ejemplo de clustering

En esta lección, aplicaremos algunas de las técnicas analíticas que aprendimos en este curso a los datos de la Universidad de California, Irvine. Específicamente, los datos que usaremos provienen del Centro de Aprendizaje Automático y Sistemas Inteligentes de la UCI. Puede obtener más información sobre los datos en http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones.Como indica esta dirección, los datos involucran teléfonos inteligentes y reconocen la actividad humana. ¿Guay, verdad?

Nuestro objetivo es mostrarle cómo utilizar el análisis de datos exploratorios para orientarle en direcciones fructíferas de investigación, es decir, hacia preguntas que se puedan responder. El análisis exploratorio de datos es un "corte preliminar" o filtro que le ayuda a encontrar las áreas de interrogación más beneficiosas para que pueda establecer sus prioridades en consecuencia.

También esperamos mostrarle que la investigación del "mundo real" no siempre es ordenada y bien definida como las preguntas de un libro de texto con respuestas claras.

El estudio que creó esta base de datos involucró a 30 voluntarios "que realizaban actividades de la vida diaria (ADL) mientras llevaban un teléfono inteligente montado en la cintura con sensores inerciales integrados ... Cada persona realizó seis actividades ... usando un teléfono inteligente (Samsung Galaxy S II) en la cintura ... Los experimentos se grabaron en video para etiquetar los datos manualmente El conjunto de datos obtenido se dividió aleatoriamente en dos conjuntos, donde se seleccionó el 70% de los voluntarios para generar los datos de entrenamiento y el 30% los datos de prueba."

```
ssd<-read.csv("C:/Users/luism/Dropbox/x.csv")
ssd<-ssd[,-c(1)]
dim(ssd)
```

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

```
names(ssd[562:563])
```

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

Estas 2 últimas columnas contienen información sobre el tema y la actividad. Vimos anteriormente que los datos recopilados se habían "dividido aleatoriamente en dos conjuntos, donde el 70% de los voluntarios fue seleccionado para generar los datos de entrenamiento y el 30% los datos de prueba". Ejecute la tabla de comandos R con ssd $ subject como argumento para ver si los datos en ssd contienen datos de prueba o de entrenamiento.

```
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
```

```r
sum(table(ssd$subject))
```

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

Asi que , estamos analizando datos de entrenamiento de un repositorio de aprendizaje automático. Podemos inferir que se supone que estos datos entrenan a las máquinas para que reconozcan la actividad recopilada de los acelerómetros y giroscopios integrados en los teléfonos inteligentes que los sujetos tenían atados a la cintura. Ejecute la tabla de comandos R en ssd $ activity para ver qué actividades se han caracterizado por estos datos.

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

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

Disponemos de 6 actividades, 3 pasivas (tumbado, de pie y sentado) y 3 activas que implican caminar.

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). We're interested in questions such as, "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?"

Primero, masajeemos un poco los datos para que sea más fácil trabajar con ellos. Ya hemos ejecutado el comando R transform en los datos para que las actividades sean factores. Esto nos permitirá codificarlos con colores cuando generemos gráficos. Veamos solo el primer tema (numerado 1). Cree la variable sub1 asignándole la salida del subconjunto de comandos R con ssd como primer argumento y el booleano, sujeto igual a 1, como segundo.

```r
sub1 <- subset(ssd, subject == 1)
sub1$activity<-as.factor(sub1$activity)
dim(sub1)
```

