# [C:\Users\jwoo5\AppData\Local\Temp\templateTermTutorial.html](http://www.calstatela.edu/centers/hipic) CIS5200 Term Project Tutorial https://avatars2.githubusercontent.com/u/4156894?v=3&s=100

California State University, Los Angeles

#### Authors: [Taya Stewart](https://www.linkedin.com/in/taya-stewart-aaa74470/?lipi=urn%3Ali%3Apage%3Ad_flagship3_people_connections%3BKRT8FEYDRxqFzv6dzUu70Q%3D%3D&licu=urn%3Ali%3Acontrol%3Ad_flagship3_people_connections-connection_profile), [Raymond Delgado](https://www.linkedin.com/in/raymond-delgado-89a677184/?lipi=urn%3Ali%3Apage%3Ad_flagship3_people_connections%3BKRT8FEYDRxqFzv6dzUu70Q%3D%3D&licu=urn%3Ali%3Acontrol%3Ad_flagship3_people_connections-connection_profile), [Navaneeth Visagan](https://www.linkedin.com/in/navo-visagan/?lipi=urn%3Ali%3Apage%3Ad_flagship3_people_connections%3BKRT8FEYDRxqFzv6dzUu70Q%3D%3D&licu=urn%3Ali%3Acontrol%3Ad_flagship3_people_connections-connection_profile), [Frank Chen](https://www.linkedin.com/in/frank-chen-46678216b/), [Ninelia Talverdi](https://www.linkedin.com/in/neltalverdi/)

#### Instructor: [Jongwook Woo](https://www.linkedin.com/in/jongwook-woo-7081a85)

#### Date: 12/17/2020

**Lab Tutorial**

Taya Stewart ([tstewar2@calstatela.edu](mailto:yourname@calstatela.edu))

Raymond Delgado ([rdelga40@calstatela.edu](mailto:yourname@calstatela.edu))

Navaneet Visagan ([nvisaga@calstatela.edu](mailto:yourname@calstatela.edu))

Frank Chen ([fchen26@calstatela.edu](mailto:yourname@calstatela.edu))

Ninelia Talverdi ([ntalver2@calstatela.edu](mailto:yourname@calstatela.edu))

12/17/2020

**E-commerce Analysis of Consumer Behavior using Hive**

**Objectives**

Business data set and its customer review data is one of the popular areas for Big Data adoption. In this hands-on tutorial, you will learn how to use Big Data Compute Edition to:

* Upload and Download data file from the local system to Hadoop HDFS and vice versa
* Create tables and views in HDFS using HiveQL
* Create Hive queries to perform the analysis
* Use IBM Cognos Analytics, SAP Predictive Analytics, Tableau, Power BI for visualization

**Prerequisites**

Everything you need to go through the scripts and queries is already provisioned with the cluster. To analyze the data using BI tools, you need to have access to IBM Cognos Analytics, SAP Predictive Analytics, Tableau, Power BI.

**Platform Spec**

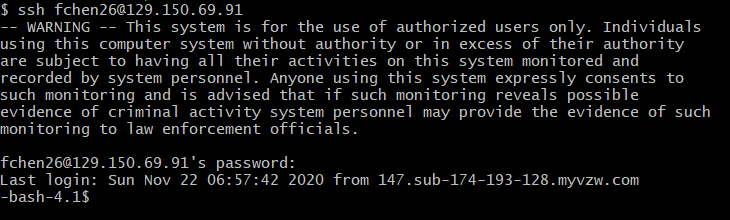
* Cluster Version – Oracle Big Data Compute Edition
* Number of Nodes – 3
* Memory size – 160 GB
* # of OCPUs – 8
* CPU speed – 2.20 GHz
* HDFS capacity – 802 GB
* Local Storage – 202 GB

Step 1: Connect to Oracle Cloud: Big Data Compute

You need to remotely access your Oracle Big Data that you executed in your Oracle Cloud account [putty](https://www.putty.org/) (mintty) or terminal (Mac/Linux, [Git Bash](https://git-scm.com/downloads)]) with ssh. For. For example, for the user name and ip address: **fchen26**, you need to run the following with the ip address given:

|  |
| --- |
| $ **ssh fchen26@129.150.69.91** |

When asked for a password, type in your user name again and press enter.



Now you are connected to Oracle cloud.

You may run the following HDFS commands to test if hdfs works well at your Oracle account.

|  |
| --- |
| $ **ls -al**  $ **hdfs dfs –ls** |

Step 2: Download Data from Amazon S3 and Load it into Oracle Big Data

Below is the location of the e-commerce data that is used for this project. You can download the data file (2019-Oct.zip) from Amazon S3:

|  |
| --- |
| $ wget -O 2019-Oct.csv.zip <https://groupthreebucket.s3-us-west-1.amazonaws.com/2019-Oct.csv.zip> |

**NOTE:** The dataset used for this project can be downloaded from [Kaggle](https://www.kaggle.com/mkechinov/ecommerce-behavior-data-from-multi-category-store). However, we only used the '2019-Oct.csv' data.

It’s better if you move your data file into the data folder, unzip it and then upload it into hdfs to avoid having space issues.

|  |
| --- |
| $ **mv 2019-Oct.csv.zip /data/**  $ **cd /data/** |

|  |
| --- |
| $ **unzip 2019-Oct.csv.zip** |

Now you need to upload the “2019-Oct.csv” file to a directory of HDFS. Run the following commands in order:

Create a directory named **ecommerce**.

|  |
| --- |
| $ **hdfs dfs -mkdir ecommerce**  $ **hdfs dfs -ls** |

Put **2019-Oct.csv** file from home directory to **ecommerce** directory.

|  |
| --- |
| $ **hdfs dfs -put 2019-Oct.csv /user/fchen26/ecommerce/** |

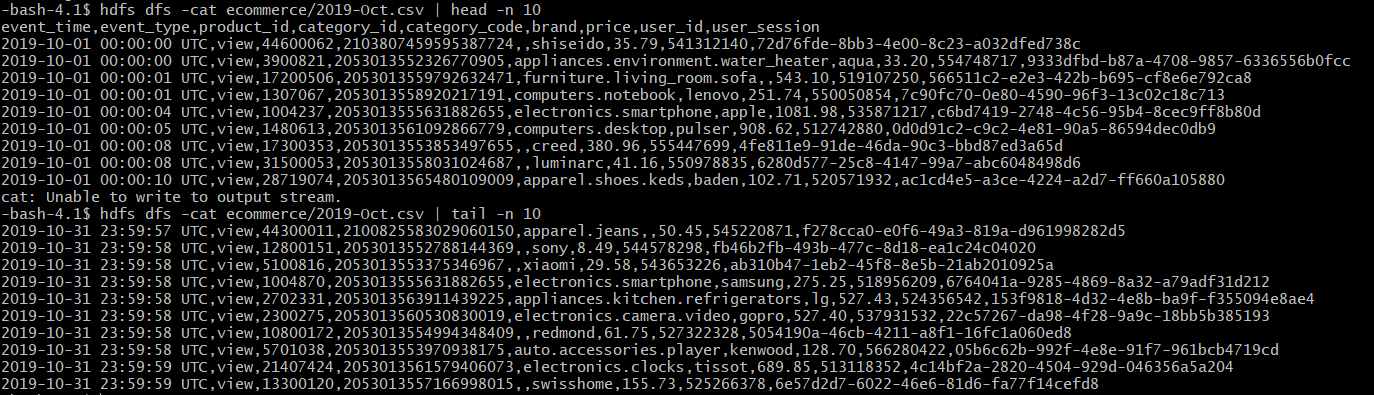
To check the file is uploaded successfully, run the below command.

|  |
| --- |
| $ **hdfs dfs -ls ecommerce** |



Run the below commands to get the first and last 10 lines of your data file:

|  |
| --- |
| $ **hdfs dfs -cat ecommerce/2019-Oct.csv | head -n 10**  $ **hdfs dfs -cat ecommerce/2019-Oct.csv | tail -n 10** |



Run the following HDFS command to make your beeline command works.

**NOTE:** There is a period at the end of the command in the below:

|  |
| --- |
| $ **hdfs dfs -chmod -R o+w .** |

Step 3: Creating Hive Tables to Query Data

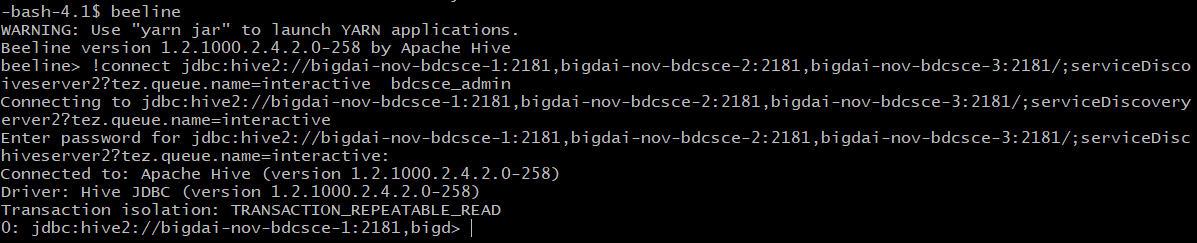
## The following Hive statement creates an external table that allows Hive to query data stored in HDFS. External tables preserve the data in the original file format, while allowing Hive to perform queries against the data within the file.

## Open another terminal and login into your account using *ssh* as in Step 1.

## Open beeline CLI (Command Line Shell Interface) that is equivalent to hive CLI environment as follows. Beeline is for multiple users’ access to Hive Server 2 of a Hadoop cluster. Press enter without putting any password when it asks for a password.

|  |
| --- |
| **beeline**  WARNING: Use "yarn jar" to launch YARN applications.  Beeline version 1.2.1000.2.4.2.0-258 by Apache Hive  beeline> **!connect jdbc:hive2://bigdai-nov-bdcsce-1:2181,bigdai-nov-bdcsce-2:2181,bigdai-nov-bdcsce-3:2181/;serviceDiscoveryMode=zooKeeper;zooKeeperNamespace=hiveserver2?tez.queue.name=interactive bdcsce\_admin**  Connecting to jdbc:hive2://bigdai-nov-bdcsce-1:2181,bigdai-nov-bdcsce-2:2181,bigdai-nov-bdcsce-3:2181/;serviceDiscoveryerver2?tez.queue.name=interactive  **Enter password** for jdbc:hive2://bigdai-nov-bdcsce-1:2181,bigdai-nov-bdcsce-2:2181,bigdai-nov-bdcsce-3:2181/;serviceDischiveserver2?tez.queue.name=interactive:  Connected to: Apache Hive (version 1.2.1000.2.4.2.0-258)  Driver: Hive JDBC (version 1.2.1000.2.4.2.0-258)  Transaction isolation: TRANSACTION\_REPEATABLE\_READ  0: jdbc:hive2://bigdai-nov-bdcsce-1:2181,bigd> |

**Note:** If you see “**CLOSED**” in the above beeline shell prompt, it is **not** connected to Hive Server2.



Now you have to create your database with your username to separate your tables with other users. For example, the user **groupthree** should run the following:

**Note:** Make sure to use your **username**.

|  |
| --- |
| **CREATE DATABASE IF NOT EXISTS groupthree;** |

Run the below command to make sure your database is created.

|  |
| --- |
| **SHOW databases;** |

Now you need to use your database to create tables.

|  |
| --- |
| **USE groupthree;** |

**Note:** Make sure to replace the database name with your **username**.

In the beeline shell CLI, you need to copy and paste the following HiveQL code to create an external table “ecommerce”.

**Note:** Don’t forget to replace **fchen26** to your account name in the following HiveQL code.

|  |
| --- |
| **DROP TABLE IF EXISTS ecommerce;**  **--create the ecommerce table on comma-separated data**  **CREATE EXTERNAL TABLE IF NOT EXISTS ecommerce (**  **event\_time STRING,**  **event\_type STRING,**  **product\_id INT,**  **category\_id BIGINT,**  **category\_code STRING,**  **brand STRING,**  **price FLOAT,**  **user\_id INT,**  **user\_session STRING**  **)**  **ROW FORMAT DELIMITED**  **FIELDS TERMINATED BY ','**  **LOCATION '/user/fchen26/ecommerce'**  **TBLPROPERTIES ('skip.header.line.count'='1');** |

Then, in the beeline shell, you need to check if the table “**ecommerce**”is shown:

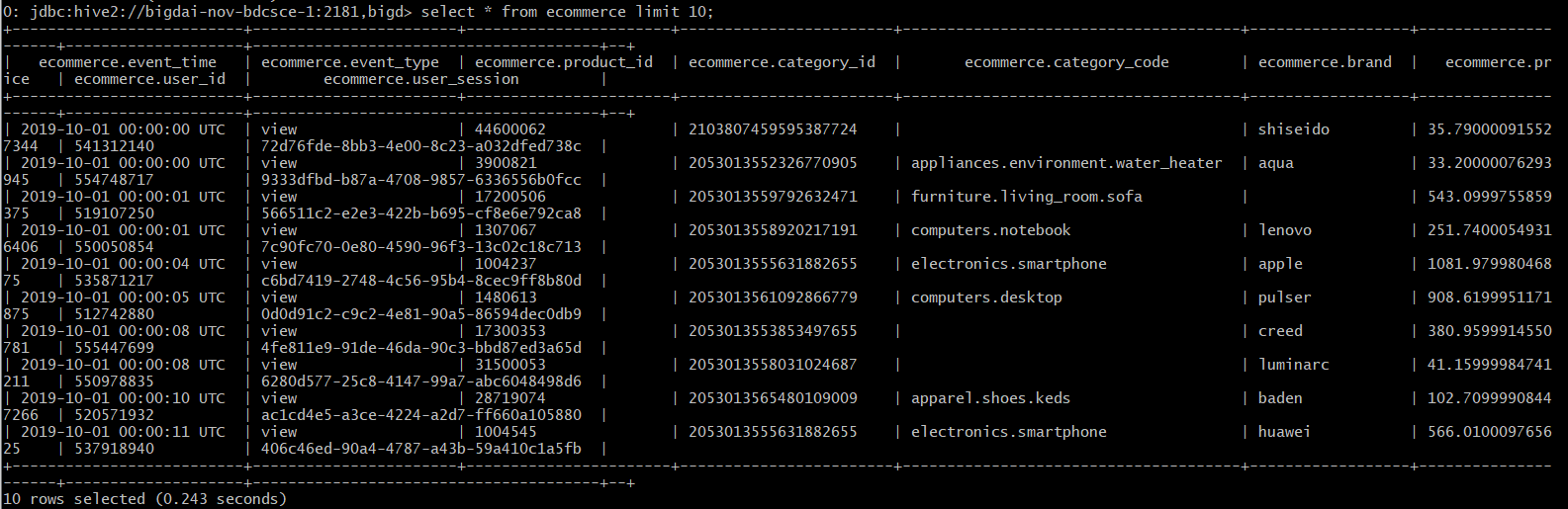
|  |
| --- |
| **SHOW tables;** |

**Note:** If you can’t see the table name, then the table is not created and you have to follow the same step again.

Now you can query the content of the **ecommerce** table to see if it has the correct data and values:

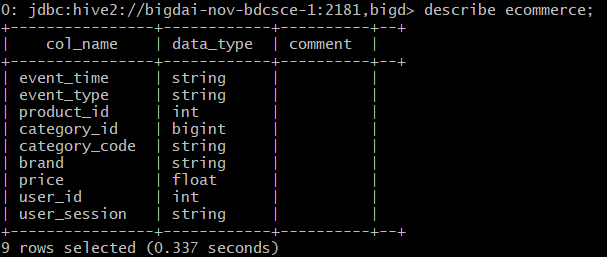
|  |
| --- |
| **SELECT \* FROM ecommerce LIMIT 10;** |

You will see a result similar to below:



You can see the structure of the table as well:

|  |
| --- |
| **DESCRIBE ecommerce;** |



Next, in the beeline shell CLI, you need to copy and paste the following HiveQL code to create a table “week\_days” and insert data into it.

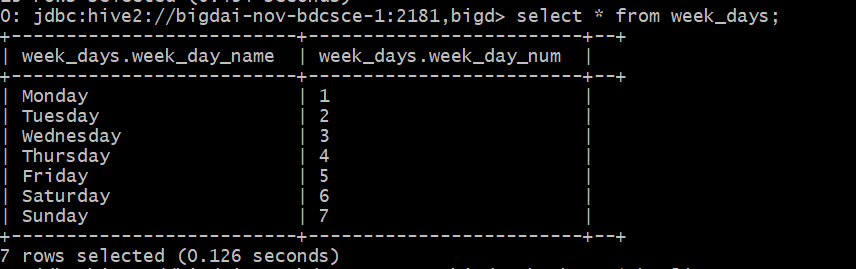
This table will hold the information regarding the days of the week. We will use this table and our ecommerce\_view (discussed later) to create our queries.

|  |
| --- |
| **DROP TABLE IF EXISTS week\_days;**  **--create the week\_days table on comma-separated data**  **CREATE TABLE IF NOT EXISTS week\_days (week\_day\_name STRING, week\_day\_num STRING)**  **ROW FORMAT DELIMITED**  **FIELDS TERMINATED BY ',';**  **INSERT INTO TABLE week\_days**  **VALUES ('Monday', '1'), ('Tuesday', '2'), ('Wednesday', '3'), ('Thursday', '4'), ('Friday', '5'), ('Saturday','6'),('Sunday','7');** |

You can run this query to make sure that your insert statement was executed successfully.

|  |
| --- |
| **SELECT \* FROM week\_days;** |

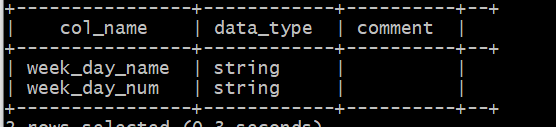
The output should look like this:



Run this query to ensure that your **week\_days** table have the right structure

|  |
| --- |
| **DESCRIBE week\_days;** |

The schema of your table should look like this:



Now, in the beeline shell CLI, you need to copy and paste the following HiveQL code to create a view “**ecommerce\_view**”.

This view will have all the data from the ecommerce table. All the subsequent queries will be based on this view and the week\_days table.

|  |
| --- |
| **DROP VIEW IF EXISTS ecommerce\_view;**  **--create the ecommerce\_view view on comma-separated data**  **CREATE VIEW IF NOT EXISTS ecommerce\_view AS**  **SELECT**  **event\_time,**  **date\_format(event\_time, 'u') as week\_day\_num,**  **day(event\_time) as day,**  **hour(event\_time) as hour,**  **unix\_timestamp(event\_time) as event\_time\_in\_seconds,**  **event\_type,**  **product\_id,**  **category\_id,**  **split(category\_code, '\\.')[0] as primary\_category,**  **brand,**  **price,**  **user\_id,**  **user\_session**  **FROM ecommerce**  **WHERE category\_code!='' AND brand!='';** |

**Note:** We will use the value for column **week\_day\_num** to join this view with the week\_day table.

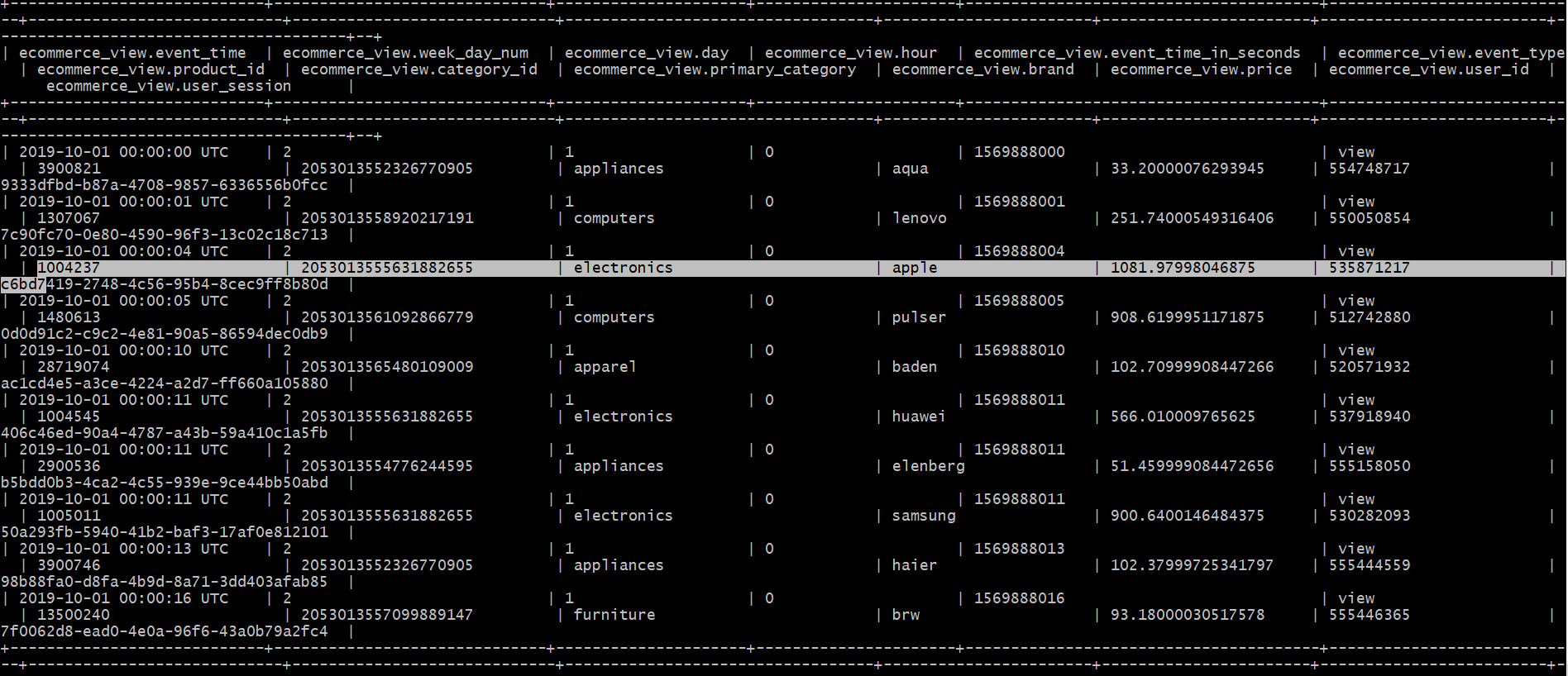
**Note:** The value of the category code consists of multiple parts that are separated by period (.). You will use the “split” method to get the first category.

**Note:** The “Where” clause that is used in the above query is for cleaning any junk data.

Now you can query the content of the **ecommerce\_view** view to see if it has the correct data and values:

|  |
| --- |
| **SELECT \* FROM ecommerce\_view LIMIT 10;** |

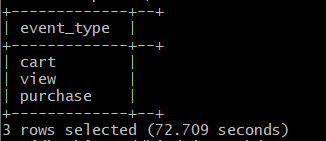
You will see a result similar to below:



Run the below query to see the event types:

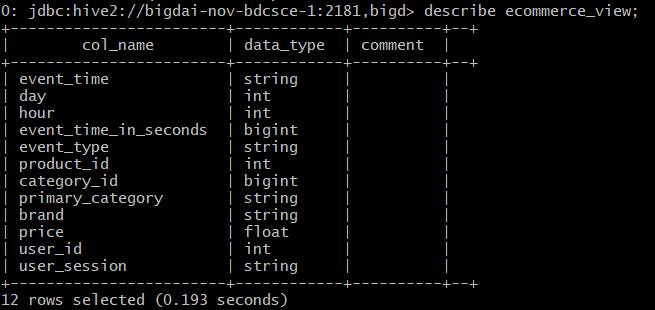
|  |
| --- |
| **SELECT event\_type FROM ecommerce\_view GROUP BY event\_type;** |

You will see a result similar to below:



You can also look over the structure of the view as well:

|  |
| --- |
| **DESCRIBE ecommerce\_view;** |



Step 4: Creating Hive Queries to Analyze data

## The following Hive queries will be used to analyze the data.

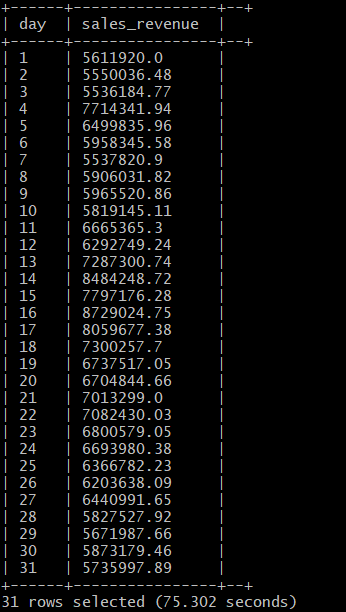
## Analysis 1:

**Step1:** Creating a table and running a query to get the sales revenue by day.

First, run the below query:

|  |
| --- |
| **SELECT day, ROUND(SUM(price), 2) AS sales\_revenue FROM ecommerce\_view WHERE event\_type='purchase' GROUP BY day ORDER BY day ASC;** |

You will see a result similar to below:



The below query will create a table using the above query and store the results in hdfs for visualization:

|  |
| --- |
| **DROP TABLE IF EXISTS sales\_revenue\_by\_day;**  **CREATE TABLE sales\_revenue\_by\_day**  **ROW FORMAT DELIMITED FIELDS TERMINATED BY ','**  **STORED AS TEXTFILE LOCATION '/user/fchen26/ecommerce/one'**  **AS**  **select day, ROUND(SUM(price), 2) AS sales\_revenue FROM ecommerce\_view WHERE event\_type='purchase' GROUP BY day ORDER BY day ASC;** |

Then, you need to check if the table is created successfully or not:

|  |
| --- |
| **SHOW tables;** |

**Step 2:** After the Hive tables are created, you can download it to your lab (or personal PC/Laptop) as follows.

Open another terminal with git bash, minty, or putty, which is to connect the Oracle cloud to download the output file 000000\_0 at the HDFS path “/user/**fchen26**/ecommerce/**one** ”.

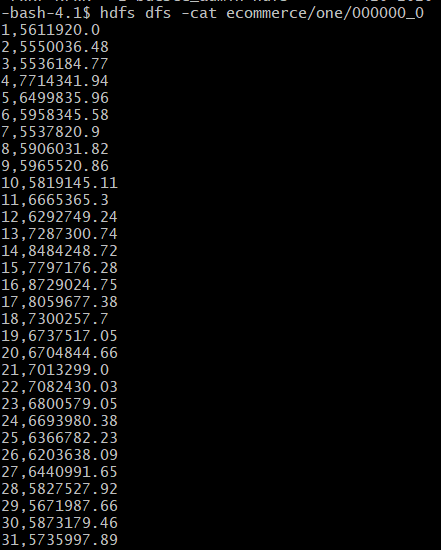
First, run the following hdfs command to list what file exists at “/user/**fchen26**/ecommerce/**one**” directory that is actually the location of Hive table. It is a file named “000000\_0”:

**Note:** Do not forget to change the username.

|  |
| --- |
| $ **hdfs dfs -ls ecommerce/one** |

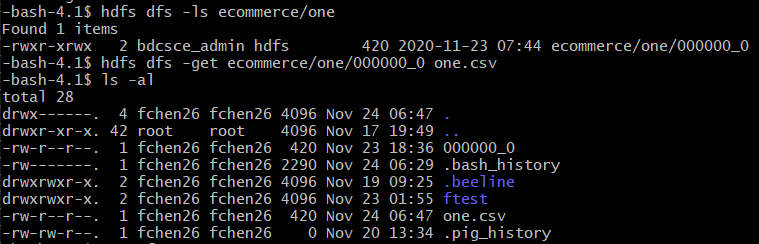
You can view the contents of the file with the below command:

|  |
| --- |
| $ **hdfs dfs -cat ecommerce/one/000000\_0** |



Then, download the file to the local file systems:

|  |
| --- |
| $ **hdfs dfs -get ecommerce/one/000000\_0 one.csv**  $ **ls -al** |



Lastly, open another terminal with git bash, minty, or putty in order to read/import the output file using your lab computer (or your PC/Laptop) - you have to download the file to your lab computer (or your PC/Laptop). For example, your output file at the oracle cloud server is located at /home/**fchen26**/one.csv and remotely copied to the file “one.csv”.

|  |
| --- |
| $ **scp fchen26@ipaddress:/home/fchen26/one.csv one.csv** |

**Note:** Make sure to replace the database name with your **username** and ip address.

Enter your password to download the file.

**Alternatively**, for Windows users, you may use psftp to download the file. You need to download it at <https://the.earth.li/~sgtatham/putty/latest/w64/psftp.exe>.

For example, in order to download 'one.csv’ from the server of Oracle Cloud, you have to run psftp as follows. You may read through the commands of *psftp* in the below:

|  |
| --- |
| psftp> **open [ipaddress]**  Login as: **fchen26**  Enter password…  psftp> **ls**  *Listing directory /home/fchen26*  psftp> **get one.csv** |

**Note:** Do not forget to change the username and put the appropriate ip address.

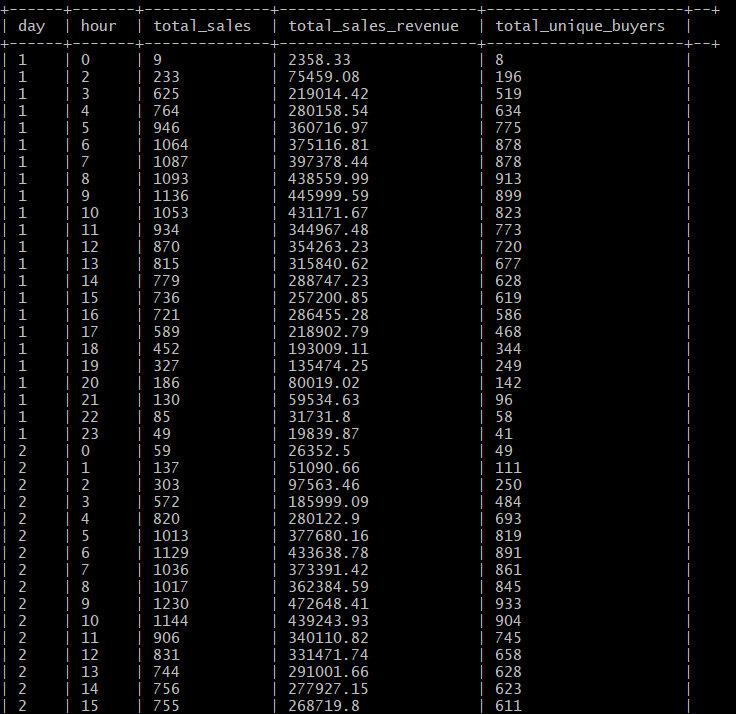
## Analysis 2:

**Step1:** Creating a table and running a query to get the total number of purchases by day and hour.

First, run the below query:

|  |
| --- |
| **SELECT day, hour ,COUNT(event\_type) AS total\_sales, ROUND(SUM(price), 2) AS total\_sales\_revenue, COUNT(DISTINCT user\_id) AS total\_unique\_buyers FROM ecommerce\_view WHERE event\_type='purchase' GROUP BY day, hour ORDER BY day ASC, hour ASC;** |

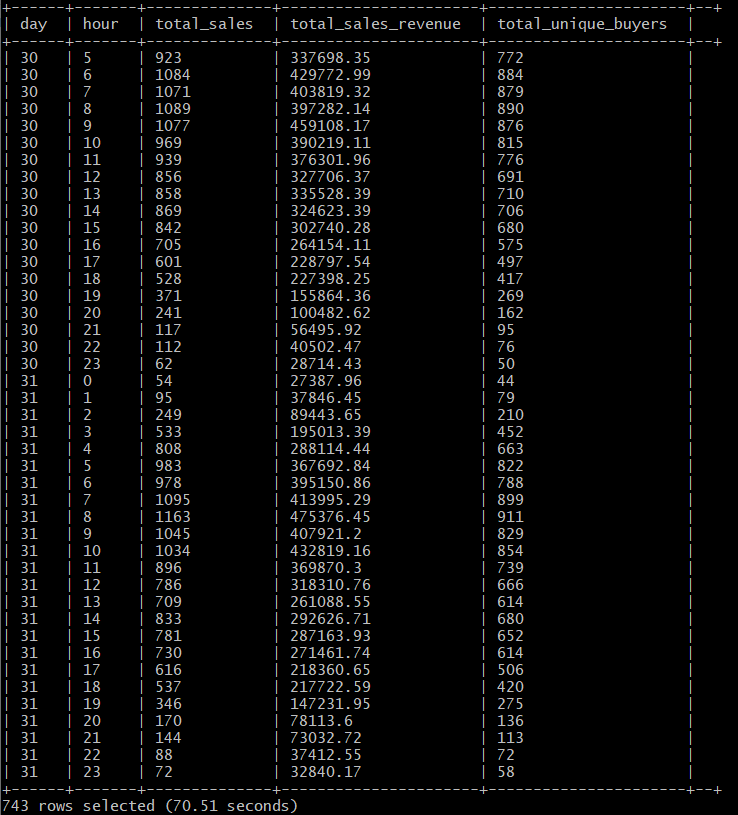
You will see a result similar to below:



.

.

.



The below query will create a table using the above query and store the results in hdfs for visualization:

|  |
| --- |
| **DROP TABLE IF EXISTS total\_purchases\_by\_day\_hour;**  **CREATE TABLE total\_purchases\_by\_day\_hour**  **ROW FORMAT DELIMITED FIELDS TERMINATED BY ','**  **STORED AS TEXTFILE LOCATION '/user/fchen26/ecommerce/two'**  **AS**  **SELECT day, hour ,COUNT(event\_type) AS total\_sales, ROUND(SUM(price), 2) AS total\_sales\_revenue, COUNT(DISTINCT user\_id) as total\_unique\_buyers FROM ecommerce\_view WHERE event\_type='purchase' GROUP BY day, hour ORDER BY day ASC, hour ASC;** |

Then, you need to check if the table is created successfully or not:

|  |
| --- |
| **SHOW tables;** |

**Step 2:** After the Hive tables are created, you can download it to your lab (or personal PC/Laptop) as follows.

Open another terminal with git bash, minty, or putty, which is to connect the Oracle cloud to download the output file 000000\_0 at the HDFS path “/user/**fchen26**/ecommerce/**two** ”.

First, run the following hdfs command to list what file exists at “/user/**fchen26**/ecommerce/**two**” directory that is actually the location of Hive table. It is a file named “000000\_0”:

**Note:** Do not forget to change the username.

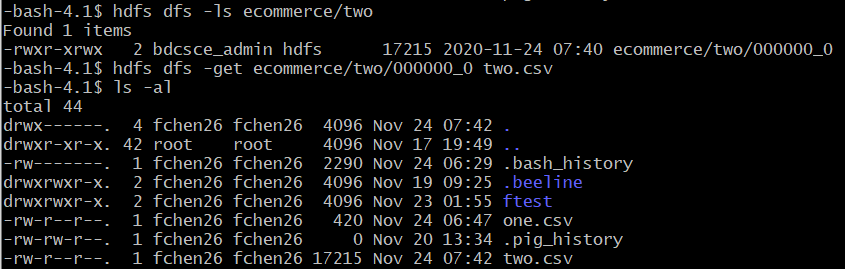
|  |
| --- |
| $ **hdfs dfs -ls ecommerce/two** |

You can view the contents of the file with the below command:

|  |
| --- |
| $ **hdfs dfs -cat ecommerce/two/000000\_0** |

Then, download the file to the local file systems:

|  |
| --- |
| $ **hdfs dfs -get ecommerce/two/000000\_0 two.csv**  $ **ls -al** |



Lastly, open another terminal with git bash, minty, or putty in order to read/import the output file using your lab computer (or your PC/Laptop) - you have to download the file to your lab computer (or your PC/Laptop). For example, your output file at the oracle cloud server is located at /home/**fchen26**/two.csv and remotely copied to the file “two.csv”.

|  |
| --- |
| $ **scp fchen26@ipaddress:/home/fchen26/two.csv two.csv** |

**Note:** Make sure to replace the database name with your **username** and put the appropriate ip address.

Enter your password to download the file.

**Alternatively**, for Windows users, you may use psftp to download the file. You need to download it at <https://the.earth.li/~sgtatham/putty/latest/w64/psftp.exe>.

For example, in order to download 'two.csv’ from the server of Oracle Cloud, you have to run psftp as follows. You may read through the commands of *psftp* in the below:

|  |
| --- |
| psftp> **open [ipaddress]**  Login as: **fchen26**  Enter password…  psftp> **ls**  *Listing directory /home/fchen26*  psftp> **get two.csv** |

**Note:** Do not forget to change the **username** and put the appropriate ip address.

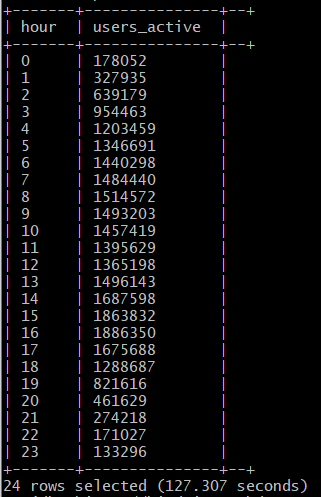
## Analysis 3:

**Step1:** Creating a table and running a query to get the total number of active users by hour.

First, run the below query:

|  |
| --- |
| **SELECT hour , COUNT(user\_id) AS users\_active FROM ecommerce\_view GROUP BY hour ORDER BY hour ASC;** |

You will see a result similar to below:



The below query will create a table using the above query and store the results in hdfs for visualization:

|  |
| --- |
| **DROP TABLE IF EXISTS active\_users\_by\_hour;**  **CREATE TABLE active\_users\_by\_hour**  **ROW FORMAT DELIMITED FIELDS TERMINATED BY ','**  **STORED AS TEXTFILE LOCATION '/user/fchen26/ecommerce/three'**  **AS**  **SELECT hour , COUNT(user\_id) AS users\_active FROM ecommerce\_view GROUP BY hour ORDER BY hour ASC;** |

Then, you need to check if the table is created successfully or not:

|  |
| --- |
| **SHOW tables;** |

**Step 2:** After the Hive tables are created, you can download it to your lab (or personal PC/Laptop) as follows.

Open another terminal with git bash, minty, or putty, which is to connect the Oracle cloud to download the output file 000000\_0 at the HDFS path “/user/**fchen26**/ecommerce/**three** ”.

First, run the following hdfs command to list what file exists at “/user/**fchen26**/ecommerce/**three**” directory that is actually the location of Hive table. It is a file named “000000\_0”:

**Note:** Do not forget to change the username.

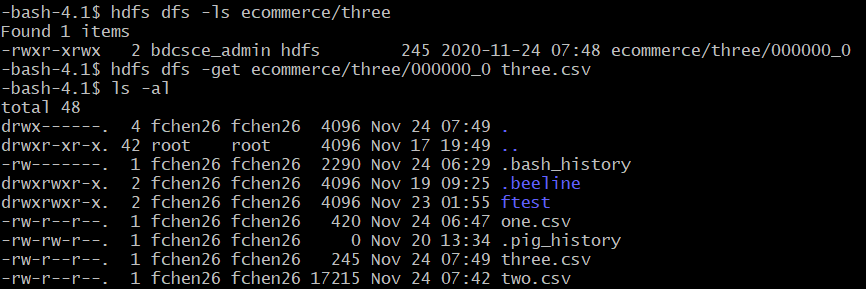
|  |
| --- |
| $ **hdfs dfs -ls ecommerce/three** |

You can view the contents of the file with the below command:

|  |
| --- |
| $ **hdfs dfs -cat ecommerce/three/000000\_0** |

Then, download the file to the local file systems:

|  |
| --- |
| $ **hdfs dfs -get ecommerce/three/000000\_0 three.csv**  $ **ls -al** |



Lastly, open another terminal with git bash, minty, or putty in order to read/import the output file using your lab computer (or your PC/Laptop) - you have to download the file to your lab computer (or your PC/Laptop). For example, your output file at the oracle cloud server is located at /home/**fchen26**/three.csv and remotely copied to the file “three.csv”.

|  |
| --- |
| $ **scp fchen26@ipaddress:/home/fchen26/three.csv three.csv** |

**Note:** Make sure to replace the database name with your **username** and put the appropriate ip address.

Enter your password to download the file.

**Alternatively**, for Windows users, you may use psftp to download the file. You need to download it at <https://the.earth.li/~sgtatham/putty/latest/w64/psftp.exe>.

For example, in order to download ‘three.csv’ from the server of Oracle Cloud, you have to run psftp as follows. You may read through the commands of *psftp* in the below:

|  |
| --- |
| psftp> **open [ipaddress]**  Login as: **fchen26**  Enter password…  psftp> **ls**  *Listing directory /home/fchen26*  psftp> **get three.csv** |

**Note:** Do not forget to change the **username** and put the appropriate ip address.

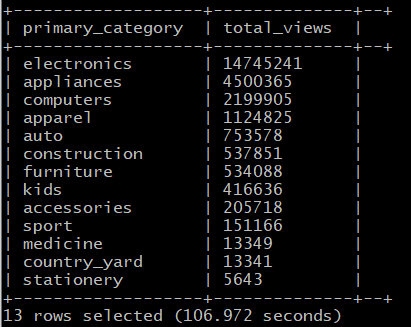
## Analysis 4:

**Step1:** Creating a table and running a query to get the most viewed product categories.

First, run the below query:

|  |
| --- |
| **SELECT primary\_category, COUNT(event\_type) AS total\_views FROM ecommerce\_view WHERE event\_type = 'view' GROUP BY primary\_category ORDER BY total\_views DESC;** |

You will see a result similar to below:



The below query will create a table using the above query and store the results in hdfs for visualization:

|  |
| --- |
| **DROP TABLE IF EXISTS most\_viewed\_categories;**  **CREATE TABLE most\_viewed\_categories**  **ROW FORMAT DELIMITED FIELDS TERMINATED BY ','**  **STORED AS TEXTFILE LOCATION '/user/fchen26/ecommerce/four'**  **AS**  **SELECT primary\_category, COUNT(event\_type) AS total\_views FROM ecommerce\_view WHERE event\_type = 'view' GROUP BY primary\_category ORDER BY total\_views DESC;** |

Then, you need to check if the table is created successfully or not:

|  |
| --- |
| **show tables;** |

**Step 2:** After the Hive tables are created, you can download it to your lab (or personal PC/Laptop) as follows.

Open another terminal with git bash, minty, or putty, which is to connect the Oracle cloud to download the output file 000000\_0 at the HDFS path “/user/**fchen26**/ecommerce/**four** ”.

First, run the following hdfs command to list what file exists at “/user/**fchen26**/ecommerce/**four**” directory that is actually the location of Hive table. It is a file named “000000\_0”:

**Note:** Do not forget to change the username.

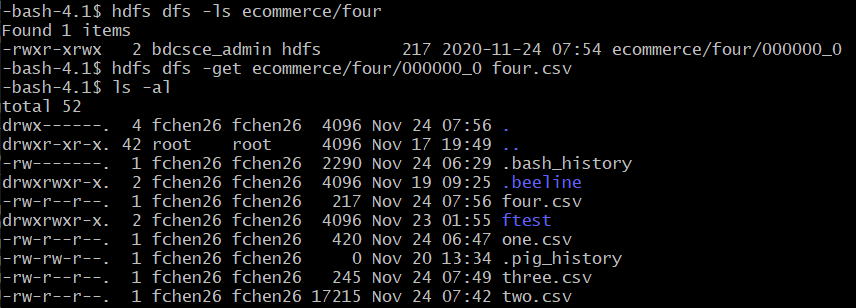
|  |
| --- |
| $ **hdfs dfs -ls ecommerce/four** |

You can view the contents of the file with the below command:

|  |
| --- |
| $ **hdfs dfs -cat ecommerce/four/000000\_0** |

Then, download the file to the local file systems:

|  |
| --- |
| $ **hdfs dfs -get ecommerce/four/000000\_0 four.csv**  $ **ls -al** |



Lastly, open another terminal with git bash, minty, or putty in order to read/import the output file using your lab computer (or your PC/Laptop) - you have to download the file to your lab computer (or your PC/Laptop). For example, your output file at the oracle cloud server is located at /home/**fchen26**/four.csv and remotely copied to the file “four.csv”.

|  |
| --- |
| $ **scp fchen26@ipaddress:/home/fchen26/four.csv four.csv** |

**Note:** Make sure to replace the database name with your **username** and put the appropriate ip address.

Enter your password to download the file.

**Alternatively**, for Windows users, you may use psftp to download the file. You need to download it at <https://the.earth.li/~sgtatham/putty/latest/w64/psftp.exe>.

For example, in order to download ‘four.csv’ from the server of Oracle Cloud, you have to run psftp as follows. You may read through the commands of *psftp* in the below:

|  |
| --- |
| psftp> **open [ipaddress]**  Login as: **fchen26**  Enter password…  psftp> **ls**  *Listing directory /home/fchen26*  psftp> **get four.csv** |

**Note:** Do not forget to change the **username** and put the appropriate ip address.

## Analysis 5:

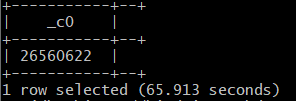
**Step1:** Creating a table and running a query to get the percentage of each event.

First, run the below query:

|  |
| --- |
| **SELECT count(event\_type) FROM ecommerce\_view;** |

This will give you the total count of all event types.

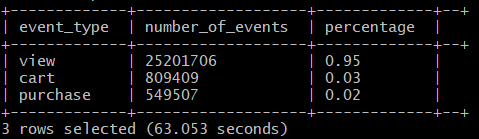
You will see a result similar to below:



Save the result and use it in the below query:

|  |
| --- |
| **SELECT event\_type,count(event\_type) AS number\_of\_events, ROUND(CAST(COUNT(event\_type) AS float)/26560622, 2) AS percentage FROM ecommerce\_view GROUP BY event\_type ORDER BY percentage DESC;** |

You will see a result similar to below:



The below query will create a table using the above query and store the results in hdfs for visualization:

|  |
| --- |
| **DROP TABLE IF EXISTS percentage\_events;**  **CREATE TABLE percentage\_events**  **ROW FORMAT DELIMITED FIELDS TERMINATED BY ','**  **STORED AS TEXTFILE LOCATION '/user/fchen26/ecommerce/five'**  **AS**  **SELECT event\_type,count(event\_type) AS number\_of\_events, ROUND(CAST(COUNT(event\_type) AS float)/26560622, 2) AS percentage FROM ecommerce\_view GROUP BY event\_type ORDER BY percentage DESC;** |

Then, you need to check if the table is created successfully or not:

|  |
| --- |
| **SHOW tables;** |

**Step 2:** After the Hive tables are created, you can download it to your lab (or personal PC/Laptop) as follows.

Open another terminal with git bash, minty, or putty, which is to connect the Oracle cloud to download the output file 000000\_0 at the HDFS path “/user/**fchen26**/ecommerce/**five** ”.

First, run the following hdfs command to list what file exists at “/user/**fchen26**/ecommerce/**five**” directory that is actually the location of Hive table. It is a file named “000000\_0”:

**Note:** Do not forget to change the username.

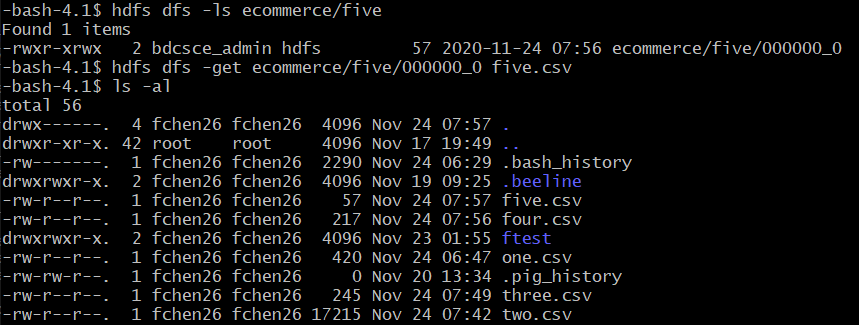
|  |
| --- |
| $ **hdfs dfs -ls ecommerce/five** |

You can view the contents of the file with the below command:

|  |
| --- |
| $ **hdfs dfs -cat ecommerce/five/000000\_0** |

Then, download the file to the local file systems:

|  |
| --- |
| $ **hdfs dfs -get ecommerce/five/000000\_0 five.csv**  $ **ls -al** |



Lastly, open another terminal with git bash, minty, or putty in order to read/import the output file using your lab computer (or your PC/Laptop) - you have to download the file to your lab computer (or your PC/Laptop). For example, your output file at the oracle cloud server is located at /home/**fchen26**/five.csv and remotely copied to the file “five.csv”.

|  |
| --- |
| $ **scp fchen26@ipaddress:/home/fchen26/five.csv five.csv** |

**Note:** Make sure to replace the database name with your **username** and put the appropriate ip address.

Enter your password to download the file.

**Alternatively**, for Windows users, you may use psftp to download the file. You need to download it at <https://the.earth.li/~sgtatham/putty/latest/w64/psftp.exe>.

For example, in order to download ‘five.csv’ from the server of Oracle Cloud, you have to run psftp as follows. You may read through the commands of *psftp* in the below:

|  |
| --- |
| psftp> **open [ipaddress]**  Login as: **fchen26**  Enter password…  psftp> **ls**  *Listing directory /home/fchen26*  psftp> **get five.csv** |

**Note:** Do not forget to change the **username** and put the appropriate ip address.

## Analysis 6:

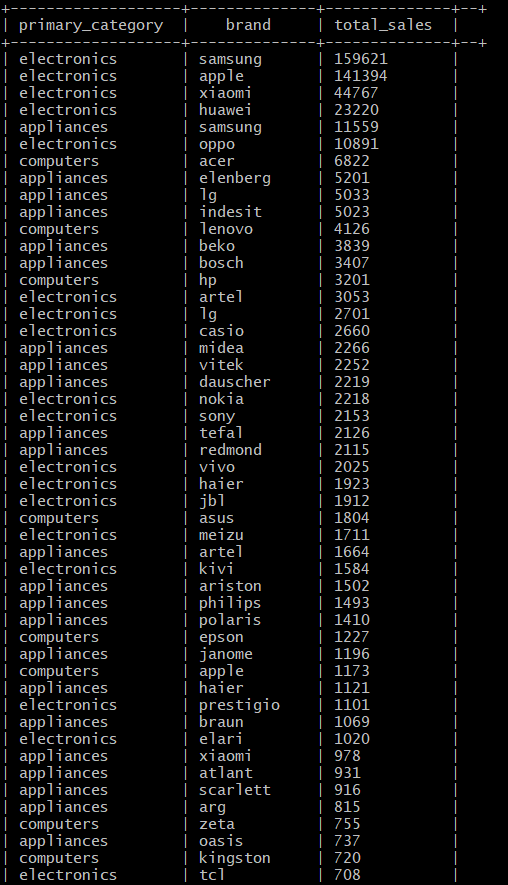
**Step1:** Creating a table and running a query to get the top 100 primary category, brand and total sales by product category and brand.

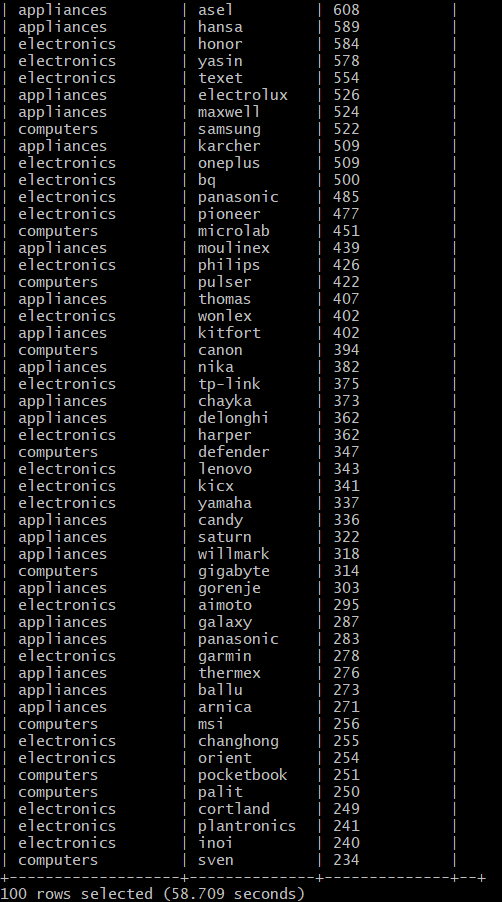
First, run the below query:

|  |
| --- |
| **SELECT primary\_category, brand, COUNT(event\_type) as total\_sales FROM ecommerce\_view WHERE primary\_category IN('electronics', 'appliances', 'computers') AND event\_type ='purchase' GROUP BY primary\_category, brand ORDER BY total\_sales DESC LIMIT 100;** |

**Note:** That scope of the query is the top 3 most viewed product categories.

You will see a result similar to below:





The below query will create a table using the above query and store the results in hdfs for visualization:

|  |
| --- |
| **DROP TABLE IF EXISTS top\_primary\_categories;**  **CREATE TABLE top\_primary\_categories**  **ROW FORMAT DELIMITED FIELDS TERMINATED BY ','**  **STORED AS TEXTFILE LOCATION '/user/fchen26/ecommerce/six'**  **AS**  **SELECT primary\_category, brand, COUNT(event\_type) as total\_sales FROM ecommerce\_view WHERE primary\_category IN('electronics', 'appliances', 'computers') AND event\_type ='purchase' GROUP BY primary\_category, brand ORDER BY total\_sales DESC LIMIT 100;** |

Then, you need to check if the table is created successfully or not:

|  |
| --- |
| **SHOW tables;** |

**Step 2:** After the Hive tables are created, you can download it to your lab (or personal PC/Laptop) as follows.

Open another terminal with git bash, minty, or putty, which is to connect the Oracle cloud to download the output file 000000\_0 at the HDFS path “/user/**fchen26**/ecommerce/**six** ”.

First, run the following hdfs command to list what file exists at “/user/**fchen26**/ecommerce/**six**” directory that is actually the location of Hive table. It is a file named “000000\_0”:

**Note:** Do not forget to change the username.

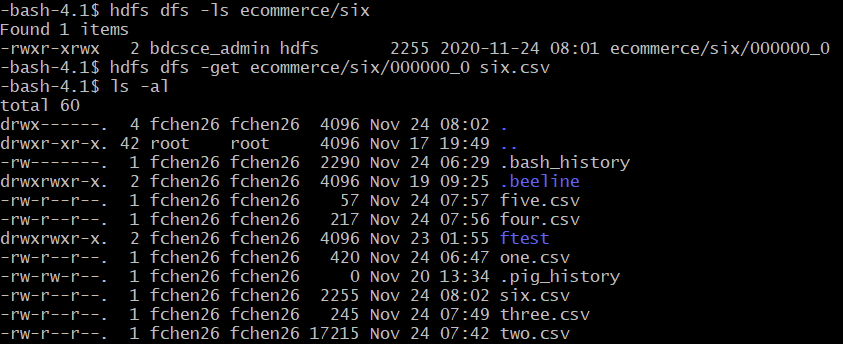
|  |
| --- |
| $ **hdfs dfs -ls ecommerce/six** |

You can view the contents of the file with the below command:

|  |
| --- |
| $ **hdfs dfs -cat ecommerce/six/000000\_0** |

Then, download the file to the local file systems:

|  |
| --- |
| $ **hdfs dfs -get ecommerce/six/000000\_0 six.csv**  $ **ls -al** |



Lastly, open another terminal with git bash, minty, or putty in order to read/import the output file using your lab computer (or your PC/Laptop) - you have to download the file to your lab computer (or your PC/Laptop). For example, your output file at the oracle cloud server is located at /home/**fchen26**/six.csv and remotely copied to the file “six.csv”.

|  |
| --- |
| $ **scp fchen26@ipaddress:/home/fchen26/six.csv six.csv** |

**Note:** Make sure to replace the database name with your **username** and put the appropriate ip address.

Enter your password to download the file.

**Alternatively**, for Windows users, you may use psftp to download the file. You need to download it at <https://the.earth.li/~sgtatham/putty/latest/w64/psftp.exe>.

For example, in order to download ‘six.csv’ from the server of Oracle Cloud, you have to run psftp as follows. You may read through the commands of *psftp* in the below:

|  |
| --- |
| psftp> **open [ipaddress]**  Login as: **fchen26**  Enter password…  psftp> **ls**  *Listing directory /home/fchen26*  psftp> **get six.csv** |

**Note:** Do not forget to change the **username** and put the appropriate ip address.

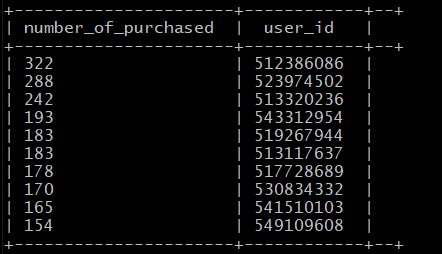
## Analysis 7:

**Step1:** Creating a table and running a query to get the top 10 users who purchased more than once.

First, run the below query:

|  |
| --- |
| **SELECT COUNT(user\_id) as number\_of\_purchased,user\_id FROM ecommerce\_view WHERE event\_type ='purchase' GROUP BY user\_id HAVING number\_of\_purchased>1 ORDER BY number\_of\_purchased DESC limit 10;** |

You will see a result similar to below:



The below query will create a table using the above query and store the results in hdfs for visualization:

|  |
| --- |
| **DROP TABLE IF EXISTS top\_users\_most\_purchased;**  **CREATE TABLE top\_users\_most\_purchased**  **ROW FORMAT DELIMITED FIELDS TERMINATED BY ','**  **STORED AS TEXTFILE LOCATION '/user/fchen26/ecommerce/seven'**  **AS**  **SELECT COUNT(user\_id) as number\_of\_purchased,user\_id FROM ecommerce\_view WHERE event\_type ='purchase' GROUP BY user\_id HAVING number\_of\_purchased>1 ORDER BY number\_of\_purchased DESC limit 10;** |

Then, you need to check if the table is created successfully or not:

|  |
| --- |
| **SHOW tables;** |

**Step 2:** After the Hive tables are created, you can download it to your lab (or personal PC/Laptop) as follows.

Open another terminal with git bash, minty, or putty, which is to connect the Oracle cloud to download the output file 000000\_0 at the HDFS path “/user/**fchen26**/ecommerce/**seven** ”.

First, run the following hdfs command to list what file exists at “/user/**fchen26**/ecommerce/**seven**” directory that is actually the location of Hive table. It is a file named “000000\_0”:

**Note:** Do not forget to change the username.

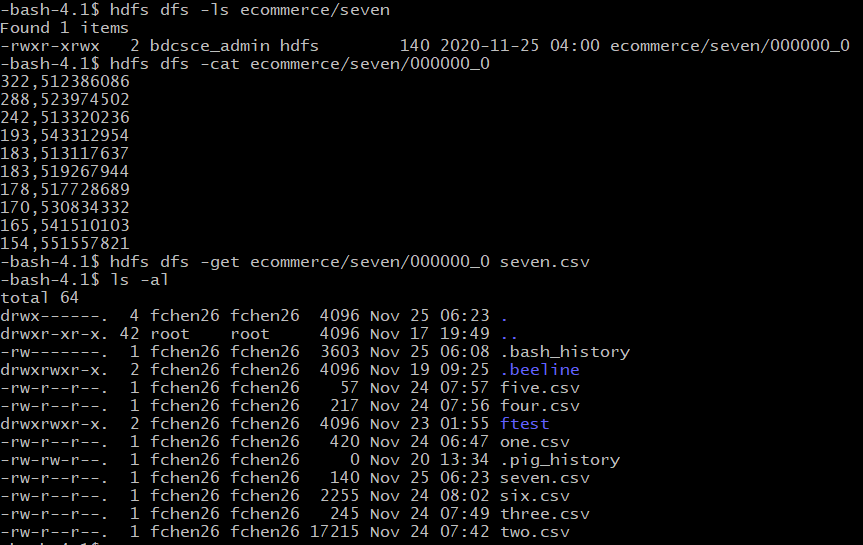
|  |
| --- |
| $ **hdfs dfs -ls ecommerce/seven** |

You can view the contents of the file with the below command:

|  |
| --- |
| $ **hdfs dfs -cat ecommerce/seven/000000\_0** |

Then, download the file to the local file systems:

|  |
| --- |
| $ **hdfs dfs -get ecommerce/seven/000000\_0 seven.csv**  $ **ls -al** |



Lastly, open another terminal with git bash, minty, or putty in order to read/import the output file using your lab computer (or your PC/Laptop) - you have to download the file to your lab computer (or your PC/Laptop). For example, your output file at the oracle cloud server is located at /home/**fchen26**/seven.csv and remotely copied to the file “seven.csv”.

|  |
| --- |
| $ **scp fchen26@ipaddress:/home/fchen26/seven.csv seven.csv** |

**Note:** Make sure to replace the database name with your **username** and put the appropriate ip address.

Enter your password to download the file.

**Alternatively**, for Windows users, you may use psftp to download the file. You need to download it at <https://the.earth.li/~sgtatham/putty/latest/w64/psftp.exe>.

For example, in order to download ‘seven.csv’ from the server of Oracle Cloud, you have to run psftp as follows. You may read through the commands of *psftp* in the below:

|  |
| --- |
| psftp> **open [ipaddress]**  Login as: **fchen26**  Enter password…  psftp> **ls**  *Listing directory /home/fchen26*  psftp> **get seven.csv** |

**Note:** Do not forget to change the **username** and put the appropriate ip address.

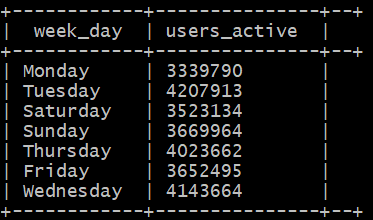
## Analysis 8:

**Step1:** Creating a table and running a query to get the total number of active users by weekday.

First, run the below query:

|  |
| --- |
| **SELECT week\_days.week\_day\_name as week\_day , COUNT(ecommerce\_view.user\_id) AS users\_active**  **FROM ecommerce\_view JOIN week\_days**  **ON (ecommerce\_view.week\_day\_num = week\_days.week\_day\_num)**  **GROUP BY week\_day\_name;** |

You will see a result similar to below:



The below query will create a table using the above query and store the results in hdfs for visualization:

|  |
| --- |
| **DROP TABLE IF EXISTS top\_users\_most\_purchased;**  **CREATE TABLE active\_users\_by\_weekday**  **ROW FORMAT DELIMITED FIELDS TERMINATED BY ','**  **STORED AS TEXTFILE LOCATION '/user/fchen26/ecommerce/eight'**  **AS**  **SELECT week\_days.week\_day\_name as week\_day , COUNT(ecommerce\_view.user\_id) AS users\_active**  **FROM ecommerce\_view JOIN week\_days**  **ON (ecommerce\_view.week\_day\_num = week\_days.week\_day\_num)**  **GROUP BY week\_day\_name;** |

Then, you need to check if the table is created successfully or not:

|  |
| --- |
| **SHOW tables;** |

**Step 2:** After the Hive tables are created, you can download it to your lab (or personal PC/Laptop) as follows.

Open another terminal with git bash, minty, or putty, which is to connect the Oracle cloud to download the output file 000000\_0 at the HDFS path “/user/**fchen26**/ecommerce/**eight** ”.

First, run the following hdfs command to list what file exists at “/user/**fchen26**/ecommerce/**eight**” directory that is actually the location of the Hive table. For some reason it is split into 47 separate files. We run a command to merge them later.

**Note:** Do not forget to change the username.

|  |
| --- |
| $ **hdfs dfs -ls ecommerce/eight** |

This is the merge command to merge the 47 files into a single file.

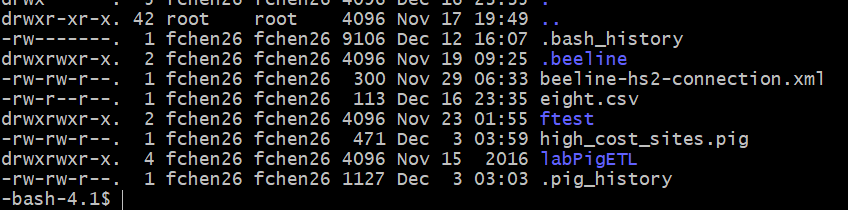
|  |
| --- |
| $ **hadoop fs -cat /user/fchen26/ecommerce/eight/\* | hadoop fs -put - /user/fchen26/ecommerce/eight/000046\_1** |

You can view the contents of the merged file with the below command:

|  |
| --- |
| $ **hdfs dfs -cat ecommerce/eight/000046\_1** |

Then, download the file to the local file systems:

|  |
| --- |
| $ **hdfs dfs -get ecommerce/eight/000046\_1 eight.csv**  $ **ls -al** |



Lastly, open another terminal with git bash, minty, or putty in order to read/import the output file using your lab computer (or your PC/Laptop) - you have to download the file to your lab computer (or your PC/Laptop). For example, your output file at the oracle cloud server is located at /home/**fchen26**/eight.csv and remotely copied to the file “eight.csv”.

|  |
| --- |
| $ **scp fchen26@ipaddress:/home/fchen26/eight.csv eight.csv** |

**Note:** Make sure to replace the database name with your **username** and put the appropriate ip address.

Enter your password to download the file.

**Alternatively**, for Windows users, you may use psftp to download the file. You need to download it at <https://the.earth.li/~sgtatham/putty/latest/w64/psftp.exe>.

For example, in order to download ‘eight.csv’ from the server of Oracle Cloud, you have to run psftp as follows. You may read through the commands of *psftp* in the below:

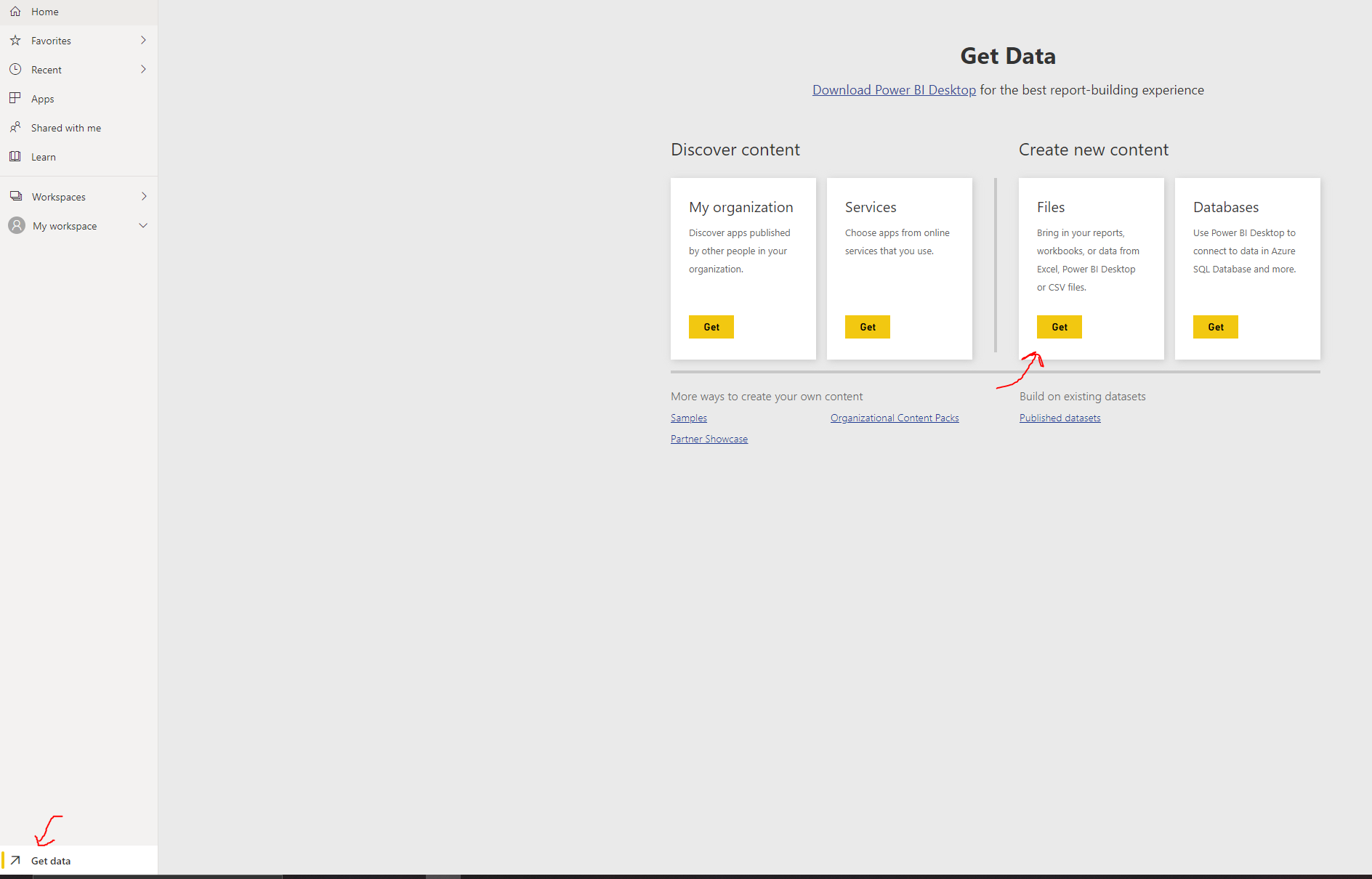
|  |
| --- |
| psftp> **open [ipaddress]**  Login as: **fchen26**  Enter password…  psftp> **ls**  *Listing directory /home/fchen26*  psftp> **get eight.csv** |

**Note:** Do not forget to change the **username** and put the appropriate ip address.

Step 5: Visualizing Data using Business Intelligence Tools

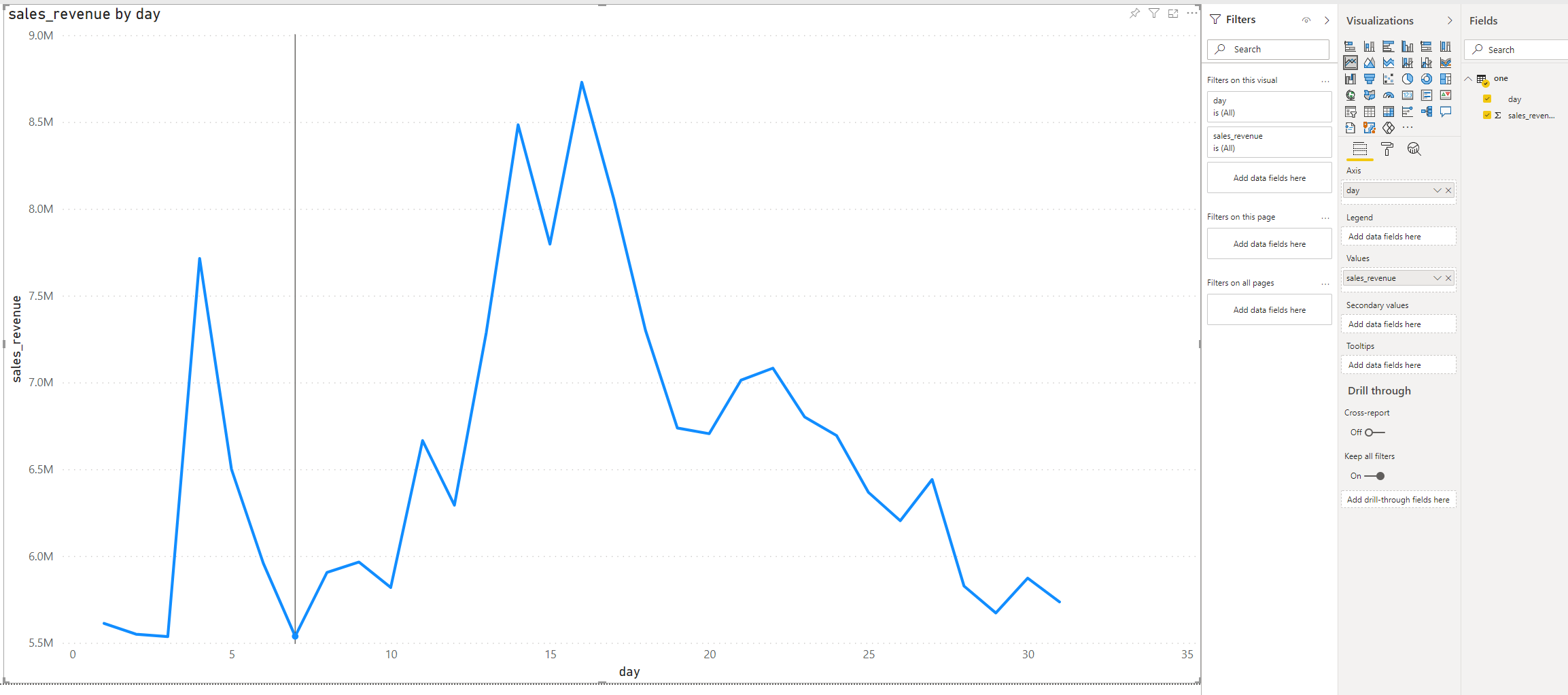
You will use the results data files (csv files) to create visualizations.

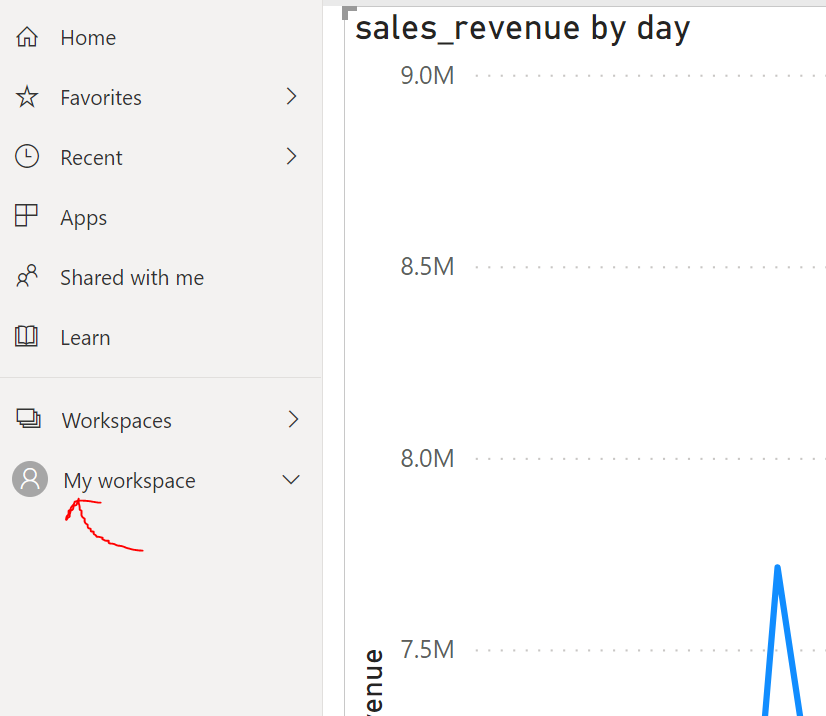
## Visualization 1 and 5: Power BI Line Chart & Pie Chart

**Step 1:** Open PowerBI,import the data into PowerBI by clicking Get Data on the bottom left and then clicking the Get in the Files to import files **one.csv** and **five.csv**.

**Step 2**: PowerBI would prompt you to the dashboard. Double click **one.csv**.

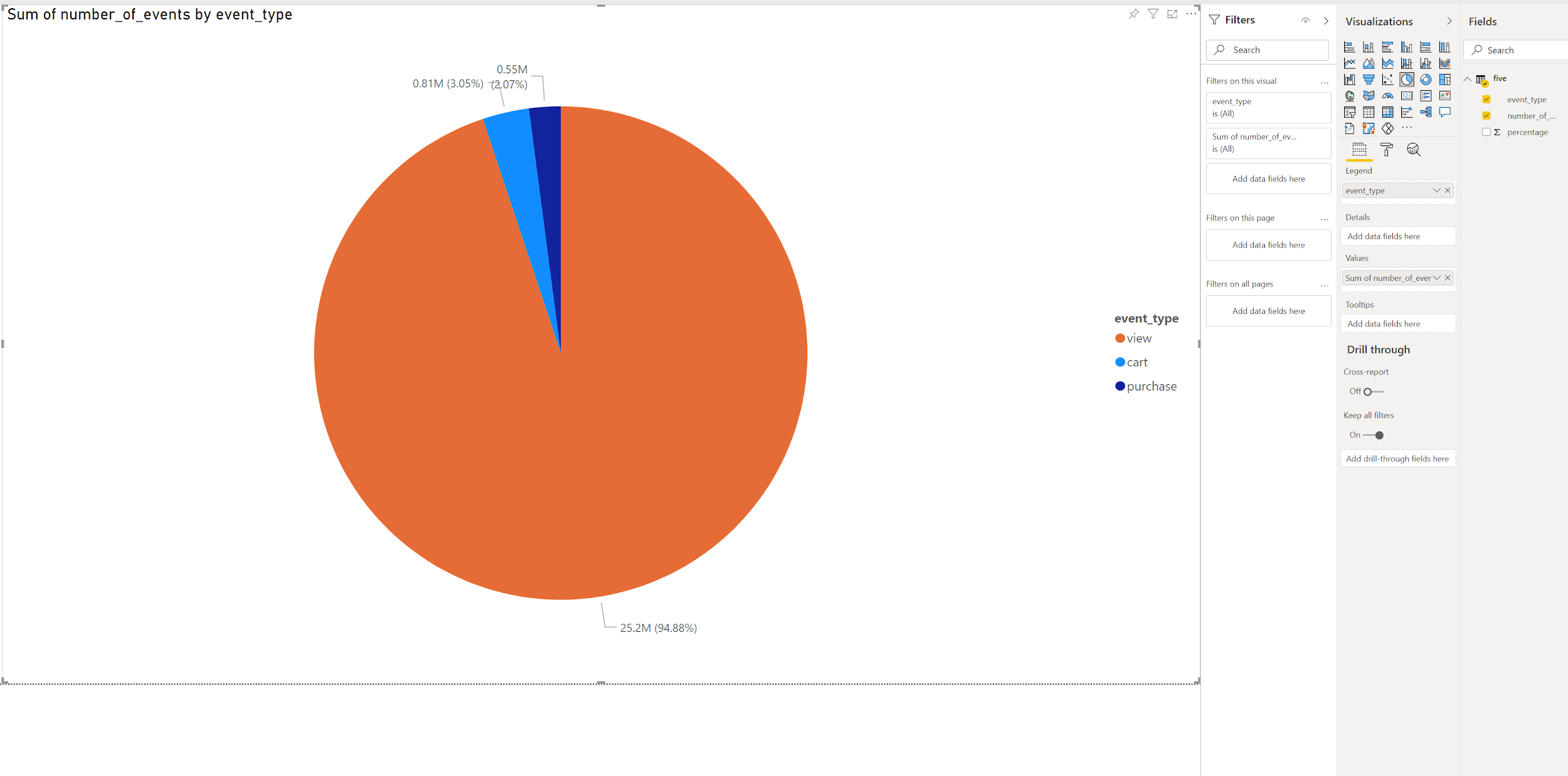
Choose the line chart from the chart options and move the day field to axis and sales revenue to Values.



**Step 3**: Since we already uploaded five.csv , we can access it in My Workspace

**Step 4:** Click **five.csv** and move to the dashboard.

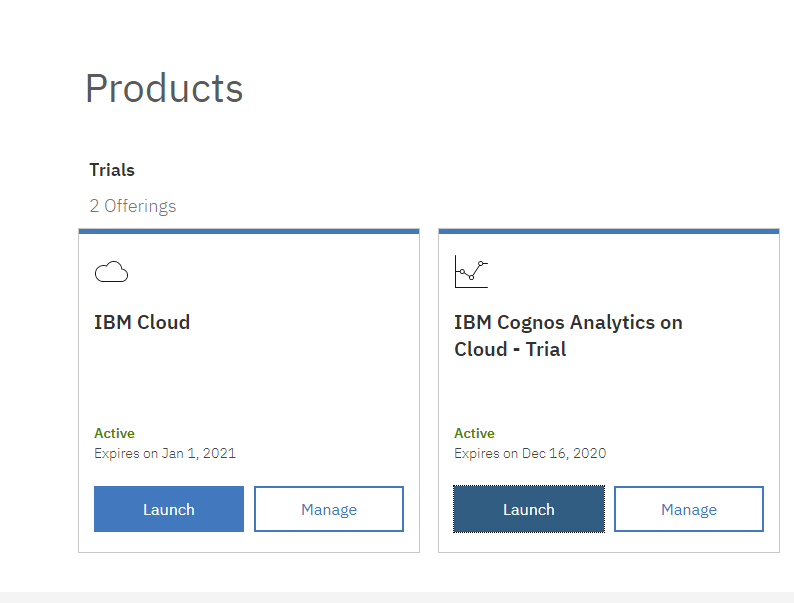
**Step 5**: Choose the Pie Chart from the options and move event type to legend and number of events into values.

The Default for the values is count so be sure to change it to SUM.

## Visualization 2: Forecasting and Trend Lines in IBM Cognos

**Step 1:** Open **two.csv** in IBM Cognos.

1. Log into your IBM Cognos Account or SignUp for a Free 30 day trial. Once you have logged in, you will see a list of your available products.



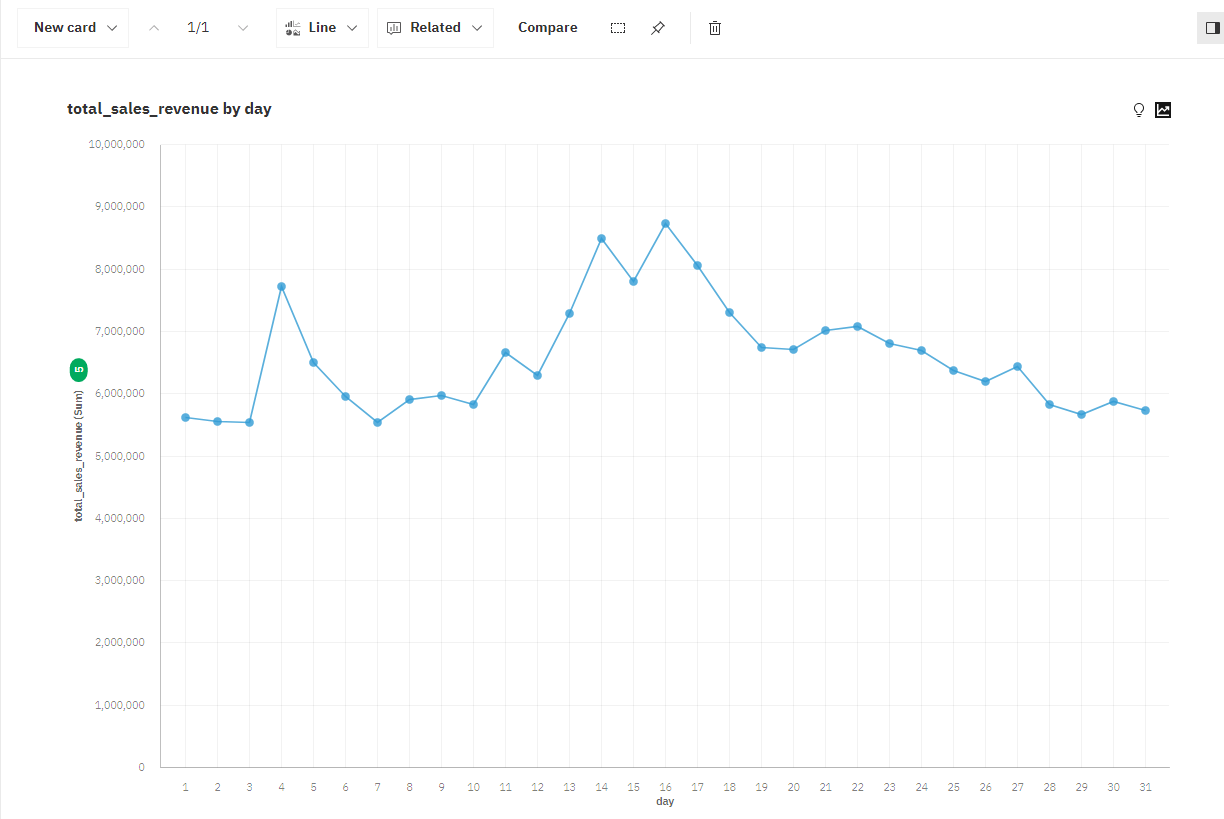
1. Launch IBM Cognos Analytics and select the My Content folder on the left panel. Select the + indicator and choose Upload Files. Search for two.csv on your desktop and open the file.

**Step 2:** Prepare the data for visualization.

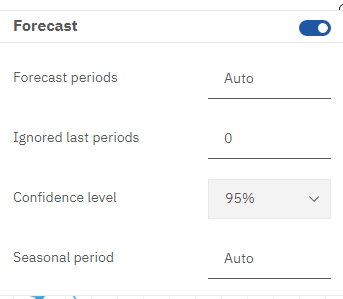
1. Return to the Home screen and select New on the bottom of the screen under the Manage tab. Once you select New, select Exploration.
2. It will bring you to the My Content folder where you will select **two.csv** and click the Add button.

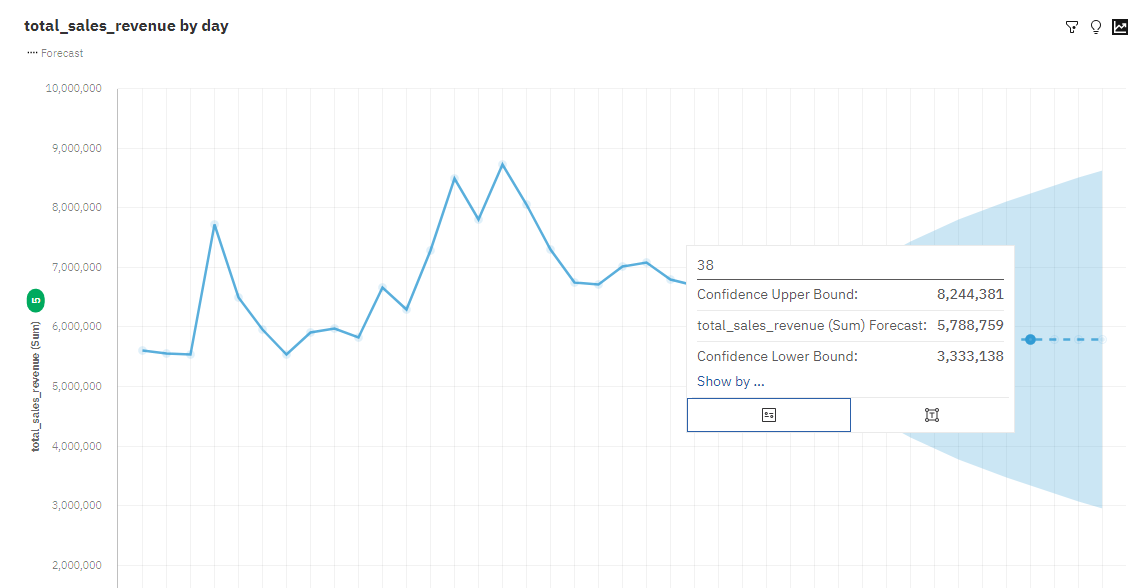
**Step 3:** Use the IBM Cognos Forecasting Feature.

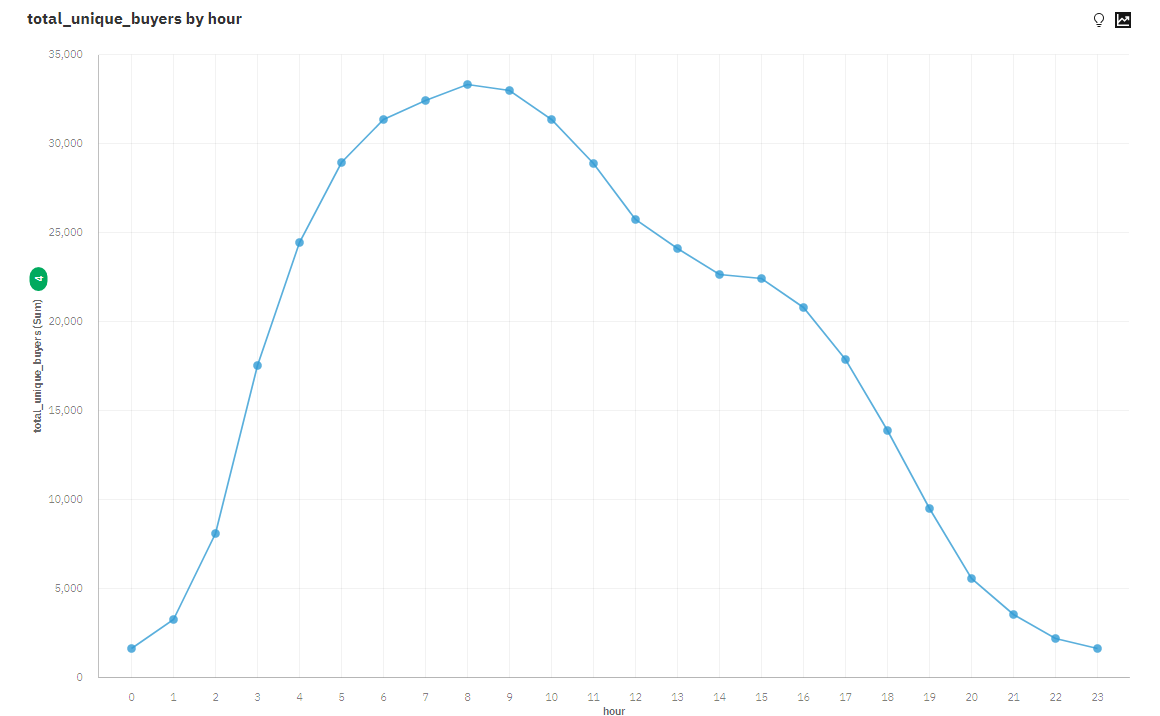
1. Select New Card and from the drop down menu, select Single. This will bring you to the Create a visualization page.
2. Select Choose a type then choose Line under the Trend section.
3. Drag total\_sales\_revenue to the y-axis, and day to the x-axis. This will give you a line graph of the total\_sales\_revenue by day.



1. Select the black forecasting icon on the upper right corner of the graph and toggle the Forecast to on. Input Forecast periods as 10, a 95% confidence level, and Seasonal period to Auto.



1. Select the 38th day, you will see the Upper Bound and Lower Bound forecast for Sales Revenue on the 38th day. 
2. Select New card and follow number 1 and 2 to create a new line graph. Drag total\_unique\_buyers to the y-axis and hour to the x-axis. You will now see an evident peak at hour 8 for unique buyers.



1. Click the blue save icon in the upper left corner to save your worksheet.

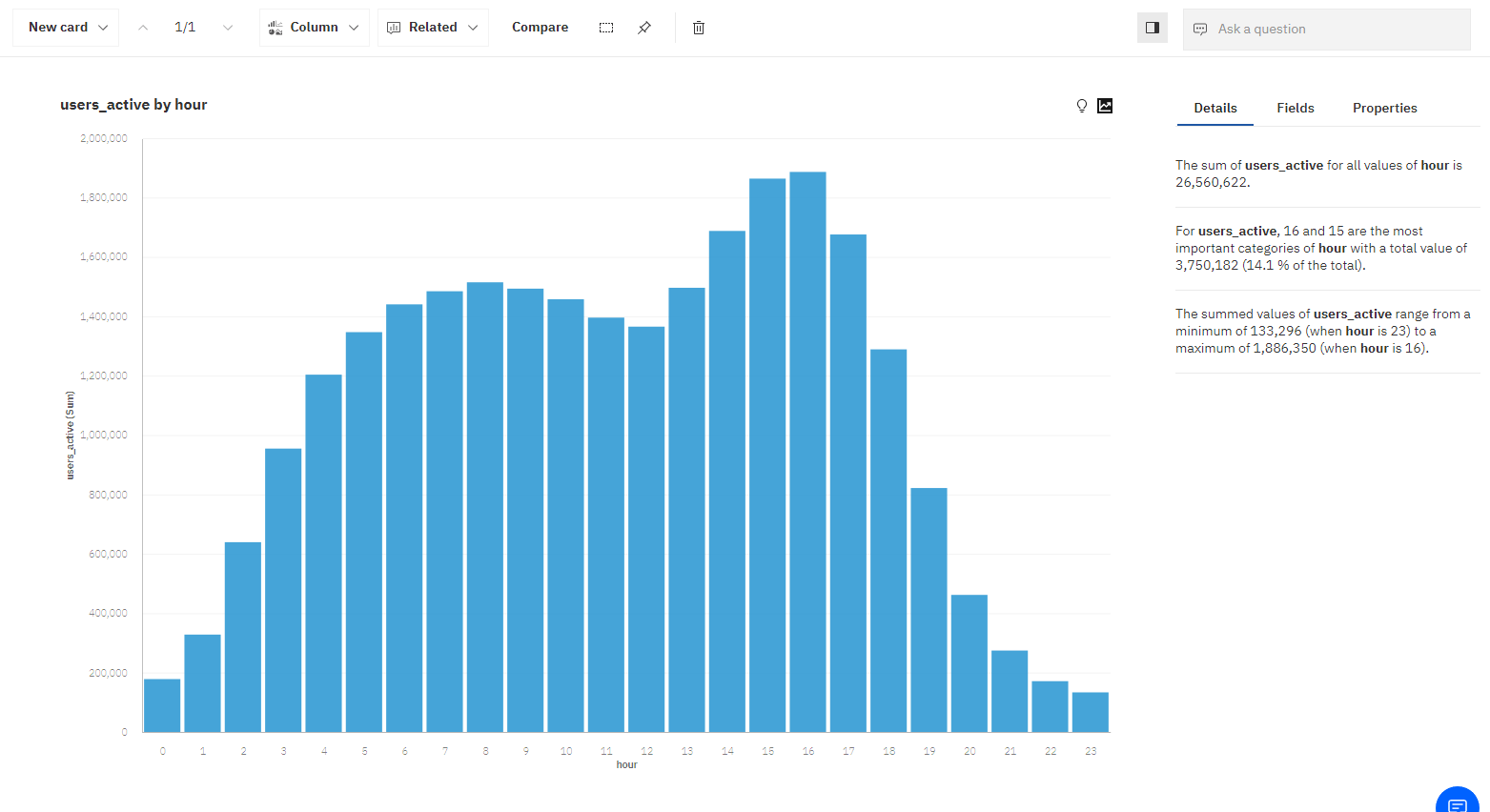
## Visualization 3: Column Charts in IBM Cognos

**Step 1:** Open **three.csv** in IBM Cognos and prepare for visualization.

1. Return the home screen after saving Visualization 3. Select the My Content folder on the left pane and click the + indicator. Upload **three.csv** to your My Content folder.
2. Select New under the Manage and tab and choose Exploration. Add three.csv from your My Content folder.

**Step 2:** Create a column chart in IBM Cognos.

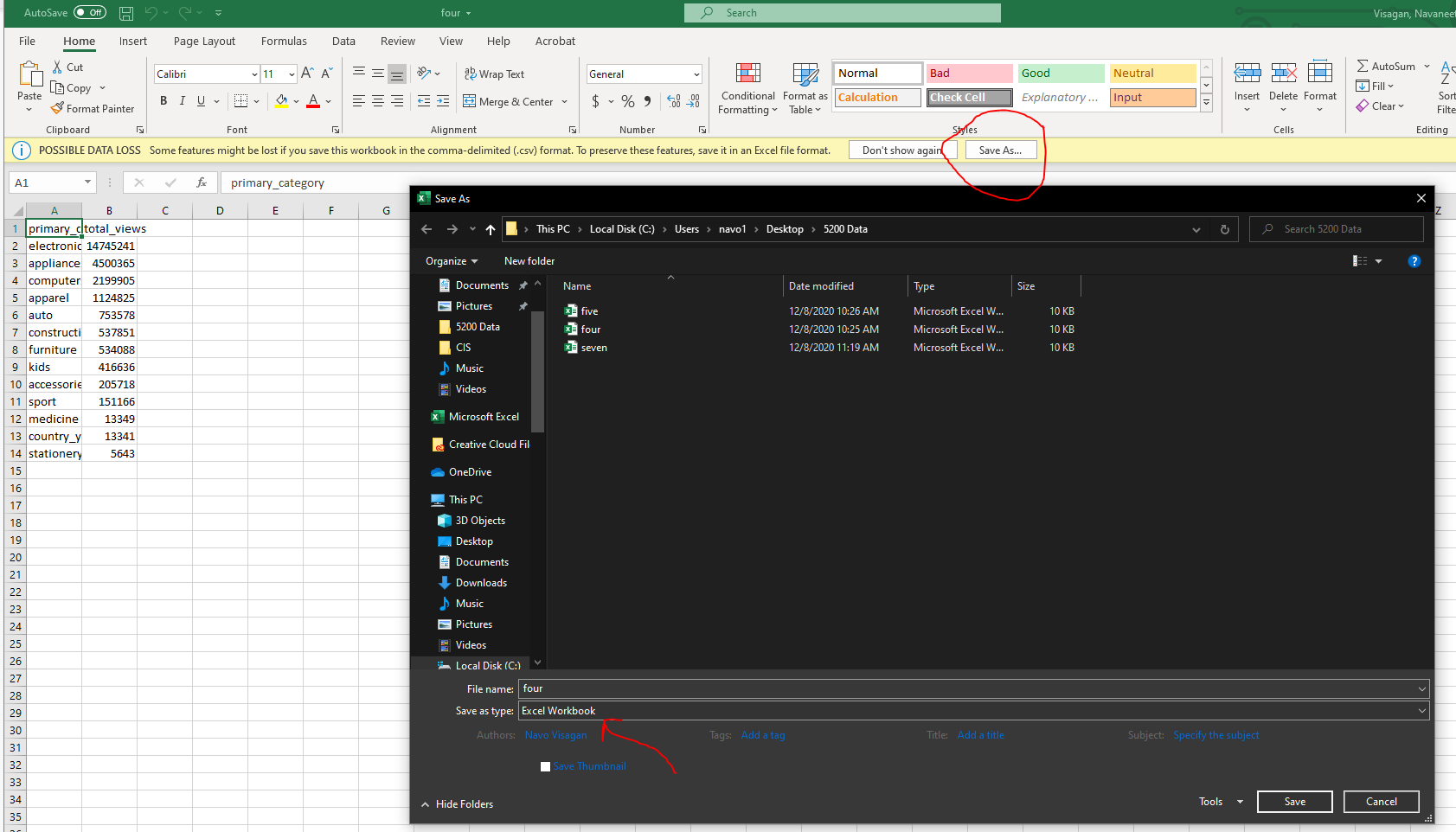
1. Select a New Card and choose Single. Under the create a visualization screen select Choose a Type. Select Column as the visualization type under the Comparison section.
2. Drag users\_active to the Length and hour to the bars. Click on the Details tab on the left to get additional insight on the column chart created. You can see that users are most active between 14:00 and 17:00.



1. Save your card using the blue disk icon in the upper left corner.

## Visualization 4 and 7: Bubble Map and Bar Chart in Tableau

**Step 1:** Convert four and seven CSV files into Excel workbook:

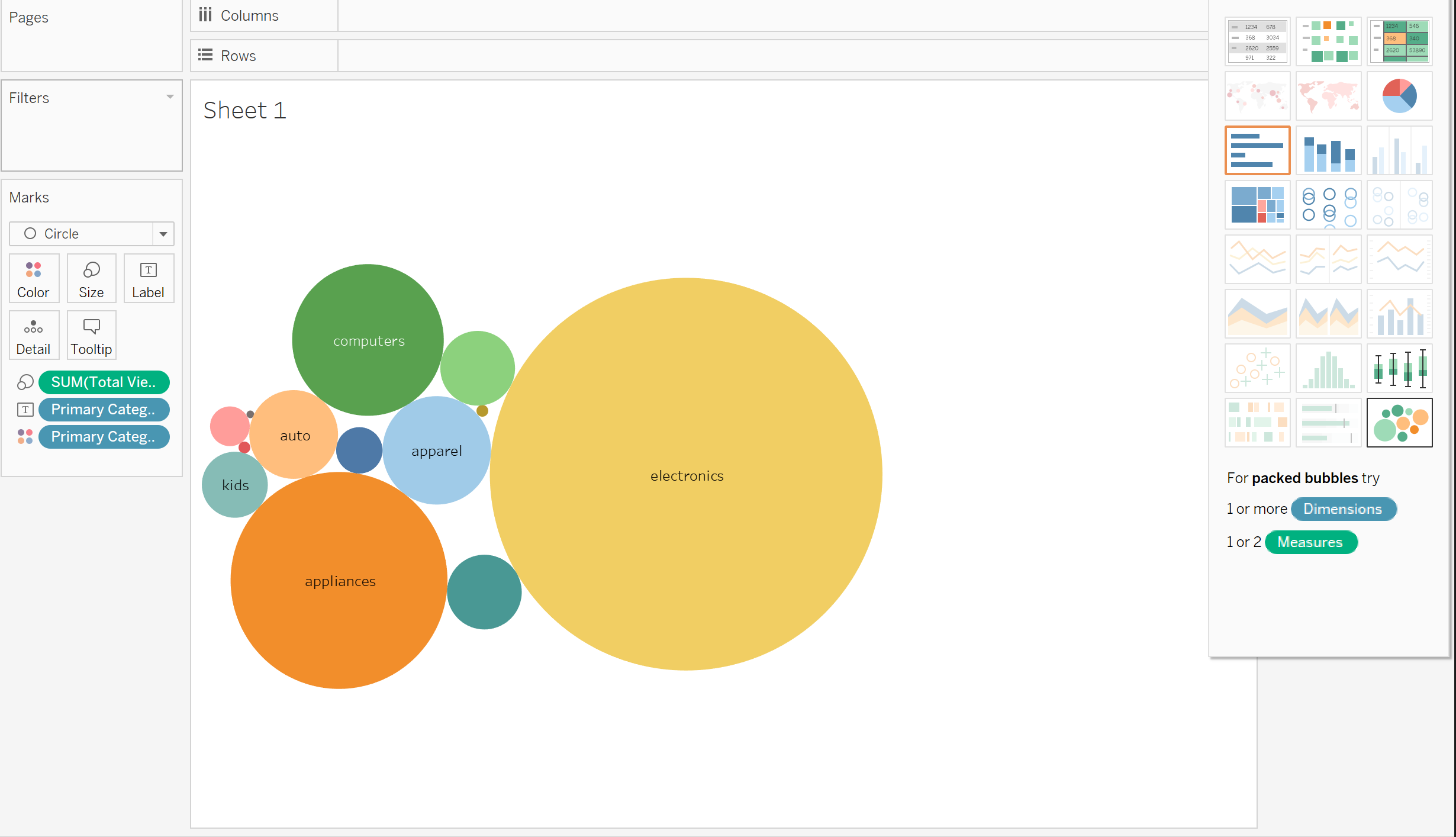
Open **four.csv** and **seven.csv** in Excel and save as Excel Workbook.

**Step 2:** Open Tableau and Import:

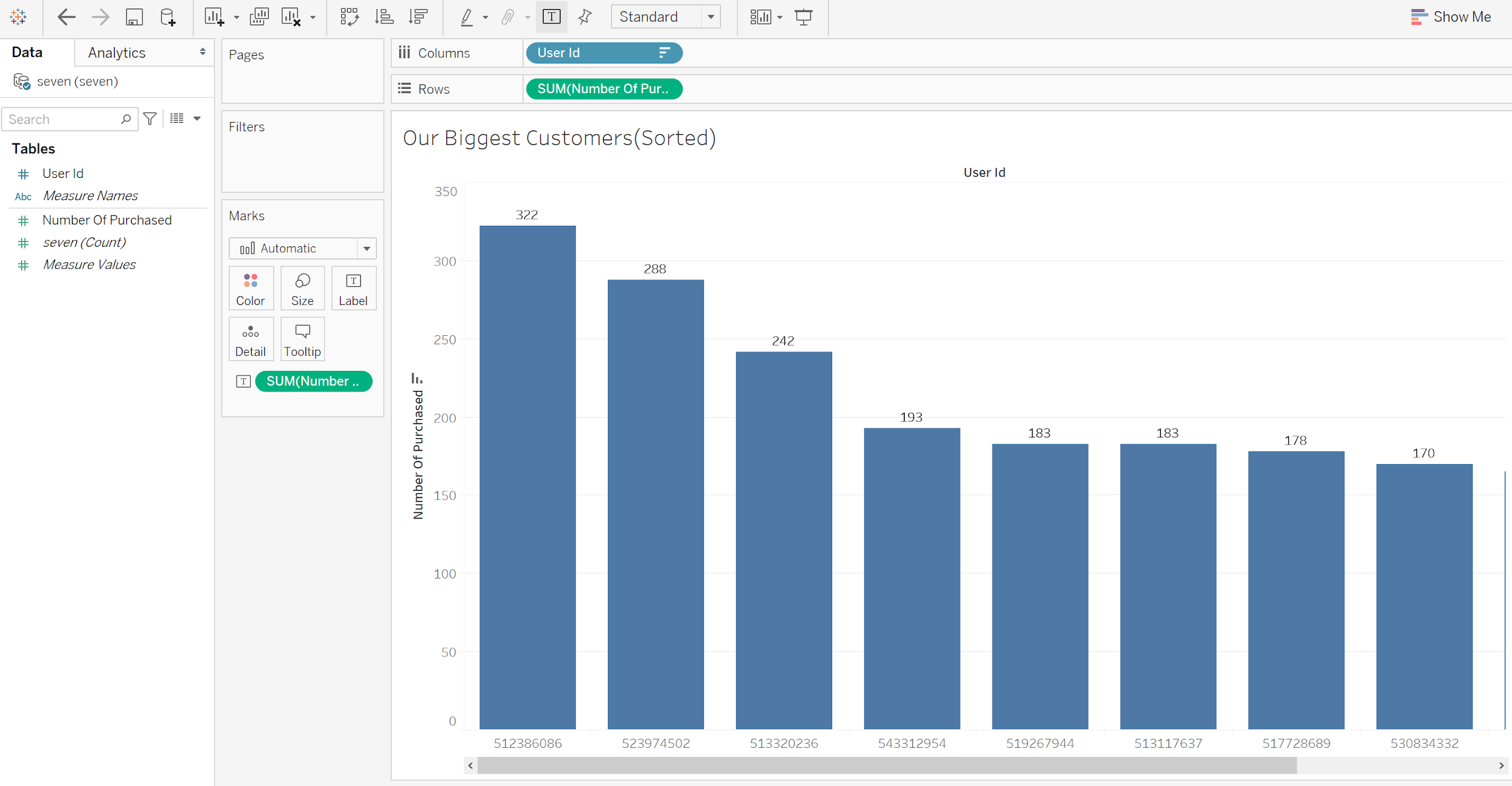
Open Tableau and choose Microsoft Excel as the data source.

When prompted choose **four.xslx** and move to Sheet 1.

**Step 3**: Move The Primary Category to the Columns and the Total Views into Rows.

**Step 4**: Click the Show Me on the top right corner and choose the very bottom right option(Bubble Map).

**Step 5**: Open a new Tableau Workbook and connect the Data Source to **seven.xlsx.**

**Step 6** : Move User ID to the Columns Tab and Number of Purchased to Rows.

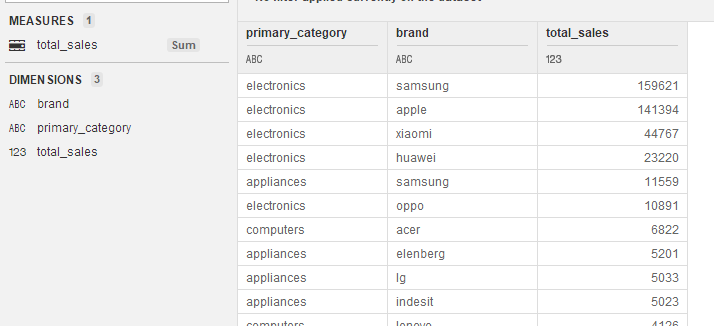
## Visualization 6: Clustering in SAP Expert Analytics

**Step 1:** Open **six.csv** in SAP Expert Analytics.

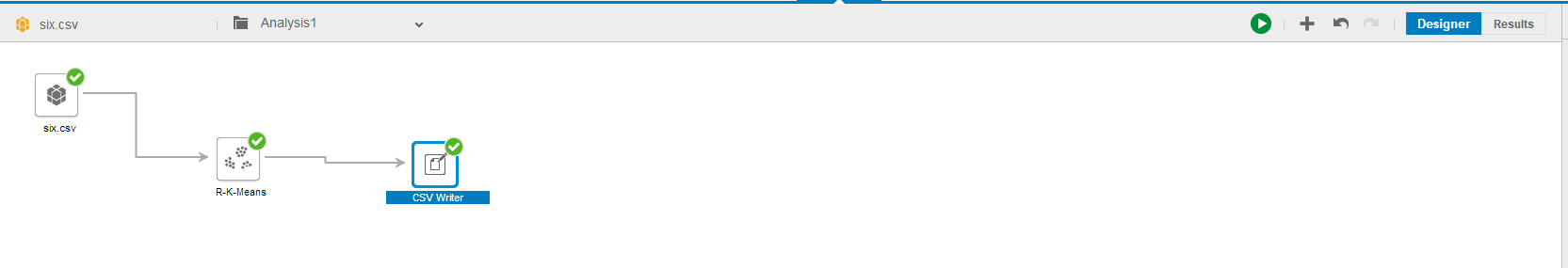
1. You must launch SAP Predictive Analytics and select the Expert Analytics tab on the left ribbon. Once you are on the Expert Analytics screen, select Expect Analytics to launch the program.
2. In the upper left hand corner, select File and from the drop down menu click new.
3. From the Add New Dataset screen, select Text as your source type, then press the Next button. Select the dataset from your desktop and ensure that the separator is toggled on Delimited By and the drop down menu has Comma selected. Click create to create your new worksheet in SAP Expert Analytics.

**Step 2:** Prepare the data for visualization.

1. In the Prepare tab, ensure that your data set has the proper measures and dimensions. Total\_sales should be aggregated as a sum, while brand and primary\_category should be ABC.

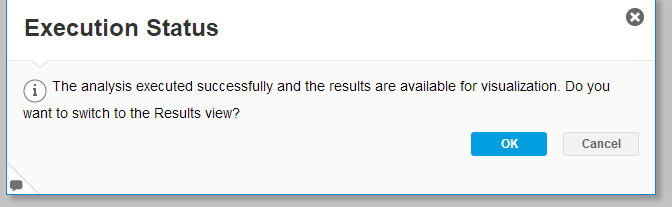


1. Select the Predict tab and you should see that **six.csv** is listed as the dataset. In the algorithms panel on the right of the screen, select R-K-Means and drag it to the center of the screen. You will see that it will link to six.csv, indicated by a green arrow.
2. Click the toggle icon on R-K-Means and select Configure Settings or F5. Denote the number of clusters as 3 and select total\_sales before clicking Done.
3. From the right panel, select Data Writers and under File Writers, select CSV Writer and drag it to the center screen. You now have **six.csv**, R-K-Means, and CSV Writer linked.
4. Select the toggle icon on CSV Writer, then Configure Settings or F5, and give the file a generic name to be saved locally.
5. Your output should look as follows before selecting the green Run button on the upper pane.

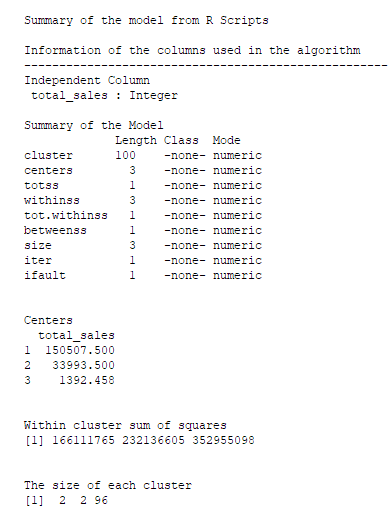


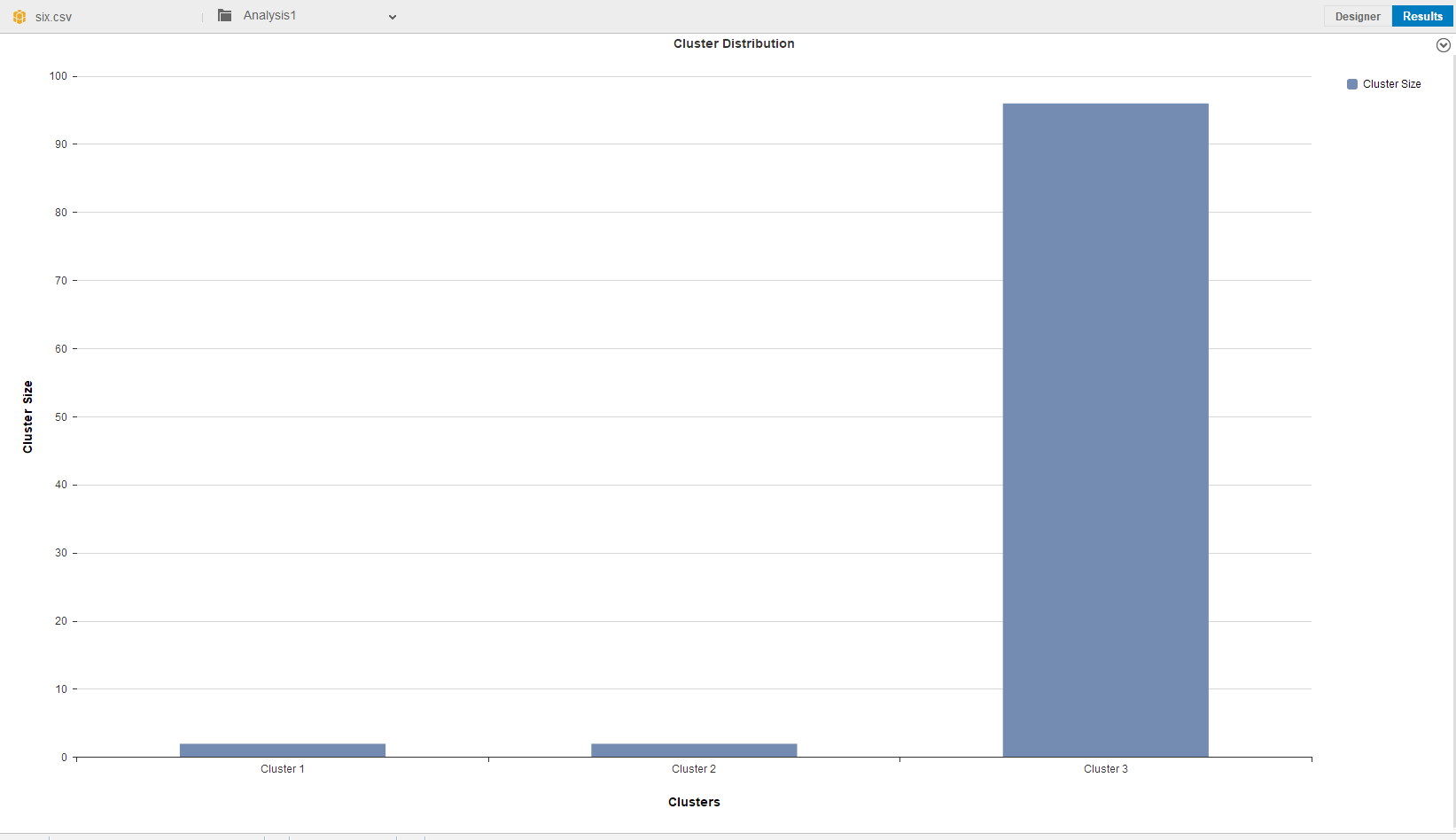
**Step3:** Configure the visualization.

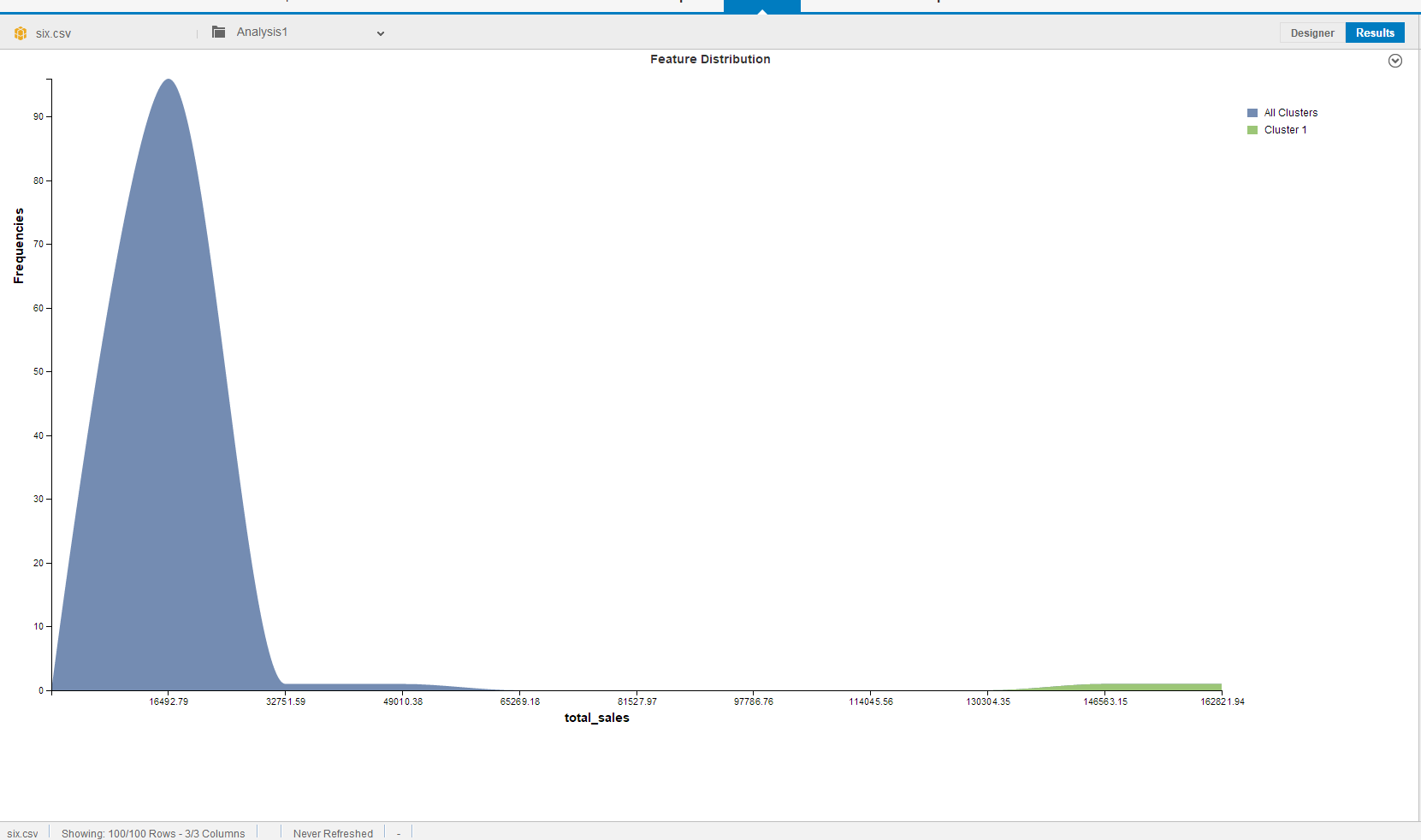
1. After selecting Run, you will get the following notification. Select OK.



1. On the Results View, you will now see ClusterNumber added as a column to your data set. Select Summary on the right panel to see a summary of the K-Means-Analysis.



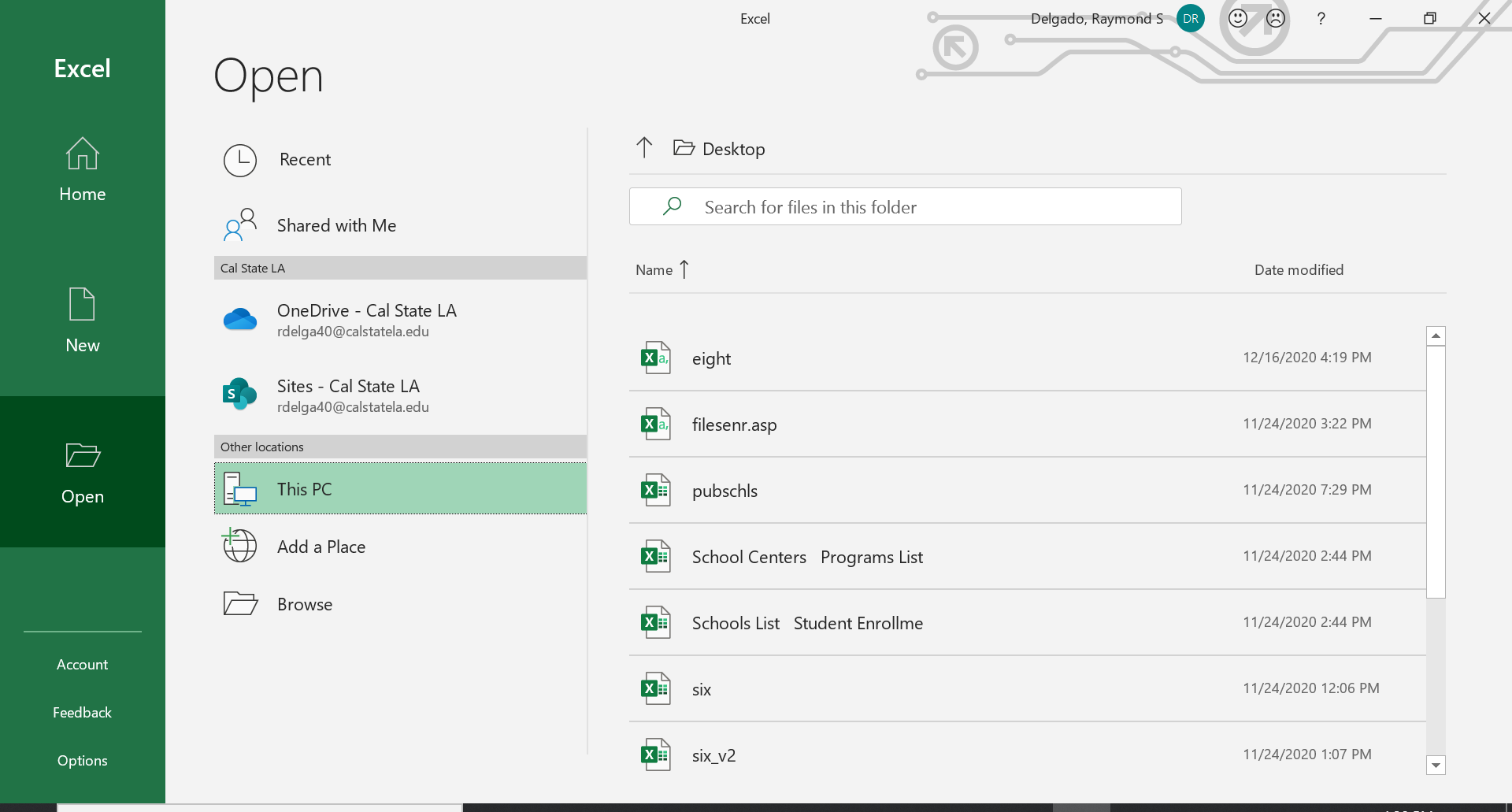
1. Select the Bar Column under Cluster Representations to see the distribution of the clusters. As you can see, most of the data lies in cluster 3. 
2. Select Feature Distribution under the Cluster Representations tab. You will see that the output is skewed to the left.



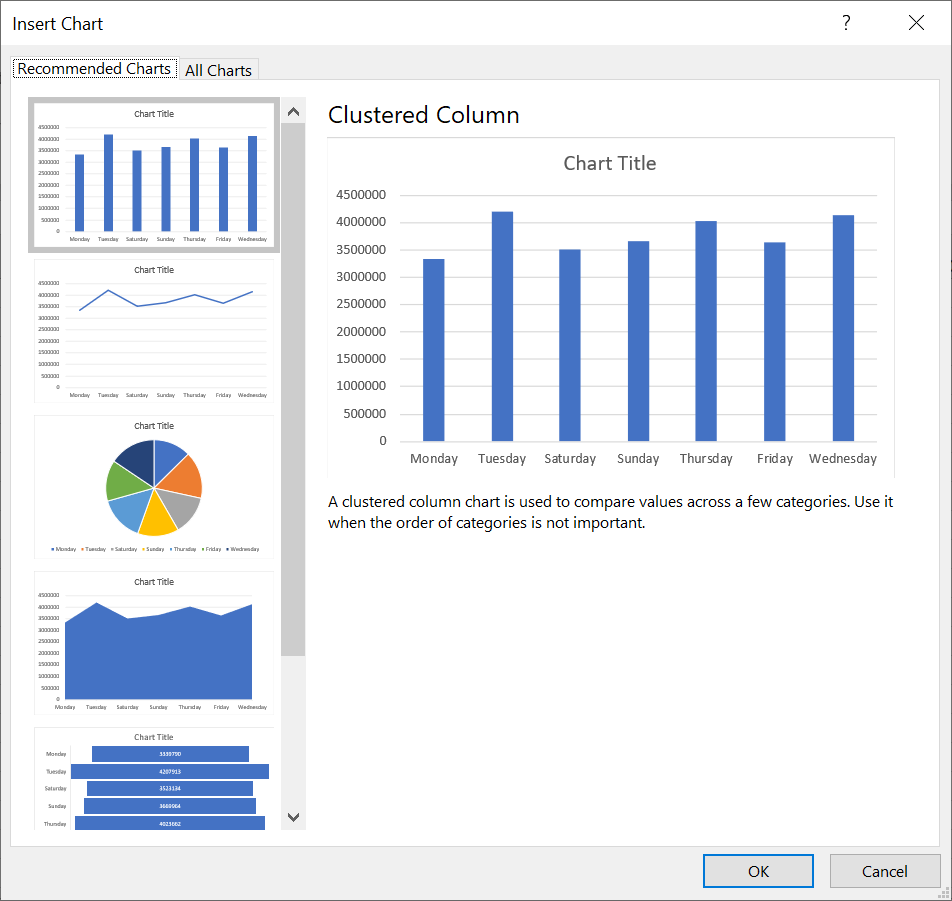
1. Select File in the upper left and Save your worksheet.

Visualization 8: Column Chart in Microsoft Excel

**Step 1:** Open the “**eight.csv**” in Microsoft Excel.



**Step 2**: Click on the Insert Tab, then select the recommended chart, and finally select the clustered column chart. Click Ok to confirm your decision.

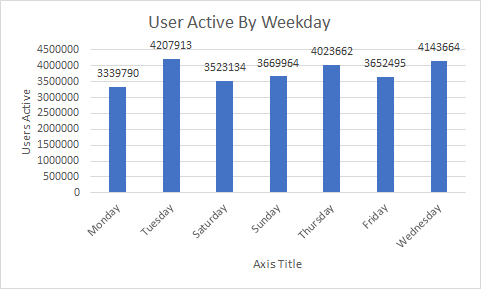


**Step 3**: Click on the green plus(+) icon to add chart elements. The chart elements that we want to add are:

* Axis titles: We are going to name the y-axis as “Active Users”
* Data Label: This will show us the total number of users active per weekday.

Please Note: It is recommended to rename the chart.

Your chart should look similar to the picture below:



References

* 1. URL of Data Source: [eCommerce behavior data from multi category store](https://www.kaggle.com/mkechinov/ecommerce-behavior-data-from-multi-category-store)
  2. URL of your Github: <https://github.com/neltf/CSULA_CIS_5200>
  3. URL of References:
* <https://medium.com/tech4she/visualising-e-commerce-user-behaviours-c833def97cc0>
* <https://www.kaggle.com/adilemrebilgic/e-commerce-analytics>
* <https://towardsdatascience.com/will-customers-buy-the-products-in-their-cart-b8ac5e30f3>