Case Study _-_IV_Hospital_Analysis_in_US

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

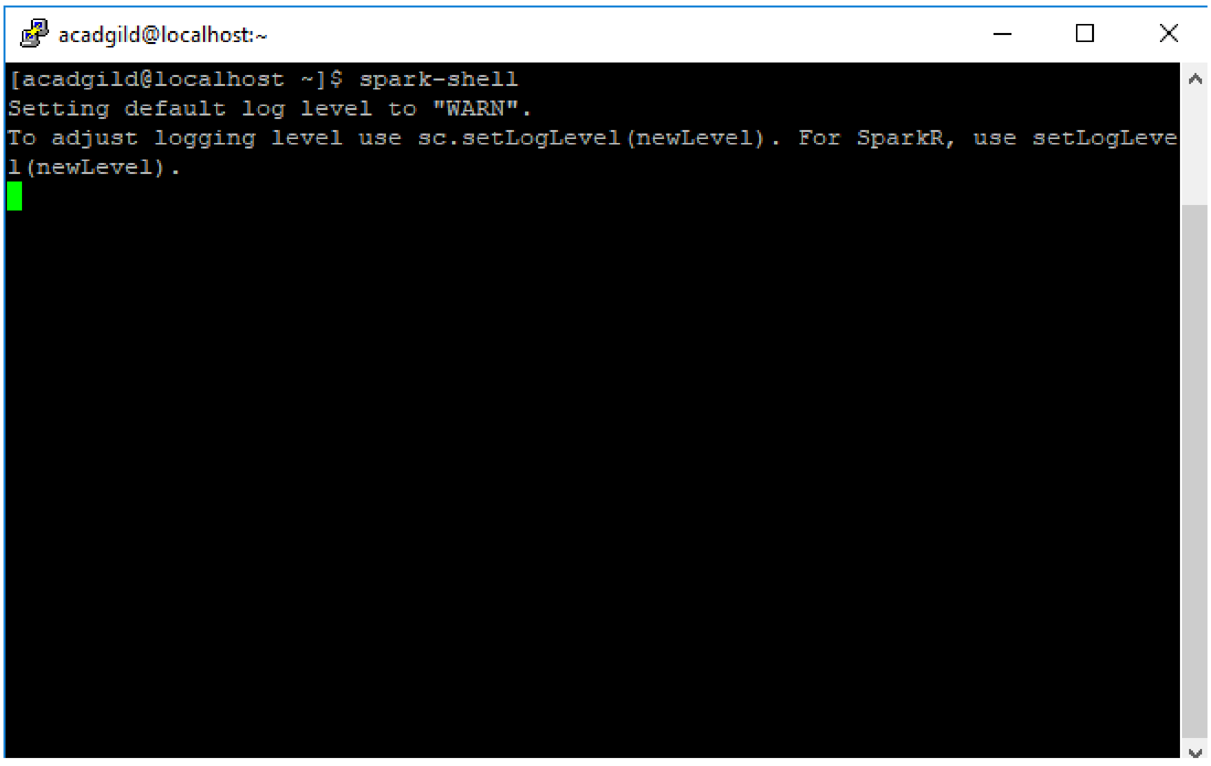
include beneficiary co-payments and deductible amounts nor any additional payments from third parties for coordination of benefits.

<u>Objective – 1</u>

Load file into spark

Step 1:

Start Spark-shell



```
[acadgild@localhost ~]$ spark-shell
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLeve
l(newLevel).
```

Step 2:

//Load File into Spark

val sqlContext = new org.apache.spark.sql.SQLContext(sc)

val data = sc.textFile("/user/acadgild/inpatientCharges.csv")

//Remove Header

val header = data.first()

val data1 = data.filter(row => row != header)

