bi-test-task-3

January 7, 2024

[1]:
```
import pandas as pd
```

[ ]:
```
#IMPORT DATASET
```

[3]:
```
df = pd.read_excel('data.xlsx', sheet_name = 'BA_task_3')
```

[3]:

	Month	Product	Revenue	Profit
0	2020-01-12	PayAlto	84021.66	909.11
1	2020-02-12	PayAlto	165429.59	5395.30
2	2020-03-12	PayAlto	291906.14	12144.56
3	2020-04-12	PayAlto	357797.28	5604.51
4	2020-05-12	PayAlto	375421.56	6400.34

[ ]:
```
#DATA INFO
```

[4]:
```
df.shape
```

[4]:
```
(119, 4)
```

[5]:
```
df.head()
```

[5]:

	Month	Product	Revenue	Profit
0	2020-01-12	PayAlto	84021.66	909.11
1	2020-02-12	PayAlto	165429.59	5395.30
2	2020-03-12	PayAlto	291906.14	12144.56
3	2020-04-12	PayAlto	357797.28	5604.51
4	2020-05-12	PayAlto	375421.56	6400.34

[6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 119 entries, 0 to 118
Data columns (total 4 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   Month    119 non-null    datetime64[ns]
 1   Product  119 non-null    object
 2   Revenue  119 non-null    float64
```

```
 3   Profit   119 non-null    float64
dtypes: datetime64[ns](1), float64(2), object(1)
memory usage: 3.8+ KB
```

[7]:
```
df.dtypes
```

[7]:
```
Month      datetime64[ns]
Product            object
Revenue           float64
Profit            float64
dtype: object
```

[ ]:
```
#CHECK MISSING VALUES
```

[8]:
```
df.isna().sum()
```

[8]:
```
Month      0
Product    0
Revenue    0
Profit     0
dtype: int64
```

There are no missing values in the dataset

[ ]:
```
#CHECK DUPLICATED VALUES
```

[9]:
```
df.duplicated().sum()
```

[9]:
```
0
```

There are no dulicated values in the dataset

[ ]:
```
#QUESTION 1: Calculate annual Revenue and Profit for each product.
```

[13]:
```
annual_summary = df.groupby('Product').agg({'Revenue' : 'sum','Profit': 'sum'}).
↪reset_index()
annual_summary
```

[13]:

	Product	Revenue	Profit
0	Mobiamo	3.268056e+08	6426827.26
1	PayAlto	3.694835e+08	7762974.27

[ ]:
```
#QUESTION 2: The Project Managers (PM) for PayAlto and Mobiamo are John Doe and␣
↪Kelly Nguyen respectively. From the result of question 1, show in table the␣
↪PM for each product. You need to create a new project manager-product table.
```

[17]:
```
#SHOW PM FOR EACH PRODUCT
```

```
annual_summary['project_manager'] = annual_summary['Product'].map({'Mobiamo':␣
↪'John Doe','PayAlto': 'Kelly Nguyen'})

annual_summary = annual_summary[['Product', 'project_manager', 'Revenue',␣
↪'Profit']]

annual_summary
```

[17]:

	Product	project_manager	Revenue	Profit
0	Mobiamo	John Doe	3.268056e+08	6426827.26
1	PayAlto	Kelly Nguyen	3.694835e+08	7762974.27

[21]:
```
#CREATE NEW PM-PRODUCT TABLE
project_manager_table = annual_summary[['project_manager','Product']]

project_manager_table
```

[21]:

	project_manager	Product
0	John Doe	Mobiamo
1	Kelly Nguyen	PayAlto

[ ]:
```
#QUESTION 3: From the result of question 2, please transform the table to long␣
↪format.
```

[26]:
```
#MELT PROFIT AND REVENUE TO ONE COLUMN
long_format_summary = pd.melt(annual_summary, id_vars=['Product',␣
↪'project_manager'], var_name='metric', value_name='amount').sort_values(by =␣
↪'Product')

long_format_summary
```

[26]:

	Product	project_manager	metric	amount
0	Mobiamo	John Doe	Revenue	3.268056e+08
2	Mobiamo	John Doe	Profit	6.426827e+06
1	PayAlto	Kelly Nguyen	Revenue	3.694835e+08
3	PayAlto	Kelly Nguyen	Profit	7.762974e+06