Publication date : 12/18/2024

Course : Data Science with Python, API Society certification

RS 6763

Practical course - 4d - 28h00 - Ref. PYS

Price : 2630 CHF E.T.

On completion of the data scientist training program, participants will master the installation and use of scientific modules in a virtual environment, collaborate on data projects, manipulate and transform data for complex analyses, and create interactive and accessible visualizations tailored to user needs.

Teaching objectives

At the end of the training, the participant will be able to:

- Discover the scientific Python ecosystem
- Data manipulation and analysis with NumPy and Pandas
- Simple, interactive data visualization with Matplotlib, Seaborn, Plotly
- Leading data science and data vizualization projects

Intended audience

Statisticians, data analysts and data scientists

Prerequisites

Basic knowledge of the Python programming language

Certification

The certification exam takes place online, off-line and in French, in the month following the training course. It consists of a theoretical test lasting 20 minutes - 40 true/false MCQ questions and information to be entered (24 answers to be validated out of 40), and a practical programming test (code exercise) lasting 120 minutes on a format of 6 exercises (10 criteria to be validated out of 21).

PARTICIPANTS

Statisticians, data analysts and data scientists

PREREQUISITES

Basic knowledge of the Python programming language

TRAINER QUALIFICATIONS

The experts leading the training are specialists in the covered subjects. They have been approved by our instructional teams for both their professional knowledge and their teaching ability, for each course they teach. They have at least five to ten years of experience in their field and hold (or have held) decision-making positions in companies.

ASSESSMENT TERMS

The trainer evaluates each participant's academic progress throughout the training using multiple choice, scenarios, hands-on work and more.
Participants also complete a placement test before and after the course to measure the skills they've developed.

Practical details

Hands-on work

Individual and group practical work, collective reflection

Teaching methods

Active pedagogy encouraging personal involvement and exchanges between participants.

Course schedule

1 The scientific Python ecosystem

- Introduction to Python data science packages.
- Installation of libraries in a virtual environment: pip and the venv module, miniconda, mamba, miniforge, WinPython.
- Development environment.
- Using IPython, Jupyter Notebook, JupyterLab and IDE environments: the Spyder example.
- Discover the text editor: VS Code.

2 The NumPy library

- Introduction and creation of tables.
- Introducing the NumPy library.
- Advantages of tables (performance, data handling).
- Array creation with array(), zeros(), ones(), full(), arange(), linspace(), logspace().
- Matrix multiplication with np.dot and the @ operator.
- Initialization with random data (random module).
- Manipulate tables and operations.
- Indexing, slicing and advanced indexing.
- Transpose and change array dimensions (transpose(), reshape()).
- Concatenate and split arrays (concatenate(), split()).
- Handle classical and mathematical functions (sum(), min(), max(), median().
- Compare and mask data with Boolean masks.
- Data management and visualization.
- Load and save arrays (loadtxt(), save(), load()).
- Use the axis option in functions.
- Extract information from data.
- Use visualization practices: choice of modules and types of graphics.
- Generate interactive graphics.

TEACHING AIDS AND TECHNICAL RESOURCES

• The main teaching aids and instructional methods used in the training are audiovisual aids, documentation and course material, hands-on application exercises and corrected exercises for practical training courses, case studies and coverage of real cases for training seminars.
• At the end of each course or seminar, ORSYS provides participants with a course evaluation questionnaire that is analysed by our instructional teams.
• A check-in sheet for each half-day of attendance is provided at the end of the training, along with a course completion certificate if the trainee attended the entire session.

TERMS AND DEADLINES

Registration must be completed 24 hours before the start of the training.

ACCESSIBILITY FOR PEOPLE WITH DISABILITIES

Do you need special accessibility accommodations? Contact Mrs. Fosse, Disability Manager, at psh-accueil@orsys.fr to review your request and its feasibility.

3 The Pandas library

- Introduction and data structures.
- Introducing the Pandas library.
- Creating series with the series class.
- Create 2D arrays or DataFrames with the DataFrame class.
- Extract row and column indices (index and columns attributes).
- Read and export data in various formats (csv, xls).
- Implement basic methods: head() and tail().
- Indexing and slicing: implicit, explicit and the use of loc and iloc indexers.
- Select data and use Boolean expressions.
- Data manipulation and transformation.
- Insert and modify data.
- Rename columns with rename().
- Concatenate data with concat() and merge/join with merge() and join().
- Copy data: shallow or deep copy (copy()).
- Handle missing data (isna(), isnull(), notna(), notnull(), dropna(), fillna(), interpolate()).
- Handling indices: set_index(), sort_index().
- Sort values with sort_values().
- Transpose data with transpose().

4 Data analysis and aggregation

- Data aggregation: sum(), cumsum(), min(), max(), count(), mean(), median(), var(), std(), quantile(), describe()
- Grouping and analysis with groupby().
- Use aggregate(), apply(), filter(), transform() functions.
- Create pivot tables with pivot_table().
- Segment data with qcut() and cut().
- Calculate rolling averages with rolling(), expanding(), ewm().
- Process temporal data through to_datetime(), to_timedelta(), date_range(), period_range()...

5 The Matplotlib library

- Introduction and creation of graphics.
- Presentation of the library.
- Display graphs from a Python script (plt.show()) or from a notebook.
- Use MATLAB style or object-oriented style to display graphics.
- Modify graph style.
- Figure and axis objects.
- Plot curves with plot().
- Chart types and interactions.
- Display point clouds with scatter().
- Display error bars with error_bar().
- Fill the area between two lines with fill_between().
- Draw histograms with hist().
- 3D graphics with mplot3d.
- Interact with Jupyter notebook graphics using the interact widget.
- Use pandas plot to create plots quickly: plot(), bar(), barh(), hist(), box(), scatter(), pie().

6 Seaborn Library

- Introduction to Seaborn and basic functionality.
- Seaborn PLC operation: distinction between figure-level and axis-level pads.
- Relational Plots: use functions to plot relationships between variables.
- Plot distributions: use functions to visualize data distributions.
- Qualitative data: plot categorical data.
- Heat maps: use the heatmap() function to draw heat maps.
- Linear regression models: plot regression models with lmplot().
- Customize graphics: change the rendering of the figure using functions.

7 The Plotly library

- Presentation of the Plotly bookshop and Kaleido: introduction and exploration of Plotly Express.
- Drawing curves with line(): figure modification with title, width, height, marker, labels, etc. options.
- Create area graphs with area(): add patterns with pattern_shape.
- Creating point clouds with scatter(): using size, size_max, opacity, symbol, color_continuous_ options
- 3D graphics: using scatter_3d() and line_3d().
- Format bar charts with bar() and histograms with histogram().
- Draw maps with line_map(), scatter_map(), line_geo(), scatter_geo(), and choropleth().

Dates and locations

REMOTE CLASS
2026 : 31 Mar., 23 June, 29 Sep., 24 Nov.