project

Franciscek Soldek & Maciej Wrona

0. Required packages

```
library(mlr3measures)
library(pdfCluster)
library(clevr)
library(fpc)
library(dendextend)
library(poLCA)
library(clustMD)
library(ContaminatedMixt)
library(clustvarsel)
library(flexmix)
library(pgmm)
library(broom)
library(spdep)
library(CARBayes)
library(sp)
library(tinytex)
library(dplyr)
library(cluster)
```

1. Analysis based on SMR

```
###.csv files

expected_data=read.csv("expected_counts.csv", header=T, sep=",")
observed_data=read.csv("respiratory_admissions.csv", header=T, sep=",")

expected_data$Id=expected_data$code
observed_data$Id=observed_data$IG

expected_data=expected_data[,-1]
```

```
observed_data=observed_data[,-1]
```

```
data=merge(expected_data, observed_data, by="Id")
data=as.data.frame(data)
data$SMR2008 <- data$Y2008/data$E2008
data$SMR2009 <- data$Y2009/data$E2009
```

```
###shape files
#install.packages("sf")
library(sf)
shape<-read_sf("SG_IntermediateZoneBdry_2001/")
```

```
sf.data <- merge(shape, data, all.x=FALSE, by.x="IZ_CODE", by.y="Id")
plot(sf.data$geometry)
```

Calculating SMR for each year

```
selected_columns <- c("E2008", "E2009", "Y2008", "Y2009", "SMR2008", "SMR2009")
result_df <- data.frame()

for (col in selected_columns) {
  # Calculate statistics for the current column
  result <- data %>%
```

```
    summarise(
      variable = col,
      mean = mean(!!sym(col)),
      median = median(!!sym(col)),
      min = min(!!sym(col)),
      max = max(!!sym(col)),
      sd = sd(!!sym(col)),
      quantile_25 = quantile(!!sym(col), 0.25),
      quantile_50 = quantile(!!sym(col), 0.50),
      quantile_75 = quantile(!!sym(col), 0.75)
    )

  result_df <- rbind(result_df, result)
}

print(result_df)
```

```
  variable       mean     median        min        max         sd quantile_25
1    E2008 92.0874597 88.8573298 47.4326143 173.751213 23.7365951  72.9991652
2    E2009 89.3212321 85.6880320 44.7319115 164.818117 22.7283142  70.4519953
3    Y2008 81.0332103 75.0000000 10.0000000 208.000000 36.9601738  53.5000000
4    Y2009 78.1033210 73.0000000 20.0000000 190.000000 34.2409278  52.5000000
5  SMR2008  0.8850904  0.8558286  0.2091262   2.187123  0.3460267   0.6176795
6  SMR2009  0.8809044  0.8663050  0.3251533   1.937355  0.3311917   0.5924330
  quantile_50 quantile_75
1  88.8573298  109.328598
2  85.6880320  106.285232
3  75.0000000  102.000000
4  73.0000000  100.000000
5   0.8558286    1.127612
6   0.8663050    1.116339
```

```
library(sp)
sp.data<-as_Spatial(sf.data)
```

For the purpose of this analysis, we present two plots, each comprising two maps: one for the year 2008 and another for the year 2009. The second plot is overlaid on OpenStreetMap, which may lead to new insights compared to mapping the risk with ggplots.

```
Warning: `tidy.SpatialPolygonsDataFrame()` was deprecated in broom 1.0.4.
i Please use functions from the sf package, namely `sf::st_as_sf()`, in favor
  of sp tidiers.
This warning is displayed once every 8 hours.
Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
generated.
```

Creating the first ggplot for SMR2008

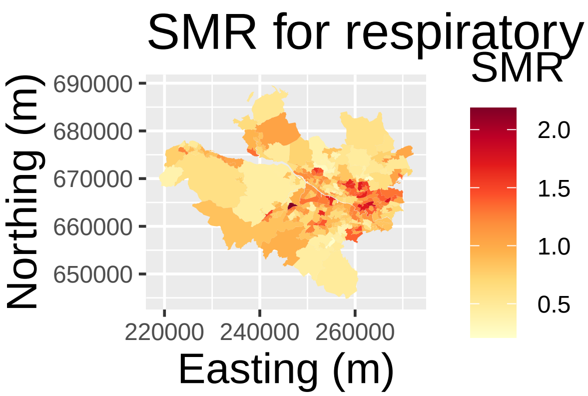
```
#
plot1 <- ggplot(data = sp.data2, aes(x=long, y=lat, group=group, fill = c(SMR2008))) +
  geom_polygon() +
  coord_equal() +
  xlab("Easting (m)") +
  ylab("Northing (m)") +
  labs(title = "SMR for respiratory hospitalisation, year 2008", fill = "SMR") +
  theme(title = element_text(size=16)) +
  scale_fill_gradientn(colors=brewer.pal(n=9, name="YlOrRd"))
```

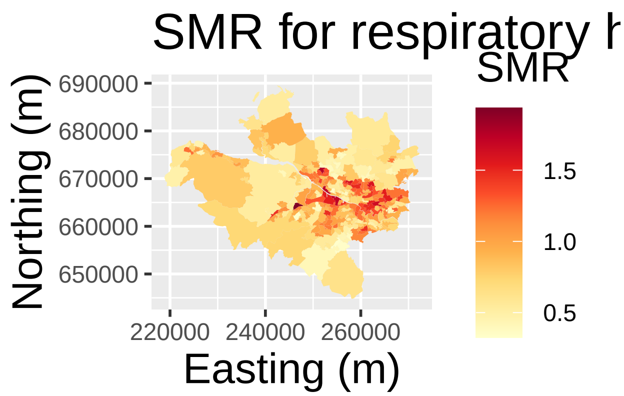
Creating the second ggplot for SMR2009

```
#
plot2 <- ggplot(data = sp.data2, aes(x=long, y=lat, group=group, fill = c(SMR2009))) +
  geom_polygon() +
  coord_equal() +
  xlab("Easting (m)") +
  ylab("Northing (m)") +
  labs(title = "SMR for respiratory hospitalisation, year 2009", fill = "SMR") +
  theme(title = element_text(size=16)) +
  scale_fill_gradientn(colors=brewer.pal(n=9, name="YlOrRd"))
```

Arranging the plots side by side

```
plots_side_by_side <- plot1 + plot2

plots_side_by_side
```





We also plotted leaflet map (only available in the Markdown file).

