A3

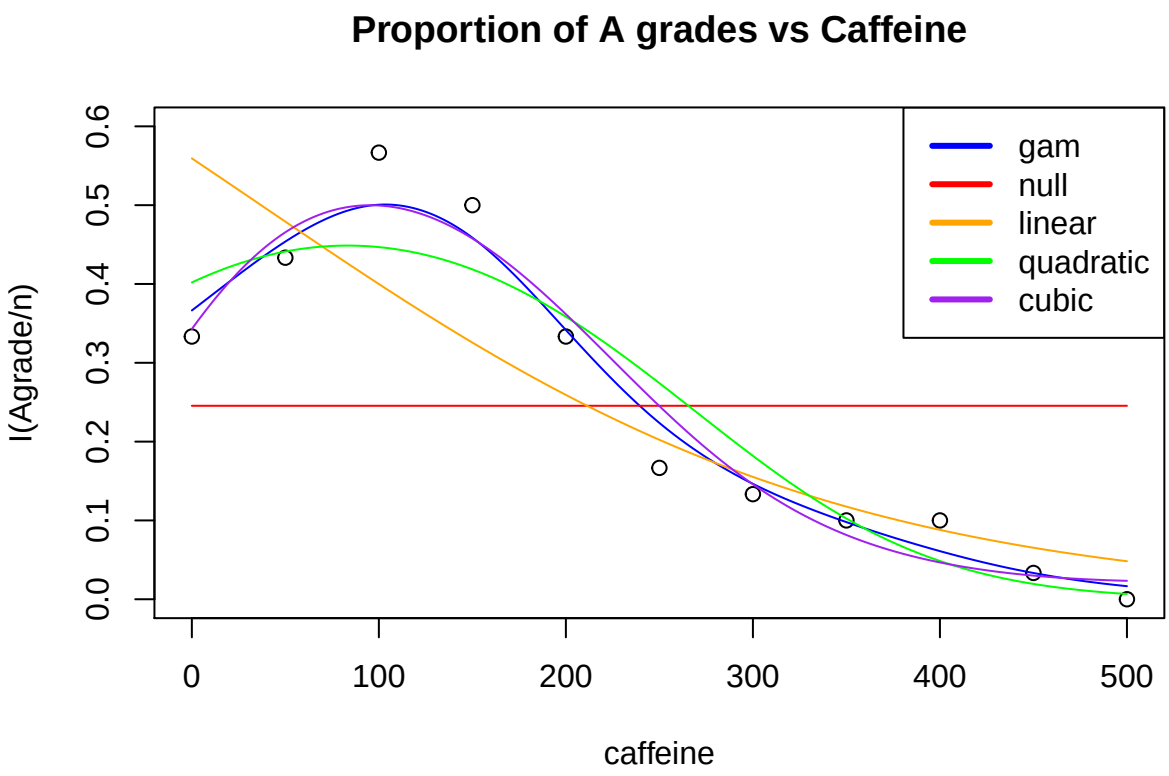
Kerui Du

2021-09-30

Q1

1a

```
## 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)

## gam
mod.gam=gam(cbind(Agrade,n-Agrade)~s(caffeine),
            family=binomial, data =Caffeine.df)

# look at null, order 1,2,3 and gam models
caffs=seq(0, 500, by=1)
new.df=data.frame(caffeine=caffs)
p0=predict(mod.0, newdata=new.df, type="response")
p1=predict(mod.1, newdata=new.df, type="response")
p2=predict(mod.2, newdata=new.df, type="response")
p3=predict(mod.3, newdata=new.df, type="response")
p.gam=predict(mod.gam, newdata=new.df,type="response")

# plot and with 5 different lines
plot(I(Agrade/n)~caffeine, ylim=c(0,.6),
     main ="Proportion of A grades vs Caffeine", data=Caffeine.df)
lines(caffs, p.gam, col="blue")
lines(caffs, p0, col="red")
lines(caffs, p1, col="orange")
lines(caffs, p2, col="green")
lines(caffs, p3, col="purple")
legend('topright', lty=1,lwd=3, col=c("blue", "red","orange",'green','purple'),
       legend=c("gam", "null","linear",'quadratic','cubic'))
```



Based on the plot, I'd say both the gam and cubic model fits the data best since either two lines are close or overlapped, and the trend of these two lines pretty much along with most of data. On the contrary, the rest of the three lines seems does not describe the pattern very well. Therefore, gam and cubic fit the data best.

