

PIG Use Case: Pokemon Data Analysis

Note: Before running pig scripts, start all Hadoop Daemons with `start-all.sh` command, and check using `jps` command whether all daemons are running or not i.e. Namenode, Datanode, SecondaryNamenode, ResourceMangaer, NodeManager. There is no need to start `jobhistoryserver` because pig is launched in **LOCAL mode.**

What has to be done?

The Pokémon Fight League (PFL) management for the 2017 match has first of all decided a **minimum criterion for the entry selection process, i.e. defense power for any Pokémon should ideally be greater than 55**. Our job is to give 2 lists consisting names of those Pokémons who will be eligible for taking part in PFL this year from the list of all the participating 800 Pokémons.

In order to complete above specified task, below steps are followed:

Step 1: Launch pig in LOCAL mode using below command.

```
[acadgild@localhost pig]$ pig -x local
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hbase/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop-2.6.0/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
2017-08-14 01:43:14,249 INFO [main] pig.ExecTypeProvider: Trying ExecType : LOCAL
2017-08-14 01:43:14,251 INFO [main] pig.ExecTypeProvider: Picked LOCAL as the ExecType
2017-08-14 01:43:14,461 [main] INFO org.apache.pig.Main - Apache Pig version 0.14.0 (r1640057) compiled Nov 16 2014, 18:02:05
2017-08-14 01:43:14,465 [main] INFO org.apache.pig.Main - Logging error messages to: /home/acadgild/Documents/pig/pig_1502655194456.log
2017-08-14 01:43:14,781 [main] INFO org.apache.pig.impl.util.Utils - Default bootstrap file /home/acadgild/.pigbootstrap not found
2017-08-14 01:43:15,962 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
2017-08-14 01:43:15,963 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2017-08-14 01:43:15,972 [main] INFO org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to hadoop file system at: file:///
2017-08-14 01:43:15,999 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - mapred.used.genericoptionsparser is deprecated. Instead, use mapreduce.client.genericoptionsparser.used
2017-08-14 01:43:16,545 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
grunt> grunt shell
```

Step 2: Load the dataset inside PIG using **LOAD** command where **loadData** relation holds loaded data. Using **DESCRIBE** command we can check the schema of loadData relation and finally using **DUMP** command we can check whether data has been loaded correctly or not from local file system to loadData relation in pig.

```
grunt> loadData = LOAD '/home/acadgild/Documents/pig/pokemon_usecase/Pokemon.csv'
  USING PigStorage(',') AS (Sno:int,Name:chararray,Type1:chararray,Type2:chararray,
  Total:int,HP:int,Attack:int,Defense:int,SpAttack:int,SpDefence:int,Speed:int)
  ;
2017-08-14 01:52:53,933 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - mapreduce.job.counters.limit is deprecated. Instead, use mapreduce.job.counters.max
2017-08-14 01:52:53,933 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
2017-08-14 01:52:53,934 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
grunt> DESCRIBE loadData; ← command shows schema
loadData: {Sno: int,Name: chararray,Type1: chararray,Type2: chararray,Total: int,HP: int,Attack: int,Defense: int,SpAttack: int,SpDefence: int,Speed: int}
grunt> DUMP loadData; ← command displays data
```

Output of DUMP command last few records

```
(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>
```

Step 3: To find the list of players that have been selected in the qualifying round (**DEFENCE>55**), below command is used:

```
grunt> selected_list = FILTER loadData BY Defense>55; ←
grunt> DESCRIBE selected_list; ←
selected_list: {Sno: int,Name: chararray,Type1: chararray,Type2: chararray,Total: int,HP: int,Attack: int,Defense: int,SpAttack: int,SpDefence: int,Speed: int}
grunt> DUMP selected_list; ←
```

Explanation of above commands:

Line 1: **loadData** relation is filtered on the basis of **Defense>55** condition and stored in **selected_list** relation, and hence out of all the 800 Pokémons, only 544 are eligible to take part in the tournament.

Line 2: schema of **selected_list** relation is displayed.

Line 3: data inside **selected_list** relation is displayed.

**Output of dump command
last few records**

```
(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> █
```

In order to get the count, refer the next step.

Step 4: To state the number of players taking part in the competition after getting selected in the qualifying round, below command is used:

```
grunt> group_selected_list = GROUP selected_list ALL;
grunt> DESCRIBE group_selected_list;
group_selected_list: {group: chararray,selected_list: {(Sno: int,Name: chararray
,Type1: chararray,Type2: chararray>Total: int,HP: int,Attack: int,Defense: int,S
pAttack: int,SpDefence: int,Speed: int)}}
grunt> count_selected_list = FOREACH group_selected_list GENERATE COUNT(selected
_list) as countOfPlayers;
grunt> DESCRIBE count_selected_list;
count_selected_list: {countOfPlayers: long}
grunt> DUMP count_selected_list;█
```

Explanation of above commands:

Line 1: **GROUP ALL** command groups all the tuples of **selected_list** relation in one group, and the result is stored in **group_selected_list** relation.

Line 2: schema of **group_selected_list** relation is displayed, where we can see all tuples are grouped into one group.

Line 3: **count_selected_list** relation stores **COUNT** of selected players.

Line 4: schema of **count_selected_list** relation is displayed, where we can see only one field appears i.e. countOfPlayers.

Line 5: data inside **count_selected_list** relation is displayed.

```
2017-08-14 02:02:44,941 [main] WARN org.apache.pig.data.SchemaTupleBackend - SchemaTupleBackend has already been initialized
2017-08-14 02:02:45,071 [main] INFO org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2017-08-14 02:02:45,071 [main] INFO org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(544) ← Output of dump command
grunt> █
```

Now, two teams of 5 Pokémons need to be extracted out randomly from the selected list of players i.e. 544 players.

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

To find random list of players refer next step.

Step 5: Using random(), generate random numbers for each Pokémon in the selected list.

Below command is used to generate random players List 1:

```
grunt> random_include1= FOREACH selected_list GENERATE RANDOM(),Name,Type1,Type2,Total,HP,Attack,Defense,SpAttack,SpDefence,Speed;
grunt> DESCRIBE random_include1; ←
random_include1: {org.apache.pig.builtin.random 10: double,Name: chararray,Type1: chararray,Type2: chararray,Total: int,HP: int,Attack: int,Defense: int,SpAttack: int,SpDefence: int,Speed: int}
grunt> DUMP random_include1; █ ←
```

Explanation of above commands:

Line 1: generates random list of players and stores in **random_include1** relation.

Line 2: schema of **random_include1** relation is displayed, which shows one extra field is added, i.e. random and rest fields are same as that of loadData relation.

Line 3: shows data inside **random_include1** relation.

Random List 1 Last few records

```
(0.13832767445893335,Noivern,Flying,Dragon,535,85,70,80,97,80,123)
(0.4588000973295189,Xerneas,Fairy,,680,126,131,95,131,98,99)
(0.8127382414380713,Yveltal,Dark,Flying,680,126,131,95,131,98,99)
(0.08214356050708749,Zygarde50% Forme,Dragon,Ground,600,108,100,121,81,95,95)
(0.773632999543264,Diancie,Rock,Fairy,600,50,100,150,100,150,50)
(0.615353810594273,DiancieMega Diancie,Rock,Fairy,700,50,160,110,160,110,110)
(0.15234929898878813,HoopaHoopa Confined,Psychic,Ghost,600,80,110,60,150,130,70)
(0.5584642848284808,HoopaHoopa Unbound,Psychic,Dark,680,80,160,60,170,130,80)
(0.5248191431083262,Volcanion,Fire,Water,600,80,110,120,130,90,70)
grunt> █
```

Step 6: Arrange the random_include1 list in a descending order according to a random column i.e. first column using below command:

```
grunt> random1_desc = ORDER random_include1 BY $0 DESC; ←
grunt> DESCRIBE random1_desc; ←
random1_desc: {org.apache.pig.builtin.random_32: double,Name: chararray,Type1: chararray,Type2: chararray,Total: int,HP: int,Attack: int,Defense: int,SpAttack: int,SpDefence: int,Speed: int}
grunt> DUMP random1_desc; █ ←
```

Explanation of above commands:

Line 1: gives us consequently a layer arrangement to pick the player from random list.

