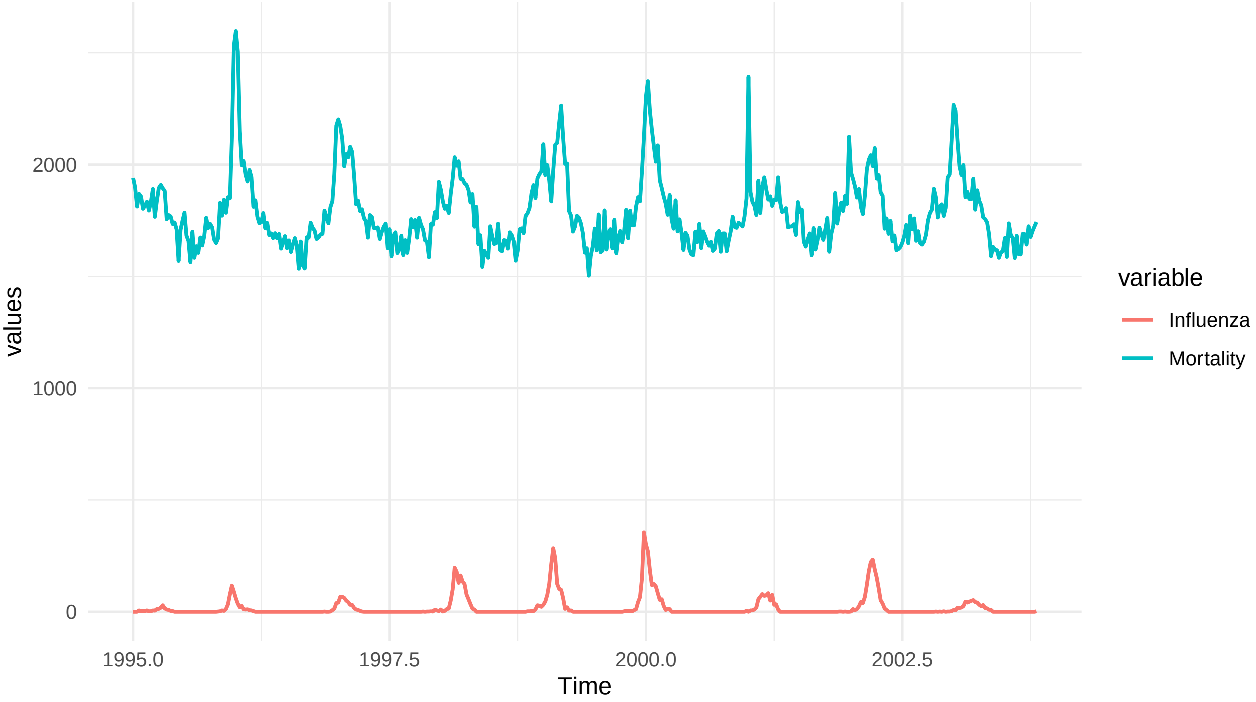
Computer lab 2 block 2

Vasileia Kampouraki & Laura Julià Melis & Mengxin Liu

12/16/2019

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

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.



Looking at the time series of Mortality against time we can observe peaks at the first weeks of every year, which means that the highest number of mortalities are during the first weeks of each year. This is an indication that there is anual seasonality in our data.

Regarding the Influenza time series we see that there is an increase in influenza cases from 1995 to 2000 with the peak being at 2000 and then there is a relative decrease again. The peaks here also appear during the first weeks of every year.

It's obvious that the mortality rate is highly connected to influenza cases, as we can observe that during the periods of each year that the influenza cases are on their peak, the highest number of death cases are recorded at that exact time too.

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.

In order to fit a GAM model we have used the `gam()` function indicating that the dependent variable (Mortality) is normally distributed by using the argument *family = gaussian()*. Also, we have set that the smoothing parameter estimation method should be generalized cross-validation with the argument *method = "GCV.Cp"*. Finally, in the formula argument, we have written the Week variable as *s(data$Week, k)*, where k is the amount of unique values of the Week variable, because the `s()` function helps to set up a model using spline-based smooths.

Given that a generalized additive model is

$$EY = \alpha + s_1(X_1) + \cdots + s_p(X_p) + \sum_{j=1}^{q} \beta_j X_{p+j}$$

The probabilistic model in our case can be written as:

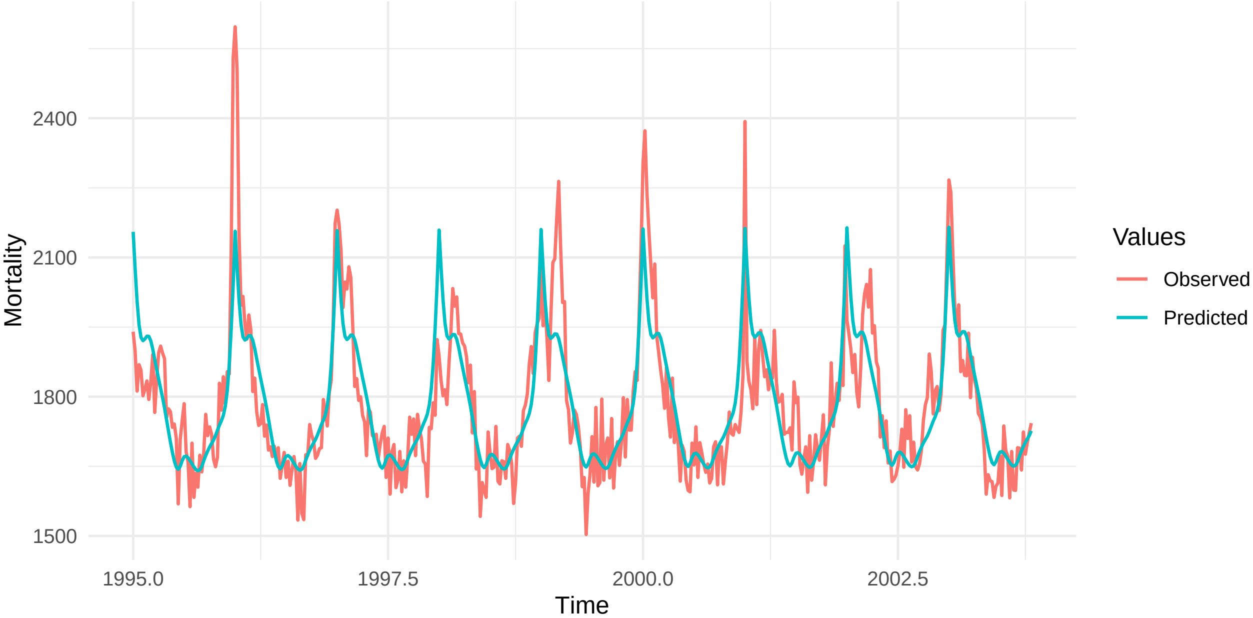
$$\text{Mortality} = \alpha + s(\text{Week}) + \beta_1 \cdot \text{Year} + \epsilon$$

where Mortality $\sim$ Normal(μ, σ^2) and $\epsilon \sim$ Normal$(0, \sigma^2)$.

