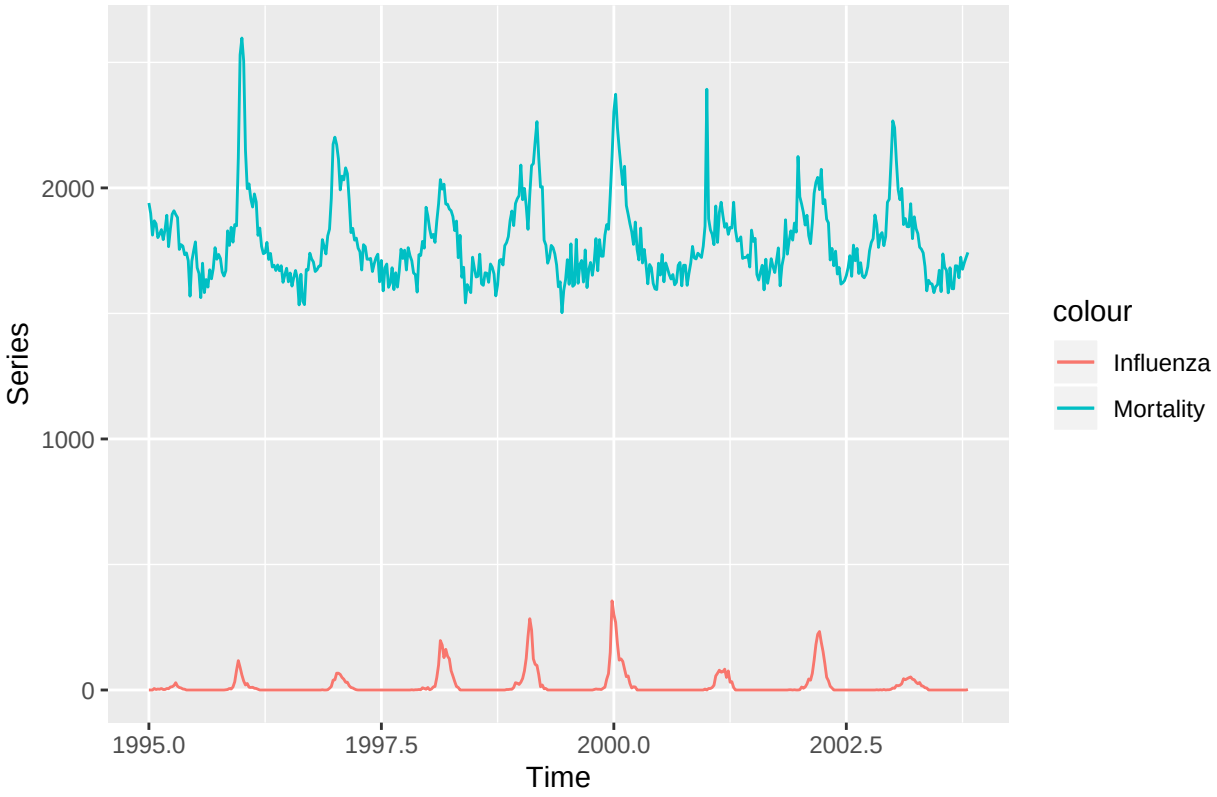
Lab2 Block2 Report

Samia Butt

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

1.1 Time Series Analysis of Mortality and Influenza



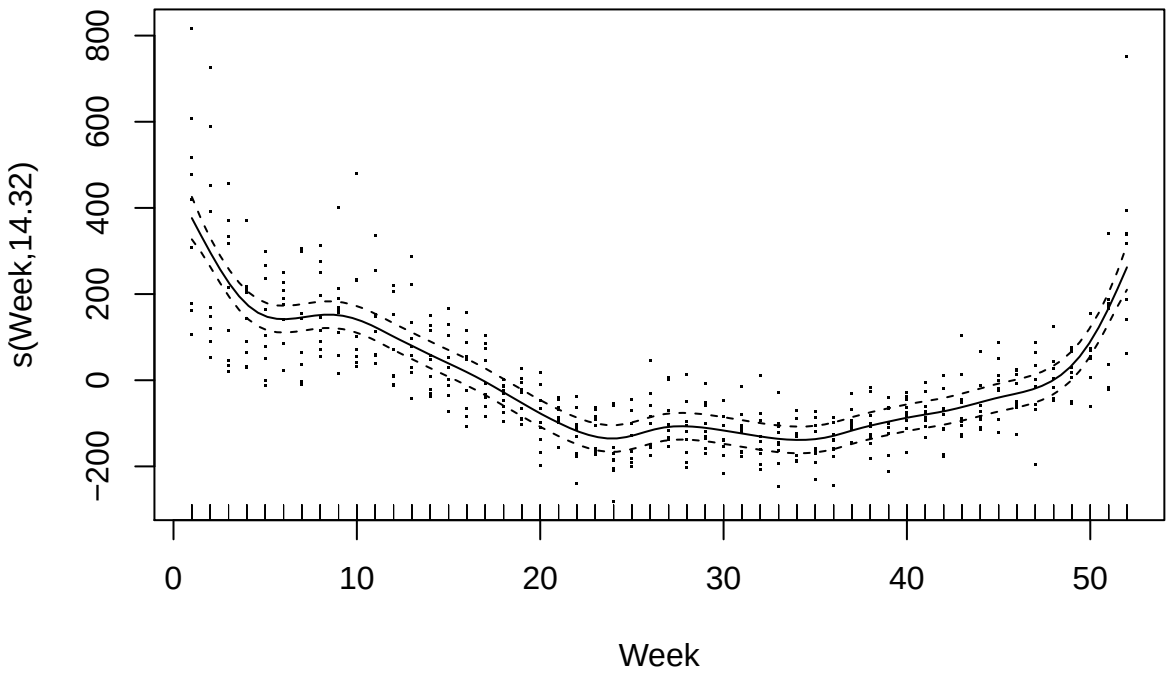
According to the above time series graph, as much as the Influenza increases, mortality is also increasing so we can conclude that there is a positive correlation between Mortality and Influenza.

