

```
In [1]: #pip install numpy
#pip install pandas
#pip install matplotlib
#pip install seaborn
```

```
In [2]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [3]: df = pd.read_csv("Student_scores.csv")
print(df.head())
```

	Unnamed: 0	Gender	EthnicGroup	ParentEduc	LunchType	TestPrep	
0	0	female	NaN	bachelor's degree	standard	none	
1	1	female	group C	some college	standard	NaN	
2	2	female	group B	master's degree	standard	none	
3	3	male	group A	associate's degree	free/reduced	none	
4	4	male	group C	some college	standard	none	

	ParentMaritalStatus	PracticeSport	IsFirstChild	NrSiblings	TransportMeans	
0	married	regularly	yes	3.0	school_bus	
1	married	sometimes	yes	0.0	NaN	
2	single	sometimes	yes	4.0	school_bus	
3	married	never	no	1.0	NaN	
4	married	sometimes	yes	0.0	school_bus	

	WklyStudyHours	MathScore	ReadingScore	WritingScore
0	< 5	71	71	74
1	5 - 10	69	90	88
2	< 5	87	93	91
3	5 - 10	45	56	42
4	5 - 10	76	78	75

```
In [4]: df.describe()
```

	Unnamed: 0	NrSiblings	MathScore	ReadingScore	WritingScore
count	30641.000000	29069.000000	30641.000000	30641.000000	30641.000000
mean	499.556607	2.145894	66.558402	69.377533	68.418622
std	288.747894	1.458242	15.361616	14.758952	15.443525
min	0.000000	0.000000	0.000000	10.000000	4.000000
25%	249.000000	1.000000	56.000000	59.000000	59.000000
50%	500.000000	2.000000	67.000000	70.000000	69.000000
75%	750.000000	3.000000	78.000000	80.000000	79.000000
max	999.000000	7.000000	100.000000	100.000000	100.000000

```
In [5]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30641 entries, 0 to 30640
Data columns (total 15 columns):
# Column          Non-Null Count  Dtype
---  --
0  Unnamed: 0      30641 non-null  int64
1  Gender          30641 non-null  object
2  EthnicGroup     28801 non-null  object
3  ParentEduc     28796 non-null  object
4  LunchType      30641 non-null  object
5  TestPrep       28811 non-null  object
6  ParentMaritalStatus 29451 non-null  object
7  PracticeSport   30619 non-null  object
8  IsFirstChild   29737 non-null  object
9  NrSiblings      29069 non-null  float64
10 TransportMeans 27597 non-null  object
11 WklyStudyHours 29686 non-null  object
12 MathScore      30641 non-null  int64
13 ReadingScore   30641 non-null  int64
14 WritingScore   30641 non-null  int64
dtypes: float64(1), int64(4), object(10)
memory usage: 3.5+ MB
```

```
In [6]: df.isnull().sum()

Unnamed: 0      0
Gender           0
EthnicGroup     1840
ParentEduc      1845
LunchType       0
TestPrep        1830
ParentMaritalStatus 1190
PracticeSport    631
IsFirstChild    904
NrSiblings       1572
TransportMeans   3134
WklyStudyHours   955
MathScore        0
ReadingScore     0
WritingScore     0
dtype: int64
```

Drop unnamed column

```
In [7]: df=df.drop("Unnamed: 0", axis = 1)
print(df.head())
```

	Gender	EthnicGroup	ParentEduc	LunchType	TestPrep	
0	female	NaN	bachelor's degree	standard	none	
1	female	group C	some college	standard	NaN	
2	female	group B	master's degree	standard	none	
3	male	group A	associate's degree	free/reduced	none	
4	male	group C	some college	standard	none	

	ParentMaritalStatus	PracticeSport	IsFirstChild	NrSiblings	TransportMeans	
0	married	regularly	yes	3.0	school_bus	
1	married	sometimes	yes	0.0	NaN	
2	single	sometimes	yes	4.0	school_bus	
3	married	never	no	1.0	NaN	
4	married	sometimes	yes	0.0	school_bus	