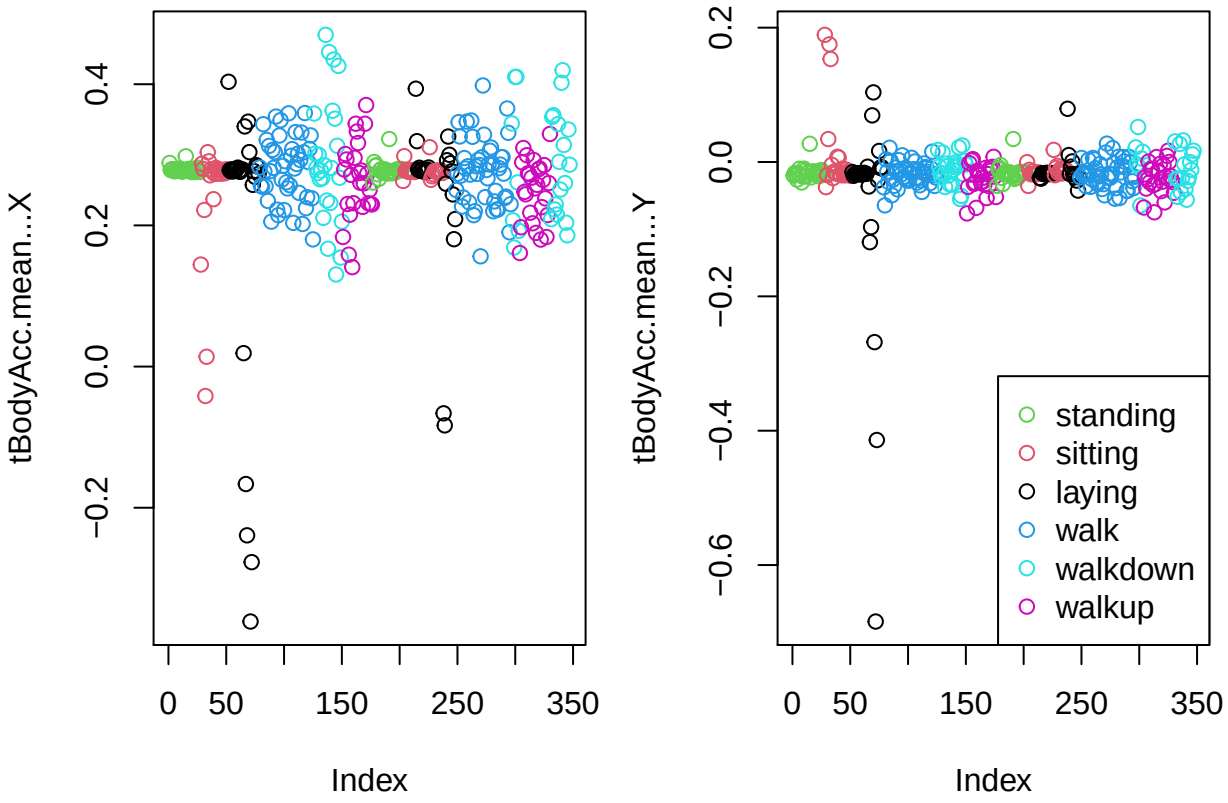
```
## [1] 347 563
```

```r
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"
```

Vemos X, Y y Z (3 dimensiones) de diferentes aspectos de las medidas de aceleración corporal, como la media y la desviación estándar. Hagamos ahora algunas comparaciones de actividades observando gráficos de la aceleración media del cuerpo en las direcciones X e Y.

```r
par(mfrow=c(1, 2), mar = c(5, 4, 1, 1))
plot(sub1[, 1], col = sub1$activity, ylab = names(sub1)[1])
plot(sub1[, 2], col = sub1$activity, ylab = names(sub1)[2])
legend("bottomright",legend=unique(sub1$activity),col=unique(sub1$activity), pch = 1)
```



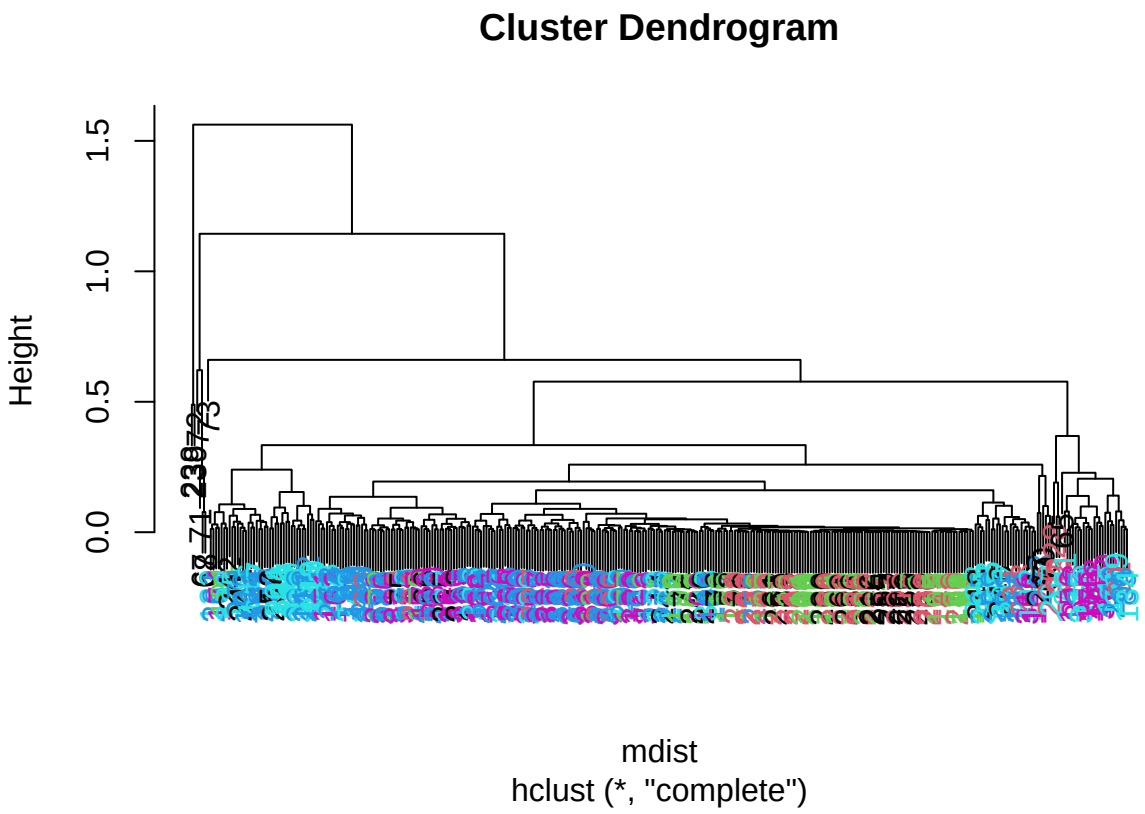

```
par(mfrow=c(1,1))
```

Ves tanto el código como su salida! Las tramas están un poco aplastadas, pero vemos que las actividades activas relacionadas con caminar (mostradas en los dos azules y magenta) muestran más variabilidad que las actividades pasivas (mostradas en negro, rojo y verde), particularmente en la dimensión X.

Create a distance matrix, mdist, of thefirst 3 columns of sub1, by using the R command dist.Use the x[,1:3] notation to specify the columns.

```
myplclust<-function( hclust, lab=hclust$labels, lab.col=rep(1,length(hclust$labels)), hang=0.1,...){
## modifiction of plclust for plotting hclust objects *in colour*!
## Copyright Eva KF Chan 2009
## Arguments:
##    hclust:    hclust object
##    lab:        a character vector of labels of the leaves of the tree
##    lab.col:    colour for the labels; NA=default device foreground colour
##    hang:     as in hclust & plclust
## Side effect:
##    A display of hierarchical cluster with coloured leaf labels.
y <- rep(hclust$height,2)
x <- as.numeric(hclust$merge)
y <- y[which(x<0)]
x <- x[which(x<0)]
x <- abs(x)
y <- y[order(x)]
x <- x[order(x)]
```

```
 plot( hclust, labels=FALSE, hang=hang, ... )
 text( x=x, y=y[hclust$order]-(max(hclust$height)*hang), labels=lab[hclust$order], col=lab.col[hclust$o
mdist <- dist(sub1[,1:3])
hclustering <- hclust(mdist)
myplclust(hclustering, lab.col = unclass(sub1$activity))
```



Bueno, ese dendrograma no parece muy útil, ¿verdad? No hay una agrupación clara de colores, excepto que los colores activos (azul y magenta) están cerca unos de otros al igual que los pasivos (negro, rojo y verde). Entonces, la aceleración promedio no nos dice mucho. ¿Qué tal la aceleración máxima? Veamos eso para el primer sujeto (en nuestra matriz sub1) para las dimensiones X e Y. Estos están en las columnas 10 y 11.

Aquí se representan uno al lado del otro, la dimensión X a la izquierda y la Y a la derecha. El eje x de cada uno muestra las más de 300 observaciones y el eje y indica la aceleración máxima.

Finalmente estamos viendo algo vagamente interesante! Centrémonos entonces en las 3 dimensiones de máxima aceleración, almacenadas en las columnas 10 a 12 de sub1. Cree una nueva matriz de distancia, mdist, de estas 3 columnas de sub1, utilizando el comando R dist. Nuevamente, use la notación x [, 10: 12] para capturar las columnas.

```
mdist <- dist(sub1[,10:12])
hclustering <- hclust(mdist)
myplclust(hclustering, lab.col = unclass(sub1$activity))
```

