

2. Selección un dataset tabular de al menos 1000 columnas, 14 filas. Si elige imágenes igualmente puede convertir la imagen en datos tabulares de NxM.

De esta selección indique cual es la clase o si no tiene.

1. Elementos Clave del Dataset

- **Elemento Clasificador:** **Rating of the Class Experience** (Evaluación de la experiencia en clases en línea). Esta columna clasifica la experiencia de los estudiantes en clases virtuales con valores como "Muy pobre", "Buena", etc., lo que la convierte en una variable categórica que puede servir como objetivo en un análisis supervisado.
- **Tipo de Dataset:**
 - **Supervisado (posible):** Sin embargo, si definimos una variable como objetivo (por ejemplo, "Rating of Online Class experience"), el dataset podría utilizarse para un análisis supervisado, como la predicción de la satisfacción estudiantil basada en otros factores (tiempo en redes sociales, tiempo de estudio, etc.).

2. Descripción de las Columnas y su Importancia:

A continuación, se describen las columnas y su relevancia en el análisis:

1. **ID:** Identificador único para cada registro. No tiene valor analítico directo, pero es útil para identificar cada observación individualmente.
2. **Region of Residence (Región de Residencia):** Importante para analizar diferencias geográficas en el acceso a la educación y otras variables. Podría ser un buen predictor de desigualdades en el acceso a tecnología o variaciones en las experiencias de los estudiantes.
3. **Age of Subject (Edad del Sujeto):** Relevante para identificar cómo diferentes grupos etarios se enfrentaron a la educación en línea. Podría ser útil para segmentar a los estudiantes y analizar variaciones en sus hábitos y actitudes.
4. **Time spent on Online Class (Tiempo Dedicado a Clases en Línea):** Indicador clave para medir el compromiso académico y la disponibilidad de los estudiantes para las clases en línea. Este dato es crucial para correlacionar con el rendimiento académico o el bienestar general.
5. **Rating of Online Class Experience (Evaluación de la Experiencia en Clases en Línea):** Columna crítica para medir la satisfacción estudiantil. Podría usarse como variable objetivo en un análisis supervisado para predecir la satisfacción en función de otras variables como el tiempo de estudio o la conectividad.
6. **Medium for Online Class (Medio para Clases en Línea):** Importante para entender las desigualdades tecnológicas. La disponibilidad de dispositivos

adecuados para la educación en línea podría influir directamente en la calidad percibida de la experiencia educativa.

7. **Time spent on self-study (Tiempo Dedicado al Estudio Independiente):** Variable esencial para evaluar el nivel de autonomía y motivación de los estudiantes. Puede correlacionarse con el éxito académico o la calidad de la experiencia en clases en línea.
8. **Time spent on fitness (Tiempo Dedicado al Ejercicio Físico):** Esta variable tiene relevancia en el bienestar físico y mental de los estudiantes. El equilibrio entre el estudio y las actividades físicas podría influir en su salud general durante la pandemia.
9. **Time spent on sleep (Horas de Sueño):** El sueño es un factor clave en el bienestar mental y físico. Esta variable es crítica para entender cómo la pandemia y el confinamiento afectaron los hábitos de descanso de los estudiantes.
10. **Time spent on social media (Tiempo en Redes Sociales):** Importante para medir el impacto de las redes sociales en la vida diaria de los estudiantes. Un mayor uso de redes sociales podría estar asociado a una disminución en la productividad o el bienestar emocional.
11. **Preferred social media platform (Plataforma de Redes Sociales Preferida):** Esta variable permite analizar las preferencias de los estudiantes y su comportamiento digital. Puede ser útil para segmentar a los estudiantes en función de sus plataformas favoritas.
12. **Time spent on TV (Tiempo dedicado a la TV):** El tiempo en medios tradicionales como la TV podría estar correlacionado con el tiempo en redes sociales o con el uso del tiempo en general durante la pandemia.
13. **Number of meals per day (Número de Comidas por Día):** Refleja los hábitos alimenticios de los estudiantes. Este dato puede tener implicaciones en la salud física y mental, así como en la energía para las actividades académicas.
14. **Change in your weight (Cambio en el Peso):** Indica cómo la pandemia afectó físicamente a los estudiantes. Los cambios en el peso podrían ser el resultado de alteraciones en los hábitos alimenticios y de ejercicio.
15. **Health issue during lockdown (Problemas de Salud durante el Confinamiento):** Relevante para evaluar cómo el confinamiento impactó en la salud general de los estudiantes. Esto puede estar relacionado con factores como el tiempo en actividad física, el tiempo en redes sociales o el sueño.
16. **Stress busters (Actividades Relajantes):** Muestra las estrategias que utilizaron los estudiantes para gestionar el estrés. Es útil para entender cómo los estudiantes lidiaron con la presión emocional del confinamiento.
17. **Time utilized (Tiempo Utilizado):** Indicador de si los estudiantes sintieron que hicieron un uso productivo del tiempo durante el confinamiento. Este dato puede relacionarse con la motivación y los resultados académicos.

