Seaborn Assignment

January 8, 2024

0.0.1 Que 1: Name any five plots that we can plot using the Seaborn library. Also, state the uses of each plot.

0.0.2 Seaborn is a popular Python library for data visualization. Here are five types of plots that can be created using Seaborn:

- Line plot: This plot is used to visualize the relationship between two continuous variables. It is useful for showing trends over time or space. For example, you can use a line plot to show how the temperature changes over the course of a day.
- Scatter plot: This plot is used to visualize the relationship between two continuous variables. It is useful for identifying patterns and outliers in the data. For example, you can use a scatter plot to show the relationship between a person's height and weight.
- Bar plot: This plot is used to visualize the relationship between a categorical variable and a continuous variable. It is useful for comparing values across different categories. For example, you can use a bar plot to show the average salary of employees in different departments.
- Histogram: This plot is used to visualize the distribution of a continuous variable. It is useful for identifying the shape of the data and detecting outliers. For example, you can use a histogram to show the distribution of ages in a population.
- Heatmap: This plot is used to visualize the relationship between two categorical variables. It is useful for identifying patterns and trends in the data. For example, you can use a heatmap to show the number of sales of different products in different regions.

[ ]:

0.0.3 Que 2: Load the "fmri" dataset using the load_dataset function of seaborn. Plot a line plot using x = "timepoint" and y = "signal" for different events and regions.

- Note: timepoint, signal, event, and region are columns in the fmri dataset.

[1]:
```
import seaborn as sns
```

[4]:
```
fmri=sns.load_dataset('fmri')
```

[5]:
```
fmri
```

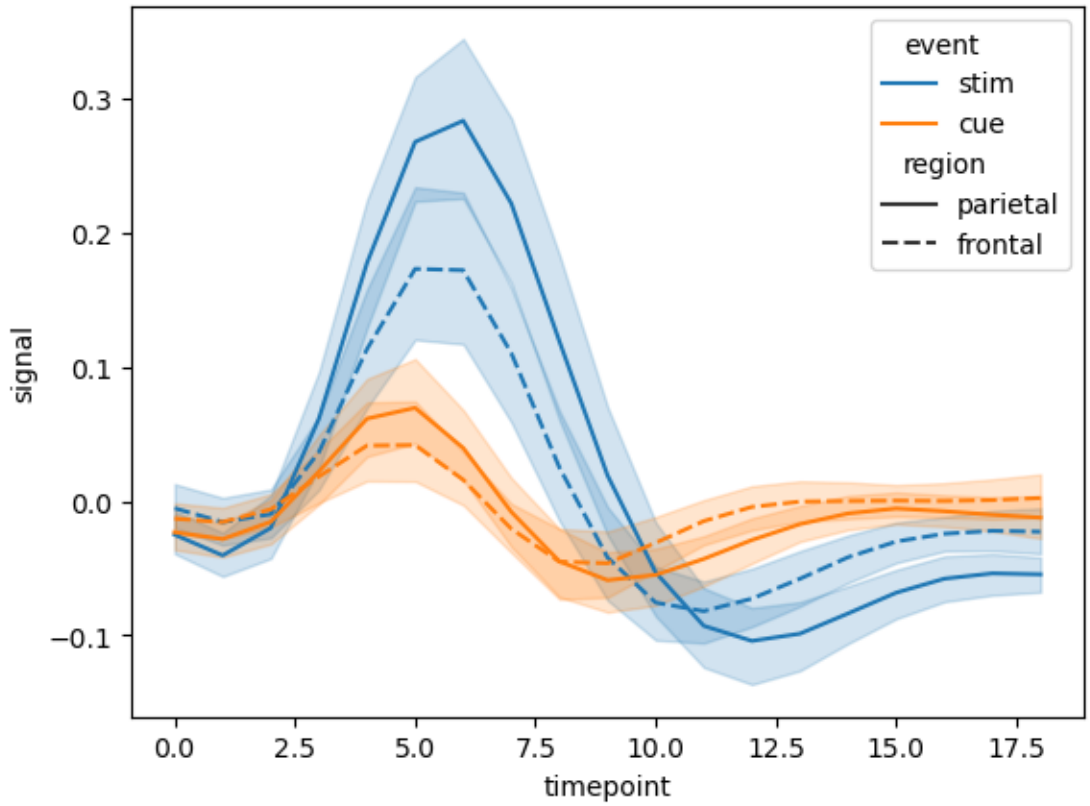
[5]:

	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
…	…	…	…	…	…
1059	s0	8	cue	frontal	0.018165
1060	s13	7	cue	frontal	-0.029130
1061	s12	7	cue	frontal	-0.004939
1062	s11	7	cue	frontal	-0.025367
1063	s0	0	cue	parietal	-0.006899

[1064 rows x 5 columns]

[13]:
```
sns.lineplot( x = 'timepoint', y =␣
↪'signal',hue='event',style='region',data=fmri)
```

[13]: <AxesSubplot: xlabel='timepoint', ylabel='signal'>



[ ]:

0.0.4 Que 3: Load the "titanic" dataset using the load_dataset function of seaborn. Plot two box plots using x = 'pclass', y = 'age' and y = 'fare'.

- Note: pclass, age, and fare are columns in the titanic dataset.

[14]:
```
import seaborn as sns
titanic=sns.load_dataset('titanic')
```

[15]:
```
titanic
```

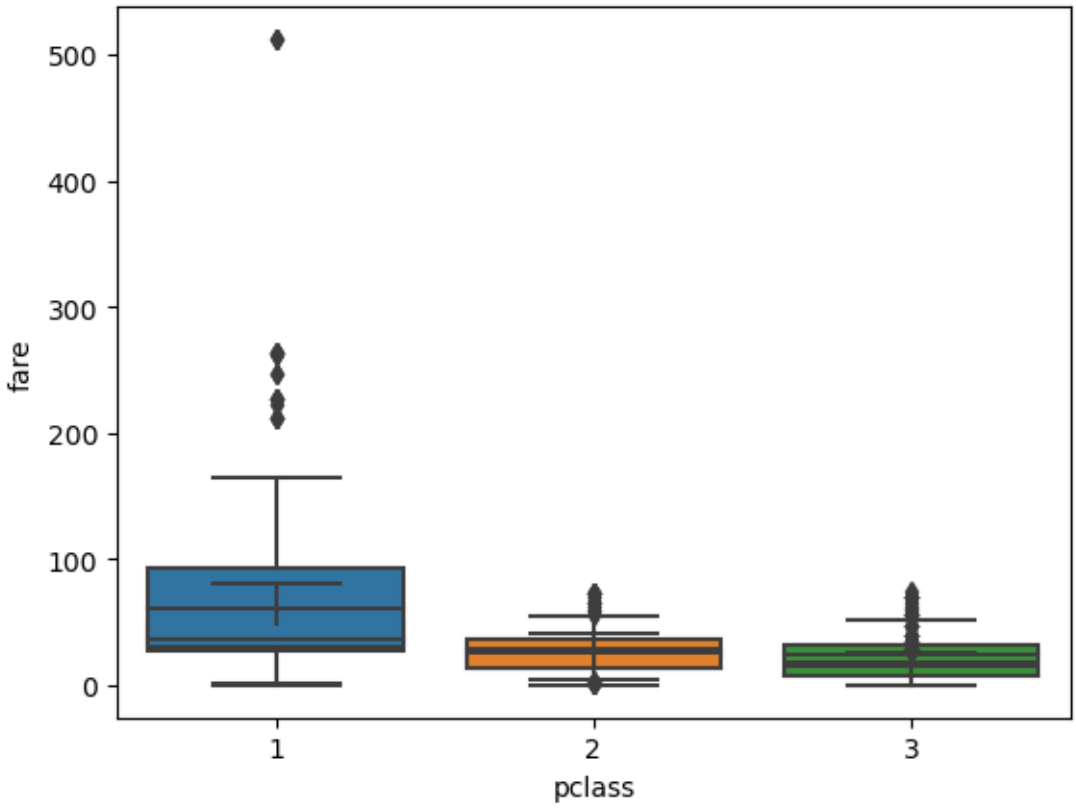
[15]:
```
     survived  pclass     sex   age  sibsp  parch     fare embarked   class  \
0           0       3    male  22.0      1      0   7.2500        S   Third
1           1       1  female  38.0      1      0  71.2833        C   First
2           1       3  female  26.0      0      0   7.9250        S   Third
3           1       1  female  35.0      1      0  53.1000        S   First
4           0       3    male  35.0      0      0   8.0500        S   Third
..        ...     ...     ...   ...    ...    ...      ...      ...     ...
886         0       2    male  27.0      0      0  13.0000        S  Second
887         1       1  female  19.0      0      0  30.0000        S   First
888         0       3  female   NaN      1      2  23.4500        S   Third
889         1       1    male  26.0      0      0  30.0000        C   First
890         0       3    male  32.0      0      0   7.7500        Q   Third

       who  adult_male deck  embark_town alive  alone
0      man        True  NaN  Southampton    no  False
1    woman       False    C    Cherbourg   yes  False
2    woman       False  NaN  Southampton   yes   True
3    woman       False    C  Southampton   yes  False
4      man        True  NaN  Southampton    no   True
..     ...         ...  ...          ...   ...    ...
886    man        True  NaN  Southampton    no   True
887  woman       False    B  Southampton   yes   True
888  woman       False  NaN  Southampton    no  False
889    man        True    C    Cherbourg   yes   True
890    man        True  NaN   Queenstown    no   True

[891 rows x 15 columns]
```

[25]:
```
sns.boxplot( x = 'pclass', y = 'age',data=titanic)
sns.boxplot( x = 'pclass', y = 'fare',data=titanic)
```

[25]:
```
<AxesSubplot: xlabel='pclass', ylabel='fare'>
```



[ ]:

0.0.5 Que 4: Use the "diamonds" dataset from seaborn to plot a histogram for the 'price' column. Use the hue parameter for the 'cut' column of the diamonds dataset.

[26]:
```
import seaborn as sns
diamonds=sns.load_dataset('diamonds')
```

[27]:
```
diamonds
```

[27]:

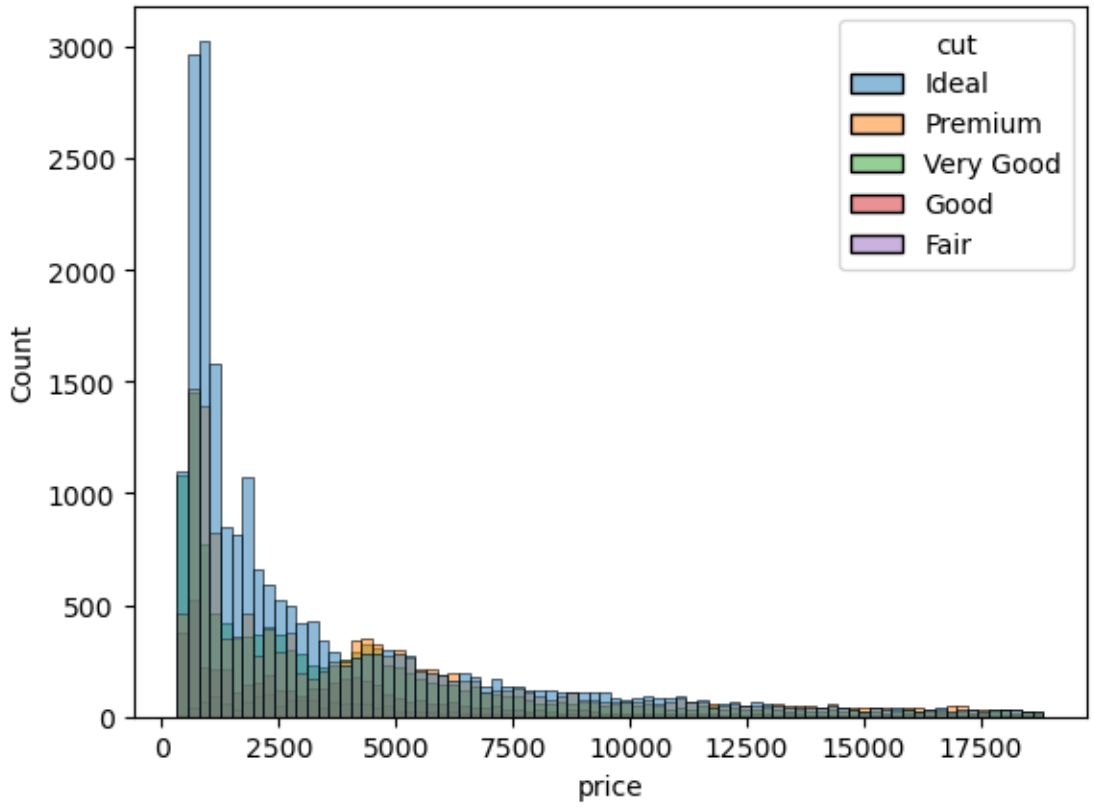
	carat	cut	color	clarity	depth	table	price	x	y	z
0	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43
1	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31
2	0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31
3	0.29	Premium	I	VS2	62.4	58.0	334	4.20	4.23	2.63
4	0.31	Good	J	SI2	63.3	58.0	335	4.34	4.35	2.75
…	…	…	…	…	…	…	…	…	…	…
53935	0.72	Ideal	D	SI1	60.8	57.0	2757	5.75	5.76	3.50
53936	0.72	Good	D	SI1	63.1	55.0	2757	5.69	5.75	3.61
53937	0.70	Very Good	D	SI1	62.8	60.0	2757	5.66	5.68	3.56

```
53938   0.86    Premium     H     SI2   61.0   58.0   2757  6.15  6.12  3.74
53939   0.75      Ideal     D     SI2   62.2   55.0   2757  5.83  5.87  3.64

[53940 rows x 10 columns]
```

