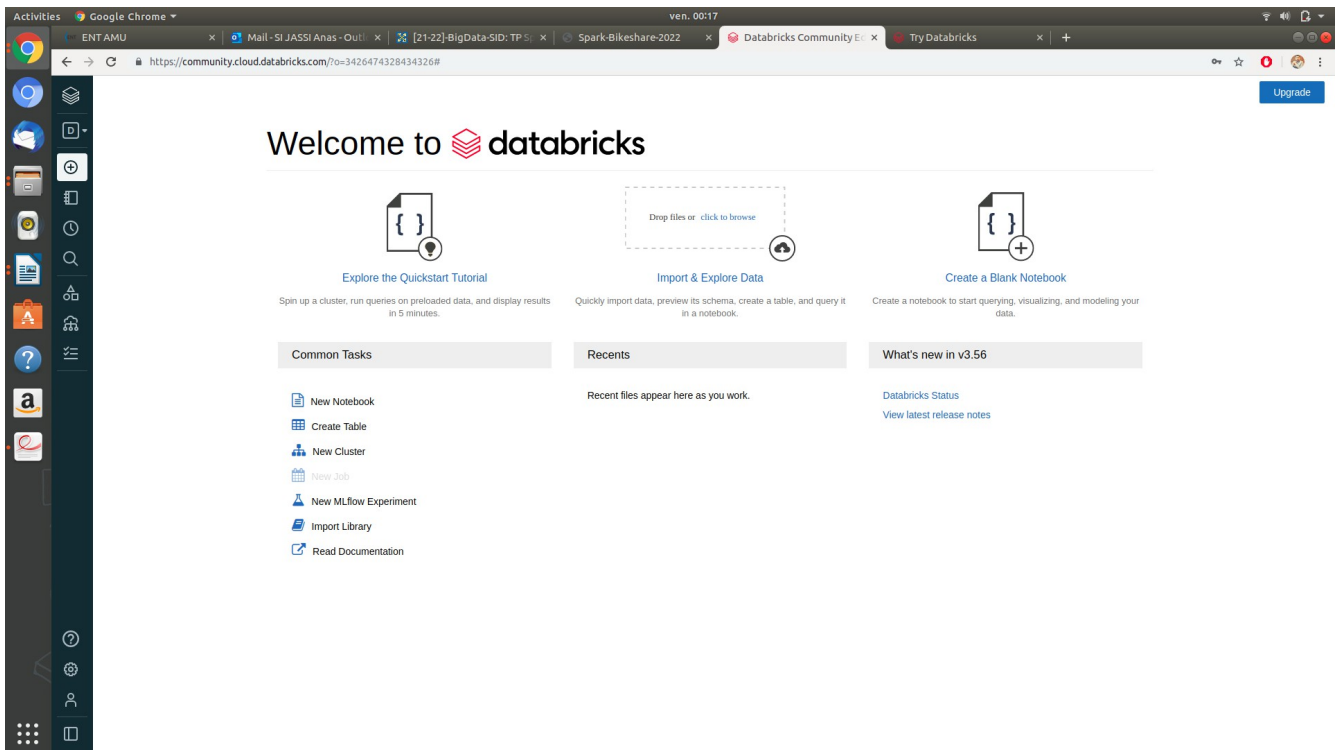


# Compte Rendu TP SPARK | DataBricks

***Par Anas SI JASSI***

## Partie A :

Pour la question 1 - 2 - 3 : La page afficher apres la creation du compte



Question 4 : Quickstart notebook

Activities Google Chrome ven.. 00:22

ENT AMU Mail - SI JASSI Anas - Out [21:22] BigData-SID: TP Spark-Bikeshare-2022 Quickstart Notebook - D Try Databricks

https://community.cloud.databricks.com/?o=3426474328434326#notebook/1404313059608902/command/1404313059608903

Quickstart Notebook (SQL)

Detached Cmd 1

## Databricks in 5 minutes



Cmd 2

### Create a quickstart cluster

1. In the sidebar, right-click the **Compute** button and open the link in a new window.
2. On the Clusters page, click **Create Cluster**.
3. Name the cluster **Quickstart**.
4. In the Databricks Runtime Version drop-down, select **7.3 LTS (Scala 2.12, Spark 3.0.1)**.
5. Click **Create Cluster**.

Cmd 3

### Attach the notebook to the cluster and run all commands in the notebook

1. Return to this notebook.
2. In the notebook menu bar, select **⚙ Detached** > **Quickstart**.
3. When the cluster changes from  to , click **⏏ Run All**.

Cmd 4

### The next command creates a table from a Databricks dataset

Cmd 5

```
1 DROP TABLE IF EXISTS diamonds;
2
3 CREATE TABLE diamonds
4 USING csv
5 OPTIONS (path "/databricks-datasets/Rdatasets/data-001/csv/ggplot2/diamonds.csv", header "true")
6
```

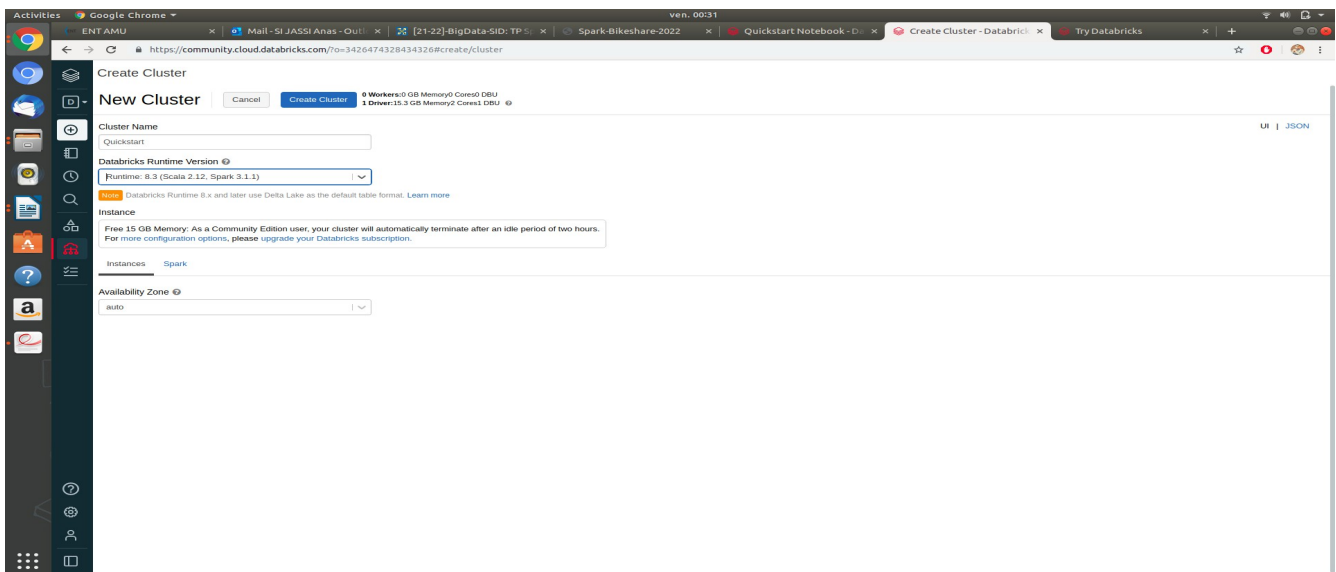
OK

Command took 1.35 seconds -- by a user at 6/15/2021, 7:13:29 PM on unknown cluster

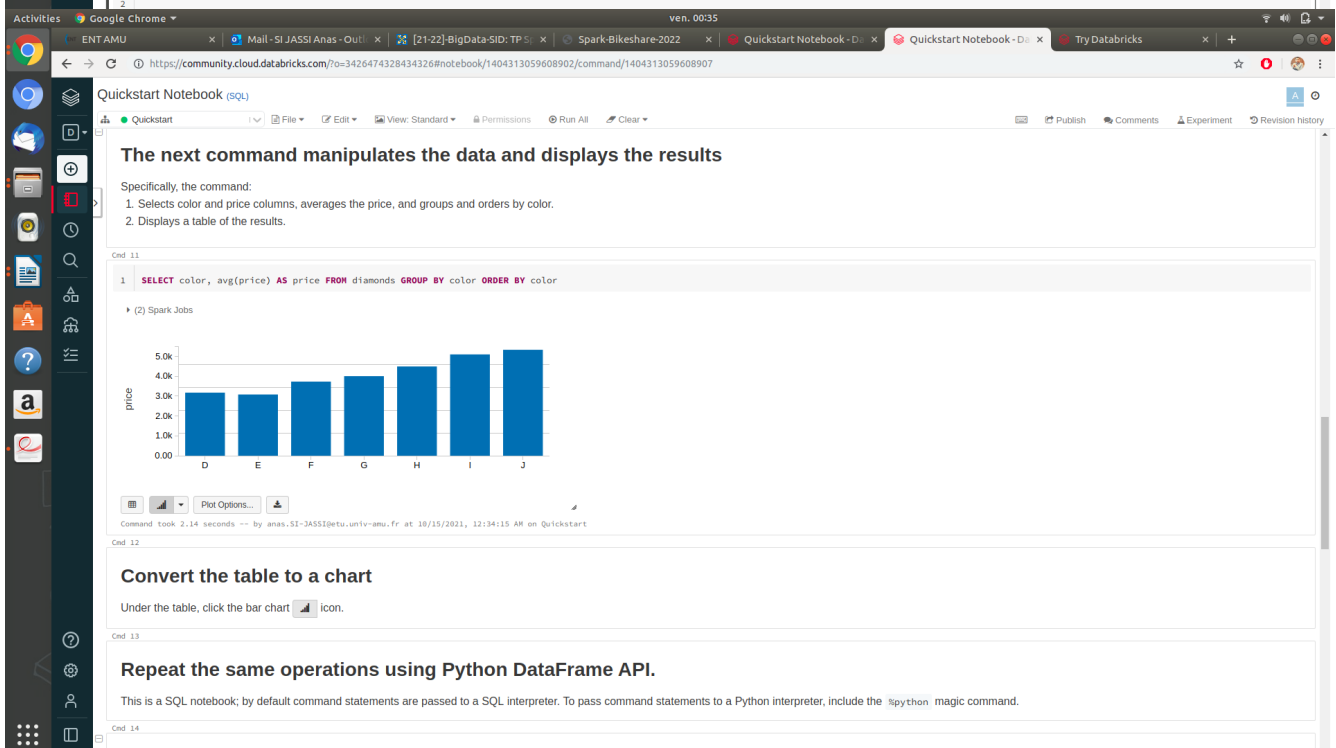
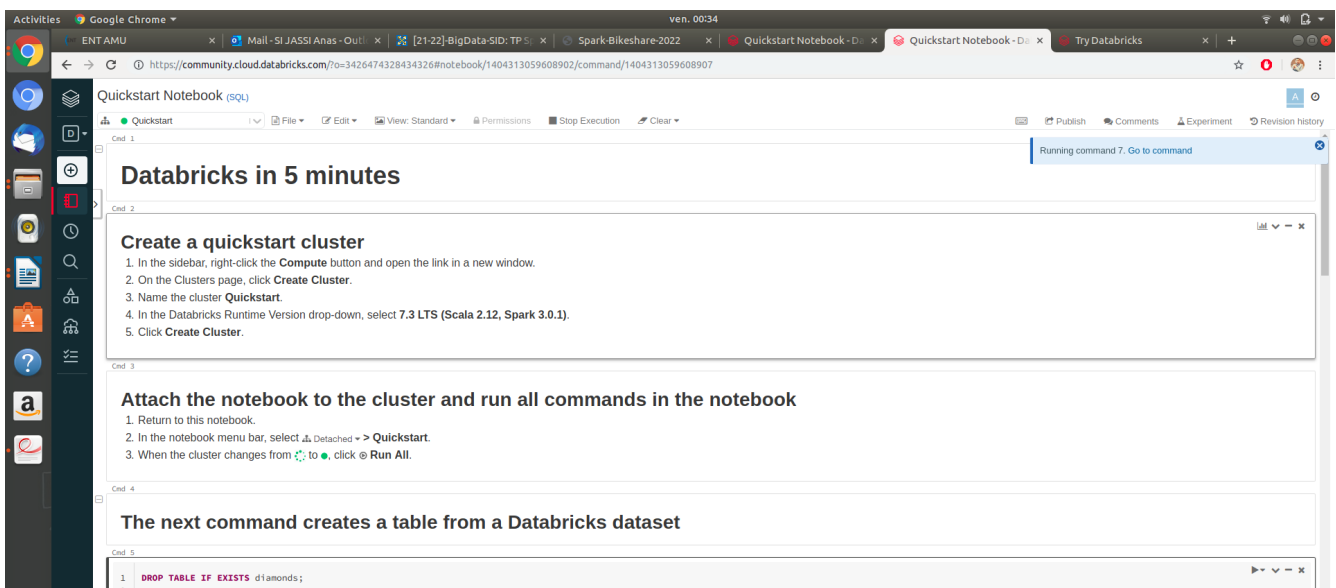
Cmd 6

```
1 SELECT * from diamonds
```

**Create a quickstart cluster :**

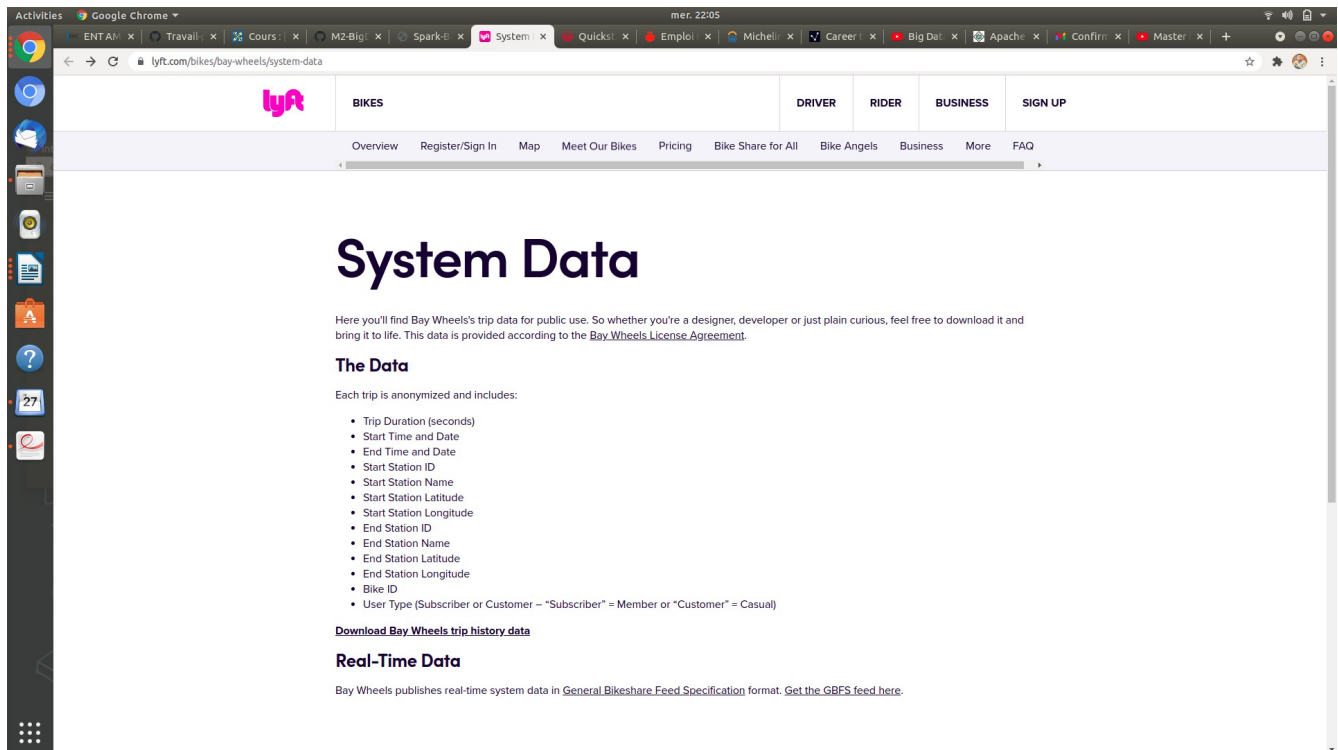


**Attach the notebook to the cluster and run all commands in the notebook**



## ***Partie B :***

1) sur le site <https://www.lyft.com/bikes/bay-wheels/system-data> :



2) *dataset télécharger est la suivante* : " [201802-fordgobike-tripdata.csv.zip](#) " lien :

3) Insertion des données via une commande :

2021-10-20 - DBFS Example (Python)

Overview

This notebook will show you how to create and query a table or DataFrame that you uploaded to DBFS. DBFS is a Databricks File System that allows you to store data for querying inside of Databricks. This notebook assumes that you have a file already inside of DBFS that you would like to read from.

This notebook is written in Python so the default cell type is Python. However, you can use different languages by using the `%LANGUAGE` syntax. Python, Scala, SQL, and R are all supported.

```

1 # File location and type
2 file_location = "/FileStore/tables/201802_fordgobike_tripdata.csv"
3 file_type = "csv"
4
5 # CSV options
6 infer_schema = "false"
7 first_row_is_header = "false"
8 delimiter = ","
9
10 # The applied options are for CSV files. For other file types, these will be ignored.
11 df = spark.read.format(file_type) \
12   .option("inferSchema", infer_schema) \
13   .option("header", first_row_is_header) \
14   .option("sep", delimiter) \
15   .load(file_location)
16
17 display(df)

```

```

1 # Create a view or table
2
3 temp_table_name = "201802_fordgobike_tripdata_csv"
4
5 df.createOrReplaceTempView(temp_table_name)

```

```

1 %sql
2
3 /* Query the created temp table in a SQL cell */
4

```

201802-fordg...zip   201802-fordg...zip   Show all

2021-10-20 - DBFS Example (Python)

Quickstart

This notebook is written in Python so the default cell type is Python. However, you can use different languages by using the `%LANGUAGE` syntax. Python, Scala, SQL, and R are all supported.

```

1 # File location and type
2 file_location = "/FileStore/tables/201802_fordgobike_tripdata.csv"
3 file_type = "csv"
4
5 # CSV options
6 infer_schema = "false"
7 first_row_is_header = "false"
8 delimiter = ","
9
10 # The applied options are for CSV files. For other file types, these will be ignored.
11 df = spark.read.format(file_type) \
12   .option("inferSchema", infer_schema) \
13   .option("header", first_row_is_header) \
14   .option("sep", delimiter) \
15   .load(file_location)
16
17 display(df)

```

(2) Spark Jobs

df: pyspark.sql.dataframe.DataFrame = [c0, string, c1, string ... 12 more fields]

	c0	c1	c2	c3	c4	c5	c6	c7	c8
1	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_latitude	start_station_longitude	end_station_id	end_station_name
2	598	2018-02-28 23:59:47.0970	2018-03-01 00:09:45.1870	284	Yerba Buena Center for the Arts (Howard St at 3rd St)	37.7848720844	-122.4006756876	114	Rhode Island St at 17th St
3	943	2018-02-28 23:21:16.4950	2018-02-28 23:36:59.9740	6	The Embarcadero at Sansome St	37.80477	-122.403234	324	Union Square (Powell St at Post St)
4	18587	2018-02-28 18:20:55.1900	2018-02-28 23:30:42.9250	93	4th St at Mission Bay Blvd S	37.7704074	-122.3911984	15	San Francisco Ferry Building (Harr
5	18558	2018-02-28 18:20:53.6210	2018-02-28 23:30:12.4500	93	4th St at Mission Bay Blvd S	37.7704074	-122.3911984	15	San Francisco Ferry Building (Harr
6	885	2018-02-28 23:15:12.8580	2018-02-28 23:29:58.6080	308	San Pedro Square	37.336802	-121.8940901	297	Locust St at Grant St
7	601	2018-02-28 23:14:19.1700	2018-02-28 23:29:40.4370	312	San Jose Diridon Station	37.329732	-121.901782	288	Mission St at 1st St

Truncated results, showing first 1000 rows.

Command took 1.96 seconds -- by anas.SI-3A55@etu.univ-amu.fr at 10/20/2021, 10:53:08 PM on Quickstart

```

1 # Create a view or table
2
3 temp_table_name = "201802_fordgobike_tripdata_csv"

```

201802-fordg...zip   201802-fordg...zip   Show all

**Travail a faire :**

création d'un dataFram a partir de notre dataset :

Implémentation partie C : le Travail a faire :

Show code

Cod 7

we will first creates a DataFrame from a Databricks dataset

```
1 %python
2 Tripdata = spark.read.csv("/FileStore/tables/201802_fordgobike_tripdata.csv", header="true", inferSchema="true")
```

▶ (2) Spark Jobs

▶ Tripdata: pyspark.sql.dataframe.DataFrame = [duration\_sec: integer, start\_time: string ... 12 more fields]

Command took 3.65 seconds -- by anas.SI-3A55I@etu.univ-amu.fr at 18/28/2021, 11:18:11 PM on Quickstart

Shift+Enter to run

201802-fordg....zip ^

201802-fordg....zip ^

Show all x

premiere pas avec la visualisation : exploration univariée :

Activities Google Chrome ven. 04:09

community.cloud.databricks.com/?o=3426474328434326#notebook/3702466441651073/command/4051303280496327

2021-10-20 - DBFS Example Python

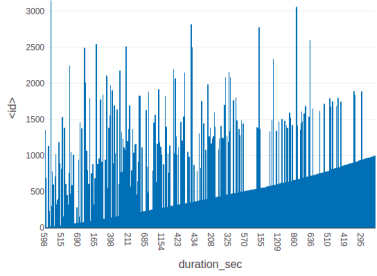
Quickstart File Edit View: Standard Permissions Run All Clear

Cod 17

Univariate Exploration : I'll start by looking at the distribution of the main variable of interest: duration\_sec.

```
1 display(Tripdata.select('duration_sec'))
2
```

▶ (1) Spark Jobs



Showing sample based on the first 1000 rows.

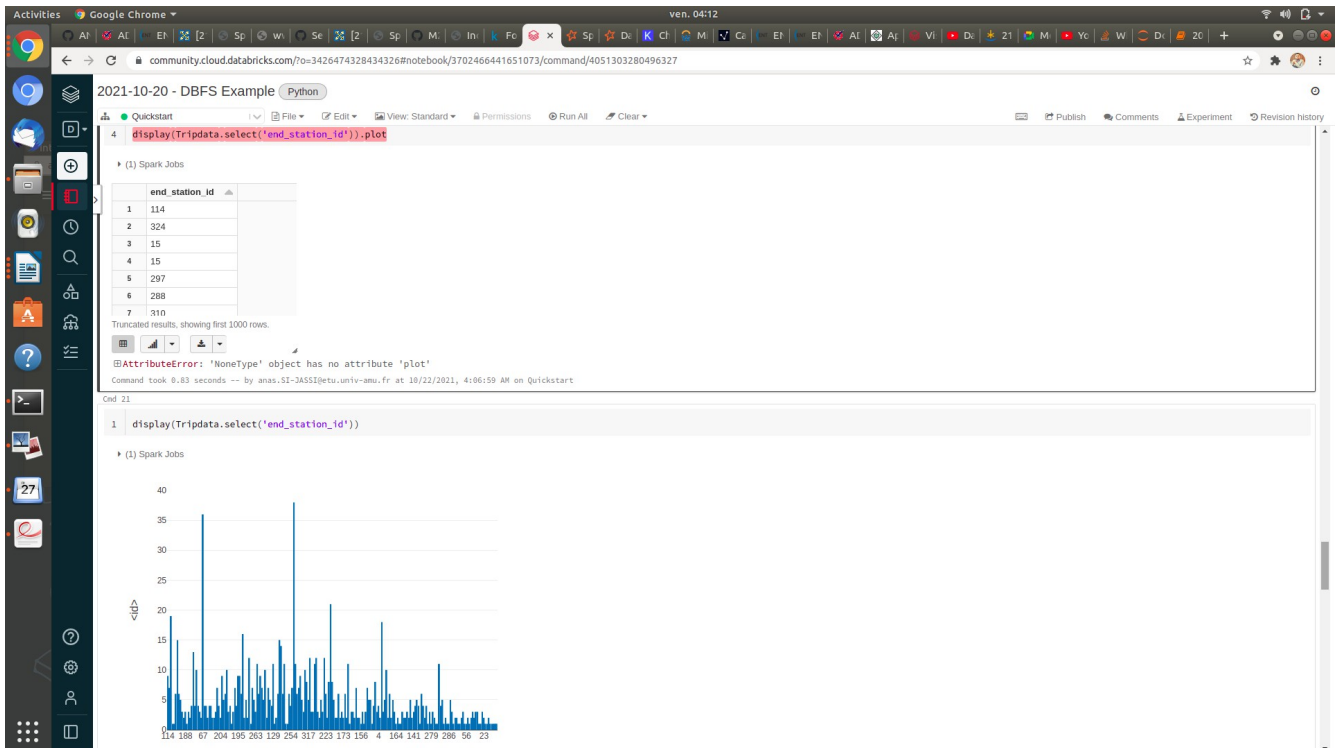
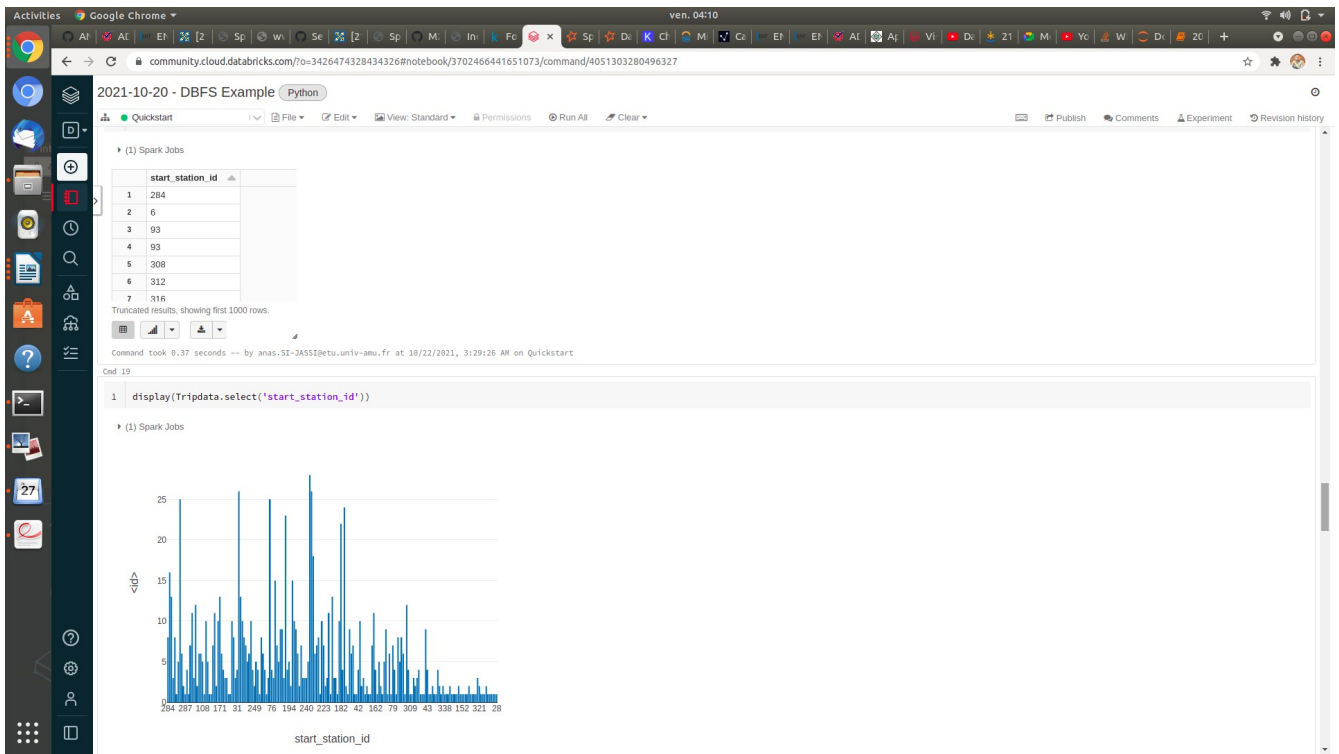
Command took 6.38 seconds -- by anas.SI-3A55I@etu.univ-amu.fr at 18/22/2021, 3:29:26 AM on Quickstart

Cod 18

Now lets look at other factors like start and end station id and birth year

```
1
2
3 display(Tripdata.select('start_station_id'))
```

▶ (1) Spark Jobs



<< la suite est noté sur le notebook >>

Lien notebook databrick :

[https://community.cloud.databricks.com/?  
o=3426474328434326#notebook/3702466441651073/command/405130328049  
6327](https://community.cloud.databricks.com/?o=3426474328434326#notebook/3702466441651073/command/4051303280496327)