99-Lab2-Block2-Group Report

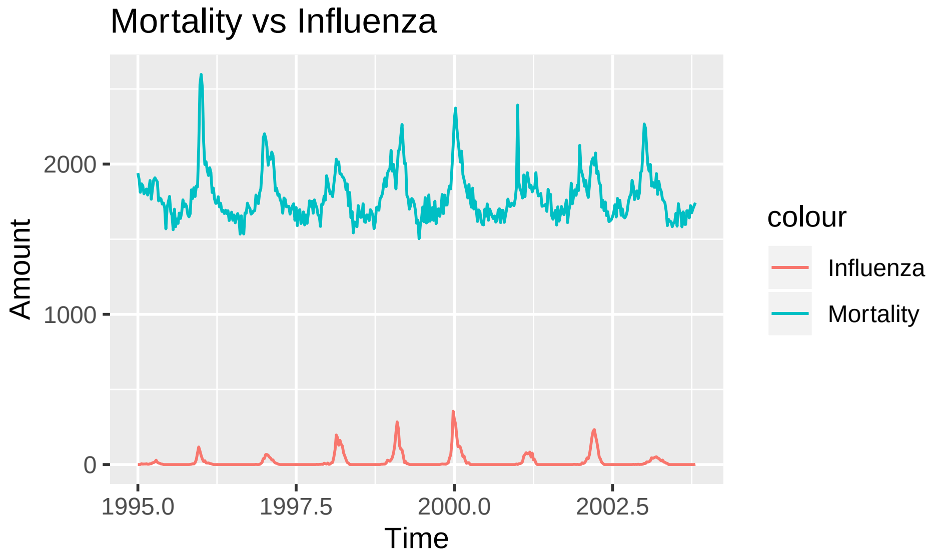
Weng Hang Wong Jooyoung Lee Fengjuan Chen

12/16/2019

Assignment 1–Weng Hang Wong(task1-4) Jooyoung Lee(task5-6)

1

From the graph belowed, we can see that when the Mortality cases increase, the Influenza cases will also increase at the same time, it seems to have relation to each other. However, we still not sure if the two variables are correlated to each other.



2.

The underlying probabilistic model: Y~EF(mu=year+s(week))

```
## GAM model
```

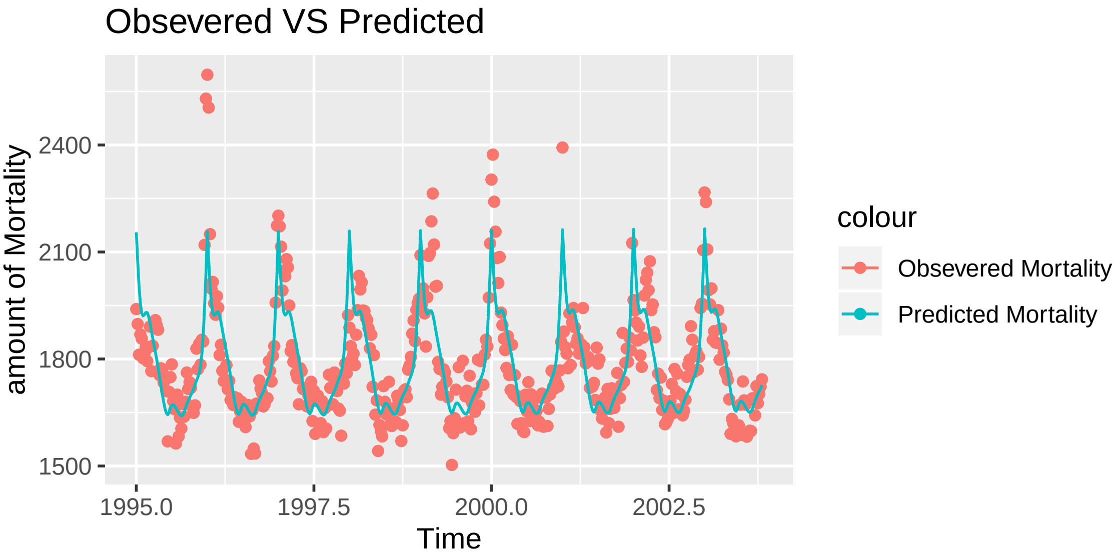
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Mortality ~ Year + s(Week, k = length(unique(data$Week)))
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -680.598   3367.760  -0.202    0.840
## Year           1.233      1.685   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
```

3.

The quality of the fitting model seems is not the best fit from the plot belowed. We can still spot many outliners from the below and a few on the top of the plot.

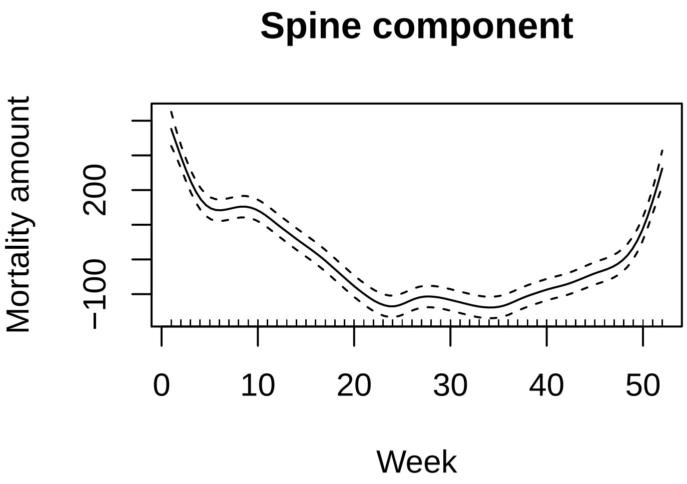
According to the ouput of GAM model, the most significant part is the spline function of Week*** , since the p-value of which is very small (<0.0001), which means the spine weeks have the most significant impact to the model than year.



