



Introduction to Big Data

Data Boot Camp
Lesson 22.1





Instructor Demonstration
Welcome Class

Class Objectives:

By the end of today's class you will be able to:



Identify the pieces of the Hadoop ecosystem.



Identify the differences and similarities between Hadoop and Spark.



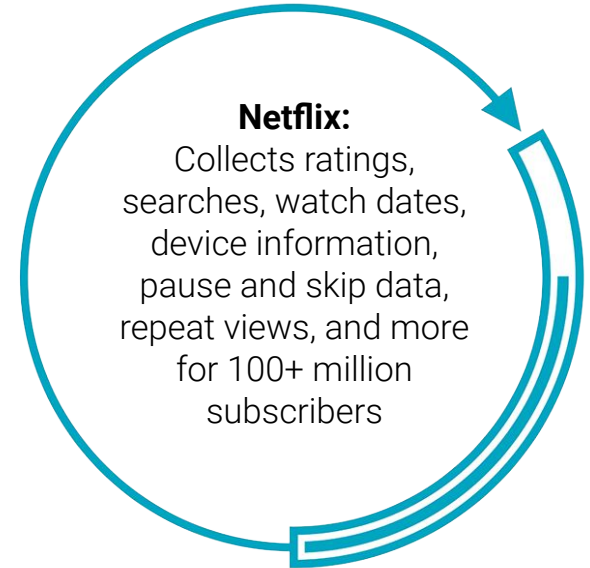
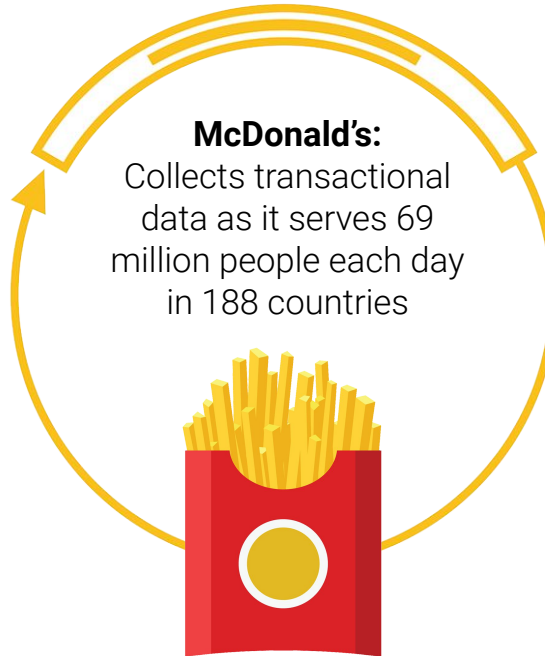
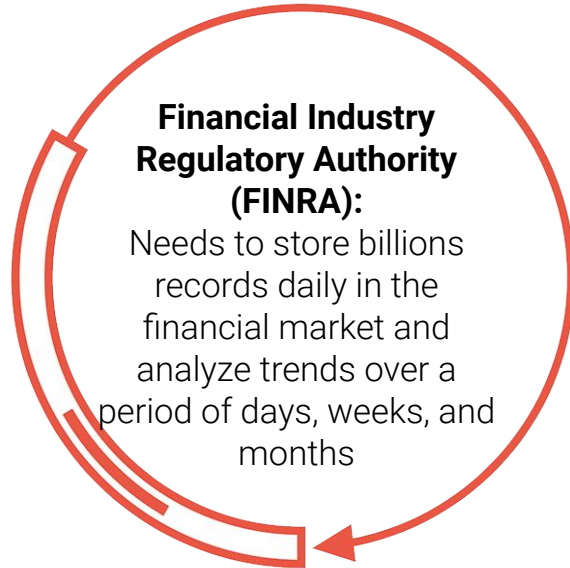
Write MapReduce jobs locally with mrjob.



Manipulate data using PySpark DataFrames.

