Data Analysis of San Francisco Crime Case in Apache Spark

Data source: https://data.sfgov.org/Public-Safety/Police-Department-Incident-Reports-Historical-2003/tmnf-yvry (https://data.sfgov.org/Public-Safety/Police-Department-Incident-Reports-Historical-2003/tmnf-yvry)

Contents
1. Import Package and Data
2. OLAP tasks
3. Conclusions and Suggestions

1. Import Package and Data

1.1 Import Package

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

1.2 Import Data

```
import urllib.request
urllib.request.urlretrieve("https://data.sfgov.org/api/views/tmnf-yvry/rows.csv?accessType=DOWNLOAD", "/tmp/myxxxx.csv")
dbutils.fs.mv("file:/tmp/myxxxx.csv", "dbfs:/laioffer/spark_hw1/data/sf_03_18.csv")
display(dbutils.fs.ls("dbfs:/laioffer/spark_hw1/data/"))
```

	path	name	size
1	dbfs:/laioffer/spark_hw1/data/sf_03_18.csv	sf_03_18.csv	559169754

Showing all 1 rows.

```
data_path = "dbfs:/laioffer/spark_hw1/data/sf_03_18.csv"
```

1.3 Get dataframe and SQL

```
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")
```

	PdId	IncidntNum	Incident Code	Category	Descript	D
1	7121491514040	071214915	14040	OTHER OFFENSES	INDECENT EXPOSURE	T
2	7036663851040	070366638	51040	NON-CRIMINAL	AIDED CASE	T
3	4059322571000	040593225	71000	NON-CRIMINAL	LOST PROPERTY	T
4	3085157264070	030851572	64070	SUSPICIOUS OCC	SUSPICIOUS OCCURRENCE	T
5	13067727606304	130677276	06304	LARCENY/THEFT	GRAND THEFT FROM A BUILDING	T
6	11056206728135	110562067	28135	OTHER OFFENSES	HARASSING PHONE CALLS	W
7	11081732316710	110817323	16710	DRUG/NARCOTIC	POSSESSION OF NARCOTICS PARAPHERNALIA	M
8	12086972806153	120869728	06153	LARCENY/THEFT	GRAND THEFT FROM PERSON	S

Showing the first 1000 rows.

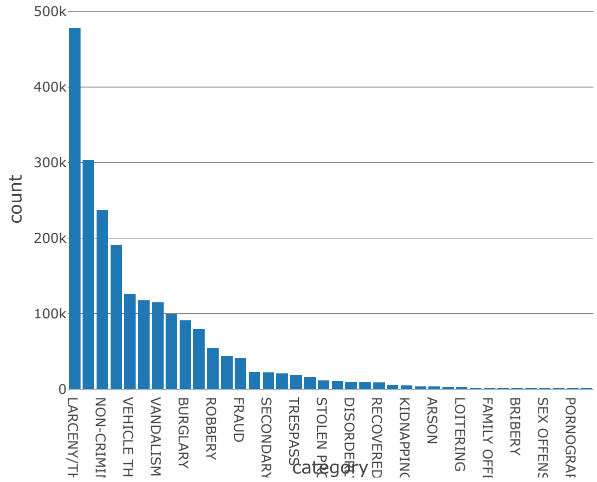
```
type(df_opt1)
```

```
Out[6]: pyspark.sql.dataframe.DataFrame
```

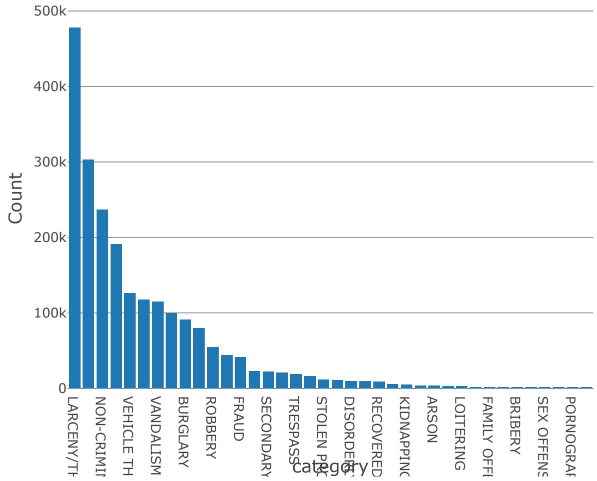
2. OLAP tasks

2.1 Counts the number of crimes for different category.

```
q1_result = df_opt1.groupBy('category').count().orderBy('count', ascending=False)
display(q1_result)
```



```
#Spark SQL based
crimeCategory = spark.sql("SELECT  category, COUNT(*) AS Count \
                          FROM sf_crime \
                          GROUP BY category \
                          ORDER BY Count DESC")
display(crimeCategory)
```



```
crimes_pd_df = crimeCategory.toPandas()
type(crimes_pd_df)
```

```
Out[9]: pandas.core.frame.DataFrame
```

```
display(crimes_pd_df)
```

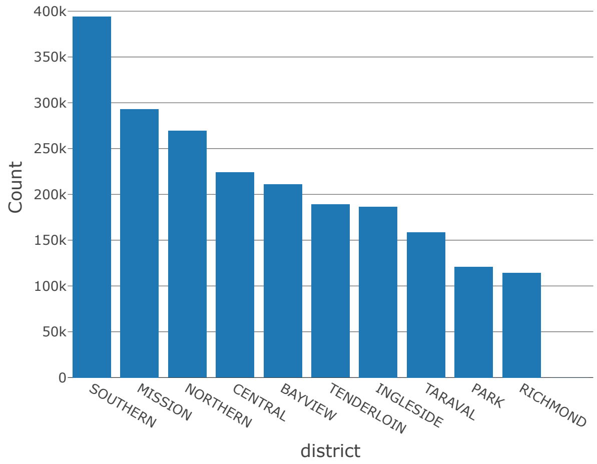
	category	Count
1	LARCENY/THEFT	477975
2	OTHER OFFENSES	303027
3	NON-CRIMINAL	236937

4	ASSAULT	191384
5	VEHICLE THEFT	126228
6	DRUG/NARCOTIC	117875
7	VANDALISM	114718
8	WARRANTS	99821

Showing all 38 rows.

2.2 Counts the number of crimes for different district, and visualize the results.

```
df_q2 = spark.sql("SELECT  pddistrict as district, COUNT(*) AS Count \
                           FROM sf_crime \
                           GROUP BY pddistrict \
                           ORDER BY Count DESC")
display(df_q2)
```

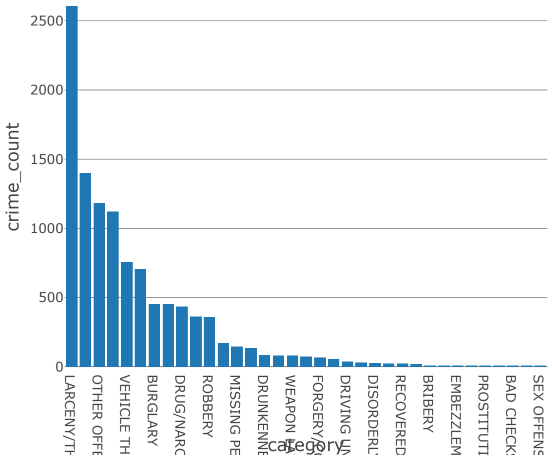


2.3 Count the number of crimes each "Sunday" at "SF downtown".

I assume SF downtown spacial range: X (-122.4213,-122.4313), Y(37.7540,37.7740).

```
df_q3 = spark.sql("select category, count(*) as crime_count \
                   from sf_crime \
                   where DayOfWeek ==  'Sunday' \
                   and X > -122.4313 and X < -122.4213 \
                   and Y > 37.7540 and Y < 37.7740 \
                   group by category \
                   order by crime_count desc")


display(df_q3)
```



2.4 Analysis the number of crime in each month of 2015, 2016, 2017, 2018.

```
from pyspark.sql.functions import *
df_update = df_opt1.withColumn('Date', to_date(col('Date'), 'MM/dd/yyyy'))
display(df_update)
```

	PdId	IncidntNum	Incident Code	Category	Descript	D
1	7121491514040	071214915	14040	OTHER OFFENSES	INDECENT EXPOSURE	T
2	7036663851040	070366638	51040	NON-CRIMINAL	AIDED CASE	T
3	4059322571000	040593225	71000	NON-CRIMINAL	LOST PROPERTY	T
4	3085157264070	030851572	64070	SUSPICIOUS OCC	SUSPICIOUS OCCURRENCE	T
5	13067727606304	130677276	06304	LARCENY/THEFT	GRAND THEFT FROM A BUILDING	T
6	11056206728135	110562067	28135	OTHER OFFENSES	HARASSING PHONE CALLS	W
7	11081732316710	110817323	16710	DRUG/NARCOTIC	POSSESSION OF NARCOTICS PARAPHERNALIA	M
8	12086972806153	120869728	06153	LARCENY/THEFT	GRAND THEFT FROM PERSON	S

Showing the first 1000 rows.

```
df_update.createOrReplaceTempView('sf_crime')
```

```
q4_2015 = spark.sql("select * from sf_crime \
                    where year(Date) = '2015' \
                    ")
display(q4_2015)
```

	PdId	IncidntNum	Incident Code	Category	Descript	D
1	15608177871000	156081778	71000	NON-CRIMINAL	LOST PROPERTY	W
2	15103362704134	151033627	04134	ASSAULT	BATTERY	S
3	15611910371000	156119103	71000	NON-CRIMINAL	LOST PROPERTY	S
4	15111066704134	151110667	04134	ASSAULT	BATTERY	F
5	16000378272000	160003782	72000	NON-CRIMINAL	FOUND PROPERTY	T
6	15025031371000	150250313	71000	NON-CRIMINAL	LOST PROPERTY	F
7	15031786675000	150317866	75000	MISSING PERSON	FOUND PERSON	S
8	15029320004134	150293200	04134	ASSAULT	BATTERY	F

Showing the first 1000 rows.

```
q4_2016 = spark.sql("select * from sf_crime \
                    where year(Date) = '2016' \
                    ")

q4_2017 = spark.sql("select * from sf_crime \
                    where year(Date) = '2017' \
                    ")

q4_2018 = spark.sql("select * from sf_crime \
                    where year(Date) = '2018' \
                    ")
```

```
q4_2015.createOrReplaceTempView('sf_crime_2015')
q4_2016.createOrReplaceTempView('sf_crime_2016')
q4_2017.createOrReplaceTempView('sf_crime_2017')
q4_2018.createOrReplaceTempView('sf_crime_2018')
```

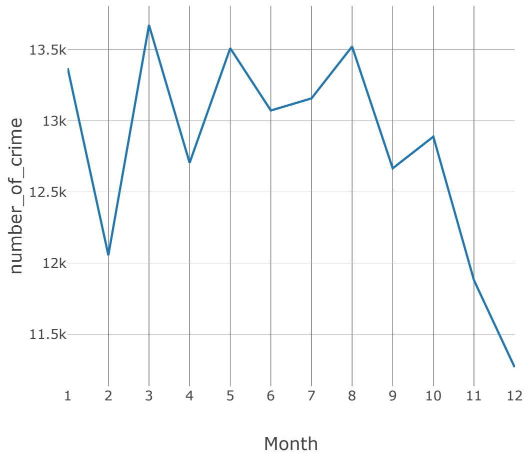
```
q4_2015_month = spark.sql("select month(date) as Month, count(*) as number_of_crime \
                          from sf_crime_2015 \
                          group by Month \
                          order by Month asc")
q4_2016_month = spark.sql("select month(date) as Month, count(*) as number_of_crime \
                          from sf_crime_2016 \
                          group by Month \
                          order by Month asc")

q4_2017_month = spark.sql("select month(date) as Month, count(*) as number_of_crime \
                          from sf_crime_2017 \
                          group by Month \
                          order by Month asc")

q4_2018_month = spark.sql("select month(date) as Month, count(*) as number_of_crime \
                          from sf_crime_2018 \
                          group by Month \
                          order by Month asc")
```

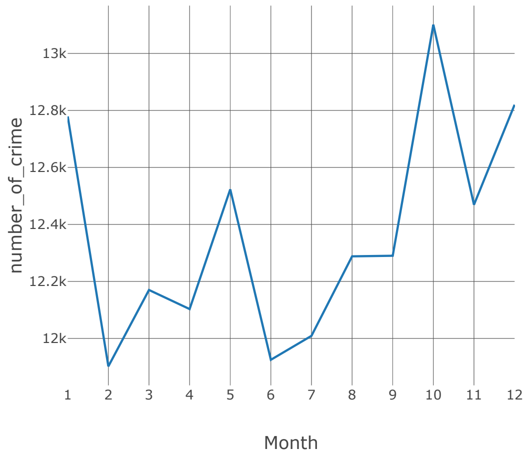
2015

```
display(q4_2015_month)
```



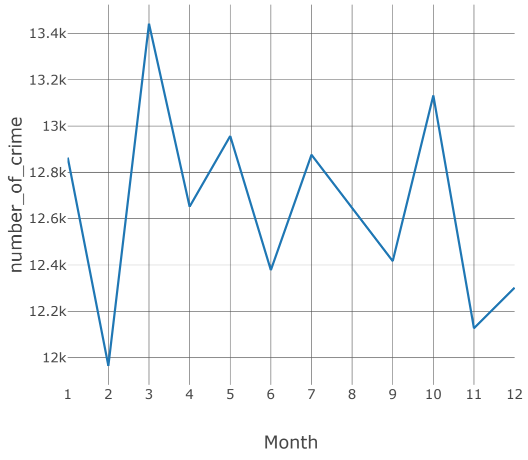
2016

```
display(q4_2016_month)
```



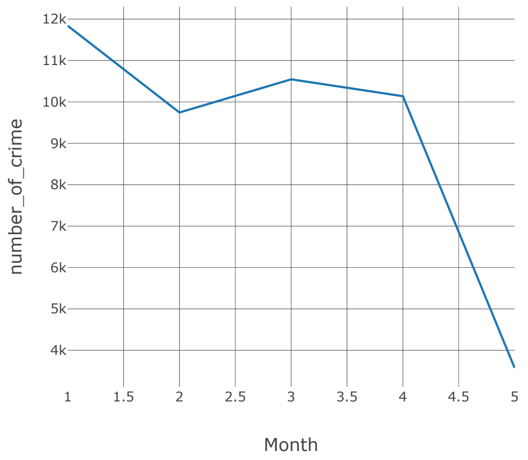
2017

```
display(q4_2017_month)
```



2018

```
display(q4_2018_month)
```

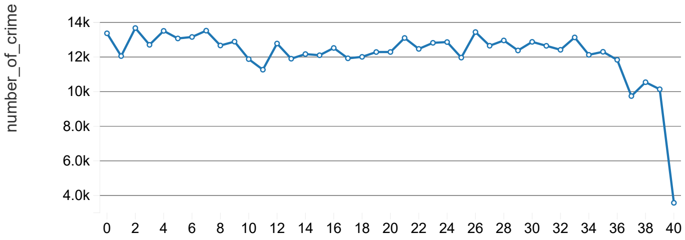


show 2015 to 2018

```
q4_2015_month.createOrReplaceTempView('sf_crime_2015_month')
q4_2016_month.createOrReplaceTempView('sf_crime_2016_month')
q4_2017_month.createOrReplaceTempView('sf_crime_2017_month')
q4_2018_month.createOrReplaceTempView('sf_crime_2018_month')
```

```
Q4_union = spark.sql("select Month, number_of_crime from sf_crime_2015_month union select Month, number_of_crime from sf_crime_2016_month union
select Month, number_of_crime from sf_crime_2017_month union select Month, number_of_crime from sf_crime_2018_month")
```

```
display(Q4_union)
```



Comment of Q4: For the first 3 years(2015, 2016 and 2017) the number of crime in each month is relatively stable,

however, in 2018, it decreases sharply. This may lead to the resurrection of Physical business.

2.5 Analysis the number of crime w.r.t the hour in certian day like 2015/12/15, 2016/12/15, 2017/12/15. Then, give your travel suggestion to visit SF.

