stats330 A2

Kerui Du

02/09/2021

Question 1a

```
mean(Visits.df$visits)
```

```
## [1] 5.844076
```

```
var(Visits.df$visits)
```

```
## [1] 37.84379
```

```
observed=table(Visits.df$visits)
observed
```

```
##
##   0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19
## 635 520 493 368 362 275 268 227 190 173 122 108 102  92  70  61  53  49  26  28
##  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  36  37  39  41  43
##  28  26  24  17   9  12  13   7  11   4   7   5   5   3   2   2   1   2   1   1
##  44  45  46  48
##   1   1   1   1
```

```
n=sum(observed)
n
```

```
## [1] 4406
```

Two reasons that this output let us to believe that the Poisson model may not be appropriate for describing these data.

The first reason is that as one of the properties of poisson distribution, its mean should equal to its variance, however, in this dataset, the difference between mean and variance is quite large.

The second reason is that since it has mean around 5.8, which means the peak ought to reside between 5 and 6, whereas it’s at 1.

1b

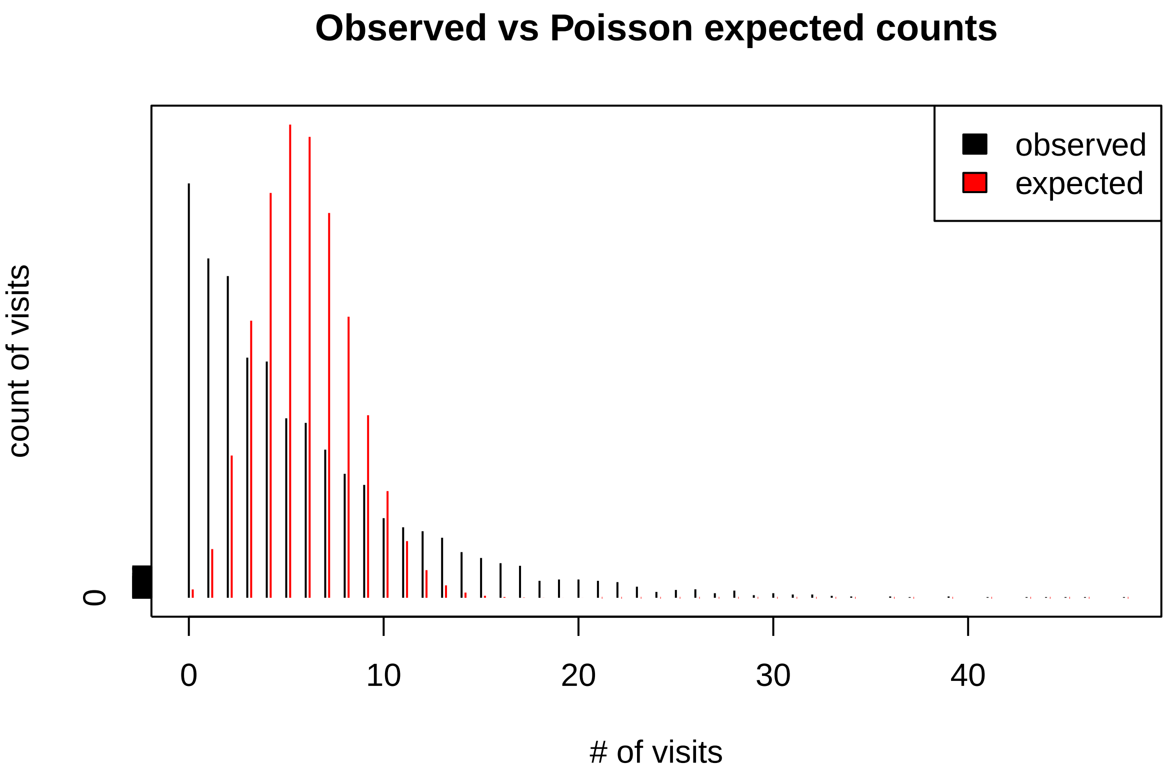
```
fit1 <- glm(visits~1, family = 'poisson', data = Visits.df)
exp(fit1$coefficients)
```

```
## (Intercept)
##    5.844076
```

Because it is a null model it will be only have one coefficient, which is its intercept, and the intercept is the mean of Y(visits).

1c

```
x=as.numeric(names(observed))
expected.Pois=n*dpois(x, lambda=exp(coef(fit1)))
plot(x,observed, type="h",lwd=1, lend="butt", xlab="# of visits",
     ylab="count of visits",
     main="Observed vs Poisson expected counts",
     xlim=range(x), ylim= c(0, max(observed,expected.Pois)))
lines(x+.2, expected.Pois,type="h",
      lwd=1, lend="butt",col="red")
legend("topright", fill=c("black","red"),
       legend=c("observed","expected"))
```



The distribution of observed is quite different from the distribution of expected, dpois give us the poisson distribution with mean 5.8, which means the peak should be somewhere at 6, but distribution of observed has peak at 0, and the count of visits decline along with the number of visits increase, and there is a large difference at 6, therefore, our model is not adequate for fitting these data.

1d

```
var <- exp(coef(fit1));unname(var)
```

```
## [1] 5.844076
```

```
prop_exp <- 1-ppois(12,exp(coef(fit1)));prop_exp
```

```
## [1] 0.007203304
```

```
prop_obs <- length(Visits.df$visits[Visits.df$visits>12])/n;prop_obs
```

```
## [1] 0.1277803
```

As the property of poisson distribution, we know that its mean is equivalent to its variance, thus, the variance is 5.84. it is quite different compared to the observed variance(37.8), and it helps to be more confident that poisson model is not adequate for fitting these data.

Meanwhile, the proportion of visits that exceed 12 of poisson distribution is around 0.72%, whereas around 12.8% for observed distribution, this can be another evidence to suggest that poisson model is not adequate for fitting these data.

1e

```
Egt5=expected.Pois>=5
E.Pois=c(expected.Pois[Egt5], sum(expected.Pois[!Egt5]))
O.Pois=c(observed[Egt5], sum(observed[!Egt5]))
E.Pois;O.Pois
```

```
##  [1]  12.764224  74.595099 217.969724 424.610563 620.364128 725.091055
##  [7] 706.247903 589.623801 430.725807 279.688274 163.451960  86.838702
## [13]  42.291000  19.011679   7.936122   4.789959
```

```
##   0   1   2   3   4   5   6   7   8   9  10  11  12  13  14
## 635 520 493 368 362 275 268 227 190 173 122 108 102  92  70 401
```

```
J = length(E.Pois);J
```

```
## [1] 16
```

```
p=1;p
```

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

The length of new expected/observed counts is 16, and there is only one single parameter in poisson distribution, which is lambda.

1f

```
x_square <- sum((O.Pois-E.Pois)^2/E.Pois);x_square
```

```
## [1] 68043.05
```

```
1-pchisq(sum((O.Pois-E.Pois)^2/E.Pois),J-p)
```

```
## [1] 0
```

Since the x-square is fairly far away from degree of freedom, resulting the p-value to be extremely small(close to 0), so we are able to reject the assumption of model adequacy. Hence the poisson model does not fit these data properly.

Question 2a

```
fit2 <- glm.nb(visits~1, data = Visits.df)
mean <- exp(fit2$coefficients[1]);mean
```

```
## (Intercept)
##    5.844076
```

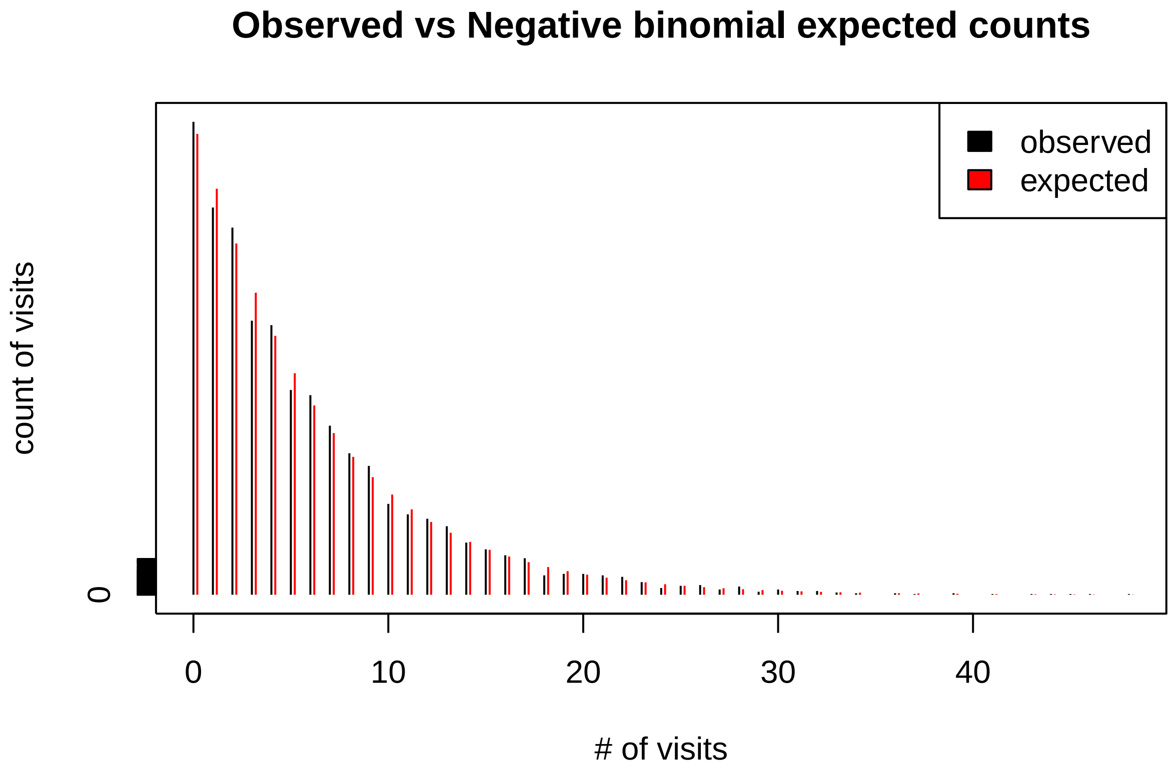
```
theta <- fit2$theta;theta
```

```
## [1] 1.037434
```

The estimate of the mean is 5.84 and theta is around 1.04.

2b

```
x=as.numeric(names(observed))
expected.nb=n*dnbinom(x, mu = mean, size = theta)
plot(x,observed, type="h",lwd=1, lend="butt", xlab="# of visits",
     ylab="count of visits",
     main="Observed vs Negative binomial expected counts",
     xlim=range(x), ylim= c(0, max(observed,expected.nb)))
lines(x+.2, expected.nb,type="h",
      lwd=1, lend="butt",col="red")
legend("topright", fill=c("black","red"),
       legend=c("observed","expected"))
```



It is obvious that the distribution of observed and expected is roughly matched, that is, amongst paired vertical line(two adjacent black and red lines) have approximately same height.

Hence, it is likely that negative binomial model is adequate for fitting these data.

2c

```
prop_nb <- 1 - pnbinom(12, mu = mean, size = theta);prop_nb
```

```
## [1] 0.1267397
```

```
prop_obs
```

```
## [1] 0.1277803
```

```
var_nb <- mean+mean^2/theta;unname(var_nb)
```

```
## [1] 38.76495
```

```
var_obs = var(Visits.df$visits);var_obs
```

```
## [1] 37.84379
```

Both variance(37.8 vs 38.8) and the proportion of visits that exceed 12(0.128 vs 0.127) for observed and negative binomial distributions are approximately same, which we may conclude that negative binomial model is adequate for fitting these data.

2d

```
expect_nb=expected.nb>=5
E.nb=c(expected.nb[expect_nb], sum(expected.nb[!expect_nb]));E.nb
```

```
##  [1] 618.815040 545.196950 471.670886 405.561547 347.643668 297.444402
##  [7] 254.178643 217.013857 185.159931 157.899863 134.597369 114.694904
## [13]  97.707727  83.216567  70.860074  60.327620  51.352691  43.706959
## [19]  37.195034  31.649867  26.928745  22.909821  19.489117  16.577939
## [25]  14.100663  11.992824  10.199489   8.673856   7.376062   6.272157
## [31]   5.333234  22.657596
```

```
O.nb=c(observed[expect_nb], sum(observed[!expect_nb]));O.nb
```

```
##   0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19
## 635 520 493 368 362 275 268 227 190 173 122 108 102  92  70  61  53  49  26  28
##  20  21  22  23  24  25  26  27  28  29  30
##  28  26  24  17   9  12  13   7  11   4   7  26
```

```
J <- length(O.nb);J
```

```
## [1] 32
```

```
p <- 2;p
```

```
## [1] 2
```

The new expected/observed counts is 32, and there are two parameters for the negative binomial distribution mu and theta.

2e

```
x_square <- sum((O.nb-E.nb)^2/E.nb);x_square
```

```
## [1] 26.35574
```

```
p_value <- 1-pchisq(x_square, df = J-p);p_value
```

```
## [1] 0.6568566
```

Since the x_square(26.4) is close to the degree of freedom(30), so the p-value(0.66) is statistically significant.

We can finally conclude that the negative binomial model is adequate for fitting these data.

Question 3

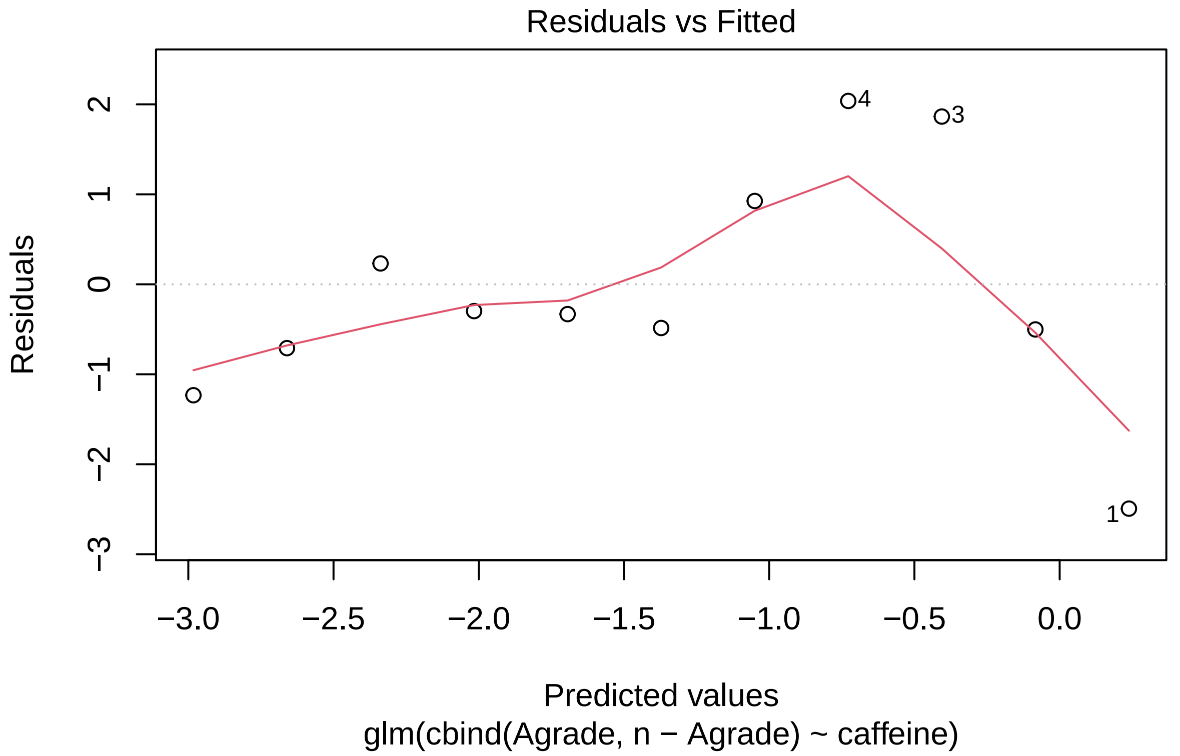
```
mod.null=glm(cbind(Agrade,n-Agrade)~1, family=binomial, data = Caffeine.df)
mod1=glm(cbind(Agrade,n-Agrade)~caffeine, family=binomial, data = Caffeine.df)
summary(mod1)
```

```
##
## Call:
## glm(formula = cbind(Agrade, n - Agrade) ~ caffeine, family = binomial,
##     data = Caffeine.df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4916  -0.6411  -0.3382   0.5642   1.9717
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.238469   0.226199   1.054    0.292
## caffeine    -0.006442   0.001009  -6.381 1.75e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 69.358  on 10  degrees of freedom
## Residual deviance: 18.625  on  9  degrees of freedom
## AIC: 55.87
##
## Number of Fisher Scoring iterations: 4
```

```
1-pchisq(deviance(mod1), df.residual(mod1))
```

```
## [1] 0.0285817
```

```
plot(mod1, which=1)
```



The p-value for the mod1 is around 0.029, which is not good enough to fit these data, it also can be prove by observing residual plot, because it is a clear curvature and non-constant variance.

3b

```
LLcaffeine=function(p,n=Caffeine.df$n, y=Caffeine.df$Agrade){
out=y*log(p)+(n-y)*log(1-p)
out[is.na(out)]=0
out
}
ps <- LLcaffeine(Caffeine.df$Agrade/Caffeine.df$n);ps
```

```
##  [1] -19.095425 -20.526953 -20.526953 -20.794415 -19.095425 -13.516836
##  [7] -11.780234  -9.752489  -9.752489  -4.384342   0.000000
```

3c

```
ave <- sum(Caffeine.df$Agrade/Caffeine.df$n)/11;ave
```

```
## [1] 0.2454545
```

```
p0 <- LLcaffeine(ave);p0
```

```
##  [1] -19.679230 -23.048241 -27.540256 -25.294249 -19.679230 -14.064211
##  [7] -12.941208 -11.818204 -11.818204  -9.572196  -8.449193
```

3d

```
sum(ps)
```

```
## [1] -149.2256
```

```
sum(p0)
```

```
## [1] -183.9044
```

```
dev_null <- 2*(sum(ps)-sum(p0));dev_null
```

```
## [1] 69.35772
```

3e

```
preds <- mod1$fitted.values
preds
```

```
##          1          2          3          4          5          6          7
## 0.55933643 0.47910903 0.39994494 0.32568285 0.25925228 0.20230652 0.15524759
##          8          9         10         11
## 0.11752241 0.08800938 0.06535894 0.04822963
```

3f

```
dev_res <- 2*(sum(ps)-sum(LLcaffeine(preds)));dev_res
```

```
## [1] 18.62452
```

3g

```
pearson_red=(Caffeine.df$Agrade-Caffeine.df$n*preds)/sqrt(Caffeine.df$n*preds*(1-preds))
pearson_red
```

```
##          1          2          3          4          5          6          7
## -2.4933594 -0.5018859  1.8640483  2.0373760  0.9259154 -0.4859299 -0.3314440
##          8          9         10         11
## -0.2980178  0.2318156 -0.7097135 -1.2329671
```

```
residuals(mod1,"pearson")
```

```
##          1          2          3          4          5          6          7 
## -2.4933594 -0.5018859  1.8640483  2.0373760  0.9259154 -0.4859299 -0.3314440 
##          8          9         10         11 
## -0.2980178  0.2318156 -0.7097135 -1.2329671
```

3h

```
mod2 <- glm(cbind(Agrade,n-Agrade)~caffeine, family='quasibinomial', data =Caffeine.df)
summary(mod2)
```

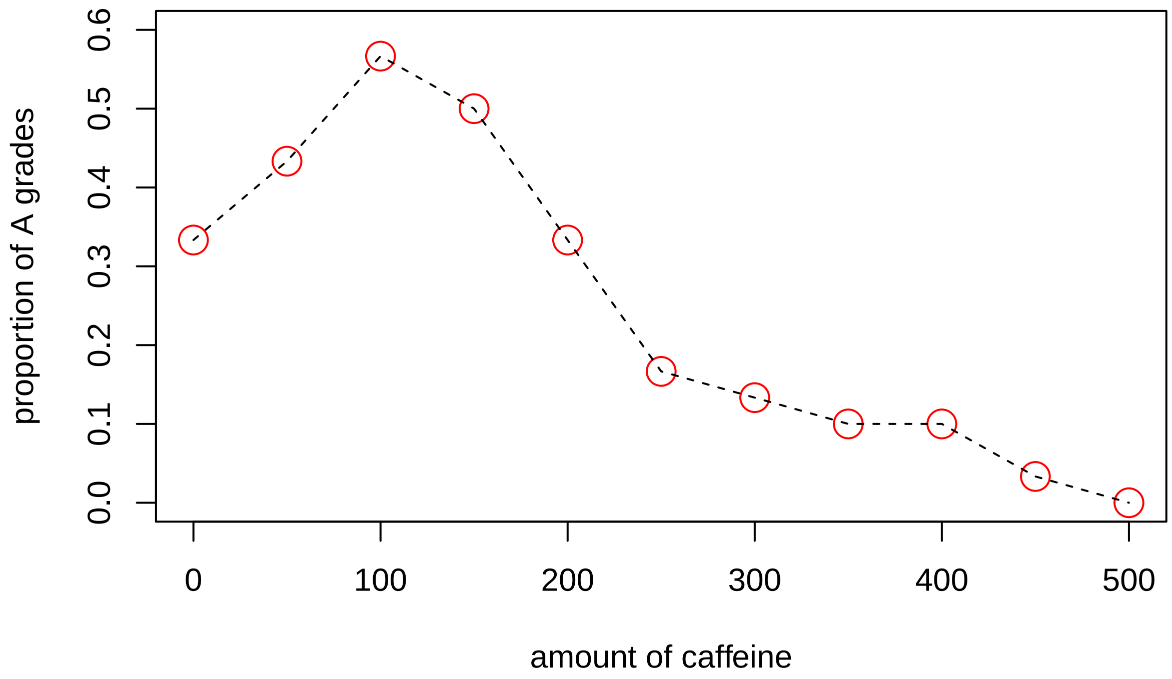
```
## 
## Call:
## glm(formula = cbind(Agrade, n - Agrade) ~ caffeine, family = "quasibinomial", 
##     data = Caffeine.df)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -2.4916  -0.6411  -0.3382   0.5642   1.9717  
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)   
## (Intercept)  0.238469   0.315095   0.757  0.46851   
## caffeine    -0.006442   0.001406  -4.581  0.00133 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for quasibinomial family taken to be 1.940452)
## 
##     Null deviance: 69.358  on 10  degrees of freedom
## Residual deviance: 18.625  on  9  degrees of freedom
## AIC: NA
## 
## Number of Fisher Scoring iterations: 4
```

```
sum(pearson_red^2)/mod2$df.residual
```

```
## [1] 1.940452
```

3i

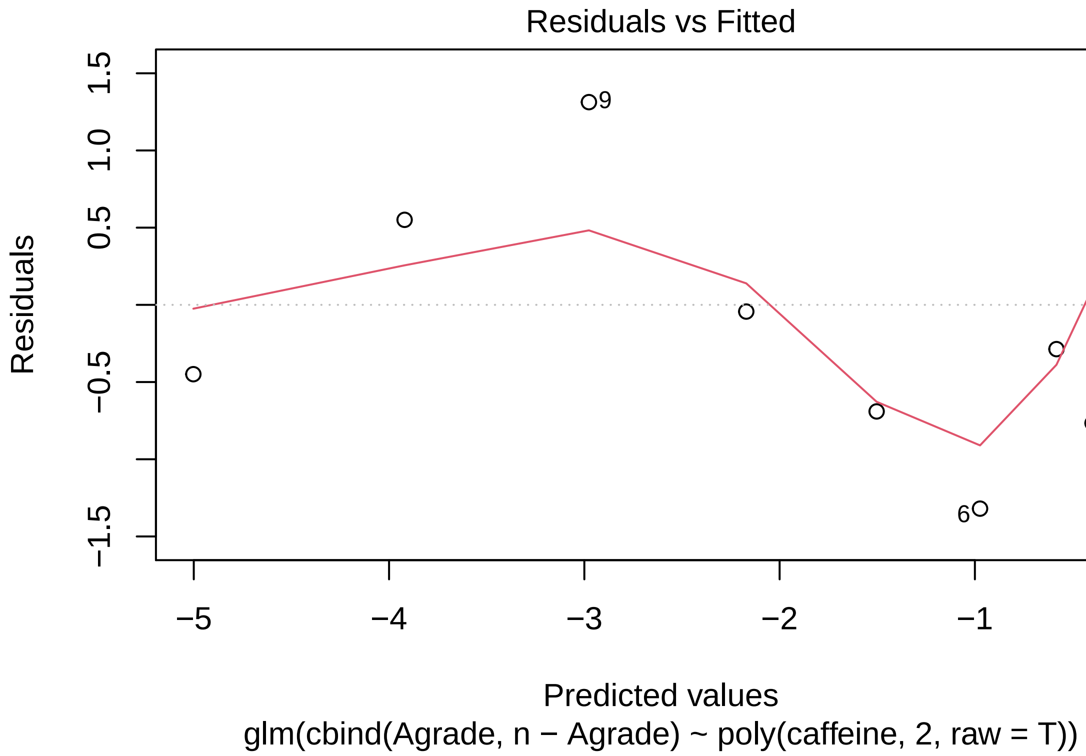
```
plot(Caffeine.df$caffeine, Caffeine.df$Agrade/Caffeine.df$n, xlab = 'amount of caffeine',
     ylab = 'proportion of A grades', ylim = c(0, 0.6),col='red',cex=2)
lines(Caffeine.df$caffeine, Caffeine.df$Agrade/Caffeine.df$n, lty=2)
```



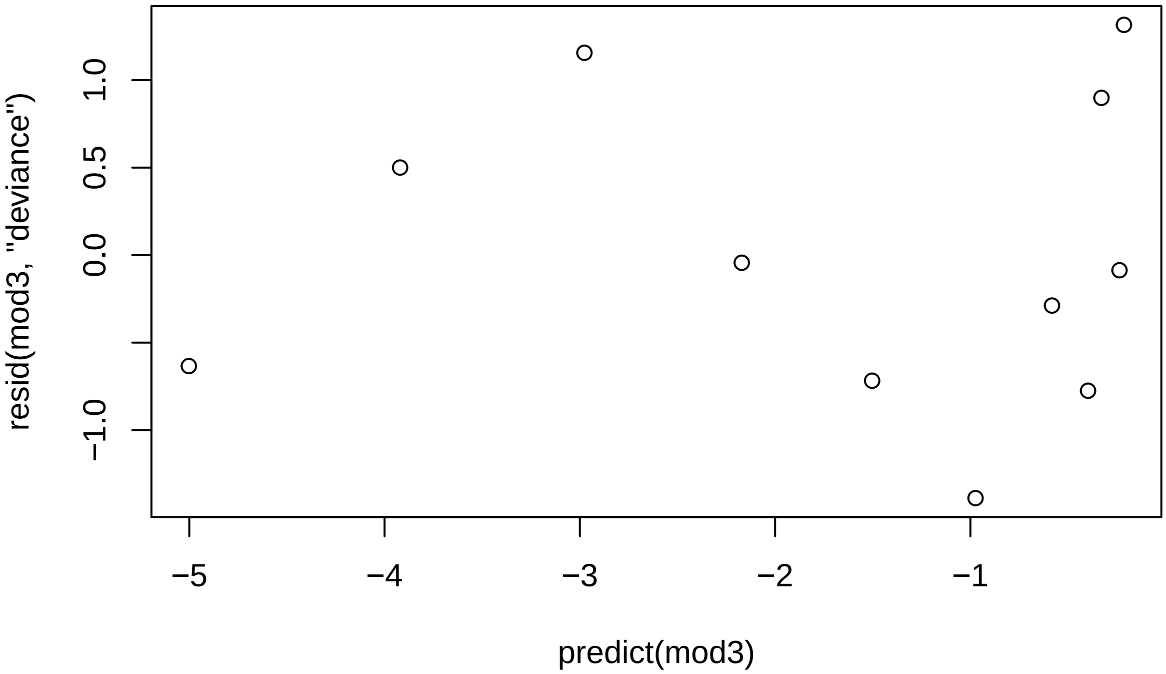
The optimal amount of caffeine for getting an A grade is around 100, since then it starts to decline gradually. The trend of the line is a curve, it may be a reason that mod1 have a curvature in its residual plot, then the quadratic term can be added to the mod1.

3j

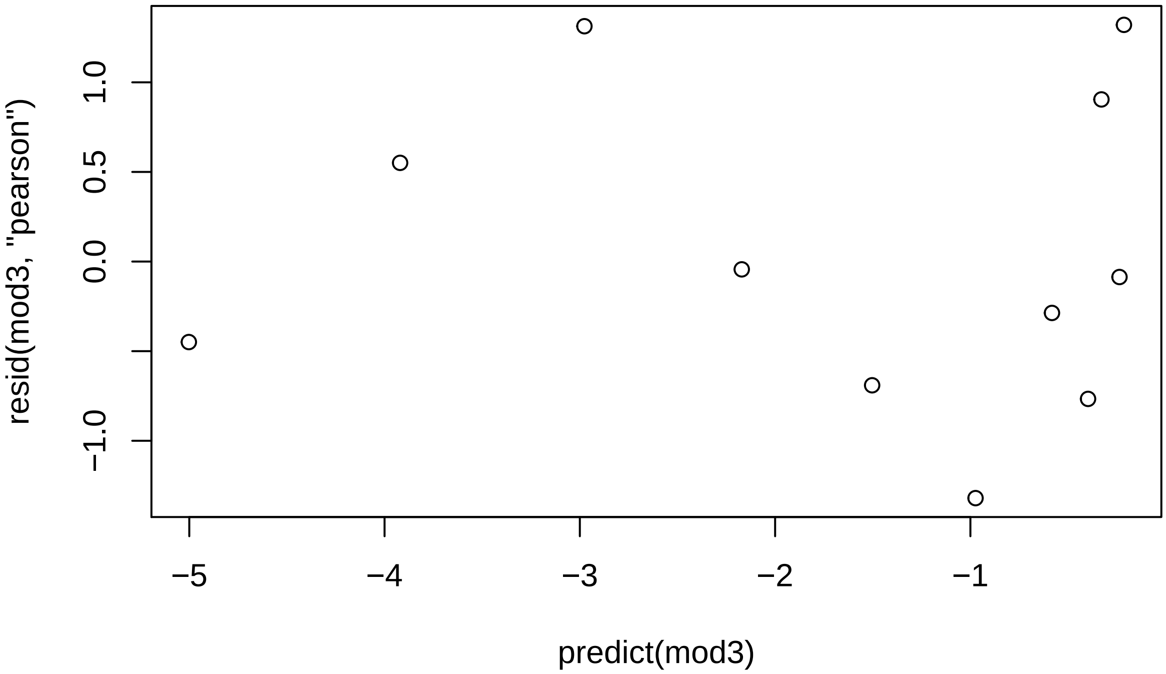
```
mod3 <- glm(cbind(Agrade,n-Agrade)~poly(caffeine, 2, raw = T), family='binomial', data =Caffeine.df)
plot(mod3,1)
```



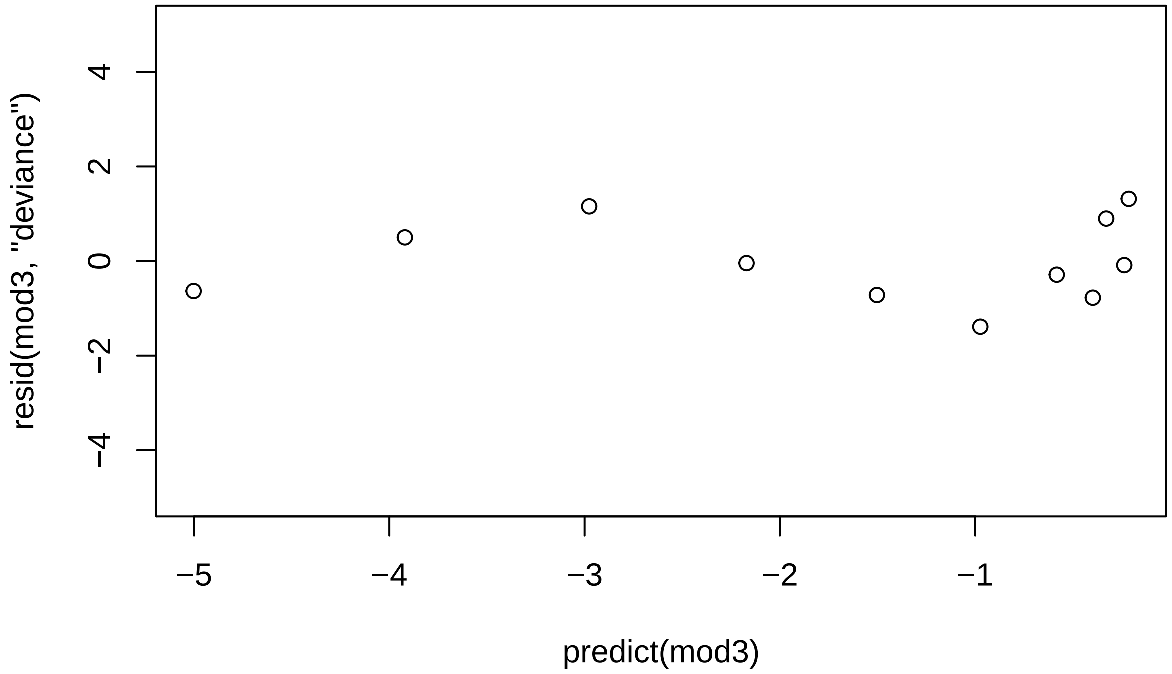
```
plot(predict(mod3), resid(mod3, 'deviance'))
```



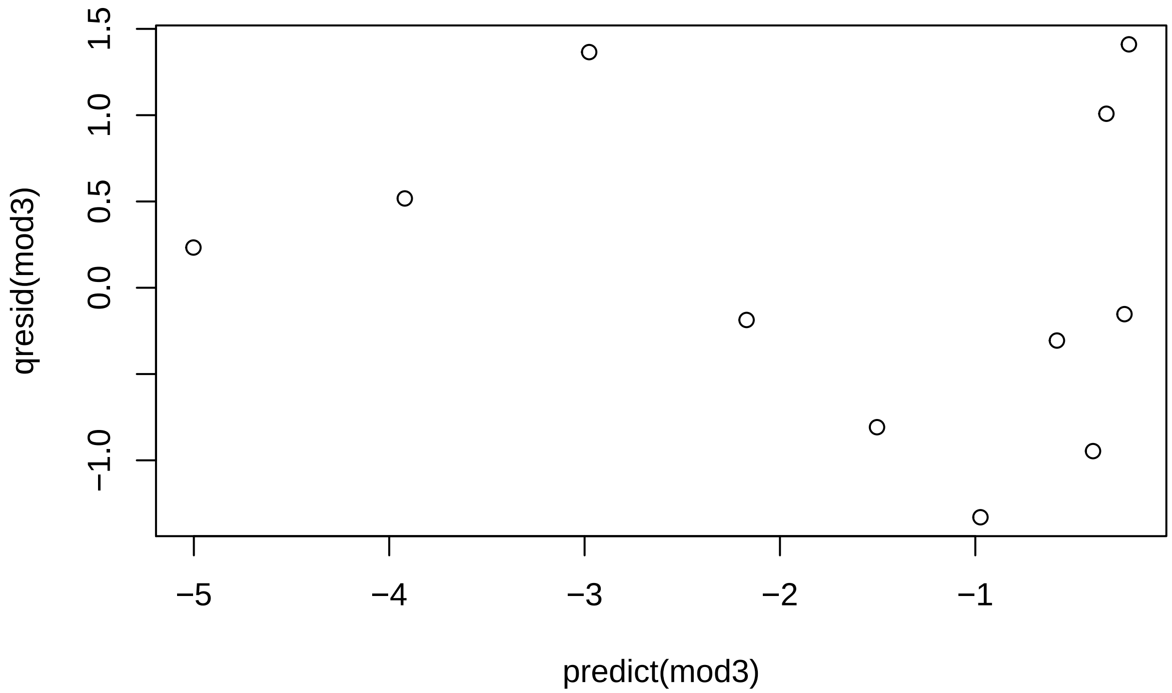
```
plot(predict(mod3), resid(mod3, 'pearson'))
```



```
plot(predict(mod3), resid(mod3, 'deviance'), ylim = c(-5, 5))
```



```
plot(predict(mod3), qresid(mod3))
```



```
summary(mod3)
```

```
##
## Call:
## glm(formula = cbind(Agrade, n - Agrade) ~ poly(caffeine, 2, raw = T),
##     family = "binomial", data = Caffeine.df)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.38859  -0.67591  -0.08634   0.69945   1.31565
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -3.974e-01  3.021e-01  -1.315  0.18836
## poly(caffeine, 2, raw = T)1  4.600e-03  3.633e-03   1.266  0.20538
## poly(caffeine, 2, raw = T)2 -2.762e-05  9.257e-06  -2.984  0.00285 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 69.3577  on 10  degrees of freedom
## Residual deviance:  7.6639  on  8  degrees of freedom
## AIC: 46.909
##
## Number of Fisher Scoring iterations: 5
```

```
1-pchisq(mod3$deviance, mod3$df.residual)
```

```
## [1] 0.4669742
```

```
exp(coef(mod3)[3])
```

```
## poly(caffeine, 2, raw = T)2
##                   0.9999724
```

```
exp(confint(mod3)[3,])
```

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

```
##     2.5 %    97.5 %
## 0.9999529 0.9999894
```

3k

As we have observed the plot in Q3i, it has a curve, then we refit the mod1 with an extra quadratic term, it turns out that the p-value have been increased to 0.467, which is adequate enough to describe these data, even though our pearson and deviance residual is still not perfect, but the randomised quantile residual plot does remove some problem of spare data.

Our final model is: log(odd)¡ = ß0 + ß1 * caffeine¡ + ß2 * caffeine¡^2 where log(odd)¡ is the odds of obtaining A grade.