ASSIGNMENT 5.3

PIG Use Case: Pokemon Data Analysis:

The Pokémon Fight League (PFL) management for the 2017 match has first of all decided a minimum criterion for the entry selection process that filters through the defense power for any Pokémon, which should ideally be greater than 55.

Hence, the eligible list will be randomly formed after filtering out the Pokémons with a defense less than 55. Furthermore, our job is to give 2 list of names of those Pokémons who will be eligible for taking part in PFL this year from the list of all the participating 800 Pokémons.

Let's load the dataset inside PIG. We can either use the local mode or the MR mode. Here consequently, we will be using the local mode.

Command:

Load_Data = LOAD '/home/acadgild/pokemon_usecase/Pokemon.csv' USING PigStorage(',') AS (Sno:int, Name:chararray, Type1:chararray, Type2:chararray, Total:int, HP:int, Attack:int, Defense:int, Sp_Atk:int, Sp_Def, Speed);

```
grunt> Load_Data = LOAD '/home/acadgild/pokemon_usecase/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);
2017-12-11 07:32:12,048 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker.persist.jobstatus.
hours is deprecated. Instead, use mapreduce.jobtracker.persist.jobstatus.hours
2017-12-11 07:32:12,048 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.heartbeats.in.second is deprec
ated. Instead, use mapreduce.jobtracker.heartbeats.in.second
2017-12-11 07:32:12,048 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - jobclient.completion.poll.interval is
 deprecated. Instead, use mapreduce.client.completion.pollinterval
2017-12-11 07:32:12,048 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.tasktracker.tasks.sleeptime-be
fore-sigkill is deprecated. Instead, use mapreduce.tasktracker.tasks.sleeptimebeforesigkill
2017-12-11 07:32:12,048 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker.http.address is de
precated. Instead, use mapreduce.jobtracker.http.address
2017-12-11 07:32:12,048 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.skip.map.max.skip.records is d
```

Output: DUMP Load_Data;

```
(712,Bergmite,Ice,,304,55,69,85,32,35,28)
(713,Avalugg,Ice,,514,95,117,184,44,46,28)
(714,Noibat,Flying,Dragon,245,40,30,35,45,40,55)
(715,Noivern,Flying,Dragon,535,85,70,80,97,80,123)
(716,Xerneas,Fairy,,680,126,131,95,131,98,99)
(717,Yveltal,Dark,Flying,680,126,131,95,131,98,99)
(718,Zygarde50% Forme,Dragon,Ground,600,108,100,121,81,95,95)
(719,Diancie,Rock,Fairy,600,50,100,150,100,150,50)
(719,DiancieMega Diancie,Rock,Fairy,700,50,160,110,160,110,110)
(720,HoopaHoopa Confined,Psychic,Ghost,600,80,110,60,150,130,70)
(720,HoopaHoopa Unbound,Psychic,Dark,680,80,160,60,170,130,80)
(721,Volcanion,Fire,Water,600,80,110,120,130,90,70)
grunt> DESCRIBE Load_Data;
Load_Data: {Sno: int,Name: chararray,Type1: chararray,Type2: chararray,Total: int,HP: int,Attack: int,Defense: int,SpAtk: int
,SpDef: int,Speed: int}
```

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

Command:

selected_list = FILTER Load_Data BY Defense>55;

```
grunt> selected_list = FILTER Load_Data BY Defense>55;
grunt> DUMP selected_list;
```

The dataset is filtered, and hence out of all the 800 Pokémons, only 544 are eligible to take part in the tournament. In order to get the count, refer the next problem statement.