Line 2: schema of relation random1_desc created at line 1 is displayed.

Line 3: data of relation is displayed.

Random List1 in descending order Last few records

```
(0.01927836925467119,Dragonair,Dragon,,420,61,84,65,70,70,70)
(0.018941829328859905,Xerneas,Fairy,,680,126,131,95,131,98,99)
(0.01582743286566146,Linoone,Normal,,420,78,70,61,50,61,100)
(0.013090852338845771,Sableye,Dark,Ghost,380,50,75,75,65,65,50)
(0.010801763676269616,PinsirMega Pinsir,Bug,Flying,600,65,155,120,65,90,105)
(0.010212223441448343,Gloom,Grass,Poison,395,60,65,70,85,75,40)
(0.009696831977590081,Vullaby,Dark,Flying,370,70,55,75,45,65,60)
(0.006979637720072485,Serperior,Grass,,528,75,75,95,75,95,113)
(0.004083284950000654,Golurk,Ground,Ghost,483,89,124,80,55,80,55)
(0.0032043197025455328,Zapdos,Electric,Flying,580,90,90,85,125,90,100)
grunt> █
```

So, using Step 5 and Step 6 first random list is created from which first player to fight will be selected, likewise we can create second random list from which second player for fight will be selected. For this refer Step 7.

Step 7: Now on a new relation again associate random numbers for each Pokémon and arrange in descending order according to a random column, below commands are used to do this:

```
grunt> random_include2= FOREACH selected_list GENERATE RANDOM(),Name,Type1,Type2,Total,HP,Attack,Defense,Sp
Attack,SpDefence,Speed;
grunt> random2_desc = ORDER random_include2 BY $0 DESC;
grunt> DESCRIBE random2_desc;
random2_desc: {org.apache.pig.builtin.random_69: double,Name: chararray,Type1: chararray,Type2: chararray,T
otal: int,HP: int,Attack: int,Defense: int,SpAttack: int,SpDefence: int,Speed: int}
grunt> DUMP random2_desc;
```

Explanation: same as in Steps 5 and 6.

Random List 2 in descending order Last few records

```
(0.019770578824593765,LatiasMega Latias,Dragon,Psychic,700,80,100,120,140,150,110)
(0.018578719409788502,Onix,Rock,Ground,385,35,45,160,30,45,70)
(0.018075110570990094,Hariyama,Fighting,,474,144,120,60,40,60,50)
(0.016487549715120564,Mismagius,Ghost,,495,60,60,60,105,105,105)
(0.015438079301353547,GiratinaAltered Forme,Ghost,Dragon,680,150,100,120,100,120,90)
(0.014433692846380186,CharizardMega Charizard Y,Fire,Flying,634,78,104,78,159,115,100)
(0.014146778235864743,Furret,Normal,,415,85,76,64,45,55,90)
(0.013967479122486082,Blaziken,Fire,Fighting,530,80,120,70,110,70,80)
(0.009499934905737528,Seadra,Water,,440,55,65,95,95,45,85)
(0.0044138511202213015,Bronzong,Steel,Psychic,500,67,89,116,79,116,33)
(0.002579527299749418,Vanilluxe,Ice,,535,71,95,85,110,95,79)
grunt>
```

Now, we have to find top 5 players from each list, for this refer below steps.

Step 8: From the two different descending lists of random Pokémons, select the top 5 Pokémons for 2 different players, using below commands:

```
grunt> limit_data_random1_desc = LIMIT random1_desc 5;
grunt> DESCRIBE limit_data_random1_desc;
limit_data_random1_desc: {org.apache.pig.builtin.random_93: double,Name: chararray,Type1: chararray,Type2:
chararray,Total: int,HP: int,Attack: int,Defense: int,SpAttack: int,SpDefence: int,Speed: int}
grunt> DUMP limit_data_random1_desc;
```


Above lines of code find top 5 players from List 1. Explanation is as follows:

Line 1: stores top 5 players of first list in **limit_data_random1_desc** relation.

Line 2: shows schema of **limit_data_random1_desc** relation.