18. **Connected with family and friends (Conexión con Familia y Amigos):** Esta variable revela si los estudiantes se sintieron más conectados emocionalmente con sus seres queridos. Es importante para entender el impacto social y emocional del confinamiento.
19. **What you miss the most (Lo que Más Extrañaron):** Proporciona una visión de los aspectos de la vida que los estudiantes valoraban más y que se vieron interrumpidos por la pandemia. Es útil para comprender el impacto emocional de la pandemia.

Complemente con lo siguiente:

- a. Sin el uso de librerías en Python programe el percentil y cuartil de cada columna. Que distribución se puede aplicar en su caso normal, Bernoulli, gaussiana, poisson, otros. Indique la razón de su uso graficando con matplotlib.

Código

```
import csv
import matplotlib.pyplot as plt

# Función para calcular percentil
def calcular_percentil(datos, percentil):
    datos_ordenados = sorted(datos)
    indice = (len(datos_ordenados) - 1) * percentil / 100
    inferior = int(indice)
    superior = inferior + 1
    if superior >= len(datos_ordenados):
        return datos_ordenados[inferior]
    else:
        peso_superior = indice - inferior
        return datos_ordenados[inferior] * (1 - peso_superior) +
datos_ordenados[superior] * peso_superior

# Función para calcular cuartiles
def calcular_cuartiles(datos):
    return calcular_percentil(datos, 25), calcular_percentil(datos,
50), calcular_percentil(datos, 75)

# Leer el archivo CSV
with open('/content/Drive/MyDrive/datos/examencovid354.csv',
```

```

newline='') as archivo_csv:
    lector = csv.reader(archivo_csv)
    columnas = next(lector) # Nombres de columnas
    datos_por_columna = {columna: [] for columna in columnas}

    for fila in lector:
        for i, valor in enumerate(fila):
            try:
                datos_por_columna[columnas[i]].append(float(valor))
            except ValueError:
                continue # Ignorar valores no numéricos

# Calcular percentiles y cuartiles para cada columna
percentiles = {}
cuartiles = {}
for columna, datos in datos_por_columna.items():
    if datos:
        percentiles[columna] = {p: calcular_percentil(datos, p) for p
in range(0, 101, 25)} # Percentiles 0, 25, 50, 75, 100
        cuartiles[columna] = calcular_cuartiles(datos)
        print(f"Columna {columna}: Percentiles: {percentiles[columna]},
Cuartiles: {cuartiles[columna]}")

# Graficar los datos
for columna, datos in datos_por_columna.items():
    if datos:
        plt.figure()
        plt.hist(datos, bins=20, density=True, alpha=0.6, color='g',
label='Histograma')
        plt.axvline(x=cuartiles[columna][0], color='r', linestyle='--',
label='Q1')
        plt.axvline(x=cuartiles[columna][1], color='b', linestyle='-',
label='Mediana (Q2)')
        plt.axvline(x=cuartiles[columna][2], color='r', linestyle='--',
label='Q3')
        plt.title(f"Distribución de {columna}")
        plt.xlabel(columna)
        plt.ylabel("Frecuencia")
        plt.legend()
        plt.show()

# Propuestas de distribuciones

