Machine Learning Lab 2 - Statistical Inferences and Exploratory Data Analysis

Submitted By
Name: Harsha KG
Register Number: 19112005
Class: **5 BSc Data Science**

Lab Overview

Objectives

- Finding some Insightful Inference using statistcal technique.
- Understanding the explorartory data analysis using matplotlib and seaborn libraray.

Problem Definition

As a data analyst find some insightful inference for exam office of a foreign University so that the lecturers can concentrate on motivating students based on their study.

Approach

Imported the Dataset using required libraries using python and did some preprocessing techniques before exploration to make the datset into standard format.

Sections

1. Conclusion

References

1. https://www.geeksforgeeks.org/introduction-to-seaborn-python/ (https://www.geeksforgeeks.org/introduction-to-seaborn-python/)
2. https://seaborn.pydata.org/examples/scatterplot_categorical.html (https://seaborn.pydata.org/examples/scatterplot_categorical.html)
3. https://www.kaggle.com/ahmadjaved097/student-performance-eda-and-visualization (https://www.kaggle.com/ahmadjaved097/student-performance-eda-and-visualization)
4. https://seaborn.pydata.org/tutorial/categorical.html (https://seaborn.pydata.org/tutorial/categorical.html)
5. https://seaborn.pydata.org/generated/seaborn.violinplot.html (https://seaborn.pydata.org/generated/seaborn.violinplot.html)

About The Datset

This Dataset contains exam scores and some personal information of 500 different students of a Foreign University. It helps find us to understandard the performance of each subject in a class by looking into their marks and background. Consists of eight columns out of 5 are Categorical (Sex.Country,ParentEducation,FinancialAid,TestPreparation), and the remaining 3 are Continous Variable:

1. Sex- It helps us to identify whether the student is a girl or a boy.
2. Country- Tells us from which country they had come
3. ParentEducation- Helps us to understand the level of education of parents of each students.
4. Financial Aid- Tells us Whether they are supported with some loan, scholarship, sponsor, etc.
5. TestPreparation- Whether students write the exam by a thorough revision or not
6. Science- Marks secured in Science subject
7. Language-Marks secured in a language subject.
8. Communication-Marks secured in a language subject

Importing Required Libraries

In [2]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

Loading the Dataset

In [3]:

```
df=pd.read_csv("ExamResults.csv")
```

In [4]:

```
#displaying the first 20 rows
df.head(20)
```

Out[4]:

	Sex	Country	ParentEducation	FinancialAid	TestPreparation	Science	Language	Communcation
0	Girl	UK	bachelor's degree	No Assistance	Minimum	75	74	75
1	Girl	USA	some college	No Assistance	Thorough	72	92	89
2	Girl	UK	master's degree	No Assistance	Minimum	93	97	94
3	Boy	India	associate's degree	Assisted	Minimum	50	59	45
4	Boy	USA	some college	No Assistance	Minimum	79	80	76
5	Girl	UK	associate's degree	No Assistance	Minimum	74	85	79
6	Girl	UK	some college	No Assistance	Thorough	91	97	93
7	Boy	UK	some college	Assisted	Minimum	43	45	40
8	Boy	Australia	high school	Assisted	Thorough	67	66	68
9	Girl	UK	high school	Assisted	Minimum	41	62	51
10	Boy	USA	associate's degree	No Assistance	Minimum	61	56	53
11	Boy	Australia	associate's degree	No Assistance	Minimum	43	54	44
12	Girl	UK	high school	No Assistance	Minimum	68	83	74
13	Boy	India	some college	No Assistance	Thorough	81	74	71
14	Girl	India	master's degree	No Assistance	Minimum	53	55	59
15	Girl	USA	some high school	No Assistance	Minimum	72	77	79
16	Boy	USA	high school	No Assistance	Minimum	91	91	87
17	Girl	UK	some high school	Assisted	Minimum	21	34	29
18	Boy	USA	master's degree	Assisted	Thorough	49	44	47
19	Girl	USA	associate's degree	Assisted	Minimum	57	60	62

Basic Inference from Data (Data Wrangling)

Finding the dimension of dataset

In [5]:

```
df.shape
```

Out[5]:

```
(499, 8)
```

There are total 499 rows and 8 columns

Getting the Concise summary of Dataframe

```
In [6]: df.info()#to get info about filled values in the dataframe for every column
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 499 entries, 0 to 498
Data columns (total 8 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   Sex              499 non-null    object
 1   Country          499 non-null    object
 2   ParentEducation  499 non-null    object
 3   FinancialAid     499 non-null    object
 4   TestPreparation  499 non-null    object
 5   Science          499 non-null    int64 
 6   Language         499 non-null    int64 
 7   Communcation     499 non-null    int64 
dtypes: int64(3), object(5)
memory usage: 31.3+ KB
```

From this we can understand that total of 5 columns are of object type and remaining three are in int64 type. The total memory usage of dataframe is 31.3 KB and 499 non-null values are present in each column.

Displaying the column names

```
In [7]: df.columns
```

```
Out[7]: Index(['Sex', 'Country', 'ParentEducation', 'FinancialAid', 'TestPreparation',
               'Science', 'Language', 'Communcation'],
              dtype='object')
```

There is a spelling mistake for column name 'Communcation' and it is better to use Gender instead of Sex.

Replacing column names

```
In [8]: df.rename(columns = {'Sex':'Gender',
                             'Communcation':'Communication'}, inplace = True)
```

Checking the no of null values

```
In [9]: df.isnull().sum()
```

```
Out[9]: Gender             0
        Country            0
        ParentEducation    0
        FinancialAid       0
        TestPreparation    0
        Science            0
        Language           0
        Communication      0
        dtype: int64
```

We can see that now there are no missing values in any of the variable.

Checking the Description of Datset

In [10]:
```
df.describe(include='all')
```

Out[10]:

	Gender	Country	ParentEducation	FinancialAid	TestPreparation	Science	Language	Communication
count	499	499	499	499	499	499.000000	499.000000	499.000000
unique	2	5	6	2	2	NaN	NaN	NaN
top	Boy	USA	associate's degree	No Assistance	Minimum	NaN	NaN	NaN
freq	253	166	120	327	328	NaN	NaN	NaN
mean	NaN	NaN	NaN	NaN	NaN	68.655311	70.434870	68.234469
std	NaN	NaN	NaN	NaN	NaN	14.966275	14.664291	15.187950
min	NaN	NaN	NaN	NaN	NaN	3.000000	19.000000	11.000000
25%	NaN	NaN	NaN	NaN	NaN	60.000000	60.000000	58.000000
50%	NaN	NaN	NaN	NaN	NaN	69.000000	71.000000	69.000000
75%	NaN	NaN	NaN	NaN	NaN	79.000000	81.000000	79.000000
max	NaN	NaN	NaN	NaN	NaN	100.000000	99.000000	99.000000

For categorical we can see that mean, median,upper and lower quartile,freq, etc are 0 thus,describe() is generally useful for numerical attributes.

The median (second quartile) of science is (69),lang(71) and for communication(69) which means these are the middle score in this list of each subject.

The First quartile of science and language is 60 and for communication it is 69 means these values are above and below of 25% of dataset. ie, 25% of 499

The Third quartile of science and communication is 79 and for language it is 81 means these values are above and below of 75% of dataset. ie, 75% of 499

Exploratory Data Analysis

Finding the count of each variable

Countplot for each imported Categorical variable using seaborn library

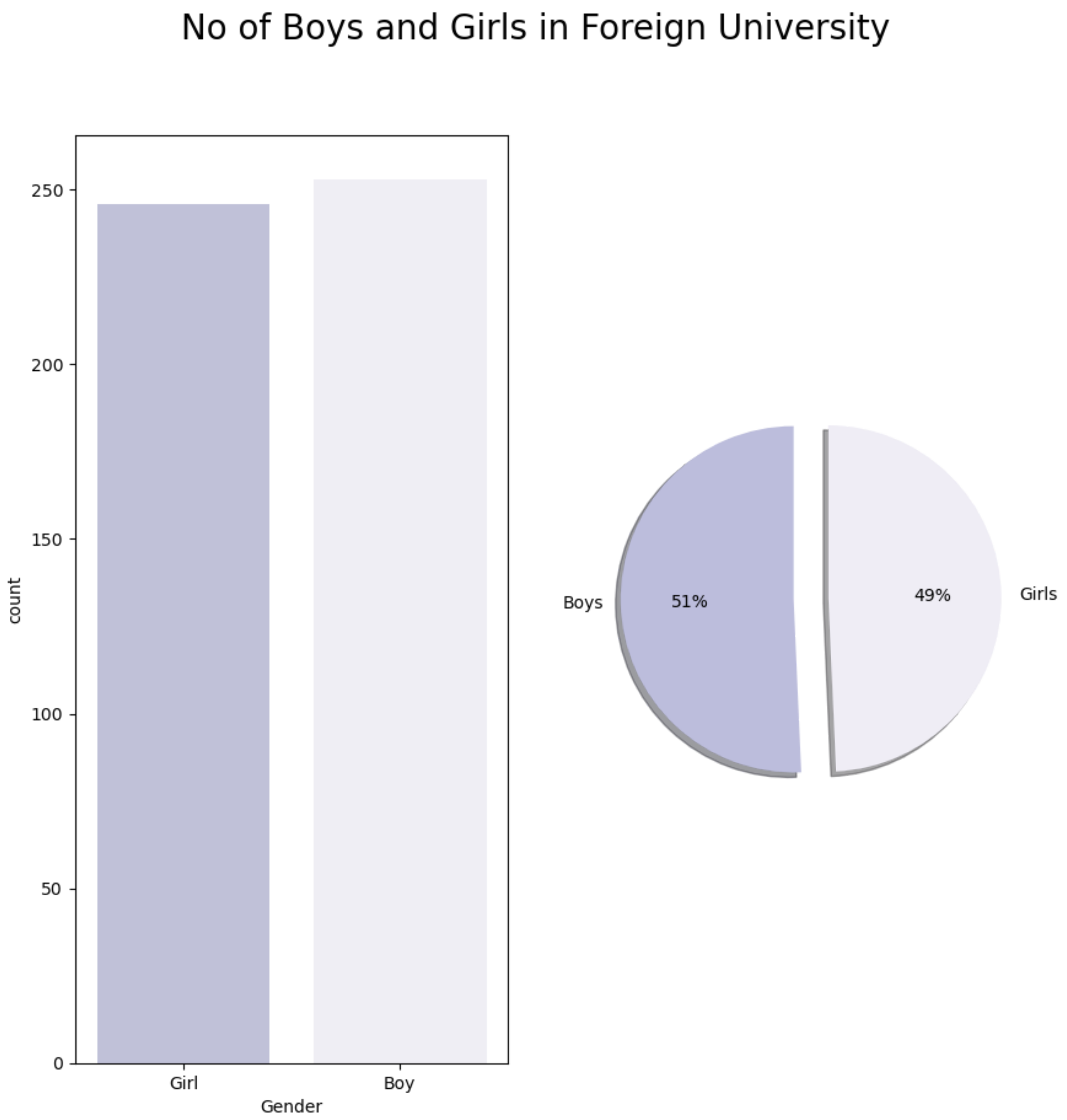
i) Gender

In [11]:
```
df['Gender'].value_counts()
```

Out[11]:
```
Boy     253
Girl    246
Name: Gender, dtype: int64
```

In [12]:
```
fig, axes = plt.subplots(ncols=2,
                         nrows=1,
                         figsize=(10, 10),
                         dpi=100)

sns.countplot(df['Gender'],palette=['#bcbddc','#efedf5'],ax=axes[0])

labels=['Boys','Girls']
colors = ('#bcbddc','#efedf5',)
axes[1].pie(df['Gender'].value_counts(),
            labels=labels,
            autopct='%1.0f%%',
            shadow=True,
            startangle=90,
            explode = (0.1, 0.1),
            colors=colors
            )

fig.suptitle('No of Boys and Girls in Foreign University', fontsize=20)
plt.show()
```

```
C:\Users\HP\anaconda3\anacondaorginal\lib\site-packages\seaborn\_decorators.py:36: FutureWarnin
g: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positiona
l argument will be `data`, and passing other arguments without an explicit keyword will result
in an error or misinterpretation.
  warnings.warn(
```



Observation:

For Analysis we have total of 253 Boys and 246 girls ie, 51% are of boys and and remaining of girls. There are 7 boys more than girls.

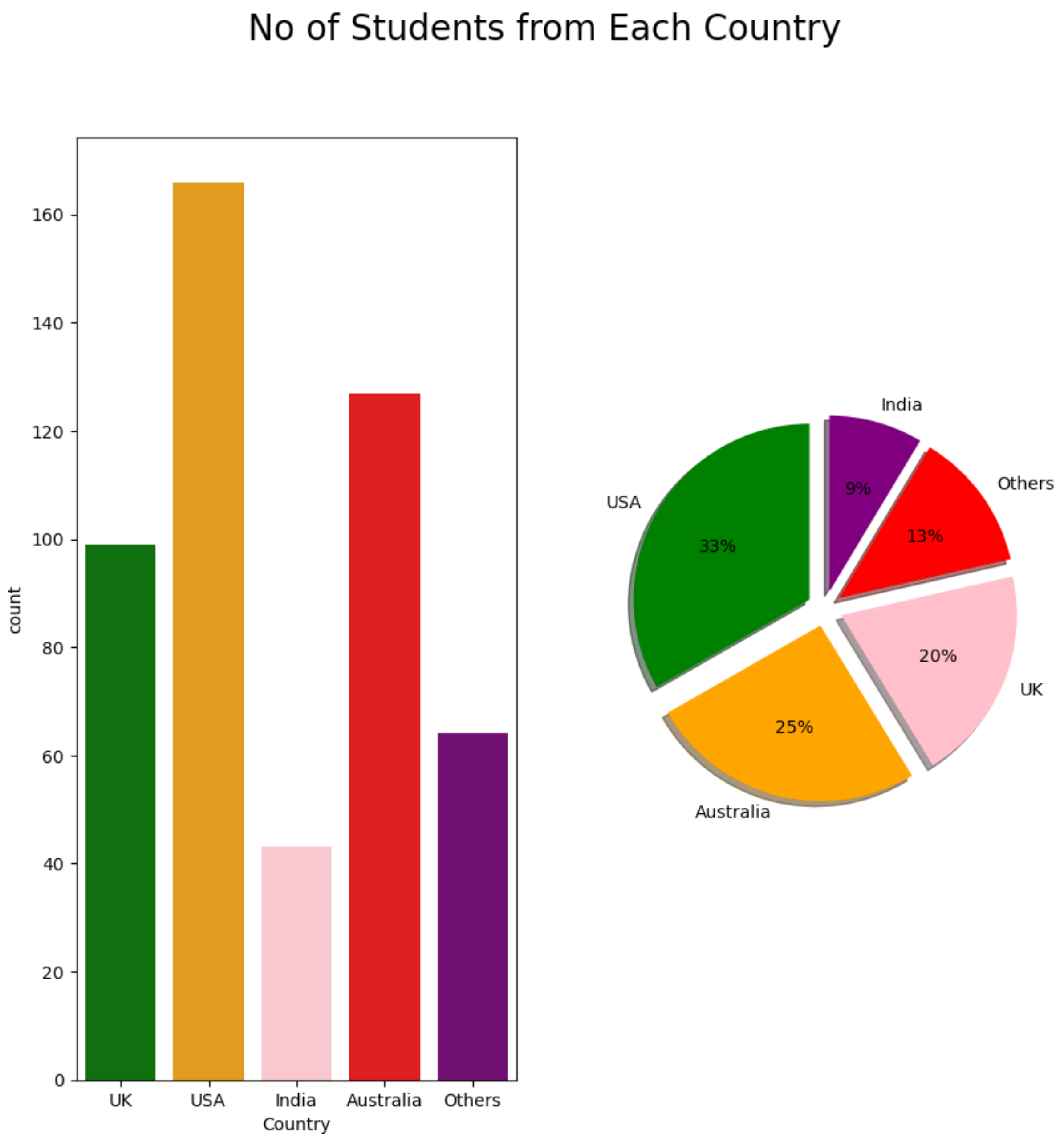
ii) Country

```
In [13]: df['Country'].value_counts()
```

```
Out[13]: USA          166
         Australia    127
         UK            99
         Others        64
         India         43
         Name: Country, dtype: int64
```

In [14]:
```
fig, axes = plt.subplots(ncols=2,
                         nrows=1,
                         figsize=(10, 10),
                         dpi=100)

sns.countplot(df['Country'],palette=['green','orange','pink','red','purple'],ax=axes[0])

labels=['USA','Australia','UK','Others','India']
colors = ('green','orange','pink','red','purple')
axes[1].pie(df['Country'].value_counts(),
            labels=labels,
            autopct='%1.0f%%',
            shadow=True,
            startangle=90,
            explode = (0.1,0.1,0.1,0.1,0.1),
            colors=colors
            )

fig.suptitle('No of Students from Each Country', fontsize=20)
plt.show()
```

```
C:\Users\HP\anaconda3\anacondaorginal\lib\site-packages\seaborn\_decorators.py:36: FutureWarnin
g: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positiona
l argument will be `data`, and passing other arguments without an explicit keyword will result
in an error or misinterpretation.
  warnings.warn(
```



Observation:

The students are mainly from five different countries (UK,USA,India,Australia,Others).

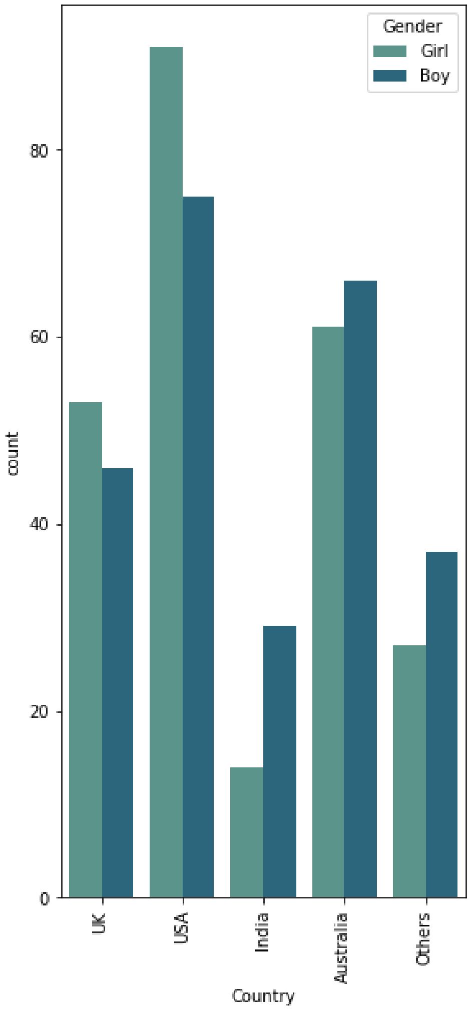
Here others referes to many of the other countries which are considered as a single country.

Majority of the students are from USA (127) and very few are from India (43).

Barplot for finding the no of girls and boys from each Country

In [15]:
```
plt.figure(figsize=(15,10))
plt.subplot(1,3,1)
sns.countplot(x = 'Country',hue='Gender',palette = "crest", data = df)
plt.xticks(rotation = 90)
```

Out[15]:
```
(array([0, 1, 2, 3, 4]), <a list of 5 Text major ticklabel objects>)
```



Observation:

From USA(highest no of students) girls are more in number than boys and for India(lowest no of students) boys are more in no than girls

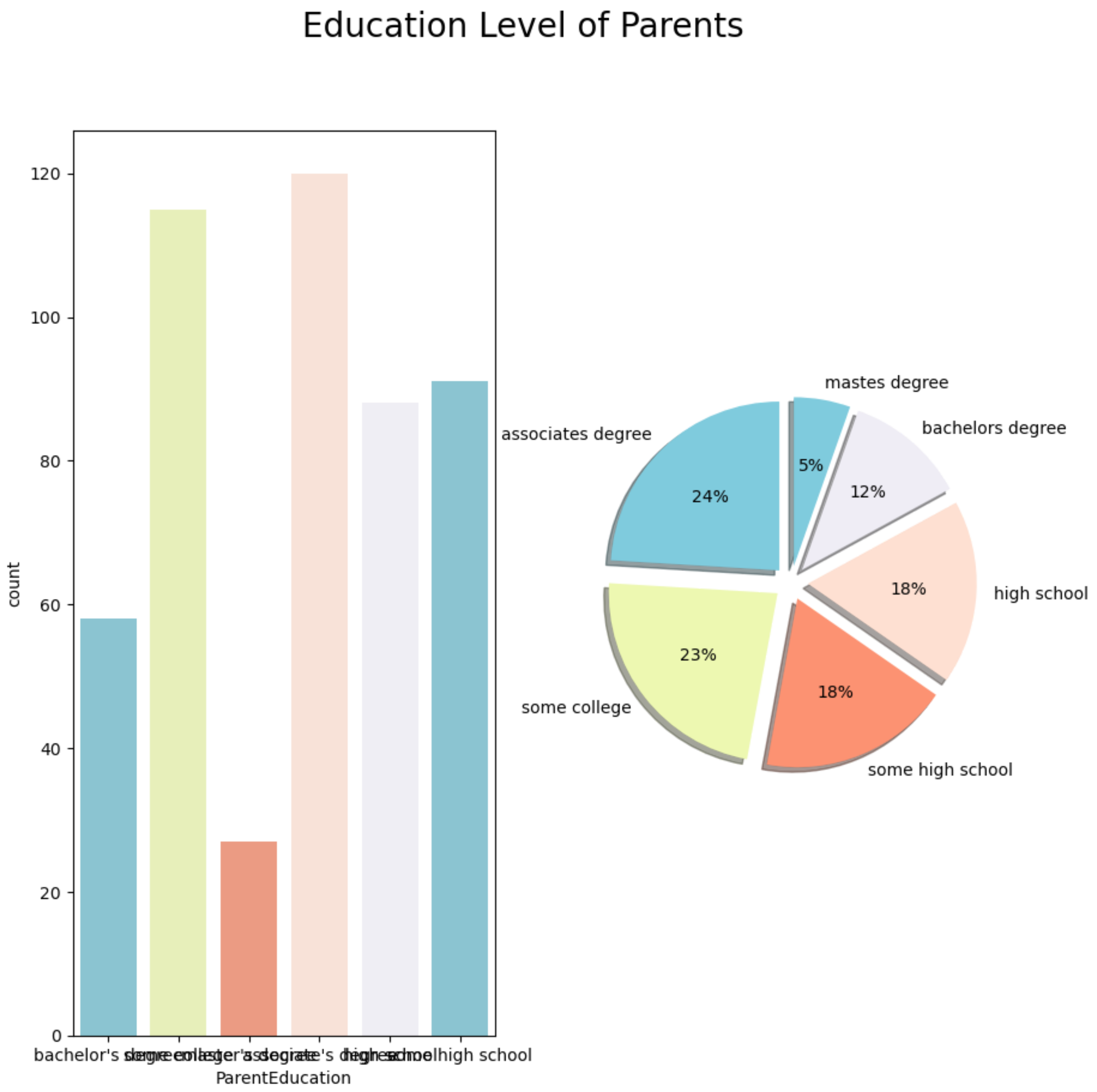
iii) ParentsEducation

In [16]:
```
df['ParentEducation'].value_counts()
```

Out[16]:
```
associate's degree    120
some college          115
some high school       91
high school            88
bachelor's degree      58
master's degree        27
Name: ParentEducation, dtype: int64
```

