Technical University of Denmark

`bestfit`: Fitting basic distributions in R

Authors:
Andreas K. Salk
Frederik Toftegaard

June 6, 2019

Contents

1 Introduction

This compendium is written with the aim to help the reader fit distributions in R. The compendium provides an R program that tests the eight most known distributions and compares them to fit the best distribution. The program only tests the eight distributions, but can easily be modified to fit mixed distributions as well. How to use the program for the basic distributions will be explained. An walk-through of the statistics and how to analyze of the results will be be provided.

The program is written with the intend of making a input analyzer, that can easily and quickly fit basic distributions in R. An understanding of basic statistics is needed to have a critical mind with regards of the results. To run the program you need R 3.4.4 or later, and an R IDE (We recommend R-Studio). The user does not need any excessive knowledge to R, everything will be explained.

2 How to use the program

If you have programmed in R before, this section can be skipped entirely. This section will explain how to use the program without knowing how the code is built.

2.1 Initialization

At first you need to download and un-zip the folder called InputAnalyzer uploaded to DTU Inside[1].

To work with the functions you need to set the working directory. The working directory is the folder in which you have placed the unpacked version of the InputAnalyzer. This can be done with the following function (remember to specify your own path, e.g. "C:/Users/Frederik/Documents" if your files are located in that folder):

```
setwd("path_to_source_file")
```

The quick command to run a line of code en R is **CTRL+Enter**. However as the code is you can run line 1-27 by marking it and pressing **CTRL+Enter** and it should work. This is because the code specified to run from the source file location. Running this will load the created functions and check if the requirements for the program is met, if not R will install the required packages (This can take some time, and if you just installed R it can take several minutes). If you are interested in how the code is build, you can open the **bestfit** R-script located in the InputAnalyzer.

To fit a distribution over data, the data must be loaded into R, this is done with the command **read.xlsx** (if the data is a xlsx file, if not other "read"-functions can be used). The input is the data files path inside quotation marks, you can also specify whether the table has headers and the sheet index.

```
dat <- read.xlsx("Input-R.xlsx",sheet = 1, colNames = FALSE)
```

If your data file is located in the working directory you just have to change the name, specify the sheet index and define if the data has column names.

[1] Also found here: https://github.com/Andreasksalk/bestfit

2.2 Basic distributions

The function bestfit computes all the necessary elements to fit a distribution and takes the input of a single numeric vector (your data) and the number of bins you want to use. Because of that you have to extract the column in which the data lies. This can be done easily by typing:

```
dat1 = dat[,2]
```

This creates the list dat1 that can has the second column of the data dat in it, meaning it will take the second column of your Excel Sheet as the input. If the data you want to fit the distribution on is in another column of your Excel Sheet, simply change the "2" to the number of number of the column you want to fit the distribution on (note: do not use "A" or "B" as column names). From there on you just run the function **bestfit**.

```
bestfit(dat1, 30)
```

Here we have specified that we want to fit an distribution over the data **dat1** with 30 bins.

The function it self will give you three elements, but how to analyze these elements will be described in section 3. The first element is a sorted list of the mean square error criterion of the eight used distributions, the second element is the 3 best distributions estimated cumulative distribution functions (ECDF), the third element is the three best histograms of the data with the distribution function plotted over, the parameters for these distributions, a Chi-Squared test, and a KS-test of the estimated distributions.

3 Statistics behind the program

The program uses a lot of built-in functions in R to calculate the statistics. This section will go through the distributions used in the program, and how we fit the the input data to distributions. It will also go through how we sort the distributions, and the statistical tests. Lastly the section will go through the output of the input analyzer. The section will not explain in detail how this is done in R, but the theory behind the statistics and how to analyze the output.

3.1 Distributions

This program only uses 8 different probability distributions. There are a lot more, but these are the most common and should be able to describe most data sets without a high error rate.

To fit the probability distributions to a data set the program uses a function called "fitdistr". This function tries to fit a data set x to the wanted distribution (however it is limited to some). It then returns the value of the parameters used to replicate this distribution else where (e.g. AnyLogic). An example of how it is used can be seen below where it returns the parameters used to replicate the data set "x" as a normal distribution.

```
ft.norm <- fitdistr(x,"normal")
```

The normal distribution is a very common distribution and uses 2 parameters: a mean and a standard deviation (typically called μ and σ). The program does this for 8 very common distribution. It will try to fit the data set to a normal, exponential, poisson, lognormal, weibull, gamma, triangular and a a uniform distribution, and try to find the best possible fit.

The way fitdistr works is that it uses Maximum likelihood estimation (MLE) to estimate the parameters for a given distribution. It only works on a univariate distributions (meaning it will only take one vector/column of data as input). It will try to maximize the likelihood function $L(\theta; x)$, where x is the data set and θ is the possible parameters for the given distribution. So the method denotes a maximum likelihood estimate:

$$\theta^* \in \{\arg\max_{\theta \in \Theta} L(\theta; x)\} \tag{1}$$

So θ^* will contain the parameters that with the maximum likelihood will describe the data set "x", meaning the parameters of the given probability distribution that bests describe the data set. The above will only happen if there exists a maximum.

3.2 Sorting criteria

The main sorting criteria in the input analyzer is the "Mean square error" (MSE). The calculations can be seen sorted in the table on the left, with the best solution at the top and the worst at the bottom. The MSE takes the sum of the relative frequency h_j of the j'th interval, minus the the probability of the drawn distribution of the j'th interval $\widehat{p}_j$ squared. The mathematical expression is seen below:

Parameters and sets of the mean squared error	
j	Intervals of the frequency count
i	Indexing of the data
n	Number of intervals (bins)
j	Intervals of the frequency count
x_i	The data
b_j	Breakpoints of the class intervals
c_j	Frequency count
$\widehat{p}_j$	probability of the j'th interval

$$h_j = \frac{c_j}{n}, \qquad \widehat{p}_j = \int_{b_{j-1}}^{b_j} \widehat{f}(x)dx$$

$$S.E. = \sum_{j=1}^{k} (h_j - \widehat{p}_j)^2$$

To get an better understanding what this means, lets look at some of the the output of best fit:

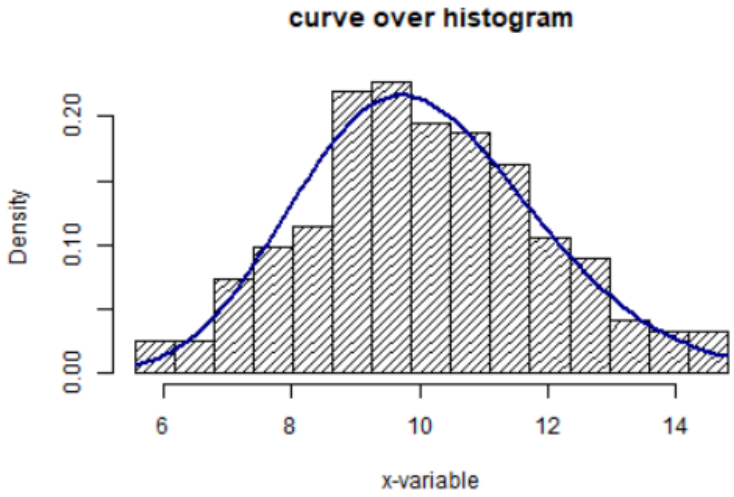

Figure 1: Histogram and PDF

What the MSE does is to for each bar it takes the error between the frequency count (the bar) and the probability density function (PDF) and takes that value squared and adds them together. With this you get an value that describes how much of the PDF is from the actual data.

3.3 Statistical tests

3.3.1 Chi-squared test

The Chi-squared test (Pearsons Chi-squared statistic) is a goodness of fit test. What the Chi-Squared tests is the null hypothesis that the data and probability function is independent (meaning that PDF does not describe the real data) in a 95% confidence interval. The data is independent of each other if the Chi value is below 0.05, and therefore you can not use the tested distribution to explain the data. However If the value is above the chi-value then we can not conclude that the data is dependent on each other. The Chi value goes from 0 to ∞, however a larger value than another does not necessarily mean larger dependence. Therefore you can not use the Chi value alone to determine how well the data fit.

3.3.2 Kolmogorov-Smirnov test

The Kolmogorov-Smirnov test (KS-test) is mostly used for the same thing as the Chi-Squared, the difference being the statistical input (you do not have to look more into this) and the null hypothesis. The null hypothesis of the KS-test is that the PDF is drawn from the reference distribution (opposite of the Chi-squared) with a 95% confidence. This meaning that if the Test statistic is below 0.05 the data is considered dependent of each other.

3.4 Visualization of results

The result is the above mentioned sorting criteria and the statistical tests. Besides this the analyzer also returns visualizations of the results, this is shown by a histogram, a fitted probabilistic distribution and the ECDF.

The aim of the plotted histogram with the probability density function is to visualize how well the fitted distribution actually fits the data set, represented by the histogram. When you run the function you choose the amount of bins, and it is important to choose the right amount of bins. If you choose too few, the fitted PDF and the histogram might be a bit off from each other.

The last plot is of the ECDF. The ECDF computes cumulative distribution of x and default plots it as a step function. This plot is used to see if there are any systematic tendencies in the fitted distribution, and how this could affect the way the cumulative distribution of the fitted probabilistic distribution behaves.

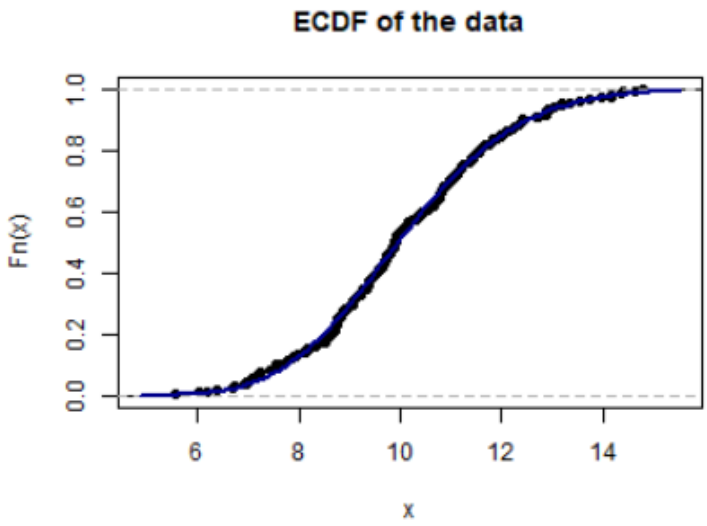

Figure 2: Example of ECDF

Above is an example of an ECDF in R. The blue line is the ECDF of the fitted probabilistic distribution, while the black dots are the actual data points in the data set. Here we see the fitted distribution fits the data set well.

Lastly the input analyzer will give you the parameters for the fitted probability distributions. Some distributions require 3 input parameters, while others require 1. So if the output says "NO VALUE", then it is because the distribution does not require more parameters to be replicated.

4 Discussion

When running the program the user have to be watchful of the results. This meaning that the user have to interpret the results and conclude if they are useful for any example that the program is used on. For example sorts the program the PDF with the lowest Mean squared error as the best solution, this is not necessarily true. The mean squared error is a good indicator but not a conclusion point. Another example is that the Poisson distribution takes discrete input, the program therefore interprets the data discrete for this distribution. This works in most cases but for some it would cause an error an give not true results. Therefore the user all ways have to look at the data with a critical mind.