INTRO TO PANDAS

LECTURE # 3

PANDAS

Library for data manipulation and analysis.

pip install pandas

import pandas as pd

PANDAS

For displaying the Data

```
[3]  data.head()
```

```
[5]  data1=data[0:50]
```

```
[ ]  data2=data[len(data)-3:]
     data2
```

PANDAS

Copy dataframes

1

<New Data Frame name> = OLD Data Frame name>

Independent Copies

2

<New Data Frame name> = OLD Data Frame name> .copy()

Dependent Copies

Unit_Cost	Total_Revenue	Total_Cost	Profit
90.93	228779.10	135031.05	93748.05
56.67	471336.91	326815.89	144521.02
502.54	3586605.09	2697132.18	889472.91

```
[ ]  # create a copy of the dataset
     df=data.copy()
     df2=df
     del df2['Profit']
     df
```

Unit_Cost	Total_Revenue	Total_Cost
159.42	2533654.00	1582243.50
117.11	576782.80	328376.44
524.96	1158502.59	933903.84

SET INDEX

```
df.set_index('Order_ID',inplace=False)
```

[22] ✓ 0.1s

Order_ID	Region	Country	Item_Type
669165933	Australia and Oceania	Tuvalu	Baby Food
963881480	Central America and the Caribbean	Grenada	Cereal
341417157	Europe	Russia	Office Supplies
514321792	Sub-Saharan Africa	Sao Tome and Principe	Fruits

```
df
```

[24] ✓ 0.0s

	Region	Country	Item_Type	S
0	Australia and Oceania	Tuvalu	Baby Food	
1	Central America and the Caribbean	Grenada	Cereal	
2	Europe	Russia	Office Supplies	
3	Sub-Saharan Africa	Sao Tome and Principe	Fruits	
4	Sub-Saharan Africa	Rwanda	Office Supplies	

SET INDEX

```
[15]  df.set_index('Order_ID',inplace=True)
      ✓ 0.0s
```

```
[17]  df2
      ✓ 0.0s
```

Order_ID	Region	Country	Item_Type
669165933	Australia and Oceania	Tuvalu	Baby Food
963881480	Central America and the Caribbean	Grenada	Cereal
341417157	Europe	Russia	Office Supplies
514321792	Sub-Saharan Africa	Sao Tome and Principe	Fruits
115456712	Sub-Saharan Africa	Rwanda	Office Supplies

LOC

<data frame name>.LOC[<index>]

```
df.loc[810711038]
```

✓ 0.0s

```
Region                Asia
Country            Malaysia
Item_Type             Fruits
Sales_Channel        Offline
Order_Priority             L
Order_Date        11/11/2011
Ship_Date         12/28/2011
Units_Sold              6267
Unit_Price              9.33
Unit_Cost               6.92
Total_Revenue       58471.11
Total_Cost          43367.64
Name: 810711038, dtype: object
```

<data frame name>.LOC[[<index>, <index>]]

```
df.loc[[514321792,810711038]]
```

✓ 0.0s

Order_ID	Region	Country	Item_Type	Sales_Channel	Order_P
514321792	Sub-Saharan Africa	Sao Tome and Principe	Fruits	Online	
810711038	Asia	Malaysia	Fruits	Offline	

LOC

<data frame name>.LOC[
[<index>, <index...>] , [col1, col2...]]

```
df.loc[[514321792,810711038],['Region','Country']]
```

✓ 0.0s

Order_ID	Region	Country
514321792	Sub-Saharan Africa	Sao Tome and Principe
810711038	Asia	Malaysia

LOC VS ILOC

drink	carbonated?	temperature	sugar(tsp)	calories
soda	True	cold	10.5	150
coffee	False	hot	3.0	31
smoothie	False	cold	6.0	85
water	False	cold	0.0	0
tea	False	hot	2.0	43
lemonade	False	cold	9.5	125

LOC VS ILOC

```
# loc function
df.loc[ ['tea'] ]
```

Program Output

drink	carbonated?	temperature	sugar(tsp)	calories
tea	False	hot	2.0	43

Indexing Using a Single Label

```
# iloc function
df.iloc[ [1] ]
```

Program Output

drink	carbonated?	temperature	sugar(tsp)	calories
coffee	False	hot	3.0	31

Indexing Using a Single Integer

ILOC

<data frame name>.iLOC[
[<index>, <index...>] , [col1, col2...]]

```
data.iloc[0:10,2:5]
```

