CIS 4560 Term Project Lab Tutorial

Authors: Azim Pathan, Clever Lemus, Walter Giron, Yuner Paredes, Sipan Kouchian

Instructor: Jongwook Woo

Date: 11/20/22

Steam Reviews Data Analysis with Tableau

Objectives

In this hands-on lab, you will learn how to:

- Download the Dataset from Kaggle.com
- Upload the Dataset to Google Drive and Create a Shareable URL
- Find FileID from Shareable URL
- Edit wget Command Template with FileID
- Upload Dataset to Oracle Linux Server
- Upload Dataset to Hadoop Distributed File System (HDFS)
- Create Database and Tables Using Beeline
- Clean Data Using MapReduce and Download it to Linux
- Download Cleaned Dataset to Local PC
- Use Tableau Prep to Prepare the Data
- Visualize the Data in Tableau Desktop

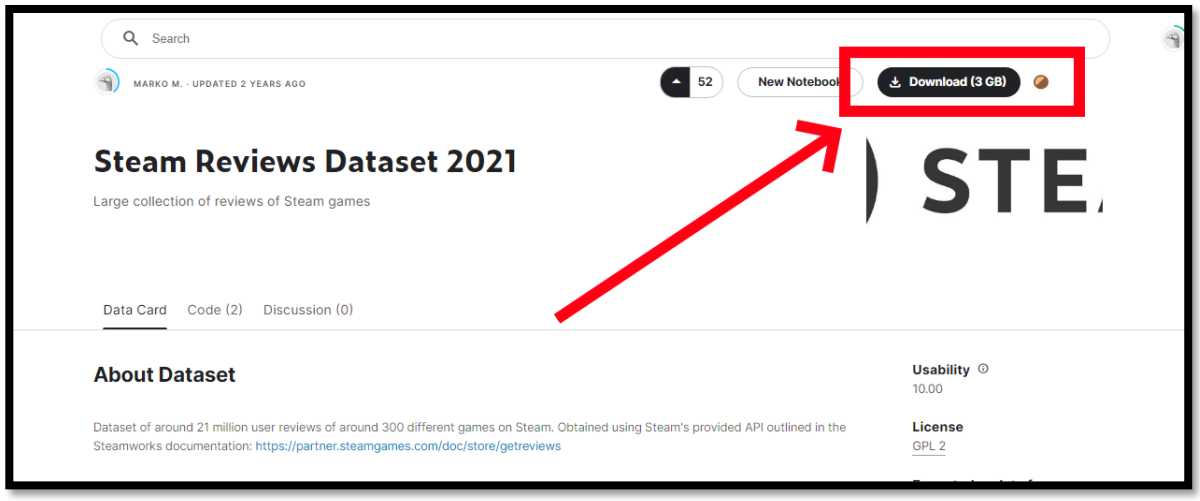
Platform Specs

- Oracle Linux (Red Hat) Big Data Server
- CPU Speed: 1995.309 MHz
- # Of CPU cores: 4
- # Of nodes: 5 Nodes – 2 Master and 3 Workers
- Total Memory Size: 58GB

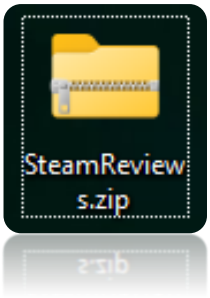
Step 1: Download the Dataset from Kaggle.com

In this step, we will download the dataset from Kaggle.com. This platform is used to publish and share datasets for data scientists to study, publish, analyze data, and more.
Note* this tutorial was created for Windows 10/11 but can also be completed using MacOS.

1. **Head** to the following URL and register for an account: https://www.kaggle.com/
2. **Download** the dataset: https://www.kaggle.com/datasets/najzeko/steam-reviews-2021
3. **Rename** the .zip file to **SteamReviews.zip** once the download is complete.



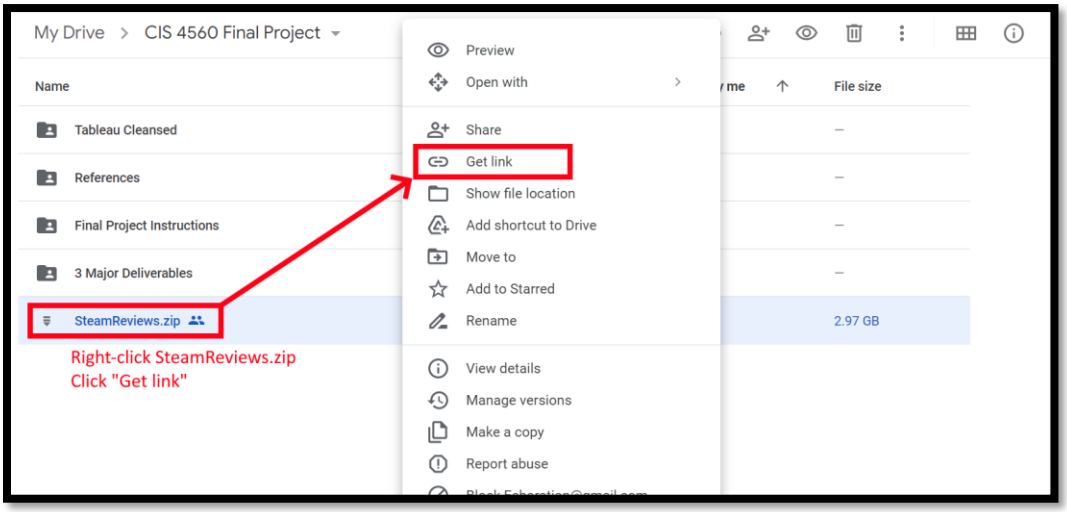
Download the Dataset & Rename it via your Desktop.



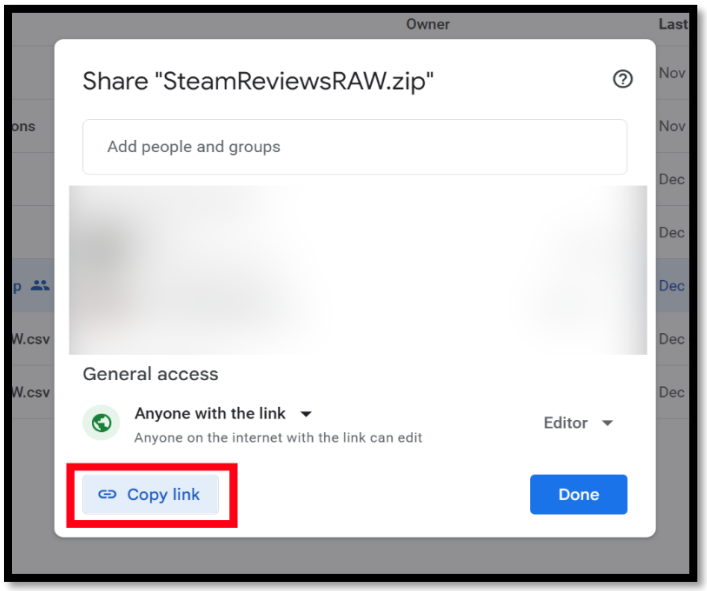
Step 2: Upload the Dataset to G-Drive & Create a Shareable URL

In this step we will upload your newly downloaded dataset (the SteamReviews.zip folder) to Google Drive and create a Shareable URL for use later.

1. **Create a Gmail Account** (unless you already own one) to access your Google Drive.
2. **Upload** the 3GB .zip to your Google Drive. This may take time depending on your upload speed.
3. **Create** a shareable URL to the 3GB document**.**
4. **Paste** this URL into **Notepad.exe** so that we can edit it later.



Create a Shareable URL, Copy Link, and Paste into Notepad.exe



Step 3: Find FileID from Shareable URL

In this step we will find the FileID of your Google Drive document. Linux requires a very specific wget command to download your file from Google Drive. To do this correctly, we will need your FileID which can be extracted from the Shareable URL.

1. **Copy** your FileID from your Shareable URL. The link is specifically placed between the **"/file/d/____yourFILEIDhere____/view?usp=share",** parts of the URL, as shown below.

Step 4: Edit wget Command Template with FileID

In this step we will edit the wget command template with your Google Drive's shareable URL FileID so that we can easily download the file to our Oracle Linux Server.

1. **Edit** the wget command below with the FileID you found in Step 3 above.
2. **Copy** the text below and **replace** both **FILEID** fields with your FileID accordingly, as shown below.

```
wget "https://docs.google.com/uc?export=download&confirm=$(wget --quiet --save-cookies
/tmp/cookies.txt --keep-session-cookies --no-check-certificate
'https://docs.google.com/uc?export=download&id=FILEID' -O- | sed -rn 's/.*confirm=([0-9A-Za-
z_]+).*/\1\n/p')&id=FILEID" -O SteamReviews.zip && rm -rf /tmp/cookies.txt
```

   - Can't copy it easily? Here's the Raw WGET Command Template if you need it.
   - If you're having trouble, try removing this part:
     - `&& rm -rf /tmp/cookies.txt`

3. **Be mindful** of accidentally adding a space or removing a single character. It will be unusable.
4. **Save** the edited command above for use in Step 5.

```
wget "https://docs.google.com/uc?export=download&confirm=$(wget --quiet --save-cookies /tmp/cookies.txt --keep-session-cookies --no-check-certificate 'https://docs.google.com/uc?export=download&id=15hobfe8Aww4Gs333QAF3KrB8tVyK6_RT' -O- | sed -rn 's/.*confirm=([0-9A-Za-z_]+).*/\1\n/p')&id=15hobfe8Aww4Gs333QAF3KrB8tVyK6_RT" -O SteamReviews.zip && rm -rf /tmp/cookies.txt
```

Your edited command should look like this. This is all we want in this step.

Step 5: Upload Dataset to Oracle Linux Server

In this step we will upload the .zip folder to our Oracle Linux Server so that we can then upload it to the Hadoop Distributed File System (HDFS).

1. **Open** your GitBash Terminal.
2. **Access** the Oracle Linux Server using:
    - `ssh yourUsername@144.24.14.145`
    - `password`
3. **Paste** your newly edited **wget** command from step 4 into the terminal.
4. **Wait** for the download to complete.
5. **In the Linux terminal, type the following commands in order:**
    - `ls`
        - to ensure that the file has been downloaded.
    - `Unzip SteamReviews.zip`
        - to unzip the .zip folder and uncover the .csv within it.
    - `ls`
        - to ensure that the .csv has been extracted successfully.
    - `du -h steam_reviews.csv`
        - to check the size of the file; it should be 7.7GB.
    - `cat steam_reviews.csv | head -5`
        - (Optional) This can be used to further ensure the integrity of the .csv file. It can help us check that the file was unzipped correctly and is not corrupted.

```
-bash-4.2$ wget "https://docs.google.com/uc?export=download&confirm=$(wget --quiet --save-cookies /tmp/cookies.txt --keep-sessio
n-cookies --no-check-certificate 'https://docs.google.com/uc?export=download&id=FILEID' -O- | sed -rn 's/.*confirm=([0-9A-Za-z_]
+).*/\1\n/p')&id=FILEID" -O SteamReviews.zip && rm -rf /tmp/cookies.txt
```

```
Saving to: 'SteamReviews.zip'

100%[=======================================================================================================>] 3,193,406,910  109MB/s   in 28s

2022-12-15 09:13:58 (109 MB/s) - 'SteamReviews.zip' saved [3193406910/3193406910]
```

```
-bash-4.2$ ls
SteamReviews.zip
-bash-4.2$ unzip SteamReviews.zip
Archive:  SteamReviews.zip
  inflating: steam_reviews.csv
-bash-4.2$ ls
steam_reviews.csv  SteamReviews.zip
-bash-4.2$ du -h steam_reviews.csv
7.7G    steam_reviews.csv
```

```
-bash-4.2$ cat steam_reviews.csv| head -5
,app_id,app_name,review_id,language,review,timestamp_created,timestamp_updated,r
ecommended,votes_helpful,votes_funny,weighted_vote_score,comment_count,steam_pur
chase,received_for_free,written_during_early_access,author.steamid,author.num_ga
mes_owned,author.num_reviews,author.playtime_forever,author.playtime_last_two_we
```

Optional Command: cat steam_reviews.csv | head -5

Step 6: Upload Dataset to Hadoop Distributed File System

In this step we will upload the dataset.csv file to HDFS. To do this, we will first make a directory in our HDFS, and upload the file to that folder.

1. **In the Linux terminal, type the following commands in order:**
    - `hdfs dfs -mkdir /user/yourName/SteamReviews`
        - to make the SteamReviews directory folder in HDFS.
    - `hdfs dfs -mkdir /user/yourName/tmp`
        - to make the tmp directory folder in HDFS. If it already exists, skip this step.
    - `hdfs dfs -ls`
        - to ensure that the file has been downloaded.
    - `hdfs dfs -put steam_reviews.csv /user/yourName/SteamReviews`
        - to put the steam_reviews.csv dataset into the newly created folder.
2. **Confirm** that your file has successfully been uploaded to HDFS by using the command:
    - `hdfs dfs -ls SteamReviews/`

```
-bash-4.2$ hdfs dfs -ls SteamReviews/
Found 1 items
-rw-r--r--   3 apathan3 hdfs 8171787229 2022-12-18 08:11 SteamReviews/steam_reviews.csv
```

Nice!

Step 7: Create Database and Tables Using Beeline

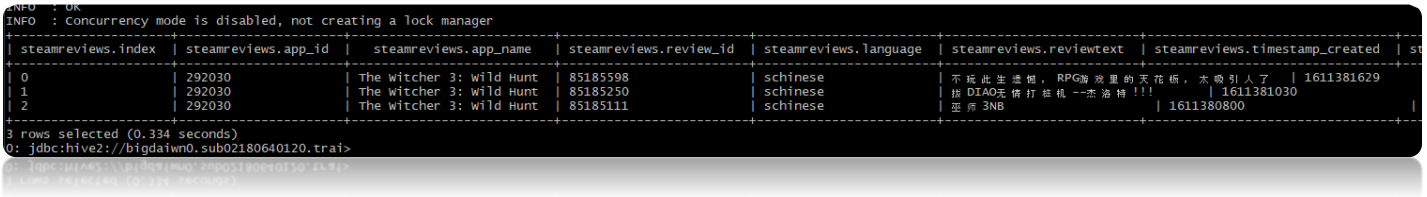
In this step we will start creating tables using Beeline. The purpose of creating tables is so that we can visualize the data we are working with. Doing this traditionally, say via Excel, would instantly crash the application because it is not capable of viewing over 1 million rows normally, let alone almost 50 million.

1. **In the Linux terminal, type the following commands in order:**
    - `beeline`
        - to open Beeline
    - `create database SteamReviewer;`
        - to create the SteamReviewer database.
    - `use SteamReviewer;`
        - to use the SteamReviewer database.
    - ```
      CREATE EXTERNAL TABLE SteamReviews(
      index INT,
      app_ID STRING,
      app_name STRING,
      review_ID STRING,
      language STRING,
      reviewText STRING,
      timestamp_created STRING,
      timestamp_updated STRING,
      recommended BOOLEAN,
      ```
      ```

```
votes_helpful INT,
votes_funny INT,
weight_vote_score DOUBLE,
comment_count INT,
steam_purchase BOOLEAN,
received_for_free BOOLEAN,
written_during_early_access BOOLEAN,
author_steamID STRING,
author_num_games_owned INT,
author_num_reviews INT,
author_playtime_forever BIGINT,
author_playtime_last_two_weeks BIGINT,
author_playtime_at_review BIGINT,
author_last_played STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ","
LOCATION "/user/yourName/SteamReviews"
TBLPROPERTIES ('skip.header.line.count' = '1');
```

- To create the table and view our data.
- Be sure to change "yourName" with your own username accordingly.

- `SELECT * FROM SteamReviews LIMIT 3;`
  - To ensure that the table has been created successfully.
- `SELECT * COUNT (*) FROM SteamReviews;`
  - To check how many rows are in our table (48.2mil)



We won't need columns such as review_id or app_id for our analysis, so let's remove them.

Step 8: Clean Data Using MapReduce and Download it to Linux

In this step we will clean the data with MapReduce. MapReduce is useful because it uses the incredible power of "Parallel Processing" to process massive amounts of data via the nodes within the cluster.

1. **In Beeline, type the following commands in order:**
   - `CREATE VIEW steamreview_reduced AS SELECT index, app_id, app_name, language, reviewText, timestamp_created FROM SteamReviews;`
     - To create a view with the columns we might want. Create as many as you wish with the columns you want for your analysis. This is fine for this example.
   - `SELECT * FROM steamreview_reduced limit 10;`
     - To create the table and view our data.
   - `INSERT OVERWRITE DIRECTORY '/user/yourName/tmp/'`
     `ROW FORMAT DELIMITED FIELDS TERMINATED BY ','`
     `SELECT * FROM steamreview_reduced;`
2. **Head back** to the Linux Terminal.
3. **Confirm** that the file was created successfully by typing the following command:

- `hdfs dfs -ls tmp/`

```
-bash-4.2$ hdfs dfs -ls tmp/
Found 1 items
-rw-r--r--   3 apathan3 hdfs 4141091008 2022-12-18 09:19 tmp/000000_0
```

4. **In the Linux Terminal, type the following commands in order:**
   - `hdfs dfs -get /user/yourName/tmp/000000_0`
     - To get the file, "000000_0" which is our cleaned dataset to the Linux Server.
   - `ls`
   - `du -h 000000_0`
     - To ensure that the data we have downloaded is indeed reduced.
     - From the screenshot below, you can see that the file was reduced from 7.7GB to 3.9GB. The MapReduce reduction was a success!

```
-bash-4.2$ du -h 000000_0
3.9G    000000_0
-bash-4.2$ du -h steam_reviews.csv
7.7G    steam_reviews.csv
```

Step 9: Download Cleaned Dataset to Local PC

In this step, we will download the dataset back to our local computer. Now that we have cleaned the data, it will be much easier to clean further via Tableau Prep and visualize it using Tableau Desktop.

1. **Open** GitBash Terminal and type: `cd ~/Desktop`
2. **Type:** `scp yourName@144.24.14.145:/home/yourName/000000_0 .`
3. **Rename** the file to steam_reviews.csv

```
azimn@Ash_Rain MINGW64 /
$ cd ~/Desktop

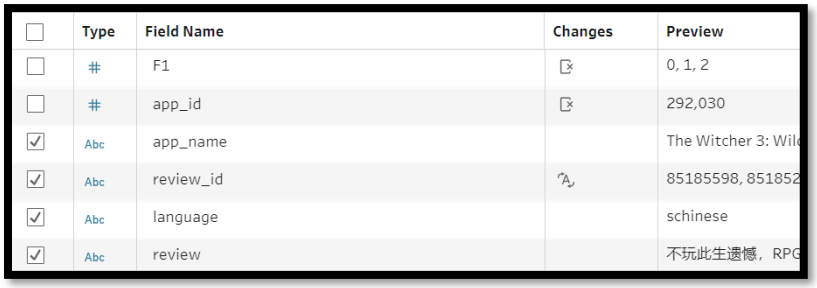
azimn@Ash_Rain MINGW64 ~/Desktop
$ scp apathan3@144.24.14.145:/home/apathan3/000000_0 .
apathan3@144.24.14.145's password:
000000_0                                      28% 1128MB  36.6MB/s   01:17 ETA
```

Step 10: Use Tableau Prep to Prepare the Data

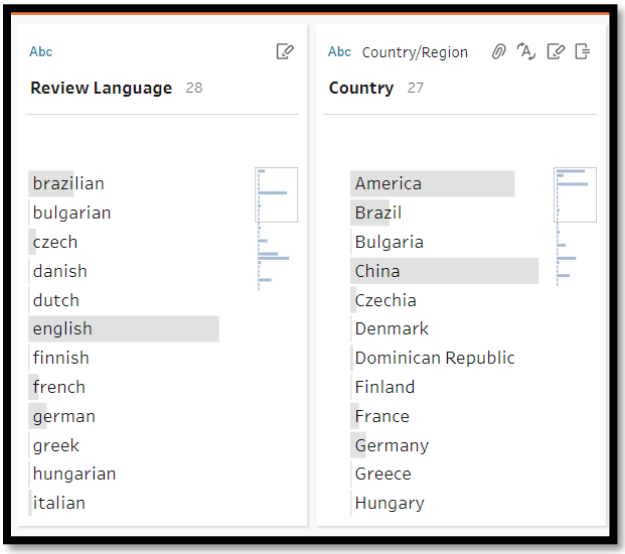
In this step we will launch Tableau Prep to prepare the data and clean it further. There is still quite a lot of unimportant data. Tableau Prep is powerful because it can help us create specific filters for the data and parse exactly what we need for each individual analysis. For example, we will show you how to create an output of ONLY the Geospatial locations for geospatial analysis, or ONLY the review text for Natural Language Processing. Or perhaps a multitude of these if your machine can handle it.

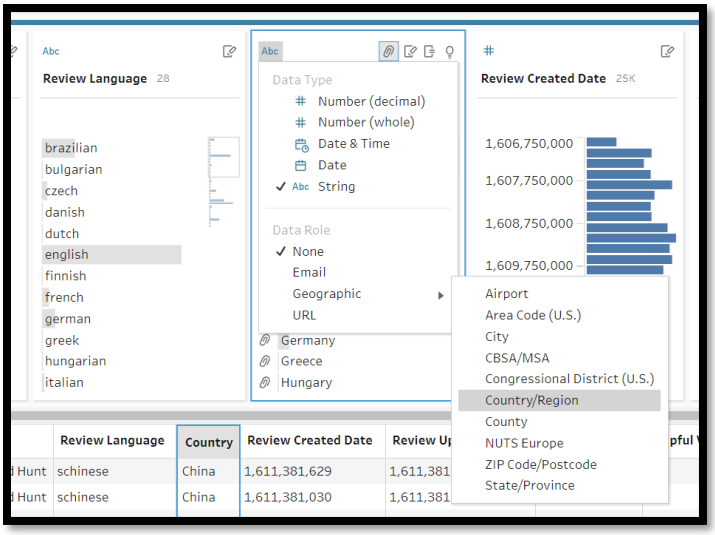
This is a critical step that involves filtering the data to specifically include points that are important to the analysis you are trying to perform. Even with so much cleaning and preparation, the data is still very dense. It could take a very long time to simply open a file like this in Tableau Desktop. Therefore, our final dataset prep requires us to know what kind of analysis we are doing and process exactly those parts of the set to work on it. This is the only way to ensure the integrity of the data.

1. **Open** Tableau Prep and connect a new workflow with *steam_reviews.csv*
2. **Apply** the following data types to your data and uncheck app_id or F1 if you have them as shown below.

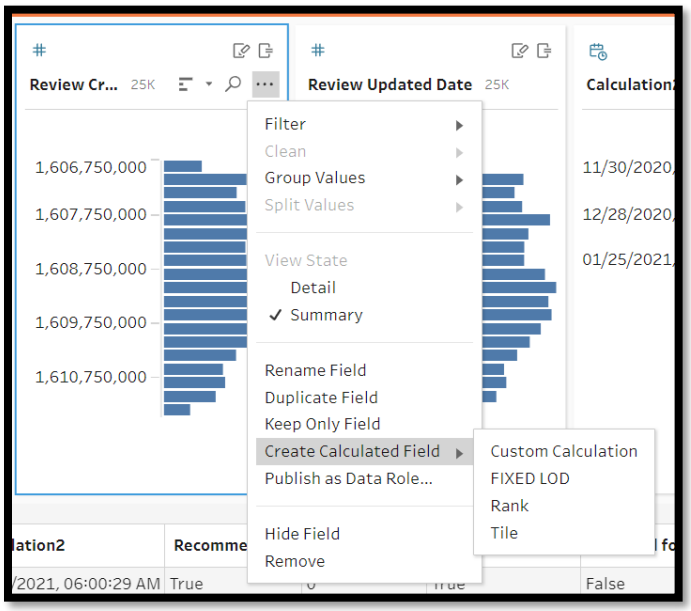


3. **Create** the “derived” Country variable. You can do this by duplicating the Review Language column and manually adding each respective country as shown below. Change to Country/Region.





4. **Convert** the UNIX Timestamp Dates to Date/Time:
    - **Right-click** the created date and select “Custom Calculation”



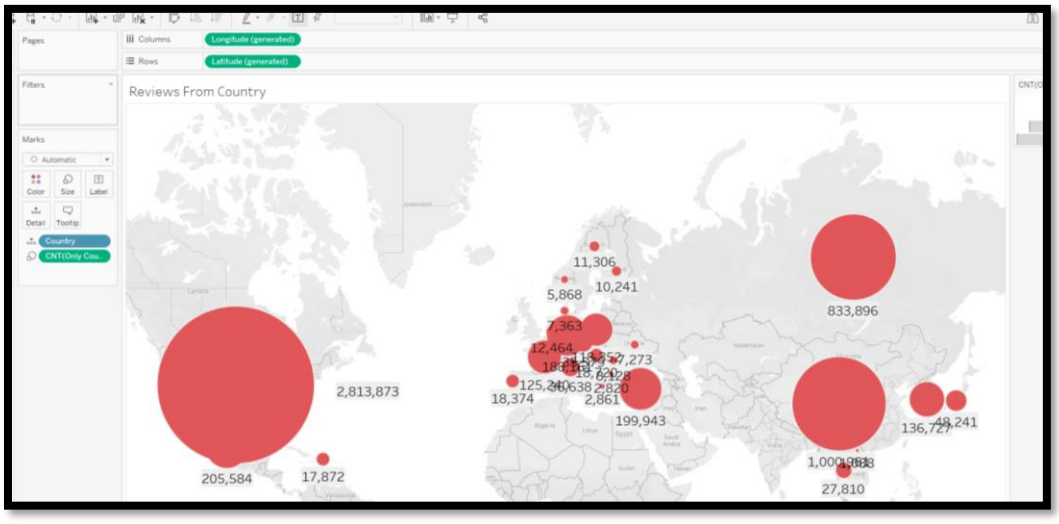
5. **Copy + Paste** the following code as shown below into the new field. Change the column name accordingly.
    - `DATEADD('second', int([Review Updated UNIX Date]), #1970-01-01#)`

6. **Clean and prepare** the data to your liking. Here is an overview of our “Analysis Filters”

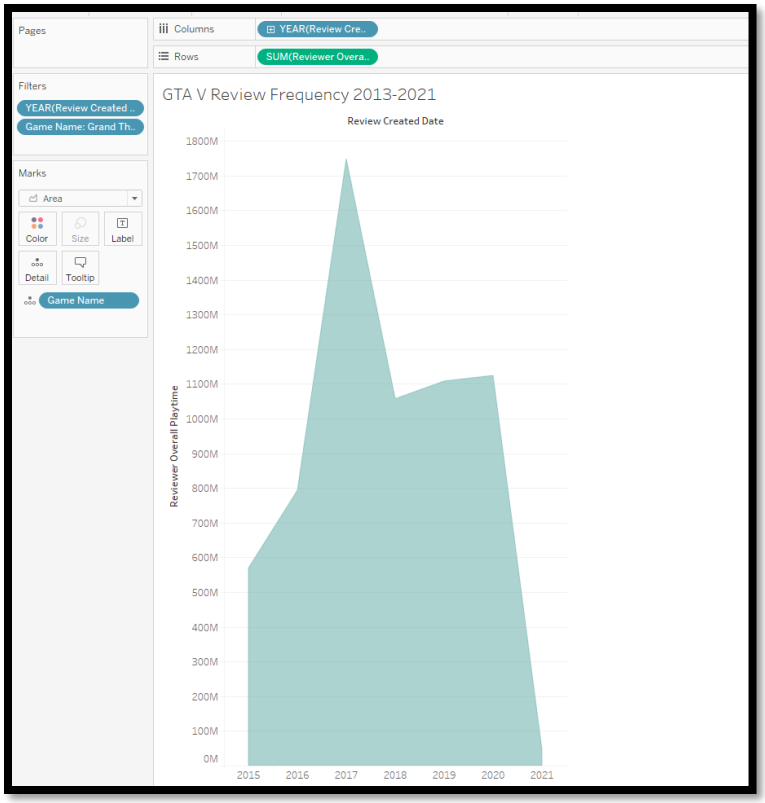
Step 11: Visualize the Data in Tableau Desktop

In this final step, we will visualize the data via Tableau Desktop. After cleaning and preparing the data, we are now able to perform incredible visualizations and representations for the data in many ways. This can help us come up with new solutions and strategies for our business and meet goals.

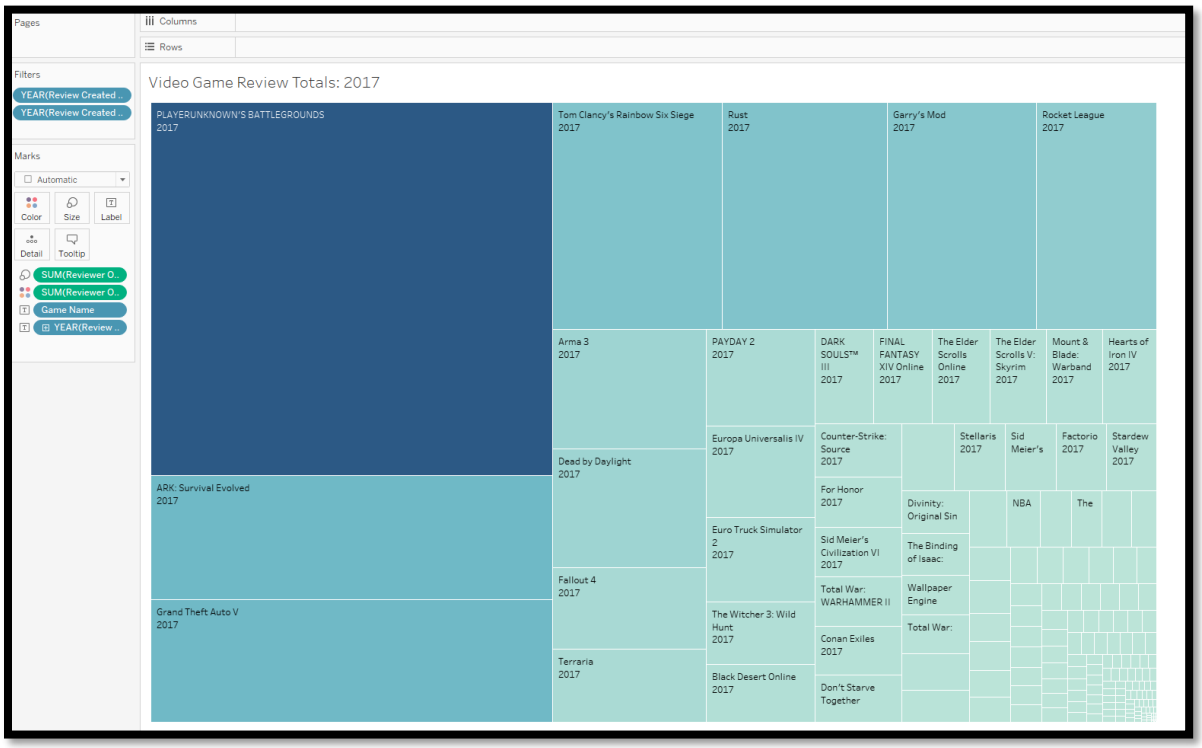
1. **Open** Tableau Desktop.
2. **Create** a new workbook and name it “Geo-spatial analysis”.
3. **Add** the following to Columns: Longitude
4. **Add** the following to Rows: Latitude
5. **Add** Country name by double clicking it.



6. **Create** a new sheet and rename it to “time-series analysis”
7. **Add** Country and Data/Time to Rows/Columns respectively.
8. **Filter** the columns to years 2013-2021, and select GTA V as the game name



9. **Create** a 3rd sheet, name it Most Reviewed Video Game 2017
10. **Add** Game name and Review Date



References

- URL of Data Source, https://www.kaggle.com/datasets/najzeko/steam-reviews-2021
- URL of your Github, https://github.com/evozims/CIS4560
- URL of References,
    - https://help.tableau.com/current/prep/en-us/prep_get_started.htm
    - https://help.tableau.com/current/guides/get-started-tutorial/en-us/get-started-tutorial-home.htm
    - https://medium.com/@acpanjan/download-google-drive-files-using-wget-3c2c025a8b99
    - https://www.kaggle.com/code/laseroverrider/k-means
    - https://www.kaggle.com/code/lorenzosantangelo/feature-engineering-k-means
    - https://partner.steamgames.com/doc/store/getreviews