

ASM Webinar

EXPLORING DATA MANIPULATION & ACTUARIAL MODELLING IN PYTHON (18.05.2021)

Agenda

- Why should we learn programming?
- Why do we choose Python?
- How do we use Python?
- How do we write Python coding?
- Case Study – Movements of policies analysis & cash flow projections

Why should We Learn Programming?



- In our view, actuarial expertise are NOT IT programmers. Instead, we consider ourselves as **modelers**. We learn programming languages to allow us to be capable to perform detailed studies on data and model calculations.
- Able to perform studies from DATA:
 - To derive conclusions and observations from policy and claim data.
 - Spreadsheet is not the correct tool to manage and study data.
 - In the past, actuarial expertise using SQL in Visual FoxPro to perform data manipulation.
- Able to model CALCULATIONS:
 - Business problems do not carry specific patterns.
 - To model calculations for business problems as per correct methodology, without needing to go through the entire development cycle of IT solutions.

Photo by [Emil Priver](#) on [Unsplash](#)

Why do We Choose Python?



Companies using Python include:

Google



NETFLIX



- As described in “*Why Actuaries Should Start Paying Attention to Python*” by Andrew M. Webster (August 2018 issue), Python is the “Swiss army knife” of the programming world.
- Our considerations in selecting a tool include (1) capable to do various modelling work; (2) easy to learn; (3) cost efficient; (4) reasonable run time.
- Reasons of why we choose Python Include:
 - As a general-purpose programming language, Python is suitable for many tasks and simple to learn.
 - Python is open source and free, which users can obtain and learn Python without incurring any costs.
 - We can link readily developed libraries / packages in our coding, which simplify and reduce development efforts.

“In the beginning, the founders of Google made the decision of “Python where we can, C++ where we must.” This meant that C++ was used where memory control was imperative and low latency was desired. In the other facets, Python enabled for ease of maintenance and relatively fast delivery.”

How do We Use Python? (1)



Python Command Line

```
Python 3.8 (64-bit)
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64-bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello, World!")
Hello, World!
>>> x = 5
>>> y = "John"
>>> print(x)
5
>>> print(y)
John
>>>
```

- Python is an interpreted high-level general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects
- We can download & install Python Command Line directly from Python Foundation website. We can also use run Python in Windows Command Line after installation.
- Similar to Excel VBA, It is not compulsory to declare a variable in advance. Some conventions in variable naming in Python:
 - A variable name must start with a letter or the underscore character
 - A variable name cannot start with a number
 - A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
 - Variable names are case-sensitive (age, Age and AGE are three different variables)

How do We Use Python? (2)

Offline



Online



- Although we can use Python Command Line to write and run Python codes, we normally use an **IDE** (integrated development environment) to develop our solutions.
- Examples of popular Python IDE are **PyCharm** (Freemium), **Atom** (Free) and **Anaconda** (Freemium). For a non-programmer, we normally use limited features of IDE when modelling our solutions, i.e., to choose an IDE that is straightforward and easy to use.
- If you just would like to learn Python without installing Python Command Line or any Python IDE, you may directly use Python Online Compiler by **Programiz**.
- Advantages of IDE compared to IDLE like Python Command Line:
 - Creating better codes
 - Debugging functionality
 - Separate codes and outputs into different windows
 - Directly store codes in separate Python files
 - Easy to search and download modules / packages / libraries

How do We Use Python? (3)



- Instead of writing all required codes from scratch, we can use modules / packages / libraries developed by other Python developers. Apart from standard libraries (e.g. “csv”, “datetime” & “decimal”), there are many packages that we can download from the internet or within IDE itself.
- By using this ready-made packages, we can save a lot of development time and efforts. Furthermore, these packages are usually developed by experienced programmers – which works better than codes written ourselves (as non-programmers).
- Two packages that we are using in our case studies are **pandas** and **NumPy**: (as explained in “*Top 10 Python Packages Every Developer Should Learn*”)
 - **Pandas** – There is a long list of Python packages designed for working with complex data sets. But arguably, Pandas is the most important. Pandas helps you manipulate and analyze large sets of data without having to learn a specialized data-processing language like R. If you need to do things like process time-series data or perform statistical analysis on a data set, Pandas has you covered.
 - **NumPy** - NumPy provides tools to help build multi-dimensional arrays and perform calculations on the data stored in them. You can solve algebraic formulas, perform common statistical operations, and much more. While NumPy is a valuable Python package for a variety of general-purpose programming tasks, it’s particularly important if you want to do machine learning, since it provides part of the foundation for libraries like TensorFlow.

How do We Write Python Coding?



Photo by [Juan pablo Garcia](#) on [Unsplash](#)

To demonstrate how Python coding is written, we shall use Python Command Line to carry calculations used to compile a claim development triangle for medical products (assuming all claims are reported within 15 months from loss dates).

- Python codes are organized by new lines and indentation. General syntax of Python coding:
 - Python codes are case sensitive, similar to Power Query. Python commands are written in lower case.
 - For every statement (which normally the first line is ended with ":"), we need to include an indentation (i.e. normally 4 spaces).
 - To join 2 separate lines, we need to insert "\n" at the end of the first line. Hence, when we want to define file paths, we need to use "\\" (e.g. "C:\\Python" instead of "C:\Python").
 - To insert a comment, we need to start a new line by "#".
 - Python also uses statements, such as *for* and *with* statements. Different from many programming languages, Python does not need to that a "closing" command at the end of the statement (e.g. *Next* for *For* statement in VBA).
- Some important aspects in Python coding:
 - To use a package / library, we need to use "import" (to include the entire package) or "from ... import..." (to include a specific sub-package).
 - There are 4 built-in data types in Python used to store collections of data, i.e. **list** (using "["), **tuple** (using "("), **set** (using "{") and **dictionary** (using "{:}").

Case Study I & II

Movements of
Policies

Cash Flow
Projection



- To provide a better idea on how we can use Python in our daily work, we shall demonstrate how we use Python in the following case studies: (simulated from real life actuarial work)
 - Case study I: Movements of policies analysis
 - Case study II: Cash flow projections, including GPV reserves calculations.
- Movements of policies analysis involves comparing two sets of monthly in force policy data, in order to identify current month new business, terminations and alterations.
- Cash flow projections involves projecting and discounting future cash flows on designated model points. These are common calculations that are carried out using actuarial software, such as Prophet.
 - Detailed results for all variables (for all projection months) are written into text files by cohorts.
 - We setup a spreadsheet template to retrieve results for required results over specific range of projection months.

Notes: As the solutions are still under development and research stage, we are unable to provide Python codes used in discussing the above case studies.

Photo by [Austin Distel](#) on [Unsplash](#)

References

Python (programming language)

[https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))

Why Actuaries Should Start Paying Attention to Python, by Andrew M. Webster

<https://www.soa.org/news-and-publications/newsletters/innovators-and-entrepreneurs/2018/august/ie-2018-iss64/why-actuaries-should-start-paying-attention-to-python/>

8 World-Class Software Companies That Use Python

<https://realpython.com/world-class-companies-using-python/>

What is Python: the major features and what it's used for

<https://python.land/python-tutorial/what-is-python>

Top 10 Python Packages Every Developer Should Learn

<https://www.activestate.com/blog/top-10-must-have-python-packages/>

Disclaimer

- The contents of this document are only for the reference of participants attending ASM Mini Sharing on 18.05.2021. Unauthorized circulation or duplication of any part of this document to any unauthorized third party is prohibited.
- The copyrights of this document are owned by ACTBuilder Actuarial Solutions Sdn. Bhd.
- ACTBuilder Actuarial Solutions Sdn. Bhd is not responsible for unauthorized use of any part of the contents of this document.

Contact Info:

ACTBuilder Actuarial Solutions Sdn. Bhd. (1108112-P) | Lim Chee Beng | CheeBeng.Lim@actbuilder.com.my