```python
# Anwar_Siraj
# DS-VS-Batch-5 (NED)
# Q: Question 1 (20 marks)
# A dataset containing information about the sales of different
products in a retail store is available at sales_data.csv. Analyze the
dataset and identify the top-selling products, the most profitable
products, and the products with the highest customer satisfaction.
Visualize your findings using appropriate charts and graphs.

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

import pandas as pd
sales_data = pd.read_csv(r"C:\Users\binary\Downloads\Exam\
sales_data.csv", encoding='latin1')

sales_data.head()
```

```
   ORDERNUMBER  QUANTITYORDERED  PRICEEACH  ORDERLINENUMBER
SALES  \
0        10107               30      95.70                2  2871.00

1        10121               34      81.35                5  2765.90

2        10134               41      94.74                2  3884.34

3        10145               45      83.26                6  3746.70

4        10159               49     100.00               14  5205.27


         ORDERDATE   STATUS  QTR_ID  MONTH_ID  YEAR_ID  ...  \
0   2/24/2003 0:00  Shipped       1         2     2003  ...
1    5/7/2003 0:00  Shipped       2         5     2003  ...
2    7/1/2003 0:00  Shipped       3         7     2003  ...
3   8/25/2003 0:00  Shipped       3         8     2003  ...
4  10/10/2003 0:00  Shipped       4        10     2003  ...

                    ADDRESSLINE1  ADDRESSLINE2           CITY STATE  \
0        897 Long Airport Avenue           NaN            NYC    NY
1             59 rue de l'Abbaye           NaN          Reims   NaN
2  27 rue du Colonel Pierre Avia           NaN          Paris   NaN
3             78934 Hillside Dr.           NaN       Pasadena    CA
4                7734 Strong St.           NaN  San Francisco    CA

  POSTALCODE COUNTRY TERRITORY CONTACTLASTNAME CONTACTFIRSTNAME
DEALSIZE
0      10022     USA       NaN              Yu             Kwai
Small
1      51100  France      EMEA         Henriot             Paul
```

```
Small
2      75508  France      EMEA        Da Cunha            Daniel
Medium
3      90003     USA       NaN           Young             Julie
Medium
4        NaN     USA       NaN           Brown             Julie
Medium

[5 rows x 25 columns]
```

```
sales_data.columns
```

```
Index(['ORDERNUMBER', 'QUANTITYORDERED', 'PRICEEACH',
'ORDERLINENUMBER',
       'SALES', 'ORDERDATE', 'STATUS', 'QTR_ID', 'MONTH_ID',
'YEAR_ID',
       'PRODUCTLINE', 'MSRP', 'PRODUCTCODE', 'CUSTOMERNAME', 'PHONE',
       'ADDRESSLINE1', 'ADDRESSLINE2', 'CITY', 'STATE', 'POSTALCODE',
       'COUNTRY', 'TERRITORY', 'CONTACTLASTNAME', 'CONTACTFIRSTNAME',
       'DEALSIZE'],
      dtype='object')
```

```
import datetime

top_selling_products = sales_data.groupby('PRODUCTCODE')
['QUANTITYORDERED'].sum().sort_values(ascending=False).head(10)
print("\nTop Selling Products:")
print(top_selling_products)
```

```
Top Selling Products:
PRODUCTCODE
S18_3232     1774
S24_3856     1052
S18_4600     1031
S700_4002    1029
S12_4473     1024
S24_3949     1008
S50_1341      999
S18_1097      999
S18_2432      998
S18_1342      997
Name: QUANTITYORDERED, dtype: int64
```

```
sales_data['PROFIT'] = sales_data['SALES'] -
(sales_data['QUANTITYORDERED'] * sales_data['PRICEEACH'])
most_profitable_products = sales_data.groupby('PRODUCTCODE')
['PROFIT'].sum().sort_values(ascending=False).head(10)
print("\nMost Profitable Products:")
print(most_profitable_products)
```

