Playing around with Histograms and Densities

Xerxes

21.11.2023

Intention

It seems, imho, that the transition from histograms to densities corresponds, in statistics, to the transition from empiricism (reality) to theory. Thus, I want to get a feeling for the relation between histograms and the corresponding probability densities.

In the following, this will be exercised using the example of a Weibull distribution. The Weibull distribution is given by the function

$$f : \mathbb{R}^+ \to \mathbb{R}^+ \, ; \; x \mapsto f(x; \lambda, k) := \frac{k}{\lambda} \left(\frac{x}{\lambda} \right)^{k-1} e^{-(x/\lambda)^k} \tag{1}$$

having two parameters: shape k and scale λ. However, the basic principle of fitting a density distribution to a histogram will be the same for all (continuous) distributions, provided that one *knows* the underlying distribution.

The following code snippets are written in R, and the present document has been generated with R Markdown, run from within the RStudio. By the way, this is the output of my very first R Markdown script, which thus not only served me to explore histograms and densities but also to get an idea of the R Markdown language.

Some parameters for the Weibull distribution

```
nRndNumbers <- 1000
shape       <-    1.7   # (= k)
scale       <-    1.0   # (= lambda; apparently just compressing or stretching
                        #            the graph along the x-axis)
```

Generate a set of approximately normal distributed pseudo-random numbers

```
# Set a seed for reproducibility
set.seed(1)   # Comment out to watch different examples!

# Generate a (pseudo-)random Weibull distribution
varRndWeibull <- rweibull(nRndNumbers, shape = shape, scale = scale)

# Create a histogram
hWeibull <- hist(varRndWeibull, breaks = 30, ylim = c(0, 120))
```

Here, the histogram has been created *by means* of a random number generator that generates random numbers according to a Weibull distribution. In real life, of course, the data underlying the histogram has usually evolved from natural or economic processes or the like. In such cases, one may not know, however, which density distrubution has ruled the respective process. Hence, one has to first find out, which is the correct density function, or, if this is not possible, to find a suitable distribution that half-way accurately describes the empiric data.

Calculate the total area of the histogram bars

```
# Calculate the width of the individual bars
barsWidth <- diff(hWeibull$breaks)    # This is a vector containing the diffs of
                                      # consecutive breaks

# Calculate the area of the individual bars
barsArea <- barsWidth * hWeibull$counts

# Calculate the total area of the histogram's bars
areaHistTotal <- sum(barsArea)
```

In fact, the total area of all the histrogram bars and the area under the density curve (to be created below) shall be identical. Thus, we have already calculated the total area of the histogram bars by means of the above code snippet.

Create the corresponding density

```
# Generate some value pairs for the density
xVals <- seq(min(varRndWeibull), max(varRndWeibull), length = 500)
                                             # 500 points should be smooth enough
yVals <- areaHistTotal * dweibull(xVals, shape = shape, scale = scale)
```

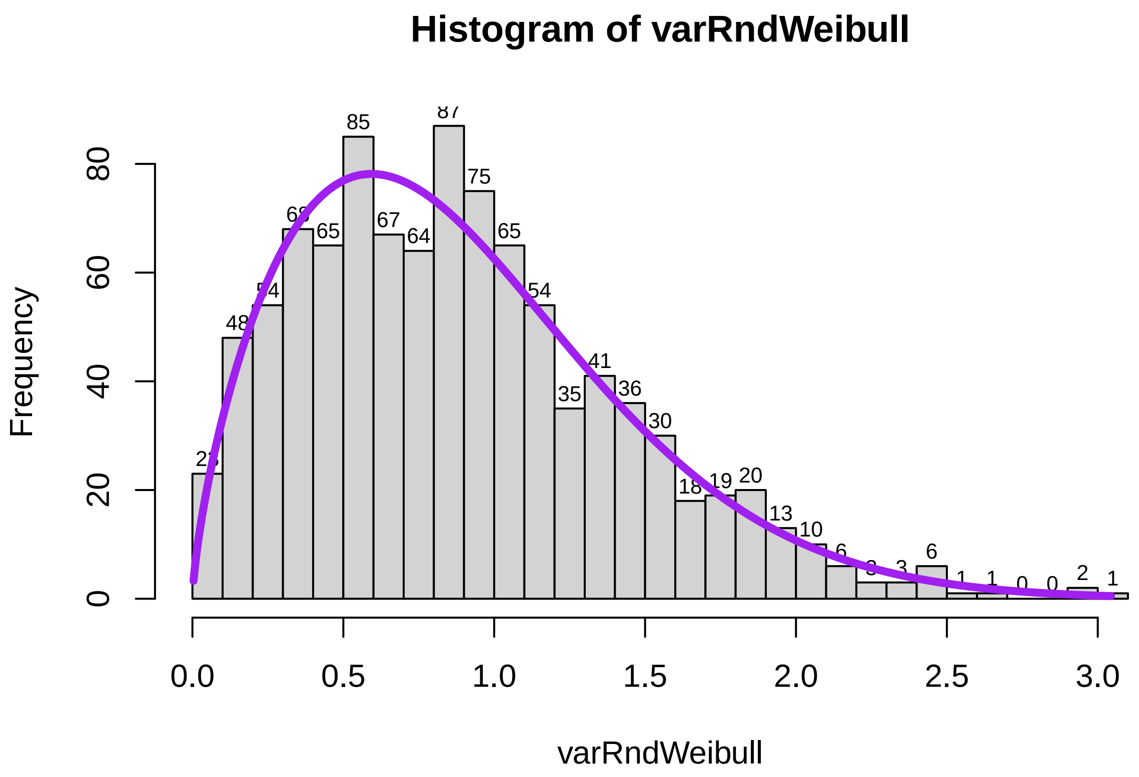
As a probability density function, the Weibull function f as given by formula (1) is normalised to 1. To get the Weibull function corresponding to the *histogram*—which is not normalised—, the normalised Weibull function is thus multiplied by the total area of the histgram bars as calculated before.

Plot the histogram and the density

```
# Plot the plain vanilla histogram created above
plot(hWeibull)

# Place the counts above the histogram bars
text(hWeibull$mids, hWeibull$counts, labels=hWeibull$counts, adj=c(0.5, -0.5), cex = 0.7)

# Draw the density plot
lines(xVals, yVals, lwd = 4, col="purple")
```


Histogram of varRndWeibull
Frequency
varRndWeibull
0
20
40
60
80
0.0
0.5
1.0
1.5
2.0
2.5
3.0
23
48
54
68
65
85
67
64
87
75
65
54
35
41
36
30
18
19
20
13
10
6
3
3
6
1
1
0
0
2
1