Assignment 3 - Stats 330

Hasnain Cheena

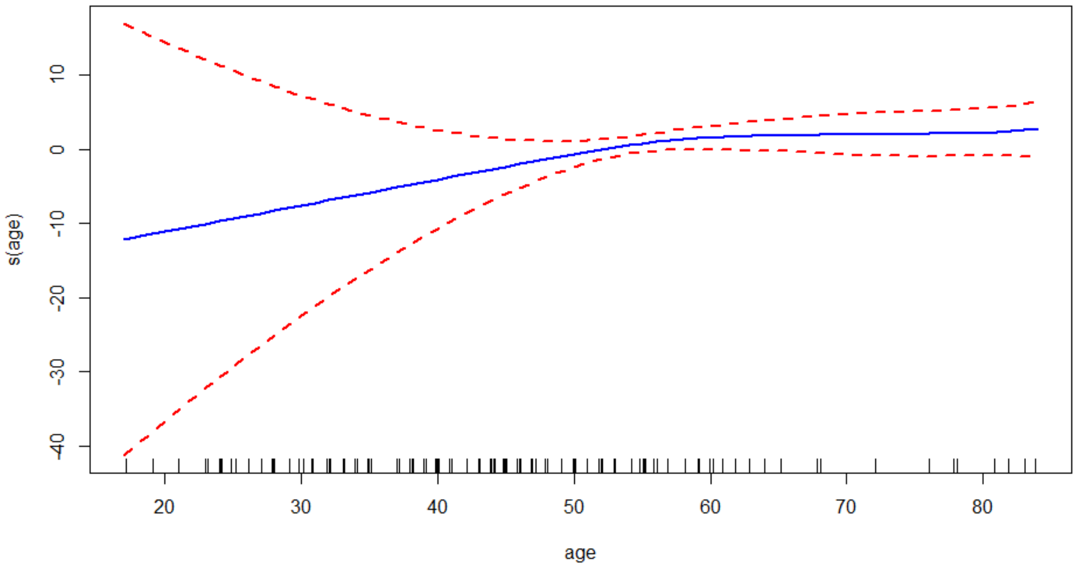
20/05/2020

Question 1: Heart Attacks

```
hearthealth.df$chol <- factor(hearthealth.df$chol) #convert chol to a factor

#fit GAM
heart.gam.fit = vgam(heartattack ~ s(age) + chol, family=binomialff, data=hea
rthealth.df)

#model summary
summary(heart.gam.fit)

##
## Call:
## vgam(formula = heartattack ~ s(age) + chol, family = binomialff,
##     data = hearthealth.df)
##
## Name of additive predictor: logitlink(prob)
##
## (Default) Dispersion Parameter for binomialff family:   1
##
## Residual deviance:  32.04316 on 115.14 degrees of freedom
##
## Log-likelihood: -16.02158 on 115.14 degrees of freedom
##
## Number of Fisher scoring iterations:  12
##
## DF for Terms and Approximate Chi-squares for Nonparametric Effects
##
##               Df Npar Df Npar Chisq   P(Chi)
## (Intercept)  1
## s(age)       1     1.9    2.51398 0.258578
## chol         1
```

The summary output shows us that we have no evidence against using a linear term for age in $ha.lin$. Hence using linear terms for both age and cholesterol in $ha.lin$ is appropriate.

```
plot(heart.gam.fit, se=TRUE, lcol="blue", scol="red", llwd=2, slwd=2)
```



From the plot you can see the standard error bands are very wide on the left-hand side. Moreover, the plot shows that *age* is reasonably linear. Furthermore, from the rugplot you can see the distribution of age is right skewed.

Mathematical formula of the GAM model:

$$logit(p_i) = \beta_0 + f_1(age_i) + \beta_2 chol_i$$

$$Y_i \sim Binomial(1, p_i)$$

Where
age_i is the age in years of a participant, $chol_i$ = 1 when a participant has low cholesterol and 0 when a participant has high cholesterol and p_i is the probability the ith participant had a heart attack. Furthermore, Y_i is a Bernoulli random variable that takes the value 1 with probability p_i and 0 with probability $1 - pi$ and f_1 is a smooth function determined from the data.

Question 2: Women’s BMI

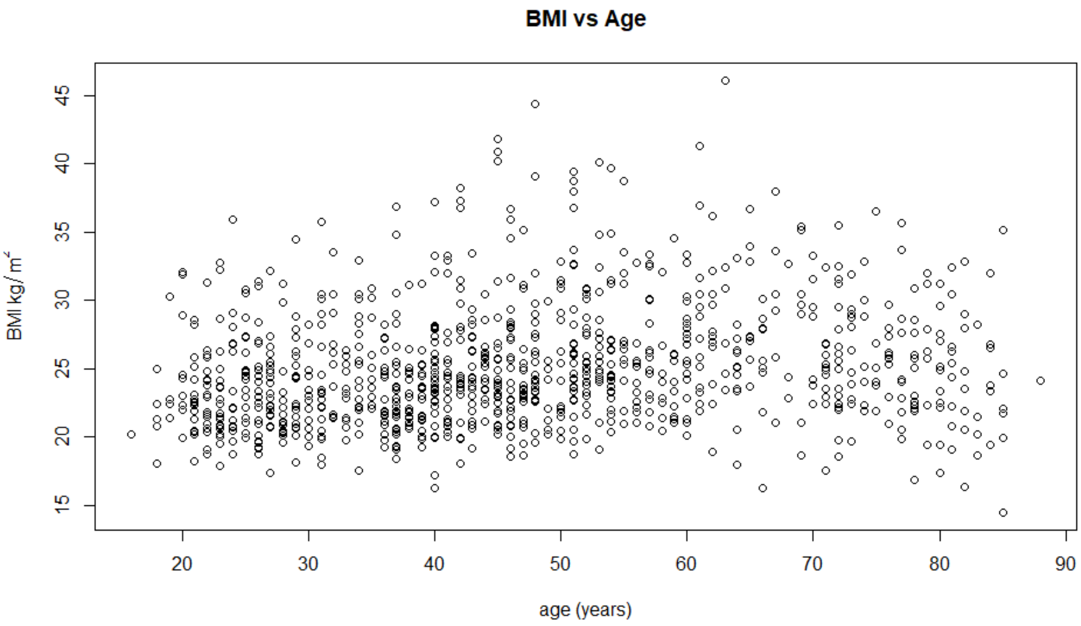
```
#read the data in
bmi.df = read.csv('feuro.csv')
#sort by age
bmi.df = bmi.df[order(bmi.df$age),]

#calculate BMI
bmi.df$bmi = bmi.df$weight / ((bmi.df$height)^2)


#proportion of obese women
prop.obese = sum(bmi.df$bmi >= 30) * 100 /nrow(bmi.df)
#proportion of overweight women
prop.overweight = sum(bmi.df$bmi >= 25 & bmi.df$bmi < 30)*100/nrow(bmi.df)

proportions = data.frame("groups"= c("Obese", "Overweight"),
                          "proportions" = c(prop.obese, prop.overweight))

proportions

##       groups proportions
## 1      Obese    12.77735
## 2 Overweight    29.03596
```

Source: https://www.who.int/news-room/fact-sheets/detail/obesity-and-overweight
The source above defines:

- Obesity as having a BMI that is greater than or equal to 30
- Overweight as having a BMI that is greater than or equal to 25

From the calculation above you can see 12.8% of the European women in our sample are classified as obese and 29% are classified as overweight but not obese.

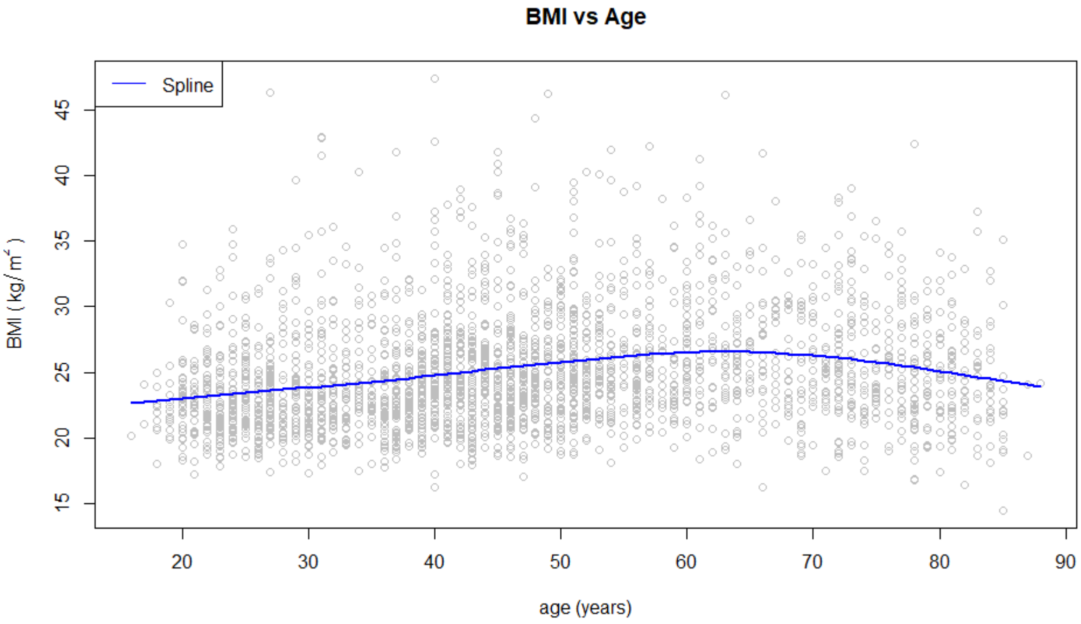
```
#dataset is very large so randomly select sample
set.seed(321)
index = sample(nrow(bmi.df), size=1000)
small.bmi.df = na.omit(bmi.df[index,])

#scatterplot
plot(bmi~age, small.bmi.df, main="BMI vs Age", xlab="age (years)", ylab=expre
ssion ("BMI"~kg/m^2))
```



A random sample of 1000 datapoints was taken. This is to produce a high-quality scatterplot in which important features can easily be seen.

From the scatterplot it is visible that at the younger age range there is less variance in BMI than at the older age range. Further it is clear there are many more observations of European women between the ages of 20-50 than people between 50-90. Moreover, from the scatterplot generally as age increases BMI also increases until around 60 years old. After this age BMI starts to decrease again.

```
#smooth spline - EDF set by GCV
bmi.spline = with(bmi.df,smooth.spline(age,bmi))
bmi.spline

## Call:
## smooth.spline(x = age, y = bmi)
##
## Smoothing Parameter  spar= 0.7975298  lambda= 0.1108176 (13 iterations)
## Equivalent Degrees of Freedom (Df): 5.075126
## Penalized Criterion (RSS): 1156.136
## GCV: 17.81344

#add to plot
plot(bmi~age, bmi.df, col="grey", main="BMI vs Age", xlab="age (years)", ylab
=expression ("BMI ("~kg/m^2 ~ ")"))
lines(bmi.spline, col="blue", lwd=2)
legend("topleft",legend=c("Spline"),col=c("blue"),lty=rep(1))
```



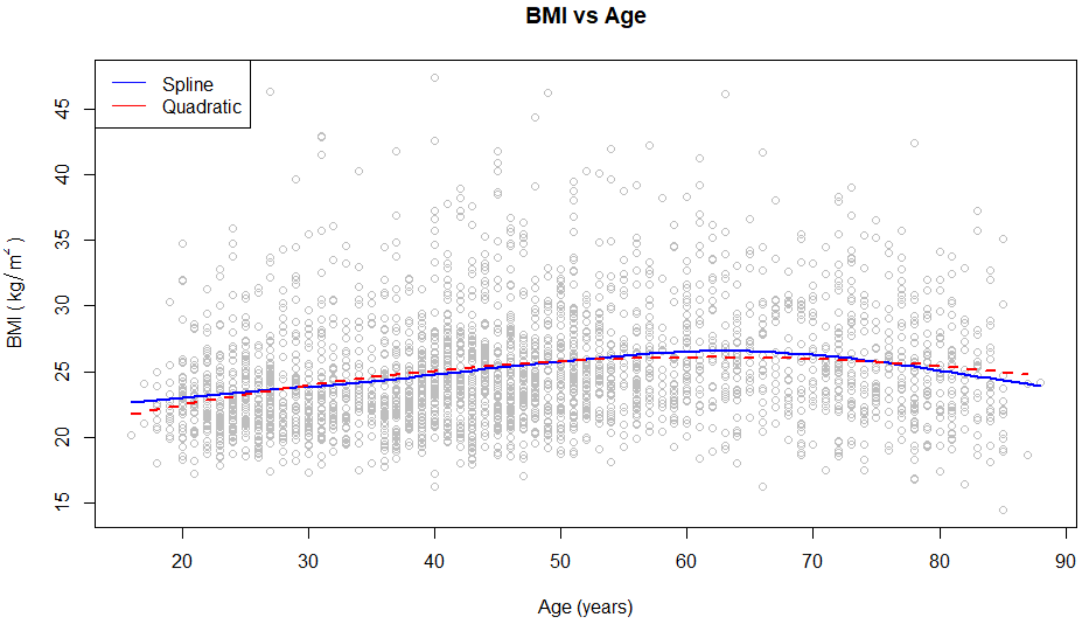
The equivalent degrees of freedom achieved using GCV to fit the spline is approximately 5, which is reasonable. From the plot the spline seems to be a good fit (not too overfit or underfit). The trend seems to be as European woman get older their BMI increases until it peaks at around 62 years old. After this age it starts to decrease again.

