Introducción

Este proyecto tiene como objetivo implementar uno o más algoritmos de machine learning a una serie de datos y sacar ciertas conclusiones.

El enfoque que le he querido dar es alrededor de los distritos del municipio de Madrid y las **operaciones del SAMUR y Protección Civil**. Se han recopilado datos sobre los distritos en base a este tema con el objetivo de realizar un Análisis de Componentes Principales y un Análisis Cluster.

Este documento recogerá todo el análisis de los datos y las conclusiones de los análisis de componentes principales y cluster. Para consultar el código correspondiente, mirar los anexos al final del documento. Ahí vendrá todo explicado.

Objetivo

El objetivo de este trabajo es realizar una análisis cluster, un análisis no supervisado, que nos ayude a determinar patrones en las operaciones del SAMUR y Protección Civil en base a distritos y código de incidencia. Para ello aplicaré el algoritmo k-means a dos datasets diferentes, el primero será todas las incidencias agrupadas por distrito. Este dataset tendrá en total 21 registros, uno por distrito. El segundo dataset serán las activaciones agrupadas por incidencias, con un total de 100 registros (al final se quedarán en 99, luego se verá en detalle por qué).

Contenido del proyecto

Este proyecto contiene los siguientes archivos

- **AB-Unit25-MachineLearning.pdf**: este informe detallando todo el proceso y las conclusiones
- **anexo1-transformacionDeDatos.Rmd**: código de la transformación y su html para una mejor lectura.
- **anexo2-pcaYClusterAgrupadosPorDistritos.Rmd**: código del PCA y Análisis Clúster por distrito y su html para una mejor lectura.
- **anexo3-pcaYClusterAgrupadosPorIncidencia.Rmd**: código del PCA y Análisis Clúster por Incidencias y su html para una mejor lectura.
- **/data**: carpeta con todos los datasets a utilizar.
- **detenctEncoding.py**: código python para detectar la codificación de los archivos.

En el caso de que se quiera acceder vía repositorio, todos los archivos de código y csv pertinentes al proyecto estarán subidos en el repositorio del siguiente enlace.

https://github.com/Araaancg/machine-learning-SAMUR

Presentación de los datos

Para hacer posible este proyecto se han recopilado un total de 8 archivos csv. 7 archivos son correspondientes a las activaciones del SAMUR y Protección Civil entre los años 2017 y 2023 (un archivo por año) y el último contiene información sobre los diferentes distritos del municipio de Madrid. Ambos archivos han sido obtenidos de los portales abiertos del Ayuntamiento y Comunidad de Madrid respectivamente. Para más información, consultar las referencias.

CSV de las Activaciones del SAMUR y Protección Civil

Entre los 7 archivos disponibles contamos con poco menos de 1 millón de registros.

Para que nos entendamos mejor, una activación la define el proveedor de los datos como "encargos de asistencia sanitaria que supone la activación de un recurso sanitario u otro tipo de vehículo"

Los CSV contienen las siguientes columnas:

- **Año**: Año en el que ocurrió la activación.
- **Mes**: Mes en el que ocurrió la activación.
- **Hora Solicitud**: Hora en la cual el recurso fue activado.
- **Hora Intervención**: Hora en la cual el recurso ha llegado a la escena.
- **Código**: Código asignado a la activación en función de la descripción que hace el demandante de la escena.
- **Distrito**: Distrito del municipio de Madrid en donde ha ocurrido la emergencia.
- **Hospital**: Nombre del hospital al cual se ha trasladado el paciente.

Es importante destacar las siguientes observaciones (descritas por el proveedor de los datos):

- Un recurso puede ser cancelado una vez se ha activado, ya sea por el demandante o porque se ha encontrado un recurso más cercano. La activación queda registrada en el csv pero no se registra una hora de intervención, en cambio se deja este campo vacío.
- Si, por ejemplo, en un accidente de coche se requieren 3 ambulancias para atender a 3 pacientes, se registran tres activaciones (correspondientes a los tres recursos enviados a la escena)
- Existen 100 códigos posibles para la columna "Código", todos descritos por el proveedor de los datos en un csv aparte y que adjuntaré en las referencias.
- En el campo distrito, no solo se registran activaciones en los 21 distritos del municipio de Madrid, sino que también se registran los siguientes casos: Carreteras

y Circunvalaciones, Pozuelo, Leganés, C.A.M. (Comunidad Autónoma de Madrid), y Fuera del Término Municipal.

- Un paciente puede no requerir asistencia hospitalaria y en ese caso se dejaría el campo hospital vacío.

CSV con información sobre los Distritos del Municipio de Madrid

Se trata de un csv pequeño, de apenas 21 filas, con información sobre los distintos distritos del municipio de Madrid. Estos serán los distritos con los que trabajaré a lo largo de esta parte del proyecto.

Los distritos son: Arganzuela, Barajas, Carabanchel, Centro, Chamartín, Chamberí, Ciudad Lineal, Fuencarral, Hortaleza, Latina, Moncloa, Moratalaz, Puente de Vallecas, Retiro, Salamanca, San Blas, Tetuán, Usera, Vicálvaro, Villa de Vallecas y Villaverde.

El CSV contiene las siguientes columnas:

- **distrito_codigo**: Código del distrito
- **distrito_nombre**: Nombre del distrito
- **municipio_codigo**: Código del municipio al que pertenece el distrito
- **municipio_nombre**: Nombre del municipio
- **superficie_km2**: Superficie en kilómetros cuadrados del distrito
- **densidad_por_km2**: Densidad de población por kilómetro cuadrado del distrito

Transformación de datos

La transformación de datos ha sido un proceso crucial para poder realizar los algoritmos de machine learning. Para los datasets del SAMUR había que procesar todos los datos obtenidos de una forma que nos sirvieran más tarde para realizar los algoritmos y por ello hice la siguiente lista definiendo los objetivos de esta transformación:

- Averiguar la codificación de los archivos para no tener problemas al leer los csv.
- Crear una columna nueva que indique el tiempo de respuesta del recurso. Básicamente el tiempo entre la hora de solicitud y la hora de intervención.
- Cambiar los nombres de las columnas por dos razones: unificar sobre todo la columna distrito, ya que será el ancla entre nuestros csv y para evitar tildes y caracteres raros que puedan entorpecer el programa.
- El campo hospital como tal no nos interesa, ya que el hospital elegido puede tener mucho que ver con la ubicación de la emergencia. En cambio, he decidido cambiar

esta columna a una que indique si el paciente fue hospitalizado o no. Importante tener en cuenta que si el recurso fue cancelado, el registro tendrá esta columna vacía pero no por ausencia de hospitalización, sino por cancelación del recurso.

- Cambiar los nombres de los distritos para que coincidan en todos los dataframes (samur-20xx e info-distritos)
- Juntar todos los archivos del SAMUR en uno solo.