```
##
## 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
```

```
## The p-value of GAM model:  0.000113193
```

According to the graph, the peak of Mortality which is at the beginning and at the end, when we know there are 52 weeks in a year, which means that it has low mortality in summer and high mortality in winter.

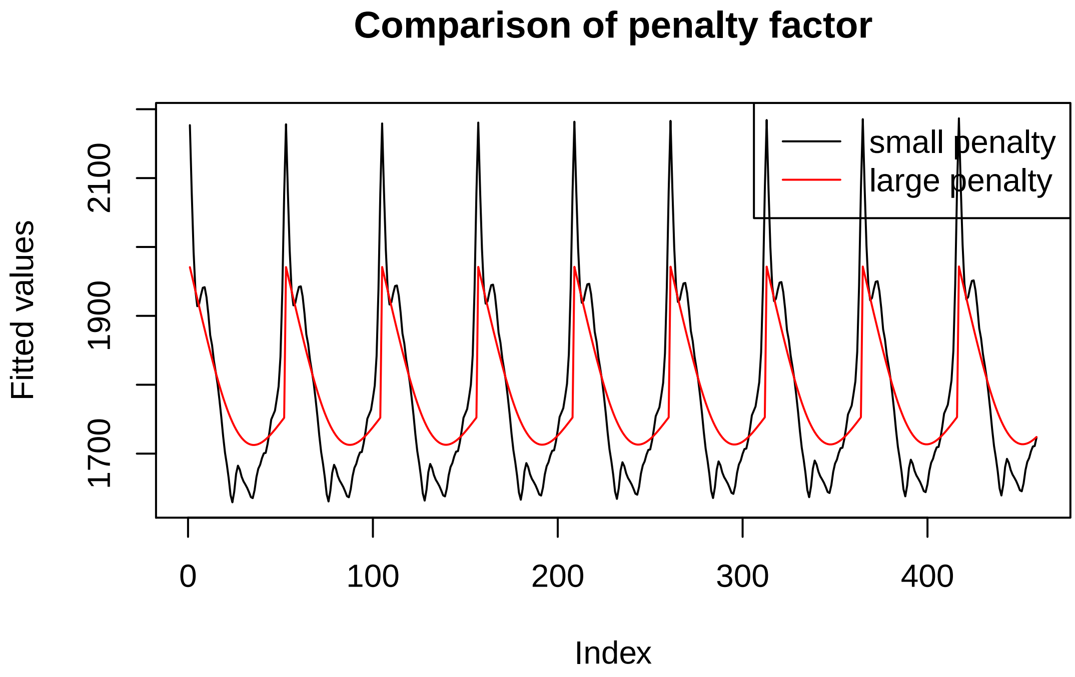


4.

We set the GAM model 1 with a very small penalty of 0.00001 and GAM model 2 with a large penalty of model 1. The plot reveals that when the penalty factor is larger, the fluctuation of model is smaller and it will be more likely get closed to a linear regression model.

The model 1 has lower deviance with smaller penalty, and model 2 has higher deviance with larger penalty. The deviance is getting higher along with the increase penalty.

The increasing degree of freedom always come with the decreasing penalty, and it is showed by the models, so, Yes, our results confirm this relationship.



