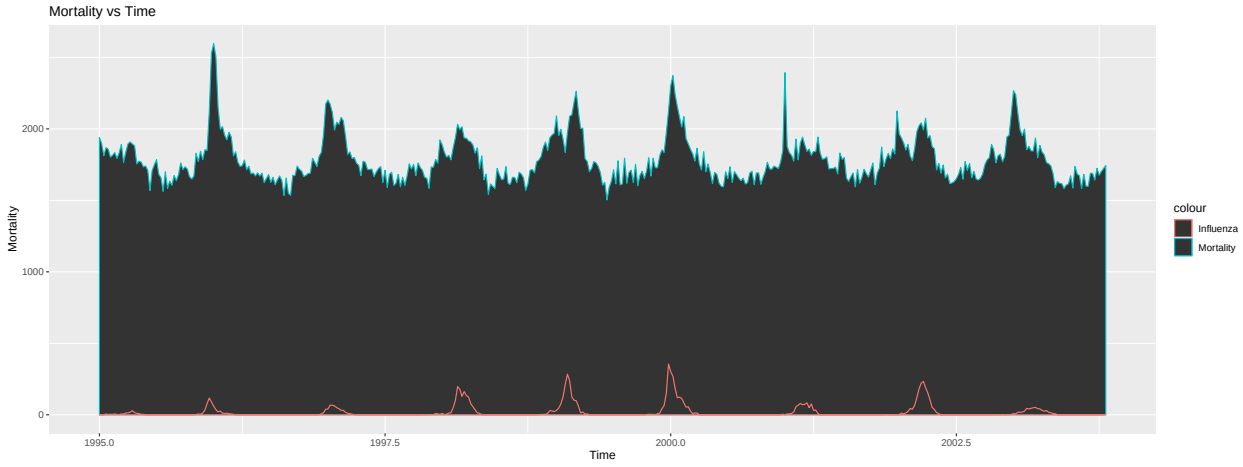
Lab 2 Block 2 Report

Yash Pawar

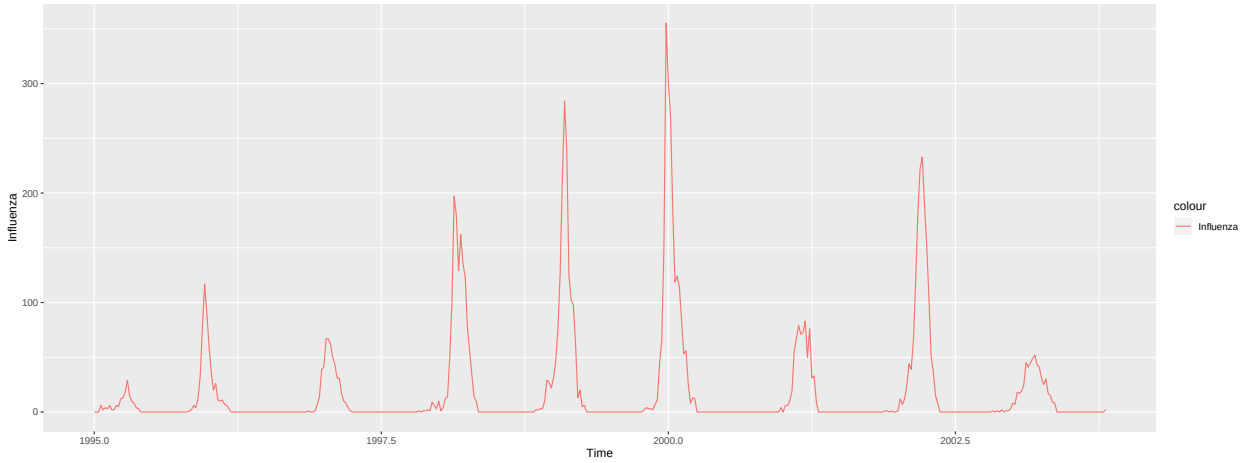
08/12/2019

Assignment 1.1

```
## [1] 1503 2597
```




```
## List of 1
##  $ plot.title:List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0.5
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi FALSE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  - attr(*, "class")= chr [1:2] "theme" "gg"
##  - attr(*, "complete")= logi FALSE
##  - attr(*, "validate")= logi TRUE
```



