Cars Dataset:

The columns in cars.csv are mpg, cylinders, cubicinches, hp, weightlbs, time-to-60, year, brand

1.Remove the headers

2.Rename to '.txt'

3.Copy the local file in hdfs

#hadoop dfs -put 'file' /

LOAD Dataset:

cars = LOAD '/cars.txt' using PigStorage (',') as

(mpg:int,cylinders:int,cubicinches:int,hp:int,weight:int,time:int,year:int,brand:chararray);

```
grunt> cars = LOAD '/cars.txt' using PigStorage (',') as (mpg:int,cylinders:int,cubicinches:int
,hp:int,weight:int,time:int,year:int,brand:chararray);
```

View Dataset:

dump cars;

```
grunt> dump cars;
2020-03-07 16:29:24,020 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapreduce.job.
counters.limit is deprecated. Instead, use mapreduce.job.counters.max
```

Analysing Dataset:

1)Average MPG By Brand:

GroupByBrand = GROUP cars BY brand;

dump GroupByBrand;

avg_mpg = foreach GroupByBrand Generate group,AVG(cars.mpg);

dump avg_mpg;

```
( US.,19.475308641975311)
( Japan.,30.0)
( Europe.,27.375)
```

2)Maximum HP By Year:

```
GroupByYear = GROUP cars BY year;
dump GroupByYear;
max_hp = foreach GroupByYear Generate group,MAX(cars.mpg);
dump max_hp;
```

```
otal input paths to process : 1
(1971,27)
(1972,35)
(1973,28)
(1974,29)
(1975,32)
(1976,33)
(1977,33)
(1978,33)
(1979,43)
(1980,37)
(1981,46)
(1982,39)
(1983,44)
```

3)Release Count by year:

```
GroupByYear = GROUP cars BY year;
dump GroupByYear;
car_release = foreach GroupByYear Generate group,COUNT(cars.mpg);
dump car_release;
```

```
(1971,23)
(1972,15)
(1973,16)
(1974,35)
(1975,12)
(1976,22)
(1977,26)
(1978,18)
(1979,21)
(1980,21)
(1981,18)
(1982,18)
(1983,16)
```

4) Release Count by year and Brand:

```
GroupByBrandYear = GROUP cars BY (brand,year);
dump GroupByBrandYear;
car_release = foreach GroupByBrandYear Generate group,COUNT(cars.mpg);
dump car_release;
```

```
(( US.,1971),18)
(( US.,1972),10)
(( US.,1973),9)
(( US.,1974),25)
(( US.,1975),6)
(( US.,1976),14)
(( US.,1977),16)
(( US.,1978),11)
(( US.,1979),12)
(( US.,1980),15)
(( US.,1981),4)
(( US.,1982),10)
(( US.,1983),12)
(( Japan.,1971),2)
(( Japan.,1972),2)
(( Japan.,1973),3)
(( Japan.,1974),4)
(( Japan.,1975),2)
```

5) Minimum cubicinches by year and Brand:

```
GroupByBrandYear = GROUP cars BY (brand,year);
dump GroupByBrandYear;
car_release = foreach GroupByBrandYear Generate group,MIN(cars.cubicinches);
dump car_release;
```

```
(( US.,1971),199)
(( US.,1972),91)
(( US.,1973),98)
(( US.,1974),140)
(( US.,1975),90)
(( US.,1976),140)
(( US.,1977),98)
(( US.,1978),98)
(( US.,1979),98)
(( US.,1980),105)
(( US.,1981),98)
(( US.,1982),86)
(( US.,1983),105)
(( Japan.,1971),97)
(( Japan.,1972),71)
(( Japan.,1973),97)
(( Japan.,1974),70)
```