Advanced Predictive Analytics Assignment 2

16343261

08/05/2023

Table of Contents

Before we begin this assignment, I will load in the data and packages as required.

```
library(COMPoissonReg)
library(AER)
library(ggplot2)
library(cowplot)
library(readxl)
library(lme4)
library(RLRsim)
library(dplyr)
library(lattice)
library(sm)
library(splines)

setwd("C:/Users/matth/Documents/Advanced Predictive Analysis")
data(couple)
```

Question 1.

Question 1a.

We will fit our Poisson GLM and interpret.

```
fit.pois <- glm(UPB ~ EDUCATION + ANXIETY, data = couple, family=poisson)
summary(fit.pois)


Call:
glm(formula = UPB ~ EDUCATION + ANXIETY, family = poisson, data = couple)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.2829  -2.0556  -1.5971   0.0018  12.5621
```

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.81695    0.04386  18.628   <2e-16 ***
EDUCATION   -0.21579    0.07047  -3.062   0.0022 **
ANXIETY      0.42169    0.03333  12.651   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 2478.3  on 386  degrees of freedom
Residual deviance: 2310.8  on 384  degrees of freedom
AIC: 2782.4

Number of Fisher Scoring iterations: 6
```

By using a significance level at 5%, we can reject the null hypothesis H0 : $\beta = 0$ for all β and accept H1: β is not equal to zero. Therefore,we find both variables are statistically significant.

Question 1b.

```
dispersiontest(fit.pois, trafo = function(x) x^2)


    Overdispersion test

data:  fit.pois
z = 3.0033, p-value = 0.001335
alternative hypothesis: true alpha is greater than 0
sample estimates:
   alpha
3.186317
```

We choose a quadratic form for the transformation of μ since the variance of the NB distribution is a quadratic function of μ, which is the most important alternative to the Poisson model to handle overdispersion. We reject the null hypothesis in favour of the alternative one (which states that the data is overdispersed) for any usual significance level considered (note that the p-value is too small). A possible cause for overdispersion is the excess of zeros.

Due to the pressence of overdispersion, I would not recommend using this Poisson GLM.

Question 1c.

Overdispersion is not an issue in ordinary linear regression, therefore the model I suggest is a standard linear regression model. This will be fitted below.

```
ln_reg <- lm(UPB ~ EDUCATION + ANXIETY, data = couple)
summary(ln_reg)
```

```
Call:
lm(formula = UPB ~ EDUCATION + ANXIETY, data = couple)

Residuals:
   Min     1Q Median     3Q    Max
-4.519 -2.379 -1.281  0.227 32.864

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   2.4829     0.3090   8.036 1.15e-14 ***
EDUCATION    -0.5058     0.4935  -1.025    0.306
ANXIETY       0.9899     0.2413   4.102 4.99e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.733 on 384 degrees of freedom
Multiple R-squared:  0.0435,    Adjusted R-squared:  0.03852
F-statistic: 8.732 on 2 and 384 DF,  p-value: 0.0001956
```

The covariate Education seems to be non-significant to be included in the model. The p-value is 0.306. By using a significance level at 5%, we do not reject the null hypothesis H0 : $\beta 1 = 0$. In other words, Education is not significant to explain the unwanted pursuit behavior perpetrations.

```
confint(ln_reg, level=0.95)

                 2.5 %    97.5 %
(Intercept)  1.8754485 3.0903896
EDUCATION   -1.4760789 0.4643804
ANXIETY      0.5154953 1.4643739

confint(fit.pois, level=0.95)

                 2.5 %     97.5 %
(Intercept)  0.7297437  0.9017034
EDUCATION   -0.3549093 -0.0785474
ANXIETY      0.3564507  0.4871405
```

Using H0 : $\beta = 0$ and H1 : β not equal to zero. For our linear regression, we find we cannot reject our null hypothesis for our education variable and therefore it may not be significant. For our other variables, we can reject our null hypothesis and accept our alternative hypothesis.

When contrasting to our poisson model, we see a clear inference change as in our poisson model our education variable is accepted.

Question 1d.

First to create our boxplot, we must first establish a zero variable, our x values matrix and our y variable.

