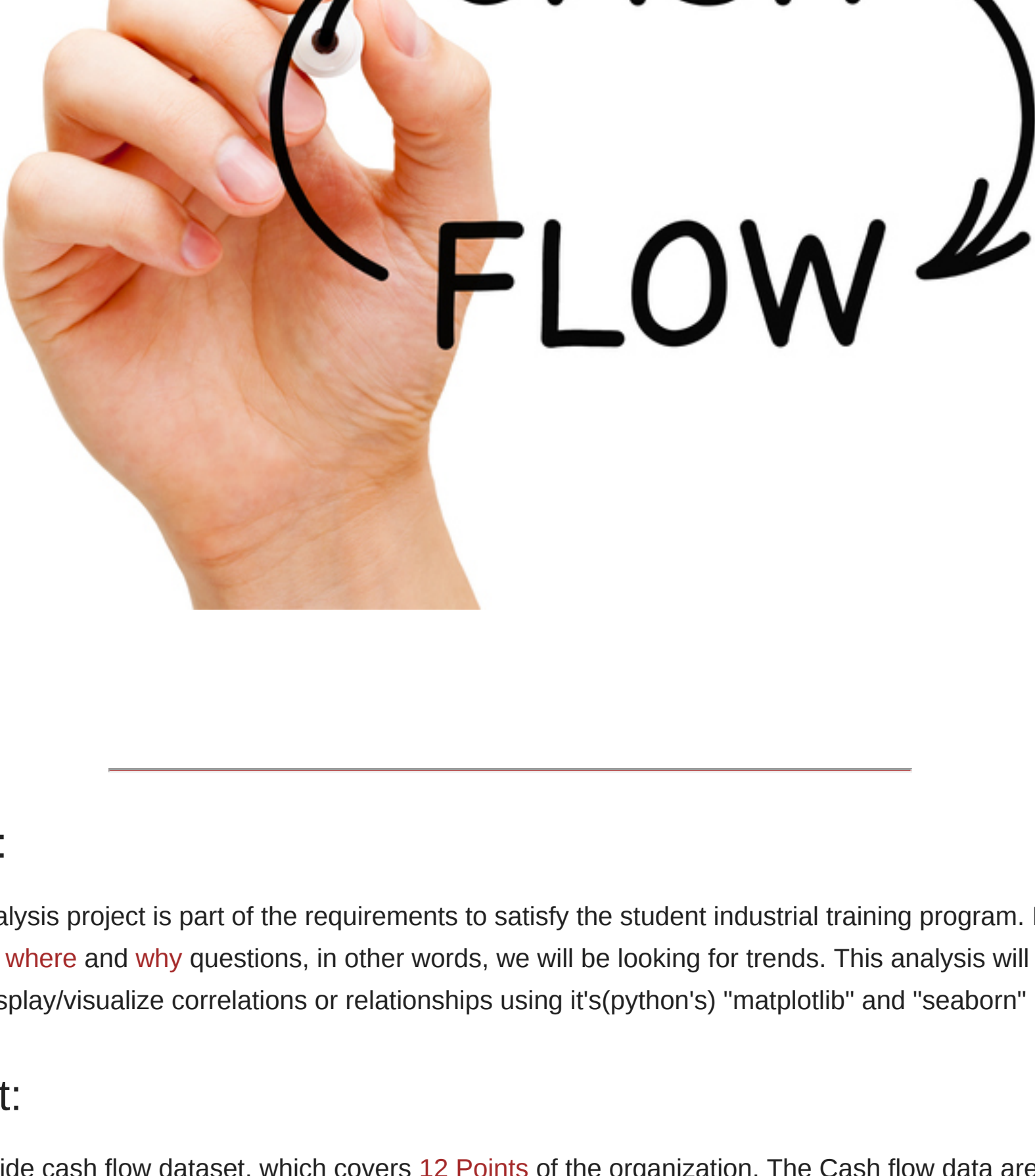


# Cash Flow - Analysis



## Introduction

### About - Project:

This Exploratory data analysis project is part of the requirements to satisfy the student industrial training program. In this analysis, we will be asking the **what**, **how**, **where** and **why** questions, in other words, we will be looking for trends. This analysis will be performed using "python", and we shall display/visualize correlations or relationships using it's(python's) "maiplotlib" and "seaborn" libraries.

### About - Dataset:

This is an organization-wide cash flow dataset, which covers 12 Points of the organization. The Cash flow data are collected from 6th of december 2021 to 6th of may 2022 using a location-garnering web application, or what some may refer to as an API(application programming interface). This web application which is hosted locally, gathers information and actively returns accurate data of each location/point. This dataset contains 8 Columns and 1705 rows , it's features include: **name** of location , **date** of shift, **means** of payment , **amount realised** and **receipt count** . we will be discussing more about this dataset in the sections below.

## Basic Questions

- How much was realised from each location within this period?
- How much was realised by the organization within this period?
- How often were the different payment methods used, and what is the ratio/percentage?
- How much does realised amount vary between locations?
- Does the realised amount depend on the location? Is there any correlation?
- How much was realised between december and january, and does the festive period impact amount realised?
- and more....

Let's get started!

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker
import matplotlib.dates as mdates

import seaborn as sns

import datetime
import matplotlib
sns.set_style('darkgrid')
```

```
In [2]: cash_df = pd.read_excel('HSL.xlsx', na_values=['?'])
```

### Data at a glance:

```
In [3]: df = cash_df

In [4]: cash_df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1705 entries, 0 to 1704
Data columns (total 8 columns):
 #   Column      Non-Null Count  Dtype
---  -
 0   Location    1627 non-null    object
 1   ShiftDate   1627 non-null    object
 2   Cash        1627 non-null    float64
 3   Cheque      1627 non-null    float64
 4   POS         1627 non-null    float64
 5   MobileMoney 1627 non-null    float64
 6   TotalAmount 1627 non-null    float64
 7   ReceiptCount 1627 non-null    float64
dtypes: float64(6), object(2)
memory usage: 106.7+ KB
```

```
In [5]: cash_df.describe()

   Cash      Cheque      POS      MobileMoney      TotalAmount      ReceiptCount
count  1.627000e+03    1627.000000    1627.000000    1627.0    1.627000e+03    1627.000000
mean    2.003025e+05    878.103258    3605.650058    1627.0    2.106302e+05    52.151199
std     2.491035e+05    25139.635517    30837.078592    0.0    2.567110e+05    46.616799
min     2.000000e+02    0.000000    0.000000    0.0    2.000000e+02    1.000000
25%     4.268644e+04    0.000000    0.000000    0.0    4.268644e+04    16.000000
50%     1.029448e+05    0.000000    0.000000    0.0    1.048072e+05    32.000000
75%     2.851495e+05    0.000000    0.000000    0.0    2.890657e+05    82.000000
max     2.597121e+06    778888.900000    876733.000000    0.0    2.597121e+06    227.000000
```

```
In [6]: cash_df['Location'].unique()

Out[6]: array(['point 1', 'point 6', 'point 2', 'point 3', 'point 4',
              'point 5', 'point 8', 'point 7', 'point 9', 'point 10',
              'HEART 2 HEART vPOINT', 'point 1', 'point 2', 'point 3',
              'HEART 2 HEART vPOINT', 'nan', 'HISTOPATHOLOGY vMORTUARY',
              'HISTOPORT vPOINT', 'ACCIDENT & EMERGENCY vPHARMACY',
              'GOPD POINT 3'], dtype=object)
```

### Data preparation and cleaning:

```
In [7]: #shuffling the dataframe
df = df.sample(frac=1)

In [8]: df = df.dropna(axis=1)

In [9]: df.loc[df['Location']!= 'POINT']

Out[9]:   Location  ShiftDate  Cash  Cheque  POS  MobileMoney  TotalAmount  ReceiptCount
```

```
In [10]: df.Location.unique()

Out[10]: array(['point 9', 'point 6', 'point 2', 'point 1', 'point 8',
              'HISTOPORT vMPOINT', 'point 7', 'point 10', 'point 5',
              'HEART 2 HEART vPOINT', 'point 4', 'point 3', 'point 1',
              'HEART 2 HEART vPOINT', 'nan', 'HISTOPATHOLOGY vMORTUARY',
              'GOPD POINT 3', 'HISTOPATHOLOGY vMORTUARY', 'ACCIDENT & EMERGENCY vPHARMACY',
              'point/pharmacy', 'point 1'], dtype=object)
```

```
In [37]: cash_df['ShiftDate'] = pd.to_datetime(cash_df['ShiftDate'], infer_datetime_format=True)

In [32]: cash_df['ShiftDate']

Out[32]: 0    2021-06-12
1    2021-06-12
2    2021-06-12
3    2021-06-12
4    2021-06-12
```

```
1709    2022-06-05
1701    2022-06-05
1702    2022-06-05
1703    2022-06-05
1704    2022-06-05
Name: ShiftDate, Length: 1705, dtype: datetime64[ns]
```

```
In [ ]:
```

### Exploratory data analysis:

```
df[['Location', 'TotalAmount']]

Out[13]:   Location  TotalAmount
835    point 9    3445.00
40     point 6    45460.00
19     point 5    260000.00
668    point 1    25373.80
1025   point 1    153352.00
377    point 7    66697.84
976    point 8    25514.39
292    point 1    51467.00
949    point 1    55116.00
866    point 2    714439.52

1627 rows x 2 columns
```

```
In [14]: import matplotlib
matplotlib.rcParams['font.size']=14
matplotlib.rcParams['figure.figsize']=(9,5)
matplotlib.rcParams['figure.facecolor']='#e6e6e6e6'
```

```
In [15]: t_sum = df[['TotalAmount']].sum()
t_mean = df[['TotalAmount']].mean()
t_sum

Out[15]: 343938441.28000003
```

## Insights:

1. The total amount realised within this period was **343,030,441.28(Three hundred and 43 million, three hundred and four thousand, forty one naira, twenty eight kobo.)**

```
In [16]: loc_bar=df['Location'].value_counts()
loc_bar

Out[16]: point 1    600
         point 4    108
         point 5    106
         point 7    104
         point 10   101
         point 6    91
         point 8    91
         HEART 2 HEART vPOINT    77
         HISTOPORT vMPOINT    75
         GOPD POINT 3    35
         HISTOPHARMACY    3
         HISTOPHARMACY    2
         ACCIDENT & EMERGENCY vPHARMACY    2
         HISTOPATHOLOGY vMORTUARY    1
         point 3    1
         Name: Location, dtype: int64
```

```
In [17]: loc_bar = df['Location'].value_counts().reset_index().head(10)
loc_bar

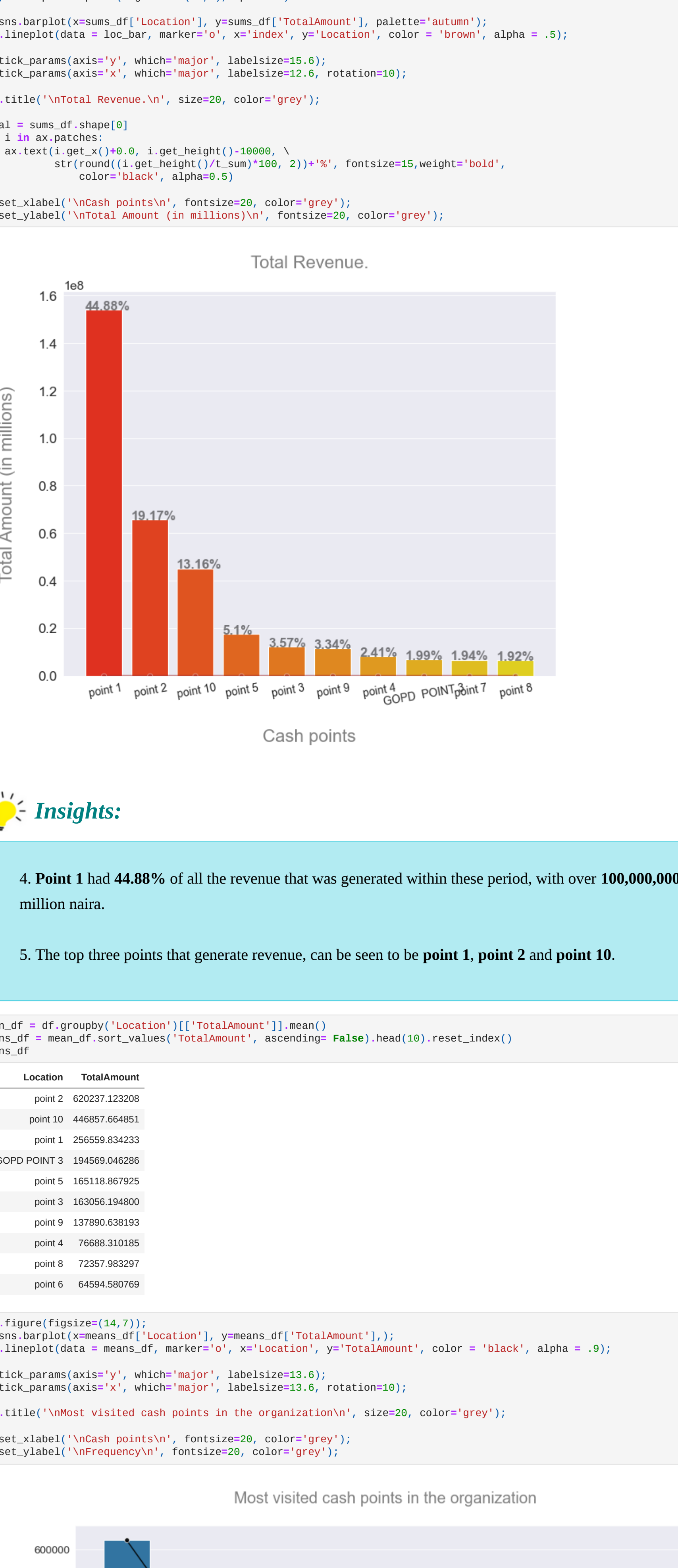
Out[17]:   index  Location
0         point 1    600
1         point 4    108
2         point 5    106
3         point 7    104
4         point 10   101
5         point 6    91
6         point 8    91
7         point 9    83
8         HEART 2 HEART vPOINT    77
```

```
In [18]: plt.figure(figsize=(14,7))
sns.barplot(x=loc_bar['Location'], y=loc_bar['index'], palette='summer');
sns.lineplot(data = loc_bar, marker='o', x='index', y='Location', color = 'brown', alpha = .5);
ax.tick_params(axis='y', which='major', labelsize=15.8);
ax.tick_params(axis='x', which='major', labelsize=14.6, rotation=30);

plt.title('\nMost visited cash points in the organization\n', size=20, color='grey');

ax.set_xlabel('\nCASH points\n', fontsize=20, color='grey');
ax.set_ylabel('\nFrequency\n', fontsize=20, color='grey');
```

Most visited cash points in the organization



## Insights:

2. Point 1 was the most visited within this period with **600** visitations.
3. HEART 2 HEART POINT was the least visited within this period with a little below **100** visitations.

```
In [19]: sum_df = df.groupby('Location')[['TotalAmount']].sum()
sums_df = sum_df.sort_values('TotalAmount', ascending=False).head(10).reset_index()
sums_df

Out[19]:   Location  TotalAmount
0         point 1  1.539350e+08
1         point 2  6.576514e+07
2         point 10  4.31320e+07
3         point 5  1.762020e+07
4         point 3  1.229232e+07
5         point 9  1.144492e+07
6         point 4  8.282338e+06
7  GOPD POINT 3  6.809917e+06
8         point 7  6.648042e+06
9         point 8  6.584576e+06
```

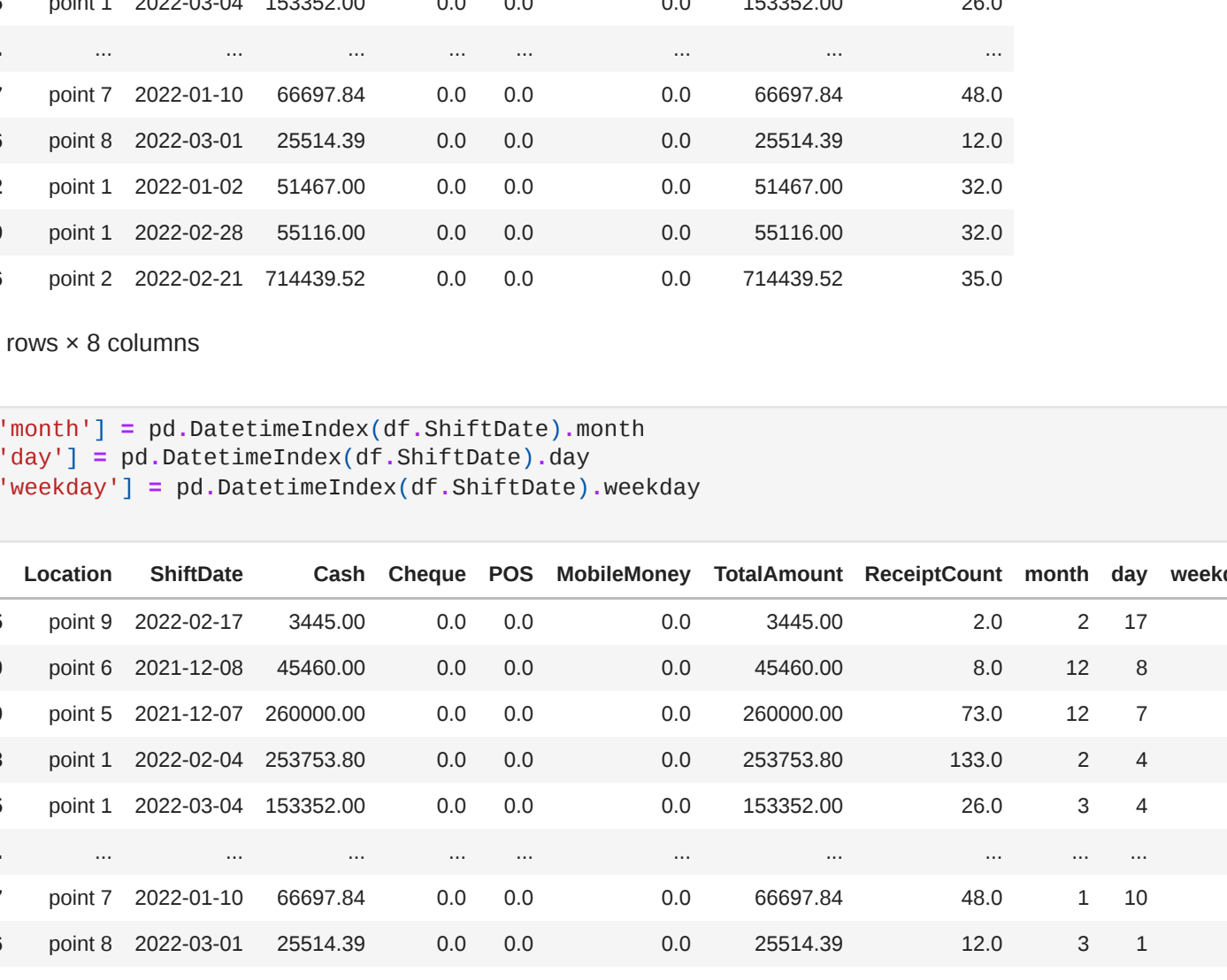
```
In [20]: fig, ax = plt.subplots(figsize = (10,8), dpi = 80)
sns.barplot(x=sums_df['Location'], y=sums_df['TotalAmount'], palette='autumn');
sns.lineplot(data = loc_bar, marker='o', x='index', y='Location', color = 'brown', alpha = .5);
ax.tick_params(axis='y', which='major', labelsize=13.6);
ax.tick_params(axis='x', which='major', labelsize=14.6, rotation=30);

plt.title('\nTotal Revenue\n', size=20, color='grey');

total = sums_df.shape[0]
for i in ax.patches:
    ax.text(i.get_x()+0.05, i.get_height()-100000,
           f'{str(round(i.get_height()/total*100, 1))}%\n',
           color='black', alpha=0.5)

ax.set_xlabel('\nCASH points\n', fontsize=20, color='grey');
ax.set_ylabel('\nTotal Amount (in millions)\n', fontsize=20, color='grey');
```

Total Revenue.



## Insights:

4. Point 1 had **44.88%** of all the revenue that was generated within these period, with over **100,000,000** million naira.
5. The top three points that generate revenue, can be seen to be **point 1, point 2 and point 10**.

```
In [21]: mean_df = df.groupby('Location')[['TotalAmount']].mean()
means_df = mean_df.sort_values('TotalAmount', ascending=False).head(10).reset_index()
means_df

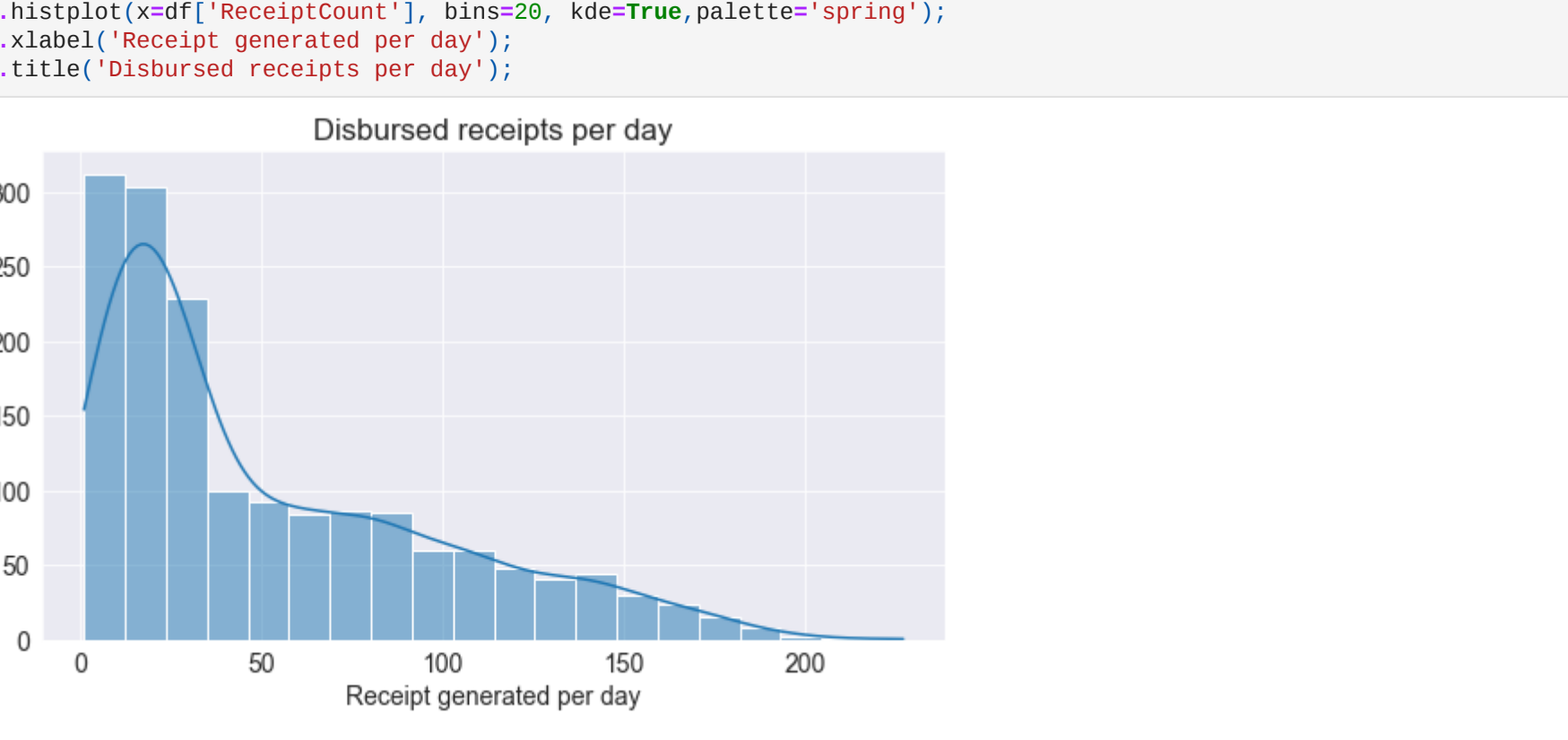
Out[21]:   Location  TotalAmount
0         point 2  62027.122508
1         point 10  44897.660681
2         point 1  25059.834233
3  GOPD POINT 3  194569.046296
4         point 5  185118.867925
5         point 3  163506.194800
6         point 9  137890.638103
7         point 4  76688.310185
8         point 8  72357.983297
9         point 6  64504.580769
```

```
In [22]: plt.figure(figsize=(14,7))
sns.barplot(x=means_df['Location'], y=means_df['TotalAmount'], color = 'black', alpha = .9);
sns.lineplot(data = means_df, marker='o', x='Location', y='TotalAmount', color = 'black', alpha = .5);
ax.tick_params(axis='y', which='major', labelsize=13.6);
ax.tick_params(axis='x', which='major', labelsize=13.6, rotation=30);

plt.title('\nMost visited cash points in the organization\n', size=20, color='grey');

ax.set_xlabel('\nCASH points\n', fontsize=20, color='grey');
ax.set_ylabel('\nFrequency\n', fontsize=20, color='grey');
```

Most visited cash points in the organization



## Insights:

6. On Average, **point 2** generated more money, while **point 6** is in the last position among the top 10.

```
In [23]: df.head()

Out[23]:   Location  ShiftDate  Cash  Cheque  POS  MobileMoney  TotalAmount  ReceiptCount
835    point 9  17/02/2022    3445.0    0.0    0.0    0.0    3445.0    2.0
40     point 6  01/2/2021    45460.0    0.0    0.0    0.0    45460.0    8.0
19     point 5  21/2/2021    260000.0    0.0    0.0    0.0    260000.0    73.0
668    point 1  17/2/2021    25373.8    0.0    0.0    0.0    25373.8    1.0
1025   point 1  4/2/2022    153352.0    0.0    0.0    0.0    153352.0    26.0
1627 rows x 8 columns
```

```
In [24]: df['ShiftDate'] = pd.to_datetime(df['ShiftDate'], infer_datetime_format=True)

In [25]: df['ShiftDate'] = pd.to_datetime(df['ShiftDate'], format = '%d-%m-%Y')
df

Out[25]:   Location  ShiftDate  Cash  Cheque  POS  MobileMoney  TotalAmount  ReceiptCount
835    point 9  2022-02-17    3445.0    0.0    0.0    0.0    3445.00    2.0
40     point 6  2021-12-08    45460.0    0.0    0.0    0.0    45460.00    8.0
19     point 5  2021-12-07    260000.0    0.0    0.0    0.0    260000.00    73.0
668    point 1  2022-02-04    25373.80    0.0    0.0    0.0    25373.80    1.0
1025   point 1  2022-03-04    153352.00    0.0    0.0    0.0    153352.00    26.0
...     ...     ...     ...     ...     ...     ...     ...
976    point 7  2022-01-10    66697.84    0.0    0.0    0.0    66697.84    48.0
377    point 8  2022-03-01    25514.39    0.0    0.0    0.0    25514.39    12.0
292    point 1  2022-01-02    51467.00    0.0    0.0    0.0    51467.00    32.0
949    point 1  2022-02-28    55116.00    0.0    0.0    0.0    55116.00    32.0
866    point 2  2022-02-21    714439.52    0.0    0.0    0.0    714439.52    35.0
1627 rows x 8 columns
```

```
In [26]: df['month'] = pd.DatetimeIndex(df['ShiftDate']).month
df['day'] = pd.DatetimeIndex(df['ShiftDate']).day
df['weekday'] = pd.DatetimeIndex(df['ShiftDate']).weekday
df

Out[26]:   Location  ShiftDate  Cash  Cheque  POS  MobileMoney  TotalAmount  ReceiptCount  month  day  weekday
835    point 9  2022-02-17    3445.00    0.0    0.0    0.0    3445.00    2.0    2    17    3
40     point 6  2021-12-08    45460.00    0.0    0.0    0.0    45460.00    8.0    12    8    2
19     point 5  2021-12-07    260000.00    0.0    0.0    0.0    260000.00    73.0    12    7    1
668    point 1  2022-02-04    25373.80    0.0    0.0    0.0    25373.80    1.0    2    4    4
1025   point 1  2022-03-04    153352.00    0.0    0.0    0.0    153352.00    26.0    3    4    4
...     ...     ...     ...     ...     ...     ...     ...
976    point 7  2022-01-10    66697.84    0.0    0.0    0.0    66697.84    48.0    1    10    0
377    point 8  2022-03-01    25514.39    0.0    0.0    0.0    25514.39    12.0    3    1    1
292    point 1  2022-01-02    51467.00    0.0    0.0    0.0    51467.00    32.0    1    2    6
949    point 1  2022-02-28    55116.00    0.0    0.0    0.0    55116.00    32.0    2    28    0
866    point 2  2022-02-21    714439.52    0.0    0.0    0.0    714439.52    35.0    2    21    0
1627 rows x 11 columns
```

```
In [27]: df['weekday'].plot(kind='box', figsize = (12,6), vert = False);
plt.title('Distribution of visitations among weekdays')

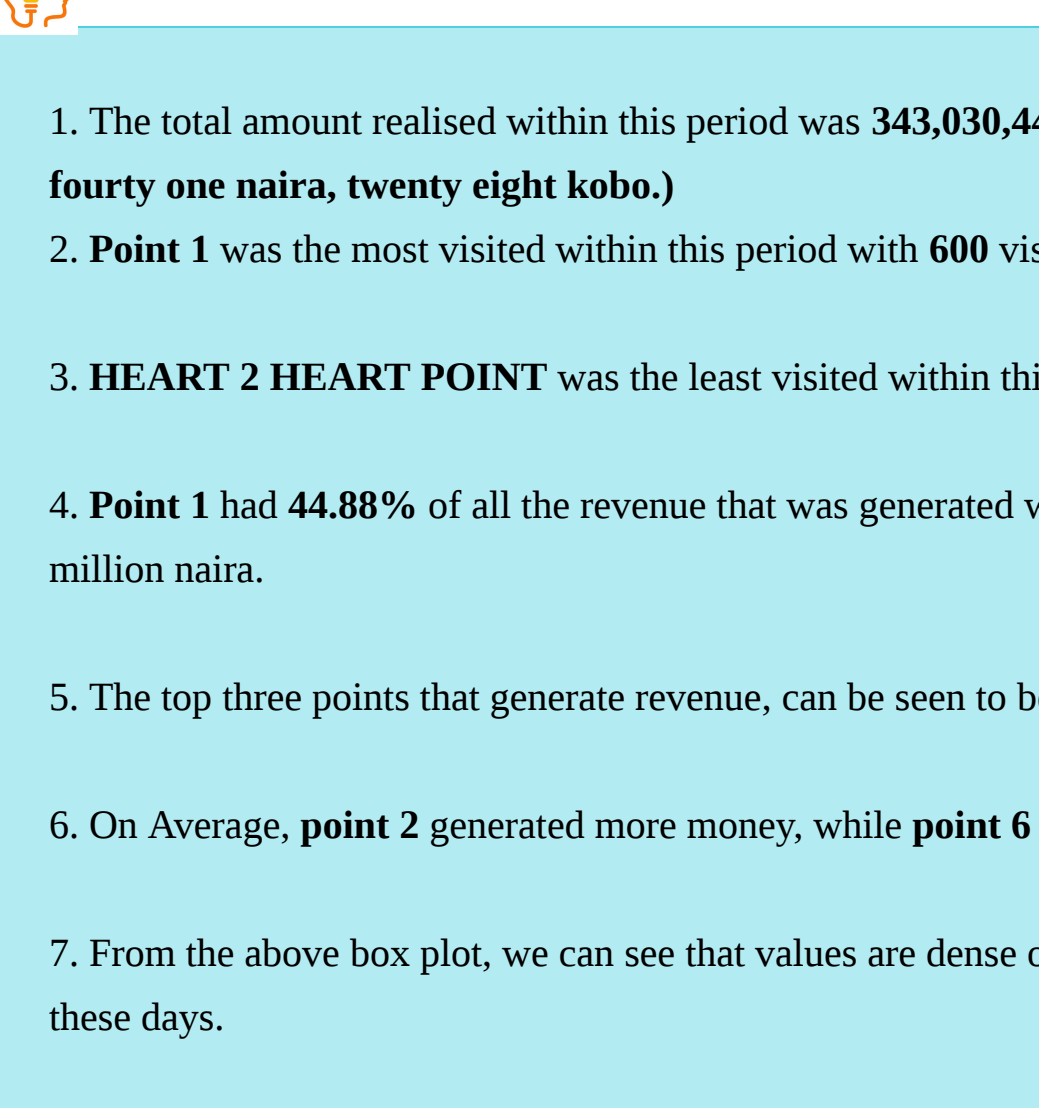
Out[27]:
```

## Insights:

7. From the above box plot, we can see that values are dense on days **1 (monday)** to **4 (thursday)**, meaning that there are more visitations on these days.

```
In [28]: sns.kstest(x=df['ReceiptCount'], bins=20, kstest=True, palette='spring');
plt.xlabel('Receipt generated per day');
plt.title('Disbursed receipts per day');

Out[28]:
```



```
In [29]: d_amount = df[(df['month'] == 1) & (df['TotalAmount'])]
d_amount['d_amount'] = df['TotalAmount'].sum()
d_amount

Out[29]:   d_amount
1    1.905724e+06
```

```
In [30]: receipt = df['ReceiptCount'].sum()
re = receipt/day

# 471.3888888888889
# 84858.9

In [31]: days = 360
t_d = t_sum/day
t_dre

Out[31]: 4942.7865796187878

In [ ]:
```

## Insights:

8. 471 receipts are printed on the average daily.

```
In [36]: janderpec = janderpec.sum()*100
jander = d_amount*dre
jander = jander['TotalAmount']
jander, janderperc

# (16287920.820000001, 30.285337485602646)
```

## Insights:

9. On the average, **1,905,724.67 (one million, nine hundred and five thousand, seven hundred and twenty four naira, 67 kobo)** was realised daily, and for each printed receipt comes **4,042.78(four thousand, forty two naira, 78 kobo)**.

```
In [36]: plt.figure(figsize=(10,6));
ax = sns.barplot(x=df['month'], y=df['TotalAmount'],);
plt.title('Revenue Generated by months')

Out[36]:
```



## Insights:

11. From the above bar plot, more revenue was generated in the month of **May**, and less was generated in the month of **August**.

## Conclusion

### All Insights:

1. The total amount realised within this period was **343,030,441.28(Three hundred and 43 million, three hundred and four thousand, forty one naira, twenty eight kobo.)**

2. Point 1 was the most visited within this period with **600** visitations.

3. HEART 2 HEART POINT was the least visited within this period with a little below **100** visitations.

4. Point 1 had **44.88%** of all the revenue that was generated within these period, with over **100,000,000** million naira.

5. The top three points that generate revenue, can be seen to be **point 1, point 2 and point 10**.

6. On Average, **point 2** generated more money, while **point 6** is in the last position among the top 10.

7. From the above box plot, we can see that values are dense on days **1 (monday)** to **4 (thursday)**, meaning that there are more visitations on these days.

8. 471 receipts are printed on the average daily.

9. On the average, **1,905,724.67 (one million, nine hundred and five thousand, seven hundred and twenty four naira, 67 kobo)** was realised daily, and for each printed receipt comes **4,042.78(four thousand, forty two naira, 78 kobo)**.

10. **103,887,926.82 (one hundred and three million, eight hundred and eighty seven thousand, nine hundred and twenty six naira, eighty two kobo)** in revenue, was generated between december 2021 and january 2022, with a percentage of **30.3**.

11. From the above bar plot, more revenue was generated in the month of **May**, and less was generated in the month of **August**.

## References

- Stackoverflow: <https://stackoverflow.com/questions/68870547/date-and-time-conversion-in-python-pandas>
- opendatasets: Python library: <https://github.com/Joaviz/MOpenDdatasets>
- Pandas User guide: [https://pandas.pydata.org/docs/user\\_guide/index.html](https://pandas.pydata.org/docs/user_guide/index.html)
- matplotlib User guide: <https://matplotlib.org/3.3.3/users/index.html>
- Seaborn User guide & tutorial: <https://seaborn.pydata.org/tutorial.html>