

```
# prompt: leer un dataset, y hacer un breve analisis de sus datos

import pandas as pd

# Reemplaza 'nombre_del_archivo.csv' con el nombre de tu archivo
df = pd.read_csv('Students_Performance_knn.csv')

# Muestra las primeras filas del DataFrame
print(df.head())

# Muestra información general sobre el DataFrame (tipos de datos, valores no nulos, etc.)
print(df.info())

# Muestra estadísticas descriptivas de las columnas numéricas
print(df.describe())

# Puedes realizar análisis más específicos según tus necesidades, por ejemplo:
# - Calcular la media de una columna: df['nombre_de_columna'].mean()
# - Contar los valores únicos en una columna: df['nombre_de_columna'].value_counts()
# - Agrupar datos y calcular estadísticas agregadas: df.groupby('nombre_de_columna').mean()
# - Crear gráficos para visualizar los datos: import matplotlib.pyplot as plt; plt.hist(df['nombre_de_columna'])
```

```
↗
gender race/ethnicity parental level of education lunch \
0 female group B bachelor's degree standard
1 female group C some college standard
2 female group B master's degree standard
3 male group A associate's degree free/reduced
4 male group C some college standard
```

```
test preparation course math score reading score writing score
0 none 72 72 74
1 completed 69 90 88
2 none 90 95 93
3 none 47 57 44
4 none 76 78 75
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 1000 entries, 0 to 999
```

```
Data columns (total 8 columns):
```

#	Column	Non-Null Count	Dtype
0	gender	1000 non-null	object
1	race/ethnicity	1000 non-null	object
2	parental level of education	1000 non-null	object
3	lunch	975 non-null	object
4	test preparation course	1000 non-null	object
5	math score	1000 non-null	int64
6	reading score	1000 non-null	int64
7	writing score	1000 non-null	int64

```
dtypes: int64(3), object(5)
```

```
memory usage: 62.6+ KB
```

```
None
```

	math score	reading score	writing score
count	1000.000000	1000.000000	1000.000000
mean	66.089000	69.169000	68.054000
std	15.163088	14.600192	15.195657
min	0.000000	17.000000	10.000000
25%	57.000000	59.000000	57.750000
50%	66.000000	70.000000	69.000000
75%	77.000000	79.000000	79.000000
max	100.000000	100.000000	100.000000

```
import matplotlib.pyplot as plt
```

```
# Datos de ejemplo (reemplaza esto con tu DataFrame real)
```

```
import pandas as pd
```

```
data = {
```

```
    'gender': ['male', 'female'],
```

```
    'math score': [70, 75],
```

```
    'reading score': [72, 78],
```

```
    'writing score': [68, 74]
```

```
}
```

```
df = pd.DataFrame(data).set_index('gender')
```

```
# Agrupa los datos por género y calcula el promedio de cada test
```

```
grouped = df.groupby('gender')[['math score', 'reading score', 'writing score']].mean()
```

```
# Crea el gráfico de columnas agrupadas
```

```

fig, ax = plt.subplots(figsize=(10, 6))

# Ancho de cada barra
width = 0.35

# Posiciones de las barras para hombres y mujeres
ind = range(len(grouped.columns))
rects1 = ax.bar(ind, grouped.loc['male'], width, color='blue', label='Male')
rects2 = ax.bar([i + width for i in ind], grouped.loc['female'], width, color='red', label='Female')

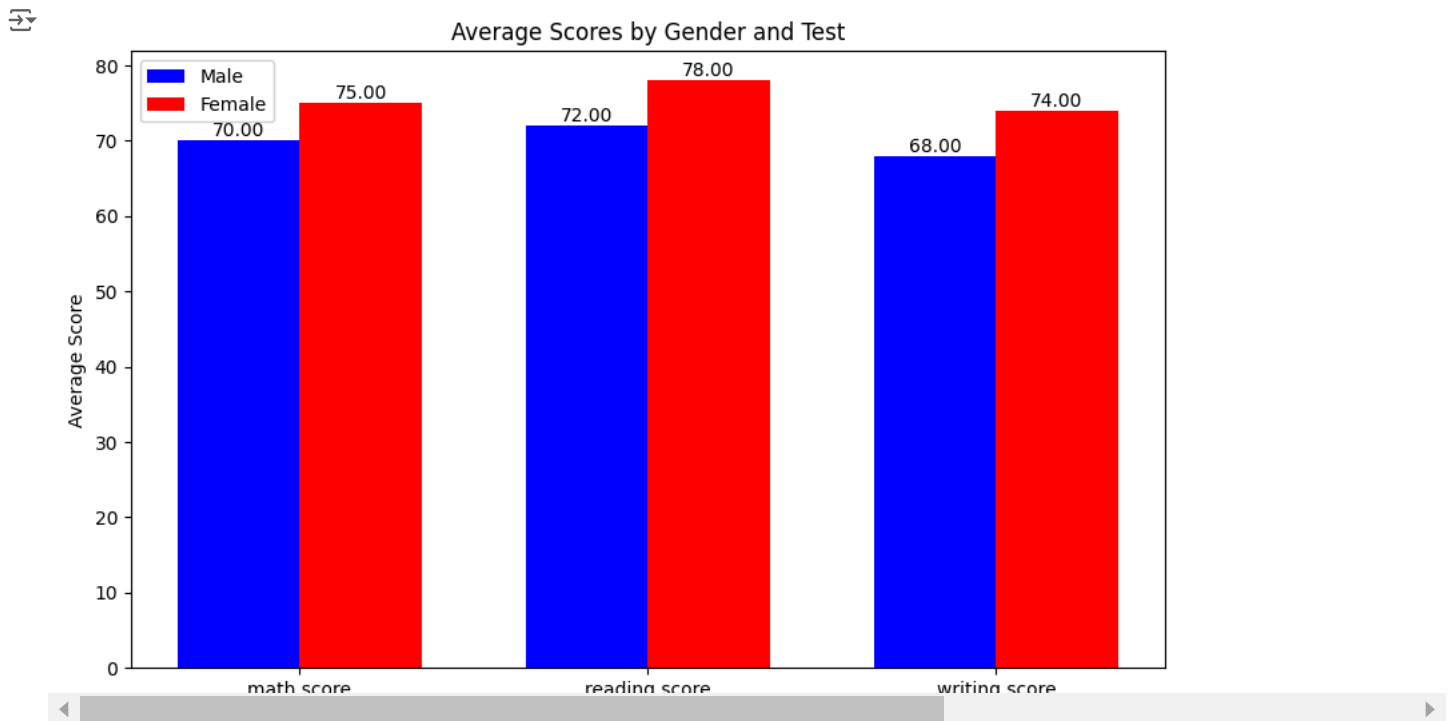
# Añadir etiquetas de valor sobre cada barra
for rect in rects1:
    height = rect.get_height()
    ax.text(rect.get_x() + rect.get_width() / 2., height,
            f'{height:.2f}', ha='center', va='bottom')

for rect in rects2:
    height = rect.get_height()
    ax.text(rect.get_x() + rect.get_width() / 2., height,
            f'{height:.2f}', ha='center', va='bottom')

# Configura las etiquetas y el título
ax.set_xticks([i + width / 2 for i in ind])
ax.set_xticklabels(grouped.columns)
ax.set_ylabel('Average Score')
ax.set_title('Average Scores by Gender and Test')
ax.legend()

plt.show()

```



```

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

# Load your original dataset
df = pd.read_csv('Students_Performance_knn.csv') # Make sure this file exists and contains the 'gender' column

# Plotting the scatter plot matrix
sns.pairplot(df, hue='gender', diag_kind='kde')
plt.suptitle('Scatter Plot Matrix: Student Performance', y=1.02)
plt.show()

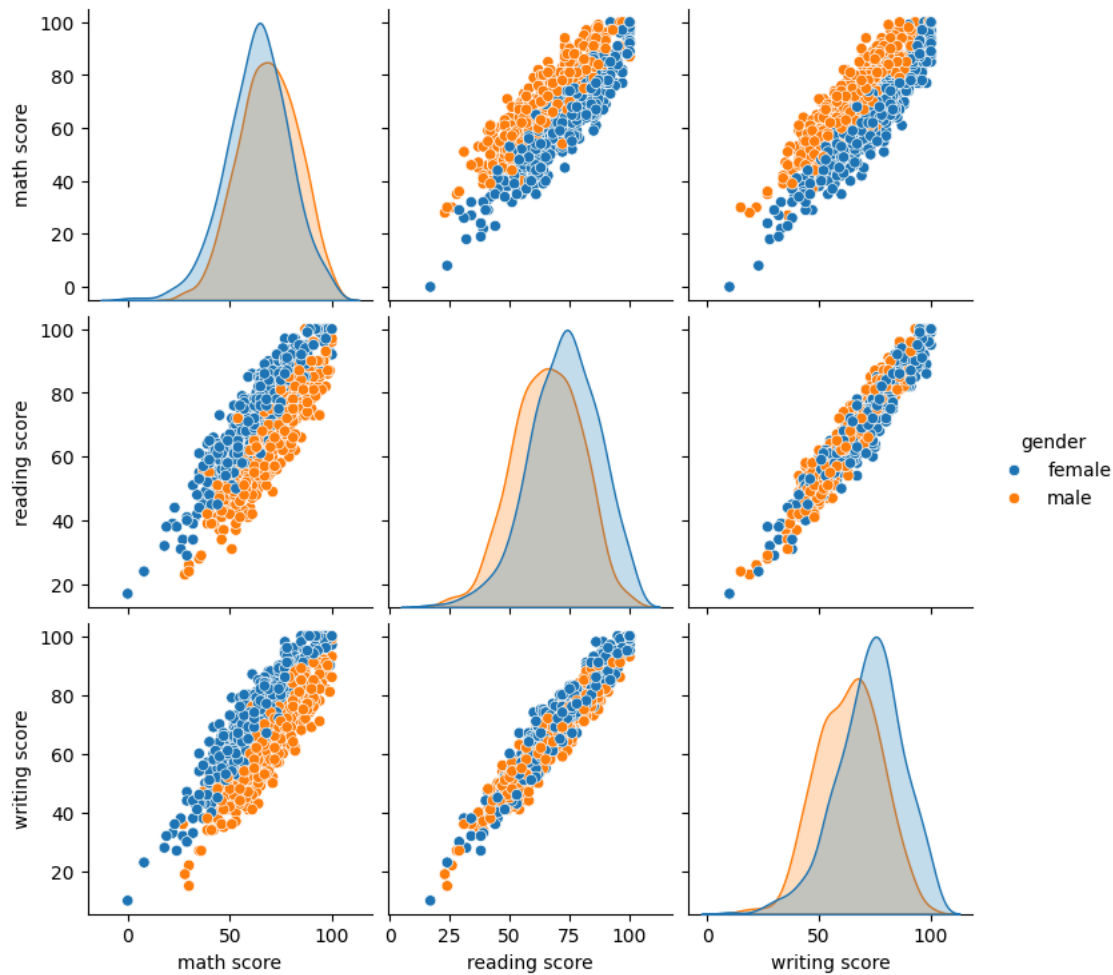
# Plotting the bar chart for average scores by parental level of education
# Use the DataFrame 'df' instead of the dictionary 'data'
avg_scores = df.groupby('parental level of education')[['math score', 'reading score', 'writing score']].mean().reset_index()
avg_scores = avg_scores.melt(id_vars='parental level of education', var_name='Subject', value_name='Score')

```

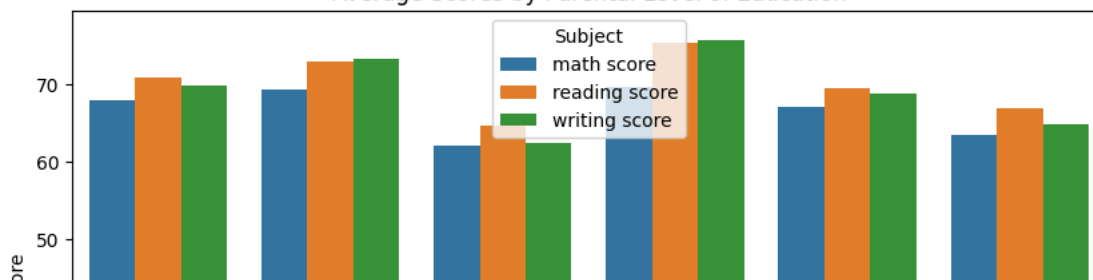
```
plt.figure(figsize=(10, 6))
sns.barplot(x='parental level of education', y='Score', hue='Subject', data=avg_scores)
plt.title('Average Scores by Parental Level of Education')
plt.xlabel('Parental Level of Education')
plt.ylabel('Average Score')
plt.xticks(rotation=45)
plt.legend(title='Subject')
plt.show()
```



Scatter Plot Matrix: Student Performance



Average Scores by Parental Level of Education



Start coding or [generate](#) with AI.

