Computer lab 2 block 2

Biswas Kumar, bisku859

12/03/2020

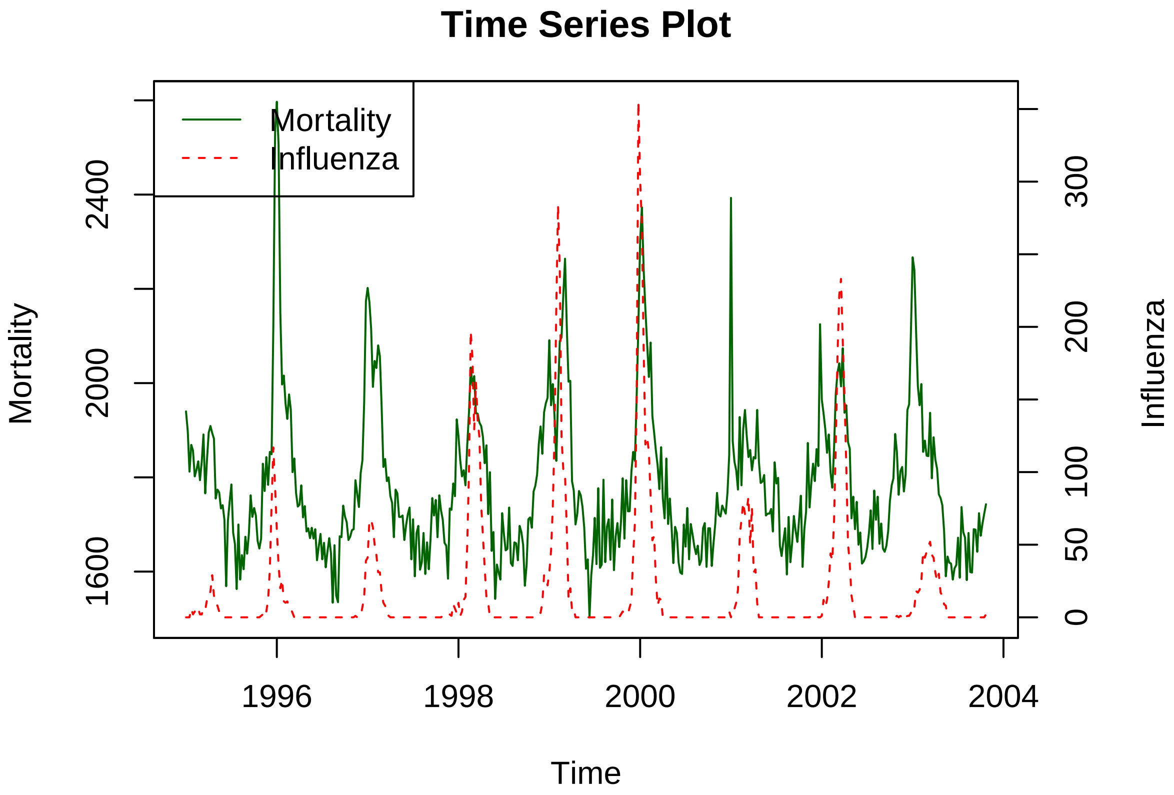
Assignment 1. Using GAM and GLM to examine the mortality rates

The Excel document 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 (temperature deficits).

1. Use time series plots to visually inspect how the mortality and influenza number vary with time (use Time as X axis). By using this plot, comment how the amounts of influenza cases are related to mortality rates.

Solution

Step 1 : Importing excel document influenza.xlsx



On analysing above plot, we see that whenever there is spike in Influenza , the mortality graph too rises and vice -versa. Though, the graph does not shows the proportional increase or decrease but somewhat captures the spikes between both Influenza and mortality at the same time.

2.Use gam() function from mgcv package to fit a GAM model in which Mortality is normally distributed and modelled as a linear function of Year and spline function of Week, and make sure that the model parameters are selected by the generalized cross-validation. Report the underlying probabilistic model.

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Mortality ~ Year + s(Week, k = length(unique(data$Week)))
##
## Estimated degrees of freedom:
## 14.3  total = 16.32
##
## GCV score: 8708.581     rank: 52/53
```

```
##
## 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) -680.598   3367.760  -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) 14.32  17.87 53.86  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Rank: 52/53
## R-sq.(adj) =  0.677   Deviance explained = 68.8%
## GCV = 8708.6  Scale est. = 8398.9    n = 459
```

Looking at the function, the probabilistic model is given by :

Mortality~N(w0+w1*Year+s(Week), sigmaˆ2) (#as comment received for our group report).

We can also write the model as below : $Mortality = w_0 + w_1 Year + s(Week) + error$

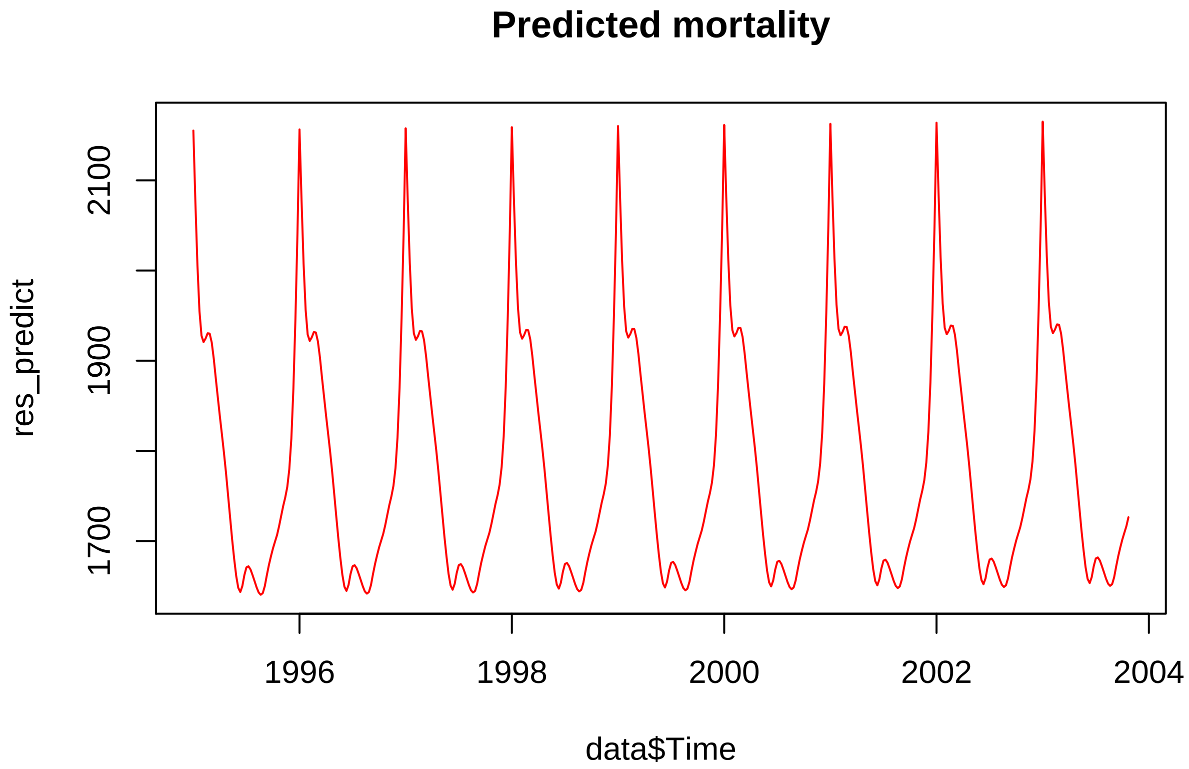
Upon substituting values in the above equation , we get : $Mortality = -680.598 + 1.233 Year + s(Week) + error$

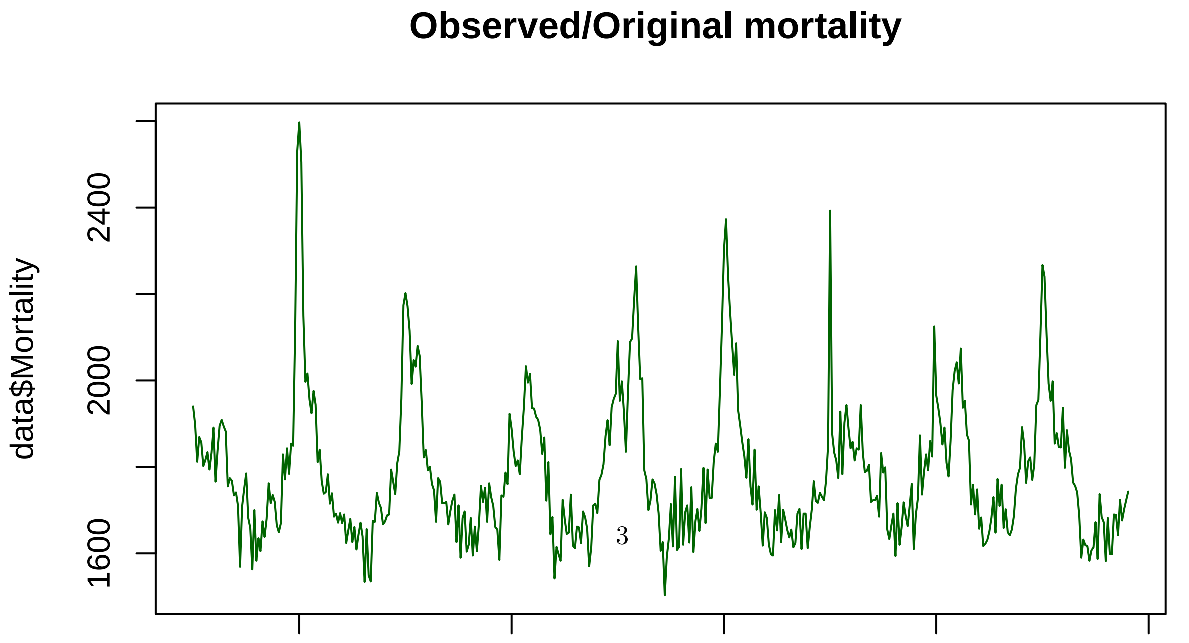
3.Plot predicted and observed mortality against time for the fitted model and comment on the quality of the fit.

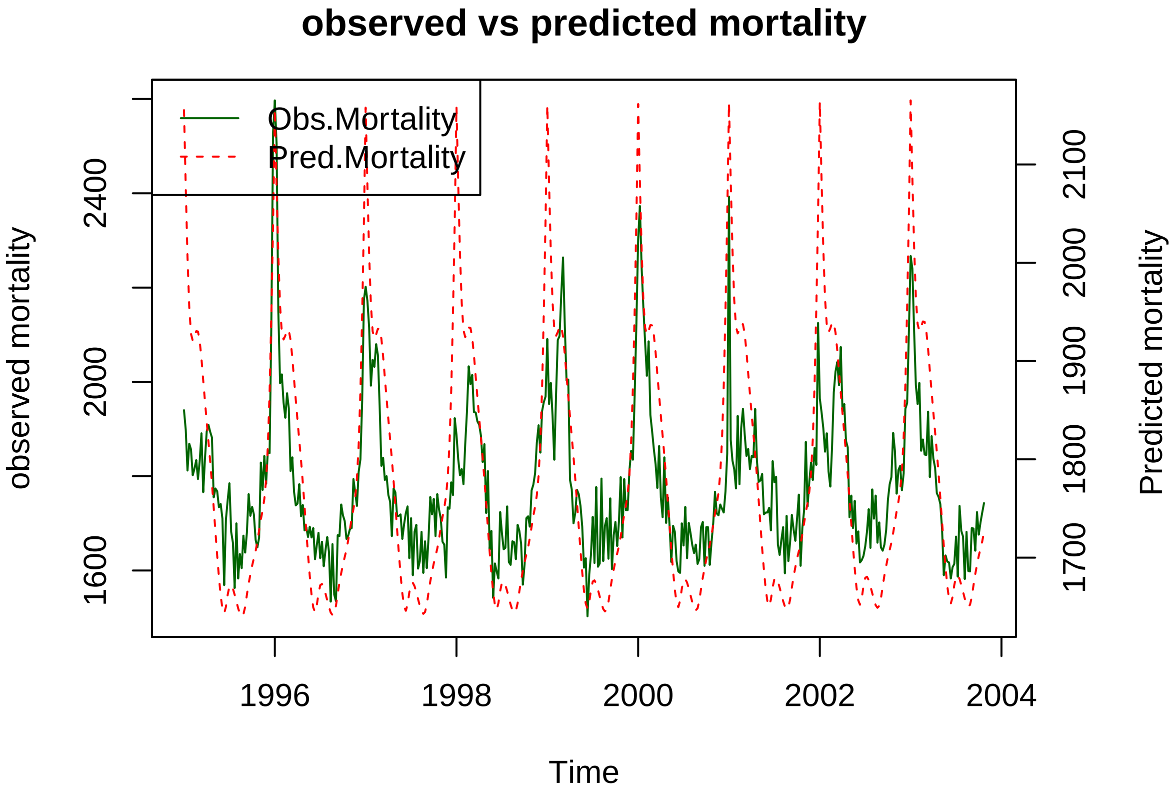
Investigate the output of the GAM model and report which terms appear to be significant in the model.

Is there a trend in mortality change from one year to another?

Plot the spline component and interpret the plot.





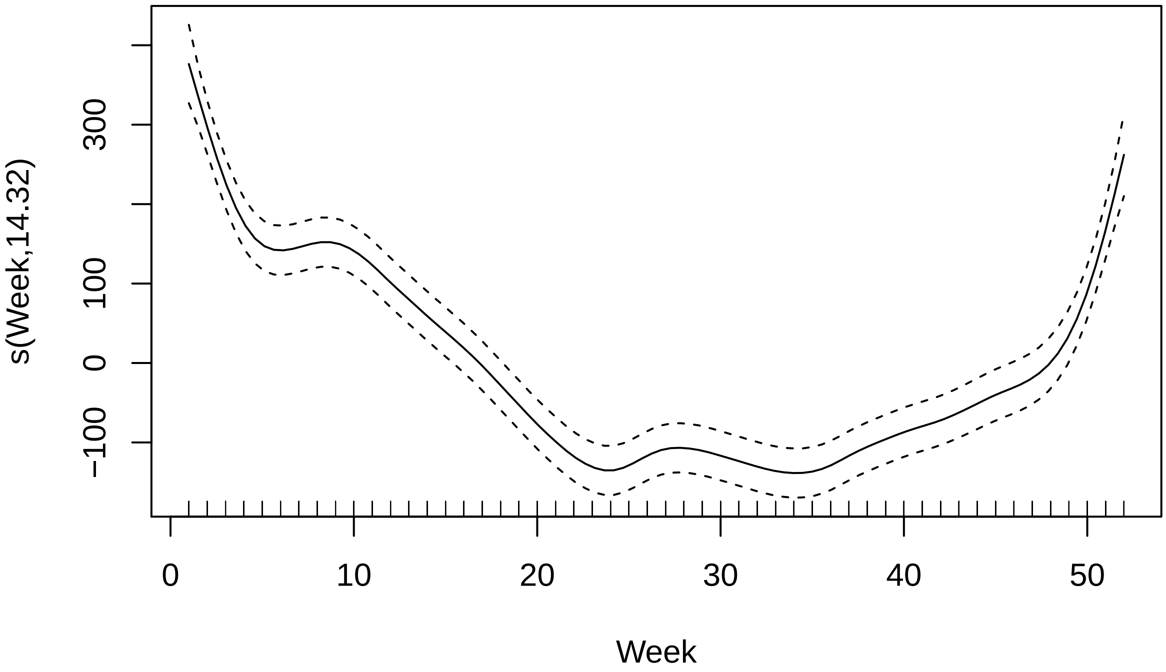


Quality of fit: Observing plot of observed /actual vs prediction as well as the residual calculation, we get that the predicted value is too high during spike and has lower lows. It implies that there is some error which is captured during peaks and lows that needs to be further smoothen with the use of higher spline model.

It can be interpreted that the model is fitting well with some outliers(or errors) at extreme.

On investigating summary(res),the GAM function, we find that the spline function of week is most significant as the same has been highlighted with 3 stars.

Now, plotting the spline component below:



The plotting of spline function which is used for smoothing of graph, highlights that it is minimum when week intervals are between 25-35.

4. Examine how the penalty factor of the spline function in the GAM model from step 2 influences the estimated deviance of the model.

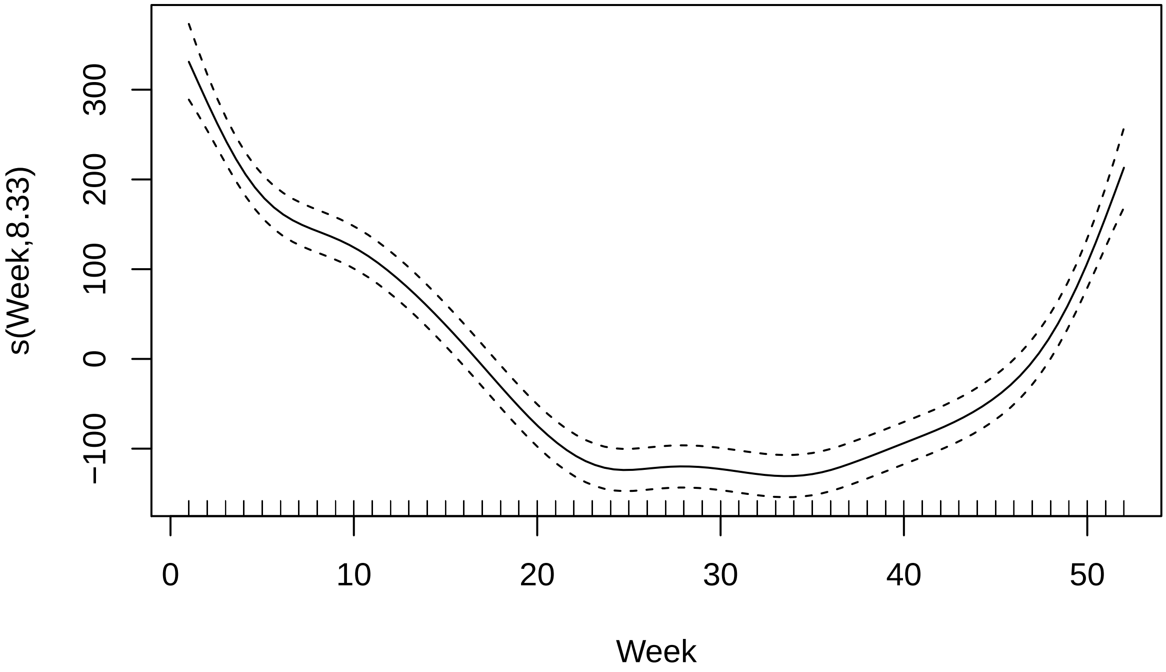
Make plots of the predicted and observed mortality against time for cases of very high and very low penalty factors.

What is the relation of the penalty factor to the degrees of freedom? Do your results confirm this relationship?

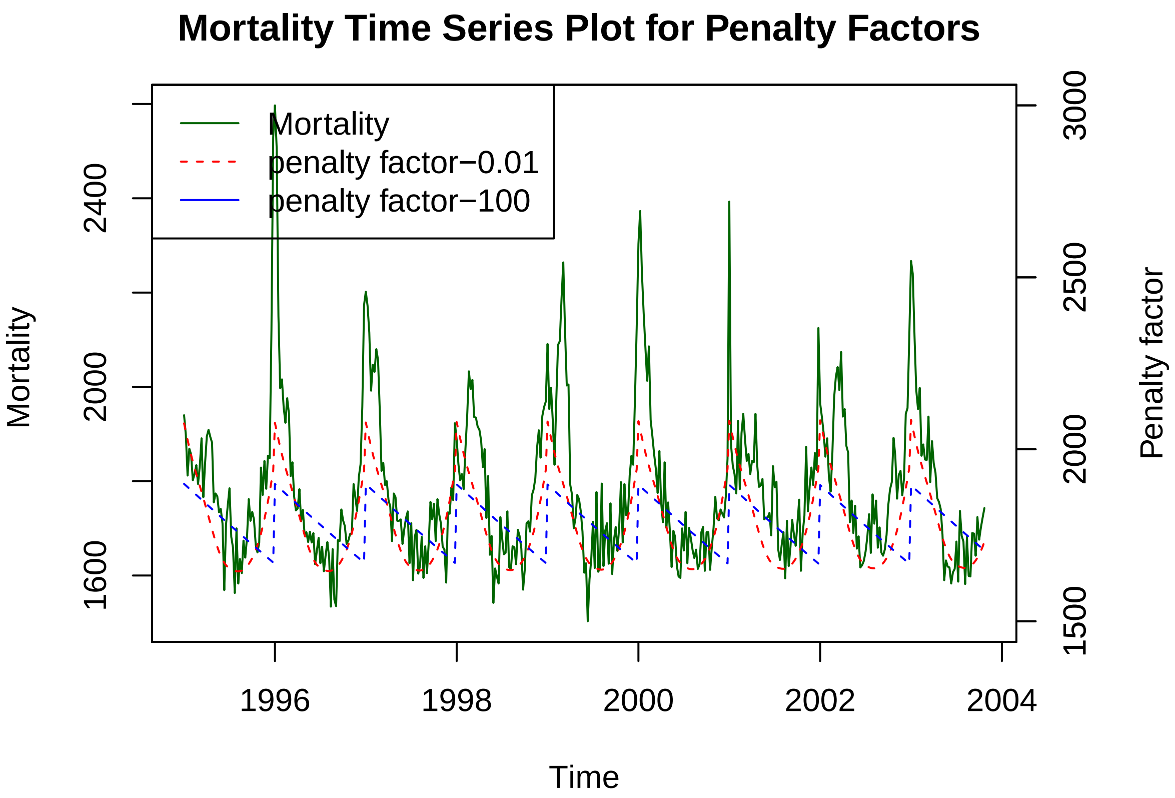
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Mortality ~ Year + s(Week, k = length(unique(data$Week)), sp = 0.001)
##
## Estimated degrees of freedom:
## 8.33  total = 10.33
##
## GCV score: 8979.703
```

```
##
```

