In [3]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [4]:
```
df = pd.read_csv("C:\\Users\\Suprit Kaur\\OneDrive\\Desktop\\Python\\Jupyter\\pr
df.head()
```

Out[4]:

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

In [5]:
```
df.describe() #it returns the columns containing numerical values
```

Out[5]:

	NrSiblings	MathScore	ReadingScore	WritingScore
count	29069.000000	30641.000000	30641.000000	30641.000000
mean	2.145894	66.558402	69.377533	68.418622
std	1.458242	15.361616	14.758952	15.443525
min	0.000000	0.000000	10.000000	4.000000
25%	1.000000	56.000000	59.000000	58.000000
50%	2.000000	67.000000	70.000000	69.000000
75%	3.000000	78.000000	80.000000	79.000000
max	7.000000	100.000000	100.000000	100.000000

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30641 entries, 0 to 30640
Data columns (total 14 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   Gender               30641 non-null  object 
 1   EthnicGroup          28801 non-null  object 
 2   ParentEduc           28796 non-null  object 
 3   LunchType            30641 non-null  object 
 4   TestPrep             28811 non-null  object 
 5   ParentMaritalStatus  29451 non-null  object 
 6   PracticeSport        30010 non-null  object 
 7   IsFirstChild         29737 non-null  object 
 8   NrSiblings           29069 non-null  float64
 9   TransportMeans       27507 non-null  object 
 10  WklyStudyHours       29686 non-null  object 
 11  MathScore            30641 non-null  int64  
 12  ReadingScore         30641 non-null  int64  
 13  WritingScore         30641 non-null  int64  
dtypes: float64(1), int64(3), object(10)
memory usage: 3.3+ MB
```

In [7]:
```
df.isnull().sum()
```

Out[7]:
```
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
```

In [8]:
```
df["WklyStudyHours"] = df["WklyStudyHours"].str.replace('45935','5-10')
```

In [9]:
```
df["WklyStudyHours"].unique()
```

Out[9]:
```
array(['< 5', '5-10', '> 10', nan], dtype=object)
```

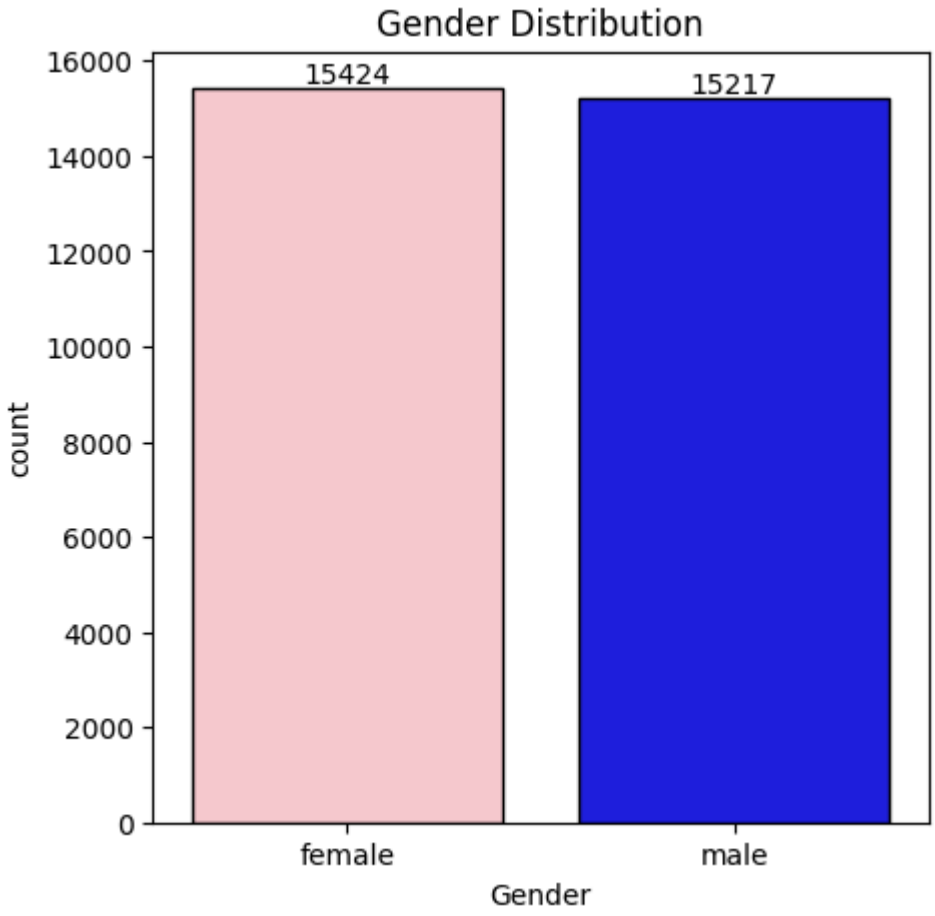
In [10]:
```
df.head()
```

Out[10]:

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

gender distribution

In [71]:
```
plt.figure(figsize = (5,5))
ax = sns.countplot(data = df , x = 'Gender' ,hue = 'Gender', palette = ["pink","
for label in ax.containers: #for adding labels to all the containers
    ax.bar_label(label)
plt.title("Gender Distribution")
plt.show()
```



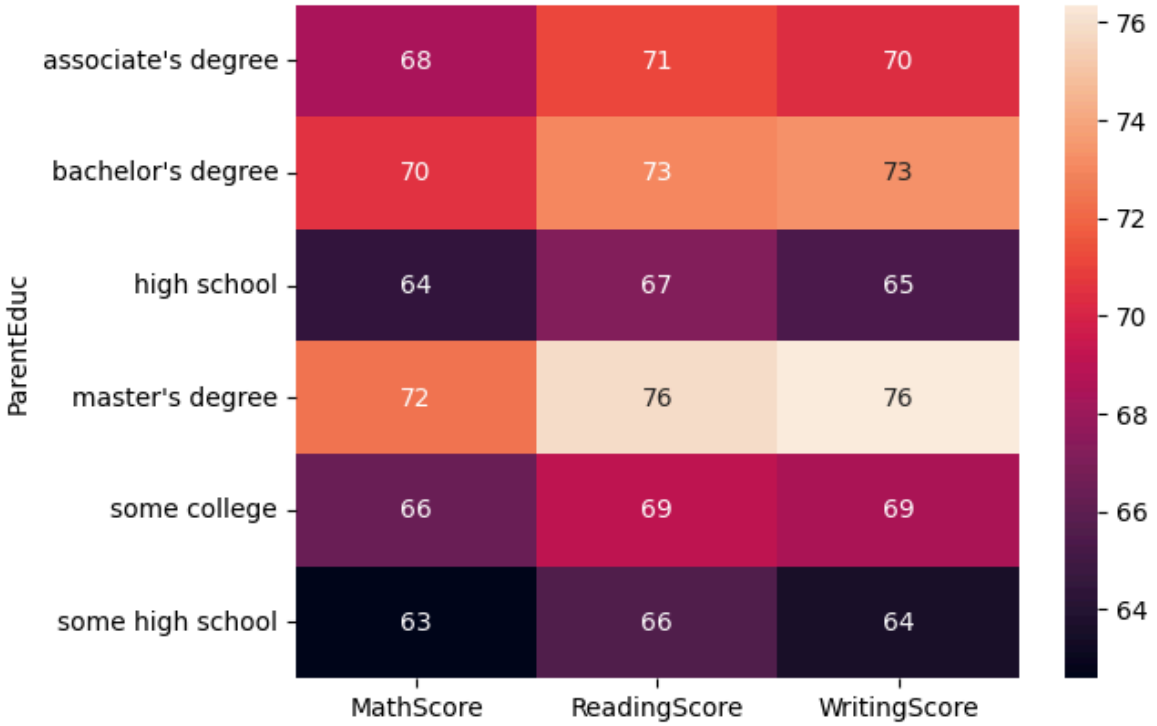
In [12]:
```
#from the above chart we have analyzed that:
# the number of females are more as compared to males in the data
```

In [13]:
```
df['ParentEduc'].unique()
```

Out[13]:
```
array(["bachelor's degree", 'some college', "master's degree",
       "associate's degree", 'high school', 'some high school', nan],
      dtype=object)
```

In [14]:
```
group= df.groupby('ParentEduc').agg({'MathScore':'mean','ReadingScore':'mean','W
print(group)
sns.heatmap(group , annot = True)
plt.show()
```

ParentEduc	MathScore	ReadingScore	WritingScore
associate's degree	68.365586	71.124324	70.299099
bachelor's degree	70.466627	73.062020	73.331069
high school	64.435731	67.213997	65.421136
master's degree	72.336134	75.832921	76.356896
some college	66.390472	69.179708	68.501432
some high school	62.584013	65.510785	63.632409



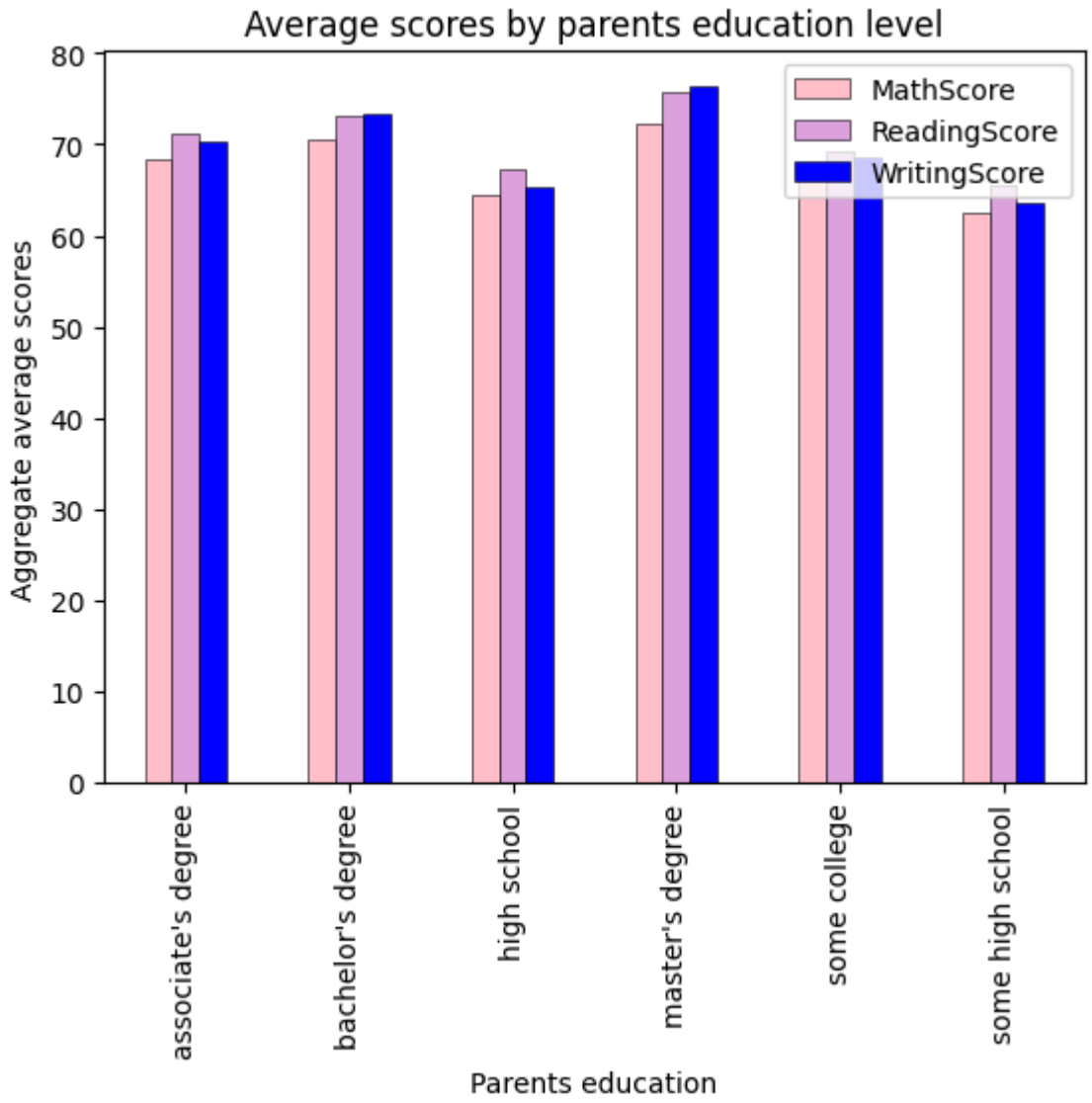
In [15]:
```
group1= df.groupby('ParentEduc').agg({'MathScore':'mean','ReadingScore':'mean','
print(group1)
```

ParentEduc	MathScore	ReadingScore	WritingScore
associate's degree	68.365586	71.124324	70.299099
bachelor's degree	70.466627	73.062020	73.331069
high school	64.435731	67.213997	65.421136
master's degree	72.336134	75.832921	76.356896
some college	66.390472	69.179708	68.501432
some high school	62.584013	65.510785	63.632409

In [16]:
```
plt.figure(figsize=(5,5))
group.plot(kind='bar' , color = ['pink' , 'plum' , 'blue'] ,linewidth = 0.4, edg
plt.title("Average scores by parents education level")
```

```
plt.xlabel("Parents education")
plt.ylabel("Aggregate average scores")
plt.show()
```

```
<Figure size 500x500 with 0 Axes>
```



In [17]:
```
#It is assumed that It is assumed that "some high school" and "some college" gro
#completed some coursework but did not finish the respective educational program
#for the above chart we have analyzed that :
#Students with parents having higher education have higher aggregate scores i.e.
#parents have good impact on students
```

In [18]:
```
df["ParentMaritalStatus"].unique()
```

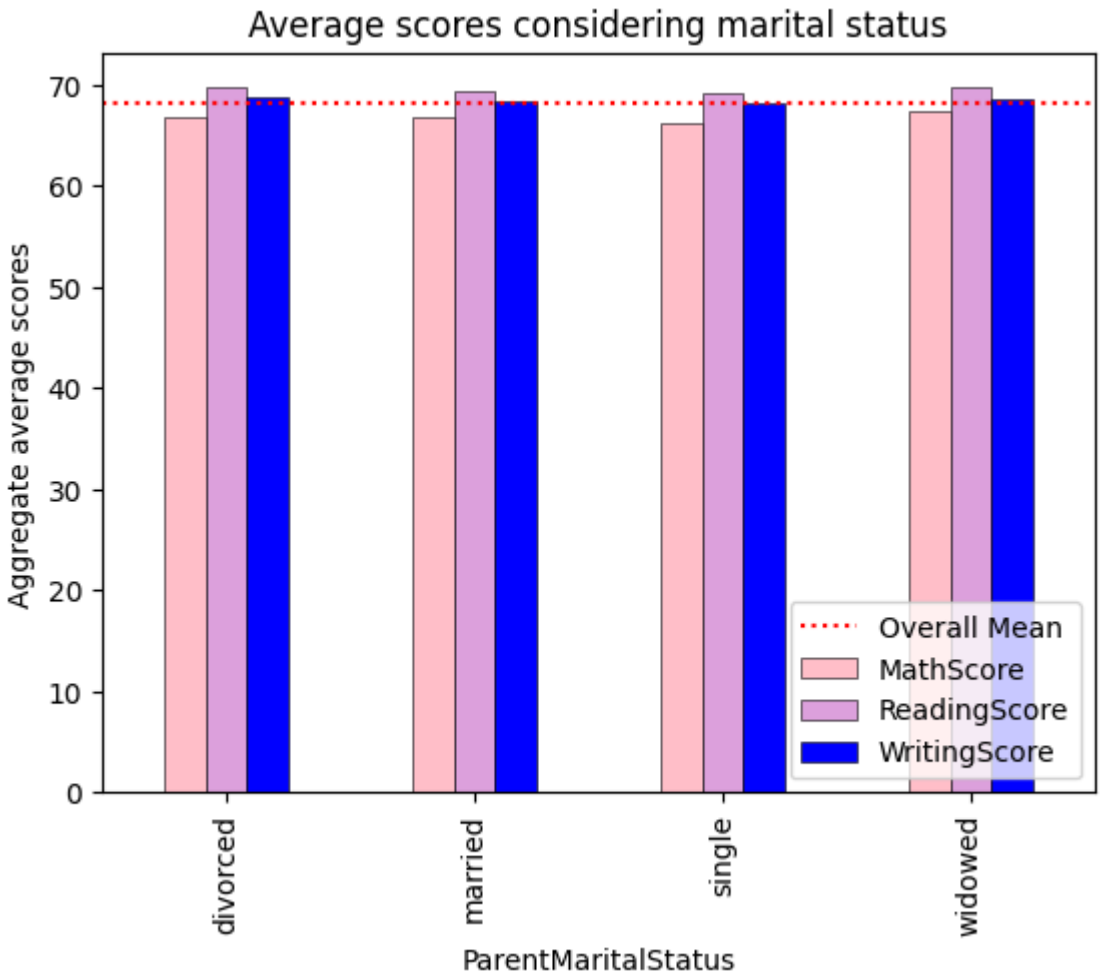
Out[18]:
```
array(['married', 'single', 'widowed', nan, 'divorced'], dtype=object)
```

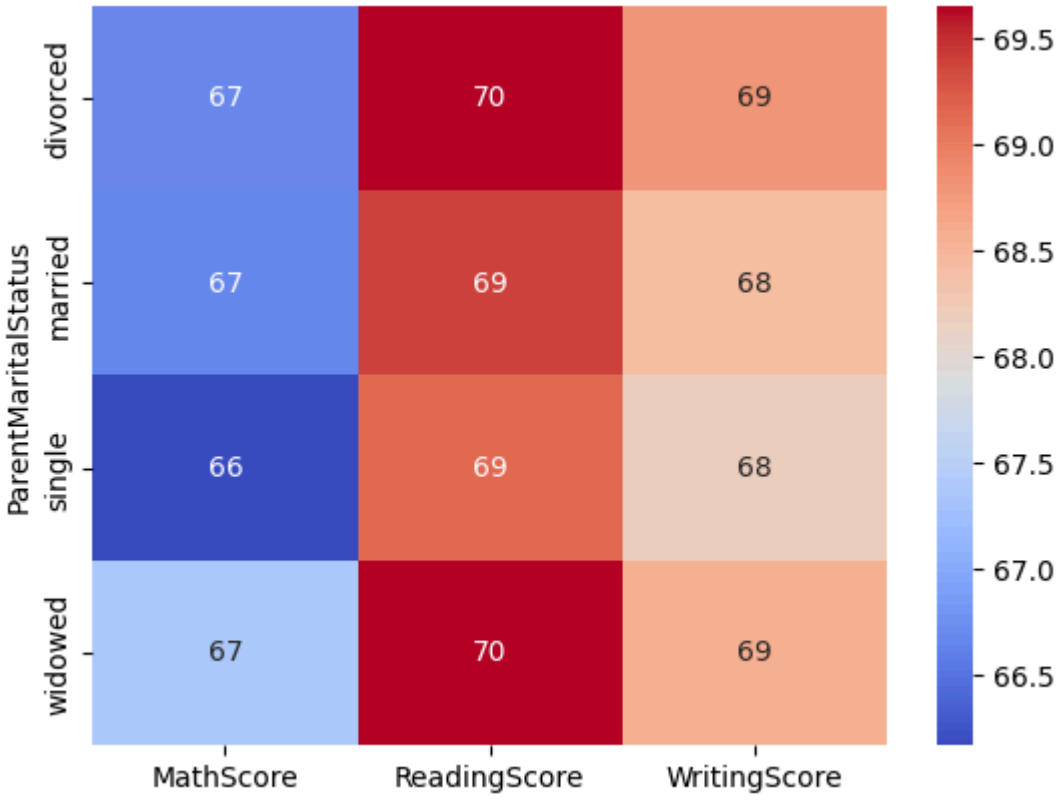
In [19]:
```
group3= df.groupby(["ParentMaritalStatus"]).agg({'MathScore':'mean','ReadingScor
print(group3)
```

```
                     MathScore  ReadingScore  WritingScore
ParentMaritalStatus
divorced             66.691197     69.655011     68.799146
married              66.657326     69.389575     68.420981
single               66.165704     69.157250     68.174440
widowed              67.368866     69.651438     68.563452
```

In [20]:
```
plt.figure(figsize=(5,5))
group3.plot(kind='bar' , color = ['pink' , 'plum' , 'blue'] ,linewidth = 0.4, ed
plt.title("Average scores considering marital status")
plt.ylabel("Aggregate average scores")
plt.axhline(group3.mean().mean() , color = 'red' , linestyle= ':' , label='Overa
plt.legend(loc="lower right")
plt.show()
sns.heatmap(group3 , annot = True ,cmap = 'coolwarm')
plt.show()
```

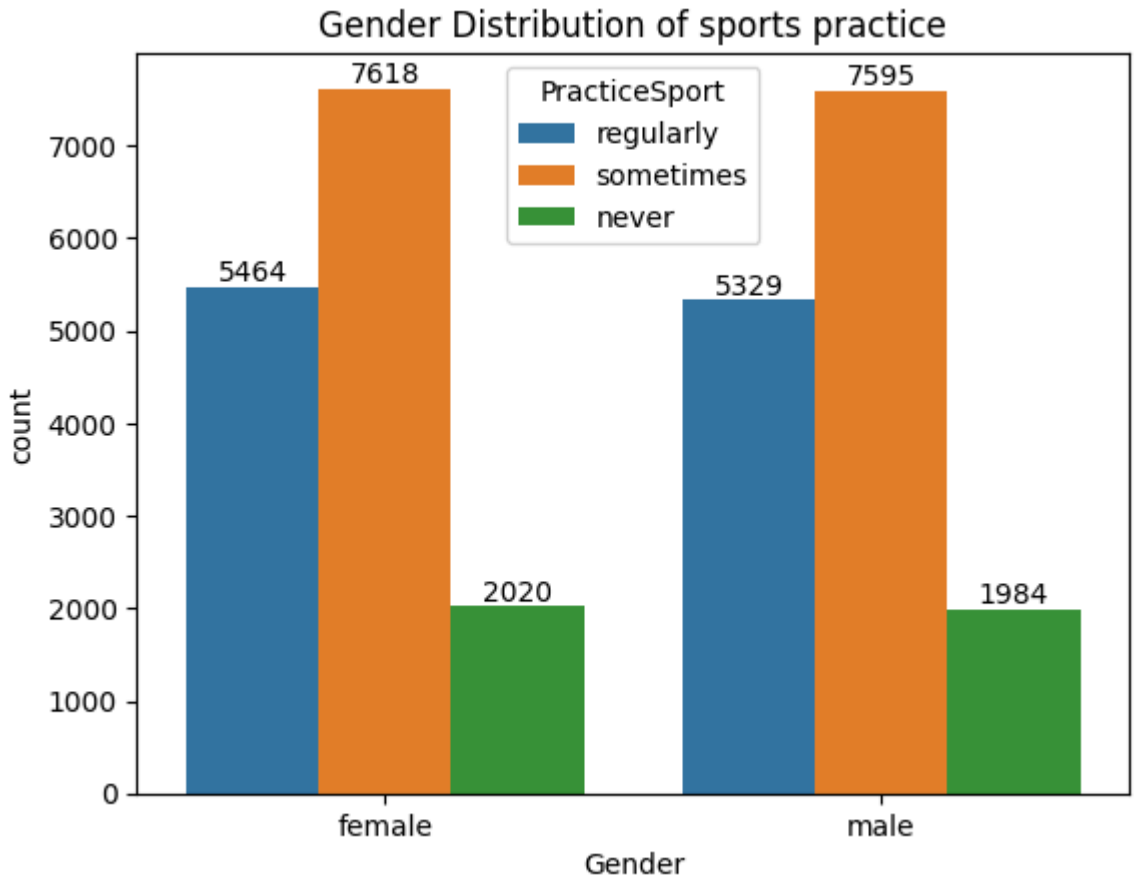
```
<Figure size 500x500 with 0 Axes>
```





In [23]:
```
#From the above chart we have concluded that the marital status of the parents h
# impact on their child's score
```

In [70]:
```
plt.title("Gender Distribution of sports practice")
chart =sns.countplot(data=df ,x='Gender', hue="PracticeSport"  )
for label in chart.containers: #for adding labels to all the containers
    chart.bar_label(label)
plt.show()
```



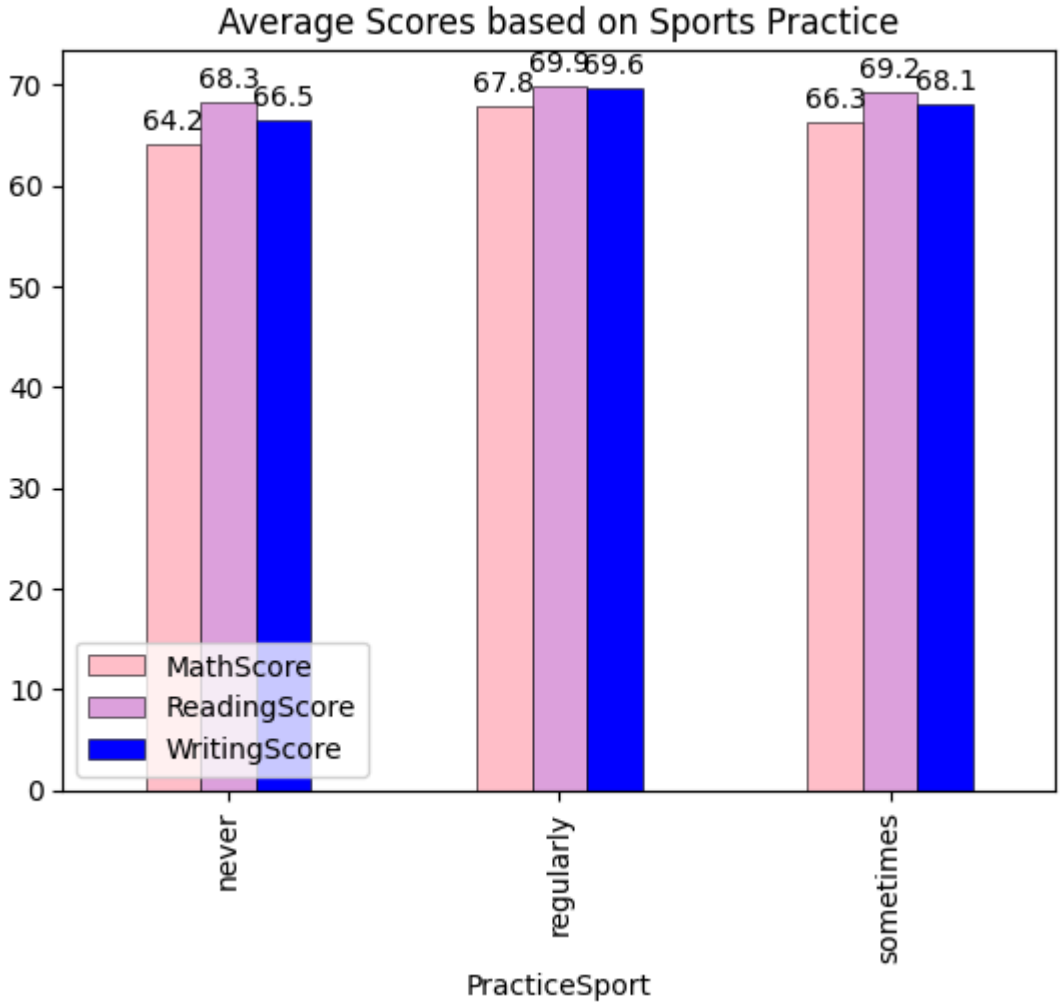
In [35]:
```
#From the above graph we have analyzed that :
# The numbers for males and females across all three categories(regularly,someti
#suggesting that sports participation is almost equal for both genders.
#However, females have slightly higher participation numbers across all categori
```

In [45]:
```
group4 = df.groupby(['PracticeSport']).agg({'MathScore':'mean',
                                            'ReadingScore':'mean',
                                            'WritingScore':'mean'}
                                            )
print(group4)
```

```
              MathScore  ReadingScore  WritingScore
PracticeSport
never         64.171079     68.337662     66.522727
regularly     67.839155     69.943019     69.604003
sometimes     66.274831     69.241307     68.072438
```

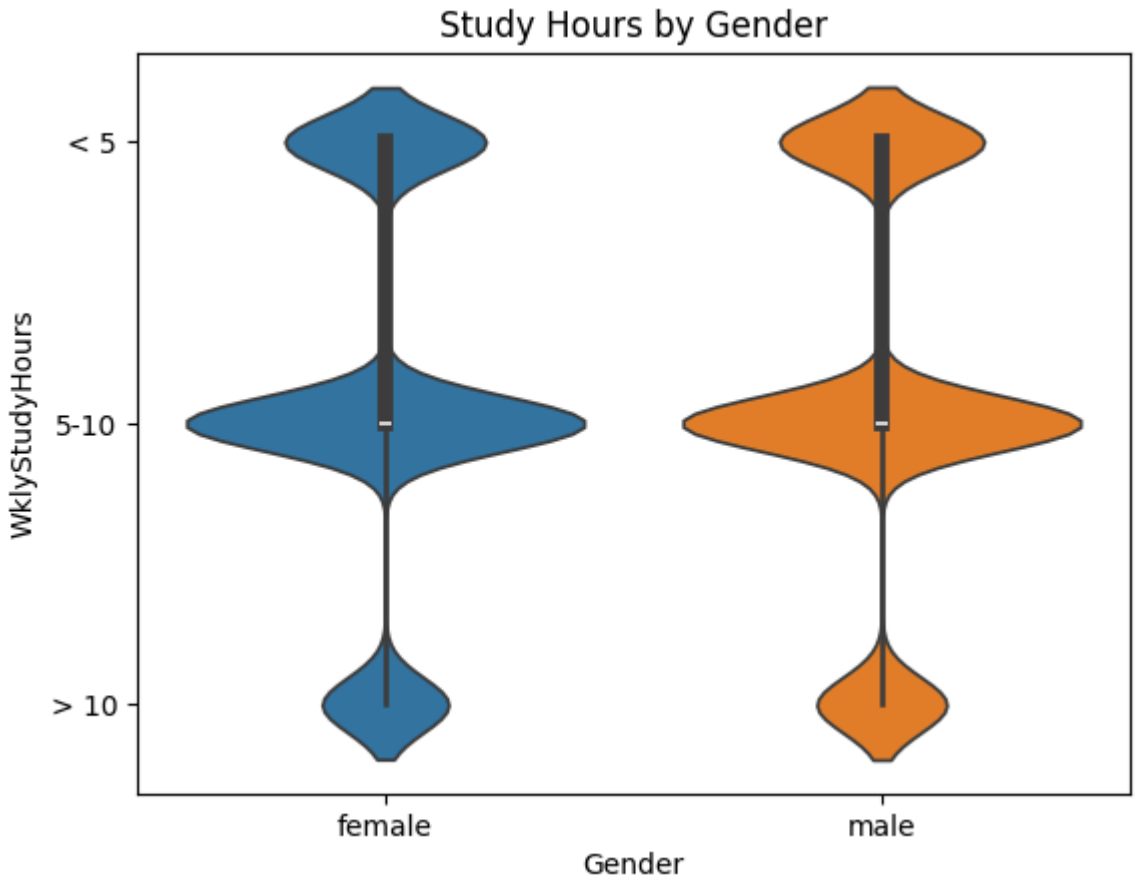
In [67]:
```
plt.figure(figsize=(5,5))
chart =group4.plot(kind='bar' , color = ['pink' , 'plum' , 'blue'] ,linewidth =
plt.title("Average Scores based on Sports Practice")
for label in chart.containers: #for adding labels to all the containers
    chart.bar_label(label,fmt='%.1f', padding=3)
plt.show()
```

```
<Figure size 500x500 with 0 Axes>
```



In [54]:
```
#Though students who play sports regularly have slightly better aggregate scores
#and never
#However, there's not much impact of sports practice on student's score
```

In [60]:
```
sns.violinplot(data=df, x="Gender", y="WklyStudyHours", hue='Gender')
plt.title("Study Hours by Gender")
plt.show()
```



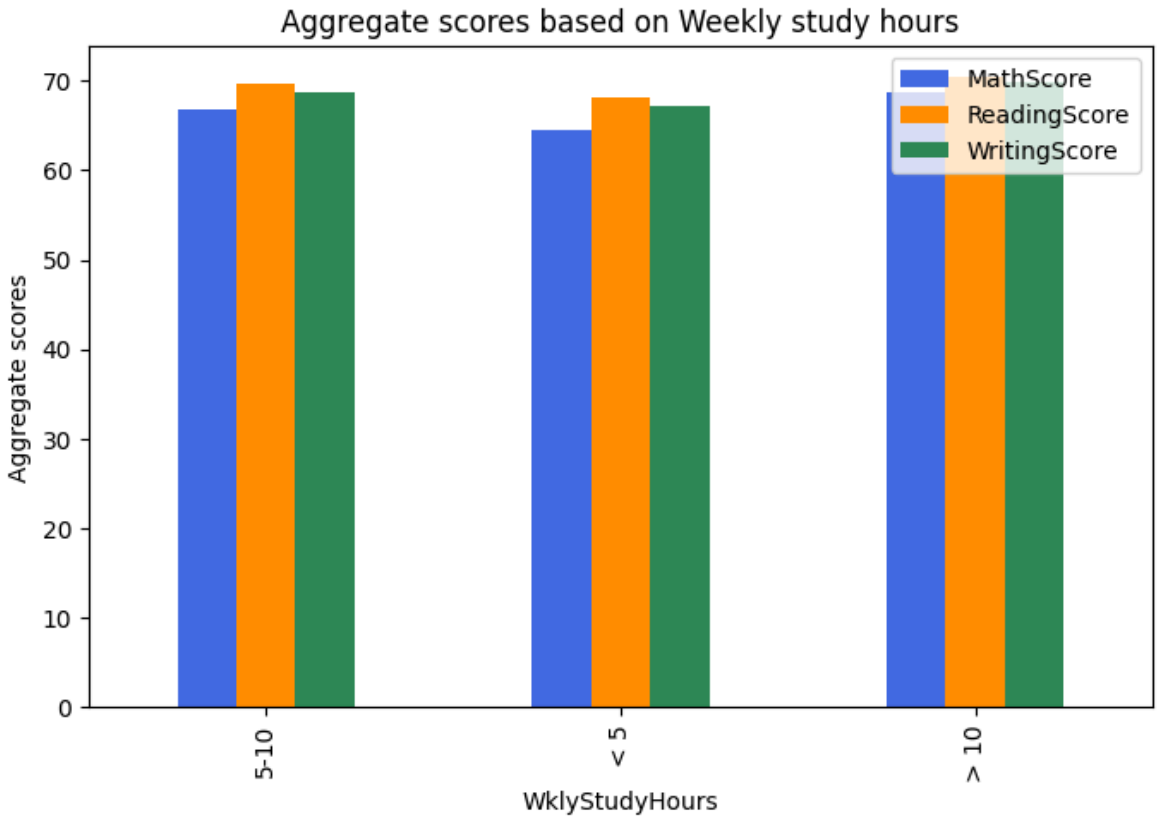
In [62]:
```
#The overall distribution of the study hours is nearly same. The widest part of
#most students study 5-10 hours weekly
```

In [64]:
```
group5 = df.groupby(['WklyStudyHours']).agg({'MathScore':'mean','ReadingScore':'
group5
```

Out[64]:

WklyStudyHours	MathScore	ReadingScore	WritingScore
5-10	66.870491	69.660532	68.636280
< 5	64.580359	68.176135	67.090192
> 10	68.696655	70.365436	69.777778

In [74]:
```
group5.plot(kind = 'bar' , figsize=(8,5), color=['#4169E1', '#FF8C00', '#2E8B57'
plt.title("Aggregate scores based on Weekly study hours")
plt.ylabel("Aggregate scores")
plt.show()
```



In [ ]:
```
#From the above chart we have analysed that :
# the students who studies more than 10 hours a week have better score than the
#or less than 5 hours.
```