Clustering

Jimmy Harvin

Importing data set, replacing instances of '?' with NA

```
pos <- read.csv("postures.csv", na.strings = c("?"), header=FALSE)
head(pos)
```

```
##      V1   V2          V3          V4           V5          V6          V7
## 1 Class User          X0          Y0           Z0          X1          Y1
## 2     0    0           0           0            0           0           0
## 3     1    0 54.26387995 71.46677614 -64.80770878 76.89563478  42.4624999
## 4     1    0 56.52755845 72.26660863 -61.93525242 39.13597811 82.53852991
## 5     1    0 55.84992755 72.46906399  -62.5627882 37.98880398 82.63134669
## 6     1    0 55.32964736 71.70727473 -63.68895606 36.56186258 81.86874878
##             V8          V9         V10          V11         V12         V13
## 1           Z1          X2          Y2           Z2          X3          Y3
## 2            0           0           0            0           0           0
## 3 -72.78054519 36.62122916 81.68055693 -52.91927237 85.23226389  67.7492195
## 4  -49.5965094 79.22374278 43.25409085 -69.98248905 87.45087295  68.4008083
## 5 -50.60625925 78.45152642 43.56740255 -70.65848926 86.83538757 68.90792498
## 6  -52.7527843 86.32062994 68.21464501 -72.22846087 61.59615713 11.25064818
##            V14         V15         V16          V17  V18  V19  V20  V21  V22
## 1           Z3          X4          Y4           Z4   X5   Y5   Z5   X6   Y6
## 2            0           0           0            0    0    0    0    0    0
## 3 -73.68413004  59.1885757 10.67893641 -71.29778131 <NA> <NA> <NA> <NA> <NA>
## 4 -70.70399093 61.58745155 11.77991903 -68.82741776 <NA> <NA> <NA> <NA> <NA>
## 5 -71.13834414 61.68642719 11.79343989 -68.88931646 <NA> <NA> <NA> <NA> <NA>
## 6 -68.95642523 77.38722541 42.71783348  -72.0151463 <NA> <NA> <NA> <NA> <NA>
##    V23  V24  V25  V26  V27  V28  V29  V30  V31  V32  V33  V34  V35  V36  V37
## 1   Z6   X7   Y7   Z7   X8   Y8   Z8   X9   Y9   Z9  X10  Y10  Z10  X11  Y11
## 2    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
## 3 <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA>
## 4 <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA>
## 5 <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA>
## 6 <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA>
##    V38
## 1  Z11
## 2    0
## 3 <NA>
## 4 <NA>
## 5 <NA>
## 6 <NA>
```

##Data Cleaning

Removing dummy row, only keeping first 5 coordinate points for each row Removing rows with NA coordinates, coercing values from strings to numbers

```
i <- c(1:17)
pos <- pos[-c(1), i]
pos <- na.omit(pos)
pos[ , i] <- apply(pos[ , i], 2, function(j) as.numeric(as.character(j)))

set.seed(1477)
```

##K-Means Clustering

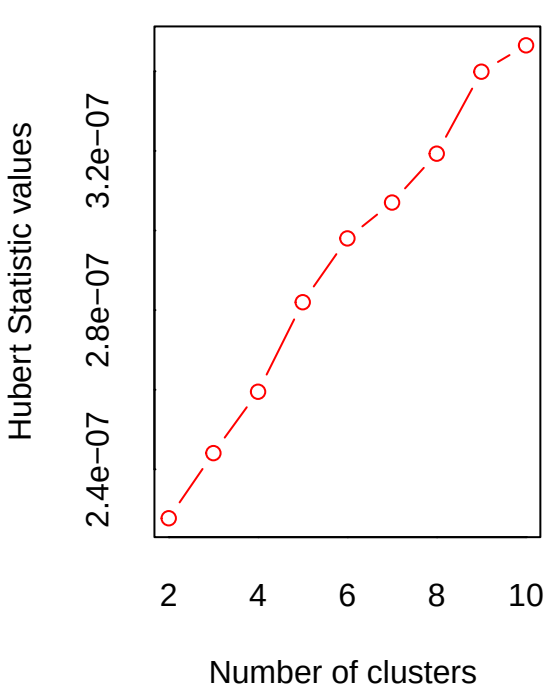
Using NbClust to find optimum number of clusters Because this data set has labels, I know that there are 5 target positions (and 14 different users), but I wanted to see if I would have come to that conclusion on my own

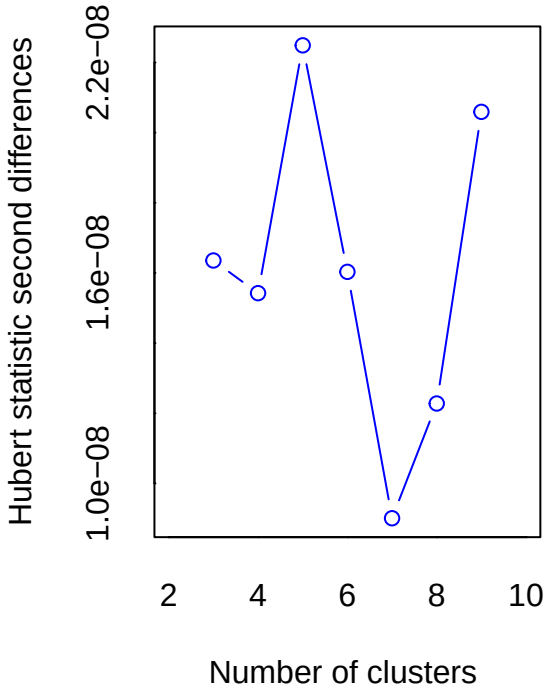
With the given subset of data, 5 clusters was not recommended a single time, which is hilariously concerning

```
library(NbClust)

subset <- pos[sample(nrow(pos), 0.01*nrow(pos), replace = FALSE), ]
subset$V1
```

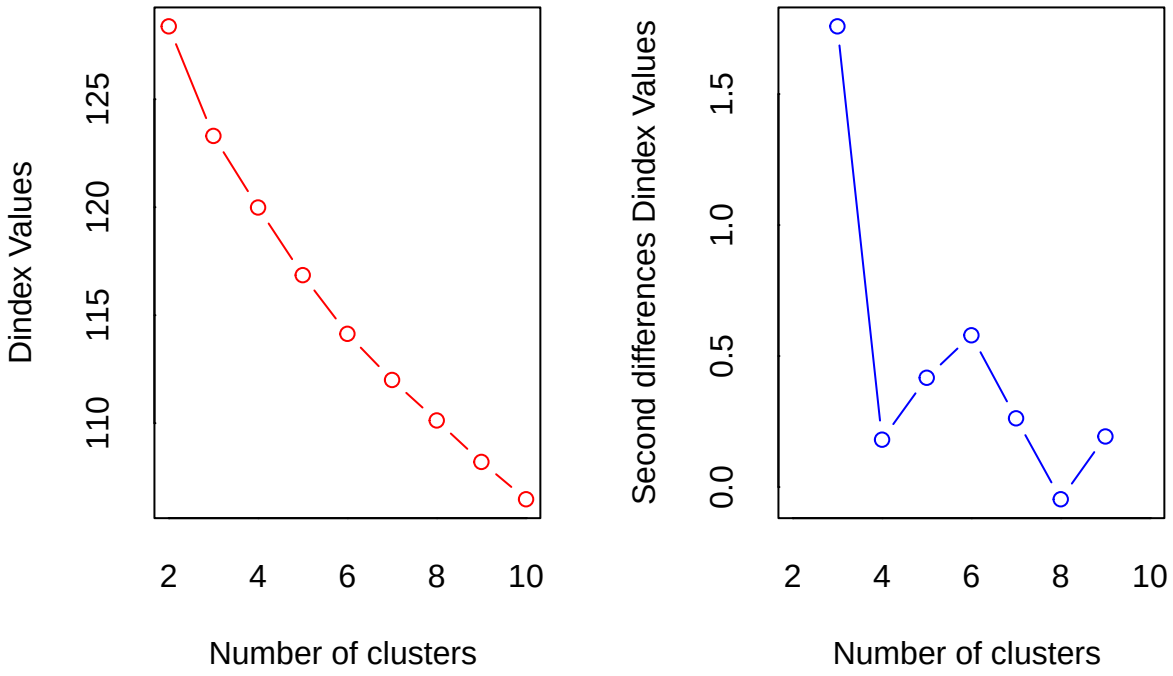
```
##   [1] 1 5 5 1 2 5 3 1 1 4 2 4 5 4 3 3 5 3 3 4 2 3 3 3 2 3 2 3 2 1 1 5 5 4 4 3 1
##  [38] 1 3 3 3 4 3 1 2 2 3 2 2 5 2 5 5 4 5 3 3 2 3 4 4 2 2 1 3 1 5 2 3 3 5 1 3 4
##  [75] 2 5 5 4 1 2 3 3 3 5 4 4 2 3 4 4 5 5 1 4 2 2 4 2 3 5 1 4 2 2 4 4 2 3 4 3 4
## [112] 1 5 5 1 1 4 4 2 1 4 5 2 3 3 3 1 5 5 4 5 5 3 3 3 2 3 1 3 4 1 3 1 4 5 5 5 2
## [149] 2 3 5 5 3 3 3 5 4 5 4 1 2 3 2 3 4 3 3 5 2 5 5 1 3 1 1 2 3 5 1 4 3 4 5 5 1
## [186] 3 1 5 4 5 4 5 4 2 1 5 4 2 5 1 3 2 4 5 1 2 5 5 3 3 3 3 2 4 1 4 2 4 5 3 1 1
## [223] 4 3 1 2 4 2 2 1 2 2 1 4 2 1 3 1 3 4 4 5 1 2 3 2 5 2 4 1 5 3 1 5 1 4 5 1 2
## [260] 3 4 4 1 5 2 3 2 3 1 4 2 1 1 3 2 3 3 2 4 2 5 5 3 1 5 1 2 1 3 4 3 1 5 2 3 3
## [297] 3 3 5 3 1 4 2 3 2 1 3 3 1 5 3 1 4 2 3 2 3 3 1 2 4 5 3 4 4 3 3 3 1 4 4 5 5
## [334] 4 2 1 3 3 3 5 1 5 5 4 1 3 5 3 1 2 2 5 1 1 5 2 4 1 2 1 2 2 4 4 3 5 5 1 5 5
## [371] 1 1 1 3 2 1 3 3 2 3 1 1 4 4 5 2 3 1 3 5 2 2 4 3 2 2 2 1 1 1 1 2 5 4 5 5 5
## [408] 3 2 2 2 4 3 2 2 2 4 4 4 5 1 3 5 4 1 1 5 2 5 5 1 5 1 4 5 5 5 1 3 2 4 2 4 3
## [445] 4 2 2 4 5 4 5 5 5 3 4 3 3 1 1 4 2 2 3 4 2 2 5 3 4 1 1 4 2 2 1 1 5 3 5 4 3
## [482] 1 1 3 4 5 5 4 4 3 3 1 3 4 1 5 3 5 5 3 3 2 2 4 5 5 3 5 3 4 2 1 2 2 4 1 3 3
## [519] 3 4 5 1 4 2 5 5 2 1 4 4 3 5 3 5 2 2 5 5 1 2 3 2 5 2 2 3 2 5 1 4 5 3 1 5 4
## [556] 2 3 3 2 2 2 4 4 2 3 3 2 5 5 2 4 3 3 3 4 3 2 1 5 3 5 2 2 3 3 3 5 3 2 1 1 2
## [593] 3 5 1 5 3 4 1 5 1 4 2 4 2 1 1 4 4 5 5 3 1 2 3 1 4 1 1 1 4 2 4 4 2 3 5 1 4
## [630] 3 5 1 2 2 1 1 3 1 5 1 2 5 2 3 3 5 2 4 4 2 2 5 5 1 4 1 1 2 4 1 2 4 4 3 3 2
## [667] 5 1 5 5 4 4 3 2 4 2 3 2 4 5 3 4 4 3 4 4 4 3 1 3 4 5 3 2 1 1 3 3 5 2 5 3 5
## [704] 3 3 4 4 1 5 3 2 3 1 2 2 4 4 5 4 4 2 4 4 3 5 2 1 4 5 2 5 3 1 4 4 2 3 2 3 3
## [741] 4 2 5 2 3 3 5 4 5
```

```
nc <- NbClust(subset[ , c(3:17)], min.nc = 2, max.nc = 10, method="kmeans")
```






```
## *** : The Hubert index is a graphical method of determining the number of clusters.
##                 In the plot of Hubert index, we seek a significant knee that corresponds to a
##                 significant increase of the value of the measure i.e the significant peak in Hubert
##                 index second differences plot.
##
```



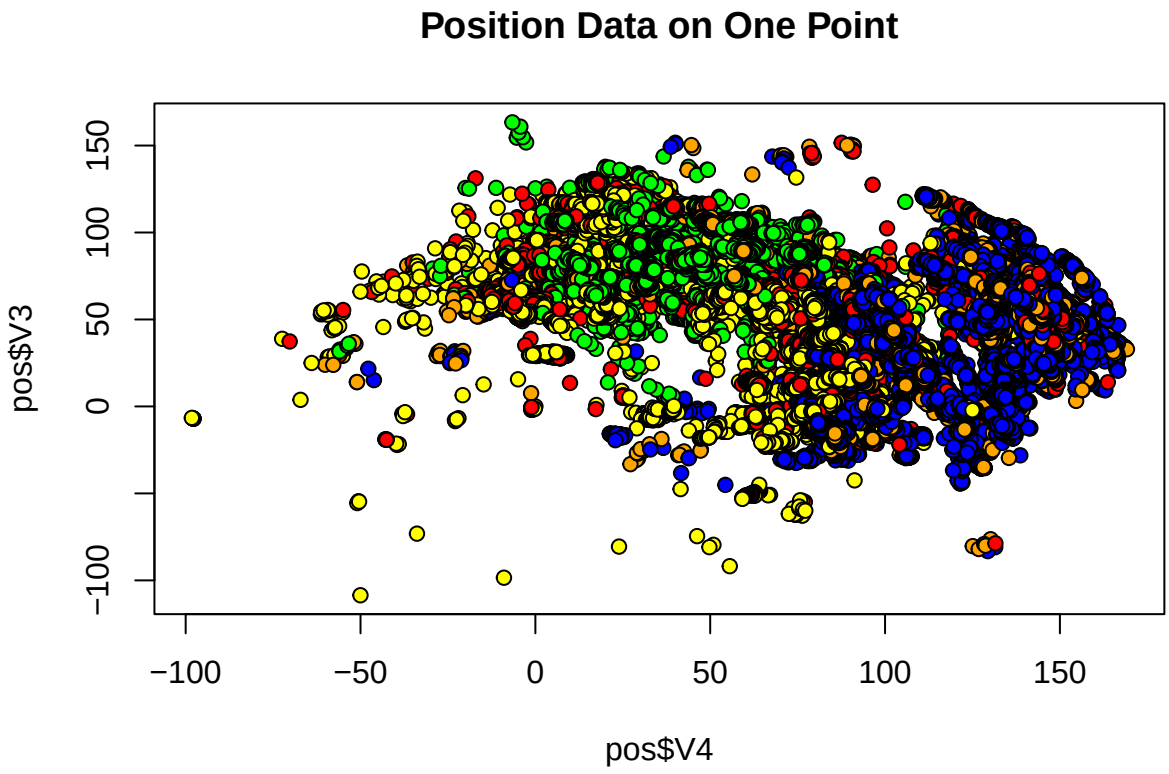

```
## *** : The D index is a graphical method of determining the number of clusters.
##                 In the plot of D index, we seek a significant knee (the significant peak in Dindex
##                 second differences plot) that corresponds to a significant increase of the value of
##                 the measure.
##
## *******************************************************************
## * Among all indices:
## * 10 proposed 2 as the best number of clusters
## * 8 proposed 3 as the best number of clusters
## * 1 proposed 7 as the best number of clusters
## * 4 proposed 10 as the best number of clusters
##
##                     ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  2
##
##
## *******************************************************************
```

Here is the actual clustering; with so many variables, it is virtually impossible to visualize every coordinate on one scatter plot, but the table and follow-up plot generally show how the target class corresponds to the chosen clusters It is important to note that the chosen clusters may not correlate with the class labels, as the starting points were random

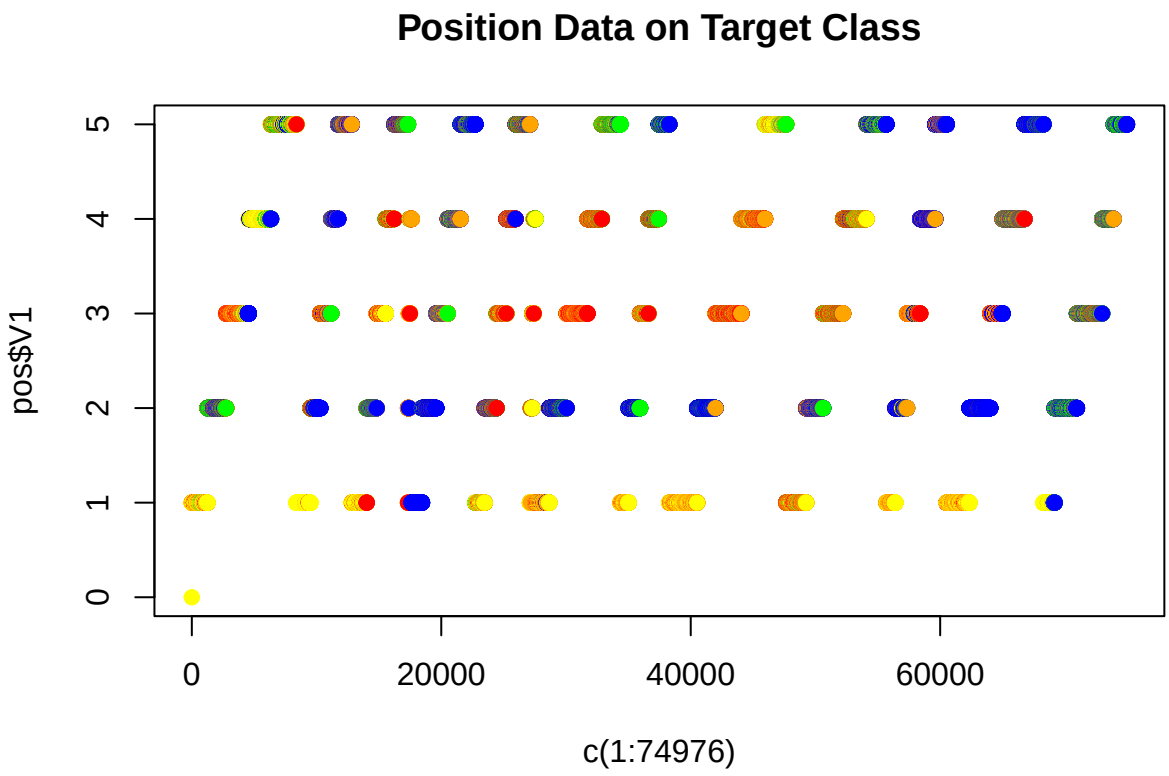
```
posCluster <- kmeans(pos[ , c(3:17)], 5)

table(posCluster$cluster, pos$V1)
```

```
##
##        0    1    2    3    4    5
##   1    0 2880  753 5787 3133 1418
##   2    0  294 2250 1245 1882 3739
##   3    0  909 9422 1531 3012 5802
##   4    1 9727  274 3121 2900 2419
##   5    0  380 2251 3696 3842 2308
```

```
plot(pos$V4, pos$V3, pch = 21, bg = c("red", "green", "blue", "yellow", "orange")[unclass(posCluster$clu
```




```
plot(c(1:74976), pos$V1, pch = 19, col = c("red", "green", "blue", "yellow", "orange")[unclass(posCluste
```




```
wss <- sum(posCluster$withinss)
print(paste("WSS: ", wss))
```

```
## [1] "WSS:  1063880635.32732"
```

Based on the table, positions 1, 2, and 3 were placed fairly well into unique clusters, but positions 5 and especially 4 are experiencing improper classifications; positions 4 and 5 are likely the cause behind the staggering WSS

##Hierarchical Clustering

No need to scale the data, all coordinates use the same units To flatten the data, I took all of the rows from one of the users so that I didn't have to worry about inconsistencies in posture and hand size from randomly sampling all users

```
library(flexclust)
```

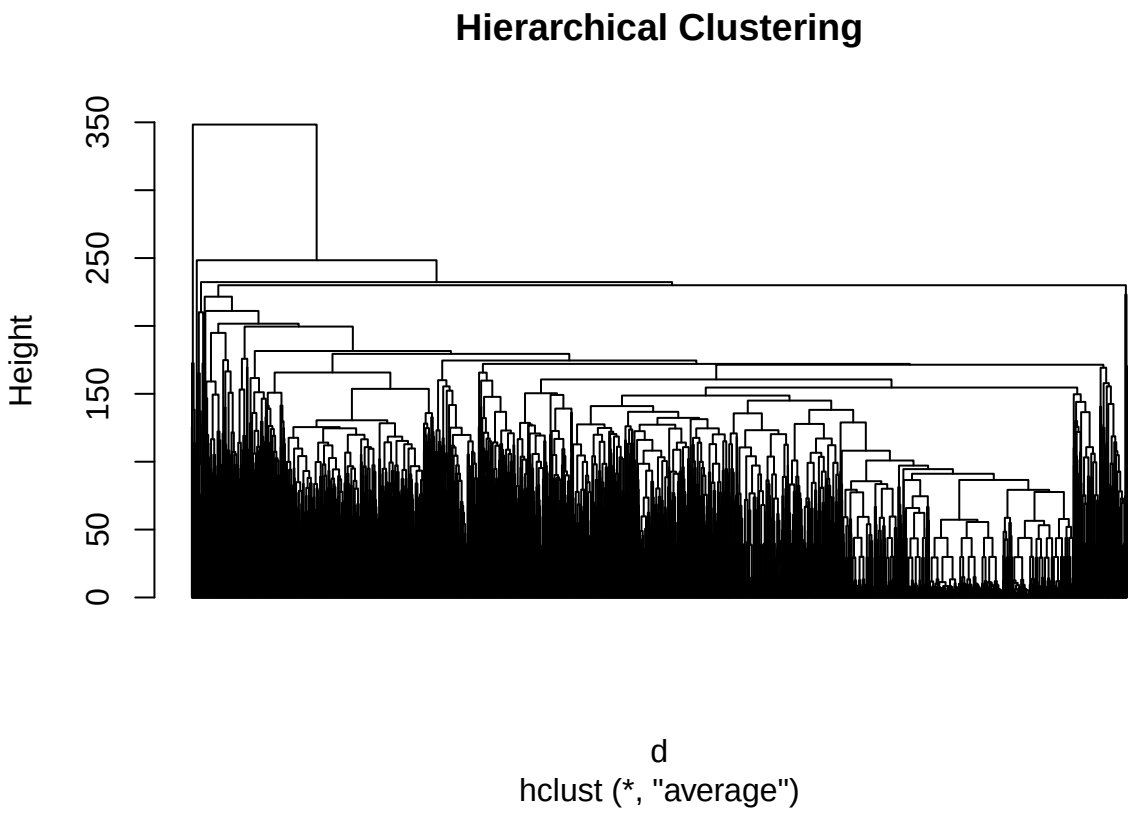
```
## Loading required package: grid
```

```
## Loading required package: lattice
```

```
## Loading required package: modeltools
```

```
## Loading required package: stats4
```

```
subset <- subset(pos, pos$V2 == 1)
d <- dist(subset[, c(3:17)])
fit.average <- hclust(d, method = "average")
plot(fit.average, hang = -1, cex = 0.01, main= "Hierarchical Clustering")
```




```
for (c in 2:20){
  cluster_cut <- cutree(fit.average, c)
  table_cut <- table(cluster_cut, subset$V1)
  print(table_cut)
  ri <- randIndex(table_cut)
  print(paste("cut=", c, "Rand index = ", ri))
}
```

```
##
## cluster_cut    1    2    3    4    5
##           1 1115  791  857  561 1085
##           2    0    0   21    0    0
## [1] "cut= 2 Rand index =  0.000384704988283338"
##
## cluster_cut    1    2    3    4    5
##           1 1109  791  857  561 1085
##           2    6    0    0    0    0
##           3    0    0   21    0    0
## [1] "cut= 3 Rand index =  9.97630230431115e-05"
##
```

```
## cluster_cut    1    2    3    4    5
##           1 1109  791  834  561 1075
##           2    6    0    0    0    0
##           3    0    0   21    0    0
##           4    0    0   23    0   10
## [1] "cut= 4 Rand index =  0.000207154369221969"
##
## cluster_cut    1    2    3    4    5
##           1 1100  791  834  561 1074
##           2    9    0    0    0    1
##           3    6    0    0    0    0
##           4    0    0   21    0    0
##           5    0    0   23    0   10
## [1] "cut= 5 Rand index =  -0.000268429954522062"
##
## cluster_cut    1    2    3    4    5
##           1 1100  791  834  561 1074
##           2    6    0    0    0    0
##           3    3    0    0    0    1
##           4    6    0    0    0    0
##           5    0    0   21    0    0
##           6    0    0   23    0   10
## [1] "cut= 6 Rand index =  -0.000271852752696005"
##
## cluster_cut    1    2    3    4    5
##           1 1100  789  834  561 1072
##           2    6    0    0    0    0
##           3    3    0    0    0    1
##           4    6    0    0    0    0
##           5    0    2    0    0    2
##           6    0    0   21    0    0
##           7    0    0   23    0   10
## [1] "cut= 7 Rand index =  -0.00028722809895563"
##
## cluster_cut    1    2    3    4    5
##           1 1100  788  834  561 1072
##           2    6    0    0    0    0
##           3    3    0    0    0    1
##           4    6    0    0    0    0
##           5    0    2    0    0    2
##           6    0    1    0    0    0
##           7    0    0   21    0    0
##           8    0    0   23    0   10
## [1] "cut= 8 Rand index =  -0.000253416723604385"
##
## cluster_cut    1    2    3    4    5
##           1 1100  788  834  561 1072
##           2    6    0    0    0    0
##           3    3    0    0    0    1
##           4    6    0    0    0    0
##           5    0    2    0    0    2
##           6    0    1    0    0    0
##           7    0    0   21    0    0
##           8    0    0   23    0    0
```

```
##           9    0    0    0    0   10
## [1] "cut= 9 Rand index =  -0.000240615049790162"
##
## cluster_cut    1    2    3    4    5
##           1  1100  678  806  555 1064
##           2     6    0    0    0    0
##           3     3    0    0    0    1
##           4     6    0    0    0    0
##           5     0  110   28    6    8
##           6     0    2    0    0    2
##           7     0    1    0    0    0
##           8     0    0   21    0    0
##           9     0    0   23    0    0
##           10    0    0    0    0   10
## [1] "cut= 10 Rand index =  0.00681571278050641"
##
## cluster_cut    1    2    3    4    5
##           1  1100  634  806  553 1057
##           2     6    0    0    0    0
##           3     3    0    0    0    1
##           4     6    0    0    0    0
##           5     0  110   28    6    8
##           6     0   44    0    2    7
##           7     0    2    0    0    2
##           8     0    1    0    0    0
##           9     0    0   21    0    0
##           10    0    0   23    0    0
##           11    0    0    0    0   10
## [1] "cut= 11 Rand index =  0.00973450022337375"
##
## cluster_cut    1    2    3    4    5
##           1  1100  634  806  553 1057
##           2     6    0    0    0    0
##           3     3    0    0    0    1
##           4     6    0    0    0    0
##           5     0   61    5    2    5
##           6     0   44    0    2    7
##           7     0   49   23    4    3
##           8     0    2    0    0    2
##           9     0    1    0    0    0
##           10    0    0   21    0    0
##           11    0    0   23    0    0
##           12    0    0    0    0   10
## [1] "cut= 12 Rand index =  0.00919657162248214"
##
## cluster_cut    1    2    3    4    5
##           1  1100  603  799  552 1011
##           2     6    0    0    0    0
##           3     3    0    0    0    1
##           4     6    0    0    0    0
##           5     0   61    5    2    5
##           6     0   31    7    1   46
##           7     0   44    0    2    7
##           8     0   49   23    4    3
```

```
##          9     0    2    0    0    2
##          10    0    1    0    0    0
##          11    0    0   21    0    0
##          12    0    0   23    0    0
##          13    0    0    0    0   10
## [1] "cut= 13 Rand index =  0.0098559454397277"
##
## cluster_cut    1    2    3    4    5
##          1  1100  203  786  251  928
##          2     6    0    0    0    0
##          3     3    0    0    0    1
##          4     6    0    0    0    0
##          5     0   61    5    2    5
##          6     0   31    7    1   46
##          7     0   44    0    2    7
##          8     0   49   23    4    3
##          9     0    2    0    0    2
##          10    0  400   13  301   83
##          11    0    1    0    0    0
##          12    0    0   21    0    0
##          13    0    0   23    0    0
##          14    0    0    0    0   10
## [1] "cut= 14 Rand index =  0.126324729137553"
##
## cluster_cut    1    2    3    4    5
##          1  1100  203  786  251  928
##          2     6    0    0    0    0
##          3     3    0    0    0    1
##          4     6    0    0    0    0
##          5     0   61    5    2    5
##          6     0   31    7    1   46
##          7     0   44    0    2    7
##          8     0   49   23    4    3
##          9     0    2    0    0    1
##          10    0  400   13  301   83
##          11    0    1    0    0    0
##          12    0    0   21    0    0
##          13    0    0   23    0    0
##          14    0    0    0    0   10
##          15    0    0    0    0    1
## [1] "cut= 15 Rand index =  0.126324632187453"
##
## cluster_cut    1    2    3    4    5
##          1  1100  203  786  251  928
##          2     6    0    0    0    0
##          3     3    0    0    0    1
##          4     6    0    0    0    0
##          5     0   61    5    2    5
##          6     0   31    7    1   46
##          7     0   36    0    2    6
##          8     0   49   23    4    3
##          9     0    2    0    0    1
##          10    0    8    0    0    1
##          11    0  400   13  301   83
```

```
##          12    0    1    0    0    0
##          13    0    0   21    0    0
##          14    0    0   23    0    0
##          15    0    0    0    0   10
##          16    0    0    0    0    1
## [1] "cut= 16 Rand index =  0.126251228783923"
##
## cluster_cut    1    2    3    4    5
##          1  1100  203  786  251  928
##          2     6    0    0    0    0
##          3     3    0    0    0    1
##          4     6    0    0    0    0
##          5     0   49    5    2    5
##          6     0   31    7    1   46
##          7     0   36    0    2    6
##          8     0   49   23    4    3
##          9     0   12    0    0    0
##          10    0    2    0    0    1
##          11    0    8    0    0    1
##          12    0  400   13  301   83
##          13    0    1    0    0    0
##          14    0    0   21    0    0
##          15    0    0   23    0    0
##          16    0    0    0    0   10
##          17    0    0    0    0    1
## [1] "cut= 17 Rand index =  0.12609886100234"
##
## cluster_cut    1    2    3    4    5
##          1  1100  201  786  218  756
##          2     6    0    0    0    0
##          3     3    0    0    0    1
##          4     6    0    0    0    0
##          5     0   49    5    2    5
##          6     0   31    7    1   46
##          7     0   36    0    2    6
##          8     0   49   23    4    3
##          9     0    2    0   33  172
##          10    0   12    0    0    0
##          11    0    2    0    0    1
##          12    0    8    0    0    1
##          13    0  400   13  301   83
##          14    0    1    0    0    0
##          15    0    0   21    0    0
##          16    0    0   23    0    0
##          17    0    0    0    0   10
##          18    0    0    0    0    1
## [1] "cut= 18 Rand index =  0.133652149806333"
##
## cluster_cut    1    2    3    4    5
##          1  1100  201  786  218  756
##          2     6    0    0    0    0
##          3     3    0    0    0    1
##          4     6    0    0    0    0
##          5     0   49    5    2    5
```

```
##          6     0   31    7    1   46
##          7     0   36    0    2    6
##          8     0   49   23    4    3
##          9     0    2    0   33  172
##          10    0   12    0    0    0
##          11    0    2    0    0    1
##          12    0    8    0    0    1
##          13    0  400   13  301   83
##          14    0    1    0    0    0
##          15    0    0    3    0    0
##          16    0    0   18    0    0
##          17    0    0   23    0    0
##          18    0    0    0    0   10
##          19    0    0    0    0    1
## [1] "cut= 19 Rand index =  0.133635859798033"
##
## cluster_cut    1    2    3    4    5
##          1  1100  201  786  218  756
##          2     6    0    0    0    0
##          3     3    0    0    0    1
##          4     6    0    0    0    0
##          5     0   49    5    2    5
##          6     0   31    7    1   46
##          7     0   36    0    2    6
##          8     0   49   23    4    3
##          9     0    2    0   33  172
##          10    0   12    0    0    0
##          11    0    2    0    0    1
##          12    0    8    0    0    1
##          13    0  400   13  301   83
##          14    0    1    0    0    0
##          15    0    0    3    0    0
##          16    0    0   18    0    0
##          17    0    0   23    0    0
##          18    0    0    0    0    3
##          19    0    0    0    0    7
##          20    0    0    0    0    1
## [1] "cut= 20 Rand index =  0.133629524739521"
```

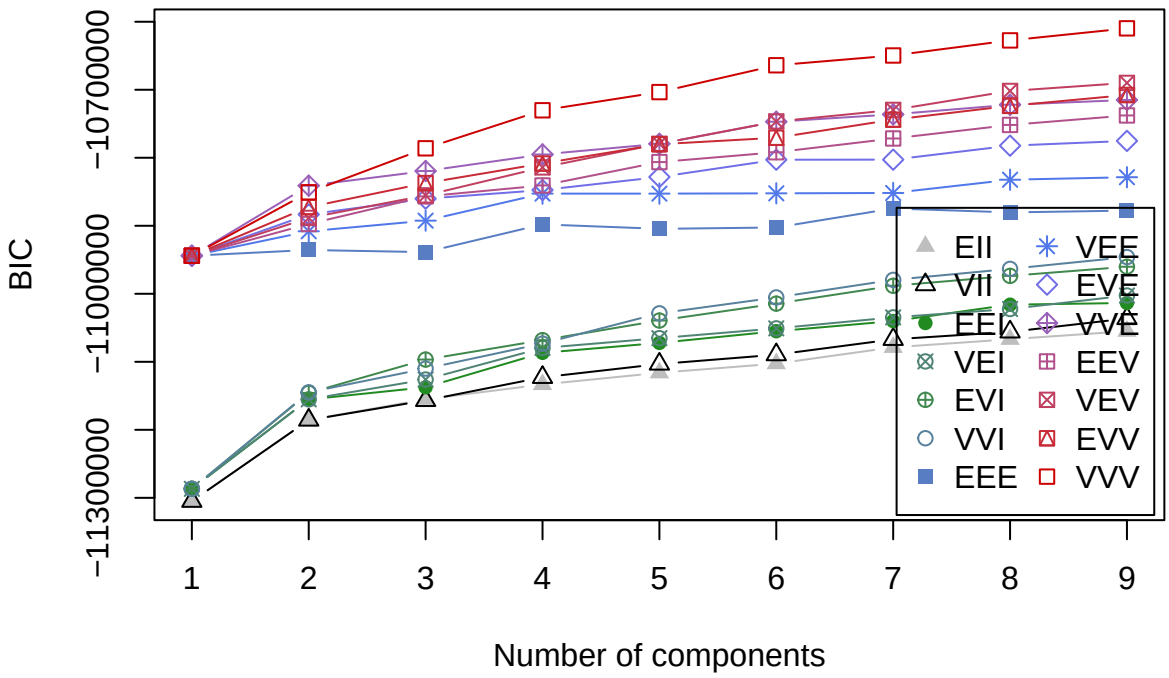
Clusters are over-fitting and placing everything on one side of the tree with these cuts That being said, cuts of 3 give the best random indexes, which seems to match the previous results

