Importing Datasets

**Learning Objectives**

* Analyze Python data using a dataset
* Identify three Python libraries and describe their uses
* Read data using Python's Pandas package
* Demonstrate how to import and export data in Python

Python Packages for Data Science:

A Python library is a collection of functions and methods that allow you to perform lots of actions without writing any code. The libraries usually contain built in modules providing different functionalities which we can use directly. And there are extensive libraries offering a broad range of facilities. We have divided the Python data analysis libraries into three groups. The first group is called scientific computing libraries.

1. Scientifics Computing Libraries

**Pandas** offers data structure and tools for effective data manipulation and analysis. It provides facts, access to structured data. The primary instrument of Pandas is the two-dimensional table consisting of column and row labels, which are called a data frame. It is designed to provide easy indexing functionality.

The **NumPy** library uses arrays for its inputs and outputs. It can be extended to objects for matrices and with minor coding changes, developers can perform fast array processing.

**SciPy** includes functions for some advanced math problems such as Integrals, solving differential equations, optimization, as well as data visualization.

1. Visualization Libraries

Using data visualization methods is the best way to communicate with others, showing them meaningful results of analysis. These libraries enable you to create graphs, charts and maps. The **Matplotlib** package is the most well-known library for data visualization. It is great for making graphs and plots. The graphs are also highly customizable. Another high-level visualization library is **Seaborn**. It is based on Matplotlib. It's very easy to generate various plots such as heat maps, time series and violin plots.

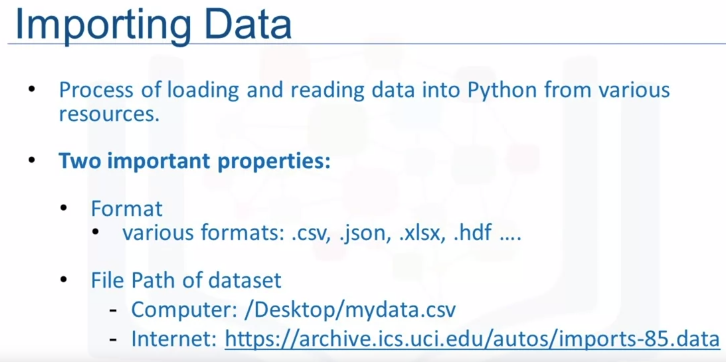
1. Algorithmic Libraries

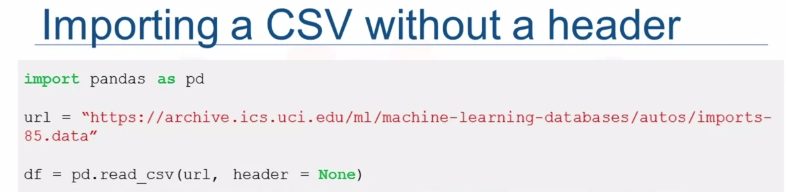
With machine learning algorithms, we're able to develop a model using our data set and obtain predictions. The algorithmic libraries tackle the machine learning tasks from basic to complex. Here we introduce two packages, the Scikit-learn library contains tools statistical modeling, including regression, classification, clustering, and so on. This library is built on NumPy, SciPy and Matplotlib. Stats models is also a Python module that allows users to explore data, estimate statistical models and perform statistical tests.

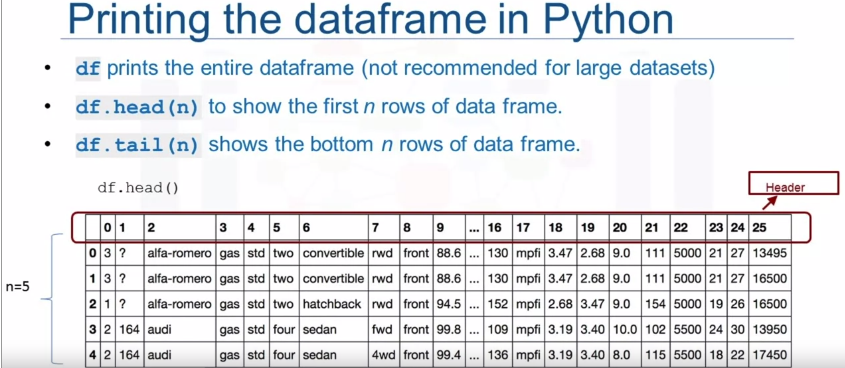
**Python Library describes:**

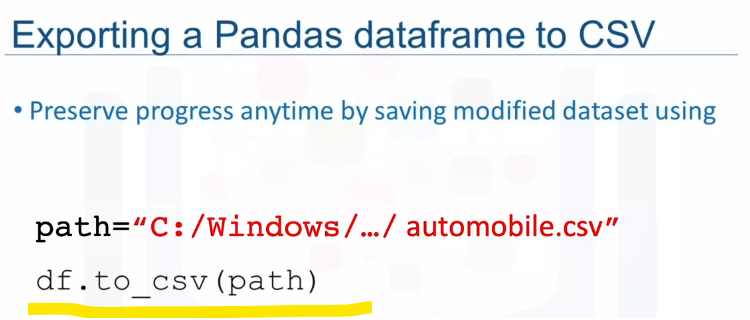
data structure and tools for effective data manipulation and analysis. It provides fast access to structured data. The primary instrument of Pandas is a two-dimensional table consisting of columns and rows labels which are called a Data Frame. It is designed to provide an easy indexing function.

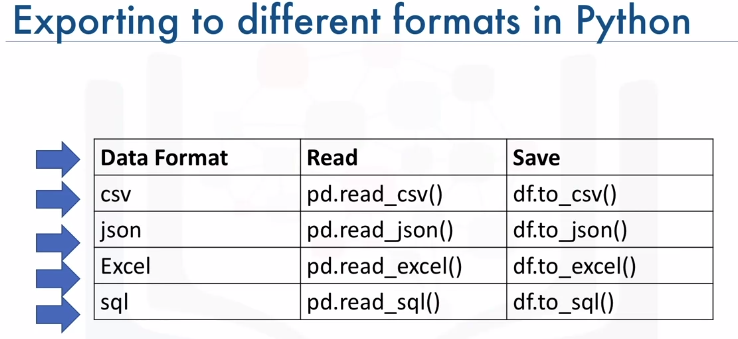
Importing and Exporting Data in Python:



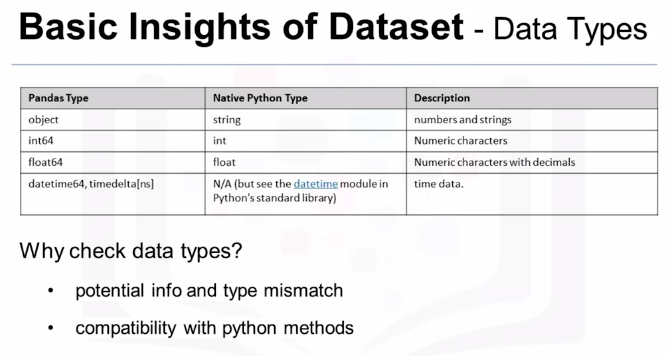


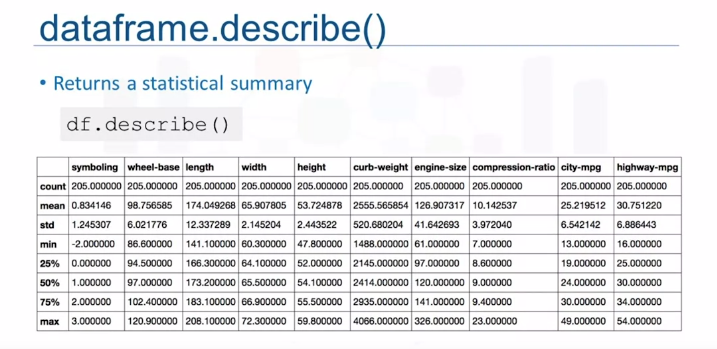






Getting Started Analyzing Data in Python:





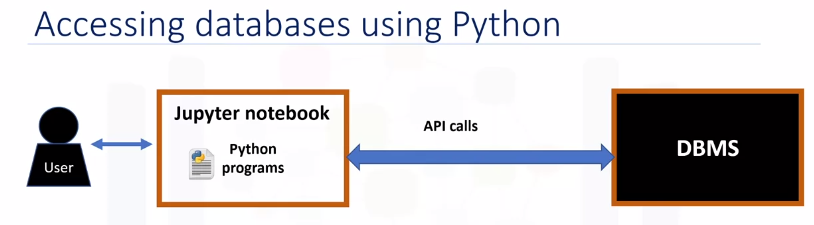
To get the quick statistics, we use the describe method. It returns the number of terms in the column as count, average column value as mean, column standard deviation as std, the maximum minimum values, as well as the boundary of each of the quartiles.



