Module 16 Challenge

New Attempt

Due Apr 25 by 12:59am **Points** 100 **Submitting** a text entry box or a website url

Background

Since your work with Jennifer on the SellBy project was so successful, you've been tasked with another, larger project: analyzing Amazon reviews written by members of the paid Amazon Vine program. The Amazon Vine program is a service that allows manufacturers and publishers to receive reviews for their products. Companies like SellBy pay a small fee to Amazon and provide products to Amazon Vine members, who are then required to publish a review.

In this project, you'll have access to approximately 50 datasets. Each one contains reviews of a specific product, from clothing apparel to wireless products. You'll need to pick one of these datasets and use PySpark to perform the ETL process to extract the dataset, transform the data, connect to an AWS RDS instance, and load the transformed data into pgAdmin. Next, you'll use PySpark, Pandas, or SQL to determine if there is any bias toward favorable reviews from Vine members in your dataset. Then, you'll write a summary of the analysis for Jennifer to submit to the SellBy stakeholders.

What You're Creating

This new assignment consists of two technical analysis deliverables and a written report. You will submit the following:

- Deliverable 1: Perform ETL on Amazon Product Reviews
- Deliverable 2: Determine Bias of Vine Reviews
- Deliverable 3: A Written Report on the Analysis (README.md)

Files

Use the following links to download the Challenge starter codes.

Download the **SQL table schema** (https://2u-data-curriculum-team.s3.amazonaws.com/dataviz-online/module_16/challenge_schema.sql).

Download the **Amazon ETL starter code** (https://2u-data-curriculum-team.s3.amazonaws.com/dataviz-online/module_16/Amazon_Reviews_ETL_starter_code.ipynb).

Before You Start

Create a new GitHub repository entitled "Amazon_Vine_Analysis", and initialize the repository with a README.

IMPORTANT

Be sure to regularly monitor your AWS usage so you don't go above the free tier. Consult Lessons **16.9.2** and **16.9.3** for more on shutting down your AWS instances and checking your AWS billing, respectively.

Deliverable 1: Perform ETL on Amazon Product Reviews (40 points)

Deliverable 1 Instructions

Using your knowledge of the cloud ETL process, you'll create an AWS RDS database with tables in pgAdmin, pick a dataset from the **Amazon Review datasets** **(https://s3.amazonaws.com/amazon-reviews-pds/tsv/index.txt)**, and extract the dataset into a DataFrame. You'll transform the DataFrame into four separate DataFrames that match the table schema in pgAdmin. Then, you'll upload the transformed data into the appropriate tables and run queries in pgAdmin to confirm that the data has been uploaded.

REWIND

For this deliverable, you've already done the following in this module:

- **Lesson 16.4.2:** Use PySpark to read in a CSV file
- **Lesson 16.4.2:** Use PySpark methods and functions to get DataFrame information
- **Lesson 16.4.3:** Use PySpark functions to transform and filter data
- **Lesson 16.7.2:** Create a RDS on AWS
- **Lesson 16.7.3:** Connect pgAdmin to a AWS RDS
- **Lesson 16.9.1:** Use PySpark to perform ETL
- **Lesson 16.9.2:** Shut down your AWS Instance
- **Lesson 16.9.3:** Check AWS billing

Follow the instructions below to complete Deliverable 1.

1. From the following **Amazon Review datasets** **(https://s3.amazonaws.com/amazon-reviews-pds/tsv/index.txt)**, pick a dataset that you would like to analyze. All the datasets have the same schemata, as shown in this image:

```
DATA COLUMNS:
marketplace       - 2 letter country code of the marketplace where the review was written.
customer_id       - Random identifier that can be used to aggregate reviews written by a single author.
review_id         - The unique ID of the review.
product_id        - The unique Product ID the review pertains to. In the multilingual dataset the reviews
                    for the same product in different countries can be grouped by the same product_id.
product_parent    - Random identifier that can be used to aggregate reviews for the same product.
product_title     - Title of the product.
product_category  - Broad product category that can be used to group reviews
                    (also used to group the dataset into coherent parts).
star_rating       - The 1-5 star rating of the review.
helpful_votes     - Number of helpful votes.
total_votes       - Number of total votes the review received.
vine              - Review was written as part of the Vine program.
verified_purchase - The review is on a verified purchase.
review_headline   - The title of the review.
review_body       - The review text.
review_date       - The date the review was written.
```

2. Create a new database with Amazon RDS just as you did in this module.
3. In pgAdmin, create a new database in your Amazon RDS server that you just create.
4. Download the `challenge_schema.sql` file to your computer.
5. In pgAdmin, run a new query to create the tables for your new database using the code from the `challenge_schema.sql` file.
    - After you run the query, you should have the following four tables in your database: `customers_table`, `products_table`, `review_id_table`, and `vine_table`.
6. Download the `Amazon_Reviews_ETL_starter_code.ipynb` file, then upload the file as a Google Colab Notebook, and rename it `Amazon_Reviews_ETL`.

NOTE

If you try to open the `Amazon_Reviews_ETL_starter_code.ipynb` with jupyter notebook it will give you an error.

7. First **extract** one of the review datasets, then create a new DataFrame.
8. Next, follow the steps below to **transform** the dataset into four DataFrames that will match the schema in the pgAdmin tables:

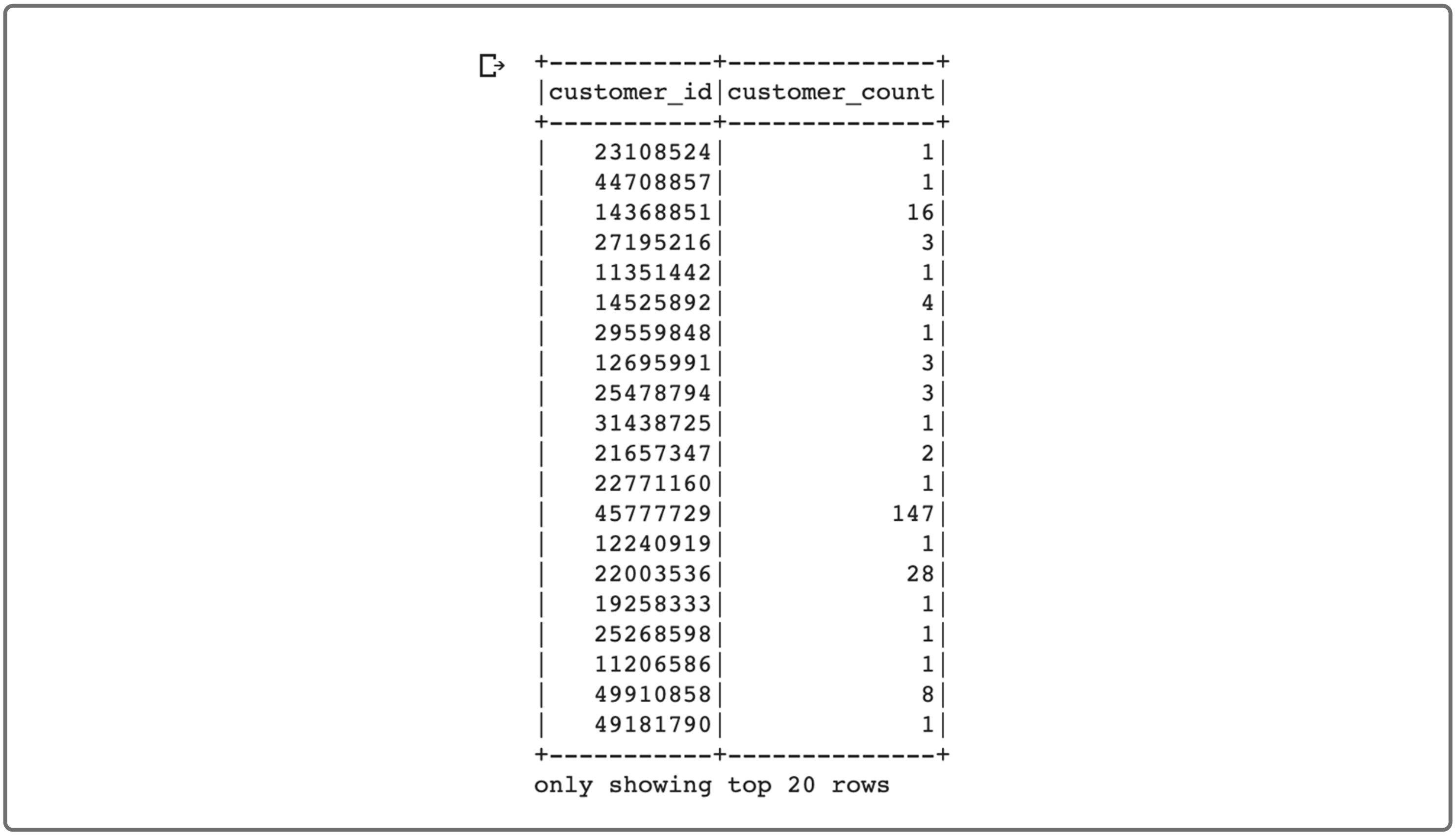
NOTE

Some datasets have a large number of rows, which will affect the time it takes to complete the following steps.

The customers_table DataFrame

To create the `customers_table`, use the code in the `Amazon_Reviews_ETL_starter_code.ipynb` file and follow the steps below to aggregate the reviews by `customer_id`.

- Use the `groupby()` function on the `customer_id` column of the DataFrame you created in Step 6.
- Count all the customer ids using the `agg()` function by chaining it to the `groupby()` function. After you use this function, a new column will be created, `count(customer_id)`.
- Rename the `count(customer_id)` column using the `withColumnRenamed()` function so it matches the schema for the `customers_table` in pgAdmin.
- The final `customers_table` DataFrame should look like this:

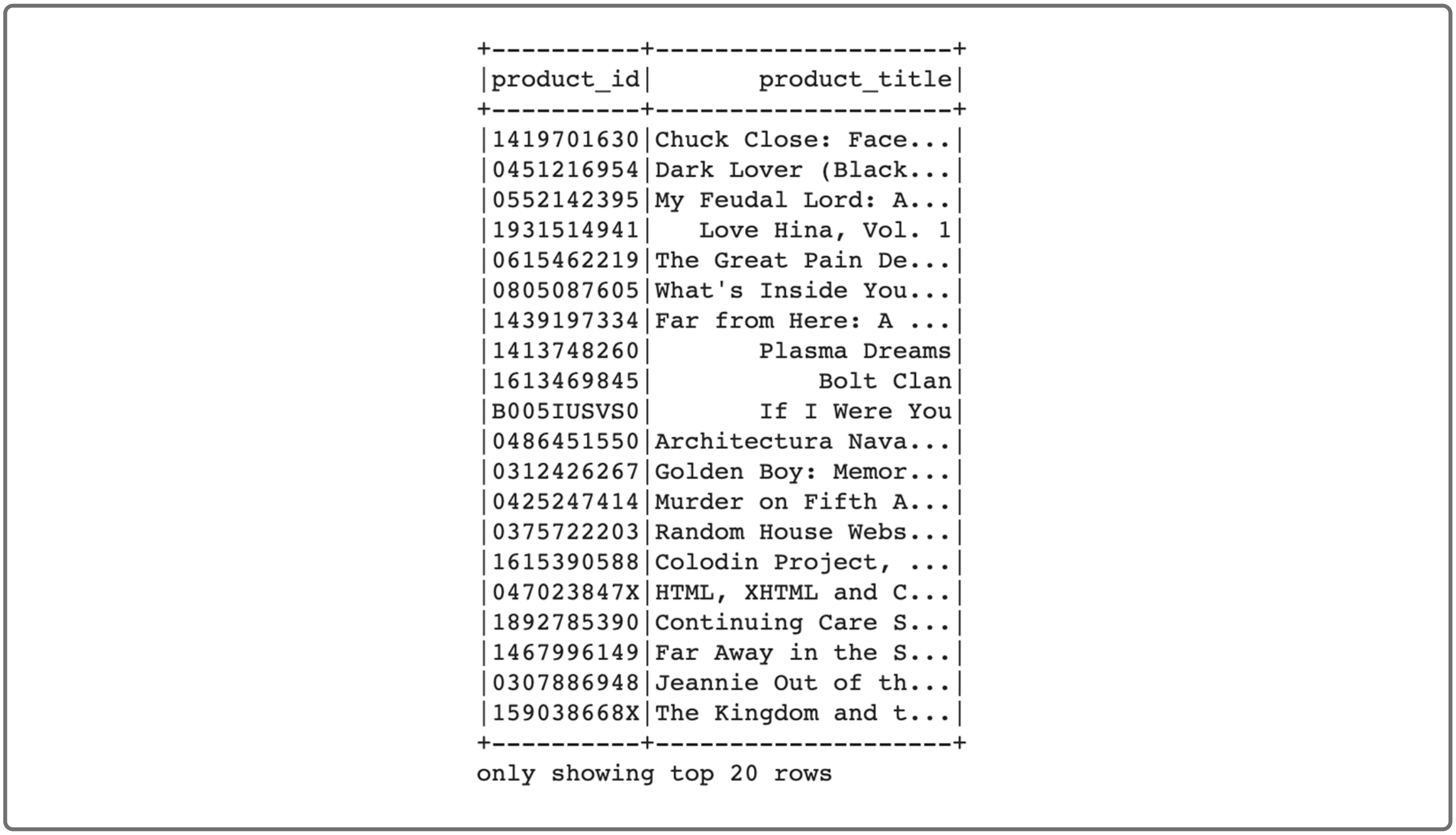


```
+-----------+--------------+
|customer_id|customer_count|
+-----------+--------------+
|   23108524|             1|
|   44708857|             1|
|   14368851|            16|
|   27195216|             3|
|   11351442|             1|
|   14525892|             4|
|   29559848|             1|
|   12695991|             3|
|   25478794|             3|
|   31438725|             1|
|   21657347|             2|
|   22771160|             1|
|   45777729|           147|
|   12240919|             1|
|   22003536|            28|
|   19258333|             1|
|   25268598|             1|
|   11206586|             1|
|   49910858|             8|
|   49181790|             1|
+-----------+--------------+
only showing top 20 rows
```

The products_table DataFrame

To create the `products_table`, use the `select()` function to select the `product_id` and `product_title`, then drop duplicates with the `drop_duplicates()` function to retrieve only unique `product_ids`. Refer to the code snippet provided in the `Amazon_Reviews_ETL_starter_code.ipynb` file for assistance.

The final `products_table` DataFrame should look like this:

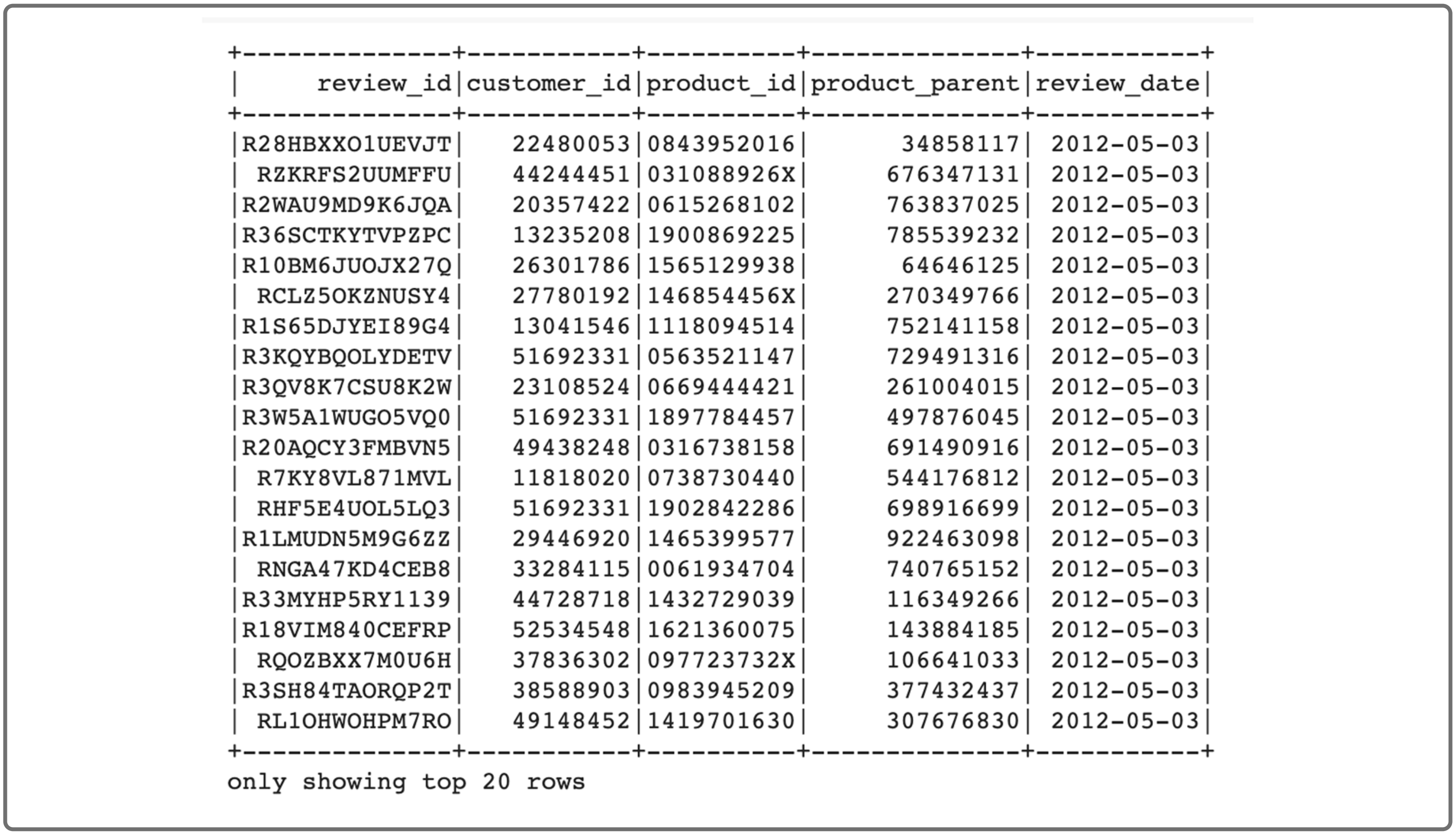


```
+----------+--------------------+
|product_id|       product_title|
+----------+--------------------+
|1419701630|Chuck Close: Face...|
|0451216954|Dark Lover (Black...|
|0552142395|My Feudal Lord: A...|
|1931514941|   Love Hina, Vol. 1|
|0615462219|The Great Pain De...|
|0805087605|What's Inside You...|
|1439197334|Far from Here: A ...|
|1413748260|       Plasma Dreams|
|1613469845|           Bolt Clan|
|B005IUSVS0|       If I Were You|
|0486451550|Architectura Nava...|
|0312426267|Golden Boy: Memor...|
|0425247414|Murder on Fifth A...|
|0375722203|Random House Webs...|
|1615390588|Colodin Project, ...|
|047023847X|HTML, XHTML and C...|
|1892785390|Continuing Care S...|
|1467996149|Far Away in the S...|
|0307886948|Jeannie Out of th...|
|159038668X|The Kingdom and t...|
+----------+--------------------+
only showing top 20 rows
```

The review_id_table DataFrame

To create the `review_id_table`, use the `select()` function to select the columns that are in the `review_id_table` in pgAdmin (as shown in the following image), and convert the `review_date` column to a date using the code snippet provided in the `Amazon_Reviews_ETL_starter_code.ipynb` file.

The final `review_id_table` DataFrame should look like this:

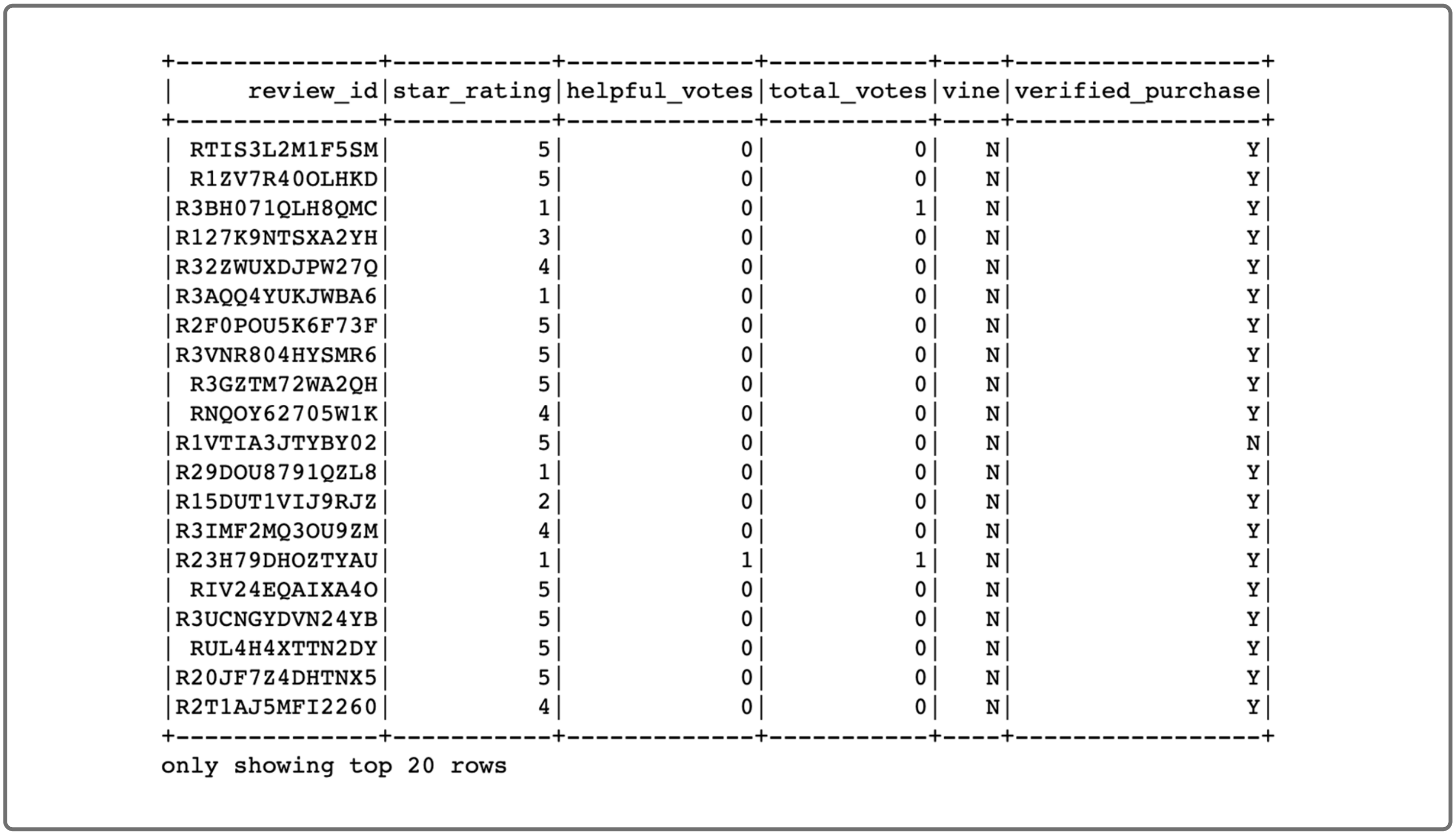


```
+--------------+-----------+----------+--------------+-----------+
|     review_id|customer_id|product_id|product_parent|review_date|
+--------------+-----------+----------+--------------+-----------+
|R28HBXXO1UEVJT|   22480053|0843952016|      34858117| 2012-05-03|
| RZKRFS2UUMFFU|   44244451|031088926X|     676347131| 2012-05-03|
|R2WAU9MD9K6JQA|   20357422|0615268102|     763837025| 2012-05-03|
|R36SCTKYTVPZPC|   13235208|1900869225|     785539232| 2012-05-03|
|R10BM6JUOJX27Q|   26301786|1565129938|      64646125| 2012-05-03|
| RCLZ5OKZNUSY4|   27780192|146854456X|     270349766| 2012-05-03|
|R1S65DJYEI89G4|   13041546|1118094514|     752141158| 2012-05-03|
|R3KQYBQOLYDETV|   51692331|0563521147|     729491316| 2012-05-03|
|R3QV8K7CSU8K2W|   23108524|0669444421|     261004015| 2012-05-03|
|R3W5A1WUGO5VQ0|   51692331|1897784457|     497876045| 2012-05-03|
|R20AQCY3FMBVN5|   49438248|0316738158|     691490916| 2012-05-03|
| R7KY8VL871MVL|   11818020|0738730440|     544176812| 2012-05-03|
| RHF5E4UOL5LQ3|   51692331|1902842286|     698916699| 2012-05-03|
|R1LMUDN5M9G6ZZ|   29446920|1465399577|     922463098| 2012-05-03|
| RNGA47KD4CEB8|   33284115|0061934704|     740765152| 2012-05-03|
|R33MYHP5RY1139|   44728718|1432729039|     116349266| 2012-05-03|
|R18VIM840CEFRP|   52534548|1621360075|     143884185| 2012-05-03|
| RQOZBXX7M0U6H|   37836302|097723732X|     106641033| 2012-05-03|
|R3SH84TAORQP2T|   38588903|0983945209|     377432437| 2012-05-03|
| RL1OHWOHPM7RO|   49148452|1419701630|     307676830| 2012-05-03|
+--------------+-----------+----------+--------------+-----------+
only showing top 20 rows
```

The vine_table DataFrame

To create the `vine_table`, use the `select()` function to select only the columns that are in the `vine_table` in pgAdmin (as shown in the following image).

The final `vine_table` DataFrame should look like this:



```
+--------------+-----------+-------------+-----------+----+-----------------+
|     review_id|star_rating|helpful_votes|total_votes|vine|verified_purchase|
+--------------+-----------+-------------+-----------+----+-----------------+
| RTIS3L2M1F5SM|          5|            0|          0|   N|                Y|
| R1ZV7R40OLHKD|          5|            0|          0|   N|                Y|
|R3BH071QLH8QMC|          1|            0|          1|   N|                Y|
|R127K9NTSXA2YH|          3|            0|          0|   N|                Y|
|R32ZWUXDJPW27Q|          4|            0|          0|   N|                Y|
|R3AQQ4YUKJWBA6|          1|            0|          0|   N|                Y|
|R2F0POU5K6F73F|          5|            0|          0|   N|                Y|
|R3VNR804HYSMR6|          5|            0|          0|   N|                Y|
| R3GZTM72WA2QH|          5|            0|          0|   N|                Y|
| RNQOY62705W1K|          4|            0|          0|   N|                Y|
|R1VTIA3JTYBY02|          5|            0|          0|   N|                N|
|R29DOU8791QZL8|          1|            0|          0|   N|                Y|
|R15DUT1VIJ9RJZ|          2|            0|          0|   N|                Y|
|R3IMF2MQ3OU9ZM|          4|            0|          0|   N|                Y|
|R23H79DHOZTYAU|          1|            1|          1|   N|                Y|
| RIV24EQAIXA4O|          5|            0|          0|   N|                Y|
|R3UCNGYDVN24YB|          5|            0|          0|   N|                Y|
| RUL4H4XTTN2DY|          5|            0|          0|   N|                Y|
|R20JF7Z4DHTNX5|          5|            0|          0|   N|                Y|
|R2T1AJ5MFI2260|          4|            0|          0|   N|                Y|
+--------------+-----------+-------------+-----------+----+-----------------+
only showing top 20 rows
```

Load the DataFrames into pgAdmin

1. Make the connection to your AWS RDS instance.
2. Load the DataFrames that correspond to tables in pgAdmin.
3. In pgAdmin, run a query to check that the tables have been populated.

IMPORTANT

Before uploading anything to GitHub be sure to remove all sensitive information such as passwords and connection strings. If you have accidentally done so already see this **link** **(https://docs.github.com/en/github/authenticating-to-github/removing-sensitive-data-from-a-repository)** for more information.

When you're done, export your `Amazon_Reviews_ETL` Google Colab Notebook as an `ipynb` file, and save it to your Amazon_Vine_Analysis GitHub repository.

NOTE

Uploading each DataFrame can take up to 10 minutes or longer, so it's a good idea to double-check your work before uploading. If you have problems uploading your work, you may have to shut down the pgAdmin server and restart. Alternatively, you may have to delete the tables and create them again, then re-run your `Amazon_Reviews_ETL` Google Colab Notebook.

IMPORTANT

> Be sure that you don’t leave your RDS instance up too long. Try to get all your work for Deliverable 1 done in one sitting, then shut down your instance. Please consult the AWS clean-up videos for more information about shutting down your RDS instance. You will not be graded on anything contained strictly in your RDS, so be sure to shut it down.

Deliverable 1 Requirements

You will earn a perfect score for Deliverable 1 by completing all requirements below:

- The `Amazon_Reviews_ETL.ipynb` file does the following:
  - An Amazon Review dataset is extracted as a DataFrame **(10 pt)**
  - The extracted dataset is transformed into four DataFrames with the correct columns **(20 pt)**
  - All four DataFrames are loaded into their respective tables in pgAdmin **(10 pt)**

Deliverable 2: Determine Bias of Vine Reviews (40 points)

Deliverable 2 Instructions

Using your knowledge of PySpark, Pandas, or SQL, you’ll determine if there is any bias towards reviews that were written as part of the Vine program. For this analysis, you'll determine if having a paid Vine review makes a difference in the percentage of 5-star reviews.

REWIND

For this deliverable, you’ve already done the following in this module using PySpark:

- **Lesson 16.4.2:** Use PySpark to read in a CSV file
- **Lesson 16.4.2:** Use PySpark methods and functions to get DataFrame information
- **Lesson 16.4.3:** Use PySpark functions to transform and filter data

Using either PySpark, Pandas, or SQL, follow the instructions below to complete Deliverable 2.

1. Filter the data and create a new DataFrame or table to retrieve all the rows where the `total_votes` count is equal to or greater than 20 to pick reviews that are more likely to be helpful and to avoid having division by zero errors later

on.

2. Filter the new DataFrame or table created in Step 1 and create a new DataFrame or table to retrieve all the rows where the number of `helpful_votes` divided by `total_votes` is equal to or greater than 50%.
   - If you use the SQL option below, you'll need to cast your columns as floats using `WHERE CAST(helpful_votes AS FLOAT)/CAST(total_votes AS FLOAT) >=0.5`.
3. Filter the DataFrame or table created in Step 2, and create a new DataFrame or table that retrieves all the rows where a review was written as part of the Vine program (paid), `vine == 'Y'`.
4. Repeat Step 3, but this time retrieve all the rows where the review was not part of the Vine program (unpaid), `vine == 'N'`.
5. Determine the total number of reviews, the number of 5-star reviews, and the percentage of 5-star reviews for the two types of review (paid vs unpaid).

NOTE

We recommend using either PySpark or Pandas to perform the analysis. Using SQL queries is more challenging, especially for Step 5 above, and is better suited for intermediate and experienced SQL programmers.

Using PySpark

1. Create a new Google Colab Notebook, and name it `Vine_Review_Analysis`.
2. Extract the dataset you used in Deliverable 1.
3. Recreate the `vine_table`, and perform your analysis using the steps above.
4. Export your `Vine_Review_Analysis` Google Colab Notebook as an `ipynb` file, and save it to your Amazon_Vine_Analysis GitHub repository.

Using Pandas

1. From pgAdmin, export the `vine_table` as a CSV file, and save it to your Amazon_Vine_Analysis GitHub repository.
2. Create a new Jupyter Notebook, and name it `Vine_Review_Analysis.ipynb`.
3. Read in the `vine_table.csv` file as a DataFrame, and perform your analysis using the steps above.
4. Save your `Vine_Review_Analysis.ipynb` file to your Amazon_Vine_Analysis GitHub repository.

Using SQL in pgAdmin

1. From your AWS database, export the `vine_table` as a CSV file and save it to your Amazon_Vine_Analysis GitHub repository.
2. In pgAdmin, create a new database that is not linked to your AWS RDS instance. This way, you don't have to keep incurring charges while connected to your AWS RDS instance.
3. Create a new SQL file and name it `Vine_Review_Analysis.sql`.

4. Recreate the `vine_table` using the schema provided in the `challenge_schema.sql` file.
5. Import the `vine_table.csv` file into the table, and perform your analysis using the steps above.
6. Save all your SQL queries to the `Vine_Review_Analysis.sql` file, then add it to your Amazon_Vine_Analysis GitHub repository.

Deliverable 2 Requirements

You will earn a perfect score for Deliverable 2 by completing all requirements below:

- The analysis does the following:
  - There is a DataFrame or table for the `vine_table` data using one of three methods above **(5 pt)**
  - The data is filtered to create a DataFrame or table where there are 20 or more total votes **(5 pt)**
  - The data is filtered to create a DataFrame or table where the percentage of `helpful_votes` is equal to or greater than 50% **(5 pt)**
  - The data is filtered to create a DataFrame or table where there is a Vine review **(5 pt)**
  - The data is filtered to create a DataFrame or table where there isn't a Vine review **(5 pt)**
  - The total number of reviews, the number of 5-star reviews, and the percentage 5-star reviews are calculated for all Vine and non-Vine reviews **(15 pt)**

Deliverable 3: A Written Report on the Analysis (20 points)

Deliverable 3 Instructions

For this part of the Challenge, you'll write a report that summarizes the analysis you performed in Deliverable 2.

The report should contain the following:

1. **Overview of the analysis:** Explain the purpose of this analysis.
2. **Results:** Using bulleted lists and images of DataFrames as support, address the following questions:
   - How many Vine reviews and non-Vine reviews were there?
   - How many Vine reviews were 5 stars? How many non-Vine reviews were 5 stars?
   - What percentage of Vine reviews were 5 stars? What percentage of non-Vine reviews were 5 stars?
3. **Summary:** In your summary, state if there is any positivity bias for reviews in the Vine program. Use the results of your analysis to support your statement. Then, provide one additional analysis that you could do with the dataset to support your statement.

Deliverable 3 Requirements

Structure, Organization, and Formatting (6 points)

The written analysis has the following structure, organization, and formatting:

- There is a title, and there are multiple sections **(2 pt)**
- Each section has a heading and subheading **(2 pt)**
- Links to images are working, and code is formatted and displayed correctly **(2 pt)**.

Analysis (14 points)

The written analysis has the following:

- Overview of the analysis of the Vine program:
  - The purpose of this analysis is well defined **(3 pt)**
- Results:
  - There is a bulleted list that addresses the three questions for unpaid and paid program reviews **(7 pt)**
- Summary:
  - The summary states whether or not there is bias, and the results support this statement **(2 pt)**
  - An additional analysis is recommended to support the statement **(2 pt)**

Submission

Once you're ready to submit, make sure to check your work against the rubric to ensure you are meeting the requirements for this Challenge one final time. It's easy to overlook items when you're in the zone!

As a reminder, the deliverables for this Challenge are as follows:

- Deliverable 1: Perform ETL on Amazon Product Reviews
- Deliverable 2: Determine Bias of Vine Reviews
- Deliverable 3: A Written Report on the Analysis (README.md)

Upload the following to your Amazon_Vine_Analysis GitHub repository:

- Your `Amazon_Reviews_ETL.ipynb` file.
- Your `Vine_Review_Analysis.ipynb` or `Vine_Review_Analysis.sql` file.
- An updated README.md that has your written analysis.

To submit your challenge assignment for grading in Bootcamp Spot, click Start Assignment, click the Website URL tab, then provide the URL of your Amazon_Vine_Analysis GitHub repository, and then click Submit. Comments are disabled for graded submissions in BootCampSpot. If you have questions about your feedback, please notify your instructional staff or the Student Success Manager. If you would like to resubmit your work for an improved grade, you can use the **Re-Submit Assignment** button to upload new links. You may resubmit up to 3 times for a total of 4 submissions.

IMPORTANT

Once you receive feedback on your Challenge, make any suggested updates or adjustments to your work. Then, add this week's Challenge to your professional portfolio.

NOTE

You are allowed to miss up to two Challenge assignments and still earn your certificate. If you complete all Challenge assignments, your lowest two grades will be dropped. If you wish to skip this assignment, click Next, and move on to the next Module.

Module-16 Rubric

Criteria	Ratings					Pts
Deliverable 1: Perform ETL on Amazon Product Reviews	**40 to >36.0 pts** **Demonstrating Proficiency** ✓An Amazon Review dataset is extracted as a DataFrame. ✓The extracted dataset is transformed into four DataFrames with the correct columns. ✓ALL FOUR DataFrames are loaded into their respective tables in pgAdmin.	**36 to >31.0 pts** **Approaching Proficiency** ✓An Amazon Review dataset is extracted as a DataFrame. ✓The extracted dataset is transformed into four DataFrames with the correct columns AND ONE df has incorrect column names. ✓THREE of the FOUR DataFrames are loaded into pgAdmin tables.	**31 to >27.0 pts** **Developing Proficiency** ✓An Amazon Review dataset is extracted as a DataFrame. ✓The extracted dataset is transformed into four DataFrames, but the products df has duplicate rows, OR ONE or more dfs has incorrect column names. ✓TWO of the FOUR DataFrames are loaded into pgAdmin tables.	**27 to >0.0 pts** **Emerging** ✓An Amazon Review dataset is extracted as a DataFrame. ✓The extracted dataset is transformed into four DataFrames, but the products df has duplicate rows, AND ONE or more dfs has incorrect column names. ✓TWO of the FOUR DataFrames are loaded into pgAdmin tables.	**0 pts** **Incomplete**	40 pts
Deliverable 2: Determine Bias of Vine Reviews	**40 to >36.0 pts** **Demonstrating Proficiency** ✓There is a DataFrame or table for the vine_table data. ✓The data is filtered to create a DataFrame or table where there are 20 or more total votes. ✓The data is filtered to create a DataFrame or table where the percentage of helpful_votes is equal to or greater than 50%. ✓The data is filtered to create a DataFrame or table where there is a Vine review. ✓The data is filtered to create a DataFrame or table where	**36 to >34.0 pts** **Approaching Proficiency** ✓There is a DataFrame or table for the vine_table data. ✓The data is filtered to create a DataFrame or table where there are 20 or more total votes. ✓The data is filtered to create a DataFrame or table where the percentage of helpful_votes is equal to or greater than 50%. ✓The data is filtered to create a DataFrame or table where there is a Vine review. ✓The data is filtered to create a DataFrame or table where	**34 to >31.0 pts** **Developing Proficiency** ✓There is a DataFrame or table for the vine_table data. ✓The data is filtered to create a DataFrame or table where there are 20 or more total votes. ✓The data is filtered to create a DataFrame or table where the percentage of helpful_votes is equal to or greater than 50%. ✓The data is filtered to create a DataFrame or table where there is a Vine review. ✓The data is filtered to create a DataFrame or table where	**31 to >0.0 pts** **Emerging** ✓There is a DataFrame or table for the vine_table data. ✓The data is filtered to create a DataFrame or table where there are 20 or more total votes. ✓The data is filtered to create a DataFrame or table where the percentage of helpful_votes is equal to or greater than 50%. ✓The data is filtered to create a DataFrame or table where there is a Vine review. ✓The data is filtered to create a DataFrame or	**0 pts** **Incomplete**	40 pts

Criteria	Ratings					Pts
	there isn't a Vine review. √The total number of votes, the number of 5 star reviews, and the percentage of 5 star reviews are calculated.	there isn't a Vine review. √The total number of votes and the number of 5 star reviews are calculated, but there is an error calculating the percentage of 5 star reviews.	there isn't a Vine review. √The total number of votes AND the number of 5 star reviews are calculated.	table where there isn't a Vine review. √Either the total number of votes OR the number of 5 star reviews are calculated.		
Deliverable 3: Structure, Organization, and Formatting	**6 to >5.0 pts** **Demonstrating Proficiency** The written analysis has ALL of the following: √There is a title, and there are multiple sections. √Each section has a heading and subheading. √There are images and references to code, and they are formatted and displayed correctly.	**5 to >4.0 pts** **Approaching Proficiency** The written analysis has ALL of the following: √There is a title, and there are multiple sections. √Each section has a heading and subheading. √There are images and references to code, and they are formatted and displayed correctly with one or two minor errors.	**4 to >3.0 pts** **Developing Proficiency** The written analysis has ALL of the following: √There is a title, and there are multiple sections. AND ONE of the following: √Each section may have a heading and subheading. √There are images and references to code, and they are formatted and displayed correctly with one or two minor errors.	**3 to >0.0 pts** **Emerging** The written analysis has ALL of the following: √There is a title. √There may be a subheading for a section. √There are no headings for each section, but there are three sections.	**0 pts** **Incomplete**	6 pts
Deliverable 3: Analysis	**14 to >12.0 pts** **Demonstrating Proficiency** √The purpose is well defined. √ALL THREE questions are addressed in the results. √The summary states whether or not there is bias and is supported by the results and there is a supporting additional recommendation.	**12 to >10.0 pts** **Approaching Proficiency** √The purpose is well defined. √ALL THREE questions are addressed in the results, but there is a misinterpretation. √The summary states whether or not there is bias and is supported by the results, but the additional recommendation doesn't support the statement.	**10 to >8.0 pts** **Developing Proficiency** √The purpose is well defined. √TWO of the THREE questions are addressed in the results. √The summary states whether or not there is bias and is supported by the results, but there's no additional recommendation.	**8 to >0.0 pts** **Emerging** √The purpose is well defined. √ONE of the THREE questions is addressed in the results. √The summary states whether or not there is bias, but it is not supported by the results.	**0 pts** **Incomplete**	14 pts
						Total Points: 100