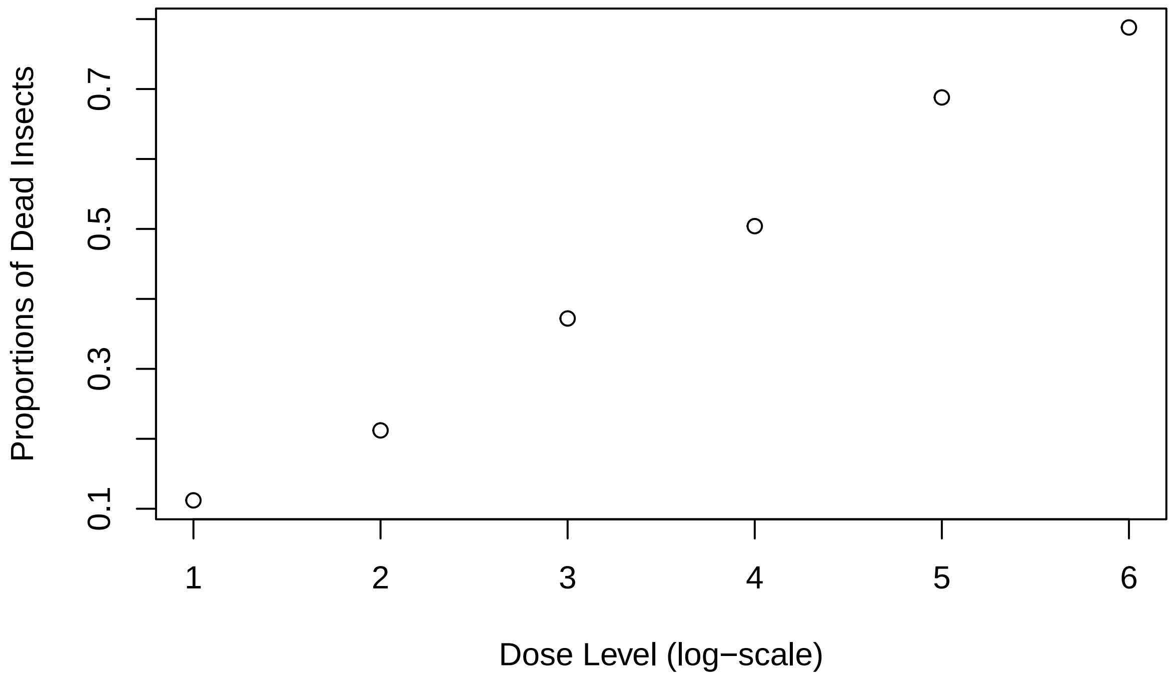
```
##         1
## 0.6090245
```

The probability is 60.902%

Problem 3

a

```
toxicity_original<-read.table('CH14PR12.txt',col.names = c('x1','nj','y'))
prob_i<-toxicity_original$y/toxicity_original$nj
plot(toxicity_original$x1,prob_i,xlab='Dose Level (log-scale)', ylab='Proportions of Dead Insects')
```



It seems apprpriate to use logistic curve here because there is an s-shape curve.

b

```
toxicity_fit<-glm(prob_i~x1,data=toxicity_original,family=binomial(link='logit'))
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
summary(toxicity_fit)
```

