Lab2 - Block 2

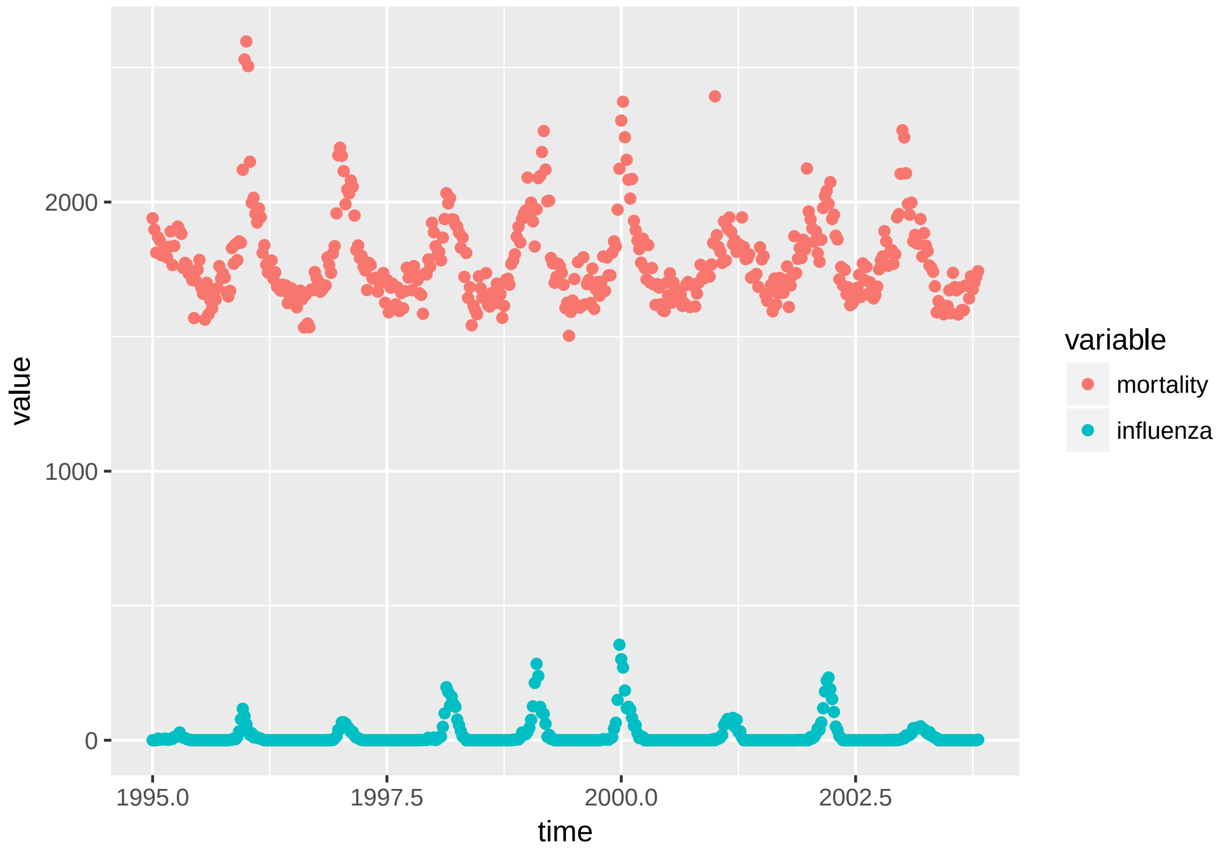
Group8

December 15, 2017

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

Part 1

Visually inspecting the relationship between mortality and influenza



From plot it can be observed that influenza cases have the peaks at the same points where the morltality plot has peaks, it shows that when there is increase in rate of influenza cases , mortality have increased drastically

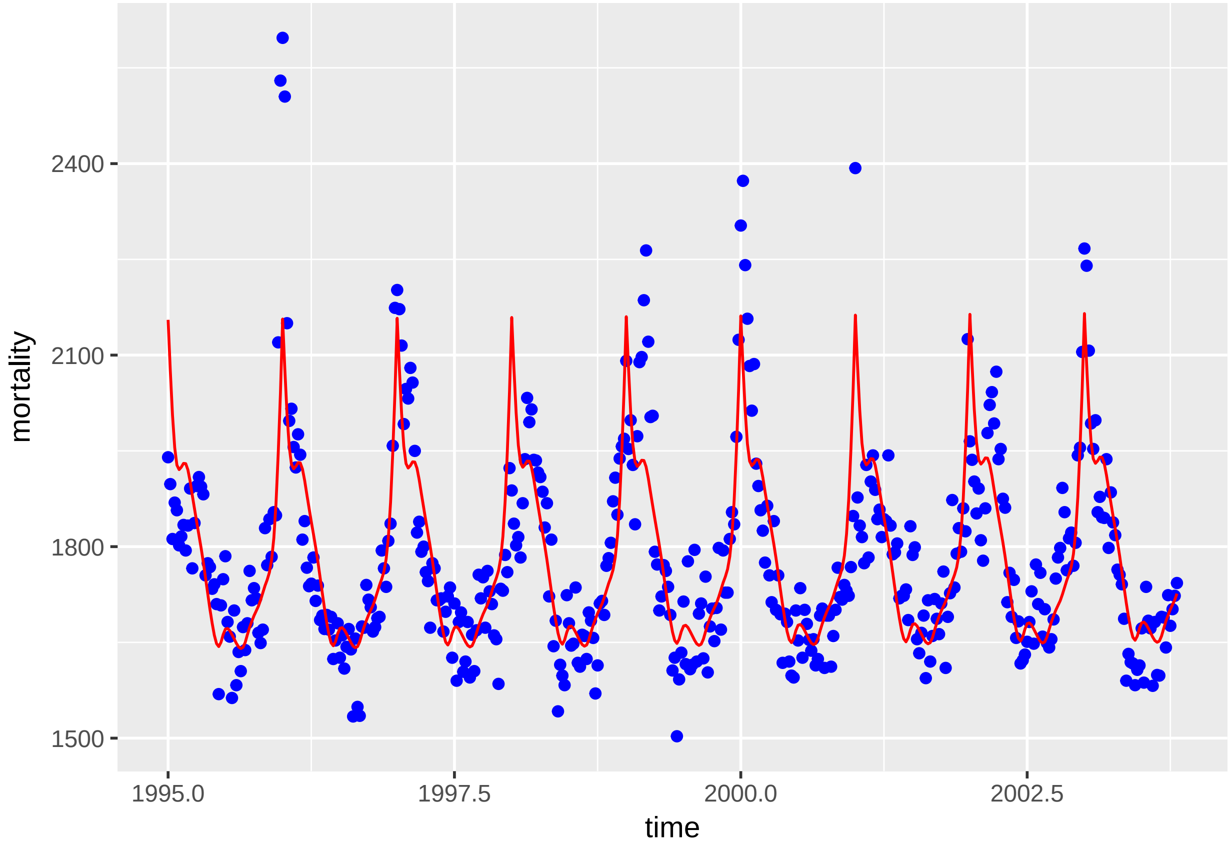
Part 2

Fitting generalized additive model keeping Year as linear function and week as spline function of mortality

Probilistic model: $y = w_o + w_1 x_1 + w_2 x_1^2 + e$

Probilistic model: $y = -680.589 + 1.233 * x_1 + s(Week) + \epsilon \sim N(0, \sigma^2)$

Part 3



Quality of fit?

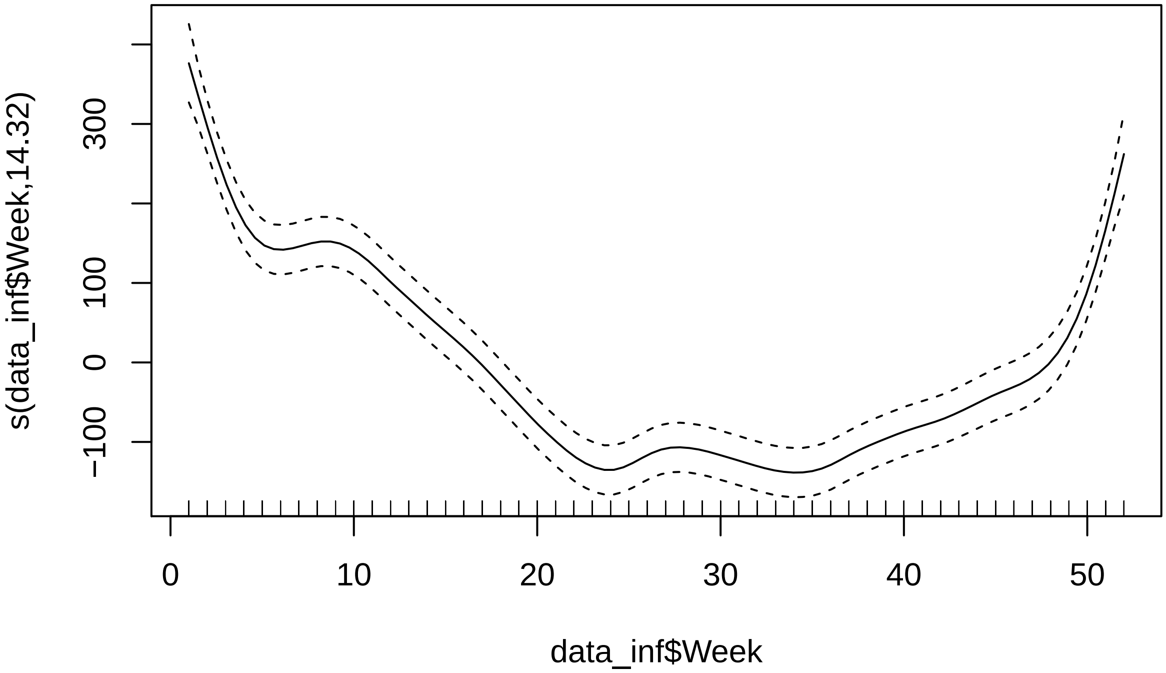
In plot, original values are indicated by blue points while predicted valuescan be observed as red line. Since there are many points that lie outside the prediction line especially at peaks where prediction line doesn't completely follows the original points, therefore it can be said that this is not the best fitting.

Identify the trend?

There is a decreasing trend in mortality from one year to another that mortality rate seems to increase at the beginning of the year and drops at the end of the year.

