732A99 Machine Learning - Block2 - Lab2

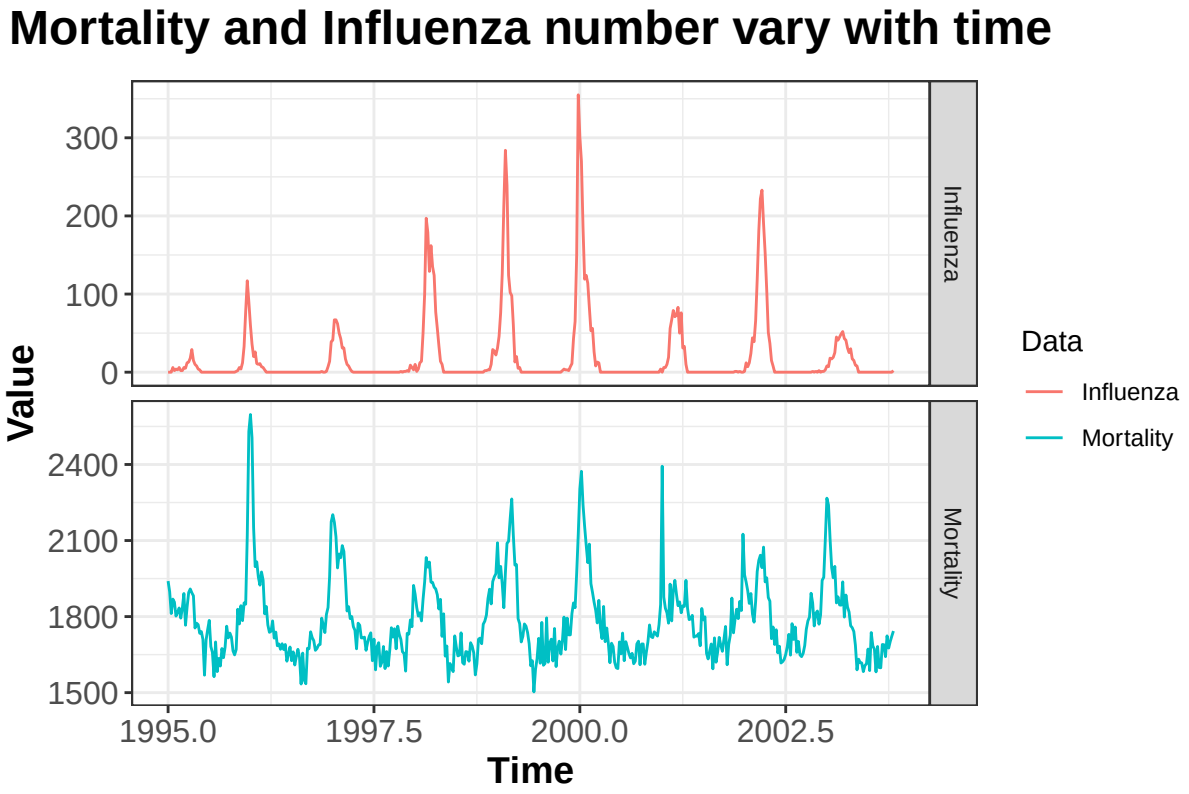
Min-Chun Shih (shimi077)

Contents

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

The Excel document influenza.xlsx contains weekly data on the mortality and the number of laboratory-confirmed cases of influenza in Sweden. In addition, there is information about population-weighted temperature anomalies (temperature deficits).

1.1 Task 1 - Mortality and Influenza number vary with time



- Comment how the amounts of influenza cases are related to mortality rates.

  Ans: From the plot above, it seems Influenza and Mortality share regular peaks. When Mortality goes up, Influenza goes up as well. But we only know they have a similar trend, we do not know if they related to each other. Maybe there is another feature, which influences them at the same time and causes them sharing similar trend.

1.2 Task 2 - GAM model

Use gam() function from mgcv package to fit a GAM model in which Mortality is normally distributed and modelled as a linear function of Year and spline function of Week, and make sure that the model parameters are selected by the generalized cross-validation.

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

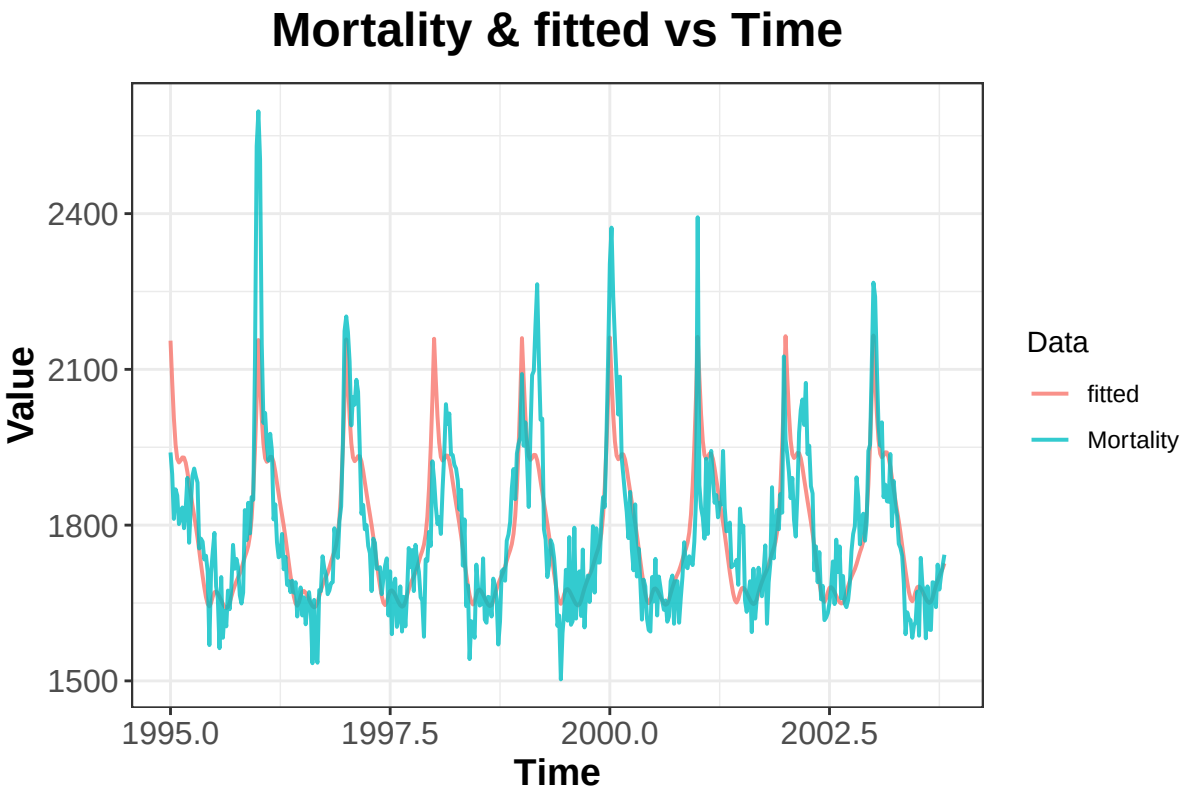
```
## Formula:
## Mortality ~ Year + s(Week, k = length(unique(influenza$Week)))
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -680.598   3367.760  -0.202    0.840
## Year           1.233      1.685   0.732    0.465
##
## Approximate significance of smooth terms:
##           edf Ref.df     F             p-value
## s(Week) 14.32  17.87 53.86 <0.0000000000000002 ***
## ---
## 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
```

- Report the underlying probabilistic model

  Ans: As the question mentions, "Mortality is normally distributed and modeled as a linear function of Year and spline function of Week." Hence we can express the probabilistic model as below.

$$Mortality \sim N(Year + s(Week), \sigma^2)$$

1.3 Task 3 - Plot predicted and observed mortality against time for the fitted model



- Comment on the quality of the fit.

