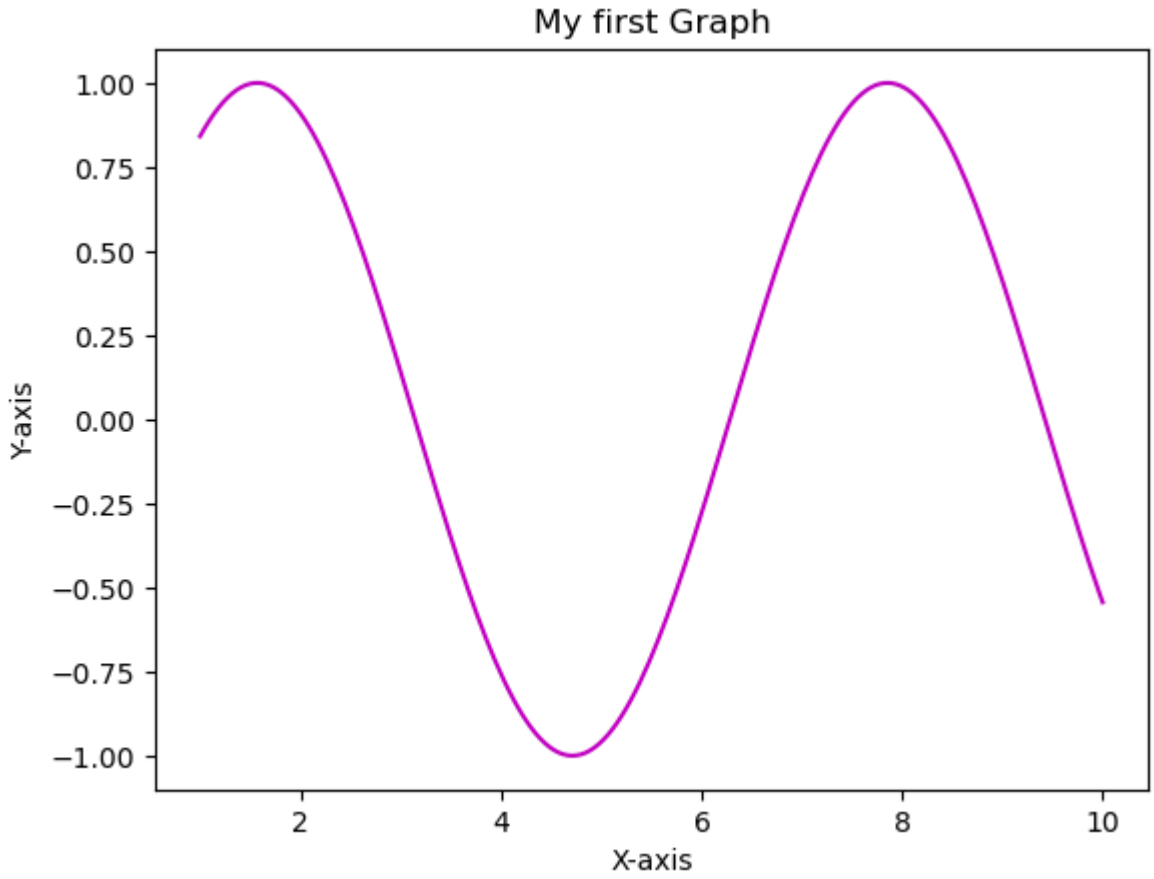
Data Visualization

By Soumya Ranjan Senapati

In [1]:
```
import matplotlib.pyplot as plt
import numpy as np
```

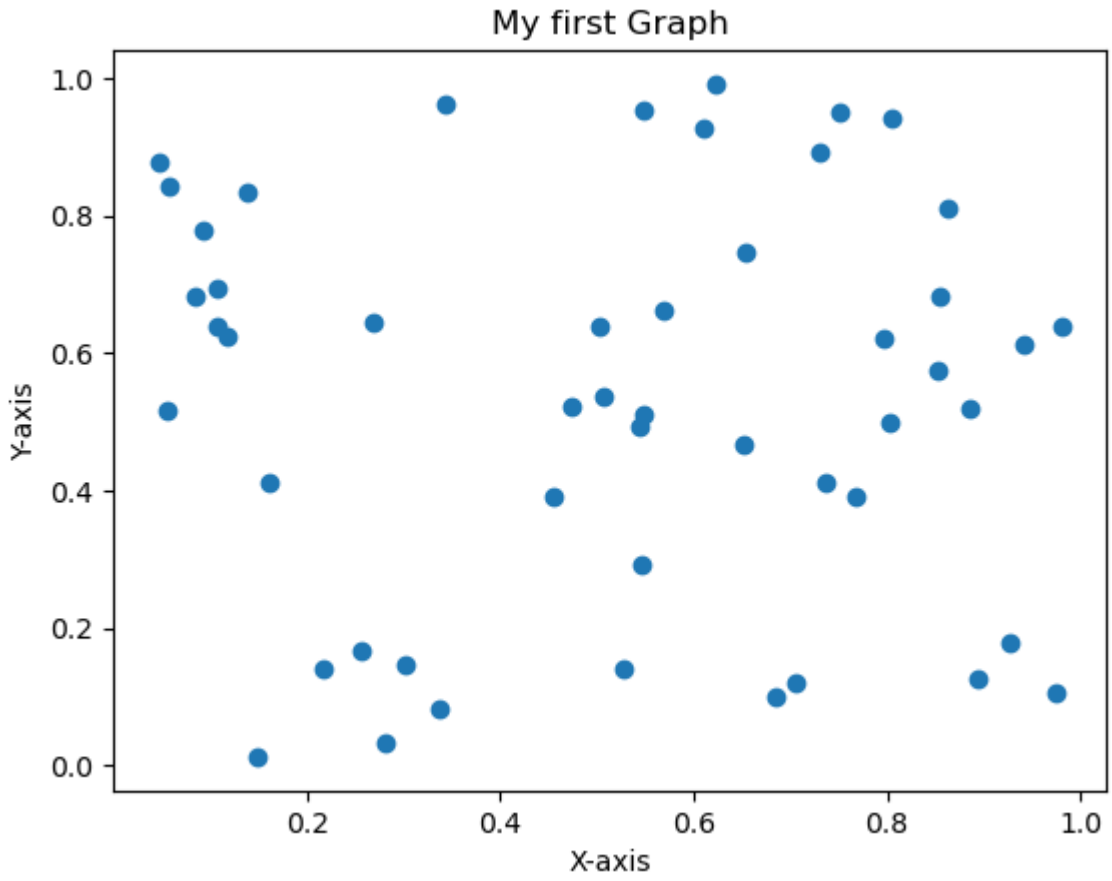
In [2]:
```
x = np.linspace(1,10,200)
y = np.sin(x)
plt.plot(x,y,color="m")
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.title("My first Graph")
```

Out[2]:
```
Text(0.5, 1.0, 'My first Graph')
```



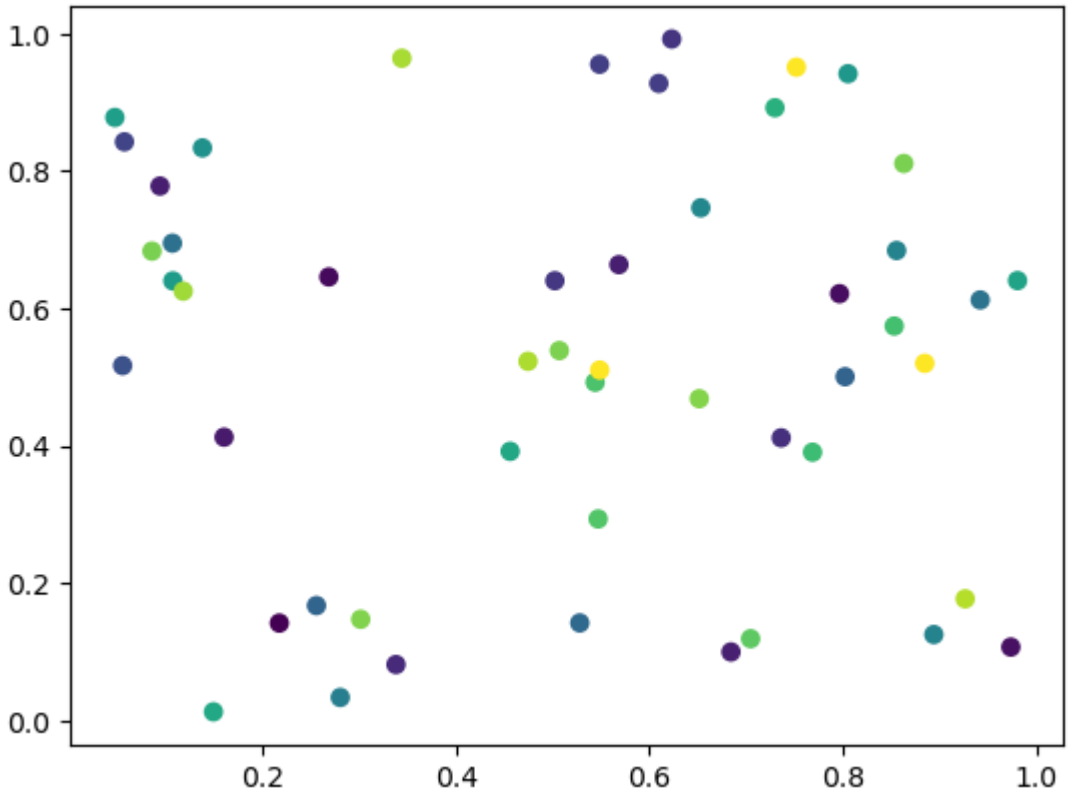
In [3]:
```
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.title("My first Graph")
x = np.random.rand(50)
y = np.random.rand(50)
plt.scatter(x,y)
```

Out[3]:
```
<matplotlib.collections.PathCollection at 0x7f4803da6bf0>
```



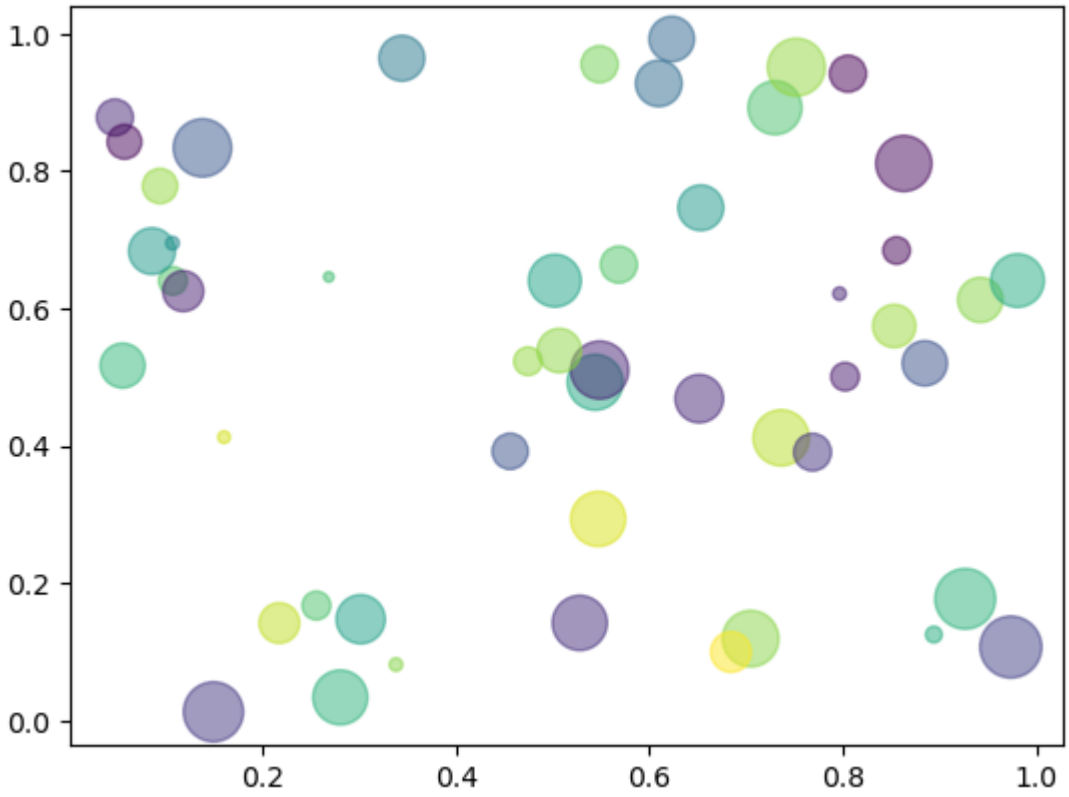
In [4]:
```
colours = np.random.rand(50)
plt.scatter(x,y,c=colours)
```

Out[4]:
```
<matplotlib.collections.PathCollection at 0x7f4803e58e20>
```



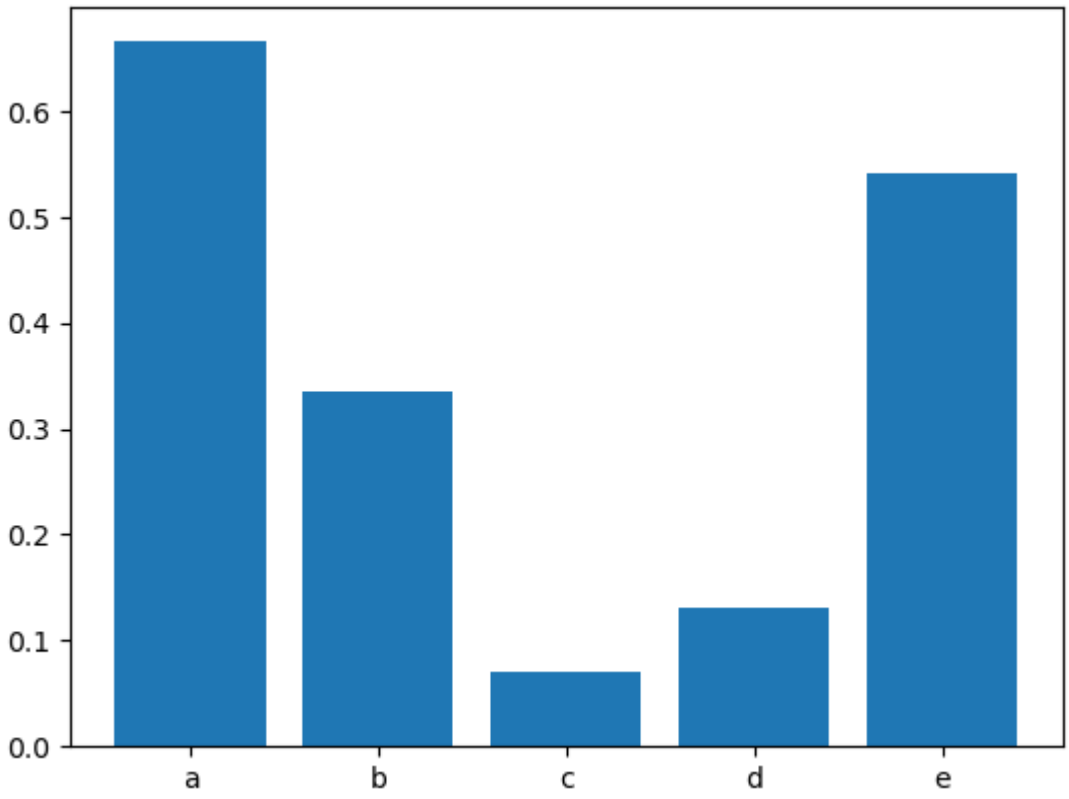
In [5]:
```
colours = np.random.rand(50)
size = 500*np.random.rand(50)
plt.scatter(x,y,c=colours,s=size,alpha=0.5)
```

Out[5]: <matplotlib.collections.PathCollection at 0x7f4803ccda20>



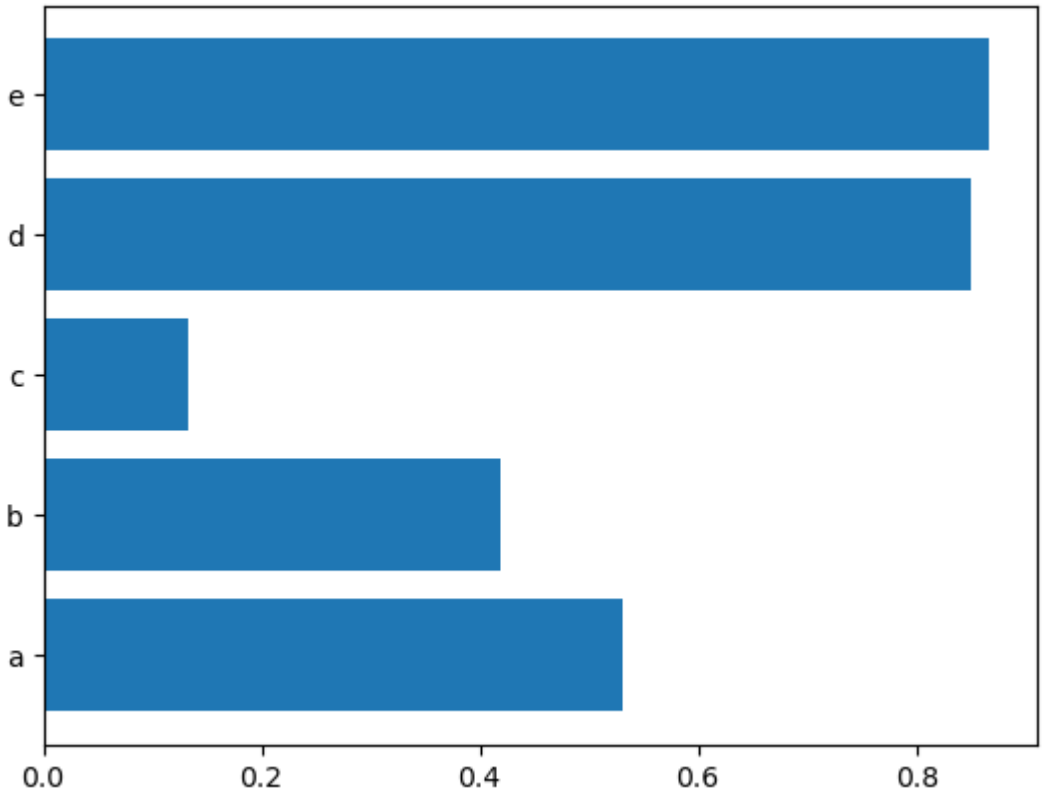
In [6]:
```
x = ['a', 'b', 'c', 'd', 'e']
y = np.random.rand(5)
plt.bar(x,y)
```

Out[6]:
```
<BarContainer object of 5 artists>
```



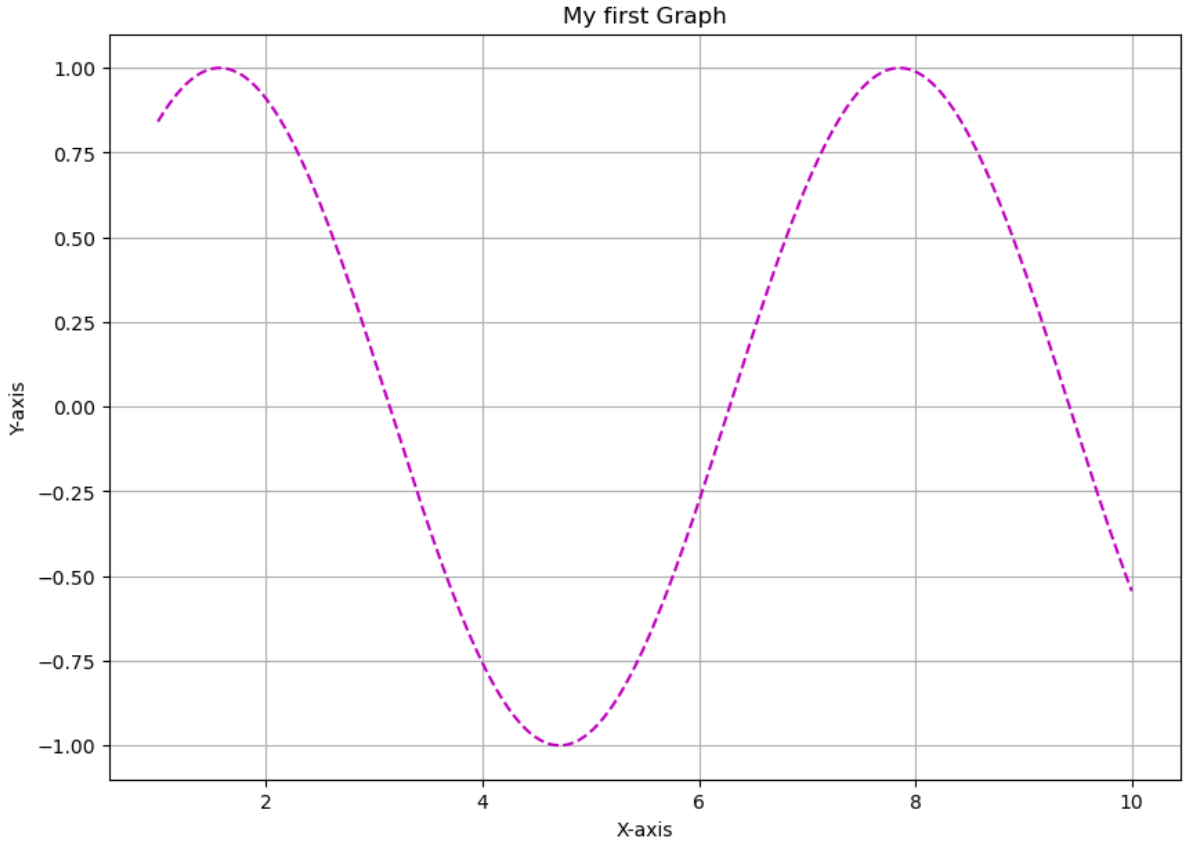
In [7]:
```
x = ['a', 'b', 'c', 'd', 'e']
y = np.random.rand(5)
plt.barh(x,y)
```

Out[7]:
```
<BarContainer object of 5 artists>
```



In [8]:
```
x = np.linspace(1,10,200)
y = np.sin(x)
plt.figure(figsize=(10,7))
plt.plot(x,y, "--b", color="m")
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.title("My first Graph")
plt.grid()
```

```
/tmp/ipykernel_768/4012355769.py:4: UserWarning: color is redundantly defined by t
he 'color' keyword argument and the fmt string "--b" (-> color='b'). The keyword a
rgument will take precedence.
  plt.plot(x,y, "--b", color="m")
```



In [9]:
```
import seaborn as sns
import matplotlib.pyplot as plt
```

In [10]:
```
df = sns.load_dataset("iris")
```

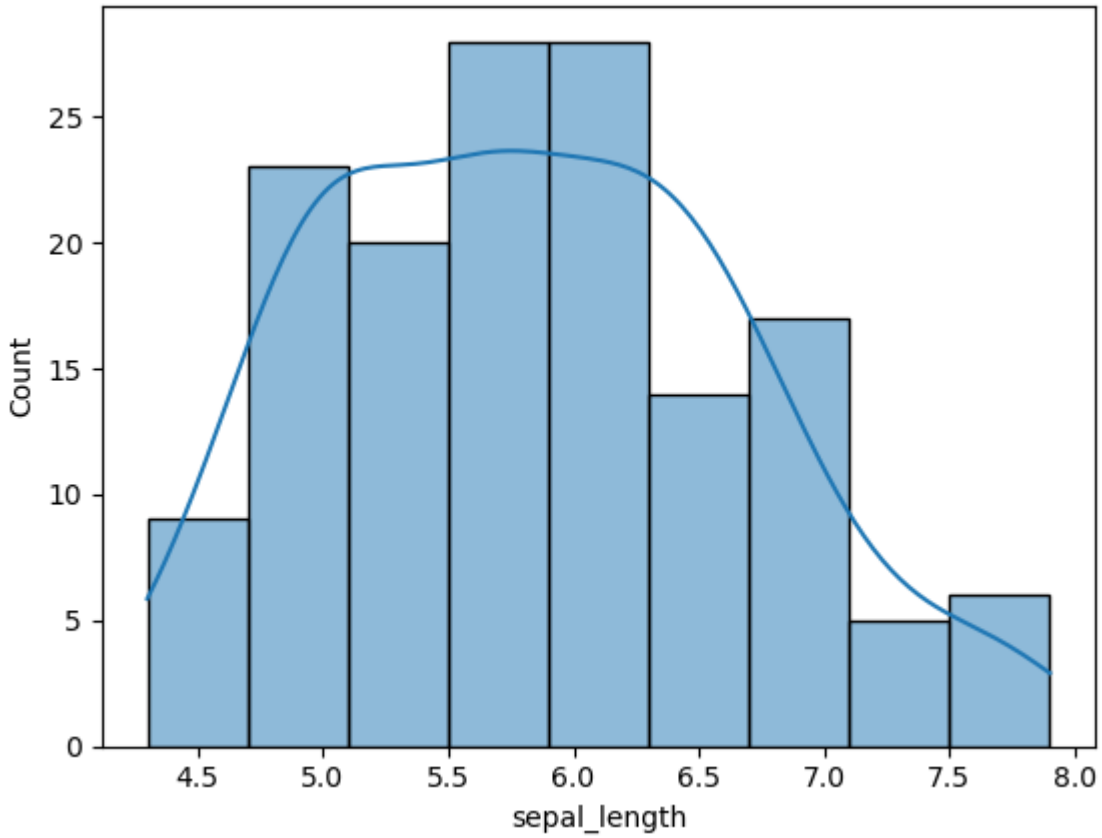
In [11]:
```
df.head()
```

Out[11]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

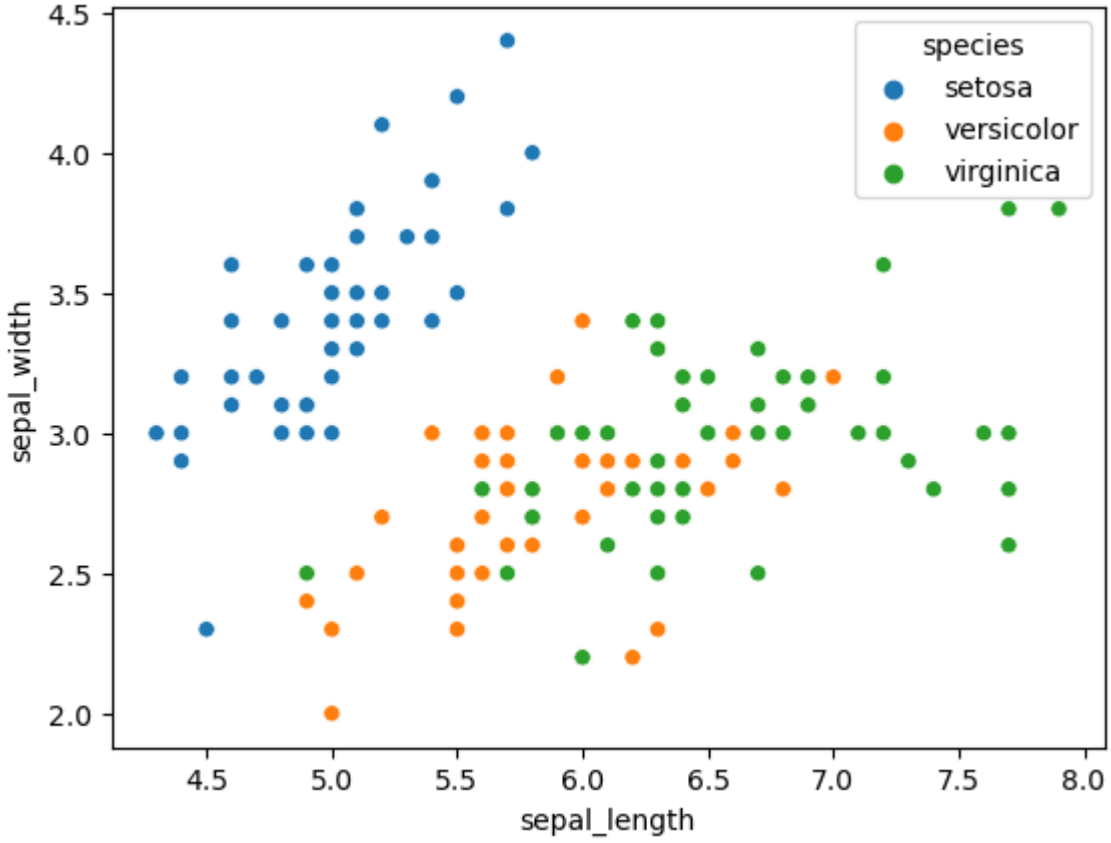
In [12]:
```
sns.histplot(df["sepal_length"],kde=True)
```

Out[12]:
```
<AxesSubplot: xlabel='sepal_length', ylabel='Count'>
```



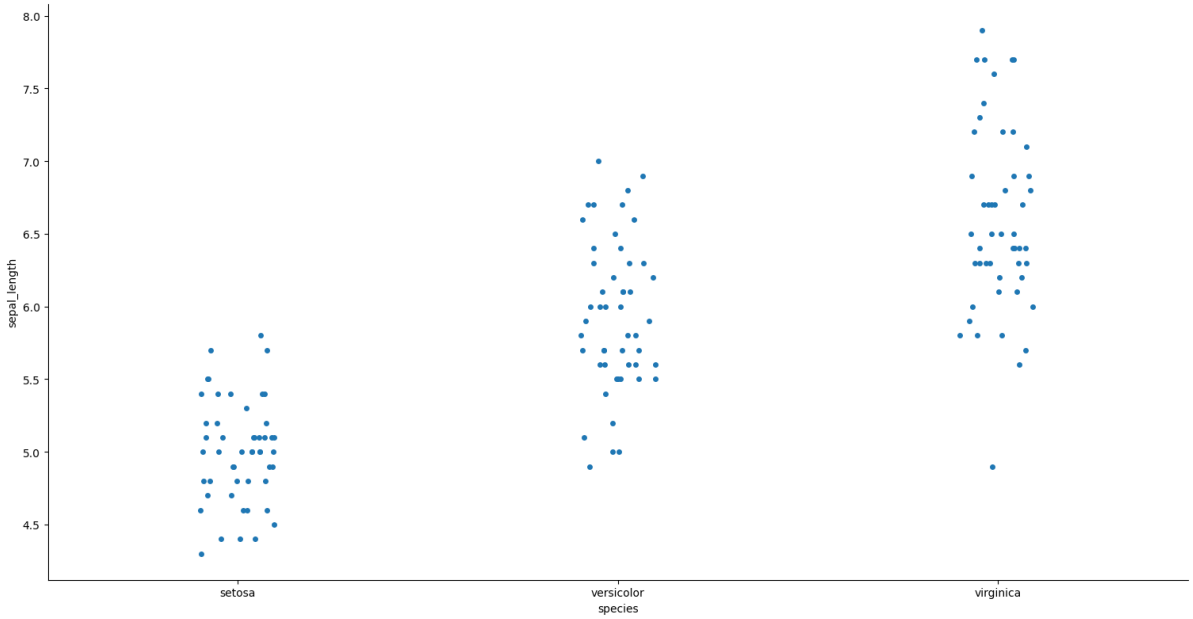
In [13]:
```
sns.scatterplot(data=df,x="sepal_length",y="sepal_width",hue="species")
```

Out[13]:
```
<AxesSubplot: xlabel='sepal_length', ylabel='sepal_width'>
```



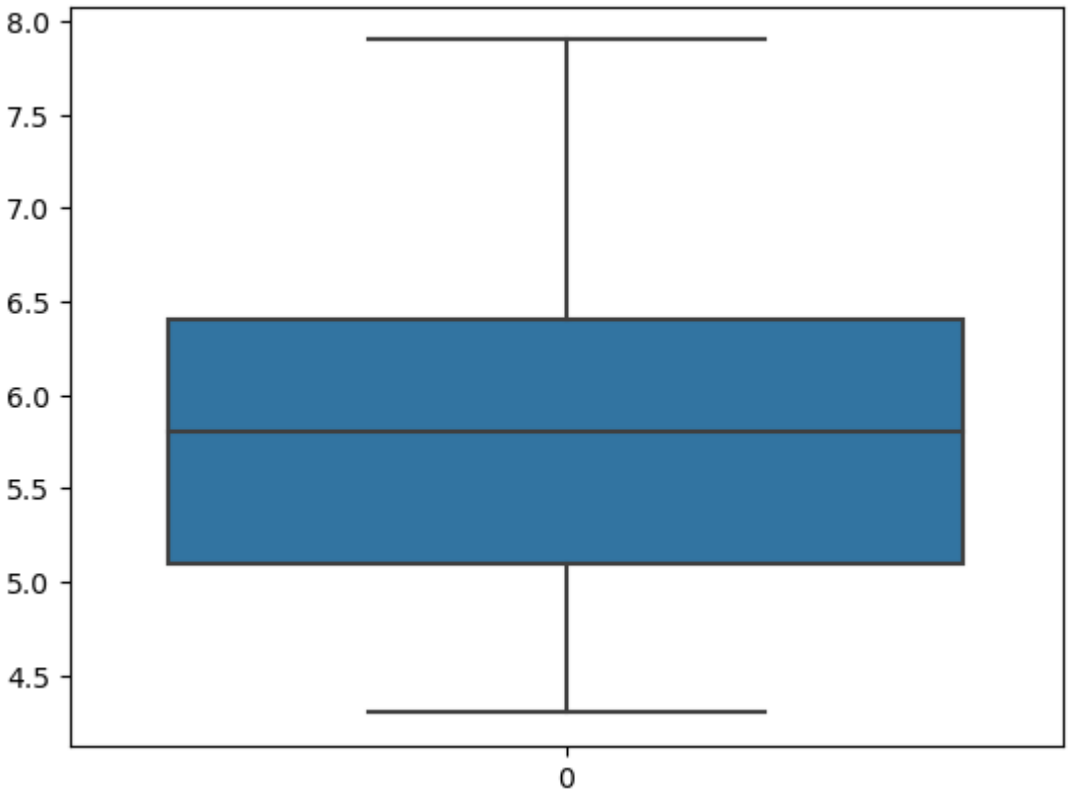
In [14]:
```
sns.catplot(data=df,x="species",y="sepal_length",height=8,aspect=1.9)
```

Out[14]:
```
<seaborn.axisgrid.FacetGrid at 0x7f47facd7490>
```



In [15]:
```
sns.boxplot(df["sepal_length"])
```

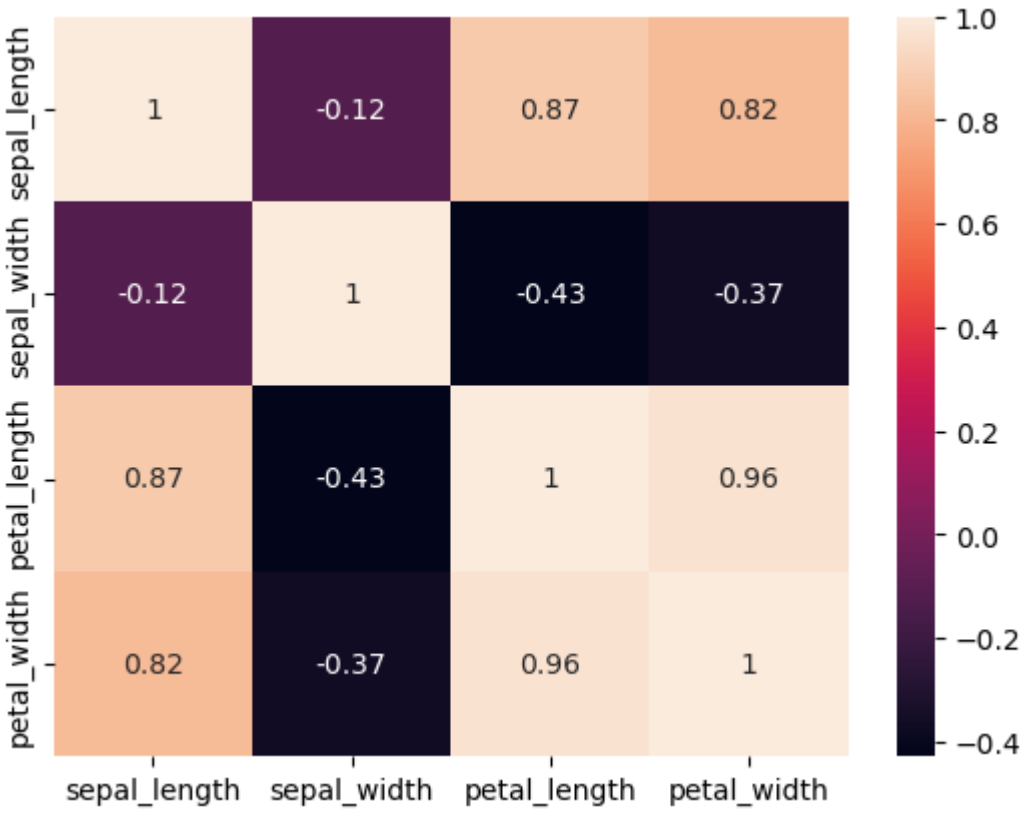
Out[15]:
```
<AxesSubplot: >
```



In [16]:
```
sns.heatmap(df.corr(),annot=True)
```

```
/tmp/ipykernel_768/4277794465.py:1: FutureWarning: The default value of numeric_on
ly in DataFrame.corr is deprecated. In a future version, it will default to False.
Select only valid columns or specify the value of numeric_only to silence this war
ning.
  sns.heatmap(df.corr(),annot=True)
```

Out[16]: <AxesSubplot: >



In [17]:
```
df = sns.load_dataset("fmri")
```

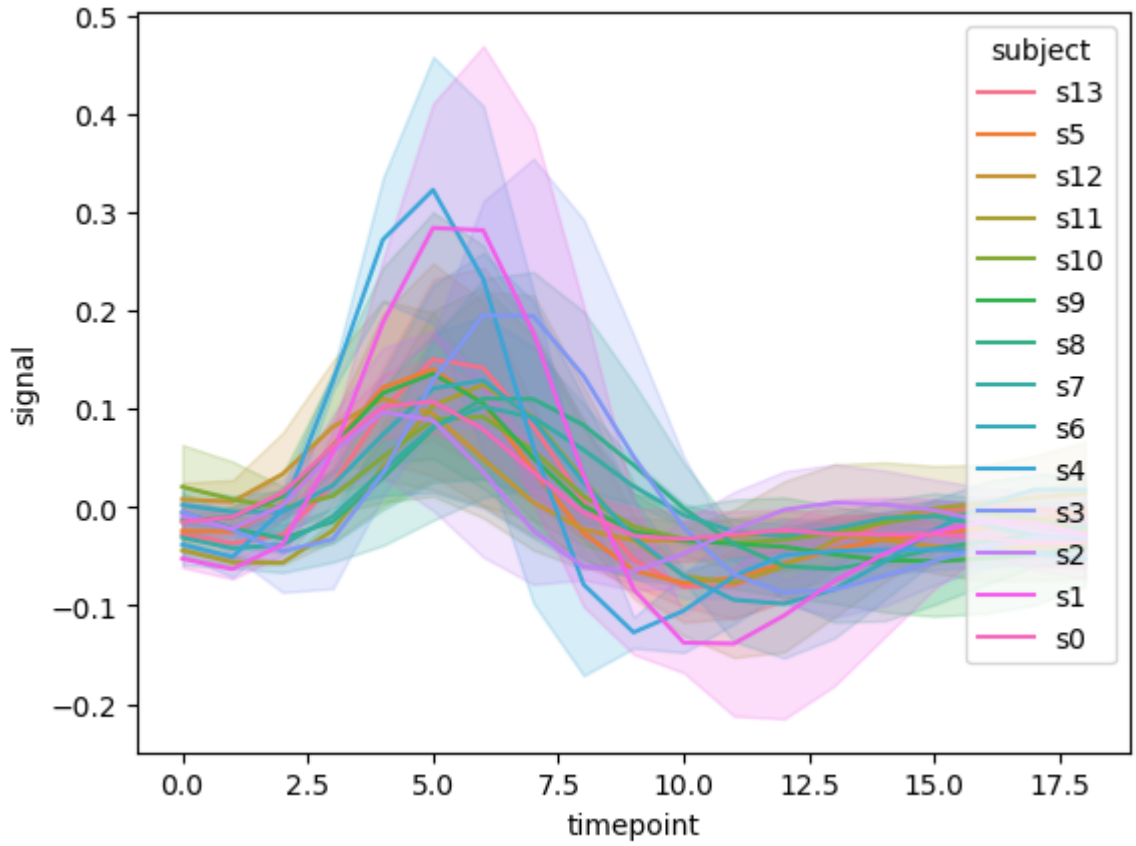
In [18]:
```
df.head()
```

Out[18]:

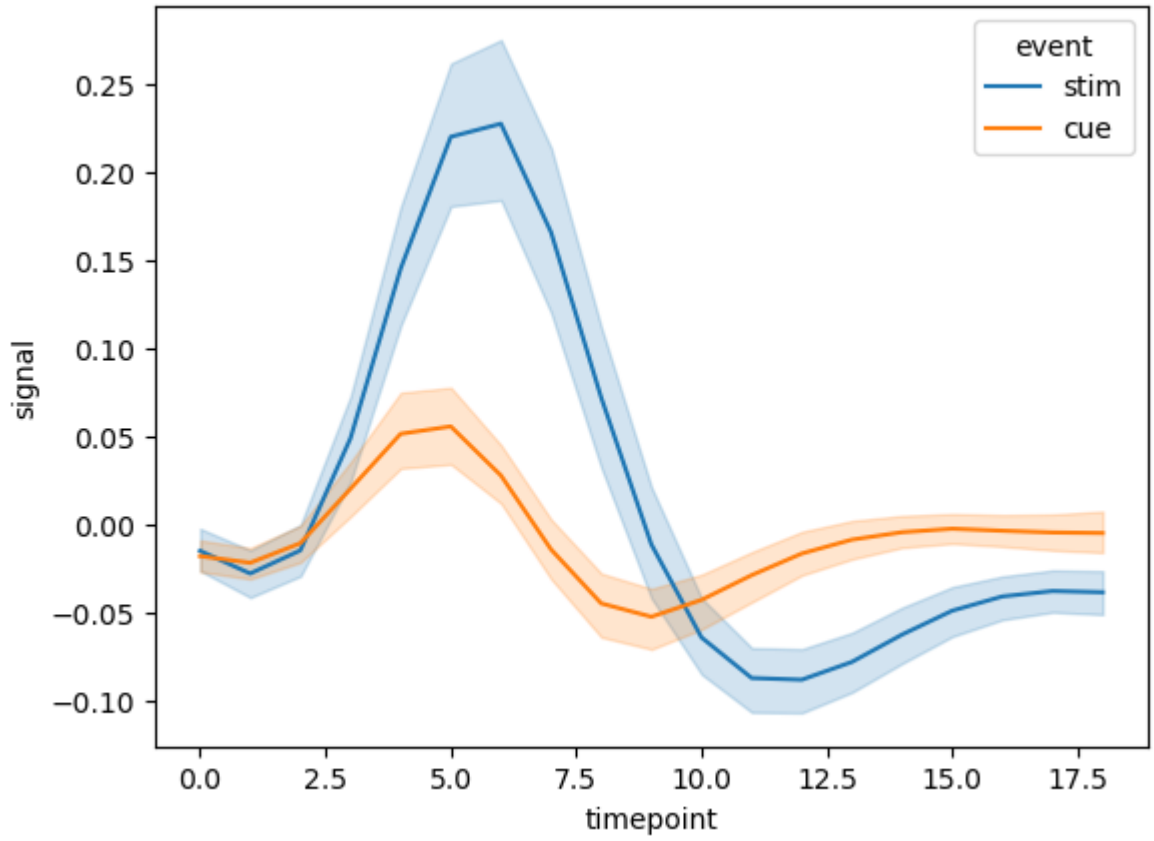
	subject	timepoint	event	region	signal
0	s13	18	stim	parietal	-0.017552
1	s5	14	stim	parietal	-0.080883
2	s12	18	stim	parietal	-0.081033
3	s11	18	stim	parietal	-0.046134
4	s10	18	stim	parietal	-0.037970

In [19]:
```
def myplotter(col):
    sns.lineplot(data=df, x="timepoint",y="signal",hue=col)
```

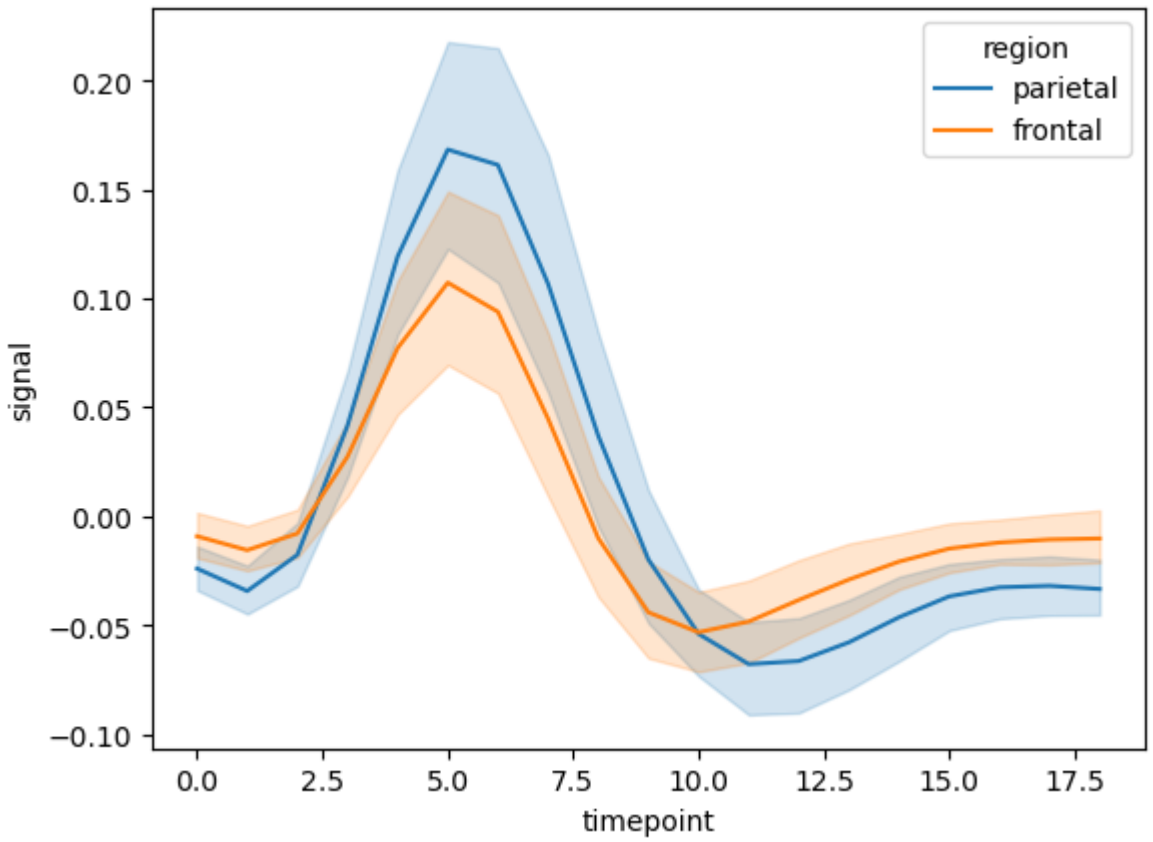
In [20]:
```
myplotter("subject")
```



In [21]:
```
myplotter("event")
```



In [22]:
```
myplotter("region")
```



In [23]:
```
df = sns.load_dataset("diamonds")
```

In [24]:
```
df.head()
```

Out[24]:

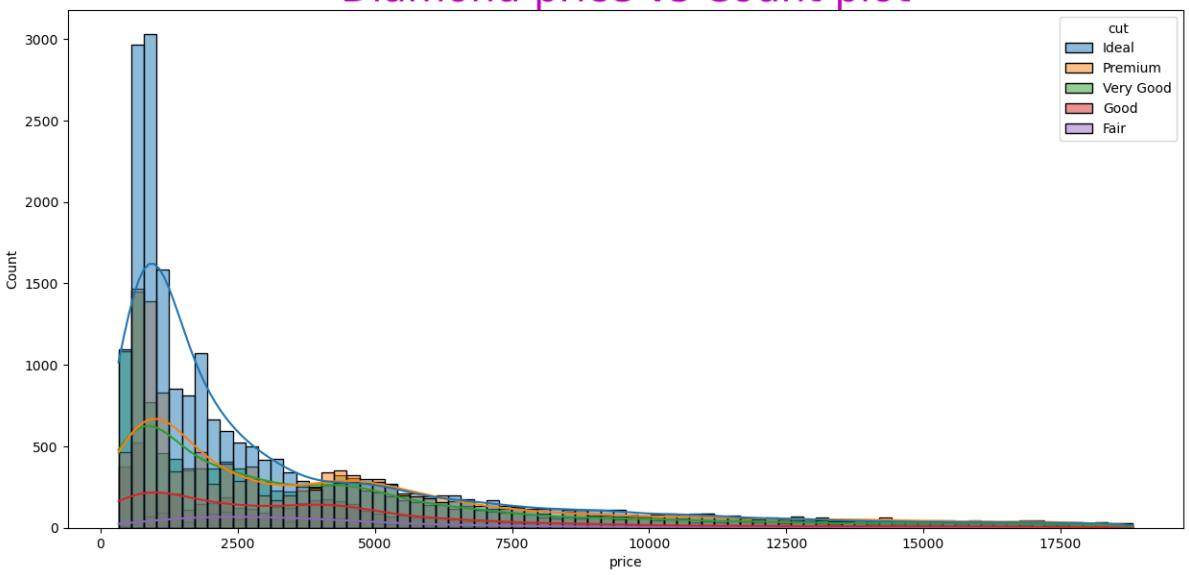
	carat	cut	color	clarity	depth	table	price	x	y	z
0	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43
1	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31
2	0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31
3	0.29	Premium	I	VS2	62.4	58.0	334	4.20	4.23	2.63
4	0.31	Good	J	SI2	63.3	58.0	335	4.34	4.35	2.75

In [25]:
```
df.shape
```

Out[25]:
```
(53940, 10)
```

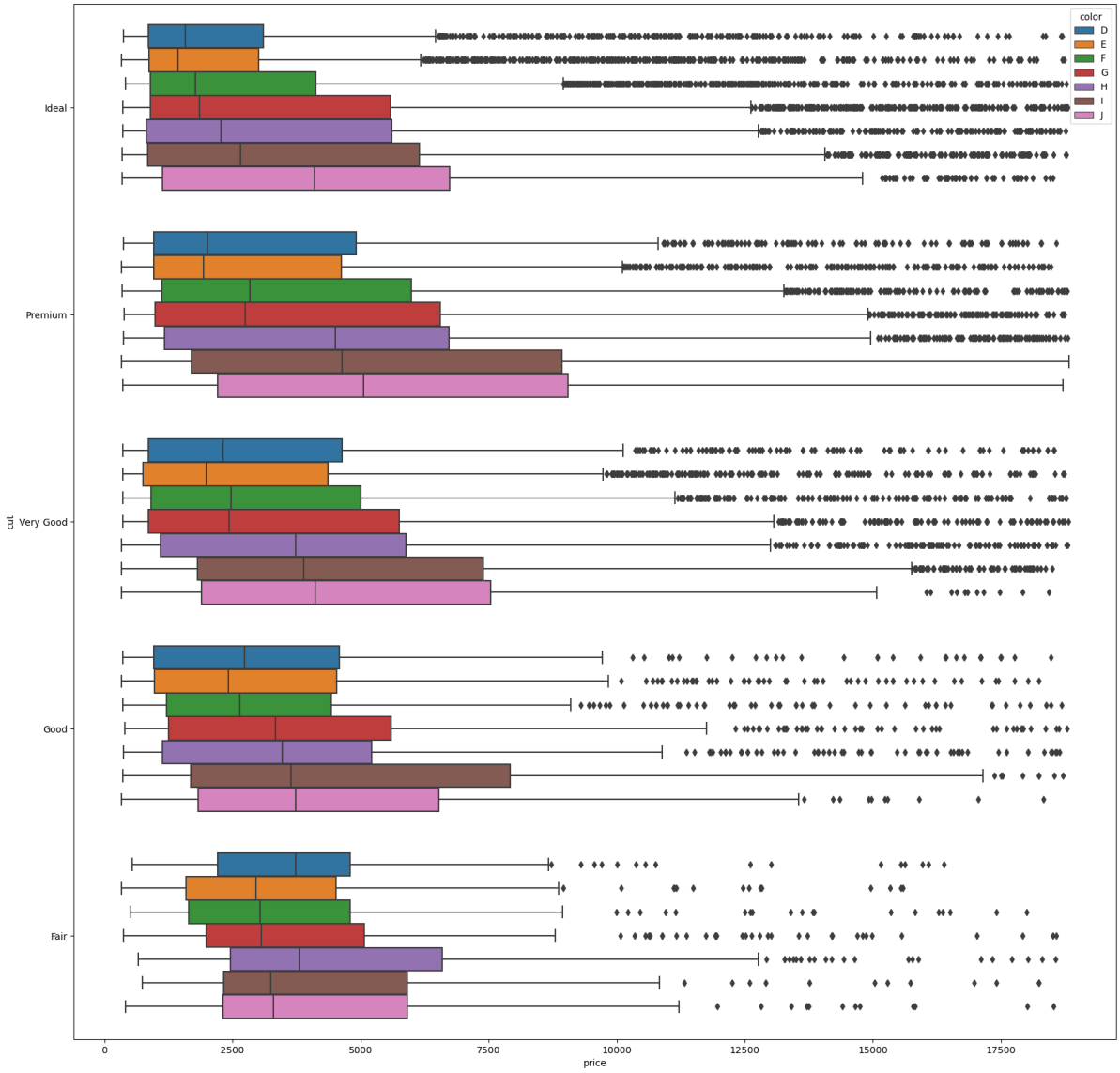
In [26]:
```
plt.figure(figsize=(15,7))
plt.title("Diamond price vs Count plot",color="m",fontsize=30)
sns.histplot(data=df,x="price",hue="cut",kde=True)
```

Out[26]:
```
<AxesSubplot: title={'center': 'Diamond price vs Count plot'}, xlabel='price', yla
bel='Count'>
```



In [27]:
```
plt.figure(figsize=(20,20))
sns.boxplot(data=df,x="price",y="cut",hue="color")
```

Out[27]:
```
<AxesSubplot: xlabel='price', ylabel='cut'>
```



In [28]:
```
df = sns.load_dataset("flights")
```

In [29]:
```
df.head()
```

Out[29]:

	year	month	passengers
0	1949	Jan	112
1	1949	Feb	118
2	1949	Mar	132
3	1949	Apr	129
4	1949	May	121

In [30]:
```
df = df[["year","passengers"]]
```

In [31]:
```
df.head()
```

Out[31]:

	year	passengers
0	1949	112
1	1949	118
2	1949	132
3	1949	129
4	1949	121

In [32]:
```
sns.heatmap(data=df)
```

Out[32]:
```
<AxesSubplot: >
```