CAS Spring Meeting 2012

ILFs with R: Limited Expected Value functions of the actuar package

Daniel Murphy
dmurphy@trinostics.com

Werner & Modlin, *Basic Ratemaking*

Ch. 11 section on Increased Limits Ratemaking

- Produce an increased limits factor from $500k to $2M based on the data in Werner & Modlin Table 11.3

11.3 Size of Loss Distribution

Size of Loss			Reported Claims	Reported Losses
	X <=	$ 100,000	2,324	$ 117,629,223
$ 100,000	< *X* <=	$ 250,000	1,923	307,599,929
250,000	< *X* <=	$ 500,000	680	222,793,514
500,000	< *X* <=	$ 1,000,000	73	43,047,470
	Total		5,000	$ 691,070,136

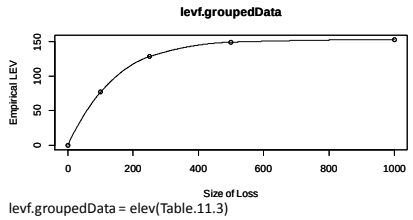
Enter the table into R

Use actuar's "grouped.data" class

```
   SizeOfLoss ReportedClaims ReportedLosses
1   (0,  100]           2324         117629
2 (100,  250]           1923         307600
3 (250,  500]            680         222794
4 (500, 1000]             73          43047
```

```
> breaks = c(0, 100, 250, 500, 1000)
> nclaims = c(2324, 1923, 680, 73)
> amounts = c(117629, 307600, 222794, 43047)
> Table.11.3 = grouped.data(SizeOfLoss = breaks, ReportedClaims = nclaims,
  ReportedLosses = amounts)
> Table.11.3
```

Plot the empirical Limited Expected Value function based on the tabled data

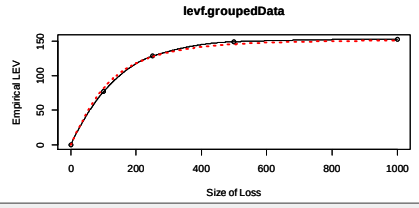



```
> levf.groupedData = elev(Table.11.3)
> plot(levf.groupedData, main="levf.groupedData", xlab="Size of Loss",
  ylab="Empirical LEV")
> x = 0:1000
> y = levf.groupedData(x)
> lines(x, y)
```

Plotting the function by name only just gives the boundary points

Evaluate the function at intermediate points to get a smooth curve

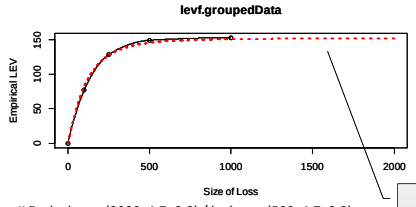
Fit a lognormal to Table 11.3 data Plot the lognormal LEV function



After a multitude of sophisticated methods: mu = 4.7, sigma = 0.8

```
> y = levlnorm(0:1000, 4.7, 0.8)
> lines(0:1000, y, col="red", lty="dotted", lwd=3)
```


Calculate the ILF from the fitted lognormal




```
> ILF = levlnorm(2000, 4.7, 0.8) / levlnorm(500, 4.7, 0.8)
> round(ILF, 3)
[1] 1.042
```

extrapolate to higher limits with lognormal

- The end