This trend could be because between the ages 20-60 people are usually in the workforce. This lifestyle can be sedentary causing increases in weight and hence BMI. However, at around 60 years old people retire and as they are older tend to have health issues causing decreases in weight and BMI.

```
predict(bmi.spline, 50)

## $x
## [1] 50
##
## $y
## [1] 25.79227
```

We estimate for a European woman that is 50 years old, her BMI will be approximately 25.8.

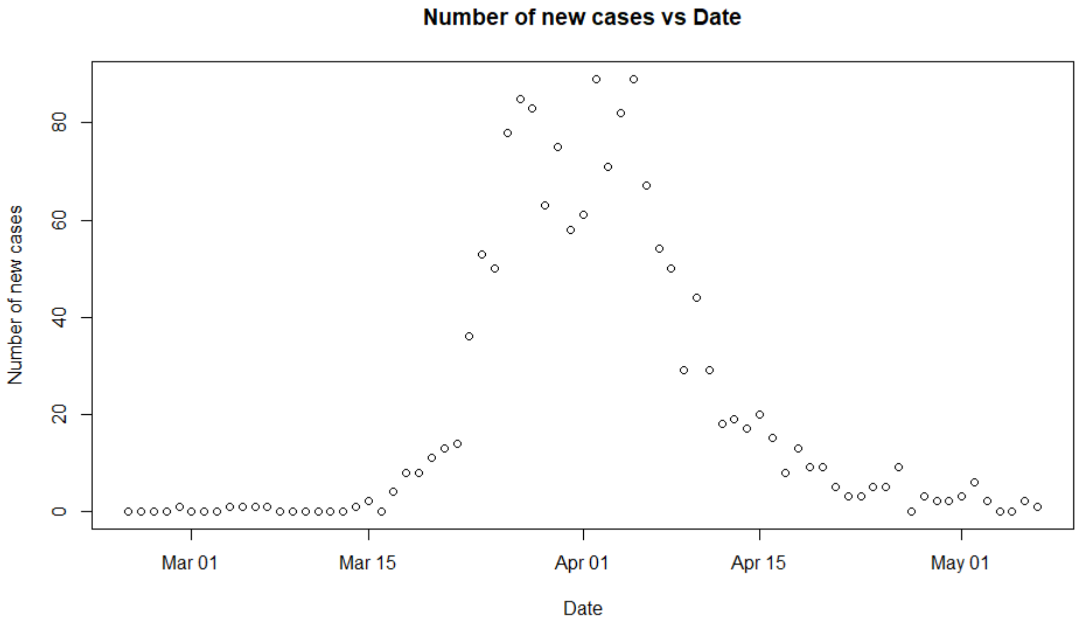
```
#quadratic
bmi.quad.fit = lm(bmi~ poly(age,2), data=bmi.df)

age = data.frame("age"=16:87)
predictions = predict(bmi.quad.fit, age, type="response")

#fit
#add to plot
plot(bmi~age, bmi.df, col="grey", xlab="Age (years)", ylab=expression ("BMI (
"~kg/m^2 ~ ")"), main="BMI vs Age")
lines(bmi.spline, col="blue", lwd=2)
lines(age$age, predictions, col="red", lty=2, lwd=2)
legend("topleft",legend=c("Spline","Quadratic"),col=c("blue","red"),lty=rep(1
,2))
```



The quadratic and smooth spline fit are very close. However, the quadratic seems to be slightly underfitting the data unlike the smooth spline.

