M2_AI3_Obesidad

Autor: Leandro Gutierrez

Este documento intenta dar respuesta a la actividad individual 3 propuesta en el Modulo **Fundamentos de Estadística** del **Master en Big Data y Ciencia de Datos**. En él se describirán cada uno de los enunciados postulados y los resultados obtenidos a través del uso de Python y Google Colab.

Junio 28, 2024

Enunciado

Con los datos facilitados sobre el total de personas diagnosticadas con obesidad en España y sobre el total de población detallado a nivel de provincia, te pedimos que halles:

- ¿Cuál es la probabilidad de padecer obesidad en España?. Se debe aportar su fórmula y su sustitución. Analizar los resultados obtenidos (15%)
- Si seleccionamos una de las personas obesas al azar, ¿Cuál es la probabilidad de que resida en Barcelona? ¿Y en el resto de provincias? Se debe aportar su fórmula y su sustitución. (70%)
- Aportar el árbol de decisión, añadir una explicación de cómo funciona, el resultado y un comentario o análisis del resultado obtenido. (15%)

Consideraciones

- Fuente de la información.
- Los datos de obesidad están basadas en diferentes fuentes, como la Encuesta Nacional de Salud (ENSANUT) y el Estudio Sobre Nutrición, Actividad Física y Salud (ENAS).
- Los datos de población son estimaciones del INE a fecha de 1 de enero de 2023.

Solución

Diccionario de datos

Variable	Definition	Key
Fecha	Identificador de toma de información.	
Provincia	Provincia que identifica lugar geográfico de la observación.	

Total Obesos	Total de habitantes en la provincia con obesidad al momento de la observación
Total Habitantes	Total de habitantes en la provincia al momento de la observación

Carga del dataset

In [16]:
```python
import requests
import pandas as pd

url = 'https://raw.githubusercontent.com/leandrogutierrez148/master/main/M2/

# token de acceso a Github personal
token = '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'

# headers para el request
headers = {
    "Authorization": f"token {token}"
}

# realizamos el request
response = requests.get(url, headers=headers)

# obtenemos los datos
with open('archivo.xlsx', 'wb') as file:
    file.write(response.content)

    # leemos el archivo descargado con pandas y openpyxl
    try:
        df_org = pd.read_excel('archivo.xlsx', engine='openpyxl')
    except Exception as e:
        print("Error al leer el archivo Excel: ", e)

df_org.head(20)
```

Out[16]:

	Fecha	Provincia	Total Obesos	Total Habitantes
0	2023-01-01	Alava	42345	323897
1	2023-01-01	Albacete	146234	398567
2	2023-01-01	Alicante	368423	1949789
3	2023-01-01	Almería	240567	702345
4	2023-01-01	Asturias	278234	1019897
5	2023-01-01	Ávila	48345	162567
6	2023-01-01	Badajoz	201234	677897
7	2023-01-01	Baleares	322567	1115234
8	2023-01-01	Barcelona	712897	5539567
9	2023-01-01	Bizkaia	238345	1154789
10	2023-01-01	Burgos	84234	365897
11	2023-01-01	Cáceres	132567	400234
12	2023-01-01	Cádiz	245897	1254567
13	2023-01-01	Cantabria	159345	582678
14	2023-01-01	Castellón	144789	510567
15	2023-01-01	Ciudad Real	162423	502987
16	2023-01-01	Córdoba	212789	745345
17	2023-01-01	Coruña	287423	1145678
18	2023-01-01	Cuenca	54567	202345
19	2023-01-01	Girona	102897	766789

Análisis del dataset

Navegaremos en el dataset para ver su estructura, esquema, completitud y los conjuntos de valores de las variables.

