|  |  |  |  |
| --- | --- | --- | --- |
| **COMP1555 (2016/17)** | **Computer Algorithms and Modelling** | **Faculty Header ID: 300034** | **Contribution: 50% of course** |
| **Course Leader: Dr Cos Ierotheou** | **A Statistical Model Using Numerical Integration** |  | **Deadline Date: Thursday 01/12/2016** |
| This coursework should take an average student who is up-to-date with tutorial work approximately 25 hours   Feedback and grades are normally made available within 15 working days of the coursework deadline | | | |
| **Learning Outcomes:**  A. Select and employ appropriate data structures. B. Formulate and solve elementary mathematical models. C. Obtain programmatic solutions using appropriate software, including a high level programming language. D. Describe and discuss the efficiency, complexity, accuracy and limitations of algorithms. | | | |

|  |
| --- |
| Plagiarism is presenting somebody else's work as your own. It includes: copying information directly from the Web or books without referencing the material; submitting joint coursework as an individual effort; copying another student's coursework; stealing coursework from another student and submitting it as your own work.  Suspected plagiarism will be investigated and if found to have occurred will be dealt with according to the procedures set down by the University. Please see your student handbook for further details of what is / isn't plagiarism.   **All material copied or amended from any source (e.g. internet, books) must be referenced correctly according to the reference style you are using.   Your work will be submitted for plagiarism checking.  Any attempt to bypass our plagiarism detection systems will be treated as a severe Assessment Offence.** |

#### Coursework Submission Requirements

#### An electronic copy of your work for this coursework must be fully uploaded on the Deadline Date of Thursday 01/12/2016 using the link on the coursework Moodle page for COMP1555.

#### For this coursework you must submit a single PDF document.  In general, any text in the document must not be an image (i.e. must not be scanned) and would normally be generated from other documents (e.g. MS Office using "Save As .. PDF"). An exception to this is hand written mathematical notation, but when scanning do ensure the file size is not excessive.

#### For this coursework you must also upload a single ****ZIP**** file containing supporting evidence.

#### There are limits on the file size (see the relevant course Moodle page).

#### Make sure that any files you upload are virus-free and not protected by a password or corrupted otherwise they will be treated as null submissions.

#### Your work will not be printed in colour. Please ensure that any pages with colour are acceptable when printed in Black and White.

#### You must NOT submit a paper copy of this coursework.

#### All courseworks must be submitted as above. Under no circumstances can they be accepted by academic staff

**The University website has details of the current Coursework Regulations, including details of penalties for late submission, procedures for Extenuating Circumstances, and penalties for Assessment Offences.  See** [**http://www2.gre.ac.uk/current-students/regs**](http://www2.gre.ac.uk/current-students/regs)

#### Detailed Specification

#### This coursework is to be done individually.

#### See below for details of the assessment itself.

#### Deliverables

#### The main electronic upload should be a pdf document containing the type written report. Any additional supporting work as part of the solution (e.g. computer program) should be uploaded as a zip file.

#### Grading Criteria

|  |  |
| --- | --- |
| Marks will be distributed to reflect: | |
| 0-30 | An unsatisfactory attempt to the problem. |
| 31-39 | A substandard submission which has no critical appraisal of the concepts of numerical integration and which contains little reporting/analysis of the results. |
| 40-49 | A satisfactory submission with reasonable understanding of the concept of numerical integration, how to implement such techniques and a basic analysis of results. |
| 50-59 | A good submission which includes a good understanding of the concept of design, implementation and documentation together with some analysis. |
| 60-69 | A very good submission that clearly demonstrates that the student knows and understands how to design, implement and document software together with some critical analysis of the results and algorithms used. |
| 70-79 | An excellent submission which demonstrates a clear understanding of problem solving and computer programming. Evidence derivation and presentation of all necessary approaches including design, implementation, documentation and clear reporting. |
| 80+ | An outstanding submission which demonstrates a very high level of g understanding. The work examined is exemplary and provides clear evidence of a complete grasp of the problem solving and programming skills. There is also ample excellent evidence of derivation and presentation of all necessary approaches including design, implementation, documentation and clear reporting. |

#### Assessment Criteria

This is graded as a percentage. See breakdown of marks in the specification below.

**Computer Algorithms and Modelling**

**Coursework 2016/17**

**A Statistical Model Using Numerical Integration**

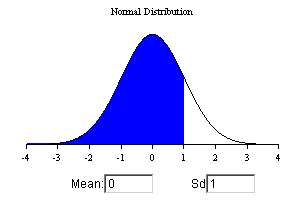
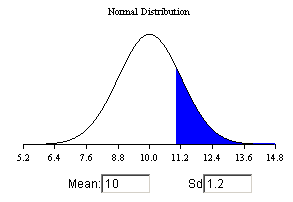
#### This coursework is to be done individually

An important computation used in Statistics involves the **normal distribution**. This distribution is represented by the following probability density function *P(x)*

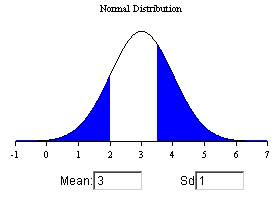
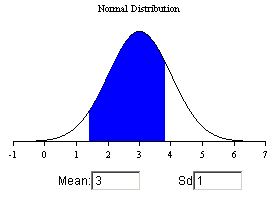


A graph of the function *P(x)* is shown in the figure below for a mean *μ*=1 and a standard deviation *σ*=0. This is called the **standard normal distribution**.

The area under the curve of a normal distribution is particularly important as it is used in well-known statistical tests. As the area represents a probability the value lies between 0 and 1. Below are examples of a number of different types of area that can be computed:



probability that x>=11 probability that x<=1

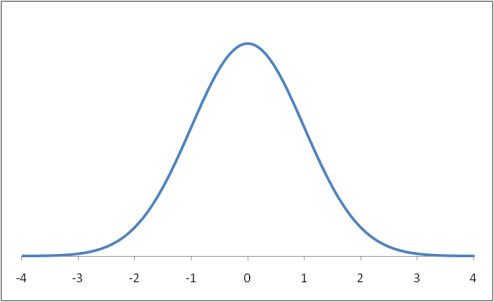


probability that x>=1.4 and x<=3.8 probability that x<=2 and x>=3.5

Alternatively, some statisticians use standard tables to determine the area under the curve for the standard normal probabilities. These contain the area less than a given value of x, but because this is applied to the standard normal distribution, the variable z is used. There is a standard formula using the mean (μ) and the standard deviation (σ) for switching between x and z given by

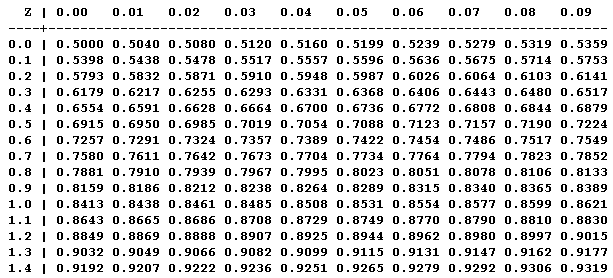


The resulting variable z is based on a mean of zero (μ=0) and the unit standard deviation (σ=1). Below is a section of a z-table



*P(z<1.43)*

*z=1.43*



Here is an example to illustrate how to read the information from the table:

Q. *What is the probability that z<1.43?*

1. This is found by looking down the left column for 1.4 and looking along the top row for 0.03 (1.4+0.03=1.43) where the row and column meet in the table is the area and hence the probability z<1.43 = 0.9236.

You’ve been asked by the Statistics Group to provide them with an application that provides information to them about the normal distribution. In your solution, you need to pay attention to the ease-of-use of the program that you write, so that for example the user provides

* their data directly through the keyboard
* selection of the type of area they are interested in (see the four different types above)
* for the standard normal distribution table the user should specify the start and end points for *z* (e.g. from 0.00 to 1.49 as shown above)
* a choice from two numerical methods

In response to the user input, your application needs to

* **calculate and display** the area
* **display a** **graphical** representation of the area
* **calculate and display** the normal distribution table

**Deliverables:**

**Deliverable 1 – report in PDF format**

Your report should contain:

Design and user documentation of how to use the program 15%

Discussion for the choice of test data you provide and a table detailing the

tests performed 10%

Critical evaluation of the **choice** and **performance** of the **data structures**

and **algorithms** used 15%

Reflection of what was learnt and how this might be used in future work 10%

**Deliverable 2 – well commented JAVA code**

The correctly working source code should be uploaded as a zip file 35%

Viva demonstration of your working solution (scheduled during teaching term) 15%

**If you are unsure about any of these instructions, then please contact the Course Leader with your query either by using email or make an appointment to see them as early as possible.**

[**https://www.sophia.org/tutorials/standard-normal-distribution**](https://www.sophia.org/tutorials/standard-normal-distribution)