Microsoft

Text Analytics
with R and Azure
Machine Learning

Contents

Overview

Summary

This lab explores text analytics and R integration with Azure Machine Learning.

It will walk through loading data from an external source, using R scripts in ML Studio, and common text analytics tasks and visualizations.

Business Case

Social media has become a very influential platform for companies, consumers, and professionals to express ideas and opinions, market new products and advertise sales, or share any other important news and information. Most social media sites include keywords or hashtags users can post related content to. If companies can access and perform advanced analytics on the keyword posts that are relevant to them, they can learn things such as customer sentiment, related products and companies, and who is buying products and where from.

For this lab, you will be working with real Twitter data pulled from a Twitter API. The data includes real Tweets that used the hashtag, Azure

The R language has an expansive collection of packages and functions for advanced text mining and analytics. The lab will use R scripts that will be executed in ML Studio. These scripts will perform data preparation, exploration, and visualization tasks common to text mining. The end result will be a visualization that provides context to frequently used terms in the analyzed Tweets.

Learning Objectives

Upon completing this lab, you will have hands-on experience with the following functions and concepts related to Azure Machine Learning and R:

- Reading data from Azure Blob Storage into an experiment in ML Studio

- Executing R Scripts in ML Studio

    o Connecting input data to R Scripts in ML Studio

    o Visualizing data outputs from an R Script in ML Studio

    o Visualizing charts and graphs from an R Script in ML Studio

- Performing common text analytics tasks in R

    o Preparing/Cleaning unstructured text data

    o Aggregating/extracting terms from unstructured data

    o Visualizing term frequency with a Word Cloud

Lab Requirements/Prerequisites

- A Microsoft account is required to access an Azure Machine Learning workspace. If you don't already have a Microsoft account, you can obtain one for free by following the link below:

    https://www.microsoft.com/en-us/account/default.aspx
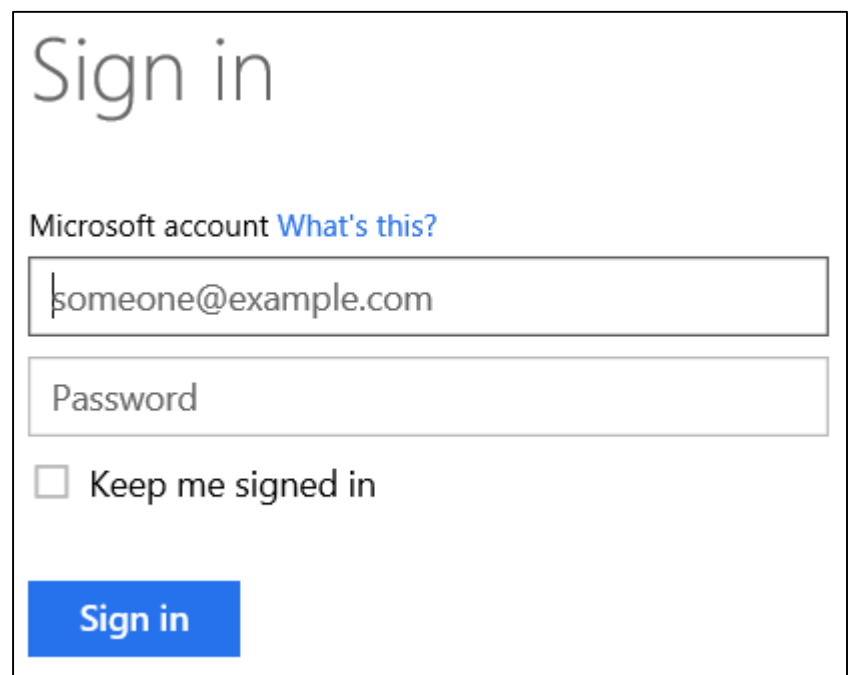
Create a New Experiment

Create/Access an Azure Machine Learning Workspace

To get started, you will need to create and log in to a free Azure Machine Learning workspace. A workspace is like an all-inclusive development environment with the tools to create, manage, and publish machine learning models.

1. Open **a web browser** using the shortcut on the desktop taskbar.

2. Go to the **ML Studio** website by typing **http://studio.azureml.net** in the address bar.
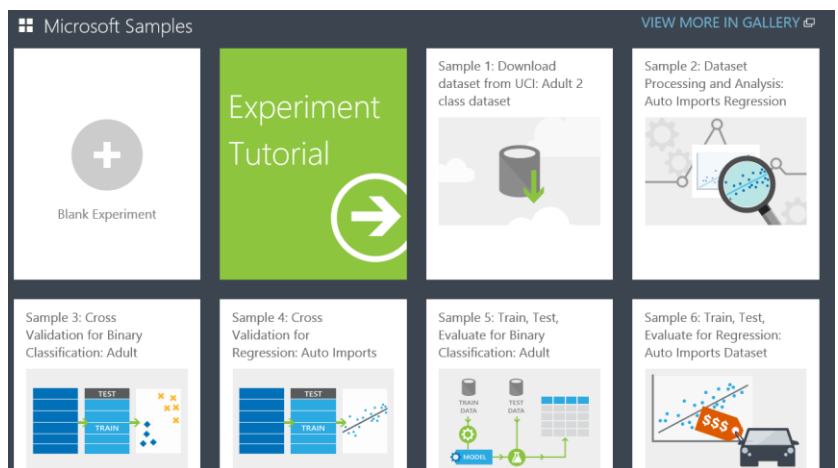
3. Click **Sign In** on the top right corner of the web page.
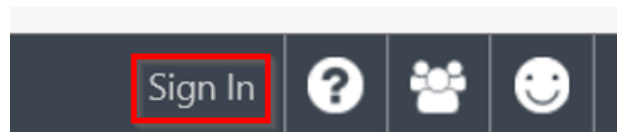
4. Enter the **email address** and **password** associated with your Microsoft ID, and click the **Sign In** button.
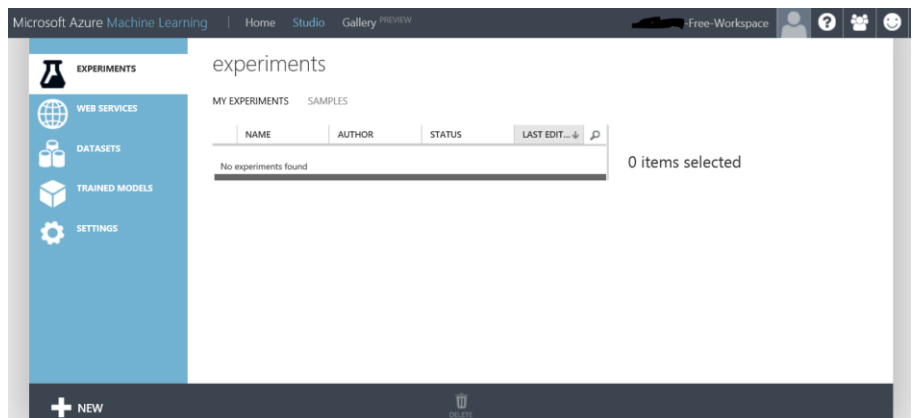
5. If upon logging in, a **Welcome** video is displayed (usually displays on the first login), click the **X** at the top right of the video to close it.

Welcome!
Here is an overview video to get you started.

SHOW ME THE BASICS ✓
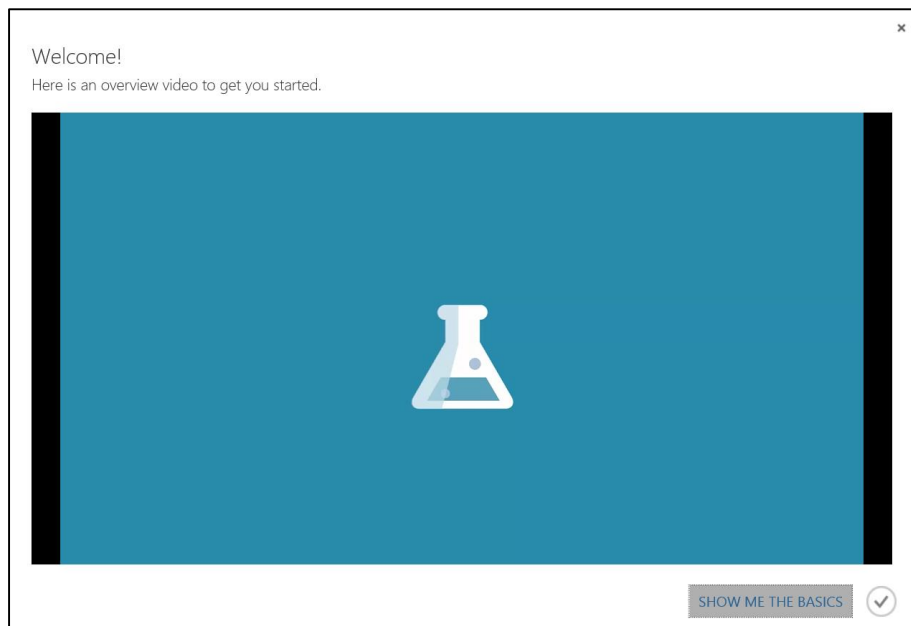
6. If the **Microsoft Samples** dialogue box is displayed (usually displays on the first login), go ahead and close it by choosing the **X** in the top right corner of the pane.

Microsoft Samples   VIEW MORE IN GALLERY

Blank Experiment

Experiment Tutorial

Sample 1: Download dataset from UCI: Adult 2 class dataset

Sample 2: Dataset Processing and Analysis: Auto Imports Regression

Sample 3: Cross Validation for Binary Classification: Adult

Sample 4: Cross Validation for Regression: Auto Imports

Sample 5: Train, Test, Evaluate for Binary Classification: Adult

Sample 6: Train, Test, Evaluate for Regression: Auto Imports Dataset

You are now logged into the free workspace associated with your Microsoft ID.

Create a Blank Experiment

Next, we will create our first experiment. An experiment is a collection of data, tasks, and machine learning algorithms that make up a model.

1. Click the **NEW** button in the bottom left corner of the page.

2. Make sure **EXPERIMENT** is highlighted in the **NEW** dialogue window, and click the **Blank Experiment** pane.
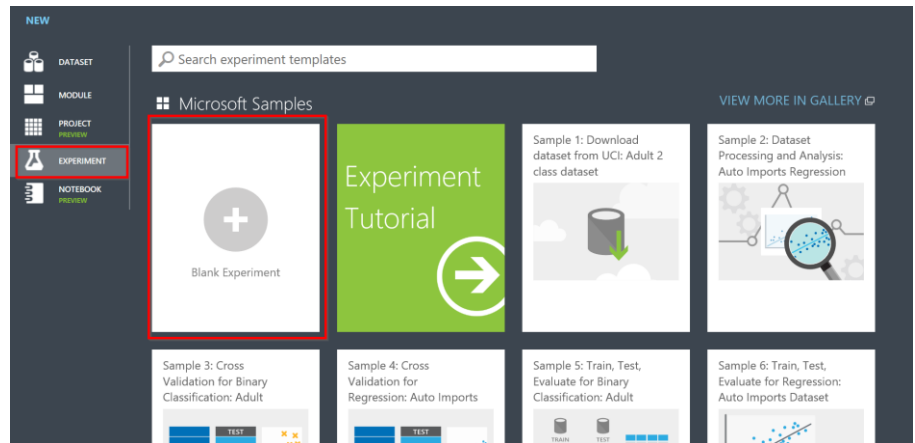
3. At the top of the **canvas**, highlight and delete the text that reads **Experiment created on…**, and replace it with **Lab – Text Mining with R.**
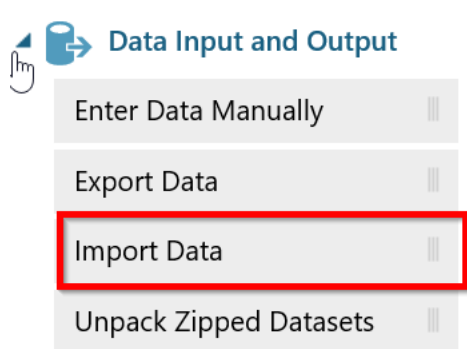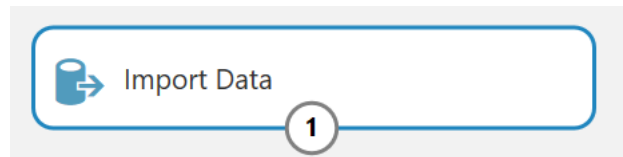
Read in External Data

Get Data from Azure Blob Storage

The data you will use in this lab is stored in Azure Blob storage. The next series of steps will pull this data into ML Studio so you can work with it.
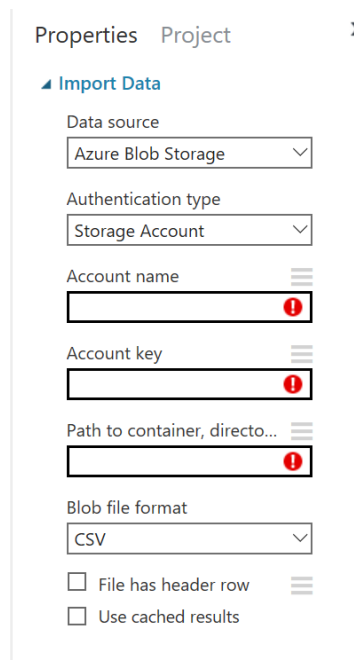
1. In the **modules** pane click and expand **Data Input and Output.**

2. Click and drag the **Import Data** module onto the canvas.

Notice the parameters in the Properties pane for the Reader module. We will modify these to pull a specific Blob from Azure Blob Storage.

3. Enter/Change the following values in the **Properties** pane for the **Reader** module:

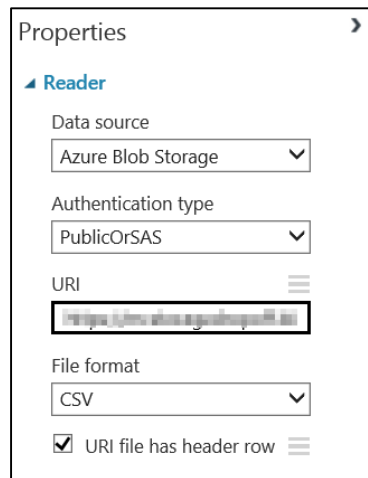Data source: **Azure Blob Storage**

Authentication type: **PublicOrSAS**

URI:
**http://bgmslabs.blob.core.windows.net/machine
learning/tweetsdataset.csv**

File format: **CSV**

Check the **URI has header row** checkbox.

**N.B WHEN COPYING-AND-PASTING THE URI FROM
THIS PDF, DOUBLE CHECK THAT NO SPACE
CHARACTER (EITHER " " OR %20) GETS ADDED INTO
THE URI.**

4. Click **RUN** at the bottom of the canvas to execute the experiment
    and read in the data from Azure Blob Storage.

5. Click the **output** port on the **Import Data** module, and select
    Visualize on the displayed menu.
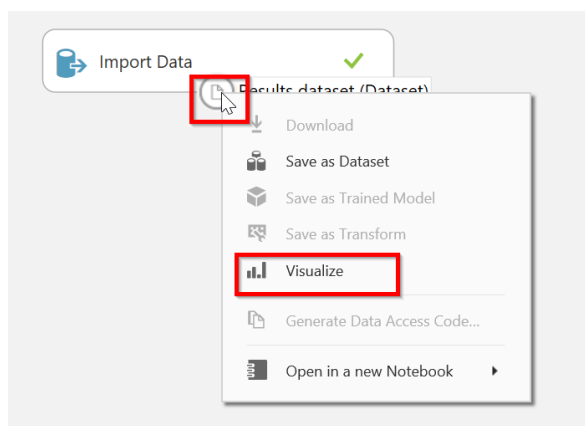
The input data includes 106 rows. Each row includes a Twitter username and the text from a Tweet the user Tweeted.

6. Click the **X** in the top right corner of the Visualize dialogue box to close it.
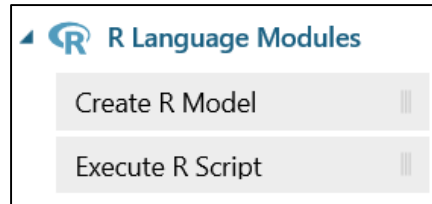
7. Click **SAVE** to save the experiment.
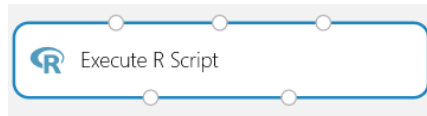
Use R Scripts in the Experiment

The Execute R Script Module

Next, you will begin working with the Execute R Script module in ML Studio. The Execute R Script module allows custom R scripts to be executed in the experiment.
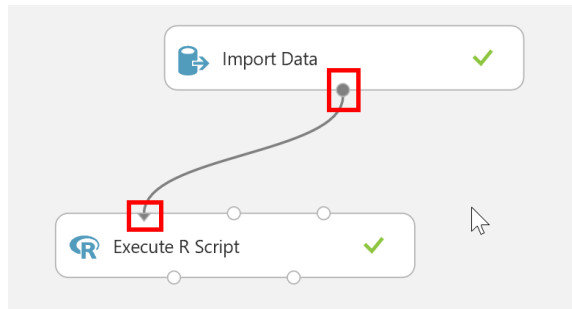
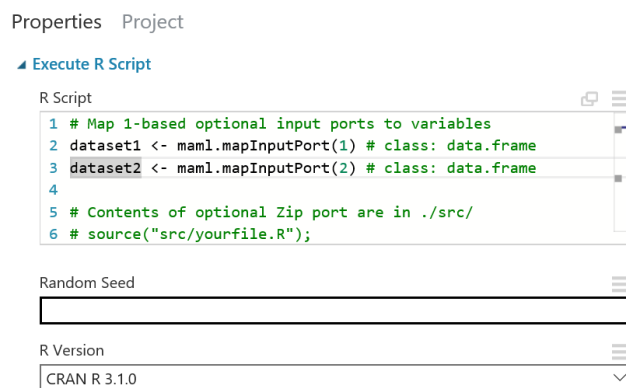1. In the **Modules** pane, click and expand **R Language Modules**.

2. Click and drag the **Execute R Script** module onto the canvas under the **Import Data** module.

3. Connect the **output** port from the **Import Data** module to the **first input** port on the **Execute R Script** module.

Notice the R Script editor in the Properties pane.

Here you can select either CRAN R or Microsoft R Open (MRO). MRO is an enhanced distribution of R – see here for more details.

4. Click anywhere in the **R Script** editor.

```
R Script
 1  # Map 1-based optional input ports to variables
 2  dataset1 <- maml.mapInputPort(1) # class: data.frame
 3  dataset2 <- maml.mapInputPort(2) # class: data.frame
 4
 5  # Contents of optional Zip port are in ./src/
 6  # source("src/yourfile.R");
 7  # load("src/yourData.rdata");
 8
 9  # Sample operation
10  data.set = rbind(dataset1, dataset2);
11
12  # You'll see this output in the R Device port.
13  # It'll have your stdout, stderr and PNG graphics device(s).
14  plot(data.set);
15
16  # Select data.frame to be sent to the output Dataset port
17  maml.mapOutputPort("data.set");
```

You can delete and edit any and all parts of the default R script.

Notice a few important elements of the default script:

    a. The first 2 lines of code map the first and second input ports of the Execute R Script module to data frame objects.

```
1  # Map 1-based optional input ports to variables
2  dataset1 <- maml.mapInputPort(1) # class: data.frame
3  dataset2 <- maml.mapInputPort(2) # class: data.frame
```
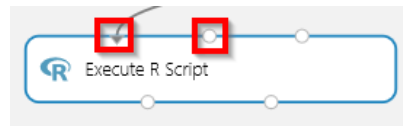

    b. Line 14 uses the plot function. Any graph, chart, or other visualization created in your script will automatically output to the right output port of the Execute R Script module.

```
14  plot(data.set);
```
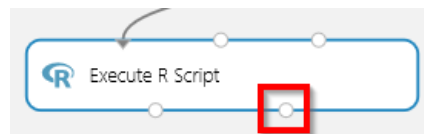

    c. The final line of code maps a data frame object to the left output port of the Execute R Script module.

```
# Select data.frame to be sent to the output Dataset port
maml.mapOutputPort("data.set");
```


5. Use the mouse to select all the code in the **R Script** editor and hit the **Delete** key on the keyboard.

6. In the folder where this lab document is contained is an R script called **script1.R** (the code is listed below). Open script1.R and Copy and paste the code contained in the file to the **R Script** editor in Azure ML.

```
library(dplyr)
library(ggplot2)

#set input data to object
Tweets<-as.data.frame(maml.mapInputPort(1))

#summarize Tweets
summaryOfTweets <- group_by(Tweets, user) %>%
summarise(numberOfTweets = n()) %>%
arrange(desc(numberOfTweets))
top5<-as.data.frame(head(summaryOfTweets, n = 5))

#visualize summary with bar chart
ggplot(top5, aes(x = reorder(user, numberOfTweets), y =
numberOfTweets, width = 0.5)) +
  geom_bar(stat = "identity", fill = "grey70", colour =
"black") +
  coord_flip() +
  ggtitle("Top users Tweeting with the keyword Azure") +
  xlab("Username") +
  ylab("Number of Tweets")

#output data
maml.mapOutputPort("top5")
```
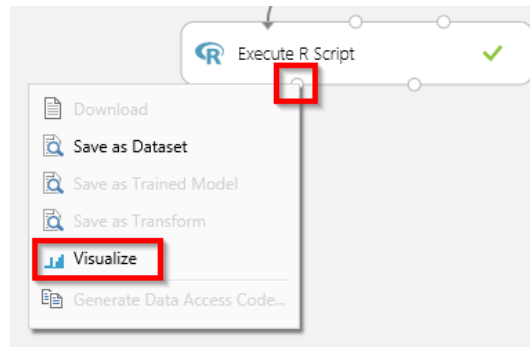
7. Click **RUN** at the bottom of the canvas to execute the experiment with the R script.

   This script loads the Tweets data from the left input port, summarizes the Tweets by user, and then creates a bar chart of the output.
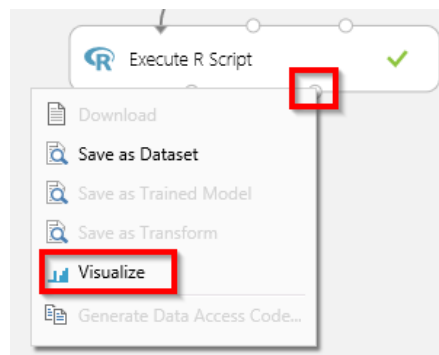
8. Once the experiment has finished executing, click the **left output** port on the **Execute R Script** module and select **Visualize** from the displayed menu.
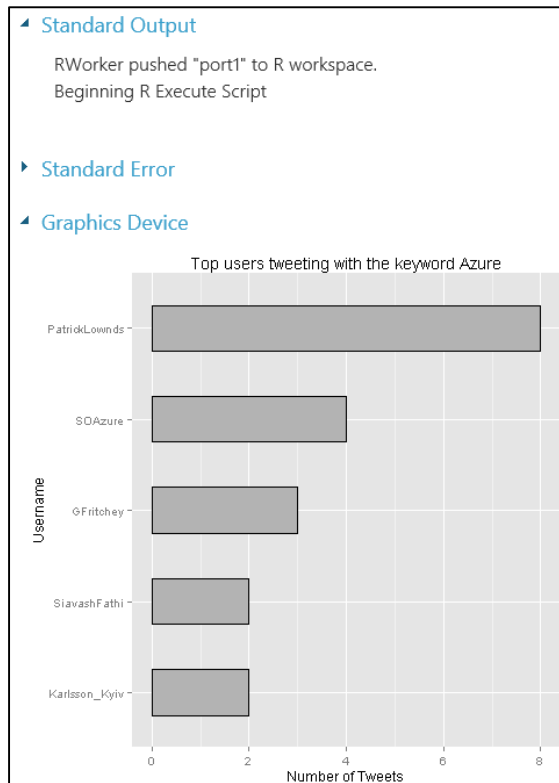
Notice the top 5 Twitter accounts by number of Tweets from our dataset.

9. Click the **X** in the top right corner of the **Visualize** dialogue box to close it.

10. Click the **right output** port on the **Execute R Script** module and select **Visualize** from the displayed menu.

Notice the bar chart that the R script created showing the top 5 users by number of Tweets.
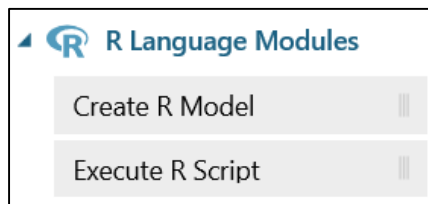
11. Click the **X** in the top right corner of the Visualize dialogue box to close it.
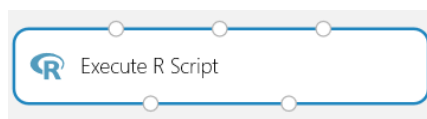
Initial Tweet Preprocessing and Analysis

Next, you will do some term extraction and cleanup on the actual Tweets.
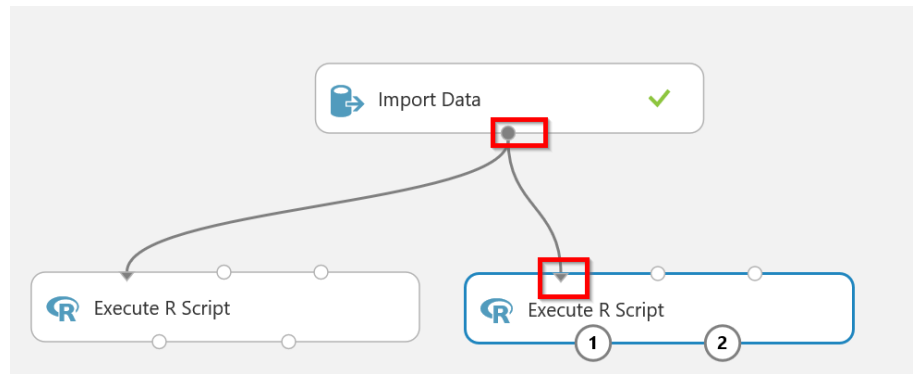
1. Make sure **R Language Modules** is expanded in the **Modules** pane

2. Click and drag the **Execute R Script** module onto the canvas under the **Import Data** module and to the right of the other **Execute R Script** module.

3. Connect the **output** port from the **Import Data** module to the **first input** port on the **Execute R Script** module.

4. Click anywhere in the **R Script** editor to expand it.

5. Use the mouse to select all the code in the **R Script** editor and hit the **Delete** key on the keyboard.

6. In the folder where this lab document is contained is an R script called **script2.R** (the code is listed below). Open script2.R and Copy and paste the code contained in the file to the **R Script** editor in Azure ML

```
library(dplyr)
library(tm)
library(SnowballC)

# set input data to object
Tweets<-as.data.frame(maml.mapInputPort(1))

# text pre-processing 1
myCorpus <- Corpus(VectorSource(Tweets$text))
myCorpus <- tm_map(myCorpus, stripWhitespace)
myCorpus <- tm_map(myCorpus,
content_transformer(tolower))
myCorpus <- tm_map(myCorpus, removePunctuation)

# create matrix of terms with frequency of terms
tdm <- TermDocumentMatrix(myCorpus, control =
list(wordLengths=c(1, Inf)))
m   <-
as.matrix(rowSums(as.matrix(tdm)),rownames.force=NA)
rownames(m) <- NULL
m <- data.frame(Terms=rownames(tdm), Frequency=m,
stringsAsFactors = FALSE)
m <- arrange(m,desc(Frequency))

# output data
maml.mapOutputPort("m")

# visualize term frequency with wordcloud
library(wordcloud)
wordcloud(m[,1], m[,2], random.order = FALSE,
random.color = FALSE, scale = c(10, .5), colors =
c(colors(),"orange"))
```

7. Click **RUN** at the bottom of the canvas to execute the experiment
   with the R script.

   This script uses only the Tweets column from the data in the left
   input port. It processes the Tweets by removing whitespace in
   words, transforming all the words to lowercase, and removing all
   the punctuation. Then it creates a dataset of the different terms in
   the Tweets and the number of occurrences (frequency) of each term.
   Finally, the term frequency is visualized in a word cloud.

8. Once the experiment has finished executing, click the **left output**
   port on the **Execute R Script** module and select **Visualize** from
   the displayed menu.
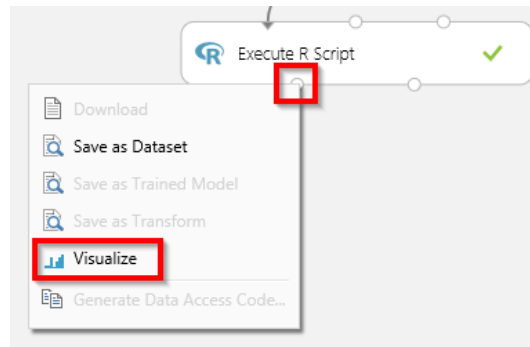
Notice the list of terms and their associated frequencies. There are some relevant terms like "Azure" and "cloud" near the top, but there are also a lot of terms like "a", "to", and "the" which do not offer much analytical value. As you scroll through the list, you will also encounter numbers like 1 and 50 in the terms list that also do not offer much analytical value at this point.

Terms	Frequency
azure	135
rt	36
a	32
the	32
to	30
for	27
in	27
cloud	20
is	20

9. Click the **X** in the top right corner of the **Visualize** dialogue box to close it.

10. Click the **right output** port on the **Execute R Script** module and select **Visualize** from the displayed menu.
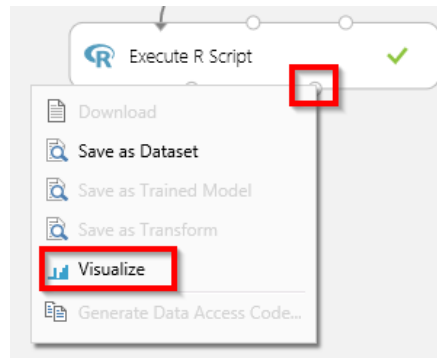
A word cloud is displayed in the Visualize dialogue box. Do not worry if your word cloud does not look exactly like the one shown below, as the R function that creates the word cloud randomly arranges the words each time it runs. In this word cloud, the size of the word represents its relative frequency compared with other words. Again, you will notice words like Azure and Microsoft are bigger, but you also see many words like "in", "with" and "is" cluttering up the cloud.
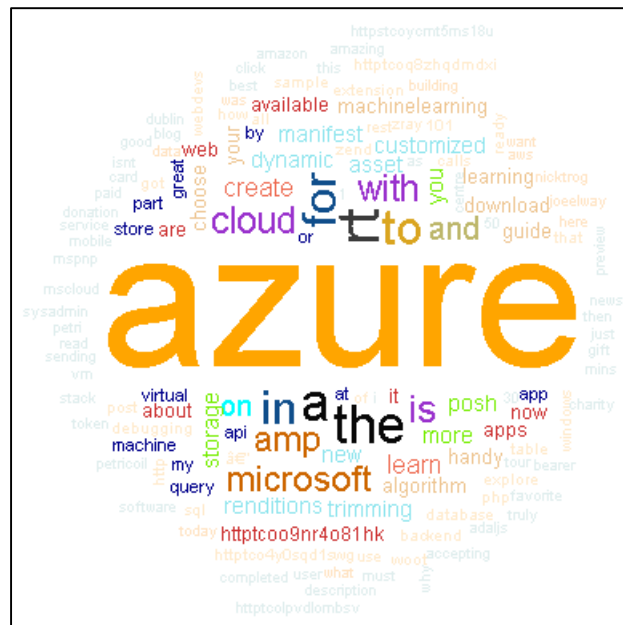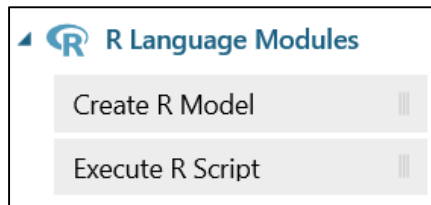

11. Click the **X** in the top right corner of the Visualize dialogue box to close it.

Remove "Stop" Words and Numbers

Next, you will use an R Script that includes additional data prep tasks for removing some of the words and numbers that are less valuable for analytics.
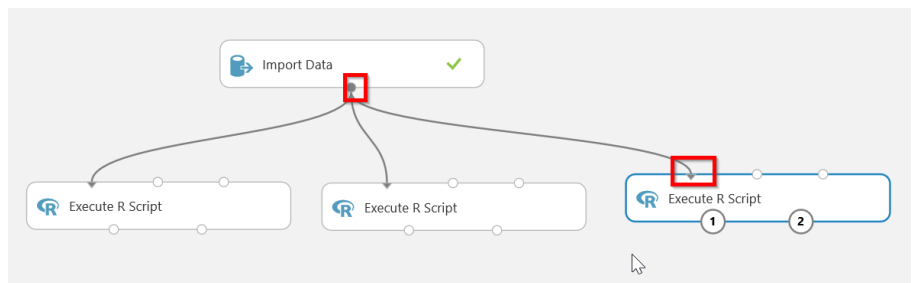
1. Make sure **R Language Modules** is expanded in the **Modules** pane

2. Click and drag the **Execute R Script** module onto the canvas under the **Import Data** module and to the right of the other **Execute R Script** modules.
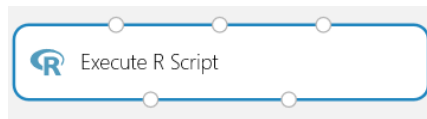
3. Connect the **output** port from the **Import** module to the **first input** port on the **Execute R Script** module.

4. Click anywhere in the **R Script** editor to expand it.

5. Use the mouse to select all the code in the **R Script** editor and hit the **Delete** key on the keyboard.

6. In the folder where this lab document is contained is an R script called **script3.R** (the code is listed below). Open script3.R and Copy and paste the code contained in the file to the **R Script** editor in Azure ML

```
library(dplyr)
library(tm)
library(SnowballC)

# set input data to object
Tweets<-as.data.frame(maml.mapInputPort(1))

# text pre-processing 1
myCorpus <- Corpus(VectorSource(Tweets$text))
myCorpus <- tm_map(myCorpus, stripWhitespace)
myCorpus <- tm_map(myCorpus,
content_transformer(tolower))
myCorpus <- tm_map(myCorpus, removePunctuation)

# text pre-processing 2: remove numbers and stop words
myCorpus <- tm_map(myCorpus, removeNumbers)
myStopWords <- c(stopwords("english"))
myCorpus <- tm_map(myCorpus, removeWords, myStopWords)

# create matrix of terms with frequency of terms
tdm <- TermDocumentMatrix(myCorpus, control =
list(wordLengths=c(1, Inf)))
m   <-
as.matrix(rowSums(as.matrix(tdm)),rownames.force=NA)
rownames(m) <- NULL
m <- data.frame(Terms=rownames(tdm), Frequency=m,
stringsAsFactors = FALSE)
m <- arrange(m,desc(Frequency))

# output data
maml.mapOutputPort("m")

# visualize term frequency with wordcloud
library(wordcloud)
wordcloud(m[,1], m[,2], random.order = FALSE,
random.color = FALSE, scale = c(10, .5), colors =
c(colors(),"orange"))
```

7. Click **RUN** at the bottom of the canvas to execute the experiment.

   This script builds on the previous script you used. It adds a series of processing steps for removing numbers and common words that are not relevant for analytics (typically called *stop words*).

8. Once the experiment has finished executing, click the **left output** port on the **Execute R Script** module and select **Visualize** from the displayed menu.
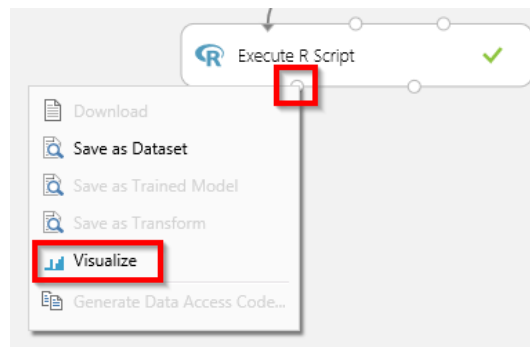
Notice the new list of words does not include the "stop" words ("a", "the", etc.) or the number terms.

Terms	Frequency
azure	135
rt	36
cloud	20
amp	19
microsoft	19
create	12
learn	12
posh	10
storage	10
asset	9
customized	9
dynamic	9
manifest	9
new	9

9. Scroll through the list and find the term "**learn**".

microsoft	19
create	12
learn	12
posh	10
storage	10
asset	9

Notice it has 12 occurrences.

10. Scroll a little further down the list until you find "**learning**".

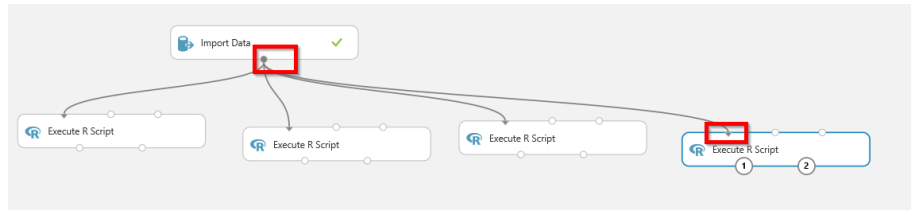| guide | 8 |
handy	8
learning	8
machinelearning	8
apps	7
available	7

Notice it has 8 occurrences.

"Learn", "Learned", and "Learning" are all derivatives of the same word. Later, we might want to group all derivatives of the same word together and sum up their occurrences as a single term (resulting in 20 total occurrences of the base word "learn").

11. Click the **X** in the top right corner of the **Visualize** dialogue box to close it.

12. Click the **right output** port on the **Execute R Script** module and select **Visualize** from the displayed menu.
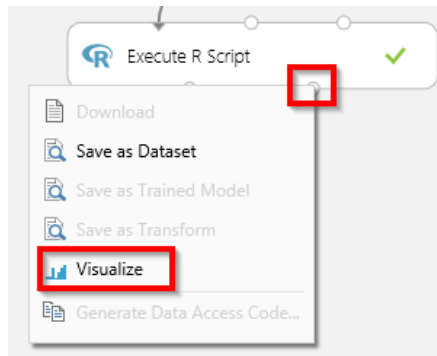
Notice the new word cloud highlights many more relevant terms for analysis than the previous word cloud.

13. Click the **X** in the top right corner of the **Visualize** dialogue box to close it.

Combine Words with Stemming

As we saw in the previous set of steps, there are instances of words we would like to combine so they are counted as 1 term. In this final set of steps, we will use a method called stemming to do this.

1. Make sure **R Language Modules** is expanded in the **Modules** pane

2. Click and drag the **Execute R Script** module onto the canvas under the **Import Data** module and to the right of the other **Execute R Script** modules.

3. Connect the **output** port from the **Import Data** module to the **first input** port on the **Execute R Script** module.

4. Click anywhere in the **R Script** editor to expand it.

```
R Script
 1  # Map 1-based optional input ports to variables
 2  dataset1 <- maml.mapInputPort(1) # class: data.frame
 3  dataset2 <- maml.mapInputPort(2) # class: data.frame
 4
 5  # Contents of optional Zip port are in ./src/
 6  # source("src/yourfile.R");
 7  # load("src/yourData.rdata");
 8
 9  # Sample operation
10  data.set = rbind(dataset1, dataset2);
11
12  # You'll see this output in the R Device port.
13  # It'll have your stdout, stderr and PNG graphics device(s).
14  plot(data.set);
15
16  # Select data.frame to be sent to the output Dataset port
17  maml.mapOutputPort("data.set");
```

5. Use the mouse to select all the code in the **R Script** editor and hit the **Delete** key on the keyboard.

6. In the folder where this lab document is contained is an R script called **script4.R** (the code is listed below). Open script4.R and Copy and paste the code contained in the file to the **R Script** editor in Azure ML

```r
library(dplyr)
library(tm)
library(SnowballC)

# set input data to object
Tweets<-as.data.frame(maml.mapInputPort(1))

# text pre-processing 1
myCorpus <- Corpus(VectorSource(Tweets$text))
myCorpus <- tm_map(myCorpus, stripWhitespace)
myCorpus <- tm_map(myCorpus,
content_transformer(tolower))
myCorpus <- tm_map(myCorpus, removePunctuation)

# text pre-processing 2: remove numbers and stop words
myCorpus <- tm_map(myCorpus, removeNumbers)
myStopWords <- c(stopwords("english"))
myCorpus <- tm_map(myCorpus, removeWords, myStopWords)

# text pre-processing 3: stem words
myCorpusCopy <- myCorpus
myCorpus <- tm_map(myCorpus, stemDocument)

# convert stems to real words
stemCompletion2 <- function(x, dictionary) {
  x <- unlist(strsplit(as.character(x), " "))
  x <- x[x != ""]
  x <- stemCompletion(x, dictionary = dictionary)
  x <- paste(x, sep="", collapse=" ")
  PlainTextDocument(stripWhitespace(x))
}
myCorpus2 <- lapply(myCorpus, stemCompletion2,
dictionary=myCorpusCopy)
myCorpus <- Corpus(VectorSource(myCorpus2))

# create matrix of terms with frequency of terms
tdm <- TermDocumentMatrix(myCorpus, control =
list(wordLengths=c(1, Inf)))
m   <-
as.matrix(rowSums(as.matrix(tdm)),rownames.force=NA)
rownames(m) <- NULL
m <- data.frame(Terms=rownames(tdm), Frequency=m,
stringsAsFactors = FALSE)
m <- arrange(m,desc(Frequency))

# output data
maml.mapOutputPort("m")

# visualize term frequency with wordcloud
library(wordcloud)
wordcloud(m[,1], m[,2], random.order = FALSE,
random.color = FALSE, scale = c(10, .5), colors =
c(colors(),"orange"))
```

7. Click **RUN** at the bottom of the canvas to execute the experiment.

This code builds on the previous scripts, but it also includes steps for stemming words. Stemming is the process of breaking a word down to its root, and then combining it with other words that resulted in the same root.

As an example, the words "like", "liked", "likes", and "liking" would all be stemmed to "lik". After being stemmed and combined, the stem word "lik" would usually be transformed back to the root word. In this example, the root word would be "like". This process would result in all derivatives of "like" (like, liked, likes, and liking) being counted as a single term.

8. Once the experiment has finished executing, click the **left output** port on the **Execute R Script** module and select **Visualize** from the displayed menu.
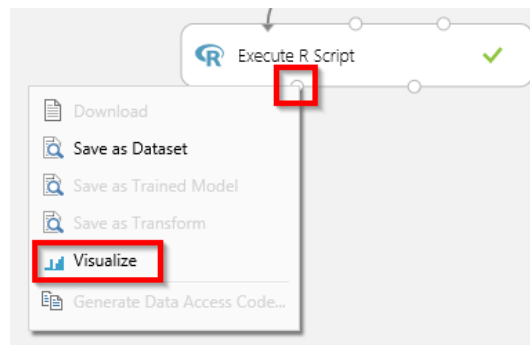
Notice the term "learn" is now a top 4 term with 20 occurrences (the sum of the original "learn" and "learning").

Terms	Frequency
azure	135
rt	36
cloud	20
learn	20
amp	19
microsoft	19

9. Click the **X** in the top right corner of the **Visualize** dialogue box to close it.

10. Click the **right output** port on the **Execute R Script** module and select **Visualize** from the displayed menu.

Notice the term "learn" is more dominant than in previous word clouds.

11. Click the **X** in the top right corner of the **Visualize** dialogue box to close it.

12. Sign out of your workspace by clicking the **profile picture** at the top right of the page and selecting **Sign Out** from the displayed menu.

Conclusion

This concludes the *Text Analytics with R and Azure Machine Learning* lab. In this lab, you loaded data from an external source, executed R scripts in ML Studio, and performed common text analytics tasks and visualizations.

The methods and scripts used in this lab can be extended further to find Tweets with specific terms, related products or ideas, or even related websites and blog posts.

Terms of Use

USE OF THE VIRTUAL LAB, OR SUITABILITY OF THE INFORMATION CONTAINED IN THE VIRTUAL LAB FOR ANY PURPOSE.

DISCLAIMER

This lab contains only a portion of the features and enhancements in Microsoft Azure Machine Learning. Some of the features might change in future releases of the product.