lab2-block2_group5_report

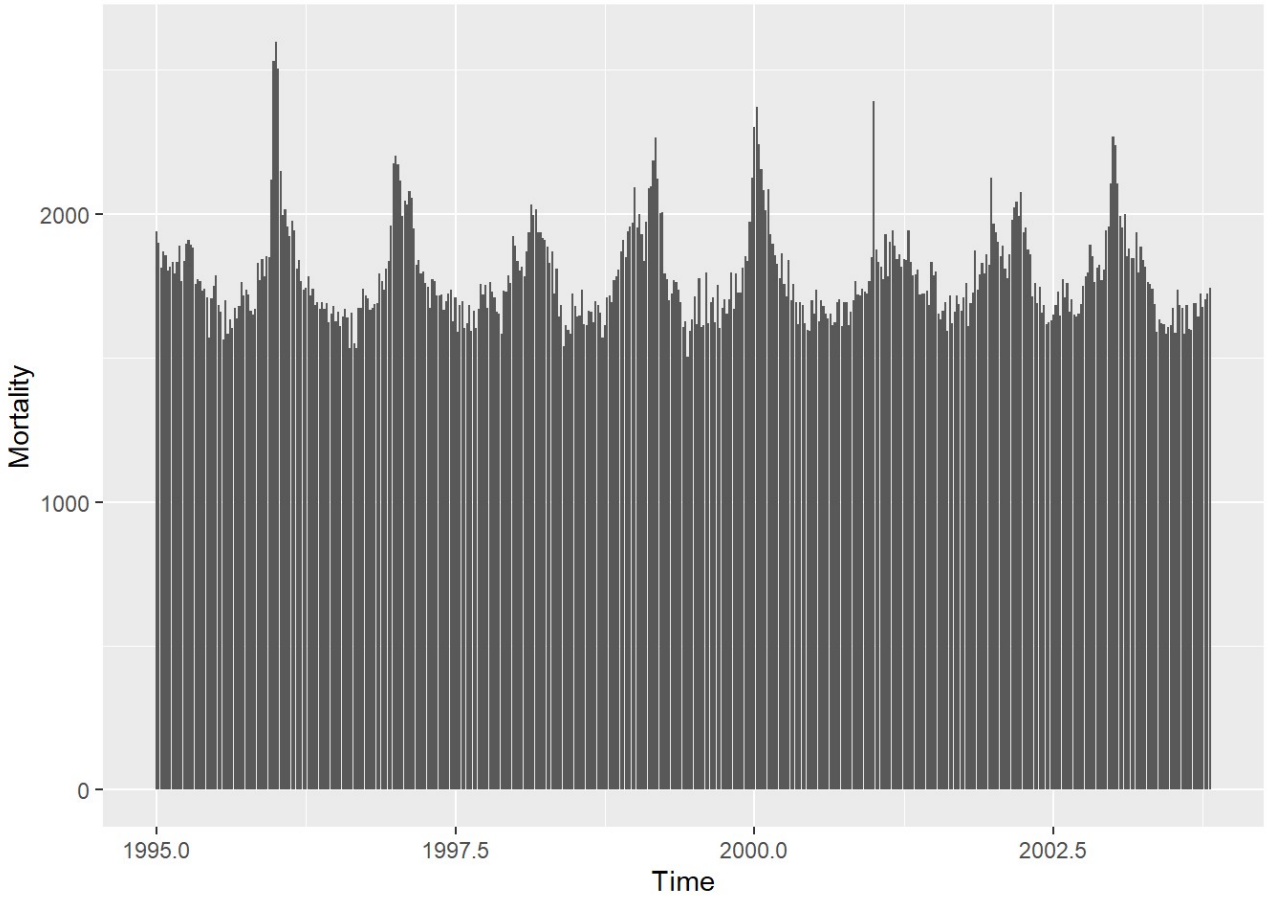
Bjorn_Hansen, Erik Anders, Ahmed Alhasan