In [3]:
```
# copiamos dataframe para no alterar original
df_aux = df_org.copy()
```

In [4]:
```
# resumen información del dataset
print(df_aux.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 4 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Fecha             50 non-null     datetime64[ns]
 1   Provincia         50 non-null     object
 2   Total Obesos      50 non-null     int64
 3   Total Habitantes  50 non-null     int64
dtypes: datetime64[ns](1), int64(2), object(1)
memory usage: 1.7+ KB
None
```

Podemos observar que la variable **Fecha** fue interpretada por pandas como **datetime64**, mientras que la variable **Provincia** como **object** y a su vez **Total Obesos** y **Total Habitantes** como **int64**

In [5]:
```
# controlamos forma del dataset
df_aux.shape
```

Out[5]:
```
(50, 4)
```

Contamos con 50 observaciones y 4 variables.

Completitud

In [6]:
```
# controlamos valores nulos en los datos
df_aux.isna().sum()
```

Out[6]:
```
Fecha              0
Provincia          0
Total Obesos       0
Total Habitantes   0
dtype: int64
```

No se registran valores nulos en el dataset.

Conjunto de valores

In [7]:
```
# hacemos un describe preeliminar de cada variable
print(df_aux['Fecha'].describe())
print(df_aux['Provincia'].describe())
print(df_aux['Total Obesos'].describe())
print(df_aux['Total Habitantes'].describe())
```

```
count                     50
mean     2023-01-01 00:00:00
min      2023-01-01 00:00:00
25%      2023-01-01 00:00:00
50%      2023-01-01 00:00:00
75%      2023-01-01 00:00:00
max      2023-01-01 00:00:00
Name: Fecha, dtype: object
count        50
unique       50
top       Alava
freq          1
Name: Provincia, dtype: object
count        50.000000
mean     183134.640000
std      170495.555137
min       11345.000000
25%       75364.500000
50%      138678.000000
75%      238345.000000
max      873423.000000
Name: Total Obesos, dtype: float64
count    5.000000e+01
mean     9.588493e+05
std      1.324569e+06
min      8.253500e+04
25%      3.228995e+05
50%      6.099560e+05
75%      1.008620e+06
max      6.775897e+06
Name: Total Habitantes, dtype: float64
```

Análisis de los tipos de variables

1. Fecha: variable alfanumérica que indica el dia de la toma de la observación. La definiremos como cualitativa ordinal, ya que nos puede sirvir para agrupar y ordenar nuestras observaciones. Podemos observar que solo se presenta el valor **2023-01-01**. Es una variable **cualitativa ordinal**.
2. Provincia: variable alfanumerica que indica el lugar geográfico donde se tomó la observación. La definiremos como cualitativa nominal, ya que nos puede servir para agrupar y clasificar nuestras observaciones pero no cuenta con un orden o secuencia específica. Es una variable **cualitatica nominal**.
3. Total Obesos: variable numérica entera la cual indica la cantidad de habitantes con obesidad en la provincia donde se realizó la observación al momento de indicado en `Fecha`. Vamos a considerar la variable **cuantitativa discreta** ya que al hablamos de individuos.
4. Total Habitantes: variable numérica entera la cual indica la cantidad total de habitantes en la provincia donde se realizó la observación al momento de indicado en `Fecha`. Vamos a considerar la variable **cuantitativa discreta** ya que al hablamos de individuos.

Apartado 1

Dado el Experimento aleatorio E el cual determina si una persona en España padece o no obesidad, cuyo Espacio muestral asociado es:

$$S = \{(p_1, p_2, \ldots, p_i, \ldots, p_n) \mid p_i \in \{Obeso, NoObeso\}, \forall i \in \{1, 2, \ldots, n\}\}$$

Podemos obtener la probabilidad del suceso **Padecer Obesidad en España** a través de la frecuencia relativa del suceso, esto es:

$$f_r(Obeso) = \frac{n_{Obeso}}{n}$$

Donde n_{Obeso} es el numero total de individuos que padecen obesidad y n es el numero total de individuos.

Ambos datos que podemos obtener a partir del dataset analizado. Considerando que hay registro de las 50 Provincias Españolas.

Encontremos la cantidad de casos que se verifica el suceso **Obesidad** y el total de repeticiones del experimento aleatorio:

In [8]:
```
# calculamos el total de casos positivos
total_obesos = df_aux['Total Obesos'].sum()

# calculamos el total de experimentos
total_hab = df_aux['Total Habitantes'].sum()
```

```
print(f'Total Obesos en España: {total_obesos}')
print(f'Total Habitatntes en España: {total_hab}')
```

```
Total Obesos en España: 9156732
Total Habitatntes en España: 47942464
```

Obtenemos una frecuencia relativa igual a:

$$f_r(Obeso) = \frac{9156732}{47942464}$$

In [9]:
```
frec_rel = total_obesos/total_hab

print(f'Frecuencia relativa de Obeso: {frec_rel}')
```

```
Frecuencia relativa de Obeso: 0.19099418836712273
```

Por lo tanto la probabilidad de **Padecer Obesidad en España** es del **19.09%**.

[ANALIZAR RESULTADOS]

Apartado 2

Dado el experimento aleatorio E' el cual determina en que provincia vive una persona residente en España y si es o no obeso, definido por el espacio muestral:

$$S' = \{(ALAV, Obeso), (ALAV, NoObeso), \ldots, (BCN, Obeso), (BCN, NoObeso), .$$

Dados los sucesos:

- Ser obeso (Obeso)
- Residir en Barcelona (BCN)

Para encontrar la probabilidad de que un individuo **Resida en Barcelona** si se sabe que es **Obeso** vamos a utilizar la probabilidad condicional, de tal manera que

$$P(BCN|_{Obeso}) = \frac{P(BCN \cap Obeso)}{P(Obeso)}$$

In [10]:
```
bcn_obeso = df_aux[df_aux['Provincia'] == 'Barcelona']['Total Obesos']
bcn_obeso
```

Out[10]:
```
8    712897
Name: Total Obesos, dtype: int64
```

$$P(BCN|_{Obeso}) = \frac{712897}{9156732}$$

In [11]:
```
cond_bcn_obeso = bcn_obeso / total_obesos
cond_bcn_obeso
```

Out[11]:
```
8    0.077855
Name: Total Obesos, dtype: float64
```

La probabilidad de que se cumpla la condición **Residir en Barceolna** habiendo seleccionado azarozamente un individuo calificado como **Obeso** es del **7.78%**

Extendamos el análisis para el resto de provincias.

Agregaremos a nuestro Dataframe la variable *Provincia|Obeso* la cual indica la probabilidad de ser de la provicia siendo que se sabe se ha seleccionado aleatoriamente un individuo con **Obesidad**.

In [14]:
```
df_aux['Provincia|Obeso'] = df_aux['Total Obesos']/total_obesos
```

In [17]:
```
df_aux.head(20)
```

Out[17]:

	Fecha	Provincia	Total Obesos	Total Habitantes	Provincia|Obeso
0	2023-01-01	Alava	42345	323897	0.004624
1	2023-01-01	Albacete	146234	398567	0.015970
2	2023-01-01	Alicante	368423	1949789	0.040235
3	2023-01-01	Almería	240567	702345	0.026272
4	2023-01-01	Asturias	278234	1019897	0.030386
5	2023-01-01	Ávila	48345	162567	0.005280
6	2023-01-01	Badajoz	201234	677897	0.021977
7	2023-01-01	Baleares	322567	1115234	0.035227
8	2023-01-01	Barcelona	712897	5539567	0.077855
9	2023-01-01	Bizkaia	238345	1154789	0.026029
10	2023-01-01	Burgos	84234	365897	0.009199
11	2023-01-01	Cáceres	132567	400234	0.014478
12	2023-01-01	Cádiz	245897	1254567	0.026854
13	2023-01-01	Cantabria	159345	582678	0.017402
14	2023-01-01	Castellón	144789	510567	0.015812
15	2023-01-01	Ciudad Real	162423	502987	0.017738
16	2023-01-01	Córdoba	212789	745345	0.023239
17	2023-01-01	Coruña	287423	1145678	0.031389
18	2023-01-01	Cuenca	54567	202345	0.005959
19	2023-01-01	Girona	102897	766789	0.011237

[ANALIZAR RESULTADOS]