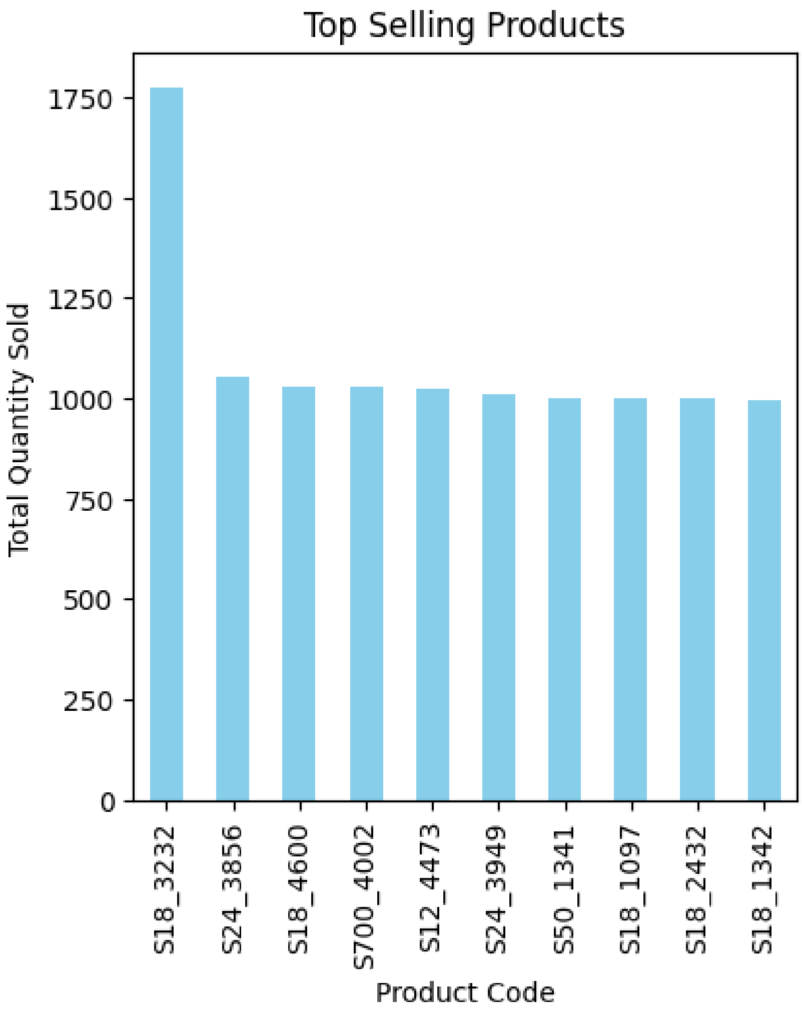
```
Most Profitable Products:
PRODUCTCODE
S18_3232    112218.79
S10_1949     94973.03
S10_4698     79488.77
S12_1108     77514.82
S18_2238     58323.95
S12_1099     57049.58
S12_3891     57042.11
S18_2795     49280.30
S18_1749     48252.66
S18_1662     47765.04
Name: PROFIT, dtype: float64
```

```python
customer_satisfaction = sales_data.groupby('PRODUCTCODE')
['SALES'].mean().sort_values(ascending=False).head(10)
print("\nProducts with Highest Customer Satisfaction:")
print(customer_satisfaction)
```

```
Products with Highest Customer Satisfaction:
PRODUCTCODE
S10_1949    6824.036786
S10_4698    6553.887308
S12_1108    6484.050769
S18_1749    5786.837273
S18_2238    5726.812963
S12_3891    5589.693846
S18_3232    5543.181154
S12_1099    5487.080400
S12_2823    5384.852308
S18_1662    5362.383462
Name: SALES, dtype: float64
```

