FACULTAD DE INGENIERIA
Universidad de Buenos Aires

# CIENCIA DE DATOS PARA LA TOMA DE DECISIONES

## TRABAJO DE APLICACIÓN

## CLUSTERIZACIÓN DE VIAJES ECOBICI

1er cuatrimestre 2020


| Alumno | Padrón |
|---|---|
| Pedro Mordeglia | 100883 |
| Blas Casado | 101082 |

# RESUMEN EJECUTIVO

Este trabajo consistió en la segmentación de los distintos viajes realizados en el año 2018 y la metodología empleada utilizando como variables de segmentación las propias del viaje, las características de la estación y las condiciones climáticas al momento de realizar el viaje.

Para ello, abarcamos los siguientes temas:

- Distancia de Manhattan
- Clusterización no jerárquica
    - Kmeans
    - CLARA (no visto en la materia)
- Clusterización jerárquica
- Análisis de correspondencias

A partir de lo anterior se pudo clasificar a los viajes en diez grupos distintos usando ambos algoritmos de clasificación no supervisada (y no jerárquica) realizando un análisis previo para determinar el número de grupos óptimo. Finalmente, se usó la herramienta Power BI para la visualización y el análisis posterior de los grupos.

# INTRODUCCIÓN - OBJETIVO

El objetivo del presente trabajo es realizar una segmentación de los viajes realizados en bicicleta durante el año 2018 a partir de los datos recolectados del sistema EcoBicis de CABA (2.8 millones de viajes). El propósito de esto es tipificar los viajes de bicicletas públicas dentro de CABA que posteriormente podría ser utilizado como base de análisis para el sistema actual de Ecobicis (modificado a inicios del año 2019), para la localización de nuevas estaciones, modo de reposición de las bicicletas, incidencia del clima en los viajes o para un análisis sociocultural y turístico en caso de disponer de variables adicionales.

Se utilizaron los algoritmos de K-means y CLARA, propios de la clusterización no supervisada. Ambos siguen un método de optimización de agrupamientos definiendo a priori un número "k" de grupos y distribuyendo las "semillas" de forma aleatoria en el espacio. A partir de ese momento el algoritmo procede a asignar todos los puntos a la semilla más cercana, formando los grupos y reacomodando los centros. Esto lo hace de forma iterativa hasta que el algoritmo converge y finaliza.

Respecto a las variables utilizadas, se pueden clasificar en variables temporales (que indican la fecha en la que se realizó el viaje), variables del viaje (distancia, velocidad, etc.), variables de las estaciones que intervienen en el viaje y variables de las condiciones climáticas al momento del viaje. Luego, se definió una base de segmentación para el análisis y poder así encontrar grupos de viajes a partir de las mismas.

# METODOLOGÍA

## Algoritmos de clusterización

El eje principal del algoritmo K-means se basa en calcular para cada iteración, el centro de cada grupo usando una medida de distancia previamente establecida. Una vez calculados todos los centros, se reasigna cada punto al grupo más cercano y se calculan los centros nuevamente. Si bien esto podría parecer sencillo, lo hace muy sensible al algoritmo a los outliers, afectando así a la asignación de puntos a cada grupo. La otra limitación del algoritmo K-means es que no es posible usarlo para variables categóricas, dado que no podría calcular la media para variables categóricas, ni tendría una medida de distancia para las mismas.

Una solución a esto la provee el algoritmo PAM (partitioning around medoids), que busca aquellos centroides pertenecientes a la base de datos que representan mejor al grupo, es decir, no define los centros de cada grupo con el promedio de las distancias de todos ellos, sino que define al centroide como a aquel punto cuya distancia al resto de los puntos del grupo es la menor de todos los puntos pertenecientes al grupo, y luego reasigna cada punto de las observaciones al grupo correspondiente. Al trabajar con centroides, no es necesario calcular "una media", y por lo tanto es posible trabajar con variables categóricas también. El objetivo es entonces, encontrar K puntos representativos dentro de las observaciones que minimicen la suma de las diferencias a su objeto más representativo.

Si bien el algoritmo PAM presenta una solución robusta frente a la sensibilidad del algoritmo K-means, tiene como contrapartida que requiere mayor memoria y tiempo computacional que este último dado que calcula la suma de las distancias de para cada punto perteneciente al grupo y en base a eso define el centroide. El algoritmo CLARA (clustering for large applications) avanza un paso más en esta dirección evitando aplicar el algoritmo en todos los datos, sino que lo corre para una muestra, generando así un grupo óptimo de centroides. Una vez hecho esto, asigna el resto de los

puntos al grupo correspondiente. CLARA repite este proceso un número prefijado de veces de modo de minimizar el "bias" de la muestra, quedándose con el mejor resultado.

### Determinación del número óptimo de clusters

Dado que la metodología de aplicación consiste en una clasificación no supervisada, esto implica que no se conoce a ciencia cierta en cuantos grupos se puede dividir la población. Sin embargo, hay diversos métodos que permiten determinar el número óptimo de clusters en los que dividir una muestra. El método utilizado en el presente trabajo es el de la silueta promedio (average silhouette width), que explicaremos a continuación.

Este método consiste en calcular para cada punto "(i)" la distancia promedio a todos los puntos que pertenecen al mismo clúster "a(i)" y la distancia promedio a todos los puntos que pertenecen al clúster más cercano "b(i)". Es decir, para cada punto se calculan ambas distancias. Seguido a esto, se calcula la silueta del punto en cuestión con la siguiente expresión.

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

De la expresión anterior se puede observar que –1 < s(i) < 1 y que si s(i)>0, el método considera que el punto se clasificó bien y sino no. En un caso de pocas dimensiones y grupos muy homogéneos y heterogéneos entre sí se espera que la silueta promedio sea de 1, cosa que en la práctica no sucede y es mucho menor, del mismo modo que no todos los s(i) son mayores que cero.

Por último, para determinar el número óptimo de clusters se calcula la silueta para distintos números de grupos (ej: 1 a 50) y se toma como óptimo aquel que dé como resultado la mayor silueta de todos.

## DESARROLLO DEL ANÁLISIS

### Datasets iniciales

Inicialmente se dispuso de cuatro "data sets", los cuales contenían las siguientes variables:

1. **Viajes del sistema Ecobici en el año 2018.** Fuente: Gobierno de la ciudad de Buenos Aires (luego de la depuración del mismo, quedaron 2.5 M viajes para analizar).
   - → fecha_origen_recorrido: fecha y hora de inicio del viaje.
   - → id_estación_origen: número de estación desde la cual se inició el viaje.
   - → id_estación_destino: número de estación en la cual se finalizó el viaje.
   - → long_estación_origen: coordenada longitudinal de la estación de origen.
   - → lat_estación_origen: coordenada latitudinal de la estación de origen.
   - → long_estación_destino: coordenada longitudinal de la estación de destino.
   - → lat_estación_destino: coordenada latitudinal de la estación de destino.
   - → Hora_inicio_recorrido: hora en la que se inició el viaje.
   - → Hora_fin_recorrido: hora en la que se terminó el viaje.
   - → duración_recorrido: tiempo en "HS:MIN:SEG" de duración del viaje.

2. **Histórico 2018 de las condiciones meteorológicas.** Fuente: Aeroparque y Central de Buenos Aires.
   - → FECHA: fecha de la observación como "dd/mm/aaaa"

- → HORA [HOA]:  hora a la que se tomó la muestra como “hh”
- → TEMP [°C]:  temperatura (en °C).
- → HUM [%]: humedad relativa (en %).
- → PNM [hPa]: presión (en hPa).
- → FF [km/hr]: velocidad del viento (en Km/h).

Para cada fecha y hora se tenía un valor de cada parámetro.

3. **Histórico 2018 de las condiciones meteorológicas.** Fuente: WindGurú.
   - → FECHA: fecha de la observación como “dd/mm/aaaa”
   - → HORA [HOA]: hora a la que se tomó la muestra como “hh”. Intervalos de tres horas.
   - → FF [nudos]: velocidad del viento (en nudos) medida en ese momento.
   - → Lluvia (mm): cantidad de milímetros de agua que hubo de precipitaciones.

   Para cada fecha y hora se tenía un valor de cada parámetro.

4. **Feriados en Argentina en el año 2018.** Fuente: Diario La Nación

Se obtuvieron valores de los feriados y días no laborables de ese año.

## Definición y transformación de variables

De estos cuatro Datasets finales, se pudo observar que algunas hacían referencia a:

- **A la fecha del viaje:** Fecha_origen_recorrido (vn), Hora_inicio_recorrido (vn), Día no laborable (vn)
- **Al viaje en sí:** Id_estación_origen (vn), Id_estación_destino (vn), Lat_estación_origen (vc) , Long_estación_origen (vc), Hora_inico_recorrido (vn), Hora_fin_recorrido (vn)
- **A las estaciones de inicio y fin:** Id_estación_origen (vn), Id_estación_destino (vn)
- **Al clima al momento de iniciar el viaje:** Temperatura (vc), Presión (vc), Humedad (vc), Precipitaciones (vc), Viento (vc)

Referencia: vn: variable nominal, vo: variable ordinal, vc: variable continua

A partir de estas variables, se definieron otras que explicaremos a continuación, separadas en distintos grupos:

### Variables temporales:

A la fecha origen del recorrido se la separó en dos variables Mes y Día además de agregar una variable llamada día de la semana (Lunes-Domingo) y otra llamada Día no laborable (incluye feriados, y días de fin de semana)

Dado que en el caso de análisis no hay un ordenamiento entre fechas (1/12/2018 no es mayor que el 16/5/2018) ni se podía representar la continuidad dado que la misma estaba dividida en meses, días y horas para segmentar con variables más influyentes.

### Variables del viaje:

#### *Distancia Manhattan*

Para calcular la distancia recorrida durante el viaje, es decir desde la estación de origen a la estación de destino, se realizó una aproximación con la distancia Manhattan. A pesar de que el diseño urbano de las manzanas de la Ciudad de Buenos Aires no se asemeja a los casos en que se utiliza este tipo de distancia (es decir en forma de grilla), se decidió utilizar de todas formas siendo esta una mejor opción que la distancia euclídea (se cortarían a las manzanas por la mitad) y notando buenos resultados al calcular la distancia real de distintos viajes. Las variables que tuvimos cuenta fueron las coordenadas longitudinales y latitudinales de la estación de origen y destino (long_estación_origen, lat_estación_origen, long_estación_destino y lat_estación_destino). Se calculó las diferencias entre las coordenadas longitudinales y latitudinales entre estaciones, se ralizó las conversiones a metros y se sumaron ambas distancias para llegar al valor de distancia total del recorrido. A continuación, se presentan las fórmulas:

$$Distancia\ latitudinal = \left(\mathrm{lat}_{\mathrm{estación}_{\mathrm{origen}}} - \mathrm{lat}_{\mathrm{estación}_{\mathrm{destino}}}\right) * 0.92\ \frac{\mathrm{km}}{latitud}$$

$$Distancia\ longitudinal = \left(\mathrm{long}_{\mathrm{estación}_{\mathrm{origen}}} - \mathrm{long}_{\mathrm{estación}_{\mathrm{destino}}}\right) * 1.11\ \frac{\mathrm{km}}{longitud}$$

$$Distancia\ total = Distancia\ latitudinal + Distancia\ longitudinal$$

Se puede observar que la coordenada latitudinal y longitudinal no coinciden, esto se debe a la curvatura de la tierra, y es por esto que usamos los valores distintos para calcular la distancia de una forma más exacta.

Además, se calculó el tiempo del viaje con las horas, minutos y segundos de inicio y de fin del recorrido y al dividir la distancia por el tiempo, se obtuvo la velocidad media del viaje.

## Variables de la estación

### *Clusterización jerárquica*

Debido al gran número de estaciones frente al cual nos encontrábamos, íbamos a tener que considerar 388 (194 estaciones * 2) dimensiones más a la hora de querer realizar el clustering de viajes, y todas las posibles combinaciones entre ellas. Es por esto que decidimos realizar primero un clustering jerárquico de las estaciones para así reducir la dimensionalidad que aportaban.

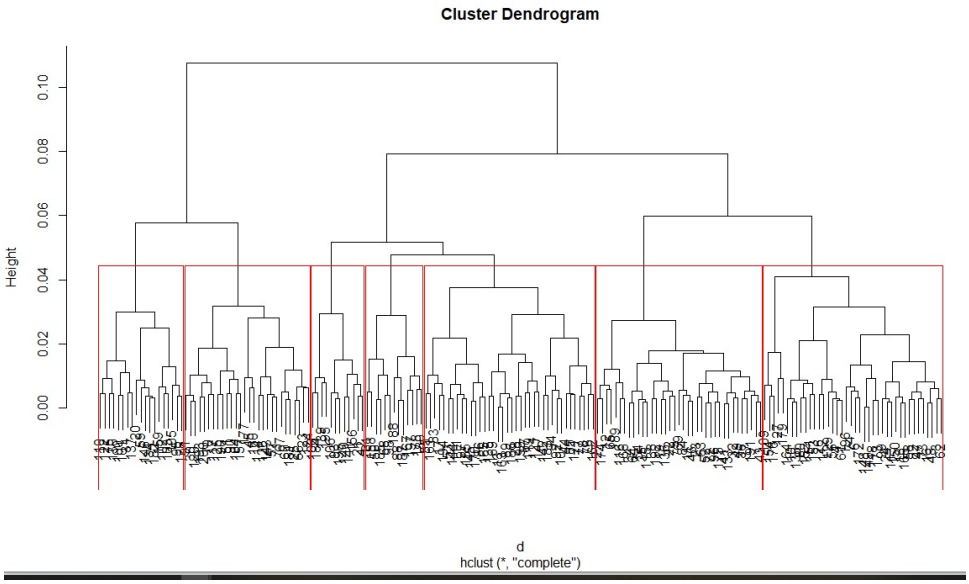
Algunas consideraciones que tomamos:

- El clustering jerárquico se realizó por el método de distancias manhattan que hay entre las estaciones.
- Se consideró las estaciones de origen para agrupar ya que nos parecía más importante la decisión del usuario de dónde retirar la bicicleta.
- Se decidió unificar las tres estaciones de Retiro (Id 2,130,131) debido a que estaban muy próximas entre sí (máximo a una cuadra de distancia), por lo que se consideró que resultaba indistinto sacar una bicicleta de cualquiera de ellas, sino que dependía más bien, de la disponibilidad de bicicletas de bicicletas de las mismas en ese instante.
- Luego se observó cuales estaciones de origen tenían una mayor cantidad de viajes y se tomó la decisión de no considerarlas en el algoritmo de clustering jerárquico ya que perderían relevancia al ser asignadas dentro de un grupo. Esto se debe a que se realizó el clustering por las distancias, sin tener en cuenta el número de viajes de cada estación.