  Ans: From the plot above, I think the prediction fits the data quite well, the peaks and bottoms of fitted values are somewhat similar as real data. However, the fitted model looks to have the same shape in every interval. Hence there are some higher peaks, or lower depths doesn't fit well. But in general, this model looks good.

- Investigate the output of the GAM model and report which terms appear to be significant in the model.

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Mortality ~ Year + s(Week, k = length(unique(influenza$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 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

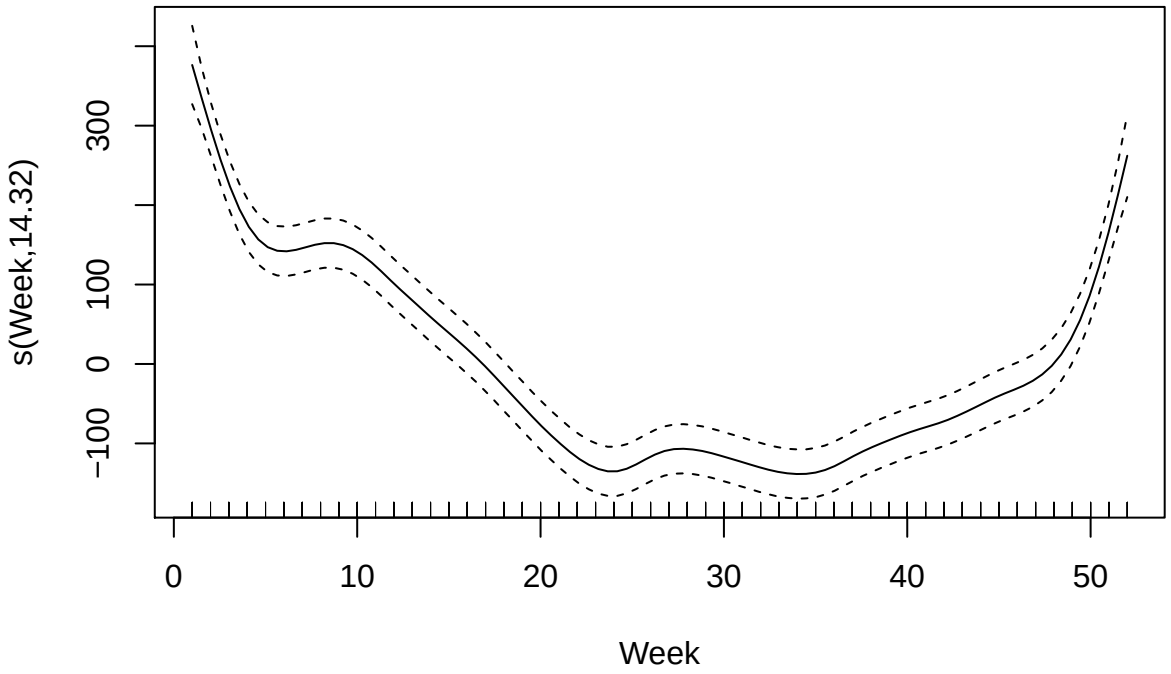
```
## Rank: 52/53
## R-sq.(adj) =  0.677   Deviance explained = 68.8%
## GCV = 8708.6  Scale est. = 8398.9    n = 459
```

Ans: From `summary()` function, the p-value of s(week) is very samll (with ***), which means s(week) is significant in the model.

- Is there a trend in mortality change from one year to another?

  Ans: I think there's no a significant change in mortality trend from one year to another. But the value of mortality does have similar pattern during a year.

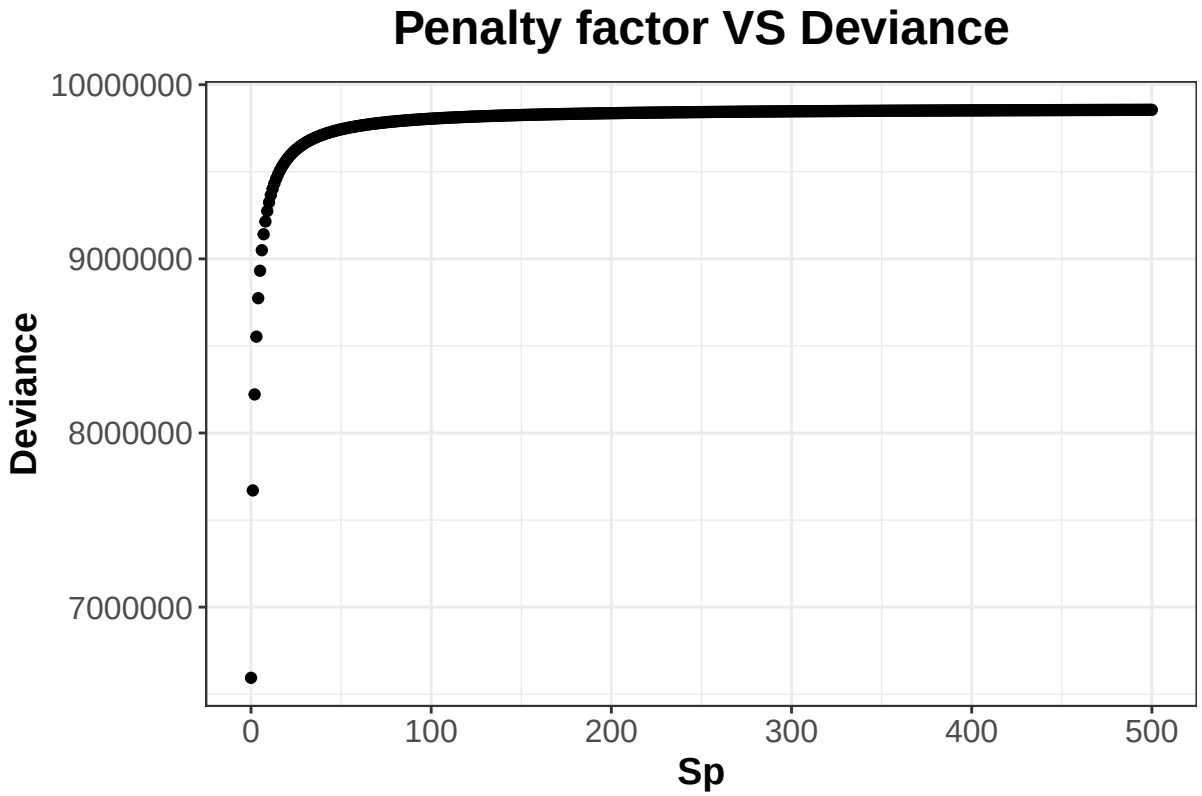
- Plot the spline component and interpret the plot

