723A95 - Machine Learning
Lab 2 Block 2
Report

Thi Pham (thiph169)
Fanny Karelius (fanka300)

18 December 2017

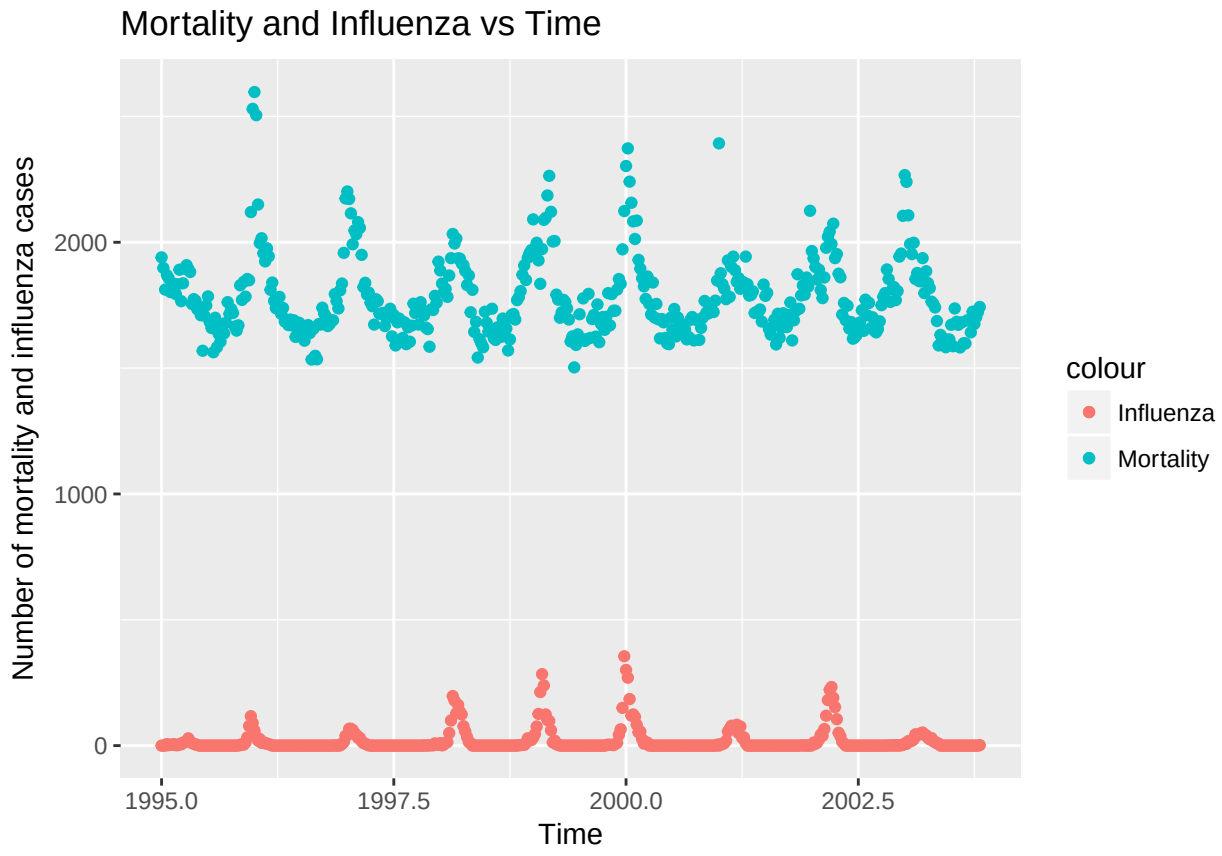
Assignment 1 was contributed by Thi Pham and Assignment 2 was contributed by Fanny Karelius.

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

In this assignment, the *influenza.xlsx* was given, which contains weekly data on the mortality and the number of laboratory-confirmed cases of influenza in Sweden.

1.1

To inspect how the mortality influenza number vary with time, a time-series was plotted.



From the plot, it can be seen that there was a positive correlation between influenza numbers and mortality across time.

1.2

In this task, the *gam* function was used to fit a GAM model in which *Mortality* is normally distributed and modelled as a linear function of *Year* and spline function Of *Week*. The probablistic model is:

$$Y_M = b_0 + b_1(Year) + s(Week) + \epsilon$$

```
## Intercept = -680.598
```

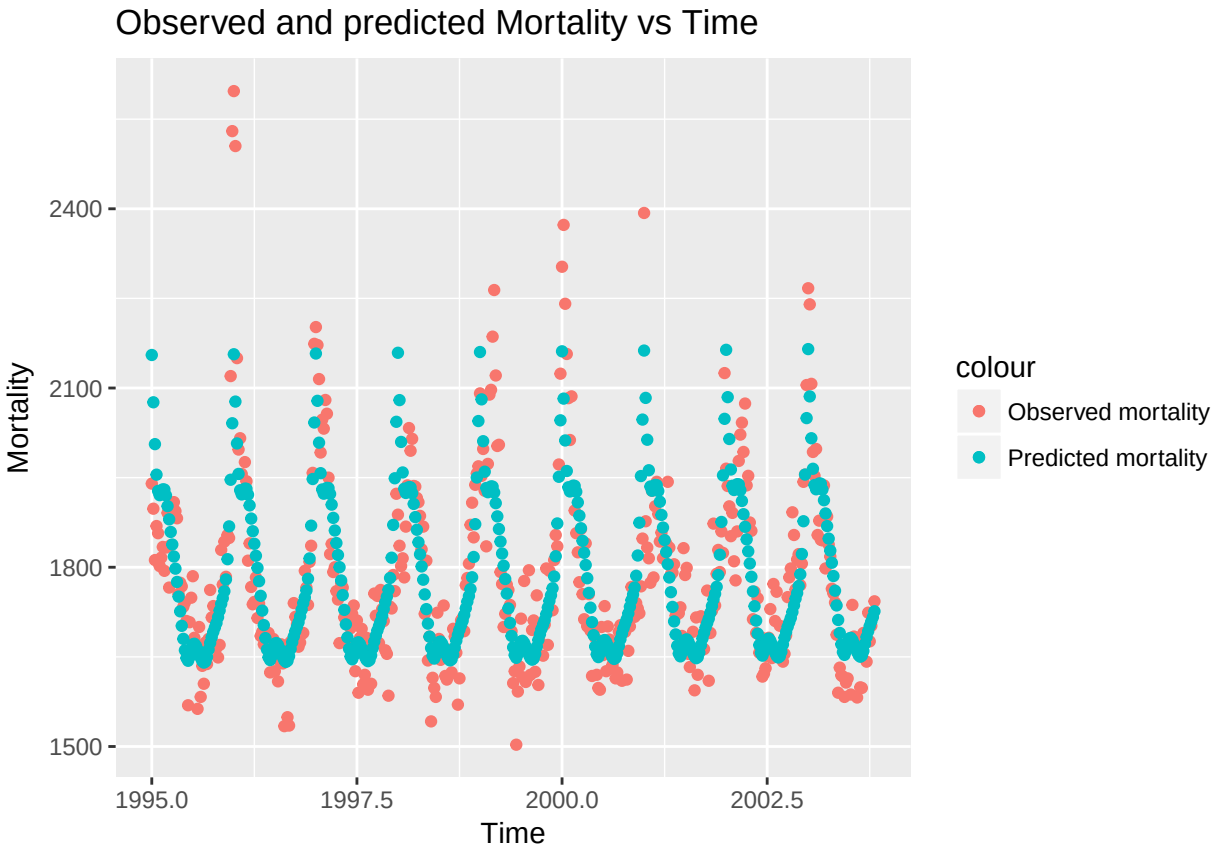
```
## Year coefficent = 1.232846
```

The probabilistic model for given information is:

$$Y_M = -680.598 + 1.23(Year) + s(Week) + \epsilon$$

1.3

In this section, the predicted and observed mortality against time for the fitted model was plotted.



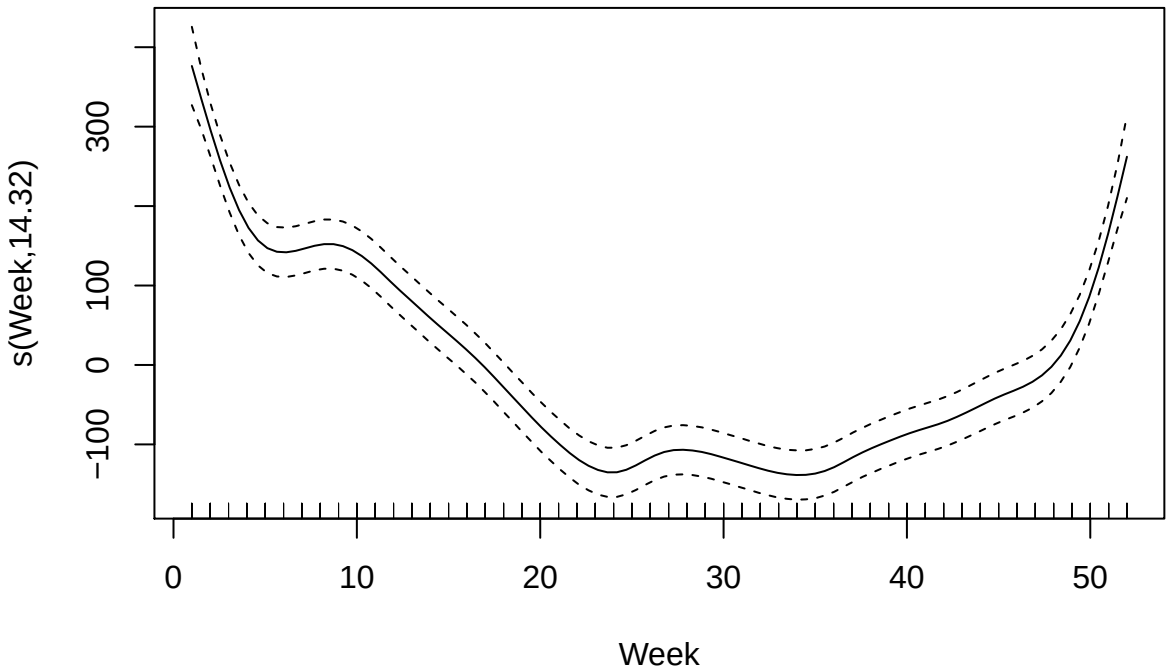
It can be seen clearly that the predicted values follows quite closely with observed values, therefore it fits quite well with the data.

The output of the GAM model is below:

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

It can be seen that the coefficients of *Intercepts* and *Year* are not significant. Only *Week* are taken into account. Therefore there is no trend that mortality change from one year to another.

The plot of spline components *Week* below.



It can be seen that the mortality numbers is changing throughout the year. It is higher in the first few weeks

of the year (which are winter), then it decreases to its lowest around week 33 (which are summer), then it increases again to the end of a year. There exists a seasonally pattern of *Mortality* numbers every year and the same weeks each year since the *Year* component is not significant in this model.

1.4

The task was to examine how the penalty factor of the spline function in the GAM model from step 2 influences the estimated deviance of the model. The choosen values of penalty factors were 0.1 and 100. The plots presented below

From the plot above, the bigger the smoothing parameter, the lower the variance of the model is. The lower penalty factor fits better than higher penalty factor.

The relationship between the penalty factor and the degrees of freedom is:

$$df_\lambda = \sum_{k=1}^{N} \frac{1}{1+\lambda d_k}$$

Therefore, the higher the lambda, the smaller the degrees of freedom, hence higher the penalty factor.

The summary of two models are below:

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

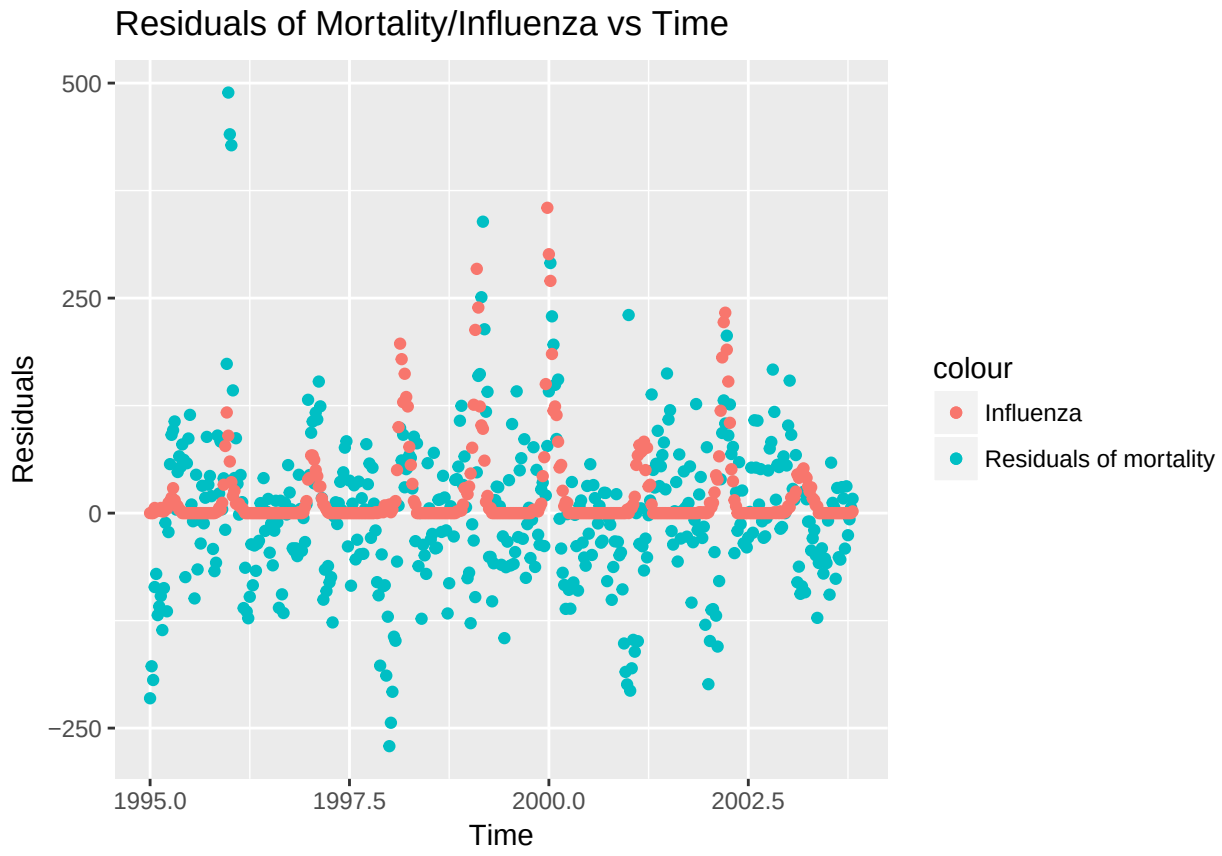
```
## Mortality ~ Year + s(Week, k = k_week)
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2789.6277  5381.0481   0.518    0.604
## Year          -0.5032     2.6920  -0.187    0.852
##
## Approximate significance of smooth terms:
##           edf Ref.df     F p-value
## s(Week) 1.007  1.014 95.05  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.172   Deviance explained = 17.6%
## GCV =  21643  Scale est. = 21501     n = 459
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Mortality ~ Year + s(Week, k = k_week)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -220.669   3677.375  -0.060    0.952
## Year           1.003      1.840   0.545    0.586
##
## Approximate significance of smooth terms:
##           edf Ref.df     F p-value
## s(Week) 2.638  3.297 200.4  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.614   Deviance explained = 61.7%
## GCV =  10128  Scale est. = 10025     n = 459
```

As show in the *summary* above, the penalty factor are 0.1 and 100 and the degrees of freedom are 2.638 and 1.007 respectively. Thus, the results I got confirm the relationship between the penalty factor and the degrees of freedom.

1.5

The residuals of the fitted model and the *Influenza* values against *Time* were plotted



There is clearly a pattern in the resuduals correlated to the outbreaks of influenza.

1.6

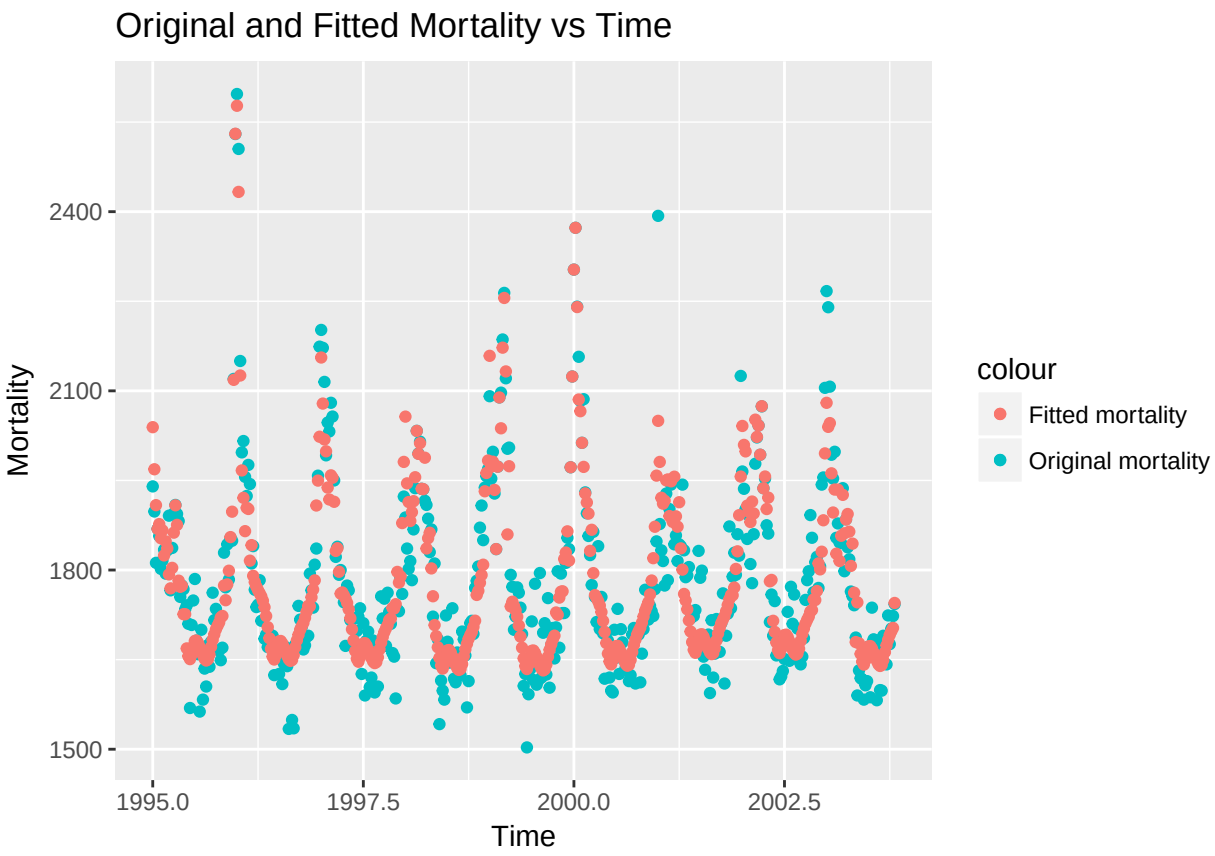
In this task, the GAM model is fitted which *Mortality* is be modelled as an additive function of the spline functions of *Year*, *Week* and number of confrimed cases of influenza.

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Mortality ~ s(Year, k = k_year) + s(Week, k = k_week) + s(Influenza,
##     k = k_influenza)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1783.8        3.2   557.5   <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)      4.663  5.677  1.487   0.181
## s(Week)     14.641 18.248 18.533  <2e-16 ***
## s(Influenza) 69.740 72.833  5.600  <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 = 5846.7  Scale est. = 4699.8    n = 459
```

In this model, the components of *Year*, *Week* and *Influenza* are all significant. Therefore the *Mortality* is influenced by the outbreaks of influenza.

Plot of the original and fitted *Mortality* against *Time* below:



It is clearly that this model fits the data really well compare to previous models.

Assignment 2: High-dimensional methods

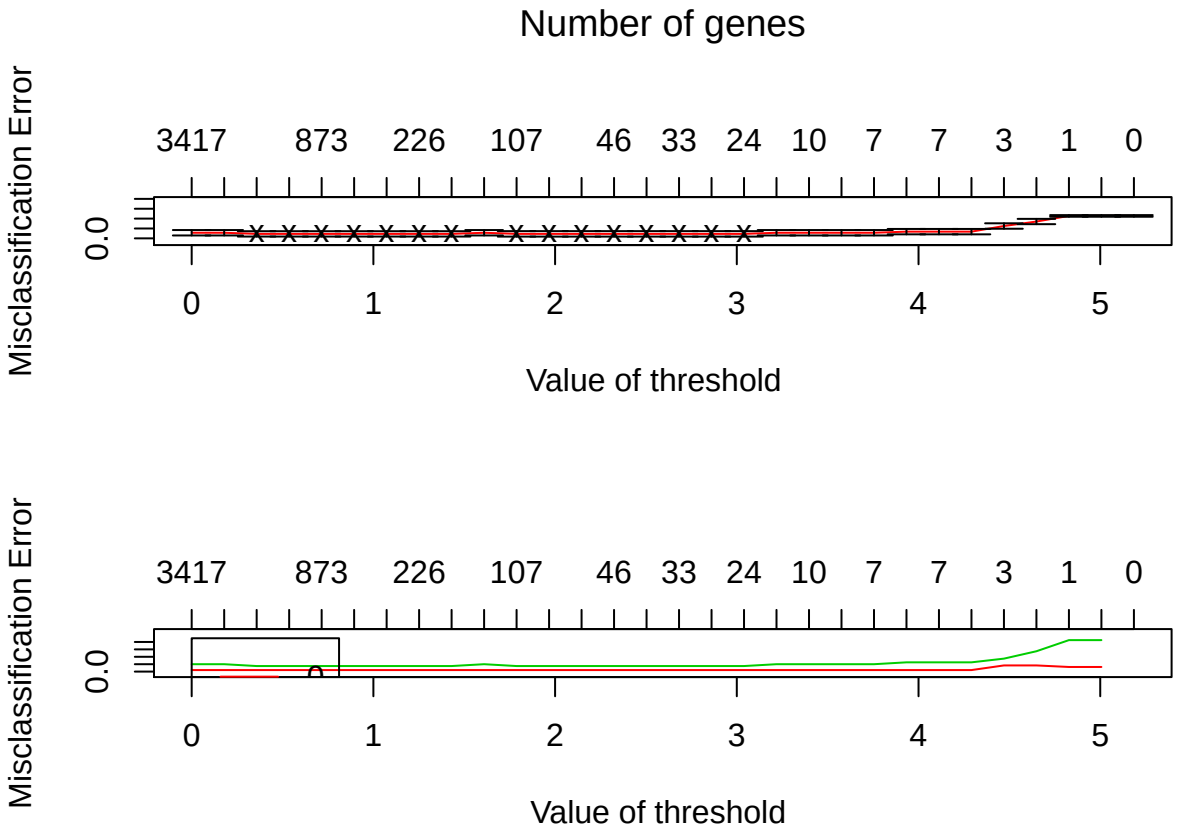
2.1 Nearest shrunken centroid

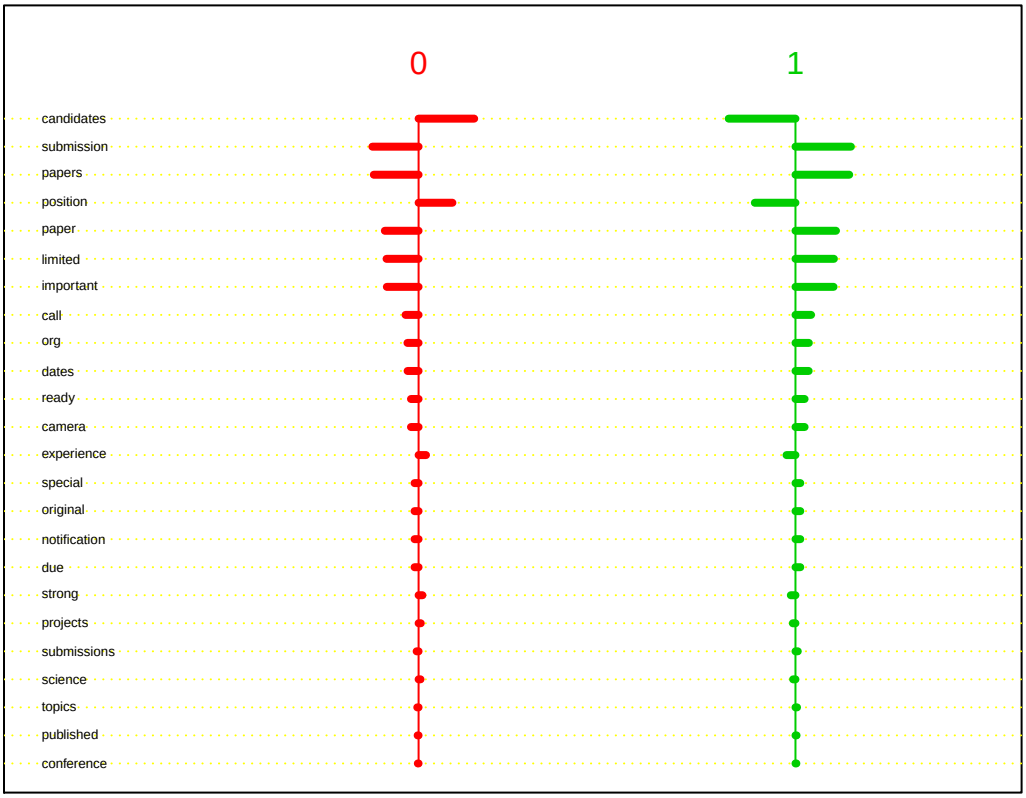
The centroid plot below shows the significance of each feature in classifying an e-mail as Conference (1) or everything else (0).

```
##  [1] 1 0 0 1 1 1 0 0 0 0 1 0 0 1 1 0 0 1 0 1 0 1 0 0 0 1 1 0 0 0 0 0 0 1 1 1
## [36] 1 1 0 0 1 1 0 0 1
## Levels: 0 1
```

```
## 12345678910111213141516171819202122232425262728293O
```

```
## 12Fold 1 :12345678910111213141516171819202122324252627282930
## Fold 2 :12345678910111213141516171819202122324252627282930
## Fold 3 :12345678910111213141516171819202122324252627282930
## Fold 4 :12345678910111213141516171819202122324252627282930
## Fold 5 :12345678910111213141516171819202122324252627282930
## Fold 6 :12345678910111213141516171819202122324252627282930
## Fold 7 :12345678910111213141516171819202122324252627282930
## Fold 8 :12345678910111213141516171819202122324252627282930
## Fold 9 :12345678910111213141516171819202122324252627282930
## Fold 10 :12345678910111213141516171819202122324252627282930
```






```
##       id   0-score 1-score
##  [1,] 607  0.2953  -0.3543
##  [2,] 4060 -0.2452 0.2943
##  [3,] 3036 -0.2377 0.2852
##  [4,] 3187 0.1797  -0.2156
##  [5,] 3035 -0.1791 0.215
##  [6,] 2463 -0.1701 0.2041
##  [7,] 2049 -0.1685 0.2022
##  [8,] 596  -0.0691 0.0829
##  [9,] 2974 -0.0592 0.071
## [10,] 1045 -0.0584 0.0701
## [11,] 599  -0.0404 0.0485
## [12,] 3433 -0.0404 0.0485
## [13,] 1501 0.0396  -0.0475
## [14,] 1262 -0.0214 0.0257
## [15,] 2889 -0.0214 0.0257
## [16,] 2990 -0.0214 0.0257
## [17,] 3952 -0.0214 0.0257
## [18,] 4039 0.0206  -0.0248
## [19,] 3312 0.0118  -0.0142
## [20,] 4061 -0.0104 0.0125
## [21,] 3725 0.0104  -0.0125
## [22,] 4282 -0.0071 0.0085
## [23,] 3364 -0.005  0.006
## [24,] 869  -0.0041 0.0049

## 10 most contributing features:  candidates submission papers position paper limited important call o
```

```
## Misclassification rate for test data:  0.1
```

24 features were selected by the model and the 10 most contributing features (listed above) seem reasonable since they are words that may very well be in an e-mail announcing a conference.

2.2

a) Elastic net

```
## Misclassification rate for test data:  0.2
```

```
## Number of selected features:  26
```

b) SVM

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

```
## Misclassification rate for test data:  0.15
```

```
## Number of selected features:  43
```

```
## Comparison of models:
```

```
##         Test.error Features
## NSC           0.10       24
## Elastic       0.20       26
## SVM           0.15       43
```

Comparing the test errors in the table above, the nearest shrunken centroid model is the best fit since it has the lowest test error. Nearest shrunken centroid also selected the fewest number of features, which makes for a simpler model. Therefore, we would prefer nearest shrunken centroid.

2.3 Benjamini-Hochberg method

```
##         features     p.values
## 1         papers 1.116910e-10
## 2     submission 7.949969e-10
## 3       position 8.219362e-09
## 4      published 1.835157e-07
## 5      important 3.040833e-07
## 6           call 3.983540e-07
## 7     conference 5.091970e-07
## 8     candidates 8.612259e-07
## 9          dates 1.398619e-06
## 10         paper 1.398619e-06
## 11        topics 5.068373e-06
## 12       limited 7.907976e-06
## 13     candidate 1.190607e-05
## 14        camera 2.099119e-05
## 15         ready 2.099119e-05
## 16       authors 2.154461e-05
## 17           phd 3.382671e-05
## 18      projects 3.499123e-05
## 19           org 3.742010e-05
## 20        chairs 5.860175e-05
```

```
## 21           due 6.488781e-05
## 22      original 6.488781e-05
## 23  notification 6.882210e-05
## 24        salary 7.971981e-05
## 25        record 9.090038e-05
## 26        skills 9.090038e-05
## 27          held 1.529174e-04
## 28          team 1.757570e-04
## 29         pages 2.007353e-04
## 30      workshop 2.007353e-04
## 31     committee 2.117020e-04
## 32   proceedings 2.117020e-04
## 33         apply 2.166414e-04
## 34        strong 2.246309e-04
## 35 international 2.295684e-04
## 36        degree 3.762328e-04
## 37     excellent 3.762328e-04
## 38          post 3.762328e-04
## 39     presented 3.765147e-04
```

When we run a statistical test with $\alpha = 0.05$ there is a 5% chance that a feature will be significant eventhough it is not (false positive, type I error). With the Benjamini-Hochberg method the chance of getting a false positive is reduced. The 39 features selected as signficant above are less likely to be a result of a type I error than if Benjamini-Hochberg had not been used.

3. Appendix

3.1 R code for assignment 1

```
####Assignment 1#####

#Reading data
influenzaData <- read.csv("Influenza.csv")

#1.1
library(ggplot2)

ggplot(influenzaData) +
  geom_point(aes(x = Time, y = Mortality, color = "Mortality")) +
  geom_point(aes(x = Time, y = Influenza, color = "Influenza")) +
  ggtitle("Mortality and Influenza vs Time") +
  ylab("Number of mortality and influenza cases")

#1.2
library(mgcv)

k_week = length(unique(influenzaData$Week))
gm_model <- gam(Mortality ~ Year + s(Week, k = k_week), data = influenzaData, method = "GCV.Cp")

cat("Intercept =" , coef(gm_model)["(Intercept)"], "\n")
cat("Year coefficent =" , coef(gm_model)["Year"])
```

```r
#1.3

pred_mortality <- predict(gm_model, newdata = influenzaData)

ggplot(influenzaData) +
  geom_point(aes(x = Time, y = Mortality, color = "Observed mortality")) +
  geom_point(aes(x = Time, y = pred_mortality, color = "Predicted mortality")) +
  ggtitle("Observed and predicted Mortality vs Time")

#output of GAM model
gm_model$coefficients

summary(gm_model)

#plot the spline component
plot(gm_model)

#1.4

#very low and high penelaty factors

gm_model_low_sp <- gam(Mortality ~ Year + s(Week, k = k_week), data = influenzaData, sp = 0.1, method="
coef_low_sp <- gm_model_low_sp$coefficients
pred_low_sp <- predict(gm_model_low_sp, newdata = influenzaData)

gm_model_high_sp <- gam(Mortality ~ Year + s(Week, k = k_week), data = influenzaData, sp = 100, method=
coef_high_sp <- gm_model_high_sp$coefficients
pred_high_sp <- predict(gm_model_high_sp, newdata = influenzaData)

#plot

ggplot(influenzaData) +
  geom_point(aes(x = Time, y = Mortality, color = "Mortality")) +
  geom_point(aes(x = Time, y = pred_low_sp, color = "Sp = 0.1")) +
  geom_point(aes(x = Time, y = pred_high_sp, color = "Sp = 100")) +
  ggtitle("Observed, High/Low Penalty Factors vs time")

summary(gm_model_low_sp)
summary(gm_model_high_sp)

#1.5

#plot residuals and influenze values agaist time

ggplot(influenzaData) +
  geom_point(aes(x = Time, y = gm_model$residuals, color = "Residuals of mortality")) +
  geom_point(aes(x = Time, y = Influenza, color = "Influenza")) +
  ggtitle("Residuals of Mortality/Influenza vs Time") +
  ylab("Residuals")

#1.6

k_year <- length(unique(influenzaData$Year))
```

```r
k_week <- length(unique(influenzaData$Week))
k_influenza <- length(unique(influenzaData$Influenza))
gam_mortality_additive <- gam(Mortality ~ s(Year, k = k_year) +
                          s(Week, k = k_week) +
                          s(Influenza, k = k_influenza),
                          data = influenzaData, method="GCV.Cp")
summary(gam_mortality_additive)

#plot
fitted_val <- gam_mortality_additive$fitted.values
ggplot(influenzaData) +
  geom_point(aes(x = Time, y = Mortality, color = "Original mortality")) +
  geom_point(aes(x = Time, y = fitted_val, color = "Fitted mortality")) +
  ggtitle("Original and Fitted Mortality vs Time")

################################################
```

3.2 R code for assignment 2

```r
####High-dimensional methods####
# Assignment 2

# Read in and prepare data
data0 <- read.csv2("data.csv",fileEncoding="iso-8859-15")
data <- as.data.frame(data0)
data$Conference<-as.factor(data$Conference)

# Split into training and test data
set.seed(1234567890)
n <- nrow(data)
indices <- sample(1:n, floor(0.7*n))
train <- data[indices,]
test <- data[-indices,]

#1 Nearest shrunken centroid
library(pamr)

# Prepare data for fitting
x<-t(train[,-which(colnames(train)=="Conference")])
y<-train[[which(colnames(train)=="Conference")]]
train$Conference
mydata <- list(x=x, y=as.factor(y), geneid=as.character(1:nrow(x)), genenames=rownames(x))

# Fit model and do cross-validation
data_fit <- pamr.train(mydata)
fit_cv <- pamr.cv(data_fit, mydata)
pamr.plotcv(fit_cv)

# Choose threshold that produce smallest error
min_thresholds<-which(fit_cv$error==min(fit_cv$error)) #3-9,11-18 best threshold
chosen_threshold<-fit_cv$threshold[max(min_thresholds)] #choose threshold with fewest genes
```

```r
# Plot centroids and list contributing features
pamr.plotcen(data_fit,mydata, threshold = chosen_threshold)
pamr.listgenes(data_fit, mydata, threshold = chosen_threshold)

# Find the most contributing features
fit_list<-pamr.listgenes(data_fit, mydata, threshold=chosen_threshold)[1:10,]
results <- as.numeric(fit_list[,1])
most_contributing<-colnames(train[results])
most_contributing
#yes these seem reasonable, especially papers, important, candidates, dates

# Test error
x_test <- t(test[,-which(colnames(test)=="Conference")])
pred <- pamr.predict(data_fit, x_test, threshold=chosen_threshold) #predicitons

# Function that creates confusion matrix and returns misclassification rate
test_error <- function(testdata, predictions){
  conf_mat<-table(True=testdata, Pred=predictions) #Confusion matrix
  misclass_rate<- 1 - (sum(diag(conf_mat))/sum(conf_mat)) #misclassification rate
  list(conf_mat, misclass_rate)
}

cat("Misclassification rate for test data: ", nsc_error<-test_error(test$Conference, pred)[[2]])

#pamr.confusion(data_fit, threshold=chosen_threshold, extra = TRUE) #training error

# 2
library(glmnet)
# a) Elastic net, alpha=0.5

# Fit elastic net model
set.seed(1234567890)
elastic_fit <- cv.glmnet(x = t(x), y=y, family = "binomial", alpha=0.5, type.measure = "class")
elastic_model <- glmnet(x=t(x), y=y, family="binomial", alpha=0.5, lambda = elastic_fit$lambda.min)

# Predictions and misclassification rate
pred_el <- as.numeric(predict(elastic_model, t(x_test), type="class"))
cat("Misclassification rate for test data: ", el_error<-test_error(test$Conference, pred_el)[[2]])

# Contributing features
feature_coef <- as.matrix(coef(elastic_fit, elastic_fit$lambda.min)) #coefficients of final model
num_features<-length(feature_coef[feature_coef!=0]) #selected features

cat("Number of selected features: ", num_features)

# b) Support vector machine with vanilladot kernel
library(kernlab)
set.seed(1234567890)

# Fit model
svm_fit <- ksvm(Conference~. , data=train, kernel="vanilladot")
svm_num<- length(coef(svm_fit)[[1]]) #number of selected features
```

```r
# Predict and compute test error
pred_svm <- predict(svm_fit, t(x_test))
svm_error<-test_error(test$Conference, pred_svm)[[2]]

cat("Misclassification rate for test data: ", svm_error)
cat("Number of selected features: ", svm_num)

# Compare models
comp_table<-data.frame("Test error"=c(nsc_error, el_error, svm_error), Features=c(24, num_features, svm
rownames(comp_table)<-c("NSC", "Elastic", "SVM")
comp_table

# 3 Benjamini-Hochberg

p_values <- vector(length = (length(names(data0))-1))
for(i in 1:(length(names(data0))-1)){
  formula <- as.formula(paste(names(data0)[i],"~ Conference"))
  p_values[i]<- t.test(formula = formula, data = data0, alternative = "two.sided")$p.value
}
p_values_ordered<-p_values[order(p_values, decreasing = FALSE)]
a<-0.05 #alpha value, 95%
critical_values <- a*c(1:length(p_values))/length(p_values)
comparison <- vector(length=length(p_values))

for(i in 1:length(p_values)){
  if(p_values_ordered[i]<critical_values[i]){
    comparison[i]<-p_values_ordered[i]
  }
  else{
    comparison[i]<-NA
  }
}

results_df <- data.frame(features = names(data0[order(p_values)]), "p values" = p_values_ordered, compa
results_df<-results_df[which(is.na(results_df$comparison)==FALSE),1:2]
results_df
#Reference: http://www.statisticshowto.com/benjamini-hochberg-procedure/
##############################
```