Data Visualization

PGD Batch - 5

Atif Salam

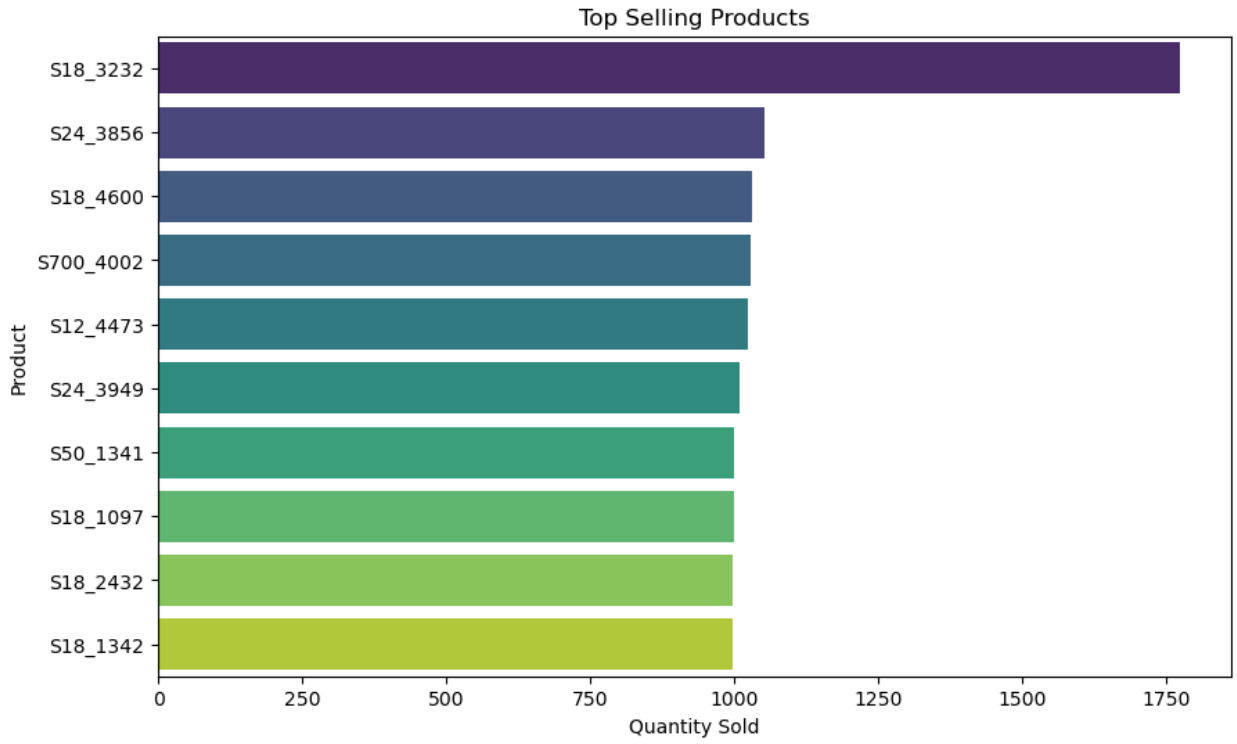
CNIC: 42201-9446591-3

```
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

path = r"C:\Users\atif\Dropbox\Data Science NED\Data Visualization\
Final Paper\Hybrid Exam Paper\sales_data.csv"
sales = pd.read_csv(path, encoding="latin1",
parse_dates=["ORDERDATE"])
sales.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 2003-02-24  Shipped       1         2     2003  ...
1 2003-05-07  Shipped       2         5     2003  ...
2 2003-07-01  Shipped       3         7     2003  ...
3 2003-08-25  Shipped       3         8     2003  ...
4 2003-10-10  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]
```

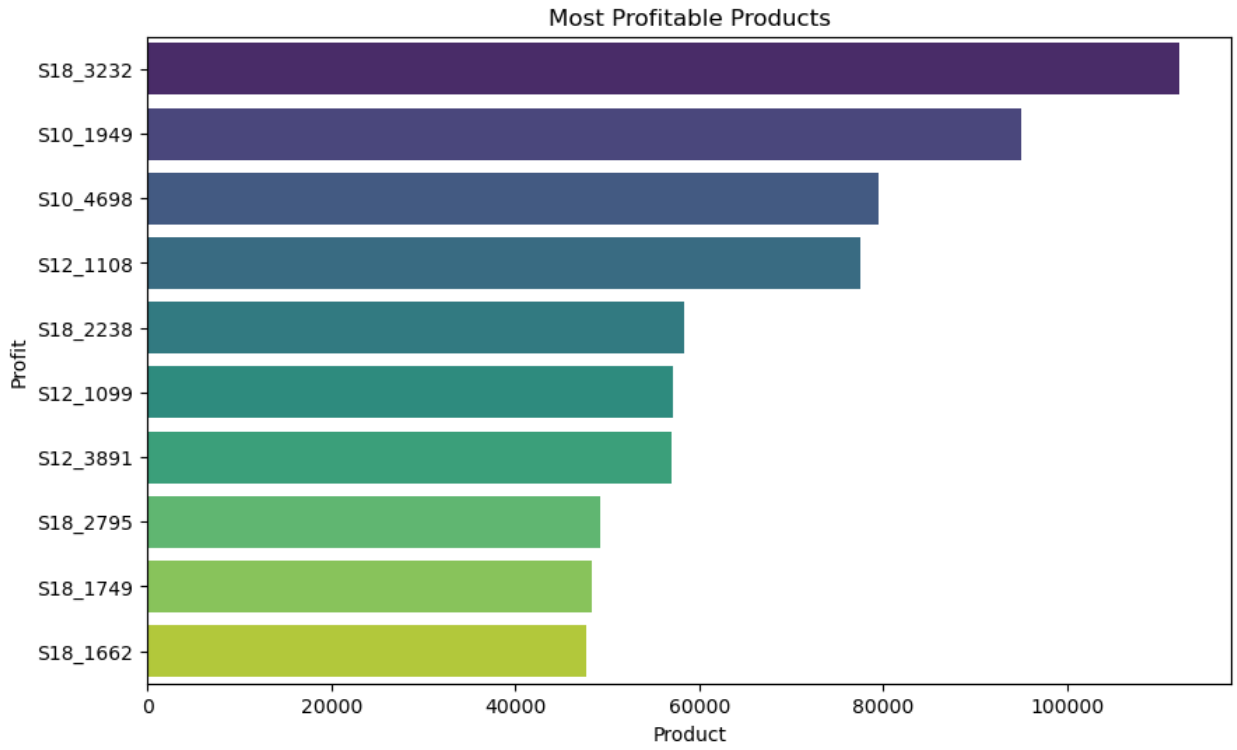
Top Selling Products

```
top_selling_products = sales.groupby("PRODUCTCODE")
["QUANTITYORDERED"].sum().sort_values(ascending=False).head(10)

plt.figure(figsize=(10, 6))
sns.barplot(x=top_selling_products.values,
y=top_selling_products.index, palette="viridis")
plt.title("Top Selling Products")
plt.xlabel("Quantity Sold")
plt.ylabel("Product")
plt.show()
```



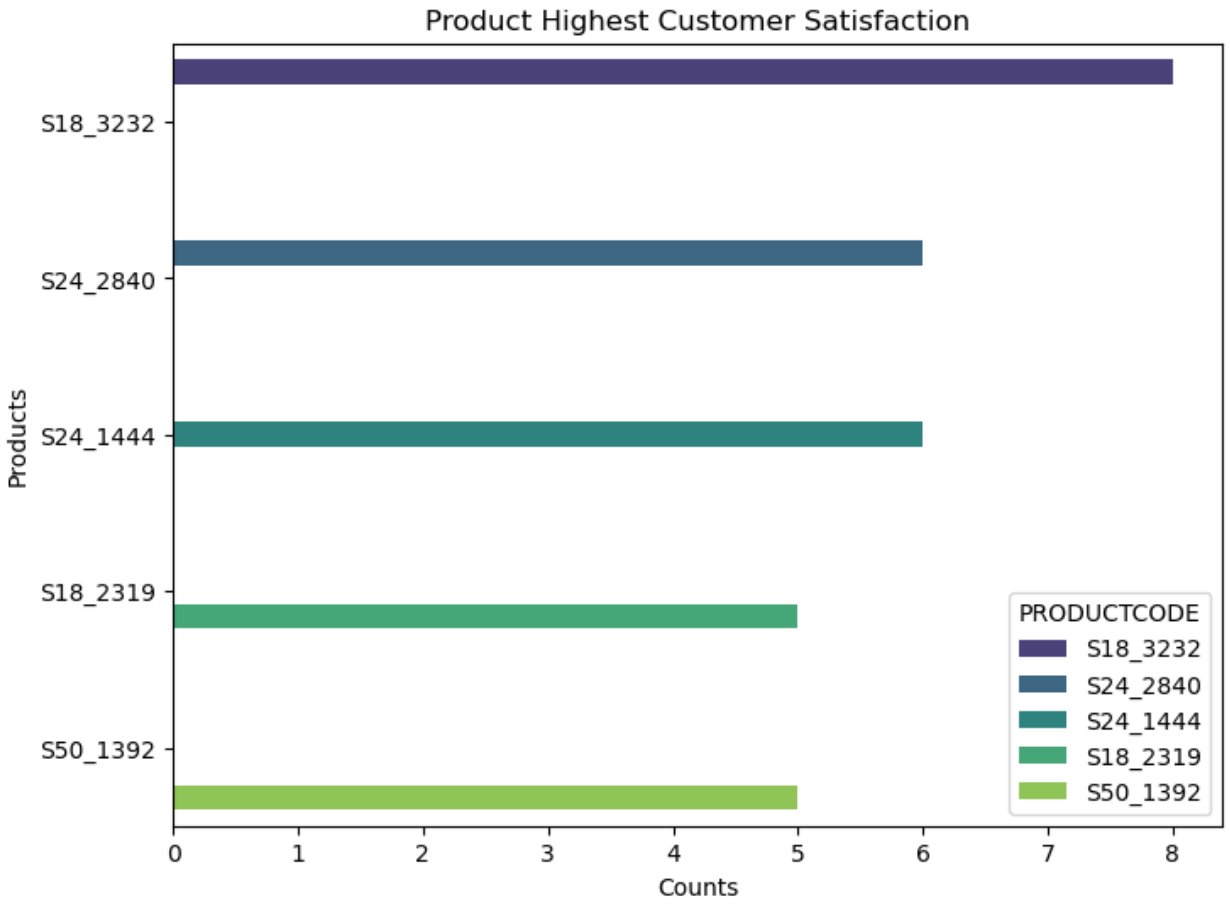
Most Profitable Products

```
sales["PROFIT"] = sales["SALES"] - (sales["QUANTITYORDERED"] *
sales["PRICEEACH"])
most_profitable_products = sales.groupby("PRODUCTCODE")
["PROFIT"].sum().sort_values(ascending=False).head(10)

plt.figure(figsize=(10, 6))
sns.barplot(x=most_profitable_products.values,
y=most_profitable_products.index, palette="viridis")
plt.title("Most Profitable Products")
plt.xlabel("Product")
plt.ylabel("Profit")
plt.show()
```



Product Highest Customer Satisfaction

```
sales["COMPLETENAME"] =
pd.concat([sales["CONTACTFIRSTNAME"].astype(str),
sales["CONTACTLASTNAME"]], axis=1).agg(' '.join, axis=1)
high_cust_satis = sales.groupby("PRODUCTCODE")
["COMPLETENAME"].value_counts().sort_values(ascending=False).head(5)
high_cust_satis = high_cust_satis.reset_index(name="COUNTS")

plt.figure(figsize=(8, 6))
sns.barplot(x="COUNTS", y="PRODUCTCODE", hue="PRODUCTCODE",
data=high_cust_satis, palette="viridis")
plt.title("Product Highest Customer Satisfaction")
plt.xlabel("Counts")
plt.ylabel("Products")
plt.show()
```



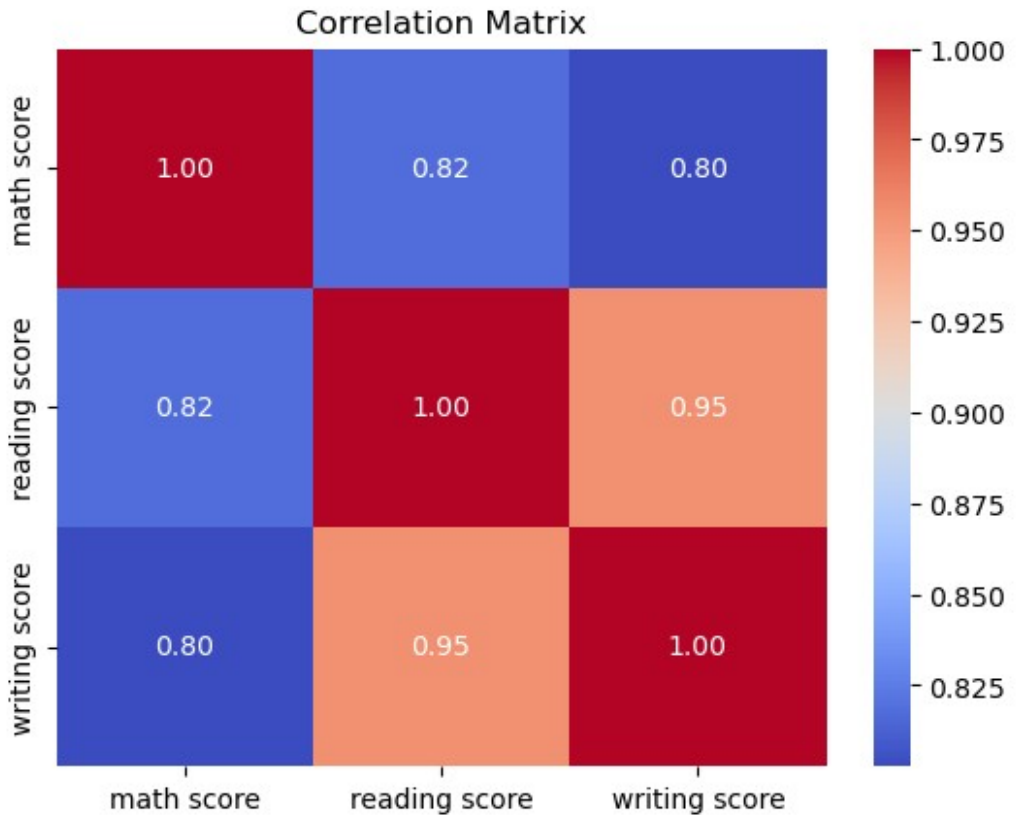

```
Question 2 (20 marks)

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.

path = r"C:\Users\atif\Dropbox\Data Science NED\Data Visualization\
Final Paper\Hybrid Exam Paper\Performance.csv"
performance = pd.read_csv(path)
performance.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

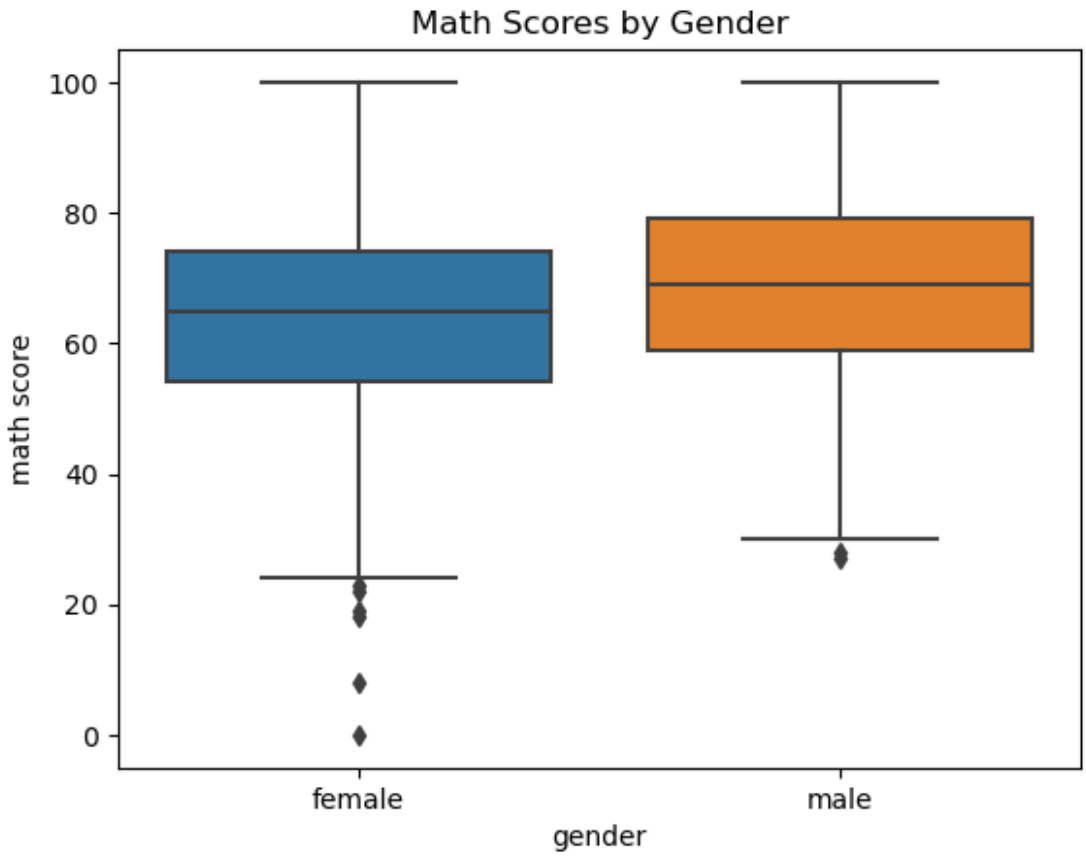
Correlation matrix

```
correlation_matrix = performance.corr()
sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm",
fmt=".2f")
plt.title("Correlation Matrix")
plt.show()
```



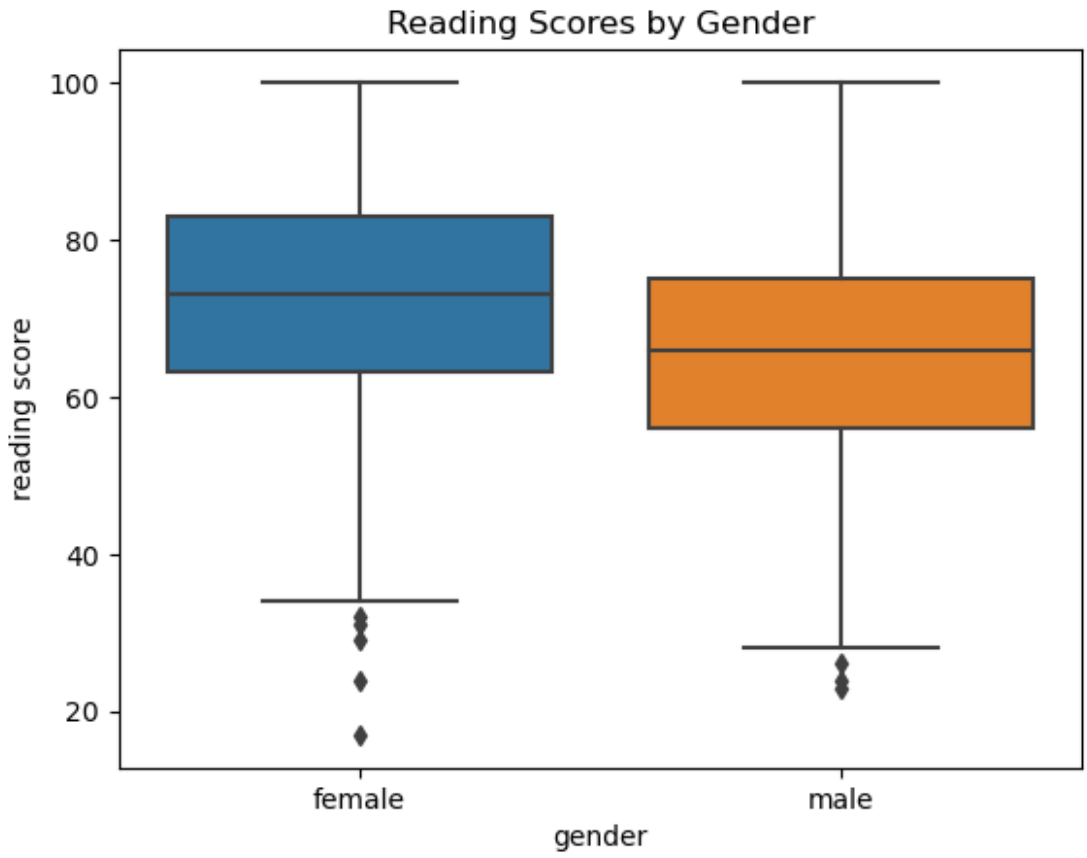
Visualize scores by gender

```
sns.boxplot(x='gender', y='math score', data=performance)
plt.title("Math Scores by Gender")
plt.show()
```



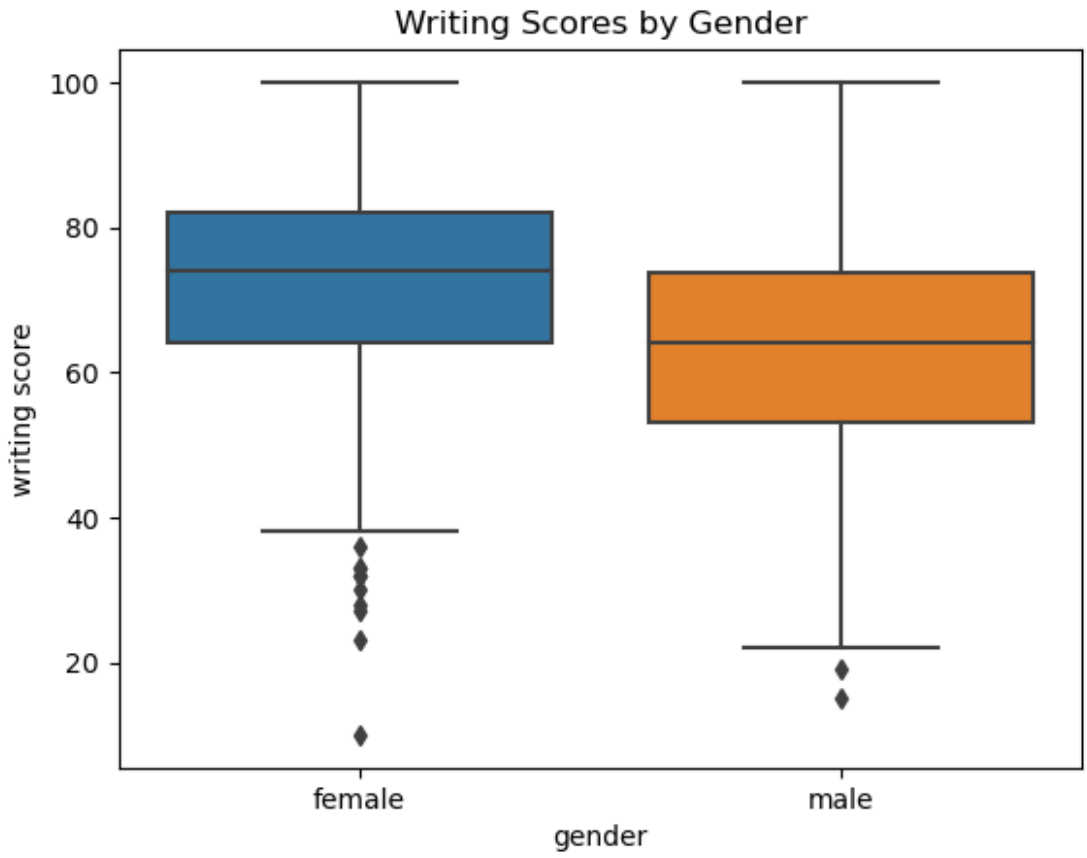
Reading Scores by Gender

```
sns.boxplot(x='gender', y='reading score', data=performance)
plt.title("Reading Scores by Gender")
plt.show()
```



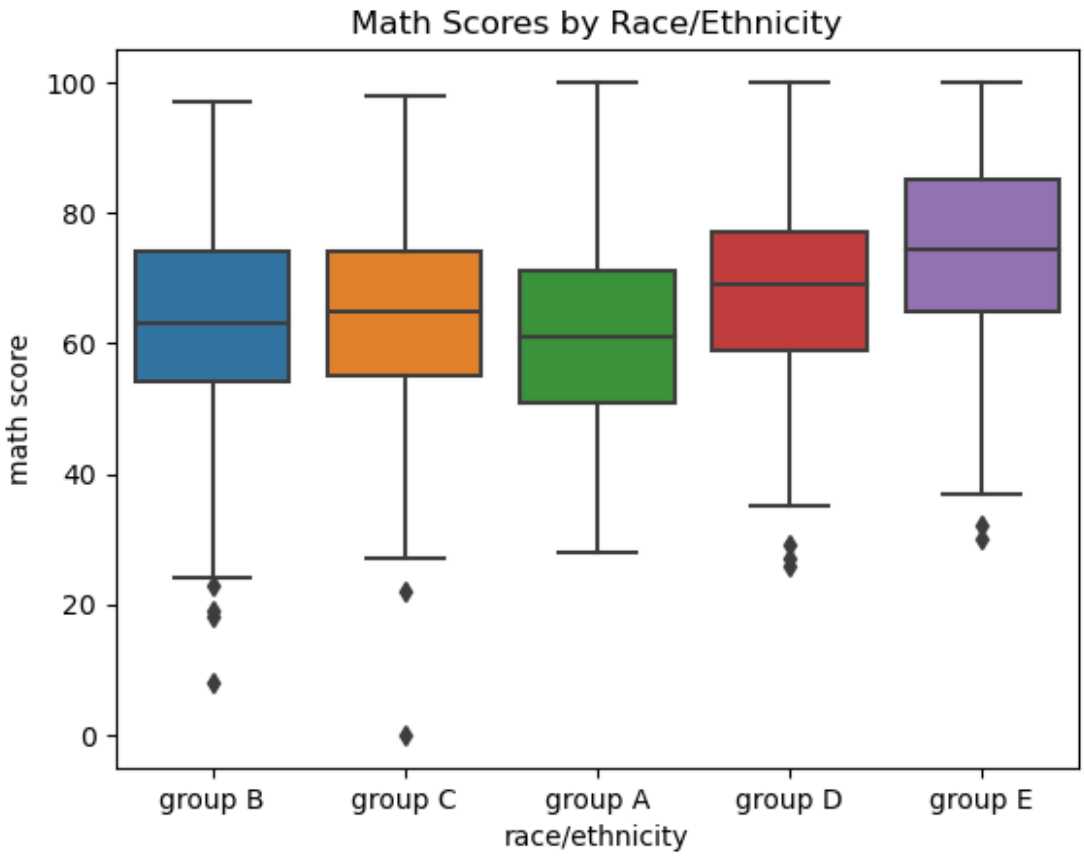
Writing Scores by Gender

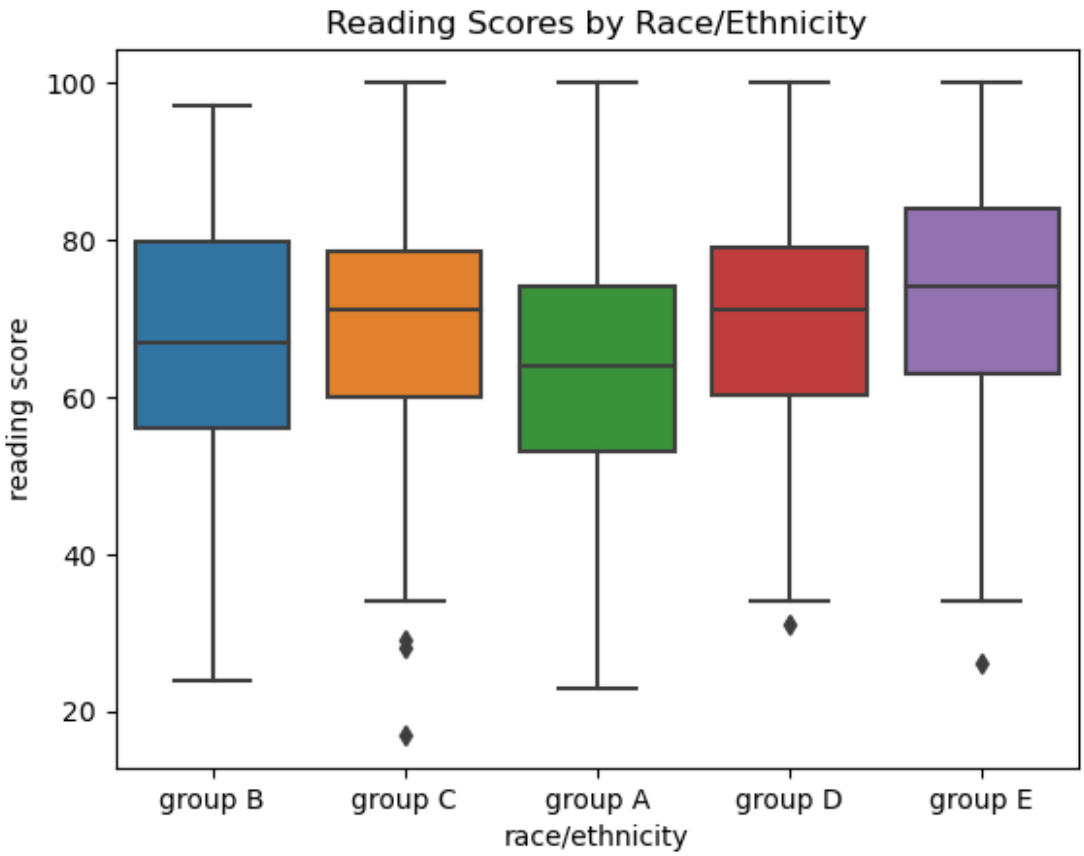
```
sns.boxplot(x='gender', y='writing score', data=performance)
plt.title("Writing Scores by Gender")
plt.show()
```

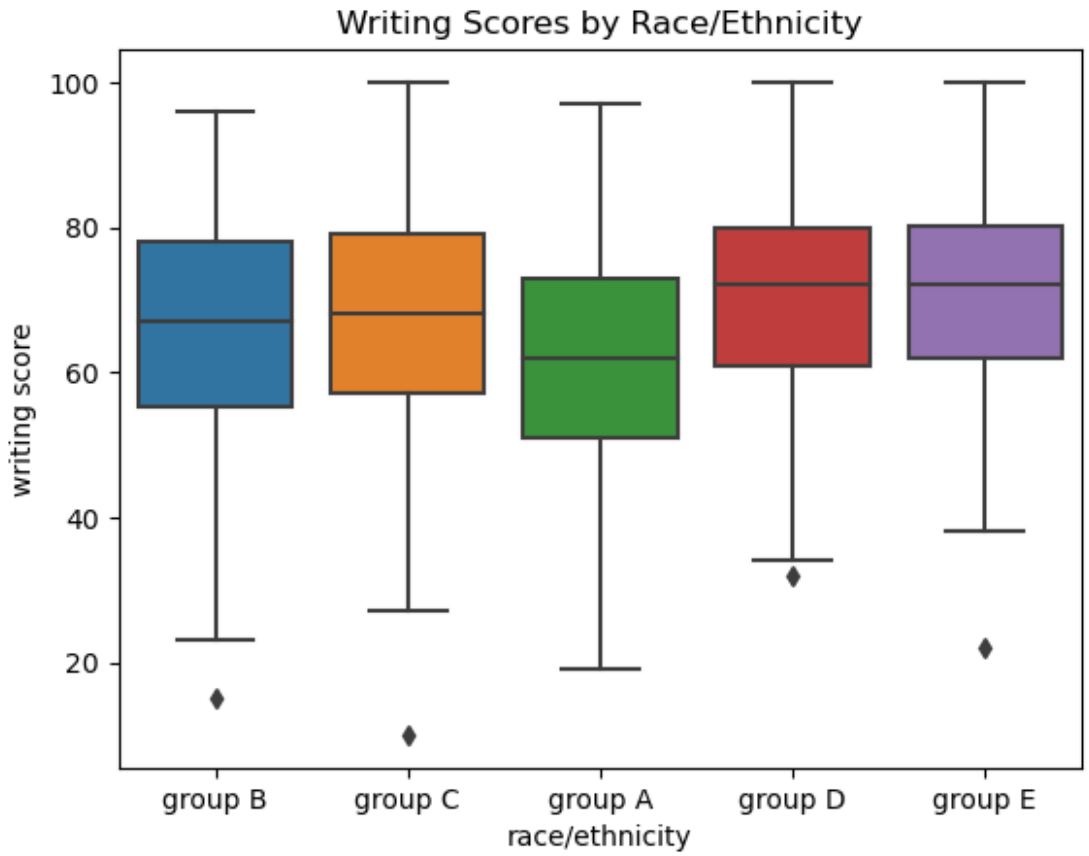


Visualize scores by race/ethnicity

```
sns.boxplot(x='race/ethnicity', y='math score', data=performance)
plt.title("Math Scores by Race/Ethnicity")
plt.show()

sns.boxplot(x='race/ethnicity', y='reading score', data=performance)
plt.title("Reading Scores by Race/Ethnicity")
plt.show()

sns.boxplot(x='race/ethnicity', y='writing score', data=performance)
plt.title("Writing Scores by Race/Ethnicity")
plt.show()
```

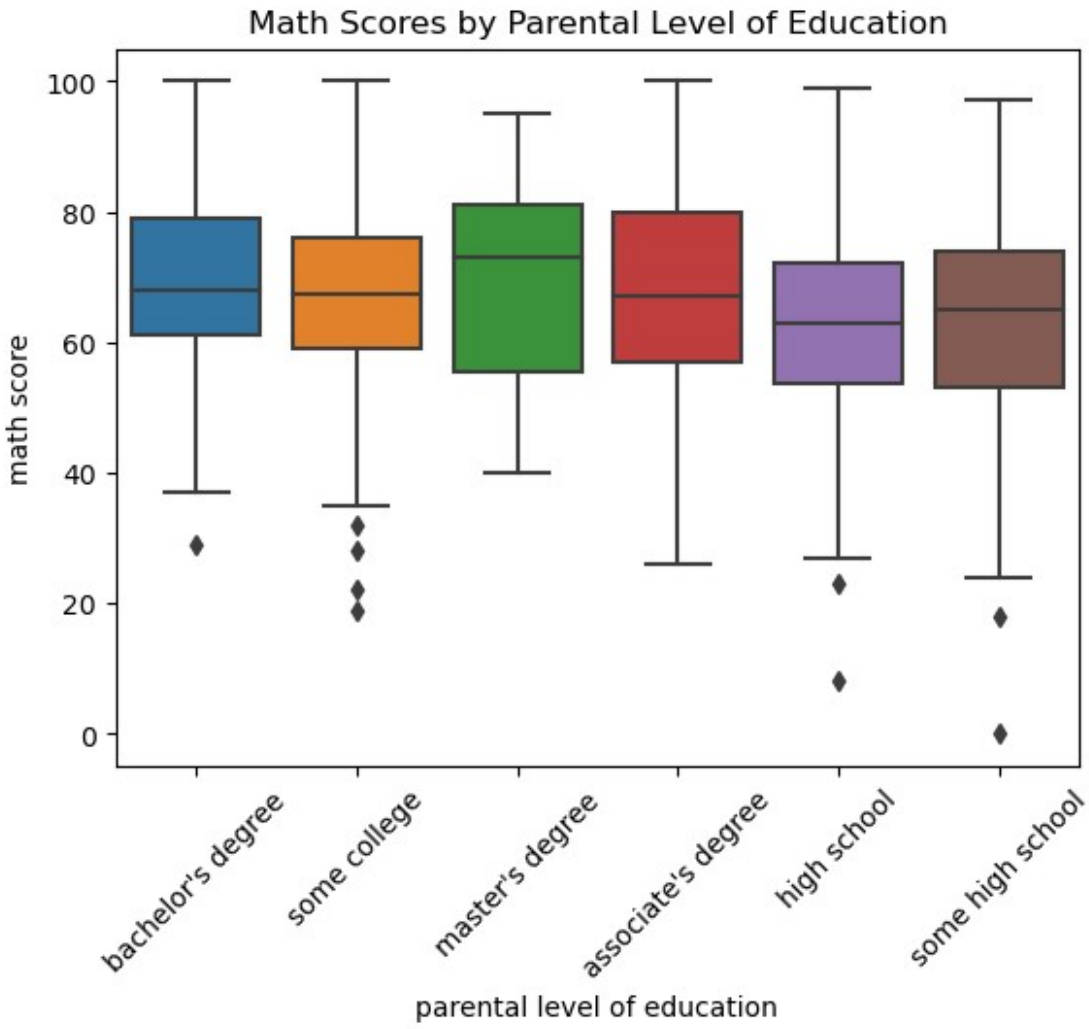

Math Scores by Race/Ethnicity
math score
100
80
60
40
20
0
group B
group C
group A
group D
group E
race/ethnicity

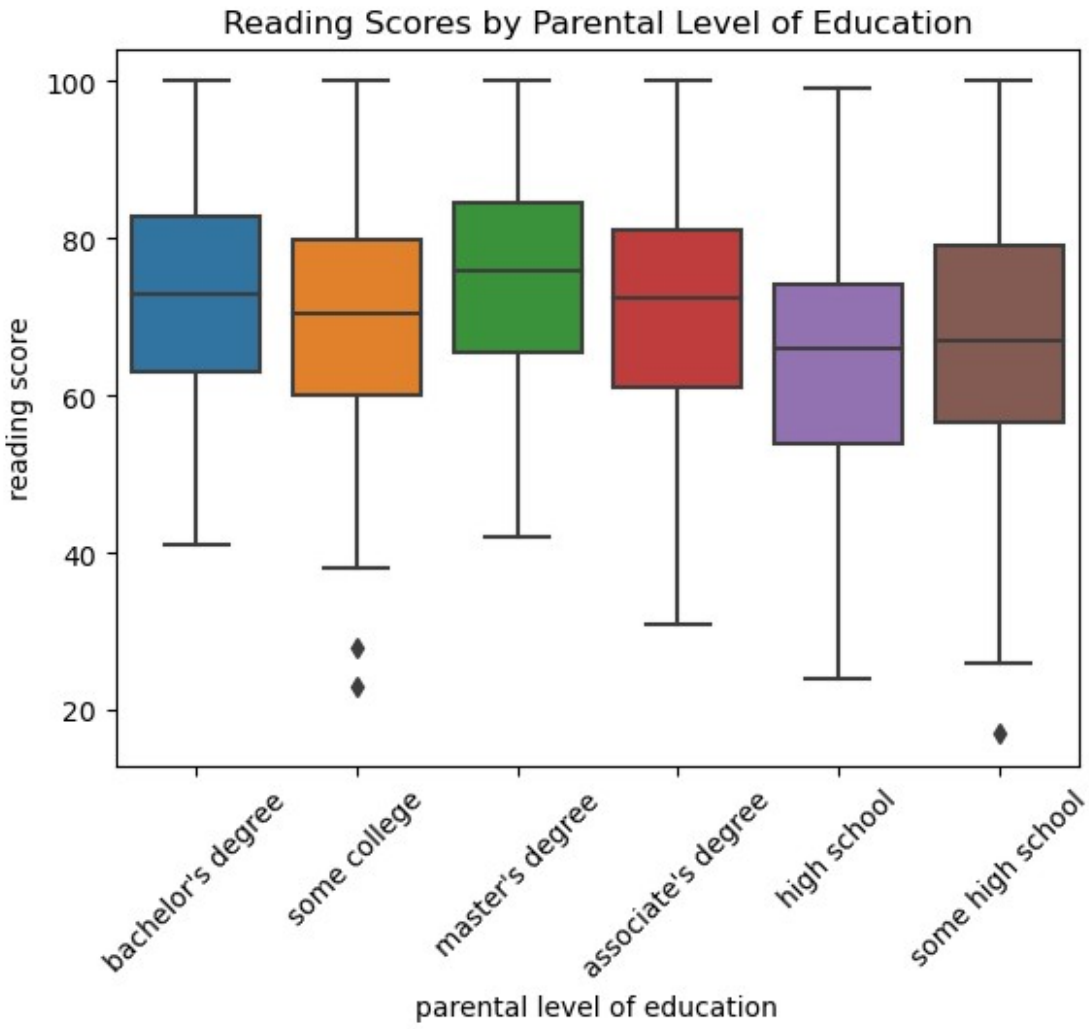

Reading Scores by Race/Ethnicity
reading score
100
80
60
40
20
group B
group C
group A
group D
group E
race/ethnicity

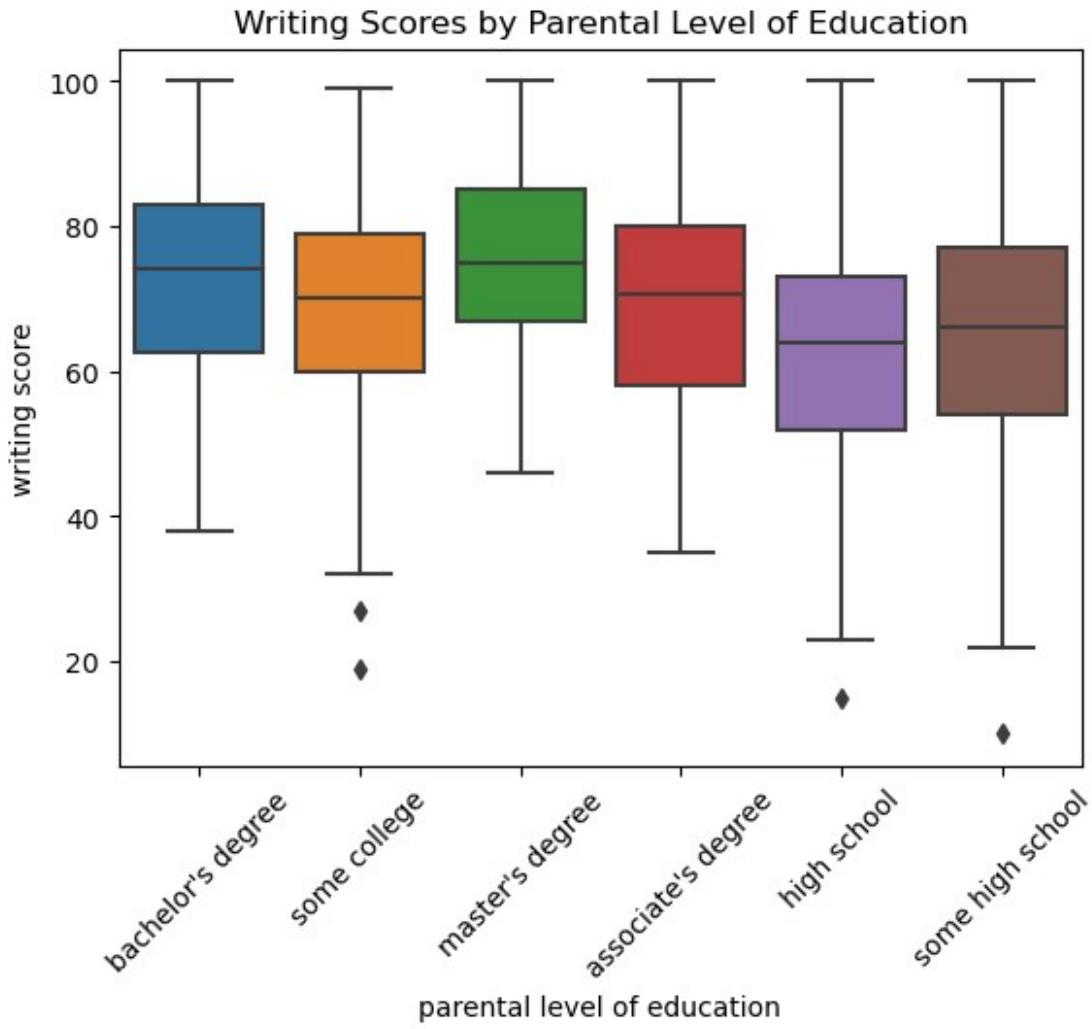


Visualize scores by parental level of education

```
sns.boxplot(x='parental level of education', y='math score',
data=performance)
plt.title("Math Scores by Parental Level of Education")
plt.xticks(rotation=45)
plt.show()

sns.boxplot(x='parental level of education', y='reading score',
data=performance)
plt.title("Reading Scores by Parental Level of Education")
plt.xticks(rotation=45)
plt.show()

sns.boxplot(x='parental level of education', y='writing score',
data=performance)
plt.title("Writing Scores by Parental Level of Education")
plt.xticks(rotation=45)
plt.show()
```

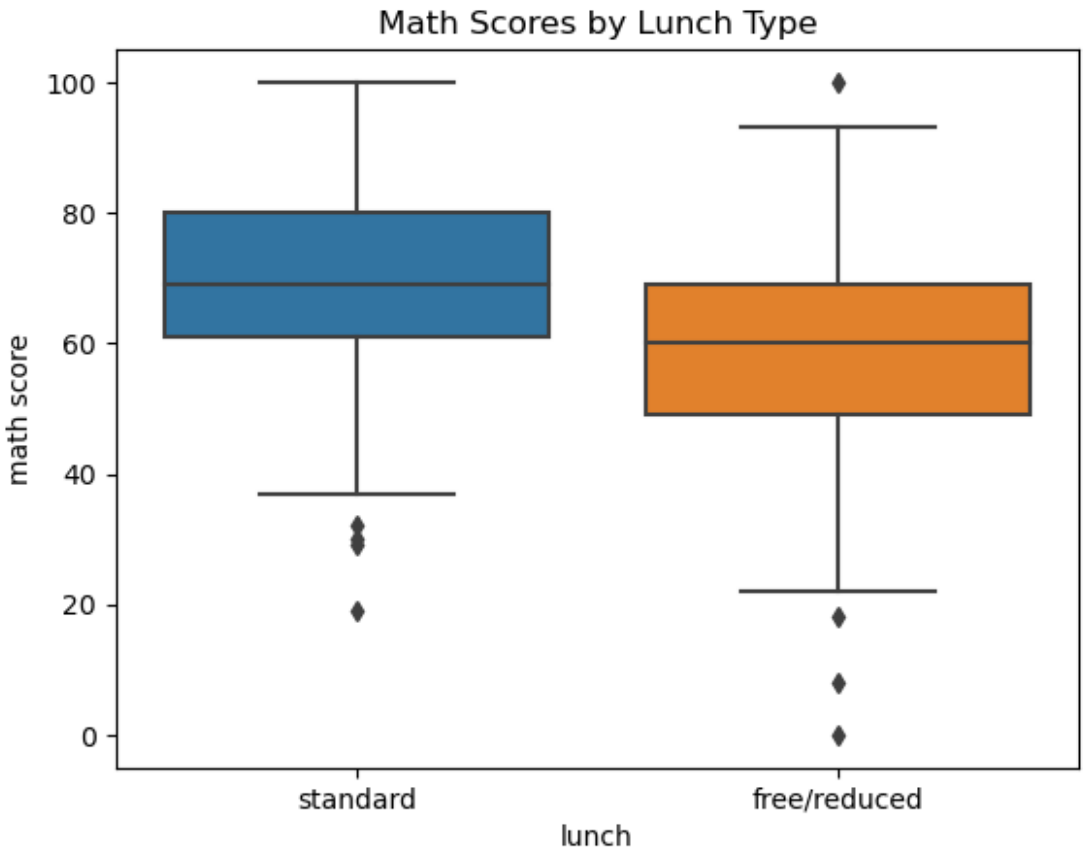

Math Scores by Parental Level of Education
math score
100
80
60
40
20
0
bachelor's degree
some college
master's degree
associate's degree
high school
some high school
parental level of education

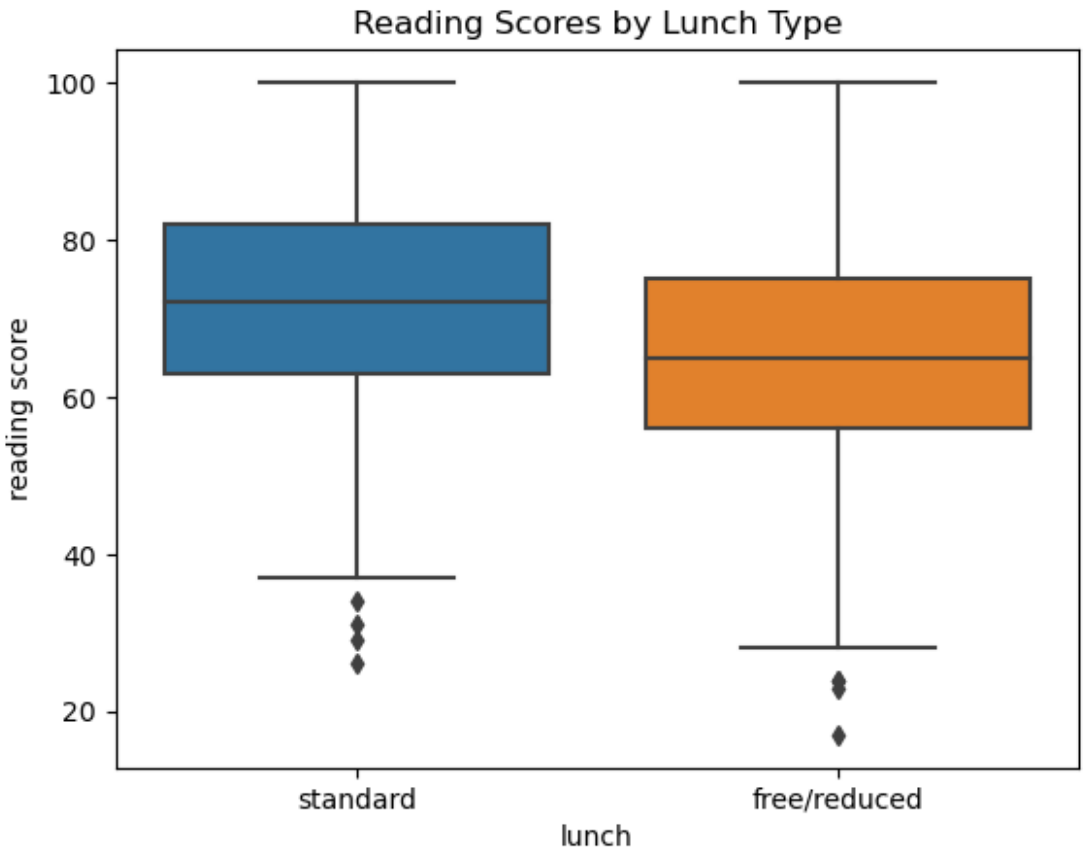

Reading Scores by Parental Level of Education
reading score
100
80
60
40
20
bachelor's degree
some college
master's degree
associate's degree
high school
some high school
parental level of education

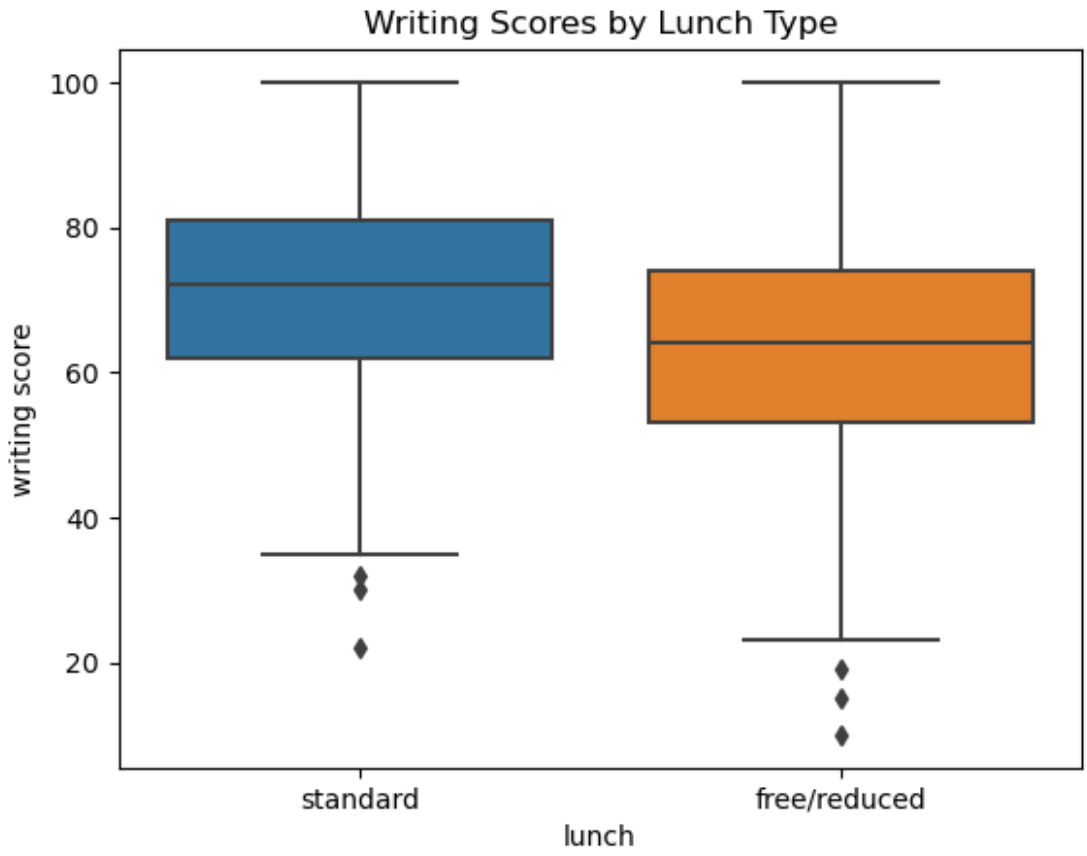


Visualize the impact of lunch on scores

```
sns.boxplot(x="lunch", y="math score", data=performance)
plt.title("Math Scores by Lunch Type")
plt.show()

sns.boxplot(x="lunch", y="reading score", data=performance)
plt.title("Reading Scores by Lunch Type")
plt.show()

sns.boxplot(x="lunch", y="writing score", data=performance)
plt.title("Writing Scores by Lunch Type")
plt.show()
```

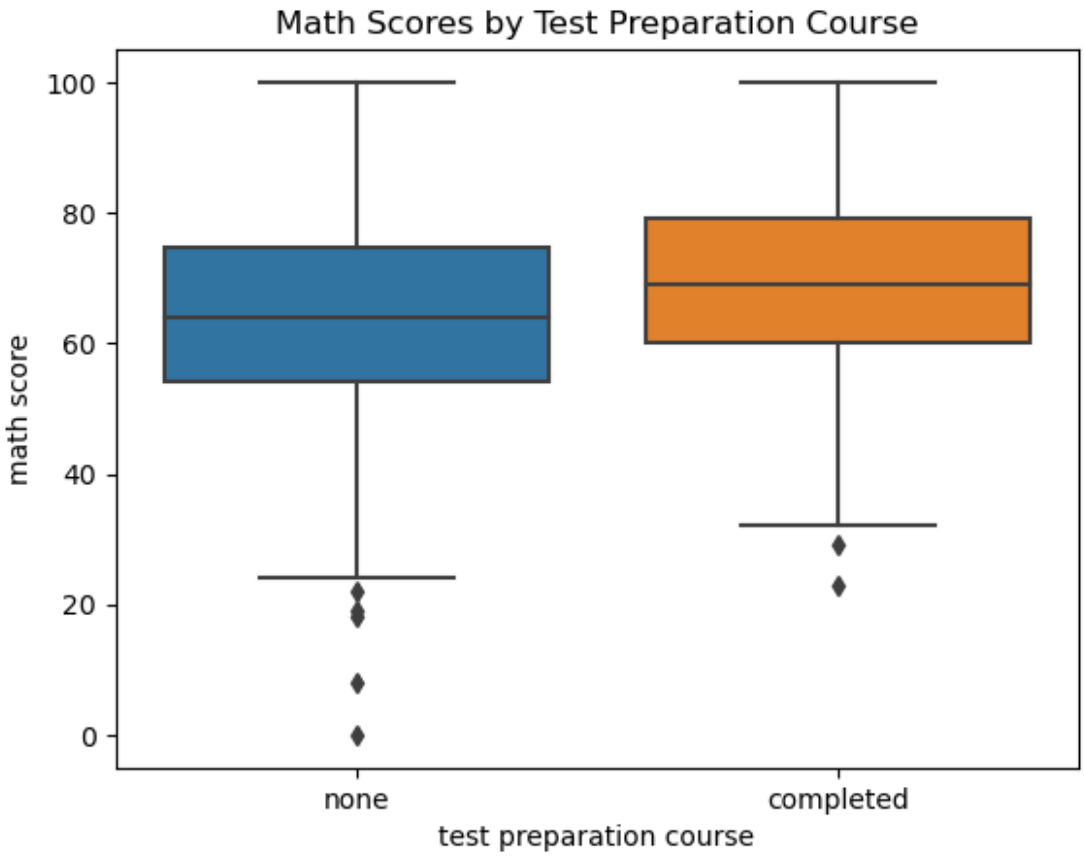

Math Scores by Lunch Type
100
80
60
40
20
0
math score
standard
free/reduced
lunch

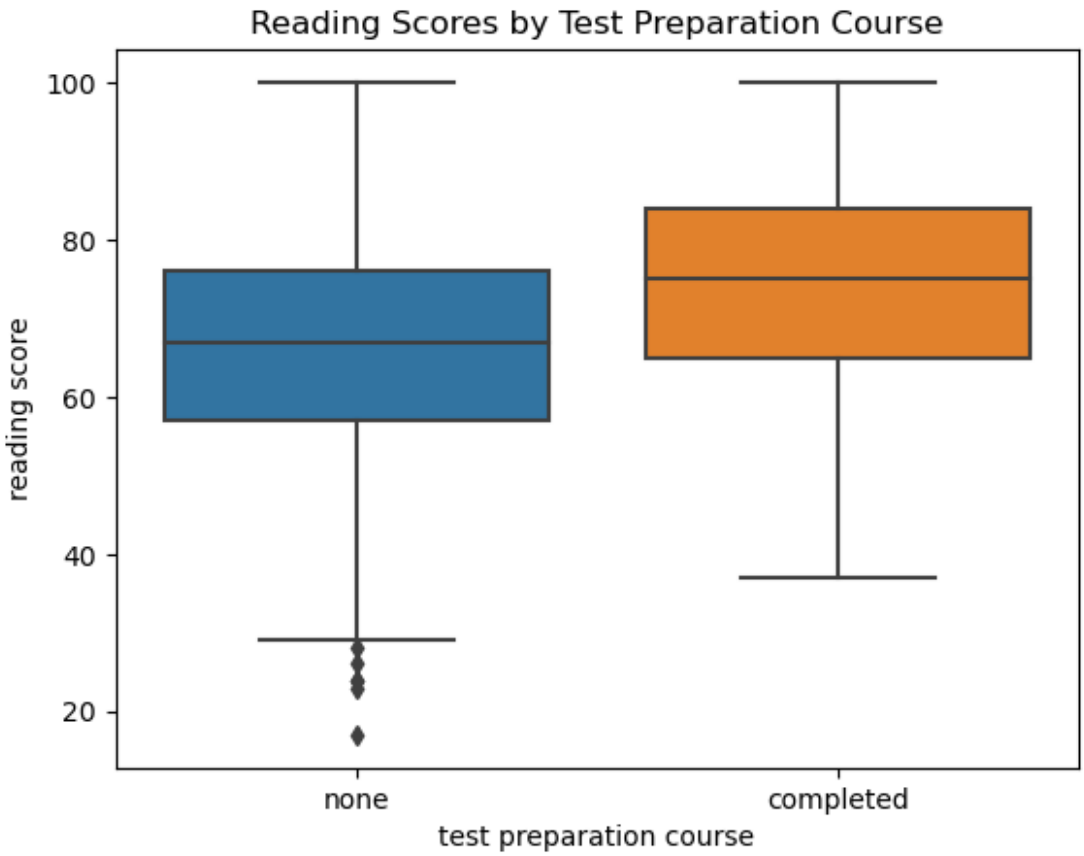

Reading Scores by Lunch Type
100
80
60
40
20
reading score
standard
free/reduced
lunch

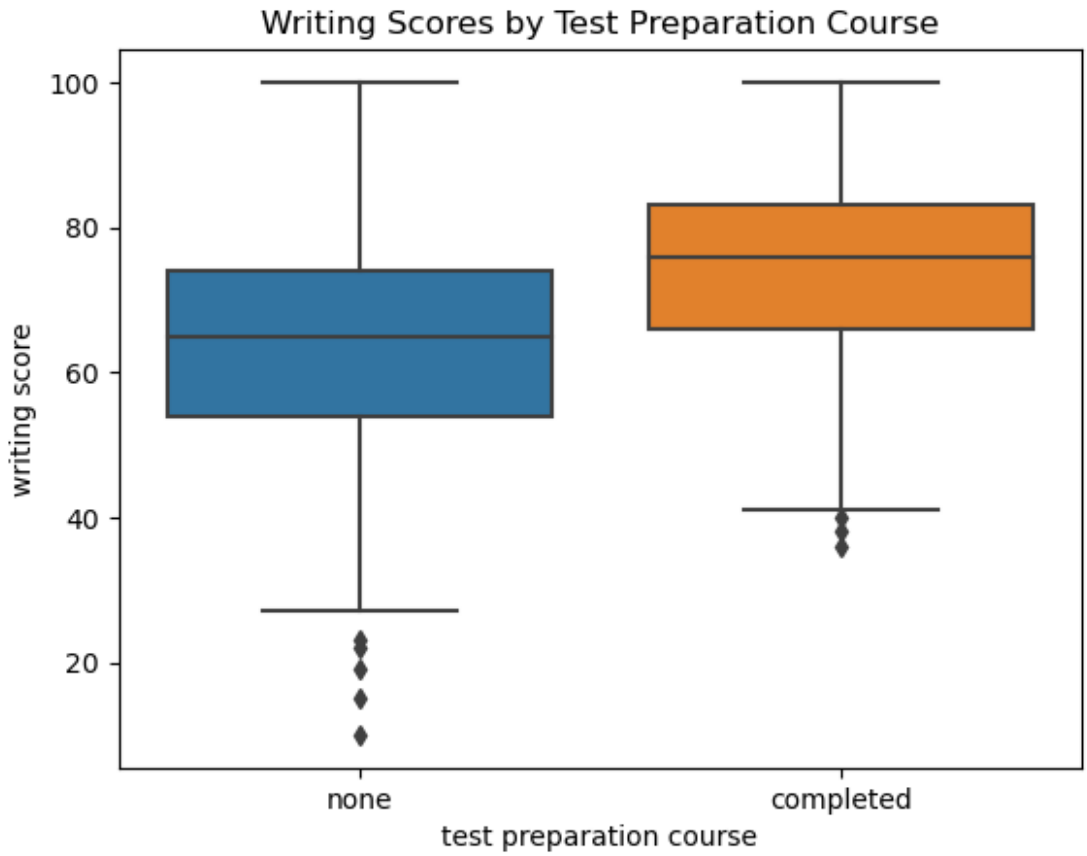


Visualize the impact of test preparation course

