machine learning(732A99) lab2 block2

Anubhav Dikshit(anudi287)

17 December 2018

Contents

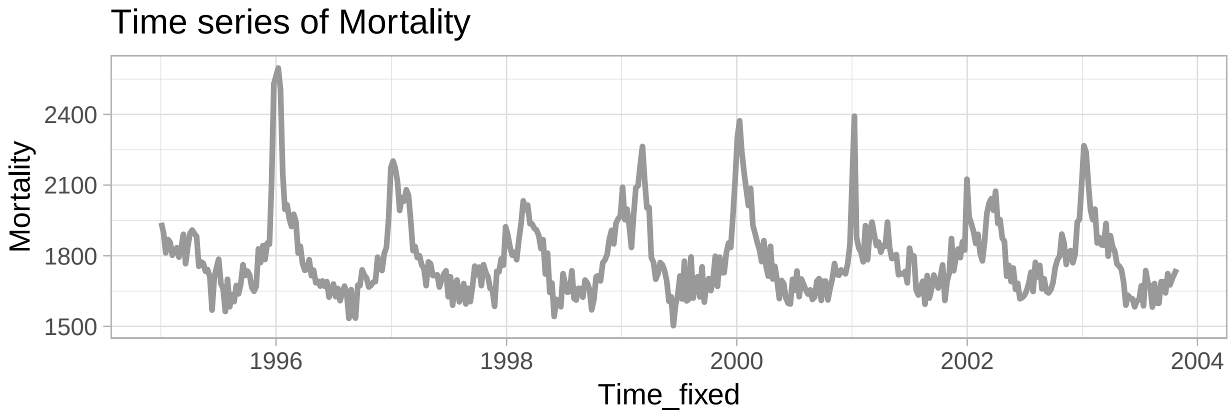
Assignment 1

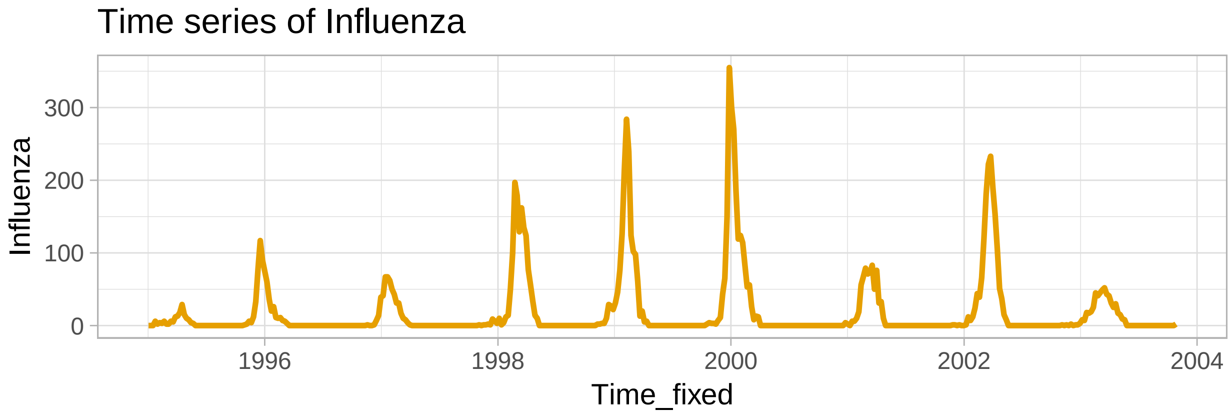
Loading The Libraries

```
if (!require("pacman")) install.packages("pacman")
pacman::p_load(xlsx, ggplot2, tidyr, dplyr, reshape2, gridExtra,
               mgcv, rgl, akima, pamr, caret, glmnet, kernlab)

set.seed(12345)
options("jtools-digits" = 2, scipen = 999)

# colours (colour blind friendly)
cbPalette <- c("#999999", "#E69F00", "#56B4E9", "#009E73", "#F0E442", "#0072B2",
               "#D55E00", "#CC79A7")

## Making title in the center
theme_update(plot.title = element_text(hjust = 0.5))
```

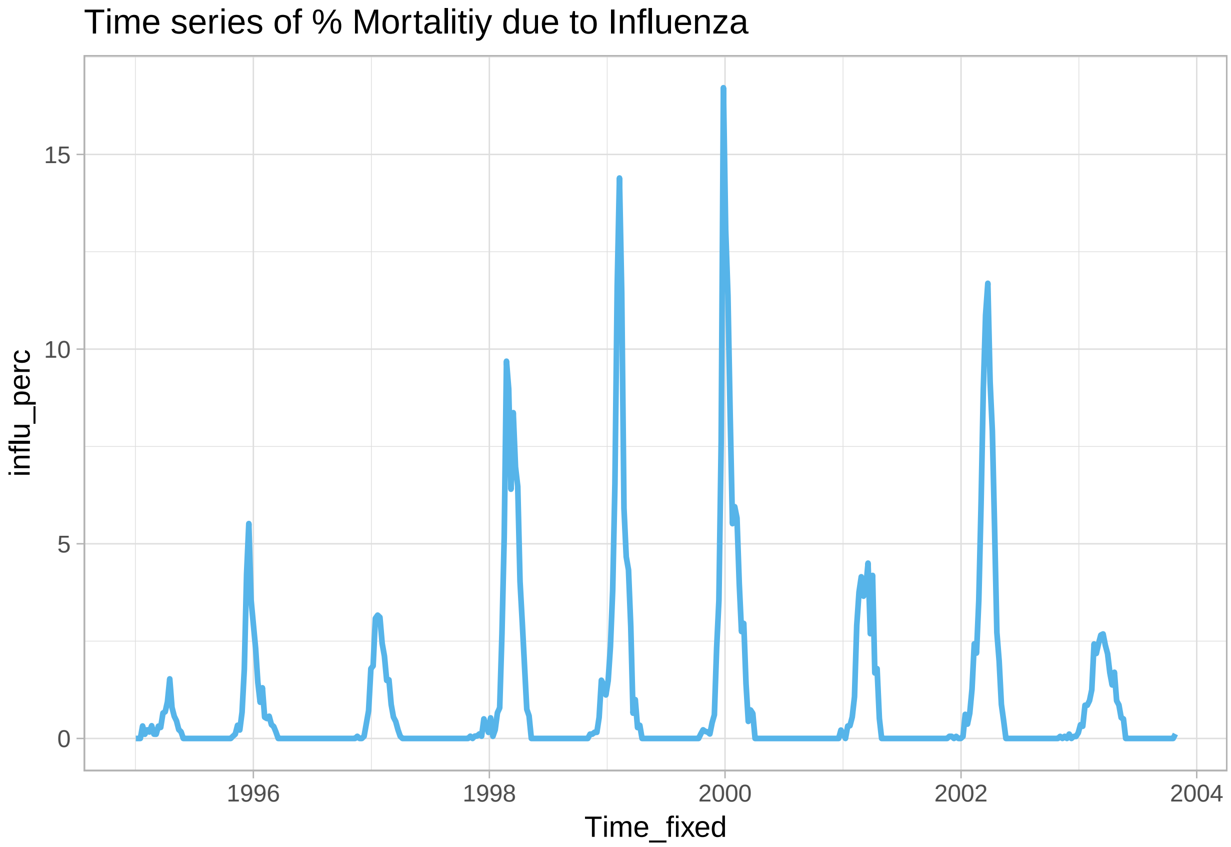
1. Use time series plots to visually inspect how the mortality and influenza number vary with time (use Time as X axis). By using this plot, comment how the amounts of influenza cases are related to mortality rates.

```
set.seed(12345)

# Importing data
flu_data = read.xlsx("influenza.xlsx", sheetName = "Raw data")
flu_data$Time_fixed <- as.Date(paste(flu_data$Year, flu_data$Week, 1, sep="-"), "%Y-%U-%u")
flu_data$influ_perc <- (flu_data$Influenza/flu_data$Mortality) * 100

# Plot

p1 <- ggplot(flu_data, aes(x=Time_fixed, y = Mortality)) +
  geom_line(color = "#999999", size = 1) +
    scale_fill_brewer() +
      theme_light() +
  ggtitle("Time series of Mortality")

p2 <- ggplot(flu_data, aes(x=Time_fixed, y = Influenza)) +
  geom_line(color = "#E69F00", size = 1) +
      scale_fill_brewer() +
      theme_light() +
  ggtitle("Time series of Influenza")

p3 <- ggplot(flu_data, aes(x=Time_fixed, y = influ_perc)) +
  geom_line(color = "#56B4E9", size = 1) +
      scale_fill_brewer() +
      theme_light() +
  ggtitle("Time series of % Mortalitiy due to Influenza")

gridExtra::grid.arrange(p1, p2, ncol=1)
```


Time series of Mortality


Time series of Influenza

```
p3
```



Analsis: From the plots is we can defintely see that Influenza and Mortalitiy in the given dataset are in sync, everytime Mortality peaks so does influenza, however the magnitiude of peaking is not in sync, that is the highest cases of mortaility were observed in '1996' while for influenza its in year '2000'.

From the third plot, we can see the percentage of mortalitiy due to influenza, here also the peaks match with the other plots, suggests that these two events are closely correleated.

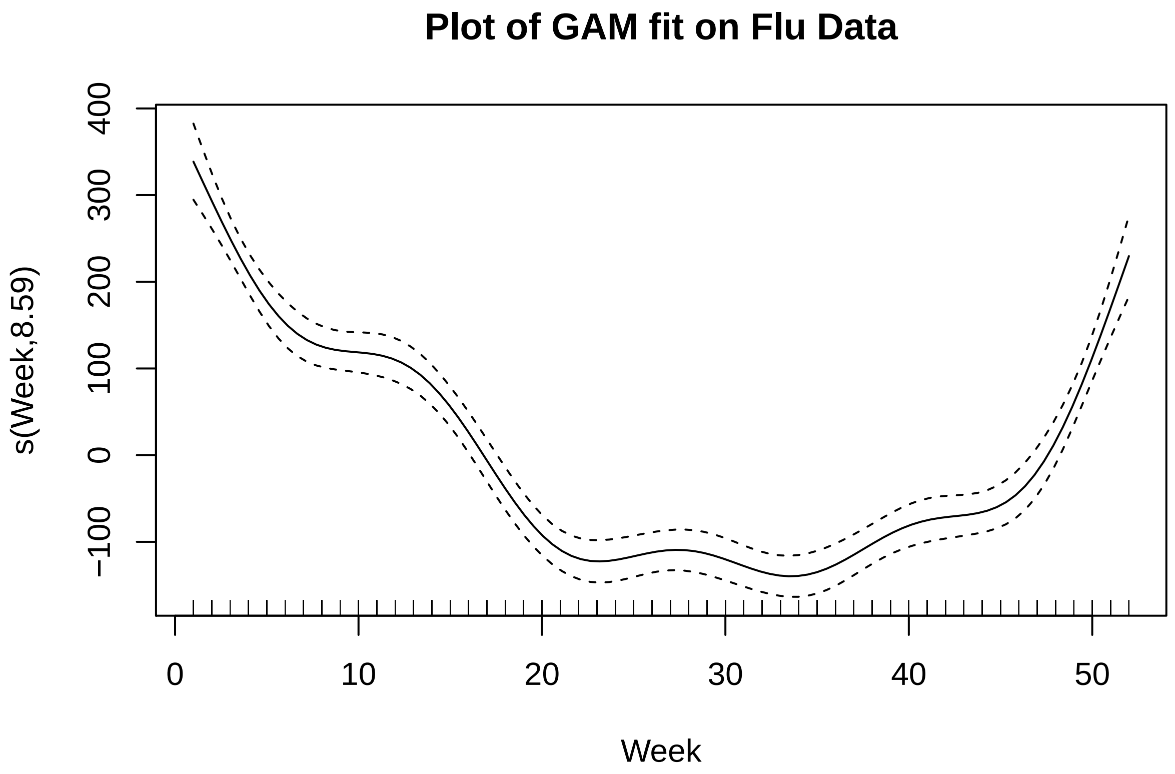
2. Use gam() function from mgcv package to fit a GAM model in which Mortality is normally distributed and modelled as a linear function of Year and spline function of Week, and make sure that the model parameters are selected by the generalized cross-validation. Report the underlying probabilistic model.

```
gam_model <- mgcv::gam(data = flu_data, Mortality~Year+s(Week), method = "GCV.Cp")
summary(gam_model)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Mortality ~ Year + s(Week)
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -652.060   3448.379  -0.189     0.85
## Year           1.219      1.725   0.706     0.48
```

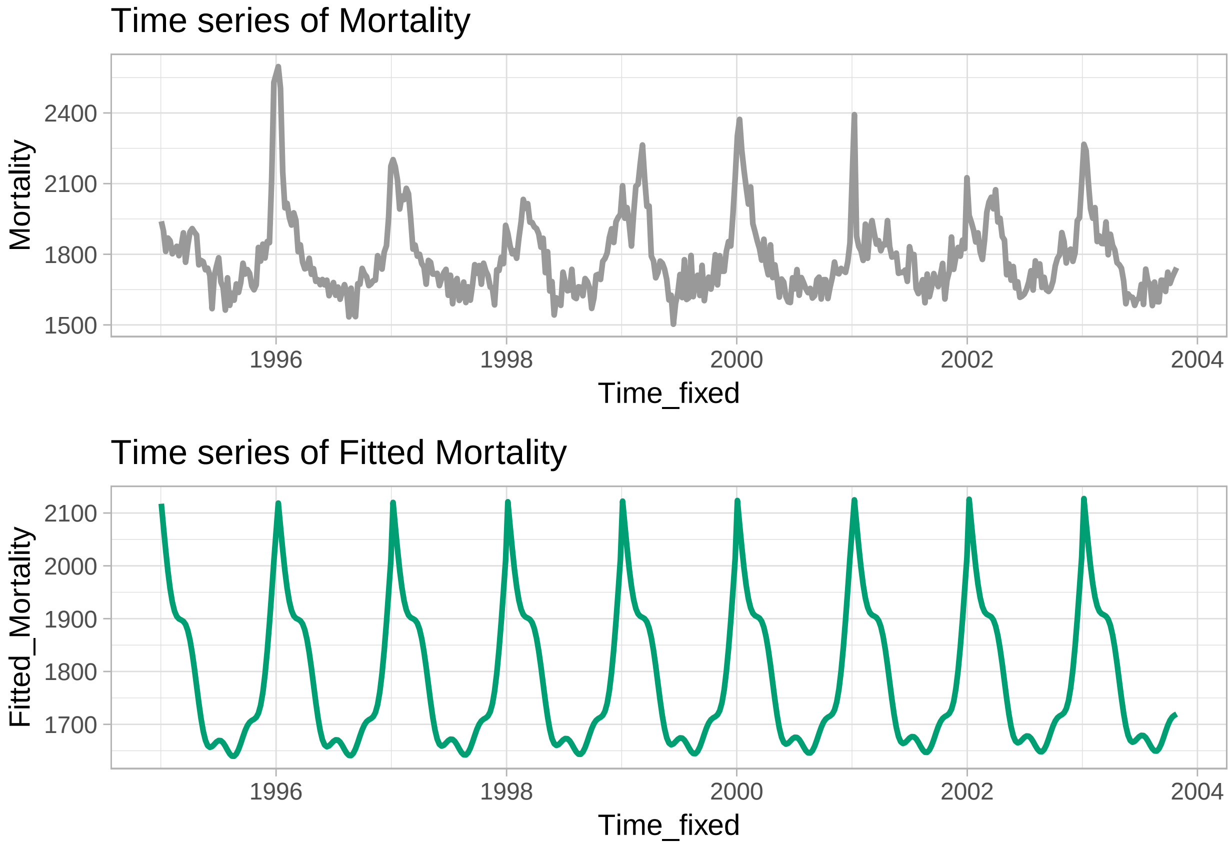
```
##
## Approximate significance of smooth terms:
##           edf Ref.df     F              p-value
## s(Week) 8.587  8.951 100.3 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.661   Deviance explained = 66.8%
## GCV = 9014.6  Scale est. = 8806.7    n = 459
```

```
#plot the fit

p4 <- plot(gam_model, main= "Plot of GAM fit on Flu Data")
```



3. Plot predicted and observed mortality against time for the fitted model and comment on the quality of the fit. Investigate the output of the GAM model and report which terms appear to be significant in the model. Is there a trend in mortality change from one year to another? Plot the spline component and interpret the plot.

```
temp <- flu_data
temp$Fitted_Mortality <- gam_model$fitted.values

p5 <- ggplot(data=temp, aes(x = Time_fixed, y = Fitted_Mortality)) +
   geom_line(color = "#009E73", size = 1) +
    scale_fill_brewer() +
      theme_light() +
  ggtitle("Time series of Fitted Mortality")

grid.arrange(p1, p5, nrow = 2)
```

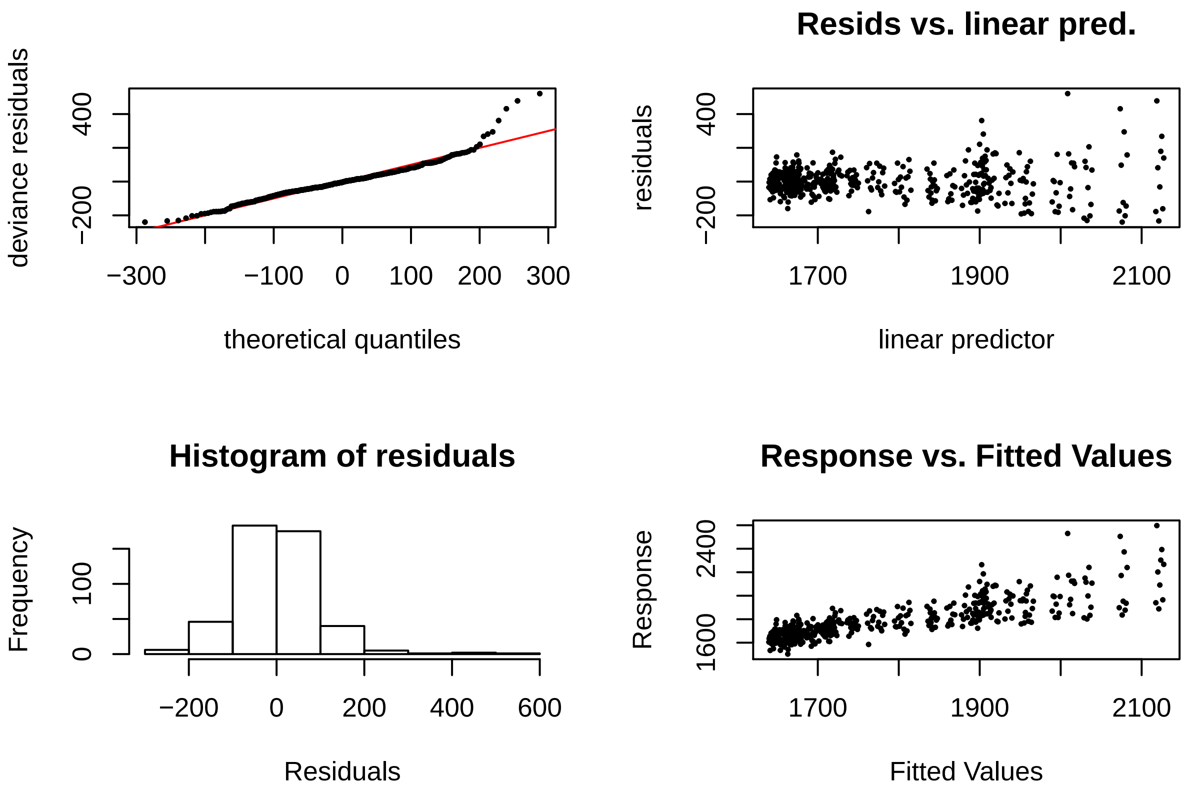


```
summary(gam_model)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Mortality ~ Year + s(Week)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -652.060   3448.379  -0.189     0.85
## Year           1.219      1.725   0.706     0.48
##
## Approximate significance of smooth terms:
##           edf Ref.df     F             p-value
## s(Week) 8.587  8.951 100.3 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## R-sq.(adj) =  0.661   Deviance explained = 66.8%
## GCV = 9014.6  Scale est. = 8806.7    n = 459
```

```
gam.check(gam_model,pch=19,cex=.3)
```



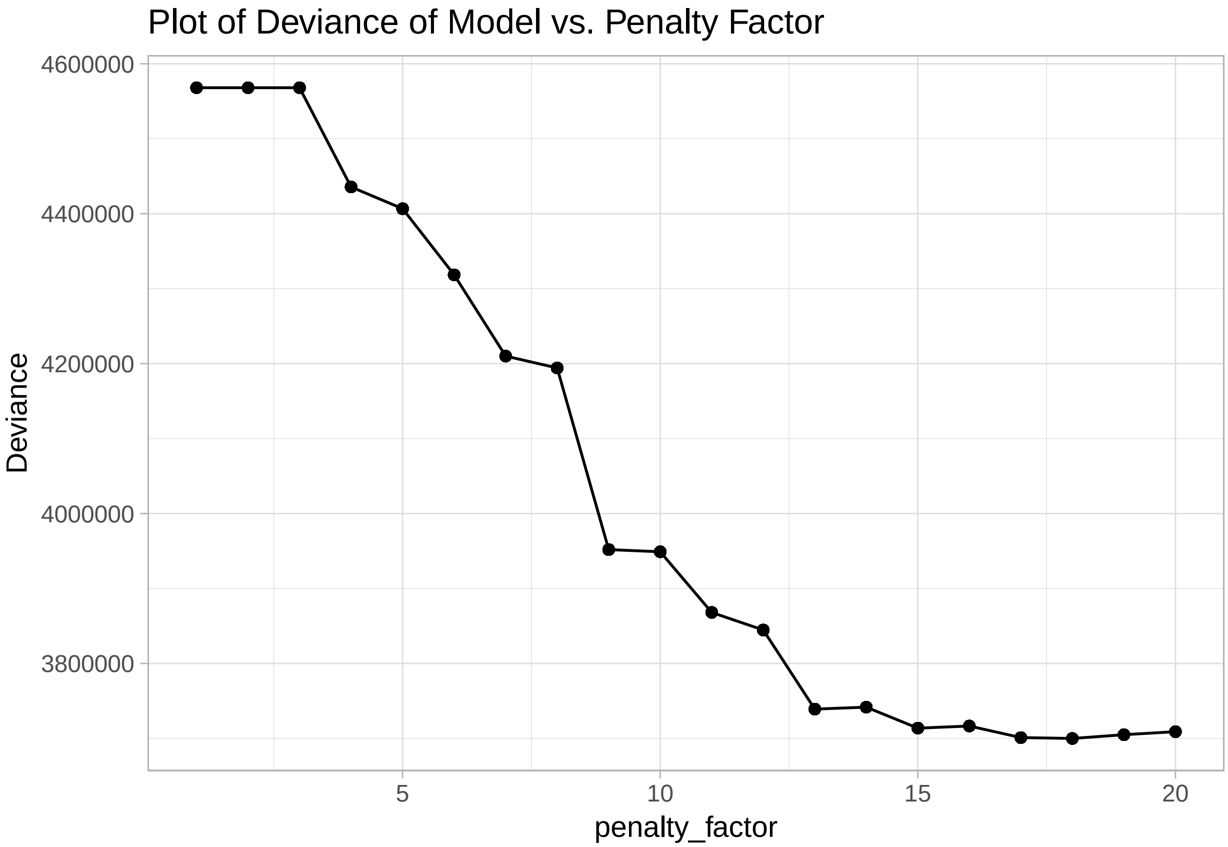
```
##
## Method: GCV   Optimizer: magic
## Smoothing parameter selection converged after 7 iterations.
## The RMS GCV score gradient at convergence was 0.114505 .
## The Hessian was positive definite.
## Model rank =  11 / 11
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##           k'  edf k-index p-value
## s(Week) 9.00 8.59    1.04    0.74
```

```
s=interp(temp$Year,temp$Week, fitted(gam_model))
persp3d(s$x, s$y, s$z, col="red")
```

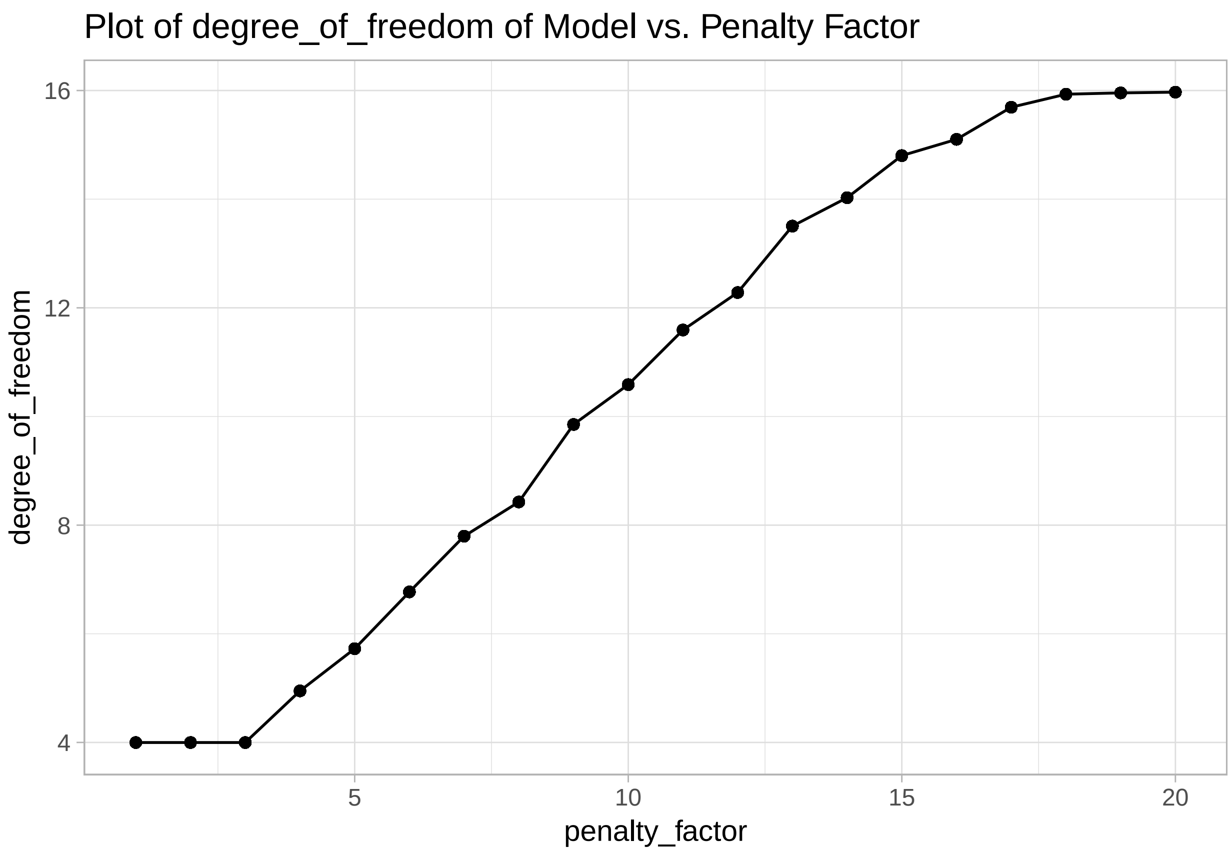
Analysis: From the plot of residuals we can see that the resisuals are normally distributed. Thus this is a good fit.

4. 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? Do your results confirm this relationship?

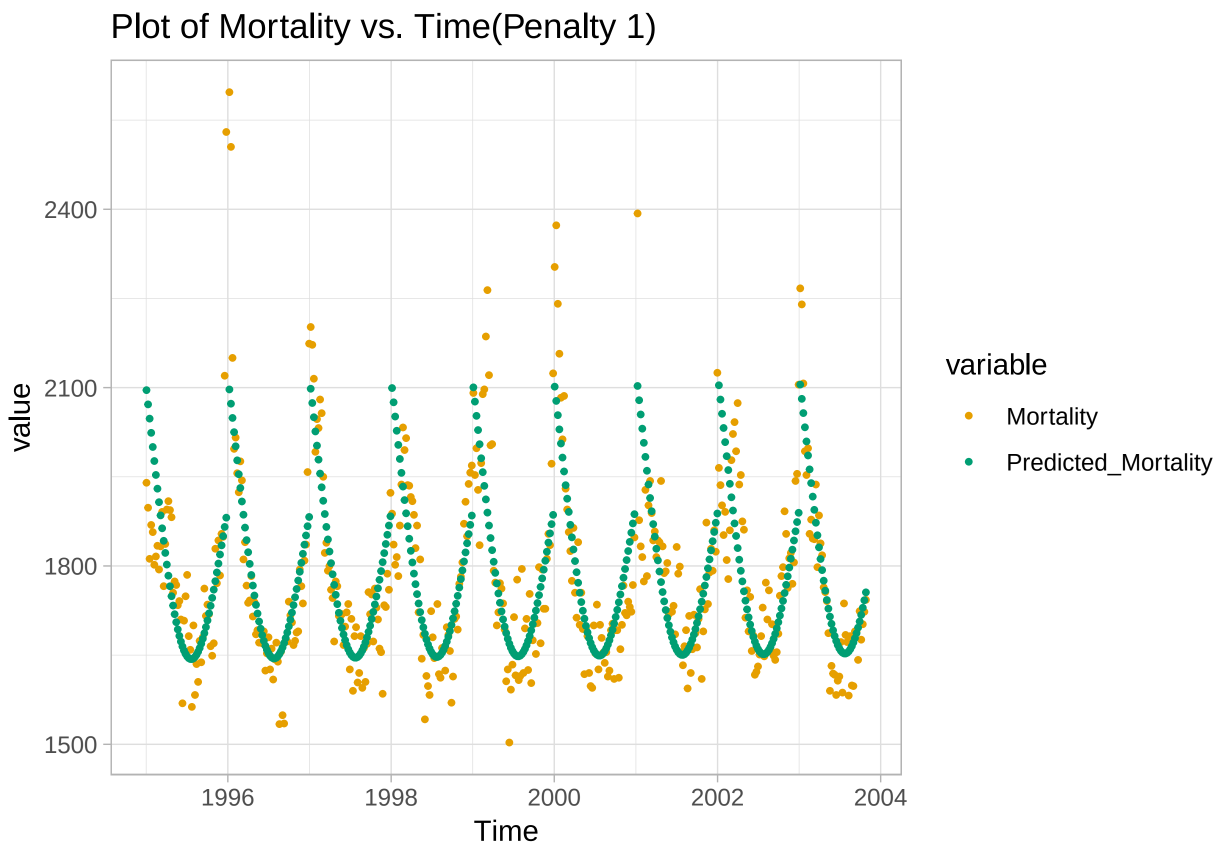
```r
model_deviance <- NULL
for(k in seq(from = 1, to = 20, by = 1))
{
gam_model <- mgcv::gam(data = flu_data, Mortality~Year+s(Week, k=k), method = "GCV.Cp")
temp <- cbind(gam_model$deviance, gam_model$fitted.values, gam_model$y, flu_data$Time_fixed,
              k, sum(influence(gam_model)))

model_deviance <- rbind(temp, model_deviance)
}
model_deviance <- as.data.frame(model_deviance)
colnames(model_deviance) <- c("Deviance", "Predicted_Mortality", "Mortality", "Time",
                              "penalty_factor", "degree_of_freedom")
model_deviance$Time <- as.Date(model_deviance$Time, origin = '1970-01-01')


# plot of deviance
p6 <- ggplot(data=model_deviance, aes(x = penalty_factor, y = Deviance)) +
geom_point() +
  geom_line() +
      theme_light() +
ggtitle("Plot of Deviance of Model vs. Penalty Factor")
p6
```



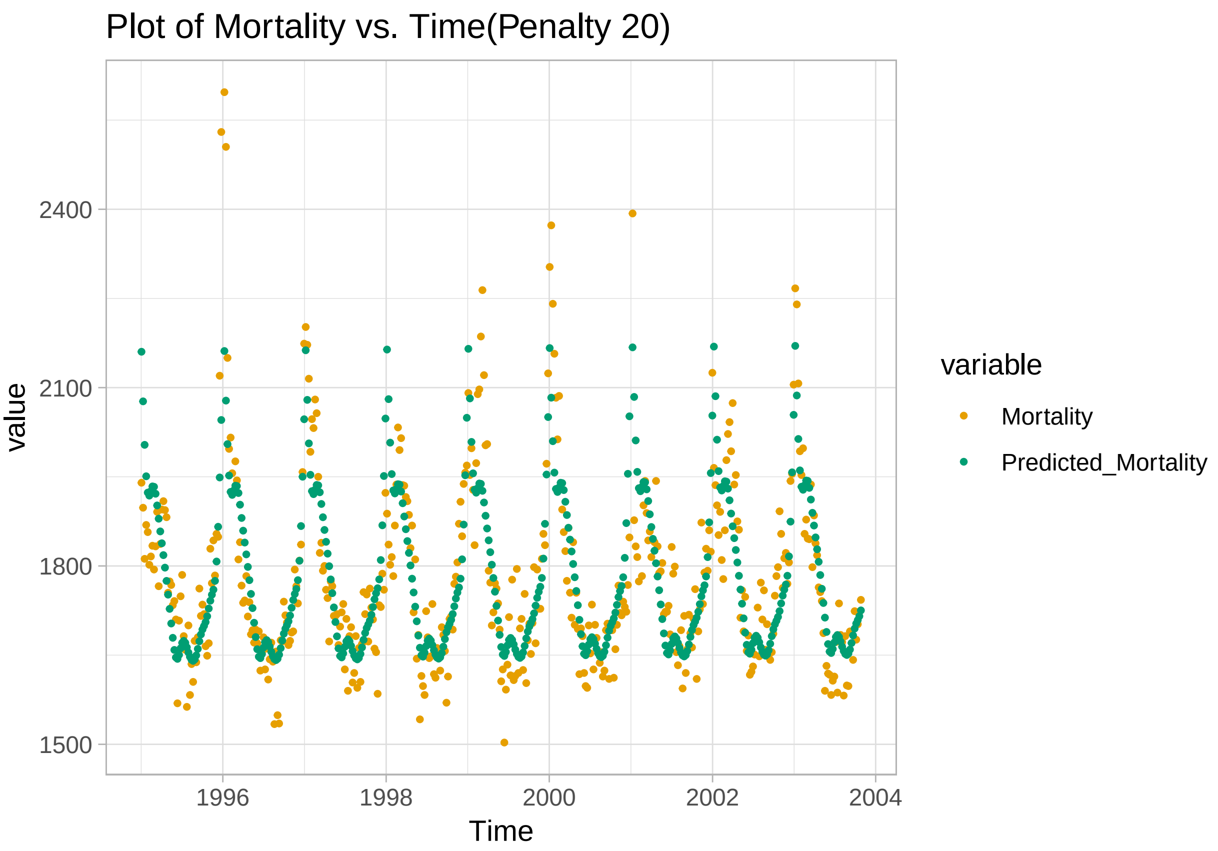
```
# plot of degree of freedom
p7 <- ggplot(data=model_deviance, aes(x = penalty_factor, y = degree_of_freedom)) +
geom_point() +
  geom_line() +
      theme_light() +
ggtitle("Plot of degree_of_freedom of Model vs. Penalty Factor")
p7
```



```
model_deviance_wide <- melt(model_deviance[,c("Time", "penalty_factor",
                                              "Mortality", "Predicted_Mortality")],
                            id.vars = c("Time", "penalty_factor")) %>%
  filter(penalty_factor %in% c("1","5","15", "20"))

# plot of predicted vs. observed mortality
p8 <- ggplot(data=model_deviance_wide[model_deviance_wide$penalty_factor == 1,],
             aes(x= Time, y = value)) +
  geom_point(aes(color = variable), size=0.7) +
  scale_color_manual(values=c("#E69F00", "#009E73")) +
  theme_light() +
  ggtitle("Plot of Mortality vs. Time(Penalty 1)")

p9 <- ggplot(data=model_deviance_wide[model_deviance_wide$penalty_factor == 20,],
             aes(x= Time, y = value)) +
  geom_point(aes(color = variable), size=0.7) +
  scale_color_manual(values=c("#E69F00", "#009E73")) +
    theme_light() +
  ggtitle("Plot of Mortality vs. Time(Penalty 20)")

p8
```


Plot of Mortality vs. Time(Penalty 1)
value
2400
2100
1800
1500
1996
1998
2000
2002
2004
Time
variable
● Mortality
● Predicted_Mortality

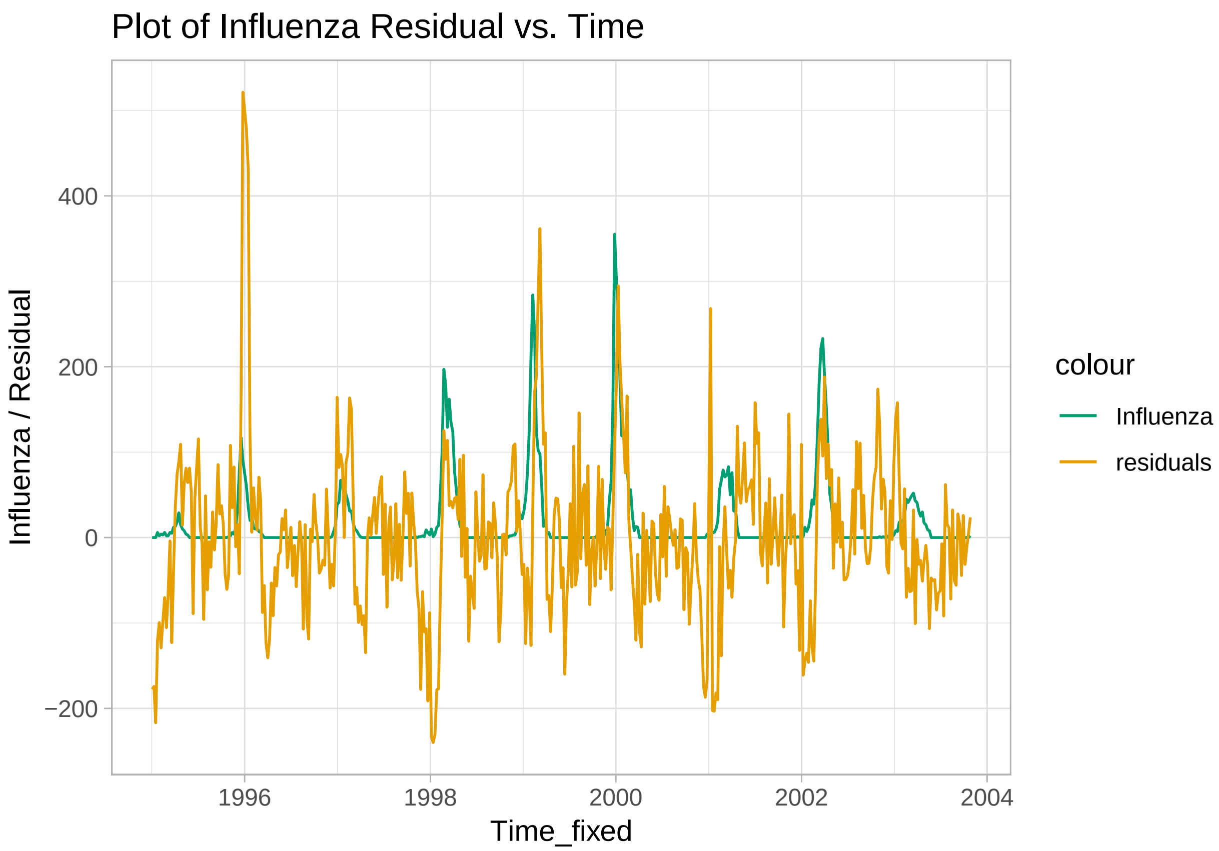
```
p9
```



Analysis: theortical maximum degree of freedom is k-1.

5. 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?

```r
gam_model <- mgcv::gam(data = flu_data, Mortality~Year+s(Week), method = "GCV.Cp")

temp <- flu_data
temp <- cbind(temp, residuals = gam_model$residuals)


p10 <- ggplot(data = temp, aes(x = Time_fixed)) +
  geom_line(aes( y = Influenza, color = "Influenza")) +
  geom_line(aes(y = residuals, color = "residuals")) +
      theme_light() +
  scale_color_manual(values=c(Influenza = "#009E73", residuals = "#E69F00")) +
  labs(y = "Influenza / Residual") +
  ggtitle("Plot of Influenza Residual vs. Time")

p10
```



6. 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.

```r
#gam_model_additive <- mgcv::gam(data = flu_data, Mortality~s(Year)+s(Week), method = "GCV.Cp")
```

Assignment 2

1. 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? Report the test error.

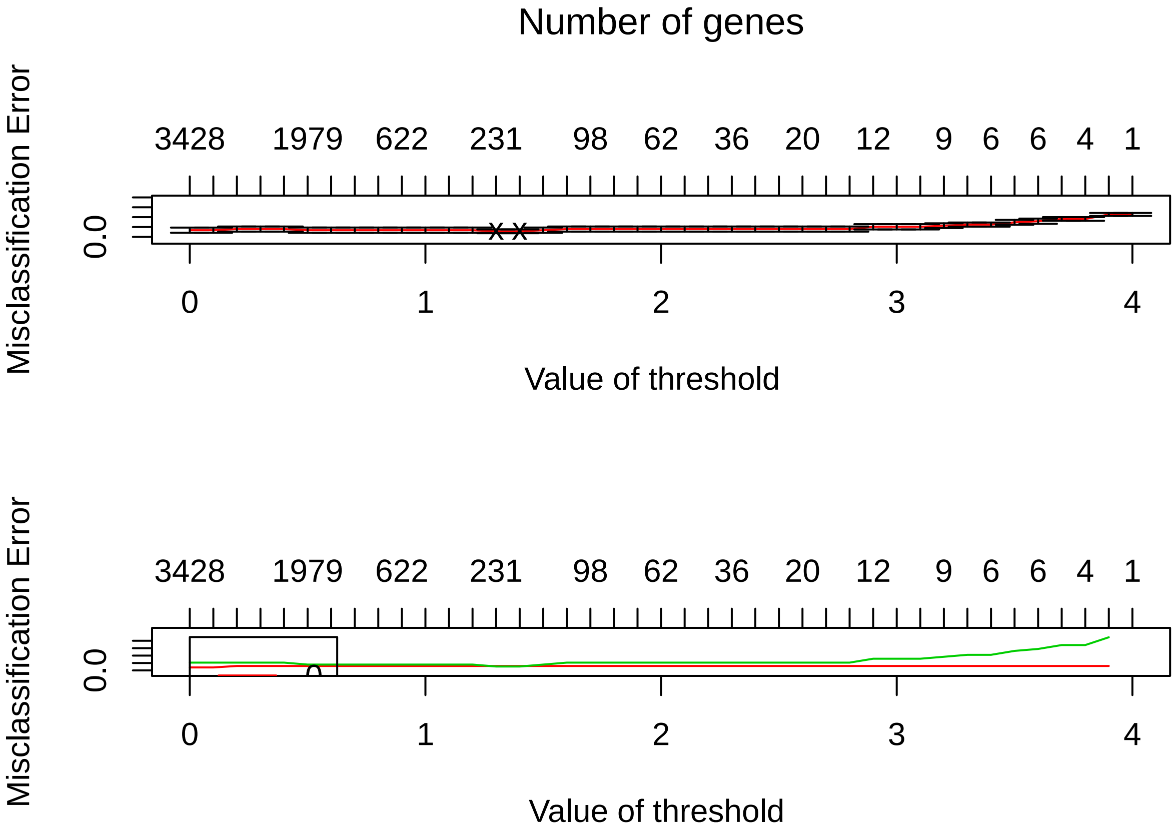
```
rm(list=ls())
gc()
```

```
##           used  (Mb) gc trigger  (Mb) max used  (Mb)
## Ncells 2909439 155.4    4746658 253.5  4746658 253.5
## Vcells 4774877  36.5   10146329  77.5  8388576  64.0
```

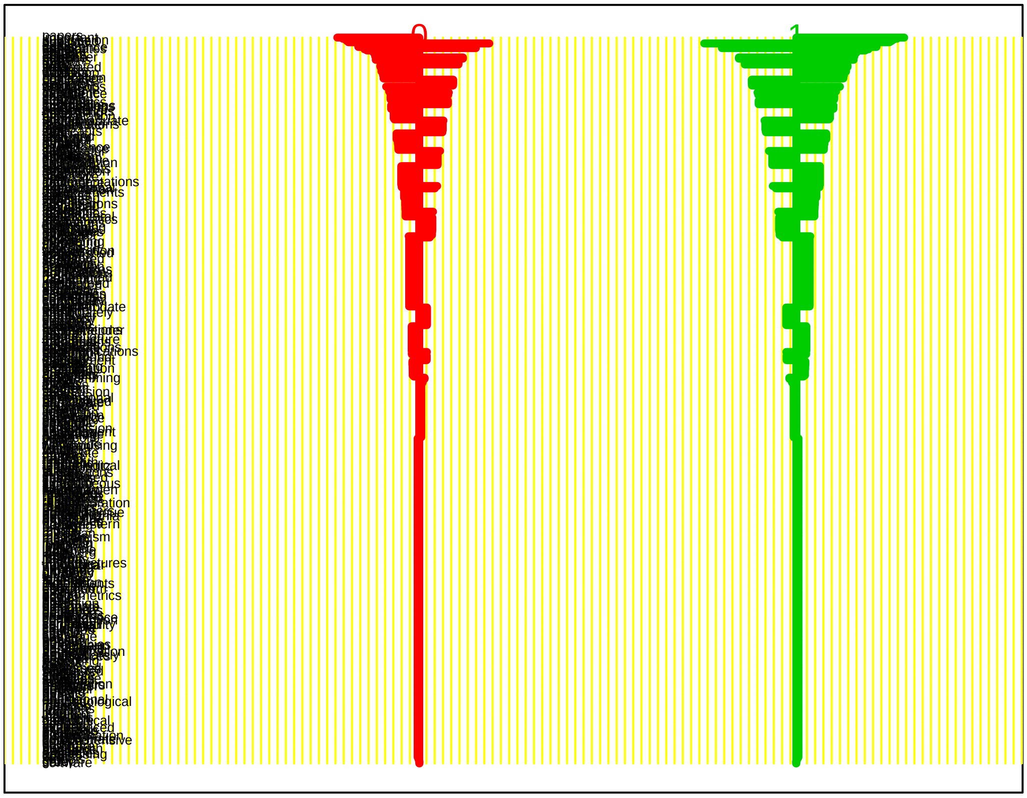
```
data <- read.csv(file = "data.csv", sep = ";", header = TRUE)
```

```
n=NROW(data)
data$Conference <- as.factor(data$Conference)
set.seed(12345)
id=sample(1:n, floor(n*0.7))
train=data[id,]
test = data[-id,]

rownames(train)=1:nrow(train)
x=t(train[,-4703])
y=train[[4703]]

rownames(test)=1:nrow(test)
x_test=t(test[,-4703])
y_test=test[[4703]]

mydata = list(x=x,y=as.factor(y),geneid=as.character(1:nrow(x)), genenames=rownames(x))
mydata_test = list(x=x_test,y=as.factor(y_test),geneid=as.character(1:nrow(x)), genenames=rownames(x))
model=pamr.train(mydata,threshold=seq(0,4, 0.1))

cvmodel=pamr.cv(model, mydata)
important_gen <- as.data.frame(pamr.listgenes(model, mydata, threshold = 0.9))
predicted_scc_test <- pamr.predict(model, newx = x_test, threshold = 0.9)
```

```
pamr.plotcv(cvmodel)
```



```
pamr.plotcen(model, mydata, threshold = 0.9)
```



```r
conf_scc <- table(y_test, predicted_scc_test)
names(dimnames(conf_scc)) <- c("Actual Test", "Predicted Srunken Centroid Test")
result_scc <- caret::confusionMatrix(conf_scc)
caret::confusionMatrix(conf_scc)
```

```
## Confusion Matrix and Statistics
##
##              Predicted Srunken Centroid Test
## Actual Test  0  1
##            0 10  0
##            1  2  8
##
##                Accuracy : 0.9
##                  95% CI : (0.683, 0.9877)
##     No Information Rate : 0.6
##     P-Value [Acc > NIR] : 0.003611
##
##                   Kappa : 0.8
##  Mcnemar's Test P-Value : 0.479500
##
##             Sensitivity : 0.8333
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.8000
##              Prevalence : 0.6000
##          Detection Rate : 0.5000
##    Detection Prevalence : 0.5000
##       Balanced Accuracy : 0.9167
##
##        'Positive' Class : 0
##
```

2. Compute the test error and the number of the contributing features for the following methods fitted to the training data: a. Elastic net with the binomial response and alpha = 0.5 in which penalty is selected by the cross-validation. b. Support vector machine with "vanilladot" kernel. 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?

```r
x = train[,-4703] %>% as.matrix()
y = train[,4703]

x_test = test[,-4703] %>% as.matrix()
y_test = test[,4703]

cvfit = cv.glmnet(x=x, y=y, alpha = 0.5, family =   "binomial")
predicted_elastic_test <- predict.cv.glmnet(cvfit, newx = x_test, s = "lambda.min", type = "class")

conf_elastic_net <- table(y_test, predicted_elastic_test)
names(dimnames(conf_elastic_net)) <- c("Actual Test", "Predicted ElasticNet Test")
result_elastic_net <- caret::confusionMatrix(conf_elastic_net)
caret::confusionMatrix(conf_elastic_net)
```

```
## Confusion Matrix and Statistics
##
##            Predicted ElasticNet Test
## Actual Test  0  1
##           0 10  0
##           1  2  8
##
##                Accuracy : 0.9
##                  95% CI : (0.683, 0.9877)
##     No Information Rate : 0.6
##     P-Value [Acc > NIR] : 0.003611
##
##                   Kappa : 0.8
##  Mcnemar's Test P-Value : 0.479500
##
##             Sensitivity : 0.8333
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.8000
##              Prevalence : 0.6000
##          Detection Rate : 0.5000
##    Detection Prevalence : 0.5000
##       Balanced Accuracy : 0.9167
##
##        'Positive' Class : 0
```

```
##
```

```r
# svm
svm_fit <- kernlab::ksvm(x, y, kernel="vanilladot", scale = FALSE, type = "C-svc")
```

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

```r
predicted_svm_test <- predict(svm_fit, x_test, type="response")

conf_svm_tree <- table(y_test, predicted_svm_test)
names(dimnames(conf_svm_tree)) <- c("Actual Test", "Predicted SVM Test")
result_svm <- caret::confusionMatrix(conf_svm_tree)
caret::confusionMatrix(conf_svm_tree)
```

```
## Confusion Matrix and Statistics
##
##             Predicted SVM Test
## Actual Test  0  1
##           0 10  0
##           1  1  9
##
##                Accuracy : 0.95
##                  95% CI : (0.7513, 0.9987)
##     No Information Rate : 0.55
##     P-Value [Acc > NIR] : 0.0001114
##
##                   Kappa : 0.9
##  Mcnemar's Test P-Value : 1.0000000
##
##             Sensitivity : 0.9091
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.9000
##              Prevalence : 0.5500
##          Detection Rate : 0.5000
##    Detection Prevalence : 0.5000
##       Balanced Accuracy : 0.9545
##
##        'Positive' Class : 0
##
```

```r
# creating table
final_result <- cbind(result_scc$overall[[1]]*100,
                      result_elastic_net$overall[[1]]*100,
                      result_svm$overall[[1]] *100) %>% as.data.frame()

colnames(final_result) <- c("Accuracy of Nearest Shrunken Centroid Model",
                                   "Accuracy of ElasticNet",
                                   "Accuracy SVM Model")

knitr::kable(final_result, caption = "Accuracy of Model on Test dataset")
```

Table 1: Accuracy of Model on Test dataset

Accuracy of Nearest Shrunken Centroid Model	Accuracy of ElasticNet	Accuracy SVM Model
90	90	95

Accuracy of Nearest Shrunken Centroid Model	Accuracy of ElasticNet	Accuracy SVM Model

3. Implement Benjamini-Hochberg method for the original data, and use t.test() for computing p-values. Which features correspond to the rejected hypotheses? Interpret the result.

Appendix

```
knitr::opts_chunk$set(echo = TRUE)
if (!require("pacman")) install.packages("pacman")
pacman::p_load(xlsx, ggplot2, tidyr, dplyr, reshape2, gridExtra,
               mgcv, rgl, akima, pamr, caret, glmnet, kernlab)

set.seed(12345)
options("jtools-digits" = 2, scipen = 999)

# colours (colour blind friendly)
cbPalette <- c("#999999", "#E69F00", "#56B4E9", "#009E73", "#F0E442", "#0072B2",
               "#D55E00", "#CC79A7")

## Making title in the center
theme_update(plot.title = element_text(hjust = 0.5))
set.seed(12345)

# Importing data
flu_data = read.xlsx("influenza.xlsx", sheetName = "Raw data")
flu_data$Time_fixed <- as.Date(paste(flu_data$Year, flu_data$Week, 1, sep="-"), "%Y-%U-%u")
flu_data$influ_perc <- (flu_data$Influenza/flu_data$Mortality) * 100

# Plot

p1 <- ggplot(flu_data, aes(x=Time_fixed, y = Mortality)) +
  geom_line(color = "#999999", size = 1) +
    scale_fill_brewer() +
      theme_light() +
  ggtitle("Time series of Mortality")

p2 <- ggplot(flu_data, aes(x=Time_fixed, y = Influenza)) +
  geom_line(color = "#E69F00", size = 1) +
      scale_fill_brewer() +
      theme_light() +
  ggtitle("Time series of Influenza")

p3 <- ggplot(flu_data, aes(x=Time_fixed, y = influ_perc)) +
  geom_line(color = "#56B4E9", size = 1) +
      scale_fill_brewer() +
      theme_light() +
  ggtitle("Time series of % Mortalitiy due to Influenza")

gridExtra::grid.arrange(p1, p2, ncol=1)
```

```
p3

gam_model <- mgcv::gam(data = flu_data, Mortality~Year+s(Week), method = "GCV.Cp")
summary(gam_model)
#plot the fit

p4 <- plot(gam_model, main= "Plot of GAM fit on Flu Data")
temp <- flu_data
temp$Fitted_Mortality <- gam_model$fitted.values

p5 <- ggplot(data=temp, aes(x = Time_fixed, y = Fitted_Mortality)) +
   geom_line(color = "#009E73", size = 1) +
    scale_fill_brewer() +
      theme_light() +
  ggtitle("Time series of Fitted Mortality")

grid.arrange(p1, p5, nrow = 2)

summary(gam_model)
gam.check(gam_model,pch=19,cex=.3)

s=interp(temp$Year,temp$Week, fitted(gam_model))
persp3d(s$x, s$y, s$z, col="red")
model_deviance <- NULL
for(k in seq(from = 1, to = 20, by = 1))
{
gam_model <- mgcv::gam(data = flu_data, Mortality~Year+s(Week, k=k), method = "GCV.Cp")
temp <- cbind(gam_model$deviance, gam_model$fitted.values, gam_model$y, flu_data$Time_fixed,
              k, sum(influence(gam_model)))

model_deviance <- rbind(temp, model_deviance)
}
model_deviance <- as.data.frame(model_deviance)
colnames(model_deviance) <- c("Deviance", "Predicted_Mortality", "Mortality", "Time",
                              "penalty_factor", "degree_of_freedom")
model_deviance$Time <- as.Date(model_deviance$Time, origin = '1970-01-01')


# plot of deviance
p6 <- ggplot(data=model_deviance, aes(x = penalty_factor, y = Deviance)) +
geom_point() +
  geom_line() +
      theme_light() +
ggtitle("Plot of Deviance of Model vs. Penalty Factor")
p6

# plot of degree of freedom
p7 <- ggplot(data=model_deviance, aes(x = penalty_factor, y = degree_of_freedom)) +
geom_point() +
  geom_line() +
      theme_light() +
ggtitle("Plot of degree_of_freedom of Model vs. Penalty Factor")
p7
```

```
model_deviance_wide <- melt(model_deviance[,c("Time", "penalty_factor",
                                              "Mortality", "Predicted_Mortality")],
                            id.vars = c("Time", "penalty_factor")) %>%
  filter(penalty_factor %in% c("1","5","15", "20"))

# plot of predicted vs. observed mortality
p8 <- ggplot(data=model_deviance_wide[model_deviance_wide$penalty_factor == 1,],
             aes(x= Time, y = value)) +
  geom_point(aes(color = variable), size=0.7) +
  scale_color_manual(values=c("#E69F00", "#009E73")) +
  theme_light() +
  ggtitle("Plot of Mortality vs. Time(Penalty 1)")

p9 <- ggplot(data=model_deviance_wide[model_deviance_wide$penalty_factor == 20,],
             aes(x= Time, y = value)) +
  geom_point(aes(color = variable), size=0.7) +
  scale_color_manual(values=c("#E69F00", "#009E73")) +
    theme_light() +
  ggtitle("Plot of Mortality vs. Time(Penalty 20)")

p8
p9

gam_model <- mgcv::gam(data = flu_data, Mortality~Year+s(Week), method = "GCV.Cp")

temp <- flu_data
temp <- cbind(temp, residuals = gam_model$residuals)


p10 <- ggplot(data = temp, aes(x = Time_fixed)) +
  geom_line(aes( y = Influenza, color = "Influenza")) +
  geom_line(aes(y = residuals, color = "residuals")) +
      theme_light() +
  scale_color_manual(values=c(Influenza = "#009E73", residuals = "#E69F00")) +
  labs(y = "Influenza / Residual") +
  ggtitle("Plot of Influenza Residual vs. Time")

p10

#gam_model_additive <- mgcv::gam(data = flu_data, Mortality~s(Year)+s(Week), method = "GCV.Cp")
rm(list=ls())
gc()
data <- read.csv(file = "data.csv", sep = ";", header = TRUE)
n=NROW(data)
data$Conference <- as.factor(data$Conference)
set.seed(12345)
id=sample(1:n, floor(n*0.7))
train=data[id,]
test = data[-id,]

rownames(train)=1:nrow(train)
x=t(train[,-4703])
y=train[[4703]]
```

```r
rownames(test)=1:nrow(test)
x_test=t(test[,-4703])
y_test=test[[4703]]

mydata = list(x=x,y=as.factor(y),geneid=as.character(1:nrow(x)), genenames=rownames(x))
mydata_test = list(x=x_test,y=as.factor(y_test),geneid=as.character(1:nrow(x)), genenames=rownames(x))
model=pamr.train(mydata,threshold=seq(0,4, 0.1))

cvmodel=pamr.cv(model, mydata)
important_gen <- as.data.frame(pamr.listgenes(model, mydata, threshold = 0.9))
predicted_scc_test <- pamr.predict(model, newx = x_test, threshold = 0.9)
pamr.plotcv(cvmodel)
pamr.plotcen(model, mydata, threshold = 0.9)
conf_scc <- table(y_test, predicted_scc_test)
names(dimnames(conf_scc)) <- c("Actual Test", "Predicted Srunken Centroid Test")
result_scc <- caret::confusionMatrix(conf_scc)
caret::confusionMatrix(conf_scc)

x = train[,-4703] %>% as.matrix()
y = train[,4703]

x_test = test[,-4703] %>% as.matrix()
y_test = test[,4703]

cvfit = cv.glmnet(x=x, y=y, alpha = 0.5, family =   "binomial")
predicted_elastic_test <- predict.cv.glmnet(cvfit, newx = x_test, s = "lambda.min", type = "class")

conf_elastic_net <- table(y_test, predicted_elastic_test)
names(dimnames(conf_elastic_net)) <- c("Actual Test", "Predicted ElasticNet Test")
result_elastic_net <- caret::confusionMatrix(conf_elastic_net)
caret::confusionMatrix(conf_elastic_net)


# svm
svm_fit <- kernlab::ksvm(x, y, kernel="vanilladot", scale = FALSE, type = "C-svc")
predicted_svm_test <- predict(svm_fit, x_test, type="response")

conf_svm_tree <- table(y_test, predicted_svm_test)
names(dimnames(conf_svm_tree)) <- c("Actual Test", "Predicted SVM Test")
result_svm <- caret::confusionMatrix(conf_svm_tree)
caret::confusionMatrix(conf_svm_tree)

# creating table
final_result <- cbind(result_scc$overall[[1]]*100,
                      result_elastic_net$overall[[1]]*100,
                      result_svm$overall[[1]] *100) %>% as.data.frame()

colnames(final_result) <- c("Accuracy of Nearest Shrunken Centroid Model",
                                   "Accuracy of ElasticNet",
                                   "Accuracy SVM Model")

knitr::kable(final_result, caption = "Accuracy of Model on Test dataset")
```