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## data_inf$Mortality ~ data_inf$Year + s(data_inf$Week, k = 52)
##
## Parametric coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -680.598   3367.760  -0.202    0.840
## data_inf$Year      1.233      1.685   0.732    0.465
##
## Approximate significance of smooth terms:
##                    edf Ref.df     F p-value
## s(data_inf$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
```

since “year” and “week” both gives p-values less tan the significance level 0.05, therefore both are significant



The mortality rate seems to decrease from beginning and has the lowest values between weeks 20 to 30 and then mortality again rises from week 35 onwards

Part 4

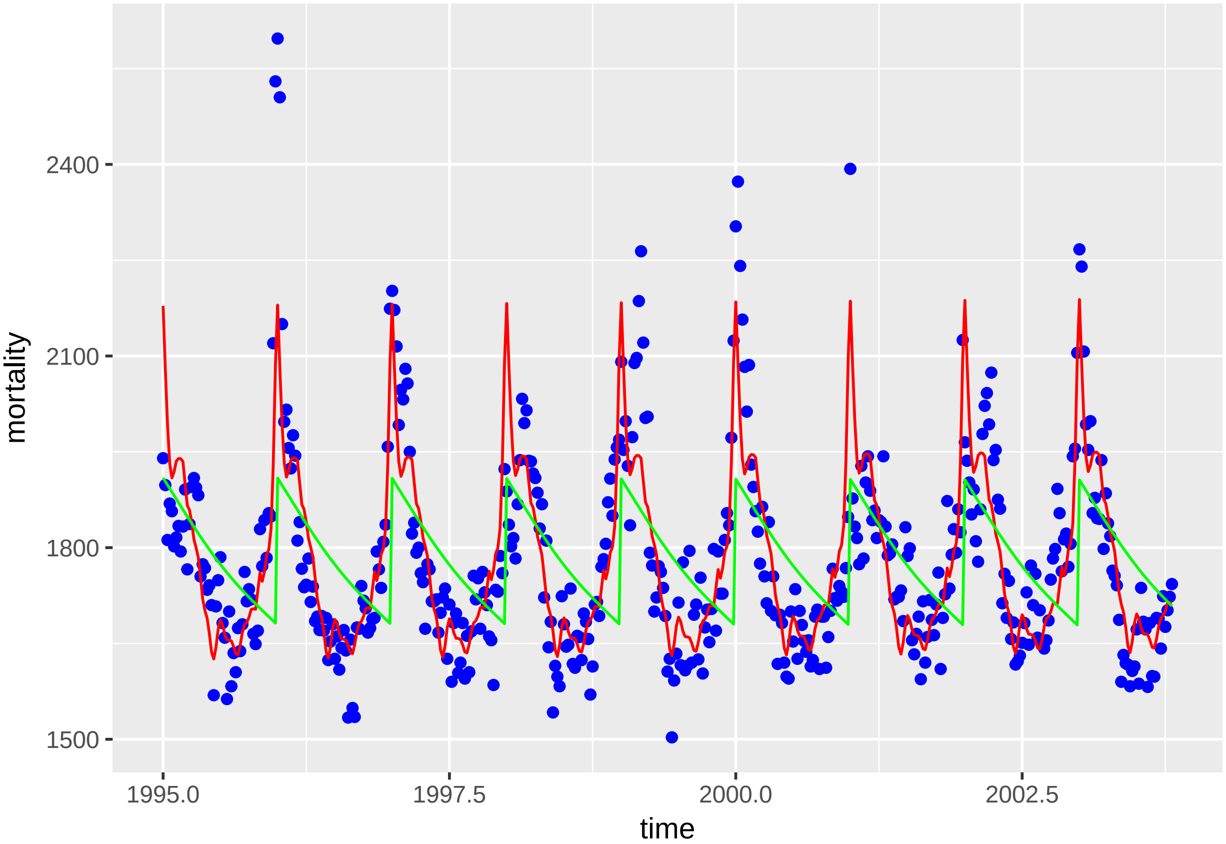
The deviance at the selected penalty factor is high that is 68.8% since the penalty factor selected by cross validation is too low.

Fitting at lower penalty factor:

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## data_inf$Mortality ~ Year + s(data_inf$Week, k = 52, sp = 1e-09)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -710.838   3383.965  -0.210    0.834
## Year           1.248      1.693   0.737    0.461
##
## Approximate significance of smooth terms:
##                   edf Ref.df     F p-value
## s(data_inf$Week)  27     27 36.05  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Rank: 29/53
## R-sq.(adj) =  0.674   Deviance explained = 69.4%
## GCV = 9049.7  Scale est. = 8477.9    n = 459
```

Fitting at high penalty factor:

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## data_inf$Mortality ~ Year + s(data_inf$Week, k = 52, sp = 10)
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2614.4022  5234.1496   0.499    0.618
## Year          -0.4155     2.6185  -0.159    0.874
##
## Approximate significance of smooth terms:
##                     edf Ref.df     F p-value
## s(data_inf$Week) 1.069  1.136 106.1  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.217   Deviance explained = 22.1%
## GCV =  20478  Scale est. = 20341     n = 459
```



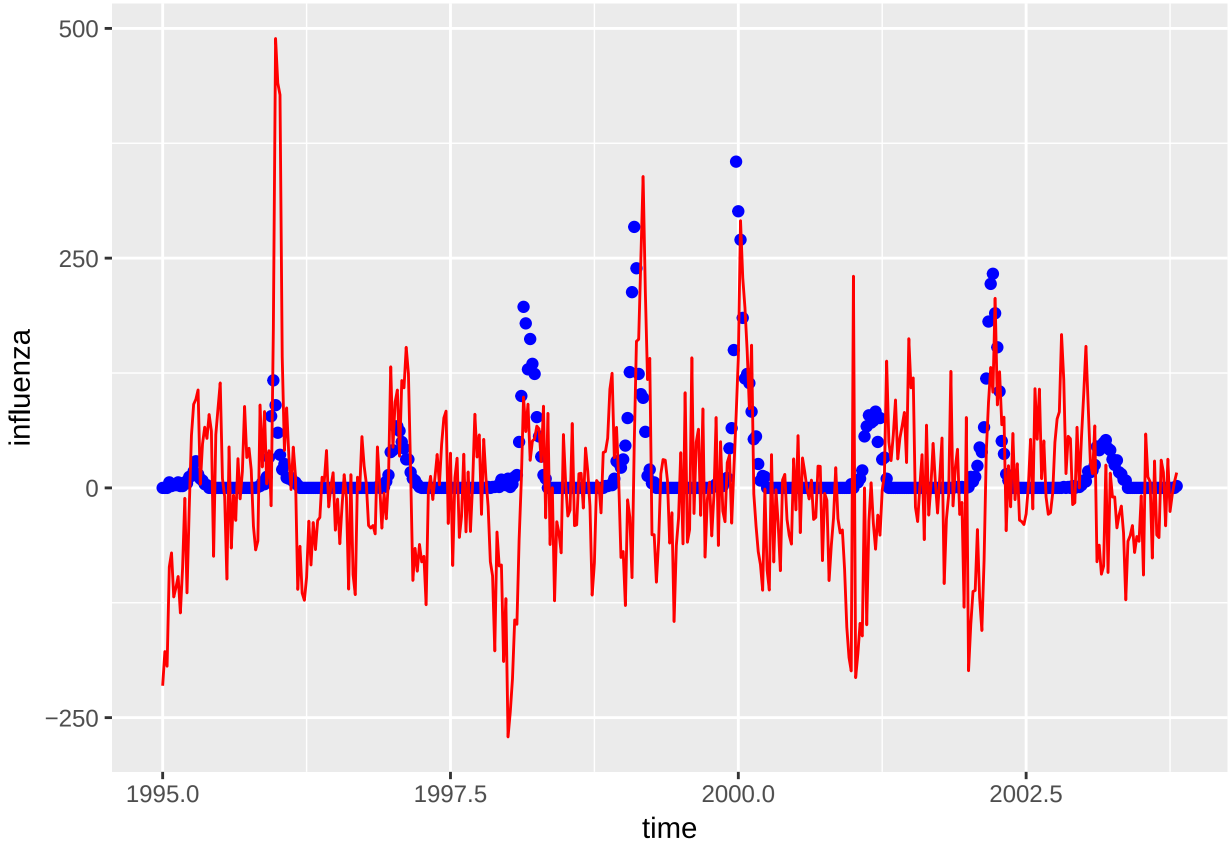
The very high annd very low values of penalty factor leads to overfitting of model, as it is evident from the plot in which green line represents the predicted values when penalty factor is too high while red line represents the mortality when penalty factor is too low.

Since the relationship $df_\lambda = \sum_{k=1}^{N} \frac{1}{1+\lambda d_k}$ it is obvious that λ is inversely proportional to df_λ

From plot, it can be observed that deviance and degrees of freedom decreases with increase in penalty factor which satisfies the above equation

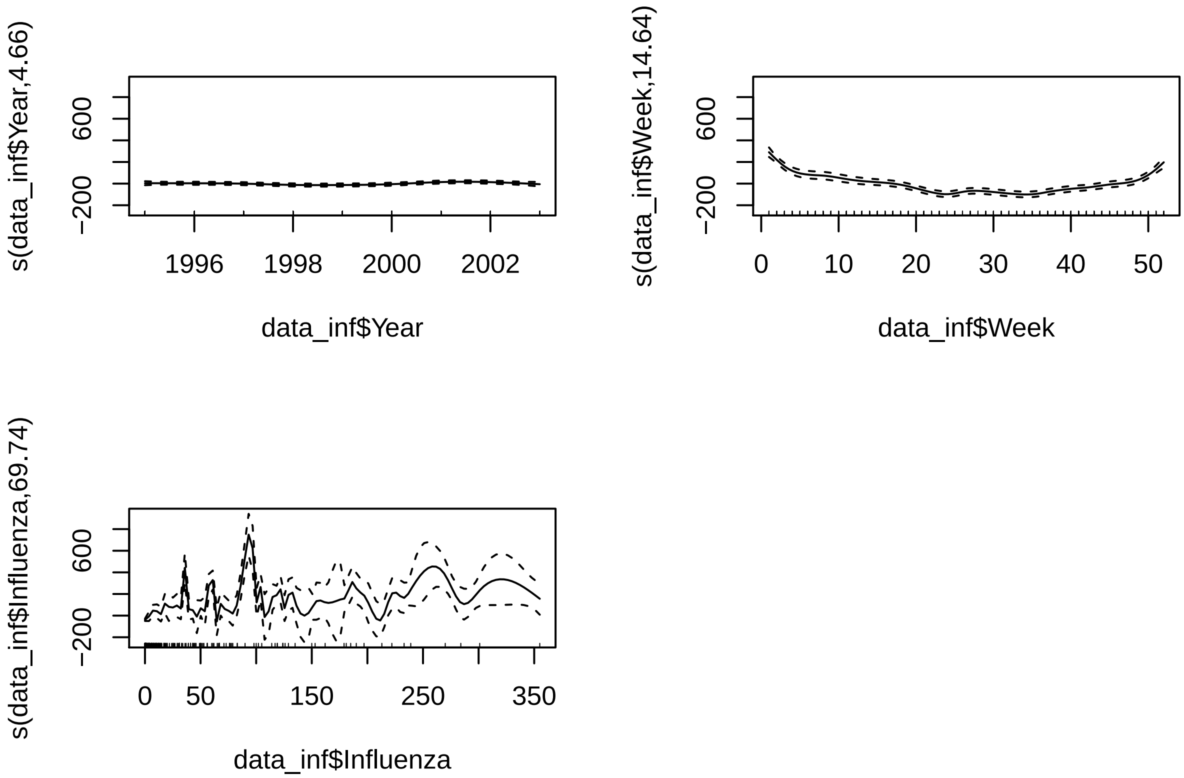
Part 5

```
df_plot <- data.frame(time= data_inf$Time,
                      influenza = data_inf$Influenza,
                      residual = as.vector(fit_week$residuals))

p2 <- ggplot(df_plot, aes(x= time, y = influenza)) +
  geom_point(colour= "blue") +
  geom_line(aes(time,residual),colour = "red")

p2
```



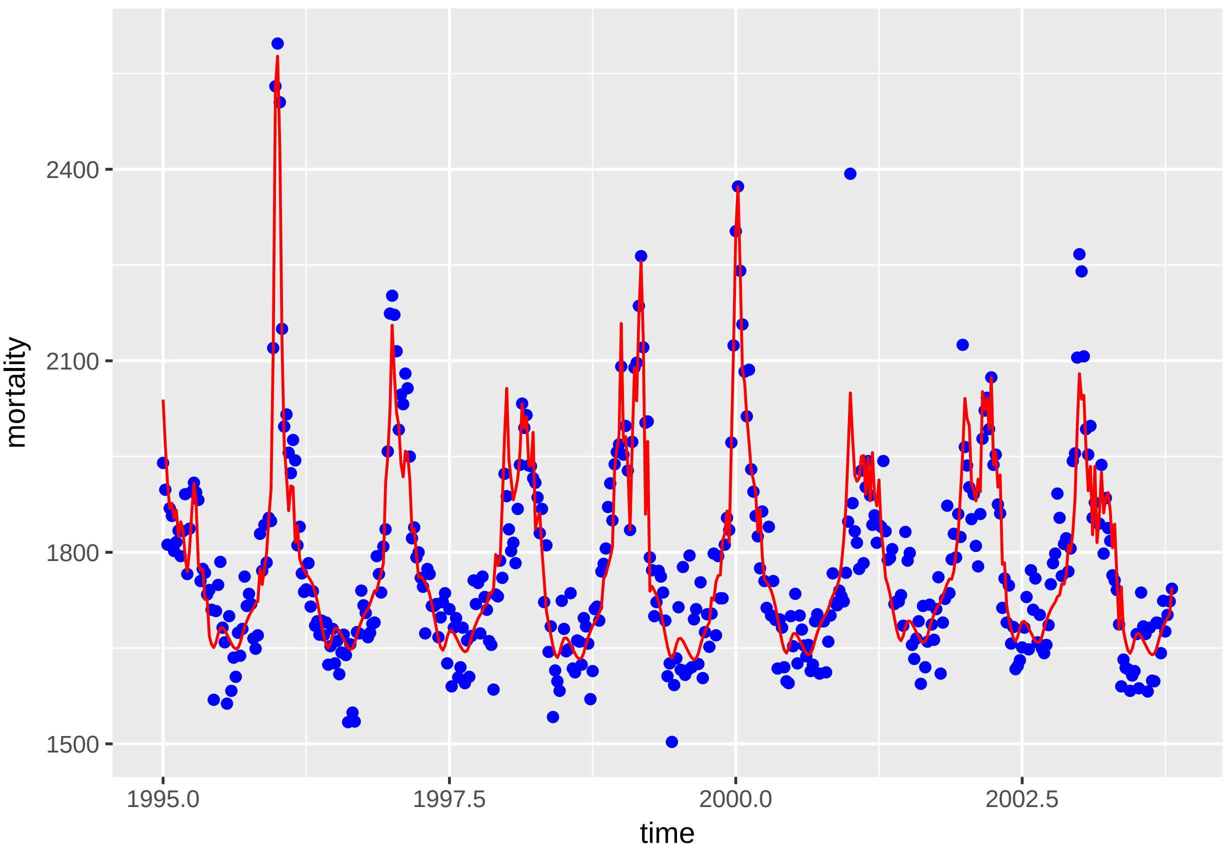
Yes, it is evident from the plot that the temporal pattern in residuals seems to be correlated to the outbreak of influenza since the peaks in influenza appears relative to the peaks in residuals obtained from Generalized Additive Model applied in step 2.

part 6



It can be illustrated from the plots of spline components that mortality does not depend much on year and have little change with weeks, but the mortality shows a significant relationship with influenza that is with increasing cases of influenza, mortality increases.

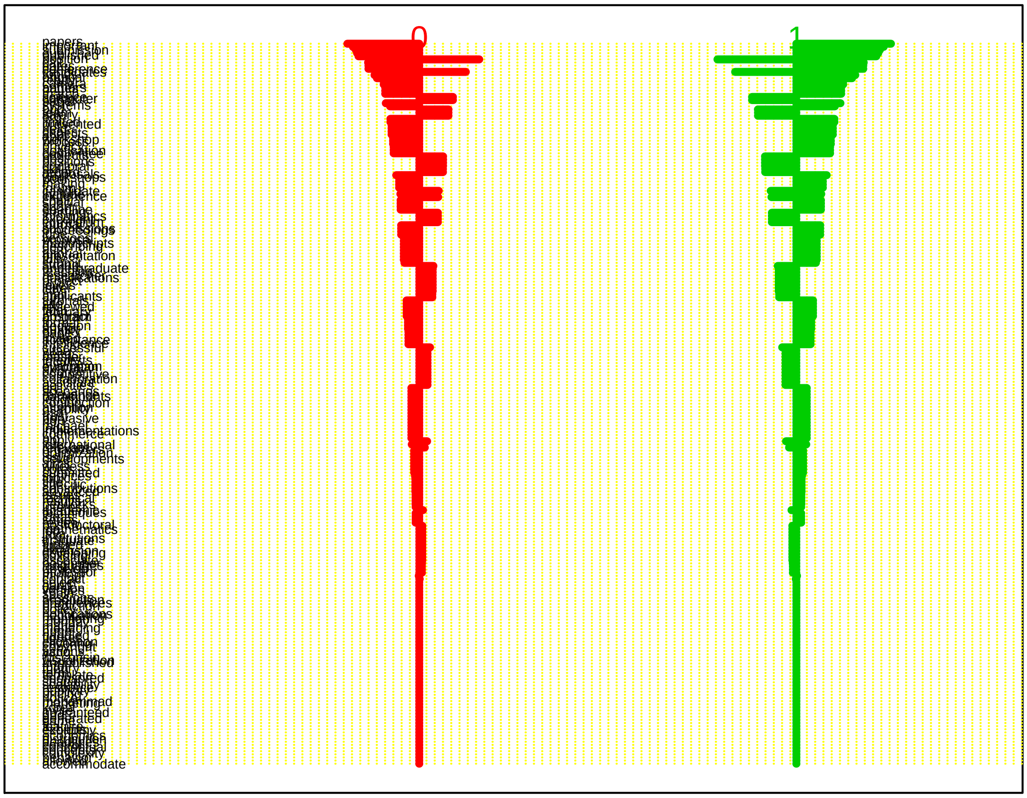
To support the above phenomenon, plot of predicted values based on model which includes influenza as spline function and true values is presented below:



Predicted values is represented by the line while true values is presented with blue points.The plot of original and predicted values implies that this model is better than the previous models as it gives the predicted values closest to the original values that is predicted values captures most of the points in original data. This also indicates that including influenza in modelling has a significant impact on fitting.

Assignment 2. High-dimensional Methods

PART 1



The features that have the non-zero degrees of freedom contributes to classification and are shown in plot with a horizontal bars on 0,1 line. In figure, 1 represents the features that corresponds to the announcement of conference while 0 represents everything else.

The features that are in top contributes the most for classification while those at the bottom are nearly shrunk to zero.

Since there are many features that contributes to classification, therefore the features are not clearly visible and distinguished in the plot.

How many features were selected by the method?

231 features have been selected in total.

Which features contributes the most for classification?

```
## papers
## important
## submission
## due
## published
## position
## call
## conference
## dates
## candidates
```

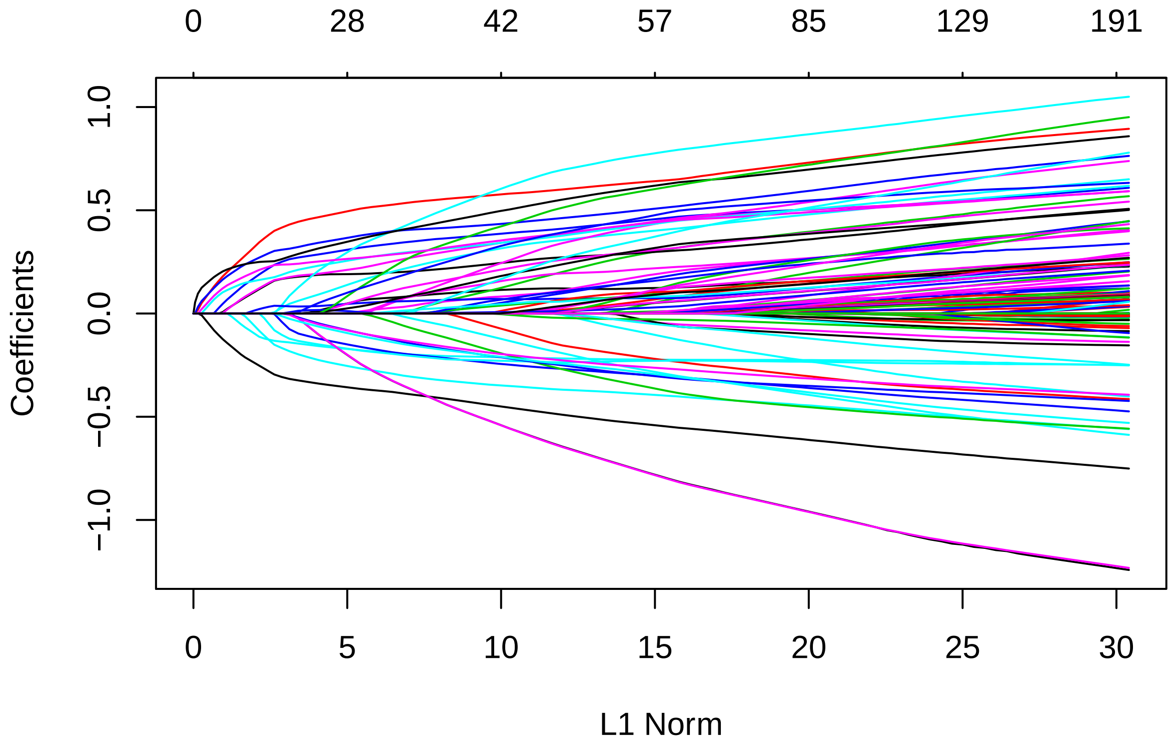
The above features are the most contributing based on the score which greatly discriminates conference announcements from other emails since the score difference between the two classes is highest for these features

Test Error

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

Part 2

Elastic Net



Test Error

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

`100 features` have been selected by the elastic net model.

The misclassification for elastic net is calculated to be `0.1`

Support Vector Machine

Test Error

```
## [1] 0.05
```

`43 features` have been selected

The misclassification rate of svm model is 0.05

comparision of misclassification rate of centroid, elastic net and svm

Table 1: Misclassification Table

centroid	elastic	svm
0.1	0.1	0.05

Part 3

Benjamini Hochberg Method

```
##   [1] "research"       "http"           "www"            "information"    "applications"   "systems"
##  [30] "including"      "papers"         "related"        "researchers"    "based"          "design"
##  [59] "committee"      "details"        "email"          "english"        "institute"      "journal"
##  [88] "communication"  "considered"     "december"       "due"            "html"           "january"
## [117] "special"        "user"           "apply"          "chairs"         "context"        "date"
## [146] "number"         "pdf"            "projects"       "required"       "search"         "semantic
## [175] "content"        "cross"          "databases"      "february"       "forum"          "free"
## [204] "active"         "algorithms"     "art"            "background"     "contributions"  "database
## [233] "community"      "conferences"    "doctoral"       "electronic"     "environment"    "focus"
## [262] "communications" "complex"        "curriculum"     "efficient"      "electronically" "evaluati
## [291] "reviewed"       "seeking"        "sensor"         "site"           "springer"       "statemen
```

The features that corresponds to the rejected hypothesis are the ones which are most relevant to "announcement of conference", that is which contributes the most for the classification of announcement of conference

APPENDIX

```r
# Assignment 1

library(readxl)
library(ggplot2)
#library(grid)
#library(gridExtra)
library(reshape2)
data <- read_excel("influenza.xlsx")
data_inf <- as.data.frame(data)

df_plot <- data.frame(time= data_inf$Time,
                      mortality = data_inf$Mortality,
                      influenza = data_inf$Influenza)
```

```
df <- melt(df_plot,measure.vars = c("mortality","influenza"))
ggplot(df,aes(x= time, y = value,colour = variable)) +
  geom_point()


## part 2
library(mgcv)

w <- unique(data_inf$Week)
fit_week <- gam(data_inf$Mortality ~ data_inf$Year + s(data_inf$Week,k=52),
                data = data_inf,
                family = "gaussian",
                method = "GCV.Cp")

pred <- predict(fit_week)

# Probilistic model: y = wo + w1x1 + w2x1^2 + e
#(where wo=intercept,  w1= est value of 1st var,
#w2= est value of 2nd var)
# Probilistic model:
#$y = -680.589 + 1.233*x1 + s(Week) + {\epsilon}~N(0,{\sigma}^2)


## part3

df_plot <- data.frame(time= data_inf$Time,
                      mortality = data_inf$Mortality,
                      pred = as.vector(pred))

p1 <- ggplot(df_plot, aes(x= time, y = mortality)) +
  geom_point(colour= "blue") +
  geom_line(aes(time,pred),colour = "red")+
  coord_cartesian(xlim = c(1995,2003))
p1

plot(fit_week)

summary(fit_week) # to check significant values using p values

# part 4

fit1 <- gam(
  data_inf$Mortality ~ Year + s(data_inf$Week,
  k=52,sp=as.numeric(t(fit_week$sp))),
  data = data_inf,
  family = "gaussian")

summary(fit1)
pred1 <- predict(fit1)

fit2 <- gam(
          data_inf$Mortality ~ Year + s(data_inf$Week,k=52,sp=0.000000001),
          data = data_inf,family = "gaussian")

summary(fit2)
pred2 <- predict(fit2)


fit3 <- gam(data_inf$Mortality ~ Year + s(data_inf$Week,k=52,sp=1.5),
            data = data_inf,family = "gaussian")
summary(fit3)
pred3 <- predict(fit3)

df_plot <- data.frame(time= data_inf$Time,
                      mortality = data_inf$Mortality,
                      pred1 = as.vector(pred2),
                      pred2 = as.vector(pred3))

p1 <- ggplot(df_plot, aes(x= time, y = mortality)) +
  geom_point(colour= "blue") +
  geom_line(aes(time,pred1),colour = "red")+
  geom_line(aes(time,pred2),colour = "green")

p1

# part 5

df_plot <- data.frame(time= data_inf$Time,
                      influenza = data_inf$Influenza,
                      residual = as.vector(fit_week$residuals))
#
p2 <- ggplot(df_plot, aes(x= time, y = influenza)) +
  geom_point(colour= "blue") +
  geom_line(aes(time,residual),colour = "red")

p2


## part 6

w <- unique(data_inf$Week)
y <- unique(data_inf$Year)
i <- unique(data_inf$Influenza)