1b

```
test1=anova(mod.0,mod.1,test = 'Chisq')
test2=anova(mod.1,mod.2,test = 'Chisq');test2
```

```
## 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
```

```
test3=anova(mod.2,mod.3,test = 'Chisq');test3
```

```
## Analysis of Deviance Table
##
```

```
## Model 1: cbind(Agrade, n - Agrade) ~ caffeine + I(caffeine^2)
## Model 2: cbind(Agrade, n - Agrade) ~ caffeine + I(caffeine^2) + I(caffeine^3)
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1         8     7.6639
## 2         7     5.1452  1   2.5186   0.1125
```

```
coef(summary(mod.3));coef(summary(mod.2))
```

```
##                    Estimate   Std. Error   z value   Pr(>|z|)
## (Intercept)   -6.505864e-01 3.478606e-01 -1.870250 0.06144905
## caffeine       1.450848e-02 7.226071e-03  2.007797 0.04466492
## I(caffeine^2) -8.990914e-05 3.980423e-05 -2.258784 0.02389685
## I(caffeine^3)  9.713504e-08 5.879389e-08  1.652128 0.09850842
```

```
##                    Estimate   Std. Error   z value    Pr(>|z|)
## (Intercept)   -0.3974368048 3.021274e-01 -1.315461 0.188355049
## caffeine       0.0046002018 3.632605e-03  1.266364 0.205382646
## I(caffeine^2) -0.0000276194 9.257295e-06 -2.983528 0.002849461
```

By mathematical definition, we know that the null model is the submodel of the linear model. When β_1 is 0, then the null model is the same as the linear model. The linear model is the submodel of the quadratic model; when β_2 is 0, then the linear model is the same as the quadratic model. The quadratic model is the submodel of a cubic model; when $\beta_3 = 0$, then the quadratic model is the same as the cubic model. However, when we inspect test3(above), we realize that the p-value is not statistically significant, which means we may have no evidence to reject the null hypothesis. That is, the submodel is appropriate. There is no cubic effect of grade by caffeine. Alternatively, we can see that the p-value of the cubic term in the cubic model is above 0.05, which suggests we need to get rid of it. For test2, we have strong evidence against the null hypothesis. Overall, the quadratic model best fits these data with the least number of terms used.

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
```

As we can see, the difference between mod.0 and the rest of the models is quite significant, and the difference between mod.1 to mod.2/mod.3/mod.gam is around 10, then it is sensible to consider mod.2/mod.3/mod.gam rather than mod.0 and mod.1, since there is a slightly different (within 2) amongst mod.2/mod.3/mod.gam. Hence, it is reasonable to pick mod.2 as the best model because it has a smaller df(less complexity) than mod.3 and mod.gam, and its AIC is good enough compared with mod.3/mod.gam.

1d

```r
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)"
```

```r
# adaption
head(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)
```

```r
first.model <- get.models(all.fits, 1)[[1]]
summary(first.model)
```

```
##
## Call:
## glm(formula = cbind(Agrade, n - Agrade) ~ caffeine + I(caffeine^2) +
##     1, 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
## 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
```

It seems that the quadratic model is the best model to fit the data, since it has significant term(I(caffeine^2)) on its model and also it owns the relatively smaller AICc value.

1e

I would propose mod.2 to be the best model. Here is a couple of reasons why I pick this model. First of all, the significant term, it's apparent that cubic term is not significant at the cubic model, whereas, quadratic term is significant at the quadratic model, in which suggests us the cubic term is not appropriate, secondly when we apply the anova test, it also suggests us that there may not have the cubic effect. Finally, AICc(because the dataset only has a few observations, use AICc instead of AIC) advises us to trust our intuition, as it has less complexity and smaller AICc value.

1f

```
mod.3a=glm(cbind(Agrade,n-Agrade)~poly(caffeine,3),family=binomial, data=Caffeine.df)
pred = predict(mod.3a,newdata=new.df,type="response")
diff = round(pred,6)==round(p3,6);table(diff)
```

```
## diff
## TRUE
##  501
```

```
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
```

By using 'predict' and 'cor' function above, it's evident that the predictions of the data we observe using mod.3 and mod.3a are identical(first 3 snippets above), meanwhile, the correlation among these terms have been eliminated(all become 0)

Q2

2a

```
Caffeine2.df=data.frame(n=rep(300,5),levels=seq(0,200,50),
                        A_grades=c(109,155,175,158,103))
mod.quad=glm(cbind(A_grades,n-A_grades)~levels+I(levels^2),
             family=binomial, data=Caffeine2.df)
summary(mod.quad)
```

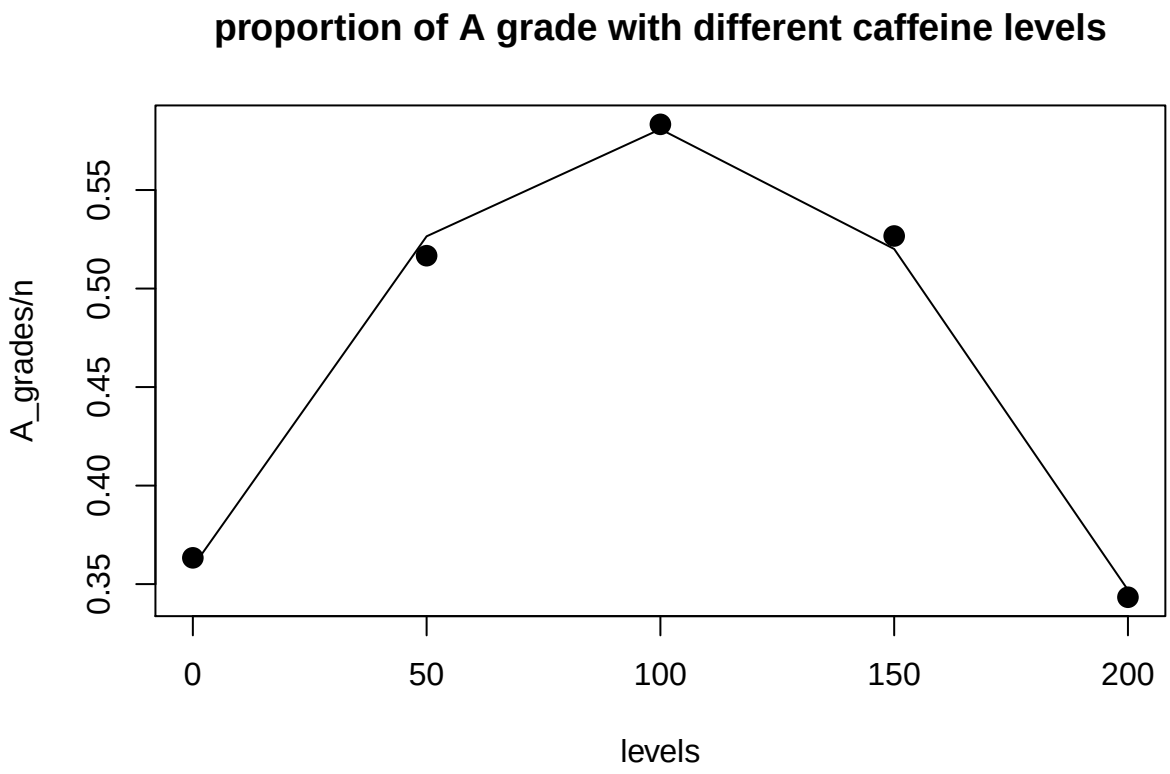
```
##
## Call:
## glm(formula = cbind(A_grades, n - A_grades) ~ levels + I(levels^2),
##     family = binomial, data = Caffeine2.df)
##
## Deviance Residuals:
##        1         2         3         4         5
##  0.16292  -0.34144   0.08575   0.22850  -0.13481
```

```
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.805e-01  1.128e-01  -5.146 2.66e-07 ***
## levels       1.840e-02  2.651e-03   6.940 3.93e-12 ***
## I(levels^2) -9.327e-05  1.272e-05  -7.331 2.28e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 55.54424  on 4  degrees of freedom
## Residual deviance:  0.22086  on 2  degrees of freedom
## AIC: 36.793
##
## Number of Fisher Scoring iterations: 3
```

```
vcov(mod.quad)
```

```
##               (Intercept)        levels   I(levels^2)
## (Intercept)  1.272482e-02 -2.226622e-04  8.287286e-07
## levels      -2.226622e-04  7.027689e-06 -3.232232e-08
## I(levels^2)  8.287286e-07 -3.232232e-08  1.618650e-10
```

```
plot(A_grades/n~levels,data = Caffeine2.df,pch=16,cex=1.5,
     main='proportion of A grade with different caffeine levels')
lines(Caffeine2.df$levels,fitted(mod.quad),)
```



Looks like there is a quadratic relationship between the different levels of caffeine and probability of getting A, and the caffeine level around 100 has the highest chance to A, also, the extreme levels(0 or 200) of caffeine seems to have less chance to gain A.

2b