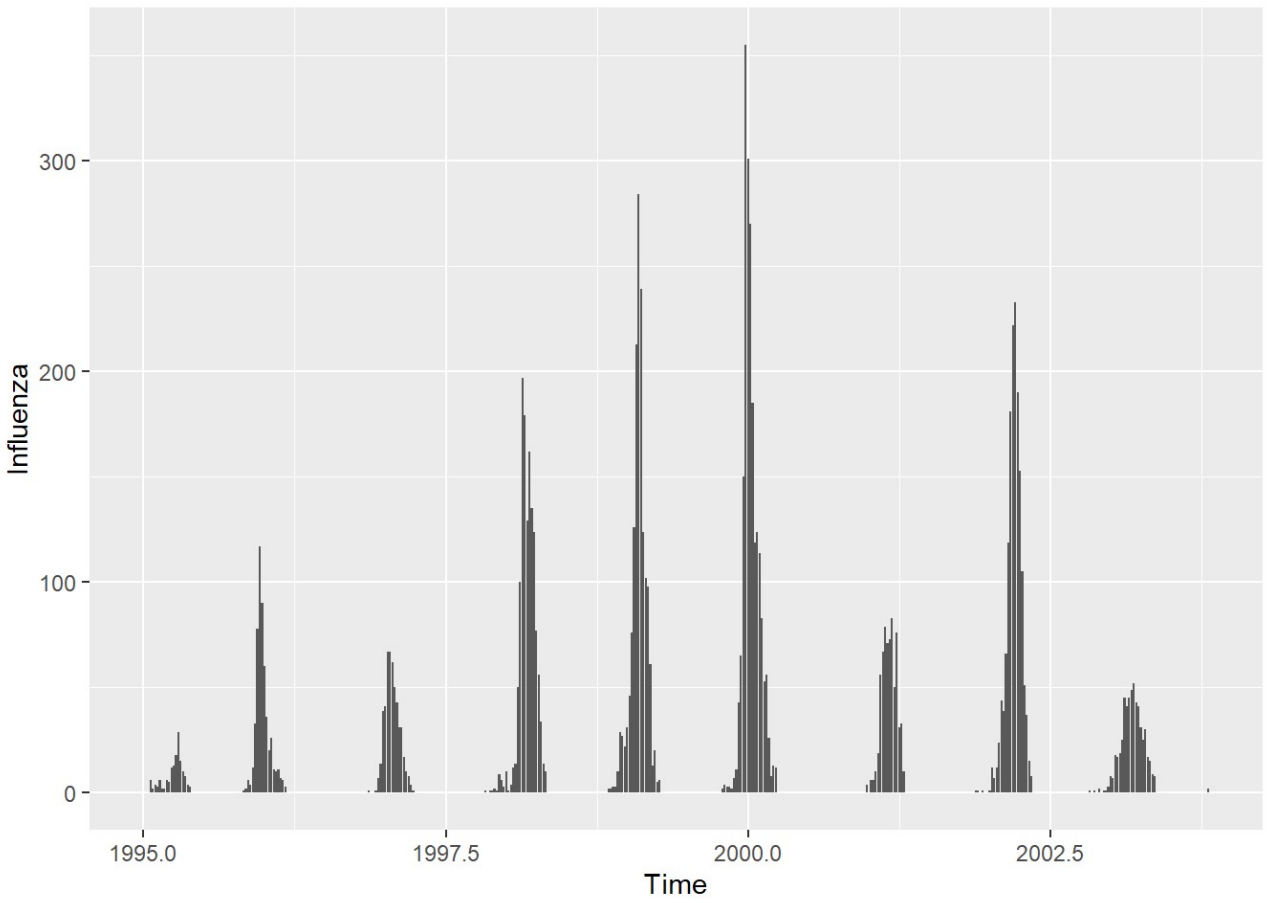
12/16/2019

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

1

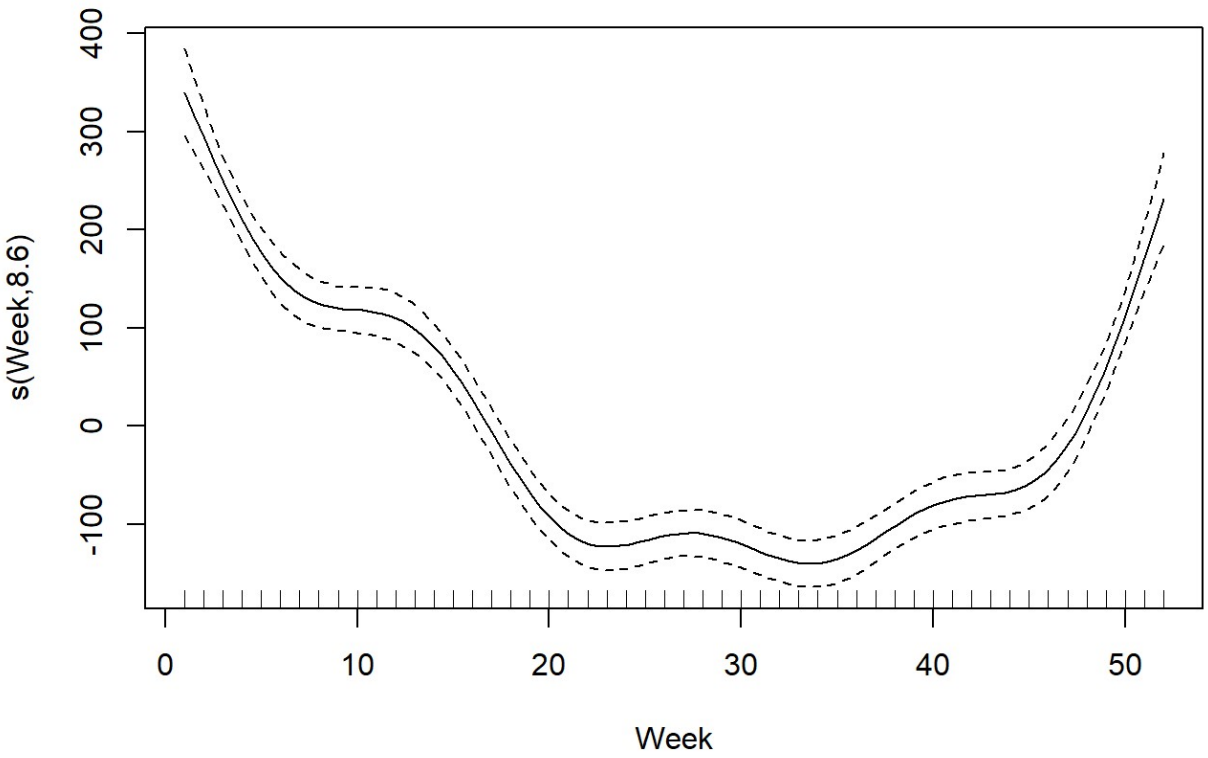
As can be seen from the plots below there does not seem to be a clear relationship between influenza and mortality by viewing the plots. There are perhaps peaks of influenza when there are relative peaks of mortality during the year of 2000.

