Assignment 1

0801CS221114

```
import pandas as pd
df = pd.read_csv('st.csv')
df.head()
```

	Unnamed: 0	Gender	EthnicGroup	ParentEduc	LunchType	TestPrep	MathScore	ReadingScore	WritingScore
0	0	female	group B	bachelor's degree	standard	none	72	72	74
1	1	female	group C	some college	standard	completed	69	90	88
2	2	female	group B	master's degree	standard	none	90	95	93
3	3	male	group A	associate's degree	free/reduced	none	47	57	44
4	4	male	group C	some college	standard	none	76	78	75

```
df.describe(include='all')
```

	Gender	EthnicGroup	ParentEduc	LunchType	TestPrep	MathScore	ReadingScore	WritingScore
count	30641	30641	30641	30641	30641	30641.000000	30641.000000	30641.000000
unique	2	5	6	2	2	NaN	NaN	NaN
top	female	group C	some college	standard	none	NaN	NaN	NaN
freq	15424	9816	7048	19905	20068	NaN	NaN	NaN
mean	NaN	NaN	NaN	NaN	NaN	66.749355	69.624980	68.468327
std	NaN	NaN	NaN	NaN	NaN	15.206049	14.671572	15.307814
min	NaN	NaN	NaN	NaN	NaN	0.000000	10.000000	5.000000
25%	NaN	NaN	NaN	NaN	NaN	56.000000	60.000000	58.000000
50%	NaN	NaN	NaN	NaN	NaN	67.000000	70.000000	69.000000
75%	NaN	NaN	NaN	NaN	NaN	78.000000	80.000000	79.000000
max	NaN	NaN	NaN	NaN	NaN	100.000000	100.000000	100.000000

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30641 entries, 0 to 30640
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   Unnamed: 0    30641 non-null  int64 
 1   Gender        30641 non-null  object
 2   EthnicGroup   30641 non-null  object
 3   ParentEduc    30641 non-null  object
 4   LunchType     30641 non-null  object
 5   TestPrep      30641 non-null  object
 6   MathScore     30641 non-null  int64 
 7   ReadingScore  30641 non-null  int64 
 8   WritingScore  30641 non-null  int64 
dtypes: int64(4), object(5)
memory usage: 2.1+ MB
```

```
if 'Unnamed: 0' in df.columns:
    df = df.drop(columns=['Unnamed: 0'])

X = df.drop(columns=['MathScore', 'ReadingScore', 'WritingScore'])
y = df[['MathScore', 'ReadingScore', 'WritingScore']]

X.head()
y.head()
```

```
   MathScore  ReadingScore  WritingScore
0         72            72            74
1         69            90            88
2         90            95            93
3         47            57            44
4         76            78            75
```

```
df.isnull().sum()
```

```
              0
Gender        0
EthnicGroup   0
ParentEduc    0
LunchType     0
TestPrep      0
MathScore     0
ReadingScore  0
WritingScore  0

dtype: int64
```

```
unique_categories = {col: df[col].nunique() for col in ['MathScore', 'ReadingScore', 'WritingScore']}
unique_categories
```

```
{'MathScore': 94, 'ReadingScore': 88, 'WritingScore': 92}
```

```
from statistics import mean, median, mode, variance, stdev
```

```
mean_value = mean(df['MathScore'])
mean_value
```

```
66.74935543879116
```

```
median_value = median(df['MathScore'])
median_value
```

```
67
```

```
mode_value = mode(df['MathScore'])
mode_value
```

```
68
```

```
variance_value = variance(df['MathScore'])
variance_value
```

```
231.22392460084583
```

```
std_dev_value = stdev(df['MathScore'])
std_dev_value
```

```
15.206048947732802
```

```
import random
from math import sqrt


point1 = df.sample(1).iloc[0]
point2 = df.sample(1).iloc[0]

euclidean_distance = sqrt((point1['MathScore'] - point2['MathScore']) ** 2)

euclidean_distance
```

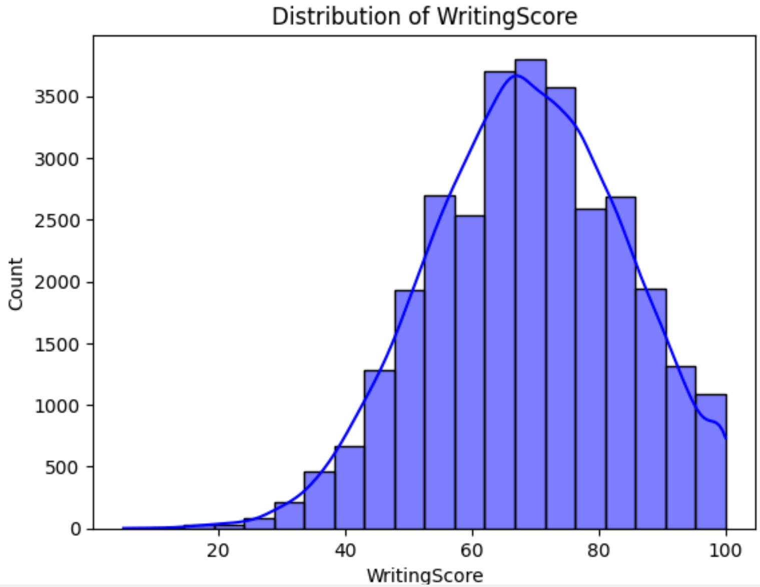
```
9.0
```

```
point1 = df.sample(1).iloc[0]
point2 = df.sample(1).iloc[0]

manhattan_distance = abs(point1['MathScore'] - point2['MathScore'])
manhattan_distance
```

```
19
```

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


sns.histplot(df['WritingScore'], kde=True, bins=20, color='blue')
plt.title('Distribution of WritingScore')
```

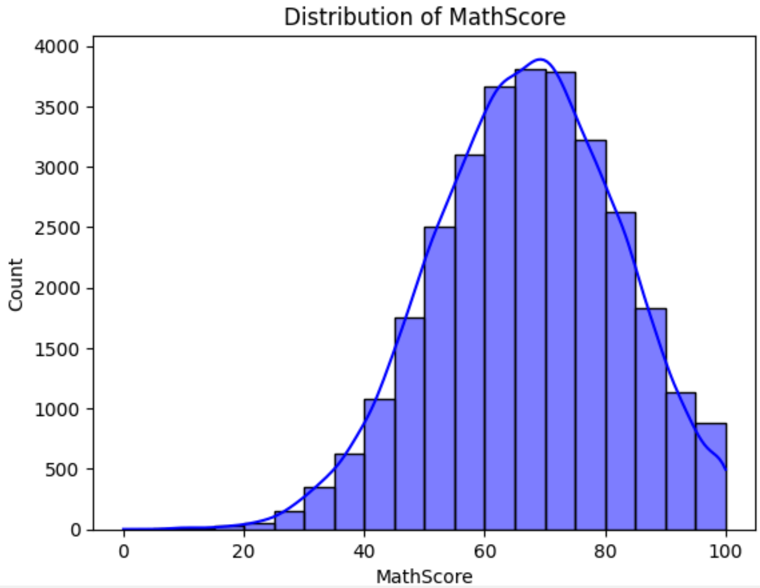
```
Text(0.5, 1.0, 'Distribution of WritingScore')
```



symmetric

```
sns.histplot(df['MathScore'], kde=True, bins=20, color='blue')
plt.title('Distribution of MathScore')
```

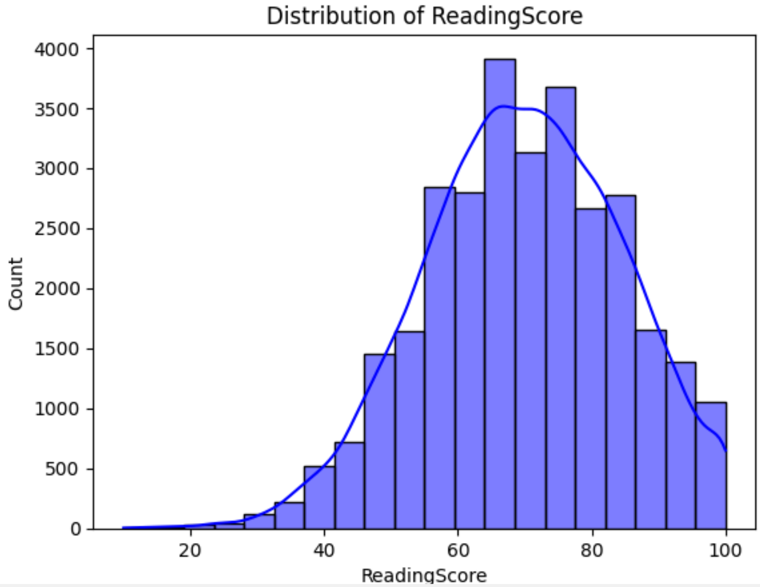
```
Text(0.5, 1.0, 'Distribution of MathScore')
```



scores spread out evenly between 40 and 90.

```
sns.histplot(df['ReadingScore'], kde=True, bins=20, color='blue')
plt.title('Distribution of ReadingScore')
```

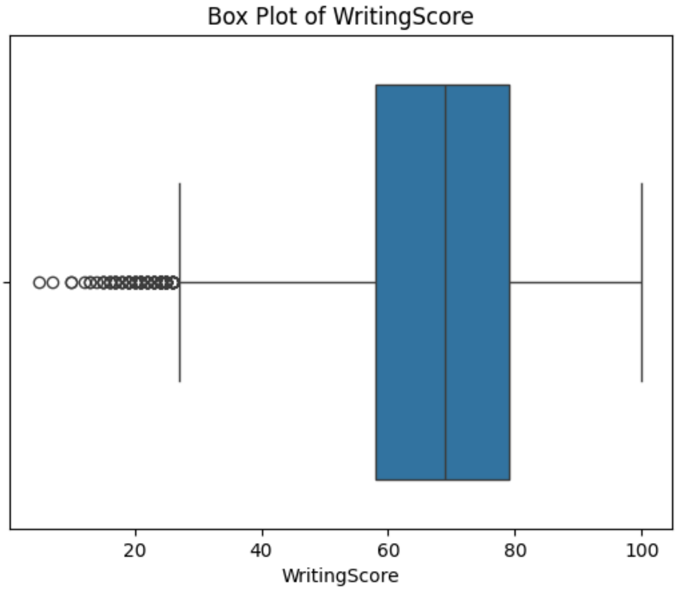
```
Text(0.5, 1.0, 'Distribution of ReadingScore')
```



The distribution is slightly right-skewed, with most values concentrated between 60 and 90.

```
sns.boxplot(x=df['WritingScore'])
plt.title('Box Plot of WritingScore')
```

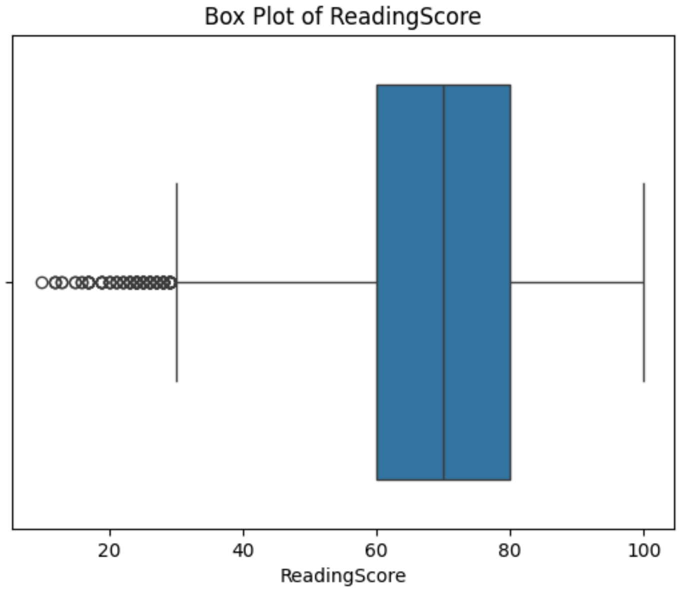
```
Text(0.5, 1.0, 'Box Plot of WritingScore')
```



The median is around 74

```
sns.boxplot(x=df['ReadingScore'])
plt.title('Box Plot of ReadingScore')
```

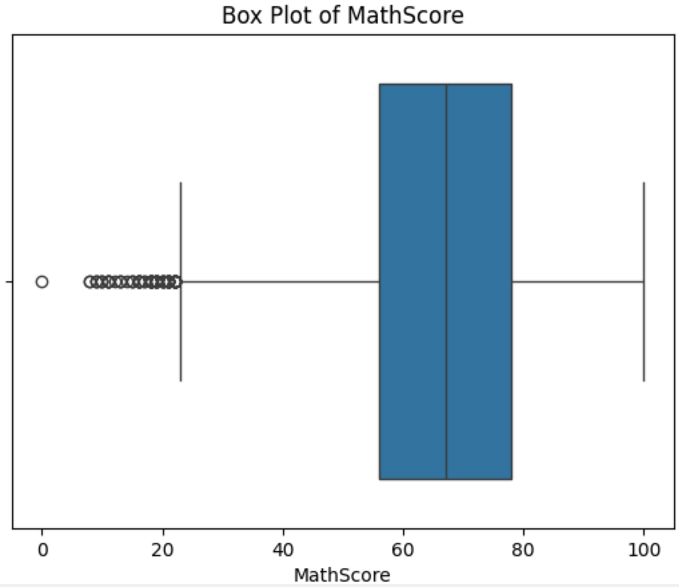
```
Text(0.5, 1.0, 'Box Plot of ReadingScore')
```



few outliers on the lower end, with the median around 75.

```
sns.boxplot(x=df['MathScore'])
plt.title('Box Plot of MathScore')
```

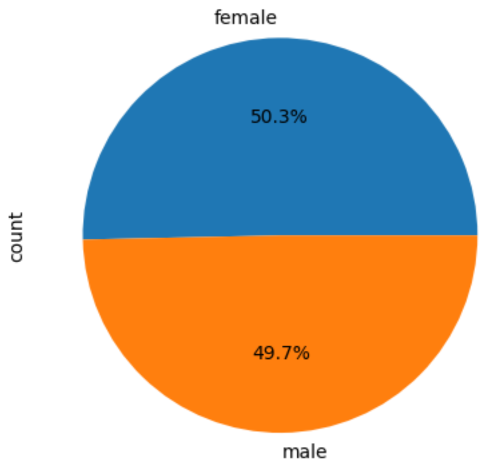
```
Text(0.5, 1.0, 'Box Plot of MathScore')
```



The median is around 65, with outliers on lower ends.

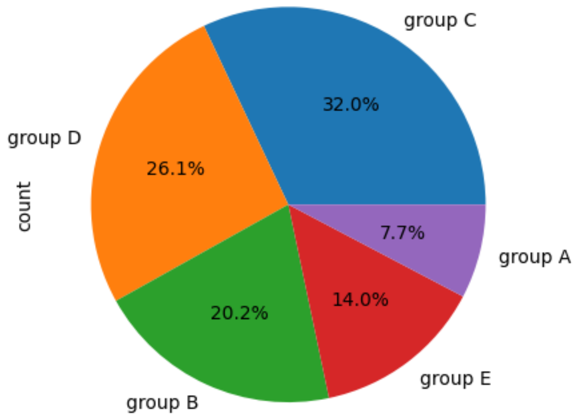
```
df['Gender'].value_counts().plot.pie(autopct='%1.1f%%')
```

```
<Axes: ylabel='count'>
```



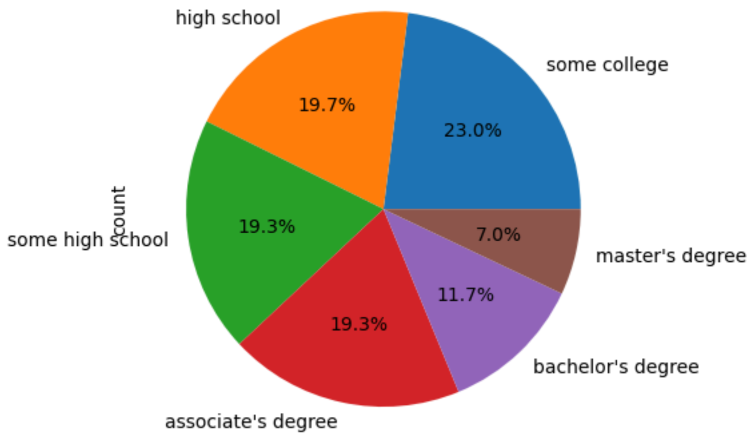
```
df['EthnicGroup'].value_counts().plot.pie(autopct='%1.1f%%')
```

```
<Axes: ylabel='count'>
```



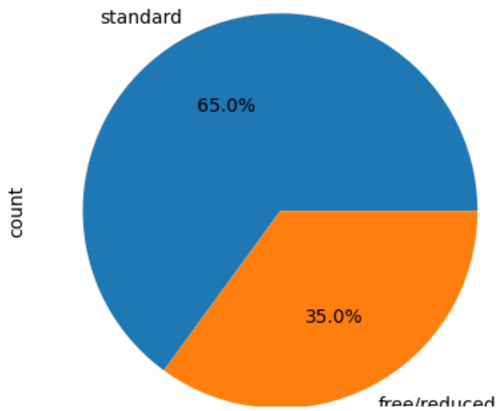
```
df['ParentEduc'].value_counts().plot.pie(autopct='%1.1f%%')
```

```
<Axes: ylabel='count'>
```



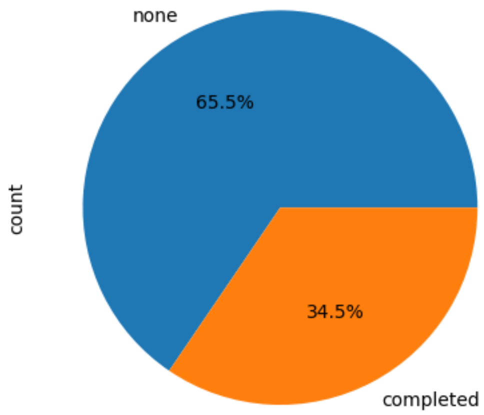
```
df['LunchType'].value_counts().plot.pie(autopct='%1.1f%%')
```

```
<Axes: ylabel='count'>
```



```
df['TestPrep'].value_counts().plot.pie(autopct='%1.1f%%')
```

```
<Axes: ylabel='count'>
```



```
plt.scatter(df['MathScore'], df['ReadingScore'])
```

```
<matplotlib.collections.PathCollection at 0x7cb9c059fe90>
```