La codificación de los archivos la averigüé a través de un pequeño y simple programa de python, el cual se encontrará como anexo 1. El programa analizaba todos los csv obtenidos y como output soltaba la codificación de cada uno. El resultado fue el siguiente: el csv de información de distritos y los primeros 3 años (2017-2019) de activaciones del SAMUR tienen la codificación ISO-8859-1, mientras que el resto de csv (últimos 4 años, 2020-2023) tienen UTF-8-SIG. Sorprendentemente, los archivos del SAMUR no son homogéneos.

Una vez tenemos la codificación de los datos procedemos a la transformación, expuesta en el anexo 2.

Por otro lado, la transformación del dataset de los distritos ha sido mucho más leve. En este dataset nada más que había que quedarse con las columnas de distrito, superficie en km2 y la densidad de población. Se cambiaron los nombres de los distritos, al igual que en el dataset del SAMUR, para poder juntarlos de forma simple en pasos posteriores.

Con estos dos dataframes aún no podemos trabajar con las ideas propuestas. Los datos que necesitamos deben estar agrupados por distrito y por incidencia (columna código) por lo que antes de dar por terminada esta transformación, mostraré el proceso de agrupamiento y que columnas he creado nuevas.

Datos agrupados por distritos

Este dataframe contendrá las siguientes columnas:

- **distrito**: Nombre del distrito.
- **tie_medio_intervencion**: Media del tiempo de intervención medida en segundos.
- **activaciones_totales**: Número de activaciones totales registradas a ese distrito.
- **activaciones_canceladas**: Ratio de activaciones canceladas en ese distrito en función de las activaciones totales del distrito.
- **hospital**: Ratio de hospitalización en función de las activaciones realizadas.
- **superficie**: Superficie total del distrito medida en kilómetros cuadrados.
- **densidad_poblacion:** Densidad de población por cada kilómetro cuadrado.

El primer paso fue agrupar las intervenciones, y posteriormente se juntó esta información con el dataframe de información de distritos. Al agrupar el dataset del SAMUR he decidido

quedarme con todos aquellos que tengan la columna distrito rellenada y sea uno de los 21 distritos oficiales del municipio. Todas aquellas activaciones que tengan otro valor asignado, como por ejemplo "CARRETERAS Y CIRCUNVALACIONES" o "LEGANES" no se utilizarán. También lo voy a juntar con el dataframe que contiene la información básica de los distritos.

Datos agrupados por incidencia (código)

Este dataframe tendrá las siguientes columnas:

- **codigo**: Código de emergencia
- **tie_medio_intervencion**: Media del tiempo de intervención.
- **activaciones_totales**: Número de activaciones totales registradas a ese código.
- **activaciones_totales**: Ratio de activaciones canceladas con ese código. En función de las activaciones totales del código.
- **hospital**: Ratio de hospitalización en función de las activaciones realizadas.
- **noche**: Ratio de activaciones realizadas por la noche. Considero noche todas las horas entre las 21:00 y las 9:00

He decidido prescindir de todos los registros que no tienen un código asignado, ya que no dan información relevante para este caso. Adicionalmente, investigando un poco me he dado cuenta de que el código "Pacientes en RCP Prolongada" solo ha ocurrido 1 vez en 7 años, y al parecer la activación fue cancelada ya que no disponemos de hora de intervención. Dado que este registro tampoco aporta información de valor en el análisis, he decidido prescindir de él también, por lo que nos quedarían 99 registros en total.

Todos los pasos y los dataframes finales se pueden consultar en el anexo 2.

Prueba 1. Datos agrupados por distrito

Todo el código pertinente a esta sección se encontrará en el anexo 2. En este informe solo se expondrá las conclusiones del análisis de componentes principales y las conclusiones del análisis cluster. Para consultar el análisis de los datos consultar el anexo 2.

Análisis de Componentes Principales (PCA)

Análisis de Componentes Principales (de ahora en adelante PCA por sus siglas en inglés) es una técnica de reducción de dimensionalidad. Su objetivo es simplificar conjuntos de datos complejos conservando la mayor cantidad posible de información. PCA coge un conjunto de datos con muchas características y variables y las transforma en un nuevo conjunto de variables llamadas componentes principales, ordenados en función de la cantidad de varianzas que explican.

En este caso, aplicaré PCA para ver si puedo reducir el número de columnas que nos encontramos a unos pocos componentes guardando la mayor información posible.

Antes de meternos en el proceso de PCA, es importante tener en cuenta que los datos están en escalas altamente diferentes, y esto podría llevar a casos como que una variable con mayor varianza arrastre al resto. Por eso mismo vamos a escalar las variables, en concreto todas menos las columnas hospital e intervenciones_canceladas.

Una vez hemos escalado, procedemos a aplicar el análisis de componentes principales.

	PC1	PC2	PC3	PC4	PC5	PC6
tie_medio_intervencion	0.564163111	0.083704129	0.55005330	-0.609850095	-0.012906083	-0.0084418542
activaciones_totales	-0.402816216	-0.703489648	0.58266427	0.056623442	-0.011161813	-0.0036000279
activaciones_canceladas	0.003641604	-0.001086692	0.00374743	-0.002787583	-0.197614621	0.9802612945
hospital	0.005727765	0.003797869	-0.01314379	0.017438905	-0.980038178	-0.1974868719
superficie	0.410695358	-0.702132254	-0.54363935	-0.206844973	0.003318962	-0.0001449231
densidad_poblacion	-0.592235033	0.071348631	-0.24942569	-0.762744989	-0.013094390	-0.0015760420

Resultado del Análisis de Componentes Principales. Dataframe de distritos.

```
Importance of components:
                          PC1    PC2    PC3     PC4     PC5     PC6
Standard deviation     1.8268 1.0664 0.8934 0.67228 0.47971 0.21230
Proportion of Variance 0.5562 0.1895 0.1330 0.07533 0.03835 0.00751
Cumulative Proportion  0.5562 0.7458 0.8788 0.95414 0.99249 1.00000
```

Importancia de los componentes. Dataframe de distritos.

Con los tres primeros componentes podemos explicar el 95% de la variabilidad. Incluso con los dos primeros ya tendríamos más de 3 tercios.

