

# Fitting Wave Height Data to a Probability Distribution

## ▼ Introduction

The University of Maine records real-time accelerometer data from buoys deployed in the Gulf of Maine and the Caribbean (<http://gyre.umeoce.maine.edu/buoyhome.php>). The data can be downloaded from their website, and includes the significant wave height recorded at regular intervals for the last few months.

This application does the following:

- Downloads accelerometer data for Buoy PR206 (located just off the coast of Puerto Rico at a latitude of  $18^{\circ} 28.46'$  N and a longitude of  $66^{\circ} 5.94'$  W)
- Fits the significant wave height to a Weibull distribution by using two methods: maximum likelihood estimation and moment matching
- Plots the fitted distributions on top of a histogram of the experimental data

> restart :

with(plots) : with(Statistics) : with(Optimization) : *with(ColorTools)* :

## ▼ Download and Plot Significant Wave Height Data in a Histogram

> url :=

"http://gyre.umeoce.maine.edu/data/gomoos/buoy/php/view\_csv\_file.php?ncfile=  
/data/gomoos/buoy/archive/PR206/realtime/PR206.waves.triaxys.realtime.nc" :

data := ImportMatrix(url)

```

data := [
    "Time[GMT]"      "significant_wave_height[m](0m)"  "dominant_wave_period[s](0m)"  "max
"2015-04-25 15:00"      1.15                        14.3
"2015-04-25 16:00"      0.91                        14.3
"2015-04-25 17:00"      1.17                        14.3
"2015-04-25 18:00"      1.04                        13.3
"2015-04-25 19:00"      1.25                        13.3
"2015-04-25 20:00"      1.05                        13.3
"2015-04-25 21:00"      1.10                        13.3
"2015-04-25 22:00"      0.93                        12.5
"2015-04-25 23:00"      0.93                        13.3
    :
    :
    :

```

```
> sigWaveHeight := RemoveNonNumeric( data[ .., 2] ) :
```

```
> n := numelems( sigWaveHeight )
```

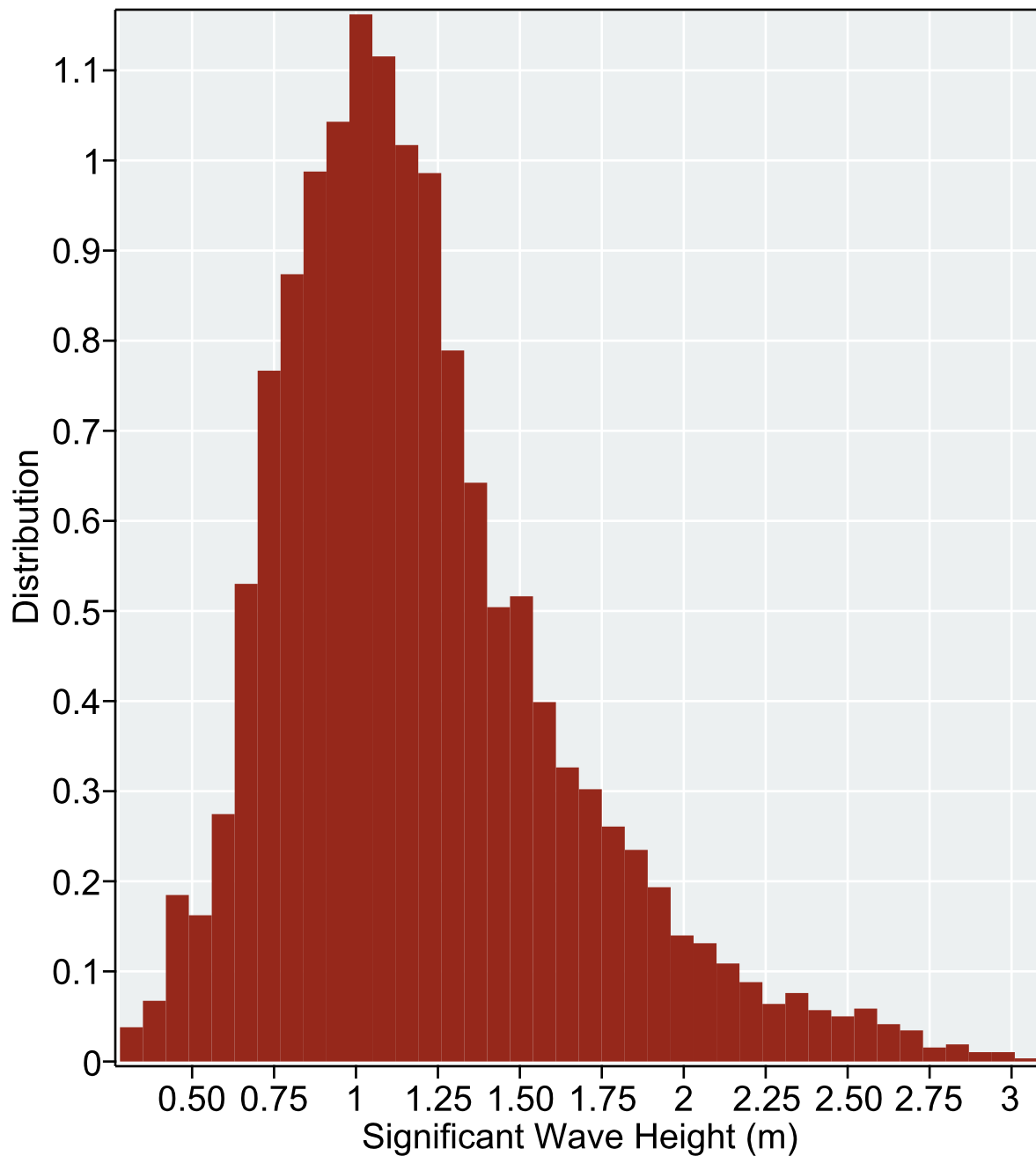
$n := 8273$

(2.2)

```
> numBins := 40 :
```

```
> p1 := Histogram( sigWaveHeight, bincount = numBins, color = COLOR( RGB,  $\frac{150}{255}$ ,  $\frac{40}{255}$ ,
 $\frac{27}{255}$  ), thickness = 0, style = patchnogrid, transparency = 0.5, background = ColorTools:-
Color( "RGB", [  $\frac{236}{255}$ ,  $\frac{240}{255}$ ,  $\frac{241}{255}$  ] ), axis = [ gridlines = [ 10, color = RGB( 1, 1, 1 ) ] ], axesfont
= [ Arial ], labels = [ "Significant Wave Height (m)", "Distribution" ], labeldirections
= [ horizontal, vertical ], labelfont = [ Arial ], size = [ 800, 500 ] ) :
```

```
> display( p1 )
```



## ▼ Maximum Likelihood Estimation

>  $P := \text{unapply}(\text{ProbabilityDensityFunction}(\text{Weibull}(\alpha, \beta), t), t, \alpha, \beta)$

$$P := (t, \alpha, \beta) \mapsto \begin{cases} 0 & t < 0 \\ \frac{\beta t^{-1+\beta} e^{-\left(\frac{t}{\alpha}\right)^\beta}}{\alpha^\beta} & \text{otherwise} \end{cases} \quad (3.1)$$

>  $\text{maxLike} := \text{proc}(\alpha, \beta) \text{ local } i :$   
 $\text{return add}(\ln(P(\text{sigWaveHeight}_i, \alpha, \beta)), i = 1..n) :$

