

```
In [17]: import pandas as pd
import numpy as np

In [4]: df = pd.read_excel('fbdata.xlsx')

In [5]: df.head()

Out[5]:
```

	ad_id	xyz_campaign_id	fb_campaign_id	age	gender	interest	Impressions	Clicks	Spent	Total_Conversion	Approved_Conversion
0	708746	916	103916	30-34	M	15	7350	1	1.43	2	1
1	708749	916	103917	30-34	M	16	17861	2	1.82	2	0
2	708771	916	103920	30-34	M	20	693	0	0.00	1	0
3	708815	916	103928	30-34	M	28	4259	1	1.25	1	0
4	708818	916	103928	30-34	M	28	4133	1	1.29	1	1

```
In [6]: df.shape
Out[6]: (1143, 11)
In [7]: df.columns

Out[7]: Index(['ad_id', 'xyz_campaign_id', 'fb_campaign_id', 'age', 'gender',
            'interest', 'Impressions', 'Clicks', 'Spent', 'Total_Conversion',
            'Approved_Conversion'],
            dtype='object')

In [8]: df[['Impressions', 'Clicks', 'Spent',
            'Total_Conversion', 'Approved_Conversion']].describe()

Out[8]:
```

	Impressions	Clicks	Spent	Total_Conversion	Approved_Conversion
count	11430000e+03	1143.000000	1143.000000	8943.000000	1143.000000
mean	1.867321e+05	33.390201	51.360656	2.855643	0.944007
std	3.127622e+05	56.892438	86.908418	4.483593	1.737708
min	8.700000e+01	0.000000	0.000000	0.000000	0.000000
25%	6.503500e+03	1.000000	1.480000	1.000000	0.000000
50%	5.150900e+04	8.000000	12.370000	1.000000	1.000000
75%	2.217690e+05	37.500000	60.025000	3.000000	1.000000
max	3.052003e+06	421.000000	639.949998	60.000000	21.000000

```
In [9]: print("Number of Unique Ads :", df['ad_id'].nunique())
print("Number of Campaigns :", df['xyz_campaign_id'].nunique())
print("Number of Facebook Campaigns :", df['fb_campaign_id'].nunique())
print("Number of Interest Groups :", df['interest'].nunique())
print("Number of Age Groups :", df['age'].nunique())

Number of Unique Ads : 1143
Number of Campaigns : 3
Number of Facebook Campaigns : 691
Number of Interest Groups : 40
Number of Age Groups : 4

In [11]: df = df.drop(['ad_id', 'fb_campaign_id'],axis = 1)

In [15]: df.shape
Out[15]: (1143, 9)

In [16]: df.isnull().sum()

Out[16]:
```

	xyz_campaign_id	age	gender	interest	Impressions	Clicks	Spent	Total_Conversion	Approved_Conversion	dtype
	0	0	0	0	0	0	0	0	0	int64

```
In [19]: df['xyz_campaign_id'].value_counts()

Out[19]:
```

	xyz_campaign_id	count
1178	625	625
936	464	464
916	54	54

```
Name: xyz_campaign_id, dtype: int64

In [20]: df['xyz_campaign_id'] = df['xyz_campaign_id'].replace((1178,936,916),('campaign C','campaign B','campaign A'))
df['xyz_campaign_id'].value_counts()

Out[20]:
```

	campaign	count
campaign C	625	625
campaign B	464	464
campaign A	54	54

```
Name: xyz_campaign_id, dtype: int64

In [28]: df[['xyz_campaign_id', 'Approved_Conversion', 'Spent', 'Impressions']].groupby(['xyz_campaign_id'],
as_index = False).agg('mean').rename(columns = {'xyz_campaign_id': 'campaign', 'Approved_Conversion': 'Average_Conversion',
'spent': 'Average_Expense',
'Impressions': 'Average_Visibility'})

Out[28]:
```

	campaign	Approved_Conversion	Spent	Average_Visibility
0	campaign A	0.444444	2.772407	8943.055556
1	campaign B	0.394397	6.235711	17517.644397
2	campaign C	1.395200	89.059440	327717.945600

```
In [26]: df[['xyz_campaign_id', 'Approved_Conversion', 'Spent', 'Impressions']].groupby(['xyz_campaign_id'],
as_index = False).agg('mean').rename(columns = {'xyz_campaign_id': 'Campaign',
Approved_Conversion': 'Average_Conversion',
'Spent': 'Average_Expense',
'Impressions': 'Average_Visibility'})

Out[26]:
```

	Campaign	Average_Conversion	Average_Expense	Average_Visibility
0	campaign A	0.444444	2.772407	8943.055556
1	campaign B	0.394397	6.235711	17517.644397
2	campaign C	1.395200	89.059440	327717.945600

```
In [29]: df['ClickThroughRate'] = ((df['Clicks']/df['Impressions'])*100)
df['CostPerClick'] = (df['Spent']/df['Clicks'])
df['ConversionValue'] = (df['Approved_Conversion']/df['Total_Conversion'])
df['CostPerConversion'] = (df['Spent']/df['Approved_Conversion'])

In [31]: df['ConversionValue'] = df['Approved_Conversion']*100
df['ROAS'] = round(df['ConversionValue']/df['Spent'], 2)
df['CostPerMille'] = round((df['Spent']/df['Impressions'])*1000, 2)

In [33]: df = df.replace([np.inf, -np.inf], np.nan).dropna(axis=0)
print("Shape of the Data After Removing Nans and Infs :", df.shape)

Shape of the Data After Removing Nans and Infs : (513, 16)

In [80]: df[['xyz_campaign_id', 'ClickThroughRate', 'CostPerClick', 'ConversionRate',
'ConversionValue', 'CostPerMille', 'ROAS']].groupby(['xyz_campaign_id'],
as_index = False).agg('mean').rename(columns = {'xyz_campaign_id': 'Campaign',
ClickThroughRate': 'Average_CTR',
'CostPerClick': 'Average_Cost/Click',
'ConversionRate': 'Average_ConversionRate'})

-----
KeyError                                Traceback (most recent call last)
<ipython-input-80-f9de30fa281> in <module>
----> 1 df[['xyz_campaign_id', 'ClickThroughRate', 'CostPerClick', 'ConversionRate',
2      'ConversionValue', 'CostPerMille', 'ROAS']].groupby(['xyz_campaign_id'],
3      as_index = False).agg('mean').rename(columns = {'xyz_campaign_id': 'Campaign',
4      ClickThroughRate': 'Average_CTR',
5      'CostPerClick': 'Average_Cost/Click',
C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\frame.py in __getitem__(self, key)
2986         if is_iterator(key):
2987             key = list(key)
-> 2988         indexer = self._iindex_with_key(indexer, key, axis=1, raise_missing=True)[1]
2989
2990         # take() does not accept boolean indexers
C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\indexing.py in _iindex_with_key(self, key, axis, raise_missing)
1252         keyarr, indexer, new_indexer = ax._reindex_non_unique(keyarr)
-> 1254         self._validate_read_indexer(keyarr, indexer, axis, raise_missing=raise_missing)
1255         return keyarr, indexer
C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\indexing.py in _validate_read_indexer(self, key, indexer, axis, raise_missing)
1302         if raise_missing:
1303             not_found = list(set(key) - set(ax))
-> 1304             raise KeyError(f"{not_found} not in index")
1305
1306         # we skip the warning on Categorical
KeyError: "[ 'ConversionRate'] not in index"

In [37]: import seaborn as sns
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')

In [50]: plt.rcParams['figure.figsize'] = (14,4)

plt.subplot(1,2,1)
sns.barplot(df['gender'],df['ROAS'],
hue=df['xyz_campaign_id'],
palette='cool')
plt.xlabel(' ')

plt.subplot(1,2,2)
sns.barplot(df['age'],df['ROAS'],
hue = df['xyz_campaign_id'],palette = 'cool')
plt.xlabel(' ')

plt.suptitle('Impact of Gender and Age on ROAS', fontsize = 20)
plt.show()

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be 'data', and passing other arguments without an explicit keyword will result in an error or misinterpretation.
warnings.warn(
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be 'data', and passing other arguments without an explicit keyword will result in an error or misinterpretation.
warnings.warn(

Impact of Gender and Age on ROAS
```

```
In [52]: plt.rcParams['figure.figsize'] = (28,3)
sns.barplot(df['interest'],df['ROAS'])
plt.title('Impact of age groups on ROAS', fontsize = 20)
plt.xlabel('\n Different Interest Groups on ROAS', fontsize = 20)
plt.show()

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be 'data', and passing other arguments without an explicit keyword will result in an error or misinterpretation.
warnings.warn(

Impact of age groups on ROAS
```

```
In [55]: df[['interest', 'ROAS']].sort_values(by = 'ROAS',
ascending = False).head(5).style.background_gradient(cmap='cool')

Out[55]:
```

	interest	ROAS
267	63	595.860000
215	19	204.080000
244	16	175.440000
318	29	138.890000
484	28	125.790000

```
In [58]: plt.rcParams['figure.figsize'] = (8,4)
sns.boxplot(df['xyz_campaign_id'],df['CostPerMille'],palette = 'winter')
plt.title('Campaign vs Brand Awareness')
plt.xlabel(' ')
plt.show()

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be 'data', and passing other arguments without an explicit keyword will result in an error or misinterpretation.
warnings.warn(

Campaign vs Brand Awareness
```

```
In [61]: plt.rcParams['figure.figsize'] = (12,4)
sns.barplot(df['age'],df['CostPerMille'],
hue = df['gender'],palette='cool')
plt.xlabel('Age Groups')
plt.title('Analyzing the Groups to focus on Brand Awareness')
plt.show()

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be 'data', and passing other arguments without an explicit keyword will result in an error or misinterpretation.
warnings.warn(

Analyzing the Groups to focus on Brand Awareness
```

```
In [67]: plt.rcParams['figure.figsize'] = (20,4)

plt.subplot(1,2,1)
sns.barplot(df['age'],df['Clicks'],
hue = df['gender'],palette = 'cool')
plt.xlabel('Age Group')

plt.subplot(1,2,2)
sns.barplot(df['age'],df['Impressions'],
hue=df['gender'],palette = 'cool')
plt.xlabel('Age Groups')

plt.suptitle('Analysing the Clicks And Impression')
plt.show()

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be 'data', and passing other arguments without an explicit keyword will result in an error or misinterpretation.
warnings.warn(
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be 'data', and passing other arguments without an explicit keyword will result in an error or misinterpretation.
warnings.warn(

Analysing the Clicks And Impression
```

```
In [68]: plt.rcParams['figure.figsize'] = (15,4)
sns.scatterplot(df['Impressions'],df['Clicks'],color = 'red')
plt.title('Relationship between Clicks and Impression',fontsize = 20)
plt.show()

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be 'data', and passing other arguments without an explicit keyword will result in an error or misinterpretation.
warnings.warn(

Relationship between Clicks and Impression
```

```
In [75]: def cat(x):
    if x <= "100":
        return '0-100'
    elif x <= "200":
        return '100-200'
    elif x <= "300":
        return '200-300'
    else:
        return '300+'

df['Clicks'] = df['Clicks'].apply(cat)

df[['Impressions', 'Clicks']].groupby('Clicks').agg(['mean']).style.background_gradient(cmap = 'Wistia')

Out[75]:
```

	Impressions	Clicks
mean	103307.390777	100
0-100	800683.319444	100-200
100-200	1203176.360000	200-300
200-300	2229958.250000	300+

```
In [74]: plt.rcParams['figure.figsize'] = (17, 6)
sns.heatmap(df.corr(),
annot = True, linewidths = 2.0,
cmap = 'summer')
plt.title('Correlation Heatmap for the KPIs',n', fontsize = 20)
plt.show()

Correlation Heatmap for the KPIs
```

	interest	Impressions	Spent	Total_Conversion	Approved_Conversion	ClickThroughRate	CostPerClick	ConversionRate	CostPerConversion	ConversionValue	ROAS	CostPerMille
interest	1	0.037	0.0066	0.094	0.017	-0.19	-0.11	-0.16	0.014	0.017	-0.057	-0.22
Impressions	0.037	1	0.97	0.81	0.69	-0.15	0.2	-0.54	0.55	0.69	-0.3	-0.09
Spent	0.0066	0.97	1	0.73	0.6	-0.047	0.14	-0.53	0.65	0.6	-0.31	0.011
Total_Conversion	0.094	0.81	0.73	1	0.88	-0.18	0.17	-0.49	0.18	0.88	-0.21	-0.13
Approved_Conversion	0.017	0.69	0.6	0.88	1	-0.14	0.16	-0.21	-0.038	1	-0.13	-0.11
ClickThroughRate	-0.19	-0.15	-0.047	-0.18	-0.14	1	-0.41	0.27	0.015	-0.14	0.14	0.93
CostPerClick	-0.11	0.2	-0.14	0.17	0.16	-0.41	1	-0.22	0.1	0.16	-0.5	0.078
ConversionRate	-0.16	-0.54	-0.53	-0.49	-0.21	0.27	-0.22	1	-0.56	-0.21	0.39	0.21
CostPerConversion	0.014	0.55	0.65	0.18	-0.038	0.015	0.1	-0.56	1	-0.038	-0.32	0.071
ConversionValue	0.017	0.69	0.6	0.88	1	-0.14	0.16	-0.21	-0.038	1	-0.13	-0.11
ROAS	-0.057	-0.3	-0.31	-0.21	-0.13	0.14	-0.5	0.39	-0.32	-0.13	1	-0.048
CostPerMille	-0.22	-0.09	0.011	-0.13	-0.11	0.93	-0.078	0.21	0.071	-0.11	-0.048	1

```
In [ ]:
```