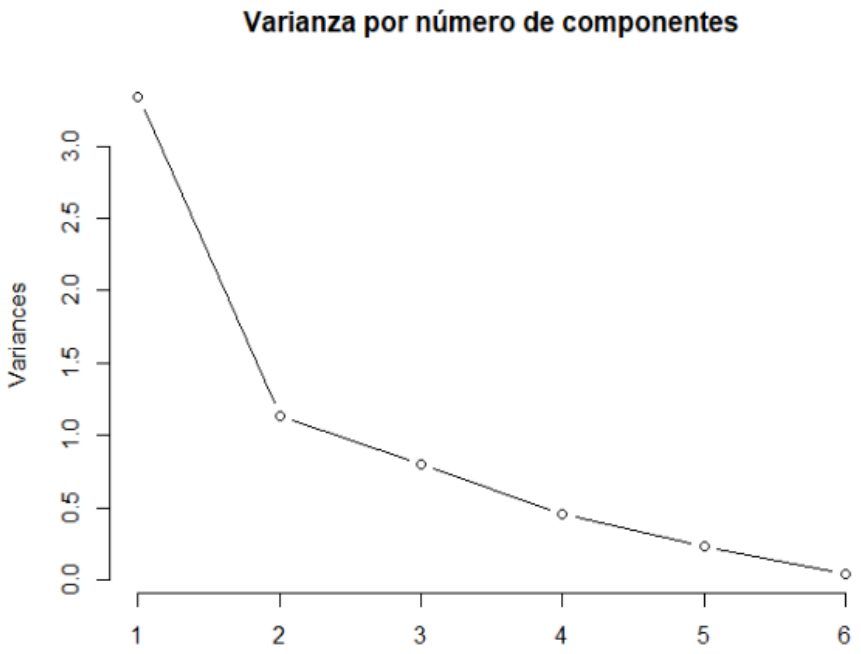
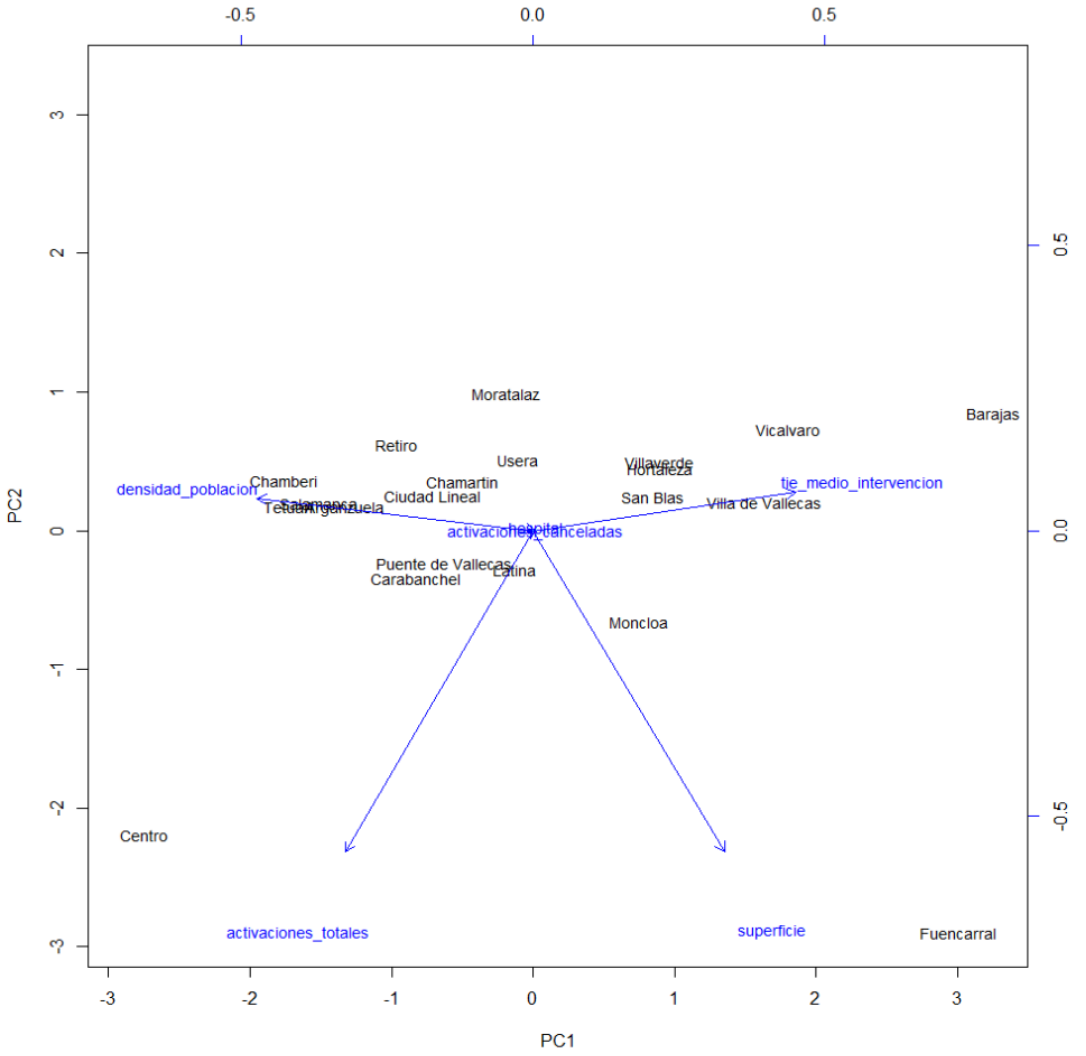


Gráfico mostrando la varianza por número de componentes. Dataframe de distritos



Representación del resultado del Análisis de Componentes Principales. Dataframe de distritos.

Conclusiones del resultado de PCA

El distrito Centro es uno de los más alejados. Como ya he comentado antes, es el que más activaciones tiene registradas y puede ser motivo por el que esté tirando en esa dirección.

El distrito Fuencarral está en la esquina inferior derecha, tirando fuertemente de la flecha de superficie y tiempo de intervención.

El distrito de Barajas es otro de los que están bastante dispersos. Antes hemos visto que es el distrito que mayor media de tiempo de intervención tiene y uno de los que menos densidad de población tiene también por lo que es normal que tire en esa dirección, completamente contraria a la densidad de población.

A la derecha, donde la flecha de densidad de población, se puede ver una nube de distritos, correspondientes a los 4 con mayor densidad de población: Chamberí, Tetuán, Salamanca y Arganzuela.

Puente de Vallecas y Carabanchel son los distritos que más intervenciones tienen después del Centro.

Análisis Cluster

El análisis cluster que voy a aplicar primero será el K-Means. K-Means es un algoritmo de agrupamiento de datos en diferentes clusters. Funciona iterativamente asignando puntos de datos al centroide más cercano y actualizando los centroides para minimizar la suma de las distancias cuadradas dentro de cada clúster. Es sensible a los valores atípicos y la elección de los centroides iniciales, y típicamente utiliza la métrica de distancia euclidiana para el agrupamiento.

La K es un número que representa en cuántos clusters queremos dividir los datos. Hay varios métodos para hacer una estimación del número de clústers óptimos para un set de datos. En mi caso, para hacer esta estimación voy a aplicar “Gap Statistic”. El Gap Statistic es un método que compara la dispersión intra-cluster (dentro de los grupos) obtenida con diferentes valores de K con la dispersión esperada bajo una distribución de referencia nula (sin agrupaciones evidentes). El valor óptimo de K se elige como aquel que maximiza la diferencia (el "gap") entre la dispersión observada y la esperada, lo que indica una estructura de agrupamiento significativamente diferente de una distribución aleatoria.