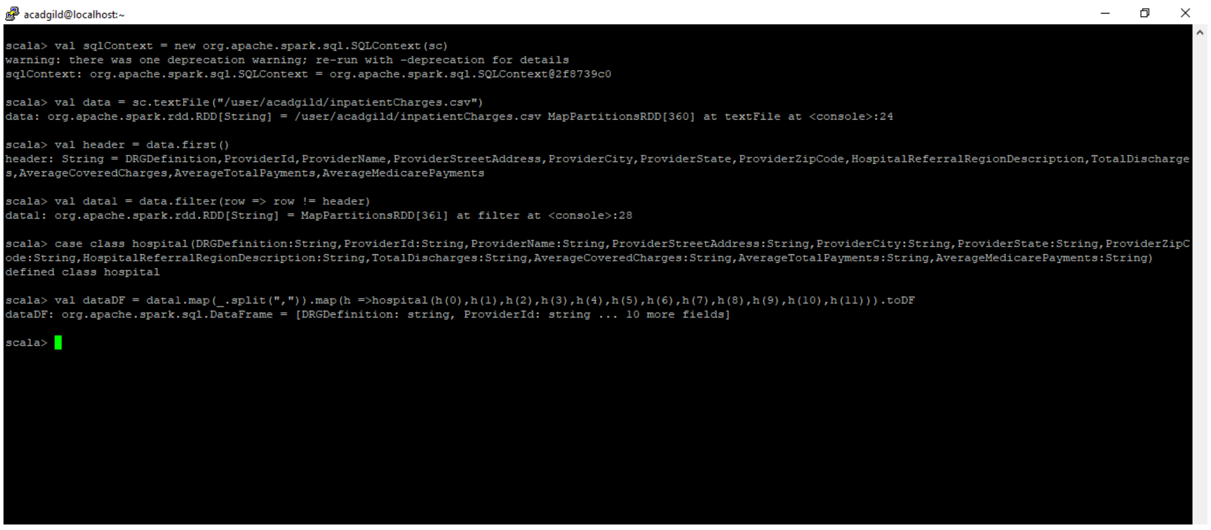
//Define case class Hospital

case class hospital(DRGDefinition:String,ProviderId:String,ProviderName:String,Provide

rStreetAddress:String,ProviderCity:String,ProviderState:String,ProviderZipCode:String,HospitalReferralRegionDescription:String,TotalDischarges:String,AverageCoveredCharges:String,AverageTotalPayments:String,AverageMedicarePayments:String)

//Convert to DataFrames

val dataDF = data1.map(_.split(",")).map(h =>hospital(h(0),h(1),h(2),h(3),h(4),h(5),h(6),h(7),h(8),h(9),h(10),h(11))).toDF

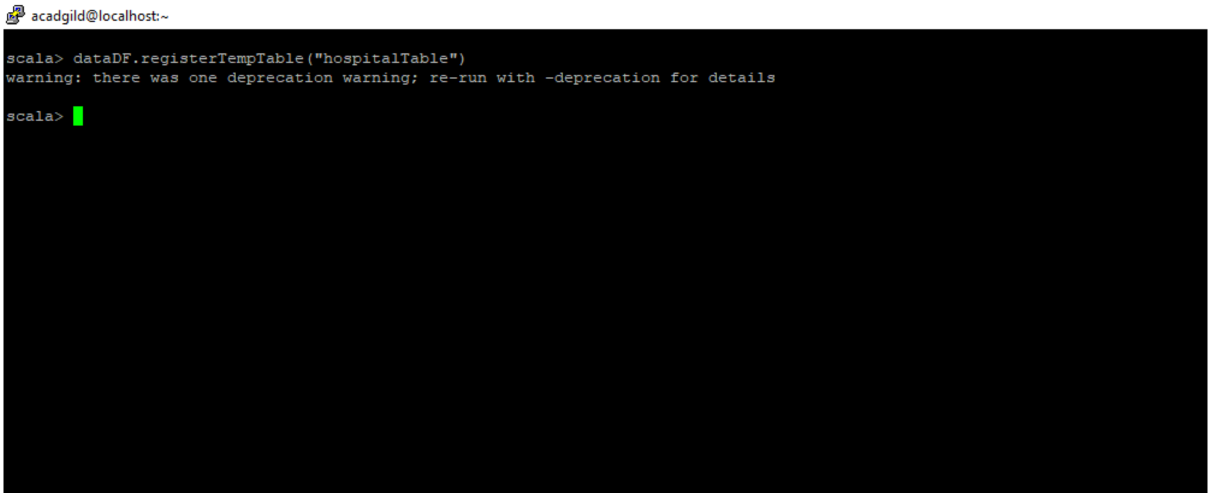


```
scala> val sqlContext = new org.apache.spark.sql.SQLContext(sc)
warning: there was one deprecation warning; re-run with -deprecation for details
sqlContext: org.apache.spark.sql.SQLContext = org.apache.spark.sql.SQLContext@2f8739c0

scala> val data = sc.textFile("/user/acadgild/inpatientCharges.csv")
data: org.apache.spark.rdd.RDD[String] = /user/acadgild/inpatientCharges.csv MapPartitionsRDD[360] at textFile at <console>:24

scala> val header = data.first()
header: String = DRGDefinition,ProviderId,ProviderName,ProviderStreetAddress,ProviderCity,ProviderState,ProviderZipCode,HospitalReferralRegionDescription,TotalDischarge
s,AverageCoveredCharges,AverageTotalPayments,AverageMedicarePayments

scala> val data1 = data.filter(row => row != header)
data1: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[361] at filter at <console>:28

scala> case class hospital(DRGDefinition:String,ProviderId:String,ProviderName:String,ProviderStreetAddress:String,ProviderCity:String,ProviderState:String,ProviderZipC
ode:String,HospitalReferralRegionDescription:String,TotalDischarges:String,AverageCoveredCharges:String,AverageTotalPayments:String,AverageMedicarePayments:String)
defined class hospital

scala> val dataDF = data1.map(_.split(",")).map(h =>hospital(h(0),h(1),h(2),h(3),h(4),h(5),h(6),h(7),h(8),h(9),h(10),h(11))).toDF
dataDF: org.apache.spark.sql.DataFrame = [DRGDefinition: string, ProviderId: string ... 10 more fields]

scala> 
```

