Use Case 4

Gabriel D Hofer

July 20, 2021

Apply data frames on Walmart dataset and solve the below problems:

1. Create a new dataframe with a column called HV Ratio that is the ratio of the High Price versus volume of stock traded for a day.

```
scala> val df = spark.read.option("header",true).csv("/home/gabriel/Data/
    walmart_stock.txt")
```

```
scala> df.withColumn("HV Ratio", col("High") / col("Volume"))
```

2. What day had the Peak High in Price?

```
scala> df.sort(col("High").desc).show(1)
+----------+---------+---------+-----+---------+-------+---------+
|      Date|     Open|     High|  Low|    Close| Volume|Adj Close|
+----------+---------+---------+-----+---------+-------+---------+
|2015-01-13|90.800003|90.970001|88.93|89.309998|8215400|83.825448|
+----------+---------+---------+-----+---------+-------+---------+
only showing top 1 row
```

3. What is the mean of the Close column?

```
scala> df.select(mean(col("Close"))).show()
+-----------------+
|       avg(Close)|
+-----------------+
|72.38844998012726|
+-----------------+
```

4. What is the max and min of the Volume column?

```
scala> df.agg(max(col("Volume")), min(col("Volume"))).show()
+-----------+-----------+
|max(Volume)|min(Volume)|
+-----------+-----------+
|    9994400|   10010500|
+-----------+-----------+
```

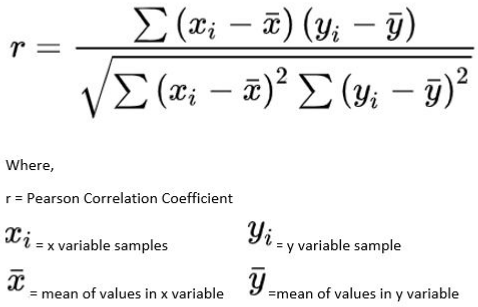
5. How many days was the Close lower than 60 dollars?

```
scala> df.filter(col("Close") < lit(60.0)).count()
res10: Long = 81
```

6. What percentage of the time was the High greater than 80 dollars ?

```
scala> df.agg((sum(when(col("High") > lit(80.0),1).otherwise(0)) / count("*")).as("High > 80")).show()
+-------------------+
|          High > 80|
+-------------------+
|0.09141494435612083|
+-------------------+
```

7. What is the Pearson correlation between High and Volume?



$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

Where,

r = Pearson Correlation Coefficient

x_i = x variable samples y_i = y variable sample

$\bar{x}$ = mean of values in x variable $\bar{y}$ =mean of values in y variable

```
scala> val highAvg = df.select(mean(col("High"))).first().getDouble(0)

scala> val volumeAvg = df.select(mean(col("Volume"))).first().getDouble(0)

scala> val numerator = df.agg(sum((col("High") - highAvg) * (col("Volume") - volumeAvg))).first().getDouble(0)
numerator: Double = -1.3013592106342264E10

scala> val denominator = math.sqrt(df.agg( sum((col("High") - highAvg) * (col("High") - highAvg)) * sum((col("Volume") - volumeAvg) * (col("Volume") - volumeAvg))).first().getDouble(0))
denominator: Double = 3.845253639556948E10

scala> val r = numerator / denominator
r: Double = -0.33843260617371645
```

8. What is the max High per year?

```
scala> df.groupBy(substring(col("Date"),0,4).as("Year")).agg(max(col("High"))).orderBy(col("Year")).show()
+----+---------+
|Year|max(High)|
+----+---------+
|2012|77.599998|
|2013|81.370003|
|2014|88.089996|
|2015|90.970001|
|2016|75.190002|
+----+---------+
```

9. What is the average Close for each Calendar Month?

```

scala> df.groupBy(substring(col("Date"),0,7).as("Year-Month")).agg(avg(col("Close"))).show(100)
+----------+------------------+
|Year-Month|        avg(Close)|
+----------+------------------+
|   2013-05| 77.81636368181817|
|   2013-09|        74.4395005|
|   2013-12|  78.7752382857143|
|   2013-06| 74.97800020000001|
|   2016-02| 66.24800044999999|
|   2015-05| 77.33599970000002|
|   2012-08| 73.04478265217392|
|   2015-12| 59.98681827272728|
|   2012-02|            60.898|
|   2012-04|60.149000150000006|
|   2016-12| 70.51904728571428|
|   2012-05|61.456363409090905|
|   2016-09| 72.00857180952381|
|   2016-03| 67.55499963636365|
|   2016-10| 69.23952366666666|
|   2012-12| 69.71100009999999|
|   2012-07| 72.40666661904763|
|   2014-01| 76.53142833333334|
|   2015-02| 85.52315805263159|
|   2015-08|  69.2866677142857|
|   2014-03| 75.30238076190474|
|   2014-08| 74.67666623809525|
|   2013-11| 78.97300075000001|
|   2014-02| 74.05578978947368|
|   2012-01|        60.2354999|
|   2015-10|61.564545636363626|
|   2012-11| 71.10952333333333|
|   2013-02| 70.62315857894738|
|   2015-11|58.911999949999995|
|   2014-09| 76.33619004761903|
|   2016-05| 68.05285676190476|
|   2012-06| 67.50380961904762|
|   2013-07| 77.11545418181818|
|   2014-05| 77.38095276190477|
|   2012-09| 74.18157921052631|
|   2015-01|       87.60949975|
|   2016-04| 68.82523861904761|
|   2016-11| 70.30476261904762|
|   2013-08| 75.22409204545455|
|   2014-11| 81.88526321052632|
|   2016-01| 63.22105263157895|
|   2016-07| 73.54149939999999|
|   2015-07| 72.75000036363635|
|   2016-06| 71.34636304545454|
|   2014-04| 77.80857085714285|
|   2014-10| 76.48869486956522|
|   2015-09| 64.25238128571429|
|   2014-12|  85.1259102727273|
|   2013-03| 73.43649940000002|
|   2013-10| 74.97913104347826|
|   2014-06| 76.01000033333332|
|   2012-10| 75.30619061904761|
|   2014-07| 76.21090877272728|
|   2015-04| 79.56047561904764|
|   2015-06| 72.79727304545456|
|   2013-01| 69.09476142857143|
|   2016-08|  72.8300000869565|
|   2013-04| 77.68954572727273|
|   2012-03|60.433636818181796|
|   2015-03| 82.47318172727273|
+----------+------------------+
```