The main conclusions are of the visual analisys are as follow:

1. The smaller the civil parish (administrative unit) the higher SMR seems to be. This fact is visible for both years.

2. Civil parishes through which the main roads run seems to have a higher SMR than the other administrative units. However, it would be very difficult to confirm such hypothesis, as we do not have access to such data.

We also present the analysis of spatial autocorrelation in each of the years by using Moran's I statistic.

```r
##spatial autocorrelation
library(spdep)
W.nb <- poly2nb(sf.data, row.names = rownames(sf.data))

summary(W.nb)
```

```
Neighbour list object:
Number of regions: 271
Number of nonzero links: 1424
Percentage nonzero weights: 1.938971
Average number of links: 5.254613
```

```
2 disjoint connected subgraphs
Link number distribution:

 1  2  3  4  5  6  7  8  9 10 11 14 15 20 
 4 15 30 51 62 50 32 12  5  5  2  1  1  1 
4 least connected regions:
202 227 239 265 with 1 link
1 most connected region:
234 with 20 links
```

```r
W <- nb2mat(W.nb, style = "B")

W.list <- nb2listw(W.nb, style = "B")


sf.data$SMR2008= sf.data$Y2008/sf.data$E2008
sf.data$SMR2009= sf.data$Y2009/sf.data$E2009

moran.mc(x = sf.data$SMR2008, listw = W.list, nsim = 10000)
```

```

    Monte-Carlo simulation of Moran I

data:  sf.data$SMR2008 
weights: W.list  
number of simulations + 1: 10001 

statistic = 0.40434, observed rank = 10001, p-value = 9.999e-05
alternative hypothesis: greater
```

We conclude that I statistic equals to 0.404 and is significantly different from independence, thus it provides evidence that there is spatial autocorrelation in the SMR2008 variable

```r
moran.mc(x = sf.data$SMR2009, listw = W.list, nsim = 10000)
```

```

    Monte-Carlo simulation of Moran I

data:  sf.data$SMR2009 
weights: W.list  
number of simulations + 1: 10001 

statistic = 0.39019, observed rank = 10001, p-value = 9.999e-05
alternative hypothesis: greater
```

We conclude that I statistic equals to 0.39 and is significantly different from independence, thus it provides evidence that there is spatial autocorrelation in the SMR2009s variable

2. Leroux model

```r
formula2008 <- Y2008 ~ offset(log(E2008))
formula2009 <- Y2009 ~ offset(log(E2009))

library(CARBayes)
model2008 <- S.CARleroux(formula=formula2008, family="poisson", data=sf.data, W=W,
burnin=10000, n.sample=100000, thin=10, verbose=FALSE)
print(model2008)
```

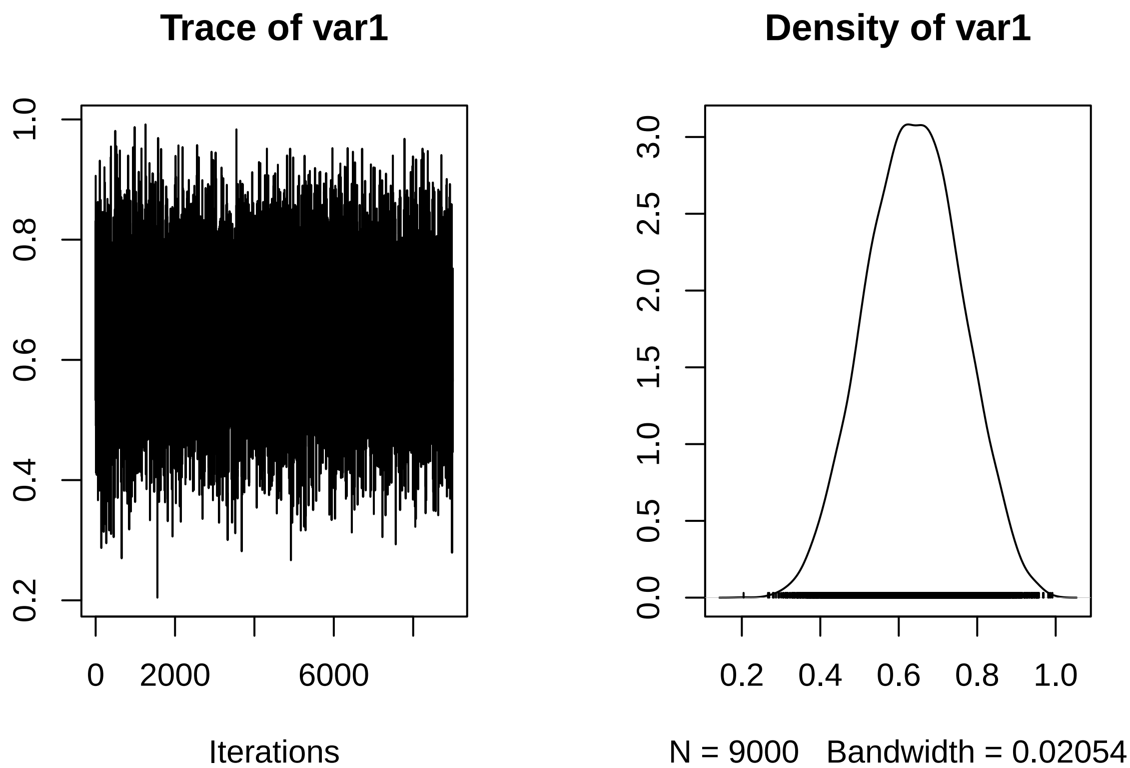
```

#################
#### Model fitted
#################
Likelihood model - Poisson (log link function)
Random effects model - Leroux CAR
Regression equation - Y2008 ~ offset(log(E2008))


#################
#### MCMC details
#################
Total number of post burnin and thinned MCMC samples generated - 9000
Number of MCMC chains used - 1
Length of the burnin period used for each chain - 10000
Amount of thinning used - 10

############
#### Results
############
Posterior quantities and DIC

               Mean    2.5%   97.5% n.effective Geweke.diag
(Intercept) -0.1953 -0.2138 -0.1773      6907.5         0.2
tau2         0.3440  0.2586  0.4467      6871.1        -0.3
rho          0.6386  0.4072  0.8678      6452.3        -1.3

DIC =  2175.731       p.d =  236.6717       LMPL =  -1179.63
```

```r
model2009 <- S.CARleroux(formula=formula2009, family="poisson", data=sf.data, W=W,
burnin=10000, n.sample=100000, thin=10, verbose=FALSE)
print(model2009)
```

```

#################
```

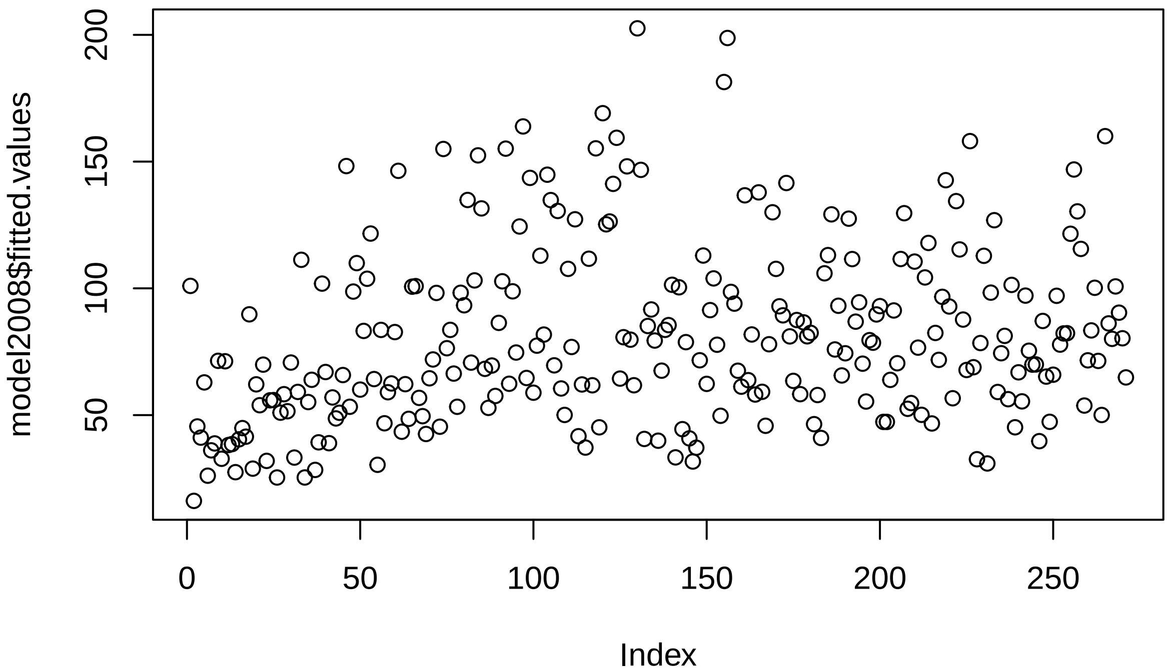
```
#### Model fitted
#################
Likelihood model - Poisson (log link function)
Random effects model - Leroux CAR
Regression equation - Y2009 ~ offset(log(E2009))


#################
#### MCMC details
#################
Total number of post burnin and thinned MCMC samples generated - 9000
Number of MCMC chains used - 1
Length of the burnin period used for each chain - 10000
Amount of thinning used - 10

############
#### Results
############
Posterior quantities and DIC

               Mean    2.5%   97.5% n.effective Geweke.diag
(Intercept) -0.1935 -0.2098 -0.1778      9000.0        -0.7
tau2         0.3116  0.2346  0.4027      7283.6         0.7
rho          0.6450  0.4149  0.8752      6040.0        -0.3

DIC =  2159.924       p.d =  232.3109       LMPL =  -1159.96
```

```r
plot_1_2008 <- plot(model2008$samples$rho)
```



The left plot is the traceplot which shows no trend and hence convergence, while the right plot shows a density estimate of the samples. Additionally, these samples show the estimated value of rho (ρ) is close to 0.64, suggesting the spatial dependence in these data after adjusting for the covariates is moderate to high.

```
plot_1_2009 <- plot(model2008$fitted.values)
```




```
model2008$fitted.values
```

```
  [1] 100.96394  16.15972  45.51845  41.10778  62.89696  26.09232  36.09022
  [8]  38.82293  71.40344  32.76934  71.23440  38.19335  38.55330  27.47412
 [15]  40.42565  44.86018  41.48451  89.75543  28.87982  62.12591  53.89398
 [22]  69.90587  31.93220  55.82955  55.91314  25.37622  50.95282  58.24939
 [29]  51.51829  70.72640  33.23912  59.13568 111.26392  25.38550  55.13404
 [36]  63.92143  28.36899  39.25139 101.87497  66.99541  38.91034  56.96673
 [43]  48.68043  50.90533  65.79646 148.23746  53.28233  98.72849 109.95674
 [50]  60.05806  83.20444 103.78599 121.61896  64.20872  30.39433  83.60276
 [57]  46.76814  59.01684  62.47943  82.74923 146.36438  43.43238  62.19696
 [64]  48.51806 100.66910 100.87188  56.78996  49.57533  42.53806  64.50830
 [71]  71.94364  98.18971  45.41408 154.98499  76.38150  83.61168  66.37040
 [78]  53.27491  98.25380  93.31858 134.85805  70.70629 103.13835 152.45737
 [85] 131.54183  68.24317  52.84488  69.55296  57.54774  86.37997 102.77918
 [92] 155.11567  62.35738  98.82174  74.67932 124.39922 163.87461  64.63943
 [99] 143.57959  58.83514  77.38821 112.86926  81.76662 144.83353 134.81889
[106]  69.62230 130.52572  60.49579  50.06234 107.68501  76.86624 127.23832
[113]  41.67196  62.08917  37.17858 111.68601  61.76088 155.23598  45.17116
[120] 169.10323 125.24932 126.37381 141.22217 159.39435  64.39658  80.72739
[127] 148.12477  79.75582  61.76665 202.57656 146.73935  40.58310  85.14655
[134]  91.66104  79.41269  39.89797  67.52654  83.62059  85.49589 101.40656
```

```
[141]  33.30898 100.41162  44.44693  78.77319  40.80699  31.68055  37.12521
[148]  71.63040 112.94739  62.31807  91.40952 103.92132  77.75888  49.72244
[155] 181.40233 198.76362  98.60148  93.96759  67.49487  61.21059 136.69997
[162]  63.77668  81.80477  58.11568 137.86510  59.18559  45.77601  77.94587
[169] 130.00381 107.67763  92.91691  89.34890 141.56051  81.04936  63.49227
[176]  87.50103  58.25692  86.57301  81.07750  82.44492  46.42557  57.91416
[183]  40.96913 105.92689 113.10746 129.19759  75.88836  93.12583  65.63494
[190]  74.38095 127.49097 111.54818  86.86047  94.50318  70.28098  55.34828
[197]  79.56204  78.55655  89.69879  92.97900  47.31551  47.30582  63.88771
[204]  91.27677  70.45179 111.58034 129.64803  52.47952  54.74900 110.55326
[211]  76.60791  50.10401 104.29980 117.90956  46.68916  82.40848  71.78652
[218]  96.64934 142.67075  92.83799  56.64225 134.40896 115.35384  87.68624
[225]  67.78838 158.08434  68.87917  32.58280  78.42296 112.83503  30.91837
[232]  98.31127 126.87797  59.11156  74.40232  81.24895  56.26919 101.36224
[239]  45.18505  66.86558  55.41271  97.13542  75.38847  69.90637  69.98246
[246]  39.71923  87.13746  65.18861  47.34763  65.92692  97.05599  77.80309
[253]  82.21331  82.29401 121.53431 146.88465 130.36568 115.55983  53.75454
[260]  71.60534  83.40150 100.22574  71.36391  50.03276 160.03641  86.16420
[267]  80.07109 100.75451  90.40578  80.29955  64.85726
```

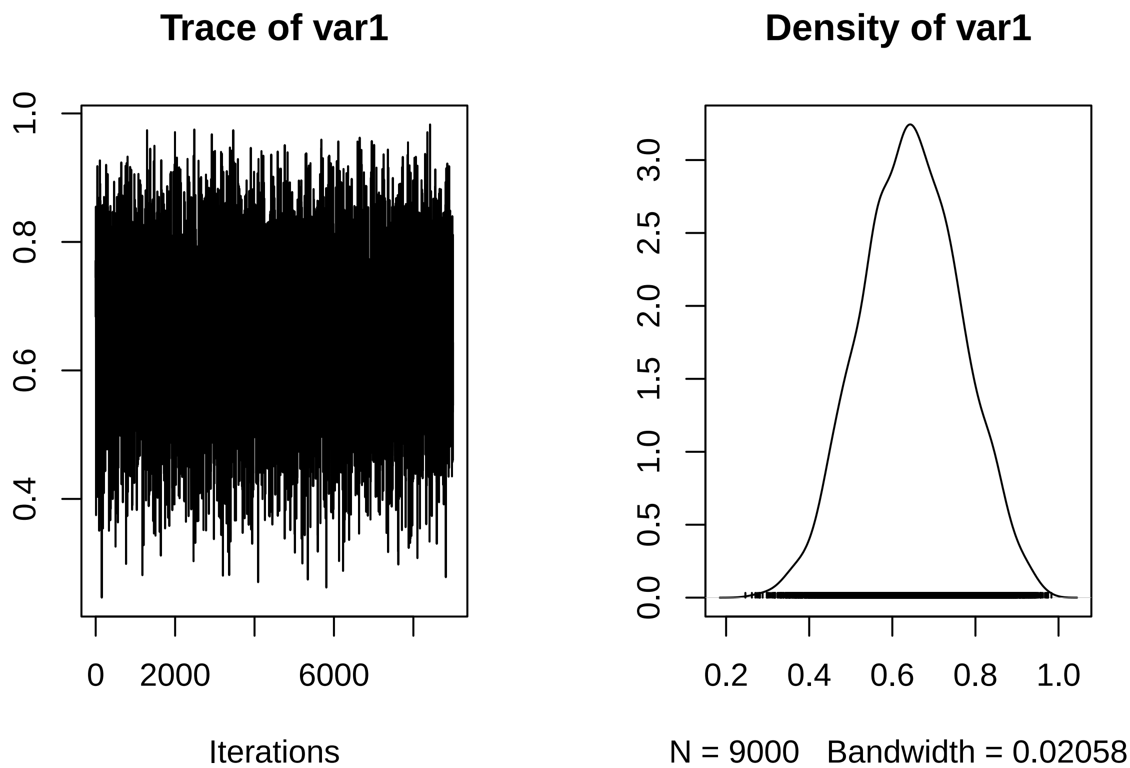
Judging by the plot of SMR2008 mean fitted values, there are no outliers in the data as well as any clear trend.

```r
summary(model2008)
```

```
                    Length Class      Mode     
summary.results      21    -none-     numeric  
samples               6    -none-     list     
fitted.values       271    -none-     numeric  
residuals             2    data.frame list     
modelfit              6    -none-     numeric  
accept                4    -none-     numeric  
localised.structure   0    -none-     NULL     
formula               3    formula    call     
model                 2    -none-     character
mcmc.info             5    -none-     numeric  
X                   271    -none-     numeric  
```

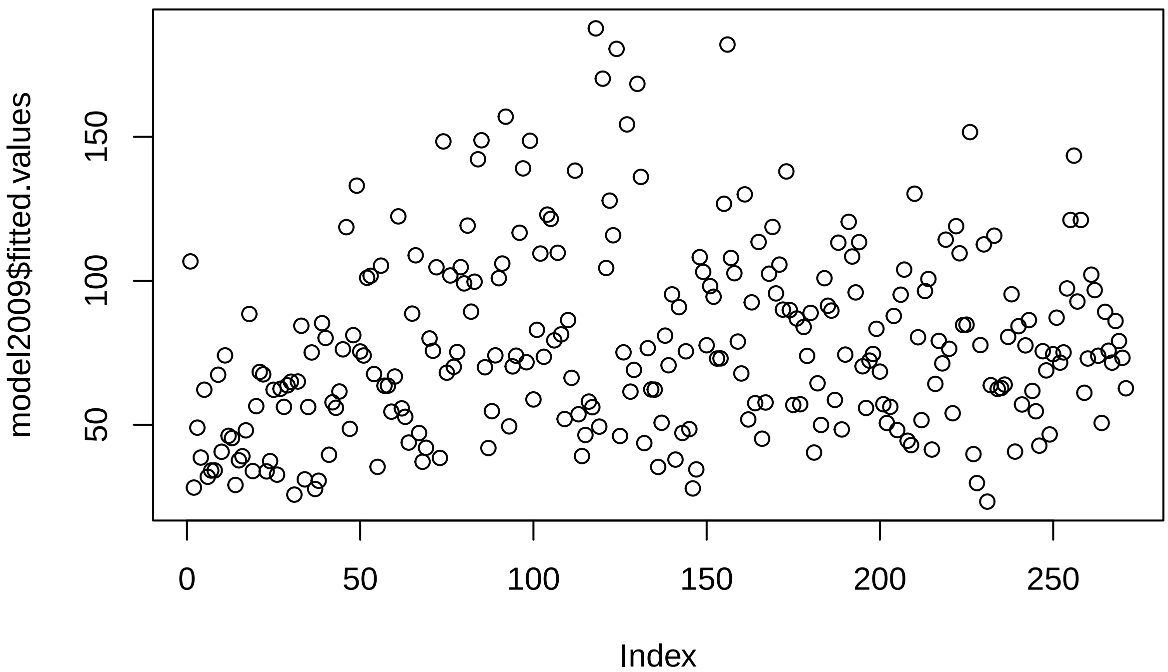
These samples show the estimated value of rho (ρ) is close to 0.64, suggesting the spatial dependence in these data after adjusting for the covariates is moderate to high.

```r
plot(model2009$samples$rho)
```



Judging by the plot of SMR2008 mean fitted values, there are no outliers in the data as well as any clear trend.

```
plot(model2009$fitted.values)
```




```r
summary(model2009)
```

```
                    Length Class      Mode
summary.results      21    -none-     numeric
samples               6    -none-     list
fitted.values       271    -none-     numeric
residuals             2    data.frame list
modelfit              6    -none-     numeric
accept                4    -none-     numeric
localised.structure   0    -none-     NULL
formula               3    formula    call
model                 2    -none-     character
mcmc.info             5    -none-     numeric
X                   271    -none-     numeric
```

Assessing goodness of fit

```r
moran.mc(x = residuals(model2008, type="pearson"), listw = W.list, nsim = 10000)
```

```
    Monte-Carlo simulation of Moran I

data:  residuals(model2008, type = "pearson")
```

```
weights: W.list
number of simulations + 1: 10001

statistic = -0.055709, observed rank = 731, p-value = 0.9269
alternative hypothesis: greater
```

```r
moran.mc(x = residuals(model2009, type="pearson"), listw = W.list, nsim = 10000)
```

```
    Monte-Carlo simulation of Moran I

data:  residuals(model2009, type = "pearson")
weights: W.list
number of simulations + 1: 10001

statistic = -0.068763, observed rank = 319, p-value = 0.9681
alternative hypothesis: greater
```

The statistic and accompanying p-value suggest there is no spatial correlation remaining in the residuals from this model, indicating that the spatial CAR model has adequately removed the correlation from the data.

Assessing goodness of fit

```r
moran.mc(x = residuals(model2008, type="pearson"), listw = W.list, nsim = 10000)
```

```
    Monte-Carlo simulation of Moran I

data:  residuals(model2008, type = "pearson")
weights: W.list
number of simulations + 1: 10001

statistic = -0.055709, observed rank = 715, p-value = 0.9285
alternative hypothesis: greater
```

```r
moran.mc(x = residuals(model2009, type="pearson"), listw = W.list, nsim = 10000)
```

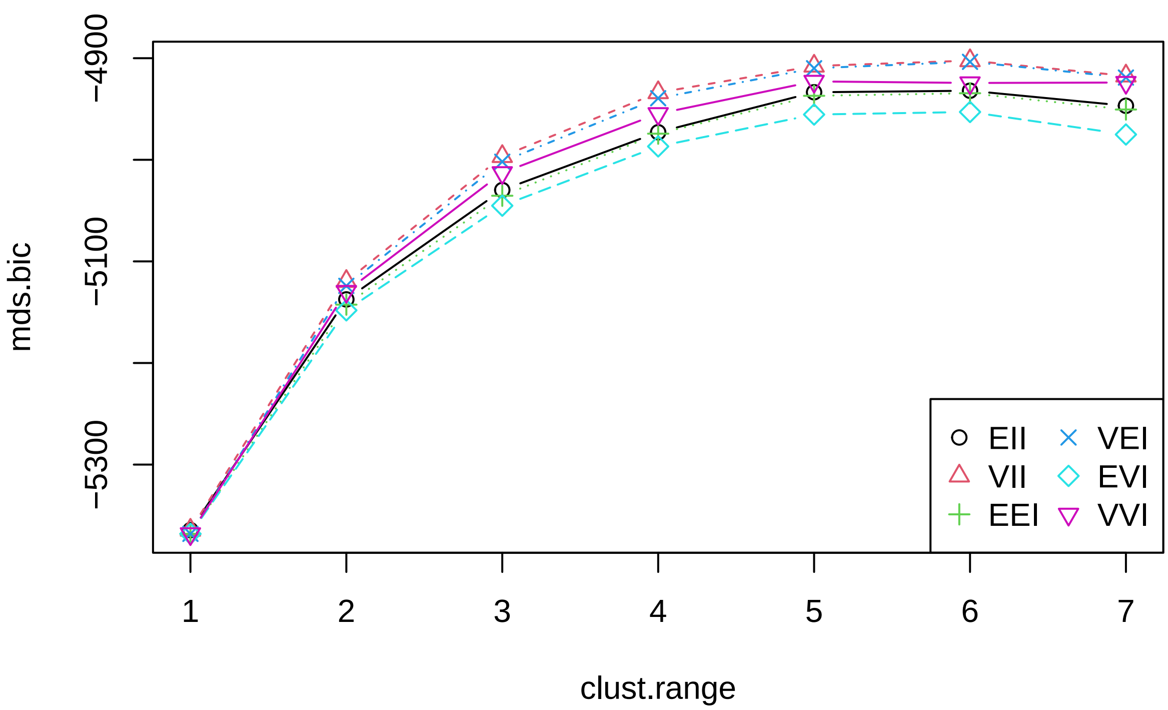
```
    Monte-Carlo simulation of Moran I

data:  residuals(model2009, type = "pearson")
weights: W.list
number of simulations + 1: 10001

statistic = -0.068763, observed rank = 342, p-value = 0.9658
alternative hypothesis: greater
```

The statistic and accompanying p-value suggest there is no spatial correlation remaining in the residuals from this model, indicating that the spatial CAR model has adequately removed the correlation from the data.

3. bivariate mixture model

We plot the BIC values for the range of number of clusters with a different line for each variance parameterisation.

```
# Plot the lines
matplot(clust.range, mds.bic, type = "b", pch = 1:6, col = 1:6)
# Add legend
legend("bottomright", legend = c("EII", "VII", "EEI", "VEI", "EVI", "VVI"), col = 1:6, p
```



The highest BIC is for “VII” model with 5 clusters. From now on, we will analise mentioned model.

