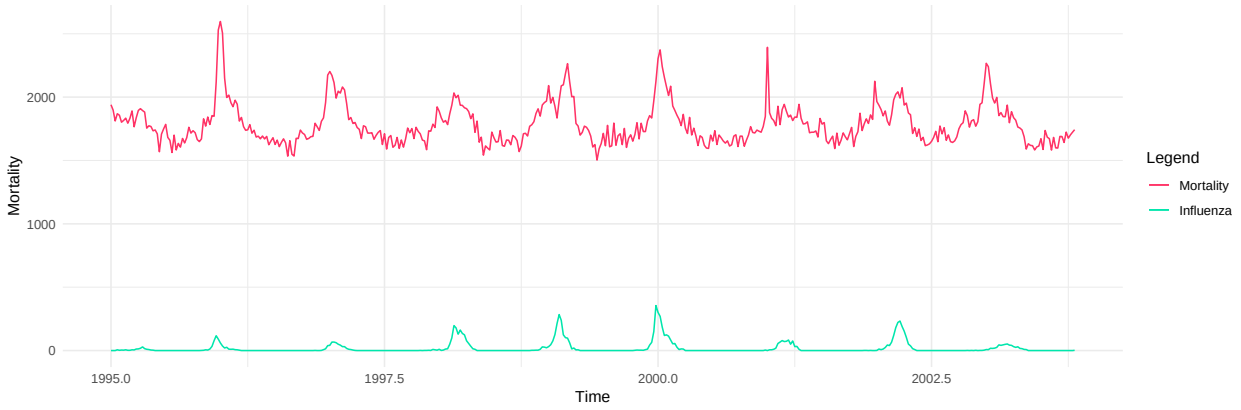
Lab 2 Block 2

Ahmed Alhasan

12/13/2019

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

1.1 Relation between Influenza & Mortality



- Since high Mortality rates correspond with high number of Influenza outbreaks, it indicates there is a relationship between Influenza and Mortality.
- This relationship is not necessarily a cause and effect relationship, it could either be that "Mortality is a direct of effect to Influenza outbreaks", and/or it could mean they are both direct effects of winter (the first and last weeks in the year).
- It can be seen that the increase in the number of Influenza outbreaks every winter does not linearly correspond with the increase in Mortality rates(does not increase in the same ratio). One instance is that winter 95-96 happened to have the highest Mortality rate in the recorded period but the same can not be said about the number of Influenza outbreaks.

1.2 GAM model

Probabilistic Model:

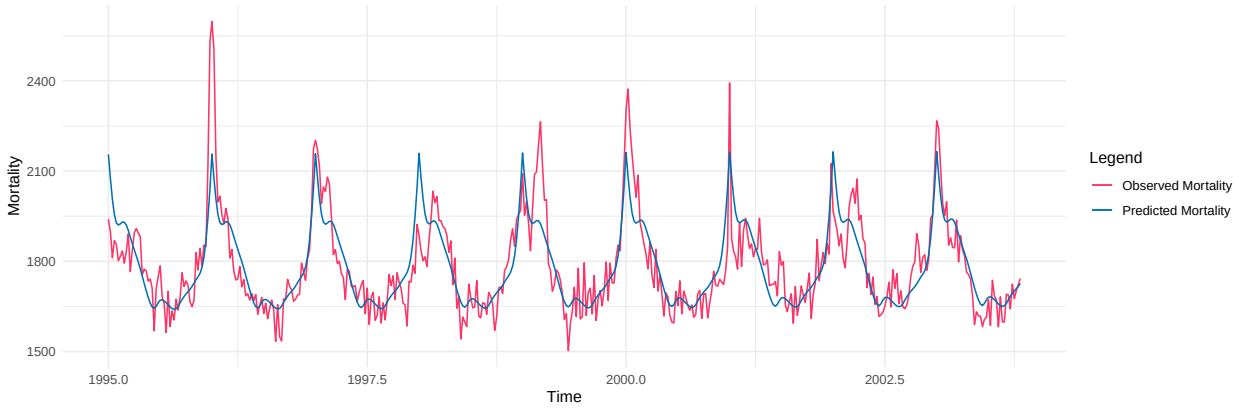
$$y \sim N(\mu, \sigma^2)$$

$$\hat{y} = -680.598 + 1.232846\ (Year_i) + s_i(Week_k) + \epsilon_i$$

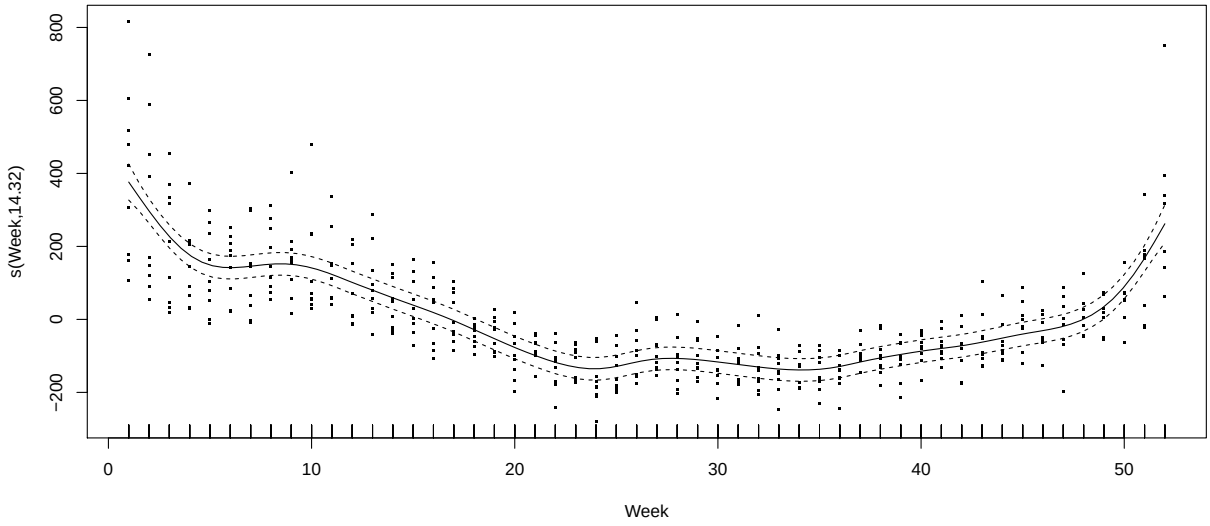
$$where\ i = 1, 2, ..., 9\ number\ of\ years,\ \ k = 1, 2, ..., 52\ number\ of\ weeks$$

The resulting coefficients matrix will have 459 rows(i*k), k actually is less than 9 since last year is not fully recorded, and 52 columns (k).

1.3 Predicted vs Observed Mortality




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



Report which terms appear to be significant in the model

- "Year" which is linearly linked to Mortality is insignificant variable and has a very slight contribution to the model.
- "Week" actually is very significant and it has non-linear relationship with Mortality.
- k (52 basis dimensions) is set to the total number of weeks represents the knots between the splines, while the Effective Degrees of Freedom "edf" is 14.32 which is selected based on the penalty factor set by GCV.
- The model explains 68.8% of the variance (almost all of this contribution is from the "Week" spline).

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

- The actual values of Mortality change greatly within the same year and moderately between years.
- The prediction however changes only within the year itself and the change between different years basically non-existent, this is due to the fact we took linear relationship with "Year" in our model which has intangible contribution to the model. This linearity might not well descripe the actual relationship with Mortality"

