Visualización de datos

La visualización de datos en estadística es crucial para el análisis exploratorio, ya que permite identificar patrones, tendencias y anomalías de manera rápida y efectiva. A través de gráficas, los datos complejos se vuelven más comprensibles, facilitando la toma de decisiones informadas y la detección de relaciones ocultas entre variables.

Elegir el tipo adecuado de gráfica para el análisis exploratorio o la narración de una historia con datos no es un proceso sencillo. Es fundamental comprender el propósito de cada visualización y qué tipo de información aporta cada gráfica. El fin de cualquier visualización es comunicar la información de manera más clara, intuitiva y atractiva.

Por ello, es crucial seleccionar correctamente el tipo de gráfica, ya que las visualizaciones permiten identificar patrones que los números por sí solos no revelan.

En términos generales, las visualizaciones se clasifican en cuatro grandes categorías, como se detalla a continuación:

Correlación: Gráficas de dispersión, mapas de calor.

Distribución: Histogramas, mapas, Gráfico de cuantiles.

Comparación: Gráficas de barras, líneas, cajas y bigotes.

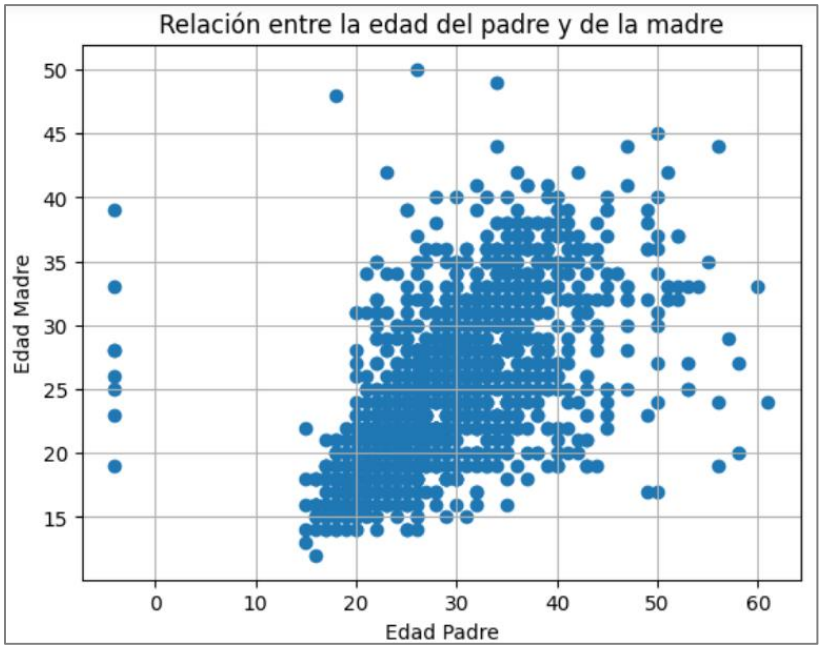
Composición: Gráficas de área, barras apiladas.

Gráficos de correlación

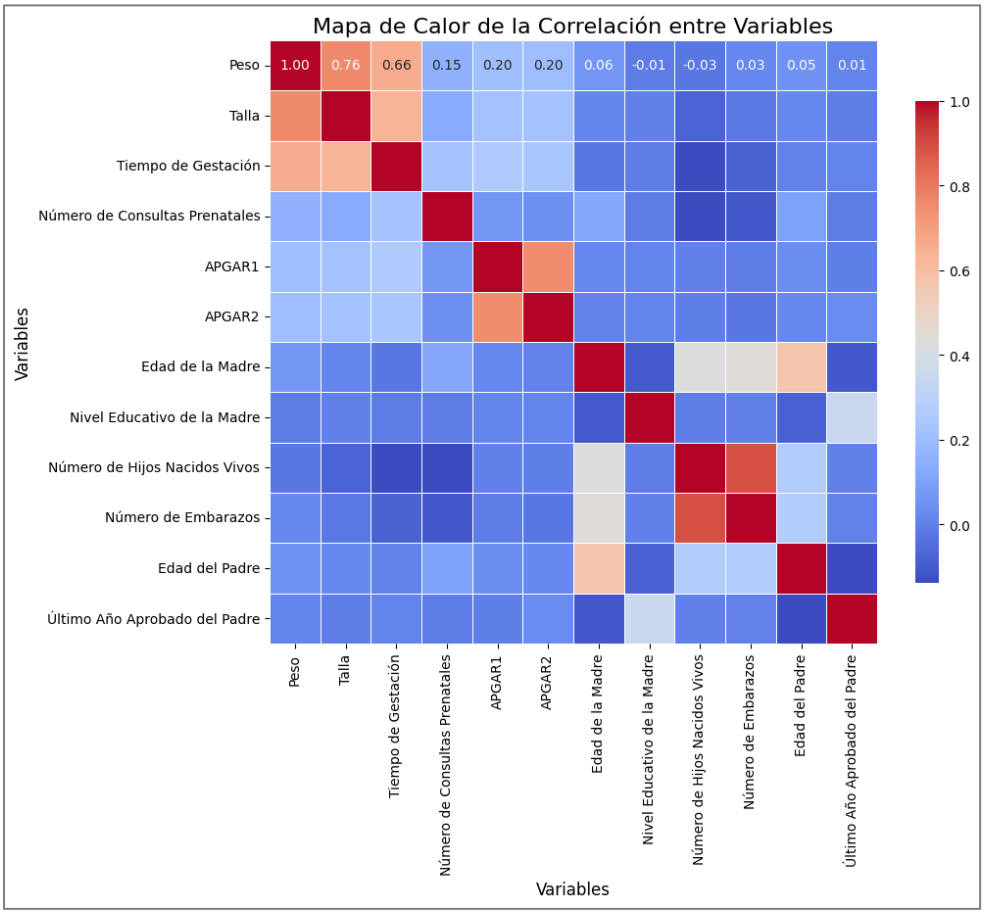
Este grupo incluye las visualizaciones de dispersión y mapas de calor. Las gráficas de correlación muestran si hay relación entre dos o más variables. Sin embargo, es importante recordar que correlación no significa causalidad; es decir, que una variable (x) influya en una variable (y) es decir, no siempre implica que una cause cambios en la otra. Por eso, los resultados deben estar respaldados por investigaciones y experimentos sólidos.

Gráficos de Dispersión: Los gráficos de dispersión permiten visualizar la relación entre dos atributos numéricos y observar la correlación entre los ejes X y Y. Son útiles para comparar medidas, identificar patrones, tendencias, agrupamientos y detectar valores atípicos. Además, permiten analizar el grado de correlación entre las variables. Este tipo de gráfico es especialmente valioso en análisis estadísticos como la regresión lineal y logística.

En este tipo de gráficos pueden usarse 2 o más variables, según las que se empleen para agrupar por colores o marcadores. Añadir colores y líneas de referencia ayuda a mejorar el análisis exploratorio y facilita la identificación de valores atípicos.



Gráficos de Mapas de Calor: Los mapas de calor permiten comparar tres variables: dos categóricas en los ejes X y Y, y una numérica representada con un degradado de color. Esto facilita la visualización de la correlación entre pares de atributos. Son frecuentemente usados para mostrar la matriz de correlación de forma visual.

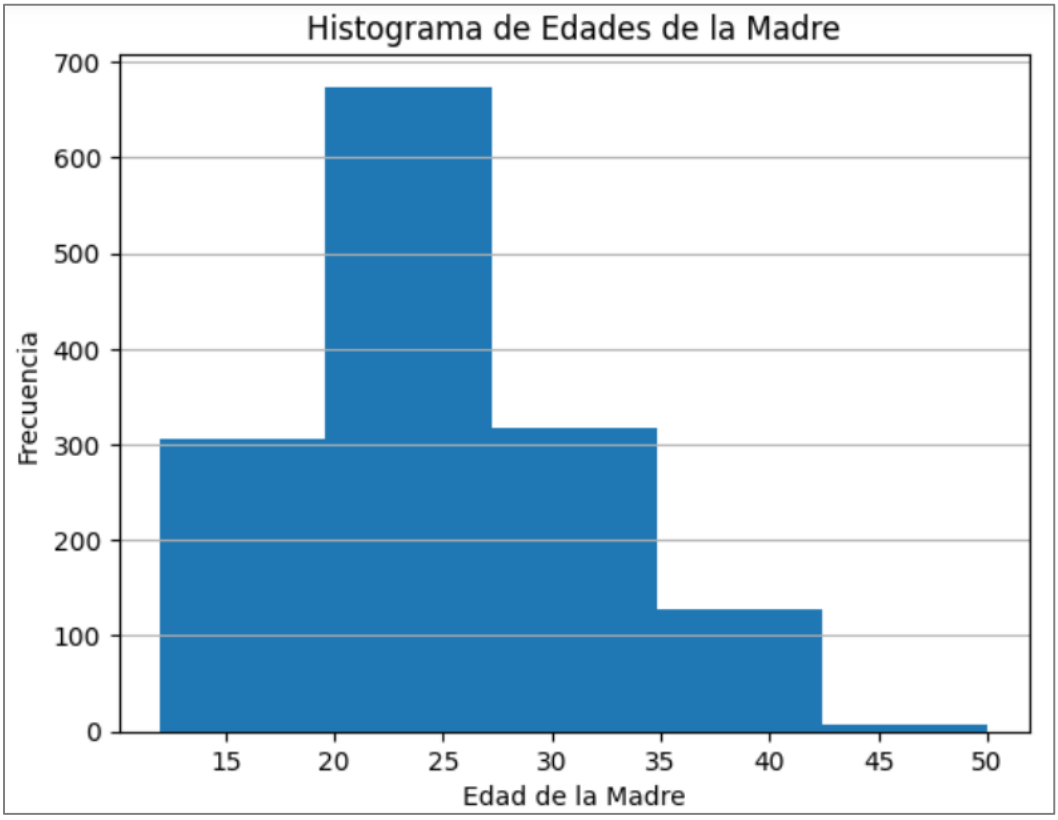


Gráficos de distribución

Este grupo incluye histogramas y mapas. Las gráficas de distribución son útiles en el análisis univariado (una sola variable), especialmente en las etapas iniciales de la analítica, ya que muestran dónde los datos son más densos o escasos. Se emplean frecuentemente en investigaciones de mercado, análisis demográficos y segmentación de clientes.