```
sns.boxplot(x="test preparation course", y="math score",
data=performance)
plt.title("Math Scores by Test Preparation Course")
plt.show()

sns.boxplot(x="test preparation course", y="reading score",
data=performance)
plt.title("Reading Scores by Test Preparation Course")
plt.show()

sns.boxplot(x="test preparation course", y="writing score",
data=performance)
plt.title("Writing Scores by Test Preparation Course")
plt.show()
```


Math Scores by Test Preparation Course
100
80
60
40
20
0
math score
none
completed
test preparation course


Reading Scores by Test Preparation Course
100
80
60
40
20
reading score
none
completed
test preparation course




```
Question 3 (20 marks)

A dataset containing information about the weather in a city is
available at weatherHistory.csv. Analyze the dataset and identify the
trends in temperature, precipitation, and humidity over time.
Visualize your findings using appropriate charts and graphs.

path = r"C:\Users\atif\Dropbox\Data Science NED\Data Visualization\
Final Paper\Hybrid Exam Paper\weatherHistory.csv"
weather = pd.read_csv(path)
weather.head()

                  Formatted Date        Summary Precip Type
Temperature (C)  \
0  2006-04-01 00:00:00.000 +0200  Partly Cloudy        rain
9.472222
1  2006-04-01 01:00:00.000 +0200  Partly Cloudy        rain
9.355556
2  2006-04-01 02:00:00.000 +0200  Mostly Cloudy        rain
9.377778
3  2006-04-01 03:00:00.000 +0200  Partly Cloudy        rain
8.288889
4  2006-04-01 04:00:00.000 +0200  Mostly Cloudy        rain
```

```
8.755556

   Apparent Temperature (C)  Humidity  Wind Speed (km/h)  \
0                  7.388889      0.89            14.1197
1                  7.227778      0.86            14.2646
2                  9.377778      0.89             3.9284
3                  5.944444      0.83            14.1036
4                  6.977778      0.83            11.0446

   Wind Bearing (degrees)  Visibility (km)  Loud Cover  Pressure
(millibars)  \
0                   251.0          15.8263         0.0
1015.13
1                   259.0          15.8263         0.0
1015.63
2                   204.0          14.9569         0.0
1015.94
3                   269.0          15.8263         0.0
1016.41
4                   259.0          15.8263         0.0
1016.51

                       Daily Summary
0  Partly cloudy throughout the day.
1  Partly cloudy throughout the day.
2  Partly cloudy throughout the day.
3  Partly cloudy throughout the day.
4  Partly cloudy throughout the day.
```

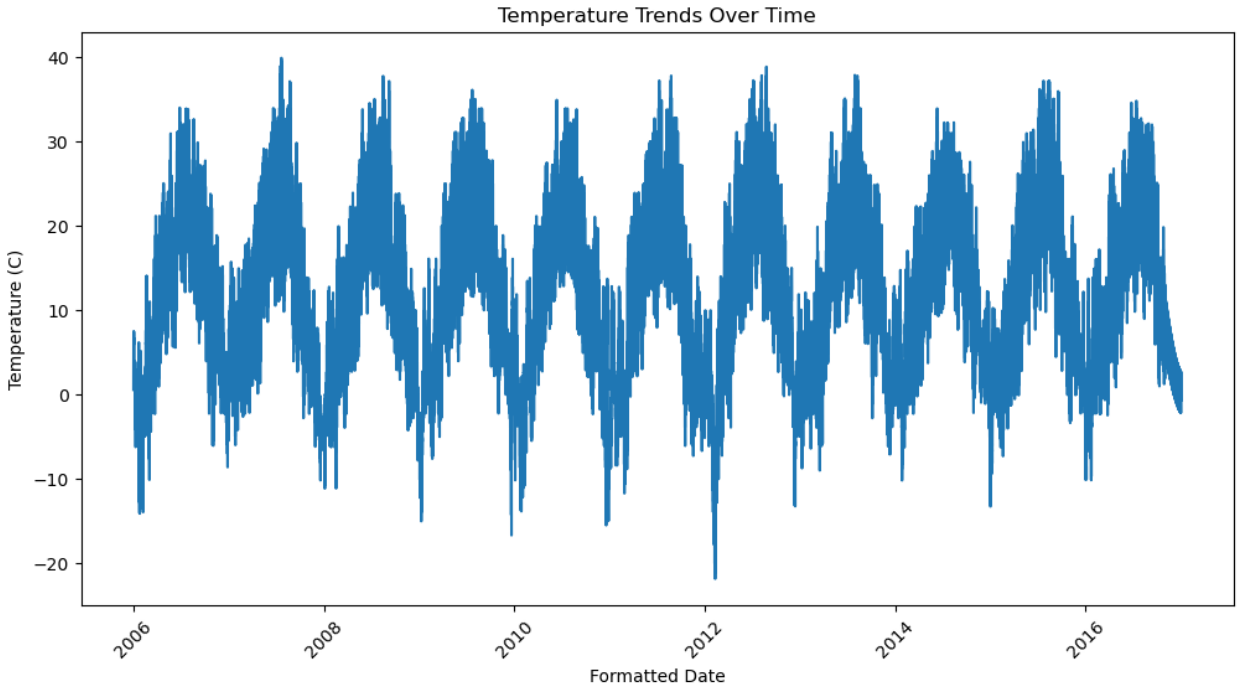
```
weather["Formatted Date"] = pd.to_datetime(weather["Formatted Date"],
utc=True)

weather.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 96453 entries, 0 to 96452
Data columns (total 12 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Formatted Date            96453 non-null  datetime64[ns, UTC]
 1   Summary                   96453 non-null  object
 2   Precip Type               95936 non-null  object
 3   Temperature (C)           96453 non-null  float64
 4   Apparent Temperature (C)  96453 non-null  float64
 5   Humidity                  96453 non-null  float64
 6   Wind Speed (km/h)         96453 non-null  float64
 7   Wind Bearing (degrees)    96453 non-null  float64
 8   Visibility (km)           96453 non-null  float64
 9   Loud Cover                96453 non-null  float64
 10  Pressure (millibars)      96453 non-null  float64
```

```
 11  Daily Summary             96453 non-null  object
dtypes: datetime64[ns, UTC](1), float64(8), object(3)
memory usage: 8.8+ MB
```

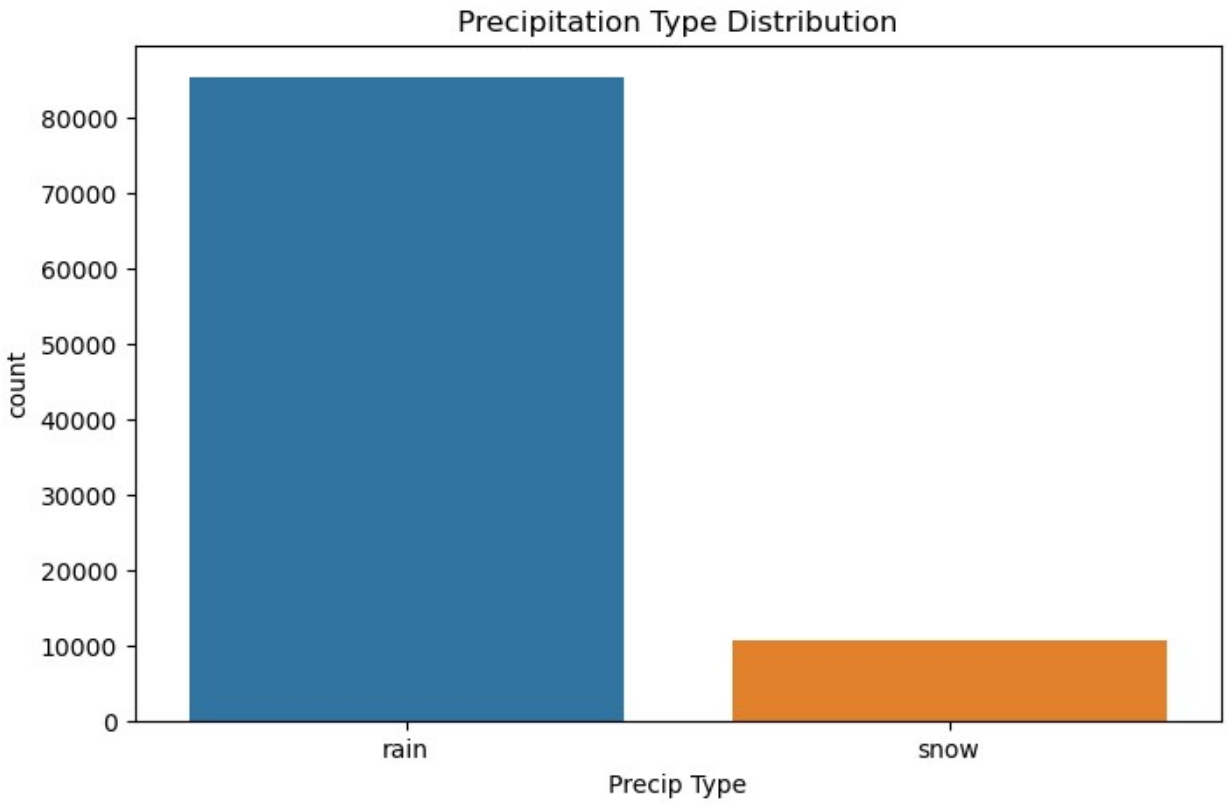
For Temperature Over Time

```
plt.figure(figsize=(12, 6))
sns.lineplot(x="Formatted Date", y="Temperature (C)", data=weather)
plt.title("Temperature Trends Over Time")
plt.xticks(rotation=45)
plt.show()
```



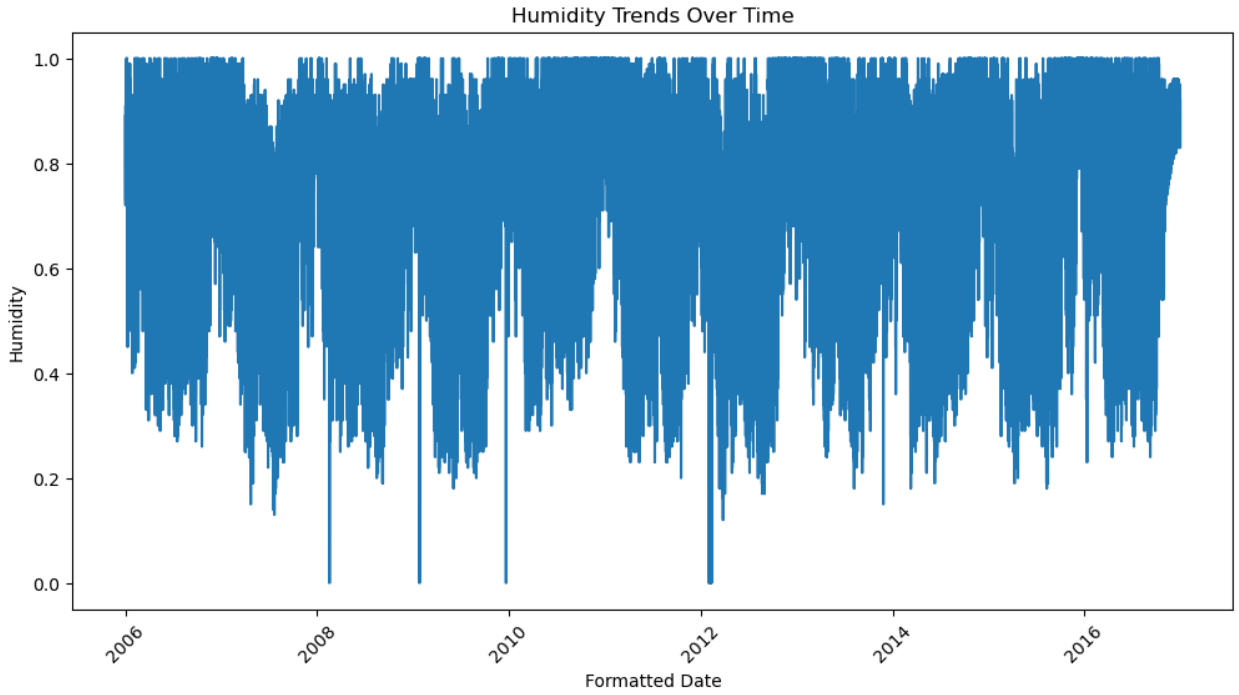
For Precipitation

```
plt.figure(figsize=(8, 5))
sns.countplot(x="Precip Type", data=weather)
plt.title("Precipitation Type Distribution")
plt.show()
```



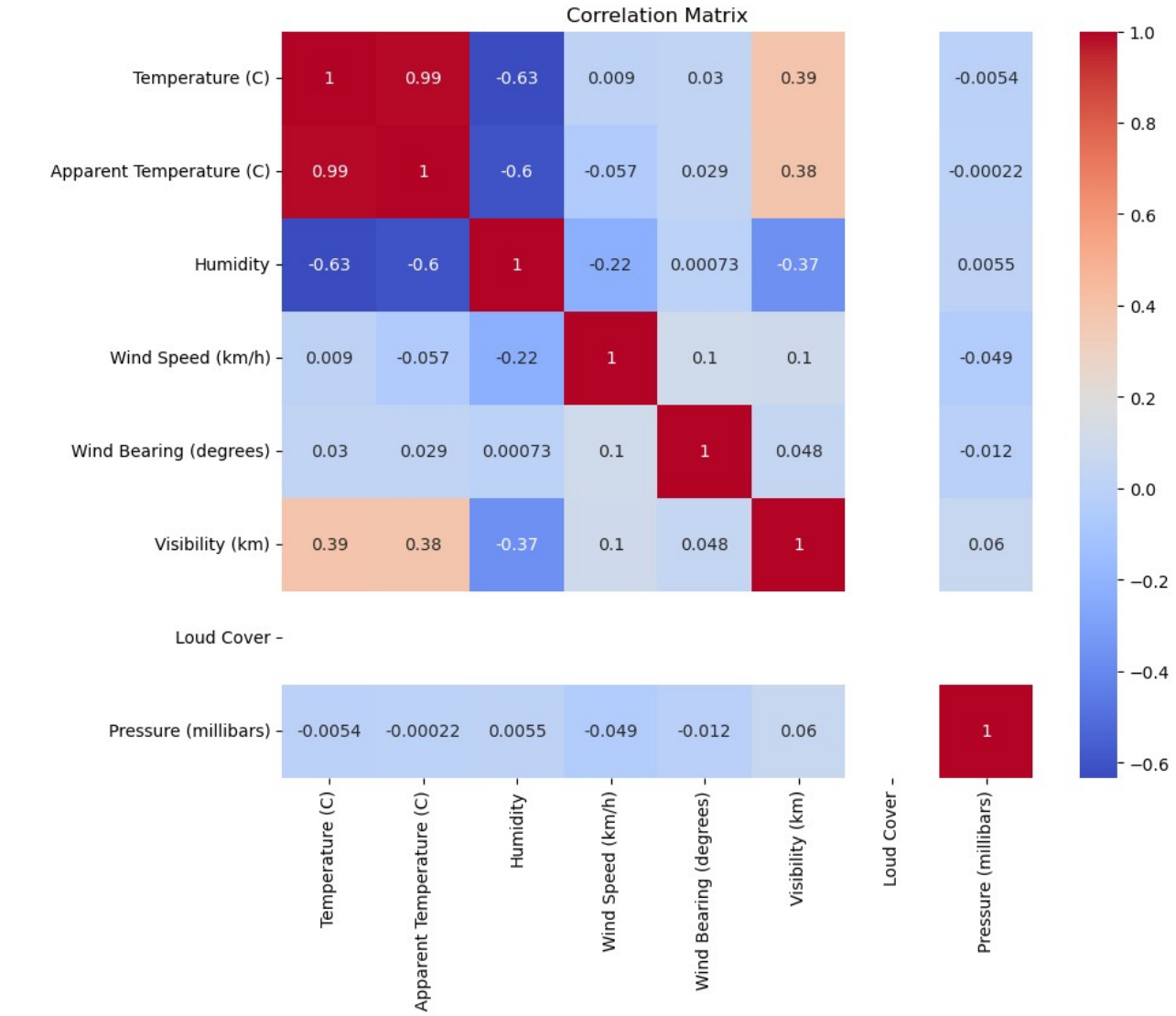
For Humidity Over Time

```
plt.figure(figsize=(12, 6))
sns.lineplot(x="Formatted Date", y="Humidity", data=weather)
plt.title("Humidity Trends Over Time")
plt.xticks(rotation=45)
plt.show()
```



Correlation Matrix

```
correlation_matrix = weather.corr()
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm")
plt.title("Correlation Matrix")
plt.show()
```


Correlation Matrix
Temperature (C)
Apparent Temperature (C)
Humidity
Wind Speed (km/h)
Wind Bearing (degrees)
Visibility (km)
Loud Cover
Pressure (millibars)
1
0.99
-0.63
0.009
0.03
0.39
-0.0054
0.99
1
-0.6
-0.057
0.029
0.38
-0.00022
-0.63
-0.6
1
-0.22
0.00073
-0.37
0.0055
0.009
-0.057
-0.22
1
0.1
0.1
-0.049
0.03
0.029
0.00073
0.1
1
0.048
-0.012
0.39
0.38
-0.37
0.1
0.048
1
0.06
-0.0054
-0.00022
0.0055
-0.049
-0.012
0.06
1
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6