***Assignment 1: Likelihood optimization***

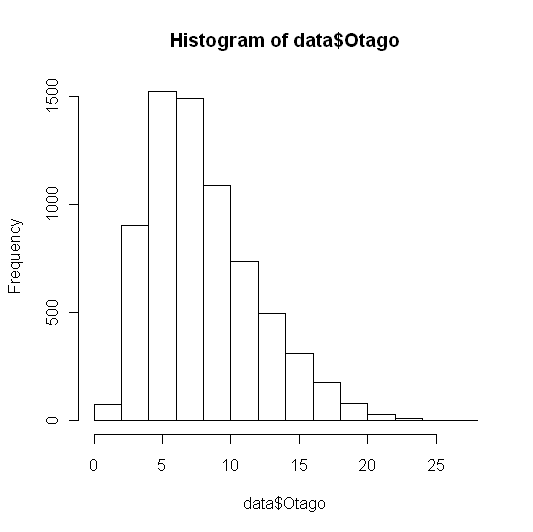
The data set ***wind.xls*** represents **wind speed** records for twelve Meridian sites distributed around the New Zealand. It has long been known that the wind speed matches well with the Weibull distribution. Your task is to make **some inference** about the parameter values for the city **Otago**

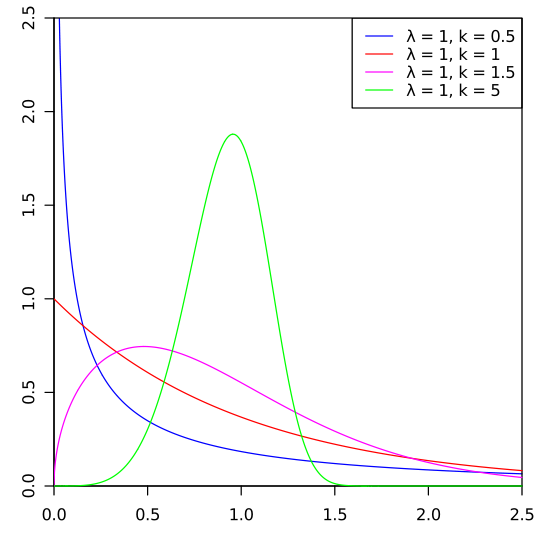
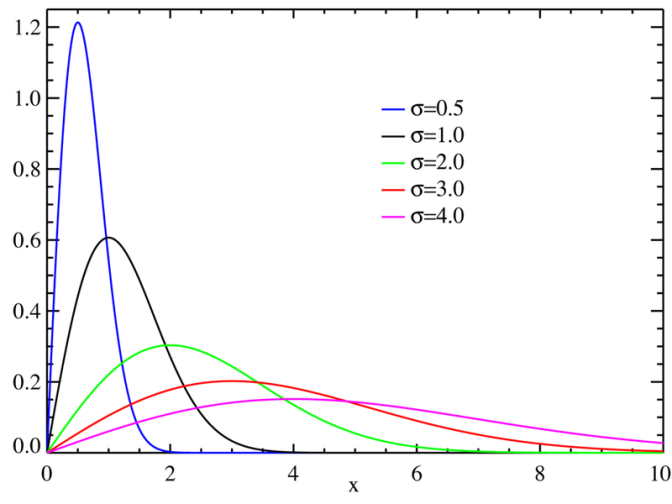
1. Load necessary information concerning **Otago city to R.** Plot the histogram of the **wind speed** and **comment** whether it looks like Weibull (you may also compare it with the shape of the related **Rayleigh distribution**)

**Solution:**

data<-read.csv("otago-wind.csv",header=TRUE,",");

hist(data$Otago) **\*0.5p**

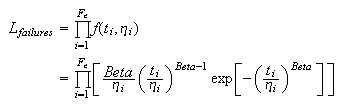


[](http://upload.wikimedia.org/wikipedia/commons/5/58/Weibull_PDF.svg) [](http://upload.wikimedia.org/wikipedia/commons/6/6a/Rayleigh_distributionPDF.png)

The wind speed will have a Rayleigh distribution if the components of the two-dimensional wind velocity vector are uncorrelated and normally distributed with equal variance. So it seems like Rayleigh. **\*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). **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?**

**Solution:**



############# log likelihodd code#############

data<-read.csv("otago-wind.csv",header=TRUE,",");

hist(data$Otago)

initParams <- c(1, 1.5);

loglikelihood<-function(k){

shape<-k[1];

scale<-k[2];

temp<- sum(dweibull(data$Otago, shape=shape, scale = scale, log = TRUE));

return(-temp);

}

###### Maximization using optim with CG and BFGS####

> initParams <- c(1, 1.5);

> loglikelihood(initParams)

[1] 39826.94

>

> optim (initParams, loglikelihood, method="CG")

$par

[1] 2.163650 9.063085

$value

[1] 18899.50

$counts

function gradient

540 101

$convergence

[1] 1

$message

NULL

There were 25 warnings (use warnings() to see them)

>

> optim (initParams, loglikelihood, method="BFGS")

$par

[1] 2.163649 9.063077

$value

[**1] 18899.50**

$counts

**function gradient**

**38 9**

$convergence

[1] 0

$message

NULL

**\*2p**

***What kind of problems may one encounter when these methods are used for the search of the maximum log-likelihood estimates?***

In case of conjugate gradient, because this is the large data set having weibel distribution might need hessian estimates to find out the correct shape & scale.

**\*0.5p this is not the main reason**

***What other methods can you see as alternatives that do not have these problems?***

Maybe Newton’s method converge more accurate with hessian matrix rather than approximation of it as BFGS do. **\*0p Newton will have some other problems…**

1. Modify your function in such way that it **returns only *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**. Compute 95% confidence bounds (**using percentile method only!**) for **the parameter *shape* and answer whether the data can in principle come from Rayleigh distribution**.

**Solution:**

loglikelihood<-function(k){

shape<-k[1];

scale<-k[2];

index<-k[3];

temp<- sum(dweibull(data$Otago[index], shape=shape, scale = scale, log = TRUE));

return(-temp);

}

funcshapescale<-function(data,index){

initParams <- c(1, 1.5,index);

res<-optim (initParams , loglikelihood, method="BFGS");

res$par[1]

}

res<-boot(data$Otago, funcshapescale, R=10, sim="ordinary", stype="i");

boot.ci(boot.out = res, conf=0.95, type=c("perc"));

**\*0.5p Code needs to be further optimized…**

***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 p**ercent *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?
2. Generate a sample of size 1000 by using your function and plot the histogram. Does the sample look like it should?
3. 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***).
4. Use the obtained sample to compute the value of the integral **to derive 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?
5. Compute the same integral by using the function *integrate* in R and comment on the result.

## Submission procedure

Submit your solutions via It’s learning at latest by 24 May, 13:00