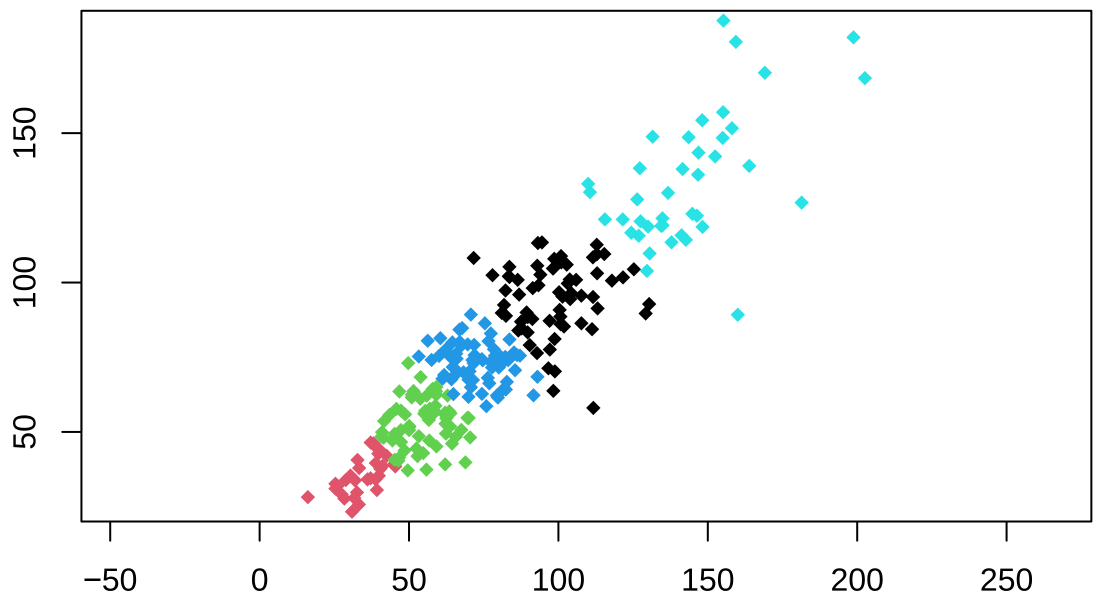
Now, we fit the model with the highest BIC

```
#
md.min = clustMD(fitted_values, G = 5, CnsIndx = 2, OrdIndx = 2, Nnorms = 100, MaxIter =
```

| |

We plot the data with the clusters labelled by colour.

```
library(MASS)
eqscplot(fitted_values[,1:2], col = md.min$cl, pch = 18)
```



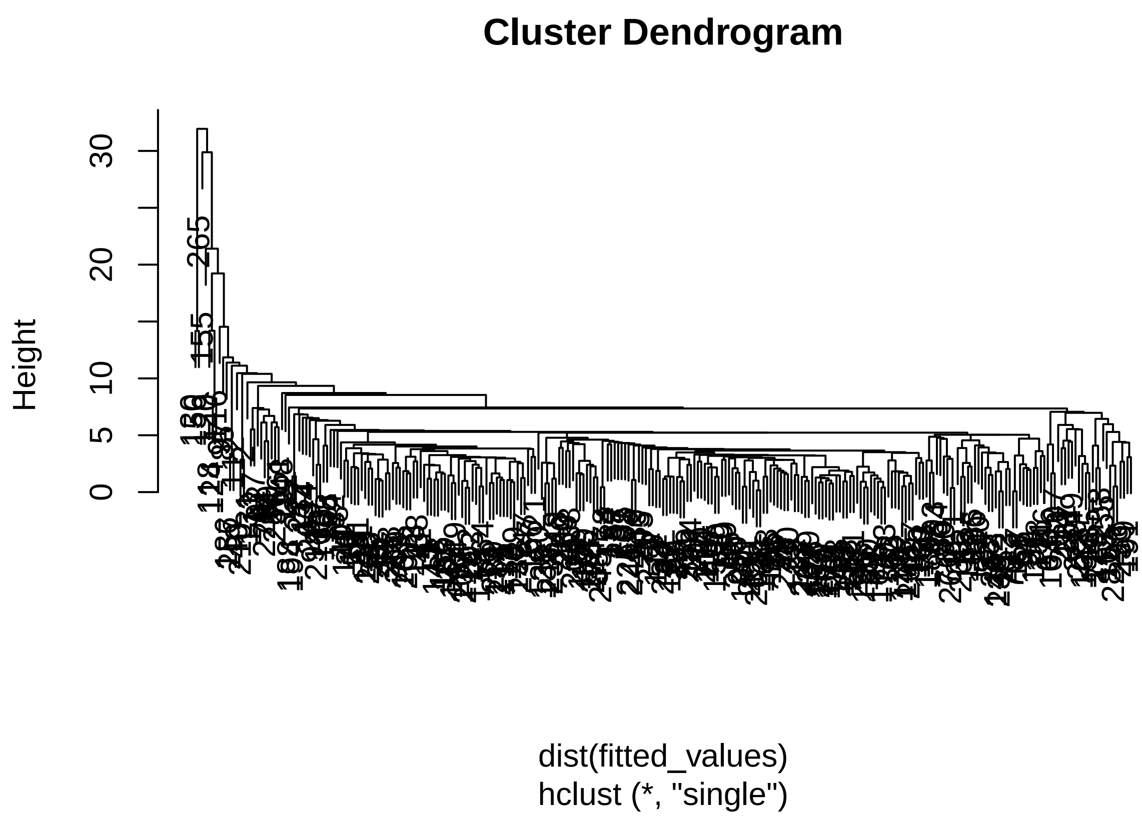
Now, we calculate Average Silhouette Width for the aforementioned model.

```
d1<-dist(fitted_values)
si1<-silhouette(md.min$cl,d1)
#si1
ave.silh1<-mean(si1[,3])
ave.silh1
```

```
[1] 0.407252
```

4. k-means algorithm

```
#Check for outliers using single linkage
single.res <- hclust(dist(fitted_values),"single")
plot(single.res)
```

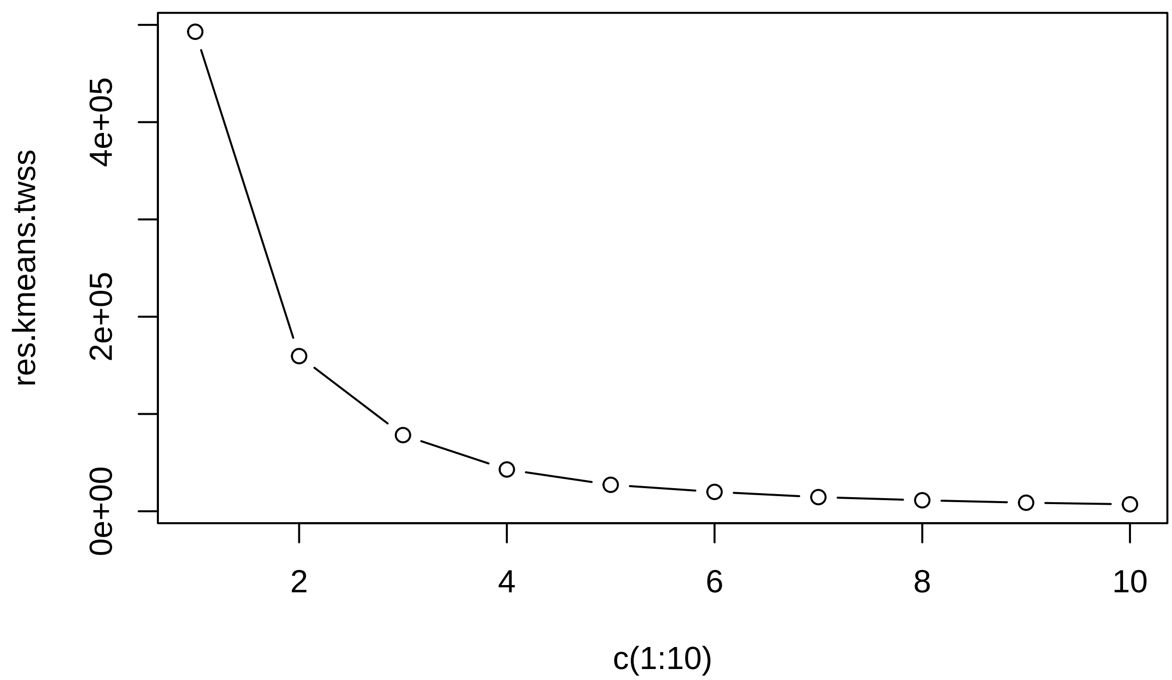



```
#Look if cutting tree identifies singletons joining later
temp<-cutree(single.res,k=5)

table(temp)
```

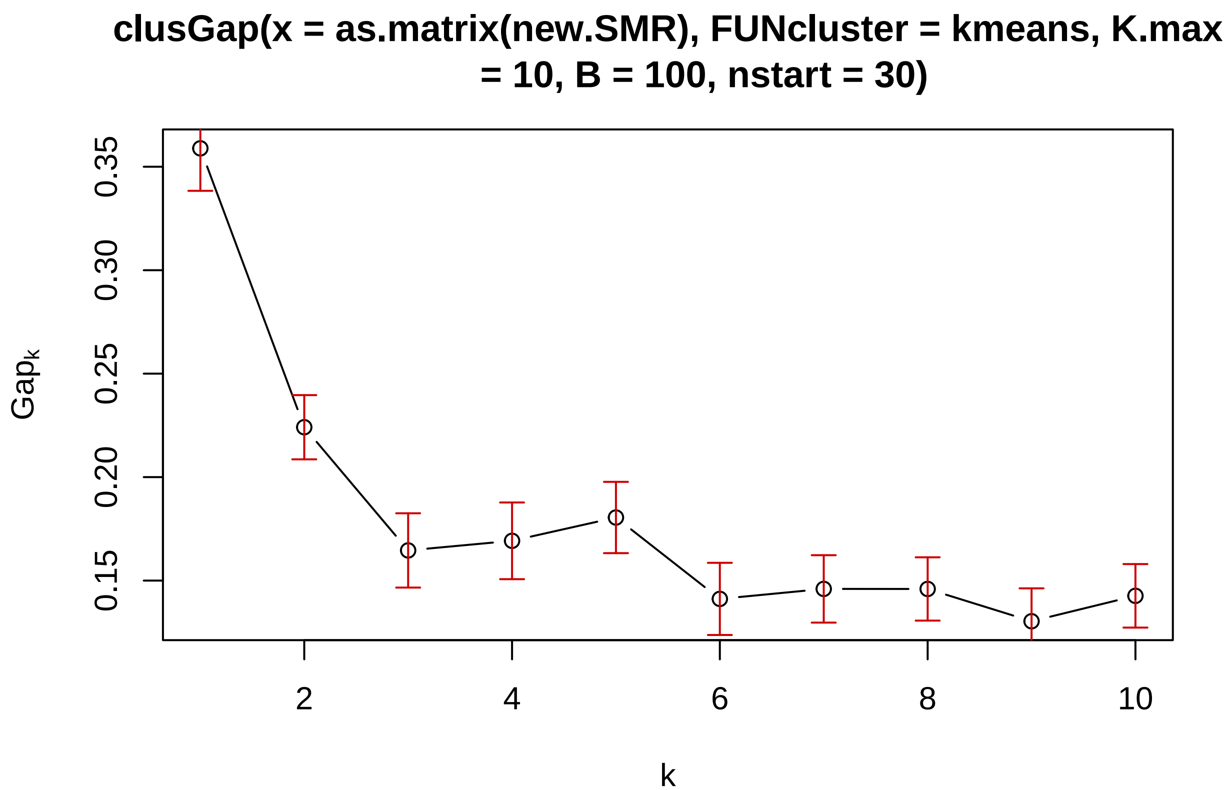
```
temp
  1   2   3   4   5
264   3   2   1   1
```

```
#Remove the four singleton outliers (only keep observations in clusters 1 or 2 from te
new.SMR<-as.matrix(fitted_values[temp==1])


#Run k-means for k from 2 to 10 and record the total within cluster sums of squares fo
res.kmeans.twss<-rep(NA,10)
n<-length(new.SMR)
res.kmeans.twss[1]<-sum((n-1)*apply(new.SMR,2,var))

for(i in 2:10)
{
  res.kmeans.twss[i]<-kmeans(new.SMR,centers=i, nstart=30)$tot.withinss
}
```

```
# Plot the elbow graph
plot(c(1:10),res.kmeans.twss,type="b")
```




```
# Calculate and plot the gap statistic
gap.kmeans<-clusGap(as.matrix(new.SMR), FUN=kmeans, nstart=30,K.max=10,B=100)
plot(gap.kmeans)
```




```
#We see that the bending in the elbow plot is for k=2, so it might suggest that 2 clus


# Calculate the average silhouette width for each k and find the best k
ave.silh<-rep(NA,120)
d<-dist(new.SMR)
length(new.SMR)
```

```
[1] 528
```

```
for(i in 2:120)
{
  res.kmeans<-kmeans(new.SMR,centers=i, nstart=100)
  si<-silhouette(res.kmeans$cluster,d)
  ave.silh[i]<-mean(si[,3])
}
ave.silh
```

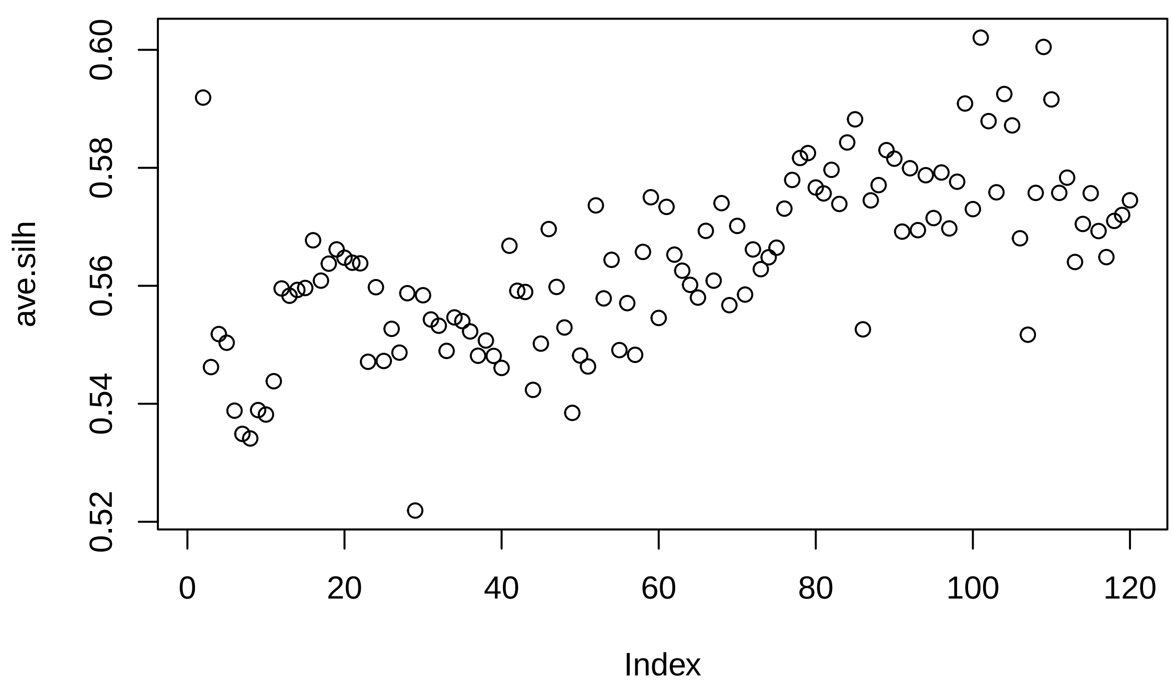
```
  [1]        NA 0.5919050 0.5462232 0.5518193 0.5503496 0.5388323 0.5348971
  [8] 0.5341202 0.5389289 0.5381753 0.5438271 0.5595491 0.5583173 0.5593022
 [15] 0.5596280 0.5677141 0.5608752 0.5637529 0.5661723 0.5647615 0.5639168
 [22] 0.5638055 0.5471222 0.5597647 0.5472607 0.5527043 0.5486713 0.5587476
 [29] 0.5219031 0.5583986 0.5542896 0.5532321 0.5489649 0.5546407 0.5540140
```

```
 [36] 0.5522628 0.5481425 0.5507221 0.5481090 0.5460772 0.5667891 0.5591611
 [43] 0.5589564 0.5423580 0.5501995 0.5696185 0.5598110 0.5529338 0.5384589
 [50] 0.5481751 0.5463218 0.5736306 0.5578642 0.5644010 0.5491013 0.5570599
 [57] 0.5483006 0.5657450 0.5750139 0.5545471 0.5733773 0.5652849 0.5625612
 [64] 0.5601644 0.5579976 0.5693137 0.5608734 0.5740154 0.5567200 0.5701525
 [71] 0.5585125 0.5661672 0.5628196 0.5648354 0.5664583 0.5730820 0.5779462
 [78] 0.5816641 0.5824989 0.5766629 0.5756641 0.5796613 0.5738731 0.5842974
 [85] 0.5882192 0.5526161 0.5744794 0.5770712 0.5829949 0.5815467 0.5691856
 [92] 0.5799245 0.5694351 0.5787465 0.5714840 0.5792063 0.5697255 0.5776414
 [99] 0.5908996 0.5730001 0.6020656 0.5879148 0.5758483 0.5925124 0.5871897
[106] 0.5680568 0.5517096 0.5757475 0.6004894 0.5915937 0.5757529 0.5783344
[113] 0.5640345 0.5704818 0.5756922 0.5692824 0.5648649 0.5710012 0.5720063
[120] 0.5744947
```

```r
ave.silh[2]
```

```
[1] 0.591905
```

```r
plot(ave.silh)
```




```r
#The highest value for ASW is for k=3, suggesting that 3-means is the best fit.

#Compare the 2 cluster k-means solution on the SMR data to the 3 clusters solution
```

```r
kmeans.2<-kmeans(new.SMR,2,nstart=30)
kmeans.3<-kmeans(new.SMR,3,nstart=30)

#Take a look at the 2-cluster k-medoids clustering and compare it too
library(cluster)
pam.2<-pam(new.SMR,2)
#plot(pam.2)

# Calculate the average silhouette width for each k and find the best k
ave.silh2<-rep(NA,30)
d<-dist(new.SMR)
length(new.SMR)
```

```
[1] 528
```

```r
si2<-silhouette(pam.2$clust,d)
ave.silh2<-mean(si2[,3])

ave.silh2
```

```
[1] 0.5735949
```

```r
#cl1<-c(rep(1,109),rep(2,108))
#cl2<-c(rep(1,55),rep(2,54),rep(3,54),rep(4,54))
```

Comparing the results within different algorithms.

```r
#ASW for k-means with 3 clusters
ave.silh[2]
```

```
[1] 0.591905
```

```r
#ASW for k-medoids with 3 clusters
ave.silh2
```

```
[1] 0.5735949
```

```r
#ASW for Mixture Clustering
ave.silh1
```

```
[1] 0.407252
```

```r
c(ave.silh[2],ave.silh2,ave.silh1)
```

```
[1] 0.5919050 0.5735949 0.4072520
```

We see that Average Silhouette Width for 3-means algorithm is the highest, so it is the best algorithm for clustering the mean fitted values SMR data.