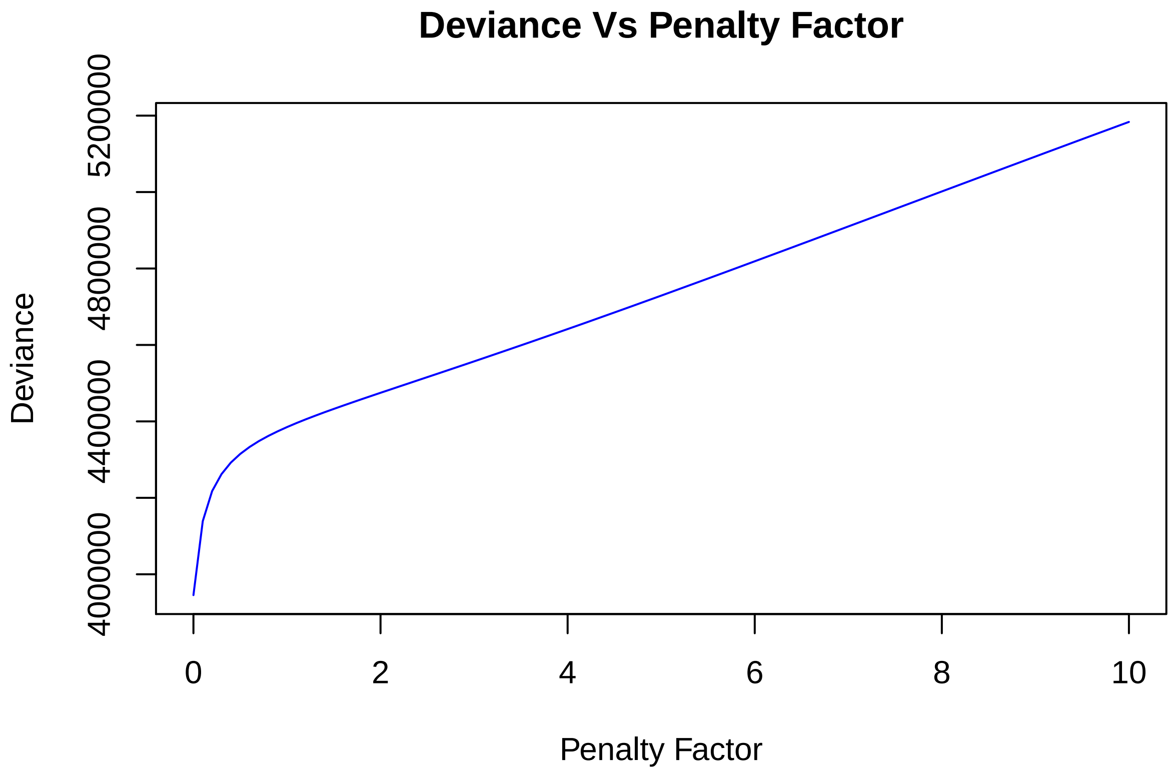
```
## Family: gaussian
## Link function: identity
##
## Formula:
## Mortality ~ Year + s(Week, k = length(unique(data$Week)), sp = 0.001)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -745.837   3442.424  -0.217    0.829
## Year           1.265      1.722   0.735    0.463
##
## Approximate significance of smooth terms:
##           edf Ref.df     F p-value
## s(Week) 8.333  10.41 85.18  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.662   Deviance explained = 66.9%
## GCV = 8979.7  Scale est. = 8777.6    n = 459
```

