**Module learning outcomes assessed by this task:**

1. Apply the principles of programming and data management to solve problems.

2. Demonstrate the use of an object-oriented paradigm when solving software problems.

3. Design and implement algorithms for numerical analysis.

4. Demonstrate the use of proactive error handling techniques to address software reliability and program vulnerability issues.

5. Critique and reflect on alternative solutions to a given problem or on their own work in a constructive way.

6. Undertake independent research activities with relation to innovative approaches to data science problem solving.

7. Demonstrate the use of Data Visualisation techniques for supporting numerical data analysis.

8. Demonstrate the use of a version control system (such as Git) as part of an integrated development process.

# Completing your assessment

**What am I required to do on this assessment?**

For this assessment, you are required to complete three different tasks. A brief outline is given below. Exact details of what is required are given in **Appendix 1**.

1. Develop a set of functions to solve a programming problem using ONLY built-in Python functions and data structures.
2. Perform basic data analysis of a given dataset and identify an “interesting” pattern or trend within the data.
3. Write a reflective report about the process you followed while developing solutions to the two main programming tasks (i.e., 1 & 2 above).

**Where should I start?**

To demonstrate your understanding and programming skills it is important that you develop a sufficient knowledge of the module materials and gain practical experience of coding in Python before you begin this assessment. You should read the detailed description of each task given in **Appendix 1**.

Firstly, you should create a GitHub account and follow the instructions given by the tutor for accessing the GitHub Classroom that has been set up for this assessment. How to complete this will be covered during one of your workshops. In addition, there is a pre-recorded explanation of how to do this available in the Assessment folder on Blackboard. Secondly, you need to clone your GitHub repository to your local machine. Now, you should open a Jupyter Notebook console from Anaconda Navigator and load the Programming Task 1 Template. You can now begin working through the programming requirements set out in Section A.

**What do I need to do to pass?**

To pass this coursework assessment you will need to achieve an overall mark of 50% or above. Realistically, this will not be possible without at least attempting both programming tasks. However, you should make sure to attempt the other task to ensure that you have maximised your mark for this assessment.

**How do I achieve high marks in this assessment?**

High marks can be achieved by carefully following the requirements set out in Appendix 1. Marks will be deducted for solutions which do not follow the requirements precisely. In addition, you should make sure that you demonstrate good coding standards, write an insightful reflective (rather than descriptive) process report, and follow all naming conventions set out in this assessment.

**How does the learning and teaching relate to the assessment?**

Week 1 focuses on Git and so following this material is important for accessing the assessment materials and submitting your work. Weeks 2 through 6 focus on basic Python programming. You should pay particular attention to Week 6 to identify built-in functions. These are important for the first task. Weeks 7 through 9 focus on how to use Python for data analysis and are important for the second task. Week 11’s Data Science demonstration may also be useful for the second task.

**What additional resources may help me complete this assessment?**

Additional resources that you might find useful for completing this assessment include:

* Reflective Writing course at <https://xerte.uwe.ac.uk/play_4988>
* Referencing information at <https://www.uwe.ac.uk/study/study-support/study-skills/referencing>
* Module Discussion Boards: Coursework Queries and FAQs

The Module Leader and Module Tutors will also available via email to clarify any issues you may be having with the assessment. Formative feedback can be requested during the tutorial sessions.

**What do I do if I am concerned about completing this assessment?**

UWE Bristol offer a range of Assessment Support Options that you can explore through [this link](https://www.uwe.ac.uk/study/academic-information/personal-circumstances), and both [Academic Support](https://www.uwe.ac.uk/study/study-support/student-support-advisers) and [Wellbeing Support](https://www.uwe.ac.uk/life/health-and-wellbeing/get-wellbeing-support) are available.

For further information, please see the [Academic Survival Guide](https://www.uwe.ac.uk/study/academic-information/academic-survival-guide).

**How do I avoid an Assessment Offence on this module? 2**

Use the support above if you feel unable to submit your own work for this module. The most common form of Assessment Offense for this type of assessment is copying code from another source (such as a forum, webpage, another student, etc) without referencing (and citing) it correctly. Referencing is an important part of academia, and you should become clear about when you need to reference an external source and how to reference it (more information is available in the study skills link above). However, it should be made clear that any copied code may result in no marks for any sub-task in which it is used.

During the marking phase, an analysis of submissions will be made across the cohort to identify any evidence of collusion and/or plagiarism.

* + Avoid copying and pasting any work into this assessment, including your own previous assessments, work from other students or internet sources.
  + Develop your own style, arguments, and wording, so avoid copying sources and changing individual words but keeping, essentially, the same sentences and/or structures from other sources.
  + Never give your work to others who may copy it
  + If an individual assessment, develop your own work and preparation, and do not allow anyone to make amends on your work (including proof-readers, who may highlight issues but not edit the work)

**When submitting your work, you will be required to confirm that the work is your own.**

# Marks and Feedback

**Your assessment will be marked according to the marking criteria set out in each task in Appendix 1. You can use these to evaluate your own work before you submit.**

# Appendix 1 – Assessment Overview

This single coursework assessment involves three separate tasks. The requirements for each task are detailed below together with deliverables, submission details and grading criteria. Below is a breakdown of percentage weighting per task:

|  |  |
| --- | --- |
| Task | % Weighting |
| Programming Task 1 | 48 |
| Programming Task 2 | 38 |
| Process Development Report | 14 |
| Total | **100** |

## Section A. Programming Task 1

* This programming task focuses on using Python to calculate a set of Kendall Tau Correlation Coefficients for a given dataset using ONLY built-in functions and data structures.
  + For Programming Task 1, you **MUST NOT** import any Python library functions. This means you cannot use Python modules such as *math*, *SciPy, csv* or libraries such as *Pandas* or *NumPy*.
* To print the Kendall Tau Rank Correlation Coefficient for a given pair of Python Lists, it would be very easy to use the ***kendalltau()*** function provided in the SciPy library. However, this programming task is designed to assess your coding abilities and by preventing you from using this function you are forced to gain a deeper understanding of how to complete that task. To do this, you will need to develop your own algorithm. Try typing “calculate Kendall Tau Correlation Coefficient by hand” into your favourite search engine.
* There is a single data file available in your GitHub repository for use in this programming task. The file contains data about Google health term searches in a set of US districts for 2017.
  + The data file is called ***task1.csv***. This CSV file includes a header row with multiple named data values.
* Students are expected to follow appropriate coding standards such as code commenting, docstrings, consistent identifier naming, code readability, and appropriate use of data structures.

### A.1. Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Requirement | Description | Marks Available |
| FR1 | Develop a function to read a single specified column of data from a CSV file | The function should accept two parameters: the data file name and a column number. The column number specifies which of the columns to read. It can range between 0 and n-1 (where n is the number of columns). The function should return two values: the column name and a List containing all the specified column’s data values. You should use the ***task1.csv*** data file to test your function but your function should also work for other CSV files. An illustration of this is given in Appendix 2. | **6** |
| FR2 | Develop a function to read CSV data from a file into memory | The ***task1.csv*** data file contains several columns of data values. This function should accept a single parameter: the data file name. It should make use of the function developed in FR1 to read all columns of data from the data file and add them to a Dictionary data structure. The Dictionary should contain one entry for each column in the CSV data file. An illustration of this is given in Appendix 3. | **6** |
| FR3 | Develop a function to calculate the Kendall Tau Rank Correlation Coefficient for two lists of data | This function should calculate the Kendall Tau Rank Correlation Coefficient for two lists of data. The function should take two lists of data (of equal length) as parameters. The function should ensure that the lists are of equal length otherwise raise an error. The function should return the calculated coefficient value. | **12** |
| FR4 | Develop a function to generate a set of Kendall Rank Correlation Coefficients for a given data file | The function should accept one parameter: the Dictionary data structure generated in FR2. This function should make use of the function developed in FR3 to generate a Kendall Tau Rank Correlation Coefficient for every pair of columns in the input data structure parameter. The function should return a list of tuples, each tuple containing the two column names and associated correlation coefficient value. An illustration of this is given in Appendix 4. | **10** |
| FR5 | Develop a function to print a custom table | This function should output the Kendall Tau Rank Correlation Coefficient for a subset of the column pairs generated in FR4. The function should take three parameters: list of correlation coefficient tuples, border character to use and which columns to include. High marks will be given for good use of padding in the table cells to improve readability. An illustration of this is given in Appendix 5. | **9** |

### A.2. Deliverables

* A Jupyter Notebook file (in .ipynb format) containing a complete solution to this Programming Task.
  + You must use the template provided[[1]](#footnote-1).

### A.3. Submission

* You should commit your completed Jupyter Notebook file to your GitHub repository with an appropriate commit message.

### A.4. Grading Criteria

* Marks are allocated as follows:
  + up to 43 marks for the Python code solution
    - Marks will be awarded for each requirement **according to the level of completion**.
    - To gain high marks you must follow the requirement instruction precisely.
  + up to 5 marks for adherence to good coding standards.

## Section B. Programming Task 2

* This programming task focuses on using NumPy/SciPy, Pandas, and Matplotlib/Seaborn to combine and analyse two datasets related to mental health.
* Two data files have been provided in your GitHub repository for this task. These data files provide some real data about mental health from around the world from 1990 through to 2017.
  + The ***task2a.csv*** data file contains the prevalence of different types of mental health disorders for each country, region, and continent around the world including schizophrenia, depression, bipolar and anxiety.
  + The ***task2b.csv*** data file contains the prevalence of drug and alcohol dependence for each country, region, and continent around the world.
* Students are expected to follow appropriate coding standards such as code commenting, consistent identifier naming, code readability, and appropriate use of data structures.

### B.1. Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Requirement | Description | Marks Available |
| FR6 | Read CSV data from two files and merge it into a single Data Frame | For this task you should use the ***task2a.csv*** and ***task2b.csv*** data files. | **4** |
| FR7 | Explore the dataset to identify an "interesting" pattern or trend[[2]](#footnote-2) | Use an appropriate visualisation tool (such as Matplotlib or Seaborn) to illustrate your exploration. You should include at least three visualisations as part of your exploration. You could consider other ways to explore the data such as data summaries or transformations. You must include an explanation of the dataset exploration, your selected "interesting" pattern or trend and your reasons for selecting it. | **10** |
| FR8 | Detect and remove any outliers in the data used for your "interesting" pattern or trend | Using an appropriate technique to detect and remove any outliers in the data used for your "interesting" pattern or trend. You must include an explanation of the detection method used, how it works, and the any outliers detected. NOTE: there may not be any detectable outliers using the selected detection method – if this is the case, please state this clearly in the explanation given. | **6** |
| FR9 | Define a hypothesis to test your “interesting” pattern or trend | Using an appropriate hypothesis testing formulation to define a hypothesis and provide an explanation for your choices. | **6** |
| FR10 | Test your hypothesis with statistical significance level of 0.05 | Using an appropriate Python library, test the hypothesis stated in FR9. You must include a detailed explanation of your findings to achieve good marks for this task. | **7** |

### B.2. Deliverables

* A Jupyter Notebook file (in .ipynb format) containing a complete solution to this Programming Task.
  + You must use the template provided [[3]](#footnote-3).

### B.3. Submission

* You should commit your completed Jupyter Notebook file to your GitHub repository with an appropriate commit message.

### B.4. Grading Criteria

* Marks are allocated as follows:
  + up to 33 marks for the Python code solution
    - Marks will be awarded for each requirement **according to the level of completion**.
    - To gain high marks you must follow the requirement instruction precisely.
  + up to 5 marks for adherence to good coding standards.

## Section C. Process Development Report

* You are expected to identify the strengths/weaknesses of your approach to your coding tasks.
* For this coursework, you must write a reflective report which focuses on the process you took to develop a solution to the two programming tasks described in Section A and Section B above. Please reflect on your experiences rather than simply describing what you did.
* The report must be split into TWO different sections – one for each programming task.
* Each section should:
  + include an explanation of how you approached the task:
    - describe your thought process.
    - did you find it easy or difficult? Why?
    - what problems did you encounter? How did you overcome them?
  + identify any strengths/weaknesses of the approaches used.
  + consider how the approaches used could be improved.
  + suggest alternative approaches that could have been taken instead of the ones you used.

### C.1. Requirements

* The development process report MUST be submitted in .docx format – pdf, pages, or any other file format will NOT be accepted for this task.
* The report must not exceed 800 words. Please indicate the word count at the end of the document.

### C.2. Deliverables

* A development process report written in .docx format.

### C.3. Submission

* You should commit the report to your GitHub repository with an appropriate commit message.

### D.4. Grading Criteria

* There are 14 marks available for the report – 7 marks per section.
  + Marks will be awarded for appropriate use of technical language, critical reflection on development process and quality of engagement with the reflective process.

# Appendix 2 – Example Column Extraction

For the following illustration, you should assume that the column number parameter is equal to ***1*** for the data file. There are 9 columns in this file and so column number can range between 0 and 8. For this data, the function would return two values: ***“Glucose”*** and ***[148,85,183,89,137,116,78,115,197,125,110,168,139]***

# Table Description automatically generated with medium confidence

# Appendix 3 – In-Memory Data Structure

Using the file illustrated in Appendix 2, the Dictionary produced in FR2 should look something like the illustration below. However, you must ensure that your function can work for any CSV file with a similar structure (such as a file with 5 columns and 100 rows or with 20 columns and 1000 rows).

**{**

**"Pregnancies" : [6,1,8,1,0,5,3,10,2,8,4,10,10],**

**"Glucose" : [148,85,183,89,137,116,78,115,197,125,110,168,139],**

**"BloodPressure" : [72,66,64,66,40,74,50,0,70,96,92,74,80],**

**"SkinThickness" : [35,29,0,23,35,0,32,0,45,0,0,0,0],**

**"Insulin" : [0,0,0,94,168,0,88,0,543,0,0,0,0],**

**"BMI" : [33.6,26.6,23.3,28.1,43.1,25.6,31,35.3,30.5,0,37.6,38,27.1],**

**"DiabetesPedigreeFunction" : [0.627,0.351,0.672,0.167,2.288,0.201, 0.248,0.134,0.158,0.232,0.191,0.537,1.441],**

**"Age" : [50,31,32,21,33,30,26,29,53,54,30,34,57],**

**"Outcome" : [1,0,1,0,1,0,1,0,1,1,0,1,0]**

**}**

# Appendix 4 – Statistical data based on In-Memory Data Structure

Using the in-memory data structure illustrated in Appendix 3, the List of Tuples produced in FR4 should look something like the illustration below. The full data output is too large to include here and so only some of the data has been included to help illustrate what is required. Remember that different CSV data files will result in different data being stored. The data file you have been provided with does not include any of the data shown below. Don’t be tempted to simply copy the result below into your Jupyter Notebook.

**[**

**("Pregnancies", "Glucose", 0.337),**

**("Pregnancies", "BloodPressure", -0.0025),**

**("Pregnancies", "SkinThickness", -0.7481),**

**("Pregnancies", "Insulin", -0.4772),**

**("Pregnancies", "BMI", -0.2313),**

**("Pregnancies", "DiabetesPedigreeFunction", -0.0872),**

**("Pregnancies", "Age", 0.3428),**

**("Pregnancies", "Outcome", 0.0167),**

**("Glucose", "Pregnancies", 0.337),**

**("Glucose", "BloodPressure", 0.1429),**

**("Glucose", "SkinThickness", -0.0028),**

**("Glucose", "Insulin", 0.4304),**

**("Glucose", "BMI", 0.0584),**

**("Glucose", "DiabetesPedigreeFunction", 0.2192),**

**("Glucose", "Age", 0.5328),**

**("Glucose", "Outcome", 0.5465),**

**+++++++ More data would be included here ++++++++**

**("Outcome", "Pregnancies", 0.0167),**

**("Outcome", "Glucose", 0.5465),**

**("Outcome", "BloodPressure", 0.0755),**

**("Outcome", "SkinThickness", 0.3585),**

**("Outcome", "Insulin", 0.3355),**

**("Outcome", "BMI", -0.0768),**

**("Outcome", "DiabetesPedigreeFunction", 0.2185),**

**("Outcome", "Age", 0.314)**

**]**

# Appendix 5 – Output table for Statistics

Using the output from the function produced in FR4, the following table outputs a subset of the available columns (as defined by the function parameter) using the border character **\*** and padding within the cells to ensure the table is readable:

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\* Glucose \* BloodPressure \* BMI \* Age \***

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\* Glucose \* - \* 0.1429 \* 0.0584 \* 0.5328 \***

**\* BloodPressure \* 0.1429 \* - \* -0.4522 \* 0.4194 \***

**\* BMI \* 0.0584 \* -0.4522 \* - \* -0.3847 \***

**\* Age \* 0.5328 \* 0.4194 \* -0.3847 \* - \***

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

1. There is a Jupyter Notebook template available in your GitHub repository - ***UFCFVQ-15-M\_Programming\_Task 1\_Template.ipynb*** [↑](#footnote-ref-1)
2. An “interesting” pattern or trend might include a correlation between two columns of data, equality of two columns of data or estimating a linear or non-linear relationship between columns of data. [↑](#footnote-ref-2)
3. There is a Jupyter Notebook template available in your GitHub repository - ***UFCFVQ-15-M\_Programming\_Task\_2\_Template.ipynb*** [↑](#footnote-ref-3)