DATA preparation

data file: Police_Department_Incident_Reports__2018_to_Present_20241128.csv
https://data.sfgov.org/Public-Safety/Police-Department-Incident-Reports-2018-to-Present/wg3w-h783/about_data

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
from csv import reader
from pyspark.sql import Row
from pyspark.sql import SparkSession
from pyspark.sql.types import *
import pandas as pd
import numpy as np
import seaborn as sb
import matplotlib.pyplot as plt
import warnings
import pandas as pd

import os
os.environ["PYSPARK_PYTHON"] = "python3"
```

```
# data_path = "./Police_Department_Incident_Reports__2018_to_Present_20241128.csv"
data_path = "/content/drive/MyDrive/480final/Police_Department_Incident_Reports__2018_to_Present_20241128.csv"
```

```
crime_data_lines = pd.read_csv(data_path)
print(crime_data_lines.head(0))
```

```
Empty DataFrame
Columns: [Incident Datetime, Incident Date, Incident Time, Incident Year, Incident Day of Week, Report Datetime, Row ID, Incident ID, Incident Number, CAD Number, Incident Code, Incident Category, Incident Subcategory, Incident Description, Resolution, Intersection, CNN, Police District, Analysis Neighborhood, Supervisor District, Supervisor District 2012, Latitude, Longitude, Point, Neighborhoods, ESNCAG - Bou
Index: []
```

```
crime_data_lines.head(5)
```

	Incident Datetime	Incident Date	Incident Time	Incident Year	Incident Day of Week	Report Datetime	Row ID	Incident ID	Incident Number	CAD Number	...	Longitude	Point
0	2023/03/13 11:41:00 PM	2023/03/13	23:41	2023	Monday	2023/03/13 11:41:00 PM	125373607041	1253736	230167874	NaN	...	NaN	NaN
1	2023/03/01 05:02:00 AM	2023/03/01	05:02	2023	Wednesday	2023/03/11 03:40:00 PM	125379506374	1253795	236046151	NaN	...	NaN	NaN
2	2023/03/13 01:16:00 PM	2023/03/13	13:16	2023	Monday	2023/03/13 01:17:00 PM	125357107041	1253571	220343896	NaN	...	NaN	NaN
3	2023/03/13 10:59:00 AM	2023/03/13	10:59	2023	Monday	2023/03/13 11:00:00 AM	125355107041	1253551	230174885	NaN	...	NaN	NaN
4	2023/03/14 06:44:00 PM	2023/03/14	18:44	2023	Tuesday	2023/03/14 06:45:00 PM	125402407041	1254024	230176728	NaN	...	NaN	NaN

5 rows × 35 columns

```
from pyspark.sql import SparkSession
spark = SparkSession \
    .builder \
    .appName("crime analysis") \
```

```
    .config("spark.some.config.option", "some-value") \
    .getOrCreate()

df_opt1 = spark.read.format("csv").option("header", "true").load(data_path)
display(df_opt1)
df_opt1.createOrReplaceTempView("sf_crime")
```

```
DataFrame[Incident Datetime: string, Incident Date: string, Incident Time: string, Incident Year: string, Incident Day of Week: string, Report Datetime: string, Row ID: string, Incident ID: string, Incident Number: string, CAD Number: string, Report Type Code: string, Report Type Description: string, Filed Online: string, Incident Code: string, Incident Category: string, Incident Subcategory: string, Incident Description: string, Resolution: string, Intersection: string, CNN: string, Police District: string, Analysis Neighborhood: string, Supervisor District: string, Supervisor District 2012: string, Latitude: string, Longitude: string, Point: string, Neighborhoods: string, ESNCAG - Boundary File: string, Central
```

the number of crimes for different category

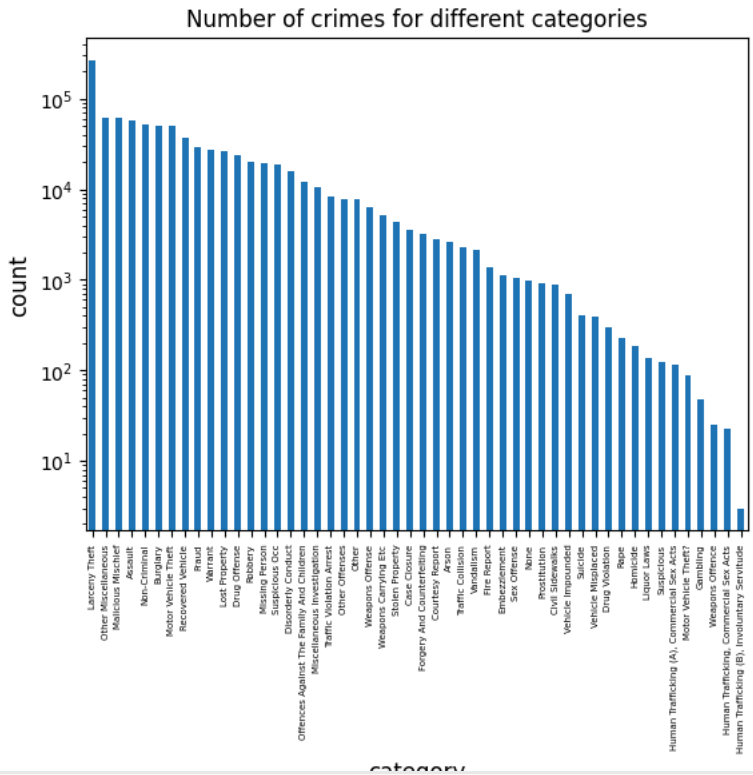
```
crime_category = spark.sql("""
                    SELECT `Incident Category`, COUNT(*) AS Count
                    FROM sf_crime
                    GROUP BY `Incident Category`
                    ORDER BY Count DESC
                    """)
crime_category.show()
```