**endproc:**

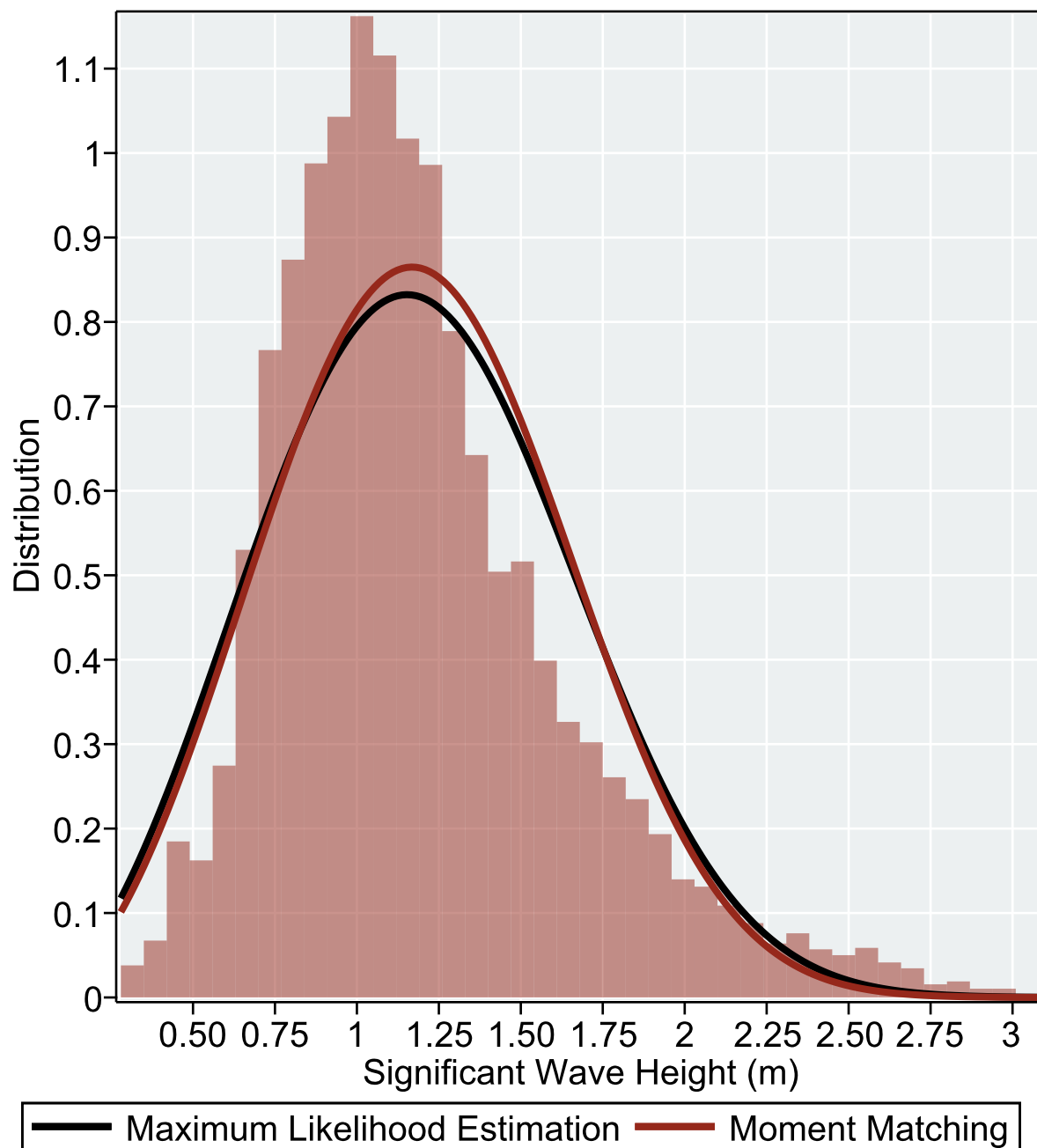
```
> resultsMLE := Maximize( maxLike(  $\alpha$ ,  $\beta$  ),  $\alpha = 0.01 \dots 5$ ,  $\beta = 0.01 \dots 5$  )  
resultsMLE := [ -4839.27465547207703, [  $\alpha = 1.34452570778020$ ,  $\beta = 2.83250005643047$  ] ] (3.2)  
> p2 := plot( eval( P( t, alpha, beta ), resultsMLE[ 2 ] ) , t = min( sigWaveHeight )  
    ..max( sigWaveHeight ), color = black, legend = "Maximum Likelihood Estimation",  
    thickness = 3, legendstyle = [ font = [ Arial ] ] ) :
```

## ▼ Moment Matching

```
> resultsMM := fsolve( [ Moment( sigWaveHeight, 1 ) = Moment( Weibull(  $\alpha$ ,  $\beta$  ), 1 ),  
    Moment( sigWaveHeight, 2 ) = Moment( Weibull(  $\alpha$ ,  $\beta$  ), 2 ) ], {  $\alpha = 1$ ,  $\beta = 1$  } )  
resultsMM := {  $\alpha = 1.342796022$ ,  $\beta = 2.958762705$  } (4.1)  
> p3 := plot( eval( P( t, alpha, beta ), resultsMM ) , t = min( sigWaveHeight )  
    ..max( sigWaveHeight ), color = RGB(  $\frac{150}{255}$ ,  $\frac{40}{255}$ ,  $\frac{27}{255}$  ), thickness = 3, legend  
    = "Moment Matching", legendstyle = [ font = [ Arial ] ] ) :
```

## ▼ Results

```
> display( p1, p2, p3, size = [ 800, 500 ] )
```



>