1.2 GAM model with spline

Probablistic Model

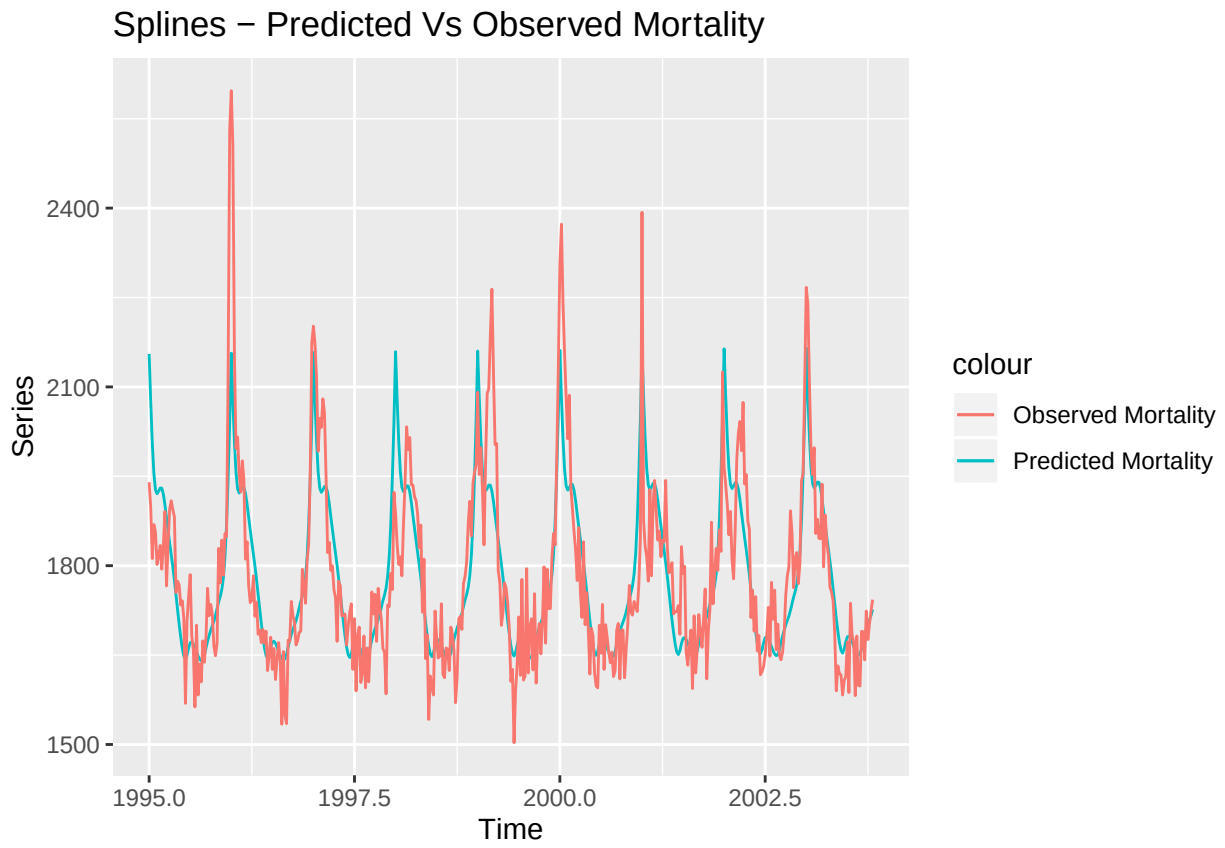
$$Mortality \sim \mathcal{N}(\beta_0 + \beta_1 * year + s(week), \sigma^2)$$

$$Mortality \sim \mathcal{N}(-680.5979826 + 1.2328461 * year + s(week), \sigma^2)$$



According to the above graph mortality rate is higher at the begining of the year and at the end of the year. Mortality rate is very low around middle of the year.

1.3 Prediction from GAM model



Quality of fit Predicted values have the same change rate as original values have e.g. for both observed and predicted values, higher mortality rate in the begning and at the end of the year, while in middle of the year mortality rate is lower. however, observed values have the higher and the lower peak points, while the predcited values cant reach to the peak points.

Trend in mortality change from one year to another?

Predicted and observed Mortality rate decreases in the begning and at the end of the year while in the middle of the year mortality rate is on its peak and this trend is same for almost all the years. Overall, we can say that mortality rate is changing within year but it is not changing between years.

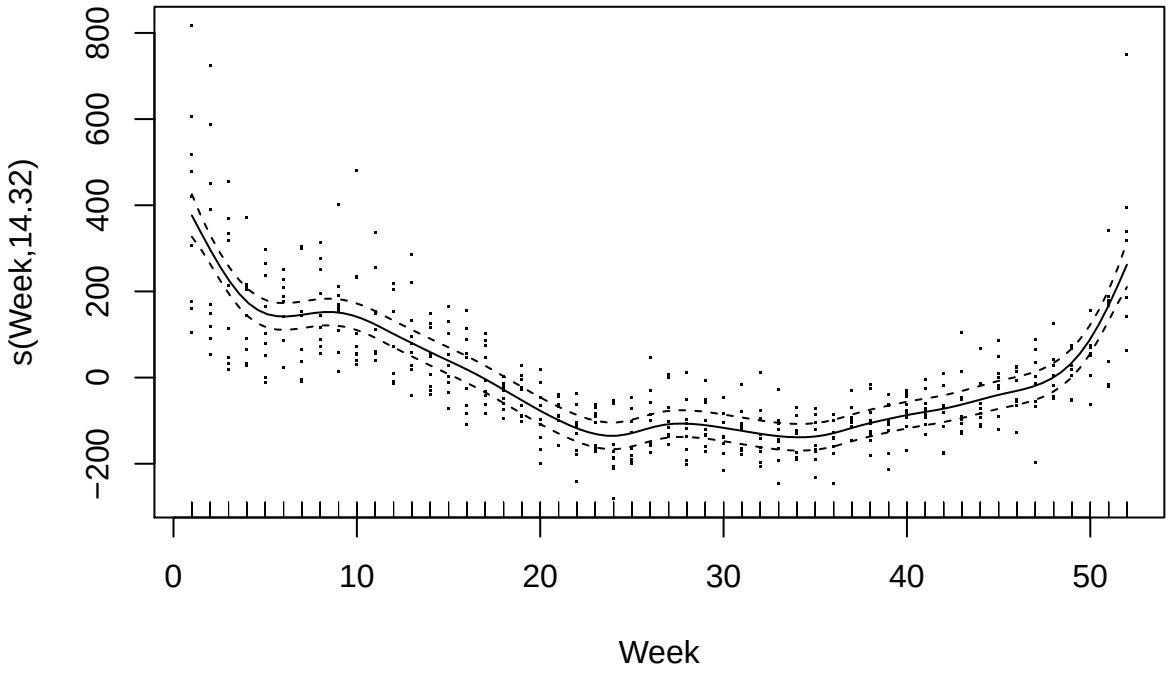
output of the GAM model & Significant terms in the model

```
##
## 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) -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
```

As we can see in the above model that Week has lowest p value so we can say that week is the most significant term in our model.

Plot interpretation



The above graph has Weeks on x-axis while the spline week on the y-axis and the dots show the residuals. Solid curvy line shows the relationship between the mortality and the weeks(in a year), which means mortality is high in the bigning and at the end of the year while in the middle of the year mortality is low.

