# CIS5200 Term Project Tutorial

**Authors: Lekha Ajitkumar, Sushmitha Dandu, Dauren Omarov, Navyasree Sriramoju**

**Instructor: Jongwook Woo**

**Date: 07/12/2022**

## Lab Tutorial

Lekha Ajitkumar (lajitku@calstatela.edu)

Sushmitha Dandu (sdandu@calstatela.edu)

Dauren Omarov (domarov@calstatela.edu)

Navyasree Sriramoju (nsriram@calstatela.edu)

07/12/2022

# Ecommerce Behavior Data from Multi Category Store

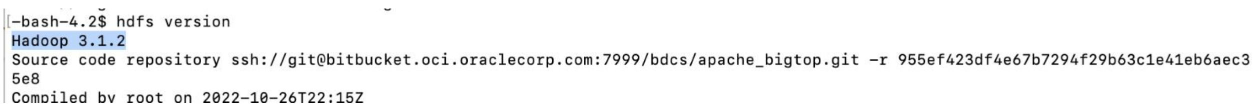
### Objectives

In this hands-on lab, you will learn how to:

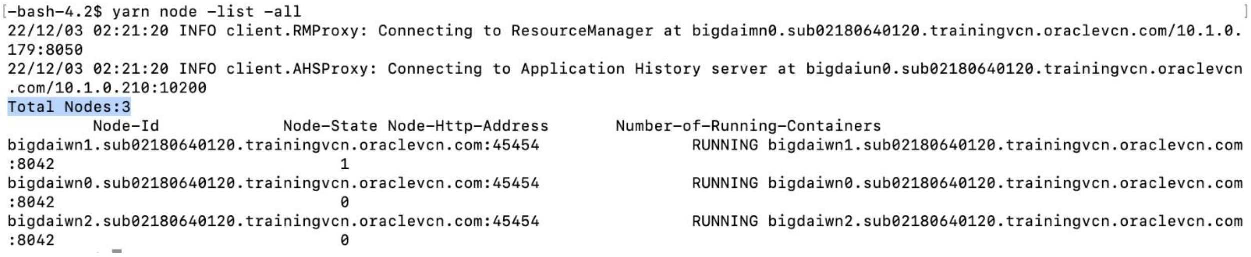
- Download dataset from the Kaggle website
- Using SCP upload the data to the Hadoop cluster
- Create Hive tables in HDFS using HiveQL
- Create HiveQL queries to manipulate and analyze the data
- Visualize the result in Excel, Power BI and Tableau

### Platform Spec

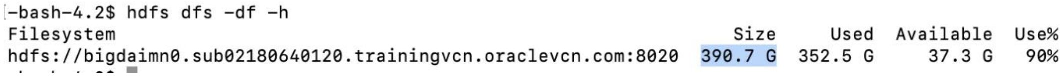
- Cluster Version: Hadoop 3.1.2
- CPU Speed: 1995.309 MHz
- # of CPU cores: 4
- # of nodes: 3
- Total Memory Size: 390.7 GB



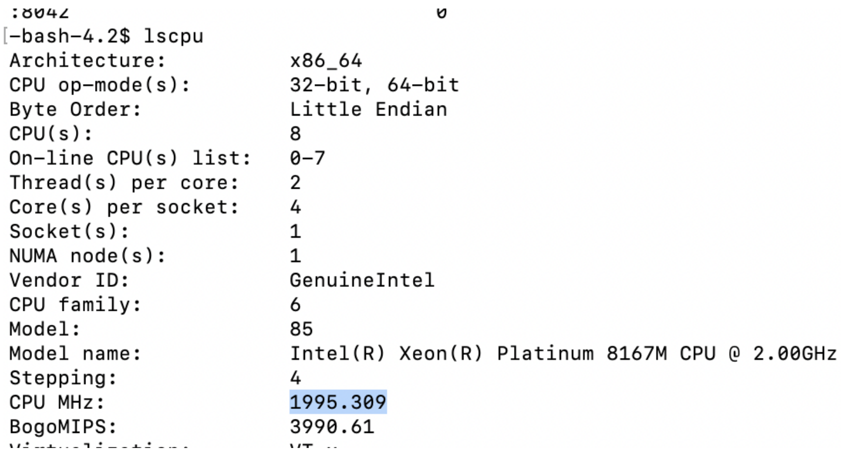
```
-bash-4.2$ hdfs version
Hadoop 3.1.2
Source code repository ssh://git@bitbucket.oci.oraclecorp.com:7999/bdcs/apache_bigtop.git -r 955ef423df4e67b7294f29b63c1e41eb6aec3
5e8
Compiled by root on 2022-10-26T22:15Z
```



```
-bash-4.2$ yarn node -list -all
22/12/03 02:21:20 INFO client.RMProxy: Connecting to ResourceManager at bigdaimn0.sub02180640120.trainingvcn.oraclevcn.com/10.1.0.
179:8050
22/12/03 02:21:20 INFO client.AHSProxy: Connecting to Application History server at bigdaiun0.sub02180640120.trainingvcn.oraclevcn
.com/10.1.0.210:10200
Total Nodes:3
         Node-Id             Node-State Node-Http-Address       Number-of-Running-Containers
bigdaiwn1.sub02180640120.trainingvcn.oraclevcn.com:45454                RUNNING bigdaiwn1.sub02180640120.trainingvcn.oraclevcn.com
:8042                              1
bigdaiwn0.sub02180640120.trainingvcn.oraclevcn.com:45454                RUNNING bigdaiwn0.sub02180640120.trainingvcn.oraclevcn.com
:8042                              0
bigdaiwn2.sub02180640120.trainingvcn.oraclevcn.com:45454                RUNNING bigdaiwn2.sub02180640120.trainingvcn.oraclevcn.com
:8042                              0
```



```
-bash-4.2$ hdfs dfs -df -h
Filesystem                                                      Size     Used  Available  Use%
hdfs://bigdaimn0.sub02180640120.trainingvcn.oraclevcn.com:8020  390.7 G  352.5 G     37.3 G   90%
```



```
:8042                              0
-bash-4.2$ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                8
On-line CPU(s) list:   0-7
Thread(s) per core:    2
Core(s) per socket:    4
Socket(s):             1
NUMA node(s):          1
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 85
Model name:            Intel(R) Xeon(R) Platinum 8167M CPU @ 2.00GHz
Stepping:              4
CPU MHz:               1995.309
BogoMIPS:              3990.61
```

## Dataset Details

- DATASET NAME: Ecommerce Behavior Data from Multi Category Store

- DATASET URL: https://www.kaggle.com/datasets/mkechinov/ecommerce-behavior-data-from-multi-category-store?select=2019-Oct.csv

- TOTAL SIZE: 15.83 GB

- MONTHS CONSIDERED: October and November

- NUMBER OF FILES: 2

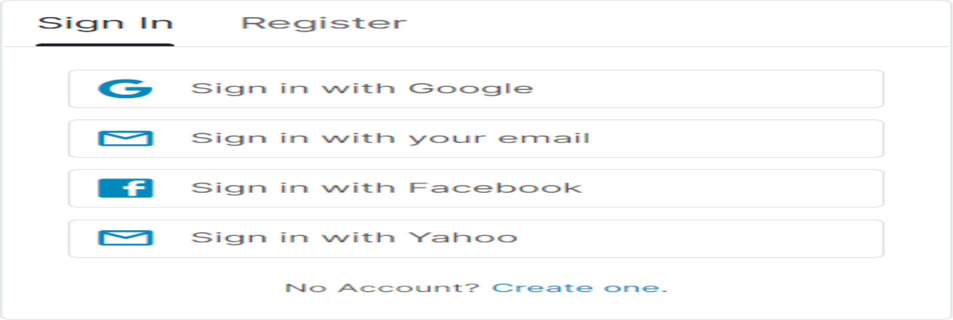
- FILE FORMAT: CSV

## Step 1: Download the Dataset

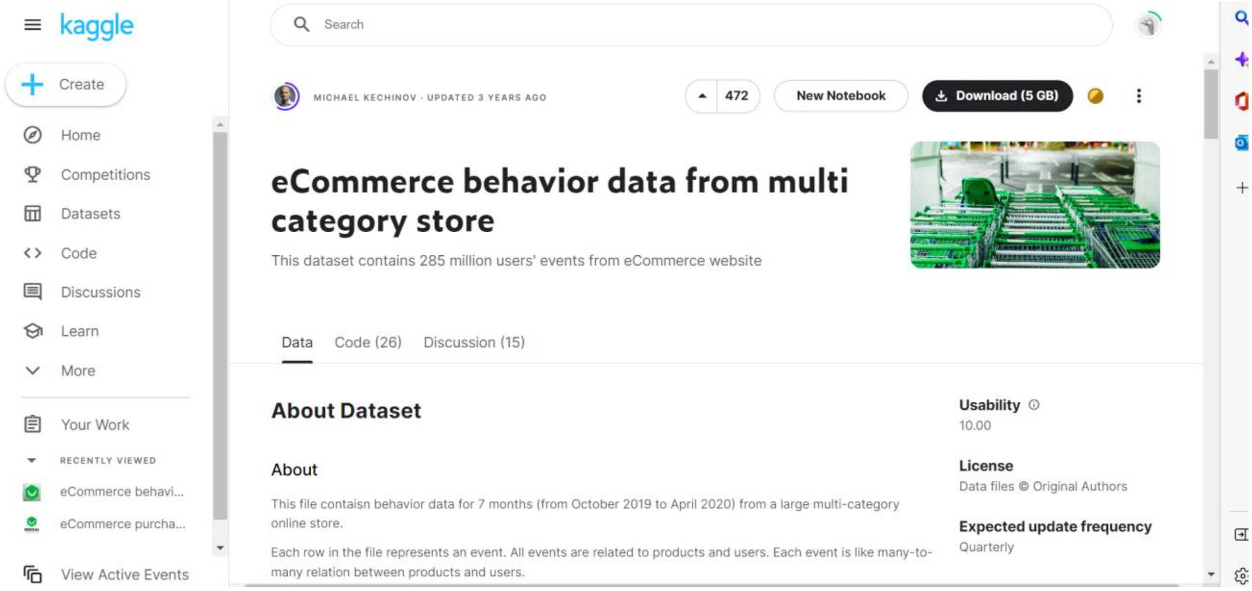
This step is to get data manually. You need to remotely access your Oracle Cloud Big Data Compute Editions that you executed in your Oracle Cloud account using ssh using the information - ip address and connect command in beeline CLI

Ecommerce Behavior Data from Multi Category Store Dataset - Download Dataset to local machine from Kaggle Website, Sign in to Kaggle with any of the following Options.

kaggle



Scroll down until you find the 2 csv files on right side.



eCommerce behavior data from multi category store

This dataset contains 285 million users' events from eCommerce website

About Dataset

About

This file contaisn behavior data for 7 months (from October 2019 to April 2020) from a large multi-category online store.

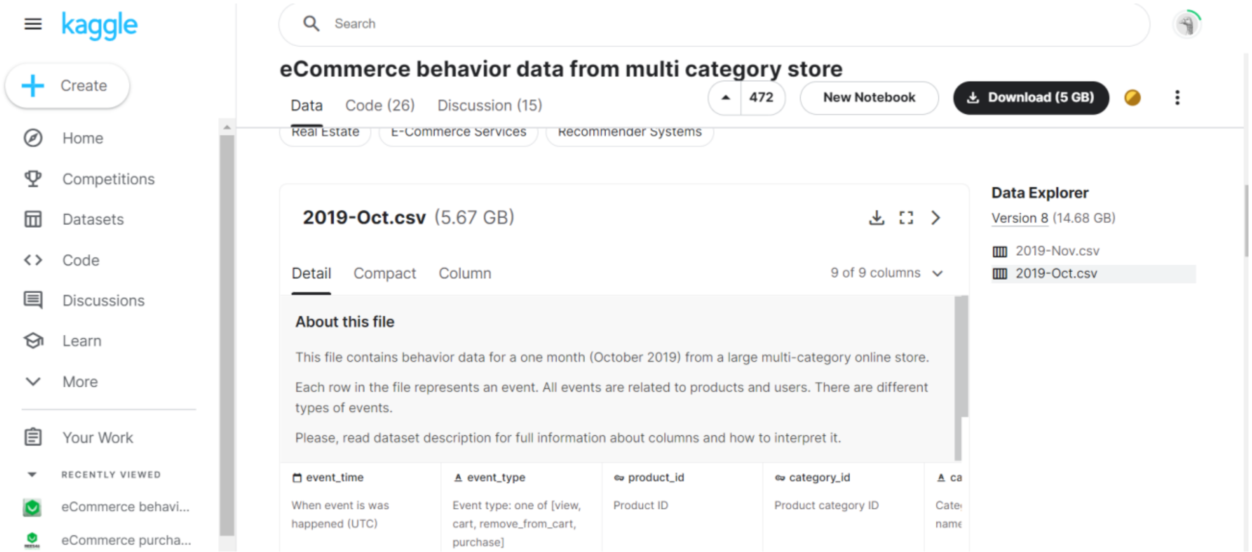
Each row in the file represents an event. All events are related to products and users. Each event is like many-to-many relation between products and users.

Usability
10.00

License
Data files © Original Authors

Expected update frequency
Quarterly

Download 2019-Nov.csv and 2019-Oct.csv, You will see Two zip files in downloads of your Personal Computer



eCommerce behavior data from multi category store

2019-Oct.csv (5.67 GB)

About this file

This file contains behavior data for a one month (October 2019) from a large multi-category online store.

Each row in the file represents an event. All events are related to products and users. There are different types of events.

Please, read dataset description for full information about columns and how to interpret it.

| event_time | event_type | product_id | category_id | ca |
|---|---|---|---|---|
| When event is was happened (UTC) | Event type: one of [view, cart, remove_from_cart, purchase] | Product ID | Product category ID | Cate<br>name |

Extract the Zip files then you can find 2 csv files of October & November which should be uploaded in HDFS.

## Step 2: Upload Files to Hadoop File System (HDFS)

### Using SCP:

Open a command prompt session and from the directory of the extracted files in the previous step and perform the following commands:

```
scp /Users/lekhaajit/November.csv lajitku@144.24.14.145:/tmp
scp /Users/lekhaajit//October.csv lajitku@144.24.14.145:/tmp
```

**Note:** Use your own userid and server ip address.

Connect to server provided by the instructor.

You need to remotely access your server provided by the instructor using ssh. Your CalStateLA username(lajitku) should be a username/password to connect to the Hadoop cluster as follows:

**Note:** Do not forget to change lajitku with your username.

```
ssh lajitku@144.24.14.145
```

Create Directories and transfer the October and November files from tmp to ecommerce1 and ecommerce2 respectively.

Hdfs dfs -mkdir ecommerce1

Hdfs dfs -mkdir ecommerce2

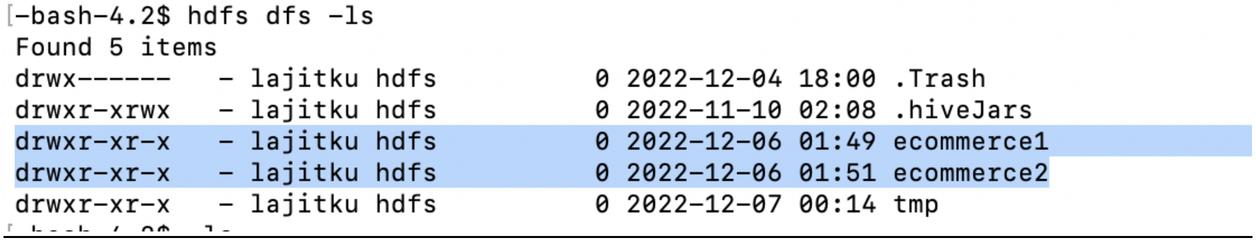
Cd tmp/

hdfs dfs -put 2019-Oct.csv ecommerce_behavior1/

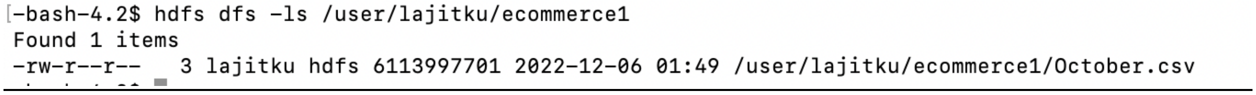
hdfs dfs -put 2019-Nov.csv ecommerce_behavior2/

Confirm files transferred using ls command.

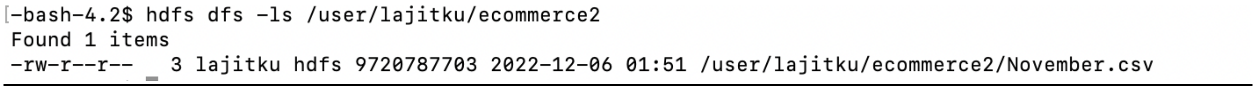
Hdfs dfs -ls



```
[-bash-4.2$ hdfs dfs -ls
 Found 5 items
 drwx------   - lajitku hdfs          0 2022-12-04 18:00 .Trash
 drwxr-xrwx   - lajitku hdfs          0 2022-11-10 02:08 .hiveJars
 drwxr-xr-x   - lajitku hdfs          0 2022-12-06 01:49 ecommerce1
 drwxr-xr-x   - lajitku hdfs          0 2022-12-06 01:51 ecommerce2
 drwxr-xr-x   - lajitku hdfs          0 2022-12-07 00:14 tmp
```



```
[-bash-4.2$ hdfs dfs -ls /user/lajitku/ecommerce1
 Found 1 items
 -rw-r--r--   3 lajitku hdfs 6113997701 2022-12-06 01:49 /user/lajitku/ecommerce1/October.csv
```



```
[-bash-4.2$ hdfs dfs -ls /user/lajitku/ecommerce2
 Found 1 items
 -rw-r--r--   3 lajitku hdfs 9720787703 2022-12-06 01:51 /user/lajitku/ecommerce2/November.csv
```

## Step 3: Create Hive Tables

The following Hive statement creates an external table that allows Hive to query data stored in HDFS.

External tables preserve the data in the original file format while allowing the Hive to perform queries against the data within the file.

The Hive statements below creates a new table, by describing the fields and the delimiter (Comma) between fields from the file.

Now you have to open another terminal window and login into your account using ssh command.

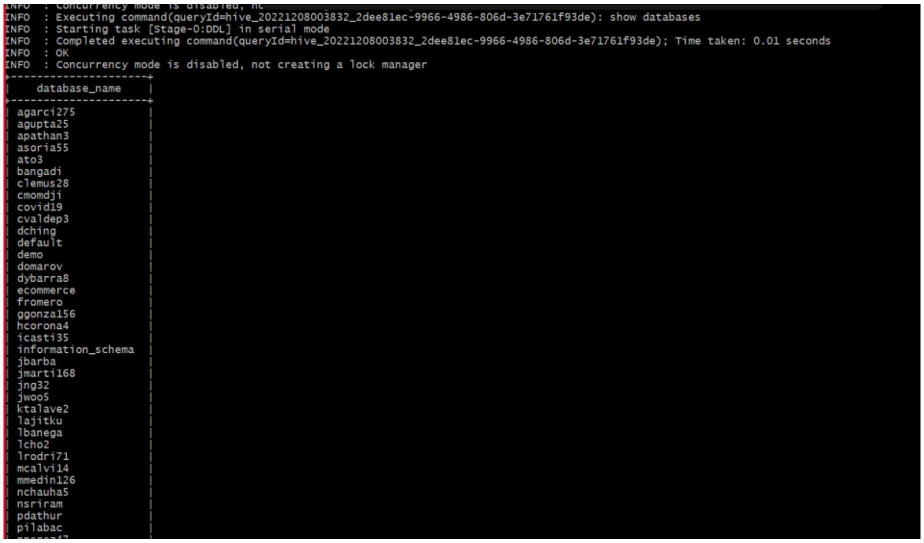
Open beeline Command Line Interface using the following command to run hive queries. Beeline is for multiple users access to Hive Server 2 of a Hadoop cluster.

```
-bash-4.2$ beeline
```

Now you must create your database with your username to separate your tables from other users. For example, the user (lajitku) should run the following:

```
0: jdbc:hive2://bigdaiwn0.sub02180640120.trai> CREATE DATABASE IF NOT EXISTS lajitku;
```

```
0: jdbc:hive2://bigdaiwn0.sub02180640120.trai> show databases;
```



```
INFO  : Concurrency mode is disabled, nc
INFO  : Executing command(queryId=hive_20221208003832_2dee81ec-9966-4986-806d-3e71761f93de): show databases
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20221208003832_2dee81ec-9966-4986-806d-3e71761f93de); Time taken: 0.01 seconds
INFO  : OK
INFO  : Concurrency mode is disabled, not creating a lock manager
+---------------------+
|    database_name    |
+---------------------+
| agarci275           |
| agupta25            |
| apathan3            |
| asoria55            |
| ato3                |
| bangadi             |
| clemus28            |
| cmomdji             |
| covid19             |
| cvaldep3            |
| dching              |
| default             |
| demo                |
| domarov             |
| dybarra8            |
| ecommerce           |
| fromero             |
| ggonza156           |
| hcorona4            |
| icasti35            |
| information_schema  |
| jbarba              |
| jmarti168           |
| jng32               |
| jwoo5               |
| ktalave2            |
| lajitku             |
| lbanega             |
| lcho2               |
| lrodri71            |
| mcalvi14            |
| mmedin126           |
| nchauha5            |
| nsriram             |
| pdathur             |
| pilabac             |
```

```
0: jdbc:hive2://bigdaiwn0.sub02180640120.trai> use lajitku;
```

Note: use your database name instead of lajitku

**October month:**

```
CREATE EXTERNAL TABLE IF NOT EXISTS Octuncleaned (
sno INT,
event_time STRING,
event_type STRING,
product_id INT,
category_id BIGINT,
category_code STRING,
brand STRING,
price DOUBLE,
user_id INT,
user_session STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE LOCATION '/user/lajitku/ecommerce1/'
TBLPROPERTIES ('skip.header.line.count'='1');
```

**November month:**

```
CREATE EXTERNAL TABLE IF NOT EXISTS Novuncleaned (
sno INT,
event_time STRING,
event_type STRING,
product_id INT,
category_id BIGINT,
category_code STRING,
brand STRING,
price DOUBLE,
user_id INT,
user_session STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE LOCATION '/user/lajitku/ecommerce2/'
TBLPROPERTIES ('skip.header.line.count'='1');
```

## Data Cleaning and Creation of New Tables:

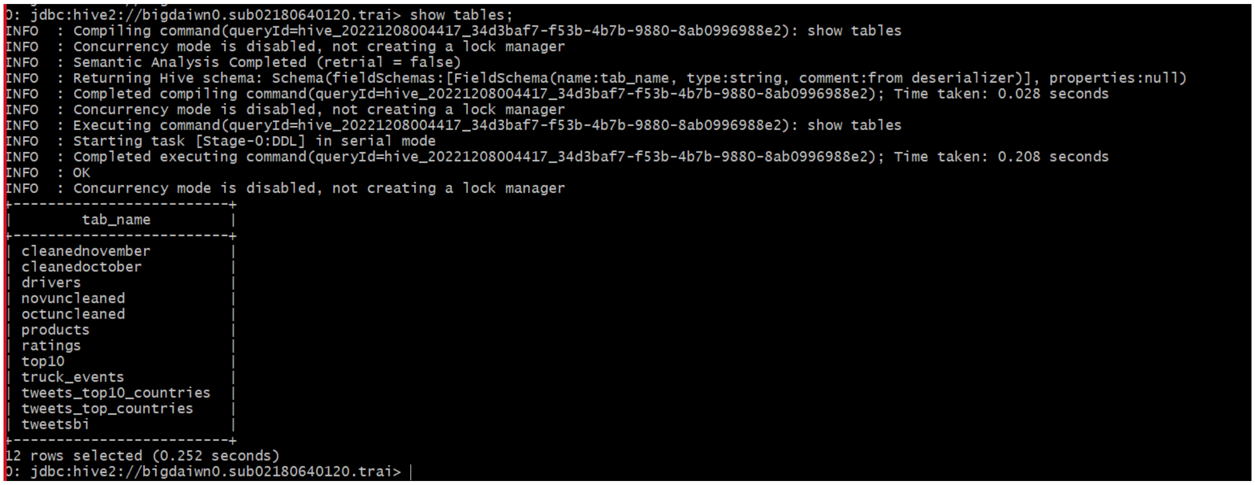
### October month:

```
CREATE TABLE IF NOT EXISTS cleanedoctober
AS SELECT * from octuncleaned
where category_code not like "NULL" AND brand not like "NULL" AND user_session not like "NULL";
```

### November month:

```
CREATE TABLE IF NOT EXISTS cleanednovember
AS SELECT * from novuncleaned
where category_code not like "NULL" AND brand not like "NULL" AND user_session not like "NULL";
```

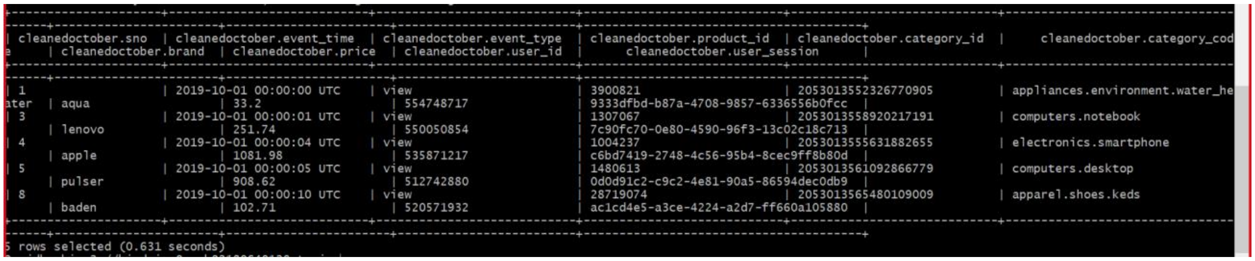
Confirm the Tables creation using Show Tables;



```
0: jdbc:hive2://bigdaiwn0.sub02180640120.trai> show tables;
INFO  : Compiling command(queryId=hive_20221208004417_34d3baf7-f53b-4b7b-9880-8ab0996988e2): show tables
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Semantic Analysis Completed (retrial = false)
INFO  : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:tab_name, type:string, comment:from deserializer)], properties:null)
INFO  : Completed compiling command(queryId=hive_20221208004417_34d3baf7-f53b-4b7b-9880-8ab0996988e2); Time taken: 0.028 seconds
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Executing command(queryId=hive_20221208004417_34d3baf7-f53b-4b7b-9880-8ab0996988e2): show tables
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20221208004417_34d3baf7-f53b-4b7b-9880-8ab0996988e2); Time taken: 0.208 seconds
INFO  : OK
INFO  : Concurrency mode is disabled, not creating a lock manager
+------------------------+
|        tab_name        |
+------------------------+
| cleanednovember        |
| cleanedoctober         |
| drivers                |
| novuncleaned           |
| octuncleaned           |
| products               |
| ratings                |
| top10                  |
| truck_events           |
| tweets_top10_countries |
| tweets_top_countries   |
| tweetsbi               |
+------------------------+
12 rows selected (0.252 seconds)
0: jdbc:hive2://bigdaiwn0.sub02180640120.trai> 
```

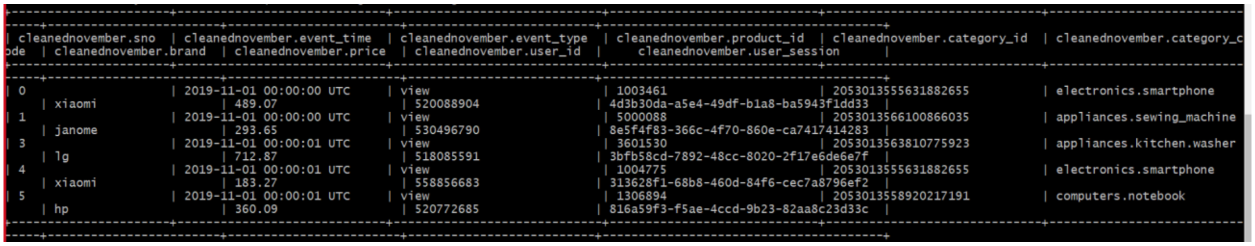
Confirm contents in table with the SELECT statement.

0: jdbc:hive2://bigdaiwn0.sub02180640120.trai> SELECT * from cleanedoctober limit 5;



```
+-------------------+-------------------------+--------------------------+------------------------------------+-----------------------------+-------------------------------
-----+----------------------+-----------------------+-------------------------+--------------------------------------+
| cleanedoctober.sno  | cleanedoctober.event_time  | cleanedoctober.event_type  | cleanedoctober.product_id  | cleanedoctober.category_id  |     cleanedoctober.category_cod
e     | cleanedoctober.brand  | cleanedoctober.price  | cleanedoctober.user_id  |      cleanedoctober.user_session      |
+-------------------+-------------------------+--------------------------+------------------------------------+-----------------------------+-------------------------------
-----+----------------------+-----------------------+-------------------------+--------------------------------------+
| 1                   | 2019-10-01 00:00:00 UTC    | view                       | 3900821                    | 2053013552326770905         | appliances.environment.water_he
ater  | aqua                  | 33.2                  | 554748717               | 9333dfbd-b87a-4708-9857-6336556b0fcc  |
| 3                   | 2019-10-01 00:00:01 UTC    | view                       | 1307067                    | 2053013558920217191         | computers.notebook
      | lenovo                | 251.74                | 550050854               | 7c90fc70-0e80-4590-96f3-13c02c18c713  |
| 4                   | 2019-10-01 00:00:04 UTC    | view                       | 1004237                    | 2053013555631882655         | electronics.smartphone
      | apple                 | 1081.98               | 535871217               | c6bd7419-2748-4c56-95b4-8cec9ff8b80d  |
| 5                   | 2019-10-01 00:00:05 UTC    | view                       | 1480613                    | 2053013561092866779         | computers.desktop
      | pulser                | 908.62                | 512742880               | 0d0d91c2-c9c2-4e81-90a5-86594dec0db9  |
| 8                   | 2019-10-01 00:00:10 UTC    | view                       | 28719074                   | 2053013565480109009         | apparel.shoes.keds
      | baden                 | 102.71                | 520571932               | ac1cd4e5-a3ce-4224-a2d7-ff660a105880  |
+-------------------+-------------------------+--------------------------+------------------------------------+-----------------------------+-------------------------------
-----+----------------------+-----------------------+-------------------------+--------------------------------------+
5 rows selected (0.631 seconds)
```

0: jdbc:hive2://bigdaiwn0.sub02180640120.trai> SELECT * from cleanednovember limit 5;



```
+-----------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+--------------------------
-----+-----------------------+-----------------------------+-----------------------------------------+
| cleanednovember.sno  | cleanednovember.event_time  | cleanednovember.event_type  | cleanednovember.product_id  | cleanednovember.category_id  | cleanednovember.category_c
ode  | cleanednovember.brand  | cleanednovember.price  | cleanednovember.user_id  |     cleanednovember.user_session      |
+-----------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+--------------------------
-----+-----------------------+-----------------------------+-----------------------------------------+
| 0                    | 2019-11-01 00:00:00 UTC     | view                        | 1003461                     | 2053013555631882655          | electronics.smartphone
     | xiaomi               | 489.07                | 520088904                | 4d3b30da-a5e4-49df-b1a8-ba5943f1dd33  |
| 1                    | 2019-11-01 00:00:00 UTC     | view                        | 5000088                     | 2053013566100866035          | appliances.sewing_machine
     | janome               | 293.65                | 530496790                | 8e5f4f83-366c-4f70-860e-ca7417414283  |
| 3                    | 2019-11-01 00:00:01 UTC     | view                        | 3601530                     | 2053013563810775923          | appliances.kitchen.washer
     | lg                   | 712.87                | 518085591                | 3bfb58cd-7892-48cc-8020-2f17e6de6e7f  |
| 4                    | 2019-11-01 00:00:01 UTC     | view                        | 1004775                     | 2053013555631882655          | electronics.smartphone
     | xiaomi               | 183.27                | 558856683                | 313628f1-68b8-460d-84f6-cec7a8796ef2  |
| 5                    | 2019-11-01 00:00:01 UTC     | view                        | 1306894                     | 2053013558920217191          | computers.notebook
     | hp                   | 360.09                | 520772685                | 816a59f3-f5ae-4ccd-9b23-82aa8c23d33c  |
+-----------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+--------------------------
-----+-----------------------+-----------------------------+-----------------------------------------+
```

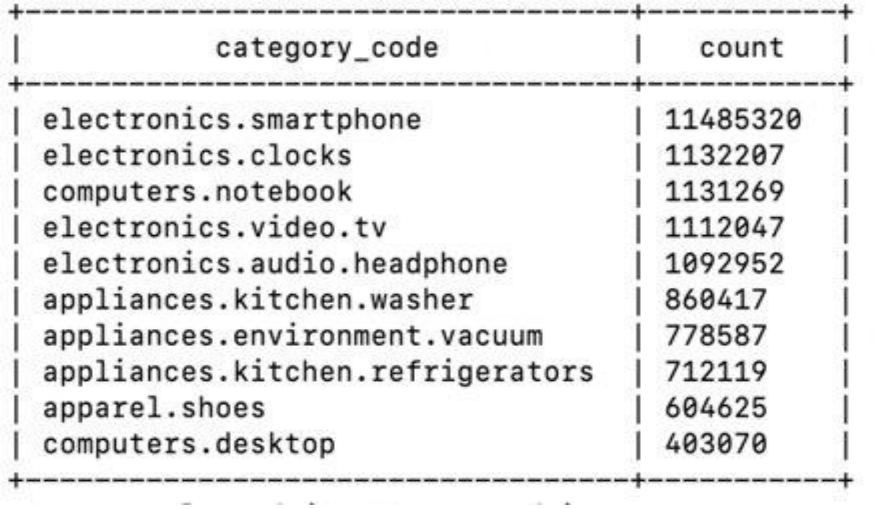
## Step 4: Create Hive Table Queries

The following Queries will help us to figure out the Visualization and analyze the Customer Behavior

### 1. Top 10 popular categories in October and November

**October:**

select category_code, count(category_code) as count from cleanedoctober group by category_code order by count(category_code) desc limit 10;



```
+-------------------------------------+------------+
|            category_code            |   count    |
+-------------------------------------+------------+
| electronics.smartphone              | 11485320   |
| electronics.clocks                  | 1132207    |
| computers.notebook                  | 1131269    |
| electronics.video.tv                | 1112047    |
| electronics.audio.headphone         | 1092952    |
| appliances.kitchen.washer           | 860417     |
| appliances.environment.vacuum       | 778587     |
| appliances.kitchen.refrigerators    | 712119     |
| apparel.shoes                       | 604625     |
| computers.desktop                   | 403070     |
+-------------------------------------+------------+
```

At Hive (beeline) terminal, create a csv file as an output using the script as follows:

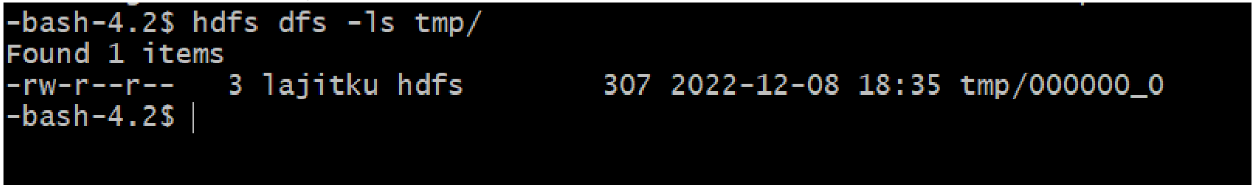
INSERT OVERWRITE DIRECTORY '/user/lajitku/tmp/'

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

select category_code, count(category_code) as count from cleanedoctober group by category_code order by count(category_code) desc limit 10;

- Go to the shell terminal to run the following command, which shows the file 000000_0:

-bash-4.2$ hdfs dfs -ls tmp/



```
-bash-4.2$ hdfs dfs -ls tmp/
Found 1 items
-rw-r--r--   3 lajitku hdfs        307 2022-12-08 18:35 tmp/000000_0
-bash-4.2$
```

Download the output file “000000_0” to “October1.csv” using the following hdfs command:

bash-4.2$ hdfs dfs -get tmp/000000_0 October1.csv

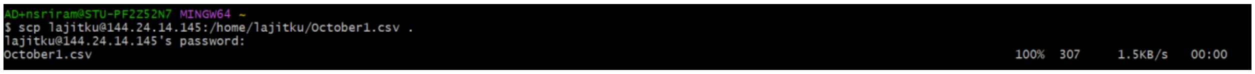
At your PC with git bash, xterminal, or pscp.exe, you can remotely download the output

file “October1.csv” to your PC to visualize it using Excel .

NOTE: the following code has “.” at the end; You actually can connect from Tableau to the Hadoop

cluster to get this file but Hadoop Cloud does not have the connector.

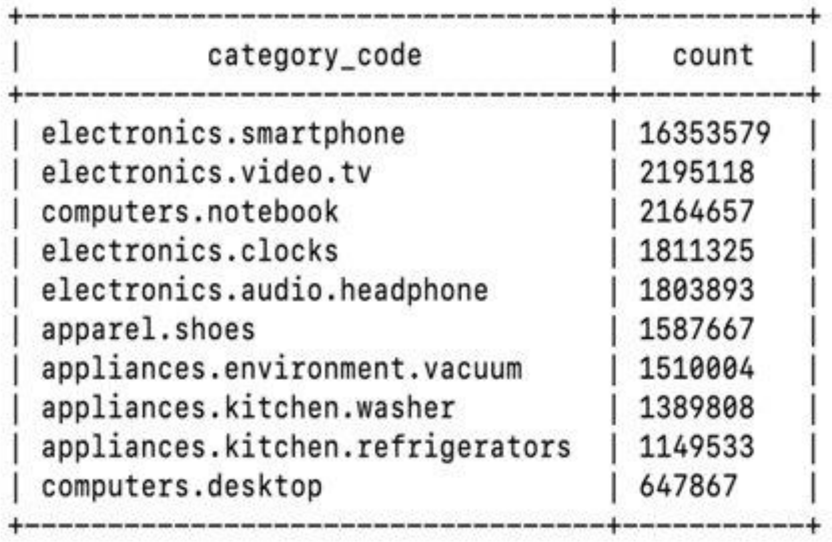
scp lajitku@144.24.14.145:/home/lajitku/October1.csv .



```
AD+nsriram@STU-PF2Z52N7 MINGW64 ~
$ scp lajitku@144.24.14.145:/home/lajitku/October1.csv .
lajitku@144.24.14.145's password:
October1.csv                                                              100%  307     1.5KB/s   00:00
```

**November:**

select category_code, count(category_code) as count from cleanednovember group
by category_code order by count(category_code) desc limit 10;



```
+-----------------------------------+-----------+
|           category_code           |   count   |
+-----------------------------------+-----------+
| electronics.smartphone            | 16353579  |
| electronics.video.tv              | 2195118   |
| computers.notebook                | 2164657   |
| electronics.clocks                | 1811325   |
| electronics.audio.headphone       | 1803893   |
| apparel.shoes                     | 1587667   |
| appliances.environment.vacuum     | 1510004   |
| appliances.kitchen.washer         | 1389808   |
| appliances.kitchen.refrigerators  | 1149533   |
| computers.desktop                 | 647867    |
+-----------------------------------+-----------+
```

At Hive (beeline) terminal, create a csv file as an output using the script as follows:

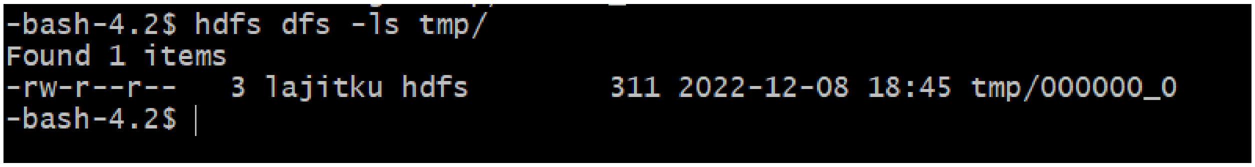
INSERT OVERWRITE DIRECTORY '/user/lajitku/tmp/'

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

select category_code, count(category_code) as count from cleanednovember group by category_code order by count(category_code) desc limit 10;

Go to the shell terminal to run the following command, which shows the file 000000_0:

-bash-4.2$ hdfs dfs -ls tmp/



```
-bash-4.2$ hdfs dfs -ls tmp/
Found 1 items
-rw-r--r--   3 lajitku hdfs        311 2022-12-08 18:45 tmp/000000_0
-bash-4.2$ |
```

Download the output file “000000_0” to “November1.csv” using the following hdfs command:

bash-4.2$ hdfs dfs -get tmp/000000_0 November1.csv

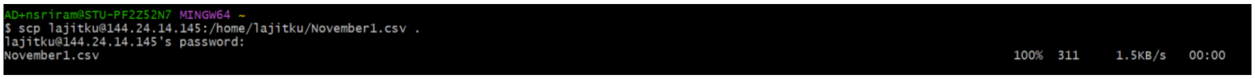
At your PC with git bash, xterminal, or pscp.exe, you can remotely download the output

file “November1.csv” to your PC to visualize it using Excel .

NOTE: the following code has “.” at the end; You actually can connect from Tableau to the Hadoop

cluster to get this file but Hadoop Cloud does not have the connector.

scp lajitku@144.24.14.145:/home/lajitku/November1.csv .

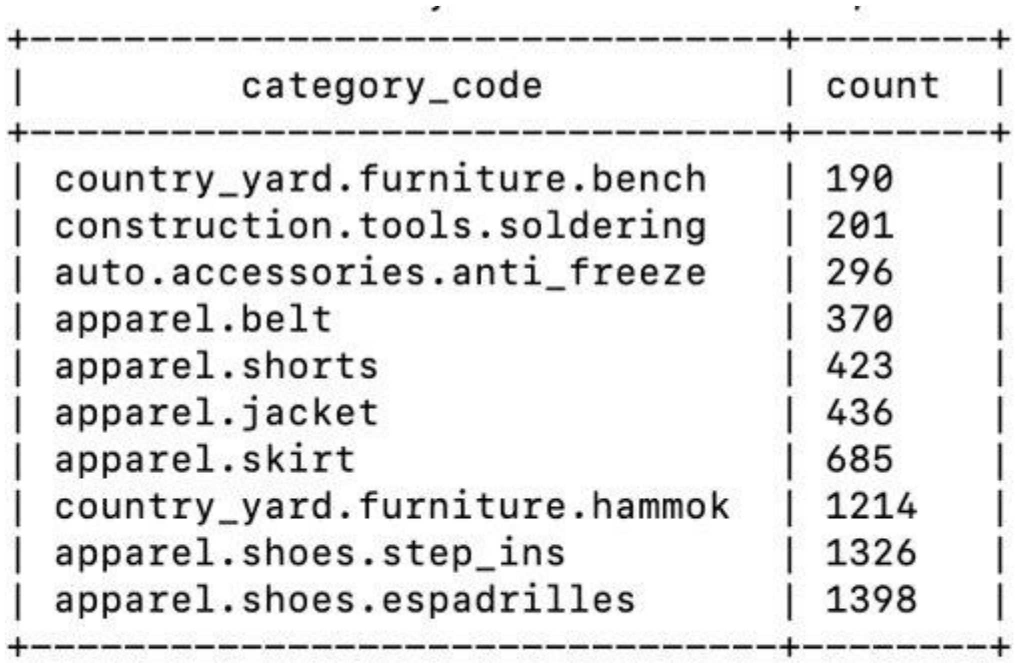


```
AD+nsriram@STU-PF2Z52N7 MINGW64 ~
$ scp lajitku@144.24.14.145:/home/lajitku/November1.csv .
lajitku@144.24.14.145's password:
November1.csv                                                          100%  311     1.5KB/s   00:00
```

## 2. Top 10 Least popular categories in October and November

### October

select category_code, count(category_code) as count from cleanedoctober group by category_code order by count(category_code) limit 10;



```
+-----------------------------------+--------+
|          category_code            | count  |
+-----------------------------------+--------+
| country_yard.furniture.bench      | 190    |
| construction.tools.soldering      | 201    |
| auto.accessories.anti_freeze      | 296    |
| apparel.belt                      | 370    |
| apparel.shorts                    | 423    |
| apparel.jacket                    | 436    |
| apparel.skirt                     | 685    |
| country_yard.furniture.hammok     | 1214   |
| apparel.shoes.step_ins            | 1326   |
| apparel.shoes.espadrilles         | 1398   |
+-----------------------------------+--------+
```

At Hive (beeline) terminal, create a csv file as an output using the script as follows:

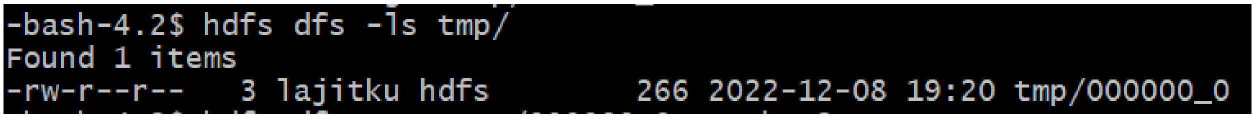
INSERT OVERWRITE DIRECTORY '/user/lajitku/tmp/'

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

select category_code, count(category_code) as count from cleanedoctober group by category_code order by count(category_code) limit 10;

Go to the shell terminal to run the following command, which shows the file 000000_0:

-bash-4.2$ hdfs dfs -ls tmp/



```
-bash-4.2$ hdfs dfs -ls tmp/
Found 1 items
-rw-r--r--   3 lajitku hdfs        266 2022-12-08 19:20 tmp/000000_0
```

Download the output file “000000_0” to “October2.csv” using the following hdfs command:

bash-4.2$ hdfs dfs -get tmp/000000_0 October2.csv

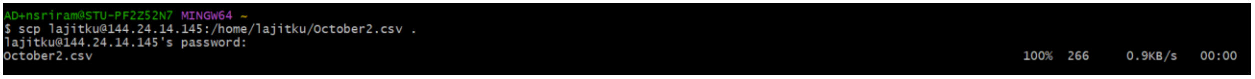
At your PC with git bash, xterminal, or pscp.exe, you can remotely download the output

file “October2.csv” to your PC to visualize it using Excel .

NOTE: the following code has “.” at the end; You actually can connect from Tableau to the Hadoop

cluster to get this file but Hadoop Cloud does not have the connector.

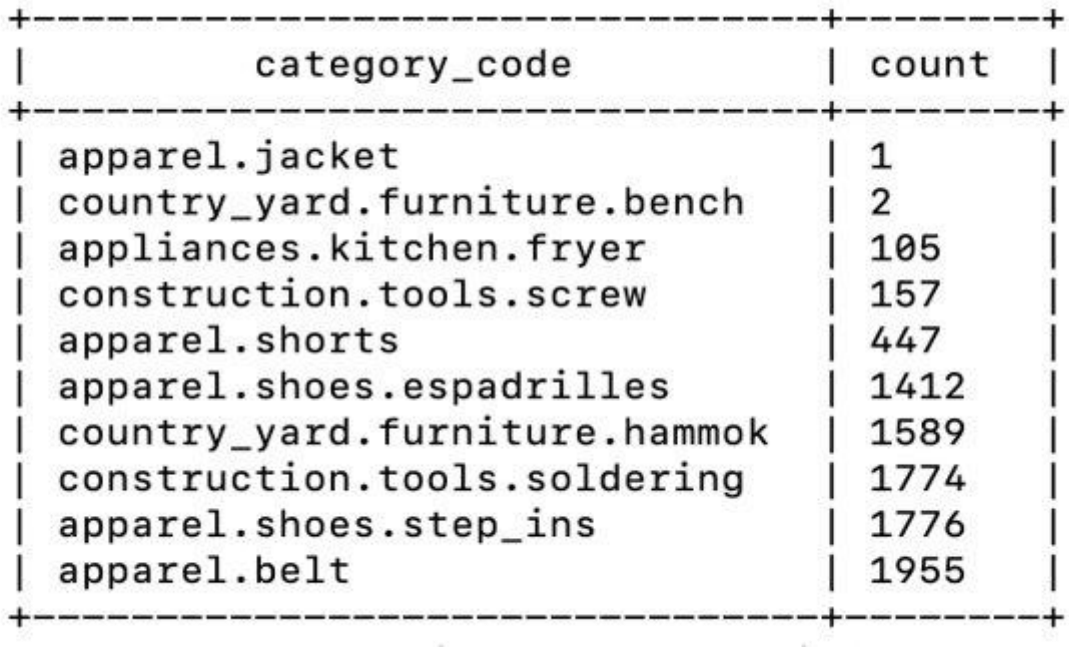
scp lajitku@144.24.14.145:/home/lajitku/October2.csv .



```
AD+nsriram@STU-PF2Z52N7 MINGW64 ~
$ scp lajitku@144.24.14.145:/home/lajitku/October2.csv .
lajitku@144.24.14.145's password:
October2.csv                                                                  100%  266     0.9KB/s   00:00
```

**November**

select category_code, count(category_code) as count from cleanednovember group by category_code order by count(category_code) limit 10;



```
+-----------------------------------+--------+
|           category_code           | count  |
+-----------------------------------+--------+
| apparel.jacket                    | 1      |
| country_yard.furniture.bench      | 2      |
| appliances.kitchen.fryer          | 105    |
| construction.tools.screw          | 157    |
| apparel.shorts                    | 447    |
| apparel.shoes.espadrilles         | 1412   |
| country_yard.furniture.hammok     | 1589   |
| construction.tools.soldering      | 1774   |
| apparel.shoes.step_ins            | 1776   |
| apparel.belt                      | 1955   |
+-----------------------------------+--------+
```

At Hive (beeline) terminal, create a csv file as an output using the script as follows:

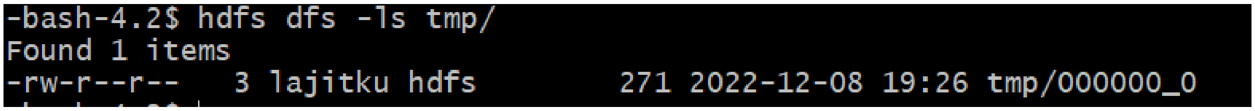
INSERT OVERWRITE DIRECTORY '/user/lajitku/tmp/'

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

select category_code, count(category_code) as count from cleanednovember group by category_code order by count(category_code) limit 10;

Go to the shell terminal to run the following command, which shows the file 000000_0:

-bash-4.2$ hdfs dfs -ls tmp/



```
-bash-4.2$ hdfs dfs -ls tmp/
Found 1 items
-rw-r--r--   3 lajitku hdfs        271 2022-12-08 19:26 tmp/000000_0
```

Download the output file “000000_0” to “November2.csv” using the following hdfs command:

bash-4.2$ hdfs dfs -get tmp/000000_0 November2.csv

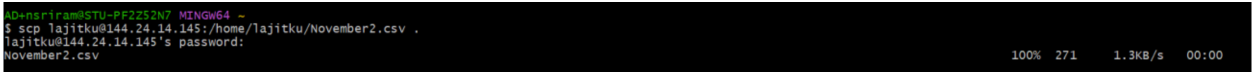
At your PC with git bash, xterminal, or pscp.exe, you can remotely download the output

file “November2.csv” to your PC to visualize it using Excel .

NOTE: the following code has “.” at the end; You actually can connect from Tableau to the Hadoop

cluster to get this file but Hadoop Cloud does not have the connector.

scp lajitku@144.24.14.145:/home/lajitku/November2.csv .

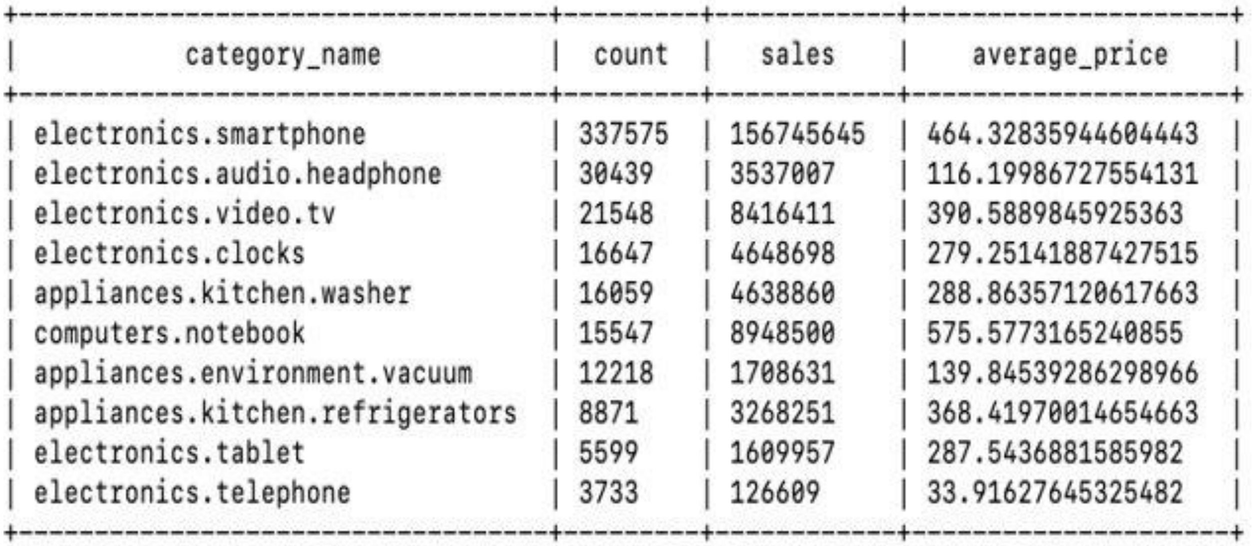


```
AD+nsriram@STU-PF2Z52N7 MINGW64 ~
$ scp lajitku@144.24.14.145:/home/lajitku/November2.csv .
lajitku@144.24.14.145's password:
November2.csv                                                              100%  271     1.3KB/s   00:00
```

### 3. Top 10 purchased categories and their sales count and average price in October and November.

**October**

select category_code as category_name, count(category_code) as count, cast(sum(price) as bigint) as sales, avg(price) as average_price from cleanedoctober where event_type like 'purchase' group by category_code order by count(category_code) desc limit 10;



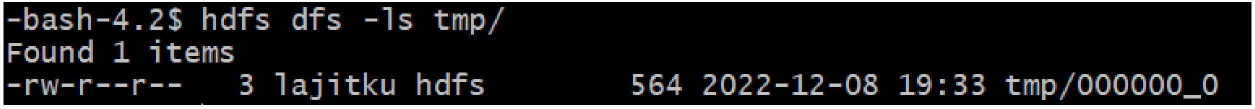
```
+-----------------------------------+---------+-----------+--------------------+
|           category_name           |  count  |   sales   |   average_price    |
+-----------------------------------+---------+-----------+--------------------+
| electronics.smartphone            | 337575  | 156745645 | 464.3283594460443  |
| electronics.audio.headphone       | 30439   | 3537007   | 116.19986727554131 |
| electronics.video.tv              | 21548   | 8416411   | 390.5889845925363  |
| electronics.clocks                | 16647   | 4648698   | 279.25141887427515 |
| appliances.kitchen.washer         | 16059   | 4638860   | 288.86357120617663 |
| computers.notebook                | 15547   | 8948500   | 575.5773165240855  |
| appliances.environment.vacuum     | 12218   | 1708631   | 139.84539286298966 |
| appliances.kitchen.refrigerators  | 8871    | 3268251   | 368.41970014654663 |
| electronics.tablet                | 5599    | 1609957   | 287.5436881585982  |
| electronics.telephone             | 3733    | 126609    | 33.91627645325482  |
+-----------------------------------+---------+-----------+--------------------+
```

At Hive (beeline) terminal, create a csv file as an output using the script as follows:

```
INSERT OVERWRITE DIRECTORY '/user/lajitku/tmp/'
 ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
select category_code as category_name, count(category_code) as count, cast(sum(price) as bigint) as sales, avg(price) as average_price from cleanedoctober where event_type like 'purchase' group by category_code order by count(category_code) desc limit 10;
```

Go to the shell terminal to run the following command, which shows the file 000000_0:

```
-bash-4.2$ hdfs dfs -ls tmp/
```

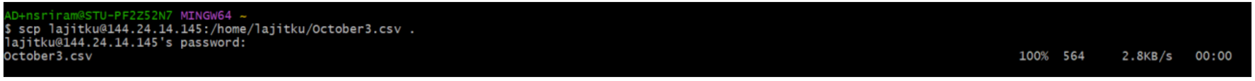


```
-bash-4.2$ hdfs dfs -ls tmp/
Found 1 items
-rw-r--r--   3 lajitku hdfs        564 2022-12-08 19:33 tmp/000000_0
```

Download the output file “000000_0” to “October3.csv” using the following hdfs command:
bash-4.2$ hdfs dfs -get tmp/000000_0 October3.csv

At your PC with git bash, xterminal, or pscp.exe, you can remotely download the output file “October3.csv” to your PC to visualize it using Excel .
NOTE: the following code has “.” at the end; You actually can connect from Tableau to the Hadoop cluster to get this file but Hadoop Cloud does not have the connector.

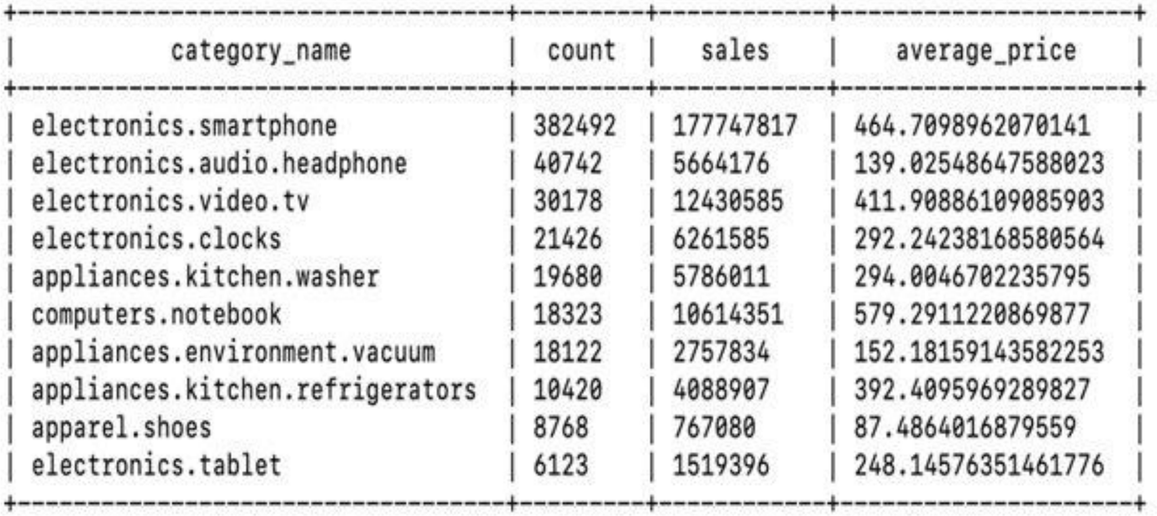
```
scp lajitku@144.24.14.145:/home/lajitku/October3.csv .
```



```
AD+nsriram@STU-PF2Z52N7 MINGW64 ~
$ scp lajitku@144.24.14.145:/home/lajitku/October3.csv .
lajitku@144.24.14.145's password:
October3.csv                                                                100%  564     2.8KB/s   00:00
```

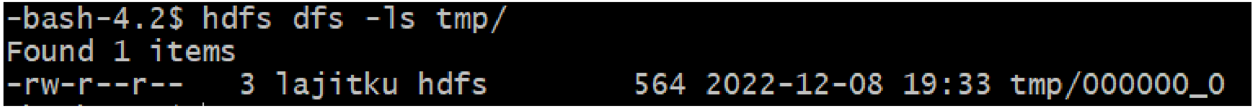
**November**

```
select category_code as category_name, count(category_code) as count, cast(sum(price) as bigint) as sales, avg(price) as average_price from cleanednovember where event_type like 'purchase' group by category_code order by count(category_code) desc limit 10;
```



```
+-----------------------------------+---------+------------+---------------------+
|           category_name           |  count  |   sales    |    average_price    |
+-----------------------------------+---------+------------+---------------------+
| electronics.smartphone            | 382492  | 177747817  | 464.7098962070141   |
| electronics.audio.headphone       | 40742   | 5664176    | 139.02548647588023  |
| electronics.video.tv              | 30178   | 12430585   | 411.90886109085903  |
| electronics.clocks                | 21426   | 6261585    | 292.24238168580564  |
| appliances.kitchen.washer         | 19680   | 5786011    | 294.0046702235795   |
| computers.notebook                | 18323   | 10614351   | 579.2911220869877   |
| appliances.environment.vacuum     | 18122   | 2757834    | 152.18159143582253  |
| appliances.kitchen.refrigerators  | 10420   | 4088907    | 392.4095969289827   |
| apparel.shoes                     | 8768    | 767080     | 87.4864016879559    |
| electronics.tablet                | 6123    | 1519396    | 248.14576351461776  |
+-----------------------------------+---------+------------+---------------------+
```

At Hive (beeline) terminal, create a csv file as an output using the script as follows:
INSERT OVERWRITE DIRECTORY '/user/lajitku/tmp/'
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
select category_code as category_name, count(category_code) as count, cast(sum(price) as bigint) as sales, avg(price) as average_price from cleanednovember where event_type like 'purchase' group by category_code order by count(category_code) desc limit 10;
Go to the shell terminal to run the following command, which shows the file 000000_0:
-bash-4.2$ hdfs dfs -ls tmp/

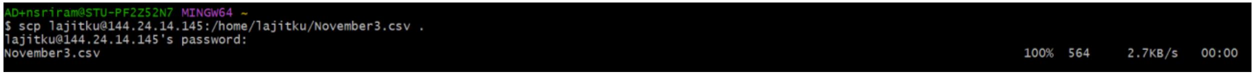


```
-bash-4.2$ hdfs dfs -ls tmp/
Found 1 items
-rw-r--r--   3 lajitku hdfs        564 2022-12-08 19:33 tmp/000000_0
```

Download the output file "000000_0" to "November3.csv" using the following hdfs command:
bash-4.2$ hdfs dfs -get tmp/000000_0 November3.csv

At your PC with git bash, xterminal, or pscp.exe, you can remotely download the output file "November3.csv" to your PC to visualize it using Excel .
NOTE: the following code has "." at the end; You actually can connect from Tableau to the Hadoop cluster to get this file but Hadoop Cloud does not have the connector.

scp lajitku@144.24.14.145:/home/lajitku/November3.csv .

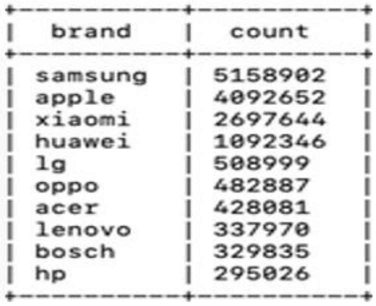


```
AD+nsriram@STU-PF2Z52N7 MINGW64 ~
$ scp lajitku@144.24.14.145:/home/lajitku/November3.csv .
lajitku@144.24.14.145's password:
November3.csv                                                  100%  564     2.7KB/s   00:00
```

## 4. Top 10 popular brands October and November

### October:

select brand, count(brand) as count from cleanedoctober group by brand order by count(brand) desc limit 10;



```
+-----------+-----------+
|  brand    |  count    |
+-----------+-----------+
| samsung   | 5158902   |
| apple     | 4092652   |
| xiaomi    | 2697644   |
| huawei    | 1092346   |
| lg        | 508999    |
| oppo      | 482887    |
| acer      | 428081    |
| lenovo    | 337970    |
| bosch     | 329835    |
| hp        | 295026    |
+-----------+-----------+
```

At Hive (beeline) terminal, create a csv file as an output using the script as follows:

INSERT OVERWRITE DIRECTORY '/user/lajitku/tmp/'

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

select brand, count(brand) as count from cleanedoctober group by brand order
by count(brand) desc limit 10;

Go to the shell terminal to run the following command, which shows the file 000000_0:

-bash-4.2$ hdfs dfs -ls tmp/

Download the output file “000000_0” to “October4.csv” using the following hdfs command:

bash-4.2$ hdfs dfs -get tmp/000000_0 October4.csv

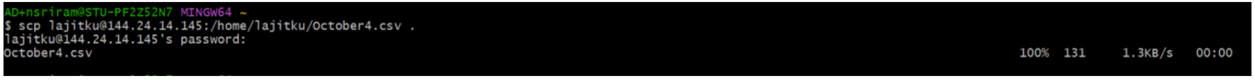
At your PC with git bash, xterminal, or pscp.exe, you can remotely download the output

file “October4.csv” to your PC to visualize it using Excel .

NOTE: the following code has “.” at the end; You actually can connect from Tableau to the Hadoop

cluster to get this file but Hadoop Cloud does not have the connector.

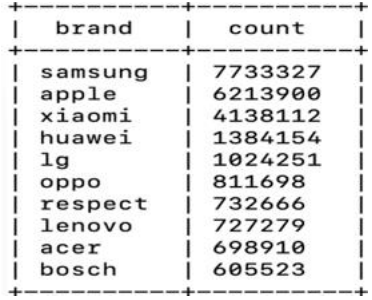
scp lajitku@144.24.14.145:/home/lajitku/October4.csv .



```
AD+nsriram@STU-PF2Z52N7 MINGW64 ~
$ scp lajitku@144.24.14.145:/home/lajitku/October4.csv .
lajitku@144.24.14.145's password:
October4.csv                                                          100%  131     1.3KB/s   00:00
```

## November:

select brand, count(brand) as count from cleanednovember group by brand order
by count(brand) desc limit 10;



```
+----------+----------+
|  brand   |  count   |
+----------+----------+
| samsung  | 7733327  |
| apple    | 6213900  |
| xiaomi   | 4138112  |
| huawei   | 1384154  |
| lg       | 1024251  |
| oppo     | 811698   |
| respect  | 732666   |
| lenovo   | 727279   |
| acer     | 698910   |
| bosch    | 605523   |
+----------+----------+
```

At Hive (beeline) terminal, create a csv file as an output using the script as follows:

INSERT OVERWRITE DIRECTORY '/user/lajitku/tmp/'

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

select brand, count(brand) as count from cleanednovember group by brand order by count(brand) desc limit 10;

Go to the shell terminal to run the following command, which shows the file 000000_0:

-bash-4.2$ hdfs dfs -ls tmp/

Download the output file “000000_0” to “November4.csv” using the following hdfs command:

bash-4.2$ hdfs dfs -get tmp/000000_0 November4.csv

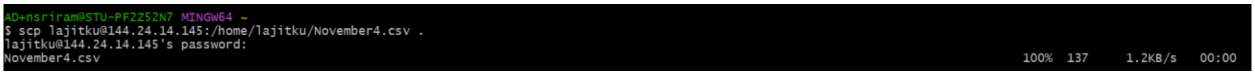
At your PC with git bash, xterminal, or pscp.exe, you can remotely download the output

file “November4.csv” to your PC to visualize it using Excel .

**NOTE:** the following code has “.” at the end; You actually can connect from Tableau to the Hadoop

cluster to get this file but Hadoop Cloud does not have the connector.

scp lajitku@144.24.14.145:/home/lajitku/November4.csv .

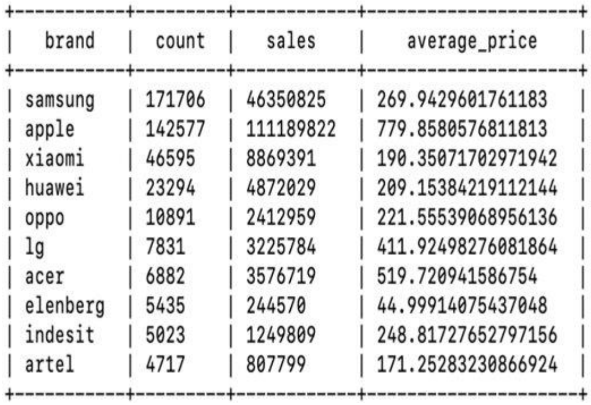


```
AD+nsriram@STU-PF2Z52N7 MINGW64 ~
$ scp lajitku@144.24.14.145:/home/lajitku/November4.csv .
lajitku@144.24.14.145's password:
November4.csv                                                         100%  137     1.2KB/s   00:00
```

## 5.Top 10 Purchased Brands of October and November

### October:

select brand, count(brand) as count, cast(sum(price) as bigint) as sales, avg(price) as average_price from cleanedoctober where event_type like 'purchase' group by brand order by count(brand) desc limit 10;



```
+-----------+---------+------------+---------------------+
|   brand   |  count  |   sales    |    average_price    |
+-----------+---------+------------+---------------------+
| samsung   | 171706  | 46350825   | 269.9429601761183   |
| apple     | 142577  | 111189822  | 779.8580576811813   |
| xiaomi    | 46595   | 8869391    | 190.35071702971942  |
| huawei    | 23294   | 4872029    | 209.15384219112144  |
| oppo      | 10891   | 2412959    | 221.55539068956136  |
| lg        | 7831    | 3225784    | 411.92498276081864  |
| acer      | 6882    | 3576719    | 519.720941586754    |
| elenberg  | 5435    | 244570     | 44.99914075437048   |
| indesit   | 5023    | 1249809    | 248.81727652797156  |
| artel     | 4717    | 807799     | 171.25283230866924  |
+-----------+---------+------------+---------------------+
```

At Hive (beeline) terminal, create a csv file as an output using the script as follows:

```
INSERT OVERWRITE DIRECTORY '/user/lajitku/tmp/'
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
select brand, count(brand) as count, cast(sum(price) as bigint) as sales, avg(price) as average_price from
cleanedoctober where event_type like 'purchase' group by brand order by count(brand) desc limit 10;
```

Go to the shell terminal to run the following command, which shows the file 000000_0:

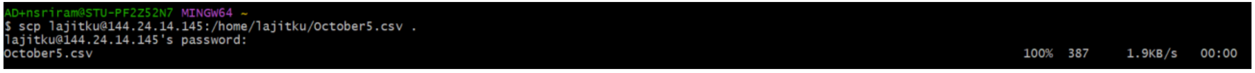
```
-bash-4.2$ hdfs dfs -ls tmp/
```

Download the output file “000000_0” to “October5.csv” using the following hdfs command:

```
bash-4.2$ hdfs dfs -get tmp/000000_0 October5.csv
```

At your PC with git bash, xterminal, or pscp.exe, you can remotely download the output file “October5.csv” to your PC to visualize it using Excel .
NOTE: the following code has “.” at the end; You actually can connect from Tableau to the Hadoop cluster to get this file but Hadoop Cloud does not have the connector.

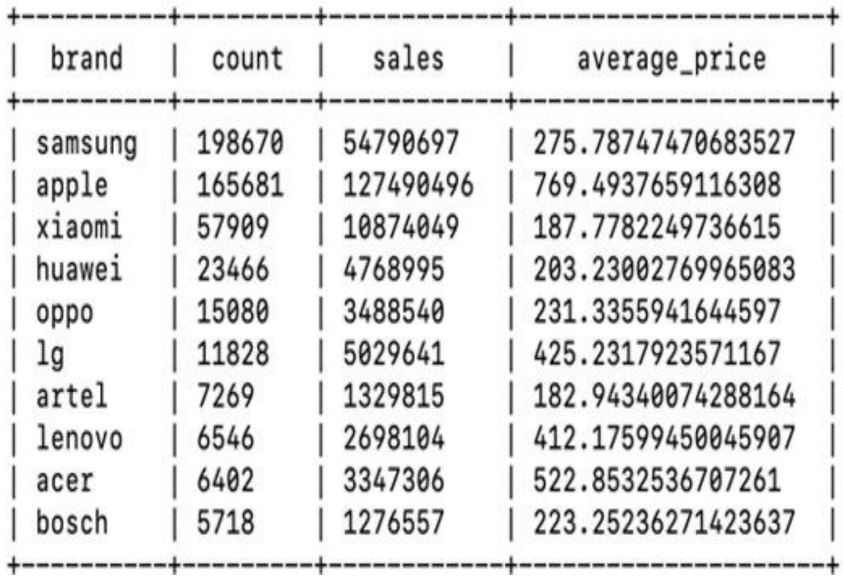
```
scp lajitku@144.24.14.145:/home/lajitku/October5.csv .
```



```
AD+nsriram@STU-PF2Z52N7 MINGW64 ~
$ scp lajitku@144.24.14.145:/home/lajitku/October5.csv .
lajitku@144.24.14.145's password:
October5.csv                                                           100%  387     1.9KB/s   00:00
```

**November:**

```
select brand, count(brand) as count, cast(sum(price) as bigint) as sales, avg(price) as average_price from
cleanednovember where event_type like 'purchase' group by brand order by count(brand) desc limit 10;
```



```
+----------+---------+------------+---------------------+
|  brand   |  count  |   sales    |    average_price    |
+----------+---------+------------+---------------------+
| samsung  | 198670  | 54790697   | 275.78747470683527  |
| apple    | 165681  | 127490496  | 769.4937659116308   |
| xiaomi   | 57909   | 10874049   | 187.7782249736615   |
| huawei   | 23466   | 4768995    | 203.23002769965083  |
| oppo     | 15080   | 3488540    | 231.3355941644597   |
| lg       | 11828   | 5029641    | 425.2317923571167   |
| artel    | 7269    | 1329815    | 182.94340074288164  |
| lenovo   | 6546    | 2698104    | 412.17599450045907  |
| acer     | 6402    | 3347306    | 522.8532536707261   |
| bosch    | 5718    | 1276557    | 223.25236271423637  |
+----------+---------+------------+---------------------+
```

At Hive (beeline) terminal, create a csv file as an output using the script as follows:

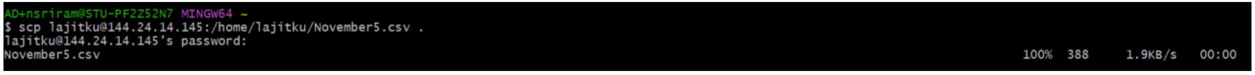
INSERT OVERWRITE DIRECTORY '/user/lajitku/tmp/'
 ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
select brand, count(brand) as count, cast(sum(price) as bigint) as sales, avg(price) as average_price from cleanednovember where event_type like 'purchase' group by brand order by count(brand) desc limit 10;

Go to the shell terminal to run the following command, which shows the file 000000_0:
-bash-4.2$ hdfs dfs -ls tmp/

Download the output file “000000_0” to “November5.csv” using the following hdfs command:
bash-4.2$ hdfs dfs -get tmp/000000_0 November5.csv

At your PC with git bash, xterminal, or pscp.exe, you can remotely download the output file “November5.csv” to your PC to visualize it using Excel .
NOTE: the following code has “.” at the end; You actually can connect from Tableau to the Hadoop cluster to get this file but Hadoop Cloud does not have the connector.

scp lajitku@144.24.14.145:/home/lajitku/November5.csv .

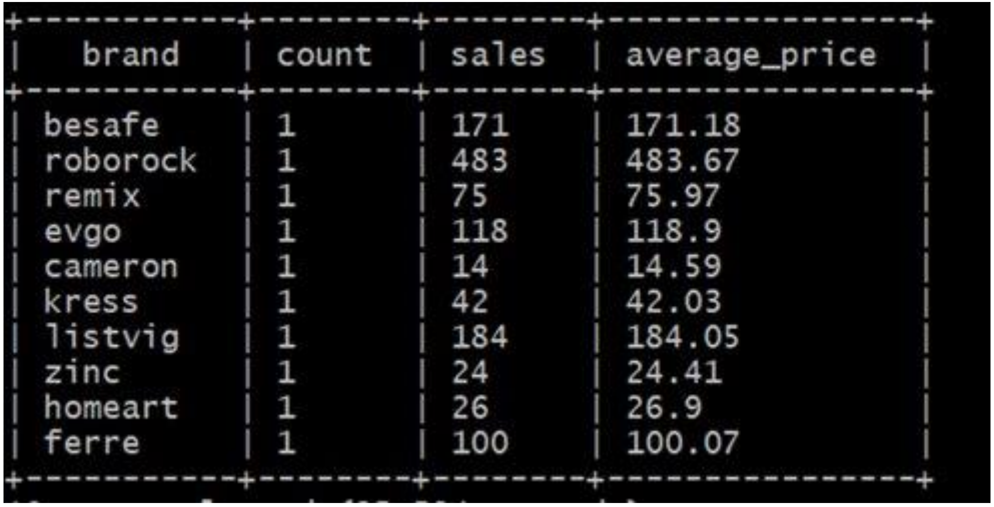


```
AD+nsriram@STU-PF2Z52N7 MINGW64 ~
$ scp lajitku@144.24.14.145:/home/lajitku/November5.csv .
lajitku@144.24.14.145's password:
November5.csv                                                                     100%  388     1.9KB/s   00:00
```

## 6. Top 10 Least Purchased Brands of October and November

### October:

select brand, count(brand) as count, cast(sum(price) as bigint) as sales, avg(price) as average_price from cleanedoctober where event_type like 'purchase' group by brand order by count(brand) limit 10;



```
+------------+---------+---------+-----------------+
|   brand    |  count  |  sales  |  average_price  |
+------------+---------+---------+-----------------+
| besafe     | 1       | 171     | 171.18          |
| roborock   | 1       | 483     | 483.67          |
| remix      | 1       | 75      | 75.97           |
| evgo       | 1       | 118     | 118.9           |
| cameron    | 1       | 14      | 14.59           |
| kress      | 1       | 42      | 42.03           |
| listvig    | 1       | 184     | 184.05          |
| zinc       | 1       | 24      | 24.41           |
| homeart    | 1       | 26      | 26.9            |
| ferre      | 1       | 100     | 100.07          |
+------------+---------+---------+-----------------+
```

At Hive (beeline) terminal, create a csv file as an output using the script as follows:

```
INSERT OVERWRITE DIRECTORY '/user/lajitku/tmp/'
 ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
select brand, count(brand) as count, cast(sum(price) as bigint) as sales, avg(price) as average_price from
cleanedoctober where event_type like 'purchase' group by brand order by count(brand) limit 10;
```

Go to the shell terminal to run the following command, which shows the file 000000_0:

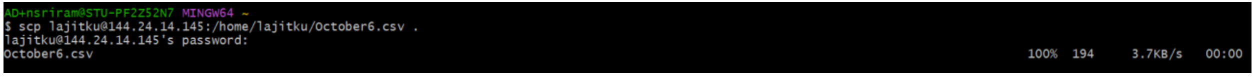
```
-bash-4.2$ hdfs dfs -ls tmp/
```

Download the output file “000000_0” to “October6.csv” using the following hdfs command:

```
bash-4.2$ hdfs dfs -get tmp/000000_0 October6.csv
```

At your PC with git bash, xterminal, or pscp.exe, you can remotely download the output file “October6.csv” to your PC to visualize it using Excel .
NOTE: the following code has “.” at the end; You actually can connect from Tableau to the Hadoop cluster to get this file but Hadoop Cloud does not have the connector.

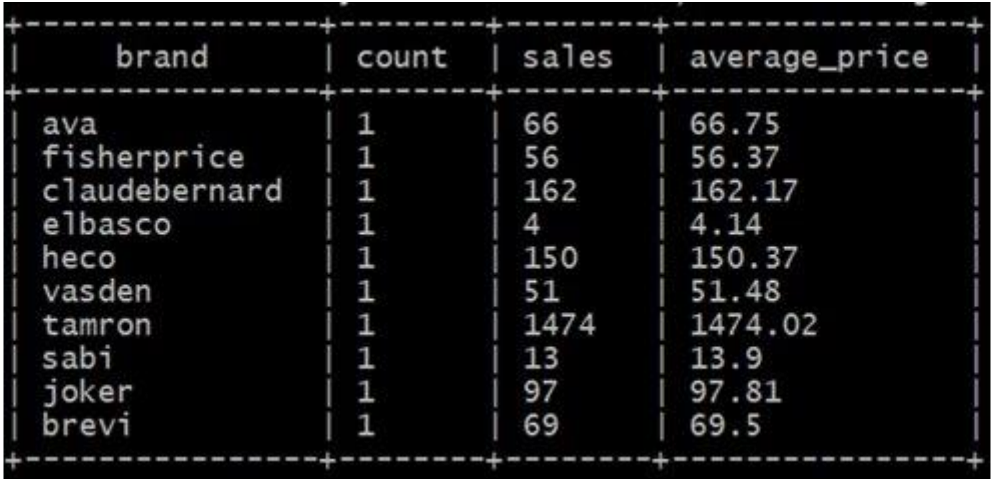
```
scp lajitku@144.24.14.145:/home/lajitku/October6.csv .
```



```
AD+nsriram@STU-PF2Z52N7 MINGW64 ~
$ scp lajitku@144.24.14.145:/home/lajitku/October6.csv .
lajitku@144.24.14.145's password:
October6.csv                                                     100%  194     3.7KB/s   00:00
```

**November:**

```
select brand, count(brand) as count, cast(sum(price) as bigint) as sales, avg(price) as average_price from
cleanednovember where event_type like 'purchase' group by brand order by count(brand) limit 10;
```



```
+----------------+--------+--------+----------------+
|     brand      | count  | sales  | average_price  |
+----------------+--------+--------+----------------+
| ava            | 1      | 66     | 66.75          |
| fisherprice    | 1      | 56     | 56.37          |
| claudebernard  | 1      | 162    | 162.17         |
| elbasco        | 1      | 4      | 4.14           |
| heco           | 1      | 150    | 150.37         |
| vasden         | 1      | 51     | 51.48          |
| tamron         | 1      | 1474   | 1474.02        |
| sabi           | 1      | 13     | 13.9           |
| joker          | 1      | 97     | 97.81          |
| brevi          | 1      | 69     | 69.5           |
+----------------+--------+--------+----------------+
```

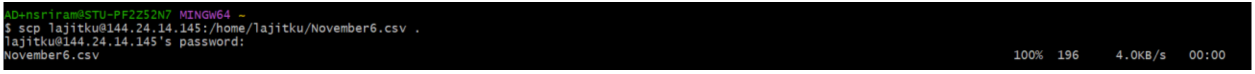
At Hive (beeline) terminal, create a csv file as an output using the script as follows:

```
INSERT OVERWRITE DIRECTORY '/user/lajitku/tmp/'
 ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
select brand, count(brand) as count, cast(sum(price) as bigint) as sales, avg(price) as average_price from
cleanednovember where event_type like 'purchase' group by brand order by count(brand) limit 10;
```

Go to the shell terminal to run the following command, which shows the file 000000_0:
-bash-4.2$ hdfs dfs -ls tmp/

Download the output file “000000_0” to “November6.csv” using the following hdfs command:
bash-4.2$ hdfs dfs -get tmp/000000_0 November6.csv
At your PC with git bash, xterminal, or pscp.exe, you can remotely download the output file “November6.csv” to your PC to visualize it using Excel .
NOTE: the following code has “.” at the end; You actually can connect from Tableau to the Hadoop cluster to get this file but Hadoop Cloud does not have the connector.

scp lajitku@144.24.14.145:/home/lajitku/November6.csv .

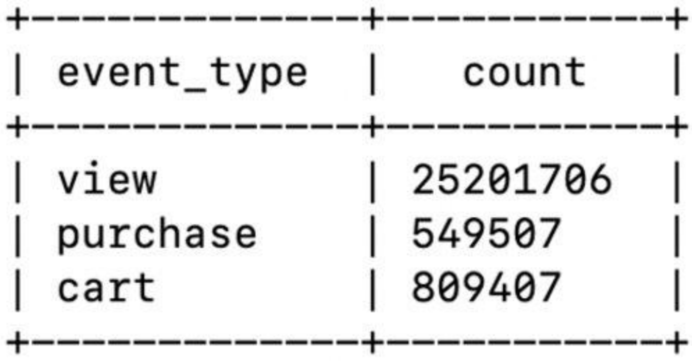


```
AD+nsriram@STU-PF2Z52N7 MINGW64 ~
$ scp lajitku@144.24.14.145:/home/lajitku/November6.csv .
lajitku@144.24.14.145's password:
November6.csv                                                                      100%  196     4.0KB/s   00:00
```

## 7. Views, Purchases, In-Carts in October and November

### October:

select event_type, count(event_type) as count from cleanedoctober group by event_type;



```
+-------------+-----------+
| event_type  |   count   |
+-------------+-----------+
| view        | 25201706  |
| purchase    | 549507    |
| cart        | 809407    |
+-------------+-----------+
```

At Hive (beeline) terminal, create a csv file as an output using the script as follows:

INSERT OVERWRITE DIRECTORY '/user/lajitku/tmp/'
 ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
select event_type, count(event_type) as count from cleanedoctober group by event_type;

Go to the shell terminal to run the following command, which shows the file 000000_0:
-bash-4.2$ hdfs dfs -ls tmp/

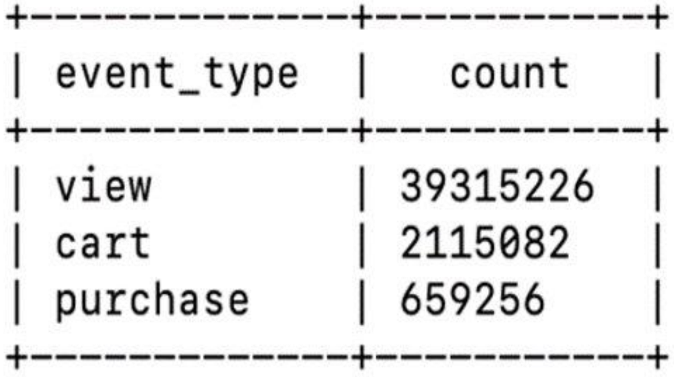
Download the output file “000000_0” to “October7.csv” using the following hdfs command:
bash-4.2$ hdfs dfs -get tmp/000000_0 October7.csv

At your PC with git bash, xterminal, or pscp.exe, you can remotely download the output file “October7.csv” to your PC to visualize it using Excel .
NOTE: the following code has “.” at the end; You actually can connect from Tableau to the Hadoop cluster to get this file but Hadoop Cloud does not have the connector.

```
scp lajitku@144.24.14.145:/home/lajitku/October7.csv .
```

### November:

```
select event_type, count(event_type) as count from cleanednovember group by event_type;
```

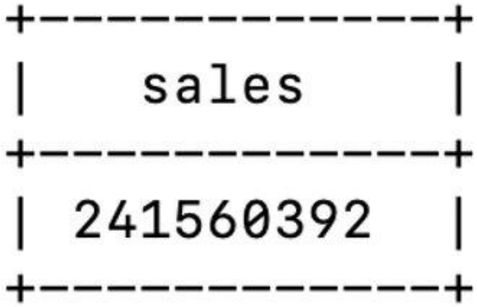


```
+-------------+------------+
| event_type  |   count    |
+-------------+------------+
| view        | 39315226   |
| cart        | 2115082    |
| purchase    | 659256     |
+-------------+------------+
```

## 8. Sum of Sales in both October and November

### October:

```
select cast(sum(price) as bigint) as sales from cleanedoctober where event_type like 'purchase’;
```



```
+------------+
|   sales    |
+------------+
| 241560392  |
+------------+
```

At Hive (beeline) terminal, create a csv file as an output using the script as follows:

```
INSERT OVERWRITE DIRECTORY '/user/lajitku/tmp/'
 ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
select cast(sum(price) as bigint) as sales from cleanedoctober where event_type like 'purchase';
```

Go to the shell terminal to run the following command, which shows the file 000000_0:

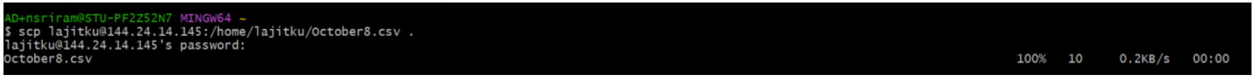
```
-bash-4.2$ hdfs dfs -ls tmp/
```

Download the output file “000000_0” to “October8.csv” using the following hdfs command:

```
bash-4.2$ hdfs dfs -get tmp/000000_0 October8.csv
```

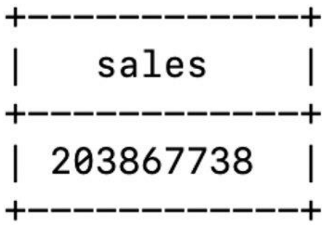
At your PC with git bash, xterminal, or pscp.exe, you can remotely download the output file "October8.csv" to your PC to visualize it using Excel .
NOTE: the following code has "." at the end; You actually can connect from Tableau to the Hadoop cluster to get this file but Hadoop Cloud does not have the connector.

scp lajitku@144.24.14.145:/home/lajitku/October8.csv .



```
AD+nsriram@STU-PF2Z52N7 MINGW64 ~
$ scp lajitku@144.24.14.145:/home/lajitku/October8.csv .
lajitku@144.24.14.145's password:
October8.csv                                                          100%   10     0.2KB/s   00:00
```

**November :**
select cast(sum(price) as bigint) as sales from cleanednovember where event_type like 'purchase';



```
+------------+
|   sales    |
+------------+
| 203867738  |
+------------+
```

At Hive (beeline) terminal, create a csv file as an output using the script as follows:

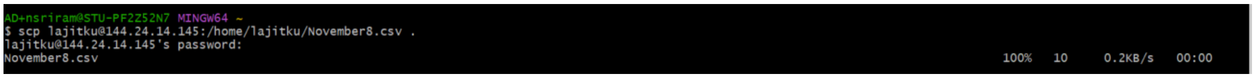
INSERT OVERWRITE DIRECTORY '/user/lajitku/tmp/'
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
select cast(sum(price) as bigint) as sales from cleanednovember where event_type like 'purchase';

Go to the shell terminal to run the following command, which shows the file 000000_0:
-bash-4.2$ hdfs dfs -ls tmp/

Download the output file "000000_0" to "November8.csv" using the following hdfs command:
bash-4.2$ hdfs dfs -get tmp/000000_0 November8.csv

At your PC with git bash, xterminal, or pscp.exe, you can remotely download the output file "November8.csv" to your PC to visualize it using Excel .
NOTE: the following code has "." at the end; You actually can connect from Tableau to the Hadoop cluster to get this file but Hadoop Cloud does not have the connector.

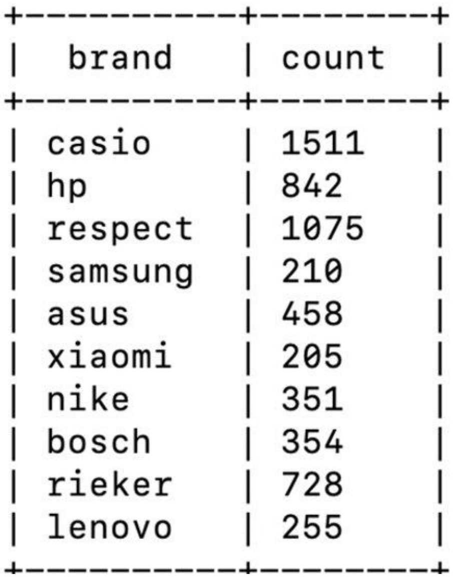
scp lajitku@144.24.14.145:/home/lajitku/November8.csv .



```
AD+nsriram@STU-PF2Z52N7 MINGW64 ~
$ scp lajitku@144.24.14.145:/home/lajitku/November8.csv .
lajitku@144.24.14.145's password:
November8.csv                                                         100%   10     0.2KB/s   00:00
```

### 9. Exit rate- Most viewed brand but not purchased

select brand, count(distinct product_id) as count from cleanedoctober where event_type = 'view' and product_id NOT IN (select product_id from cleanedoctober where event_type = 'purchase') group by brand order by count(product_id) desc limit 10;



```
+----------+--------+
|  brand   | count  |
+----------+--------+
| casio    | 1511   |
| hp       | 842    |
| respect  | 1075   |
| samsung  | 210    |
| asus     | 458    |
| xiaomi   | 205    |
| nike     | 351    |
| bosch    | 354    |
| rieker   | 728    |
| lenovo   | 255    |
+----------+--------+
```

At Hive (beeline) terminal, create a csv file as an output using the script as follows:

```
INSERT OVERWRITE DIRECTORY '/user/lajitku/tmp/'
 ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
select brand, count(distinct product_id) as count from cleanedoctober where event_type = 'view' and product_id NOT IN (select product_id from cleanedoctober where event_type = 'purchase') group by brand order by count(product_id) desc limit 10;
```

Go to the shell terminal to run the following command, which shows the file 000000_0:

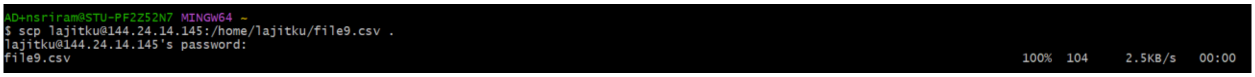
```
-bash-4.2$ hdfs dfs -ls tmp/
```

Download the output file “000000_0” to “file9.csv” using the following hdfs command:

```
bash-4.2$ hdfs dfs -get tmp/000000_0 file9.csv
```

At your PC with git bash, xterminal, or pscp.exe, you can remotely download the output file “November1.csv” to your PC to visualize it using Excel .
NOTE: the
following code has “.” at the end; You actually can connect from Tableau to the Hadoop cluster to get this file but Hadoop Cloud does not have the connector.

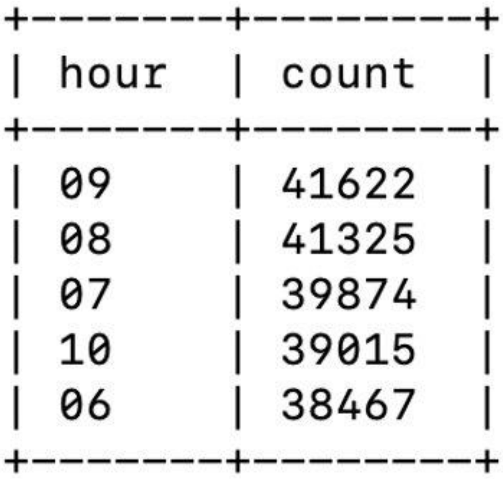
```
scp lajitku@144.24.14.145:/home/lajitku/file9.csv .
```



```
AD+nsriram@STU-PF2Z52N7 MINGW64 ~
$ scp lajitku@144.24.14.145:/home/lajitku/file9.csv .
lajitku@144.24.14.145's password:
file9.csv                                                                                        100%  104     2.5KB/s   00:00
```

## 10. Top 5 hours with most purchases in November

Select substr(event_time, 12, 2) as hour, count(substr(event_time, 12, 2)) as count from cleanednovember where event_type like 'purchase' group by substr(event_time, 12, 2) order by count(substr(event_time, 12, 2)) desc limit 5;



```
+-------+--------+
| hour  | count  |
+-------+--------+
| 09    | 41622  |
| 08    | 41325  |
| 07    | 39874  |
| 10    | 39015  |
| 06    | 38467  |
+-------+--------+
```

At Hive (beeline) terminal, create a csv file as an output using the script as follows:

```
INSERT OVERWRITE DIRECTORY '/user/lajitku/tmp/'
 ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
Select substr(event_time, 12, 2) as hour, count(substr(event_time, 12, 2)) as count from
cleanednovember where event_type like 'purchase' group by substr(event_time, 12, 2) order by
count(substr(event_time, 12, 2)) desc limit 5;
```

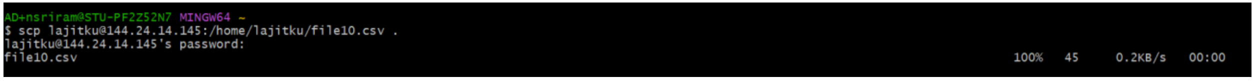
Go to the shell terminal to run the following command, which shows the file 000000_0:

```
-bash-4.2$ hdfs dfs -ls tmp/
```

Download the output file “000000_0” to “file10.csv” using the following hdfs command:
bash-4.2$ hdfs dfs -get tmp/000000_0 file10.csv

At your PC with git bash, xterminal, or pscp.exe, you can remotely download the output file “file10.csv” to your PC to visualize it using Excel .
NOTE: the following code has “.” at the end; You actually can connect from Tableau to the Hadoop cluster to get this file but Hadoop Cloud does not have the connector.

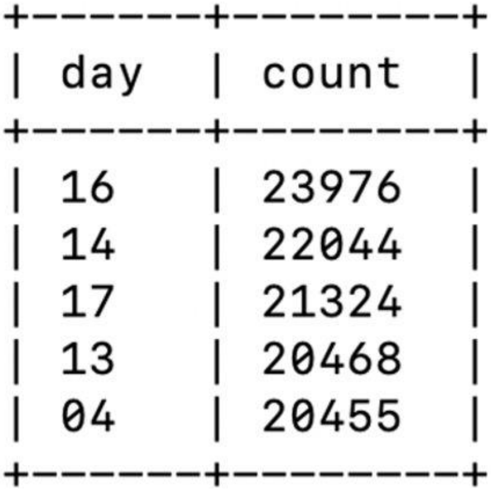
scp lajitku@144.24.14.145:/home/lajitku/file10.csv .



```
AD+nsriram@STU-PF2Z52N7 MINGW64 ~
$ scp lajitku@144.24.14.145:/home/lajitku/file10.csv .
lajitku@144.24.14.145's password:
file10.csv                                                                100%   45     0.2KB/s   00:00
```

### 11. Top 5 days with most purchases in October

Select substr(event_time, 9, 2) as day, count(substr(event_time, 9, 2)) as count from cleanedoctober where event_type = 'purchase' group by substr(event_time, 9, 2) order by count(substr(event_time, 9, 2)) desc limit 5;



```
+------+--------+
| day  | count  |
+------+--------+
| 16   | 23976  |
| 14   | 22044  |
| 17   | 21324  |
| 13   | 20468  |
| 04   | 20455  |
+------+--------+
```

At Hive (beeline) terminal, create a csv file as an output using the script as follows:

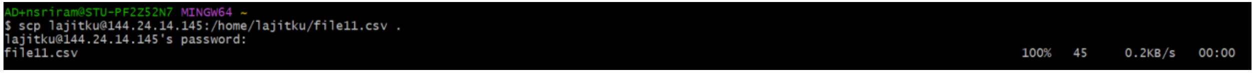
INSERT OVERWRITE DIRECTORY '/user/lajitku/tmp/'
 ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
Select substr(event_time, 9, 2) as day, count(substr(event_time, 9, 2)) as count from cleanedoctober where event_type = 'purchase' group by substr(event_time, 9, 2) order by count(substr(event_time, 9, 2)) desc limit 5;

Go to the shell terminal to run the following command, which shows the file 000000_0:
-bash-4.2$ hdfs dfs -ls tmp/

Download the output file “000000_0” to “file11.csv” using the following hdfs command:
bash-4.2$ hdfs dfs -get tmp/000000_0 file11.csv

At your PC with git bash, xterminal, or pscp.exe, you can remotely download the output file “file11.csv” to your PC to visualize it using Excel .
NOTE: the following code has “.” at the end; You actually can connect from Tableau to the Hadoop cluster to get this file but Hadoop Cloud does not have the connector.

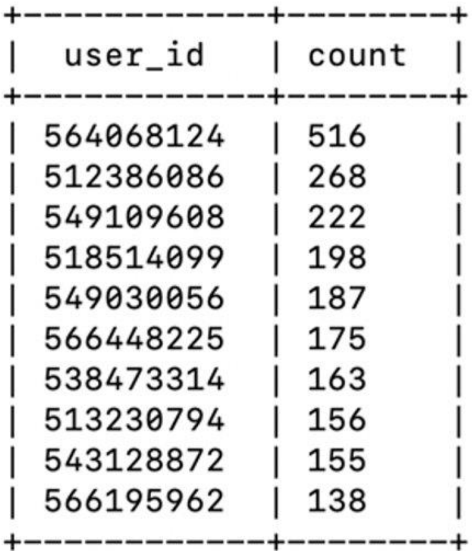
scp lajitku@144.24.14.145:/home/lajitku/file11.csv .



```
AD+nsriram@STU-PF2Z52N7 MINGW64 ~
$ scp lajitku@144.24.14.145:/home/lajitku/file11.csv .
lajitku@144.24.14.145's password:
file11.csv                                                          100%   45     0.2KB/s   00:00
```

**12. Top 10 Users who made the most purchases in November**

select user_id, count(user_id) as count from cleanednovember where event_type = 'purchase' group by user_id order by count(user_id) limit 10;



```
+------------+--------+
|  user_id   | count  |
+------------+--------+
| 564068124  | 516    |
| 512386086  | 268    |
| 549109608  | 222    |
| 518514099  | 198    |
| 549030056  | 187    |
| 566448225  | 175    |
| 538473314  | 163    |
| 513230794  | 156    |
| 543128872  | 155    |
| 566195962  | 138    |
+------------+--------+
```

At Hive (beeline) terminal, create a csv file as an output using the script as follows:

INSERT OVERWRITE DIRECTORY '/user/lajitku/tmp/'
 ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
select user_id, count(user_id) as count from cleanednovember where event_type = 'purchase' group by user_id order by count(user_id) limit 10;

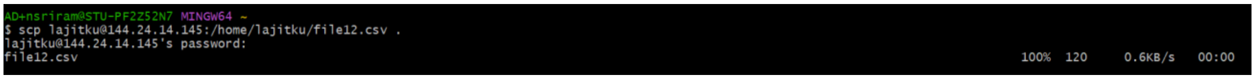
Go to the shell terminal to run the following command, which shows the file 000000_0:
-bash-4.2$ hdfs dfs -ls tmp/

Download the output file “000000_0” to “file12.csv” using the following hdfs command:
bash-4.2$ hdfs dfs -get tmp/000000_0 file12.csv

At your PC with git bash, xterminal, or pscp.exe, you can remotely download the output file “file12.csv” to your PC to visualize it using Excel .

NOTE: the following code has “.” at the end; You actually can connect from Tableau to the Hadoop cluster to get this file but Hadoop Cloud does not have the connector.

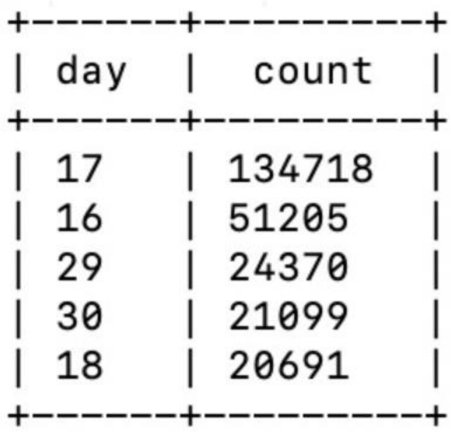
scp lajitku@144.24.14.145:/home/lajitku/file12.csv .



```
AD+nsriram@STU-PF2Z52N7 MINGW64 ~
$ scp lajitku@144.24.14.145:/home/lajitku/file12.csv .
lajitku@144.24.14.145's password:
file12.csv                                                                                     100%  120     0.6KB/s   00:00
```

**13. Top 5 days with most purchases in November**

Select substr(event_time, 9, 2) as day, count(substr(event_time, 9, 2)) as count from cleanednovember where event_type = 'purchase' group by substr(event_time, 9, 2) order by count(substr(event_time, 9, 2)) desc limit 5;



```
+------+---------+
| day  |  count  |
+------+---------+
| 17   | 134718  |
| 16   | 51205   |
| 29   | 24370   |
| 30   | 21099   |
| 18   | 20691   |
+------+---------+
```

At Hive (beeline) terminal, create a csv file as an output using the script as follows:

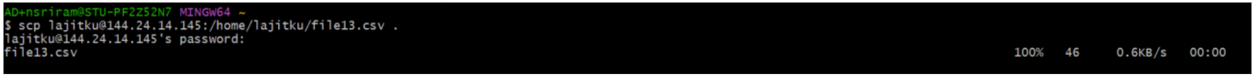
```
INSERT OVERWRITE DIRECTORY '/user/lajitku/tmp/'
 ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
Select substr(event_time, 9, 2) as day, count(substr(event_time, 9, 2)) as count from cleanednovember
where event_type = 'purchase' group by substr(event_time, 9, 2) order by count(substr(event_time, 9,
2)) desc limit 5;
```

Go to the shell terminal to run the following command, which shows the file 000000_0:
-bash-4.2$ hdfs dfs -ls tmp/

Download the output file “000000_0” to “file13.csv” using the following hdfs command:
bash-4.2$ hdfs dfs -get tmp/000000_0 file13.csv

At your PC with git bash, xterminal, or pscp.exe, you can remotely download the output file “file13.csv” to your PC to visualize it using Excel .
NOTE: the following code has “.” at the end; You actually can connect from Tableau to the Hadoop cluster to get this file but Hadoop Cloud does not have the connector.

scp lajitku@144.24.14.145:/home/lajitku/file13.csv .



```
AD+nsriram@STU-PF2Z52N7 MINGW64 ~
$ scp lajitku@144.24.14.145:/home/lajitku/file13.csv .
lajitku@144.24.14.145's password:
file13.csv                                                                     100%   46     0.6KB/s   00:00
```

## Step 5: Visualization using Excel and Tableau

This step is to show the Visualization for the above Queries.

**NOTE:** All the Visualizations are done using Excel except the fifth and Thirteenth (Top 10 Purchased Brands of October and November)

To visualize results on Graphs, convert csv file to excel and click on Graphs button under insert tab.

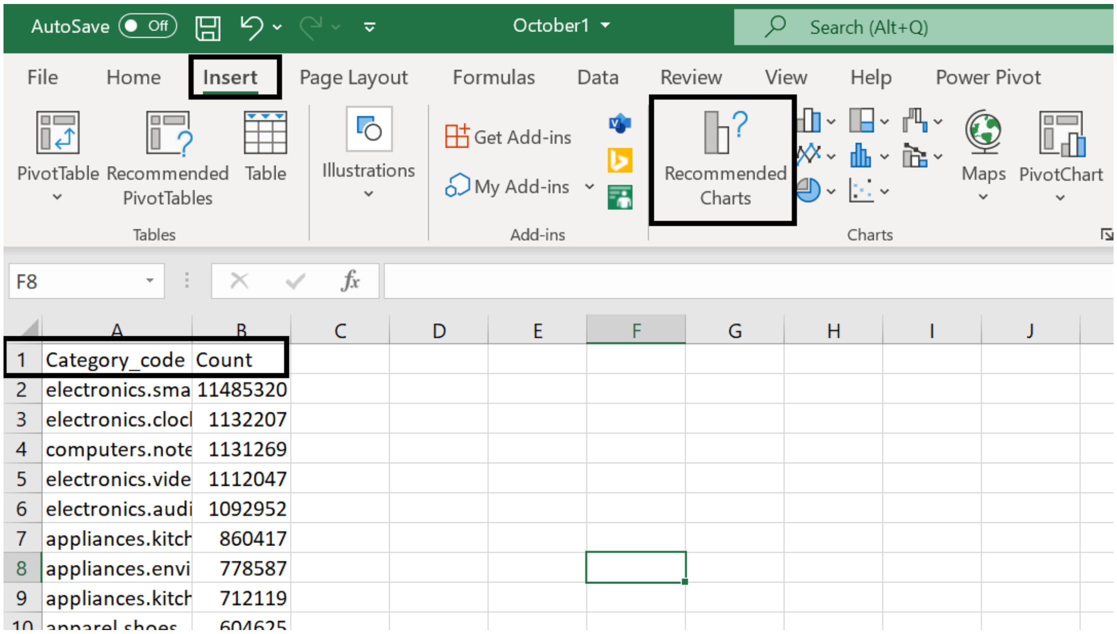
open "October1.csv" at excel. Open your Excel first, then open the data file from Excel in order to read the data as multiple records in multiple rows.

**NOTE:** if your data is displayed in a single row, it is not correct. Thus, you have to find out the way to display it in multiple rows.

For the first row of the file, you need to insert the header to each column as follows:

Category_code Count

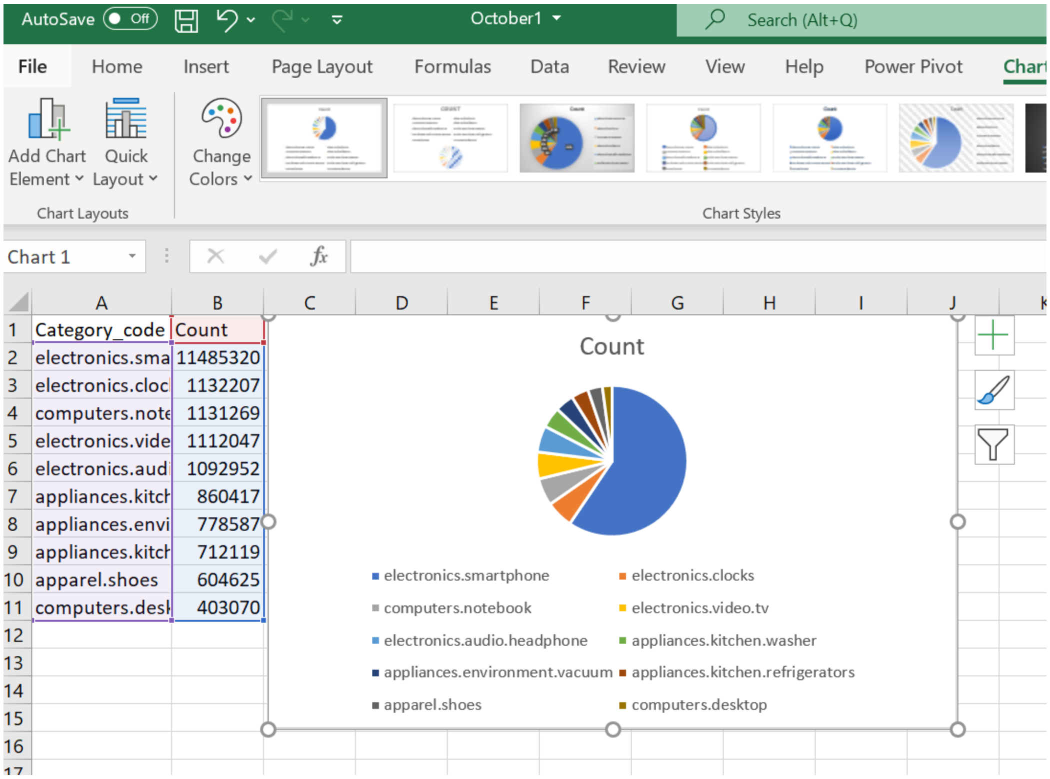
Then, Go to “insert” tab to find out the menu



|  | A | B |
|---|---|---|
| 1 | Category_code | Count |
| 2 | electronics.sma | 11485320 |
| 3 | electronics.clocl | 1132207 |
| 4 | computers.note | 1131269 |
| 5 | electronics.vide | 1112047 |
| 6 | electronics.audi | 1092952 |
| 7 | appliances.kitch | 860417 |
| 8 | appliances.envi | 778587 |
| 9 | appliances.kitch | 712119 |
| 10 | apparel.shoes | 604625 |

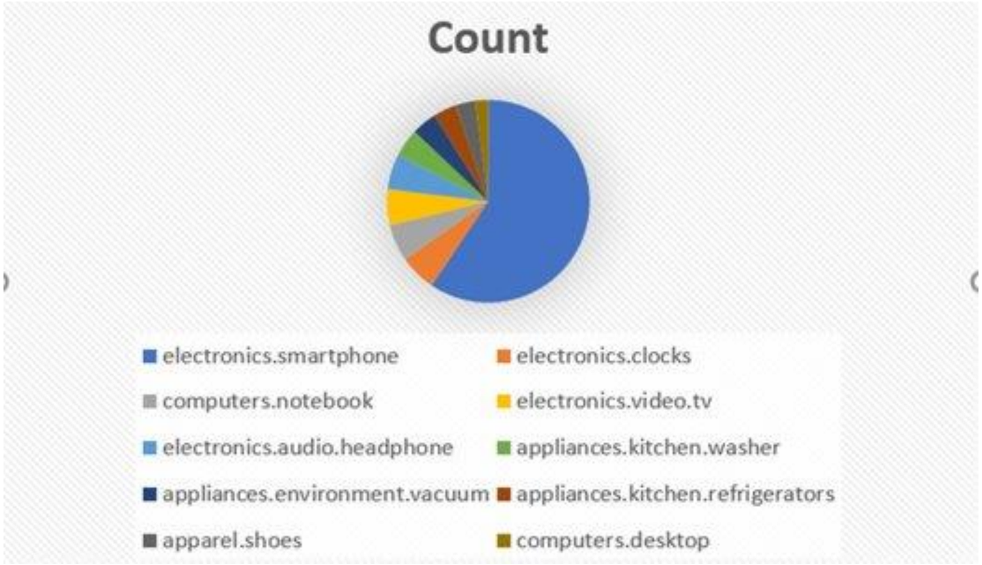
You will see the following Recommend Chart.

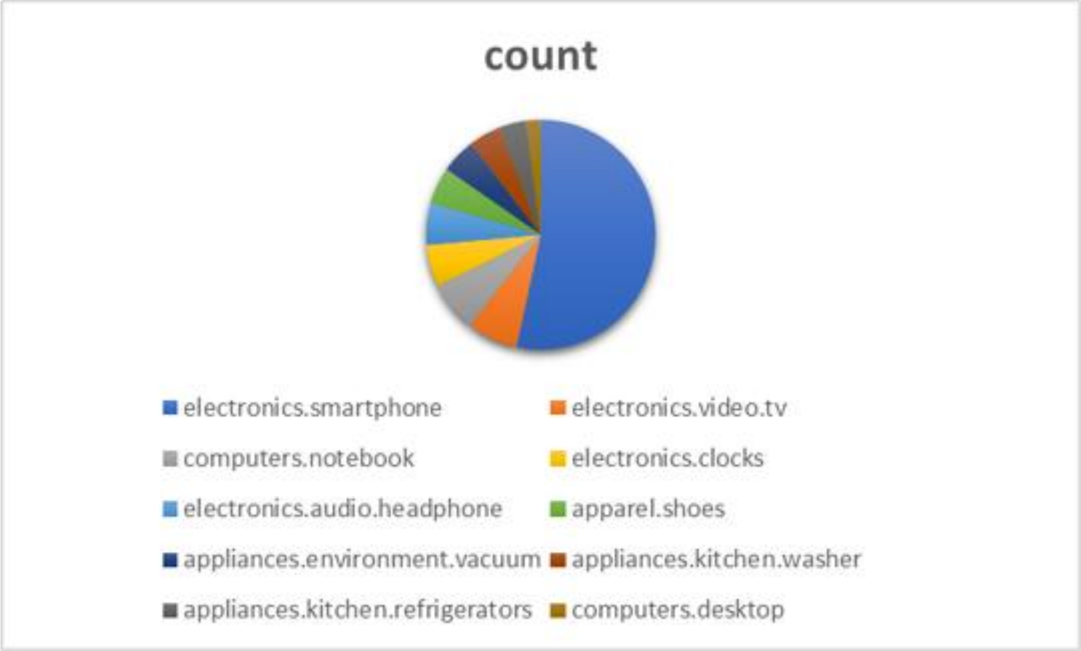
NOTE: If you don’t see the layer frame in the right side, you may select all data manually before opening the Chart:



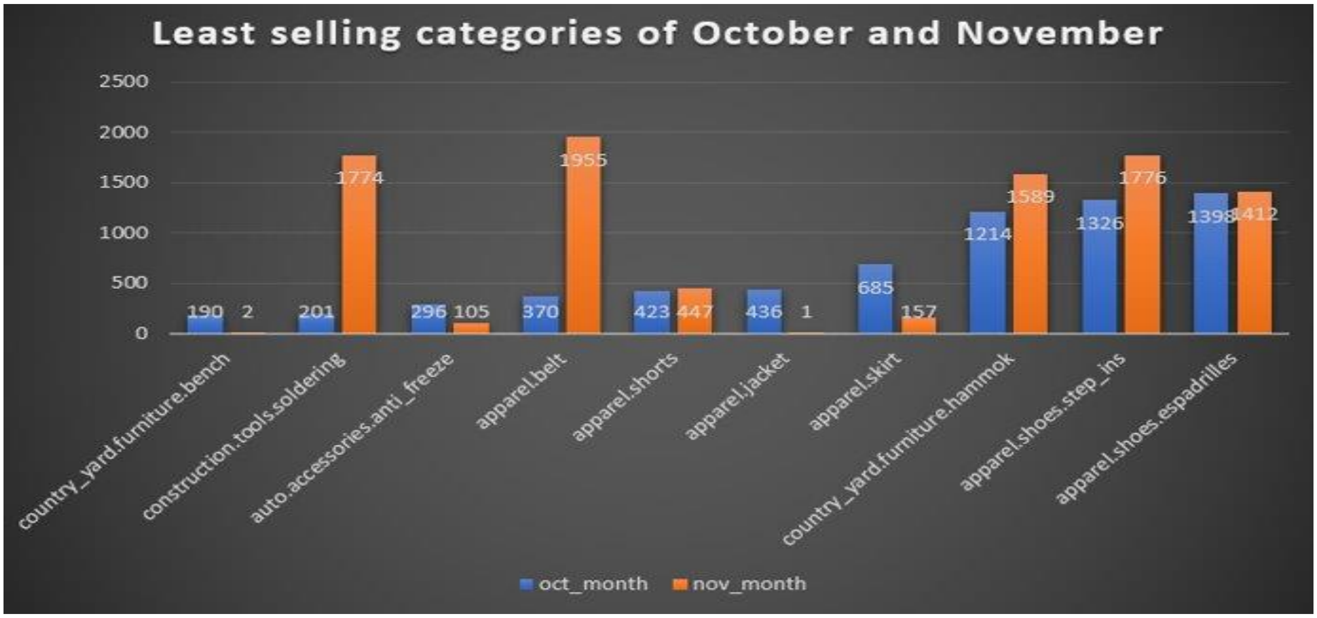
| | A | B |
|---|---|---|
| 1 | Category_code | Count |
| 2 | electronics.sma | 11485320 |
| 3 | electronics.cloc | 1132207 |
| 4 | computers.note | 1131269 |
| 5 | electronics.vide | 1112047 |
| 6 | electronics.aud | 1092952 |
| 7 | appliances.kitch | 860417 |
| 8 | appliances.envi | 778587 |
| 9 | appliances.kitch | 712119 |
| 10 | apparel.shoes | 604625 |
| 11 | computers.desk | 403070 |
| 12 | | |
| 13 | | |
| 14 | | |
| 15 | | |
| 16 | | |
| 17 | | |

### 1. Top 10 Popular categories in October and November

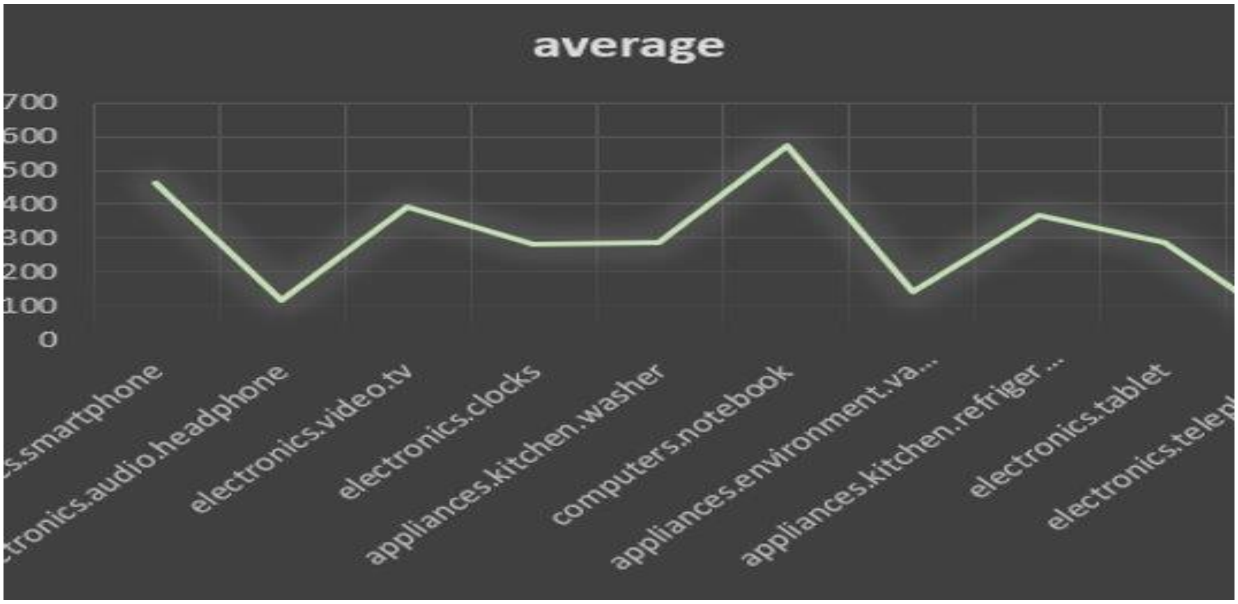


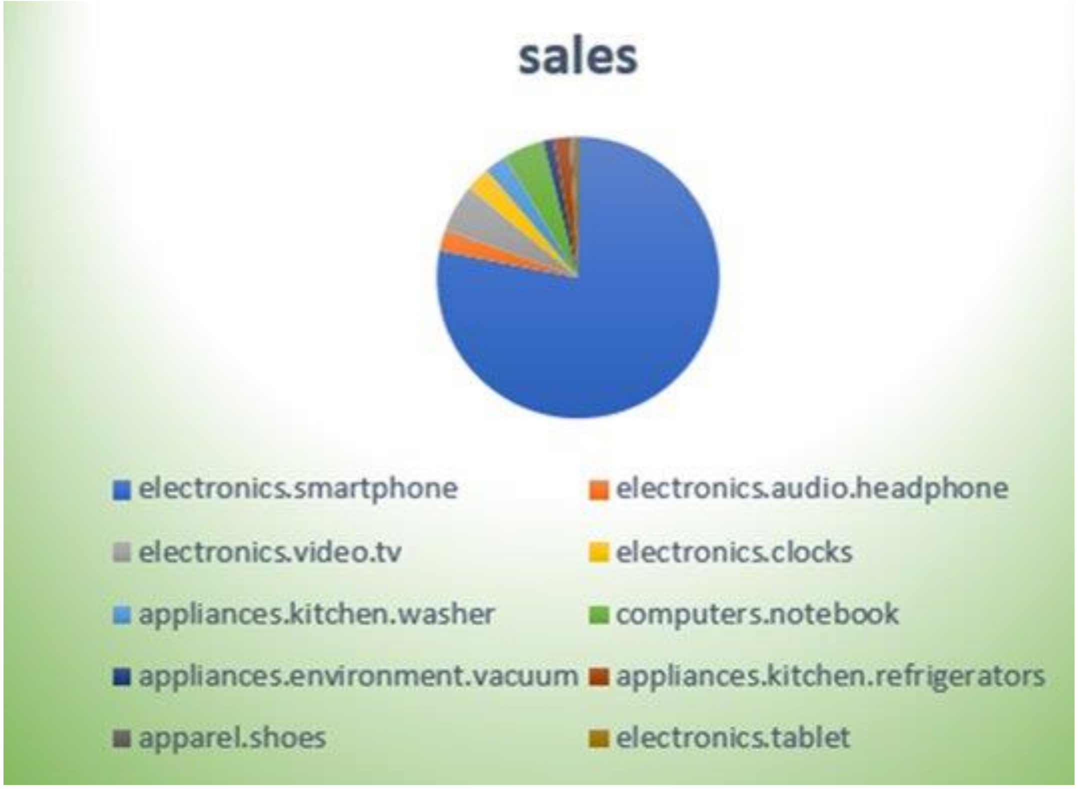


**2. Top 10 Least popular categories in October and November**

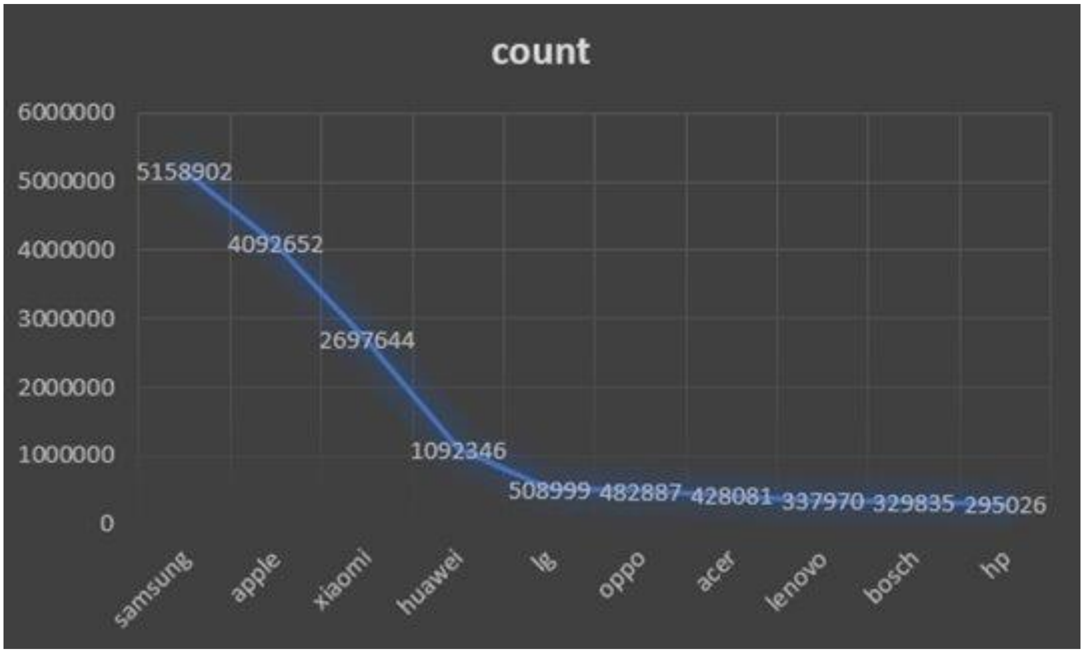


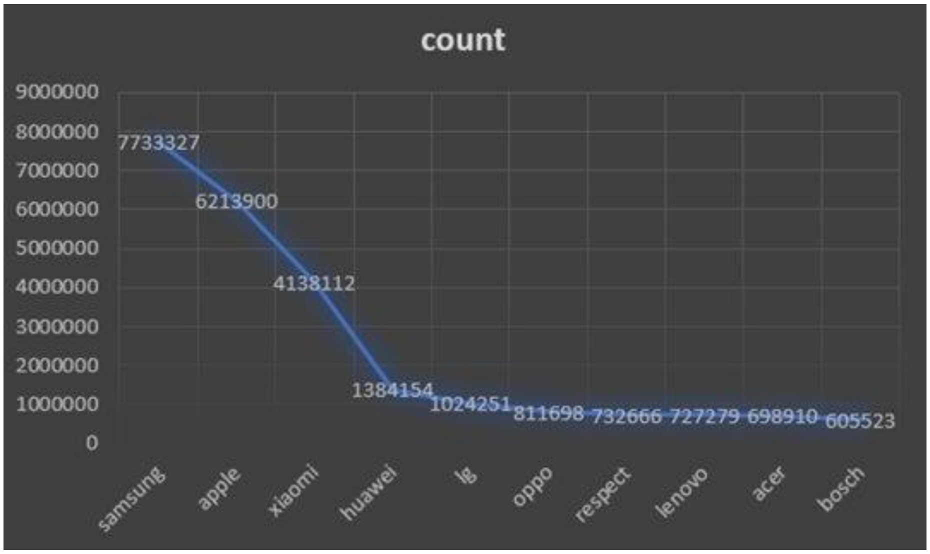
**3. Top 10 purchased categories, sales count and average price in October and November.**





**4. Top 10 popular brands October and November**





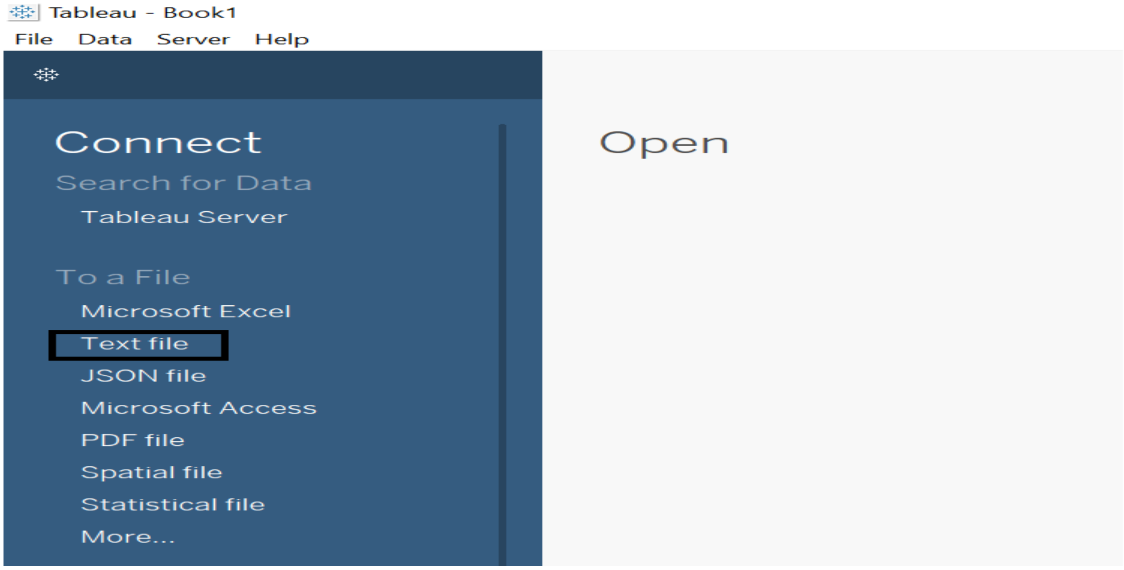
**5. Top 10 Purchased Brands of October and November**

**TABLEAU TO IMPORT HADOOP FILE AT HADOOP CLOUD**

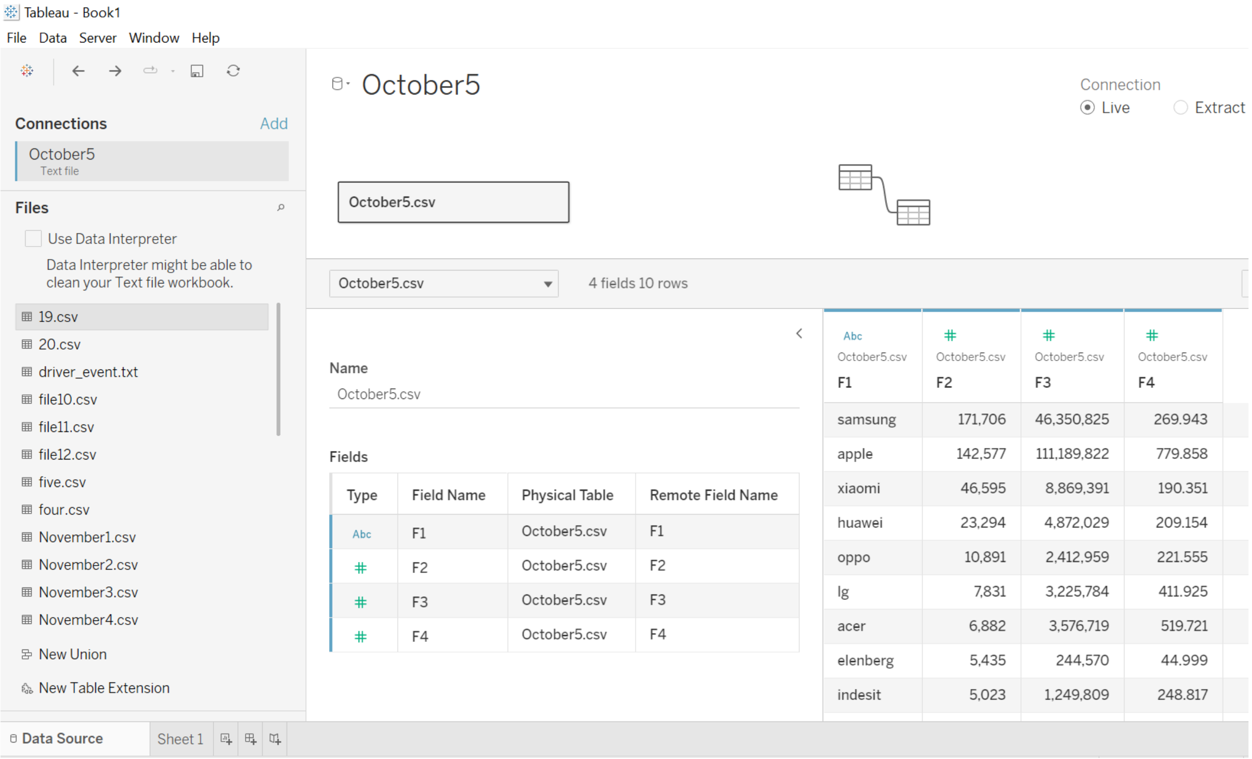
Open your tableau at your local computer

TABLEA TO OPEN DATA FILE DIRECTLY FROM TABLEAU AND VISUALIZATION

1. Open your Tableau to connect your server. You need to select Text File to open the file October5



2. You will see the following data at Data Source – F1: Brand, F2: Count, F3: Sales, F4: Average Price.



Connection
● Live ○ Extract

October5.csv 4 fields 10 rows

Name
October5.csv

Fields

| Type | Field Name | Physical Table | Remote Field Name |
|---|---|---|---|
| Abc | F1 | October5.csv | F1 |
| # | F2 | October5.csv | F2 |
| # | F3 | October5.csv | F3 |
| # | F4 | October5.csv | F4 |

| Abc<br>October5.csv<br>F1 | #<br>October5.csv<br>F2 | #<br>October5.csv<br>F3 | #<br>October5.csv<br>F4 |
|---|---|---|---|
| samsung | 171,706 | 46,350,825 | 269.943 |
| apple | 142,577 | 111,189,822 | 779.858 |
| xiaomi | 46,595 | 8,869,391 | 190.351 |
| huawei | 23,294 | 4,872,029 | 209.154 |
| oppo | 10,891 | 2,412,959 | 221.555 |
| lg | 7,831 | 3,225,784 | 411.925 |
| acer | 6,882 | 3,576,719 | 519.721 |
| elenberg | 5,435 | 244,570 | 44.999 |
| indesit | 5,023 | 1,249,809 | 248.817 |

Select Sheet 1 next to Data Source, which will present the following frame. Then, rename F1, F2, F3 by right-clicking each value as F1: Brand, F2: Count, F3: Sales, F4: Average Price. Then, change its data type as: Brand (String), Count (Whole Number), Sales (Whole Number), Average Price(Decimal Number):

Now You have to drag Brand to Columns and Average_Price, Sales and Count to Rows.

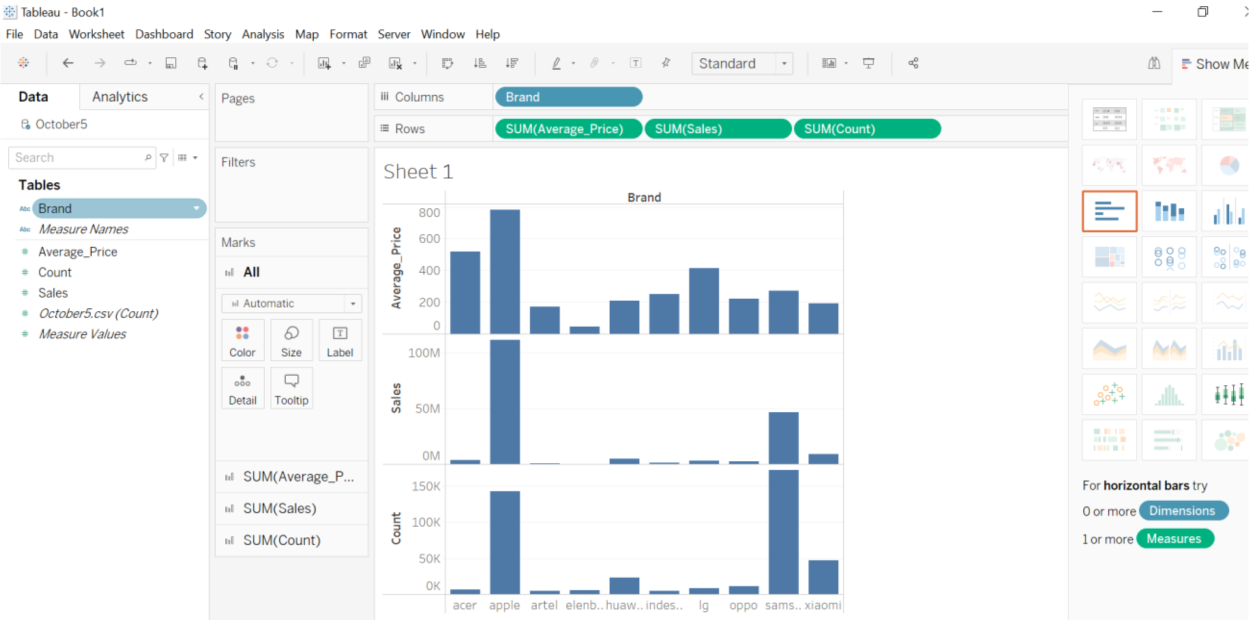


Tableau - Book1

File Data Worksheet Dashboard Story Analysis Map Format Server Window Help

Data | Analytics

October5

Tables
- Brand
- *Measure Names*
- Average_Price
- Count
- Sales
- *October5.csv (Count)*
- *Measure Values*

Pages

Filters

Marks: All — Automatic
- SUM(Average_P...
- SUM(Sales)
- SUM(Count)

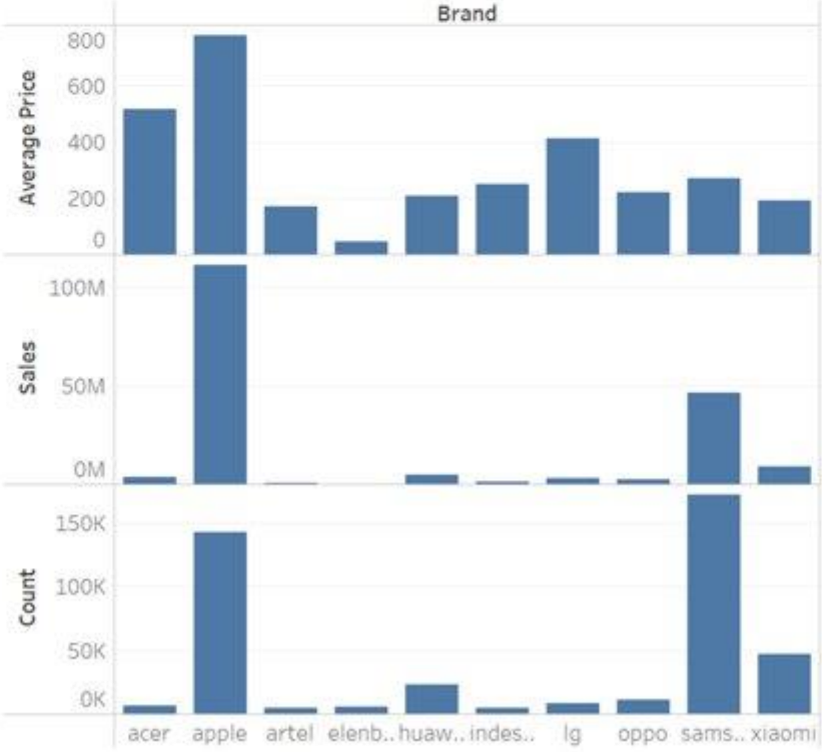
Columns: Brand

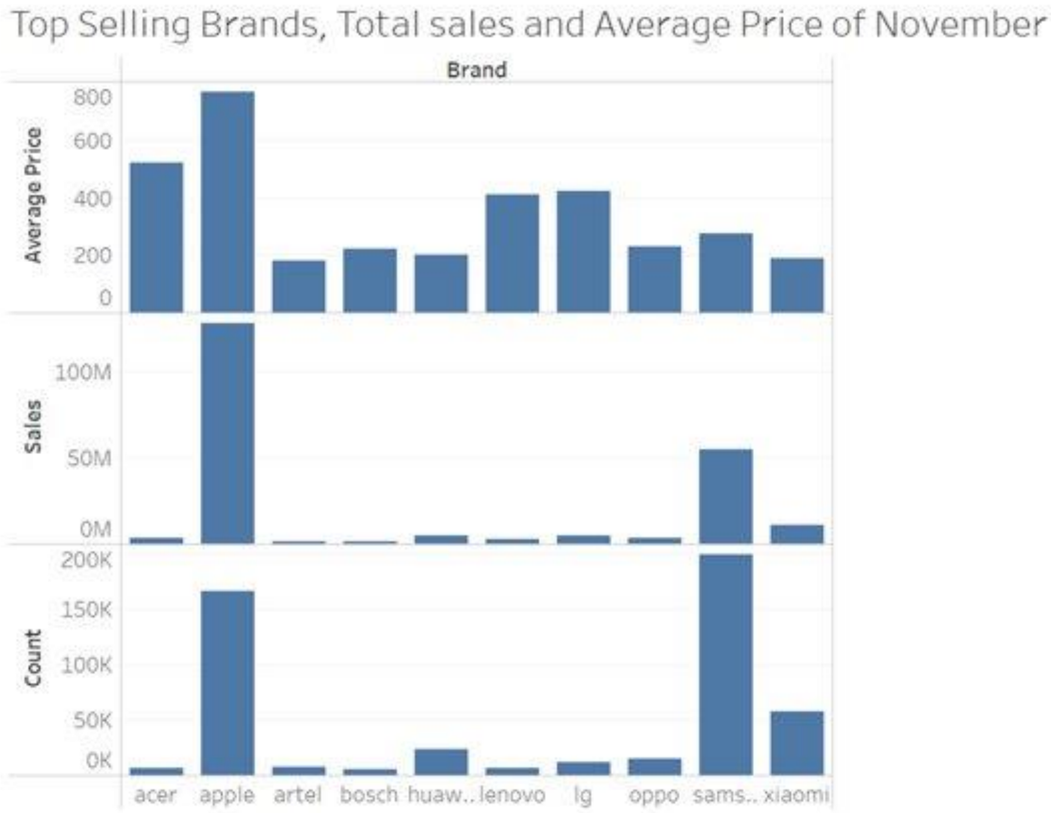
Rows: SUM(Average_Price) SUM(Sales) SUM(Count)

Sheet 1

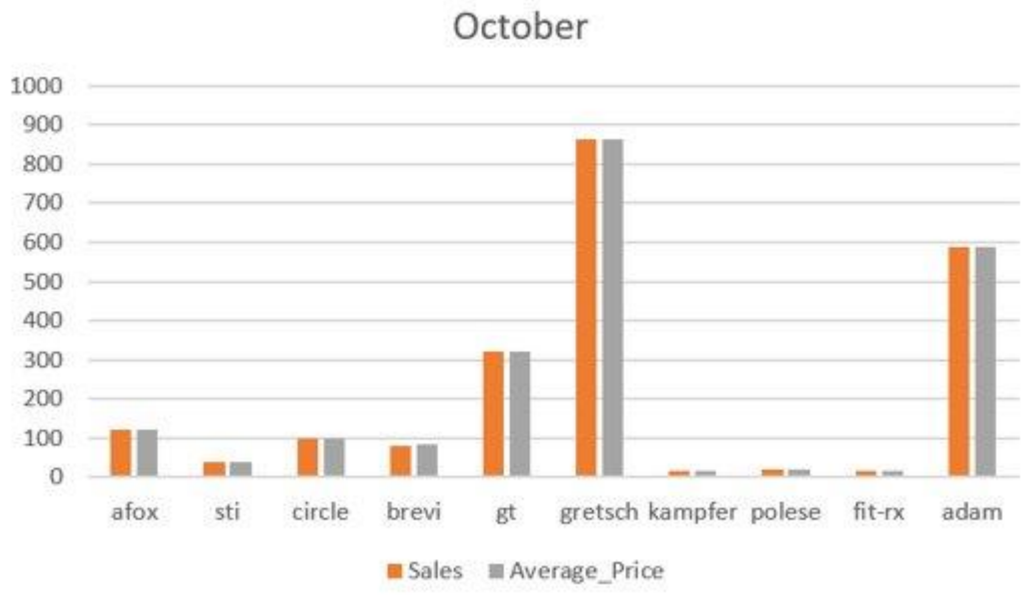
For **horizontal bars** try
0 or more Dimensions
1 or more Measures

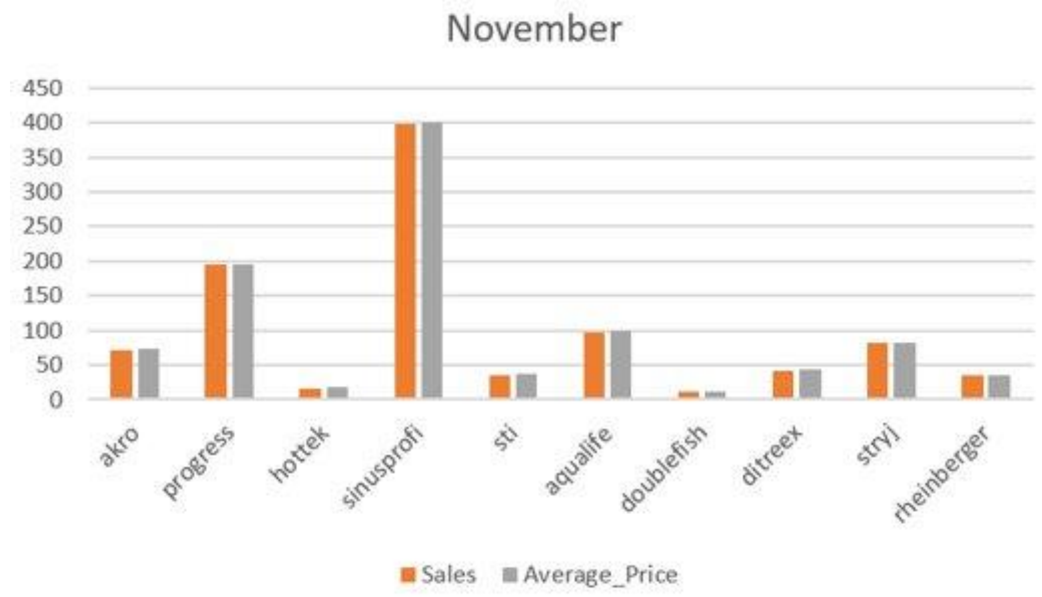
Top Selling Brands, Total Sales and Average Price of October



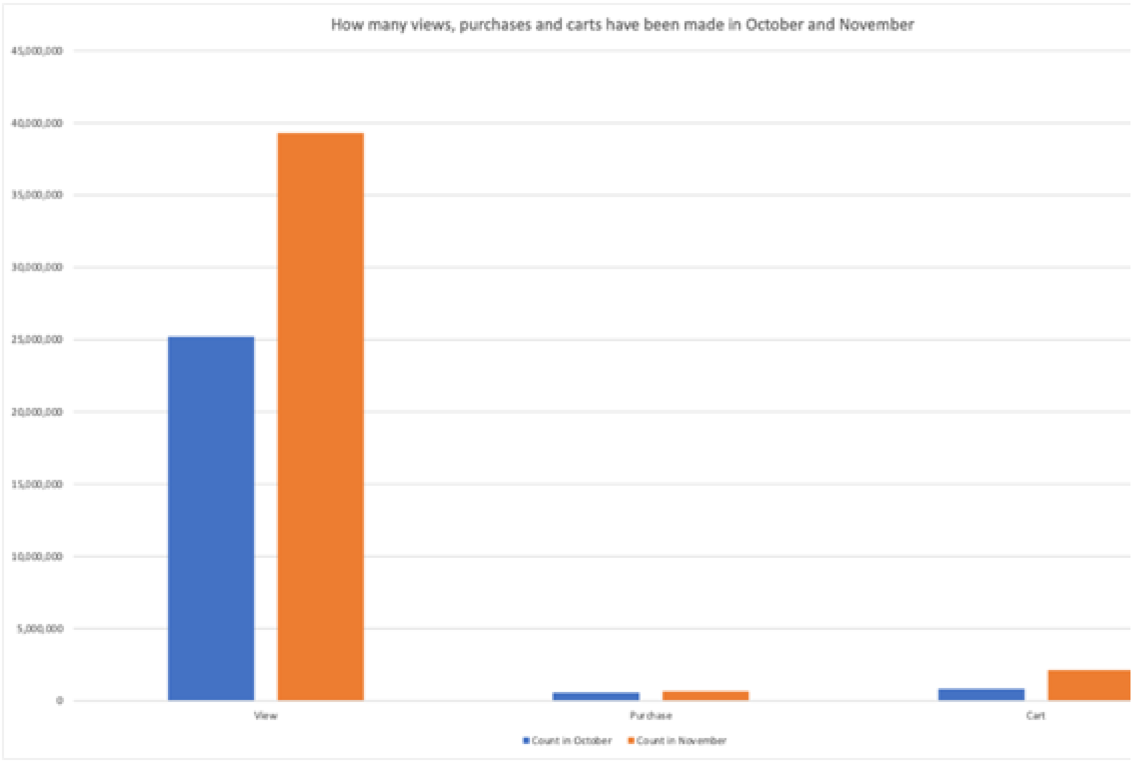


## 6. Top 10 Least Purchased Brands of October and November

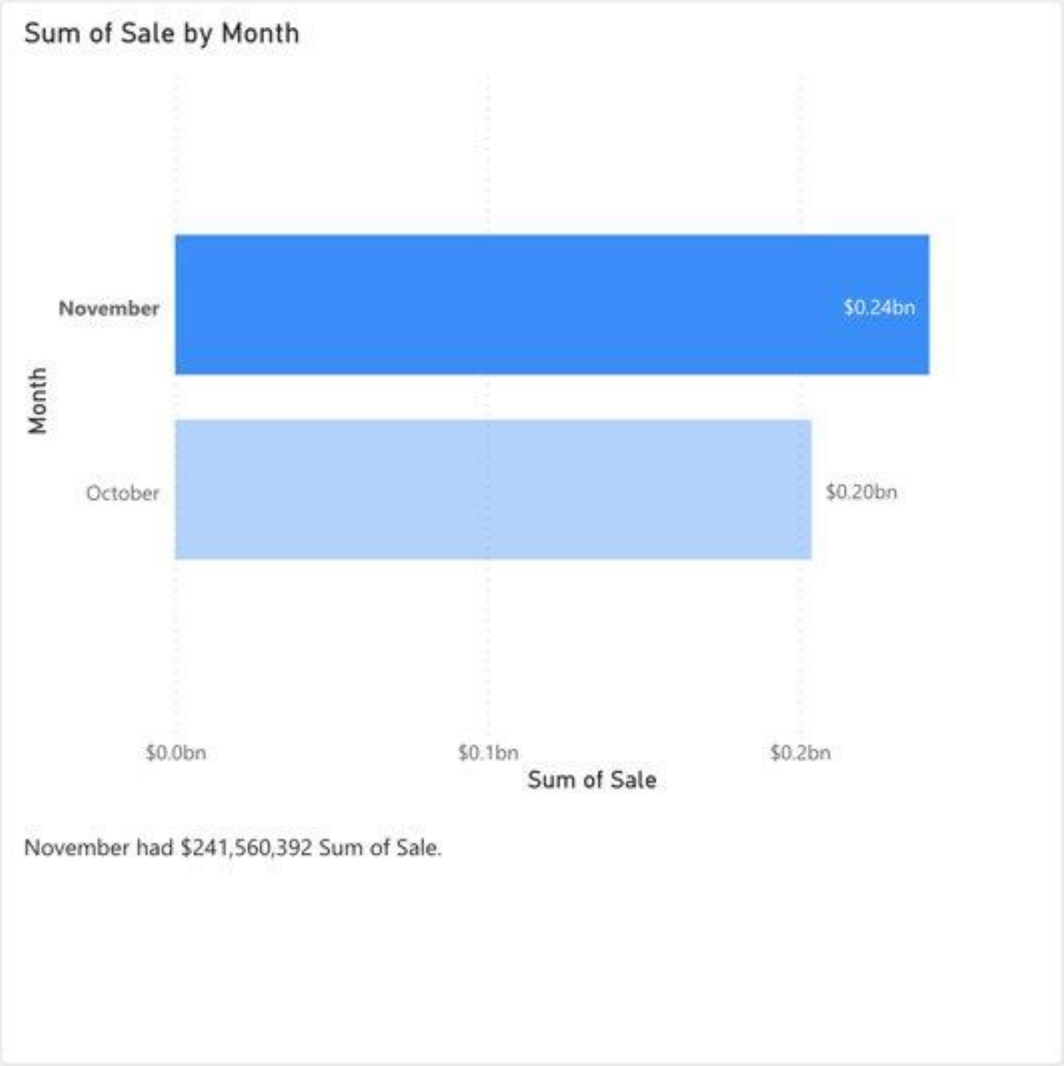




### 7. Views, Purchases, In-Carts in October and November

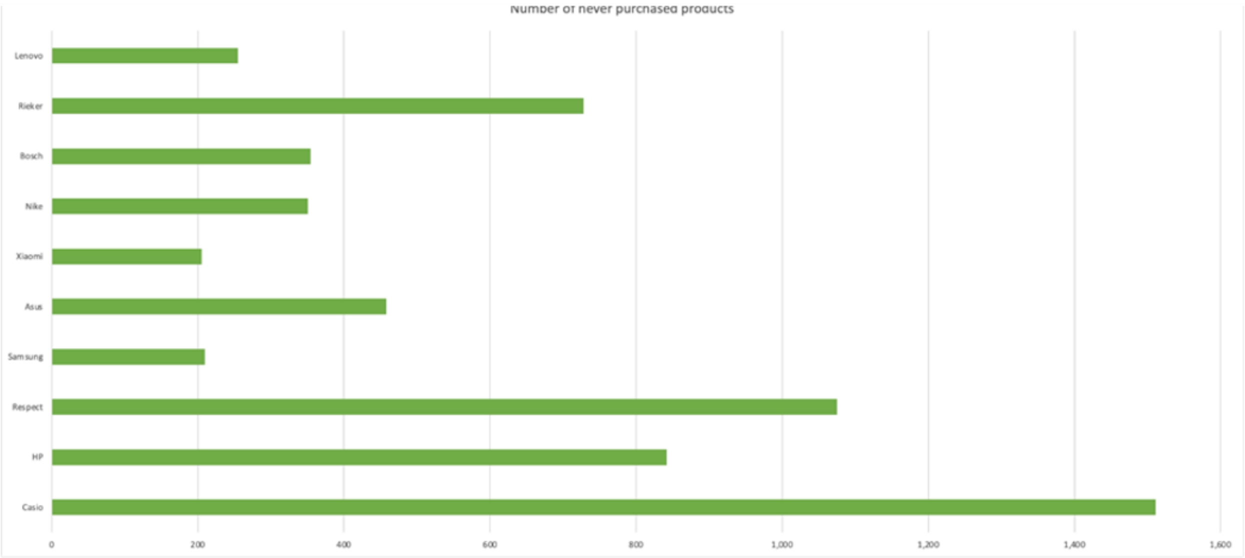


### 8. Sum of Sales in both October and November

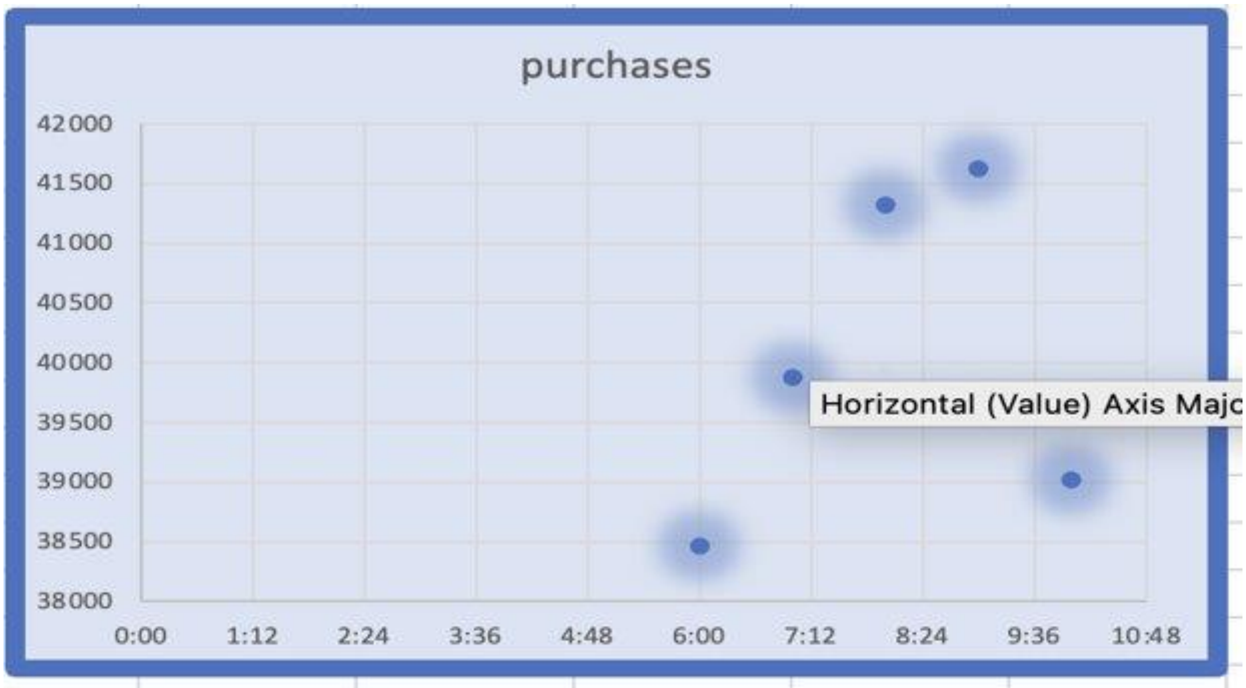


November had $241,560,392 Sum of Sale.

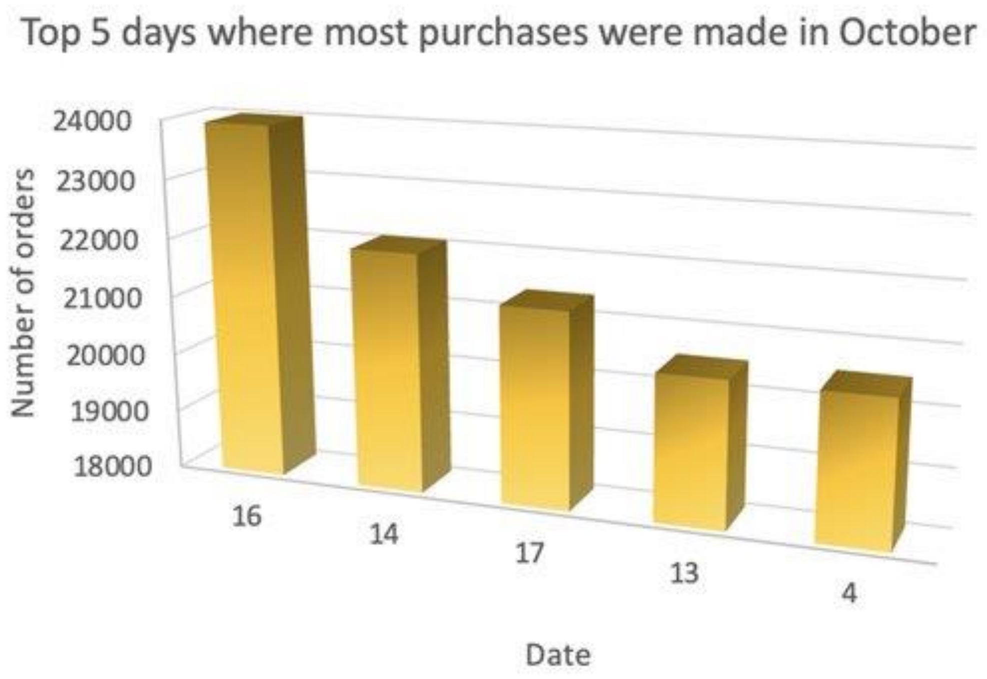
### 9. Exit Rate - Most viewed brand but not purchased



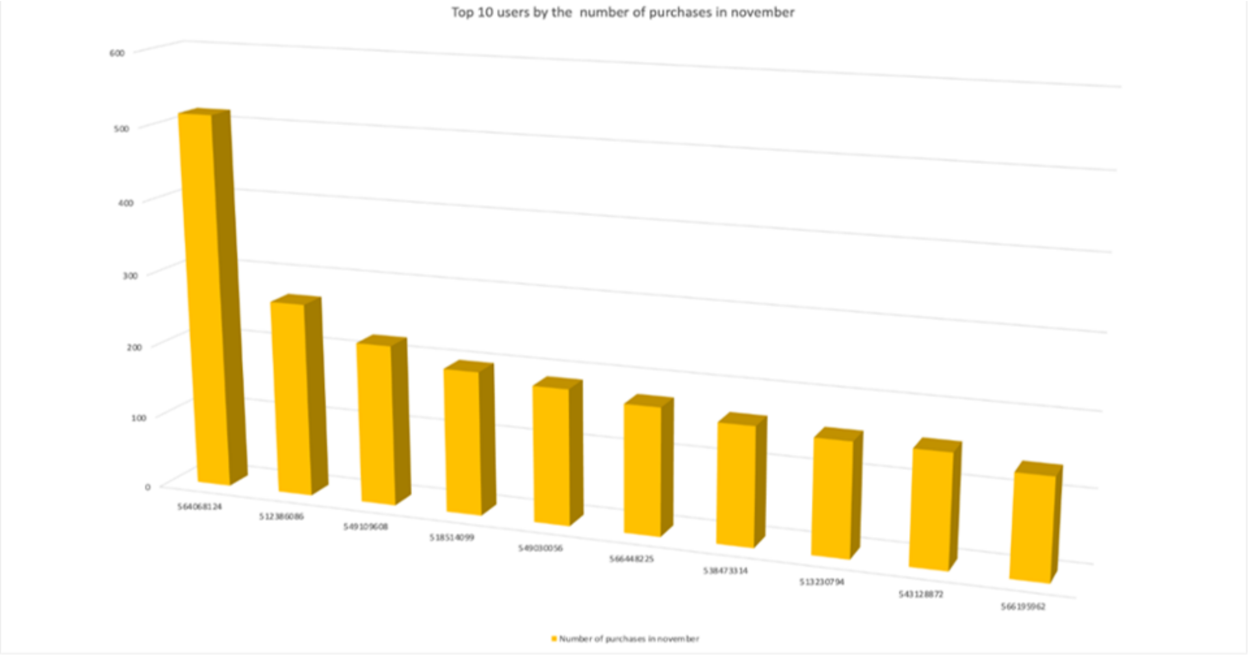
### 10. Top 5 hours with most purchases in November



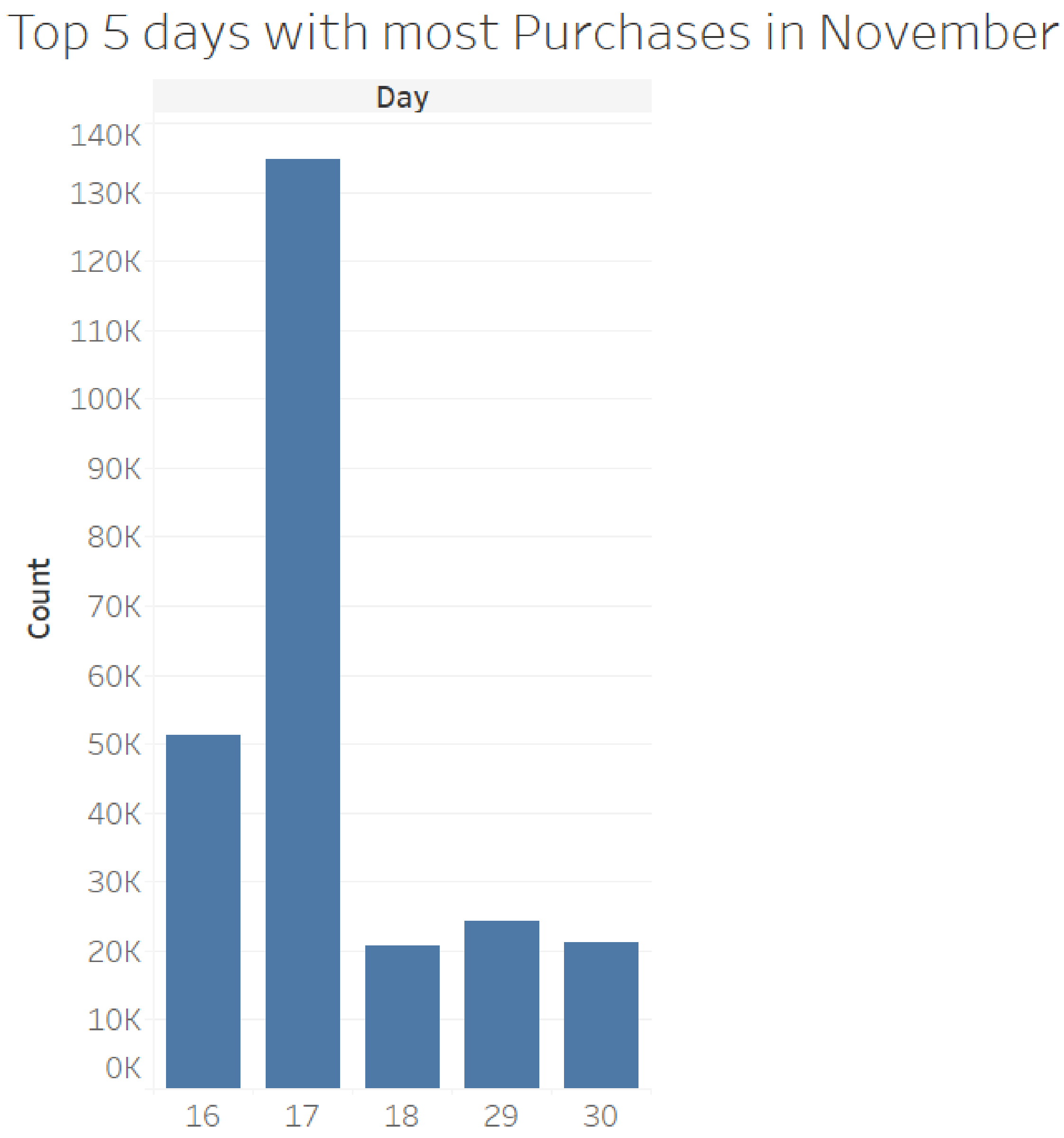
## 11. Top 5 days with most purchases in October



## 12. Top 10 Users who made the most purchases in November



**13. Top 5 days with most purchases in November**



## References

1. URL of Data Source: eCommerce behavior data from multi category store | Kaggle
2. URL of your Github: https://github.com/Lekha19202/E-commerce-customer-behaviour-uding-Hadoop.git
3. URL of References:
    - https://sanyasachdeva1.github.io/Portfolio/files/Analysis%20of%20e-commerce%20behavior%20in%20Multi-Category%20Store.pdf
    - https://stackoverflow.com/questions/51097895/hive-sql-find-most-popular-value-across-multiple-columns