Step 3:

//Loading Hospital.csv file into temporary table

dataDF.registerTempTable("hospitalTable")

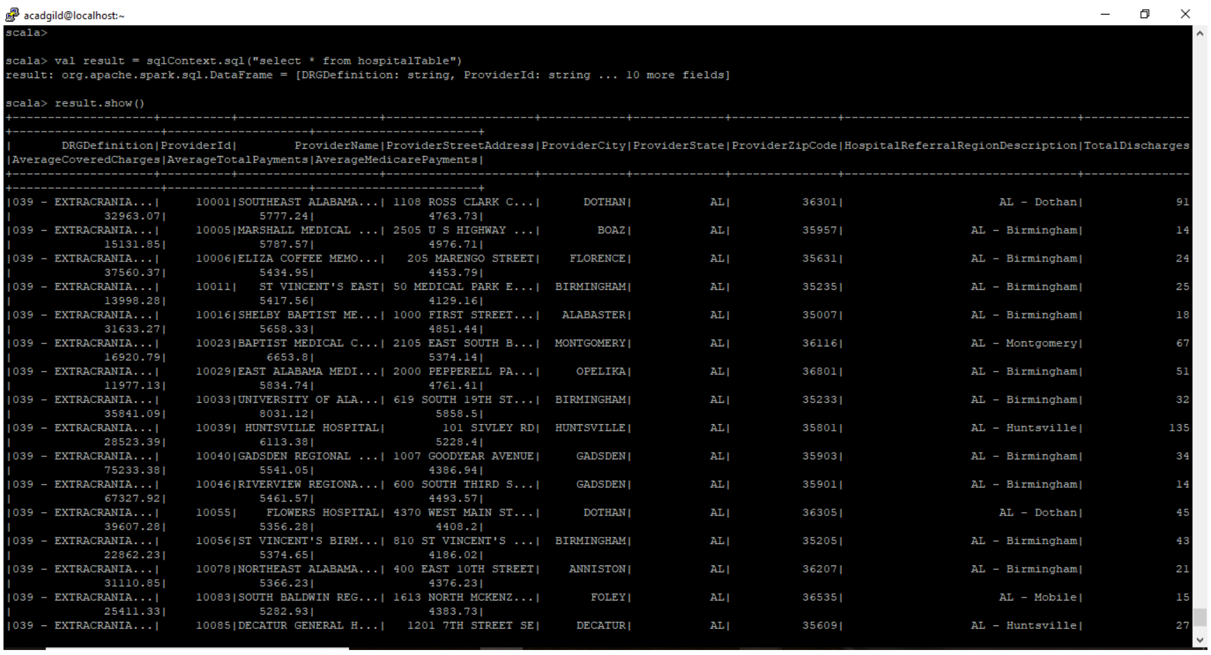


```
scala> dataDF.registerTempTable("hospitalTable")
warning: there was one deprecation warning; re-run with -deprecation for details

scala> 
```

Step 4:

val result = sqlContext.sql("select * from hospitalTable")

result.show()



acadgild@localhost:~

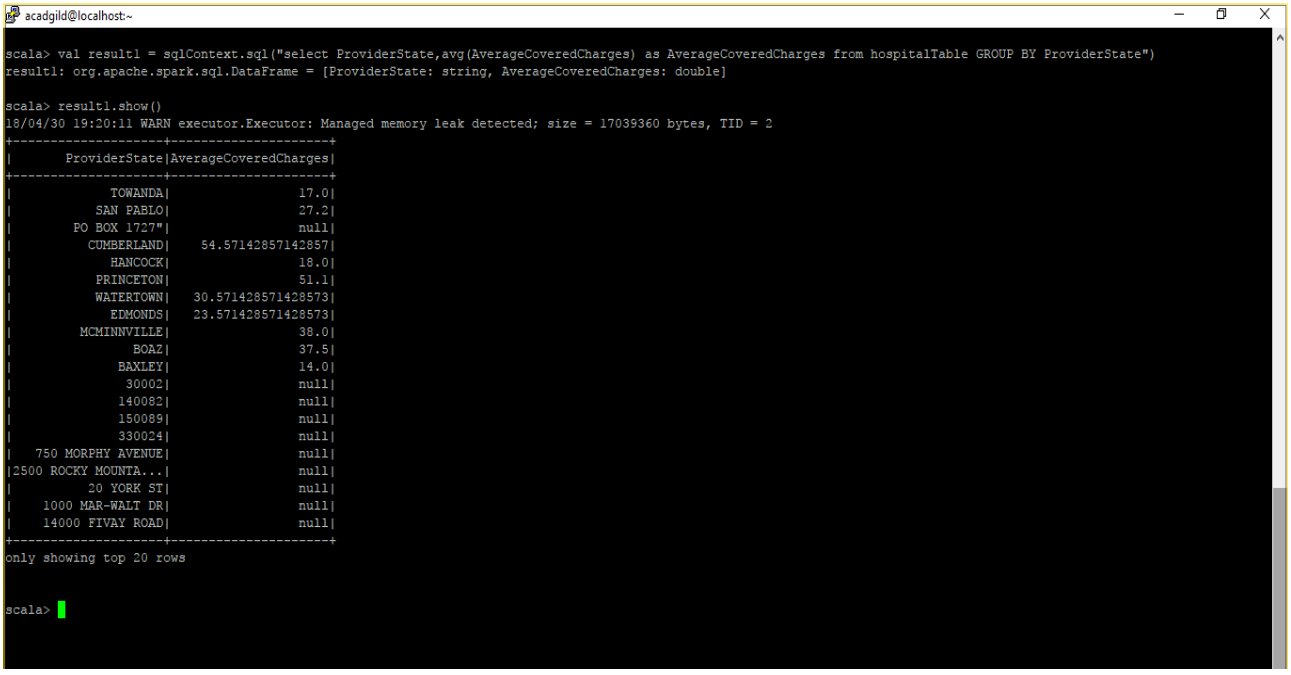
```
scala>

scala> val result = sqlContext.sql("select * from hospitalTable")
result: org.apache.spark.sql.DataFrame = [DRGDefinition: string, ProviderId: string ... 10 more fields]

scala> result.show()
+--------------------+----------+--------------------+---------------------+------------+-------------+---------------+---------------------------------+---------------
+---------------------+--------------------+-----------------------+
|       DRGDefinition|ProviderId|        ProviderName|ProviderStreetAddress|ProviderCity|ProviderState|ProviderZipCode|HospitalReferralRegionDescription|TotalDischarges
|AverageCoveredCharges|AverageTotalPayments|AverageMedicarePayments|
+--------------------+----------+--------------------+---------------------+------------+-------------+---------------+---------------------------------+---------------
+---------------------+--------------------+-----------------------+
|039 - EXTRACRANIA...|     10001|SOUTHEAST ALABAMA...| 1108 ROSS CLARK C...|      DOTHAN|           AL|          36301|                      AL - Dothan|             91
|             32963.07|             5777.24|                4763.73|
|039 - EXTRACRANIA...|     10005|MARSHALL MEDICAL ...| 2505 U S HIGHWAY ...|        BOAZ|           AL|          35957|                  AL - Birmingham|             14
|             15131.85|             5787.57|                4976.71|
|039 - EXTRACRANIA...|     10006|ELIZA COFFEE MEMO...|   205 MARENGO STREET|    FLORENCE|           AL|          35631|                  AL - Birmingham|             24
|             37560.37|             5434.95|                4453.79|
|039 - EXTRACRANIA...|     10011|   ST VINCENT'S EAST| 50 MEDICAL PARK E...|  BIRMINGHAM|           AL|          35235|                  AL - Birmingham|             25
|             13998.28|             5417.56|                4129.16|
|039 - EXTRACRANIA...|     10016|SHELBY BAPTIST ME...| 1000 FIRST STREET...|   ALABASTER|           AL|          35007|                  AL - Birmingham|             18
|             31633.27|             5658.33|                4851.44|
|039 - EXTRACRANIA...|     10023|BAPTIST MEDICAL C...| 2105 EAST SOUTH B...|  MONTGOMERY|           AL|          36116|                  AL - Montgomery|             67
|             16920.79|              6653.8|                5374.14|
|039 - EXTRACRANIA...|     10029|EAST ALABAMA MEDI...| 2000 PEPPERELL PA...|     OPELIKA|           AL|          36801|                  AL - Birmingham|             51
|             11977.13|             5834.74|                4761.41|
|039 - EXTRACRANIA...|     10033|UNIVERSITY OF ALA...| 619 SOUTH 19TH ST...|  BIRMINGHAM|           AL|          35233|                  AL - Birmingham|             32
|             35841.09|             8031.12|                 5858.5|
|039 - EXTRACRANIA...|     10039| HUNTSVILLE HOSPITAL|        101 SIVLEY RD|  HUNTSVILLE|           AL|          35801|                  AL - Huntsville|            135
|             28523.39|             6113.38|                 5228.4|
|039 - EXTRACRANIA...|     10040|GADSDEN REGIONAL ...| 1007 GOODYEAR AVENUE|     GADSDEN|           AL|          35903|                  AL - Birmingham|             34
|             75233.38|             5541.05|                4386.94|
|039 - EXTRACRANIA...|     10046|RIVERVIEW REGIONA...| 600 SOUTH THIRD S...|     GADSDEN|           AL|          35901|                  AL - Birmingham|             14
|             67327.92|             5461.57|                4493.57|
|039 - EXTRACRANIA...|     10055|    FLOWERS HOSPITAL| 4370 WEST MAIN ST...|      DOTHAN|           AL|          36305|                      AL - Dothan|             45
|             39607.28|             5356.28|                 4408.2|
|039 - EXTRACRANIA...|     10056|ST VINCENT'S BIRM...| 810 ST VINCENT'S ...|  BIRMINGHAM|           AL|          35205|                  AL - Birmingham|             43
|             22862.23|             5374.65|                4186.02|
|039 - EXTRACRANIA...|     10078|NORTHEAST ALABAMA...| 400 EAST 10TH STREET|    ANNISTON|           AL|          36207|                  AL - Birmingham|             21
|             31110.85|             5366.23|                4376.23|
|039 - EXTRACRANIA...|     10083|SOUTH BALDWIN REG...| 1613 NORTH MCKENZ...|       FOLEY|           AL|          36535|                      AL - Mobile|             15
|             25411.33|             5282.93|                4383.73|
|039 - EXTRACRANIA...|     10085|DECATUR GENERAL H...|   1201 7TH STREET SE|     DECATUR|           AL|          35609|                  AL - Huntsville|             27
```

<u>Objective – 2</u>

1) What is the average amount of AverageCoveredCharges per state

val result1 = sqlContext.sql("select ProviderState,avg(AverageCoveredCharges) as AverageCoveredCharges from hospitalTable GROUP BY ProviderState")

result1.show()

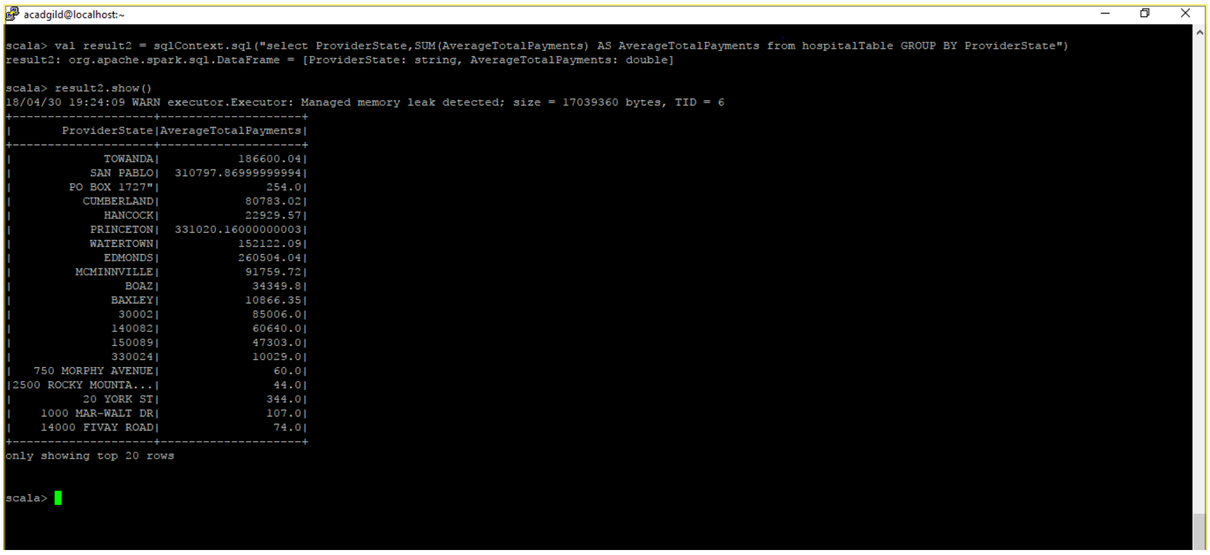


```
scala> val result1 = sqlContext.sql("select ProviderState,avg(AverageCoveredCharges) as AverageCoveredCharges from hospitalTable GROUP BY ProviderState")
result1: org.apache.spark.sql.DataFrame = [ProviderState: string, AverageCoveredCharges: double]

scala> result1.show()
18/04/30 19:20:11 WARN executor.Executor: Managed memory leak detected; size = 17039360 bytes, TID = 2
+--------------------+---------------------+
|       ProviderState|AverageCoveredCharges|
+--------------------+---------------------+
|             TOWANDA|                 17.0|
|           SAN PABLO|                 27.2|
|         PO BOX 1727"|                 null|
|          CUMBERLAND|    54.57142857142857|
|             HANCOCK|                 18.0|
|           PRINCETON|                 51.1|
|           WATERTOWN|   30.571428571428573|
|             EDMONDS|   23.571428571428573|
|          MCMINNVILLE|                 38.0|
|                BOAZ|                 37.5|
|              BAXLEY|                 14.0|
|               30002|                 null|
|              140082|                 null|
|              150089|                 null|
|              330024|                 null|
|    750 MORPHY AVENUE|                 null|
|2500 ROCKY MOUNTA...|                 null|
|          20 YORK ST|                 null|
|     1000 MAR-WALT DR|                 null|
|     14000 FIVAY ROAD|                 null|
+--------------------+---------------------+
only showing top 20 rows


scala> 
```

2) find out the AverageTotalPayments charges per state

val result2 = sqlContext.sql("select ProviderState,SUM(AverageTotalPayments) AS AverageTotalPayments from hospitalTable GROUP BY ProviderState")

result2.show()

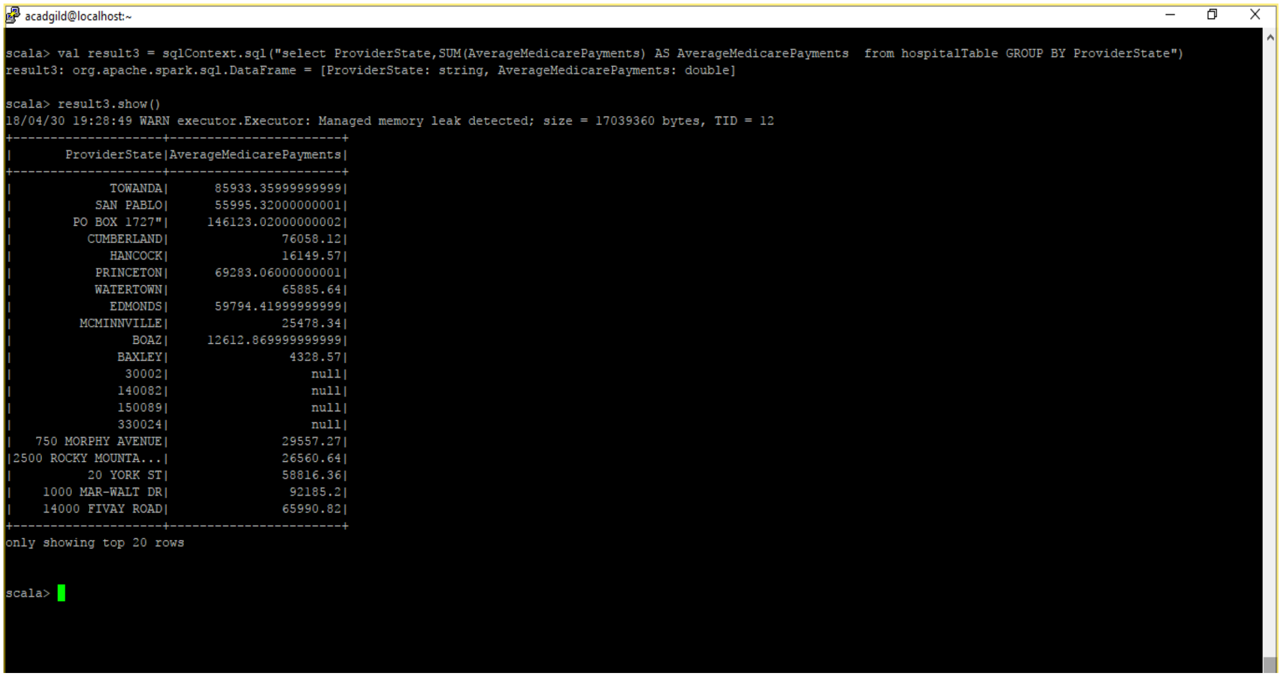


```
scala> val result2 = sqlContext.sql("select ProviderState,SUM(AverageTotalPayments) AS AverageTotalPayments from hospitalTable GROUP BY ProviderState")
result2: org.apache.spark.sql.DataFrame = [ProviderState: string, AverageTotalPayments: double]

scala> result2.show()
18/04/30 19:24:09 WARN executor.Executor: Managed memory leak detected; size = 17039360 bytes, TID = 6
+--------------------+--------------------+
|       ProviderState|AverageTotalPayments|
+--------------------+--------------------+
|             TOWANDA|           186600.04|
|           SAN PABLO|  310797.86999999994|
|         PO BOX 1727"|               254.0|
|          CUMBERLAND|            80783.02|
|             HANCOCK|            22929.57|
|           PRINCETON|  331020.16000000003|
|           WATERTOWN|           152122.09|
|             EDMONDS|           260504.04|
|          MCMINNVILLE|            91759.72|
|                BOAZ|             34349.8|
|              BAXLEY|            10866.35|
|               30002|             85006.0|
|              140082|             60640.0|
|              150089|             47303.0|
|              330024|             10029.0|
|    750 MORPHY AVENUE|                60.0|
|2500 ROCKY MOUNTA...|                44.0|
|          20 YORK ST|               344.0|
|     1000 MAR-WALT DR|               107.0|
|     14000 FIVAY ROAD|                74.0|
+--------------------+--------------------+
only showing top 20 rows

scala> 
```

3) find out the AverageMedicarePayments charges per state.

val result3 = sqlContext.sql("select ProviderState,SUM(AverageMedicarePayments) AS AverageMedicarePayments from hospitalTable GROUP BY ProviderState")

result3.show()

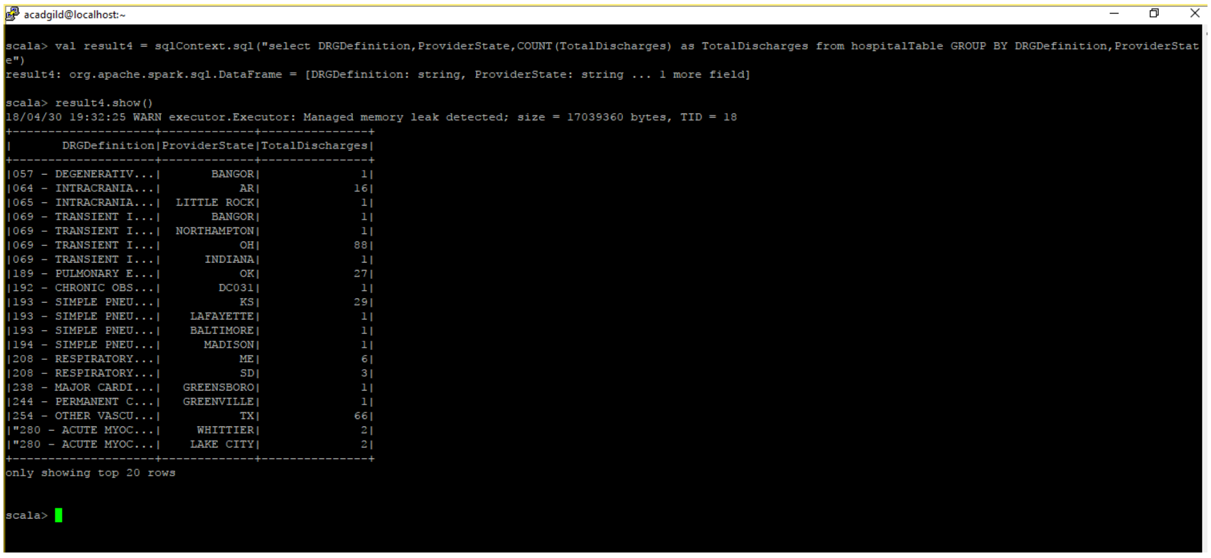


```
scala> val result3 = sqlContext.sql("select ProviderState,SUM(AverageMedicarePayments) AS AverageMedicarePayments  from hospitalTable GROUP BY ProviderState")
result3: org.apache.spark.sql.DataFrame = [ProviderState: string, AverageMedicarePayments: double]

scala> result3.show()
18/04/30 19:28:49 WARN executor.Executor: Managed memory leak detected; size = 17039360 bytes, TID = 12
+--------------------+-----------------------+
|       ProviderState|AverageMedicarePayments|
+--------------------+-----------------------+
|             TOWANDA|      85933.35999999999|
|           SAN PABLO|      55995.32000000001|
|        PO BOX 1727"|     146123.02000000002|
|          CUMBERLAND|               76058.12|
|             HANCOCK|               16149.57|
|           PRINCETON|      69283.06000000001|
|           WATERTOWN|               65885.64|
|             EDMONDS|      59794.41999999999|
|         MCMINNVILLE|               25478.34|
|                BOAZ|     12612.869999999999|
|              BAXLEY|                4328.57|
|               30002|                   null|
|              140082|                   null|
|              150089|                   null|
|              330024|                   null|
|    750 MORPHY AVENUE|               29557.27|
|2500 ROCKY MOUNTA...|               26560.64|
|          20 YORK ST|               58816.36|
|     1000 MAR-WALT DR|                92185.2|
|     14000 FIVAY ROAD|               65990.82|
+--------------------+-----------------------+
only showing top 20 rows


scala> 
```

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

val result4 = sqlContext.sql("select DRGDefinition,ProviderState,COUNT(TotalDischarges) as TotalDischarges from hospitalTable GROUP BY DRGDefinition,ProviderState")

result4.show()



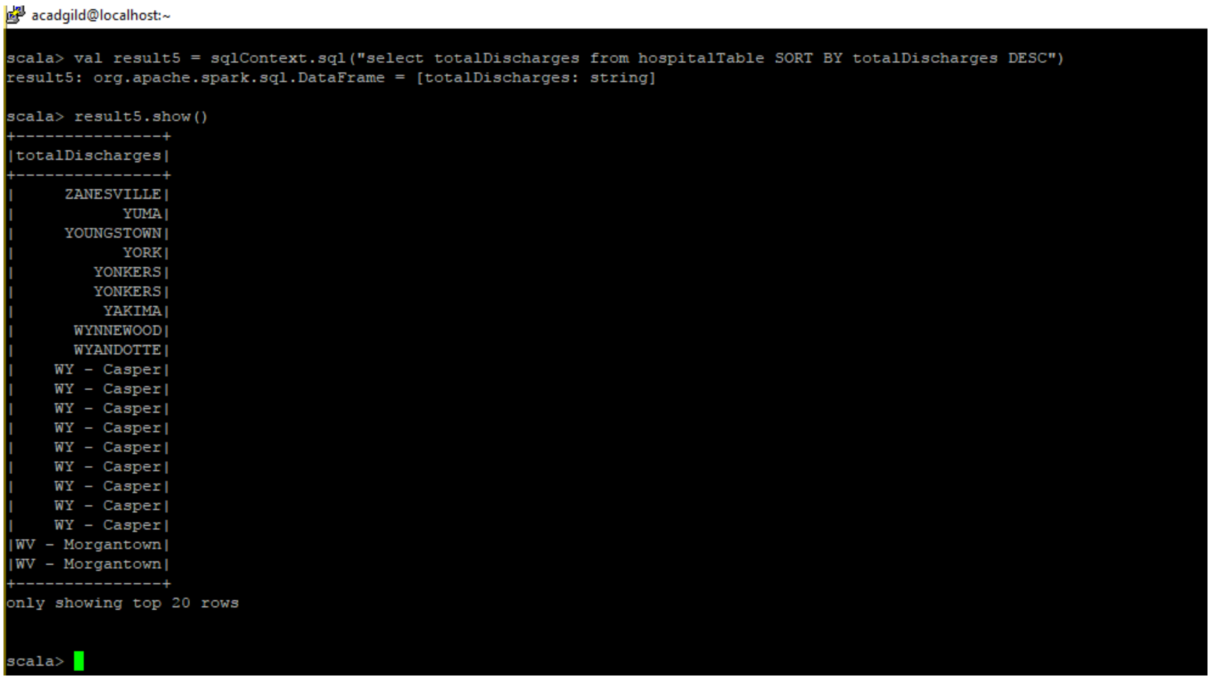
acadgild@localhost:~

```
scala> val result4 = sqlContext.sql("select DRGDefinition,ProviderState,COUNT(TotalDischarges) as TotalDischarges from hospitalTable GROUP BY DRGDefinition,ProviderStat
e")
result4: org.apache.spark.sql.DataFrame = [DRGDefinition: string, ProviderState: string ... 1 more field]

scala> result4.show()
18/04/30 19:32:25 WARN executor.Executor: Managed memory leak detected; size = 17039360 bytes, TID = 18
+--------------------+-------------+---------------+
|       DRGDefinition|ProviderState|TotalDischarges|
+--------------------+-------------+---------------+
|057 - DEGENERATIV...|       BANGOR|              1|
|064 - INTRACRANIA...|           AR|             16|
|065 - INTRACRANIA...|  LITTLE ROCK|              1|
|069 - TRANSIENT I...|       BANGOR|              1|
|069 - TRANSIENT I...|  NORTHAMPTON|              1|
|069 - TRANSIENT I...|           OH|             88|
|069 - TRANSIENT I...|      INDIANA|              1|
|189 - PULMONARY E...|           OK|             27|
|192 - CHRONIC OBS...|        DC031|              1|
|193 - SIMPLE PNEU...|           KS|             29|
|193 - SIMPLE PNEU...|    LAFAYETTE|              1|
|193 - SIMPLE PNEU...|    BALTIMORE|              1|
|194 - SIMPLE PNEU...|      MADISON|              1|
|208 - RESPIRATORY...|           ME|              6|
|208 - RESPIRATORY...|           SD|              3|
|238 - MAJOR CARDI...|   GREENSBORO|              1|
|244 - PERMANENT C...|   GREENVILLE|              1|
|254 - OTHER VASCU...|           TX|             66|
|"280 - ACUTE MYOC...|     WHITTIER|              2|
|"280 - ACUTE MYOC...|    LAKE CITY|              2|
+--------------------+-------------+---------------+
only showing top 20 rows


scala> 
```

5) Sort the output in descending order of totalDischarges

val result5 = sqlContext.sql("select totalDischarges from hospitalTable SORT BY totalDischarges DESC")

result5.show()



acadgild@localhost:~

```
scala> val result5 = sqlContext.sql("select totalDischarges from hospitalTable SORT BY totalDischarges DESC")
result5: org.apache.spark.sql.DataFrame = [totalDischarges: string]

scala> result5.show()
+---------------+
|totalDischarges|
+---------------+
|     ZANESVILLE|
|           YUMA|
|     YOUNGSTOWN|
|           YORK|
|        YONKERS|
|        YONKERS|
|         YAKIMA|
|      WYNNEWOOD|
|      WYANDOTTE|
|    WY - Casper|
|    WY - Casper|
|    WY - Casper|
|    WY - Casper|
|    WY - Casper|
|    WY - Casper|
|    WY - Casper|
|    WY - Casper|
|    WY - Casper|
|WV - Morgantown|
|WV - Morgantown|
+---------------+
only showing top 20 rows


scala> 
```