I choose 2015/12/15 to analysis.

```
df_q5 = spark.sql("select * from sf_crime \
                   where month(date) == '12' and day(date) == '15' and year(date) == '2015' \
                   ")
```

```
display(df_q5)
```

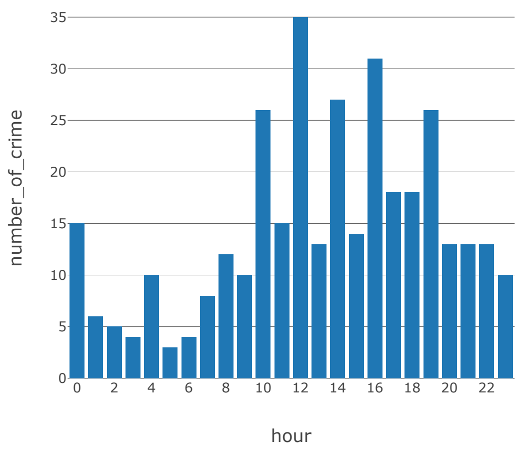
	PdId	IncidntNum	Incident Code	Category	Descript	DayO
1	15108112119057	151081121	19057	ASSAULT	THREATS AGAINST LIFE	Tuesd
2	16101659460190	161016594	60190	SUICIDE	SUICIDE BY STRANGULATION	Tuesd
3	16607328206372	166073282	06372	LARCENY/THEFT	PETTY THEFT OF PROPERTY	Tuesd
4	16601312506372	166013125	06372	LARCENY/THEFT	PETTY THEFT OF PROPERTY	Tuesd
5	16602133871000	166021338	71000	NON-CRIMINAL	LOST PROPERTY	Tuesd
6	15108105227195	151081052	27195	TRESPASS	TRESPASSING	Tuesd
7	15108368804012	151083688	04012	ASSAULT	AGGRAVATED ASSAULT WITH A KNIFE	Tuesd
8	15110637172000	151106371	72000	NON-CRIMINAL	FOUND PROPERTY	Tuesd

Showing all 349 rows.

```
df_q5.createOrReplaceTempView('sf_q5')
```

```
df_q5_hour = spark.sql("select hour(time) as hour, count(*) as number_of_crime from sf_q5 \
                        group by hour \
                        order by hour asc")
```

```
display(df_q5_hour)
```



Suggestion: According to the plot, we can find that there is risk of crime for every hours, however, the hours in range 10 to 19 have the most frequent rate of crime, should take care when visiting SF at this time range.

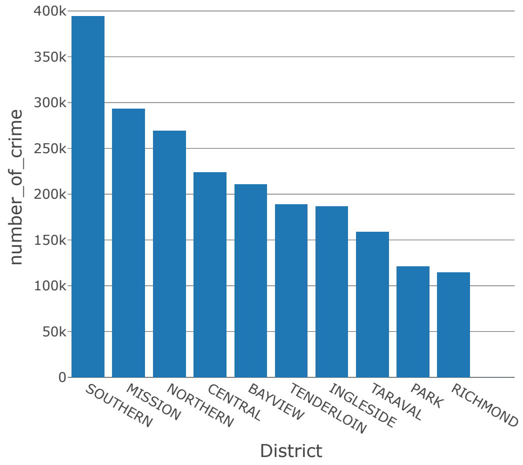
2.6 Advice to distribute the police

(1) Step1: Find out the top-3 danger disrict
(2) Step2: find out the crime event w.r.t category and time (hour) from the result of step 1
(3) give your advice to distribute the police based on your analysis results.

(1) Step 1

```
df_Q6 = spark.sql("select pddistrict as District, count(*) as number_of_crime \
                   from sf_crime \
                   group by District \
                   order by number_of_crime desc")
```

```
display(df_Q6)
```

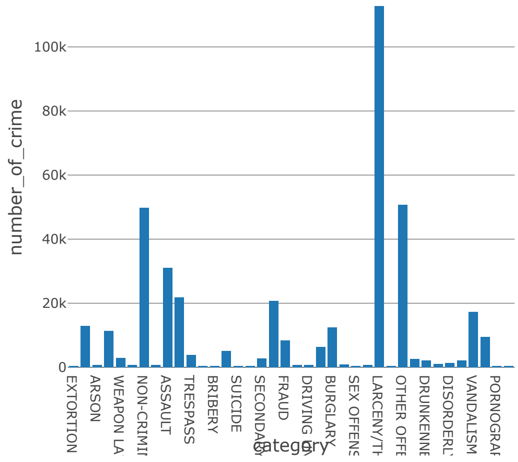


The most dangerous 3 district is 'Southern', 'Mission' and 'Northern'.

