# A Practical Guide to Teaching Python as a Computational Tool in an Introductory Data Analysis Course

**Serveh Sharifi Far**

School of Mathematics and Maxwell Institute for Mathematical Sciences, University of Edinburgh
Serveh.Sharifi@ed.ac.uk

**Ruini Qu**

Edinburgh Business School, University of Edinburgh
R.Qu@ed.ac.uk

**Stuart King**

School of Mathematics and Maxwell Institute for Mathematical Sciences, University of Edinburgh
S.King@ed.ac.uk

## Teaching Data Analysis and Programming: Does One Need to Come First?

Some introductory data science courses must teach both programming and statistics to students with no prior experience in either. Students may first take an introductory programming course; however, this is not always possible, and often programming must be integrated alongside data science concepts (Young et al. 2026). Teaching a data science or statistics course without a computational tool is not possible due to the amount of computations required in working with even small datasets. On the other hand, syntax-heavy programming courses may not immediately show relevance to data science and often struggle to engage students, who find it difficult to invest in abstract concepts without a clear sense of their practical application.

This chapter presents a structured, stepwise practical guide to introducing Python as a computational tool in an introductory data science course, allowing students to balance learning both technical skills (coding) with conceptual skills (statistical and data science thinking). The emphasis is on six fundamental elements to teach students to complement their data analysis learning in a process that supports a gradual transition from simple scripts to structured analysis. Each element introduces just enough Python to support statistical thinking, promote motivation, and empower inexperienced learners to use this tool for producing meaningful data analysis results.

## Overview of Such a Course

This approach has been implemented by the authors in an introductory data science course for diverse learners at the MSc level in an eleven-week semester (King and Sharifi Far 2025). We suggest structuring such a course so that statistical concepts and methods are taught in lectures, while computer workshop sessions focus on applying the methods using Python. In the workshops, Jupyter notebooks that combine explanatory text and executable code can be used as a common and effective tool to practice coding (El Gemayel, Budiarto, and Bell 2026). This benefits beginners by keeping instructional material and code side by side and reducing the cognitive load that can come with more complex development environments. Furthermore, working on notebooks in the workshops can then be done in the pair-programming format to stimulate student discussion and peer learning (Orzechowski, Blankinship, and Banas 2026). To support learning, we also recommend integrating continuous assessment throughout the course. Using automated marking tools for coding tasks can provide timely, objective feedback and help students identify specific areas for improvement, which reinforces understanding and encourages regular engagement with the material.

## Core Code Elements

Here we list the core code elements and Python functionalities fundamental to them, and the minimum set of inbuilt or module-specific functions needed to teach them. For each element, we describe its pedagogical goal and value, highlight common coding pitfalls observed in student learning, of which it can be helpful for instructors to be aware, and suggest real-world datasets to use that provide meaningful opportunities for practical application.

### Element 1 - Foundational Coding: Expressions, Variables, and Basic Calculations

- **Goal**: Introducing students to how Python solves simple mathematical problems and handles expressions, assigns variables, outputs results, and applies simple built-in functions.
- **Pedagogical value**: This stage invites students to become comfortable with the Python syntax and environment. It allows them to store and reuse results to make their code more readable and flexible. Combining variables with simple functions introduces the core logic of programming. This stage builds confidence by showing how basic Python code can mimic a calculator while adding structure and repeatability.
- **Essential functions**: Creating lists of data using `[]`, assignment operator (=), arithmetic operators (+, -, *, /, **), printing results with `print()`, checking type of variables with `type()`. Some built-in mathematical functions, for example, `round()`, `sum()`, `len()`, `min()`, `max()`, `abs()`.
- **Data suggestion**: Using minimal lists and arrays of numbers and characters is helpful at this stage. Students can be invited to make a small list of their favorite fruits, colors, or a short series of numeric values (e.g., prices, scores).
- **Common pitfalls**: Misspelling variables or functions names; reusing variable names unintentionally; calling an unassigned variable (e.g., the notebook has defined some variables in a cell and students call these variables without executing the cell); forgetting to assign results of calculations to variables; misunderstanding variable type changes (e.g., overwriting a numeric/integer variable with a text/string).

