COMPSCI 2034b / DIGIHUM 2144b

Data Analytics: Principles and Tools

Assignment #2

Youtube Comment Analytics

Posted:	March 4th 2024
Due:	March 18th 2023 11:55PM
Total:	100 Points

Learning Outcomes

By completing this assignment, you will gain and demonstrate skills relating to:

- Data Munging.
- Using Regular Expressions.
- Textual Analysis.
- VBA String Functions.
- Using Nested Loops.

Instructions

In this assignment, you will follow the directions given in each task in this document precisely and produce a Word file named *userid_assign2.docx*, an Excel Macro Enabled Workbook named *userid_assign2.xlsm* (where *userid* is your UWO user id), and a Tab-separated Values (.tsv) file named *vidname.tsv* (where *vidname* is the name of the dataset you used). You must assume that the data in your sheet can change (i.e. you may not hardcode your answers). Each step must be followed precisely including the file naming convention given in the Submission Section.

It is expected that you will **document your code using comments** in sufficient detail that the purpose and function of each line is clear to the TA marking your assignment. You should have at least one comment before each VBA function documenting what the function does, what arguments it takes and what value it returns. You should also have comments inside your functions documenting any complex lines of code.

You will be assessed on the following:

- Using the correct files.
- Properly cleaning and importing the comments into Excel.
- Your Excel formulas and operations.
- Your VBA code.
- Completion of each task correctly.
- Coding each function as described.
- Using the given function headers without modification.
- **Commenting your code in sufficient detail.**
- Indenting your code appropriately.
- Assignment submission via OWL.

Problem Description

In this assignment, you will pre-process, analyze, and present data relating to individual's current opinion on a given Youtube video. You will be given the option to select one Youtube video to perform your analysis on from a small set of preexisting comment datasets.

Regardless of the video you choose, the dataset will contain between 1,000 to 3,500 comments made on the video.

You will act as a data analyst and perform a textual analysis on this data to attempt to derive some meaning. In this case, the current sentiment or opinion Youtube users have of your chosen video. This sort of sentiment analysis can be valuable for business and organizations to judge the general public's response to recent announcements or product launches.

To derive this sentiment you will be required to perform the following tasks (described in detail in the subsequent sections of this document):

- **Obtain Data:** Obtain the dataset.
- **Data Munging**: Clean the dataset using regular expression.
- **Importing:** Import the data into Excel, sort it and use Excel's built in remove duplicate tool.
- **Remove Duplicates:** Use VBA to create our own duplicate removal tool to further clean the data.
- **Calculate Sentiment:** Use VBA to calculate the sentiment of each comment.
- **Analysis:** Use Excel formulas to analyze the result and present your findings.

Tasks

Task 0: Obtain The Dataset

Disclaimer

These datasets contain real comments from videos that have been posted to Youtube. They are unfiltered and can contain offensive and/or inappropriate language. The inclusion of a particular comment is in no way an endorsement of the comment's contents.

You will select **one and only one video** described in this section to perform your analysis on. Once your have decided on the video you would like to use in your analysis, download the corresponding dataset from OWL (should be attached to the assignment on the Assignments tab) and continue with Task 1.

Video 1: Barbie Trailer

Video Title: Barbie — Main Trailer
URL: https://www.youtube.com/watch?v=pBk4NYhWNMM
Dataset Name: Barbie.txt
Background: The Barbie movie was released in theatres on July 21st, 2023 and had significant box office success. But what did Youtuber's think of the trailer, and did this view change depending on the country of the commenter?

Video 2: Dune Part 2 Trailer

Video Title: Dune: Part Two — Official Trailer
URL: https://www.youtube.com/watch?v=Way9Dexny3w
Dataset Name: DunePart2.txt
Background: The highly anticipated movie Dune: Part 2 is releasing in theatres on March 15th, 2024. How hyped (excited) are Youtubers for it's upcoming release?

Video 3: Youtube Rewind 2019

Video Title: YouTube Rewind 2019: For the Record
URL: https://www.youtube.com/watch?v=2lAe1cqCOXo
Dataset Name: Rewind2019.txt
Background: From 2010 to 2019 each year YouTube would produce a video highlighting the viral videos, events, trends, and music of the last year. This trend ended in 2019. Could youtuber's reactions to the last rewind video give us some clues into why it was discontinued?

Video 4: Google Stadia

Video Title: Stadia GDC 2019 Gaming Announcement
URL: https://www.youtube.com/watch?v=nUih5C5rOrA
Dataset Name: Stadia.txt
Background: Google Stadia was a failed cloud gaming service offered by Google from 2019 to 2023. Stadia users were ultimately unimpressed with the the quality of the service, but what did Youtubers think about the launch announcement at the time?

Video 5: CyberTruck

Video Title: CYBERTRUCK
URL: https://www.youtube.com/watch?v=DsonSEllPmU
Dataset Name: CyberTruck.txt
Background: Recently Tesla launched the Tesla Cybertruck, a fully electric full-size pickup truck. Tesla has faced some controversy about the safety and build quality of the truck, but what do Youtuber's who commented on this Cyberturck advertisement think?

Video 6: Apple Vision Pro

Video Title: Using Apple Vision Pro: What It's Actually Like!
URL: https://www.youtube.com/watch?v=dtp6b76pMak
Dataset Name: AppleVisionPro.txt
Background: Apple recently launched the Apple Vision Pro, a high end mixed-reality headset. Unfortunately, Apple does not allow comments on their Apple Vision Pro videos, so we will instead perform our analysis on a video by the Youtuber Marques Brownlee giving his initial impressions. What do Marques fans think of the Vision Pro?

Video 7: Cyberpunk 2077

Video Title: Cyberpunk 2077 — Our Commitment to Quality
URL: https://www.youtube.com/watch?v=O3V4UBZmC9o
Dataset Name: CyberpunkApology.txt
Background: The video game Cyberpunk 2077 developed by CD Projekt Red was released on December 10th, 2020. Technical issues and a lack of promised content and features lead to significant criticism from gamers, leading to CD Projekt issuing an apology video and reconfirming their commitment to quality. But how did Youtuber's react to this apology?

Video 8: Avatar: The Last Airbender

Video Title: Avatar: The Last Airbender — Official Teaser — Netflix
URL: https://www.youtube.com/watch?v=waJKJW_XU9O
Dataset Name: LastAirbender.txt
Background: On February 22, 2024 Netflix released the first episodes of their live action adaptation of Avatar: The Last Airbender based on a beloved 2005 animated television series of the same name. Fans had significant concerns after previous attempts at live adaptations had been critically condemned. Did the trailer for the new series manage to win over fans?

Video 9: GameStop Hearing

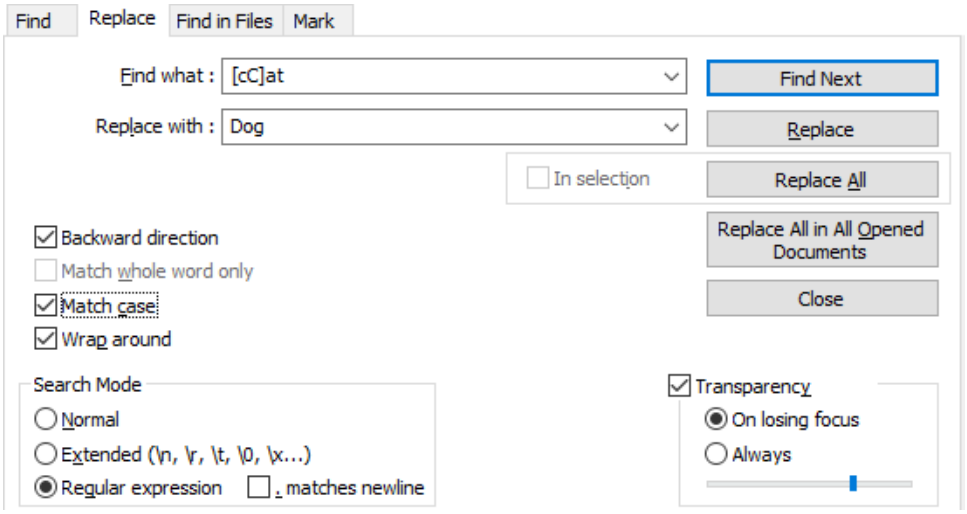
Video Title: Keith Gill delivers his testimony at GameStop hearing: 'I like the stock'
URL: https://www.youtube.com/watch?v=ukXQGBpXaVM
Dataset Name: GameStopHearing.txt
Background: During the 2021 GameStop short squeeze hearing, Keith Gill, also known as "Roaring Kitty", defended his role in the stock market frenzy that caused GameStop's stock price to surge. Gill, a retail investor and YouTube personality, argued that he was simply sharing his investment thesis on the stock and did not manipulate the market. But what did Youtubers think of Gill and his comments at the hearing?

Task 1: Data Munging

For each Step in this Task (except Steps 1.1 and 1.8), record the regular expression pattern you used for the **find field** in Notepad++[1] and the pattern (if any) you used for the **replace field**. You will be required to submit a Word document of your answers when submitting your assignment.

For example, if you used the following fields for doing a substitution:

[1]You may use a program other than Notepad++ if you are using a macOS based computer, but keep in mind that the regular expression syntax might be different in these programs. **The syntax you submit must work in Notepad++**, if you have any doubts about this, test your regular expressions on the Windows based lab computers on campus.



you should record the following in your Word document:

Find: [cC]at
Replace: Dog
Match Case: Yes
Mode: Regular Expression

If you do not include **Match Case** or **Mode**, it will be assumed that **Match Case** is on (Yes) and that the **Mode** is Regular Expression. If you wish to replace the text with nothing, simply put:

Replace:

That is, "Replace:" followed by no text.

If your regular expression includes a space that might be hard to see (e.g. at the end or start of the pattern or multiple spaces in a row), make sure it is clear to the reader that the space is there. For example, you might use the ␣ character[2] to denote a space in your pattern. If you do this leave a note stating something to the effect of ␣ = space so that your intent is clear to the reader.

Step 1.1: Understanding the Data

Open your dataset in Notepad++ (or equivalent program). This file should contain at least 1,500 comments (likely more) about a publicly traded company as described in the problem description. Unfortunately for us, the format of this file is a bit unusual and can not be imported into Excel directly.

Each comment in this file is separated by a row of dashes (e.g. ----------) and contains both the comment text as well as metadata about the comment and it's author.

[2]You can input a ␣ in programs like Word by typing U+2423 and then pressing ALT and X.

Each data value is listed as a value name (e.g. *"Comment by"* for user name of the comment's author) followed by a colon (*:*), a space character, then a data value (e.g. *@joebloggs5*), and finally a line feed character (\n). That is to say that each data value is prefixed with the name of that value. For example, the number of likes the comment has is prefixed with the text *"Likes: "* and the user's country with the text *"Country: "*. If a value could not be determined, it will be listed as *"None"*.

The following table describes each data value in the file:

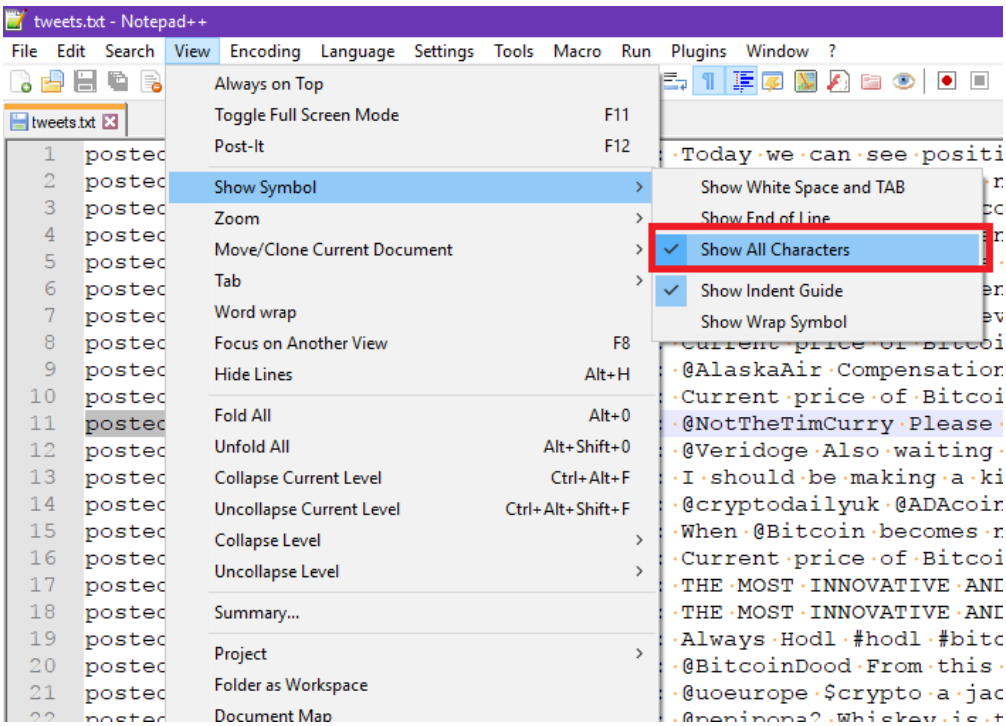
Data Value Name	**Description**
Comment by:	The Youtube handle of the user who posted this comment. The handle is always prefixed with a @ such as *@jbloggs5* or *@dr.servos*. Handle's are between 3-30 characters, made up of alphanumeric characters, and may also include: underscores (_), hyphens (-), and periods (.).
Comment Text:	The text of the comment. May contain special characters, emojis, and non-english characters. Line breaks have been removed from the comments as well as tab characters and colons (*:*).
Likes:	The number of likes this comment has received.
Replies:	The number of replies made directly to this comment.
Country:	The country the poster of this comment resides in, but only if they listed it in their profile. If the country could not be determined the value will be *"None"*. The country is represented by a two letter ISO 3166 country code.
Comment Published on:	The date and time the comment was posted. The format of this date is *day/month/year hour:minute:second-timezone AM/PM* for example *28/02/2024 08:52:28-0500 PM*. The day, month, hour, minute, and second values will always be two digits and the timezone always four digits with a leading plus or minus sign.
Comment Last Updated:	When the comment was last updated or edited. If the comment was never edited, this value will be the same as *"Comment Published on"*. This value uses the same date format as *"Comment Published on"*.
Commenter's Channel URL:	The URL for the commenter's channel. Should be a standard URL starting with *"http://www.youtube.com/"*.
Commenter's Channel Title:	The title/name of the commenter's channel. May contain spaces and special characters but not line breaks or a colon (*:*).
Commenter's Channel Publish Date:	The date the commenter's channel was published. This can give us an idea as to how long this commenter has been a Youtuber. This value uses the same date format as *"Comment Published on"*.

The data values will always be in the same order as shown in the above table and the data value name is followed by a colon (:) and then a single space before the data value. For example:

```
Comment by: @jbloggs5\n
```

where Comment by is the name of the data value, a ␣ is a single space, and \n is the line feed character that starts a new line in the file. The data value name is followed by a colon and single space (:␣) and then the data value, @jbloggs5.

The best way to fully understand this format is to take a look at the data yourself using a program like Notepad++. If you are using Notepad++ you may display the invisible tab and space characters via the View menu:



This will display tab characters as orange arrows, space characters as orange dots, and end of line characters as a LF.

It may also be helpful to ensure that "Word wrap" is disabled in NotePad++ to ensure that the comment text is displayed on one line.

Note that these datasets are formatted with Unix style end of line characters. That is to say end line ends with one \n character (shown in Notepad++ as a LF when All Symbols are shown).

It is highly recommended to save your work to a new file after each step. For example, after step 1.2, save your text file as *vidname_step1_2.txt* where *vidname* is the name of your dataset. You do not need to submit each of these files (see the last section of this document for what to submit) but it is important to save different copies of the dataset as we go so we can easily come back to this step if we make an error later. Also make sure you are recording all of the regular expressions and settings you used in your word document as described at the start of this task.

Step 1.2: Remove @ From the User Handles

The @ character in front of the user handle's in the *Comment by:* value are not very useful to us. Remove all of the @ characters from the file but only if they occur in the value listed after *Comment by:*. Do **not** remove them from the *Channel URL*, *Comment Text*, or anywhere else in the file.

For example lines such as:

```
Comment␣by:␣@jbloggs5
Comment␣by:␣@dr.servos
Comment␣by:␣@d-serv␣os.5
```

should all have the @ removed and no other changes:

```
Comment␣by:␣jbloggs5
Comment␣by:␣dr.servos
Comment␣by:␣d-serv␣os.5
```

where the ␣ character represents a normal space.

Hint 1: Remember the @ we want to remove always happens after the text "Comment by: " at the start of a line.

Hint 2: If you have to match text you don't want to remove, such as "Comment by: ", you can always put it back in the replace field.

Step 1.3 Reformat the Dates

The dates in the dataset are in the format *day/month/year hour:minute:second-timezone AM/PM* for example *28/02/2024 08:52:28-0500 PM*. The day, month, hour, minute, and second values will always be two digits and the timezone always four digits with a leading plus or minus sign. For example, if the timezone was +0400 then the date might look like this: *28/02/2024 08:52:28+0400 PM*.

We would like to transform these dates to follow the Canadian standard and be easier to important into Excel. Write a regular expression find and replace pattern to convert the dates to the format *year-month-day hour:minute:second AM/PM* such as *2024-02-28 08:52:28 PM* for *28/02/2024 08:52:28-0500 PM*. The timezone should be removed completely.

For example, the following dates:

```
26/02/2024 07:58:39-0500 AM
03/08/2023 08:25:05-0400 AM
02/12/2023 03:20:19+0200 PM
```

should become:

```
2024-02-26 07:58:39 AM
2023-08-03 08:25:05 AM
2023-12-02 03:20:19 PM
```

For full marks, this must be done in one Find/Replace for all dates in the file and not impact any characters in the file that are not part of a date.

Hint 3: This will require the use of groupings in your find pattern and recalling their values in the replace field.

Step 1.4 Remove Extra Characters at the Start of Comment Text

In some cases the comment text starts with a character that could cause issues once we important the data into Excel. We need to remove any -, =, ?, !, ", ', or space (i.e. the minus sign, equals sign, question mark, double quote, single quote, or the space character) **but only at the start** of the comment text.

Create a regular expression based Find and Replace pattern to remove any number of -, =, ?, !, ", ', or space if they occur at the start of the comment's text. You should only remove these characters if they are the first characters of a comment's text and not everywhere in the file. After the replacement there should still be a single space between "Comment Text:" and the first letter of the comment's text.

For example, the following lines:

```
Comment Text:  !!!!I LOVE THIS MOVIE!!!!
Comment Text:  ␣␣␣␣␣␣I␣␣think␣␣␣there␣is␣␣something␣wrong␣␣␣␣␣with␣my␣␣␣␣s␣pa␣ce␣b␣ar.
Comment Text:  -10 is my rating.
Comment Text:  -=?!"'␣␣␣␣hello world
```

should become:

```
Comment Text:  I LOVE THIS MOVIE!!!!
Comment Text:  I␣␣think␣␣␣there␣is␣␣something␣wrong␣␣␣␣␣with␣my␣␣␣␣s␣pa␣ce␣b␣ar.
Comment Text:  10 is my rating.
Comment Text:  hello world
```

Hint 4: It's possible there are no or few lines like this in your file. You must still write a find/replace pattern for this step even if your file does not include comments like this.

Step 1.5 Cleaning up the Spaces

Some of the comments text in our dataset may have multiple spaces in a row such as "Hello␣␣␣␣␣World!" where each ␣ represents a normal space. Create a regular expression based Find and Replace pattern to remove any instances of two or more spaces in a row in the dataset and replace them with a single space. Make sure you don't simply remove all spaces from the file, we only want to remove occurrences of two or more spaces.

For example the line:

```
Comment Text:  ␣␣␣␣␣I␣␣think␣␣␣there␣is␣␣something␣wrong␣␣␣␣␣with␣my␣␣␣␣s␣pa␣ce␣b␣ar.
```

should become:

```
Comment Text:  I␣think␣there␣is␣something␣wrong␣with␣my␣s␣pa␣ce␣b␣ar.
```

Step 1.6 Anonymize the User Data

As we will not be using the user handle's or their channel information we should anonymize them to protect the user's identity. Alter each user handle in the dataset such that only the first and last character of the handle is shown, separated by exactly four asterisks (*). For example, the handle `jbloggs5` would become `j****5` and `DanS` would become `D****S`.

Only handle listed directly after "Comment by:" should be changed. Handle's listed in the Channel URL, Channel Title, Comment Text, or elsewhere in the file should not be altered. Provide a regular expression based Find and Replace pattern to perform this substitution.

For example the lines:

```
Comment by:  jbloggs5
Comment by:  DanS
Comment by:  dr.servos
Comment by:  dgs
```

should become:

```
Comment by:  j****5
Comment by:  D****S
Comment by:  d****s
Comment by:  d****s
```

Also write one more regular expression based Find and Replace patterns to remove the lines starting with "Commenter's Channel URL:" or "Commenter's Channel Title:" as we will

not need these values in our analysis. You should delete only these lines and account for both types in one Find and Replace pattern.

Hint 5: For this step, you should have two Find and Replace patterns. One to anonymize the user handle's and the other to delete the "Commenter's Channel URL:" and "Commenter's Channel Title:" lines.

Hint 6: You will need groupings in your regular expression and replace fields to anonymize the user handle's.

Hint 7: Using the OR operator (|) maybe helpful for deleting lines that start with "Commenter's Channel URL:" OR "Commenter's Channel Title:".

Step 1.7 Transform into a TSV File

Before we can import the dataset into Excel we need to transform it into a TSV (Tab-Separated Values) file. To do this, we need to remove all of the data value names, the colon (:), the single space they are followed by, and the line breaks. The goal is to get all values for the same comment on one line and separated only by tabs.

1.7.1 Remove the Data Value Names

To accomplish this we will have to write a few regular expression based Find and Replace patterns to transform our data. The first pattern we should create is one to remove the data value names such as "Comment by:", "Likes:", "Country:" and so on from the dataset. You should write this pattern without using the OR operator and such that it works for any possible data value name.

For example, the lines:

```
Comment by: u****n
Comment Text: I liked the dancing hippo
Likes: 0
Replies: 0
Country: None
Comment Published on: 2024-02-28 07:33:35 PM
Comment Last Updated: 2024-02-28 07:33:35 PM
Commenter's Channel Publsh Date: 2023-12-24 12:04:43 AM
```

should become:

```
u****n
I liked the dancing hippo
0
0
None
2024-02-28 07:33:35 PM
```

```
2024-02-28 07:33:35 PM
2023-12-24 12:04:43 AM
```

Make sure there is no extra space character left at the start of each line.

1.7.2 Remove the Line Breaks

After the data value names are removed, we need to get all of the data for each comment onto one line. We can do this by replacing the line feed character (\n) with a tab character (\t). This will make everything in the dataset on one line, but don't worry we will fix it in the next step.

Write a regular expression based Find and Replace patterns to replace all line feed characters (\n) with a tab characters (\t).

Note that this may take Notepad++ a few minutes to complete this transformation after you click replace.

1.7.3 Put the Breaks Back

We now want to put back a line break (\n) after each comment, as the last step put everything on one line. Lucky we have an easy way to identify when one comment ends and another starts. The dataset should now have a tab character (\t) followed by a line of 111 dash (-) characters followed by a tab character (\t) after each comment.

Write a regular expression to replace these occurrences of a tab, exactly 111 dashes, and then another tab with a line break. For full marks do not type the dash character output 111 times in your pattern, rather use repetitions so you only have to write the dash character once in your pattern.

Example:

Before these steps your data should look something like this:

```
Comment by: K****n
Comment Text: Hi Barbie
Likes: 0
Replies: 0
Country: None
Comment Published on: 2024-02-28 08:52:28 PM
Comment Last Updated: 2024-02-28 08:52:28 PM
Commenter's Channel Publsh Date: 2023-11-06 07:31:16 PM
-----------------------------------------------------------------------------------------------------------
Comment by: u****n
Comment Text: I liked the dancing hippo
Likes: 0
Replies: 0
Country: None
Comment Published on: 2024-02-28 07:33:35 PM
Comment Last Updated: 2024-02-28 07:33:35 PM
Commenter's Channel Publsh Date: 2023-12-24 12:04:43 AM
-----------------------------------------------------------------------------------------------------------
```

```
Comment by: g****9
Comment Text: Finally saw this movie. Can't believe how BAD it was.
Likes: 0
Replies: 0
Country: None
Comment Published on: 2024-02-28 07:28:17 PM
Comment Last Updated: 2024-02-28 07:28:55 PM
Commenter's Channel Publsh Date: 2015-09-07 01:42:23 PM
-----------------------------------------------------------------------------------------------------------------
```

after steps 1.7.1 to 1.7.3 your data should now look like:

```
K****n→Hi Barbie→0→0→None→2024-02-28 08:52:28 PM→2024-02-28 08:52:28 PM→2023-11-06 07:31:16 PM
u****n→I liked the dancing hippo→0→0→None→2024-02-28 07:33:35 PM→2024-02-28 07:33:35 PM→2023-12-24 12:04:43 AM
g****9→Finally saw this movie. Can't believe how BAD it was.→0→0→None→2024-02-28 07:28:17 PM→2024-02-28 07:28:55 PM→....
```

The → characters shown above represent a tab character (\t). The ... on the last line is simply indicating that this line is to long to be shown in this document. Your file should now be in Tab-Separated Values format (TSV).

Hint 8: For step 1.7.1 it maybe helpful to use an anchor to match the start of a line and the colon to match the end of a data value name.

Hint 9: For step 1.7.2, don't worry if this makes everything go on one line. That's the idea. We add the line breaks back in step 1.7.3.

Hint 10: For step 1.7.3, don't forget to match the tabs before and after the line of dashes.

Step 1.8 Clean up and Save The Data

Manually (by hand) add a new line to the top of the file with header names for each column like so:

```
User→Text→Likes→Replies→Country→Published→Updated→Channel Date
```

The → characters shown above represent a tab character. Make sure these are tabs in your file and not literal arrow characters.

Make any other manual changes needed to convert this file into a TSV file. In most cases, this should not be necessary if the regular expression patterns were applied correctly.

Once you are done, save the file as *vidname.tsv* where *vidname* is the name of your dataset. Make sure you have given the file just a .tsv extension and not .tsv.txt, .txt.tsv, .tsv.tsv, or any other inappropriate file extension name.

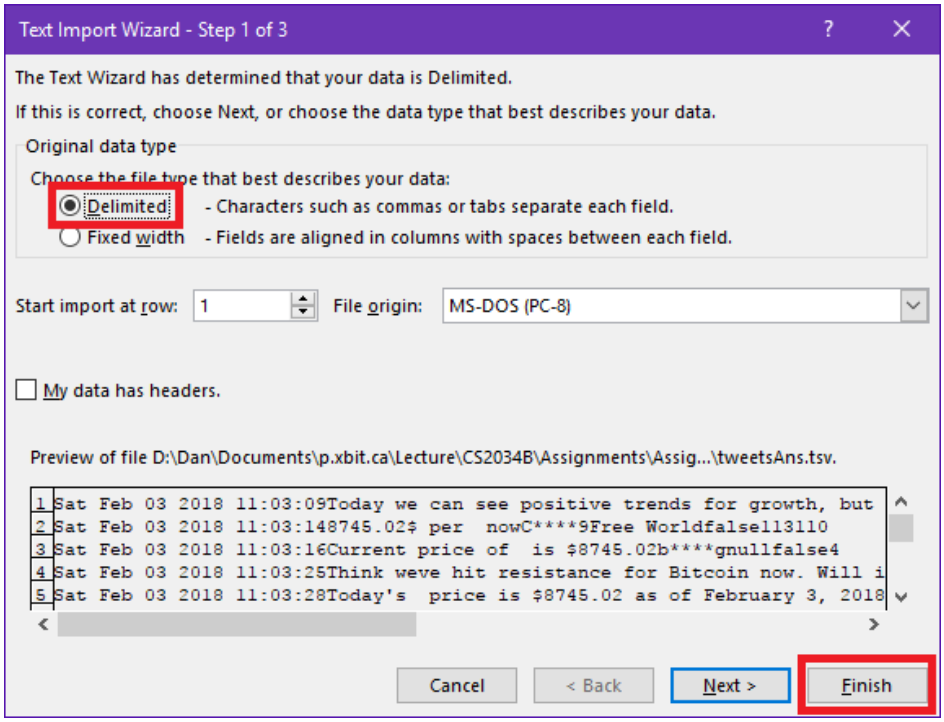
Task 2: Importing Data Into Excel

Step 2.1 Import the Data

Make a copy of your processed dataset file named *vidname.tsv* where *vidname* is the name of your dataset. Make sure you do not delete or modify your processed file as you are required to submit this file with your assignment.

Open the TSV file in Excel (note that you may have to change the file type drop down to "All Files (*.*)" rather than "All Excel Files"). You can also attempt to open it by dragging and dropping the file into Excel.

If the following window is shown, make sure "Delimited" is checked and simply press the "Finish" button:

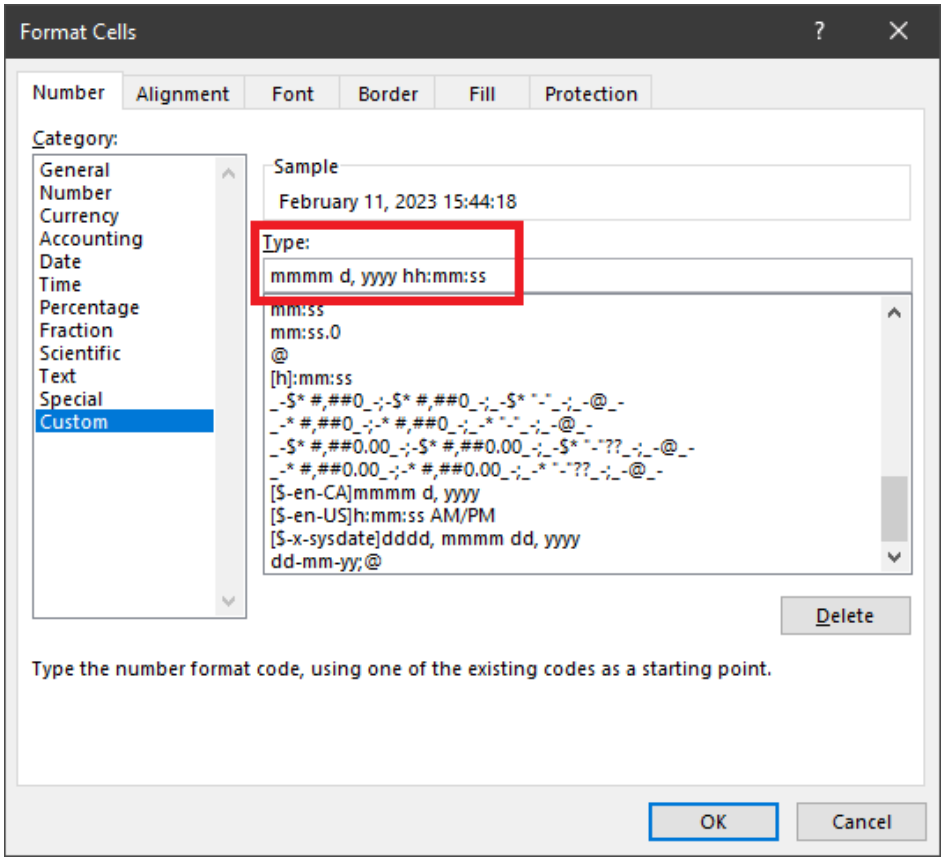


Save your file as an Excel Macro Enabled Workbook named *userid_assign2.xlsm* where *userid* is your UWO user id. If you keep working on it as a TSV file, you will lose all of your formatting, formulas and code if you close and reopen it!

Perform following formatting steps:

1. Adjust the column widths to show all of the data (no columns should be showing as all ########s).
2. If your data was imported with any empty columns (e.g. a blank column A) delete this column (just make sure it truly is empty).

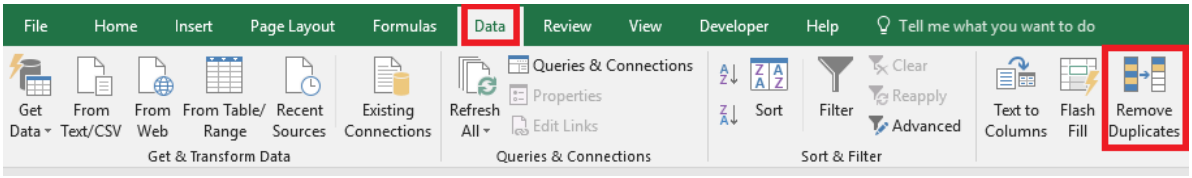
3. Make the header text bold and centred.
4. Format the date columns as a Custom date format by highlighting the date values in the column, right clicking, selecting "Format Cells", and on the Number tab select a Number Category of "Custom". Input the format code `mmmm d, yyyy hh:mm:ss` under "Type" as shown in the screen shot displayed after this list.
5. Look through the data in your worksheet and ensure everything looks correct, that no values are in the wrong column, and nothing that should have been removed is still in the data. You may clean this up a bit by hand, but keep in mind most of these issues should have been dealt with by your regular expressions.
6. Name the sheet "Data".



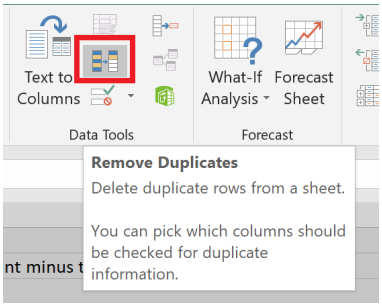
Step 2.2 Sort the Data and Remove Duplicates using Excel

Sort your data by the **comment text** (should be column B) using the order A-Z.

Use the Excel's remove duplicate feature (found in the Data Toolbar) to remove the duplicate comments based on the comment text (column B).



It may also look like this in the Data tab:



Make sure only the comment text column is selected and that you indicate that your data has headers. Depending on your dataset a different number of rows may be removed. In some cases it may be possible for no rows to be removed if there are no exact duplicates in your dataset.

Task 3: Clean Text & Country Names

Step 3.1: Clean Text

Before we can perform our sentiment analysis, we need to clean the comment text. To do this, we will create a VBA function that takes as input the text of a comment and returns the comment's text with all punctuation and other non letter characters and spaces removed.

The header for this function must be:

```
Function CleanText(text As String) As String
```

Your CleanText function must do the following:

1. Convert all letters to lower case.
2. Remove any non letter characters (characters that are anything except for an English letter).
3. Trim the text (remove any leading or trailing spaces).

Once your function is working correctly, insert a new column between column B and C and give it the header "Cleaned Text". In this column use your CleanText function to clean the text of the comment on the same row (in column B).

Example Inputs:

```
I <3 this Movie!!!
I'm not sure if this is rated PG or PG13?
?!?!?!??!!??!
```

Example Output:

```
i  this movie
im not sure if this is rated pg or pg
```

Note that the last input would result in an empty string as there are no letters or spaces.

Hint 11: Your VBA function will need to loop through the characters in the string and build a new string to return that omits any non letter/space characters.

Hint 12: You can make use of VBA string functions such as LCase, Mid, and Trim to make this easier.

Hint 13: VBA does not have a function to check if a character is a letter, but you can compare strings using the > and < operators (as well as others such as >=). If a string with a single character is between "a" and "z" inclusively, it means it is a lower case letter.

Step 3.1: Country Names

Currently the country names are displayed as a two letter ISO 3166 country code. We would rather have the full country name.

Download the file *CountryCodes.csv* from OWL and import their codes into a new sheet in your Excel WorkBook named "Countries". Make the headers bold and make sure both columns have enough width to display all of the data.

Create a VBA function named *FindCountryName* that takes as input a two letter country code and returns the full country name. It's up to you to create the correct function header for this function.

Your function should accomplish this by reading the values from the *Countries* sheet using a hardcoded ranges in your VBA code. You can not use an Excel function for this.

If an invalid or unknown code is given to your function it should return the text "ERROR". If the country value is not set and is just "None" in your data, this function should return "None" rather than a country code.

Example Inputs:

```
CA
US
GB
ZX
None
```

Example Output:

```
Canada
United States of America (the)
United Kingdom of Great Britain and Northern Ireland (the)
ERROR
None
```

Note that ZX was not a valid code, and thus resulted in the text "ERROR" being returned.

Once your function is working add a new Column between columns F and G and give it the header "Full Country". In this column use your *FindCountryName* function to look up the full name of the country for each row based on the country code in column F.

Hint 14: To get the ranges from the **Countries** sheet use Worksheet and Range object like so:

```
Dim codes As Range
Set codes = Worksheets("Countries").Range("B2:B250")
```

This will give you the range `B2:B250` from the sheet named **Countries** as the variable named `codes`. You will have to do something similar to access the country names.

Task 4: Calculate Sentiment

Step 4.0: Keywords Worksheet

Copy the data in the *keywords.csv* file (downloaded from OWL) and add it as a new sheet with the name **keywords** in your workbook. You will be using this sheet with the functions you make in the following tasks.

This file contains a list of keywords in column A. The keywords are both positive such "happy", "great", and "good" as well as negative such as "ban", "worst", and "panic". Each word is associated with a score in column B. The higher the score the more positive the word, the lower the score the more negative the word. This list of keywords will be used in our sentiment analysis to determine if a comment is positive or negative overall.

Copy this worksheet into your Excel Workbook in a new sheet named "Keywords", make sure you copy all words and scores and that the data starts in cell A1.

Step 4.1: sentimentCalc Function

Create a VBA function named `sentimenCalc` that determines the sentiment of each comment based on its contents. The header for this function must be:

```
Function sentimentCalc(text As String) As Integer
```

This function should check each word in the comment text and if the word exists as one of the keywords in the Keywords sheet it should impact the overall sentiment value by adding the sentiment value given in the Keywords sheet to the total for this comment. Access the keywords and scores as ranges from the Keywords sheet within your VBA code. The case of the words should be inconsequential. For instance, happy, HAPPY, or hApPy are all treated as the same word regardless of the capitalization.

For example, if the comment text is:
`awesome, our family loved watching this`

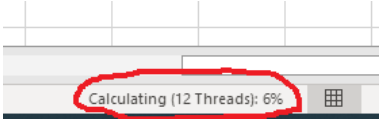
The score returned by the sentimentCalc function should be 7. 4 points for "awesome" and 3 points for "loved" (these point values come from the Keywords sheet).

As another example, if the comment text is:
`i hate hate hate hate this video but i like the main actor and i am a loyal fan`

The score returned by the sentimentCalc function should be -4. Each instance of "hate" is -3 points (-12 in total), "like" is 2 points, "loyal" is 3, and "fan" is 3.

If the text contained no words from the Keywords sheet, the result should be a sentiment score of 0 (zero).

Use this function in your **processedData** worksheet to create a new column (column K) that calculates the sentiment value for each comment **based on the Cleaned Text** in column C. Give your new column the header "Sentiment". Note that this may take Excel some time to calculate, the progress Excel has made will be displayed in the bottom right as shown below. It is a good idea to **save before imputing this formula** and to wait for this calculation to complete before making any other edits to your Excel workbook.



Some Hints:

You will need to use the string function Split in this function to break the text into words.

To get the ranges from the **keywords** sheet use Worksheet and Range object like you did in Step 3.1.
You will need to use nested loops. One to go through each word in the keywords and one to go through each word in the comment text.

Many of the comments in your dataset are likely to have a neutral sentiment (zero score). This is fine if it is correct. You may want to check some comments by hand to ensure you are getting the correct result.

Step 4.2: sentimentCategory Function

Create a VBA function named `sentimentCategory` that categorizes a sentiment value into "Positive", "Negative" or "Neutral". The header for this function must be:

```
Function sentimentCategory(sentVal As Integer) As String
```

This function should return the sentiment category as a String based on the given Integer sentiment value such that:

- If the sentiment value is greater than 0, the category is "Positive".
- If the sentiment value is less than 0, its category is "Negative".
- If the sentiment value is equal to 0, its category is "Neutral".

In column L, use the above function to determine the category of each comment based on the sentiment value in column K (calculated in Step 4.1). Give this column the header "Category".

Task 5: Analysis

Step 5.1: Calculations

Create a new worksheet in your workbook called **Analysis** where we will present the results of our analysis. Recall that you can reference other worksheets in an Excel formula using `!`. For example, `=Data!L2` would be equal to the sentiment category of the 1st comment in the **Data** sheet, even if you use this formula in the **Analysis** sheet.

In this sheet you should calculate the average sentiment and total number of positive, negative and neutral comments for a few different groups of users. You should only use Excel formulas and built in Excel functions and not VBA code for all parts of this task (Task 5). Do not hard code any values or results.

Your worksheet should look like the following screen shot after you are finished (although your numbers will be different) including the formatting of the cells:

Overall Sentiment	
Average Sentiment	0.870430906
Total Positive Comments	1105
Total Negative Comments	484
Total Neutral Comments	1776

Liked Sentiment	
Average Sentiment of Liked Comments	0.855516
Total Positive Liked Comments	505
Total Negative Liked Comments	305
Total Neutral Liked Comments	595
Average Number of Likes on Positive Comments	6.353846
Average Number of Likes on Negative Comments	1.522727
Average Number of Likes on Neutral Comments	1.771959

Average Sentiment by Location	
Australia	0.75
Azerbaijan	0.25
Bangladesh	-0.75
Brazil	0.777777778
Canada	2.5
Germany	1.454545455
India	0.636363636
Jamaica	0.25
Japan	0
New Zealand	1.5
Pakistan	-0.42857143
Russian Federation (the)	0.5
Turkey	0.647058824
United Kingdom of Great Britain and Northern Ireland (the)	2.058823529
United States of America (the)	0.806122449

Reply Sentiment	
Average Sentiment of Comments with Replies	0.975138
Total Positive Comments with Replies	291
Total Negative Comments with Replies	190
Total Neutral Comments with Replies	243
Number of Replies on Postive Comments	972
Number of Replies on Negative Comments	583
Number of Replies on Neutral Comments	625

Overall Sentiment should present the average sentiment (the average of all the sentiment values) and counts of the number of comments with a positive, negative, and neutral sentiment (based on the sentiment category) of all of the data in the **Data** sheet.

Liked Sentiment

- **Average Sentiment of Liked Comments:** The average sentiment of comments that have at least one like.
- **Total Positive/Negative/Neutral Liked Comments:** The total number of comments that have at least one like and are classified as Positive/Negative/Neutral (depending on which item in the table it is for).
- **Average Number of Likes on Positive/Negative/Neutral Comments:** The average number of likes a comment has, but only for comments classified as Positive/Negative/Neutral (depending on which item in the table it is for).

Reply Sentiment

- **Average Sentiment of Comments with Replies:** The average sentiment of comments that have at least one reply.
- **Total Positive/Negative/Neutral Comments with Replies:** The total number of comments that have at least one reply and are classified as Positive/Negative/Neutral (depending on which item in the table it is for).
- **Number of Replies on Positive/Negative/Neutral Comments:** The total number of replies for all comments classified as Positive/Negative/Neutral (depending on which item in the table it is for).

Average Sentiment by Location the average sentiment for users that reported their location as being from this country. You must list 15 countries here. Each country you list should have at least two comments in your dataset. They don't have to be the same countries as shown in the screenshot but you must use the same spelling of the countries name as shown in your Countries worksheet (they have to match exactly).

Your formula must not hardcode the countries name, and it should be possible to fill (copy and paste) the formula down for each country to find it's sentiment without manually modifying the formula.

To get a good idea of how many comments were made from each country you can add a column to your Countries sheet to count how many comments were made from each country using `COUNTIF`. You should select the countries with the most comments to include in your analysis.

Step 5.2: Summary

Add a textbox to the Analysis worksheet with at least two paragraphs of text that gives a summary of our analysis and comments on the sentiment of the comments in your dataset. You should consider if this keyword has a mostly positive or negative sentiment in the different categories we looked at and why this might be (most comments will have a neutral sentiment so focus on things like the ratio of positive v.s. negative comments). Make sure you consider potential sources of bias in your summary. Consider where this video is coming from (who posted it), who the audience is, what type of person is likely to be commenting on it, how large the sample size is, and if any of these factors could introduce bias in your results. You are free to do additional calculations as part of your analysis such as find the percentage of positive/negative comments or the ratio of positive v.s. negative comments if you wish (but this is not required).

Submission

You must submit the following files to OWL:

1. Your Excel file as a .xlsm file (Macro Enabled Workbook) and name it *"userid_assign2.xlsm"* where *userid* is your user id. For example, if your uwo e-mail was *"dservos5@uwo.ca"*, the file should be named *"dservos5_assign2.xlsm"*.
2. A Word document that contains the regular expressions and replacements you used in Task 1. Name the Word document *"userid_assign2.docx"* where *userid* is your user id.
3. A copy of *vidname.tsv* after you have performed the Data Munging tasks in Task 1.

Your files must be in the correct format. Files in any other format (.PDF, etc.) will not be marked and will receive a zero grade.

Before submitting, ensure that your .xlsm file contains all code for your functions and that your assignment works correctly on the GenLab computers and with Excel for Windows. Also check that the regular expressions you have listed in the Word document will work in Notepad++.

In addition to the late penalty (as outlined in the course syllabus), penalties will be given for failing to submit all files through OWL correctly (a **minimum** of 8 marks per file), naming files incorrectly (3 marks per file), or otherwise failing to follow instructions outlined in this document. Your VBA code must be documented with comments as described at the beginning of this document.