```
ln.er <- array(0,c(100,1))
pois.er <- array(0,c(100,1))
x <- model.matrix(UPB ~., couple)[, -1]
y <- couple$UPB
```

Next we will use a for loop with 100 iterations to get the 100 test error values for both models as described.

```
for(i in 1:100){
train <- sample(1:nrow(couple), nrow(couple) / 2)
test <- (-train)
y.test <- couple$UPB[test]

ln_reg <- lm(y[train] x̃[train, ] )
fit.pois <- glm(UPB[train] ~ EDUCATION[train] + ANXIETY[train], data =
couple, family=poisson)

ln.pred<-cbind(1,x[test, ])%*%ln_reg$coef
ln.er[i] <- mean((ln.pred - y.test)^2)

pois.pred<- cbind(1,x[test, ])%*%fit.pois$coef
pois.er[i] <- mean((pois.pred - y.test)^2)
}
```

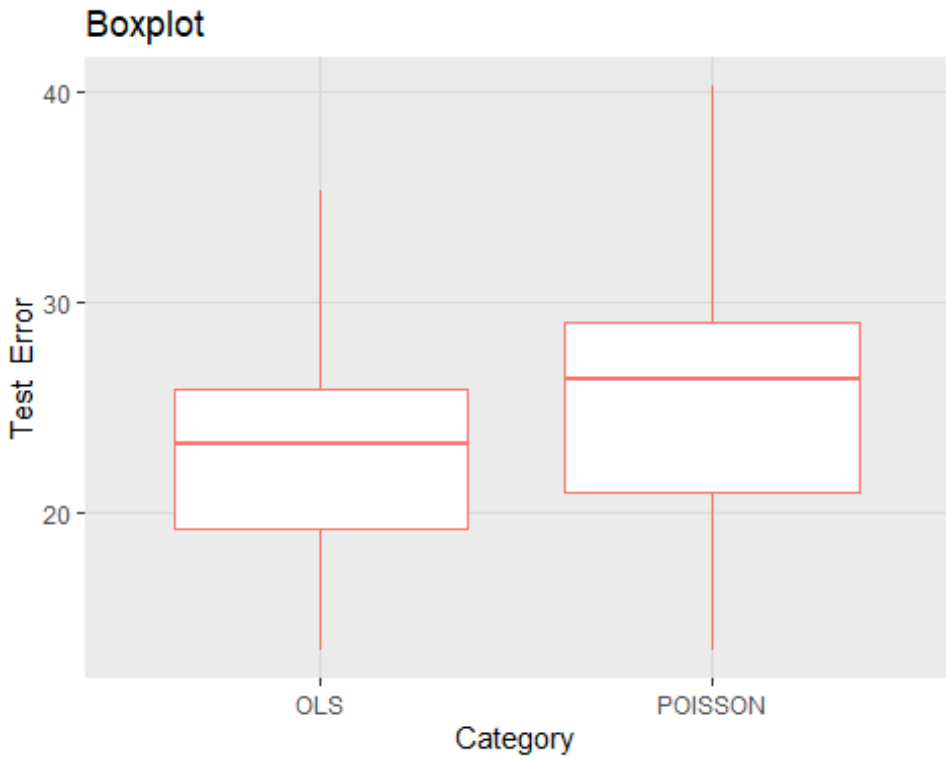
Next we will create a category variable for plotting purposes and create our plotting values.

```
cat <- c()
cat[1:100] <- "OLS"
cat[101:200] <- "POISSON"

test_error <- c()
test_error[1:100] <- ln.er
test_error[101:200] <- pois.er

plot <- data.frame(cat,test_error)
plot$cat <- as.factor(plot$cat)
```

Now let's produce our plot.