### Element 2 - Handling Data With `pandas`: Reading and Cleaning

- **Goal**: Equip students with the essential skills to load, explore, and clean datasets.
- **Pedagogical value**: It is essential for students to learn to work with structured data in the form of data frames. Introducing them to how to load datasets, explore their structure, and select or clean rows and columns is needed for visualisation, modeling, and interpretation.
- **Essential functions**: Importing `pandas`, functions such as `pd.DataFrame`, `pd.read_csv()`, `pd.read_excel()`, `head()`, `info()`, `describe()`, `loc[]`, `iloc[]`, selection of rows and columns with `[]` and operations on them (e.g., `df['col'].mean()`), handling missing data (`np.nan`, `dropna()`, `fillna()`, `replace()`), extra useful functions such as `groupby()`, `filter()`, `apply()`, `value_counts()`, `sort_values()`.
- **Real data suggestion**: Turtle Size dataset (Phillips et al. 2021) is a simply structured and helpful dataset which includes numerical and categorical variables. Penguins dataset (Gorman, Williams, and Fraser 2014) is another interesting and manageable dataset with some missing values for students to explore.
- **Common pitfalls**: Confusing `loc[]` with `iloc[]`; forgetting to save cleaned data in a variable; misunderstanding how `[]` behaves differently with rows and columns; forgetting that column names are strings when selecting them (e.g., `df["ColumnName"]`); forgetting to use a list when selecting multiple columns/rows (e.g., `dataframe.iloc[[5,1]]` vs `dataframe.iloc[5,1]`).

### Element 3 - Visualising Data With `matplotlib` and `seaborn`: Explore, Compare, and Communicate

- **Goal**: Use visual tools to describe the data, explore distributions, trends, and relationships among variables.
- **Pedagogical value**: Visualisation helps students in conducting exploratory analysis of the data, identifying outliers, checking distributions, and spotting potential relationships between variables, and reinforces the connection between statistical concepts and their visual representations.
- **Essential functions**: Importing `matplotlib.pyplot`, then `plt.plot()` and `plt.show()` with `plt.xlabel()`, `plt.ylabel()`, `plt.title()`. Importing `seaborn` and using the general format of `sns.---plot(data=---, x=---, y=---, kind=---, ...)` for different `kind` of plots.
- **Real data suggestion**: House Sales in King County (Harlfoxem 2016) data is a manageable dataset for students to explore many ways of visualising house prices based on various numerical and categorical features. The Hotel Booking Demand dataset (Antonio, Almeida, and Nunes 2019) also provides a chance to visualise a hotel room prices and availabilities in Portugal based on several factors.
- **Common pitfalls**: Forgetting to label axes or add titles; confusing `kind` options in plots; overcomplicating plots by including many variables; difficulties with data restructuring if required for a plot; difficulty in understanding the connection between `seaborn` and `matplotlib` for beginners.

### Element 4 - Summary Statistics: Describing Data

- **Goal**: Enable students to summarise variables and distributions and examine relationships between variables using descriptive statistics and correlation.
- **Pedagogical value**: Introducing summary statistics measures provides students with the language to describe data precisely. The concept of correlation offers an early opportunity to explore potential associations between variables, before modeling.
- **Essential functions**: Measurements of center and spread of data from `numpy` after importing it. The important functions are, `np.mean()`, `np.median()`, `np.mode()`, `np.std()`, `np.var()`, `np.quantile()`. Calculating Pearson linear correlation between two variables using `corr(method="pearson")`.
- **Real data suggestion**: Sleep in Mammals (Allison and Cicchetti 1976) and Animals Life Expectancy (Che-Castaldo et al. 2019) are interesting datasets about animals with various features in different types that can be used to encourage students to explore the pattern of how long these animals sleep and live.
- **Common pitfalls**: Applying numeric-only summary functions to non-numeric variables; ignoring or not checking for missing values; not inspecting the shape and possible skewness of the distributions.

### Element 5 - Normal Linear Regression With `statsmodels`: Explain and Predict Relationships

- **Goal**: Provide students with tools to build linear regression models to explain relationships between variables and make predictions, using syntax that mirrors traditional statistical notation.
- **Pedagogical value**: Linear models are simple yet foundational, offering students a strong basis for deeper study of statistical modelling. Introducing the normal linear model, along with the binary linear model (logistic regression) if time permits, could be sufficient at this stage. Attention should be given to checking the model assumptions and its goodness-of-fit.
- **Essential functions**: Importing the library `statsmodels.formula.api`, using the `ols` function in the form of `model = smf.ols("y ~ x_1 + x_2", data=df).fit()` and the corresponding required functions, `model.summary()`, `model.predict()`, `model.resid`. Some useful plots are `sns.lmplot()`, `sns.residplot()`, and `qqplot()` for visualisation and checking the model assumptions.
- **Real data suggestion**: World Happiness Report ("The World Happiness Report," n.d.) data from most countries around the world and different years can be used to aim to model the countries level of happiness based on various measured social elements. Programme for International Student Assessment (Organisation for Economic Co-operation and Development, n.d.) data provides standardised exam scores for a large sample of students in different countries along with many social and educational indicators.
- **Common pitfalls**: Difficulty in coding categorical variables; in extending the model formula to variable names including spaces; and in inputting new data in the `predict` function or passing in wrong data types.

### Element 6 - Machine Learning With `scikit-learn`: Classification

- **Goal**: Introduce students to basic supervised machine learning for classification problems, using intuitive models like k-Nearest Neighbors, decision trees, and ensemble methods, and reinforcing best practices in model evaluation.
- **Pedagogical value**: Introducing classification methods shows students how algorithms can learn patterns from labelled data to make predictions.
- **Essential functions**: From `scikit-learn` use `train_test_split` to prepare data for training and assessing models. Then apply different classification methods, using `KNeighborsClassifier`, `DecisionTreeClassifier`, `RandomForestClassifier`, `BaggingClassifier`, and `HistGradientBoostingClassifier`, which all need `fit()` and `predict()`. Evaluation of the classification can be done using `classification_report`, `confusion_matrix`, and a simple scatter plot like `sns.scatterplot()` is useful in visualisation.
- **Real data**: Pima Indian Diabetes dataset (Smith et al. 1988) includes various health indicators to use in modelling to predict whether participants would develop diabetes. Behavioral Risk Factor Surveillance System (Centers for Disease Control and Prevention, n.d.) data is available from several years and includes many health indicators and records of presence of many health conditions to use in different classification problems.
- **Common pitfalls**: Not applying the train/test split or mishandling its output; difficulty in choosing and applying the appropriate evaluation methods.

Depending on the course level and learning objectives, Element 6 may be included only if an introduction to machine learning falls within the scope of the course. Here, we presented classification as an example of a supervised learning method, although an unsupervised method such as clustering could also be introduced following a similar structure.

## What Can Students Take Away?

By working through these core elements, students gain a practical foundation in both Python programming and data analysis. This progression helps students move from simple code scripts to structured analytical workflows, equipping them with the confidence and technical ability to engage with real-world datasets. By the end of this material, students should be able to implement core Python functions for data analysis independently and recognise how coding supports statistical thinking. The common pitfalls noted throughout are not mere technical errors, but can be important learning opportunities.

## References


Allison, Truett, and Domenic V Cicchetti. 1976. “Sleep in Mammals: Ecological and Constitutional Correlates.” *Science* 194 (4266): 732–34.

Antonio, Nuno, Ana de Almeida, and Luis Nunes. 2019. “Hotel Booking Demand Datasets.” *Data in Brief* 22: 41–49. https://doi.org/10.1016/j.dib.2018.11.126.

Centers for Disease Control and Prevention. n.d. “Behavioral Risk Factor Surveillance System (BRFSS) Data.” https://www.cdc.gov/brfss/annual_data/annual_data.htm.

Che-Castaldo, Judy P, Amy Byrne, Kaitlyn Perišin, and Lisa J Faust. 2019. “Sex-Specific Median Life Expectancies from Ex Situ Populations for 330 Animal Species.” *Scientific Data* 6 (1): 190019. https://doi.org/10.1038/sdata.2019.19.

El Gemayel, Joseph, Arif Budiarto, and William Bell. 2026. “Notebook for Novices? Pros and Cons of Jupyter.” In *Teaching Programming Across Disciplines*. Edinburgh Diamond.

Gorman, Kristen B, Tony D Williams, and William R Fraser. 2014. “Ecological Sexual Dimorphism and Environmental Variability Within a Community of Antarctic Penguins (Genus Pygoscelis).” *PloS One* 9 (3): e90081. https://doi.org/10.1371/journal.pone.0090081.

Harlfoxem. 2016. “House Sales in King County, USA.” Kaggle dataset. https://www.kaggle.com/datasets/harlfoxem/housesalesprediction.

King, Stuart, and Serveh Sharifi Far. 2025. “Teaching Data Science to Diverse Learners: A Hybrid and Interdisciplinary Approach.” *Teaching Statistics*. https://doi.org/10.1111/test.70014.

Organisation for Economic Co-operation and Development. n.d. “PISA Dataset.” https://www.oecd.org/en/about/programmes/pisa/pisa-data.html.

Orzechowski, Pawel, Brittany Blankinship, and Kasia Banas. 2026. “Where Do i Even Start with Pair Programming in My Classroom? A Conversation with Seasoned Practitioners.” In *Teaching Programming Across Disciplines*. Edinburgh Diamond.

Phillips, Katrina F, Gustavo D Stahelin, Ryan M Chabot, and Katherine L Mansfield. 2021. “Long-Term Trends in Marine Turtle Size at Maturity at an Important Atlantic Rookery.” *Ecosphere* 12 (7): e03631. https://doi.org/10.1002/ECS2.3631.

Smith, Jack W, James E Everhart, William C Dickson, William C Knowler, and Robert Scott Johannes. 1988. “Using the ADAP Learning Algorithm to Forecast the Onset of Diabetes Mellitus.” In *Proceedings of the Annual Symposium on Computer Application in Medical Care*, 261.

“The World Happiness Report.” n.d. https://worldhappiness.report/.

Young, Robert S., Rebecca L. Colquhoun, Tiago A. Marques, Brittany Blankinship, Ozan Evkaya, and William P. Kay. 2026. “Sequential Vs. Simultaneous: Approaches to Learning Programming and Statistics.” In *Teaching Programming Across Disciplines*. Edinburgh Diamond.