 School of Computing and Creative Technologies

**Assessment Specification**

## Module Details

|  |  |
| --- | --- |
| **Module Code** | UFCFVQ-15-M |
| **Module Title** | Programming for Data Science |
| **Module Leader** | Dr David Wyatt; Dr Jan Van lent |
| **Module Tutors** | Dr Eman Qaddoumi; Dr Mahmoud Elbattah |
| **Year** | 2024 |
| **Task** | 1 |
| **Total number of assessments for this module** | 1 |
| **Weighting** | 100% |

## Dates

|  |  |
| --- | --- |
| **Date issued to students** | 4th November 2024 |
| **Date to be returned to students** | 13th February 2025 |
| **Submission Date** | 16th January 2025 |
| **Submission Place** | GitHub Classroom |
| **Submission Time** | 2pm |
| **Submission Notes** | This assessment qualifies for a 48-hour late submission window extension, i.e. 2pm on 18th January 2025. |

## Feedback

|  |  |
| --- | --- |
| **Feedback provision will be** | Annotated Jupyter Notebook submissions committed to student’s GitHub Repository |

# Contents

[Section 1: Overview of Assessment 2](#_Toc172199733)

[Section 2: Task Specification 3](#_Toc172199734)

[Section 3: Deliverables 8](#_Toc172199735)

[Section 4: Marking Criteria 8](#_Toc172199736)

[Section 5: Feedback mechanisms 10](#_Toc172199737)

[Section 6: Appendices 10](#_Toc172199738)

# 

# Section 1: Overview of Assessment

This assessment assesses the following module learning outcomes:

MO1 Apply the principles of programming and data management to solve problems.

MO2 Demonstrate the use of an object-oriented paradigm when solving software problems.

MO3 Design and implement algorithms for numerical analysis.

MO4 Demonstrate the use of proactive error handling techniques to address software reliability and program vulnerability issues.

MO5 Critique and reflect on alternative solutions to a given problem or on their own work in a constructive way.

MO6 Undertake independent research activities with relation to innovative approaches to data science problem solving.

MO7 Demonstrate the use of Data Visualisation techniques for supporting numerical data analysis.

MO8 Demonstrate the use of a version control system (such as Git) as part of an integrated development process.

The assessment is worth 100% of the overall mark for the module. The assessment is made up of three different parts:

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 sets of programming tasks listed above

The assessment is described in more detail in section 2.

This is an individual assessment.

Working on this assessment will help you to practise your Python programming skills and use your knowledge of Data Science libraries to perform an investigation of a dataset to find an interesting pattern or trend. If you have questions about this assessment, please post them to the discussion board on Blackboard.

# Section 2: Task Specification

This assessment is split into three parts.

The first two parts include a set of related programming tasks for you to complete. You are expected to follow appropriate coding standards such as code commenting, use of docstrings, consistent identifier naming, code readability, and appropriate use of data structures. You will be provided with a template document for each part.

The final part requires you to reflect on your experiences working on this assessment making extensive use of your reflective journal to write a report to identify any strengths/weaknesses of your approach to your coding tasks. A template document will be provided for you to use for this part of the assessment.

Part 1 is expected to take around 14 hours to complete. Part 2 is expected to take approximately 12 hours to complete. Part 3 is expected to take approximately 4 hours to complete.

All the files required to complete this assessment are available through the GitHub Classroom interface. To access them you will need to create a GitHub account, join the GitHub Classroom set-up for this course and clone the GitHub repository created for you by GitHub Classroom to your local machine. You will be shown how to complete this during one of the workshop sessions.

## Part 1

* This part focuses on using Python to calculate a set of Pearson Correlation Coefficients for a given dataset using ONLY built-in functions and data structures.
  + For Part 1, you **MUST NOT** import any Python library functions. This means you cannot use Python built-in modules such as *math*, *csv* or external libraries such as *SciPy*, *Pandas* or *NumPy*.
* To print the Pearson Correlation Coefficient for a given pair of Python Lists, it would be very easy to use the ***pearsonr()*** function provided in the SciPy library. However, this part of the assessment 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 the task. You will need to develop your own algorithm. Try typing “calculate Pearson Correlation Coefficient by hand” into your favourite search engine.
* There is a single CSV data file available in your GitHub repository for use in this part of the assessment - ***part\_1.csv***. The file contains a record of US police criminal incidents for the year 2015.
* In addition, you have been provided with a Jupyter Notebook template, ***Part\_1.ipynb***, which MUST be used to complete the 5 tasks described below.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Requirement | Description | Marks Available |
| Task 1 | 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 ***part\_1.csv*** data file to test your function but your function should also work for other CSV files. An illustration of this is given in the *Column Extraction Example* section below. | **8** |
| Task 2 | Develop a function to read CSV data from a file into memory | The ***part\_1.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 Task 1 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 the *In-Memory Data Structure Example* section below. | **8** |
| Task 3 | Develop a function to calculate the Pearson Correlation Coefficient for two lists of data | This function should calculate the Pearson 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** |
| Task 4 | Develop a function to generate a set of Pearson Correlation Coefficients for a given data file | The function should accept one parameter: a Dictionary data structure, such as the one generated in Task 2. This function should make use of the function developed in Task 3 to generate a Pearson 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 the *Statistical data based on In-Memory Data Structure Example* section below. | **8** |
| Task 5 | Develop a function to print a custom table | This function should output the Pearson Correlation Coefficient for a subset of the column pairs generated in Task 4. The function should take three parameters: a list of correlation coefficient tuples, a border character to use and a list of 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 the *Output table for Statistics Example* section below. | **10** |

### Column Extraction Example

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 numbers 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

### In-Memory Data Structure Example

Using the file illustrated above, the Dictionary produced in Task 2 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]**

**}**

### Statistical data based on In-Memory Data Structure Example

Using the in-memory data structure illustrated above, the List of Tuples produced in Task 4 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. Do not 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)**

**]**

### Output table for Statistics Example

Using the output from the function produced in Task 4, 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 \* - \***

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

## Part 2

* This part focuses on using NumPy/SciPy, Pandas, and Matplotlib/Seaborn to combine and analyse two datasets related to click events for a set of OU students.
* Two data files have been provided in your GitHub repository for this part - ***part\_2a.csv*** and ***part\_2b.csv*** . These data files provide some real data from the Open University.
  + The ***part\_2a.csv*** data file contains background information about 26746 students including gender, age, disability status and score.
  + The ***part\_2b.csv*** data file contains information about the number of click events made by 26074 students using the University’s Virtual Learning Environment (VLE) system.
* In addition, you have been provided with a Jupyter Notebook template, ***Part\_2.ipynb***, which MUST be used to complete the 3 tasks described below.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Requirement | Description | Marks Available |
| Task 6 | Explore the dataset to identify an "interesting" pattern or trend[[1]](#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. | **16** |
| Task 7 | 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. | **8** |
| Task 8 | Test your hypothesis with statistical significance level of 0.05 | Define a hypothesis using an appropriate hypothesis testing formulation. Provide an explanation for your choice. Using an appropriate Python library, test this hypothesis. You must include a detailed explanation of your findings to achieve good marks for this task. | **10** |

## Part 3

* You are expected to identify the strengths/weaknesses of your approach to your coding tasks.
* For this assessment, you must write a reflective report which focuses on the process you took to develop a solution to the two sets of programming tasks described in Part 1 and Part 2 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 part.
* Each section should:
  + include an explanation of how you approached the set of 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 approach used.
  + consider how the approach used could be improved.
  + suggest alternative approaches that could have been taken instead of the ones you used.
* You should make extensive use of your reflective journal when writing this report.
* The report must not exceed 1500 words. Please indicate the word count at the end of the document.
* In addition, you are provided with a Microsoft Word template, ***Reflection.docx***, which should be used to complete Part 3 of the assessment.

# Section 3: Deliverables

|  |  |  |
| --- | --- | --- |
| Item | Detail | Date & Submission Mechanism |
| Part\_1 Jupyter Notebook | You must use the Jupyter Notebook template provided to complete your Part 1 programming tasks. You MUST NOT delete text cells containing any red text. These are required for marking. | GitHub Classroom |
| Part\_2 Jupyter Notebook | You must use the Jupyter Notebook template provided to complete your Part 2 programming tasks. You MUST NOT delete text cells containing any red text. These are required for marking. | GitHub Classroom |
| Reflective Report | You must use the Word document template provided for this part of the assessment. The report must not exceed 1500 words. Please indicate the word count at the end of the document. | GitHub Classroom |

# Section 4: Marking Criteria

Below is a breakdown of percentage weighting per component of this assessment:

|  |  |
| --- | --- |
| Description | % Weighting |
| Part 1 tasks | 46 |
| Part 2 tasks | 34 |
| Reflective Report | 20 |
| Total | **100** |

This assessment has been structured to help you understand exactly what is required for each task in each part of the assessment. Higher marks can only be gained by ensuring that you meet all the requirements detailed in this document. It is therefore important that you read this document very carefully. If you are unsure what is expected for any task, then please contact a tutor as soon as possible to clarify.

| **Percentage mark** | **86-100** | **70-85** | **60-69** | **50-59** | **40-49** | **30-39** | **0-30** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Overall Descriptor** | **Outstanding** | **Excellent** | **Very good** | **Good** | **Adequate** | **Poor/Inadequate** | **Very poor** |
| **Assessment criteria** |  |  |  |  |  |  |  |
| **Task 1**  **(8%)**  **Relates to MO1 & MO3** | All task requirements met in full including good error handling and clear readable code | Parameters/return values correct; includes all required functionality and some error handling | Parameters/return values correct; includes all required functionality | Parameters/return values mostly correct; includes most required functionality | Incorrect parameters/ return values; includes only part of required functionality | Parameters missing; return values missing; faulty functionality; missing function def | Uses an external library; runtime error when executed |
| **Task 2**  **(8%)**  **Relates to MO1** | All task requirements met in full including good error handling and clear readable code | Parameter/return value correct; includes all required functionality and some error handling | Parameter/return value correct; includes all required functionality | Parameter/return value mostly correct; includes most required functionality | Incorrect parameter/ return value; includes only part of required functionality | Parameter missing; return value missing; faulty functionality; missing function def | Use of an external library; runtime error when executed |
| **Task 3**  **(12%)**  **Relates to MO3 & MO4** | All task requirements met in full including good error handling and clear readable code | Parameter/return value correct; includes all required functionality and good error handling | Parameter/return value correct; includes all required functionality and some error handling | Parameter/return value mostly correct; includes most required functionality | Incorrect parameter/ return value; includes only part of required functionality | Parameter missing; return value missing; faulty functionality; missing function def | Use of an external library; runtime error when executed |
| **Task 4**  **(8%)**  **Relates to MO1** | All task requirements met in full including good error handling and clear readable code | Parameters/return value correct; includes all required functionality and some error handling | Parameter/return value correct; includes all required functionality | Parameter/return value mostly correct; includes most required functionality | Incorrect parameter/ return value; includes only part of required functionality | Parameter missing; return value missing; faulty functionality; missing function def | Use of an external library; runtime error when executed |
| **Task 5**  **(10%)**  **Relates to MO7** | All task requirements met in full including good error handling, and clear readable code | Parameters/return values correct; includes all required functionality; good use of padding | Parameters/return values correct; includes all required functionality; attempted use of padding | Parameters/return values mostly correct; includes most required functionality | Incorrect parameters/ return values; includes only part of required functionality | Parameters missing; return values missing; faulty functionality; missing function def | Use of an external library; runtime error when executed |
| **Task 6**  **(16%)**  **Relates to MO2 & MO7** | At least 3 plots and an excellent presentation of the dataset exploration; a detailed insightful explanation of a trend or pattern; an excellent data analysis which illustrates a deeper understanding | At least 3 plots and clear presentation of the dataset exploration; an insightful explanation of a trend or pattern; an excellent data analysis which illustrates a good depth of understanding | At least 2 plots and clear presentation of the dataset exploration; good explanation of a trend or pattern; a good data analysis but lacks sufficient depth for higher marks | At least 2 plots and a possibly confused presentation of the dataset exploration; basic explanation of a trend or pattern with little depth of understanding demonstrated | At least 2 plots with little attempt to explore dataset; basic explanation which fails to clearly identify a trend or pattern, or the interpretation given is faulty | A single poor plot is provided with no explanation; poor attempt to identify a trend or pattern; dataset incorrectly constructed from raw files | No plots are provided; no explanations provided; dataset incorrectly constructed from raw files |
| **Task 7**  **(8%)**  **Relates to MO2 & MO6** | Excellent attempt to detect/remove outliers; these datapoints are illustrated; alternative methods are considered | Excellent attempt to detect/ remove outliers; detection method used is clearly explained | Good attempt to detect/ remove outliers; detection method used is named but is not explained | Good attempt to detect outliers but these are not removed; detection method used is named but is not explained | A basic attempt to detect and remove outliers, but the method used is not named or explained | A faulty attempt to detect and/or remove outliers; no explanation given of the method used | Little or no attempt to detect or remove outliers was made |
| **Task 8**  **(10%)**  **Relates to MO2, MO3 & M06** | Appropriate hypothesis with good explanation; appropriate test with insightful interpretation | Appropriate hypothesis with good explanation; appropriate test with good interpretation | Appropriate hypothesis with good explanation; appropriate test with basic interpretation | Appropriate hypothesis with basic explanation; appropriate test with poor interpretation | Appropriate hypothesis with poor explanation; inappropriate test | Inappropriate hypothesis with no explanation; inappropriate test | No hypothesis is provided; inappropriate or no test used |
| **Reflective Report**  **(20%)**  **Relates to MO5** | Insightful reflections on experiences, overcoming problems and alternative approaches; excellent use of reflective journal | Insightful reflections on your practise/experiences; alternative approaches considered; good use of reflective journal | Mostly reflective about improving practise and learning experiences; good use of reflective journal | A mixture of reflection/ description; some consideration given to improving practise; some use of reflective journal | Includes some elements of reflection, but lacks consideration on how to improve practise | Mostly descriptive; lacks a reflection on your thought process and approach to the assessment tasks | A descriptive report; fails to identify strengths/ weaknesses |

# Section 5: Feedback mechanisms

The main summative feedback for your work will be given after the submission date. This feedback will suggest ways that you might have improved your submission. Each task in the assessment is marked individually and feedback provided for that task. By providing feedback on a task-by-task basis a clearer link can be made between the feedback given and the solution provided. You can also request formative feedback from module tutors during the final two weeks of the course.

# Section 6: Appendices

## Completing your assessment

**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.

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. You will be shown how to complete during one of your workshop sessions. 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. Your cloned repository includes the following assessment files:

* a **Part\_1.ipynb** Jupyter Notebook file
* a **Part\_2.ipynb** Jupyter Notebook file
* a **Reflection.docx** template file
* a **part\_1.csv** data file
* a **part\_2a.csv** data file
* a **part\_2b.csv** data file

Now, you should use a Jupyter Notebook interface, such as Jupyter Lab, to open the **Part\_1.ipynb** file. You can now begin working through the programming tasks set out in section Part 1.

**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 parts. However, you should make sure to attempt the third part 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 the Task Specification section. 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) report, and follow all naming conventions set out in this assessment.

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

Weeks 1 through 3 focus on basic Python programming. You should pay particular attention to Week 3 to identify built-in functions when working on the first part of the assessment. Weeks 4 through 6 focus on how to use Python for data analysis and are important for the second part.

**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
* Module Reading List at <https://rl.talis.com/3/uwe/lists/81EFE26F-4E45-CC0A-28F3-DCA8DAE62245.html>

The Module Leader and Module Tutors will also available via email to clarify any issues you may be having with the assessment.

**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).

## Assessment Content

In line with UWE Bristol’s [Assessment Content Limit Policy](https://www.uwe.ac.uk/about/structure-and-governance/policies) (formerly the Word Count Policy), word count includes all text, including (but not limited to): the main body of text (including headings), all citations (both in and out of brackets), text boxes, tables and graphs, figures and diagrams, quotes, lists.

## Assessment Offences

**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 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.

UWE Bristol’s [UWE’s Assessment Offences Policy](https://www.uwe.ac.uk/study/academic-information/assessments/assessment-offences) requires that you submit work that is entirely your own and reflects your own learning, so it is important to:

* + Ensure you reference all sources used, using the [UWE Harvard](https://www.uwe.ac.uk/study/study-support/study-skills/referencing/uwe-bristol-harvard) and the guidance available on [UWE’s Study Skills referencing pages](https://www.uwe.ac.uk/study/study-support/study-skills/referencing).
  + 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,** and text-matching software and other methods are routinely used to check submissions against other submissions to the university and internet sources. Details of what constitutes plagiarism and how to avoid it can be found on UWE’s Study Skills [pages about avoiding plagiarism](https://www.uwe.ac.uk/study/study-support/study-skills/reading-and-writing/plagiarism).

1. An “interesting” pattern or trend might include a correlation between two columns of data, equality of two columns of data or an approximate linear or non-linear relationship between columns of data. [↑](#footnote-ref-2)