Unidad 4

Análisis multivariado de datos

IU Digital de Antioquia

INSTITUCIÓN UNIVERSITARIA DIGITAL DE ANTIOQUIA

ORD No 74 de 2017 · VIGILADA MinEducación

Unidad 4.
Análisis multivariado de datos

Introducción

La estadística multivariada es una rama de la estadística tradicional que se ocupa de la observación y análisis de las posibles relaciones que se dan entre variables.

Se puede definir igualmente como un conjunto de distribuciones de probabilidad multivariadas, en lo que respecta a la manera en que se representan las observaciones (datos observados) y su uso en la inferencia estadística.

Para entender los usos y ventajas de la estadística multivariada, en esta tercera unidad exploraremos diferentes definiciones, tablas y ejemplos de conceptos como, por ejemplo: el análisis de correlación y de varianza, la agrupación, la discriminación y análisis de agregación de datos a través de las técnicas de análisis clúster, entre otros temas.

Esperamos que, al finalizar los contenidos del curso, puedas aplicar todo lo que has aprendido en diferentes ámbitos de tu vida profesional y académica.

Objetivos de aprendizaje:

- Identificar los tipos de relaciones entre conjuntos de datos.
- Desarrollar algunas soluciones aplicando los conceptos de coeficientes de correlación, uso de hipótesis y valor de significancia de relaciones entre datos.
- Aplicar métodos de agrupación, discriminación y análisis de agregación de datos en la solución de problemas.

Unidad 4. Actividad de aprendizaje 1: Datos multivariados: relaciones, comportamientos y predicción

En esta actividad de aprendizaje abordaremos diferentes conceptos que nos permitirán identificar las posibles relaciones entre los datos y la forma en que estas se generan.
Por consiguiente, en esta tercera unidad, aprenderemos a trabajar los conjuntos de datos, agrupándolos, reduciéndolos y caracterizándolos. Además, aplicaremos el concepto de hipótesis nula para reconocer su utilidad en la solución de problemas.

En consecuencia, los temas que estudiaremos son los siguientes:

- Tipificación de relaciones entre conjuntos de datos.
- Covarianza, correlación, valores de significancia y su importancia en la declaración de la hipótesis nula.
- Agrupación, discriminación y análisis de agregación de datos.

Al terminar el desarrollo de esta unidad, comprenderás la forma en que se establecen las relaciones entre datos, lo cual te ayudará en la resolución de problemas mediante el uso de las hipótesis y las diversas técnicas para agrupar, desagregar y clasificar datos.

Unidad 4:
Análisis multivariado de datos

1. Tipificación de relaciones entre conjuntos de datos

Cuando hablamos de análisis multivariado, nos referimos a un grupo de técnicas que tienen por objetivo examinar un determinado conjunto de variables. Existen tres aspectos claves a tener en cuenta al momento de hacer un análisis multivariado:

- **Identificar las unidades de análisis:** reconocer cuáles son los objetos que vamos a estudiar.

- **Los datos:** valores de los atributos que representan las unidades de análisis.
- **Las variables:** conjunto ordenado de datos, compuesto por valores de un solo tipo. Una variable puede ser numérica, categórica o continua. Más adelante veremos en detalle este concepto.

La forma en que se abordan las relaciones entre los conjuntos de datos define cuál de las diferentes técnicas de análisis multivariado deberá ser utilizada: relaciones, clasificación o reducción o resumen.

En la siguiente figura se ilustran las diferentes técnicas de análisis de acuerdo al tipo de relación planteado entre estos conjuntos de datos.

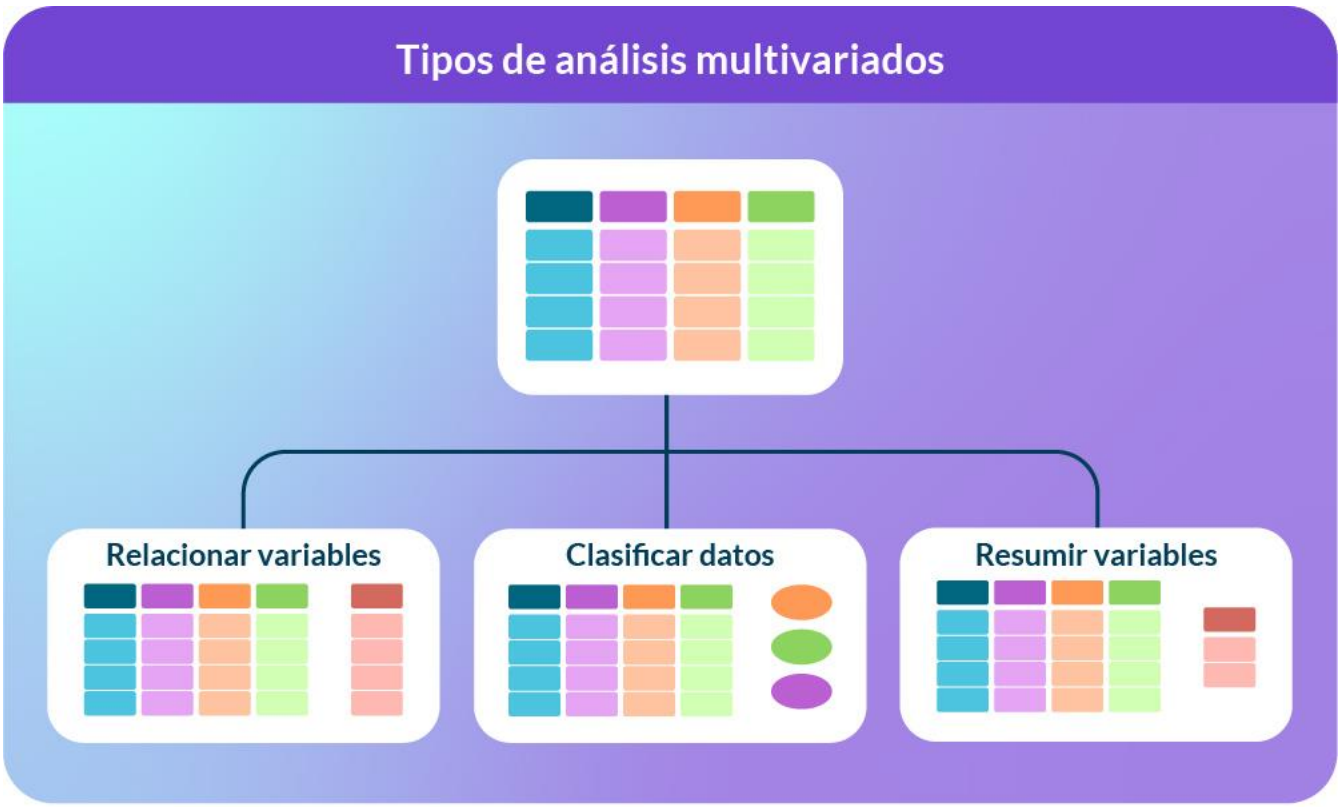


Figura 1. Tipos de análisis multivariados

- **Métodos para relacionar variables:** tienen como objetivo analizar el comportamiento de las variables independientes y la manera en que modifican el comportamiento de las variables dependientes.
- **Clasificar datos:** agrupar los datos de acuerdo a sus características.
- **Resumir variables:** crear constructos que representan el comportamiento de un conjunto de variables a partir de otro más pequeño.

Importante

El análisis multivariado tuvo un importante impulso en el siglo XX, gracias al psicólogo Charles Spearman, quien determinó que su objetivo no radicaba en las matemáticas, sino en estudiar la inteligencia de las personas. Spearman propuso que la inteligencia de una persona no podía ser medida directamente, ya que debían considerarse varios rasgos antes de evaluarla.

Tal aporte supone la esencia de la estadística multivariada, es decir, el análisis de un conjunto de variables para la comprensión de un fenómeno.

1.1. Análisis exploratorio multivariado de conjuntos de datos

Podemos clasificar los diferentes tipos de análisis de la siguiente manera:

- Exploratorios
- Confirmatorios
- Explicativos

El análisis exploratorio tiene por objetivo familiarizarse con los datos, buscar patrones y determinar que métodos pueden resultar útiles para abordar los datos de mejor manera.

Técnicas como la visualización de datos, la estadística y el agrupamiento definen una buena ruta para familiarizarse con los datos sin intentar explicar nada en forma explícita, pues no hay definiciones estrictas de análisis confirmatorio o explicativo.

El agrupamiento en particular puede ser usado para todos los tipos de análisis, pero como ya se ha indicado, se utiliza usualmente para el análisis exploratorio. Adicionalmente, es útil para confirmar creencias pasadas.

Aunque un determinado conjunto de observaciones puede pertenecer a un agrupamiento, no es algo común, ya que la clasificación es una técnica más precisa para resolverlo. En cualquiera de los casos anteriores, el objetivo es lograr una aproximación a la comprensión de los datos para definir las estrategias necesarias que permitan llevar las observaciones a estados esperados.

En relación con la visualización de datos, es posible afirmar que esta es una de las formas de uso más común en el análisis exploratorio, utilizando, por ejemplo, gráficos como los diagramas de dispersión, que permiten representar en dos (o tres) dimensiones cada una de las posibles parejas (o tríos) de variables.
Este tipo de diagrama permite identificar posibles formas de agrupación y localizar datos extremos. Para ello, es importante garantizar que las variables estén representadas en unidades semejantes (estandarización). Además, vale la pena tener en cuenta que el orden de las variables afecta la interpretación.

En el análisis multivariante, los datos multivariados se encuentran en tablas que contienen más de dos variables. Cada una de ellas se almacena en una columna, medidas en múltiples unidades estadísticas, tales como individuos, unidades de producción, localizaciones espaciales, entre otros, los cuales se almacenan en filas y se les conoce como observaciones.

Las preguntas de carácter exploratorio nos permiten abordar conjuntos de datos multivariantes, sin la necesidad de suponer la validación de alguna hipótesis.

Problema

Un banco tiene en una tabla los siguientes datos de 200 clientes: edad, saldo promedio en ahorro y si les ha sido aprobado o no un crédito. ¿Podría identificarse si existe una relación entre la edad y el saldo promedio en ahorro?

Nota: *los datos se encuentran en el siguiente archivo, debes hacer clic en el ícono para descargarlo.*

datosBanco.csv

Solución

En primer lugar, debes cargar el archivo que contiene los datos en la interfaz de *R Cloud.*

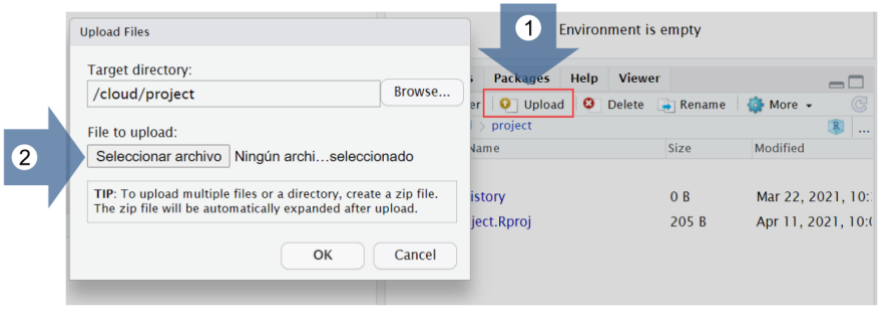


Figura 2. Carga de archivos

A continuación, importa los datos del archivo para crear un *dataset*.

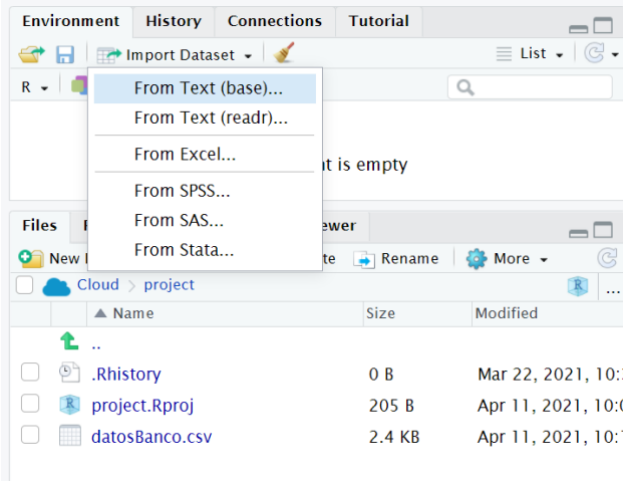


Figura 3. Importación de un *dataset*

En la ventana de diálogo, asegúrate de marcar la casilla que permite importar los datos de tipo texto *(string)* como factores.

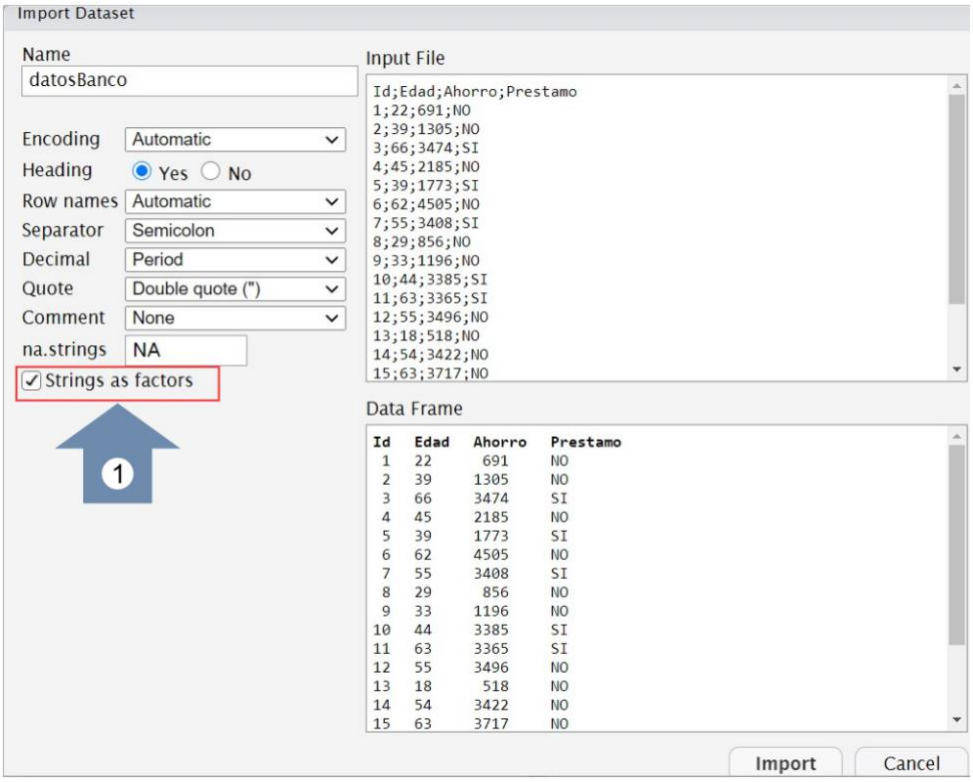


Figura 4. Parámetros de Importación de un *dataset*

En el *script*, asigna a la variable ***data*** el *dataset* de ***datosBanco***. Luego, imprime los puntos que corresponden a los valores de ***Edad*** y ***Ahorro*** para cada una de las observaciones.

```
data = datosBanco
plot(x = data$Edad, y =data$Ahorro)
```

El resultado de la gráfica del diagrama de dispersión es el siguiente:

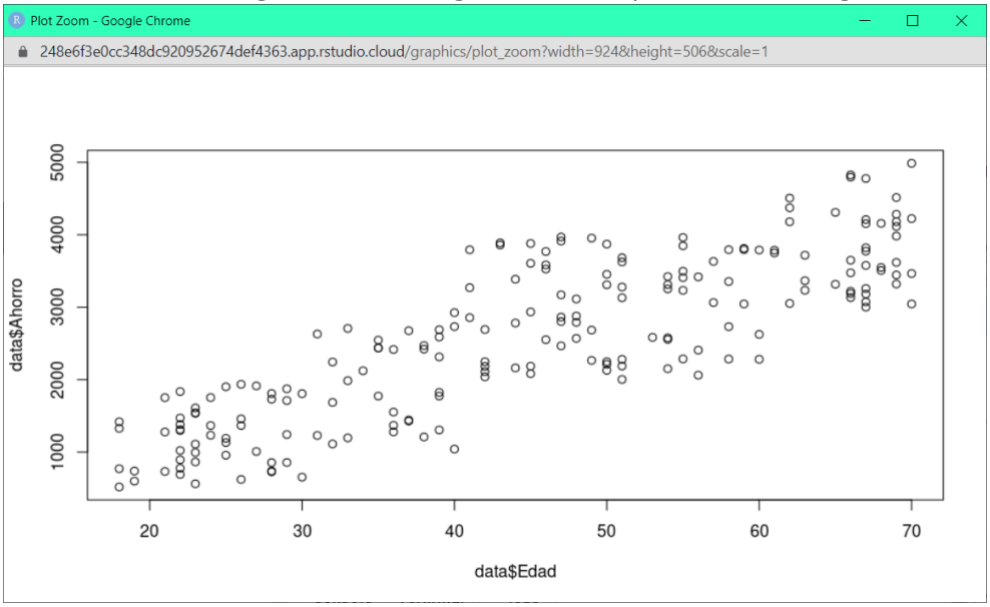


Figura 5. Ventana con el *zoom* de un gráfico

Allí podemos identificar que efectivamente hay una relación entre la edad y el valor del ahorro. A menor edad, los valores de ahorro disminuyen, y a mayor edad, aumentan.

Si quieres mostrar también en esta visualización cuáles de las observaciones corresponden a ahorradores que han accedido a un préstamo, agrega "***col***" y establece la columna que definirá la categorización en la simbología.

Por otro lado, el componente ***legend*** permite definir en qué lugar va la leyenda, cuáles son los valores que se utilizarán, los colores y el título para el cuadro.

```
data = datosBanco
plot(x = data$Edad, y = data$Ahorro,
     col = data$Prestamo)
legend(x = "topleft",
     legend = c("SI", "NO"),
     fill = c("Black", "Red"),
     title = "Préstamo")
```

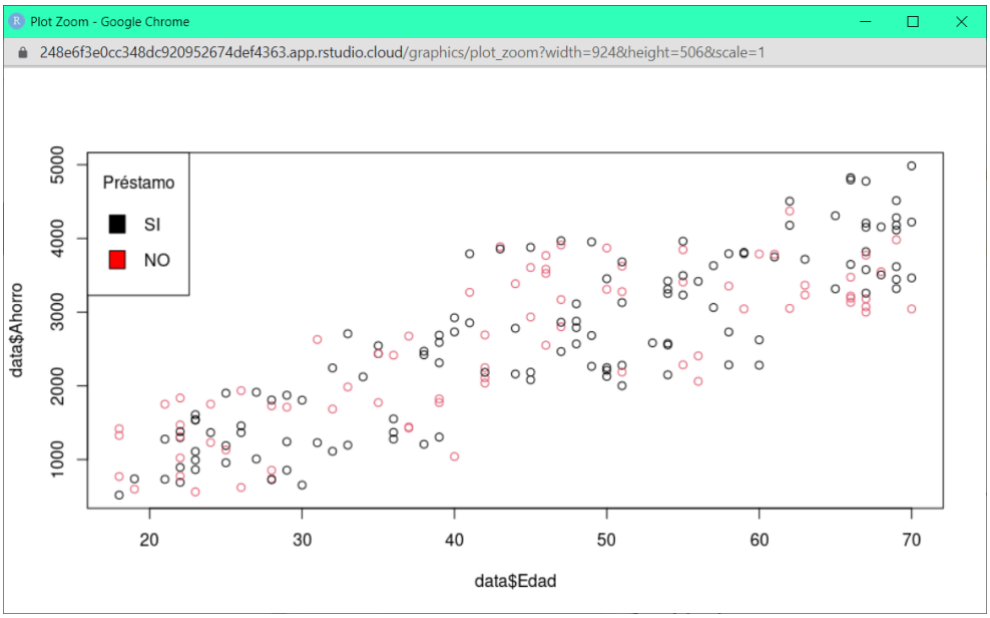


Figura 6. Ventana con el *zoom* de un gráfico con leyenda

En las siguientes líneas debes estar alerta a la posibilidad de que la leyenda de los ejes tome el mismo nombre de las variables en las que se almacenarán los valores de las columnas.

```
data = datosBanco
Edad = data$Edad
Ahorro_en_Miles = data$Ahorro
plot(x = Edad, y = Ahorro_en_Miles,
     col = data$Prestamo)
legend(x = "topleft",
     legend = c("SI", "NO"),
     fill = c("Black", "Red"),
     title = "Préstamo")
```

El paso anterior debe generar el gráfico que se ve a continuación:

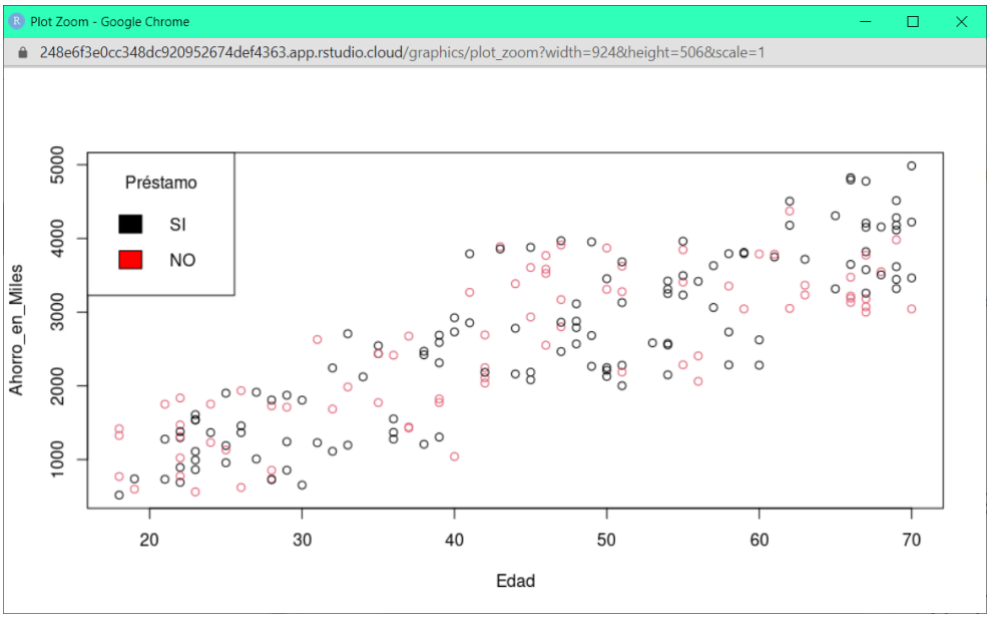


Figura 7. Ventana con el *zoom* de un gráfico personalizando los nombres de los ejes

1.2. Coeficiente de correlación

Para estudiar la relación lineal que existe entre dos variables continuas, se requieren ciertos parámetros que hacen posible cuantificarla. Uno de ellos es la covarianza, la cual indica el grado de variación o variabilidad conjunta de dos variables aleatorias.

La covarianza mide la cantidad de relación lineal que existe entre las variables y su sentido, ofreciendo resultados como los siguientes:

- **Relación lineal positiva:** si una variable crece, la otra también.
- **Relación lineal negativa:** si una variable crece, la otra decrece.
- **No hay relación lineal entre las variables:** el comportamiento de una variable no depende de la otra.

Esto puede concluirse a partir de una simple observación de la nube de puntos del diagrama de dispersión, como el ilustrado en la figura 8.

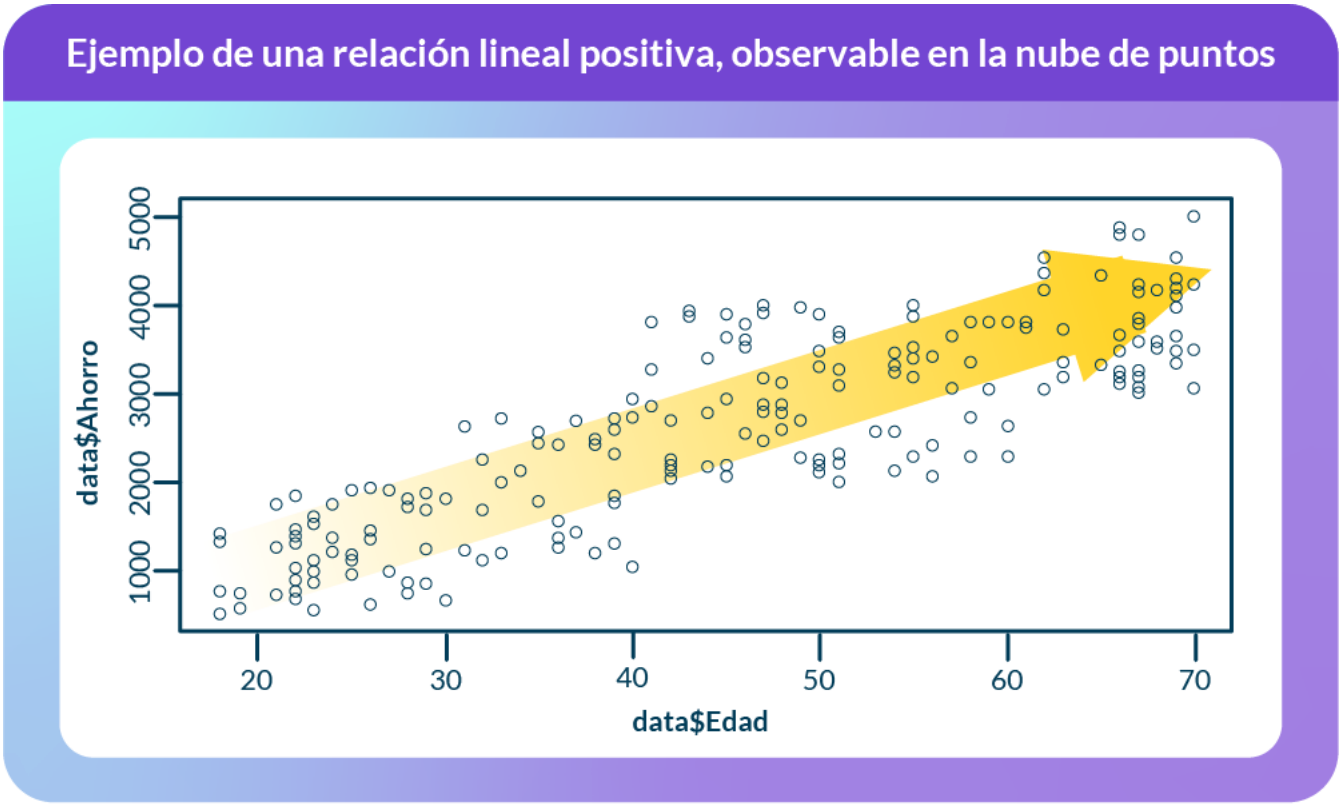


Figura 8. Ejemplo de una relación lineal positiva, observable en la nube de puntos

El análisis de correlación, por otro lado, permite comprobar la relación entre las variables. En este sentido, el coeficiente de correlación mide la fuerza y la dirección de la relación lineal que hay entre las variables, de las cuales ya se ha identificado su correlación. Los resultados que se obtienen para un coeficiente de correlación oscilan entre -1 y +1.
La representación abreviada del coeficiente de correlación es una (***r***) minúscula.

Para las variables con las que se cuenta, es necesario considerar las escalas empleadas para su medición, ya que no es posible hacer comparaciones entre pares de variables distintos. Por esta razón, para generar los coeficientes de correlación, se debe estandarizar la covarianza.

Los coeficientes de correlación más comunes son: *Pearson* y *Spearman*.

Recuerda que el valor del coeficiente varía entre -1 y +1, y miden la fuerza de asociación o tamaño del efecto que se interpretan, un ejemplo de ello se puede observar en la siguiente figura:

Figura 9. Niveles de la fuerza de asociación según valor del coeficiente de correlación

Veamos para qué sirven dichos coeficientes:

- **Correlación de *Pearson*:** se recomienda para casos donde se tienen datos cuantitativos que presentan una distribución normal y presentan una mayor sensibilidad a los valores extremos.
- **Correlación de *Spearman*:** es útil cuando se tienen datos en intervalos, que son ordinales, o cuando son datos continuos que no presentan una distribución normal y se pueden transformar a rangos. Este método no es paramétrico.

Adicionalmente, se calcula el *p-value,* o valor de (***p***), que es un valor que oscila entre 0 y 1, el cual indica la probabilidad de que un valor observado sea igual o más extremo que cierto valor y declara si se rechaza la hipótesis nula (H_0).

En este sentido, se afirma que, si el valor de ***p*** tiene un nivel de significancia menor que 0,05, rechazamos la hipótesis nula de que no hay diferencia entre las medias. Y concluimos que sí existe una diferencia significativa. Por otro lado, si el valor ***p*** es mayor que 0,05, no podemos concluir que existe una diferencia significativa y, por tanto, no se rechaza H_0.

Hipótesis nula

Una hipótesis se refiere a cierto tipo de afirmación sobre un parámetro estadístico que se presenta en la población, tales como la media o la desviación típica. La forma de representarla es con H_0. La hipótesis nula se construye planteándola de forma contraria a lo que se quiere probar, hasta que el análisis de los datos demuestre que el punto de partida era falso o absurdo, lo que deriva en su rechazo y, por tanto, comprueba lo que se quería contrastar.
Por otra parte, hablamos de hipótesis alternativa (H_1) cuando los resultados del análisis rechazan la hipótesis nula y se encuentra algún tipo de relación entre los datos.

Adicional al valor del coeficiente de correlación, y de su significancia, también es posible calcular el tamaño del efecto asociado, conocido como coeficiente de determinación R_2. El cual se puede interpretar como la varianza de ***Y*** explicada por ***X***. La forma de calcularlo es elevando al cuadrado el coeficiente de correlación en los casos indicados del coeficiente de *Pearson* y de *Spearman*.

Problema

Considerando el mismo conjunto de datos del problema anterior, asume la siguiente hipótesis nula:

H_0 = No existe una relación lineal entre la edad y el saldo promedio de los ahorros de los clientes del banco.

¿Es posible rechazar o no esta hipótesis nula?, ¿se debe considerar como hipótesis alternativa?, ¿cuál sería esa hipótesis alternativa?

datosBanco.csv

Solución

Asumiendo que ya has cargado los datos e importado el *dataset*, tal como se indicó en la solución al problema anterior, utiliza las siguientes líneas de código para almacenar el *dataset* en una variable llamada ***data***.

La segunda línea, de la cual nos ocuparemos en esta parte del contenido, se encarga de calcular la covarianza entre las columnas elegidas.

```
data = datosBanco
cov(data$Edad,data$Ahorro)
```

```
[1] 14590.85
```

Observa que la covarianza en el conjunto de datos es positiva, es decir, se identifica una relación lineal ascendente, ya que las dos variables crecen en el mismo sentido, lo que se evidencia en el diagrama de dispersión. También se identifica que es positiva en el resultado del cálculo de la covarianza.

Sin embargo, el número obtenido del cálculo es difícil de interpretar por su dependencia de las unidades en las que se representan las variables, por esta razón, en su lugar, se debe trabajar con la covarianza y el coeficiente de correlación lineal.

En *R,* utilizando la función **cor()** calculamos el coeficiente de correlación con el método de *Pearson* y el de *Spearman*, tal como se indica a continuación.

Nota: en gris aparece el código y en negro el resultado en la consola de *R*.

```
cor(data$Edad,data$Ahorro,method = "pearson")
[1] 0.8408979

cor(data$Edad,data$Ahorro,method = "spearman")
[1] 0.8280579
```

De acuerdo con el resultado, y considerando el concepto de la fuerza de asociación ilustrado en la figura 8, se puede observar que las variables presentan una fuerte relación lineal positiva.

Puedes usar también la función **cor.test()**, la cual sirve para encontrar el nivel de significancia, y adicionalmente el valor de p *(p-value)*.

La sintaxis de la función es la siguiente:

```
cor.test (x, y, alternative = , method = )
```

En donde:

- ***x*:** corresponde a la variable continua explicativa.
- ***y***: corresponde a la variable continua respuesta.
- ***alternative*:** es el parámetro de referencia a las colas (en la gráfica de distribución): *less* (a la izquierda - menor valor), *greater* (a la derecha – mayor valor), o *two.side* (a la izquierda y derecha en la gráfica)
- ***method***: define el tipo de correlación que se usará para el cálculo: *pearson, spearman, kendall.*

¿Para qué son las colas?
Las colas que elegimos en el parámetro ***alternative*** en la función **cor.test()** indican a la prueba si queremos saber si la correlación es menor a cero (“*less*”), mayor a cero (“*greater*”) y diferente de cero (“*two.side*”).

El código a continuación permite hacer el cálculo usando ambas colas con el método de *Pearson*.

```
cor.test(x = data$Edad,
         y = data$Ahorro,
         alternative = "two.sided",
         method      = "pearson")
```

```
   Pearson's product-moment correlation

data:  data$Edad and data$Ahorro
t = 21.864, df = 198, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.7948949 0.8772883
sample estimates:
      cor
0.8408979
```

Observemos que en este resultado el coeficiente de correlación es muy cercano a 1, además, presenta un *p-value* inferior a 0.05 (5%), lo que nos permite rechazar la hipótesis nula por su nivel de significancia.

A continuación, se puede observar el código y el resultado de la función utilizando el método de *Spearman*.

```
cor.test(x = data$Edad,
      y = data$Ahorro,
      alternative = "two.sided",
      method     = "spearman", exact = FALSE)
```

```
   Spearman's rank correlation rho

data:  data$Edad and data$Ahorro
S = 229250, p-value < 2.2e-16
alternative hypothesis: true rho is not equal to 0
sample estimates:
      rho
0.8280579
```

2. Métodos de análisis exploratorio de datos

Existen diversos métodos y enfoques para hacer el análisis exploratorio de datos multivariados. Algunas clasificaciones sugieren dos grandes grupos:

- ***El análisis factorial lineal:*** el cual incluye análisis de componentes principales, análisis de correspondencia (binaria y múltiple) y análisis discriminante.

- ***El análisis no lineal de los datos basados en núcleos:*** incluye el análisis de componentes principales del núcleo y análisis discriminante del núcleo. Existe un segundo subgrupo en el análisis no lineal basado en redes neuronales.

A continuación, desarrollaremos algunos de los métodos más usados. Es preciso anotar que estos métodos, a menudo, conllevan a procedimientos de reducción de dimensionalidad de los datos, más aún cuando hablamos de grandes volúmenes, lo que hace esta exploración más eficiente.

Los métodos que desarrollaremos serán: de agrupación, discriminación y agregación de datos.

2.1. Agrupación

Antes de iniciar, es preciso aclarar la diferencia entre clasificación y agrupación. En pocas palabras, la clasificación hace una predicción de una categoría de salida a partir de unos datos de entrada, para ello, se prepara un modelo con los datos de entrenamiento, el cual se usa luego para predecir. Por otra parte, el agrupamiento permite agrupar observaciones basadas en las semejanzas entre ellas y las diferencias respecto a otras.

Para explicar el agrupamiento, utilizaremos el método *K-medias*, que es el más popular. Ahora, esta técnica de agrupamiento implica dos prerrequisitos matemáticos: el cálculo de la distancia entre dos puntos y la definición del término *centroide*.

El primero trata de calcular la distancia euclidiana entre dos puntos a partir del teorema de Pitágoras. Este concepto es importante ya que permite obtener la distancia entre los agrupamientos.

El concepto de *centroide* se refiere a la posición media de un conjunto de puntos, lo que en física es conocido como el centro de masa. Para dos puntos, el centroide es el punto medio de la distancia que los separa, y para una nube de puntos, el centroide seguramente estará en algún lugar dentro de la nube.

Cuando utilizamos un método como el de *K-Medias*, se debe establecer inicialmente el número de grupos (***k***) que se intenta identificar. Después, es necesario especificar la semilla (una por cada agrupamiento), que se refiere al centroide de inicio, lo cual se puede establecer aleatoriamente o por un experto que conozca el comportamiento del conjunto de datos.

El método genera un conjunto de iteraciones que asigna cada uno de los puntos de la nube a una semilla con base en la proximidad. Luego, recalcula el centroide de cada grupo. Como resultado, se recalcula el centroide para cada una de las semillas y se vuelve a asignar cada uno de los puntos de la nube a la semilla más cercana. Lo anterior puede derivar en que un punto asignado a una semilla sea reasignado a otra, en otro grupo, como resultado de la reubicación de los centroides. Este proceso se realiza hasta que no se generen más cambios en la reagrupación.

Una forma resumida de presentar el flujo entre los pasos se ilustra en la siguiente figura:

Figura 10. Flujo de iteración del algoritmo de *K-medias*

Problema

Considera el siguiente archivo plano en formato *CSV* con las coordenadas de los centroides de los polígonos de los barrios de Medellín.

¿Es posible aplicar el método de *K-Medias* para crear grupos con base en la distribución espacial de estos centroides?

En caso de ser posible, ¿cuál sería el resultado para 2, 5 o 10 grupos?

Barrios_Medellin.csv

Con el propósito de incorporar una nueva herramienta en el desarrollo de este tipo de soluciones, utilizaremos *Python,* aprovechando la IDE de *Google Colaboratory,* o *Google Colab,* como se conoce comúnmente, a la cual se puede acceder desde el siguiente enlace: https://colab.research.google.com/

A diferencia de la IDE de *R Cloud*, los archivos necesarios no se alojan en el proyecto que se crea, sino que deben almacenarse en el *Drive* asociado a la cuenta de Google con que se trabaja.

El procedimiento es el siguiente:
Importa el método *drive* del submódulo *Colab* de Google, tal y como se muestra en la línea 1 del *script* de la siguiente figura:

Figura 11. Código para importar el método *drive* de *google.colab* y montar la unidad de almacenamiento

Debes crear y ejecutar las dos líneas del *script* que se muestra y luego hacer clic sobre el enlace que aparece. Automáticamente, aparece una ventana en la que debes elegir la cuenta de *Drive* en la cual está la carpeta que quieres montar.

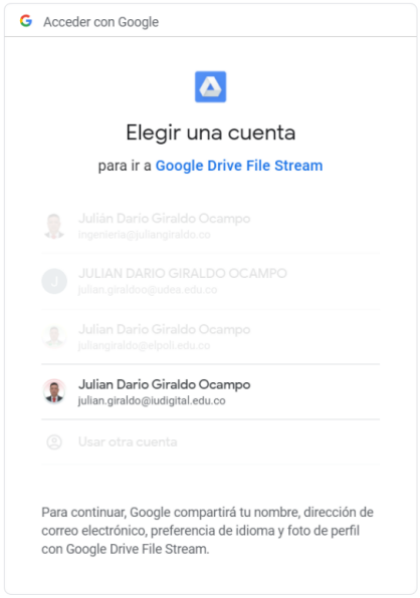


Figura 12. Ventana para elección de la cuenta de Google para elegir el drive que se asocia al *script*

En seguida, aparece otra ventana en la que debes hacer clic sobre ***Permitir***.

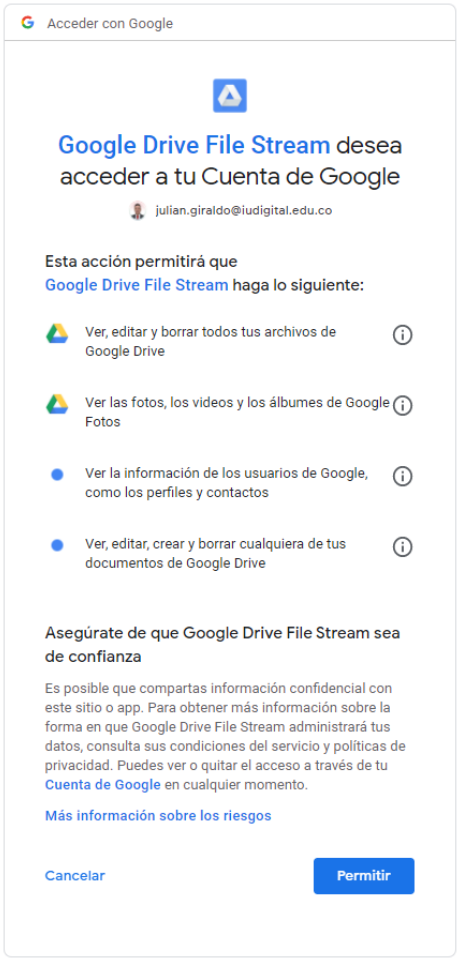


Figura 13. Ventana para autorización de acceso a *Google Drive File Stream*

Luego, copia el código que aparece en la ventana que aparecerá a continuación.

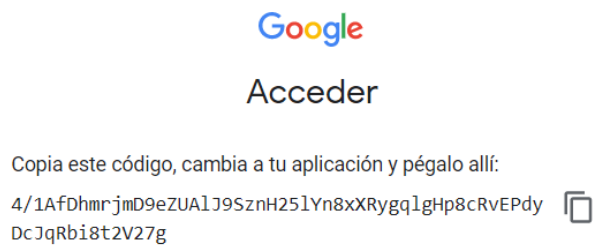


Figura 14. Ventana con el código de autorización

Pega el código en el campo que dice: "Enter your autohorization code:". Y luego presiona ***Enter.***

Figura 15. Interfaz de *Colab* con el *script* para ingresar el código de autorización

Si tienes éxito, deberá aparecer el mensaje:
`Mounted at /content/drive`

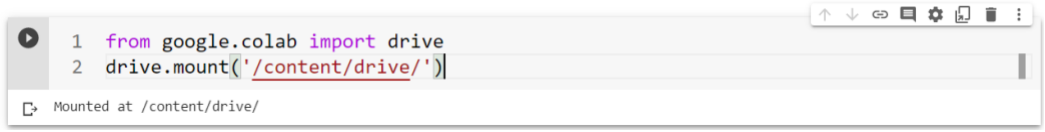


Figura 16. Confirmación del montaje de la unidad de *Drive*

Ya con la unidad montada, haz clic en el ícono de la carpeta ubicado en la barra izquierda (1).

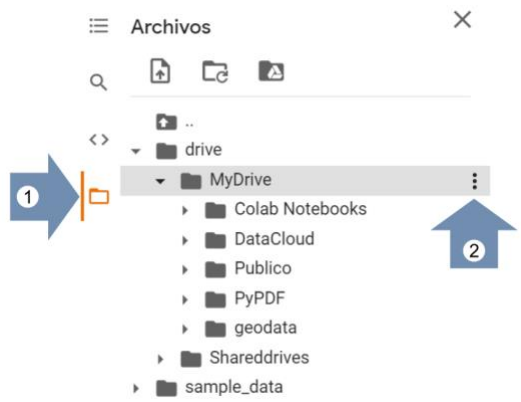


Figura 17. Acceso a las carpetas y archivos del *Drive* de la cuenta asociada

Si quieres subir algún archivo, selecciona la carpeta en la cual se deba almacenar, haz clic en los tres puntos a la derecha del nombre, y elige la opción ***Subir***.

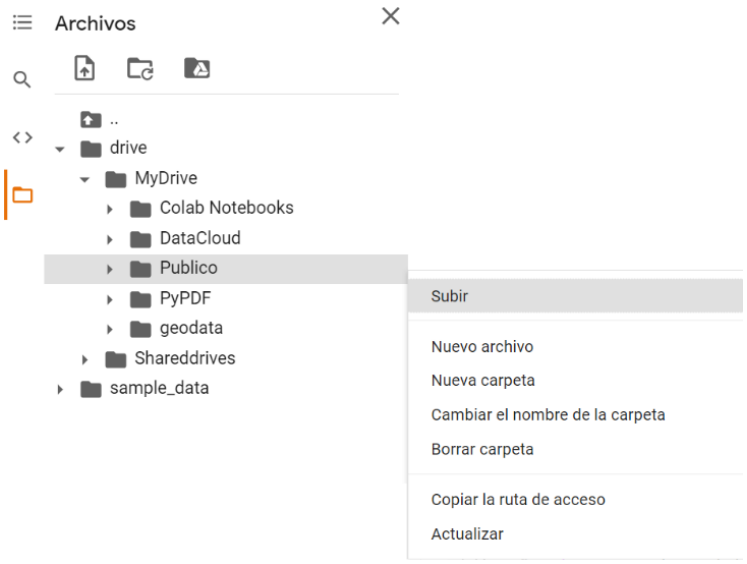


Figura 18. Opción de carga de archivos al *Drive* desde la interfaz de *Colab*

Para terminar, basta con copiar la ruta de acceso al archivo para acceder al mismo en el código, y cargarlo como variable o *dataset* en el *script*.

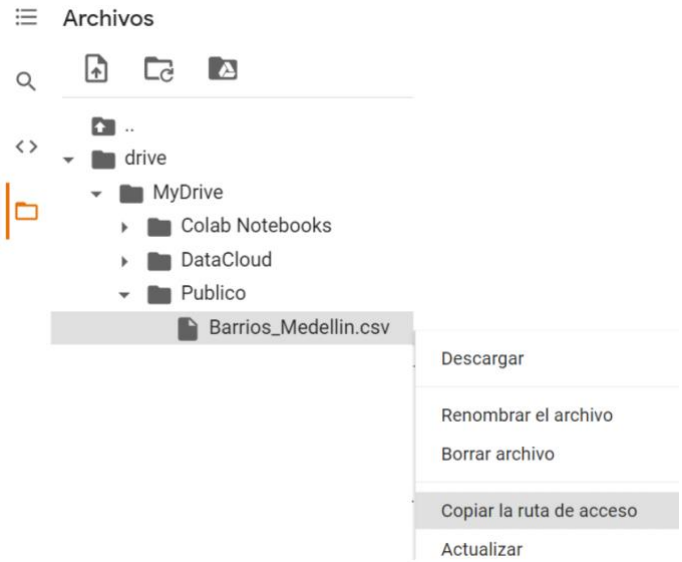


Figura 19. Captura de la URL del archivo para cargarlo en el *script*

Solución

En primer lugar, monta la unidad de *drive* en la que está almacenado el archivo ***.csv.***

Se recomienda que los bloques de instrucciones propuest
desarrollo de este ejercicio, se ubiquen en celdas indepen
esta manera, es posible ejecutar bloques de código especí
que correr el código completo.