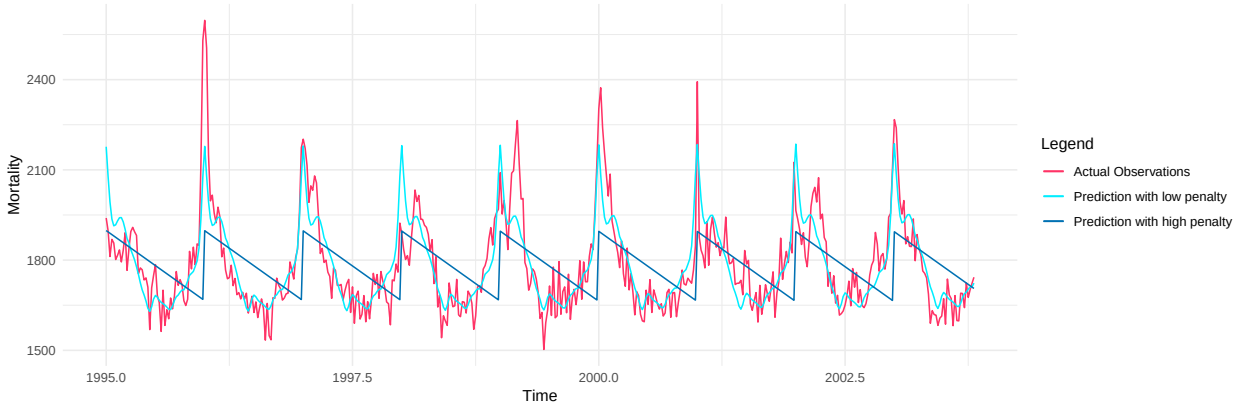
Interpret the spline plot?

- The x-axis represent the weeks, and the y-axis represent the residual values. The line represent the relation betwen Mortality and Week and it is the highest in the first and last weeks (winter) and lowest in mid-year and the interval around it is the 95% confidence interval of the Expected value of the model. The dots represent the residuals.
- From the plot we can see the spline fit well on one year average data.

1.4 Prediction with Different Penalty Factors

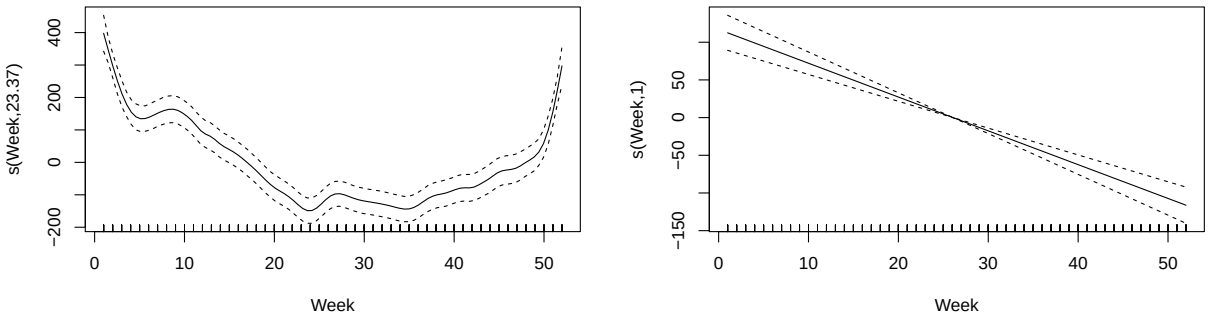
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Mortality ~ Year + s(Week, k = w, sp = 1e-05)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -693.869   3370.267  -0.206    0.837
## Year           1.239      1.686   0.735    0.463
##
## Approximate significance of smooth terms:
##           edf Ref.df     F p-value
## s(Week) 23.37  27.43 35.63  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Rank: 36/53
## R-sq.(adj) =  0.676   Deviance explained = 69.4%
## GCV = 8902.1  Scale est. = 8410      n = 459
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Mortality ~ Year + s(Week, k = w, sp = 10000)
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2810.0661  5398.4273   0.521    0.603
## Year          -0.5134     2.7007  -0.190    0.849
##
## Approximate significance of smooth terms:
##         edf Ref.df    F p-value
## s(Week)   1      1 93.8  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.167   Deviance explained = 17.1%
## GCV =  21783  Scale est. = 21640     n = 459
```



Examine how the penalty factor influences the estimated deviance of the model?

- Increasing the penalty factor make the model underfitted and reduced the accuracy of the model, therefore; the deviance explained by the model got reduced.



- This Increase in penalty reduces the wiggliness of the spline until it becomes a straight line. And this line is repeated over the years, that's why it gives this zigzag shape in the time series plot.

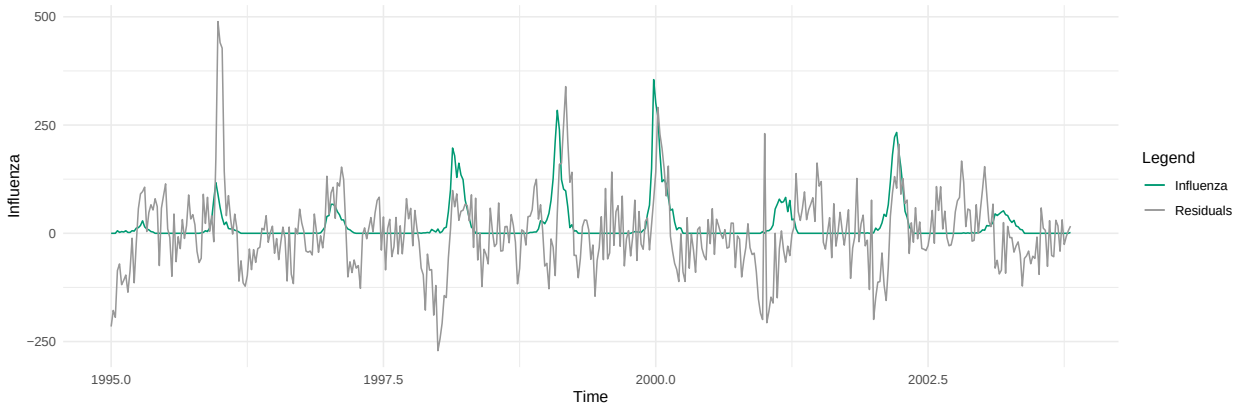
What is the relation of the penalty factor to the degrees of freedom?

- Increasing the penalty factor eliminates the insignificant basis dimentions until it reaches 1 which means "Week" is linearly related to Mortality.

Do your results confirm this relationship?

- Yes, it is.

1.5 Correlation between Residuals and Influenza

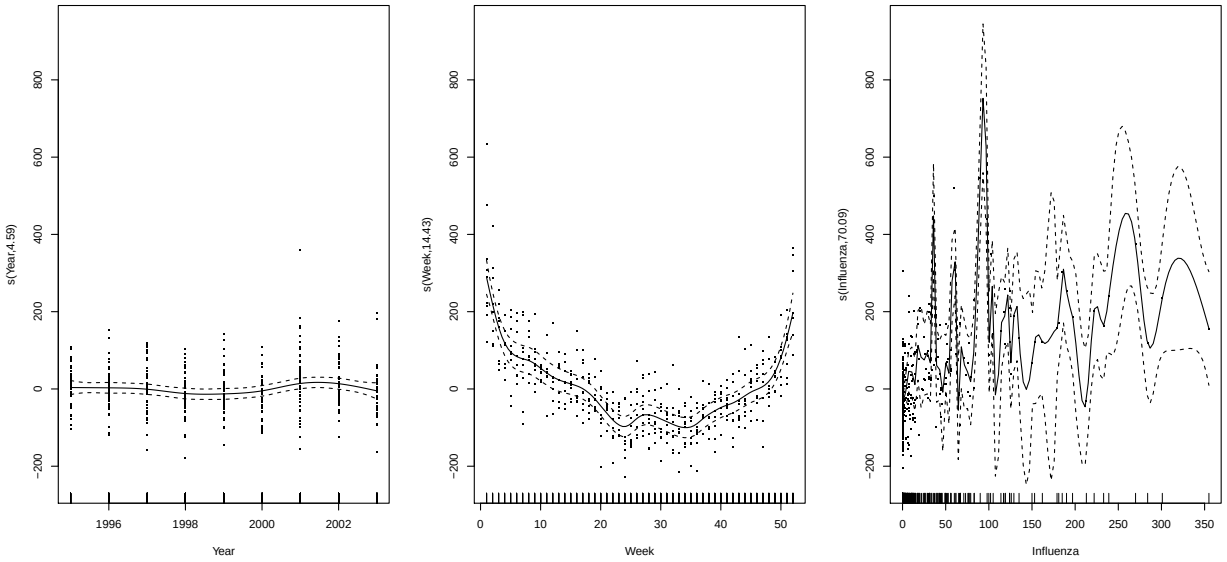


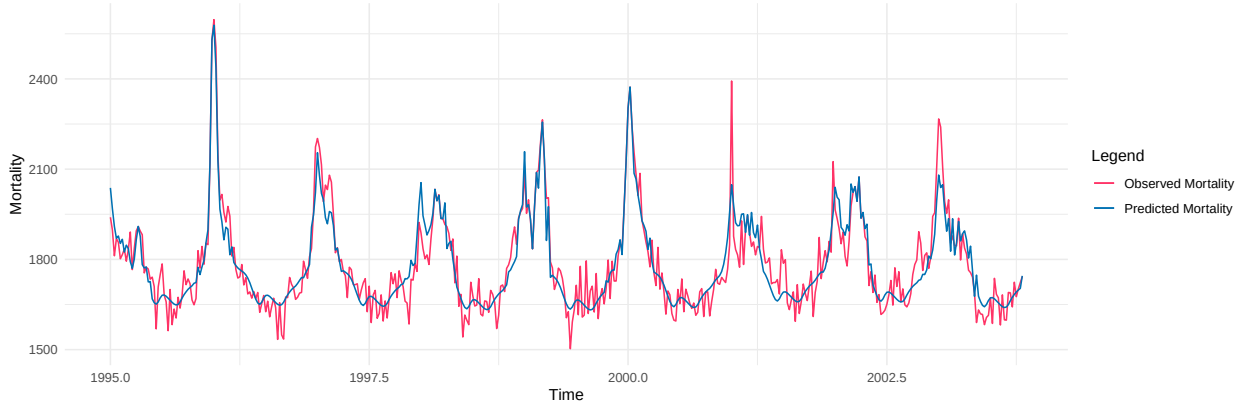
Is the temporal pattern in the residuals correlated to the outbreaks of influenza?

- The Influenza could explain some of the positive residuals (where the model in 1.2 underestimates the mortality), but it still can't explain the negative and some of the positive residuals. So there is some correlation between the two.

1.6 Final Model

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Mortality ~ s(Year, k = y) + s(Week, k = w) + s(Influenza, k = f)
##
## 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
```





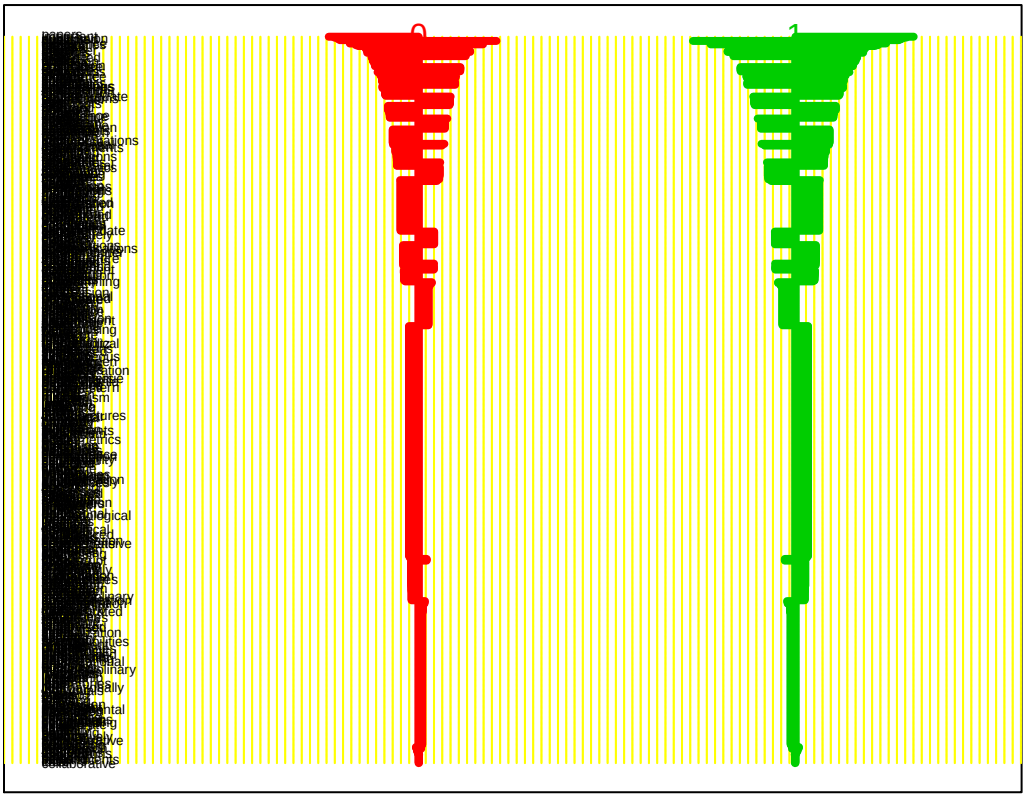
Conclude whether or not the mortality is influenced by the outbreaks of influenza.

- Adding the Influenza to the model increased the accuracy considerably but that could be overfitting.
- It can be concluded that outbreaks of influenza have some influence on mortality.

Assignment 2. High-dimensional methods

2.1 Nearest Shrunken Centroid

Provide a centroid plot and interpret it.



- The plot shows only the features that play a role in classification, in this case (using set.seed(12345)) 862 features selected. This number is determined by C.V. based on the threshold that gave the least errors. The words with the longest pars at the top are the ones than can classify more correctly. Because the top words can only be seen in one of the two classes, we can see them exclusively either on the "1" side or the "0".

How many features were selected by the method?

```
## [1] 862
```

List the names of the 10 most contributing features

```
##       [,1]
##  [1,] "papers"
##  [2,] "important"
##  [3,] "submission"
##  [4,] "due"
##  [5,] "published"
##  [6,] "position"
```

```
##  [7,] "call"
##  [8,] "conference"
##  [9,] "dates"
## [10,] "candidates"
```

comment whether it is reasonable that they have strong effect on the discrimination between the conference mails and other mails?

- The NSC algorithm select every feature that have effect on the classification depending on the threshold selected by the C.V. function, in this case the threshold is relatively low thats why a lot of the features have been selected even with minimum effect.

Report the test error.

```
## [1] 0.05
```

2.2 Comparision with Elastic Net & SVM

	Error Rate	Features Selected
NSC	0.05	862
Elastic Net	0.1	39
SVM	0.1	43

Which model would you prefer and why?

- The NSC gave the least error but used too many variables, and could have been a higher rate if set.seed was different. The Elastic Net and SVM perform very close to each other, however Elastic Net is more interpretable and preferable to the other two.

2.3 Benjamini-Hochberg

Which features correspond to the rejected hypotheses?

```
##                 P.Values      T_F
## papers       1.116910e-10 Rejected
## submission   7.949969e-10 Rejected
## position     8.219362e-09 Rejected
## published    1.835157e-07 Rejected
## important    3.040833e-07 Rejected
## call         3.983540e-07 Rejected
## conference   5.091970e-07 Rejected
## candidates   8.612259e-07 Rejected
## dates        1.398619e-06 Rejected
## paper        1.398619e-06 Rejected
## topics       5.068373e-06 Rejected
```

```
## limited       7.907976e-06 Rejected
## candidate     1.190607e-05 Rejected
## camera        2.099119e-05 Rejected
## ready         2.099119e-05 Rejected
## authors       2.154461e-05 Rejected
## phd           3.382671e-05 Rejected
## projects      3.499123e-05 Rejected
## org           3.742010e-05 Rejected
## chairs        5.860175e-05 Rejected
## due           6.488781e-05 Rejected
## original      6.488781e-05 Rejected
## notification  6.882210e-05 Rejected
## salary        7.971981e-05 Rejected
## record        9.090038e-05 Rejected
## skills        9.090038e-05 Rejected
## held          1.529174e-04 Rejected
## team          1.757570e-04 Rejected
## pages         2.007353e-04 Rejected
## workshop      2.007353e-04 Rejected
## committee     2.117020e-04 Rejected
## proceedings   2.117020e-04 Rejected
## apply         2.166414e-04 Rejected
## strong        2.246309e-04 Rejected
## international 2.295684e-04 Rejected
## degree        3.762328e-04 Rejected
## excellent     3.762328e-04 Rejected
## post          3.762328e-04 Rejected
## presented     3.765147e-04 Rejected
```

Interpret the result.

- The list of words selected by Benjamini-Hochberg method emphisize on lowering the false-discovery rate, meaning these words are the ones that give the least False Positive errors.

Appendix

```
setwd('E:/Workshop/Machine Learning/Block 2/Lab 2 Block 2')
suppressWarnings(RNGversion('3.5.1'))

library(readxl)
library(ggplot2)
library(mgcv)

flu <- read_excel("Data/influenza.xlsx")

ggplot(flu)+
  geom_line(aes(x = Time, y = Mortality, color = "Mortality")) +
  geom_line(aes(x = Time, y = Influenza, color = "Influenza")) +
  scale_color_manual("Legend",
                     breaks = c("Mortality", "Influenza"),
                     values = c("#00E6AC", "#FF3366"))+
  theme_minimal()
```

```r
w <- length(unique(flu$Week))

model <- gam(Mortality ~ Year + s(Week, k = w),
             data   = flu,
             family = "gaussian",
             method = "GCV.Cp")

pred <- predict(model)

df1 <- data.frame(Time = flu$Time,
                  Mortality  = flu$Mortality,
                  Prediction = pred,
                  Influenza  = flu$Influenza,
                  Residuals  = model$residuals)

ggplot(df1)+
  geom_line(aes(x = Time, y = Mortality, color = "Observed Mortality")) +
  geom_line(aes(x = Time, y = Prediction, color = "Predicted Mortality")) +
  scale_color_manual("Legend",
                     breaks = c("Observed Mortality", "Predicted Mortality"),
                     values = c("#FF3366", "#0071B3"))+
  theme_minimal()

summary(model)
plot(model, residuals = TRUE, cex = 2)

low_model  <- gam(Mortality ~ Year + s(Week, k = w, sp=0.00001),
                  data   = flu,
                  family = "gaussian")
low_pred   <- predict(low_model)

summary(low_model)

high_model <- gam(Mortality ~ Year + s(Week, k = w, sp=10000),
                  data   = flu,
                  family = "gaussian")
high_pred  <- predict(high_model)

summary(high_model)

df2 <- data.frame(Time  = flu$Time,
                  Mortality = flu$Mortality,
                  pred1 = low_pred,
                  pred2 = high_pred)

ggplot(df2, aes(x = Time))+
  geom_line(aes(y = Mortality, colour="Actual Observations"))+
  geom_line(aes(y = pred1, colour="Prediction with low penalty"))+
  geom_line(aes(y = pred2, colour="Prediction with high penalty"))+
  scale_colour_manual("Legend",
                      breaks = c("Actual Observations", "Prediction with low penalty", "Prediction with
                      values = c("#FF3366","#0071B3","#00EEFF"))+
  theme_minimal()
```

```r
par(mfrow = c(1,2))
plot(low_model)
plot(high_model)

ggplot(df1)+
  geom_line(aes(x = Time, y = Influenza, color = "Influenza")) +
  geom_line(aes(x = Time, y = Residuals, color = "Residuals")) +
  scale_color_manual("Legend",
                     breaks = c("Influenza", "Residuals"),
                     values = c("#009A73", "#989898"))+
  theme_minimal()

y <- length(unique(flu$Year))
f <- length(unique(flu$Influenza))

flu_model <- gam(Mortality ~ s(Year, k = y) + s(Week, k = w) + s(Influenza, k = f),
                 data   = flu,
                 family = "gaussian",
                 method = "GCV.Cp")

flu_pred  <- predict(flu_model)

summary(flu_model)

par(mfrow=c(1,3))
plot(flu_model, residuals = TRUE, cex = 2)

df3 <- data.frame(Time = flu$Time,
                  Mortality  = flu$Mortality,
                  Prediction = flu_pred,
                  Influenza  = flu$Influenza,
                  Residuals  = flu_model$residuals)

ggplot(df3)+
  geom_line(aes(x = Time, y = Mortality, color = "Observed Mortality")) +
  geom_line(aes(x = Time, y = Prediction, color = "Predicted Mortality")) +
  scale_color_manual("Legend",
                     breaks = c("Observed Mortality", "Predicted Mortality"),
                     values = c("#FF3366", "#0071B3"))+
  theme_minimal()

library(pamr)
library(glmnet)
library(kernlab)

data <- read.csv2("Data/data.csv", check.names = FALSE)
data$Conference <- as.factor(data$Conference)

n     <- dim(data)[1]
set.seed(12345)
ind   <- sample(1:n, floor(n*0.7))
train <- data[ind,]
test  <- data[-ind,]
```

```r
#train
rownames(train) <- 1:nrow(train)
x_train         <- t(train[,-4703]) # remove dependent variable
y_train         <- train[[4703]]    # vector of the dependent variable
mytrain_data    <- list(x = x_train,
                        y = y_train,
                        geneid    = as.character(1:nrow(x_train)),
                        genenames = rownames(x_train))
#test
rownames(test) <- 1:nrow(test)
x_test         <- t(test[,-4703])
y_test         <- test[[4703]]


cen_model   <- pamr.train(mytrain_data)

set.seed(12345)
cvmodel     <- pamr.cv(cen_model, mytrain_data)

#print(cvmodel)
pamr.plotcv(cvmodel)

pamr.plotcen(cen_model,
             mytrain_data,
             threshold = cvmodel$threshold[which.min(cvmodel$error)])

features = pamr.listgenes(cen_model,
                  mytrain_data,
                  threshold = cvmodel$threshold[which.min(cvmodel$error)],
                  genenames = TRUE)

nrow(features)

as.matrix(features[1:10,2])

#cat(paste(colnames(data)[as.numeric(features[,1])], collapse='\n'))
# top10 <- as.matrix(colnames(data)[as.numeric(features[1:10,1])])
# top10

cen_pred <- pamr.predict(cen_model,
                         newx = x_test,
                         type = "class",
                         threshold = cvmodel$threshold[which.min(cvmodel$error)])

cen_mat   <- table(y_test, cen_pred)
cen_rate  <- 1 - sum(diag(cen_mat)) / sum(cen_mat)
cen_rate

res1 <- list("Error Rate" = cen_rate, "Features Selected" = nrow(features))

set.seed(12345)
elastic_cv <- cv.glmnet(x = t(x_train),
                        y = y_train,
```

```
                        family="binomial",
                        alpha = 0.5)

# par(mfrow = c(2,1))
# plot(elastic_cv)
# plot(elastic_cv$glmnet.fit)

elastic_pred <- predict.cv.glmnet(elastic_cv,
                                  newx = t(x_test),
                                  s = elastic_cv$lambda.min,
                                  type = "class",
                                  exact = TRUE)

elastic_mat  <- table(y_test, elastic_pred)
elastic_rate <- 1 - sum(diag(elastic_mat)) / sum(elastic_mat)

coefs <-as.matrix(coef(elastic_cv, elastic_cv$lambda.min))
elastic_features <- length(names(coefs[coefs != 0,]))

res2 <- list("Error Rate" = elastic_rate, "Features Selected" = elastic_features)


invisible(capture.output(
  svm <- ksvm(Conference ~ .,
              data = train,
              kernel="vanilladot",
              scaled = FALSE)))


svm_pred <- predict(svm, newdata = test)

svm_mat  <- table(y_test, elastic_pred)
svm_rate <- 1 - sum(diag(svm_mat)) / sum(svm_mat)

res3 <- list("Error Rate" = svm_rate, "Features Selected" = svm@nSV)

result <- rbind("NSC" = res1, "Elastic Net" = res2, "SVM" = res3)
knitr::kable(result)

hochberg <- function(x, y, alpha) {
  p <- apply(x, 2, function(x_data){t.test(x_data ~ y, alternative = "two.sided")$p.value})

  rank     <- as.matrix(sort(p))
  l        <- length(p)
  values   <- (1:l/l) * alpha
  T_F      <- matrix(0,4702,1)
  z        <- data.frame("P-Values" = rank,"T_F" = T_F)

  for(i in 1:4702){
    if(rank[i] <= values[i]){
      z[i,2] <- "Rejected"
    }
    else{z[i,2] <- "Accepted"}
```

```
  }
  lowest_p <- subset(z, T_F == "Rejected")
  return(lowest_p)
}

lowest_p <- hochberg(x = data[,-4703], y = data[,4703], alpha=0.05)

lowest_p
```