1.4 GAM model analysis

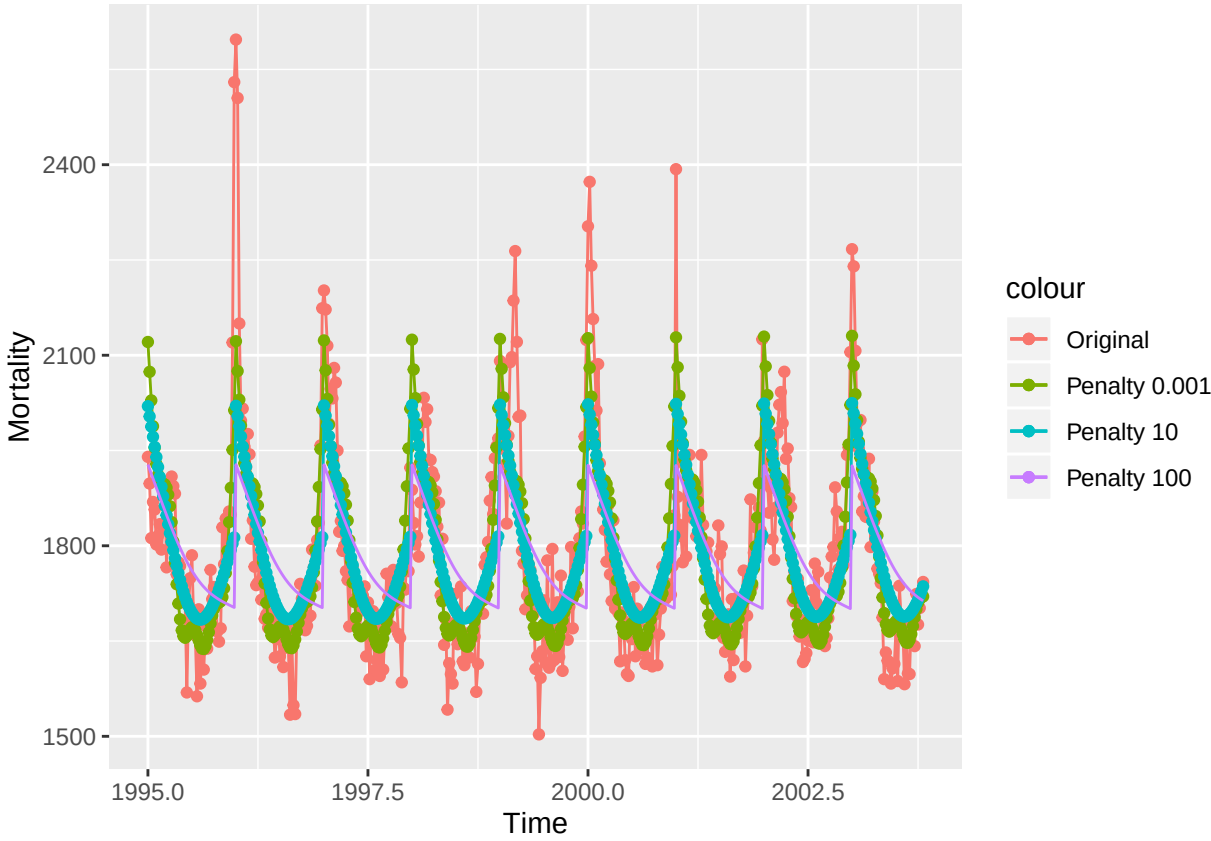
Examine how the penalty factor of the spline function in the GAM model from step2 influences the estimated deviance of the model.

```
##
## Family: gaussian
## Link function: identity
##
```

```
## Formula:
## Mortality ~ Year + s(Week, sp = 0.001)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -632.253   3448.117  -0.183    0.855
## Year           1.209      1.725   0.701    0.484
##
## Approximate significance of smooth terms:
##           edf Ref.df     F p-value
## s(Week) 8.881  8.996 100.3  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.661   Deviance explained = 66.8%
## GCV = 9018.9  Scale est. = 8805.1    n = 459
```

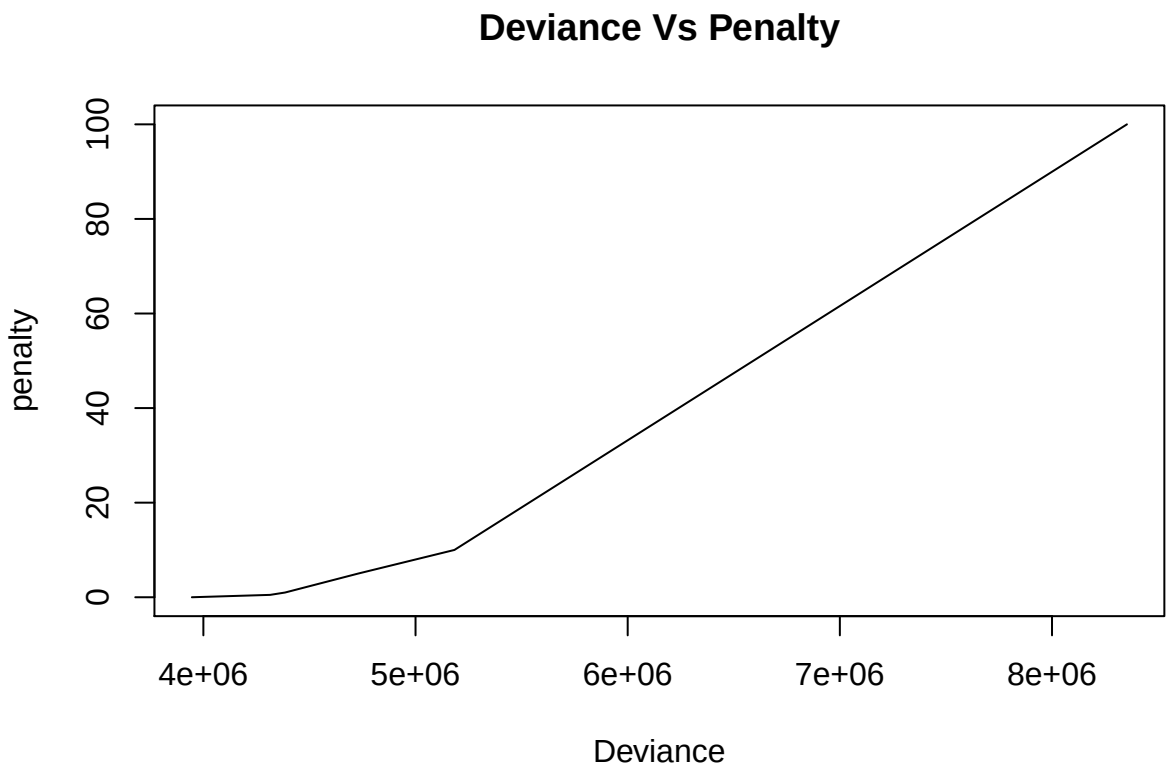
By comparing the results from step2 model summary and above summary we can see the that model with penalty factore has low deviance(3.9457189×10^6 %) while spline model without penalty factor has high deviance (3.7180121×10^6).

plots of the predicted and observed mortality against time for cases of very high and very low