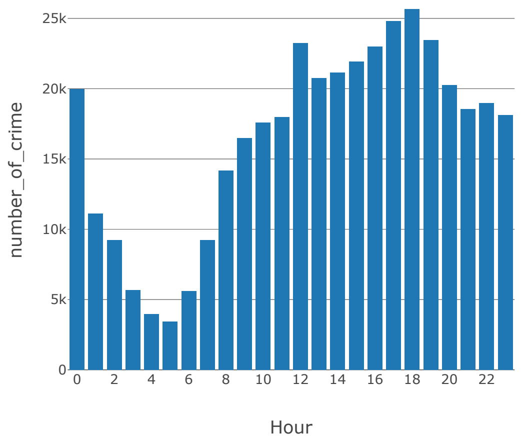
(2) Step 2

Southern

```
df_Q6_Southern = spark.sql("select category, hour(time) as Hour, count(*) as number_of_crime \
                            from sf_crime \
                            where pddistrict == 'SOUTHERN' \
                            group by category, Hour \
                            order by Hour asc \
                            ")
display(df_Q6_Southern)
```

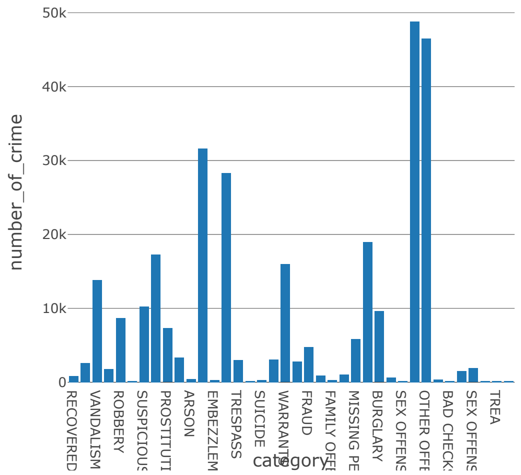


```
display(df_Q6_Southern)
```

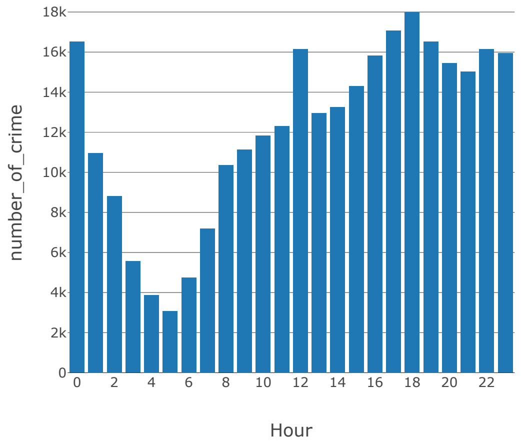


Mission

```
df_Q6_Mission = spark.sql("select category, hour(time) as Hour, count(*) as number_of_crime \
                           from sf_crime \
                           where pddistrict == 'MISSION' \
                           group by category, Hour \
                           order by Hour asc \
                           ")
display(df_Q6_Mission)
```

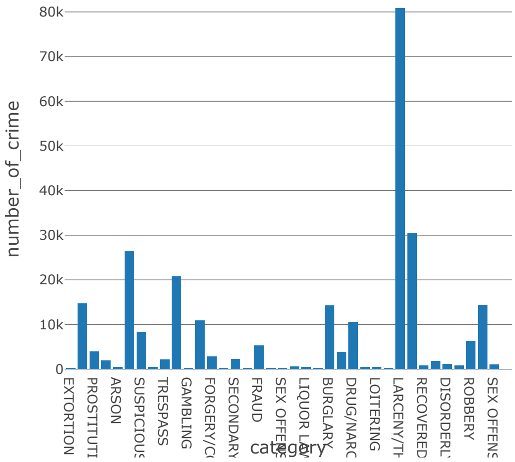


```
display(df_Q6_Mission)
```

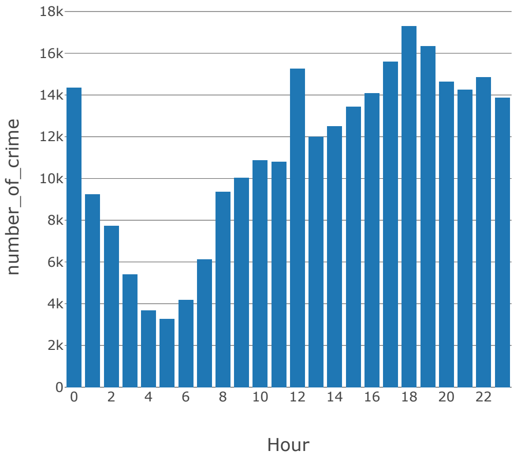


Northern

```
df_Q6_Northern = spark.sql("select category, hour(time) as Hour, count(*) as number_of_crime \
                             from sf_crime \
                             where pddistrict == 'NORTHERN' \
                             group by category, Hour \
                             order by Hour asc \
                             ")
display(df_Q6_Northern)
```



```
display(df_Q6_Northern)
```



Advice to distribute the police:

1. For the 3 district, the most frequent crime type is 'Larceny/Theft'.

2. For the 3 district, the most of the crimes happen during 14-20 hour.

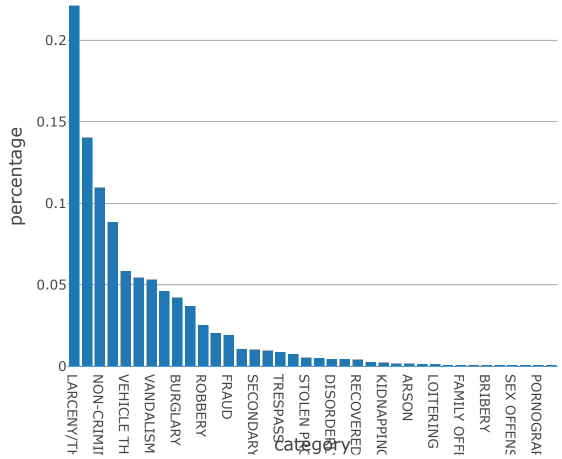
2.7 For different category of crime, find the percentage of them. Based on the output, give my hints to adjust the policy.

```
df_Q7_total = spark.sql("select count(*) as total \
                          from sf_crime")
display(df_Q7_total)
```

	total
1	2160953

Showing all 1 rows.

```
df_Q7 = spark.sql("select category, count(*)/2160953 as percentage \
                  from sf_crime \
                  group by category \
                  order by percentage desc")
display(df_Q7)
```



Hint 1: Larceny/Theft is the most common crime among all of those crimes.

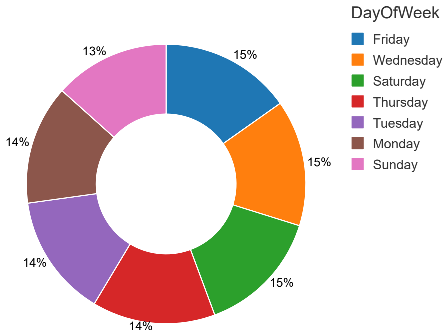
Hint 2: Nearly half of the crime is contributed by 'Larceny/Theft', 'Other offenses' and 'Non-criminal'.

Police should pay more attention to the categories mentioned above.

2.8 For different weekdays, find the percentage of resolution.

```
df_Q8 = spark.sql("select DayOfWeek, count(*)/2160953 as percentage \
                   from sf_crime \
                   group by DayOfWeek \
                   order by percentage desc")


display(df_Q8)
```



Comment: We can find that the frequency of crime on each weekdays is even and it is slightly higher on Friday, Wednesday and Saturday.

3. Conclusion and Suggestions

This project is a kind of data analysis work.

The goal of this is to discover whether there are some hidden law or relationship between the crime(amount, type...) and all of the features.(District, time, date...)

In this project, I apply Python spark and SQL to analyze these data via the data structure of Dataframe.

To analyze, I also finish some Online Analytical Processing(OLAP) to find the hidden relationship and make some plots to visualize them.

Among the results I get, I think the most valuable information can be summarized as the following 4 points:

1. The district 'Southern', 'Mission' and 'Northern' have the highest frequency of crime, the police should pay more attention and strengthen the police at these district.
2. 'Larceny/Theft' is the most frequent type of crime, the police should pay more attention and tell the residents to pay attention to this type of crime.
3. 14-20 is the time range that have the most frequent crime happening, the police should strengthen the police at that time range and tell the residents to pay more attention at that time range.
4. Roughly and on the whole, the frequency of crime is decreasing from 2018, however, that may because of the data missing on this year, more data for this year is needed.