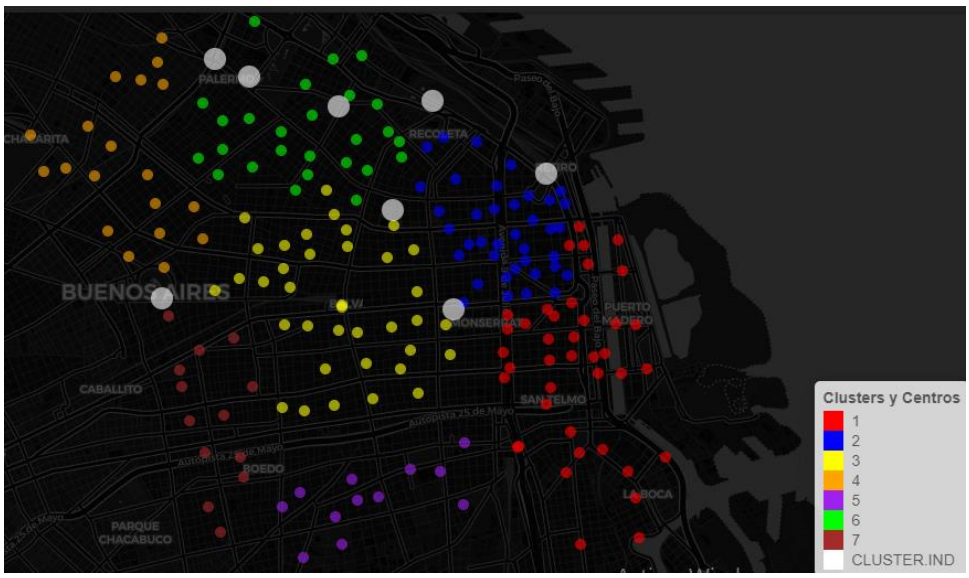
Estaciones omitidas del clustering:

| id_estacion | nombre_estacion |
|---|---|
| 1 | Facultad de Derecho |
| 2 | Retiro |
| 5 | Plaza Italia |
| 8 | Congreso |
| 9 | Parque Las Heras |
| 14 | Pacífico |
| 29 | Parque Centenario |
| 33 | Facultad de Medicina |

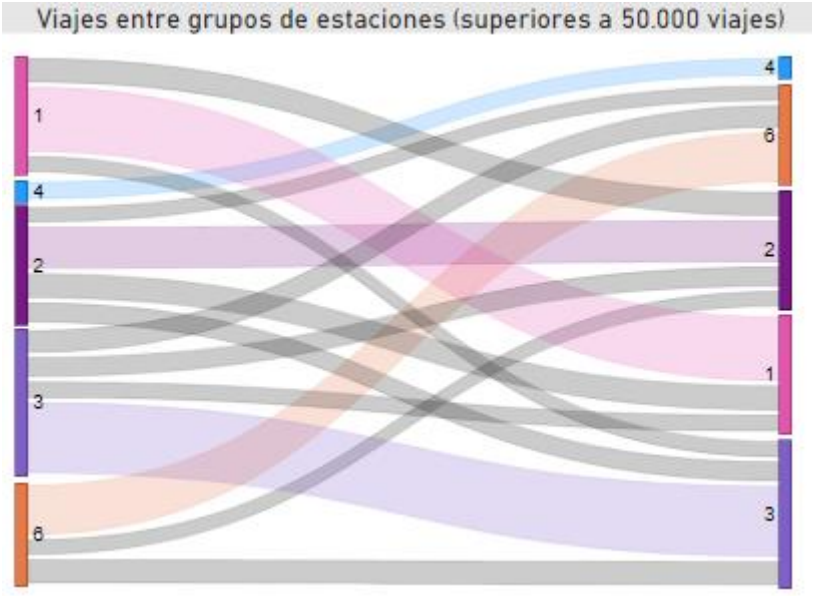
Realizamos el clustering jerárquico con diferentes números de grupos observando cómo se agrupaban los mismos en un dendograma. A continuación, se muestra el dendograma con el corte realizado en 7 grupos:




Para visualizar mejor los clusters de estaciones que se armaron se representó a estos 7 grupos en un mapa utilizando la librería leaflet y publicando el mismo en Rpubs como para poder compartirlo: [https://rpubs.com/Bcasado/654186](https://rpubs.com/Bcasado/654186)




Para entender el flujo de viajes que se da entre estos grupos de estaciones se visualizó en un diagrama de Sankey. Se respresentó solo aquellos con un flujo de viajes más significativo, es decir, mayor a 50.000 viajes. Se observan principalmente viajes dentro del mismo grupo. Y luego viajes entre grupos más “centricos” de la ciudad”. Ya a medida que nos alejamos de la zona centrica solo se dan viajes dentro del grupo como es el caso del grupo 4.



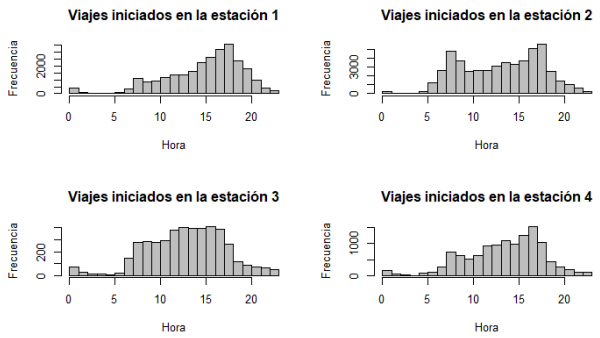

*Caracterización de la estación:*

También se clasificó cada estación según el horario de uso para poder recabar más información de las mismas. Para esto, se definieron intervalos de 0-7, 8-15 y 16-23 hs. Además, se consideró que cada viaje iniciaba y terminaba en el mismo intervalo.

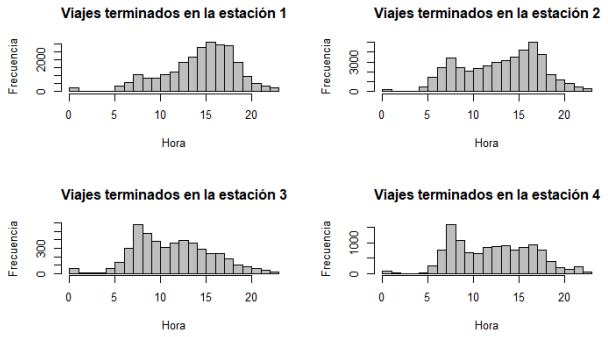
Al sumar cada uno de los 2.5 millones de viajes según si pertenecían a la estación origen o a la estación destino, se determinó que el total de viajes en uno de los tres intervalos era mucho mayor a los otros dos (un 150% mayor), entonces se podía afirmar que la mayoría de los viajes se daban en ese intervalo.

Por ejemplo, las salidas de la estación 7 “obelisco”, se clasifican en: (323; 2745; 2746) y los arribos en (255; 3016; 1861), por lo que a la misma se la puede clasificar en “distribuida” para los viajes de origen, y “mayormente de 8-15" para los arribos.

Con esto último, se agregaron dos nuevas variables: “Tipo estación origen” y “Tipo estación destino”, caracterizando así a cada viaje según el tipo de estación (mayormente de 0-7, mayormente de 8-15, mayormente de 16-23, distribuido) y aportando más información que simplemente el “id” o el grupo de localización geográfica al cual pertenecían.



Distribución horaria de los viajes originados en las estaciones 1 a 4 en todo el año 2018




Distribución horaria de los viajes terminados en las estaciones 1 a 4 en todo el año 2018

Se puede observar entonces, que para la estación de la facultad de derecho (1), su uso se da mayormente entre las 16 y las 23hs tanto como origen como como fin de los viajes; la estación de retiro (2) tiene un uso distribuido; la estación aduana (3) y la estación plaza roma (4) tienen un uso distribuido para el origen, pero se las utiliza principalmente como estación de destino entre las 8 y las 15hs.

## Variables climáticas:

- Temperatura (C) (vc)
- Humedad (%) (vc)
- Presión (hPa) (vc)
- Viento (km/h) (vc)
- Precipitación (mm) (vc)

Cabe aclarar que los datos de las precipitaciones estaban dados en intervalos de tres horas, por lo que, para poder trabajar con todas las variables climáticas en el mismo rango, se interpoló en las horas intermedias para disponer de los datos. Por otro lado, al disponer de dos datos de la velocidad del viento, se utilizó únicamente el de Aeroparque, para evitar la correlación innecesaria entre las variables habiendo hecho previamente una correlación entre ambas estaciones y observando que la misma era muy alta. Asimismo, se disponía de otras variables como la dirección del viento, o la nubosidad, las cuales fueron descartadas al no poder trabajarse fácilmente o por poseer muchos datos faltantes.

## Variables utilizadas para la segmentación

En resumen, las variables utilizadas para realizar la segmentación son las siguientes:

- Mes
- Día
- Hora
- Día de la semana
- Día no laborable
- Grupo estación origen
- Grupo estación destino
- Tipo estación origen
- Tipo estación destino
- Distancia

- Velocidad
- Temperatura
- Humedad
- Temperatura
- Presión
- Viento
- Precipitación

Donde las primeras nueve son variables categóricas y el resto variables continuas. Dado que el algoritmo k-means no admite variables categóricas (ver metodología), se realizó un análisis de correspondencias para estas variables.

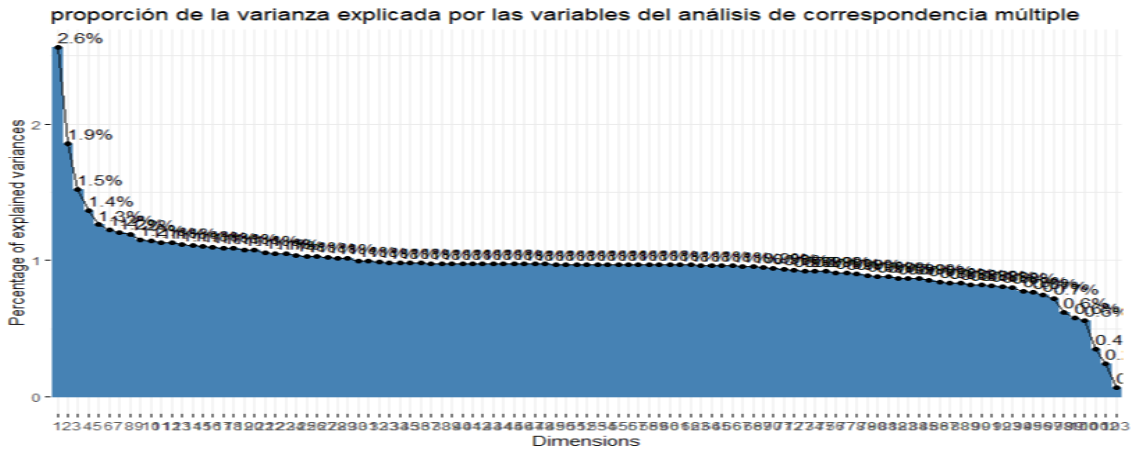
## Análisis de correspondencia

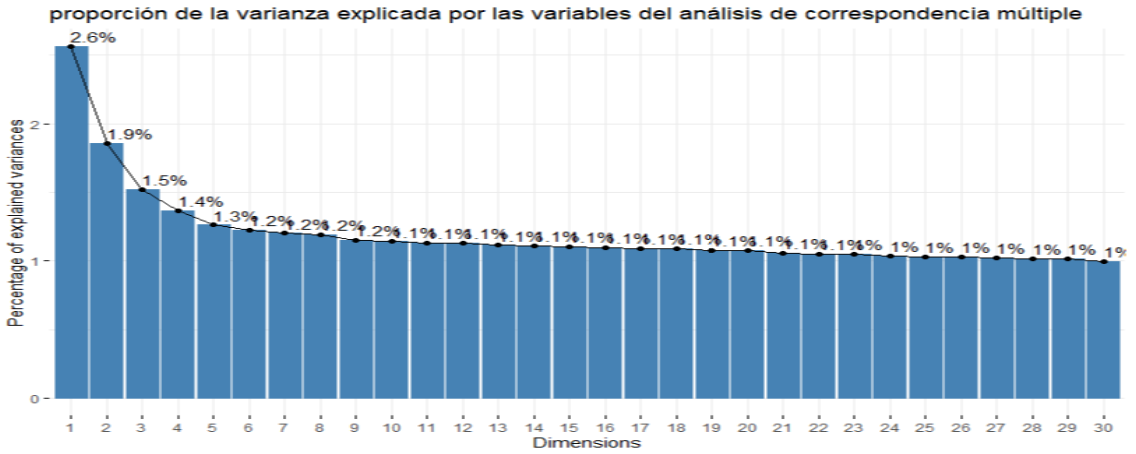
Debido a que la naturaleza del análisis del trabajo práctico es la de encontrar grupos de viajes similares y a que los datos a analizar poseían variables categóricas y sin un ordenamiento establecido, fue necesario realizar un análisis de correspondencias entre las mismas para obtener variables continuas. Esto se debe a que los algoritmos de clusterización, como el K-means o CLARA que explicaremos a continuación, interpretan a todas las variables com o continuas, y esto podría llevar a errores de análisis posterior.