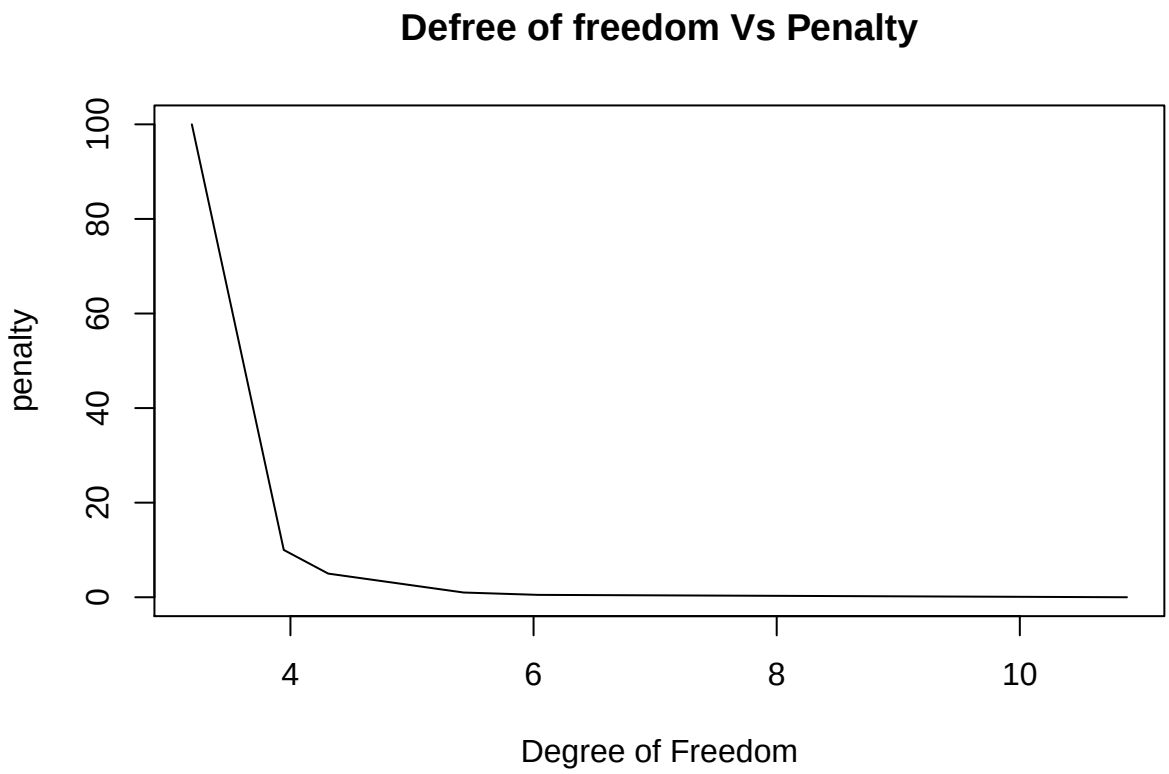
penalty factors

As we can see from the above penalty graph, for the lower penalty, model is covering almost all the points and tries to reach to the lower and uper extreme values, which makes our model overfit. While for very high penalty our model has becom underfit.



According to the above graph if penalty increases, deviance is also increasing.

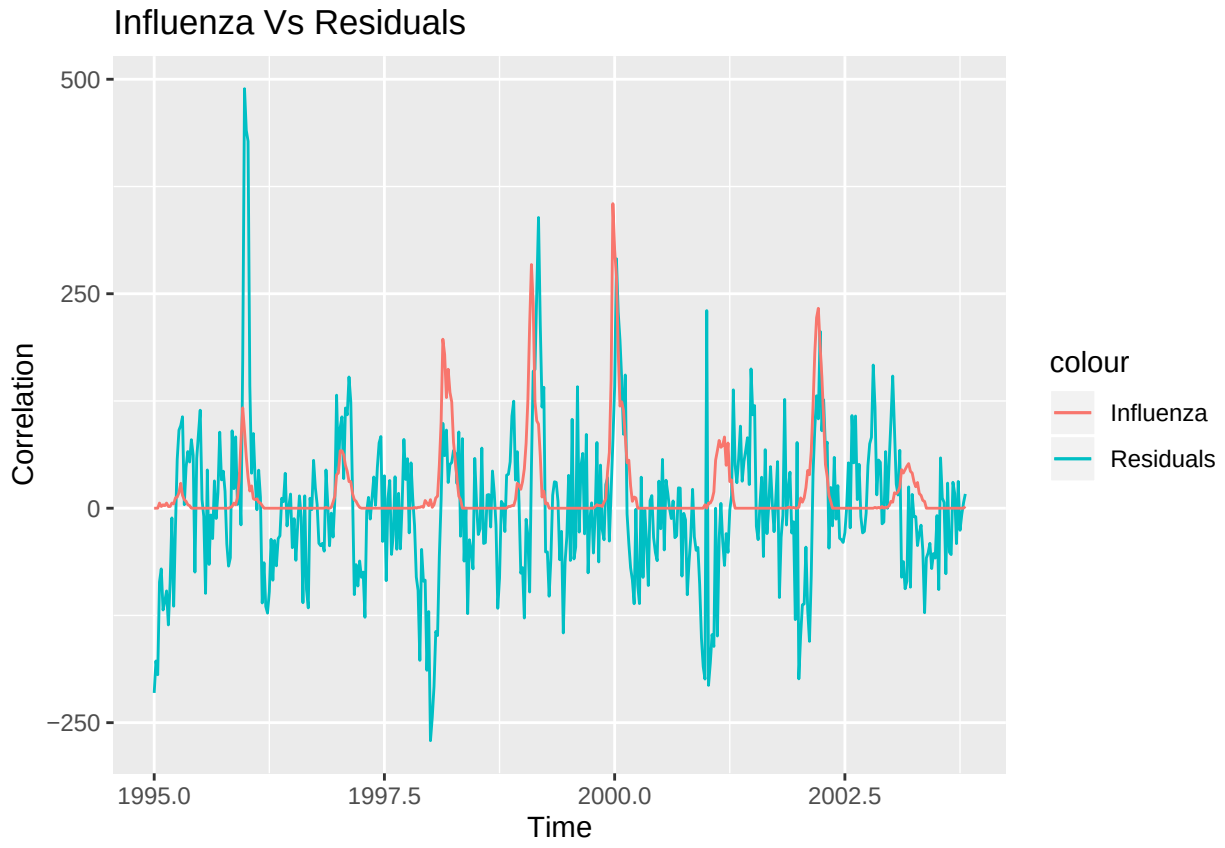
What is the relation of the penalty factor to the degrees of freedom? When we increase the penalty factor, degree of freedom starts decreasing, purple color lines in the above penalty graph demonstrate the highest penalty results, which shows that the model is not flexible, while the green lines shows the lowest penalty result, which depicts that our model is very flexible.



As we can see in the above graph as well that if penalty increases, degree of freedom starts decreasing.

Do your results confirm this relationship? By analysing the above graphs, we can conclude that our results confirms the relationship between penalty, deviance and degree of freedom.