fit_f <- gam(data_inf$Mortality ~ s(data_inf$Year,k=length(y)) +
               s(data_inf$Week,k=length(w))
             + s(data_inf$Influenza,k=length(i)), data = data_inf,
             family = "gaussian", method = "GCV.Cp")

pred_f <- predict(fit_f)

par(mfrow=c(2,2))
plot(fit_f)

par(mfrow=c(1,1))
```

```r
df_plot <- data.frame(time= data_inf$Time,
                      mortality = data_inf$Mortality,
                      predicted = as.vector(pred_f))

p3 <- ggplot(df_plot, aes(x= time, y = mortality)) +
  geom_point(colour= "blue") +
  geom_line(aes(time,predicted),colour = "red")

p3



# Assignment 2

data <- read.csv2("data.csv", header = TRUE, sep = ";", quote = "\"",
                  dec = ",", fill = TRUE, check.names = FALSE)
data1 <- as.data.frame(data)

data_email <- as.data.frame(data)

data_email$Conference <- as.factor(data$Conference)

n=dim(data_email)[1]
set.seed(12345)
# 70% Training Data
id=sample(1:n, floor(n*0.7))
train=data_email[id,]
# 30% validation & testing Data
test = data_email[-id,]


library(pamr)

rownames(train) <- 1:nrow(train)

which(colnames(train)=="Conference")

x <- t(train[,-4703])
y <- train[[4703]]

test_x <- t(test[,-4703])

mydata <- list(x=x, y= as.factor(y),
               geneid=as.character(1:nrow(x)),genenames = rownames(x))

model_train <- pamr.train(mydata)

cv_model <- pamr.cv(model_train, data = mydata)

pamr.plotcv(cv_model) # legend is not shown in plot and only 2 lines are drawn why?
print(cv_model)

model_fit <- pamr.train(mydata,
                         threshold = cv_model$threshold[which.min(cv_model$error)])

par(mfrow=c(1,1),mar=c(2,2,2,2))
pamr.plotcen(model_train, mydata, threshold = cv_model$threshold[which.min(cv_model$error)])

features = pamr.listgenes(model_train,
                          mydata, threshold = 1.306,
                          genenames=TRUE)
cat( paste( colnames(train)[as.numeric(features[1:10,1])], collapse='\n' ) )

ypredict <- pamr.predict(model_train, newx = test_x,
                         type = "class", threshold = 1.306)

conf_mat <- table(ypredict, test$Conference)
misclas_centroid <- 1 - (sum(diag(conf_mat))/sum(conf_mat))

#misclas <- pamr.confusion(cv_model, threshold = 1.306)

df_misclas <- data.frame(centroid = misclas_centroid, elastic = misclassification_elastic,
                         svm = misclas_svm)


## Part 2

library(glmnet)
set.seed(12345)
response <- train$Conference
predictors <- as.matrix(train[,-4703])

elastic_model <- glmnet(x=predictors,y=response,family = "binomial",alpha = 0.5)


cv.fit <- cv.glmnet(x=predictors,y=response,family="binomial",alpha = 0.5)
cv.fit$lambda.min

par(mar=c(2,2,2,2))
plot(cv.fit)
plot(elastic_model)

predictor_test <- as.matrix(test[,-4703])

ypredict <- predict(object = elastic_model,newx = predictor_test, s = cv.fit$lambda.min,
                    type = "class", exact = TRUE)

confusion_mat <- table(ypredict,test$Conference)

misclassification <- 1 - (sum(diag(confusion_mat))/sum(confusion_mat))


library(kernlab)

x <- as.matrix(train[,-4703])
y <- train[,4703]
```

```
svm_fit <- ksvm(data = train,Conference ~ . ,kernel="vanilladot",
                scaled = FALSE)

ypred <- predict(svm_fit, newdata = test, type="response")

confusion_mat <- table(ypred,test$Conference)

misclas_svm <- 1 - sum (diag(confusion_mat))/sum(confusion_mat)


## Part 3

pvals <- c()

for (i in 1:length(data1)) {
  ttest <- t.test(data1[i], data = data1, alternative = "two.sided")
  pvals[i] <- ttest$p.value
}

pvalues_df <- data.frame(p_value=pvals,feature=1:length(data1))
pvalues_df <- pvalues_df[order(pvalues_df$p_value),]


a <- 0.05
L <- c()
it <- 1
# Let alpha = 0.05  ## ask oleg
for (j in 1:nrow(pvalues_df)) {
  if(pvalues_df$p_value[j] < a * (j / nrow(pvalues_df)) )
  {
    L[it] <- j
    it <- it +1
  }
}
max(L)

LL = pvalues_df$p_value[max(L)]
LL
newPvalues <- c()
pvalue_feature <- c()
pvalue_status <- c()
j<- 1
for (j in 1:nrow(pvalues_df))
{

  pvalue_status[j] <- TRUE
  newPvalues[j] <- pvalues_df$p_value[j]
  pvalue_feature[j] <- pvalues_df$feature[j]
  if(pvalues_df$p_value[j] <= LL)
  {
    pvalue_status[j] <- FALSE
  }
```

```r
}
result <- data.frame(p_value=newPvalues, feature=pvalue_feature,status=pvalue_status)
result

rejected_features <- c()
k <- 1
for (j in 1:ncol(data1)) {

  if(result$status[j] == FALSE)
  {
    rejected_features[k] <- colnames(data1[result$feature[j]])
    k<- k + 1
  }
}
rejected_features
```