Output: DUMP selected_list;

```
(711,GourgeistSuper Size,Ghost,Grass,494,85,100,122,58,75,54)
(712,Bergmite,Ice,,304,55,69,85,32,35,28)
(713,Avalugg,Ice,,514,95,117,184,44,46,28)
(715,Noivern,Flying,Dragon,535,85,70,80,97,80,123)
(716,Xerneas,Fairy,,680,126,131,95,131,98,99)
(717,Yveltal,Dark,Flying,680,126,131,95,131,98,99)
(718,Zygarde50% Forme,Dragon,Ground,600,108,100,121,81,95,95)
(719,Diancie,Rock,Fairy,600,50,100,150,100,150,50)
(719,DiancieMega Diancie,Rock,Fairy,700,50,160,110,160,110,110)
(720,HoopaHoopa Confined,Psychic,Ghost,600,80,110,60,150,130,70)
(720,HoopaHoopa Unbound,Psychic,Dark,680,80,160,60,170,130,80)
(721,Volcanion,Fire,Water,600,80,110,120,130,90,70)
grunt> DESCRIBE selected_list;
selected_list: {Sno: int,Name: chararray,Type1: chararray,Type2: chararray,Total: int,HP: int,Attack: int,Defense: int,SpAtk:
 int,SpDef: int,Speed: int}
grunt>
```

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

Command:

group_selcted_list = Group selected_list All;

count_selcted_list = foreach group_selcted_list GENERATE COUNT(selected_list);

```
grunt> group_selcted_list = Group selected_list All;
grunt> count_selcted_list = foreach group_selcted_list GENERATE COUNT(selected_list);
```

Output: DUMP count_selcted_list;

```
2017-12-11 09:09:39,249 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2017-12-11 09:09:39,249 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to pro
cess : 1
(544)
grunt> DESCRIBE count_selcted_list;
count_selcted_list: {long}
grunt>
```

So, All the 544 players taking part will be alphabetically arranged and two teams of 5 Pokémons need to be extracted out randomly from the earlier list.

Seems like, this way we will have 2 lists containing 5 Pokémon each so to fight each other.

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

Command:

random_include1 = foreach selected_list GENERATE RANDOM(), Name, Type1, Type2, Total, HP, Attack, Defense, Sp_Atk, Sp_Def, Speed;

```
grunt> random_include1 = FOREACH selected_list GENERATE RANDOM(),Name,Type1,Type2,Total,HP,Attack,Defense,SpAtk,SpDef,Speed;
grunt> DUMP random_include1;
```

Hence sample for the list after adding random numbers:

Output: DUMP random_include1;

```
(0.6243759525041724,GourgeistSuper Size,Ghost,Grass,494,85,100,122,58,75,54)
(0.301607482666461,Bergmite,Ice,,304,55,69,85,32,35,28)
(0.12379636167031516,Avalugg,Ice,,514,95,117,184,44,46,28)
(0.6332172777176096,Noivern,Flying,Dragon,535,85,70,80,97,80,123)
(0.7365058368399285,Xerneas,Fairy,,680,126,131,95,131,98,99)
(0.8172176102535874,Yveltal,Dark,Flying,680,126,131,95,131,98,99)
(0.1761181942555542,Zygarde50% Forme,Dragon,Ground,600,108,100,121,81,95,95)
(0.014813224665234603,Diancie,Rock,Fairy,600,50,100,150,100,150,50)
(0.8845672873910095,DiancieMega Diancie,Rock,Fairy,700,50,160,110,160,110,110)
(0.37086364753992096,HoopaHoopa Confined,Psychic,Ghost,600,80,110,60,150,130,70)
(0.04645417484138881,HoopaHoopa Unbound,Psychic,Dark,680,80,160,60,170,130,80)
(0.6999666512985456,Volcanion,Fire,Water,600,80,110,120,130,90,70)
grunt> DESCRIBE random_include1;
random_include1: {org.apache.pig.builtin.random_13: double,Name: chararray,Type1: chararray,Type2: chararray,Total: int,HP: i
nt,Attack: int,Defense: int,SpAtk: int,SpDef: int,Speed: int}
grunt>
```

Question 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.

Command:

random1_desending = ORDER random_include1 BY $0 DESC;

```
(0.301607482666461,Bergmite,Ice,,304,55,69,85,32,35,28)
(0.12379636167031516,Avalugg,Ice,,514,95,117,184,44,46,28)
(0.6332172777176096,Noivern,Flying,Dragon,535,85,70,80,97,80,123)
(0.7365058368399285,Xerneas,Fairy,,680,126,131,95,131,98,99)
(0.8172176102535874,Yveltal,Dark,Flying,680,126,131,95,131,98,99)
(0.1761181942555542,Zygarde50% Forme,Dragon,Ground,600,108,100,121,81,95,95)
(0.014813224665234603,Diancie,Rock,Fairy,600,50,100,150,100,150,50)
(0.8845672873910095,DiancieMega Diancie,Rock,Fairy,700,50,160,110,160,110,110)
(0.37086364753992096,HoopaHoopa Confined,Psychic,Ghost,600,80,110,60,150,130,70)
(0.04645417484138881,HoopaHoopa Unbound,Psychic,Dark,680,80,160,60,170,130,80)
(0.6999666512985456,Volcanion,Fire,Water,600,80,110,120,130,90,70)
grunt> DESCRIBE random_include1;
random_include1: {org.apache.pig.builtin.random_13: double,Name: chararray,Type1: chararray,Type2: chararray,Total: int,HP: i
nt,Attack: int,Defense: int,SpAtk: int,SpDef: int,Speed: int}
grunt> random1_desending = ORDER random_include1 BY $0 DESC;
grunt> DUMP random1_desending;
```

Output: DUMP random1_desending;

```
(0.04208686126480177,Yanmega,Bug,Flying,515,86,76,86,116,56,95)
(0.03756128673252479,Venusaur,Grass,Poison,525,80,82,83,100,100,80)
(0.03588310615676105,AggronMega Aggron,Steel,,630,70,140,230,60,80,50)
(0.03469682624096926,ScizorMega Scizor,Bug,Steel,600,70,150,140,65,100,75)
(0.03063419154339253,Claydol,Ground,Psychic,500,60,70,105,70,120,75)
(0.029642757853089563,Servine,Grass,,413,60,60,75,60,75,83)
(0.017033449026275793,Dewott,Water,,413,75,75,60,83,60,60)
(0.01681047512608469,Cottonee,Grass,Fairy,280,40,27,60,37,50,66)
(0.016196843119121396,AltariaMega Altaria,Dragon,Fairy,590,75,110,110,110,105,80)
(0.013908365108518339,Vanilluxe,Ice,,535,71,95,85,110,95,79)
(0.012465404520018541,PidgeotMega Pidgeot,Normal,Flying,579,83,80,80,135,80,121)
(0.005534418074599645,Klink,Steel,,300,40,55,70,45,60,30)
(0.0035980137980614613,Arbok,Poison,,438,60,85,69,65,79,80)
(0.0034642897209260504,Darkrai,Dark,,600,70,90,90,135,90,125)
grunt> DESCRIBE random1_desending;
random1_desending: {org.apache.pig.builtin.random_26: double,Name: chararray,Type1: chararray,Type2: chararray,Total: int,HP:
 int,Attack: int,Defense: int,SpAtk: int,SpDef: int,Speed: int}
grunt>
```

Yet we want 1 more list with random arrangements of Pokémons which will be therefore chosen by the 2nd player later on.

Question 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.

Command:

random_include2 = FOREACH selected_list GENERATE RANDOM(), Name, Type1, Type2, Total, HP, Attack, Defense, Sp_Atk, Sp_Def, Speed;

random2_desending = ORDER random_include2 BY $0 DESC;

```
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;
grunt> DUMP random2_desending;
```

Hence sample for the list.

Output: DUMP random2_desending;

```
(0.013088837473008597,Mesprit,Psychic,,580,80,105,105,105,105,80)
(0.012953379648325214,Omanyte,Rock,Water,355,35,40,100,90,55,35)
(0.011996919516167881,Garchomp,Dragon,Ground,600,108,130,95,80,85,102)
(0.010461182366103494,Tranquill,Normal,Flying,358,62,77,62,50,42,65)
(0.010331993891446012,Delphox,Fire,Psychic,534,75,69,72,114,100,104)
(0.008826902888022903,Sandshrew,Ground,,300,50,75,85,20,30,40)
(0.008755013195242745,Audino,Normal,,445,103,60,86,60,86,50)
(0.008531436519653712,Mewtwo,Psychic,,680,106,110,90,154,90,130)
(0.008344311593311726,Phione,Water,,480,80,80,80,80,80,80)
(0.006565030696500496,Persian,Normal,,440,65,70,60,65,65,115)
(0.004929166340346014,Sudowoodo,Rock,,410,70,100,115,30,65,30)
(0.004068599875750145,Qwilfish,Water,Poison,430,65,95,75,55,55,85)
(0.0010923218294142112,Samurott,Water,,528,95,100,85,108,70,70)
grunt> DESCRIBE random2_desending;
random2_desending: {org.apache.pig.builtin.random_94: double,Name: chararray,Type1: chararray,Type2: chararray,Total: int,HP:
 int,Attack: int,Defense: int,SpAtk: int,SpDef: int,Speed: int}
grunt>
```

Now, especially relevant selecting the top 5.

Question 6: From the two different descending lists of random Pokémons, select the top 5 Pokémons for 2 different players.

Commands:

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 ;
grunt> DUMP limit_data_random1_desending;
```

Hence sample for the list:

Output 1: DUMP limit_data_random1_desending;

```
2017-12-11 09:49:34,368 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2017-12-11 09:49:34,368 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to pro
cess : 1
(0.999112416316631,Slowpoke,Water,Psychic,315,90,65,65,40,40,15)
(0.9983508953225769,Chespin,Grass,,313,56,61,65,48,45,38)
(0.9971078814063578,WormadamPlant Cloak,Bug,Grass,424,60,59,85,79,105,36)
(0.9969881249411938,Solrock,Rock,Psychic,440,70,95,85,55,65,70)
(0.988529187163317,CharizardMega Charizard X,Fire,Dragon,634,78,130,111,130,85,100)
grunt> DESCRIBE limit_data_random1_desending;
limit_data_random1_desending: {org.apache.pig.builtin.random_137: double,Name: chararray,Type1: chararray,Type2: chararray,To
tal: int,HP: int,Attack: int,Defense: int,SpAtk: int,SpDef: int,Speed: int}
grunt>
```

Output 2: DUMP limit_data_random2_desending;

```
2017-12-11 09:51:00,898 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2017-12-11 09:51:00,898 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to pro
cess : 1
(0.9996956666801585,Serperior,Grass,,528,75,75,95,75,95,113)
(0.9993085587247786,Phanpy,Ground,,330,90,60,60,40,40,40)
(0.9935911902876993,Articuno,Ice,Flying,580,90,85,100,95,125,85)
(0.9915389516420899,Snorlax,Normal,,540,160,110,65,65,110,30)
(0.9881292243504103,Fearow,Normal,Flying,442,65,90,65,61,61,100)
grunt> DESCRIBE limit_data_random2_desending;
limit_data_random2_desending: {org.apache.pig.builtin.random_150: double,Name: chararray,Type1: chararray,Type2: chararray,To
tal: int,HP: int,Attack: int,Defense: int,SpAtk: int,SpDef: int,Speed: int}
grunt>
```

Question 7: 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).

Commands:

filter_only_name1 = foreach limit_data_random1_desending Generate ($1, HP);

filter_only_name2 = foreach limit_data_random2_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;
```

Since for Player1 we have:

Output 1: DUMP filter_only_name1;

```
2017-12-11 09:57:38,425 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2017-12-11 09:57:38,425 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to pro
cess : 1
((Vespiquen,70))
((Nincada,31))
((Bisharp,65))
((Squirtle,44))
((Herdier,65))
grunt> DESCRIBE filter_only_name1;
filter_only_name1: {org.apache.pig.builtin.totuple_HP_341: (Name: chararray,HP: int)}
grunt>
```

Since for Player2 we have:

Output 2: DUMP filter_only_name2;

```
2017-12-11 09:58:52,474 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2017-12-11 09:58:52,474 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to pro
cess : 1
((Beldum,40))
((PumpkabooSuper Size,59))
((Probopass,60))
((Marshtomp,70))
((KeldeoOrdinary Forme,91))
grunt> DESCRIBE filter_only_name2;
filter_only_name2: {org.apache.pig.builtin.totuple_HP_373: (Name: chararray,HP: int)}
grunt>
```

In conclusion, let's store this result in our local system.

STORE limit_data_random1_desending INTO '/home/acadgild/ pokemon_usecase/player1.txt';

Verification:

```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.2.0   0.14.0  acadgild        2017-12-11 10:09:01     2017-12-11 10:09:05     ORDER_BY,FILTER,LIMIT

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduceTime   MinReduceTime   AvgRe
duceTime        MedianReducetime        Alias   Feature Outputs
job_local1277839803_0047        1       1       n/a     n/a     n/a     n/a     n/a     n/a     n/a     n/a     random1_desen
ding    SAMPLER
job_local166459461_0049 1       1       n/a     n/a     n/a     n/a     n/a     n/a     n/a     n/a     random1_desending   /
home/acadgild/pokemon_usecase/player1.txt,
job_local853062016_0048 1       1       n/a     n/a     n/a     n/a     n/a     n/a     n/a     n/a     random1_desending   O
RDER_BY,COMBINER
job_local857434701_0046 1       0       n/a     n/a     n/a     n/a     0       0       0       0       Load_Data,random_incl
ude1,selected_list      MAP_ONLY

Input(s):
Successfully read 801 records from: "/home/acadgild/pokemon_usecase/Pokemon.csv"

Output(s):
Successfully stored 5 records in: "/home/acadgild/pokemon_usecase/player1.txt"

Counters:
Total records written : 5
Total bytes written : 0
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_local857434701_0046 ->      job_local1277839803_0047,
job_local1277839803_0047        ->      job_local853062016_0048,
job_local853062016_0048 ->      job_local166459461_0049,
job_local166459461_0049
```

STORE limit_data_random2_desending INTO ‘/home/acadgild/ pokemon_usecase/player2.txt’;

Verification:

```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.2.0   0.14.0  acadgild        2017-12-11 10:10:35     2017-12-11 10:10:39     ORDER_BY,FILTER,LIMIT

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduceTime   MinReduceTime   AvgRe
duceTime        MedianReducetime        Alias   Feature Outputs
job_local1051787510_0052        1       1       n/a     n/a     n/a     n/a     n/a     n/a     n/a     n/a     random2_desen
ding    ORDER_BY,COMBINER
job_local1200377808_0051        1       1       n/a     n/a     n/a     n/a     n/a     n/a     n/a     n/a     random2_desen
ding    SAMPLER
job_local1430098404_0050        1       0       n/a     n/a     n/a     n/a     0       0       0       0       Load_Data,ran
dom_include2,selected_list      MAP_ONLY
job_local347243476_0053 1       1       n/a     n/a     n/a     n/a     n/a     n/a     n/a     n/a     random2_desending   /
home/acadgild/pokemon_usecase/player2.txt,

Input(s):
Successfully read 801 records from: "/home/acadgild/pokemon_usecase/Pokemon.csv"

Output(s):
Successfully stored 5 records in: "/home/acadgild/pokemon_usecase/player2.txt"

Counters:
Total records written : 5
Total bytes written : 0
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_local1430098404_0050        ->      job_local1200377808_0051,
job_local1200377808_0051        ->      job_local1051787510_0052,
job_local1051787510_0052        ->      job_local347243476_0053,
job_local347243476_0053
```