```
## The deviance of model 1(low penalty):  3646841
```

```
## The deviance of model 2(high penalty):  6593860
```

```
## Lower penalty model with higher degree of freedom
```

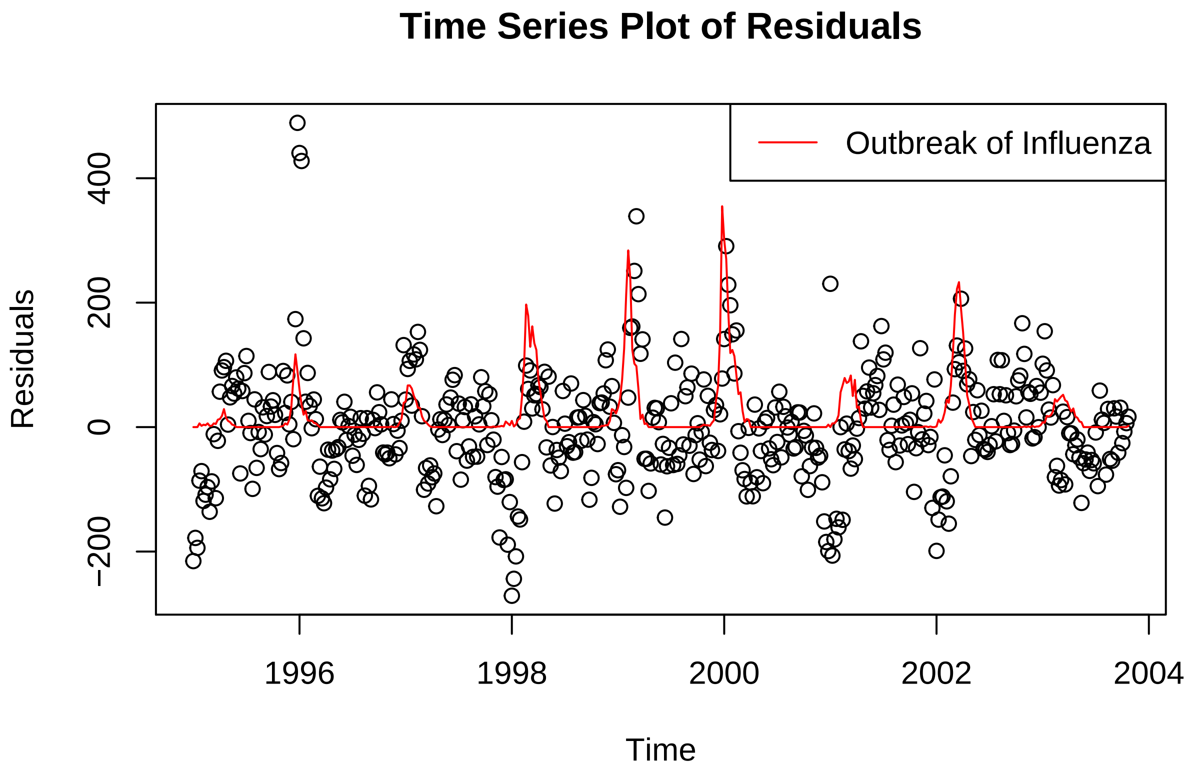
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Mortality ~ Year + s(Week, k = length(unique(data$Week)))
##
## Estimated degrees of freedom:
## 23.4  total = 25.37
##
## GCV score: 8902.083     rank: 36/53
```

```
## Higher penalty model with lower degree of freedom
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Mortality ~ Year + s(Week, k = length(unique(data$Week)))
##
```

```
## Estimated degrees of freedom:
## 1.49  total = 3.49
##
## GCV score: 14586.8
```

5.

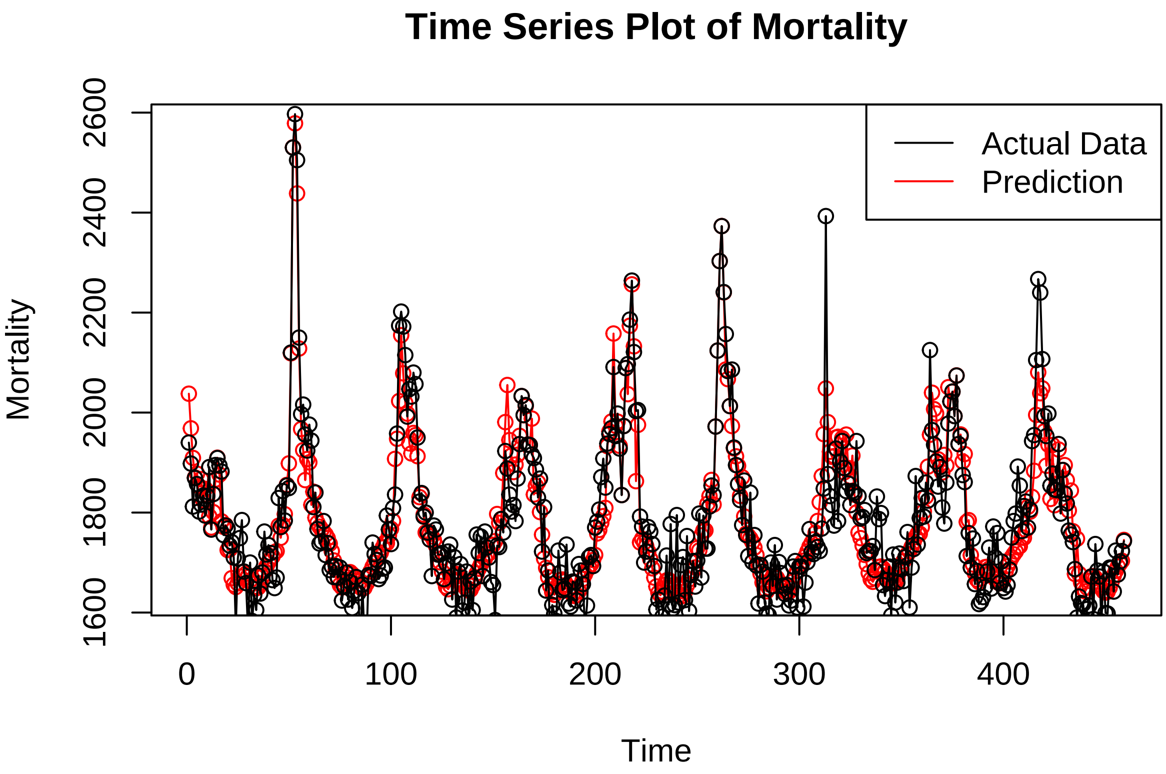


The plot above shows the residuals of mortality is correlated to the values of outbreak of influenza. However, such correlation is not strong, as the outbreak of influenza does not go along with negative residuals.

6.

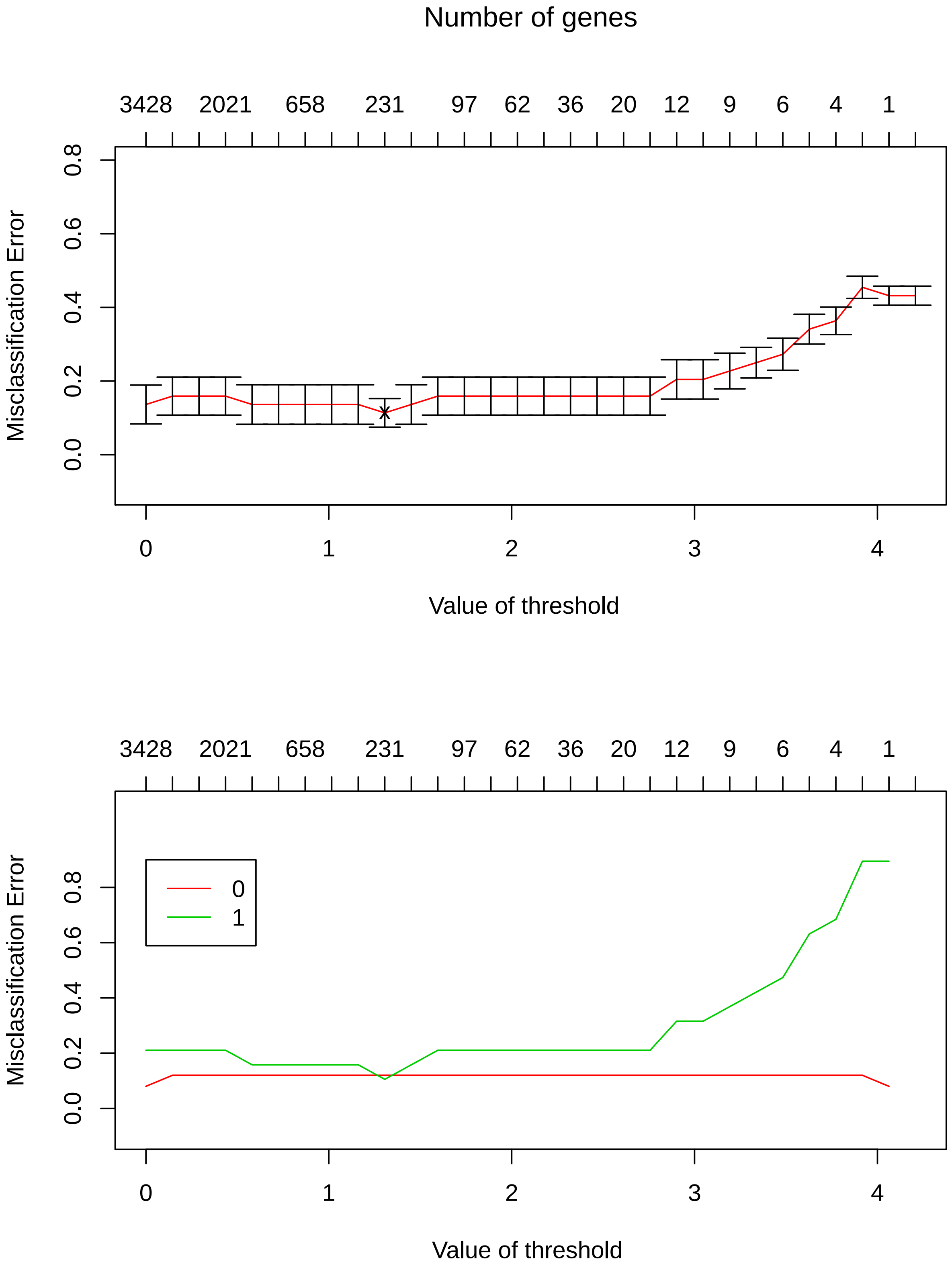
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Mortality ~ s(Year, k = length(unique(data$Year))) + s(Week,
##     k = length(unique(data$Week))) + s(Influenza, k = length(unique(data$Influenza)))
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1783.765      3.198   557.8   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

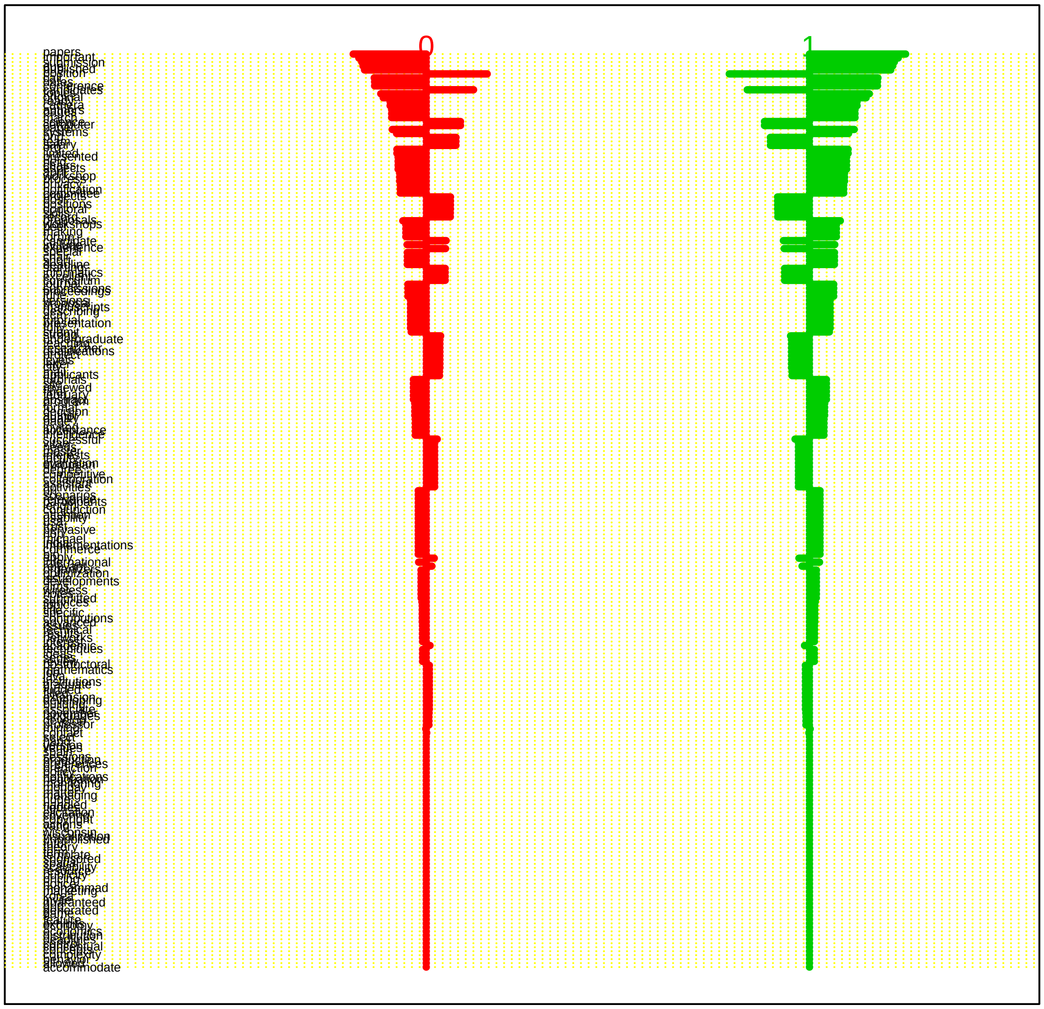
```
##
## Approximate significance of smooth terms:
##                 edf Ref.df      F p-value    
## s(Year)       4.587  5.592  1.500   0.178    
## s(Week)      14.431 17.990 18.763  <2e-16 ***
## s(Influenza) 70.094 72.998  5.622  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 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
```



Mortality is influenced by the outbreak of influenza, considering p-value as shown. This model better fits the data compare to the one previously presented; extreme prediction values disappeared and predicted values are mostly placed at where the real data points are.

1


Number of genes
3428
2021
658
231
97
62
36
20
12
9
6
4
1
Misclassification Error
0.0
0.2
0.4
0.6
0.8
0
1
2
3
4
Value of threshold
3428
2021
658
231
97
62
36
20
12
9
6
4
1
Misclassification Error
0.0
0.2
0.4
0.6
0.8
0
1
0
1
2
3
4
Value of threshold



The centroid plot tells us that some features are important in determining the classes of the observations. For example, the feature "paper" is on the top of the list in the plot, so it has the most important role in classification.

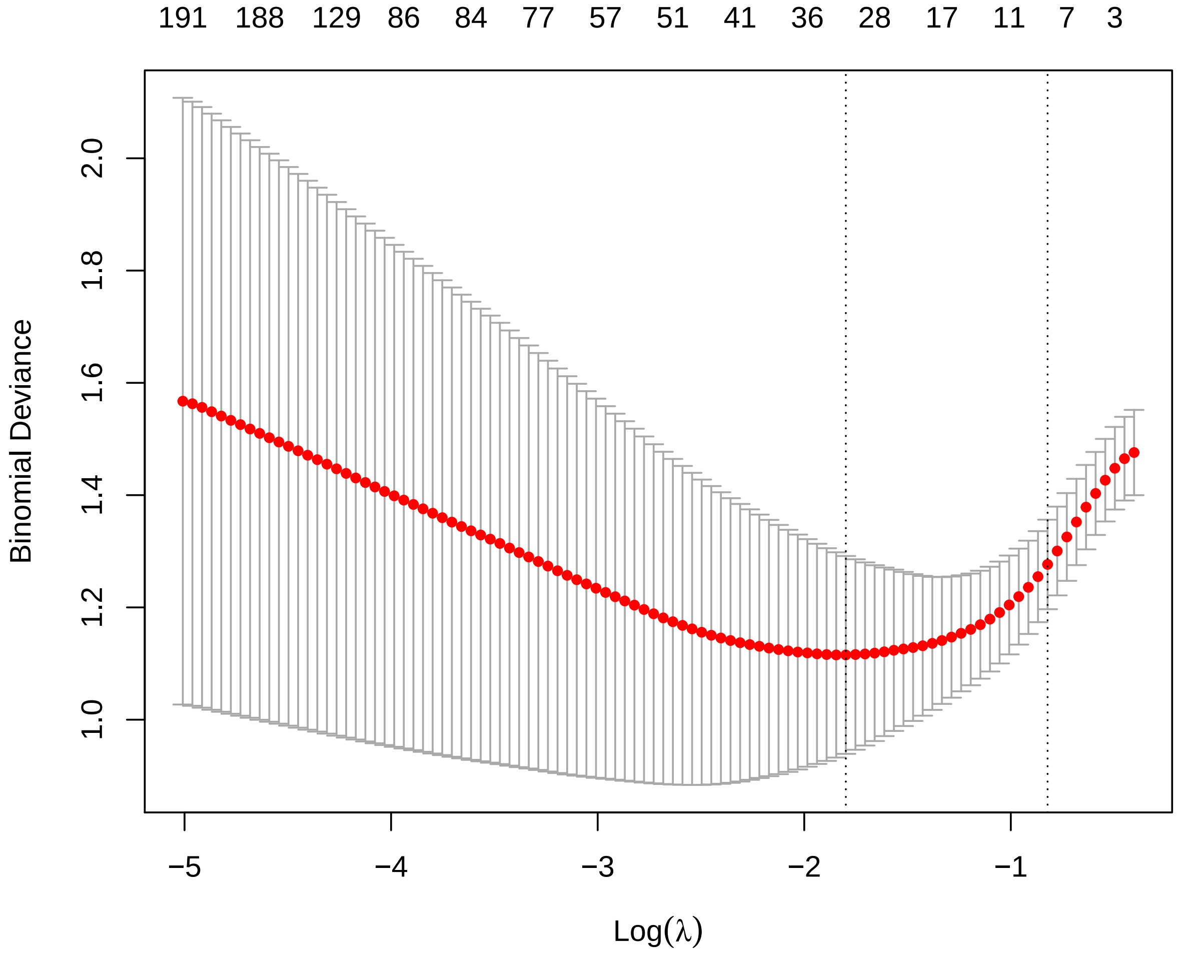
```
## 231 features were selected.
```

Table 1: The 10 most contributing features

Rank	Feature
1	papers
2	important
3	submission
4	due
5	published
6	position
7	call
8	conference
9	dates
10	candidates

```
## The test error is: 0.1
```

2



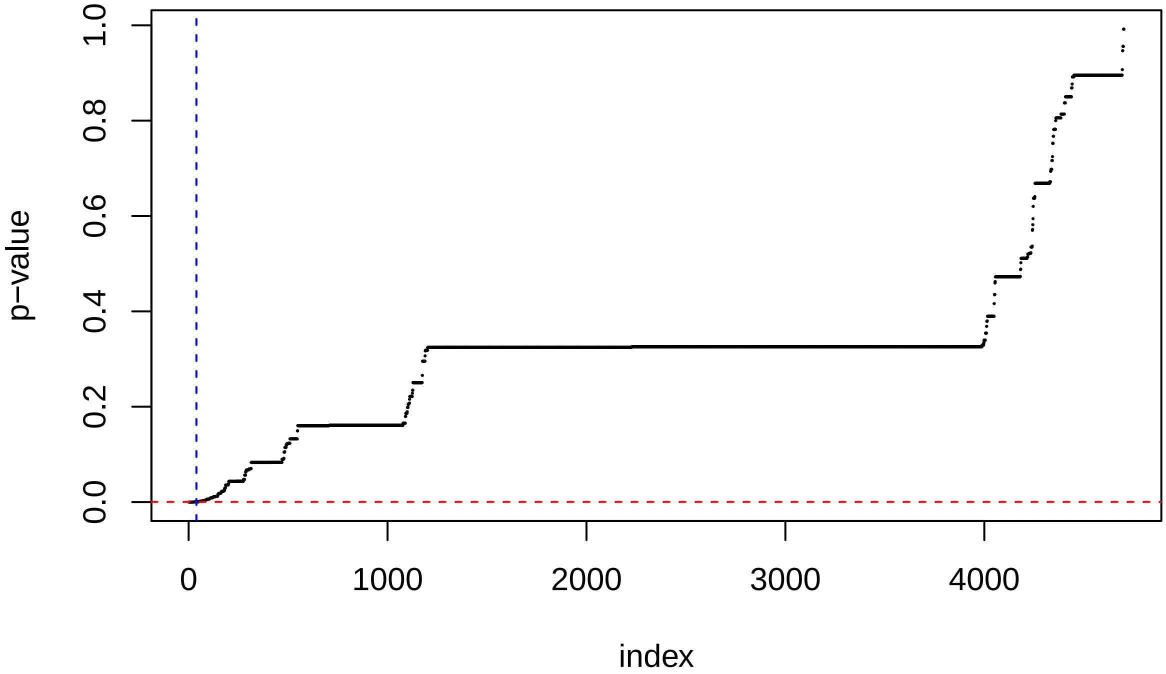
```
##  Setting default kernel parameters
```

Table 2: comparative table

Method	Test_error	Feature_number
NSC	0.10	231
Elasticnet	0.10	32
SVM	0.05	43

From the comaprative table, if we want more concise prediction, we choose SVM. If we want more easier interpretation, we choose Elasticnet.

3



We use $\alpha = 0.05$ to implement Benjamini-Hochberg method for the original data. Then there are 39 features corresponding to the rejected hypotheses. We use talbe 5 to show their names and p-values.

Table 3: Features correpsond to the rejected hypotheses

	feature	p_value	rank	imq
3036	papers	1.11690979691045e-10	1	0.0000106
4060	submission	7.94996892853816e-10	2	0.0000213
3187	position	8.2193623639657e-09	3	0.0000319
3364	published	1.83515727946722e-07	4	0.0000425
2049	important	3.04083345811897e-07	5	0.0000532
596	call	3.98353962926113e-07	6	0.0000638
869	conference	5.09196977301001e-07	7	0.0000744
607	candidates	8.61225948487058e-07	8	0.0000851
1045	dates	1.39861857386056e-06	9	0.0000957
3035	paper	1.39861857386056e-06	10	0.0001063
4282	topics	5.06837296820396e-06	11	0.0001170
2463	limited	7.90797589514876e-06	12	0.0001276
606	candidate	1.19060734289307e-05	13	0.0001382
599	camera	2.09911877899371e-05	14	0.0001489
3433	ready	2.09911877899371e-05	15	0.0001595
389	authors	2.15446089370647e-05	16	0.0001701
3125	phd	3.38267054292409e-05	17	0.0001808
3312	projects	3.49912277550768e-05	18	0.0001914

	feature	p_value	rank	imq
2974	org	3.74201040256446e-05	19	0.0002020
681	chairs	5.86017469952769e-05	20	0.0002127
1262	due	6.48878090910497e-05	21	0.0002233
2990	original	6.48878090910497e-05	22	0.0002339
2889	notification	6.88221014991065e-05	23	0.0002446
3671	salary	7.97198143095279e-05	24	0.0002552
3458	record	9.0900377280383e-05	25	0.0002658
3891	skills	9.0900377280383e-05	26	0.0002765
1891	held	0.000152917414319803	27	0.0002871
4177	team	0.000175757009301381	28	0.0002977
3022	pages	0.000200735299729565	29	0.0003084
4628	workshop	0.000200735299729565	30	0.0003190
810	committee	0.000211701960673742	31	0.0003296
3285	proceedings	0.000211701960673742	32	0.0003403
272	apply	0.000216641377784692	33	0.0003509
4039	strong	0.000224630903517758	34	0.0003615
2175	international	0.000229568399555802	35	0.0003722
1088	degree	0.000376232827024858	36	0.0003828
1477	excellent	0.000376232827024858	37	0.0003934
3191	post	0.000376232827024858	38	0.0004041
3243	presented	0.000376514731179718	39	0.0004147

The null hypothesis H_0 for each feature is the treatment has no effect on that feature.Thus, we can conclude that these 39 features are significant since they were rejected.

Appendix

```
# Assignment 1
library(readxl)

# 1.

data <- readxl::read_xlsx("Influenza.xlsx")

library(ggplot2)
ggplot(data, aes(x=Time, y=Mortality,colour="Mortality"))+
  geom_line()+
  geom_line(aes(y=Influenza, colour="Influenza"))+
  labs(y="Amount", title="Mortality vs Influenza")

# 2 fit a GAM model
library(mgcv)
library(akima)
library(plotly)
gam_model <- gam(Mortality~Year+
                  s(Week,k=length(unique(data$Week))),
              data=data,method="GCV.Cp",family="gaussian")
# s=interp(data$Year,data$Week, fitted(gam_model))
cat("GAM model")
```

```
summary(gam_model)
# normal dist+linear regression model
#plot(x=~s$x,y=~s$y, z=~s$z, type="surface")
#spline function normal dist, ~N(mean, std), mean=year, s(week)

# 3

gam_pred <- predict(gam_model, data, type="response")
ggplot(data=data,aes(x=data$Time, y=data$Mortality,colour="Obsevered Mortality"))+
  geom_point()+
  geom_line( aes(y=gam_model$fitted.values,colour="Predicted Mortality"))+
  labs(y="amount of Mortality", x="Time",title="Obsevered VS Predicted")

# lines(x=data$Time,y=data$Mortality,col="blue")
# The quality of the plot is quite good
summary(gam_model)
cat("The p-value of GAM model: ",gam_model$sp )
# The p-value is very small and <0.0001 so it's signficant
# the most significant part is the spline function of Week ***
plot(gam_model, ylab="Mortality amount", main="Spine component")
#According to the graph, it shows that it has low
# mortality in summer and high mortality in winter

# 4

gam_model <- gam(Mortality~Year+
                s(Week,k=length(unique(data$Week))),
                data=data,method="GCV.Cp",
                family="gaussian" ,sp=0.00001)
plot(gam_model$fitted.values,type="l",
     main="Comparison of penalty factor",
     ylab="Fitted values")

gam_model2 <- gam(Mortality~Year+
                    s(Week,k=length(unique(data$Week))),
                  data=data,method="GCV.Cp",
                  family="gaussian" ,sp=1)
lines(gam_model2$fitted.values,type="l",col="red")
legend("topright",legend=c("small penalty",
                           "large penalty"),
         col=c("black","red"), lwd=c(1,1))

cat("The deviance of model 1(low penalty): ", gam_model$deviance, "\n")
cat("The deviance of model 2(high penalty): ", gam_model2$deviance,
    "\n")
cat("   ")
cat("Lower penalty model with higher degree of freedom")
gam_model
cat("    ")
cat("Higher penalty model with lower degree of freedom")
gam_model2

# 5
```

```r
plot(data$Time, mortality$residuals,
     main="Time Series Plot of Residuals",
     ylab="Residuals", xlab="Time")
lines(data$Time, data$Influenza, col="red")
legend("topright", legend="Outbreak of Influenza", col="red", lty=1)

# 6

mortality_s <- gam(Mortality~s(Year,
      k=length(unique(data$Year)))+s(Week,
      k=length(unique(data$Week)))+s(Influenza,
      k=length(unique(data$Influenza))), data=data)
summary(mortality_s)
mortality_s_pred <- predict(mortality_s, newdata = data)
plot(mortality_s_pred, type="o",
     main="Time Series Plot of Mortality",
     xlab="Time", ylab="Mortality", col="red")
points(data$Mortality, type="o", col="black")
legend("topright",
       legend=c("Actual Data", "Prediction"),
       col=c("black", "red"), lty=c(1,1))


# Assignment 2

# 1
RNGversion('3.5.1')
set.seed(12345)

data2 <- read.csv2("data.csv",check.names = FALSE)
# originalcolnames <- colnames(data2)
# colnames(data2) <- paste("F",sep = "",c(1:4703))
n=dim(data2)[1]
id=sample(1:n, floor(n*0.7))
train=data2[id,]
test=data2[-id,]
train[,4703] <- as.vector(train[,4703])
test[,4703] <- as.vector(test[,4703])

library(pamr)
x <- t(train[,-4703])
y <- train[,4703]
mydata <- list(x=x,y=y,
               geneid=as.character(1:nrow(x)),
               genenames=rownames(x))
model <- pamr.train(mydata)
cvmodel <- pamr.cv(model,mydata)
pamr.plotcv(cvmodel)
plot(cvmodel$threshold,cvmodel$error)
# from the cvmodel plot find the optimal threshold
#  here it is around 1
# use print(cvmodel) to confirm
#       that optimal threshold=1.306
# use this pamr model with threshold=1.306 to
# give centroid plot
optthres <- 1.306
pamr.plotcen(model, mydata,threshold = optthres)
# the features selected by the model
fea <- pamr.listgenes(model,
                      mydata,threshold =optthres)

# list the names of the top 10
feanum <- nrow(fea)
first10 <- as.numeric(fea[1:10,1])
# first10 <- originalcolnames[first10]
first10 <- colnames(data2)[first10]
# test error
x1 <- t(test[,-4703])
y1 <- test[,4703]
test_data <- list(x=x1,y=y1,
                  geneid=as.character(1:nrow(x1)),
                  genenames=rownames(x1))
pre <- pamr.predict(model, test_data$x,
                    threshold = optthres,type = "class")
confu_matrix <- table(test[,4703],pre)
# confu_matrix <- table(y1,pre)
misclass <- (confu_matrix[1,2]+
               confu_matrix[2,1])/sum(confu_matrix)

# 2  compute Elastic net and SVM test errors
library(glmnet)
x2 <- as.matrix(train[,-4703])
y2 <- train[,4703]

x3 <- as.matrix(test[,-4703])
y3 <- test[,4703]

cv_ela <-cv.glmnet(x2,y2,alpha=0.5,
                   family="binomial")
plot(cv_ela,main="cv of elastic net")
opt_lambda <- cv_ela$lambda.min
ela <- glmnet(x2,y2,family = "binomial",
               alpha = 0.5,
               lambda =opt_lambda )
feanum2 <- ela$df

pre_ela <- predict(ela,newx = x3, type = "class")
confu_matrix2 <- table(test[,4703],pre_ela)

misclass2 <- (confu_matrix2[1,2]+
               confu_matrix2[2,1])/sum(confu_matrix2)
# svm
library(kernlab)

 colnames(train) <- paste("F",sep = "",c(1:4703))
 train$F4703 <- as.factor(train$F4703)
```

```
svm2 <- ksvm(F4703~., data=train,
            kernel="vanilladot")
colnames(test) <- paste("F",sep = "",c(1:4703))
pre_svm2 <-predict(svm2, test[,-4703])


confu_matrix3 <- table(test[,4703],pre_svm2)
# confu_matrix <- table(y1,pre)
misclass3 <- (confu_matrix3[1,2]+
               confu_matrix3[2,1])/sum(confu_matrix3)

feanum3 <- svm2@nSV

# put results from three models together
tab_comp <- data.frame(Method=c("NSC",
                "Elasticnet","SVM"),
                Test_error=c(misclass,
                             misclass2,
                             misclass3),
                Feature_number=c(feanum,
                                 feanum2,
                                 feanum3))

# 3. Implement Benjamini-Hochberg method

# calculate p-value of each features

pvalue <- data.frame(feature=colnames(data2)[-4703],
                     p_value=rep(0,ncol(data2)-1),
                     rank=rep(0,ncol(data2)-1),
                     imq=rep(0,ncol(data2)-1))
for (i in 1:nrow(pvalue)) {
  re <- t.test(data2[,i]~data2[,4703],
               data = data2, alternative="two.sided")
  pvalue[i,2] <- re$p.value
}

plot(c(1:4702),pvalue[,2],cex=0.05)
# order the p-value and add rank and imq
pvalue <- pvalue[order(pvalue[,2]),]
pvalue[,3] <- c(1:(ncol(data2)-1))
alpha <- 0.05
pvalue[,4] <- alpha * pvalue[,3]/nrow(pvalue)

# find the maximum j which p-value < imq
j <- 0
for (i in 1:nrow(pvalue)) {
  if(pvalue[i,2]<pvalue[i,4])  j <- i
}


plot(c(1:4702),pvalue[,2],cex=0.08,
     xlab="index", ylab="p-value")
```

```
abline(a=pvalue[j,2],b=0,col="red",lty=2)
abline(v=j,col="blue",lty=2)

# print the features correspond to
#  the rejected hypotheses
# because j=39, 1-39 features are rejected

pvalue[1:j,]
```