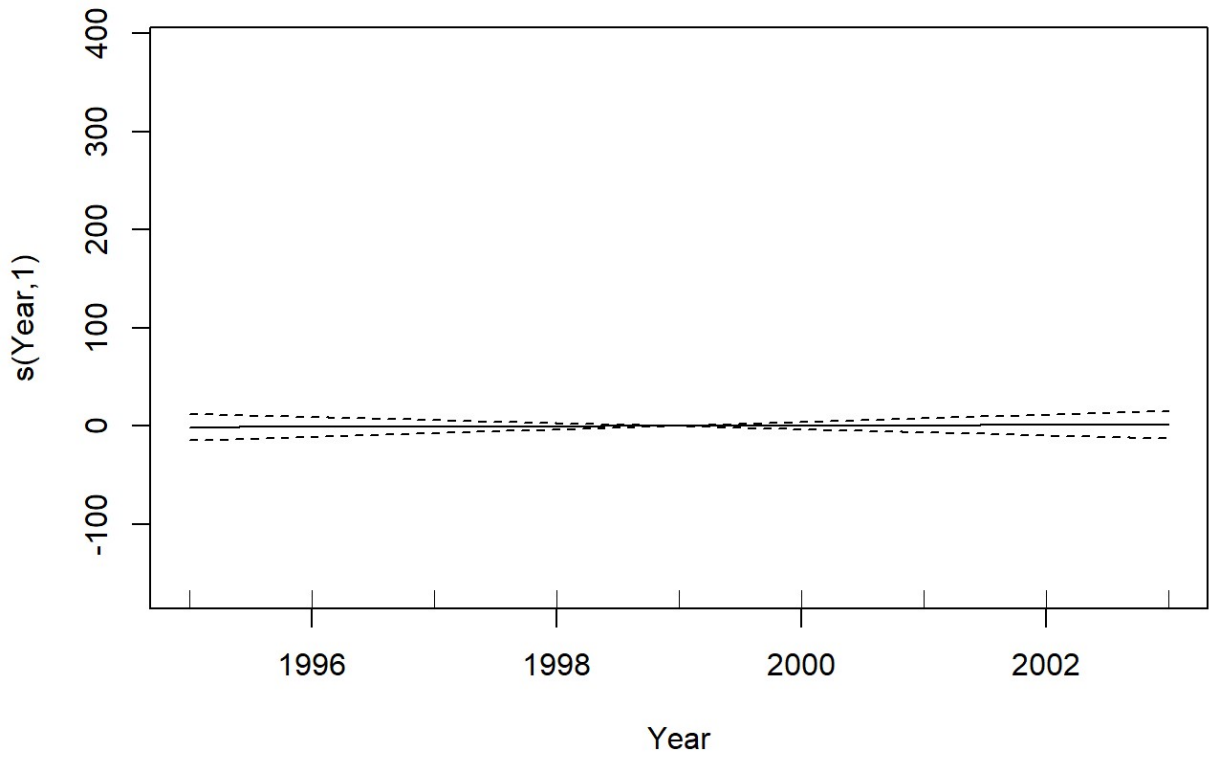



Influenza
300
200
100
0
1995.0
1997.5
2000.0
2002.5
Time

2

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Mortality ~ Year + s(Week, k = length(unique(data$Week)))
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -680.598   3367.760  -0.202    0.840
## Year           1.233      1.685   0.732    0.465
##
## Approximate significance of smooth terms:
##           edf Ref.df     F p-value
## s(Week) 14.32  17.87 53.86  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Rank: 52/53
## R-sq.(adj) =  0.677   Deviance explained = 68.8%
## GCV = 8708.6  Scale est. = 8398.9    n = 459
```

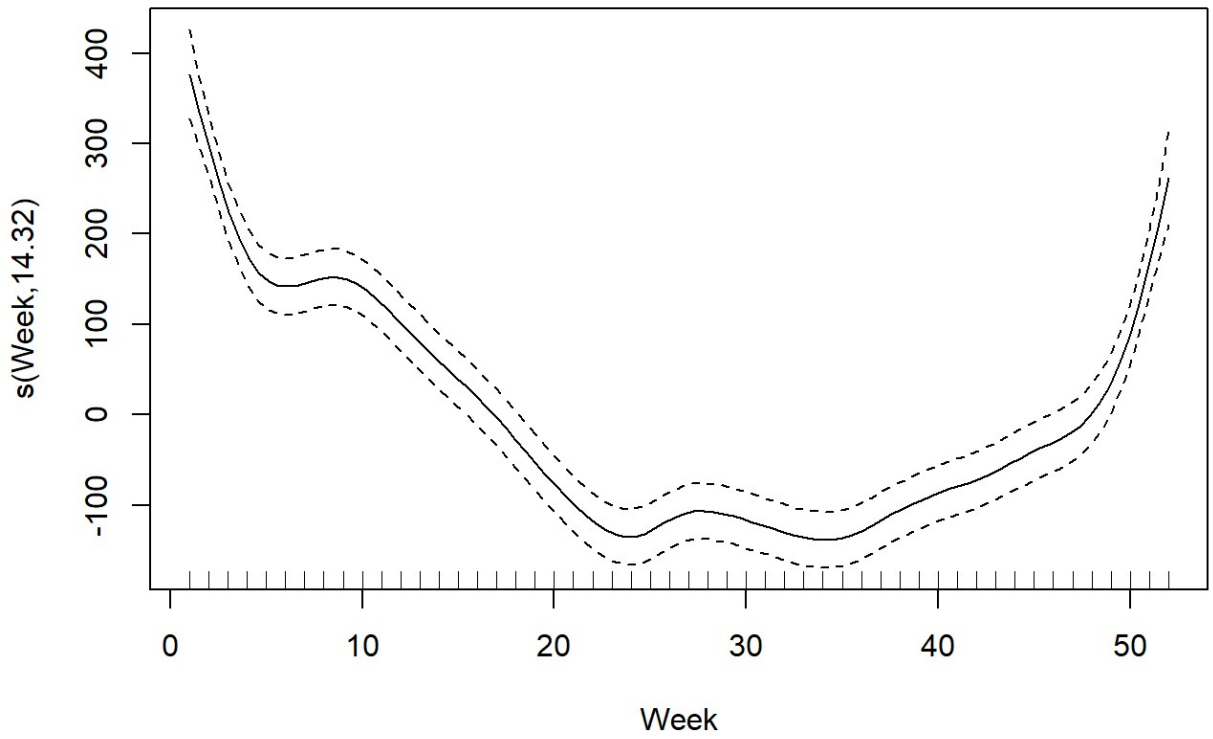

s(Week,8.6)
400
300
200
100
0
-100
0
10
20
30
40
50
Week

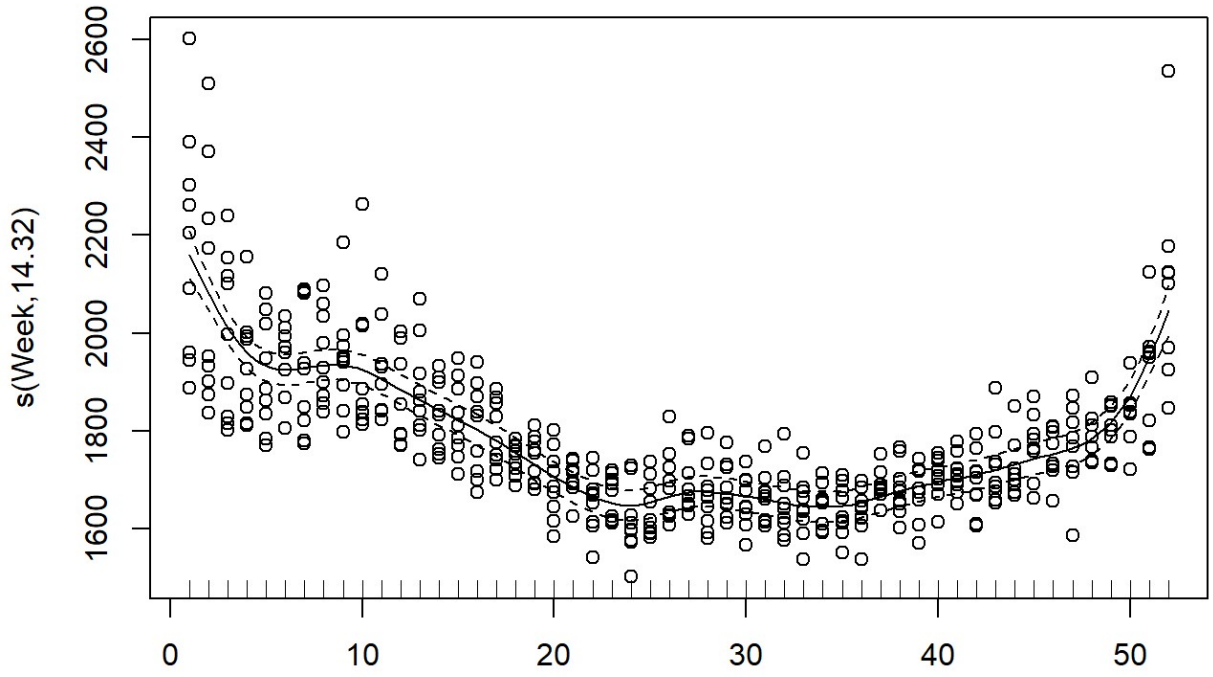


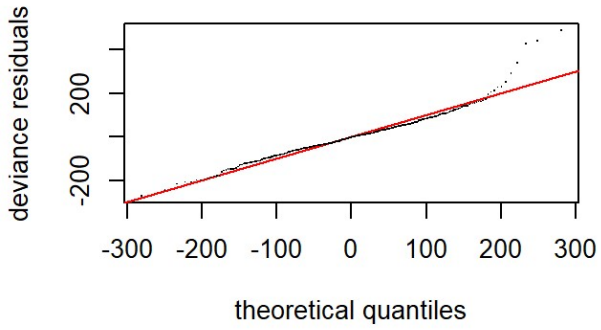
3

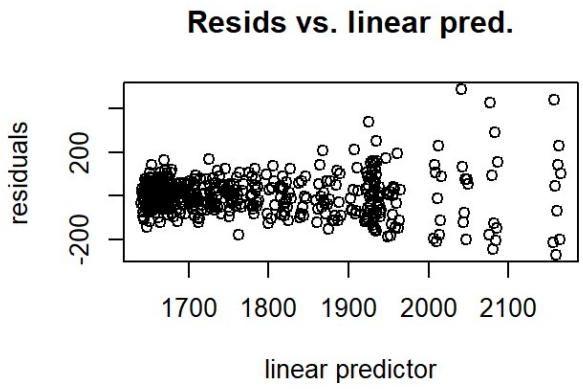
As is seen in the plots below, the histogram of the residuals seem to be normaly distributed and the model seems fit the data well. The week of the year is the most significant feature. The year is not of importance as mortality does not change much between years. It can also be seen from the previous summary of the model that the year has a large p-value.

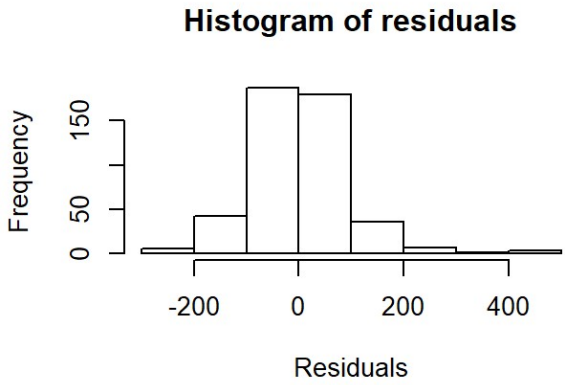
```
##      s(Week)
## 0.0001131932
```

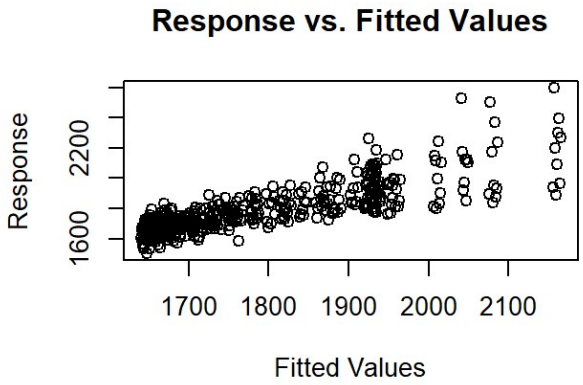

s(Week,14.32)
400
300
200
100
0
-100
0
10
20
30
40
50
Week


s(Week,14.32)
1600
1800
2000
2200
2400
2600
0
10
20
30
40
50

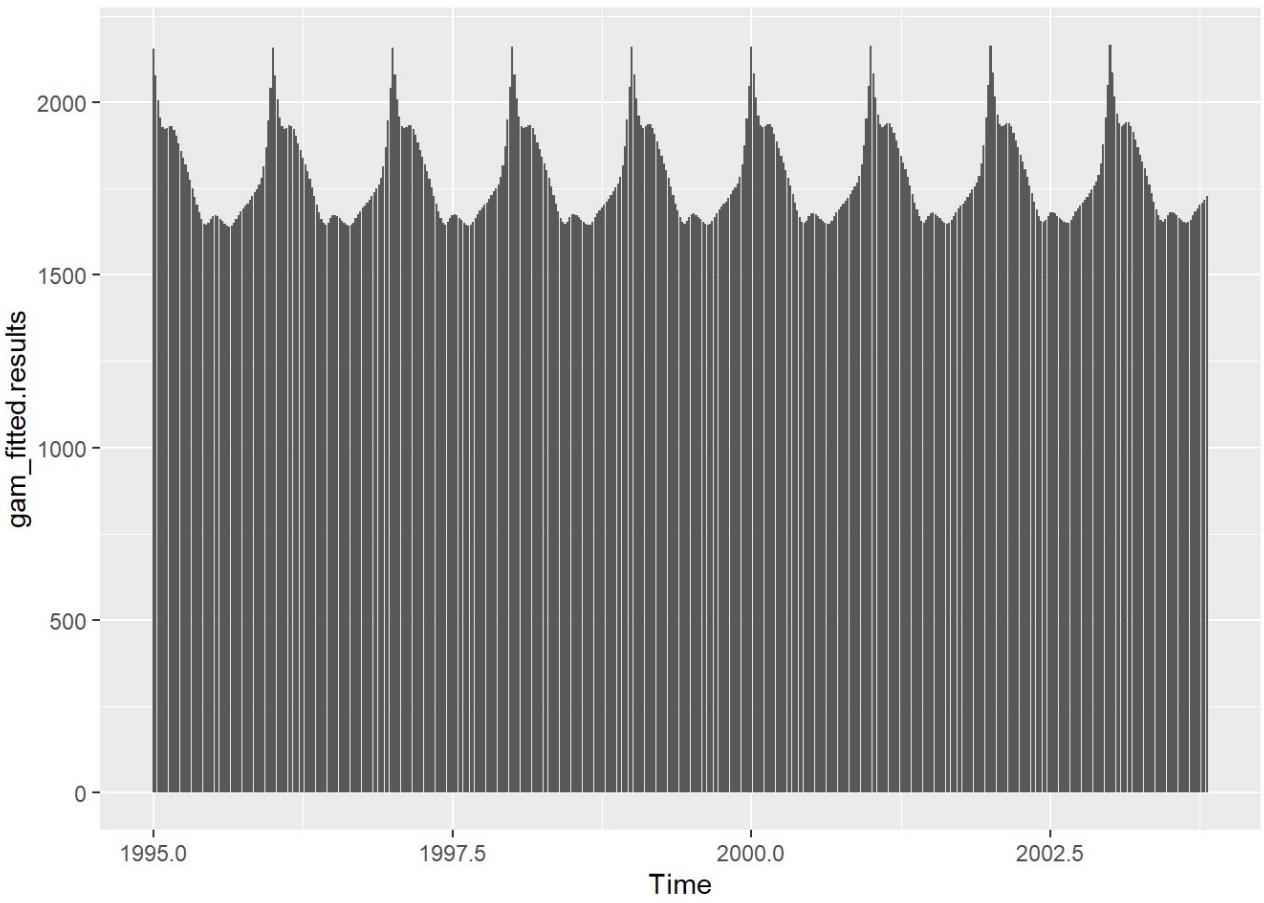






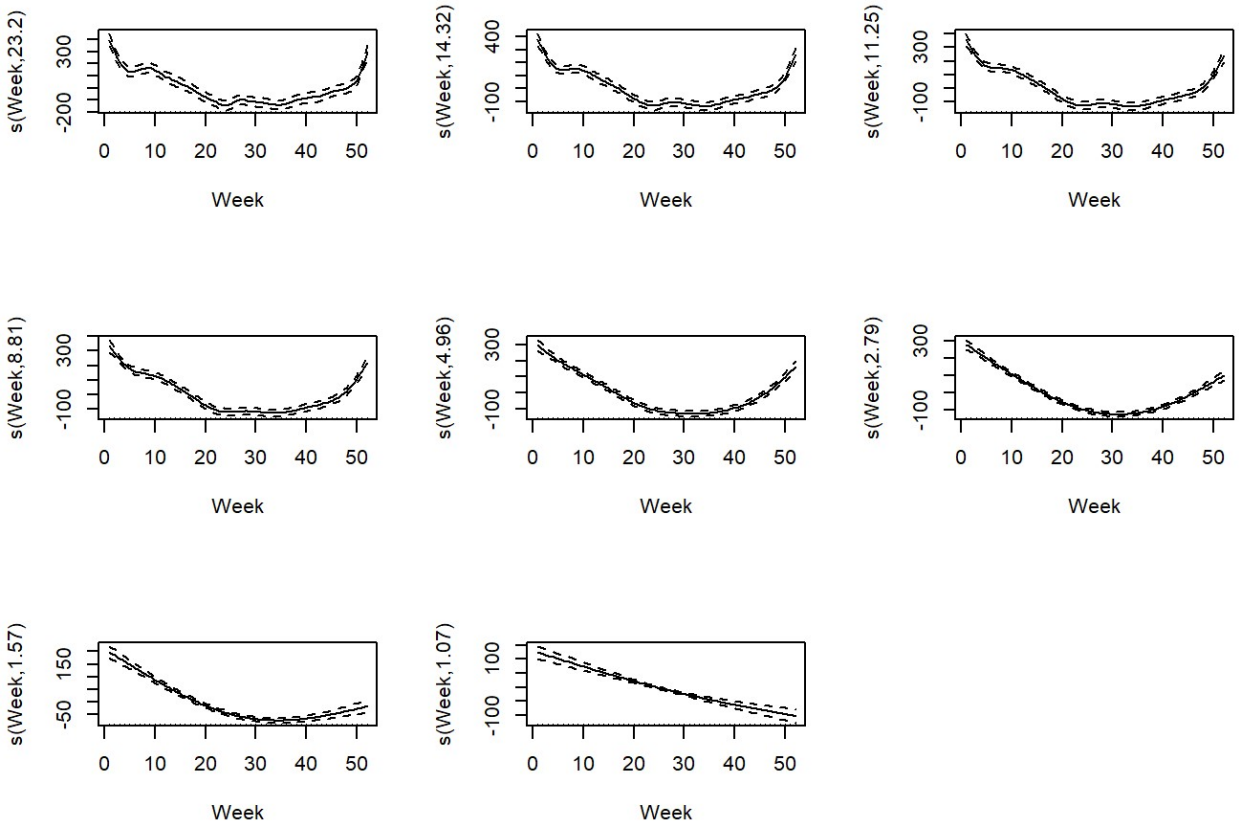



```
##
## Method: GCV   Optimizer: magic
## Smoothing parameter selection converged after 9 iterations by steepest
## descent step failure.
## The RMS GCV score gradient at convergence was 0.00106719 .
## The Hessian was positive definite.
## Model rank =  52 / 53
##
## Basis dimension (k) checking results. 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) 51.0 14.3    1.09    0.98
```


2000
1500
1000
500
0
gam_fitted.results
1995.0
1997.5
2000.0
2002.5
Time

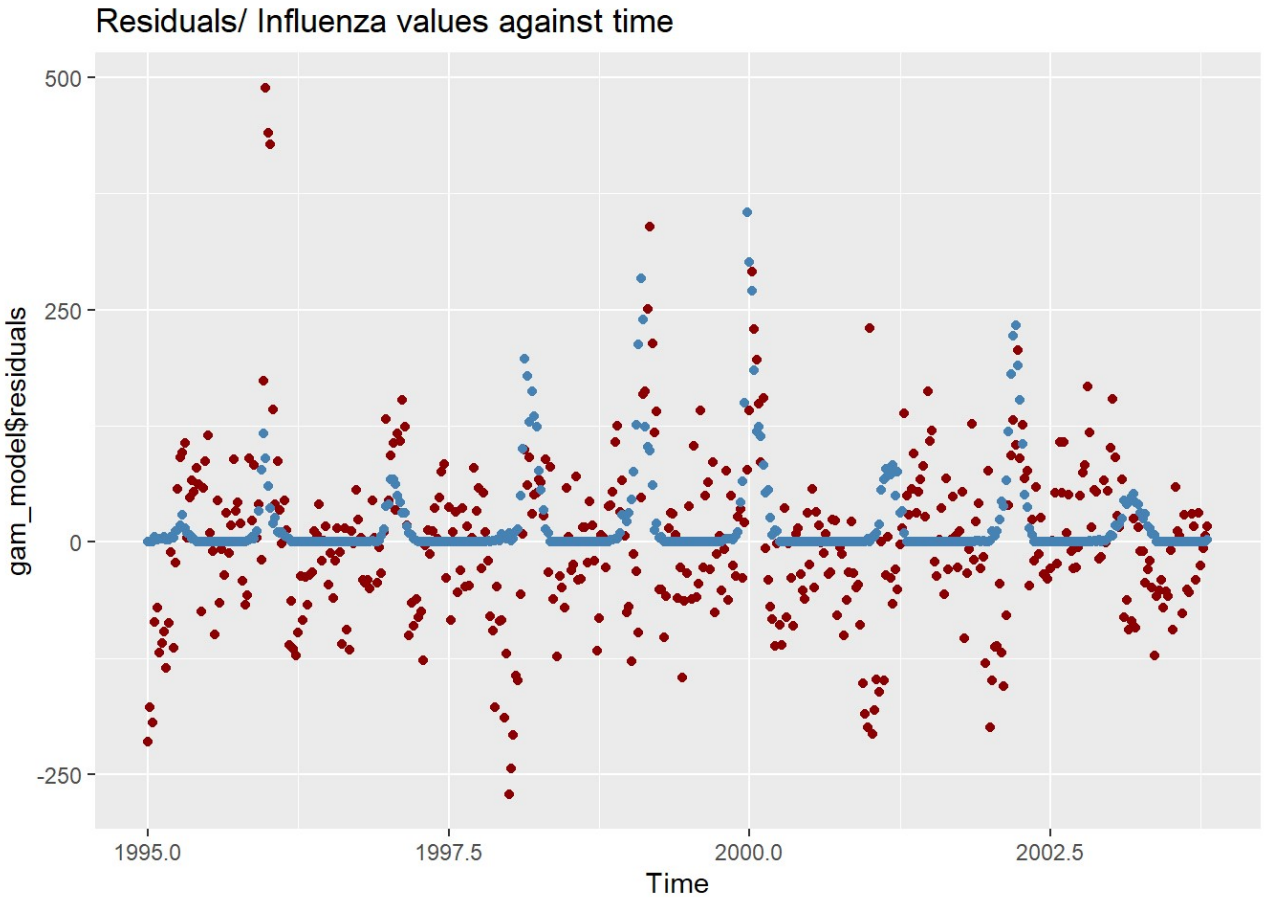
4

As can be seen from the output below, the lower penatly factor makes the model fit the data more loosly whereas a larger penalty factor gives the model a exact fit. The fit of the model does however not change much after a penalty factor of 13 is introduced.



5

Viewing the plot bleow it would seem viable to say that the temporal pattern in the residuals correlate to the outbreaks of influenza.

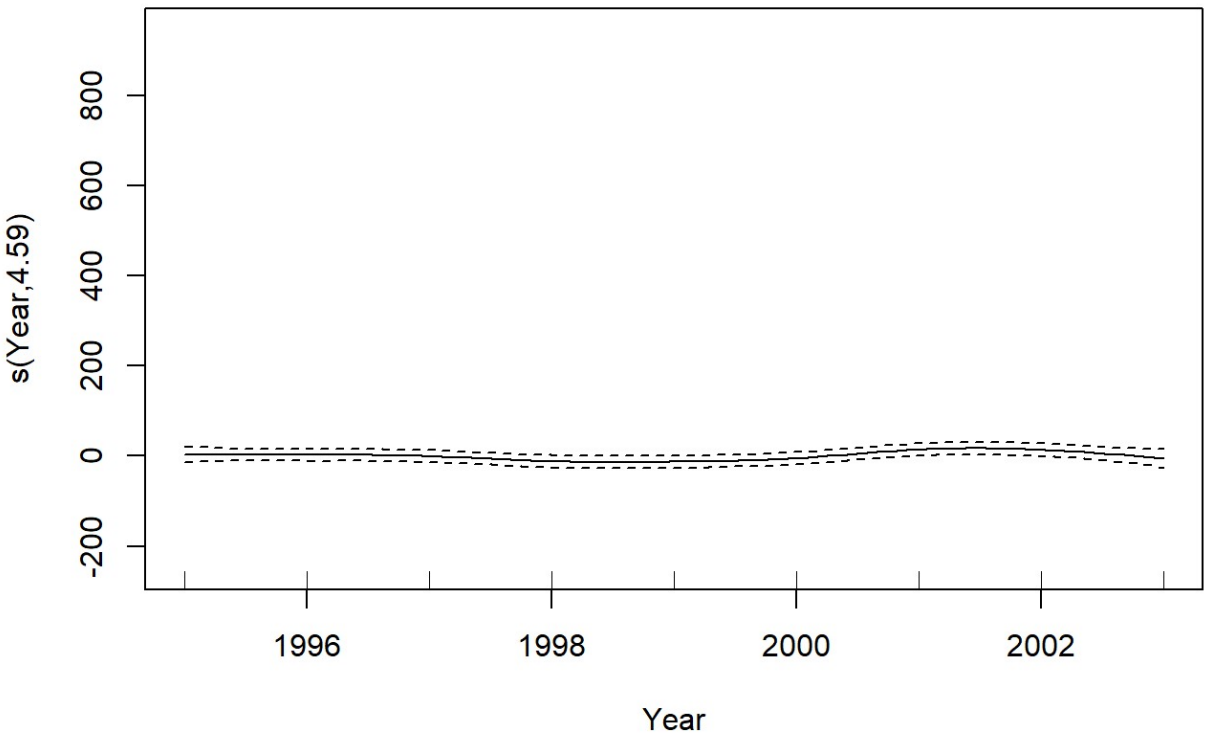


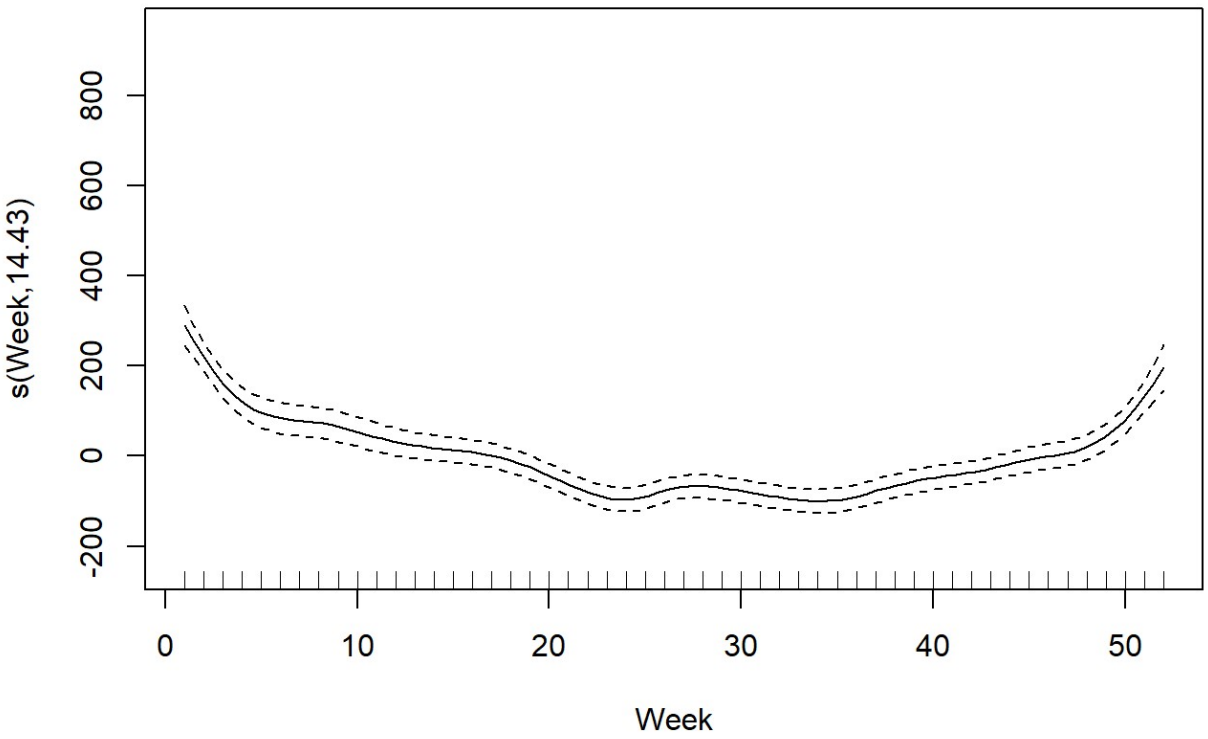
6

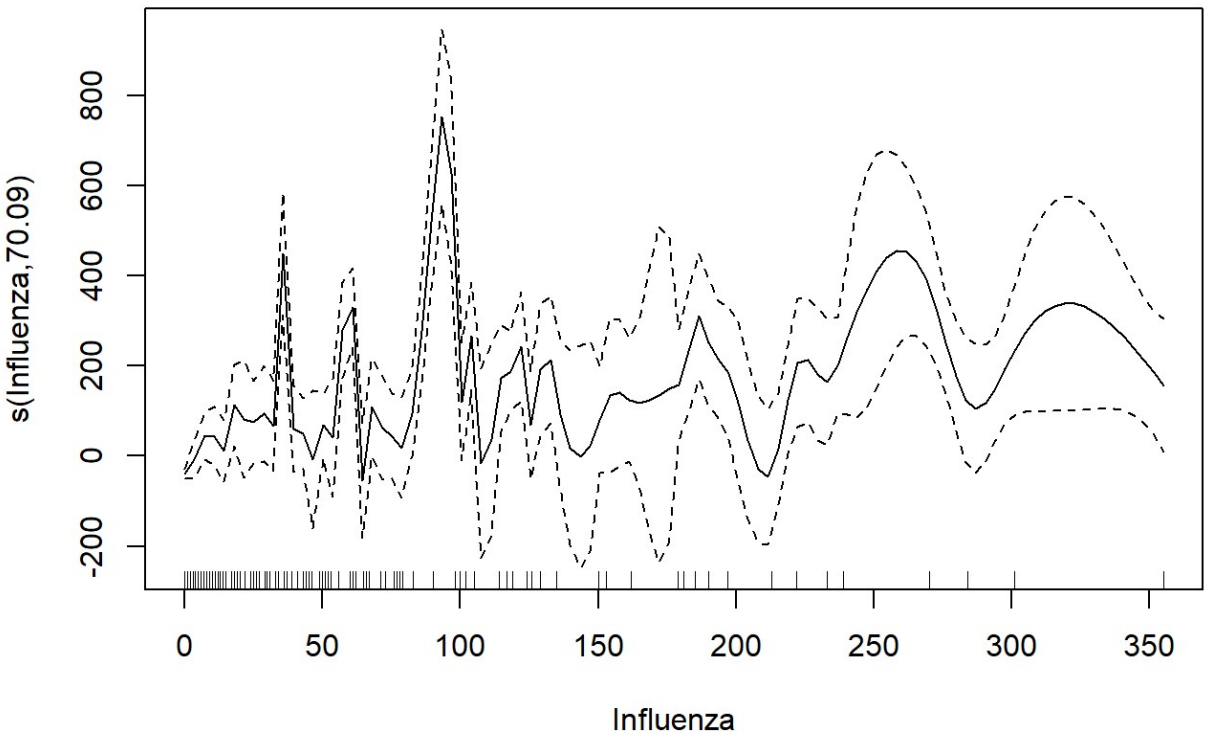