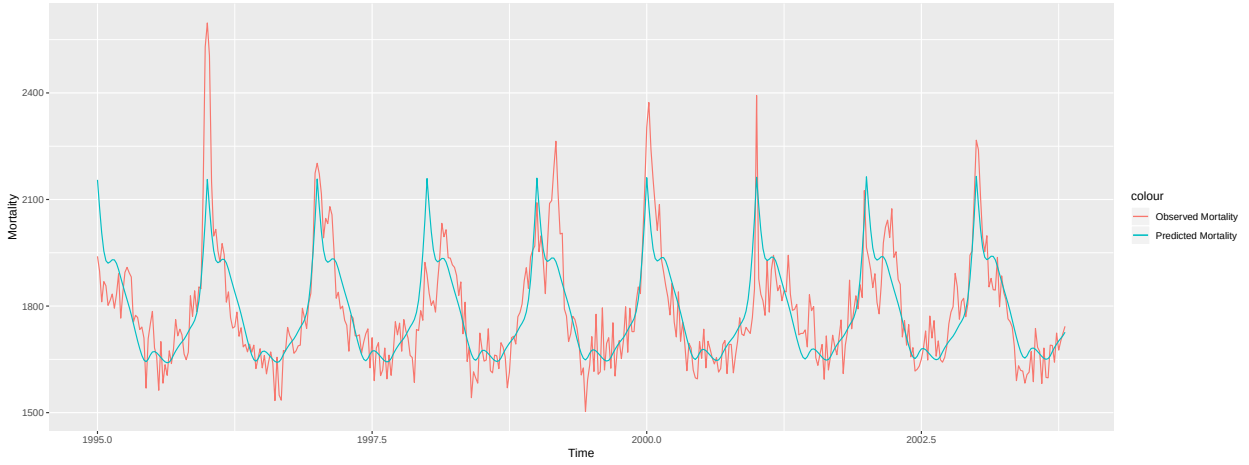
It can be seen that the mortality peaks during Winter time every year, which happens when the influenza cases are high.

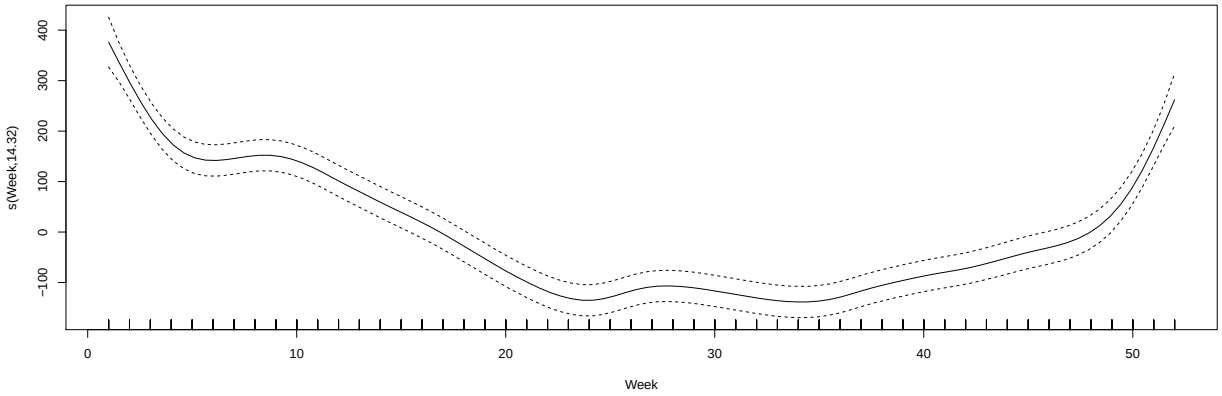
Assignment 1.2

```
## [1] 3718012
```

The probablistic model: Mortality ~ N(Year + s(Week), σ^2)

Assignment 1.3



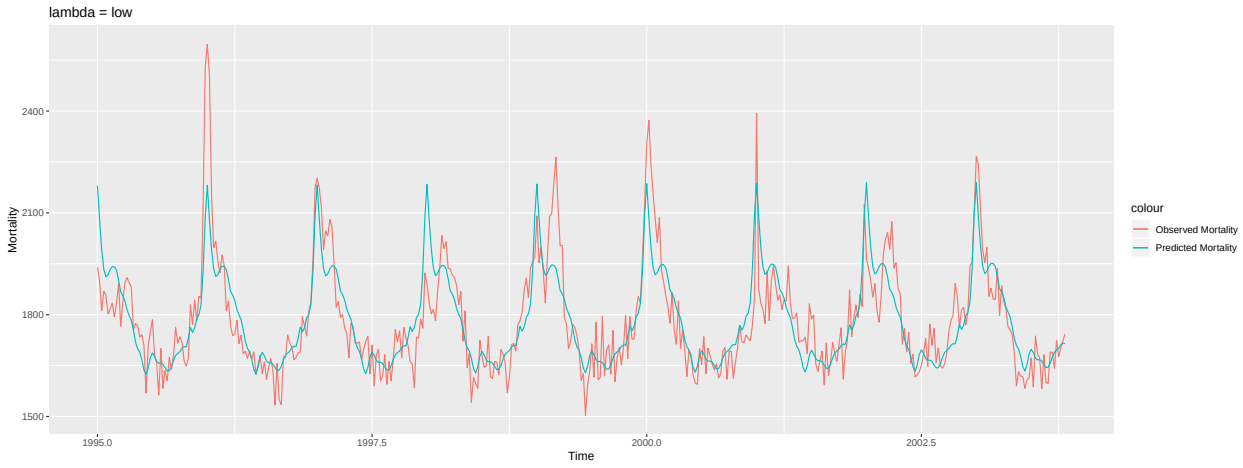


The fit of the predictions is close to the actual data. However, the predictions are smoother as compared to the actual mortality. The predictions of peaks are not accurate as per the original data. The gam model suggests that the s(week) appears to be significant with the significance level of 0.1%. The mortality attains its peak during the winter season, which from the earlier inference suggests that the highest number of cases of Influenza occur during the winter which leads to more number of mortalities. The mortality is significantly low during the Summer season.

There is no change in trend of mortality from one year to another.

The plot gives us the "weeks" vs "spline of weeks" which tells us how the model has been formed over the course of weeks. We get splines which are close to the mean predicitons.

Assignment 1.4




```
## List of 1
##  $ plot.title:List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0.5
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi FALSE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  - attr(*, "class")= chr [1:2] "theme" "gg"
##  - attr(*, "complete")= logi FALSE
##  - attr(*, "validate")= logi TRUE
```


```
## List of 1
##  $ plot.title:List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
```

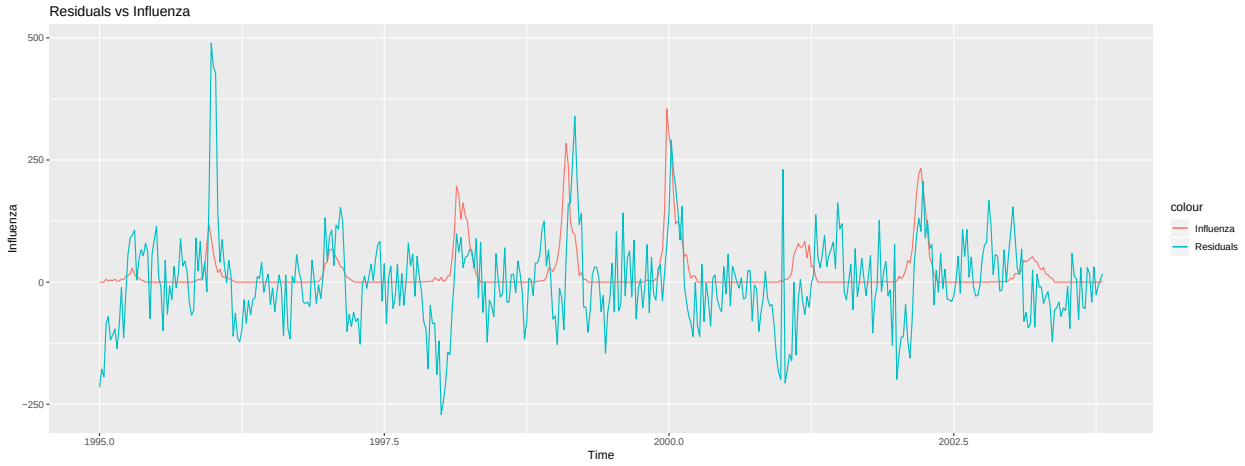
```
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0.5
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi FALSE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  - attr(*, "class")= chr [1:2] "theme" "gg"
##  - attr(*, "complete")= logi FALSE
##  - attr(*, "validate")= logi TRUE
```

The predictions of the mortality are the most accurate with lambda of order (10^-7). Thus, we can see that the deviance is the lowest for the best penalty factor(lambda). As the lambda increases the predicitons become inaccurate.

It was also observed that with increasing penalty factor, the degress of freedom decreases. This can be confirmed from the fact that the lower degrees of freedom leads to underfitting of model

** Higher λ leading to higher penalizing of coefficients leading to reduced degree of freedom.

Assignment 1.5




```
## List of 1
##  $ plot.title:List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0.5
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi FALSE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  - attr(*, "class")= chr [1:2] "theme" "gg"
```

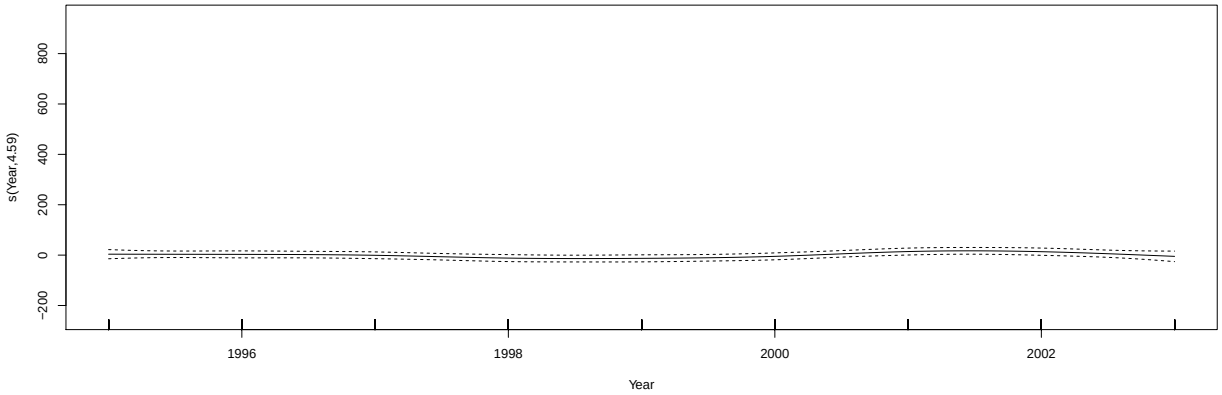
```
##  - attr(*, "complete")= logi FALSE
##  - attr(*, "validate")= logi TRUE
```

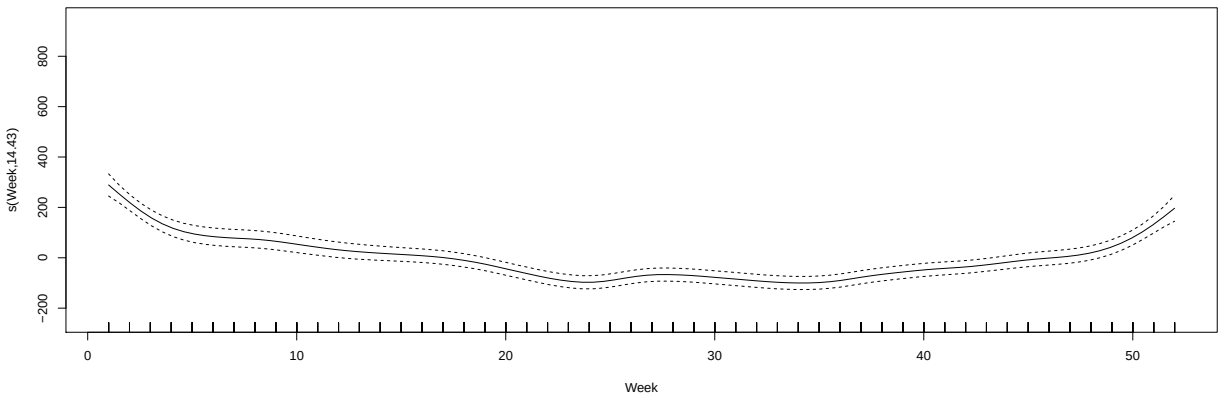
```
## [1] -270.9930  488.9808
```

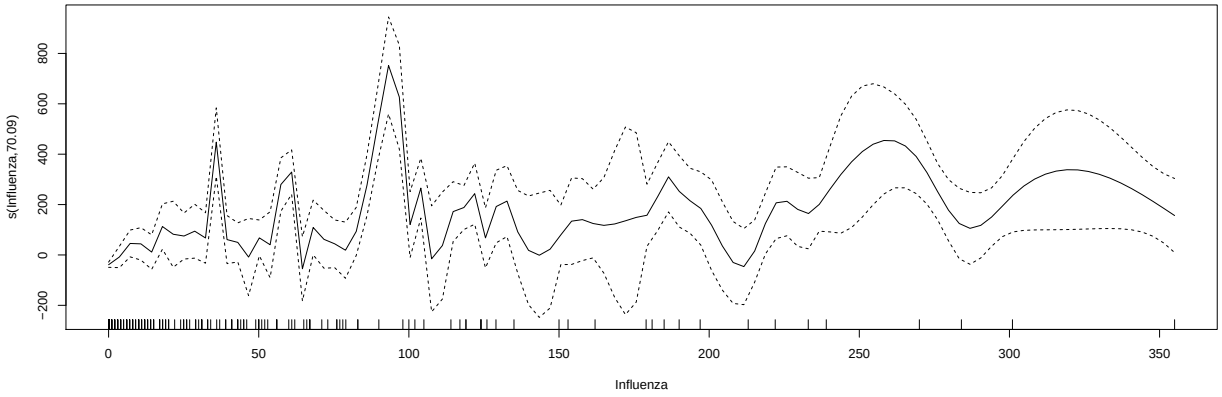
The outbreaks in case of influenza coressponds to higher mortality rates, however the GAM models could not efficiently predict the peaks which leads to higher residuals at peak values.

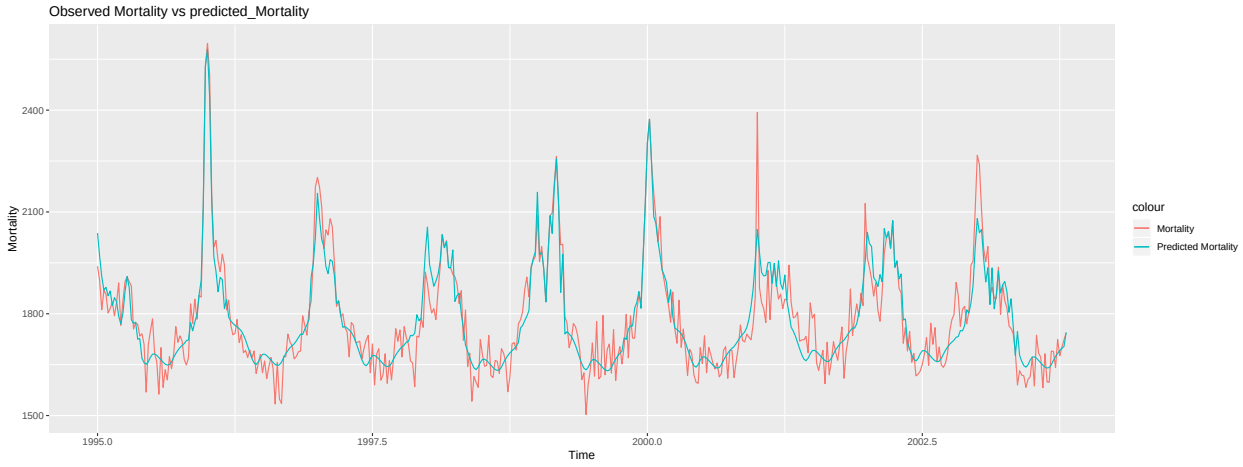
The correlation in temporal patterns of residuals is highly corelated to outbreaks in Influenza.

