

Let's use again pyspark to explore the DataFrame API.  
We assume that the input files are located in the directory /data

```
# Creating new DataFrame objects from text, csv, JSON, and other
files
# can be done easily with spark.read.
# If the DataFrame schema is specified on the first line of the
file, use
# spark.read.option("header", True).
# Additionally, using spark.createDataFrame() you can create
DataFrames from
# existing Pandas DataFrames, RDDs, numpy arrays, and lists.

# Create a DataFrame from CSV file
df = spark.read.csv("/sabd-data/address.csv", header=True)
# Display the top rows of a DataFrame
df.show()
# Display the schema of a DataFrame
df.printSchema()

# From JSON to Parquet
peopleDF = spark.read.json("/sabd-data/people.json")
peopleDF.write.parquet("/sabd-data/people.parquet")
parquetFile = spark.read.parquet("/sabd-data/people.parquet")

# DataFrame and Spark SQL share the same execution engine so they
# can be interchangeably used seamlessly.
# Let's register the DataFrame as a table and run a SQL query
parquetFile.createOrReplaceTempView("parquetFile")
teenagers = spark.sql("SELECT name FROM parquetFile WHERE age >= 13
AND age <= 19")
teenagers.show()

# Compute mean value using DataFrames
from pyspark.sql.functions import avg
data_df = spark.createDataFrame([("Brooke", 20), ("Denny", 31),
    ("Jules", 30), ("TD", 35), ("Brooke", 25)],
    ["name", "age"])
# Group the same names together, aggregate their ages,
# and compute an average
avg_df = data_df.groupBy("name").agg(avg("age"))
# Show the result
avg_df.show()
# Visualize the DAG in Spark's UI
```