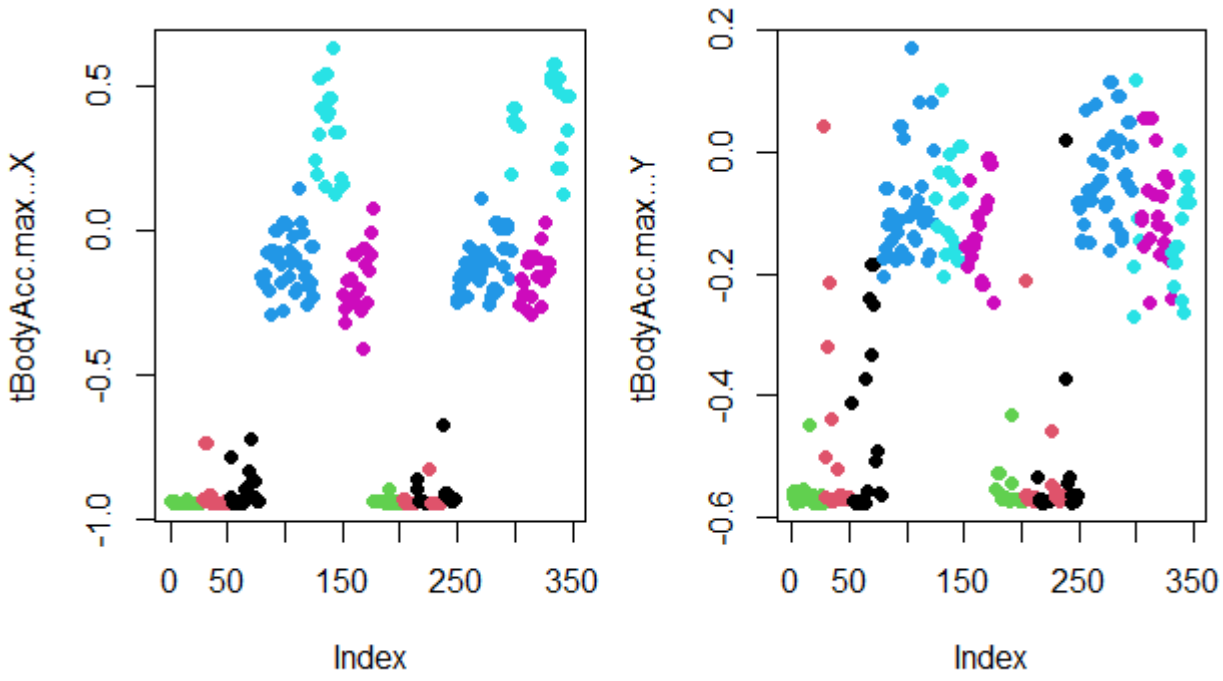
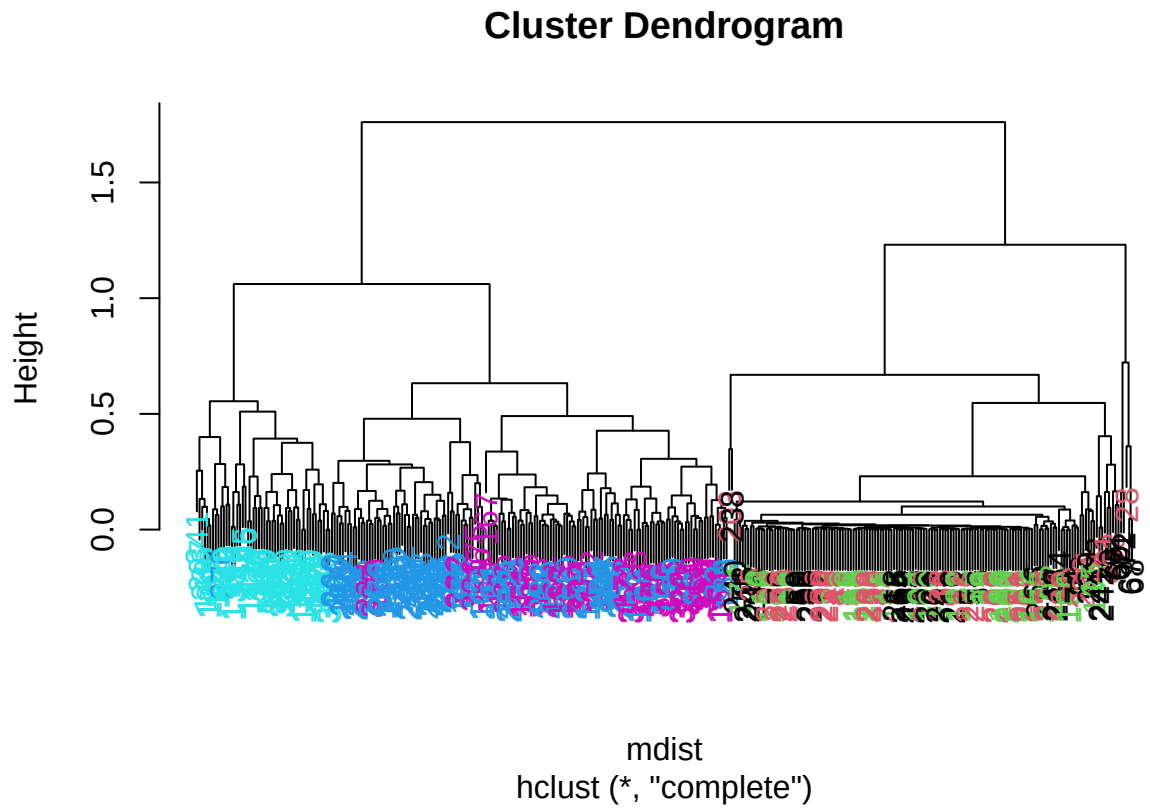


Figure 1: A caption



Ahora vemos claramente que los datos se dividen en 2 grupos, actividades activas y pasivas. Además, el azul claro (caminar hacia abajo) es claramente distinto de las otras actividades para caminar. El azul oscuro (nivel para caminar) también parece estar algo agrupado. Las actividades pasivas, sin embargo, parecen todas mezcladas sin un patrón claro visible.

Probemos algo de SVD ahora. Cree la variable svd1 asignándole la salida de una llamada al comando R svd. El argumento de svd debe ser scale (sub1 [, - c (562,563)]). Esto eliminará las últimas 2 columnas de sub1 y escalará los datos. Recuerde que las últimas 2 columnas contienen información sobre la actividad y el tema que no necesitaremos.

```
sub1[,-c(562,563)]<-sapply(sub1[,-c(562,563)],as.numeric)
svd1 <- svd(scale(sub1[,-c(562,563)]))
dim(svd1$u)
```

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

Vemos que la matriz u es una matriz de 347 por 347. Cada fila en u corresponde a una fila en la matriz sub1. Recuerde que en sub1 cada fila tiene una actividad asociada.

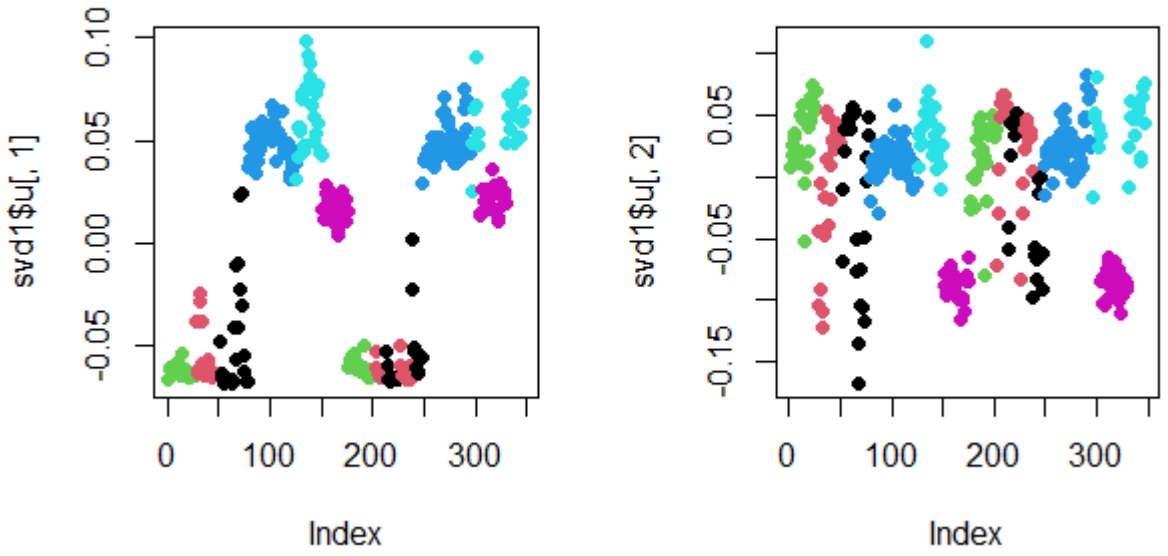


Figure 2: A caption

Aquí estamos viendo los 2 vectores singulares izquierdos de svd1 (las 2 primeras columnas de svd1 $ u). Cada entrada de las columnas pertenece a una fila en particular con una de las 6 actividades asignadas. Vemos las actividades distinguidas por el color. Moviéndose de izquierda a derecha, la primera sección de filas es verde (de pie), la segunda roja (sentada), la tercera negra (acostada), etc. La primera columna de u muestra la separación de los no móviles (negro, rojo y verde) de las actividades de caminata. La segunda columna es más difícil de interpretar. Sin embargo, el grupo magenta, que representa caminar hacia arriba, parece separado de los demás.

Intentaremos averiguar por qué es así. Para hacer eso, tendremos que encontrar cuál de las más de 500 medidas (representadas por las columnas de sub1) contribuye a la variación de ese componente.

Como estamos interesados en las columnas sub1, veremos los vectores singulares DERECHA (las columnas de svd1v)$y, en particular, el segundo desde la separación del clúster magenta se destacó en la segunda columna de svd1u.

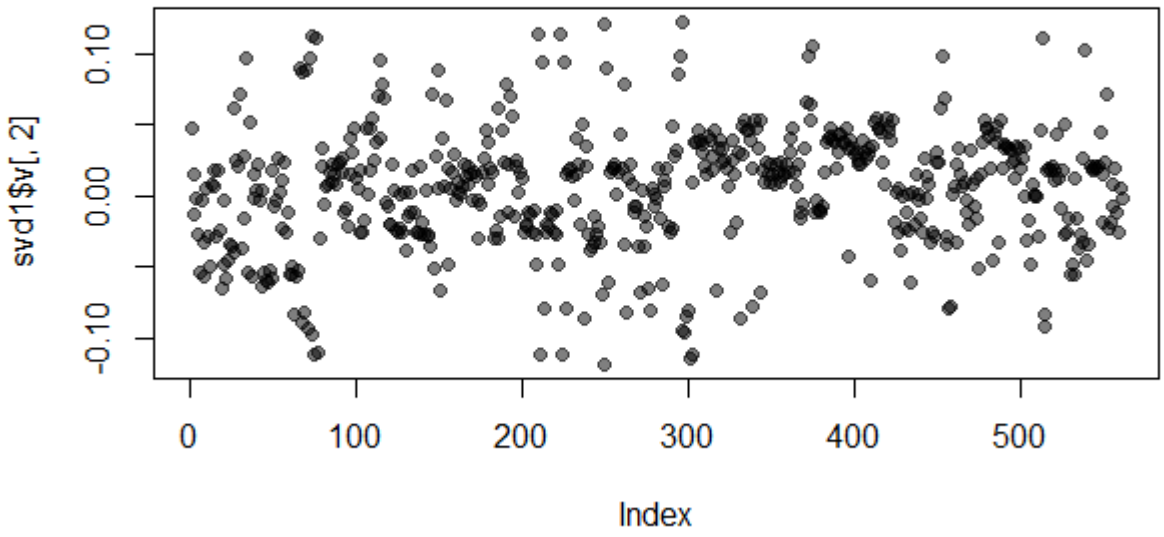
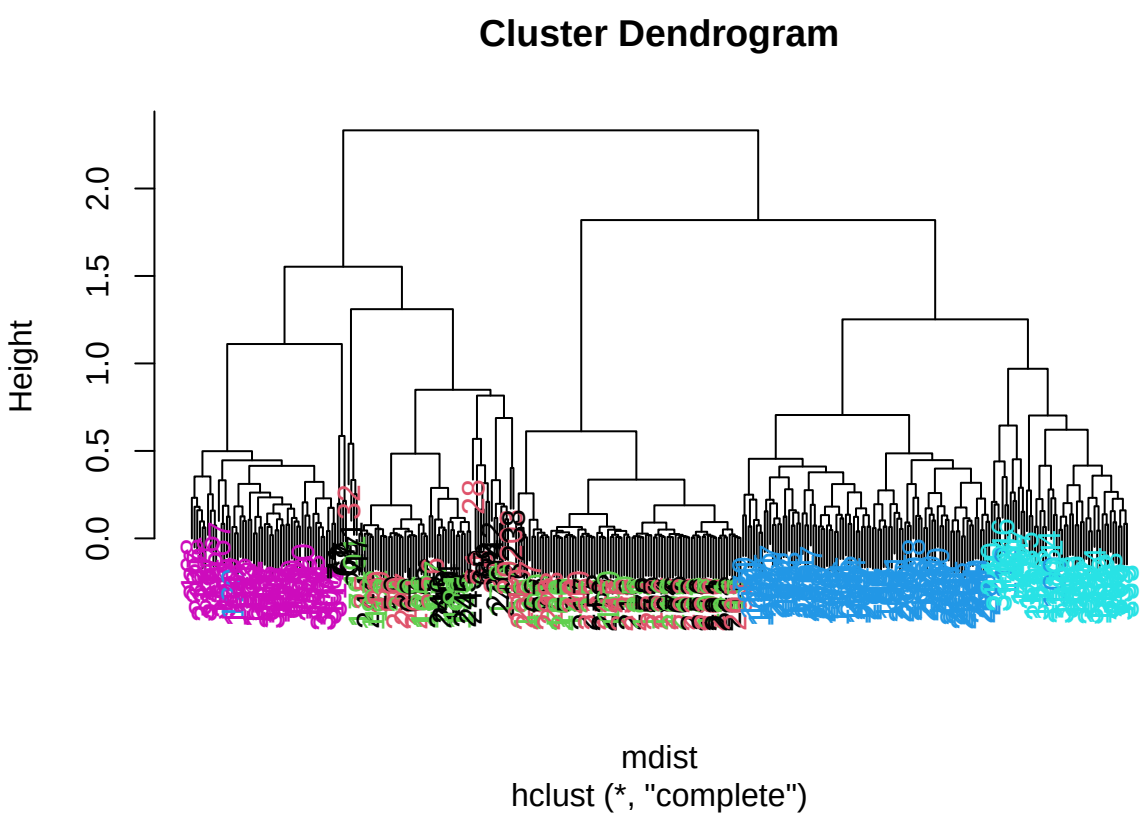


Figure 3: A caption

Aquí hay una gráfica de la segunda columna de svd1v. Usamos transparencia en nuestro trazado, pero aquí no se destaca nada claro

```
maxCon <- which.max(svd1$v[,2])
mdist <- dist(sub1[,c(10:12,maxCon)])
hclustering <- hclust(mdist)
myplclust(hclustering, lab.col = unclass(sub1$activity))
```




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

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

Ahora vemos una separación real. Magenta (caminar hacia arriba) está en el extremo izquierdo, y las otras dos actividades para caminar, los dos azules, están en el extremo derecho, pero en grupos separados entre sí. Las actividades inmóviles todavía están mezcladas.

Entonces, la aceleración media del cuerpo en el dominio de la frecuencia en la dirección Z es el principal contribuyente a este fenómeno de agrupamiento que estamos viendo. Pasemos a la agrupación de k-medias para ver si esta técnica puede distinguir entre las actividades.

Cree la variable kClust asignándole la salida del comando R kmeans con 2 argumentos. El primero es sub1 con las últimas 2 columnas eliminadas. (Recuerde que estos no tienen información pertinente para el análisis de conglomerados). El segundo argumento para kmeans son centros establecidos en 6, el número de actividades que sabemos que tenemos.

```
kClust <- kmeans(sub1[, -c(562, 563)], centers = 6)
table(kClust$cluster, sub1$activity)
```

```
##
##     laying sitting standing walk walkdown walkup
##   1     19      13        5    0        0      0
##   2      0       0        0    0       48      0
##   3      5       0        0    0        0     53
```

```
##   4     26      34       48    0        0      0
##   5      0       0        0   45        0      0
##   6      0       0        0   50        1      0
```

Su salida exacta dependerá del estado de su generador de números aleatorios. Observamos que cuando solo ejecutamos con 1 inicio aleatorio, los grupos tienden a agrupar las actividades inmóviles en un grupo. Las actividades de caminar parecen agruparse individualmente por sí mismas. Puede ejecutar la llamada a kmeans con un inicio aleatorio nuevamente y probablemente obtendrá un resultado ligeramente diferente, pero en su lugar llame a kmeans con 3 argumentos, el último de los cuales le dirá que intente más inicios aleatorios y devuelva el mejor. Los primeros 2 argumentos deben ser los mismos que antes (sub1 con las últimas 2 columnas eliminadas y los centros establecidos en 6). El tercero es nstart establecido igual a 100. Vuelva a poner el resultado en kClust.

```
kClust <- kmeans(sub1[, -c(562, 563)], centers = 6, nstart=100)
table(kClust$cluster, sub1$activity)
```

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

Vemos que incluso con 100 inicios aleatorios, las actividades pasivas tienden a agruparse. Uno de los grupos contiene solo tendido, pero en otro grupo, grupo de pie y sentado juntos.

Utilice dim para encontrar las dimensiones de los centros de kClust. Utilice la notación x$y para acceder a ellos.

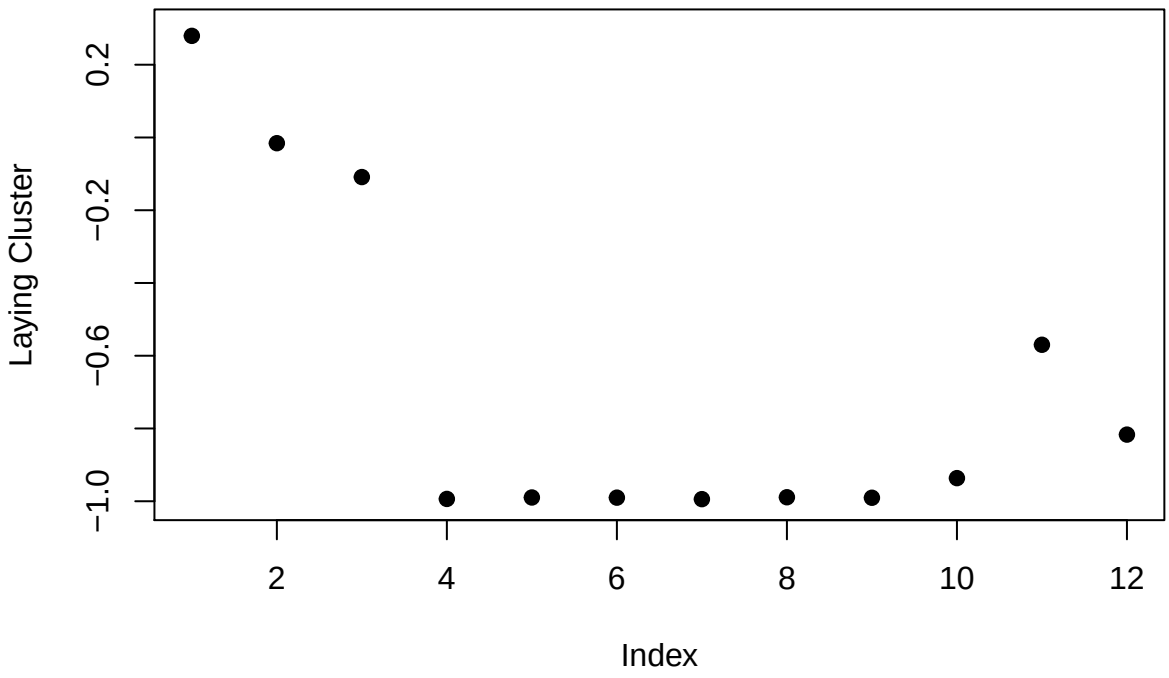
```
dim(kClust$centers)
```

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

So the centers are a 6 by 561 array. Sometimes it's a good idea to look at the features (columns) of these centers to see if any dominate.

Create the variable laying and assign to it the output of the call to the Rcommand which with the argument kClust$size==29.

```
laying <- which(kClust$size==29)
plot(kClust$centers[laying, 1:12],pch=19,ylab="Laying Cluster",xlim = c(1,12),ylim = c(-1,0.3))
```



Vemos que las primeras 3 columnas dominan este centro del grupo. Ejecute nombres con las primeras 3 columnas de sub1 como argumento para recordar lo que contienen estas columnas.

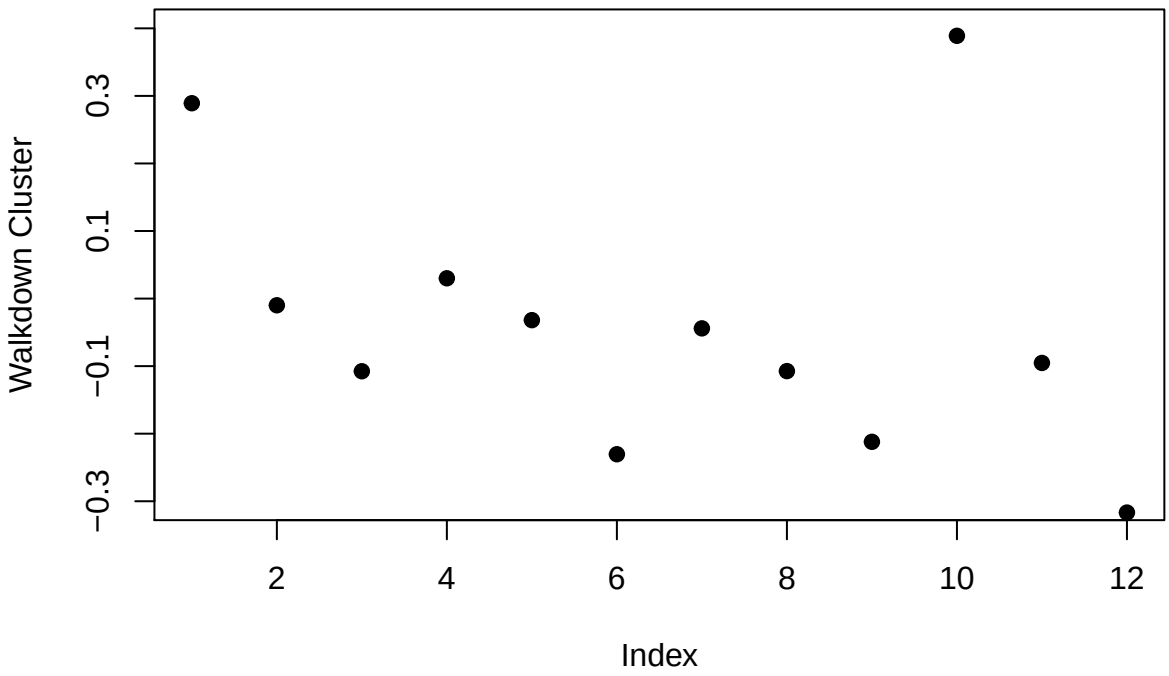
```
names(sub1[,1:3])
```

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

Entonces, las 3 direcciones de la aceleración corporal media parecen tener el mayor efecto en la postura.

Cree la variable walkdown y asígnele la salida de la llamada al comando R que con el argumento kClust $ size == 49.

```
walkdown <- which(kClust$size==49)
plot(kClust$centers[walkdown, 1:12],pch=19,ylab="Walkdown Cluster",xlim = c(1,12),ylim = c(-0.3,0.4))
```



Vemos un patrón interesante aquí. De izquierda a derecha, mirando las 12 medidas de aceleración en grupos de 3, los puntos disminuyen de valor. La dirección X domina, seguida de Y luego Z. Esto podría decirnos algo más sobre la actividad de caminar hacia abajo.

Terminaremos aquí y esperamos que este ejemplo lo convenza de que el análisis del mundo real puede ser frustrante a veces y no siempre obvio. Es posible que deba probar varias técnicas de análisis de datos exploratorios antes de encontrar una que valga la pena y lo lleve a las misiones que serán más prometedoras para explorar.

Vimos aquí que las mediciones del sensor eran bastante buenas para discriminar entre las 3 actividades de caminar, pero las actividades pasivas eran más difíciles de distinguir entre sí. Estos pueden requerir más análisis o un conjunto completamente diferente de medidas sensoriales.