Exploratory Data Analysis: Human Activity Recognition using Smartphones

Mr. Sachin B.

1. Goal: To show, How to use exploratory data analysis to point in fruitful directions of research, that is, towards answerable questions.

- Exploratory data analysis is a “rough cut” or filter which helps you to find the most beneficial areas of questioning so you can set your priorities accordingly.
- To show you that “real-world” research isn’t always neat and well-defined like textbook questions with clearcut answers.

2. Case Study: (Understanding)

Human Activity Recognition using Smartphones

- The study creating this database involved 30 volunteers "performing activities of daily living (ADL) while carrying a waist-mounted smartphone with embedded inertial sensors.
- Each person performed six activities wearing a smartphone (Samsung Galaxy S II) on the waist.
- The experiments have been video-recorded to label the data manually.
- The obtained dataset has been randomly partitioned into two sets, where,
  - 70% of the volunteers was selected for generating the training data and
  - 30% the test data."

2.1 Dataset

```
ssd <- read.table("./data/ssd.csv")
```

2.2 Understanding Data

```
dim(ssd)
```

2.2.1 Scope of dataset

```
## [1] 7352  563
```

Explanation:

- ssd is pretty big, 7352 observations, each of 563 variables.
- We’ll only use a small portion of this “Human Activity Recognition database”.

```
# Last 2 columns

lc1 <- length(ssd)
lc2 <- length(ssd)-1

names(ssd[lc2:lc1])
```

2.2.2 Last Columns (Classification Label of observation)

```
## [1] "subject"  "activity"
```

```
ssd[sample(1:lc1,10),c(lc2:lc1)]
```

```
##     subject activity
## 182       1 standing
## 233       1  sitting
## 427       3   laying
## 27        1 standing
## 10        1 standing
## 485       3 walkdown
## 400       3  sitting
## 452       3     walk
## 536       3 standing
## 220       1   laying
```

Explanation:

- These last 2 columns contain subject and activity information.

2.2.3 Dataset is Training or Testing Set ?

- We saw above that the gathered data had "been randomly partitioned into two sets, where 70% of the volunteers was selected for generating the training data and 30% the test data."

```
# which subject out of 30 involved in this dataset
table(ssd$subject)
```

```
##
##   1   3   5   6   7   8  11  14  15  16  17  19  21  22  23  25  26  27  28  29
## 347 341 302 325 308 281 316 323 328 366 368 360 408 321 372 409 392 376 382 344
##  30
## 383
```

```
# count of subject
length(table(ssd$subject))
```

```
## [1] 21
```

Explanation:

- As 21 out of 30 subject is available in dataset. this dataset must be training dataset.

Observation 1:

- So we're looking at training data from a machine learning repository.
- **We can infer that this data is supposed to train machines to recognize activity collected from the accelerometers and gyroscopes built into the smartphones that the subjects had strapped to their waists.**

```
table(ssd$activity)
```

2.2.4 Activities characterized by dataset

```
##
##   laying  sitting standing     walk walkdown   walkup
##     1407     1286     1374     1226      986     1073
```

Explanation:

- Because it's training data, each row is labeled with the correct activity (from the 6 possible) and associated with the column measurements (from the accelerometer and gyroscope).

3. Question: "Is the correlation between the measurements and activities good enough to train a machine?"

- so that "Given a set of 561 measurements, would a trained machine be able to determine which of the 6 activities the person was doing?"

3.0 Creating Subset for subject 1

```
## Tranforming 'Activity' column as factor

ssd <- transform(ssd, activity = factor(activity))

# subset of ssd for subject 1
sub1 <- subset(ssd,subject == 1)

# dimension of sub1
dim(sub1)
```

```
## [1] 347 563
```

- So sub1 has fewer than 400 rows now, but still a lot of columns which contain measurements.

```r
# names of some of the column of subject1 subset
names(sub1[,1:12])
```

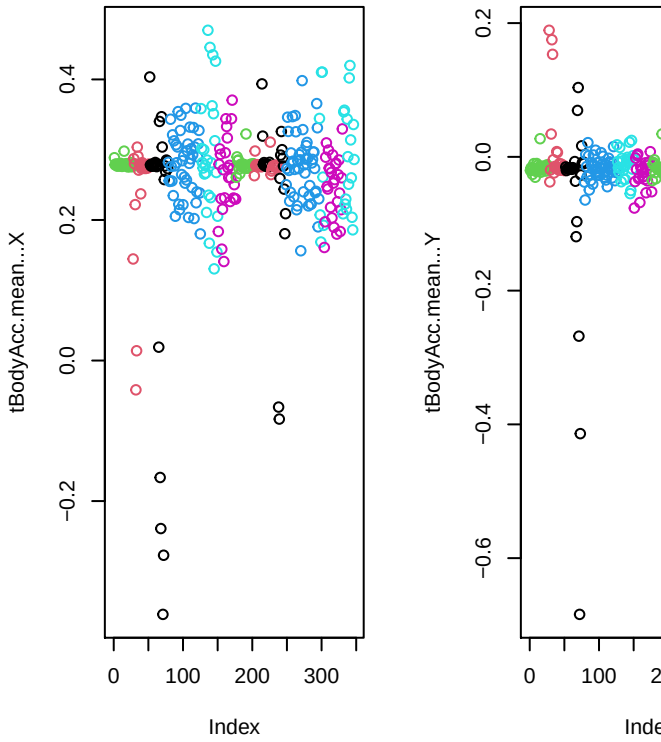
```
##  [1] "tBodyAcc.mean...X" "tBodyAcc.mean...Y" "tBodyAcc.mean...Z"
##  [4] "tBodyAcc.std...X"  "tBodyAcc.std...Y"  "tBodyAcc.std...Z"
##  [7] "tBodyAcc.mad...X"  "tBodyAcc.mad...Y"  "tBodyAcc.mad...Z"
## [10] "tBodyAcc.max...X"  "tBodyAcc.max...Y"  "tBodyAcc.max...Z"
```

Explanation:

- We see X, Y, and Z (3 dimensions) of different aspects of body acceleration measurements, such as mean and standard deviation.

3.1 Finding Pattern in "Body Acceleration - Mean"

```r
par(mfrow = c(1,3))

plot(sub1[,1], col = sub1$activity, ylab = names(sub1)[1])

plot(sub1[,2], col = sub1$activity, ylab = names(sub1)[2])

plot(sub1[,3], col = sub1$activity, ylab = names(sub1)[3])

legend("bottomright",legend=unique(sub1$activity),col=unique(sub1$activity), pch = 1)
```

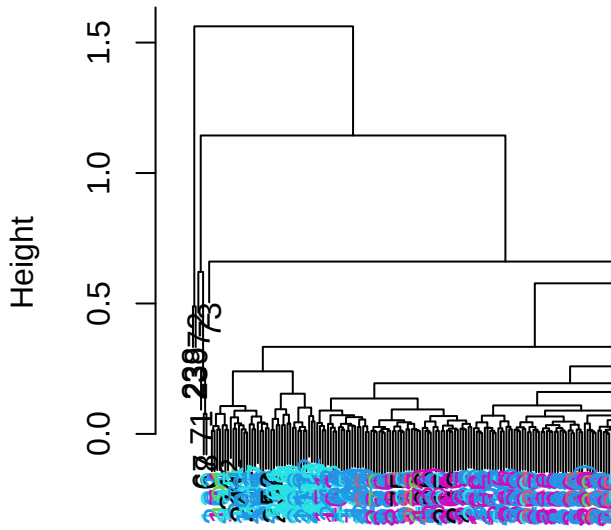


3.1.1 Plot for "Body Acceleration - Mean" - [1:3] Columns

Explanation:

- The plots are a little squished, but we see that the active activities related to walking (shown in the two blues and magenta) show more variability than the passive activities (shown in black, red, and green), particularly in the X dimension.

```
source("myplclust.R")

mdist1 <- dist(sub1[,1:3])
mclust1 <- hclust(mdist1)

myplclust(mclust1, lab.col = unclass(sub1$activity))
```



hclust (

3.1.2 hclust() for "Body Acceleration - Mean" - [1:3] Columns

Explanation:

- dendrogram doesn't look too helpful, There's no clear grouping of colors, except that active colors (blues and magenta) are near each other as are the passive (black, red, and green).
- So mean body acceleration doesn't tell us much.

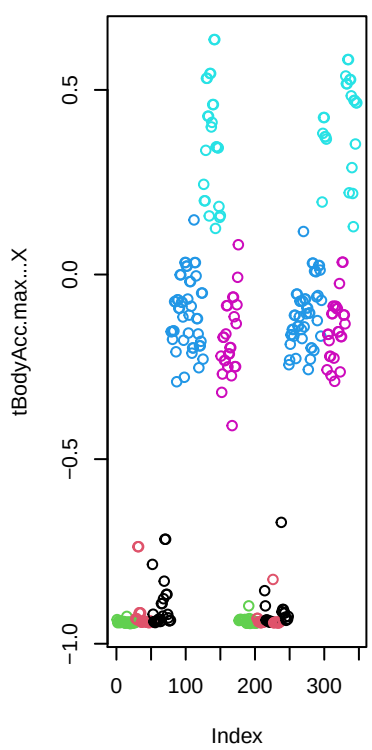
3.2 Finding Pattern in "Body Acceleration - Max"

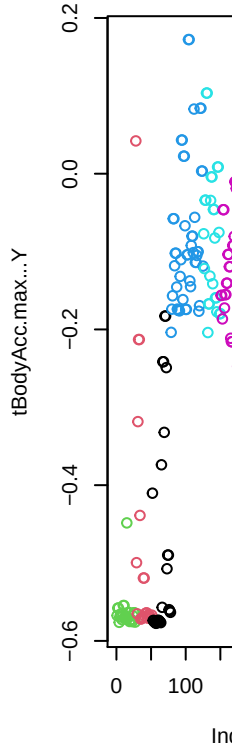
```
# names of some of the column of subject1 subset
names(sub1[,1:12])
```

```
##  [1] "tBodyAcc.mean...X" "tBodyAcc.mean...Y" "tBodyAcc.mean...Z"
##  [4] "tBodyAcc.std...X"  "tBodyAcc.std...Y"  "tBodyAcc.std...Z"
##  [7] "tBodyAcc.mad...X"  "tBodyAcc.mad...Y"  "tBodyAcc.mad...Z"
## [10] "tBodyAcc.max...X"  "tBodyAcc.max...Y"  "tBodyAcc.max...Z"
```

```
par(mfrow = c(1,3))

plot(sub1[,10], col = sub1$activity, ylab = names(sub1)[10])
```

```
plot(sub1[,11], col = sub1$activity, ylab = names(sub1)[11])

plot(sub1[,12], col = sub1$activity, ylab = names(sub1)[12])

legend("bottomright",legend=unique(sub1$activity),col=unique(sub1$activity), pch = 1)
```



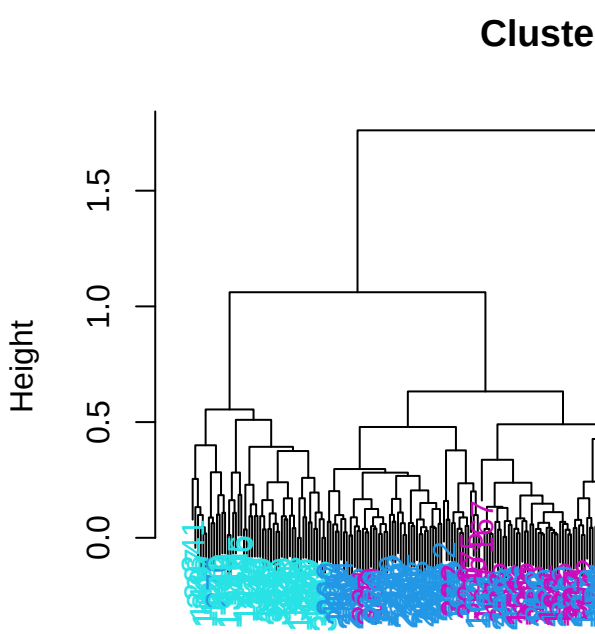


3.2.1 Plot for "Body Acceleration - Max" - [10:12] Columns

Explanation:

- The x-axis of each show the 300+ observations and the y-axis indicates the maximum acceleration.
- **From above plot we can see that, passive activities mostly fall below the walking activities**

```
source("myplclust.R")

mdist2 <- dist(sub1[,10:12])
mclust2 <- hclust(mdist2)

myplclust(mclust2, lab.col = unclass(sub1$activity))
```



3.2.2 hclust() for "Body Acceleration - Max" - [10:12] Columns

Explanation:

- Now we see clearly that the data splits into 2 clusters, active and passive activities.
- Moreover, the light blue (walking down) is clearly distinct from the other walking activities.
- The dark blue (walking level) also seems to be somewhat clustered.
- The passive activities, however, seem all jumbled together with no clear pattern visible.

3.3 Finding Pattern using Singular Value Decomposition (SVD)

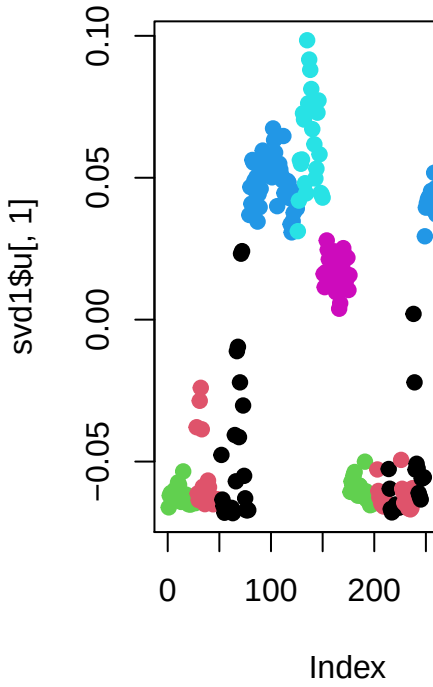
```
svd1 <- svd(scale(sub1[,-c(lc2,lc1)])) # lc2 & lc1 are 2nd last and last column respectively.


# To see dimension of LEFT singular vectors of sub1
dim(svd1$u)
```

```
## [1] 347 347
```

EXplanation:

- Each row in u corresponds to a row in the matrix sub1.
- Recall that in sub1 each row has an associated activity.

```
par(mfrow = c(1,2))

plot(svd1$u[,1], col = sub1$activity, pch = 19)

plot(svd1$u[,2], col = sub1$activity, pch = 19)
```



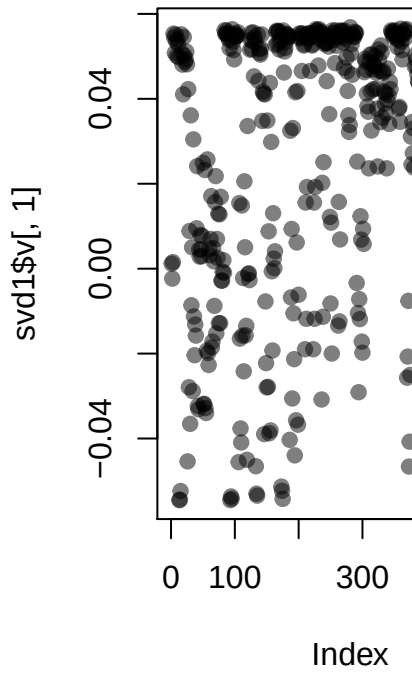
3.3.1 Finding Pattern in 1st & 2nd LEFT Singular Vectors of svd1 (svd1$u)

Explanation

- Here we're looking at the 2 left singular vectors of svd1 (the first 2 columns of svd1$u).
- Each entry of the columns belongs to a particular row with one of the 6 activities assigned to it.
- We see the activities distinguished by color.
- Moving from left to right, the first section of rows are green (standing), the second red (sitting), the third black (laying), etc.
- The first column of u shows separation of the nonmoving (black, red, and green) from the walking activities.
- The second column is harder to interpret. However, **the magenta cluster, which represents walking up, seems separate from the others.**

- **we'll look at the RIGHT singular vectors (the columns of svd1$v), and in particular, the second one since the separation of the magenta cluster stood out in the second column of svd1$u**.

```
par(mfrow = c(1,2))

plot(svd1$v[,1], col = rgb(0,0,0,0.5), pch = 19)

plot(svd1$v[,2], col = rgb(0,0,0,0.5), pch = 19)
```



3.3.2 Finding Pattern in 1st & 2nd RIGHT Singular Vectors of svd1 (svd1$v)

Explanation

- Here's a plot of the second column of svd1$v.
- We used transparency in our plotting but **nothing clearly stands out here**.

```
maxCon <- which.max(svd1$v[,2])

names(sub1[maxCon])
```

3.3.3 Use of clustering to find the feature (out of the 500+) which contributes the most to the variation of this second column of svd1$v.

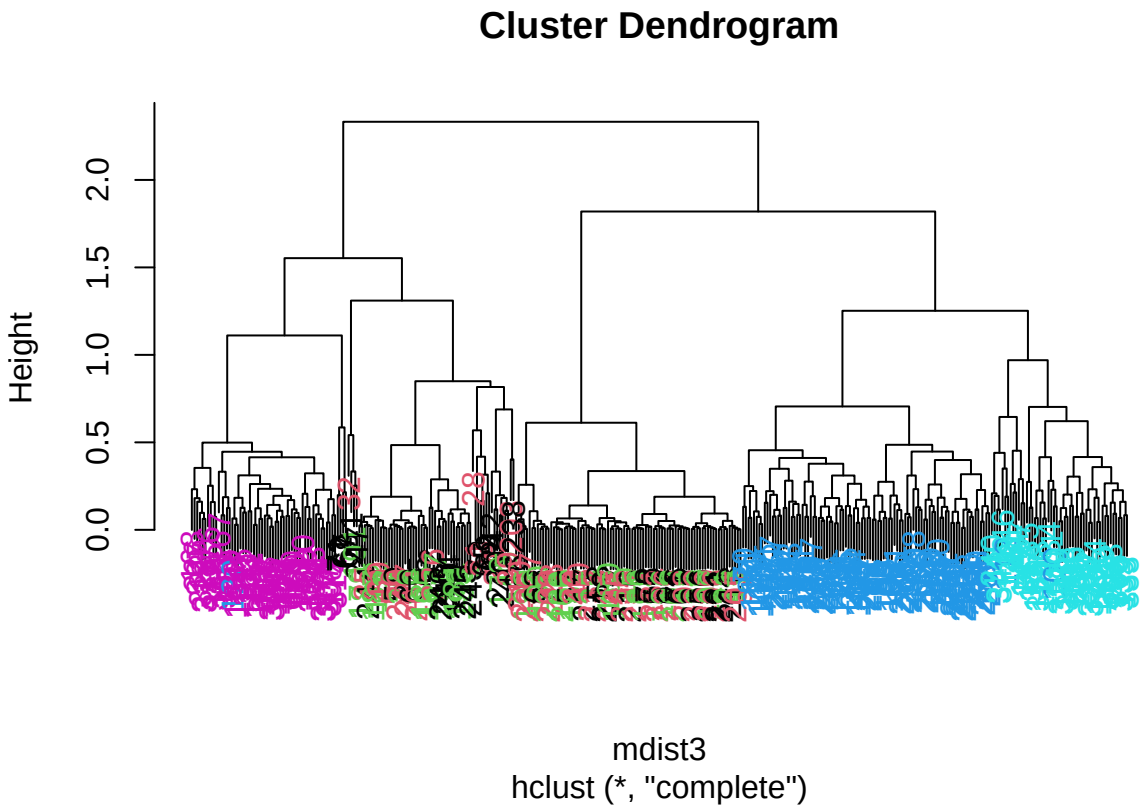
```
## [1] "fBodyAcc.meanFreq...Z"
```

Explanation:

- So the mean body acceleration in the frequency domain in the Z direction is the main contributor.

```
source("myplclust.R")

mdist3 <- dist(sub1[,c(10:12,maxCon)])
mclust3 <- hclust(mdist3)

myplclust(mclust3, lab.col = unclass(sub1$activity))
```

3.3.3.1 hclust() for "Body Acceleration - Max" - [10:12] Columns PLUS which.max()



Explanation:

- Now we see some real separation.
- Magenta (walking up) is on the far left, and the two other walking activities, the two blues, are on the far right, but in separate clusters from one another.
- The nonmoving activities still are jumbled together.

```
names(sub1[maxCon])
```

```
## [1] "fBodyAcc.meanFreq...Z"
```

Explanation:

- So the mean body acceleration in the frequency domain in the Z direction is the main contributor to this clustering phenomenon.

3.4 Finding Pattern using Kmeans Clustering

- Create the variable kClust by assigning to it the output of the R command kmeans with 2 arguments.
    1. The first is sub1 with the last 2 columns removed. (Recall these don't have pertinent information for clustering analysis.)
    2. The second argument to kmeans is centers set equal to 6, the number of activities we know we have.

```
kClust<- kmeans(sub1[,-c(lc2,lc1)], centers = 6)

table(kClust$cluster, sub1$activity)
```

3.4.1 Kmeans with 1 Random Start

```
##
##     laying sitting standing walk walkdown walkup
##   1      0       0        0   22        0      0
##   2      0       0        0   46        0      0
##   3     42      45       53    0        0      0
##   4      8       2        0    0        0     53
##   5      0       0        0    0       48      0
##   6      0       0        0   27        1      0
```

Explanation:

- Your exact output will depend on the state of your random number generator.
- We notice that when we just run with 1 random start, the clusters tend to group the nonmoving activities together in one cluster.
- The walking activities seem to cluster individually by themselves.
- You could run the call to kmeans with one random start again and you'll probably get a slightly different result

```
kClust<- kmeans(sub1[,-c(lc2,lc1)], centers = 6, nstart = 100)

table(kClust$cluster, sub1$activity)
```

3.4.2 Kmeans with 100 Random Start

```
##
##     laying sitting standing walk walkdown walkup
##   1     29       0        0    0        0      0
##   2      0       0        0   95        0      0
##   3      3       0        0    0        0     53
##   4      0       0        0    0       49      0
##   5      0      37       51    0        0      0
##   6     18      10        2    0        0      0
```

Explanation:

- We see that even with 100 random starts, the passive activities tend to cluster together.
- **One of the clusters contains only laying**, but in another cluster, standing and sitting group together.

3.4.3 Plot to look at the features (columns) of Kmeans Cluster centers to see if any dominate.

```
#to find the dimensions of kClust's centers
dim(kClust$centers)
```

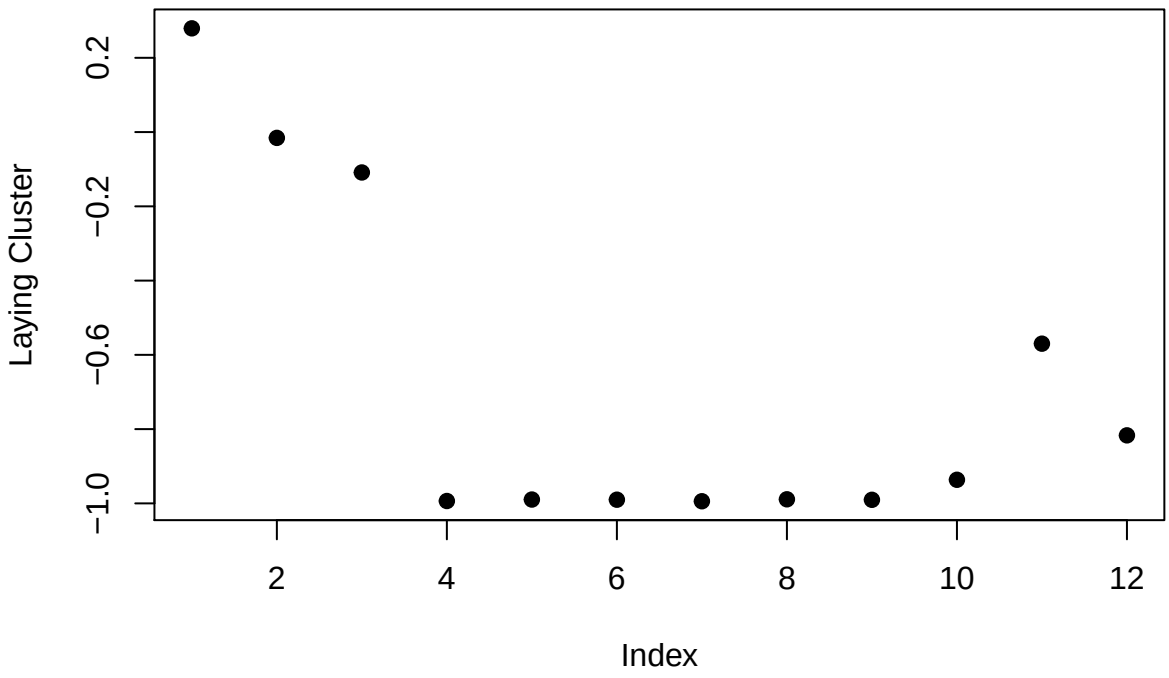
3.4.3.1 laying <- which(kClust$size==29)

```
## [1]   6 561
```

```
# Why laying is equal to 29
table(kClust$cluster, sub1$activity)
```

```
##
##     laying sitting standing walk walkdown walkup
##   1     29       0        0    0        0      0
##   2      0       0        0   95        0      0
##   3      3       0        0    0        0     53
##   4      0       0        0    0       49      0
##   5      0      37       51    0        0      0
##   6     18      10        2    0        0      0
```

```
laying <- which(kClust$size==29)

plot(kClust$centers[laying,1:12],pch=19,ylab="Laying Cluster")
```



Explanation:

- We see the first 3 columns dominate this cluster center.

```
names(sub1[,1:3])
```

```
## [1] "tBodyAcc.mean...X" "tBodyAcc.mean...Y" "tBodyAcc.mean...Z"
```

Explanation:

- **So the 3 directions of mean body acceleration seem to have the biggest effect on laying.**

```
#to find the dimensions of kClust's centers
dim(kClust$centers)
```

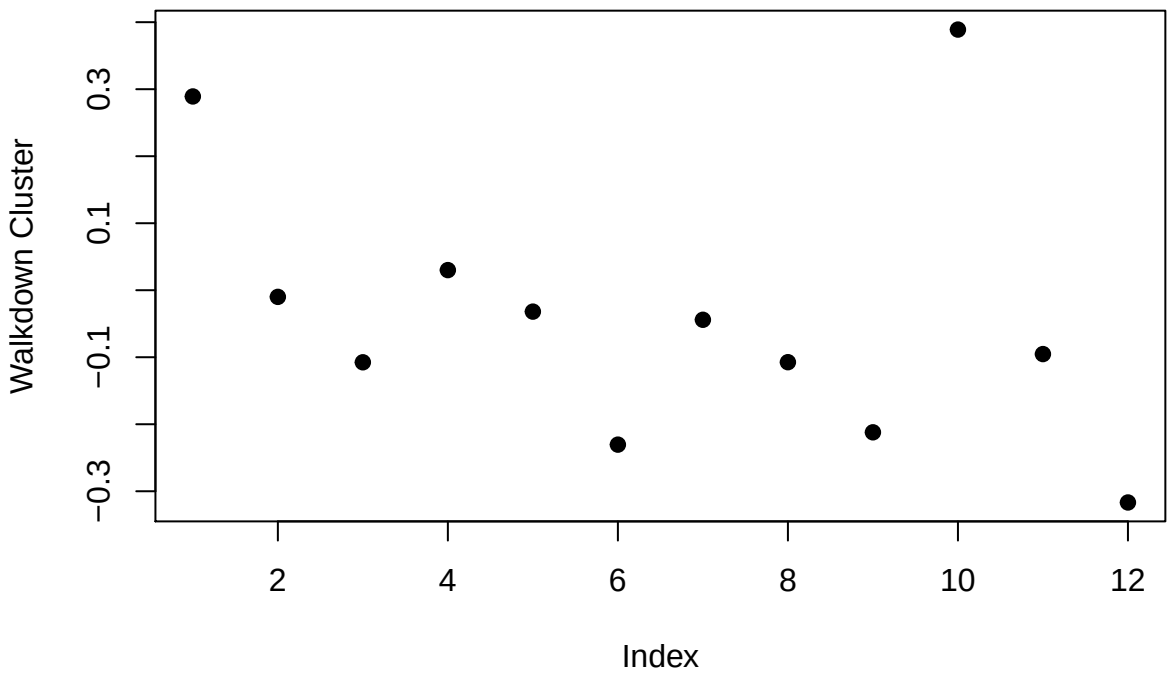
3.4.3.2 walkdown <- which(kClust$size==49)

```
## [1]   6 561
```

```r
# Why walkdown is equal to 49
table(kClust$cluster, sub1$activity)
```

```
##
##      laying sitting standing walk walkdown walkup
##   1     29       0        0    0        0      0
##   2      0       0        0   95        0      0
##   3      3       0        0    0        0     53
##   4      0       0        0    0       49      0
##   5      0      37       51    0        0      0
##   6     18      10        2    0        0      0
```

```r
walkdown <- which(kClust$size==49)

plot(kClust$centers[walkdown,1:12],pch=19,ylab="Walkdown Cluster")
```



Explanation:

- We see an interesting pattern here. From left to right, looking at the 12 acceleration measurements in groups of 3, the points decrease in value.
- The X direction dominates, followed by Y then Z.
- This might tell us something more about the walking down activity.

4. Summary

1. This example might have convinced you that real world analysis can be frustrating sometimes and not always obvious.
2. You might have to try several techniques of exploratory data analysis before you hit one that pays off and leads you to the questioms that will be the most promising to explore.
3. We saw here that the sensor measurements were pretty good at discriminating between the 3 walking activities, but the passive activities were harder to distinguish from one another.

- These might require more analysis or an entirely different set of sensory measurements.