```
x_peak = -coef(mod.quad)[2]/coef(mod.quad)[3]/2;x_peak
```

```
##   levels
## 98.61706
```

2c

```
Delta.g = c(0, -1/coef(mod.quad)[3]/2, coef(mod.quad)[2]/coef(mod.quad)[3]^2/2)
Delta.g <- unname(Delta.g);Delta.g
```

```
## [1]       0.000    5360.597 1057292.592
```

Seems x_{peak} does not contain β_0 and the gradient vector of x_{peak} is around (0,5360.6,1057292.6).

2d

```
x_peak_var=t(Delta.g)%*%vcov(mod.quad)%*%Delta.g;x_peak_var
```

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

We estimate that the variance of x_{peak} is around 16.5, then we know $se(\hat{x})$ is our square root of x_peak_var.

2e

```
CI.x <- -coef(mod.quad)[2]/coef(mod.quad)[3]/2 +
  1.96 * c(-1, 1) * c(sqrt(x_peak_var))
CI.x
```

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

So the 95% CI for $x_{peak} = -\frac{\beta_1}{2\beta_2}$ is approximately between 90.7 to 106.6 mg of caffeine.

Q3

3a

```
ns=Caffeine2.df$n
xs=Caffeine2.df$levels
preds = fitted(mod.quad)
unname(preds);Caffeine2.df$A_grades/Caffeine2.df$n
```

```
## [1] 0.3588175 0.5265125 0.5808913 0.5200770 0.3470352
```

```
## [1] 0.3633333 0.5166667 0.5833333 0.5266667 0.3433333
```

```
ys=rbinom(length(ns),size=ns, prob=preds)
ys
```

```
## [1] 121 162 183 150 112
```

The true proportion of obtain A grade by different caffeine levels is roughly same as the predicted proportion from mod.quad.

3b

```r
n = 1e3
xpeaks <- devs <- numeric(n)
for (i in 1:n) {
  ys=rbinom(length(ns),size=ns, prob=preds)
  mod.sim=glm(cbind(ys,ns-ys)~xs+I(xs^2),family=binomial)
  xpeaks[i]=-coef(mod.sim)[2]/coef(mod.sim)[3]/2
  devs[i]=deviance(mod.sim)
}
head(xpeaks);head(devs);median(devs)
```

```
## [1]  92.43552 104.87424 100.39508  94.69605  98.98864  98.95791
```

```
## [1] 0.9319914 0.6757292 0.5798735 3.7073904 0.5412323 0.3454475
```

```
## [1] 1.411341
```

```r
mean(xpeaks);mean(devs>qchisq(.95,2))
```

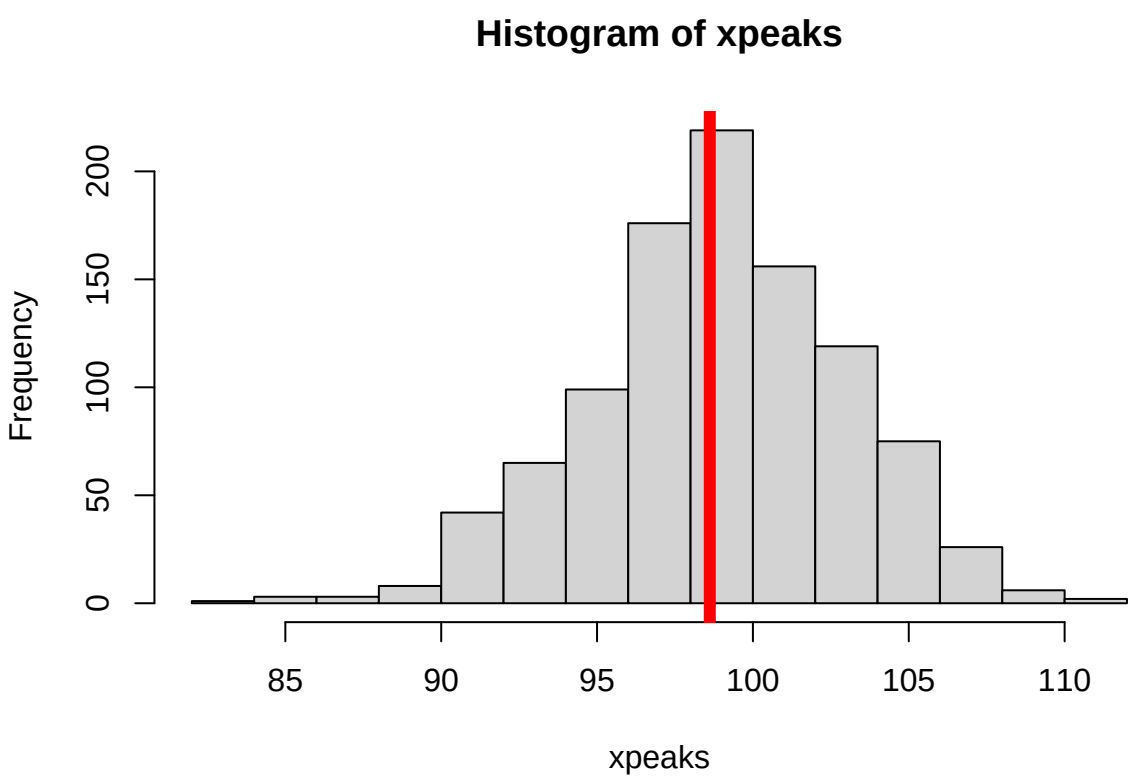
```
## [1] 98.87751
```