Las variables categóricas eran las siguientes (categorías):

- Mes (12)
- Día (31)
- Hora (24): hora de inicio del viaje
- Día de la semana (7)
- Día no laborable (2): variable binaria
- Grupo origen (15)
- Grupo destino (15): ver apartado clusterización jerárquica
- Tipo estación origen (4)
- Tipo estación destino (4): ver apartado caracterización de la estación

Se puede observar que un mes “5.6” o un día “0.23” (no laborable un 23%) no tienen sentido interpretable. Tampoco se puede dar un valor numérico a las estaciones cuyo uso es “mayormente 16-23", ni se les puede asignar un orden respecto de “distribuido”, dado que no parecería haber una jerarquización natural, cosa que si se daría entre la educación inicial, secundaria y universitaria. Por otro lado, al tener 114 categorías, aumenta mucho el peso de las mismas en la clusterización y esto podría llevar a resultados erróneos



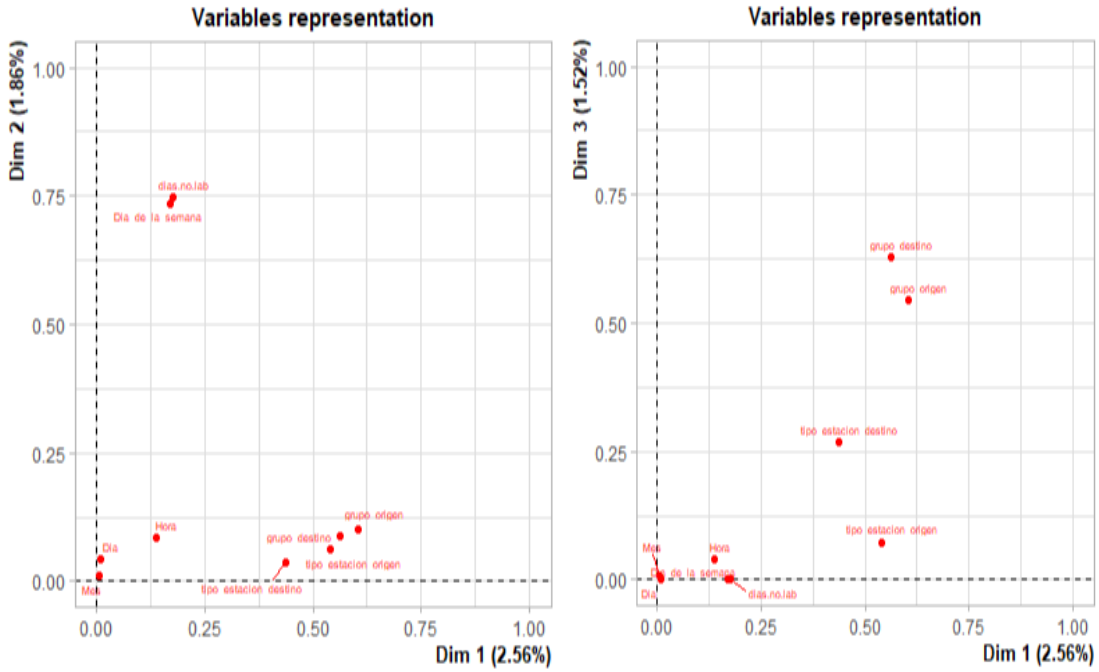

La segunda imagen es una ampliación de la primera para poder ver claramente que, a partir de la sexta variable el porcentaje de la varianza explicada por las variables siguientes ronda el 1%, y recién decrece rápidamente en la variable 100 aproximadamente.

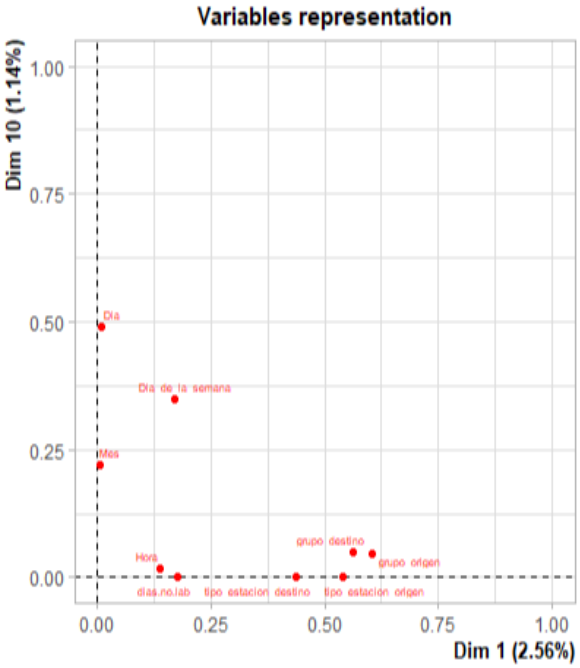
Si bien no hay un criterio exacto para elegir el número de variables a adoptar, es recomendable que se abarque el “codo” del Scereeplot y que la inercia de cada variable seleccionada esté siempre por encima de la inercia promedio del total de las variables. Por lo tanto, del total de las dimensiones se optó por tomar a priori las diez primeras, que cumplían con estas dos recomendaciones, para luego usar algunas de estas en la base de segmentación.

```
> mean(mcorresp[["eig"]][,1])
[1] 0.1111111
> mcorresp[["eig"]][1:10,1]
    dim 1     dim 2     dim 3     dim 4     dim 5     dim 6     dim 7     dim 8     dim 9
0.2930394 0.2124931 0.1739643 0.1561873 0.1442460 0.1402747 0.1373548 0.1359335 0.1316579
   dim 10
0.1304235
>
```

Inercias del análisis de correspondencias

Cabe aclarar que la razón de no adoptar más variables se debió a que se intentó mantener una cantidad de variables homogénea para evitar dar más importancia a las variables de tiempo u estación que a las climáticas y como una solución de compromiso debido a que para correr un MCA de esta cantidad de dimensiones, la computadora tardaba unas 12 o 16 hs, y sin poder utilizarla para otra tarea.



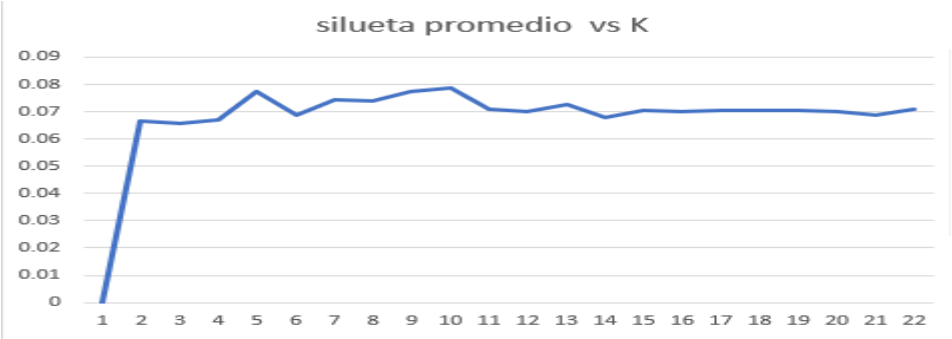

En las imágenes anteriores se puede observar la representación de las variables en los planos de las dimensiones correspondientes y como aporta cada una a la explicación de la varianza total. Se pudo observar que:

- La dimensión 1 agrupa a las variables de la estación, ya sean por el tipo o el grupo
- La dimensión 2 indica que las variables día de la semana y días no laborables están muy relacionados (hay una mayor cantidad de sábados y domingos que feriados en el resto de los días)
- La dimensión 3 distingue entre los grupos y los tipos de estación
- La dimensión 9 representa lo anterior aún más
- Las dimensiones 8 y 10 explican las variables de día y mes

## Segmentación

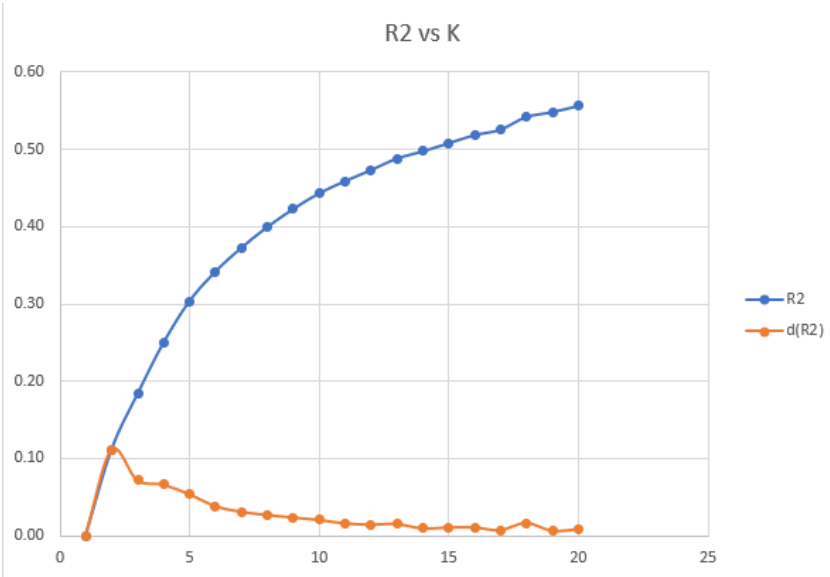
### Número óptimo de clusters

Dado que no se podía usar la librería de R *factoextra* para determinar el número óptimo de clusters a partir de la silueta promedio de los mismos debido a la gran cantidad de datos de análisis. Se procedió a realizar la misma de forma iterativa de 1 a 22 clusters para una pequeña cantidad de repeticiones buscando aquel que maximice la silueta promedio (usando CLARA) y determinar así el número óptimo de clusters.



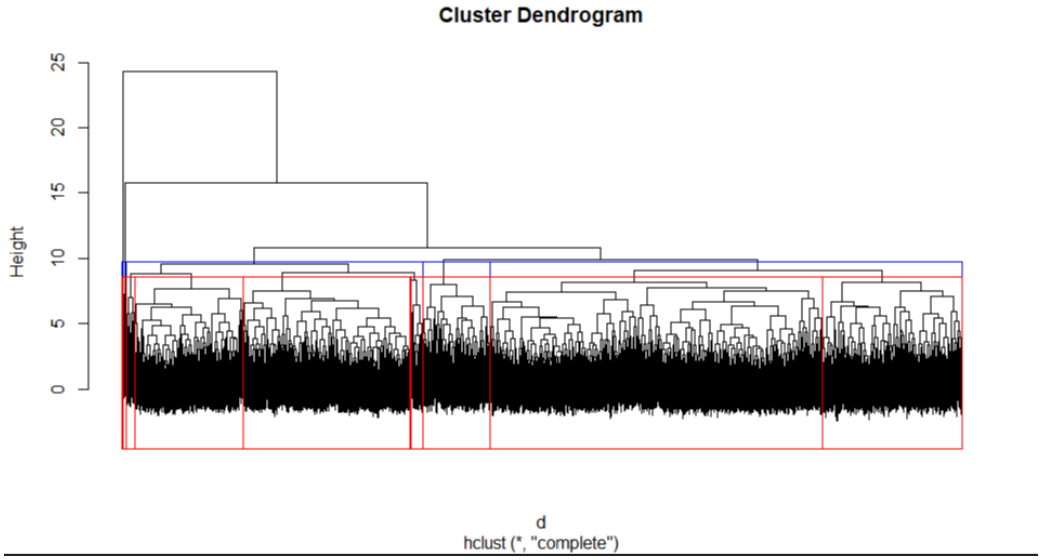

Del gráfico que se puede observar a continuación se puede observar que el número óptimo de clusters es de 10, y es el que empleamos para la segmentación.

Adicionalmente a esto, se iteró de 1 a 20 grupos, calculando el R2 para cada uno de ellos usando el algoritmo K-means para ver dónde se estaciona, independientemente del nivel.




Del gráfico se puede observar que, si bien el R2 siempre aumenta, el crecimiento es mucho menor en 10 grupos que en 5. Esto complementa el análisis de la silueta promedio, definiendo como base de segmentación diez grupos en vez de cinco, y sabiendo que, en caso de querer dividir los viajes en más grupos, habría muchos que serían parecidos entre sí.

Por último, se tomó una muestra de 10000 viajes y se le realizó una clusterización jerárquica, para observar los distintos niveles de corte de 5 y 10 grupos, que son los que dan la silueta máxima.




El inconveniente de este método es que, al realizarse un muestreo de 10000 viajes, la muestra es del 0.0004% de la población total (2.5 M de viajes) y, por lo tanto, variaba mucho de corrida a corrida. Si bien la mayoría daba que la partición de cinco grupos quedaba muy próxima a la de 6, hubo casos en los que no. Sin embargo, dado que no se podía generar una matriz de distancias más grande, no se pudo ampliar la base de muestreo.

## Segmentación Kmeans

Para la segmentación por Kmeans, se utilizó el siguiente código:

```
Library (cluster)

K=10

 Kclu <- kmeans(analisis, K ,nstart=25,iter.max=1000, algorithm = "Lloyd",trace=T)
```

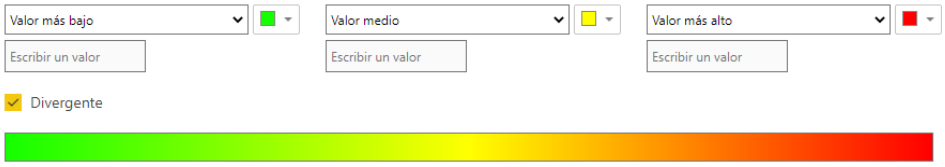
Dónde análisis es la base utilizada para segmentación, que incluye las variables de temperatura, humedad, presión, viento precipitación y las tres primeras variables del análisis de correspondencia. La razón de usar tres se debió a que estaba próximo al codo del gráfico screeplot y a que balanceaba bien el peso de las variables continuas y las del análisis de correspondencia.

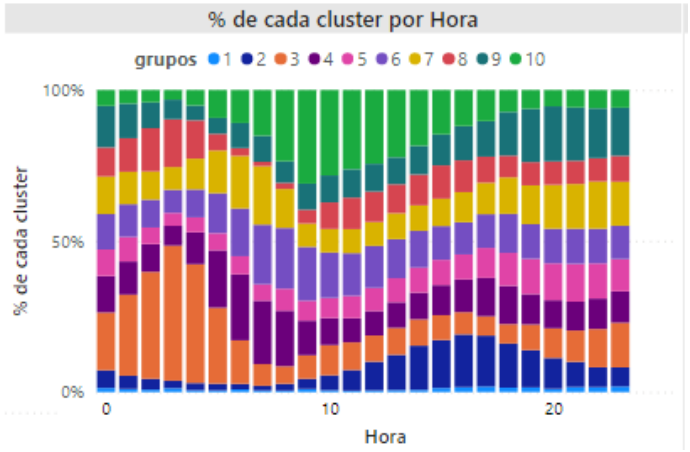
Además, se realizaron 25 repeticiones y el algoritmo iteraba hasta 1000 veces como máximo antes de converger. El algoritmo es “Lloyd”, que el que normalmente se utliza para el K-means.

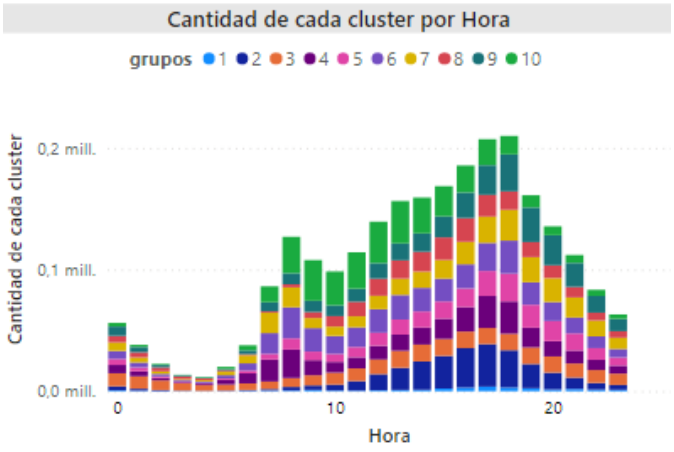
## Resultados Kmeans

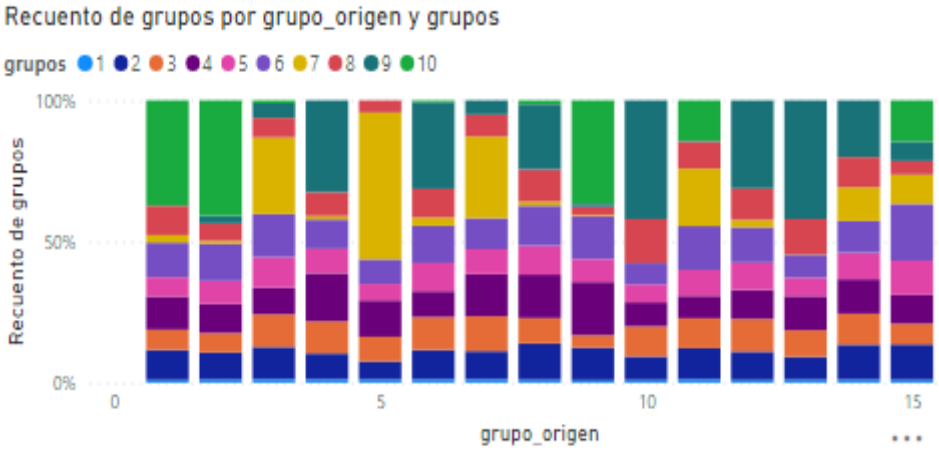
| grupos | Recuento de grupos | Promedio de dias.no.lab | Promedio de Circulares (estacion) | Promedio de Circulares (grupo) | Promedio de Dist.total | Máx. de Dist.total | Promedio de Velocidad.km.h | Máx. de Velocidad. km.h |
|---|---|---|---|---|---|---|---|---|
| 1 | 29288 | 0,24 | 0,09 | 0,36 | 2,51 | 13,20 | 8,28 | 25,24 |
| 2 | 260264 | 0,08 | 0,04 | 0,29 | 2,71 | 12,27 | 8,52 | 25,78 |
| 3 | 245424 | 1,00 | 0,00 | 0,21 | 3,33 | 10,96 | 10,51 | 32,81 |
| 4 | 281895 | 0,04 | 0,00 | 0,02 | 5,68 | 14,76 | 12,75 | 68,76 |
| 5 | 225319 | 0,11 | 0,03 | 0,27 | 2,73 | 11,02 | 9,34 | 27,19 |
| 6 | 327732 | 0,13 | 0,02 | 0,26 | 2,76 | 11,09 | 9,60 | 26,94 |
| 7 | 280986 | 0,04 | 0,14 | 0,52 | 1,83 | 6,57 | 6,52 | 22,58 |
| 8 | 207936 | 1,00 | 0,33 | 0,68 | 1,00 | 7,44 | 2,17 | 12,47 |
| 9 | 296809 | 0,02 | 0,12 | 0,40 | 1,83 | 6,61 | 6,81 | 20,84 |
| 10 | 361585 | 0,02 | 0,12 | 0,54 | 1,56 | 6,07 | 6,23 | 19,69 |

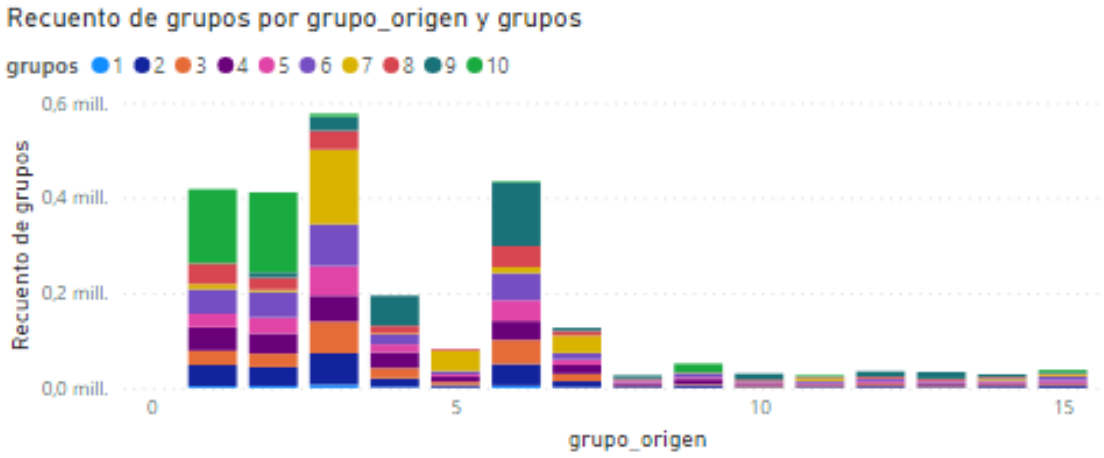
| grupos | Recuento de grupos | Promedio de viento.Km.h | Máx. de viento.Km.h | Mín. de Precipitacion. mm | Promedio de Precipitacion. mm | Máx. de Precipitacion .mm | Mín. de Temp.C | Promedio de Temp.C | Máx. de Temp.C |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 29288 | 16,99 | 57 | 1,67 | 3,59 | 12,50 | 11,10 | 21,77 | 29,20 |
| 2 | 260264 | 14,54 | 35 | 0,00 | 0,02 | 1,60 | 13,50 | 25,41 | 35,20 |
| 3 | 245424 | 12,19 | 33 | 0,00 | 0,06 | 2,20 | 4,90 | 17,92 | 31,80 |
| 4 | 281895 | 13,29 | 41 | 0,00 | 0,04 | 2,37 | 3,70 | 18,14 | 34,60 |
| 5 | 225319 | 27,24 | 56 | 0,00 | 0,06 | 2,20 | 7,00 | 16,54 | 29,30 |
| 6 | 327732 | 10,64 | 24 | 0,00 | 0,01 | 2,00 | 3,00 | 11,46 | 23,10 |
| 7 | 280986 | 13,17 | 39 | 0,00 | 0,05 | 2,13 | 4,10 | 18,62 | 33,40 |
| 8 | 207936 | 14,94 | 44 | 0,00 | 0,03 | 2,20 | 3,20 | 19,11 | 31,80 |
| 9 | 296809 | 13,27 | 35 | 0,00 | 0,05 | 2,00 | 4,10 | 18,33 | 32,50 |
| 10 | 361585 | 13,50 | 37 | 0,00 | 0,04 | 2,00 | 3,70 | 18,01 | 33,40 |

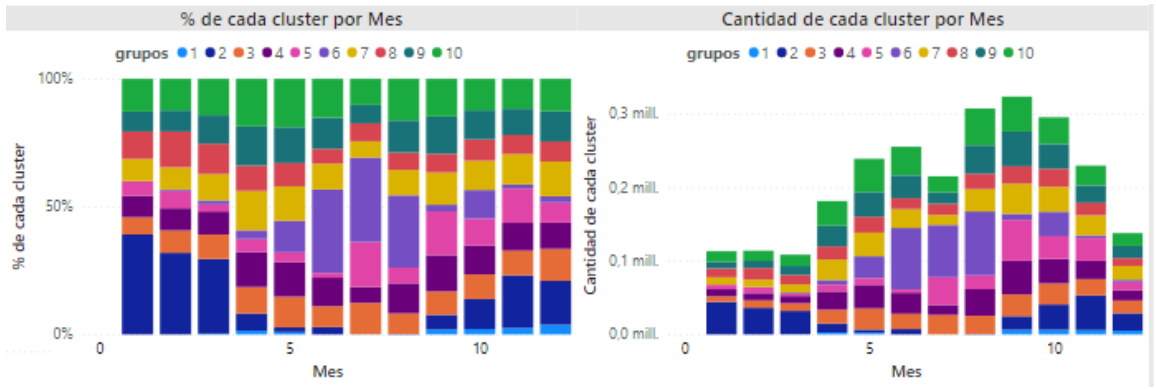


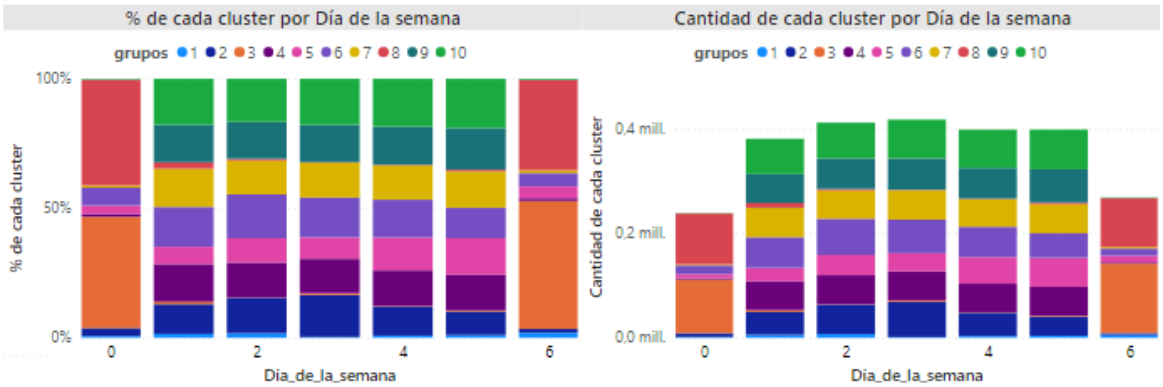




Cantidad de cada cluster por Hora
grupos 1 2 3 4 5 6 7 8 9 10
Cantidad de cada cluster
0,2 mill.
0,1 mill.
0,0 mill.
0
10
20
Hora


Recuento de grupos por grupo_origen y grupos
grupos 1 2 3 4 5 6 7 8 9 10
Recuento de grupos
100%
50%
0%
0
5
10
15
grupo_origen


Recuento de grupos por grupo_origen y grupos
grupos 1 2 3 4 5 6 7 8 9 10
Recuento de grupos
0,6 mill.
0,4 mill.
0,2 mill.
0,0 mill.
0
5
10
15
grupo_origen




**Grupo 1 (29k viajes):**

Este grupo es el que menos viajes tiene, entendemos que es porque es el que agrupa la mayor parte de los viajes de lluvia mientras que el resto de los grupos tiene otras características. Cabe destacar, además, que los recorridos se caracterizan por ser medios, dentro de el mismo grupo de estaciones, cuando quizás es difícil conseguir un medio de transporte público, pero es necesario llegar a destino medianamente rápido, incluso con lluvia.

**Grupo 2 (260k viajes):**

Este grupo se caracteriza por ser viajes de verano, en dónde las temperaturas son más altas y en un rango horario mayormente durante 10-20 y los días laborales en su gran mayoría. Esto se puede entender como que los que realizan los viajes prefieren la bicicleta cuando la media es más alta antes que otro tipo de transporte. Cabe destacar que las distancias en este caso también son medias (unas 27 cuadras), con un 30% de los viajes que terminan dentro del grupo.

**Grupo 3 (245k viajes):**

Estos viajes se dan por completo los días no laborables y mayormente en el fin de semana, con un uso medio a lo largo del día y a lo largo del año. Además, se caracterizan por ser uso recreacional, durante todo el año, fin de semana, y dentro de todo, velocidades más altas que el resto de los otros grupos. Hay que destacar también que, si bien la mayoría de los viajes son durante el día, conforman una buena parte de los viajes nocturnos (en proporción al resto de los grupos).

**Grupo 4 (282k viajes):**

En este caso, los viajes se caracterizan por recorrer una mayor distancia (promedio de 57 cuadras) y a gran velocidad (promedio de 12,75 km/h). Hay que destacar también, que muchos de estos viajes se dan por la mañana (8 hs) y otros a por la tarde (17-19 hs)

**Grupo 5 (225k viajes):**

Este grupo se compone principalmente en viajes de julio a noviembre (invierno-primavera, con temperaturas medias crecientes), más que nada por el horario de la tarde y durante la semana. Con velocidades medias a bajas y en general de distancias medias. Se podría decir que son viajes más que nada ventosos (Promedio de 27,24 km/h)

**Grupo 6 (328k viajes):**

Estos viajes se dan mayormente con temperaturas bajas (11`C), lo cual es normal en los meses en los que se dan principalmente, que son junio - septiembre. Además, el horario es bastante distribuido durante el día, pero al igual que el grupo cuatro, tiene picos en los horarios de entrada y salida al trabajo.

**Grupo 7 (281k viajes):**

Este grupo se caracteriza por tener la mitad de los viajes el mismo grupo de estación, las distancias son relativamente cortas (18 cuadras), así como la velocidad. Su uso se da más que nada los días laborales, por la tarde y durante todo el año. Notar además que los viajes se dan más que nada en la zona sur de las estaciones, es decir, los grupos 3, 5 y 7, (Caballito, Almagro, etc.)

**Grupo 8 (208k viajes):**

Este grupo, al igual que el 3, se da mayormente en los días no laborables (principalmente sábados y domingos) a lo largo de todo el año y en el horario de media tarde. Se podrían describir como viajes recreacionales, a baja velocidad, donde prácticamente no hay precipitaciones y las distancias son relativamente cortas. Además, es el grupo que más aglomera los viajes circulares ya sea de estación o de grupo de estaciones, es decir, que comienzan y terminan por la zona y encima las velocidades son bajas.

**Grupo 9 (297k viajes):**

Estos viajes se dan principalmente para distancias cortas y baja velocidad, cuando el 40% de los viajes son dentro de la misma zona dentro de los días laborales. Al darse mayormente por la tarde, intuimos que muchos de estos podrían ser utilizados por Rappis u otras plataformas de Delivery, dado que en aquel momento era común verlos usando las Ecobicis para eso.

**Grupo 10 (361k viajes):**

Dentro de todo, este grupo parecería tener una distribución de viajes en los que las temperaturas son frías, pero no muy frías (de ahí que no toma tantos viajes en el mes de julio) y más que nada durante el horario de la mañana, y durante los días laborales. Respecto a la distancia, también es baja (en promedio unas 15 cuadras) gran parte de los viajes terminan por la zona.

## Segmentación CLARA

Ahora bien, para la segmentación por CLARA, se utilizó el siguiente código:

```
library(cluster)

k=10
```

clarax= clara(analisis,k,metric = "euclidean",pamLike = T,stand=T,correct.d = T,sampsize = 10000,samples=50,trace = T)

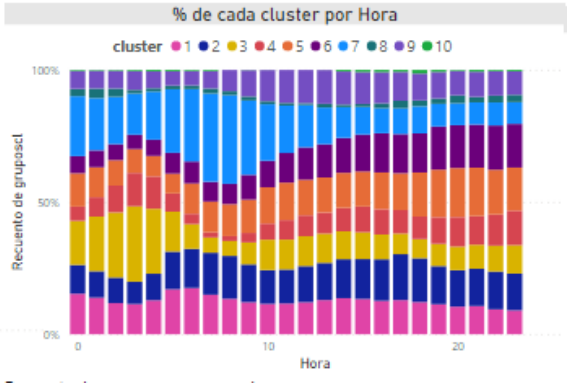
Dónde análisis es la base utilizada para la segmentación, que incluye todas las variables de Mes, día, hora, día de la semana, temperatura, humedad, presión, viento, días no lab., dist. Total, velocidad del viaje, precipitación, grupo origen, grupo destino, tipo estación origen, tipo estación destino.

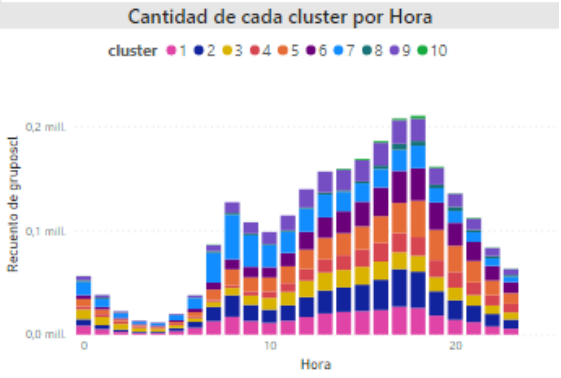
El tamaño de la muestra para clara fue de 10000 y se realizaron 50 repeticiones para llegar al para encontrar la que mejor ajustara los grupos. La métrica utilizada para calcular la distancia entre los puntos (variable continua) es la euclídea y todas las variables fueron estandarizadas.

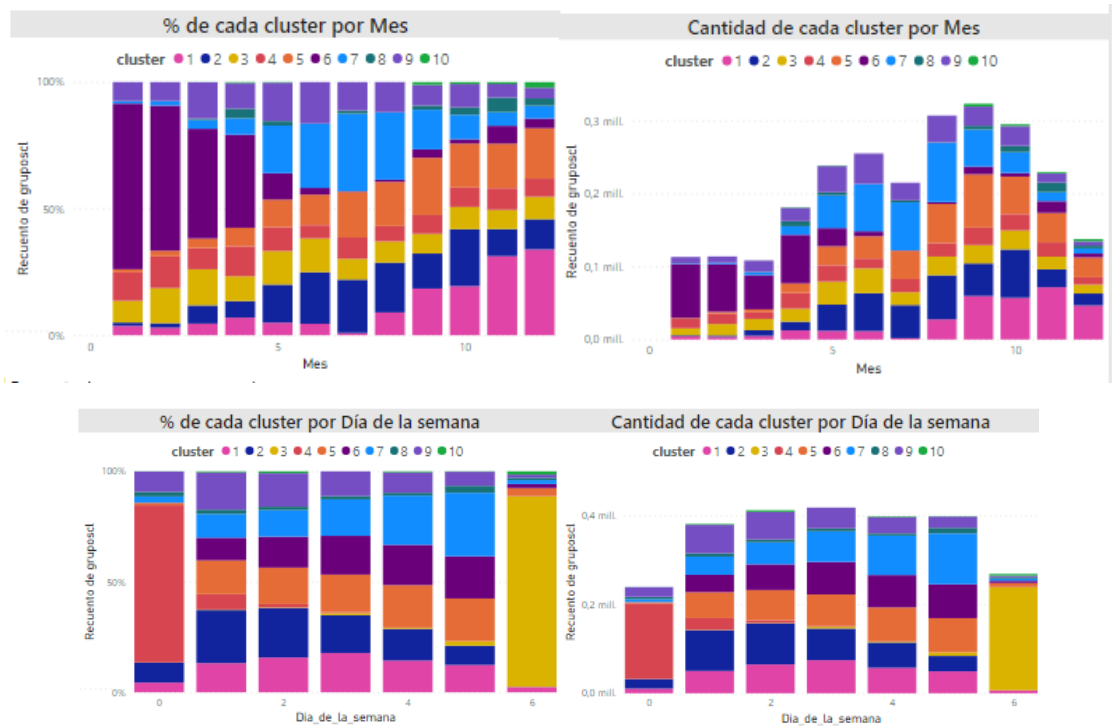
## Resultados CLARA

| gruposcl | Recuento de gruposcl | Promedio de dias.no.lab | Promedio de Circulares (estación) | Promedio de Circulares (grupo) | Promedio de Dist.total | Máx. de Dist.total | Promedio de Velocidad.km.h | Máx. de Velocidad. km.h |
|---|---|---|---|---|---|---|---|---|
| 1 | 313290 | 0,07 | 0,01 | 0,27 | 3,12 | 13,86 | 10,48 | 55,28 |
| 2 | 368438 | 0,08 | 0,00 | 0,15 | 4,19 | 14,76 | 11,43 | 68,76 |
| 3 | 248203 | 1,00 | 0,10 | 0,37 | 2,59 | 13,95 | 7,59 | 29,64 |
| 4 | 206385 | 0,98 | 0,16 | 0,41 | 2,22 | 14,12 | 5,89 | 26,55 |
| 5 | 362190 | 0,04 | 0,15 | 0,36 | 1,77 | 9,66 | 6,45 | 25,84 |
| 6 | 324571 | 0,03 | 0,09 | 0,39 | 2,41 | 13,91 | 7,41 | 25,37 |
| 7 | 373704 | 0,04 | 0,03 | 0,37 | 2,55 | 11,82 | 9,09 | 26,96 |
| 8 | 46192 | 0,17 | 0,07 | 0,34 | 2,59 | 13,63 | 8,49 | 25,15 |
| 9 | 261669 | 0,13 | 0,21 | 0,60 | 1,51 | 9,59 | 4,63 | 19,00 |
| 10 | 12596 | 0,34 | 0,10 | 0,37 | 2,53 | 12,44 | 8,25 | 25,24 |

| gruposcl | Recuento de gruposcl | Promedio de viento.Km.h | Máx. de viento.Km.h | Min. de Precipitacion. mm | Promedio de Precipitacion. mm | Máx. de Precipitacion. mm | Min. de Temp.C | Promedio de Temp.C | Máx. de Temp.C |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 313290 | 17,29 | 56 | 0,00 | 0,02 | 1,00 | 7,10 | 20,81 | 31,50 |
| 2 | 368438 | 13,03 | 41 | 0,00 | 0,01 | 1,00 | 3,00 | 15,41 | 31,40 |
| 3 | 248203 | 12,69 | 41 | 0,00 | 0,03 | 0,90 | 4,20 | 17,71 | 31,10 |
| 4 | 206385 | 15,14 | 41 | 0,00 | 0,03 | 1,07 | 3,00 | 19,41 | 35,10 |
| 5 | 362190 | 16,70 | 56 | 0,00 | 0,03 | 1,00 | 4,10 | 16,62 | 33,20 |
| 6 | 324571 | 13,32 | 56 | 0,00 | 0,02 | 0,93 | 10,10 | 24,87 | 35,20 |
| 7 | 373704 | 13,17 | 39 | 0,00 | 0,03 | 1,00 | 3,00 | 13,40 | 26,00 |
| 8 | 46192 | 16,09 | 57 | 0,83 | 1,72 | 3,23 | 7,60 | 20,84 | 31,80 |
| 9 | 261669 | 13,50 | 56 | 0,00 | 0,01 | 1,07 | 3,00 | 17,04 | 31,80 |
| 10 | 12596 | 16,69 | 52 | 3,30 | 5,20 | 12,50 | 13,10 | 22,56 | 29,20 |






**Grupo 1 (313k viajes):**

Se pueden describir como viajes en jornada laboral, ya sea a la ida del trabajo o a la vuelta, cubriendo un amplio rango de temperaturas, pero concentrándose en los meses de primavera-verano en los que el día es caluroso y sin lluvia, y quizás tomar otro medio de transporte no es tan conveniente. No son viajes de recreación, si bien pueden ser utilizados para trayectos cortos, la media de estos viajes es de unas treinta cuadras.

**Grupo 2 (368k viajes):**

Este grupo aglomera los viajes en jornada laboral, ya sea a la ida del trabajo o a la vuelta, cubriendo un rango aún más amplio de temperaturas, pero concentrándose en los meses de invierno-primavera en los que el día ya no es tan caluroso y sin lluvia. No son viajes de recreación, y se caracterizan por ser viajes largos en distancia, rápidos y pocos terminan dentro del mismo grupo de estaciones.

**Grupo 3 (368k viajes):**

Se caracteriza por contener viajes más recreacionales, ya que todos se realizan en días no laborales, más que nada los sábados y en un horario bastante distribuido a lo largo de todo el año, pero sin lluvia. En general se desarrollan en la misma zona, (25 cuadras).

**Grupo 4 (206k viajes)**

Este grupo es bastante parecido al anterior, pero con la diferencia que agrupa los viajes en el día domingo en vez de los sábados. Con la particularidad de que agrupa los viajes en el día domingo, pero con una menor velocidad.

**Grupo 5 (362k viajes):**

Este grupo se caracteriza por los días laborales a lo largo de toda la semana y con viajes a lo largo del día, pero con un clima bastante ventoso de 16.7 km/h y llegando a los 56 km/h. Además, se dan principalmente en verano.

**Grupo 6 (324k viajes):**

Se pueden describir como viajes en días laborales, pero del inicio de año, siendo su uso medio a lo largo del día y con una media de temperaturas más altas para distancias que van de entre veinte y treinta cuadras. Además, muchos de estos viajes terminan quedando en la misma zona de CABA.