	WklyStudyHours	MathScore	ReadingScore	WritingScore
0	< 5	71	71	74
1	5 - 10	69	90	88
2	< 5	87	93	91
3	5 - 10	45	56	42
4	5 - 10	76	78	75

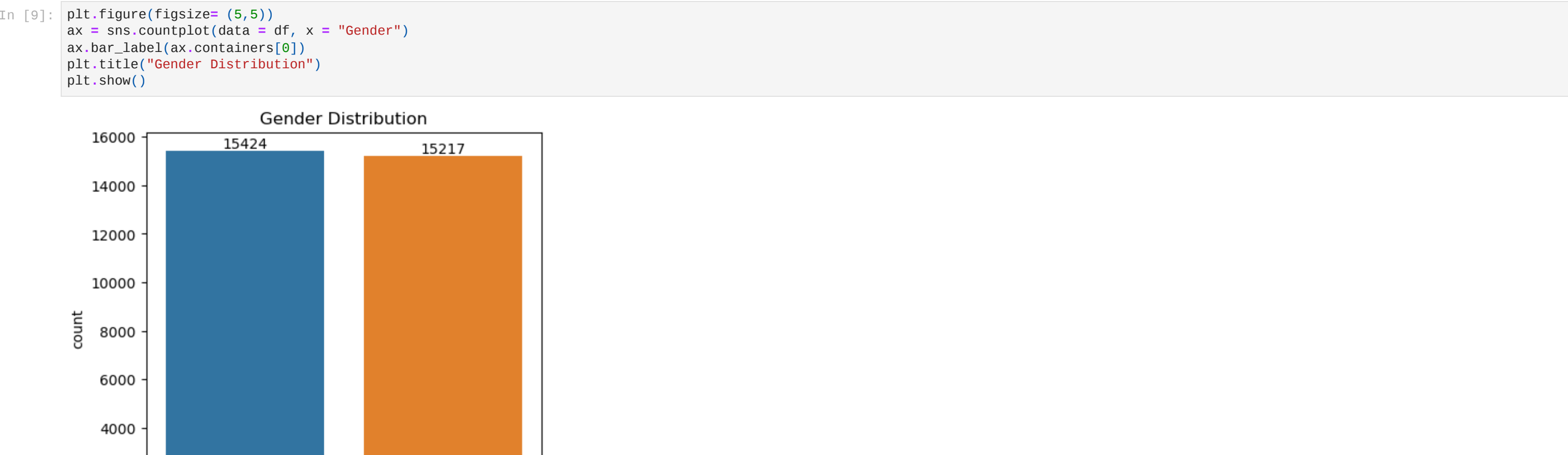
Change Weekly Study Hours Column

```
In [8]: df["WklyStudyHours"] =df["WklyStudyHours"].str.replace("05-Oct","5-10")
df.head()
```

	Gender	EthnicGroup	ParentEduc	LunchType	TestPrep	ParentMaritalStatus	PracticeSport	IsFirstChild	NrSiblings	TransportMeans	WklyStudyHours	MathScore	ReadingScore	WritingScore
0	female	NaN	bachelor's degree	standard	none	married	regularly	yes	3.0	school_bus	< 5	71	71	74
1	female	group C	some college	standard	NaN	married	sometimes	yes	0.0	NaN	5 - 10	69	90	88
2	female	group B	master's degree	standard	none	single	sometimes	yes	4.0	school_bus	< 5	87	93	91
3	male	group A	associate's degree	free/reduced	none	married	never	no	1.0	NaN	5 - 10	45	56	42
4	male	group C	some college	standard	none	married	sometimes	yes	0.0	school_bus	5 - 10	76	78	75

Gender Distribution

```
In [9]: plt.figure(figsize=(5,5))
ax = sns.countplot(data = df, x = "Gender")
ax.bar_label(ax.containers[0])
plt.title("Gender Distribution")
plt.show()
```



```
In [10]: #from the above chart we have analysed that:
#the number of females in the data is more than the number of males
```

```
In [11]: gb = df.groupby("ParentEduc").agg({"MathScore": "mean", "ReadingScore": "mean", "WritingScore": "mean"})
print(gb)
```

ParentEduc	MathScore	ReadingScore	WritingScore
associate's degree	68.365586	71.124324	70.299099
bachelor's degree	70.486627	73.062020	73.331069
high school	64.435731	67.233987	65.421136
master's degree	72.336134	75.832921	76.356896
some college	66.399472	69.179708	68.501432
some high school	62.504813	65.510785	63.632409

```
In [12]: plt.figure(figsize=(4,4))
sns.heatmap(gb, annot = True)
plt.title("Relationship between Parent's Education and Student's score")
plt.show()
```

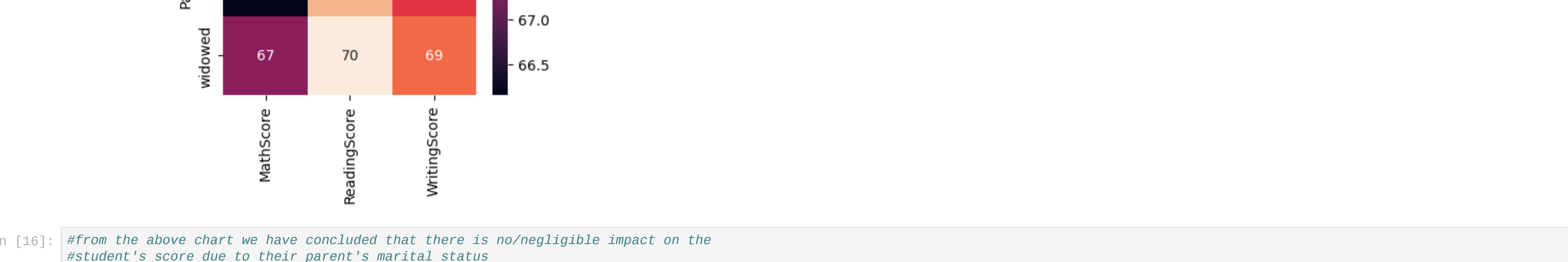


```
In [13]: #from the above chart we have concluded that the education of the process have a good impact on their student's score due to their parent Educ
```

```
In [14]: gb1 = df.groupby("ParentMaritalStatus").agg({"MathScore": "mean", "ReadingScore": "mean", "WritingScore": "mean"})
print(gb1)
```

ParentMaritalStatus	MathScore	ReadingScore	WritingScore
divorced	66.691197	69.655811	68.799146
married	66.657326	69.389575	68.420981
single	66.165704	69.157250	68.174440
widowed	67.368866	69.651438	68.563452

```
In [15]: plt.figure(figsize=(4,4))
sns.heatmap(gb1, annot = True)
plt.title("Relationship between Parent's Marital Status and Student's Score")
plt.show()
```

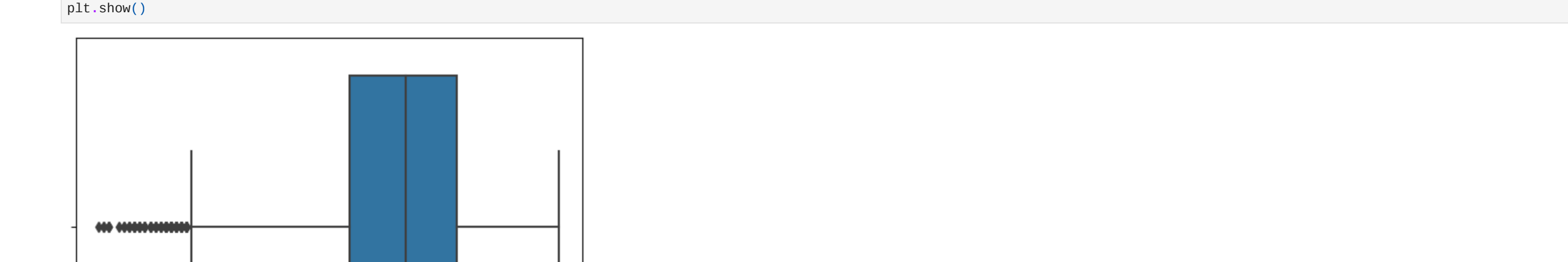


```
In [16]: #from the above chart we have concluded that there is no/negligible impact on the
#student's score due to their parent's marital status
```

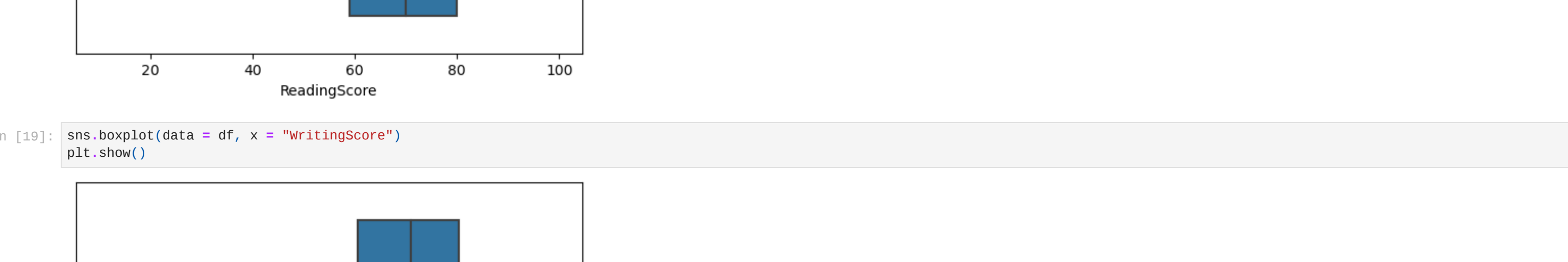
```
In [17]: sns.boxplot(data = df, x = "MathScore")
plt.show()
```



```
In [18]: sns.boxplot(data = df, x = "ReadingScore")
plt.show()
```



```
In [19]: sns.boxplot(data = df, x = "WritingScore")
plt.show()
```



```
In [20]: print(df["EthnicGroup"].unique())
[ nan 'group C' 'group B' 'group A' 'group D' 'group E' ]
```

Distribution of Ethnic Groups

```
In [24]: groupA = df.loc[(df['EthnicGroup'] == "group A")].count()
groupB = df.loc[(df['EthnicGroup'] == "group B")].count()
groupC = df.loc[(df['EthnicGroup'] == "group C")].count()
groupD = df.loc[(df['EthnicGroup'] == "group D")].count()
groupE = df.loc[(df['EthnicGroup'] == "group E")].count()

l = ["group A", "group B", "group C", "group D", "group E"]
mlist = [groupA["EthnicGroup"], groupB["EthnicGroup"], groupC["EthnicGroup"], groupD["EthnicGroup"], groupE["EthnicGroup"]]
mlist = [groupA["EthnicGroup"], groupB["EthnicGroup"], groupC["EthnicGroup"], groupD["EthnicGroup"], groupE["EthnicGroup"]]
plt.pie(mlist, labels = 1, autopct = "%1.2f%%")
plt.title("Distribution of Ethnic Groups")
plt.show()
```

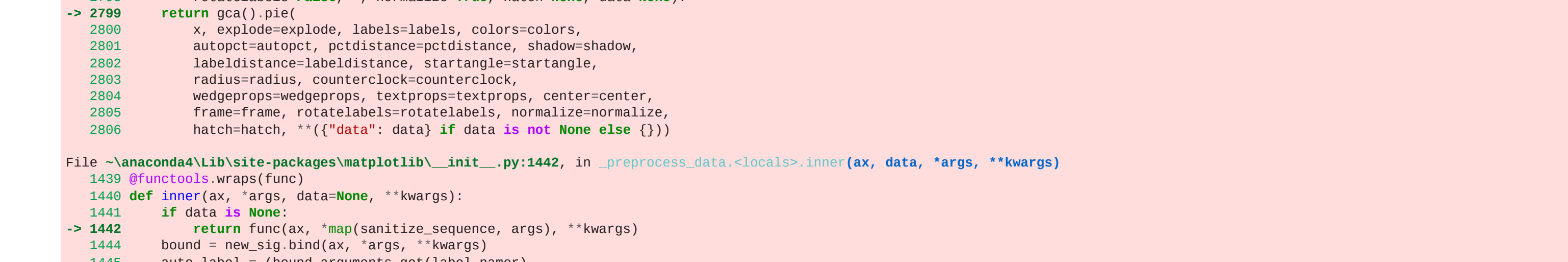
```
-----
TypeError                                Traceback (most recent call last)
Cell In[24], line 10
      7 l = ["group A", "group B", "group C", "group D", "group E"]
      8 mlist = [groupA["EthnicGroup"], groupB["EthnicGroup"], groupC["EthnicGroup"], groupD["EthnicGroup"], groupE["EthnicGroup"]]
-->  9 plt.pie(mlist, labels = 1, autopct = "%1.2f%%")
    10 plt.pie(mlist, labels = 1, autopct = "%1.2f%%")
    11 plt.title("Distribution of Ethnic Groups")
    12 plt.show()
```

```
File ~\anaconda4\lib\site-packages\matplotlib\pyplot.py:2799, in fig(x, explode, labels, colors, autopct, pctdistance, shadow, labeldistance, startangle, radius, counterclock, wedgeprops, t
2792 @copy_docstring_and_deprecators(Axes.pie)
2793 def pie(
2794     x, explode=None, labels=None, colors=None, autopct=None,
2795     (*, ...),
2796     textprops=None, center=(0, 0), frame=False,
2797     rotatelabels=False, *, normalize=True, hatch=None, data=None):
-> 2799     return gca().pie(
2800         x, explode=explode, labels=labels, colors=colors,
2801         autopct=autopct, pctdistance=pctdistance, shadow=shadow,
2802         labeldistance=labeldistance, startangle=startangle,
2803         radius=radius, counterclock=counterclock,
2804         wedgeprops=wedgeprops, textprops=textprops, center=center,
2805         frame=frame, rotatelabels=rotatelabels, normalize=normalize,
2806         hatch=hatch, **(({"data": data} if data is not None else {}))
2807     )
```

```
File ~\anaconda4\lib\site-packages\matplotlib\axes\_axes.py:1442, in _preprocess_data._locals().inner(ax, data, *args, **kwargs)
1439 @functools.wraps(func)
1440 def inner(ax, *args, data=None, **kwargs):
1441     if data is None:
-> 1442         return func(ax, *map(sanitize_sequence, args), **kwargs)
    1443     bound = new_sig.bind(ax, *args, **kwargs)
    1444     auto_label = (bound.arguments.get('label_name')
    1445                  or bound.kwargs.get('label_name'))
```

```
File ~\anaconda4\lib\site-packages\matplotlib\axes\_axes.py:3208, in Axes.pie(self, x, explode, labels, colors, autopct, pctdistance, shadow, labeldistance, radius, counterclock, wedgeprops, textprops, center, frame, rotatelabels, normalize, hatch)
3206 if explode is None:
3207     explode = [] + len(x)
-> 3208 if len(x) != len(labels):
    3209     raise ValueError("'label' must be of length 'x'")
    3210 if len(x) != len(explode):
```

```
TypeError: object of type 'int' has no len()
```



Thank You

```
In [25]: ax = sns.countplot(data = df, x = "EthnicGroup")
ax.bar_label(ax.container[0])
```

```
-----
AttributeError                            Traceback (most recent call last)
Cell In[25], line 2
      1 ax = sns.countplot(data = df, x = "EthnicGroup")
-->  2 ax.bar_label(ax.container[0])

AttributeError: 'Axes' object has no attribute 'container'
```

