In [ ]:
```
import pandas as pd
```

In [ ]:
```
import numpy as  np
```

In [ ]:
```
pd.set_option('display.min_rows',50)
```

In [ ]:
```
df = pd.read_csv("E:\python project\Pandas_Project_Session\data\global_superstore\global_superstore_2016.csv",enco
```

In [149…
```
df.head()
```

Out[149...

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Postal Code	City	State	Country
0	40098	CA-2014-AB10015140-41954	2014-11-11	2014-11-13	First Class	AB-100151402	Aaron Bergman	Consumer	73120.0	Oklahoma City	Oklahoma	United States
1	26341	IN-2014-JR162107-41675	2014-02-05	NaT	Second Class	JR-162107	Justin Ritter	Corporate	NaN	Wollongong	New South Wales	Australia
2	25330	IN-2014-CR127307-41929	NaT	2014-10-18	First Class	CR-127307	Craig Reiter	Consumer	NaN	Brisbane	Queensland	Australia
3	13524	ES-2014-KM1637548-41667	NaT	2014-01-30	First Class	KM-1637548	Katherine Murray	Home Office	NaN	Berlin	Berlin	Germany
4	47221	SG-2014-RH9495111-41948	2014-11-05	NaT	Same Day	RH-9495111	Rick Hansen	Consumer	NaN	Dakar	Dakar	Senegal

In [ ]:
```
df.info()
```

In [ ]:
```
df.describe()
```

convert OrderDate and ShipDate to datetime object

In [ ]:
```
df['Order Date'] = pd.to_datetime(df['Order Date'],errors='coerce')
```

In [ ]:
```
df['Ship Date'] = pd.to_datetime(df['Ship Date'],errors='coerce')
```

In [ ]:
```
df.dtypes
```

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

In [ ]:
```
df[['Order Date','Ship Date']]
```

In [ ]:
```
df
```

In [ ]:
```
df['Shipping Cost'].astype(int)
```

check whether all the values of columns 'Sales' & 'Profit' contains $ symbol or not

In [ ]:
```
if df['Sales'].str.contains('$').all():
    print("Yes:$ is present")
else:
    print("No: $ is not present")
```

In [ ]:
```
if df['Profit'].str.contains('$').all():
    print("Yes:$ IS present")
else:
    print("No: $ is not present")
```

In [ ]:
```
df.dtypes
```

clean the columns 'Sales' & 'Profit' whcih consists of '$', '(', & ')' symbols

In [ ]:
```
df['Sales'] = df['Sales'].str.replace('\$',' ',regex = True)
df['Sales'][:10]
```

In [ ]:
```
df['Sales'] = df['Sales'].str.replace(',','',regex = True)
df['Sales'][:10]
```

In [ ]:
```
df['Profit'] = df['Profit'].str.replace('\$',' ',regex = True)
```

In [ ]:
```
df['Profit'][:10]
```

In [ ]:
```
df['Profit'] = df['Profit'].str.replace('\$',' ',regex = True)
```

In [ ]:
```
df['Profit'] = df['Profit'].str.replace('\)',' ',regex = True)
```

In [ ]:
```
df['Profit'] = df['Profit'].str.replace('\(', '',regex = True)
```

In [ ]:
```
df['Profit']
```

In [ ]:
```
df.info()
```

In [ ]:
```
df
```

Let's begin the analysis with the Project Questions

1. Total how many orders have cross the shipping cost of 500?

In [ ]:
```
shipping_cost = (df['Shipping Cost']>500)
shipping_cost[:5]
```

In [150…
```
total_orders = shipping_cost.sum()
total_orders
```

Out[150… np.int64(120)

2. Count the number of segments, countries, regions, markets, categories, and sub-categories present in the global_superstore_2016 data.

In [ ]:
```
df.Segment.value_counts()
```

In [ ]:
```
df['Country'].value_counts()
```

In [ ]:
```
len(df['Country'].value_counts())
```

In [ ]:
```
df['Region'].value_counts()
```

In [ ]:
```
df['Market'].value_counts()
```

In [ ]:
```
df['Category'].value_counts()
```

In [ ]:
```
df['Sub-Category'].value_counts()
```

In [ ]:
```
df.columns
```

3. Get the list of Order ID's where the Indian customer's have bought the things under the category 'Technology' after paying the Shipping Cost more than 500.

In [ ]:
```
list_indian_customers = (df['Country']== 'India') & (df['Category']=='Technology') & (df['Shipping Cost'] >= 500)
list_indian_customers
```

In [ ]:
```
order_id = df[list_indian_customers]
order_id
```

In [ ]:
```
df.dtypes
```

In [ ]:
```
df['Sales'] = pd.to_numeric(df['Sales'],errors = 'coerce')
```

In [ ]:
```
df['Sales'] = df['Sales'].round().astype(int)
df.dtypes
```

In [ ]:
```
df['Shipping Cost'] = df['Shipping Cost'].round().astype(int)
```

In [ ]:
```
df['Profit'] = pd.to_numeric(df['Profit'],errors = 'coerce')
```

In [ ]:
```
df['Profit'] = df['Profit'].fillna(0)
```

In [ ]:
```
df['Profit'] = df['Profit'].round().astype(int)
```

In [ ]:
```
df.dtypes
```

4. Get the list of Order ID's where the Indian customer's have bought the things under the category 'Technology' where the Sales is greater than 500.

In [ ]:
```
bought = (df['Country']=='India') & (df['Category']=='Technology') & (df['Sales'] > 500)
```

5.How many people from the State 'Karnataka' have bought the things under the category 'Technology'.

In [ ]:
```
c = df[(df['State'] == 'Karnataka')& (df['Category']=='Technology')]
pd.set_option('display.max_columns', None)
c
```

6. Get the list of countries where the 'Profit' and 'Shipping Cost's are greater than or equal to 2000 and 300 respectively.¶

In [ ]:
```
profit_greater = (df['Profit'] >= 2000) & (df['Shipping Cost'] >= 300)
cond1 = df[profit_greater]
```

In [ ]:
```
cond1.head(5)
```

7. Find the list of Indian states where the people have purchased the things under the category Technology.

In [ ]:
```
cond_states_tech = (df['Country']=='India') & (df['Category']=='Technology')
```

In [ ]:
```
Indian_States = df[cond_states_tech]
```

In [ ]:
```
Indian_States['State'].drop_duplicates().tolist()
```

8. Find the overall rank of "India" where the 'Profit' is maximum under the category 'Technology'.

In [ ]:
```
pro_max =  (df['Profit'].max()) & (df['Category']=='Technology')
```

In [ ]:
```
maxx = df[pro_max]
```

In [ ]:
```
maxx
```

In [ ]:
```
global_countries = maxx.sort_values(by = ['Profit'],ascending=False)
global_countries.head(5)
```

In [ ]:
```
ranked_data = global_countries['Profit'].rank(method='min', ascending=False)
ranked_data
```

In [ ]:
```
global_countries['Country']
```

In [ ]:
```
Top_profit_countries = global_countries['Country'].drop_duplicates().tolist()
```

In [ ]:
```

```

In [ ]:
```
Top_profit_countries
```

In [ ]:
```
ranked_countries = global_countries['Country'].drop_duplicates()
```

In [ ]:
```
ranked_series = pd.concat([ranked_countries,ranked_data], axis=1)
```

In [ ]:
```
ranked_series
```

9. Display the data with min, max, average and std of 'Profit' & 'Sales' for each Sub-Category under each Category

In [ ]:
```
grouped_data = df.groupby(by=['Country', 'Category', 'Sub-Category'], as_index=True)
grouped_data
```

In [ ]:
```
functions = [('min_value', 'min'), ('max_value', 'max'), ('mean_value', 'mean'), ('std_value', 'std')]
```

In [ ]:
```
aggregate_data = grouped_data[['Sales','Profit']].agg(functions)
```

In [152…
```
aggregate_data
```

Out[152...

			Sales						
			min_value	max_value	mean_value	std_value	min_value	max_value	mean_value
Country	Category	Sub-Category							
Afghanistan	Furniture	Bookcases	732	2070	1401.000000	946.108873	102	849	475.500000
		Chairs	417	914	665.500000	351.432070	46	357	201.500000
		Furnishings	85	220	135.333333	51.348483	1	80	33.000000
		Tables	267	4626	2113.666667	2254.452114	35	648	415.666667
	Office Supplies	Appliances	669	669	669.000000	NaN	281	281	281.000000
		Art	30	178	111.000000	69.942834	8	76	33.250000
		Binders	34	252	97.666667	82.918434	1	53	21.666667
		Envelopes	79	231	171.333333	81.094595	9	86	39.666667
		Fasteners	14	45	29.500000	21.920310	2	2	2.000000
		Labels	13	64	39.333333	25.540817	1	22	13.666667
		Paper	15	123	77.250000	45.154365	3	43	14.750000
		Storage	62	333	190.500000	121.766169	21	71	46.750000
		Supplies	69	244	154.666667	87.557600	19	104	56.666667
	Technology	Accessories	115	1471	409.400000	593.736726	4	515	112.800000
		Copiers	427	712	569.500000	201.525433	4	7	5.500000
		Machines	346	346	346.000000	NaN	14	14	14.000000
		Phones	413	1168	732.500000	345.771119	25	444	237.250000

			Sales						
			min_value	max_value	mean_value	std_value	min_value	max_value	mean_value
Country	Category	Sub-Category							
Albania	Furniture	Bookcases	364	414	389.000000	35.355339	40	203	121.500000
		Furnishings	58	58	58.000000	NaN	3	3	3.000000
	Office Supplies	Art	16	174	74.333333	86.731386	2	24	10.333333
		Binders	6	28	17.000000	15.556349	0	2	1.000000
		Labels	11	11	11.000000	NaN	2	2	2.000000
		Storage	19	219	114.666667	100.281271	9	55	35.666667
	Technology	Machines	85	1619	852.000000	1084.701802	1	259	130.000000
		Phones	182	554	368.000000	263.043723	22	40	31.000000
...	...	...	...	...	...	...	...	...	...
Zambia	Office Supplies	Labels	9	55	24.833333	20.970614	3	20	7.166667
		Paper	18	110	57.400000	41.385988	2	40	16.800000
		Storage	9	1023	217.125000	241.927227	3	184	53.000000
		Supplies	18	84	57.500000	31.596413	3	40	21.750000
	Technology	Accessories	41	246	110.750000	91.928867	6	31	17.500000
		Copiers	122	1181	415.571429	379.685333	0	261	68.285714
		Machines	598	1247	937.333333	325.515489	75	263	196.333333
		Phones	89	1272	426.800000	501.335915	4	407	135.800000

			Sales						
			min_value	max_value	mean_value	std_value	min_value	max_value	mean_value
Country	Category	Sub-Category							
Zimbabwe	Furniture	Bookcases	51	86	68.500000	24.748737	109	181	145.000000
		Chairs	50	627	338.500000	408.000613	0	56	28.000000
		Furnishings	12	13	12.500000	0.707107	20	22	21.000000
		Tables	143	143	143.000000	NaN	276	276	276.000000
	Office Supplies	Appliances	29	161	115.333333	74.808645	50	284	156.333333
		Art	4	112	27.055556	28.814768	5	133	42.055556
		Binders	1	40	7.166667	11.019267	1	89	13.583333
		Envelopes	6	43	21.200000	14.515509	10	83	35.200000
		Fasteners	4	8	5.333333	2.309401	5	10	8.333333
		Labels	3	14	6.000000	5.354126	3	18	7.500000
		Paper	7	32	21.000000	11.633286	16	74	37.250000
		Storage	3	162	43.333333	52.127248	7	125	51.666667
		Supplies	11	67	38.666667	24.889087	12	56	37.833333
	Technology	Accessories	78	78	78.000000	NaN	60	60	60.000000
		Copiers	154	445	299.500000	205.768073	257	371	314.000000
		Machines	15	104	53.250000	42.295587	20	174	77.000000
		Phones	22	102	62.000000	56.568542	48	170	109.000000

2009 rows × 8 columns

In [ ]: