Machine Learning Computer lab 2 block 2

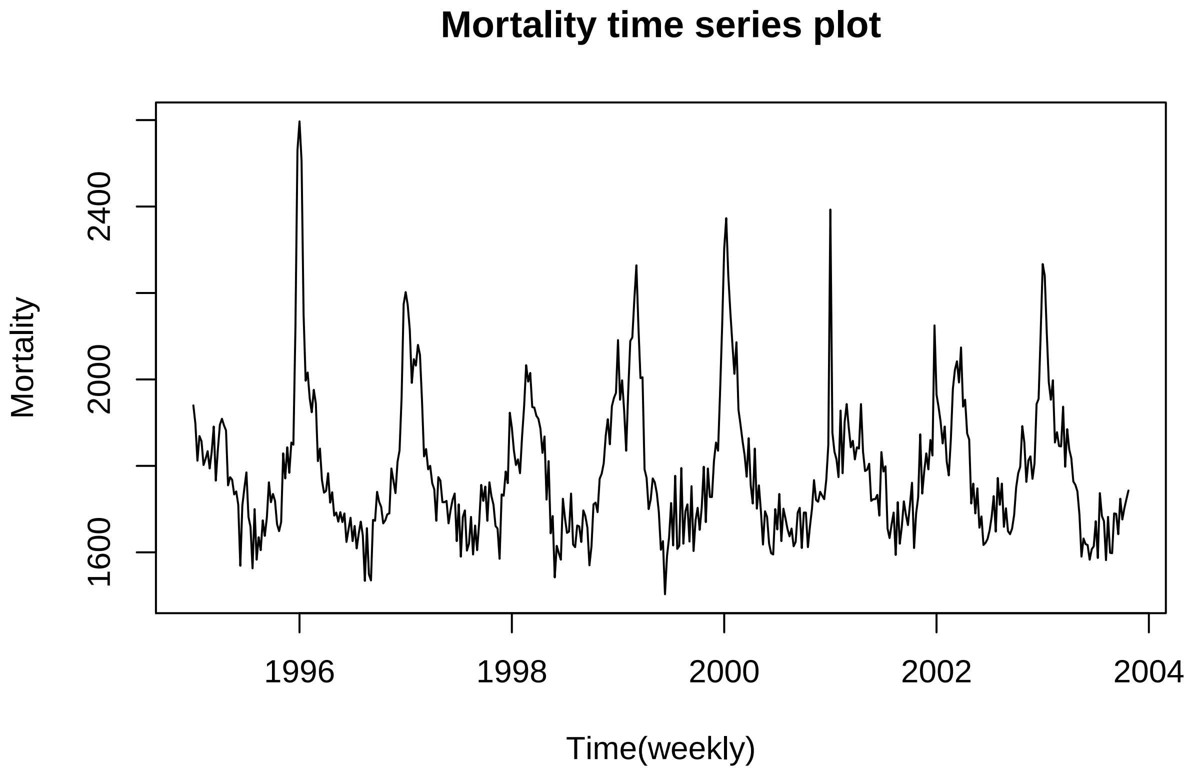
Aashana Nijhawan (aasni448), Nahid Farazmand (nahfa911), Sae Won Jun (saeju204)

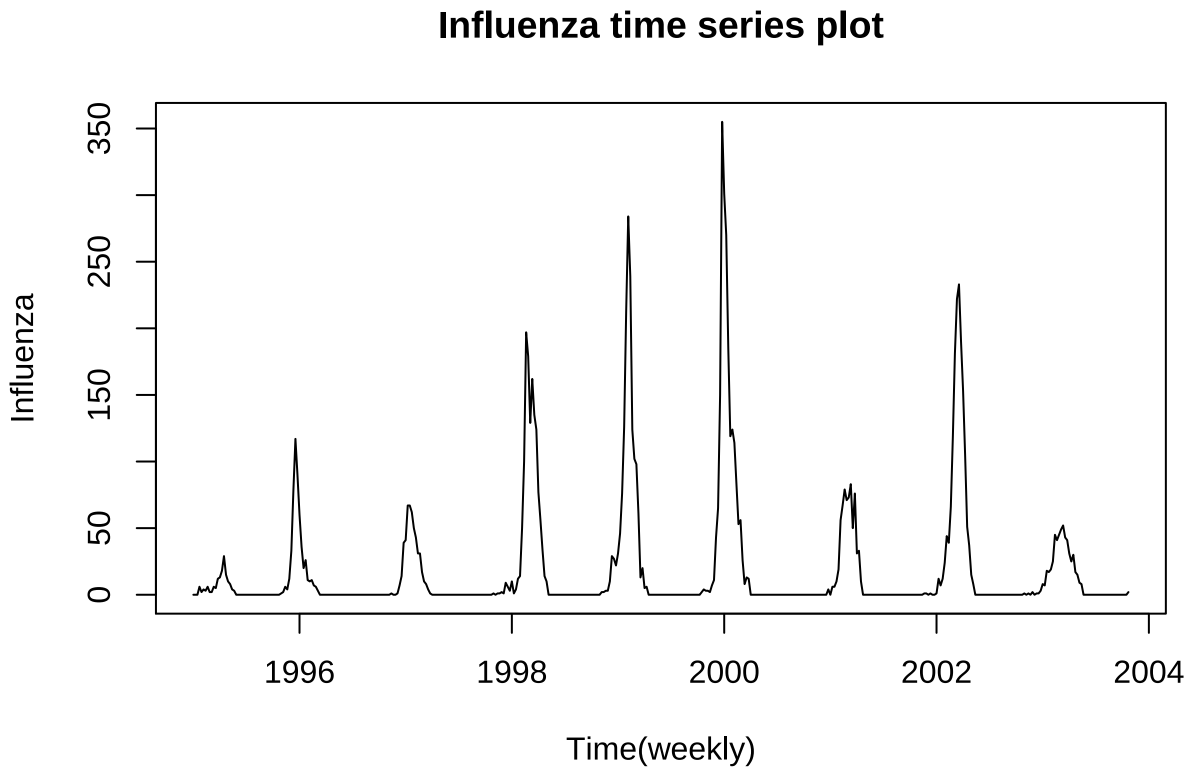
December 13, 2018

Assignment 1. Using GAM and GLM to examine the mortality rates

influenza.xlsx contains *weekly* data on the mortality and the number of laboratory-confirmed cases of influenza in Sweden. In addition, there is information about population-weighted temperature anomalies.

