

```
In [1]: #loading in a dataset and basic python packages
#this dataset is stored in jupyter, but there are ways to upload from
import pandas as pd
import numpy as np
import statistics as st
import matplotlib.pyplot as plt
df = pd.read_excel('StockX Click Dataset.xlsx')
df
```

Out [1]:

	Campaign Name	Objective	Platform	Media Co
0	C3_Dynamic-Remarketing_FBIG_US_CO_CatalogSales...	CatalogSales	FBIG	107906.7000
1	C3_Dynamic-Remarketing_FBIG_US_CO_CatalogSales...	CatalogSales	FBIG	102058.1599
2	C3_DABA_FBIG_US_CO_CatalogSales_AlwaysOn_Socia...	CatalogSales	FBIG	101864.2999
3	C3_Dynamic-Remarketing_FBIG_US_CO_CatalogSales...	CatalogSales	FBIG	89141.4499
4	C3_Dynamic-Remarketing_FBIG_US_CO_CatalogSales...	CatalogSales	FBIG	81975.4600
...	...	...	...	
647	C3_FR-SiteVisitors_FBIG_FR_CO_Traffic_Evergree...	Traffic	FBIG	0.0000
648	C3_Dynamic-Remarketing_FBIG_US_CO_CatalogSales...	CatalogSales	FBIG	N.
649	C3_DABA_FBIG_US_CO_CatalogSales_AlwaysOn_Socia...	CatalogSales	FBIG	N.
650	C3_BrandTraffic_TikTok_US_AW_Traffic_AlwaysOn_...	Traffic	TikTok	N.
651	C3_InfluencersBoost_IG_US_AW_Awareness_Other_S...	Awareness	IG	N.

652 rows × 6 columns

```
In [2]: #getting basic statistical meausre for the columns
df.describe()
```

Out [2]:

	Media Cost	Link Clicks	Month
count	648.000000	6.520000e+02	652.000000
mean	11586.033630	7.001707e+04	5.915644
std	14617.400784	2.006539e+05	3.274526
min	0.000000	0.000000e+00	1.000000
25%	3173.188267	8.257500e+02	3.000000
50%	7809.258551	7.518000e+03	6.000000
75%	14246.742272	2.817225e+04	9.000000
max	107906.700007	2.320789e+06	12.000000

In [3]: *#filtering a column by a number criteria*

```
cost = 50
df = df[df['Media Cost'] >= cost]
```

In [4]: *#creating a calculated field based upon two other columns*

```
df['CPCs'] = df['Media Cost']/df['Link Clicks']
df['CPCs'] = df['CPCs'].round(2)
df.head(5)
```

<ipython-input-4-205b3025d2ce>:2: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy) (https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy)

```
df['CPCs'] = df['Media Cost']/df['Link Clicks']
```

<ipython-input-4-205b3025d2ce>:3: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy) (https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy)

```
df['CPCs'] = df['CPCs'].round(2)
```

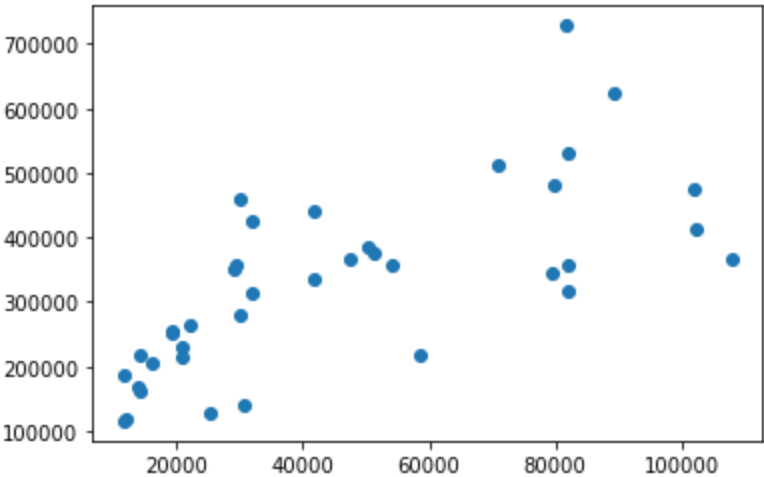
Out [4]:

