Logistic regression

```
library(faraway)
library(MASS)
data(orings)
str(orings)
```

```
## 'data.frame':    23 obs. of  2 variables:
##  $ temp  : num  53 57 58 63 66 67 67 67 68 69 ...
##  $ damage: num  5 1 1 1 0 0 0 0 0 0 ...
```

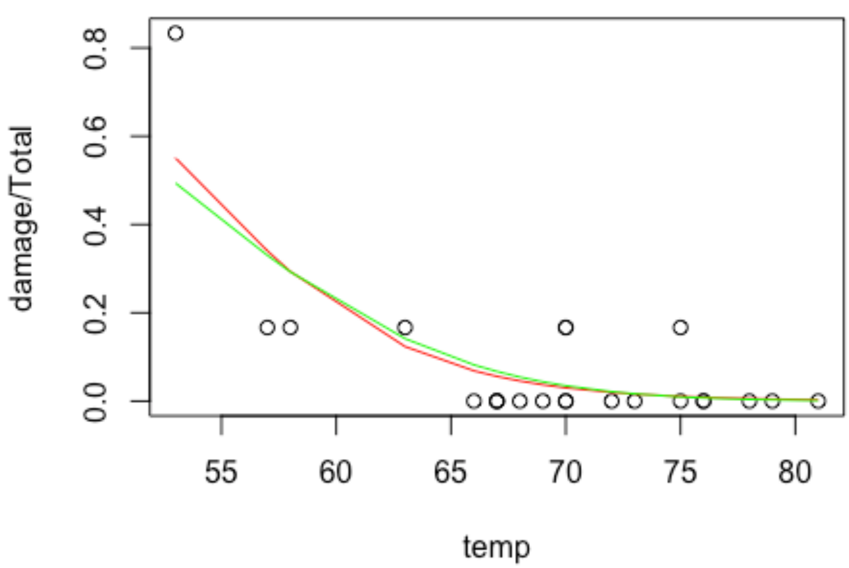
```
# If we feed glm( ) a table (or matrix) in which the first column is
number of successes and the second column is number of failures, R will
take care of the coding for us.

Total<-6
summary(orings)
```

```
##       temp           damage
##  Min.   :53.00   Min.   :0.0000
##  1st Qu.:67.00   1st Qu.:0.0000
##  Median :70.00   Median :0.0000
##  Mean   :69.57   Mean   :0.4783
##  3rd Qu.:75.00   3rd Qu.:1.0000
##  Max.   :81.00   Max.   :5.0000
```

```
plot(damage/Total ~ temp, data=orings)

modl = glm(cbind(damage, Total-damage) ~ temp,family=binomial(logit),
data=orings)
modp <- glm(cbind(damage, Total-
damage)~temp,data=orings,family=binomial(probit))
plot(damage/Total ~ temp, data=orings)
lines(orings$temp, modl$fitted, type="l", col="red")
lines(orings$temp,modp$fitted,type="l",col="green")
```




```
#Let's look at how closely the fitted values from our logistic
regression match the observed values...

summary(modl) # display results
```

```
##
## Call:
## glm(formula = cbind(damage, Total - damage) ~ temp, family =
```

```
## glm(formula = cbind(damage, total - damage) ~ temp, family =
binomial(logit),
##     data = orings)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9529  -0.7345  -0.4393  -0.2079   1.9565
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 11.66299    3.29626   3.538 0.000403 ***
## temp        -0.21623    0.05318  -4.066 4.78e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 38.898  on 22  degrees of freedom
## Residual deviance: 16.912  on 21  degrees of freedom
## AIC: 33.675
##
## Number of Fisher Scoring iterations: 6
```

```
confint(modl) # 95% CI for the coefficients
```

```
## Waiting for profiling to be done...
```

```
##                 2.5 %    97.5 %
## (Intercept)  5.575195 18.737598
## temp        -0.332657 -0.120179
```

```
exp(coef(modl)) # exponentiated coefficients
```

```
##  (Intercept)         temp
## 1.161909e+05 8.055471e-01
```

```
exp(confint(modl)) # 95% CI for exponentiated coefficients
```

```
## Waiting for profiling to be done...
```

```
##                   2.5 %       97.5 %
## (Intercept) 263.8010254 1.372889e+08
## temp          0.7170161 8.867617e-01
```

```
predict(modl, type="response") # predicted values
```

```
##           1           2           3           4           5
6
## 0.550478817 0.340216592 0.293475686 0.123496147 0.068597710
0.056005745
##           7           8           9          10          11
 12
## 0.056005745 0.056005745 0.045612000 0.037071413 0.030079600
0.030079600
##          13          14          15          16          17
 18
## 0.030079600 0.030079600 0.019727169 0.015952356 0.010409884
0.010409884
##          19          20          21          22          23
## 0.008402660 0.008402660 0.005468670 0.004409961 0.002866088
```

```
residuals(modl, type="deviance") # residuals
```

```
##          1          2          3          4          5          6
##  1.4704007 -0.9529154 -0.7205297  0.3074051 -0.9234543 -0.8316384
##          7          8          9         10         11         12
## -0.8316384 -0.8316384 -0.7484783 -0.6732847  1.3807586 -0.6053885
##         13         14         15         16         17         18
##  1.3807586 -0.6053885 -0.4889705 -0.4392853 -0.3543633  1.9565043
##         19         20         21         22         23
## -0.3182105 -0.3182105 -0.2565232 -0.2302967 -0.1855867
```

```
# Deviance goodness of fit
1-pchisq(deviance(modl),df.residual(modl))
```

```
## [1] 0.7164099
```