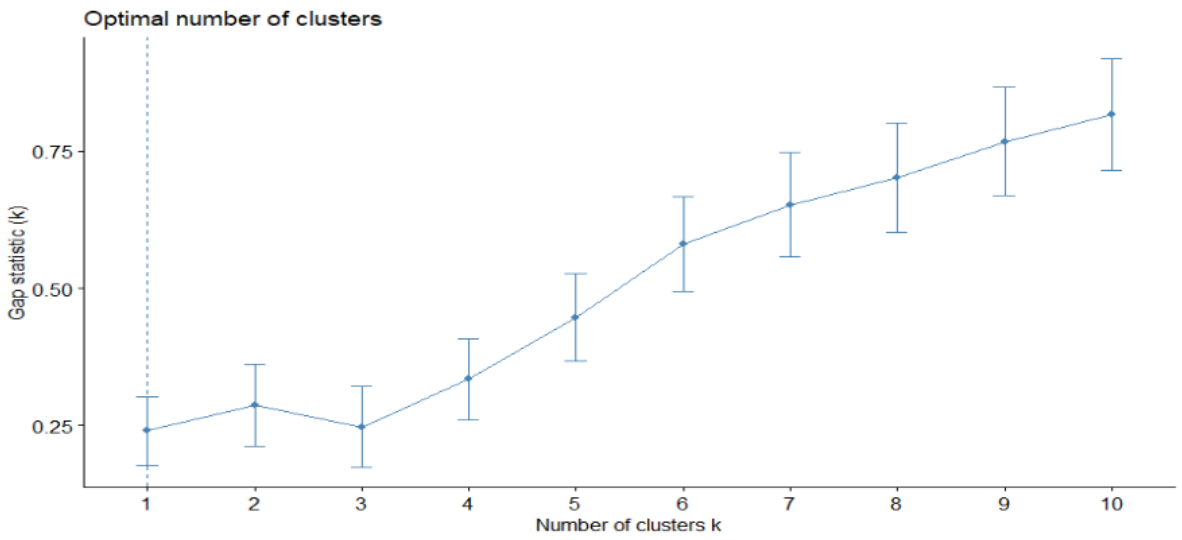


Gráfico Gap después de escalar los datos de distritos. K óptima 1

El Gap Statistic es una técnica utilizada para determinar el número óptimo de clusters en un algoritmo de clustering como k-means. Funciona comparando la dispersión intra-cluster (dentro de los grupos) de los datos reales con la dispersión esperada de datos generados aleatoriamente con la misma distribución.

¿Cómo funciona?

Datos reales: Se calcula la dispersión intra-cluster (por ejemplo, la suma de distancias al cuadrado de cada punto a su centroide) para diferentes valores de k (número de clusters).

Datos de referencia: Se generan datos aleatorios con la misma distribución que los datos reales y se calcula la dispersión intra-cluster para los mismos valores de k.

Gap Statistic: Se calcula la diferencia entre la dispersión intra-cluster de los datos reales y la esperada (promedio de las dispersiones de los datos de referencia) para cada valor de k. Esta diferencia es el Gap Statistic.

Interpretación del gráfico:

Hacia arriba: Cuando el Gap Statistic aumenta a medida que k aumenta, sugiere que añadir un cluster más mejora significativamente la estructura del clustering. Es decir, los datos se agrupan mejor con un k mayor.

Hacia abajo: Cuando el Gap Statistic disminuye a medida que k aumenta, indica que añadir más clusters no aporta una mejora significativa y puede ser indicativo de sobreajuste. Es decir, los clusters adicionales no reflejan una estructura real en los datos.

Dado que esta técnica se basa en las distancias entre los datos, en vez de escalar los datos (como hemos hecho en el pca) vamos a norma

Conclusiones del Gap Statistic

El número de clusters óptimo según esta estadística es 1. Esto se debe a que la variabilidad de los datos no debe dejar espacio para que añadir más de 1 clúster tenga sentido. Por eso mismo se ha optado por normalizar los datos, intentado cambiar distancias ya que este sistema se basa mucho en ellas.

Al normalizar los datos nos sale el siguiente gráfico.

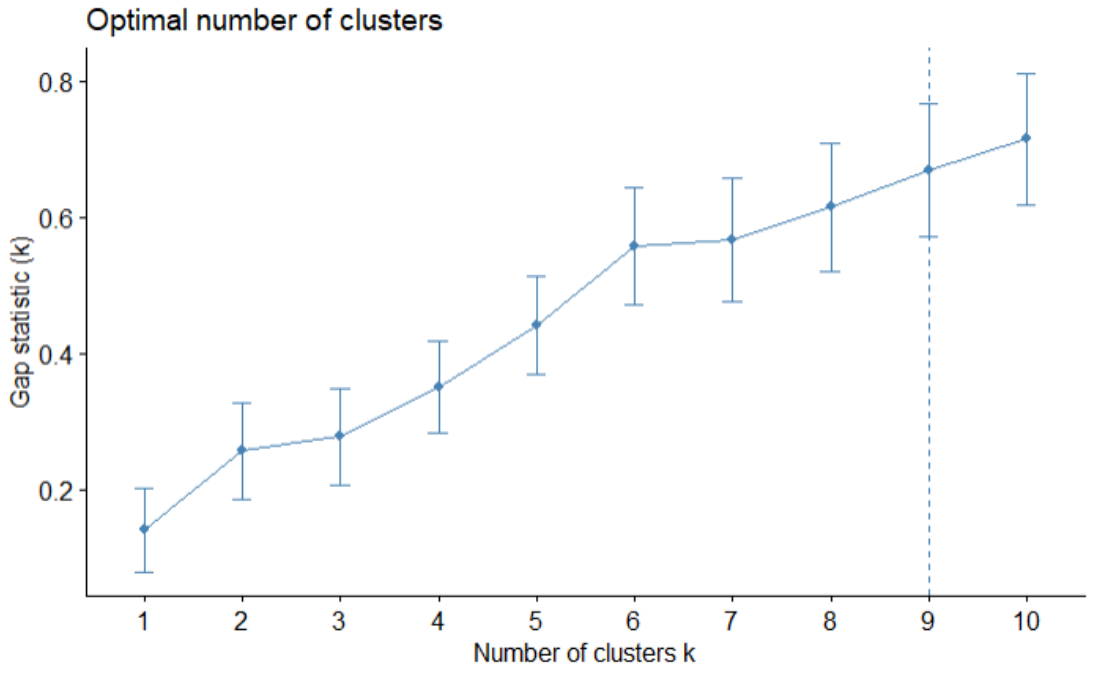


Gráfico Gap después de normalizar los datos. K óptima 9

Ahora nos sale que el valor óptimo para K es 9.

K-Means

Aplicamos el algoritmo K-Means con K = 9 y 25 sets random ya que es una medida bastante standard.

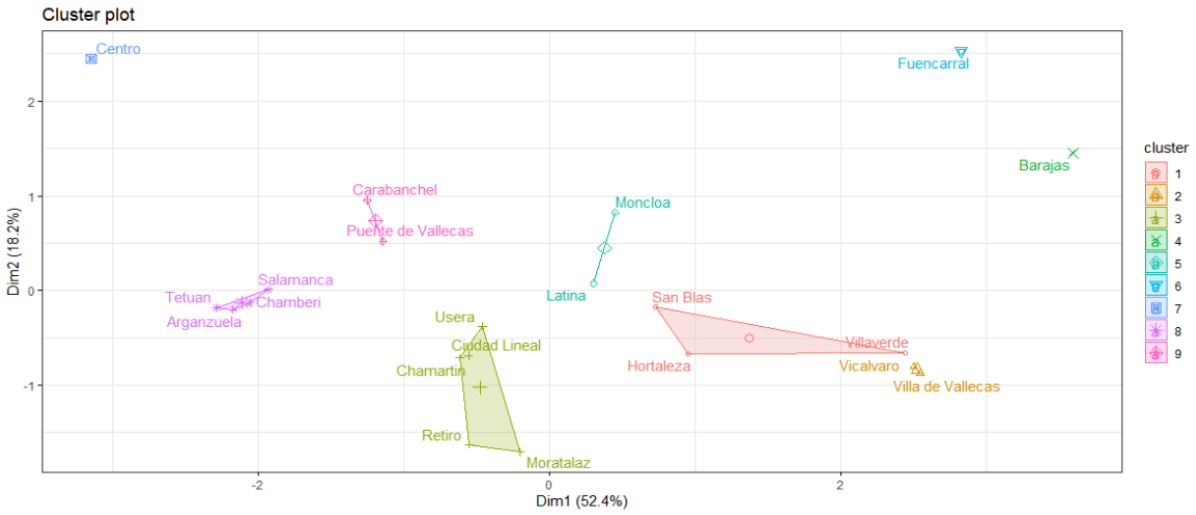


Imagen con los 9 clústers separando distritos

De estos 9 clústers podemos sacar las siguientes conclusiones:

- Los distritos **Centro, Barajas y Fuencarral** son distritos que forman su propio clúster ellos mismos. Centro es el distrito con más intervenciones. Fuencarral es el distrito más grande y uno con menos densidad de población. Barajas es otro distrito con muy poca densidad de población y con el mayor ratio de cancelación que hay.
- El cluster morado, con **Salamanca, Chamberí, Tetuán y Arganzuela** se caracteriza por ser los distritos con más densidad de población, además de tener una media de intervención bastante similar. Por lo demás, la verdad que no destacan en nada, su media de hospitalización está bastante en la media (nunca mejor dicho) y cancelación más de lo mismo.
- El clúster verde pistacho de abajo contiene los distritos de **Retiro, Chamartín, User, Ciudad Lineal y Moratalaz**. Son distritos que están en la media de densidad de población, no son los que más tienen pero no se quedan muy atrás de los antes mencionados y de los que menos ratio de intervenciones canceladas tienen. No parece que sean distritos que tengan mucho en común.
- **Carabanchel y Puente de Vallecas** conforman el clúster rosa y son los distritos después del Centro que más intervenciones tienen.
- **Moncloa y Latina** están en un clúster verde turquesa en el medio de la imagen. Este clúster la verdad que no tiene mucho sentido. No tienen nada en común más de lo analizado previamente más que están en números similares de intervenciones totales. El resto tienen valores muy dispersos.

- **Vicálvaro y Villa de Vallecas** son el clúster marrón naranja abajo a la derecha y conforman el clúster con mayor ratio de hospitalización que hay. Es verdad que les sigue muy de cerca Villaverde, pero este distrito es parte de otro clúster.
- **San Blas, Villaverde y Hortaleza**, estos tres clústers forman la nube roja. Son clústers con métricas muy muy similares prácticamente en todas las variables. Densidad de población, superficie, ratio de activaciones canceladas y media de tiempo de intervención muy similares. San Blas y Hortaleza no se alejan mucho en ratio de hospitalización, pero como ya he dicho antes, Villaverde se acerca mucho más a los dos distritos del clúster marrón naranja.

Para terminar con este análisis, vamos a ver el gráfico silhouette.

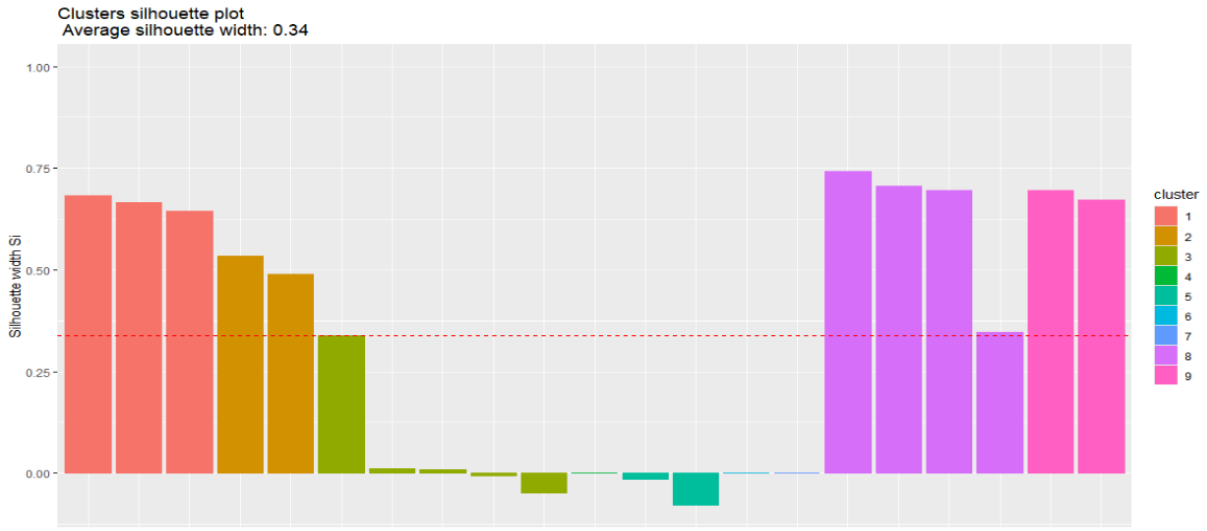


Gráfico Silueta sobre los 9 clústers aplicados a los distritos.

Los clústers que solo tienen un dato saldrán como cero, ya que al calcular distancias con otros puntos no hay, por lo que el cálculo final sale 0. Los clústers rojo, marrón, morado y rosa están muy conseguidos ya que los datos están por enigma de la línea roja. Coinciden con mis análisis, son distritos que tienen bastante similitud. Sin embargo, el clúster turquesa y el clúster verde pistacho no son clústers que tengan mucho que ver los datos entre ellos, de hecho hay un par de datos que podrían confundirse con otros clústers por lo que silhouette lo representa poniendo barras hacia abajo.

Prueba 2. Datos agrupados por incidencia

Todo el código pertinente a esta sección se encontrará en el anexo 3. En este informe solo se expondrá las conclusiones del análisis de componentes principales y las conclusiones del análisis cluster. Para consultar el análisis de los datos consultar el anexo 3.

Análisis de Componentes Principales (PCA)

Una vez más, antes de realizar el análisis cluster voy a aplicar PCA al dataframe. Es importante tener en cuenta una vez más las escalas de los datos.

	PC1	PC2	PC3	PC4	PC5
tie_medio_intervencion	-0.70929093	-0.003289782	-0.70163112	-0.06201915	-0.027621509
activaciones_totales	0.66731255	-0.311488963	-0.67589316	0.02808564	0.006944460
activaciones_canceladas	-0.02366208	0.008597091	-0.01654685	0.01222029	0.999471391
noche	-0.21878651	-0.949532058	0.22284415	0.02866690	0.006326663
hospital	0.05632855	-0.035759802	0.03080418	-0.99719291	0.014343562

Resultado del Análisis de Componentes Principales dataframe de incidencias.

Importance of components:

	PC1	PC2	PC3	PC4	PC5
Standard deviation	1.0759	1.0005	0.9208	0.21455	0.08430
Proportion of Variance	0.3784	0.3271	0.2771	0.01504	0.00232
Cumulative Proportion	0.3784	0.7055	0.9826	0.99768	1.00000

Con los 3 primeros componentes ya se explicaría un 98% de la variabilidad de los datos.

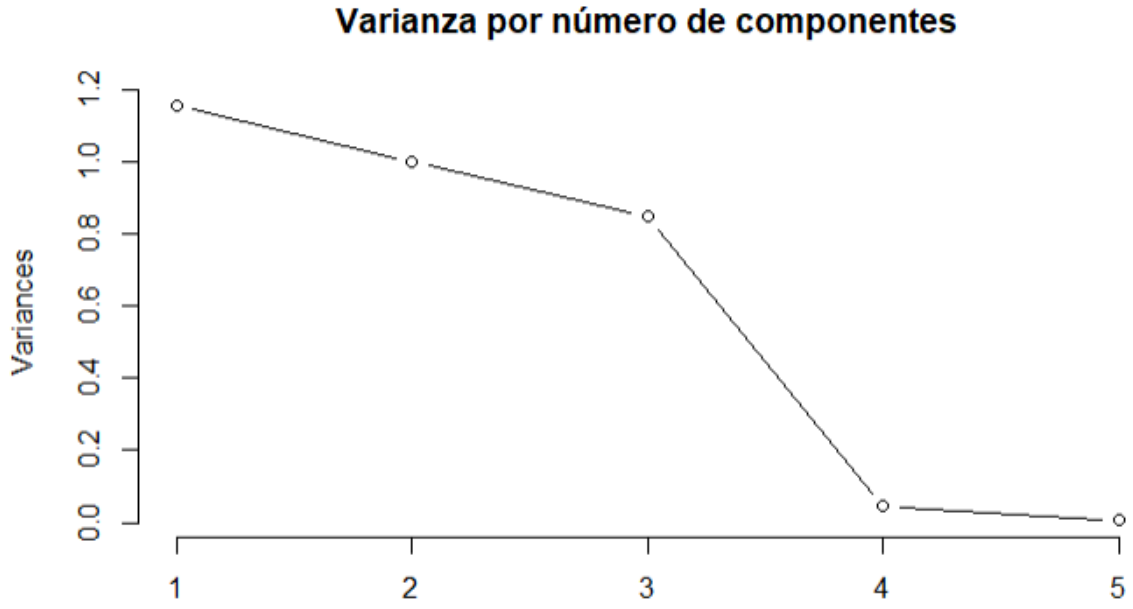
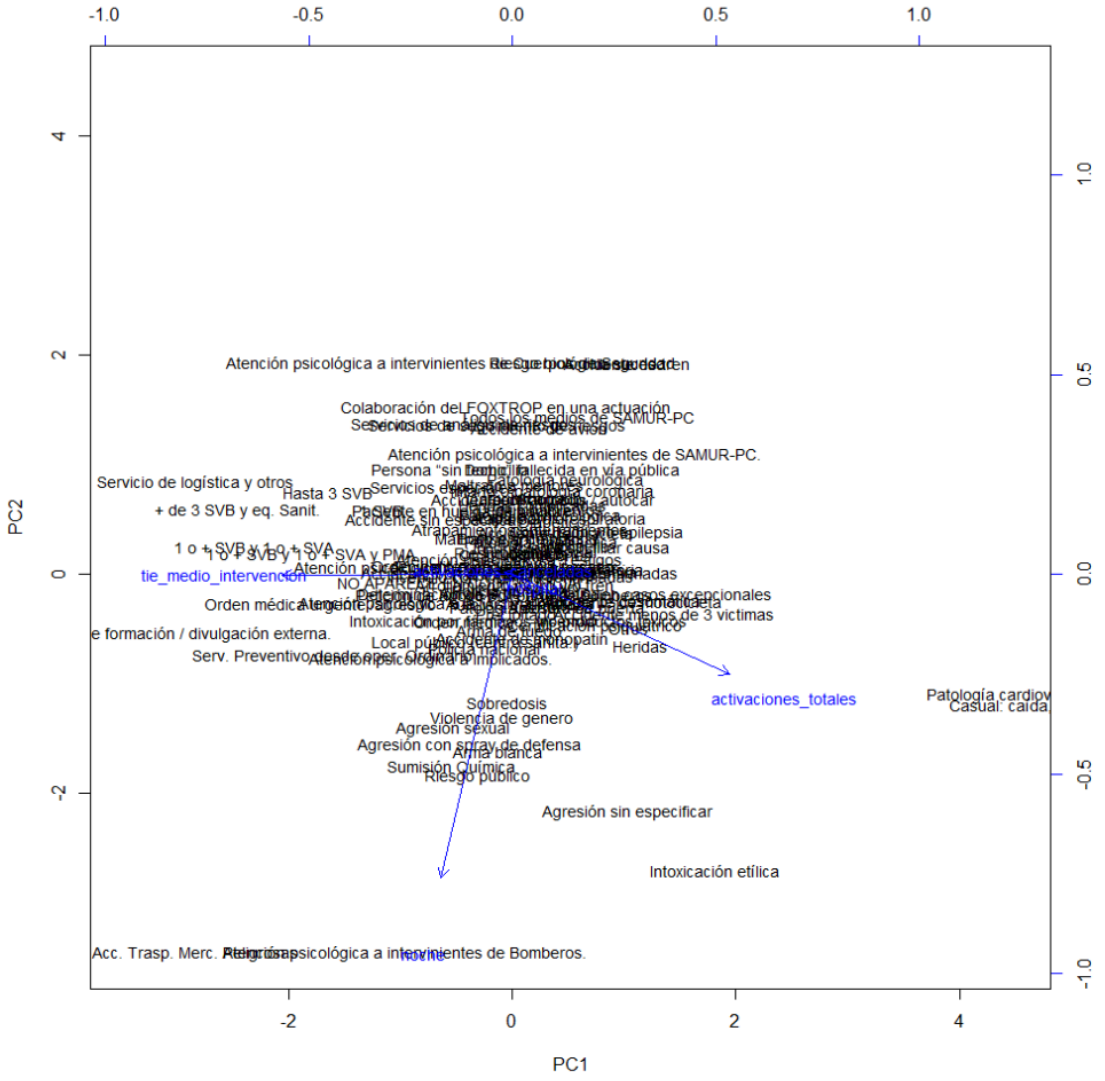


Gráfico mostrando la varianza por número de componentes. Dataframe de incidencias.



Representación del resultado del Análisis de Componentes Principales. Dataframe de incidencias.

Conclusiones del resultado de PCA

En la parte de la derecha, se encuentran las incidencias de Patología Cardíaca y Casual. Estas son las incidencias que mayor número de ocurrencias tienen y es lógico que estén en el extremo de la derecha tirando fuertemente de esa flecha.

Abajo nos encontramos con Atención Psicológica a Bomberos, Accidentes con Transporte de Mercancia Peligrosa. Estas son las incidencias que SIEMPRE han ocurrido por la noche, que también se puede deber a que solo haya ocurrido una vez y justo haya sido por la noche. Por ello están en el extremo.

Arriba del todo en cambio tenemos justo lo contrario, todos aquellos código que solo han pasado por el día, como Atención Psicológica a intervinientes.

Si seguimos la flecha que refleja la variable noche, vemos uqe hay una nube de codigos a mitad: Riesgo Público, Sumisión Química, Sobredosis, Agresión Sexual… Todos estos son los códigos que, sin ser 1, tienen el mayor ratio de ocurrencias por la noche, junto con Agresión sin especificar e Intoxicación Etílica, los cuales se encuentran tirando también de la flecha de activaciones totales.

Subiendo un poco por la izquierda vemos que hay unas pocas como Servicio de logística y otros, Formación y Divulgación u Órden Médica los cuales son los que mayor tiempo de intervención tienen, cosa que tiene sentido ya que son código que no implican una emergencia por sí solos. Ninguna ambulancia que vaya a divulgar primeros auxilios a un colegio debería ir a 180 por la Castellana, algo que es bastante comprensible.

La nube negra que hay por el centro de la imagen son códigos que están bastante en la media.

Análisis Cluster

Al igual que en el otro caso, vamos a utilizar el Gap Statistic para determinar el número óptimo de clústers.

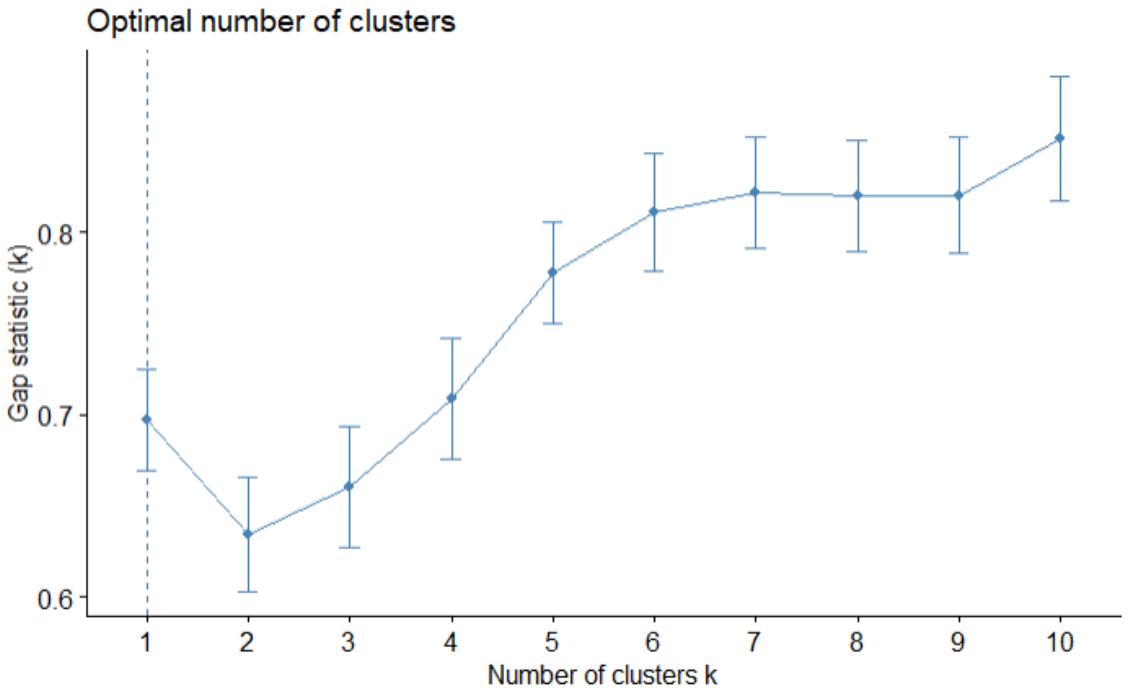


Gráfico Gap después de escalar los datos de incidencias. K óptima 1

Una vez más nos sale que el número óptimo es 1, por lo que procedemos a hacer exactamente lo mismo, en vez de escalar los datos, los normalizamos.

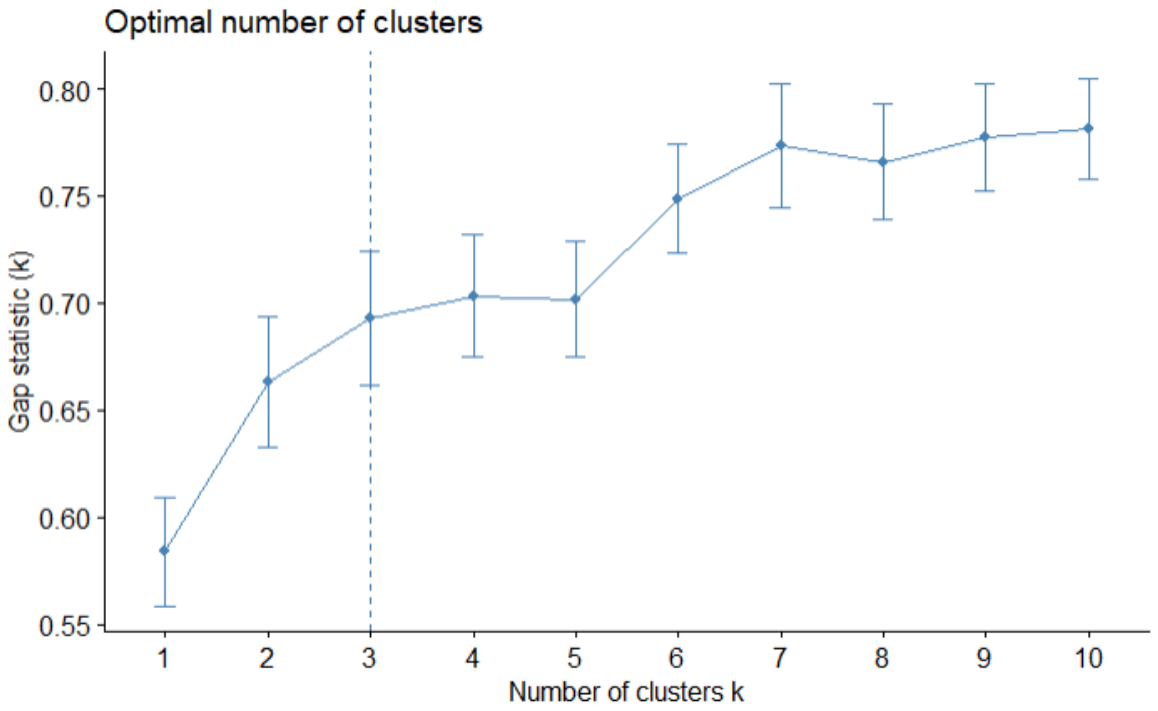


Gráfico Gap después de normalizar los datos de incidencias. K óptima 3

Esta vez sale que el número óptimo es 3, por lo que procedemos a realizar la representación de los clústers.

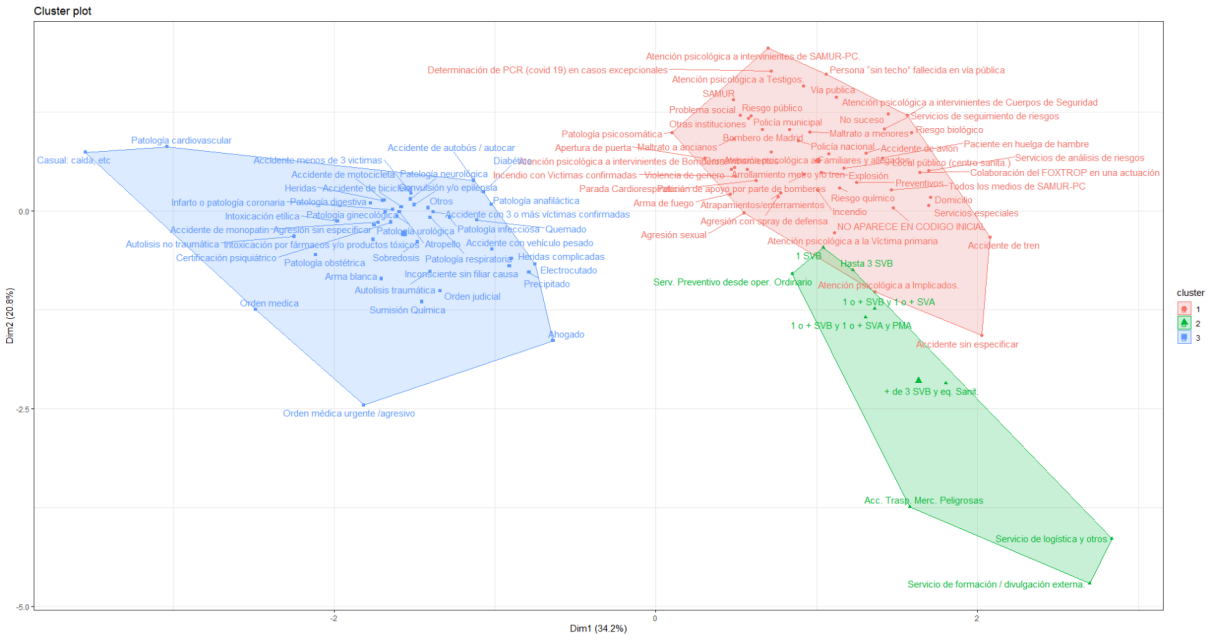


Imagen con los 9 clústers separando incidencias

Obtenemos los 3 clústers de la imagen.

- **El clúster azul** contiene los códigos que mayor ratio de hospitalizaciones y mayor número de intervenciones tienen . Pueden calificarse como los códigos que más pasan y que más probabilidad hay de que te vayas al hospital.
- **El clúster verde** sería el que contiene aquellos que más tardan en llegar.
- Por último tendríamos el **clúster rojo**, que son códigos que no destacan en sí por nada, están bastante en la media todos. Tienen métricas similares.

Para analizar cuánto de bien están hechas las agrupaciones analizamos el gráfico de silueta.

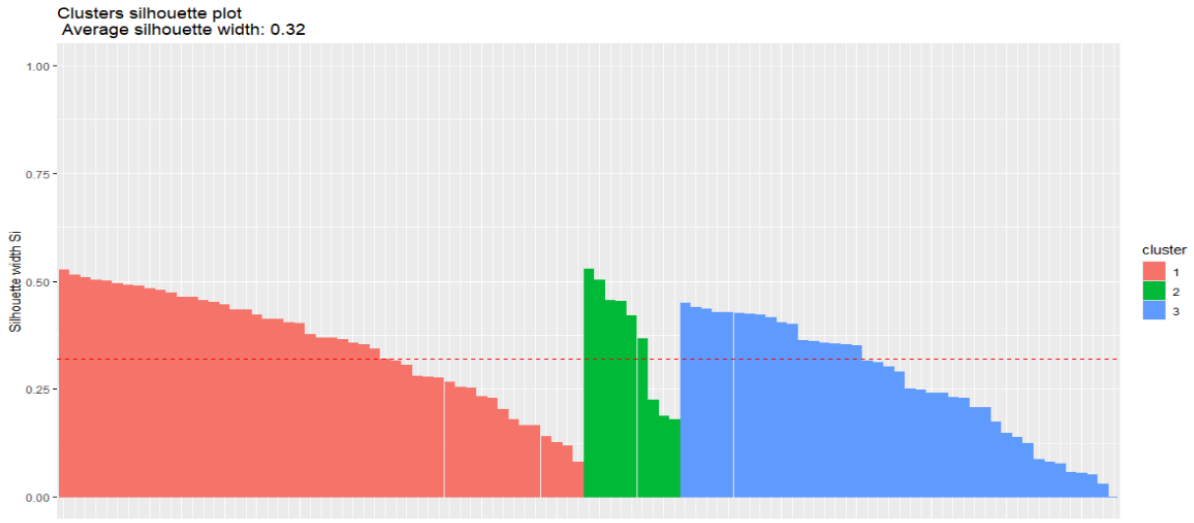


Gráfico Silueta sobre los 3 clústers aplicados a las incidencias.

En este caso podemos ver que no hay ningún valor por debajo de 0, por lo que podríamos concluir que de una forma u otra, todos los datos estarían bien agrupados. El clúster verde sería el que menos datos tiene pero que más ha acertado, ya que todos los datos de verdad destacan en su tiempo de intervención. Por otra parte el clúster rojo, vemos tiene la mayoría de datos y bastante acertados ya que se encuentran muchos por encima de la línea roja. Por último el clúster azul que tampoco está mal encaminado, aunque es verdad que hay algún dato que puede desajustar un poco la métrica.

Referencias

Datos abiertos del Ayuntamiento de Madrid (2023) Distritos del Municipio de Madrid [en línea] disponible en https://datos.comunidad.madrid/catalogo/dataset/distritos_municipio_madrid [consulta: 06-03-2023]

Datos abiertos del Ayuntamiento de Madrid (2023) Activaciones del SAMUR-Protección civil y Clasificación del incidente (código y descripción) 2022 [en línea] disponible en: https://datos.madrid.es/portal/site/egob/menuitem.c05c1f754a33a9fbe4b2e4b284f1a5a0/?vgnextoid=50d7d35982d6f510VgnVCM1000001d4a900aRCRD&vgnextchannel=374512b9ace9f310VgnVCM100000171f5a0aRCRD&vgnextfmt=default [consulta: 06-03-2024]