	Campaign Name	Objective	Platform	Media Cost
0	C3_Dynamic-Remarketing_FBIG_US_CO_CatalogSales...	CatalogSales	FBIG	107906.700007
1	C3_Dynamic-Remarketing_FBIG_US_CO_CatalogSales...	CatalogSales	FBIG	102058.159996
2	C3_DABA_FBIG_US_CO_CatalogSales_AlwaysOn_Socia...	CatalogSales	FBIG	101864.299983
3	C3_Dynamic-Remarketing_FBIG_US_CO_CatalogSales...	CatalogSales	FBIG	89141.449996
4	C3_Dynamic-Remarketing_FBIG_US_CO_CatalogSales...	CatalogSales	FBIG	81975.460000

In [5]: *#filtering a column based upon a non numeric value*

```
obj = 'CatalogSales'
df = df[df['Objective']== obj]
df = df.sort_values(by='CPCs', ascending=False)
```

```
In [6]: #basic scatter plot
plt.scatter(df['Media Cost'],df['Link Clicks'])
plt.show()
```



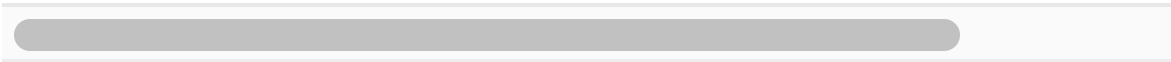
```
In [7]: #creating a new dataframe off of the original based upon filtering cr
df = pd.read_excel('StockX Click Dataset.xlsx')

fbig_platform = 'FBIG'
df_FBIG = df[df['Platform'] == fbig_platform]
df_FBIG
```

Out [7]:

	Campaign Name	Objective	Platform	Media Co
0	C3_Dynamic-Remarketing_FBIG_US_CO_CatalogSales...	CatalogSales	FBIG	107906.7000
1	C3_Dynamic-Remarketing_FBIG_US_CO_CatalogSales...	CatalogSales	FBIG	102058.1599
2	C3_DABA_FBIG_US_CO_CatalogSales_AlwaysOn_Socia...	CatalogSales	FBIG	101864.2999
3	C3_Dynamic-Remarketing_FBIG_US_CO_CatalogSales...	CatalogSales	FBIG	89141.4499
4	C3_Dynamic-Remarketing_FBIG_US_CO_CatalogSales...	CatalogSales	FBIG	81975.4600
...	...	...	...	...
644	C3_UK-SiteVisitors_FBIG_UK_CO_Traffic_Evergree...	Traffic	FBIG	0.0000
645	C3_XpressShipTraffic_FBIG_US_Brand_Traffic_Pro...	Traffic	FBIG	0.0000
647	C3_FR-SiteVisitors_FBIG_FR_CO_Traffic_Evergree...	Traffic	FBIG	0.0000
648	C3_Dynamic-Remarketing_FBIG_US_CO_CatalogSales...	CatalogSales	FBIG	N.
649	C3_DABA_FBIG_US_CO_CatalogSales_AlwaysOn_Socia...	CatalogSales	FBIG	N.

110 rows × 6 columns



```
In [8]: #dropping rows withs NaN values
df = df.dropna()
```

```
In [9]: #filtering using two number criteria
cost_low = 1000
cost_high = 100000

criteria = (df['Media Cost']>cost_low) & (df['Media Cost']<cost_high)
df = df[~criteria]
```

```
In [10]: #dropping a certain column
df = df.drop(columns='Month')
df
```

Out[10]:

	Campaign Name	Objective	Platform	Media Co
0	C3_Dynamic-Remarketing_FBIG_US_CO_CatalogSales...	CatalogSales	FBIG	107906.7000
1	C3_Dynamic-Remarketing_FBIG_US_CO_CatalogSales...	CatalogSales	FBIG	102058.1599
2	C3_DABA_FBIG_US_CO_CatalogSales_AlwaysOn_Socia...	CatalogSales	FBIG	101864.2999
571	C3_Apparel_Google_US_Nonbrand_CPA_Search_Searc...	CPA	Google	983.8898
572	C3_REM-BrandProduct_Google_US_Brand_CPA_Search...	CPA	Google	969.3009
...	...	...	...	
643	C3_BTSTraffic_FBIG_US_Brand_Traffic_AlwaysOn_S...	Traffic	FBIG	0.0000
644	C3_UK-SiteVisitors_FBIG_UK_CO_Traffic_Evergree...	Traffic	FBIG	0.0000
645	C3_XpressShipTraffic_FBIG_US_Brand_Traffic_Pro...	Traffic	FBIG	0.0000
646	C3_SparkAds_TikTok_US_AW_Reach_Promo_Social_St...	Reach	TikTok	0.0000
647	C3_FR-SiteVisitors_FBIG_FR_CO_Traffic_Evergree...	Traffic	FBIG	0.0000

80 rows × 5 columns

```
In [11]: #filtering a dataframe based upon number and text criteria
cost_low = 1000
Platform = 'FBIG'

criteria = (df['Media Cost']>cost_low)|(df['Platform']<Platform)
df = df[~criteria]
df.head(5)
```

Out[11]:

	Campaign Name	Objective	Platform	Media Cost	L Cli
571	C3_Apparel_Google_US_Nonbrand_CPA_Search_Searc...	CPA	Google	983.889830	1
572	C3_REM-BrandProduct_Google_US_Brand_CPA_Search...	CPA	Google	969.300962	1
577	C3_BrandProduct_Google_US_Brand_CPA_Search_Sea...	CPA	Google	907.093376	6
578	C3_REM-Accessories_Google_US_Nonbrand_CPA_Sear...	CPA	Google	858.991805	
579	C3_BrandProduct_Google_US_Brand_CPA_Search_Sea...	CPA	Google	835.820000	7

```
In [12]: click_low = 100

df = df[df['Link Clicks']>click_low]

platform = 'FBIG'

df = df[df['Platform']!=platform]
```

```
In [13]: df = df.drop_duplicates()
```

```
In [14]: #filtering using two text criteria
mask = df['Campaign Name'].str.contains('C3|c3')

# Use the mask with loc to select rows based on the condition
df = df.loc[mask]
```

```
In [15]: median = df['Link Clicks'].median()
median
```

Out[15]: 906.5

```
In [16]: #manipulating columns
df['Link Clicks'] = df['Link Clicks']+1
```

```
In [17]: import pandas as pd
import numpy as np
import statistics as st
import matplotlib.pyplot as plt
df = pd.read_excel('StockX Click Dataset.xlsx')
df = df.dropna()
```

```
In [18]: mask= df['Objective'].str.contains('Traffic|Awareness')
dff = df.loc[mask]
```

```
In [19]: dff
```

Out[19]:

	Campaign Name	Objective	Platform	Media Cost
15	C3_BroadCTV_TTD_US_Brand_Awareness_Promo_CTV_S...	Awareness	TTD	59995.948195
57	C3_SpringTraffic_FBIG_US_Brand_Traffic_AlwaysO...	Traffic	FBIG	26600.299995
72	C3_SpringTraffic_FBIG_US_Brand_Traffic_AlwaysO...	Traffic	FBIG	22738.380005
76	C3_SpringTraffic_FBIG_US_Brand_Traffic_AlwaysO...	Traffic	FBIG	21450.420005
89	C3_FallAlwaysOnTraffic_FBIG_US_Brand_Traffic_A...	Traffic	FBIG	19037.080004
...	...	...	...	...
642	C3_BroadCTV_TTD_FR_Brand_Awareness_Promo_CTV_S...	Awareness	TTD	0.000000
643	C3_BTSTraffic_FBIG_US_Brand_Traffic_AlwaysOn_S...	Traffic	FBIG	0.000000
644	C3_UK-SiteVisitors_FBIG_UK_CO_Traffic_Evergree...	Traffic	FBIG	0.000000
645	C3_XpressShipTraffic_FBIG_US_Brand_Traffic_Pro...	Traffic	FBIG	0.000000
647	C3_FR-SiteVisitors_FBIG_FR_CO_Traffic_Evergree...	Traffic	FBIG	0.000000

110 rows × 6 columns

```
In [20]: df = pd.read_excel('StockX Click Dataset.xlsx')
df
```

Out[20]:

	Campaign Name	Objective	Platform	Media Co
0	C3_Dynamic-Remarketing_FBIG_US_CO_CatalogSales...	CatalogSales	FBIG	107906.7000
1	C3_Dynamic-Remarketing_FBIG_US_CO_CatalogSales...	CatalogSales	FBIG	102058.1599
2	C3_DABA_FBIG_US_CO_CatalogSales_AlwaysOn_Socia...	CatalogSales	FBIG	101864.2999
3	C3_Dynamic-Remarketing_FBIG_US_CO_CatalogSales...	CatalogSales	FBIG	89141.4499
4	C3_Dynamic-Remarketing_FBIG_US_CO_CatalogSales...	CatalogSales	FBIG	81975.4600
...	...	...	...	
647	C3_FR-SiteVisitors_FBIG_FR_CO_Traffic_Evergree...	Traffic	FBIG	0.0000
648	C3_Dynamic-Remarketing_FBIG_US_CO_CatalogSales...	CatalogSales	FBIG	N.
649	C3_DABA_FBIG_US_CO_CatalogSales_AlwaysOn_Socia...	CatalogSales	FBIG	N.
650	C3_BrandTraffic_TikTok_US_AW_Traffic_AlwaysOn_...	Traffic	TikTok	N.
651	C3_InfluencersBoost_IG_US_AW_Awareness_Other_S...	Awareness	IG	N.

652 rows × 6 columns

```
In [21]: new_df = df.groupby('Platform')['Media Cost','Link Clicks'].sum()
```

<ipython-input-21-dba9853a8d4c>:1: FutureWarning: Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a list instead.

```
new_df = df.groupby('Platform')['Media Cost','Link Clicks'].sum()
```

```
In [22]: #creating a new pivot table out of the original dataframe
new_df = df.groupby('Platform')['Media Cost','Link Clicks'].sum()
new_df['cpc'] = new_df['Media Cost']/new_df['Link Clicks']
new_df
```

<ipython-input-22-7d12dc594f79>:2: FutureWarning: Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a list instead.

```
new_df = df.groupby('Platform')['Media Cost','Link Clicks'].sum()
```

Out[22]:

	Media Cost	Link Clicks	cpc
Platform			
Bing	4.604870e+03	39745	0.115860
DV360	3.086136e+06	2479940	1.244440
FBIG	2.306379e+06	34931043	0.066027
Google	1.160063e+06	7408534	0.156585
IG	1.262056e+05	38526	3.275856
Snapchat	2.751152e+05	338642	0.812407
TTD	1.177072e+05	16295	7.223516
TikTok	3.896818e+05	398404	0.978107
YouTube	4.185748e+04	0	inf

```
In [23]: new_df['Percent LC'] = new_df['Link Clicks']/new_df['Link Clicks'].su
new_df['Percent LC'] = new_df['Percent LC'].round(2)
```

In [24]: new\_df

Out[24]:

	Media Cost	Link Clicks	cpc	Percent LC
Platform				
Bing	4.604870e+03	39745	0.115860	0.00
DV360	3.086136e+06	2479940	1.244440	0.05
FBIG	2.306379e+06	34931043	0.066027	0.77
Google	1.160063e+06	7408534	0.156585	0.16
IG	1.262056e+05	38526	3.275856	0.00
Snapchat	2.751152e+05	338642	0.812407	0.01
TTD	1.177072e+05	16295	7.223516	0.00
TikTok	3.896818e+05	398404	0.978107	0.01
YouTube	4.185748e+04	0	inf	0.00



In [ ]: