Object Oriented Programming I
Assignment 2, Instructions

Analyzing Data Distributions

1 Data

- In the files `data2.txt`,...,`data7.txt` you find data sets containing floating point numbers. The files are contained in `Data.zip` available for download as part of this assignment.
- All numbers in the files are in the range $[0, 1000]$.

2 Creating Histograms

A *histogram* is a certain representation of numerical data. Let real numbers $s_0, ..., s_n$ be given. Then we can choose some intervals $I_i = [a_i, b_i)$, $i = 1, ..., k$, and find out how many of the values $s_0, ..., s_n$ are in each interval I_i. We call this number N_i, i.e.,

$$N_i = |\{j \in \{0, ..., n\} : a_i \leq s_j < b_i\}|$$

for $i = 1, ..., k$. Here, $|M|$ stands for the number of elements of a set M. The intervals I_i are called *bins* of the histogram. Histograms usually are represented graphically, but here we are only interested in the numbers N_i.

Example 1

- Data:

j	0	1	2	3	4	5	6	7	8	9	10
s_j	2	5	2	4	5	9	2	5	9	1	6

- Intervals: $I_i = [2i, 2i + 2)$, $i = 0, ..., 4$
- Numbers of values in the intervals:

i	0	1	2	3	4
N_i	1	3	4	1	2

Problem 1 Write a C++ program that does the following for each of the data sets `data2.txt`, ... ,`data7.txt`:

- Read the values from the file into a `vector<double>`.
- Compute the numbers $N_0, ..., N_{999}$, where N_i is the number of values x in the file with $i \leq x < i + 1$.

For checking the correctness of your program, some values of N_i for the file `data2.txt` are given below.

i	300	301	302	303	304	305	306	307	308	309
N_i	81	73	82	78	70	80	81	74	70	69

3 Underlying distributions

- For each data set, we want to find out an *underlying distribution* (at least approximately).
- An underlying distribution is a function $f : [0, 1000] \to \mathbb{R}$ which approximates the values N_i "as well as possible". This means that, on average, $f(i)$ should be as close to N_i as possible.
- The distributions we use are NOT "normalized", i.e. $\int_0^{1000} f(x)dx$ can be different from 1.

List of possible distributions:

Distribution	**Specification**
Weibull	$f(x) = ax^{b-1}e^{-(x/c)^b}$ for some constants a, b, c
Rayleigh	$f(x) = axe^{-x^2/b}$ for some constants a, b
Cauchy	$f(x) = \frac{a}{1+(x-b)^2}$ for some constants a, b
Logistic	$f(x) = ae^{-x/b}/(1 + e^{-x/b})^2$ for some constants a, b
Gaussian	$f(x) = ae^{-(x-b)^2/c}$ for some constants a, b, c
Exponential	$f(x) = ae^{-x/b}$ for some constants a, b
Uniform	$f(x) = a$ for some constant a

Problem 2 Continuing your program from Problem 1, write a C++ program that does the following for each of the data sets `data2.txt`, ... ,`data7.txt`:

- For some suitable function f from the list on the last page, find values for a, b, c (where applicable) such that
$$S := \sum_{i=0}^{999} (f(i) - N_i)^2 < 10^5. \quad (1)$$
Note: If (1) holds, then $f(i)$ is a quite good approximation to N_i on average.

- For example, for `data2.txt`, the exponential distribution $f(x) = ae^{-x/b}$ with $a = 750$, $b = 134$ works – the sum S in (1) is roughly 90290 for this choice of f.

- For each data set, you may pick one suitable distribution and restrict a, b, c to a suitable range. For this, you need to study the properties of the data. Describe in detail the reasons for your restrictions as comments in the .cpp file. For example, for `data2.txt` one could argue that $100 \leq a \leq 1000$ and $10 \leq b \leq 1000$ if the exponential distribution is chosen for the approximation, and the C++ program then only needs to search for suitable a, b in this range.

- The program should print the results to the screen as follows:
```
data2.txt:  Exponential distribution with a=750, b=134, S=90920
data3.txt:  ...
...
data7.txt:  ...
```

- The execution of the program on a PC or laptop should not take longer than 3 minutes.

Hints:

- Any method to find reasonable estimates for a, b, c is allowed, even Excel plots etc.

- For each data set, it is enough to search for a, b, c for only *one* distribution and give a justification why others where excluded (as a C++ comment).

- It is possible to get quite good estimates for a, b, c (where applicable) for most cases by some simple considerations. For example, for the Gaussian distribution, the parameter a should roughly be the maximum of the $N[i]$ and b should be roughly the mean of the values in the data file.

4 Submission

Put your complete solution for Assignment 2 (including Problems 1 and 2) into one file `assignment2.cpp` and submit this file in NTULearn. You must make sure that your program compiles and runs correctly under Dev-C++ or XCode.