Line 3: displays data inside **limit_data_random1_desc** relation.

```
deprecated. Instead, use fs.defaultFS
2017-08-14 02:17:14,639 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - mapreduce.job.count
ers.limit is deprecated. Instead, use mapreduce.job.counters.max
2017-08-14 02:17:14,639 [main] WARN org.apache.pig.data.SchemaTupleBackend - SchemaTupleBackend has alread
y been initialized
2017-08-14 02:17:14,700 [main] INFO org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input pa
ths to process : 1
2017-08-14 02:17:14,700 [main] INFO org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total
input paths to process : 1
(0.9964653383521593,Galvantula,Bug,Electric,472,70,77,60,97,60,108)
(0.9912189350622649,Combusken,Fire,Fighting,405,60,85,60,85,60,55)
(0.9911658753739143,Zweilous,Dark,Dragon,420,72,85,70,65,70,58)
(0.9901066833640759,Lanturn,Water,Electric,460,125,58,58,76,76,67)
(0.9881200821785234,Cresselia,Psychic,,600,120,70,120,75,130,85)
grunt> █
```

Top 5 players of List 1

To find top 5 players from List 2, follow below lines of code:

```
grunt> limit_data_random2_desc = LIMIT random2_desc 5;
grunt> DESCRIBE limit_data_random2_desc;
limit_data_random2_desc: {org.apache.pig.builtin.random_118: double,Name: chararray,Type1: chararray,Type2:
chararray>Total: int,HP: int,Attack: int,Defense: int,SpAttack: int,SpDefence: int,Speed: int}
grunt> DUMP limit_data_random2_desc;
```

```
2017-08-14 02:18:43,713 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - mapreduce.job.count
ers.limit is deprecated. Instead, use mapreduce.job.counters.max
2017-08-14 02:18:43,713 [main] WARN org.apache.pig.data.SchemaTupleBackend - SchemaTupleBackend has alread
y been initialized
2017-08-14 02:18:43,752 [main] INFO org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input pa
ths to process : 1
2017-08-14 02:18:43,752 [main] INFO org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total
input paths to process : 1
(0.9984178021125091,ShayminSky Forme,Grass,Flying,600,100,103,75,120,75,127)
(0.9980357507673348,HeracrossMega Heracross,Bug,Fighting,600,80,185,115,40,105,75)
(0.9938188403536057,Turtwig,Grass,,318,55,68,64,45,55,31)
(0.992761491129575,Xerneas,Fairy,,680,126,131,95,131,98,99)
(0.9924005005125319,Simisage,Grass,,498,75,98,63,98,63,101)
grunt> █
```

Top 5 players of List 2

Step 9: 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).

Filtering List 1 to find top 5 players with Name and HP only, using below commands:

```
grunt> filter_only_name1 = FOREACH limit_data_random1_desc GENERATE ($1,HP); ←
grunt> DESCRIBE filter_only_name1; ←
filter_only_name1: {org.apache.pig.builtin.totuple_HP_153: (Name: chararray,HP: int)}
grunt> DUMP filter_only_name1; ←
```

```
2017-08-14 02:21:08,617 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - mapreduce.job.count
ers.limit is deprecated. Instead, use mapreduce.job.counters.max
2017-08-14 02:21:08,617 [main] WARN org.apache.pig.data.SchemaTupleBackend - SchemaTupleBackend has alread
y been initialized
2017-08-14 02:21:08,661 [main] INFO org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input pa
ths to process : 1
2017-08-14 02:21:08,661 [main] INFO org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total
input paths to process : 1
((Electabuzz,65))
((Huntail,55))
((Malamar,86))
((Dunsparce,100))
((Gastrodon,111))
```

Players of List 1

Filtering List 2 to find top 5 players with Name and HP only, using below commands:

```
grunt> filter_only_name2 = FOREACH limit_data_random2_desc GENERATE ($1,HP); ←
grunt> DESCRIBE filter_only_name2; ←
filter_only_name2: {org.apache.pig.builtin.totuple_HP_206: (Name: chararray,HP: int)}
grunt> DUMP filter_only_name2; ←
```

```
2017-08-14 02:24:34,035 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is
deprecated. Instead, use fs.defaultFS
2017-08-14 02:24:34,035 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - mapreduce.job.count
ers.limit is deprecated. Instead, use mapreduce.job.counters.max
2017-08-14 02:24:34,035 [main] WARN org.apache.pig.data.SchemaTupleBackend - SchemaTupleBackend has alread
y been initialized
2017-08-14 02:24:34,073 [main] INFO org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input pa
ths to process : 1
2017-08-14 02:24:34,074 [main] INFO org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total
input paths to process : 1
((Ludicolo,80))
((Dodrio,60))
((Trevenant,85))
((Gurdurr,85))
((Zangoose,73))
grunt>
```

Players of List 2

Two lists of Players have been generated, now storing the result in Local file system.

Store List 1

```
grunt> STORE limit_data_random1_desc INTO '/home/acadgild/Documents/pig/player1.txt';
```

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

Output(s):
Successfully stored 5 records in: "/home/acadgild/Documents/pig/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_local1034474391_0032 -> job_local1689693395_0033,
job_local1689693395_0033 -> job_local93445475_0034,
job_local93445475_0034 -> job_local1066255538_0035,
job_local1066255538_0035

2017-08-14 02:28:52,561 [main] INFO org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
```

Statements show that data has successfully been stored at specified location

Store List 2

```
grunt> STORE limit_data_random2_desc INTO '/home/acadgild/Documents/pig/player2.txt';
```

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

Output(s):
Successfully stored 5 records in: "/home/acadgild/Documents/pig/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_local822162980_0036 -> job_local346499309_0037,
job_local346499309_0037 -> job_local1703083702_0038,
job_local1703083702_0038 -> job_local493730320_0039,
job_local493730320_0039
```

Statements show that data has successfully been stored at specified location

Below commands verify that data has been stored successfully at location /home/acadgild/Documents/pig.

```
[acadgild@localhost pig]$ ls -lrt p*.txt
player1.txt: ✓
total 4
-rw-r--r--. 1 acadgild acadgild 317 Aug 14 02:28 part-r-00000
-rw-r--r--. 1 acadgild acadgild  0 Aug 14 02:28 _SUCCESS

player2.txt: ✓
total 4
-rw-r--r--. 1 acadgild acadgild 320 Aug 14 02:30 part-r-00000
-rw-r--r--. 1 acadgild acadgild  0 Aug 14 02:30 _SUCCESS
[acadgild@localhost pig]$
```

```
[acadgild@localhost pig]$ ls -lrt player1.txt ←
total 4
-rw-r--r--. 1 acadgild acadgild 317 Aug 14 02:28 part-r-00000 ✓
-rw-r--r--. 1 acadgild acadgild  0 Aug 14 02:28 _SUCCESS ✓
[acadgild@localhost pig]$ cat player1.txt/part-r-00000 ←
0.9938292801117562      Slowpoke      Water  Psychic 315    90    65    6
5      40      40      15
0.9871159705222213      Emboar Fire    Fighting 528    110   123    6
5      100     65      65
0.9844032031660561      Roserade     Grass  Poison  515    60    70    6
5      125     105     90
0.9830694996227776      Raticate     Normal  413    55    81    6
0      50      70      97
0.9817703161299848      Escavalier   Bug     Steel   495    70    135   1
05     60      105     20
[acadgild@localhost pig]$ ls -lrt player2.txt ←
total 4
-rw-r--r--. 1 acadgild acadgild 320 Aug 14 02:30 part-r-00000 ✓
-rw-r--r--. 1 acadgild acadgild  0 Aug 14 02:30 _SUCCESS ✓
[acadgild@localhost pig]$ cat player2.txt/part-r-00000 ←
0.9894769742710033      Graveler     Rock    Ground  390    55    95    1
15     45      45      35
0.9842170215560896      Heatran Fire  Steel   600    91    90    106   1
30     106     77
0.983638860301221      KangaskhanMega Kangaskhan Normal  590    1
05     125     100     60    100    100
0.9827111792155653      Shellder     Water   305    30    65    1
00     45      25      40
0.9813187683688477      Basculin     Water   460    70    92    6
5      80      55      98
[acadgild@localhost pig]$
```