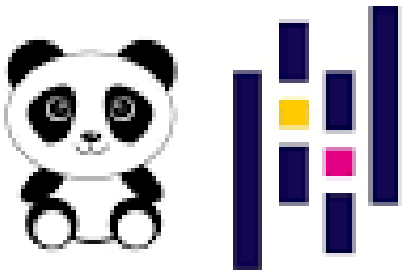
FAMILIARIZACIÓN CON UN DATASET



Asignatura: Acceso a Datos

Autores: Pablo Rodríguez Peña
Pablo Rodríguez Segura

Centro: IES Ramón Valle Inclán

Curso: 2024/25

ÍNDICE

Pablo Rodríguez Peña
Pablo Rodríguez Segura

Apartado 1

Hemos elegido un dataset relacionado con personas con obesidad , esta es la URL https://archive.ics.uci.edu/dataset/544/estimation+of+obesity+levels+based+on+eating+habits+and+physical+condition

En el conjunto de datos presentado, las columnas representan características específicas de cada individuo y, en conjunto, describen los factores relacionados con su salud, estilo de vida y peso. A continuación, se explica cada columna:

- Gender: Género del individuo (Male o Female).
- Age: Edad del individuo en años.
- Height: Altura del individuo en metros.
- Weight: Peso del individuo en kilogramos.
- family_history_with_overweight: Indica si existe historial familiar de sobrepeso (yes o no).
- FAVC (Frequent consumption of high-calorie food): Indica si el individuo consume alimentos de alta densidad calórica con frecuencia (yes o no).
- FCVC (Frequency of consumption of vegetables): Frecuencia con la que consume vegetales (1: Baja, 2: Media, 3: Alta).
- NCP (Number of main meals): Número de comidas principales que realiza al día.
- CAEC (Consumption of food between meals): Indica la frecuencia de consumo de alimentos entre comidas (no, Sometimes, Frequently, Always).
- SMOKE: Indica si el individuo fuma (yes o no).
- CH2O (Consumption of water daily): Cantidad aproximada de agua consumida diariamente en litros (1: Baja, 2: Media, 3: Alta).
- SCC (Monitoring of calorie consumption): Indica si el individuo monitorea su ingesta calórica (yes o no).
- FAF (Physical activity frequency): Frecuencia de actividad física semanal (0: Ninguna, 1: Baja, 2: Media, 3: Alta).
- TUE (Time using technology devices): Tiempo promedio diario utilizando dispositivos electrónicos en horas.

Pablo Rodríguez Peña
Pablo Rodríguez Segura

- CALC (Alcohol consumption): Frecuencia de consumo de alcohol (no, Sometimes, Frequently, Always).
- MTRANS (Transportation): Medio de transporte principal utilizado (Automobile, Walking, Public_Transportation, Motorbike, etc.).
- NObeyesdad: Clasificación del estado de peso según el índice de masa corporal (BMI):
- Insufficient_Weight: Peso insuficiente.
- Normal_Weight: Peso normal.
- Overweight_Level_I: Sobrepeso nivel 1.
- Overweight_Level_II: Sobrepeso nivel 2.
- Obesity_Type_I: Obesidad tipo I.
- Obesity_Type_II: Obesidad tipo II.
- Obesity_Type_III: Obesidad tipo III.

Este conjunto de datos es útil para analizar patrones relacionados con la salud, identificar factores de riesgo asociados con la obesidad y explorar las correlaciones entre las características individuales y los resultados relacionados con el peso.

Apartado 2

Cargamos el dataset elegido -> resultados.csv

```
df = pd.read_csv("resultados.csv")
print(df)
```

Exploramos el dataframe:

SHAPE

```
Forma del DataFrame (filas, columnas):
(2111, 17)
```


Pablo Rodríguez Peña
Pablo Rodríguez Segura

HEAD

```
Primeras 5 filas del DataFrame:
   Gender   Age  Height  Weight  ...  TUE        CALC                MTRANS           NObeyesdad
0  Female  21.0    1.62    64.0  ...  1.0          no  Public_Transportation       Normal_Weight
1  Female  21.0    1.52    56.0  ...  0.0   Sometimes  Public_Transportation       Normal_Weight
2    Male  23.0    1.80    77.0  ...  1.0  Frequently  Public_Transportation       Normal_Weight
3    Male  27.0    1.80    87.0  ...  0.0  Frequently               Walking  Overweight_Level_I
4    Male  22.0    1.78    89.8  ...  0.0   Sometimes  Public_Transportation  Overweight_Level_II

[5 rows x 17 columns]
```

INFO

```
Información general del DataFrame:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2111 entries, 0 to 2110
Data columns (total 17 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Gender                          2111 non-null   object
 1   Age                             2111 non-null   float64
 2   Height                          2111 non-null   float64
 3   Weight                          2111 non-null   float64
 4   family_history_with_overweight  2111 non-null   object
 5   FAVC                            2111 non-null   object
 6   FCVC                            2111 non-null   float64
 7   NCP                             2111 non-null   float64
 8   CAEC                            2111 non-null   object
 9   SMOKE                           2111 non-null   object
 10  CH2O                            2111 non-null   float64
 11  SCC                             2111 non-null   object
 12  FAF                             2111 non-null   float64
```

```
 13  TUE                             2111 non-null   float64
 14  CALC                            2111 non-null   object
 15  MTRANS                          2111 non-null   object
 16  NObeyesdad                      2111 non-null   object
dtypes: float64(8), object(9)
memory usage: 280.5+ KB
None
```

DESCRIBE

Código:

print("\nResumen estadístico de las columnas numéricas:")

print(df.describe())

Pablo Rodríguez Peña
Pablo Rodríguez Segura

Resultado:

```
Resumen estadístico de las columnas numéricas:
              Age       Height       Weight         FCVC          NCP         CH2O          FAF
   TUE
count  2111.000000  2111.000000  2111.000000  2111.000000  2111.000000  2111.000000  2111.000000  2
000000
mean     24.312600     1.701677    86.586058     2.419043     2.685628     2.008011     1.010298
657866
std       6.345968     0.093305    26.191172     0.533927     0.778039     0.612953     0.850592
608927
min      14.000000     1.450000    39.000000     1.000000     1.000000     1.000000     0.000000
000000
25%      19.947192     1.630000    65.473343     2.000000     2.658738     1.584812     0.124505
000000
50%      22.777890     1.700499    83.000000     2.385502     3.000000     2.000000     1.000000
625350
75%      26.000000     1.768464   107.430682     3.000000     3.000000     2.477420     1.666678
000000
max      61.000000     1.980000   173.000000     3.000000     4.000000     3.000000     3.000000
000000
```

Comentario sobre la columna Age

La columna Age representa la edad de los individuos en el dataset. A partir del resumen estadístico, podemos analizar e interpretar la distribución de estos valores:

Resumen estadístico de Age:

Número de registros (Count): Hay 2111 registros no nulos en la columna Age, lo que significa que no hay valores faltantes.

Edad promedio (Media): La edad promedio es de 24.31 años, lo que indica que, en promedio, los individuos del dataset son jóvenes adultos.

Desviación estándar (Std): La desviación estándar es 6.35 años, lo que indica que hay una variabilidad moderada en las edades. Esto sugiere que la mayoría de los individuos se encuentran cerca de la media, pero también hay una cantidad significativa de personas más jóvenes o mayores.

Edad mínima (Min): La edad mínima es 14 años, lo que implica que el dataset incluye personas bastante jóvenes, posiblemente adolescentes.

Percentiles:

El percentil 25 es de 19.95 años, lo que significa que el 25% de los individuos tiene 19.95 años o menos. Esto nos muestra que una parte considerable de los individuos son bastante jóvenes.

La mediana o percentil 50 es de 22.78 años, lo que indica que la mitad de los individuos tiene 22.78 años o menos, lo que refuerza que la mayoría son adultos jóvenes.

El percentil 75 es de 26 años, lo que significa que el 75% de los individuos tiene 26 años o menos.

Edad máxima (Max): La edad máxima es de 61 años, lo que indica que aunque la mayoría de los individuos son jóvenes, también hay personas significativamente mayores dentro del dataset.

Apartado 3

Mostraremos las columnas de Género , Edad , Medio de Transporte usado y Frecuencia de actividad física semanal

Código:

```
# Seleccionamos 4 columnas representativas
columnas_representativas = ["Gender", "Age", "MTRANS", "FAF"]
df_seleccion = df[columnas_representativas]

# Mostramos las primeras 5 filas de las columnas seleccionadas
print("\n")
print("Selección de 4 columnas representativas (5 primeras filas):")
print(df_seleccion.head())
```

Resultado:

```
Selección de 4 columnas representativas (5 primeras filas):
   Gender   Age                MTRANS  FAF
0  Female  21.0  Public_Transportation  0.0
1  Female  21.0  Public_Transportation  3.0
2    Male  23.0  Public_Transportation  2.0
3    Male  27.0               Walking  2.0
4    Male  22.0  Public_Transportation  0.0
```

Apartado 4

Hemos decidido agregar esta columna:

water_intake_per_meal (Consumo de agua por comida)

Este atributo se calcula combinando el consumo diario de agua (CH2O) y el número de comidas principales al día (NCP)

Refleja la cantidad promedio de agua que el individuo consume en cada comida principal.

Código:

```
# Añadir una nueva columna 'water_intake_per_meal' con la ingesta de agua por comida
df["water_intake_per_meal"] = df["CH2O"] / df["NCP"]
```


Pablo Rodríguez Peña
Pablo Rodríguez Segura

```
nuevas_columnas_representativas = [
    "Gender",
    "Age",
    "MTRANS",
    "FAF",
    "water_intake_per_meal",
]
df_seleccion_nueva_columna = df[nuevas_columnas_representativas]

print("\n")
print("Mostramos las columnas rerpesentativas + la nueva columna")
print(df_seleccion_nueva_columna.head())
```

Resultado:

```
 Mostramos las columnas rerpesentativas + la nueva columna
    Gender   Age                MTRANS  FAF  water_intake_per_meal
 0  Female  21.0  Public_Transportation  0.0               0.666667
 1  Female  21.0  Public_Transportation  3.0               1.000000
 2    Male  23.0  Public_Transportation  2.0               0.666667
 3    Male  27.0               Walking  2.0               0.666667
 4    Male  22.0  Public_Transportation  0.0               2.000000
```

Rangos para análisis:

Bajo: ≤ 0.33 L/comida

Moderado: 0.34 - 0.66 L/comida

Alto: > 0.66 L/comida

Apartado 5

Mostramos las columnas de Género, Edad, Altura y Peso de las personas que fuman y beben frecuentemente

Código:

```
# Mostramos las filas donde el paciente es fumador y bebe frecuentemente
columnas_seleccion = [
    "Gender",
    "Age",
    "Height",
    "Weight",
]

fumadores_bebedores = df[(df["SMOKE"] == "yes") & (df["CALC"] == "Frequently")]
print("\nPersonas fumadoras y que consumen alchol de manera frecuente ")
print(fumadores_bebedores[columnas_seleccion])
```

Resultado:

Pablo Rodríguez Peña
Pablo Rodríguez Segura

```
Personas fumadoras y que consumen alchol de manera frecuente
     Gender   Age  Height  Weight
43     Male  21.0    1.66    62.0
68     Male  30.0    1.76   112.0
132  Female  19.0    1.65    56.0
142    Male  23.0    1.74    93.5
178    Male  26.0    1.91    84.0
205  Female  23.0    1.60    78.0
252    Male  56.0    1.79    90.0
```

