Next Generation Higher National Unit Specification

Data Science (SCQF level 8)

Unit code: J6C5 48

SCQF level: 8 (24 SCQF credit points)

Valid from: session 2022–23

Prototype unit specification for use in pilot delivery only (version 1.0) June 2022

This unit specification provides detailed information about this unit to ensure consistent and transparent assessment year on year.

This unit specification is for teachers and lecturers and contains all the mandatory information required to deliver and assess this unit.

This edition: June 2022 (version 1.0)

Unit purpose

The purpose of this unit is to introduce learners to source data and prepare it for analysis. They apply mathematics, statistics, and scientific methods to extract insights from the data using descriptive and predictive analytics. They program applications that automate data processing and calculations, and use data visualisation to tell stories that clearly convey meaning to stakeholders with various levels of technical knowledge.

This is a specialist unit, suitable for those who want to develop their existing data skills as part of their current work role or gain specialist skills in data science. We recommend that learners have experience of data analysis. They should be familiar with numerical software (such as spreadsheets) and have well-developed numeracy skills.

On completion of this unit, learners can progress to a variety of more specialist units, such as Data Visualisation at SCQF level 8 and Machine Learning at SCQF level 8.

Unit outcomes

Learners who complete this unit can:

1 explain the data science workflow
2 extract, transform and load data to prepare for analysis
3 describe statistical methods relevant to data science
4 perform data analysis to obtain insights using statistical tools and methods
5 write programs to automate data processing and analysis
6 communicate insights using a range of data visualisations and reports

Evidence requirements

Learners must provide knowledge and product evidence over an extended period under lightly controlled conditions.

Knowledge evidence

Learners must produce evidence for all knowledge statements. The amount of evidence should be the minimum to infer competence.

If you use a test, you can sample knowledge evidence. Learners must produce evidence under controlled conditions in terms of location, timing and access to reference materials. The sampling frame must cover all outcomes, with particular focus on outcomes 1 and 3. The evidence can be captured, stored and presented in a range of media (including audio and video) and formats (analogue and digital).

Product evidence

Product evidence relates to outcomes 2, 4, 5 and 6, and consists of the analysis (including statistical analysis) and the program code produced as part of these outcomes.

Evidence must demonstrate that learners can ethically apply a range of data science methods to a large dataset, and report on data insights in a clear and visually attractive way. The dataset must relate to a real context and come from two or more sources, of which one must be external. It must be multivariate and include text data, date fields, categorical and numerical data with at least 5000 records.

Learners can produce product evidence with access to reference materials. You must authenticate the evidence and the Guide to Assessment provides further advice on methods of authentication.

The evidence can include:

- screenshots of software package output
- source code of programs
- output from test data
- printed output of data analysis
- visualisations in digital or print form
- reports in digital or print form

Learners can present their product evidence in folders or in an e-portfolio.

The standard of evidence should be consistent with the SCQF level of this unit.

Knowledge and skills

The following table shows the knowledge and skills covered by the unit outcomes:

Knowledge	Skills
Learners should understand: ♦ data quality and data bias ♦ stages in the data analysis process (workflow) ♦ ethical considerations in data sourcing and processing ♦ generation of data insights through descriptive and predictive analysis ♦ prescriptive analysis and data-driven decision-making ♦ types of data and data formats ♦ data structures including tables and databases (SQL and NoSQL) ♦ data models and table relationships ♦ data joins ♦ types of data transformation, including scaling and normalisation ♦ populations and samples ♦ sampling methods ♦ probability and frequency distributions ♦ descriptive (summary) statistics and data plots ♦ model parameters and statistical estimates ♦ correlation and linear regression ♦ statistical inference ♦ confidence intervals and tests of hypothesis ♦ programming languages for data processing and analysis ♦ software development tools such as integrated development environments (IDEs) and code repositories ♦ code libraries for data manipulation and statistical analysis ♦ application programming interfaces (APIs) for data sources	Learners can: ♦ extract data from a source, including the use of APIs ♦ apply joins to data tables ♦ apply methods to clean data including the treatment of missing values ♦ create transformed data, including scaling and normalisation ♦ apply one hot encoding to categorical variables ♦ load data ready for analysis ♦ perform exploratory data analysis using statistical software ♦ produce summary statistics and frequency distributions ♦ produce summary tables, charts and plots from data ♦ perform a linear regression using statistical methods ♦ perform a test of hypothesis ♦ write a program to process data from an external source ♦ write a program to transform data ♦ write a program to perform exploratory data analysis and produce summary statistics ♦ test and debug a program ♦ select appropriate visualisations of data ♦ create visualisations using software ♦ create a story of data insights using visualisations

Knowledge	**Skills**
Learners should understand: ♦ programming techniques, including debugging ♦ source and version control ♦ test data and test plans ♦ data visualisation as a tool for decision-making ♦ types of data visualisation ♦ key elements of data visualisations ♦ misleading visualisations ♦ reports and dashboards ♦ characteristics of effective data storytelling ♦ ethical issues in data analysis and reporting	

Meta-skills

Throughout this unit, learners develop meta-skills to enhance their employability in the computing sector.

Self-management

This meta-skill includes:

- focusing: filtering out non-essential information, sorting information into categories and understanding the relationship between information
- integrity: raising questions of ethics, being self-aware and exercising self-control in reporting insights
- adapting: accepting new ideas and reflecting critically on them, self-educating and responding constructively to change
- initiative: taking responsibility for actions and managing risks in processing data, underpinned by self-belief and trust in their own judgment

Social intelligence

This meta-skill includes:

- communicating: listening to and understanding instructional content, including the directions given in relation to practical work, communicating data insights through reporting and visualisations, and knowing their audience
- collaborating: relationship building and teamworking, as well as social perceptiveness and cultural awareness
- leading: influencing and inspiring others, and sharing expertise and knowledge through coaching and exemplification

Innovation

This meta-skill includes:

- curiosity: exploration of data, noting significant aspects and questioning assumptions
- creativity: generating ideas, constructing solutions and using visualisation in data storytelling
- sense-making: pattern recognition, systematic analysis of data to uncover relationships, and an ability to see the big picture
- critical thinking: deconstructing the data science problem, applying logical thinking and constructing a computational method for its solution

Literacies

Throughout this unit, learners have opportunities to develop their literacy skills.

Numeracy

Learners develop advanced skills in numeracy data analysis methods. They need an understanding of data types, and then they apply numeracy skills to a dataset, such as data transformation and statistical analysis.

Communication

Learners develop communication skills throughout this unit. They receive instruction in various media formats and communicate their findings effectively in their report, which they present.

Digital

Learners develop advanced skills in digital literacy through using and applying digital tools in data processing and analysis, and then creating a report of their findings and any visualisations of data.

Delivery of unit

You should deliver outcomes in a sequential order. We suggest the following distribution of time:

Outcome 1 — Explain the data science workflow
(10 hours)
Outcome 2 — Extract, transform and load data to prepare for analysis
(25 hours)
Outcome 3 — Describe statistical methods relevant to data science
(15 hours)
Outcome 4 — Perform data analysis to obtain insights using statistical tools and methods
(30 hours)
Outcome 5 — Write programs to automate data processing and analysis
(20 hours)
Outcome 6 — Communicate insights using a range of data visualisations and reports
(20 hours)

Learners require access to appropriate hardware and software throughout this unit. You can use a range of software to help learning (see 'Additional guidance' section). This is practical in nature and your focus should be on the acquisition of practical skills in data analysis. You should introduce the required underpinning concepts and theory in context. For example, learners should understand the concepts of data types, data formats, data structures and data models before they apply that knowledge to data modelling and the operations of extracting and transforming data. If learners are in employment, you should give them the opportunity to work with datasets relevant to their work roles.

Teach learners how to extract, clean and transform data, beginning with available software tools such as Weka or Microsoft Excel, and then writing program code to automate these processes. You should also introduce the concepts of data science code libraries and the use of APIs. Throughout this unit, there are opportunities for presentations and classroom discussions, as well as group work.

The treatment of statistical concepts should be as though learners have no prior exposure to probability and statistics. Choose small datasets to illustrate the concepts of summary statistics, distributions, plots and charts, correlation and regression. You can then use these to draw examples of insights into data, leading to the concepts of confidence intervals and hypothesis testing.

In acquiring competence in programming, learners should focus on one high-level language in a selected development environment, but you should also make them aware of current alternatives. There are opportunities for learners to consolidate their learning with practical exercises, such as setting up an IDE, installing appropriate software and utilising version control software (GitHub, Bitbucket). It is not a prerequisite for learners to have programming experience. So, they should take time over the key concepts needed to create efficient code. Learners should be familiar with these concepts to help them write their programs and functions.

As far as possible, you should deliver the preparation of data for visualisation and the creation of data visualisations through practical activities. For the preparation of data, it is helpful to create most visualisations from real data. There are visualisations openly available across many domains for this purpose, as well as open datasets that you can use to create visualisations.

The popularity of data science has resulted in a wide range of resources for learners who want to learn more about statistics as it relates to data analysis.

Additional guidance

The guidance in this section is not mandatory.

Content and context for this unit

You can use a variety of software tools in this unit. We recommend that, if possible, you expose learners to a range of software. For example, you could deliver this unit using a general-purpose programming language, such as Python, a dedicated data analysis toolset, such as Microsoft Excel or Power Query, or a notebook, such as Noteable. Ideally, you should introduce learners to a variety of analysis tools. Whatever tool(s) you use, some degree of automation is needed, so it is unlikely that spreadsheet software alone is enough. At SCQF level 8, we expect learners to become competent in analysing large, complex, unfamiliar datasets.

Explain the data science workflow (outcome 1)

This outcome relates to the data science workflow and, the knowledge and skills statements are self-explanatory. This is a non-technical outcome, with the emphasis on business requirements, business processes and data flows. You should emphasise the importance of understanding business processes, and working with domain experts. Also emphasise the importance of data quality, particularly the time-consuming nature of data cleaning.

In discussion about data quality, you should include data bias and the difficulty of eliminating bias from datasets (particularly historical datasets). Cover the concept of model (algorithmic) bias, and explore the tension between fairness and accuracy.

Extract, transform and load data to prepare for analysis (outcome 2)

This outcome relates to data extraction and transformation, and the knowledge and skills statements are self-explanatory. You should limit modelling to only the most common data models, such as star schema. Whenever possible, learners should use real data from a variety of internal and external sources, rather than artificial datasets.

At SCQF level 8, we expect learners to be able to carry out complex transformations on the source datasets. For example, to understand (and be able to apply) different types of joins to datasets. We also expect them to be aware of the processor, storage and data management implications of working with large datasets.

You should inform learners of the implications of transforming data in terms of reporting it, that they are no longer visualising ‘raw’ data and may need to communicate this to the audience for the resultant data visualisation. This can require an explanation of how the data has been transformed. For each type of transformation, you should teach learners the situations where it can be applied and why the transformation is useful, as well as having an opportunity to practice applying it to data.

Make learners aware that, in addition to cleaning data, it may require additional types of transformation to prepare it for visualisation. Transformation types you can include are:

- conversion, for example converting numbers to percentages and percentages to numbers, or changing a numeric variable to a nominal variable (such as an age to 'adult' or 'child')
- rescaling, such as normalisation or standardisation
- aggregation, for example aggregation of observations containing a date variable by day, week, month, or year, or aggregation of observations prior to applying a summary function such as mean, max or sum
- extraction, for example extracting the 'year' component of a date
- mapping, for example geocoding or converting grid references to latitude and longitude
- transforming continuous data to a normal distribution by applying a log, square, square root or other function to see a relationship 'hidden' in the data

We expect learners to automate data transformations so it can be stored and updated using new data. They can do this in several ways, such as writing code (for example, in Python) or using applied steps (for example, in Power Query).

Describe statistical methods relevant to data science (outcome 3)

This outcome gives learners confidence and competence in performing descriptive statistical analysis, in addition to an understanding of the underlying probabilistic concepts. They develop a familiarity with a software tool for performing statistical analysis. Suitable tools may include generic software, such as Microsoft Excel (with appropriate add-ins), and/or dedicated software, such as SPSS or RStudio.

You should introduce learners to basic probability concepts that underpin statistical concepts, such as inferential statistics and sampling. You should introduce descriptive versus inferential statistics and populations versus samples.

The focus is to introduce learners to a variety of statistical methods that they can apply to a dataset. Use the following statements as a guide for this:

- When introducing the correlation co-efficient calculation, it is essential to emphasise that correlation does not imply causation. Real-life examples of spurious statistics can be very helpful.
- When exploring distribution analysis, an emphasis on the importance of the normal distribution with real-life examples helps to solidify its usefulness.
- When introducing the most common graphs and charts used in statistical analysis, box plots can help learners grasp the basics so that they learn to quickly review data for potential sources of difficulty like outliers and skew.
- You should cover the theory behind classical methods of hypotheses testing.

You need time in this outcome to teach statistical techniques and methods. The treatment of probability can be basic but sufficient for learners to understand the concept behind probability distributions, which should include discrete and continuous distributions (including Gaussian).

Perform data analysis to obtain insights using statistical tools and methods (outcome 4)

This outcome provides an opportunity to introduce exploratory data analysis. Learners take sourced and transformed data, and perform analyses on it.

At SCQF level 8, we expect learners to carry out sophisticated analyses on datasets. You should introduce a range of tools, and tool types, including general-purpose packages (such as Microsoft Excel), dedicated software (such as Power BI), notebooks (such as Noteable) and programming languages (such as Python). We recommend that the outcome concludes with a review of the tools that can be used in data analysis.

Write programs to automate data processing and analysis (outcome 5)

This outcome focuses on exploring the programming languages used to create programs for data analysis. Learners look at the range of languages used in data analysis, such as Python, R, and Julia. Although you should initially expose learners to a range of languages, it is essential that they then focus on acquiring skills in just one programming language.

Learners look at where datasets are stored (such as in text files and spreadsheets) and how they are read into programs, using the programs read functions and libraries. They also explore data types, how data is stored in the language, and how it flows through the program. For example, in Python, learners can use Pandas to replace a Microsoft Excel table with data frames. You should emphasise the use of modularity, using functions, libraries and code re-use.

Help learners appreciate the importance of setting up a proper data programming environment and consider the options for setting up, installing and configuring different development environments. For example, if using Python programming, this could include setting up Anaconda Python distribution as a package and environment manager, and an IDE like Atom or Jupyter Notebooks. If using R programming language, learners can use RStudio as an IDE and Git for source control.

This outcome is practical in nature and involves learners writing a series of functions and programs. They should follow the processes and structures they learned in outcomes 1 and 2 when writing their programs. The functions and programs they write should include:

- reading data from external sources (at least two)
- data cleaning algorithms
- data transformation and encoding
- statistical analyses
- visualisations

Learners set up test data for their programs, writing the code to run the test and reviewing the test after completion.

Communicate insights using a range of data visualisations and reports (outcome 6)

This outcome introduces learners to how data visualisation is used as a tool to help individuals and organisations make informed decisions in a variety of domains, such as business, science and sport. It may be possible to find examples that match the interests and experience of learners. You should give examples of the kinds of questions that can be answered using data visualisations, for a variety of domains.

The focus is on how to create data visualisations and use them to communicate insights to others in a way that informs their decision-making. This centres on data visualisation as a communication tool.

The range of data visualisations used across the outcomes should go beyond the fundamental types (such as bar charts, histograms, scatter plots and line charts) and could include bubble charts, side-by-side box plots, time-series charts, and maps. Additionally, you should introduce faceting and small multiples for some of these visualisation types, as a way of comparing different partitions within a dataset.

The range of types of visualisations chosen should include those from more than two of the following categories:

- Charts that support the comparison of categories, such as donut charts.
- Charts that enable correlations to be explored, such as scatter plots.
- Charts relating to hierarchies and part-to-whole relationships, such as waffle charts.
- Charts that show trends over time, such as a time-series line graph.
- Charts that show spatial patterns, such as a dot map.

The datasets used to illustrate different visualisations:

- should be in a ‘tidy’ (normalised) format
- should be easily accessible — it should be easy for learners to extract the data
- may contain variables of more complex types, such as dates or geospatial data
- should include the information needed to understand them, such as titles and labels

You should show learners several good examples from each of the following forms of data communication:

- Dashboards can be a popular means to communicate data in a business context, particularly for data that relates to KPIs and other business metrics. You should show learners examples of dashboards used in different business functions, including interactive dashboards. Explore the role of interactive dashboards to enable end-users to explore data, rather than simply consume it.
- Charts of the types described above can be found on topical sites such as DataWrapper and news publications such as the BBC.
- Storytelling is the construction of a verbal or written narrative structure that complements one or more data visualisations to make the ‘message’ conveyed by the visualisation(s) more compelling.

Learners should be able to select an appropriate visualisation to create based on the goals and purpose of the communication, the audience and the data. They should make appropriate design choices about their visualisations. You can construct a variety of contexts and purposes for them to create their visualisations.

Using a software package, learners could create the following charts from a variety of simple datasets, for example:

- side-by-side box plots
- donut charts
- scatter plots
- waffle charts
- time-series line graphs
- dot maps

They should select charts that are commonly used and are of a level of complexity appropriate for this level.

Using a software package, learners should create at least one interactive dashboard. Any one of the following features would be suitable for them to make interactive:

- filtering
- sorting
- enabling the user to increase and decrease the level of detail for the data visualised

The range of interactive features available is determined by the visualisation software selected to create the dashboard, for example Tableau, Power BI and Google Data Studio, amongst others, provide these features.

For learners who are competent using Python or R, both programming languages provide very good data visualisation packages. They are the most used programming languages in data science, making them the most suitable languages for data visualisation.

Using storytelling techniques, learners create a ‘story’ about the data and the visualisation(s) for their intended audience. This can be achieved by including some time dimension to the data. For charts that already have a time dimension, such as time-series line graphs, the storytelling element could consist of a narrative built around how and why the line changes over time. For charts that do not have a temporal dimension, learners could use a ‘before’ and ‘after’ visualisation of a dataset, with the ‘storytelling’ element being a narrative constructed about the change between the two visualisations.

Learners should be able to make a recommendation or propose a course of action based on their analysis and justify this by referring to the data.

Approaches to assessment

We recommend that you assess all the outcomes in this unit using evidence collated in an e-portfolio. This allows learners to take ownership of their work, assemble the portfolio themselves, and continuously update it as they gather materials. It is important that the portfolio learners submit for assessment contains items that clearly recognise and record their achievements.

Examples of evidence that learners can use includes:

- screenshots of installing software, interacting with Git, setting up IDEs
- blog posts reflecting on programming languages, different interfaces, for example
- audio or video presentations
- code listings
- documentation
- test plans and results
- created charts
- interaction with source control software

E-portfolios allow different types of electronic evidence to be used for assessment in its original format, offering an assessment approach that is ‘learner-centred’. It offers flexibility for learners to assemble one portfolio and tailor it to specific audiences by tagging items for different purposes (including different assessments). Evidence can be stored in a manner that is more secure and more accessible to learners, teachers, lecturers and verifiers.

E-portfolios make it easier for you to give feedback, strengthening the links between formative and summative assessment, and between learning and assessment.

Equality and inclusion

This unit is designed to be as fair and as accessible as possible with no unnecessary barriers to learning or assessment.

You should take into account the needs of individual learners when planning learning experiences, selecting assessment methods or considering alternative evidence.

Guidance on assessment arrangements for disabled learners and/or those with additional support needs is available on the assessment arrangements web page: www.sqa.org.uk/assessmentarrangements.

Information for learners

Data Science (SCQF level 8)

This section explains:

- what the unit is about
- what you should know or be able to do before you start
- what you need to do during the unit
- opportunities for further learning and employment

Unit information

This unit develops your knowledge and skills in data management, data analysis and visualisation, and programming for data, including statistical techniques. It is suitable if you wish to improve your existing skills in this area. It is also suitable if you are interested in pursuing a career in data analysis or data science.

Before starting, we recommend that you have experience of data analysis. You should be familiar with numerical software (such as spreadsheets) and have well-developed numeracy skills.

During this unit, you work with data sourced from public datasets. You learn how to use data analysis to gain insights into these datasets and make data-driven decisions based on your analysis.

You use a variety of software to carry out your analyses. This might include Microsoft Excel, Power Query, Power BI or Tableau, Jupyter Notebooks or Noteable, or programming languages, such as Python or R.

You learn the stages of the data analysis workflow, beginning with considerations of data quality and bias and any ethical issues that may arise in the use of the data. Techniques for cleaning, transforming and storing data are introduced and practised, along with modelling of the data and its structure.

You are introduced to the probabilistic and statistical foundations behind statistics-driven data analysis and develop your competence in using software to perform statistical analysis. You gain an understanding of probability concepts and how to make statistical inferences. The statistical methods you learn include:

- sampling methods
- descriptive statistics
- correlation and linear regression
- measures of statistical significance
- hypothesis testing

You become familiar with the concepts, principles and purposes around programming for data. The primary focus is on writing good quality, reusable, clean code needed to interrogate data. You learn how to set up a programming environment and a means of tracking versions of a program during development. You also learn how to identify data types and structures and use programming language features and structures to write code that performs a data management or analysis function. You prepare a test plan for each program and record the outcomes from test data, then use these to correct or improve code. The emphasis is on the importance of documenting a program.

You learn that there are various software libraries that can be used within a program to perform data analysis, including descriptive statistics, graphs and plots and statistical analysis.

Finally, you develop an understanding of the importance of communicating the insights that your data analysis has revealed. You achieve this through a combination of data visualisations and data storytelling. You learn the most common visualisations and the contexts in which each is appropriate, including charts, graphs and plots. You also learn how to develop an interactive dashboard to display various features of a dataset.

Throughout this unit, you develop meta-skills covering self-management, social intelligence and innovation.

You can be assessed in several ways, most of which are practical in nature, including practical assignments or case studies. Your knowledge could be assessed using a formal test or by gathering evidence of your work in an e-portfolio.

Administrative information

Published: June 2022 (version 1.0)

Superclass: CB

History of changes

Version	Description of change	Date

Note: please check SQA's website to ensure you are using the most up-to-date version of this document.