1.5 Influenza Vs Residuals

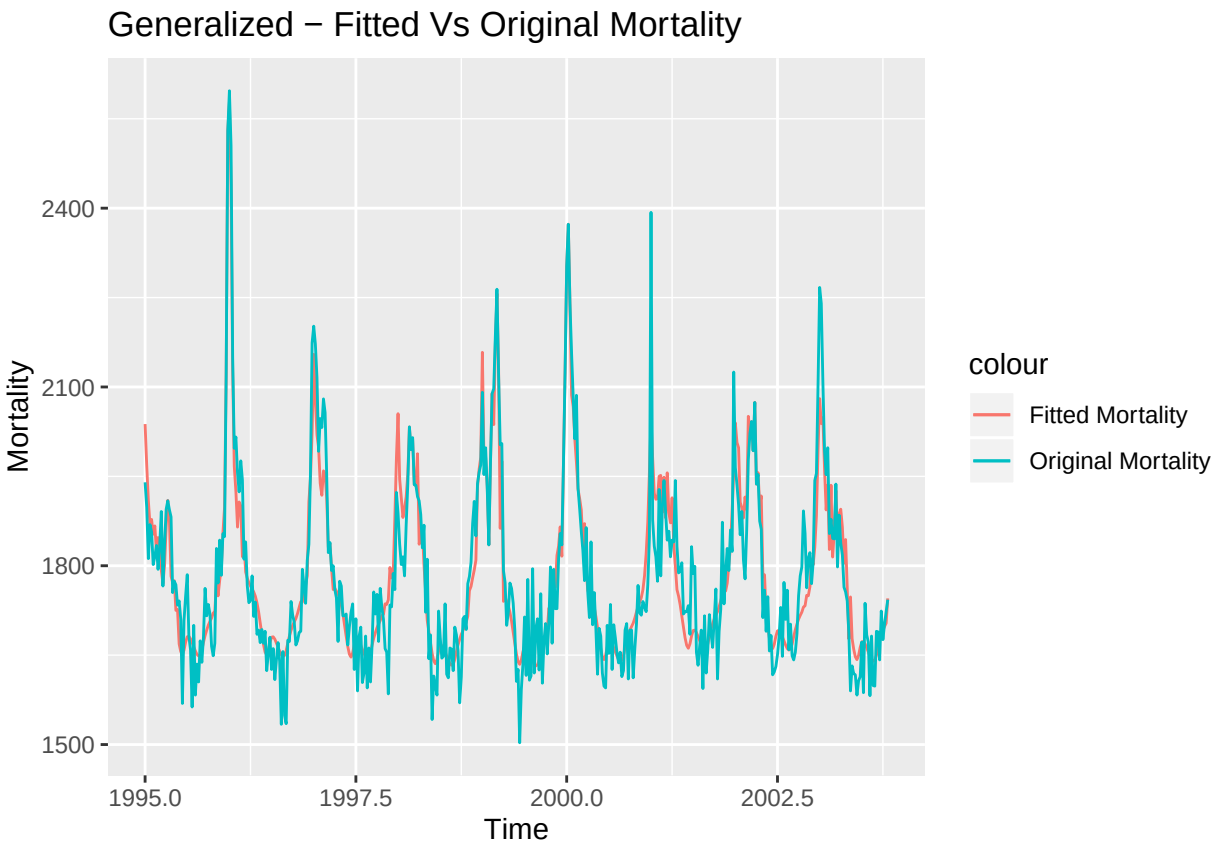


Is the temporal pattern in the residuals correlated to the outbreaks of influenza? According to the above graph, residuals are increasing as the influenza increases so we can conclude that the influenza and the Residuals are correlated to eachother.

1.6 GAM model with generalized spline

```
##
## 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
```

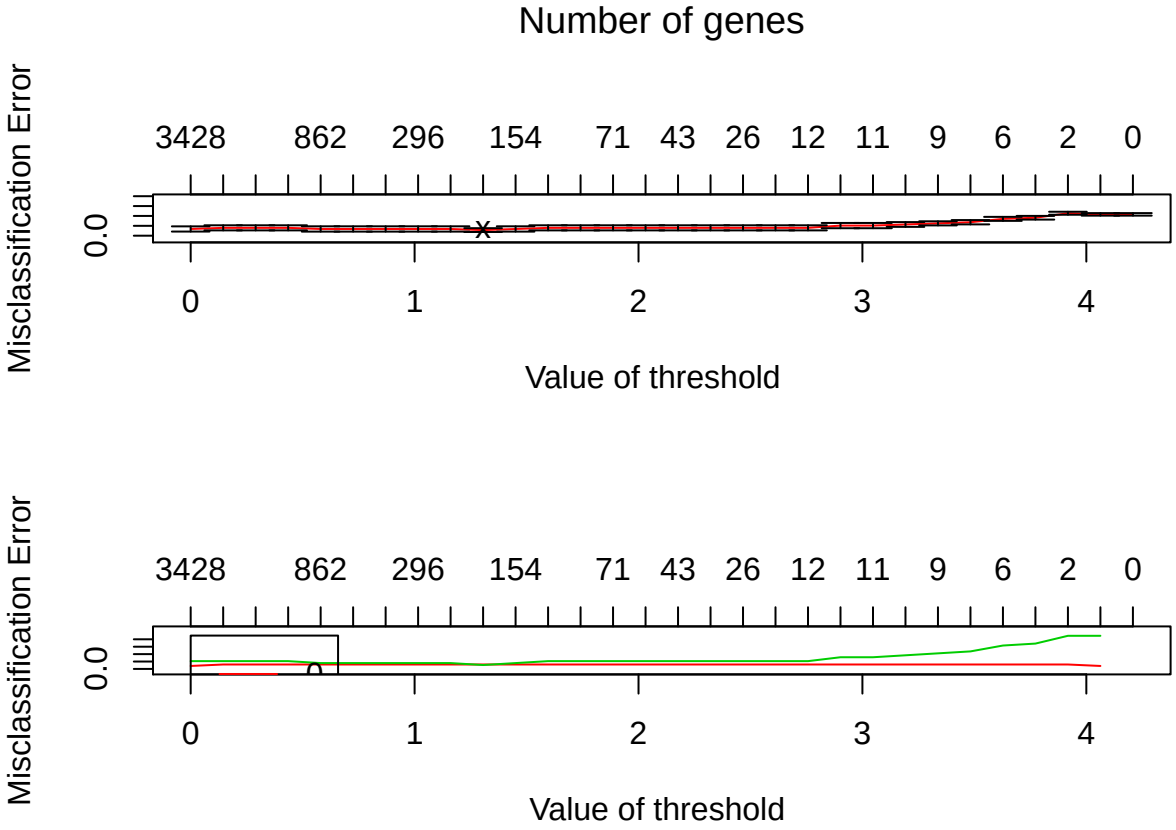


Comment on current gam model Vs previous gam model According the above graph, we can see that if we use generalized spline model then the model is working more accurately as compare to the previous model.

Mortality influence by the outbreak of influenza As we can see in the above results that by using influenza as spline function, now the model is trying to reach to the peak values, in other words we can say that our model has predicted the values which are alomost similar to the original values. By this behaviour of our model we can conclude that mortality has the influence by the outbreak of inluenza.

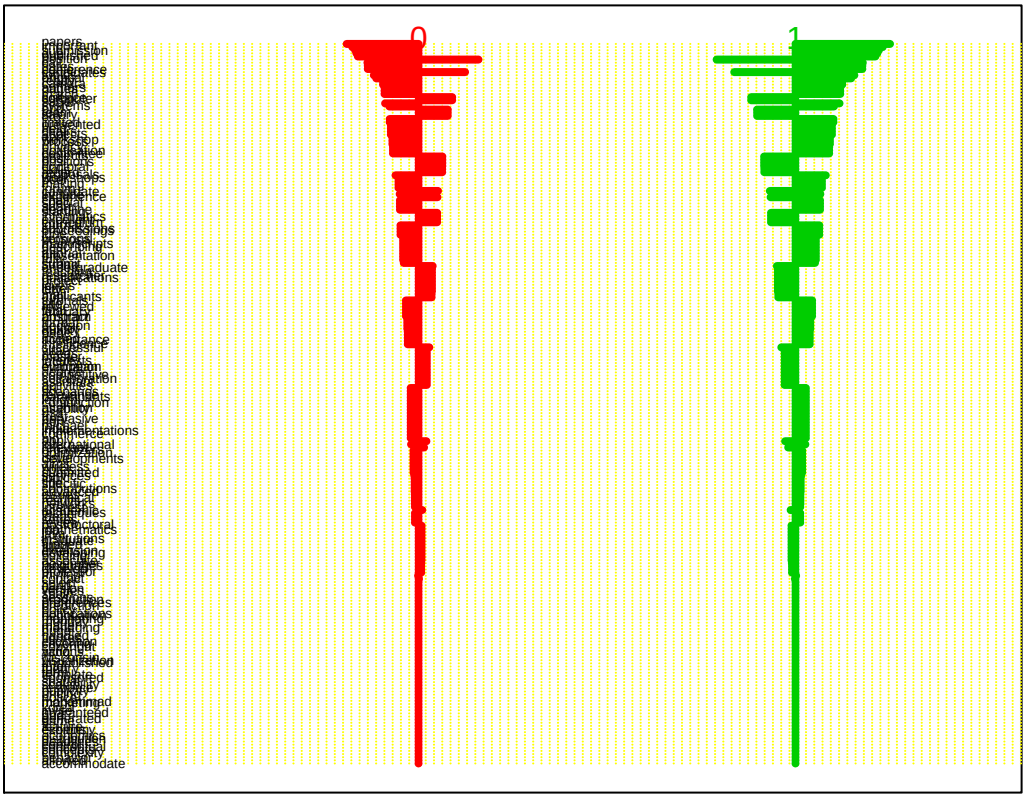
Assignment 2. High-dimensional methods

2.1 Nearest shrunken centroid classification



Threshold value selected by cross validation is 1.3059335.

2.1.1 Centroid Plot and its interpretation



As we have only two classes 0 and 1 so shrunken centriod is ploting two classes as red and green. Red denoting 0 class and green is denoting the class 1 threshold 1.3059335 while most left are the row names. In the above graph Shrunken centriod is shrinking features from top to bottom according to their highest significance to lowest significance respectively on the target variable.

2.1.2 Total Selected Features Total number of Selected features by model for threshold 1.3059335 are 231

2.1.3 10 Most significant Features

id	names	0-score1	1-score
3036	papers	-0.3814	0.5019
2049	important	-0.3519	0.4631
4060	submission	-0.3368	0.4431
1262	due	-0.3301	0.4344
3364	published	-0.3223	0.4241
3187	position	0.318	-0.4184
596	call	-0.2717	0.3575
869	conference	-0.2698	0.355
1045	dates	-0.2698	0.355
607	candidates	0.2468	-0.3247

As we have mentoned in step 2.1.1 that shrunken centriod shrinks the features from top to bottom according to their highest to lowest imapct on the target variable so we conclude that by selecting top 10 features means they have the highest impact on the target variable(Conference).

2.1.4 Error Rate

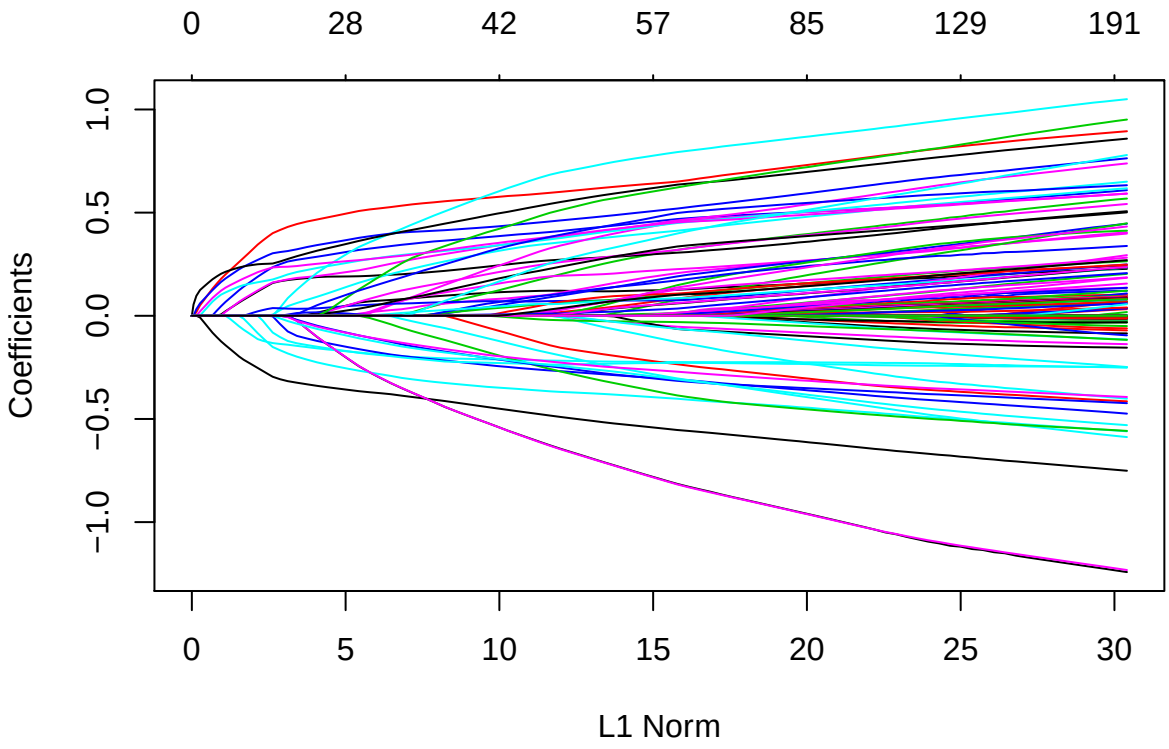
```
##          testdata
```

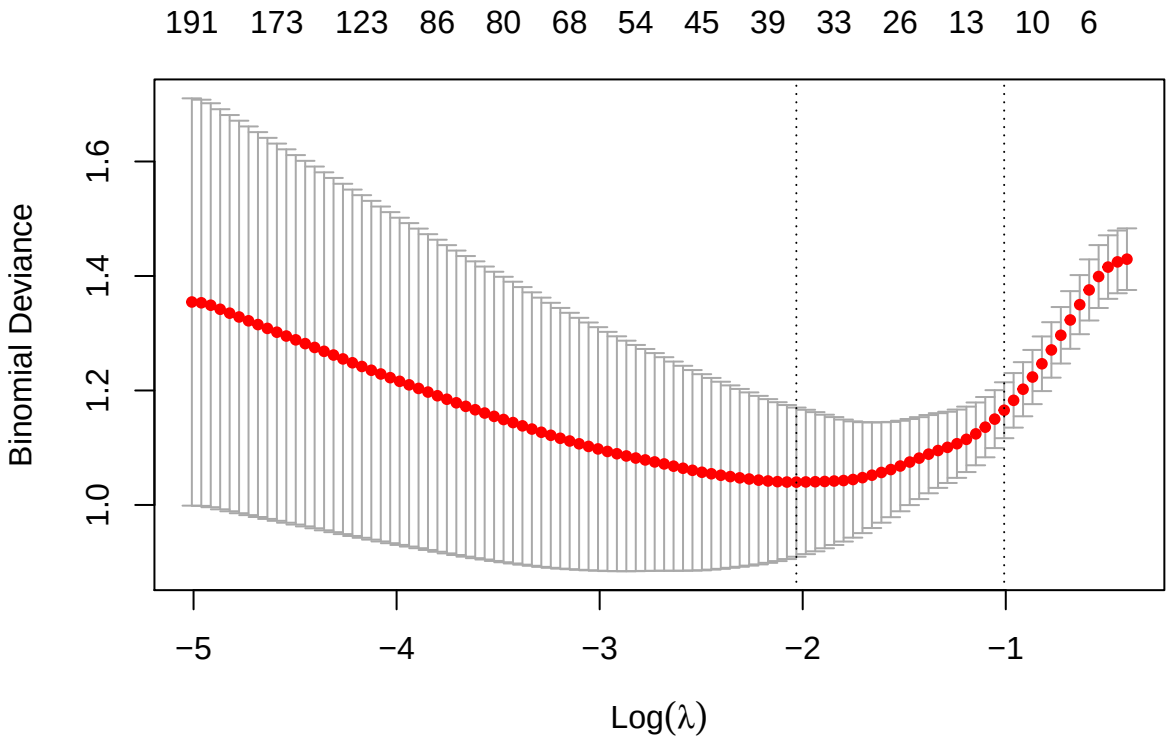
```
## prediction  0  1
##          0 10  2
##          1  0  8
```

Test error of Nearest shrunken centroid model is 0.1

2.2 Elastic net and Support vector machine Comparison

2.2.1 Elastic net method





Penalty selected by cross validation minimum lambda selected by cross validation is 0.1311628

selected features
abstracts
aspects
attention
bio
call
candidates
computer
conceptual
conference
dates
due
evaluation
exhibits
forum
important
interests
languages
making
manuscripts
original
papers
peer
position
privacy
projects
proposals
published
queries
record
relevant
scalability
scenarios
spatial
submission
systems
team
versions
visualization

Features Selection

By using elastic net model, total number of Selected features are 39

Error rate Test Error rate is 0.1

2.2.2 Support vector machine

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

```
## Support Vector Machine object of class "ksvm"
##
## SV type: C-svc  (classification)
##  parameter : cost C = 1
##
## Linear (vanilla) kernel function.
```

```
##
## Number of Support Vectors : 43
##
## Objective Function Value : -2.0817
## Training error : 0.022727
```

Total selected features total number of selected features by the SVM model are 43

Error rate

```
##           testy
## prediction  0  1
##          0 10  1
##          1  0  9
```

Error rate for the SVM model is 0.05

2.2.3 Comparison of models

Model_Type	Features	Error
Centriod	231	0.10
ElasticNet	39	0.10
SVM	43	0.05

By comparing the error rates of shrunken centroid, Elastic net and SVM, we can say that SVM is the best model with lowest error rate as 0.05.

While if we choose the model based on complexity, we will say that elastic net is less complex model as compare to Centriod or SVM.

However, Overall conclusion is that Nearest shrunken centriod model didnt work well while our selection moves around only Elasticnet and SVM.

2.3 Benjamini-Hochberg method

In this task we will perform Benjamini-Hochberg method by using alpha = 0.05. Sometimes very small p-values less than 5% happen by chance which can lead to incorrectly reject the null hypothesis so Benjamini-Hochberg method helps to avoid type1 error e.g. false postive.

```
pval <- sapply(1:(ncol(data2)-1), function(k){

   t.test(data2[,k]~Conference,data2, alternative = "two.sided")$p.value
})
```

```
tes = sapply(1:length(orderPval$pavlues), function(i){

  thresh = (i/length(orderPval$pavlues))*0.05

  if(orderPval$pavlues[i] <= thresh){
  paj_val <<- append(paj_val, orderPval$pavlues[i])
  padj_ind <<- append(padj_ind,orderPval$index[i])
  }
})

padjusted = cbind(padj_ind,paj_val)

total_n_select_feat = dim(padjusted)[1]
```

We have performed a hypothesis test by considering

$$H_0 = Feature\ is\ not\ significant$$

$$H_a = Feature\ is\ significant$$

```
rejected = pval[which(pval<0.05)]
```

To perform Benjamini-Hochberg method manually first we have calculated the p-values by using t.test() method. Number of feature rejected or marked significant by t.test method are 281.

After getting p-values we have perform p-adjust step by using formula

$$threshold = (i/m) * \alpha$$

(i = each p-value rank, m="total number of tests", =false discovery rate: which is 0.05)

We have rejected all those features(hypothesis) for which p-value <= threshold. Th Rejected number of features or marked as significant features by Benjamini-Hochberg method are 39.

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

```
## [38,] "post"
## [39,] "presented"
```

Result Interpretation According to above results of BHM and normal test method we can see that normal test has marked 281 features whose pval is less than our threshold(0.05) while using BHM after pajdust it has marked only 39 features whose pval is less than the threshold.

Appendix

```
knitr::opts_chunk$set(echo = TRUE)
RNGversion('3.5.1')
library(readxl)
library(ggplot2)
library(mgcv)

library(pamr)
library(glmnet)
library(ggplot2)
library(kernlab)
library(kableExtra)

data = read_excel("Influenza.xlsx")

ggplot(data)+geom_line(aes(x = Time, y = Mortality, col = "Mortality"))+
  geom_line(aes(x = Time, y = Influenza, col= "Influenza"))+ylab("Series")+ggtitle("")

gammodel = gam(formula =  Mortality~Year+s(Week,k=length(unique(data$Week))), data =  data ,method="GCV
b0 = gammodel$coefficients[1]
b1 = gammodel$coefficients[2]
plot(gammodel, residuals=T)
gampredict = predict(gammodel,data)
ggplot(data)+geom_line(aes(x= Time, y = gampredict, col = "Predicted Mortality"))+
geom_line(aes(x = Time, y = data$Mortality, col = "Observed Mortality"))+ylab("Series")+ggtitle("Spline

summary(gammodel)
plot(gammodel, residuals=T)
penalty = c(0.001, 0.5 , 1, 5, 10, 100)

degF  = 60

pendata <- sapply(penalty, function(i){
  gammodels = gam(formula =  Mortality~Year+s(Week, sp = i), data =  data)

  predResult = predict(gammodels,data)

  df =  sum(influence(gammodels))

  data[paste0("Penalty_",i)] <<- predResult

  list(penalty = i ,deviance = gammodels$deviance, degFreedom = df )

})
```

```
 newdata = data.frame(Penalty = unlist(pendata[1,]) ,
deviance = unlist(pendata[2,]),
degreeFreedom = unlist(pendata[3,]))


  gammodels.pen.low = gam(formula =  Mortality~Year+s(Week, sp = 0.001), data =  data)

  summary(gammodels.pen.low)
ggplot(data) +
geom_line(aes(x = Time, y = Mortality, col = "Original")) +
geom_point(aes(x = Time, y = Mortality, col = "Original")) +
geom_line(aes(x = Time, y = Penalty_0.001, colour = "Penalty 0.001")) +
  geom_point(aes(x = Time, y = Penalty_0.001, colour = "Penalty 0.001")) +
geom_line(aes(x = Time, y = Penalty_10, colour = "Penalty 10")) +
  geom_point(aes(x = Time, y = Penalty_10, colour = "Penalty 10")) +
geom_line(aes(x = Time, y = Penalty_100, colour = "Penalty 100"))

plot(newdata$deviance,penalty, type = "l", xlab = "Deviance", main = "Deviance Vs Penalty")

plot(newdata$degreeFreedom,penalty, type = "l", xlab = "Degree of Freedom", main = "Defree of freedom Vs

dfplot = cbind(data, "Residual" =  residuals(gammodel))

ggplot(dfplot) + geom_line(aes(x = Time, y = Residual , col = "Residuals"))+
  geom_line(aes(x = Time, y = Influenza, col = "Influenza"))+ylab("Correlation")+xlab("Time")+ggtitle("
gngammodel = gam(formula =  Mortality~s(Year, k=length(unique(data$Year)))
                  +s(Week, k=length(unique(data$Week)))
                  +s(Influenza, k=length(unique(data$Influenza)))
                  , data =  data)

summary(gngammodel)

gnpredict = predict(gngammodel,data)


ggplot(data) + geom_line(aes(x = Time, y = gnpredict , col = "Fitted Mortality"))+
  geom_line(aes(x = Time, y = data$Mortality, col = "Original Mortality"))+ylab("Mortality")+xlab("Time

data2 = read.csv2("data.csv",sep = ";",header = TRUE)
set.seed(12345)
n = dim(data2)[1]
id=sample(1:n, floor(n*0.7))
train=data2[id,]
test=data2[-id,]

xtestdata=t(test[,-4703])
ytestdata=as.factor(test[,4703]) #Conference column data

xdata=t(train[,-4703])
ydata= train[[4703]] #Conference column data

shdata = list(x=xdata, y = ydata, geneid=as.character(1:nrow(xdata)),
                  genenames=rownames(xdata))
```

```
shtrainmodel = pamr.train(shdata)
shtrainmodel
shcv = pamr.cv(shtrainmodel,shdata)


pamr.plotcv(shcv)
thresh = shcv$threshold[which.min(shcv$error)]
pamr.plotcen(shtrainmodel ,shdata, threshold=thresh)
features = pamr.listgenes(shtrainmodel,shdata,threshold=thresh,genenames=TRUE)
sfeatures = nrow(features)

kable(col.names = c("id","names","0-score1","1-score"),features[1:10,], align = 'c')

presult = pamr.predict(shtrainmodel, xtestdata,threshold = thresh, type = "class")
centerr = mean(presult != ytestdata)

testdata = ytestdata
prediction = presult
table(prediction, testdata)
set.seed(12345)

trainx = as.matrix(train[, -4703])
trainy = as.matrix(train[,4703])
testx = as.matrix(test[, -4703])
testy = as.matrix(test[,4703])


glmmodel = glmnet(x = trainx ,y = trainy, family = "binomial",alpha = 0.5)

cvglmmodel = cv.glmnet(x = trainx ,y = trainy,alpha = 0.5,family = "binomial")

minLambda = cvglmmodel$lambda.min

glmmodelthresh = glmnet(x = trainx ,y = trainy, family = "binomial",alpha = 0.5, lambda = minLambda)


plot(glmmodel)
plot(cvglmmodel)
coefficients = coef(glmmodelthresh, s = cvglmmodel$lambda.min)

#selected_features = length(coefficients@Dimnames[1][coefficients@i + 1])

coeff = as.matrix(coefficients)

s_features = names(which((coeff <0 | coeff>0),coeff)[-1,1])

kable(col.names  = "selected features",s_features)

n_s_features = length(s_features)+1


glmPredict = predict(glmmodelthresh,testx, type = "class" )
```

```
elasticnet_err = mean(glmPredict != testy)

svmModel = ksvm(as.matrix(trainx),as.factor(trainy),kernel="vanilladot",scaled = FALSE)
svmModel
selectFeat =coef(svmModel)
selectedFeat = length(selectFeat[[1]])

 prediction = predict(svmModel,testx)
 svmerrorRate = mean(prediction != testy)
 table(prediction,testy)

kable(data.frame(Model_Type = c("Centriod","ElasticNet","SVM"),Features = c(sfeatures,n_s_features,sele

pval <- sapply(1:(ncol(data2)-1), function(k){

   t.test(data2[,k]~Conference,data2, alternative = "two.sided")$p.value
})

ind_df = data.frame(index = 1:length(pval), pavlues = pval)

orderPval = ind_df[order(ind_df$pavlues),]

padj_ind = list()
paj_val = list()

tes = sapply(1:length(orderPval$pavlues), function(i){

  thresh = (i/length(orderPval$pavlues))*0.05

  if(orderPval$pavlues[i] <= thresh){
  paj_val <<- append(paj_val, orderPval$pavlues[i])
  padj_ind <<- append(padj_ind,orderPval$index[i])
  }
})

padjusted = cbind(padj_ind,paj_val)

total_n_select_feat = dim(padjusted)[1]

rejected = pval[which(pval<0.05)]
selected_features_names = names(data2[unlist(padjusted[1:dim(padjusted)[1],1])])
as.matrix(selected_features_names)
``
```