In [17]:
```
fig, axes = plt.subplots(ncols=2,
                         nrows=1,
                         figsize=(10,10),
                         dpi=100)

sns.countplot(df['ParentEducation'],palette=['#7fcbdd','#edf8b1','#fc9272','#fee0d2','#efedf5'
],ax=axes[0])
labels=['associates degree','some college','some high school','high school','bachelors degree',
'mastes degree']
colors = ('#7fcbdd','#edf8b1','#fc9272','#fee0d2','#efedf5')
axes[1].pie(df['ParentEducation'].value_counts(),
            labels=labels,
            autopct='%1.0f%%',
            shadow=True,
            startangle=90,
            explode = (0.1,0.1,0.1,0.1,0.1,0.1),
            colors=colors
            )

fig.suptitle('Education Level of Parents', fontsize=20)
plt.show()
```

```
C:\Users\HP\anaconda3\anacondaorginal\lib\site-packages\seaborn\_decorators.py:36: FutureWarnin
g: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positiona
l argument will be `data`, and passing other arguments without an explicit keyword will result
in an error or misinterpretation.
  warnings.warn(
```



Observation:

Most of the students parents hold a associate's degree (120) and some college (115) very few of them are only having master's degree(27)

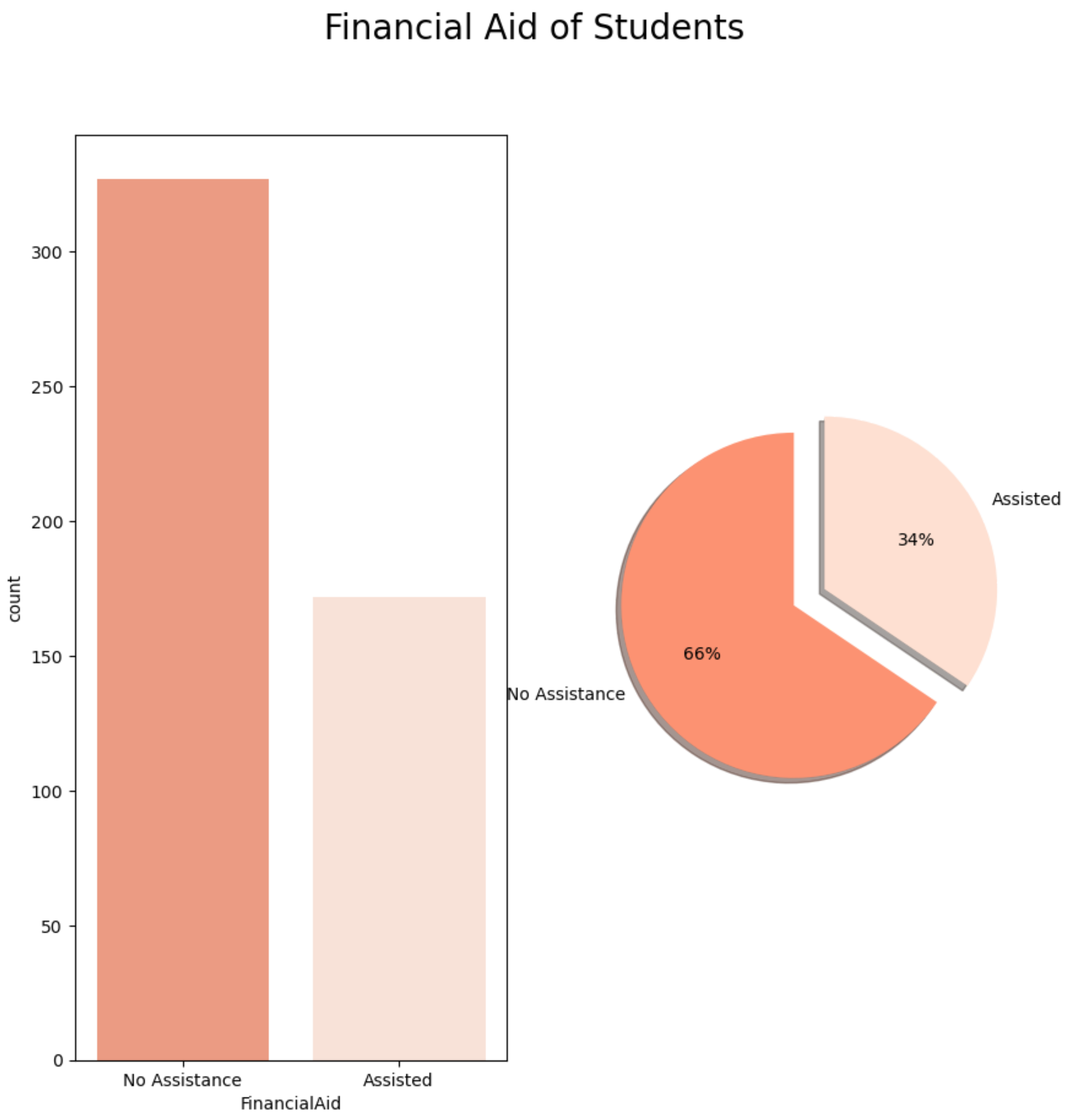
iv) Financial Aid

In [18]:
```
df['FinancialAid'].value_counts()
```

Out[18]:
```
No Assistance    327
Assisted         172
Name: FinancialAid, dtype: int64
```

In [19]:
```
fig, axes = plt.subplots(ncols=2,
                         nrows=1,
                         figsize=(10, 10),
                         dpi=100)

sns.countplot(df['FinancialAid'],palette=['#fc9272','#fee0d2'],ax=axes[0])

labels=['No Assistance','Assisted']
colors = ('#fc9272','#fee0d2',)
axes[1].pie(df['FinancialAid'].value_counts(),
            labels=labels,
            autopct='%1.0f%%',
            shadow=True,
            startangle=90,
            explode = (0.1, 0.1),
            colors=colors
            )

fig.suptitle('Financial Aid of Students', fontsize=20)
plt.show()
```

```
C:\Users\HP\anaconda3\anacondaorginal\lib\site-packages\seaborn\_decorators.py:36: FutureWarnin
g: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positiona
l argument will be `data`, and passing other arguments without an explicit keyword will result
in an error or misinterpretation.
  warnings.warn(
```



Observations:

60% percent of the students are studying in the university without any financial support ie, without a scholarship or loan and the remaning 34% percent are.

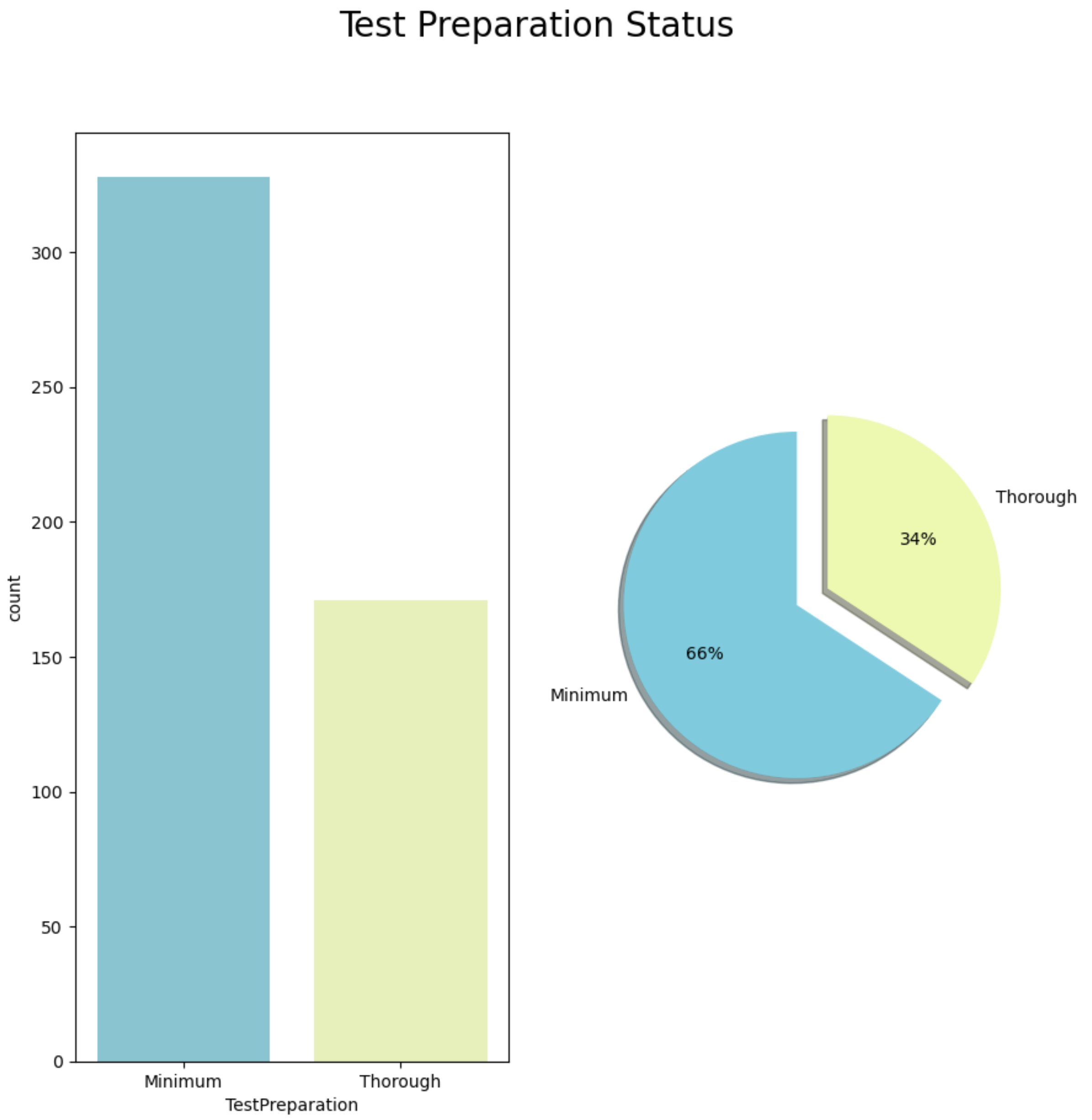
v) TestPreparation Status

In [20]:
```
df['TestPreparation'].value_counts()
```

Out[20]:
```
Minimum     328
Thorough    171
Name: TestPreparation, dtype: int64
```

In [21]:
```
fig, axes = plt.subplots(ncols=2,
                         nrows=1,
                         figsize=(10, 10),
                         dpi=100)

sns.countplot(df['TestPreparation'],palette=['#7fcbdd','#edf8b1'],ax=axes[0])

labels=['Minimum','Thorough ']
colors = ('#7fcbdd','#edf8b1',)
axes[1].pie(df['TestPreparation'].value_counts(),
            labels=labels,
            autopct='%1.0f%%',
            shadow=True,
            startangle=90,
            explode = (0.1, 0.1),
            colors=colors
            )

fig.suptitle('Test Preparation Status', fontsize=20)
plt.show()
```

```
C:\Users\HP\anaconda3\anacondaorginal\lib\site-packages\seaborn\_decorators.py:36: FutureWarnin
g: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positiona
l argument will be `data`, and passing other arguments without an explicit keyword will result
in an error or misinterpretation.
  warnings.warn(
```



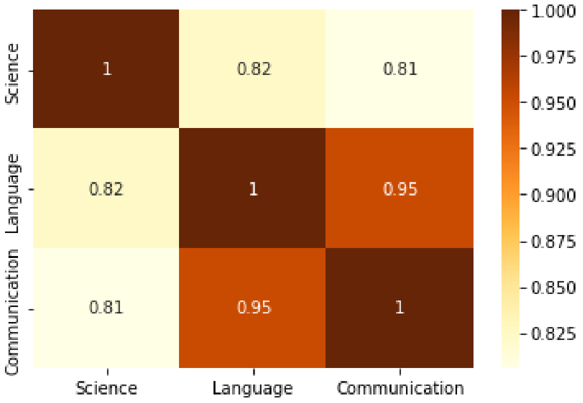
Observation:

1/3 of the students are undertaking the exams without any serious preparation which need to be strictly noted down.

HeatMap to find correlation between continuous Variable

In [22]:

```
sns.heatmap(df.corr(), annot = True, cmap='YlOrBr')

plt.show()
```



Observation:

There is strong correlation between score's of subject.

Plotting the distribution of students marks using Histogram

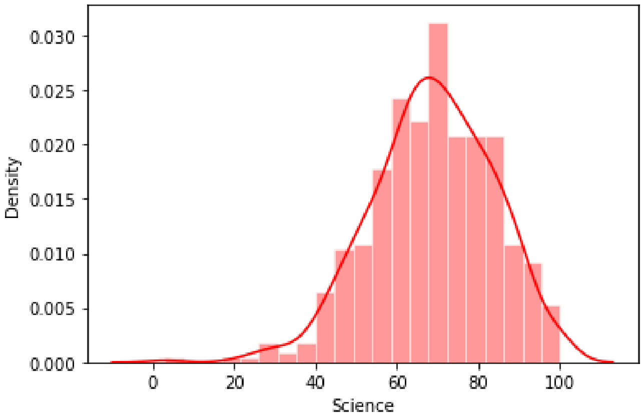
In [23]:

```
sns.distplot(a=df.Science, color='red',
             hist_kws={"edgecolor": 'white'})
```

```
C:\Users\HP\anaconda3\anacondaorginal\lib\site-packages\seaborn\distributions.py:2619: FutureWa
rning: `distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) or `hist
plot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[23]:

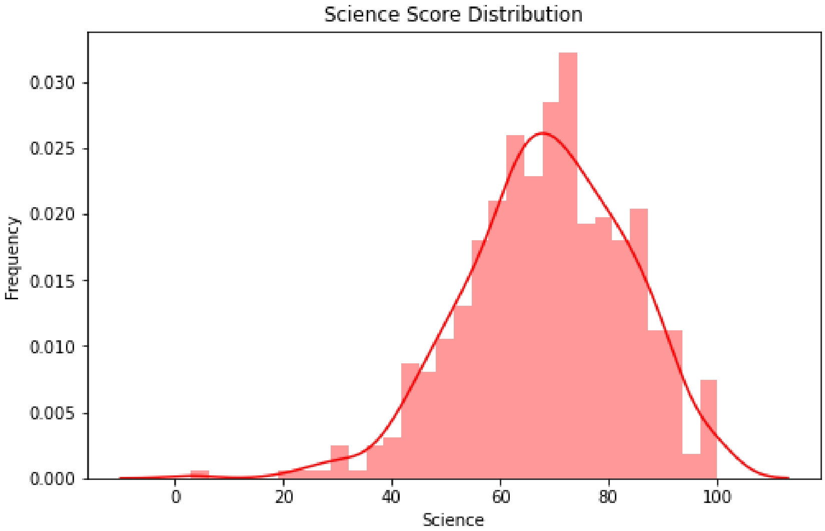
```
<matplotlib.axes._subplots.AxesSubplot at 0x2481e0d8610>
```



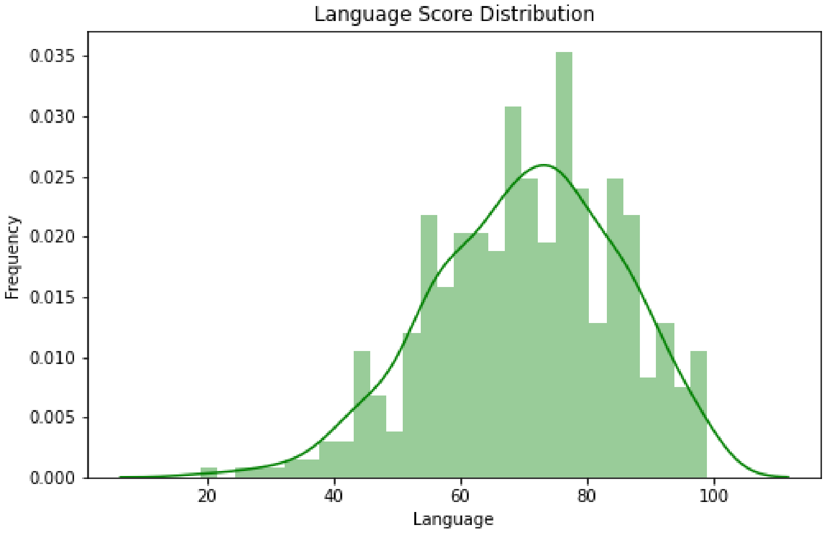
In [24]:

```
 plt.figure(figsize=(8,5))
sns.distplot(df['Science'], kde = True, color='r', bins = 30)
plt.ylabel('Frequency')
plt.title('Science Score Distribution')
plt.show()

plt.figure(figsize=(8,5))
sns.distplot(df['Language'], kde = True, color='g', bins = 30)
plt.ylabel('Frequency')
plt.title('Language Score Distribution')
plt.show()

plt.figure(figsize=(8,5))
sns.distplot(df['Communication'], kde = True, color='y', bins = 30)
plt.ylabel('Frequency')
plt.title('Communication Score Distribution')
plt.show()
```

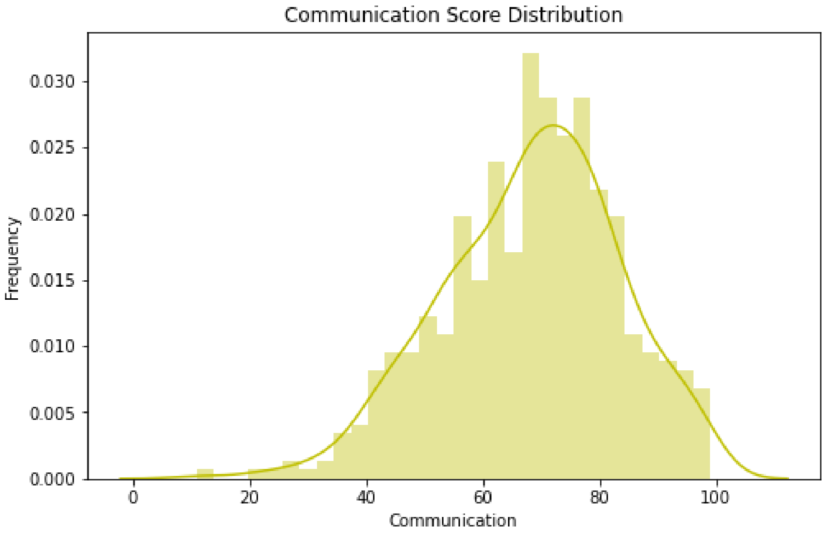
```
C:\Users\HP\anaconda3\anacondaorginal\lib\site-packages\seaborn\distributions.py:2619: FutureWa
rning: `distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) or `hist
plot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



```
C:\Users\HP\anaconda3\anacondaorginal\lib\site-packages\seaborn\distributions.py:2619: FutureWa
rning: `distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) or `hist
plot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



```
C:\Users\HP\anaconda3\anacondaorginal\lib\site-packages\seaborn\distributions.py:2619: FutureWa
rning: `distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) or `hist
plot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



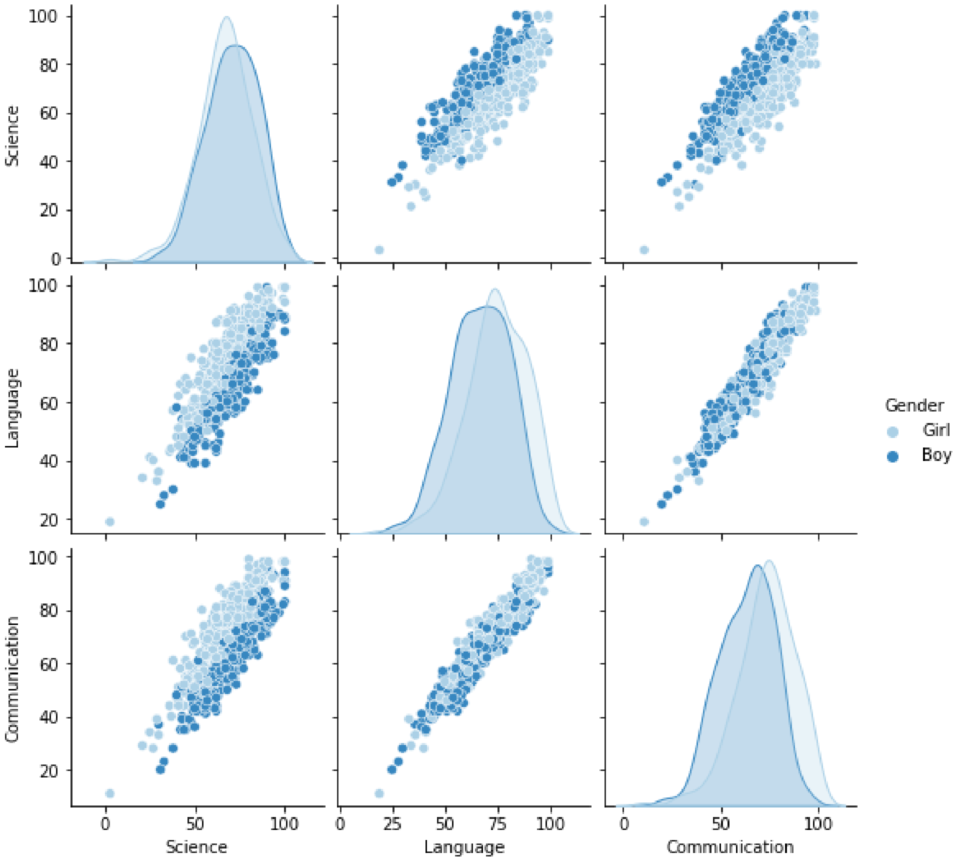
Observation:

For all the three subject, it is negatively skewed means average of the scores of the students lies between 50 and 75.

Thus, the performnce of the students need to be improved in three subject.

Pairplot for finding relationship between combination fo variable

In [25]:
```
sns.pairplot(data=df,palette='Blues',hue='Gender')
plt.show()
```



Observation:

There is a strong correlation between each of the variable

Calculating the total marks

In [26]:

```
df['Total']=df['Language'] + df['Communication']+df['Science']
(df)
```

Out[26]:

	Gender	Country	ParentEducation	FinancialAid	TestPreparation	Science	Language	Communication	Total
0	Girl	UK	bachelor's degree	No Assistance	Minimum	75	74	75	224
1	Girl	USA	some college	No Assistance	Thorough	72	92	89	253
2	Girl	UK	master's degree	No Assistance	Minimum	93	97	94	284
3	Boy	India	associate's degree	Assisted	Minimum	50	59	45	154
4	Boy	USA	some college	No Assistance	Minimum	79	80	76	235
...	...	...	...	...	...	...	...	...	...
494	Girl	UK	high school	No Assistance	Minimum	57	66	69	192
495	Boy	Australia	high school	No Assistance	Thorough	71	66	67	204
496	Girl	USA	some college	No Assistance	Minimum	57	50	53	160
497	Girl	Australia	some college	Assisted	Thorough	62	80	77	219
498	Girl	UK	some high school	No Assistance	Minimum	69	71	69	209

499 rows × 9 columns

Observation:

In order to find the overall perfomance of each students total marks of subjects are calculated together.

All of the three columns of marks are added together and the result is added into a new column called Total.

These values can furthur help in assigning final grade and percentage.

Finding Percentage

In [27]:

```
df['Percent'] = (df['Total'] /300*100)
df
```

Out[27]:

	Gender	Country	ParentEducation	FinancialAid	TestPreparation	Science	Language	Communication	Total	P
0	Girl	UK	bachelor's degree	No Assistance	Minimum	75	74	75	224	74.6
1	Girl	USA	some college	No Assistance	Thorough	72	92	89	253	84.3
2	Girl	UK	master's degree	No Assistance	Minimum	93	97	94	284	94.6
3	Boy	India	associate's degree	Assisted	Minimum	50	59	45	154	51.3
4	Boy	USA	some college	No Assistance	Minimum	79	80	76	235	78.3
...	...	...	...	...	...	...	...	...	...	
494	Girl	UK	high school	No Assistance	Minimum	57	66	69	192	64.0
495	Boy	Australia	high school	No Assistance	Thorough	71	66	67	204	68.0
496	Girl	USA	some college	No Assistance	Minimum	57	50	53	160	53.3
497	Girl	Australia	some college	Assisted	Thorough	62	80	77	219	73.0
498	Girl	UK	some high school	No Assistance	Minimum	69	71	69	209	69.6

499 rows × 10 columns

Observation

Now inorder to make the calculation easy we findpercentage of each student dy dividing total marks with maximum marks(300).

Assigning the Grades

In order to categorize each of the students into different category based on each of subject level marks we assign a categorical value for each the continuous values.

In [28]:

```
def determine_grade(scores):
    if scores >= 85 and scores <= 100:
        return 'A'
    elif scores >= 70 and scores < 85:
        return 'B'
    elif scores >= 55 and scores < 70:
        return 'C'
    elif scores >= 35 and scores < 55:
        return 'D'
    elif scores >= 0 and scores < 35:
        return 'Fail'

df['Science_Grades']=df['Science'].apply(determine_grade)
df
```

Out[28]:

	Gender	Country	ParentEducation	FinancialAid	TestPreparation	Science	Language	Communication	Total	P
0	Girl	UK	bachelor's degree	No Assistance	Minimum	75	74	75	224	74.6
1	Girl	USA	some college	No Assistance	Thorough	72	92	89	253	84.3
2	Girl	UK	master's degree	No Assistance	Minimum	93	97	94	284	94.6
3	Boy	India	associate's degree	Assisted	Minimum	50	59	45	154	51.3
4	Boy	USA	some college	No Assistance	Minimum	79	80	76	235	78.3
...	...	...	...	...	...	...	...	...	...	
494	Girl	UK	high school	No Assistance	Minimum	57	66	69	192	64.0
495	Boy	Australia	high school	No Assistance	Thorough	71	66	67	204	68.0
496	Girl	USA	some college	No Assistance	Minimum	57	50	53	160	53.3
497	Girl	Australia	some college	Assisted	Thorough	62	80	77	219	73.0
498	Girl	UK	some high school	No Assistance	Minimum	69	71	69	209	69.6

499 rows × 11 columns

In [29]:

```
def determine_grade(scores):
    if scores >= 85 and scores <= 100:
        return 'A'
    elif scores >= 70 and scores < 85:
        return 'B'
    elif scores >= 55 and scores < 70:
        return 'C'
    elif scores >= 35 and scores < 55:
        return 'D'
    elif scores >= 0 and scores < 35:
        return 'Fail'

df['Language_Grades']=df['Language'].apply(determine_grade)
df
```

Out[29]:

	Gender	Country	ParentEducation	FinancialAid	TestPreparation	Science	Language	Communication	Total	P
0	Girl	UK	bachelor's degree	No Assistance	Minimum	75	74	75	224	74.6
1	Girl	USA	some college	No Assistance	Thorough	72	92	89	253	84.3
2	Girl	UK	master's degree	No Assistance	Minimum	93	97	94	284	94.6
3	Boy	India	associate's degree	Assisted	Minimum	50	59	45	154	51.3
4	Boy	USA	some college	No Assistance	Minimum	79	80	76	235	78.3
...	...	...	...	...	...	...	...	...	...	
494	Girl	UK	high school	No Assistance	Minimum	57	66	69	192	64.0
495	Boy	Australia	high school	No Assistance	Thorough	71	66	67	204	68.0
496	Girl	USA	some college	No Assistance	Minimum	57	50	53	160	53.3
497	Girl	Australia	some college	Assisted	Thorough	62	80	77	219	73.0
498	Girl	UK	some high school	No Assistance	Minimum	69	71	69	209	69.6

499 rows × 12 columns

In [30]:

```python
def determine_grade(scores):
    if scores >= 85 and scores <= 100:
        return 'A'
    elif scores >= 70 and scores < 85:
        return 'B'
    elif scores >= 55 and scores < 70:
        return 'C'
    elif scores >= 35 and scores < 55:
        return 'D'
    elif scores >= 0 and scores < 35:
        return 'Fail'

df['Communication_Grades']=df['Communication'].apply(determine_grade)
df.head(34)
```

Out[30]:

	Gender	Country	ParentEducation	FinancialAid	TestPreparation	Science	Language	Communication	Total	Pe
0	Girl	UK	bachelor's degree	No Assistance	Minimum	75	74	75	224	74.66
1	Girl	USA	some college	No Assistance	Thorough	72	92	89	253	84.33
2	Girl	UK	master's degree	No Assistance	Minimum	93	97	94	284	94.66
3	Boy	India	associate's degree	Assisted	Minimum	50	59	45	154	51.33
4	Boy	USA	some college	No Assistance	Minimum	79	80	76	235	78.33
5	Girl	UK	associate's degree	No Assistance	Minimum	74	85	79	238	79.33
6	Girl	UK	some college	No Assistance	Thorough	91	97	93	281	93.66
7	Boy	UK	some college	Assisted	Minimum	43	45	40	128	42.66
8	Boy	Australia	high school	Assisted	Thorough	67	66	68	201	67.00
9	Girl	UK	high school	Assisted	Minimum	41	62	51	154	51.33
10	Boy	USA	associate's degree	No Assistance	Minimum	61	56	53	170	56.66
11	Boy	Australia	associate's degree	No Assistance	Minimum	43	54	44	141	47.00
12	Girl	UK	high school	No Assistance	Minimum	68	83	74	225	75.00
13	Boy	India	some college	No Assistance	Thorough	81	74	71	226	75.33
14	Girl	India	master's degree	No Assistance	Minimum	53	55	59	167	55.66
15	Girl	USA	some high school	No Assistance	Minimum	72	77	79	228	76.00
16	Boy	USA	high school	No Assistance	Minimum	91	91	87	269	89.66
17	Girl	UK	some high school	Assisted	Minimum	21	34	29	84	28.00
18	Boy	USA	master's degree	Assisted	Thorough	49	44	47	140	46.66
19	Girl	USA	associate's degree	Assisted	Minimum	57	60	62	179	59.66
20	Boy	Australia	high school	No Assistance	Minimum	69	71	64	204	68.00
21	Girl	UK	some college	Assisted	Thorough	68	77	71	216	72.00
22	Boy	Australia	some college	No Assistance	Minimum	47	56	54	157	52.33
23	Girl	USA	some high school	No Assistance	Minimum	72	75	74	221	73.66
24	Boy	Australia	bachelor's degree	Assisted	Thorough	77	73	81	231	77.00
25	Boy	India	master's degree	Assisted	Minimum	76	76	73	225	75.00
26	Boy	UK	some college	No Assistance	Minimum	72	56	56	184	61.33
27	Girl	USA	bachelor's degree	No Assistance	Minimum	70	71	76	217	72.33
28	Boy	USA	high school	No Assistance	Minimum	73	72	66	211	70.33
29	Girl	Australia	master's degree	No Assistance	Minimum	65	72	76	213	71.00
30	Girl	Australia	some college	No Assistance	Minimum	72	76	75	223	74.33
31	Girl	UK	some college	No Assistance	Minimum	66	67	62	195	65.00
32	Girl	Others	master's degree	Assisted	Minimum	59	74	66	199	66.33

	Gender	Country	ParentEducation	FinancialAid	TestPreparation	Science	Language	Communication	Total	Pe
33	Boy	Australia	some college	No Assistance	Minimum	43	44	39	126	42.0

observation:

The marks are divided into each category

ie,scores >= 85 and scores <= 100-Grade A

scores >= 70 and scores < 85-Grade B

scores >= 55 and scores < 70-Grade C

scores >= 35 and scores < 55-Grade D

scores >= 0 and scores < 35-Grade Fail

Finding The overall Performance

In [31]:

```
def determine_result(scores):
    if scores >= 85 and scores <= 100:
        return 'Oustanding'
    elif scores >= 80 and scores < 85:
        return 'Excellent'
    elif scores >= 55 and scores < 80:
        return 'First class'
    elif scores >= 35 and scores < 55:
        return 'second cass'
    elif scores >= 0 and scores < 35:
        return 'Fail'

df['Result']=df['Percent'].apply(determine_result)
df
```

Out[31]:

	Gender	Country	ParentEducation	FinancialAid	TestPreparation	Science	Language	Communication	Total	P
0	Girl	UK	bachelor's degree	No Assistance	Minimum	75	74	75	224	74.6
1	Girl	USA	some college	No Assistance	Thorough	72	92	89	253	84.3
2	Girl	UK	master's degree	No Assistance	Minimum	93	97	94	284	94.6
3	Boy	India	associate's degree	Assisted	Minimum	50	59	45	154	51.3
4	Boy	USA	some college	No Assistance	Minimum	79	80	76	235	78.3
...	...	...	...	...	...	...	...	...	...	
494	Girl	UK	high school	No Assistance	Minimum	57	66	69	192	64.0
495	Boy	Australia	high school	No Assistance	Thorough	71	66	67	204	68.0
496	Girl	USA	some college	No Assistance	Minimum	57	50	53	160	53.3
497	Girl	Australia	some college	Assisted	Thorough	62	80	77	219	73.0
498	Girl	UK	some high school	No Assistance	Minimum	69	71	69	209	69.6

499 rows × 14 columns

Observation:

Based on their overall performance they are awarded into each category

ie,scores >= 85 and scores <= 100-Outstanding

scores >= 80 and scores < 85-Excellent

scores >= 55 and scores < 80-First class

scores >= 35 and scores < 55-Second class

scores >= 0 and scores < 35-Fail

Renaming The columns

For Better clarity

In [32]:
```
df.rename(columns = {'Total':'Total_Marks'}, inplace = True)
df
```

Out[32]:

	Gender	Country	ParentEducation	FinancialAid	TestPreparation	Science	Language	Communication	Total_Mark
0	Girl	UK	bachelor's degree	No Assistance	Minimum	75	74	75	22
1	Girl	USA	some college	No Assistance	Thorough	72	92	89	25
2	Girl	UK	master's degree	No Assistance	Minimum	93	97	94	28
3	Boy	India	associate's degree	Assisted	Minimum	50	59	45	15
4	Boy	USA	some college	No Assistance	Minimum	79	80	76	23
...	...	...	...	...	...	...	...	...	
494	Girl	UK	high school	No Assistance	Minimum	57	66	69	19
495	Boy	Australia	high school	No Assistance	Thorough	71	66	67	20
496	Girl	USA	some college	No Assistance	Minimum	57	50	53	16
497	Girl	Australia	some college	Assisted	Thorough	62	80	77	21
498	Girl	UK	some high school	No Assistance	Minimum	69	71	69	20

499 rows × 14 columns

Rearraging the columns for a easy unserstanding

In [33]:
```
df = df[['Gender', 'Country', 'ParentEducation','FinancialAid','TestPreparation','Science','Sci
ence_Grades','Language','Language_Grades','Communication','Communication_Grades','Total_Marks',
'Percent','Result']]
```

In [34]:
```
df
```

Out[34]:

	Gender	Country	ParentEducation	FinancialAid	TestPreparation	Science	Science_Grades	Language	Language_
0	Girl	UK	bachelor's degree	No Assistance	Minimum	75	B	74	
1	Girl	USA	some college	No Assistance	Thorough	72	B	92	
2	Girl	UK	master's degree	No Assistance	Minimum	93	A	97	
3	Boy	India	associate's degree	Assisted	Minimum	50	D	59	
4	Boy	USA	some college	No Assistance	Minimum	79	B	80	
...	...	...	...	...	...	...	...	...	
494	Girl	UK	high school	No Assistance	Minimum	57	C	66	
495	Boy	Australia	high school	No Assistance	Thorough	71	B	66	
496	Girl	USA	some college	No Assistance	Minimum	57	C	50	
497	Girl	Australia	some college	Assisted	Thorough	62	C	80	
498	Girl	UK	some high school	No Assistance	Minimum	69	C	71	

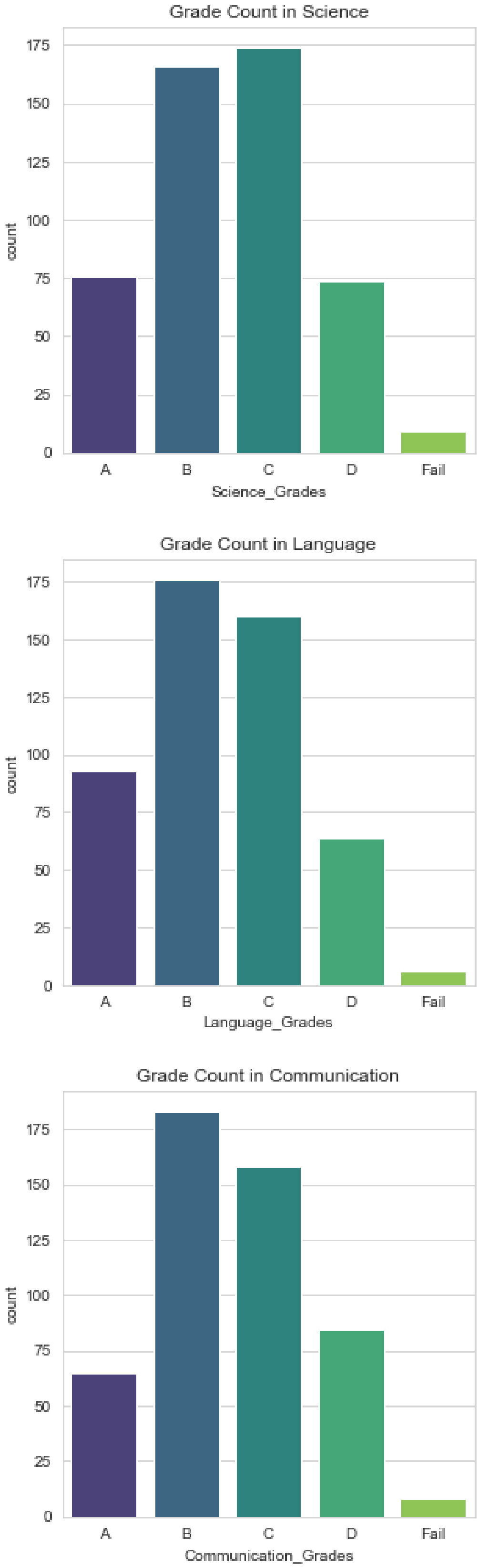
499 rows × 14 columns

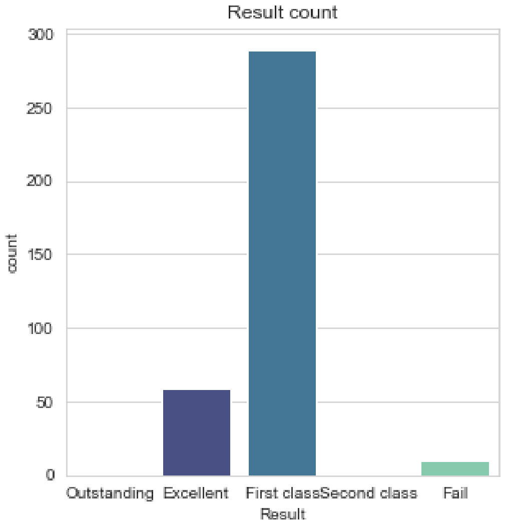
Plotting Grade Statistics

In [35]:

```
sns.set_style('whitegrid')
plt.figure(figsize=(16,5))
plt.subplot(1,3,1)
sns.countplot(x ='Science_Grades', data = df,order = ['A','B','C','D','Fail'],palette='viridis'
)
plt.title('Grade Count in Science')

sns.set_style('whitegrid')
plt.figure(figsize=(16,5))
plt.subplot(1,3,1)
sns.countplot(x ='Language_Grades', data = df,order = ['A','B','C','D','Fail'],palette='viridi
s')
plt.title('Grade Count in Language')

sns.set_style('whitegrid')
plt.figure(figsize=(16,5))
plt.subplot(1,3,1)
sns.countplot(x ='Communication_Grades', data = df,order = ['A','B','C','D','Fail'],palette='vi
ridis')
plt.title('Grade Count in Communication')


sns.set_style('whitegrid')
plt.figure(figsize=(16,5))
plt.subplot(1,3,1)
sns.countplot(x ='Result', data = df,order = ['Outstanding','Excellent','First class','Second c
lass','Fail'],palette='mako')
plt.title('Result count')
```

```
Out[35]: Text(0.5, 1.0, 'Result count')
```





For Science,Majority of the students are awarded with C and B grade only few have scored A grade and which is not much compared with D grade.Thus, the overall performance of the class is average only.

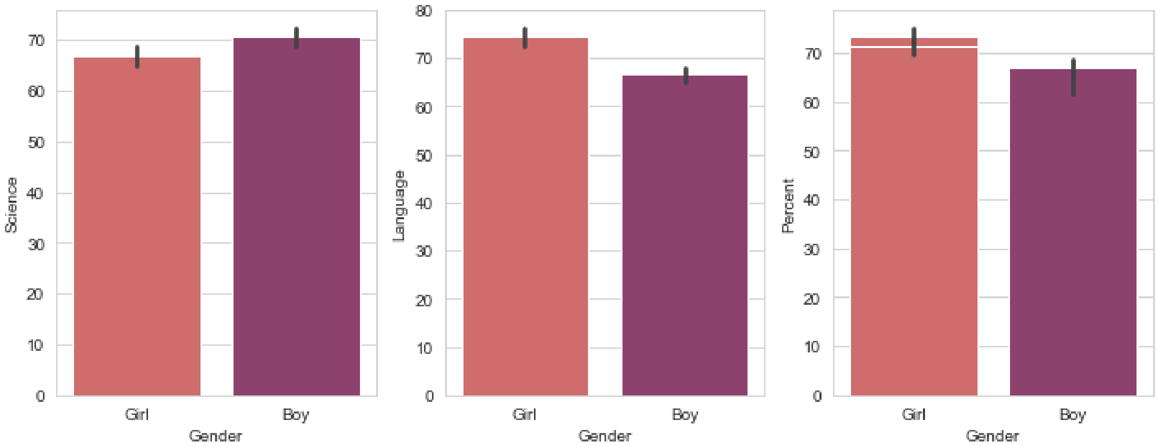
Same goes with other two subjects also.

Science teacher need to have a serious look into making the no of students in first class more than second class to keep up with a moderate performance of class.

Overall, around 90% percent of the students have won first class which show class is having a moderate performance but there is a lack of outstanding performance so teachers should focus into that also.

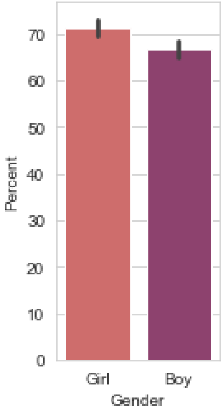
In [36]:
```
plt.figure(figsize=(10,4))

plt.subplot(1,3,1)
sns.barplot(x = 'Gender', y = 'Science',palette = "flare", data = df)

plt.subplot(1,3,2)
sns.barplot(x = 'Gender', y = 'Language',palette = "flare", data = df)

plt.subplot(1,3,3)
sns.barplot(x = 'Gender', y = 'Communication',palette = "flare", data = df)

plt.subplot(1,3,3)
sns.barplot(x = 'Gender', y = 'Percent',palette = "flare", data = df)

plt.tight_layout()
```

```
<ipython-input-36-a6463d86efbc>:12: MatplotlibDeprecationWarning: Adding an axes using the same
arguments as a previous axes currently reuses the earlier instance.  In a future version, a new
instance will always be created and returned.  Meanwhile, this warning can be suppressed, and t
he future behavior ensured, by passing a unique label to each axes instance.
  plt.subplot(1,3,3)
```



In [37]:
```
plt.subplot(1,3,3)
sns.barplot(x = 'Gender', y = 'Percent',palette = "flare", data = df)
```

Out[37]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x2481d560be0>
```



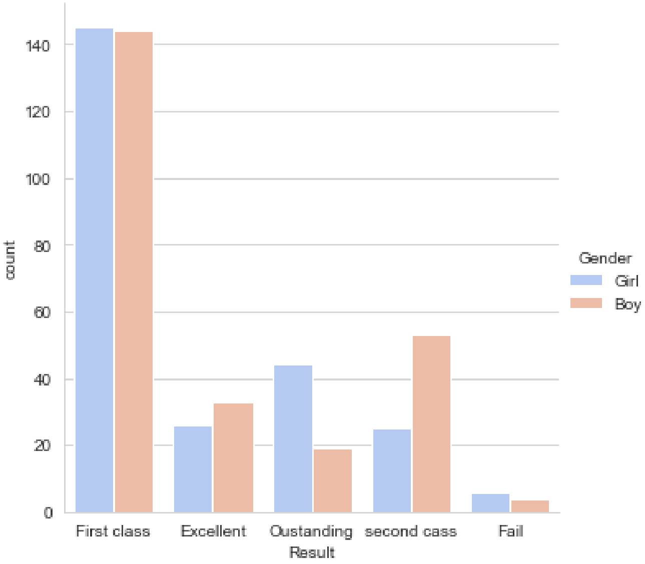
Observation:

For science, boys have outshined than girls for the other two girls have performed more than boys.

Overall girls have performed well than boys.

In [206]:
```
sns.catplot(x="Result", kind="count",hue='Gender',palette='coolwarm', data=df)
```

Out[206]:
```
<seaborn.axisgrid.FacetGrid at 0x24bcc8c68b0>
```



Observation:

Girls have high perfomance than boys(excelllent and outstanding result) but looking into the average result both of them have same result (first class) and boys are the one who performed very badly than girls (second class) and fail ratio is very few for both.

Teacher should care academics of boys more than girls.

Grade Statistics of each Subject

In [43]:
```
print('-------- GRADE STATISTICS --------')
print('====SCIENCE GRADE ====')
print(df['Science_Grades'].value_counts())
print('==== LANGUAGE GRADE ====')
print(df['Language_Grades'].value_counts())
print('==== COMMUNICATION GRADE ====')
print(df['Communication_Grades'].value_counts())
```

```
-------- GRADE STATISTICS --------
====SCIENCE GRADE ====
C       174
B       166
A        76
D        74
Fail      9
Name: Science_Grades, dtype: int64
==== LANGUAGE GRADE ====
B       176
C       160
A        93
D        64
Fail      6
Name: Language_Grades, dtype: int64
==== COMMUNICATION GRADE ====
B       183
C       158
D        85
A        65
Fail      8
Name: Communication_Grades, dtype: int64
```

Status of each subject

In [212]:
```
print('------STATUS OF SCIENCE SUBJECT-----')
print('----Females----')
print('Max. Science Score: ', df[df['Gender'] == 'Girl']['Science'].max())
print('Min. Science Score: ', df[df['Gender'] == 'Girl']['Science'].min())
print('Average Science Score: ',df[df['Gender'] == 'Girl']['Science'].mean())
print('----Males----')
print('Max. Science Score: ', df[df['Gender'] == 'Boy']['Science'].max())
print('Min. Science Score:', df[df['Gender'] == 'Boy']['Science'].min())
print('Average Science Score ', df[df['Gender'] == 'Boy']['Science'].mean())
```

```
------STATUS OF SCIENCE SUBJECT-----
----Females----
Max. Science Score:  100
Min. Science Score:  3
Average Science Score:  66.65447154471545
----Males----
Max. Science Score:  100
Min. Science Score: 30
Average Science Score  70.60079051383399
```

```
In [213]: print('------STATUS OF LANGUAGE SUBJECT-----')
          print('----Females----')
          print('Max.language Score: ', df[df['Gender'] == 'Girl']['Language'].max())
          print('Min. Language Score: ', df[df['Gender'] == 'Girl']['Language'].min())
          print('Average Language Score: ',df[df['Gender'] == 'Girl']['Language'].mean())
          print('----Males----')
          print('Max. Language Score: ', df[df['Gender'] == 'Boy']['Language'].max())
          print('Min. Language Score:', df[df['Gender'] == 'Boy']['Language'].min())
          print('Average Language Score ', df[df['Gender'] == 'Boy']['Language'].mean())
```

```
------STATUS OF LANGUAGE SUBJECT-----
----Females----
Max.language Score:  99
Min. Language Score:  19
Average Language Score:  74.47560975609755
----Males----
Max. Language Score:  99
Min. Language Score: 25
Average Language Score  66.50592885375494
```

```
In [217]: print('------STATUS OF COMMUNICATION SUBJECT-----')
          print('----Females----')
          print('Max.Communication Score: ', df[df['Gender'] == 'Girl']['Communication'].max())
          print('Min. Communication Score: ', df[df['Gender'] == 'Girl']['Communication'].min())
          print('Average Communication Score: ',df[df['Gender'] == 'Girl']['Communication'].mean())
          print('----Males----')
          print('Max. Communication Score: ', df[df['Gender'] == 'Boy']['Communication'].max())
          print('Min. Communication Score:', df[df['Gender'] == 'Boy']['Communication'].min())
          print('Average Communication Score ', df[df['Gender'] == 'Boy']['Communication'].mean())
```

```
------STATUS OF COMMUNICATION SUBJECT-----
----Females----
Max.Communication Score:  99
Min. Communication Score:  11
Average Communication Score:  73.32520325203252
----Males----
Max. Communication Score:  96
Min. Communication Score: 20
Average Communication Score  63.284584980237156
```

No of students having maximum grades in each subject

```
In [218]: print('No. of students having maximum grade in Science: ', len(df[df['Science_Grades'] == 'A'
          ]))
          print('No. of students having maximum grade in Language: ', len(df[df['Language_Grades'] == 'A'
          ]))
          print('No. of students having maximum grade in Communication: ', len(df[df['Communication_Grade
          s'] == 'A']))
```

```
No. of students having maximum grade in Science:  76
No. of students having maximum grade in Language:  93
No. of students having maximum grade in Communication:  65
```

No of students having minimum grades in each subject

```
In [219]: print('No. of students having minimum grade in Science: ', len(df[df['Science_Grades'] == 'Fai
          l']))
          print('No. of students having minimum grade in Language: ', len(df[df['Language_Grades'] == 'Fa
          il']))
          print('No. of students having minimum grade in Communication: ', len(df[df['Communication_Grade
          s'] == 'Fail']))
```

```
No. of students having minimum grade in Science:  9
No. of students having minimum grade in Language:  6
No. of students having minimum grade in Communication:  8
```

Student who secured highest mark in all Exams

In [59]:

```
Highest_in_Science= df['Science'] == 100
Highest_in_Language= df['Language'] == 99
Highest_in_Communcation = df['Communication'] == 99

Highest_scorer= df[(Highest_in_Science) & (Highest_in_Language) & (Highest_in_Communcation)]
Highest_scorer
```

Out[59]:

	Gender	Country	ParentEducation	FinancialAid	TestPreparation	Science	Science_Grades	Language	Language_Gr

Person who secured Lowest marks in all Exams

In [220]:

```
Lowest_in_Science= df['Science'] == 3
Lowest_in_Language= df['Language'] == 19
Lowest_in_Communcation = df['Communication'] == 11

Lowest_scorer= df[(Lowest_in_Science) & (Lowest_in_Language) & (Lowest_in_Communcation)]
Lowest_scorer
```

Out[220]:

	Gender	Country	ParentEducation	FinancialAid	TestPreparation	Science	Science_Grades	Language	Language_(
59	Girl	USA	some high school	Assisted	Minimum	3	Fail	19	

Observation

No student have secured highest in all subjects only 1 have secured minimum marks in all subjects

Total no of students failed

In [221]:

```
#Failed Students
failed_students = df[(Lowest_in_Science) | (Lowest_in_Language)|(Lowest_in_Communcation)]
failed = len(failed_students)
print('Total Number of students who failed are: {}' .format(len(failed_students)))
```

```
Total Number of students who failed are: 1
```

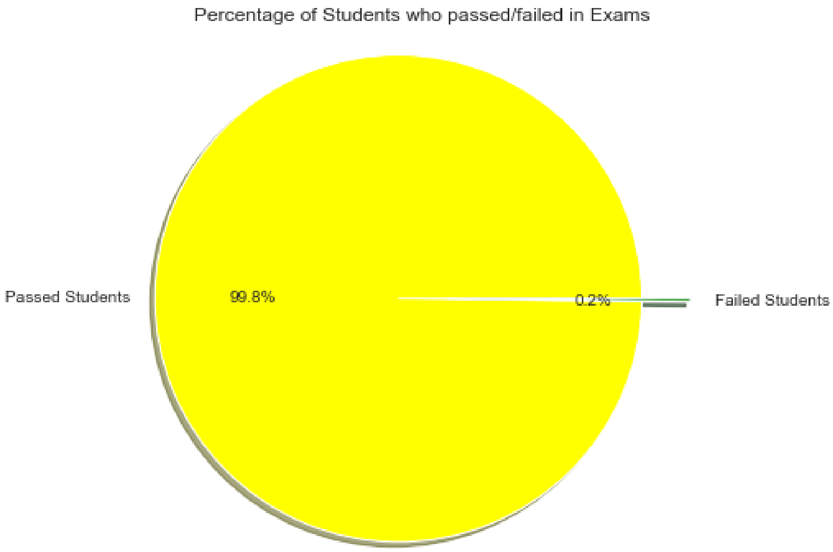
Total no of students Passed

In [222]:

```
#Passed Students
passed_students = len(df) - len(failed_students)
print('Total Number of students who passed are: {}' .format(passed_students))
```

```
Total Number of students who passed are: 498
```

```
In [223]: plt.figure(figsize=(8,6))

          #Data to plot
          labels = 'Passed Students', 'Failed Students'
          sizes = [passed_students,failed]
          colors = ['yellow','green']
          explode = (.2,0)

          #Plot
          plt.pie(sizes,explode = explode, labels = labels,colors = colors,
                 autopct='%1.1f%%',shadow = True, startangle=360)
          plt.axis('equal')
          plt.title('Percentage of Students who passed/failed in Exams')
          plt.show()
```

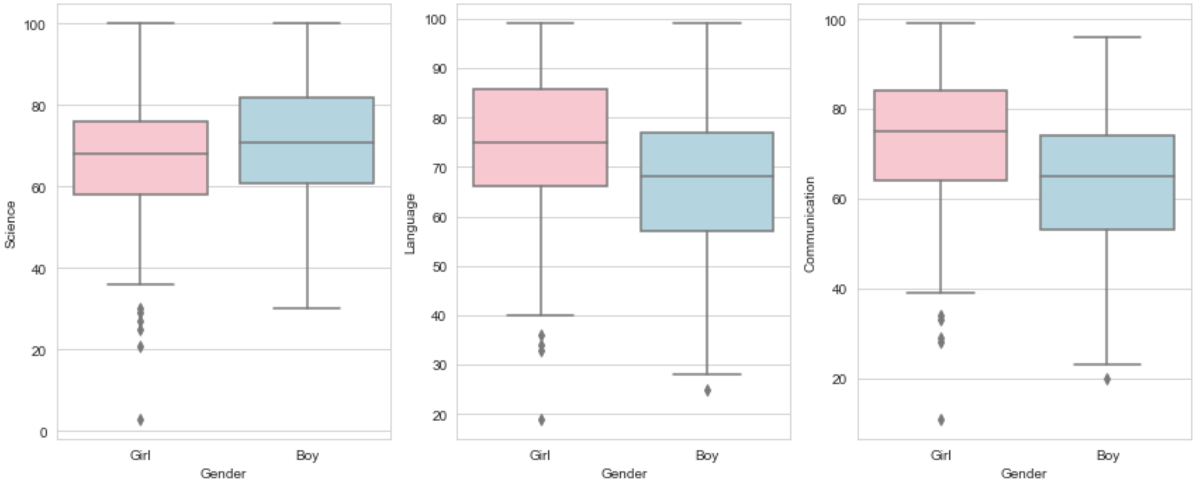


Observation

Majority(99.8%) of students passed in all the three subjects.Only 0.2% students failed in atleast one of the three subjects.

Boxplot to find the difference between marks distribution for each subject for girls and Boys

```
In [64]: plt.figure(figsize=(12,5))

         plt.subplot(1,3,1)
         sns.boxplot(x = 'Gender', y = 'Science', data = df,palette = ['pink', 'lightblue'])

         plt.subplot(1,3,2)
         sns.boxplot(x = 'Gender', y = 'Language', data = df,palette = ['pink', 'lightblue'])

         plt.subplot(1,3,3)
         sns.boxplot(x = 'Gender', y = 'Communication', data = df,palette = ['pink', 'lightblue'])

         plt.tight_layout()
```



Observation:

For the first two there is no much differnce between the mark distribution but for the communications subject there is a difference in the marks between boys and girls.

Except for the marks of boys for science, all other we can spot a outlier

For science marks, the median of girls lie to third quartile thus it has slight negative Skew and for boys it is a slightly postive skewed.

For Language marks, the median of girls lie to first quartile thus it has slight positive Skew and for boys it is a slightly negatively skewed.

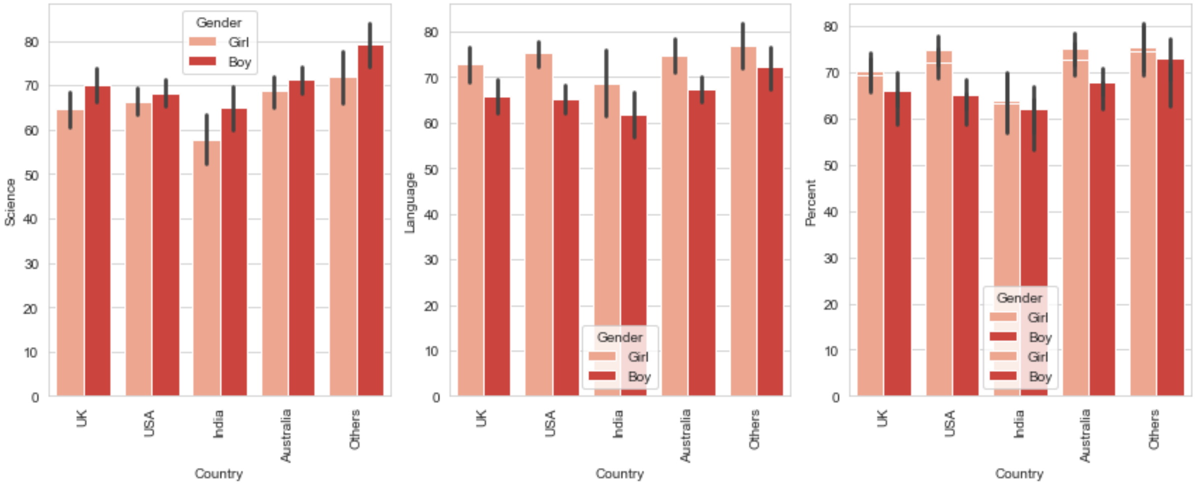
For Communication marks, the median of girls lie above first quartile and same for boys also.

Country VS subject marks for boys and Girls

In [38]:

```
plt.figure(figsize=(12,5))
plt.subplot(1,3,1)
sns.barplot(x = 'Country', y = 'Science',hue='Gender',palette = "Reds", data = df)
plt.xticks(rotation = 90)

plt.subplot(1,3,2)
sns.barplot(x = 'Country', y = 'Language',hue='Gender',palette = "Reds", data = df)
plt.xticks(rotation = 90)

plt.subplot(1,3,3)
sns.barplot(x = 'Country', y = 'Communication',hue='Gender',palette = "Reds", data = df)
plt.xticks(rotation = 90)

plt.subplot(1,3,3)
sns.barplot(x = 'Country', y = 'Percent',hue='Gender',palette = "Reds", data = df)
plt.xticks(rotation = 90)



plt.tight_layout()
```

```
<ipython-input-38-fcc4fa295518>:14: MatplotlibDeprecationWarning: Adding an axes using the same
arguments as a previous axes currently reuses the earlier instance.  In a future version, a new
instance will always be created and returned.  Meanwhile, this warning can be suppressed, and t
he future behavior ensured, by passing a unique label to each axes instance.
  plt.subplot(1,3,3)
```

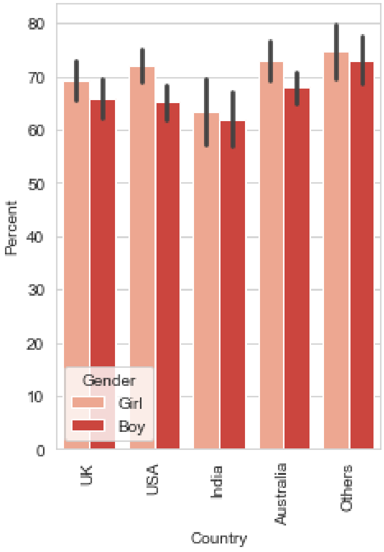


In [40]:

```
plt.figure(figsize=(12,5))
plt.subplot(1,3,3)
sns.barplot(x = 'Country', y = 'Percent',hue='Gender',palette = "Reds", data = df)
plt.xticks(rotation = 90)
```

Out[40]:

```
(array([0, 1, 2, 3, 4]), <a list of 5 Text major ticklabel objects>)
```



Observation:

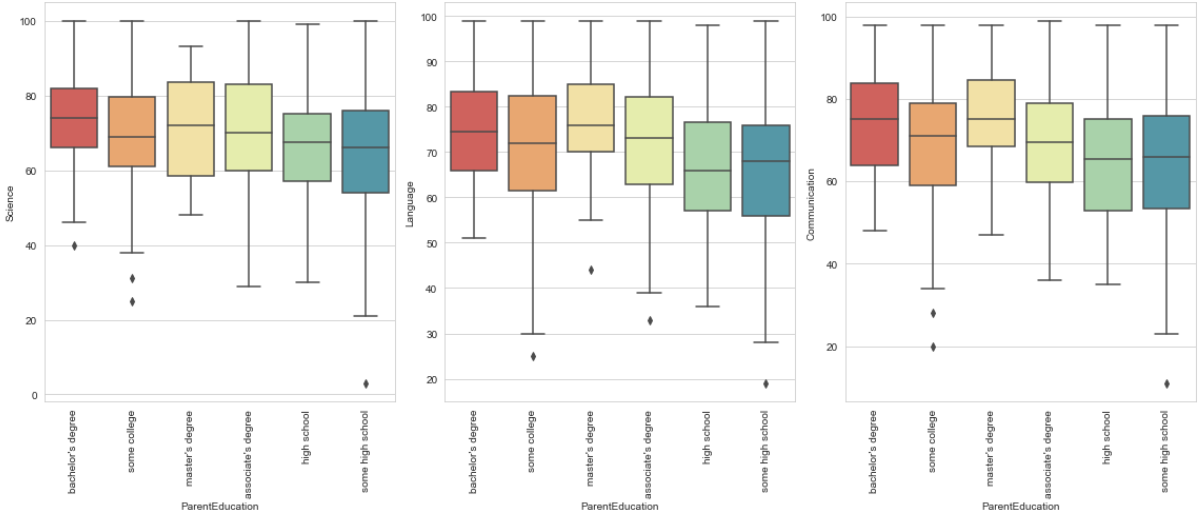
For science_score, it is students from other countries who scored more marks and for other two subject girls from every countries are having good performance and for boys it is from other countries.

Overall, Girls and boys from other countries have performed well and students from India are having a low performance.

Barplot of ParentEducation VS Subject Plot

In [234]:

```
sns.set_style('whitegrid')
plt.figure(figsize=(16,7))
plt.subplot(1,3,1)
sns.boxplot(x ='ParentEducation' , y = 'Science', palette='Spectral',data = df)
plt.xticks(rotation = 90)

plt.subplot(1,3,2)
sns.boxplot(x ='ParentEducation' , y='Language',palette = 'Spectral', data = df)
plt.xticks(rotation = 90)

plt.subplot(1,3,3)
sns.boxplot(x ='ParentEducation' ,y='Communication', palette = 'Spectral', data = df)
plt.xticks(rotation = 90)

plt.tight_layout()
```



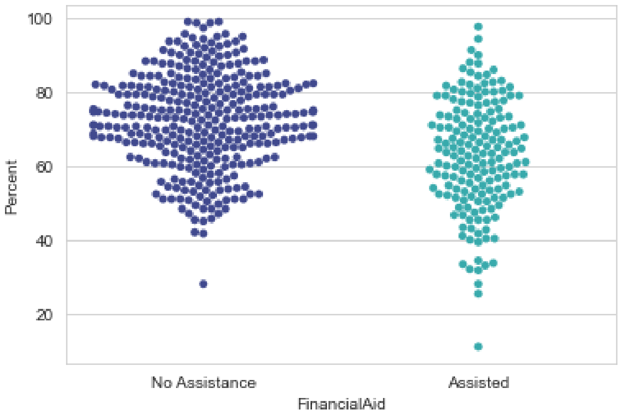
Observation

Student's whose parents have a Master's degree have scored higher compared to others whereas Student's whose parent's went to high school have obtained low marks compared to others.

Swarmplot of FinancialAid Vs Percent

```
In [114]: sns.swarmplot(x="FinancialAid", y="Percent",palette='mako',data=df)
```

```
Out[114]: <matplotlib.axes._subplots.AxesSubplot at 0x24bcb35b0a0>
```

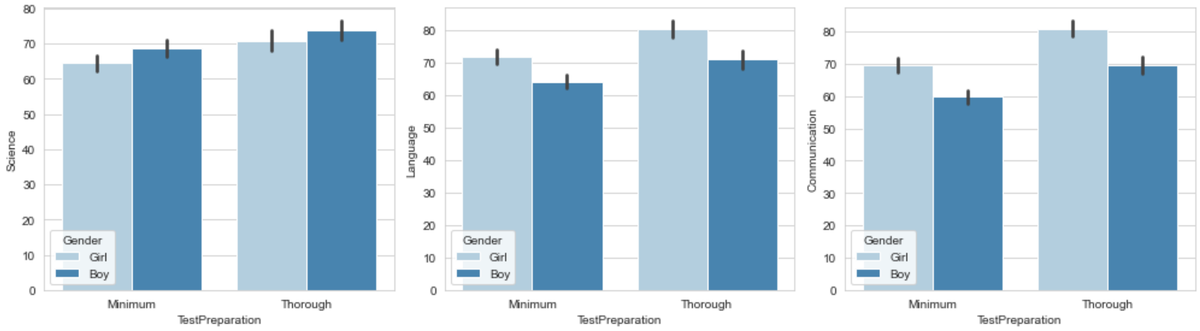


Observation

Students who have no financial assitance have performed well than the people who have assisted them thus, a serious action should be taken to students who have understaken some form of finanical assiatance else it may lead them to some trouble in future

Barplot of TestPrep Vs Subject for Boys and Girls

```
In [239]: plt.figure(figsize=(14,4))

          plt.subplot(1,3,1)
          sns.barplot(x = 'TestPreparation', y = 'Science', hue = 'Gender',palette='Blues', data = df)

          plt.subplot(1,3,2)
          sns.barplot(x = 'TestPreparation', y = 'Language',hue = 'Gender',palette='Blues', data = df)

          plt.subplot(1,3,3)
          sns.barplot(x = 'TestPreparation', y = 'Communication',hue = 'Gender',palette='Blues', data = d
          f)

          plt.tight_layout()
```



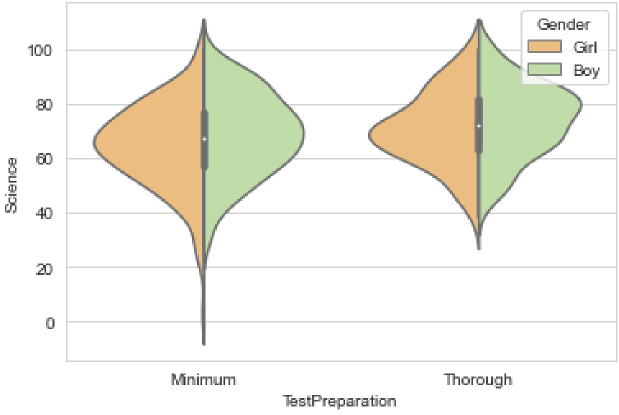
Observation

Students who have completed the Test Prepration Course have scores higher in all three categories than those who haven't taken the course

Violin Plot for Test Preparation VS Marks of each subject

In [110]:
```
sns.violinplot(x='TestPreparation', y='Science',hue='Gender',palette = "Spectral", data=df, split=True)
```

Out[110]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x24bd0936b80>
```



Observation

Mnimum preparation

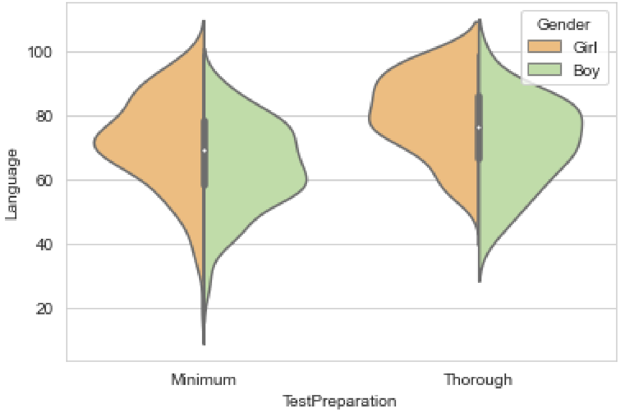
For girls mean value is between 60 and 70 because we can see a high density between 60 and 70 and for boys it is between 60 to 80

Thorough preparation

For girls mean value is at 70 because we can see a high density at 70 and for boys it is at 80.

In [111]:
```
sns.violinplot(x='TestPreparation', y='Language',hue='Gender',palette = "Spectral", data=df, split=True)
```

Out[111]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x24bd0ac80a0>
```



Observation

Mnimum preparation

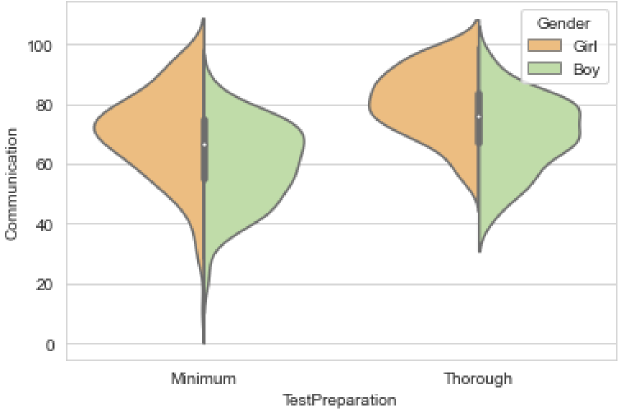
For girls mean value is between 70 and 80 because we can see a high density between 70 and 80 and for boys it is from 60 to 70.

Thorough preparation

For girls mean value is between and 80 because we can see a high density and for boys it is between 70 and 80

In [112]:
```
sns.violinplot(x='TestPreparation', y='Communication',hue='Gender',palette = "Spectral", data=d
f, split=True)
```

Out[112]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x24bd0870f10>
```



Observation

Mnimum preparation

For girls mean value is between 60 and 80 because we can see a high density between 60 and 80 and for boys also it is between 60 to 70

Thorough preparation

For girls mean value is at 80 because we can see a high density at 80 and for boys it is between 70 and 80

Conclusion

In this lab we have tried to find some Insightful Inference using matplotlib and seaborn from ExamResult dataset. Based on the observation; we can understands

*Girls have outshined than Boys.

*Student's whose parents have a Master's degree have scored higher compared to others whereas Student's whose parent's went to high school have obtained low marks compared to others.

*Students who have no financial assitance have performed well than the people who have assisted them thus, a serious action should be taken to students who have understaken some form of finanical assiatance else it may lead them to some trouble in future.

*Students who have completed the Test Prepration Course have scores higher in all three categories than those who haven't taken the course

*Overall, Girls and boys from other countries have performed well and students from India are having a low performance.

*Science teacher need to have a serious look into the no of students in first class more than second class to keep up with a moderate performance of class.

*Overall, around 90% percent of the students have won first class which shows class is having a moderate performance but there is a lack of outstanding performance so teachers should focus into that also.

*Majority(99.8%) of students passed in all the three subjects.Only 0.2% students failed in atleast one of the three subjects.

*Teacher should care academics of boys more than girls because sometime they fail keep up with average performance.

Thus, we understand the best way to tackle a problem is exploratory data analysis (EDA). EDA is essential for a structured data science project and should be performed before any statistical and machine learning model phase.

From the data perspective it helps us to identify the patterns, outliers, and how to proceed with the problem in hand.

In [ ]: