In [2]:
```
#import thư viện
import pandas as pd
import numpy as np
import matplotlib as plt
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime
print("import completed")
```

```
import completed
```

In [3]:
```
#import dataset
df = pd.read_csv(r"./datasets/marketplace_data_2019.csv")
print(df.head())
```

```
         date ttt_group  clicks_A  clicks_B  clicks_C  cost_A  cost_B  cost_C  \
0  2019-01-06     short    117272     68608     27152  113299   61987   21848
1  2019-01-06    medium     96050     12415    137291   74761    7483  111740
2  2019-01-06      long      7060      9568    408676    3407    4796  327505
3  2019-01-13     short    109700     64097     25477  105743   57771   20462
4  2019-01-13    medium    128847     16787    172639  101018   10201  140393

   bookings_A  bookings_B  bookings_C  booking_rev_A  booking_rev_B  \
0        5664        2651        1311         864767         423745
1        3738         386        5343         565066          60799
2         170         184        9813          27480          27867
3        5277        2467        1225         794086         387316
4        5049         526        6766         795529          80124

   booking_rev_C
0         197976
1         812847
2        1506297
3         193915
4        1020371
```

In [1]:
```
#Read file from drive link if needed
# url = "https://drive.google.com/file/d/1DqHfdYcEprFxmPXJBHV5HK27q5IoMxlA/view?usp=sharing"
# url='https://drive.google.com/uc?id=' + url.split('/')[-2]
# df_test = pd.read_csv(url)
# df_test.head()
```

In [4]:
```
# Review data
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1095 entries, 0 to 1094
Data columns (total 14 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   date           1095 non-null   object
 1   ttt_group      1095 non-null   object
 2   clicks_A       1095 non-null   int64 
 3   clicks_B       1095 non-null   int64 
 4   clicks_C       1095 non-null   int64 
 5   cost_A         1095 non-null   int64 
 6   cost_B         1095 non-null   int64 
 7   cost_C         1095 non-null   int64 
 8   bookings_A     1095 non-null   int64 
 9   bookings_B     1095 non-null   int64 
 10  bookings_C     1095 non-null   int64 
 11  booking_rev_A  1095 non-null   int64 
 12  booking_rev_B  1095 non-null   int64 
 13  booking_rev_C  1095 non-null   int64 
dtypes: int64(12), object(2)
memory usage: 119.9+ KB
None
```

In [5]:
```
#check numerical data
print(df.shape)
print(df.describe())
```

```
(1095, 14)
          clicks_A      clicks_B       clicks_C        cost_A         cost_B  \
count  1.095000e+03  1.095000e+03    1095.000000  1.095000e+03    1095.000000
mean   2.904898e+05  1.337045e+05  220404.340639  2.686955e+05  118263.228311
std    3.398313e+05  2.095589e+05  132328.225244  3.300333e+05  191975.004493
min    3.240000e+02  2.533000e+03   14651.000000  1.270000e+02    1276.000000
25%    6.878500e+03  8.029500e+03  123992.000000  3.285500e+03    4270.000000
50%    1.876150e+05  1.919800e+04  191998.000000  1.542680e+05   11610.000000
75%    4.491345e+05  2.146910e+05  295214.000000  3.984460e+05  194146.500000
max    1.714072e+06  1.041133e+06  727874.000000  1.661516e+06  956674.000000

              cost_C    bookings_A    bookings_B    bookings_C  booking_rev_A  \
count    1095.000000   1095.000000   1095.000000   1095.000000   1.095000e+03
mean   177837.655708  13382.377169   5072.309589   7633.052055   2.092607e+06
std    106864.506556  16435.779118   8140.733786   4189.684237   2.565217e+06
min     11849.000000      6.000000     41.000000    705.000000   9.670000e+02
25%     99865.000000    163.500000    169.500000   4450.500000   2.546450e+04
50%    154798.000000   7697.000000    619.000000   6875.000000   1.187892e+06
75%    237919.000000  19849.000000   8330.500000  10001.000000   3.150882e+06
max    596599.000000  82702.000000  40205.000000  23564.000000   1.315033e+07

       booking_rev_B  booking_rev_C
count   1.095000e+03   1.095000e+03
mean    7.901320e+05   1.191080e+06
std     1.264995e+06   6.544745e+05
min     6.152000e+03   1.073490e+05
25%     2.630700e+04   6.915185e+05
50%     9.648000e+04   1.072385e+06
75%     1.306488e+06   1.565771e+06
max     6.117843e+06   3.669508e+06
```

In [6]:
```
#check for missing data
print(df.isnull().sum())
```

```
date            0
ttt_group       0
clicks_A        0
clicks_B        0
clicks_C        0
cost_A          0
cost_B          0
cost_C          0
bookings_A      0
bookings_B      0
bookings_C      0
booking_rev_A   0
booking_rev_B   0
booking_rev_C   0
dtype: int64
```

In [15]:
```
#Sort by the values along "date"(ascending) and "ttt_group"(descending) columns
df = df.sort_values(by = ["date","ttt_group"], ascending = [True,False])
print(df.head())
```

```
           date ttt_group  clicks_A  clicks_B  clicks_C  cost_A  cost_B  \
312  2019-01-01     short    110658     62984     23897  107515   57033
313  2019-01-01    medium     61201      8115     81676   47501    4893
314  2019-01-01      long      4945      7970    301992    2378    4068
471  2019-01-02     short     99122     54777     21459   96726   49634
472  2019-01-02    medium     59130      8040     79556   45954    4869

     cost_C  bookings_A  bookings_B  bookings_C  booking_rev_A  booking_rev_B  \
312   19170        5359        2440        1157         815380         374662
313   65722        2373         252        3167         371986          39742
314  242810         118         153        7231          19093          23636
471   17295        4798        2121        1039         720475         328767
472   64165        2294         250        3086         360364          39281

     booking_rev_C
312         173965
313         512900
314        1190356
471         164653
472         474731
```

In [8]:
```
#check unique data in "ttt_group" column
print(df["ttt_group"].unique())
```

```
['short' 'medium' 'long']
```

In [9]:
```
#Review "date" column
print(df["date"].value_counts().min())
print(df["date"].value_counts().max())
print(df["date"].value_counts())
```

```
3
3
2019-01-01    3
2019-09-09    3
2019-09-07    3
2019-09-06    3
2019-09-05    3
             ..
2019-05-01    3
2019-04-30    3
2019-04-29    3
2019-04-28    3
2019-12-31    3
Name: date, Length: 365, dtype: int64
```

In [10]:
```
#check for duplication
dup = df.duplicated()
duprow = df[dup]
print(duprow)
```

```
Empty DataFrame
Columns: [date, ttt_group, clicks_A, clicks_B, clicks_C, cost_A, cost_B, cost_C, bookings_A, bookings_B, booki
ngs_C, booking_rev_A, booking_rev_B, booking_rev_C]
Index: []
```

In [14]:
```
#export data
df.to_csv('./datasets/marketplace_cleandata_2019.csv', index = False)
print("export completed")
```

```
export finished
```