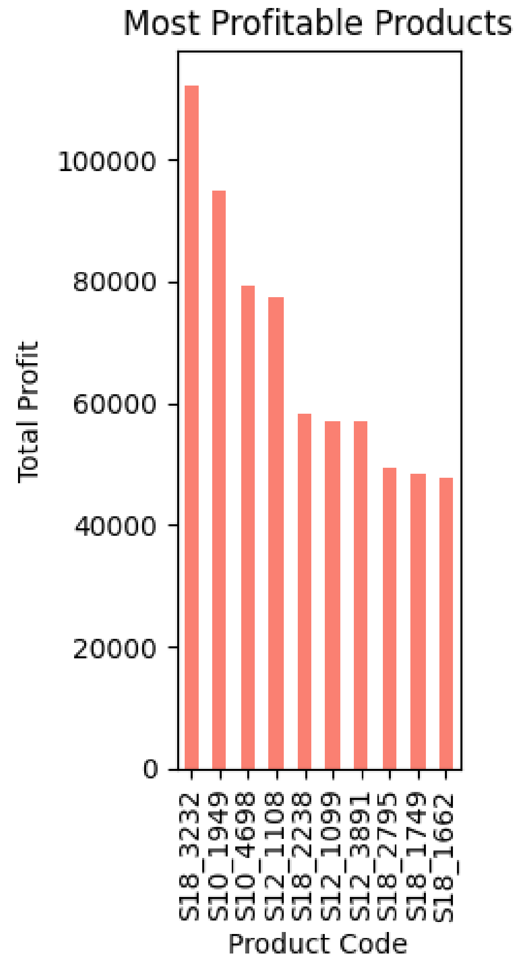
```python
plt.figure(figsize=(15, 5))
plt.subplot(1, 3, 1)
top_selling_products.plot(kind='bar', color='skyblue')
plt.title('Top Selling Products')
plt.xlabel('Product Code')
plt.ylabel('Total Quantity Sold')
```

```
Text(0, 0.5, 'Total Quantity Sold')
```

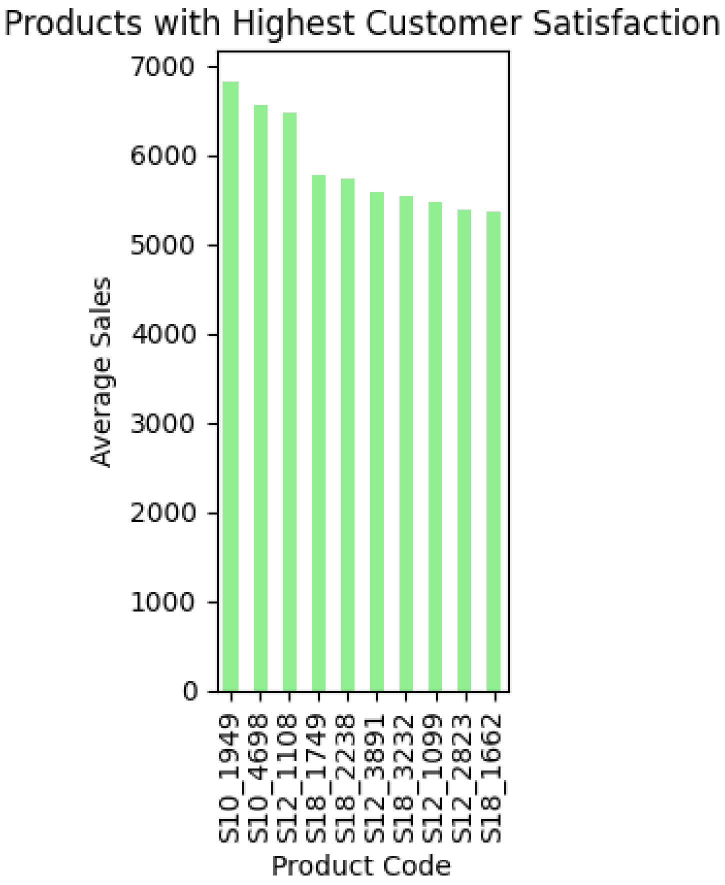


```
plt.subplot(1, 3, 2)
most_profitable_products.plot(kind='bar', color='salmon')
plt.title('Most Profitable Products')
plt.xlabel('Product Code')
plt.ylabel('Total Profit')
```

```
Text(0, 0.5, 'Total Profit')
```



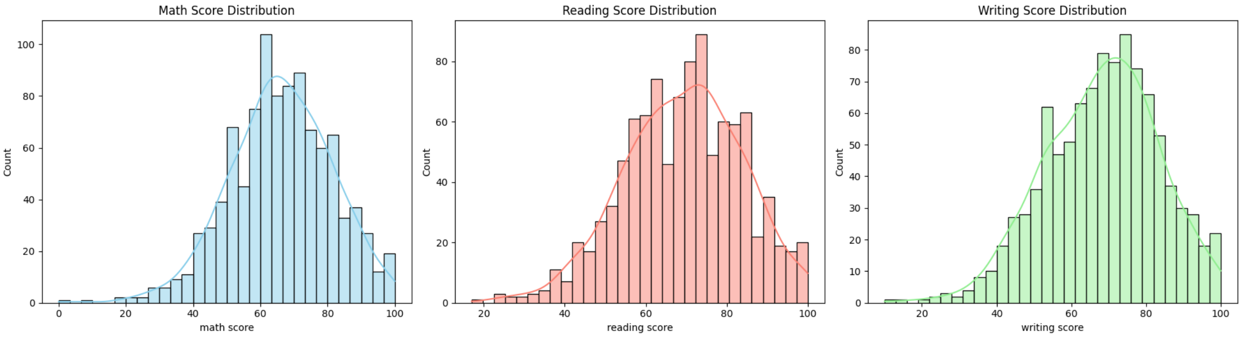
```
plt.subplot(1, 3, 3)
customer_satisfaction.plot(kind='bar', color='lightgreen')
plt.title('Products with Highest Customer Satisfaction')
plt.xlabel('Product Code')
plt.ylabel('Average Sales')

plt.tight_layout()
plt.show()
```



```
# Q2 : Question 2 (20 marks)

A dataset containing information about the performance of students in
a school is available at Performance.csv. Analyze the dataset and
identify the factors that contribute to student success. Visualize
your findings using appropriate charts and graphs.

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

df = pd.read_csv(r"C:\Users\binary\Downloads\Exam\Performance.csv",
encoding='Latin-1')

df.head()
```

```
   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   
```

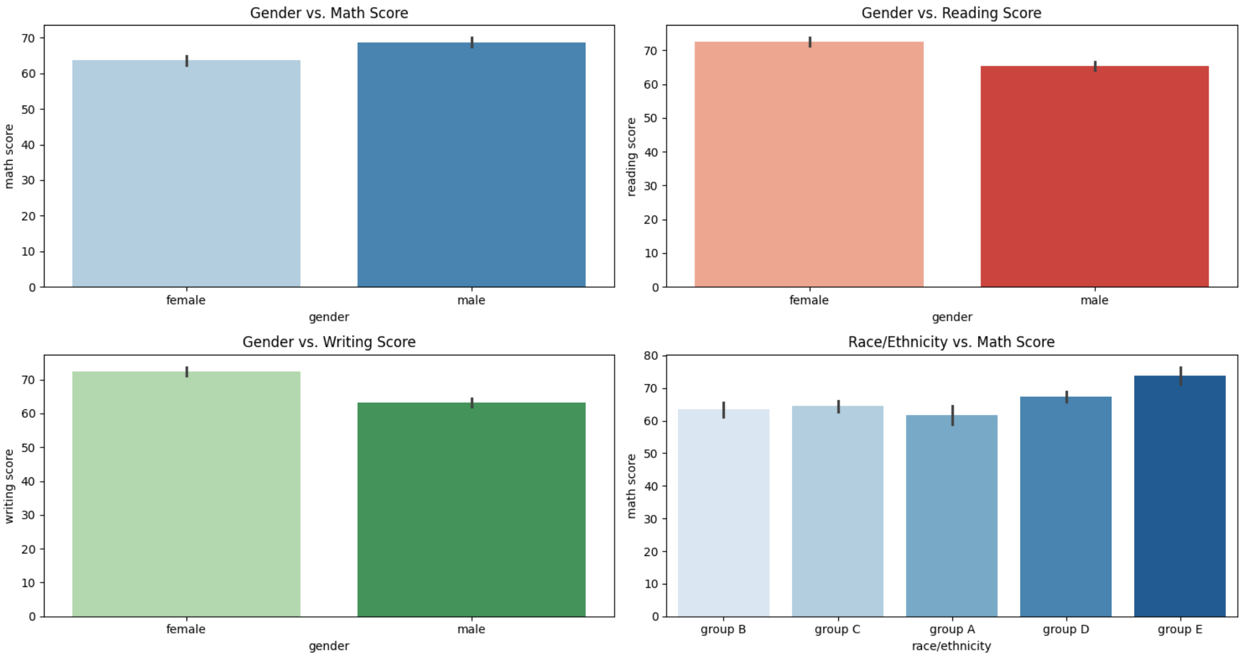
```
  test preparation course  math score  reading score  writing score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none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

# Visualize the distribution of scores
plt.figure(figsize=(18, 5))

# Math Score Distribution
plt.subplot(1, 3, 1)
sns.histplot(df['math score'], bins=30, kde=True, color='skyblue')
plt.title('Math Score Distribution')

# Reading Score Distribution
plt.subplot(1, 3, 2)
sns.histplot(df['reading score'], bins=30, kde=True, color='salmon')
plt.title('Reading Score Distribution')

# Writing Score Distribution
plt.subplot(1, 3, 3)
sns.histplot(df['writing score'], bins=30, kde=True,
color='lightgreen')
plt.title('Writing Score Distribution')

plt.tight_layout()
plt.show()
```



```
# Visualize factors influencing student success
plt.figure(figsize=(15, 8))

# Gender vs. Average Scores
plt.subplot(2, 2, 1)
sns.barplot(x='gender', y='math score', data=df, palette='Blues')
plt.title('Gender vs. Math Score')

plt.subplot(2, 2, 2)
sns.barplot(x='gender', y='reading score', data=df, palette='Reds')
```

```python
plt.title('Gender vs. Reading Score')

plt.subplot(2, 2, 3)
sns.barplot(x='gender', y='writing score', data=df, palette='Greens')
plt.title('Gender vs. Writing Score')

# Race/Ethnicity vs. Average Scores
plt.subplot(2, 2, 4)
sns.barplot(x='race/ethnicity', y='math score', data=df,
palette='Blues')
plt.title('Race/Ethnicity vs. Math Score')

plt.tight_layout()
plt.show()
```

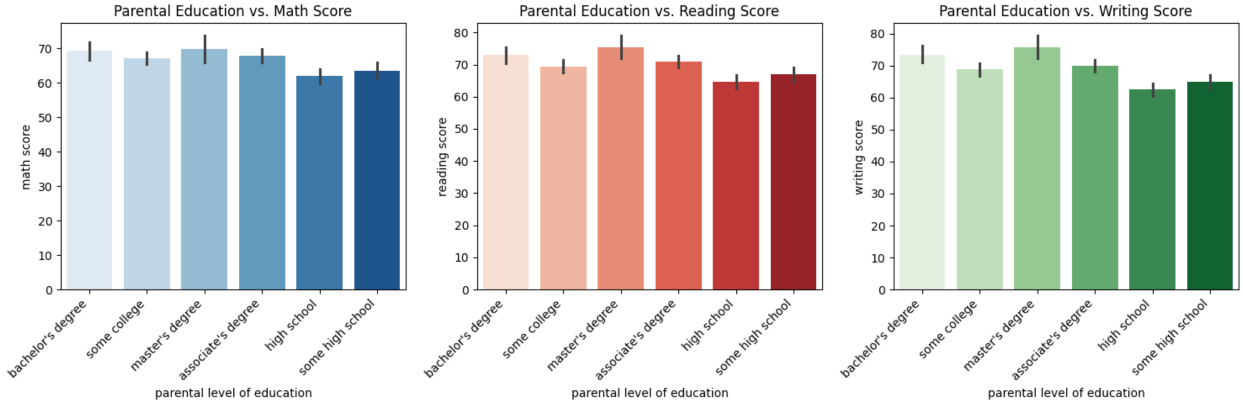
```
C:\Users\binary\AppData\Local\Temp\ipykernel_520\1063706735.py:6:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(x='gender', y='math score', data=df, palette='Blues')
C:\Users\binary\AppData\Local\Temp\ipykernel_520\1063706735.py:10:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(x='gender', y='reading score', data=df, palette='Reds')
C:\Users\binary\AppData\Local\Temp\ipykernel_520\1063706735.py:14:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(x='gender', y='writing score', data=df,
palette='Greens')
C:\Users\binary\AppData\Local\Temp\ipykernel_520\1063706735.py:19:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(x='race/ethnicity', y='math score', data=df,
palette='Blues')
```



```python
# Parental Level of Education vs. Average Scores
plt.figure(figsize=(15, 5))

plt.subplot(1, 3, 1)
sns.barplot(x='parental level of education', y='math score', data=df,
palette='Blues')
plt.title('Parental Education vs. Math Score')
plt.xticks(rotation=45, ha='right')

plt.subplot(1, 3, 2)
sns.barplot(x='parental level of education', y='reading score',
data=df, palette='Reds')
plt.title('Parental Education vs. Reading Score')
plt.xticks(rotation=45, ha='right')

plt.subplot(1, 3, 3)
sns.barplot(x='parental level of education', y='writing score',
data=df, palette='Greens')
plt.title('Parental Education vs. Writing Score')
plt.xticks(rotation=45, ha='right')

plt.tight_layout()
plt.show()
```