Apartado 6

Primero insertamos una fila de valores nulos, ya que nuestro csv no tiene

```
# Crear un DataFrame con una fila de valores None
nueva_fila_nula = pd.DataFrame([{col: None for col in df.columns}])

# Concatenar la nueva fila al DataFrame original
df = pd.concat([df, nueva_fila_nula], ignore_index=True)
```

Imprimimos los todas las filas que contengan al menos un valor nulo

```
# Verificar y mostrar las filas con valores nulos
valores_nulos = df[df.isnull().any(axis=1)]

# Imprimir las filas con valores nulos
print(valores_nulos)
```

Salida:

```
Filas con valores nulos:
     Gender  Age  Height  Weight family_history_with_overweight  ... TUE  CALC  MTRANS NObeyesdad water_intake_per_meal
2111   None  NaN     NaN     NaN                           None  ... NaN  None    None       None                    NaN

[1 rows x 18 columns]
```

df[df.isnull().any(axis=1)]: Filtra el DataFrame para mostrar solo las filas que contienen al menos un valor nulo.

isnull() es un método de Pandas que devuelve un nuevo DataFrame de la misma forma que df, pero en lugar de los valores originales, este nuevo DataFrame contiene True donde hay valores nulos (es decir, donde hay NaN o valores ausentes) y False donde los valores no son nulos.

Pablo Rodríguez Peña
Pablo Rodríguez Segura

axis=1 especifica que se debe operar a lo largo de las columnas (horizontalmente). Esto significa que para cada fila, la función any() verifica si hay algún valor True (es decir, algún valor nulo en esa fila).
any() devuelve True si al menos uno de los valores en la fila es True (es decir, si hay un valor nulo en alguna de las columnas de esa fila). Si todos los valores de la fila son False (es decir, no hay valores nulos), entonces devuelve False.

iterrows(): Este método paccederermite iterar sobre las filas del DataFrame. Para cada fila, devuelve un índice y la fila misma.

row.items(): Permite acceder a cada columna y su valor en la fila.
pd.isnull(value): Verifica si el valor es nulo (NaN), y si es así, lo imprime junto con el nombre de la columna.

-Número de filas antes y después

```
# Contamos el número de filas de nuestro csv
print("Numero de registros del csv")
print(df.shape[0])
```

```
Numero de registros del csv: 2112

Numero de registros despues de eliminar las filas con valores nulos: 2111
```

Apartado 7

Creamos algunos valores nulos para poder rellenarlos
Mostramos las 6 primeras filas y observamos algunos campos que son nulos:

```
6 primeras filas, antes de rellenar valores nulos
   Gender   Age  Height  Weight family_history_with_overweight FAVC  FCVC  NCP  ... CH2O  SCC  FAF  TUE        CALC                MTRANS           NObeyesdad water_intake_per_meal
0  Female  21.0    1.00   64.0                            yes   no   2.0  3.0  ...  2.0   no  0.0  1.0          no                   NaN       Normal_Weight              0.666667
1  Female  21.0    1.52   56.0                            yes   no   3.0  3.0  ...  3.0  yes  3.0  0.0   Sometimes  Public_Transportation       Normal_Weight              1.000000
2    Male   NaN    1.80   77.0                            yes   no   2.0  3.0  ...  2.0   no  2.0  1.0  Frequently  Public_Transportation       Normal_Weight              0.666667
3    Male  27.0    1.80   87.0                             no   no   3.0  3.0  ...  2.0   no  2.0  0.0         NaN               Walking  Overweight_Level_I              0.666667
4    Male   NaN    1.78   89.8                             no   no   2.0  1.0  ...  2.0   no  0.0  0.0   Sometimes  Public_Transportation  Overweight_Level_II              2.000000
5    Male  29.0    1.62   53.0                             no  yes   2.0  3.0  ...  2.0   no  0.0  0.0   Sometimes            Automobile       Normal_Weight              0.666667
```

Sustituimos los valores nulos por “Nulo rellenado”

```
6 primeras filas, después de rellenar valores nulos
   Gender            Age  Height Weight family_history_with_overweight FAVC  FCVC  ...  SCC  FAF  TUE            CALC                MTRANS           NObeyesdad  water_intake_per_me
0  Female           21.0    1.00   64.0                            yes   no   2.0  ...   no  0.0  1.0              no         Nulo rellenado       Normal_Weight              0.6666
1  Female           21.0    1.52   56.0                            yes   no   3.0  ...  yes  3.0  0.0       Sometimes  Public_Transportation       Normal_Weight              1.0000
2    Male  Nulo rellenado    1.80   77.0                            yes   no   2.0  ...   no  2.0  1.0      Frequently  Public_Transportation       Normal_Weight              0.6666
3    Male           27.0    1.80   87.0                             no   no   3.0  ...   no  2.0  0.0  Nulo rellenado               Walking  Overweight_Level_I              0.6666
4    Male  Nulo rellenado    1.78   89.8                             no   no   2.0  ...   no  0.0  0.0       Sometimes  Public_Transportation  Overweight_Level_II              2.0000
5    Male           29.0    1.62   53.0                             no  yes   2.0  ...   no  0.0  0.0       Sometimes            Automobile       Normal_Weight              0.6666
```


Pablo Rodríguez Peña
Pablo Rodríguez Segura

Apartado 8

Hacemos una agrupación doble por consumo de alcohol y por clasificación de estado de peso, por lo que podemos ver la relación del consumo de alcohol y el peso de la persona

```
# Agrupamos por frecuencia de consumo de alcohol ("CALC") y tipo de obesidad ("OBESITY")
agrupacion = (
    df.groupby(["CALC", "NObeyesdad"]).size().reset_index(name="Numero de Personas")
)
```

Salida:

```
Número de personas según la frecuencia de consumo de alcohol y tipo de obesidad:
          CALC           NObeyesdad  Numero de Personas
0       Always        Normal_Weight                   1
1   Frequently  Insufficient_Weight                   1
2   Frequently        Normal_Weight                  18
3   Frequently       Obesity_Type_I                  14
4   Frequently      Obesity_Type_II                   2
5   Frequently    Overweight_Level_I                  16
6   Frequently   Overweight_Level_II                  19
7    Sometimes  Insufficient_Weight                 154
8    Sometimes        Normal_Weight                 161
9    Sometimes       Obesity_Type_I                 172
10   Sometimes      Obesity_Type_II                 224
11   Sometimes     Obesity_Type_III                 323
12   Sometimes    Overweight_Level_I                 224
13   Sometimes   Overweight_Level_II                 143
14          no  Insufficient_Weight                 117
15          no        Normal_Weight                 107
16          no       Obesity_Type_I                 165
17          no      Obesity_Type_II                  71
18          no     Obesity_Type_III                   1
19          no    Overweight_Level_I                  50
20          no   Overweight_Level_II                 128
```

Además, hacemos el promedio de peso por consumo de alimentos entre comidas (CAEC)

```
# Promedio de peso por consumo de alimentos entre comidas (CAEC)
promedio_peso_por_caec = df.groupby("CAEC")[["Weight"]].mean()
print("\nPromedio de peso por consumo de alimentos entre comidas (CAEC):")
print(promedio_peso_por_caec)
```

Salida:

```
Promedio de peso por consumo de alimentos entre comidas (CAEC):
              Weight
CAEC
Always     71.090566
Frequently 58.885678
Sometimes  91.360344
no         68.902489
```


Pablo Rodríguez Peña
Pablo Rodríguez Segura

Anexo

```python
import pandas as pd


# EJERCICIO 1

# Mostramos el dataset
df = pd.read_csv("resultados.csv")
print(df)

# EJERCICIO 2

# Exploración básica del DataFrame
print("Forma del DataFrame (filas, columnas):")
print(df.shape)

print("\nPrimeras 5 filas del DataFrame:")
print(df.head())

print("\nInformación general del DataFrame:")
print(df.info())

print("\nResumen estadístico de las columnas numéricas:")
print(df.describe())


# EJERCICIO 3

# Seleccionamos 4 columnas representativas
columnas_representativas = ["Gender", "Age", "MTRANS", "FAF"]
df_seleccion = df[columnas_representativas]

# Mostramos las primeras 5 filas de las columnas seleccionadas
print("\n")
print("Selección de 4 columnas representativas (5 primeras filas):")
print(df_seleccion.head())


# EJERICIO 4
```


Pablo Rodríguez Peña
Pablo Rodríguez Segura

```
# Añadir una nueva columna 'water_intake_per_meal' con la ingesta de
agua por comida
df["water_intake_per_meal"] = df["CH2O"] / df["NCP"]

nuevas_columnas_representativas = [
    "Gender",
    "Age",
    "MTRANS",
    "FAF",
    "water_intake_per_meal",
]
df_seleccion_nueva_columna = df[nuevas_columnas_representativas]

print("\n")
print("Mostramos las columnas rerpesentativas + la nueva columna")
print(df_seleccion_nueva_columna.head())


# EJERCICIO 5

# Mostramos las filas donde el paciente es fumador y bebe
frecuentemente
columnas_seleccion = [
    "Gender",
    "Age",
    "Height",
    "Weight",
]

fumadores_bebedores = df[(df["SMOKE"] == "yes") & (df["CALC"] ==
"Frequently")]
print("\nPersonas fumadoras y que consumen alchol de manera frecuente
")
print(fumadores_bebedores[columnas_seleccion])


# Crear un DataFrame con una fila de valores None
nueva_fila_nula = pd.DataFrame([{col: None for col in df.columns}])

# Concatenar la nueva fila al DataFrame original
df = pd.concat([df, nueva_fila_nula], ignore_index=True)
```


Pablo Rodríguez Peña
Pablo Rodríguez Segura

```
# Verificar y mostrar las filas con valores nulos
valores_nulos = df[df.isnull().any(axis=1)]
print("\nFilas con valores nulos:")
print(valores_nulos)

# Para cada fila con valores nulos, mostrar solo las columnas que son
nulas
# for index, row in valores_nulos.iterrows():
#     print(f"Fila {index}:")
#     for column, value in row.items():
#         if pd.isnull(value):
#             print(f"  - {column} es nulo")


# Contamos el número de filas de nuestro csv
print(f"\nNumero de registros del csv: {df.shape[0]}")


# Borramos filas con valores nulos y mostramos el número luego de
eliminarlas
df_limpio = df.dropna()  # Crear un nuevo DataFrame sin valores nulos
print(f"\nNumero de registros despues de eliminar las filas con valores
nulos: {df_limpio.shape[0]}")


# Agrupamos por frecuencia de consumo de alcohol ("CALC") y tipo de
obesidad ("OBESITY")
agrupacion = (
   df.groupby(["CALC", "NObeyesdad"]).size().reset_index(name="Numero
de Personas")
)

# Mostrar resultados
print("\nNúmero de personas según la frecuencia de consumo de alcohol y
tipo de obesidad:")
print(agrupacion)

# Promedio de peso por consumo de alimentos entre comidas (CAEC)
promedio_peso_por_caec = df.groupby("CAEC")[["Weight"]].mean()
print("\nPromedio de peso por consumo de alimentos entre comidas
(CAEC):")
print(promedio_peso_por_caec)
```


Pablo Rodríguez Peña
Pablo Rodríguez Segura