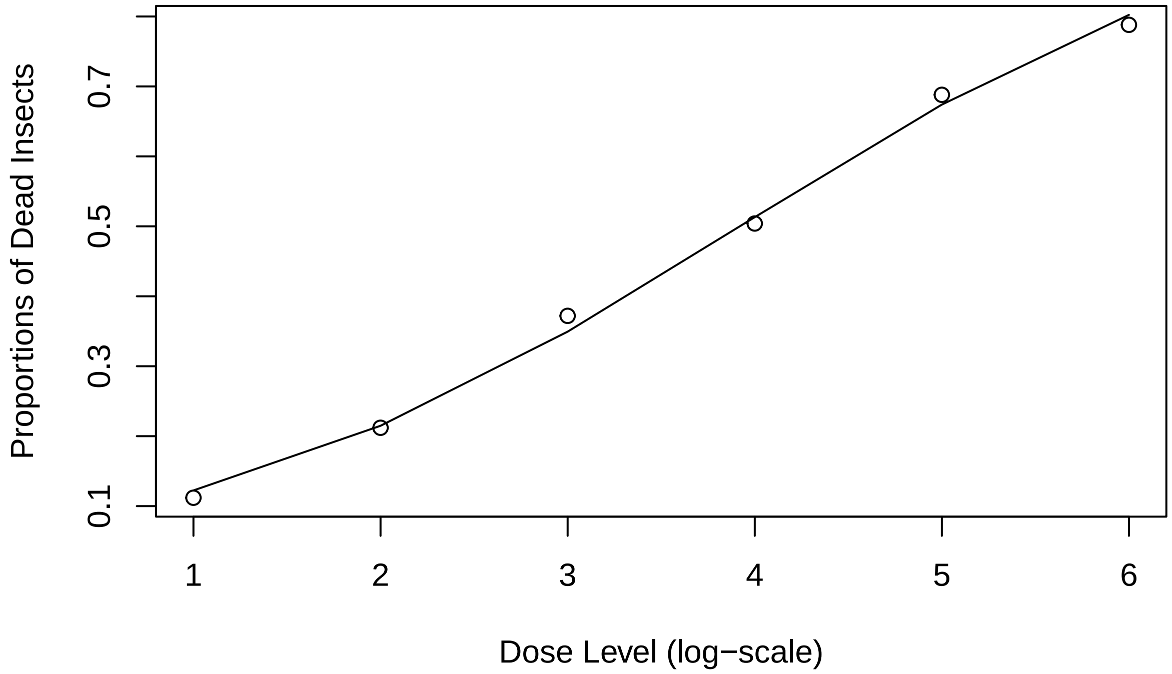
```
##
## Call:
## glm(formula = prob_i ~ x1, family = binomial(link = "logit"),
##     data = toxicity_original)
##
## Deviance Residuals:
##         1          2          3          4          5          6
## -0.032203  -0.007051   0.047185  -0.018146   0.030001  -0.035409
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.6437     2.4682  -1.071    0.284
## x1            0.6740     0.6184   1.090    0.276
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1.5322779  on 5  degrees of freedom
## Residual deviance: 0.0057964  on 4  degrees of freedom
## AIC: 8.1692
##
## Number of Fisher Scoring iterations: 4
```

B0= -2.6437 and B1=0.6740

response function= pi_hat= [1+exp(2.6437-0.6740X)]ˆ-1

c

```
plot(toxicity_original$x1,prob_i,xlab='Dose Level (log-scale)', ylab='Proportions of Dead Insects')
lines(toxicity_original$x1,fitted(toxicity_fit))
```



Yes, the fitted response funciton fits the data very well.

d

```
exp(toxicity_fit$coefficients[2])
```

```
##       x1
## 1.962056
```

This number is the odds ratio of B1. When the dosage level is increased by 1 (on log scale) then the odds of an insect dying are increased by 1.962056

e

```
toxic_pred<-predict(toxicity_fit,newdata=data.frame(x1=3.5),type='response')
toxic_pred
```

```
##         1
## 0.4293018
```

It is about a 42.93% chance.

f

```
(log((1/0.5)-1)-2.6437)/-0.6740
```

```
## [1] 3.922404
```

We have to manipulate the response function from part b of this problem to solve for x: 0.5=[1+exp(2.6437-0.0674X)]ˆ-1 1/0.5=[1+exp(2.6437-0.0674X)] (1/0.5)-1=exp(2.6437-0.0674X) ln((1/0.5)-1)=2.6437-0.0674X ln((1/0.5)-1)-2.6437=-0.0674X (ln((1/0.5)-1)-2.6437)/-0.0674=x =3.922404

that is the median lethal dose.