Instructor Do: Welcome Class

Big Data, Big Problems



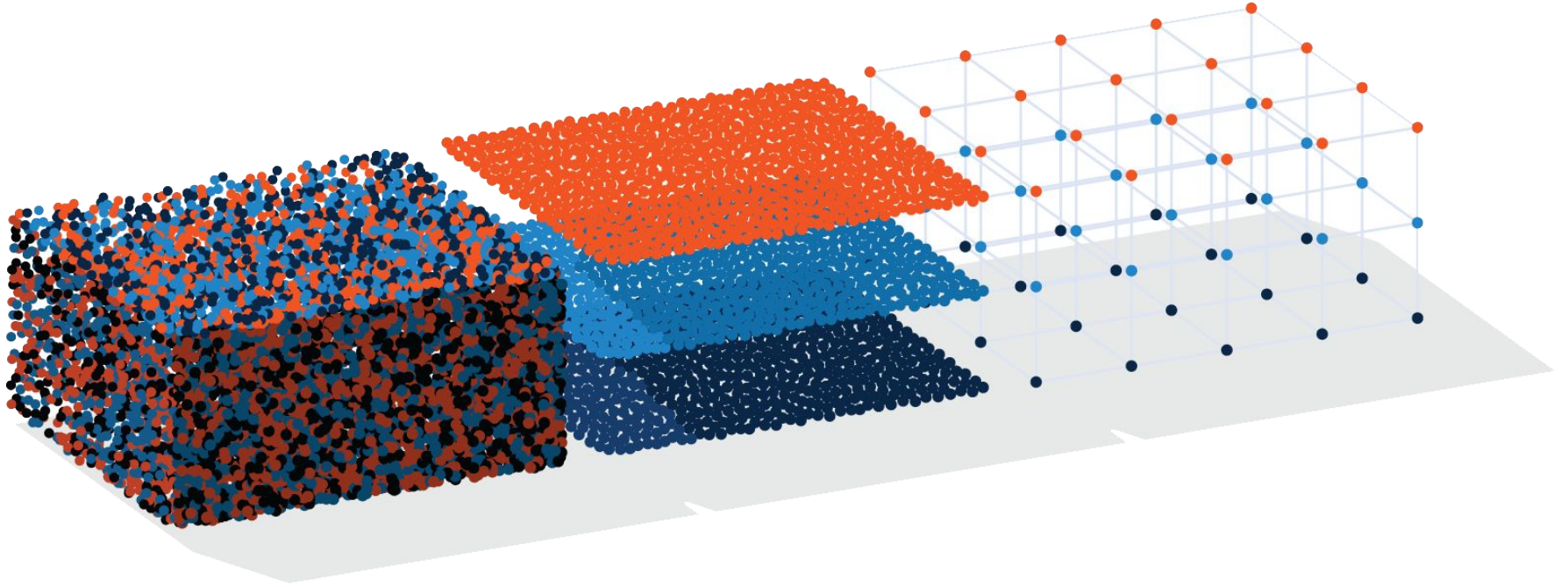


Instructor Demonstration

Intro to Big Data

Instructor Do: Intro to Big Data

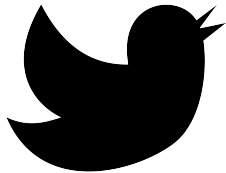
What Is Big Data?



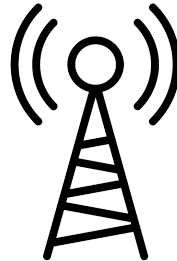
Instructor Do: Intro to Big Data

What Is Big Data?

Big data includes stock exchange data, emails, and social media posts such as Facebook statuses and tweets.



It also includes lesser known things like supply chains, barcodes, and cell towers.

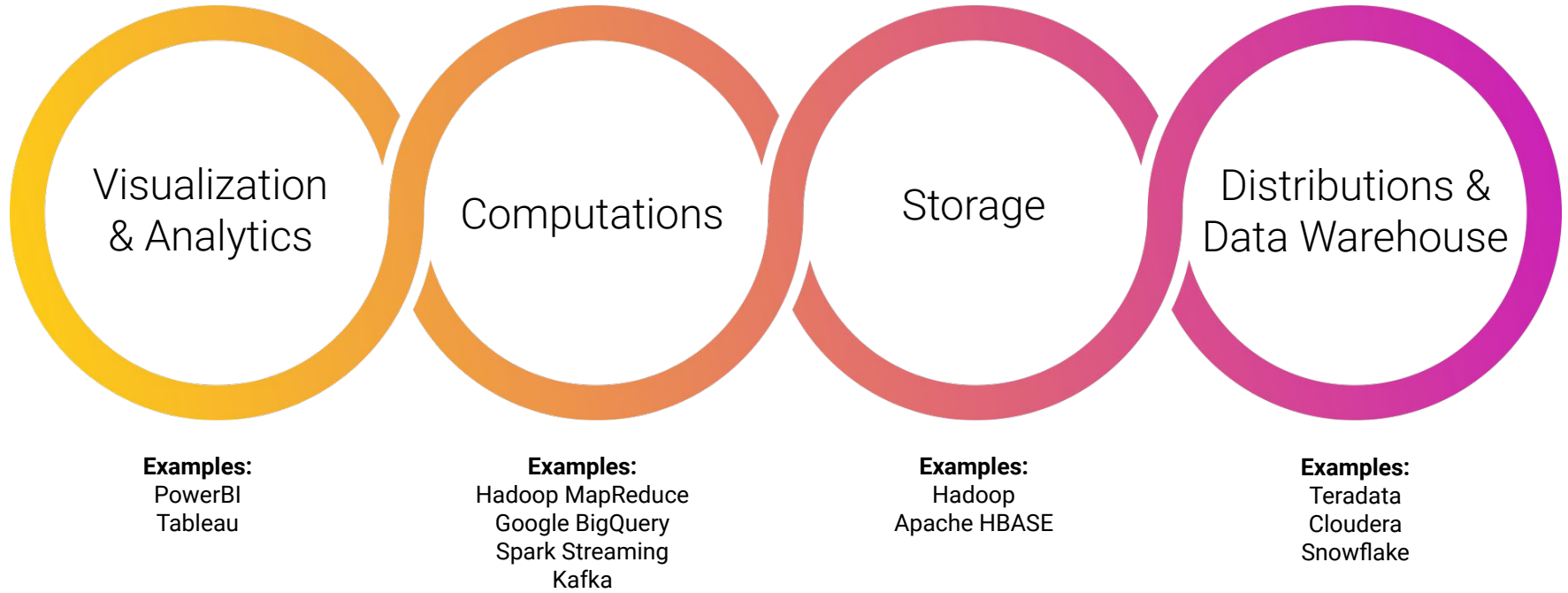


Big data has always existed, there just hasn't been a way to gather and analyze it.



Instructor Do: Intro to Big Data

Big data Overview





**What are some issues that
you might encounter when
dealing with extremely
large datasets?**

Instructor Do: Intro to Big Data

Issues you might encounter when dealing with extremely large datasets:



Need a place to store massive amounts of data.



Need a way to access data quickly.



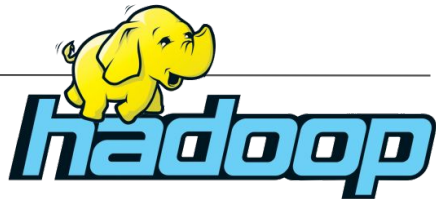
Need backups for hardware failure.



Need ways to analyze data quickly.

Instructor Do: Intro to Big Data

Hadoop Overview



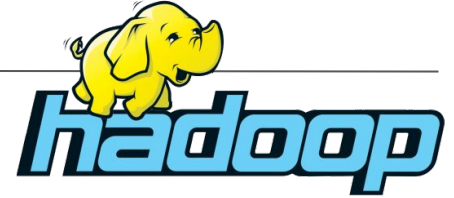
“The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures.”

—Hadoop website

Hadoop was named after a stuffed yellow elephant belonging to the son of Doug Cutting, the Hadoop creator.

Instructor Do: Intro to Big Data

Hadoop Distributed File System (HDFS)



HDFS is a file system that is used to store data across server clusters, and is **scalable**, **fault-tolerant**, and **distributed**.



Instructor Do: Intro to Big Data

Four Vs of Big Data

01

Volume: Size of the data (terabytes of stock data)

02

Velocity: How quickly data is coming in (car sensors sending information every second)

03

Variety: Different forms of data (social media posts, comments, photos, etc.)

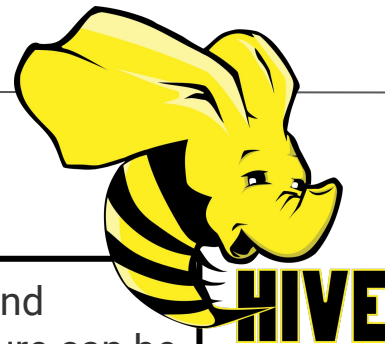
04

Veracity: Uncertainty of data (social media data may not be precise, come from bots, etc.)

Instructor Do: Intro to Big Data

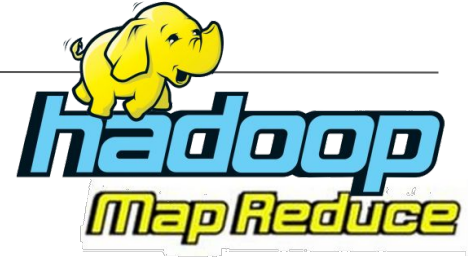
The Apache Hive

The Apache Hive™ data warehouse software facilitates reading, writing, and managing large datasets residing in distributed storage using SQL. Structure can be projected onto data already in storage. A command line tool and JDBC driver are provided to connect users to Hive.



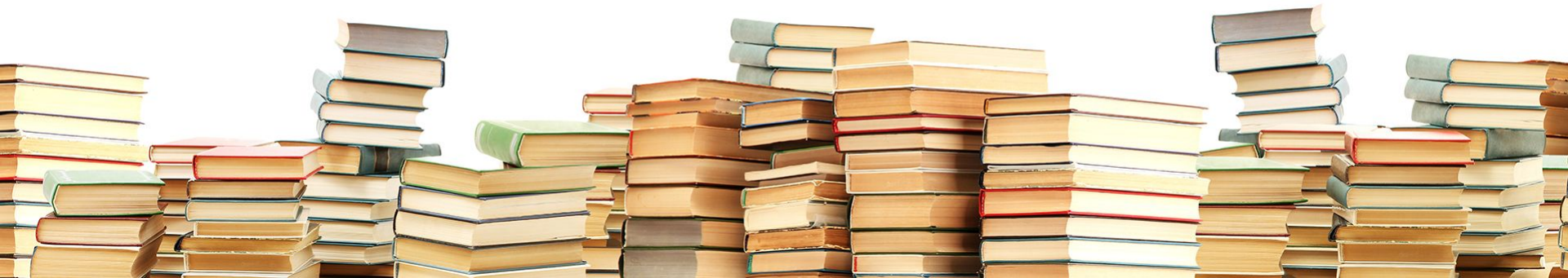
Instructor Do: Intro to Big Data

Example: Counting the Number of Books in a Library



Map: You count this half of the library, and I'll count the other.

Reduce: We get together and add up our counts.





Instructor Demonstration

Intro to MapReduce with mrjob

Instructor Do: Intro to MapReduce with mrjob



What is MapReduce job?

It was originally the product of research done by Google.



Designed to solve a single problem: how to index all the information on the Internet

Instructor Do: Intro to MapReduce with mrjob



What is a **Job**?



A job is defined by a class that inherits from `MRJob`.



This class contains methods that define the steps of your job.



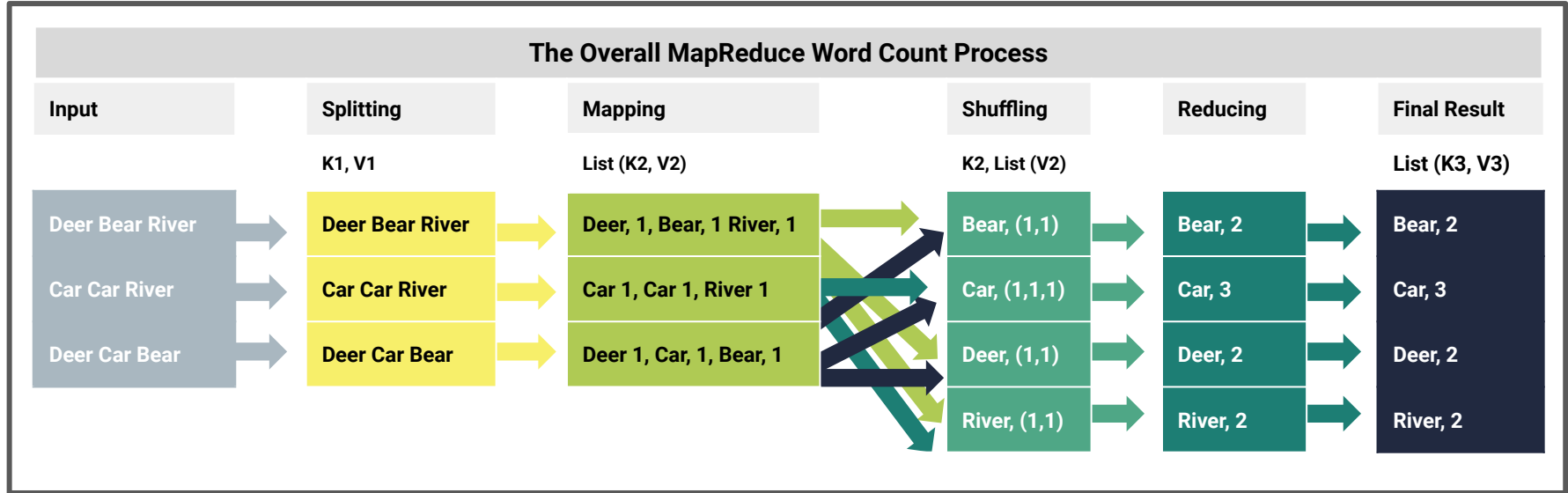
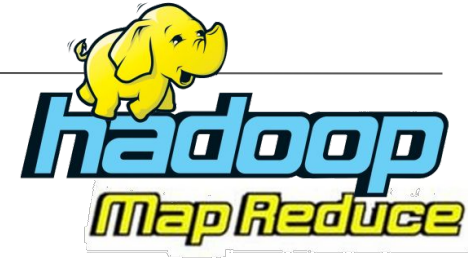
Can translate the job and run it locally or on a Hadoop cluster.



Note: the shuffle step is handled behind the scenes

Instructor Do: Intro to MapReduce with mrjob

Classic Word Count Example



Instructor Do: Intro to MapReduce with mrjob



The **Map** Part of MapReduce



Execute the `Map()` function on data.

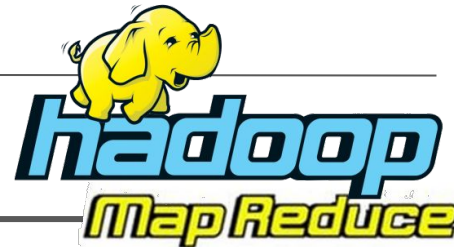


Execute on each node.



Output `<key, value>` pairs on each node.

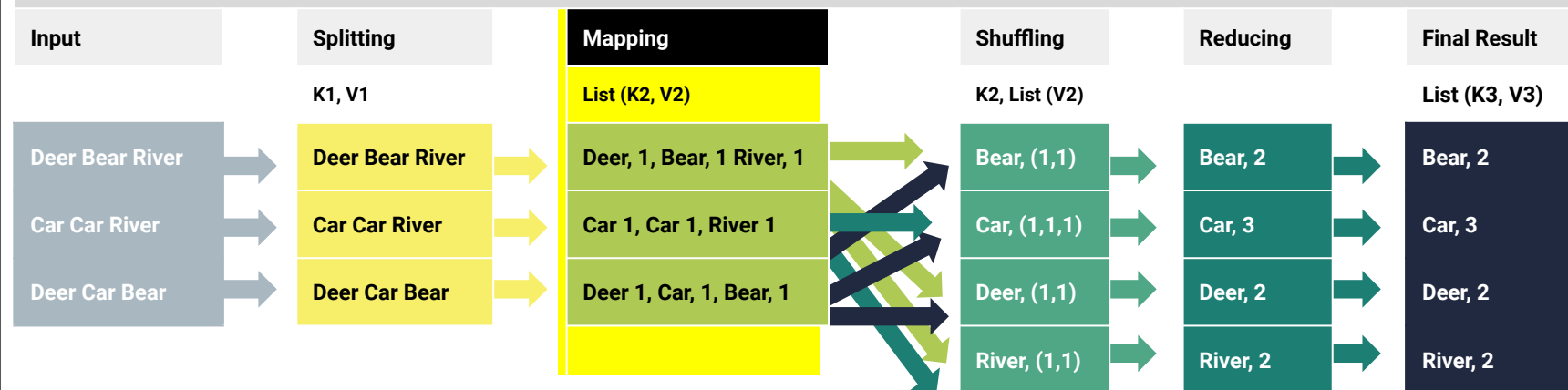
Instructor Do: Intro to MapReduce with mrjob



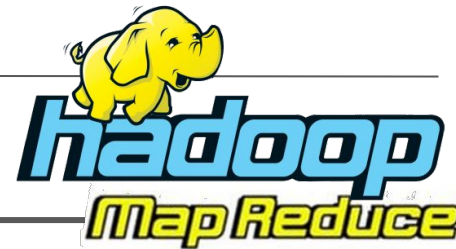
The **Map** in MapReduce

The mapping step takes a small piece of the input and maps the data to key-value pairs. A common example is sending a line of text to a mapper function, and the mapper generates a key-value pair for each word. The values will be added up later in the reducing step.

The Overall MapReduce Word Count Process



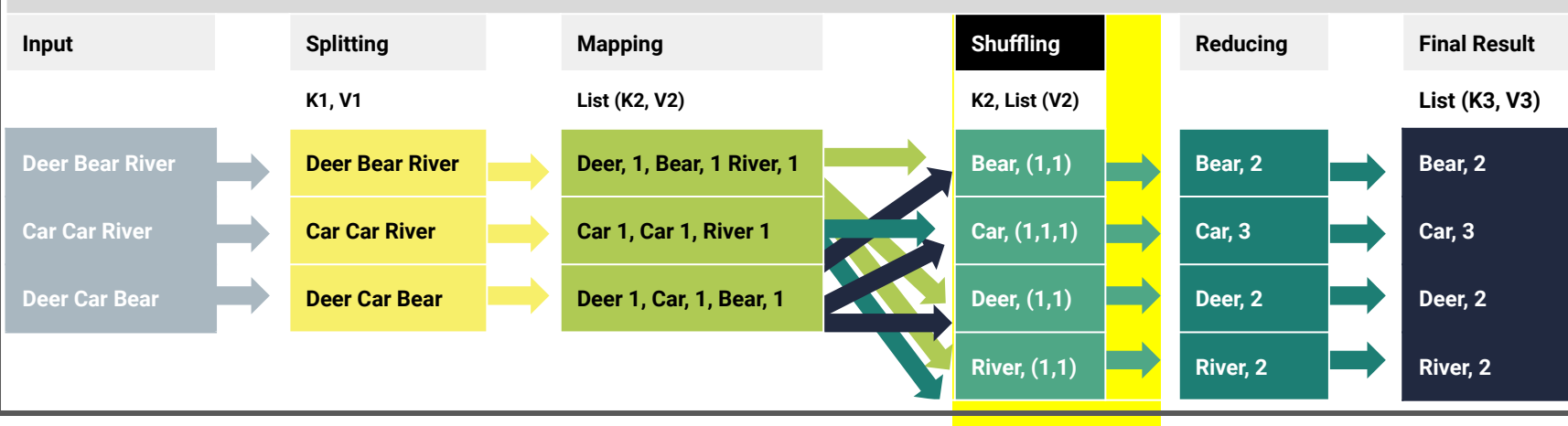
Instructor Do: Intro to MapReduce with mrjob



The **Shuffle** in MapReduce

The shuffle step groups the keys together. Each value found for a key is appended to the list of values.

The Overall MapReduce Word Count Process



Instructor Do: Intro to MapReduce with mrjob



The **Reduce** Part of MapReduce



Execute the `Reduce()` function on data.



Execute on some node.



Aggregates sets of `<key, value>` pairs on some nodes.



Output a combine list.

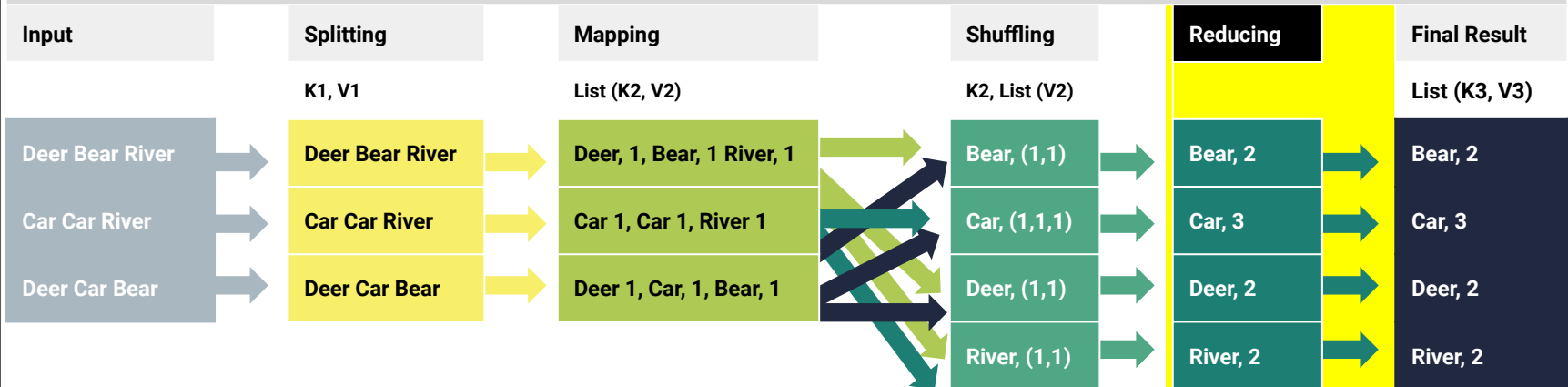
Instructor Do: Intro to MapReduce with mrjob



The **Reduce** in MapReduce

The reducing step reduces the list of the values for a key to a single value. In this example, the values are added to get the count of each word.

The Overall MapReduce Word Count Process





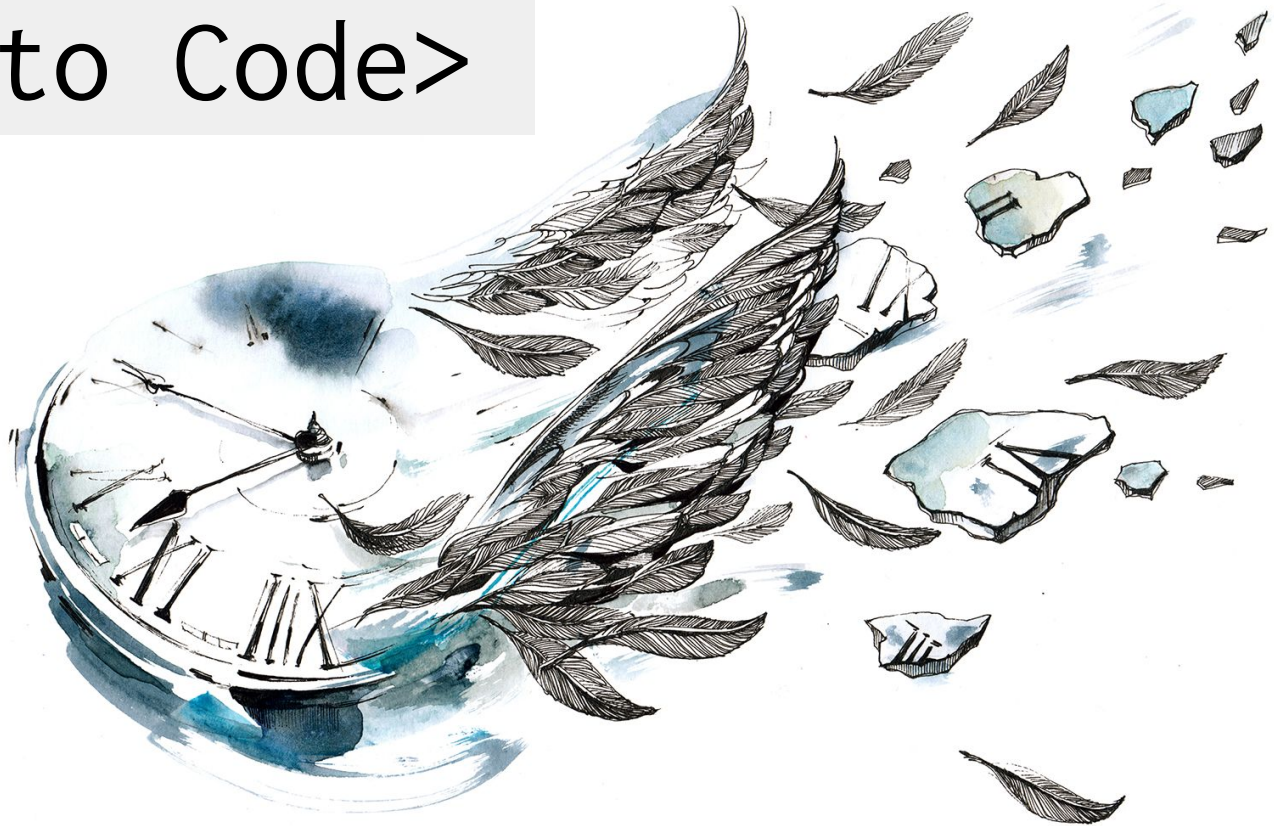
Everyone Do: Word Count

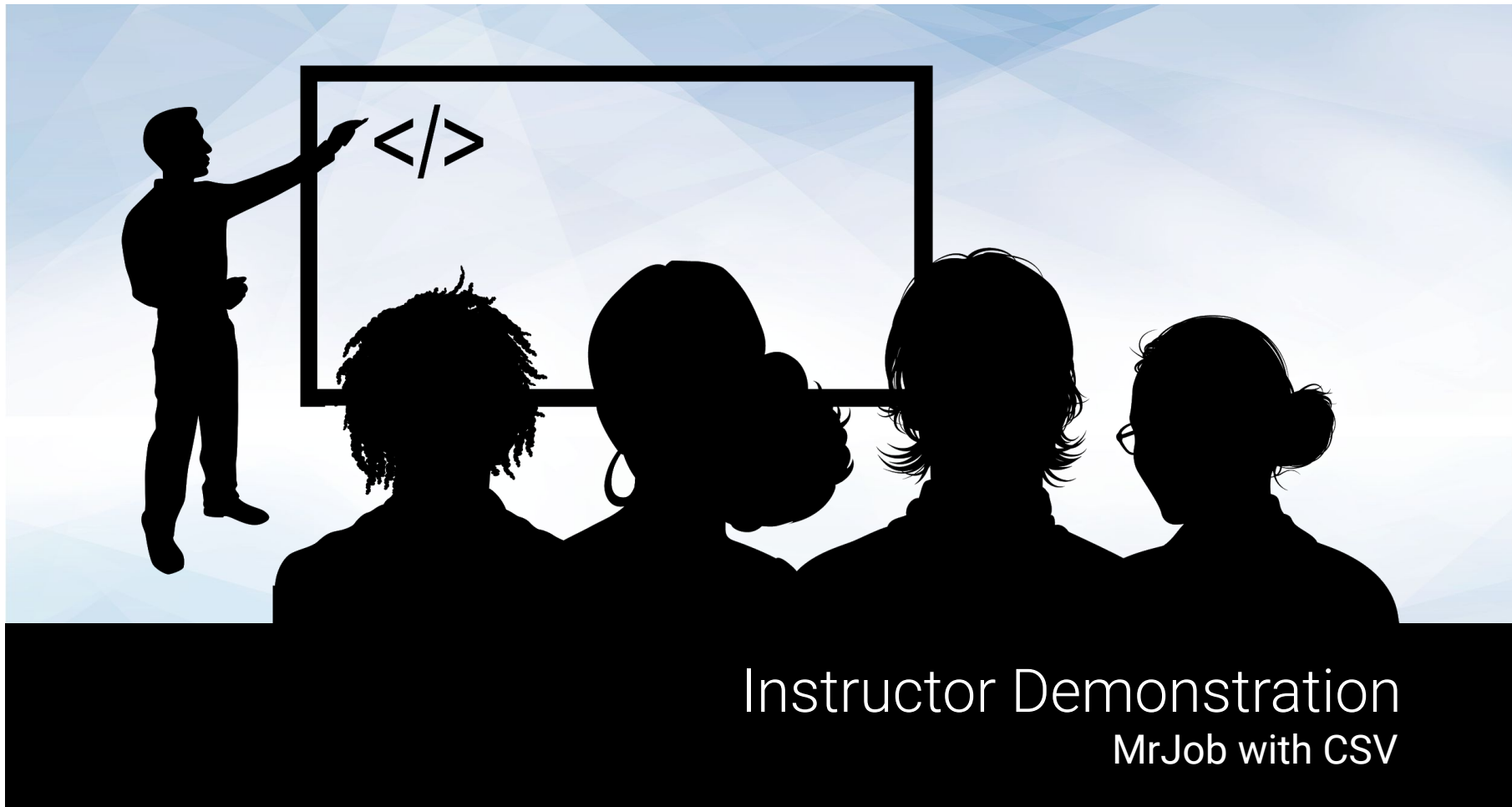
In this activity, we will have hands on experience writing our first job.

Suggested Time:
10 Minutes



<Time to Code>





Instructor Demonstration

MrJob with CSV

<Time to Code>





Activity: Snow in Austin

In this activity, you will use `mrjob` to determine the dates in which it snowed in Austin, Texas.

Suggested Time:
15 Minutes



Instructions:

Activity: Snow in Austin

→ Use mrjob to list the days in which it snowed in Austin, Texas.

- **Bonus:**

- Calculate the maximum amount of snow per date.

- **Hint:**



- Use the `max()` function to reduce the values for a date to a single value.



Time's Up! Let's Review.



Instructor Demonstration

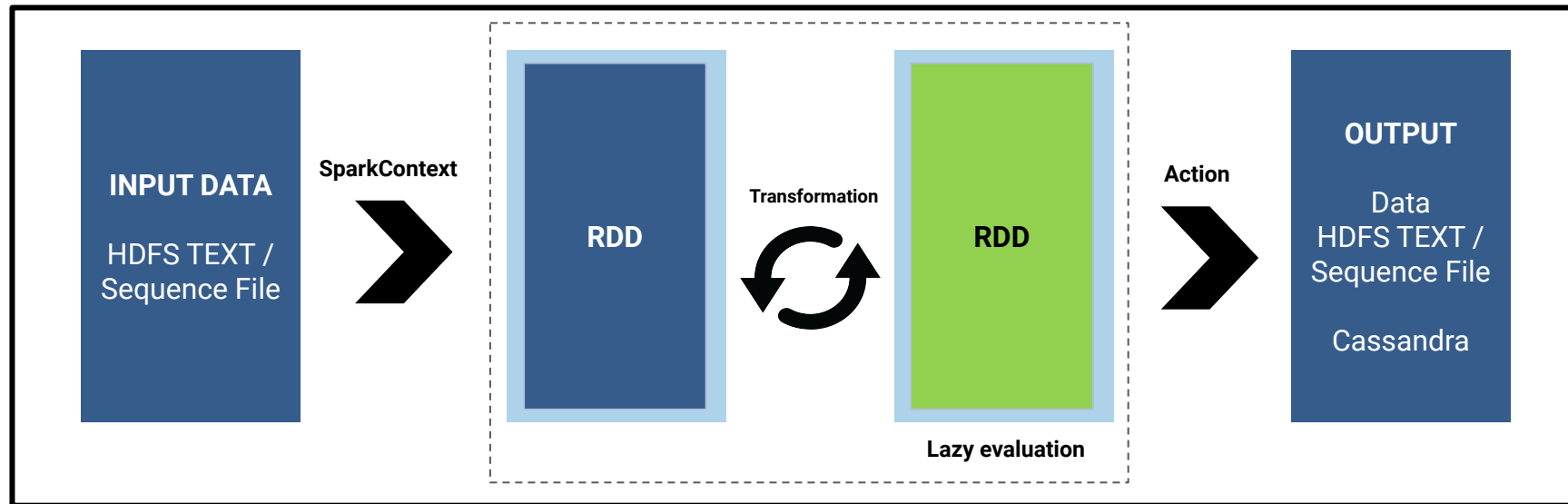
Spark Overview

Instructor Do: Spark Overview



What is Spark?

Apache Spark is a unified analytics engine for large-scale data processing. It lets you write applications in Java, Scala, Python, R, and SQL and runs on Hadoop, stand-alone, or in the cloud (and many other platforms). Spark can be 100 times faster than Hadoop.



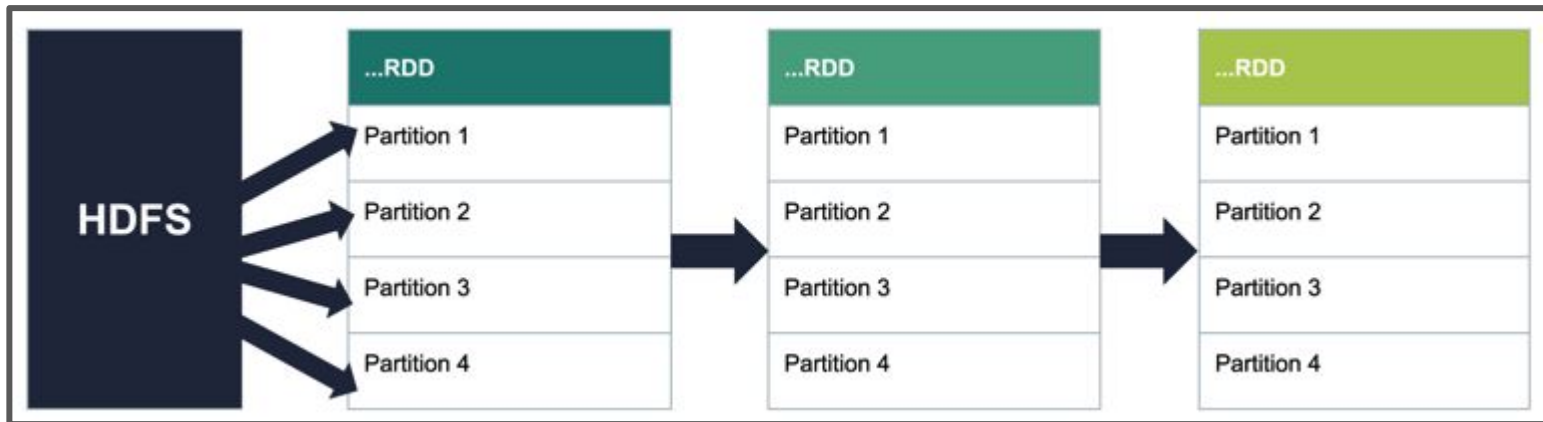
Instructor Do: Spark Overview



A Resilient Distributed DataSet (RDD)

A resilient distributed dataset (RDD) is the basic abstraction in Spark.

It represents an immutable, partitioned collection of elements that can be operated on in parallel.





Countdown timer

15:00

(with alarm)



Break



Instructor Demonstration

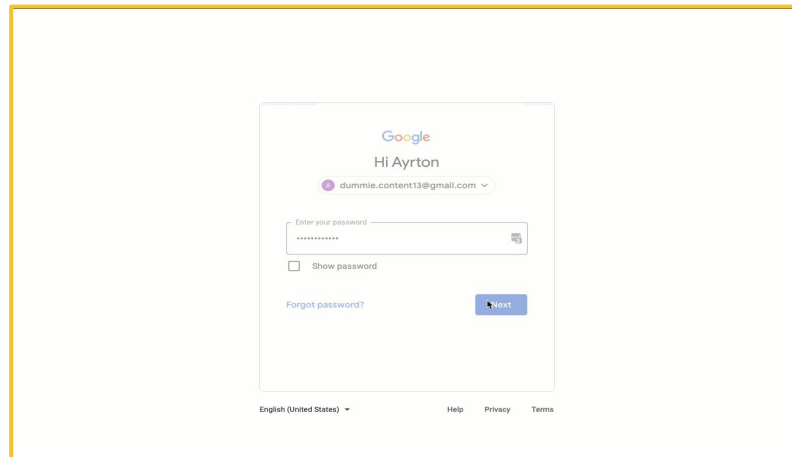
Setup Google Colab

Instructor Do: Setup Google Colab



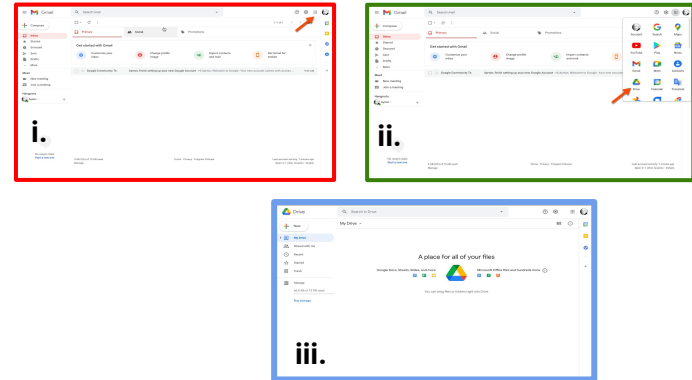
Starting with Google Colaboratory

- Google Colaboratory, or Colab, is a Google-hosted cloud-based notebooks.
- We will use Colab to run Spark.
- These cloud based notebooks allow for easy installation of Spark and the use of cloud computing power.

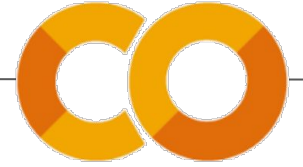


1. Login to Gmail


2. Navigating to Google Drive



Instructor Do: Setup Google Colab




3. Install Colab to the Drive

i. Click 

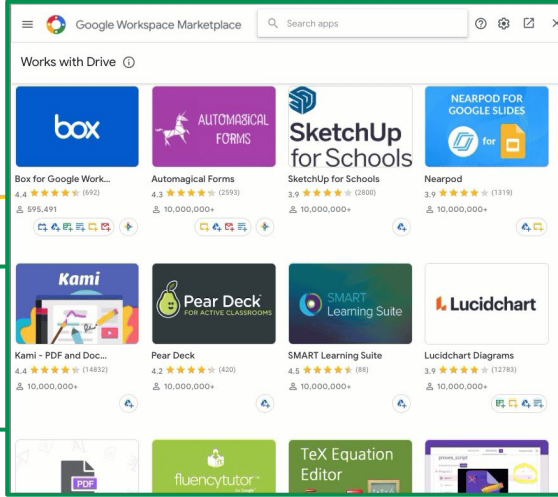
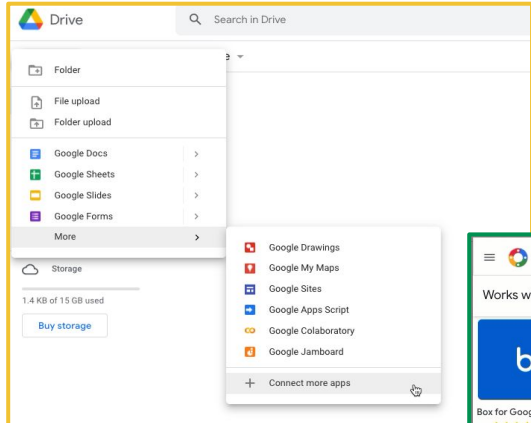
ii. More

iii. + Connect more apps

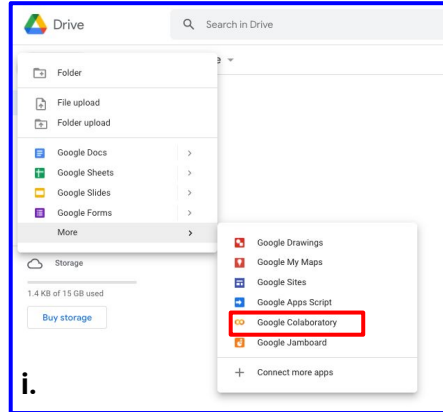
iv. Type "colab" in the search field.

v. Click 


vi. Click Install



i.



ii.



4. Create a Colab Notebook



Instructor Demonstration

PySpark DataFrame Basics

Instructor Do: PySpark DataFrame Basics



Spark DataFrames are similar to Pandas.



They hold data in a column and row format.



Each column represents a variable or feature.



Each row represents one data point.



Unlike Pandas, Spark DataFrames can scale to handle petabytes of data on clusters of servers or in the cloud.

<Time to Code>





Activity: Demographic DataFrame Basics.

In this activity, you will use the basics of PySpark DataFrame to analyze a demographic CSV.

Suggested Time:
15 Minutes



Activity: Demographic DataFrame Basics

→ Follow the comments in the Notebook to:.

- Clean the data.
- Display the data.
- Use Spark DataFrame.

- **Hint:**

- Read the PySpark Documentation.





Time's Up! Let's Review.



Instructor Demonstration

PySpark DataFrame Filtering

<Time to Code>





Activity: PySpark Demographic Filtering.

In this activity, you will use PySpark filtering to filter through the demographic dataset.

Suggested Time:
15 Minutes



Instructions:

Activity: PySpark Demographic Filtering

→ Using PySpark methods and the demographics dataset, answer the following questions:

- Which occupation had the highest salary?
- Which occupation had the lowest salary?
- What is the mean salary of this dataset?
- What is the `max` and `min` of the salary column?

- **Bonus:**
 - What is the average age and height for each academic degree type?

- **Hint:**
 - You will need to use `groupby` to answer this question.





Time's Up! Let's Review.



Instructor Demonstration

PySpark DataFrame Dates

<Time to Code>





Activity: Plotting Bigfoot

In this activity, you will use PySpark and Pandas to clean a Bigfoot dataset and create a plot.

Suggested Time:
15 Minutes



Instructions:

Activity: Plotting Bigfoot

→ Part 1

1. Using the Bigfoot data, import the time functions and load in the DataFrame.
2. Create a new DataFrame with column `Year`.
3. Save `Year` as a new column.
4. Find the total number of Bigfoot sightings per year.

→ Part 2

1. Import the summarized data to a Pandas DataFrame for plotting.
2. Clean the data and rename the columns `Year` and `Sightings`.
3. Plot the year and sightings.



Time's Up! Let's Review.