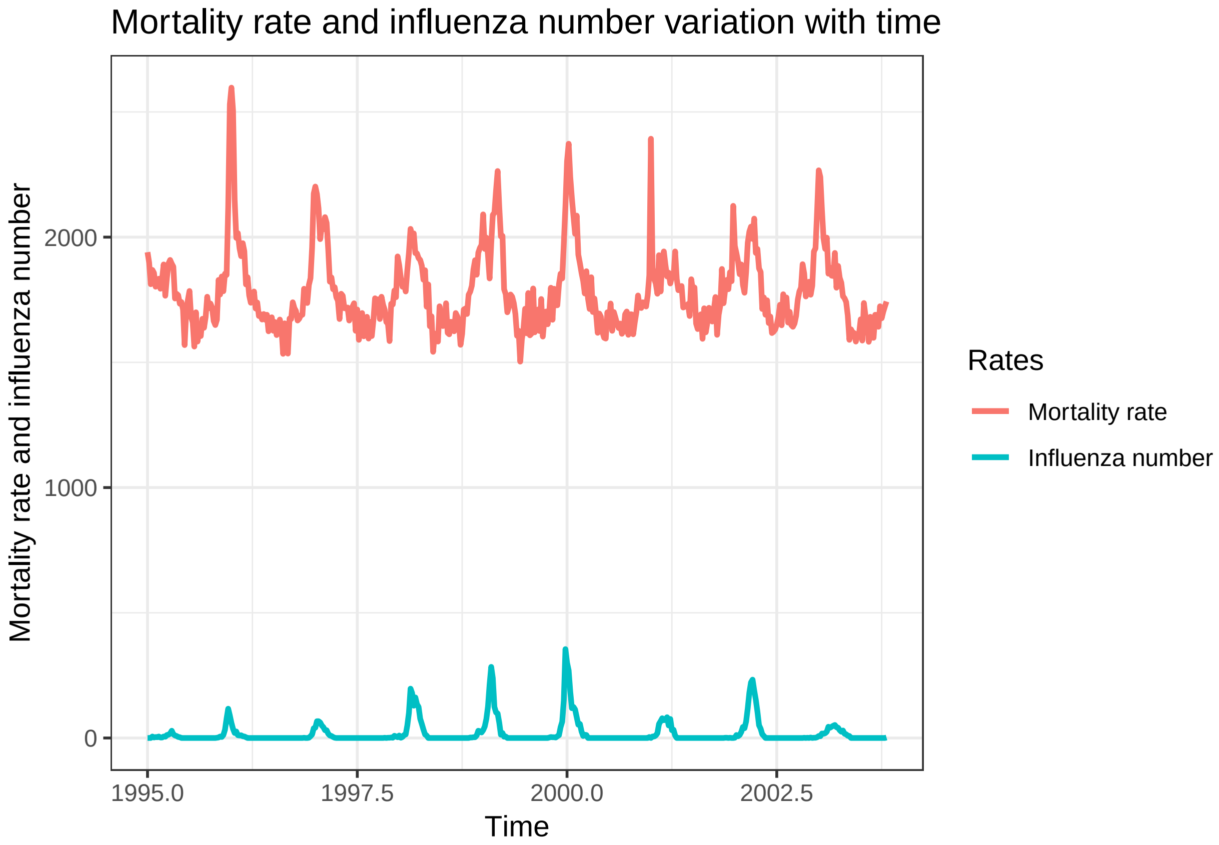
Block 2 - Lab 2

Roshni Sundaramurthy

16 December 2018

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

1.1 Time series plots



Analysis:

The plot depicts that the number of laboratory-confirmed cases of influenza in Sweden is high during 52nd week of year 1999 with mortality rate of 2124. The temperature deficit seems to be 0.25150. The mortality rate is high during 1st week of year 1996 with 60 influenza cases. The temperature deficit seems to be 5.33818.

They both exhibit the same pattern, such that, when the number of influenza cases increases, the mortality rate also increases for specific period of time. But still they are not much interrelated since they both have maximum range in different year.

1.2 Fitting a GAM model

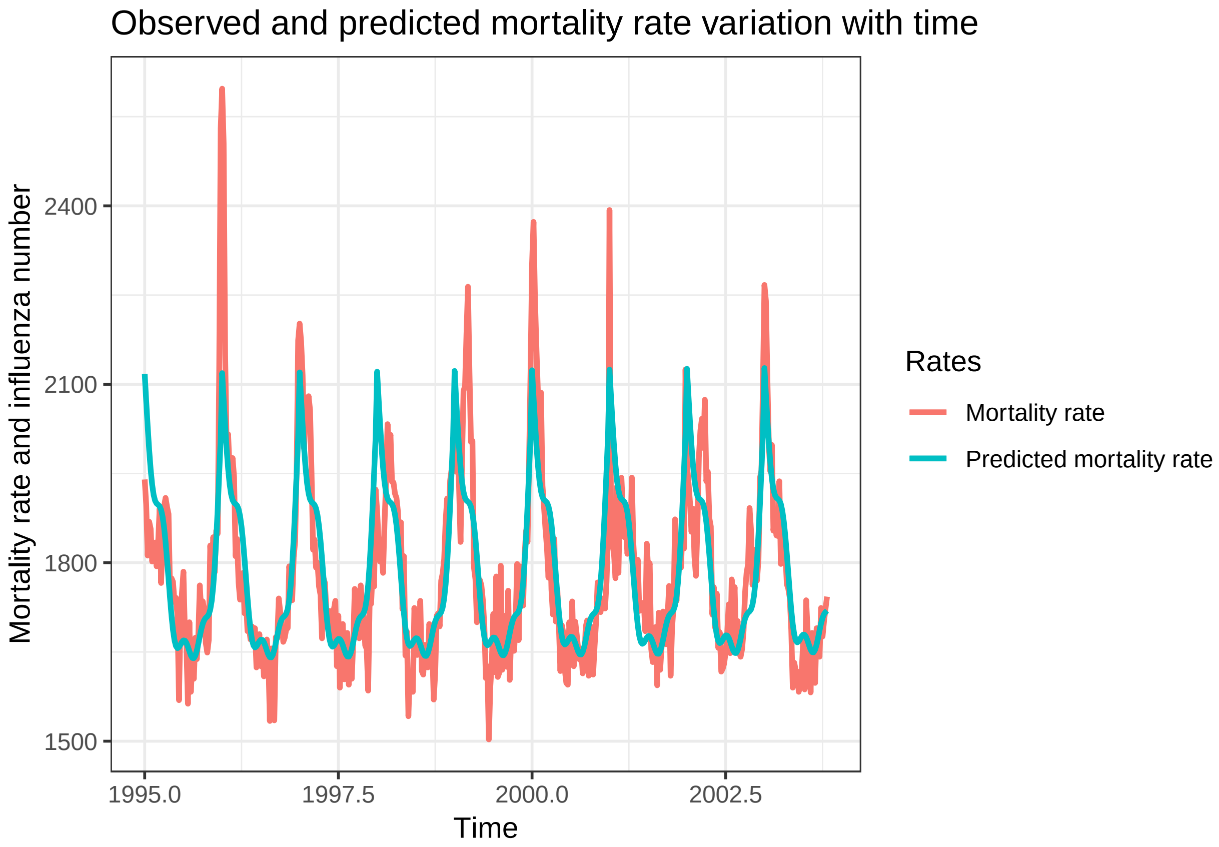
Analysis:

The underlying probablistic model where mortality is normally distributed :

$$g(E(y_i)) = (-652.058) + (1.2185 * Year) + s(Week) + \epsilon$$

where Y is mortality and ϵ is $N(0, \sigma^2)$

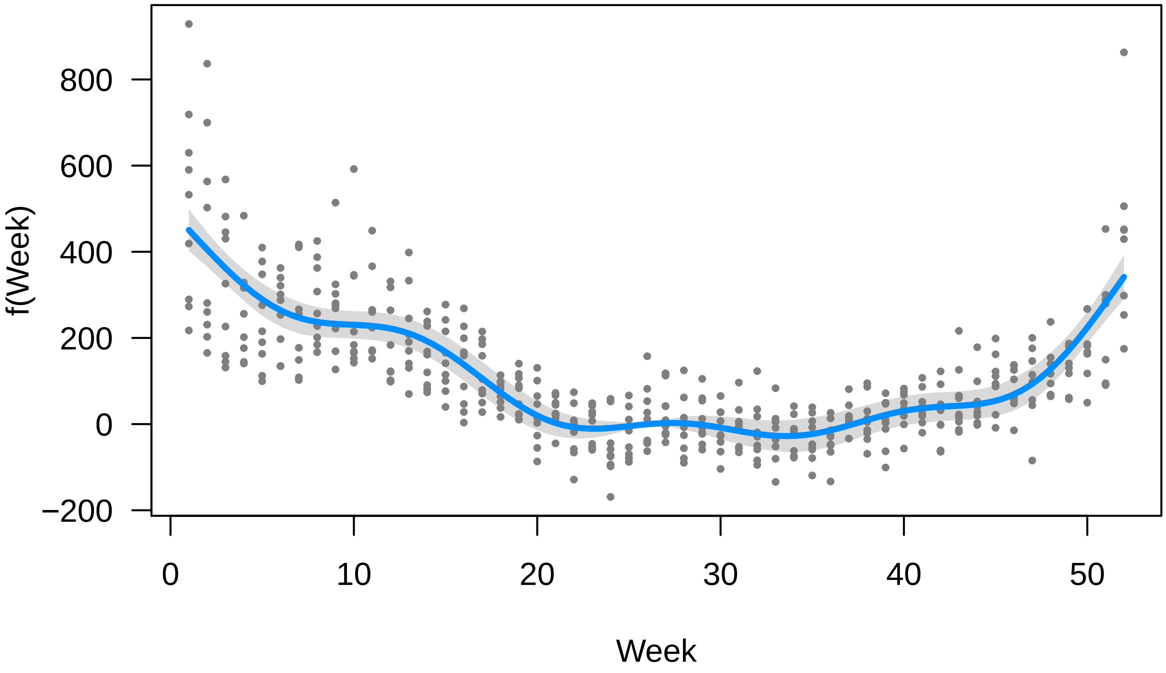
1.3 Plot predicted and observed mortality



```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Mortality ~ Year + s(Week)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -652.058   3448.379  -0.189     0.85
## Year           1.219      1.725   0.706     0.48
##
## Approximate significance of smooth terms:
##           edf Ref.df     F p-value
## s(Week) 8.587  8.951 100.6  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## R-sq.(adj) =  0.661   Deviance explained = 66.8%
## GCV = 9014.6  Scale est. = 8806.7    n = 459
```

```
## [1] "Spline component (Week) plot"
```



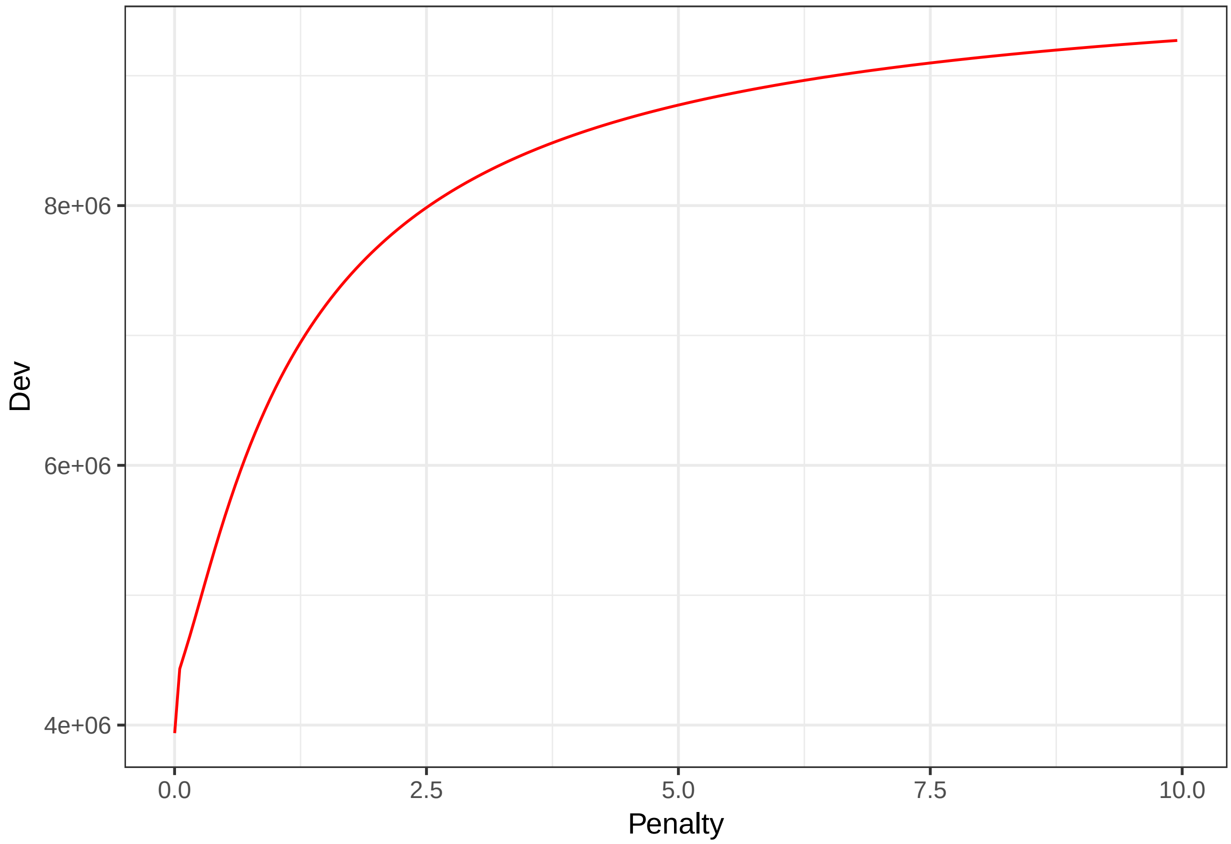
Analysis:

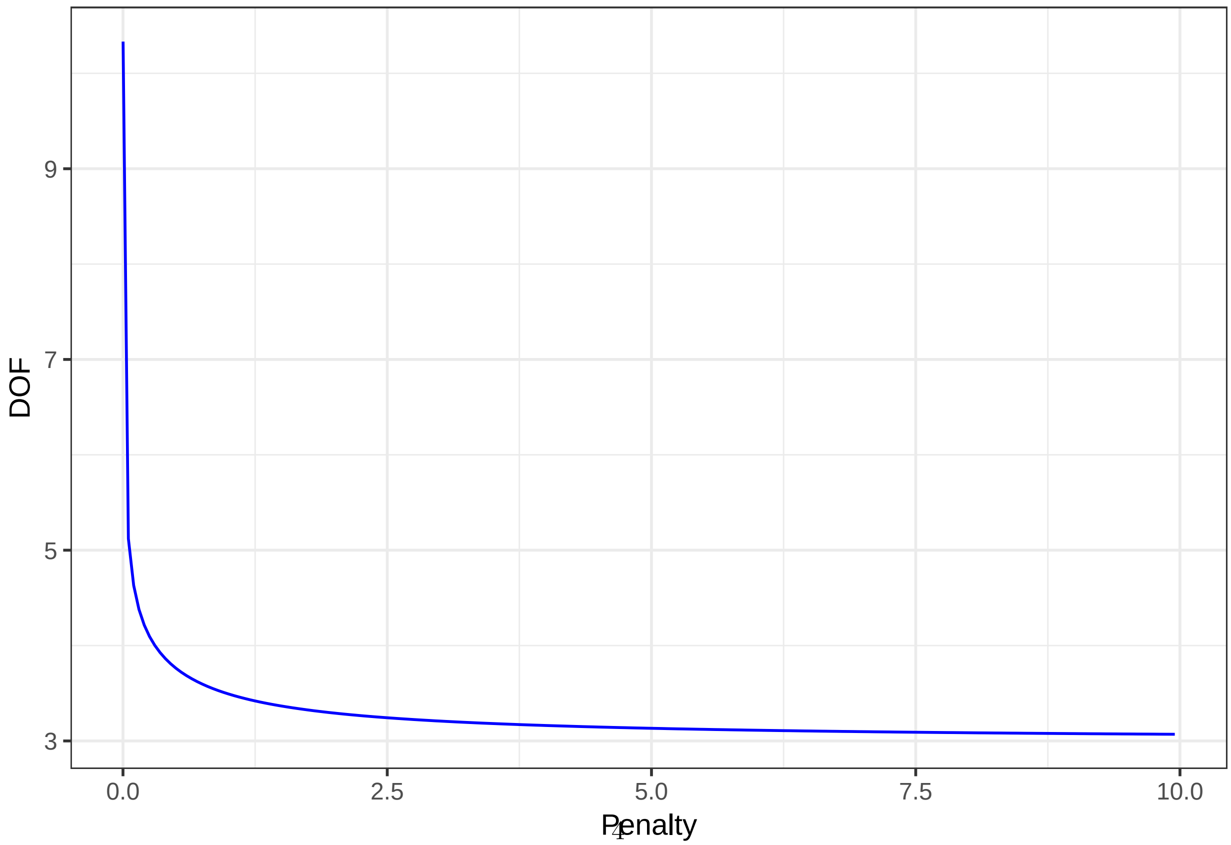
The original mortality rate varies widely with time but the predicted mortality rate seems to exhibit the same pattern of peaks for every peaks of original mortality. The predicted mortality fails to cover all the data points. This prediction does not seems to be good enough so, the quality of the fit is not good.

The p value for spline component *Week* is < .00000000000000022 and is much smaller than the conventional value of 0.05 that is often used as a criterion for statistical significance. The p value for both intercept and year is greater than 0.05. Hence, Week term appears to be significant in the model. The mortality rate rises at the beginning and end of every year.

From the plot of spline component (Week), it is observed that the mortality rate decreases in the middle of the year i.e, in 20th - 30th week of the year. This is normally the summer season in every year.

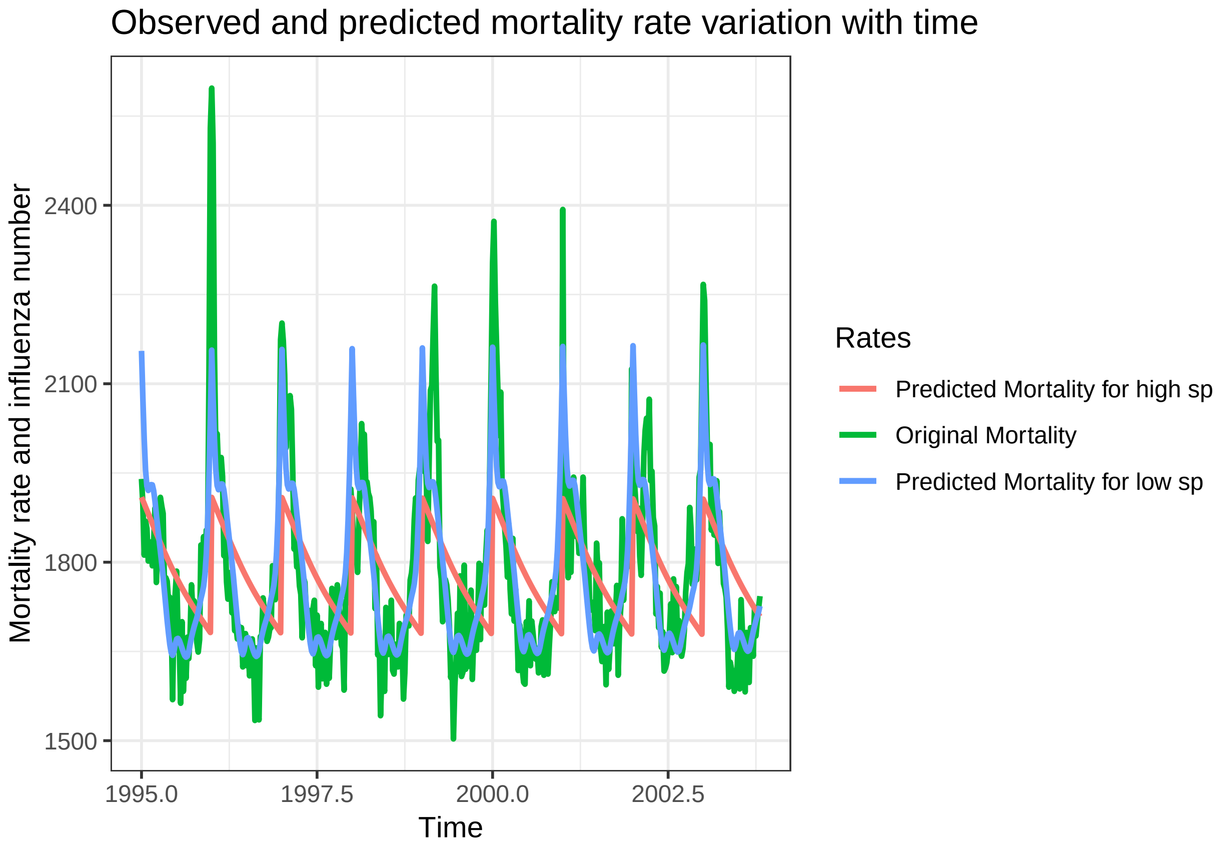
1.4 How the penalty factor of the spline function influences the estimated deviance of the model





Analysis:

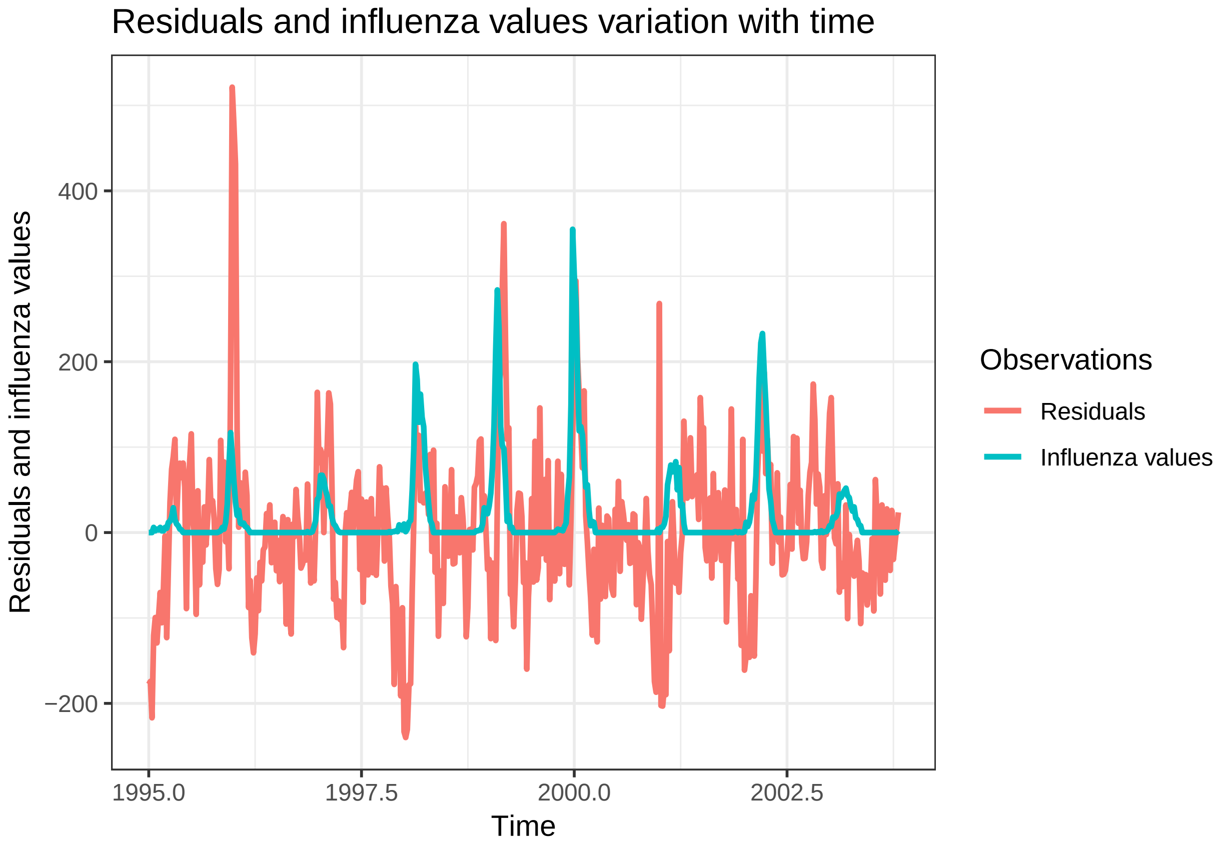
From the plot of penalty against deviance and degrees of freedom, it is observed that when the penalty factor λ increases, the deviance also increases. But the degrees of freedom df_λ decreases. Higher the λ, penalization is higher.



Analysis:

The predicted and observed mortality against time for cases of very high (sp=10) and very low (sp=-10) penalty factor were plotted. Both predicted mortality rates follows some constant pattern. It is observed that, when the penalty factor is high, the complexity of the model is less. And when λ is less, complexity will be high.

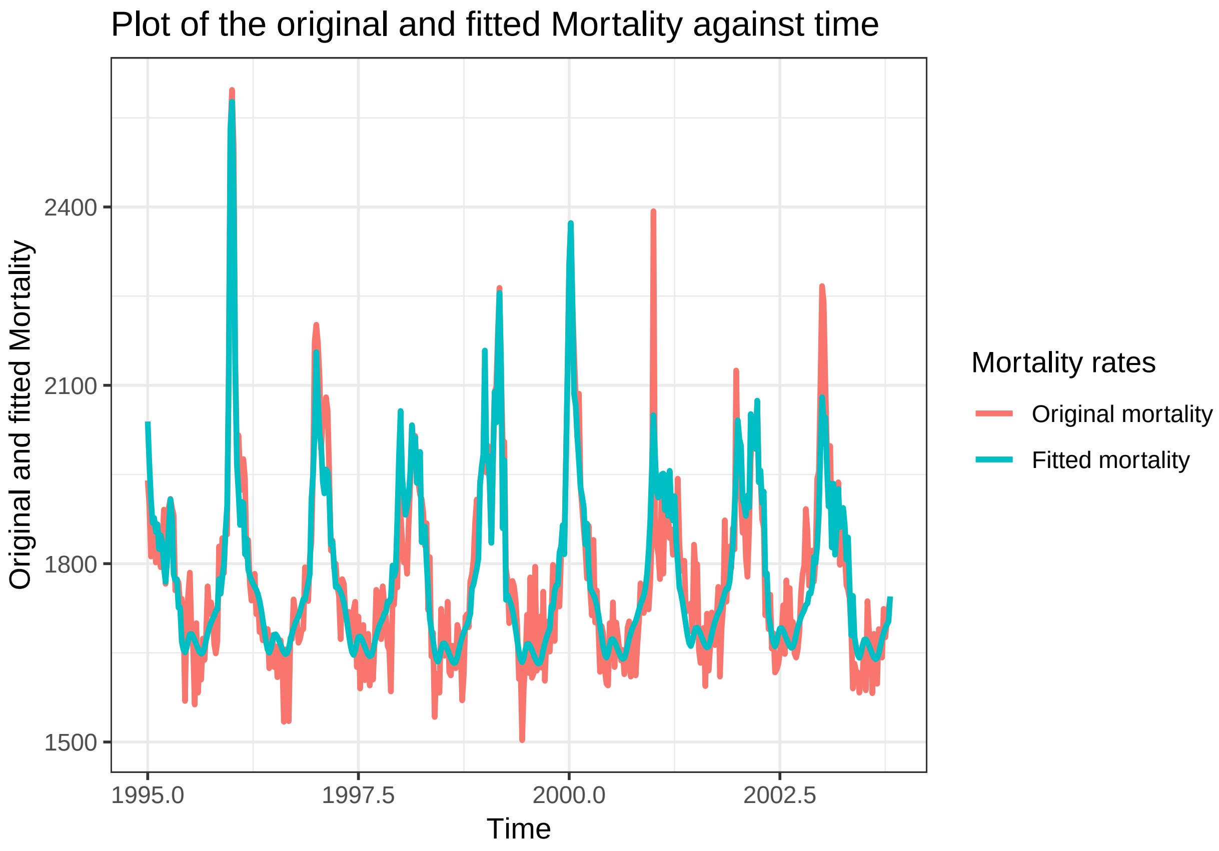
1.5 Plot the residuals and the influenza values against time



Analysis:

Some of the peaks of influenza values are correlated to the peaks of residuals. But many residuals seems to be missed by the influenza values. So, the temporal pattern in the residuals does not correlated much to the outbreaks of influenza.

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



Analysis:

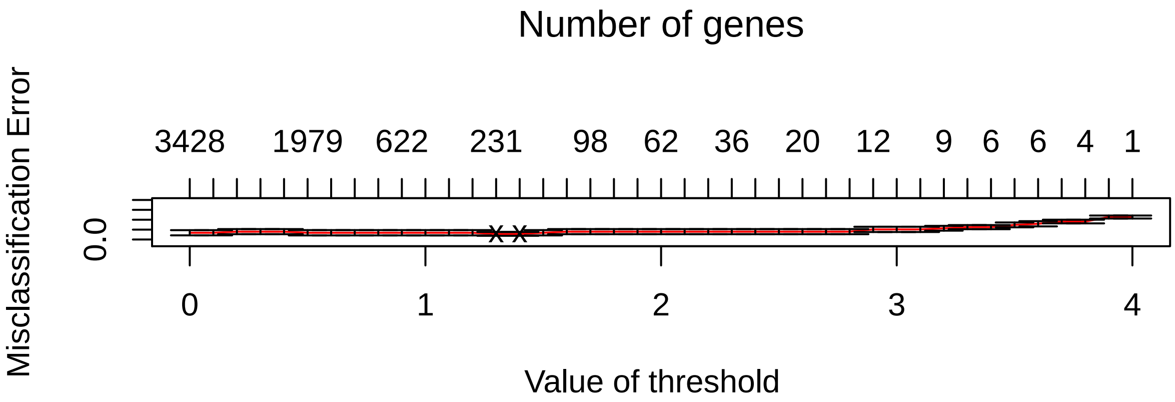
Now the mortality is modelled as an additive function of the spline functions of year, week, and the number of confirmed cases of influenza. The p value for intercept and spline components *Week* and *Influenza* is < .00000000000000022 and is much smaller than the conventional value of .05. And also the fitted mortality seems to be covering all the data points and the peaks matches the peaks of original mortality. The mortality is well influenced by the outbreaks of influenza and this model seems to be better than the previous GAM models.

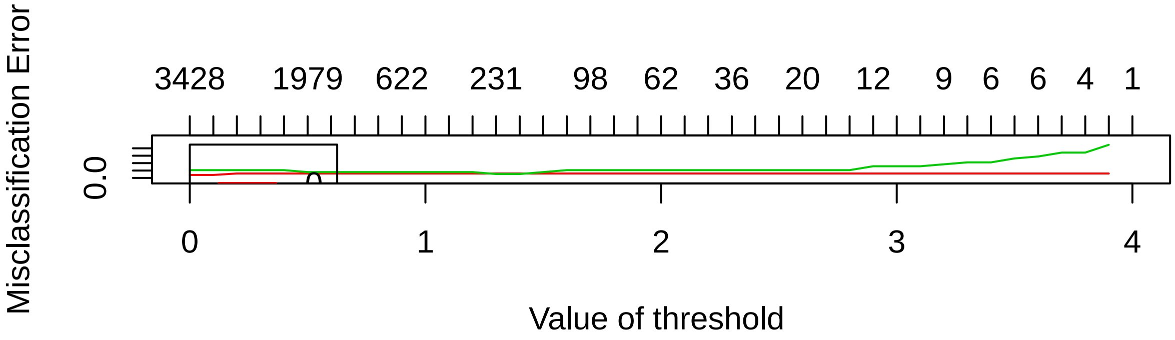
Assignment 2. High-dimensional methods

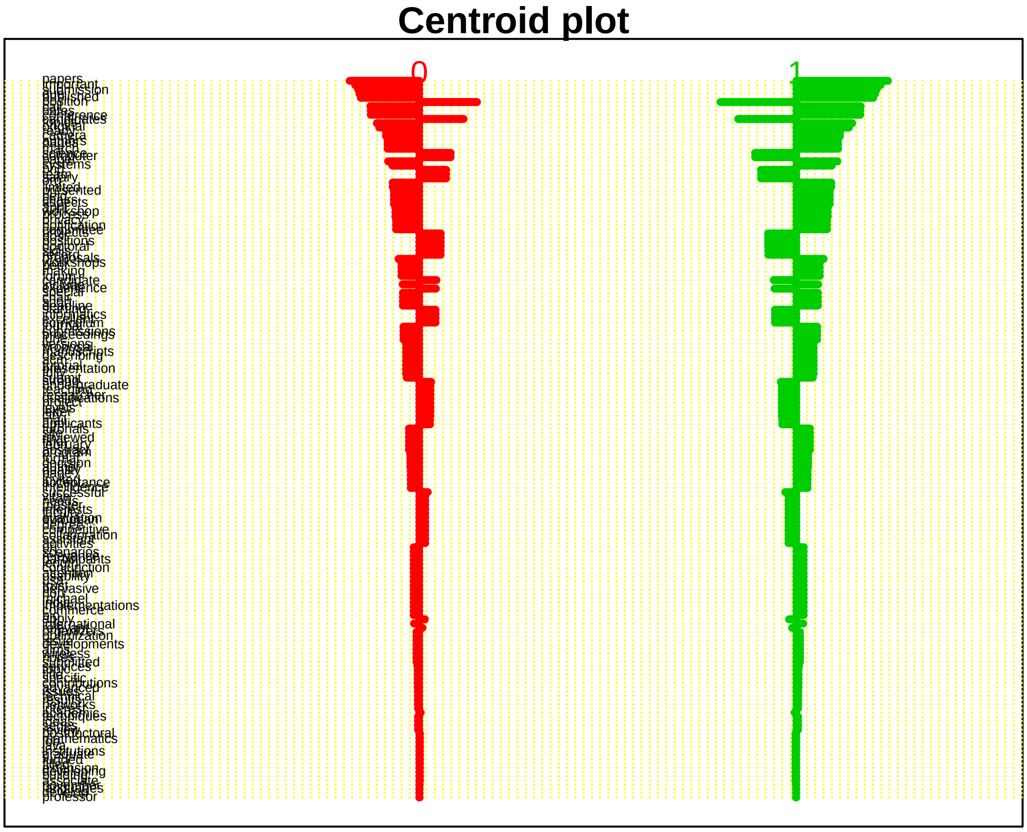
2.1 Perform nearest shrunken centroid classification of training data

```
## 1234567891011121314151617181920212223242526272829303132333435363738394041
```

```
## 12Fold 1 :1234567891011121314151617181920212223242526272829303132333435363738394041
## Fold 2 :1234567891011121314151617181920212223242526272829303132333435363738394041
## Fold 3 :1234567891011121314151617181920212223242526272829303132333435363738394041
## Fold 4 :1234567891011121314151617181920212223242526272829303132333435363738394041
## Fold 5 :1234567891011121314151617181920212223242526272829303132333435363738394041
## Fold 6 :1234567891011121314151617181920212223242526272829303132333435363738394041
## Fold 7 :1234567891011121314151617181920212223242526272829303132333435363738394041
## Fold 8 :1234567891011121314151617181920212223242526272829303132333435363738394041
## Fold 9 :1234567891011121314151617181920212223242526272829303132333435363738394041
## Fold 10 :1234567891011121314151617181920212223242526272829303132333435363738394041
```







```
## 10 most contributing features

## [[1]]
## [1] "acceptance"
##
## [[2]]
## [1] "X59â"
##
## [[3]]
## [1] "adhere"
##
## [[4]]
## [1] "X1st"
##
## [[5]]
## [1] "acquiring"
##
## [[6]]
## [1] "accessibility"
##
## [[7]]
## [1] "agenda"
##
## [[8]]
## [1] "aicit"
##
## [[9]]
## [1] "X5102011"
##
## [[10]]
## [1] "agents"
```

```
## [1] "Nearest shrunken centroid classification"
```

```
## [1] "Confusion matrix for test data : "
```

```
##       pred
## actual  0  1
##      0 10  0
##      1  2  8
```

```
## Misclassification error rate of test data :  0.1
```

Analysis:

The model is fitted with train data and the threshold values are choosen by cross validation. The two cross symbols specifies the lowest threshold values and the optimal threshold values are ** 1.3 and 1.4** .We choosen *1.4* since 170 non zeros (number of genes) are selected with 5 errors (min error) in cvmodel.

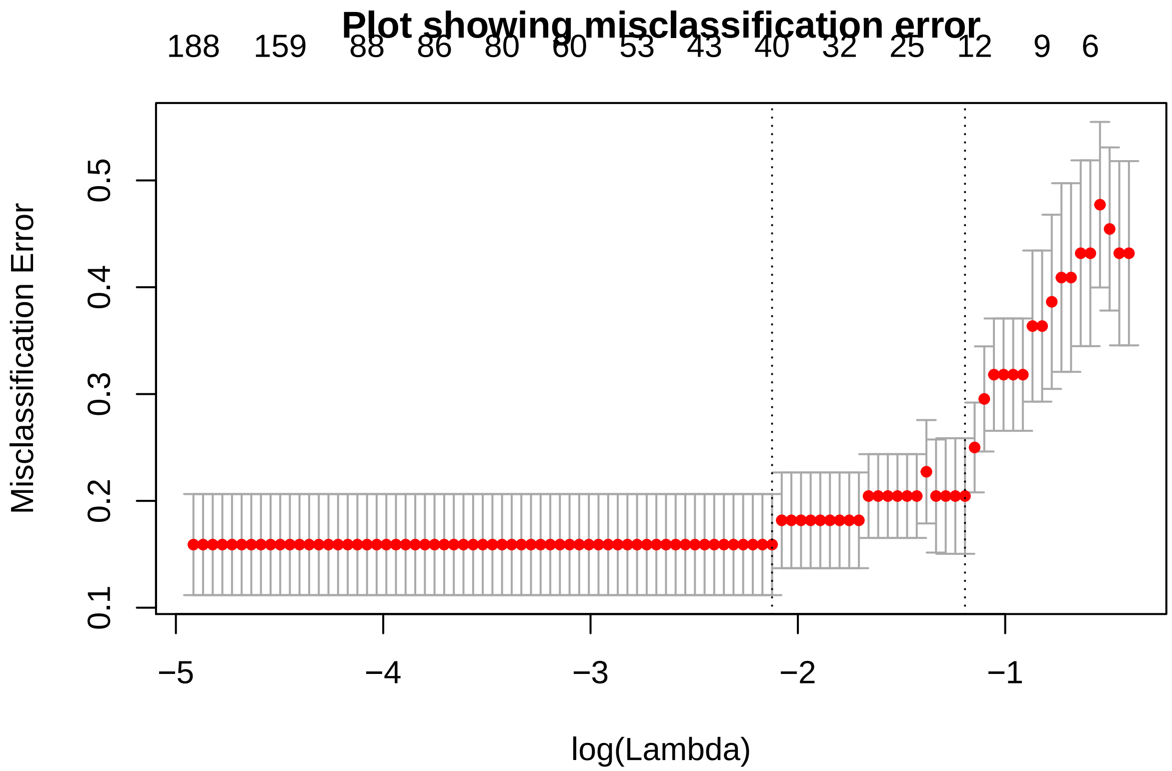
The shrunken class centroids for each class (0s and 1s), for genes surviving the threshold for at least once class is plotted. This shrinkage consists of moving the centroid towards zero by threshold, setting it equal to zero if it hits zero. This make the classifier more accurate by reducing the effect of noisy genes.

Totally 170 features have been selected. The top 10 features are listed above. Few words like acceptance, agenda, agents, X5102011 are reasonable that they have strong effect on the discrimination between the conference mails and other mails. For example, X5102011 word can be used for notifying conference date.

It seems that 8 emails (conference mails) are classified correctly. The test error rate is 0.1.

2.2 Compute the test error and the number of the contributing features

Elastic net with the binomial response



```
## [1] "Number of contributing features using Elastic net:  41"
```

```
## [1] "Elastic Net"
```

```
## [1] "Confusion matrix for test data : "
```

```
##       pred
## actual  0  1
##      0 10  0
##      1  2  8
```

```
## Misclassification error rate of test data :  0.1
```

Analysis:

Using elastic net method, the confusion matrix depicts that 8 emails (conference mails) are classified correctly. The test error is 0.1. The number of contributing features is 41.

Support vector machine with “vanilladot” kernel

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

```
## [1] "Number of contributing features using SVM:  43"
```

```
## [1] "SVM"
```

```
## [1] "Confusion matrix for test data : "
```

```
##       pred
## actual  0  1
##      0 10  0
##      1  1  9

## Misclassification error rate of test data :  0.05
```

Analysis:

Using SVM, the model is trained with linear kernel (vanilladot). The confusion matrix depicts that 9 emails (conference mails) are classified correctly. The test error is 0.05 which seems to be lesser than elastic method. The number of support vectors is 43. This suggests that 43 features have been selected by this model.

Comparative table:

Methods	Test errors	Number of features
Nearest shrunken centroid	0.1	170
Elastic net	0.1	41
SVM	0.05	43

Analysis:

From the above table, we can conclude that the SVM method is the best amongst all the three methods. The test error is lesser than the nearest shrunken and elastic net method. The accuracy seems to be high for SVM.

2.3 Implementing Benjamini-Hochberg method

```
## [1] "Selected features : "

##   [1] "abstract"        "academic"        "acceptance"      "accepted"        "access"          "acm
##  [28] "bio"             "call"            "calls"           "camera"          "canada"          "canc
##  [55] "contributions"   "copyright"       "covering"        "cross"           "curriculum"      "date
##  [82] "expected"        "experience"      "extension"       "feature"         "february"        "figu
## [109] "include"         "included"        "india"           "infrastructures" "initially"       "inst
## [136] "letter"          "levels"          "limited"         "liu"             "looking"         "madi
## [163] "ontologies"      "opportunity"     "optimization"    "org"             "organizers"      "orga
## [190] "privacy"         "proceedings"     "process"         "professor"       "proficiency"     "prog
## [217] "scalability"     "scenarios"       "science"         "scope"           "security"        "seri
## [244] "taiwan"          "takes"           "tasks"           "teaching"        "team"            "tech
## [271] "versions"        "vienna"          "visualization"   "vitae"           "wang"            "wire

## [1] "Number of features :  281"
```

Analysis:

The Benjamini-Hochberg method is implemented and as instructed t.test is used to obtain various p-values for the two sided test for feature assessment.

The selected features are totally 281 with p values (>=0.05) and they are displayed above. The number of selected features is greater than all the other methods implemented before. This is not the efficient one since it selects more features that are more unrelated to the conference mails.

