

Google Chrome

swaroop280201 (Swaroop Patil) x Fire and Smoke - Youtube.ipynb x OnlineSalesAI.ipynb - Colaboratory x

https://colab.research.google.com/drive/1v9DGIXkvqGh7IDNLTtRt1xmpMF7U14L\_V#scrollTo=aYnhgy5VmrqP

OnlineSalesAI.ipynb ☆

File Edit View Insert Runtime Tools Help Saving...

Comment Share Settings Profile

RAM 100% Disk 100%

Editing

+ Code + Text

Name: Swaroop Patil

From Vishwakarma Institute of Technology, Pune

PRN No.-11910410

OnlineSales.ai Task

Importing Required Libraries

+ Code + Text

1 import pandas as pd  
2 import plotly as py  
3 import numpy as np  
4 import folium  
5 import warnings  
6 from IPython.display import display, HTML  
7 import plotly.graph\_objects as go  
8 import numpy as np  
9 import plotly.express as px  
10 from ipywidgets import interact  
11 import ipywidgets as widgets  
12 import seaborn as sns  
13 warnings.simplefilter('ignore')  
14 import matplotlib.pyplot as plt  
15 from wordcloud import WordCloud

1s completed at 2:54 PM

Type here to search

14:54 08-01-2023

GoogleColabFire and Smoke - Youtube.ipynbOnlineSalesAI.ipynb - Colaboratory

https://colab.research.google.com/drive/1v9DGIXkvqGh7IDNLTtRt1xmpMF7U14L\_V#scrollTo=GuxtFtOnmrqY

OnlineSalesAI.ipynb

FileEditViewInsertRuntimeToolsHelp

+ Code+ Text

RAMDisk

Editing

Creating date parser for parsing dates

[2]1 def myparser(x):2 | return datetime.strptime(x, '%d/%B/%Y %H:%M:%S')3

Reading csv file

(Data Augumentation)

[3]1 from google.colab import drive2 drive.mount('/content/drive')

Mounted at /content/drive

[4]1 df = pd.read\_csv('/content/drive/MyDrive/OnlineSales.ai/police\_department\_data.csv', parse\_dates=True, date\_parser=myparser) # ,index\_col=['crime\_date']

1 crime\_df = df.copy()2 crime\_df.head()

incident_id	category	crime_description	crime_date	department_district	resolution	address	department_id	location
100050070	WEAPON	POSS OF PROHIBITED	2016-01-29	SOUTH BRAN	ARREST,	800 BLOCK OF	100000070000000	(37.775420706711,

0s completed at 2:55 PM

Type here to search

14:5509-01-2023



Google Chrome

swaroop280201 (Swaroop Pati) x Fire and Smoke - Youtube.ipynb x OnlineSalesAl.ipynb - Colaboratory x

https://colab.research.google.com/drive/1v9DGIXkvqGh7IDNLTtRt1xmpMF7U14L\_V#scrollTo=-\_inKu\_4mrqc

OnlineSalesAl.ipynb

File Edit View Insert Runtime Tools Help

Comment Share

RAM Disk

Editing

+ Code + Text

Getting the columns details

1 crime\_df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150500 entries, 0 to 150499
Data columns (total 9 columns):
#   Column                Non-Null Count  Dtype
---  -
0   incident_id            150500 non-null int64
1   category               150500 non-null object
2   crime_description       150500 non-null object
3   crime_date             150500 non-null object
4   department_district     150499 non-null object
5   resolution             150500 non-null object
6   address               150500 non-null object
7   department_id          150500 non-null int64
8   location              150500 non-null object
dtypes: int64(2), object(7)
memory usage: 10.3+ MB
```

Removing the brackets and commas from location columns

Splitting the location value into lat and long individual columns

0s completed at 2:55 PM

Type here to search

14:55 00-01-2023

Google Chrome tabs: swaroop280201 (Swaroop Pati), Fire and Smoke - Youtube.ipynb, OnlineSalesAl.ipynb - Colaboratory

Address bar: https://colab.research.google.com/drive/1v9DGIXkvqGh7IDNLTtRt1xmpMF7U14L\_V#scrollTo=kU4XMi6-mrqe

OnlineSalesAl.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

RAM 100% Disk 100% Editing

Removing the brackets and commas from location columns

Splitting the location value into lat and long individual columns

Maintaining atomcity of the columns

```
1 crime_df['location'] = crime_df['location'].str.replace('(', '')
2 crime_df['location'] = crime_df['location'].str.replace(')', '')
3 crime_df['location'] = crime_df['location'].str.replace(',', '')
4 crime_df[['lat','long']] = crime_df.location.str.split(expand=True)
5 crime_df.head(1)
```

	incident_id	category	crime_description	crime_date	department_district	resolution	address	department_id	location	lat
0	120058272	WEAPON LAWS	POSS OF PROHIBITED WEAPON	2016-01-29 11:00:00	SOUTHERN	ARREST, BOOKED	800 Block of BRYANT ST	12005827212120	37.775420706711 -122.403404791479	37.775420706711 -122.40340479

Dropping the location columns after seprating lat and long, location became redundant

0s completed at 2:56 PM

Type here to search

Taskbar icons: File Explorer, Microsoft Edge, Google Chrome, Outlook, WhatsApp, etc.

System tray: 14:56, 08-01-2023, ENG

OnlineSalesAI.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

RAM 100% Disk 100% Editing

1

▼ Dropping the location columns after seprating lat and long, location became redundant

```
[8] 1 crime_df= crime_df.drop('location', axis=1)
      2
```

▼ Checking the cardanility of data

```
[9] 1 crime_df.shape

(150500, 10)
```

▼ Grouping data into categories of commited crime and visualizing data using bubble chart

```
1 category_df = pd.DataFrame((crime_df.groupby( [ "category" ] ).count()).reset_index())
2 category_df.drop(['crime_description','crime_date','department_district','resolution','address','department_id','lat','long'], axis =1, inplace =True)
3 category_df = category_df.rename(columns={'incident_id': 'incident_count'})
4
```

1 category\_df.sorted = category\_df.sort\_values('incident\_count', ascending=False)

0s completed at 2:56 PM

Type here to search

14:56 08-01-2023



OnlineSalesAI.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

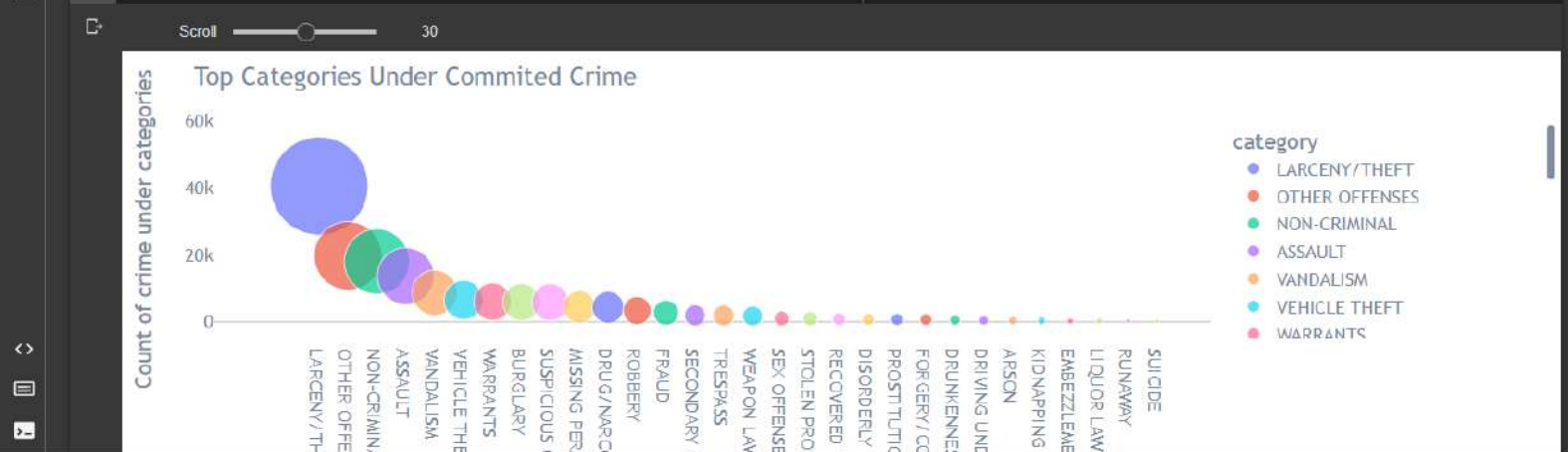
```
1 category_df_sorted = category_df.sort_values('incident_count',ascending=False)
2 ipywidgetLayout = widgets.Layout(border='solid 2px green')
3
4 def bubble_chart(Scroll):
5
6     figBubble = px.scatter(
7         category_df_sorted.head(Scroll),
8         x = 'category',
9         y = 'incident_count',
10        size= 'incident_count',
11        color = 'category',
12        hover_name='category' ,
13        size_max=60
14    )
15    figBubble.update_layout(
16
17        title="Top Categories Under Committed Crime",
18        yaxis_title="Count of crime under categories",
19        xaxis_title="Categories",
20        font=dict(family="Trebuchet MS",size=16,color="#778899"),
21        xaxis_showgrid=True,
22        yaxis_showgrid=True,
23        plot_bgcolor='rgba(0,0,0,0)',
24
25    )
26
27
28    figBubble.update_xaxes(zeroLine=True, zeroLineWidth=2, zeroLineColor='#D0D0D0')
```

0s completed at 2:57 PM

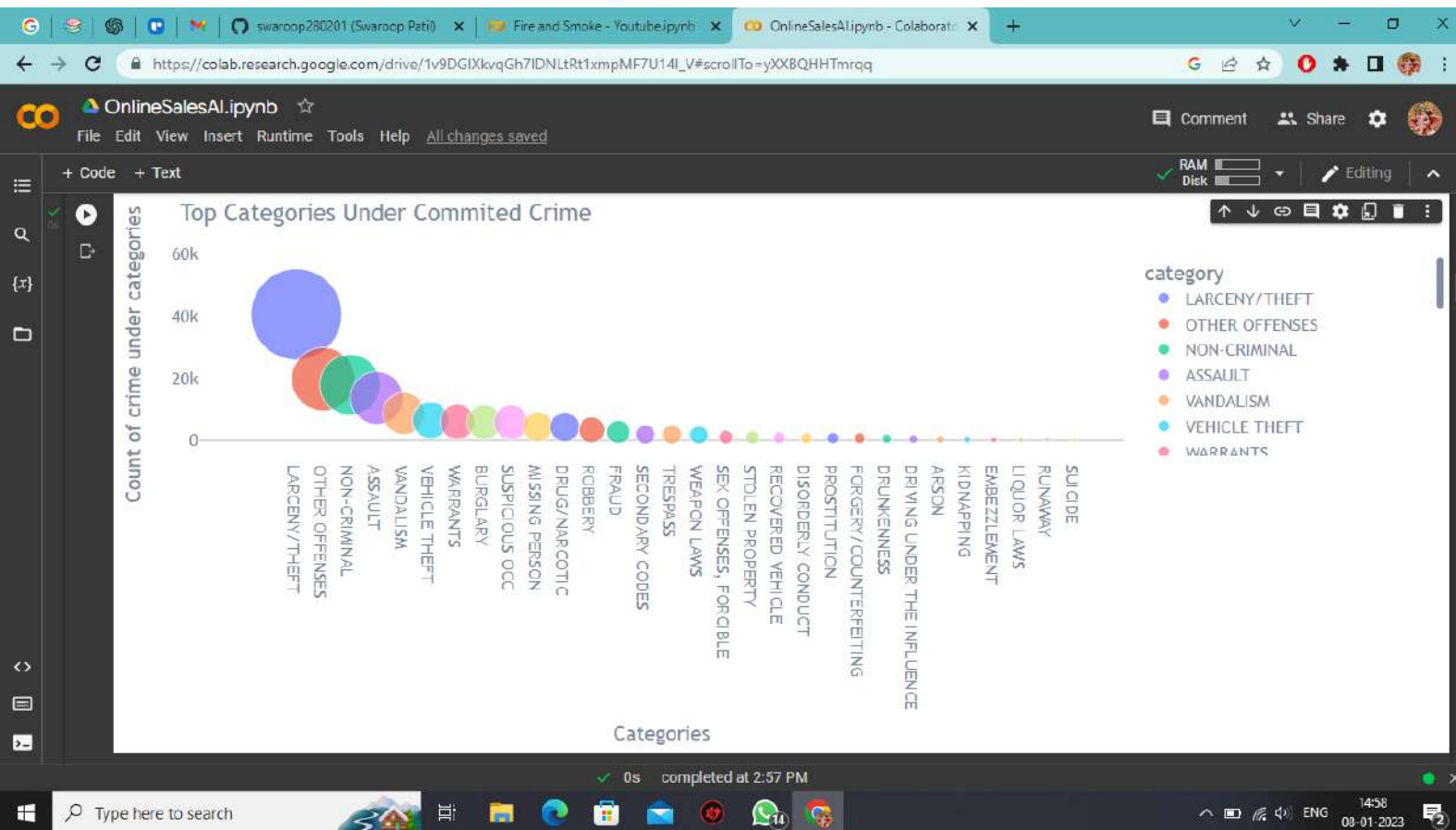
Type here to search

14:57 08-01-2023

```
27
28 figBubble.update_xaxes(zeroline=True, zerolinewidth=2, zerolinecolor='#D0D0D0')
29 figBubble.update_yaxes(zeroline=True, zerolinewidth=2, zerolinecolor='#D0D0D0')
30
31 figBubble.show();
32
33 interact(bubble_chart, Scroll=30);
```







OnlineSalesAI.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

RAM  
Disk

Editing

### Grouping data into department category of offences and visualizing data using bubble chart & bar chart

```
[14] 1 department_district = pd.DataFrame((crime_df.groupby(["department_district"]).count()).reset_index())
2 department_district.drop(['crime_description', 'crime_date', 'category', 'resolution', 'address', 'department_id', 'lat', 'long'], axis=1, inplace=True)
3 department_district = department_district.rename(columns={'incident_id': 'incident_count'})

1 department_district_sorted = department_district.sort_values('incident_count', ascending=False)
2 ipyWidgetLayout = widgets.Layout(border='solid 2px green')
3
4 def bubble_chart(Scroll):
5
6     figBubble = px.scatter(
7         department_district_sorted.head(Scroll),
8         x='department_district',
9         y='incident_count',
10        size='incident_count',
11        color='department_district',
12        hover_name='department_district',
13        size_max=60
14    )
15    figBubble.update_layout(
16
17        title="Top Crime Committed in District wise Department",
18        yaxis_title="Count of crime committed",
19        xaxis_title="Departments",
```

0s completed at 2:59 PM

Type here to search

14:59  
08-01-2023

OnlineSalesAI.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

```
9
10
11     y = 'incident_count',
12     size= 'incident_count',
13     color = 'department_district',
14     hover_name='department_district',
15     size_max=60
16 )
17 figBubble.update_layout(
18     title="Top Crime Committed in District wise Department",
19     yaxis_title="Count of crime committed",
20     xaxis_title="Departments",
21     font=dict(family="Trebuchet MS",size=16,color="#778899"),
22     xaxis_showgrid=True,
23     yaxis_showgrid=True,
24     plot_bgcolor='rgba(255,255,255)',
25 )
26
27
28 figBubble.update_xaxes(zeroLine=True, zeroLineWidth=2, zeroLineColor='red')
29 figBubble.update_yaxes(zeroLine=True, zeroLineWidth=2, zeroLineColor='red')
30
31 figBubble.show();
32
33 interact(bubble_chart, Scroll=10);
```

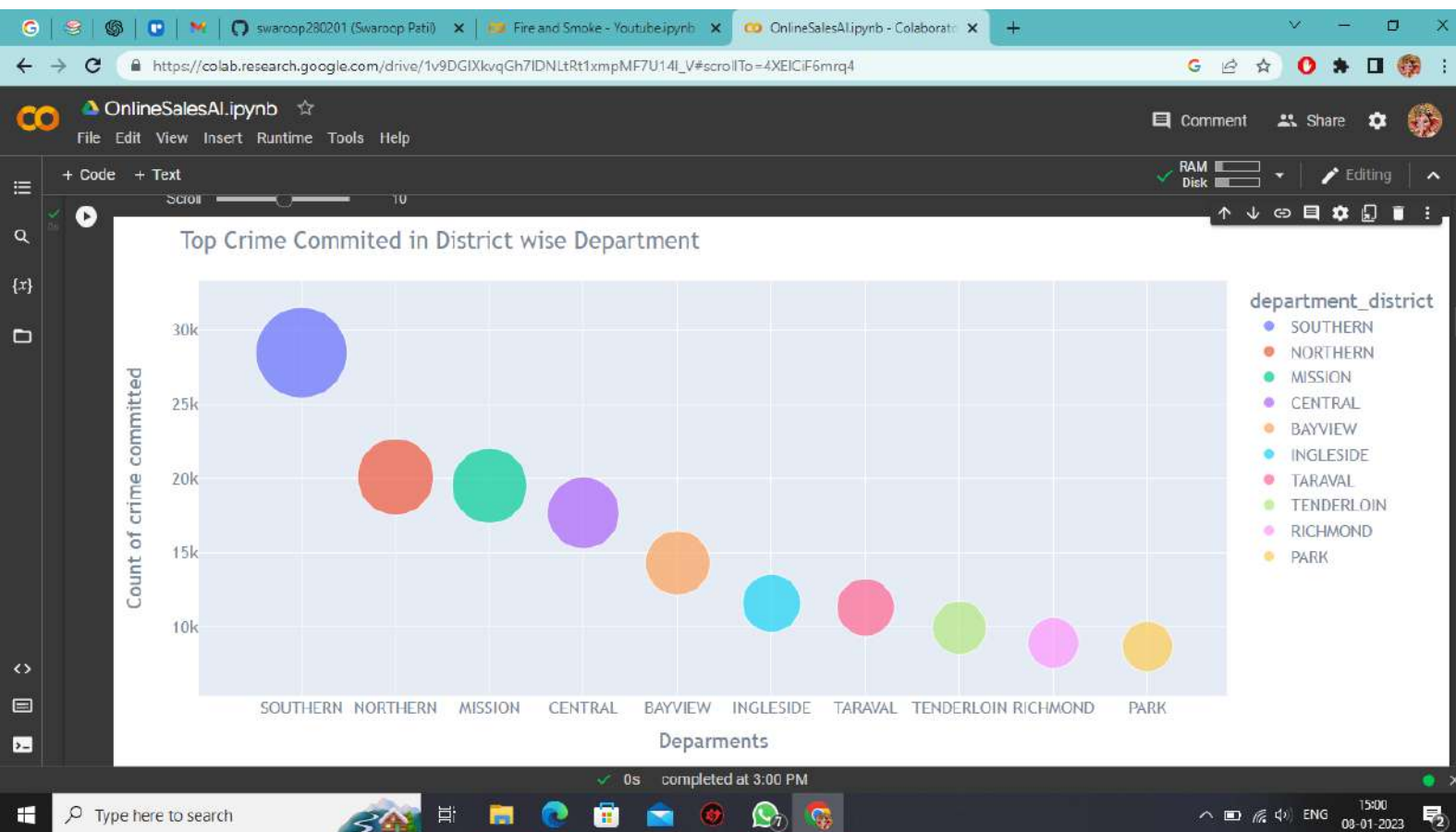
RAM  
Disk

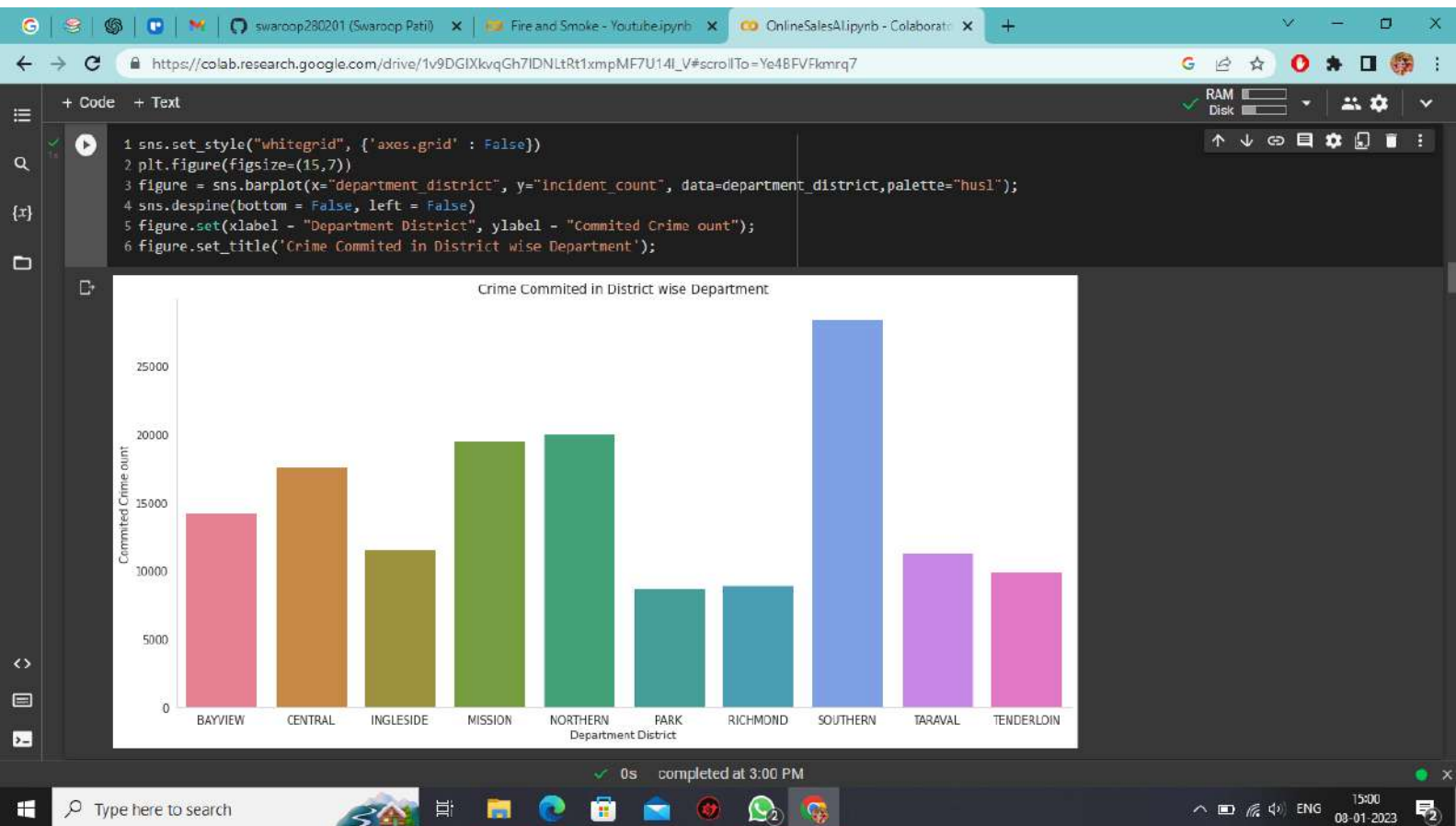
Editing

0s completed at 3:00 PM

Type here to search

15:00  
08-01-2023

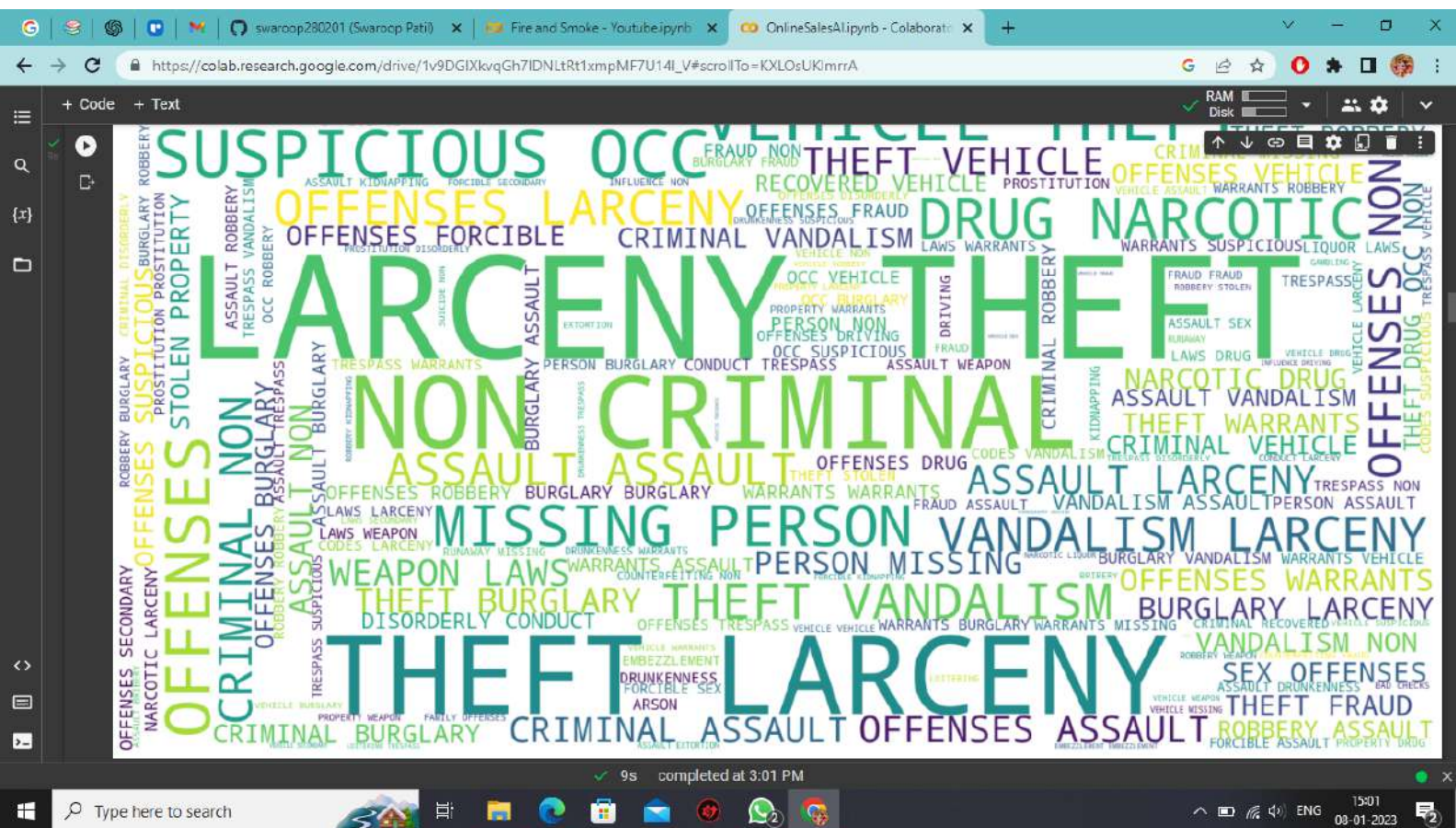


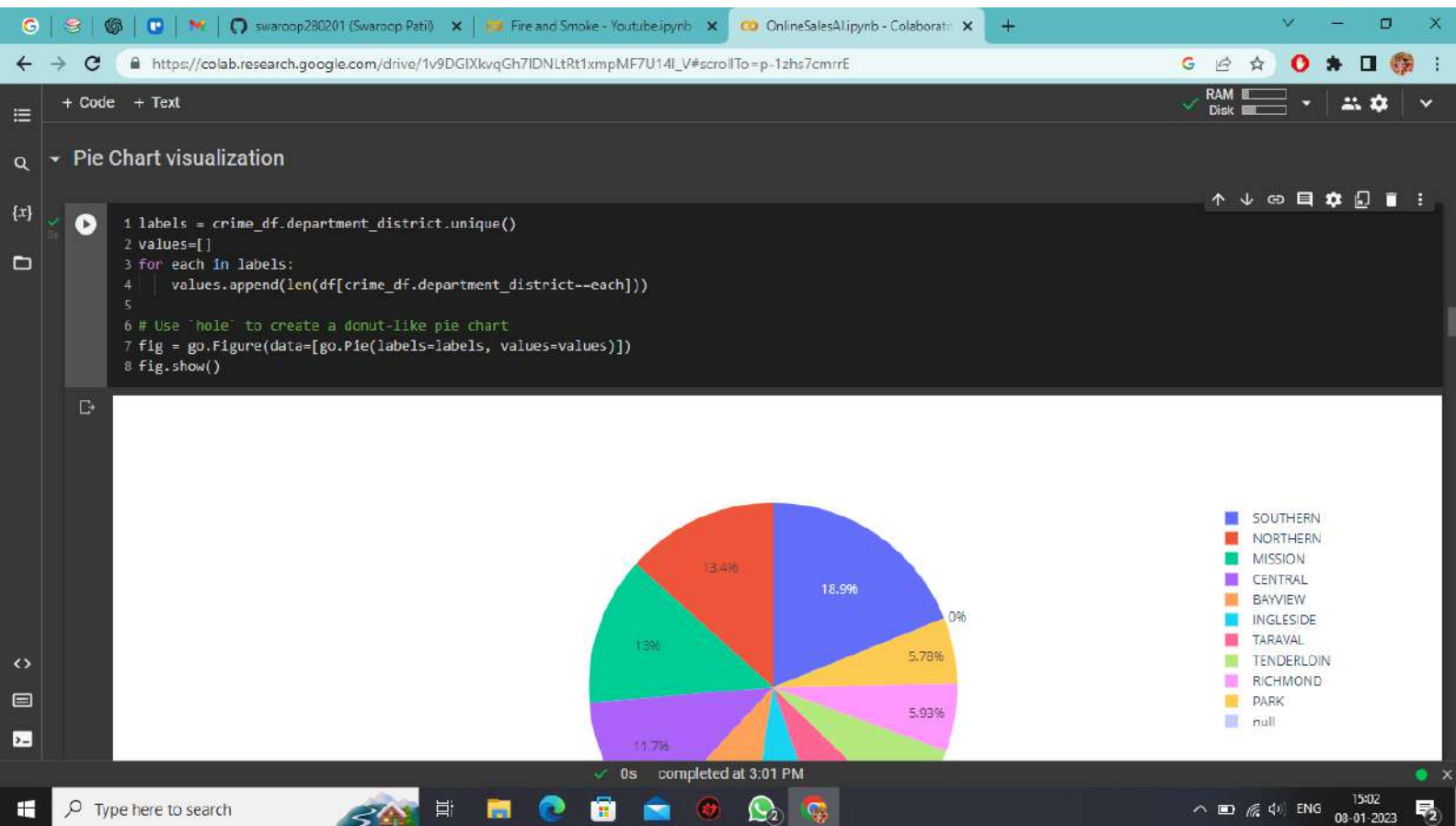


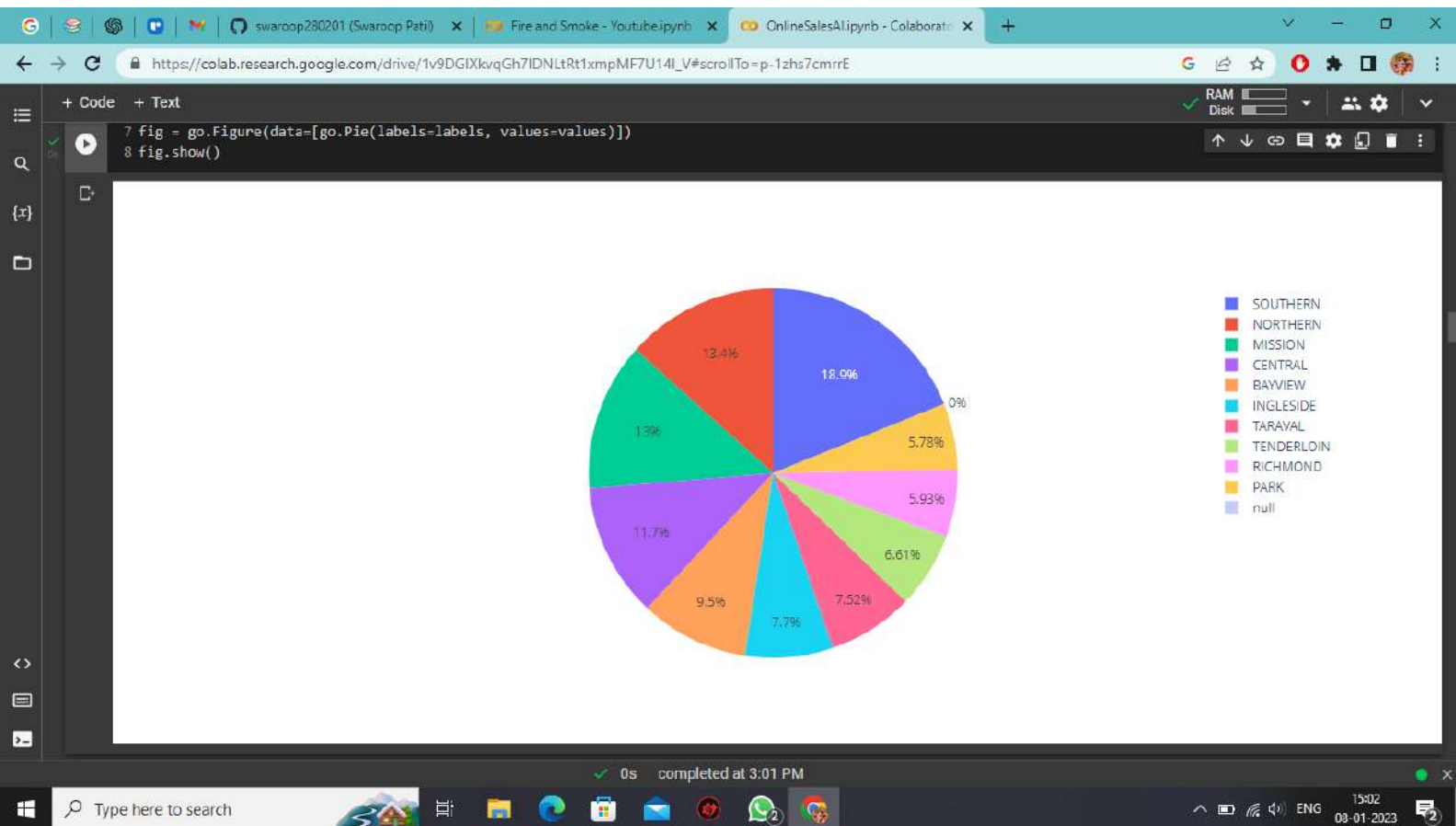


[illegible]









RAM

Disk

↑

↓

↻

🗨

⚙

📄

🗑

⋮

19

1

```
1 date_df = pd.DataFrame((crime_df.groupby(["crime_date"]).count()).reset_index())
2 date_df.set_index("crime_date",inplace=True)
3 date_df.head()
```

incident\_id category crime\_description department\_district resolution address department\_id lat long

crime\_date

2016-01-01 00:01:00	68	68	68	68	68	68	68	68	68
2016-01-01 00:05:00	3	3	3	3	3	3	3	3	3
2016-01-01 00:07:00	3	3	3	3	3	3	3	3	3
2016-01-01 00:09:00	2	2	2	2	2	2	2	2	2
2016-01-01 00:10:00	4	4	4	4	4	4	4	4	4

Analysis of data with respect to dates

0s

completed at 3:02 PM

Type here to search

15:02

08-01-2023

ENG

### Analysis of data with respect to dates

```
1 date_df = pd.DataFrame((crime_df.groupby(["crime_date"]).count()).reset_index())
2 date_df.set_index('crime_date',inplace=True)
3 date_df.head()
```

crime_date	incident_id	category	crime_description	department_district	resolution	address	department_id	lat	long
2016-01-01 00:01:00	68	68	68	68	68	68	68	68	68
2016-01-01 00:05:00	3	3	3	3	3	3	3	3	3
2016-01-01 00:07:00	3	3	3	3	3	3	3	3	3
2016-01-01 00:09:00	2	2	2	2	2	2	2	2	2
2016-01-01 00:10:00	4	4	4	4	4	4	4	4	4

### Quarterly analysis of reporting of crime and visualizing by bar chart

```
[ ] 1 date_df.index = pd.to_datetime(date_df.index,unit='ns',errors = 'coerce')
2 daily_df = date_df.resample('Q').mean()
3
4 daily_df.drop(['crime_description','category','resolution','address','department_id','lat','long','department_district'], axis =1, inplace =True)
5 daily_df = daily_df.rename(columns={'incident_id': 'crime_count'})
6
7 daily_df = daily_df.reset_index()
```

0s completed at 3:02 PM



swaroop280201 (Swaroop Pati) x Fire and Smoke - Youtube.ipynb x OnlineSalesAl.ipynb - Colaborat...

https://colab.research.google.com/drive/1v9DGIXkvqCh7IDNLTt1xmpMF7U14L\_V#scrollTo=63OzzwNtmrM

+ Code + Text

RAM Disk

[21] 2016-01-01 00:07:00 3 3 3 3 3 3 3 3 3  
2016-01-01 00:09:00 2 2 2 2 2 2 2 2 2  
2016-01-01 00:10:00 4 4 4 4 4 4 4 4 4

Quaterly analysis of reporting of crime and visualizing by bar chart

```
1 date_df.index = pd.to_datetime(date_df.index,unit='ns',errors = 'coerce')
2 daily_df = date_df.resample('Q').mean()
3
4 daily_df.drop(['crime_description','category','resolution','address','department_id','lat','long','department_district'], axis =1, inplace =True)
5 daily_df = daily_df.rename(columns={'incident_id': 'crime_count'})
6
7 daily_df = daily_df.reset_index()
8 daily_df = pd.DataFrame(daily_df)
9
10 daily_df
```

	crime_date	crime_count
0	2016-03-31	2.277155
1	2016-06-30	2.197702
2	2016-09-30	2.239021
3	2016-12-31	2.253111

0s completed at 3:02 PM

Type here to search

15:02 08-01-2023



Google Chrome browser window showing a Google Colab notebook titled "Quarterly Crimes analysis". The notebook is open to a code cell with the following Python code:

```
1 daily_df['month'] = pd.DateTimeIndex(daily_df['crime_date']).month
2
3
4 def mapper(month):
5     return month.strftime('%d/%b/%Y')
6
7 daily_df['crime_date'] = daily_df['crime_date'].apply(mapper)
8 daily_df
```

The code cell output shows a table with the following data:

	crime_date	crime_count	month
0	31/Mar/2016	2.277155	3
1	30/Jun/2016	2.197702	6
2	30/Sep/2016	2.239021	9
3	31/Dec/2016	2.253111	12

Below the table, there is a code cell with the following Python code:

```
[ ] 1 sns.set_style("whitegrid", {'axes.grid' : False})
2 plt.figure(figsize=(10,5))
3 figure = sns.barplot(x="crime_date", y="crime_count", data=daily_df,palette="husl");
4 figure.set_title('Quarterly Crimes analysis')
5 sns.despine(bottom = False, left = False)
6 figure.set(xlabel = "Quarterly Month", ylabel = "Average Committed Crime count");
```

The notebook interface shows a progress bar at the bottom indicating the execution status: "Quarterly Crimes analysis" completed at 3:02 PM.

Google Chrome browser window showing a Google Colab notebook titled "OnlineSalesAlipyrb - Colaboratory". The URL is [https://colab.research.google.com/drive/1v9DGIXkvqCh7IDNLTtRt1xmpMF7U14L\\_V#scrollTo=53hOHf3EmrW](https://colab.research.google.com/drive/1v9DGIXkvqCh7IDNLTtRt1xmpMF7U14L_V#scrollTo=53hOHf3EmrW).

The notebook interface shows a code cell with the following Python code:

```
1 sns.set_style("whitegrid", {'axes.grid' : False})
2 plt.figure(figsize=(10,5))
3 figure = sns.barplot(x="crime_date", y="crime_count", data=daily_df,palette="husl");
4 figure.set_title('Quarterly Crimes analysis')
5 sns.despine(bottom = False, left = False)
6 figure.set(xlabel = "Quarterly Month", ylabel = "Average Committed Crime count");
```

The output of the code is a bar chart titled "Quarterly Crimes analysis". The x-axis is labeled "Quarterly Month" and the y-axis is labeled "Average Committed Crime count". The chart displays four bars representing the average crime count for each quarter of 2016:

Quarterly Month	Average Committed Crime count
31/Mar/2016	~2.1
30/jun/2016	~2.0
30/Sep/2016	~2.0
31/Dec/2016	~2.0

Below the chart, a status bar indicates: "Week days analysis reporting of crime and visualizing by bar chart".

The bottom of the image shows the Windows taskbar with the search bar, task view button, and several application icons (File Explorer, Edge, Mail, etc.). The system clock shows 15:03 on 08-01-2023.

Google Chrome browser window showing a Google Colab notebook titled "Week days analysis reporting of crime and visualizing by bar chart".

The notebook contains the following code blocks:

```
1 daily_df['crime_date'] = pd.to_datetime(daily_df['crime_date'], errors='coerce')
2
```

Executing the code results in the following output:

crime_date	crime_count	month
Friday	2.239021	9.0
Saturday	2.253111	12.0
Thursday	2.237429	4.5

```
1 week_df = daily_df.groupby(daily_df['crime_date'].dt.day_name()).mean()
2 week_df
3
```

```
1 week_df = week_df.reset_index()
2 week_df = week_df.rename(columns={'crime_date': 'Day of Week'})
3 week_df.drop('month', axis=1, inplace=True)
```

The notebook interface shows a RAM and Disk usage indicator at the top right. The bottom status bar indicates the execution completed at 3:03 PM on 08-01-2023.

```
[28] 1 week_df[ 'crime_count' ].values  
array([2.23902144, 2.25311107, 2.23742867])
```

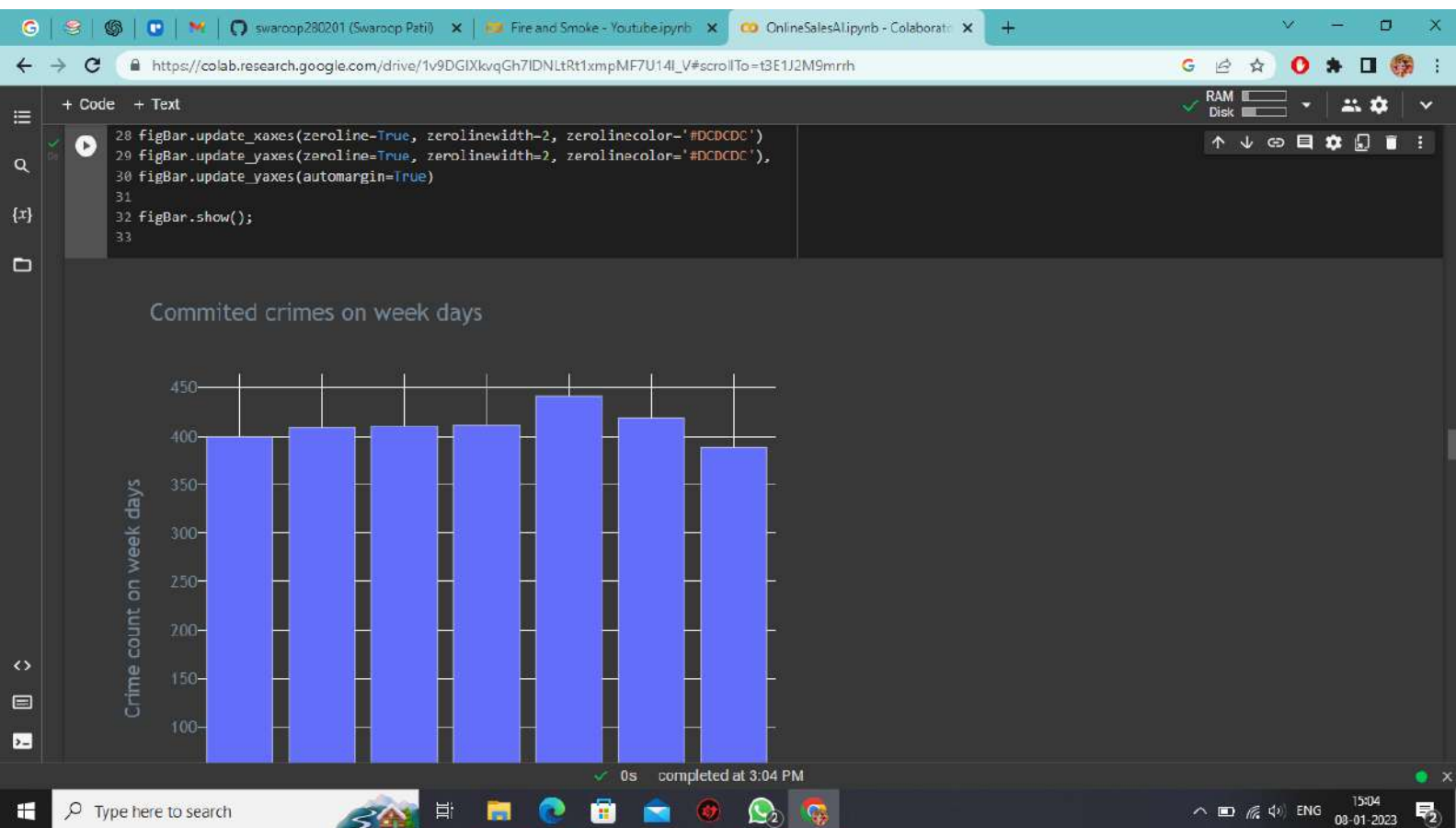
```
1 sns.set_style("whitegrid", {'axes.grid' : False})  
2 plt.figure(figsize=(10,5))  
3 figure = sns.barplot(x=['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday'],  
4 y=[399.673077,408.500000,410.230769,411.442308,440.962264,418.339623,388.557692],  
5 data=week_df,  
6 palette="husl");  
7 sns.despine(bottom = False, left = False)  
8 figure.set_title('Committed Crimes on week days')  
9 figure.set(xlabel = "Week Days", ylabel = "Average count of Committed Crime in week Days ");
```



Google Chrome browser window showing a Google Colab notebook. The address bar displays the URL: [https://colab.research.google.com/drive/1v9DGIXkvqCh7IDNLTtRt1xmpMF7U14L\\_V#scrollTo=t3E1J2M9mrh](https://colab.research.google.com/drive/1v9DGIXkvqCh7IDNLTtRt1xmpMF7U14L_V#scrollTo=t3E1J2M9mrh). The notebook interface shows a code cell with the following Python code:

```
1 figBar = go.Figure(  
2     data=[  
3         go.Bar(  
4             name="Category",  
5             x= ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday'],  
6             y= [399.673077, 408.500000, 410.230769, 411.442368, 440.962264, 418.339623, 388.557692],  
7             #marker=dict(color='#0000FF')  
8         ),  
9     ],  
10     layout=go.Layout(  
11         title="Committed crimes on week days",  
12         yaxis_title="Crime count on week days",  
13         xaxis_title="Week Days",  
14         font=dict(family="Trebuchet MS", size=16, color="#778899"),  
15         xaxis_showgrid=True,  
16         yaxis_showgrid=True,  
17         width=700,  
18         height=600,  
19         paper_bgcolor='rgba(0,0,0,0)',  
20         plot_bgcolor='rgba(0,0,0,0)'  
21     )  
22 )  
23  
24 figBar.update_xaxes(zeroLine=True, zeroLineWidth=2, zeroLineColor='DCDCDC')  
25 figBar.update_yaxes(zeroLine=True, zeroLineWidth=2, zeroLineColor='DCDCDC'),  
26 figBar.update_yaxes(automargin=True)
```

The code cell status bar indicates "0s completed at 3:04 PM". The Windows taskbar at the bottom shows the search bar and various application icons. The system clock in the bottom right corner displays "15:04 00-01-2023".







swaroop280201 (Swaroop Pati) x Fire and Smoke - Youtube.ipynb x OnlineSalesAl.ipynb - Colaborat...

https://colab.research.google.com/drive/1v9DGIXkvqCh7IDNLTtRt1xmpMF7U14L\_V#scrollTo=TG-J1fHLmrrm

RAM Disk

Monthly analysis reporting of crime and visualizing by bubble chart and bar chart

```
1 date_df.index = pd.to_datetime(date_df.index)
2 Monthly_df = date_df.resample('M').sum()
3
4 Monthly_df.drop(['crime_description', 'category', 'resolution', 'address', 'department_id', 'lat', 'long', 'department_district'], axis=1, inplace=True)
5 Monthly_df = Monthly_df.rename(columns={'incident_id': 'crime_count'})
6
7 Monthly_df = Monthly_df.reset_index()
8 Monthly_df = pd.DataFrame(Monthly_df)
9
```

Extracting the Months from crime date

```
1 Monthly_df['month'] = pd.DatetimeIndex(Monthly_df['crime_date']).month
2 Monthly_df['month']
3
4 def mapper(month):
5     return month.strftime('%b')
6 Monthly_df['crime_date'] = Monthly_df['crime_date'].apply(mapper)
7 Monthly_df
```

	crime_date	crime_count	month
0	Jan	12946	1
1	Feb	12092	2
2	Mar	12362	3

0s completed at 3:04 PM

Type here to search

15:04 08-01-2023

Google Chrome browser window showing a Google Colab notebook. The address bar displays the URL: [https://colab.research.google.com/drive/1v9DGIXkvqGh7IDNLTt1xmpMF7U14L\\_V#scrollTo=TG-J1fHLmrrm](https://colab.research.google.com/drive/1v9DGIXkvqGh7IDNLTt1xmpMF7U14L_V#scrollTo=TG-J1fHLmrrm). The notebook interface shows a code cell with the following Python code:

```
1 Monthly_df['month'] = pd.DatetimeIndex(Monthly_df['crime_date']).month
2 Monthly_df['month']
3
4 def mapper(month):
5     return month.strftime('%b')
6 Monthly_df['crime_date'] = Monthly_df['crime_date'].apply(mapper)
7 Monthly_df
```

The output of the code cell is a table with 12 rows, indexed 0 to 11, showing the crime\_date, crime\_count, and month for each month of the year:

	crime_date	crime_count	month
0	Jan	12946	1
1	Feb	12092	2
2	Mar	12362	3
3	Apr	12317	4
4	May	12713	5
5	Jun	12076	6
6	Jul	12166	7
7	Aug	12428	8
8	Sep	12473	9
9	Oct	13331	10
10	Nov	12670	11
11	Dec	12926	12

The bottom status bar indicates the code was completed at 3:04 PM. The Windows taskbar at the bottom shows the search bar, task view, and various application icons. The system clock shows 15:04 on 08-01-2023.

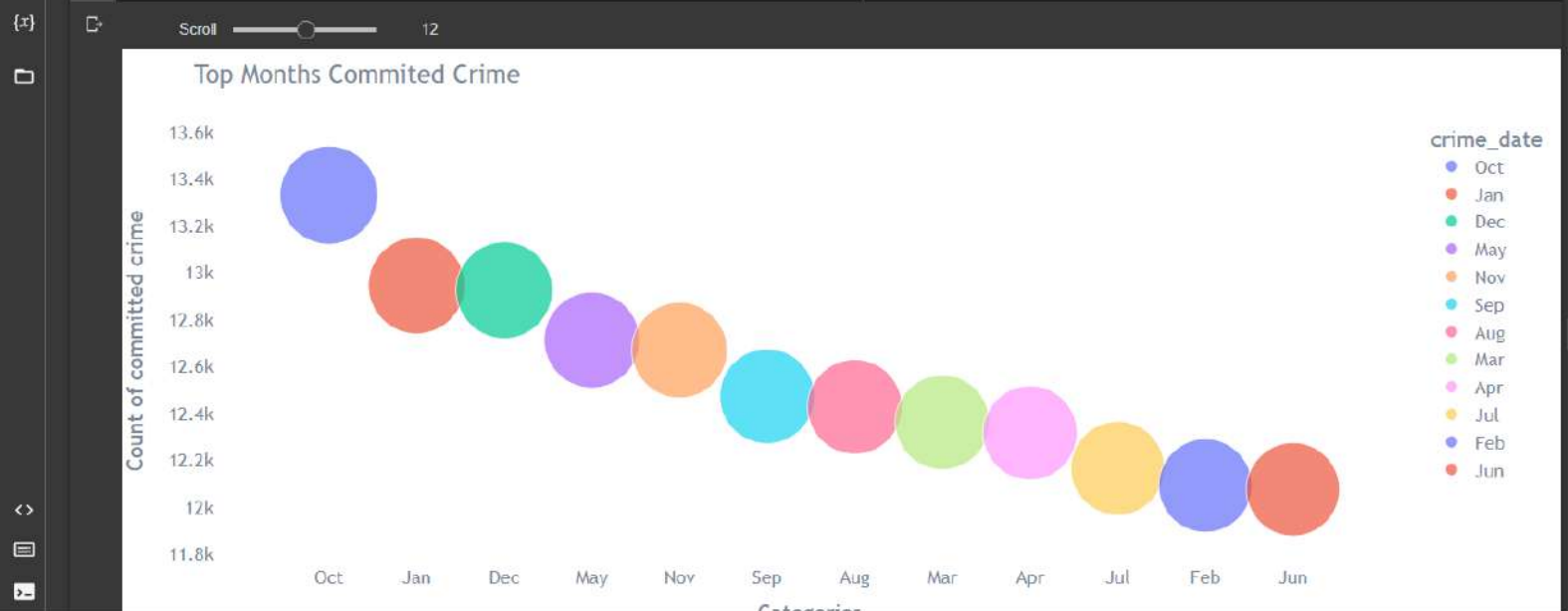
Google Chrome browser window showing a Google Colab notebook. The address bar displays the URL: [https://colab.research.google.com/drive/1v9DGIXkvqCh7IDNltRt1xmpMF7U14L\\_V#scrollTo=XBwiAciymrrn](https://colab.research.google.com/drive/1v9DGIXkvqCh7IDNltRt1xmpMF7U14L_V#scrollTo=XBwiAciymrrn). The notebook interface includes a toolbar with icons for code, text, and execution. The code editor shows the following Python code:

```
1 Monthly_df_sorted = Monthly_df.sort_values('crime_count',ascending=False)
2 ipywlayout = widgets.Layout(border='solid 2px green')
3
4 def bubble_chart(Scroll):
5
6     figBubble = px.scatter(
7         Monthly_df_sorted.head(Scroll),
8         x = 'crime_date',
9         y = 'crime_count',
10        size= 'crime_count',
11        color = 'crime_date',
12        hover_name='crime_date' ,
13        size_max=60
14    )
15    figBubble.update_layout(
16
17        title="Top Months Committed Crime",
18        yaxis_title="Count of committed crime",
19        xaxis_title="Categories",
20        font=dict(family="Trebuchet MS",size=16,color="#778899"),
21        xaxis_showgrid=True,
22        yaxis_showgrid=True,
23        plot_bgcolor='rgba(0,0,0,0)',
24
25    )
26
27
28    figBubble.update_xaxes(zeroLine=True, zerolinewidth=2, zerolinecolor='#D8D8D8')
29    figBubble.update_yaxes(zeroLine=True, zerolinewidth=2, zerolinecolor='#D8D8D8')
30
```

The code defines a function `bubble_chart` that takes a `Scroll` parameter. It uses `px.scatter` to create a bubble chart from the top `Scroll` rows of `Monthly_df_sorted`. The x-axis is labeled 'crime\_date', the y-axis is labeled 'crime\_count', and the bubble size is proportional to 'crime\_count'. The chart is styled with a 'Trebuchet MS' font, size 16, color '#778899', and a white background. The chart is updated with a title 'Top Months Committed Crime', y-axis title 'Count of committed crime', and x-axis title 'Categories'. The chart is also updated with a zero line and a grid.

The bottom status bar shows the execution time: 0s, completed at 3:04 PM. The Windows taskbar at the bottom displays the search bar, task view, and various application icons. The system clock shows the date 08-01-2023 and time 15:05.

```
+ Code + Text  
31 figBubble.show();  
32  
33 interact(bubble_chart, Scroll=12);
```

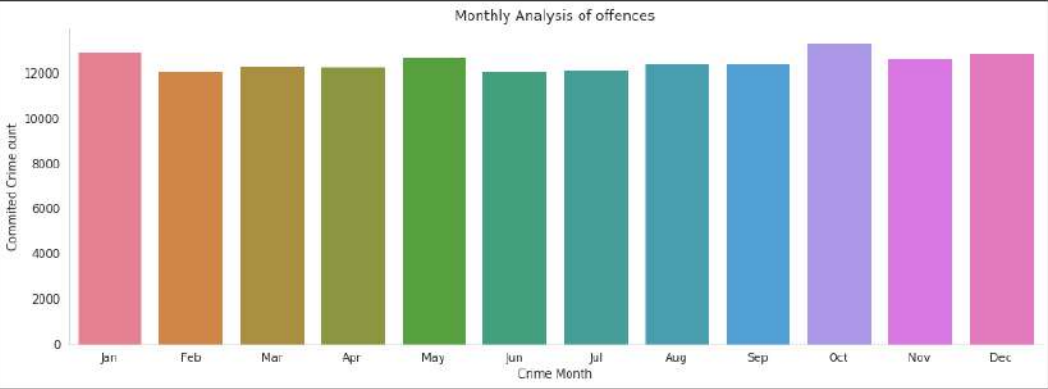


Google Chrome browser window showing a Google Colab notebook titled "OnlineSalesAlipyrb - Colaboratory". The URL is [https://colab.research.google.com/drive/1v9DGIXkvqCh7IDNLTtRt1xmpMF7U14L\\_V#scrollTo=gigMw6c8mrrp](https://colab.research.google.com/drive/1v9DGIXkvqCh7IDNLTtRt1xmpMF7U14L_V#scrollTo=gigMw6c8mrrp).

The notebook interface shows a code cell with the following Python code:

```
1 sns.set_style("whitegrid", {'axes.grid' : False})
2 plt.figure(figsize=(15,5))
3 figure = sns.barplot(x="crime_date", y="crime_count", data=Monthly_df,palette="husl");
4 sns.despine(bottom = False, left = False)
5 figure.set_title('Monthly Analysis of offences')
6 figure.set(xlabel = "Crime Month", ylabel = "Committed Crime count");
```

The output of the code is a bar chart titled "Monthly Analysis of offences". The x-axis is labeled "Crime Month" and the y-axis is labeled "Committed Crime count". The chart displays the crime count for each month from January to December. The bars are colored using the "husl" palette.



Crime Month	Committed Crime count
Jan	12500
Feb	12000
Mar	12200
Apr	12200
May	12500
Jun	12000
Jul	12000
Aug	12200
Sep	12200
Oct	12800
Nov	12500
Dec	12500

Below the chart, a text box contains the instruction: "Grouping data into categories, department, & resolution inorder to find the pattern of type of offences".

The bottom status bar shows the execution time as "0s" and "completed at 3:06 PM". The Windows taskbar at the bottom indicates the date as "08-01-2023" and the time as "15:06".



RAM

Disk

+

Code

+

Text

Grouping data into categories, department, & resolution inorder to find the pattern of type of offences committed in each district

[38]

```
1 crime_count = crime_df.drop(['incident_id', 'crime_description', 'crime_date', 'address', 'department_id', 'lat'], axis=1)
2 crime_count = crime_count.groupby(['category', 'department_district', 'resolution']).count()
3 crime_count = crime_count.rename(columns={'long': 'count'})
4 crime_count = crime_count.reset_index()
```

Visualisind the data by bar plot

0s

```
1 menu_district_dp = np.array(crime_df['department_district'].values.tolist())
2 menu_district_dp = np.unique(menu_district_dp)
3
4 districtCrimeBarWidget = widgets.Dropdown(options=menu_district_dp,
5     value='BAYVIEW',
6     description='<p style = "font-family:Trebuchet MS;font-size:15px;font-style:bold;font-color:#999999;"><b>District</b></p> :</br></br>',
7     disabled=False
8 )
9
10 def district(district_dep):
11     crime_count
12
13     disFig = go.Figure(
14         data=[
15             go.Bar(
16                 name="District",
```

Type here to search

15:06

08-01-2023

Google Chrome browser window showing a Google Colab notebook. The address bar displays the URL: [https://colab.research.google.com/drive/1v9DGIXkvqGh7IDNLTt1xmpMF7U14L\\_V#scrollTo=VAIQdRQpmrr0](https://colab.research.google.com/drive/1v9DGIXkvqGh7IDNLTt1xmpMF7U14L_V#scrollTo=VAIQdRQpmrr0). The notebook interface shows a code editor with the following Python code:

```
14 data=[
15     go.Bar(
16         name="District",
17         x= crime_count['category'].unique(),
18         y= crime_count['count'],
19         marker=dict(color='#0000FF')
20     ),
21 ],
22 layout=go.Layout(
23     title="Crimes in District",
24     yaxis_title="Count",
25     xaxis_title="District",
26     font=dict(family="Trebuchet MS",size=16,color="#778899"),
27     xaxis_showgrid=True,
28     yaxis_showgrid=True,
29     width=700,
30     height=600,
31     paper_bgcolor='rgba(0,0,0,0)',
32     plot_bgcolor='rgba(0,0,0,0)'
33 )
34 )
35 disFig.update_xaxes(zeroline=True, zerolinewidth=2, zerolinecolor='#DCDCDC')
36 disFig.update_yaxes(zeroline=True, zerolinewidth=2, zerolinecolor='#DCDCDC'),
37 disFig.update_yaxes(automargin=True)
38
39 disFig.show();
40
41
42
43 widgets.interact(district, district_dep = districtCrimeBarWidget);
```

The code defines a bar chart titled "Crimes in District" showing the count of crimes for different districts. The x-axis is labeled "District" and the y-axis is labeled "Count". The chart uses a blue color for the bars and a white background. The plot area has a black background. The chart is displayed using the `disFig.show()` method.

The bottom status bar indicates the code was completed at 3:06 PM. The Windows taskbar at the bottom shows the search bar, task view button, and several application icons. The system clock shows 15:07 on 08-01-2023.



Visualizing maximum number of which type of crime committing in district wise

Creating dataframe of resolution and department to check the efficiency of department

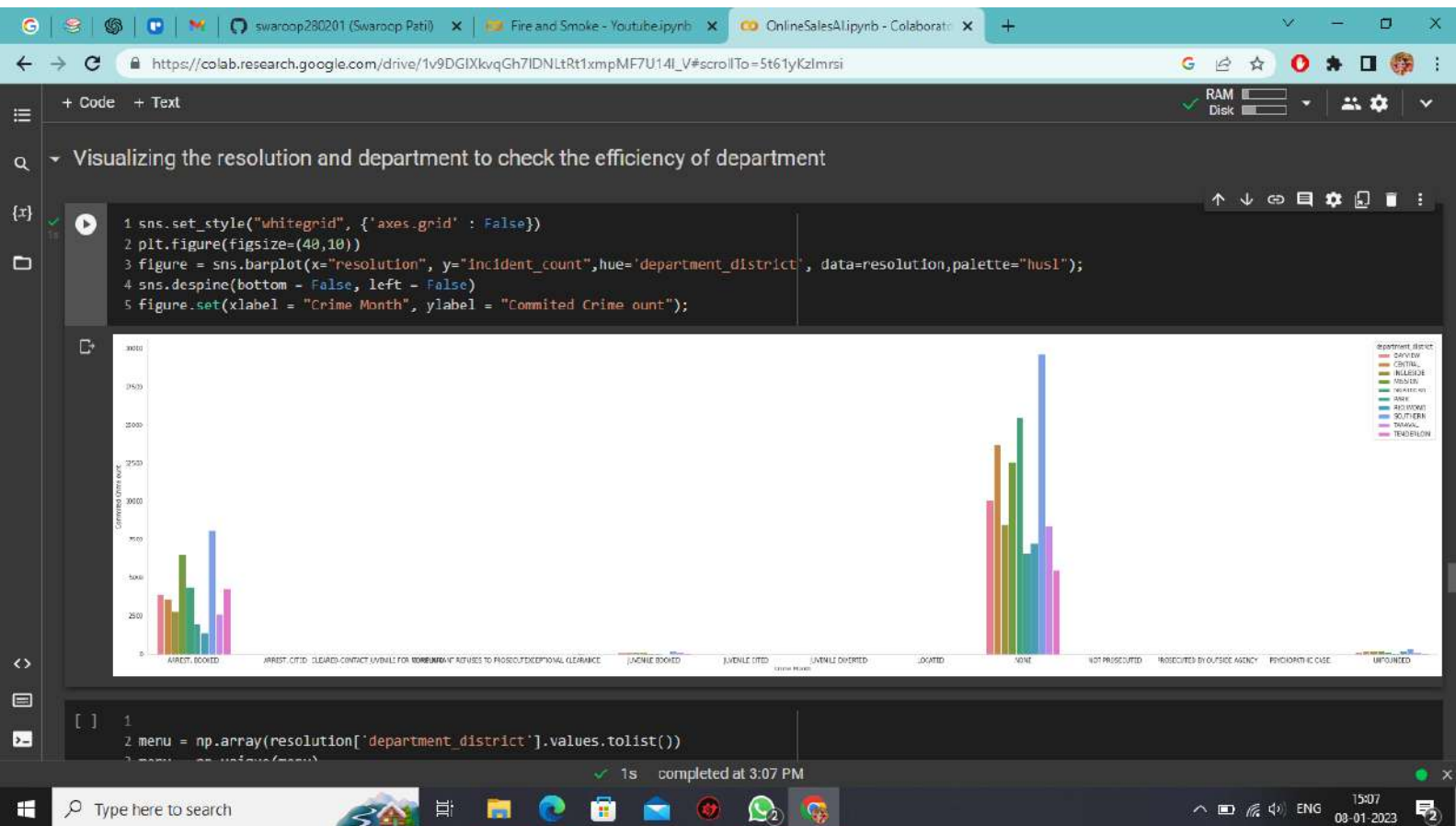
```
1 resolution = pd.DataFrame((crime_df.groupby( [ "resolution",'department_district'] ).count()).reset_index())
2 resolution.drop(['crime_description','crime_date','address','department_id','category'], axis =1, inplace =True)
3 resolution = resolution.rename(columns={'incident_id': 'incident_count'})
4 resolution.head()
```

	resolution	department_district	incident_count	lat	long
0	ARREST, BOOKED	BAYVIEW	3890	3890	3890
1	ARREST, BOOKED	CENTRAL	3572	3572	3572
2	ARREST, BOOKED	INGLESIDE	2755	2755	2755
3	ARREST, BOOKED	MISSION	6531	6531	6531
4	ARREST, BOOKED	NORTHERN	4352	4352	4352

Visualizing the resolution and department to check the efficiency of department

```
[ ] 1 sns.set_style("whitegrid", {'axes.grid' : False})
2 plt.figure(figsize=(40,10))
3 figure = sns.barplot(x="resolution", y="incident_count", hue="department_district", data=resolution, palette="husl");
4 sns.despine(bottom = False, left = False)
```

0s completed at 3:07 PM



Google Chrome browser window showing a Google Colab notebook. The address bar displays the URL: [https://colab.research.google.com/drive/1v9DGIXkvqCh7IDNltRt1xmpMF7U14L\\_V#scrollTo=Wzi0IL4\\_mrsI](https://colab.research.google.com/drive/1v9DGIXkvqCh7IDNltRt1xmpMF7U14L_V#scrollTo=Wzi0IL4_mrsI). The notebook interface shows a code editor with the following Python code:

```
1 menu = np.array(resolution['department_district'].values.tolist())
2 menu = np.unique(menu)
3
4
5 departmentdistrictBar = widgets.Dropdown(options=menu,
6     value='CENTRAL',
7     description='<p style = "font-family:Trebuchet MS;font-size:15px;font-style:bold;font-color:#999999;"><b>District</b></p> :</br></br>',
8     disabled=False
9 )
10
11 def department_district(department_district):
12     department_district = resolution.loc[resolution['department_district'] == department_district]
13
14 figBar = go.Figure(
15     data=[
16         go.Bar(
17             name="Resolution",
18             x= department_district['resolution'].values,
19             y= department_district['incident_count'].values,
20             marker=dict(color='#CB33FF')
21         ),
22     ],
23
24
25 ],
26 layout=go.Layout(
27     title="Efficiency of the department to resolve issues",
28     yaxis_title="count",
29     xaxis_title="offences category",
30     font=dict(family="Trebuchet MS",size=16,color="#778899"),
31     xaxis_showgrid=True,
```

The code is executed, and the status bar at the bottom indicates "0s completed at 3:07 PM". The Windows taskbar is visible at the bottom of the screen, showing the search bar and various application icons.



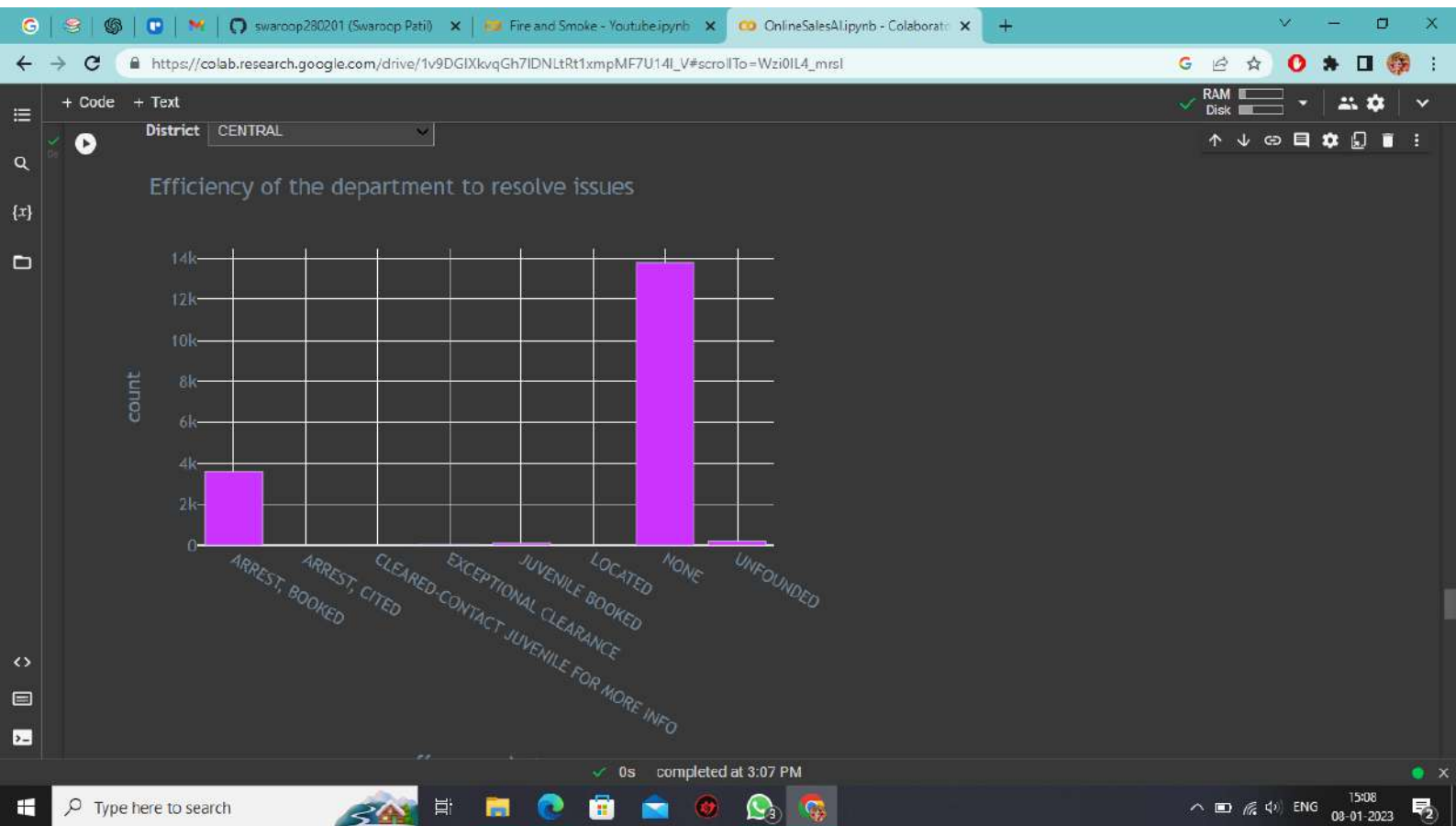
Google Chrome browser window showing a Google Colab notebook titled "OnlineSalesAl.ipynb - Colaboratory". The URL is [https://colab.research.google.com/drive/1v9DGIXkvqCh7IDNLTtRt1xmpMF7U14L\\_V#scrollTo=Wzi0IL4\\_mrsI](https://colab.research.google.com/drive/1v9DGIXkvqCh7IDNLTtRt1xmpMF7U14L_V#scrollTo=Wzi0IL4_mrsI).

The code editor shows the following Python code:

```
32     yaxis_showgrid=True,
33     width=700,
34     height=600,
35     paper_bgcolor='rgba(0,0,0,0)',
36     plot_bgcolor='rgba(0,0,0,0)'
37 )
38
39 )
40
41 figBar.update_xaxes(zero=True, zerolinewidth=2, zerolinecolor='#DCDCDC')
42 figBar.update_yaxes(zero=True, zerolinewidth=2, zerolinecolor='#DCDCDC'),
43 figBar.update_yaxes(automargin=True)
44
45 figBar.show();
46
47 interact(department_district, department_district = departmentdistrictBar):
```

Below the code editor, there is a dropdown menu labeled "District" with "CENTRAL" selected. The plot area displays the title "Efficiency of the department to resolve issues". The y-axis is labeled "int" and ranges from 8k to 14k. A single blue bar is visible, representing the efficiency for the "CENTRAL" district.

The status bar at the bottom indicates "0s completed at 3:07 PM". The Windows taskbar at the bottom shows the search bar, task view, and various application icons. The system clock shows 15:08 on 09-01-2023.



swaroop280201 (Swaroop Pati) x Fire and Smoke - Youtube.ipynb x OnlineSalesAl.ipynb - Colaborat...

https://colab.research.google.com/drive/1v9DGIXkvqCh7IDNLTt1xmpMF7U14L\_V#scrollTo=LqRMt5a0mrso

RAM Disk

Creating dataframe for location using lat long

1 location\_df = crime\_df[['category','crime\_date','crime\_description','department\_district','lat','long']]  
2 location\_df = pd.concat([location\_df,category\_df], axis=1)  
3 location\_df

	category	crime_date	crime_description	department_district	lat	long	category	incident_count
0	WEAPON LAWS	2016-01-29 11:00:00	POSS OF PROHIBITED WEAPON	SOUTHERN	37.775420706711	-122.403404791479	ARSON	286.0
1	WEAPON LAWS	2016-01-29 11:00:00	FIREARM, LOADED, IN VEHICLE, POSSESSION OR USE	SOUTHERN	37.775420706711	-122.403404791479	ASSAULT	13577.0
2	WARRANTS	2016-04-25 14:59:00	WARRANT ARREST	BAYVIEW	37.7299809672996	-122.368856204292	BAD CHECKS	34.0
3	NON-CRIMINAL	2016-01-05 23:50:00	LOST PROPERTY	TENDERLOIN	37.7857883766888	-122.412970537591	BRIBERY	66.0
4	NON-CRIMINAL	2016-01-01 00:30:00	LOST PROPERTY	MISSION	37.7650501214668	-122.415671780295	BURGLARY	5802.0
...	...	...	...	...	...	...	...	...
150495	ASSAULT	2016-12-30 21:01:00	BATTERY	PARK	37.7714278595913	-122.453981622365	NaN	NaN
150496	NON-CRIMINAL	2016-12-30 08:00:00	LOST PROPERTY	CENTRAL	37.7966261239618	-122.401857374739	NaN	NaN
150497	LARCENY/THEFT	2016-12-29 20:00:00	PETTY THEFT OF PROPERTY	CENTRAL	37.7906727649886	-122.41226909105	NaN	NaN

0s completed at 3:08 PM

Type here to search

15:08 09-01-2023

swaroop280201 (Swaroop Pati) xFire and Smoke - Youtube.ipynb xOnlineSalesAl.ipynb - Colaborat...

https://colab.research.google.com/drive/1v9DGIXkvqCh7IDNltRt1xmpMF7U14L\_V#scrollTo=LqRMt5a0mrse

RAMDisk

+ Code + Text

categorycrime\_datecrime\_descriptiondepartment\_districtlatlongcategoryattachment\_download

0WEAPON LAWS2016-01-29 11:00:00POSS OF PROHIBITED WEAPONSOUTHERN37.775420706711-122.403404791479ARSON286.0

1WEAPON LAWS2016-01-29 11:00:00FIREARM, LOADED, IN VEHICLE, POSSESSION OR USESOUTHERN37.775420706711-122.403404791479ASSAULT13577.0

2WARRANTS2016-04-25 14:59:00WARRANT ARRESTBAYVIEW37.7299809672996-122.388856204292BAD CHECKS34.0

3NON-CRIMINAL2016-01-05 23:50:00LOST PROPERTYTENDERLOIN37.7857883766888-122.412970537591BRIBERY66.0

4NON-CRIMINAL2016-01-01 00:30:00LOST PROPERTYMISSION37.7650501214668-122.419671780295BURGLARY5802.0

...

150495ASSAULT2016-12-30 21:01:00BATTERYPARK37.7714278595913-122.453981622365NaNNaN

150496NON-CRIMINAL2016-12-30 08:00:00LOST PROPERTYCENTRAL37.7966261239618-122.401857374739NaNNaN

150497LARCENY/THEFT2016-12-29 20:00:00PETTY THEFT OF PROPERTYCENTRAL37.7906727649886-122.41226909106NaNNaN

150498LARCENY/THEFT2016-12-30 10:00:00GRAND THEFT OF PROPERTYCENTRAL37.7882745285785-122.406658711005NaNNaN

150499OTHER OFFENSES2016-12-02 14:00:00FRAUDULENT GAME OR TRICK, OBTAINING MONEY OR P...SOUTHERN37.775420706711-122.403404791479NaNNaN

150500 rows x 8 columns

0s completed at 3:08 PM

Type here to search

15:0808-01-2023

Google Chrome browser window showing a Google Colab notebook titled "Plotting the lat long and vializing the offences". The notebook contains the following code:

```
1 district_crime_count = crime_count.groupby(['department_district']).count().sort_values('count', ascending=False)
2 district_crime_count = district_crime_count.drop(['category', 'resolution'], axis=1)
3 district_crime_count = district_crime_count.reset_index()
4 district_crime_count
```

The output of the code is a DataFrame with the following data:

	department_district	count
0	MISSION	141
1	SOUTHERN	141
2	BAYVIEW	124
3	INGLESIDE	124
4	NORTHERN	121
5	TARAVAL	110
6	TENDERLOIN	105
7	RICHMOND	102
8	CENTRAL	100
9	PARK	99

The bottom of the image shows the Windows taskbar with the search bar, task view button, and several application icons (File Explorer, Edge, Task View, Mail, WhatsApp, and a game icon). The system tray shows the date and time: 15:08, 08-01-2023.

Google Chrome browser window showing a Google Colab notebook. The address bar displays the URL: [https://colab.research.google.com/drive/1v9DGIXkvqGh7IDNLTtRt1xmpMF7U14L\\_V#scrollTo=EyNl6mWqmrst](https://colab.research.google.com/drive/1v9DGIXkvqGh7IDNLTtRt1xmpMF7U14L_V#scrollTo=EyNl6mWqmrst). The notebook interface includes a left sidebar with icons for file management and a top toolbar with options like '+ Code', '+ Text', and RAM/Disk usage indicators.

The code in the notebook is as follows:

```
1 # Set CSS properties for th elements in dataframe
2 th_props = [
3     ('font-size', '20px'),
4     ('text-align', 'center'),
5     ('font-weight', 'bold'),
6     ('color', '#999999'),
7     ('background-color', '#f7f7f9')
8 ]
9
10
11 # Set table styles
12 styles = [
13     dict(selector="th", props=th_props)
14 ]
15
16
17
18 # Fetching the number of desired rows using interaction.
19
20 fig = go.FigureWidget( layout=go.Layout() )
21
22 def highlight_col(x):
23
24
25     department_district = 'background-color: #99ebff'
26     count = 'background-color: #99ff99'
27
28
29
30 df1 = pd.DataFrame('', index=x.index, columns=x.columns)
```

The bottom status bar indicates the execution completed at 3:08 PM.



Google Chrome browser window showing a Google Colab notebook. The address bar displays the URL: [https://colab.research.google.com/drive/1v9DGIXkvqCh7IDNLTtRt1xmpMF7U14L\\_V#scrollTo=EyNl6mWqmrst](https://colab.research.google.com/drive/1v9DGIXkvqCh7IDNLTtRt1xmpMF7U14L_V#scrollTo=EyNl6mWqmrst). The notebook interface includes a toolbar with icons for code, text, and execution. The code editor shows the following Python code:

```
31 df1.iloc[:, 0] = department_district
32 df1.iloc[:, 1] = count
33
34
35
36
37 return df1
38
39 def show_Details(Number):
40     n = int(Number)
41     pd.options.display.float_format = '{:,.2f}'.format
42     s = district_crime_count.head(n).style.apply(highlight_col, axis=None)
43     s = s.hide_index()
44     s.set_precision(20)
45     s.set_table_styles(styles)
46     s.set_properties(**{'border-style': 'solid',
47                         'border-color': '#e0e0eb',
48                         'border-width': '0.015px',
49                         'color': 'black',
50                         'font-family': 'Trebuchet MS',
51                         'font-weight': 'bold',
52                         'text-align': 'center',
53                         'width': '119.5px'
54                     })
55     #s = s.set_caption('<h3>Top Worst Hit State</h3><br>')
56     pd.options.display.float_format = '{:,.2f}'.format
57     pd.option_context('display.colheader_justify', 'center')
58     return s;
59
60
61 interact(show_Details, Number='10');
62
```

The code defines a function `show_Details` that takes a number `Number` and returns a styled pandas DataFrame. The function uses `pd.options.display.float_format` to format floating-point numbers, `pd.set_precision` to set the precision of the display, and `pd.set_table_styles` to apply custom styles to the table. The styles include a solid border, a specific border color and width, black text, and a specific font family and weight. The function also uses `pd.option_context` to justify the column headers to the center. The code is executed in a Jupyter notebook environment, and the output is displayed in the right-hand pane.

The bottom status bar indicates the execution completed at 3:08 PM. The Windows taskbar at the bottom shows the search bar, task view, and various application icons, including File Explorer, Microsoft Edge, and the Windows Store. The system clock shows the time as 15:09 on 08-01-2023.

Google Chrome browser window showing a Google Colab notebook. The address bar displays the URL: [https://colab.research.google.com/drive/1v9DGIXkvqGh7IDNLTtRt1xmpMF7U14L\\_V#scrollTo=EyNl6mWqmrst](https://colab.research.google.com/drive/1v9DGIXkvqGh7IDNLTtRt1xmpMF7U14L_V#scrollTo=EyNl6mWqmrst).

The notebook interface shows a code cell with the following Python code:

```
[ ] 1 lat_long = crime_df[['department_district', 'lat', 'long']].drop_duplicates(['department_district'])
    2 lat_long.reset_index(drop=True, inplace=True)
    3 lat_long
```

The output of the code cell is a table with 10 rows and 2 columns: department\_district and count.

department_district	count
MISSION	141
SOUTHERN	141
BAYVIEW	124
INGLESIDE	124
NORTHERN	121
TARAVAL	110
TENDERLOIN	105
RICHMOND	102
CENTRAL	100
PARK	99

Below the table, a preview of the 'lat\_long' DataFrame is shown:

	department_district	lat	long
0	SOUTHERN	37.775420706711	-122.403404791479

The bottom status bar indicates the code execution completed at 3:08 PM.

Google Chrome browser window showing a Google Colab notebook. The address bar displays the URL: [https://colab.research.google.com/drive/1v9DGIXkvqGh7IDNltRt1xmpMF7U14L\\_V#scrollTo=5AFCnhZdmrsw](https://colab.research.google.com/drive/1v9DGIXkvqGh7IDNltRt1xmpMF7U14L_V#scrollTo=5AFCnhZdmrsw).

The notebook interface includes a toolbar with options for Code, Text, RAM, and Disk. The code cell contains the following Python code:

```
1 lat_long = crime_df[['department_district', 'lat', 'long']].drop_duplicates(['department_district'])
2 lat_long.reset_index(drop=True, inplace=True)
3 lat_long
```

The output of the code is a DataFrame with 11 rows and 3 columns: department\_district, lat, and long. The data is as follows:

	department_district	lat	long
0	SOUTHERN	37.775420706711	-122.403404791479
1	BAYVIEW	37.7299809672996	-122.388856204292
2	TENDERLOIN	37.7857883766888	-122.412970537591
3	MISSION	37.7650501214668	-122.419671780296
4	NORTHERN	37.788018555829	-122.426077177375
5	TARAVAL	37.7644781578695	-122.477375524003
6	INGLESIDE	37.7292705199592	-122.432325871028
7	CENTRAL	37.791642962384	-122.40090869869
8	RICHMOND	37.7714939969416	-122.507750131004
9	PARK	37.7749912944366	-122.437799703468
10	NaN	37.7082024584902	-122.413351985284

The bottom status bar indicates the execution time as 0s and completion at 3:09 PM. The Windows taskbar at the bottom shows the search bar and various application icons.

Google Chrome browser window showing a Google Colab notebook. The address bar displays the URL: [https://colab.research.google.com/drive/1v9DGIXkvqCh7IDNLTtRt1xmpMF7U14L\\_V#scrollTo=CstGCowSmrsz](https://colab.research.google.com/drive/1v9DGIXkvqCh7IDNLTtRt1xmpMF7U14L_V#scrollTo=CstGCowSmrsz). The notebook interface includes a toolbar with '+ Code' and '+ Text' buttons, and a status bar showing 'RAM' and 'Disk' usage.

The code editor shows the following Python code:

```
[49] 1 district_crime_count = pd.merge(district_crime_count, lat_long, on='department_district')
      2

[50] 1 district_crime_count.dropna(inplace=True)
```

The output of the code is a DataFrame with the following data:

	department_district	count	lat	long
0	MISSION	141	37.7650501214668	-122.419671780296
1	SOUTHERN	141	37.775420706711	-122.403404791479
2	BAYVIEW	124	37.7299809572996	-122.388856204292
3	INGLESIDE	124	37.7292705199592	-122.432325871028
4	NORTHERN	121	37.788018555829	-122.426077177375
5	TARAVAL	110	37.7644781578695	-122.477376524003
6	TENDERLOIN	105	37.7857883766888	-122.412970537591
7	RICHMOND	102	37.7714939969416	-122.507760131004
8	CENTRAL	100	37.791642982384	-122.40090869889
9	PARK	99	37.7749912944366	-122.437799703468

The bottom of the notebook shows the execution status: '0s completed at 3:09 PM'. The Windows taskbar at the bottom displays the search bar, task icons, and system tray information including the date '08-01-2023' and time '15:09'.

Google Chrome browser window showing a Google Colab notebook. The address bar displays the URL: [https://colab.research.google.com/drive/1v9DGIXkvqGh7IDNLTt1xmpMF7U14L\\_V#scrollTo=CWvDV1u2mrs1](https://colab.research.google.com/drive/1v9DGIXkvqGh7IDNLTt1xmpMF7U14L_V#scrollTo=CWvDV1u2mrs1).

The notebook interface shows a code editor with the following Python code:

```
1 northing = 37.78
2 easting = -122.40
3 world_map = folium.Map(location=[northing,easting], tiles="cartodbpositron", zoom_start=12, max_zoom = 25, min_zoom = 2)
4
5
6 for i in range(0,len(district_crime_count)):
7     folium.Marker(
8         location=[district_crime_count.iloc[i]['lat'], district_crime_count.iloc[i]['long']],
9         fill=True,
10        #radius=(int(location_df.iloc[i,-1])),
11        color='#ffbb3d',
12        fill_color='indigo',
13        tooltip = "<div style='margin: 0; background-color: #cccccc; color: #000000;'>"+
14            "<h4 style='text-align:center;font-weight: bold;'>"+district_crime_count.iloc[i]['department_district'] + "</h4>"
15            "<hr style='margin:10px;color: #4d4d4d;'>"+
16            "<ul style='color: #000000;font-weight: bold;list-style-type:circle;align-items:left;padding-left:20px;padding-right:20px;'>"+
17                "<li>District: "+str(district_crime_count.iloc[i,0])+"</li>"+
18                "<li>Crime count: "+str(district_crime_count.iloc[i,1])+"</li>"+
19            "</ul></div>",
20        ).add_to(world_map)
21
22 world_map
```

The code is executed, and the output shows a map of the San Francisco area, specifically highlighting the Berkeley region. The map displays the coastline and surrounding areas, with labels for "Sausalito" and "BERKELEY".

The bottom status bar indicates the execution completed at 3:10 PM.

Google Chrome browser window showing a Google Colab notebook interface. The address bar displays the URL: [https://colab.research.google.com/drive/1v9DGIXkvqGh7IDNLTt1xmpMF7U14L\\_V#scrollTo=CWvDV1u2mrs1](https://colab.research.google.com/drive/1v9DGIXkvqGh7IDNLTt1xmpMF7U14L_V#scrollTo=CWvDV1u2mrs1).

The notebook interface includes a toolbar with options for Code, Text, and RAM/Disk usage. The main content area displays a map of the San Francisco Bay Area, showing various cities and locations. Several blue location pins are placed on the map, primarily clustered around San Francisco and the surrounding areas.

The map labels include: Richmond, El Cerrito, Berkeley, Orinda, Lafayette, Walnut Creek, Pleasant Hill, Clayton, Mill Valley, Corte Madera, Tiburon, Sausalito, Emeryville, Piedmont, Moraga, Alamo, Danville, Oakland, Alameda, San Ramon, San Leandro, Castro Valley, Dublin, Hayward, Pleasanton, Daly City, Colma, Brisbane, and South San Francisco.

The bottom status bar shows the system clock at 15:10 on 08-01-2023, along with various system icons and a search bar.



Google Chrome browser window showing a Google Colab notebook interface.

Address bar: [https://colab.research.google.com/drive/1v9DGIXkvqGh7IDNLTt1xmpMF7U14L\\_V#scrollTo=CWvDV1u2mrs1](https://colab.research.google.com/drive/1v9DGIXkvqGh7IDNLTt1xmpMF7U14L_V#scrollTo=CWvDV1u2mrs1)

Navigation bar: + Code + Text

Map view: SAN MATEO, Foster City, Burlingame

Main content area: Thank You!!

Footer: Colab paid products - [Cancel contracts here](#)

System tray: 0s completed at 3:10 PM

Windows taskbar: Type here to search, 15:10, 09-01-2023