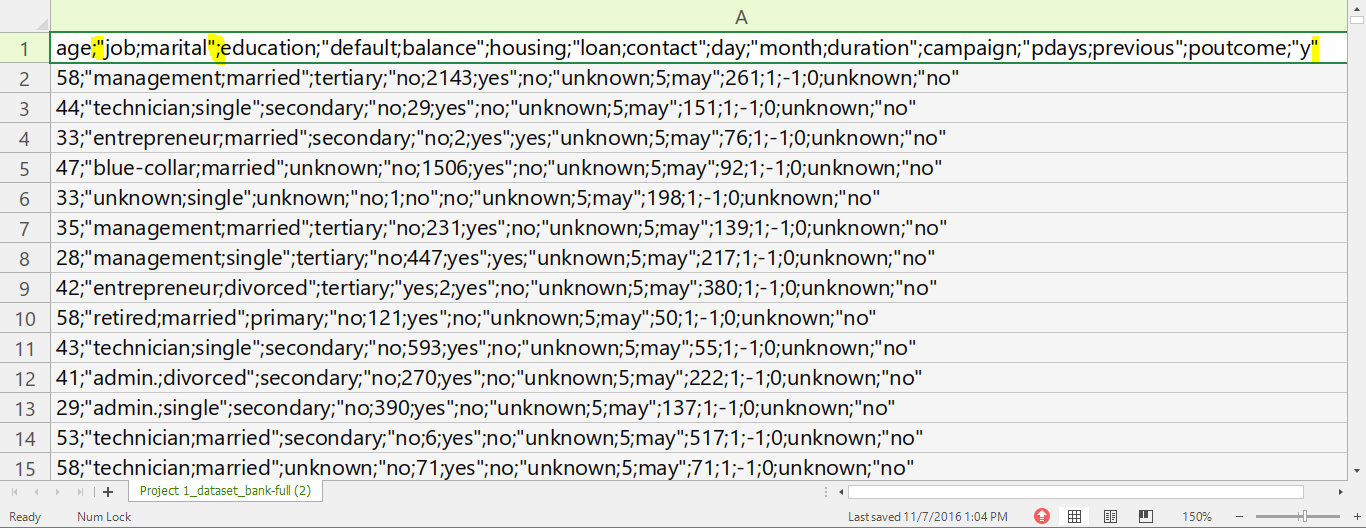
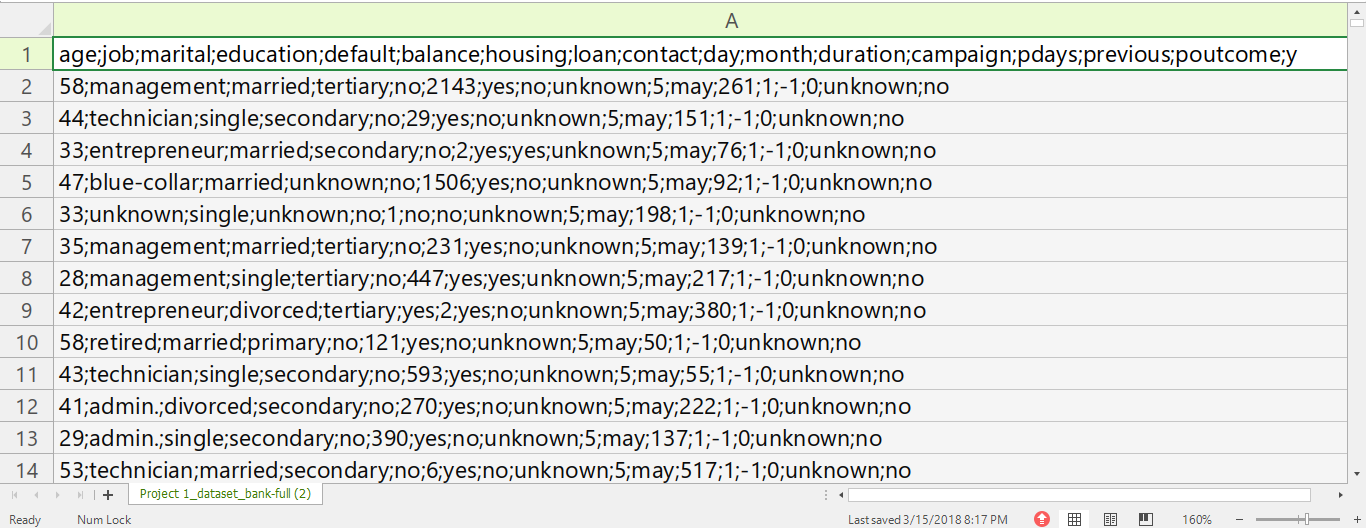
**Project 1\_Spark\_Marketing\_Analysis**

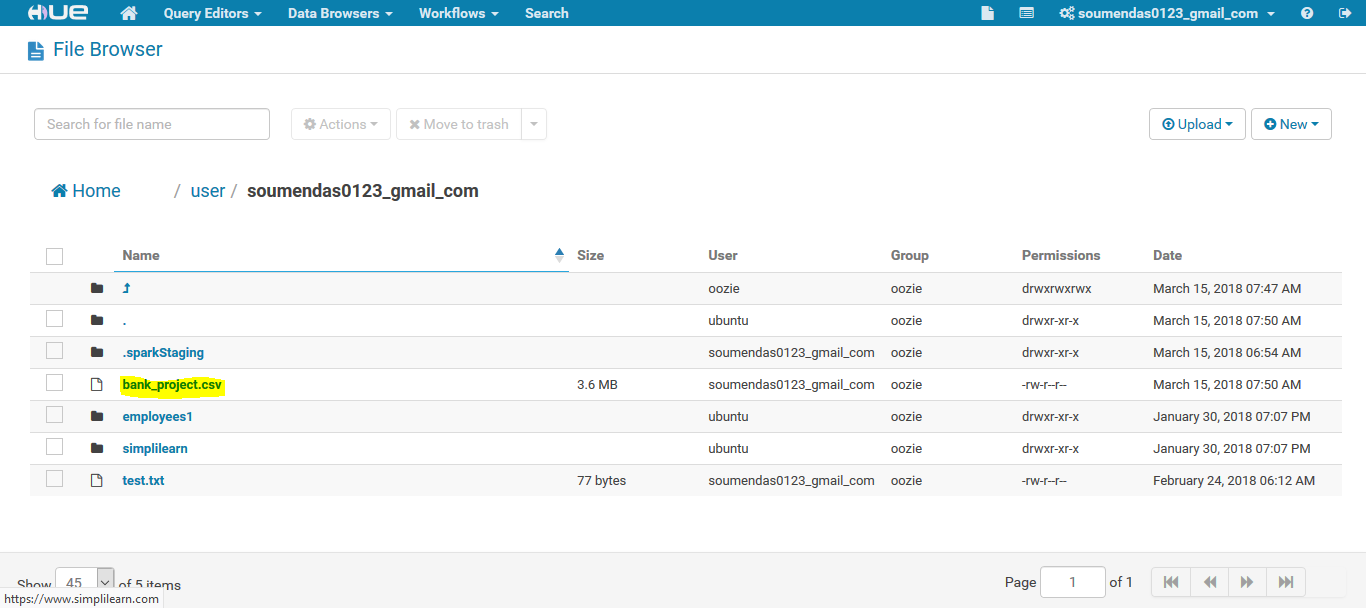
As we can see double qoutes are in random order in this csv file, we will remove them manually.



So clean csv file will look like this.

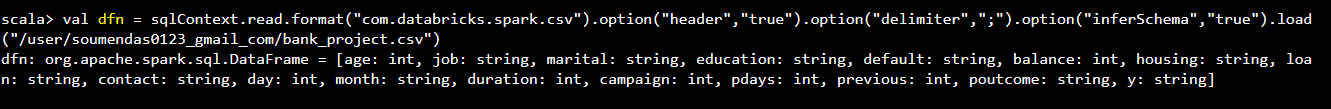


Now we will upload it to HDFS using HUE & rename it to bank\_project.csv .



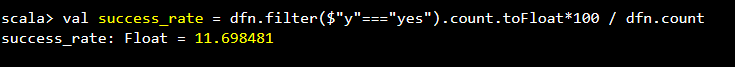
1. Load data and create Spark data frame :

val dfn = sqlContext.read.format("com.databricks.spark.csv").option("header","true").option("delimiter",";").option("inferSchema","true").load("/user/soumendas0123\_gmail\_com/[bank\_project.csv](http://cloudera.mettl.com:8889/filebrowser/view=/user/soumendas0123_gmail_com/bank_project.csv)")



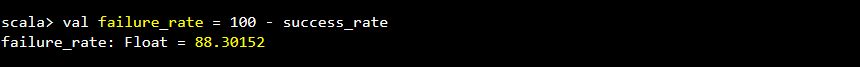
2. Give marketing success rate. (No. of people subscribed / total no. of entries) :

val success\_rate = dfn.filter($"y"==="yes").count.toFloat\*100 / dfn.count



2a Give marketing failure rate :

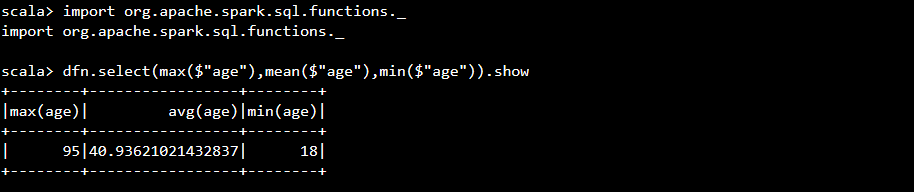
val failure\_rate = 100 - success\_rate



3. Maximum, Mean, and Minimum age of average targeted customer :

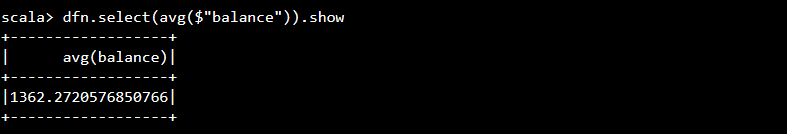
import org.apache.spark.sql.functions.\_

dfn.select(max($"age"),mean($"age"),min($"age")).show



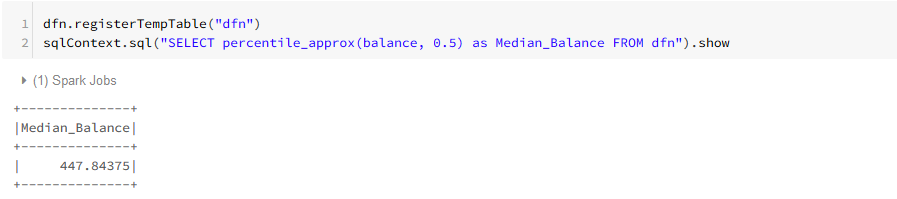
4. Check quality of customers by checking average balance, median balance of customers :

dfn.select(avg($"balance")).show



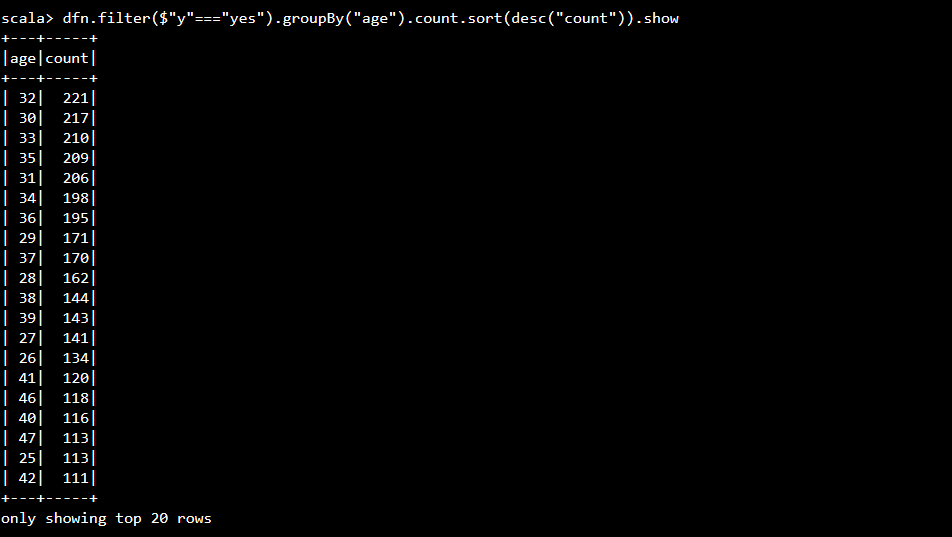
dfn.registerTempTable("dfn")

sqlContext.sql("SELECT percentile\_approx(balance, 0.5) as Median\_Balance FROM dfn").show



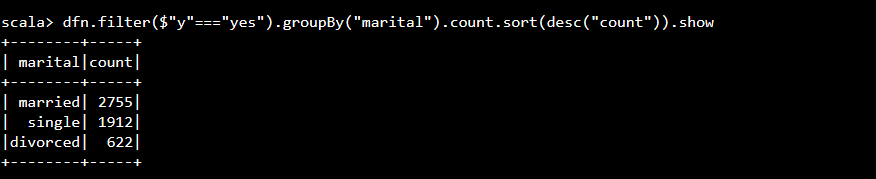
5. Check if age matters in marketing subscription for deposit :

dfn.filter($"y"==="yes").groupBy("age").count.sort(desc("count")).show

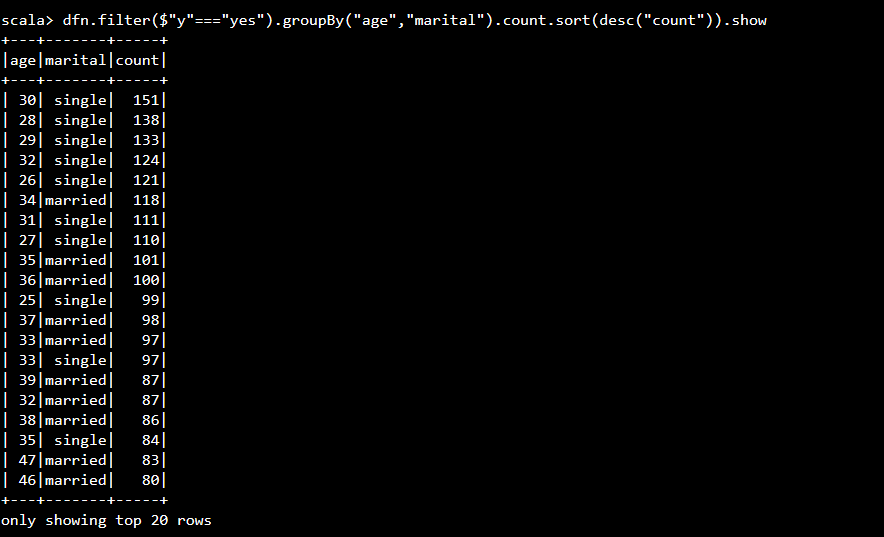


6. Check if marital status mattered for subscription to deposit :

dfn.filter($"y"==="yes").groupBy("marital").count.sort(desc("count")).show



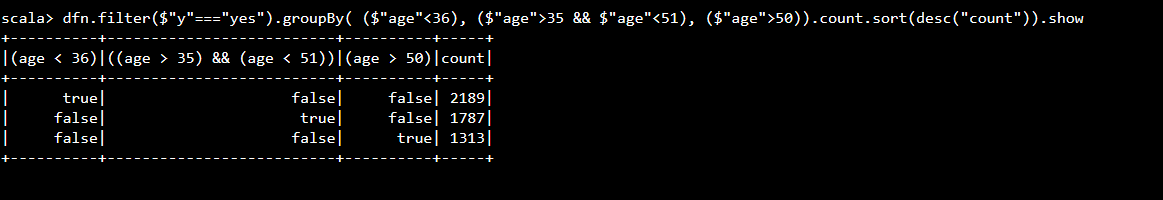
7. Check if age and marital status together mattered for subscription to deposit scheme

dfn.filter($"y"==="yes").groupBy("age","marital").count.sort(desc("count")).show

8. Do feature engineering for column—age and find right age effect on campaign :

dfn.filter($"y"==="yes").groupBy( ($"age"<36), ($"age">35 && $"age"<51), ($"age">50)).count.sort(de

sc("count")).show



So we can see subscription for deposit for age group is as below:

Note: Min Age = 17, Max Age = 51

A. 17<age <36 -- 2189

B. 35<age<51 -- 1787

C. 50<age<96 -- 1313