```
+--------------------+------+
|   Incident Category| Count|
+--------------------+------+
|       Larceny Theft|269694|
| Other Miscellaneous| 62102|
|  Malicious Mischief| 61508|
|             Assault| 56766|
|        Non-Criminal| 52934|
|            Burglary| 50696|
| Motor Vehicle Theft| 50038|
|   Recovered Vehicle| 37008|
|               Fraud| 29403|
|             Warrant| 27074|
|       Lost Property| 26672|
|        Drug Offense| 23691|
|             Robbery| 20452|
|      Missing Person| 19763|
|      Suspicious Occ| 18839|
|  Disorderly Conduct| 15931|
|Offences Against ...| 12132|
|Miscellaneous Inv...| 10643|
|Traffic Violation...|  8361|
|      Other Offenses|  7829|
+--------------------+------+
only showing top 20 rows
```

双击（或按回车键）即可修改

```
crimes_pd_df = crime_category.toPandas()
plt.figure()
ax = crimes_pd_df.plot(kind = 'bar',x = 'Incident Category',y = 'Count',logy= True, legend = False, align = 'center')
ax.set_ylabel('count',fontsize = 12)
ax.set_xlabel('category',fontsize = 12)
plt.xticks(fontsize=5, rotation=90)
plt.title('Number of crimes for different categories')
display()
```

```
<Figure size 640x480 with 0 Axes>
```

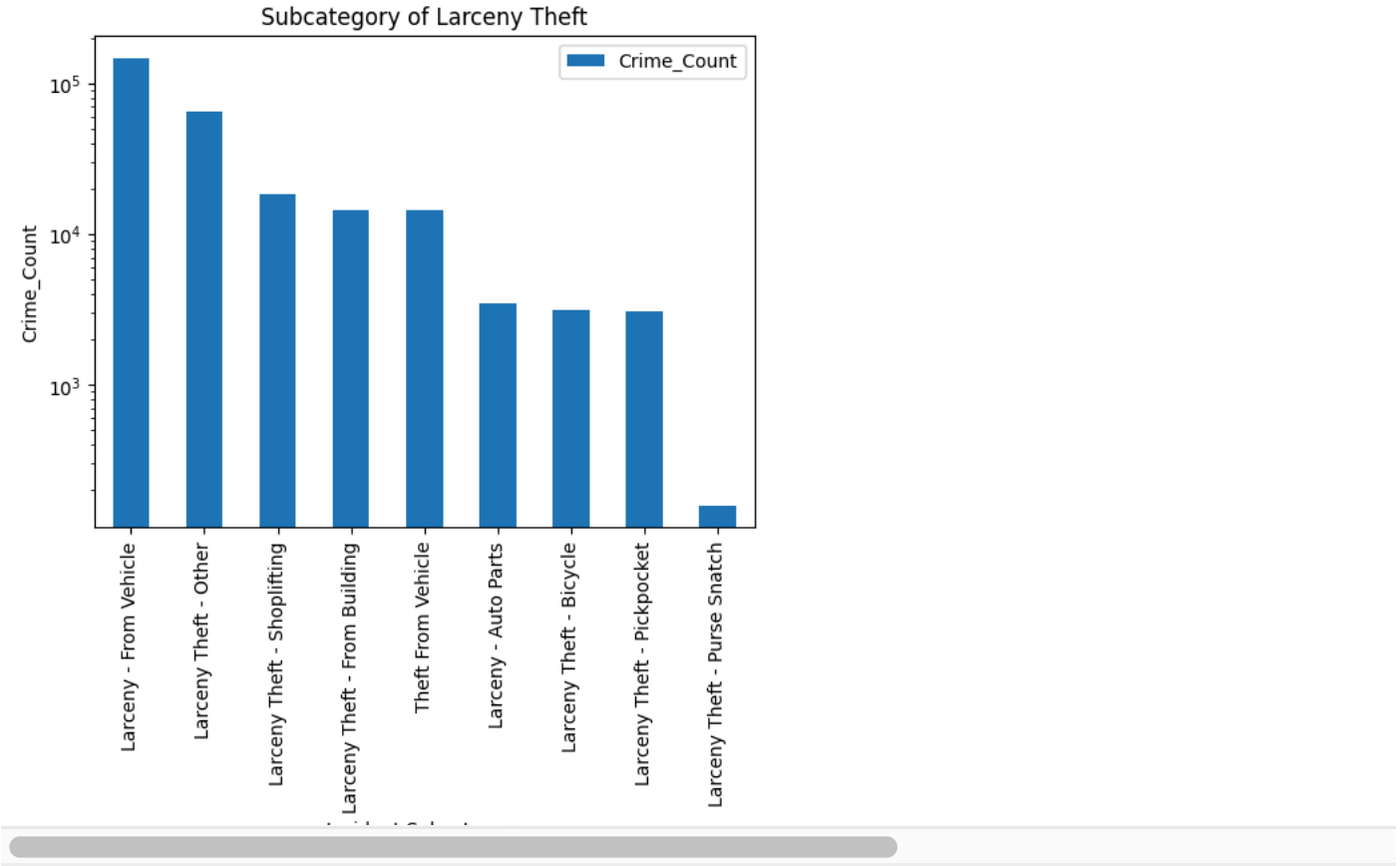



```
sub_Larceny_Theft = spark.sql("""
    SELECT `Incident Category`, `Incident Subcategory`,COUNT(*) AS Crime_Count
    FROM sf_crime
    WHERE `Incident Category` IN ('Larceny Theft')
    GROUP BY `Incident Category`,`Incident Subcategory`
    ORDER BY Crime_Count DESC
""")
sub_Larceny_Theft.show()
```

```
+----------------+--------------------+-----------+
|Incident Category|Incident Subcategory|Crime_Count|
+----------------+--------------------+-----------+
|    Larceny Theft|Larceny - From Ve...|     148191|
|    Larceny Theft|Larceny Theft - O...|      64418|
|    Larceny Theft|Larceny Theft - S...|      18481|
|    Larceny Theft|Larceny Theft - F...|      14418|
|    Larceny Theft|  Theft From Vehicle|      14305|
|    Larceny Theft|Larceny - Auto Parts|       3479|
|    Larceny Theft|Larceny Theft - B...|       3142|
|    Larceny Theft|Larceny Theft - P...|       3101|
|    Larceny Theft|Larceny Theft - P...|        159|
+----------------+--------------------+-----------+
```

