



Google Ad_Hours_Analysis_With_Python.

Date : 10/06/2023

Project Start Date - End Date	<ul style="list-style-type: none">● Start Date – 08 -06 -2023● End Date – 10-06 2023
Objectives	<ul style="list-style-type: none">● To analyses how many people who clicked on the advertisement enrolled in our course● General exploratory analyses● General descriptive analyses
Milestones accomplished the week of Start Date - End Date:	<ul style="list-style-type: none">● Descriptive analyses● Exploratory analyses● Classification of data with respect to term

Milestones accomplished the week of Start Date - End Date	<ul style="list-style-type: none"> • Data import • Data overview • Data preprocessing • Descriptive analysis. • linear regreesion • Exploratory data analysis. • Main alogorithms • Correlations • Conclusion • Insights

Contact Information

This project is performed for educational purpose of under the guidance of Siddhivinayak Sir .

Project Manager

Name : Siddhivinayak Phulwadkar

Mobile: 9028965955

Email:

siddhivinayakphulwadkar@gmail.com

Student Name

Name : Veerbhadra Govind Lavharale

Mobile: 9607194266

Email: virbhadralavrale@gmail.com.

Project Abstract

The dataset is about showing ... 1. **DATA IMPORT AND OVERVIEW:-** import data set by using pandas ,numpy seaboarn, by ploting use matplotlib .pyplot. then overview the data how data big or small there is no null find out .

```
In [274]: import pandas as ma
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [115]: vs=ma.read_excel("~/desktop/Marketing Data Google Ads 6th june.xlsx")
```

```
In [116]: vs
```

14	07:00:00	2135	21.3500	0.149450
15	07:30:00	18498	369.9600	2.589720
16	08:00:00	17843	535.2900	3.747030
17	08:30:00	12741	445.9350	3.121545
18	09:00:00	18973	683.0280	4.781196
19	09:30:00	11746	469.8400	3.288880
20	10:00:00	10967	219.3400	1.535380
21	10:30:00	8742	419.6160	2.937312
22	11:00:00	4059	48.7080	0.340956
23	11:30:00	20565	411.3000	2.879100
24	12:00:00	24486	710.0940	4.970658
25	12:30:00	20491	676.2030	4.733421

2. DATA PREPROCESSING :-

#check number of unique value from all data set

```
vs.isnull().sum()
```

#remove columns

```

In [66]: #data preprocessing

In [67]: #check number of unique value from all data set

In [118]: vs.select_dtypes(include='object').nunique()
Out[118]: Sr no    48
          dtype: int64

In [119]: vs.shape
Out[119]: (48, 4)

In [120]: vs.isnull().sum()
Out[120]: Sr no    0
          Impressions    0
          Clicks    0
          Sales Unit    0
          dtype: int64

#remove columns

In [121]: ty=vs.drop("Sr no",axis=1)

In [122]: ty

```

36	22504	720.1280	5.040896
37	39613	1980.6500	13.864550
38	20654	206.5400	1.445780
39	14229	284.5800	1.992060
40	11294	112.9400	0.790580

3.DESRIPTIVE ANALYSIS:-

It describe the information about data set like its std . deviation nad mean ,max and count etc

```

In [ ]: #descriptive analysis.

In [127]: c=don.isnull().sum()

In [132]: c
Out[132]: Impressions    0
          Clicks    0
          Sales Unit    0
          dtype: int64

In [ ]: #in that data set there is no null value.

In [133]: vs.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48 entries, 0 to 47
Data columns (total 4 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   Sr no       48 non-null    object
1   Impressions  48 non-null    int64
2   Clicks      48 non-null    float64
3   Sales Unit  48 non-null    float64
dtypes: float64(2), int64(1), object(1)
memory usage: 1.6+ KB

In [ ]: # to get a better understanding of the dataset,
# we can also see the statistical summary of the dataset.

In [134]: vs.describe()
Out[134]:

```

	Impressions	Clicks	Sales Unit
count	48.000000	48.000000	48.000000

In [133]: vs.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48 entries, 0 to 47
Data columns (total 4 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   Sr no           48 non-null    object
1   Impressions      48 non-null    int64
2   Clicks           48 non-null    float64
3   Sales Unit       48 non-null    float64
dtypes: float64(2), int64(1), object(1)
memory usage: 1.6+ KB
```

In []: *# to get a better understanding of the dataset,*
we can also see the statistical summary of the dataset.

In [134]: vs.describe()

Out[134]:

	Impressions	Clicks	Sales Unit
count	48.000000	48.000000	48.000000
mean	45981.187500	1756.474554	12.295322
std	77958.290851	4133.072442	28.931507
min	139.000000	1.390000	0.009730
25%	2855.750000	31.362500	0.219537
50%	15359.000000	391.092000	2.737644
75%	21116.500000	689.794500	4.828561
max	294484.000000	22969.752000	160.788264

Linear regression.

```
[344]: don.nunique()

t[344]: Impressions    48
        Clicks        48
        Sales Unit    48
        dtype: int64

[137]: don.sum()

t[137]: Impressions    2.207097e+06
        Clicks        8.431078e+04
        Sales Unit    5.901755e+02
        dtype: float64

In [ ]: # Impression are indicating total visibility of ads on 6th june 2023
        # total impression of day is 2.207097e+06.
        #Clicks are indicating futher intersted pepole who clicks on there website for getting information about this ad
        # total Clicks of that day is 8.431078e+04
        #sales unit indicates to how many people enroll this course after clicks made in 30 min slot
        # total sales units is 5.901755e+02

[139]: don.mean()

t[139]: Impressions    45981.187500
        Clicks        1756.474554
        Sales Unit    12.295322
        dtype: float64

In [ ]: # there is all avaerage sum of impression, clicks , sales units .
        # to get identify how many sales are done .

[140]: don.isnull().sum()

t[140]: Impressions    0
        Clicks        0
        Sales Unit    0
        dtype: int64

n [75]: from sklearn.model_selection import train_test_split
        x_train, x_test, Y_train, Y_test = train_test_split( x,y, test_size = 0.30, random_state = 0)

n [76]: from sklearn.linear_model import LinearRegression

n [77]: lr= LinearRegression()

n [78]: lr

ut[78]: ▼ LinearRegression
        LinearRegression()

n [79]: lr.fit(x_train,Y_train)

ut[79]: ▼ LinearRegression
        LinearRegression()

n [80]: y_predict=lr.predict(x_test)

n [81]: y_predict

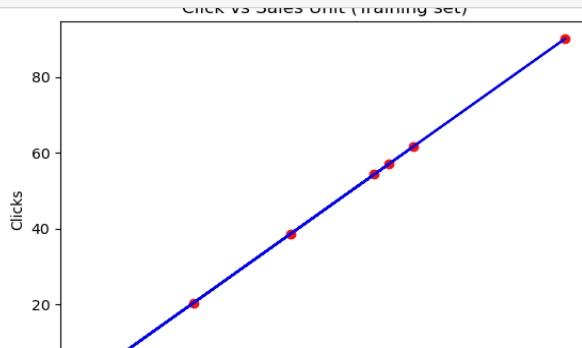
ut[81]: array([ 3.75060000e+00,  1.25580000e-01,  8.67961920e+00,  3.75060000e-01,
                4.13133000e+00,  1.38645500e+01,  4.04042100e+00,  7.90580000e-01,
                1.00800000e-02,  2.77200000e-02,  1.91800000e-02,  3.09561000e+00,
                2.95976800e+00,  1.41730400e+01,  1.60788264e+02])

n [82]: plt.scatter(x_train, Y_train, color = 'red')
        plt.plot(x_train, lr.predict(x_train), color = 'blue')
        plt.title('Click vs Sales Unit (Training set)')
        plt.xlabel('Sales Unit')
        plt.ylabel('Clicks')
        plt.show()
```

Exploratory data analysis.- Exploratory data analysis is an approach of analyzing data sets to summarize their main characteristics, often using statistical graphics and other data visualization methods. A statistical model can be used or not, but primarily EDA is for seeing what the data can tell us beyond the formal modeling or hypothesis testing task.

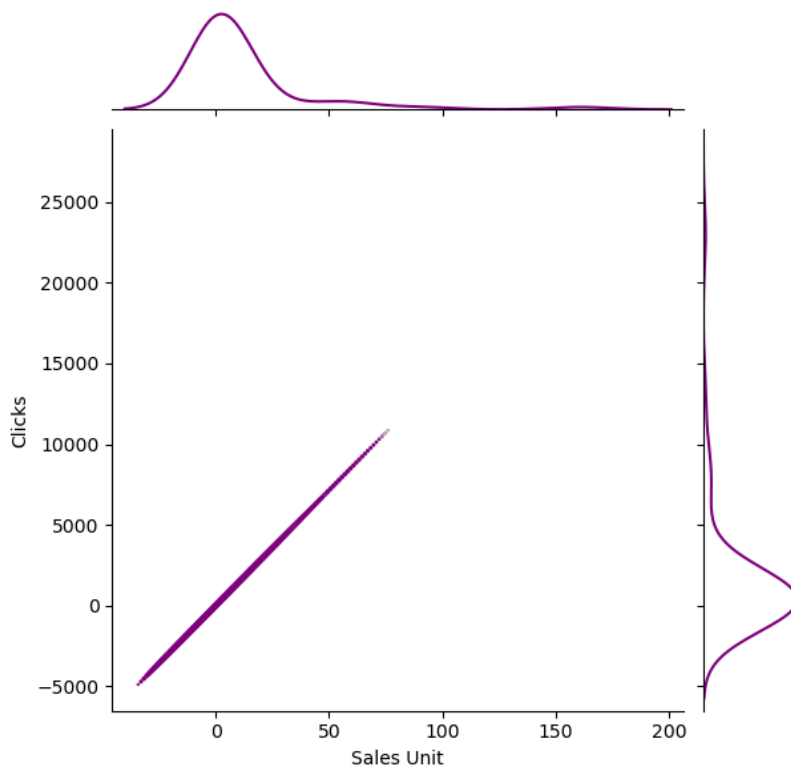
4.13133000e+00, 1.38645500e+01, 4.04042100e+00, 7.90580000e-01,
1.00800000e-02, 2.77200000e-02, 1.91800000e-02, 3.09561000e+00,
2.95976800e+00, 1.41730400e+01, 1.60788264e+02])

```
n [82]: plt.scatter(x_train, Y_train, color = 'red')
plt.plot(x_train, lr.predict(x_train), color = 'blue')
plt.title('Click vs Sales Unit (Training set)')
plt.xlabel('Sales Unit')
plt.ylabel('Clicks')
plt.show()
```



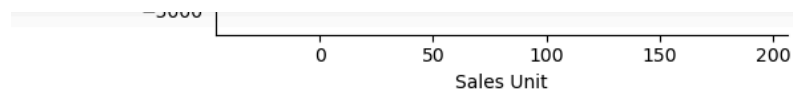
```
n [83]: # In exploratory data analysis there is analysis of how many people who clicked on the advertisement enrolled in our
# there is increase the impression or clicks to increase the sales.
# there is peak time have people whose clicks more on day time .
```

```
n [86]: sns.jointplot(x='Sales Unit',y='Clicks',data=vs,color='purple',kind='kde');
```



```
n [87]: #main algorithms
```

Main Alogorithms



```
n [87]: #main alogorithms
```

```
n [88]: #Correlations
```

```
n [89]: # For whole dataset
```

```
n [90]: don.corr()
```

```
ut[90]:
```

	Impressions	Clicks	Sales Unit
Impressions	1.000000	0.888854	0.888854
Clicks	0.888854	1.000000	1.000000
Sales Unit	0.888854	1.000000	1.000000

```
n [91]: vs.corr()
```

```
/var/folders/fg/fp2383x97vj0tjyfhglnn_40000gn/T/ipykernel_1253/332366957.py:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.
```

```
ut[91]:
```

	Impressions	Clicks	Sales Unit
Impressions	1.000000	0.888854	0.888854
Clicks	0.888854	1.000000	1.000000
Sales Unit	0.888854	1.000000	1.000000

```
n [92]: # For some selected coulmns or attributes
```



```
n [92]: # For some selected coulmns or attributes
```

```
n [93]: rt=vs[['Clicks','Sales Unit']].corr()
```

```
n [94]: rt
```

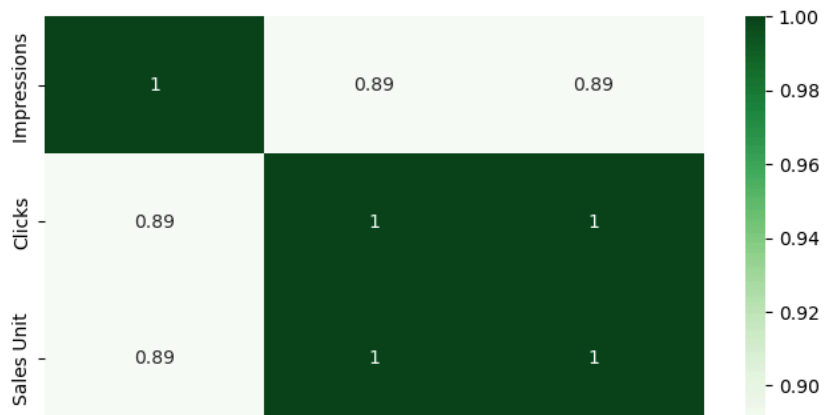
```
ut[94]:
```

	Clicks	Sales Unit
Clicks	1.0	1.0
Sales Unit	1.0	1.0

```
n [95]: corr = don.corr()
```

```
plt.figure(figsize=(8,4))  
sns.heatmap(corr,cmap="Greens",annot=True)
```

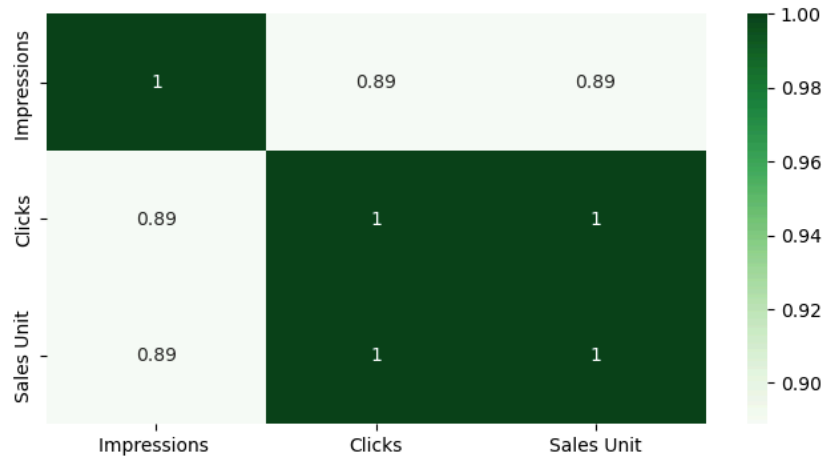
```
ut[95]: <Axes: >
```



Sales Unit 1.0 1.0

```
n [95]: corr = don.corr()
plt.figure(figsize=(8,4))
sns.heatmap(corr,cmap="Greens",annot=True)
```

```
ut[95]: <Axes: >
```



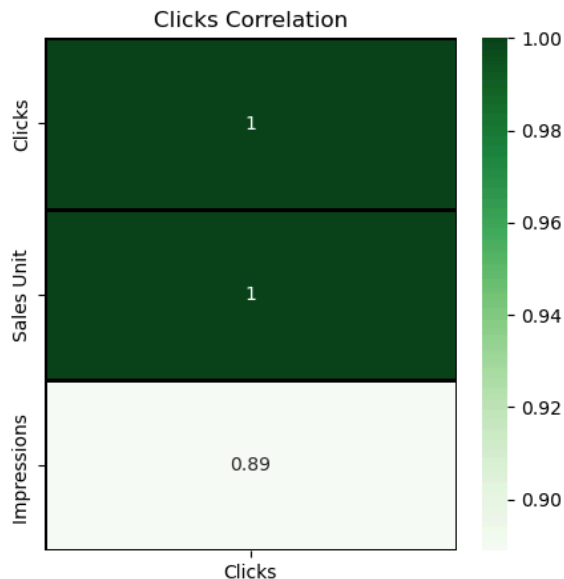
```
n [96]: corr = don.corrwith(don['Clicks']).sort_values(ascending = False ).to_frame()
corr.columns = ['Clicks']
plt.subplots(figsize=(5,5))
sns.heatmap(corr,annot= True,cmap = 'Greens',linewidths=2,linestyle='black');
plt.title('Clicks Correlation')
```

```
ut[96]: Text(0.5, 1.0, 'Clicks Correlation')
```

Correlations – relation between two variables .

```
n [96]: corr = don.corrwith(don['Clicks']).sort_values(ascending = False ).to_frame()
corr.columns = ['Clicks']
plt.subplots(figsize=(5,5))
sns.heatmap(corr,annot= True,cmap = 'Greens',linewidths=2,linestyle='black');
plt.title('Clicks Correlation')
```

```
ut[96]: Text(0.5, 1.0, 'Clicks Correlation')
```



```
n [97]: corr = don.corrwith(don['Sales Unit']).sort_values(ascending = False ).to_frame()
```

conclusion of analysis.

#various phases of data analysis including data collection, cleaning and analysis are discussed briefly.

#Explorative data analysis is mainly studied here.

#For the implementation, Python programming language is used.

#For detailed research, jupyter notebook is used. Different Python libraries and packages are introduced.

We can see that the Impression ,Clicks and Sales units there are interrelation between them

we can see that when clicks increase Sales also increase .

The clicks had the best sales.

We can see that the linerregression to increase sales with increase the clicks and impression

we can see that the average std deviation of impression very spread due to marketing canablize.

We can see that the std deviation of clicks is some low than Impression because there is low awarenes of this ads.

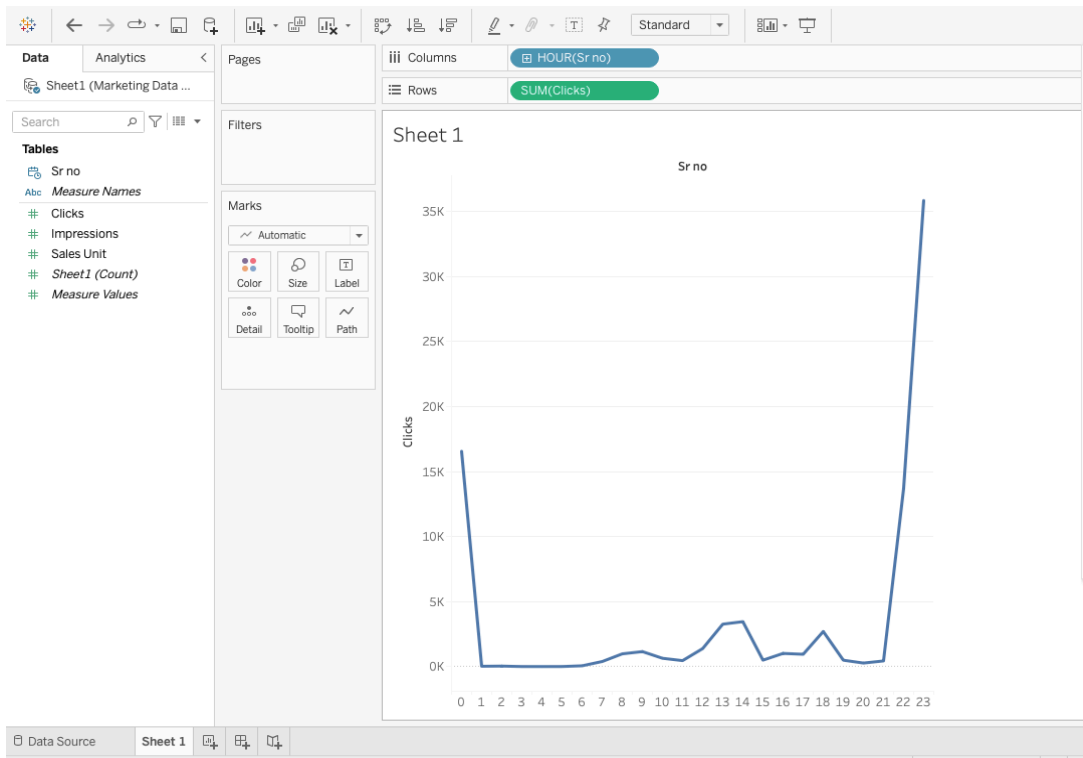
Insights

various phases of data analysis including data collection, cleaning and analysis are discussed briefly. Explorative data analysis is mainly studied here. For the implementation, Python programming language is used. For detailed research, jupyter notebook is used. Different Python libraries and packages are introduced. Using various analysis and visulaization methods, numerous results are extracted.

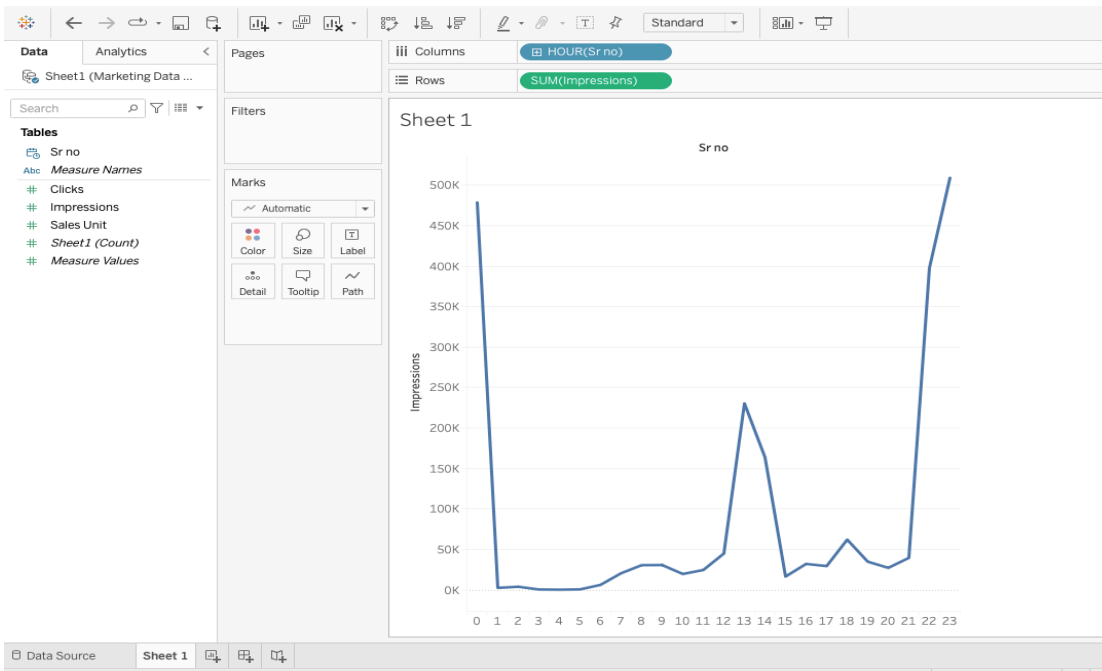
In all about analaysis dataset to inform that general marketing and how the people was aware about advertisment

this advertisment was 6th june. It main think that there was tuesday is a working day . people mindset was to do workaholic or motivated .that day they search or aware about cources.some people was went house from office that time is about 12.am some people go to saw this particular ads but not click . to all dataset analysis there was impression was slightly peak but not click this ads .some people to aware this ads more information was find to click them this ads then this ads useful for this these people want to sale this course. By the analysis all 30 min data there will changes starting 2 hr more impression but less clicks . then there was less peak of impression and clicks in 12 to 6 am and sales units due to late night there was some people sleep. So morning was high peak for impression and clicks . there was weak point for marketing . marketing behaviour currently on average impression was 45981.18 but clicks is low was 1756.47. and 12.49 percentage people was sales this course and impression very spread due to marketing canablize. In marketing strategy was difference there is changes is done in marketing . there was people mindset change per hours.

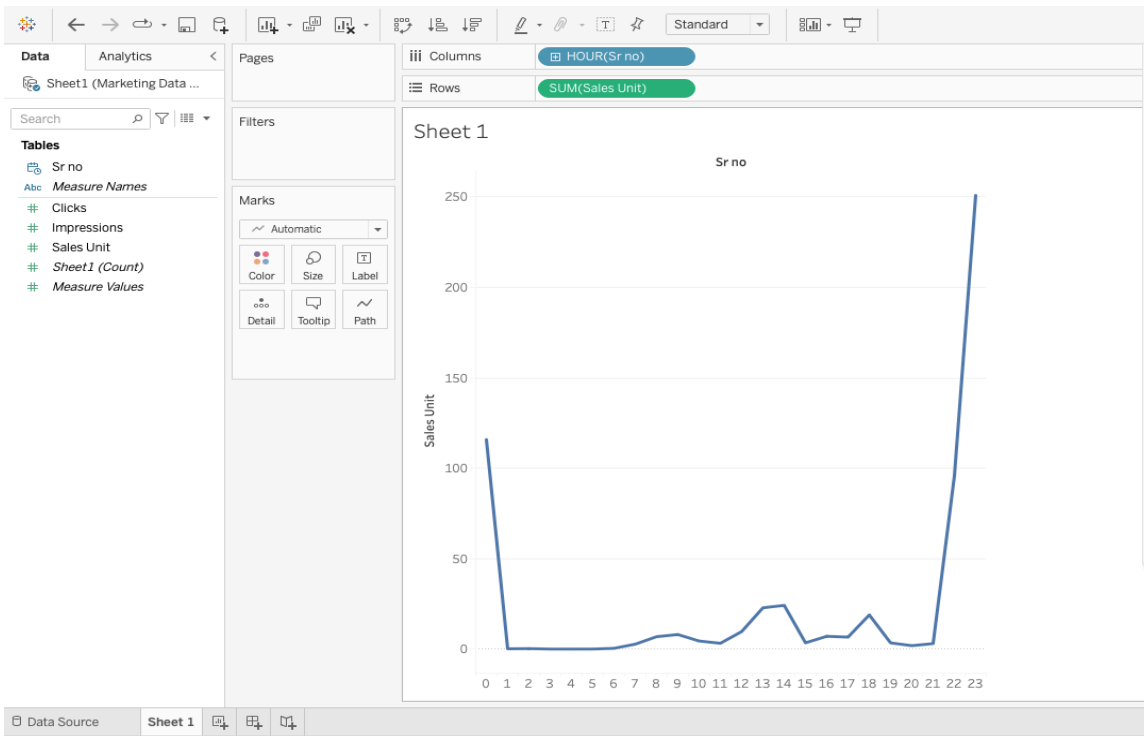
I suggest this solution to analysis the data advertising was in morning time show or publish there was people doing some activity in starting day and some people was motivated to do some work or learn some thing . new technology . there was anlaysis google ads in per hour how to changes its behavior or mindset to analysis and what time ads show on google .



It is all data visualize form data set in tableau there is changes in per hours of impression and hours . it data inform or solution gives us per hours people mindset was changes and there was timeing was matter when people was thinking about its carrier that's times was sales is more there is peak is high in morning time 11 to 1pm it was a peak time



In that visualize there is time changes sales also changes .all about people mindset .



there was timing was matter when people was thinking about its carrier that's times was sales is more there is peak is high in morning time 11 to 1pm it was a peak time so that analysis this google ad in hour . by exploratory analysis there is x-axis shows sales unit y-axis shows clicks it was dependent variables there is interdependent on each other so clicks is high sales units is more.