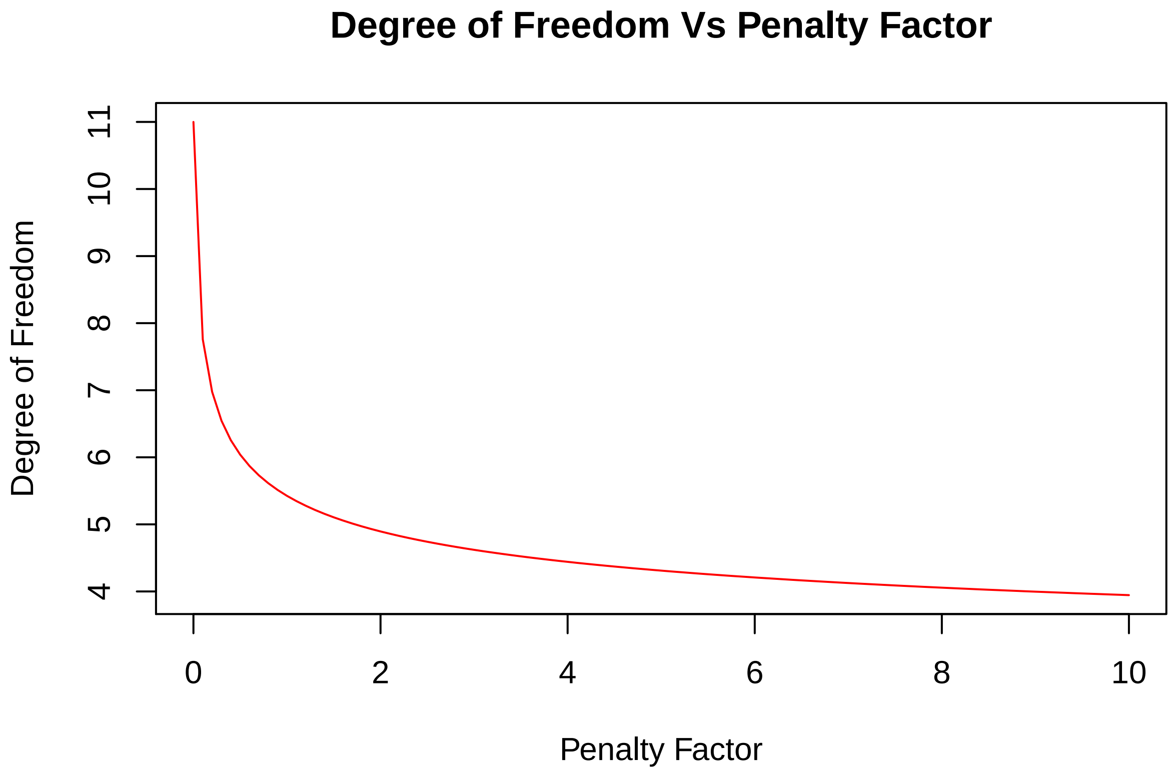


```
## [1] 3938200
```



Looking at graph we see that , low penalty factor (such as 0.01) tends to correlate/follow the spike and lows better than the high penalty factors. This means that high penalty factors though smoothens the graph in this case, gives higher error when compared with the predicted values.


Deviance Vs Penalty Factor
Deviance
4000000
4400000
4800000
5200000
0
2
4
6
8
10
Penalty Factor

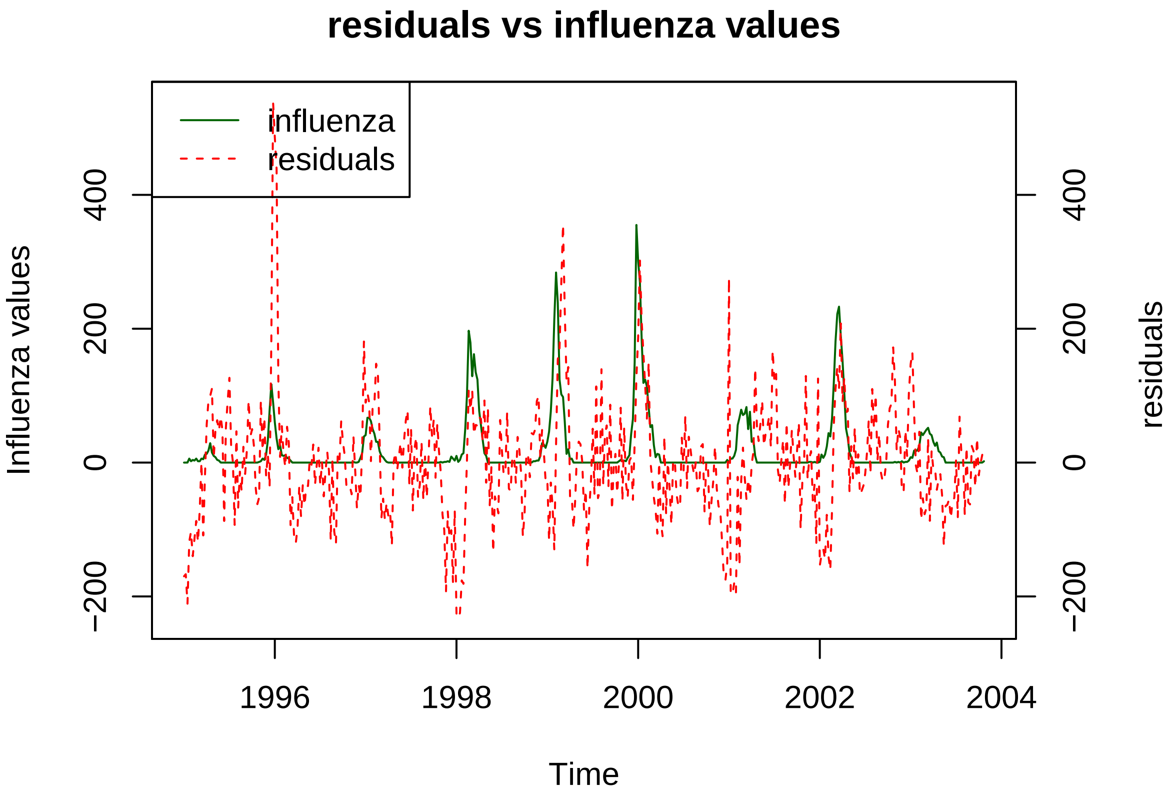


Looking at the (Deviance Vs Penalty Factor) plot above, we can say that Deviance goes up with penalty factor.

On the other hand, the visualization of the graph (Degree of freedom (DoF) vs penalty factor), we find that DoF decrease drastically with increase in penalty factor. It highlights that Dof and penalty factor has an inverse relationship.

5.Use the model obtained in step 2 and plot the residuals and the influenza values against time (in one plot).

Is the temporal pattern in the residuals correlated to the outbreaks of influenza?



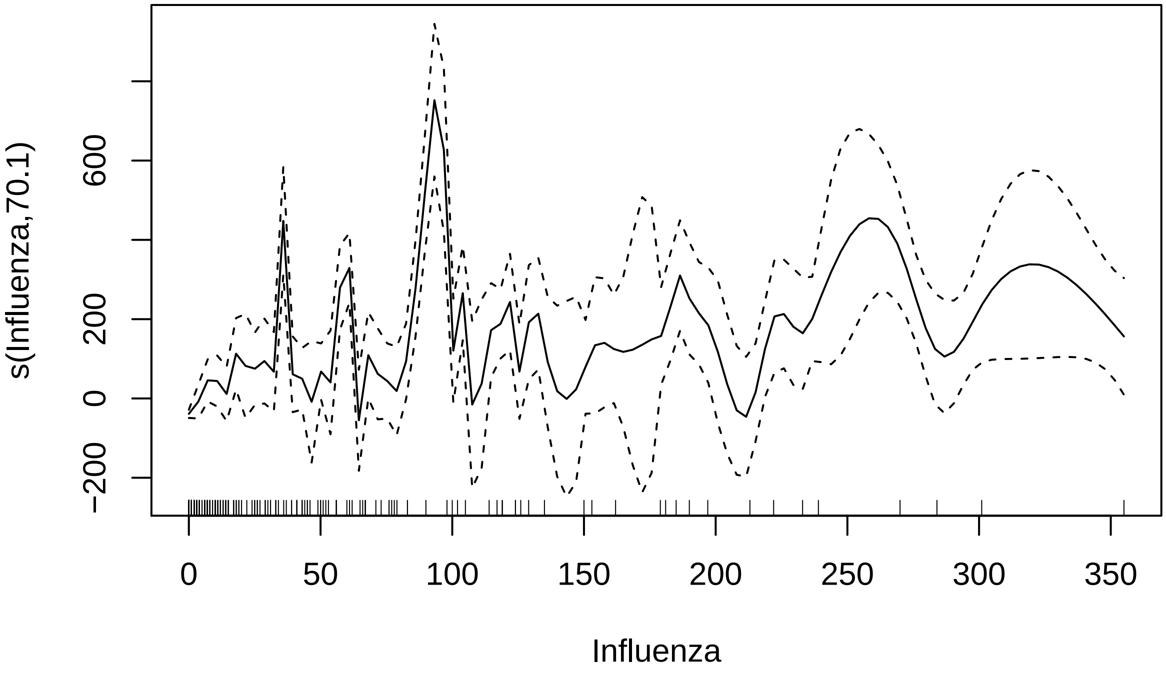
Looking at the graph, we can say that the residuals (in red) do spikes when there is a spike in the influenza value (or the outbreak). However, it does not give an accurate or even picture for the relative magnitude of spike.