```
from logging import log
sub_Larceny_Theft = sub_Larceny_Theft.toPandas()
plt.figure()
ax = sub_Larceny_Theft.plot(kind = 'bar',
                  x = 'Incident Subcategory',
                  y = 'Crime_Count',
                  logy = True,
                  legend = 'False',
                  align = 'center')
ax.set_ylabel('Crime_Count')
ax.set_xlabel('Incident Subcategory')
plt.title('Subcategory of Larceny Theft')
```

```
Text(0.5, 1.0, 'Subcategory of Larceny Theft')
<Figure size 640x480 with 0 Axes>
```



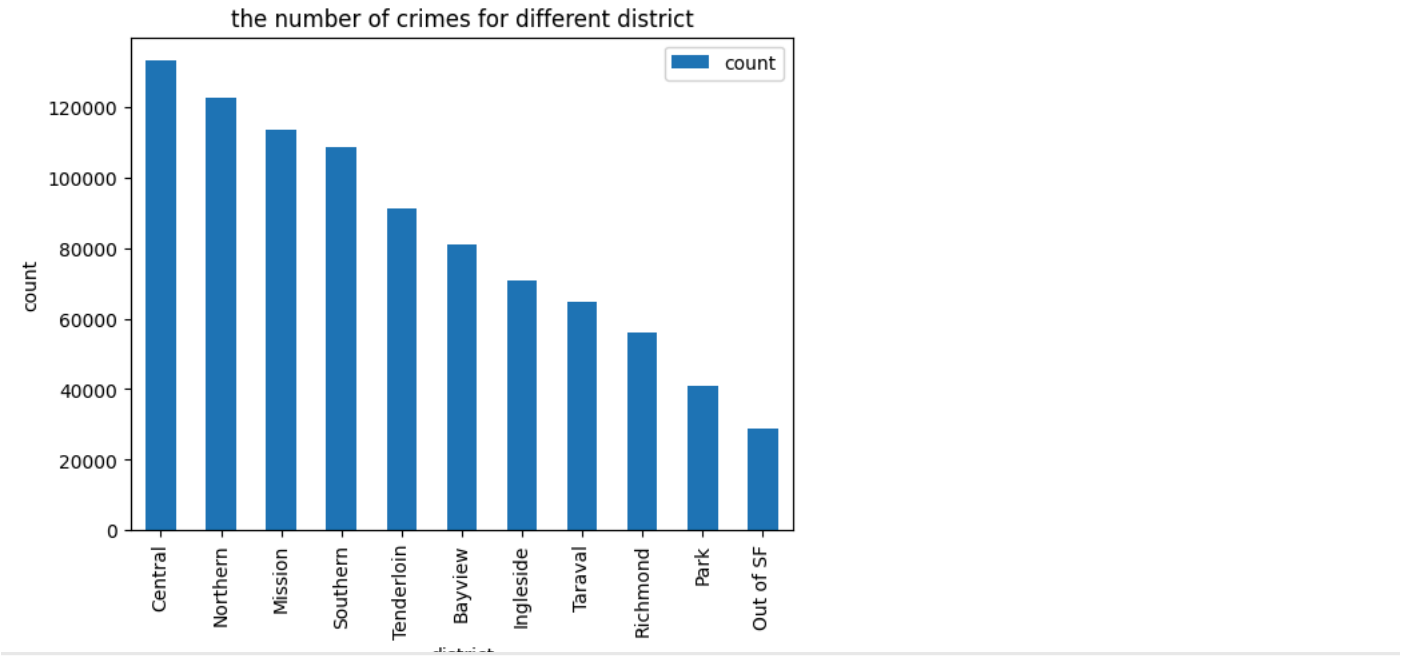
the number of crimes for different district

```
num_crime = spark.sql("""
                      SELECT `Police District`,
                      COUNT(*) AS count
                      FROM sf_crime
                      GROUP BY 1
                      ORDER BY 2 DESC
                      """)
num_crime.show()
```

```
+---------------+------+
|Police District| count|
+---------------+------+
|        Central|133128|
|       Northern|122518|
|        Mission|113400|
|       Southern|108457|
|     Tenderloin| 91277|
|        Bayview| 80965|
|      Ingleside| 70702|
|        Taraval| 64724|
|       Richmond| 55933|
|           Park| 41021|
|      Out of SF| 28764|
+---------------+------+
```

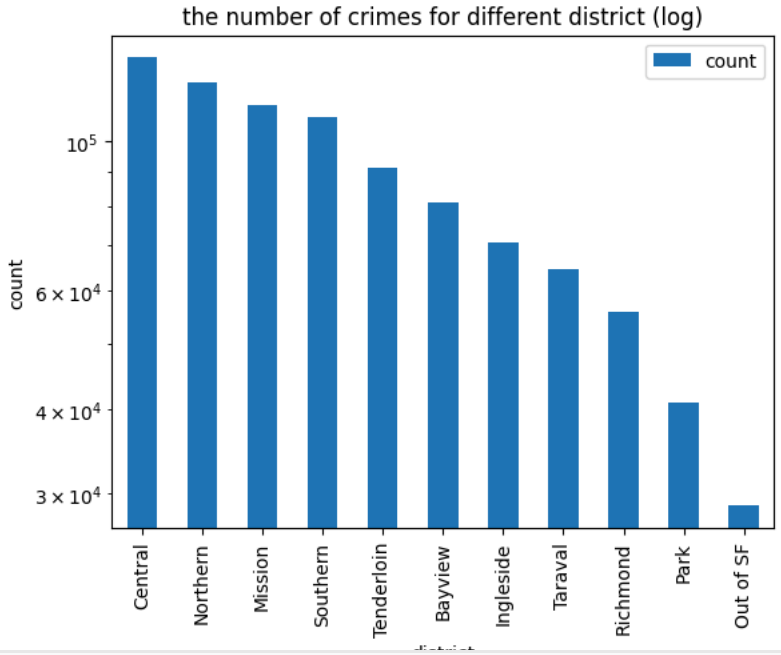
```
num_crime = num_crime.toPandas()
plt.figure()
ax = num_crime.plot(kind = 'bar',
                    x = 'Police District',
                    y = 'count',
                    legend = 'False',
                    align = 'center')
ax.set_ylabel('count')
ax.set_xlabel('district')
plt.title('the number of crimes for different district')
```

```
Text(0.5, 1.0, 'the number of crimes for different district')
<Figure size 640x480 with 0 Axes>
```




```
plt.figure()
ax = num_crime.plot(kind = 'bar',
                    x = 'Police District',
                    y = 'count',
                    logy = True,
                    legend = 'False',
                    align = 'center')
ax.set_ylabel('count')
ax.set_xlabel('district')
plt.title('the number of crimes for different district (log)')
```

```
Text(0.5, 1.0, 'the number of crimes for different district (log)')
<Figure size 640x480 with 0 Axes>
```




```
df_opt1.show(1)
```

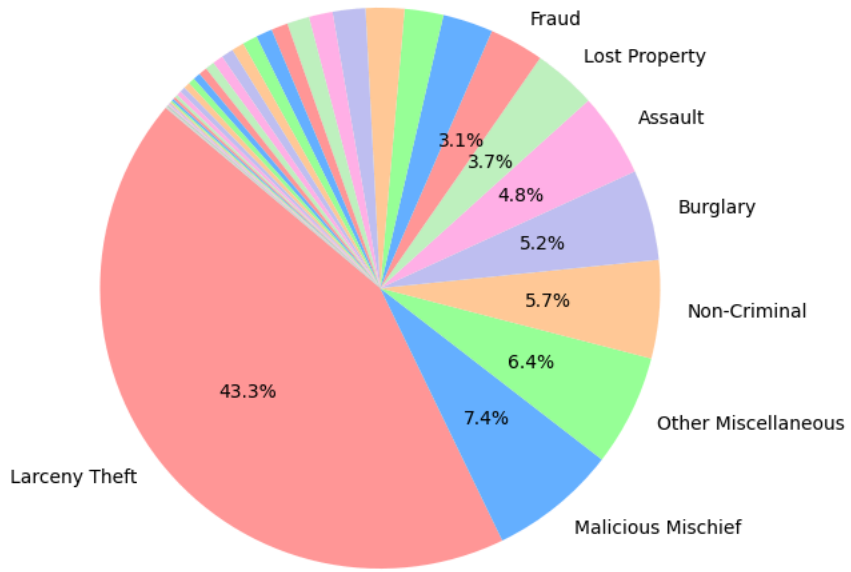
```
+-------------------+-------------+-------------+-------------+--------------------+--------------------+------------+-----
|   Incident Datetime|Incident Date|Incident Time|Incident Year|Incident Day of Week|     Report Datetime|      Row ID|Incid
+-------------------+-------------+-------------+-------------+--------------------+--------------------+------------+-----
|2023/03/13 11:41:...|   2023/03/13|        23:41|         2023|              Monday|2023/03/13 11:41:...|125373607041|    1
+-------------------+-------------+-------------+-------------+--------------------+--------------------+------------+-----
only showing top 1 row
```

```
central = spark.sql("""
    SELECT `Police District`, `Incident Category`, COUNT(*) AS Crime_Count
    FROM sf_crime
    WHERE `Police District` IN ('Central')
    GROUP BY `Police District`, `Incident Category`
    ORDER BY Crime_Count DESC
""")
central.show()
```

```
+---------------+--------------------+-----------+
|Police District|   Incident Category|Crime_Count|
+---------------+--------------------+-----------+
|        Central|       Larceny Theft|      57608|
|        Central|  Malicious Mischief|       9783|
|        Central| Other Miscellaneous|       8549|
|        Central|        Non-Criminal|       7533|
|        Central|            Burglary|       6941|
|        Central|             Assault|       6423|
|        Central|       Lost Property|       4870|
|        Central|               Fraud|       4181|
|        Central| Motor Vehicle Theft|       3856|
|        Central|             Warrant|       2979|
|        Central|             Robbery|       2969|
|        Central|      Suspicious Occ|       2513|
|        Central|      Missing Person|       1825|
|        Central|  Disorderly Conduct|       1710|
|        Central|        Drug Offense|       1273|
|        Central|   Recovered Vehicle|       1249|
|        Central|Offences Against ...|       1145|
|        Central|               Other|        949|
|        Central|Miscellaneous Inv...|        919|
|        Central|Traffic Violation...|        810|
+---------------+--------------------+-----------+
only showing top 20 rows
```

```
central_df = central.toPandas()
crime_category_counts = central_df.groupby('Incident Category')['Crime_Count'].sum().reset_index()
crime_category_counts = crime_category_counts.sort_values(by='Crime_Count', ascending=False)
total_count = crime_category_counts['Crime_Count'].sum()
crime_category_counts['Percentage'] = (crime_category_counts['Crime_Count'] / total_count) * 100
threshold = 3
labels = crime_category_counts.apply(
    lambda row: row['Incident Category'] if row['Percentage'] >= threshold else '', axis=1
)

def autopct_format(pct):
    return f'{pct:.1f}%' if pct > 3 else ''
custom_colors = ['#ff9999','#66b3ff','#99ff99','#ffcc99', '#c2c2f0', '#ffb3e6', '#c2f0c2']
plt.figure(figsize=(10, 6))
plt.pie(
    crime_category_counts['Crime_Count'],
    labels=labels,
    autopct=autopct_format,
    startangle=140,
    colors=custom_colors
)
plt.title('Crime Category Proportion in Central District (Filtered)')
plt.axis('equal')
plt.show()
```



monthly analytics

```
df_opt1
```

```
DataFrame[Incident Datetime: string, Incident Date: string, Incident Time: string, Incident Year: string, Incident Day of Week: string, Report Datetime: string, Row ID: string, Incident ID: string, Incident Number: string, CAD Number: string, Report Type Code: string, Report Type Description: string, Filed Online: string, Incident Code: string, Incident Category: string, Incident Subcategory: string, Incident Description: string, Resolution: string, Intersection: string, CNN: string, Police District: string, Analysis Neighborhood: string, Supervisor District: string, Supervisor District 2012: string, Latitude: string, Longitude: string, Point: string, Neighborhoods: string, ESNCAG - Boundary File: string, Central Market/Tenderloin Boundary Polygon - Updated: string, Civic Center Harm Reduction Project Boundary: string, HSOC Zones as of 2018-06-05: string, Invest In Neighborhoods (IIN) Areas: string, Current Supervisor Districts: string, Current Police Districts: string]
```

```
df_opt2 = df_opt1[['Incident Datetime','Incident Date','Incident Time','Incident Year',
                   'Incident Day of Week','Report Datetime','Incident Number','Incident Category',
                   'Resolution','Police District','Latitude','Longitude']]
display(df_opt2)
df_opt2.createOrReplaceTempView("sf_crime")
```

```
DataFrame[Incident Datetime: string, Incident Date: string, Incident Time: string, Incident Year: string, Incident Day of
```

```
df_opt2.show()
```

```
+-------------------+-------------+-------------+-------------+--------------------+-------------------+---------------+--
|   Incident Datetime|Incident Date|Incident Time|Incident Year|Incident Day of Week|     Report Datetime|Incident Number|In
+-------------------+-------------+-------------+-------------+--------------------+-------------------+---------------+--
|2023/03/13 11:41:...|   2023/03/13|        23:41|         2023|              Monday|2023/03/13 11:41:...|      230167874|Re
|2023/03/01 05:02:...|   2023/03/01|        05:02|         2023|           Wednesday|2023/03/11 03:40:...|      236046151|
|2023/03/13 01:16:...|   2023/03/13|        13:16|         2023|              Monday|2023/03/13 01:17:...|      220343896|Re
|2023/03/13 10:59:...|   2023/03/13|        10:59|         2023|              Monday|2023/03/13 11:00:...|      230174885|Re
|2023/03/14 06:44:...|   2023/03/14|        18:44|         2023|             Tuesday|2023/03/14 06:45:...|      230176728|Re
|2023/02/15 03:00:...|   2023/02/15|        03:00|         2023|           Wednesday|2023/03/11 04:55:...|      236046123|
|2023/03/11 12:30:...|   2023/03/11|        12:30|         2023|            Saturday|2023/03/12 04:15:...|      236046004|
|2023/03/13 11:26:...|   2023/03/13|        11:26|         2023|              Monday|2023/03/13 01:37:...|      236046850|
|2023/03/11 03:00:...|   2023/03/11|        15:00|         2023|            Saturday|2023/03/13 08:29:...|      236045937|
|2023/03/11 02:00:...|   2023/03/11|        14:00|         2023|            Saturday|2023/03/15 11:21:...|      230182844|
|2023/03/13 07:30:...|   2023/03/13|        07:30|         2023|              Monday|2023/03/14 07:11:...|      236047096|
|2022/06/27 12:00:...|   2022/06/27|        12:00|         2022|              Monday|2023/03/15 05:20:...|      230184129|
|2023/03/16 09:26:...|   2023/03/16|        09:26|         2023|            Thursday|2023/03/16 09:26:...|      230185672|
|2023/03/16 05:30:...|   2023/03/16|        17:30|         2023|            Thursday|2023/03/16 06:02:...|      230187101|
|2023/02/21 02:15:...|   2023/02/21|        14:15|         2023|             Tuesday|2023/02/25 09:56:...|      236047529|
|2023/03/16 01:49:...|   2023/03/16|        13:49|         2023|            Thursday|2023/03/16 01:49:...|      230178047|Re
|2023/03/16 10:15:...|   2023/03/16|        22:15|         2023|            Thursday|2023/03/17 12:03:...|      236049456|
|2023/03/13 11:35:...|   2023/03/13|        11:35|         2023|              Monday|2023/03/18 05:16:...|      236046816|
|2023/02/11 02:00:...|   2023/02/11|        14:00|         2023|            Saturday|2023/03/18 01:20:...|      236050049|
|2023/03/16 09:00:...|   2023/03/16|        21:00|         2023|            Thursday|2023/03/16 09:15:...|      230172566|Re
+-------------------+-------------+-------------+-------------+--------------------+-------------------+---------------+--
only showing top 20 rows
```

```
monthly_result = spark.sql("""
                    SELECT SUBSTRING(`Incident Date`,6,2) AS MONTH, SUBSTRING(`Incident Date`,1,4) AS YEAR, COUNT(*) AS Count
                    From sf_crime
                    GROUP BY Year, Month
                    HAVING Year in (2018,2019,2020,2021,2022,2023)
                    ORDER BY Year, Month
                    """)
monthly_result.show()
```

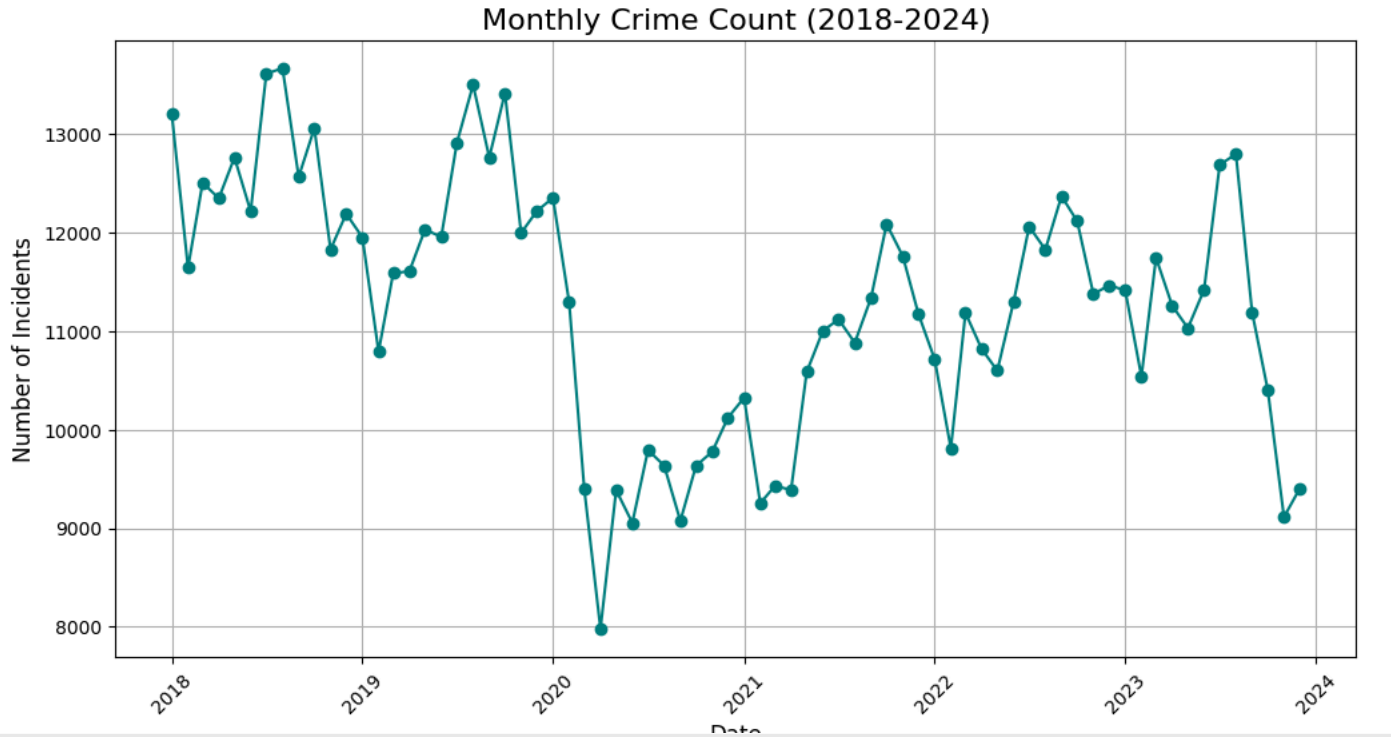
```
+-----+----+-----+
|MONTH|YEAR|Count|
+-----+----+-----+
|   01|2018|13200|
|   02|2018|11642|
|   03|2018|12504|
|   04|2018|12346|
|   05|2018|12760|
|   06|2018|12214|
|   07|2018|13610|
|   08|2018|13665|
|   09|2018|12564|
|   10|2018|13056|
|   11|2018|11827|
|   12|2018|12189|
|   01|2019|11950|
|   02|2019|10791|
|   03|2019|11586|
|   04|2019|11606|
|   05|2019|12031|
|   06|2019|11953|
|   07|2019|12909|
|   08|2019|13497|
+-----+----+-----+
only showing top 20 rows
```

```
monthly_result = monthly_result.toPandas()
monthly_result['Date'] = pd.to_datetime(monthly_result['YEAR'] + '-' + monthly_result['MONTH'] + '-01')
monthly_result = monthly_result.sort_values(by='Date')

plt.figure(figsize=(12, 6))
plt.plot(monthly_result['Date'], monthly_result['Count'], marker='o', color='teal')
plt.title('Monthly Crime Count (2018-2024)', fontsize=16)
plt.xlabel('Date', fontsize=12)
plt.ylabel('Number of Incidents', fontsize=12)
plt.xticks(rotation=45)
```

```
plt.grid(True)

plt.show()
```




```
LT_result = spark.sql("""
                      SELECT SUBSTRING(`Incident Date`,1,4) AS Year,
                             SUBSTRING(`Incident Date`,6,2) AS Month,
                             COUNT(*) AS Larceny_Theft_Count
                      FROM sf_crime
                      WHERE `Incident Category` = 'Larceny Theft'
                            AND SUBSTRING(`Incident Date`,1,4)
                            IN ('2018', '2019', '2020', '2021','2022','2023')
                      GROUP BY Year, Month
                      ORDER BY Year, Month
                      """)

LT_result.show()
```

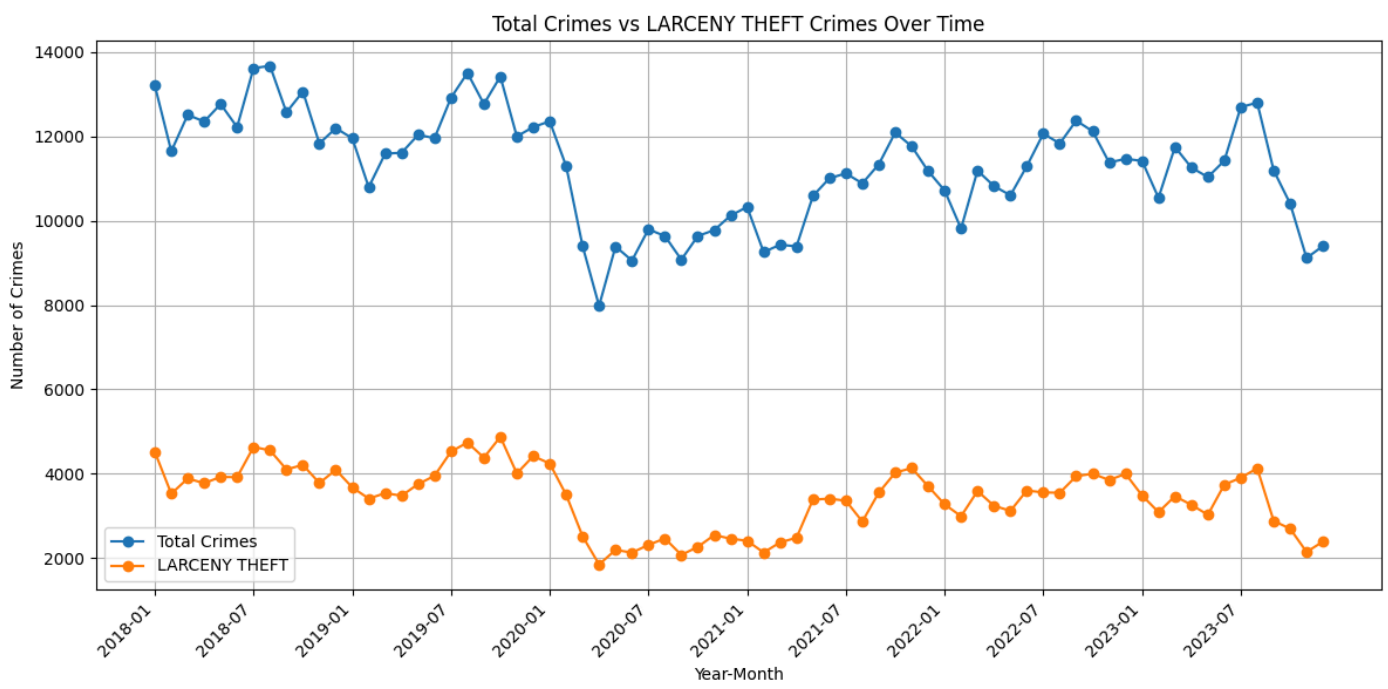
```
+----+-----+-------------------+
|Year|Month|Larceny_Theft_Count|
+----+-----+-------------------+
|2018|   01|               4521|
|2018|   02|               3531|
|2018|   03|               3890|
|2018|   04|               3780|
|2018|   05|               3929|
|2018|   06|               3922|
|2018|   07|               4637|
|2018|   08|               4558|
|2018|   09|               4106|
|2018|   10|               4213|
|2018|   11|               3783|
|2018|   12|               4096|
|2019|   01|               3680|
|2019|   02|               3415|
|2019|   03|               3543|
|2019|   04|               3484|
|2019|   05|               3757|
|2019|   06|               3955|
|2019|   07|               4528|
|2019|   08|               4743|
+----+-----+-------------------+
only showing top 20 rows
```

```
df_total = monthly_result
df_theft = LT_result.toPandas()
```

```
df_total['YearMonth'] = df_total['YEAR'] + '-' + df_total['MONTH']
df_theft['YearMonth'] = df_theft['Year'] + '-' + df_theft['Month']

df_total = df_total.sort_values('YearMonth')
df_theft = df_theft.sort_values('YearMonth')

plt.figure(figsize=(12, 6))
plt.plot(df_total['YearMonth'], df_total['Count'], label='Total Crimes', marker='o')
plt.plot(df_theft['YearMonth'], df_theft['Larceny_Theft_Count'], label='LARCENY THEFT', marker='o')
plt.legend()
plt.title('Total Crimes vs LARCENY THEFT Crimes Over Time')
plt.xlabel('Year-Month')
plt.ylabel('Number of Crimes')
xticks = range(0, len(df_total['YearMonth']), 6)
plt.xticks(xticks, df_total['YearMonth'][xticks], rotation=45, ha='right')
# plt.xticks(rotation=45, ha='right')
plt.grid(True)
plt.tight_layout()
plt.show()
```



the number of crime with respsect to the hour at Christmas

```
df_opt2.show(2)
```

```
+--------------------+-------------+-------------+-------------+--------------------+--------------------+---------------+--
|   Incident Datetime|Incident Date|Incident Time|Incident Year|Incident Day of Week|     Report Datetime|Incident Number|In
+--------------------+-------------+-------------+-------------+--------------------+--------------------+---------------+--
|2023/03/13 11:41:...|   2023/03/13|        23:41|         2023|              Monday|2023/03/13 11:41:...|      230167874|Re
|2023/03/01 05:02:...|   2023/03/01|        05:02|         2023|           Wednesday|2023/03/11 03:40:...|      236046151|
+--------------------+-------------+-------------+-------------+--------------------+--------------------+---------------+--
only showing top 2 rows
```

```
christmas_result = spark.sql("""
    SELECT
        SUBSTRING(`Incident Time`,1,2) AS Hour,
        SUBSTRING(`Incident Date`,6,5) AS Date,
        COUNT(*) AS Crime_Count
    FROM sf_crime
```

```
    WHERE SUBSTRING(`Incident Date`,6,5) IN ('12/25')
    GROUP BY Hour, Date
    ORDER BY Hour
""")
christmas_result.show(50)
```

```
+----+-----+-----------+
|Hour| Date|Crime_Count|
+----+-----+-----------+
|  00|12/25|        107|
|  01|12/25|         26|
|  02|12/25|         60|
|  03|12/25|         24|
|  04|12/25|         35|
|  05|12/25|         22|
|  06|12/25|         27|
|  07|12/25|         29|
|  08|12/25|         33|
|  09|12/25|         64|
|  10|12/25|         44|
|  11|12/25|         88|
|  12|12/25|        121|
|  13|12/25|         73|
|  14|12/25|         64|
|  15|12/25|         73|
|  16|12/25|         78|
|  17|12/25|         77|
|  18|12/25|         72|
|  19|12/25|         79|
|  20|12/25|         71|
|  21|12/25|         68|
|  22|12/25|         60|
|  23|12/25|         51|
+----+-----+-----------+
```

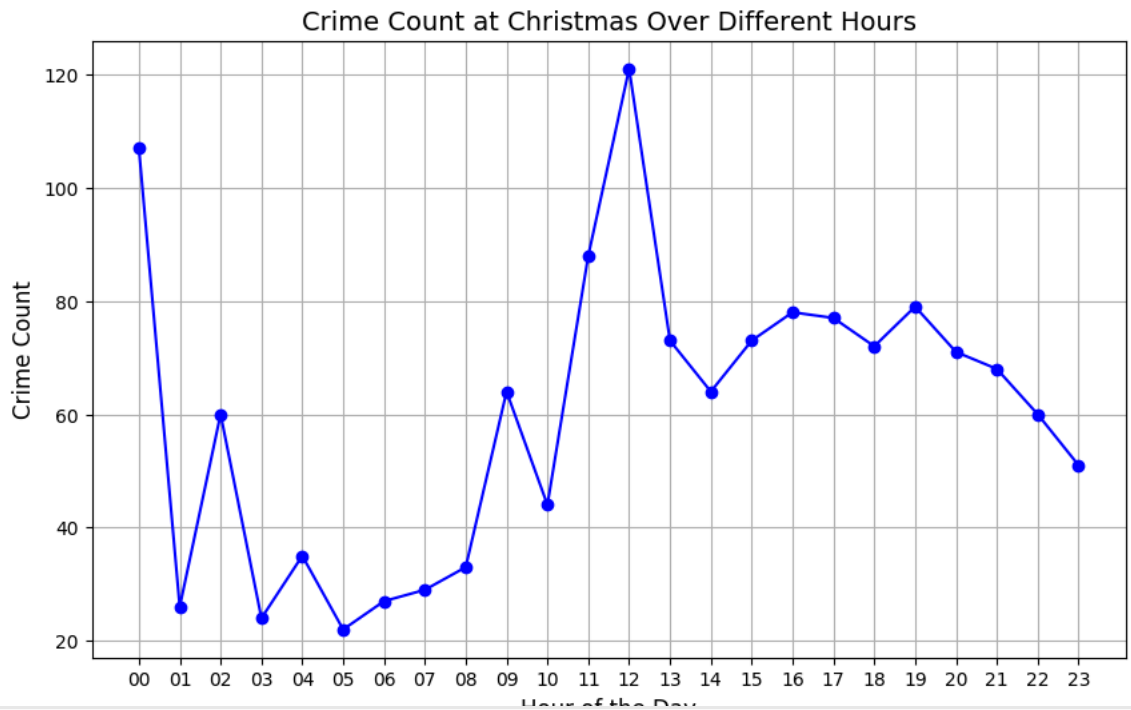
```
christmas2018_result = spark.sql("""
    SELECT
        SUBSTRING(`Incident Time`,1,2) AS Hour,
        COUNT(*) AS Crime_Count
    FROM sf_crime
    WHERE SUBSTRING(`Incident Date`,1,10) IN ('2018/12/25')
    GROUP BY Hour
    ORDER BY Hour
""")
christmas2018_result.show(50)
```