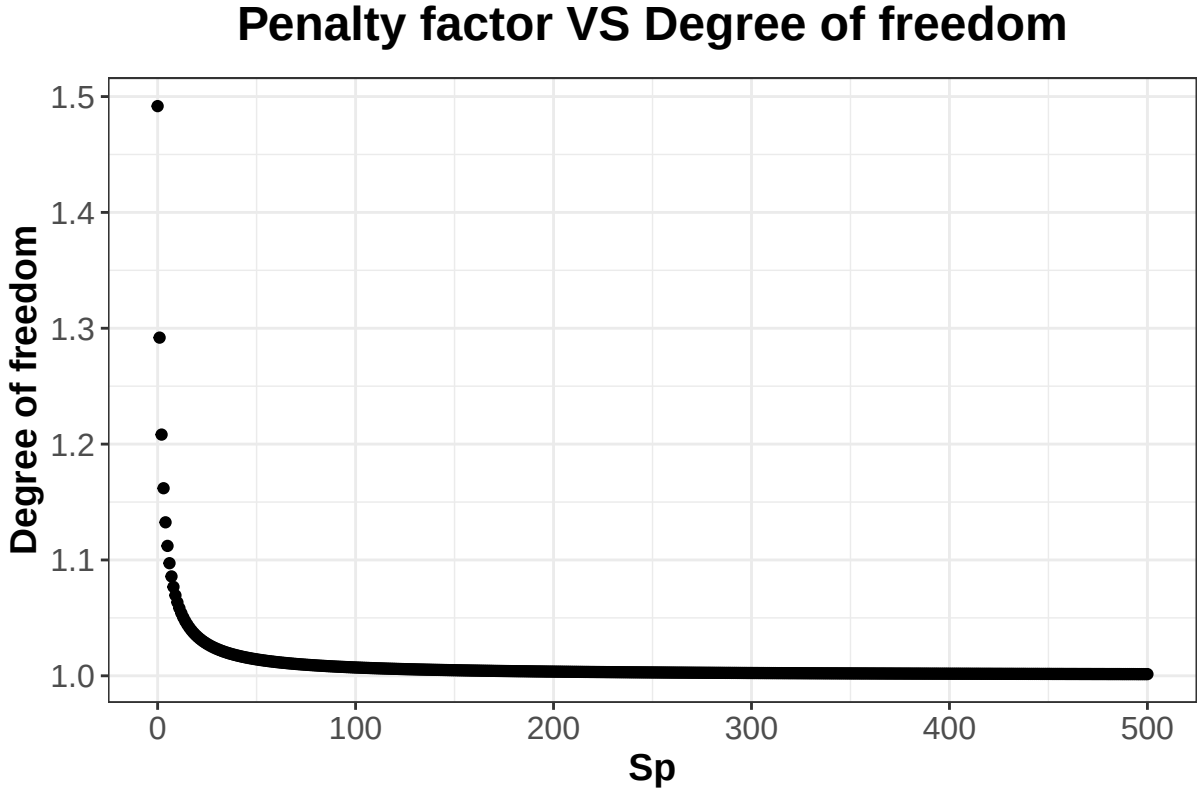


Ans: From the plot above, we can not find a linear line to the fit the shape, which means the spline has different values for different weeks intervals.

1.4 Task 4 - How the penalty factor influences the estimated deviance of the model

Examine how the penalty factor of the spline function in the GAM model from step 2 influences the estimated deviance of the model. Make plots of the predicted and observed mortality against time for cases of very high and very low penalty factors.

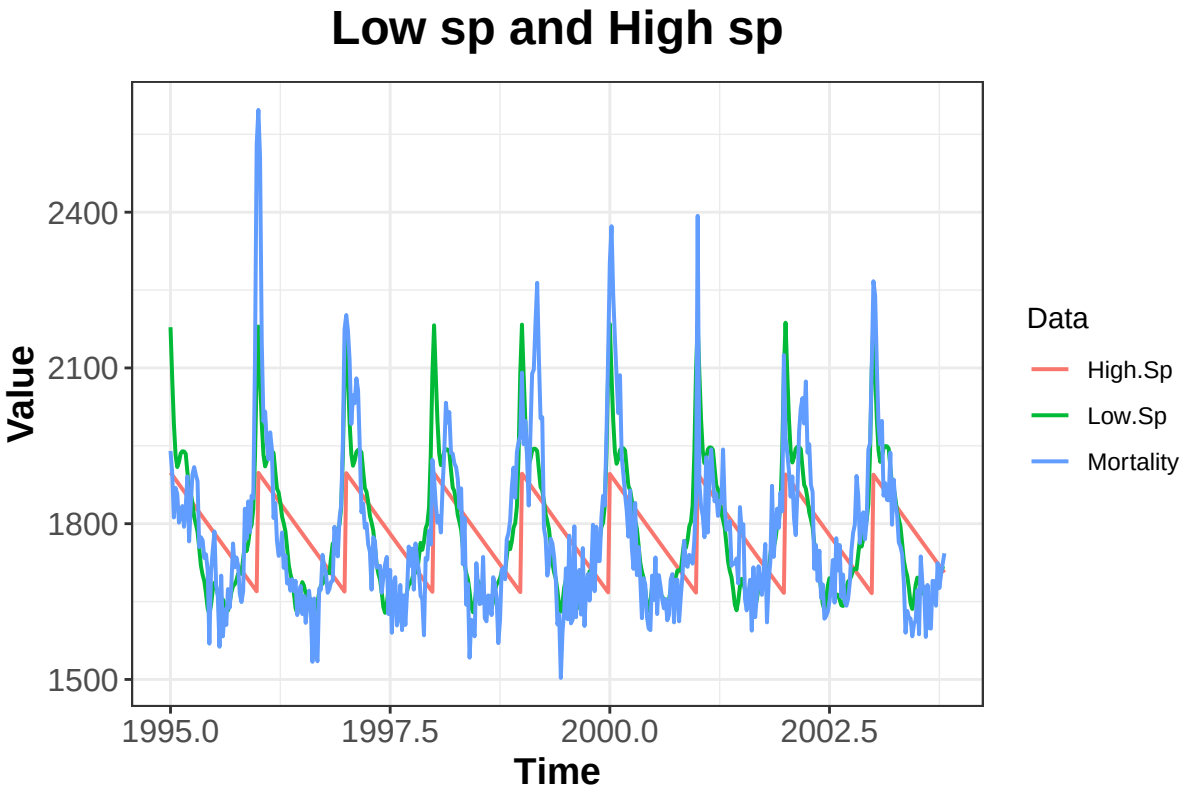




- What is the relation of the penalty factor to the degrees of freedom?

  Ans: As the plot `Penalty factor VS Deviance` we can see, when the penalty factor(Sp) is not too large, the Deviance increases as penalty factor increases. However, after about `Sp = 200`

there is not much influence for Deviance. On the other hand, for the plot of `Penalty factor VS Degree of freedom`, the degree of freedom goes down when Sp goes up. But it does not have much affection when the Sp is getting larger than 200, the Degree of freedom becomes 1.

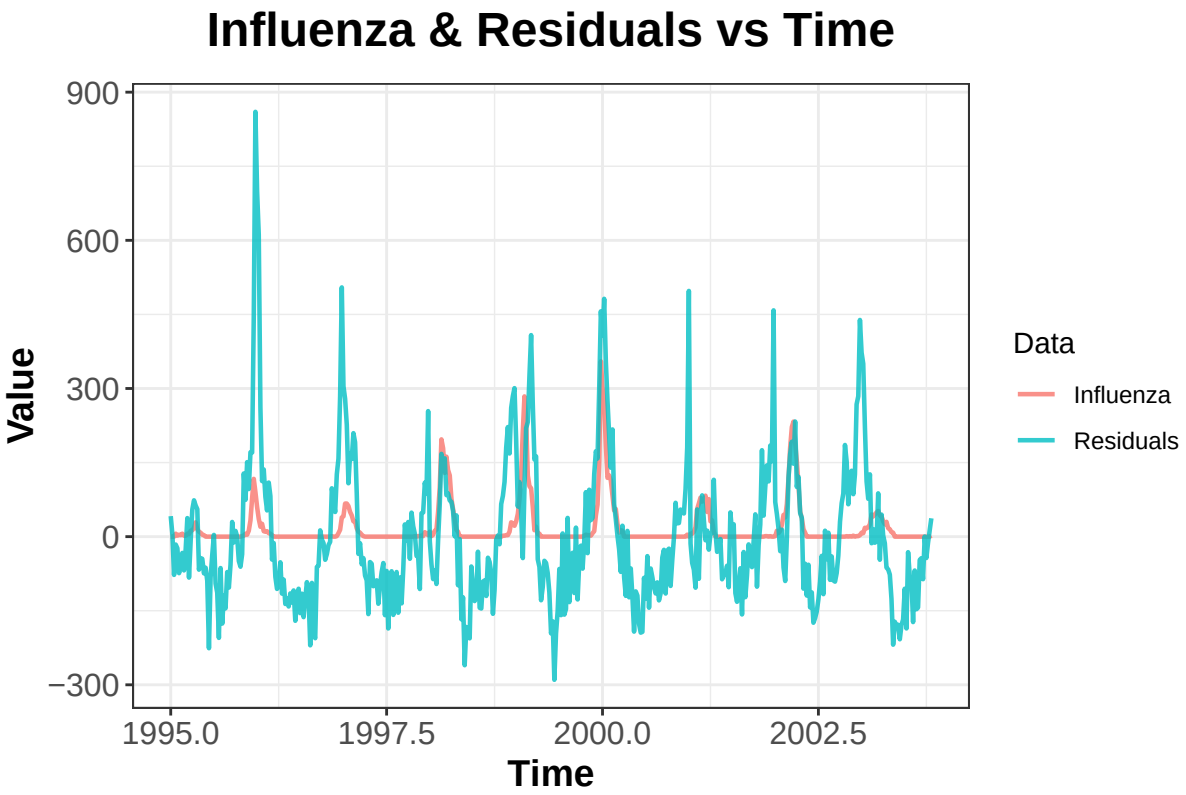


- Do your results confirm this relationship?

  Ans: From the plot above, the graph with high Sp is a line in every interval. We know when Sp is large, Degree of Freedom becomes 1. Hence the graph becomes a line for every range. Other the other hand, when Sp is small, the Degree of Freedom is relatively larger. Thus the plot for `Sp = 0` is bumpy.

1.5 Task 5 - Residuals and the influenza values against time

Use the model obtained in step 2 and plot the residuals and the influenza values against time (in one plot).



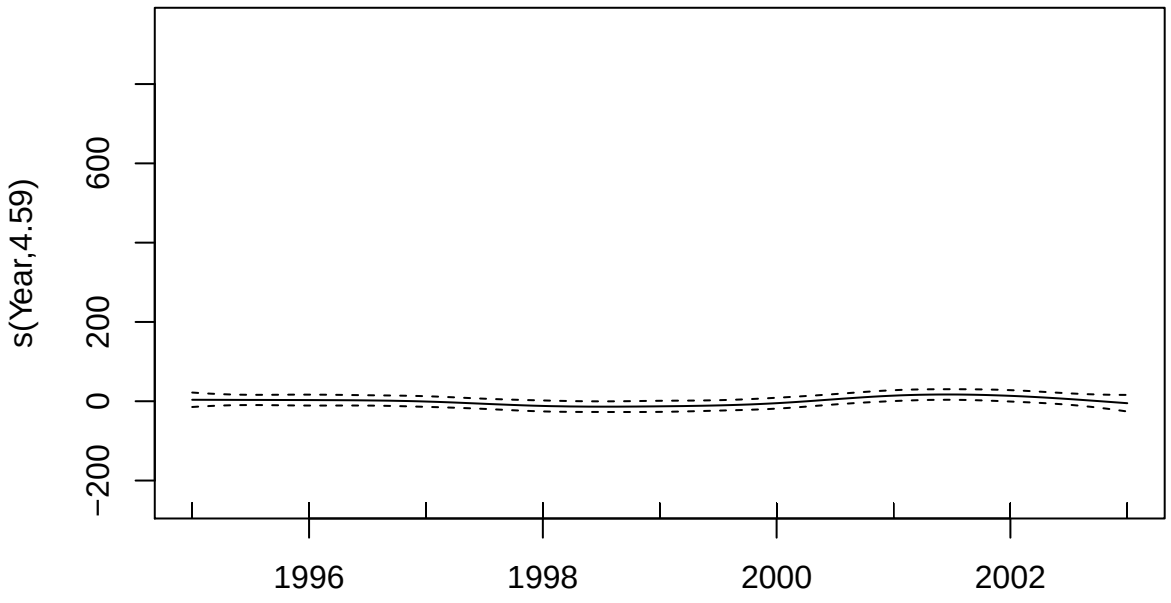
- Is the temporal pattern in the residuals correlated to the outbreaks of influenza?

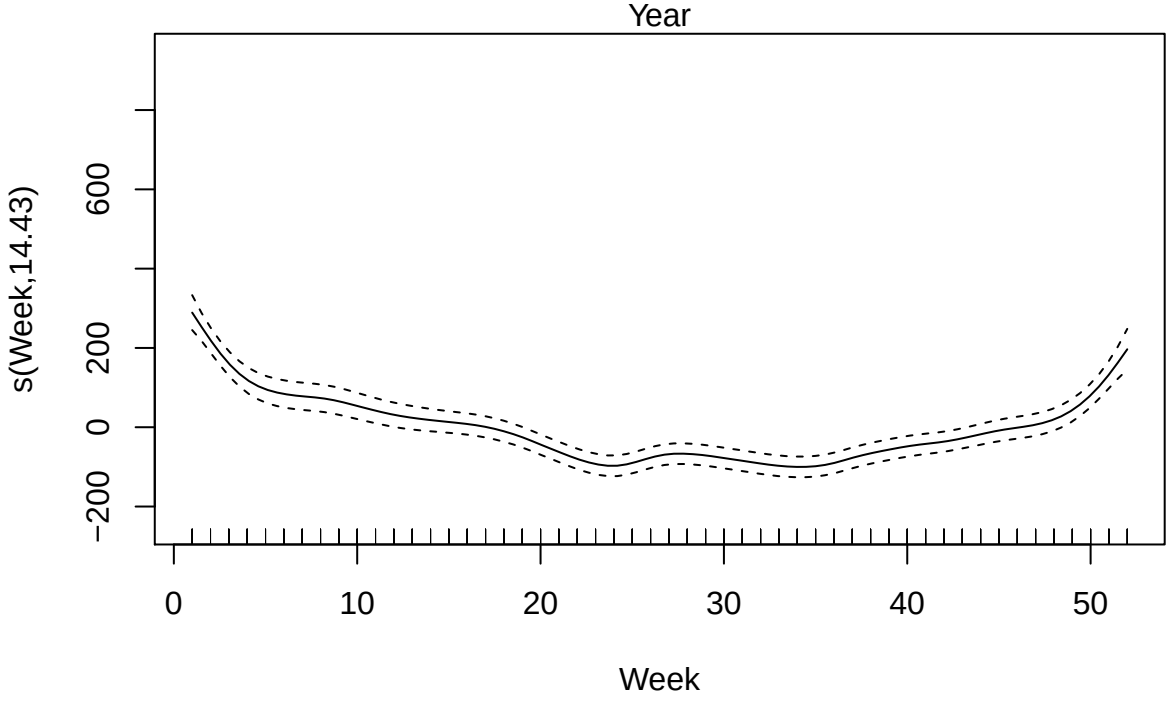
  Ans: Yes, from the graph above, the peaks of residuals are also peaks of influenza. We might guess they correlate.

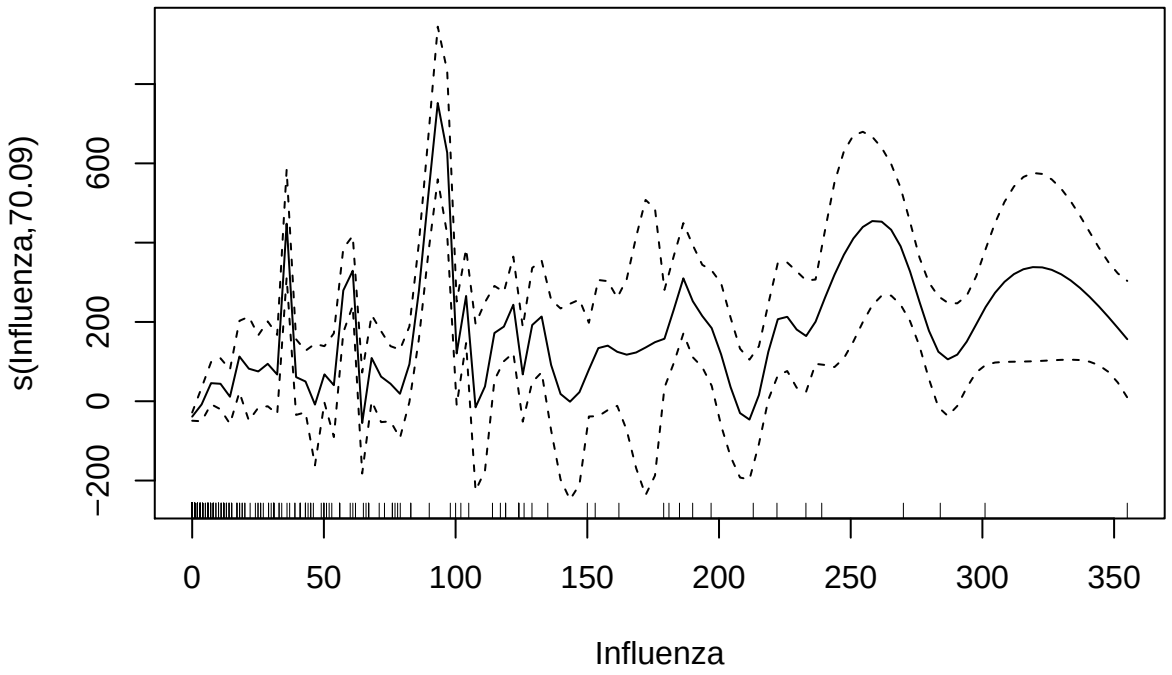
1.6 Task 6 - GAM model

Fit a GAM model in R in which mortality is be modelled as an additive function of the spline functions of year, week, and the number of confirmed cases of influenza.

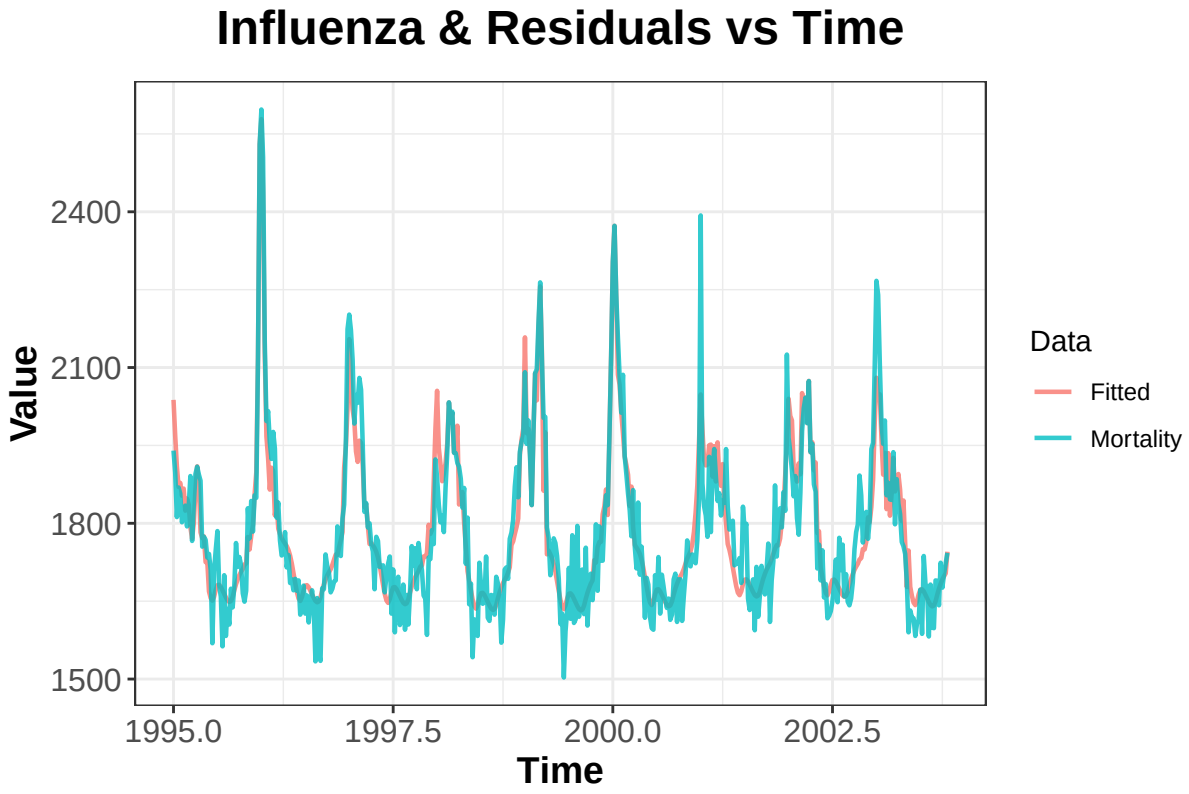
Use the output of this GAM function to conclude whether or not the mortality is influenced by the outbreaks of influenza. Provide the plot of the original and fitted Mortality against Time and comment whether the model seems to be better than the previous GAM models.


s(Year,4.59)
600
200
0
−200
1996
1998
2000
2002
Year


s(Week,14.43)
600
200
0
−200
0
10
20
30
40
50
Week



Ans: From the three plots, we can see s(Year) does not have a different pattern for year intervals. For S(week), there is smoothly changing in weeks. s(Influenza) has the most significant variable for every different Influenza values.



- Comment whether the model seems to be better than the previous GAM models.

  Ans: This plot looks better than previous GAM model. The former model we only considered Year and Week, which made all intervals have the same pattern. Here we include Influenza as a

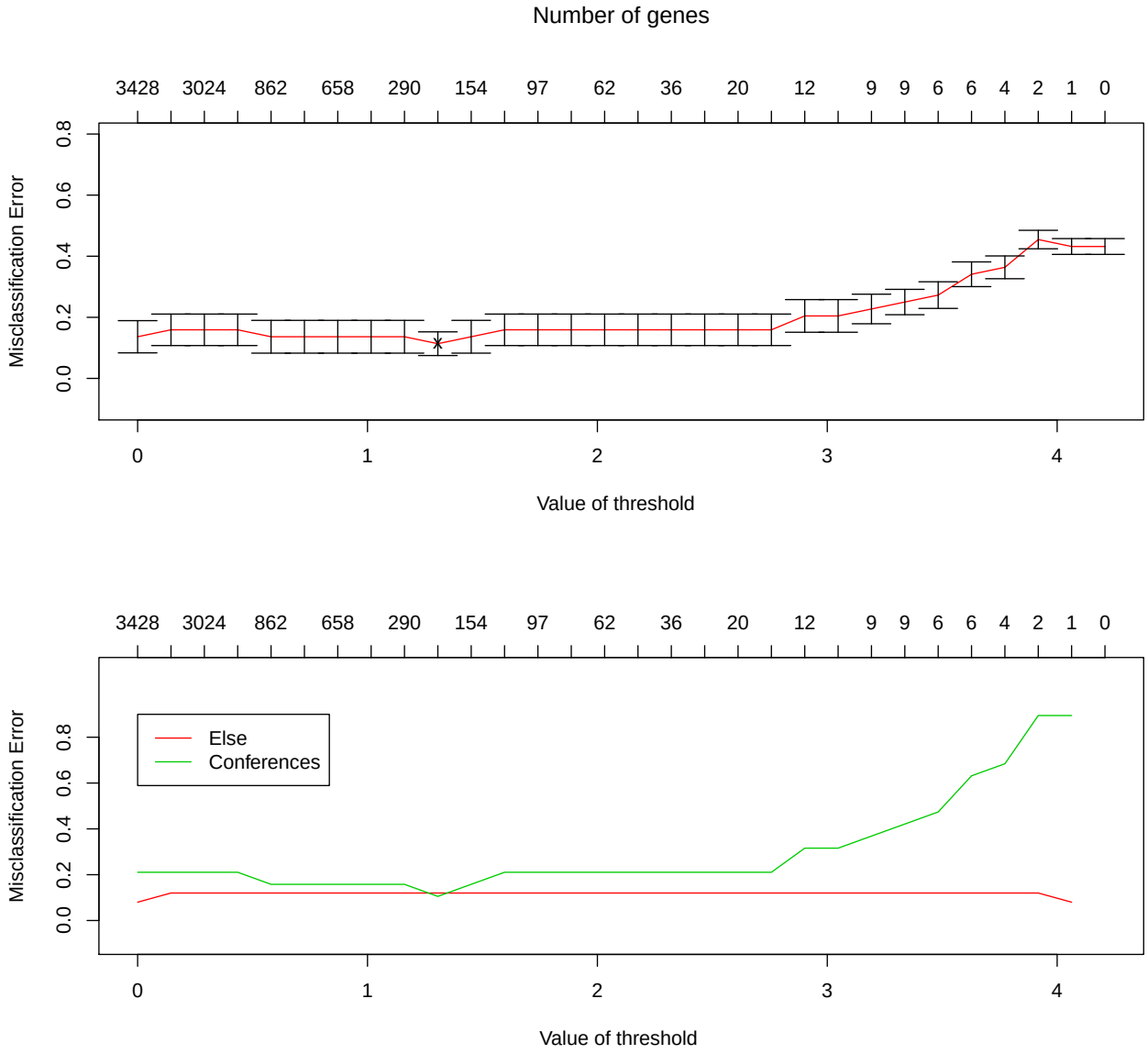
feature, which has various values at different intervals. This new feature improves the performance of fitting real Mortality values significantly. It satisfies the result of the S(Influenza) plot above that S(Influenza) has a different pattern for different values.

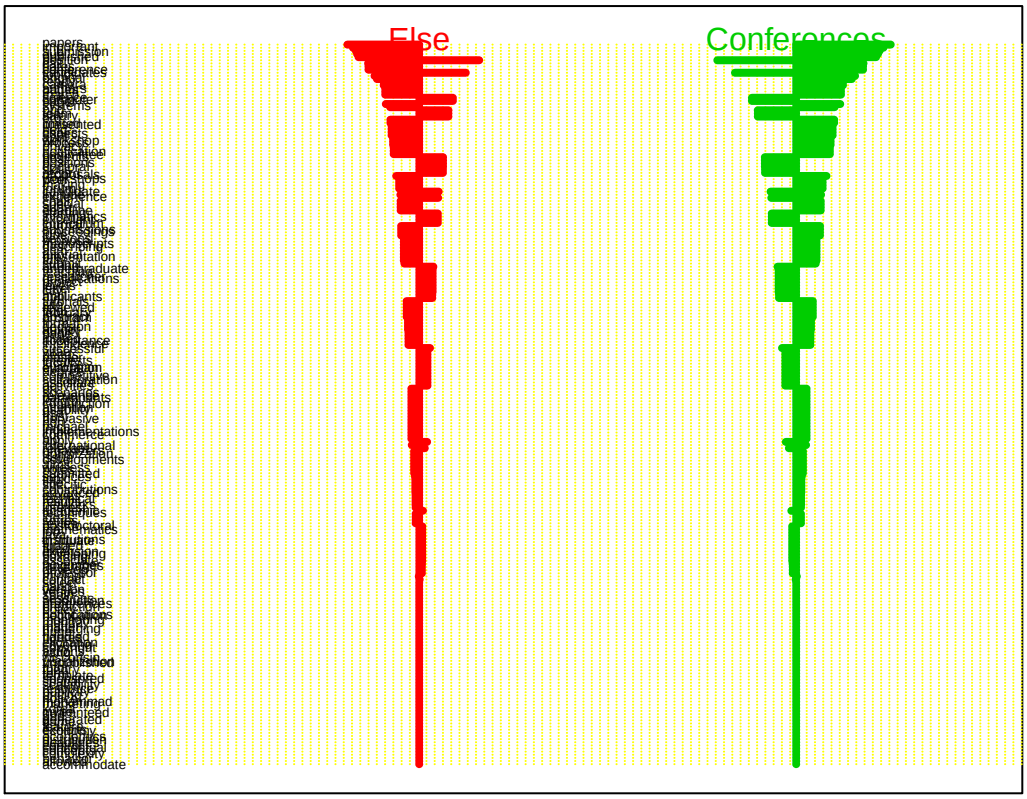
2 Assigment 2 - High-dimensional methods

2.1 Task 1 - Nearest Shrunken Centroids

Divide data into training and test sets (70/30) without scaling. Perform nearest shrunken centroid classification of training data in which the threshold is chosen by cross-validation.

- Provide a centroid plot and interpret it.





- How many features were selected by the method? List the names of the 10 most contributing features and comment whether it is reasonable that they have strong effect on the discrimination between the conference mails and other mails?

```
## There are 231 features are selected by the models.
## 10 most contributing features are: papers,important,submission,due,published
## position,call,conference,dates,candidates
```

id	name	Else-score	Conferences-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

Ans: I think the 10 most contributing features are quite reasonable for conference emails because they are related to conferences, which can be mention in conferences mails quite often. If we see more details, except position and candidates, are down regulated to conferences mails, all other words are positively regulated. Hence, the emails with these positive values in conference scores words would be though more regard to conference.

- Report the test error

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

2.2 Task 2 - Elastic net

degree of freedom 較高的
model 為較好的 model

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

model	Number.of.Error	Test.Error	Number.of.features
NSC	2	0.10	231
Elastic net	2	0.10	32
SVM	1	0.05	4702

- Compare the results of these models with the results of the nearest shrunken centroids (make a comparative table). Which model would you prefer and why?

  Ans: From the table above, we can see the SVM model has the lowest Test Error (0.05), and Elastic net model has the lowest Number of features. If we want to choose the model which has lowest test error, then SVM would be the preferred model, but if we're going to select the model which is more interpretable, the Elastic net model would be the chosen model. However, if we see the result in more detail, the number of error observations between Elastic net model and SVM model only has one different. Hence, I think the Elastic net would be a better choice in this case.

2.3 Task 3 - Benjamini-Hochberg method

Implement Benjamini-Hochberg method for the original data, and use t.test() for computing p-values.

Algorithm of Benjamini-Hochberg method

1. Fix the false discovery rate α and let $p(1) \leq p(2) \leq \ldots \leq p(M)$ denote the ordered p-value.
2. Define
$$L = max\{j : p(j) < \alpha \cdot \frac{j}{M}\}$$
3. Reject all hypotheses H_{0j} for which $p_j \leq p(L)$, the BH rejection threshold.

- Which features correspond to the rejected hypotheses? Interpret the result.

```
## There are 39 features correspond to the rejected hypotheses:
## papers,submission,position,published,important,call,conference,candidates,dates,paper
## topics,limited,candidate,camera,ready,authors,phd,projects,org,chairs
## due,original,notification,salary,record,skills,held,team,pages,workshop
## workshop,committee,proceedings,apply,strong,international,degree,excellent,post,presented
```

Ans: The result from Benjamini-Hochberg method is quite similar as the result of NCS model, but it gets more features then NCS model. The selected features for this algorithm are reliable in general because most of the words seem related to the conference. However, there are some words correspond to the rejected hypotheses I am not sure why they are related to conferences, like camera, ready, and salary. But if we have more information about conferences we can interpret more detail about the features.

```r
knitr::opts_chunk$set(echo = FALSE, message = FALSE, warning = FALSE)
options(scipen=999)
library(ggplot2)
library(tidyr)
library(knitr)
library(kableExtra)
library(openxlsx)
#-------------------------------------------
# Task 1.1
#-------------------------------------------
influenza <- read.xlsx("Influenza.xlsx")

d <- influenza[, c("Time", "Mortality", "Influenza")] %>%
  gather(data, value, -Time)

scaleFactor <- max(influenza$Mortality) / max(influenza$Influenza)

ggplot(d, aes(x = Time, y = value, colour = data)) +
  geom_line() +
  facet_grid(rows = vars(data), scales = "free") +
  labs(title = "Mortality and Influenza number vary with time",
       x = "Time", y = "Value", color = "Data") +  # legend title
  theme_bw() +
  theme(plot.margin = margin(.5, .5, .5, .3, "cm"), # graph margin
        axis.text = element_text(size = rel(1.1)),  # axis labels size
        axis.title = element_text(size = rel(1.3), face = "bold"), # axis names size
        plot.title = element_text(size = rel(1.6), face = "bold",
                                  hjust = 0.5, margin = unit(c(1, 0, 4, 0), "mm"))
  )
#-------------------------------------------
# Task 1.2
#-------------------------------------------
library(mgcv)
gam.model <- gam(Mortality ~ Year + s(Week, k = length(unique(influenza$Week))), data = influenza,
                 family = gaussian(), method = "GCV.Cp")
summary(gam.model)
#-------------------------------------------
# Task 1.3
#-------------------------------------------
fitted <- gam.model$fitted.values

d <- influenza[, c("Time", "Mortality")]
d$fitted <- fitted

d <- d %>%
  gather(data, value, -Time)

ggplot(d, aes(x = Time, y = value, colour = data)) +
  geom_line(alpha = 0.8, size = 0.7) +
  labs(title = "Mortality & fitted vs Time",
       x = "Time", y = "Value", color = "Data") +  # legend title
  theme_bw() +
  theme(plot.margin = margin(.5, .5, .5, .3, "cm"), # graph margin
```

```
        axis.text = element_text(size = rel(1.1)),  # axis labels size
        axis.title = element_text(size = rel(1.3), face = "bold"), # axis names size
        plot.title = element_text(size = rel(1.6), face = "bold",
                                  hjust = 0.5, margin = unit(c(1, 0, 4, 0), "mm"))
  )
summary(gam.model)
plot(gam.model)
#-----------------------------------------
# Task 1.4
#-----------------------------------------
sp <- 0:500
dev <- data.frame(x = numeric(), df = numeric(), dev = numeric())
for (i in 1:length(sp)) {
  gam.model <- gam(Mortality ~ s(Week, k = 52, sp = i) + Year, data = influenza, method = "GCV.Cp")
  dev[i, "dev"] <- gam.model$deviance
  dev[i, "df"] <- sum(gam.model$edf) - gam.model$nsdf
  dev[i, "x"] <- i
}

ggplot(dev, aes(x = sp, y = dev)) +
  geom_point() +
  labs(title = "Penalty factor VS Deviance",
       x = "Sp", y = "Deviance") +  # legend title
  theme_bw() +
  theme(plot.margin = margin(.5, .5, .5, .3, "cm"), # graph margin
        axis.text = element_text(size = rel(1.1)),  # axis labels size
        axis.title = element_text(size = rel(1.3), face = "bold"), # axis names size
        plot.title = element_text(size = rel(1.6), face = "bold",
                                  hjust = 0.5, margin = unit(c(1, 0, 4, 0), "mm"))
  )
ggplot(dev, aes(x = sp, y = df)) +
  geom_point() +
  labs(title = "Penalty factor VS Degree of freedom",
       x = "Sp", y = "Degree of freedom") +  # legend title
  theme_bw() +
  theme(plot.margin = margin(.5, .5, .5, .3, "cm"), # graph margin
        axis.text = element_text(size = rel(1.1)),  # axis labels size
        axis.title = element_text(size = rel(1.3), face = "bold"), # axis names size
        plot.title = element_text(size = rel(1.6), face = "bold",
                                  hjust = 0.5, margin = unit(c(1, 0, 4, 0), "mm"))
  )
# gam model with low sp
gam.model.low <- gam(Mortality ~  Year + s(Week, k = length(unique(influenza$Week)), sp = 0),
                     data = influenza, family = gaussian(), method = "GCV.Cp")
# gam model with high sp
gam.model.high <- gam(Mortality ~ Year + s(Week, k = length(unique(influenza$Week)), sp = 500), data = i
                  family = gaussian(), method = "GCV.Cp")

d <- data.frame(Low.Sp = fitted(gam.model.low), High.Sp = fitted(gam.model.high),
                Mortality = influenza$Mortality, Time = influenza$Time)
d <- d %>%
  gather(Data, Value, -Time)
```

```r
ggplot(d, aes(x = Time, y = Value, colour = Data)) +
  geom_line(size = .7) +
  labs(title = "Low sp and High sp") +
  theme_bw() +
  theme(plot.margin = margin(.5, .5, .5, .3, "cm"),
        axis.text = element_text(size = rel(1.1)),
        axis.title = element_text(size = rel(1.3), face = "bold"),
        plot.title = element_text(size = rel(1.6), face = "bold",
                                  hjust = 0.5, margin = unit(c(1, 0, 4, 0), "mm"))
  )
#-----------------------------------------
# Task 1.5
#-----------------------------------------
d <- data.frame(Influenza = influenza$Influenza, Residuals = gam.model$residuals, Time = influenza$Time)
d <- d %>%
  gather(Data, Value, -Time)
ggplot(d, aes(x = Time, y = Value, colour = Data)) +
  geom_line(size = 0.8, alpha = 0.8) +
  labs(title = "Influenza & Residuals vs Time") +
  theme_bw() +
  theme(plot.margin = margin(.5, .5, .5, .3, "cm"), # graph margin
        axis.text = element_text(size = rel(1.1)),  # axis labels size
        axis.title = element_text(size = rel(1.3), face = "bold"), # axis names size
        plot.title = element_text(size = rel(1.6), face = "bold",
                                  hjust = 0.5, margin = unit(c(1, 0, 4, 0), "mm"))
  )
#-----------------------------------------
# Task 1.6
#-----------------------------------------
gam.model <- gam(Mortality ~
                   s(Year, k = length(unique(influenza$Year))) +
                   s(Week, k = length(unique(influenza$Week))) +
                   s(Influenza, k = length(unique(influenza$Influenza))), data = influenza, method="GCV
plot(gam.model)
fitted <- gam.model$fitted.values

d <- data.frame(Mortality = influenza$Mortality, Fitted = fitted, Time = influenza$Time)
d <- d %>%
  gather(Data, Value, -Time)
ggplot(d, aes(x = Time, y = Value, colour = Data)) +
  geom_line(size = 0.8, alpha = 0.8) +
  labs(title = "Influenza & Residuals vs Time") +
  theme_bw() +
  theme(plot.margin = margin(.5, .5, .5, .3, "cm"), # graph margin
        axis.text = element_text(size = rel(1.1)),  # axis labels size
        axis.title = element_text(size = rel(1.3), face = "bold"), # axis names size
        plot.title = element_text(size = rel(1.6), face = "bold",
                                  hjust = 0.5, margin = unit(c(1, 0, 4, 0), "mm"))
  )
#-----------------------------------------
# Task 2.1
#-----------------------------------------
# divide data into training and test data
```

```r
set.seed(12345)
d <- read.delim("data.csv", sep = ";", encoding = "latin1")
d$Conference <- factor(d$Conference, labels = c("Else", "Conferences"))
n <- nrow(d)
id <- sample(1:n, floor(n*0.7))
train <- d[id, ]
test <- d[-id, ]

# NSC
library(pamr)
x <- t(as.matrix(train[, -which(names(train) == "Conference")]))
y <- train$Conference
x.list <- list(x = x, y = y, geneid = as.character(1:nrow(x)), genenames = rownames(x))
nsc.model <- pamr.train(x.list)

# choose threshold by cross-validation
mycv <- pamr.cv(nsc.model, x.list)
pamr.plotcv(mycv)
# theshold is choosen by min of error
threshold <- mycv$threshold[which.min(mycv$error)]
pamr.plotcen(nsc.model, x.list, threshold = threshold)
# get features
a <- pamr.listgenes(nsc.model, x.list, threshold = threshold, genenames = TRUE)
nsc.n.features <- nrow(a)
cat("There are", nsc.n.features, "features are selected by the models. \n\n")
cat("10 most contributing features are:", paste(a[1:5, "name"], collapse = ","), "\n")
cat(paste(a[6:10, "name"], collapse = ","))

kable(a[1:10, ]) %>%
  kable_styling(bootstrap_options = "striped", full_width = F, position = "center")
x.test <- t(as.matrix(test[, -which(names(test) == "Conference")]))
ncs.pred <- pamr.predict(nsc.model, x.test, threshold = threshold)

ncs.error.n <- length(which(test$Conference != ncs.pred))
ncs.test.error <- ncs.error.n/length(ncs.pred)

cat("The test error:", ncs.test.error)
#---------------------------------------------
# Task 2.2
#---------------------------------------------
# Elastic net
library(glmnet)
x <- as.matrix(train[, -which(names(train) == "Conference")])
y <- (train$Conference)

net.model <- cv.glmnet(x, y, family = "binomial", alpha = 0.5)

x.test <- as.matrix(test[, -which(names(test) == "Conference")])
net.pred <- predict.cv.glmnet(net.model, newx = x.test, s = "lambda.min", type = "class")

net.error.n <-  length(which(test$Conference != as.factor(net.pred)))
net.test.error <- net.error.n/length(ncs.pred)
```

```r
net.features <- coef(net.model, s = "lambda.min")
net.features <- data.frame(features = as.vector(net.features), names = rownames(net.features))
net.features <- net.features[which(net.features$features != 0)[-1], ]
net.features <- as.character(net.features[order(abs(net.features$features)), "names"])
net.n.features <- length(net.features)
library(kernlab)
svm.model <- ksvm(Conference ~. , data = train, kernel = "vanilladot")
svm.pred <- predict(svm.model, test)
svm.error.n <- length(which(svm.pred != test$Conference))
svm.test.error <- svm.error.n/length(svm.pred)
# svm.n.features <- length(svm.model@coef[[1]])-1
svm.n.features <- ncol(train) - 1
compare.table <- data.frame(model = c("NSC", "Elastic net", "SVM"),
                            `Number of Error` = c(ncs.error.n, net.error.n, svm.error.n),
                            `Test Error` = c(ncs.test.error, net.test.error, svm.test.error),
                            `Number of features` = c(nsc.n.features, net.n.features, svm.n.features))
kable(compare.table) %>%
  kable_styling(bootstrap_options = "striped", full_width = F, position = "center")
#-----------------------------------------------
# Task 2.3
#-----------------------------------------------
n <- ncol(d) - 1
p.value <- c()
for (i in 1:n) {
  f <- d[, i]
  res <- t.test(f ~ Conference, data = d, alternative = "two.sided")
  p.value[i] <- res$p.value
}

names(p.value) <- names(d)[1:n]

alpha <- 0.05
p.value <- sort(p.value)
L <- max(p.value[(p.value - alpha*1:n/n) < 0 ])
rejected <- p.value[1:which(p.value == L)]
cat("There are", length(rejected), "features correspond to the rejected hypotheses: \n")
cat(paste(names(rejected)[1:10], collapse = ","), "\n")
cat(paste(names(rejected)[11:20], collapse = ","), "\n")
cat(paste(names(rejected)[21:30], collapse = ","), "\n")
cat(paste(names(rejected)[30:length(rejected)], collapse = ","), "\n")
```