

Simplilearn Project 3 Market Analysis in Banking Domain

```
scala> val bank_people_data = spark.read.option("multiline","true").json("/user/harrshavardhanangalakudur/bank_edited.json");
21/09/13 01:34:47 WARN lineage.LineageWriter: Lineage directory /var/log/spark/lineage doesn't exist or is not writable. Lineage for this application will be disabled.
bank_people_data: org.apache.spark.sql.DataFrame = [age: bigint, balance: bigint ... 15 more fields]

scala> bank_people_data.show()
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|age|balance|campaign|contact|day|default|duration|education|housing|job|loan|marital|month|pdays|poutcome|previous|y|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|58|2143|1|unknown|5|no|261|tertiary|yes|management|no|married|may|-1|unknown|0|no|
|44|29|1|unknown|5|no|151|secondary|yes|technician|no|single|may|-1|unknown|0|no|
|33|2|1|unknown|5|no|76|secondary|yes|entrepreneur|yes|married|may|-1|unknown|0|no|
|47|1506|1|unknown|5|no|92|unknown|yes|blue-collar|no|married|may|-1|unknown|0|no|
|33|1|1|unknown|5|no|198|unknown|no|unknown|no|single|may|-1|unknown|0|no|
|35|231|1|unknown|5|no|139|tertiary|yes|management|no|married|may|-1|unknown|0|no|
|28|447|1|unknown|5|no|217|tertiary|yes|management|yes|single|may|-1|unknown|0|no|
|42|2|1|unknown|5|yes|380|tertiary|yes|entrepreneur|no|divorced|may|-1|unknown|0|no|
|58|121|1|unknown|5|no|50|primary|yes|retired|no|married|may|-1|unknown|0|no|
|43|593|1|unknown|5|no|55|secondary|yes|technician|no|single|may|-1|unknown|0|no|
|41|270|1|unknown|5|no|222|secondary|yes|admin.|no|divorced|may|-1|unknown|0|no|
|29|390|1|unknown|5|no|137|secondary|yes|admin.|no|single|may|-1|unknown|0|no|
|53|6|1|unknown|5|no|517|secondary|yes|technician|no|married|may|-1|unknown|0|no|
|58|71|1|unknown|5|no|71|unknown|yes|technician|no|married|may|-1|unknown|0|no|
|57|162|1|unknown|5|no|174|secondary|yes|services|no|married|may|-1|unknown|0|no|
|51|229|1|unknown|5|no|353|primary|yes|retired|no|married|may|-1|unknown|0|no|
|45|13|1|unknown|5|no|98|unknown|yes|admin.|no|single|may|-1|unknown|0|no|
|57|52|1|unknown|5|no|38|primary|yes|blue-collar|no|married|may|-1|unknown|0|no|
|60|60|1|unknown|5|no|219|primary|yes|retired|no|married|may|-1|unknown|0|no|
|33|0|1|unknown|5|no|54|secondary|yes|services|no|married|may|-1|unknown|0|no|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
only showing top 20 rows

scala>

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

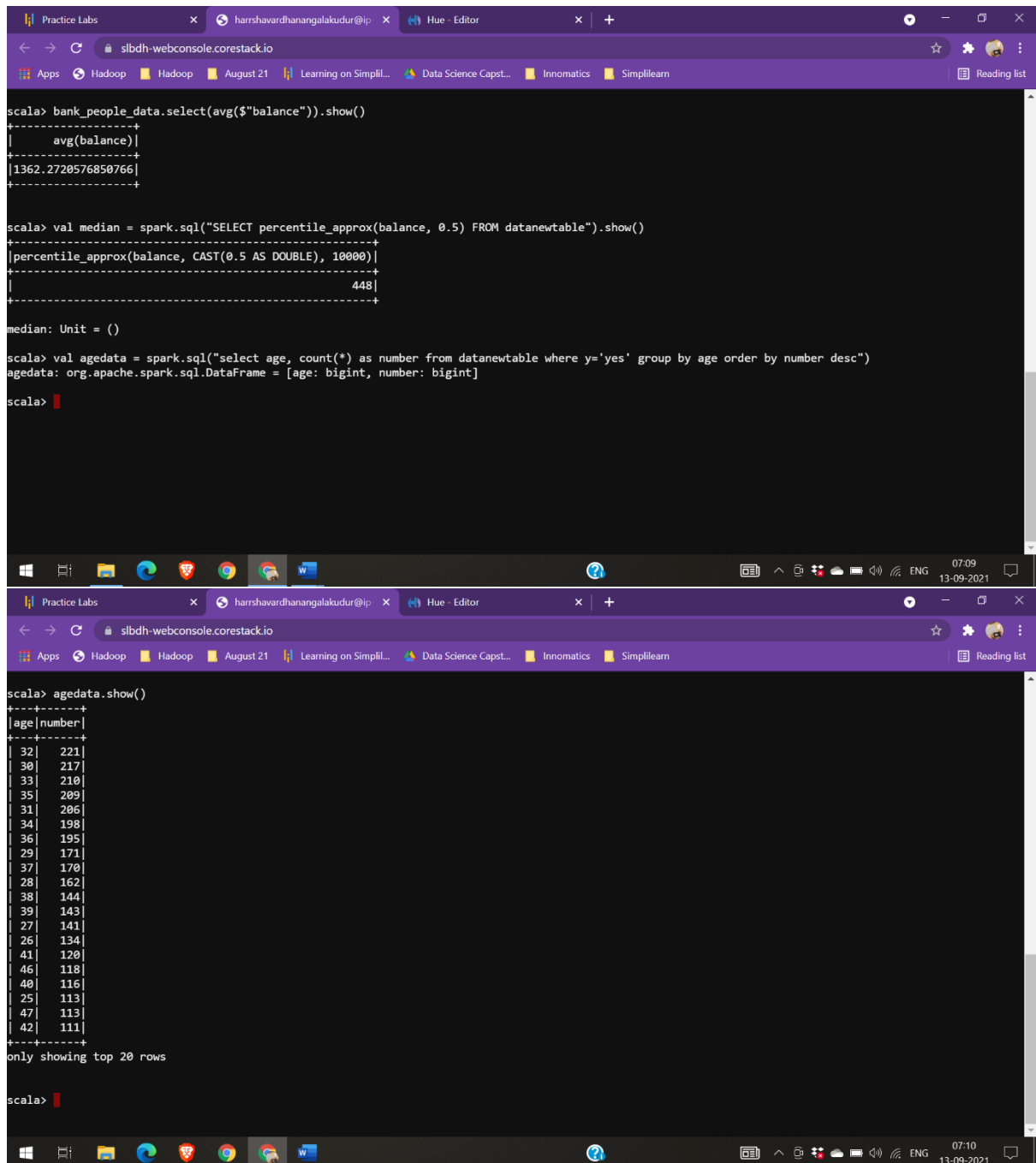
scala> bank_people_data.select(max($"age")).show()
+-----+
|max(age)|
+-----+
|95|
+-----+

scala> bank_people_data.select(min($"age")).show()
+-----+
|min(age)|
+-----+
|18|
+-----+

scala> bank_people_data.select(avg($"age")).show()
+-----+
|avg(age)|
+-----+
|40.93621021432837|
+-----+

scala>
```

Simplilearn Project 3 Market Analysis in Banking Domain



The screenshot displays a web console interface with two panels. The top panel shows the execution of Spark SQL queries to calculate the average balance and the median balance from a dataset named 'bank_people_data'. The bottom panel shows the execution of a query to filter and count data based on age and a 'yes' response, resulting in a DataFrame with 20 rows displayed.

```
scala> bank_people_data.select(avg($"balance")).show()
+-----+
|      avg(balance)|
+-----+
|1362.2720576850766|
+-----+

scala> val median = spark.sql("SELECT percentile_approx(balance, 0.5) FROM datanewtable").show()
+-----+
|percentile_approx(balance, CAST(0.5 AS DOUBLE), 10000)|
+-----+
|                        448|
+-----+

median: Unit = ()

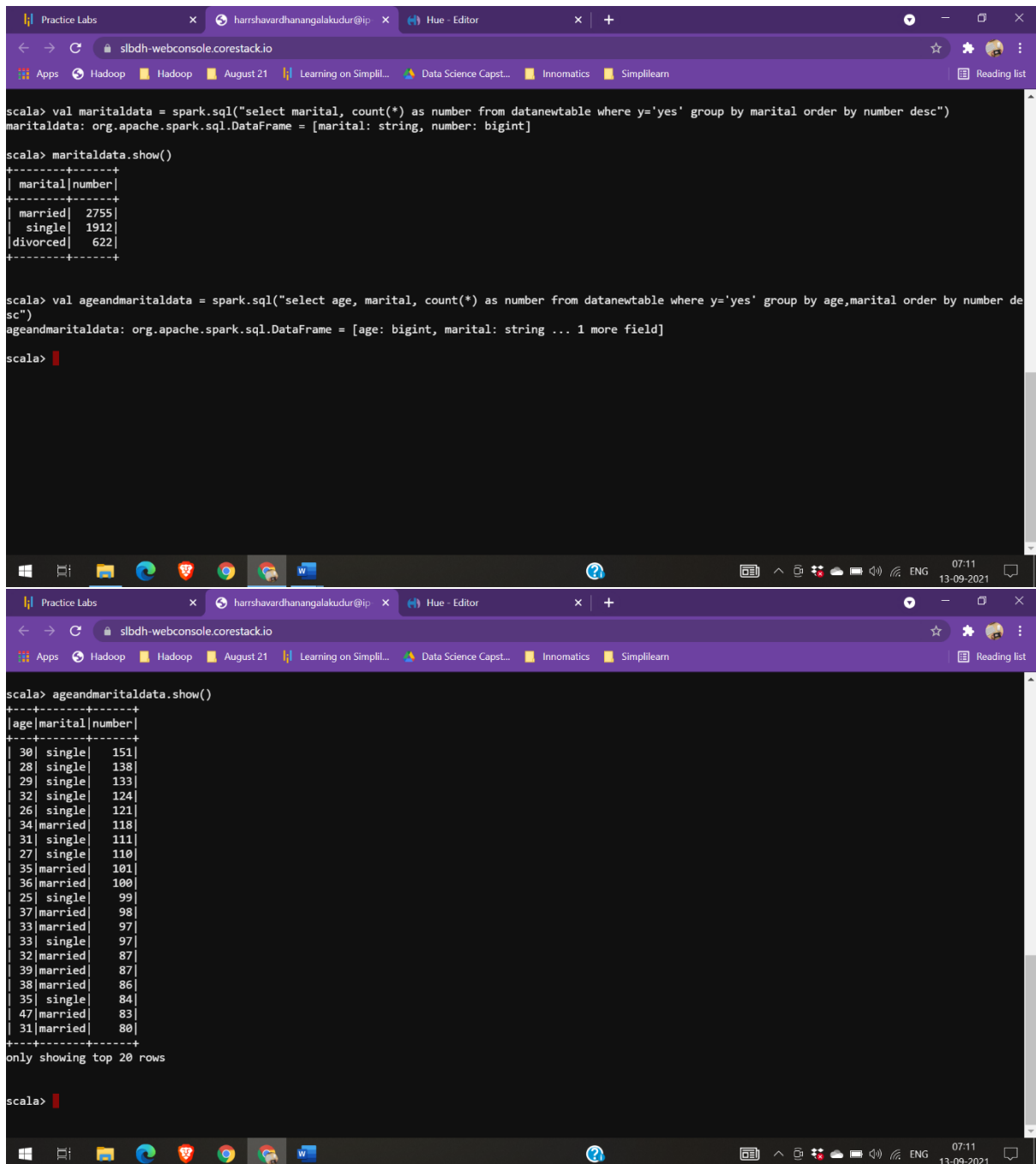
scala> val agedata = spark.sql("select age, count(*) as number from datanewtable where y='yes' group by age order by number desc")
agedata: org.apache.spark.sql.DataFrame = [age: bigint, number: bigint]

scala>
```

```
scala> agedata.show()
+----+-----+
|age|number|
+----+-----+
|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|
|25|  113|
|47|  113|
|42|  111|
+----+-----+
only showing top 20 rows

scala>
```

Simplilearn Project 3 Market Analysis in Banking Domain



The screenshot displays a web console interface with a purple header and a dark terminal area. The browser tabs include 'Practice Labs', 'harrshavardhanangalakudur@ip:', and 'Hue - Editor'. The address bar shows 'slbdh-webconsole.corestack.io'. The terminal shows the following code and output:

```
scala> val maritaldata = spark.sql("select marital, count(*) as number from datanewtable where y='yes' group by marital order by number desc")
maritaldata: org.apache.spark.sql.DataFrame = [marital: string, number: bigint]

scala> maritaldata.show()
+-----+-----+
| marital|number|
+-----+-----+
| married| 2755|
| single | 1912|
| divorced| 622|
+-----+-----+

scala> val ageandmaritaldata = spark.sql("select age, marital, count(*) as number from datanewtable where y='yes' group by age,marital order by number desc")
ageandmaritaldata: org.apache.spark.sql.DataFrame = [age: bigint, marital: string ... 1 more field]

scala>
```

The second screenshot shows the output of the second query:

```
scala> ageandmaritaldata.show()
+-----+-----+-----+
| age|marital|number|
+-----+-----+-----+
| 30| single | 151|
| 28| single | 138|
| 29| single | 133|
| 32| single | 124|
| 26| single | 121|
| 34| married| 118|
| 31| single | 111|
| 27| single | 110|
| 35| married| 101|
| 36| married| 100|
| 25| single | 99|
| 37| married| 98|
| 33| married| 97|
| 33| single | 97|
| 32| married| 87|
| 39| married| 87|
| 38| married| 86|
| 35| single | 84|
| 47| married| 83|
| 31| married| 80|
+-----+-----+-----+
only showing top 20 rows

scala>
```

Simplilearn Project 3 Market Analysis in Banking Domain

```
scala> val banknewDF = bank_people_data.withColumn("age",agedata(bank_people_data("age")))
banknewDF: org.apache.spark.sql.DataFrame = [age: string, balance: bigint ... 15 more fields]

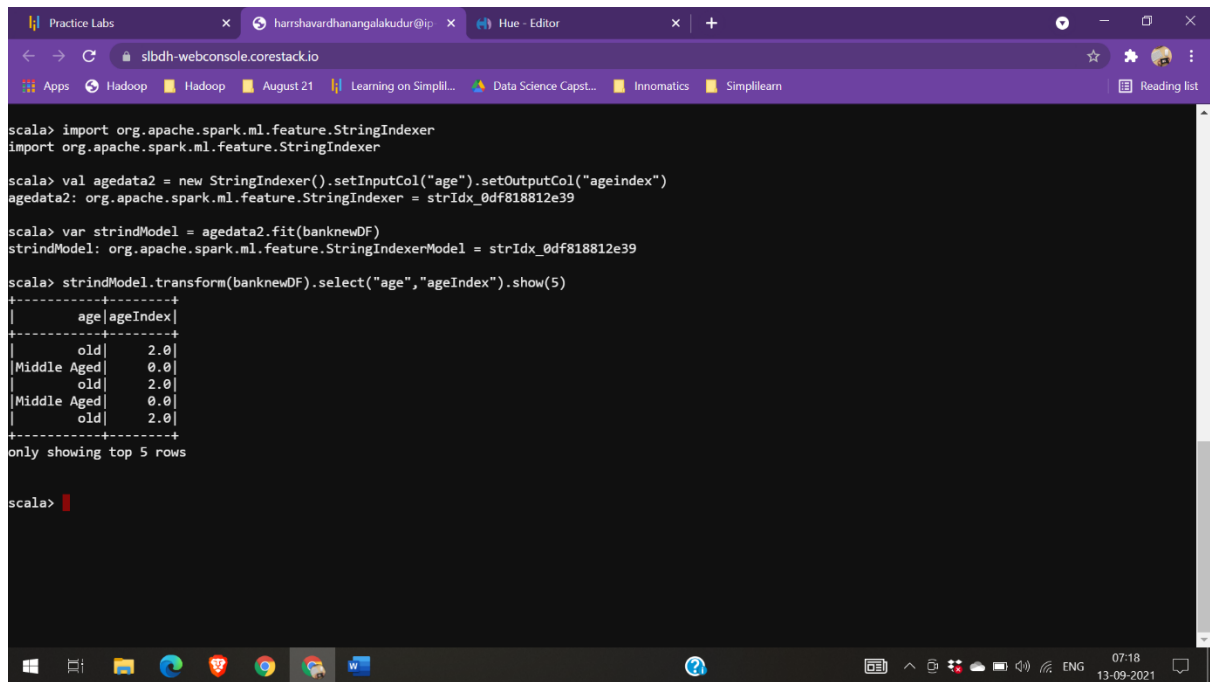
scala> banknewDF.show()
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|age|balance|campaign|contact|day|default|duration|education|housing|job|loan|marital|month|pdays|poutcome|previous|y|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|old|2143|1|unknown|5|no|261|tertiary|yes|management|no|married|may|-1|unknown|0|no|
|Middle Aged|29|1|unknown|5|no|151|secondary|yes|technician|no|single|may|-1|unknown|0|no|
|old|2|1|unknown|5|no|76|secondary|yes|entrepreneur|yes|married|may|-1|unknown|0|no|
|Middle Aged|1506|1|unknown|5|no|92|unknown|yes|blue-collar|no|married|may|-1|unknown|0|no|
|old|1|1|unknown|5|no|198|unknown|no|unknown|no|single|may|-1|unknown|0|no|
|Middle Aged|231|1|unknown|5|no|139|tertiary|yes|management|no|married|may|-1|unknown|0|no|
|Young|447|1|unknown|5|no|217|tertiary|yes|management|yes|single|may|-1|unknown|0|no|
|Middle Aged|2|1|unknown|5|yes|380|tertiary|yes|entrepreneur|no|divorced|may|-1|unknown|0|no|
|old|121|1|unknown|5|no|50|primary|yes|retired|no|married|may|-1|unknown|0|no|
|Middle Aged|593|1|unknown|5|no|55|secondary|yes|technician|no|single|may|-1|unknown|0|no|
|Middle Aged|270|1|unknown|5|no|222|secondary|yes|admin.|no|divorced|may|-1|unknown|0|no|
|Young|390|1|unknown|5|no|137|secondary|yes|admin.|no|single|may|-1|unknown|0|no|
|Middle Aged|6|1|unknown|5|no|517|secondary|yes|technician|no|married|may|-1|unknown|0|no|
|old|71|1|unknown|5|no|71|unknown|yes|technician|no|married|may|-1|unknown|0|no|
|old|162|1|unknown|5|no|174|secondary|yes|services|no|married|may|-1|unknown|0|no|
|Middle Aged|229|1|unknown|5|no|353|primary|yes|retired|no|married|may|-1|unknown|0|no|
|Middle Aged|13|1|unknown|5|no|98|unknown|yes|admin.|no|single|may|-1|unknown|0|no|
|old|52|1|unknown|5|no|38|primary|yes|blue-collar|no|married|may|-1|unknown|0|no|
|old|60|1|unknown|5|no|219|primary|yes|retired|no|married|may|-1|unknown|0|no|
|old|0|1|unknown|5|no|54|secondary|yes|services|no|married|may|-1|unknown|0|no|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
only showing top 20 rows

scala>
warning: there was one deprecation warning; re-run with -deprecation for details

scala> val targetage = spark.sql("select age, count(*) as number from banknewtable where y='yes' group by age order by number desc")
targetage: org.apache.spark.sql.DataFrame = [age: string, number: bigint]

scala> targetage.show()
+-----+-----+
|age|number|
+-----+-----+
|Middle Aged|2601|
|Young|1539|
|old|1131|
|Teen|18|
+-----+-----+
```

Simplilearn Project 3 Market Analysis in Banking Domain



The screenshot shows a web browser window with a terminal interface. The terminal displays the following Scala code and its output:

```
scala> import org.apache.spark.ml.feature.StringIndexer
import org.apache.spark.ml.feature.StringIndexer

scala> val agedata2 = new StringIndexer().setInputCol("age").setOutputCol("ageindex")
agedata2: org.apache.spark.ml.feature.StringIndexer = strIdx_0df818812e39

scala> var strindModel = agedata2.fit(banknewDF)
strindModel: org.apache.spark.ml.feature.StringIndexerModel = strIdx_0df818812e39

scala> strindModel.transform(banknewDF).select("age", "ageIndex").show(5)
+-----+-----+
|   age   | ageIndex |
+-----+-----+
|    old  |    2.0   |
|Middle Aged|    0.0   |
|    old  |    2.0   |
|Middle Aged|    0.0   |
|    old  |    2.0   |
+-----+-----+
only showing top 5 rows

scala>
```

The output shows a table with two columns: 'age' and 'ageIndex'. The 'age' column contains categorical values 'old' and 'Middle Aged', and the 'ageIndex' column contains their corresponding numerical indices, 2.0 and 0.0 respectively. The table is truncated after 5 rows.