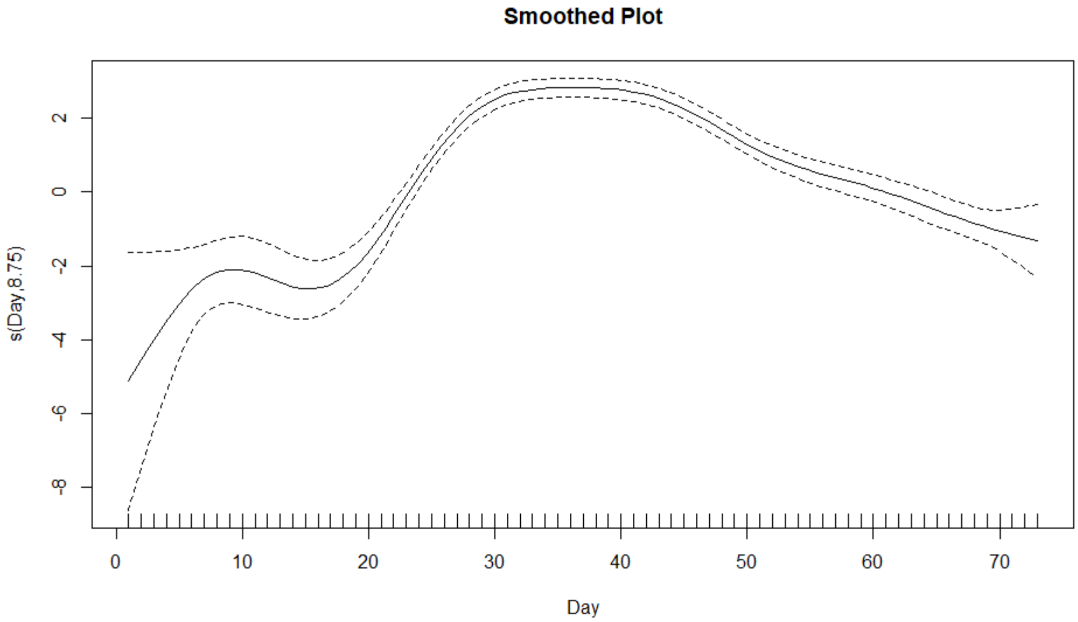
Question 3: COVID-19

```
#read in data
covid.df = read.csv('covid19nz.csv')

#convert doy to date type
covid.df$doy = as.Date(covid.df$doy)

#replace -1 with 0
covid.df[covid.df$newcases == -1,]$newcases = 0

#create day variable
covid.df$Day = seq(1,nrow(covid.df))

#scatterplot
plot(newcases~doy, data=covid.df, main="Number of new cases vs Date", ylab="N
umber of new cases", xlab="Date")
```



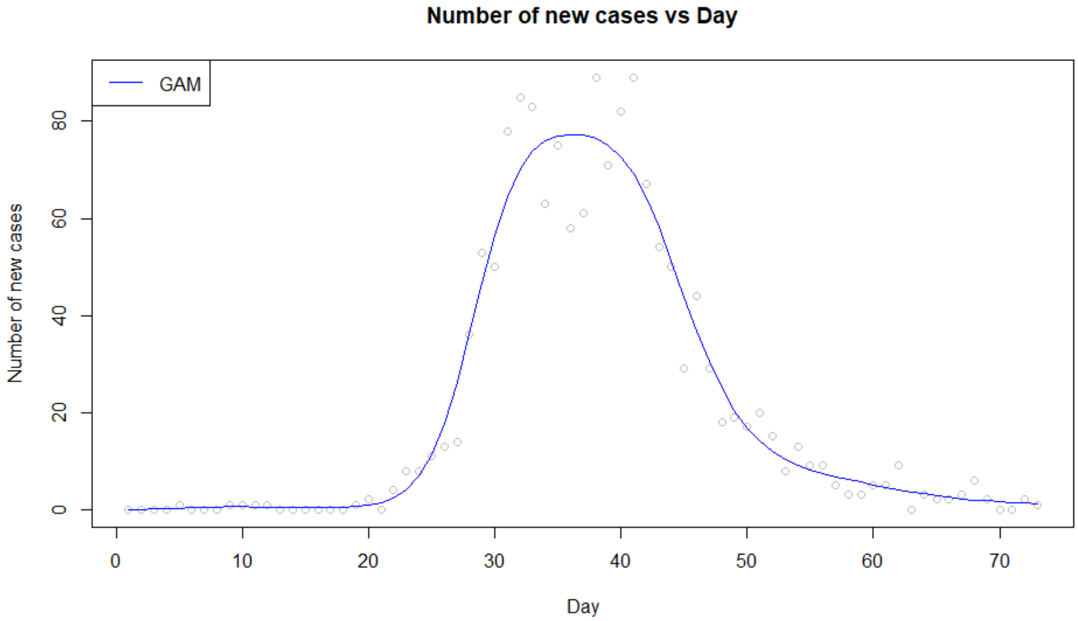
The scatterplot shows that as the number of new cases increases the variance also increases. Furthermore, the plot shows that the distribution of *newcases* is reasonably symmetric. It also shows that the number of new cases peaked twice; once around the end of March and again at the beginning of April.

```r
#fit GAM
covid.gam.fit = gam(newcases~s(Day), data=covid.df, family=poisson)
#plot GAM
plot(covid.gam.fit, main="Smoothed Plot")
```



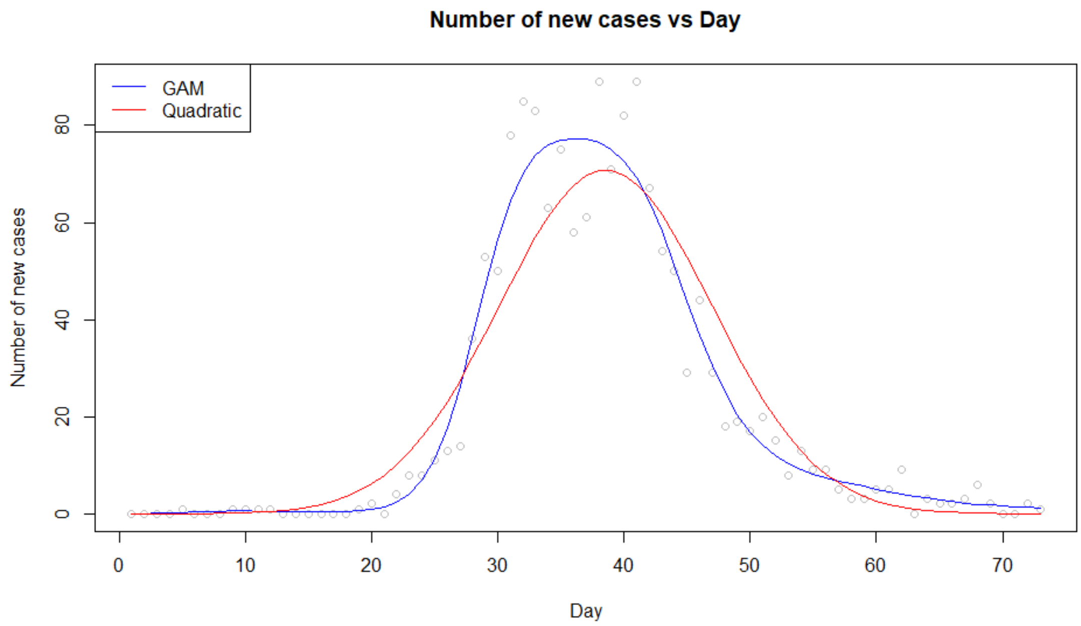
The smoothed plot shows that a linear term for *Day* is not appropriate. Therefore, a non-linear term such as a quadratic may be more appropriate.

```r
#plot scatter and the line
plot(newcases~Day, covid.df, col="grey", main="Number of new cases vs Day", y
lab="Number of new cases")
lines(fitted(covid.gam.fit), col="blue")
legend("topleft",legend=c("GAM"),col=c("blue"),lty=rep(1))
```



The plot of the fitted values versus the observations indicates that the GAM has fit the data reasonably well (not too overfit or underfit).

```
#quadratic
covid.quad.fit = glm(newcases~ poly(Day,2), data=covid.df, family=poisson)

#add to plot
plot(newcases~Day, covid.df, col="grey", main="Number of new cases vs Day", y
lab="Number of new cases")
lines(fitted(covid.gam.fit), col="blue")
lines(fitted(covid.quad.fit), col="red")
legend("topleft",legend=c("GAM","Quadratic"),col=c("blue","red"),lty=rep(1,2)
)
```



```
#use AIC to compare the models
AIC(covid.gam.fit,covid.quad.fit)

##                    df      AIC
## covid.gam.fit  9.7485 357.5734
## covid.quad.fit 3.0000 559.7816
```

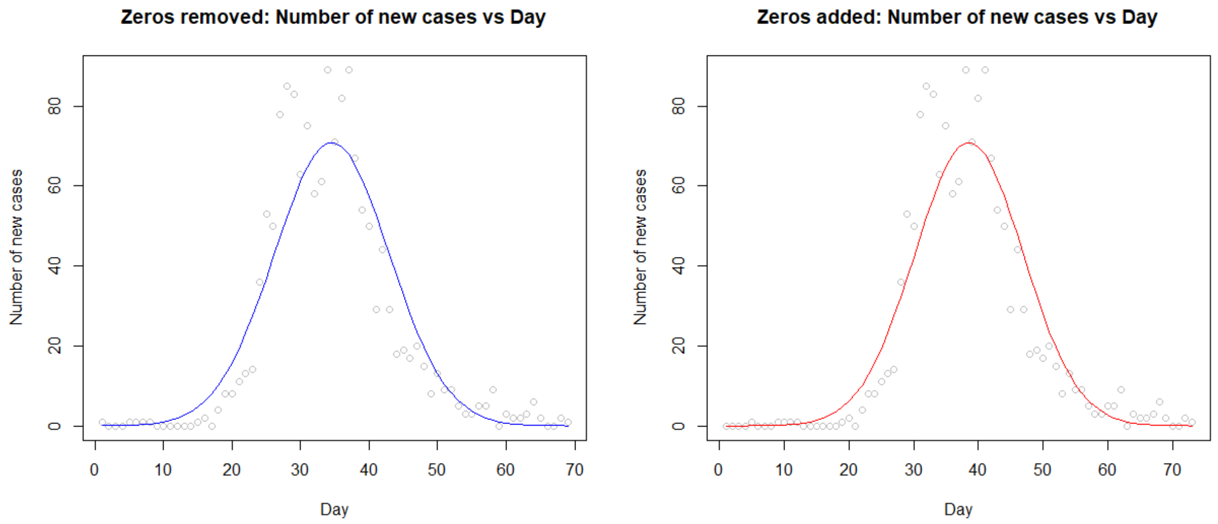
From the plot you can see the quadratic slightly underfits the data compared to the GAM. It overestimates at lower new case numbers and underestimates at the higher new case numbers. Additionally, it predicts the peak is later than the GAM.

Furthermore, the AIC score of the GAM is much lower than the quadratic. This means we have evidence to suggest that the GAM fits the data better than the quadratic model.

```
#GAM peak
covid.df$doy[which.max(covid.gam.fit$fitted.values)]

## [1] "2020-03-31"

#Quadratic peak
covid.df$doy[which.max(covid.quad.fit$fitted.values)]

## [1] "2020-04-03"
```

The GAM estimates that the number of new cases peaked on the 31st of March 2020, compared to the quadratic which estimates the number of new cases peaked on the 3rd of April 2020. The true answer is the cases peaked on the 2nd and 5th of April 2020 at 89 new cases.

```
#remove the additional zeros before the first case
covid.red.df = covid.df[5:nrow(covid.df),]
covid.red.df$Day = 1:nrow(covid.red.df)
#refit quadratic with zero's removed
covid.red.quad.fit = glm(newcases~ poly(Day,2), data=covid.red.df, family=poi
sson)

#plot both quadratics
par(mfrow = c(1, 2))
plot(newcases~Day, covid.red.df, col="grey", main="Zeros removed: Number of n
ew cases vs Day", ylab="Number of new cases")
lines(fitted(covid.red.quad.fit), col="blue")
plot(newcases~Day, covid.df, col="grey", main="Zeros added: Number of new cas
es vs Day", ylab="Number of new cases")
lines(fitted(covid.quad.fit), col="red")
```



A number of 0's was added prior to the first case. As can be seen from the plots above the addition of 0's means the tails of the quadratic model are more similar in length. As the quadratic is a symmetric model this allowed us to use it without performing any other ladder-of-powers transformations.