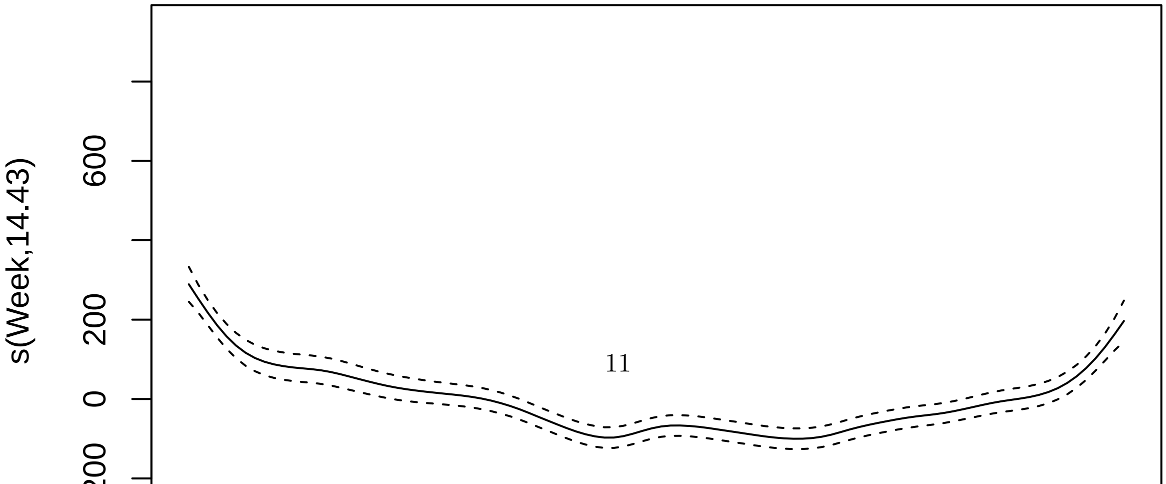
6.Fit a GAM model in R in which mortality is be modelled as an additive function of the spline functions of year, week,

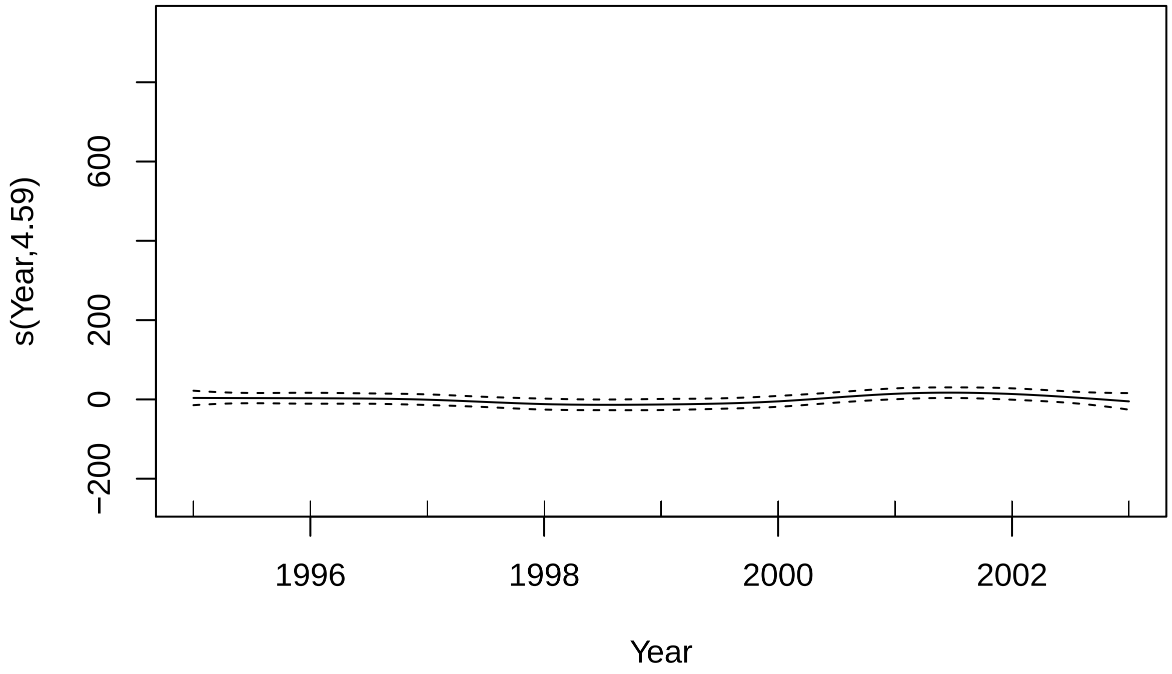
and the number of confirmed cases of influenza. Use the output of this GAM function to conclude whether or not

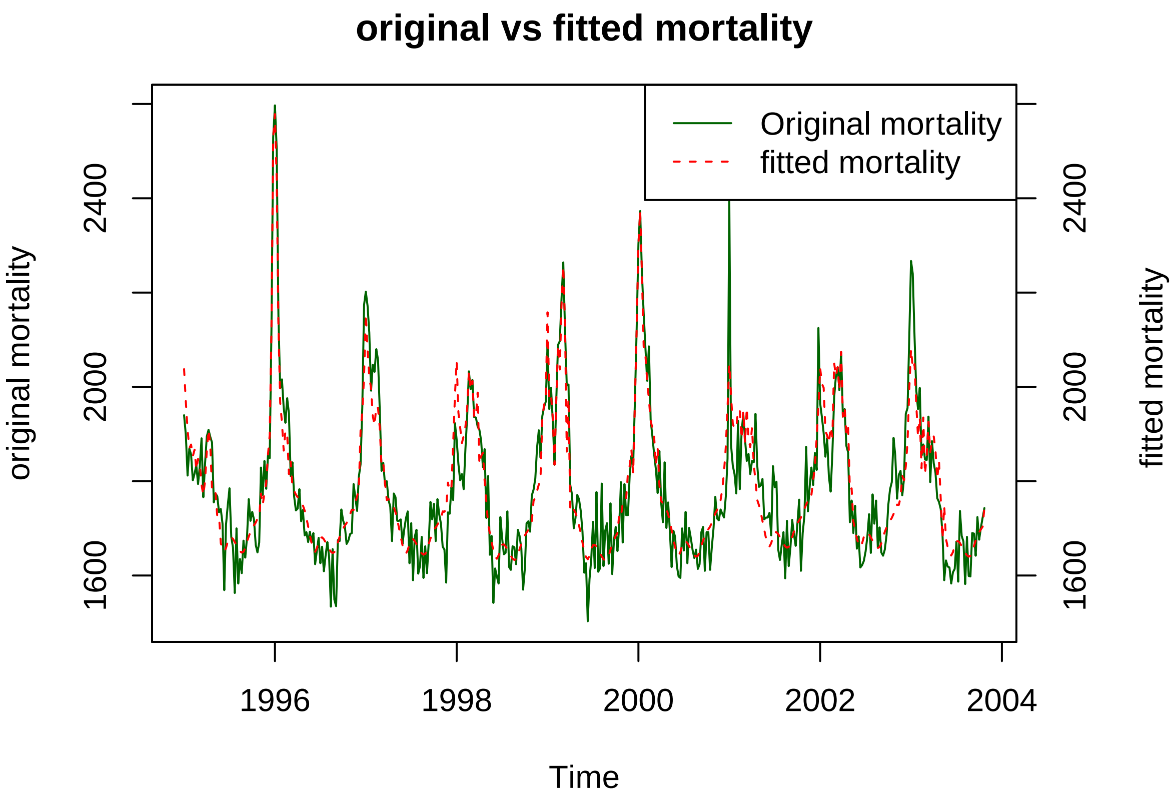
the mortality is influenced by the outbreaks of influenza. Provide the plot of the original and fitted Mortality

against Time and comment whether the model seems to be better than the previous GAM models.






s(Year,4.59)
−200
0
200
600
1996
1998
2000
2002
Year



```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Mortality ~ s(Influenza, k = length(unique(data$Influenza))) +
##     s(Week, k = length(unique(data$Week))) + s(Year, k = length(unique(data$Year)))
##
## 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(Influenza) 70.103 72.997  5.623  <2e-16 ***
## s(Week)      14.431 17.990 18.763  <2e-16 ***
## s(Year)       4.587  5.592  1.500   0.178
## ---
## 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
```

The above plot is capturing the spikes in the motality (original vs fitted) along time with higher accuracy than the previous one. It may also be noted that all the periodic spikes are been captured with better curve response at both extreme (i.e highs and lows) when compared with previous fit.

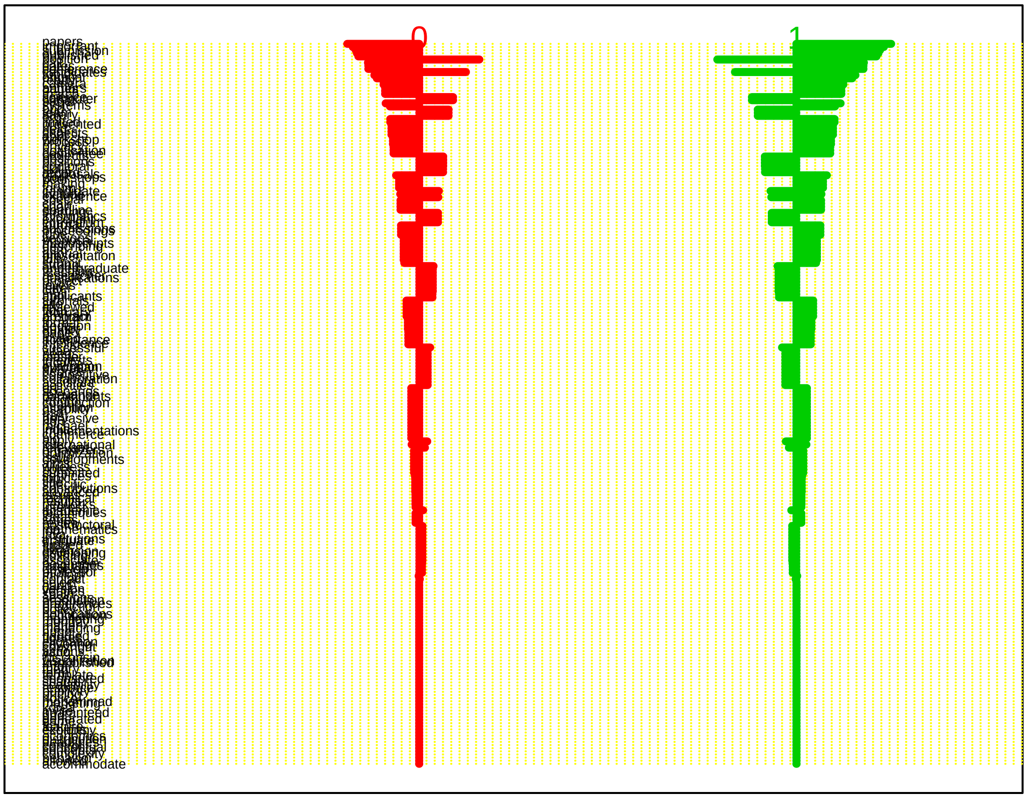
Assignment 2. High-dimensional methods

The data file data.csv contains information about 64 e-mails which were manually collected from DBWorld mailing list. They were classified as: ‘announces of conferences’ (1) and ‘everything else’ (0) (variable Conference)

1. Divide data into training and test sets (70/30) without scaling. Perform nearest shrunken centroid classification of training data in which the threshold is chosen by cross-validation. Provide a centroid plot and interpret it. How many features were selected by the method? List the names of the 10 most contributing features and comment whether it is reasonable that they have strong effect on the discrimination between the conference mails and other mails? Report the test error.

```
## Warning in RNGkind("Mersenne-Twister", "Inversion", "Rounding"): non-uniform
## 'Rounding' sampler used
```

```
## 1234567891011121314151617181920212223242526272829303132333435363738394041
```



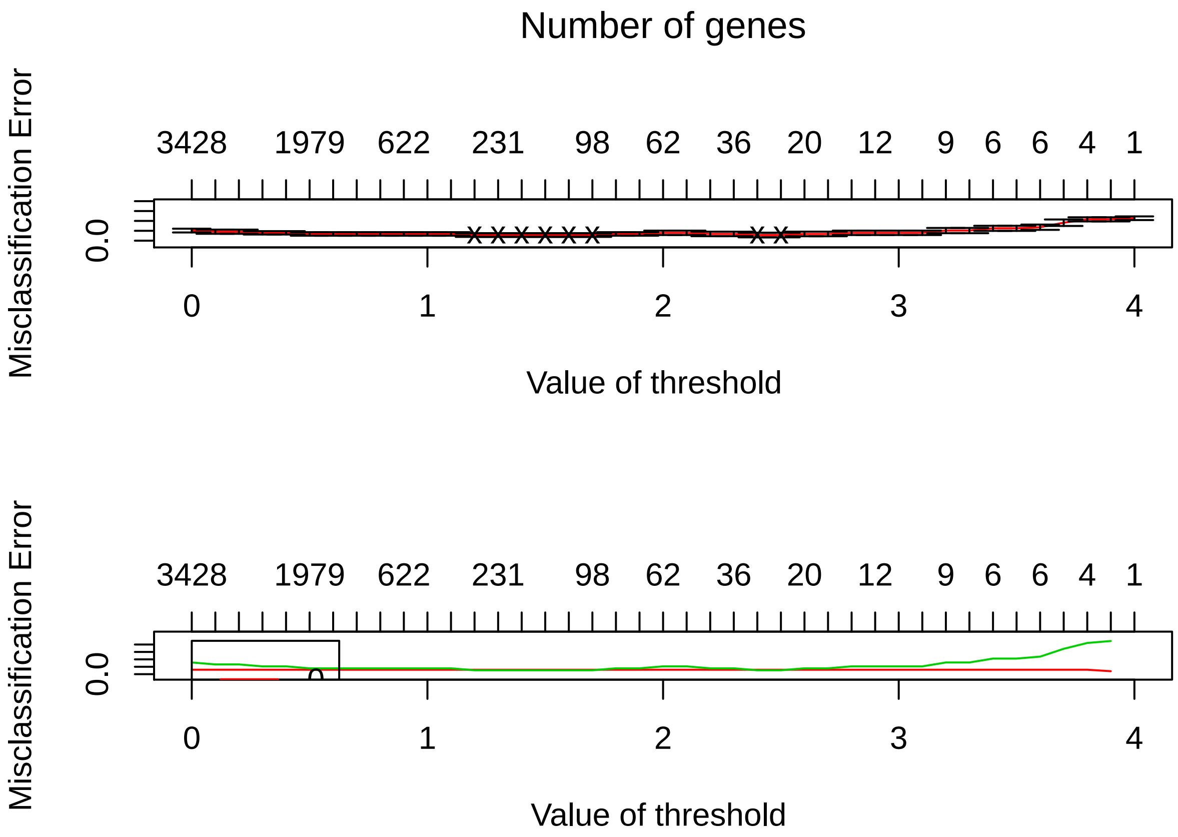
```r
cat(paste("The number of selected features are :",length(Features)),"\n")
```

```
## The number of selected features are : 693
```

```r
cat("The top 10  selected features are given below : \n",
    paste(colnames(data)[as.numeric(Features[,1])][1:10],"\n" ))
```

```
## The top 10  selected features are given below :
##  papers
##  important
##  submission
##  due
##  published
##  position
##  call
##  conference
##  dates
##  candidates
```

```r
#print(cvmodel)
pamr.plotcv(cvmodel)
```



```
train_error=pamr.confusion(cvmodel,threshold = min_threshold,extra=TRUE)
```

```
##    0  1 Class Error rate
## 0 22  3        0.1200000
## 1  2 17        0.1052632
## Overall error rate= 0.113
```

```
#Reporting of test error
x=t(as.matrix(test_set[,-4703]))
y=as.matrix(test_set$Conference)
mydata=list(x=x,y=as.factor(y),geneid=as.character(1:nrow(x)), genenames=rownames(x))
#cat("\tConfusion  Matix ")
```

```
# Confusion Matrix
plan=table( test_model,test_data=y)
plan
```

```
##           test_data
## test_model  0  1
##          0 10  2
##          1  0  8
```

```
#True Positive, True Negative, False Positive & False Negative calculation
TrueP=plan[2,2];TrueN=plan[1,1];FalseN=plan[2,1];FalseP=plan[1,2]
cat("\t Misclasification rate ",(FalseP + FalseN)/ sum(plan))
```

```
##   Misclasification rate  0.1
```

```
Misclassification_error_NSC <- (FalseP + FalseN)/ sum(plan)
```

The top 10 features selected from the model, seems to have high effect in discrimination of conference mails and other mails. Looking closely at the selected features, these should be big watch-words when we talk about conferences.

The misclassification rate of 10% further signifies the importance level of the selected features and the model overall.

2. 2. Compute the test error and the number of the contributing features for the following methods fitted to the training data:

a. Elastic net with the binomial response and alpha=0.5 in which penalty is selected by the cross-validation

```
##
## elastic_train_projection  0  1
##                         0 24  0
##                         1  1 19
```

```
##   Misclasification rate  0.02272727
```

```
##
## elastic_test_projection  0  1
##                       0 10  2
##                       1  0  8
```

```
##   Misclasification rate  0.1
```

```
##
##  Number of features selected by Elastic net : 35
```

2b. Support vector machine with “vanilladot” kernel

```
##  Setting default kernel parameters
```

```
## Warning in .local(x, ...): Variable(s) `' constant. Cannot scale data.
```

```
##
## ker_predict  0  1
##           0 10  1
##           1  0  9
```

```
##   Misclasification rate  0.05
```

```
##
##  Number of features selected by SVM : 43
```

Compare the results of these models with the results of the nearest shrunken centroids (make a comparative table).Which model would you prefer and why?

```
##                                       Misclassification(Test) Error Selected Features
## Nearest Shrunken Centroid             10%                           693
## Elastic Net                           10%                           35
## Support Vector Machine with Vanilladot 5%                            43
```

On comparing the test and train error of all the three above listed module, we find that the misclassification rate is minimum in Support Vector Machine (SVM) with vanilladot method. It is therefore, SVM with vanilladot should be preferred over nearest shrunken centroid and Elastic net models.

3. Implement Benjamini-Hochberg method for the original data, and use t.test() for computing p-values. Which features

correspond to the rejected hypothesis? Interpret the result.

```
##
## The total number of features are : 39
```

```
## [1] "\n The features are listed below :"
##         Features
## 1          apply
## 2        authors
## 3           call
## 4         camera
## 5      candidate
## 6     candidates
## 7         chairs
## 8      committee
## 9     conference
## 10         dates
## 11        degree
## 12           due
## 13     excellent
## 14          held
## 15     important
## 16 international
## 17       limited
## 18  notification
## 19           org
## 20      original
## 21         pages
## 22         paper
## 23        papers
## 24           phd
## 25      position
## 26          post
## 27     presented
## 28   proceedings
## 29      projects
## 30     published
## 31         ready
## 32        record
## 33        salary
## 34        skills
## 35        strong
## 36    submission
## 37          team
## 38        topics
## 39      workshop
```

The 39 features that rejected the hypothesis are listed above.

These features seems to be a good range of selection for the conference email filter.

Code Appendix

```r
knitr::opts_chunk$set(echo = TRUE)
library(readxl)
library(mgcv)
library(interp)
library(plotly)
library(akima)
library(pamr)
library(glmnet)
library(kernlab)
library(sgof)
library('scales')

#reading the data file in excel
data=read_excel("influenza.xlsx")
#data # Displaying data sample for understanding
#attach(data)
#head(data)
# Using plot function and plotting both mortality and influenza in same graph for comparision
par(mar = c(5, 5, 3, 5)) #size of plot
plot(data$Time,data$Mortality,type ="l", ylab = "Mortality ",main = "Time Series Plot",
     xlab = "Time",col = "dark green")
par(new = TRUE) # for combining two plots
plot(data$Influenza, type = "l", xaxt = "n", yaxt = "n",ylab = "", xlab = "", col = "red", lty = 2)
axis(side = 4)
mtext("Influenza ", side = 4, line = 3);legend("topleft", c("Mortality", "Influenza"),
                                               col = c("dark green", "red"), lty = c(1, 2))
# Indexing the plot at top left position

# Selecting linear function of year & spline function of week in gam model for mortality distribution
# the reference study material is lecture slides
res=gam(Mortality~Year+s(Week,k=length(unique(data$Week))),data=data)
print(res)
#reporting the underlying probabilistic model using summary function
summary(res)

#plot of predicted and oserved mortality against time
res_predict <- predict(res,data=data$Mortality,interval = "prediction")
# head(residuals(res)) # or head(data$Mortality-res_predict) to check on quality of prediction
plot(data$Time,res_predict, col = "red",type ="l",main = "Predicted mortality")
#to plot predicted mortality rate
plot(data$Time,data$Mortality,col = "dark green",type ="l",,main = "Observed/Original mortality ")
#to plot original/observed mortality rate
#Combining two plot to produce optimal result for easy comparision
par(mar = c(5, 5, 3, 5)) #size of plot
plot(data$Time,data$Mortality,type ="l", ylab = "observed mortality",
     main = "observed vs predicted mortality", xlab = "Time",col = "dark green")
par(new = TRUE) # for combining two plots
plot(res_predict, type = "l", xaxt = "n", yaxt = "n",ylab = "", xlab = "", col = "red", lty = 2)
axis(side = 4)
mtext("Predicted mortality ", side = 4, line = 3);
legend("topleft", c("Obs.Mortality", "Pred.Mortality"),col = c("dark green", "red"), lty = c(1, 2))
# Indexing the plot at top left position

plot(res)
#sp is penalty factor
res=gam(Mortality~Year+s(Week,k=length(unique(data$Week)),sp=.001),data=data)
res_predict <- predict(res,data=data$Mortality,interval = "prediction")
print(res)
```

```r
summary(res)
plot(res)
res$deviance


# penalty factor of the spline function in the GAM model from step 2 influences
# the estimated deviance of the model
# Let us assume low penalty factor Sp= 0.01, high penalty factor sp=100)
res_sp0.01<-gam(Mortality~Year+s(Week,k=length(unique(data$Week)),sp=0.01),data=data)
res_sp100<-gam(Mortality~Year+s(Week,k=length(unique(data$Week)),sp=100),data=data)
res_predict_sp0.01<-predict(res_sp0.01,data=data,interval = "prediction")
res_predict_sp100<-predict(res_sp100,data=data,interval = "prediction")

par(mar = c(5, 5, 3, 5)) #size of plot
plot(data$Time,data$Mortality,type ="l", ylab = "Mortality ",
     main = "Mortality Time Series Plot for Penalty Factors ", xlab = "Time",
     col = "dark green",ylim=range(c(data$Mortality,res_predict_sp100)))
par(new = TRUE) # for combining plots
plot(res_predict_sp0.01, type = "l", xaxt = "n", yaxt = "n",ylab = "", xlab = "",
     col = "red", lty = 2,ylim = c(1500,3000))
axis(side = 4)
par(new = TRUE) # for combining plots
plot(res_predict_sp100, type = "l", xaxt = "n", yaxt = "n",ylab = "", xlab = "",
     col = "blue", lty = 2,ylim = c(1500,3000))
axis(side = 4)
mtext("Penalty factor", side = 4, line = 3);
legend("topleft", c("Mortality", "penalty factor-0.01","penalty factor-100"),
       col = c("dark green", "red","blue"), lty = c(1, 2)) # Indexing the plot at top left position

#Creating null deviance, dof ,and res model to save from the while loop over sequence of PF
res_degree_of_freedom <- list()
deviance <-c()
degree_of_freedom <-c()
penalty_factor <- seq(0,10,0.1)
i=1
while(i <= length(penalty_factor)){
res_degree_of_freedom <- gam(Mortality~Year+s(Week, sp=penalty_factor[i]),data=data)
deviance[i] <- res_degree_of_freedom$deviance
degree_of_freedom[i] <- sum(res_degree_of_freedom$edf)
i=i+1
}
# building up data fram of all the variables to call in plot
data_table<- data.frame(Penalty_Factor=penalty_factor,
                        Deviance=deviance ,DOF= degree_of_freedom)
plot(x=data_table$Penalty_Factor,y=data_table$Deviance,
     xlab="Penalty Factor",ylab="Deviance",main ="Deviance Vs Penalty Factor",type="l",col="blue")

plot(x=data_table$Penalty_Factor,y=data_table$DOF, xlab="Penalty Factor",
     ylab="Degree of Freedom",main ="Degree of Freedom Vs Penalty Factor",type="l",col="red")
#residual of mortality
res_mortality=(data$Mortality-res_predict)
#Combining two plot to produce optimal result for easy comparision
par(mar = c(5, 5, 3, 5)) #size of plot
```

```
plot(data$Time,data$Influenza,type ="l", ylab = "Influenza values",
     main = "residuals vs influenza values", xlab = "Time",col = "dark green",
     ylim=range(c(data$Influenza,res_mortality)))
par(new = TRUE) # for combining two plots
plot(res_mortality, type = "l", xaxt = "n", yaxt = "n",ylab = "", xlab = "", col = "red", lty = 2)
axis(side = 4)
mtext("residuals ", side = 4, line = 3);legend("topleft", c("influenza", "residuals"),
                                               col = c("dark green", "red"), lty = c(1, 2))
# Indexing the plot at top left position

res1=gam(Mortality~s(Influenza,k=length(unique(data$Influenza)))+s(Week, k=length(unique(data$Week)))
         +s(Year,k=length(unique(data$Year))),data=data)
plot(res1)
#Fitted res 1
res2=fitted(res1)
#print(res1)
#plot of the original and fitted Mortality
par(mar = c(5, 5, 3, 5)) #size of plot
plot(data$Time,data$Mortality,type ="l", ylab = "original mortality",
     main = "original vs fitted mortality",
     xlab = "Time",col = "dark green",ylim=range(c(data$Mortality,res2)))
par(new = TRUE) # for combining two plots
plot(data$Time,res2, type = "l", xaxt = "n", yaxt = "n",ylab = "", xlab = "",
     col = "red", lty = 2,ylim=range(c(data$Mortality,res2)))
axis(side = 4)
mtext("fitted mortality ", side = 4, line = 3);
legend("topright", c("Original mortality", "fitted mortality"),
                                                      col = c("dark green", "red"),
       lty = c(1, 2)) # Indexing the plot at top left position

summary(res1)

data <- read.csv2("data.csv")
data$Conference=as.factor(data$Conference)
#length(data$emails) is 64
RNGversion('3.5.1')
set.seed(12345)
#setting training and test sets (70/30)
indexset=sample(1:length(data$Conference), floor(length((data$Conference))*0.7))
train_set=data[indexset,]
test_set=data[-indexset,]
#dim(train_set)
#dim(test_set)
# on understanding and using information from lecture slides, I proceed as below :
library(pamr)
rownames(data)=1:nrow(data)
#the column of Conference; x will contain train set without Conference column,
# while y train the conference column only
x=t(train_set[,-4703])
y=train_set$Conference
mydata=list(x=x,y=as.factor(y),geneid=as.character(1:nrow(x)), genenames=rownames(x))
#using pamr.train() to train model from thresholding 0 to 4 with a gap of 0.1 (lecture notes)
train_model=pamr.train(mydata,threshold=seq(0,4, 0.1))
```

```
cv_train<- pamr.cv(train_model,mydata) # not printing the result as it is not required

min_threshold <-cv_train$threshold[which.min(cv_train$error)]
#On running pamr.cv function,We get minimum error 5 at threshold=1.3 & 1.4

# Therefore, Centroid plot at threshold = min_threshold as a
# outcome for minimum error (5) is as below :
pamr.plotcen(train_model, mydata, threshold=min_threshold)

Features=pamr.listgenes(train_model,mydata,threshold=min_threshold)
# : Not priniting it anymore as per instruction received


cat(paste("The number of selected features are :",length(Features)),"\n")
cat("The top 10  selected features are given below : \n",
    paste(colnames(data)[as.numeric(Features[,1])][1:10],"\n" ))
cvmodel=pamr.cv(train_model,mydata)
#print(cvmodel)
pamr.plotcv(cvmodel)
train_error=pamr.confusion(cvmodel,threshold = min_threshold,extra=TRUE)

#Reporting of test error
x=t(as.matrix(test_set[,-4703]))
y=as.matrix(test_set$Conference)
mydata=list(x=x,y=as.factor(y),geneid=as.character(1:nrow(x)), genenames=rownames(x))
#cat("\tConfusion  Matix ")
test_model=pamr.predict(train_model,x,threshold=min_threshold)

# Confusion Matrix
plan=table( test_model,test_data=y)
plan
#True Positive, True Negative, False Positive & False Negative calculation
TrueP=plan[2,2];TrueN=plan[1,1];FalseN=plan[2,1];FalseP=plan[1,2]
cat("\t Misclasification rate ",(FalseP + FalseN)/ sum(plan))
Misclassification_error_NSC <- (FalseP + FalseN)/ sum(plan)


datax<-as.matrix(train_set[,(colnames(train_set)!=c("Conference"))])
datay<-as.matrix(train_set[,colnames(train_set)==c("Conference")])
#Implementing cross validation function below and obtain lambda output
elastic_cv<-cv.glmnet(x=datax,y=datay,alpha = 0.5,family="binomial")
# So we got Lambda min & Lambda 1se values from crossvalidation function above
# We can choose any lambda from above two to proceed further.
elastic<- glmnet(x=datax,y=datay,alpha =0.5,lambda=elastic_cv$lambda.min,family="binomial")

elastic_train_projection <- predict(elastic,newx = as.matrix(train_set[,-4703]), type="class")
#creating confusion martix below on train data
plan=table( elastic_train_projection,train_set$Conference);
plan
#True Positive, True Negative, False Positive & False Negative calculation
TrueP=plan[2,2];TrueN=plan[1,1];FalseN=plan[2,1];FalseP=plan[1,2]
cat("\t Misclasification rate ",(FalseP + FalseN)/ sum(plan))
Misclassification_error <- (FalseP + FalseN)/ sum(plan)
```

```
elastic_test_projection <- predict(elastic,newx = as.matrix(test_set[,-4703]), type="class")

plan=table(elastic_test_projection,test_set$Conference);
plan
#True Positive, True Negative, False Positive & False Negative calculation
TrueP=plan[2,2];TrueN=plan[1,1];FalseN=plan[2,1];FalseP=plan[1,2]
cat("\t Misclasification rate ",(FalseP + FalseN)/ sum(plan))
Misclassification_rate_elastic <- (FalseP + FalseN)/ sum(plan)
# Features Selection
features_elastic<-elastic$df
cat(paste("\n Number of features selected by Elastic net :",features_elastic))


# Kernel method on train data
ker_train<-ksvm(x=as.matrix(train_set[,-4703]),y=train_set$Conference,kernel="vanilladot" )


# Kernel model on test data for prediction
ker_predict<- predict(ker_train,newdata = as.matrix(test_set[,-4703]))

plan=table(ker_predict,test_set$Conference);plan
#True Positive, True Negative, False Positive & False Negative calculation
TrueP=plan[2,2];TrueN=plan[1,1];FalseN=plan[2,1];FalseP=plan[1,2]
cat("\t Misclasification rate ",(FalseP + FalseN)/ sum(plan))
Misclassification_rate_SVM<-c((FalseP + FalseN)/ sum(plan))
# Features Selection
features_svm<-length(ker_train@coef[[1]])
cat(paste("\n Number of features selected by SVM :",features_svm))


comparative_table <- matrix(c(percent
                              (Misclassification_error_NSC),
                              percent( Misclassification_rate_elastic) ,percent(Misclassification_rate_S
                              length(Features),features_elastic,features_svm),ncol=2)
colnames(comparative_table) <- c("Misclassification(Test) Error", "Selected Features")
rownames(comparative_table) <- c("Nearest Shrunken Centroid","Elastic Net",
                                 "Support Vector Machine with Vanilladot")
comparative_table <- as.table(comparative_table)
comparative_table

pvalue<- rep_len(0,length.out=4702)
#storing variable leaving out conference column , hence 4703-1 =4702
variable<-1:4702
data <- read.csv2("data.csv")
#using for loop on t.test()and storing the values back in pvalue
for(i in variable){
     t=t.test(data[,i]~data$Conference,alternative="two.sided")
     pvalue[i]=t$p.value
                  }
#adjusting pvalue using BH method
PBH=p.adjust(pvalue,method="BH")
#head(PBH)
```

```r
BH=BH(pvalue, alpha = 0.05)
#BH$FDR
#BH$Rejections
#index, which- index - name of feature
# which adjusted values is less than 0.05
k=which(PBH<0.05)

#filtering out features list  that belongs to k
feature_list=colnames(data)[k]
#listing features
feature_dataframe<-data.frame(Features= feature_list,
                              stringsAsFactors = FALSE)
cat("\nThe total number of features are :" ,length(feature_list))
print("\n The features are listed below :")
feature_dataframe
```