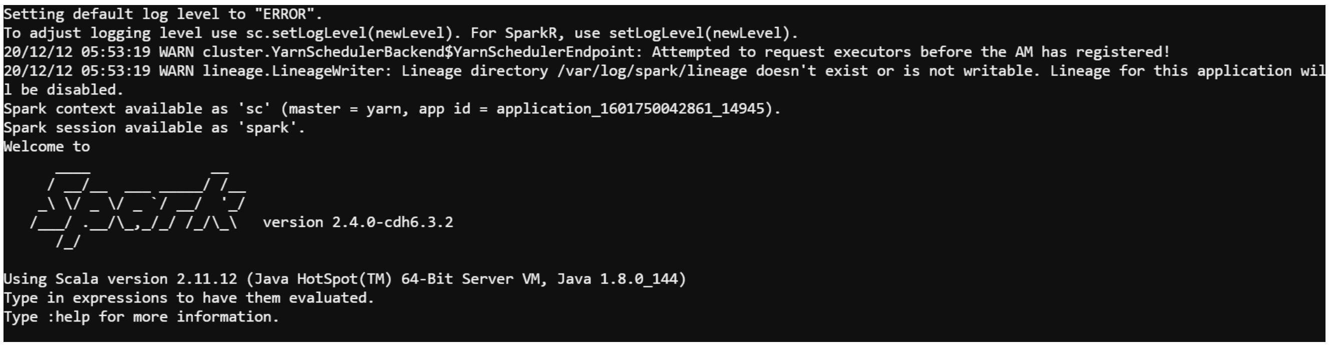
Market Analysis in Banking Domain

Analysis tasks to be done-:

The data size is huge and the marketing team has asked you to perform the below analysis-



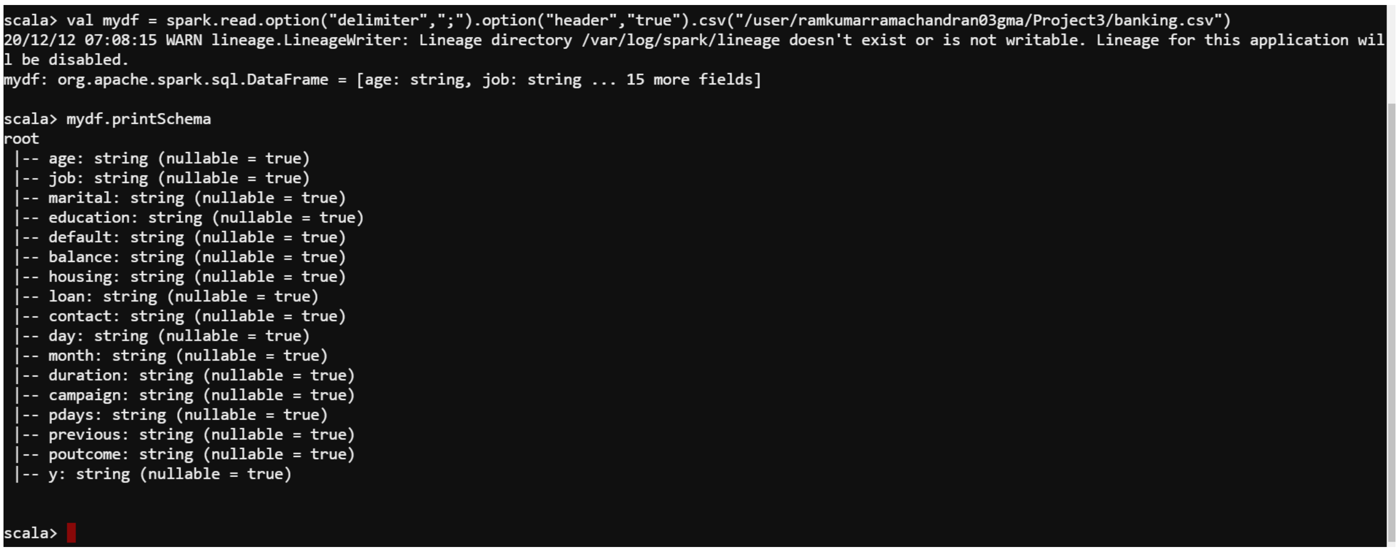
```
Setting default log level to "ERROR".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
20/12/12 05:53:19 WARN cluster.YarnSchedulerBackend$YarnSchedulerEndpoint: Attempted to request executors before the AM has registered!
20/12/12 05:53:19 WARN lineage.LineageWriter: Lineage directory /var/log/spark/lineage doesn't exist or is not writable. Lineage for this application wil
l be disabled.
Spark context available as 'sc' (master = yarn, app id = application_1601750042861_14945).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 2.4.0-cdh6.3.2
      /_/

Using Scala version 2.11.12 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_144)
Type in expressions to have them evaluated.
Type :help for more information.
```

Question 1: Load data and create a Spark data frame

Uploaded the datasheet into HDFS, to create a dataframe as "mydf" in spark

val mydf = spark.read.option("delimiter",";").option("header","true").csv("/user/ramkumarramachandran03gma/Project3/banking.csv")

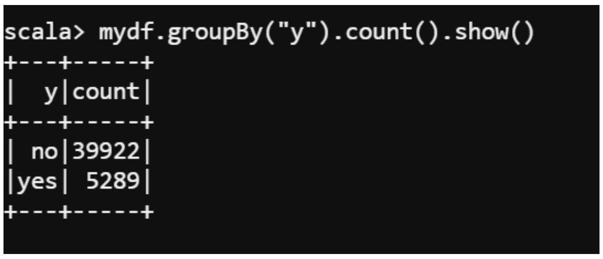


```
scala> val mydf = spark.read.option("delimiter",";").option("header","true").csv("/user/ramkumarramachandran03gma/Project3/banking.csv")
20/12/12 07:08:15 WARN lineage.LineageWriter: Lineage directory /var/log/spark/lineage doesn't exist or is not writable. Lineage for this application wil
l be disabled.
mydf: org.apache.spark.sql.DataFrame = [age: string, job: string ... 15 more fields]

scala> mydf.printSchema
root
 |-- age: string (nullable = true)
 |-- job: string (nullable = true)
 |-- marital: string (nullable = true)
 |-- education: string (nullable = true)
 |-- default: string (nullable = true)
 |-- balance: string (nullable = true)
 |-- housing: string (nullable = true)
 |-- loan: string (nullable = true)
 |-- contact: string (nullable = true)
 |-- day: string (nullable = true)
 |-- month: string (nullable = true)
 |-- duration: string (nullable = true)
 |-- campaign: string (nullable = true)
 |-- pdays: string (nullable = true)
 |-- previous: string (nullable = true)
 |-- poutcome: string (nullable = true)
 |-- y: string (nullable = true)


scala>
```

Question 2: Give marketing success rate (No. of people subscribed / total no. of entries) and Give marketing failure rate



```
scala> mydf.groupBy("y").count().show()
+---+-----+
|  y|count|
+---+-----+
| no|39922|
|yes| 5289|
+---+-----+
```

scala> val no_of_subscribed = mydf.select("y").filter(col("y")==="yes").count.toDouble
no_of_subscribed: Double = 5289.0

scala> val no_of_failured = mydf.select("y").filter(col("y")==="no").count.toDouble
no_of_failured: Double = 39922.0

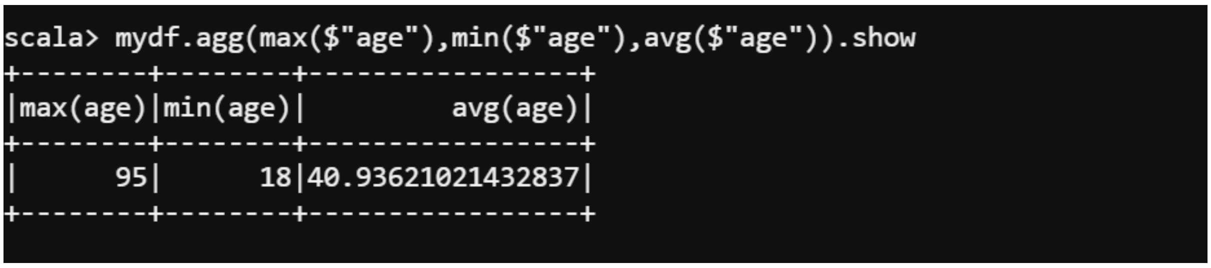
scala> val totalentry = mydf.count.toDouble
totalentry: Double = 45211.0

scala> val success_rate = no_of_subscribed / totalentry*100
success_rate: Double = 11.698480458295547

scala> val failure_rate = no_of_failured / totalentry*100
failure_rate: Double = 88.30151954170445

Question 3: Give the maximum, mean, and minimum age of the average targeted customer

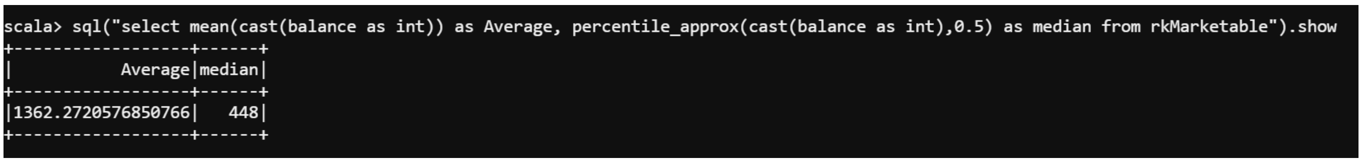
scala>mydf.agg(max($"age"),min($"age"),avg($"age")).show



```
scala> mydf.agg(max($"age"),min($"age"),avg($"age")).show
+--------+--------+-----------------+
|max(age)|min(age)|         avg(age)|
+--------+--------+-----------------+
|      95|      18|40.93621021432837|
+--------+--------+-----------------+
```

Question 4: Check the quality of customers by checking average balance, median balance of customers

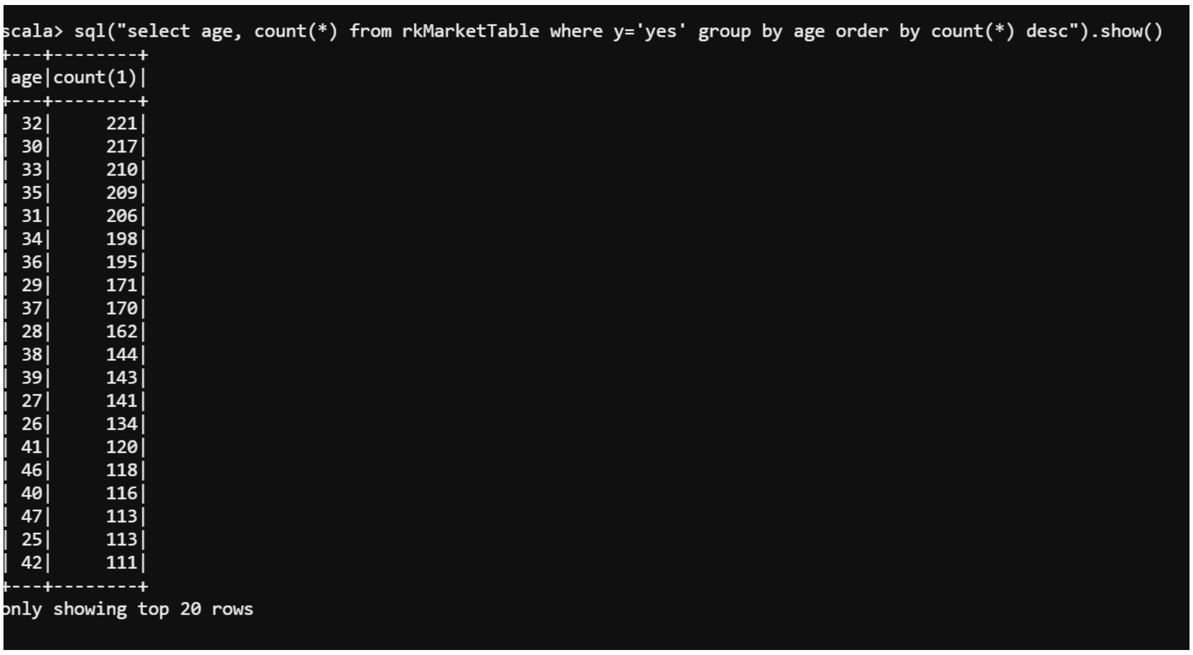
scala> mydf.registerTempTable("rkMarketable")
scala>sql("select mean(cast(balance as int)) as Average, percentile_approx(cast(balance as int),0.5) as median from rkMarketable").show



```
scala> sql("select mean(cast(balance as int)) as Average, percentile_approx(cast(balance as int),0.5) as median from rkMarketable").show
+-----------------+------+
|          Average|median|
+-----------------+------+
|1362.2720576850766|   448|
+-----------------+------+
```

Question 5: Check if age matters in marketing subscription for deposit

scala>sql("select age, count(*) from rkMarketTable where y='yes' group by age order by count(*) desc").show()

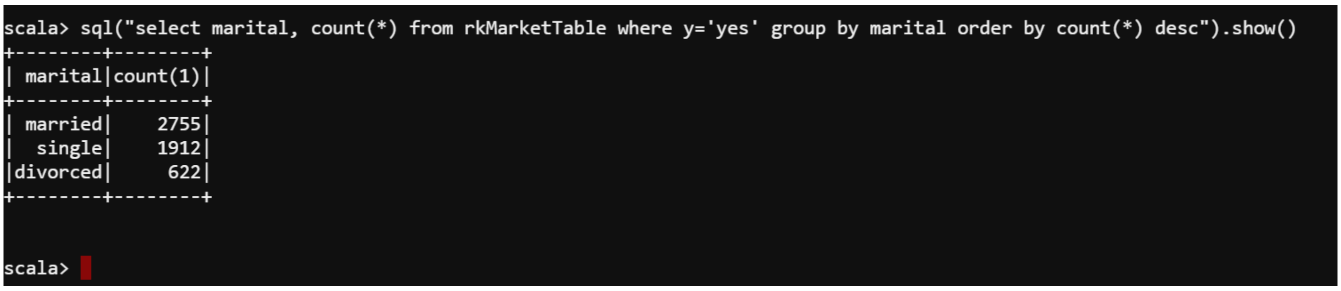


```
scala> sql("select age, count(*) from rkMarketTable where y='yes' group by age order by count(*) desc").show()
+---+--------+
|age|count(1)|
+---+--------+
| 32|     221|
| 30|     217|
| 33|     210|
| 35|     209|
| 31|     206|
| 34|     198|
| 36|     195|
| 29|     171|
| 37|     170|
| 28|     162|
| 38|     144|
| 39|     143|
| 27|     141|
| 26|     134|
| 41|     120|
| 46|     118|
| 40|     116|
| 47|     113|
| 25|     113|
| 42|     111|
+---+--------+
only showing top 20 rows
```

Yes, age matters to the Subscription to deposit

Question 6: Check if marital status mattered for a subscription to deposit

scala>sql("select marital, count(*) from rkMarketTable where y='yes' group by marital order by count(*) desc").show()



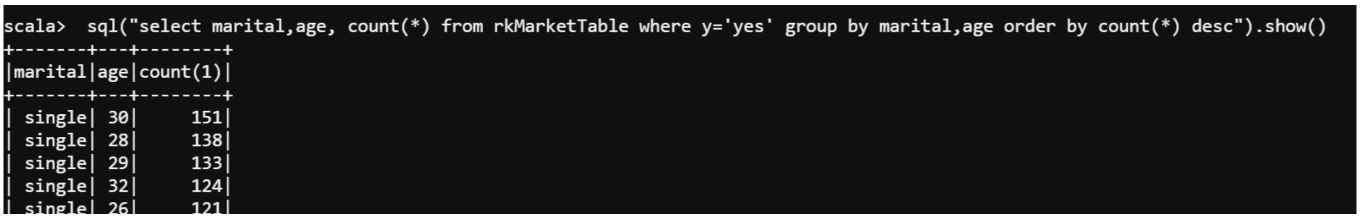
```
scala> sql("select marital, count(*) from rkMarketTable where y='yes' group by marital order by count(*) desc").show()
+--------+--------+
| marital|count(1)|
+--------+--------+
| married|    2755|
|  single|    1912|
|divorced|     622|
+--------+--------+


scala>
```

Subscription to deposit – based on the Martial Status

<u>Question 7: Check if age and marital status together mattered for a subscription to deposit scheme</u>

scala> sql("select marital,age, count(*) from rkMarketTable where y='yes' group by marital,age order by count(*) desc").show()

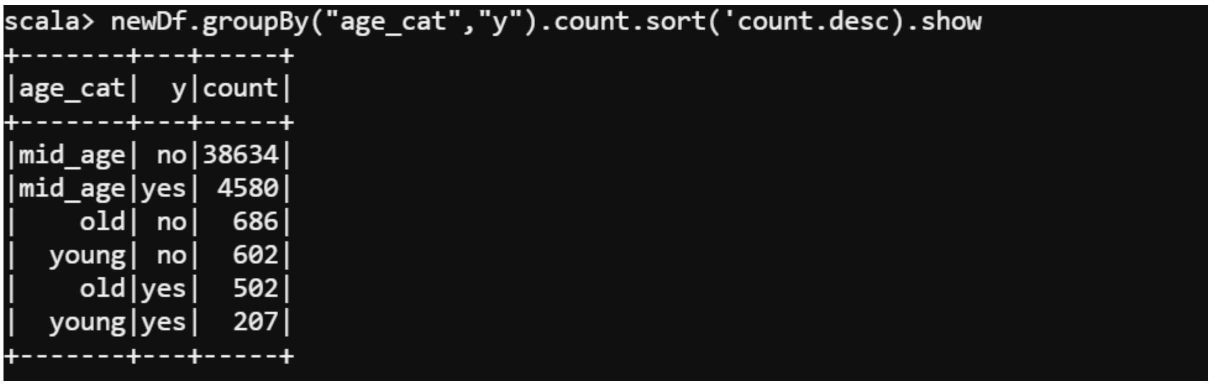


```
scala>  sql("select marital,age, count(*) from rkMarketTable where y='yes' group by marital,age order by count(*) desc").show()
+-------+---+--------+
|marital|age|count(1)|
+-------+---+--------+
| single| 30|     151|
| single| 28|     138|
| single| 29|     133|
| single| 32|     124|
| single| 26|     121|
```

<u>Question 8: Do feature engineering for the bank and find the right age effect on the campaign.</u>

Have created new dataframe (newDf) with additional field as "age_cat" by grouping the age

scala>val newDf =mydf.withColumn("age_cat",when ($"age" < 25,"young").otherwise(when($"age" > 60,"old") .otherwise("mid_age")))
scala> newDf.groupBy("age_cat","y").count.sort('count.desc).show



```
scala> newDf.groupBy("age_cat","y").count.sort('count.desc).show
+-------+---+-----+
|age_cat|  y|count|
+-------+---+-----+
|mid_age| no|38634|
|mid_age|yes| 4580|
|    old| no|  686|
|  young| no|  602|
|    old|yes|  502|
|  young|yes|  207|
+-------+---+-----+
```

Subscription to deposit – Mid Age (age >25 & age < 60) has impact