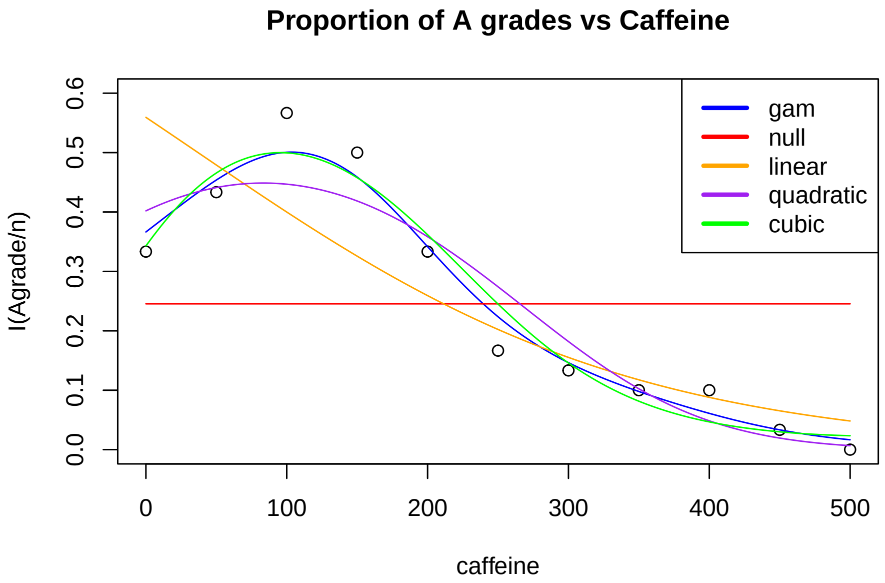
STATS 330 Assignment 3

Anish Hota

2025-05-15

1A

```r
## null model (order 0)
mod.0=glm(cbind(Agrade,n-Agrade)~1, family=binomial,
data =Caffeine.df)
## linear (order 1)
mod.1=glm(cbind(Agrade,n-Agrade)~caffeine,
family=binomial, data =Caffeine.df)
## quadratic (order 2)
mod.2=glm(cbind(Agrade,n-Agrade)~caffeine+I(caffeine^2),
family=binomial, data =Caffeine.df)
## cubic (order 2)
mod.3=glm(cbind(Agrade,n-Agrade)~caffeine +I(caffeine^2)+I(caffeine^3),
family=binomial, data =Caffeine.df)
mod.gam=gam(cbind(Agrade,n-Agrade)~s(caffeine),
family=binomial, data =Caffeine.df)
# look at null, order 1 and GAM fits (adapt this below )
plot(I(Agrade/n)~caffeine, ylim=c(0,.6),
main ="Proportion of A grades vs Caffeine", data=Caffeine.df)
# add lines
caffs=seq(0, 500, by=1)
new.df=data.frame(caffeine=caffs)
p.gam=predict(mod.gam, newdata=new.df,type="response")
lines(caffs, p.gam, col="blue")
p0=predict(mod.0, newdata=new.df, type="response")
lines(caffs, p0, col="red")
p1=predict(mod.1, newdata=new.df, type="response")
lines(caffs, p1, col="orange")
p2=predict(mod.2, newdata=new.df, type="response")
lines(caffs, p2, col="purple")
p3=predict(mod.3, newdata=new.df, type="response")
lines(caffs, p3, col="green")
legend('topright', lty=1,lwd=3, col=c("blue", "red","orange","purple","green") ,
legend=c("gam", "null","linear","quadratic","cubic"))
```



The GAM model seems to fit the data the best, with cubic and quadratic not too far off. It seems that the higher the order of the model the better fit it has on the data.

1B

```
anova(mod.0, mod.1, mod.2, mod.3, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: cbind(Agrade, n - Agrade) ~ 1
## Model 2: cbind(Agrade, n - Agrade) ~ caffeine
## Model 3: cbind(Agrade, n - Agrade) ~ caffeine + I(caffeine^2)
## Model 4: cbind(Agrade, n - Agrade) ~ caffeine + I(caffeine^2) + I(caffeine^3)
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1        10     69.358
## 2         9     18.625  1   50.733 1.058e-12 ***
## 3         8      7.664  1   10.961 0.0009307 ***
## 4         7      5.145  1    2.519 0.1125092
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This test shows that the linear and quadratic model improve the fit of the graph with a small p-value but the cubic is not taht significant of a fit. So by Occam's razor the quadratic model is the best fit for the least number of terms needed.

1C

```
AIC(mod.0, mod.1, mod.2, mod.3, mod.gam)
```

```
##               df       AIC
## mod.0   1.000000 104.60325
## mod.1   2.000000  55.87004
## mod.2   3.000000  46.90939
## mod.3   4.000000  46.39077
## mod.gam 4.268157  45.03682
```

The GAM model has the lowest AIC do it is the best model, but there is not much difference between the quadratic and cubic models, so thw quadratic model is probably the best, as it still has a small AIC while not being as complex as the other two models.

1D

```
library(MuMIn)
options(na.action = "na.fail")
msubset <- expression(dc(caffeine, `I(caffeine^2)`, `I(caffeine^3)`))
all.fits <- dredge(mod.3, subset=msubset)
```

```
## Fixed term is "(Intercept)"
```

```
all.fits
```

```
## Global model call: glm(formula = cbind(Agrade, n - Agrade) ~ caffeine + I(caffeine^2) +
##     I(caffeine^3), family = binomial, data = Caffeine.df)
## ---
## Model selection table
##    (Intrc)     caffn    caffn^2   caffn^3 df  logLik  AICc delta weight
## 4 -0.3974  0.004600 -2.762e-05            3 -20.455  50.3  0.00  0.777
## 8 -0.6506  0.014510 -8.991e-05 9.714e-08  4 -19.195  53.1  2.72  0.200
## 2  0.2385 -0.006442                       2 -25.935  57.4  7.03  0.023
## 1 -1.1230                                 1 -51.302 105.0 54.71  0.000
## Models ranked by AICc(x)
```

The lowest AIC is when caffeine and I(caffeine^2) is included and not the cubic term, which corresponds with the ANOVA test, meaning that this function suggests that we should use the quadratic model as well.

1E

After all these tests, its fair to say the the best model to fit this data is the quadratic model. It is the most optimal model for this data due to its low ACC as well as its not as complex as the cubic or GAM models.

1F

```
round(cor(model.matrix(mod.3)[,-1]),3)
```

```
##               caffeine I(caffeine^2) I(caffeine^3)
## caffeine         1.000         0.963         0.909
## I(caffeine^2)    0.963         1.000         0.986
## I(caffeine^3)    0.909         0.986         1.000
```

```
mod.3a=glm(cbind(Agrade,n-Agrade)~poly(caffeine,3),
family=binomial, data=Caffeine.df)
preds1 <- predict(mod.3, newdata = new.df, type = "response")
preds2 <- predict(mod.3a, newdata = new.df, type = "response")
all.equal(preds1, preds2)
```

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

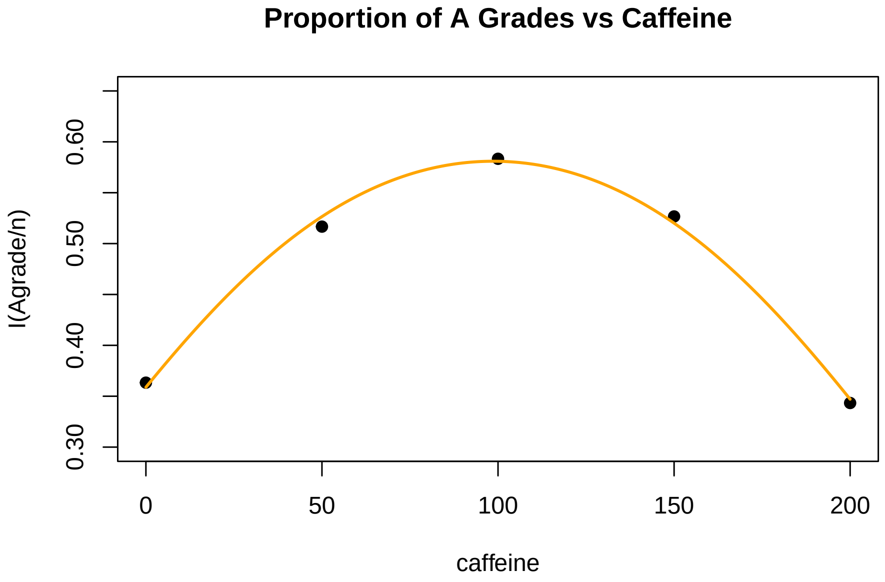
```
round(cor(model.matrix(mod.3a)[,-1]),3)
```

```
##                    poly(caffeine, 3)1 poly(caffeine, 3)2 poly(caffeine, 3)3
## poly(caffeine, 3)1                  1                  0                  0
## poly(caffeine, 3)2                  0                  1                  0
## poly(caffeine, 3)3                  0                  0                  1
```

The predictions for both models shows that these two models are identical, this new model also solves the MC problem as we now have orthogonal polynomials, as seen in the table above, it has dropped the redundant predictors.

2A

```
Caffeine2.df <- data.frame(caffeine = c(0, 50, 100, 150, 200),Agrade = c(109, 155, 175, 158, 103),n = re
mod.quad <- glm(cbind(Agrade, n - Agrade) ~ caffeine + I(caffeine^2), family = binomial, data = Caffeine
plot(I(Agrade/n) ~ caffeine, data = Caffeine2.df, main = "Proportion of A Grades vs Caffeine", ylim = c

xvals <- seq(0, 200, by = 1)
newdata <- data.frame(caffeine = xvals)
preds <- predict(mod.quad, newdata = newdata, type = "response")
lines(xvals, preds, col = "orange", lwd = 2)
```



As seen in the graph the quadratic model fits very well as the data is curved and the line aligns with points.

2B

```
b <- coef(mod.quad)
x_peak <- -b[2] / (2 * b[3])
x_peak
```

```
## caffeine
## 98.61706
```

The caffeine level that maximizes the probability of an A-Grade is 98.62mg, which seems reasonable as it is between 90 - 120 mg.

2C

```
Delta.g <- c(0, -1 / (2 * b[3]), b[2] / (2 * b[3]^2))
Delta.g
```

```
##               I(caffeine^2)      caffeine
##         0.000      5360.597   1057292.592
```

This is the vector for the variance estimation and is used for uncertainty

2D

```r
Varx_peak <- t(Delta.g) %*% vcov(mod.quad) %*% Delta.g
Varx_peak
```

```
##          [,1]
## [1,] 16.50362
```

We have a variance of 16.50 which we can use to find the standard deviation estimate.

2E

```r
CI_lower <- x_peak - 1.96 * sqrt(Varx_peak)
CI_upper <- x_peak + 1.96 * sqrt(Varx_peak)
c(CI_lower, CI_upper)
```

```
## [1]  90.65463 106.57949
```

This seems reasonable as our previous estimate for x_peak falls within the range. The interval is relatively narrow meaning it is somewhat precise.

3A

```r
ns=Caffeine2.df$n
xs=Caffeine2.df$caffeine
preds <- predict(mod.quad, type = "response")
ys=rbinom(length(ns),size=ns, prob=preds)
ys
```

```
## [1] 110 148 174 169 115
```

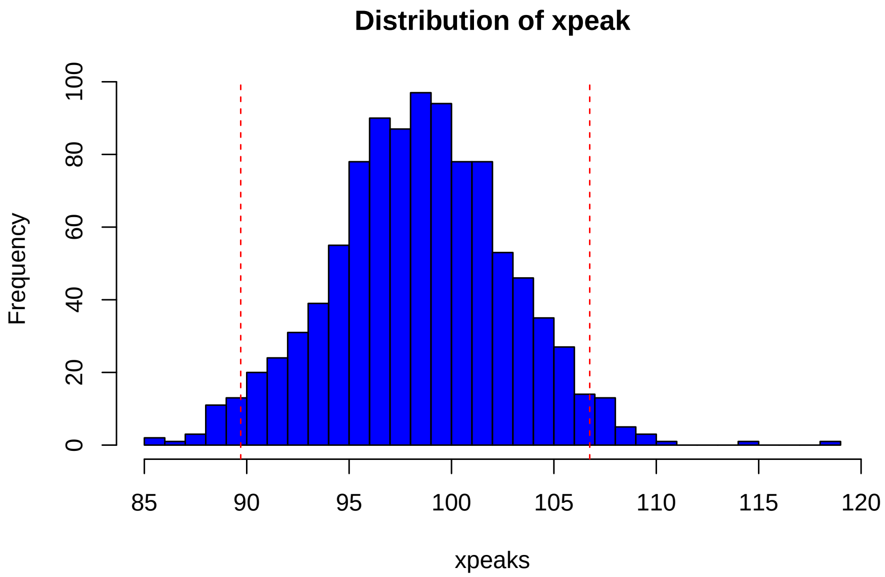
These value are fairly similar to the actual A-grade values but still not the exactly same.

3B

```r
xpeaks <- numeric(1000)
devs <- numeric(1000)

for (i in 1:1000) {
  ysim <- rbinom(length(ns), size = ns, prob = preds)
  temp.df <- Caffeine2.df
  temp.df$Agrade <- ysim

  mod.sim <- glm(cbind(Agrade, n - Agrade) ~ caffeine + I(caffeine^2),
                 family = binomial, data = temp.df)

  b.sim <- coef(mod.sim)
  xpeaks[i] <- -b.sim[2] / (2 * b.sim[3])
  devs[i] <- deviance(mod.sim)
}
```

3C

```r
hist(xpeaks, breaks = 40, col = "blue", main = "Distribution of xpeak")
abline(v = quantile(xpeaks, c(0.025, 0.975)), col = "red", lty = 2)
```



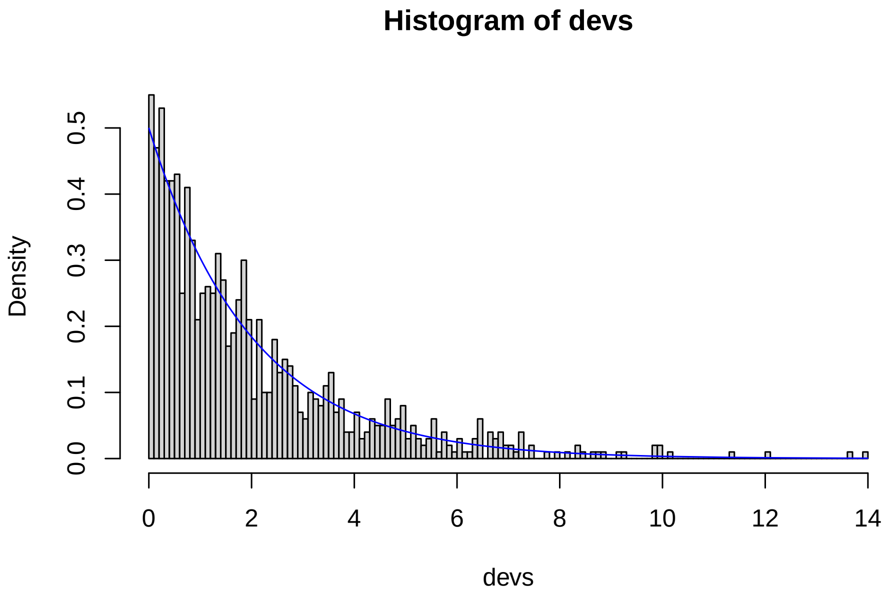
```r
quantile(xpeaks, c(0.025, 0.975))
```

```
##      2.5%     97.5%
##  89.70675 106.74853
```

The confidence intervals are very similar to the intervals obtained in 2E. The histogram is relatively bell shaped meaning that it is approximately normal.

3D

```r
hist(devs,breaks=100,prob=T)
dvs=sort(devs)
lines(dvs, dchisq(dvs,df = 2),col="blue")
```



The Chi-squared curve fits the data relatively well, so this model is appropriate for the data given.