Double-click (or enter) to edit

```python
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
data ={
    "Name": ["John", "Bob", "Charlie", "David"],
    "Age": [24, 19, 22,  25],
    "Score": [88,92,85,95]
}
df = pd.DataFrame(data)
print(df)
```

```
      Name  Age  Score
0     John   24     88
1      Bob   19     92
2  Charlie   22     85
3    David   25     95
```

```python
dfSortedByAge = df.sort_values(by='Age')
dfSortedByAge
```

	Name	Age	Score
1	Bob	19	92
2	Charlie	22	85

```python
dfSortedByAge = df.sort_values(by='Age')
dfSortedByAge
```

	Name	Age	Score
1	Bob	19	92
2	Charlie	22	85
0	John	24	88
3	David	25	95

```python
sortedByIndex = df.sort_index()
sortedByIndex
```

	Name	Age	Score
0	John	24	88
1	Bob	19	92
2	Charlie	22	85
3	David	25	95

```python
subset_iloc= df.iloc[1:3, 0:2]
print("subset using iloc:")
print(subset_iloc)
```

```python
subset_iloc= df.iloc[1:3, 0:2]
print("subset using iloc:")
print(subset_iloc)
```

```
subset using iloc:
      Name  Age
1      Bob   19
2  Charlie   22
```

```python
subset_loc = df.loc[df['Age'] > 20,['Name', 'Score']]
print("\nsubset using loc:")
print(subset_loc)
```

```
subset using loc:
      Name  Score
0     John     88
2  Charlie     85
3    David     95
```

```python
data = {
    'category': ['A','B','B','A','A'],
    'Value': [10,15,12,18,20]
}

# Create DataFrame
df = pd.DataFrame(data)

# Print the column names
```

```python
print("Columns:", df.columns.tolist())

# Filter example (Value > 15)
filtered = df[df['Value'] > 15]

print("\nFiltered Data:")
print(filtered)
```

```
Columns: ['category', 'Value']

Filtered Data:
  category  Value
3        A     18
4        A     20
```

```python
data = {
    'Name' : ['Alice', 'Bob', 'Charlie', 'David'],
    'Department' : ['HR', 'IT', 'Finance', 'HR'],
    'Talab' : [50000, 60000, 55000, 52000]
}
df = pd.DataFrame(data)
df
```

	Name	Department	Talab
0	Alice	HR	50000

1	Bob	IT	60000
2	Charlie	Finance	55000
3	David	HR	52000

```
data = {
    'category': ['A','B','B','A'],
    'Value': [10,15,12,18,]
}

df = pd.DataFrame(data)
print(df)
```

```
   category  Value
0         A     10
1         B     15
2         B     12
3         A     18
```

```
data = {
    'category': ['A','B','B','A'],
    'Value': [10,20,30,40,]
}

df = pd.DataFrame(data)
grouped = df.groupby('category')['Value'].mean()
print(grouped)
```

```
data = {
    'category': ['A','B','B','A'],
    'Value': [10,20,30,40,]
}

filter = df.groupby('category').filter(lambda x: x['Value'].sum() > 50)
print(filtered)
```

```
   category  Value
3         A     18
4         A     20
```
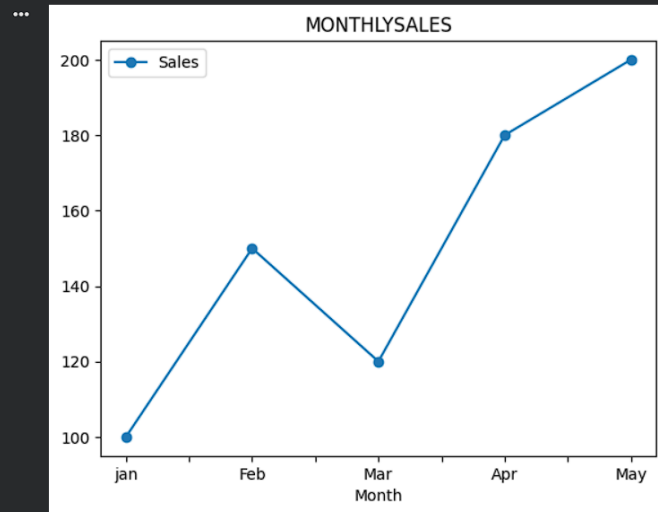
DATA visualization

+ Code   + Text

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

```
data = {'Month':['jan','Feb','Mar','Apr','May'],
        'Sales': [100,150,120,180,200]
}
df = pd.DataFrame(data)
df
```
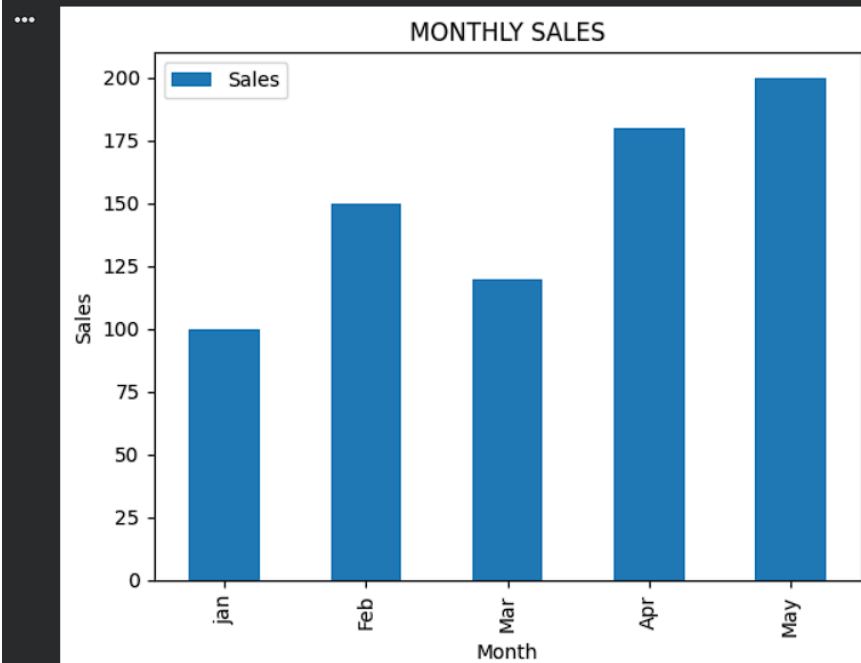
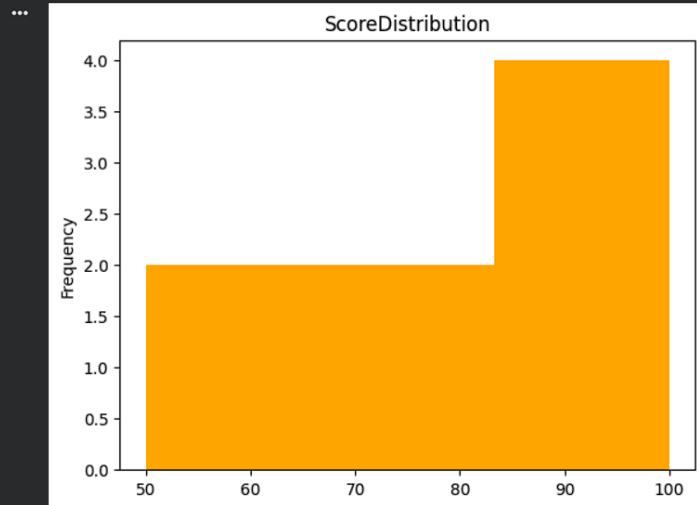|	Month	Sales
0	jan	100

```
df.plot(x = 'Month', y = 'Sales', kind = 'line', marker='o', title='MONTHLYSALES')
plt.show()
```



```
df.plot(x='Month', y='Sales', kind='bar', title='MONTHLY SALES')
plt.xlabel('Month')
plt.ylabel('Sales')
plt.show()
```

```python
data = {
    'Score' : [50, 60, 70, 80, 90, 96, 98, 100]
}
df = pd.DataFrame(data)
df['Score'].plot(kind='hist', bins=3, title='ScoreDistribution', color='orange')
plt.show()
```
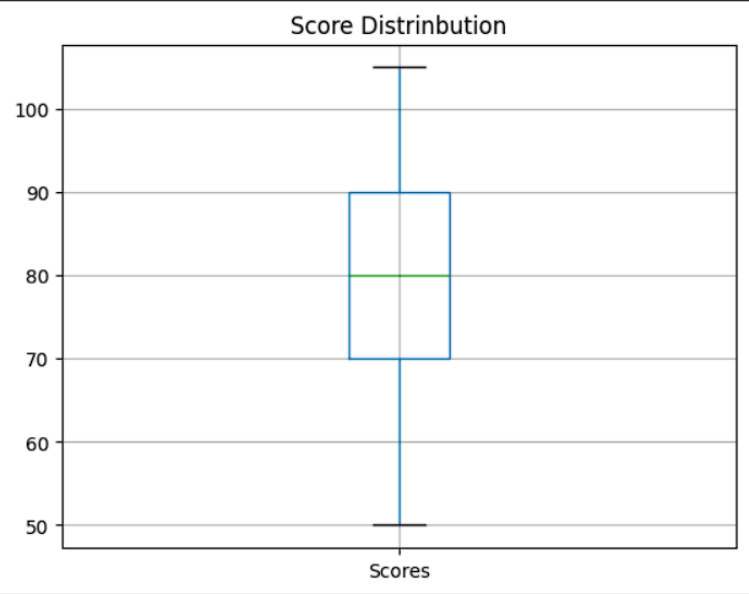


```python
df = pd.DataFrame(data)

df.plot(x='height', y='weight', kind='scatter', title='Height vs Weight')

plt.show()
```
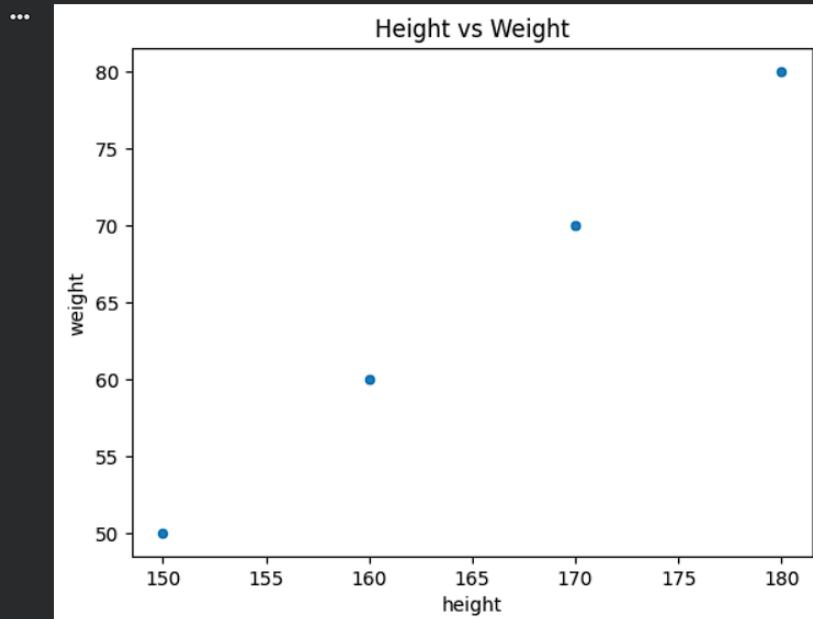
```
data = {'Scores': [50, 60, 70, 75, 80, 85, 90, 95, 105]}
df = pd.DataFrame(data)
df.boxplot(column='Scores')
plt.title('Score Distrinbution')
plt.show()
```


Score Distrinbution (boxplot of Scores)

3.1 Warm Up Exercises:

Sorting and Subsetting: Complete all following Task:

• Dataset for the Task: "titanic.csv"

Following task is common for all the problem:

Load the provided dataset and import in pandas DataFrame.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
tic = pd.read_csv("/content/drive/MyDrive/concept of Technology of AI/Copy of Titanic-Dataset.csv")
tic.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

Next steps: Generate code with tic | New interactive sheet

2.Check info of the DataFrame and identify following: Problem 1 - Sorting:

Create a DataFrame called fare that contains only the Fare column of the Titanic dataset. Print the head of the result.

```python
fare = tic[['Fare']]
fare.head()
```

	Fare
0	7.2500
1	71.2833
2	7.9250
3	53.1000
4	8.0500

Next steps: Generate code with `fare`   New interactive sheet

2.Create a DataFrame called class age that contains only the Pclass and Age columns of the Titanic dataset, in that order. Print the head of the result.

```python
classAge = tic[['Pclass','Age']]
classAge.head()
```

	Pclass	Age
0	3	22.0
1	1	38.0
2	3	26.0
3	1	35.0
4	3	35.0

Next steps: Generate code with `classAge`   New interactive sheet

3.Create a DataFrame called survived gender that contains the Survived and Sex columns of the Titanic dataset, in that order.Print the head of the result.

```python
SurvivedSex = tic.loc[tic["Survived"] == 1, "Sex"]
SurvivedSex.tail()
```

	Sex
875	female
879	female
880	female
887	female
889	male

Problem - 2 - Subsetting:

Complete all the following Task:

Subsetting Rows:

Filter the Titanic dataset for cases where the passenger's fare is greater than 100, assigning it to fare gt 100.View the printed result.

```
faregt100 = tic[tic['Fare'] > 100]
faregt100.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
27	28	0	1	Fortune, Mr. Charles Alexander	male	19.0	3	2	19950	263.0000	C23 C25 C27	S
31	32	1	1	Spencer, Mrs. William Augustus (Marie Eugenie)	female	NaN	1	0	PC 17569	146.5208	B78	C
88	89	1	1	Fortune, Miss. Mabel Helen	female	23.0	3	2	19950	263.0000	C23 C25 C27	S
118	119	0	1	Baxter, Mr. Quigg Edmond	male	24.0	0	1	PC 17558	247.5208	B58 B60	C
195	196	1	1	Lurette, Miss. Elise	female	58.0	0	0	PC 17569	146.5208	B80	C

Next steps: (Generate code with `faregt100`) (New interactive sheet)

2.Filter the Titanic dataset for cases where the passenger's class (Pclass) is 1, assigning it to first class. View the printed result.

```
tic1 = tic[tic["Pclass"] == 1]
tic1
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S
11	12	1	1	Bonnell, Miss. Elizabeth	female	58.0	0	0	113783	26.5500	C103	S
23	24	1	1	Sloper, Mr. William Thompson	male	28.0	0	0	113788	35.5000	A6	S
...	...	...	...	...	...	...	...	...	...	...	...	...
871	872	1	1	Beckwith, Mrs. Richard Leonard (Sallie Monypeny)	female	47.0	1	1	11751	52.5542	D35	S
872	873	0	1	Carlsson, Mr. Frans Olof	male	33.0	0	0	695	5.0000	B51 B53 B55	S
879	880	1	1	Potter, Mrs. Thomas Jr (Lily Alexenia Wilson)	female	56.0	0	1	11767	83.1583	C50	C
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C

216 rows × 12 columns

Next steps: (Generate code with `tic1`) (New interactive sheet)

3.Filter the Titanic dataset for cases where the passenger's age is less than 18 and the passenger is female (Sex is "female"), assigning it to female under 18. View the printed result.

3.Filter the Titanic dataset for cases where the passenger's age is less than 18 and the passenger is female (Sex is "female"), assigning it to female under 18. View the printed result.

```python
fem_18 = tic[(tic["Age"] < 18) & (tic["Sex"] == "female")]
fem_18.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708	NaN	C
10	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	PP 9549	16.7000	G6	S
14	15	0	3	Vestrom, Miss. Hulda Amanda Adolfina	female	14.0	0	0	350406	7.8542	NaN	S
22	23	1	3	McGowan, Miss. Anna "Annie"	female	15.0	0	0	330923	8.0292	NaN	Q
24	25	0	3	Palsson, Miss. Torborg Danira	female	8.0	3	1	349909	21.0750	NaN	S

Next steps: Generate code with `fem_18`   New interactive sheet

Subsetting Rows by Categorical variables:

1.Filter the Titanic dataset for passengers whose Embarked port is either "C" (Cherbourg) or "S" (Southampton), assigning the result to embarked c or s. View the printed result.

```python
tic_emb = tic[(tic["Embarked"] == "C") | (tic["Embarked"] == "S")]
tic_emb.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

Next steps: Generate code with `tic_emb`   New interactive sheet

2.Filter the Titanic dataset for passengers whose Pclass is in the list [1, 2] (indicating first or second class), assigning the result to first second class.View the printed result.

```python
tic_pc = tic[(tic["Pclass"] == 1) | (tic["Pclass"] == 2)]
tic_pc.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708	NaN	C

3.2 Exploratory Data Analysis Practice Exercise - 1.

Warning: Handle missing values in the Age column by filling them with the median age of the dataset before performing the division.)

Answer the following questions from Dataset: Which passenger had the highest fare paid relative to their age? To answer the question perform following operations:

1.Add a column to the Titanic dataset, fare per year, containing the fare divided by the age of the passenger(i.e., Fare/Age).

```python
median_age = tic['Age'].median()
tic['Age'] = tic['Age'].fillna(median_age)
tic['fare_per_year']= tic['Fare']/tic['Age']
print(tic.head())
```

```
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3

                                                Name     Sex   Age  SibSp  \
0                            Braund, Mr. Owen Harris    male  22.0      1
1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0      1
2                             Heikkinen, Miss. Laina  female  26.0      0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                           Allen, Mr. William Henry    male  35.0      0

   Parch            Ticket     Fare Cabin Embarked  fare_per_year
0      0         A/5 21171   7.2500   NaN        S       0.329545
1      0          PC 17599  71.2833   C85        C       1.875876
2      0  STON/O2. 3101282   7.9250   NaN        S       0.304808
```

2.Subset rows where fare per year is higher than 5, assigning this to high fare age.

```
tic_high = tic[tic["fare_per_year"] > 5]
tic_high.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	fare_per_year
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750	NaN	S	10.537500
16	17	0	3	Rice, Master. Eugene	male	2.0	4	1	382652	29.1250	NaN	Q	14.562500
27	28	0	1	Fortune, Mr. Charles Alexander	male	19.0	3	2	19950	263.0000	C23 C25 C27	S	13.842105
31	32	1	1	Spencer, Mrs. William Augustus (Marie Eugenie)	female	28.0	1	0	PC 17569	146.5208	B78	C	5.232886
43	44	1	2	Laroche, Miss. Simonne Marie Anne Andree	female	3.0	1	2	SC/Paris 2123	41.5792	NaN	C	13.859733

Next steps: Generate code with `tic_high`   New interactive sheet

3.Sort high fare age by descending fare per year, assigning this to high fare age srt.

```
tic_des = tic_high.sort_values(by="fare_per_year", ascending=False)
tic_des.head()
```

4.Select only the Name and fare per year columns of high fare age srt and save the result as result. 5.Look at the result.

```
tic1 = tic_high[["Name","fare_per_year"]]
tic1.head()
```

	Name	fare_per_year
7	Palsson, Master. Gosta Leonard	10.537500
16	Rice, Master. Eugene	14.562500
27	Fortune, Mr. Charles Alexander	13.842105
31	Spencer, Mrs. William Augustus (Marie Eugenie)	5.232886
43	Laroche, Miss. Simonne Marie Anne Andree	13.859733

Next steps: Generate code with `tic1`   New interactive sheet

Which adult male passenger (age ≥ 18 and Sex is 'male') paid the highest fare relative to their class? To answer the question perform following operations:

1.Add a column to the Titanic dataset, fare per class, containing the fare divided by the passenger class i.e. Fare / Pclass.

```
tic["fare_per_class"] = tic["Fare"]/tic["Pclass"]
tic.head()
```

2.Subset rows where fare per year is higher than 5, assigning this to high fare age.

```
high_fare_age = tic[tic["fare_per_class"] > 5]
high_fare_age.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	fare_per_year	fare_per_class
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C	1.875876	71.2833
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S	1.517143	53.1000
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S	0.960417	51.8625
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750	NaN	S	10.537500	7.0250
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708	NaN	C	2.147914	15.0354

Next steps: Generate code with `high_fare_age`   New interactive sheet

3.

Sort high fare age by descending fare per year, assigning this to high fare_age_srt.

```
high_fare_age_start = high_fare_age.sort_values(by="fare_per_class", ascending=False)
high_fare_age_start.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	fare_per_year	fare_per_class
737	738	1	1	Lesurer, Mr. Gustave J	male	35.0	0	0	PC 17755	512.3292	B101	C	14.637977	512.3292

3.

Sort high fare age by descending fare per year, assigning this to high fare_age_srt.

```python
high_fare_age_start = high_fare_age.sort_values(by="fare_per_class", ascending=False)
high_fare_age_start.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	fare_per_year	fare_per_class
737	738	1	1	Lesurer, Mr. Gustave J	male	35.0	0	0	PC 17755	512.3292	B101	C	14.637977	512.3292
679	680	1	1	Cardeza, Mr. Thomas Drake Martinez	male	36.0	0	1	PC 17755	512.3292	B51 B53 B55	C	14.231367	512.3292
258	259	1	1	Ward, Miss. Anna	female	35.0	0	0	PC 17755	512.3292	NaN	C	14.637977	512.3292
341	342	1	1	Fortune, Miss. Alice Elizabeth	female	24.0	3	2	19950	263.0000	C23 C25 C27	S	10.958333	263.0000
88	89	1	1	Fortune, Miss. Mabel Helen	female	23.0	3	2	19950	263.0000	C23 C25 C27	S	11.434783	263.0000

Next steps: (Generate code with `high_fare_age_start`) (New interactive sheet)

4.Select only the Name and fare per year columns of high fare age srt and save the result as result.

```python
tic_1 = high_fare_age_start[["Name","fare_per_year"]]
tic_1.head()
```

	Name	fare_per_year
737	Lesurer, Mr. Gustave J	14.637977

4.Select only the Name and fare per year columns of high fare age srt and save the result as result.

```python
tic_1 = high_fare_age_start[["Name","fare_per_year"]]
tic_1.head()
```

	Name	fare_per_year
737	Lesurer, Mr. Gustave J	14.637977
679	Cardeza, Mr. Thomas Drake Martinez	14.231367
258	Ward, Miss. Anna	14.637977
341	Fortune, Miss. Alice Elizabeth	10.958333
88	Fortune, Miss. Mabel Helen	11.434783

Next steps: (Generate code with `tic_1`) (New interactive sheet)

3.3 Exploratory Data Analysis with Group-by Method Practice Exercise: Based on the dataset Answer the following question:

What percent of the total fare revenue came from each passenger class? To answer the question perform following operation:

1.Calculate the total Fare paid across all passengers in the Titanic dataset.

```python
total_fare = tic["Fare"].sum()
print(total_fare)
```
28693.9493

```python
total_fare = tic["Fare"].sum()
print(total_fare)
```
28693.9493

2.Subset for passengers in first class (Pclass is 1) and calculate their total fare.

```python
p1_fare = tic[tic["Pclass"] == 1]
p1_fare_sum = p1_fare["Fare"].sum()
p1_fare_sum
```
np.float64(18177.4125)

3.Do the same for second class (Pclass is 2) and third class (Pclass is 3)

```python
p2_fare = tic[tic["Pclass"] == 2]
p2_fare_sum = p2_fare["Fare"].sum()
p2_fare_sum
```
np.float64(3801.8417)

```python
p3_fare = tic[tic["Pclass"] == 3]
p3_fare_sum = p3_fare["Fare"].sum()
p3_fare_sum
```
np.float64(6714.6951)

4.Combine the fare totals from first, second, and third classes into a list.

```python
p_comb = [
    tic[tic["Pclass"] == 1]["Fare"].sum(),
    tic[tic["Pclass"] == 2]["Fare"].sum(),
    tic[tic["Pclass"] == 3]["Fare"].sum()
]

p_comb
```

```
[np.float64(18177.4125), np.float64(3801.8417), np.float64(6714.6951)]
```

5.Divide the totals for each class by the overall total fare to get the proportion of fare revenue by class.

```python
fare_totals = [tic[tic["Pclass"] == c]["Fare"].sum() for c in [1, 2, 3]]

overall_total = sum(fare_totals)

fare_proportions = [x / overall_total for x in fare_totals]

fare_proportions
```

```
[np.float64(0.6334928771899656),
 np.float64(0.1324962855496507),
 np.float64(0.23401083726038366)]
```

Based on the dataset Answer the following question: What percent of the total number of passengers on the Titanic belonged to each age group (e.g., child, adult, senior)?

Based on the dataset Answer the following question: What percent of the total number of passengers on the Titanic belonged to each age group (e.g., child, adult, senior)?

To answer the question perform following operation:

1.Create a new column, age group, that categorizes passengers into "child" (age < 18), "adult" (age 18{64), and "senior" (age 65 and above).

```python
def categorize_age(age):
    if age < 18:
        return "child"
    elif age < 65:
        return "adult"
    else:
        return "senior"

tic["age_group"] = tic["Age"].apply(categorize_age)
tic.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	fare_per_year	fare_per_class	age_group
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S	0.329545	2.416667	adult
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C	1.875876	71.283300	adult
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S	0.304808	2.641667	adult
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S	1.517143	53.100000	adult

2.Calculate the total number of passengers on the Titanic.

```
total = len(tic)
total

891
```

3.Count the number of passengers in each age group.

```
group_counts = tic["age_group"].value_counts()
group_counts
```

	count
age_group	
adult	767
child	113
senior	11

dtype: int64

4.Divide the count of each age group by the total number of passengers to get the proportion of passengers in each age group. 5.Display the proportion as a percentage.

```
tic = pd.read_csv("/content/drive/MyDrive/concept of Technology of AI/Copy of Titanic-Dataset.csv")
group_counts = tic['Survived'].value_counts()
total_passengers = len(tic)
group_percentage = (group_counts / total_passengers) * 100
print(group_percentage)

Survived
0    61.616162
1    38.383838
Name: count, dtype: float64
```