CP1370
Lab 4
John-Michael Woodrow

1.

Just now (16s) 1 Python

```python
from pyspark.sql.functions import to_date, col, desc, count, lower, lit

data = spark.read.csv('dbfs:/FileStore/crimes_2017.csv', header=True, inferSchema=True).withColumn('Date', to_date(col('Date'), 'MM/dd/yyyy hh:mm:ss a'))
data.show(5)
```

(3) Spark Jobs
- Job 0 View (Stages: 1/1)
- Job 1 View (Stages: 1/1)
- Job 2 View (Stages: 1/1)

data: pyspark.sql.dataframe.DataFrame = [ID: integer, Case Number: string ... 20 more fields]

```
+--------+-----------+----------+--------------------+----+--------------------+--------------------+--------------------+------+--------+----+--------+----+--------------+--------+--
----------+------------+----+--------------------+--------+---------+--------+
|      ID|Case Number|      Date|               Block|IUCR|        Primary Type|         Description|Location Description|Arrest|Domestic|Beat|District|Ward|Community Area|FBI Code|X
Coordinate|Y Coordinate|Year|          Updated On|Latitude|Longitude|Location|
+--------+-----------+----------+--------------------+----+--------------------+--------------------+--------------------+------+--------+----+--------+----+--------------+--------+--
----------+------------+----+--------------------+--------+---------+--------+
|11227634|   JB147599|2017-08-26| 001XX W RANDOLPH ST|0281|CRIM SEXUAL ASSAULT|      NON-AGGRAVATED|         HOTEL/MOTEL| false|   false| 122|       1|  42|            32|      02|
null|        null|2017|02/11/2018 03:57:...|    null|     null|    null|
|11037549|   JA371812|2017-07-30| 004XX N DEARBORN ST|0870|               THEFT|      POCKET-PICKING|               ALLEY| false|   false|1831|      18|  42|             8|      06|
null|        null|2017|08/06/2017 03:53:...|    null|     null|    null|
|11228565|   JB148941|2017-12-16|102XX S EBERHART AVE|0610|            BURGLARY|      FORCIBLE ENTRY|     VACANT LOT/LAND| false|   false| 511|       5|   9|            49|      05|
null|        null|2017|02/12/2018 03:49:...|    null|     null|    null|
|11243119|   JB168573|2017-11-02|   063XX N MOZART ST|2825|       OTHER OFFENSE|HARASSMENT BY TEL...|           APARTMENT| false|   false|2413|      24|  50|             2|      26|
null|        null|2017|03/01/2018 03:54:...|    null|     null|    null|
|11229320|   JB150109|2017-09-29|058XX N CUMBERLAN...|0820|               THEFT|      $500 AND UNDER|         CTA STATION| false|   false|1614|      16|  41|            76|      06|
null|        null|2017|02/13/2018 03:58:...|    null|     null|    null|
+--------+-----------+----------+--------------------+----+--------------------+--------------------+--------------------+------+--------+----+--------+----+--------------+--------+--
----------+------------+----+--------------------+--------+---------+--------+
only showing top 5 rows
```

2.

Just now (8s) 2 Python

```python
# Group by date and count the number of crimes per day
top_dates = data.groupBy('Date').count().orderBy(desc('count'))

# Display top 5 dates with the most reported crimes
top_dates.show(5)
```

(2) Spark Jobs

top_dates: pyspark.sql.dataframe.DataFrame = [Date: date, count: long]

```
+----------+-----+
|      Date|count|
+----------+-----+
|2017-01-01| 1256|
|2017-08-01|  965|
|2017-07-01|  937|
|2017-08-05|  928|
|2017-08-04|  926|
+----------+-----+
only showing top 5 rows
```

3.