```

```


for columna, datos in datos_por_columna.items():
    if datos:
        media = sum(datos) / len(datos)
        print(f"Propuesta de distribución para {columna}:")

        if all(x == 0 or x == 1 for x in datos): # Datos binarios
            print(f"- Bernoulli, basada en valores binarios (0/1).")
        elif all(isinstance(x, int) and x >= 0 for x in datos): #
Datos discretos
            print(f"- Poisson o Binomial, si se trata de conteos.")
        else: # Datos continuos
            print(f"- Distribución Normal o Gaussiana, si los datos
parecen seguir una curva de campana.")

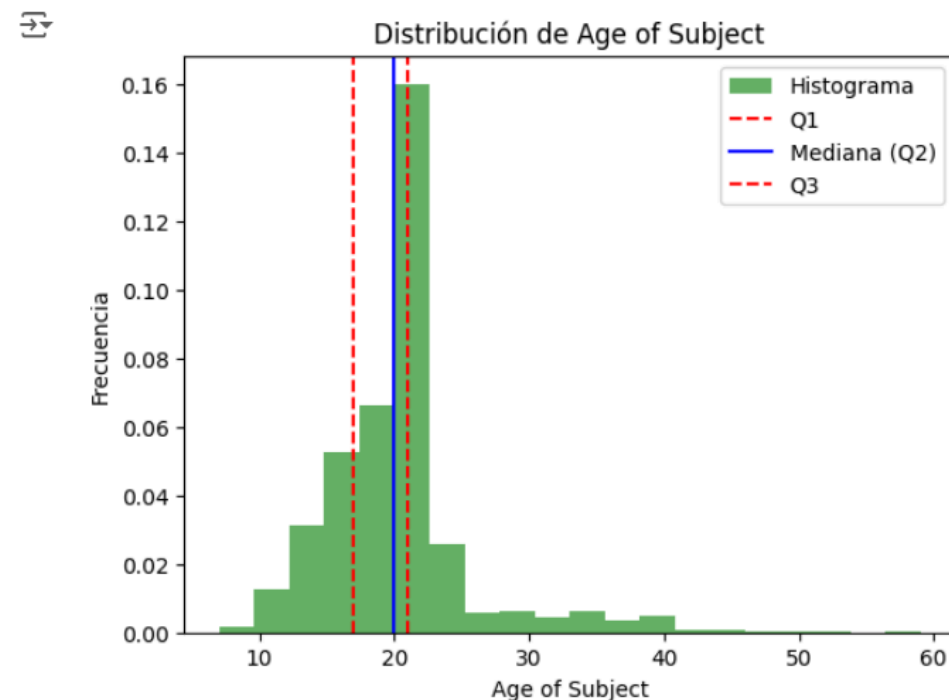
```

Corrida

Se muestran los percentil y cuartil:

 Columna Age of Subject: Percentiles: {0: 7.0, 25: 17.0, 50: 20.0, 75: 21.0, 100: 59.0}, Cuartiles: (17.0, 20.0, 21.0)
 Columna Time spent on Online Class: Percentiles: {0: 0.0, 25: 2.0, 50: 3.0, 75: 5.0, 100: 10.0}, Cuartiles: (2.0, 3.0, 5.0)
 Columna Time spent on self study: Percentiles: {0: 0.0, 25: 2.0, 50: 2.0, 75: 4.0, 100: 18.0}, Cuartiles: (2.0, 2.0, 4.0)
 Columna Time spent on fitness: Percentiles: {0: 0.0, 25: 0.0, 50: 1.0, 75: 1.0, 100: 5.0}, Cuartiles: (0.0, 1.0, 1.0)
 Columna Time spent on sleep: Percentiles: {0: 4.0, 25: 7.0, 50: 8.0, 75: 9.0, 100: 15.0}, Cuartiles: (7.0, 8.0, 9.0)
 Columna Time spent on social media: Percentiles: {0: 0.0, 25: 1.0, 50: 2.0, 75: 3.0, 100: 10.0}, Cuartiles: (1.0, 2.0, 3.0)
 Columna Time spent on TV: Percentiles: {0: 0.0, 25: 0.0, 50: 1.0, 75: 2.0, 100: 15.0}, Cuartiles: (0.0, 1.0, 2.0)
 Columna Number of meals per day: Percentiles: {0: 1.0, 25: 2.0, 50: 3.0, 75: 3.0, 100: 8.0}, Cuartiles: (2.0, 3.0, 3.0)

Graficamente:



b. De al menos tres columnas seleccionadas por usted indique que datos son relevantes de estas, grafique la misma (puede ser dispersión o mapa de calor, otros), indique al menos 4 características por columna seleccionada.

Código

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Cargar el archivo CSV
data = pd.read_csv("/content/Drive/MyDrive/datos/examencovid354.csv")

# Selección de las columnas
df_seleccionadas = data[['Age of Subject', 'Time spent on self
study', 'Time spent on sleep']]

# 1. Estadísticas descriptivas de las columnas seleccionadas
print("Estadísticas descriptivas:")
print(df_seleccionadas.describe())

# 2. Graficar la relación entre las columnas seleccionadas

# Mapa de calor para ver la correlación entre las tres columnas
plt.figure(figsize=(8, 6))
sns.heatmap(df_seleccionadas.corr(), annot=True, cmap='coolwarm')
plt.title('Mapa de calor de la correlación entre Edad del sujeto,
Tiempo dedicado al autoestudio y Tiempo dedicado al sueño')
plt.show()

# Gráfico de dispersión entre Edad del sujeto y Tiempo dedicado al
autoestudio
plt.figure(figsize=(8, 6))
plt.scatter(df_seleccionadas['Age of Subject'],
df_seleccionadas['Time spent on self study'], color='blue',
alpha=0.5)
plt.title('Relación entre Edad y Tiempo dedicado al autoestudio')
plt.xlabel('Edad')
plt.ylabel('Tiempo dedicado al autoestudio (Time spent on self
study)')
plt.show()
```

```

# Gráfico de dispersión entre Edad del sujeto y Tiempo dedicado al
sueño
plt.figure(figsize=(8, 6))
plt.scatter(df_seleccionadas['Age of Subject'],
df_seleccionadas['Time spent on sleep'], color='green', alpha=0.5)
plt.title('Relación entre Edad del sujeto y Tiempo dedicado al
sueño')
plt.xlabel('Edad')
plt.ylabel('Tiempo dedicado al sueño (Time spent on sleep)')
plt.show()

# Gráfico de dispersión entre Tiempo dedicado al sueño y Tiempo
dedicado al autoestudio
plt.figure(figsize=(8, 6))
plt.scatter(df_seleccionadas['Time spent on sleep'],
df_seleccionadas['Time spent on self study'], color='red', alpha=0.5)
plt.title('Relación entre Tiempo dedicado al sueño y Tiempo dedicado
al autoestudio')
plt.xlabel('Tiempo dedicado al sueño (Time spent on sleep)')
plt.ylabel('Tiempo dedicado al autoestudio (Time spent on self
study)')
plt.show()

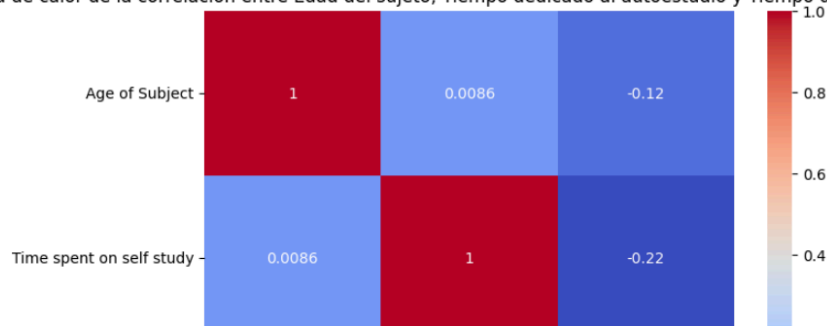
```

Ejecución

Estadísticas descriptivas:

	Age of Subject	Time spent on self study	Time spent on sleep
count	1182.000000	1182.000000	1182.000000
mean	20.165821	2.911591	7.871235
std	5.516467	2.140590	1.615762
min	7.000000	0.000000	4.000000
25%	17.000000	2.000000	7.000000
50%	20.000000	2.000000	8.000000
75%	21.000000	4.000000	9.000000
max	59.000000	18.000000	15.000000

Mapa de calor de la correlación entre Edad del sujeto, Tiempo dedicado al autoestudio y Tiempo dedicado al sueño



c. Obteniendo la media, mediana, moda con el uso de librerías, grafique un diagrama de cajas-bigote de al menos 3 columnas. Explique el resultado.

Codigo

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)

# Cargar el archivo CSV
data = pd.read_csv("/content/Drive/MyDrive/datos/examencovid354.csv")

# Seleccionar las columnas de interés
columnas_seleccionadas = ['Age of Subject', 'Time spent on self
study', 'Time spent on sleep']
df = data[columnas_seleccionadas]

# 1. Obtener la media, mediana, y moda de las columnas seleccionadas
estadisticas = {}
for col in columnas_seleccionadas:
    media = df[col].mean()
    mediana = df[col].median()
    moda = df[col].mode().iloc[0] # Asegúrate de usar iloc para
obtener el primer valor de la moda
    estadisticas[col] = {'media': media, 'mediana': mediana, 'moda':
moda}

    print(f"Para la columna {col}:")
    print(f"Media: {media}")
    print(f"Mediana: {mediana}")
    print(f"Moda: {moda}")
    print()

# 2. Graficar diagrama de cajas y bigotes (boxplot)
plt.figure(figsize=(12, 7))
sns.boxplot(data=df, palette="Set2") # Añadir una paleta para hacer
el gráfico más colorido

# Añadir líneas para la media, mediana y moda
for i, col in enumerate(columnas_seleccionadas):
```



```

media = estadisticas[col]['media']
mediana = estadisticas[col]['mediana']
moda = estadisticas[col]['moda']

plt.axhline(y=media, color='red', linestyle='--', label='Media'
if i == 0 else "")
plt.axhline(y=mediana, color='blue', linestyle='-',
label='Mediana' if i == 0 else "")
plt.axhline(y=moda, color='green', linestyle=':', label='Moda' if
i == 0 else "")

# Personalización del gráfico
plt.title('Diagrama de Cajas y Bigotes para Edad del sujeto, Tiempo
dedicado al autoestudio y Tiempo dedicado al sueño', fontsize=16)
plt.ylabel('Valores', fontsize=14)
plt.xticks(fontsize=12)
plt.grid(axis='y', linestyle='--', alpha=0.7) # Añadir una
cuadrícula solo en el eje y
plt.legend(title='Estadísticas', fontsize=12)
plt.tight_layout() # Ajustar automáticamente el espacio
plt.show()

```

Corrida

Para la columna Age of Subject:
Media: 20.165820642978005
Mediana: 20.0
Moda: 20

Para la columna Time spent on self study:
Media: 2.911590524534687
Mediana: 2.0
Moda: 2.0

Para la columna Time spent on sleep:
Media: 7.871235194585448
Mediana: 8.0
Moda: 8.0

Diagrama de Cajas y Bigotes para Edad del sujeto, Tiempo dedicado al autoestudio y Tiempo dedicado al sueño

