**PIG Use Case: Pokemon Data Analysis**

<https://acadgild.com/blog/pig-use-case-pokemon-data-analysis>

Copying Data

[cloudera@quickstart pig]$ cd /home/cloudera/

[cloudera@quickstart ~]$ cd pig

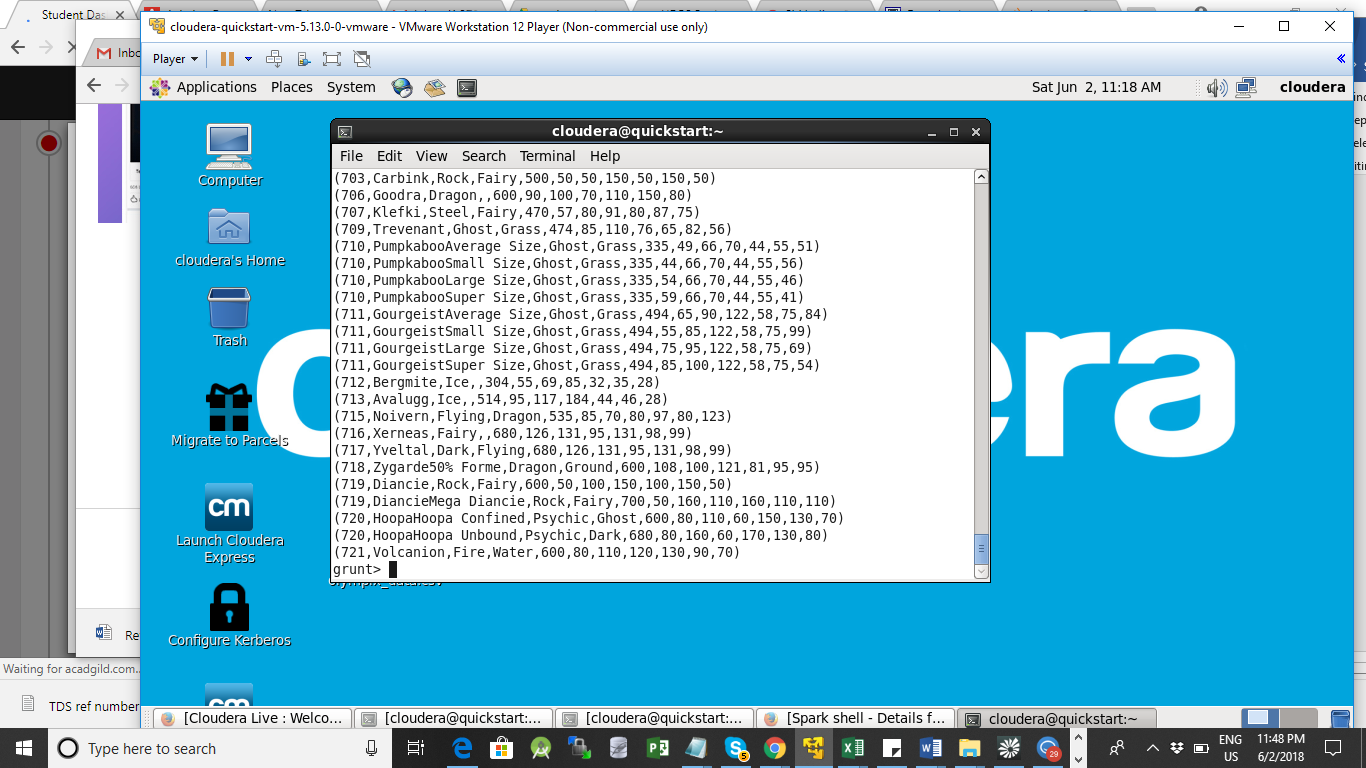
[cloudera@quickstart pig]$ mv ../Desktop/Pokemon.csv .

grunt> Load\_Data= LOAD 'Pokemon.csv' using PigStorage(',') ;

Load\_Data = LOAD 'Pokemon.csv' using PigStorage(',') AS(Sno:int,Name:chararray,Type1:chararray,Type2:chararray,Total:int,HP:int,Attack:int,Defense:int,SpAtk:int,SpDef:int,Speed:int);

**Ques 1: Find the list of players that have been selected in the qualifying round (DEFENCE>55).**

selected\_list = FILTER Load\_Data BY Defense>55;



**Ques 2: State the number of players taking part in the competition after getting selected in the qualifying round.**

**Explanation**:  
*Command*  
gourp\_selcted\_list = Group selected\_list All;

grunt> gourp\_selcted\_list = Group selected\_list All;

grunt> count\_selcted\_list = foreach gourp\_selcted\_list GENERATE COUNT(selected\_list);

dump count\_selected\_list;

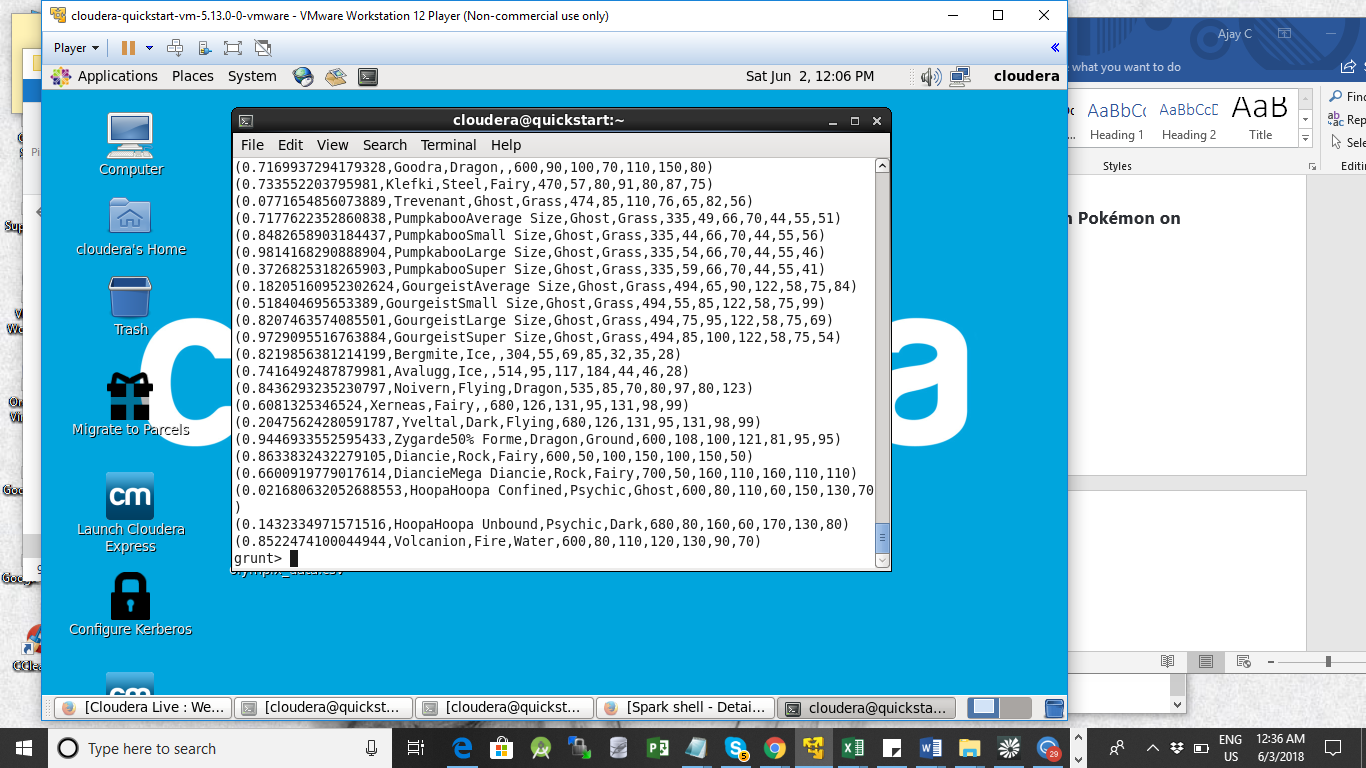
ne.util.MapRedUtil - Total input paths to process : 1

**(544)**

**Ques 3: Using random() generate random numbers for each Pokémon on the selected list.**

**Explanation**:

grunt> random\_include1 = foreach selected\_list GENERATE RANDOM(),Name,Type1,Type2,Total,HP,Attack,Defense,SpAtk,SpDef,Speed;

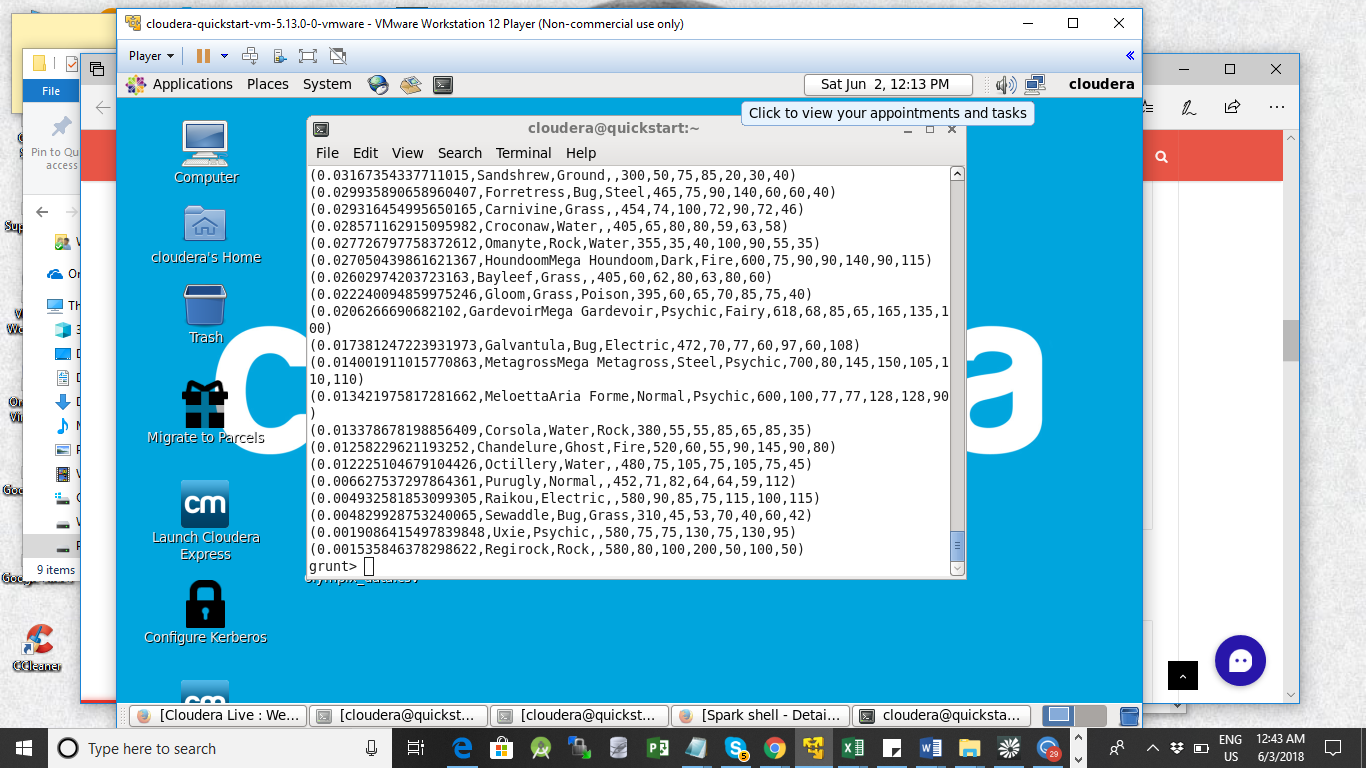


**Ques 4: Arrange the new list in a descending order according to a column randomly.**

**Explanation**: This will give us consequently a layer arranged to pick the random list which 1st player will choose.

random1\_desending = ORDER random\_include1 BY $0 DESC;

grunt> random1\_desending = ORDER random\_include1 BY $0 DESC;



#### Ques 5: Now on a new relation again associate random numbers for each Pokémon and arrange in descending order according to column random.

**Explanation**: We will be repeating above two steps again to form the 2nd list.

grunt> random\_include2 = foreach selected\_list GENERATE RANDOM(),Name,Type1,Type2,Total,HP,Attack,Defense,SpAtk,SpDef,Speed;

grunt> random2\_desending = ORDER random\_include2 BY $0 DESC;

limit\_data\_random1\_desending = LIMIT random1\_desending 5 ;  
limit\_data\_random2\_desending = LIMIT random2\_desending 5 ;

grunt> limit\_data\_random1\_desending = LIMIT random1\_desending 5 ;

grunt> limit\_data\_random2\_desending = LIMIT random2\_desending 5 ;

dump limit\_data\_random1\_desending;

**Output (Team 1)**

(0.99537203471002,Vanilluxe,Ice,,535,71,95,85,110,95,79)

(0.9902201348328671,Clawitzer,Water,,500,71,73,88,120,89,59)

(0.9883270270004234,Umbreon,Dark,,525,95,65,110,60,130,65)

(0.9865921172383645,Tirtouga,Water,Rock,355,54,78,103,53,45,22)

(0.9860283452341607,Avalugg,Ice,,514,95,117,184,44,46,28)

grunt> dump limit\_data\_random2\_desending ;

**Output (Team 2)**

(0.9983620946620061,DiancieMega Diancie,Rock,Fairy,700,50,160,110,160,110,110)

(0.9922490177271954,Armaldo,Rock,Bug,495,75,125,100,70,80,45)

(0.97556306987006,Herdier,Normal,,370,65,80,65,35,65,60)

(0.9746247314268183,Eelektross,Electric,,515,85,115,80,105,80,50)

(0.9684402264337542,GroudonPrimal Groudon,Ground,Fire,770,100,180,160,150,90,90)

#### Ques: Store the data on a local drive to announce for the final match. By the name player1 and player2 (only show the NAME and HP).

**Explanation**:  
*Commands*  
filter\_only\_name1 = foreach limit\_data\_random1\_desending Generate ($1,HP);

grunt> filter\_only\_name1 = foreach limit\_data\_random1\_desending Generate ($1,HP);

grunt> filter\_only\_name2 = foreach limit\_data\_random2\_desending Generate ($1,HP);

grunt> dump filter\_only\_name1 ;

((Tentacruel,80))

((Druddigon,77))

((Mesprit,80))

((Regice,80))

((KeldeoOrdinary Forme,91))

grunt> dump filter\_only\_name2 ;

((Slowking,95))

((Swadloon,55))

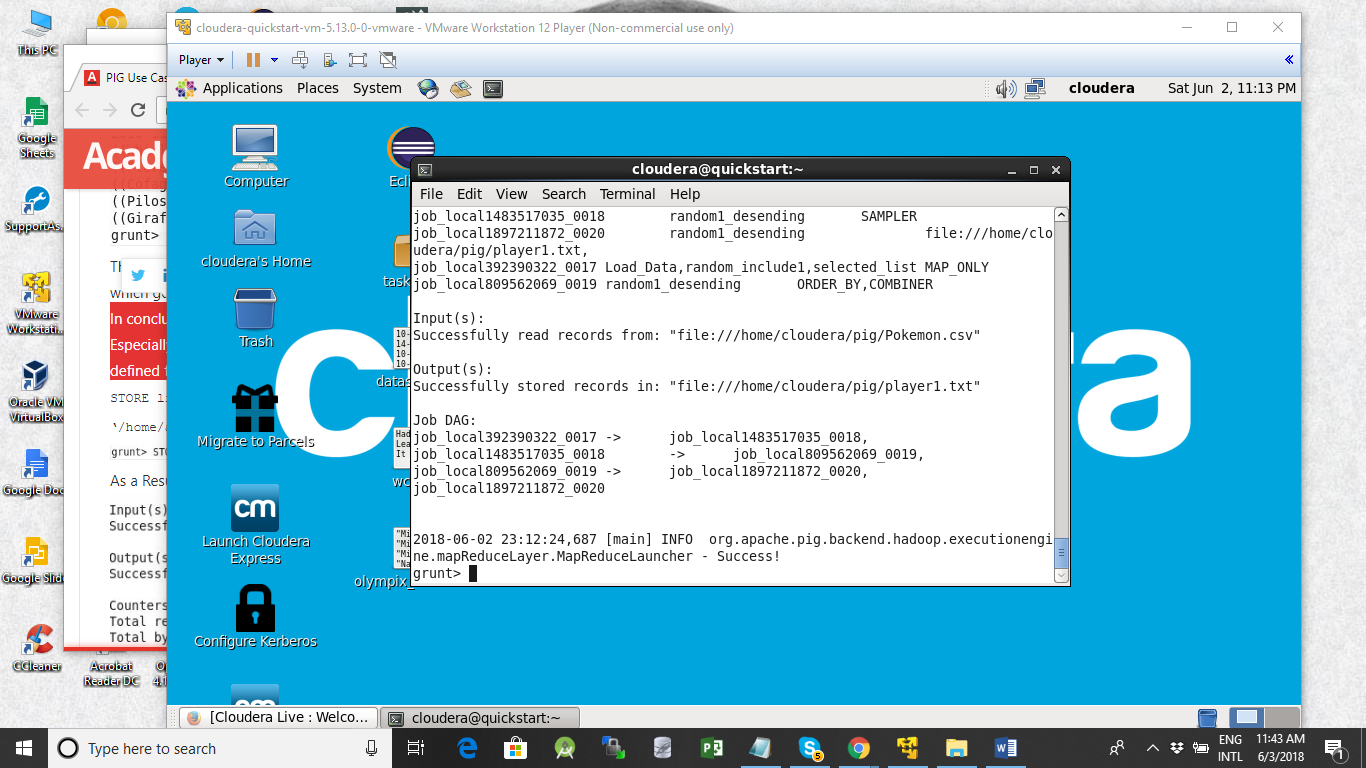
((GourgeistLarge Size,75))

((Gardevoir,68))

((Gligar,65))

In conclusion, let’s store this result in our local system  so we can use it as input to our next blog. Especially relevant where we will see UDF using PIG and calculations will be done through user-defined formulas.

STORE limit\_data\_random1\_desending INTO ‘player1.txt’;



grunt> STORE limit\_data\_random1\_desending INTO 'player2.txt' ;

