***Objective – 1 : Load file into spark***

***Objective- 2:***

* ***What is the average amount of AverageCoveredCharges per state***
* ***find out the AverageTotalPayments charges per state***
* ***find out the AverageMedicarePayments charges per state.***

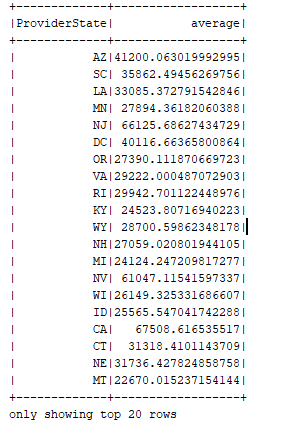
***Objectiv-3 :***

* ***Find out the total number of Discharges per state and for each disease***
* ***Sort the output in descending order of totalDischarges***

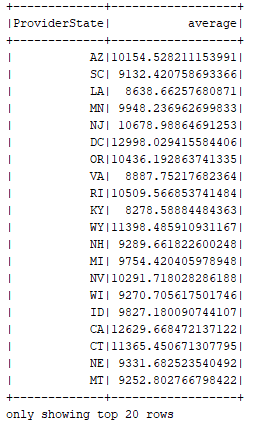
**Code logic**

**package** SQL  
  
**import** SQL.SparkSQLAssignment.sportsdata  
**import** org.apache.spark.sql.SparkSession  
  
**object** SparkSQLInterviewPrep2 {  
  
  
 **def** main(args: Array[String]): Unit = {  
  
 *println*(**"hey scala"**)  
  
 **val** spark = SparkSession  
 .*builder*()  
 .master(**"local"**)  
 .appName(**"Spark Hospital case study"**)  
 .config(**"spark.some.config.option"**, **"some-value"**)  
 .getOrCreate()  
  
 *println*(**"Spark Session Object created"**)  
  
 *//Set the log level as warning* spark.sparkContext.setLogLevel(**"WARN"**)  
  
 **val** df = spark.read.format(**"csv"**).option(**"header"**, **"true"**).load(**"G:\\Bigdata\\inpatientCharges.csv"**)  
  
 *println*(**"Hospital data->>"** + df.count())  
  
 **import** org.apache.spark.sql.functions.\_  
  
 *//What is the average amount of AverageCoveredCharges per state* **val** dfavgCC = df.groupBy(**"ProviderState"**).agg(*avg*(**"AverageCoveredCharges"**).as(**"average"**)).select(**"ProviderState"**,**"average"**)  
  
 dfavgCC.show()  
  
 *//find out the AverageTotalPayments charges per state* **val** dfavgTP = df.groupBy(**"ProviderState"**).agg(*avg*(**"AverageTotalPayments"**).as(**"average"**)).select(**"ProviderState"**,**"average"**)  
  
 dfavgTP.show()  
  
 *//find out the AverageMedicarePayments charges per state* **val** dfavgMP = df.groupBy(**"ProviderState"**).agg(*avg*(**"AverageMedicarePayments"**).as(**"average"**)).select(**"ProviderState"**,**"average"**)  
  
 dfavgMP.show()  
  
 *//Find out the total number of Discharges per state and for each disease* **val** dfTotalDischarge = df.groupBy(**"ProviderState"**, **"DRGDefinition"**).agg(*sum*(**"TotalDischarges"**).as(**"Total"**)).select(**"ProviderState"**,**"DRGDefinition"**,**"Total"**)  
  
 dfTotalDischarge.show()  
  
 *//Sort the output in descending order of totalDischarges* **val** dfTotalDischargeSort = df.groupBy(**"ProviderState"**, **"DRGDefinition"**).agg(*sum*(**"TotalDischarges"**).as(**"Total"**)).select(**"ProviderState"**,**"DRGDefinition"**,**"Total"**).sort(*desc*(**"Total"**)).show()  
  
}  
}

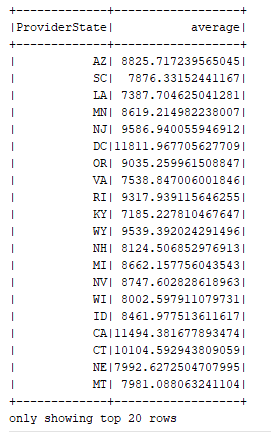
* ***What is the average amount of AverageCoveredCharges per state***



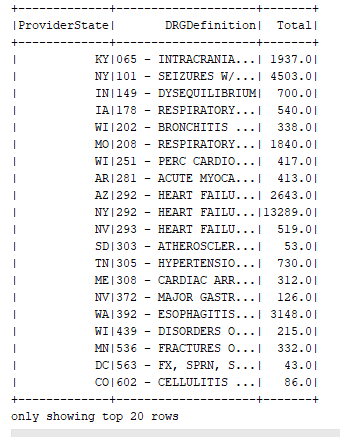
* ***find out the AverageTotalPayments charges per state***



* ***find out the AverageMedicarePayments charges per state.***



* ***Find out the total number of Discharges per state and for each disease***



* ***Sort the output in descending order of totalDischarges***

