2. DATA VISUALIZATION – PART - 2

Contents

2. DATA VISUALIZATION – PART - 2

1. Titanic Introduction

- ✓ The Titanic was known as the unsinkable ship and was the largest, most luxurious passenger ship.
- ✓ Sadly, the British ocean liner sank on April 15, 1912, killing over many people while just few people got survived.
- ✓ Let's do analyse titanic dataset

2. Seaborn library

- ✓ Seaborn is advanced data visualization library.
- ✓ By using this we can visualize the data.

2.1. Environment

- ✓ We can install this library by using pip command.

Seaborn installation

```
pip install seaborn
```

Program Name: Loading titanic dataset
demo1.py

```
import seaborn as sns

df = sns.load_dataset('titanic')
print(df.head())
```

Output

```
   survived  pclass     sex   age  sibsp  parch  ...    who adult_male deck  embark_town  alive  alone
0         0       3    male  22.0      1      0  ...    man       True  NaN  Southampton     no  False
1         1       1  female  38.0      1      0  ...  woman      False    C    Cherbourg    yes  False
2         1       3  female  26.0      0      0  ...  woman      False  NaN  Southampton    yes   True
3         1       1  female  35.0      1      0  ...  woman      False    C  Southampton    yes  False
4         0       3    male  35.0      0      0  ...    man       True  NaN  Southampton     no   True

[5 rows x 15 columns]
```

3. Titanic data set understanding

- ✓ Let's understand the titanic dataset.
- ✓ Data Set Column Descriptions
  - pclass: Passenger Class (1 = 1st; 2 = 2nd; 3 = 3rd)
  - survived: Survival (0 = No; 1 = Yes)
  - name: Name
  - sex: type of gender
  - age: Age
  - sibsp: Number of siblings/spouses aboard
  - parch: Number of parents/children aboard
  - fare: Passenger fare (British pound)
  - embarked: Port of embarkation (C = Cherbourg; Q = Queenstown; S = Southampton)
  - adult_male: A male 18 or older (0 = No, 1=Yes)
  - deck: Deck of the ship
  - who: man (18+), woman (18+), child (<18)
  - alive: Yes, no
  - embarked_town: Port of embarkation (Cherbourg, Queenstown, Southampton)
  - class: Passenger class (1st; 2nd; 3rd)
  - alone: 1 = alone, 0 = not alone (you have at least 1 sibling, spouse, parent or child on board)

Program Name: Number of rows and columns
demo2.py

```
import seaborn as sns

df = sns.load_dataset('titanic')
print(df.shape)
```

Output

```
(891, 15)
```

Program Name: Display the columns
demo3.py

```
import seaborn as sns

df = sns.load_dataset('titanic')
print(df.columns)
```

Output

```
Index(['survived', 'pclass', 'sex', 'age', 'sibsp', 'parch', 'fare',
'embarked', 'class', 'who', 'adult_male', 'deck', 'embark_town',
'alive', 'alone'], dtype='object')
```

Program Name: DataFrame information
demo4.py

```
import seaborn as sns

df = sns.load_dataset('titanic')
df.info()
```

Output

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 15 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   survived     891 non-null    int64
 1   pclass       891 non-null    int64
 2   sex          891 non-null    object
 3   age          714 non-null    float64
 4   sibsp        891 non-null    int64
 5   parch        891 non-null    int64
 6   fare         891 non-null    float64
 7   embarked     889 non-null    object
 8   class        891 non-null    category
 9   who          891 non-null    object
 10  adult_male   891 non-null    bool
 11  deck         203 non-null    category
 12  embark_town  889 non-null    object
 13  alive        891 non-null    object
 14  alone        891 non-null    bool
dtypes: bool(2), category(2), float64(2), int64(4), object(5)
memory usage: 80.7+ KB
```

Program Name: unique values for sex(gender) column
demo5.py

```
import seaborn as sns
import pandas as pd

df = sns.load_dataset('titanic')
result = df['sex'].unique()
print(result)
```

Output

```
['male' 'female']
```

4. Data Analysis

- ✓ From the data **max price/fare** a passenger paid for a ticket in this data set was 512.3292 British pounds, and the **minimum price/fare** was 0 British pounds.
- ✓ There is missing data for age column.
- ✓ The **mean** age is 29.699 and the oldest passenger in this data set was 80 years old, while the youngest was only .42 years old (about 5 months).

Program Name: describe() method
demo6.py

```
import seaborn as sns

df = sns.load_dataset('titanic')
print(df.describe())
```

Output

```
          survived      pclass         age       sibsp       parch        fare
count  891.000000  891.000000  714.000000  891.000000  891.000000  891.000000
mean     0.383838    2.308642   29.699118    0.523008    0.381594   32.204208
std      0.486592    0.836071   14.526497    1.102743    0.806057   49.693429
min      0.000000    1.000000    0.420000    0.000000    0.000000    0.000000
25%      0.000000    2.000000   20.125000    0.000000    0.000000    7.910400
50%      0.000000    3.000000   28.000000    0.000000    0.000000   14.454200
75%      1.000000    3.000000   38.000000    1.000000    0.000000   31.000000
max      1.000000    3.000000   80.000000    8.000000    6.000000  512.329200
```

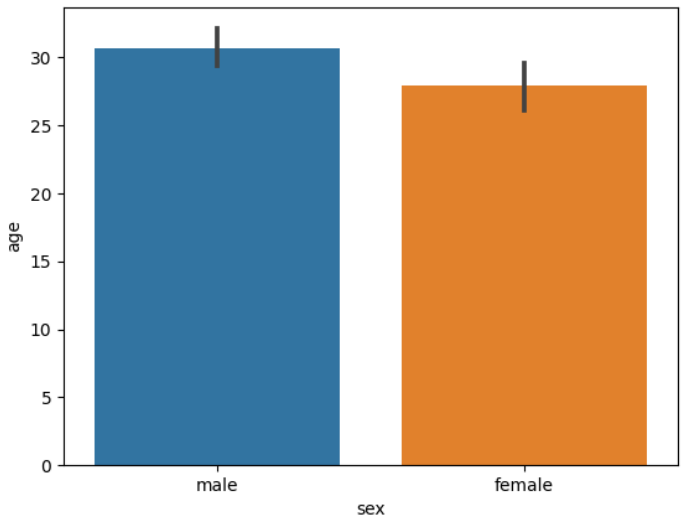
5. Bar Plot

✓ A bar plot shows the mean value of every value in a categorical column.

Program Name: Creating bar plot
demo7.py

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

import warnings
warnings.filterwarnings('ignore')

df = sns.load_dataset('titanic')
sns.barplot(x = 'sex', y = 'age', data = df)

plt.show()
```

Output



✓ The plot clearly shows that the average age for all male passengers is above 30 while the average age of the female passengers is between 25 and 30.

6. Get a count of the number of survivors

- ✓ **0** represents not survived
- ✓ **1** means survived.

Program Name: Get a count of the number of survivors
demo8.py

```
import seaborn as sns

df = sns.load_dataset('titanic')
print(df['survived'].value_counts())
```

Output

```
0    549
1    342
Name: survived, dtype: int64
```

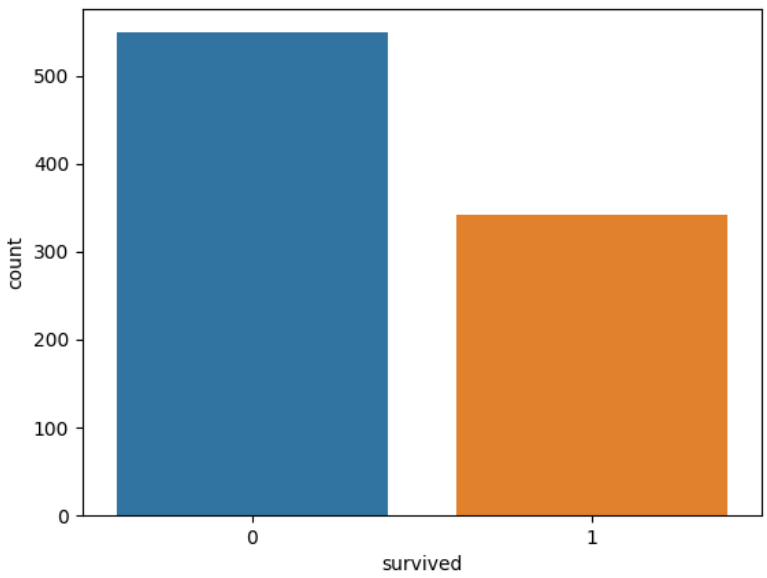
7. Count Plot

- ✓ This type of plot is similar to the bar plot, it displays the count of categories in a specific column.
- ✓ By using we can calculate the total number or count of survived and not survived.

Program Name: Get a count of the number of survivors
demo9.py

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

import warnings
warnings.filterwarnings('ignore')

df = sns.load_dataset('titanic')
sns.countplot(x = "survived", data = df)

plt.show()
```

Output



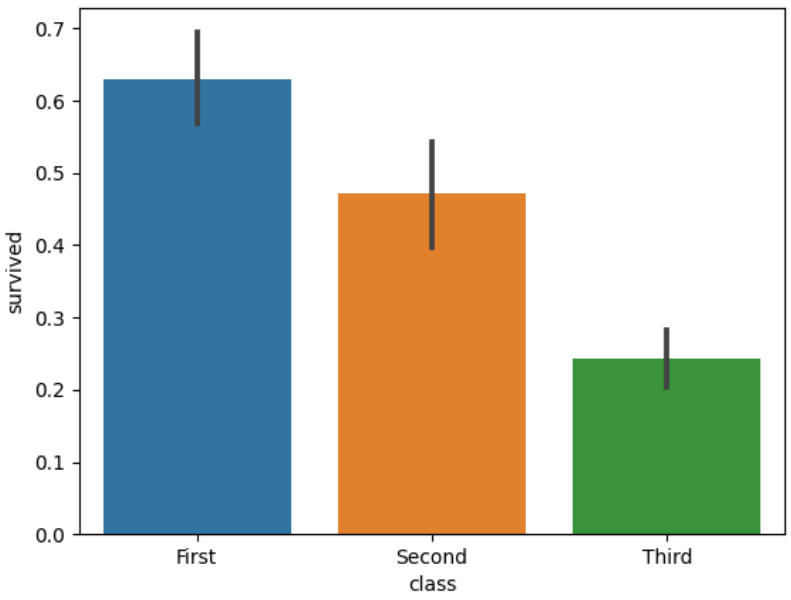
9. Plot the survival rate of each class

- ✓ A little over 60% of the passengers in first class survived. Less than 30% of passengers in third class survived.
- ✓ That means less than half of the passengers in third class survived, compared to the passengers in first class.

Program Name: Plot the survival rate of each class
demo10.py

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

import warnings
warnings.filterwarnings('ignore')

df = sns.load_dataset('titanic')
sns.barplot(x = 'class', y = 'survived', data = df)

plt.show()
```

Output



10. Let's understand the survival rate by gender and class.

- ✓ From the pivot table below, we see that females in first class had a survival rate of about 96.8%, meaning the majority of them survived.
- ✓ Males in third class had the lowest survival rate at about 13.54%, meaning the majority of them did not survive.

Program Name: Plot the survival rate of each class
demo11.py

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

import warnings
warnings.filterwarnings('ignore')

df = sns.load_dataset('titanic')

result = df.pivot_table('survived', index = 'sex', columns = 'class')

print(result)
```

Output

```
class       First    Second     Third
sex
female  0.968085  0.921053  0.500000
male    0.368852  0.157407  0.135447
```

11. Let's understand the survival rate by gender, age and class.

Program Name: Plot the survival rate by gender, age and class
demo12.py

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

import warnings
warnings.filterwarnings('ignore')

df = sns.load_dataset('titanic')

diff_ages = pd.cut(df['age'], [0, 18, 80])

result = df.pivot_table('survived', ['sex', diff_ages], 'class')

print(result)
```

Output

```
class                First    Second     Third
sex    age
female (0, 18]    0.909091  1.000000  0.511628
       (18, 80]   0.972973  0.900000  0.423729
male   (0, 18]    0.800000  0.600000  0.215686
       (18, 80]   0.375000  0.071429  0.133663
```

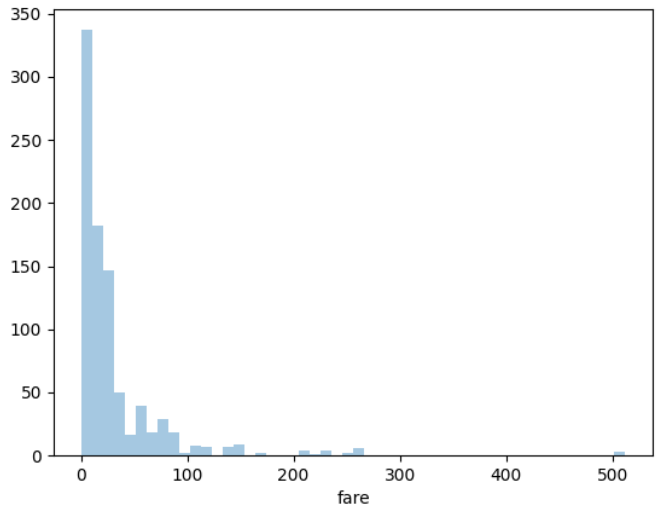
12. Dist Plot

- ✓ To create distribution plot we need to call distplot(p) function.
- ✓ This will create histogram distribution of a dataset for a column.
- ✓ We can plot the price of the ticket for every passenger

Program Name: Finding Most of the tickets, using dist plot
demo13.py

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

import warnings
warnings.filterwarnings('ignore')

df = sns.load_dataset('titanic')
sns.distplot(df['fare'], kde = False)

plt.show()
```

Output



- ✓ The above plot shows that most of the tickets have been sold between 0 and 50 dollars.

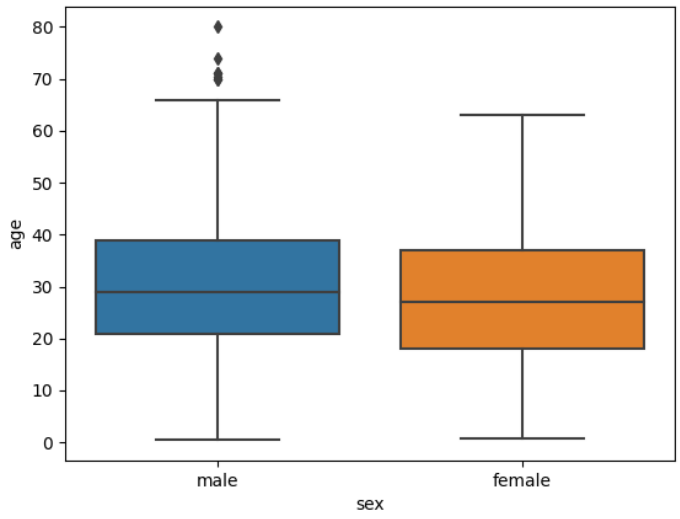
13. Box Plot

- ✓ The box plot is used to display the distribution of the categorical data in the form of quartiles like Q1, Q2, Q3 and Q3.
- ✓ The center of the box shows the median value.
- ✓ Now let's plot a box plot that displays the distribution for the age with respect to each gender.

Program Name: Creating a boxplot with sex column(gender) survived
demo14.py

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

import warnings
warnings.filterwarnings('ignore')

df = sns.load_dataset('titanic')
sns.boxplot(x = 'sex', y = 'age', data = df)

plt.show()
```

Output



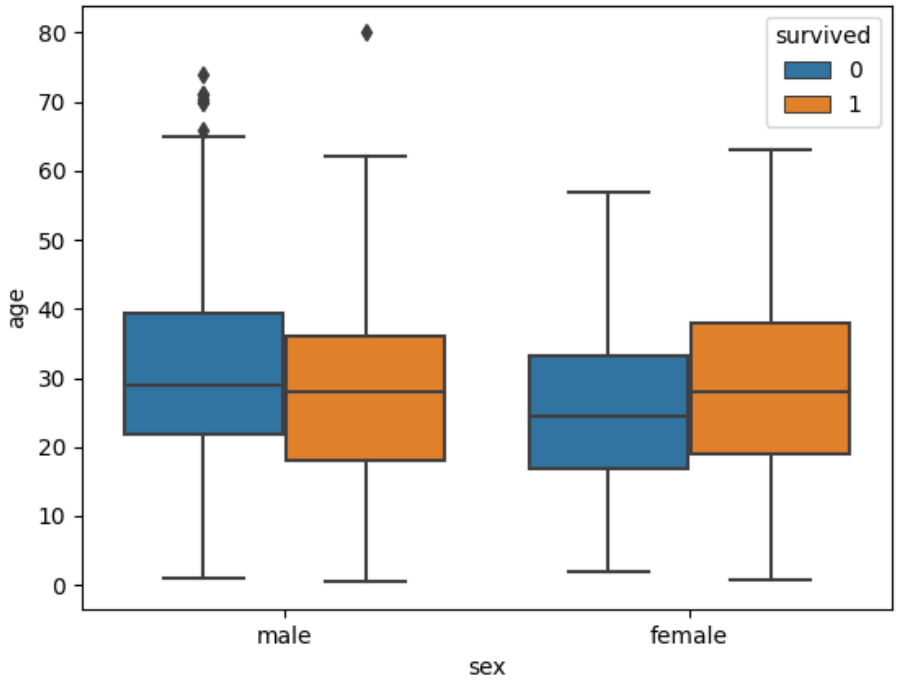
- ✓ The first quartile-Q1 starts at around 3 and ends at 22 which mean that 25% of the passengers are aged between 5 and 22.
- ✓ The second quartile-Q2 starts at around 23 and ends at around 28 which mean that 25% of the passengers are aged between 23 and 28.
- ✓ Similarly, the third quartile-Q3 starts and ends between 29 and 38, hence 25% passengers are aged within this range and finally the fourth or last quartile—Q4 starts at 39 and ends around 76.
- ✓ The part between the upper quartile and the lower quartile is known as the **Inter Quartile Range (IQR)** and helps in approximating 50% of the middle data.

Program Name: Creating a boxplot with survived

demo15.py

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

import warnings
warnings.filterwarnings('ignore')

df = sns.load_dataset('titanic')
sns.boxplot(x = 'sex', y = 'age', data = df, hue = "survived")

plt.show()
```

Output



- ✓ Other than the information about the age of the passengers, the above plot also shows the distribution of passengers who survived.
- ✓ The plot shows that most young males survived compared to females.

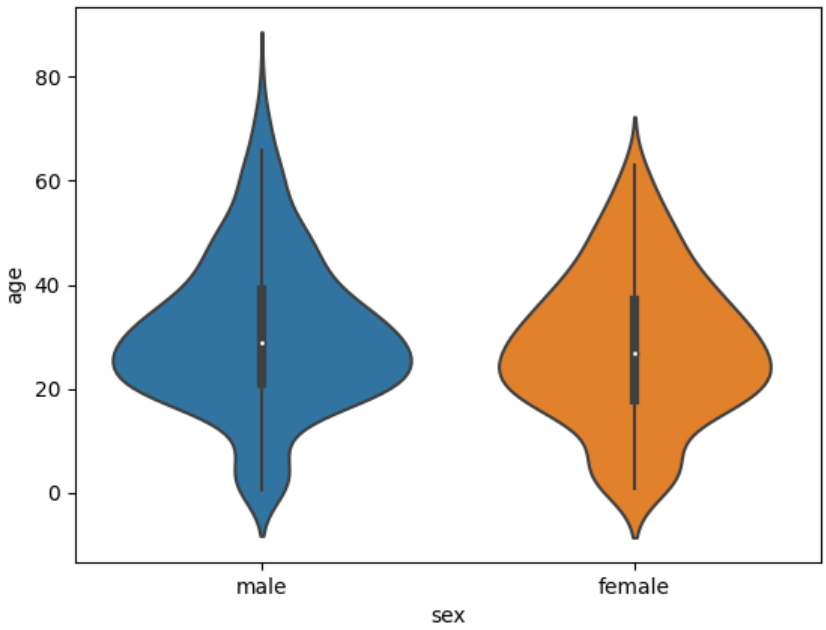
14. Violin Plot

✓ This type of plot is the same as the box plot, but with a violin plot, we can display all components corresponding to a data point.

Program Name: Creating violin plot
demo16.py

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

import warnings
warnings.filterwarnings('ignore')

df = sns.load_dataset('titanic')
sns.violinplot(x = 'sex', y = 'age', data = df)

plt.show()
```

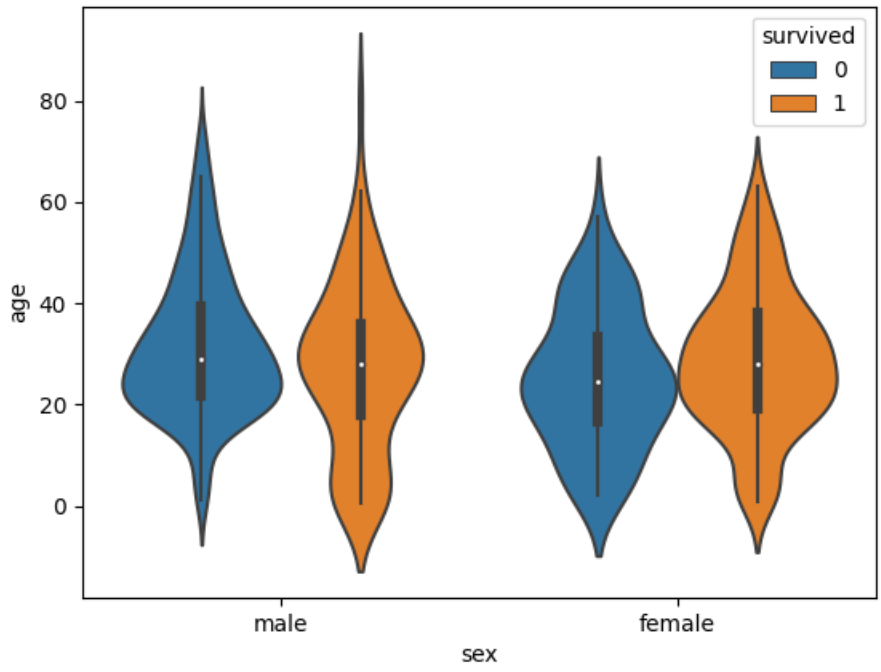
Output



Program Name: Creating violin plot with survived
demo17.py

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

import warnings
warnings.filterwarnings('ignore')

df = sns.load_dataset('titanic')
sns.violinplot(x = 'sex', y = 'age', data = df, hue = 'survived')

plt.show()
```

Output



15. Word Cloud

- ✓ A word cloud is a data visualization technique.
- ✓ This technique displays most used words in large font and the least used words in small font.
- ✓ It helps to get an idea about your text data,

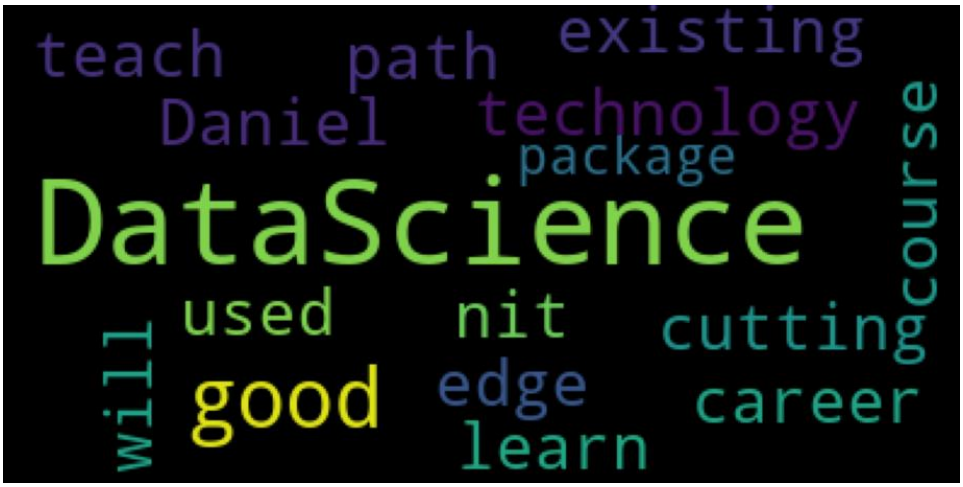
We need to install

```
pip install wordcloud
```

Program Name: Wordcloud example
demo18.py

```
import matplotlib.pyplot as plt
from wordcloud import WordCloud

text = "DataScience having good career path, DataScience is
cutting edge technology, DataScience course is existing in nit,
Daniel used to teach DataScience, if we learn DataScience then we
will get good package"

wc = WordCloud()
wc.generate(text)

plt.figure(figsize = (12, 12))
plt.imshow(wc)

plt.axis('off')
plt.show()
```

Output



16. Sunburst plot

- ✓ A sunburst plot is a very popular data visualization technique used to visualize hierarchical data.
- ✓ In every level of the hierarchy is represented by a ring or circle.
- ✓ Whereas the innermost circle or ring is the highest level of the hierarchy.

We need to install

```
pip install plotly
```

Program Name: Sunburst plot example
demo19.py

```
import plotly.express as px

data = px.data.tips()
figure = px.sunburst(data, path = ["day", "sex"], values =
"total_bill")

figure.show()
```

Output