Name: Prerna Sunil Jadhav
Sap Id: 60004220127
Batch: C2-2
course: Big Data Infrastructure laboratory
course code: DJ19CEEL6011

EXPERIMENT 08

AIM: To study and Implement SparkQL using PySpark.

THEORY: SparkQL is a query language for RDF data. It is used to retrieve and manipulate stored data in RDF format. SparkQL stands for "SparkQL Protocol and RDF Query language". It was developed by world wide web consortium (W3C) and is a standard for querying RDF data. RDF stands for Resource Description framework. It is a standard for describing resources on web. RDF data is stored in triplets, which consists of a subject, a predicate and an object. SparkQL Query is used to retrieve data from a RDF dataset. It consists of a set of patterns that match against RDF data. The patterns are written in a syntax similar to SQL, but with some differences.

Pyspark provides an easy-to-use interface to SparkSQL, allowing users to perform complex data processing tasks with few lines of code with PySpark. Users can create spark DataFrames, which is similar to Pandas DataFrame and can be queried using Spark SQL.

CONCLUSION: By studying and implementing SparkQL using PySpark, we leverage the power of SQL queries to effectively analyze and process big data within spark, enhancing data processin capabilities.

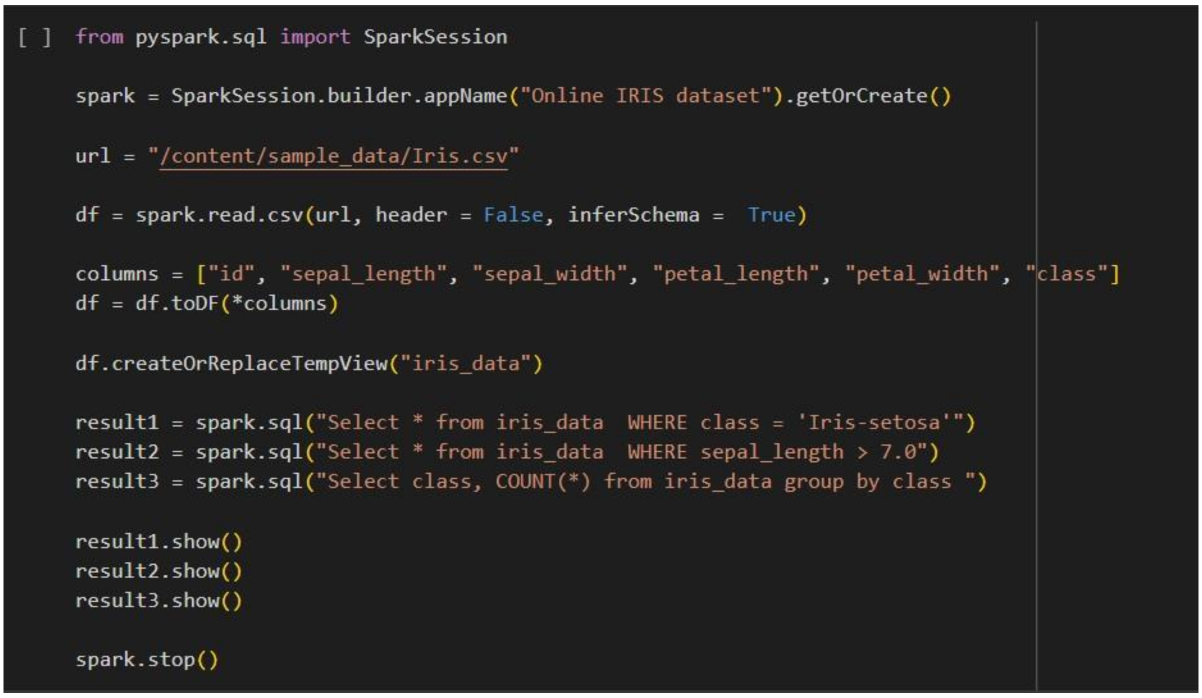
Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Academic Year: 2022-2023

Name:	Prerna Sunil Jadhav
Sap Id:	60004220127
Class:	T. Y. B. Tech (Computer Engineering)
Course:	Big Data Infrastructure Laboratory
Course Code:	DJ19CEEL6011
Experiment No.:	08

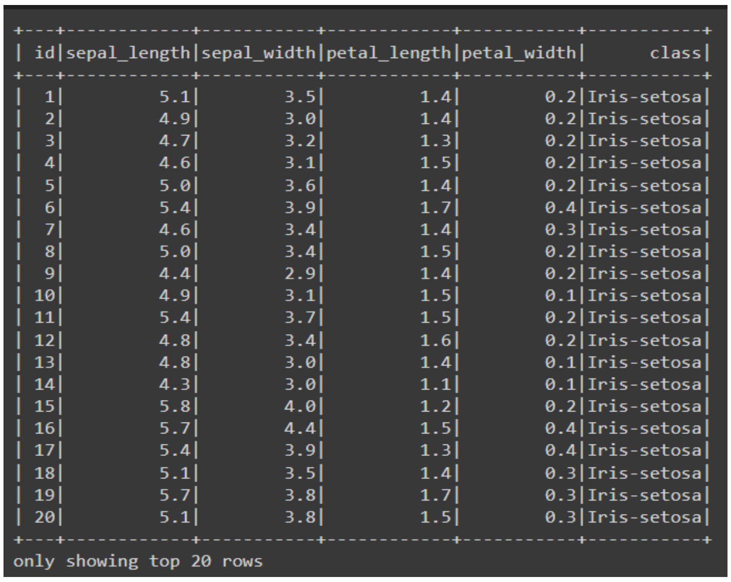
AIM: To study and Implement SparkQL using PySpark

IRIS DATASET:
CODE:

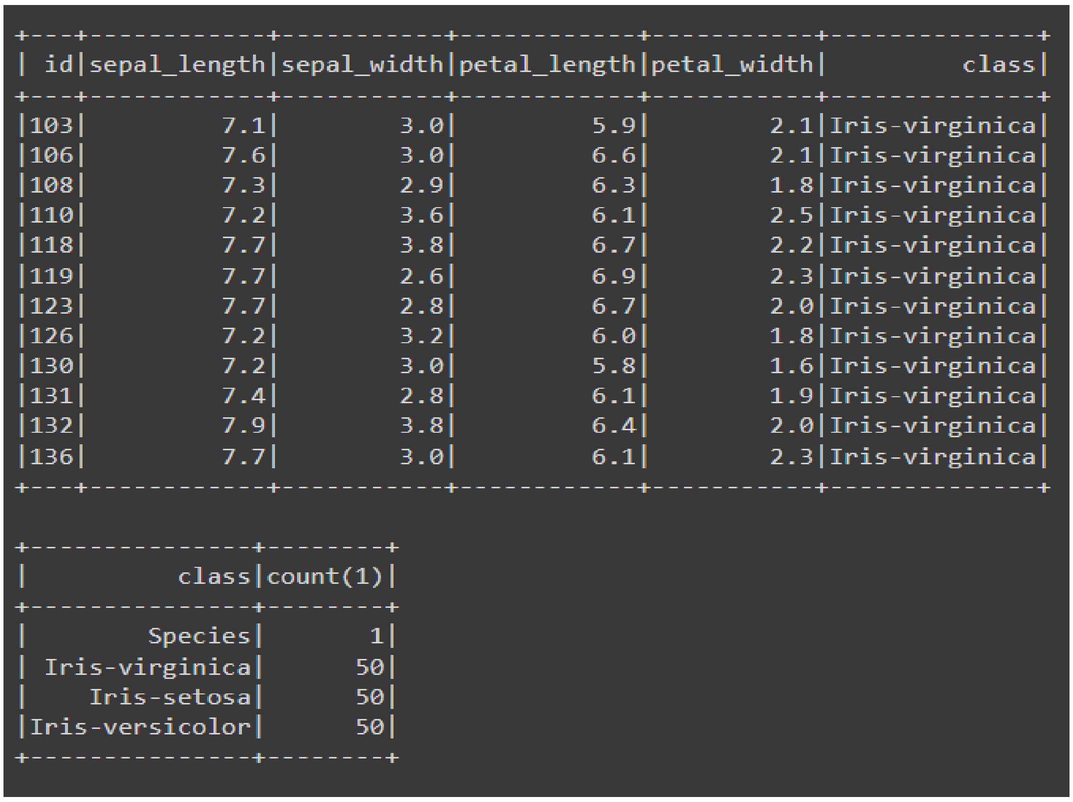


```
[ ] from pyspark.sql import SparkSession

    spark = SparkSession.builder.appName("Online IRIS dataset").getOrCreate()

    url = "/content/sample_data/Iris.csv"

    df = spark.read.csv(url, header = False, inferSchema =  True)

    columns = ["id", "sepal_length", "sepal_width", "petal_length", "petal_width", "class"]
    df = df.toDF(*columns)

    df.createOrReplaceTempView("iris_data")

    result1 = spark.sql("Select * from iris_data  WHERE class = 'Iris-setosa'")
    result2 = spark.sql("Select * from iris_data  WHERE sepal_length > 7.0")
    result3 = spark.sql("Select class, COUNT(*) from iris_data group by class ")

    result1.show()
    result2.show()
    result3.show()

    spark.stop()
```

OUTPUT:



```
+---+------------+-----------+------------+-----------+-----------+
| id|sepal_length|sepal_width|petal_length|petal_width|      class|
+---+------------+-----------+------------+-----------+-----------+
|  1|         5.1|        3.5|         1.4|        0.2|Iris-setosa|
|  2|         4.9|        3.0|         1.4|        0.2|Iris-setosa|
|  3|         4.7|        3.2|         1.3|        0.2|Iris-setosa|
|  4|         4.6|        3.1|         1.5|        0.2|Iris-setosa|
|  5|         5.0|        3.6|         1.4|        0.2|Iris-setosa|
|  6|         5.4|        3.9|         1.7|        0.4|Iris-setosa|
|  7|         4.6|        3.4|         1.4|        0.3|Iris-setosa|
|  8|         5.0|        3.4|         1.5|        0.2|Iris-setosa|
|  9|         4.4|        2.9|         1.4|        0.2|Iris-setosa|
| 10|         4.9|        3.1|         1.5|        0.1|Iris-setosa|
| 11|         5.4|        3.7|         1.5|        0.2|Iris-setosa|
| 12|         4.8|        3.4|         1.6|        0.2|Iris-setosa|
| 13|         4.8|        3.0|         1.4|        0.1|Iris-setosa|
| 14|         4.3|        3.0|         1.1|        0.1|Iris-setosa|
| 15|         5.8|        4.0|         1.2|        0.2|Iris-setosa|
| 16|         5.7|        4.4|         1.5|        0.4|Iris-setosa|
| 17|         5.4|        3.9|         1.3|        0.4|Iris-setosa|
| 18|         5.1|        3.5|         1.4|        0.3|Iris-setosa|
| 19|         5.7|        3.8|         1.7|        0.3|Iris-setosa|
| 20|         5.1|        3.8|         1.5|        0.3|Iris-setosa|
+---+------------+-----------+------------+-----------+-----------+
only showing top 20 rows
```

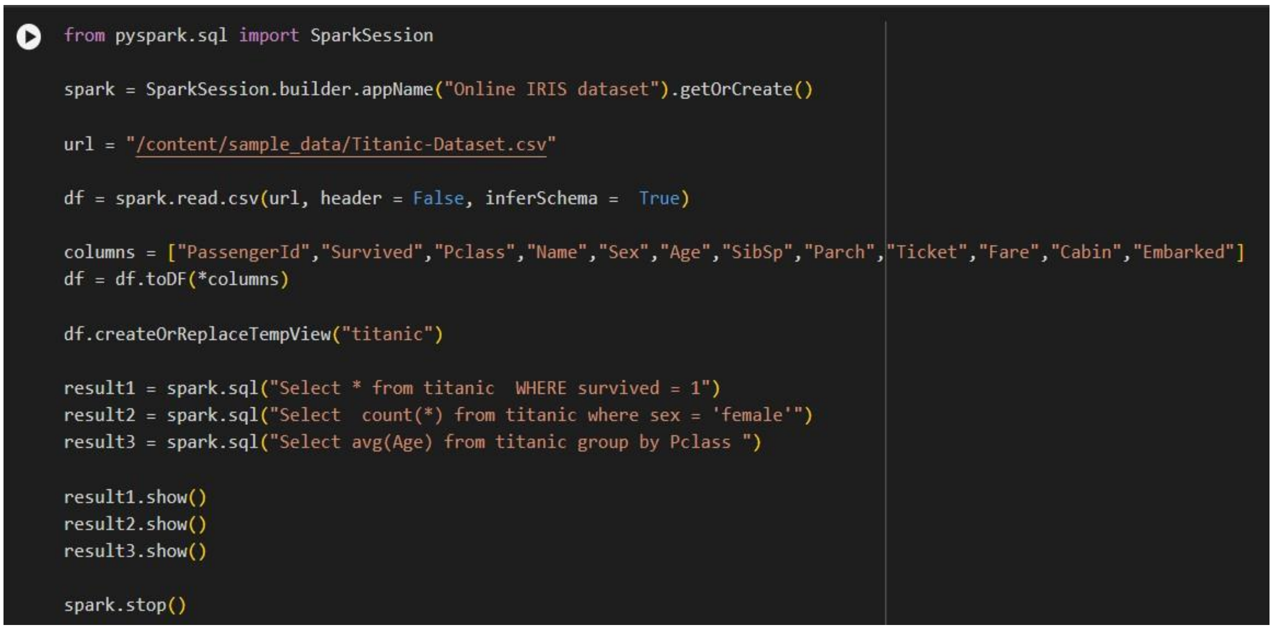


```
+---+------------+-----------+------------+-----------+--------------+
| id|sepal_length|sepal_width|petal_length|petal_width|         class|
+---+------------+-----------+------------+-----------+--------------+
|103|         7.1|        3.0|         5.9|        2.1|Iris-virginica|
|106|         7.6|        3.0|         6.6|        2.1|Iris-virginica|
|108|         7.3|        2.9|         6.3|        1.8|Iris-virginica|
|110|         7.2|        3.6|         6.1|        2.5|Iris-virginica|
|118|         7.7|        3.8|         6.7|        2.2|Iris-virginica|
|119|         7.7|        2.6|         6.9|        2.3|Iris-virginica|
|123|         7.7|        2.8|         6.7|        2.0|Iris-virginica|
|126|         7.2|        3.2|         6.0|        1.8|Iris-virginica|
|130|         7.2|        3.0|         5.8|        1.6|Iris-virginica|
|131|         7.4|        2.8|         6.1|        1.9|Iris-virginica|
|132|         7.9|        3.8|         6.4|        2.0|Iris-virginica|
|136|         7.7|        3.0|         6.1|        2.3|Iris-virginica|
+---+------------+-----------+------------+-----------+--------------+

+---------------+--------+
|          class|count(1)|
+---------------+--------+
|        Species|       1|
| Iris-virginica|      50|
|    Iris-setosa|      50|
|Iris-versicolor|      50|
+---------------+--------+
```

<u>**Titanic Dataset - Perform SQL Queries to find**</u>:

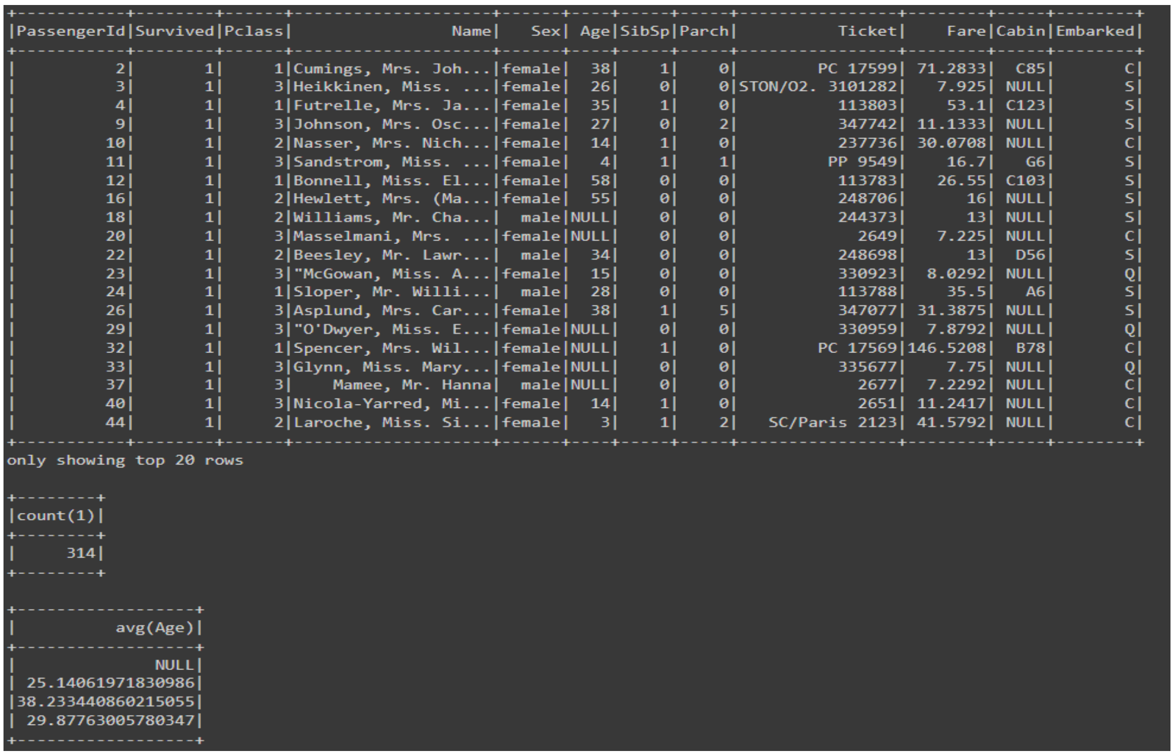
A. **What is the number of passengers who survived the Titanic Disaster?**
B. **How many female passengers were on board the Titanic?**
C. **What is the average age of passengers in each passenger class?**

CODE:



```
from pyspark.sql import SparkSession

spark = SparkSession.builder.appName("Online IRIS dataset").getOrCreate()

url = "/content/sample_data/Titanic-Dataset.csv"

df = spark.read.csv(url, header = False, inferSchema =  True)

columns = ["PassengerId","Survived","Pclass","Name","Sex","Age","SibSp","Parch","Ticket","Fare","Cabin","Embarked"]
df = df.toDF(*columns)

df.createOrReplaceTempView("titanic")

result1 = spark.sql("Select * from titanic  WHERE survived = 1")
result2 = spark.sql("Select  count(*) from titanic where sex = 'female'")
result3 = spark.sql("Select avg(Age) from titanic group by Pclass ")

result1.show()
result2.show()
result3.show()

spark.stop()
```

OUTPUT:



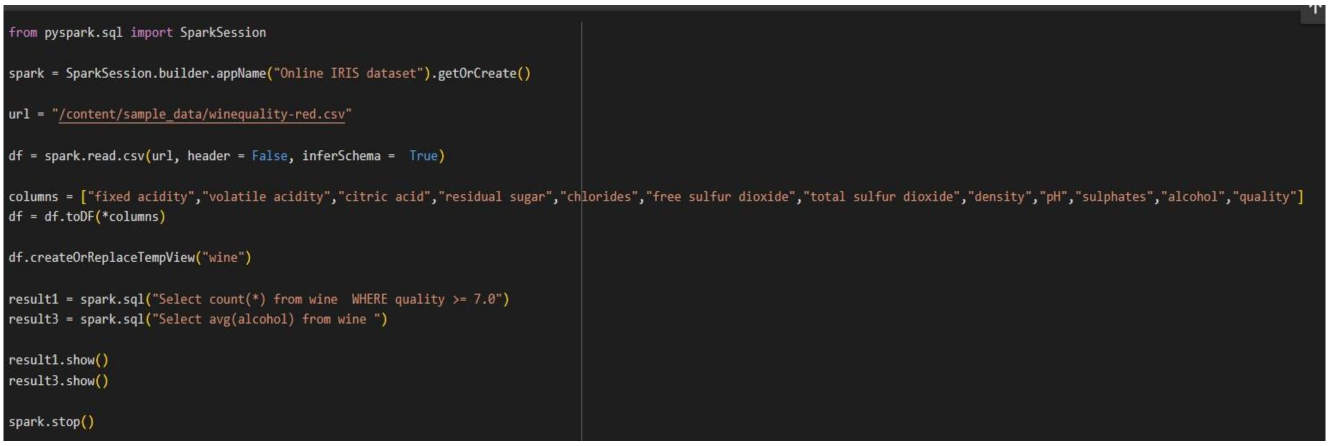
```
+-----------+--------+------+--------------------+------+----+-----+-----+----------------+--------+-----+--------+
|PassengerId|Survived|Pclass|                Name|   Sex| Age|SibSp|Parch|          Ticket|    Fare|Cabin|Embarked|
+-----------+--------+------+--------------------+------+----+-----+-----+----------------+--------+-----+--------+
|          2|       1|     1|Cumings, Mrs. Joh...|female|  38|    1|    0|        PC 17599| 71.2833|  C85|       C|
|          3|       1|     3|Heikkinen, Miss. ...|female|  26|    0|    0|STON/O2. 3101282|   7.925| NULL|       S|
|          4|       1|     1|Futrelle, Mrs. Ja...|female|  35|    1|    0|          113803|    53.1| C123|       S|
|          9|       1|     3|Johnson, Mrs. Osc...|female|  27|    0|    2|          347742| 11.1333| NULL|       S|
|         10|       1|     2|Nasser, Mrs. Nich...|female|  14|    1|    0|          237736| 30.0708| NULL|       C|
|         11|       1|     3|Sandstrom, Miss. ...|female|   4|    1|    1|         PP 9549|    16.7|   G6|       S|
|         12|       1|     1|Bonnell, Miss. El...|female|  58|    0|    0|          113783|   26.55| C103|       S|
|         16|       1|     2|Hewlett, Mrs. (Ma...|female|  55|    0|    0|          248706|      16| NULL|       S|
|         18|       1|     2|Williams, Mr. Cha...|  male|NULL|    0|    0|          244373|      13| NULL|       S|
|         20|       1|     3|Masselmani, Mrs. ...|female|NULL|    0|    0|            2649|   7.225| NULL|       C|
|         22|       1|     2|Beesley, Mr. Lawr...|  male|  34|    0|    0|          248698|      13|  D56|       S|
|         23|       1|     3|"McGowan, Miss. A...|female|  15|    0|    0|          330923|  8.0292| NULL|       Q|
|         24|       1|     1|Sloper, Mr. Willi...|  male|  28|    0|    0|          113788|    35.5|   A6|       S|
|         26|       1|     3|Asplund, Mrs. Car...|female|  38|    1|    5|          347077| 31.3875| NULL|       S|
|         29|       1|     3|"O'Dwyer, Miss. E...|female|NULL|    0|    0|          330959|  7.8792| NULL|       Q|
|         32|       1|     1|Spencer, Mrs. Wil...|female|NULL|    1|    0|        PC 17569|146.5208|  B78|       C|
|         33|       1|     3|Glynn, Miss. Mary...|female|NULL|    0|    0|          335677|    7.75| NULL|       Q|
|         37|       1|     3|    Mamee, Mr. Hanna|  male|NULL|    0|    0|            2677|  7.2292| NULL|       C|
|         40|       1|     3|Nicola-Yarred, Mi...|female|  14|    1|    0|            2651| 11.2417| NULL|       C|
|         44|       1|     2|Laroche, Miss. Si...|female|   3|    1|    2|   SC/Paris 2123| 41.5792| NULL|       C|
+-----------+--------+------+--------------------+------+----+-----+-----+----------------+--------+-----+--------+
only showing top 20 rows

+--------+
|count(1)|
+--------+
|     314|
+--------+

+------------------+
|          avg(Age)|
+------------------+
|              NULL|
| 25.14061971830986|
|38.233440860215055|
| 29.87763005780347|
+------------------+
```

Wine Quality Dataset Example

A. **How many wines are considered high quality (quality score of 7 or higher)**

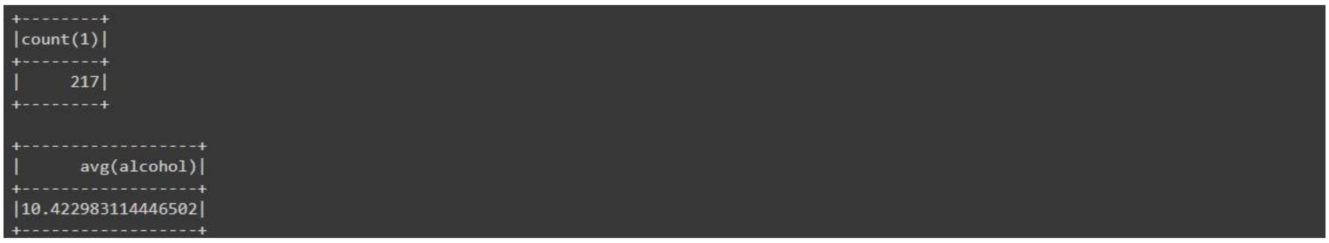
B. **What is the average alcohol content of the wines in the dataset**

CODE:



```
from pyspark.sql import SparkSession

spark = SparkSession.builder.appName("Online IRIS dataset").getOrCreate()

url = "/content/sample_data/winequality-red.csv"

df = spark.read.csv(url, header = False, inferSchema =  True)

columns = ["fixed acidity","volatile acidity","citric acid","residual sugar","chlorides","free sulfur dioxide","total sulfur dioxide","density","pH","sulphates","alcohol","quality"]
df = df.toDF(*columns)

df.createOrReplaceTempView("wine")

result1 = spark.sql("Select count(*) from wine  WHERE quality >= 7.0")
result3 = spark.sql("Select avg(alcohol) from wine ")

result1.show()
result3.show()

spark.stop()
```

OUTPUT:

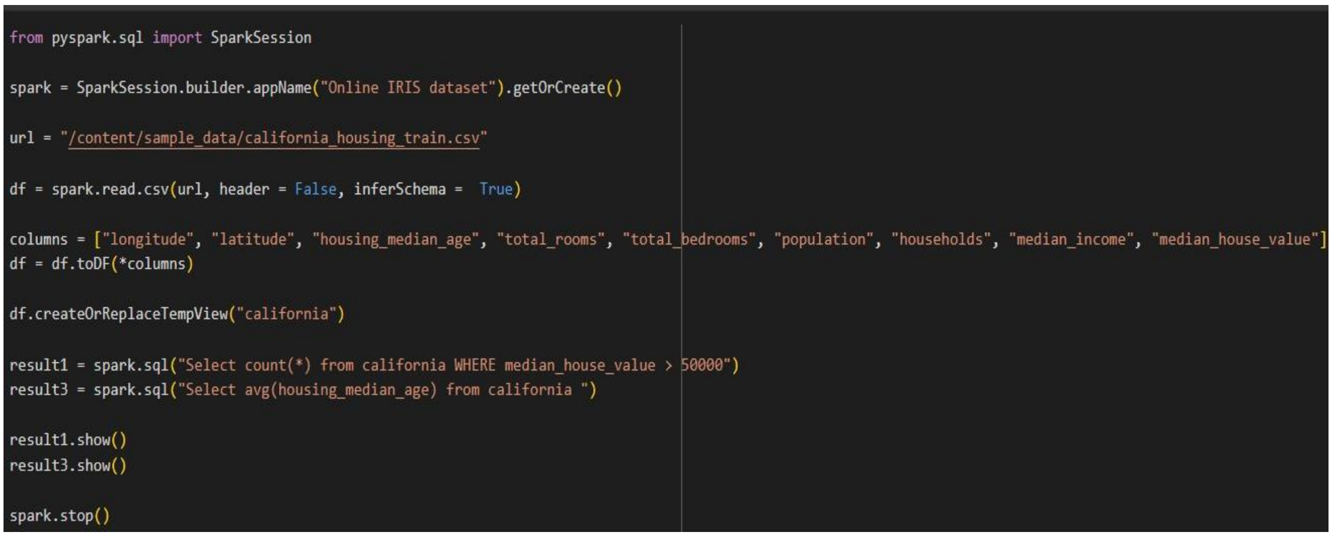


```
+--------+
|count(1)|
+--------+
|     217|
+--------+

+------------------+
|      avg(alcohol)|
+------------------+
|10.422983114446502|
+------------------+
```

California Housing Dataset Example:

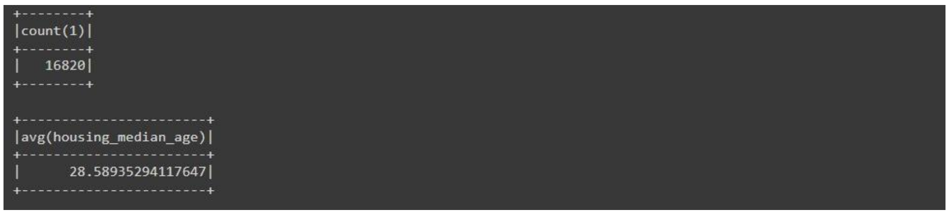
A. **How many houses have a median value above $5000,000 in California**
B. **What is the average age of the houses in the dataset**

CODE:



```
from pyspark.sql import SparkSession

spark = SparkSession.builder.appName("Online IRIS dataset").getOrCreate()

url = "/content/sample_data/california_housing_train.csv"

df = spark.read.csv(url, header = False, inferSchema =  True)

columns = ["longitude", "latitude", "housing_median_age", "total_rooms", "total_bedrooms", "population", "households", "median_income", "median_house_value"]
df = df.toDF(*columns)

df.createOrReplaceTempView("california")

result1 = spark.sql("Select count(*) from california WHERE median_house_value > 50000")
result3 = spark.sql("Select avg(housing_median_age) from california ")

result1.show()
result3.show()

spark.stop()
```

OUTPUT:



```
+--------+
|count(1)|
+--------+
|   16820|
+--------+

+-----------------------+
|avg(housing_median_age)|
+-----------------------+
|      28.58935294117647|
+-----------------------+
```