TASK 3 (10 min)

A **DataFrame** is a tabular data structure that contains data organized in rows and different columns, very similar to an Excel spreadsheet. The user can perform operations on the data stored in it using different methods (functions) that are available in Python libraries like Pandas, Polars, etc.

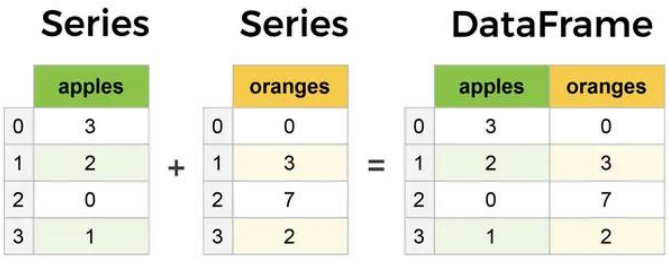
For example, consider an Excel spreadsheet with data from students including Name, Surname, Date of Birth and Score. If we store this spreadsheet as a DataFrame, it would appear like this:

	Name	Surname	Date of bith	Score
0	Ana	García	2021-02-14	8.5
1	Luis	Martínez	2023-09-14	7.0
2	Pedro	Pérez	2021-06-20	9.3

As you can see in the example from above, each row has an index (like a row label) and each column can store a specific type of data (integer, datetime, float, etc.) that is automatically inferred or specified the user.

A **Series** is a column of our DataFrame, that is to say, a one-dimensional array object that can hold data.

Series are the structural units of DataFrames. We can think a DataFrame as a collection of Series arranged in a tabular format where each Series represents a column and shares a common index that identifies the rows.



In other words, a Series can be thought of as a single column of data, while a DataFrame represents the entire table.

TASK 4

Using the formula of the linear combination of two random variables:

$$\mathrm{Var}(aX + bY) = a^2\mathrm{Var}(X) + b^2\mathrm{Var}(Y) + 2ab\mathrm{Cov}(X, Y)$$

In this case:

$$aX + bY = 2X - 3Y$$

Therefore:

$$a = 2$$
$$b = -3$$

Given these variables:

$$Var(X) = 2$$
$$Var(Y) = 3.5$$
$$Cov(X, Y) = -0.8$$

We replace and calculate $Var(2x - 3Y)$

$$Var(Z) = Var(2X - 3Y)$$
$$Var(2X - 3Y) = 2^2 * 2 + (-3)^2 * (3.5) + 2 * 2 * (-3) * (-0.8)$$
$$Var(2X - 3Y) = 8 + 31.5 + 9.6$$
$$Var(2x - 3Y) = 49.1$$

TEXT TEST TASK

PART 1

Utilización del método agg()

Para utilizar el método [data_to_aggregate].agg(), necesitamos pasar un argumento que indique las funciones que queremos aplicar a la Serie “purchase”. Para hacer esto, debemos especificar el nombre de la columna y los nombres de las funciones estructurados en un diccionario.

Recuerda que un diccionario está formado claves y valores. El diccionario de abajo contiene un par clave-valor: 'purchase' es la clave, mientras que 'size' y 'sum' son los valores.

{'purchase': ['size', 'sum']}

Aquí la clave es el nombre de la columna a la que deben aplicarse las funciones y los valores son las funciones que queremos aplicar.

El método agg() (Como lo redactaría yo)

Este método aplica una o más funciones de agregación a los valores de una serie o de un DataFrame. Toma como parámetro de entrada un argumento llamado “func” a través del cual se especifican las columnas de nuestro DataFrame y las operaciones que deseemos realizar sobre cada una de ellas.

El argumento “func” puede ser especificado de diferentes maneras. Una de ellas es por medio de un diccionario, indicando la función o lista de funciones a aplicar a cada variable.

¿Cómo se aplica lo explicado anteriormente?

Es importante recordar que un diccionario es una estructura de datos formada por claves y valores. En este caso, las claves serían los nombres de las columnas y los valores las funciones que necesitemos aplicar.

Supongamos que tenemos el siguiente DataFrame y queremos obtener la cantidad de alumnos y el promedio y la suma de las calificaciones:

	Name	Surname	Date of bith	Score
0	Ana	García	2021-02-14	8.5
1	Luis	Martínez	2023-09-14	7.0
2	Pedro	Pérez	2021-06-20	9.3

Utilizando la columna “Score” (nombre de la variable y clave de nuestro diccionario), podemos contar la cantidad de filas, calcular la media y la suma de las calificaciones (lista de funciones):

```
data.aggregate({
                "Score": ["count","mean", "sum"]
                })
```

Lo que nos devuelve lo siguiente:

	Score
count	3.000000
mean	8.266667
sum	24.800000

Justificación de las modificaciones realizadas

En el texto original no se entiende la finalidad del método agg(), por lo cual es necesario hacer una pequeña introducción explicando este aspecto. De esta manera, el lector va a poder entender cuáles son los elementos que necesita para utilizar esta funcionalidad.

Adicionalmente es importante especificar que el parámetro “func” admite diferentes formatos y que en este caso se aborda uno solo: utilizando un diccionario.

Está bien recordar qué es un diccionario y aplicar el concepto a este caso específico. Sin embargo, en el texto original existen redundancias y también errores en la forma de estructurar el diccionario: se confunden claves con valores.

Finalmente, creo que un ejemplo es adecuado para mostrar un caso real de uso.

PART 2

- **El método de Bonferroni**

Es la corrección más común y más simplificada del nivel de significación requerido. El nivel de significación en cada una de las comparaciones es "m" veces menor que el nivel de significación requerido para una sola comparación. En pocas palabras, el nivel de significación α se divide entre el número de hipótesis:

$$\alpha_1 = \ldots = \alpha_m = \alpha/m.$$

Por ejemplo, si estás probando 10 hipótesis y quieres un α general de 0,05, el nivel de significación de cada prueba será 0,05 / 10 = 0,005.

- **El método de Westfall**

El método de Westfall también garantiza que el error familiar de tipo I (FWER) sea menor que α, pero sus requisitos con respecto al nivel de significación son más moderados. Se trata de un procedimiento secuencial de reducción: para la primera comparación, el nivel de significación requerido es igual a α dividido por el número total de comparaciones, para la segunda comparación es α dividido por el número total de comparaciones menos 1, y así sucesivamente. Para la última comparación, el nivel de significación será igual a α:

$$\alpha_1 = \frac{\alpha}{m},\ \alpha_2 = \frac{\alpha}{m-1},\ \ldots,\ \alpha_i = \frac{\alpha}{m-i+1},\ \ldots,\ \alpha_m = \alpha.$$

- **El método de Fisher**

El procedimiento de Fisher también garantiza que el error familiar de tipo I (FWER) sea menor que α. El valor corregido del nivel de significación requerido se encuentra con la siguiente fórmula:

$$\alpha_1 = \alpha_2 = \ldots = \alpha_m = 1 - (1-\alpha)^{\frac{1}{m}}.$$

Por ejemplo, si α = 0.05 y hay dos comparaciones, el nivel de significación requerido se puede encontrar de la siguiente manera: 1 - (1 - 0.05)^(1/2) = 0.0253. Con cuatro comparaciones será 1 - (1 - 0,05)^(1/4) = 0,0127.

La corrección de Bonferroni es la más común debido a su simplicidad, ya que no resulta difícil dividir el nivel de significación deseado entre el número de comparaciones, las cuales se realizan con los mismos datos sin necesidad de obtener nuevas observaciones para cada prueba. Si recopilas nuevos datos para cada prueba de hipótesis, realiza la prueba de la manera estándar, seleccionando el valor p necesario como lo hiciste en la parte del curso sobre estadística.

Cuanto mayores sean tus requisitos con respecto al nivel de significación, menor será el poder de tu prueba, lo que significa que no verás las diferencias entre los grupos con mayor frecuencia. Entonces, si necesitas aumentar el poder de tu prueba manteniendo FWER < `α`, usa el método de Westfall o Fisher.

(Como lo redactaría yo)

En muchos experimentos, uno de los tratamientos es el control y el investigador está interesado en comparar cada una de las otras K-1 medias de los tratamientos contra el control, por lo cual existen K-1 comparaciones. Cuando se realizan múltiples comparaciones, el riesgo de obtener un resultado falso positivo aumenta. Para controlar este riesgo, se ajusta el nivel de significación (α). Este proceso se llama corrección para comparaciones múltiples o ajuste de α y existen diferentes métodos:

- **El método de Bonferroni**

El Método de Bonferroni es una más simples de ajustar α:

- Con m comparaciones, se divide el nivel de significación α original por m.
- El nivel de significación ajustado para cada comparación es α’ = α/m

Por ejemplo, si estás probando 10 hipótesis y quieres un α general de 0,05, el nivel de significación de cada prueba será 0,05 / 10 = 0,005.

El método de Dunnett, en lugar de simplemente dividir α por el número de comparaciones, utiliza una distribución específica que tiene en cuenta la estructura de las comparaciones múltiples, lo que proporciona un ajuste más preciso y menos conservador que Bonferroni.

- **El método de Westfall**

También garantiza FWER (family-wise error rate) < α. Es una técnica estadística utilizada para ajustar los niveles de significancia en comparaciones múltiples, similar a los métodos de Bonferroni y Dunnett.

Es un procedimiento de reducción: para la primera comparación, el nivel de significación requerido es igual a la relación de α con el número de comparaciones por pares, para la segunda es la relación de α con el número de comparaciones - 1, etc. Para la última comparación, el nivel de significación será igual a α:

$$\alpha_1 = \frac{\alpha}{m},\ \alpha_2 = \frac{\alpha}{m-1},\ \ldots,\ \alpha_i = \frac{\alpha}{m-i+1},\ \ldots,\ \alpha_m = \alpha.$$

- **Método de Fisher**

Tambien garantiza FWER (family-wise error rate) < α. El valor corregido del nivel de significación requerido se encuentra con la siguiente fórmula:

$$\alpha_1 = \alpha_2 = \ldots = \alpha_m = 1 - (1-\alpha)^{\frac{1}{m}}\ .$$

Por ejemplo, si α = 0.05 y hay dos comparaciones, el nivel de significación requerido se puede encontrar de la siguiente manera: 1 - (1 - 0.05)^(1/2) = 0.0253. Con cuatro comparaciones será 1 - (1 - 0,05)^(1/4) = 0,0127.

Cuanto mayores sean los requisitos con respecto al nivel de significación, menor será el poder de la prueba, lo que significa que no se verán las diferencias entre los grupos con mayor frecuencia. Si bien el método de Bonferroni es el más sencillo, si se necesita aumentar el poder de la prueba manteniendo FWER < α, se debe usar el método de Westfall o Fisher.

Justificación de las modificaciones realizadas

Creo que no se explica previamente la finalidad de cada uno de los métodos mencionados, por lo cual he aportado una breve introducción a la temática.

Algunas redacciones no eran lo suficientemente claras y/o coherentes, por lo cual he redactado nuevamente algunas oraciones y ejemplos.

Adicionalmente, el Método de Dunnet que se menciona en primer lugar no coincide con la fórmula de ajuste de α. Esta aproximación se corresponde con el Método de Bonferroni, por lo que sería un error conceptual.

PART 3

INNER JOIN

INNER JOIN selecciona solo los datos que cumplen con la condición de unión.

Aquí tienes un ejemplo de consulta con INNER JOIN:

```
SELECT --indicando solo los campos que se necesitan
   TABLE_1.field_1,
   TABLE_1.field_2,
   ...
   TABLE_2.field_n
FROM
   TABLE_1
   INNER JOIN TABLE_2 ON TABLE_2.field_1 = TABLE_1.field_2;
```

Vamos a echar un vistazo más de cerca a la sintaxis.

- INNER JOIN es el nombre del método de unión. Luego viene el nombre de la tabla que se unirá a la tabla del bloque SELECT.

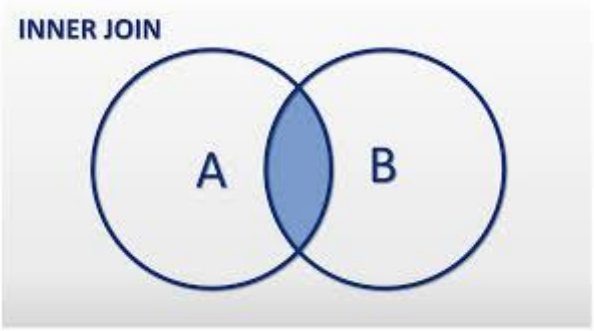
- ON precede a la condición de unión: `TABLE_2.field_1 = TABLE_1.field_2`. Esto significa que solo se unirán las filas de la tabla que cumplan esta condición. En nuestro caso, la condición es que `field_2` de la primera tabla coincida con `field_1` de la segunda.

Dado que los campos en diferentes tablas pueden tener los mismos nombres, se debe hacer referencia a ellos tanto por el nombre de la tabla como por el nombre del campo. Primero va el nombre de la tabla y después el campo: `TABLE_1.field_1`.

(Como lo redactaría yo)

INNER JOIN

Esta sentencia es un método de unión que combina los registros de dos tablas si hay valores coincidentes en un campo común. Esto quiere decir que el conjunto de resultado obtenido se corresponde con la intersección de ambas tablas.



Veamos un ejemplo:

```
SELECT *
FROM
  TABLE_1
  INNER JOIN TABLE_A ON TABLE_A.field_1 = TABLE_B.field_2;
```

En este caso, la consulta une la tabla A y la tabla B utilizando el campo field_1 de la tabla A y field_2 de la tabla B. El * luego del SELECT indica que se van a seleccionar todas las columnas de ambas tablas.

¿Por qué sería necesario unir dos tablas?

Se trata de una práctica habitual cuando es necesario enriquecer nuestros datos utilizando otros existentes en otra tabla de nuestra base de datos.

Por ejemplo, supongamos que tenemos una tabla con los datos personales de los alumnos y otra tabla con las calificaciones. Cada alumno puede tener un identificador único (ID) y, de esta manera, podremos hacer un INNER JOIN de ambas tablas utilizando este ID.

Una práctica adecuada para que la consulta no nos devuelva todos los campos de ambas tablas consiste en especificar en el SELECT únicamente las columnas que son necesarias. Debido a que ambas tablas pueden

compartir columnas con el mismo nombre, siempre es necesario especificar la tabla de la cual deseamos extraer la columna. De otra manera habrá un error de sintaxis.

Justificación de las modificaciones realizadas

El texto original no presenta de forma clara y visual la sentencia INNER JOIN. Considero que debería incluir un diagrama para representar de manera gráfica lo que pretende obtener al utilizarla.

Hay un error en la frase que alerta sobre el orden de las tablas y el resultado a obtener, ya que el INNER JOIN es conmutativo. También hay un error en la explicación de la sintaxis, en relación a los campos de cada una de las tablas.

En el ejemplo indicado no se explica de manera clara lo que se busca en la consulta. Por este motivo he agregado una explicación junto con un ejemplo de un caso real.

La aclaración del final es válida, pero hay que aclarar que forma parte de las buenas practicas.