MATH 523 - ASSIGNMENT 1 : SOLUTIONS

(a) Using the results from lectures, after reparameterization, we have that the βs are related to the log odds of being in the different categories: we have for the estimators that $\widehat{\beta}_0 = Y_1/(m_1 - Y_1)$, and

$$\widehat{\beta}_i = \log\left(\frac{Y_i/(m_i - Y_i)}{Y_1/(m_1 - Y_1)}\right) \qquad i = 2, \ldots, 11.$$

The estimates can be computed in R as follows:

```
> attach(Turbines)
> log(y[1]/(m[1]-y[1]))
[1] -2.917771
> log((y/(m-y))/(y[1]/(m[1]-y[1])))
 [1] 0.0000000 0.4122448 0.6151856 0.6740261 1.3083328 1.7137979 1.6184877
 [8] 2.7636201 3.5239065 3.0178542 3.2542430
> detach(Turbines)
```

Using `glm` in R, as in the O-rings problem from Notes, we may repeat the computation, and compute standard errors etc.

```
>summary(glm(cbind(y,m-y) ~ factor(x),data=Turbines,family=binomial))
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    -2.9178     0.7260  -4.019 5.84e-05 ***
factor(x)1000   0.4122     0.8930   0.462 0.644336
factor(x)1400   0.6152     0.9454   0.651 0.515210
factor(x)1800   0.6740     0.8277   0.814 0.415435
factor(x)2200   1.3083     0.8758   1.494 0.135210
factor(x)2600   1.7138     0.8194   2.091 0.036488 *
factor(x)3000   1.6185     0.8176   1.980 0.047748 *
factor(x)3400   2.7636     0.9146   3.022 0.002515 **
factor(x)3800   3.5239     0.8098   4.351 1.35e-05 ***
factor(x)4200   3.0179     0.7920   3.810 0.000139 ***
factor(x)4600   3.2542     0.8008   4.064 4.83e-05 ***
```

4 MARKS

(b) This can be computed routinely using code from the Notes:

```
>summary(glm(cbind(y,m-y) ~ x,data=Turbines,family=binomial))

Call:glm(formula = cbind(y, m - y) ~ x, family = binomial, data = Turbines)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.6070615  0.3533875 -10.207   <2e-16 ***
x            0.0009121  0.0001084   8.416   <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: 98.5054  on 10  degrees of freedom
Residual deviance:  7.9075  on  9  degrees of freedom
AIC: 50.291
Number of Fisher Scoring iterations: 4
```

4 MARKS

(c) With the `probit` link:

```
>summary(glm(cbind(y,m-y) ~ x,data=Turbines,family=binomial(link=probit)))

Call:glm(formula = cbind(y, m - y) ~ x, family = binomial(link = probit),
    data = Turbines)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.080e+00  1.852e-01 -11.228   <2e-16 ***
x            5.230e-04  5.973e-05   8.757   <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: 98.5054  on 10  degrees of freedom
Residual deviance:  8.5135  on  9  degrees of freedom
AIC: 50.897

Number of Fisher Scoring iterations: 4
```

3 MARKS

(d) With the `cloglog` link:

```
>summary(glm(cbind(y,m-y) ~ x,data=Turbines,family=binomial(link=cloglog)))

Call: glm(formula = cbind(y, m - y) ~ x, family = binomial(link = cloglog),
    data = Turbines)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.360e+00  3.061e-01 -10.977   <2e-16 ***
x            7.480e-04  8.622e-05   8.675   <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: 98.5054  on 10  degrees of freedom
Residual deviance:  8.7065  on  9  degrees of freedom
AIC: 51.09

Number of Fisher Scoring iterations: 4
```

3 MARKS

(e) We may use the `predict` function in `R` to compute the predictions with associated standard errors:

```
modl<-(glm(cbind(y,m-y) ~ x,data=Turbines,family=binomial))
modp<-(glm(cbind(y,m-y) ~ x,data=Turbines,family=binomial(link=probit)))
modc<-(glm(cbind(y,m-y) ~ x,data=Turbines,family=binomial(link=cloglog)))

newd<-data.frame(x=2000)
predl<-predict(modl,newdata=newd,se.fit=TRUE)
predp<-predict(modp,newdata=newd,se.fit=TRUE)
predc<-predict(modc,newdata=newd,se.fit=TRUE)

> preds<-
rbind(c(predl$fit-2*predl$se.fit,predl$fit,predl$fit+2*predl$se.fit),
+ c(predp$fit-2*predp$se.fit,predp$fit,predp$fit+2*predp$se.fit),
+ c(predc$fit-2*predc$se.fit,predc$fit,predc$fit+2*predc$se.fit))

>preds
             1         1          1
[1,] -2.120517 -1.782886 -1.4452542
[2,] -1.210206 -1.033799 -0.8573924
[3,] -2.169735 -1.863879 -1.5580239
```

Note that `predict` works by default on the **linear predictor** scale, and computes

$$\widehat{\eta} = \widehat{\beta}_0 + \widehat{\beta}_1 x$$

with the associated standard error. To obtain the results on the probability scale, we need to back-transform via the link function:

$$\begin{aligned} \texttt{logit} &: g^{-1}(t) = e^t/(1+e^t) \\ \texttt{probit} &: g^{-1}(t) = \Phi^{-1}(t) \\ \texttt{cloglog} &: g^{-1}(t) = 1 - \exp(-\exp(t)) \end{aligned}$$

Here are the probabilities of fissure development, with 95% confidence intervals:

```
> ppreds<-preds
> ppreds[1,]<-exp(preds[1,])/(1+exp(preds[1,]))
> ppreds[2,]<-pnorm(preds[2,])
> ppreds[3,]<-1-exp(-exp(preds[3,]))
> dimnames(ppreds)[[2]]<-c("Lower","Est","Upper")
> ppreds
         Lower       Est     Upper
[1,] 0.1071186 0.1439472 0.1907330
[2,] 0.1131000 0.1506150 0.1956140
[3,] 0.1079275 0.1436447 0.1898629
```

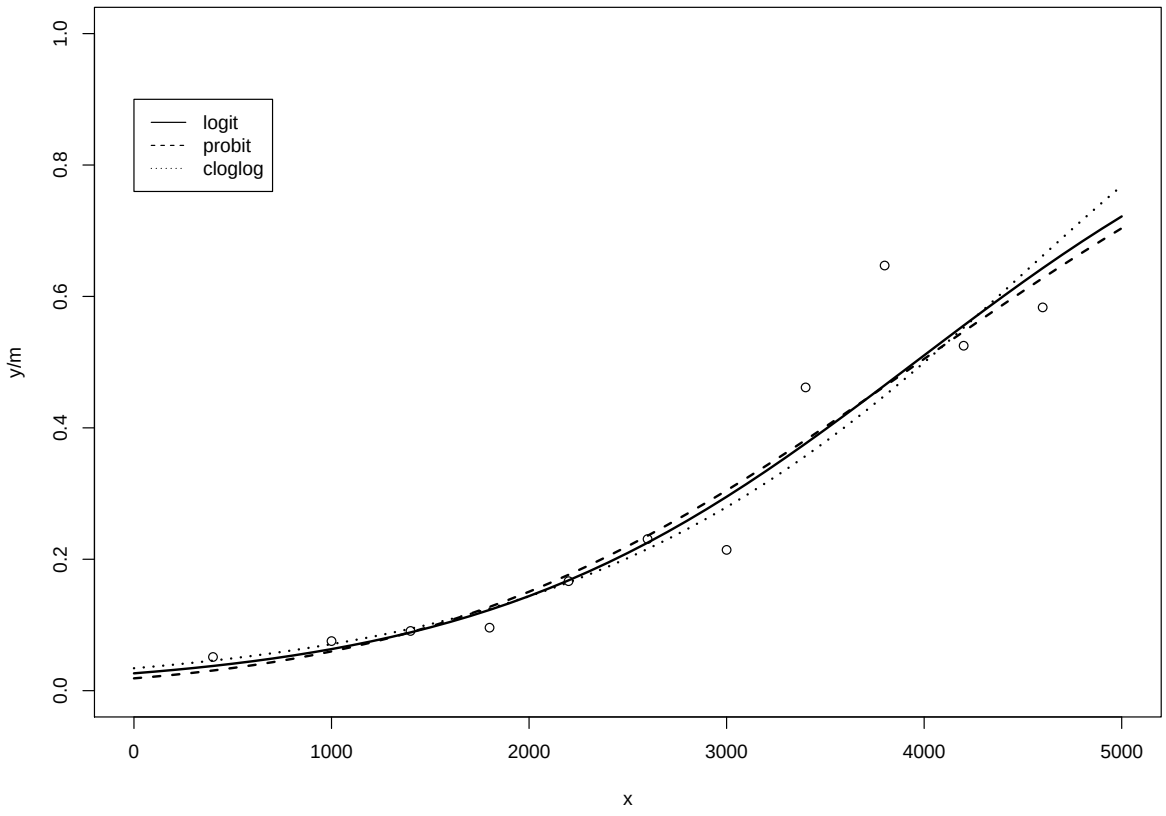
that is, there is not a tremendous difference.

3 MARKS

(f) Code to produce a suitable plot (of y/m vs x, with fitted lines for each link) is given below:

```
xv<-seq(0,5000,by=100)
newx<-data.frame(x=xv)
yl<-predict(modl,newdata=newx,type="response")
yp<-predict(modp,newdata=newx,type="response")
yc<-predict(modc,newdata=newx,type="response")

attach(Turbines)
pdf("Assign1-Fit.pdf",paper="USr",height=8.5,width=11)
plot(x,y/m,ylim=range(0,1),xlim=range(0,5000))
detach(Turbines)
lines(xv,yl,lty=1,lwd=2)
lines(xv,yp,lty=2,lwd=2)
lines(xv,yc,lty=3,lwd=2)
legend(0,0.9,c("logit","probit","cloglog"),lty=1:3)
dev.off()
```



Again, there is not a great deal of difference between the fitted response curves on this probability scale. The fit generally appears adequate, although one outlying point is evident.

3 MARKS