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.

We already have the fitted model predictions of expected value for each observation in our gam object, in the model fitted in step (2): `gam_fit$fitted.values`.

We can create a series plot of the predicted and the observed values of mortality:



The quality of the fit seems to be good because the GAM model is able to predict all the peaks at the right time although the length of the peaks is always the same so that the highest values of the peaks are not always predicted correctly.

In addition, we observe a trend in mortality: mortality values appear to be cyclic. Around each beginning or end of the year (during winter) the mortality values increase significantly compared to those of the other periods of the year.

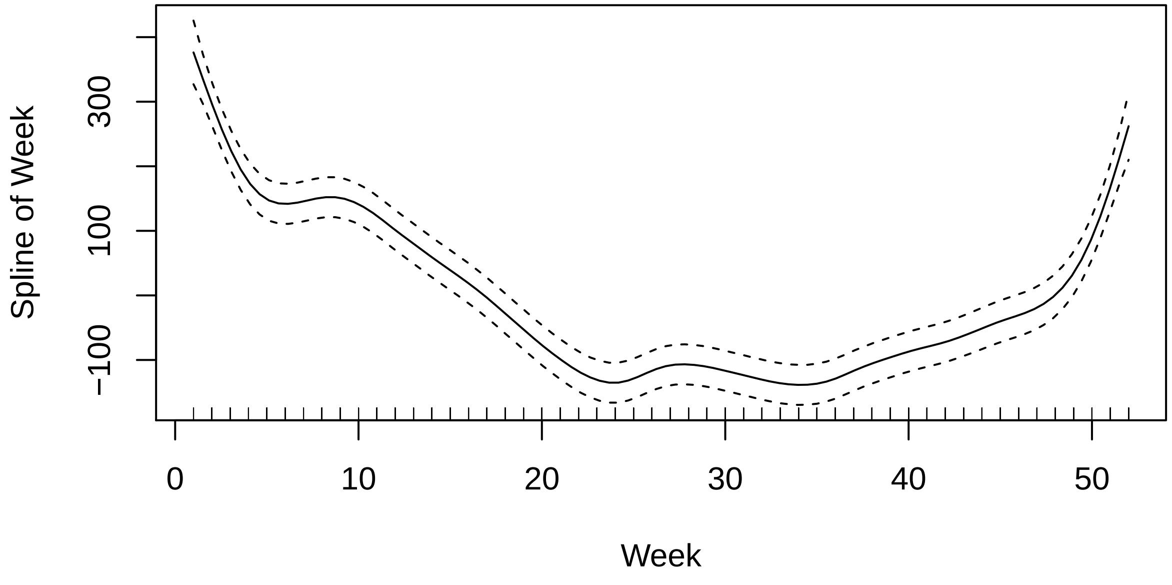
The output of the fitted GAM model is as follows:

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## data$Mortality ~ data$Year + s(data$Week, k = length(unique(data$Week)))
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -680.598   3367.760  -0.202    0.840
## data$Year      1.233      1.685   0.732    0.465
##
## Approximate significance of smooth terms:
##                edf Ref.df     F p-value
## s(data$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
```

From the output values, we can see that neither the intercept (α) nor the coefficient associated with the Year are significant in the model because their p-values are much greater than 0,05 (at 95% confidence level)[1]. This means that the variable Year is not relevant to explain the Mortality values. On the other hand, the spline of the Week is smaller than 0.05 meaning that is significant.

Also, we observe that the adjusted R^2 is 0.677 which indicates that 67.7% of all the variability of the response data is explained by the model.

Finally, we will plot the spline component:



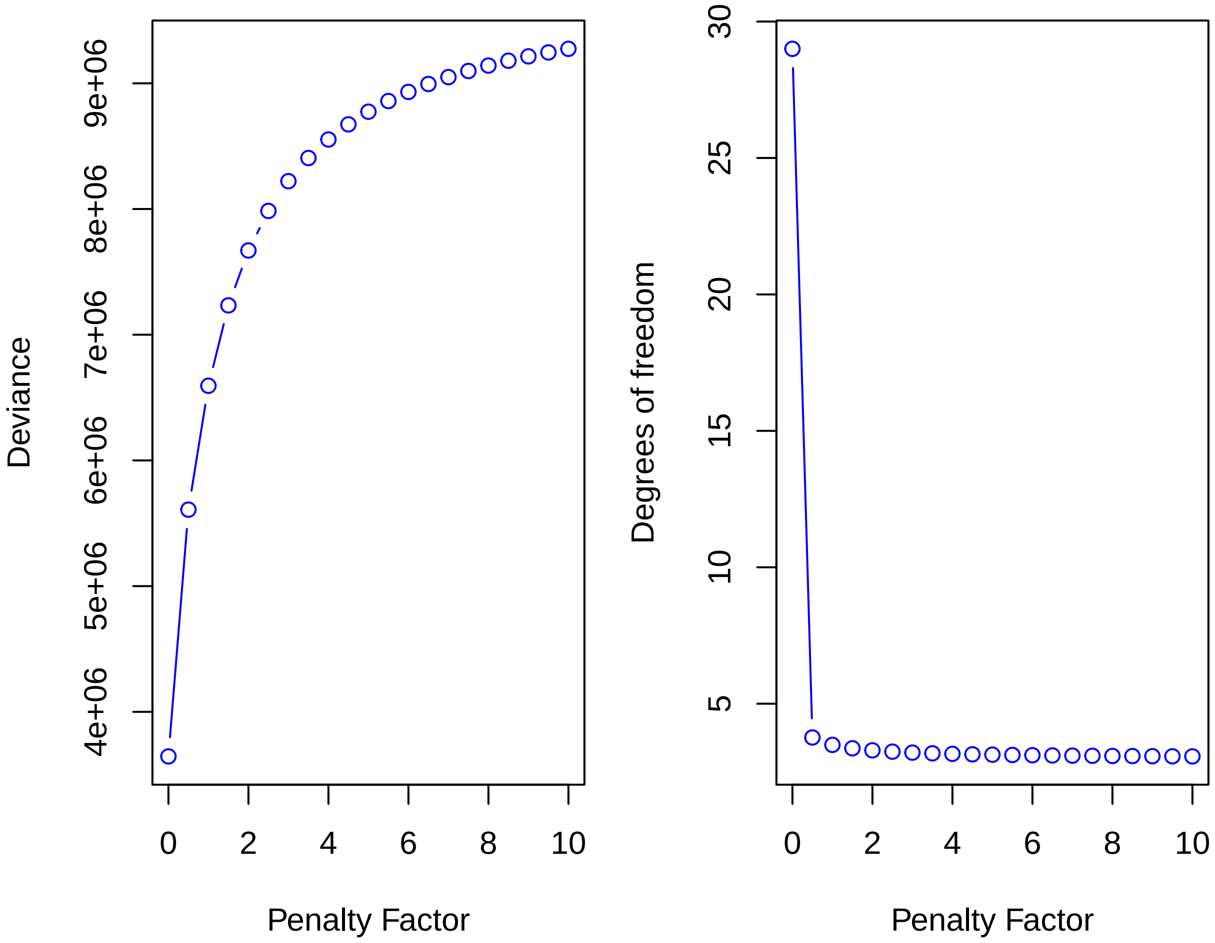
The dashed curves show pointwise 2σ limits for the fitted curve. Looking at the spline plot we can detect the seasonal pattern, where the values are very high during the first weeks, then they drop quickly and in the very ending of the year they rise again.

[1] When p-value < 0.05 we reject $H_0 : \beta_i = 0$ which expresses that the coefficient i is not significant (equals 0).

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?

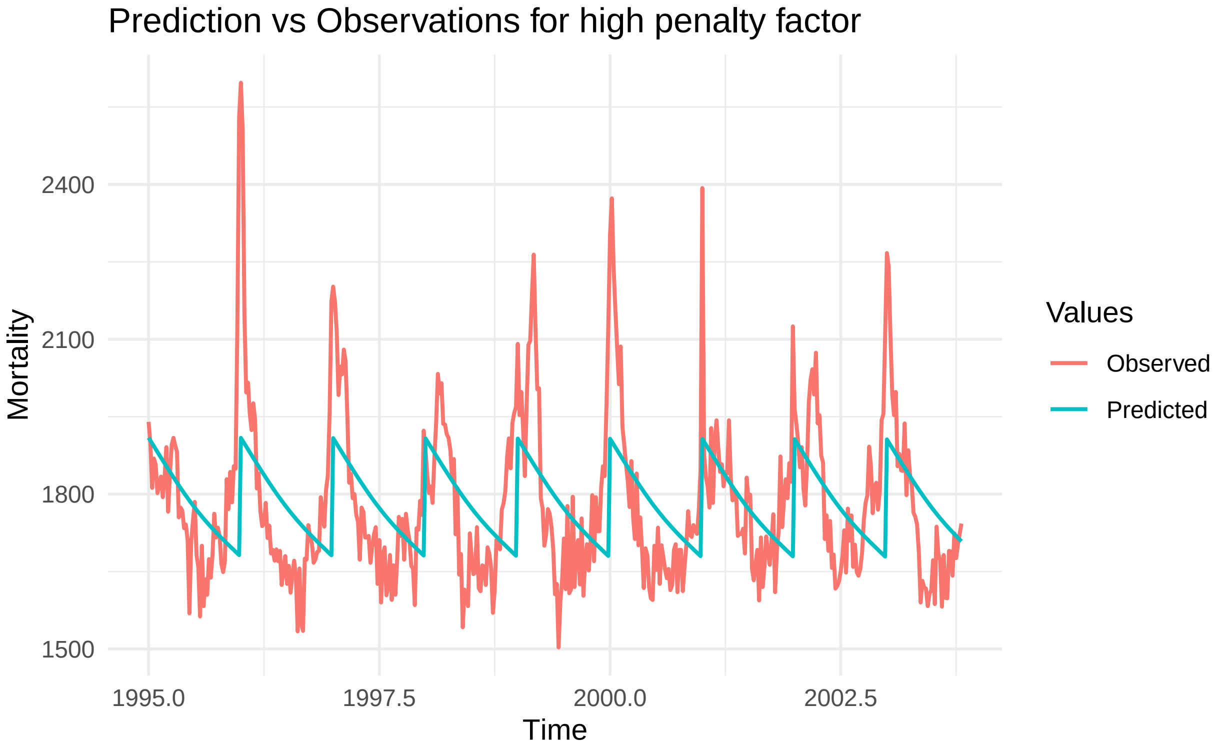
We can set different values for the smoothing penalty factor with the *sp* argument of the function `s()`. So, we will fit a GAM model for penalty factor values $\text{sp} = 0, 0.1, 0.2..., 4.9, 5$ and we will see how the deviance (`gam_fit$deviance`) and the degrees of freedom (difference between the number of observations included in the model and the model residual degrees of freedom) change in each case.

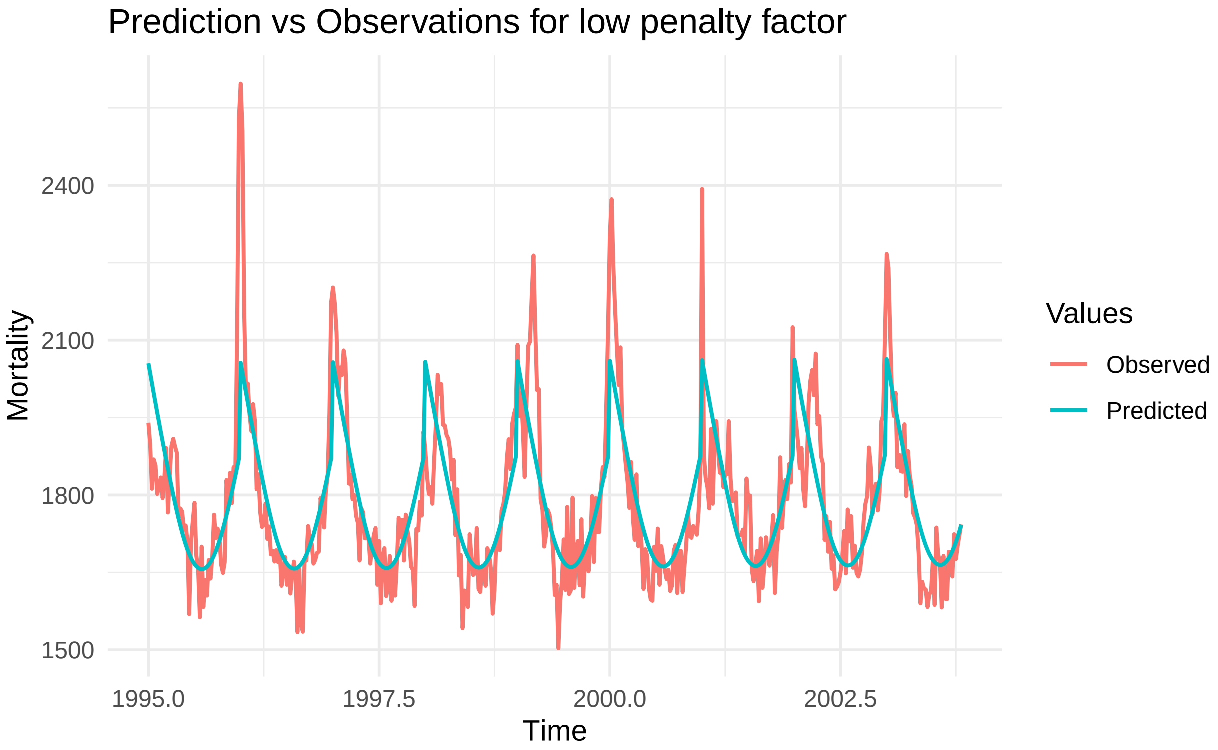
The plots of the different smoothing penalty factors against the deviance and the degrees of freedom of the model are:



The penalty factor penalizes the coefficients in order to control the degree of smoothness. So, when a high penalty factor is used to fit the GMA model, the smoothness performed is higher and as a result, we obtain a simpler model. From the plots, we observe this because as the penalty factor increases, the degrees of freedom decrease and the deviance increase.

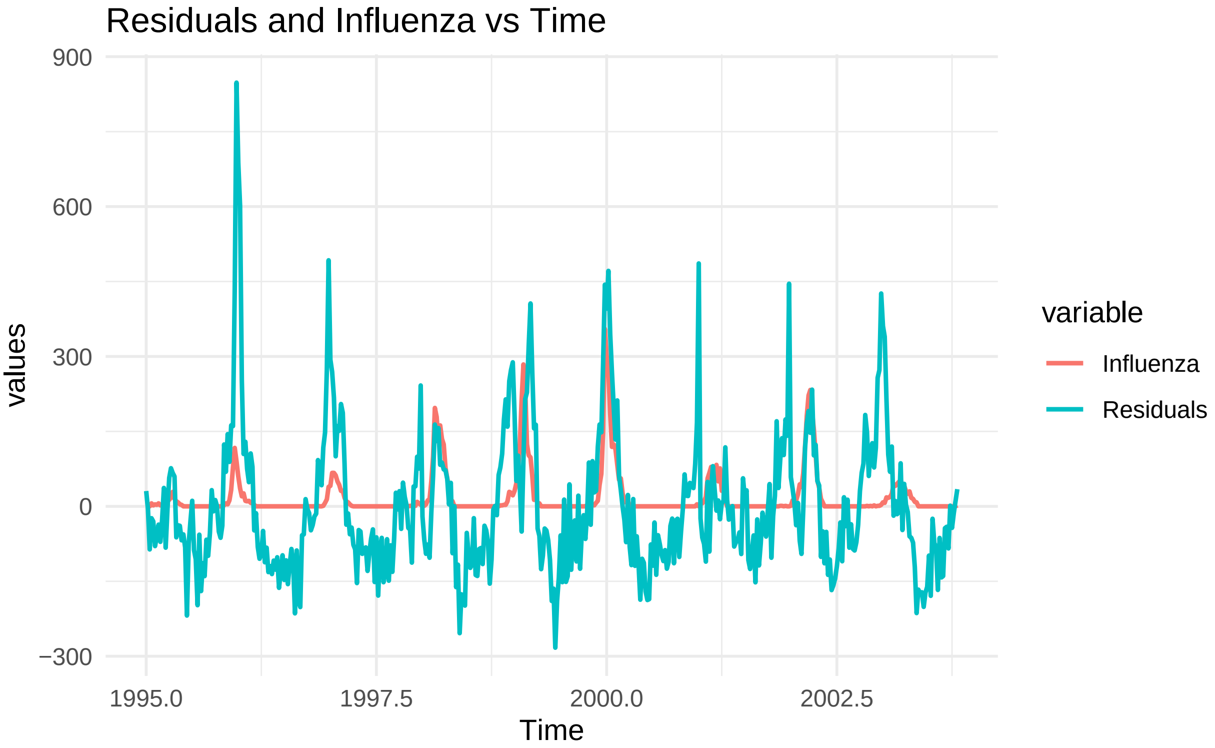
Now, we will fit two different models: one with a high penalty factor (sp=10) and one with a low penalty factor (sp=0.1). Then, if we plot the the predicted and observed mortality values against time we obtain the following plots:





These plots confirm what we said at the beginning of this question: the higher the penalty factor, the simpler the model. So, the model with a low penalty factor predicts better than the model with a high penalty factor.

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?



Looking at the plot of influenza and residuals against time we can see that there is a correlation between them. More precisely, we observe that the peaks of the residuals follow in some way the peaks of influenza.

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.

In this case, we want to fit the following probability model:

$$\text{Mortality} = \alpha + s(\text{Year}) + s(\text{Week}) + s(\text{Influenza}) + \epsilon$$

where Mortality $\sim$ Normal(μ, σ^2), $\epsilon \sim$ Normal$(0, \sigma^2)$ and the dimension of the basis used to represent the smooth term (argument k) is the number of unique values of each variable in each case.

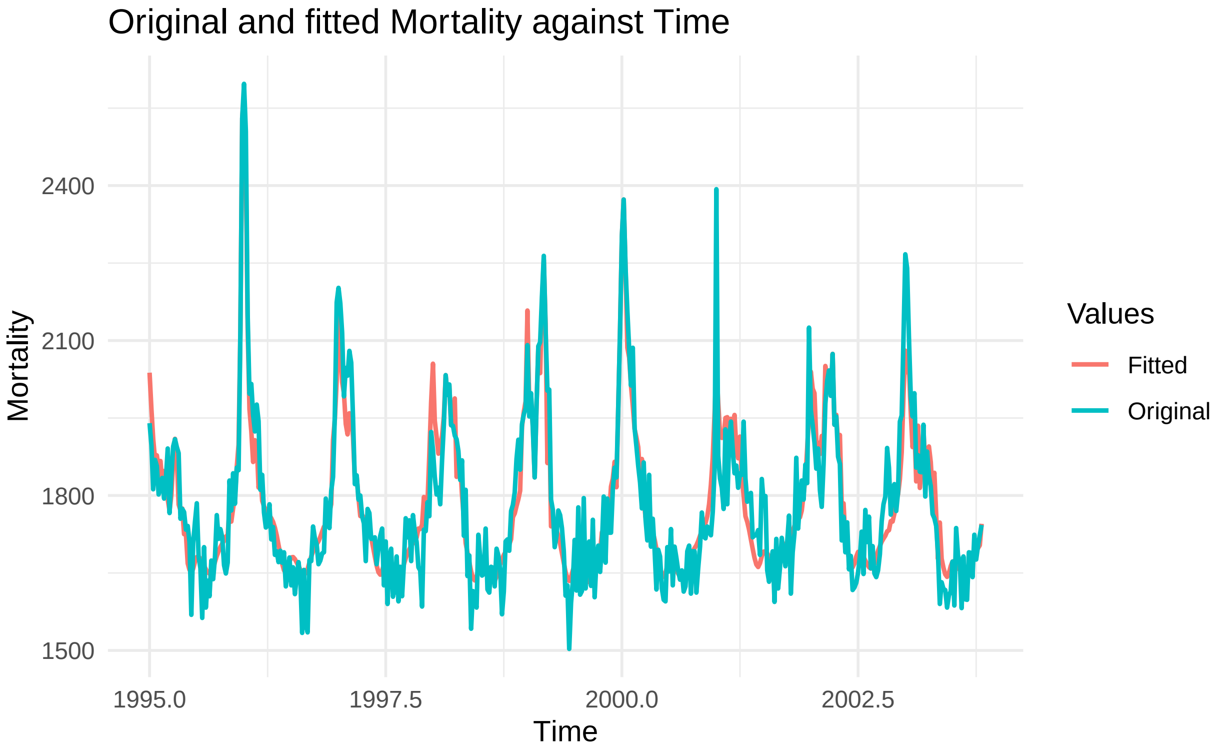
The output of the fitted GAM model is:

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## data$Mortality ~ s(data$Year, k = length(unique(data$Year))) +
##     s(data$Week, k = length(unique(data$Week))) + s(data$Influenza,
##     k = length(unique(data$Influenza)))
##
## Parametric coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1783.765      3.198   557.8   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                        edf Ref.df      F p-value
## s(data$Year)         4.587  5.592  1.500   0.178
## s(data$Week)        14.431 17.990 18.763  <2e-16 ***
## s(data$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
```

