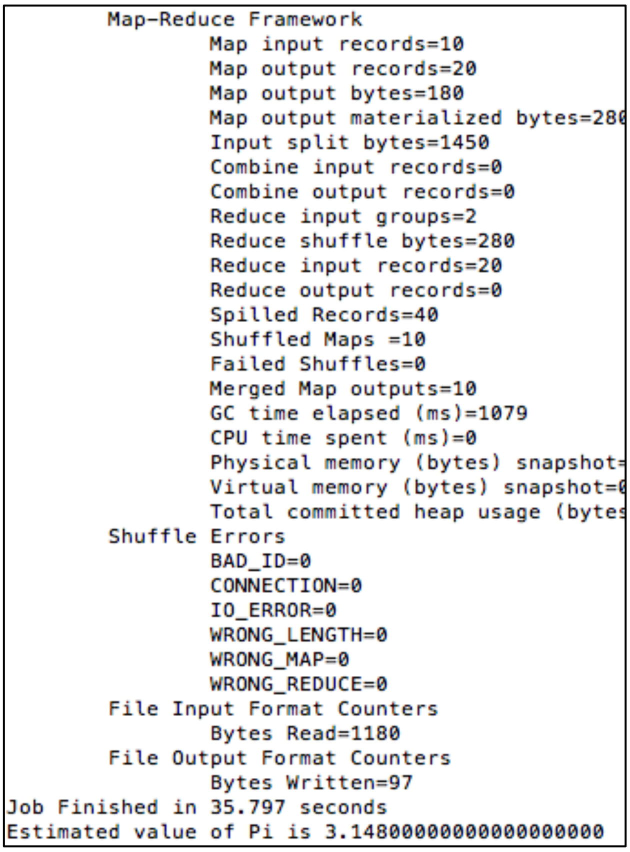
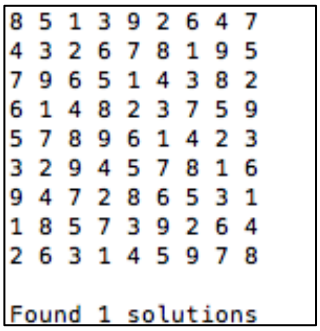
Assignment -1
EECS 6893
Vibhuti Mahajan (vm2486)

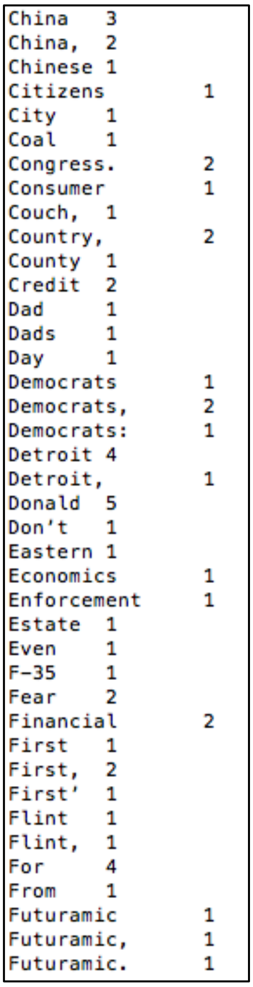
1. Installed hadoop and ran the examples in the reference:



```
        Map-Reduce Framework
                Map input records=10
                Map output records=20
                Map output bytes=180
                Map output materialized bytes=28
                Input split bytes=1450
                Combine input records=0
                Combine output records=0
                Reduce input groups=2
                Reduce shuffle bytes=280
                Reduce input records=20
                Reduce output records=0
                Spilled Records=40
                Shuffled Maps =10
                Failed Shuffles=0
                Merged Map outputs=10
                GC time elapsed (ms)=1079
                CPU time spent (ms)=0
                Physical memory (bytes) snapshot=
                Virtual memory (bytes) snapshot=0
                Total committed heap usage (bytes
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=1180
        File Output Format Counters
                Bytes Written=97
Job Finished in 35.797 seconds
Estimated value of Pi is 3.14800000000000000000
```



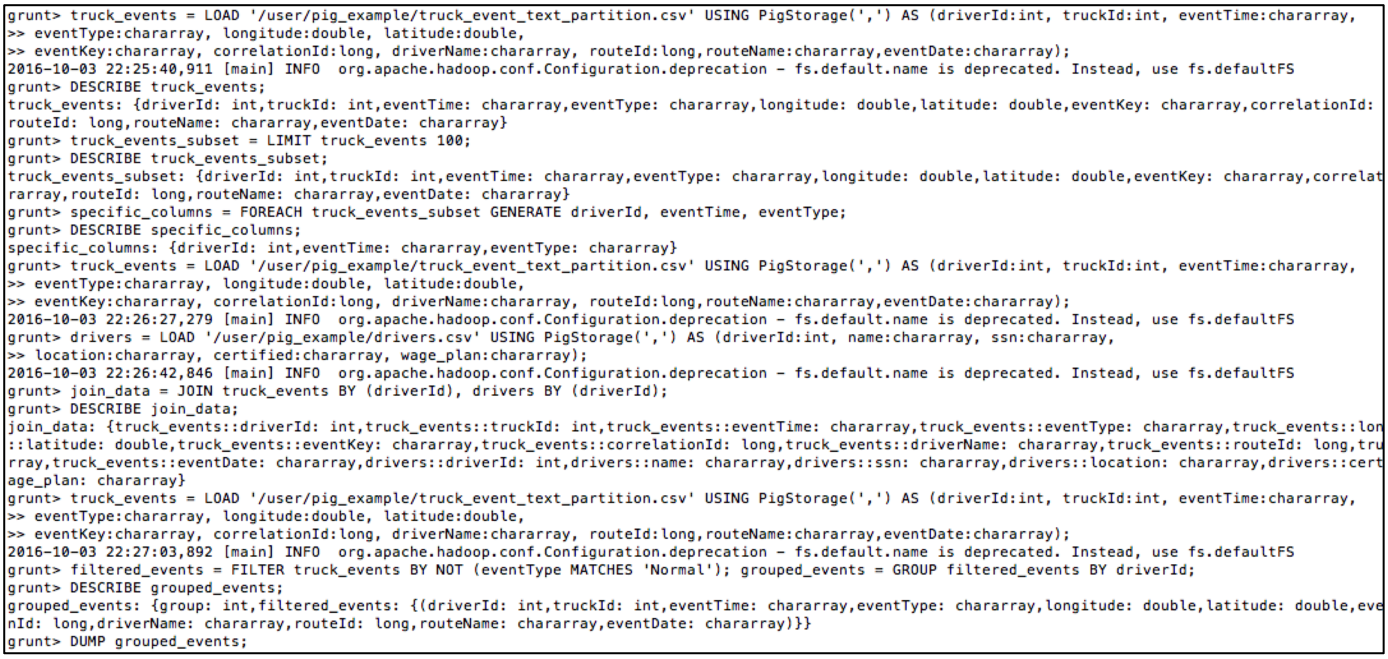
```
8 5 1 3 9 2 6 4 7
4 3 2 6 7 8 1 9 5
7 9 6 5 1 4 3 8 2
6 1 4 8 2 3 7 5 9
5 7 8 9 6 1 4 2 3
3 2 9 4 5 7 8 1 6
9 4 7 2 8 6 5 3 1
1 8 5 7 3 9 2 6 4
2 6 3 1 4 5 9 7 8

Found 1 solutions
```



```
China   3
China,  2
Chinese 1
Citizens        1
City    1
Coal    1
Congress.       2
Consumer        1
Couch,  1
Country,        2
County  1
Credit  2
Dad     1
Dads    1
Day     1
Democrats       1
Democrats,      2
Democrats:      1
Detroit 4
Detroit,        1
Donald  5
Don't   1
Eastern 1
Economics       1
Enforcement     1
Estate  1
Even    1
F-35    1
Fear    2
Financial       2
First   1
First,  2
First'  1
Flint   1
Flint,  1
For     4
From    1
Futuramic       1
Futuramic,      1
Futuramic.      1
```

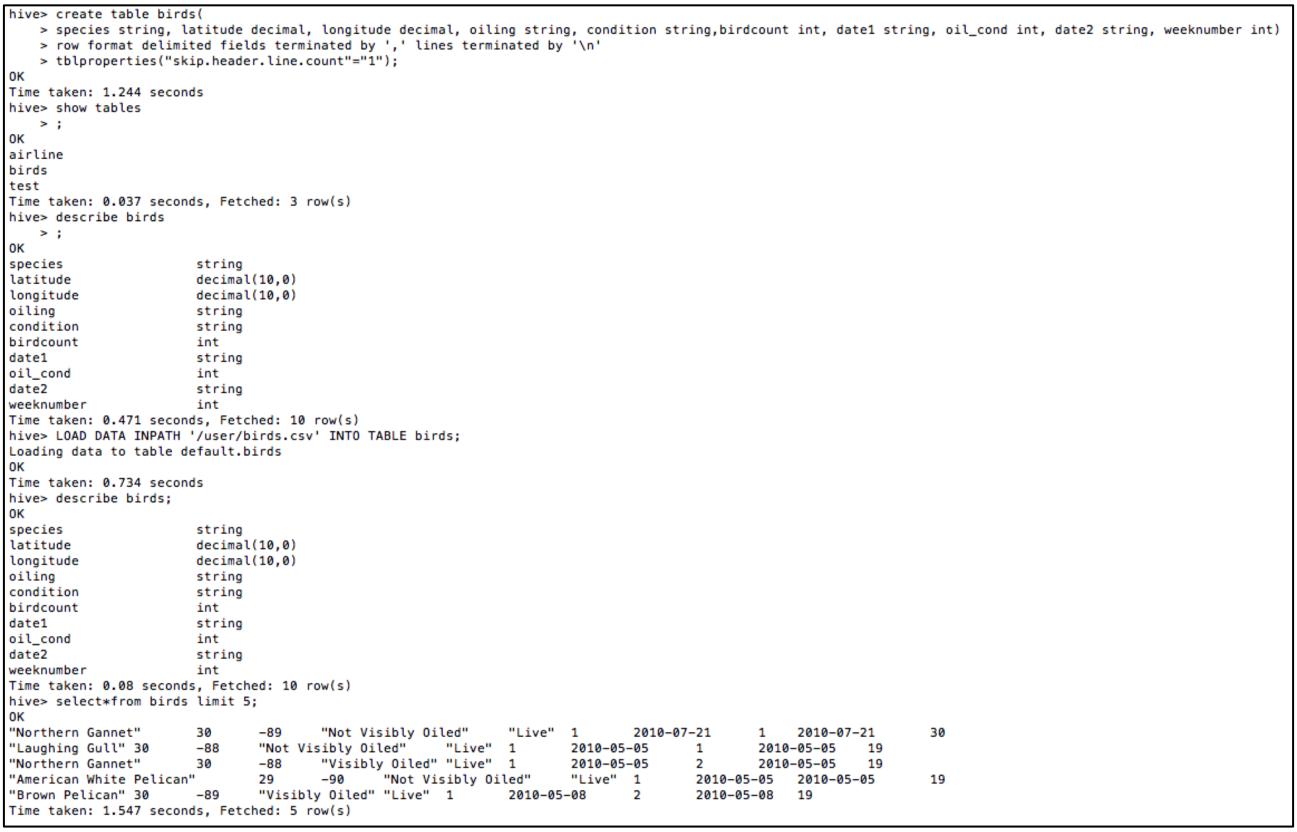
2. Downloaded the airline data and birds.csv from 2011: Deepwater horizon oil spill. Saved birds.csv in /Users/abc/Desktop/birds.csv

3. PIG example from reference:

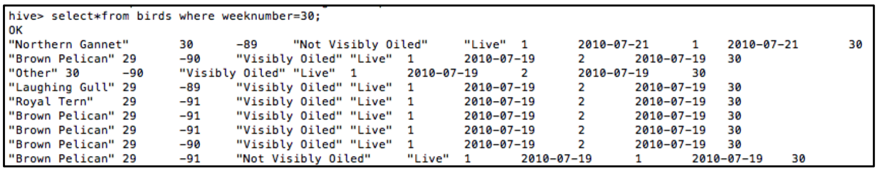


```
grunt> truck_events = LOAD '/user/pig_example/truck_event_text_partition.csv' USING PigStorage(',') AS (driverId:int, truckId:int, eventTime:chararray,
>> eventType:chararray, longitude:double, latitude:double,
>> eventKey:chararray, correlationId:long, driverName:chararray, routeId:long,routeName:chararray,eventDate:chararray);
2016-10-03 22:25:40,911 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
grunt> DESCRIBE truck_events;
truck_events: {driverId: int,truckId: int,eventTime: chararray,eventType: chararray,longitude: double,latitude: double,eventKey: chararray,correlationId:
routeId: long,routeName: chararray,eventDate: chararray}
grunt> truck_events_subset = LIMIT truck_events 100;
grunt> DESCRIBE truck_events_subset;
truck_events_subset: {driverId: int,truckId: int,eventTime: chararray,eventType: chararray,longitude: double,latitude: double,eventKey: chararray,correlat
rarray,routeId: long,routeName: chararray,eventDate: chararray}
grunt> specific_columns = FOREACH truck_events_subset GENERATE driverId, eventTime, eventType;
grunt> DESCRIBE specific_columns;
specific_columns: {driverId: int,eventTime: chararray,eventType: chararray}
grunt> truck_events = LOAD '/user/pig_example/truck_event_text_partition.csv' USING PigStorage(',') AS (driverId:int, truckId:int, eventTime:chararray,
>> eventType:chararray, longitude:double, latitude:double,
>> eventKey:chararray, correlationId:long, driverName:chararray, routeId:long,routeName:chararray,eventDate:chararray);
2016-10-03 22:26:27,279 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
grunt> drivers = LOAD '/user/pig_example/drivers.csv' USING PigStorage(',') AS (driverId:int, name:chararray, ssn:chararray,
>> location:chararray, certified:chararray, wage_plan:chararray);
2016-10-03 22:26:42,846 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
grunt> join_data = JOIN truck_events BY (driverId), drivers BY (driverId);
grunt> DESCRIBE join_data;
join_data: {truck_events::driverId: int,truck_events::truckId: int,truck_events::eventTime: chararray,truck_events::eventType: chararray,truck_events::lon
::latitude: double,truck_events::eventKey: chararray,truck_events::correlationId: long,truck_events::driverName: chararray,truck_events::routeId: long,tru
rray,truck_events::eventDate: chararray,drivers::driverId: int,drivers::name: chararray,drivers::ssn: chararray,drivers::location: chararray,drivers::cert
age_plan: chararray}
grunt> truck_events = LOAD '/user/pig_example/truck_event_text_partition.csv' USING PigStorage(',') AS (driverId:int, truckId:int, eventTime:chararray,
>> eventType:chararray, longitude:double, latitude:double,
>> eventKey:chararray, correlationId:long, driverName:chararray, routeId:long,routeName:chararray,eventDate:chararray);
2016-10-03 22:27:03,892 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
grunt> filtered_events = FILTER truck_events BY NOT (eventType MATCHES 'Normal'); grouped_events = GROUP filtered_events BY driverId;
grunt> DESCRIBE grouped_events;
grouped_events: {group: int,filtered_events: {(driverId: int,truckId: int,eventTime: chararray,eventType: chararray,longitude: double,latitude: double,eve
nId: long,driverName: chararray,routeId: long,routeName: chararray,eventDate: chararray)}}
grunt> DUMP grouped_events;
```

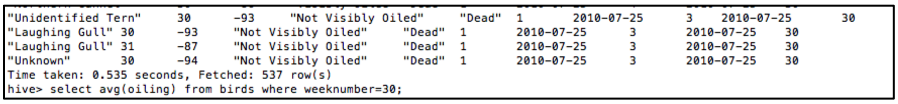
4. Hive example on birds.csv



```
hive> create table birds(
    > species string, latitude decimal, longitude decimal, oiling string, condition string,birdcount int, date1 string, oil_cond int, date2 string, weeknumber int)
    > row format delimited fields terminated by ',' lines terminated by '\n'
    > tblproperties("skip.header.line.count"="1");
OK
Time taken: 1.244 seconds
hive> show tables
    > ;
OK
airline
birds
test
Time taken: 0.037 seconds, Fetched: 3 row(s)
hive> describe birds
    > ;
OK
species                 string
latitude                decimal(10,0)
longitude               decimal(10,0)
oiling                  string
condition               string
birdcount               int
date1                   string
oil_cond                int
date2                   string
weeknumber              int
Time taken: 0.471 seconds, Fetched: 10 row(s)
hive> LOAD DATA INPATH '/user/birds.csv' INTO TABLE birds;
Loading data to table default.birds
OK
Time taken: 0.734 seconds
hive> describe birds;
OK
species                 string
latitude                decimal(10,0)
longitude               decimal(10,0)
oiling                  string
condition               string
birdcount               int
date1                   string
oil_cond                int
date2                   string
weeknumber              int
Time taken: 0.08 seconds, Fetched: 10 row(s)
hive> select*from birds limit 5;
OK
"Northern Gannet"       30      -89     "Not Visibly Oiled"     "Live"  1       2010-07-21      1   2010-07-21      30
"Laughing Gull" 30      -88     "Not Visibly Oiled"     "Live"  1       2010-05-05      1       2010-05-05      19
"Northern Gannet"       30      -88     "Visibly Oiled" "Live"  1       2010-05-05      2       2010-05-05      19
"American White Pelican"        29      -90     "Not Visibly Oiled"     "Live"  1       2010-05-05  2010-05-05      19
"Brown Pelican" 30      -89     "Visibly Oiled" "Live"  1       2010-05-08      2       2010-05-08  19
Time taken: 1.547 seconds, Fetched: 5 row(s)
```

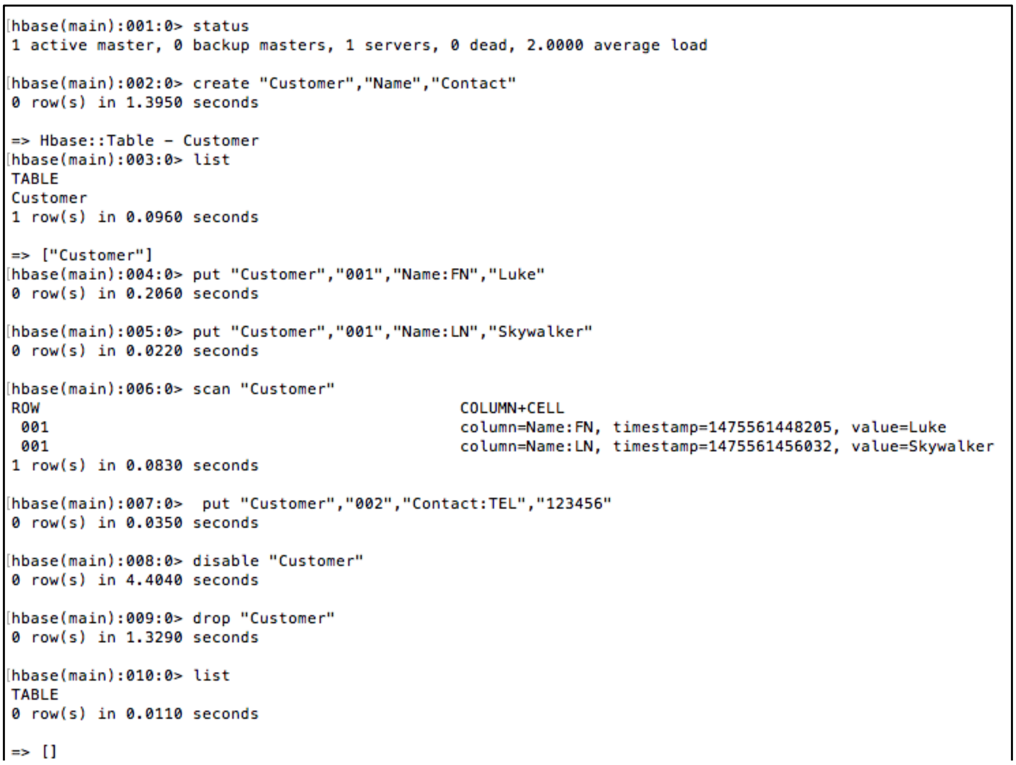


```
hive> select*from birds where weeknumber=30;
OK
"Northern Gannet"       30      -89     "Not Visibly Oiled"     "Live"  1       2010-07-21      1   2010-07-21      30
"Brown Pelican" 29      -90     "Visibly Oiled" "Live"  1       2010-07-19      2       2010-07-19  30
"Other" 30      -90     "Visibly Oiled" "Live"  1       2010-07-19      2       2010-07-19      30
"Laughing Gull" 29      -89     "Visibly Oiled" "Live"  1       2010-07-19      2       2010-07-19  30
"Royal Tern"    29      -91     "Visibly Oiled" "Live"  1       2010-07-19      2       2010-07-19  30
"Brown Pelican" 29      -91     "Visibly Oiled" "Live"  1       2010-07-19      2       2010-07-19  30
"Brown Pelican" 29      -91     "Visibly Oiled" "Live"  1       2010-07-19      2       2010-07-19  30
"Brown Pelican" 29      -90     "Visibly Oiled" "Live"  1       2010-07-19      2       2010-07-19  30
"Brown Pelican" 29      -91     "Not Visibly Oiled"     "Live"  1       2010-07-19      1       2010-07-19  30
```



```
"Unidentified Tern"     30      -93     "Not Visibly Oiled"     "Dead"  1       2010-07-25      3   2010-07-25      30
"Laughing Gull" 30      -93     "Not Visibly Oiled"     "Dead"  1       2010-07-25      3       2010-07-25  30
"Laughing Gull" 31      -87     "Not Visibly Oiled"     "Dead"  1       2010-07-25      3       2010-07-25  30
"Unknown"       30      -94     "Not Visibly Oiled"     "Dead"  1       2010-07-25      3       2010-07-25  30
Time taken: 0.535 seconds, Fetched: 537 row(s)
hive> select avg(oiling) from birds where weeknumber=30;
```

5. Hbase example:



```
hbase(main):001:0> status
1 active master, 0 backup masters, 1 servers, 0 dead, 2.0000 average load

hbase(main):002:0> create "Customer","Name","Contact"
0 row(s) in 1.3950 seconds

=> Hbase::Table - Customer
hbase(main):003:0> list
TABLE
Customer
1 row(s) in 0.0960 seconds

=> ["Customer"]
hbase(main):004:0> put "Customer","001","Name:FN","Luke"
0 row(s) in 0.2060 seconds

hbase(main):005:0> put "Customer","001","Name:LN","Skywalker"
0 row(s) in 0.0220 seconds

hbase(main):006:0> scan "Customer"
ROW                                            COLUMN+CELL
 001                                           column=Name:FN, timestamp=1475561448205, value=Luke
 001                                           column=Name:LN, timestamp=1475561456032, value=Skywalker
1 row(s) in 0.0830 seconds

hbase(main):007:0>  put "Customer","002","Contact:TEL","123456"
0 row(s) in 0.0350 seconds

hbase(main):008:0> disable "Customer"
0 row(s) in 4.4040 seconds

hbase(main):009:0> drop "Customer"
0 row(s) in 1.3290 seconds

hbase(main):010:0> list
TABLE
0 row(s) in 0.0110 seconds

=> []
```