Case Study IV

For Hospital Data Analysis in the United States

- Prachi Mohite

Dataset Description

DRG Definition: The code and description identifying the MS-DRG. MS-DRGs are a classification system that groups similar
clinical conditions (diagnoses) and procedures furnished by the hospital during their stay.
Provider Id: The CMS Certification Number (CCN) assigned to the Medicare-certified hospital facility.
Provider Name: The name of the provider.
Provider Street Address: The provider's street address.
Provider City: The city where the provider is located.
Provider State: The state where the provider is located.
Provider Zip Code: The provider's zip code.
Provider HRR: The Hospital Referral Region (HRR) where the provider is located.
Total Discharges: The number of discharges billed by the provider for inpatient hospital services.

Average Covered Charges: The provider's average charge for services covered by Medicare for all discharges in the
MS-DRG. These will vary from hospital to hospital because of the differences in hospital charge structures.
Average Total Payments: The average total payments to all providers for the MS-DRG including the MSDRG amount,
teaching, disproportionate share, capital, and outlier payments for all cases. Also included in the average total
payments are co-payment and deductible amounts that the patient is responsible for and any additional payments by
third parties for coordination of benefits.

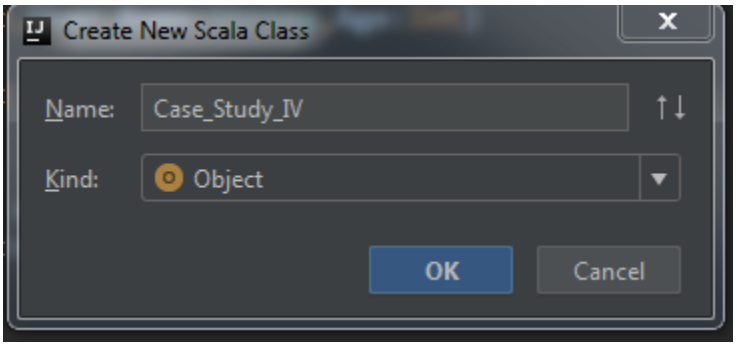
Average Medicare Payments: The average amount that Medicare pays to the provider for Medicare's share of the
MS-DRG. Average Medicare payment amounts include the MS-DRG amount, teaching, disproportionate share,
capital, and outlier payments for all cases. Medicare payments DO NOT include beneficiary co-payments and
deductible amounts nor any additional payments from third parties for coordination of benefits.
You can download the dataset used in this spark SQL use case from below link.4
https://drive.google.com/open?id=13_YDmwENxOQI5asLRa6tOP8FgiqqM9jc

Tasks

1. Load file into spark
2. What is the average amount of AverageCoveredCharges per state
3. find out the AverageTotalPayments charges per state
4. find out the AverageMedicarePayments charges per state.
5. Find out the total number of Discharges per state and for each disease
6. Sort the output in descending order of totalDischarges

In this Assignment we will be using IDEA IntelliJ to Complete the given Task

1. Created new Project and added scala object named as Case_Study_VI as below



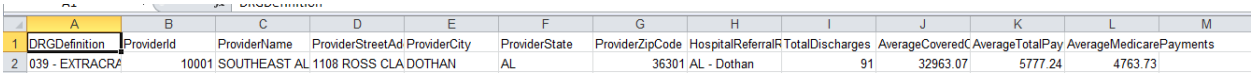
2. To add the required dependencies we have created scala sbt project in IDEA and added library dependency from maven repository as below

```
name := "Project1"

version := "0.1"

scalaVersion := "2.11.7"
libraryDependencies += "org.apache.spark" %% "spark-core" % "2.1.0"
libraryDependencies += "org.apache.spark" %% "spark-sql" % "2.1.0" % "provided"
```

3. Added main function and created the spark object as below

```scala
class Case_Study_IV {
  def main(args: Array[String]): Unit =
  {

    println("hey scala")

    //Create spark object
    val spark = SparkSession
      .builder()
      .master( master = "local")
      .appName( name = "Hospital Case Study IV")
      .config("spark.some.config.option", "some-value")
      .getOrCreate()

    println("Spark Session Object created")
  }

}
```

4. We will be using below dataset for this assignment – inpatientCharges.csv



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	DRGDefinition	ProviderId	ProviderName	ProviderStreetAd	ProviderCity	ProviderState	ProviderZipCode	HospitalReferralR	TotalDischarges	AverageCoveredC	AverageTotalPay	AverageMedicarePayments	
2	039 - EXTRACRA	10001	SOUTHEAST AL	1108 ROSS CLA	DOTHAN	AL	36301	AL - Dothan	91	32963.07	5777.24	4763.73	

   a. Columns are - DRGDefinition, ProviderId, ProviderName, ProviderStreetAddress, ProviderCity, ProviderState, ProviderZipCode, HospitalReferralRegionDescription, TotalDischarges, AverageCoveredCharges, AverageTotalPayments, AverageMedicarePayments

5. To Complete the assignment first we have to load the data from these local files to Dataframes in Spark SQL as below
   As discussed in Case study III there are two ways to create dataframe from RDD.
   Approach 1 : Inferring the Schema using Reflection
   a. Created the case class to map the details in Dataframe from text files
      Case Class for loading the inpatientCharges

```scala
//case class for In Patient details
case class InPatientCharges (DRGDefinition :String, ProviderId:Long,
ProviderName:String, ProviderStreetAddress:String,
ProviderCity:String,
```

```
  ProviderState:String, ProviderZipCode:Int,
HospitalReferralRegionDescription:String, TotalDischarges:Int,
  AverageCoveredCharges:Double, AverageTotalPayments:Double,
AverageMedicarePayments:Double)
```

b. Loaded the data into RDD by using the 'textFile' method as below

Note : To load the csv file using textFile method , we had to replace the comma which was present in content of file (apart from column separator) by any other special character here we took it as semicolon (;).

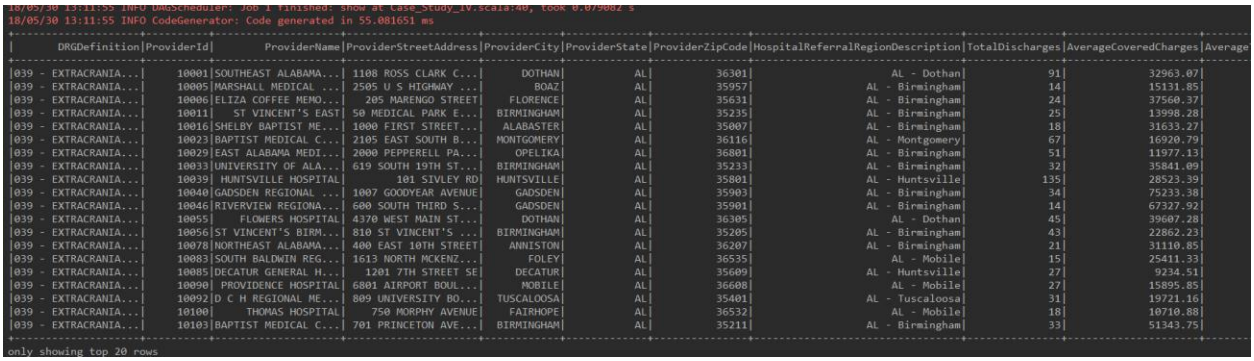
Task 1 : Load the file into Spark

```
6. //Load the csv file into the RDD and create Dataframe from the
   same after mapping with case class
   val data = spark.sparkContext.textFile("E:\\Prachi
   IMP\\Hadoop\\Case Studies - Assignment\\Case Study
   IV\\inpatientCharges.csv")

   import spark.implicits._
   //Create Dataframe
   val inPatientChargesDF = data.map(_.split(",")).map(x=>
   InPatientCharges(DRGDefinition=x(0),ProviderId=x(1).toLong,
                          ProviderName =
   x(2),ProviderStreetAddress=x(3),
   ProviderCity=x(4),ProviderState=x(5),ProviderZipCode=x(6).toInt,

   HospitalReferralRegionDescription=x(7),TotalDischarges=x(8).toInt
   ,AverageCoveredCharges=x(9).toDouble,

   AverageTotalPayments=x(10).toDouble,AverageMedicarePayments=x(11)
   .toDouble)).toDF()
```

c. Show the content of the Dataframe

```
 //Task 1 : Loaded the data and displaying Rows of the Data
inPatientChargesDF.show()
```

Output



18/05/30 13:11:55 INFO CodeGenerator: Code generated in 55.081651 ms

DRGDefinition	ProviderId	ProviderName	ProviderStreetAddress	ProviderCity	ProviderState	ProviderZipCode	HospitalReferralRegionDescription	TotalDischarges	AverageCoveredCharges	Average
039 - EXTRACRANIA...	10001	SOUTHEAST ALABAMA...	1108 ROSS CLARK C...	DOTHAN	AL	36301	AL - Dothan	91	32963.07	
039 - EXTRACRANIA...	10005	MARSHALL MEDICAL ...	2505 U S HIGHWAY ...	BOAZ	AL	35957	AL - Birmingham	14	15131.85	
039 - EXTRACRANIA...	10006	ELIZA COFFEE MEMO...	205 MARENGO STREET	FLORENCE	AL	35631	AL - Birmingham	24	37560.37	
039 - EXTRACRANIA...	10011	ST VINCENT'S EAST	50 MEDICAL PARK E...	BIRMINGHAM	AL	35235	AL - Birmingham	25	13998.28	
039 - EXTRACRANIA...	10016	SHELBY BAPTIST ME...	1000 FIRST STREET...	ALABASTER	AL	35007	AL - Birmingham	18	31633.27	
039 - EXTRACRANIA...	10023	BAPTIST MEDICAL C...	2105 EAST SOUTH B...	MONTGOMERY	AL	36116	AL - Montgomery	67	16920.79	
039 - EXTRACRANIA...	10029	EAST ALABAMA MEDI...	2000 PEPPERELL PA...	OPELIKA	AL	36801	AL - Birmingham	51	11977.13	
039 - EXTRACRANIA...	10033	UNIVERSITY OF ALA...	619 SOUTH 19TH ST...	BIRMINGHAM	AL	35233	AL - Birmingham	32	35841.09	
039 - EXTRACRANIA...	10039	HUNTSVILLE HOSPITAL	101 SIVLEY RD	HUNTSVILLE	AL	35801	AL - Huntsville	135	28523.39	
039 - EXTRACRANIA...	10040	GADSDEN REGIONAL ...	1007 GOODYEAR AVENUE	GADSDEN	AL	35903	AL - Birmingham	34	75233.38	
039 - EXTRACRANIA...	10046	RIVERVIEW REGIONA...	600 SOUTH THIRD S...	GADSDEN	AL	35901	AL - Birmingham	14	67327.92	
039 - EXTRACRANIA...	10055	FLOWERS HOSPITAL	4370 WEST MAIN ST...	DOTHAN	AL	36305	AL - Dothan	45	39607.28	
039 - EXTRACRANIA...	10056	ST VINCENT'S BIRM...	810 ST VINCENT'S ...	BIRMINGHAM	AL	35205	AL - Birmingham	43	22862.23	
039 - EXTRACRANIA...	10078	NORTHEAST ALABAMA...	400 EAST 10TH STREET	ANNISTON	AL	36207	AL - Birmingham	21	31110.85	
039 - EXTRACRANIA...	10083	SOUTH BALDWIN REG...	1613 NORTH MCKENZ...	FOLEY	AL	36535	AL - Mobile	15	25411.33	
039 - EXTRACRANIA...	10085	DECATUR GENERAL H...	1201 7TH STREET SE	DECATUR	AL	35609	AL - Huntsville	27	9234.51	
039 - EXTRACRANIA...	10090	PROVIDENCE HOSPITAL	6801 AIRPORT BOUL...	MOBILE	AL	36608	AL - Mobile	27	15895.85	
039 - EXTRACRANIA...	10092	D C H REGIONAL ME...	809 UNIVERSITY BO...	TUSCALOOSA	AL	35401	AL - Tuscaloosa	31	19721.16	
039 - EXTRACRANIA...	10100	THOMAS HOSPITAL	750 MORPHY AVENUE	FAIRHOPE	AL	36532	AL - Mobile	18	10710.88	
039 - EXTRACRANIA...	10103	BAPTIST MEDICAL C...	701 PRINCETON AVE...	BIRMINGHAM	AL	35211	AL - Birmingham	33	51343.75	

only showing top 20 rows

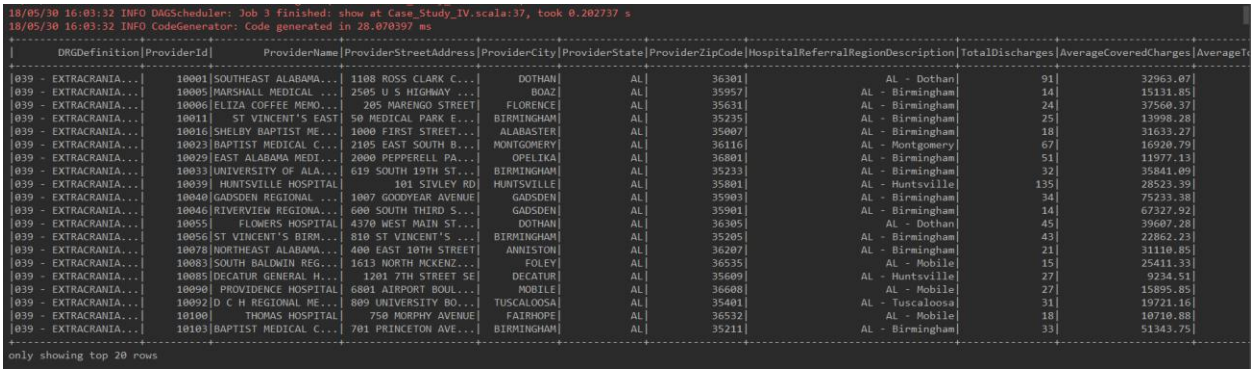
Approach 2: Programmatically Specifying the Schema

1. However we have not specified schema here and reading the csv file with option as *inferSchema = true* which indicates read the column as 'String' datatype by default.

Code

```
//Approach 2 With spark.read.format
val data1 = spark.read.format("CSV")
  .option("header",true)
  .option("inferSchema",true)
  .load("E:\\Prachi IMP\\Hadoop\\Case Studies - Assignment\\Case Study IV\\inpatientCharges1.csv")

data1.show()
```

Output



18/05/30 16:03:32 INFO DAGScheduler: Job 3 finished: show at Case_Study_IV.scala:37, took 0.202737 s
18/05/30 16:03:32 INFO CodeGenerator: Code generated in 28.070397 ms

DRGDefinition	ProviderId	ProviderName	ProviderStreetAddress	ProviderCity	ProviderState	ProviderZipCode	HospitalReferralRegionDescription	TotalDischarges	AverageCoveredCharges	AverageTo
039 - EXTRACRANIA...	10001	SOUTHEAST ALABAMA...	1108 ROSS CLARK C...	DOTHAN	AL	36301	AL - Dothan	91	32963.07	
039 - EXTRACRANIA...	10005	MARSHALL MEDICAL ...	2505 U S HIGHWAY ...	BOAZ	AL	35957	AL - Birmingham	14	15131.85	
039 - EXTRACRANIA...	10006	ELIZA COFFEE MEMO...	205 MARENGO STREET	FLORENCE	AL	35631	AL - Birmingham	24	37560.37	
039 - EXTRACRANIA...	10011	ST VINCENT'S EAST	50 MEDICAL PARK E...	BIRMINGHAM	AL	35235	AL - Birmingham	25	13998.28	
039 - EXTRACRANIA...	10016	SHELBY BAPTIST ME...	1000 FIRST STREET...	ALABASTER	AL	35007	AL - Birmingham	18	31633.27	
039 - EXTRACRANIA...	10023	BAPTIST MEDICAL C...	2105 EAST SOUTH B...	MONTGOMERY	AL	36116	AL - Montgomery	67	16920.79	
039 - EXTRACRANIA...	10029	EAST ALABAMA MEDI...	2000 PEPPERELL PA...	OPELIKA	AL	36801	AL - Birmingham	51	11977.13	
039 - EXTRACRANIA...	10033	UNIVERSITY OF ALA...	619 SOUTH 19TH ST...	BIRMINGHAM	AL	35233	AL - Birmingham	32	35841.09	
039 - EXTRACRANIA...	10039	HUNTSVILLE HOSPITAL	101 SIVLEY RD	HUNTSVILLE	AL	35801	AL - Huntsville	135	28523.39	
039 - EXTRACRANIA...	10040	GADSDEN REGIONAL ...	1007 GOODYEAR AVENUE	GADSDEN	AL	35903	AL - Birmingham	34	75233.38	
039 - EXTRACRANIA...	10046	RIVERVIEW REGIONA...	600 SOUTH THIRD S...	GADSDEN	AL	35901	AL - Birmingham	14	67327.92	
039 - EXTRACRANIA...	10055	FLOWERS HOSPITAL	4370 WEST MAIN ST...	DOTHAN	AL	36305	AL - Dothan	45	39607.28	
039 - EXTRACRANIA...	10056	ST VINCENT'S BIRM...	810 ST VINCENT'S ...	BIRMINGHAM	AL	35205	AL - Birmingham	43	22862.23	
039 - EXTRACRANIA...	10078	NORTHEAST ALABAMA...	400 EAST 10TH STREET	ANNISTON	AL	36207	AL - Birmingham	21	31110.85	
039 - EXTRACRANIA...	10083	SOUTH BALDWIN REG...	1613 NORTH MCKENZ...	FOLEY	AL	36535	AL - Mobile	15	25411.33	
039 - EXTRACRANIA...	10085	DECATUR GENERAL H...	1201 7TH STREET SE	DECATUR	AL	35609	AL - Huntsville	27	9234.51	
039 - EXTRACRANIA...	10090	PROVIDENCE HOSPITAL	6801 AIRPORT BOUL...	MOBILE	AL	36608	AL - Mobile	27	15895.85	
039 - EXTRACRANIA...	10092	D C H REGIONAL ME...	809 UNIVERSITY BO...	TUSCALOOSA	AL	35401	AL - Tuscaloosa	31	19721.16	
039 - EXTRACRANIA...	10100	THOMAS HOSPITAL	750 MORPHY AVENUE	FAIRHOPE	AL	36532	AL - Mobile	18	10710.88	
039 - EXTRACRANIA...	10103	BAPTIST MEDICAL C...	701 PRINCETON AVE...	BIRMINGHAM	AL	35211	AL - Birmingham	33	51343.75	

only showing top 20 rows

Task 2 What is the average amount of AverageCoveredCharges per state

Solution Approach –

1. To get the average per state we have to group the dataframe on state and get the sum of AverageCoveredCharges and count of AverageCoveredCharges
2. To get the *Average amount of AverageCoveredCharges = sum Of AverageCoveredCharges / count of AverageCoveredCharges per state*
3. This can be achived using avg function with group by function

Approach 1 : Using SPARK SQL Transformations → Used avg and groupby

```
//Task 2 : What is the average amount of AverageCoveredCharges per
state
//Approach 1 : Using SQL Spark Transformations
inPatientChargesDF.groupBy("ProviderState").avg("AverageCoveredCharges
").show()
```

Output

```
18/05/30 13:12:57 INFO CodeGenerator: Code genera
+-------------+--------------------------+
|ProviderState|avg(AverageCoveredCharges)|
+-------------+--------------------------+
|           AZ|        41200.063019992995|
|           SC|         35862.49456269756|
|           LA|        33085.372791542846|
|           MN|         27894.36182060388|
|           NJ|         66125.68627434729|
|           DC|         40116.66365800864|
|           OR|        27390.111870669723|
|           VA|        29222.000487072903|
|           RI|        29942.701122448976|
|           KY|         24523.80716940223|
|           WY|         28700.59862348178|
|           NH|        27059.020801944105|
|           MI|        24124.247209817277|
|           NV|         61047.11541597337|
|           WI|        26149.325331686607|
|           ID|        25565.547041742288|
|           CA|           67508.616535517|
|           CT|          31318.4101143709|
|           NE|        31736.427824858758|
|           MT|        22670.015237154144|
+-------------+--------------------------+
only showing top 20 rows
```

Approach 2: Using SQL Queries

Code

```
//Approach 2: Using the SQL Query
inPatientChargesDF.createOrReplaceTempView("InPatientCharges_Details")
spark.sql("Select ProviderState , avg(AverageCoveredCharges) from
InPatientCharges_Details  group by ProviderState").show()
```

Output

```
18/05/30 13:16:41 INFO DAGScheduler: ResultStage 17 (
18/05/30 13:16:41 INFO DAGScheduler: Job 9 finished:
+-------------+--------------------------+
|ProviderState|avg(AverageCoveredCharges)|
+-------------+--------------------------+
|           AZ|        41200.063019992995|
|           SC|         35862.49456269756|
|           LA|        33085.372791542846|
|           MN|         27894.36182060388|
|           NJ|         66125.68627434729|
|           DC|         40116.66365800864|
|           OR|        27390.111870669723|
|           VA|        29222.000487072903|
|           RI|        29942.701122448976|
|           KY|         24523.80716940223|
|           WY|         28700.59862348178|
|           NH|        27059.020801944105|
|           MI|        24124.247209817277|
|           NV|         61047.11541597337|
|           WI|        26149.325331686607|
|           ID|        25565.547041742288|
|           CA|           67508.616535517|
|           CT|          31318.4101143709|
|           NE|        31736.427824858758|
|           MT|        22670.015237154144|
+-------------+--------------------------+
only showing top 20 rows

18/05/30 13:16:41 INFO SparkContext: Invoking stop()
18/05/30 13:16:41 INFO SparkUI: Stopped Spark web UI
```

Task 3 find out the AverageTotalPayments charges per state

1. To get the average per state we have to group the dataframe on state and get the sum of AverageTotalPayments and count of AverageTotalPayments
2. To get the *Average amount of AverageTotalPayments = sum Of AverageTotalPayments / count of AverageTotalPayments per state*
3. This can be achieved using avg function with group by function

Approach 1 : Using SPARK SQL Transformations → Used avg and groupby

```
//Task 3 : What is the average amount of AverageTotalPayments  per
state
//Approach 1 : Using SQL Spark Transformations
inPatientChargesDF.groupBy("ProviderState").avg("AverageTotalPayments"
).show()
```

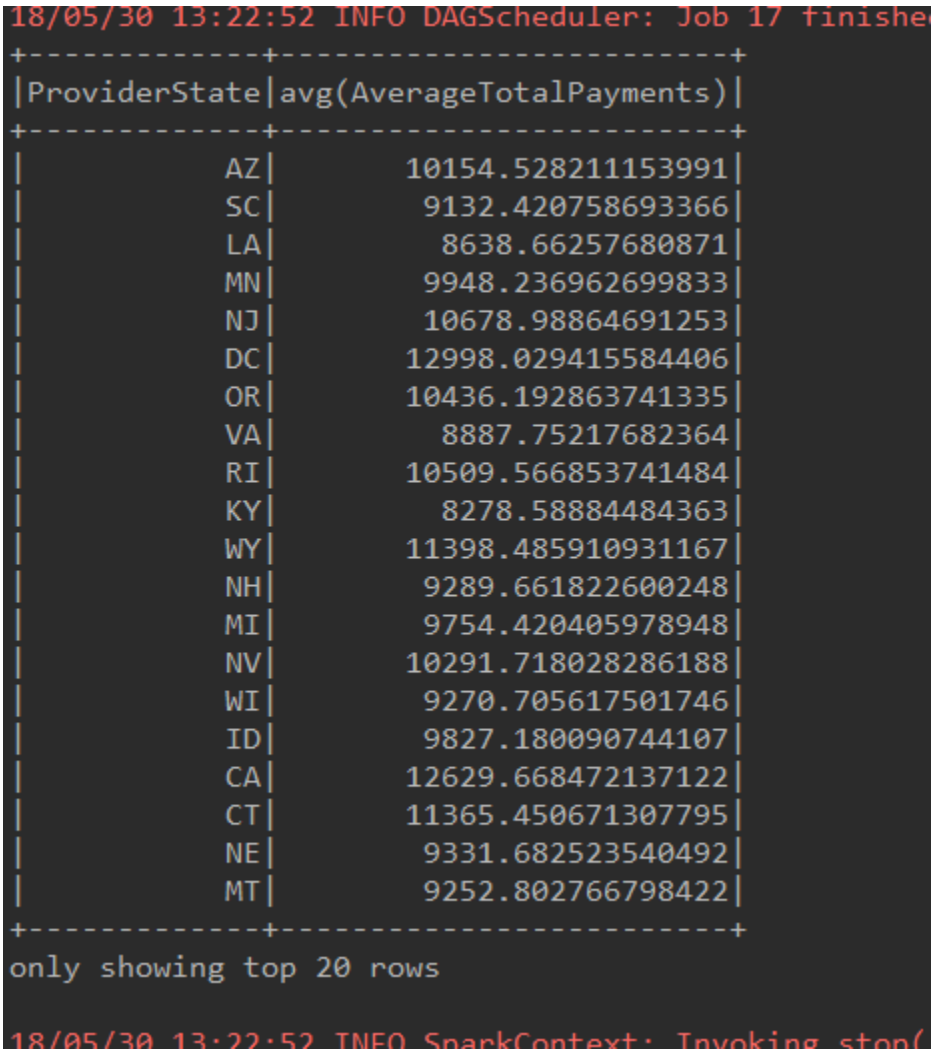
Output

```
18/05/30 13:22:50 INFO DAGScheduler: Job 13 finishe
+-------------+--------------------------+
|ProviderState|avg(AverageTotalPayments)|
+-------------+--------------------------+
|           AZ|       10154.528211153991|
|           SC|        9132.420758693366|
|           LA|         8638.66257680871|
|           MN|        9948.236962699833|
|           NJ|        10678.98864691253|
|           DC|       12998.029415584406|
|           OR|       10436.192863741335|
|           VA|         8887.75217682364|
|           RI|       10509.566853741484|
|           KY|         8278.58884484363|
|           WY|       11398.485910931167|
|           NH|        9289.661822600248|
|           MI|        9754.420405978948|
|           NV|       10291.718028286188|
|           WI|        9270.705617501746|
|           ID|        9827.180090744107|
|           CA|       12629.668472137122|
|           CT|       11365.450671307795|
|           NE|        9331.682523540492|
|           MT|        9252.802766798422|
+-------------+--------------------------+
only showing top 20 rows

18/05/30 13:22:50 INFO SparkSqlParser: Parsing comm
18/05/30 13:22:50 INFO SparkSqlParser: Parsing comm
```

Approach 2: Using SQL Queries

```
//Approach 2: Using the SQL Query
`spark.sql("Select ProviderState , avg(AverageTotalPayments) from
InPatientCharges_Details  group by ProviderState").show()
```

Output



```
18/05/30 13:22:52 INFO DAGScheduler: Job 17 finishe
+-------------+-------------------------+
|ProviderState|avg(AverageTotalPayments)|
+-------------+-------------------------+
|           AZ|       10154.528211153991|
|           SC|        9132.420758693366|
|           LA|         8638.66257680871|
|           MN|        9948.236962699833|
|           NJ|        10678.98864691253|
|           DC|       12998.029415584406|
|           OR|       10436.192863741335|
|           VA|         8887.75217682364|
|           RI|       10509.566853741484|
|           KY|         8278.58884484363|
|           WY|       11398.485910931167|
|           NH|        9289.661822600248|
|           MI|        9754.420405978948|
|           NV|       10291.718028286188|
|           WI|        9270.705617501746|
|           ID|        9827.180090744107|
|           CA|       12629.668472137122|
|           CT|       11365.450671307795|
|           NE|        9331.682523540492|
|           MT|        9252.802766798422|
+-------------+-------------------------+
only showing top 20 rows

18/05/30 13:22:52 INFO SparkContext: Invoking stop(
```

Task 4 find out the AverageMedicarePayments charges per state.

1. To get the average per state we have to group the dataframe on state and get the sum of AverageMedicarePayments and count of AverageMedicarePayments
2. *To get the Average amount of AverageMedicarePayments = sum Of AverageMedicarePayments / count of AverageMedicarePayments per state*
3. This can be achieved using avg function with group by function

Approach 1 : Using SPARK SQL Transformations → Used avg and groupby

```
//Task 4 : find out the AverageMedicarePayments charges per state.
//Approach 1 : Using SQL Spark Transformations
inPatientChargesDF.groupBy("ProviderState").avg("AverageMedicarePaymen
ts").show()
```

Output

```
18/05/30 15:23:21 INFO SparkSqlParser: Parsing commanc
+-------------+----------------------------+
|ProviderState|avg(AverageMedicarePayments)|
+-------------+----------------------------+
|           AZ|           8825.717239565045|
|           SC|            7876.33152441167|
|           LA|           7387.704625041281|
|           MN|           8619.214982238007|
|           NJ|           9586.940055946912|
|           DC|          11811.967705627709|
|           OR|           9035.259961508847|
|           VA|           7538.847006001846|
|           RI|           9317.939115646255|
|           KY|           7185.227810467647|
|           WY|           9539.392024291496|
|           NH|           8124.506852976913|
|           MI|           8662.157756043543|
|           NV|           8747.602828618963|
|           WI|           8002.597911079731|
|           ID|           8461.977513611617|
|           CA|          11494.381677893474|
|           CT|          10104.592943809059|
|           NE|          7992.6272504707995|
|           MT|           7981.088063241104|
+-------------+----------------------------+
only showing top 20 rows

18/05/30 15:23:21 INFO SparkContext: Starting job: sh
```

Approach 2: Using SQL Queries

```
//Approach 2: Using the SQL Query
 spark.sql("Select ProviderState , avg(AverageMedicarePayments) from
InPatientCharges_Details  group by ProviderState").show()
```

Output

```
18/05/30 15:23:22 INFO DAGScheduler: ResultStage 49 (show
18/05/30 15:23:22 INFO DAGScheduler: Job 25 finished: show
+-------------+---------------------------+
|ProviderState|avg(AverageMedicarePayments)|
+-------------+---------------------------+
|           AZ|           8825.717239565045|
|           SC|            7876.33152441167|
|           LA|           7387.704625041281|
|           MN|           8619.214982238007|
|           NJ|           9586.940055946912|
|           DC|          11811.967705627709|
|           OR|           9035.259961508847|
|           VA|           7538.847006001846|
|           RI|           9317.939115646255|
|           KY|           7185.227810467647|
|           WY|           9539.392024291496|
|           NH|           8124.506852976913|
|           MI|           8662.157756043543|
|           NV|           8747.602828618963|
|           WI|           8002.597911079731|
|           ID|           8461.977513611617|
|           CA|          11494.381677893474|
|           CT|          10104.592943809059|
|           NE|          7992.6272504707995|
|           MT|           7981.088063241104|
+-------------+---------------------------+
only showing top 20 rows

18/05/30 15:23:22 INFO SparkContext: Invoking stop() from
```

Task 5 Find out the total number of Discharges per state and for each disease

Solution Approach -

1. To get the total number of diseases we need to group the table o state and disease and get the sum of total discharges. The output of this tasked is saved for task 6.

Approach 1: Using SPARK SQL Transformations

```
//Task 5 Find out the total number of Discharges per state and for
each disease
//Approach 1 : Using SQL Spark Transformations
val task5Output=
inPatientChargesDF.groupBy("ProviderState","DRGDefinition").sum("Total
Discharges").
  withColumnRenamed("sum(TotalDischarges)","sum")
task5Output.show()
```

Output

```
18/05/30 15:48:11 INFO DAGScheduler: Job 26 finishe
+-------------+--------------------+-----+
|ProviderState|       DRGDefinition|  sum|
+-------------+--------------------+-----+
|           KY|065 - INTRACRANIA...| 1937|
|           NY|101 - SEIZURES W/...| 4503|
|           IN|149 - DYSEQUILIBRIUM|  700|
|           IA|178 - RESPIRATORY...|  540|
|           WI|202 - BRONCHITIS ...|  338|
|           MO|208 - RESPIRATORY...| 1840|
|           WI|251 - PERC CARDIO...|  417|
|           IA|280 - ACUTE MYOCA...|  692|
|           AZ|292 - HEART FAILU...| 2643|
|           NY|292 - HEART FAILU...|13289|
|           NV|293 - HEART FAILU...|  519|
|           SD|303 - ATHEROSCLER...|   53|
|           TN|305 - HYPERTENSIO...|  730|
|           ME|308 - CARDIAC ARR...|  312|
|           NV|372 - MAJOR GASTR...|  126|
|           WI|439 - DISORDERS O...|  215|
|           MN|536 - FRACTURES O...|  332|
|           CO|602 - CELLULITIS ...|   86|
|           OR|603 - CELLULITIS ...|  680|
|           DE|640 - MISC DISORD...|  199|
+-------------+--------------------+-----+
only showing top 20 rows
```

Approach 2: Using SQL Query

```
//Approach 2 : Using SQL Query
spark.sql("select ProviderState, DRGDefinition,sum(TotalDischarges)
from InPatientCharges_Details group by
ProviderState,DRGDefinition").show()
```

Output

```
18/05/30 15:48:12 INFO ShuffleBlockFetcherIterator: Started 0 remot
+-------------+--------------------+--------------------+
|ProviderState|       DRGDefinition|sum(TotalDischarges)|
+-------------+--------------------+--------------------+
|           KY|065 - INTRACRANIA...|                1937|
|           NY|101 - SEIZURES W/...|                4503|
|           IN|149 - DYSEQUILIBRIUM|                 700|
|           IA|178 - RESPIRATORY...|                 540|
|           WI|202 - BRONCHITIS ...|                 338|
|           MO|208 - RESPIRATORY...|                1840|
|           WI|251 - PERC CARDIO...|                 417|
|           IA|280 - ACUTE MYOCA...|                 692|
|           AZ|292 - HEART FAILU...|                2643|
|           NY|292 - HEART FAILU...|               13289|
|           NV|293 - HEART FAILU...|                 519|
|           SD|303 - ATHEROSCLER...|                  53|
|           TN|305 - HYPERTENSIO...|                 730|
|           ME|308 - CARDIAC ARR...|                 312|
|           NV|372 - MAJOR GASTR...|                 126|
|           WI|439 - DISORDERS O...|                 215|
|           MN|536 - FRACTURES O...|                 332|
|           CO|602 - CELLULITIS ...|                  86|
|           OR|603 - CELLULITIS ...|                 680|
|           DE|640 - MISC DISORD...|                 199|
+-------------+--------------------+--------------------+
only showing top 20 rows

18/05/30 15:48:12 WARN Executor: Managed memory leak detected; size
18/05/30 15:48:12 INFO Executor: Finished task 0.0 in stage 53.0 (T
```

Task 6 Sort the output in descending order of totalDischarges

Solution Approach –

1. Need to sort the dataframe on column sum of 'totalDischarges'
2. The output was saved in dataframe named as *'task5Output'*

Approach 1: Using SPARK Transformations

```
//Task 6 Sort the output in descending order of totalDischarges
//Approach 1: Using SQL Spark Transformations
task5Output.sort(desc("sum")).show()
//OR
task5Output.orderBy(($"sum").desc).show()
```

Output

```
18/05/30 15:48:13 INFO DAGScheduler: ResultStage 55 (
18/05/30 15:48:13 INFO DAGScheduler: Job 28 finished:
+-------------+--------------------+-----+
|ProviderState|       DRGDefinition|  sum|
+-------------+--------------------+-----+
|           CA|871 - SEPTICEMIA ...|34284|
|           TX|470 - MAJOR JOINT...|30095|
|           FL|470 - MAJOR JOINT...|29985|
|           CA|470 - MAJOR JOINT...|29731|
|           TX|871 - SEPTICEMIA ...|23144|
|           NY|871 - SEPTICEMIA ...|21970|
|           FL|392 - ESOPHAGITIS...|21298|
|           IL|470 - MAJOR JOINT...|20095|
|           NY|470 - MAJOR JOINT...|19371|
|           FL|871 - SEPTICEMIA ...|18660|
|           TX|690 - KIDNEY & UR...|17384|
|           NY|392 - ESOPHAGITIS...|17337|
|           MI|470 - MAJOR JOINT...|16847|
|           PA|470 - MAJOR JOINT...|16712|
|           FL|292 - HEART FAILU...|16639|
|           FL|690 - KIDNEY & UR...|16405|
|           OH|470 - MAJOR JOINT...|16062|
|           NC|470 - MAJOR JOINT...|15820|
|           IL|871 - SEPTICEMIA ...|15610|
|           MI|871 - SEPTICEMIA ...|15548|
+-------------+--------------------+-----+
only showing top 20 rows

18/05/30 15:48:13 INFO SparkSqlParser: Parsing comman
18/05/30 15:48:13 INFO SparkSqlParser: Parsing comman
18/05/30 15:48:13 INFO SparkContext: Starting job: sh
```

Approach 2: Using SQL Query

```
//Approach 2 : Using SQL Query
task5Output.createOrReplaceTempView("task5Output_Table")
spark.sql("select * from task5Output_Table order by sum desc").show()
```

Output

```
18/05/30 15:48:14 INFO DAGScheduler: ResultStage 57 (show at
18/05/30 15:48:14 INFO DAGScheduler: Job 29 finished: show a
+-------------+--------------------+-----+
|ProviderState|       DRGDefinition|  sum|
+-------------+--------------------+-----+
|           CA|871 - SEPTICEMIA ...|34284|
|           TX|470 - MAJOR JOINT...|30095|
|           FL|470 - MAJOR JOINT...|29985|
|           CA|470 - MAJOR JOINT...|29731|
|           TX|871 - SEPTICEMIA ...|23144|
|           NY|871 - SEPTICEMIA ...|21970|
|           FL|392 - ESOPHAGITIS...|21298|
|           IL|470 - MAJOR JOINT...|20095|
|           NY|470 - MAJOR JOINT...|19371|
|           FL|871 - SEPTICEMIA ...|18660|
|           TX|690 - KIDNEY & UR...|17384|
|           NY|392 - ESOPHAGITIS...|17337|
|           MI|470 - MAJOR JOINT...|16847|
|           PA|470 - MAJOR JOINT...|16712|
|           FL|292 - HEART FAILU...|16639|
|           FL|690 - KIDNEY & UR...|16405|
|           OH|470 - MAJOR JOINT...|16062|
|           NC|470 - MAJOR JOINT...|15820|
|           IL|871 - SEPTICEMIA ...|15610|
|           MI|871 - SEPTICEMIA ...|15548|
+-------------+--------------------+-----+
only showing top 20 rows

18/05/30 15:48:14 INFO SparkContext: Invoking stop() from sh
18/05/30 15:48:14 INFO SparkUI: Stopped Spark web UI at http
18/05/30 15:48:14 INFO MapOutputTrackerMasterEndpoint: MapOu
18/05/30 15:48:14 INFO MemoryStore: MemoryStore cleared
18/05/30 15:48:14 INFO BlockManager: BlockManager stopped
18/05/30 15:48:14 INFO BlockManagerMaster: BlockManagerMaste
```