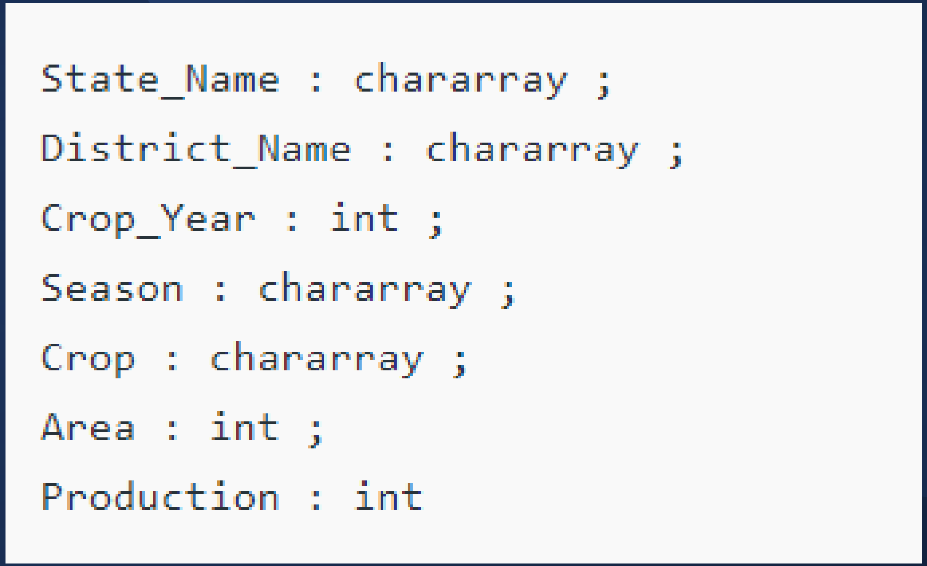
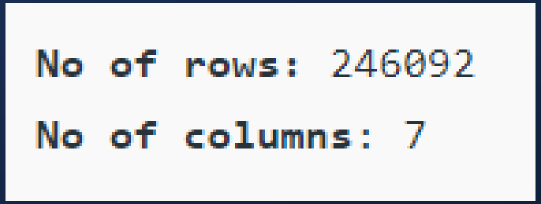
Operators for data processing

1. Download the dataset containing the Agriculture related data about crops in various regions and their area and produce. The link for dataset –https://www.kaggle.com/abhinand05/crop-production-in-india The dataset contains 7 columns namely as follows.

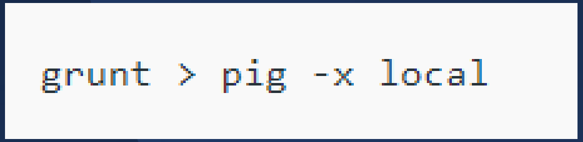


```
State_Name : chararray ;
District_Name : chararray ;
Crop_Year : int ;
Season : chararray ;
Crop : chararray ;
Area : int ;
Production : int
```



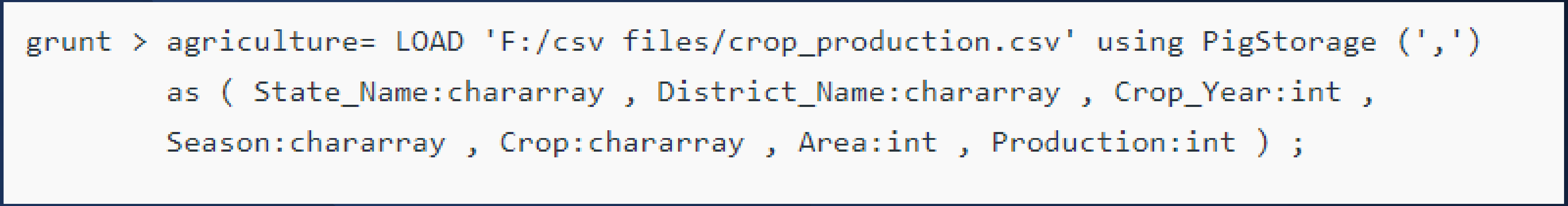
```
No of rows: 246092
No of columns: 7
```

2. Enter pig local mode using



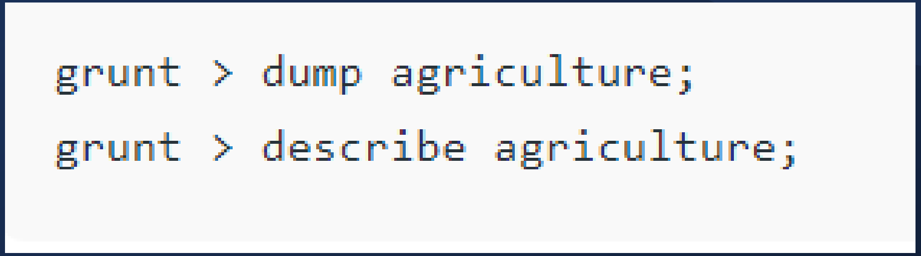
```
grunt > pig -x local
```

3. Load the dataset in the local mode



```
grunt > agriculture= LOAD 'F:/csv files/crop_production.csv' using PigStorage (',')
        as ( State_Name:chararray , District_Name:chararray , Crop_Year:int ,
        Season:chararray , Crop:chararray , Area:int , Production:int ) ;
```

4. Dump and describe the data set agriculture using

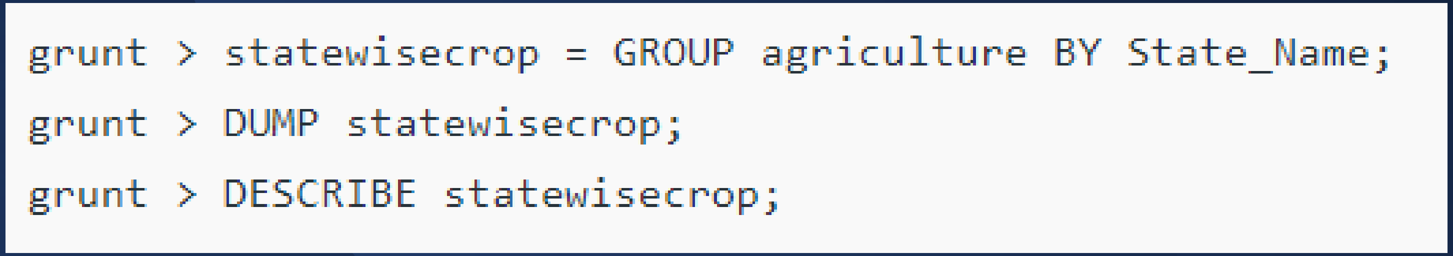


```
grunt > dump agriculture;
grunt > describe agriculture;
```

5. Executing the PIG queries in local mode

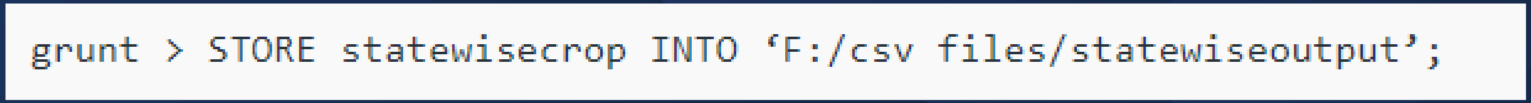
Query 1: Grouping All Records State wise.

This command will group all the records by the colomn State_Name.



```
grunt > statewisecrop = GROUP agriculture BY State_Name;
grunt > DUMP statewisecrop;
grunt > DESCRIBE statewisecrop;
```

Now store the result of the query in a CSV file for better understanding. We have to mention the name of the object and the path where it needs to be stored.

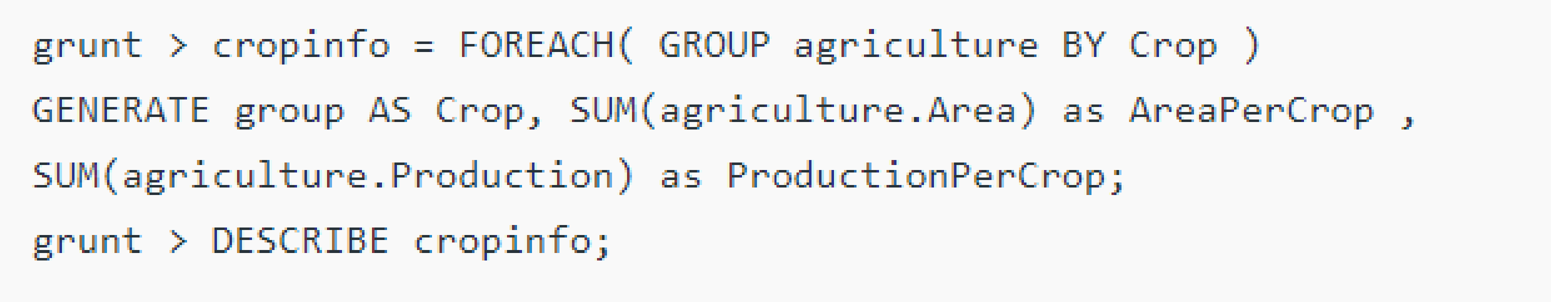


```
grunt > STORE statewisecrop INTO ‘F:/csv files/statewiseoutput’;
```

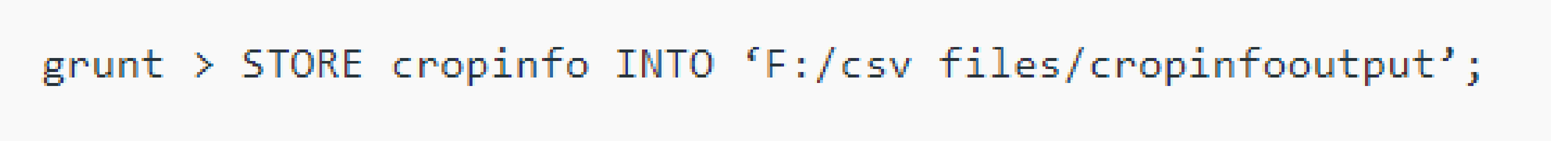
The output will be in a file named **‘part-r-00000’** which needs to be renamed as ‘**part-r-00000.csv**’ to be opened in the Excel format and to make it readable. You will find this file in the path that we have mentioned in the above query. In my case it was in the path ‘F:/csv files/statewiseoutput/’.

Query 2: Generate Total Crop wise Production and Area

In the above query, we need to group by Crop type and then find the SUM of their Productions and Area.



```
grunt > cropinfo = FOREACH( GROUP agriculture BY Crop )
GENERATE group AS Crop, SUM(agriculture.Area) as AreaPerCrop ,
SUM(agriculture.Production) as ProductionPerCrop;
grunt > DESCRIBE cropinfo;
```

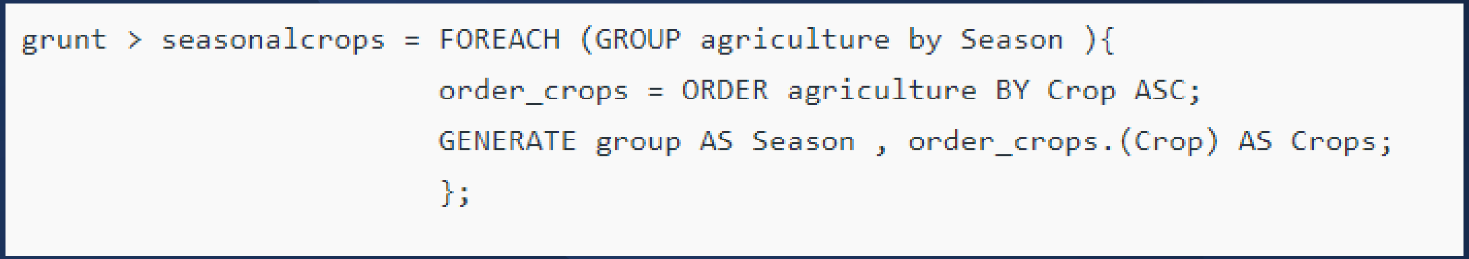


```
grunt > STORE cropinfo INTO ‘F:/csv files/cropinfooutput’;
```

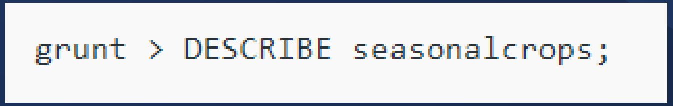
The output will be in a file named **‘part-r-00000’** which needs to be renamed as **‘part-r-00000.csv’** to be opened in the Excel format and to make it readable.

Query 3: The majority of crops are grown in a Season and in which year.

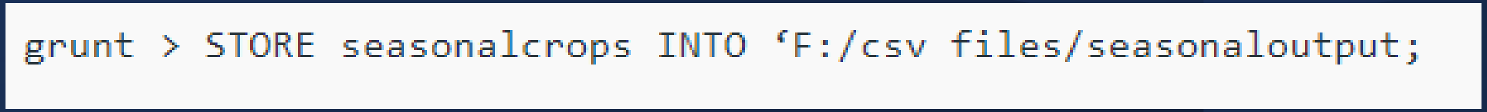
In this query, we need to group the crops by season and order them alphabetically. Also, this will tell us which crops are found in a season and with year.



```
grunt > seasonalcrops = FOREACH (GROUP agriculture by Season ){
                        order_crops = ORDER agriculture BY Crop ASC;
                        GENERATE group AS Season , order_crops.(Crop) AS Crops;
                        };
```



```
grunt > DESCRIBE seasonalcrops;
```

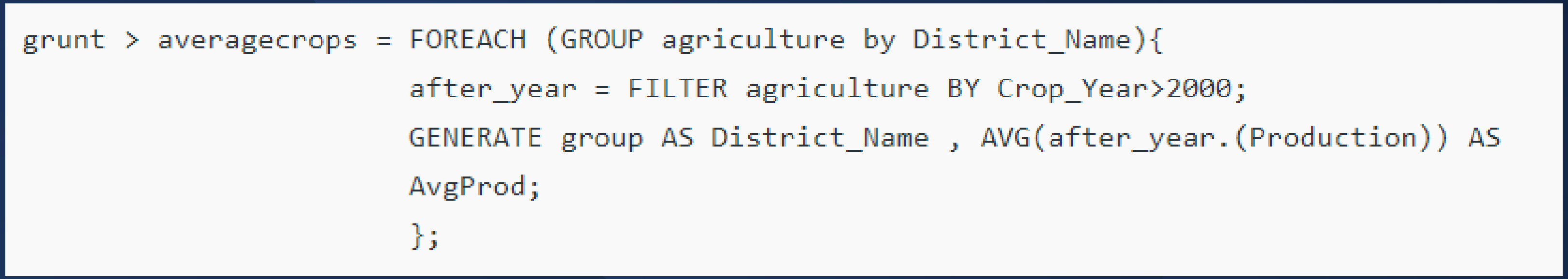


```
grunt > STORE seasonalcrops INTO ‘F:/csv files/seasonaloutput;
```

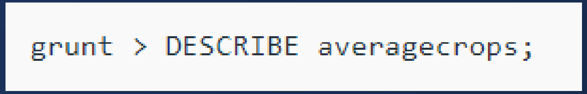
The output will be in a file named ‘part-r-00000’ which needs to be renamed as ‘part-r-00000.csv’ to be opened in the Excel format and to make it readable

Query 4: Average crop production in each district after the year 2000.

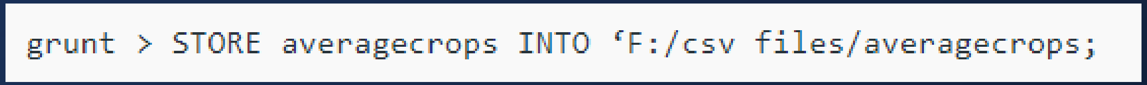
First, we need to group by district name and then find the average of the total crop production but only after the year 2000.



```
grunt > averagecrops = FOREACH (GROUP agriculture by District_Name){
                        after_year = FILTER agriculture BY Crop_Year>2000;
                        GENERATE group AS District_Name , AVG(after_year.(Production)) AS
                        AvgProd;
                        };
```



```
grunt > DESCRIBE averagecrops;
```

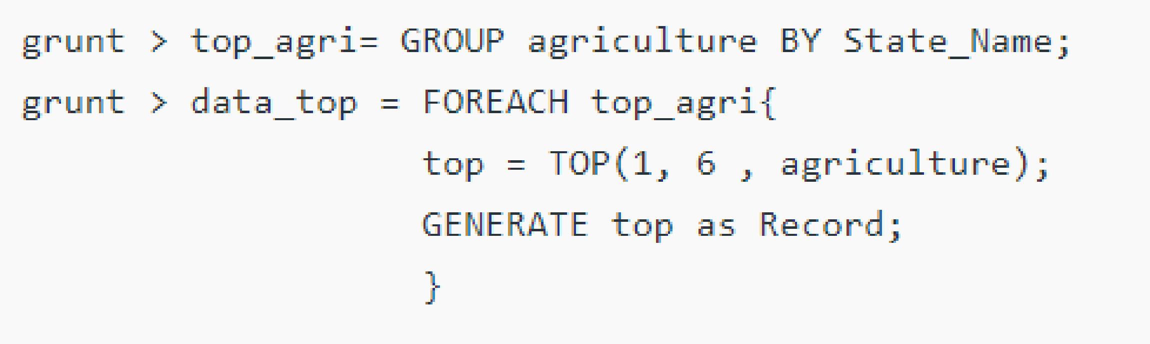


```
grunt > STORE averagecrops INTO ‘F:/csv files/averagecrops;
```

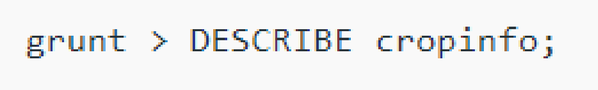
You can check the output from the ‘part-r-00000.csv’ by opening the file. This file will contain two columns. The first one has all distinct district names and the second one will have the average production of all crops in each district after the year 2000.

Query 5: Highest produced crops and details from each State.

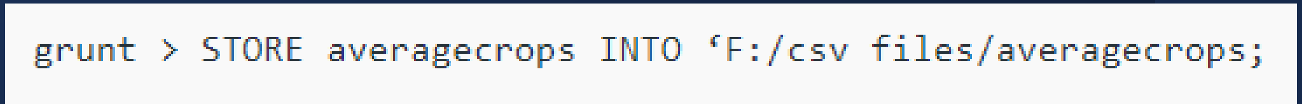
First, we need to group the input by the state name. Then iterate through each grouped record and then find the TOP 1 record with the highest Production from each state.



```
grunt > top_agri= GROUP agriculture BY State_Name;
grunt > data_top = FOREACH top_agri{
                   top = TOP(1, 6 , agriculture);
                   GENERATE top as Record;
                   }
```



```
grunt > DESCRIBE cropinfo;
```



```
grunt > STORE averagecrops INTO ‘F:/csv files/averagecrops;
```

You can check the output from the ‘part-r-00000.csv’ by opening the file. This file contains records from each unique state who are having the highest Production amount. Read above and follow the steps to create ‘part-r-00000.csv’.