# Intro to Spark DataFrame API

## What is a DataFrame?

- ▶ DataFrame is a structured data type in Spark that can lead better expressivity, performance and space efficiency.
- ▶ Dataset is the other structured data type in Spark that is also statically typed (so it is only available in Scala and Java).
- ▶ The focus is on *"what to do?"* instead of *"how to do?"*
- ▶ The data is structured as a table (similar to a table in a relational database). We can formally describe the schema either directly in the host language or via a Data Definition Language (DDL).
- ▶ Data types supported in DataFrame in Spark.
- ▶ A RDD can be easily converted into a `DataFrame` and vice versa but there is a cost in the transformation.

# Examples

- Basic examples:
  - Example 1: DataFrame Schemas
  - Example 2: Working with Columns
  - Example 3: Working with Rows
  - Example 4: Wordcount with DataFrames
- Two complete examples of using data frames.
  - Example 5: Distribution of M&Ms by State
  - Example 6: Fire Call Analytics for San Francisco

# DataFrames/Datasets vs RDDs (1))

- ▶ If you want to tell Spark "*what to do*" versus "*how to do*", use DataFrames or Datasets.

- ▶ If you want high-level abstractions, and domain specific language operators, use DataFrames or Datasets.

- ▶ If you want static compile-time safety and don't mind multiple classes for specific Dataset[T]. use Datasets (you would have to use Scala or Java).

- ▶ If your processing demands high-level expressions, filters, maps, aggregations, computing averages or sums, SQL queries, columnar access, or use of relational operators on semi-structured data, use DataFrames or Datasets.

- ▶ If you want to take advantage of and benefit from efficient serialization, use Datasets.

# DataFrames/Datasets vs RDDs (2)

- ▶ If you are a Python user, use DataFrames and drop down to RDDs if you need more control.
- ▶ If you are an R user, use DataFrames.
- ▶ If you want errors caught during compilation rather than runtime, choose the appropriate API as shown below.

|                 | SQL     | DataFrames   | Datasets     |
|-----------------|---------|--------------|--------------|
| Syntax Errors   | Runtime | Compile Time | Compile Time |
| Analysis Errors | Runtime | Run Time     | Compile Time |

- ▶ RDDs remain the fundamental data structure that underlies DataFrames and Datasets. However, future development work will have a DataFrame (and Dataset) interface.
- ▶ We do, often need to use RDDs if third party package is written in RDDs. Some operators are only available on RDDs.