De los datos a la información

El dato en la inteligencia artificial

CONTENIDO

1. Objetivos
2. Análisis de datos
3. Tipos de análisis
4. Extracción de información de un *dataset*
   - Principales resúmenes estadísticos
   - Análisis univariable
   - Análisis multivariable
5. Bibliografía

Objetivos

- Comprender que es el análisis de datos y los tipos de análisis que podemos realizar sobre un conjunto.
- Conocer los principales análisis estadísticos que aportan valor al conjunto de datos para su análisis, haciendo uso de las distintas funciones predefinidas que proporciona Python, de forma que pueda extraer conclusiones.
- Identificar las características intrínsecas de cada variable, y su impacto dentro del conjunto, de forma que pueda determinar su importancia dentro del conjunto y las relaciones en las que se ven involucradas.

Análisis de datos

En el temario se evidenciaron los principales tipos de datos tanto básicos como complejos que existen, las operaciones que se pueden realizar con ellos y los distintos flujos de control que existen.

Una vez que se conozcan estos aspectos imprescindibles previos al análisis de datos, se puede dar paso a comprender que implica este.

Como indica Guillermo Westreicher en Economipedia [1], la definición técnicas es que "El análisis de datos es el estudio exhaustivo de un conjunto de información cuyo objetivo es obtener conclusiones que permitan a una empresa o entidad tomar una decisión".

Existen múltiples definiciones para este concepto tan importante hoy en día, que como se indica en el informe anterior, es un proceso de la ciencia de datos, que incluye también la creación de modelos para poder usar el análisis de esta información en la generación de predicciones.

Tipos de análisis

Debido a la diversidad con la que se cuenta en los datos, existen diferentes formas de realizar el análisis, sin embargo, retomando el informe anterior [1], la forma más común de clasificarlos es en los dos siguientes grupos:

- **Cuantitativo.** Es el análisis que se realiza sobre los datos de forma numérica aplicando métodos estadísticos.
- **Cualitativo.** Es el análisis que no se basa en datos numéricos, sino en clasificaciones categóricas textuales, al que también se le puede aplicar diferentes métodos estadísticos.

Un conjunto de datos puede presentar ambos tipos de características. De hecho, un mismo atributo se puede presentar en ambas formas. Por ejemplo, la valoración de un producto puede ser de forma cualitativa, en las categorías "Malo, Normal y Bueno", o también de forma numérica del 1 al 5.

Además de este tipo de división, se evidenciará como se pueden usar los métodos estadísticos de forma numérica a lo largo de este temario. Para añadir valor a este, se analizará cómo también se puede mostrar esta información estadística de forma gráfica.

Extracción de información de un *dataset*

Antes de comenzar con la labor del análisis estadístico, para poder extraer conocimiento sobre un conjunto de datos es necesario plantear ciertas preguntas sobre las propiedades de los datos, como, por ejemplo:

- **¿De qué tipo son nuestros valores?** No se pueden aplicar las mismas técnicas si se tienen valores categóricos o si, por el contrario, los valores son numéricos.
- **¿Tienen todos la misma magnitud?** Es importante tener en cuenta en el análisis las magnitudes de los datos, y si es conveniente o no modificar su escala para poder realizar comparaciones.
- **¿Están muy dispersos nuestros datos?** Tiene que existir suficiente variabilidad para que un atributo sea característico y permita diferenciar ciertas condiciones en el *dataset*.
- **¿Siguen los atributos una distribución normal?** Es muy importante conocer su distribución, ya que muchos métodos asumen que esta es normal.
- **¿Tenemos valores atípicos?** ¿Son valores inusuales pero posibles, o se pueden descartar como errores?
- **¿Cómo influyen unas variables en otras?** Se debe comprobar si existe una relación entre las variables.
- **¿Son todos los atributos importantes?** Puede que dos atributos expresen la misma información, pero con distinta magnitud.

Estas preguntas son genéricas a todos los problemas, y permitirán extraer conclusiones sobre los datos que fundamenten la selección del algoritmo de modelado a aplicar.

Para dar respuesta a estas preguntas, es fundamental conocer las características que los definen, analizándolos mediante métodos estadísticos, ya sea numéricos o gráficos.

Aunque se comentará en posteriores secciones la diferencia entre los tipos de predicciones a realizar sobre los datos, también se debe conocer si el problema es de clasificación o regresión. Si es de regresión, el objetivo es predecir un valor continuo en función de ciertos atributos; por ejemplo, el tiempo que tardará en necesitar reparación una máquina en función del calor al que se encuentra expuesta, su antigüedad, marca y potencia. Por el contrario, si es de clasificación, se tiene una variable objetivo o *target* dividido en clases entre las que se quiere agrupar una instancia dada. Por ejemplo, Iris, donde en función de las medidas de su pétalo y sépalo se quiere saber a qué clase de flor pertenece una instancia.

Este tipo de problemas permiten aplicar análisis de los atributos en función de la clase, de forma que se puede determinar la influencia que ejerce cada uno de ellos. En las siguientes secciones se va a usar este *dataset* de clasificación.

Para evaluar todo lo anterior, se van a dividir los tipos de análisis en función de la dimensión, resumiéndose en univariables, bivariables y multivariables. Los bivariables son un caso concreto de los multivariables, donde solo se analizan dos propiedades de forma conjunta.

Principales resúmenes estadísticos

Los resúmenes estadísticos son aquellos que se basan en la estadística descriptiva para, como su nombre indica, describir las propiedades de los datos y el conjunto.

Estos ofrecen una visión global sintetizada de las características de los datos, cuyas principales métricas se reflejan en la *(tabla 1)*.

Para evaluar los resúmenes estadísticos de los que se dispone se va a usar el *dataset iris* del repositorio UCI *machine learning* [2], donde se puede encontrar gran cantidad de conjuntos de datos disponibles. Se puede cargar el *dataset* leyéndolo desde un fichero local como se hizo en los temas anteriores, o usar el módulo *dataset* de la librería *sklearn*. Una vez cargado, se puede usar la función "head()" para ver los atributos de este dataset. Se observa como la clase de flor se determina por 4 características referentes a las medidas del sépalo y pétalo *(figura 1)*.

La función "describe()" permite obtener un resumen estadístico de un *data frame*:

```
dataset.describe() # dataset obtenido con pd.read_csv('iris.data')
```

Al ejecutar el comando sobre el *dataset*, mostrará por defecto los siguientes estadísticos, teniendo en cuenta que todas las columnas son numéricas *(figura 2)*.

Se puede extraer varias conclusiones de este análisis obtenido para cada columna:

- ***Count:*** esta columna muestra el número de muestras para esta propiedad. En este caso se evidencia que se dispone de 150 muestras por característica. Es un atributo muy importante ya que se puede ver a simple vista si se cuenta con pocos datos para un atributo respecto al tamaño del *dataset* y, por tanto, no es conveniente utilizar dicha propiedad en el análisis.
- ***Mean:*** indica el valor medio de una propiedad considerando todos los valores con el mismo peso. Si el tipo de la propiedad es numérico no ordenado, se puede usar la media, pero para valores ordenados o categóricos, se necesita usar la moda.
- **Std:** muestra la dispersión de los valores, es decir, la medida en la que se alejan o no de la media. Se puede ver que esta será mayor conforme más distancia exista entre el valor mínimo y el máximo. Debido a que es sensible a la magnitud de la variable, es conveniente normalizar los datos.
- **Mínimo y máximo:** valor mínimo y máximo de un atributo.
- **25 %, 50 %, 75 %:** corresponden a los cuartiles de cada muestra. Por el ejemplo, el cuartil 25 indica el valor a partir del cual se encontrarán por debajo el 25 % de los datos de la muestra. Estos valores son muy importantes ya que indican la probabilidad de que un nuevo valor se encuentre por debajo de un determinado umbral.

Estadístico	Descripción	Definición
Media aritmética	Valor característico que indica el centro de gravedad de la muestra.	mean()
Mediana	Valor central del conjunto ordenado de una variable.	median()
Moda	Valor que aparece con más frecuencia en un conjunto de datos.	mode()
Desviación típica	Cuantifica la dispersión del conjunto de datos.	std()
Varianza	Corresponde a la esperanza del cuadrado de la desviación típica de dicha variable respecto a su media.	var()

Tabla 1. Principales resúmenes estadísticos en Python.

