machine learning(732A99) lab2 block2

Anubhav Dikshit(anudi287)

17 December 2018

Contents

Assignment 1 **2**

Loading The Libraries . . . 2

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. . . . 2
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. . . . 4
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. . . . 5
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? . . . 9
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? . 13
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. . . . 14

Assignment 2 **16**

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. . . . 16
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? . . . 22
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. . . . 25

Appendix **26**

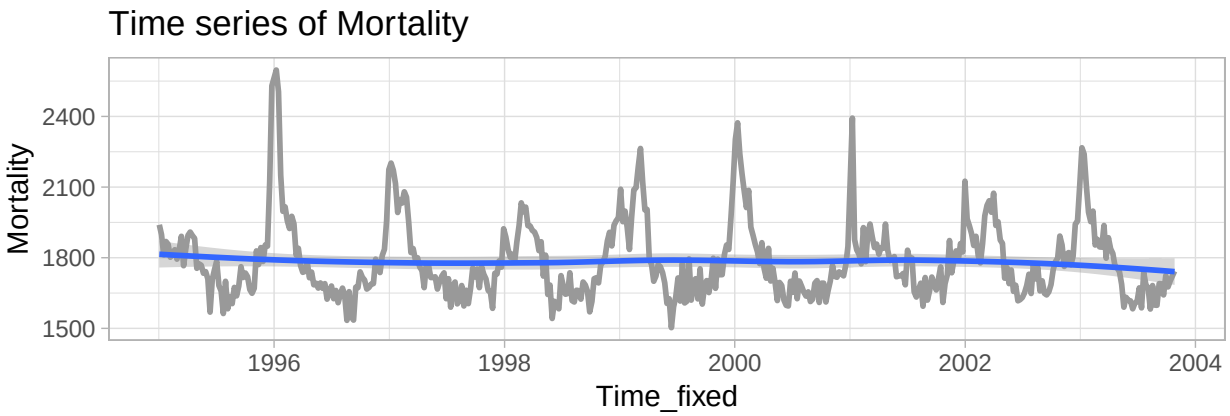
Assignment 1

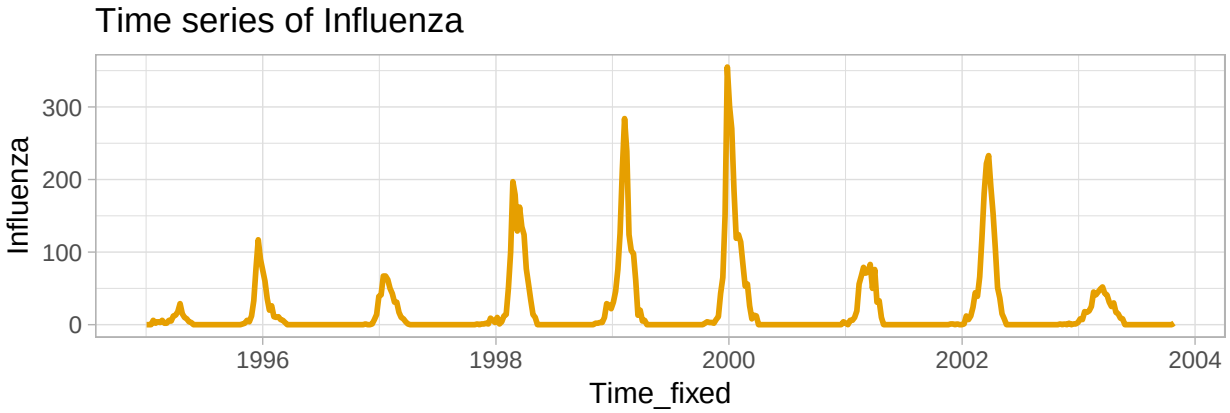
Loading The Libraries

```
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, width=80)

# 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))
```

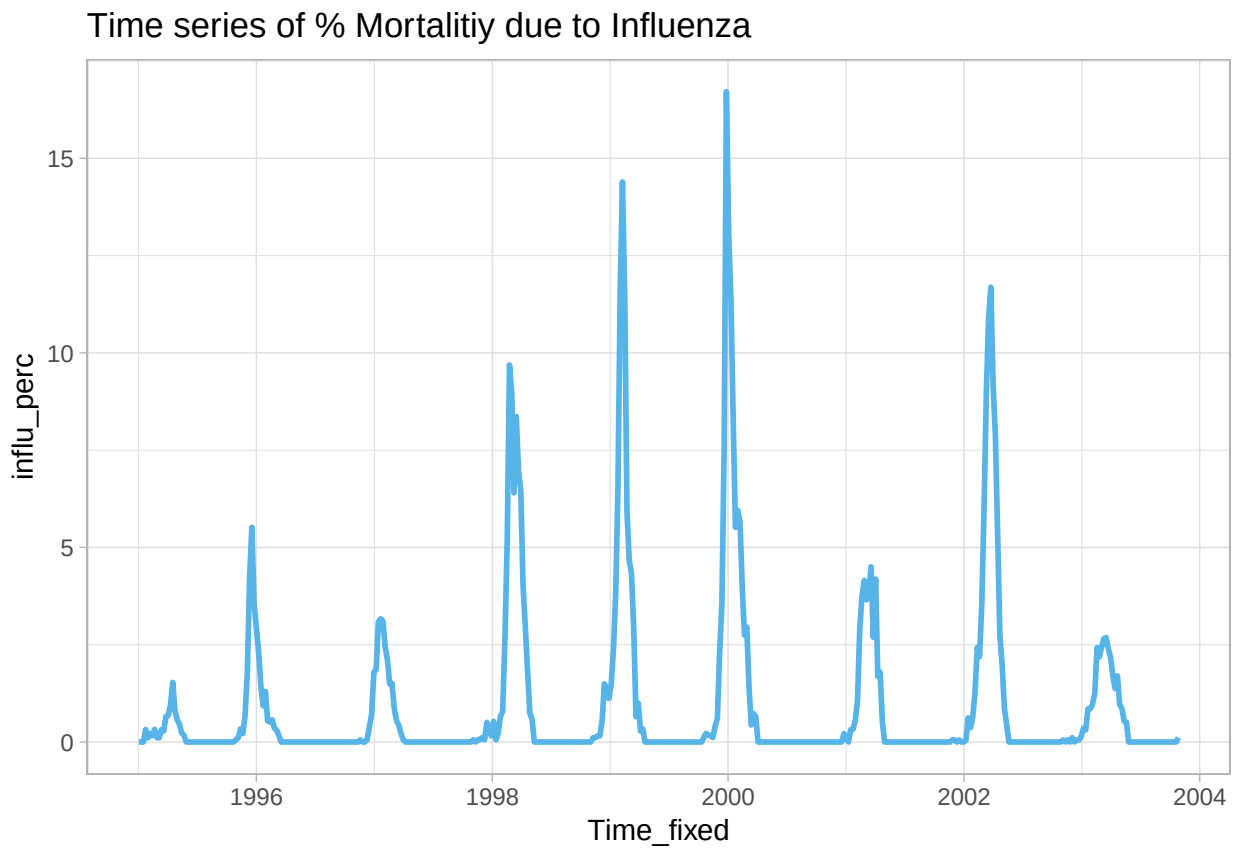
1. Use time series plots to visually inspect how the mortality and influenza number vary with time (use Time as X axis). By using this plot, comment how the amounts of influenza cases are related to mortality rates.

```
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) +
  geom_smooth(method = "loess") +
    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
Mortality
2400
2100
1800
1500
1996
1998
2000
2002
2004
Time_fixed


Time series of Influenza
Influenza
300
200
100
0
1996
1998
2000
2002
2004
Time_fixed