```
## [1] 0.053
```

Recall that x_peak we have calculated was 98.6, which almost same as our mean of estimated x_peak, and around 5% deviance is not significant

3c

```r
hist(xpeaks);abline(v=x_peak,col='red',lwd=6)
```




```
quantile(xpeaks,c(.025,.975));mean(xpeaks)
```

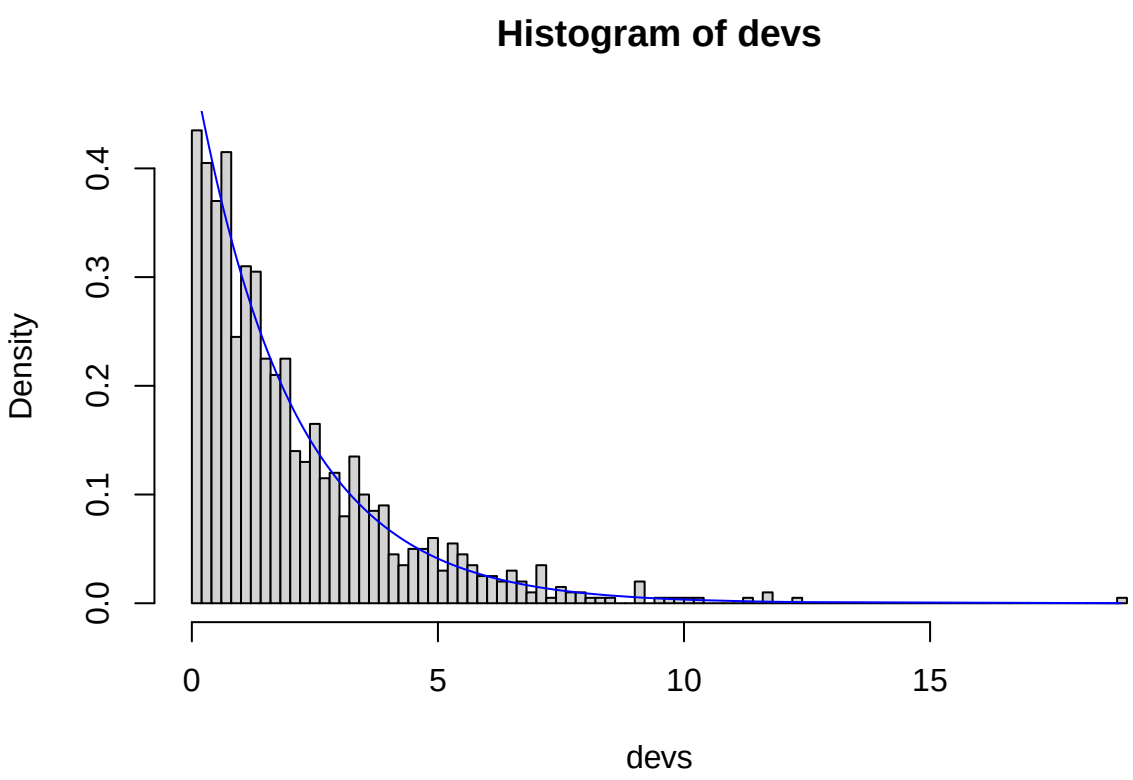
```
##     2.5%    97.5%
##  90.7148 106.2926
```

```
## [1] 98.87751
```

As we can that the x_peak is at the right spot of histogram(peak), which means we done well, meanwhile, x_peak is within the interval of 95% quantiles(about 90 to 106).

3d

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



The ??? is 2 as it is equivalent to the degree of freedom of deviance. It's obvious that the line go along with the trend of data, which means the deviance that we have obtained seem to come from an appropriate Chi-squared distribution.