Lab2 Block 2

Priya Kurian pullolickal

December 15, 2017

1.Using GAM and GLM to examine the mortality rates

The Excel document influenza.xlsx contains weekly data on the mortality and the number of laboratory-confirmed cases of influenza in Sweden. In addition, there is information about population-weighted temperature anomalies (temperature deficits). The aim of this project is to understand the GAM model and for that we take three variables into consideration(MOrtality,year and Week).From the output of the model we study which features are significant. Effect of penalty factor of spline component on deviance is also studied.Relationship of the penalty factor and degrees of freedom is also observed.

1.1

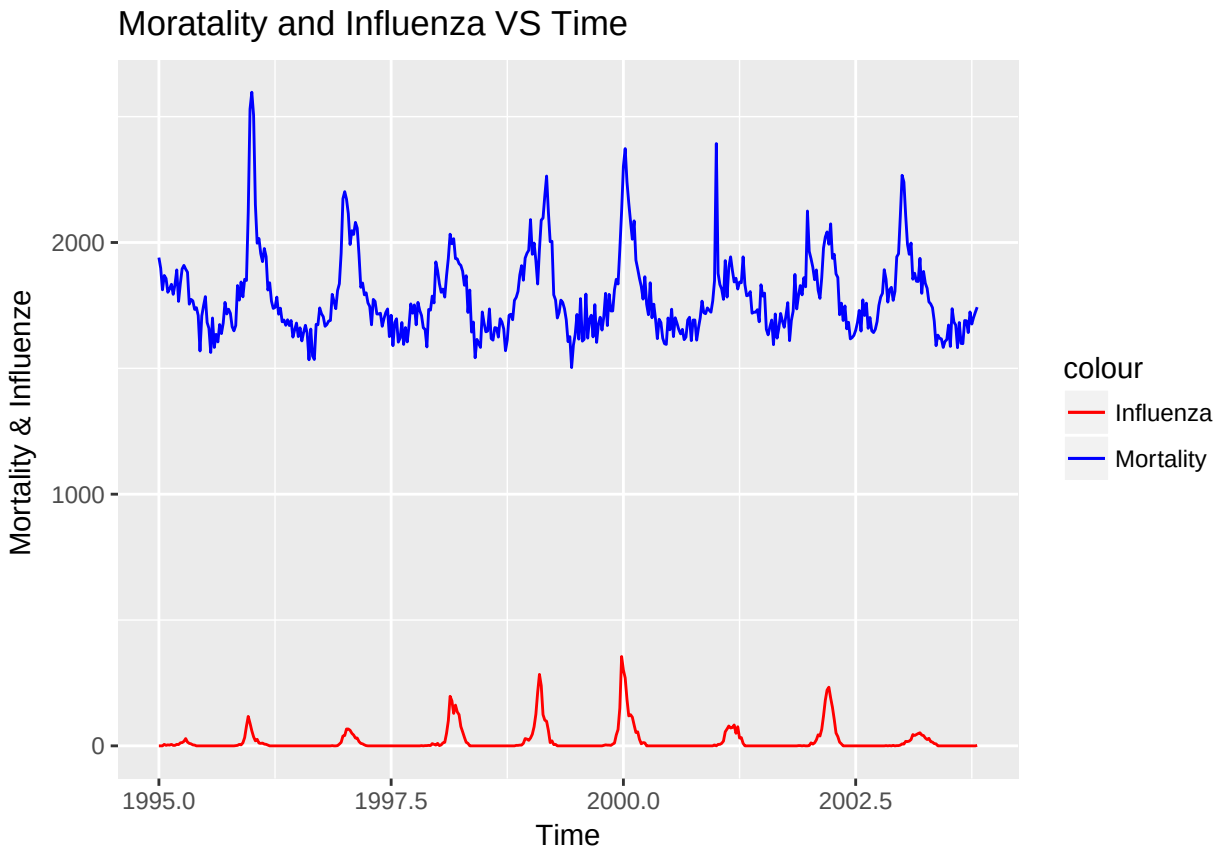
At first two variables (influenza and mortality) are plotted against time.

```
## Loading required package: nlme
```

```
## This is mgcv 1.8-22. For overview type 'help("mgcv-package")'.
```

```
influenza<-read.csv("/home/george/Documents/732A95/lab3/Influenza.csv", sep=";",dec=",")

  ggplot(influenza, aes(Time)) +
  geom_line(aes(y = Mortality, colour = "Mortality")) +
  geom_line(aes(y = Influenza, colour = "Influenza"))+scale_colour_manual(values=c("red", "blue"))+ylab
```



From the graphs it can be concluded that influenza is a reason for the mortality.The infuenza graph follows a seasonal distribution.The dependencies of the two graphs are not very strong.Hence we can say that influenza is just only one of the many reasons for the mortality.

1.2

Implemented a GAM model by using a generalized cross-validation to fit a gaussian model which has a linear terms in Year and spline term in Week.

```
model1 <- gam(Mortality ~ Year + s(Week,k= 52), data = influenza,method="GCV.Cp")
coef(model1)["(Intercept)"]
coef(model1)["Year"]
```

