Solucionario Clustering

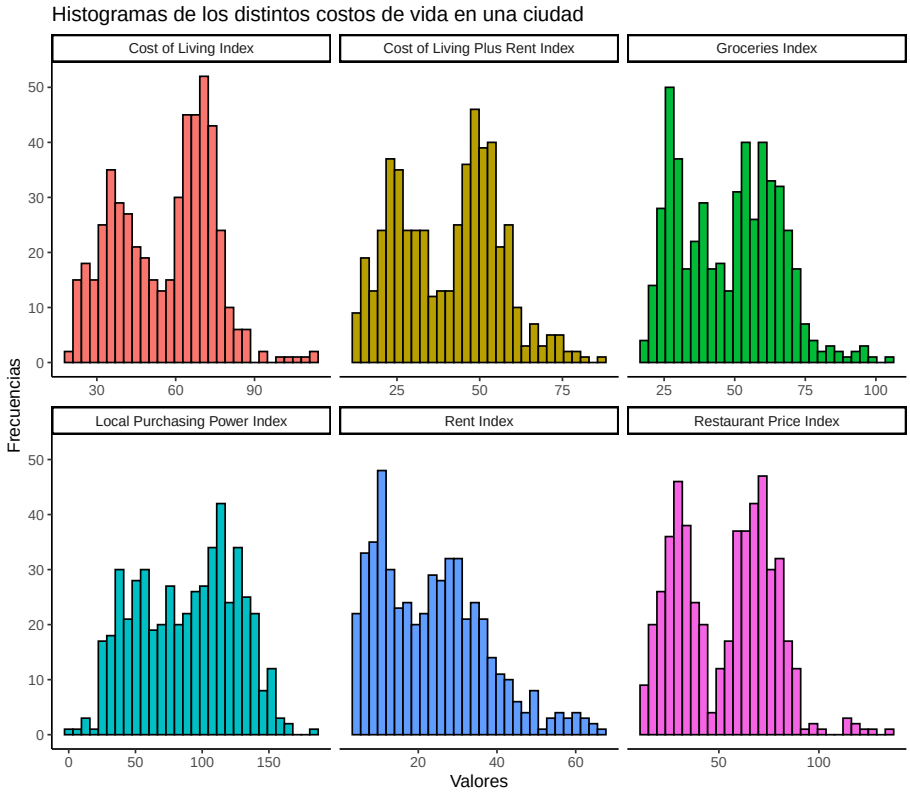
- **Analizar previamente las variables a utilizar en k-means**
  - Comentar sobre la naturaleza y formato de las variables
  - Analizar el ID (nombre de la ciudad) verificar que no se repite ninguna
  - Comentar sobre que se utilizarán todos los índices, son variables continuas
  - Comentar sobre las escalas de medición, rango de las variables, sus valores mínimos y máximos
  - Comentar sobre las diferencias de variabilidad
- **Considerar tratamientos previos a los datos**
  - Especificar la estandarización por distintos rangos y variabilidades entre los índices
  - Utilizar algún método para determinar outliers
  - Identificar outliers y calcular la tasa de outliers presente en la base de datos, comentar sobre ésta
  - Proponer metodología, especificar por qué quita o no quita los outliers, si los quita entonces especificar cómo los clasificará
- **Probar distintos valores de k y proponer en base a criterios explícitos**
  - Utilizar semilla 2019 del enunciado
  - Probar una grilla de valores lo suficientemente grande como para concluir
  - Realizar el gráfico de sedimentación y comentar sobre todos aquellos valores en los que se observa un cambio de pendiente perceptible en base a *betweenss* y *tot.withinss*
  - Especificar cuál valor de k utilizará y por qué
- **Realizar la clusterización y comentar detalles importantes a la hora de analizar la segmentación obtenida**
  - Importante comentar sobre los tamaños de los clusters, si alguno es demasiado grande o pequeño, es relevante de estudiar
  - Obtener centroides, de ser posible comentar sobre caracterizaciones de cada clusters
- **Pregunta 2 posterior clusterización**
  - Añadir la clusterización final a la base de datos
  - Identificar a qué clúster pertence `Santiago, Chile` no hacerlo manualmente

– Identificar otras ciudades que pertenecen a dicho clúster (entregar 3)
– Concluir, que dentro de clúster se espera homogeneidad y por lo tanto, en base a la clusterización encontrada, se espera que dichas ciudades sean similares en términos de índices de costo de vida

Código

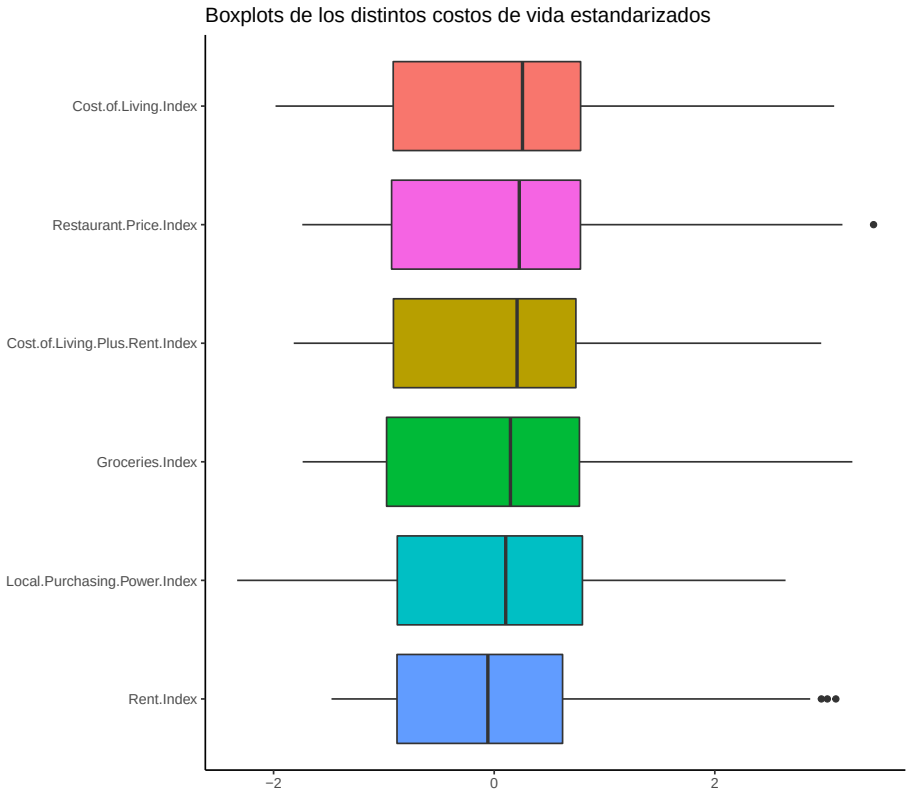
```
library(readr)
Cost_of_living <- read_delim("/cloud/project/Cost of living.csv",
                             ";", escape_double = FALSE, trim_ws = TRUE)

#Favor, antes de trabajar con la base de datos verificar que se ha cargado correctamente


#Vista de la data

names(Cost_of_living)

## [1] "City"                          "Cost of Living Index"
## [3] "Rent Index"                    "Cost of Living Plus Rent Index"
## [5] "Groceries Index"               "Restaurant Price Index"
## [7] "Local Purchasing Power Index"

print(Cost_of_living)

## # A tibble: 518 x 7
##    City  `Cost of Living~ `Rent Index` `Cost of Living~ `Groceries Inde~
##    <chr>            <dbl>        <dbl>            <dbl>            <dbl>
##  1 Reyk~            113.          57.4             86.4             98.5
##  2 Luga~            112.          49.6             82.4            105.
##  3 Stav~            111           38.8             76.8             96.0
##  4 Oslo~            107.          48.6             79.4             93.8
##  5 Berg~            103.          39.4             72.8             87.0
##  6 Tron~             99.7         39.9             71.4             82.3
##  7 Hono~             93.7         64.0             79.6             96.3
##  8 Anch~             93.2         39.4             67.8             96.7
##  9 Sant~             88.3         56.7             73.3             84.4
## 10 Cope~             87.9         47.8             68.9             64.6
## # ... with 508 more rows, and 2 more variables: `Restaurant Price Index` <dbl>,
## #   `Local Purchasing Power Index` <dbl>

#Nos aseguramos de que las variables se lean en el formato adecuado:

str(Cost_of_living)

## tibble [518 x 7] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ City                          : chr [1:518] "Reykjavik, Iceland" "Lugano, Switzerland" "Stavanger, Norway" "Oslo, Norway" .
##  $ Cost of Living Index          : num [1:518] 113 112 111 107 103 ...
##  $ Rent Index                    : num [1:518] 57.4 49.6 38.9 48.6 39.4 ...
##  $ Cost of Living Plus Rent Index: num [1:518] 86.5 82.4 76.8 79.4 72.8 ...
##  $ Groceries Index               : num [1:518] 98.5 104.8 96 93.8 87 ...
##  $ Restaurant Price Index        : num [1:518] 128 121 135 114 116 ...
##  $ Local Purchasing Power Index  : num [1:518] 94.2 131.3 112.6 104.3 108.2 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   City = col_character(),
##   ..   `Cost of Living Index` = col_double(),
```

```
##   ..   `Rent Index` = col_double(),
##   ..   `Cost of Living Plus Rent Index` = col_double(),
##   ..   `Groceries Index` = col_double(),
##   ..   `Restaurant Price Index` = col_double(),
##   ..   `Local Purchasing Power Index` = col_double()
##   .. )

#Sólo la variable City se lee como caracter, las demás variables
# son de tipo numeric


dim(Cost_of_living)  #518 ciudades, ¿son distintas?

## [1] 518   7

table(table(Cost_of_living$City))  #Son todas las ciudades distintas

##
##   1
## 518

#Análisis de las variables

library(ggplot2)
library(dplyr)
library(tidyverse)


##Histogramas

Cost_of_living %>%
  gather(Attributes, value, 2:7) %>%
  ggplot(aes(x=value, fill=Attributes)) +
  geom_histogram(colour="black", show.legend=FALSE,bins=30) +
  facet_wrap(~Attributes, scales="free_x") +
  labs(x="Valores", y="Frecuencias",
       title="Histogramas de los distintos costos de vida en una ciudad") +
  theme_classic()
```



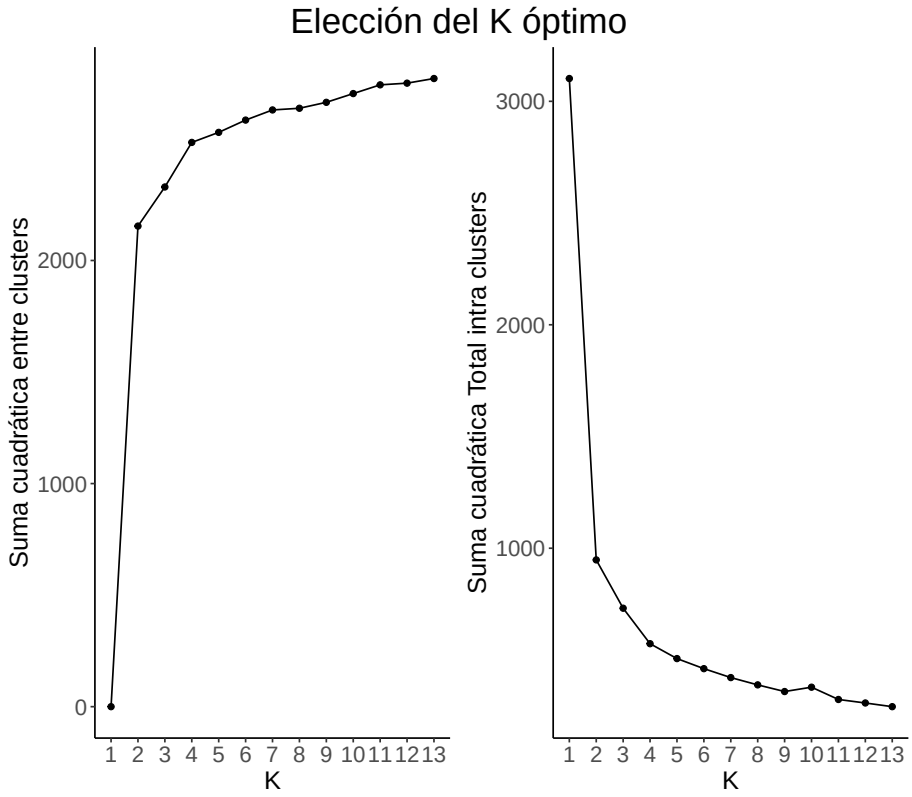

```
#Si bien, todos los índices están en una misma escala de medición
#respecto a Nueva york, es importante notar que hay variables que alcanzan
# valores más altos que otros, por ejemplo, Local Purchasing Power Index, el poder
# adquisitivo dado un sueldo promedio alcanza valores mayores  a 150

#Alcanzan distintos valores las variables, quizás sea necesario estandarizar

#Variabilidad de las variables:

diag(var(Cost_of_living[,-1]))

##           Cost of Living Index                     Rent Index
##                       339.4757                       185.7787
## Cost of Living Plus Rent Index                Groceries Index
##                       240.0642                       303.6500
##         Restaurant Price Index   Local Purchasing Power Index
##                       560.5626                      1366.1630

#Notar que las variabilidades cambian bastante a pesar de que se encuentran
# en una misma escala. La variable Local Purchasing Power Index presenta una
# variabilidad de 1371, versos 270 (variabilidad de Rent Index) por ejemplo

#Es necesario estandarizar

stand<-scale(Cost_of_living[,-1])

stand<-data.frame(stand)

stand %>%
```

```
  gather(Attributes, values) %>%
  ggplot(aes(x=reorder(Attributes, values, FUN=median), y=values, fill=Attributes)) +
  geom_boxplot(show.legend=FALSE) +
  labs(title="Boxplots de los distintos costos de vida estandarizados") +
  theme_classic() +
  theme(axis.title.y=element_blank(),
        axis.title.x=element_blank())  +
  coord_flip()
```




```
#Solo dos variables presentan outliers

#Restaurant Price Index

#Rent Index


#Identificamos aquellos outliers:

outrest<-which(stand$Restaurant.Price.Index < boxplot(stand$Restaurant.Price.Index,plot=FALSE)$stats[1] | stand$Restaurant.Price.

outrent<-which(stand$Rent.Index < boxplot(stand$Rent.Index,plot=FALSE)$stats[1] | stand$Rent.Index > boxplot(stand$Rent.Index,plo


outliers<-unique(c(outrest,outrent))

(length(outliers)/nrow(Cost_of_living))*100

## [1] 0.7722008
```

```r
#Tasa porcentual de observaciones outliers es bajísima


#Existen dos opciones:


#a) La primera es continuar con el clustering incluyendo
#outliers, esto especificando que la tasa es bastante baja
#o que los outliers no se alejan demasiado de la masa de los datos
# no son outliers abismantes.

#b) Quitar los outliers, bajo algún criterio que especifique, o
#proponer metodología.


#K-means

###Eleccion del k optimo


n<-13        #Cantidad de valores k a probar

#Pueden usar distintas grillas, pero a partir de 12 aproximadamente
# no se observan mayores diferencias

bss <- rep(NA,n)
wss <- rep(NA,n)

set.seed(2019)

for(i in 1:n){
  bss[i] <- kmeans(stand, centers=i)$betweenss
  wss[i] <- kmeans(stand, centers=i)$tot.withinss

}


#Graficas

betweenplot <- qplot(1:n, bss, geom=c("point", "line"),
                     xlab="K", ylab="Suma cuadrática entre clusters") +
  scale_x_continuous(breaks=seq(0, n, 1)) +
  theme_classic()+theme(axis.text.y = element_text(size=14),axis.title.y = element_text(size=16),
                        axis.text.x = element_text(size=14),axis.title.x = element_text(size=16))


whithinplot <- qplot(1:n, wss, geom=c("point", "line"),
                     xlab="K", ylab="Suma cuadrática Total intra clusters") +
  scale_x_continuous(breaks=seq(0, n, 1)) +
  theme_classic()+theme(axis.text.y = element_text(size=14),axis.title.y = element_text(size=16),
                        axis.text.x = element_text(size=14),axis.title.x = element_text(size=16))


library(ggpubr)
plot<-ggarrange(betweenplot, whithinplot, ncol=2)

annotate_figure(plot,top=text_grob("Elección del K óptimo",size=22))
```




```
#El valor más sugerente pudiera ser k=2, pero en k=4 también
# se observa una diferencia que pudiera ser importante

#La elección del k dependerá de la justificación que realicen, cualquiera de los 2
#pudiera usarse


#El procedimiento con K=2 sería análogo

# K = 4

set.seed(2019)

cluster<-kmeans(stand, centers=4)

cluster$cluster
```

```
##   [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3 1 3 3 1 3 3 3 3 3 1 3 1 3 1 3 1 3 1 3
##  [38] 1 3 3 1 3 1 3 3 1 1 1 1 3 3 1 3 1 1 3 1 1 1 1 1 3 3 1 1 1 1 3 1 3 1 3 3 1
##  [75] 1 1 1 1 3 3 1 3 1 1 1 1 3 1 1 1 3 1 1 3 3 1 1 1 1 3 1 1 1 3 3 3 1 1 1 1 1
## [112] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 2 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1
## [149] 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1
## [186] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 1 1
## [223] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 2 1 1 1 1 1 3 1 2 1 2 1 2 1 1
## [260] 1 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [297] 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 4 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [334] 2 2 2 2 2 2 4 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 4 2 2 2 4 4 2 4 4 2 4 4 2 2 4
## [371] 4 4 4 4 4 2 4 4 2 4 4 2 4 2 4 2 4 4 4 4 4 4 4 4 4 4 4 4 4 2 4 4 4 4 4 4 4 4
## [408] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
```

```
## [445] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
## [482] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
```

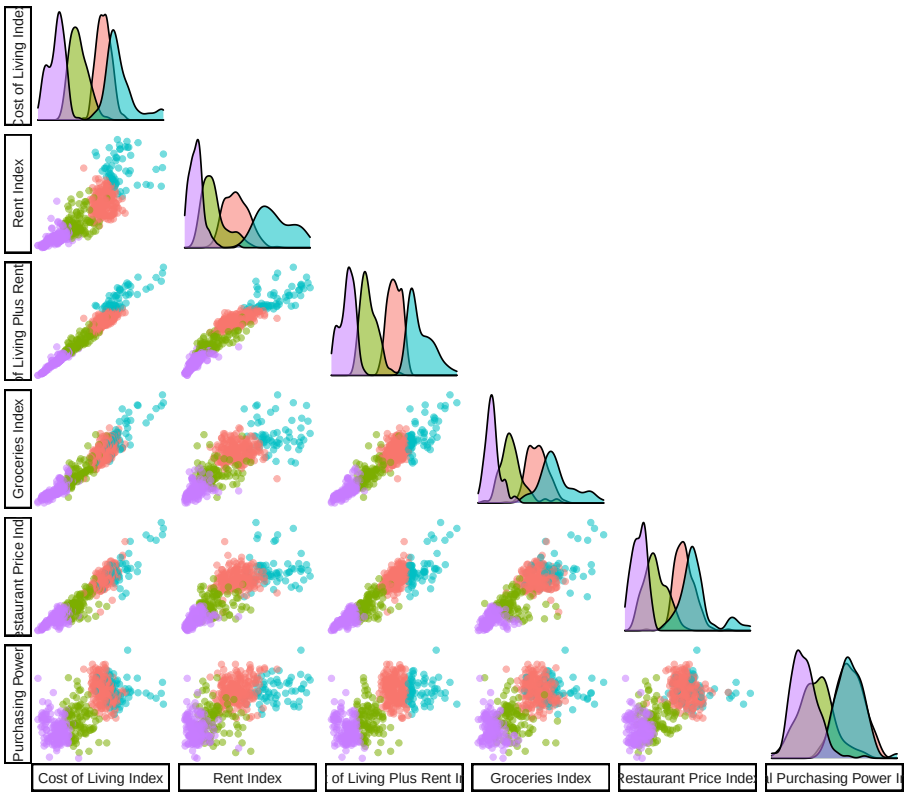
```
cluster$centers  #Centroides
```

```
##   Cost.of.Living.Index Rent.Index Cost.of.Living.Plus.Rent.Index
## 1            0.6783685  0.4152691                       0.5977543
## 2           -0.3351911 -0.4028340                      -0.3776649
## 3            1.3239655  1.8231889                       1.5883301
## 4           -1.2432963 -1.0467461                      -1.2144914
##   Groceries.Index Restaurant.Price.Index Local.Purchasing.Power.Index
## 1       0.6206922              0.7160619                     0.7783268
## 2      -0.3904260             -0.4782995                    -0.5568452
## 3       1.3938170              1.1950434                     0.7905802
## 4      -1.1580099             -1.1495269                    -1.0182925
```

```
cluster$size  #No hay ningun cluster demasiado pequeño en
```

```
## [1] 206  98  62 152
```

```
#comparacion a otro



library(GGally)

ggpairs(cbind(Cost_of_living[-1], Cluster=as.factor(cluster$cluster)),
        columns=1:6, aes(colour=Cluster, alpha=0.5),
        lower=list(continuous="points"),
        upper=list(continuous="blank"),
        axisLabels="none", switch="both") +
  theme_classic()
```




```
#Es interesante observar cómo se diferencias las variables
#por clustering


#Se agrega la clusterizacion encontrada:

Cost_of_living$cluster<-cluster$cluster


### Santiago, Chile

which(Cost_of_living$City=="Santiago, Chile")

## [1] 310

#Observación 310 corresponde a Chile

Cost_of_living$cluster[310]  #Pertenece al cluster 2

## [1] 2

Cost_of_living$City[which(Cost_of_living$cluster=="2")]

##  [1] "Male, Maldives"                    "Naples, Italy"
##  [3] "Kingston, Jamaica"                 "Taichung, Taiwan"
##  [5] "Palermo, Italy"                    "Nicosia, Cyprus"
##  [7] "Beirut, Lebanon"                   "Athens, Greece"
##  [9] "Port of Spain, Trinidad And Tobago" "Montevideo, Uruguay"
```

```
## [11] "Zaragoza (Saragossa), Spain"        "Thessaloniki, Greece"
## [13] "San Jose, Costa Rica"               "Limassol, Cyprus"
## [15] "Accra, Ghana"                       "Kaohsiung, Taiwan"
## [17] "Ljubljana, Slovenia"                "Valencia, Spain"
## [19] "Manama, Bahrain"                    "Tallinn, Estonia"
## [21] "Panama City, Panama"                "Amman, Jordan"
## [23] "Seville (Sevilla), Spain"           "Larnaca, Cyprus"
## [25] "Lisbon, Portugal"                   "Alicante, Spain"
## [27] "Split, Croatia"                     "Bangkok, Thailand"
## [29] "Sharjah, United Arab Emirates"      "Rijeka, Croatia"
## [31] "Zagreb, Croatia"                    "Makati, Philippines"
## [33] "Coimbra, Portugal"                  "Muscat, Oman"
## [35] "Al Khobar, Saudi Arabia"            "Funchal, Portugal"
## [37] "Riga, Latvia"                       "Porto, Portugal"
## [39] "Shanghai, China"                    "Granada, Spain"
## [41] "Malaga, Spain"                      "Santiago, Chile"
## [43] "Jeddah (Jiddah), Saudi Arabia"      "Addis Ababa, Ethiopia"
## [45] "Tartu, Estonia"                     "Quito, Ecuador"
## [47] "Maribor, Slovenia"                  "Santa Cruz de Tenerife, Spain"
## [49] "Riyadh, Saudi Arabia"               "Bratislava, Slovakia"
## [51] "Braga, Portugal"                    "Ad Dammam, Saudi Arabia"
## [53] "Vilnius, Lithuania"                 "Prague, Czech Republic"
## [55] "Bandar Seri Begawan, Brunei"        "Harare, Zimbabwe"
## [57] "Las Palmas de Gran Canaria, Spain"  "San Salvador, El Salvador"
## [59] "Guayaquil, Ecuador"                 "Baghdad, Iraq"
## [61] "Phuket, Thailand"                   "Klaipeda, Lithuania"
## [63] "Kosice, Slovakia"                   "Pretoria, South Africa"
## [65] "Santo Domingo, Dominican Republic"  "Kaunas, Lithuania"
## [67] "Windhoek, Namibia"                  "Phnom Penh, Cambodia"
## [69] "Johannesburg, South Africa"         "Brno, Czech Republic"
## [71] "Moscow, Russia"                     "Selangor, Malaysia"
## [73] "Campinas, Brazil"                   "Guatemala City, Guatemala"
## [75] "Dar es Salaam, Tanzania"            "Petaling Jaya, Malaysia"
## [77] "Sao Paulo, Brazil"                  "Rio de Janeiro, Brazil"
## [79] "Brasilia, Brazil"                   "Penang, Malaysia"
## [81] "Beijing, China"                     "Budapest, Hungary"
## [83] "Gdansk, Poland"                     "Johor Bahru, Malaysia"
## [85] "Pattaya, Thailand"                  "Kuala Lumpur, Malaysia"
## [87] "Warsaw, Poland"                     "Shenzhen, China"
## [89] "Peterborough, United Kingdom"       "Cape Town, South Africa"
## [91] "Olomouc, Czech Republic"            "Gdynia, Poland"
## [93] "Durban, South Africa"               "Guangzhou, China"
## [95] "Wroclaw, Poland"                    "Krakow (Cracow), Poland"
## [97] "Saint Petersburg, Russia"           "Suzhou, China"

#Otras ciudades que pertenecen al mismo cluster son:

#Rio de Janeiro, Brazil
#Moscow, Russia
#Shanghai, China
```