**Grupo 7 (373k viajes):**

Este grupo, junto con el 2 y el 5 son los que más engloban los viajes de trabajo con la particularidad de que en este caso los viajes se realizan durante el invierno y principalmente a la mañana los días en los que prácticamente no hay lluvia y a una velocidad normal para un ciclista.

**Grupo 8 (46k viajes):**

Teniendo una significativamente menor cantidad de viajes este grupo engloba a todos los viajes con lluvia media (1.72 mm) y vientos dentro de todo fuertes. No es posible ver en que proporción está distribuido en los meses debido a la poca cantidad de viajes que tiene en comparación con otros grupos.

**Grupo 9 (260k viajes)**:

Este grupo engloba a todos los viajes de distancias cortas y la mayoría de estos son viajes circulares, ya sea porque la estación de origen es la misma que la de destino o porque ambas están en el mismo grupo. No hay una incidencia particular en los meses del año, pero si en que en general se dan en días laborables.

**Grupo 10 (12.5k viajes):**

Por último, este último grupo es parecido al grupo 8, con la particularidad de que engloba a los viajes con lluvias mucho más intensas, vientos más fuertes. El rango de temperaturas es elevado y más acotado, lo cual corresponde con que se den principalmente en los meses de primavera-verano, que es donde hay mayor cantidad de lluvias.

# CONCLUSIONES

A la hora de querer caracterizar los grupos a partir de los algoritmos utilizados se observó que ambos lograron una buena segmentación. En el caso del algoritmo CLARA las características distintivas de cada grupo se pudieron identificar con mayor facilidad que en el caso del algoritmo K-means. En los grupos obtenidos a partir del algoritmo K-means se dio una distinción más compleja, en la que no se segmentaba simplemente por una única variable, sino por un conjunto de ellas. Esto puede ser debido a que, dado que el k-means usa una base de segmentación que posee un análisis de correspondencia previo, despreciando información, por considerarla de ruido y ponderando mayormente por las variables continuas, (más que nada las del viento, distancia, lluvia y velocidad).

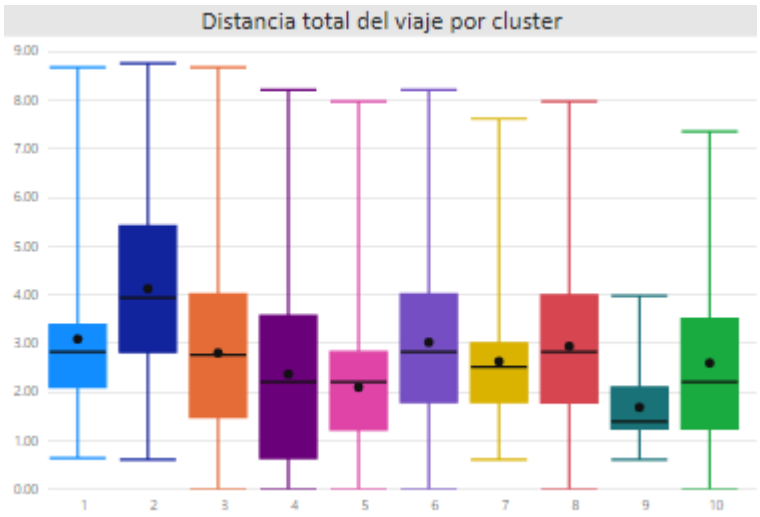
Por otro lado, el algoritmo CLARA exige bastante menos memoria computacional que el K-means dado que trabaja con muestras y no con el total de la base de datos. Si bien no pierde información desde el punto de vista de las variables, es necesario hacer unas cuantas repeticiones hasta lograr disminuir el error de muestreo. Respecto a la interpretación, consideramos que fue más

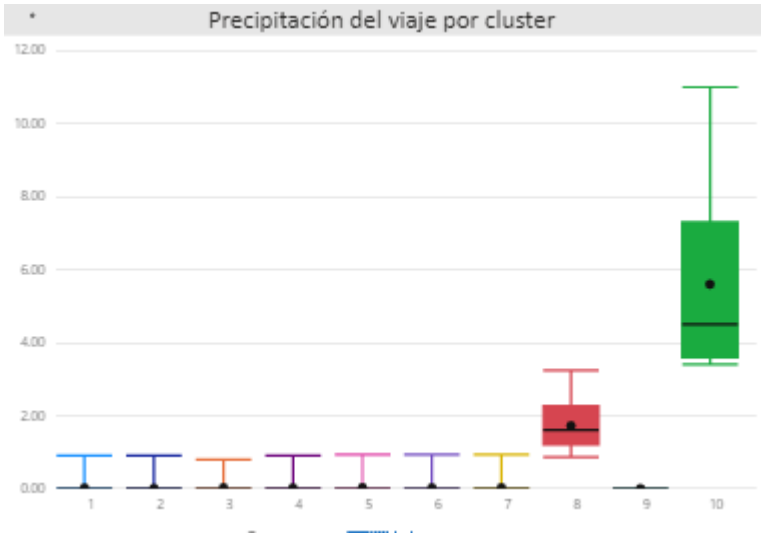
sencilla realizarla con CLARA, pero eso no significa que la agrupación sea mejor, por ejemplo, k-means separó los viajes de días no laborales en dos grupos, donde más o menos la cantidad de viajes correspondía por igual a sábado o domingo, mientras que en CLARA los dividió según el día. En el caso de la variable precipitación, en el algoritmo K-means agrupó la mayoría de los viajes con lluvia en un solo grupo, mientras que en k-means lo separó en uno de lluvias fuertes y otro de lluvias medias.

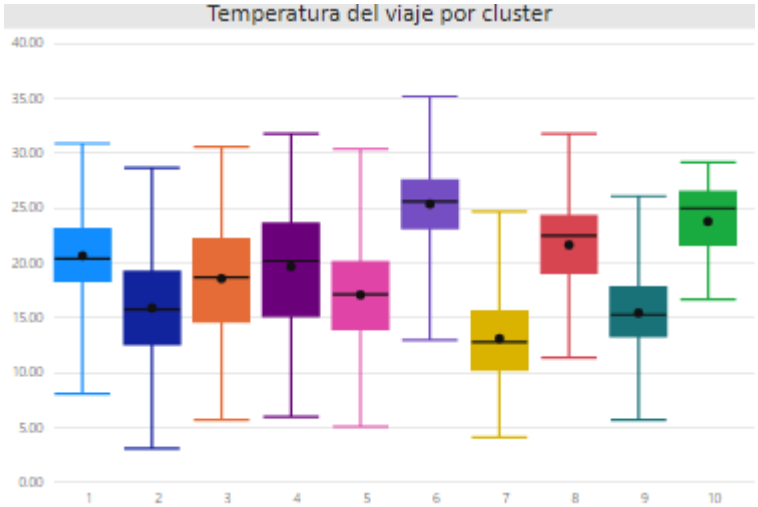
Respecto a la segmentación en sí, en ambos casos dio resultados coherentes. Si bien el análisis no puede ser utilizado para predecir cuándo se dará un viaje u otro según las condiciones climáticas, podría servir de base para estudios sociales de comportamiento de las personas si se tuviera el id del usuario y se lo empleará para ver si una persona realiza más un tipo de viaje u otro. Otro análisis podría ser el de verificar que se mantienen los grupos si cambia el sistema de Ecobici (modificado en 2019) o para evaluar la localización de las estaciones y proyectar nuevas en caso de haber mucha demanda en estaciones cercanas. También se podría estudiar a aquellos viajes considerados como “recreacionales” para estudiar entre que estaciones se dan, y si estas son cercanas a algún centro artístico o lugar histórico. Por último, el mismo análisis se podría hacer para aquellos considerados como laborales, para proveer a las estaciones de mayor demanda en ese horario y época del año de más bicicletas.

# APÉNDICES

CLARA BOX PLOTS:




## Tipos de las estaciones

setwd("C:/Users/Pedro/Desktop/Facultad 2020/Primer cuatrimestre/91.54 Ciencia de datos para la toma de decisiones/Trabajo práctico/Datos")

```
Ecobici2_cat=read.csv("Ecobici2_cat.csv")

## columnas: X, id_estación_origen, id_estación_destino, Mes, Día, Hora, Día_de_la_semana, dias_no_lab.

Ecobici2_cat=Ecobici2_cat[,-1] ## eliminamos la columna que indica el número de fila y ponemos a las variables como factores

Ecobici2_cat$id_estacion_origen=as.factor(Ecobici2_cat$id_estacion_origen)

Ecobici2_cat$id_estacion_destino=as.factor(Ecobici2_cat$id_estacion_destino)

Ecobici2_cat$Mes=as.factor(Ecobici2_cat$Mes)

Ecobici2_cat$Dia=as.factor(Ecobici2_cat$Dia)

Ecobici2_cat$Hora=as.numeric(Ecobici2_cat$Hora)

Ecobici2_cat$Dia_de_la_semana=as.factor(Ecobici2_cat$Dia_de_la_semana)

Ecobici2_cat$dias.no.lab=as.factor(Ecobici2_cat$dias.no.lab)

 Ecobici2_cat$Hora=as.numeric(Ecobici2_cat$Hora)

## ahora creamos un data frame que indicará el id en la primera columna, y en las otras la cantidad de viajes por rango horario (3 rangos)

## luego la clasificación en la columna 5. hacemos lo misimo para la cantidad de llegadas de viajes por rango horario y la clasificación en la columna 9

i=1

y=data.frame(0)

 for(i in 1:200){

 y[i,1]=i

 y[i,2]=sum(Ecobici2_cat$id_estacion_origen==i & Ecobici2_cat$Hora<=7)

 y[i,3]=sum(Ecobici2_cat$id_estacion_origen==i & Ecobici2_cat$Hora>=8 & Ecobici2_cat$Hora<=15)

 y[i,4]=sum(Ecobici2_cat$id_estacion_origen==i & Ecobici2_cat$Hora>=16 )

 ifelse(as.numeric(y[i,3])*1.5<as.numeric(y[i,2]) & as.numeric(y[i,4])*1.5<as.numeric(y[i,2]),

       y[i,5]<- "mayormente 0-7" ,

       ifelse (as.numeric(y[i,2])*1.5<as.numeric(y[i,3]) & as.numeric(y[i,4])*1.5<as.numeric(y[i,3]),

             y[i,5]<-"mayormente 8-15" ,

             ifelse (as.numeric(y[i,2])*1.5<as.numeric(y[i,4])& as.numeric(y[i,3])*1.5<as.numeric(y[i,4]),

                   y[i,5]<-"mayormente 16-23" ,

                   y[i,5]<-"distribuido")))


 y[i,6]=sum(Ecobici2_cat$id_estacion_destino==i & Ecobici2_cat$Hora<=7)

 y[i,7]=sum(Ecobici2_cat$id_estacion_destino==i & Ecobici2_cat$Hora>=8 & Ecobici2_cat$Hora<=15)

 y[i,8]=sum(Ecobici2_cat$id_estacion_destino==i & Ecobici2_cat$Hora>=16 )

 # nota, en realidad son las horas de salida, pero para simplificar, asumimos que todos los viajes se terminan dentro de esa hora, o dentro de ese rango

 ifelse(as.numeric(y[i,3])*1.5<as.numeric(y[i,6]) & as.numeric(y[i,4])*1.5<as.numeric(y[i,6]),

       y[i,9]<- "mayormente 0-7" ,

       ifelse (as.numeric(y[i,6])*1.5<as.numeric(y[i,7]) & as.numeric(y[i,8])*1.5<as.numeric(y[i,7]),

             y[i,9]<-"mayormente 8-15" ,
```

```
            ifelse (as.numeric(y[i,6])*1.5<as.numeric(y[i,8])& as.numeric(y[i,7])*1.5<as.numeric(y[i,8]),
                    y[i,9]<-"mayormente 16-23" ,
                    y[i,9]<-"distribuido")))
 ## el print es para marcar en que iteración estamos
  print (i)
}
colnames(y)=c("id","salidas 0-7","salidas 8-15","salida 16-23","tipo_estacion_origen", "arribos 0-7","arribos 8-15","arribos 16-23","tipo_estacion_destino")
## unificamos las estaciones de retiro
y[2,c(2,3,4,6,7,8)]=y[2,c(2,3,4,6,7,8)]+y[130,c(2,3,4,6,7,8)]+y[131,c(2,3,4,6,7,8)]
y=y[-c(15,44,130,131,159,162),] ## estaciones inexistentes

i=2
ifelse(as.numeric(y[i,3])*1.5<as.numeric(y[i,2]) & as.numeric(y[i,4])*1.5<as.numeric(y[i,2]),
     y[i,5]<- "mayormente 0-7" ,
     ifelse (as.numeric(y[i,2])*1.5<as.numeric(y[i,3]) & as.numeric(y[i,4])*1.5<as.numeric(y[i,3]),
           y[i,5]<-"mayormente 8-15" ,
           ifelse (as.numeric(y[i,2])*1.5<as.numeric(y[i,4])& as.numeric(y[i,3])*1.5<as.numeric(y[i,4]),
                 y[i,5]<-"mayormente 16-23" ,
                 y[i,5]<-"distribuido")))
 ifelse(as.numeric(y[i,3])*1.5<as.numeric(y[i,6]) & as.numeric(y[i,4])*1.5<as.numeric(y[i,6]),
       y[i,9]<- "mayormente 0-7" ,
       ifelse (as.numeric(y[i,6])*1.5<as.numeric(y[i,7]) & as.numeric(y[i,8])*1.5<as.numeric(y[i,7]),
             y[i,9]<-"mayormente 8-15" ,
             ifelse (as.numeric(y[i,6])*1.5<as.numeric(y[i,8])& as.numeric(y[i,7])*1.5<as.numeric(y[i,8]),
                   y[i,9]<-"mayormente 16-23" ,
                   y[i,9]<-"distribuido")))
sum(y[,5]=="mayormente 0-7")
sum(y[,5]=="mayormente 8-15")
sum(y[,5]=="mayormente 16-23")
sum(y[,5]=="distribuido")
sum(y[,9]=="mayormente 0-7")
sum(y[,9]=="mayormente 8-15")
sum(y[,9]=="mayormente 16-23")
sum(y[,9]=="distribuido")
write.csv(y,"horarios de las estaciones.csv")
```

## Clusterización jerárquica de estaciones

```
#Importar dataset
library(readxl)
Estaciones = read.csv("estaciones_sistema_viejo.csv",sep=";")

Estaciones <- Estaciones[,c(1,3,4)]
Estaciones  = Estaciones[Estaciones$id_estacion!=15,]
Estaciones  = Estaciones[Estaciones$id_estacion!=39,]
Estaciones=na.omit(Estaciones)
Estaciones[2,2]=sum(Estaciones[2,2]
              +Estaciones[Estaciones$id_estacion==130,2]
              +Estaciones[Estaciones$id_estacion==131,2])/3
Estaciones[2,2]=sum(Estaciones[2,3]
              +Estaciones[Estaciones$id_estacion==130,3]
              +Estaciones[Estaciones$id_estacion==131,3])/3
Estaciones  = Estaciones[Estaciones$id_estacion!=130,]
Estaciones  = Estaciones[Estaciones$id_estacion!=131,]

omitidas=rbind(Estaciones[Estaciones$id_estacion==1,],
          Estaciones[Estaciones$id_estacion==2,],
          Estaciones[Estaciones$id_estacion==5,],
          Estaciones[Estaciones$id_estacion==8,],
          Estaciones[Estaciones$id_estacion==9,],
          Estaciones[Estaciones$id_estacion==14,],
          Estaciones[Estaciones$id_estacion==29,],
          Estaciones[Estaciones$id_estacion==33,]
          )
Estaciones=Estaciones[Estaciones$id_estacion!=1,]
Estaciones=Estaciones[Estaciones$id_estacion!=2,]
Estaciones=Estaciones[Estaciones$id_estacion!=5,]
Estaciones=Estaciones[Estaciones$id_estacion!=8,]
Estaciones=Estaciones[Estaciones$id_estacion!=9,]
Estaciones=Estaciones[Estaciones$id_estacion!=14,]
Estaciones=Estaciones[Estaciones$id_estacion!=29,]
Estaciones=Estaciones[Estaciones$id_estacion!=33,]
```

```
datasclu <- Estaciones[,2:3]


# Hierarchical Clustering - Complete linkage, Ward, Single linkage

d <- dist(datasclu, method = "manhattan") # distance matrix

cluh <- hclust(d, method="complete")
```

## Unificador de estaciones con grupos de estaciones y tipos de estaciones

```
## cargamos los datos extraidos del power bi habiendo unificado previamente a cada viaje los datos del clima, el cálculo de distancia de viaje y velocidad.

 setwd("C:/Users/Pedro/Desktop/Facultad 2020/Primer cuatrimestre/91.54 Ciencia de datos para la toma de decisiones/Trabajo práctico/Datos")

Ecobici = read.csv2("Ecobici data final.csv")

 #Eliminamos las filas de viajes del 11/02/2018 al no contar con info. meteorologica de esa fecha

Ecobici<- Ecobici[!grepl("2018-02-11", Ecobici$Fecha_y_hora), ]

 #Subsetting de : id_estacion_origen, #id_estacion_destino, mes, día, hora, dia_de_la_semana, temp, hum, presion, viento, dias_no_lab, dist_total, velociad km/h, precipitación mm

Ecobici2 <- Ecobici[c(3,6,9,10,11,15,16,17,18,19,20,25,30,31)]

Ecobici2[is.na(Ecobici2)]<-0

Ecobici2$id_estacion_destino=as.factor(Ecobici2$id_estacion_destino)

Ecobici2$id_estacion_origen=as.factor(Ecobici2$id_estacion_origen)

Ecobici2$Mes=as.factor(Ecobici2$Mes)

Ecobici2$Dia=as.factor(Ecobici2$Dia)

Ecobici2$Dia_de_la_semana=as.factor(Ecobici2$Dia_de_la_semana)

Ecobici2$Hora=as.factor(Ecobici2$Hora)

Ecobici2$dias.no.lab=as.factor(Ecobici2$dias.no.lab)

Ecobici2=Ecobici2[Ecobici2$Presion.hPa!=0,]

### unimos las estaciones de retiro en una sola

Ecobici2[Ecobici2$id_estacion_origen==130,1]=2

Ecobici2[Ecobici2$id_estacion_destino==130,2]=2

Ecobici2[Ecobici2$id_estacion_origen==131,1]=2

Ecobici2[Ecobici2$id_estacion_destino==131,2]=2

rm(Ecobici)


sum(Ecobici2$id_estacion_origen==130)

sum(Ecobici2$id_estacion_destino==130)

sum(Ecobici2$id_estacion_origen==131)

sum(Ecobici2$id_estacion_destino==131)


##  abrimos los archivos de grupos de estaciones y tipos de estaciones
```

```
setwd("C:/Users/Pedro/Desktop/Facultad 2020/Primer cuatrimestre/91.54 Ciencia de datos para la toma de decisiones/Trabajo práctico")

Grupos_estaciones=read.csv("Estaciones.csv") ## grupos de estaciones por clusterización jerárquica según la latitud y la longitud.

setwd("C:/Users/Pedro/Desktop/Facultad 2020/Primer cuatrimestre/91.54 Ciencia de datos para la toma de decisiones/Trabajo
práctico/Datos")

horarios_estaciones=read.csv("horarios de las estaciones.csv") ## caracteriza los tipos de estaciones según el horario.

## generamos las nuevas variables que se usarán para el merge

Grupos_estaciones=Grupos_estaciones[,c(2,5)]

tipo_estacion_origen=horarios_estaciones[,c(2,6)]

tipo_estacion_destino=horarios_estaciones[,c(2,10)]

rm(horarios_estaciones)

 ## unimos los datasets

Ecobici2=merge(Ecobici2,Grupos_estaciones,by.x="id_estacion_origen",by.y = "id_estacion",all.x = T)

colnames(Ecobici2)=c("id_estacion_origen" , "id_estacion_destino" , "Mes" ,  "Dia" ,

                     "Hora"  , "Dia_de_la_semana"  ,  "Temp.C" , "Hum." ,

                     "Presion.hPa"  , "viento.Km.h" , "dias.no.lab" , "Dist.total" ,

                     "Velocidad.km.h" , "Precipitacion.mm" ,"groupo_origen" )


## nota: ¡ver que acá hace una cosa medio rara y cambia los nombres! Por eso volvemos a renombrar las columnas.

Ecobici2=merge(Ecobici2,Grupos_estaciones,by.x="id_estacion_destino",by.y = "id_estacion",all.x = T)

colnames(Ecobici2)=c("id_estacion_origen" , "id_estacion_destino" , "Mes" ,  "Dia" ,

                     "Hora"  , "Dia_de_la_semana"  ,  "Temp.C" , "Hum." ,

                     "Presion.hPa"  , "viento.Km.h" , "dias.no.lab" , "Dist.total" ,

                     "Velocidad.km.h" , "Precipitacion.mm" ,"groupo_destino" , "groupo_origen" )


## asigno los tipos de estación

Ecobici2=merge(Ecobici2,tipo_estacion_origen,by.x="id_estacion_origen",by.y = "id",all.x = T)


## nota: ¡ver que acá hace una cosa medio rara y cambia los nombres! Por eso volvemos a renombrar las columnas.

Ecobici2=merge(Ecobici2,tipo_estacion_destino,by.x="id_estacion_destino",by.y = "id",all.x = T)

colnames(Ecobici2)=c("id_estacion_origen" , "id_estacion_destino" , "Mes" ,  "Dia" ,

                     "Hora"  , "Dia_de_la_semana"  ,  "Temp.C" , "Hum." ,

                     "Presion.hPa"  , "viento.Km.h" , "dias.no.lab" , "Dist.total" ,

                     "Velocidad.km.h" , "Precipitacion.mm" ,"grupo_origen" ,

                     "grupo_destino","tipo_estacion_destino","tipo_estacion_origen" )

## separamos las variables categóricas y continuas

Ecobici2.cat=cbind.data.frame(Ecobici2$id_estacion_origen,

                              Ecobici2$id_estacion_destino,

                              Ecobici2$Mes,
```

```
                    Ecobici2$Dia,
                    Ecobici2$Hora,
                    Ecobici2$Dia_de_la_semana,
                    Ecobici2$dias.no.lab,
                    Ecobici2$grupo_origen,
                    Ecobici2$grupo_destino,
                    Ecobici2$tipo_estacion_origen,
                    Ecobici2$tipo_estacion_destino
)
colnames(Ecobici2.cat)=c("id_estacion_origen" , "id_estacion_destino" , "Mes" ,  "Dia" ,
                "Hora"  , "Dia_de_la_semana"  , "dias.no.lab" , "grupo_origen" ,
                "grupo_destino","tipo_estacion_origen","tipo_estacion_destino" )
 Ecobici2.cont=cbind.data.frame(Ecobici2$Temp.C,
                     Ecobici2$Hum.,
                     Ecobici2$Presion.hPa,
                     Ecobici2$viento.Km.h,
                     Ecobici2$Dist.total,
                     Ecobici2$Velocidad.km.h,
                     Ecobici2$Precipitacion.mm)
colnames(Ecobici2.cont)= c("Temp.C","Hum.","Presion.hPa","viento.Km.h",
                   "Dist.total","Velocidad.km.h","Precipitacion.mm")
 write.csv(Ecobici2,file="Ecobici2.csv",sep=",")
write.csv(Ecobici2.cat,file="Ecobici2_cat.csv",sep=",")
write.csv(Ecobici2.cont,file="Ecobici2_cont.csv",sep=",")
```

## Análisis de correspondencia

```
## tomamos las variables categóricas y realizamos el análisis de correspondencias
setwd("C:/Users/Pedro/Desktop/Facultad 2020/Primer cuatrimestre/91.54 Ciencia de datos para la toma de decisiones/Trabajo práctico/Datos")
Ecobici2_cat=read.csv("Ecobici2_cat.csv")
Ecobici2_cat=Ecobici2_cat[,-1]
Ecobici2_cat$id_estacion_origen=as.factor(Ecobici2_cat$id_estacion_origen)
Ecobici2_cat$id_estacion_destino=as.factor(Ecobici2_cat$id_estacion_destino)
Ecobici2_cat$Mes=as.factor(Ecobici2_cat$Mes)
Ecobici2_cat$Dia=as.factor(Ecobici2_cat$Dia)
Ecobici2_cat$Hora=as.factor(Ecobici2_cat$Hora)
Ecobici2_cat$Dia_de_la_semana=as.factor(Ecobici2_cat$Dia_de_la_semana)
```

```
Ecobici2_cat$dias.no.lab=as.factor(Ecobici2_cat$dias.no.lab)

Ecobici2_cat$grupo_origen=as.factor(Ecobici2_cat$grupo_origen)

Ecobici2_cat$grupo_destino=as.factor(Ecobici2_cat$grupo_destino)

Ecobici2_cat$tipo_estacion_origen=as.factor(Ecobici2_cat$tipo_estacion_origen)

Ecobici2_cat$tipo_estacion_destino=as.factor(Ecobici2_cat$tipo_estacion_destino)

 analisis=Ecobici2_cat[,-c(1:2)]

rm(Ecobici2_cat)

 library(factoextra)

library(FactoMineR)

## aumentamos la memoria disponible para R en la computadora, para que pueda procesar el MCA

 memory.limit(900000000)

 ## no correr a menos de que se pueda correr máquina sin usar durante 24 hs aprox

mcorresp = MCA(analisis,ncp = 10)

 ## cargar esto, que es el workspace del mca.

load("mcorresp.RData")

 ## screeplot

fviz_screeplot(mcorresp,addlabels=T,ncp=103,main="proporción de la varianza explicada por las variables del análisis de correspondencia múltiple")

fviz_screeplot(mcorresp,addlabels=T,ncp=30,main="proporción de la varianza explicada por las variables del análisis de correspondencia múltiple")

 mfcol=c(2,5)

 #### gráfico de variables explicadas

plot.MCA(mcorresp,c(1,2), choix="var",cex=0.45)

## Dimensión 2: explica Día de la seman y día no laborable, los cuales están muy relacionados (sábado y domingo priorizan sobre los otros)

## Dimensión 1: agrupa las estaciones de origen y destino marcando que hay una relación entre el grupo (división geográfica) y el tipo (división horaria)

plot.MCA(mcorresp,c(1,3), choix="var",cex=0.45)

## Dimensión 3: diferencia el grupo de estaciones del tipo de estaciones

## Dimensión 8 y 10: muestra un poco la varianza del día y el mes, junto con algo del día de la semana

plot.MCA(mcorresp,c(2,8), choix="var",cex=0.45)

## Dimensión 9: grupo origen y destino

plot.MCA(mcorresp,c(1,9), choix="var",cex=0.45)


plot.MCA(mcorresp,c(8,10), choix="var",cex=0.45)

 mean(mcorresp[["eig"]][,1])

mcorresp[["eig"]][1:10,1]

## variables para guardar más adelante

mcorresp.ind.coord= mcorresp[["ind"]][["coord"]]

mcorresp.ind.contrib= mcorresp[["ind"]][["contrib"]]
```

```
mcorresp.ind.cos2= mcorresp[["ind"]][["cos2"]]
mcorresp.eig= mcorresp[["eig"]]
mcorresp.var.coord= mcorresp[["var"]][["coord"]]
mcorresp.var.contrib= mcorresp[["var"]][["contrib"]]
mcorresp.var.cos2= mcorresp[["var"]][["cos2"]]
mcorresp.var.vtest= mcorresp[["var"]][["v.test"]]
mcorresp.var.eta2= mcorresp[["var"]][["eta2"]]
mcorresp.svd.vs= mcorresp[["svd"]][["vs"]]
mcorresp.svd.u= mcorresp[["svd"]][["U"]]
mcorresp.svd.v= mcorresp[["svd"]][["V"]]
 ## guardamos las variables
write.csv(mcorresp.ind.coord,file="mcorresp_ind_coord.csv")
write.csv(mcorresp.ind.contrib,file="mcorresp_ind_contrib.csv")
write.csv(mcorresp.ind.cos2,file="mcorresp_ind_cos2.csv")
write.csv(mcorresp.eig,file="mcorresp_eig.csv")
write.csv(mcorresp.var.coord,file="mcorresp_var_coord.csv")
write.csv(mcorresp.var.contrib,file="mcorresp_var_contrib.csv")
write.csv(mcorresp.var.cos2,file="mcorresp_var_cos2.csv")
write.csv(mcorresp.var.vtest,file="mcorresp_var_vtest.csv")
write.csv(mcorresp.var.eta2,file="mcorresp_var_eta2.csv")
write.csv(mcorresp.svd.vs,file="mcorresp_svd_vs.csv")
write.csv(mcorresp.svd.u,file="mcorresp_svd_u.csv")
write.csv(mcorresp.svd.v,file="mcorresp_svd_v.csv")

rm(Ecobici2_cat)
rm(mcorresp.eig)
rm(mcorresp.ind.contrib)
rm(mcorresp.ind.coord)
rm(mcorresp.ind.cos2)
rm(mcorresp.svd.u)
rm(mcorresp.svd.v)
rm(mcorresp.var.contrib)
rm(mcorresp.var.coord)
rm(mcorresp.var.cos2)
rm(mcorresp.var.eta2)
rm(mcorresp.svd.vs)
rm(mcorresp.var.vtest)
```

```
mcorresp.ind.coord= mcorresp[["ind"]][["coord"]]

## cargamos el otro data frame, y luego los unimos

Ecobici2=read.csv("Ecobici2.csv")


summary(mcorresp.ind.coord)

mcorresp.ind.coord=as.data.frame(mcorresp.ind.coord)


Base_de_datos_Ecobici=cbind.data.frame(Ecobici2,mcorresp.ind.coord)

write.csv(Base_de_datos_Ecobici,file = "Base_de_datos_Ecobici.csv") ## base de datos final a la que le haremos la clusterización

rm(Ecobici2)

rm(mcorresp.ind.coord)
```

## Determinación del número de clusters óptimos

```
x=matrix(ncol=2,nrow=40)

 x=rbind(z)

colnames(x)=c("Number of clusters K", "Average silhouette width")

i=0

library(cluster)

set.seed(1)

for (i in 1:22){

 clarax= clara(Ecobici2,i,metric = "euclidean",pamLike = T,stand = T,correct.d = T,sampsize = 10000,samples=10)

 x[i,1]=i

  if (i==1)

   x[i,2]=0

 else

   x[i,2]=clarax$silinfo[[3]]

 print(i)

}

## obs: no llegamos a los 40 clusters debido a que a medida que aumentaba el número de clusters, se tardaba cada vez más en hacer la corrida. De ahí que nos quedamos con los primeros 22 para el análisis

 plot(x)

 write.csv2(x, file="X.csv",sep=",")

## análisis jerárquico de los viajes para determinar la cantidad de grupos


Base_de_datos_Ecobici=read.csv("Base_de_datos_Ecobici.csv")

Base_de_datos_Ecobici=Base_de_datos_Ecobici[,-c(1,2)]


analisis=Base_de_datos_Ecobici[,c(7:10,12:14,19:21)]
```

```
X=sample(2517238,10000,replace=F)


datasclu=analisis[X,]

datasclu.stand=scale(datasclu)

d <- dist(datasclu.stand, method = "euclidean")


cluh <- hclust(d, method="complete")

plot(cluh,labels=F) # display dendrogram

# formo un vector que dice a que grupo pertenece cada punto


rect.hclust(cluh, k=5, border="blue")

rect.hclust(cluh, k=10, border="red")

setwd("C:/Users/Pedro/Desktop/Facultad 2020/Primer cuatrimestre/91.54 Ciencia de datos para la toma de decisiones/Trabajo práctico/Datos")

Base_de_datos_Ecobici=read.csv("Base_de_datos_Ecobici.csv")

Base_de_datos_Ecobici=Base_de_datos_Ecobici[,-c(1,2)]


analisis=Base_de_datos_Ecobici[,c(7:10,12:14,19:21)]

analisis.stand=scale(analisis)



library (cluster)


R2=vector()


for (K in 1:20) {

 clu <- kmeans(analisis.stand, K ,nstart=20,iter.max=1000, algorithm = "Lloyd")

 R2[K]=1-sum(clu$withinss)/clu$totss

 print(K)

}

plot(R2)

write.csv(R2,"R2 final.csv")
```

## Clusterización por K-means

write.csv(R2,"R2 final.csv")

```
## tp k-means
```

```
setwd("C:/Users/Pedro/Desktop/Facultad 2020/Primer cuatrimestre/91.54 Ciencia de datos para la toma de decisiones/Trabajo práctico/Datos")

Base_de_datos_Ecobici=read.csv("Base_de_datos_Ecobici.csv")

Base_de_datos_Ecobici=Base_de_datos_Ecobici[,-c(1,2)]


analisis=Base_de_datos_Ecobici[,-c(1:6,11,15:18)]


analisis$Temp.C=as.vector(analisis$Temp.C)

analisis$Temp.C=scale(analisis$Temp.C)

analisis$Hum.=as.vector(analisis$Hum.)

analisis$Hum.=scale(analisis$Hum.)

analisis$Presion.hPa=as.vector(analisis$Presion.hPa)

analisis$Presion.hPa=scale(analisis$Presion.hPa)

analisis$viento.Km.h=as.vector(analisis$viento.Km.h)

analisis$viento.Km.h=scale(analisis$viento.Km.h)

analisis$Dist.total=as.vector(analisis$Dist.total)

analisis$Dist.total=scale(analisis$Dist.total)

analisis$Velocidad.km.h=as.vector(analisis$Velocidad.km.h)

analisis$Velocidad.km.h=scale(analisis$Velocidad.km.h)

analisis$Precipitacion.mm=as.vector(analisis$Precipitacion.mm)

analisis$Precipitacion.mm=scale(analisis$Precipitacion.mm)

analisis$Dim.1=scale(analisis$Dim.1)

analisis$Dim.2=scale(analisis$Dim.2)

analisis$Dim.3=scale(analisis$Dim.3)

analisis$Dim.4=scale(analisis$Dim.4)

analisis$Dim.5=scale(analisis$Dim.5)

analisis$Dim.6=scale(analisis$Dim.6)

analisis$Dim.7=scale(analisis$Dim.7)

analisis$Dim.8=scale(analisis$Dim.8)

analisis$Dim.9=scale(analisis$Dim.9)

analisis$Dim.10=scale(analisis$Dim.10)

str(analisis)


analisis=analisis[,1:10]



K=10
```

```
Kclu <- kmeans(analisis, K ,nstart=25,iter.max=1000, algorithm = "Lloyd",trace=T)
1-sum(Kclu$withinss)/Kclu$totss

grupos=Kclu$cluster

Base_de_datos_Ecobici_10=cbind(Base_de_datos_Ecobici,grupos)

grupo1=Base_de_datos_Ecobici_10[Base_de_datos_Ecobici_10$grupos==1,]
grupo2=Base_de_datos_Ecobici_10[Base_de_datos_Ecobici_10$grupos==2,]
grupo3=Base_de_datos_Ecobici_10[Base_de_datos_Ecobici_10$grupos==3,]
grupo4=Base_de_datos_Ecobici_10[Base_de_datos_Ecobici_10$grupos==4,]
grupo5=Base_de_datos_Ecobici_10[Base_de_datos_Ecobici_10$grupos==5,]
grupo6=Base_de_datos_Ecobici_10[Base_de_datos_Ecobici_10$grupos==6,]
grupo7=Base_de_datos_Ecobici_10[Base_de_datos_Ecobici_10$grupos==7,]
grupo8=Base_de_datos_Ecobici_10[Base_de_datos_Ecobici_10$grupos==8,]
grupo9=Base_de_datos_Ecobici_10[Base_de_datos_Ecobici_10$grupos==9,]
grupo10=Base_de_datos_Ecobici_10[Base_de_datos_Ecobici_10$grupos==10,]
```

## Clusterización por CLARA

```
Base_de_datos_Ecobici=read.csv("Base_de_datos_Ecobici.csv")
Base_de_datos_Ecobici=Base_de_datos_Ecobici[,-c(1,2)]
## seleccionamos las variables que vamos a usar para la segmentación. No nos importan las variables DIM del mca
analisis=Base_de_datos_Ecobici[,c(3:18)]
 ## definimos el formato de las variables y las escalamos
analisis$Temp.C=as.vector(analisis$Temp.C)
analisis$Temp.C=scale(analisis$Temp.C)
analisis$Hum.=as.vector(analisis$Hum.)
analisis$Hum.=scale(analisis$Hum.)
analisis$Presion.hPa=as.vector(analisis$Presion.hPa)
analisis$Presion.hPa=scale(analisis$Presion.hPa)
analisis$viento.Km.h=as.vector(analisis$viento.Km.h)
analisis$viento.Km.h=scale(analisis$viento.Km.h)
analisis$Dist.total=as.vector(analisis$Dist.total)
analisis$Dist.total=scale(analisis$Dist.total)
analisis$Velocidad.km.h=as.vector(analisis$Velocidad.km.h)
analisis$Velocidad.km.h=scale(analisis$Velocidad.km.h)
analisis$Precipitacion.mm=as.vector(analisis$Precipitacion.mm)
```

```
analisis$Precipitacion.mm=scale(analisis$Precipitacion.mm)
 ## variables categóricas
analisis$Mes=as.factor(analisis$Mes)
analisis$Dia=as.factor(analisis$Dia)
analisis$Hora=as.factor(analisis$Hora)
analisis$Dia_de_la_semana=as.factor(analisis$Dia_de_la_semana)
analisis$dias.no.lab=as.factor(analisis$dias.no.lab)
analisis$grupo_origen=as.factor(analisis$grupo_origen)
analisis$grupo_destino=as.factor(analisis$grupo_destino)
analisis$tipo_estacion_destino=as.factor(analisis$tipo_estacion_destino)
analisis$tipo_estacion_origen=as.factor(analisis$tipo_estacion_origen)

 ## librería para la clusterización por CLARA de los viajes
library(cluster)
k=10
clarax= clara(analisis,k,metric = "euclidean",pamLike = T,stand=T,correct.d = T,sampsize = 10000,samples=50,trace = T)
## vector de clusters correspondientes a cada viaje
 gruposcl=clarax$clustering
##  lo unimos a la base de datos principal
Base_de_datos_Ecobici_10_cl=cbind(Base_de_datos_Ecobici,gruposcl)
 ## definimos df para cada grupo por separado
grupo1=Base_de_datos_Ecobici_10_cl[Base_de_datos_Ecobici_10_cl$grupos==1,]
grupo2=Base_de_datos_Ecobici_10_cl[Base_de_datos_Ecobici_10_cl$grupos==2,]
grupo3=Base_de_datos_Ecobici_10_cl[Base_de_datos_Ecobici_10_cl$grupos==3,]
grupo4=Base_de_datos_Ecobici_10_cl[Base_de_datos_Ecobici_10_cl$grupos==4,]
grupo5=Base_de_datos_Ecobici_10_cl[Base_de_datos_Ecobici_10_cl$grupos==5,]
grupo6=Base_de_datos_Ecobici_10_cl[Base_de_datos_Ecobici_10_cl$grupos==6,]
grupo7=Base_de_datos_Ecobici_10_cl[Base_de_datos_Ecobici_10_cl$grupos==7,]
grupo8=Base_de_datos_Ecobici_10_cl[Base_de_datos_Ecobici_10_cl$grupos==8,]
grupo9=Base_de_datos_Ecobici_10_cl[Base_de_datos_Ecobici_10_cl$grupos==9,]
grupo10=Base_de_datos_Ecobici_10_cl[Base_de_datos_Ecobici_10_cl$grupos==10,]
write.csv(grupo1,file="Grupo1 - cl.csv")
write.csv(grupo2,file="Grupo2 - cl.csv")
write.csv(grupo3,file="Grupo3 - cl.csv")
write.csv(grupo4,file="Grupo4 - cl.csv")
write.csv(grupo5,file="Grupo5 - cl.csv")
write.csv(grupo6,file="Grupo6 - cl.csv")
write.csv(grupo7,file="Grupo7 - cl.csv")
```

```
write.csv(grupo8,file="Grupo8 - cl.csv")

write.csv(grupo9,file="Grupo9 - cl.csv")

write.csv(grupo10,file="Grupo10 - cl.csv")

write.csv(Base_de_datos_Ecobici_10_cl,file="Base de datos con 10 grupos cl.csv")
```