Assignment 1.6










```
## List of 1
##  $ plot.title:List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0.5
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi FALSE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  - attr(*, "class")= chr [1:2] "theme" "gg"
##  - attr(*, "complete")= logi FALSE
##  - attr(*, "validate")= logi TRUE
```

The deviance is:

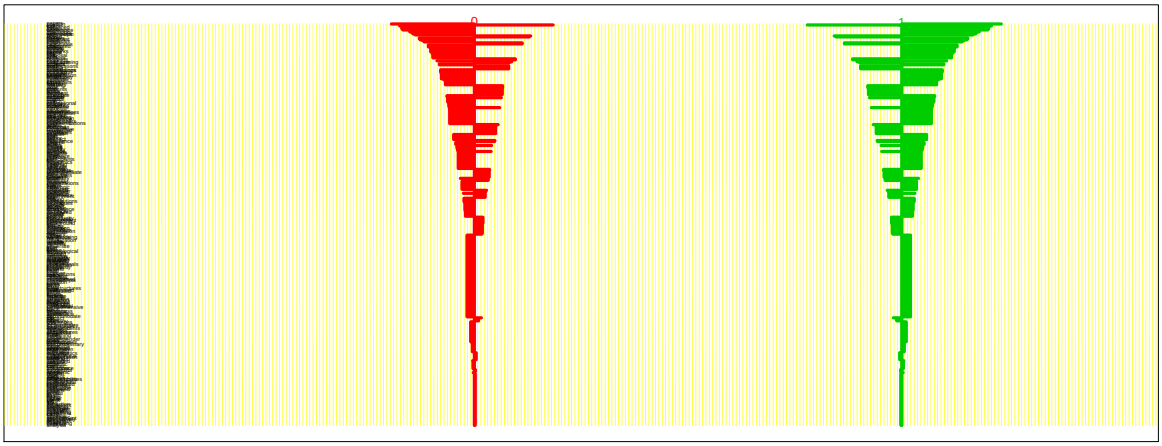
```
gam_additive$deviance
```

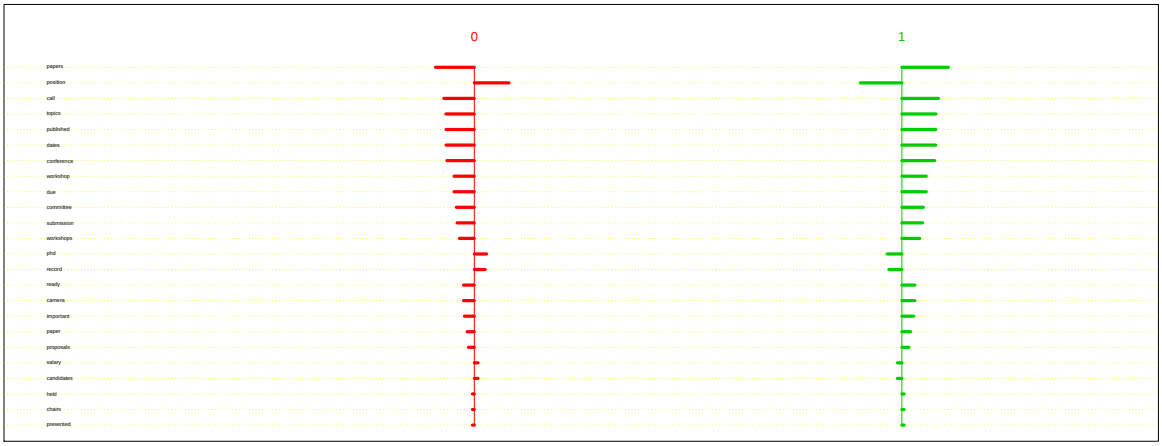
```
## [1] 1731415
```

The model seems to perform better as the deviance observed in this case is much lower compared to the previous model.