Histogramas: Organizan los datos en grupos o rangos, llamados contenedores o "bins", y muestran cuántos datos hay en cada grupo. Esto ayuda a ver cómo se distribuyen los datos, mostrando qué valores son más comunes y si los datos están equilibrados o inclinados hacia un lado. También pueden mostrar los resultados como porcentajes para una mejor comprensión.

Los contenedores o "bins" en un histograma son rangos de valores que agrupan los datos. Por ejemplo, si tienes una lista de edades, un contenedor podría incluir las edades de 10 a 20 años, otro de 21 a 30 años, y así sucesivamente. Cada contenedor cuenta cuántos datos (personas, en este caso) caen dentro de ese rango. Esto ayuda a visualizar la frecuencia de los datos en diferentes intervalos.



Gráficas de mapas: las gráficas de mapas representan datos geoespaciales combinados con información demográfica, mostrando la ubicación de los puntos de datos, como casos de COVID-19, clientes o peatones. Los valores numéricos se organizan según un atributo geográfico, como región, país o continente. Las

variaciones en color o tamaño de los puntos reflejan diferencias en la densidad de los datos entre diferentes áreas. Estos mapas son útiles para responder preguntas sobre aspectos espaciales, como el ingreso per cápita por país o la cantidad de clientes o ventas por ciudad.

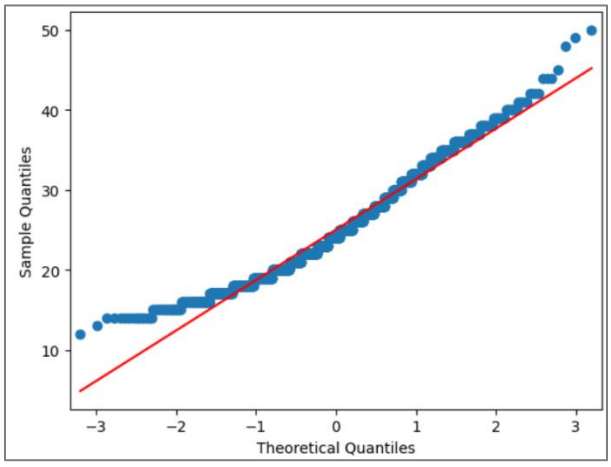
Gráfico de cuantiles: Un gráfico de cuantiles, también conocido como diagrama Q-Q (Quantile-Quantile), es una herramienta visual que compara la distribución de dos conjuntos de datos. Se utiliza principalmente para evaluar si un conjunto de datos sigue una distribución teórica específica, como la normal.

¿Por qué son importantes?

Comparación de distribuciones: Nos permite comparar visualmente la forma de dos distribuciones.

Evaluación de normalidad: Es una forma rápida de verificar si un conjunto de datos se ajusta a una distribución normal.

Identificación de outliers: Los puntos que se desvían significativamente de la línea diagonal en un gráfico Q-Q pueden indicar valores atípicos.

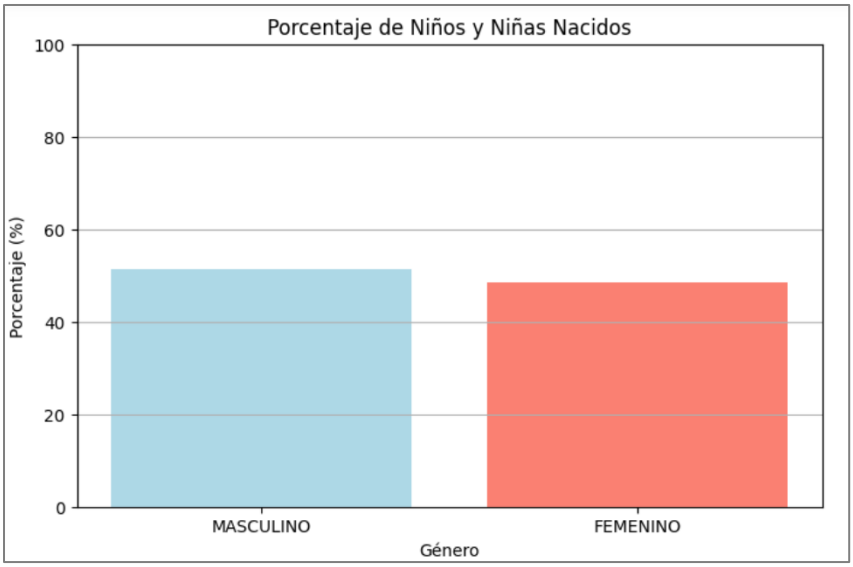


Gráficas De Comparación

Las gráficas de comparación se emplean para analizar y contrastar uno o más conjuntos de datos. Son útiles para examinar variaciones entre elementos o para ilustrar cambios a lo largo del tiempo. Entre este tipo de gráficas se incluyen los gráficos de barras, gráficos de líneas y los diagramas de cajas y bigotes.

Gráficas de barras:

Un gráfico de barras es una representación visual de datos en la que las categorías o grupos se muestran mediante barras rectangulares de diferentes alturas o longitudes. Este tipo de gráfico es útil para comparar cantidades entre diferentes grupos o categorías.



Partes de un Gráfico de Barras:

Ejes (X y Y):

Eje X (Horizontal): Muestra las categorías o grupos que se están comparando (por ejemplo, tipos de productos, regiones, etc.).

Eje Y (Vertical): Representa la cantidad o frecuencia asociada a cada categoría (por ejemplo, ventas, número de personas, etc.).

Estadística para Machine Learning - Actividad No 3

Barras:

Barras Verticales u Horizontales: Las barras se extienden desde una base en el eje hasta la altura o longitud que corresponde al valor de la categoría. La altura o longitud de la barra es proporcional a la cantidad que representa.

Título:

Describe de manera breve y clara el propósito del gráfico o lo que se está representando.

Etiquetas de Ejes:

Indican qué representa cada eje y las unidades de medida utilizadas en el eje Y.

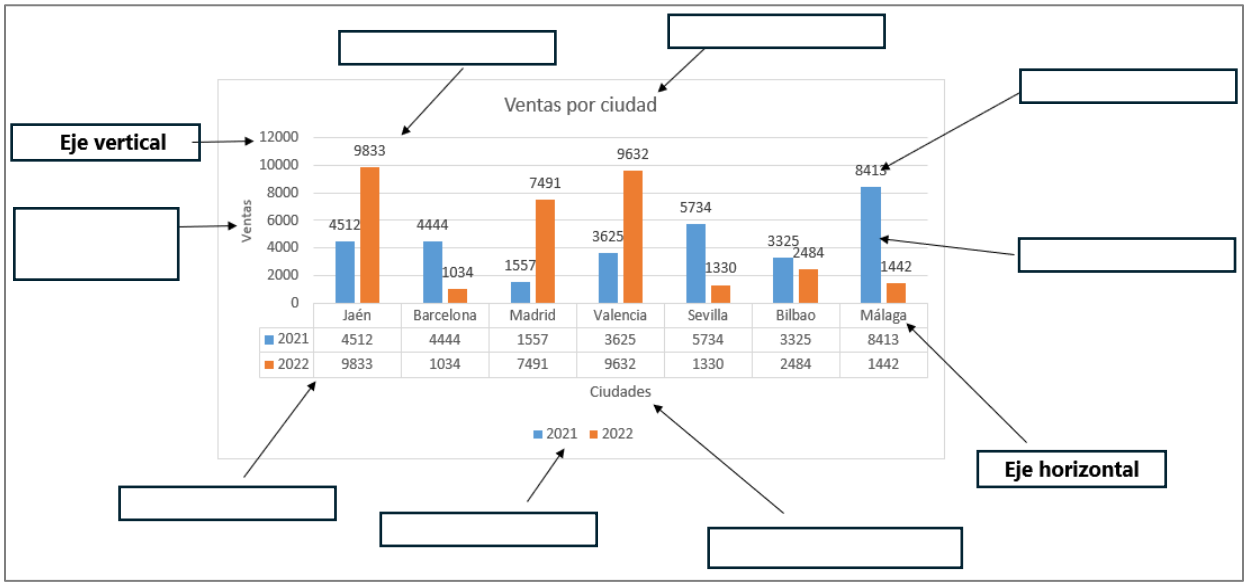
Escala:

Las marcas en el eje Y que muestran los valores numéricos y ayudan a interpretar la altura o longitud de las barras.

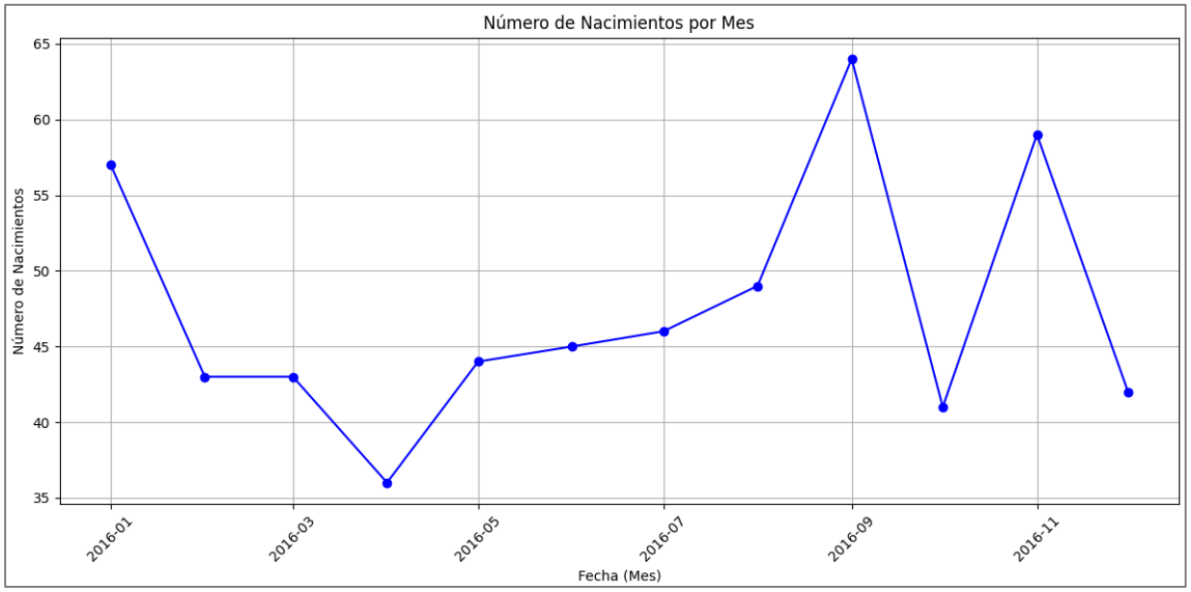
Leyenda (opcional):

Si el gráfico tiene más de una serie de datos (por ejemplo, barras apiladas o barras agrupadas), la leyenda explica lo que representa cada color o estilo de barra.

- ✓ Coloca el nombre de las partes del gráfico de barras.



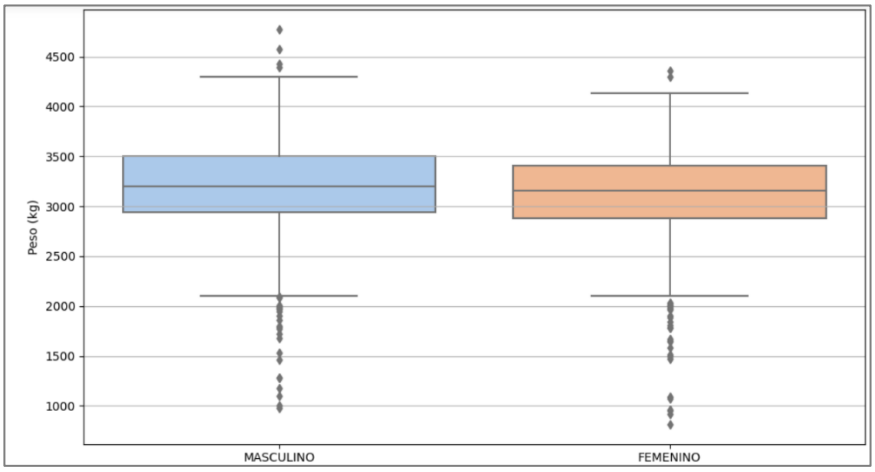
Diagramas de línea: son herramientas visuales que muestran cómo cambian los datos a lo largo del tiempo, permitiendo observar tendencias y variaciones. Son especialmente útiles en series de tiempo, donde se representan cambios en una variable numérica en relación con fechas específicas. Cada línea puede representar una comparación entre diferentes momentos históricos y, además, se pueden incluir atributos categóricos usando diferentes colores para resaltar las diferencias entre categorías. En resumen, las gráficas de línea son ideales para mostrar tendencias temporales, especialmente cuando los datos están organizados cronológicamente y la interpolación es relevante.



Diagramas de Cajas: Los diagramas de cajas, también conocidos como boxplots, son herramientas visuales que distribuyen datos a través de percentiles, facilitando la comparación entre múltiples grupos. Estos gráficos permiten realizar un análisis multivariado, ya que muestran cómo se agrupan y distribuyen los datos. Un diagrama de caja ilustra claramente la mediana, en lugar de la media, y resalta los cuartiles de un conjunto de datos. Además, permite identificar visualmente los valores atípicos, que pueden ser indicativos de casos especiales o errores en los datos. En resumen, los diagramas de cajas son útiles para representar la variabilidad y la tendencia central de los datos de manera efectiva.

Estadística para Machine Learning - Actividad No 3

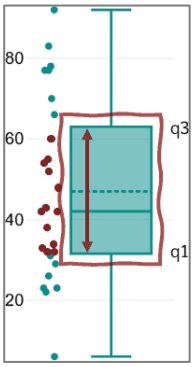
https://www.youtube.com/watch?app=desktop&v=GBNpyyApgdA

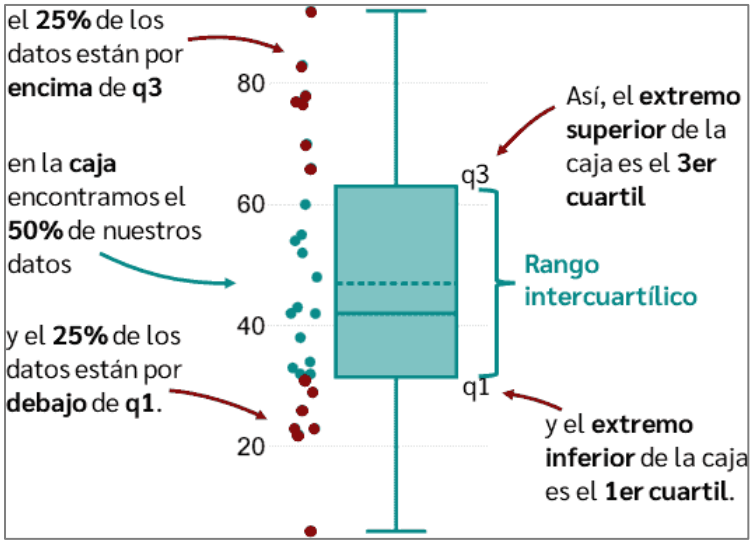


¿Cómo se interpreta un diagrama de caja?[1]

La propia caja indica el intervalo en el que se encuentra el 50% de todos los valores. Así, el extremo inferior de la caja es el 1er cuartil y el extremo superior es el 3er cuartil.

Por lo tanto, si por debajo de q1 se encuentra el 25% de los datos y por encima de q3 se encuentra el 25% de los datos, en la propia caja se encuentra el 50% de los datos.



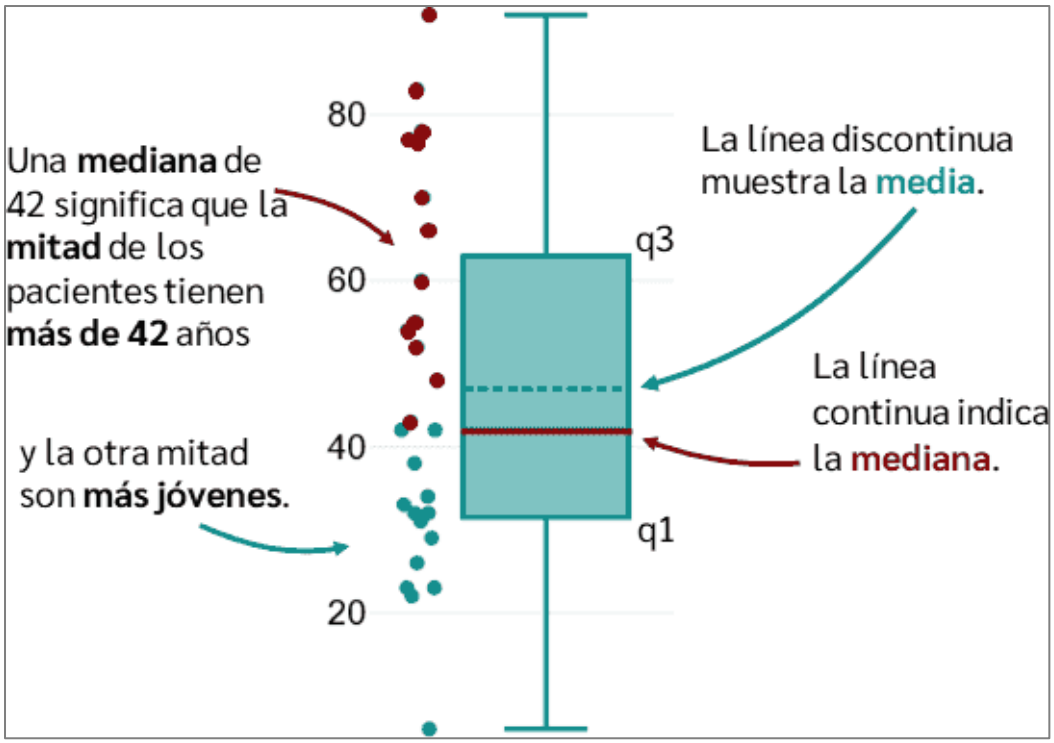


[1] Tomado de https://datatab.es/tutorial/box-plot

Estadística para Machine Learning - Actividad No 3

Supongamos que observamos la edad de los individuos en un diagrama de caja, y q1 es 31 años, entonces significa que el 25% de los participantes son menores de 31 años. Si q3 es 63 años, significa que el 25% de los participantes tienen más de 63 años, por lo que el 50% de los participantes tienen entre 31 y 63 años. Así pues, entre q1 y q3 está el rango intercuartílico.

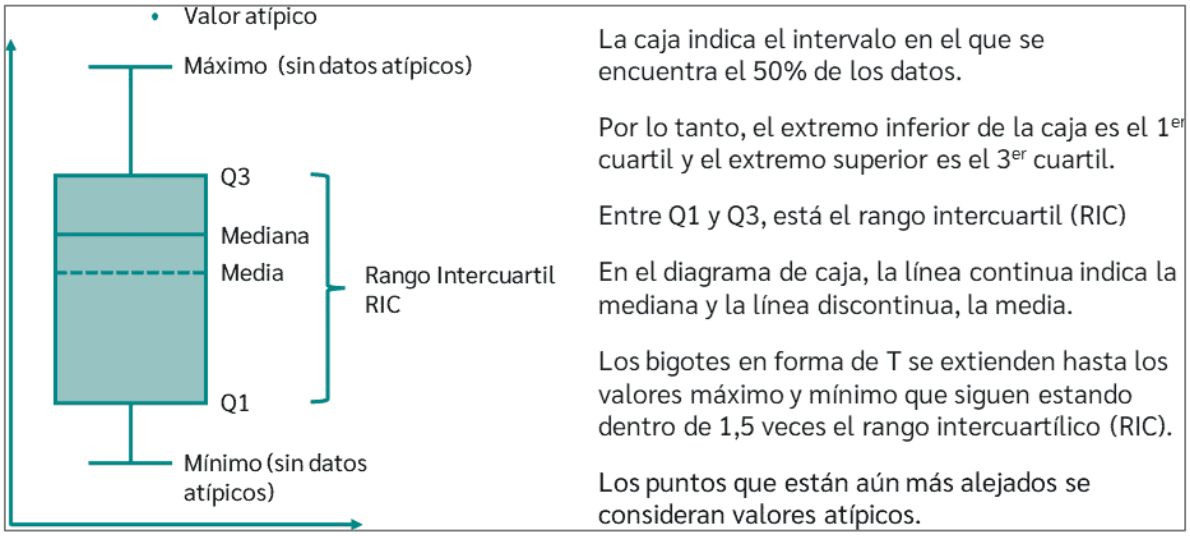
En el diagrama de caja, la línea continua indica la mediana y la línea discontinua, la media.



Por ejemplo, si la mediana es 42, significa que la mitad de los participantes son menores de 42 años y la otra mitad mayores de 42 años. Así pues, la mediana divide a los individuos en dos grupos iguales.

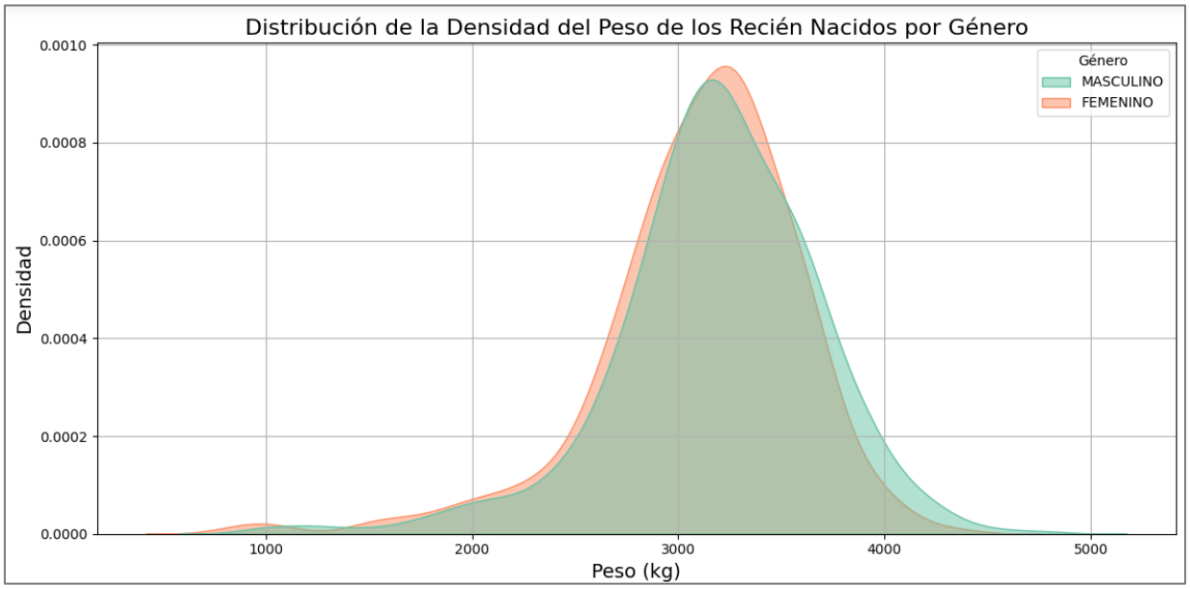
Los bigotes en forma de T llegan hasta el último punto, que sigue estando dentro de 1.5 veces el rango intercuartílico. ¿Qué significa esto? El bigote en forma de T es el valor máximo de tus datos, pero como máximo 1.5 veces el rango intercuartílico. Por lo tanto, si hay un valor atípico, el bigote llega hasta 1.5 veces el rango intercuartílico. Si no hay ningún valor atípico, el bigote es el valor máximo.

Así que el bigote superior es el valor máximo o 1.5 veces el rango intercuartílico. Dependiendo de qué valor sea menor. Lo mismo ocurre con el bigote inferior, que es el mínimo o 1.5 veces el rango intercuartílico.



Los puntos más alejados se consideran valores atípicos. Si ningún punto está más alejado que 1.5 veces el rango intercuartílico, el bigote en forma de T indica el valor máximo o mínimo.

Diagramas de Densidad: Los diagramas de Densidad representan la distribución de una variable numérica a través de una curva continua que muestra la densidad de probabilidad.



Estos diagramas se utilizan para:

Visualizar Distribuciones: Muestran la forma de la distribución (normal, sesgada, etc.).

Comparar Grupos: Facilitan la comparación de distribuciones entre diferentes categorías.

Detectar Valores Atípicos: Ayudan a identificar anomalías en los datos.

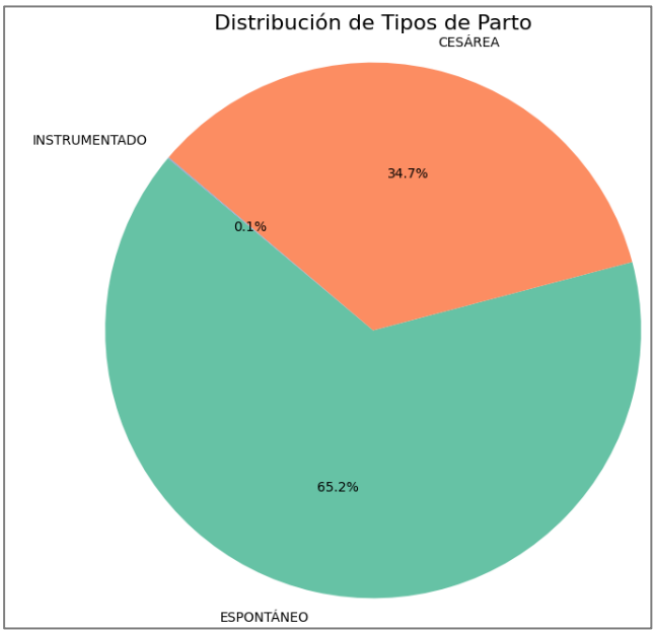
Análisis Exploratorio: Útiles en la exploración de datos para entender patrones.

Toma de Decisiones: Proporcionan información clara para fundamentar decisiones.

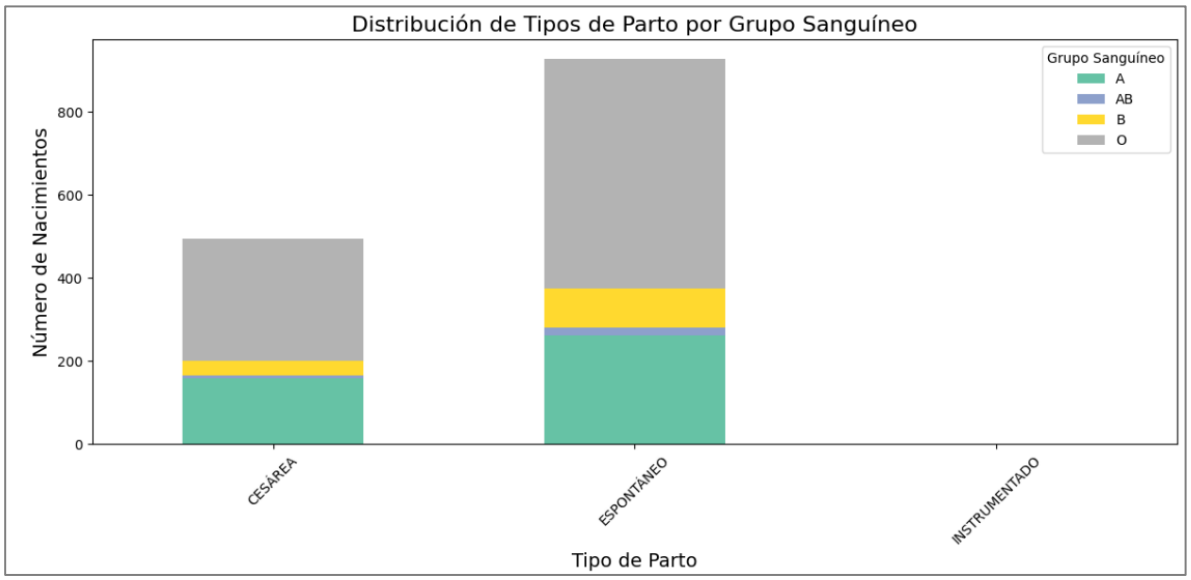
Gráficos De Composición

Las gráficas de composición sirven para representar la relación entre las partes y el conjunto total de datos. Incluyen gráficos circulares, gráficos de barras apiladas y gráficos de área. Estas visualizaciones pueden ser estáticas, reflejando la distribución actual de un total, o dinámicas, mostrando cómo esa distribución varía con el tiempo. Además, pueden ilustrar las proporciones de cada parte respecto al total o los valores absolutos dentro del conjunto.

Gráficos Circulares o de Torta: Se utilizan para mostrar la proporción de partes dentro de un conjunto categórico, comparando una sección con el total. Son efectivos con pocas categorías, facilitando la comprensión rápida de proporciones. Sin embargo, con más de diez segmentos, pueden dificultar la comparación y reducir la claridad visual.



Gráficos de Barras Apiladas: Se utilizan para desglosar una categoría principal en subcategorías, facilitando comparaciones tanto verticales como horizontales. Estas barras se apilan, de modo que la altura total refleja el resultado combinado, lo que mejora la comprensión del gráfico. Son efectivos para comparar valores lado a lado o uno sobre otro. Su principal ventaja es la facilidad de lectura, mientras que su desventaja es que pueden volverse confusos con demasiados valores debido a las limitaciones en el espacio de los ejes.



Gráficos de Área: los gráficos de área representan una dimensión categórica en relación con una variable temporal, acumulando y apilando los valores de abajo hacia arriba. Esto permite visualizar cómo cada categoría aporta al total a lo largo del tiempo. Se utilizan principalmente para mostrar tendencias en lugar de valores específicos. Existen dos tipos: los gráficos de área agrupados, que inician en el mismo eje cero, y los gráficos de área apilados, donde cada serie comienza desde el final de la anterior.

Ejemplos

Para llevar a cabo los siguientes ejemplos, puedes utilizar Anaconda Navigator o Google Colaboratory como plataforma. Además, es necesario importar las siguientes librerías:

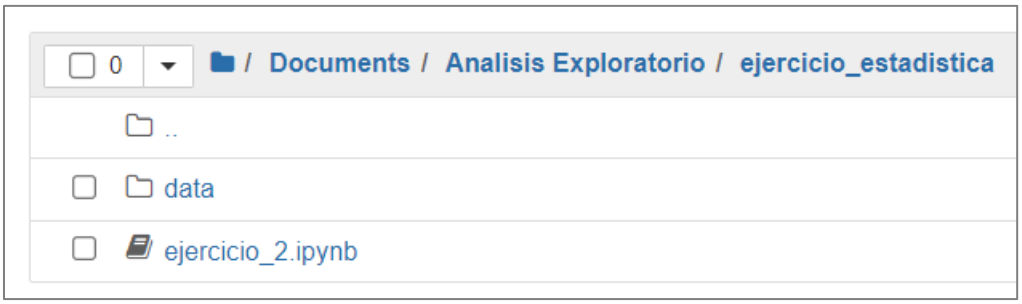
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import statsmodels.api as sm
import scipy.stats as stats
```

Debes tener en cuenta que, si no están instaladas, deberás instalarlas previamente.

- ✓ Investiga para que se utilizan estas librerías

Los datos que utilizaremos para los ejemplos son:

No olvides tener esta estructura



Al iniciar debemos cargar los datos necesarios

```
df = pd.read_csv('./data/Nacidos_Vivos_en_2016_de_Guadalajara_de_Buga.csv')
df
```

Estadística para Machine Learning - Actividad No 3

Antes de comenzar, realiza un análisis preliminar de los datos, tal como se hizo en clases anteriores. Luego, procede a crear los siguientes gráficos:

grafica de correlación

```
# Crear el gráfico de dispersión
plt.scatter(df['Edad del Padre'],df['Edad de la Madre'])

# Personalizar el gráfico
plt.title('Relación entre la edad del padre y de la madre')
plt.xlabel('Edad Padre')
plt.ylabel('Edad Madre')
plt.grid(True)

# Mostrar el gráfico
plt.show()
```

```
# Crear el gráfico de dispersión
plt.figure(figsize=(10, 6))  # Ajustar el tamaño del gráfico

# Utilizar colores basados en otra variable (por ejemplo, 'Edad del Padre')
sc = plt.scatter(df['Edad del Padre'], df['Edad de la Madre'], c=df['Edad del Padre'],
                 cmap='viridis', s=20, alpha=0.7, edgecolor='k')

# Personalizar el gráfico
plt.title('Relación entre la Edad del Padre y de la Madre', fontsize=14)
plt.xlabel('Edad del Padre', fontsize=12)
plt.ylabel('Edad de la Madre', fontsize=12)
plt.grid(True, linestyle='--', alpha=0.6)  # Estilo de la cuadrícula

# Añadir barra de colores para dar contexto al color de los puntos
cbar = plt.colorbar(sc)
cbar.set_label('Edad del Padre')

# Añadir una línea de referencia (opcional, si tiene sentido)
plt.axline((df['Edad del Padre'].mean(), df['Edad de la Madre'].mean()), slope=1, color='r', linestyle='--')

# Mostrar el gráfico
plt.show()
```

```
# Crear el gráfico de dispersión
plt.figure(figsize=(10, 6))  # Ajustar el tamaño del gráfico

# Utilizar colores basados en otra variable (por ejemplo, 'Edad del Padre')
sc = plt.scatter(df['Edad del Padre'], df['Edad de la Madre'],
                 c=df['Edad de la Madre'], cmap='viridis', s=100, alpha=0.7, edgecolor='k')

# Personalizar el gráfico
plt.title('Relación entre la Edad del Padre y de la Madre', fontsize=14)
plt.xlabel('Edad del Padre', fontsize=12)
plt.ylabel('Edad de la Madre', fontsize=12)
plt.grid(True, linestyle='--', alpha=0.6)  # Estilo de la cuadrícula

# Establecer los ticks en los ejes de 5 en 5
plt.xticks(np.arange(min(df['Edad del Padre']), max(df['Edad del Padre']) + 1, 5))
plt.yticks(np.arange(min(df['Edad de la Madre']), max(df['Edad de la Madre']) + 1, 5))

# Añadir barra de colores para dar contexto al color de los puntos
cbar = plt.colorbar(sc)
cbar.set_label('Edad de la Madre')

# Añadir una línea de referencia (opcional, si tiene sentido)
plt.axline((df['Edad del Padre'].mean(), df['Edad de la Madre'].mean()), slope=1, color='r', linestyle='--')

# Mostrar el gráfico
plt.show()
```

Mapas de calor

```
# Calcular la matriz de correlación
correlacion = df.corr()

# Crear el mapa de calor
plt.figure(figsize=(8, 6))
sns.heatmap(correlacion, annot=True, cmap='coolwarm', linewidths=0.5)

# Personalizar el gráfico
plt.title('Mapa de Calor de la Correlación', fontsize=14)
plt.show()
```

```
# Calcular la matriz de correlación
correlacion = df.corr()

# Crear el mapa de calor
plt.figure(figsize=(10, 8), dpi=100)  # Aumentar tamaño y resolución
heatmap = sns.heatmap(correlacion,
                      annot=True,
                      fmt='.2f',  # Mostrar solo dos decimales
                      cmap='coolwarm',
                      linewidths=0.5,
                      cbar_kws={"shrink": 0.8})  # Ajustar barra de color

# Personalizar el gráfico
plt.title('Mapa de Calor de la Correlación entre Variables', fontsize=16)
plt.xlabel('Variables', fontsize=12)
plt.ylabel('Variables', fontsize=12)

# Mostrar el gráfico
plt.show()
```

Histogramas

```
# Crear el histograma
plt.hist(df['Edad de la Madre'], bins=5)

# Personalizar el gráfico
plt.title('Histograma de Edades de la Madre')
plt.xlabel('Edad de la Madre')
plt.ylabel('Frecuencia')
plt.grid(axis='y')

# Mostrar el gráfico
plt.show()
```

```python
# Crear el histograma
plt.figure(figsize=(10, 6))  # Tamaño de la figura
sns.histplot(df['Edad de la Madre'], bins=5, kde=True, color='lightblue')  # Usar Seaborn para un mejor estilo

# Personalizar el gráfico
plt.title('Distribución de Edades de la Madre', fontsize=18)
plt.xlabel('Edad de la Madre', fontsize=14)
plt.ylabel('Frecuencia', fontsize=14)
plt.xticks(fontsize=12)  # Tamaño de fuente de etiquetas en el eje X
plt.yticks(fontsize=12)  # Tamaño de fuente de etiquetas en el eje Y
plt.grid(axis='y', linestyle='--', alpha=0.7)  # Líneas de rejilla más suaves

# Añadir líneas de densidad
plt.axvline(df['Edad de la Madre'].mean(), color='blue', linestyle='--', label='Media')  # Línea de la media
plt.axvline(df['Edad de la Madre'].median(), color='green', linestyle='--', label='Mediana')  # Línea de la mediana

# Añadir leyenda
plt.legend()

# Mostrar el gráfico
plt.tight_layout()  # Ajustar el layout
plt.show()
```

```python
# Crear el histograma
plt.hist(df['Edad del Padre'], bins=5)

# Personalizar el gráfico
plt.title('Histograma de Edades del Padre')
plt.xlabel('Edad del Padre')
plt.ylabel('Frecuencia')
plt.grid(axis='y')

# Mostrar el gráfico
plt.show()
```

```python
# Crear el histograma
plt.figure(figsize=(10, 6))  # Tamaño de la figura
sns.histplot(df['Edad del Padre'], bins=5, kde=True, color='lightcoral')  # Usar Seaborn para un mejor estilo

# Personalizar el gráfico
plt.title('Distribución de Edades del Padre', fontsize=18)
plt.xlabel('Edad del Padre', fontsize=14)
plt.ylabel('Frecuencia', fontsize=14)
plt.xticks(fontsize=12)  # Tamaño de fuente de etiquetas en el eje X
plt.yticks(fontsize=12)  # Tamaño de fuente de etiquetas en el eje Y
plt.grid(axis='y', linestyle='--', alpha=0.7)  # Líneas de rejilla más suaves

# Añadir líneas de densidad
plt.axvline(df['Edad del Padre'].mean(), color='blue', linestyle='--', label='Media')  # Línea de la media
plt.axvline(df['Edad del Padre'].median(), color='green', linestyle='--', label='Mediana')  # Línea de la mediana

# Añadir leyenda
plt.legend()

# Mostrar el gráfico
plt.tight_layout()  # Ajustar el layout
plt.show()
```

```python
# Crear el histograma con porcentajes
plt.figure(figsize=(10, 6))  # Tamaño de la figura
# Usar 'percent' para mostrar porcentajes
hist = sns.histplot(df['Edad del Padre'], bins=5, kde=True, color='lightcoral', stat='percent')

# Personalizar el gráfico
plt.title('Distribución de Edades del Padre', fontsize=18)
plt.xlabel('Edad del Padre', fontsize=14)
plt.ylabel('Porcentaje (%)', fontsize=14)  # Cambiar la etiqueta a 'Porcentaje'
plt.xticks(fontsize=12)  # Tamaño de fuente de etiquetas en el eje X
plt.yticks(fontsize=12)  # Tamaño de fuente de etiquetas en el eje Y
plt.grid(axis='y', linestyle='--', alpha=0.7)  # Líneas de rejilla más suaves

# Añadir líneas de densidad
plt.axvline(df['Edad del Padre'].mean(), color='blue', linestyle='--', label='Media')  # Línea de la media
plt.axvline(df['Edad del Padre'].median(), color='green', linestyle='--', label='Mediana')  # Línea de la mediana

# Añadir leyenda
plt.legend()

# Mostrar el gráfico
plt.tight_layout()  # Ajustar el layout
plt.show()
```

```python
# Crear el histograma
plt.hist(df['Número de Embarazos'], bins=5)

# Personalizar el gráfico
plt.title('Histograma número de embarazos')
plt.xlabel('Número de Embarazos')
plt.ylabel('Frecuencia')
plt.grid(axis='y')

# Mostrar el gráfico
plt.show()
```

```python
# Crear el histograma
plt.figure(figsize=(10, 6))  # Tamaño de la figura
sns.histplot(df['Número de Embarazos'], bins=5, kde=True, color='skyblue')  # Usar Seaborn para un mejor estilo

# Personalizar el gráfico
plt.title('Distribución del Número de Embarazos', fontsize=18)
plt.xlabel('Número de Embarazos', fontsize=14)
plt.ylabel('Frecuencia', fontsize=14)
plt.xticks(fontsize=12)  # Tamaño de fuente de etiquetas en el eje X
plt.yticks(fontsize=12)  # Tamaño de fuente de etiquetas en el eje Y
plt.grid(axis='y', linestyle='--', alpha=0.7)  # Líneas de rejilla más suaves

# Añadir líneas de densidad
plt.axvline(df['Número de Embarazos'].mean(), color='red', linestyle='--', label='Media')  # Línea de la media
plt.axvline(df['Número de Embarazos'].median(), color='green', linestyle='--', label='Mediana')  # Línea de la mediana

# Añadir leyenda
plt.legend()

# Mostrar el gráfico
plt.tight_layout()  # Ajustar el layout
plt.show()
```

Gráfico de cuantiles

```
# datos
data = df['Edad de la Madre']
# Gráfico Q-Q comparando con una distribución normal
sm.qqplot(data, line='s')
plt.show()
```

```
# Datos
variable_a = df['Edad de la Madre']
variable_b = df['Edad del Padre']

# Ordenamos las variables para obtener los cuantiles
variable_a_sorted = np.sort(variable_a)
variable_b_sorted = np.sort(variable_b)

# Aseguramos que ambas variables tengan el mismo número de cuantiles
min_length = min(len(variable_a_sorted), len(variable_b_sorted))
variable_a_sorted = variable_a_sorted[:min_length]
variable_b_sorted = variable_b_sorted[:min_length]

# Crear la figura
plt.figure(figsize=(6, 6))

# Graficar los puntos de la primera variable (A) en azul
plt.plot(variable_a_sorted, variable_a_sorted, 'o-', color='blue', label="Cuantiles de Edad de la Madre")

# Graficar los puntos de la segunda variable (B) en naranja
plt.plot(variable_a_sorted, variable_b_sorted, 'o-', color='orange', label="Cuantiles de Edad del Padre")

# Línea de referencia y = x
plt.plot([min(variable_a_sorted), max(variable_a_sorted)],
         [min(variable_a_sorted), max(variable_a_sorted)], 'r--', label="Línea y = x")

# Añadir títulos y etiquetas
plt.title("Gráfico de Cuantiles: Comparación Edad de la Madre vs Edad del Padre")
plt.xlabel("Cuantiles de Edad de la Madre")
plt.ylabel("Cuantiles de Edad del Padre")

# **Mostrar leyenda para identificar cada color**
plt.legend(loc="best")  # La leyenda aparecerá en la mejor posición posible
plt.grid(True)
plt.show()
```

Gráfica de barras

```
# Calcular el conteo de nacimientos por género
conteo_genero = df['Género'].value_counts()

# Calcular los porcentajes
porcentajes_genero = conteo_genero / conteo_genero.sum() * 100

# Crear la gráfica de barras
plt.figure(figsize=(8, 5))
plt.bar(porcentajes_genero.index, porcentajes_genero.values, color=['lightblue', 'salmon'])

# Añadir títulos y etiquetas
plt.title('Porcentaje de Niños y Niñas Nacidos')
plt.xlabel('Género')
plt.ylabel('Porcentaje (%)')

# Mostrar la gráfica
plt.grid(axis='y')
plt.xticks(rotation=0)
plt.ylim(0, 100)  # Ajustar el límite del eje Y al 100%
plt.show()
```

```python
# Calcular el conteo de nacimientos por régimen de seguridad
conteo_regimen_seguridad = df['Régimen Seguridad'].value_counts()

# Crear la gráfica de barras
plt.figure(figsize=(12, 7))
bars = plt.bar(conteo_regimen_seguridad.index, conteo_regimen_seguridad.values,
               color=['#add8e6', '#ffcccb', '#98fb98', '#fffacd'])

# Añadir títulos y etiquetas
plt.title('Número de Nacimientos por Régimen de Seguridad', fontsize=20)
plt.xlabel('Régimen de Seguridad', fontsize=16)
plt.ylabel('Número de Nacimientos', fontsize=16)

# Añadir anotaciones en las barras
for bar in bars:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width() / 2, yval + 2, int(yval), ha='center', va='bottom', fontsize=12)

# Mejorar la estética del gráfico
plt.grid(axis='y', linestyle='--', alpha=0.7)  # Cuadrícula en el eje Y
plt.xticks(rotation=45, fontsize=12)  # Rotar las etiquetas del eje X
plt.yticks(fontsize=12)  # Tamaño de las etiquetas del eje Y
plt.ylim(0, conteo_regimen_seguridad.max() + 10)  # Ajustar el límite del eje Y

# Ajustar el diseño para que no se corten los elementos
plt.tight_layout()
plt.show()
```

Gráfico de líneas

```python
# Convertir la columna de fecha a tipo datetime
df['Fecha de Nacimiento'] = pd.to_datetime(df['Fecha de Nacimiento'])

# Agrupar por fecha de nacimiento y calcular el peso promedio
peso_promedio = df.groupby('Fecha de Nacimiento')['Peso'].mean().reset_index()

# Crear la gráfica de líneas
plt.figure(figsize=(12, 6))
plt.plot(peso_promedio['Fecha de Nacimiento'], peso_promedio['Peso'], marker='o', color='b', linestyle='-')

# Añadir títulos y etiquetas
plt.title('Tendencia del Peso Promedio de Nacimientos a lo Largo del Tiempo')
plt.xlabel('Fecha de Nacimiento')
plt.ylabel('Peso Promedio (kg)')
plt.xticks(rotation=45)  # Rotar las etiquetas del eje X para mejor visibilidad

# Mostrar la gráfica
plt.grid()
plt.tight_layout()  # Ajustar el layout
plt.show()
```

```
# Convertir la columna de fecha a tipo datetime
df['Fecha de Nacimiento'] = pd.to_datetime(df['Fecha de Nacimiento'])

# Agrupar por mes y año, y contar el número de nacimientos
nacimientos_por_mes = df.groupby(df['Fecha de Nacimiento'].dt.to_period('M')).size().reset_index(name='Número de Nacimientos')

# Convertir la columna de periodo a datetime para graficar
nacimientos_por_mes['Fecha de Nacimiento'] = nacimientos_por_mes['Fecha de Nacimiento'].dt.to_timestamp()

# Crear la gráfica de líneas
plt.figure(figsize=(12, 6))
plt.plot(nacimientos_por_mes['Fecha de Nacimiento'], nacimientos_por_mes['Número de Nacimientos'], marker='o',
         color='blue', linestyle='-')

# Añadir títulos y etiquetas
plt.title('Número de Nacimientos por Mes')
plt.xlabel('Fecha (Mes)')
plt.ylabel('Número de Nacimientos')
plt.xticks(rotation=45)  # Rotar las etiquetas del eje X para mejor visibilidad

# Mostrar la gráfica
plt.grid()
plt.tight_layout()  # Ajustar el layout
plt.show()
```

Gráficos de cajas

```
# Crear la gráfica de cajas
plt.figure(figsize=(10, 6))
sns.boxplot(x='Género', y='Peso', data=df, palette='pastel')

# Añadir títulos y etiquetas
plt.title('Distribución del Peso de los Recién Nacidos por Género')
plt.xlabel('Género')
plt.ylabel('Peso (kg)')

# Mostrar la gráfica
plt.grid(axis='y')
plt.tight_layout()
plt.show()
```

```
# Crear la gráfica de cajas
plt.figure(figsize=(10, 6))
sns.boxplot(x='Género', y='Peso', data=df, palette='Set2')

# Añadir títulos y etiquetas
plt.title('Distribución del Peso de los Recién Nacidos por Género', fontsize=16)
plt.xlabel('Género', fontsize=14)
plt.ylabel('Peso (kg)', fontsize=14)

# Añadir anotaciones para mediana y valores atípicos
for i in range(len(df['Género'].unique())):
    median = df[df['Género'] == df['Género'].unique()[i]]['Peso'].median()
    plt.text(i, median + 0.5, f'Mediana: {median:.1f} kg',
             horizontalalignment='center', color='black', fontsize=12)

# Mostrar la gráfica
plt.grid(axis='y')
plt.tight_layout()
plt.show()
```

```
#Crear la gráfica de cajas
plt.figure(figsize=(10, 6))
sns.boxplot(x='Género', y='Peso', data=df, palette='Set2')

# Añadir títulos y etiquetas
plt.title('Distribución del Peso de los Recién Nacidos por Género', fontsize=16)
plt.xlabel('Género', fontsize=14)
plt.ylabel('Peso (kg)', fontsize=14)

# Añadir anotaciones para mediana y media
for i in range(len(df['Género'].unique())):
    # Calcular mediana
    median = df[df['Género'] == df['Género'].unique()[i]]['Peso'].median()
    plt.text(i, median + 0.5, f'Mediana: {median:.1f} kg',
             horizontalalignment='center', color='black', fontsize=12)

    # Calcular media
    mean = df[df['Género'] == df['Género'].unique()[i]]['Peso'].mean()
    plt.text(i, mean + 0.5, f'Media: {mean:.1f} kg',
             horizontalalignment='center', color='red', fontsize=12)

# Mostrar la gráfica
plt.grid(axis='y')
plt.tight_layout()
plt.show()
```

```
# Crear la gráfica de cajas para Tiempo de Gestación
plt.figure(figsize=(10, 6))
sns.boxplot(x='Género', y='Tiempo de Gestación', data=df, palette='Set2')

# Añadir títulos y etiquetas
plt.title('Distribución del Tiempo de Gestación por Género', fontsize=16)
plt.xlabel('Género', fontsize=14)
plt.ylabel('Tiempo de Gestación (semanas)', fontsize=14)

# Añadir anotaciones para mediana y media
for i in range(len(df['Género'].unique())):
    # Calcular mediana
    median = df[df['Género'] == df['Género'].unique()[i]]['Tiempo de Gestación'].median()
    plt.text(i, median + 1, f'Mediana: {median:.1f} semanas',
             horizontalalignment='center', color='black', fontsize=12)

    # Calcular media
    mean = df[df['Género'] == df['Género'].unique()[i]]['Tiempo de Gestación'].mean()
    plt.text(i, mean + 1, f'Media: {mean:.1f} semanas',
             horizontalalignment='center', color='red', fontsize=12)

# Mostrar la gráfica
plt.grid(axis='y')
plt.tight_layout()
plt.show()
```

```
# Crear la gráfica de cajas para Peso desglosado por Género y Tipo de Parto
plt.figure(figsize=(12, 6))
sns.boxplot(x='Tipo de Parto', y='Peso', hue='Género', data=df, palette='Set2')

# Añadir títulos y etiquetas
plt.title('Distribución del Peso por Tipo de Parto y Género', fontsize=16)
plt.xlabel('Tipo de Parto', fontsize=14)
plt.ylabel('Peso (kg)', fontsize=14)

# Añadir leyenda
plt.legend(title='Género')

# Mostrar la gráfica
plt.grid(axis='y')
plt.tight_layout()
plt.show()
```

Gráficas de densidad

```
# Crear la gráfica de densidad
plt.figure(figsize=(12, 6))
sns.kdeplot(data=df, x='Peso', hue='Género', fill=True, common_norm=False, palette='Set2', alpha=0.5)

# Añadir títulos y etiquetas
plt.title('Distribución de la Densidad del Peso de los Recién Nacidos por Género', fontsize=16)
plt.xlabel('Peso (kg)', fontsize=14)
plt.ylabel('Densidad', fontsize=14)

# Mostrar la gráfica
plt.grid()
plt.tight_layout()
plt.show()
```

```python
# Verificar los valores únicos en la columna 'Género'
print(df['Género'].unique())

# Definir la paleta de colores
palette = {'MASCULINO': 'blue', 'FEMENINO': 'pink'}

# Crear la figura
plt.figure(figsize=(12, 6))

# Crear la gráfica de densidad para Tiempo de Gestación
sns.kdeplot(data=df, x='Tiempo de Gestación', hue='Género', fill=True, common_norm=False, palette=palette, alpha=0.5)

#Controlar posición letreros
letrero1 = 0.02
letrero2 = 0.07

# Calcular y añadir la media y mediana
for genero in df['Género'].unique():
    mean = df[df['Género'] == genero]['Tiempo de Gestación'].mean()
    median = df[df['Género'] == genero]['Tiempo de Gestación'].median()

    # Añadir líneas para la media y mediana
    plt.axvline(mean, linestyle='--', color='black')
    plt.axvline(median, linestyle=':', color='red')

    # Ajustar las posiciones de las etiquetas
    plt.text(mean, letrero1, f'Media {genero}: {mean:.1f} semanas', color='black', fontsize=10, ha='center', va='bottom')
    plt.text(median, letrero2, f'Mediana {genero}: {median:.1f} semanas', color='red', fontsize=10, ha='center', va='bottom')
    letrero1 += 0.02
    letrero2 += 0.02

# Añadir títulos y etiquetas
plt.title('Distribución de la Densidad del Tiempo de Gestación por Género', fontsize=16)
plt.xlabel('Tiempo de Gestación (semanas)', fontsize=14)
plt.ylabel('Densidad', fontsize=14)

# Añadir leyenda manualmente para los géneros
handles = [plt.Line2D([0], [0], color=palette['MASCULINO'], lw=4, label='Masculino'),
           plt.Line2D([0], [0], color=palette['FEMENINO'], lw=4, label='Femenino')]
plt.legend(handles=handles, title='Género')

# Mostrar la gráfica
plt.grid()
plt.tight_layout()
plt.show()
```

Gráfica Circular

```python
# Contar la cantidad de nacimientos por tipo de parto
parto_count = df['Tipo de Parto'].value_counts()

# Crear la gráfica circular
plt.figure(figsize=(8, 8))
plt.pie(parto_count, labels=parto_count.index, autopct='%1.1f%%', startangle=140, colors=['#66c2a5', '#fc8d62', '#8da0cb'])

# Títulos
plt.title('Distribución de Tipos de Parto', fontsize=16)
plt.axis('equal')  # Para que la gráfica sea un círculo
plt.show()
```

```
# Contar la cantidad de nacimientos por grupo sanguíneo
grupo_sanguineo_count = df['Grupo Sanguíneo'].value_counts()

# Crear la gráfica circular
plt.figure(figsize=(8, 8))
plt.pie(grupo_sanguineo_count, labels=grupo_sanguineo_count.index, autopct='%1.1f%%', startangle=140,
        colors=plt.cm.tab10.colors)

# Títulos
plt.title('Distribución de Grupos Sanguíneos', fontsize=16)
plt.axis('equal')  # Para que la gráfica sea un círculo
plt.show()
```

```
# Verificar si las columnas tienen datos
if 'Tipo de Parto' in df.columns and 'Grupo Sanguíneo' in df.columns:
    # Contar la cantidad de cada grupo sanguíneo por tipo de parto
    grupo_parto_counts = df.groupby('Tipo de Parto')['Grupo Sanguíneo'].value_counts().unstack(fill_value=0)

    # Verificar si hay datos para graficar
    if not grupo_parto_counts.empty:
        # Crear gráficos circulares para cada tipo de parto
        fig, axes = plt.subplots(nrows=1, ncols=len(grupo_parto_counts), figsize=(15, 6), sharey=True)

        # Crear un gráfico circular para cada tipo de parto
        for ax, (tipo_parto, grupos) in zip(axes, grupo_parto_counts.iterrows()):
            ax.pie(grupos, labels=grupos.index, autopct='%1.1f%%', startangle=140,
                   colors=sns.color_palette('Set2', len(grupos)))
            ax.set_title(f'Distribución de Grupos Sanguíneos\n({tipo_parto})', fontsize=14)

        # Añadir un título general
        plt.suptitle('Distribución de Grupos Sanguíneos por Tipo de Parto', fontsize=16)

        # Ajustar el aspecto de los gráficos
        plt.tight_layout()
        plt.subplots_adjust(top=0.85)  # Para dar espacio al título general

        # Mostrar el gráfico
        plt.show()
    else:
        print("No hay datos disponibles para graficar.")
else:
    print("Las columnas 'Tipo de Parto' y 'Grupo Sanguíneo' no están presentes en el DataFrame.")
```

Barras Apiladas

```
# Contar el número de ocurrencias de cada combinación de Tipo de Parto y Grupo Sanguíneo
data = df.groupby(['Tipo de Parto', 'Grupo Sanguíneo']).size().unstack(fill_value=0)

# Crear la gráfica de barras apiladas
ax = data.plot(kind='bar', stacked=True, figsize=(12, 6), colormap='Set2')

# Añadir títulos y etiquetas
plt.title('Distribución de Tipos de Parto por Grupo Sanguíneo', fontsize=16)
plt.xlabel('Tipo de Parto', fontsize=14)
plt.ylabel('Número de Nacimientos', fontsize=14)

# Mostrar la gráfica
plt.xticks(rotation=45)
plt.legend(title='Grupo Sanguíneo')
plt.tight_layout()
plt.show()
```

Estadística para Machine Learning - Actividad No 3

```python
# Contar el número de ocurrencias de cada combinación de Tipo de Parto y Grupo Sanguíneo
data = df.groupby(['Tipo de Parto', 'Grupo Sanguíneo']).size().unstack(fill_value=0)

# Crear la gráfica de barras apiladas
fig, ax = plt.subplots(figsize=(12, 8))
data.plot(kind='bar', stacked=True, ax=ax, colormap='Set2', edgecolor='black')

# Añadir títulos y etiquetas
ax.set_title('Distribución de Tipos de Parto por Grupo Sanguíneo', fontsize=18, fontweight='bold')
ax.set_xlabel('Tipo de Parto', fontsize=14)
ax.set_ylabel('Número de Nacimientos', fontsize=14)
ax.legend(title='Grupo Sanguíneo', title_fontsize='13', fontsize='12')

# Añadir etiquetas de valor en las barras
for i in range(len(data)):
    for j in range(len(data.columns)):
        ax.text(i, data.values[i].cumsum()[j] - data.values[i][j] / 2,
                data.values[i][j],
                ha='center', va='center', color='white', fontsize=10)

# Mejorar la estética
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()

# Mostrar la gráfica
plt.show()
```

```python
# Contar el número de ocurrencias de cada combinación de Tipo de Parto y Pertenencia Étnica
data = df.groupby(['Tipo de Parto', 'Pertenencia Étnica']).size().unstack(fill_value=0)

# Crear la gráfica de barras apiladas
fig, ax = plt.subplots(figsize=(12, 8))
data.plot(kind='bar', stacked=True, ax=ax, colormap='Set2', edgecolor='black')

# Añadir títulos y etiquetas
ax.set_title('Distribución de Tipos de Parto por Pertenencia Étnica', fontsize=18, fontweight='bold')
ax.set_xlabel('Tipo de Parto', fontsize=14)
ax.set_ylabel('Número de Nacimientos', fontsize=14)
ax.legend(title='Pertenencia Étnica', title_fontsize='13', fontsize='12')

# Añadir etiquetas de valor en las barras
for i in range(len(data)):
    for j in range(len(data.columns)):
        ax.text(i, data.values[i].cumsum()[j] - data.values[i][j] / 2,
                data.values[i][j],
                ha='center', va='center', color='white', fontsize=10)

# Mejorar la estética
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()

# Mostrar la gráfica
plt.show()
```

Estadística para Machine Learning - Actividad No 3

- ✓ Investiga los conceptos de otros tipos de gráficas, así como su implementación en Python, y analiza en qué situaciones se emplean y cuál es su propósito. Algunos ejemplos incluyen:
  - Histogramas en paralelo
  - Diagramas de barras agrupados
  - Diagramas de Pareto
  - Gráficos en mosaico
  - Diagramas en árbol
  - Diagramas de caja en paralelo
  - Diagramas de tallo y hojas

Bibliografía

https://www.plantillaspyme.com/blog-pymes/excel/partes-de-un-grafico-excel

https://www.datos.gov.co/Salud-y-Protecci-n-Social/Nacidos-Vivos-en-el-a-o-2016-de-Guadalajara-de-Bug/u5y2-ufx9/data

https://datatab.es/tutorial/box-plot

https://www.datos.gov.co/Salud-y-Protecci-n-Social/Nacidos-vivos-en-el-Municipio-de-Puerto-L-pez/7w5q-pt2y/about_data

https://www.datos.gov.co/Salud-y-Protecci-n-Social/Nacidos-vivos-en-el-Municipio-de-Acac-as/7ifb-wrqs/about_data

https://www.datos.gov.co/Salud-y-Protecci-n-Social/Nacidos-vivos-en-el-Municipio-de-Puerto-L-pez/7w5q-pt2y/about_data

https://www.datos.gov.co/Salud-y-Protecci-n-Social/Guadalajara-de-Buga-Nacidos-Vivos-2018-2021-Guadal/vskz-jk5x/about_data

https://www.datos.gov.co/Salud-y-Protecci-n-Social/43-Nacidos-Vivos-en-Municipio-de-Bucaramanga-enero/x5xp-9w4b/about_data