```
# Boxplot
bp <- ggplot(plot, aes(x=cat, y=test_error, color='red')) +
  geom_boxplot() +
  xlab('Category') +
ylab('Test  Error') +
  theme(legend.position = "none")
# Add gridlines
```

```
bp + background_grid(major = "xy", minor = "none")+
ggtitle('Boxplot ')
```



Here we find the test error for our Poisson model has noticeably higher test error than our ordinary least squares regression. From this, it is clear that Ordinary Least Squares regression is performing better in terms of prediction.

Question 2.

Question 2a.

To answer this question, we will load in our mathach data set and create our sampled school variables, create a new data set and fit our linear regression for our fixed factor as described.

```
data<-read_xlsx("HSAB.xlsx", sheet = 1)

set.seed(112)
sample <- sample(data$school, 5)
sample

[1] 1296 4325 9104 9397 3881

data2 <- data[which(data$school == sample), ]
data2$school <- as.factor(data2$school)
```

```
fixed_model = lm(math.achieve ~ school, data = data2,REML=FALSE)
summary(fixed_model)
```

```
Call:
lm(formula = math.achieve ~ school, data = data2, REML = FALSE)

Residuals:
     Min       1Q   Median       3Q      Max
-16.1251  -4.4762  -0.5953   4.3483  14.8809

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   8.2911     2.1383   3.877 0.000356 ***
school3881    0.7764     3.0240   0.257 0.798589
school4325    3.6520     2.9474   1.239 0.222053
school9104   10.0591     2.8833   3.489 0.001133 **
school9397    5.2620     3.0240   1.740 0.089000 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.415 on 43 degrees of freedom
Multiple R-squared:  0.2702,    Adjusted R-squared:  0.2023
F-statistic: 3.979 on 4 and 43 DF,  p-value: 0.007805
```

From our model, we find the 9104 and 1296 (through our intercept) schools are statistically significant. Here we find school #9104 has a particularly positive affect on maths achievement scores with a coefficient of 10.0591. In our model, school #9397 is statistically significant for a p value of 0.1, but not significant for 0.05.

Question 2b.

Our model below has fixed and random effects components. The fixed effect here is just the intercept represented by the first 1 in the model formula. The random effect is represented by (1|school) indicating that the data is grouped by school and the 1 indicating that the random effect is the same within each group.

```
random_model = lmer(math.achieve ~ 1 + (1 |school), data = data2,REML=FALSE)
summary(random_model)
```

```
Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: math.achieve ~ 1 + (1 | school)
   Data: data2

     AIC      BIC   logLik deviance df.resid
   326.2    331.9   -160.1    320.2       45

Scaled residuals:
     Min       1Q   Median       3Q      Max
-2.45174 -0.80855  0.02646  0.76429  2.11232
```

```
Random effects:
 Groups   Name        Variance Std.Dev.
 school   (Intercept)  9.106   3.018
 Residual             41.070   6.409
Number of obs: 48, groups:  school, 5

Fixed effects:
            Estimate Std. Error t value
(Intercept)   12.316      1.637   7.523
```

Residuals are expected to be approximately symmetric (median around zero) and the extremes around ±3 if the model is well-fitted to the data. Here we find our residuals are closer to 2 than 3, but certainly approximately symmetric.

Here we find the estimated standard deviation among schools to be: 3.513. Here we find the estimated standard deviation among individuals in the schools to be: 6.411.

There does not appear to be an inferential change compared to the model considered in a.

Question 2c.

We find our intraclass correlation coefficient from our summary of our random model.

```
12.34/(12.34+41.10)
```

```
[1] 0.2309132
```

This implies 23.1% of the variation in students' math achievement scores is "attributable" to differences among schools.

Next our confidence interval.

```
confint(random_model, method="boot")
```

```
               2.5 %    97.5 %
.sig01      0.000000  4.884513
.sigma      5.086249  7.611759
(Intercept) 9.092911 15.626212
```

Question 2d.

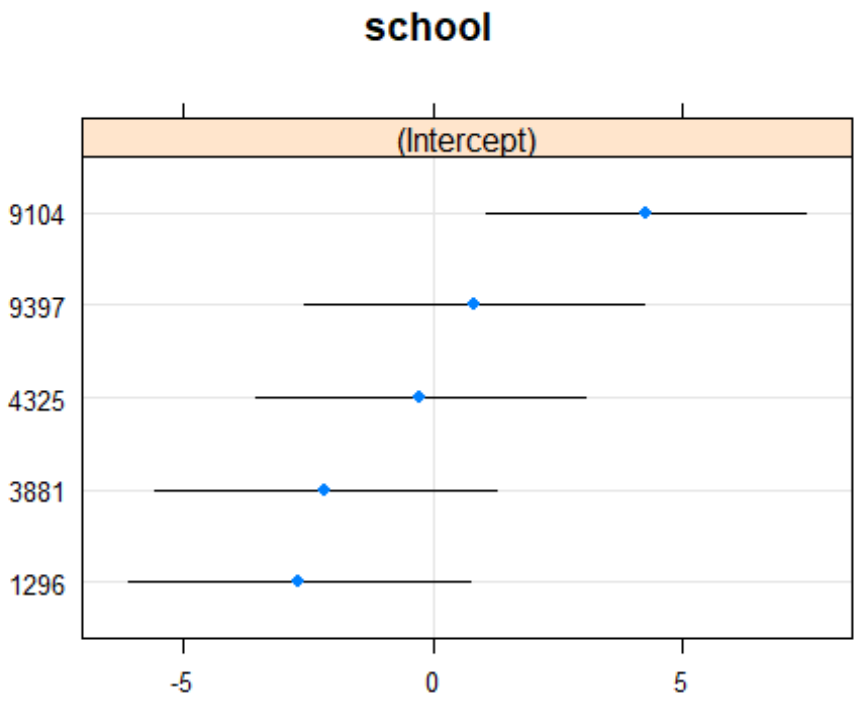
Here we can predict the random effects using the bootstrap method shown below by comparing our model to a fixed model without random events. We will take these results to get our findings.

```
nullfit <- lm(math.achieve~1, data = data2)
LR <- as.numeric(2*(logLik(random_model)-logLik(nullfit)))
lrstat <- numeric(1000)

set.seed(123)
for(i in 1:1000){
```

```
y <- unlist(simulate(nullfit))
null.model <- lm(y~1, data = data2)
alt.model <- lmer(y~1+(1|school), data = data2, REML=FALSE)
lrstat[i] <- as.numeric(2*(logLik(alt.model)-logLik(null.model)))
}
```

Using the dotplot function we can display our 95% confidence intervals for our predictions.

```
exactLRT(random_model, nullfit)


    simulated finite sample distribution of LRT. (p-value based on 10000
    simulated values)

data:
LRT = 4.2363, p-value = 0.0063

dotplot(ranef(random_model, condVar=TRUE))

$school
```



Here we see similar findings to our earlier fixed model.

Question 3.

Question 3a.

For this question, lets start by defining our 'f' function.

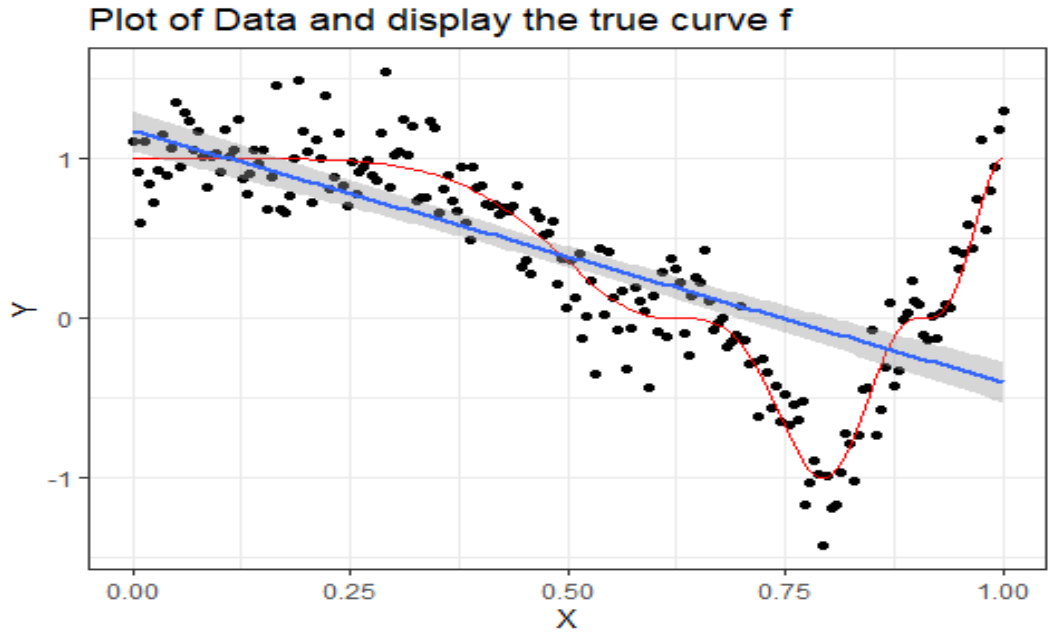
```
f <-function(x){
(cos(2*(pi)*(x)^(3)))^3
}
```

Next we will define our error,obtain our y values and create our required dataset.

```
sd <- sqrt(0.04)
err <- rnorm(200,0,sd)
x <- seq(0,1, by = 1/199)
Y <- f(x) + err

gen <- as.data.frame(x)
gen$y <-Y
gen$f <- f(x)
```

Now we will plot our 'f' function over our data.

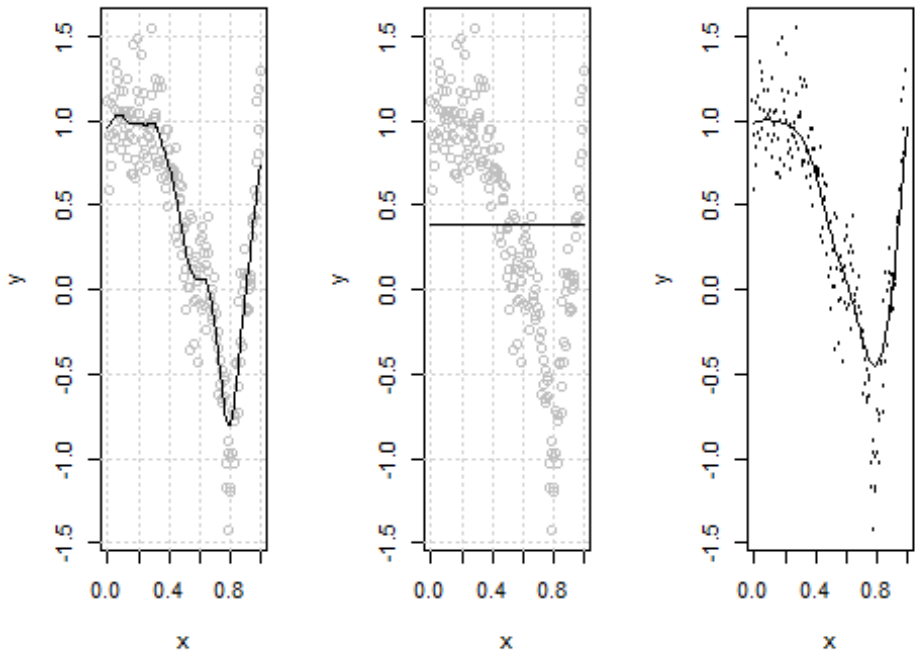
```
ggplot(gen, aes(x, y)) +
geom_point() +
geom_line(aes(y = f(x)), color = "red") +
stat_smooth(method = "lm") +
xlab('X') +
ylab('Y') +
theme_bw()+
ggtitle('Plot of Data and display the true curve f')
```



Question 3b.

Using the sm.regression function we can fit our curve to the data using kernal smoothing. This can be seen below.