The probablistic model is

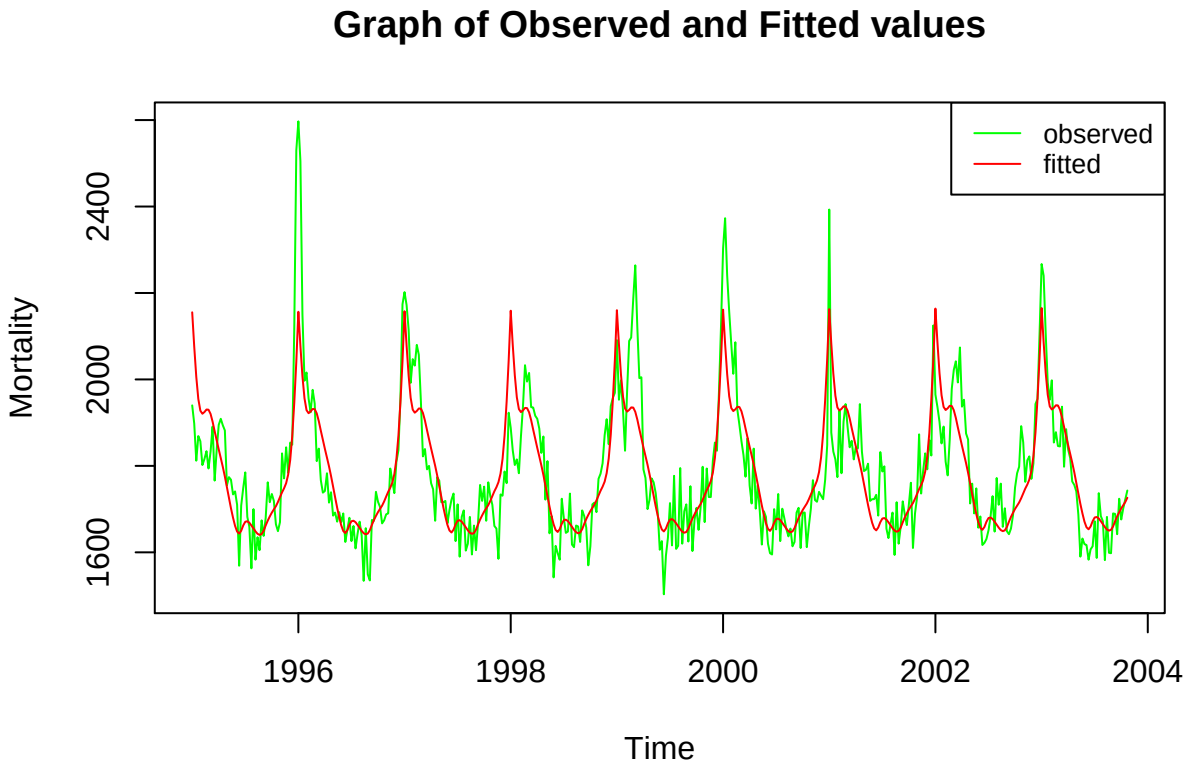
$$Y(Mortality) = \beta_0 + \beta_1(Year) + \ s(week)$$

Our model would be

$$Y(Mortality) = -680.598 + 1.233(Year) + \ s(week)$$

1.3

the graph of fitted vs observed is plotted



From the graphs it is clear that the the fitted graph is similar to the observed graph.Hence we can say that quality of fit is good.

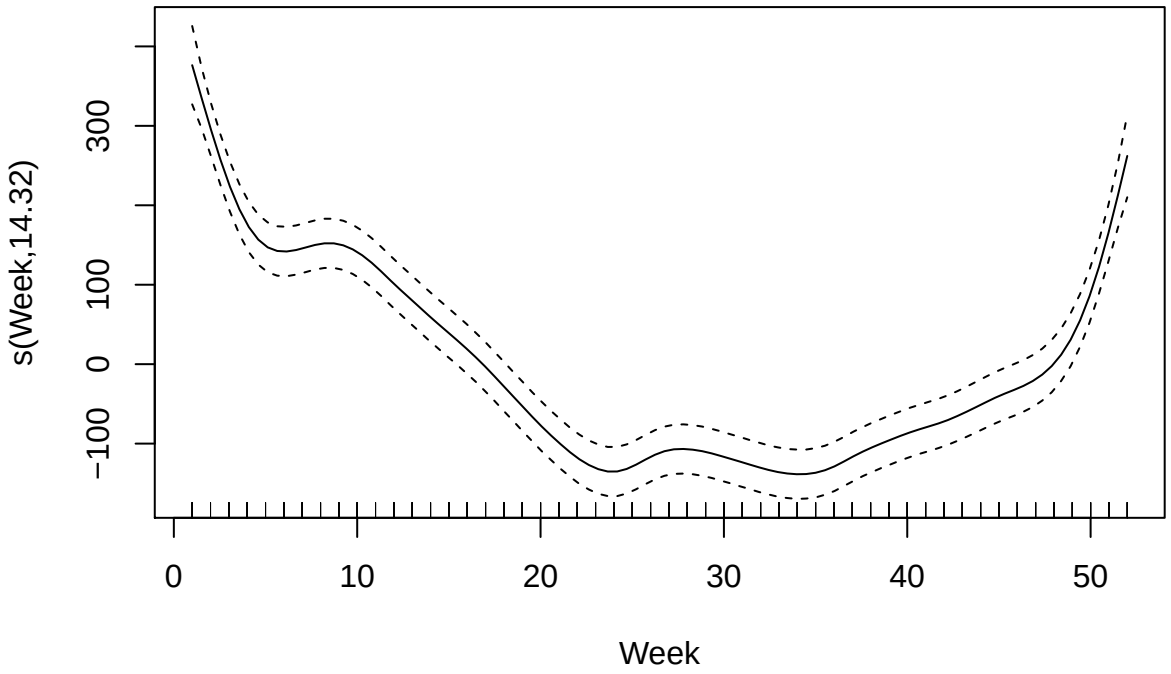
The output of the model is a s below

```
summary(model1)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Mortality ~ Year + s(Week, k = 52)
##
## 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
```

The output shows that the component *Year* is not significant whereas component regarding of *Week* is statistically significant. Hence, there is no significant trend saying that the mortality rate changes from one

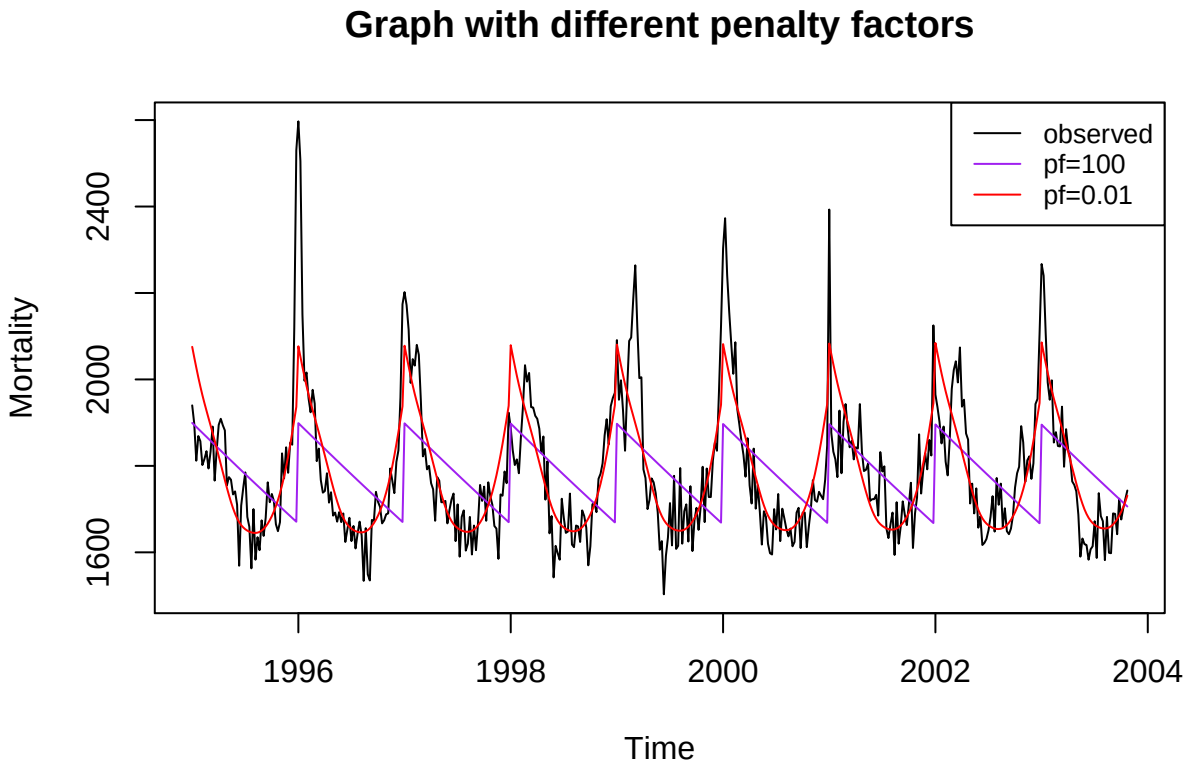
year to another.



This plot visualizes the output of spline component *Week* with standard errors in dashed line. It shows that the mortality has a high value around first weeks. The value suddenly drops during the middle of the week and increases again from week 35 throughout the end of the year. Thus, it can be easily seen that the mortality has a higher value in winter.The conclusion about the model fit is that it seems to be a good model. The pbserved and fitted values follows the same pattern even though *year* is insignificant.

1.4

penalty factor,degrees of freedom,deviance are studied below



Lower penalty factor gives a better fit to the model.Lower penalty has highe degrees of freedom.Our model also proves the same.

```
summary(fit.sp1)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Mortality ~ Year + s(Week, k = 52, sp = 100)
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2789.6277  5381.0481   0.518    0.604
## Year          -0.5032     2.6920  -0.187    0.852
##
## Approximate significance of smooth terms:
##           edf Ref.df     F p-value
## s(Week) 1.007  1.014 95.05  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.172   Deviance explained = 17.6%
## GCV =  21643  Scale est. = 21501     n = 459
```

```
summary(fit.sp2)
```

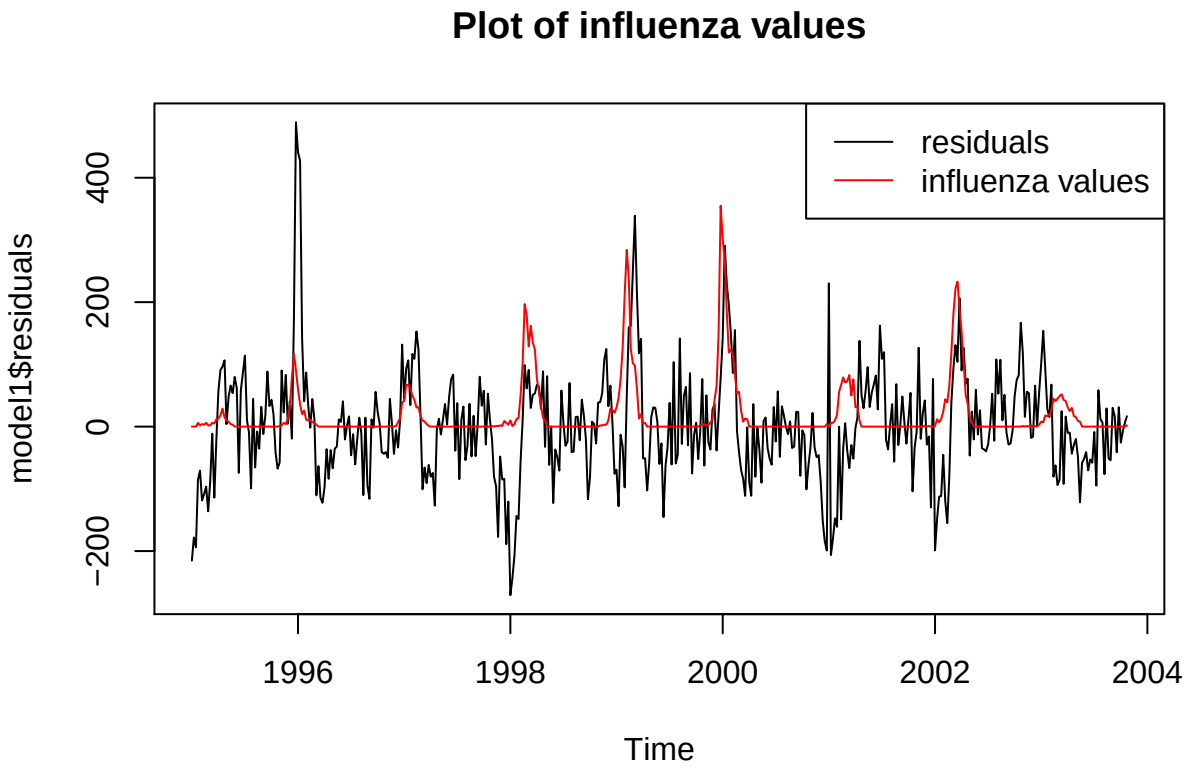
```
##
## Family: gaussian
## Link function: identity
```

```
##
## Formula:
## Mortality ~ Year + s(Week, k = 52, sp = 0.01)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -773.336   3561.839  -0.217    0.828
## Year           1.279      1.782   0.718    0.473
##
## Approximate significance of smooth terms:
##          edf Ref.df     F p-value
## s(Week) 4.69  5.861 134.7  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.638   Deviance explained = 64.3%
## GCV = 9539.1  Scale est. = 9400.1    n = 459
```

For the penalty factor of 0.01,the edf is 4.69 and for penalty factor 100 the edf is 1.007.So our model shows that high degrees of freedom is for lower penaly factor.The Deviance explained = 17.6% for penalt factor 100 and deviance expalined is 64.3% for .01 penalt factor.So the deviance explained is high for lower penaly factor.The deviation from the observed data is less for lower penaly factor.

1.5

Variation of residuals are also studied



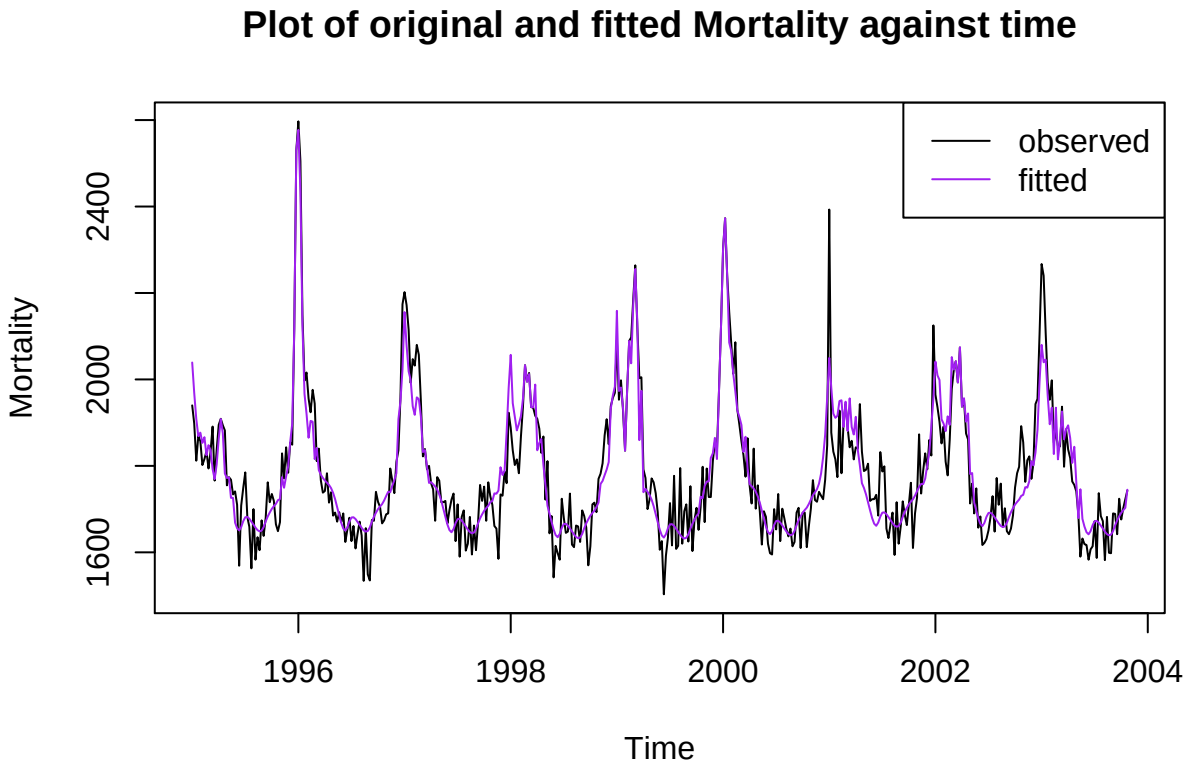
The residuals are correlated well when there is rise in the Influenza values.

1.6

Make a model with additive components

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Mortality ~ s(Influenza, k = 85) + s(Year, k = 9) + s(Week, k = 52)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1783.8        3.2   557.5   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                edf Ref.df      F p-value
## s(Influenza) 69.733 72.841  5.600  <2e-16 ***
## s(Year)       4.663  5.677  1.487   0.181
## s(Week)      14.641 18.248 18.533  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Rank: 134/144
## R-sq.(adj) =  0.819   Deviance explained = 85.4%
## GCV = 5846.7  Scale est. = 4699.8    n = 459
```

The output shows that only the *week* and *influenza* of the spline component are significant. *Year* does not seem to be significant .Thus it can be concluded that mortality is influenced by the outbreaks of influenza.



The plot shows that this model performs really well and it is better than model 2.Hence it can be concluded that the model with *year*,*week* and *influenza* as spline function perform better than the model in the step 2.

Assignment2:High-dimensional methods

The data.csv has the data of 64 emails.Announces of conferences are marked as 1 and the rest as 0.

1.1

The data is first divided into training and test data(70/30) and then we perform nearest shrunken centroid classification of training data in which the threshold is chosen by cross-validation.

The threshhold by cross validation was obtained as 1.3.

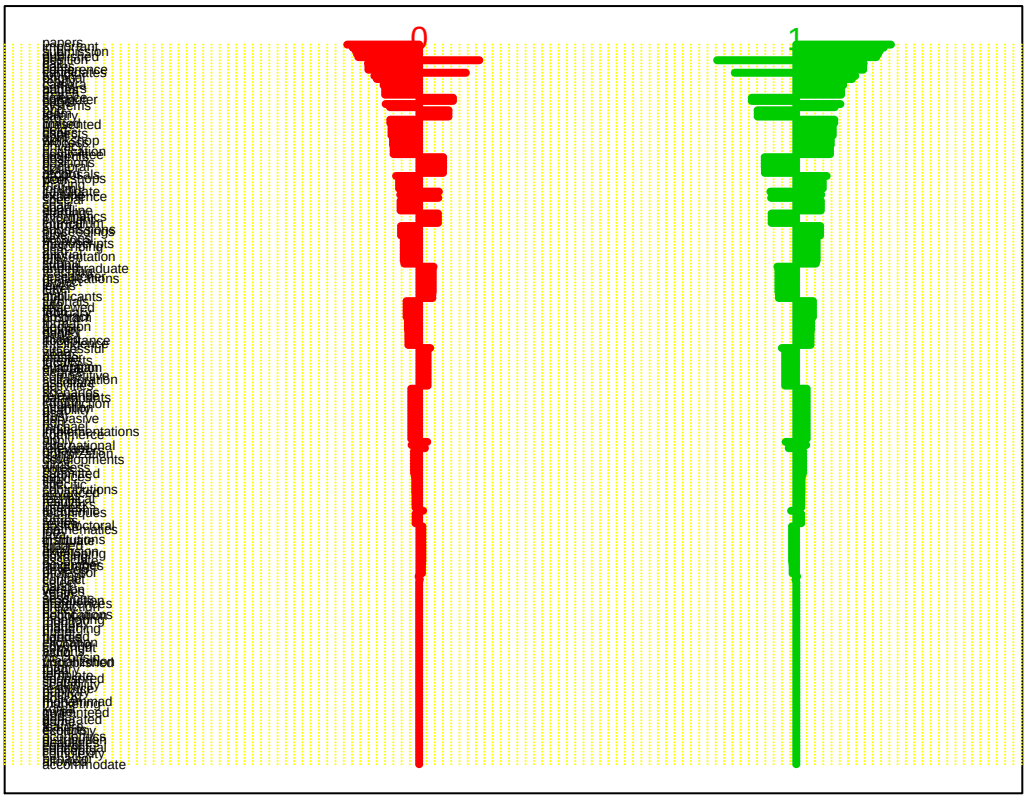
With this threshhold we do the centroid plot as below.

```
## Loading required package: cluster

## Loading required package: survival

## 1234567891011121314151617181920212223242526272829303132333435363738394041

## 12Fold 1 :1234567891011121314151617181920212223242526272829303132333435363738394041
## Fold 2 :1234567891011121314151617181920212223242526272829303132333435363738394041
## Fold 3 :1234567891011121314151617181920212223242526272829303132333435363738394041
## Fold 4 :1234567891011121314151617181920212223242526272829303132333435363738394041
## Fold 5 :1234567891011121314151617181920212223242526272829303132333435363738394041
## Fold 6 :1234567891011121314151617181920212223242526272829303132333435363738394041
## Fold 7 :1234567891011121314151617181920212223242526272829303132333435363738394041
## Fold 8 :1234567891011121314151617181920212223242526272829303132333435363738394041
## Fold 9 :1234567891011121314151617181920212223242526272829303132333435363738394041
## Fold 10 :1234567891011121314151617181920212223242526272829303132333435363738394041
```



We can see from the centroid plot that the centroids are getting smaller when going down showing they are irrelevant.The total number of features selected with the threshhold of 1.3 is 231.

```
##         id   0-score 1-score
##    [1,] 3036 -0.3822 0.5029
##    [2,] 2049 -0.3527 0.4641
##    [3,] 4060 -0.3376 0.4441
##    [4,] 1262 -0.3309 0.4354
##    [5,] 3364 -0.3231 0.4252
##    [6,] 3187 0.3188  -0.4195
##    [7,] 596  -0.2725 0.3585
##    [8,] 869  -0.2706 0.356
##    [9,] 1045 -0.2706 0.356
##   [10,] 607  0.2476  -0.3258
##   [11,] 4282 -0.2383 0.3136
##   [12,] 2990 -0.2254 0.2966
##   [13,] 599  -0.1896 0.2495
##   [14,] 3433 -0.1896 0.2495
##   [15,] 389  -0.1815 0.2389
##   [16,] 2588 -0.1815 0.2389
##   [17,] 3022 -0.1815 0.2389
##   [18,] 850  0.1793  -0.2359
##   [19,] 3725 0.1793  -0.2359
##   [20,] 3035 -0.1785 0.2349
##   [21,] 4129 -0.1559 0.2051
##   [22,] 3125 0.1559  -0.2051
##   [23,] 4177 0.1556  -0.2047
##   [24,] 3671 0.1556  -0.2047
##   [25,] 2974 -0.1542 0.2029
```

```
##  [26,] 2463 -0.1542 0.2029
##  [27,] 329  -0.148  0.1947
##  [28,] 681  -0.148  0.1947
##  [29,] 1891 -0.148  0.1947
##  [30,] 3243 -0.148  0.1947
##  [31,] 283  -0.14   0.1842
##  [32,] 4628 -0.14   0.1842
##  [33,] 3286 -0.14   0.1842
##  [34,] 3274 -0.1368 0.18
##  [35,] 810  -0.1368 0.18
##  [36,] 2889 -0.1368 0.18
##  [37,] 1233 0.1272  -0.1674
##  [38,] 3188 0.1272  -0.1674
##  [39,] 3191 0.1272  -0.1674
##  [40,] 3312 0.1272  -0.1674
##  [41,] 3891 0.1265  -0.1664
##  [42,] 3458 0.1265  -0.1664
##  [43,] 3324 -0.1231 0.162
##  [44,] 1643 -0.1077 0.1417
##  [45,] 2561 -0.1077 0.1417
##  [46,] 3090 -0.1077 0.1417
##  [47,] 4629 -0.1077 0.1417
##  [48,] 606  0.1041  -0.137
##  [49,] 2058 -0.1012 0.1332
##  [50,] 1501 0.1012  -0.1332
##  [51,] 3952 -0.1    0.1316
##  [52,] 680  -0.1    0.1316
##  [53,] 3836 -0.1    0.1316
##  [54,] 1061 -0.0998 0.1314
##  [55,] 1007 0.0995  -0.1309
##  [56,] 1477 0.0995  -0.1309
##  [57,] 2103 0.0995  -0.1309
##  [58,] 3992 0.0995  -0.1309
##  [59,] 2295 -0.0971 0.1278
##  [60,] 4061 -0.0971 0.1278
##  [61,] 2305 -0.097  0.1276
##  [62,] 3285 -0.097  0.1276
##  [63,] 92   -0.0832 0.1094
##  [64,] 1127 -0.0832 0.1094
##  [65,] 2583 -0.0832 0.1094
##  [66,] 3323 -0.0832 0.1094
##  [67,] 4500 -0.0832 0.1094
##  [68,] 1698 -0.0832 0.1094
##  [69,] 3241 -0.0832 0.1094
##  [70,] 4364 -0.0832 0.1094
##  [71,] 4062 -0.0796 0.1048
##  [72,] 4039 0.0757  -0.0996
##  [73,] 740  0.0721  -0.0949
##  [74,] 2438 0.0721  -0.0949
##  [75,] 2442 0.0721  -0.0949
##  [76,] 3311 0.0721  -0.0949
##  [77,] 3383 0.0721  -0.0949
##  [78,] 3559 0.0721  -0.0949
##  [79,] 4176 0.0721  -0.0949
```

```
##  [80,] 4402 0.0721  -0.0949
##  [81,] 267  0.0701  -0.0923
##  [82,] 2553 0.0701  -0.0923
##  [83,] 63   -0.0681 0.0896
##  [84,] 1563 -0.0681 0.0896
##  [85,] 1594 -0.0681 0.0896
##  [86,] 3589 -0.0681 0.0896
##  [87,] 3882 -0.0681 0.0896
##  [88,] 4365 -0.0681 0.0896
##  [89,] 3301 -0.0612 0.0805
##  [90,] 1636 -0.061  0.0802
##  [91,] 1072 -0.061  0.0802
##  [92,] 386  -0.061  0.0802
##  [93,] 2198 -0.0586 0.0771
##  [94,] 3021 -0.0586 0.0771
##  [95,] 3386 -0.0586 0.0771
##  [96,] 76   -0.0583 0.0767
##  [97,] 2150 -0.0583 0.0767
##  [98,] 4075 0.0579  -0.0762
##  [99,] 107  0.0447  -0.0589
## [100,] 336  0.0447  -0.0589
## [101,] 776  0.0447  -0.0589
## [102,] 831  0.0447  -0.0589
## [103,] 1088 0.0447  -0.0589
## [104,] 1450 0.0447  -0.0589
## [105,] 1456 0.0447  -0.0589
## [106,] 1542 0.0447  -0.0589
## [107,] 2170 0.0447  -0.0589
## [108,] 2613 0.0447  -0.0589
## [109,] 2837 0.0447  -0.0589
## [110,] 4529 0.0447  -0.0589
## [111,] 363  -0.0429 0.0564
## [112,] 879  -0.0429 0.0564
## [113,] 2433 -0.0429 0.0564
## [114,] 3051 -0.0429 0.0564
## [115,] 3514 -0.0429 0.0564
## [116,] 3711 -0.0429 0.0564
## [117,] 4449 -0.0429 0.0564
## [118,] 501  -0.0429 0.0564
## [119,] 803  -0.0429 0.0564
## [120,] 2046 -0.0429 0.0564
## [121,] 2082 -0.0429 0.0564
## [122,] 2690 -0.0429 0.0564
## [123,] 2877 -0.0429 0.0564
## [124,] 3118 -0.0429 0.0564
## [125,] 4342 -0.0429 0.0564
## [126,] 4451 -0.0429 0.0564
## [127,] 4452 -0.0429 0.0564
## [128,] 272  0.0425  -0.0559
## [129,] 2175 -0.0408 0.0537
## [130,] 3515 0.0302  -0.0397
## [131,] 172  -0.0284 0.0373
## [132,] 1149 -0.0284 0.0373
## [133,] 2219 -0.0284 0.0373
```

```
## [134,] 2964 -0.0284 0.0373
## [135,] 2984 -0.0284 0.0373
## [136,] 2887 -0.0284 0.0373
## [137,] 4605 -0.0284 0.0373
## [138,] 4064 -0.028  0.0369
## [139,] 3800 -0.0238 0.0313
## [140,] 134  -0.0222 0.0292
## [141,] 919  -0.0222 0.0292
## [142,] 3957 -0.0222 0.0292
## [143,] 4268 -0.0222 0.0292
## [144,] 4281 -0.0222 0.0292
## [145,] 2220 -0.0211 0.0277
## [146,] 2847 -0.0211 0.0277
## [147,] 3582 -0.0211 0.0277
## [148,] 4181 -0.0211 0.0277
## [149,] 2167 -0.0204 0.0268
## [150,] 67   0.0204  -0.0268
## [151,] 2005 -0.0203 0.0267
## [152,] 4185 -0.0203 0.0267
## [153,] 3588 -0.0203 0.0267
## [154,] 3794 -0.0203 0.0267
## [155,] 579  0.017   -0.0223
## [156,] 1147 0.017   -0.0223
## [157,] 1524 0.017   -0.0223
## [158,] 1591 0.017   -0.0223
## [159,] 1702 0.017   -0.0223
## [160,] 1797 0.017   -0.0223
## [161,] 2141 0.017   -0.0223
## [162,] 2251 0.017   -0.0223
## [163,] 2278 0.017   -0.0223
## [164,] 2619 0.017   -0.0223
## [165,] 3194 0.017   -0.0223
## [166,] 340  0.017   -0.0223
## [167,] 2894 0.0156  -0.0205
## [168,] 1144 0.0148  -0.0195
## [169,] 2392 0.0148  -0.0195
## [170,] 3295 0.0148  -0.0195
## [171,] 2713 -0.0036 0.0048
## [172,] 899  0.0032  -0.0042
## [173,] 1859 -0.0011 0.0015
## [174,] 3764 -0.0011 0.0015
## [175,] 104  -0.0011 0.0015
## [176,] 940  -0.0011 0.0015
## [177,] 967  -0.0011 0.0015
## [178,] 1343 -0.0011 0.0015
## [179,] 1587 -0.0011 0.0015
## [180,] 1861 -0.0011 0.0015
## [181,] 1965 -0.0011 0.0015
## [182,] 2574 -0.0011 0.0015
## [183,] 2623 -0.0011 0.0015
## [184,] 2754 -0.0011 0.0015
## [185,] 2757 -0.0011 0.0015
## [186,] 2839 -0.0011 0.0015
## [187,] 2890 -0.0011 0.0015
```

```
## [188,] 3169 -0.0011 0.0015
## [189,] 3224 -0.0011 0.0015
## [190,] 3231 -0.0011 0.0015
## [191,] 3289 -0.0011 0.0015
## [192,] 3802 -0.0011 0.0015
## [193,] 3943 -0.0011 0.0015
## [194,] 4490 -0.0011 0.0015
## [195,] 4499 -0.0011 0.0015
## [196,] 84   -0.0011 0.0015
## [197,] 196  -0.0011 0.0015
## [198,] 455  -0.0011 0.0015
## [199,] 837  -0.0011 0.0015
## [200,] 856  -0.0011 0.0015
## [201,] 857  -0.0011 0.0015
## [202,] 920  -0.0011 0.0015
## [203,] 1062 -0.0011 0.0015
## [204,] 1214 -0.0011 0.0015
## [205,] 1291 -0.0011 0.0015
## [206,] 1292 -0.0011 0.0015
## [207,] 1490 -0.0011 0.0015
## [208,] 1560 -0.0011 0.0015
## [209,] 1721 -0.0011 0.0015
## [210,] 1745 -0.0011 0.0015
## [211,] 1818 -0.0011 0.0015
## [212,] 1833 -0.0011 0.0015
## [213,] 2197 -0.0011 0.0015
## [214,] 2359 -0.0011 0.0015
## [215,] 2598 -0.0011 0.0015
## [216,] 2746 -0.0011 0.0015
## [217,] 2888 -0.0011 0.0015
## [218,] 3259 -0.0011 0.0015
## [219,] 3361 -0.0011 0.0015
## [220,] 3570 -0.0011 0.0015
## [221,] 3703 -0.0011 0.0015
## [222,] 3948 -0.0011 0.0015
## [223,] 3966 -0.0011 0.0015
## [224,] 4202 -0.0011 0.0015
## [225,] 4212 -0.0011 0.0015
## [226,] 4236 -0.0011 0.0015
## [227,] 4363 -0.0011 0.0015
## [228,] 4435 -0.0011 0.0015
## [229,] 4526 -0.0011 0.0015
## [230,] 4606 -0.0011 0.0015
## [231,] 4664 -0.0011 0.0015
```

The 10 most contributing features are as below

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

```
## papers
## important
## submission
## due
## published
```

```
## position
## call
## conference
## dates
## candidates
```

The words above are defenetly used in conference calls and hence it is reasonable that they have strong effect on the discrimination between the conference mail and others.

The test error is as below

```
best_model <- pamr.train(mydata, threshold = tr)
```

```
## 1
```

```
pred_centroid <- pamr.predict(fit = best_model, newx = as.matrix(test), threshold = tr, type = "centroi

tabl_cenroid<-pamr.confusion(fit = model, threshold = tr, extra = TRUE)
```

```
##    0  1 Class Error rate
## 0 22  3       0.12000000
## 1  1 18       0.05263158
## Overall error rate= 0.091
```

2.2

Elastic net

Elastic net is another method for training the data and we find the test error for that. The test error for the elastic net is calculated as below

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loading required package: foreach
```

```
## Loaded glmnet 2.0-13
```

```
##selecting the penalty by corss validation
glmnet_cv<-cv.glmnet(x=t(x),y=y,family="binomial",alpha=0.5)

penalt<-glmnet_cv$lambda.min

elas_model<-glmnet(x=t(x),y=y,family = "binomial",lambda = penalt,alpha = 0.5)



##Finding the test error


pred<-predict(elas_model,newx=t(x_test),type="class")
tabl_elsnet<-table(pred,test$Conference)
misclas<-1-(sum(diag(tabl_elsnet))/sum(tabl_elsnet))
```

The test error is found as *0.1*.

The features are selected the following way

```
features_selected <- length(which(coef(elas_model)!=0))
```

the number of features selected are *39*.

b.Support vector machine

For support vector machine the test error is calculated as below

```
library(kernlab)
```

```
##
## Attaching package: 'kernlab'
```

```
## The following object is masked from 'package:ggplot2':
##
##     alpha
```

```
svm_model<-ksvm(Conference ~.,data=train_data,kernel="vanilladot")
```

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

```
## Warning in .local(x, ...): Variable(s) `' constant. Cannot scale data.
```

```
##Finding the test error

pred2<-predict(svm_model,t(x_test))
svm_tabl<-table(prediction=pred2,test$Conference)
misclas2<-1-(sum(diag(svm_tabl))/sum(svm_tabl))

features_svm <- length(coef(svm_model)[[1]])
```

The test errror is obtained as *0.5*.

The number of contributing features are *43*.

From the results above test error is minimum for the svm and features are like 43.Hence we conclude that svm is better than centroid and elastic net.

3.Benjamini-Hockberg

Implemented Benjamini-Hochberg method for the original data and t.test with two sided is used for computing p-values.

Amount of features corresponding to the rejected hypothesis are as below

```
##         apply       authors          call        camera     candidate    candidates        chairs
##           272           389           596           599           606           607           681
##    submission          team        topics      workshop
##          4060          4177          4282          4628
```

features correspond to the rejected hypotheses are those that contributes most to the announcements of conference.

```
library(mgcv)

library(ggplot2)
##1
influenza<-read.csv("/home/george/Documents/732A95/lab3/Influenza.csv", sep=";",dec=",")
```

```r
# plot(x=influenza$Time, y=influenza$Mortality, type='l', ylab="Mortality", xlab="Time", col="red", mai
# plot(x=influenza$Time, y=influenza$Influenza, type='l', ylab="Influenza number", xlab="Time", col="re


  ggplot(influenza, aes(Time)) +
  geom_line(aes(y = Mortality, colour = "Mortality")) +
  geom_line(aes(y = Influenza, colour = "Influenza"))+scale_colour_manual(values=c("red", "blue"))+ylab



##2

model1 <- gam(Mortality ~ Year + s(Week,k= 52), data = influenza,method="GCV.Cp")

model1$coefficients

##3

model1 <- gam(Mortality ~ Year + s(Week,k= 52), data = influenza,method="GCV.Cp")

plot(influenza$Time, influenza$Mortality, type='l',col="green" ,xlab="Time", ylab="Mortality",main = "G
points(influenza$Time, model1$fitted.values, type='l', col="red")
legend("topright", c("observed", "fitted"), col=c("green","red"), lwd=1, cex=0.8)


summary(model1)

plot(model1)



##4


fit.sp1 <- gam(Mortality ~ Year + s(Week,k=52, sp=100), data=influenza)

fit.sp2 <- gam(Mortality ~ Year + s(Week,k=52, sp=.01), data=influenza)

plot(x=influenza$Time, y=influenza$Mortality, type='l', xlab="Time", ylab="Mortality", lwd=1)
points(x=influenza$Time, y=fit.sp1$fitted.values, type='l', col="purple", lwd=1)
points(x=influenza$Time, y=fit.sp2$fitted.values, type='l', col="red", lwd=1)
legend("topright", c("observed","pf=100","pf=0.01"), col=c("black","purple","red"),lwd=1, cex=0.8)


summary(fit.sp2)


##5.
plot(x=influenza$Time, y=model1$residuals, type='l', xlab="Time", col="black",main = "Plot of influenza
points(influenza$Time, influenza$Influenza, type='l', col='red')
legend("topright", c("residuals","influenza values"), col=c("black","red"), lwd=1)
```

```
##6.

model2 <- gam(Mortality ~ s(Influenza,k =85) + s(Year,k=9) + s(Week,k=52), data = influenza)
summary(model2)
plot(x=influenza$Time, y=influenza$Mortality, type='l', xlab="Time", ylab="Mortality", main = "Plot of"
points(x=influenza$Time, y=model2$fitted.values, type='l', col="purple")
legend("topright", c("observed","fitted"), col=c("black","purple"), lwd=1)


################################################################################

library(pamr)
setwd("/home/george/Documents/732A95/lab3/")
datam<-read.csv("data.csv",stringsAsFactors = FALSE, sep=";", fileEncoding = 'WINDOWS-1252')

data=datam
data=as.data.frame(data)
data$Conference=as.factor(datam$Conference)
rownames(data)=1:nrow(data)
n=dim(data)[1]
set.seed(12345)
trainIndex=sample(1:n, floor(n*.7))
train_data=data[trainIndex,]
test=data[-trainIndex,]
x=t(train_data[,-ncol(train_data)])
x_test=t(test[,-ncol(test)])
y=train_data$Conference
mydata=list(x=x,y=as.factor(y),geneid=as.character(1:nrow(x)),genenames=rownames(x))
model=pamr.train(mydata,threshold=seq(0,4, 0.1))

##doing cross vaildation to find the threshhold
cvmodel=pamr.cv(model,mydata)


##tr is the threshold
tr <- cvmodel$threshold[which.min(cvmodel$error)]


##finding the centroid plot
pamr.plotcen(model, mydata, threshold=tr)
 a=pamr.listgenes(model,mydata,threshold=tr)
 #10 most contributing features
 cat(paste(colnames(data)[as.numeric(a[,1])][1:10], collapse = "\n"))

best_model <- pamr.train(mydata, threshold = tr)
##finding the test error
pred_centroid <- pamr.predict(fit = best_model, newx = as.matrix(test), threshold = tr, type = "centroi

tabl_cenroid<-pamr.confusion(fit = model, threshold = tr, extra = TRUE)


# tabl_cenroid<-table(pred_centroid,test$Conference)
# misclas_centroid<-1-(sum(diag(tabl_cenroid))/sum(tabl_cenroid))
```

```r
# misclas_centroid

####Elastic net

library(glmnet)
##selecting the penalty by corss validation
glmnet_cv<-cv.glmnet(x=t(x),y=y,family="binomial",alpha=0.5)

penalt<-glmnet_cv$lambda.min

elas_model<-glmnet(x=t(x),y=y,family = "binomial",lambda = penalt,alpha = 0.5)



##Finding the test error


pred<-predict(elas_model,newx=t(x_test),type="class")
tabl_elsnet<-table(pred,test$Conference)
misclas<-1-(sum(diag(tabl_elsnet))/sum(tabl_elsnet))
misclas

features_selected <- length(which(coef(elas_model)!=0))
####using SVM

library(kernlab)

svm_model<-ksvm(Conference ~.,data=train_data,kernel="vanilladot")

##Finding the test error

pred2<-predict(svm_model,t(x_test))
svm_tabl<-table(prediction=pred2,test$Conference)
misclas2<-1-(sum(diag(svm_tabl))/sum(svm_tabl))
misclas2

###Benjamini Hockberg Method
#

#Benjamini-Hochberg method
pval=c()
pval <-apply(data[-4703],2, FUN=function(x){
  as.numeric(t.test(x~data$Conference)$p.value)})

j <- 1:length(pval)
M <- length(pval)
slope <- 0.05*j/M

pval.ord <- sort(as.numeric(pval))
for(i in 1:length(pval)){
  if(pval.ord[i] >= slope[i]){break}

}
```

```
col=c()
slopeval <- as.numeric(pval[order(pval)[i]])
col[pval.ord<slopeval] <- 1
col[pval.ord>=slopeval] <- 2
```