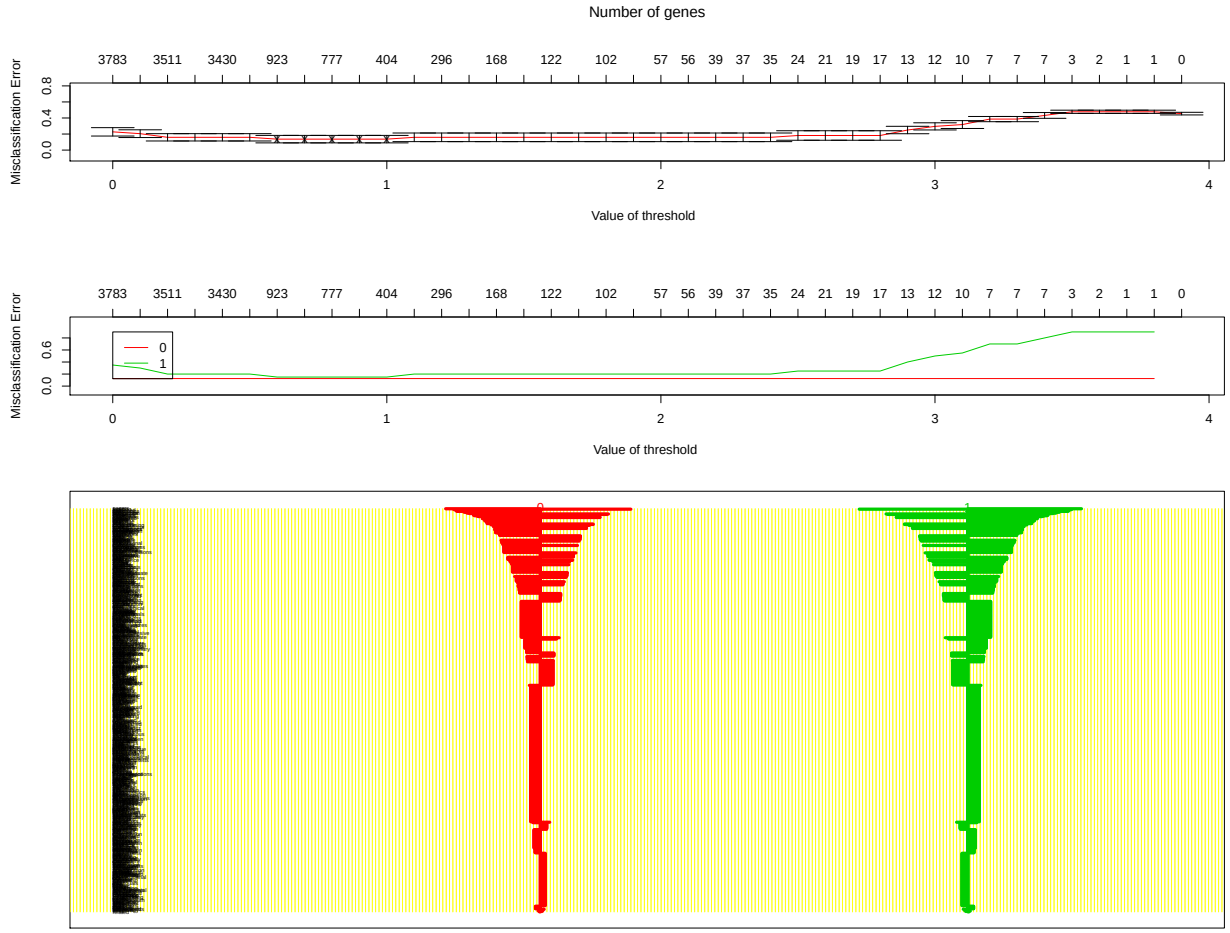
Assignment 2.1

```
## 1234567891011121314151617181920212223242526272829303132333435363738394041
```






```
## 12Fold 1 :12345678910111213141516171819202122232425262728293031323334353637383940
## Fold 2 :12345678910111213141516171819202122232425262728293031323334353637383940
## Fold 3 :12345678910111213141516171819202122232425262728293031323334353637383940
## Fold 4 :12345678910111213141516171819202122232425262728293031323334353637383940
## Fold 5 :12345678910111213141516171819202122232425262728293031323334353637383940
## Fold 6 :12345678910111213141516171819202122232425262728293031323334353637383940
## Fold 7 :12345678910111213141516171819202122232425262728293031323334353637383940
## Fold 8 :12345678910111213141516171819202122232425262728293031323334353637383940
## Fold 9 :12345678910111213141516171819202122232425262728293031323334353637383940
## Fold 10 :12345678910111213141516171819202122232425262728293031323334353637383940
```




```
## 1
```

```
## [1] 923
```

The centroid plot suggests that that the shrinkage of coefficients varies with different thresholds, The shrinkage for threshold 1.3 is the best as it gives the least misclassification rate. The total number of features selected by this method are 231.

The mis classification rate for the test_data is 0.1

Assignment 2.2

As we can see, the misclassification rate for NSC is 0.1 which is greater than the error rate of SVM model.

Result Comparison of NSC and, Elastic net and SVM:

```
##   models misclass_comparison coef_comparison
## 1      1                0.05            923
## 2      2                0.05             11
## 3      3                0.15             43
```

From the table we can infer that the SVM model has the lowest error rate with 43 variable selections which makes it the best model as opposed to the other models as they have higher eroor rates. The variables selected in case of elastic net model are the least with highst error rate.