```
par(mfrow=c(1,3))
for(bw in c(0.1,18)){
with(gen,{
plot(y ~ x, col=gray(0.75))
grid()
lines(ksmooth(x,y,"normal",bw))
})}
with(gen,sm.regression(x, y,
h=h.select(x,y)))
```




```
fit<-with(gen,sm.regression(x, y,
h=h.select(x, y)))
```

This fit does not look satisfactory as when compared to the fitted curve with band width = to 0.1, we see a clear difference in fit with the data.

Question 3c.

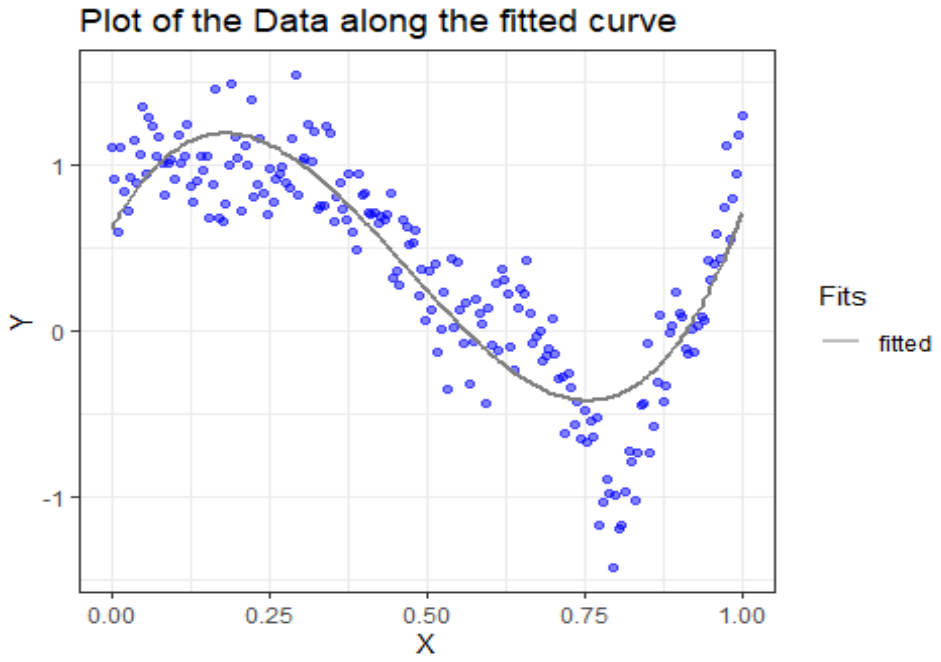
Below we fit our model using smoothing splines with the automatically chosen amount of smoothing.

```
# generate appropriate spline basis
xtilde<-bs(x)
```

```r
# least squares to determine the coefficients
fit<-lm(y ~ xtilde, data= gen)
gen$fitted <- predict(fit)
summary(fit)
```

```
Call:
lm(formula = y ~ xtilde, data = gen)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.02767 -0.17365  0.02063  0.20345  0.72656 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.61638    0.08371   7.363 4.85e-12 ***
xtilde1      2.30616    0.24225   9.520  < 2e-16 ***
xtilde2     -3.32071    0.15465 -21.473  < 2e-16 ***
xtilde3      0.09858    0.13186   0.748    0.456    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3015 on 196 degrees of freedom
Multiple R-squared:  0.7876,    Adjusted R-squared:  0.7843 
F-statistic: 242.2 on 3 and 196 DF,  p-value: < 2.2e-16
```