1.1. The mortality and influenza rate change with time





Certain time period where shows the higher value for Mortality also show the higher value for Influenza. In other word, considering the peaks of the plots, these two time series have the same pattern. So we can assume that those to variables are positively corrlated.

1.2. Mortality is normally distributed and modelled as a linear function of Year and splice function of Week

Fitted model model is as follow:

$$Mortality \ \sim \ N(\mu, \sigma^2)$$

$$g(\mu) = \alpha + s_1(Week)$$

$$E(Mortality) \ = \ \alpha + s_1(Week) + Year\beta$$

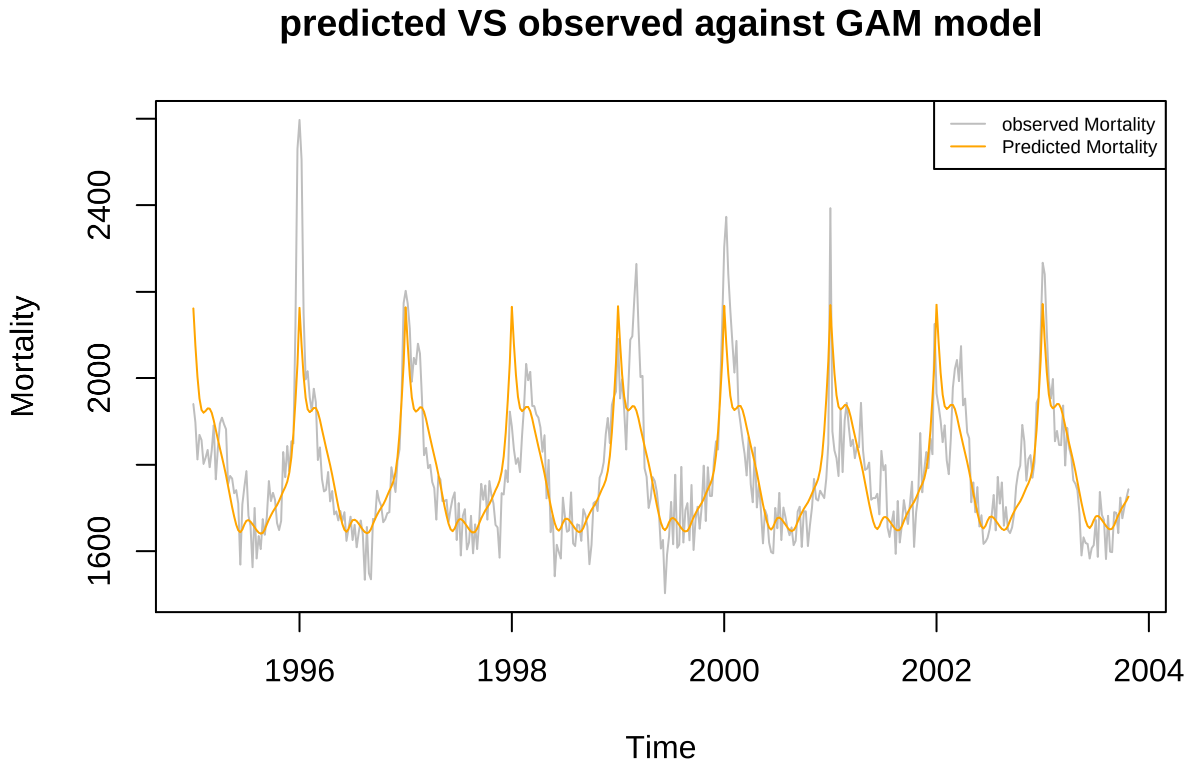
- g : Link fumction

```
- $s_i(X)$ : smoother, normally splines
```

1.3

1.3.a. predicted and observed mortality against time

1.3 Plot predicted and observed mortality against the output of the GAM model.



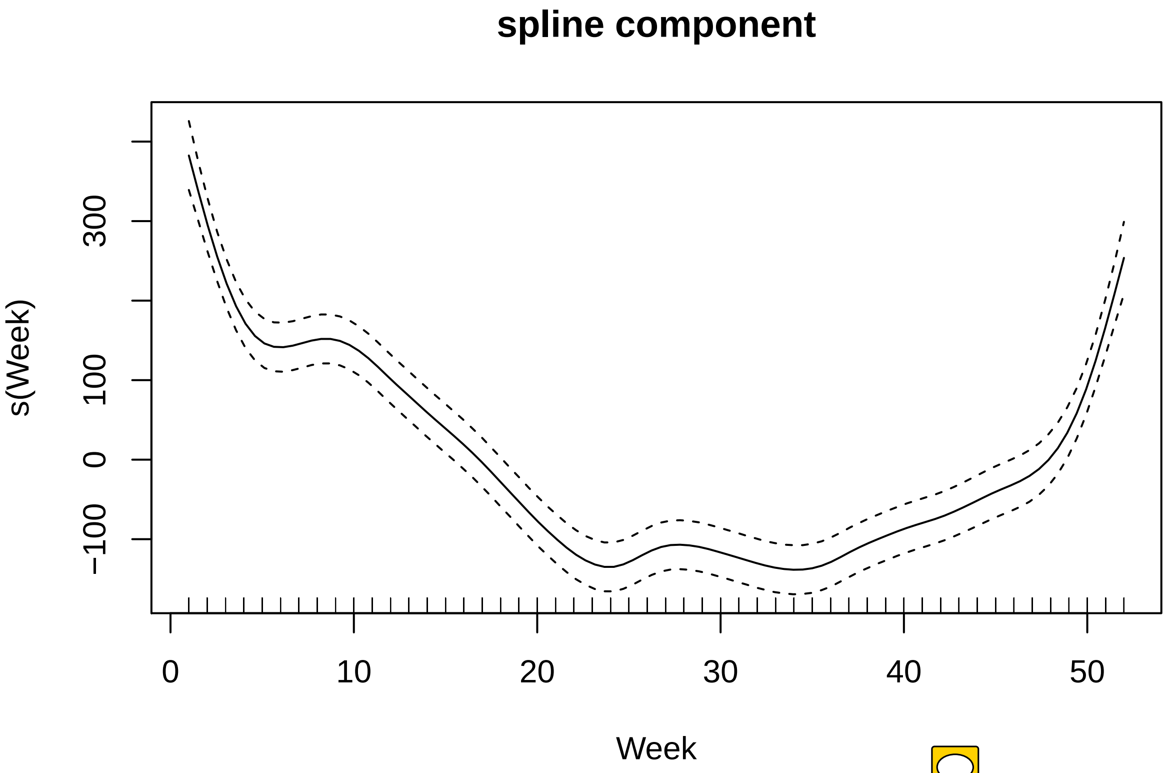
The quality of fit can be explained by adjusted R-squre value. The GAM model roughly fit the original data and we can see that the predicted values have the same pattern as observed values.

