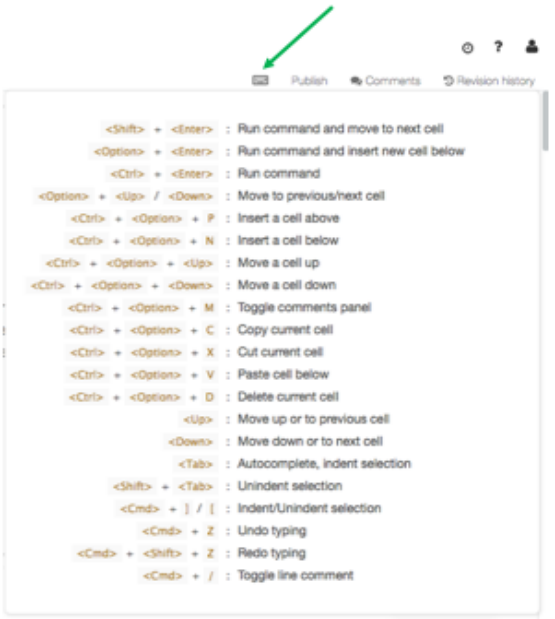
You can bring up a popup of keyboard shortcuts by clicking on the keyboard icon at the top of a notebook.



Learning Apache Spark

For this week, there is one tutorial (cs105_lab1a_spark_tutorial) and one lab to complete (cs105_lab1b_word_count). **ONLY the lab (cs105_lab1b_word_count) will be submitted to the autograder.**

The word count lab is due September 10, 2016 at 23:59 UTC.

(1a) SPARK TUTORIAL

This tutorial will teach you how to use Apache Spark, a framework for large-scale data processing, within a notebook. Many traditional frameworks were designed to be run on a single computer. However, many datasets today are too large to be stored on a single computer, and even when a

dataset can be stored on one computer (such as the datasets in this tutorial), the dataset can often be processed much more quickly using multiple computers. Spark has efficient implementations of a number of transformations and actions that can be composed together to perform data processing and analysis. Spark excels at distributing these operations across a cluster while abstracting away many of the underlying implementation details. Spark has been designed with a focus on scalability and efficiency. With Spark you can begin developing your solution on your laptop, using a small dataset, and then use that same code to process terabytes or even petabytes across a distributed cluster.

During this tutorial we will cover:

- *Part 1:* Basic notebook usage and Python integration
- *Part 2:* An introduction to using Apache Spark with the PySpark SQL API running in a notebook
- *Part 3:* Using DataFrames and chaining together transformations and actions
- *Part 4:* Python Lambda functions and User Defined Functions
- *Part 5:* Additional DataFrame actions
- *Part 6:* Additional DataFrame transformations
- *Part 7:* Caching DataFrames and storage options
- *Part 8:* Debugging Spark applications and lazy evaluation

The following transformations will be covered:

- select(), filter(), distinct(), dropDuplicates(), orderBy(), groupBy()

The following actions will be covered:

- first(), take(), count(), collect(), show()

Also covered:

- cache(), unpersist()

Note that, for reference, you can look up the details of the relevant methods in Spark's PySpark SQL API.

The tutorial is not a graded exercise. You do not submit the tutorial to the course autograder or edX.

(1b) WORD COUNT LAB: BUILDING A WORD COUNT APPLICATION

This lab will build on the techniques covered in the Spark tutorial to develop a simple word count application. The volume of unstructured text in existence is growing dramatically, and Apache Spark is an excellent tool for analyzing this type of data. In this exercise, we will write code that calculates the most common words in the Complete Works of William Shakespeare retrieved from Project Gutenberg. The code you write could also be scaled to larger applications, such as finding the most common words in Wikipedia.

During this lab we will cover:

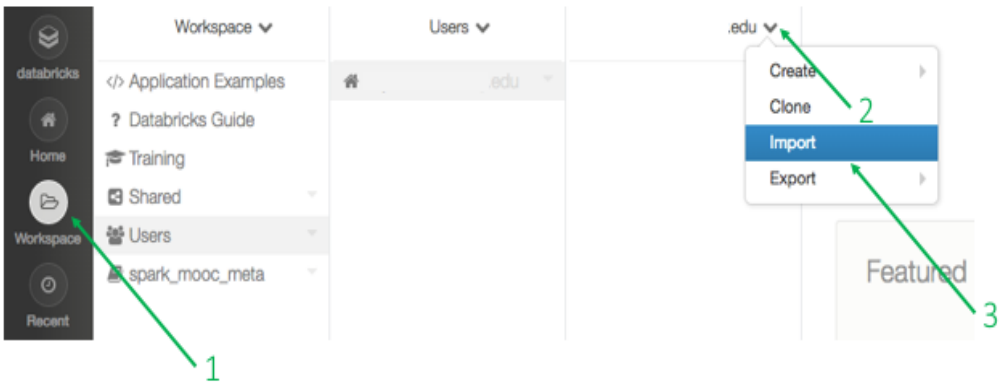
- *Part 1:* Creating a base DataFrame and performing operations
- *Part 2:* Counting with Spark SQL and DataFrames
- *Part 3:* Finding unique words and a mean value
- *Part 4:* Apply word count to a file

The word count lab is a graded exercise, and you should submit the word count lab (cs105x_lab1b) to the course autograder and edX.

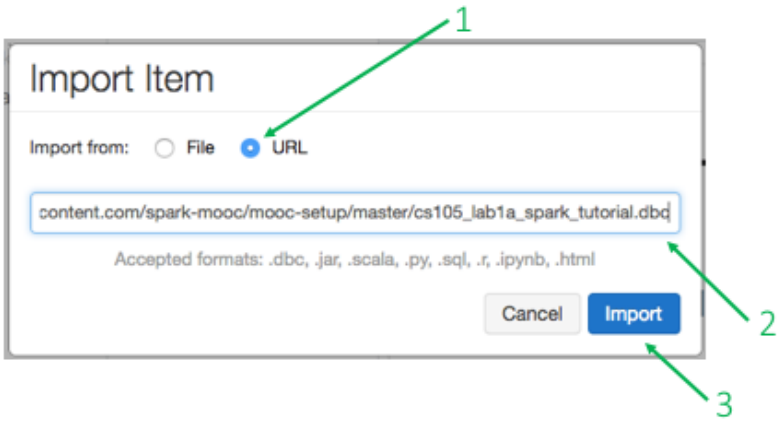
(1a) Spark Tutorial

Here are the instructions for the Spark Tutorial

1. Login to your Databricks Community Edition account.
2. Import the lab 1a Spark Tutorial Notebook by selecting "Workspace", clicking on the dropdown next to your username, and selecting "Import".



3. Select the URL radio button, copy the address of the Spark Tutorial notebook (https://raw.githubusercontent.com/spark-mooc/mooc-setup/master/cs105_lab1a_spark_tutorial.dbc), paste the copied Spark Tutorial notebook link into the text box, and click "Import" to import the Spark Tutorial notebook.



4. Follow the instructions in the notebook to run it.
5. When you have finished with the notebook, proceed to the next section to work on the Word Count lab notebook.

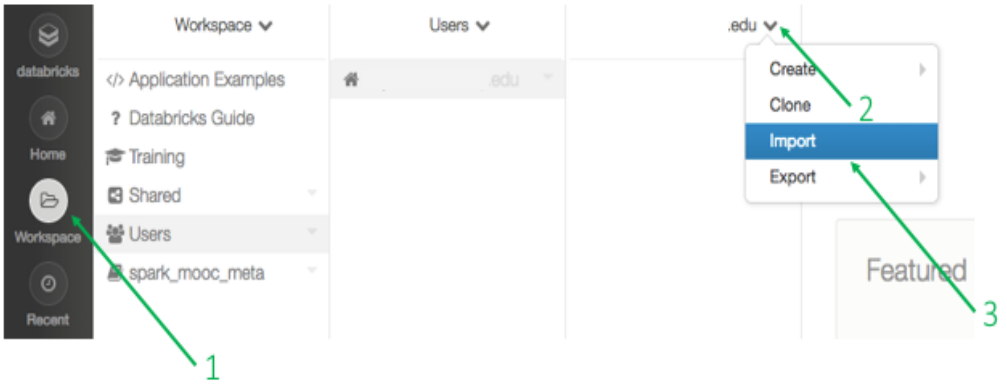
Notes:

1. If you get the error "ImportError: No module named databricks_test_helper" - you should re-read last week's "Part 1: Attach and test class helper library" section of the cs105_lab0 notebook and follow the steps closely. You can also see how to add the helper library in last week's video.
2. You can also download the notebook to your computer and upload it to your Databricks Community Edition workspace using the same Import dialog as above and dragging the downloaded file into the highlighted area. If you download the notebook, make sure that the file extension is .dbc. If the download adds an extension (e.g. ".txt"), rename the file so that the extension is just .dbc. You can also download the Python source for the Spark Tutorial here.

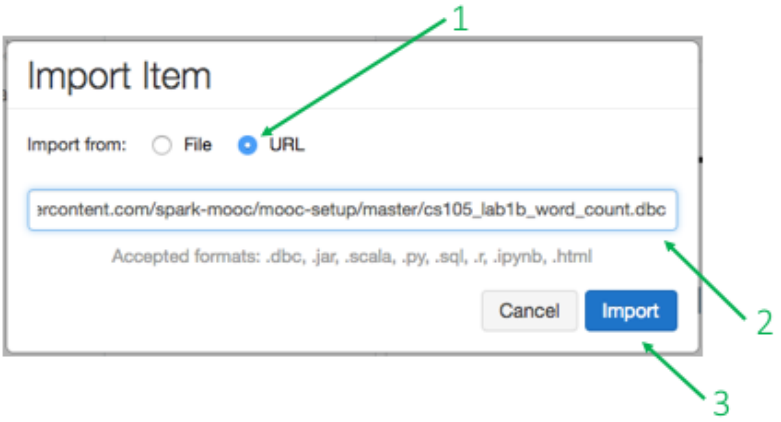
(1b) Word Count Lab

Here are the instructions for the Word Count lab:

1. Login to your Databricks Community Edition account.
2. Import the lab 1b Word Count Notebook by selecting "Workspace", clicking on the dropdown next to your username, and selecting "Import".



3. Select the URL radio button, copy the address of the Word Count notebook (https://raw.githubusercontent.com/spark-mooc/mooc-setup/master/cs105_lab1b_word_count.dbc), paste the copied Word Count notebook link into the text box, and click "Import" to import the Word Count notebook.



4. Follow the instructions in the notebook to run it.
5. When you have finished with the notebook, proceed to the next module to submit the Word Count notebook to the course autograder.

Notes:

1. If you get the error "ImportError: No module named databricks_test_helper" - you should re-read last week's "Part 1: Attach

and test class helper library" section of the cs105_lab0 notebook and follow the steps closely. You can also see how to add the helper library in last week's video.

2. You can also download the notebook to your computer and upload it to your Databricks Community Edition workspace using the same Import dialog as above and dragging the downloaded file into the highlighted area. If you download the notebook, make sure that the file extension is .dbc. If the download adds an extension (e.g. ".txt"), rename the file so that the extension is just .dbc. You can also download the Python source for the Word Count lab here.