```r
ggplot(gen, aes(x = x, y = y, color='grey'))+
geom_point(col = 'blue', alpha = 0.5)+
geom_line(aes(x = x, y = fitted),size=1)+
scale_color_manual(name = "Fits", values = c("fitted" = "grey"))+
xlab('X') +
ylab('Y') +
theme_bw()+
ggtitle("Plot of the Data along the fitted curve")
```



Here we find the automatic choice of 3 degrees of freedom was not satisfactory. As we can see from our curve, it does not fully fit to our data. Furthermore, our 3rd x tilde is not statistically significant. We may need to try higher degrees of freedom and compare our vsalues.

Question 3d.

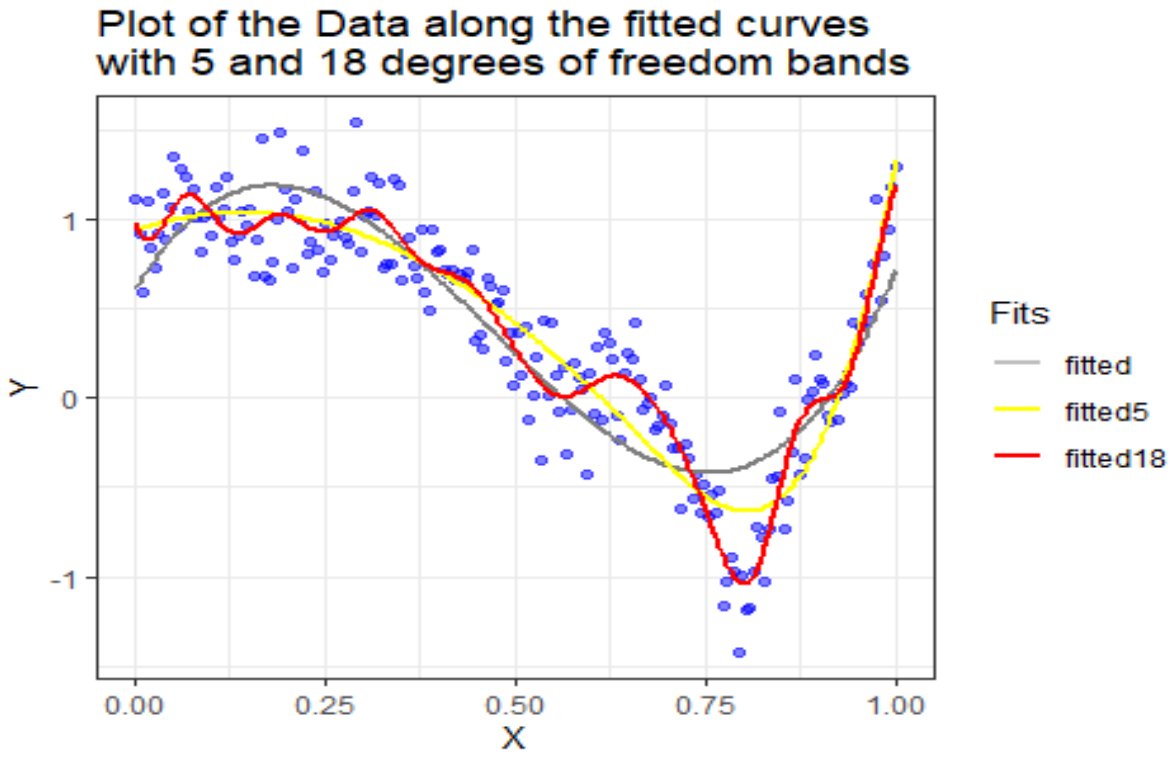
To fit our 5 and 18 degrees of freedom using xtilde for df = 5 and xtilde for df = 18. Additionally, we will add our values to our Gen dataset.

```
xtilde5<-bs(x, df = 5)
fit5<-lm(y ~ xtilde5, data= gen)
xtilde18<-bs(x, df = 18)
fit18<-lm(y ~ xtilde18, data= gen)
gen$fitted5 <- predict(fit5)
gen$fitted18 <- predict(fit18)
```

Lastly we will plot the data along our fitted curves.

```
ggplot(gen, aes(x = x, y = y, color='grey'))+
geom_point(col = 'blue', alpha = 0.5)+
geom_line(aes(x = x, y = fitted),size=1)+
geom_line(aes(x = x, y = fitted5),col = 'yellow',size=1)+
geom_line(aes(x = x, y = fitted18),col = 'red',size=1)+
  scale_color_manual(name = "Fits", values = c("fitted" = "grey", "fitted5" =
"yellow", "fitted18" = "red"))+
xlab('X') +
ylab('Y') +
theme_bw()+
```

```
ggtitle("Plot of the Data along the fitted curves
with 5 and 18 degrees of freedom bands")
```



From this question, I believe that the 18 degrees of freedom clearly provides a better estimate of the true curve than our other curves. This is because it fits the data better than any of our plots except possibly our kernal smoothing plot with a 0.1 bandwidth smoothing parameter.