1.3.b. Summary of the fitted model

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Mortality ~ Year + s(Week, k = length(unique(data$Week)))
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -681.281   3368.846  -0.202    0.840
## Year           1.233      1.685   0.732    0.465
##
## Approximate significance of smooth terms:
##           edf Ref.df     F p-value
## s(Week) 13.79     51 18.85  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.677   Deviance explained = 68.7%
## GCV = 8703.7  Scale est. = 8404.4    n = 459
```

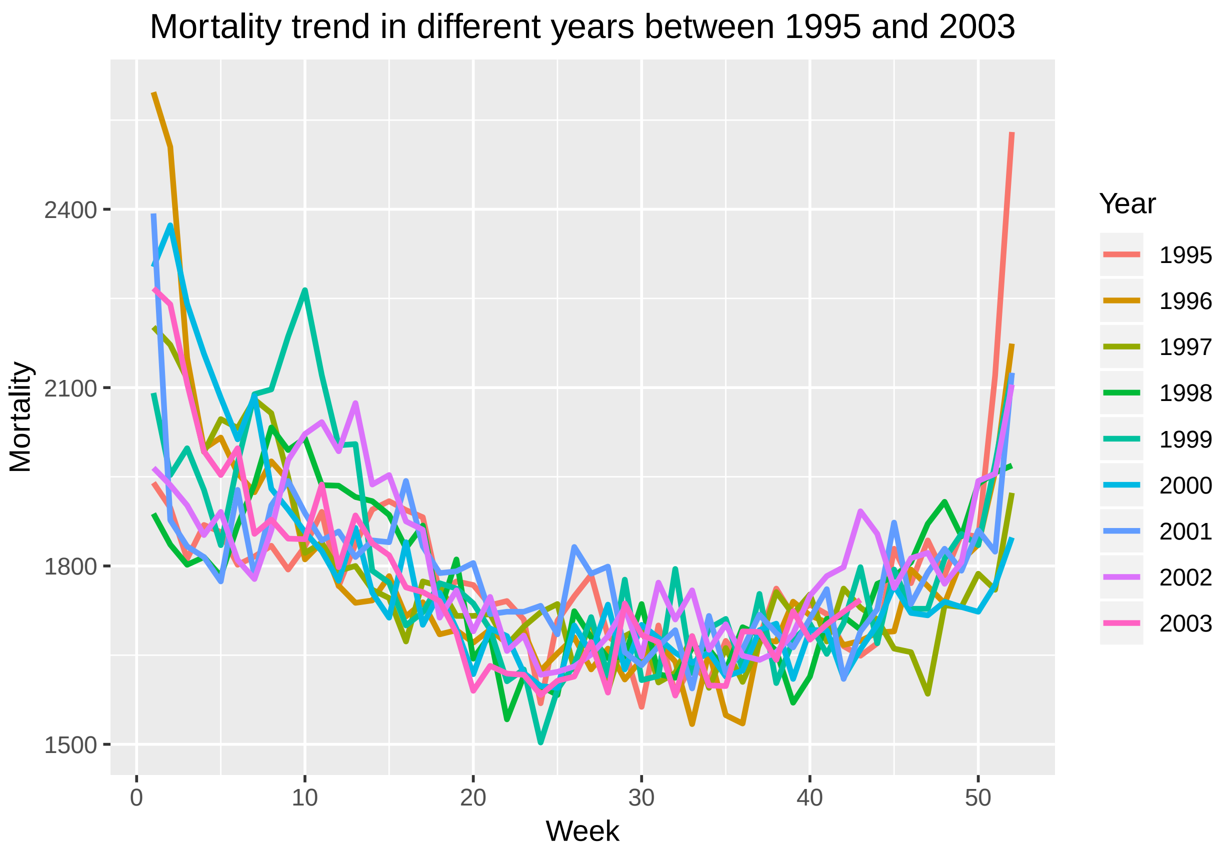
It can be clearly seen that the significance of smooth term is so high but p-value of Year is between 0.1 and 1; so the smooth term is more correlated to the Mortality. In other words, we can figure out that spline term s(Week) appear to be significant with p value close to 0.

1.3.c. spline component plot and Mortality trend in different years between 1995 and 2003



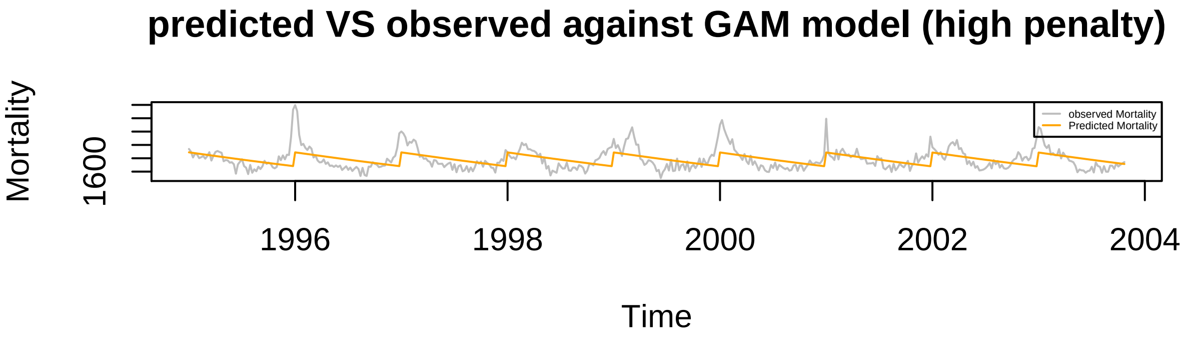
We know that there is a strong relation between s(Week) and mortality; so the plot of spline component should show the trend of mortality during a year from one week to another. Considering 52 weeks = 1 year from the plots above, we can say that there is a trend in mortality change from one year to another. First from the plot “predicted VS observed against GAM model”, we can observe that Mortality rate increases at the very begining/end of each year and decreases for the rest of year. From the plot “spline component”, we can also observe the similar trend. We can relate this trend to the fact that the given data is the cases of influenza in Sweden - influenza virus is more common during the winter season.

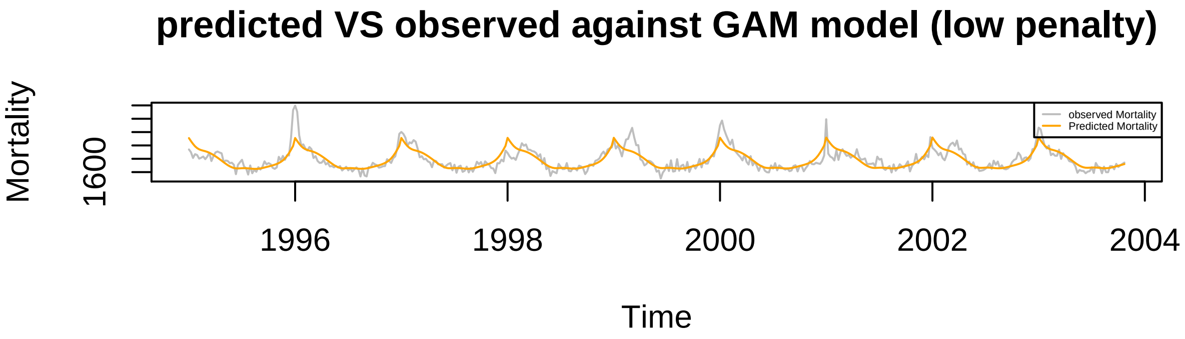
Plot below is the mortality trend in different years; we can obviousely see that the trend is same as the spline component plot!


Mortality trend in different years between 1995 and 2003
Mortality
2400
2100
1800
1500
0
10
20
30
40
50
Week
Year
1995
1996
1997
1998
1999
2000
2001
2002
2003

1.4. Penalization parameter and degree of freedom

1.4.a. High penalty vs low penalty in GAM

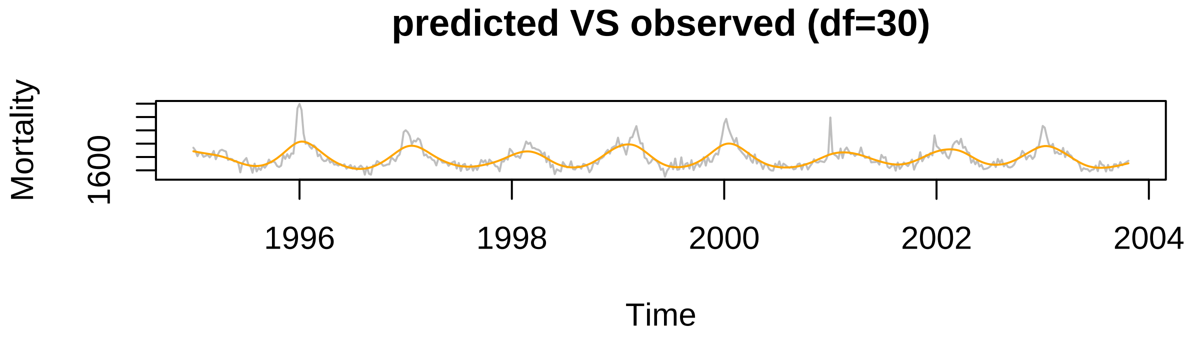


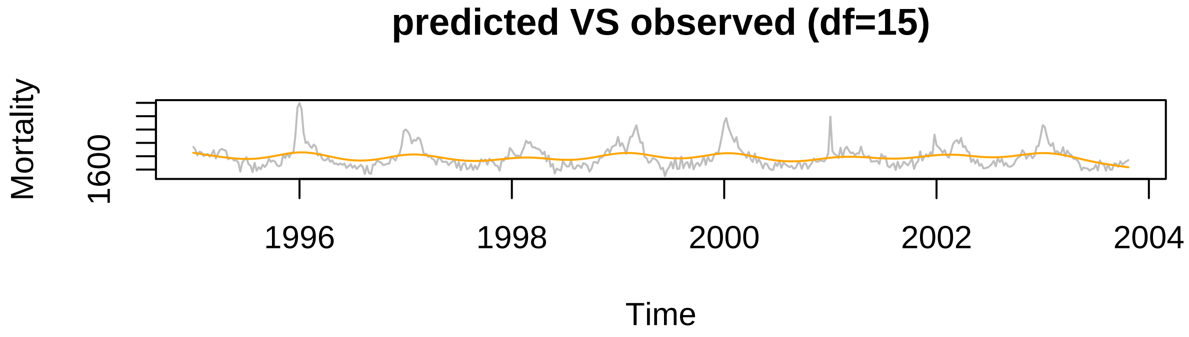


When the Penalization parameter increases the model will be over smoothed and the predicted values cannot follow the observed values.

To be able to show the relation between Penalization parameter and degree of freedom, we have shown the splines with different amount of degree of freedom here:

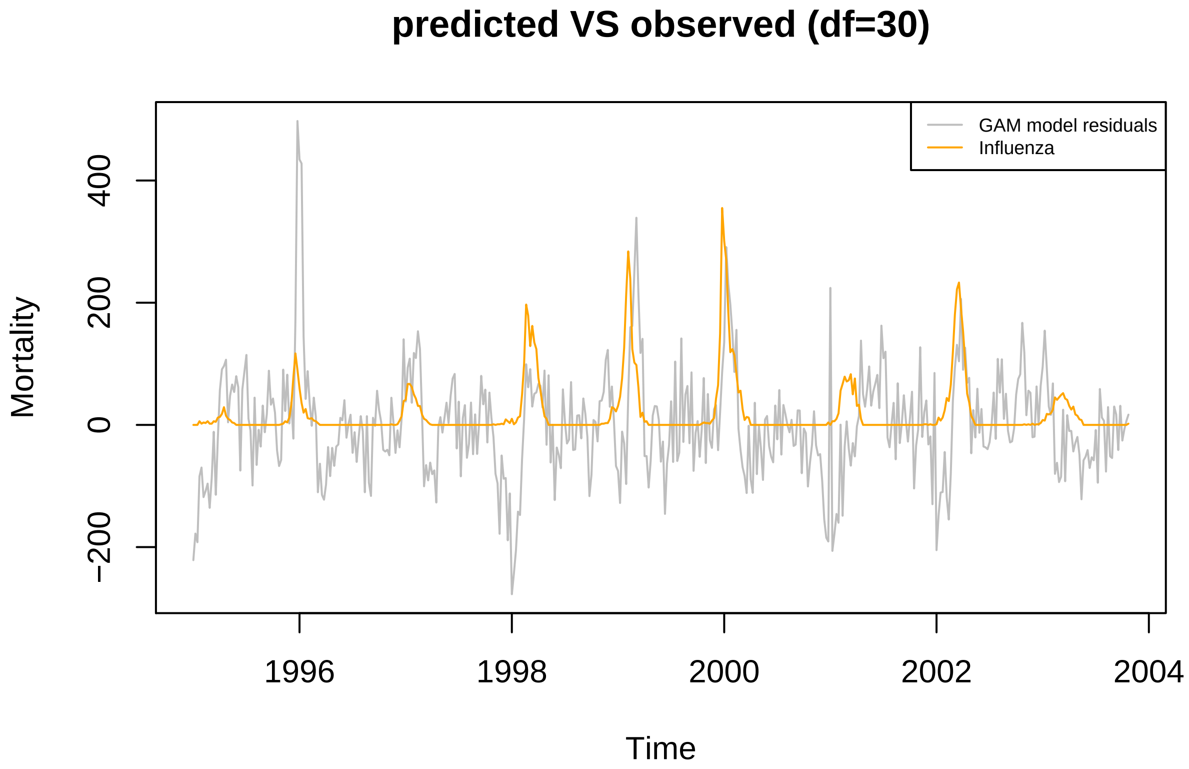
1.4.b. splines with different degrees of freedom





As you can see from the plots above, as degrees of freedom increases, the fitted value goes closer to the original value. However when it comes to penalty factor, it works in reverse way. As penalty factor grows, the fitted value goes closer to the original value which can result in over fitting.

1.5. Prediction residuals and observed Influenza against time



Temporal patterns can be seen when there were outbreaks of influenza with huge mortality, and for those period, the absolute value of residuals are comparably huge.

As we saw in question 1, the peaks of the mortality occurs in influenza outbreaks and since the smooth line couldn't capture the peaks of observed values, the peaks of residuals are also the same; again, we can confirm that there is a positive correlation between mortality and influenza.

1.6. Mortality as an additive function of the spline functions of year, week, and the number of confirmed cases of influenza

$$Mortality \sim N(\mu, \sigma^2)$$

$$E(Mortality) = \alpha + s_1(Year) + s_2(Week) + s_3(Influenza)$$

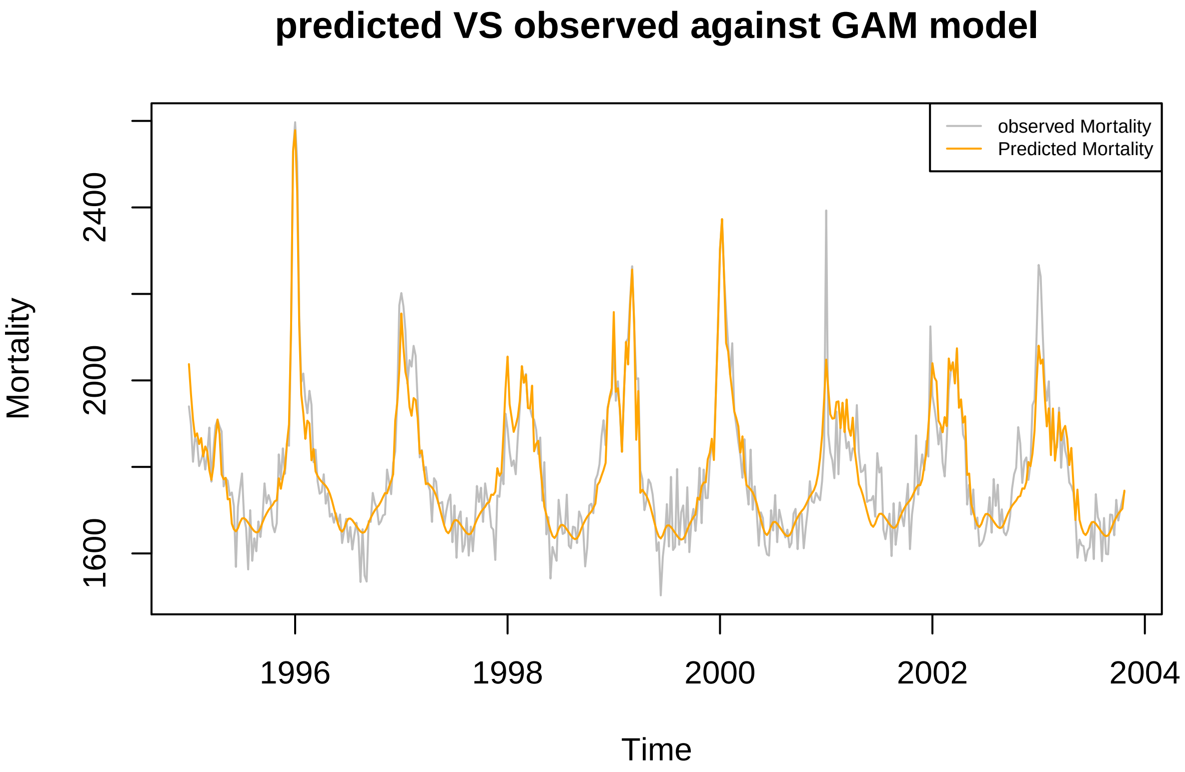
1.6.a. Summary of the model

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Mortality ~ s(Year, k = length(unique(data$Year))) + s(Week,
##     k = length(unique(data$Week))) + s(Influenza, k = length(unique(data$Influenza)))
##
```

```
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1783.765      3.198   557.8   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                 edf Ref.df      F p-value
## s(Year)       4.587  5.592  1.500   0.178
## s(Week)      14.431 17.990 18.763  <2e-16 ***
## s(Influenza) 70.094 72.998  5.622  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Rank: 134/144
## R-sq.(adj) =  0.819   Deviance explained = 85.4%
## GCV = 5840.5  Scale est. = 4693.7    n = 459
```

From the resulf above, we can see that by adding Influenza to the model and using spline for all features, the significance levels burgeoned! So we can conclude that the mortality is influenced by the outbreaks of influenza.

1.6.b. Predicted and observed mortality against time

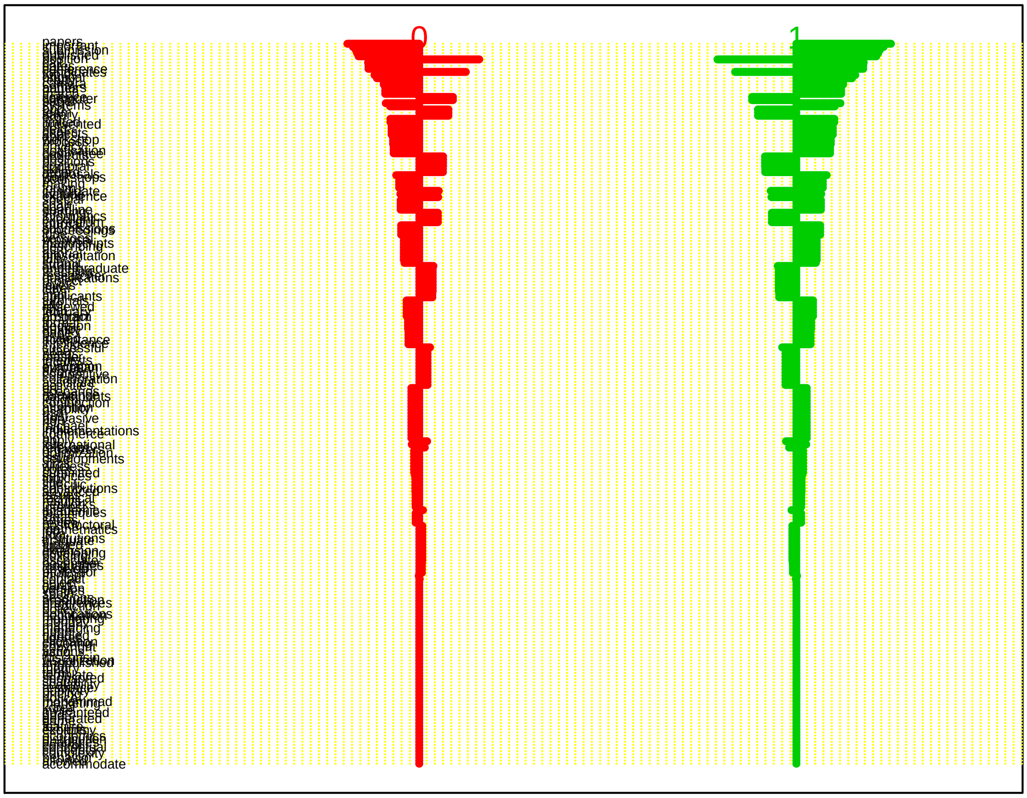


The adjusted R-sq is 0.762, and 77.6% of Deviance is explained with this model, where adjusted R-sq was 0.677 and 68.7% of Deviance is explained with previous model. So GAM model where mortality is be modelled as an additive function of the spline functions of year, week, and the number of confirmed cases of influenza seems perform better.

Assignment 2. High-dimensional methods

2.1. Nearest shrunken centroid classification of training data

Centroid plot with threshold equal to 1.306



1.306 is chosen by cross validation with the lowest misclassification error. By using 1.306 as threshold, 231 features have been selected and the CV error is 5.

The centroid plot for 0 and centroid plot for 1 is symmetric and the length of each line in the plot shows you how that certain data is far (deviated) from the centroid, and it's directrion shows rather it is positive or negative. So for example for the first variable "paper" , it is far from the centroid for both way but for 0 in negative way, for 1 in positve way so we can say they have counter effect on classifying into 0 and 1.

Top 10 contributing features

```
##                      0-score   1-score   av-rank-in-CV prop-selected-in-CV
##  [1,] "papers"     "-0.3814" "0.5018"  "2.3"         "1"
##  [2,] "important"  "-0.3519" "0.4631"  "3.3"         "1"
##  [3,] "submission" "-0.3368" "0.4431"  "4.1"         "1"
##  [4,] "due"        "-0.3301" "0.4344"  "4.35"        "1"
##  [5,] "published"  "-0.3223" "0.4241"  "5.2"         "1"
##  [6,] "position"   "0.318"   "-0.4184" "5.05"        "1"
##  [7,] "call"       "-0.2717" "0.3575"  "9.25"        "1"
##  [8,] "conference" "-0.2698" "0.355"   "8.4"         "1"
##  [9,] "dates"      "-0.2698" "0.355"   "9.2"         "1"
## [10,] "candidates" "0.2468"  "-0.3247" "9.65"        "1"
```

Listed names of 10 most contributing features are related to the conference mails. So we can say it is reasonable that they have strong effect on the discrimination betweem the conference mails and other mails.

Test error (misclassification rate)

```
## [1] 0.1
```

2.2. Elastic net and Spport Vector Machine (SVM)

```
- 2.2.a. test error of Elastic net with the binomial response and alpha = 0.5

- 2.2.b. test error of Support vector machine with "vanilladot" kernel
```

comparative table

```
##                   algorithm  CV_error test_error nonzeroCoefs
## 1 nearest shrunken centroid 5.0000000       0.10          231
## 2               Elastic net 1.1153559       0.20           32
## 3    Support vector machine 0.1388889       0.05           43
```

Frome the table above, we can check that SVM with "vanilladot" kernel shows the lowest test errors while EN shows the lowest number of features selected. Compare to other methods, test error for the SVM method is lowest and its number of of features selected is still not the best but seems good enough. So in this case, it is reasonable to choose SVM method for this data.

2.3. Hypothesis testing (Benjamini-Hochberg method)

Number of features in rejection region

```
## [1] 4663
```

Based on Benjamini-Hochberg method, 4663 features do not affect the response and just using 39 remaining variables in modeling is reasonable.

List of features within the confidence interval

```
##          feature      P_value
## 1         papers 1.116910e-10
## 2     submission 7.949969e-10
## 3       position 8.219362e-09
## 4      published 1.835157e-07
## 5      important 3.040833e-07
## 6           call 3.983540e-07
## 7     conference 5.091970e-07
## 8     candidates 8.612259e-07
## 9          dates 1.398619e-06
## 10         paper 1.398619e-06
## 11        topics 5.068373e-06
## 12       limited 7.907976e-06
## 13     candidate 1.190607e-05
## 14        camera 2.099119e-05
## 15         ready 2.099119e-05
## 16       authors 2.154461e-05
## 17           phd 3.382671e-05
## 18      projects 3.499123e-05
## 19           org 3.742010e-05
## 20        chairs 5.860175e-05
## 21           due 6.488781e-05
## 22      original 6.488781e-05
## 23  notification 6.882210e-05
## 24        salary 7.971981e-05
## 25        record 9.090038e-05
## 26        skills 9.090038e-05
## 27          held 1.529174e-04
## 28          team 1.757570e-04
## 29         pages 2.007353e-04
## 30      workshop 2.007353e-04
## 31     committee 2.117020e-04
## 32   proceedings 2.117020e-04
## 33         apply 2.166414e-04
## 34        strong 2.246309e-04
## 35 international 2.295684e-04
## 36        degree 3.762328e-04
## 37     excellent 3.762328e-04
## 38          post 3.762328e-04
## 39     presented 3.765147e-04
```

We can see that 10 most contributing features from step 1 are also included.

Appendix

```
library(readxl)
library(ggplot2)
data <- read_excel("influenza.xlsx", sheet = 1)

plot.ts(data$Time, data$Mortality, type="l",
        main="Mortality time series plot", xlab="Time(weekly)", ylab="Mortality")
plot.ts(data$Time, data$Influenza, type="l",
        main="Influenza time series plot", xlab="Time(weekly)", ylab="Influenza")


library(mgcv)
gam_model <- gam(Mortality ~ Year + s(Week, k=length(unique(data$Week))),
                 data=data, family="gaussian", select=TRUE, method="GCV.Cp")
#k:amount of unique values of variables in smoothing spline
#gam in mgcv solves the smoothing parameter estimation problem
#by using the Generalized Cross Validation (GCV) criterion
#The smoothing parameter estimation method. "GCV.Cp" to
#use GCV for unknown scale parameter


plot.ts(data$Time, data$Mortality, type="l", col="grey",
        main="predicted VS observed against GAM model",
        xlab="Time", ylab="Mortality")
points(data$Time, gam_model$fitted.values, type="l", col="orange")
legend("topright", legend=c("observed Mortality","Predicted Mortality"),
       col = c("grey","orange"), lty = c(1,1), cex = 0.59)

summary(gam_model)
```

```
plot(gam_model,main="spline component",ylab="s(Week)")
#seeing pattern
##-1.3.c
data3 <- data
data3$Year <- as.factor(data3$Year)
ggplot(data = data3) +
geom_line(mapping = aes(x = Week, y = Mortality, color = Year)
          , size = 1) +
ggtitle('Mortality trend in different years between 1995 and 2003') +
theme(plot.title = element_text(hjust = 0.5))
## 1.4. Penalization parameter and degree of freedom

### 1.4.a. High penalty vs low penalty in GAM
par(mfrow=c(2,1))
gam_model_hp <- gam(Mortality ~ Year + s(Week, k=length(unique(data$Week)),
                                         sp=c(50,50)),
                 data=data, family="gaussian", select=TRUE, method="GCV.Cp"
                 )

plot.ts(data$Time, data$Mortality, type="l", col="grey",
        main="predicted VS observed against GAM model (high penalty)",
        xlab="Time", ylab="Mortality")
points(data$Time, gam_model_hp$fitted.values, type="l", col="orange")
legend("topright", legend=c("observed Mortality","Predicted Mortality"),
       col = c("grey","orange"), lty = c(1,1), cex = 0.3)

#with very low penalty
gam_model_lp <- gam(Mortality ~ Year + s(Week, k=length(unique(data$Week)),
                                         sp=c(0.001,0.001)),
                 data=data, family="gaussian", select=TRUE, method="GCV.Cp"
                 )

plot.ts(data$Time, data$Mortality, type="l", col="grey",
        main="predicted VS observed against GAM model (low penalty)",
        xlab="Time", ylab="Mortality")
points(data$Time, gam_model_lp$fitted.values, type="l", col="orange")
legend("topright", legend=c("observed Mortality","Predicted Mortality"),
       col = c("grey","orange"), lty = c(1,1), cex = 0.3)

par(mfrow=c(2,1))
res1 <- smooth.spline(data$Time,data$Mortality,df=15)
#predict(res1,x=data$Time)$y

res2 <- smooth.spline(data$Time,data$Mortality,df=30)
#predict(res2,x=data$Time)$y


plot.ts(data$Time, data$Mortality, type="l", col="grey",
        main="predicted VS observed (df=30)",
        xlab="Time", ylab="Mortality")
points(data$Time, predict(res2,x=data$Time)$y, type="l", col="orange")

plot.ts(data$Time, data$Mortality, type="l", col="grey",
        main="predicted VS observed (df=15)",
        xlab="Time", ylab="Mortality")
points(data$Time, predict(res1,x=data$Time)$y, type="l", col="orange")

gam_model <- gam(Mortality ~ Year + s(Week, k=length(unique(data$Week))),
                 data=data, family="gaussian", select=TRUE, method="GCV.Cp")

plot.ts(data$Time, gam_model$residuals, type="l", col="grey",
        main="predicted VS observed (df=30)",
        xlab="Time", ylab="Mortality")
points(data$Time, data$Influenza, type="l", col="orange")
legend("topright", legend=c("GAM model residuals","Influenza"),
       col = c("grey","orange"), lty = c(1,1), cex = 0.59)


gam_model2 <- gam(Mortality ~ s(Year, k=length(unique(data$Year))) +
                          s(Week, k=length(unique(data$Week))) +
                          s(Influenza,k=length(unique(data$Influenza))),
                  data=data)#, family="gaussian", select=TRUE, method="GCV.Cp")
summary(gam_model2)
plot.ts(data$Time, data$Mortality, type="l", col="grey",
        main="predicted VS observed against GAM model",
        xlab="Time", ylab="Mortality")
points(data$Time, gam_model2$fitted.values, type="l", col="orange")
legend("topright", legend=c("observed Mortality","Predicted Mortality"),
       col = c("grey","orange"), lty = c(1,1), cex = 0.59)

## Assignment 2. High-dimensional methods
library(pamr)
library(glmnet)
library(kernlab)
data <- read.csv2('data.csv')
set.seed(12345)
n = dim(data)[1]
set.seed(12345)
id = sample(1:n, floor(n*0.7))
train0 = data[id,]
test0 = data[-id,]

### 2.1. Nearest shrunken centroid classification of training data
### --- Fitting model
train = train0
train$conference = as.factor(train0$conference)
rownames(train) = 1:nrow(train)
x = t(train[,-4703])
x0 <- matrix(as.numeric(x), ncol = 44, nrow = 4702)
rownames(x0) <- rownames(x)
y = train[[4703]]
train_f = list(x=x0
              ,y=as.factor(y)
              ,geneid=as.character(1:nrow(x0))
              ,genenames=rownames(x0))
```

```
classifier <- pamr.train(data = train_f)

###--Cross validation
###-- Cross validation

cv_model <- pamr.cv(classifier,train_f)

###-- Centroid plot
pamr.plotcen(fit = classifier, train_f, threshold = 1.306)

#### Top 10 contributing features
a <- pamr.listgenes(classifier, data = train_f,fitcv = cv_model, threshold = 1.306)
cbind(train_f$genenames[as.numeric(a[1:10,1])],a[1:10,-1])
##-- Test error
###-- Preparing test data
test = test0
test$conference = as.factor(test0$conference)
rownames(test) = 1:nrow(test)
x = t(test[,-4703])
x0 <- matrix(as.numeric(x), ncol = 20, nrow = 4702)
rownames(x0) <- rownames(x)
y = test[[4703]]
test_f = list(x=x0
               ,y=as.factor(y)
               ,geneid=as.character(1:nrow(x0))
               ,genenames=rownames(x0))

###-- misclassification rate
test_pred <- pamr.predict(fit = classifier, newx = test_f$x
             ,threshold = 2,type = 'class')

cm <- table(test_f$y, test_pred)
centroid_error <- (cm[1,2] + cm[2,1])/sum(cm)
centroid_error

## 2.2. Elastic net and Spport Vector Machine (SVM)
### 2.2.a. Elastic net
x = as.matrix(train0[,-4703])
y = as.factor(train0[,4703])
####-- Finding best lambda by cross validation
cv_glmnet <- cv.glmnet(x = x ,y = y
                       ,family = 'binomial'
                       ,alpha = 0.5)
lambda_min <- cv_glmnet$lambda.min
y_hat_test <- predict(object = cv_glmnet
                      , newx = as.matrix(test0[,-4703])
                      , type = 'class'
                      , lambda = lambda_min)
y_hat_test <- as.numeric(y_hat_test)
net_cv_Error <- cv_glmnet$cvm[which(cv_glmnet$lambda == lambda_min)]
net_nonzeroCoefs <- cv_glmnet$nzero[which(cv_glmnet$lambda == lambda_min)]
```

```
####-- misclassification rate
cm <- table(as.factor(test0[,4703]),y_hat_test)
net_error <- (cm[1,2] + cm[2,1])/sum(cm)

### 2.2.b. Support Vector Machine
test_set <- as.matrix(test0[,-4703])
test_response <- as.factor(test0[,4703])
####-- Fitting model
svm <- ksvm(x = x, y = y
            ,kernel = vanilladot()
            ,type = 'C-svc'
            ,cross = 5)
svm_cv_error <- cross(svm)
svm_features <- length(coef(svm)[[1]])
####-- Misclassification rate
y_hat_svm <- predict(object = svm,test_set)
cm <- table(test_response,y_hat_svm)
svm_error <- (cm[1,2]+cm[2,1])/sum(cm)
table <- data.frame(algorithm = c('nearest shrunken centroid'
                                  ,'Elastic net'
                                  ,'Support vector machine')
                    ,CV_error = c(5,net_cv_Error,svm_cv_error)
                    ,test_error = c(centroid_error,net_error,svm_error)
                    ,nonzeroCoefs =c(231,net_nonzeroCoefs,svm_features) )
table


## 2.3. Hypothesis testing
y <- as.factor(data[,4703])
x <- as.matrix(data[,-4703])

p_values <- data.frame(feature = '',P_value = 0,stringsAsFactors = FALSE)
for(i in 1:ncol(x)){
  res = t.test(x[,i]~y, data = data,
               alternative="two.sided"
               ,conf.level = 0.95)
  p_values[i,] <- c(colnames(x)[i],res$p.value)

}
p_values$P_value <- as.numeric(p_values$P_value)

p <- p.adjust(p_values$P_value, method = 'BH')

length(p[which(p > 0.05)])
out <- p_values[which(p <= 0.05),]
out <- out[order(out$P_value),]
rownames(out) <- NULL
out
```