CASE STUDY 3 **Hospital Charges Data Analysis in US**

In this case study, we have an inpatientCharges.csv data set which have the following description:

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.

We have the following objectives to perform on the above data sets, which we will perform using SPARK-SQL in **IntelliJ IDEA** application to get the results

Objective-1: Load File into Spark

To load the inpatientCharges.csv file into the SPARK-SQL context we use the following programing in scala.

```
package Hospital_Data_Analysis

import org.apache.spark.sql.SparkSession


object Hospital_Objective1

{

def main(args: Array[String]): Unit = {

println("Hospital data analysis in US")

//Create the spark session

val spark = SparkSession.builder() .master("local[*]") .appName("Hospital Data Analysis in US")

    .config("spark.some.config.option", "some-value").getOrCreate()

//Use CSV load method  to load the data and use infer schema  as an option so it will
automatically infer the data type of the columns

val in_patient_charges = spark.read.format("com.databricks.spark.csv")

    .option("header", "true")

    .option("inferSchema", "true")

    .load("C:\\Users\\Bhaskar\\Desktop\\AcadGild\\CaseStudies\\CaseStudy4 Hospital\\inpatientCharges.csv")

///Use printSchema to check the schema of the loaded file

 in_patient_charges.printSchema()

//Save the data in table hospital_charges  by registering it in temp table

 in_patient_charges.registerTempTable("hospital_charges")

 } }
```

Below screen, shot shows the schema of temporary table **hospital_charges** which is same as data described above.

```
18/05/15 03:02:49 INFO BlockManagerInfo: Removed broadcast_2_piece0 on 192.168.56.1:563
root
 |-- DRGDefinition: string (nullable = true)
 |-- ProviderId: integer (nullable = true)
 |-- ProviderName: string (nullable = true)
 |-- ProviderStreetAddress: string (nullable = true)
 |-- ProviderCity: string (nullable = true)
 |-- ProviderState: string (nullable = true)
 |-- ProviderZipCode: integer (nullable = true)
 |-- HospitalReferralRegionDescription: string (nullable = true)
 |-- TotalDischarges: integer (nullable = true)
 |-- AverageCoveredCharges: double (nullable = true)
 |-- AverageTotalPayments: double (nullable = true)
 |-- AverageMedicarePayments: double (nullable = true)

18/05/15 03:02:50 INFO SparkSqlParser: Parsing command: hospital_charges
inpatientcharges data is registered in temporary table hospital-charges
18/05/15 03:02:50 INFO SparkContext: Invoking stop() from shutdown hook
```

Objective-2:

What is the average amount of Average Covered Charges per state.

Find out the Average Total Payments charges per state.

Find out the Average Medicare Payments charges per state.

All the above three problems of object2 are coded in the below scala program:

```
package Hospital_Data_Analysis

import org.apache.spark.sql.SparkSession

object Hospital_Objective2

{

  def main(args: Array[String]): Unit =

  {

println("Hospital data analysis in US")


    //Create the spark session

val spark = SparkSession .builder().master("local[*]").appName("Hospital Data Analysis in US")

      .config("spark.some.config.option", "some-value") .getOrCreate()


//Use CSV load method to load the data and use infer schema  as an option so it will automatically infer the data type of the columns

 val in_patient_charges = spark.read.format("com.databricks.spark.csv")

      .option("header", "true").option("inferSchema", "true")
```

```
.load("C:\\Users\\Bhaskar\\Desktop\\AcadGild\\CaseStudies\\CaseStudy4 Hospital\\inpatientCharges.csv")
```

//Use groupBy sql function to group ProviderState and average function to average AverageCoveredCharges respect to ProviderState

```
in_patient_charges.groupBy("ProviderState").avg("AverageCoveredCharges").show()
```

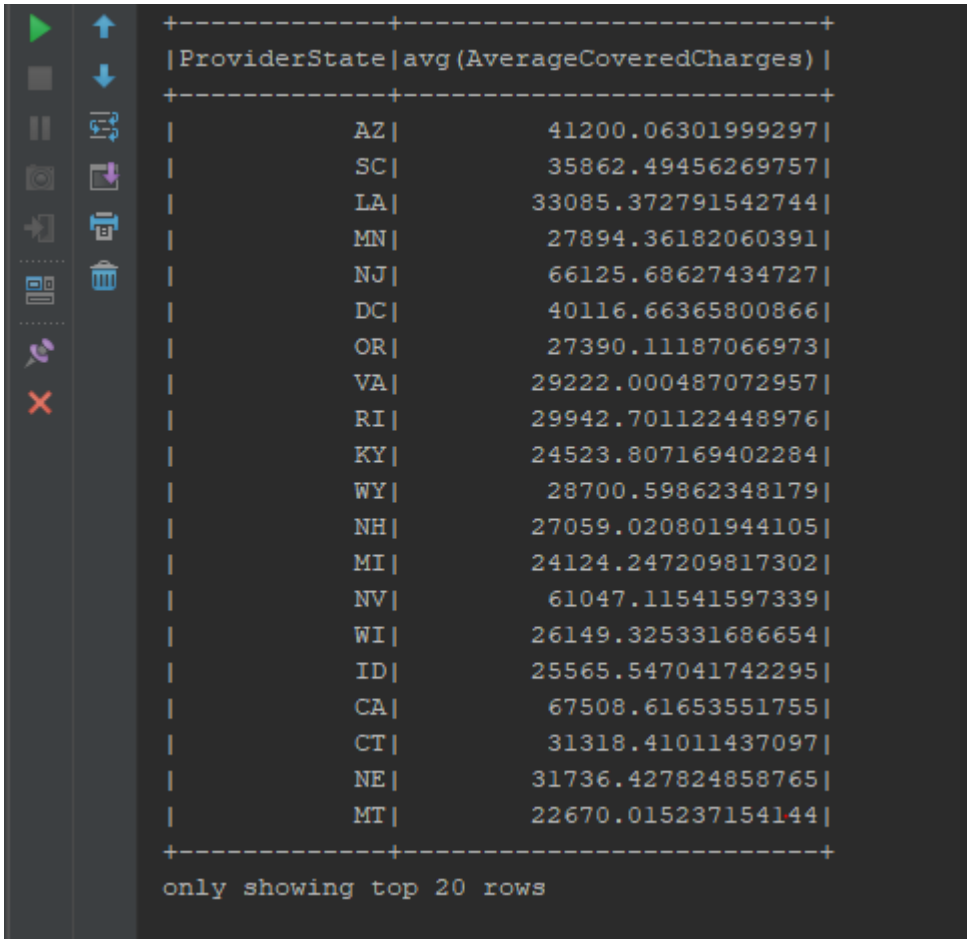
//Use groupBy sql function to group ProviderState and average function to average AverageTotalPayments respect to ProviderState

```
in_patient_charges.groupBy("ProviderState").avg("AverageTotalPayments").show()
```

//Use groupBy sql function to group ProviderState and average function to average AverageMedicarePayments respect to ProviderState

```
in_patient_charges.groupBy("ProviderState").avg("AverageMedicarePayments").show()
 }
}
```

Below screen, shot shows the average amount of **AverageCoveredCharges** per state



ProviderState	avg(AverageCoveredCharges)
AZ	41200.06301999297
SC	35862.49456269757
LA	33085.372791542744
MN	27894.36182060391
NJ	66125.68627434727
DC	40116.66365800866
OR	27390.11187066973
VA	29222.000487072957
RI	29942.701122448976
KY	24523.807169402284
WY	28700.59862348179
NH	27059.020801944105
MI	24124.247209817302
NV	61047.11541597339
WI	26149.325331686654
ID	25565.547041742295
CA	67508.61653551755
CT	31318.41011437097
NE	31736.427824858765
MT	22670.015237154144

only showing top 20 rows

Below screen, shot shows **AverageTortalPAyments** charges per state

```
+-------------+-------------------------+
|ProviderState|avg(AverageTotalPayments)|
+-------------+-------------------------+
|           AZ|       10154.528211153982|
|           SC|         9132.42075869336|
|           LA|        8638.662576808716|
|           MN|         9948.23696269982|
|           NJ|       10678.988646912556|
|           DC|       12998.029415584415|
|           OR|       10436.192863741338|
|           VA|        8887.752176823638|
|           RI|       10509.566853741495|
|           KY|          8278.5888448436|
|           WY|       11398.485910931175|
|           NH|        9289.661822600241|
|           MI|        9754.420405978964|
|           NV|       10291.718028286188|
|           WI|        9270.705617501762|
|           ID|        9827.180090744105|
|           CA|       12629.668472137168|
|           CT|       11365.450671307808|
|           NE|         9331.68252354049|
|           MT|        9252.802766798417|
+-------------+-------------------------+
only showing top 20 rows
```

Below screen, shot shows **AverageMedicarePayments** charges per state

```
+-------------+----------------------------+
|ProviderState|avg(AverageMedicarePayments)|
+-------------+----------------------------+
|           AZ|           8825.717239565063|
|           SC|           7876.331524411663|
|           LA|           7387.704625041294|
|           MN|            8619.21498223801|
|           NJ|            9586.94005594695|
|           DC|          11811.967705627703|
|           OR|           9035.259961508855|
|           VA|           7538.847006001844|
|           RI|            9317.93911564626|
|           KY|           7185.227810467634|
|           WY|             9539.3920242915|
|           NH|           8124.506852976911|
|           MI|           8662.157756043538|
|           NV|            8747.60282861897|
|           WI|           8002.597911079743|
|           ID|           8461.977513611615|
|           CA|          11494.381677893432|
|           CT|           10104.59294380905|
|           NE|           7992.627250470812|
|           MT|           7981.088063241105|
+-------------+----------------------------+
only showing top 20 rows
```

Objective-3

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

Sort the output in descending order of totalDischarges.

Above two problems are coded in the below in scala programme.

```
package Hospital_Data_Analysis

import org.apache.spark.sql.SparkSession

import org.apache.spark.sql.functions._

object Hospital_Objective3

{

 def main(args: Array[String]): Unit =

  {

 println("Hospital data analysis in US")

 //Create the spark session
```

```
 val spark = SparkSession.builder() .master("local[*]").appName("Hospital Data Analysis in US")
.config("spark.some.config.option", "some-value") .getOrCreate()


 //Use CSV load method  to load the data and use infer schema  as an option so it will
automatically infer the data type of the columns

val in_patient_charges = spark.read.format("com.databricks.spark.csv") .option("header", "true")
.option("inferSchema", "true") .load("C:\\Users\\Bhaskar\\Desktop\\AcadGild\\CaseStudies\\CaseStudy4
Hospital\\inpatientCharges.csv")


//Use groupBy sql function to group ProviderState and DRGDefinition and sum function to
sum up TotalDischarges value respect to ProviderState

 in_patient_charges.groupBy(("ProviderState"),("DRGDefinition")).sum("TotalDischarges").show()

println("The total number of Discharges per state and for each disease")


 //Use groupBy sql function to group ProviderState and DRGDefinition and sum function to
sum up TotalDischarges value respect to ProviderState

in_patient_charges.groupBy(("ProviderState"),("DRGDefinition")).sum("TotalDischarges").orderBy(desc(sum("
TotalDischarges").toString)).show()

println("The output in descending order of totalDischarges")

  }}
```

Below screen, shot shows total number of discharges per state and for each disease

```
+-------------+--------------------+--------------------+
|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|
|           AR|281 - ACUTE MYOCA...|                 413|
|           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|
|           WA|392 - ESOPHAGITIS...|                3148|
|           WI|439 - DISORDERS O...|                 215|
|           MN|536 - FRACTURES O...|                 332|
|           DC|563 - FX, SPRN, S...|                  43|
|           CO|602 - CELLULITIS ...|                  86|
+-------------+--------------------+--------------------+
only showing top 20 rows

The total number of Discharges per state and for each disease
```

Below screen, shot shows above output in sorted and in descending order

```
Run:  Hospital_Objective3
+-------------+--------------------+--------------------+
|ProviderState|       DRGDefinition|sum(TotalDischarges)|
+-------------+--------------------+--------------------+
|           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

The output in descending order of totalDischarges
```