```
+----+-----------+
|Hour|Crime_Count|
+----+-----------+
|  00|         14|
|  01|          3|
|  02|          7|
|  03|          3|
|  04|          1|
|  05|          1|
|  06|          5|
|  07|          7|
|  08|          5|
|  09|         11|
|  10|          6|
|  11|         23|
|  12|         24|
|  13|         21|
|  14|         14|
|  15|         12|
|  16|          9|
|  17|         16|
|  18|         19|
|  19|         14|
|  20|         15|
|  21|          8|
|  22|         10|
|  23|          9|
+----+-----------+
```

```
df = christmas_result.toPandas()

plt.figure(figsize=(10, 6))
plt.plot(df['Hour'], df['Crime_Count'], marker='o', linestyle='-', color='b')
```

```
plt.title('Crime Count at Christmas Over Different Hours', fontsize=14)
plt.ylabel('Crime Count', fontsize=12)
plt.xlabel('Hour of the Day', fontsize=12)
plt.grid(True)
plt.xticks(df['Hour'])
plt.show()
```



precentation of resolution

```
res_num = spark.sql("""
                    SELECT `Incident Category`, resolution, COUNT(*) AS N_res
                    FROM sf_crime
                    GROUP BY `Incident Category`, resolution
                    """)
res_num.createOrReplaceTempView("res_num")


cate_num = spark.sql("""
                     SELECT `Incident Category`, COUNT(*) AS N_cate
                     FROM sf_crime
                     GROUP BY `Incident Category`
                     """)
cate_num.createOrReplaceTempView("cate_num")


resolution_result = spark.sql("""
                      SELECT distinct sf_crime.`Incident Category`, sf_crime.resolution, res_num.N_res/cate_num.N_cate AS Percen
                      FROM (sf_crime
                      LEFT JOIN res_num ON sf_crime.`Incident Category` = res_num.`Incident Category` AND sf_crime.resolution =
                      LEFT JOIN cate_num ON sf_crime.`Incident Category` = cate_num.`Incident Category`
                      ORDER BY `Incident Category`, resolution
                      """)
resolution_result.createOrReplaceTempView("resolution_result")


resolution_result.show()
```

```
+-----------------+--------------------+--------------------+
|Incident Category|          resolution|          Percentage|
+-----------------+--------------------+--------------------+
|             NULL|Cite or Arrest Adult|                NULL|
|             NULL|   Exceptional Adult|                NULL|
|             NULL|      Open or Active|                NULL|
|             NULL|           Unfounded|                NULL|
|            Arson|Cite or Arrest Adult| 0.20391705069124424|
|            Arson|   Exceptional Adult|0.004224270353302612|
```

```
|            Arson|      Open or Active|  0.7880184331797235|
|            Arson|           Unfounded|0.003840245775729...|
|          Assault|Cite or Arrest Adult|   0.277648592467322|
|          Assault|   Exceptional Adult|0.003628932811894...|
|          Assault|      Open or Active|  0.7173484127822992|
|          Assault|           Unfounded|0.001374061938484304|
|         Burglary|Cite or Arrest Adult| 0.11529509231497553|
|         Burglary|   Exceptional Adult|5.917626637210036E-4|
|         Burglary|      Open or Active|  0.8836200094682026|
|         Burglary|           Unfounded|4.931355531008363E-4|
|     Case Closure|Cite or Arrest Adult|  0.5510033444816054|
|     Case Closure|   Exceptional Adult| 0.20011148272017837|
|     Case Closure|      Open or Active|0.020903010033444816|
|     Case Closure|           Unfounded| 0.22798216276477146|
+-----------------+--------------------+--------------------+
only showing top 20 rows
```

```
LT = spark.sql("""
               SELECT resolution, Percentage
               FROM resolution_result
               WHERE `Incident Category` = 'Larceny Theft'
               ORDER BY Percentage DESC
               """)
LT.show()
```

```
+--------------------+--------------------+
|          resolution|          Percentage|
+--------------------+--------------------+
|      Open or Active|  0.9604959695061811|
|Cite or Arrest Adult| 0.03864750420847331|
|           Unfounded|5.598938055722412E-4|
|   Exceptional Adult|2.966324797733728E-4|
+--------------------+--------------------+
```

```
MM = spark.sql("""
               SELECT resolution, Percentage
               FROM resolution_result
               WHERE `Incident Category` = 'Malicious Mischief'
               ORDER BY Percentage DESC
               """)
MM.show()
```

```
+--------------------+--------------------+
```