Assignment 2 STATS 330

Anish Hota

2025-04-13

Mean and Vairance Analysis

```
Visits.df <- read.csv("Visits.csv")
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
```

As we can see by th output the variance is much larger than the mean, so there is over dispersion. Since the Poisson model assumes that the mean equals variance, this model is not a good fit for this set of data.

Fitting Poisson Model

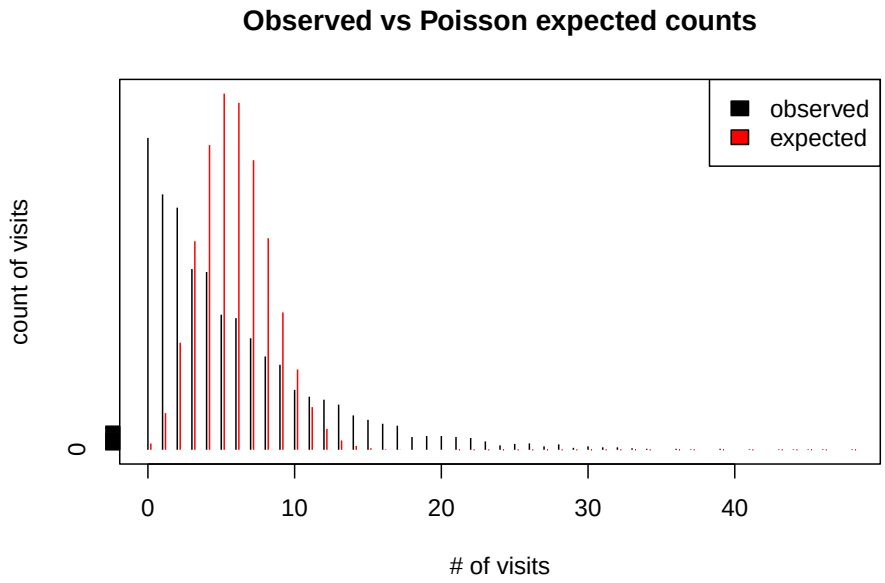
```
fit1<- glm(visits~1,family=poisson,data=Visits.df)
```

The intercept is $\beta_0 = log(\mu)$.

Poisson Graph

```
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"))
```



As you can see the red and blue bars don't match closely meaning that this model doesn't fit well.

Poisson Variance

```
poi_variance <- exp(coef(fit1))
poi_tail_probability <- ppois(12,lambda = poi_variance, lower.tail = FALSE)
poi_tail <- sum(Visits.df$visits >12)/n
poi_variance
```

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

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

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

```
poi_tail_probability
```

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

```
poi_tail
```

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

Since the sample variance is larger than the poisson variance this suggests over dispersion. Since the predicted tail larger than 12 is larger than the observed it shows that there is a lack of fit.

Expected Counts for Poisson Model

```r
## inspecting the data>>=
Egt5=expected.Pois>=5
# note ! is R's way of saying NOT
E.Pois=c(expected.Pois[Egt5], sum(expected.Pois[!Egt5]))
O.Pois=c(observed[Egt5], sum(observed[!Egt5]))
E.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
```

```r
J <- length(E.Pois)
J
```

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

There is one parameter(the mean) so p = 1 and J = 16

Chi-Squared for Poisson

```r
chisq <- sum((O.Pois - E.Pois)^2/E.Pois)
p_value <- pchisq(chisq, df = J - 1, lower.tail = FALSE)
chisq
```

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

```r
p_value
```

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

p-value is less than 0.05 so the Poisson model doesn't fit very well.

Negative Binomial Fit

```r
library(MASS)
fit2 <- glm.nb(visits ~ 1, data = Visits.df)
summary(fit2)
```

```
##
## Call:
## glm.nb(formula = visits ~ 1, data = Visits.df, init.theta = 1.037433909,
##     link = log)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.76543    0.01605     110   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(1.0374) family taken to be 1)
##
##     Null deviance: 5029.2  on 4405  degrees of freedom
## Residual deviance: 5029.2  on 4405  degrees of freedom
## AIC: 25085
##
```

```
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  1.0374
##           Std. Err.:  0.0272
##
##  2 x log-likelihood:  -25081.2590
```

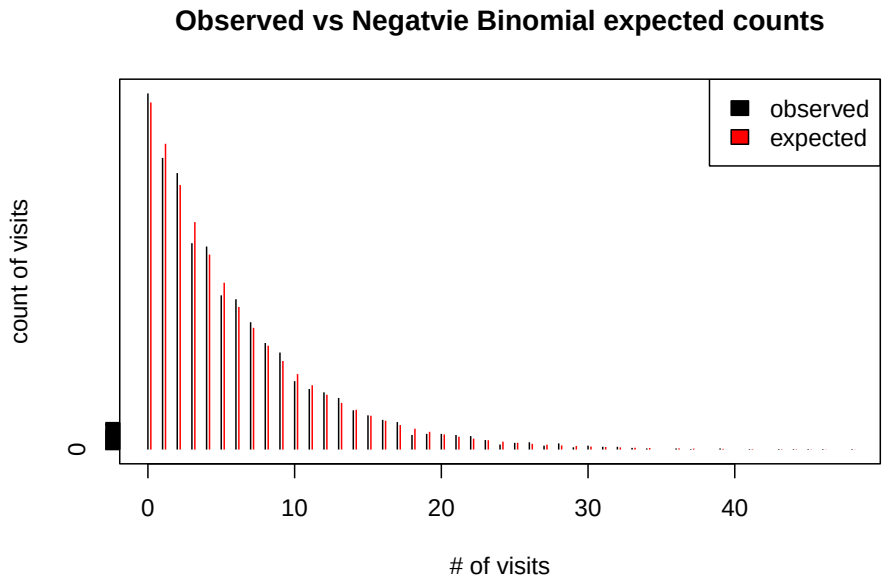
```
mean_nb <- exp(coef(fit2))
mean_nb
```

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

Theta is small so overdispersion is high.

Negative Binomial Graph

```
x=as.numeric(names(observed))
theta_hat <- fit2$theta
expected.nb=n*dnbinom(x, mu = mean_nb, size = theta_hat)
plot(x,observed, type="h",lwd=1, lend="butt", xlab="# of visits",
ylab="count of visits",
main="Observed vs Negatvie 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"))
```



The black and red bars in this model is very similar so the NB model is a much better fit.

Negtaive Binomial Variance

```
nb_variance <- mean_nb * (1+mean_nb/theta_hat)
nb_tail_probability <- pnbinom(12,mu = mean_nb,size = theta_hat ,lower.tail = FALSE)
nb_tail <- sum(Visits.df$visits >12)/n
nb_variance
```

```
## (Intercept)
##    38.76495
```

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

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

```
nb_tail_probability
```

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

```
nb_tail
```

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

The variances and tails are very similar to each other meaning that the NB model is much better fit

Expected Counts for NB Model

```
## inspecting the data>>=
Egt5_nb=expected.nb>=5
# note ! is R's way of saying NOT
E.nb=c(expected.nb[Egt5_nb], sum(expected.nb[!Egt5_nb]))
O.nb=c(observed[Egt5_nb], sum(observed[!Egt5_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
```

```
J_nb <- length(E.nb)
J_nb
```

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

p_nb = 2 cause we got mean and dispersion, and J = 32

Chi-Squared for NB

```
chisq_nb <- sum((O.nb - E.nb)^2/E.nb)
p_value_nb <- pchisq(chisq_nb, df = J_nb - 2, lower.tail = FALSE)
chisq_nb
```

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

```
p_value_nb
```

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

The p-value is very large so the NB model is a very good fit for this data and much better than the Poisson model.

Caffeine Model

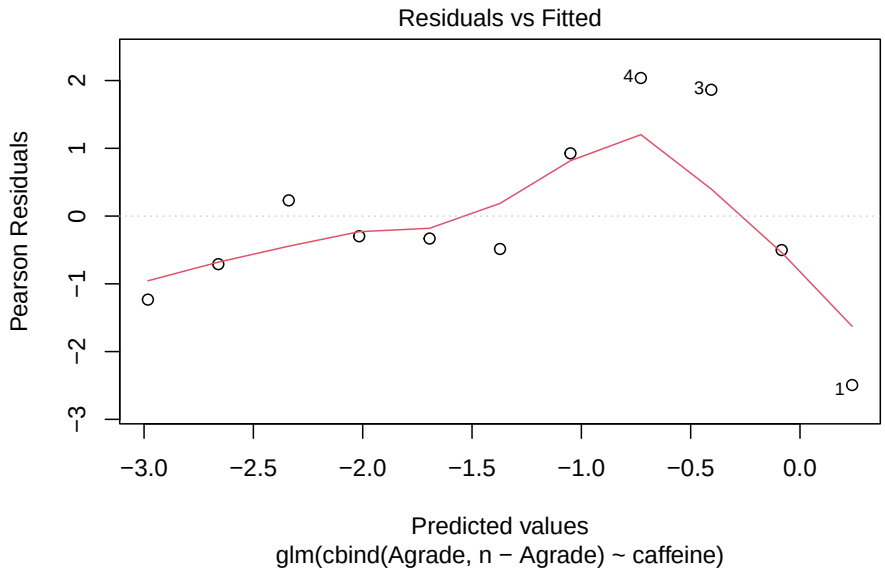
```
Caffeine.df=read.csv("Caffeine.csv")
## null model
mod.null=glm(cbind(Agrade,n-Agrade)~1, family=binomial, data =Caffeine.df)
## linear log-odds model
mod1=glm(cbind(Agrade,n-Agrade)~caffeine, family=binomial, data =Caffeine.df)
summary(mod1)
```

```
##
## Call:
## glm(formula = cbind(Agrade, n - Agrade) ~ caffeine, family = binomial,
##     data = Caffeine.df)
##
## 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)
```



We have a model for the log-odds of getting an A-grade based on the amount of caffeine consumed. We have a small p-value (less than 0.05) so this model isn't a good fit. The residuals are not constant so no EOV.

Likelihood Function and Saturated Model

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

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

Likelihhod Function in Null Model

```
p0 <- fitted(mod.null)
LL_null <- LLcaffeine(rep(p0[1], nrow(Caffeine.df)))
LL_null
```

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

Null Deviance

```
null_deviance <- (-2) * (sum(LL_null) - sum(LL_saturated))
null_deviance
```

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

```
mod1$null.deviance
```

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

Likelihood in Linear Caffeine Model

```
preds <- fitted(mod1)
LL_model <- LLcaffeine(preds)
LL_model
```

```
##          1          2          3          4          5          6          7
## -22.199513 -20.653399 -22.218823 -22.738289 -19.501440 -13.640628 -11.837441
##          8          9         10         11
##  -9.798970  -9.778326  -4.688050  -1.482944
```

Residual Deviance

```
residual_deviance <- (-2) * (sum(LL_model)-sum(LL_saturated))
residual_deviance
```

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

```
mod1$deviance
```

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

Pearson Residuals

```
observed <- Caffeine.df$Agrade
expected <- preds * Caffeine.df$n
residual_pearson <- (observed - expected) / sqrt(expected * (1 - preds))
residual_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
```

```
residuals(mod1, type = "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
```

Quasi-Binomial Model

```
mod2 <- glm(cbind(Agrade, n-Agrade) ~ caffeine, family = quasibinomial, data = Caffeine.df)
pearson_residuals <- residuals(mod1, type = "pearson")
dispersion_k <- sum(pearson_residuals^2) / df.residual(mod1)
summary(mod2)
```

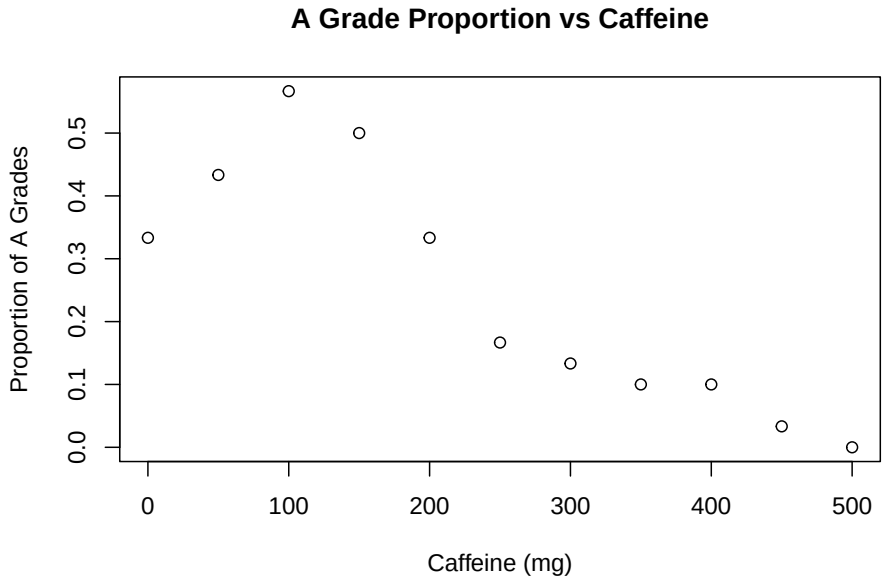
```
##
## Call:
## glm(formula = cbind(Agrade, n - Agrade) ~ caffeine, family = quasibinomial,
##     data = Caffeine.df)
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.238469   0.315095   0.757  0.46851
## caffeine    -0.006442   0.001406  -4.581  0.00133 **
## ---
## 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
```

```
dispersion_k
```

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

Plot of Proportions of A-Grades

```
A_grades <- Caffeine.df$Agrade / Caffeine.df$n
plot(Caffeine.df$caffeine, A_grades, xlab = "Caffeine (mg)", ylab = "Proportion of A Grades", main = "A
```



It seems that at around 100mg of Caffeine is the most optimal amount for getting an A Grade and once you start have more than this amount it gets lower and lower.

New Model

```
mod3 <- glm(cbind(Agrade, n-Agrade) ~ caffeine + I(caffeine^2), family = binomial, data = Caffeine.df)
summary(mod3)
```

```
##
## Call:
## glm(formula = cbind(Agrade, n - Agrade) ~ caffeine + I(caffeine^2),
##     family = binomial, data = Caffeine.df)
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -3.974e-01  3.021e-01  -1.315  0.18836
## caffeine       4.600e-03  3.633e-03   1.266  0.20538
## I(caffeine^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(deviance(mod3), df.residual(mod3))
```

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

```
anova(mod1, mod3, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: cbind(Agrade, n - Agrade) ~ caffeine
## Model 2: cbind(Agrade, n - Agrade) ~ caffeine + I(caffeine^2)
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1         9    18.6245
## 2         8     7.6639  1   10.961 0.0009307 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value is very low for the squared term so it is significant and a better fit for this data. This allows for a parabola shaped graph to exist.

Executive Summary

The first model used was a linear model which seemed to have poor fit for the data given. There was over dispersion and lack of consistent residuals. After using a quasi-binomial model however we had a much better fit and could easily compare the A grades to the caffeine intake, we were able to see that at 100mg of caffeine, this was the most potimal point of getting an A grade.