	Longitud sépalo	Ancho sépalo	Longitud pétalo	Ancho pétalo	Clase
0	5.1	3.5	1.4	0.2	Iris - setosa
1	4.9	3.0	1.4	0.2	Iris - setosa
2	4.7	3.2	1.3	0.2	Iris - setosa
3	4.6	3.1	1.5	0.2	Iris - setosa
4	5.0	3.6	1.4	0.2	Iris - setosa

Figura 1. Primeras instancias del dataset Iris en Python.

	Longitud sépalo	Ancho sépalo	Longitud pétalo	Ancho pétalo
Count	150.000000	150.000000	150.000000	150.000000
Mean	5.8433333	3.054000	3.758667	1.198667
Std	0.828066	0.433594	1.764420	0.763161
Min	4.300000	2.000000	1.000000	0.100000
25 %	5.100000	2.800000	1.600000	0.300000
50 %	5.800000	3.000000	4.350000	1.300000
75 %	6.400000	3.300000	5.100000	1.800000
Max	7.900000	4.400000	6.900000	2.500000

Figura 2. Resumen estadístico del dataset Iris en Python.

Si ahora se selecciona un *dataset* que incluya datos categóricos, se puede ver como el resumen estadístico cambia. Se va a cargar, por ejemplo, "breast cancer" [3] que contiene datos de pacientes con cáncer de mama.

En sus atributos se puede observar que numérico es solo el atributo "deg-malig", mientras que el resto son categóricos o lógicos *(figura 3)*.

Para poder obtener el resumen estadístico de todos los valores, se tiene que indicar el parámetro "include= all":

```
dataset.describe(include="all") # dataset obtenido con
pd.read_csv('breast-cancer.data')
```

Al tratarse de un dataset con atributos categóricos, se puede observar que muchos de los valores obtenidos son *NaN* (*not a number*), ya que no puede calcularse, por ejemplo, la media aritmética de un rango o de la clase.

Sin embargo, incluye medidas adicionales como *(figura 4)*:

- ***Unique:*** número de valores categóricos distintos. Por ejemplo, en este *dataset* la clase únicamente puede tomar 2 valores (tener cáncer o no).
- ***Top:*** hace referencia a la moda siendo el valor más frecuente. Para la clase lo más frecuente será no tener cáncer, ya que el porcentaje de personas sanas es mayor.
- ***Freq:*** frecuencia del valor más común. Si este *dataset* cuenta con 286 instancias, como podemos ver en *count*, 201 serían pacientes sanos.

Análisis univariable

El análisis univariante permite focalizar el estudio en cada variable, observando su comportamiento y distribución. Algunos de los estadísticos que se evidencian de forma independiente se incluyen por defecto en el resumen estadístico anterior. A continuación, se mostrarán las siguientes cuestiones:

- **Distribución de frecuencias.** Muestra cómo se reparten los valores de un atributo.
- **Media/moda/mediana.** Muestra cuál es su centro de gravedad o valor más frecuente.
- **Desviación típica / coeficiente de variación.** Permite determinar su dispersión.
- **Rango intercuartílico / percentiles.** Muestra cuáles son sus distribuciones probabilísticas.
- **Normalidad.** Indica si su tipo de distribución se asemeja a una normal.

En primer lugar, se va a estudiar sobre el *dataset Iris* la distribución de frecuencias de un atributo, que varía en función del tipo de dato que se trate, ya que hay que tener en cuenta que cuando es categórico o entero sí que se puede contar cuántas veces aparece un valor, pero con valores reales será necesario establecer rangos.

```
dataset['Clase'].value_counts() #Nos muestra la frecuencia
por valor categórico

dataset['Longitud sépalo'].value_counts(bins=5) #Nos
muestra la frecuencia agrupada en rangos para los valores
numéricos
```

	class	age	menopause	tumor-size	inv-nodes	node-caps	deg-malig	breast	breast-quad	irradiat
	no-recurrence-events	30-39	premeno	30-34	0-2	no	3	left	left_low	no
	no-recurrence-events	40-49	premeno	20-24	0-2	no	2	left	right_up	no
	no-recurrence-events	40-49	premeno	20-24	0-2	no	2	left	right_low	no
	no-recurrence-events	60-69	ge40	15-19	0-2	no	2	left	right_low	no
	no-recurrence-events	40-49	premeno	0-4	0-2	no	2	left	right_low	no

Figura 3. Primeras instancias del *dataset* "breast cancer" en Python.

	class	age	menopause	tumor-size	inv-nodes	node-caps	deg-malig	breast	breast-quad	irradiat
count	286	286	286	286	286	286	286.000000	286	286	286
unique	2	6	3	11	7	3	NaN	2	6	2
top	no-recurrence-events	50-59	premeno	30-34	0-2	no	NaN	left	left_low	no
freq	201	96	150	60	213	222	NaN	152	110	218
mean	NaN	NaN	NaN	NaN	NaN	NaN	2.048951	NaN	NaN	NaN
std	NaN	NaN	NaN	NaN	NaN	NaN	0.738217	NaN	NaN	NaN
min	NaN	NaN	NaN	NaN	NaN	NaN	1.000000	NaN	NaN	NaN
25%	NaN	NaN	NaN	NaN	NaN	NaN	2.000000	NaN	NaN	NaN
50%	NaN	NaN	NaN	NaN	NaN	NaN	2.000000	NaN	NaN	NaN
75%	NaN	NaN	NaN	NaN	NaN	NaN	3.000000	NaN	NaN	NaN
max	NaN	NaN	NaN	NaN	NaN	NaN	3.000000	NaN	NaN	NaN

Figura 4. Resumen estadístico del *dataset* "breast cancer" en Python.

Respecto a la clase se puede ver si un *dataset* está o no balanceado. Es decir, si el número de observaciones que se tiene de una clase es muy inferior al de otras, al aplicar un modelo este las tendrá menos en cuenta, produciendo una desventaja en su clasificación.

Por otro lado, se puede visualizar cómo se distribuye una determinada propiedad. En este caso la longitud del sépalo tiende a valores cercanos al rango [5.02,6.46], siendo menos común las flores de sépalo de 7 centímetros.

El análisis univariante tiene que ir acompañado del bivariante, donde se podría verificar si este sépalo mayor se debe a que dentro de una determinada clase existe una tendencia a ello *(figura 5)*.

Otro aspecto relevante a observar es la media, moda y mediana. Se puede escoger, al igual que con la frecuencia, las mismas columnas del *data frame*.

```
dataset['Longitud sépalo'].mean() #Media de la longitud del sépalo: 5.843333333333334

dataset['Longitud sépalo'].median() #Mediana de la longitud del sépalo: 5.8

dataset['Longitud sépalo'].mode()[0] #Moda de la longitud del sépalo: 5.0
```

Aunque en este caso no se observa tan claramente el efecto de cada una de ellas, se puede observar en un ejemplo sencillo cómo la media se ve afectada por los valores extremos.

Hay que imaginar que se tiene las alturas de una clase, donde uno de ellos es muy alto respecto al resto.

```
altura alumnos = pd.DataFrame([2.18, 1.72, 1.72, 1.69, 1.68, 1.63, 1.56])

print("Media: ",altura_alumnos[0].mean()) #Media: 1.7400000000000002

print("Mediana: ",altura_alumnos[0].median()) #Mediana: 1.69

print("Moda: ",altura_alumnos[0].mode()[0]) #Moda: 1.72
```

Se puede ver cómo, aunque la mayoría de alumnos se encuentren en el rango [1.72,1.68], la media se desplaza hasta 1.74 por el peso del alumno más alto, mientras que la mediana se mantiene en 1.69, que se puede entender como el valor cercano más probable que presentará un alumno nuevo. Por último, la moda es el valor que más se repite, en este caso 1.72.

```
Iris - virginica  50
Iris - versicolor 50
Iris - setosa           50
Name: Clase, dtype:     int64
```

```
(5.74, 6.46]      42
(5.02, 5.74]      41
(4.295, 5,02]     32
(6.46, 7.18]      24
(7.18, 7.9]       11
Name: Longitud s+epalo, dtype: int64
```

Figura 5. Frecuencias de la clase y la longitud del sépalo en Python.

El tercer factor a tener en cuenta es la desviación típica, que mide la dispersión de los datos respecto a la media. Este permite conocer, por ejemplo, en el caso anterior, si la altura de la clase es más o menos homogénea. Si se vuelve al caso de Iris, se puede ver si existe más dispersión en el ancho o el largo del sépalo.

```
dataset['Longitud sépalo'].std() #Desviación típica de la longitud del sépalo: 0.828066127

dataset['Ancho sépalo'].std() #Desviación típica del ancho del sépalo: 0.43359431136217
```

Se observa que la dispersión es mayor en la longitud que en el ancho. Es decir, si no se conocen las medidas de una nueva flor y se asigna la media, el error probablemente sería mayor en la longitud que en su anchura.

Sin embargo, ¿es adecuada esta comparación? Imagine que se quiere estimar aleatoriamente la altura y peso de una persona, y se confunde en 1 metro, y 3 kilos. Se está cometiendo un mayor error en el peso si se evalúa la cantidad sin tener en cuenta las unidades, pero, a nivel relativo, en la altura el error es mayor porque el rango de estimación es mucho menor que si se refiere al peso.

Para evitar este problema, el coeficiente de variación (CV) es adimensional, y permite valorar la variación de una muestra. Se obtiene con la división de la desviación típica entre la media. En el caso anterior, para la longitud del sépalo y el ancho, se evidencia que el coeficiente de variación (la distribución de sus valores respecto a la media), es similar. Si el CV es menor o igual al 30 %, significa que la media aritmética es representativa del conjunto de datos.

```
import scipy.stats as ss

ss.variation(dataset['Longitud sépalo']) # C.V. de la longitud del sépalo: 0.1412380989934

ss.variation(dataset['Ancho sépalo']) #C.V. del ancho del sépalo: 0.141501827135083
```

Otra medida que muestra la distribución de los datos es el rango intercuartilico (IQR), que indica la diferencia entre el tercer y el primer cuartil. Un percentil concreto, indica el valor a partir del cual se encuentra un porcentaje de las observaciones.

Por ejemplo, si se indica el cuartil 25, se mostrará el valor a partir del cual se encuentran por debajo del 25 % de las observaciones.

```
dataset['Longitud sépalo'].quantile(0.25) #Cuartil 25 de la longitud del sépalo: 5.1

dataset['Longitud sépalo'].quantile(0.75) #Cuartil 75 de la longitud del sépalo: 6.4

dataset['Longitud sépalo'].quantile(0.75) - dataset['Longitud sépalo'].quantile(0.25)  #IQR de la longitud del sépalo: 1.3
```

Se observa que para la longitud del sépalo los valores centrales se encuentran en el rango [5.1,6.4].

Por último, se va a evaluar la normalidad de este atributo. Dado que la mayoría de los métodos que se aplicarán consideran que la distribución de las muestras es normal, conviene comprobar mediante test que efectivamente nuestras variables cumplen esta hipótesis.

Existen dos propiedades que indican si la distribución de una variable se desvía de una normal. Estos son los estadísticos de asimetría y curtosis [4].

Un valor de curtosis y/o coeficiente de asimetría entre -1 y 1, es generalmente considerada una ligera desviación de la normalidad, mientras que entre -2 y 2 hay una evidente desviación de la normal pero no extrema. Los valores que se obtendrán en una normal serían cercanos a 0.

```
Fromm scipy.stats import kurtosis, skew

kurtosis(dataset['Longitud sépalo']) # Curtosis de la longitud del sépalo: -0.573567948924

skew(dataset['Longitud sépalo']) #Asimetría de la longitud del sépalo: 0.31175305850229
```

Se observa cómo se desvía ligeramente de la normal, ya que se aleja de 0.

Como métodos analíticos de normalidad [5], se puede usar el *test* de Shapiro-Wilk, aconsejable para muestras pequeñas (<50) por su sensibilidad a las desviaciones. Para muestras más grandes se puede usar el *test* de Kolmogorov-Smirnov, que permite estudiar en función de su media y desviación típica si una muestra procede de una población con una determinada distribución.

Este segundo *test* no es viable en aquellas muestras en las que la media y varianza poblacional no sean representativas. Por ello, también se añade el *test* de Jarque Bera que no requiere estimaciones de los parámetros que caracterizan la normal.

En todos ellos, se puede rechazar la hipótesis de que la distribución de la variable es normal si el *p-value* es menor que un determinado valor, por lo general alfa=0.05; es decir:

- ***p<=alfa:*** rechazar hipótesis, se asegura que no es normal.
- ***p>alfa:*** no se puede rechazar la hipótesis, se asume normalidad porque es muy probable que sea cierto tras la prueba realizada.

```
from scipy.stats import shapiro, kstest, jarque_bera

# Test de Shapiro-Wilk

stat, p = shapiro(dataset['Longitud sépalo']) # p-value: 0.010180278681218624

# Test de Kolmogorov-Smirnov

stat, p = kstest(dataset['Longitud sépalo']) # p-value: 0.0

# Test de Jarque-Bera

stat, p = jarque_bera(dataset['Longitud sépalo']) # p-value: 0.10614621817187797
```

En este caso todos los dos primeros test indican que la longitud del sépalo no sigue una distribución normal, ya que p-value<0.05. Sin embargo, con el *test* de Jarque Bera se podría afirmar que esto es así.

Se ha de tener en cuenta que la muestra de la población es muy pequeña, y que por tanto puede ser un conjunto que no represente de forma fiel al conjunto real.

Si se reproduce el mismo experimento con una muestra pequeña generada a partir de una distribución normal, en función de la semilla escogida se puede obtener muestras donde los test nos indiquen que se ha de rechazar la hipótesis de normalidad, ya que por fruto del azar podemos tener muestras poco representativas.

Al aumentar el número de instancias de la población se reduce la probabilidad de que esto ocurra, de forma que las conclusiones sobre la normalidad de la muestra serán más robustas.

Análisis multivariable

Aunque se han visto los estadísticos que se pueden aplicar a cada una de nuestras variables, la realidad es que su comportamiento dependerá de la influencia de su entorno.

El objetivo de realizar un análisis bivariante es observar de forma simultánea el comportamiento de dos atributos, o más, si es un análisis multivariante. A continuación, se van a ver distintos análisis y técnicas que se pueden aplicar. La mayoría de ellos pueden extenderse a más de dos variables:

- **Tablas de contingencia.** Permiten crear agrupaciones sobre varias variables.
- **Covarianza / correlación.** Para el análisis de relación lineal bivariante, se pueden usar estos dos métodos para comprobar si existen relaciones entre variables.

Se analizará con Python las funciones que existen para examinar el análisis con más de una variable. En primer lugar, las tablas de contingencia están diseñadas para evaluar valores categóricos. Si se muestran sobre una columna con datos numéricos creará un rango de agrupación para cada uno de los posibles valores, teniendo tantos rangos como evidencias existan.

Por eso, si se quiere aplicarla sobre una columna de este tipo, se deben definir previamente los rangos en los que se quiere que se agrupen dichos valores.

```
x = dataset['Longitud sépalo']

#Definimos rangos entre el min y el max con step=0.5

bins = np.arange(min(x),max(x),0.5)

rangos = pd.cut(dataset['Longitud sépalo'],bins =bins)
#Dividimos en rangos nuestra var

pd.crosstab(rangos,dataset['Clase'])
```

Se observa que la clase Iris-setosa presenta flores de longitud del sépalo pequeños, mientras que para la clase virginica estas son más grandes.

En este caso, queda bien representada la distribución de cada clase en función de esta propiedad, ya que si dan una flor con una longitud de sépalo entre [4.3,4.8] podemos afirmar que es de la clase setosa, por ejemplo.

La mayor dificultad en este caso sería conocer a qué clase pertenece una flor con longitud de sépalo entre [5.5,6], donde se necesitaría más información para poder clasificar correctamente *(figura 6)*.

Por otro lado, la covarianza permite encontrar si existe cierta relación lineal entre variables, es decir, cuanto más aumenta una más aumenta la otra, y viceversa. Por ejemplo, la anchura de la muñeca depende de la altura, de forma que una persona alta tendrá una muñeca de mayor grosor que una persona bajita.

La función "cov()" permite generar fácilmente esta información:

```
dataset.cov() #Muestra la matriz de covarianzas
```

Clase	Iris-setosa	Iris-versicolor	Iris-virginica
Longitud de sépalo			
(4.3, 4.8]	15	0	0
(4.8, 5.3]	24	5	1
(5.3, 5.8]	10	19	5
(5.8, 6.3]	0	15	13
(6.3, 6.8]	0	9	16
(6.8, 7.3]	0	2	8
(7.3, 7.8]	0	0	6

Figura 6. Tabla de contingencia de la clase frente a la longitud del sépalo.

En esta matriz se evidencia como la longitud del pétalo está relacionada con su ancho positivamente, es decir, entre más largo sea el pétalo más ancho será, y viceversa.

También se observa que la relación entre la longitud del pétalo y el ancho del sépalo es inversa, por lo que este tenderá a decrecer conforme aumenta el primero.

Uno de los problemas que presenta esta medida es que su valor depende de las unidades de las propiedades, por lo que tan solo se puede analizar si la relación es positiva o no, ya que definir si están muy relacionadas o no dependerá de sus unidades de medida *(figura 7)*.

Por último, se va a ver la correlación entre variables, que muestra la covarianza si estandarizamos las variables (media 0 y varianza 1). Este estadístico también se conoce como el coeficiente de correlación de Pearson, y puede tomar valores desde -1 a 1, en función del signo de la relación, negativa o positiva.

```
dataset.corr() #Muestra la matriz de covarianzas
```

Se observa como la correlación con la misma variable es 1, el máximo. Además, se mantienen las relaciones que se habían analizado en la matriz de covarianzas, pero ahora se pueden establecer comparaciones, de forma que la relación entre la longitud del pétalo y su ancho es positiva y muy fuerte, mientras que entre el ancho del sépalo y la longitud del sépalo es negativa y débil, casi neutra por ser cercana a 0.0, siendo el punto en el que no existe relación *(figura 8)*.

	Longitud sépalo	Ancho sépalo	Longitud pétalo	Ancho pétalo
Longitud sépalo	0.685694	-0.039268	1.273682	0.516904
Ancho sépalo	-0.039268	0.188004	-0.321713	-0.117981
Longitud pétalo	1.273682	-0.321713	3.113179	1.296387
Ancho pétalo	0.516904	-0.117981	1.296387	0.582414

Figura 7. Matriz de covarianzas del *dataset Iris*.

	Longitud sépalo	Ancho sépalo	Longitud pétalo	Ancho pétalo
Longitud sépalo	1.000000	-0.109369	0.871754	0.817954
Ancho sépalo	-0.109369	1.000000	-0.420516	-0.356544
Longitud pétalo	0.871754	-0420516	1.000000	0.962757
Ancho pétalo	0.817954	-0.356544	0.962757	1.000000

Figura 8. Matriz de correlación del *dataset Iris*.

Bibliografía

[1] G. Westreicher. "Análisis de datos - Economipedia". Economipedia. [En línea] Disponible en: https://economipedia.com/definiciones/analisis-de-datos.html [Accedido el 12 de abril de 2022].

[2] "UCI Machine Learning Repository". Iris Data Set. [En línea] Disponible en: https://archive.ics.uci.edu/dataset/53/iris [Accedido el 12 de abril de 2022].

[3] "UCI Machine Learning Repository". Breast Cancer Data Set. [En línea] Disponible en: https://archive.ics.uci.edu/dataset/14/breast+cancer [Accedido el 12 de abril de 2022].

[4] "Asimetría y kurtosis". Odio la Estadística. [En línea] Disponible en: https://www.odiolaestadistica.com/estadistica-python/asimetria/ [Accedido el 12 de abril de 2022].

[5] J. Amat Rodrigo. "Análisis de normalidad: gráficos y contrastes de hipótesis con R". Ciencia de datos, teoría y ejemplos prácticos en R y Python. [En línea] Disponible en: https://www.cienciadedatos.net/documentos/8_analisis_normalidad [Accedido el 12 de abril de 2022].