Appendix

```r
################################## Assignment 1 ####################################################
# data splitting
library(xlsx)
library(ggplot2)
influenza_df <- read.xlsx("influenza.xlsx",1)
#plotting mortality
ggplot(influenza_df)+
geom_line(aes(x=Time, y=Mortality,color="blue"),size=1)+
  geom_line(aes(x=Time, y=Influenza,color="red"),size=1)+
  ggtitle("Mortality rate and influenza number variation with time")+
  ylab("Mortality rate and influenza number")+
  scale_color_discrete(name = "Rates", labels = c("Mortality rate","Influenza number"))+
  theme_bw()

library(mgcv)
set.seed(12345)
# The prediction error criteria used are Generalized (Approximate) Cross Validation
#(GCV or GACV) when the scale parameter is unknown
influenza_model <- gam(Mortality~Year+s(Week), data=influenza_df, method = "GCV.Cp")
fit<-fitted(influenza_model)
influenza_df$Pred_mortality <- fit
#predict.gam(fit,influenza_df$Mortality)
# plotting two mortality rates
ggplot(influenza_df)+
geom_line(aes(x=Time, y=Mortality,color="blue"),size=1)+
  geom_line(aes(x=Time, y=Pred_mortality,color="red"),size=1)+
  ggtitle("Observed and predicted mortality rate variation with time")+
  ylab("Mortality rate and influenza number")+
  scale_color_discrete(name = "Rates", labels = c("Mortality rate","Predicted mortality rate"))+
  theme_bw()

# output of model
summary(influenza_model)

# plotting spline component "WEEK"
library(visreg)
paste("Spline component (Week) plot")
visreg(influenza_model, "Week", gg=F,type="contrast")
penalty <- seq(0.001,10,0.05)
full_df <- data.frame()
for (i in penalty) {
  influenza_model_0 <- gam(formula = Mortality ~ Year +
                           s(Week, k=length(unique(influenza_df$Week))),
                           data = influenza_df, sp=i)
  dof <- sum(influence(influenza_model_0))
  dev <- influenza_model_0$deviance
  dataf <- data.frame(Dev=dev,DOF=dof)
  full_df <- rbind(full_df, dataf)
}
full_df$Penalty <- penalty
# plot of penalty factor against deviance
```

```r
ggplot(full_df)+geom_line(aes(Penalty,Dev),colour="red")+theme_bw()
# plot of penalty factor against DOF
ggplot(full_df)+geom_line(aes(Penalty,DOF),colour="blue")+theme_bw()
# low penalty and high penalty
influenza_model_0 <- gam(Mortality ~ Year + s(Week, k=length(unique(influenza_df$Week)),
                        sp=-10), data=influenza_df)
fit_0<-fitted(influenza_model_0)
influenza_df$Pred_mortality_0 <- fit_0


influenza_model_1 <- gam(Mortality ~ Year + s(Week, k=length(unique(influenza_df$Week)),
                         sp=10),data=influenza_df)
fit_1<-fitted(influenza_model_1)
influenza_df$Pred_mortality_1 <- fit_1

ggplot(influenza_df)+
geom_line(aes(x=Time, y=Mortality,color="blue"),size=1)+
  geom_line(aes(x=Time, y=Pred_mortality_1,color="black"),size=1)+
  geom_line(aes(x=Time, y=Pred_mortality_0,colour="red"),size=1)+
  ggtitle("Observed and predicted mortality rate variation with time")+
  ylab("Mortality rate and influenza number")+
  scale_color_discrete(name = "Rates", labels = c("Predicted Mortality for high sp",
                      "Original Mortality","Predicted Mortality for low sp"))+
  theme_bw()
residuals <- influenza_model$residuals
influenza_df$Residuals <- residuals
ggplot(influenza_df)+
geom_line(aes(x=Time, y=Residuals,color="blue"),size=1)+
  geom_line(aes(x=Time, y=Influenza,color="red"),size=1)+
  ggtitle("Residuals and influenza values variation with time")+
  ylab("Residuals and influenza values")+
  scale_color_discrete(name = "Observations", labels = c("Residuals","Influenza values"))+
  theme_bw()
influenza_model1 <- gam(Mortality~s(Year,k=length(unique(influenza_df$Year)))+
                        s(Week, k=length(unique(influenza_df$Week)))+
                        s(Influenza, k=length(unique(influenza_df$Influenza))),
                        data=influenza_df, method = "GCV.Cp")

fit1<-fitted(influenza_model1)
influenza_df$Pred_mortality1 <- fit1
#predict.gam(fit,influenza_df$Mortality)
# plotting two mortality rates
ggplot(influenza_df)+
geom_line(aes(x=Time, y=Mortality,color="blue"),size=1)+
  geom_line(aes(x=Time, y=Pred_mortality1,color="red"),size=1)+
  ggtitle("Plot of the original and fitted Mortality against time")+
  ylab("Original and fitted Mortality")+
  scale_color_discrete(name = "Mortality rates", labels = c("Original mortality",
                       "Fitted mortality"))+theme_bw()

################################## Assignment 2 ####################################################
email_df <- read.csv("data.csv",sep = ";")
#Divide data into training and test sets (70/30)
```

```
n=dim(email_df)[1]  #nrow(email_df)
set.seed(12345)
id=sample(1:n, floor(n*0.7))
train=email_df[id,]
test=email_df[-id,]

#Perform nearest shrunken centroid classification of training data in which
#the threshold is chosen by cross-validation
library(pamr)
email_df$Conference <- as.factor(email_df$Conference)

rownames(train)=1:nrow(train)
x=t(train[,-4703])
y=train[[4703]]
mydata=list(x=x,y=as.factor(y),geneid=as.character(1:nrow(x)), genenames=rownames(x))
model=pamr.train(mydata,threshold=seq(0,4, 0.1))

# cross validation
cvmodel=pamr.cv(model,mydata)
#print(cvmodel)
pamr.plotcv(cvmodel)

#Provide a centroid plot and interpret it
pamr.plotcen(model, mydata, threshold=1.4)
title("Centroid plot")
#List of 10 features
a <- as.data.frame(pamr.listgenes(model,mydata,threshold=1.4))
sel_features <- colnames(email_df)[(a[,1])]
#cat( paste( head(sel_features,10), collapse='\n' ) )
f<-as.list(head(sel_features,10))
cat( paste( "10 most contributing features") )
f


# test error
rownames(test)=1:nrow(test)
x=t(test[,-4703])
actual=test[[4703]]
pred <- pamr.predict(model,x,threshold=1)
matrixx <- table(actual,pred)
paste("Nearest shrunken centroid classification")
paste("Confusion matrix for test data : ")
matrixx
test_misclassification_error <- 1 - sum(diag(matrixx))/sum(matrixx)
cat(paste("Misclassification error rate of test data : ", test_misclassification_error))


# Elastic net with the binomial response
library(glmnet)
set.seed(12345)
x=as.matrix(train[,-4703])
y=as.matrix(train[,4703])
# fitting
```

```
cv_fit <- cv.glmnet(x, y, alpha=0.5, family = "binomial", type.measure = "class")
# We plot the object and show the optimal values of lambda
plot(cv_fit)
title("Plot showing misclassification error")
# Num of features
num_features <- coef(cv_fit, s = "lambda.min")
Elasticnet_features <- as.data.frame(num_features@x)
paste("Number of contributing features using Elastic net: ",nrow(Elasticnet_features))

# test error
x=as.matrix(test[,-4703])
actual=test[,4703]
pred <- predict(cv_fit,x, type="class")
matrixx <- table(actual,pred)
test_misclassification_error <- 1 - sum(diag(matrixx))/sum(matrixx)
paste("Elastic Net")
paste("Confusion matrix for test data : ")
matrixx
cat(paste("Misclassification error rate of test data : ", test_misclassification_error))

# Support vector machine with "vanilladot" kernel

library(kernlab)
set.seed(12345)
x=as.matrix(train[,-4703])
y=as.matrix(train[,4703])
# fitting
SVM_model <- ksvm(x, y, kernel ="vanilladot", type="C-svc")  # vanilladot = Linear kernel

# number of contributing features
Svm_features <- as.data.frame(SVM_model@coef)
paste("Number of contributing features using SVM: ",nrow(Svm_features))

# test error
x=as.matrix(test[,-4703])
actual=test[,4703]
pred <- predict(SVM_model,x, type="response")
matrixx <- table(actual,pred)
test_misclassification_error <- 1 - sum(diag(matrixx))/sum(matrixx)
paste("SVM")
paste("Confusion matrix for test data : ")
matrixx
cat(paste("Misclassification error rate of test data : ", test_misclassification_error))
tabl <- "
|          Methods               |   Test errors |  Number of features |
|--------------------------------|:-------------:|:-------------------:|
| Nearest shrunken centroid      |      0.1      |       170           |
| Elastic net                    |      0.1      |       41            |
| SVM                            |      0.05     |       43            |
"
cat(tabl) # output the table in a format good for HTML/PDF/docx conversion

set.seed(12345)
```

```r
# use t.test()
p <- sapply(1:(ncol(email_df)-1),
            function(x) t.test(email_df[,x] ~ Conference, data = email_df)$p.val)
# p-values
pvall<-as.data.frame(p)
# alpha values (threshold) False discovery rate
pvall$Type <- ifelse( p < 0.05,1,0)
pvall$col <- as.numeric(rownames(pvall))
select <- ifelse(pvall$Type == 1, pvall$col, 0)
# for selected features
library(dplyr)
e<- as.data.frame(select) %>% filter(select!=0)
sel_features <- colnames(email_df[,as.matrix(e)])
paste("Selected features : ")
sel_features

paste("Number of features : ", length(sel_features))
```