```
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
```

Analysis:

Using the default parameter settings within the *gam*-function implies that *Mortality* is normally distributed (*family=gaussian()*). Also, since *method = "GCV.Cp"*, this leads to the usage of GCV (*Generalized Cross Validation score*) related to the smoothing parameter estimation. The underlying probabilistic model can be written as:

$$Mortality = N(\mu, \sigma^2)$$

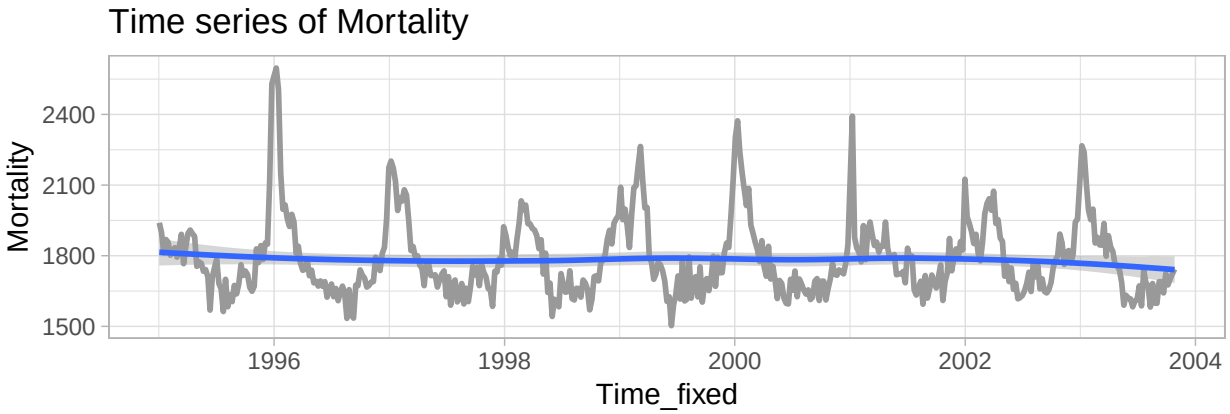
$$\hat{Mortality} = Intercept + \beta_1 Year + s(Week) + \epsilon$$

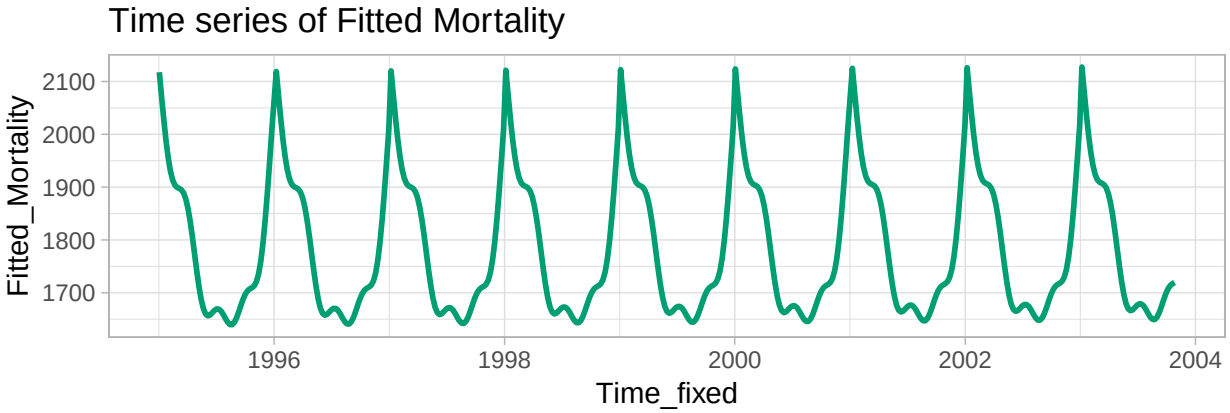
where

$$\epsilon = N(0, \sigma^2).$$

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.

```
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)
```

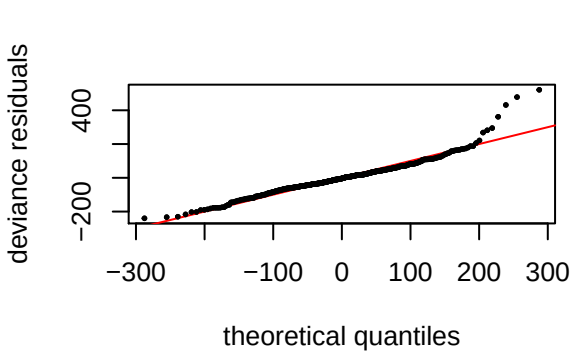


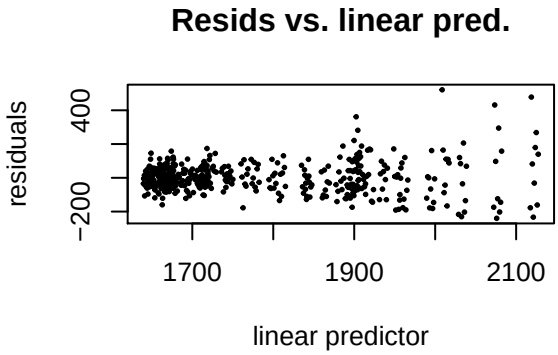


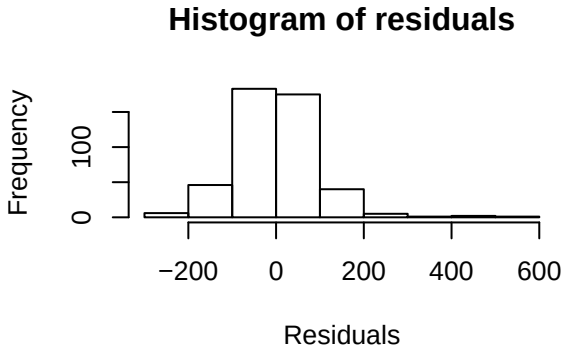

```
summary(gam_model)
```

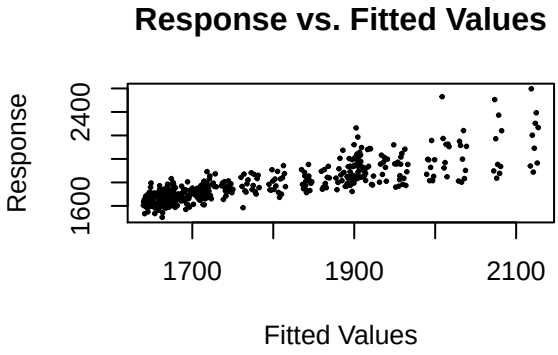
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Mortality ~ Year + s(Week)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -652.060   3448.379  -0.189     0.85
## Year           1.219      1.725   0.706     0.48
##
## Approximate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



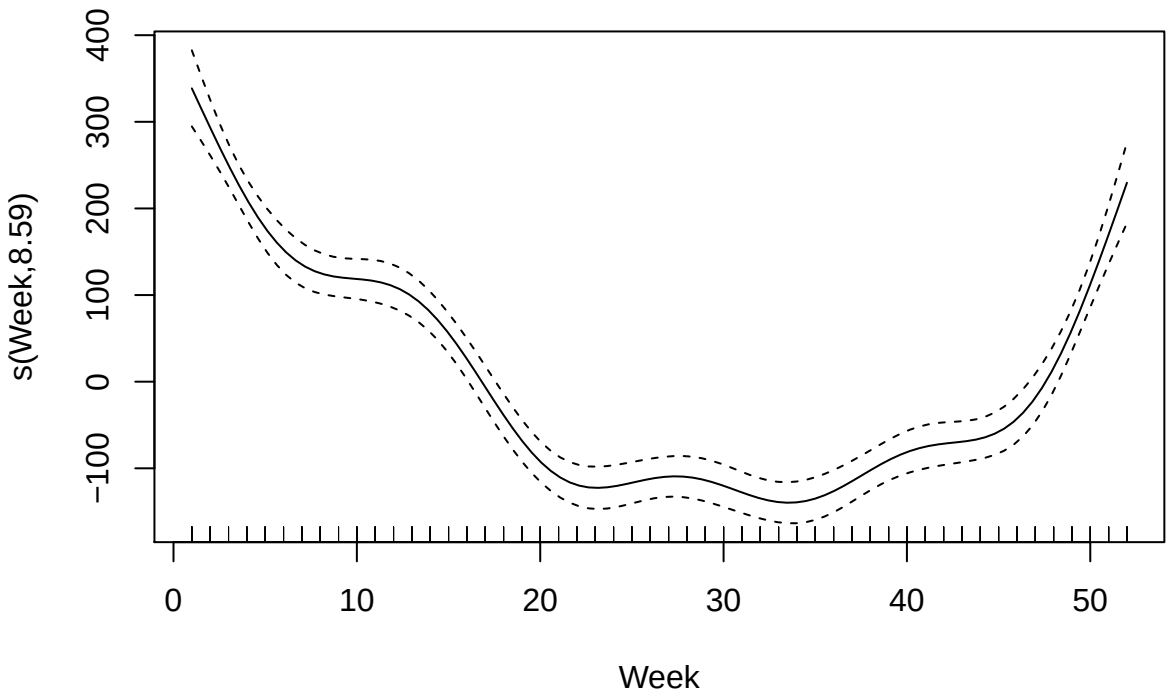







```
##
## Method: GCV   Optimizer: magic
## Smoothing parameter selection converged after 7 iterations.
## The RMS GCV score gradient at convergence was 0.114505 .
## The Hessian was positive definite.
## Model rank =  11 / 11
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## 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
```

```
plot(gam_model)
```




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

Analysis:

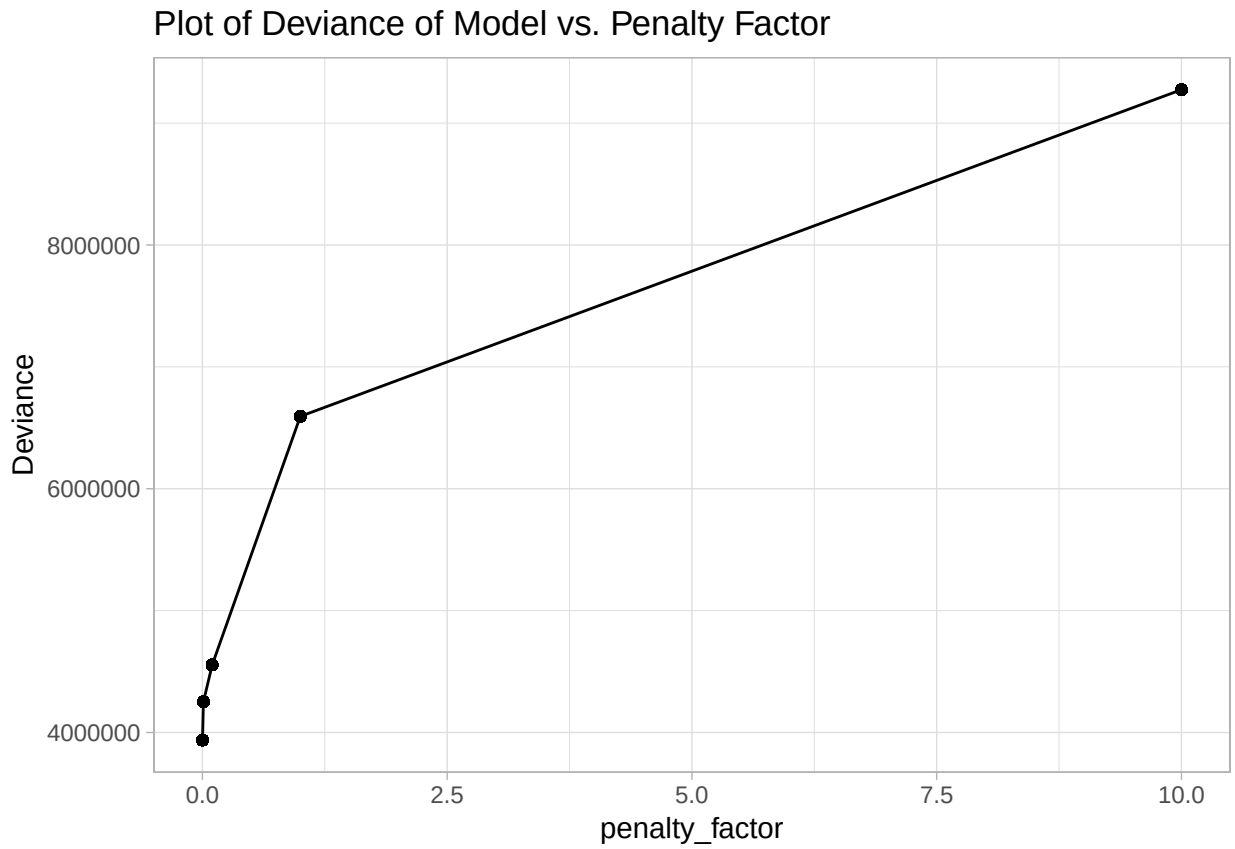
We see that the predicted mortality a cyclical component (repeating function), while the true mortality varyies with time. Thus this is not a very good model this is further supported by the fact that our adjusted R^2 is just 66.1%, thus our model accounts for only about 66% of varience.

From the model summary we can see that the p-value of the spline of Week is less than 0.05 while the p-values of Intercept and Year is more than 0.5 (A low p-value (< 0.05) indicates that you can reject the null hypothesis).We see that there is a trend component of Mortality is overall decreasing, we also see that peaks of mortality decreases thrice and rise again.

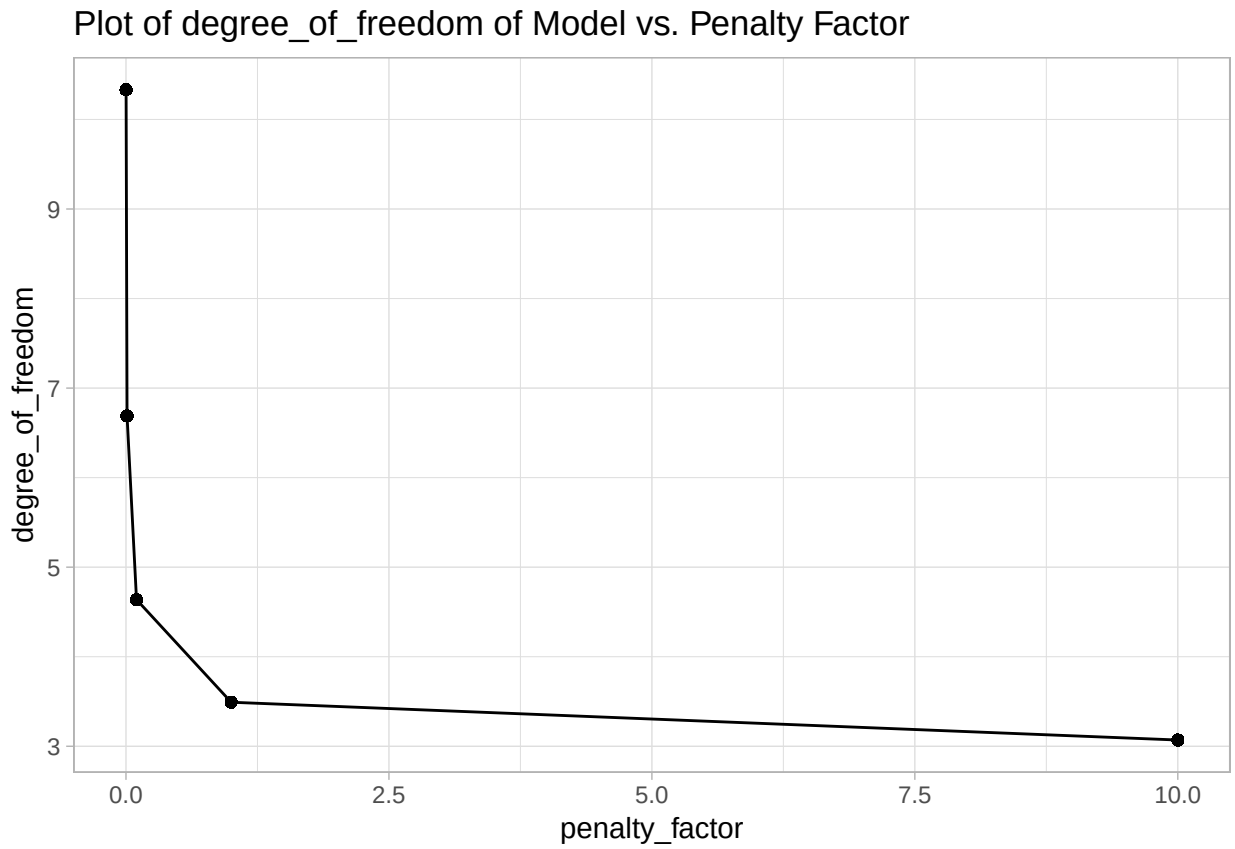
From the plot of spline component, we can see that there are 5 knots or 5 components. Clearly, at the beginning and end of the year mortality rates are very much higher than in the middel of the year. When one thinks of this, this makes sense. Influenzia affects people more in winter periods, thus the beginning and end of the calendar year, whereas in summer, the middle of the calendar year, people suffer less from influenzia, and thus less people die.

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?

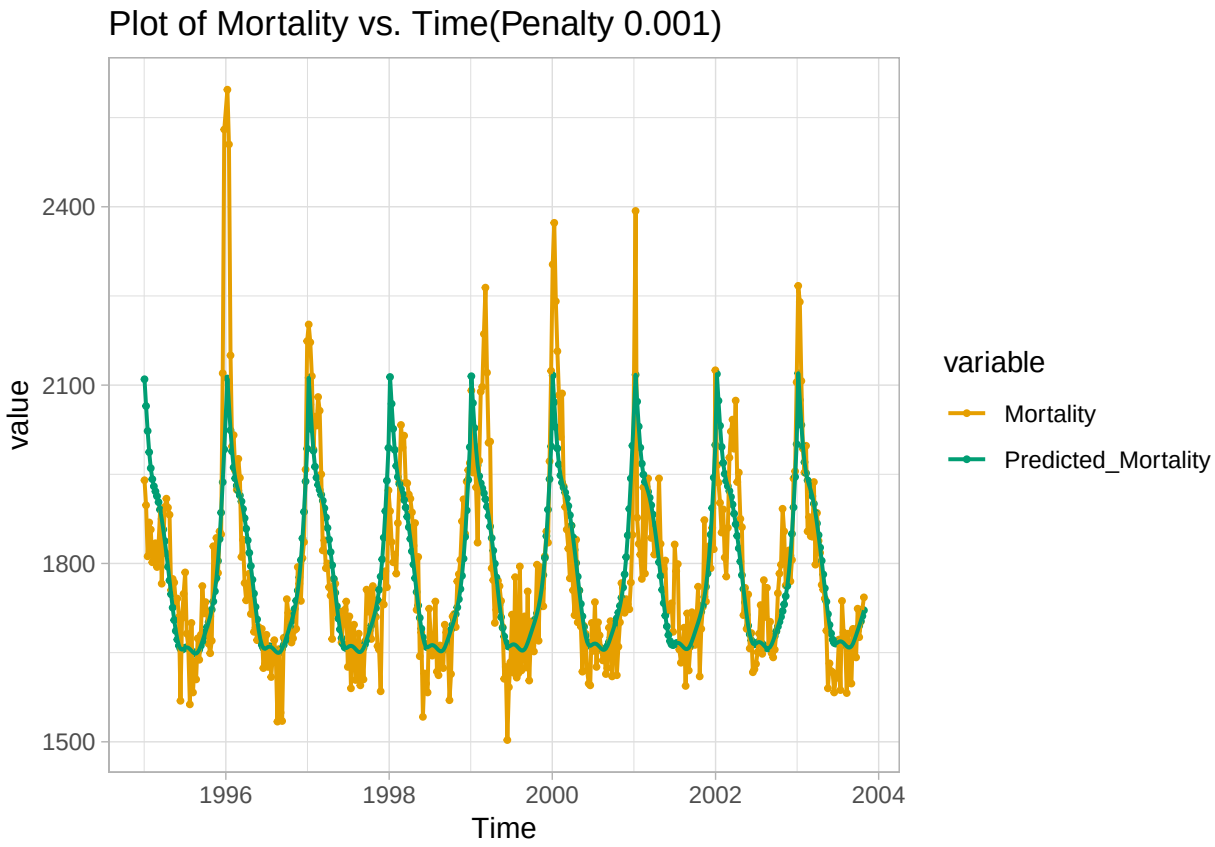
```
model_deviance <- NULL
for(sp in c(0.001, 0.01, 0.1, 1, 10))
{
  k=length(unique(flu_data$Week))

gam_model <- mgcv::gam(data = flu_data, Mortality~Year+s(Week, k=k, sp=sp), method = "GCV.Cp")
temp <- cbind(gam_model$deviance, gam_model$fitted.values, gam_model$y, flu_data$Time_fixed,
              sp, 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
```



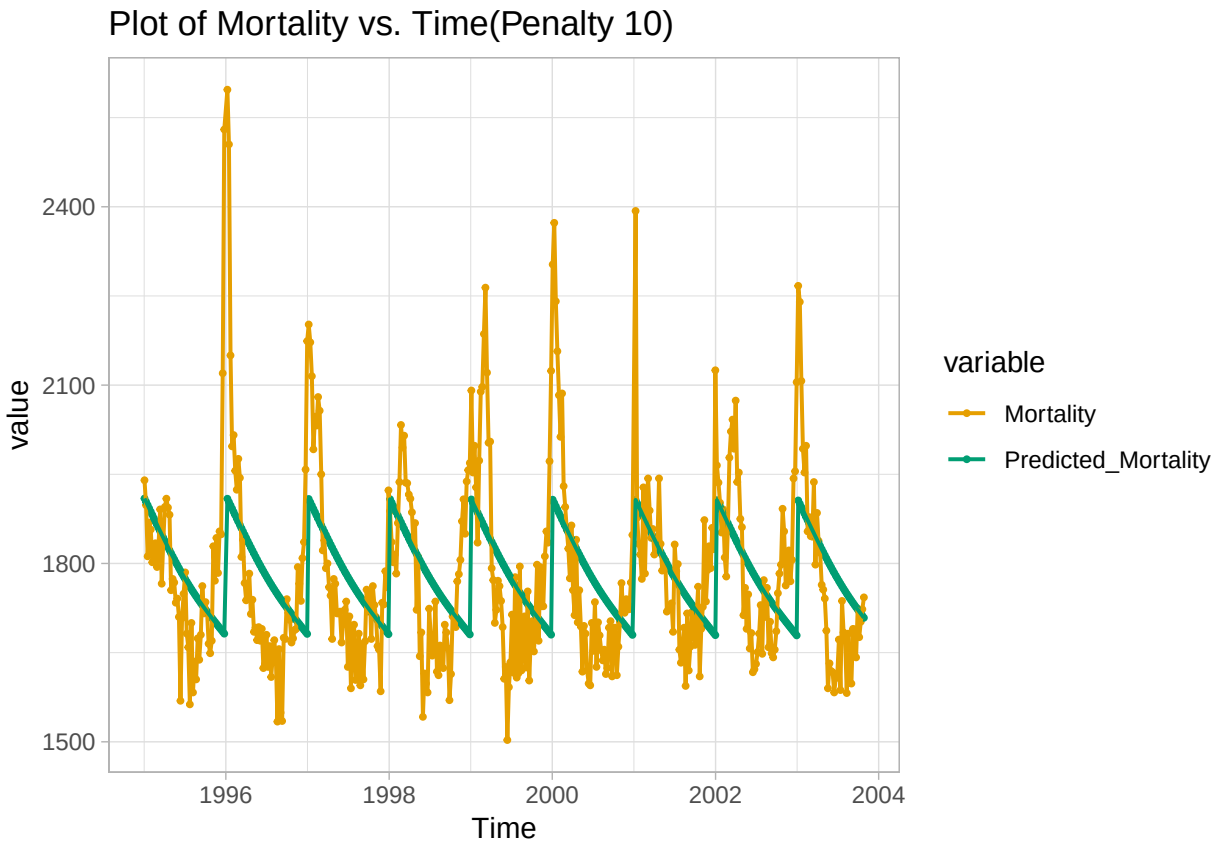

```
# plot of degree of freedom
p7 <- ggplot(data=model_deviance, aes(x = penalty_factor, y = degree_of_freedom)) +
geom_point() +
  geom_line() +
      theme_light() +
ggtitle("Plot of degree_of_freedom of Model vs. Penalty Factor")
p7
```




```
model_deviance_wide <- melt(model_deviance[,c("Time", "penalty_factor",
                                              "Mortality", "Predicted_Mortality")],
                            id.vars = c("Time", "penalty_factor"))

# plot of predicted vs. observed mortality
p8 <- ggplot(data=model_deviance_wide[model_deviance_wide$penalty_factor == 0.001,],
             aes(x= Time, y = value)) +
  geom_point(aes(color = variable), size=0.7) +
  geom_line(aes(color = variable), size=0.7) +
  scale_color_manual(values=c("#E69F00", "#009E73")) +
  theme_light() +
  ggtitle("Plot of Mortality vs. Time(Penalty 0.001)")

p9 <- ggplot(data=model_deviance_wide[model_deviance_wide$penalty_factor == 10,],
             aes(x= Time, y = value)) +
  geom_point(aes(color = variable), size=0.7) +
    geom_line(aes(color = variable), size=0.7) +
  scale_color_manual(values=c("#E69F00", "#009E73")) +
    theme_light() +
  ggtitle("Plot of Mortality vs. Time(Penalty 10)")

p8
```


Plot of Mortality vs. Time(Penalty 0.001)
value
2400
2100
1800
1500
1996
1998
2000
2002
2004
Time
variable
Mortality
Predicted_Mortality

```
p9
```



Analysis:

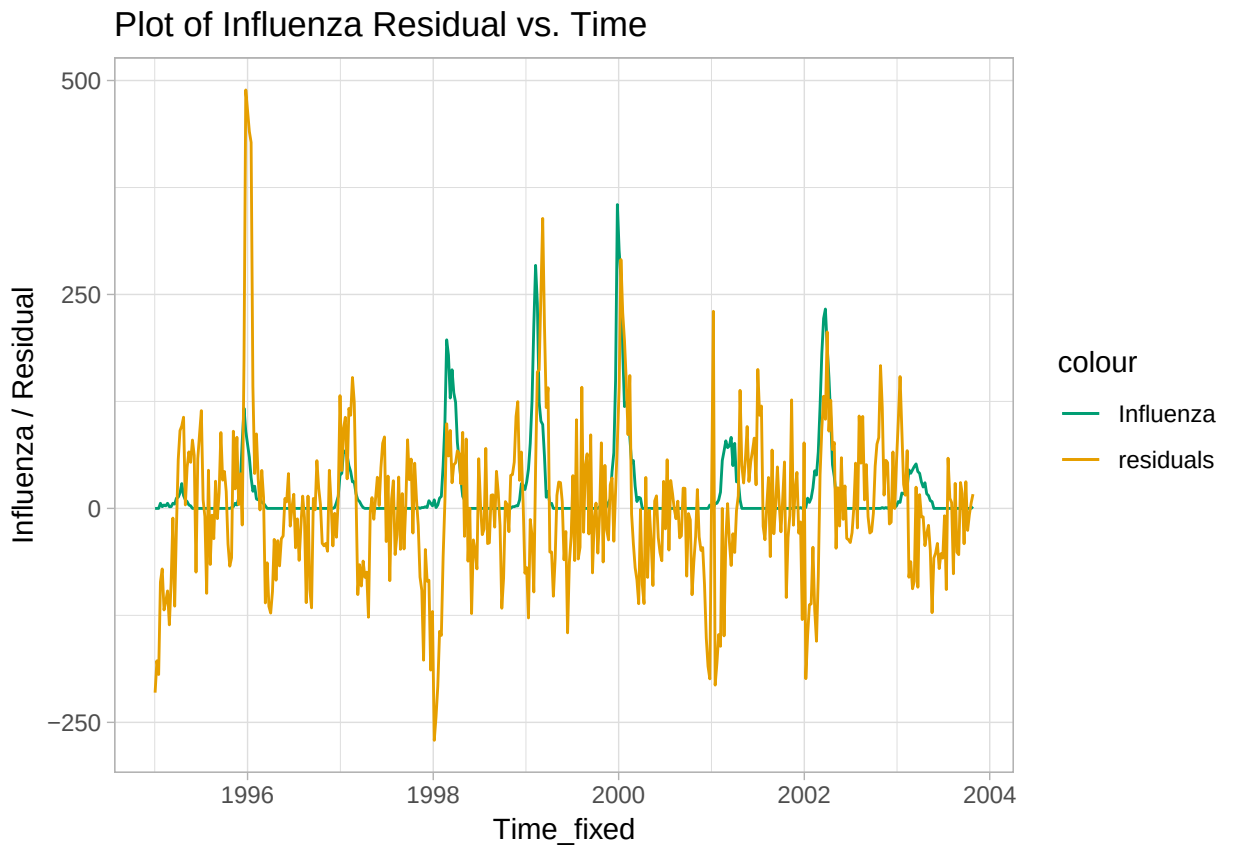
Penalty factor in the model determines the complexity of the model, higher the penalty factor the more the model will have bias and hence lesser the complexity. We can see that as the penalty factor increases the degree of freedom decreases.

From the plots of degree of freedom vs. penalty factor we see that our result to confirm our hypothesis.

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?

```
k=length(unique(flu_data$Week))
gam_model <- mgcv::gam(data = flu_data, Mortality~Year+s(Week, k=k), 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
```



Analysis:

Some of the beaks in Influenza outbreaks correspond to peaks in the residuals of the fitted model. Still, however, a lot of variance in the residuals is not correlated to Influenza outbreaks. Therefore, I would say that the Influenza outbreaks are not correlated to the residuals.

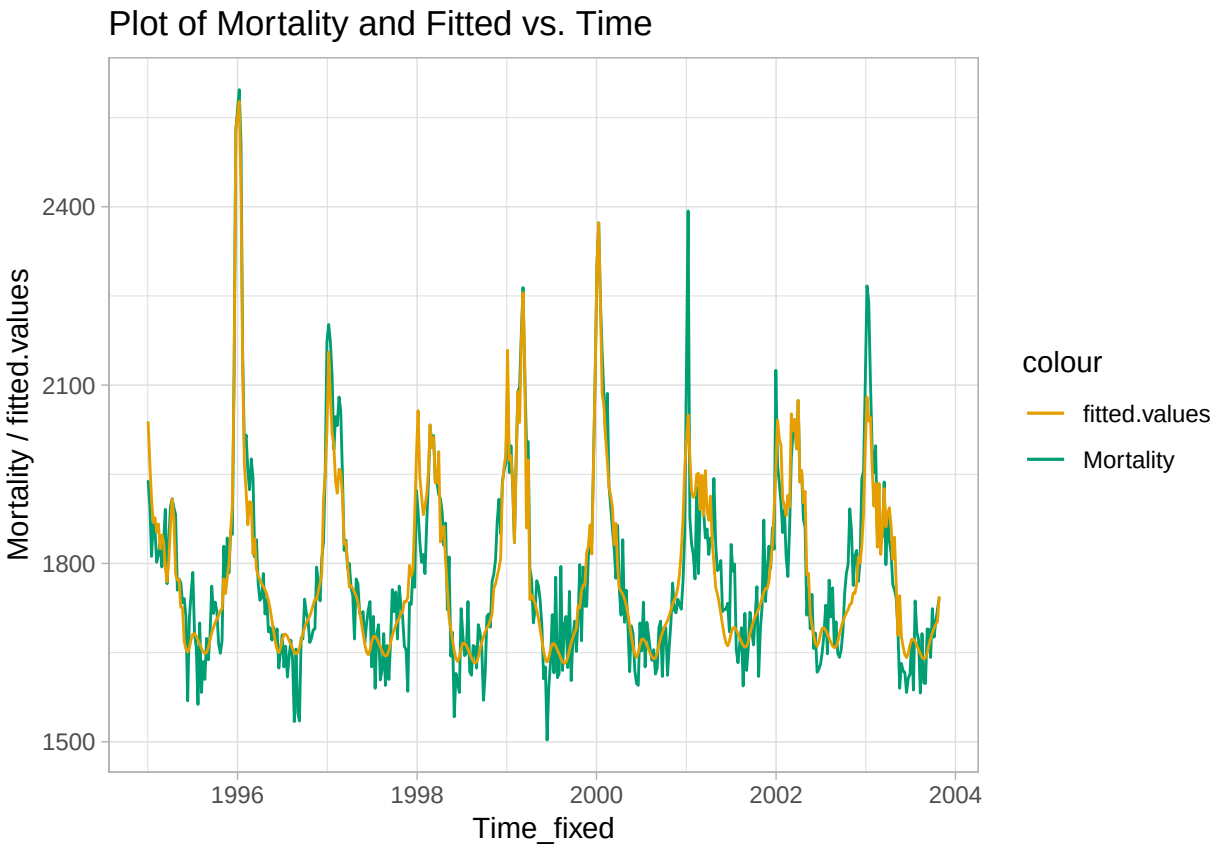
6. Fit a GAM model in R in which mortality is be modelled as an additive function of the spline functions of year, week, and the number of confirmed cases of influenza. Use the output of this GAM function to conclude whether or not the mortality is influenced by the outbreaks of influenza. Provide the plot of the original and fitted Mortality against Time and comment whether the model seems to be better than the previous GAM models.

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

k1 = length(unique(flu_data$Year))
k2 = length(unique(flu_data$Week))
k3 = length(unique(flu_data$Influenza))

gam_model_additive <- gam(Mortality ~ s(Year, k=k1) +
```

```
                                    s(Week, k=k2) +
                                   s(Influenza, k=k3),
                           data = flu_data)

summary(gam_model_additive)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Mortality ~ s(Year, k = k1) + s(Week, k = k2) + s(Influenza,
##     k = k3)
##
## Parametric coefficients:
##             Estimate Std. Error t value            Pr(>|t|)
## (Intercept)   1783.8        3.2   557.5 <0.0000000000000002 ***
## ---
## 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 <0.0000000000000002 ***
## s(Influenza) 69.737 72.854  5.599 <0.0000000000000002 ***
## ---
## 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
```

```
flu_data$fitted.values = gam_model_additive$fitted.values


p11 <- ggplot(data = flu_data, aes(x = Time_fixed)) +
  geom_line(aes( y = Mortality, color = "Mortality")) +
  geom_line(aes(y = fitted.values, color = "fitted.values")) +
      theme_light() +
  scale_color_manual(values=c(Mortality = "#009E73", fitted.values = "#E69F00")) +
  labs(y = "Mortality / fitted.values") +
  ggtitle("Plot of Mortality and Fitted vs. Time")

p11
```



Analysis:

The additive GAM model clearly has the best fit. Much of the variance of the data (81.9% is the adjusted R^2) is captured by the model. Given that the GAM models in step 2 and step 4 do not include the influenza variable from the dataset, and the the model above does, one can say that most likely mortality is influenced by the outbreaks of influenza.

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

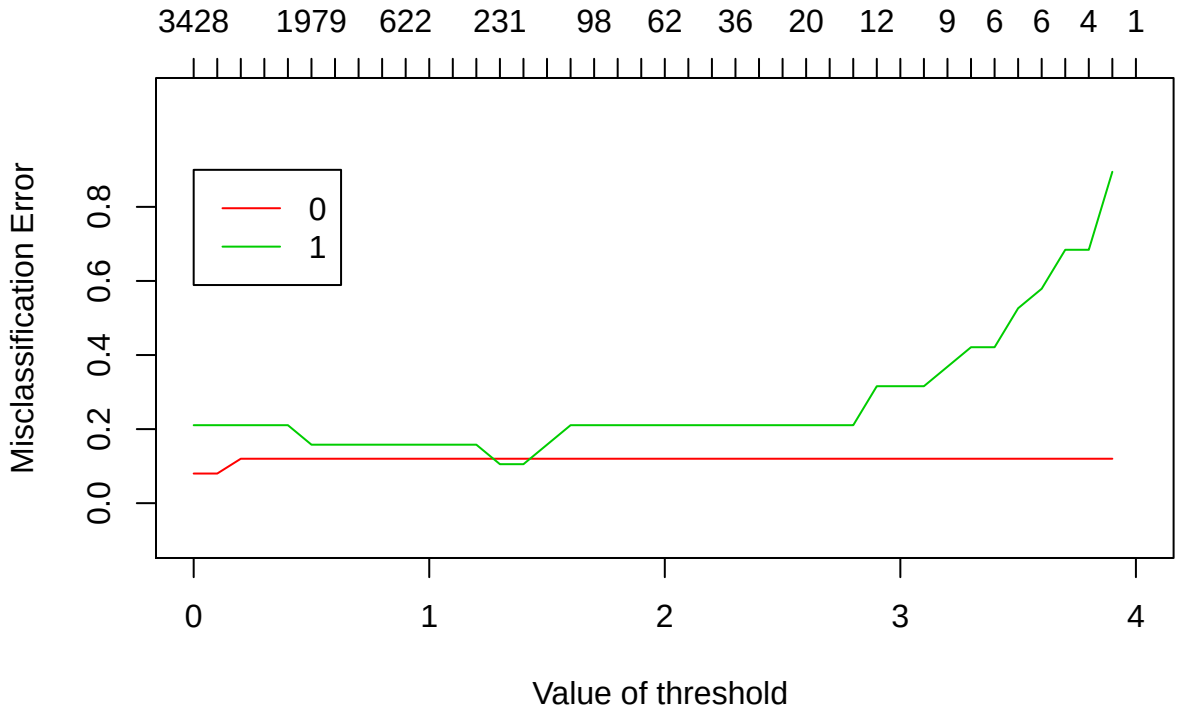
```
n=NROW(data)
data$Conference <- as.factor(data$Conference)
set.seed(12345)
```

```
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 = 1.3))
predicted_scc_test <- pamr.predict(model, newx = x_test, threshold = 1.3)
```

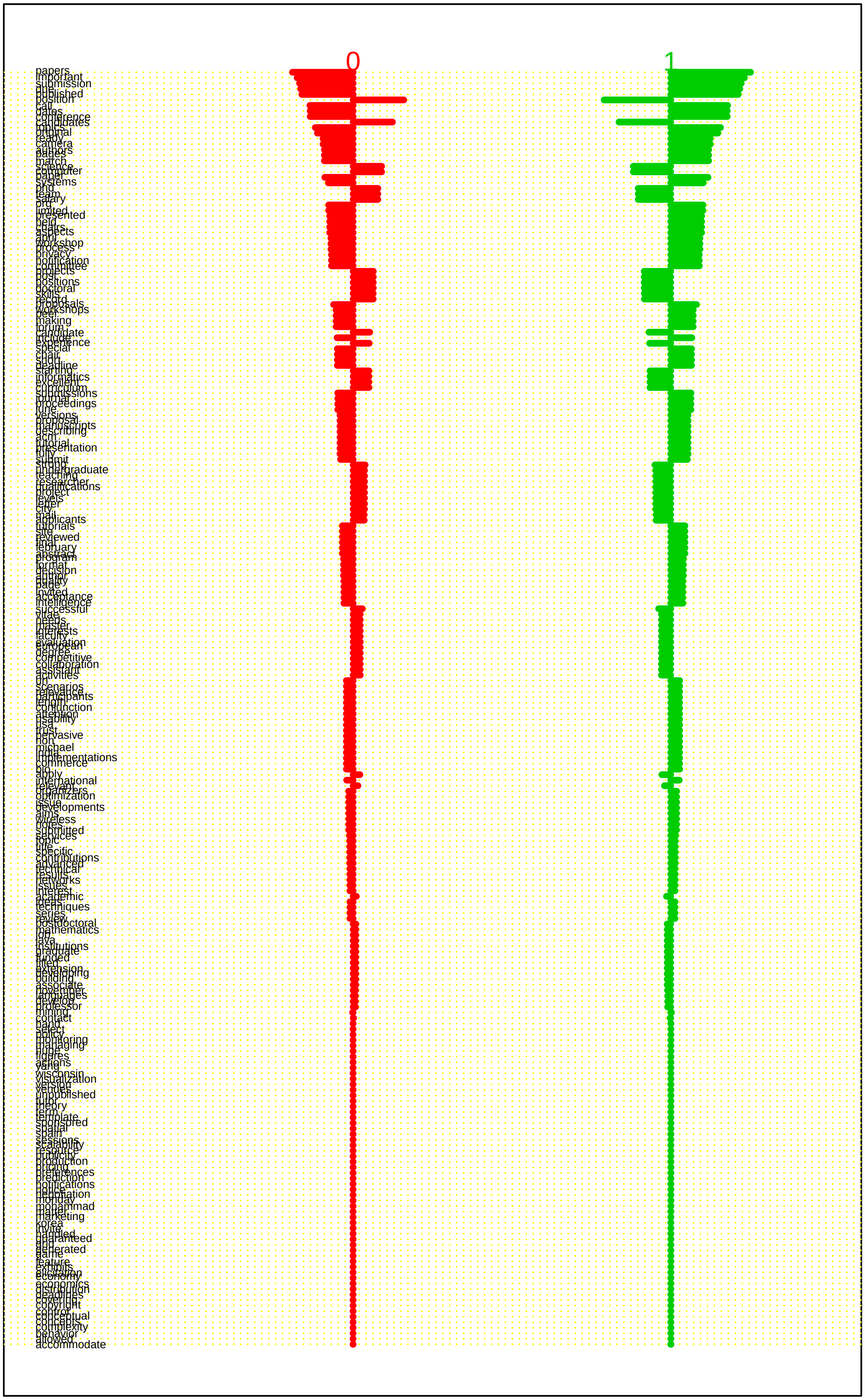
plots

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


Number of genes
3428 1979 622 231 98 62 36 20 12 9 6 6 4 1
Misclassification Error
0.0 0.2 0.4 0.6 0.8
0 1 2 3 4
Value of threshold


3428 1979 622 231 98 62 36 20 12 9 6 6 4 1
Misclassification Error
0.0 0.2 0.4 0.6 0.8
0
1
0 1 2 3 4
Value of threshold

```
pamr.plotcen(model, mydata, threshold = 1.3)
```


0
1

important features

```
## List the significant genes
NROW(important_gen)
```

```
## [1] 231
```

```
knitr::kable(colnames(data[,head(important_gen$id,10)]), caption = "Important feaures selected by Neares
```

Table 1: Important feaures selected by Nearest Shrunken Centroids

x
active
aachen
aicit
X2012call
advance
adami
ambient
anastasia
X10th
ambitious

confusion table

```
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
##
```

Analysis:

From the plot of thershold vs. misclassification error we can see that for the thershold value of 1.3, the class error is lowest.

231 features were selected by this model as the most important features. The top ten features of the model are given by the table above. Only few features make sense considering the dataset is of the year 2011, thus words like call for 2012 should be for the conference calls of 2012. Same is the case with the words like X10th, also refers to Xth conference, however rest of the features make little sense.

The test error is just 10% (accuracy is 90%) and the ability of our model to classify non-conference is 100%, while its ability to classifiy conference mail is 80%, the accuracy along with low number of samples hints that our model may very well be overfitted.

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?

```
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")
tmp_coeffs <- coef(cvfit, s = "lambda.min")
elastic_variable <- data.frame(name = tmp_coeffs@Dimnames[[1]][tmp_coeffs@i + 1], coefficient = tmp_coe
knitr::kable(elastic_variable, caption = "Contributing features in the elastic model")
```

Table 2: Contributing features in the elastic model

name	coefficient
(Intercept)	-1.0189313
abstracts	-0.3011264
aspects	0.0736776
bio	0.0228765
call	0.3319900
candidates	-0.1878311
computer	-0.2832065
conceptual	0.0380844
conference	0.1965330
dates	0.2416630
due	0.5211725
evaluation	-0.1796401
exhibits	0.3782700

name	coefficient
important	0.3924275
languages	-0.0258470
making	0.1892394
manuscripts	0.0325584
original	0.0558205
papers	0.3853810
peer	0.0967211
position	-0.3750830
process	0.0016238
projects	-0.1904080
proposals	0.0553554
published	0.2818206
queries	-0.3002459
record	-0.1162514
relevant	-0.1135564
scenarios	0.0053470
spatial	0.1925007
submission	0.2803519
team	-0.1291278
versions	0.1545749

```
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()

features_count <- cbind(NROW(important_gen), NROW(elastic_variable), length(svm_fit@coef[[1]]))

final_result <- rbind(final_result, features_count)

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

rownames(final_result) <- c("Accuracy", "Number of Features")

knitr::kable(final_result, caption = "Comparsion of Models on Test dataset")
```

Table 3: Comparsion of Models on Test dataset

	Nearest Shrunken Centroid Model	ElasticNet Model	SVM Model
Accuracy	90	90	95
Number of Features	231	33	43

Analysis:

33 variables were selected by the elastic net model as the features for classifying the mails as conference, while the svm model selects 43 features to classify the mails.

From the model comparsion we see that overall choosing SVM gives us the best accuracy, while Nearest Centroid Model and Elastic Net model both have the same accuracy, however this is not a strong point given the low number of samples. From the coefficents of the elastic net we can see that the features choosen from the elastic net are far more reasonable than the once choosen by Nearest Centroid model, thus Elastic Net features selection is superior to Nearest Centroid model in quality and quantity too.

We also know for a fact the SVM has regularization built into we see that SVM has the benefits of the elastic net too(partial).Thus my choice is the SVM Model which has the least features and highest accuracy.

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.

```
p_value <- c()
for (i in 1:4702){
  x <- data[,i]
  res <- t.test(x ~ Conference, data = data, alternative = "two.sided")
  p <- res$p.value
  p_value[i] <- p
}
p_value <- as.data.frame(p_value)
p_value$reject_flag <- as.factor(ifelse(p_value$p_value <0.05, "Retain", "Drop"))
p_value$column_index <- row.names(p_value)

keep <- ifelse(p_value$reject_flag == "Retain", as.numeric(p_value$column_index), NA)
keep <- na.omit(keep)
colnames(data[,keep])
```

```
##   [1] "abstract"        "academic"        "acceptance"      "accepted"        "access"          "acm
##  [28] "bio"             "call"            "calls"           "camera"          "canada"          "can
##  [55] "contributions"   "copyright"       "covering"        "cross"           "curriculum"      "dat
##  [82] "expected"        "experience"      "extension"       "feature"         "february"        "fig
## [109] "include"         "included"        "india"           "infrastructures" "initially"       "ins
## [136] "letter"          "levels"          "limited"         "liu"             "looking"         "mad
## [163] "ontologies"      "opportunity"     "optimization"    "org"             "organizers"      "org
## [190] "privacy"         "proceedings"     "process"         "professor"       "proficiency"     "pro
## [217] "scalability"     "scenarios"       "science"         "scope"           "security"        "ser
## [244] "taiwan"          "takes"           "tasks"           "teaching"        "team"            "tec
## [271] "versions"        "vienna"          "visualization"   "vitae"           "wang"            "wir
```

Analysis:

From the above table we can see that 281 features had significant p-values (more than 0.05), some of the features do make sense to in their ability to distinguish mails pertaining to conferences, such as 'committee', 'conference', 'international', 'keynote', 'manuscripts' etc.

Thus we see that even a simple and time tested techniques like t test can be used to get a sense of the important features for model building. Although this method does help us its still selects far too many features than the other methods that we have seen and implemented uptill now.

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, width=80)

# 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) +
  geom_smooth(method = "loess") +
    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
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)

plot(gam_model)

# s=interp(temp$Year,temp$Week, fitted(gam_model))
# persp3d(s$x, s$y, s$z, col="red")
model_deviance <- NULL
for(sp in c(0.001, 0.01, 0.1, 1, 10))
{
  k=length(unique(flu_data$Week))

gam_model <- mgcv::gam(data = flu_data, Mortality~Year+s(Week, k=k, sp=sp), method = "GCV.Cp")
temp <- cbind(gam_model$deviance, gam_model$fitted.values, gam_model$y, flu_data$Time_fixed,
              sp, 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"))

# plot of predicted vs. observed mortality
p8 <- ggplot(data=model_deviance_wide[model_deviance_wide$penalty_factor == 0.001,],
             aes(x= Time, y = value)) +
  geom_point(aes(color = variable), size=0.7) +
  geom_line(aes(color = variable), size=0.7) +
  scale_color_manual(values=c("#E69F00", "#009E73")) +
  theme_light() +
  ggtitle("Plot of Mortality vs. Time(Penalty 0.001)")

p9 <- ggplot(data=model_deviance_wide[model_deviance_wide$penalty_factor == 10,],
             aes(x= Time, y = value)) +
  geom_point(aes(color = variable), size=0.7) +
    geom_line(aes(color = variable), size=0.7) +
  scale_color_manual(values=c("#E69F00", "#009E73")) +
    theme_light() +
  ggtitle("Plot of Mortality vs. Time(Penalty 10)")

p8
p9

k=length(unique(flu_data$Week))
gam_model <- mgcv::gam(data = flu_data, Mortality~Year+s(Week, k=k), 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")

k1 = length(unique(flu_data$Year))
k2 = length(unique(flu_data$Week))
k3 = length(unique(flu_data$Influenza))

gam_model_additive <- gam(Mortality ~ s(Year, k=k1) +
                                      s(Week, k=k2) +
```

```
                                    s(Influenza, k=k3),
                            data = flu_data)

summary(gam_model_additive)

flu_data$fitted.values = gam_model_additive$fitted.values


p11 <- ggplot(data = flu_data, aes(x = Time_fixed)) +
  geom_line(aes( y = Mortality, color = "Mortality")) +
  geom_line(aes(y = fitted.values, color = "fitted.values")) +
      theme_light() +
  scale_color_manual(values=c(Mortality = "#009E73", fitted.values = "#E69F00")) +
  labs(y = "Mortality / fitted.values") +
  ggtitle("Plot of Mortality and Fitted vs. Time")

p11

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]]

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 = 1.3))
predicted_scc_test <- pamr.predict(model, newx = x_test, threshold = 1.3)
pamr.plotcv(cvmodel)
pamr.plotcen(model, mydata, threshold = 1.3)
## List the significant genes
NROW(important_gen)
knitr::kable(colnames(data[,head(important_gen$id,10)]), caption = "Important feaures selected by Neares

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)
```

```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")
tmp_coeffs <- coef(cvfit, s = "lambda.min")
elastic_variable <- data.frame(name = tmp_coeffs@Dimnames[[1]][tmp_coeffs@i + 1], coefficient = tmp_coe
knitr::kable(elastic_variable, caption = "Contributing features in the elastic model")

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()

features_count <- cbind(NROW(important_gen), NROW(elastic_variable), length(svm_fit@coef[[1]]))

final_result <- rbind(final_result, features_count)

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

rownames(final_result) <- c("Accuracy", "Number of Features")

knitr::kable(final_result, caption = "Comparsion of Models on Test dataset")

p_value <- c()
for (i in 1:4702){
  x <- data[,i]
  res <- t.test(x ~ Conference, data = data, alternative = "two.sided")
  p <- res$p.value
  p_value[i] <- p
}
p_value <- as.data.frame(p_value)
p_value$reject_flag <- as.factor(ifelse(p_value$p_value <0.05, "Retain", "Drop"))
p_value$column_index <- row.names(p_value)
```

```r
keep <- ifelse(p_value$reject_flag == "Retain", as.numeric(p_value$column_index), NA)
keep <- na.omit(keep)
colnames(data[,keep])
```