Contenido

"Python for Data Analysis" por Wes McKinney

"Data Science for Business" por Foster Provost y Tom Fawcett

"Practical Statistics for Data Scientists" por Peter Bruce y Andrew Bruce

Recursos en Línea

Kaggle Learn

URL: Kaggle: Learn

DataCamp

URL: DataCamp.

Towards Data Science en Medium

URL: Towards Data Science

Documentación Oficial

Documentación de Pandas

URL: Pandas Documentation

REALIZAR LAS MEDIDAS DESCRIPTIVAS DE LAS VARIABLES EN BASE A SU TIPO.REALIZAR LOS INFORMES SEUGUN SEAN LOS DATOS (SIMETRICOS O ASIMETRICOS).

A Partir del contenido de tu cuaderno de Jupyter. Cargamos y visualizamos los datos básicos del dataset de estudiantes.

Para continuar y calcular las medidas descriptivas según el tipo de datos de cada variable, procederemos de la siguiente manera:

Variables Numéricas: Calcularemos medidas como la media, mediana, desviación estándar, mínimo y máximo.

Variables Categóricas: Analizaremos la frecuencia de cada categoría para entender cómo se distribuyen los datos.

ANALISIS.

REVISION Y ANALISIS DE DATOS

Revisión y limpieza de los datos: Realizamos limpieza parece haber un problema con la columna "Asistencia", que incluye múltiples datos en una sola entrada. Necesitaríamos limpiar eso primero.

Cálculo de medidas descriptivas detalladas para cada tipo de variable.

Para las variables numéricas (edad, puntajes y asistencia), calcularemos la media, mediana, desviación estándar, mínimo y máximo.

Para las variables categóricas (género), obtendremos la frecuencia de cada categoría.

Procederemos a limpiar los datos y calcular las estadísticas descriptivas:

El problema radica en que la última columna tiene un formato incorrecto debido a datos duplicados y separadores adicionales. Vamos a corregir esto extrayendo solo las columnas relevantes y limpiando los datos para asegurarnos de que tengamos un conjunto de datos limpio y útil para el análisis:

Se Procedió a limpiar estos datos y luego calcular las estadísticas descriptivas.

Teníamos también un problema con la columna de "Asistencia", ya que contiene datos adicionales que no fueron correctamente separados. Revisamos este problema y aplicamos una limpieza adicional para asegurarnos de que la columna solo contenga los valores numéricos de asistencia, y luego procederemos a calcular las estadísticas descriptivas.

La columna "Asistencia" tenía múltiples valores y caracteres extra. Se extrajeron solo los valores numéricos correctos para esta columna y luego se procedió con el análisis descriptivo.

Cuando se tuvieron los datos correctamente, sacamos las estadísticas descriptivas para las variables numéricas y las frecuencias para la variable categórica "Género":

Estadísticas descriptivas para tu conjunto de datos:

Medidas Descriptivas para Variables Numéricas

Edad:

Media: 18.8 años

Desviación Estándar: 1.14 años

Mínimo: 17 años

Máximo: 20 años

Puntaje Matemáticas:

Media: 79.3

Desviación Estándar: 14.44

Mínimo: 53

Máximo: 98

Puntaje Lengua:

Media: 77.0

Desviación Estándar: 17.31

Mínimo: 51

Máximo: 100

Asistencia:

Media: 85.1%

Desviación Estándar: 10.93%

Mínimo: 68%

Máximo: 100%

Frecuencias para la Variable Categórica 'Género'

Masculino: 7 estudiantes

Femenino: 3 estudiantes

Los resultados obtenidos nos proporcionan una visión clara de las características del grupo de estudiantes analizado, incluyendo tendencias centrales y variabilidad en las calificaciones y asistencia, así como la distribución por género.

Vamos a especificar como realizamos la limpieza de datos al eliminar columnas duplicadas.

Paso 1: Identificación del problema

Primero, observamos que el conjunto de datos tenía un problema en la última columna, donde los datos estaban duplicados y separados por múltiples puntos y comas. El contenido de esta columna no solo incluía la asistencia, sino también datos repetidos de otras columnas.

Paso 2: Eliminación de columnas innecesarias

Decidimos eliminar la última columna que contenía datos duplicados y mal formateados. Sin embargo, necesitábamos extraer primero el valor correcto de asistencia de esta columna antes de descartarla. Para ello, utilizamos expresiones regulares para extraer solo el primer número de la columna, que correspondía al valor de asistencia. Aquí está el código relevante para ese paso:

```
# Extraer el número correcto de asistencia de la cadena mal formada

print("Datos originales:\n", df)

df = df.rename(columns={'Asistencia;;;;;Edad;Genero;Puntaje matematicas;Puntaje Lengua;Asistencia': 'Asistencia'})

df['Asistencia'] = df['Asistencia'].str.extract('(\d+);;;;;').astype(int)
```

Paso 3: Corrección de nombres de columnas

Una vez eliminada la columna problemática, también nos aseguramos de que las columnas restantes tuvieran nombres adecuados y comprensibles, haciendo que el conjunto de datos fuera más fácil de manejar y analizar.

Paso 4: Verificación y corrección de tipos de datos

Finalmente, verificamos y corregimos los tipos de datos de las columnas para asegurarme de que cada columna tuviera el tipo adecuado para análisis futuros. Esto incluyó convertir columnas numéricas al tipo de datos correcto usando pd.to_numeric, que facilita la realización de cálculos estadísticos.

Este proceso nos permitió tener un conjunto de datos limpio y bien estructurado, listo para el análisis estadístico.

A partir de las Estadísticas descriptivas para el conjunto de datos podemos realizar los gráficos correspondientes para las variables numéricas de Edad, puntaje en matemáticas, puntuaje en lengua y Asistencia.

Los gráficos lo podemos hacer en nuestro entorno local o cualquier plataforma que utilice para el Análisis de Datos como Python con Jupyter Notebook o RStudio

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

# Gráfico de barras de la asistencia
plt.bar(df['Asistencia'], df['Puntaje Matemáticas'])
plt.xlabel("Asistencia")
plt.ylabel("Puntaje Matemáticas")
plt.title("Relación entre asistencia y puntaje en matemáticas")
plt.show()

# Gráfico de dispersión de puntaje en matemáticas y lengua
plt.scatter(df['Puntaje Matemáticas'], df['Puntaje Lengua'])
plt.xlabel("Puntaje Matemáticas")
plt.ylabel("Puntaje Lengua")
plt.title("Relación entre puntaje en matemáticas y lengua")
plt.show()

# Histograma de la edad
plt.hist(df['Edad'])
plt.xlabel("Edad")
plt.ylabel("Frecuencia")
plt.title("Distribución de la edad")
plt.show()
```

Este código nos permitió visualizar la distribución de cada variable numérica en su conjunto de datos. Probando en Python o bibliotecas, podemos probar este código localmente para ver los gráficos.

En cuanto al uso de librería Pandas para la limpieza de los datos se debe registrar en le notebook al menos dos ejemplos de cada uno de los siguientes sucesos y las soluciones tomadas

- Datos ausentes.
- Datos en formato incorrecto
- Datos erróneos
- Datos duplicados

En cada caso agregar una descripción (celda de texto) junto con el código correspondiente para que sea clara la situación encontrada y el criterio para resolverla.

RESOLUCION

Utilizamos la librería Pandas en Python. También te proporcionamos una breve descripción para que se pueda documentar adecuadamente.

1. Datos Ausentes (Ejemplo)

Los datos ausentes son valores que faltan en el dataset, los cuales pueden afectar el análisis si no se tratan adecuadamente.

Ejemplo y Solución:

 # Ejemplo de identificación de datos ausentes

print(df.isnull().sum())

 # Solución 1: Eliminar filas con datos ausentes

df_cleaned = df.dropna()

 # Solución 2: Rellenar datos ausentes con la media (para datos numéricos)

df['columna_numerica'].fillna(df['columna_numerica'].mean(), inplace=True)

Descripción: En este ejemplo, primero identificamos cuántos valores ausentes hay en cada columna con df.isnull().sum(). Luego, mostramos dos métodos comunes para tratar con valores ausentes:

eliminando filas que contienen valores ausentes y rellenando los valores ausentes con la media de la columna.

2. Datos en Formato Incorrecto (Ejemplo)

Los datos pueden estar en un formato que no es adecuado para el análisis o la interpretación.

Ejemplo y Solución:

```
# Ejemplo de identificación de datos en formato incorrecto
print(df['fecha'].head())  # Suponiendo que 'fecha' debe ser datetime

# Solución: Convertir a formato datetime
df['fecha'] = pd.to_datetime(df['fecha'], errors='coerce')
```

Descripción: Aquí, el ejemplo muestra cómo convertir una columna que debería estar en formato datetime pero no lo está. Utilizamos pd.to_datetime para convertir la columna a datetime, usando errors='coerce' para manejar cualquier valor que no pueda ser convertido correctamente, poniéndolo como NaT (Not a Time).

3. Datos Erróneos

Los datos erróneos son aquellos que no tienen sentido dentro del contexto del dataset.

Ejemplo y Solución:

```
# Ejemplo de identificación de datos erróneos
print(df[df['edad'] < 0])  # Edades negativas no son posibles

# Solución: Corregir los datos erróneos
df = df.rename(columns={'Asistencia;;;;;Edad;Genero;Puntaje matematicas;Puntaje Lengua;Asistencia': 'Asistencia'})
df['Asistencia'] = df['Asistencia'].str.extract('(\d+);;;;;').astype(int)
df = df.rename(columns={'GÃ©nero': 'Género'})
df = df.rename(columns={'Puntaje_MatemÃ¡ticas': 'Puntaje Matemáticas'})
df = df.rename(columns={'Puntaje_Lengua': 'Puntaje Lengua'})
```

Descripción: En este ejemplo, buscamos en la columna 'edad' valores que son menores que cero, lo cual es imposible. Corregimos estos valores tomando el valor absoluto, lo cual es una suposición simplista y debe ser validada con más contexto.

4. Datos Duplicados

Los datos duplicados pueden distorsionar el análisis y deben ser identificados y tratados.

Ejemplo y Solución:

```
# Ejemplo de identificación de datos duplicados

print(df.duplicated().sum())


# Solución: Eliminar duplicados

df = df.drop_duplicates()
```

Descripción: Primero, identificamos la cantidad de filas duplicadas en el dataframe. Luego, utilizamos df.drop_duplicates() para eliminar cualquier fila duplicada, manteniendo solo la primera ocurrencia.

3- Se deberá actualizar la documentación publicada, adjuntando un resumen de las reuniones de trabajo, cómo se organizaron, roles y si debieron realizar alguna tarea no detallada en el sprint anterior. Informar si alguien miembro del equipo ya no sigue cursando.

4- Realizar las medidas descriptivas de las variables en base a su tipo. Realizar los informes según sean los datos (simétricos o asimétricos).

Roles y Organización:

1. **Líder de Proyecto RIVERA LUNA GONZALO EZEQUIEL: Coordina el equipo, asigna tareas y asegura que el proyecto se mantenga en el tiempo estimado. Organiza y lidera las reuniones de trabajo.**

2. **Analista de Datos Senior PAOLA GISELLE NAVARRO: Supervisan y revisan el análisis realizado, proponen metodologías estadísticas avanzadas y aseguran la calidad de los resultados.**
3. **Científico de Datos NAHIR DAYANA GUZMAN, SILVINA ANDREA GODOY: Se encargan de la limpieza de datos, la manipulación de los mismos y la ejecución de análisis estadísticos básicos y avanzados.**
4. **Desarrollador de Software MATEO BELTRAMONE: Implementa herramientas y scripts para automatizar procesos repetitivos y asegura la integración de los resultados en la plataforma o software del cliente.**
5. **Documentador SERGIO DANIEL ALMADA: Se encarga de actualizar la documentación técnica y de usuario, registrar las reuniones y asegurarse de que todos los entregables tengan el formato adecuado.**
6. **Especialista en Visualización de Datos YAMIL ELÍAS ORO, MARISA ROJAS: Desarrolla visualizaciones de datos claras y comprensibles para informes y presentaciones para stakeholders.**

Organización de las Reuniones:

- **Las reuniones de trabajo se realizaron semanalmente para evaluar el progreso del proyecto.**
- **En cada reunión, el líder de proyecto recopilaba actualizaciones de cada miembro y reasignaba tareas según fuera necesario.**
- **Se utilizaban herramientas de colaboración en línea para documentar el progreso y las decisiones tomadas.**
- **Cada miembro del equipo tenía roles claros pero flexibles, permitiendo adaptación según las necesidades del proyecto.**

Tareas no Detalladas en el Sprint Anterior:

- **Implementación de un nuevo método de imputación de datos ausentes descubierto durante el análisis exploratorio.**
- **Creación de una rutina de validación cruzada para asegurar la robustez de los modelos estadísticos.**

4. Medidas Descriptivas y Análisis de Asimetría

Para realizar las medidas descriptivas y clasificar los informes según la simetría de los datos, primero necesitamos calcular estas estadísticas y revisar la asimetría (skewness) de cada variable.

#Medidas Descriptivas y Análisis de Asimetría

 # Medidas de asimetría

print("\nMedidas de asimetría:")

print("Asimetría de la asistencia:", df['Asistencia'].skew())

print("Asimetría del puntaje en matemáticas:", df['Puntaje Matemáticas'].skew())

print("Asimetría del puntaje en lengua:", df['Puntaje Lengua'].skew())

print("Asimetría de la edad:", df['Edad'].skew())

Análisis según asimetría:

- **Variables con asimetría cerca de cero se pueden considerar simétricas. En estos casos, se puede usar la media y la desviación estándar para describir los datos.**
- **Variables con asimetría positiva o negativa significativa indican datos asimétricos. Para estos casos, es mejor utilizar la mediana y el rango intercuartil como medidas descriptivas.**