COMPSCI 2034b / DIGIHUM 2144b

Data Analytics: Principles and Tools

Due: Check OWL page

Total: 100 Points (5% of Final Grade)

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 download the files from OWL named *tweets.txt* and *keywords.csv*. Follow the directions given in each task in this document precisely and produce a PDF file named *userid assign2.pdf* and an Excel Macro Enabled Workbook named *userid assign2.xlsm* (where *userid* is your UWO user id). 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 enough 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 from OWL.
- Properly cleaning and importing the tweets 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.**
- Assignment submission via OWL.

Problem Description

In this assignment, you will pre-process, analyze, and present data relating to individual's current opinion on Bitcoin. The dataset (*tweets.txt*) we will be using contains just under 3000 tweets relating to Bitcoin that were made between 11:00AM and 9:30PM on February 3rd 2018. The dataset has been filtered to remove non-english tweets and retweets. Some spam filtering has also been applied to remove automated tweets trying to promote sites and services.

You will act as a data analyst and perform some textual analysis on this data to attempt to derive some meaning. In this case, the current sentiment or opinion twitter users have of Bitcoin. For currency and stock market traders, this kind of analysis of social media data can be a useful indicator of a currency or stock's current public opinion. A change to a very negative public sentiment could indicate a sell-off is coming, while a change to a positive sentiment could indicate that a rally will occur in the near future.

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

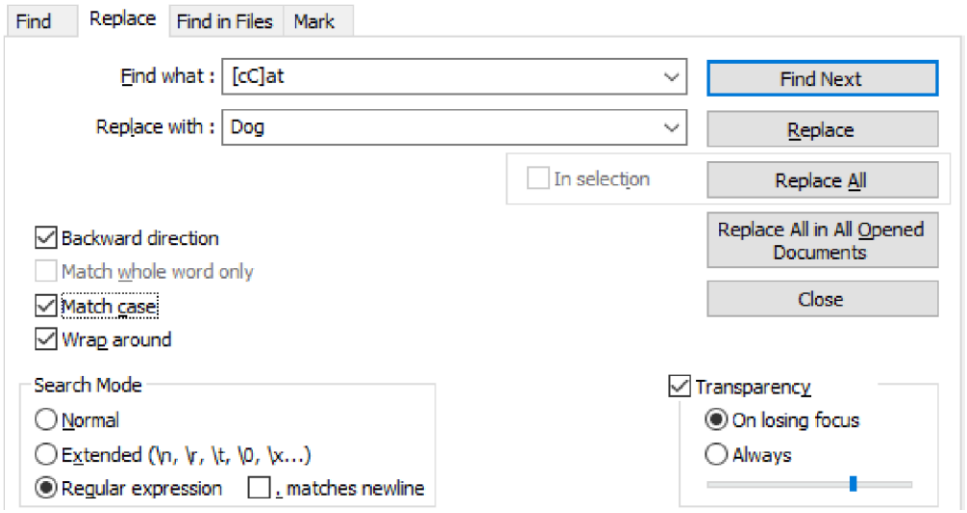
- **Data Munging**: Clean the dataset using regular expression.
- **Importing:** Import the data into Excel, sort it and use Excel's built in remove duplicate tool.
- **Remove Duplicates:** Use VBA to create our own duplicate removal tool to further clean the data.
- **Calculate Sentiment:** Use VBA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

Tasks

Task 1: Data Munging

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

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



you should record the following in your PDF 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[1] to denote a spaces in your pattern. If you do this leave a note stating something to the affect of ␣ = space so that your intent is clear to the reader.

Step 1.1: Understanding the Data

Download the *tweets.txt* file from OWL. If you are having trouble downloading tweets.txt, try right clicking on tweets.txt and selecting *“save as”*, *“save link as”* or *“save target as”* depending on the browser you are using.

This file contains 2843 tweets about Bitcoin 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 line of this file contains a single tweet as well as metadata about the tweet. Each data value is separated by a Tab character (\t) but unlike a TSV (Tab-Separated Values) file, each data value is prefixed with the name of that value. For example, the name of the user who made the tweet is prefixed with the text *"screen name: "* and the date the tweet was posted is prefixed with the text *"posted: "*.

The following table describes each data value in the file:

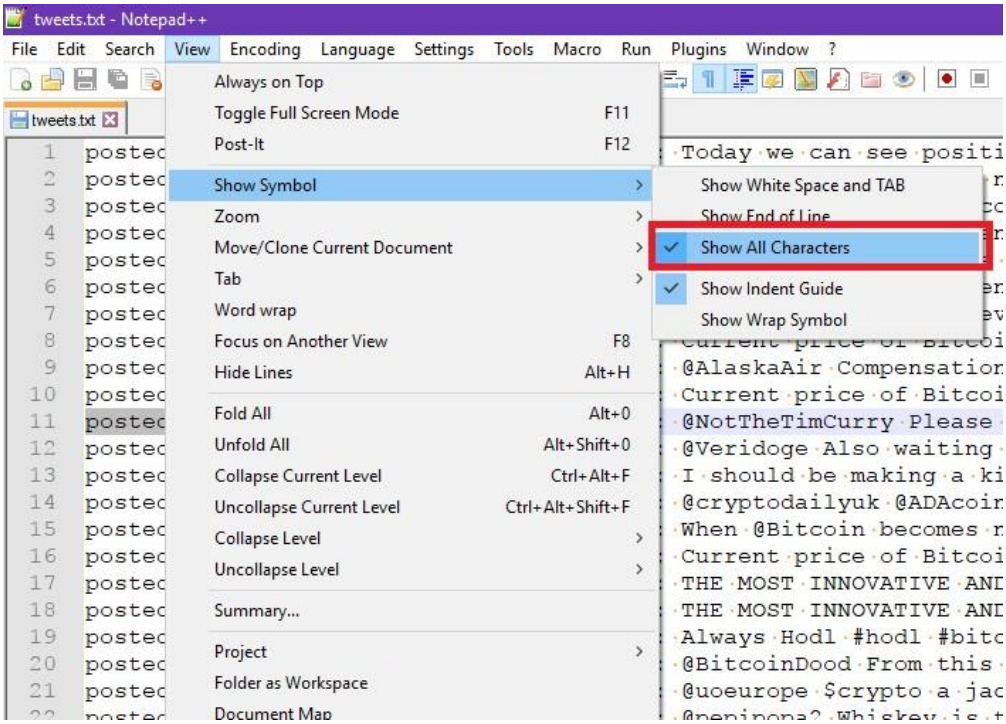
Data Value Name	**Description**
posted	The date and time the tweet was posted to twitter.
text	The text of the tweet including hash tags and mentions.
screen name	The user name of user who posted the tweet.
location	The location the user gave in their twitter profile (may not be reliable).
verified	If the user is a verified user on twitter (true or false value).
followers count	The number of followers this user has.
friends count	The number of friends this user has.
lang	The language setting this user is using (e.g. "en" for English, "ru" for Russian, etc.)
retweet count	The number of times this tweet was retweeted.
favorite count	The number of times this tweet was favorited.

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:

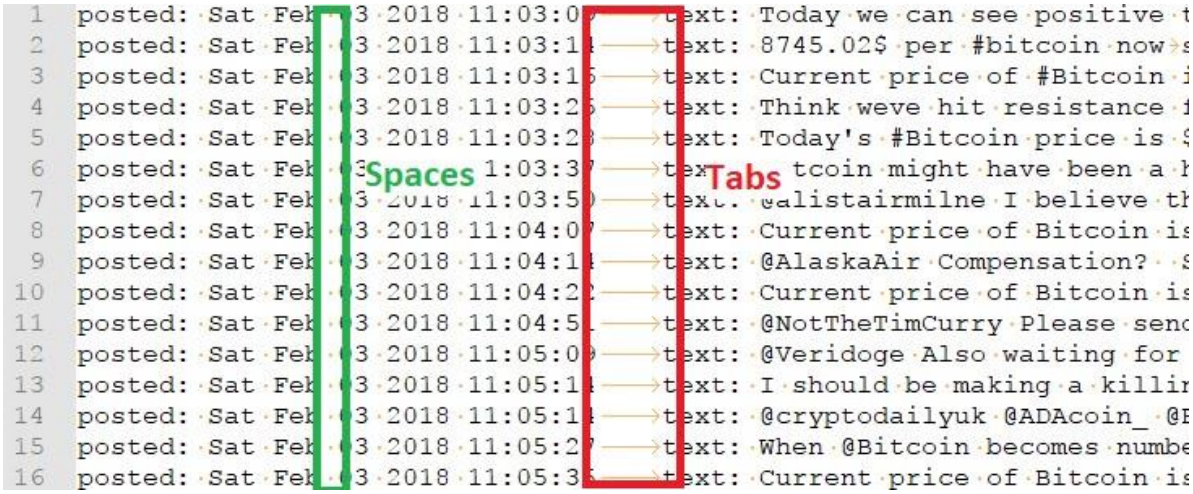
posted: Sat Feb 03 2018 11:04:51

where posted is the name of the data value and ␣ is a single space. The data value name is followed by a colon and single space (:␣) and then the data value, Sat Feb 03 2018 11:04:51.

The best way to fully understand this format and *tweets.txt* 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 and space characters as orange dots:



Step 1.2: Remove Mentions and Hashtags (5 Marks)

As we only care about the text of the tweet while doing a sentiment analysis and not the user mentions or hashtags they contain, we will need to remove them. Using Notepad++ (available for free for Windows and installed on the GenLab computers) or an equivalent program create a regular expression based Find and Replace pattern to remove all user mentions from *tweets.txt*.

User mentions start with a @ character and are followed by a twitter user name. For our purposes, assume that a twitter user name may only contain alphanumeric characters (upper-case and lower-case letters and numbers) and underscores ( ). Your pattern should also remove any occurrence of the @ character that is not followed by a username. Some example user mentions from the dataset:

@
@BitcoinDood
@pepipopa2
@WillCode 4Beer
@CryptopiaNZ

@([a-zA-Z_0-9])*

@[a-zA-Z_0-9]*( )?
连着后面的tab一块儿replace

Once you have successfully removed the user mentions, create a similar regular expression based Find and Replace pattern to remove all hashtags from *tweets.txt*. Hashtags start with a # character and may be followed by any number of characters as long as they are not spaces or tabs. This includes a single #. Some examples from the dataset:

#
#Bitcoin
#besttweet
#WeirdNewCollegeCourses
#100000000percentreturns
#2018
#Big3

#([a-zA-Z_0-9])*

#[a-zA-Z_0-9]*( )?

#[a-zA-Z_0-9]* ?
注意这里有个空格

Step 1.3 Remove any -, = or Space at the Start of a Tweet (5 Marks)

We need to remove any -, = or space (i.e. the minus sign, equals sign, or the space character) at the start of the tweet text before we can import the data into Excel or Excel will think this is a formula.

text:\s[-=\s]+ replace with text:space

Create a regular expression based Find and Replace pattern to remove any number of -, = or space if they occur at the start of a tweet. You should only remove these characters if they are the first characters of a tweet and not everywhere in the file.

Hints: You can use the data value name "text: " to help match the beginning of a tweet. If you use "text: " in your pattern make sure you put it back in the replace field or it will be deleted.

Step 1.4 Cleaning up the Spaces (3 Marks)

Removing the hashtags and user mentions in step 1.2 may have left some extra spaces in the tweets. For example, if the tweet was "Hello @alice my name is Bob.", removing @alice would leave an extra space and the tweet would be "Hello my name is Bob.".

Provide a regular expression based Find and Replace pattern to remove all occurrences of two or more spaces in the file with a single space. ( ){2,} replace with: space

Hint: \s matches both spaces and tabs. In this case, we only want to replace two or more occurrences of spaces and not tabs.

Step 1.5 Anonymize the Screen Names (8 Marks)

As we will not be using the screen names we should anonymize them to protect the user's identity. Alter each screen name in *tweets.txt* such that only the first and last character of the screen name is shown, separated by exactly four asterisks (*). For example, the screen name btc joe5 would become b****5. (screen_name: [0-9A-Za-z_])[0-9A-Za-z_]*([0-9A-Za-z_]) $1****$2

Provide a regular expression based Find and Replace pattern to perform this substitution.

Hints: You may need to use groupings in your regular expression and replace fields. It may be useful to include "screen name: " in your pattern to match only the screen names in the file, just be sure to put it back in your replace field.

Step 1.6 Remove the Data Value Names (6 Marks) (\t|^)([0-9A-Za-z_]+:\s) Replace with $1

Before we can import *tweets.txt* 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 and the colon (:) and single space they are followed by (e.g. *"posted: "*, *"text: "*, or *"screen name: "*) from the data. Do not do this step until you have completed the previous steps as removing the names will make the previous steps much harder.

Provide a regular expression based Find and Replace pattern to remove all of the data value names and the colon and space they are followed by. Your pattern should work for all possible data value names, where a name can only have lowercase letters and underscore characters, is always followed by a colon (:) and single space and always occurs after a tab or the start of a line. Your pattern should not accidentally remove colons or words from the tweet's text which is allowed to contain any number of colons.

For full marks, do this with one Find and Replace pattern that does not "hardcode" the data value names (e.g. your pattern should not have the string *"posted:"* or *"text:"* in it). For partial marks you may use multiple Find and Replace patterns and hardcode the data value names.

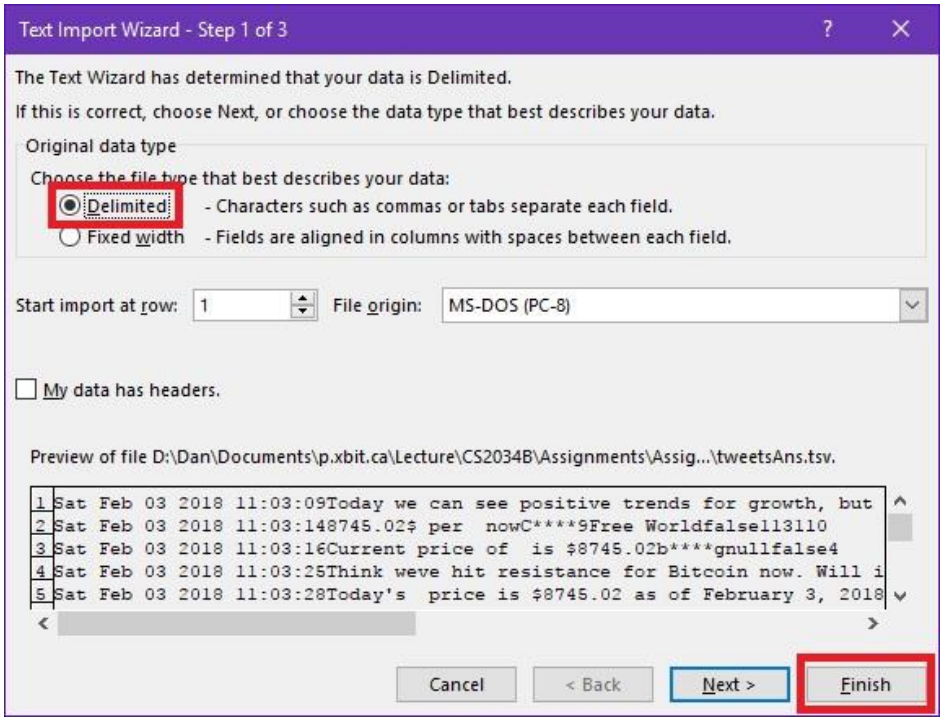
Task 2: Importing Data Into Excel

Step 2.1 Import the Data (2 Marks)

Make a copy of your processed *tweets.txt* file named *tweets.tsv*. Make sure you do not delete or modify your processed *tweets.txt* file as you are required to submit this file with your assignment.

Open *tweets.tsv* 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:

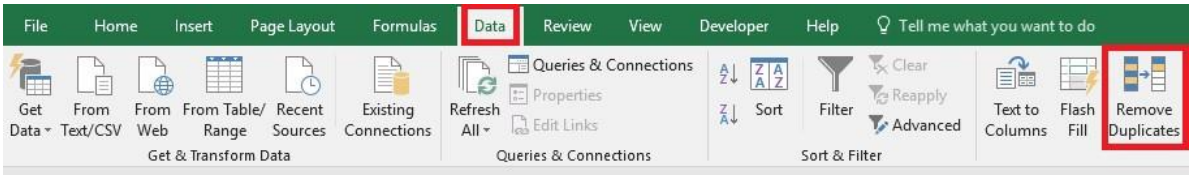
1. Adjust the column widths to show all of the data.
2. Add a new row to the top of sheet and enter some header text for each column. If you do not recall what each column is, refer to the original *tweets.txt* file and the table in Step 1.1.

3. Make the header text bold.
4. Delete columns I and J (the retweet and favourite counts) as they contain no useful data.

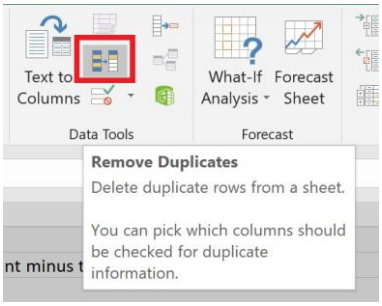
Step 2.2 Sort the Data and Remove Duplicates using Excel (5 Marks)

Sort your data by the tweet text (column B) using the order A-Z.

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



It may also look like this in the Data tab:



Make sure only the tweet text column is selected and that you indicate that your data has headers. **Approximately**, 150 rows should be removed from your sheet and you should have around 2691 rows left including your header row.

Task 3: Remove Duplicates

Step 3.1: isDup Function (20 Marks)

While Excel's built in remove duplicates feature was able to remove identical tweets, we would also like to remove similar but not identical tweets like:

$1 USD is currently worth 0.00010627 BTC! and

$1 USD is currently worth 0.00010692 BTC!

To do this we will create a VBA function named isDup that will detect if two tweets are sufficiently similar to be considered a duplicate. isDup will return True if two tweets given to it are duplicates and False otherwise. This function must have the following function header:

```
Function isDup(tweet1 As String, tweet2 As String, threshold As Double) As Boolean
```

where tweet1 and tweet2 are the text of two different tweets and threshold is a percentage of the number of words that they must have in common to be considered a duplicate. It is based on the total number of words in the first tweet. If we are using a threshold of **0.7** and the first tweet has 100 words, at least 70 of those words must be in common with the second tweet for isDup to be True. If it is less than 70 words like 56 or 34 then the tweet is not deemed a duplicate and False should be returned. Note that threshold is passed as an argument to the function with a value between 0 and 1 and should not be hard coded as 0.7

in your function.

Example:

If tweet1 is:
Hours of planning can save weeks of coding

and tweet2 is:
Weeks of programming can save you hours of planning

The correct count for words in common should be 7 out of 8 as the only difference is "coding" v.s. "planning" and isDup should return True if the threshold is less than 0.875. Note that the total number of words is based on the length of tweet1 and each word in tweet1 is matched at most once. "of" occurs twice so it is matched twice in tweet1 (and not four times). **Capitalization should be ignored.**

SomeHints:

You will need to use the string functions StrCom p and Split. Use `Split` To break the strings Into individual words and `StrComp` To compare them while ignoring capitalization.

You will need to use nested loops. One to go through each word of `tweet1` And one to go Through each word of `tweet2`.

```
Function isDup(tweet1 As String, tweet2 As String, threshold As Double) As Boolean
    Dim count As Integer
    Dim result As Double
    Dim tw1_array() As String
    Dim tw2_array() As String

    tw1_array = Split(tweet1)
    tw2_array = Split(tweet2)

    For i = LBound(tw1_array) To UBound(tw1_array)
        For j = LBound(tw2_array) To UBound(tw2_array)
            If StrComp(LCase(tw1_array(i)), LCase(tw2_array(j)), vbTextCompare) = 0 Then
                count = count + 1
                tw2_array(j) = ""
            End If
        Next j
    Next i

    result = count / (UBound(tw1_array) - LBound(tw1_array) + 1)

    If result > threshold Then
        isDup = True
    Else
        isDup = False
    End If

End Function
```

Examples:

tweet1="Hours **of** planningcansaveweeks **of** coding"
tweet2="Weeks **of** programmingcansaveyouhours **of** planning"
7/8wordsthesame

tweet1="Hours **of** planningcansaveweeks **of** coding"
tweet2="Weeks **of** programmingcansaveyouhoursplanning"
7/8wordsthesame

tweet1="Hours **of** planningcansaveweeks **of** coding"
tweet2="Weeksprogrammingcansaveyouhoursplanning"
5/8wordsthesame

tweet1="Hours **of** planningcansaveweekscoding"
tweet2="Weeks **of** programmingcansaveyouhours **of** planning"
6/7wordsthesame

tweet1="Hoursplanningcansaveweekscoding"
tweet2="Weeks **of** programmingcansaveyouhours **of** planning"
5/6wordsthesame

tweet1="Hoursplanningcansaveweekscoding"
tweet2="Weeksprogrammingcansaveyouhoursplanning"
5/6wordsthesame

Step 3.2: Use isDup to Remove Duplicates (5 Marks)

Perform the following steps in order:

1. Create a new column with the header isDup (column I).
2. Ensure that you sorted your data correctly in Step 2.2 (by tweet text) or this step will not work correctly.
3. Use the isDup function to determine whether a tweet is like the tweet that follows it. Check to see whether the tweet directly after it is a duplicate. Use a threshold of **0.7**. For each cell in column I (isDup), call the isDup function with the tweet text on the current row and the tweet text for the next row.
4. Name the current worksheet in your workbook **rawData**.
5. Reformat the column widths and headers as needed in the new worksheet.
6. Copy all the data values in the rawData sheet into a new worksheet and name it **processedData**. Make sure you are only copying values and not formulas.

7. In the **processedData** worksheet, sort the data by the isDup column.
8. Delete all rows that have a TRUE in the isDup column (you can do this manually).

After these steps you should have **approximately** 2400 rows in your **processedData** worksheet (you may have more or less depending on how you cleaned your data or coded your isDup function).

Task 4: Calculate Sentiment

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.

Step 4.1: sentimentCalc Function (20 marks)

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

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

This function should check each word in the tweet and if the word exists as one of the keywords in the positive list or negative list it should impact the overall sentiment value. The positive list and negative list words exist in the **keywords** sheet. Access the keywords as ranges within your VBA code. The case of the word is inconsequential. For instance, happy, HAPPY, or hApPy are all treated as positive words regardless of their case *(Hint: StrComp)*.

If the word is in the positive list, it should increase the sentiment value by 10, if it is in the negative list it should decrease it by 10. For instance, if the positive list includes "happy", "rally", "growth" and the negative list includes "crash", "scam", "bad" then:

> If the Tweet is *"I am Happy that Bitcoin is showing growth."*. The sentiment value will be 10 + 10 = 20
>
> If the Tweet is *"I am happy that Bitcoin is a scam and will CRASH!"* The sentiment value will be 10 – 10 – 10 = -10

You must remove the following punctuation characters from the tweet text in your VBA code before calculating the sentiment: **! . , ? :) (;**

You may do this using multiple lines each calling the Replace function or with an array, loop and one call to the Replace function. Both methods will be marked as correct.

Use this function in your **processedData** worksheet to create a new column (column J) that calculates the sentiment value for each tweet.

SomeHints:

You will need to use the string functions StrComp, Split and Replace In this function.

To get the ranges from the **keywords** Sheet use Worksheet and Range object like so:

```
Dim positive As Range
Set positive = Worksheets("keywords").Range("A2:A76")
```

This will give you the range A2:A76 From the sheet named **keywords** As the variable named positive. You can do the same for the negative range (but with different cell references And variable names).

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 tweet text.

Step 4.2: sentimentCategory Function (5 Marks)

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 K, use the above function to determine the category of each tweet based on the sentiment value in column J (calculated in Step 4.1).

Task 5: Analysis (16 Marks)

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, =processedData!K2 would be equal to the sentiment category of the 1st tweet in the **processedData** 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 tweets for a few different groups of users. You should only use Excel

formulas and built in Excel functions and not VBA code for this task. Do not hard code any values or results.

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

	B	C
Overall Sentiment		
Average		1.40
Total Positive		530
Total Negative		266
Total Neutral		1640
Verified Sentiment		
Average		1.85
Total Positive		6
Total Negative		2
Total Neutral		19
Average Sentiment By Location		
Australia		3.13
Canada		1.43
England		-0.41
France		2.22
Germany		-2.86
India		-0.74
Ireland		1.25
Japan		5.56
Netherlands		2.22
Nigeria		3.53
South Africa		1.33
Spain		2.00
USA		2.28

	F	G	H
Over 3,000 Follower Sentiment			
Average		1.77	
Total Positive		69	
Total Negative		28	
Total Neutral		214	
Average Sentiment By User Language			
English		en	1.41
Russian		ru	5.16
Spanish		es	1.76
German		de	0.00
Portuguese		pt	2.63
French		fr	-1.11
Dutch		nl	1.61
Italian		it	0.00
Japanese		ja	0.87
Turkish		tr	0.77

Overall Sentiment should present the average sentiment (the average of all the sentiment values) and counts of all of the data in the **processedData** sheet.

Verified Sentiment should present the average sentiment and tweet counts of verified users only (i.e. users that have a TRUE in the Verified column (column E) in the **processedData** sheet.

Over 3,000 Follower Sentiment should present the average sentiment and tweet counts for only users that have over 3,000 followers.

Average Sentiment by User Language should show just the average sentiment for users that reported their language as the given language code (e.g. “en” for English, etc.) in column H of the **processedData** sheet. You will have to type in the languages and codes shown in the screen shot.

Average Sentiment by Location should show just the average sentiment for users that reported their location from one of the locations shown in the screen shot above (you will have to type these in). The location can appear anywhere in the text of the cell in the Location column (column D) in the **processedData** sheet. For example, if a user's location is given as "London, Ontario, Canada" or "Canada, North America" they should both count for Canada in your analysis. You do not have to consider cities, states or provinces, just the countries and locations shown in the screen shot. *Hint: You can use wild cards in your criteria (e.g. "=*cat*" would match the text "cat" anywhere in cell if used with a function like AVERAGEIF or COUNTIFS).*

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 *"cbrogly@uwo.ca"*, the file should be named *"cbrogly_assign2.xlsm"*.
2. A PDF document that contains the regular expressions and replacements you used in Task 1. Name the PDF document *"userid assign2.pdf"* where *userid* is your user id.
3. A copy of tweets.txt after you have performed the Data Munging tasks in Task 1.

You do not need to submit a PDF of your Excel workbook.

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 2016 for Windows.

In addition to late marks (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.