##Model Clustering

```
library(mclust)
```

```
## Package 'mclust' version 5.4.10
## Type 'citation("mclust")' for citing this R package in publications.
```

```
fit <- mclustBIC(pos[ , c(3:17)])
plot(fit)
```




```
summary(fit)
```

```
## Best BIC values:
##              VVV,9       VVV,8        VVV,7
## BIC      -10609717 -10627408.5 -10649634.30
## BIC diff         0    -17691.8    -39917.57
```

```
model <- Mclust(pos, x = fit)
summary(model, parameters = TRUE)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VVV (ellipsoidal, varying volume, shape, and orientation) model with 9
## components:
##
##  log-likelihood     n   df BIC ICL
##              NA 74976 1538  NA  NA
##
## Clustering table:
## 1 2 3 4 5 6 7 8 9
## 0 0 0 0 0 0 0 0 0
##
```

```
## Mixing probabilities:
##  1  2  3  4  5  6  7  8  9
## NA NA NA NA NA NA NA NA NA
##
## Means:
##     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
## V1    NA   NA   NA   NA   NA   NA   NA   NA   NA
## V2    NA   NA   NA   NA   NA   NA   NA   NA   NA
## V3    NA   NA   NA   NA   NA   NA   NA   NA   NA
## V4    NA   NA   NA   NA   NA   NA   NA   NA   NA
## V5    NA   NA   NA   NA   NA   NA   NA   NA   NA
## V6    NA   NA   NA   NA   NA   NA   NA   NA   NA
## V7    NA   NA   NA   NA   NA   NA   NA   NA   NA
## V8    NA   NA   NA   NA   NA   NA   NA   NA   NA
## V9    NA   NA   NA   NA   NA   NA   NA   NA   NA
## V10   NA   NA   NA   NA   NA   NA   NA   NA   NA
## V11   NA   NA   NA   NA   NA   NA   NA   NA   NA
## V12   NA   NA   NA   NA   NA   NA   NA   NA   NA
## V13   NA   NA   NA   NA   NA   NA   NA   NA   NA
## V14   NA   NA   NA   NA   NA   NA   NA   NA   NA
## V15   NA   NA   NA   NA   NA   NA   NA   NA   NA
## V16   NA   NA   NA   NA   NA   NA   NA   NA   NA
## V17   NA   NA   NA   NA   NA   NA   NA   NA   NA
##
## Variances:
## [,,1]
##     V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16 V17
## V1  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V2  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V3  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V4  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V5  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V6  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V7  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V8  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V9  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V10 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V11 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V12 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V13 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V14 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V15 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V16 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V17 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## [,,2]
##     V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16 V17
## V1  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V2  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V3  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V4  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V5  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V6  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V7  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V8  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
```

```
## V9  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V10 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V11 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V12 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V13 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V14 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V15 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V16 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V17 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## [,,3]
##     V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16 V17
## V1  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V2  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V3  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V4  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V5  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V6  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V7  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V8  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V9  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V10 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V11 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V12 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V13 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V14 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V15 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V16 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V17 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## [,,4]
##     V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16 V17
## V1  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V2  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V3  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V4  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V5  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V6  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V7  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V8  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V9  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V10 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V11 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V12 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V13 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V14 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V15 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V16 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V17 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## [,,5]
##     V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16 V17
## V1  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V2  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V3  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V4  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V5  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
```

```
## V6  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V7  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V8  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V9  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V10 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V11 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V12 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V13 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V14 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V15 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V16 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V17 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## [,,6]
##     V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16 V17
## V1  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V2  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V3  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V4  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V5  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V6  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V7  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V8  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V9  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V10 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V11 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V12 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V13 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V14 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V15 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V16 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V17 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## [,,7]
##     V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16 V17
## V1  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V2  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V3  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V4  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V5  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V6  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V7  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V8  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V9  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V10 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V11 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V12 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V13 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V14 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V15 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V16 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V17 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## [,,8]
##     V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16 V17
## V1  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V2  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
```

```
## V3  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V4  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V5  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V6  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V7  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V8  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V9  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V10 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V11 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V12 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V13 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V14 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V15 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V16 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V17 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## [,,9]
##     V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16 V17
## V1  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V2  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V3  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V4  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V5  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V6  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V7  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V8  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V9  NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V10 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V11 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V12 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V13 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V14 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V15 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V16 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
## V17 NA NA NA NA NA NA NA NA NA  NA  NA  NA  NA  NA  NA  NA  NA
```

Given the ground truth in the data set, none of the algorithms seemed to work all that well in creating five distinct clusters. The intended learning process for this data is likely determining position from the relative distances and directions between each 3D coordinate point. Hierarchical clustering works best with small data sets, and even finding a subset that consisted of one user for maximum consistency resulted in poor results with no clear cut. K-Means clustering handled the large data set better, somewhat accurately finding 3 clusters and struggling with the remaining two. Using the same data for the model-based clustering, BIC recommended a VVV selection (varying shape, volume, and orientation) with nine different clusters, which is again totally different from the expected five clusters, as this function was not forced to adhere to the same parameters as K-Means clustering.