[29]:
```
sns.histplot(x='price',hue='cut',data=diamonds)
```

[29]:
```
<AxesSubplot: xlabel='price', ylabel='Count'>
```



[ ]:
```

```

0.0.6 Que 5: Use the "iris" dataset from seaborn to plot a pair plot. Use the hue parameter for the "species" column of the iris dataset.

[30]:
```
import seaborn as sns
iris=sns.load_dataset('iris')
```

[31]:
```
iris
```

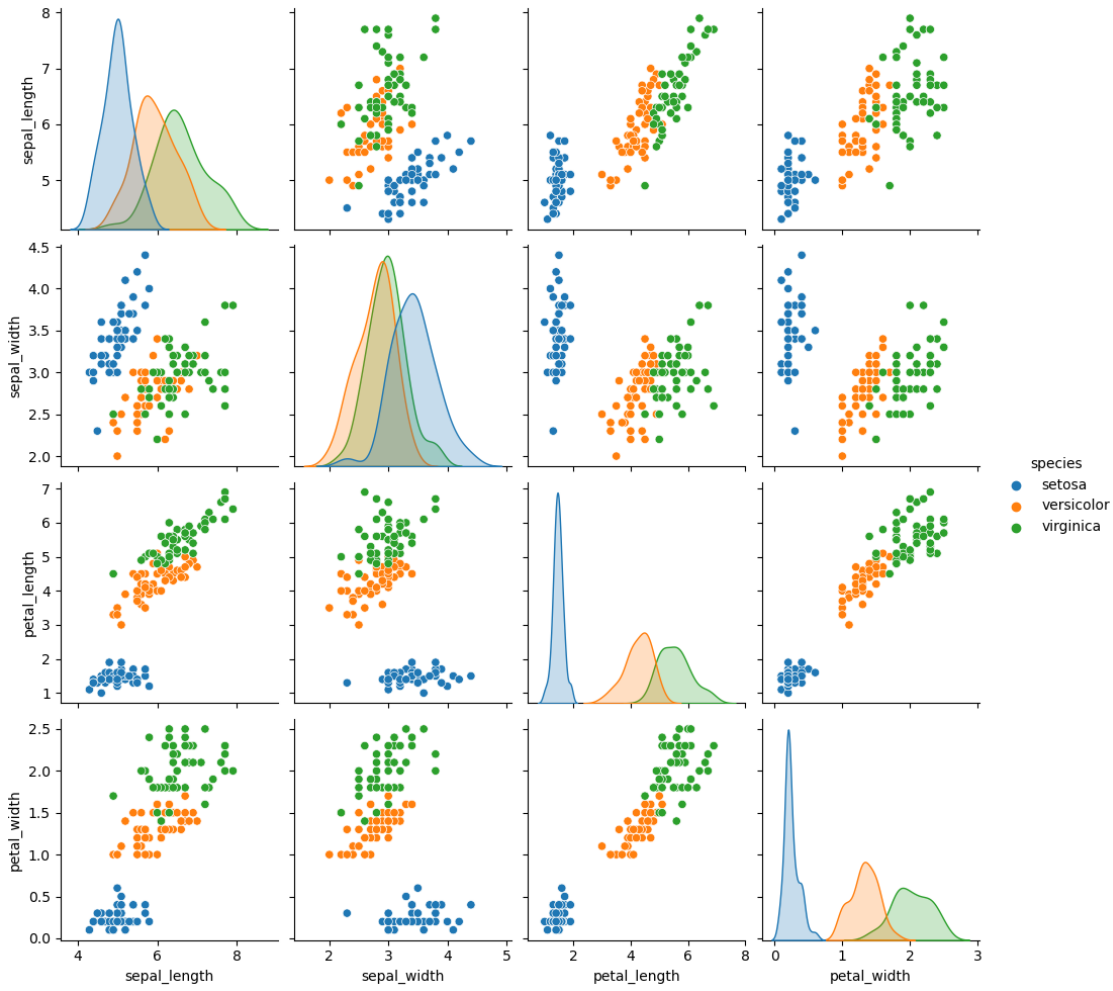
[31]:

	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
..	…	…	…	…	…
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

[150 rows x 5 columns]

[32]:
```
sns.pairplot(hue='species',data=iris)
```

[32]: <seaborn.axisgrid.PairGrid at 0x7f001a81b2b0>



[ ]:

0.0.7 Que 6: Use the "flights" dataset from seaborn to plot a heatmap.

[39]:
```
import seaborn as sns
flights=sns.load_dataset('flights')
```

[40]:
```
flights
```

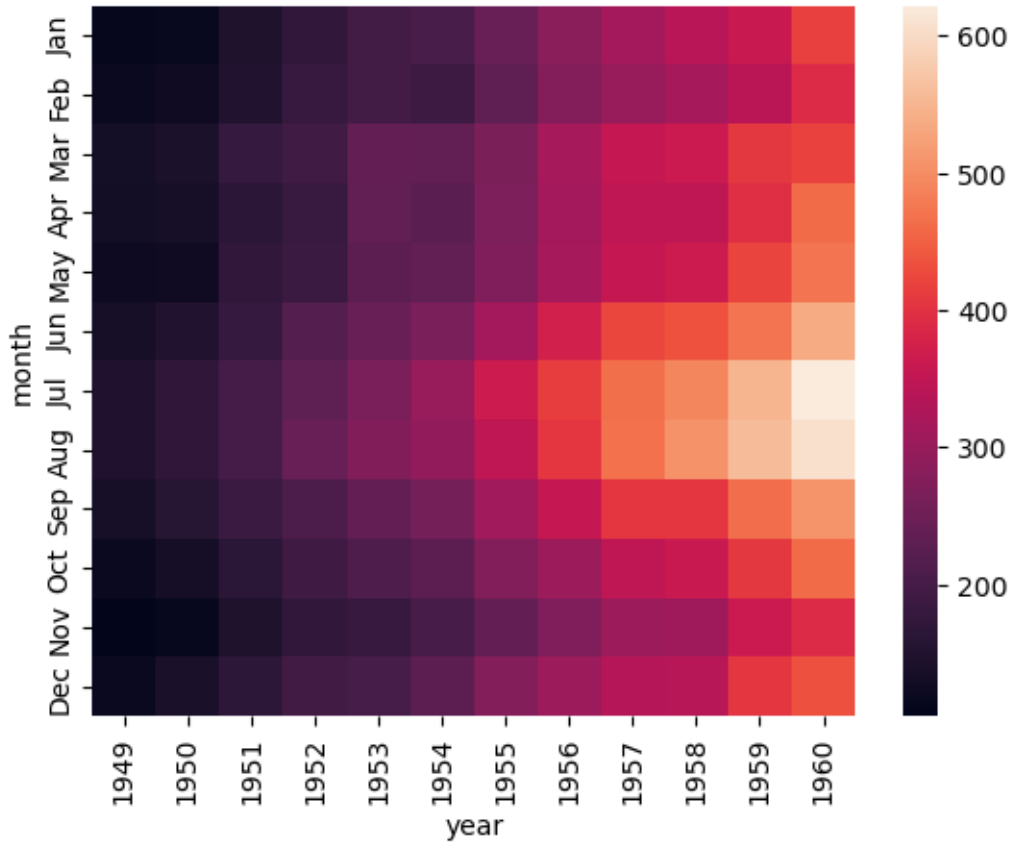
[40]:
```
     year month  passengers
0    1949   Jan         112
1    1949   Feb         118
2    1949   Mar         132
3    1949   Apr         129
4    1949   May         121
..    …    …          …
139  1960   Aug         606
140  1960   Sep         508
141  1960   Oct         461
142  1960   Nov         390
143  1960   Dec         432

[144 rows x 3 columns]
```

[45]:
```
flights = flights.pivot("month", "year", "passengers")
sns.heatmap(flights)
```

```
/tmp/ipykernel_2201/3798779447.py:1: FutureWarning: In a future version of
pandas all arguments of DataFrame.pivot will be keyword-only.
  flights = flights.pivot("month", "year", "passengers")
```

[45]:
```
<AxesSubplot: xlabel='year', ylabel='month'>
```


month
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
year
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
600
500
400
300
200

[ ]:

[ ]: