Untitled

December 24, 2023

```
[1]: %%info
```

```
<IPython.core.display.HTML object>

<IPython.core.display.HTML object>
```

```
[2]: sc.install_pypi_package("matplotlib==3.2.1")
```

```
VBox()

Starting Spark application

<IPython.core.display.HTML object>

FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
 ↪layout=Layout(height='25px', width='50%'),…

SparkSession available as 'spark'.

FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
 ↪layout=Layout(height='25px', width='50%'),…
```

```
Collecting matplotlib==3.2.1
  Downloading matplotlib-3.2.1-cp37-cp37m-manylinux1_x86_64.whl (12.4 MB)
Collecting cycler>=0.10
  Downloading cycler-0.11.0-py3-none-any.whl (6.4 kB)
Collecting pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1
  Downloading pyparsing-3.1.1-py3-none-any.whl (103 kB)
Requirement already satisfied: numpy>=1.11 in /usr/local/lib64/python3.7/site-
packages (from matplotlib==3.2.1) (1.20.0)
Collecting python-dateutil>=2.1
  Downloading python_dateutil-2.8.2-py2.py3-none-any.whl (247 kB)
Collecting kiwisolver>=1.0.1
  Downloading
kiwisolver-1.4.5-cp37-cp37m-manylinux_2_5_x86_64.manylinux1_x86_64.whl (1.1 MB)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/site-
packages (from python-dateutil>=2.1->matplotlib==3.2.1) (1.13.0)
Collecting typing-extensions; python_version < "3.8"
  Downloading typing_extensions-4.7.1-py3-none-any.whl (33 kB)
Installing collected packages: cycler, pyparsing, python-dateutil, typing-
extensions, kiwisolver, matplotlib
```

```
Successfully installed cycler-0.11.0 kiwisolver-1.4.5 matplotlib-3.2.1
pyparsing-3.1.1 python-dateutil-2.8.2 typing-extensions-4.7.1

WARNING: The directory '/home/.cache/pip' or its parent directory is not owned
or is not writable by the current user. The cache has been disabled. Check the
permissions and owner of that directory. If executing pip with sudo, you may
want sudo's -H flag.
```

[3]:

```python
df = spark.read.csv('s3://cis9760-athena-queries-rn/Unsaved/2023/12/23/
 ↪0a177fb6-a3dc-4249-b6f4-ec12de7aa045.csv', header=True)
```

```
VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
 ↪layout=Layout(height='25px', width='50%'),…
```

[4]:

```python
df.printSchema()
```

```
VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
 ↪layout=Layout(height='25px', width='50%'),…

root
 |-- COMP: string (nullable = true)
 |-- DATE: string (nullable = true)
 |-- AVG_VOL: string (nullable = true)
 |-- MAX_VOL: string (nullable = true)
 |-- MIN_VOL: string (nullable = true)
```

[5]:

```python
from pyspark.sql.functions import col

df = df.withColumn("AVG_VOL", col("AVG_VOL").cast("double"))
df = df.withColumn("MAX_VOL", col("MAX_VOL").cast("double"))
df = df.withColumn("MIN_VOL", col("MIN_VOL").cast("double"))
```

```
VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
 ↪layout=Layout(height='25px', width='50%'),…
```

[ ]:

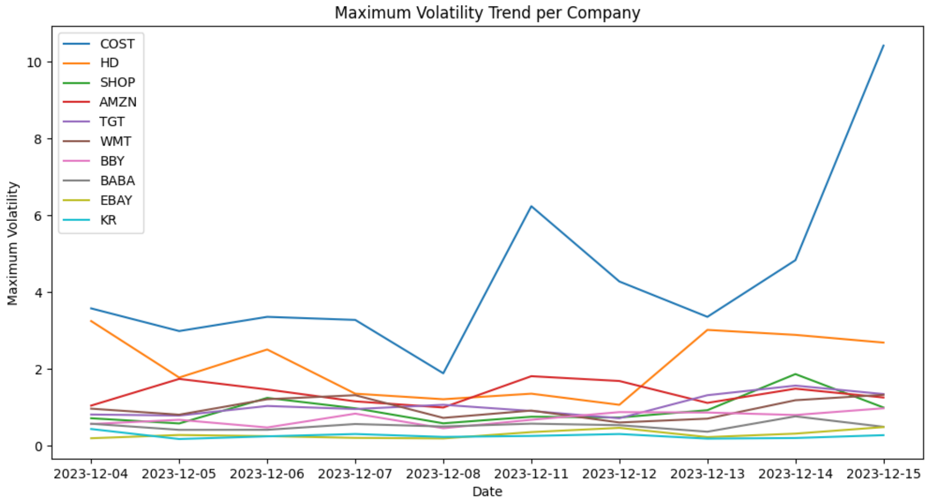
```python
sc.install_pypi_package("pandas==1.0.5")
```

[20]:

```python
#graph 2
from pyspark.sql import SparkSession
import matplotlib.pyplot as plt
from pyspark.sql.functions import col, avg, desc


max_volatility_per_company = df.groupBy("COMP").agg({"MAX_VOL": "max"})
```

```
max_volatility_ordered = max_volatility_per_company.
  ↪orderBy(desc('max(MAX_VOL)'))

result_pd_df = max_volatility_ordered.toPandas()

plt.figure(figsize=(12, 6))
for company in result_pd_df['COMP']:
    company_data = df.filter(col("COMP") == company).orderBy("DATE").toPandas()
    plt.plot(company_data['DATE'], company_data['MAX_VOL'], label=company)

plt.xlabel("Date")
plt.ylabel("Maximum Volatility")
plt.title("Maximum Volatility Trend per Company")
plt.legend()
plt.show()
%matplot plt
```

```
VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
 ↪layout=Layout(height='25px', width='50%'),…
```



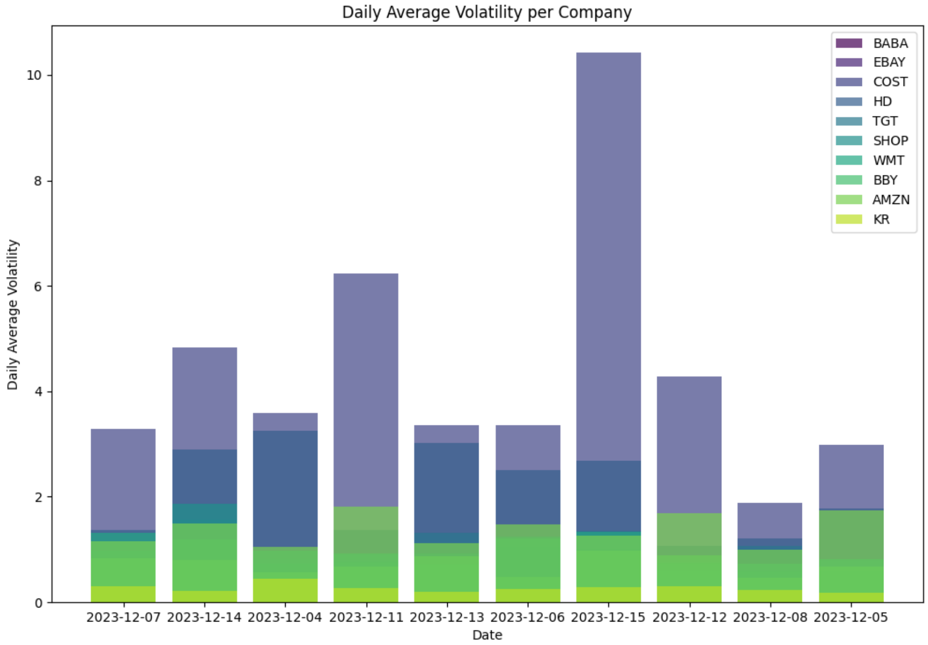
[19]:

```
# graph 2
from pyspark.sql import SparkSession
from pyspark.sql.functions import col, avg
import matplotlib.pyplot as plt

spark = SparkSession.builder.appName("VolatilityAnalysis").getOrCreate()
```

```
daily_avg_volatility = df.groupBy("COMP", "DATE").agg(avg("MAX_VOL").
 ↪alias("DAILY_AVG_VOL"))

result_pd_df = daily_avg_volatility.toPandas()

plt.figure(figsize=(12, 8))
companies = result_pd_df['COMP'].unique()

for i, company in enumerate(companies):
    company_data = result_pd_df[result_pd_df['COMP'] == company]
    plt.bar(
        company_data['DATE'],
        company_data['DAILY_AVG_VOL'],
        label=company,
        alpha=0.7,
        color=plt.cm.viridis(i / len(companies))
    )

plt.xlabel("Date")
plt.ylabel("Daily Average Volatility")
plt.title("Daily Average Volatility per Company")
plt.legend()
plt.show()
%matplot plt
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
 ↪layout=Layout(height='25px', width='50%'),…
```


Daily Average Volatility per Company
Daily Average Volatility
0
2
4
6
8
10
2023-12-07
2023-12-14
2023-12-04
2023-12-11
2023-12-13
2023-12-06
2023-12-15
2023-12-12
2023-12-08
2023-12-05
Date
BABA
EBAY
COST
HD
TGT
SHOP
WMT
BBY
AMZN
KR

```
[ ]:
```