Just now (9s) 3 Python

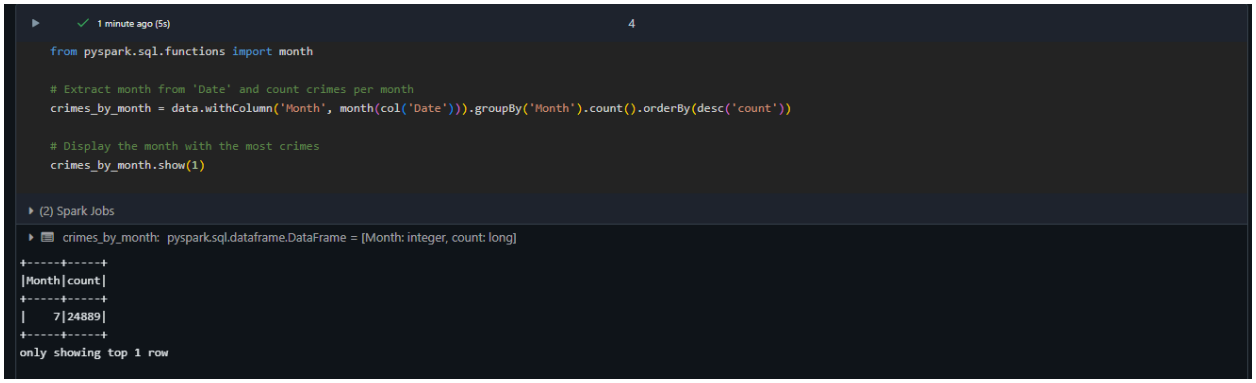
```python
# Get the date with the most reported crimes
most_crimes_date = top_dates.first()['Date']

# Filter data for that date and count occurrences of each crime
top_crimes = data.filter(col('Date') == lit(most_crimes_date)).groupBy('Primary Type').count().orderBy(desc('count'))

# Display top 3 crimes on the most crime-heavy day
top_crimes.show(3)
```

(4) Spark Jobs

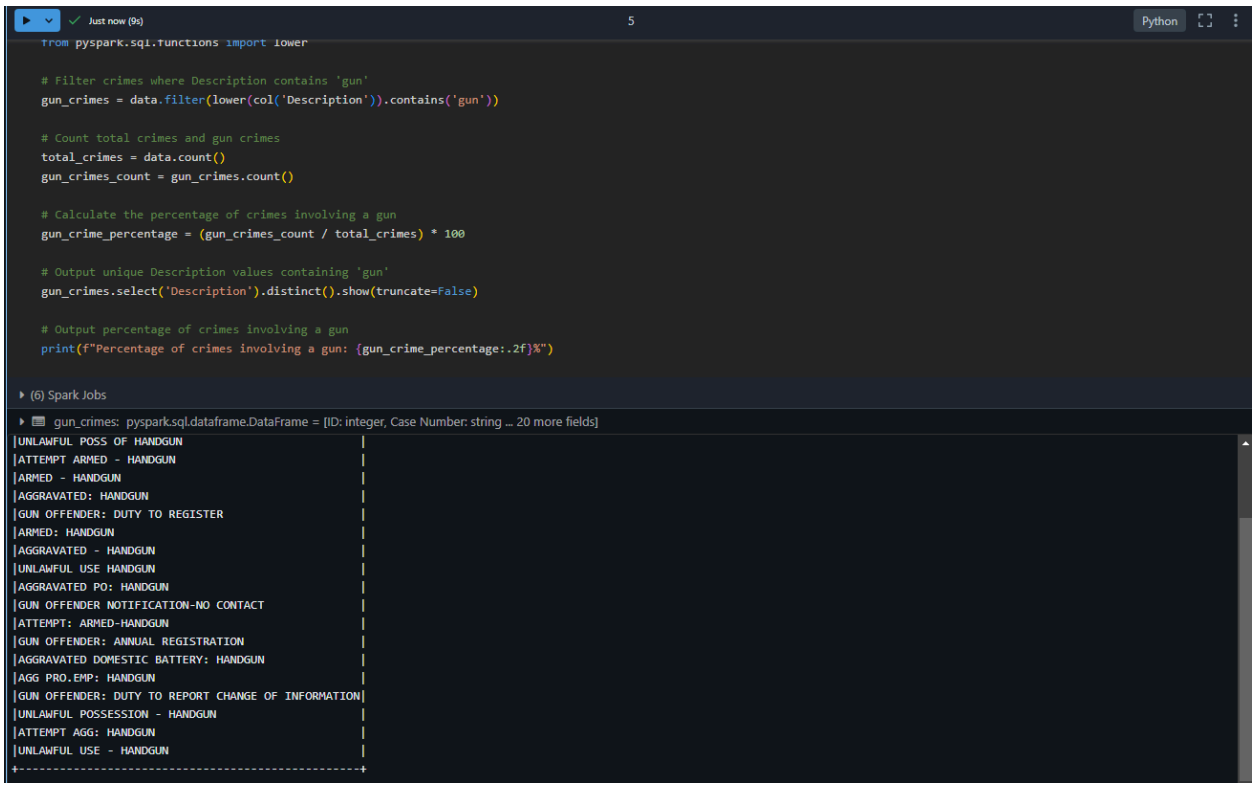
top_crimes: pyspark.sql.dataframe.DataFrame = [Primary Type: string, count: long]

```
+-----------------+-----+
|     Primary Type|count|
+-----------------+-----+
|DECEPTIVE PRACTICE|  208|
|          BATTERY|  207|
|            THEFT|  187|
+-----------------+-----+
only showing top 3 rows
```

4.


1 minute ago (5s)
4
from pyspark.sql.functions import month
# Extract month from 'Date' and count crimes per month
crimes_by_month = data.withColumn('Month', month(col('Date'))).groupBy('Month').count().orderBy(desc('count'))
# Display the month with the most crimes
crimes_by_month.show(1)
(2) Spark Jobs
crimes_by_month: pyspark.sql.dataframe.DataFrame = [Month: integer, count: long]
+-----+-----+
|Month|count|
+-----+-----+
| 7|24889|
+-----+-----+
only showing top 1 row

5.


Just now (9s)
5
Python
from pyspark.sql.functions import lower
# Filter crimes where Description contains 'gun'
gun_crimes = data.filter(lower(col('Description')).contains('gun'))
# Count total crimes and gun crimes
total_crimes = data.count()
gun_crimes_count = gun_crimes.count()
# Calculate the percentage of crimes involving a gun
gun_crime_percentage = (gun_crimes_count / total_crimes) * 100
# Output unique Description values containing 'gun'
gun_crimes.select('Description').distinct().show(truncate=False)
# Output percentage of crimes involving a gun
print(f"Percentage of crimes involving a gun: {gun_crime_percentage:.2f}%")
(6) Spark Jobs
gun_crimes: pyspark.sql.dataframe.DataFrame = [ID: integer, Case Number: string ... 20 more fields]
|UNLAWFUL POSS OF HANDGUN |
|ATTEMPT ARMED - HANDGUN |
|ARMED - HANDGUN |
|AGGRAVATED: HANDGUN |
|GUN OFFENDER: DUTY TO REGISTER |
|ARMED: HANDGUN |
|AGGRAVATED - HANDGUN |
|UNLAWFUL USE HANDGUN |
|AGGRAVATED PO: HANDGUN |
|GUN OFFENDER NOTIFICATION-NO CONTACT |
|ATTEMPT: ARMED-HANDGUN |
|GUN OFFENDER: ANNUAL REGISTRATION |
|AGGRAVATED DOMESTIC BATTERY: HANDGUN |
|AGG PRO.EMP: HANDGUN |
|GUN OFFENDER: DUTY TO REPORT CHANGE OF INFORMATION|
|UNLAWFUL POSSESSION - HANDGUN |
|ATTEMPT AGG: HANDGUN |
|UNLAWFUL USE - HANDGUN |
+-----+