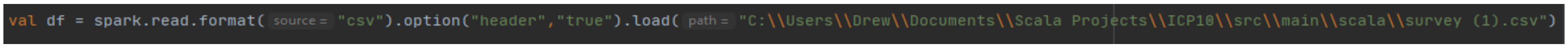
lesson-10
Lesson-10 ICP

Name: Andrew Poitras
Email: ap3h7@mail.umkc.edu

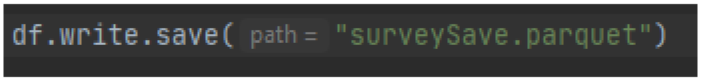
Description:

1-1. Import the dataset and create data frames directly on import

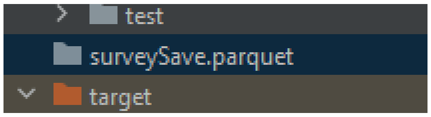


```
val df = spark.read.format( source = "csv").option("header","true").load( path = "C:\\Users\\Drew\\Documents\\Scala Projects\\ICP10\\src\\main\\scala\\survey (1).csv")
```

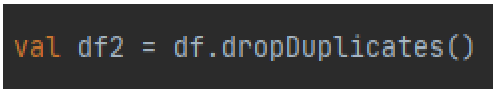
1-2. Save data to file



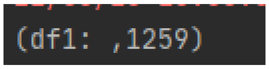
```
df.write.save( path = "surveySave.parquet")
```



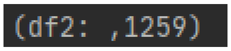
1-3. Check for duplicate records in the dataset



```
val df2 = df.dropDuplicates()
```



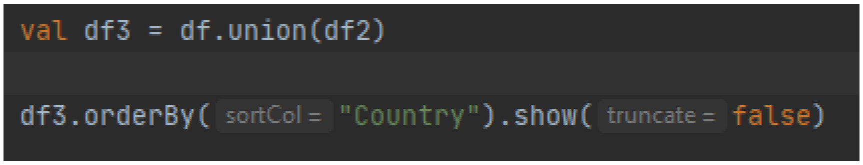
```
(df1: ,1259)
```



```
(df2: ,1259)
```

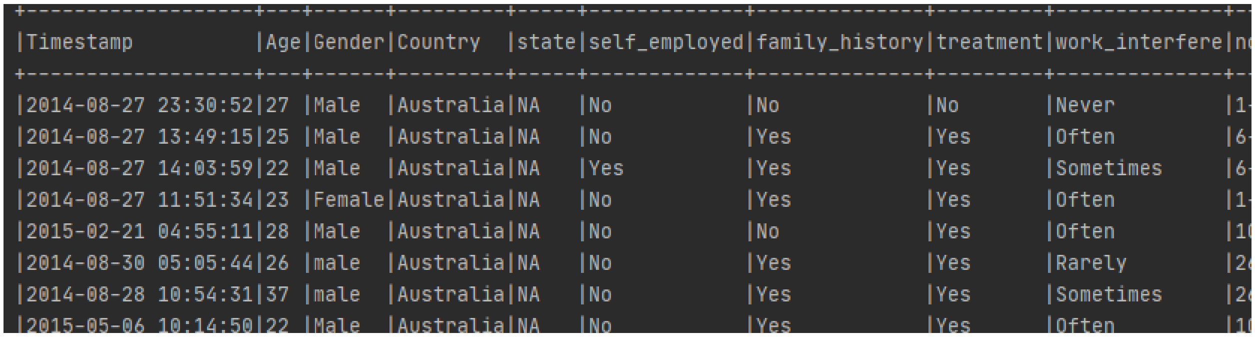
No duplicates

1-4. Apply union operation on the dataset and order the output by country name alphabetically



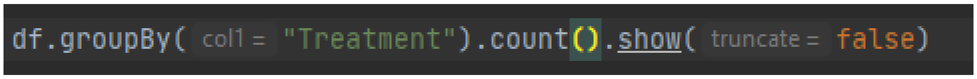
```
val df3 = df.union(df2)

df3.orderBy( sortCol = "Country").show( truncate = false)
```

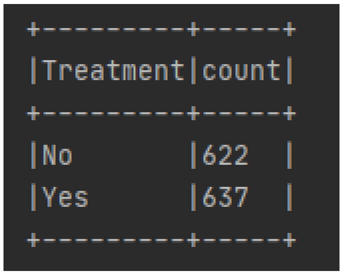


```
+-------------------+---+------+---------+-----+-------------+--------------+---------+--------------+-
|Timestamp          |Age|Gender|Country  |state|self_employed|family_history|treatment|work_interfere|n
+-------------------+---+------+---------+-----+-------------+--------------+---------+--------------+-
|2014-08-27 23:30:52|27 |Male  |Australia|NA   |No           |No            |No       |Never         |1
|2014-08-27 13:49:15|25 |Male  |Australia|NA   |No           |Yes           |Yes      |Often         |6
|2014-08-27 14:03:59|22 |Male  |Australia|NA   |Yes          |Yes           |Yes      |Sometimes     |6
|2014-08-27 11:51:34|23 |Female|Australia|NA   |No           |Yes           |Yes      |Often         |1
|2015-02-21 04:55:11|28 |Male  |Australia|NA   |No           |No            |Yes      |Often         |1
|2014-08-30 05:05:44|26 |male  |Australia|NA   |No           |Yes           |Yes      |Rarely        |2
|2014-08-28 10:54:31|37 |male  |Australia|NA   |No           |Yes           |Yes      |Sometimes     |2
|2015-05-06 10:14:50|22 |Male  |Australia|NA   |No           |Yes           |Yes      |Often         |1
```

1-5. Use groupby query based on treatment

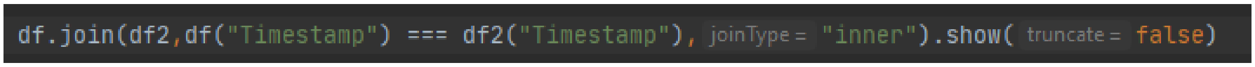


```
df.groupBy( col1 = "Treatment").count().show( truncate = false)
```

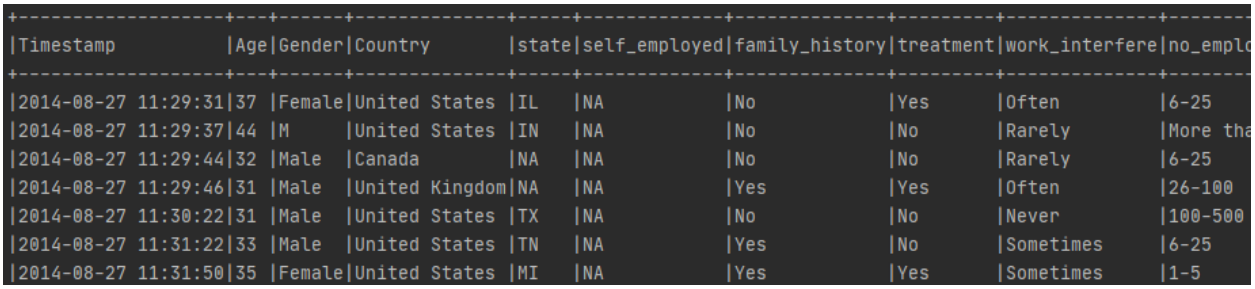


```
+---------+-----+
|Treatment|count|
+---------+-----+
|No       |622  |
|Yes      |637  |
+---------+-----+
```

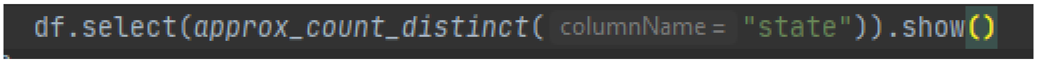
2-1. Apply the basic queries related to Joins and aggregate functions (at least 2)



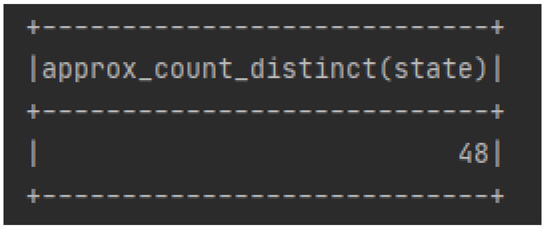
```
df.join(df2,df("Timestamp") === df2("Timestamp"), joinType = "inner").show( truncate = false)
```



```
+-------------------+---+------+--------------+-----+-------------+--------------+---------+--------------+-------
|Timestamp          |Age|Gender|Country       |state|self_employed|family_history|treatment|work_interfere|no_emplo
+-------------------+---+------+--------------+-----+-------------+--------------+---------+--------------+-------
|2014-08-27 11:29:31|37 |Female|United States |IL   |NA           |No            |Yes      |Often         |6-25
|2014-08-27 11:29:37|44 |M     |United States |IN   |NA           |No            |No       |Rarely        |More tha
|2014-08-27 11:29:44|32 |Male  |Canada        |NA   |NA           |No            |No       |Rarely        |6-25
|2014-08-27 11:29:46|31 |Male  |United Kingdom|NA   |NA           |Yes           |Yes      |Often         |26-100
|2014-08-27 11:30:22|31 |Male  |United States |TX   |NA           |No            |No       |Never         |100-500
|2014-08-27 11:31:22|33 |Male  |United States |TN   |NA           |Yes           |No       |Sometimes     |6-25
|2014-08-27 11:31:50|35 |Female|United States |MI   |NA           |Yes           |Yes      |Sometimes     |1-5
```

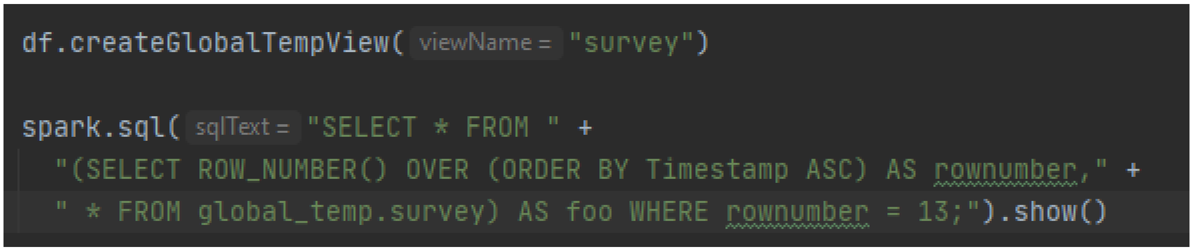


```
df.select(approx_count_distinct( columnName = "state")).show()
```



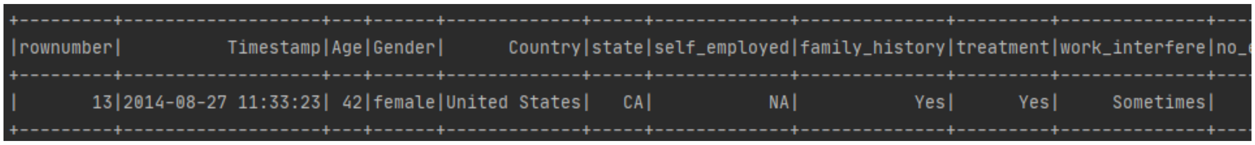
```
+----------------------------+
|approx_count_distinct(state)|
+----------------------------+
|                          48|
+----------------------------+
```

2-2. Write a query to fetch 13th Row in the dataset.



```
df.createGlobalTempView( viewName = "survey")

spark.sql( sqlText = "SELECT * FROM " +
  "(SELECT ROW_NUMBER() OVER (ORDER BY Timestamp ASC) AS rownumber," +
  " * FROM global_temp.survey) AS foo WHERE rownumber = 13;").show()
```



```
+---------+-------------------+---+------+-------------+-----+-------------+--------------+---------+--------------+---
|rownumber|          Timestamp|Age|Gender|      Country|state|self_employed|family_history|treatment|work_interfere|no_
+---------+-------------------+---+------+-------------+-----+-------------+--------------+---------+--------------+---
|       13|2014-08-27 11:33:23| 42|female|United States|   CA|           NA|           Yes|      Yes|     Sometimes|
+---------+-------------------+---+------+-------------+-----+-------------+--------------+---------+--------------+---
```