**EXAMINATION**

***Computational Statistics***

Date: May 24, 2010

**Prepared for:**

Oleg Sysoev

**By:**

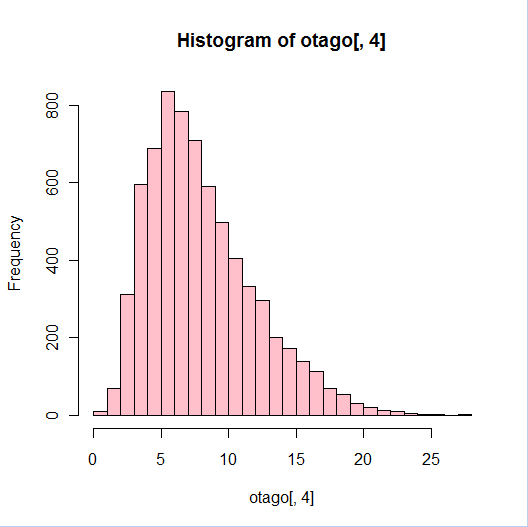
Victoria Fomichov

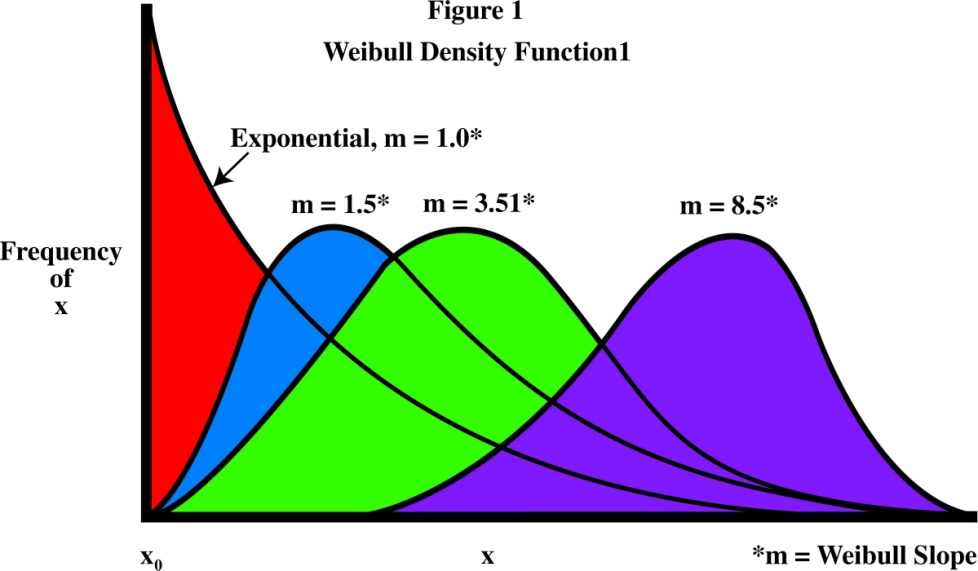
***Assignment 1: Likelihood optimization***

The data set ***wind.xls*** represents wind speed records for twelve Meridian sites distributed around the New Zealand. It has long been known that the wind speed matches well with the Weibull distribution. Your task is to make some inference about the parameter values for the city **Otago**

1. Load necessary information concerning Otago city to R. Plot the histogram of the wind speed and comment whether it looks like Weibull (you may also compare it with the shape of the related Rayleigh distribution)

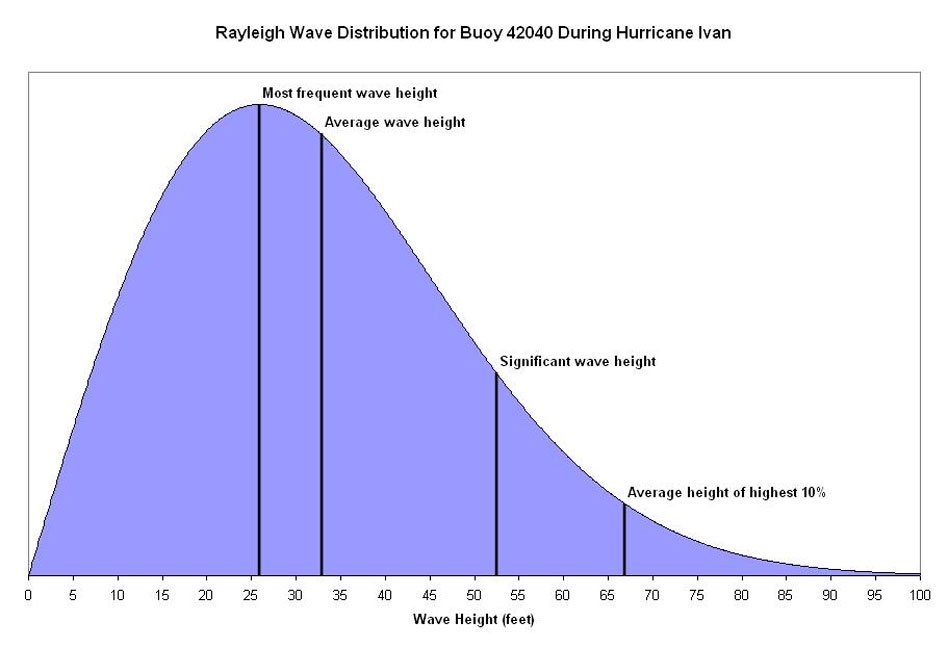




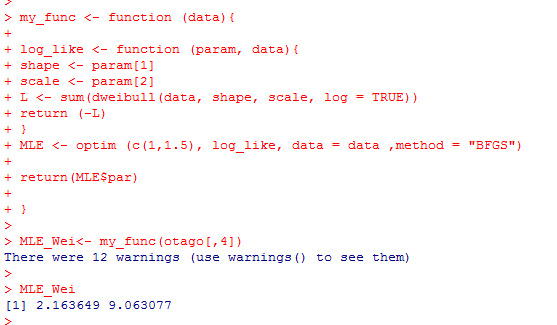


**\*0.5p**

It may look as a Weibull but I can’t be really sure. But if we compare with a Rayleigh distribution then they seem similar. **\*0.5p**



1. Write your own function as a parameter of the data that finds the maximum log-likelihood estimates of the parameters *shape* and *scale* of the Weibull distribution by using an optimization method such as BFGS or CG. Compute the parameter estimates (if you get warnings, ignore them).



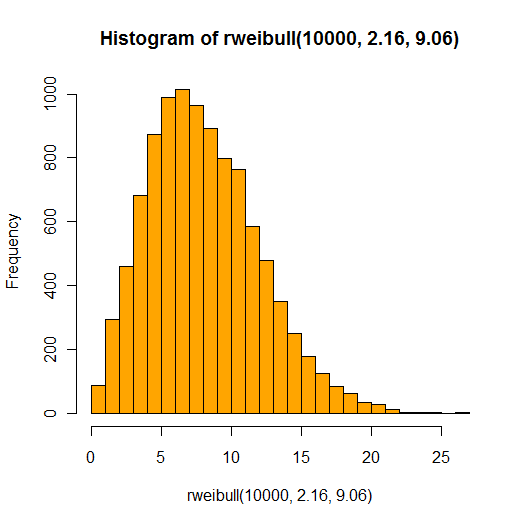
The estimated parameters are:

Shape = 2.16

Scale = 9.06

**\*2p**

If we plot a random sample of the weibull distribution with the above estimated parameters we get the following:



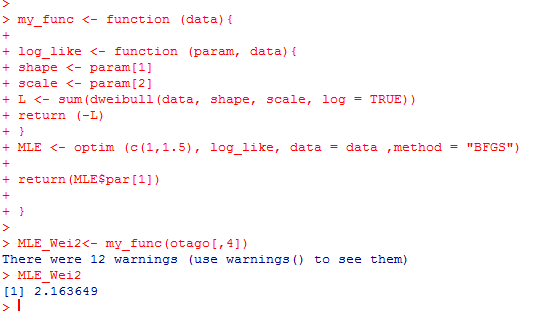
We can see the both histograms are similar.

**What kind of problems may one encounter when these methods are used for the search of the maximum log-likelihood estimates? What other methods can you see as alternatives that do not have these problems?**

That the Weibull distribution only exist for positive numbers and when we use methods as BFGS or CG we can’t put constraint that limit the optimization. **\*1p**

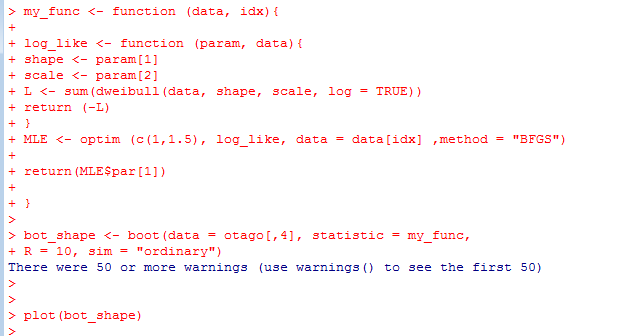
There for can it be good to use something as genetic algorithm. **\*0.5p**

1. Modify your function in such way that it returns only *shape* parameter

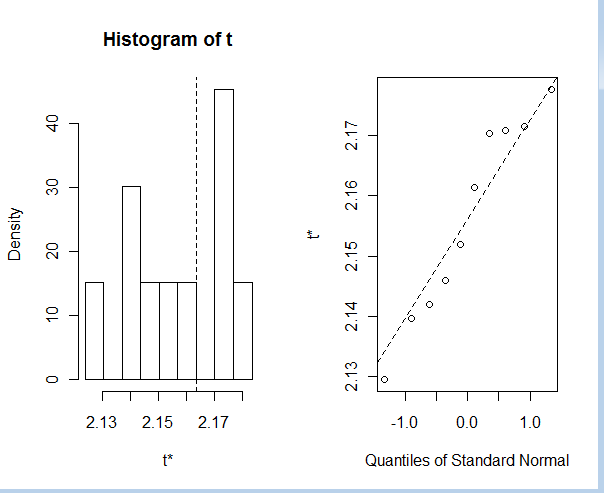


The function returns only the shape parameter.

Consider this function as an estimator of *shape* and generate 100 bootstrap estimates of this parameter. Produce an appropriate plot and make comments.

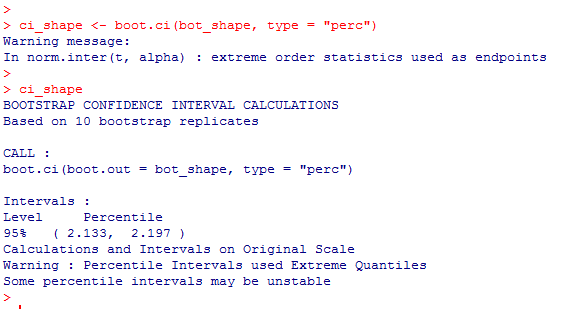
 **\*1.5p you did 10 bootstrap instances instead of 100, but the points are not reduced.**

To make a bootstrapping we introduce a second parameter to the function.



**\*0.5p**

Compute 95% confidence bounds (**using percentile method only!**) for the parameter *shape* and answer whether the data can in principle come from Rayleigh distribution.

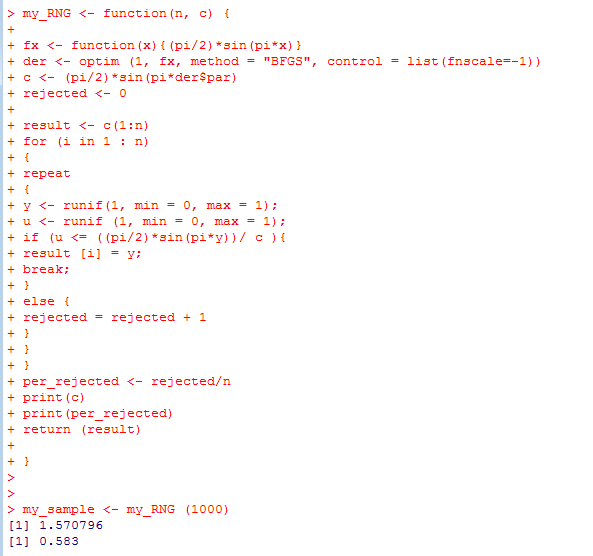


**\*0.5p**

***Assignment 2: Computing an integral***

Consider the distribution with density

1. Write your own function that can generate sample of size *n* from the distribution above using the acceptance-rejection method with uniform distribution as the majorizing density. The function should also compute what percent ***R***of the totally generated random numbers was rejected in this method. What was the value of the scaling constant *c* you used in deriving the majorizing function and why you have chosen this value?



Count the rejections

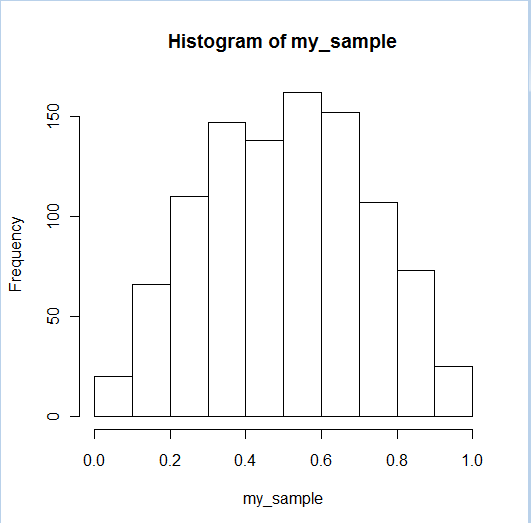
Calculate c value.

**\*1.5p+\*1p**

**“C” could be computed much easier. You did it as “computing 2+2 by using integrals ☺”**

**“Rejection rate” in the assignment was defined as rejected/total, you computed as rejected/accepted, but it is OK, the points are not reduced.**

1. Generate a sample of size 1000 by using your function and plot the histogram. Does the sample look like it should?



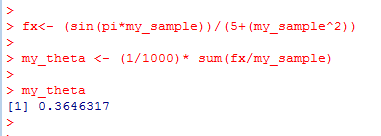
**\*0.5p**

1. Print out the computed rejection rate ***R***. Find out without using computer what would be the rejection rate ***R*** if the sample size would be very large (i.e., the expected value of ***R***).

We can see in point 2 that R is 58 %.

**\*0.5p**

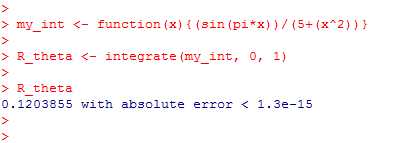
1. Use the obtained sample to compute the value of the integral by applying the importance sampling. Why using the introduced above as importance function is better than using the density of the uniform distribution?



From formula 11.7 in the curse book.

**\*0p WRONG. You should have decomposed the fuctiopn**

1. Compute the same integral by using the function *integrate* in R and comment on the result.



**\*1p**