Plot shows that the fitted values are good aproximations of the mortality values. Using the summary function it is seen from the p- values that influenza and week are significant contributers of the mortality rate. The year is however not a significant contributer.

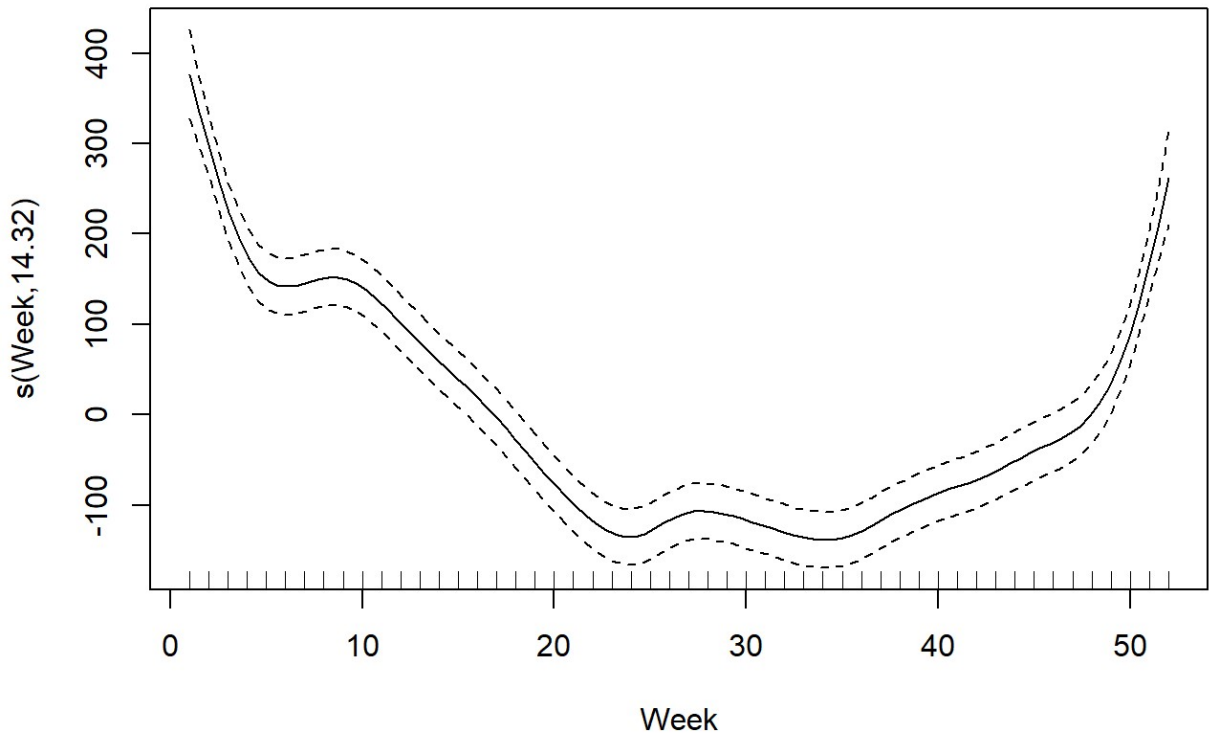
The first plot below is from the original (previous) model and the second from the spline. The spline model is a better model than the previous GAM model. The last output plot shows that the fitted values match the data very well.

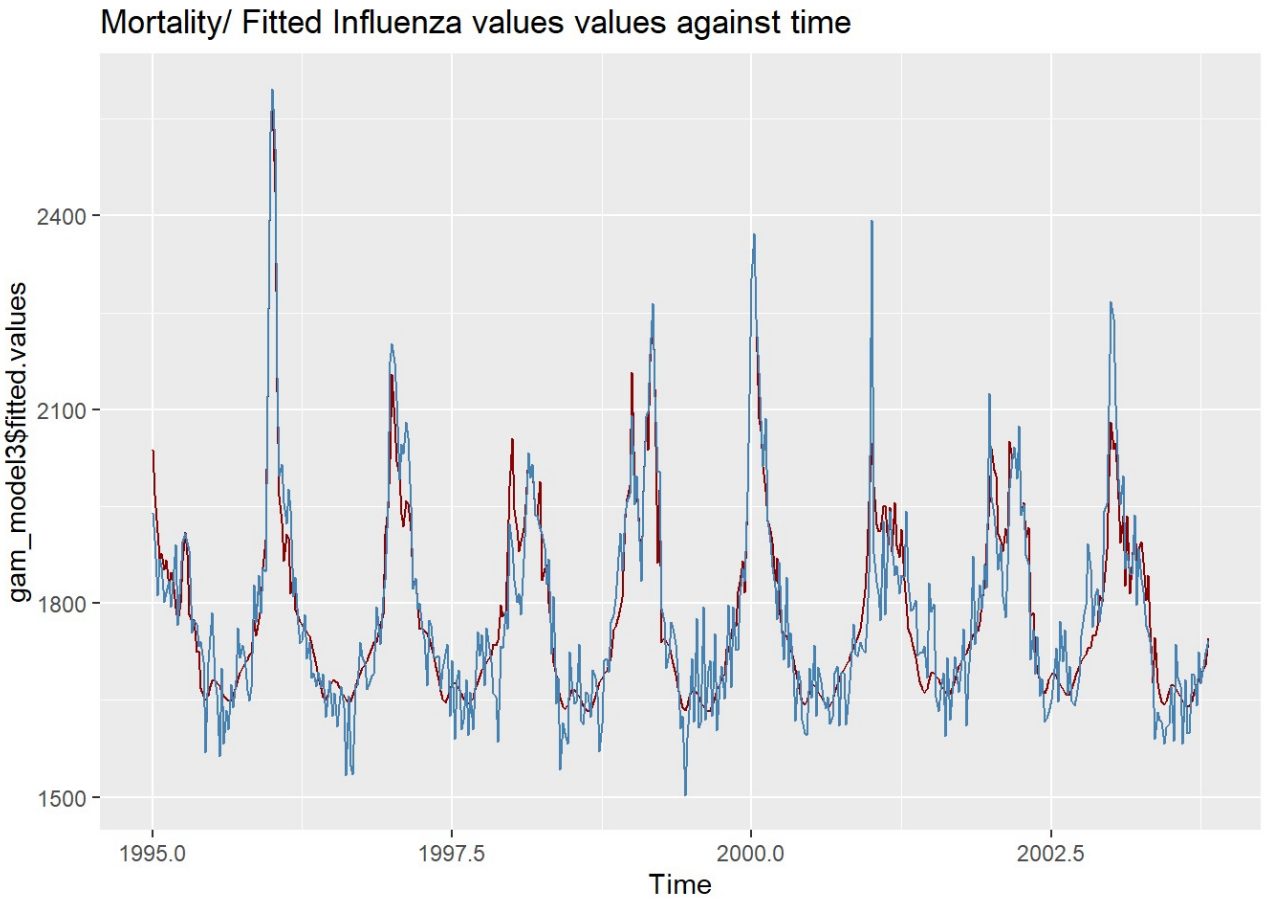
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Mortality ~ s(Year, k = length(unique(data$Year))) + s(Week,
##     k = length(unique(data$Week))) + s(Influenza, k = length(unique(data$Influenz
a)))
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1783.765      3.198   557.8   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                 edf Ref.df      F p-value
## s(Year)       4.587  5.592  1.500   0.178
## s(Week)      14.431 17.990 18.763  <2e-16 ***
## s(Influenza) 70.094 72.998  5.622  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Rank: 134/144
## R-sq.(adj) =  0.819   Deviance explained = 85.4%
## GCV = 5840.5  Scale est. = 4693.7    n = 459
```


s(Year,4.59)
800
600
400
200
0
-200
1996
1998
2000
2002
Year


s(Week,14.43)
800
600
400
200
0
-200
0
10
20
30
40
50
Week


s(Influenza,70.09)
-200
0
200
400
600
800
0
50
100
150
200
250
300
350
Influenza


s(Week,14.32)
400
300
200
100
0
-100
0
10
20
30
40
50
Week



Assignment 2. High-dimensional methods

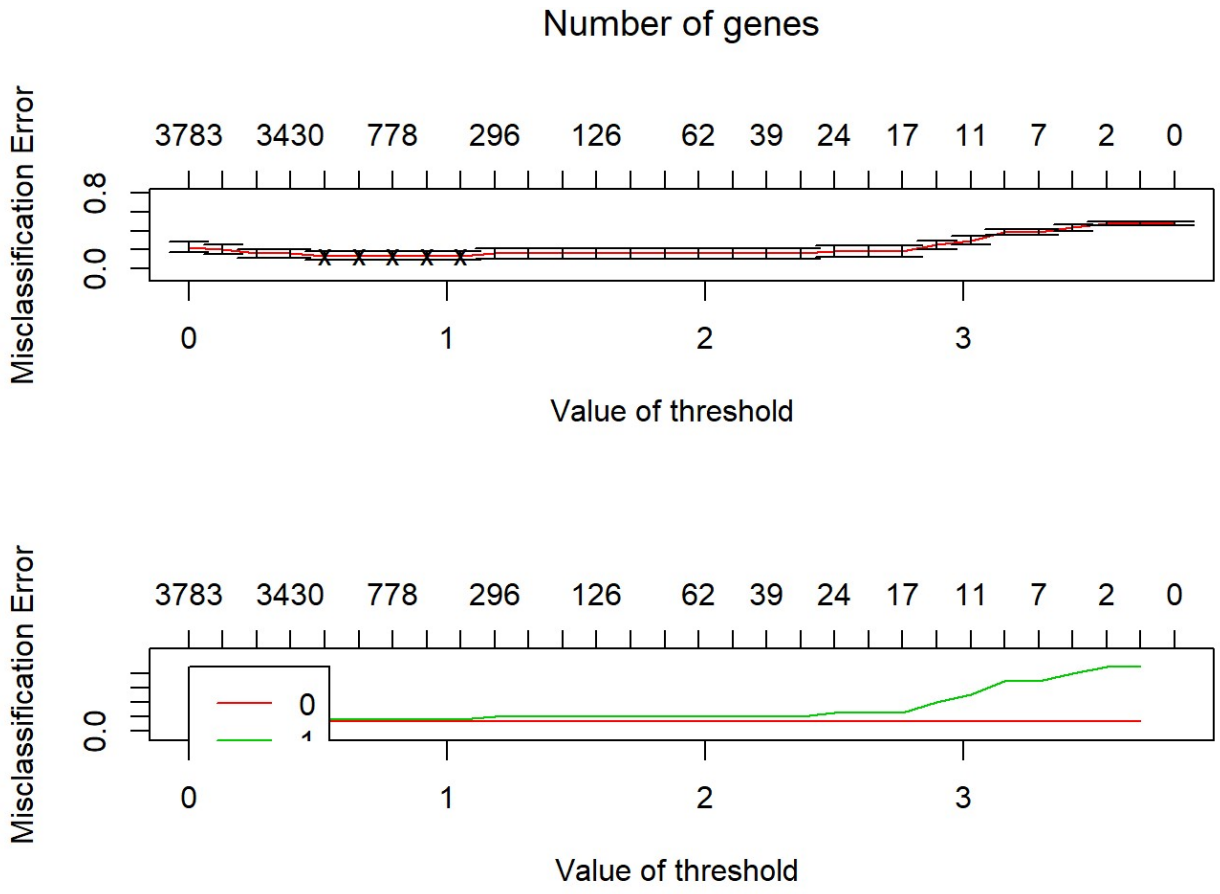
1.

The nearest squished centroids model using cross validation had at lowest 6 errors, the largest threshold with 6 errors was 2.757. This threshold was then used in the output plots below. Using the cntroid plot, the 12 most contributing features for making predictions both negative (0) and positive (1).

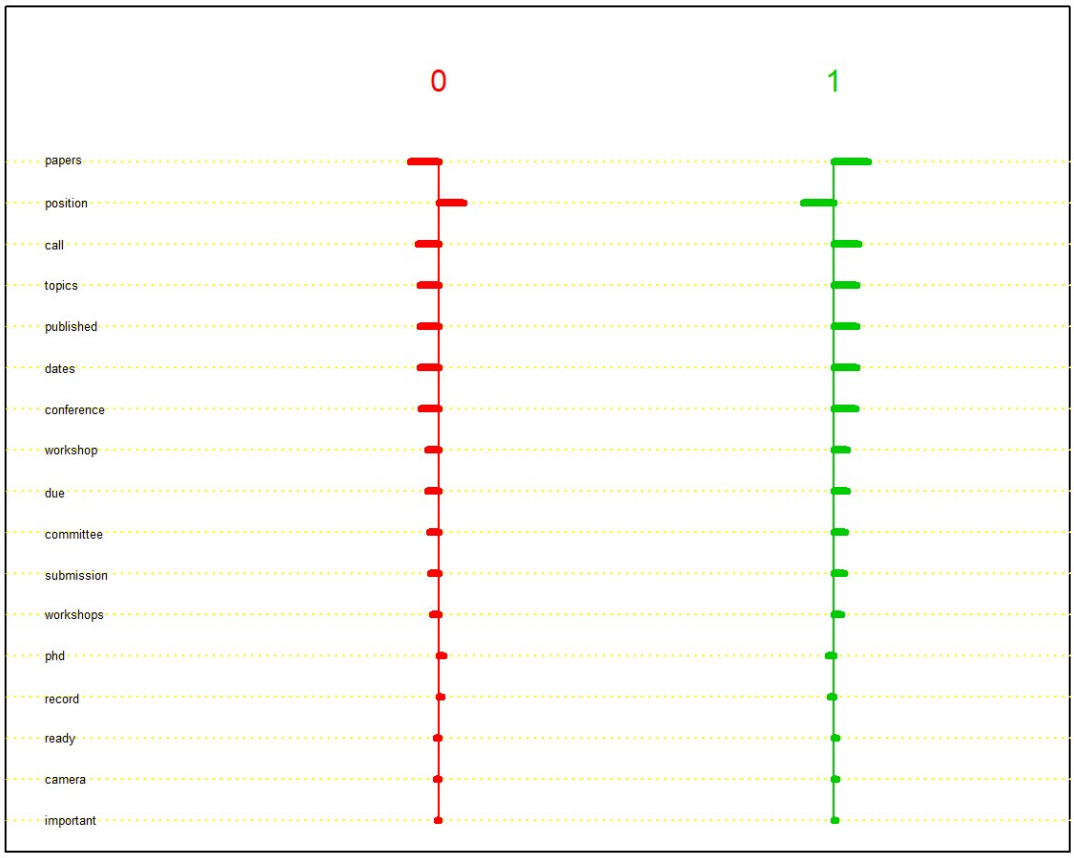
```
## 1234567891011121314151617181920212223242526272829 30
```

```
## 12Fold 1 :1234567891011121314151617181920212223242526272829 30
## Fold 2 :1234567891011121314151617181920212223242526272829 30
## Fold 3 :1234567891011121314151617181920212223242526272829 30
## Fold 4 :1234567891011121314151617181920212223242526272829 30
## Fold 5 :1234567891011121314151617181920212223242526272829 30
## Fold 6 :1234567891011121314151617181920212223242526272829 30
## Fold 7 :1234567891011121314151617181920212223242526272829 30
## Fold 8 :1234567891011121314151617181920212223242526272829 30
## Fold 9 :1234567891011121314151617181920212223242526272829 30
## Fold 10 :1234567891011121314151617181920212223242526272829 30
```

```
## Call:
## pamr.cv(fit = pamr_m, data = data2)
##    threshold nonzero errors
## 1  0.000     3783    10
## 2  0.132     3548    9
## 3  0.263     3499    7
## 4  0.395     3430    7
## 5  0.527     2368    6
## 6  0.658      922    6
## 7  0.790      778    6
## 8  0.922      717    6
## 9  1.053      339    6
## 10 1.185      296    7
## 11 1.317      188    7
## 12 1.448      158    7
## 13 1.580      126    7
## 14 1.712      111    7
## 15 1.843       87    7
## 16 1.975       62    7
## 17 2.107       56    7
## 18 2.238       39    7
## 19 2.370       37    7
## 20 2.502       24    8
## 21 2.633       19    8
## 22 2.765       17    8
## 23 2.897       13    11
## 24 3.028       11    13
## 25 3.160        9    17
## 26 3.292        7    17
## 27 3.423        7    19
## 28 3.555        2    21
## 29 3.687        1    21
## 30 3.818        0    21
```


Number of genes
Misclassification Error
3783 3430 778 296 126 62 39 24 17 11 7 2 0
0.0 0.8
0 1 2 3
Value of threshold

Misclassification Error
3783 3430 778 296 126 62 39 24 17 11 7 2 0
0.0
0
0 1 2 3
Value of threshold




```
##    Y hat
## Y    0  1
##   0 11  0
##   1  0  9
```

```
## [1] "NSC Error Rate with Test Data: 0"
```

```
##       id   0-score 1-score
##  [1,] 3036 -0.146  0.1752
##  [2,] 3187 0.1269  -0.1522
##  [3,] 596  -0.1074 0.1289
##  [4,] 4282 -0.0975 0.117
##  [5,] 1045 -0.0968 0.1161
##  [6,] 3364 -0.0968 0.1161
##  [7,] 869  -0.0926 0.1111
##  [8,] 4628 -0.0598 0.0717
##  [9,] 1262 -0.0598 0.0717
## [10,] 810  -0.0488 0.0585
## [11,] 4060 -0.0455 0.0546
## [12,] 4629 -0.0343 0.0412
## [13,] 3125 0.0221  -0.0265
## [14,] 3458 0.0154  -0.0184
## [15,] 599  -0.0154 0.0184
## [16,] 3433 -0.0154 0.0184
## [17,] 2049 -0.011  0.0132
```

```
## [1] "Top 10 most contributing features:"
```

```
## papers
## position
## call
## topics
## dates
## published
## conference
## workshop
## due
## committee
```

2a,b

```
##    Y hat
## Y    0  1
##   0 10  1
##   1  1  8
```

```
## [1] "Elastic net Error Rate with Test Data: 0.1"
```

```
## [1] "Number of contributing features: 64"
```

```
##       id   name       0-score 1-score
##  [1,] 3036 papers     -0.146  0.1752
##  [2,] 3187 position   0.1269  -0.1522
##  [3,] 596  call       -0.1074 0.1289
##  [4,] 4282 topics     -0.0975 0.117
##  [5,] 1045 dates      -0.0968 0.1161
##  [6,] 3364 published  -0.0968 0.1161
##  [7,] 869  conference -0.0926 0.1111
##  [8,] 4628 workshop   -0.0598 0.0717
##  [9,] 1262 due        -0.0598 0.0717
## [10,] 810  committee  -0.0488 0.0585
## [11,] 4060 submission -0.0455 0.0546
## [12,] 4629 workshops  -0.0343 0.0412
## [13,] 3125 phd        0.0221  -0.0265
## [14,] 3458 record     0.0154  -0.0184
## [15,] 599  camera     -0.0154 0.0184
## [16,] 3433 ready      -0.0154 0.0184
## [17,] 2049 important  -0.011  0.0132
```

	Error Rate	Features Selected
NSC	0	17
Elastic Net	0.1	64
SVM	1	43

Which model would you prefer and why?

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

3. Benjamini-Hochberg

Which features correspond to the rejected hypotheses?

	P.Values <dbl>	T_F <chr>
papers	1.116910e-10	Rejected
submission	7.949969e-10	Rejected
position	8.219362e-09	Rejected
published	1.835157e-07	Rejected
important	3.040833e-07	Rejected
call	3.983540e-07	Rejected
conference	5.091970e-07	Rejected

	P.Values <dbl>	T_F <chr>
candidates	8.612259e-07	Rejected
dates	1.398619e-06	Rejected
paper	1.398619e-06	Rejected

Interpret the result.

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

Appendix

```
knitr::opts_chunk$set(echo = TRUE)
library(readxl)
library(ggplot2)
setwd("C:/Users/Bjorn/Documents/LIU/machine_learning/labs")
data = read_excel("Influenza.xlsx")
ggplot(data=data, aes(x=Time, y=Mortality))+
  geom_bar(stat = "identity")
ggplot(data=data, aes(x=Time, y=Influenza))+
  geom_bar(stat = "identity")
library(mgcv)
gam_model = gam(Mortality~Year+s(Week, k=length(unique(data$Week))), data=data,
                family = gaussian(link = "identity"), method="GCV.Cp")
gam_model2 = gam(Mortality~Year+s(Week)+s(Year, k=length(unique(data$Year))),
                 data=data, family = gaussian(link = "identity"))
summary(gam_model)
plot(gam_model2)
gam_model$sp
# s=interp(data$Year,data$Week, fitted(gam_model))
# plot_ly(x=~s$x , y=~s$y, z=~s$z, type="surface")
plot(gam_model)
plot(gam_model, shift=mean(data$Mortality), residuals=T, pch=1, xlab="") #plot with da
ta points included.
gam.check(gam_model) #Gives some interesting inforamtion about the model.

gam_fitted.results = predict(gam_model, newdata=data)

ggplot(data=data, aes(x=Time, y=gam_fitted.results))+
  geom_bar(stat = "identity")
par(mfrow=c(3,3))
k=c(0.000011,0.0001131932,0.0003,0.0008,0.008,0.08,0.8,10)
for(i in k){
model = gam(Mortality~Year+s(Week, k=length(unique(data$Week))), data=data,
                        family = gaussian(link = "identity"), sp=i)
mod = model[i]
plot(model)
}
ggplot(data=data, aes(x=Time))+
  geom_point(aes(y=gam_model$residuals), color="darkred")+
  geom_point(aes(y=Influenza), color="steelblue")+
  ggtitle("Residuals/ Influenza values against time")
gam_model3 = gam(Mortality~ s(Year, k=length(unique(data$Year)))+s(Week, k=length(uniq
ue(data$Week)))+
                   s(Influenza, k=length(unique(data$Influenza))), data=data, family =
gaussian(link = "identity"))
summary(gam_model3)
plot(gam_model3) # plot new model.
plot(gam_model) # previous model.
ggplot(data=data, aes(x=Time))+
```

```
  geom_line(aes(y=gam_model3$fitted.values), color="darkred")+
  geom_line(aes(y=Mortality), color="steelblue")+
  ggtitle("Mortality/ Fitted Influenza values values against time")
setwd("C:/Users/Bjorn/Documents/LIU/machine_learning/labs")
data<-read.csv2("data.csv", header = TRUE, sep = ";", check.names = FALSE ,encoding =
"latin1")
#names(data)<-iconv(names(data), to = "ASCII", sub = "")

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

train<-na.omit(train)
train2<-train[,-which(names(train) == "Conference")]
train2<-t(train2)

#data2<-list(x=train[,-which(names(train) == "Conference")], y=factor(train$Conferenc
e))
data2<-list(x=train2, y=as.factor(train$Conference),geneid=as.character(1:nrow(train
2)), genenames=rownames(train2))


library(pamr)
pamr_m<-pamr.train(data2)
pamr_cv<-pamr.cv(pamr_m, data2)
pamr_cv

pamr.plotcv(pamr_cv)
pamr.plotcen(pamr_m, data2, threshold=2.757)
test2<-test[,-which(names(train) == "Conference")]
test2<-t(test2)
data3<-list(x=test2, y=test$Conference)

pred_nsc = pamr.predict(pamr_m, threshold = 2.757, newx=test2)
nsc_confmat = table("Y"=test$Conference,"Y hat"=pred_nsc)
nsc_confmat
print(paste("NSC Error Rate with Test Data:", round(1-sum(diag(nsc_confmat))/sum(nsc_c
onfmat),4)))


a=pamr.listgenes(pamr_m,data2,threshold=2.757)

print("Top 10 most contributing features:")
cat( paste( colnames(data)[as.numeric(a[,1])][1:10], collapse='\n' ) ) #paste first 10
features.
library(glmnet)
```

```
library(pamr)
library(kableExtra)
library(kernlab)

xdata = as.matrix(train[,-ncol(train)])
ydata = train[,ncol(train)]
xtest = as.matrix(test[,-ncol(test)])
ytest = test[,ncol(test)]

elasticnet_model = cv.glmnet(x=xdata,y=ydata,alpha=0.5,family="binomial")
elasticnet_model.predict = predict(elasticnet_model, newx = as.matrix(test[,-4703]), t
ype = "class", s="lambda.min")
elasticnet_confmat = table("Y"=test$Conference,"Y hat"=elasticnet_model.predict)
elasticnet_confmat
print(paste("Elastic net Error Rate with Test Data:", round(1-sum(diag(elasticnet_conf
mat))/sum(elasticnet_confmat),4)))
coefs = as.matrix(coef(elasticnet_model, elasticnet_model$lambda.min))
elastic_features <- length(names(coefs[coefs != 0,]))
print(paste("Number of contributing features:",elastic_features))

features = pamr.listgenes(pamr_m, data2, threshold = 2.757, genenames = TRUE)

res1 = list("Error Rate" = round(1-sum(diag(nsc_confmat))/sum(nsc_confmat),4), "Featur
es Selected" = nrow(features))

res2 = list("Error Rate" = round(1-sum(diag(elasticnet_confmat))/sum(elasticnet_confma
t),4), "Features Selected" = elastic_features)

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


svm_pred <- predict(svm, newdata = test)

svm_mat  <- table(ytest, svm_pred)
svm_rate <- 1 - sum(diag(svm_mat)) / sum(svm_mat)

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

result = rbind("NSC" = res1, "Elastic Net" = res2, "SVM" = res3)

knitr::kable(result)
hochberg <- function(x, y, alpha) {
  p <- apply(x, 2, function(x_data){t.test(x_data ~ y, alternative = "two.sided")$p.va
lue})
```

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

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

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

lowest_p
```