Para ejecutar el bloque o celda de código, haz clic sobre el
flecha:

Esto permite hacer seguimiento al control y depuración de
(*debug*) que puedan presentarse en cada celda.

```
from google.colab import drive
drive.mount('/content/drive')
```

A continuación, debes cargar las librerías que se necesitan.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
from sklearn.cluster import KMeans
```

- ***Pandas:*** es una librería de *Python* ampliamente utilizad datos escrita como extensión de *NumPy*, ideal para la g manipulación y análisis de datos, especialmente de estr numéricas y series temporales.

  Se importa como "***pd***" para hacer referencia a la librerí escribir el nombre completo. El uso de estos alias será e resto de las librerías importadas.

- ***Numpy:*** permite crear y trabajar con vectores y matrice multidimensionales de gran tamaño, además de un con funciones matemáticas de alto nivel.

- ***Matplotlib:*** se utiliza para crear gráficos a partir de con almacenados en listas o arreglos (*arrays*).

- ***Seaborn:*** es una librería que trabaja sobre la visualizaci *MatplotLib* y se utiliza para mejorar la estética en la pre los gráficos.

- ***Scikit-Learn o sklearn***: es una librería especializada en a automático *(Machine Learning)* que cuenta con algoritr clasificación, regresión, *clustering* y reducción de dimer Para este caso, utilizaremos solo el componente de agr *clusterización.*

Seguidamente, se cargan los datos, utilizando la ruta que hemos copiado del archivo ***.csv***, tal como se muestra en la figura 17.

La función `pd.read_csv` crea un *dataframe* con la estructura de los datos del archivo, y los almacenará en la variable que hemos llamado ***data***.

```
1  data = pd.read_csv('/content/drive/MyDrive/Publico/Barrios_Medellin.csv')
2  data
```

Como se observa en la línea 2 de este código, para obtener una previsualización del *dataframe*, basta con escribir el nombre de este.

El resultado de la ejecución del código anterior es el siguiente:

	BARRIO	Lat	Long	Nombre_Comuna	Numero_Comuna
0	EL PESEBRE	6.27526	-75.6030	SAN JAVIER	13
1	BLANQUIZAL	6.27674	-75.6090	SAN JAVIER	13
2	LA GABRIELA	6.26982	-75.6253	SAN JAVIER	13
3	JUAN XXIII - LA QUIEBRA	6.26905	-75.6191	SAN JAVIER	13
4	METROPOLITANO	6.26936	-75.6139	SAN JAVIER	13
...	...	...	...	...	...
266	CAMPO ALEGRE	6.25598	-75.6166	LA AMERICA	12
267	SANTA MÓNICA	6.25072	-75.6167	LA AMERICA	12
268	SIMÓN BOLIVAR	6.24854	-75.6087	LA AMERICA	12
269	BARRIO CRISTOBAL	6.25045	-75.6125	LA AMERICA	12
270	SANTA TERESITA	6.24676	-75.6152	LA AMERICA	12

271 rows × 5 columns

Figura 20. Resultado del despliegue del *dataframe*

En este resultado se previsualizan las primeras y últimas 5 observaciones (registros o filas) del *dataframe*, además se muestra el tamaño de las filas y columnas.

Después de verificar con esta visualización que los datos importados al *dataframe* son los esperados, genera un gráfico de dispersión utilizando la función ***scatter()*** de la librería *pd* (recuerda que es el alias de ***Pandas***).

Para la generación del gráfico de dispersión, se requieren los valores que corresponden al eje ***X*** y al eje ***Y***. En este caso, utiliza la columna que almacena la longitud en el eje ***x***, que es el componente horizontal. La latitud se podrá ubicar en el componente vertical de las coordenadas.

Para hacer referencia a la columna '***Long'***, se escribe el nombre del *dataframe* y el de la columna usando comillas entre corchetes: `data['Long']`.
Con esta misma sintaxis se hace referencia a cualquier otra columna.

Luego de calcular el gráfico, visualízalo usando la función ***show()***. A continuación, en el costado izquierdo verás el código y el resultado de su ejecución.

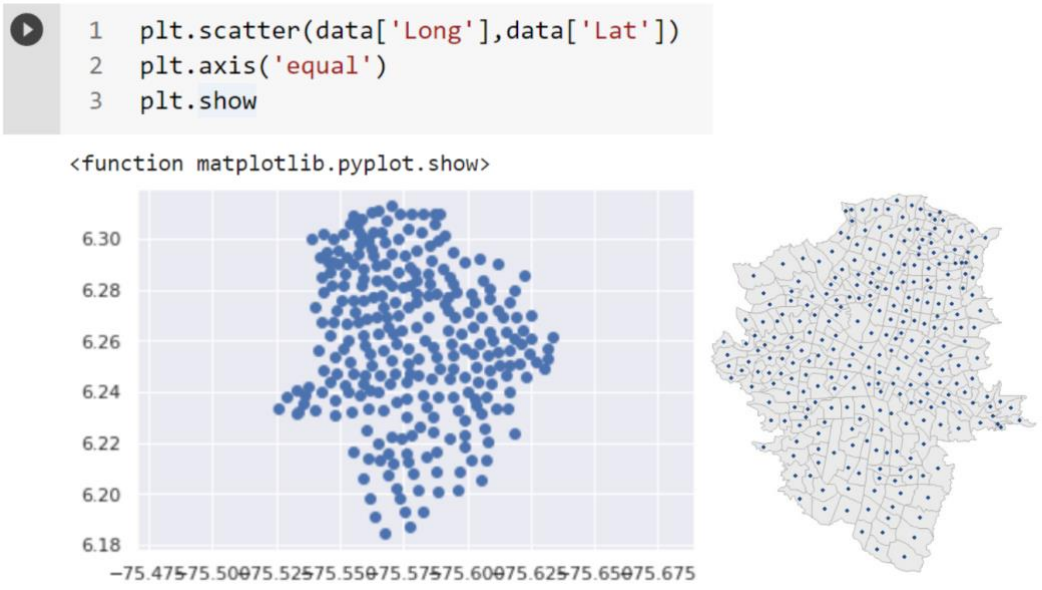


Figura 21. Comparación del resultado de la distribución de los puntos según coordenadas representadas en un modelo *scatter vs* una imagen GIS

En el costado derecho de la figura (imagen superpuesta que no es parte del resultado de *Colab*) se muestra la representación y distribución espacial de los puntos de los barrios de Medellín, utilizando *software* SIG (Sistemas de Información Geográfica). Si prestas atención, podrás darte cuenta que la distribución es la misma.

Selecciona todas las filas del *dataframe* y únicamente las columnas en las posiciones 1 y 3 (recuerda que la primera columna está en la posición 0 (cero). Lo anterior, a través del método ***iloc()*** de *Pandas,* cuyos parámetros son: `[<filas> : <columnas>]`.

Como puedes observar, antes de los dos puntos no se especifica ningún valor para indicar que se seleccionan todas las filas. Luego de los dos puntos `1:3,` se indica que son las columnas desde la 1 hasta la 3 (no incluye la columna de la posición 3).

En la siguiente figura se puede ver el código y el resultado de la ejecución.

```
1  x = data.iloc[:,1:3]
2  x
```

	Lat	Long
0	6.27526	-75.6030
1	6.27674	-75.6090
2	6.26982	-75.6253
3	6.26905	-75.6191
4	6.26936	-75.6139
...	...	...
266	6.25598	-75.6166
267	6.25072	-75.6167
268	6.24854	-75.6087
269	6.25045	-75.6125
270	6.24676	-75.6152

271 rows × 2 columns

Sobre este nuevo *dataset*, realizarás la *clusterización* definiendo el número de agrupaciones por medio del método ***KMeans()*** de la librería *sklearn*. Y luego, para hacer el agrupamiento, utiliza el método ***kmeans.fit().*** El resultado de la ejecución del código muestra los parámetros con que este agrupamiento fue hecho.

```
kmeans = KMeans(2)
kmeans.fit(x)
```

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=2, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=None, tol=0.0001, verbose=0)
```

Se pueden obtener los agrupamientos predichos para cada observación usando el método ***fit_predict()***, los cuales se almacenan en la variable ***gruposIdentificados***.

Después, imprime el resultado de la variable, que es una matriz que contiene los agrupamientos previstos. Así como puede verse en la siguiente figura, es claramente identificable que los valores corresponden a 0 y 1, dependiendo del grupo al que ha quedado asociado cada punto.

```
gruposIdentificados = kmeans.fit_predict(x)
gruposIdentificados
```

```
array([1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0], dtype=int32)
```

Para visualizar cada punto del *dataframe* original y a cuál agrupamiento ha quedado asociado, copia el *dataset* original a uno nuevo, y adiciona a este la columna con los resultados de la agrupación.

Luego, visualiza el resultado del nuevo *dataframe* con la columna que se adicionó.

```
1  datosAgrupados = data.copy()
2  datosAgrupados['Cluster'] = gruposIdentificados
3  datosAgrupados
```

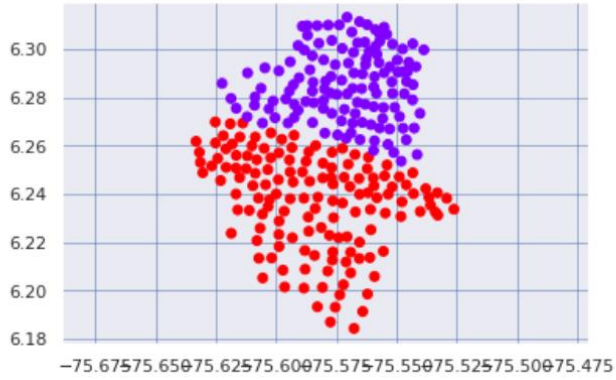
	BARRIO	Lat	Long	Nombre_Comuna	Numero_Comuna	Clust
0	EL PESEBRE	6.27526	-75.6030	SAN JAVIER	13	
1	BLANQUIZAL	6.27674	-75.6090	SAN JAVIER	13	
2	LA GABRIELA	6.26982	-75.6253	SAN JAVIER	13	
3	JUAN XXIII - LA QUIEBRA	6.26905	-75.6191	SAN JAVIER	13	
4	METROPOLITANO	6.26936	-75.6139	SAN JAVIER	13	
...	...	...	...	...	...	
266	CAMPO ALEGRE	6.25598	-75.6166	LA AMERICA	12	
267	SANTA MÓNICA	6.25072	-75.6167	LA AMERICA	12	
268	SIMÓN BOLIVAR	6.24854	-75.6087	LA AMERICA	12	
269	BARRIO CRISTOBAL	6.25045	-75.6125	LA AMERICA	12	
270	SANTA TERESITA	6.24676	-75.6152	LA AMERICA	12	

271 rows × 6 columns

Como ya has logrado hacer la agrupación, observa los resultados no solo de manera tabular, sino también gráfica. Para esto, debes generar nuevamente el gráfico de dispersión, y agrega como parámetros aquellos que permitan indicar cuál de las columnas se utilizará para definir la simbología de representación. Así como el color (o los colores) para su representación.

A continuación, se ilustra el código utilizado y el resultado de su ejecución.

```
plt.scatter(data['Long'],
            data['Lat'],
            c=datosAgrupados['Cluster'],
            cmap='rainbow')
plt.axis('equal')
plt.grid(axis = 'both',color = 'b',alpha = 0.5)
```



Repite la ejecución de todas las celdas de código desde la que define el número de agrupamientos, con un valor de 5, como se ve a continuación:

```
kmeans = KMeans(5)
kmeans.fit(x)
```

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=5, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=None, tol=0.0001, verbose=0)
```

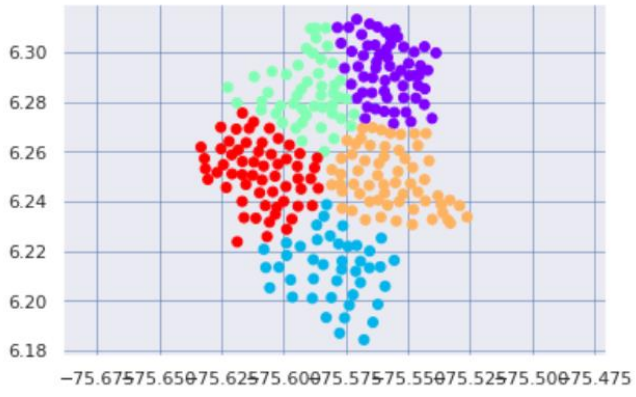
La matriz de predicción será la siguiente:

```
1  gruposIdentificados = kmeans.fit_predict(x)
2  gruposIdentificados
```

```
array([3, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 3,
       3, 3, 3, 3, 3, 3, 3, 4, 3, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 3, 3,
       3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 0,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2, 0, 2, 0, 0, 0, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 4, 4,
       4, 4, 4, 4, 4, 4, 4, 4, 1, 1, 1, 1, 4, 4, 4, 4, 4, 4, 4, 4, 4,
       4, 4, 3, 4, 1, 1, 1, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
       4, 4, 4, 4, 4, 4, 4, 4, 3, 3, 3, 3, 3, 3, 3, 4, 3, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0], dtype=int32)
```

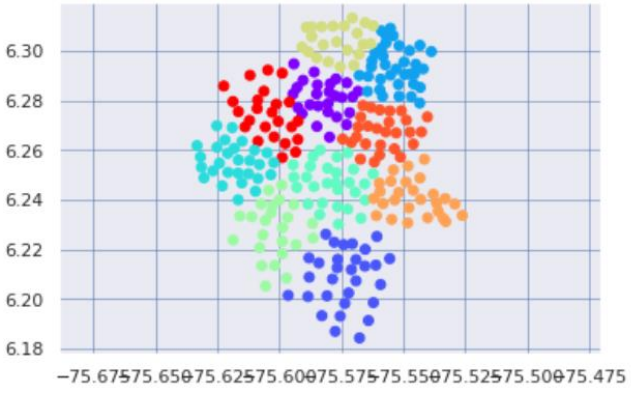
El gráfico resultante sería el siguiente:

```
1  plt.scatter(data['Long'],
2              data['Lat'],
3              c=datosAgrupados['Cluster'],
4              cmap='rainbow')
5  plt.axis('equal')
6  plt.grid(axis = 'both',color = 'b',alpha = 0.5)
```



Para un valor de 10 niveles de agrupamiento, el resultado será:

```
plt.scatter(data['Long'],
            data['Lat'],
            c=datosAgrupados['Cluster'],
            cmap='rainbow')
plt.axis('equal')
plt.grid(axis = 'both',color = 'b',alpha = 0.5)
```



Es importante anotar que, para seleccionar el número de clústeres, se puede usar la regla del codo o del pulgar, de la siguiente manera: `cluster = raíz(2/n).`

Como has notado, el anterior ejercicio ha permitido explorar otras alternativas de solución en términos del lenguaje de programación. Sin embargo, si estás interesado en indagar la forma de resolverlo, te recomendamos leer el artículo *R: A Language and Environment for Statistical Computing* (p.1519):

https://cran.r-project.org/doc/manuals/r-release/fullrefman.pdf

2.2. Discriminación

Para lograr aplicaciones de clasificación o discriminación, utilizaremos el método de *análisis discriminante,* el cual se conoce como una técnica inferencial, típicamente multivariante. Aunque este método se usa generalmente para casos donde se tengan diversas variables, también puede ser utilizada con pocas o una sola, sin que esto sea una práctica corriente.

El objetivo del análisis discriminante es identificar el patrón que siguen los datos para clasificar nuevas observaciones. Hay algunos conceptos claves asociados a este método:

- El análisis discriminante calcula la probabilidad de que una nueva observación pertenezca a una clase ya identificada.
- Es similar a la regresión logística, sobre todo cuando se tienen dos categorías.
- Se deben tener definidas dos o más poblaciones. Es decir, una serie de individuos de cada población de las cuales ya se han medido diversas variables.
- Este modelo tiene un buen desempeño cuando se tienen pocas observaciones.

En definitiva, su propósito es clasificar las observaciones y con base en esta clasificación asignar una nueva observación a alguna de las que ya se han encontrado. Por otro lado, se debe tener en cuenta que la nueva observación debe pertenecer a alguna de las clasificaciones.

Para entender mejor la aplicación del método, vamos a resolver el siguiente problema usando el lenguaje con el que más nos hemos familiarizado: *R*.

Problema

Considera la siguiente tabla con información de algunos estudiantes. Los datos en las columnas corresponden a:

- ***Aprobado***: indica si el estudiante aprobó o reprobó el año.
- ***Horas de estudio, Horas de diversión y Horas de transporte***: valor entero que corresponde al número de horas dedicada a cada uno de los eventos indicados.
- ***Nota de Matemáticas:*** valor entre 0 y 10.
- ***Mascota***: valor de 0 para *No* y un valor de 1 para *Sí*.
- ***Deporte***: valor de 0 para *No* y un valor de 1 para *Sí*.

A partir de estos datos, se debe generar una agrupación para los estudiantes que aprueban y los que reprueban. Asimismo, se requiere pronosticar si un nuevo estudiante aprueba o no el año en curso, usando los datos de horas de estudio, de diversión y transporte, así como la nota de Matemáticas.

Prueba el modelo con los siguientes casos:

Caso	Horas de estudio	Horas de diversión	Horas de transporte	Nota de Matemáticas
Caso 1	4	1.5	1.5	7
Caso 2	2	1	1	10
Caso 3	2	5	0	5
Caso 4	4	0	1	6

¿Cuáles son los resultados y que interpretación puedes hacer de estos?

dataAlumnos.csv

Solución

Carga el archivo en la interfaz de *R Cloud*, como lo has hecho hasta ahora. Luego, prepara tu entorno de trabajo:

```
rm(list=ls())
```

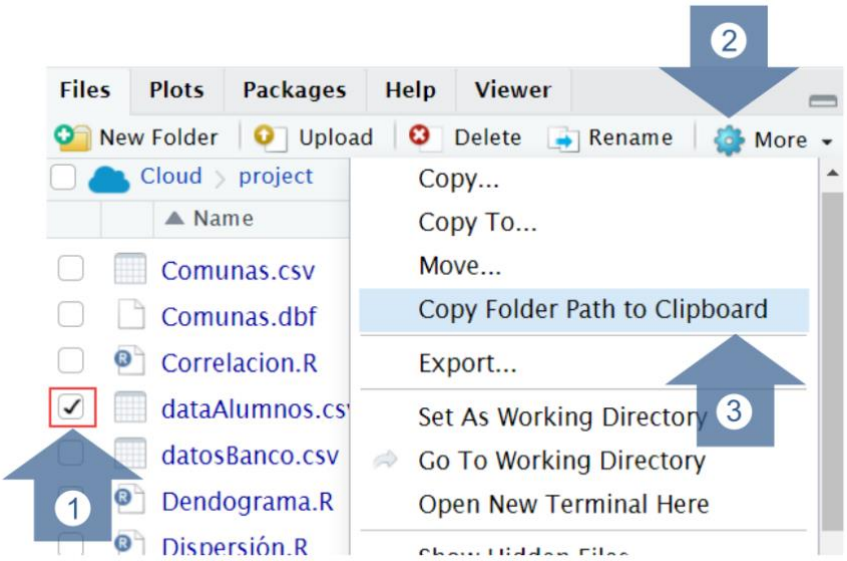
La primera instrucción limpiará tu entorno de trabajo, es decir, eliminarás *datasets*, variables y gráficos. Posteriormente, carga todas las librerías requeridas:

```
library(foreign)
library(ggplot2)
library(MASS)
```

Importa el archivo con los datos a un *dataset*. Esta vez lo harás desde el código y no con el botón de *Import Dataset*.

```
data = read.csv("/cloud/project/dataAlumnos.csv", sep=";")
```

La ruta se obtiene seleccionando el archivo, luego el ícono de la rueda dentada, y enseguida seleccionando la opción *Copy Folder Path to Clipboard* (copiar la ruta al portapapeles).



Define los factores ***Reprueba*** y ***Aprueba*** para la columna *Aprobado*, en lugar de 0 y 1, para así interpretar de mejor manera los resultados.

```
data$Aprobado<-factor(data$Aprobado,
                      levels = c(0,1),
                      labels = c("Reprueba","Aprueba"))
```

A continuación, aplica el modelo con el método ***lda()***, en el cual se consideran, para las categorías de la columna *Aprobado*, los valores de las variables que almacenan las horas de estudio, horas de diversión, horas de transporte, y la nota de Matemáticas.

```
dis=lda(Aprobado~HsEstudio+HsDiversion+HsTransporte+NotaMatematica,
    data=data,
    prior=c(0.5,0.5))
```

En este código se ha definido ***prior*** con unos valores de priorización de la probabilidad de que una observación tenga el 50% de ser agrupada en alguna de las categorías.

Enseguida, ejecuta la siguiente línea de código:

```
dis
```

La cual debe arrojar el siguiente resultado:

```
> dis
Call:
lda(Aprobado ~ HsEstudio + HsDiversion + HsTransporte + NotaMatematica,
    data = data, prior = c(0.5, 0.5))

Prior probabilities of groups:
Reprueba  Aprueba
     0.5      0.5

Group means:
         HsEstudio HsDiversion HsTransporte NotaMatematica
Reprueba  5.166667    2.500000     2.333333       2.833333
Aprueba   7.500000    2.083333     1.333333       7.750000

Coefficients of linear discriminants:
                     LD1
HsEstudio       0.2329660
HsDiversion     0.1542990
HsTransporte   -0.7837048
NotaMatematica  0.4242002
```

Esto permite identificar la función discriminante lineal como:

0.23*HsEstudio + 0.15*HsDiversion - 0.78*HsTransporte + 0.42*NotaMatematica

Sin embargo, esta conceptualización matemática no es la que se requiere, así que debes continuar con el código.

Ya que has entrenado el modelo con estos datos, para conformar la función discriminante, debes probar el modelo con los datos propuestos para predecir a qué grupo se encuentra asociada cada nueva observación.

El ejercicio te guiará en el abordaje de cada una para hacer una interpretación básica de los resultados.

Caso 1

Las siguientes líneas incorporan los datos esperados a un nuevo *dataset*. Agrega las columnas y lleva el *dataframe* de estos nuevos datos a la variable ***nuevo.data***.

```
nuevo.data=rbind(c(4.0,1.5,1.5,7.0))
colnames(nuevo.data)=colnames(data[,2:5])
nuevo.data=data.frame(nuevo.data)
```

A continuación, utiliza el método de predicción:

```
predict(dis,newdata = nuevo.data)
```

El resultado de la ejecución de estas líneas de código es el siguiente:

```
> predict(dis,newdata = nuevo.data)
$class
[1] Aprueba
Levels: Reprueba Aprueba

$posterior
  Reprueba  Aprueba
1 0.254997 0.745003
```

Podemos interpretar que, por medio de estos datos, la nueva observación, es decir, el nuevo estudiante, tiene una probabilidad del 25.49% de reprobar y un 74.50% de aprobar.

Caso 2

```
# DATOS CASO 2
nuevo.data=rbind(c(2,1,1,10))
colnames(nuevo.data)=colnames(data[,2:5])
nuevo.data=data.frame(nuevo.data)

# PREDICCION CASO 2
predict(dis,newdata = nuevo.data)
```

Resultado:

```
> predict(dis,newdata = nuevo.data)
$class
[1] Aprueba
Levels: Reprueba Aprueba

$posterior
     Reprueba   Aprueba
1 0.007945129 0.9920549
```

Un estudiante con 2 horas de estudio, 1 hora de diversión, 1 de transporte y con nota de 10 en Matemáticas, tiene una probabilidad inferior al 0.8% de reprobar y de 99.20% de aprobar.

Caso 3

```
# DATOS CASO 3
nuevo.data=rbind(c(2,5,0,5))
colnames(nuevo.data)=colnames(data[,2:5])
nuevo.data=data.frame(nuevo.data)

# PREDICCIÓN CASO 3
predict(dis,newdata = nuevo.data)
```

Resultado:

```
> predict(dis,newdata = nuevo.data)
$class
[1] Aprueba
Levels: Reprueba Aprueba

$posterior
    Reprueba   Aprueba
1 0.08197164 0.9180284
```

En este caso, encontramos que un estudiante con 2 horas de estudio, 5 horas de diversión, que no emplea tiempo en transportarse y con solo un 5 en Matemáticas, tiene una probabilidad de reprobar del 8.20% y de 91.80% de aprobar. Se destaca en este caso una nota baja y una cantidad superior en tiempo de entretenimiento.

Caso 4

```
# DATOS CASO 4
nuevo.data=rbind(c(4,0,1,6))
colnames(nuevo.data)=colnames(data[,2:5])
nuevo.data=data.frame(nuevo.data)

# PREDICCIÓN CASO 4
predict(dis,newdata = nuevo.data)
```

Resultados:

```
> predict(dis,newdata = nuevo.data)
$class
[1] Aprueba
Levels: Reprueba Aprueba

$posterior
   Reprueba   Aprueba
1 0.4529461 0.5470539
```

En este caso encontramos que un estudiante con 4 horas de estudio, sin tiempo para divertirse, 1 hora para transportarse y con nota de 6 en Matemáticas, tiene una probabilidad del 45.30% de reprobar y del 54.70% de aprobar. Nuevamente, llama la atención cómo la falta de tiempo de diversión no influye favorablemente al estudiante para aprobar.

¡Sigue explorando este caso con otros valores, y confirma o rechaza las interpretaciones que has hecho de los resultados obtenidos!

2.3. Agregación

Una de las técnicas de agregación más utilizada es el *Análisis de Componentes Principales* o PCA *(Principal Component Analysis)*. El objetivo de este método es crear constructos que sean capaces de representar el comportamiento de un grupo de variables. Justamente por esto también se le conoce como una técnica de reducción, lo que es bastante útil cuando se tiene una gran cantidad de datos.

Adicionalmente, es bastante útil para analizar comportamientos que no se pueden medir de manera directa, lo cual la destaca como una de las técnicas de aprendizaje no supervisado.

Mediante el proceso de reducción, se espera que los *componentes principales* que se obtengan, logren explicar la mayor variabilidad de los datos.

Los elementos relevantes que deben considerarse en el PCA son:

- Los *Componentes Principales* (CP), también denominados constructos, reflejan el comportamiento de las variables originales.
- El número de *componentes principales* depende de la cantidad de variables originales. Hay que decir que el mayor número de constructos es igual al número de variables, y esto podría suceder porque no es posible encontrar una forma en que una variable pueda representar por lo menos a dos de las variables originales.
- Cada *componente principal* o dimensión será una combinación lineal de las variables originales, que además no serán correlacionadas entre sí o, dicho de otro modo, independientes.
- En la práctica, las variables deben ser numéricas y estandarizadas al ingresar al modelo para compararlas fácilmente.

Problema

Se han recogido algunos datos de un grupo de 150 jóvenes sobre la manera en que ocupan su tiempo. Así pues, se ha obtenido información acerca del tiempo que emplean estudiando, leyendo, caminando y asistiendo al gimnasio.
¿Es posible identificar cierto perfil en estos jóvenes?

actividades.csv

Solución

Por el tipo del conjunto de datos con que contamos, y la naturaleza de la pregunta que esperamos resolver, lo más adecuado es utilizar el método de *análisis de componentes principales*.

Para resolverlo, prepara tu entorno de trabajo y ajusta el modelo para obtener una descripción de este. Al final, podrás identificar si es posible responder a la pregunta que se plantea.

En primer lugar, prepara el entorno de trabajo. Ya sabes que la función `rm(list=ls())` permite "limpiar" el entorno eliminando variables, *datasets,* entre otros.

Por otro lado, la función `graphics.off()` evita que se apilen y se guarden de manera temporal los gráficos que se generen, reemplazando cualquier gráfico que haya sido creado previamente.

```
rm(list=ls())
graphics.off()
```

Posteriormente, carga las librerías con las cuales vas a trabajar.

```
library(foreign)
library(factoextra)
library(dplyr)
library(knitr)
```

Si alguna librería no ha sido instalada, *R* informará sobre ello. Solo debes ejecutar las instrucciones de instalación y luego cargar las librerías.

```
install.packages("factoextra")
install.packages("dplyr")
install.packages("knitr")
```

Las que se muestran en el ejemplo, son las librerías que generalmente no están instaladas por defecto en *R Cloud*.

Luego, carga el archivo que contiene los datos y visualiza su contenido para hacerte a una idea de su estado.

```
datos = read.csv("/cloud/project/actividades.csv", sep = ";")
View(datos)
```

El resultado que se obtiene es una nueva pestaña con la tabla de los datos cargados, como se puede ver a continuación:

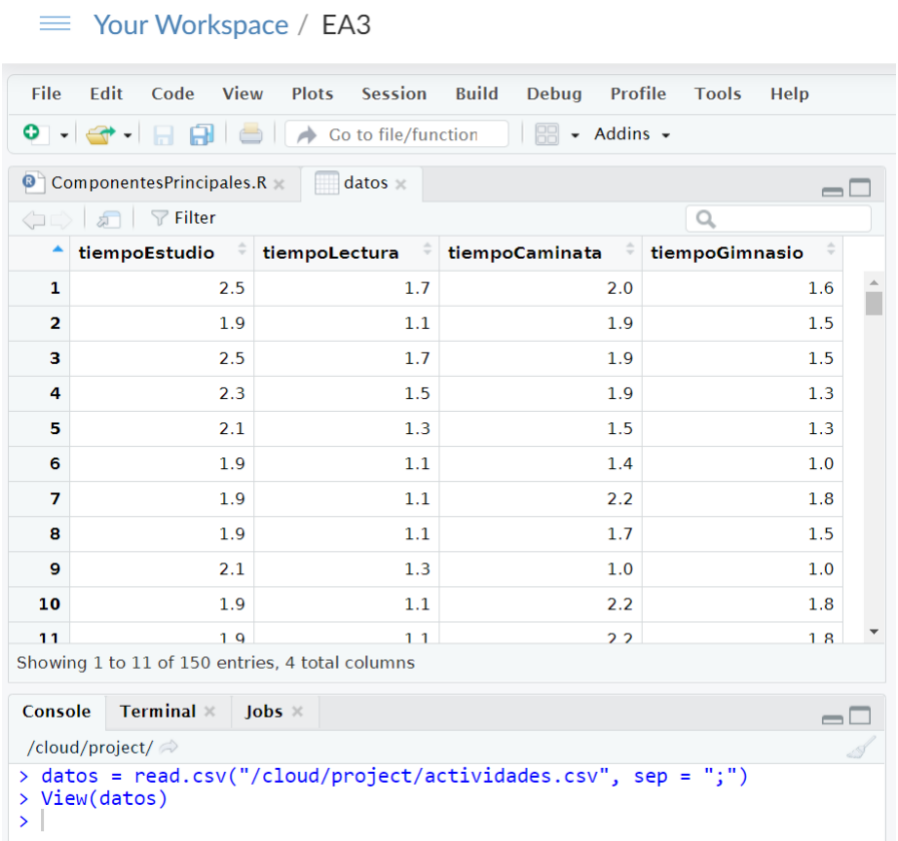


Figura 21. Visualización del *dataframe* en *R Cloud*

Inicia la construcción del modelo, y para ajustarlo, lo primero es hacer una correlación para entender el comportamiento de las variables. Aunque la idea es que todas las variables estén correlacionadas, esto no es un requisito.

```
cor(datos)
```

El resultado que se obtiene es el siguiente:

```
> cor(datos)
              tiempoEstudio tiempoLectura tiempoCaminata tiempoGimnasio
tiempoEstudio    1.00000000    1.00000000    -0.01416573      0.1614958
tiempoLectura    1.00000000    1.00000000    -0.01416573      0.1614958
tiempoCaminata  -0.01416573   -0.01416573     1.00000000      0.7713158
tiempoGimnasio   0.16149582    0.16149582     0.77131583      1.0000000
```

Es posible redondear estos datos para una vista más clara de estos resultados:

```
> round(cor(datos),2)
              tiempoEstudio tiempoLectura tiempoCaminata tiempoGimnasio
tiempoEstudio          1.00          1.00          -0.01           0.16
tiempoLectura          1.00          1.00          -0.01           0.16
tiempoCaminata        -0.01         -0.01           1.00           0.77
tiempoGimnasio         0.16          0.16           0.77           1.00
```

Como puedes ver, el tiempo de lectura está altamente correlacionado con el tiempo de estudio. Podemos afirmar esto ya que el coeficiente de correlación es igual o está bastante cerca de 1 o -1 (recordemos que, si estos valores están cerca de 0, no hay ninguna influencia entre las variables). Adicionalmente, hay una fuerte correlación entre el tiempo que se dedica al gimnasio y el que se dedica a caminar.

En el siguiente paso, debes normalizar las variables, es decir, llévalas todas a un mismo rango para que sean comparables.

```
norm <- function(x){(x-min(x))/(max(x)-min(x))}
```

Esta función, que almacenarás en `norm`, hace que las variables ya no presenten cambios de escala, reasignándoles valores que van entre 0 y 1.

A continuación, aplica la función al *dataset* original y crea uno nuevo que llamarás `datosNorm` (datos normalizados).

```
datosNorm<-data.frame(apply(datos,2,norm))
View(datosNorm)
```

El resultado del nuevo *dataset* será:

	tiempoEstudio	tiempoLectura	tiempoCaminata	tiempoGimnasio
1	0.8	0.8	0.8333333	0.750
2	0.2	0.2	0.7500000	0.625
3	0.8	0.8	0.7500000	0.625
4	0.6	0.6	0.7500000	0.375
5	0.4	0.4	0.4166667	0.375
6	0.2	0.2	0.3333333	0.000

Ahora calcula los valores: mínimo, medio y máximo, y los redondeamos a dos dígitos decimales.

```
round(apply(datosNorm, 2, min),2)
round(apply(datosNorm, 2, mean),2)
round(apply(datosNorm, 2, max),2)
```

Como puede verse, están en un rango entre 0 y 1, expresados además con solo 2 decimales:

```
> round(apply(datosNorm, 2, min),2)
 tiempoEstudio  tiempoLectura tiempoCaminata tiempoGimnasio
             0              0              0             0
> round(apply(datosNorm, 2, mean),2)
 tiempoEstudio  tiempoLectura tiempoCaminata tiempoGimnasio
          0.51           0.51           0.59          0.50
> round(apply(datosNorm, 2, max),2)
 tiempoEstudio  tiempoLectura tiempoCaminata tiempoGimnasio
             1              1              1             1
```

Después de haber normalizado el *dataset*, será posible hacer el *Análisis de Componentes Principales*, utilizando la función ***prcomp()***, a la cual enviaremos como parámetro el *dataset* normalizado.

```
acp = prcomp(datosNorm)
acp
```

Obteniendo:

```
> acp = prcomp(datosNorm)
> acp
Standard deviations (1, .., p=4):
[1] 4.092793e-01 3.538871e-01 1.239794e-01 1.193133e-16

Rotation (n x k) = (4 x 4):
                     PC1        PC2         PC3           PC4
tiempoEstudio  -0.6728827 -0.2074497  0.06475714 -7.071068e-01
tiempoLectura  -0.6728827 -0.2074497  0.06475714  7.071068e-01
tiempoCaminata -0.1416852  0.6831884  0.71636508  8.673617e-17
tiempoGimnasio -0.2727326  0.6687173 -0.69168931 -6.938894e-17
```

Tras obtener este resultado, presta especial atención a las cargas factoriales, que equivalen a las distancias que hay entre cada dato (punto) y el origen a lo largo del *componente principal*. Debido a que el modelo tiene como objetivo encontrar la mayor variabilidad de los datos, identifica la varianza acumulada de los componentes, empleando el gráfico de sedimentación.

```
screeplot(acp, type="lines")
```

El aspecto del gráfico es el siguiente:

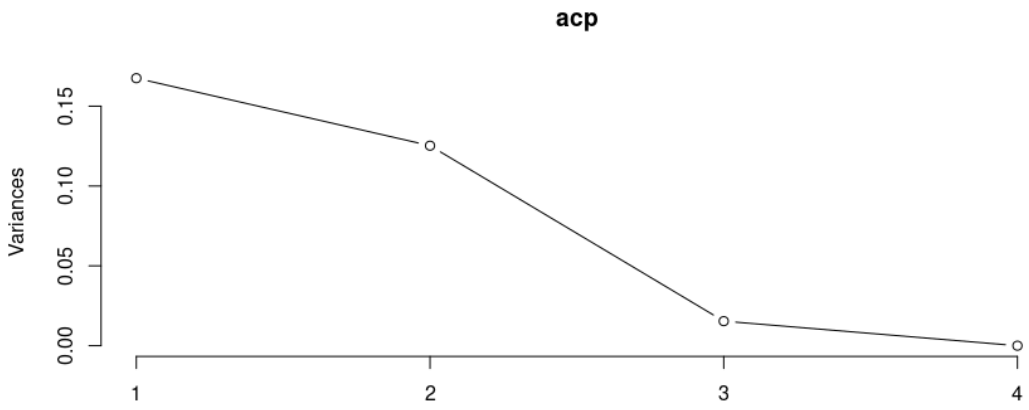


Figura 22. Gráfico tipo linea

Se puede interpretar que el primer componente logra captar la mayor variabilidad, mientras que el segundo capta una varianza menor. Los componentes 3 y 4, como puede verse, captan una variabilidad muy pequeña con respecto a los dos primeros. Con base en este análisis, puedes seleccionar solamente los dos primeros componentes.

```
cp<-data.frame(acp$x)
cp<-cp[,1:2]
```

En la primera instrucción de este código, crea un *dataframe* llamado `cp` con los componentes del *análisis de componentes principales* (`acp`). En la segunda instrucción, reemplaza lo que hay en `cp` por los que corresponden al *componente principal* 1 y 2.

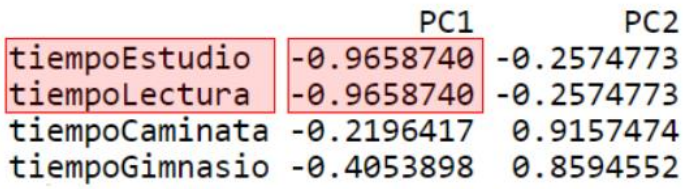
Calcula la correlación entre los datos originales y los *componentes principales* para ver su comportamiento.

```
cor(datos, cp, use = "everything", method = c("pearson"))
```

Obteniendo lo siguiente:

```
> cor(datos, cp, use = "everything", method = c("pearson"))
                    PC1        PC2
tiempoEstudio  -0.9658740 -0.2574773
tiempoLectura  -0.9658740 -0.2574773
tiempoCaminata -0.2196417  0.9157474
tiempoGimnasio -0.4053898  0.8594552
```

A partir de los resultados, se puede ver que en el componente principal - PC1, los mayores valores corresponden a las variables ***tiempoEstudio*** y ***tiempoLectura***.



Por otro lado, en el componente principal – PC2, estos valores corresponden a las variables ***tiempoCaminata*** y ***tiempoGimnasio***.

	PC1	PC2
tiempoEstudio	-0.9658740	-0.2574773
tiempoLectura	-0.9658740	-0.2574773
tiempoCaminata	-0.2196417	0.9157474
tiempoGimnasio	-0.4053898	0.8594552

El análisis que acabas de realizar permite un acercamiento a la construcción de los perfiles. Se puede afirmar que existen dos tipos de jóvenes, unos que son más académicos y sedentarios, mientras que otros son más activos y gustan de actividades fuera de sus casas. Así, podría establecerse una hipótesis acerca del comportamiento a partir de los perfiles identificados.

¡Felicitaciones!

Has finalizado con éxito la actividad de aprendizaje de la Unidad 4.

Los conocimientos que has adquirido a lo largo del estudio de estos temas, además de ser muy interesantes, te serán de gran ayuda en el desempeño de tu actividad académica y laboral.

Analizar enormes cantidades de datos y controlar las variables para comprender su relación, permiten entender contextos, plantear estrategias y resolver problemas, elementos vitales para la toma de decisiones en proyectos, empresas o comunidades.

A continuación, te invitamos a poner a prueba tus conocimientos con la evidencia de aprendizaje 3. Lee muy bien el problema a resolver, qué condiciones y restricciones tiene, y define cuál es la mejor forma de solucionarlo.

Evidencia de aprendizaje (EA) de la Unidad 4:

Pronóstico de tratamiento para paciente con cáncer de mama

Nombre de la evidencia de aprendizaje	Pronóstico de tratamiento para paciente con cáncer de mama.
Objetivo de la evidencia de aprendizaje	A partir de los datos de un grupo de pacientes con cáncer de mama, aplicar una técnica que permita pronosticar si una nueva paciente, que no se encuentra en los datos originales, será sometida a un tratamiento de irradiación.
Contenidos	Los datos para la agrupación de los mismos, y los de la paciente que debe pronosticarse, se aprovisionan en el planteamiento del problema.
Descripción de lo que debe hacer el estudiante	El estudiante debe descargar los datos, y subirlos a R Studio para diseñar el script que permita resolver el problema planteado. Luego, debe descargar el script de la solución y subirlo a la plataforma en la actividad.
Especifique lo que debe entregar el estudiante	El estudiante debe diseñar un script en R, utilizando la plataforma online de R Studio, descargar el script y subirlo como evidencia a la plataforma.

Referencias Bibliográficas

- Diseño Gráfico: Steven Miranda Cardona. Dirección de Tecnología. IUD

- *Análisis exploratorio de datos*. (2020). Universitat de Barcelona. http://www.ub.edu/aplica_infor/spss/cap2-3.htm

- Arroyo, J. (2016). Métodos de reducción de dimensionalidad: Análisis comparativo de los métodos APC, ACPP y ACPK. *UNICIENCIA, 30*(1), 115-122. doi: http://dx.doi.org/10.15359/ru.30-1.7

- CEACES. (2020). *ANÁLISIS CLUSTER*. Universitat de València. https://www.uv.es/ceaces/multivari/cluster/CLUSTER2.htm

- Conexión ESAN. (2019). *Diagrama de dispersión: ¿cómo usar esta herramienta de control de calidad?* ESAN Graduate School of Business. https://www.esan.edu.pe/apuntes-empresariales/2019/10/diagrama-de-dispersion-como-usar-esta-herramienta-de-control-de-calidad/

- Constructo. (2017). En *Glosarios de términos especializados de las Ciencias, las Artes, las Técnicas y la Sociedad*. https://glosarios.servidor-alicante.com/terminos-estadistica/constructo

- Dagnino, J. (2014). Muestras, variabilidad y error. *Revista Chilena de Anestesia, 43*, 100-103. https://revistachilenadeanestesia.cl/PII/revchilanestv43n02.04.pdf

- De La Fuente, S. (2011). *Análisis Factorial*. Universidad Autónoma de Madrid. https://www.fuenterrebollo.com/Economicas/ECONOMETRIA/MULTIVARIANTE/FACTORIAL/analisis-factorial.pdf

- López, J. F. (2021). Covarianza. En *Economipedia*. https://economipedia.com/definiciones/covarianza.html

- Marta. (2020). *Parámetros estadísticos descriptivos*. Superprof https://www.superprof.es/apuntes/escolar/matematicas/estadistica/descriptiva/parametros-estadisticos.html

- Minitab. (2019). *Significancia estadística y práctica. Minitab*. https://support.minitab.com/es-mx/minitab/18/help-and-how-to/statistics/basic-statistics/supporting-topics/basics/statistical-and-practical-significance/

- Pardo, C. (2020). *¿Qué son las redes neuronales y cómo se aplican?* Enzyme Advising Group. https://blog.enzymeadvisinggroup.com/redes-neuronales-que-son-y-aplicaciones

- Pérez, J. & Merino, M. (2018). Centroide. En *Definicion.de*. https://definicion.de/centroide/

- Pumanin, D. (2020). Agregación. En *Hypergeo*. https://www.hypergeo.eu/spip.php?article147

- Quintela, A. (2019). *Hipótesis estadísticas*. Bookdown. https://bookdown.org/aquintela/EBE/hipotesis-estadisticas.html

- Ruiz, J. (2020). *Caracterización de variables cualitativas*. Aula Virtual Matemáticas. https://sites.google.com/site/matematicasjuanmanuelista/matematicas-8/estadistica-8/1-3-caracterizacion-de-variables-cualitativas

- Sacau, P. (2004). *Definición de matriz*. Descartes 3D. http://recursostic.educacion.es/descartes/web/materiales_didacticos/Calculo_matricial_d3/defmat.htm

- Salinas, J. M. (2020). *Estadísticos de dispersión*. Universidad de Granada. https://www.ugr.es/~jsalinas/apuntes/C3.pdf

- Salvador, M. (2001). *Análisis de conglomerados o clúster*. 5campus.org. https://ciberconta.unizar.es/LECCION/cluster/inicio.html

- Varianza. (2019). En *Software DELSOL*. https://www.sdelsol.com/glosario/varianza/

- Westreicher, G. (2019). Predicción (estadística). En *Economipedia*. https://economipedia.com/definiciones/prediccion-estadistica.html

IU Digital de Antioquia

INSTITUCIÓN UNIVERSITARIA DIGITAL DE ANTIOQUIA

ORD No 74 de 2017 · VIGILADA MinEducación