# [https://avatars2.githubusercontent.com/u/4156894?v=3&s=100](http://www.calstatela.edu/centers/hipic) CIS5200 Term Project Tutorial

#### Authors: Sai Kiran Gontyala, Yoshitha Gattu, Uchechukwu Okafor, Noelle Cespedes, Oluwatofunmi Alabi

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

#### Date: 05/18/2019

**Lab Tutorial**

Sai Kiran Gontyala ([sgontya@calstatela.edu](mailto:sgontya@calstatela.edu))

Yoshitha Gattu ([ygattu@calstatela.edu](mailto:ygattu@calstatela.edu))

Uchechukwu Okafor ([uokafor3@calstatela.edu](mailto:uokafor3@calstatela.edu))

Noelle Cespedes ([ncesped3@calstatela.edu](mailto:ncesped3@calstatela.edu))

Oluwatofunmi Alabi (oalabi@calstatela.edu)

05/18/2019

**Data Analysis using Spark (PUBG death rate)**

**Objectives**

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

* Uploaded the dataset for PUBG death rate
* Data cleaning using Hive
* HiveQL commands to perform the analysis
* Visualization in Power BI, Tableau and MS Excel

**Platform Spec**

* Oracle Big Data Compute Edition
* CPU Speed: 2.195 GHz
* # of CPU cores: 4 cores, 1 Socket
* # of nodes: 5 nodes
* Total Memory Size: 8GB

**INTRODUCTION**

Video games are a rich area for data extraction due to its digital nature. PUBG is a popular game PlayerUnknown's Battlegrounds. This dataset provide over 720,000 competitive matches. PUBG is a first/third-person shooter battle royale style game that matches over 90 players on a large island where teams and players fight to the death until one remains. This project will demonstrate the usage of Hadoop, MapReduce, and Hive on big data. We will apply the knowledge learned during the lecture, extensive researches and development of HiveQL in order to generate data and visualize it on Power BI, Tableau and 3D maps.

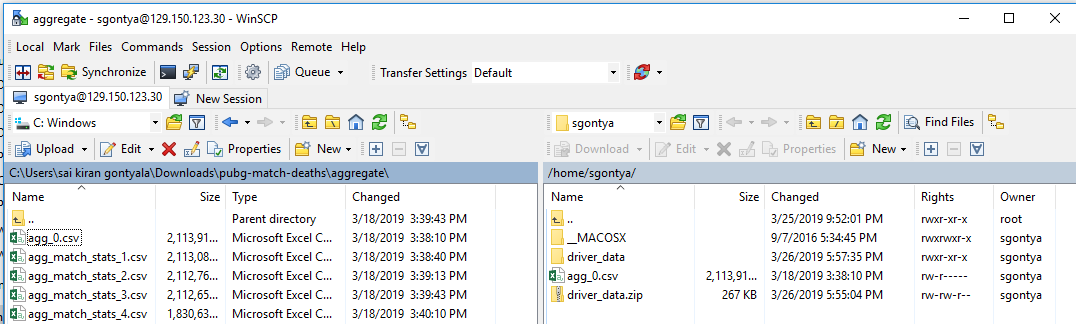
**Step 1: Data Upload to server**

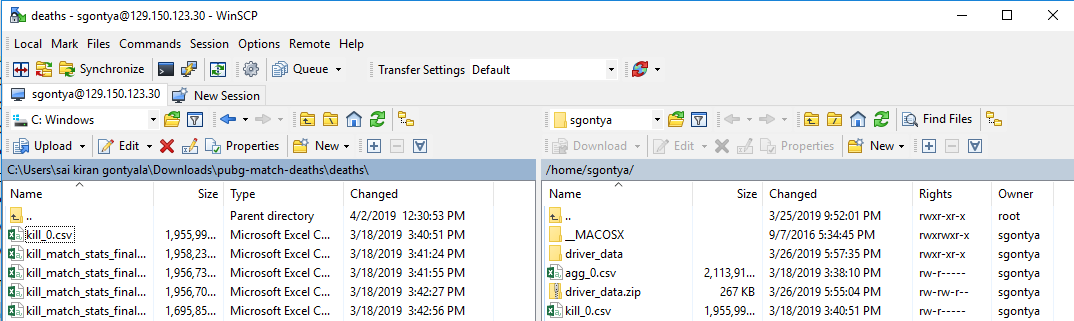
**Server IP:** **129.150.123.30**

* Download the dataset from the url:

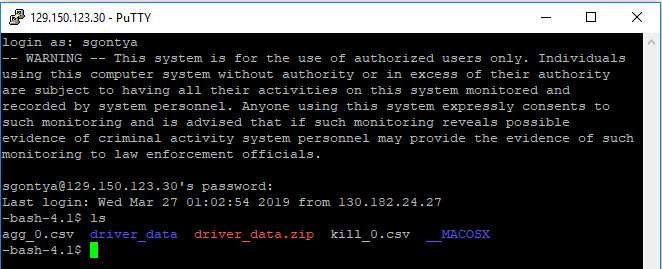
<https://www.kaggle.com/skihikingkevin/pubg-match-deaths>

* Use the software WinSCP to upload the dataset agg\_0.csv and kill\_0 of size 2GB each [8GB unzipped] from local system to oracle server.





* Use the command **ls** to list the files as a csv format.



**Step 2: Uploading data to HDFS**

* Create folders using **mkdir** command, Change the permission for the file using below command to get full access to the file.

**hdfs dfs -mkdir Group4/agg\_0**

**hdfs dfs -chmod -R o+w Group4/agg\_0**

**hdfs dfs -mkdir Group4/kill\_0**

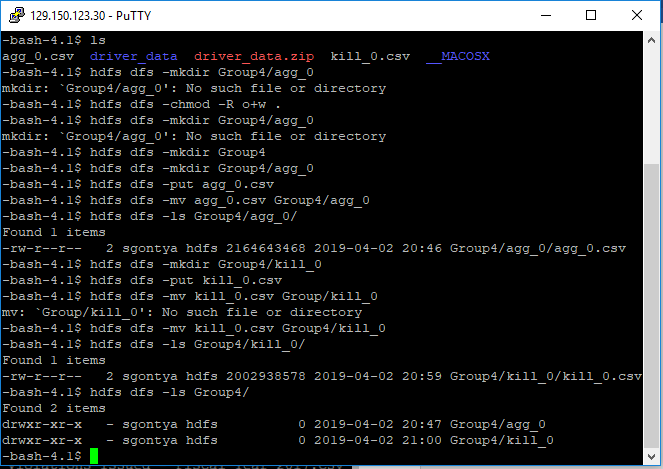
**hdfs dfs -chmod -R o+w Group4/kill\_0**

* Use **hdfs dfs -put** command to upload the data from the Linux server to HDFS

**hdfs dfs -put agg\_0.csv**

**hdfs dfs -put kill\_0.csv**

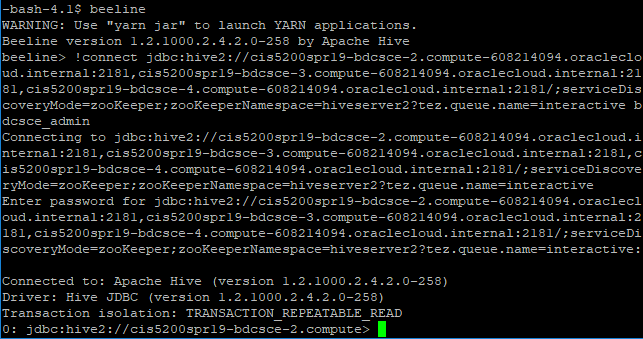
* Use **hdfs dfs -mv** to move file from hdfs parent folder to created directory.



**Step 3: Connecting server to HIVE**

* Use the command **beeline** to connect with to Hive

**!connect jdbc:hive2://cis5200spr19-bdcsce-2.compute-608214094.oraclecloud.internal:2181,cis5200spr19-bdcsce-3.compute-608214094.oraclecloud.internal:2181,cis5200spr19-bdcsce-4.compute-608214094.oraclecloud.internal:2181/;serviceDiscoveryMode=zooKeeper;zooKeeperNamespace=hiveserver2?tez.queue.name=interactive bdcsce\_admin**



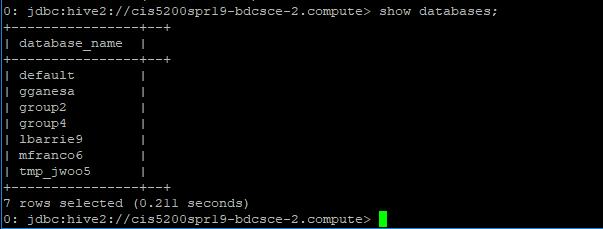
**Step 4: Creating Database**

* Use **create database** to create a database named group3.

**create database group4;**

* To view the newly created database use.

**Show databases;**



**Step 5: Creating Tables in Database**

* To **Create External table,** use below command,

CREATE EXTERNAL TABLE IF NOT EXISTS aggregate\_0

(date\_time BIGINT,game\_size INT,match\_id STRING,match\_mode STRING,party\_size INT,player\_assists INT,player\_dbno INT,player\_dist\_ride FLOAT,player\_dist\_walk FLOAT,player\_dmg INT,player\_kills INT,player\_name STRING,player\_survive\_time FLOAT,team\_id INT,team\_placement INT)

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION '/user/sgontya/Group4/agg\_0'

TBLPROPERTIES ('skip.header.line.count'='1');

CREATE EXTERNAL TABLE IF NOT EXISTS kills\_0

(killed\_by STRING,killer\_name STRING,killer\_placement INT,killer\_position\_x FLOAT,killer\_position\_y FLOAT,map\_location STRING,match\_id STRING,time INT,victim\_name STRING,victim\_placement INT,victim\_position\_x FLOAT,victim\_position\_y FLOAT)

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION '/user/sgontya/Group4/kill\_0'

TBLPROPERTIES ('skip.header.line.count'='1');

Step 6: Dataset cleaning in Hive

As the PUBG death rate dataset has more number of uwanted data like blank cells and irrelevent data it needs to be filter out and saved as New file Cln\_agg\_0, Cln\_kills\_0 in the path /user/sgontya/Group4/

* Use the below command to clean the dataset,

CREATE TABLE cln\_agg\_0

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION '/user/sgontya/Group4/cln\_agg\_0'

AS

SELECT

date\_time, game\_size, match\_id, match\_mode, party\_size, player\_assists, player\_dbno, player\_dist\_ride, player\_dist\_walk, player\_dmg, player\_kills, player\_name, player\_survive\_time, team\_id, team\_placement from aggregate\_0

where

player\_name != ' '

;

CREATE TABLE cln\_kills\_0

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION '/user/sgontya/Group4/cln\_kills\_0'

AS

SELECT

killed\_by, killer\_name, killer\_placement, killer\_position\_x, killer\_position\_y, map\_location, match\_id, time, victim\_name, victim\_placement, victim\_position\_x, victim\_position\_y from kills\_0

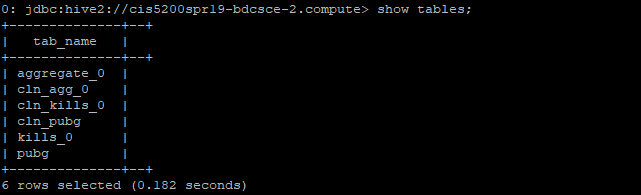
where

killer\_name != ' ' OR killer\_placement != ' ' OR killer\_position\_x != ' ' OR killer\_position\_y != ' ' OR map\_location != ' ' OR victim\_placement != ' ' LIMIT 10;

;

* Use the command to view all the tables in the database,

**Show tables;**



Step 7: Creating Hive Queries to Analyze Data

The following Hive queries analyses the PUBG death rates**.**

1. **Query to analyse the Total number of players in the given dataset.**

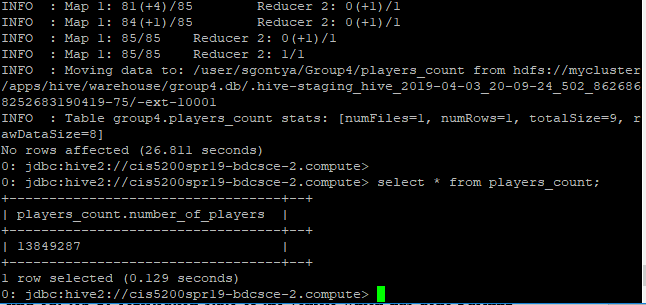
CREATE TABLE **players\_count**

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION '/user/sgontya/Group4/players\_count'

AS

select count(player\_name) as number\_of\_players from cln\_agg\_0;



1. **Query to find out map location people are interested in:**

CREATE TABLE **map\_location**

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION '/user/sgontya/Group4/map\_location'

AS

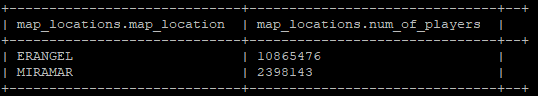
SELECT map\_location, COUNT(map\_location)

FROM cln\_kills\_0

WHERE map\_location IN ('ERANGEL' , 'MIRAMAR')

GROUP BY map\_location

HAVING COUNT(map\_location) > 1;



1. **Query to find out party size people are interested in:**

CREATE TABLE **party\_sizes**

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION '/user/sgontya/Group4/party\_sizes'

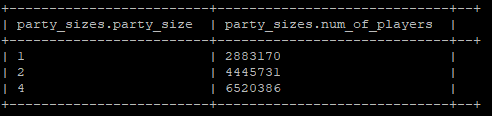
AS

SELECT party\_size, COUNT(party\_size) AS num\_of\_players

FROM cln\_agg\_0

GROUP BY party\_size

HAVING COUNT(party\_size) > 0;



1. **Query to find out number of kills by a player.**

CREATE TABLE **player\_kills**

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION '/user/sgontya/Group4/player\_kills/'

AS

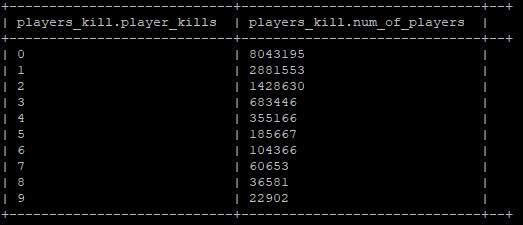
SELECT player\_kills, COUNT(player\_kills)

FROM cln\_agg\_0

GROUP BY player\_kills

HAVING COUNT(player\_kills) > 1

ORDER BY player\_kills desc LIMIT 10;



1. **Query to find out assist points scored by the players.**

CREATE TABLE **player\_assists**

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION '/user/sgontya/Group4/player\_assists/'

AS

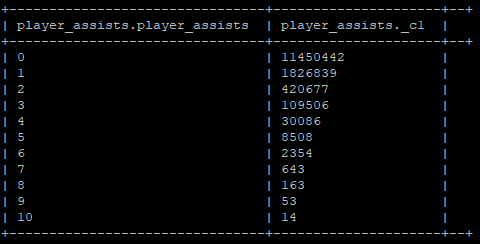
SELECT player\_assists, COUNT(player\_assists)

FROM cln\_agg\_0

GROUP BY player\_assists

HAVING COUNT(player\_assists) > 1

ORDER BY player\_assists asc;



1. **Query to find out survival time of the players**

CREATE TABLE **survival\_time**

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION '/user/sgontya/Group4/survival\_time/'

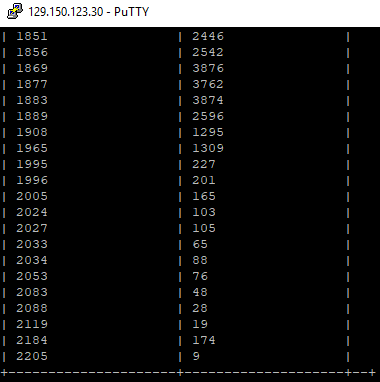
AS

SELECT time, COUNT(time)

FROM cln\_kills\_0

GROUP BY time

HAVING COUNT(time) > 1;



1. **Query to find out Highest distance covered by teams.**

CREATE TABLE **dist\_covered\_team**

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION '/user/sgontya/dist\_covered\_team'

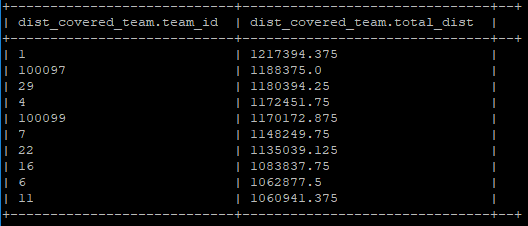
AS

SELECT team\_id, (player\_dist\_ride + player\_dist\_walk) AS total\_dist

FROM cln\_agg\_0

ORDER BY total\_dist DESC

LIMIT 10;



1. **Query to find out COORDINATES WITH HIGHEST NUMBER OF CASUALTIES (killer's position)**

CREATE TABLE **killers\_coordinates**

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION '/user/sgontya/killers\_coordinates'

AS

SELECT killer\_position\_x, killer\_position\_y, COUNT(victim\_name) as casualties

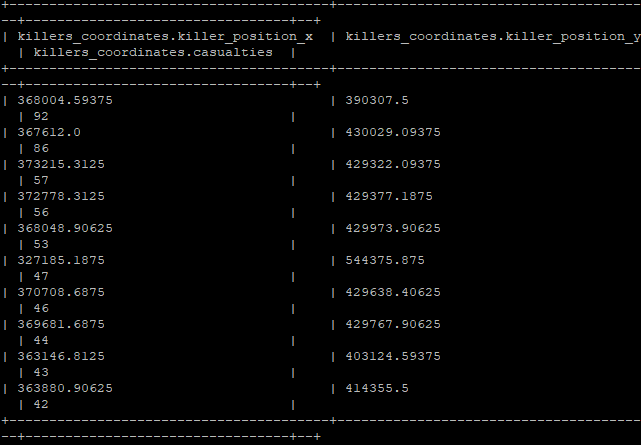
FROM cln\_kills\_0

WHERE killer\_position\_x != 'NULL' OR killer\_position\_x != '0.0'

GROUP BY killer\_position\_x, killer\_position\_y

ORDER BY casualties desc

LIMIT 10;



1. **Query to find out COORDINATES WITH HIGHEST NUMBER OF CASUALTIES(Victim's position)**

CREATE TABLE **victim\_coordinates**

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION '/user/sgontya/Group4/victim\_coordinates'

AS

SELECT victim\_position\_x, victim\_position\_y, COUNT(victim\_name) as casualties

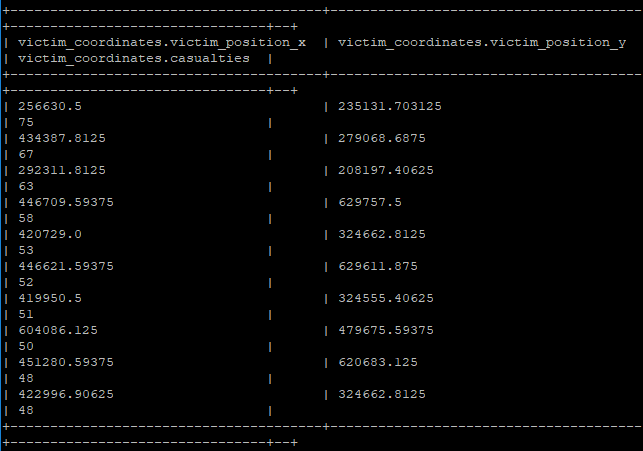
FROM cln\_kills\_0

WHERE victim\_position\_x != '0.0'

GROUP BY victim\_position\_x, victim\_position\_y

ORDER BY casualties desc

LIMIT 10;



10. **Query to find out top 10 DISTANCE COVERED ON FOOT players.**

CREATE TABLE **distance\_on\_foot**

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION '/user/sgontya/Group4/distance\_on\_foot'

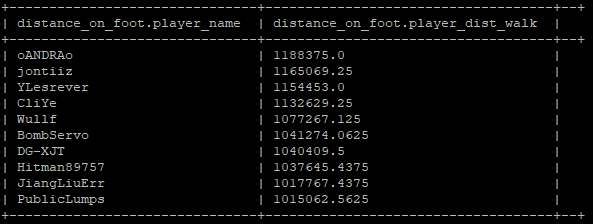
AS

SELECT player\_name, player\_dist\_walk

FROM cln\_agg\_0

ORDER BY player\_dist\_walk DESC

LIMIT 10;



11. **Query to find out HIGHEST SURVIVAL TIME**

CREATE TABLE **highest\_survival\_time**

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION '/user/sgontya/Group4/highest\_survival\_time'

AS

SELECT team\_id, count(player\_survive\_time) as team\_survive\_time

from cln\_agg\_0

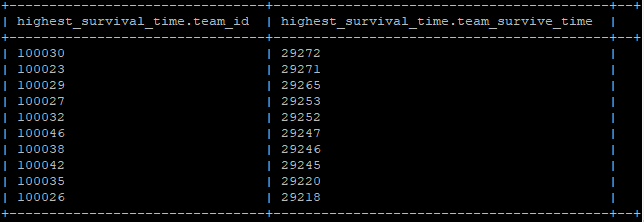
WHERE party\_size = 1

GROUP

BY team\_id

ORDER BY team\_survive\_time desc

LIMIT 10;



12. **Query to find out HIGHEST KILLS-SINGLE PLAYER**

CREATE TABLE **highest\_single\_kills**

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION '/user/sgontya/Group4/highest\_single\_kills'

AS

SELECT team\_id, party\_size, count(player\_kills) as team\_kills

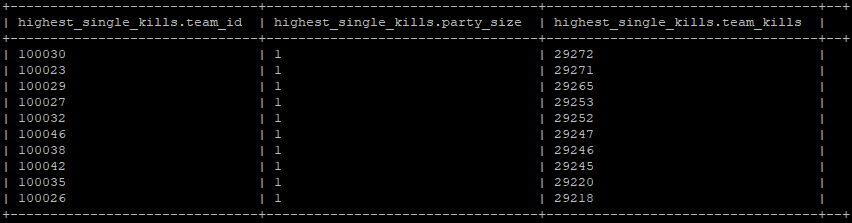
FROM cln\_agg\_0

WHERE party\_size = 1

GROUP BY team\_id, party\_size

ORDER BY team\_kills DESC

LIMIT 10;



13. **Query to find out REASONS FOR DEATH OF A PLAYER**

CREATE TABLE **Reason\_for\_death**

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION '/user/sgontya/Group4/Reason\_for\_death'

AS

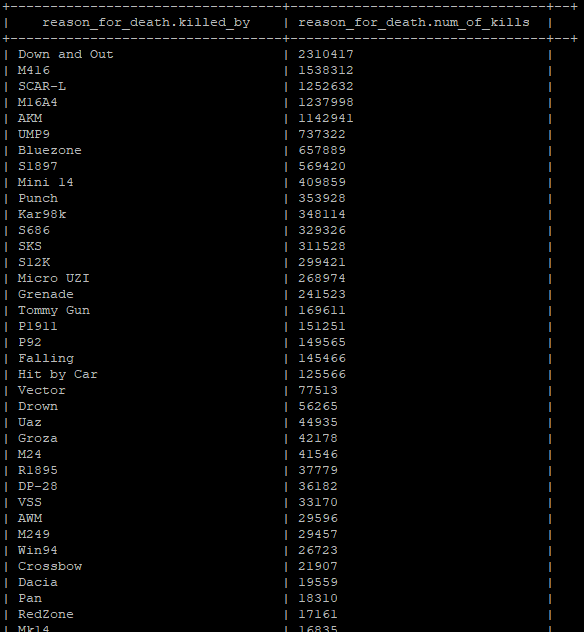
SELECT killed\_by, COUNT(killed\_by) AS Num\_of\_kills

FROM cln\_kills\_0

GROUP BY killed\_by

HAVING COUNT(killed\_by) > 1

ORDER BY Num\_of\_kills DESC;



**Step 8: Exporting data from HIVE**

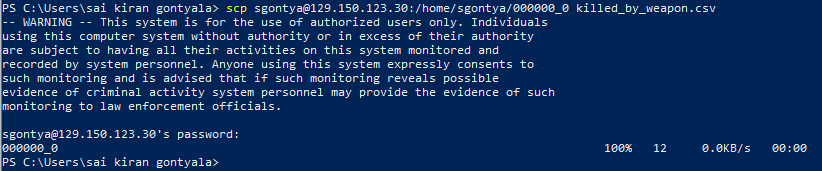
Use the below command to export the table from HDFS to Linux server,

hdfs dfs **-get** Group4/killed\_by\_weapon /000000\_0



Open Windows power shell. Use the command showed below to export the data to the base machine in the form of .csv format

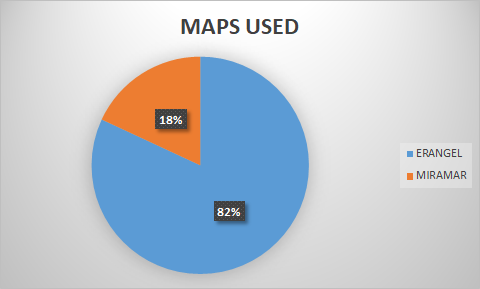
**pscp** -pw sgontya sgontya@129.150.123.30:/home/sgontya/000000\_0 killed\_by\_weapon.csv



**NOTE:** Replace **killed\_by\_weapon** with other file names. And to download csv files of other analyzed tables, first remove 000000\_0 in hdfs using **hdfs dfs -rm 000000\_0** command. Then get respective 000000\_0 of required table and download it to local file using **pscp.**

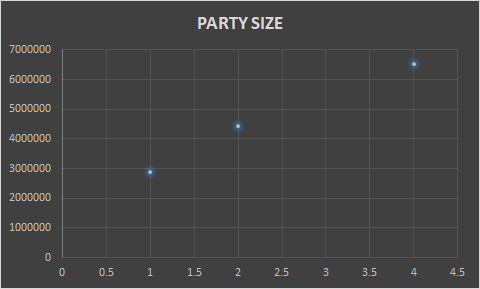
Step 9: Data Visualization using PowerBI, Tableau and Excel

1. Open your PowerBI and upload the **map\_location.csv** file.



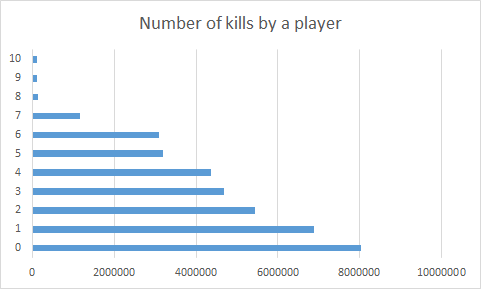
The pie chart shows the map locations players are interested in.

1. Open your Tableau to connect your server. You need to select **Text File** to open the file **party\_size.csv**



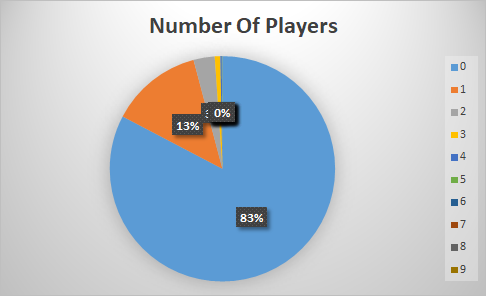
The above chart shows the party size players are interested in.

1. Open your PowerBI and upload the **players\_kill.csv** file.



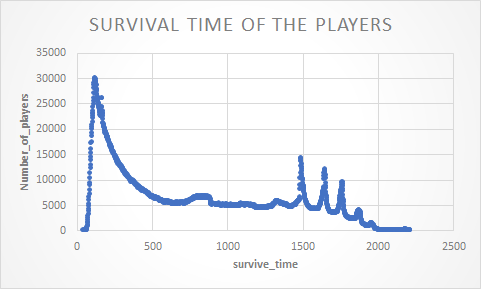
The above graph shows the number of kills by a player.

1. Open your PowerBI and upload the **player\_assists.csv** file.



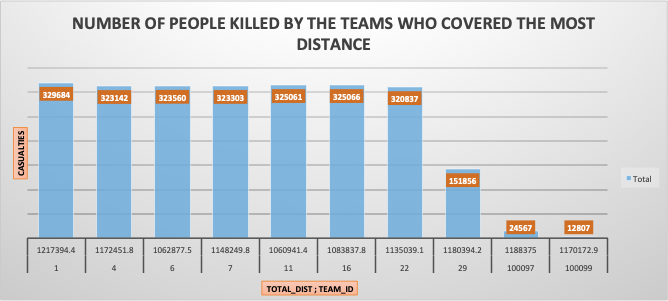
The pie chart shows the assist points scored by the players.

1. Open your PowerBI and upload the **survival\_time.csv** file.



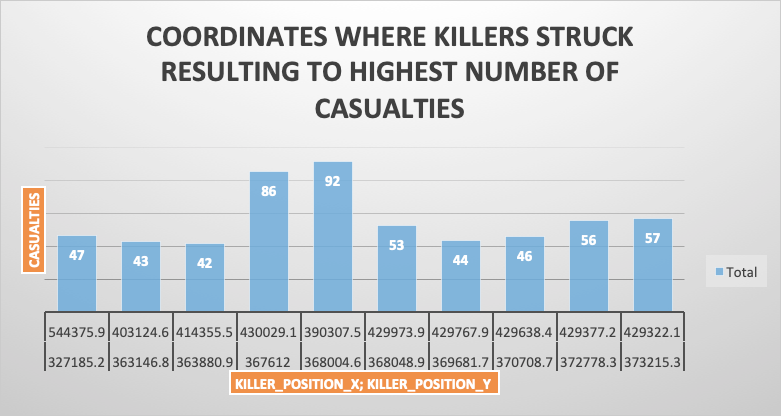
The above graph shows the survival time by different players.

1. Open your PowerBI and upload the dist\_covered\_team**.csv**.



The above graph shows the highest distance covered by teams.

1. Open your PowerBI and upload the **killers\_coordinates.csv.**



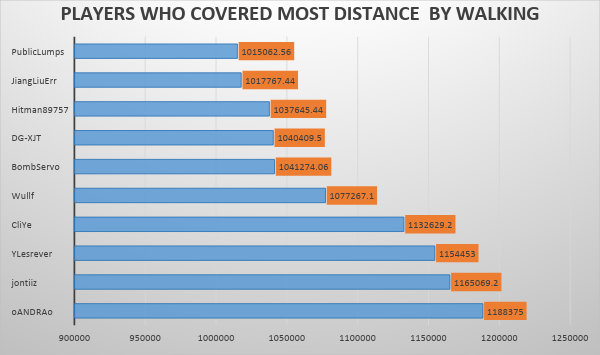
The above graph shows the coordinates where killers struck resulting to highest number of casualities.

1. Open your PowerBI and upload the **victim\_coordinates.csv.**



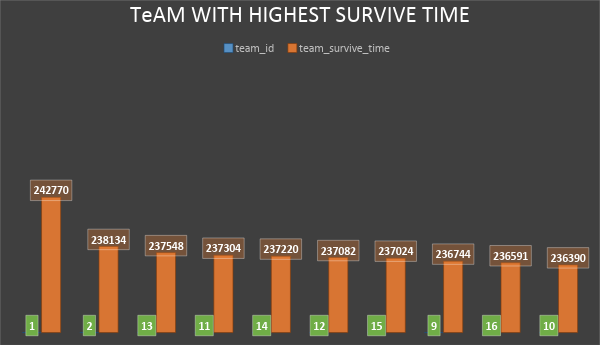
This bar chart shows the coordinates where victims were killed the most.

1. Open your PowerBI and upload the **distance\_on\_foot.csv.**



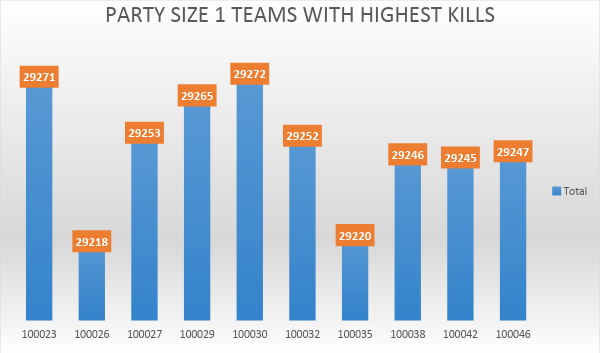
This chart shows players who covered most distance by walking.

1. Open your PowerBI and upload the **highest\_survivsl\_time.csv.**



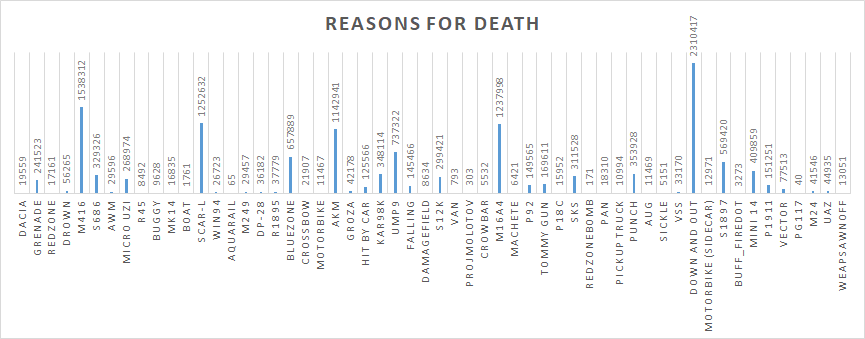
This bar chart shows the team with highest survival time.

1. Open your PowerBI and upload the **highest\_single\_kills.csv**



The above chart shows the party size 1 teams with highest kills.

1. Open your PowerBI and upload the **reason\_for\_death.csv**



The above bar chart shows the reasons for death of different players.

References

**https://github.com/sgontya/hiveanalysisonpubgdeathrate**

**https://www.kaggle.com/skihikingkevin/pubg-match-deaths**

[**https://www.pocket-lint.com/games/news/144028-what-is-pubg-mobile-and-why-is-everyone-talking-about-playerunknown-s-battlegrounds**](https://www.pocket-lint.com/games/news/144028-what-is-pubg-mobile-and-why-is-everyone-talking-about-playerunknown-s-battlegrounds)

[**https://store.steampowered.com/app/578080/PLAYERUNKNOWNS\_BATTLEGROUNDS/**](https://store.steampowered.com/app/578080/PLAYERUNKNOWNS_BATTLEGROUNDS/)