Advanced Efficiencies – The Language of SAS

Please read our Terms and Conditions and our Privacy Policy.

Duration: 6 – 8 hours (plus additional time for exercises)

Learning Overview:

The Advanced Efficiencies course covers a number of best practice coding techniques that will enable you to optimise the use of CPU time, Working Memory and Data Storage resources.

The learning on this blended learning course will be delivered over 2 live online classes with comprehensive exercises throughout to help assess and reinforce your learning.

Learners will also have the opportunity to join an end of day Q&A session with the Instructor.

Learning Outcome:

By the end of this course you will be able to:

1. Explain how data flows in to and out of a SAS session
2. Create Dataset Indexes and explain how they operate
3. Describe techniques to measure and improve program code efficiency
4. Reduce storage requirements for numeric and character data

Delivery Schedule:

The course is delivered live online over 2 x 4 hour sessions.

On day 1 we invite you to join an online welcome to meet your instructor and introduce you to our Digital Learning platform before getting you started on the course.

In addition, you will also need to set aside time to complete the Hands-On exercises.

Day 1	Welcome and Live Online Class 1	09:00 - 13:00(GMT)
Day 2	Live Online Class 2	09:00 - 13:00(GMT)

Pre-requisites:

To get the most out of this course, it is expected that you should have attended our Fundamentals and Intermediate courses in the language of SAS (or comparable courses) or have completed nine months of developing code in the language of SAS including topics:

- Global statements e.g. LIBNAME, FILENAME and OPTIONS
- Use functions to transform character data (e.g. SCAN, FIND, SUBSTR)
- Use functions to transform numeric data (e.g. CEIL, INT, FLOOR, ROUND)
- Use functions to transform dates (e.g. DAY, MONTH, YEAR, INTCK)
- KEEP and DROP to select variables
- IF and WHERE statements to select observations
- Conditional Processing with IF-THEN-ELSE
- Iterative DO Loops
- Conditional DO Loops-DO WHILE, DO UNTIL
- Compile-Time statements (e.g. LENGTH, RETAIN)
- Combining Data Sets vertically and horizontally
- Summarising data with the FREQ, MEANS and TABULATE procedures

For the hands-on practice activities in the course, you will need access to an environment that runs the programming Language of SAS. On our courses, we signpost you to some of the free tools available.

Check out the link below to review system requirements:

- SAS® OnDemand for Academics

Learning Modules:

Introduction - Optimise Task Processing

Learning Objective: Explain techniques to optimise Task Processing

- Measuring efficiencies
- Efficiency trade-offs
- Selecting appropriate efficiency options
- Know your data, processes and system

Data Set Pages and I/O Processing

Learning Objective: Explain how Data Set Pages affect I/O Processing

- Identify the attributes of a Data Set Page
- Describe how Data Set Pages are handled in I/O Processing

Creating & Using Indexes

Learning Objective: Explain how Indexes are created and used

- Describe a dataset Index
- Methods to create an Index
- How an Index is used

Efficient Program Code

Learning Objective: Describe coding choices that will deliver efficient solutions

- Do not process data unnecessarily
- Reduce the number of steps
- Store data in sorted order
- Effective use of functions
- Generate appropriate Indexes
- Use the Macro Processing Facility

Numeric Storage

Learning Objective: Identify how and when numeric storage can be reduced

- Describe how numbers are stored
- Reducing the length of Numerics

Compressing Datasets

Learning Objective: Explain how to compress datasets and summarise the advantages and disadvantages.

- Attributes of a Compressed Data Set
- Compressing a Data Set
- Comparing Compression Algorithms
- Advantages and Disadvantages

Sorting Datasets

Learning Objective: Describe methods to improve sort performance

- Review of the PROC SORT procedure
- Parallel Processing
- Space Requirements for Sorting
- Selecting Observations