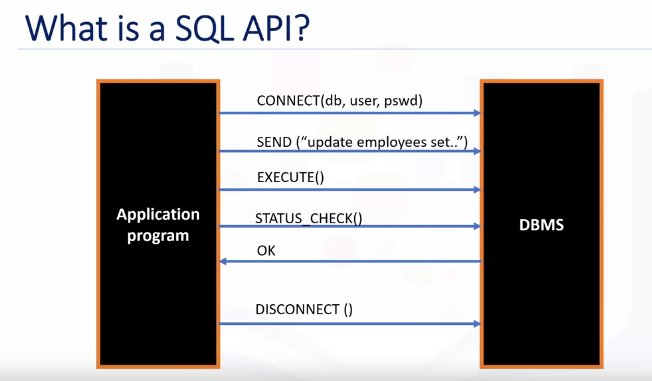
This function shows the top 30 rows and bottom 30 rows of the data frame. To view column names and data types.

Accessing Databases with Python:

Databases are powerful tools for data scientists. This is how a typical user accesses databases using Python code written on a Jupyter notebook, a web-based editor. There is a mechanism by which the Python program communicates with the DBMS.



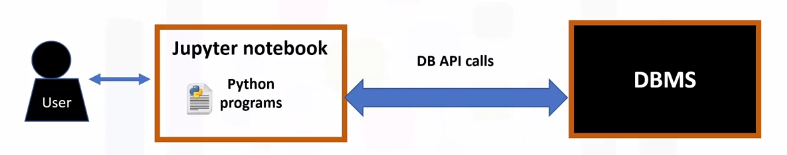
The Python code connects to the database using API calls. We will explain the basics of SQL APIs and Python DB APIs. An application programming interface is a set of functions that we can call to get access to some type of service.



The SQL API consists of library function calls as an application programming interface, API, for the DBMS. To pass SQL statements to the DBMS, an application program calls functions in the API, and it calls other functions to retrieve query results and status information from the DBMS. The basic operation of a typical SQL API is illustrated in the figure. The application program begins its database access with one or more API calls that connect the program to the DBMS. To send the SQL statement to the DBMS, the program builds the statement as a text string in a buffer and then makes an API call to pass the buffer contents to the DBMS. The application program makes API calls to check the status of its DBMS request and to handle errors. The application program ends its database access with an API call that disconnects it from the database.

**DB-API**:

DB-API is Python's standard API for accessing relational databases. It is a standard that allows us to write a single program that works with multiple kinds of relational databases instead of writing a separate program for each one. So, if you learn the DB-API functions, then we can apply that knowledge to use any database with Python.



The two main concepts in the Python DB-API are connection objects and query objects. We use connection objects to connect to a database and manage your transactions. Cursor objects are used to run queries. We open a cursor object and then run queries. The cursor works similar to a cursor in a text processing system where you scroll down in your result set and get our data into the application. Cursors are used to scan through the results of a database. Here are the methods used with connection objects.

**The connection methods:**

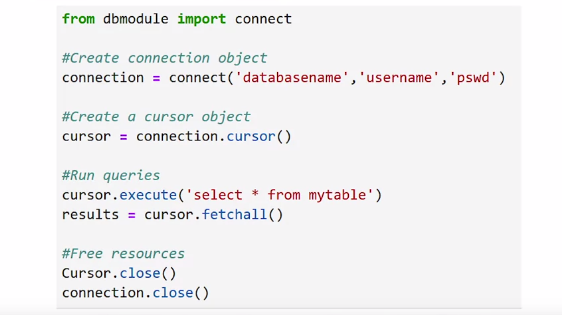
The **cursor()** method returns a new cursor object using the connection.

The **commit()** method is used to commit any pending transaction to the database.

The **rollback()** method causes the database to roll back to the start of any pending transaction.

The **close()** method is used to close a database connection.

Python application that uses the DB-API to query a database:



* First, we import our database module by using the connect API from that module.
* To open a connection to the database, we use the connection function and pass in the parameters that is, the database name, username, and password. The connect function returns connection object.
* After this, we create a cursor object on the connection object. The cursor is used to run queries and fetch results.
* After running the queries using the cursor, we also use the cursor to fetch the results of the query.
* Finally, when the system is done running the queries, it frees all resources by closing the connection. Remember that it is always important to close connections to avoid unused connections taking up resources.