ASSIGNMENT 18.3

Input Datasets:

We have an airline data with us:

user details:

user_id, name, age

holidays:

user_id, src, dest, travel_mode, distance, year_of_travel

transport:

travel_mode, cost_per_unit

Here are the datasets which we will be using for this assignment in all problems. It has been kept in local file system:

S18_Dataset_Holidays.txt

```
1,CHN,IND,airplane,200,1990
2,IND,CHN,airplane,200,1991
3,IND,CHN,airplane,200,1992
4,RUS,IND,airplane,200,1990
5,CHN,RUS,airplane,200,1992
6,AUS,PAK,airplane,200,1991
7,RUS,AUS,airplane,200,1990
8,IND,RUS,airplane,200,1991
9,CHN,RUS,airplane,200,1992
10,AUS,CHN,airplane,200,1993
1,AUS,CHN,airplane,200,1993
2,CHN,IND,airplane,200,1993
3,CHN,IND,airplane,200,1993
4,IND,AUS,airplane,200,1991
5,AUS,IND,airplane,200,1992
6,RUS,CHN,airplane,200,1993
7,CHN,RUS,airplane,200,1990
8,AUS,CHN,airplane,200,1990
9,IND,AUS,airplane,200,1991
10,RUS,CHN,airplane,200,1992
1,PAK,IND,airplane,200,1993
2,IND,RUS,airplane,200,1991
3,CHN,PAK,airplane,200,1991
4,CHN,PAK,airplane,200,1990
5,IND,PAK,airplane,200,1991
6,PAK,RUS,airplane,200,1991
7,CHN,IND,airplane,200,1990
8,RUS,IND,airplane,200,1992
9,RUS,IND,airplane,200,1992
10,CHN,AUS,airplane,200,1990
1,PAK,AUS,airplane,200,1993
5,CHN,PAK,airplane,200,1994
```

S18_Dataset_Holidays.txt | S18_Dataset_Transport.txt

```
airplane,170
car,140
train,120
ship,200
```

S18_Dataset_Holidays.txt | S18_Dataset_Transport.txt | S18_Dataset_User_details.txt

```
1,mark,15
2,john,16
3,luke,17
4,lisa,27
5,mark,25
6,peter,22
7,james,21
8,andrew,55
9,thomas,46
10,annie,44
```

Problem Statement:

1) Considering age groups of < 20, 20-35, 35 >, which age group spends the most amount of money travelling.

2) What is the amount spent by each age-group, every year in travelling?

Solution:

1. Here is the Spark code snippet to find the age group that spends most amount of money travelling:

```
// import required Spark packages

import org.apache.spark.sql.SparkSession
import org.apache.spark.sql.types.{IntegerType, StringType}


object Assignment18_2 {
  def main(args: Array[String]): Unit = {
```

```
val spark = SparkSession                    // create a SparkSession object that can be used to
  .builder()                                // create various contexts of Spark such as sqlContext
  .config("spark.sql.warehouse.dir", "file:///c:/tmp/spark-warehouse")
  .master("local[*]")
  .getOrCreate()
val sqlContext = spark.sqlContext                          // initialize sqlContext
val input_df1 = sqlContext.read.csv ("E:\\Acadgild\\Session 18\\S18_Datasets \\S18_Dataset
_Transport.txt")                                           // load input data file – transport.txt
val transportDF = input_df1.select(                        // define schema for input data loaded
  input_df1("_c0").cast(StringType).as("travel_mode"),     //assign column names to the data frame
  input_df1("_c1").cast(IntegerType).as("cost_per_unit"))
transportDF.createOrReplaceTempView("transport")           // create a temporary view - transport

val input_df2 = sqlContext.read.csv("E:\\Acadgild\\Session 18\\S18_Datasets\\S18_Dataset_
Holidays.txt")                                             // load input data file – holidays.txt
val holidaysDF = input_df2.select(                         // define schema for input data loaded
  input_df2("_c0").cast(IntegerType).as("user_id"),        //assign column names to the data frame
  input_df2("_c1").cast(StringType).as("src"),
  input_df2("_c2").cast(StringType).as("dest"),
  input_df2("_c3").cast(StringType).as("travel_mode"),
  input_df2("_c4").cast(IntegerType).as("distance"),
  input_df2("_c5").cast(IntegerType).as("year_of_travel"))
holidaysDF.createOrReplaceTempView("holidays")             // create a temporary view - holidays

val input_df3 = sqlContext.read.csv("E:\\Acadgild\\Session 18\\S18_Datasets\\S18_Dataset_
User_details.txt")                                         // load input data file – user_details.txt
val usersDF = input_df3.select(                            // define schema for input data loaded
  input_df3("_c0").cast(IntegerType).as("user_id"),        //assign column names to the data frame
  input_df3("_c1").cast(StringType).as("name"),
  input_df3("_c2").cast(IntegerType).as("age"))
usersDF.createOrReplaceTempView("users")                   // create a temporary view – users
```

```
// SQL query to find the age group that spends most amount of money travelling
sqlContext.sql(""SELECT z.age, SUM(x.cost_per_unit) as total_amount " +
    "FROM transport x, holidays y ,users z " +
    "WHERE x.travel_mode = y.travel_mode AND y.user_id = z.user_id " +
    "GROUP BY z.age " +
    "ORDER by z.age").show()                                    // print the result
 }
}
```

Assignment18_1.scala | Assignment18_2.scala | Assignment18_3.scala

```scala
object Assignment18_3 {
  def main(args: Array[String]): Unit = {
    val spark = SparkSession
      .builder()
      .config("spark.sql.warehouse.dir", "file:///c:/tmp/spark-warehouse")
      .master("local[*]")
      .getOrCreate()
    val sqlContext = spark.sqlContext
    val input_df1 = sqlContext.read.csv("E:\\Acadgild\\Session 18\\S18_Datasets\\S18_Dataset_Transport.txt")
    val transportDF = input_df1.select(
      input_df1("_c0").cast(StringType).as("travel_mode"),
      input_df1("_c1").cast(IntegerType).as("cost_per_unit"))
    transportDF.createOrReplaceTempView("transport")
    val input_df2 = sqlContext.read.csv("E:\\Acadgild\\Session 18\\S18_Datasets\\S18_Dataset_Holidays.txt")
    val holidaysDF = input_df2.select(
      input_df2("_c0").cast(IntegerType).as("user_id"),
      input_df2("_c1").cast(StringType).as("src"),
      input_df2("_c2").cast(StringType).as("dest"),
      input_df2("_c3").cast(StringType).as("travel_mode"),
      input_df2("_c4").cast(IntegerType).as("distance"),
      input_df2("_c5").cast(IntegerType).as("year_of_travel"))
    holidaysDF.createOrReplaceTempView("holidays")
    val input_df3 = sqlContext.read.csv("E:\\Acadgild\\Session 18\\S18_Datasets\\S18_Dataset_User_details.txt")
    val usersDF = input_df3.select(
      input_df3("_c0").cast(IntegerType).as("user_id"),
      input_df3("_c1").cast(StringType).as("name"),
      input_df3("_c2").cast(IntegerType).as("age"))
    usersDF.createOrReplaceTempView("users")
    sqlContext.sql("SELECT z.age, SUM(x.cost_per_unit) as total_amount " +
      "FROM transport x, holidays y ,users z " +
      "WHERE x.travel_mode = y.travel_mode AND y.user_id = z.user_id " +
      "GROUP BY z.age " +
      "ORDER by z.age").show()
  }
```

Output:

```
+---+------------+
|age|total_amount|
+---+------------+
| 15|         680|
| 16|         510|
| 17|         510|
| 21|         510|
| 22|         510|
| 25|         680|
| 27|         510|
| 44|         510|
| 46|         510|
| 55|         510|
+---+------------+
```

The total amount spent by the age group below 15 is **1700,** the age group 20 – 35 spends **2210** and the age group above 35 spends **1530.** So we can deduce that **the age group 20 – 35 spends the most on travelling.**