Student performance

October 7, 2024

[2]:

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

[3]:

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

[5]:

```
df.isnull().mean()*100
```

[5]:

```
Hours Studied                       0.0
Previous Scores                     0.0
Extracurricular Activities          0.0
Sleep Hours                         0.0
Sample Question Papers Practiced    0.0
Performance Index                   0.0
dtype: float64
```

[6]:

```
df.columns
```

[6]:

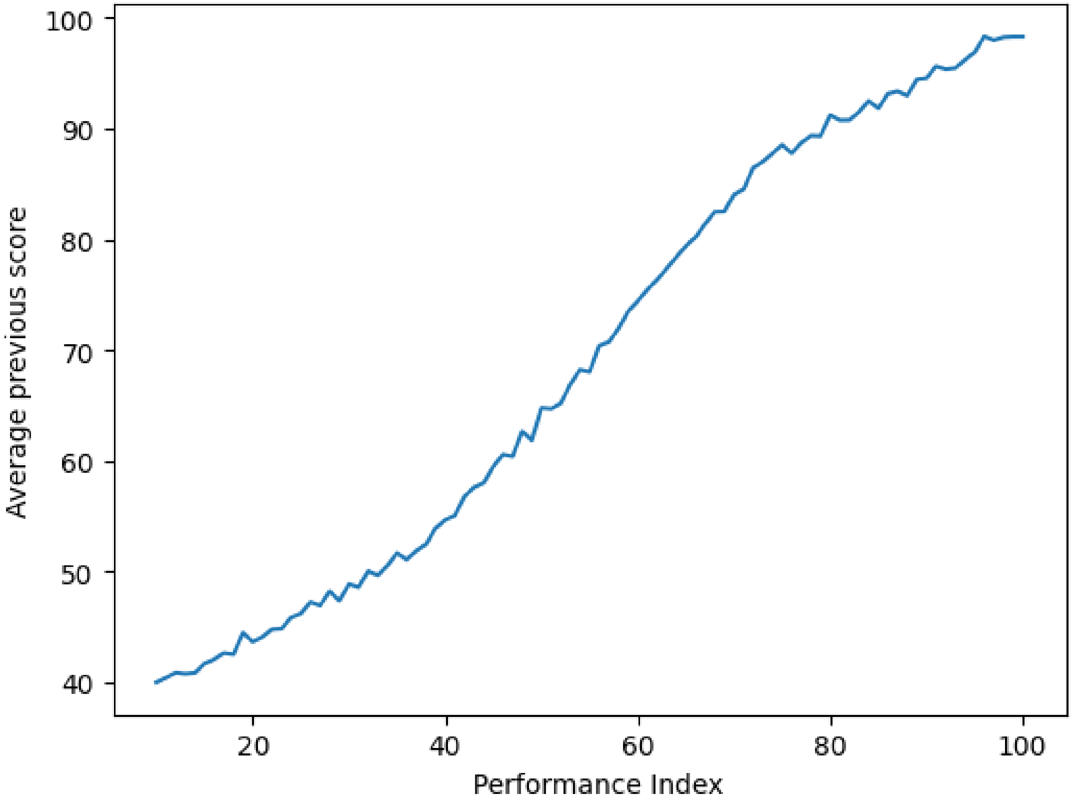
```
Index(['Hours Studied', 'Previous Scores', 'Extracurricular Activities',
       'Sleep Hours', 'Sample Question Papers Practiced', 'Performance Index'],
      dtype='object')
```

[49]:

```
previous = df.groupby("Performance Index")["Previous Scores"].mean()
plt.plot(previous.index, previous.values)
plt.ylabel("Average previous score")
plt.xlabel("  Performance Index")
#once a topper always a topper
```

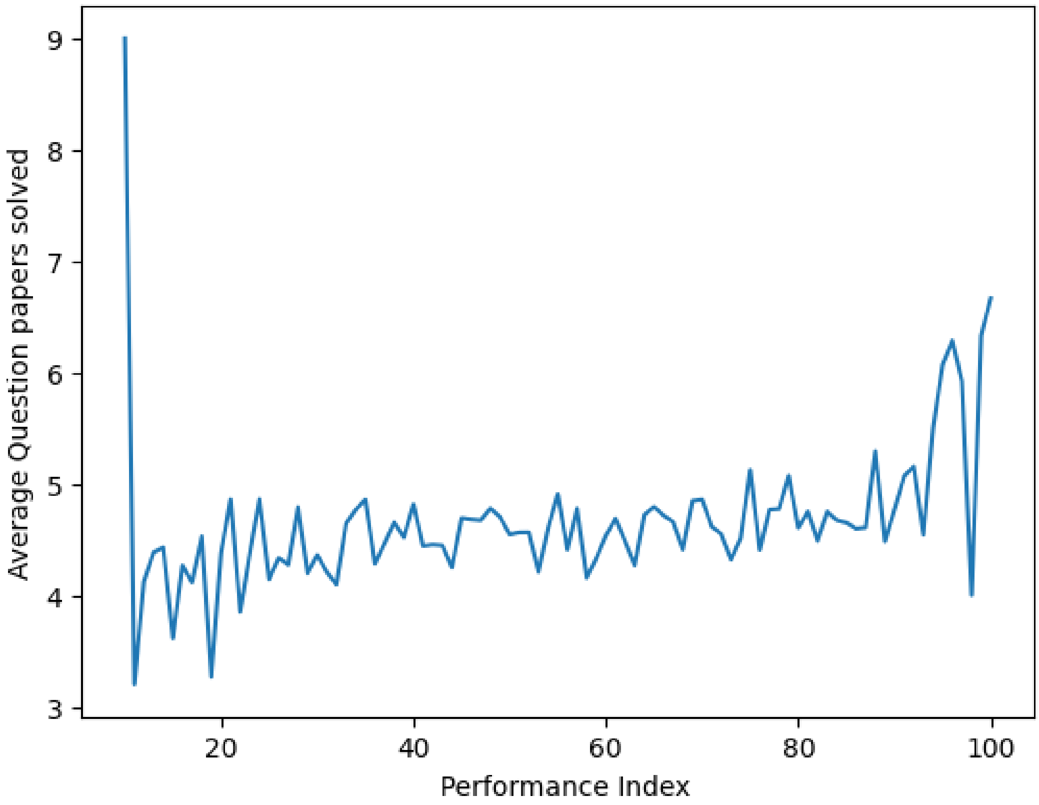
[49]:

```
Text(0.5, 0, '  Performance Index')
```



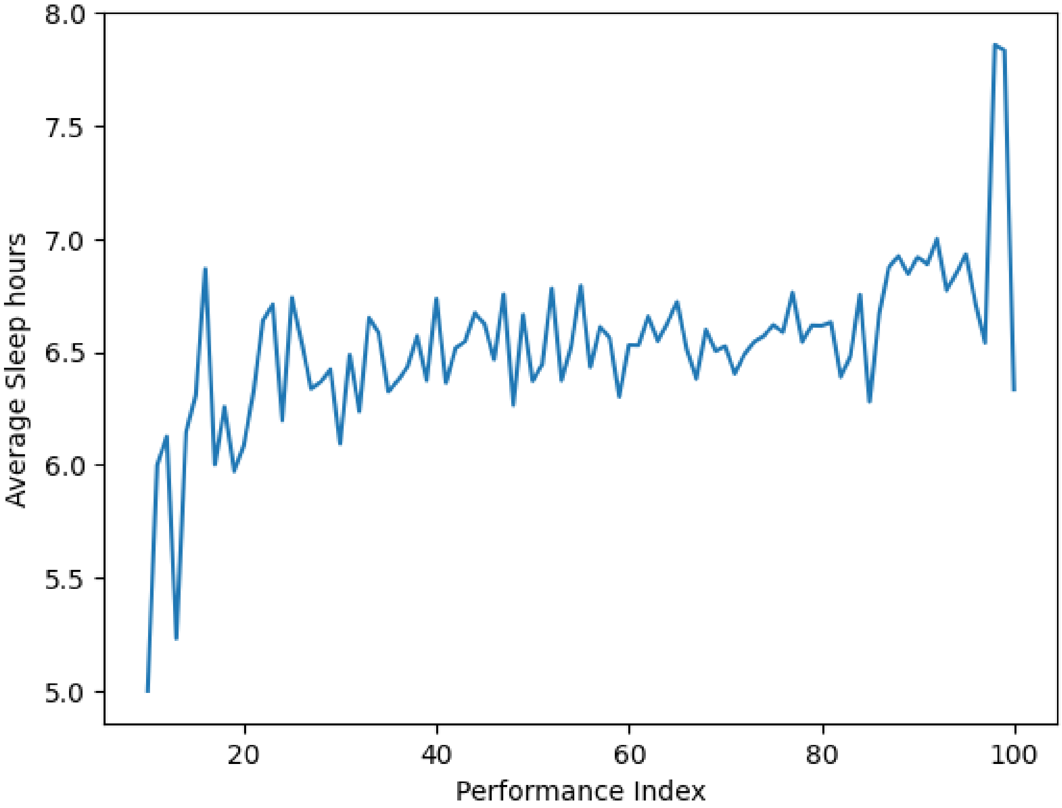
[46]:
```
questionpapersreverse = df.groupby("Performance Index")["Sample Question Papers␣
 ↪Practiced"].mean()
plt.plot(questionpapersreverse.index, questionpapersreverse.values)
plt.ylabel("Average Question papers solved")
plt.xlabel("  Performance Index")
#funny that topper haven't solved the question that much
```

[46]:
```
Text(0.5, 0, '  Performance Index')
```



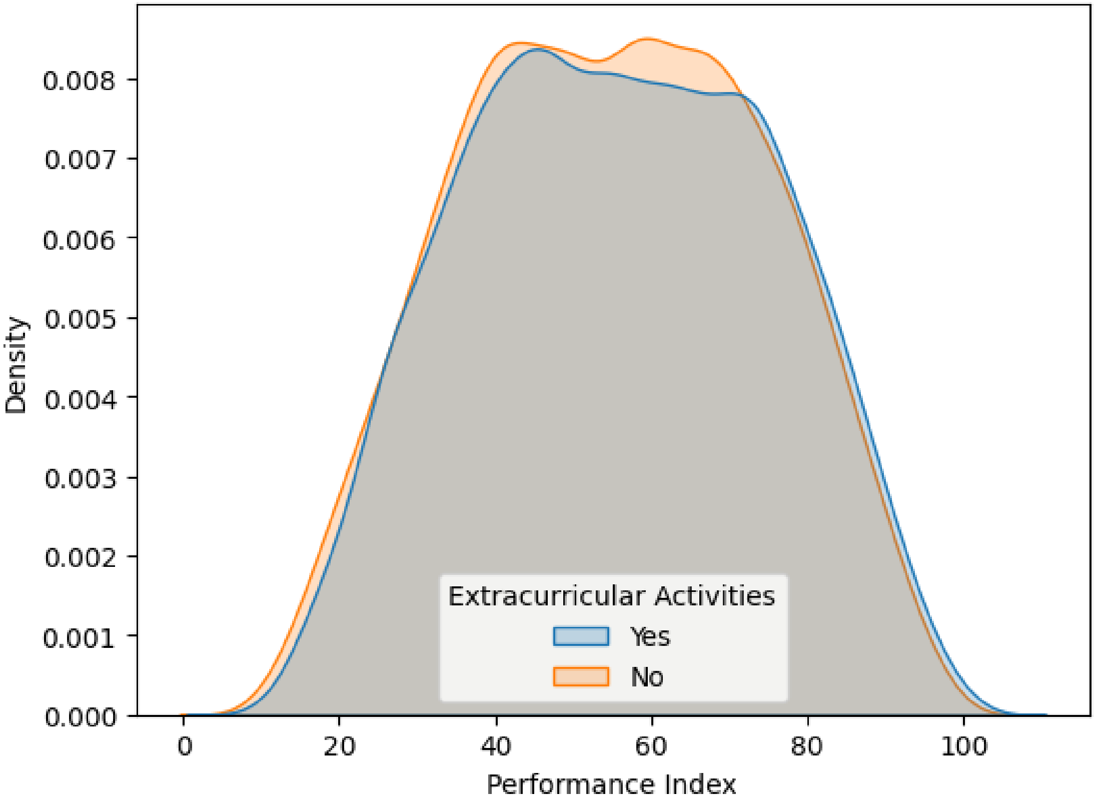
```
[52]: reverse = df.groupby("Performance Index")["Sleep Hours"].mean()
      plt.plot(reverse.index, reverse.values)
      plt.ylabel("Average Sleep hours")
      plt.xlabel("Performance Index")
      # toppers usually sleep between 7.5 to 8 hours
```

```
[52]: Text(0.5, 0, 'Performance Index')
```



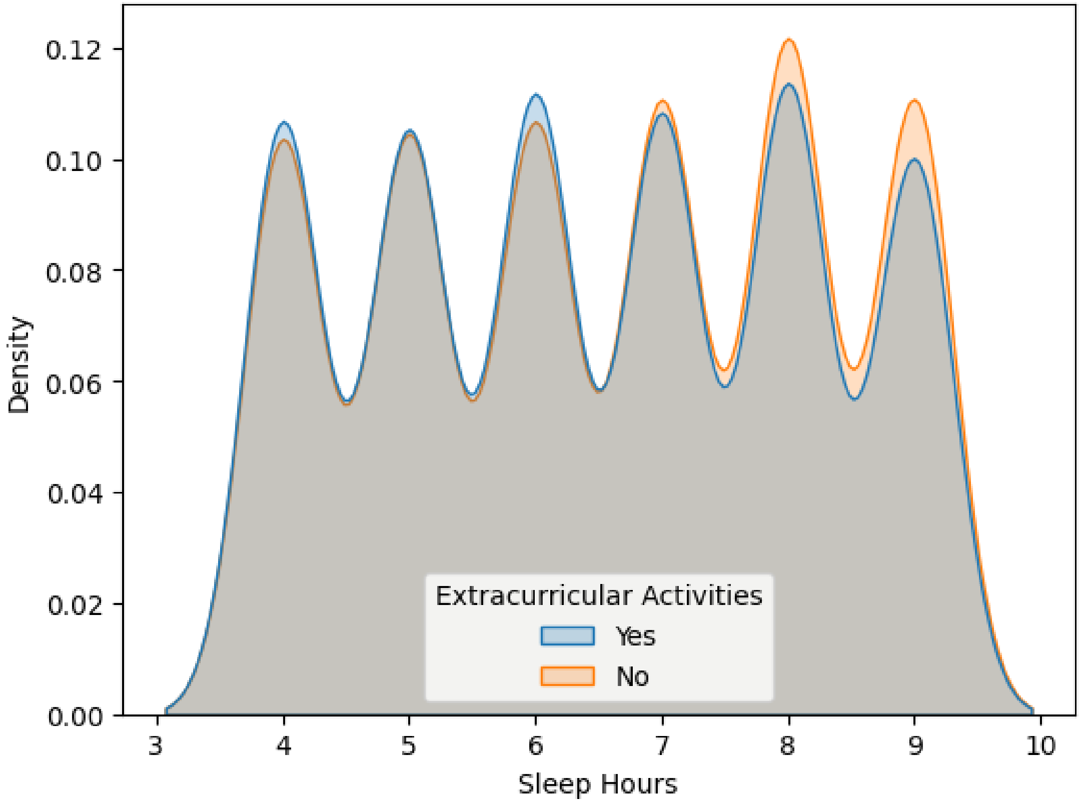
```
[11]: sns.kdeplot(data=df, x="Performance Index", hue="Extracurricular Activities",␣
       ↪fill=True)
      # max students have performance index between 30 to 70
      #and the students with extra cirricular are more than without in toppers
      #while in range of max density where perfomance is between 40 to 70 have no␣
       ↪extraciriccular activity but still score is medium
```

```
[11]: <Axes: xlabel='Performance Index', ylabel='Density'>
```



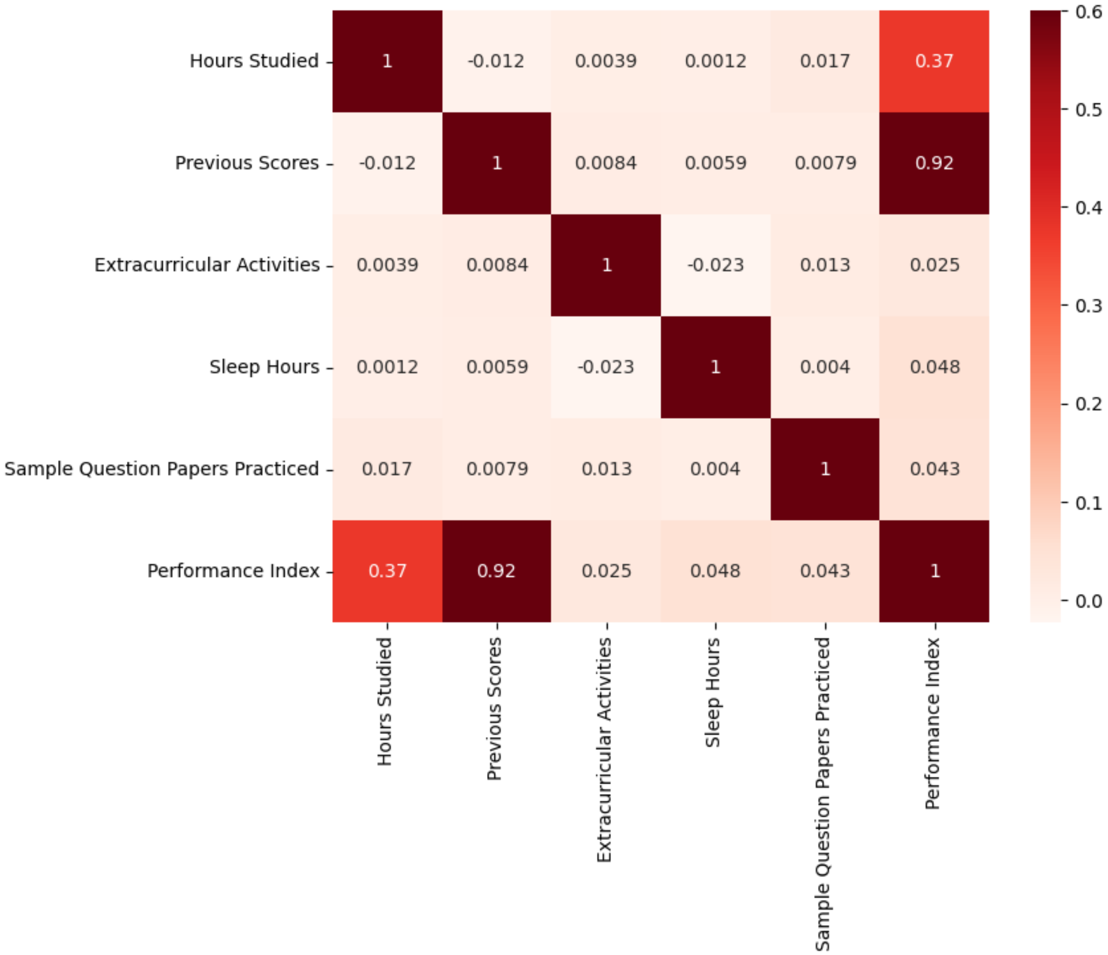
```
[14]: sns.kdeplot(data = df , x="Sleep Hours" , hue = "Extracurricular␣
      ↪Activities",fill=True)
      #no of people who sleep more are not more in extracurricular activities
```

```
[14]: <Axes: xlabel='Sleep Hours', ylabel='Density'>
```



```
[27]: df["Extracurricular Activities"] = df["Extracurricular Activities"].
       ↪replace({'Yes': 1, 'No': 0})
      plt.figure(figsize=(8,6))
      sns.heatmap(df.corr(),cmap="Reds",annot=True,annot_kws={"size":10},vmax=0.6)
      #check out all the co-relations
      # very less students become overconfident and reduce the hours studied when␣
       ↪previous score is high
      # AS above also people involved in Extracirricular activities sleep less
      # Also it is visible that the number of Hours studied is proportional to the␣
       ↪Performance Index
```

```
[27]: <Axes: >
```



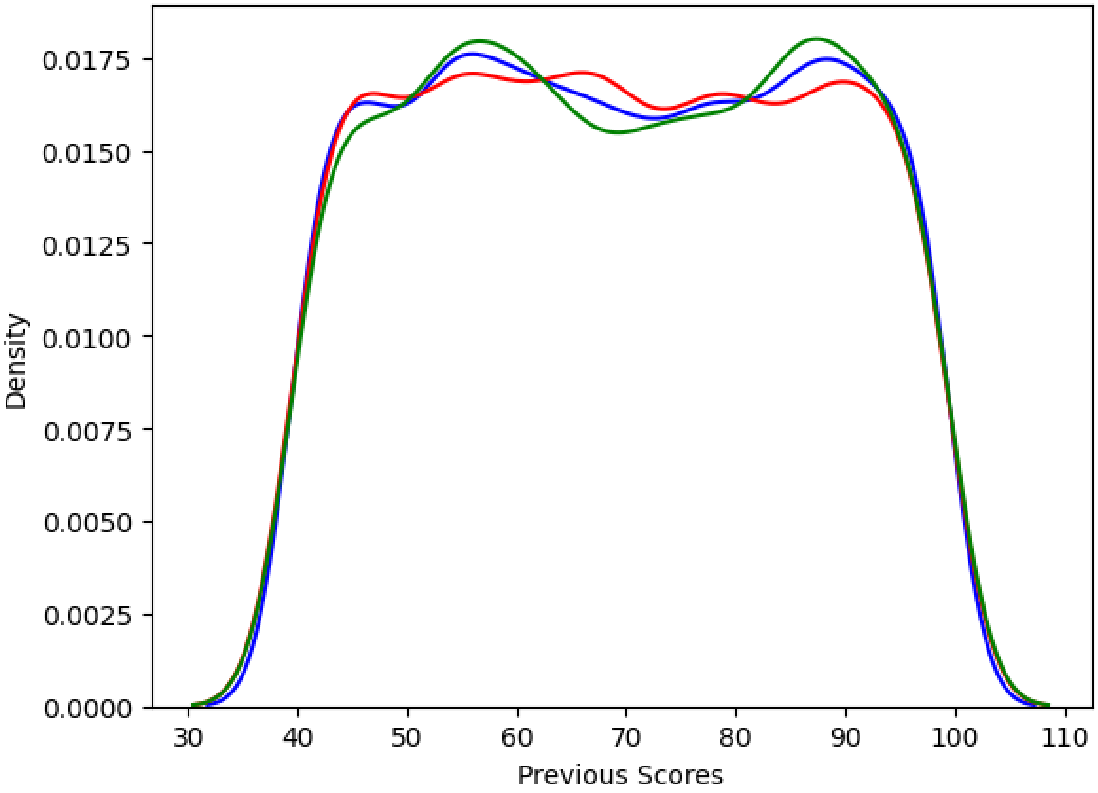
	Hours Studied	Previous Scores	Extracurricular Activities	Sleep Hours	Sample Question Papers Practiced	Performance Index
Hours Studied	1	-0.012	0.0039	0.0012	0.017	0.37
Previous Scores	-0.012	1	0.0084	0.0059	0.0079	0.92
Extracurricular Activities	0.0039	0.0084	1	-0.023	0.013	0.025
Sleep Hours	0.0012	0.0059	-0.023	1	0.004	0.048
Sample Question Papers Practiced	0.017	0.0079	0.013	0.004	1	0.043
Performance Index	0.37	0.92	0.025	0.048	0.043	1

[39]:

```
sns.kdeplot(df["Previous Scores"],color="blue")
sns.kdeplot(df[df["Extracurricular Activities"] == 0]["Previous Scores"], color="red")
sns.kdeplot(df[df["Extracurricular Activities"] == 1]["Previous Scores"], color="green")
# when the previous score in 60 to 80 then people quit the Extracirricular Activities and
#when they get above 80 score then they join Extracirricular Activities
```

[39]:

```
<Axes: xlabel='Previous Scores', ylabel='Density'>
```

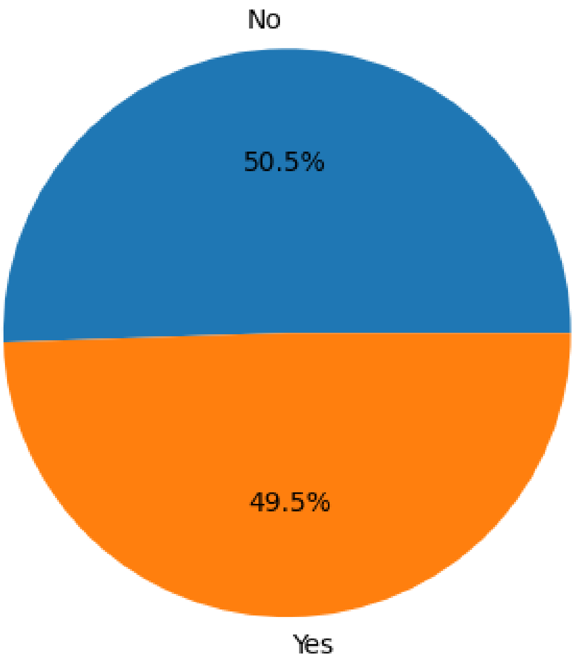


[23]:

```
extra = df["Extracurricular Activities"].value_counts()
plt.pie( extra.values,labels=extra.index, autopct='%1.1f%%' )
#data is very close
```

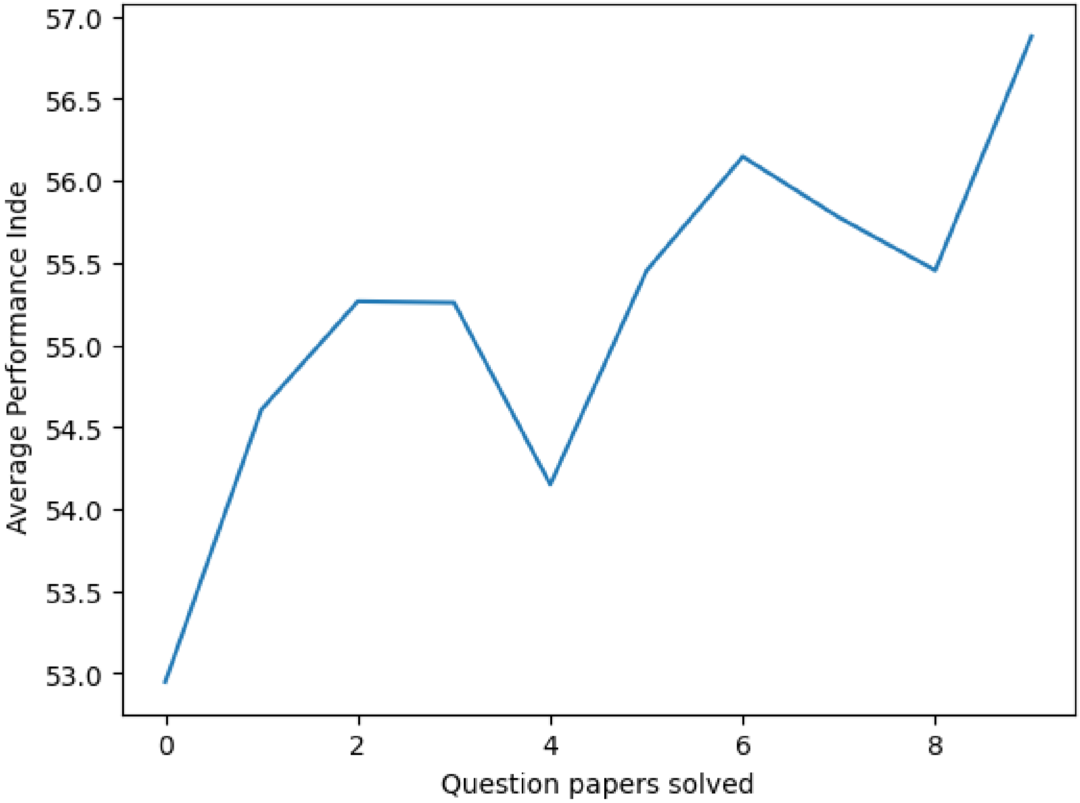
[23]:

```
([<matplotlib.patches.Wedge at 0x2568fc6b0b0>,
  <matplotlib.patches.Wedge at 0x2568fc67170>],
 [Text(-0.017969202810792655, 1.0998532210028504, 'No'),
  Text(0.01796929896301876, -1.0998532194319286, 'Yes')],
 [Text(-0.009801383351341448, 0.5999199387288274, '50.5%'),
  Text(0.009801435798010232, -0.599919937871961, '49.5%')])
```



```
[37]: questionpapers = df.groupby("Sample Question Papers Practiced")["Performance␣
      ↪Index"].mean()
      plt.plot(questionpapers.index, questionpapers.values)
      plt.xlabel("Question papers solved")
      plt.ylabel(" Average Performance Index")
      #self explaintary
```

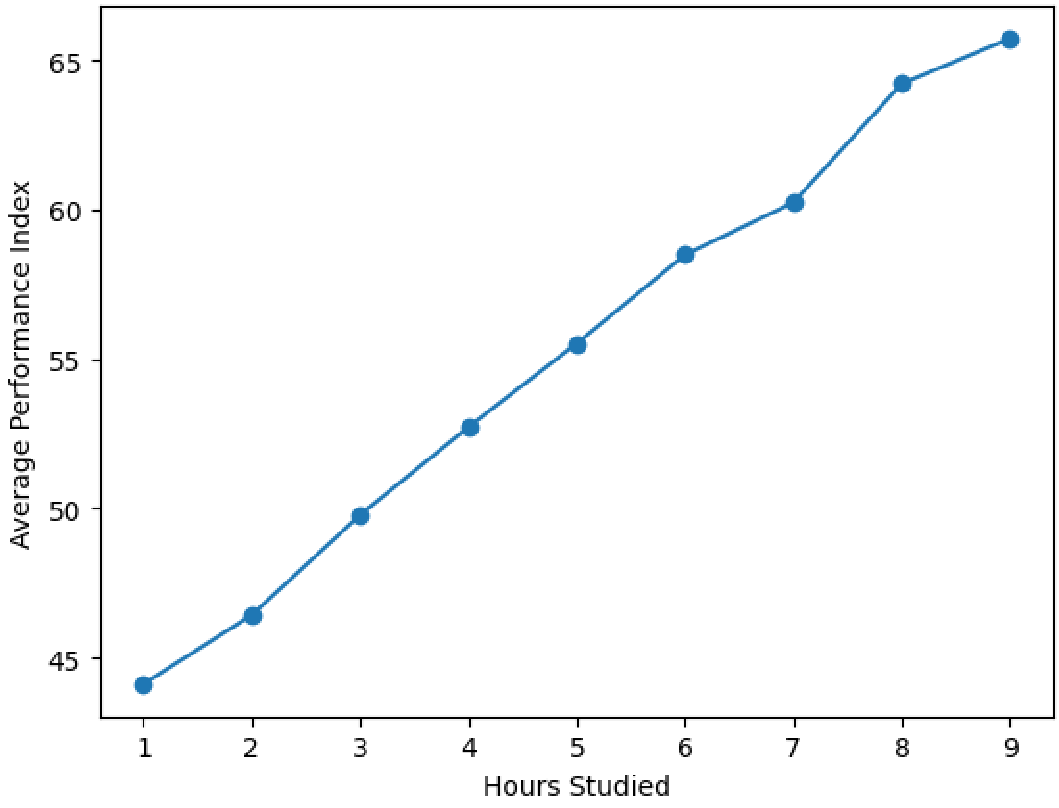
```
[37]: Text(0, 0.5, ' Average Performance Inde')
```



[11]:
```
studyhours = df.groupby("Hours Studied")["Performance Index"].mean()
```

[29]:
```
plt.plot(studyhours.index, studyhours.values, marker='o')
plt.xlabel("Hours Studied")
plt.ylabel("Average Performance Index")
ge Performance Index")
#self explaintary
```

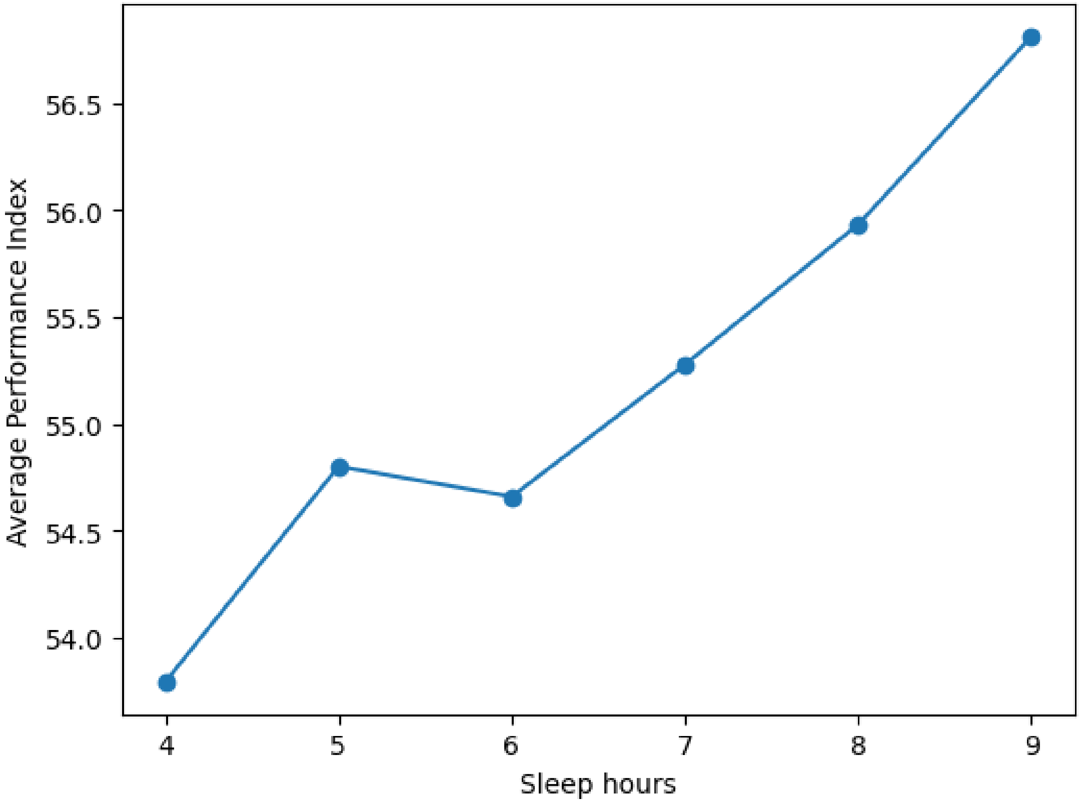
[29]:
```
Text(0, 0.5, 'Average Performance Index')
```



[19]:
```
sleephours = df.groupby("Sleep Hours")["Performance Index"].mean()
```

[28]:
```
plt.plot(sleephours.index, sleephours.values, marker='o')
plt.xlabel("Sleep hours")
plt.ylabel("Average Performance Index")
ge Performance Index")
#self explaintary
```

[28]:
```
Text(0, 0.5, 'Average Performance Index')
```


56.5
56.0
55.5
55.0
54.5
54.0
Average Performance Index
4
5
6
7
8
9
Sleep hours

```
[ ]:
```