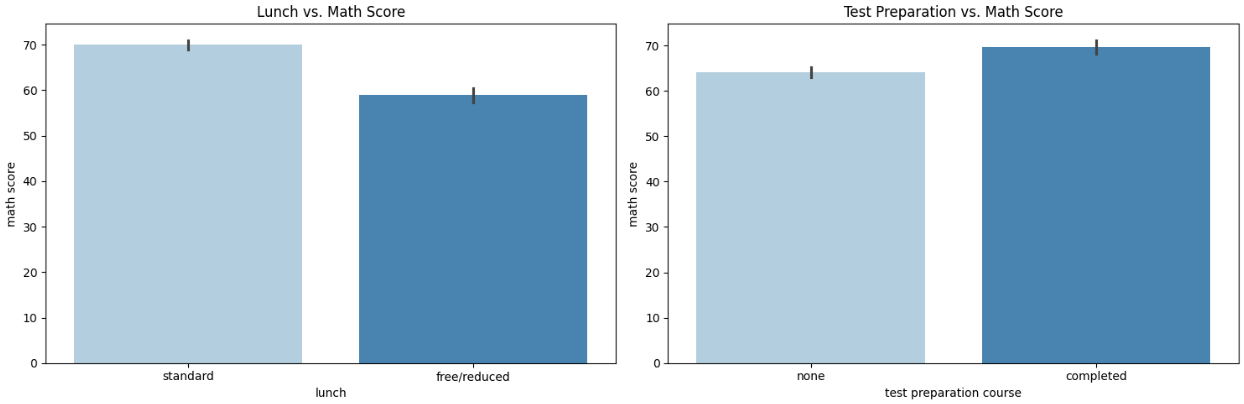
```
C:\Users\binary\AppData\Local\Temp\ipykernel_520\447597633.py:5:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.
```

```
  sns.barplot(x='parental level of education', y='math score',
data=df, palette='Blues')
C:\Users\binary\AppData\Local\Temp\ipykernel_520\447597633.py:10:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(x='parental level of education', y='reading score',
data=df, palette='Reds')
C:\Users\binary\AppData\Local\Temp\ipykernel_520\447597633.py:15:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(x='parental level of education', y='writing score',
data=df, palette='Greens')
```



```
# Lunch and Test Preparation vs. Average Scores
plt.figure(figsize=(15, 5))

plt.subplot(1, 2, 1)
sns.barplot(x='lunch', y='math score', data=df, palette='Blues')
plt.title('Lunch vs. Math Score')

plt.subplot(1, 2, 2)
sns.barplot(x='test preparation course', y='math score', data=df,
palette='Blues')
plt.title('Test Preparation vs. Math Score')

plt.tight_layout()
plt.show()
```

```
C:\Users\binary\AppData\Local\Temp\ipykernel_520\3406916652.py:5:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(x='lunch', y='math score', data=df, palette='Blues')
C:\Users\binary\AppData\Local\Temp\ipykernel_520\3406916652.py:9:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(x='test preparation course', y='math score', data=df,
palette='Blues')
```



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

weather = pd.read_csv(r"C:\Users\binary\Downloads\Exam\
weatherHistory.csv", encoding='Latin-1')

weather.head()

# Visualize temperature trends over time
plt.figure(figsize=(15, 5))
sns.lineplot(x='Formatted Date', y='Temperature (C)', data=weather)
plt.title('Temperature Trends Over Time')
plt.xlabel('Date')
plt.ylabel('Temperature (C)')
plt.show()

# Visualize precipitation trends over time
plt.figure(figsize=(15, 5))
```

```python
sns.lineplot(x='Formatted Date', y='Precip Type', data=df)
plt.title('Precipitation Trends Over Time')
plt.xlabel('Date')
plt.ylabel('Precipitation Type')
plt.show()

# Visualize humidity trends over time
plt.figure(figsize=(15, 5))
sns.lineplot(x='Formatted Date', y='Humidity', data=df)
plt.title('Humidity Trends Over Time')
plt.xlabel('Date')
plt.ylabel('Humidity')
plt.show()
```