Assignment 2.3

The features corresponding to the rejected hypothesis are:

```
##                       p_val adjusted_p_val
## papers        1.116910e-10   5.251710e-07
## submission    7.949969e-10   1.869038e-06
## position      8.219362e-09   1.288248e-05
## published     1.835157e-07   2.157227e-04
## important     3.040833e-07   2.859600e-04
## call          3.983540e-07   3.121767e-04
## conference    5.091970e-07   3.420349e-04
## candidates    8.612259e-07   5.061856e-04
## dates         1.398619e-06   6.576305e-04
## paper         1.398619e-06   6.576305e-04
## topics        5.068373e-06   2.166499e-03
## limited       7.907976e-06   3.098609e-03
## candidate     1.190607e-05   4.306335e-03
## authors       2.154461e-05   6.331422e-03
## camera        2.099119e-05   6.331422e-03
## ready         2.099119e-05   6.331422e-03
## phd           3.382671e-05   9.140486e-03
## projects      3.499123e-05   9.140486e-03
## org           3.742010e-05   9.260491e-03
## chairs        5.860175e-05   1.377727e-02
## due           6.488781e-05   1.386829e-02
## original      6.488781e-05   1.386829e-02
## notification  6.882210e-05   1.406963e-02
## salary        7.971981e-05   1.561844e-02
## record        9.090038e-05   1.643898e-02
## skills        9.090038e-05   1.643898e-02
## held          1.529174e-04   2.663028e-02
## team          1.757570e-04   2.951462e-02
## apply         2.166414e-04   3.084087e-02
## committee     2.117020e-04   3.084087e-02
## international 2.295684e-04   3.084087e-02
## pages         2.007353e-04   3.084087e-02
## proceedings   2.117020e-04   3.084087e-02
## strong        2.246309e-04   3.084087e-02
## workshop      2.007353e-04   3.084087e-02
## degree        3.762328e-04   4.539416e-02
## excellent     3.762328e-04   4.539416e-02
## post          3.762328e-04   4.539416e-02
## presented     3.765147e-04   4.539416e-02
```

The number of selected features are 39. The result suggests that out of all the features only 39 features are important. which corresponds to 0.8% of total features. It can be concluded that these features have higher corelation to the target as compared to the other features.

Appendix:

```
knitr::opts_chunk$set(echo = TRUE,fig.width=16, fig.height=6)
library(readxl)
library(ggplot2)
Influenza <- read_excel("Influenza.xlsx")
```

```r
#View(Influenza)
range(Influenza$Mortality)
ggplot(data = Influenza) +
  geom_area(aes(x = Time, y = Mortality, color = "Mortality")) +
  geom_area(aes(x = Time, y = Influenza, color = "Influenza")) +
  xlab("Time") + ylab("Mortality") + ggtitle("Mortality vs Time")
  theme(plot.title = element_text(hjust = 0.5))


  ggplot(data = Influenza) +
    geom_line(aes(x = Time, y = Influenza, color = "Influenza"))

library(mgcv)
gam_flu = gam(Mortality~Year + s(Week, k = length(unique(Influenza$Week))), family = gaussian(), data =
#gam.check(gam_flu)
#summary(gam_flu)
#plot.gam(gam_flu)
gam_flu$deviance
flu_predict = predict.gam(gam_flu)
mortality_pred_data = cbind.data.frame("Time" = Influenza$Time, "predicted_mortality" = flu_predict, "M
ggplot(data = mortality_pred_data) +
  geom_line(aes(x = Time, y = Mortality, color = "Observed Mortality")) +
  geom_line(aes(x = Time, y = predicted_mortality, color = "Predicted Mortality"))

plot.gam(gam_flu)
#gam_flu$edf


penalty_spline = gam(Mortality~Year + s(Week, k = length(unique(Influenza$Week)), sp = 1e-7), family = g
#best_lambda = penalty_spline$lambda
gam_with_lambda = predict(penalty_spline)
high_penalty_spline = gam(Mortality~Year + s(Week, k = length(unique(Influenza$Week)), sp = 10), family
gam_with_high_lambda = predict(high_penalty_spline)
plot_data_with_low_pen = cbind.data.frame("Time" = Influenza$Time, "predicted_mortality" = gam_with_lamb

ggplot(data = plot_data_with_low_pen) +
  geom_line(aes(x = Time, y = Mortality, color = "Observed Mortality")) +
  geom_line(aes(x = Time, y = predicted_mortality, color = "Predicted Mortality")) +
  ggtitle("lambda = low")
theme(plot.title = element_text(hjust = 0.5))

plot_data_high_pen = cbind.data.frame("Time" = Influenza$Time, "predicted_mortality" = gam_with_high_lam

ggplot(data = plot_data_high_pen) +
  geom_line(aes(x = Time, y = Mortality, color = "Observed Mortality")) +
  geom_line(aes(x = Time, y = predicted_mortality, color = "Predicted Mortality")) +
  ggtitle("lambda = High")
theme(plot.title = element_text(hjust = 0.5))

resid_data = cbind.data.frame("Time" = Influenza$Time, "Influenza" = Influenza$Influenza, "Residuals" =

ggplot(data = resid_data) +
  geom_line(aes(x = Time, y = Influenza, color = "Influenza")) +
```

```
  geom_line(aes(x = Time, y = Residuals, color = "Residuals")) +
  ggtitle("Residuals vs Influenza")
theme(plot.title = element_text(hjust = 0.5))
range(gam_flu$residuals)

gam_additive = gam(Mortality ~ s(Year, k = length(unique(Influenza$Year)))
                   + s(Week, k = length(unique(Influenza$Week)))
                   + s(Influenza, k = length(unique(Influenza$Influenza))),
                     family = gaussian(), data = Influenza)

pred_gam_additive = predict.gam(gam_additive)
plot.gam(gam_additive)

plot_data_additive_gam = cbind.data.frame("Time" = Influenza$Time, "Mortality" = Influenza$Mortality, "

ggplot(data = plot_data_additive_gam) +
  geom_line(aes(x = Time, y = Mortality, color = "Mortality")) +
  geom_line(aes(x = Time, y = Predicted_Mortality, color = "Predicted Mortality")) +
  ggtitle("Observed Mortality vs predicted_Mortality")
theme(plot.title = element_text(hjust = 0.5))
#summary(gam_additive)

gam_additive$deviance
library(readr)
library(pamr)
data_email <- read_csv2("data.csv")
data_email = as.data.frame(data_email)
#View(data)
set.seed(12345)
n = dim(data_email)[1]

id_1 = sample(1:n, floor(n*0.7))
train_data_email = data_email[id_1,]
test_data_email = data_email[-id_1,]
#train_data_email = t(train_data_email)
rownames(train_data_email)=1:nrow(train_data_email)
x=t(train_data_email[,-4703])
y=train_data_email[[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))
pamr.plotcen(model, mydata, threshold=1.0)
pamr.plotcen(model, mydata, threshold=2.5)



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

best_threshold = cvmodel$threshold[which.min(cvmodel$error)]
pamr.plotcen(model, mydata, threshold = best_threshold)

model_best = pamr.train(mydata, threshold = best_threshold)
```

```
#a=pamr.listgenes(model,mydata,threshold=1.3)
#cat( paste( colnames(data)[as.numeric(a[,1])], collapse='\n' ) )
model_best$nonzero

x1=t(test_data_email[,-4703])
y1=test_data_email[[4703]]
mydata_test=list(x1=x1,y1=as.factor(y1),geneID=as.character(1:nrow(x1)), genenames=rownames(x1))
predict_model = pamr.predict(model, mydata_test$x1, threshold = 1.3)

conf_matrix = table(predict_model, mydata_test$y1)
misclassification_rate_email = 1 - sum(diag(conf_matrix))/sum(conf_matrix)

## Elastic net

library(glmnet)
elastic_net_model_selection = cv.glmnet( as.matrix(train_data_email[,-4703]),as.matrix(train_data_email
                              alpha = 0.5, family = "binomial")
#plot(elastic_net_model)

selected_penalty = elastic_net_model_selection$lambda.1se

elastic_net_model = glmnet( as.matrix(train_data_email[,-4703]),as.matrix(train_data_email[,4703]),
                            family = "binomial", lambda = selected_penalty, alpha = 0.5)
#min(elastic_net_model$cvm)
coefficient_selection = coef(elastic_net_model, s = "lambda.1se")


number_coef = length(which(coefficient_selection != 0))

#plot(elastic_net_model, xvar = "lambda", label = TRUE)

predict_elastic_net = predict(elastic_net_model, as.matrix(test_data_email[,-4703]), type = "class")
conf_matrix_elastic_net = table(predict_elastic_net, as.matrix(test_data_email[,4703]))
misclass_elastic_net = 1 - sum(diag(conf_matrix_elastic_net))/sum(conf_matrix_elastic_net)

library(kernlab)
K <- as.kernelMatrix(crossprod(t(train_data_email[,-4703])))
svm_model = ksvm(K, y, kernel = "vanilladot")

Ktest = as.kernelMatrix(crossprod(t(test_data_email[,-4703]),t(train_data_email[SVindex(svm_model),-470
svm_prediction = predict(svm_model, Ktest, type = "response")
#svm_model@coef
svm_prediction = ifelse(svm_prediction>0.5,1,0)
conf_matrix_svm = table(svm_prediction, as.matrix(test_data_email[,4703]))

misclass_rate_svm = 1 - sum(diag(conf_matrix_svm))/sum(conf_matrix_svm)

misclass_comparison = c(misclassification_rate_email, misclass_elastic_net, misclass_rate_svm)
coef_comparison = c(model_best$nonzero, number_coef, length(svm_model@coef))
table_comparison = cbind.data.frame(models = seq(1,3,1) ,misclass_comparison, coef_comparison)
table_comparison
res = lapply(data_email[,-4703], function(x)
                        t.test(x~data_email[[4703]], data = data_email, alternative = c("two.sided"),
```

```
                                var.equal = FALSE))


# Getting the P- Values
p_val = sapply(X = res, FUN = getElement, name = "p.value")

# Adjusting the P-Values

adjusted_p_val = p.adjust(p_val, method = "BH")

# Sorting in ascending order

sort_data = cbind.data.frame(p_val, adjusted_p_val)
sort_data = sort_data[order(sort_data$adjusted_p_val),]
sort_data_BH = sort_data[sort_data$adjusted_p_val<0.05,]

p_val = p_val[order(p_val)]
adjusted_p_val = adjusted_p_val[order(adjusted_p_val)]


sort_data_BH
```