We can observe that in this new model, the only coefficient that seems to be not significant is the spline of the Year. Then, as the spline of Influenza is smaller than 0.05, we conclude that the variable is significant to explain the behaviour of the Mortality and consequently, the mortality is somehow influenced by the outbreaks of influenza.

Also, we can see that this new model manages to explain nearly 82% of all the variability of Mortality. Since the model fitted in question 2 explains 67.7%, we can conclude that this model is better.

Lastly, we will plot the original and fitted Mortality against Time:



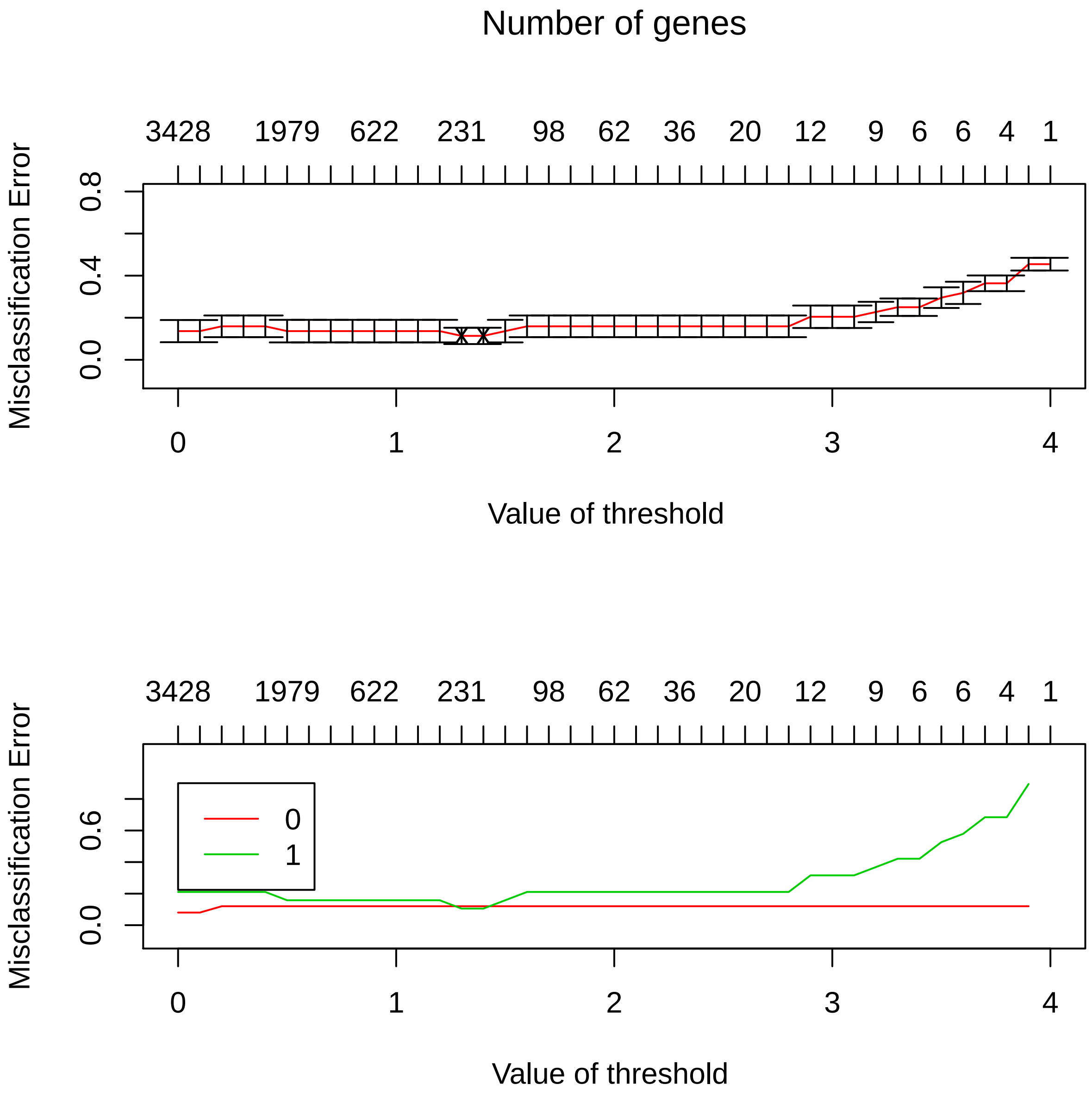
Plotting the original and fitted data of Mortality we can see that the predictions are very good and catch very well the seasonal changes of the data. Comparing this plot with the previous one shown in step (2) we observe that this one has a better predictive performance.

Assignment 2. High-dimensional methods.

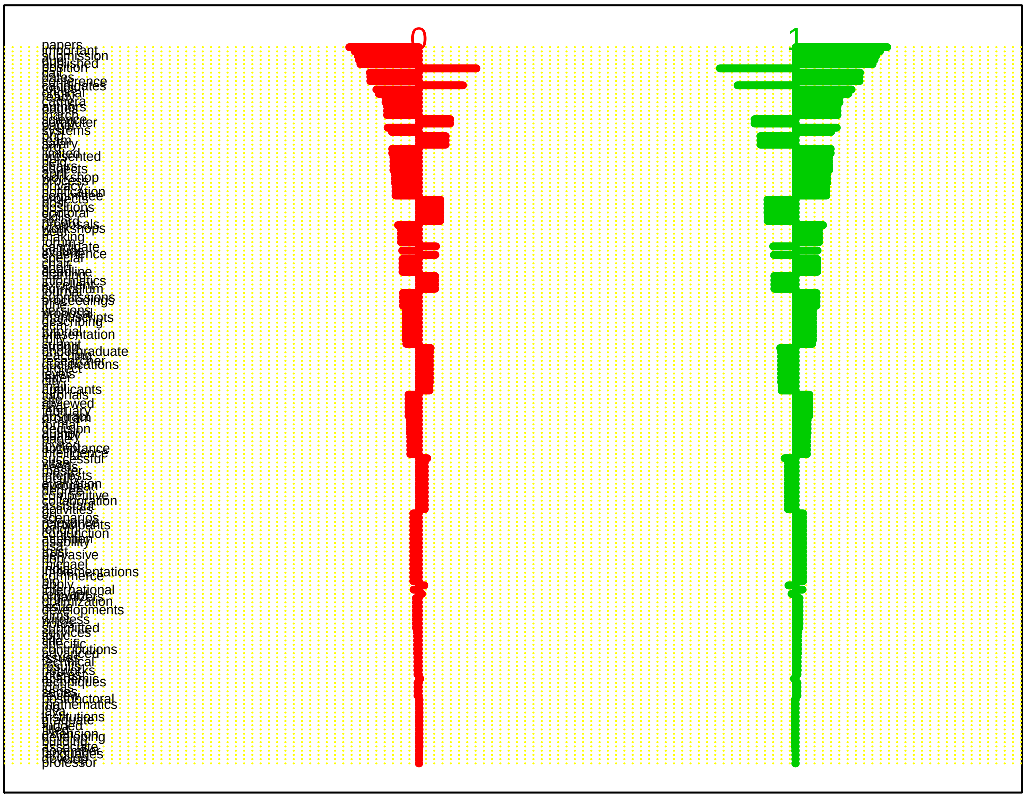
1.Perform nearest shrunken centroid classification of training data in which the threshold is chosen by cross-validation. Provide a centroid plot and interpret it. How many features were selected by the method? List the names of the 10 most contributing features and comment whether it is reasonable that they have strong effect on the discrimination between the conference mails and other mails? Report the test error.

We have performed the NSC classification model using the function `pamr.train()` and then, the obtained model has been used in the `pamr.cv()` function in order to choose the optimal threshold by cross validation.

From the following plot it can be seen graphically that the lowest misclassification error is obtained when the value of the threshold is 1.3 or 1.4. Also, we can observe that the number of features selected decreases as the threshold increases, so we will chose a threshold value of 1.4 because by selecting 170 features we keep having the minimum error.



Also, we will provide a centroid plot of the NSC model fitted with the optimal threshold value:



The function `pamr.plotcen()` plots the shrunken class centroids for each class, in our case class 1 is "announces of conferences" and class 0 is "everything else". We can see two centroids profiles d_{kj} for the data. The red and green bars are shrunken versions d'_{kj} of these centroids, obtained by soft-thresholding using $\Delta = 1.4$.

The list of the names of the 10 most contributing features is as follows:

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

It seams reasonable that those features have strong effect on the discrimination between the conference mails and other mails as they are all features that can be related to a conference. For example, submission may has to do with the submission of projects that will be presented during the conference, published may concern published papers, dates are also and important thing about a conference and of course the candidates that will participate there. So, it seems reasonalbe that all those features played a leading role in the classification of the mails.

Finally, the misclassification error for the test dataset is:

```
## [1] 0.1
```

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?

The elastic net model has been performed by using the function `glmnet()` and then, the obtained model has been used in the `cv.glmnet()` function in order to choose the optimal lambda (the penalty factor) by cross validation. This value can be obtained with `cv.net_fit$lambda.min` and in this case $\lambda_{min} = 0.1655087$. So, the optimal elastic net model is the one fitted with this lambda.

The SVM model with "vanilladot" kernel has been performed with the function `ksvm()` from the package **kernlab**. We have set in the *kernel* argument that the kernel function we want to use is the vanilladot (linear kernel).

The total number of contributing features is as follows:

```
## In the elastic net model is: 32
```

```
## In the SVM model is: 43
```

Finally, the misclassification errors for the test dataset is:

```
## For the elastic net model: 0.1
```

```
## For the SVM model: 0.05
```

- Comparison with results in step (1).

```
##          Model Number of features Test error
## 1          NSC                170        0.1
## 2 Elastic net                  32        0.1
## 3          SVM                  43       0.05
```

The Support vector machine with "vanilladot" kernel method is the best model: it has a few more features than the Elastic net model but the misclassification error for the test data is 0.05 units lower. The nearest shrunken centroid classification method is the worst among these three models because although it has the same test error as the elastic net model, it has model 6 time more features and thus it's a much more complex 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.

The Benjamini-Hochberg method has been perform in order to asses the significance of each of the $M = 4702$ features. First, we have computed a p-value for each feature using the theoretical t-distribution probability, obtained using the function `t.test()`. Then, using this set of p-values, we have tested the followiing hypothesis:

$$H_{0j}: \text{ feature j has no "effect" on Conference}$$

$$H_{1j}: \text{ feature j has an "effect" on Conference}$$

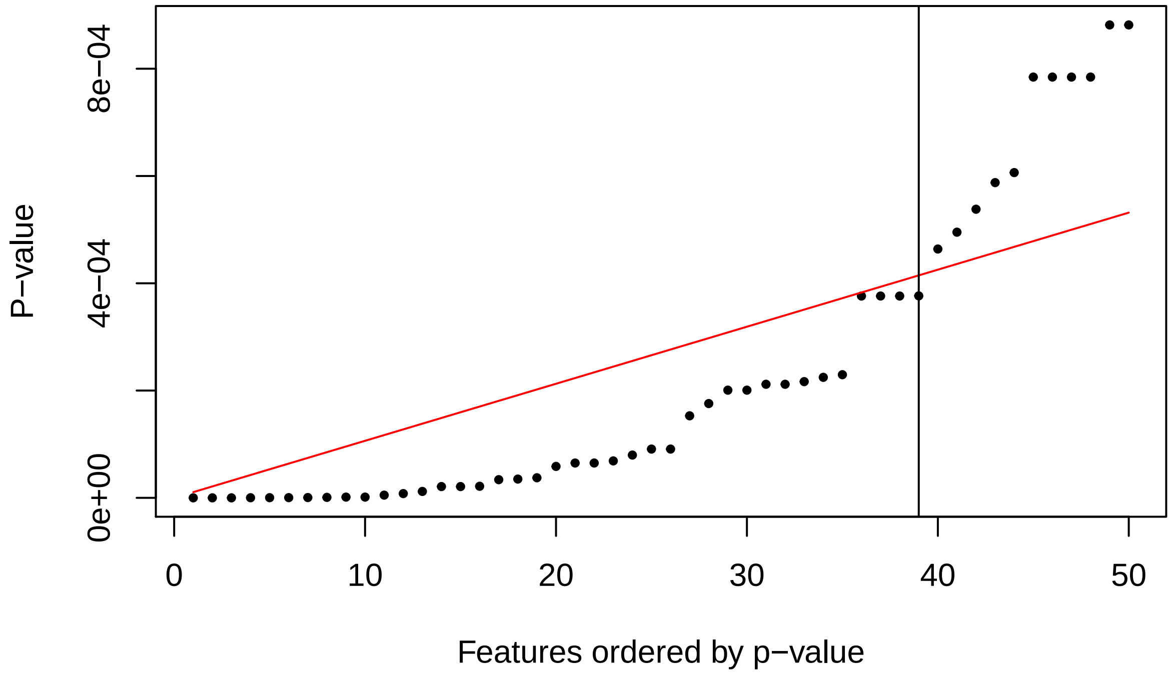
for all j= $1, 2, \ldots, M$.

Given that the BH threshold is the largest j for which the p-value is smaller than $\alpha\frac{j}{M}$, setting $\alpha = 0.05$, we will reject those hypothesis with a p-value smaller than the threshold and so, we will conclude that the features associated to those hypothesis are not significant to explain our target variable "Conference".

After performing the method, we observe that 39 features have no effect on Conference. These features and its p-values are the following:

```
##        Features       pvalues      Features       pvalues
## 1        papers 1.116910e-10           due 6.488781e-05
## 2    submission 7.949969e-10      original 6.488781e-05
## 3      position 8.219362e-09  notification 6.882210e-05
## 4     published 1.835157e-07        salary 7.971981e-05
## 5     important 3.040833e-07        record 9.090038e-05
## 6          call 3.983540e-07        skills 9.090038e-05
## 7    conference 5.091970e-07          held 1.529174e-04
## 8    candidates 8.612259e-07          team 1.757570e-04
## 9         dates 1.398619e-06         pages 2.007353e-04
## 10        paper 1.398619e-06      workshop 2.007353e-04
## 11       topics 5.068373e-06     committee 2.117020e-04
## 12      limited 7.907976e-06   proceedings 2.117020e-04
## 13    candidate 1.190607e-05         apply 2.166414e-04
## 14       camera 2.099119e-05        strong 2.246309e-04
## 15        ready 2.099119e-05 international 2.295684e-04
## 16      authors 2.154461e-05        degree 3.762328e-04
## 17          phd 3.382671e-05     excellent 3.762328e-04
## 18     projects 3.499123e-05          post 3.762328e-04
## 19          org 3.742010e-05     presented 3.765147e-04
## 20       chairs 5.860175e-05          <NA>         <NA>
```

Finally, we can plot the slope ($\frac{0.05}{4702} \cdot j$) and the ordered p-values:



We observe that the 39th feature is the largest j for wich the p-value falls bellow the line.

Appendix.

Assignment 1.

Question 1

```
# 0. Importing the data:
library(readxl)
data <- read_excel("influenza.xlsx")

# 1. Time series plot.
library(ggplot2)

# 1.1  Data frame to use in ggplot:
df <- data.frame(Time= data$Time, values= c(data$Mortality, data$Influenza),
                 variable  = rep(c("Mortality", "Influenza"), each=459))

# 1.2. Multiple line plot:
ggplot(df, aes(x = Time, y = values)) +
  geom_line(aes(color = variable), size = 1) +
  scale_color_manual(values = c("#00AFBB", "#E7B800")) +
  theme_minimal()
```

Question 2

```
# 2. Fitting a GAM model.
library(mgcv)
model <- gam(data$Mortality ~ data$Year + s(data$Week), family = gaussian(),
             data=data, method = "GCV.Cp" )
# s() sets up a model using spline-based smooths
# "GCV.Cp" to use generalized cross-validation for unknown scale parameter
```

Question 3

```
# 3.1. Predicted mortality.
pred <- gam_fit$fitted.values

# 3.2  Data frame to use in ggplot:
df <- data.frame(Time= data$Time, Mortality= c(data$Mortality, pred),
                 Values  = rep(c("Observed", "Predicted"), each=459))

# 3.3. Plot of predicted and observed mortality against time:
ggplot(df, aes(x = Time, y = Mortality)) +
  geom_line(aes(color = Values), size=0.7) +
  theme_minimal()

# 3.4. Output of the GAM model:
summary(gam_fit)

# 3.5. Plot of the spline component:
plot(gam_fit, xlab="Week", ylab="Spline of Week")
```

Question 4

```
# 4.1. Fitting GMA models with different smoothing penalty factors.
penalty <- seq(from=0, to= 10, by=0.5)

dev_values <- vector()
d_freedom <- vector()
for(i in penalty){
  gam_fit <- gam(data$Mortality ~ data$Year +
                   s(data$Week, k=length(unique(data$Week)), sp=i),
                 family = gaussian(), data=data, method = "GCV.Cp" )
  dev_values <- c(dev_values, gam_fit$deviance)
  d_freedom <- c(d_freedom1, (nrow(data)-gam_fit$df.residual))
}

# 4.2. Plots of penalty factor vs deviance and vs degrees of freedom.
par(mfrow=c(1,2), mar=c(3.9, 3.8, 1, 1) + 0.1)
plot(penalty, dev_values, type="b", col="blue", xlab="Penalty Factor",
     ylab="Deviance")
plot(penalty, d_freedom, type="b", col="blue", xlab="Penalty Factor",
     ylab="Degrees of freedom")

# 4.3. Plot of the predicted and observed mortality against time for high penalty factor.
gam_fit_high <- gam(data$Mortality ~ data$Year +
                      s(data$Week, k=length(unique(data$Week)), sp=10),
                    family = gaussian(), data=data, method = "GCV.Cp" )
pred_high <- gam_fit_high$fitted.values

df_high <- data.frame(Time= data$Time, Mortality= c(data$Mortality, pred_high),
                 Values  = rep(c("Observed", "Predicted"), each=459))

ggplot(df_high, aes(x = Time, y = Mortality)) +
  geom_line(aes(color = Values), size=0.7) +
  theme_minimal() + ggtitle("Prediction vs Observations for high penalty factor")

# 4.4. Plot of the predicted and observed mortality against time for low penalty factor.
gam_fit_low <- gam(data$Mortality ~ data$Year +
                     s(data$Week, k=length(unique(data$Week)), sp=0.1),
                   family = gaussian(), data=data, method = "GCV.Cp" )
pred_low <- gam_fit_low$fitted.values

df_low<- data.frame(Time= data$Time, Mortality= c(data$Mortality, pred_low),
                 Values  = rep(c("Observed", "Predicted"), each=459))

ggplot(df_low, aes(x = Time, y = Mortality)) +
  geom_line(aes(color = Values), size=0.7) +
  theme_minimal() + ggtitle("Prediction vs Observations for low penalty factor")
```

Question 5

```
# 5 Plot of the residuals and the influenza values against time
df <- data.frame(Time= data$Time, values= c(gam_fit$residuals, data$Influenza),
                 variable  = rep(c("Residuals", "Influenza"), each=459))

ggplot(df, aes(x = Time, y = values)) +
  geom_line(aes(color = variable), size = 0.8) +
  theme_minimal()+ ggtitle("Residuals and Influenza vs Time")
```

Question 6

```
# 6.1. Fitting the GAM model.
add_gam_fit <- gam(data$Mortality ~ s(data$Year, k=length(unique(data$Year))) +
                     s(data$Week, k=length(unique(data$Week))) +
                     s(data$Influenza, k=length(unique(data$Influenza))) ,
                   family = gaussian(), data=data, method = "GCV.Cp" )
```

```
# 6.2. Model autput.
summary(add_gam_fit)

# 6.3. Plot of the original and fitted Mortality against Time
pred <- add_gam_fit$fitted.values

df <- data.frame(Time= data$Time, Mortality= c(data$Mortality, pred),
                 Values  = rep(c("Original", "Fitted"), each=459))

ggplot(df, aes(x = Time, y = Mortality)) +
  geom_line(aes(color = Values), size = 0.8) +
  theme_minimal()+ ggtitle("Original and fitted Mortality against Time")
```

Assignment 2.

Question 1

```
# 1.1. Importing the data:
data <- read.csv(file = "data.csv", sep=";", header=TRUE, stringsAsFactors=FALSE,
                 fileEncoding="latin1",)
data$Conference <- as.factor(data$Conference)

# 1.2. Divide data into training and test sets:
n=dim(data)[1]
set.seed(12345)
id=sample(1:n, floor(n*0.7))
train=data[id,]
test=data[-id,]

# 1.3. Nearest shrunken centroid classification with threshold chosen by cross-validation.
library(pamr)

rownames(train)=1:nrow(train)
x_train=t(train[,-4703])
y_train=train[[4703]]
mydata_train=list(x=x_train,y=as.factor(y_train),geneid=as.character(1:nrow(x_train)),
                  genenames=rownames(x_train))

## Fitting the NSC model
model = pamr.train(mydata_train, threshold=seq(0, 4, 0.1))

## Selecting the optimal threshold by cross-validation.
cvmodel = pamr.cv(model,mydata_train)
optimal <- cvmodel$threshold[which(min(cvmodel$error)==cvmodel$error)]

## Fitting the optimal NSC model
optimal_model = pamr.train(mydata_train, threshold=1.4)

pamr.plotcv(cvmodel)

# 1.4. Centroid plot
pamr.plotcen(optimal_model, mydata_train, threshold=1.4)
```

```r
# 1.5. Names of the 10 most contributing features.
contribution <- pamr.listgenes(optimal_model, mydata_train,  threshold=1.4)
df <- as.data.frame(colnames(data)[as.numeric(contribution[,1])][1:10])
colnames(df) <- "Features"
df

# 1.6. Test error
x_test=t(test[,-4703])
test_pred <- pamr.predict(fit=optimal_model, newx= x_test, threshold=1.4)
mean(test_pred != test$conference)
```

Question 2

```r
# 2.0. Variables:
x_train <- as.matrix(train[,-4703])
y_train <- train[,4703]

x_test <- as.matrix(test[,-4703])
y_test <- test[,4703]

# 2.1. Elastic net with binomial response
library(glmnet)
net_fit <- glmnet(x_train, y_train, alpha = 0.5, family = "binomial")
cv.net_fit <- cv.glmnet(x_train, y_train, alpha = 0.5, family="binomial")
net_fit_optimal <- glmnet(x_train, y_train, alpha = 0.5, family = "binomial",
                          lambda = cv.net_fit$lambda.min)

# 2.2. SPV ith vanilladot kernel.
# https://www.rdocumentation.org/packages/kernlab/versions/0.9-29/topics/ksvm
# https://www.rdocumentation.org/packages/kernlab/versions/0.9-29/topics/ksvm-class
library(kernlab)
svm_fit <- ksvm(x_train, y_train, scale=FALSE, kernel="vanilladot")

# 2.3. Most contributing features
contributing_net <- colnames(data)[which(net_fit_optimal$beta !=0 )]
coeff_net <- net_fit_optimal$beta[which(net_fit_optimal$beta !=0 )]
cat("The number of contributing features in the elastic net model is:",
    length(contributing_net), "\n")

contributing_svm <- colnames(data)[SVindex(svm_fit)]
coeff_svm <- coef(svm_fit)[[1]]
cat("The number of contributing features in the SVM model is:",
    length(contributing_svm), "\n")

df <- as.data.frame(cbind(contributing_net[1:10], round(coeff_net[1:10],4),
                          contributing_svm[1:10], round(coeff_svm[1:10],5)))
colnames(df) <- c("Elastic net","Coefficients", "SVM","Coefficients")
df

# 2.4. Test errors.
pred_net <- predict(net_fit_optimal, x_test, type = "class")
pred_svm <- predict(svm_fit, x_test, type = "response")
cat("For the elastic net model:", mean(pred_net != test$conference), "\n")
cat("For the SVM model:", mean(pred_svm != test$conference), "\n")
```

```r
# 2.5. Comparison table.
df <- as.data.frame(cbind(c("NSC", "Elastic net", "SVM"), c(170,32,43),
                          c(0.3,0.3,0.25)))
colnames(df) <- c("Model", "Number of features", "Test error")
df
```

Question 3

```r
# Variables from the original data:
x <- as.matrix(data[,-4703])
y <- as.factor(data[,4703])

# Computing p-value for each variable (1,2,..,p)
alpha <- 0.05
M <- ncol(x)
pval <- data.frame("Features"=colnames(x), "pvalues"=rep(0,M))

for(j in 1:M){
  ttest <- t.test(x[,j] ~ y)
  pval[j,2] <- ttest$p.value
}

# Ordering p-values
sort_pval <- pval[order(pval$pvalues, decreasing = F),]
sort_pval$Index <- 1:M
sort_pval$critical <- (sort_pval$Index *alpha)/M

# Non significant features:
# max(sort_pval$Index[which(sort_pval$pvalues < sort_pval$critica)]) =39
sort_pval[which(sort_pval$pvalues < sort_pval$critica),1:2]

# Plot:
{plot(1:50, sort_pval$pvalues[1:50], pch=19, cex=0.5, ylab="P-value",
      xlab="Features ordered by p-value")
points(sort_pval$critical[1:50], type="l", col="red")
abline(v=39)}
```