1. MTH395 - Computational Project

In this project you are given four samples from unknown probability distributions and asked to find a combination of probability distribution and parameter values that fits the sample well enough to pass a common ”goodness of fit” test. Your data is stored in your eLearn shared file area under the name project.csv.

Possible distributions are:

Continuous	Discrete
gamma	binomial
beta	geometric
exponential	negative binomial
chi-square	poisson
normal	
lognormal	
logistic	

Your task is to identify the distributions your samples came from (including estimates of the parameter values).

You must justify your answer using a statistical test called variously the chi-square goodness of fit or Pearson goodness of fit test.

1.1. The Pearson Chi-square Goodness of Fit Test.

The test is based on the fact that if we divide the range of the random variable Y into n ranges using a sequence of $n+1$ cutoff points $x_0 < x_1, \ldots, x_{n-1} < x_n$, and compute the probability P_i that Y falls into each of these ranges, $1 \leq i \leq n$,

$$P_i = P(x_{i-1} \leq Y < x_i) \quad i = 1, 2, \ldots, n$$

then the **expected** number of observations k_i from a sample of size N that fall into range i is:

$$E_i = E(k_i) = NP_i, \quad i = 1, 2, \ldots, n$$

Given a random sample $S = \{y_1, y_2, \ldots, y_N\}$ of size N, we compute the vector of **observed** frequencies for our n ranges as:

$O_i = \text{count of values in sample with } x_{i-1} \leq y < x_i \quad = \quad n\left(\{y_j \in S \mid x_{i-1} \leq y_j < x_i\}\right)$

The test statistic is computed as:

$$X = \sum_{i=1}^{n} \frac{O_i - E_i)^2}{E_i}$$

When the data actually consists of a sample of size N from the distribution used to calculate the E_i, the random variabe X will have a chi-square distribution.

Large values of X (large enough that the probability that $X \geq x$ is less than or equal to 0.05) lead us to conlude that it is unlikely that the sample came from a population with the distribution we used to compute the E_i values.

We will use the R `chisq.test()` function which automatically supplies the correct degrees of freedom parameter for the chi-square distribution for X, as well as the probability $X \geq x$ that a chi-square random variable with that many degrees of freedom takes a value greater than or equal to the one computed from the sample.

All we have to supply are the vectors $O = [O_1, O_2, \ldots, O_n]$ and $P = [P_1, P_2, \ldots, P_n]$. The `chisq.test()` function will automatically compute the E_i values, X, the number of degrees of freedom ν, and the probability that a chi-square random variable with ν degrees of freedom takes a value greater than or equal t X.

If that probability is 0.05 or less you should conclude that the sample **does not** fit distribution. Otherwise, you may conclude that the distriution used to compute the P_i values is a reasonable model for the population from which the random sample S was drawn.

1.2. **Parameter Estimation by the Method of Moments.** All of the distributions on our list have one or more parameters. Part of the process of determining whether or not the data fits a certain distribution is estimating the values of the parameters. Even if we are assuming that the data fits, say, a chi-square distribution, we still need to have a reasonably good estimate of the single parameter (the "degrees of freedom") in order to calculate the probabilities.

While we will not examine estimation rigorously until MTH396, for now we can use a simple procedure called the "method of moments". To apply this method, using the moment-generating function or direct algebraic computation or the table in the book, find an expression for $E(X)$ in terms of the parameters of the distribution. Next, replace $E(X)$ with $\overline{x}$, the sample mean, and solve for the parameter(s).

If the distribution has more than one parameter, find an expression for $V(X)$ in terms of the parameters, and replace $V(X)$ in this expression with s^2, the sample variance. In the two parameter situation, you will end up with a system of two equations in two unknowns, one obtained by substituting $\overline{x}$ for $E(X)$, the other by substituting s^2 for $V(X)$.

For example, in the case of the chi-square distribution, you only need the sample mean $\overline{x}$ because the distribution has only one parameter ν. Since

$$E(X) = \nu$$

in this case of a chi-square distribution, the method of moments estimate of ν is simply $\overline{x}$. While you don't need the variance to compute this estimate, the fact that for the chi-square distribution,

$$V(X) = 2\nu$$

provides a quick "sanity check" by comparing s^2 to $2\overline{x}$. A wildly different value suggests maybe chi-square is not the right choice for this data.

The method of moments is easy to apply but there are problems with it. For example, often the parameter values must be nonnegative, but there is no guarantee that the solution to your system of equations using the sample mean and variance will be nonnegative. So the method of moments sometimes yields unusable values. As it turns out there are many approaches to the parameter estimation problem, some of which we will examine in MTH396.

1.3. **Computations with R.** There is no getting away from the fact that practical applications of probability and statistics involve computation - usually LOTS of computation. We are very fortunate to live in an age when computing power is readily available and inexpensive.

There are many software packages available for the most common types of statistical analyses, so many in fact that it is virtually impossible to be well versed in all or most of them.

In my experience, you do not have to be an expert in a given package to accomplish a specific task, you can usually just pick up what you need to know from manuals or tutorials and still manage to get the job done. In fact, because it is difficult to predict what software you will be expected to use on a given project, the ability to quickly pick up the basics of a package you may not have worked with before will serve you well. Keep in mind that all statistical packages basically do the same things.

While are many options for performing the necessary computations for this assignment, R is a good choice because it can do everything we need to do. R may not be as easy to use as some other programs, but you can always count on having R available.

Some of the tasks we will likely require:

- 1) Read the sample data from a `.csv` file and create an R data frame
- 2) Make the variable names visible using the `attach()` command
- 3) Find the sample mean (use the `mean()` function)
- 4) Find the sample variance (use the `var()` function)
- 5) Plot a histogram of the data (use the `hist()` function)
- 6) Partition the sample values into n ranges (using programming statements)
- 7) Produce a vector of counts for each subdivision in the partition (more programming statements)
- 8) Produce a vector of probabilities with one entry for each cell in the partition. This will contain the probability that a single observation falls in this cell (still more programming)
- 9) Run the `chisq.test()` function using the probability and count vectors

1.3.1. *Batch versus Interactive.* The concept of "batch" is foreign to many computer users today, because the user interfaces of the programs they have been exposed to have been exclusively interactive. Generally "batch" processing means creating a text file with a whole sequence of commands, then submitting the entire sequence for processing. I strongly believe that, for any nontrivial project, batch processing or something equivalent is absolutely necessary. My reasons for this are:

- 1) Documentation - there is an indisputably correct record of the command sequence. Six months from now very few people will remember a sequence of commands they entered interactively or an exact point-and-click sequence, but with a batch file there is an exact record. Projects have an annoying habit of resurfacing months later when, say, a referee for a paper asks a question about the data analysis. You have to be able to provide answers about exactly what was done, and relying on memory is a very bad idea, especially if you are involved in a lot of projects. Chances are you won't remember specific details.
- 2) Efficiency - Invariably a data analysis project will undergo several iterations before it is completed. Errors in the input data may be discovered, or errors in the data workup, additional data may become available that the investigator wants to include in the study, or it may be discoverd that certain data should be excluded for some reason. With an interactive user interface, you are always back to square one, but with a batch file you just change the data and submit it and the entire analysis is redone, exactly as before, usually in a matter of seconds.
- 3) Flexibility - If you have a couple of variations of the analysis (such as the four data vectors in this project), it is easy to copy and paste statements from one batch file to another, rather than trying to remember a different command sequence or modify one on the fly.
- 4) Comments - you can imbed comments with a detailed explanation of what the program is doing. This way when you come back in 6 months an look at the program, you can easily recall what it does.

To run R in batch mode using a file called "rfilename.r", enter the command:

```
R --vanilla<rfilename.r> rfilename.out
```

The following batch file examples are posted on the course website:

`project.r`	Sample data generator program
`setup.r`	Reading the raw data from the test.csv file
`solution_y1.r`	Analysis for variable y1 in the data generated by project.r
`solution_y2.r`	Analysis for variable y2 in the data generated by project.r
`solution_y3.r`	Analysis for variable y3 in the data generated by project.r
`solution_y4.r`	Analysis for variable y4 in the data generated by project.r
`project.mws`	Maple worksheet for solving method of moments equations for beta distribution

The data will be uploaded to your shared file area on eLearn and will be named ”test.csv”.