	Item_Type	Sales_Channel	Order_Priority
0	Baby Food	Offline	H
1	Cereal	Online	C
2	Office Supplies	Offline	L
3	Fruits	Online	C
4	Office Supplies	Offline	L
5	Baby Food	Online	C
6	Household	Offline	M
7	Vegetables	Online	H
8	Personal Care	Offline	M
9	Cereal	Online	H

GROUBY

<data frame name>.groupby
(<col name to map values on>)
[col name on which aggregation needs to be done].
sum()

GROUBY

```
region_units_sold = data.groupby('Region')['Units_Sold'].sum()
print(region_units_sold)
```

```
5]  ✓ 0.0s
```

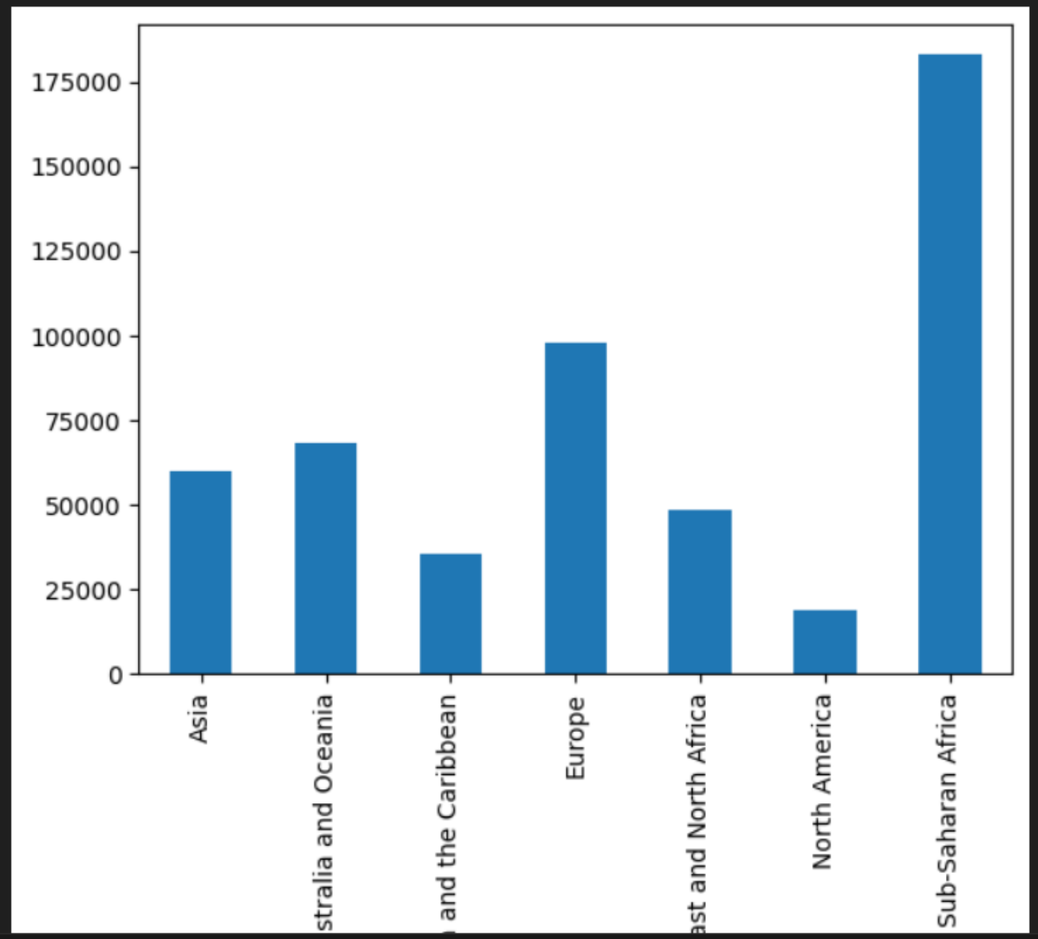
```
Region
Asia                                    59967
Australia and Oceania                   68325
Central America and the Caribbean       35771
Europe                                  98117
Middle East and North Africa            48678
North America                           19143
Sub-Saharan Africa                     182870
Name: Units_Sold, dtype: int64
```

PLOTTING USING GROUBY

```
region_units_sold.plot(kind='bar')
```

✓ 1.9s

```
<AxesSubplot: xlabel='Region'>
```



- 'bar' or 'barh' for bar plots
- 'hist' for histogram
- 'box' for boxplot
- 'kde' or 'density' for density plots
- 'area' for area plots
- 'scatter' for scatter plots
- 'hexbin' for hexagonal bin plots
- 'pie' for pie plots

GROUBY

```
grouped_data = data.groupby('Region')[['Total_Revenue', 'Total_Cost']].sum()
print(grouped_data)
```

✓ 0.0s

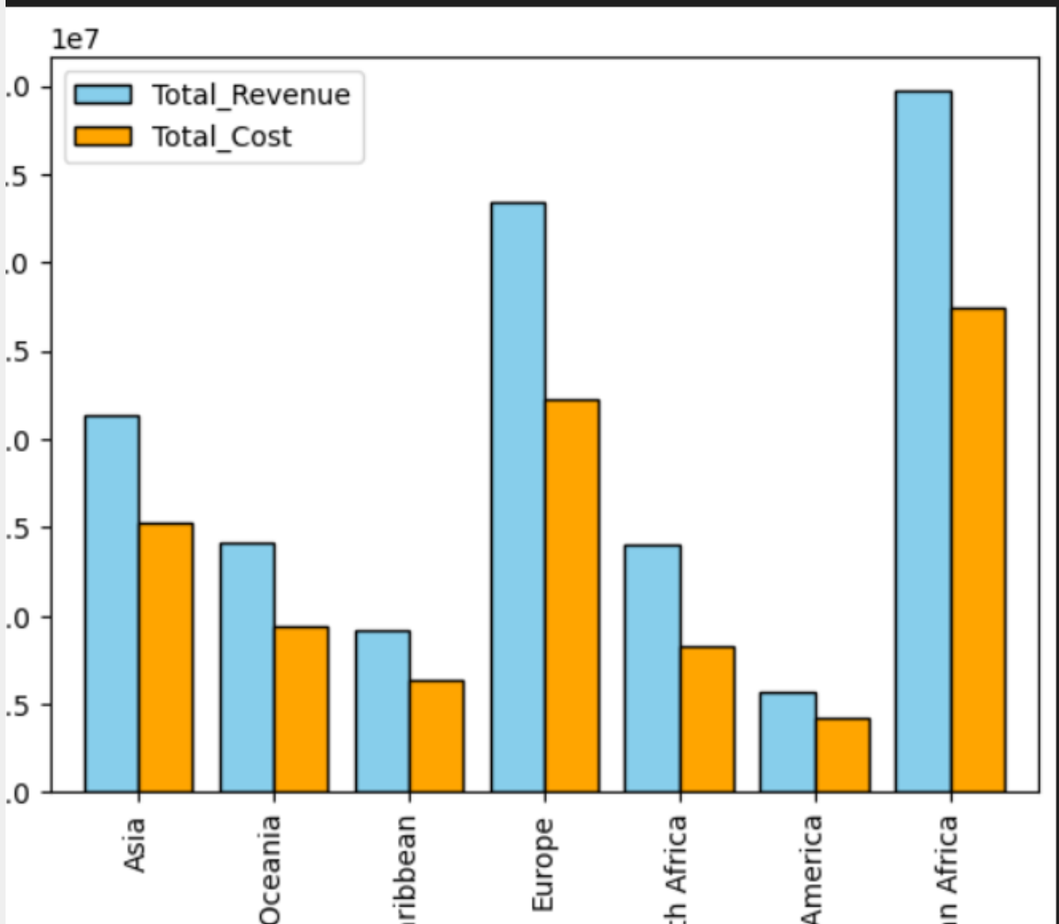
Region	Total_Revenue	Total_Cost
Asia	21347091.02	15233245.15
Australia and Oceania	14094265.13	9372105.10
Central America and the Caribbean	9170385.49	6323477.64
Europe	33368932.11	22285993.48
Middle East and North Africa	14052706.58	8291514.72
North America	5643356.55	4185413.79
Sub-Saharan Africa	39672031.43	27488820.03

GROUBY

```
grouped_data = data.groupby('Region')[['Total_Revenue', 'Total_Cost']].sum()

ax = grouped_data.plot(kind='bar', color=['skyblue', 'orange'],edgecolor='black', width=0
```

0.6s



GROUBY

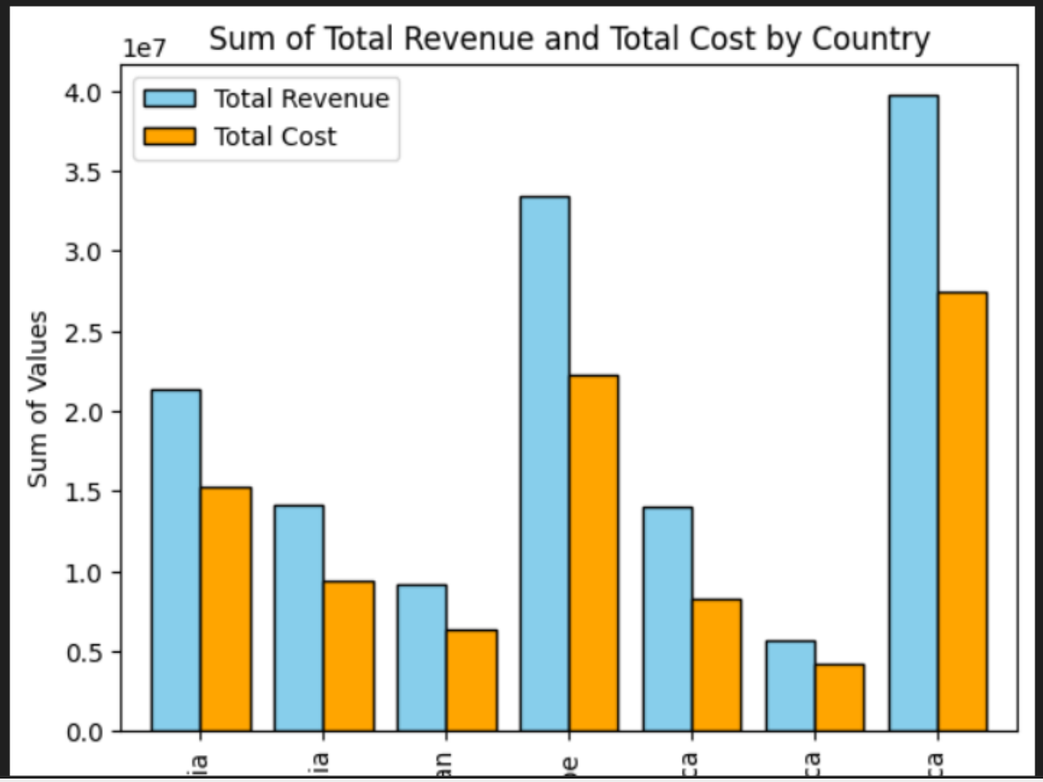
```
grouped_data = data.groupby('Region')[['Total_Revenue', 'Total_Cost']].sum()

ax = grouped_data.plot(kind='bar', color=['skyblue', 'orange'],edgecolor='black', width=0.8)

ax.set_xlabel('Country')
ax.set_ylabel('Sum of Values')
ax.set_title('Sum of Total Revenue and Total Cost by Country')
ax.legend(["Total Revenue", "Total Cost"])
```

✓ 0.7s

```
<matplotlib.legend.Legend at 0x1cae48db9a0>
```



TO FIND A MIN/MAX VALUE

```
movie_min_budget=dm.groupby("title_year")["gross"].sum()
print(movie_min_budget.idxmin())
```

[57] ✓ 0.0s

```
1916.0
```

```
movie_min_budget=dm.groupby("title_year")["gross"].sum()
print(movie_min_budget.idxmax())
```

✓ 0.0s

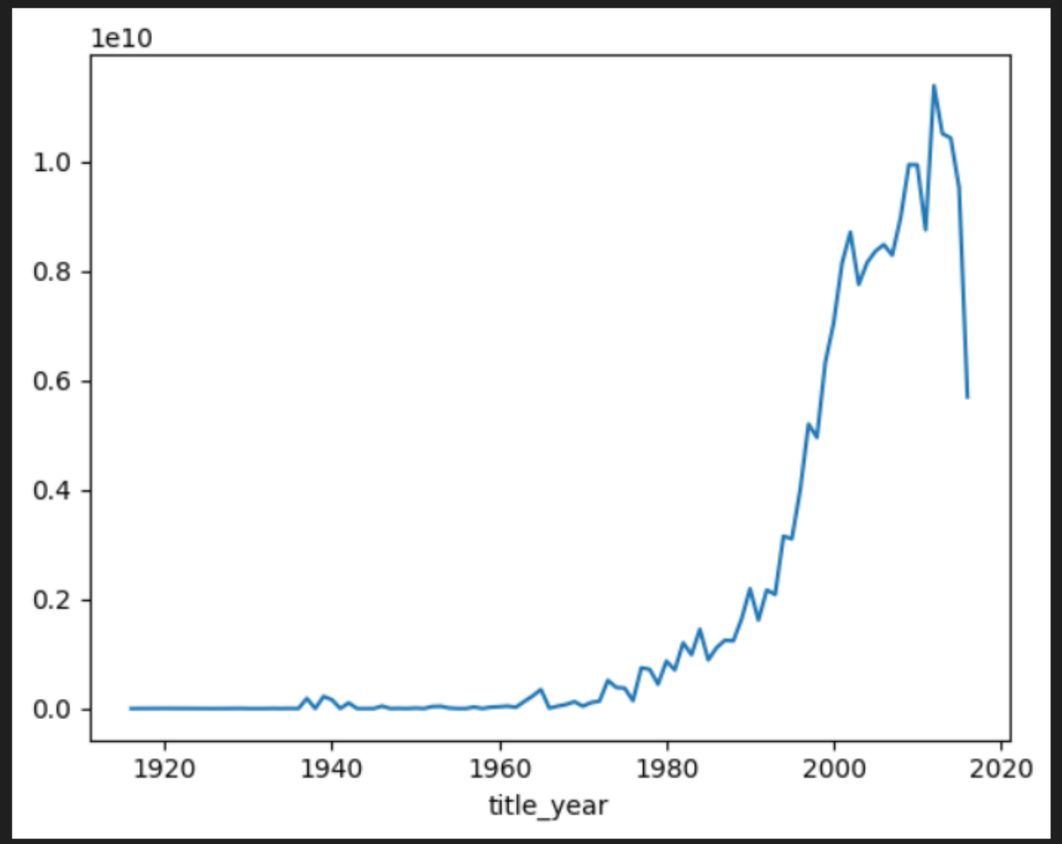
```
2012.0
```

TO FIND A MIN/MAX VALUE

```
moviebudget=dm.groupby("title_year")["gross"].sum()
moviebudget.plot()
```

✓ 0.5s

```
<AxesSubplot: xlabel='title_year'>
```



MAX VS IDXMAX

```
movie_max_budget=dm.groupby("title_year")["gross"].sum().max()
movie_max_budget
```

[66] ✓ 0.0s

```
11380108510.0
```

```
movie_max_budget=dm.groupby("title_year")["gross"].sum().idxmax()
movie_max_budget
```

[67] ✓ 0.0s

```
2012.0
```

1.Write a Pandas program to create a dataframe from a dictionary and display it.
Sample data: {'X':[78,85,96,80,86], 'Y':[84,94,89,83,86],'Z':[86,97,96,72,83]}

```
Expected Output:
    X   Y   Z
0  78  84  86
1  85  94  97
2  96  89  96
3  80  83  72
4  86  86  83
```

Write a Pandas program to get the first 3 rows of a given DataFrame.

WRITE A PANDAS PROGRAM TO SELECT THE 'X' AND 'Y' COLUMNS FROM THE FOLLOWING DATAFRAME.

RENAME X, Y, Z AS QUIZ1, QUIZ 2 AND QUIZ 3 RESPECTIVELY

MAKE QUIZ 1 AS THE INDEX

WRITE A PANDAS PROGRAM TO COUNT THE NUMBER OF ROWS AND COLUMNS OF A DATAFRAME.

for more practice questions refer to the below given links

https://www.w3resource.com/python-exercises/pandas/index-dataframe.php

https://python4csip.com/files/download/Worksheet%20-%20Pandas.pdf

https://discovery.cs.illinois.edu/guides/DataFrame-Fundamentals/dataframe-loc-vs-iloc/