UNIVERSIDAD ADOLFO IBÁÑEZ

Fuente: Michael Richmond, spiff.rit.edu

Actividad I

MÉTODOS DE CLUSTERING

Alex Escudero | Inteligencia de Negocios | 12 Septiembre 2017

Recursos Humanos

Dado el encargo de diseñar un script que pueda responder variadas preguntas sobre una base de datos de recursos humanos de una empresa, se decide usar R. Esta tarea se cumple específicamente usando técnicas de reducción dimensional y análisis de clusters usando funcionalidades incluidas en paquetes del software.

1) Primero se identifican las variables categóricas una vez teniendo visualizada la matriz de datos, estas son; (Variable/Columna) usando

```
head(employee)
```

- Attrition/2
- BusinessTravel/3
- Department/5
- EducationField/8
- Gender/12
- Jobrole/16
- MaritalStatus/18
- Over18/22
- Overtime/23

Por otro lado, las variables numéricas que permanecen constantes para todos los empleados serían:

- EmployeeCount/9
- StandardHours/27

Luego se procede a borrar tales columnas identificadas del dataframe y se estandarizan los datos.

Para borrar tales columnas se utiliza:

```
employee<-employee[-c(2,3,5,8,12,16,18,22,23,9,27)]
> head(employee)
```

Y estandarizamos,

```
scaled.employee<-scale(employee)
```

2) Ahora creamos la matriz de correlación entre estas variables:

```
> cor<-cor(scaled.employee)
> round(cor, 2)
```

La que nos da los valores de correlación entre cada par de variables pertenecientes a la matriz. Luego usamos la siguiente función:

```
> findCorrelation(cor, cutoff = .60, verbose = TRUE)
```

Que nos devuelve el par de variables que contienen una correlación absoluta mayor o igual a 0.6, lo que nos da:

1. row 18(TotalWorkingYears) column 21(YearsAtCompany) with corr 0.628

2. row 21(YearsAtCompany) column 22(YearsinCurrentRole) with corr 0.759

3. row 9(JobLevel) column 11(MonthlyIncome) with corr 0.95

4. row 22(YearsInCurrentRole) column 24(YearsWithCurrManager) with corr 0.714

5. row 14(PercentSalaryHike) column 15(PerformanceRating) with corr 0.774

Las cuales son variables que a simple vista tienen una relación directa; el número total de años trabajados con los años en la empresa, los años en la empresa con los años en el rol actual, el nivel del trabajo con el salario, cantidad de años en el rol en conjunto con la cantidad de años con el mismo jefe y por último el salario relacionado a la efectividad del trabajador.

3) En esta sección utilizamos el método PCA, tomando la variable definida anteriormente como "*scaled.employee*" la cual ya ha sido estandarizada, por lo tanto definimos otra variable y usamos la función:

```
emp.pca<-prcomp(scaled.employee,scale=FALSE)
names(emp.pca)
```

"emp.pca" es ahora nuestra variable con PCA aplicado, por lo que procedemos a calcular los eigenvalores y varianzas de esta nueva matriz de componentes principales:

```
> #Eigenvalues
> eig<-(emp.pca$sdev)^2
> #Varianzas en %
> variance<-eig*100/sum(eig)
```

```
> #Varianzas acumulativas
> cumvar<-cumsum(variance)
> emp.pca.active<-data.frame(eig=eig,variance=variance,cumvariance=cumv
ar)
> head(emp.pca.active)
```

Lo que nos produce:

```
       eig  variance cumvariance
1 4.655937 19.399738    19.39974
2 1.836218  7.650910    27.05065
3 1.755621  7.315089    34.36574
4 1.211304  5.047100    39.41284
5 1.140048  4.750200    44.16304
6 1.081311  4.505462    48.66850
```

Como se puede observar, las dos componentes principales son las 2 varianzas mayores de la lista, también la variabilidad retenida por cada una de estas componentes se ve representada por el valor de los eigenvalores, entonces:

Componente principal 1:

$$S^2 = 19.399$$

$$Eigenvalor = 4.656$$

Componente Principal 2:

$$S^2 = 7.651$$

$$Eigenvalor = 1.836$$

Para visualizar esto graficamos las 2 primeras PC:

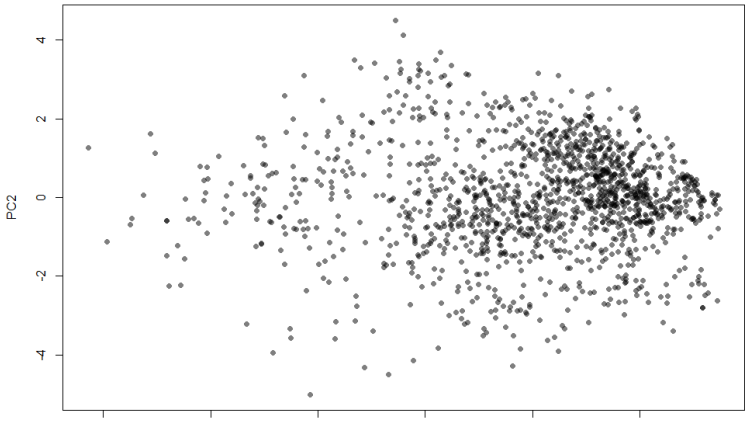

Figura 1. PC 1 vs PC2

También graficamos los loadings de las variables,

```
fviz_pca_var(emp.pca)
```

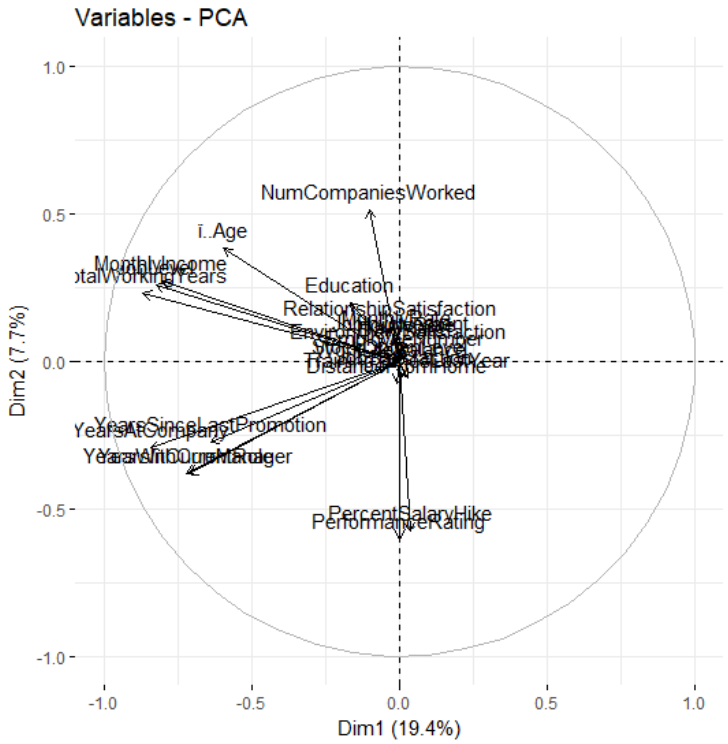

Figura 2. Loadings

Donde podemos observar las correlaciones de las variables con sus componentes principales, por lo que se muestra la proporción de variabilidad capturada por cada una de las PC.

4)

Primero se hace una prueba para encontrar el número de k óptimo usando la librería factoextra,

```
library(factoextra)
fviz_nbclust(scaled.employee,kmeans,method="gap_stat")
```

Lo que resulta en lo siguiente:

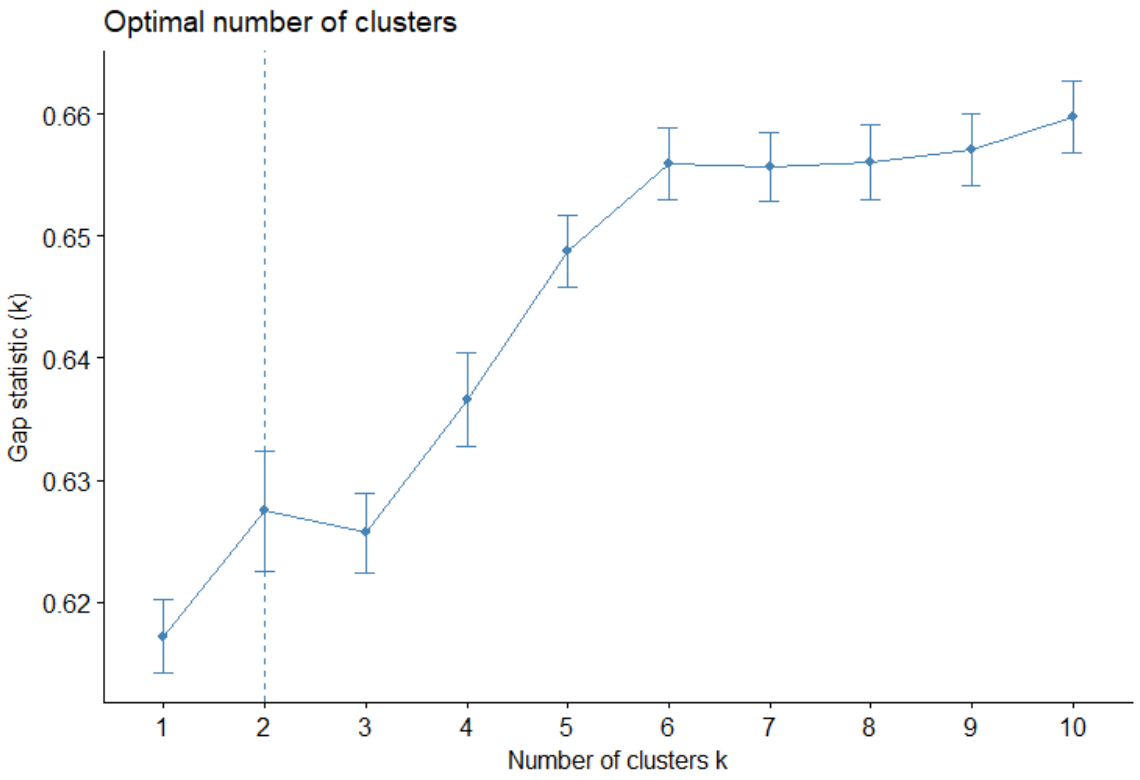

Figura 3 k óptimo Factoextra

Por lo que un posible valor de k óptimo según este método es de k = 2 clusters.

Ahora usaremos el método kmeans() en R para encontrar este k óptimo, para esto usamos el código:

```
mydata<-scaled.employee
wss<-(nrow(mydata)-1)*sum(apply(mydata,2,var))
for(i in 2:15) wss[i]<- sum(kmeans(mydata,centers=i)$withinss)
plot(1:15,wss,type="b",xlab="Number of Clusters",ylab="Within groups su
m of squares",main="Assessing the Optimal Number of Clusters with the E
lbow Method",pch=20,cex=2)
```

Lo que nos da:

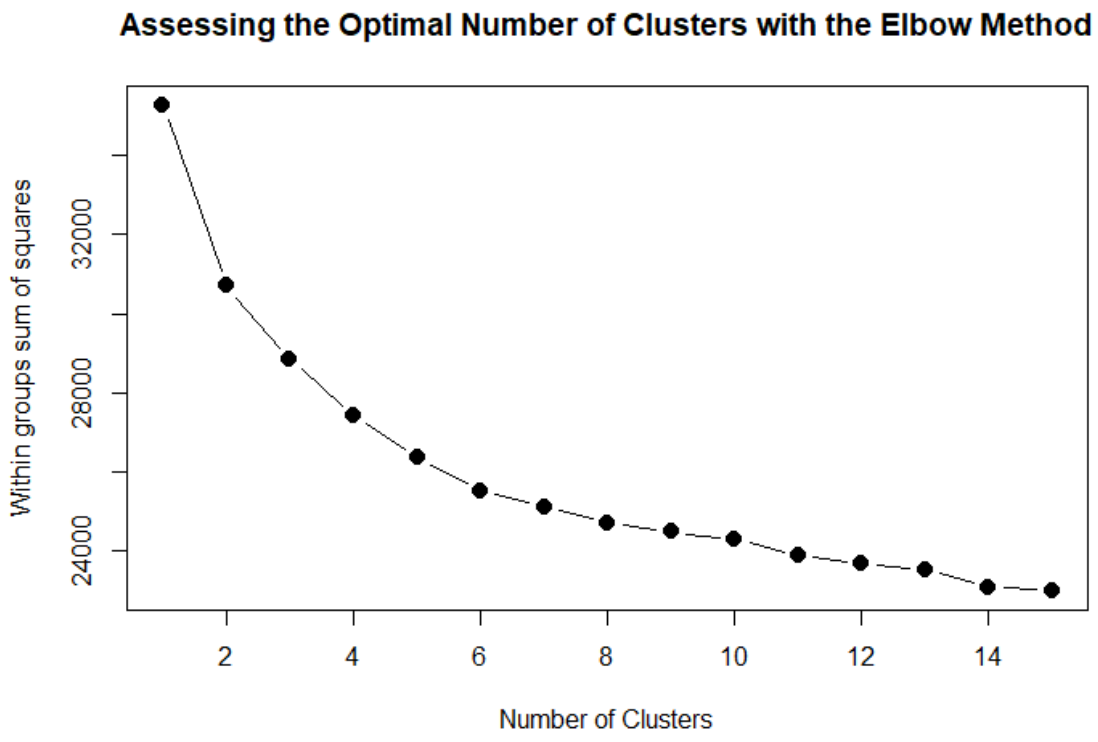

Figura 4. K óptimo Elbow Method

Lo que según el criterio de Elbow Method se puede observar que el quiebre ocurre en el número k = 2 de clusters, confirmando la primera teoría de que este era el número óptimo de clusters para este caso (se usó la matriz estandarizada de datos antes calculada).

5)

Se grafican los clusters generados por k-means utilizando los datos obtenidos por PCA en los pasos anteriores:

```
comp<-data.frame(emp.pca$x[,1:2])
plot(comp, col =(emp.cluster$cluster +1) , main="K-Means clusters vs PC
A", pch=20, cex=2)
```

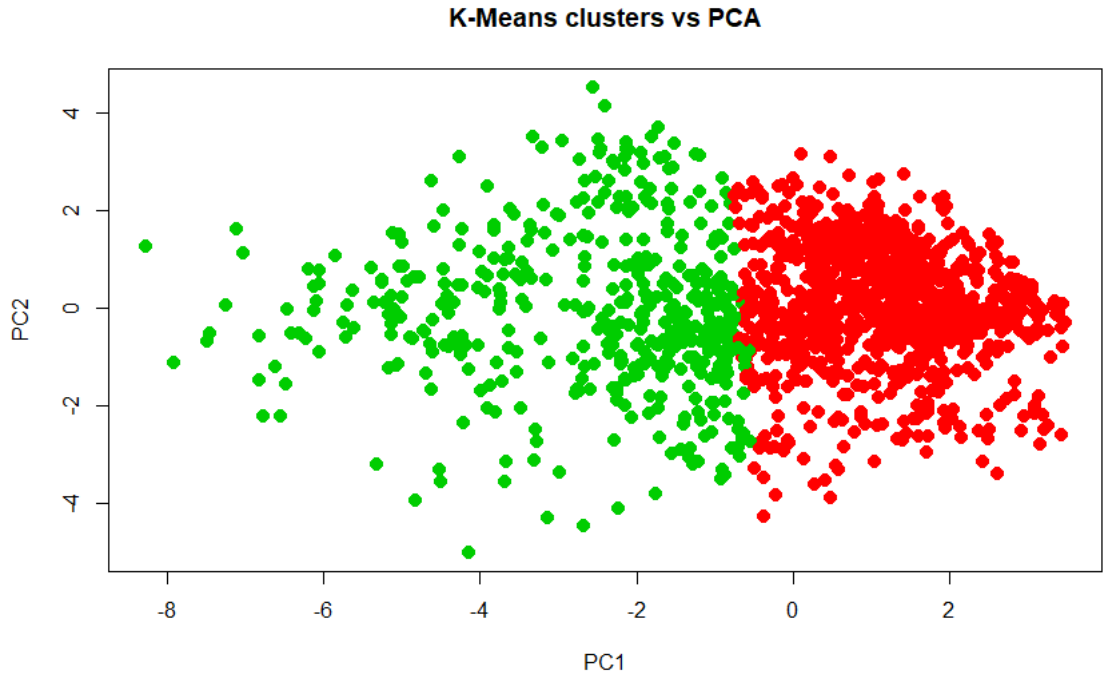

Figura 5. K-means vs PCA

6)

Para DBSCAN se puede aplicar una herramienta de R para encontrar directamente el radio (Eps) óptimo, análogo al elbow method de k-means. Su principio básico es el cálculo de las distancias promedio de todo punto a sus k vecinos más cercanos, para esto en R:

```
dbscan::kNNdistplot(scaled.employee,k=5)
> abline(h=4.9,lty=2)
```

Lo que nos devuelve le siguiente gráfico:

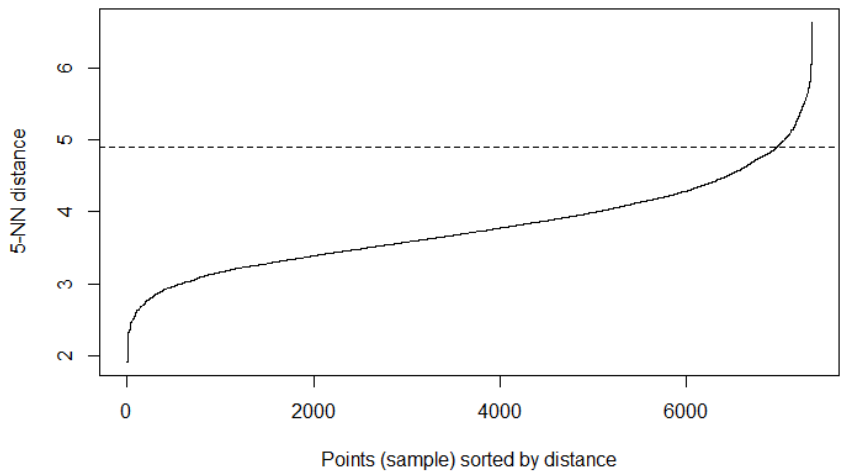

Figura 6. EPS óptimo

Como se puede observar, se puede deducir un punto de mayor cambio en el rango [4 , 4.9] por lo que "Eps" será probado dentro de estos límites.

Por lo que el óptimo es:

```
emp.dbscan<-fpc::dbscan(scaled.employee,eps=4,MinPts = 4 )
fviz_cluster(emp.dbscan, scaled.employee, stand = FALSE, ellipse = TRUE
, geom = "point")
```

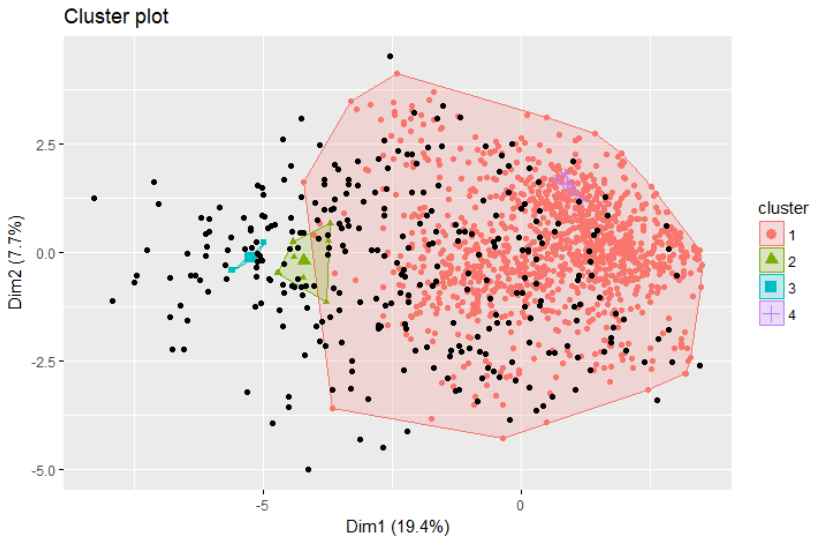

Figura 7. DBSCAN Plot

7)

Se procede a graficar los clusters obtenidos por DBSCAN utilizando los datos obtenidos por PCA:

```
plot(emp.dbscan,comp,main="DBSCAN vs PCA",frame=FALSE)
```

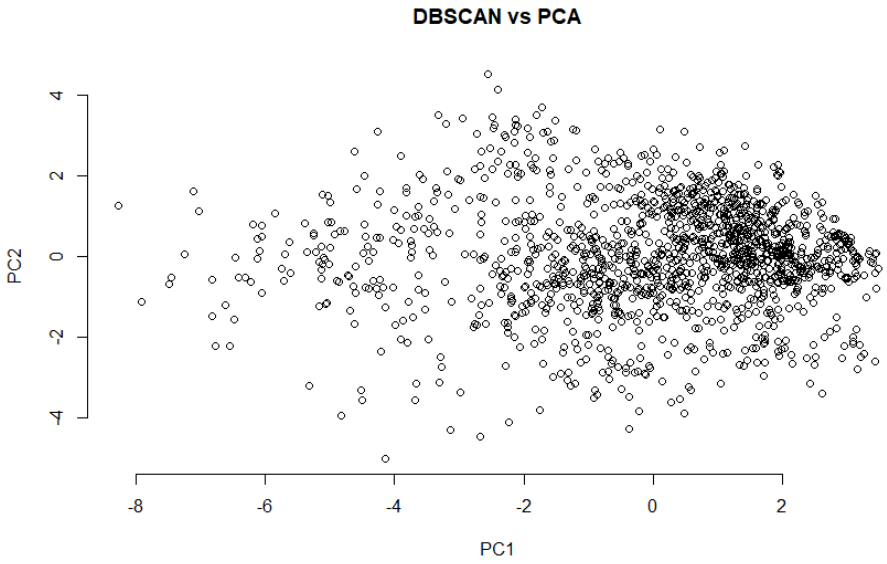

Figura 8. DBSCAN vs PCA

8)

Una vez calculados los óptimos de K-means y DBSCAN se han graficado sus formas de clusterización en comparación a los resultados obtenidos por PCA. Al hacer la comparación entre estos se puede observar que k-means hace un mejor trabajo que DBSCAN debido a la forma intrínseca del algoritmo, es decir, k-means mide directamente las distancias redefiniendo los puntos que pertenecen a cada nuevo cluster, pero en comparación DBSCAN se fija más directamente en la densidad de puntos en un cierto área, lo que lo hace inefectivo debido a la forma de los datos del data frame de empleados.

Esto se comprueba graficando las dos principales componentes entregadas por PCA, el gráfico de "K-means vs PCA" del inciso 5) entrega fácilmente una separación de la data, en cambio el gráfico "k-means vs DBSCAN" muestra de forma difusa por medio de densidades las separaciones de clusters.

9)

```
cluster.belong <- emp.cluster$cluster
belonging<-cluster.belong
employee2<-cbind(employee, belonging)
employee2.kmeans<-kmeans(employee2, centers=2, iter.max=100, nstart=20)
plot(employee2$belonging, employee2$MonthlyIncome)
```

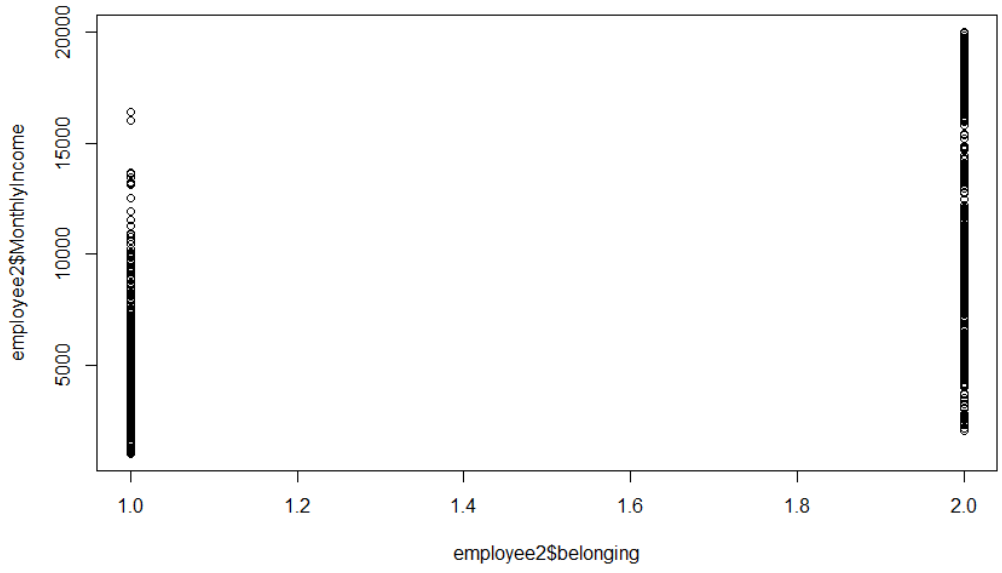

Figura 9. MonthlyIncome vs Cluster

Average Monthly Income Cluster I = $3.505,6

Average Monthly Income Cluster II = $5.130

10)

```
plot(employee2$JobRole, employee2$MonthlyIncome)
```

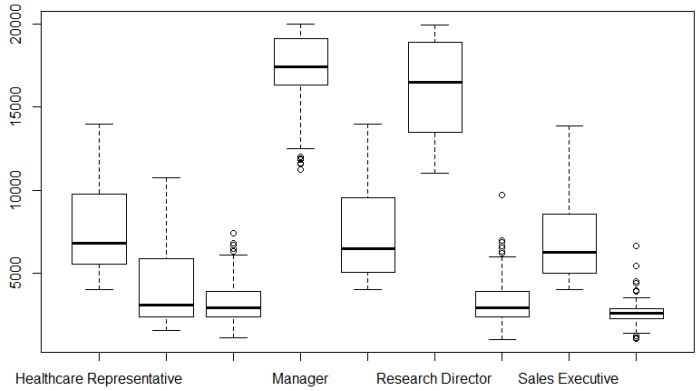

Figura 10. MonthlyIncome vs Jobrole

```
plot(employee2$JobRole, employee2$belonging, main="JobRole Cluster Belo
nging")
```

Figura 11. Jobrole vs Cluster

```
library(scatterplot3d)
> attach(mtcars)
```

```
> scatterplot3d(employee2$MonthlyIncome, employee2$belonging, employee2
$JobRole)
```

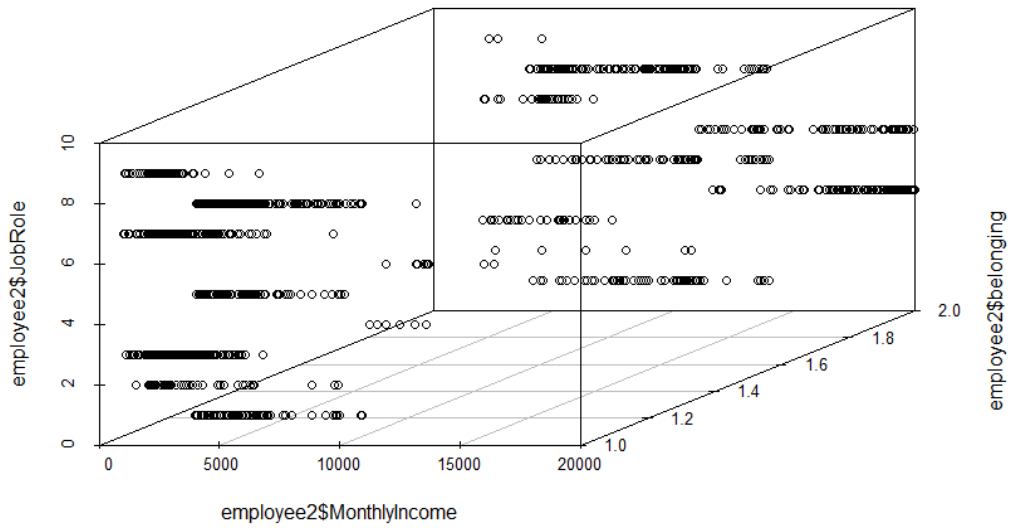

Figura 12. Jobrole vs MonthlyIncome vs Belonging(Cluster)

Como puede observarse, el rol de trabajo de "Manager" y "Research Director" que pertenecen al cluster II, son los mejores pagados y por lo tanto, definen un patrón de relación del cluster entre el rol del trabajo y sus sueldos.

11)

No es necesario separar los clusters por histograma debido a que el segundo cluster es fácil de observar dado a que este contiene pocos integrantes, en este caso:

```
hist(employee2$TotalWorkingYears)
```

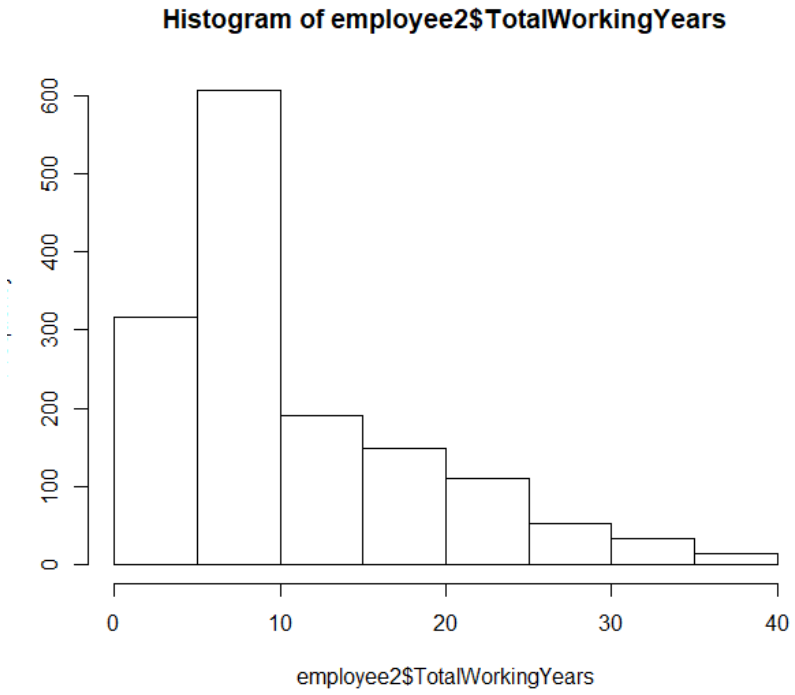

Figura 13. Histograma TotalWorkingYears

Se puede concluir que el segundo cluster (el cual tiene los mayores sueldos) es el que también contiene a los trabajadores con la mayor cantidad de años de trabajo en la empresa, por lo que si existe relación entre el tiempo en la empresa y los sueldos ganados.

*En las instrucciones se habla de horas de trabajo pero se especifíca la variable "TotalWorkingYears" por lo que se trabajo con esta y bajo ese contexto.