

Week 7 - Operational Risk and Extreme Finance

Copyright 2016, William G. Foote. All rights reserved.

Imagine this

- Remember that company you acquired last week? Not only is customer creditworthiness apt to cost you another \$80 million, but your walk-through of distribution, call-center, and production facilities had a raft of negatively impacting issues with health and safety, environmental, intellectual property, . . . , located in places rife with fraud and corruption.
- Recent (5 years ago!) hurricane damage has still not been rectified.
- Your major insurance carrier has also just pulled out of underwriting your fire-related losses.
- Three VPs of manufacturing have just been indicted for padding vendor accounts with kickbacks.
- Employee turnover rates are over 40%.
- . . . The list goes on.

Previously on Financial Analytics

- Stylized facts of financial variables
- Market risk
- Credit risk
- Common theme: thick tails, high VaR and ES

Tonight. . .

- What about the extreme but rare events?
- Poisson - Gamma potential loss
- Generalized Pareto distribution
- Mean excess loss from reliability and vulnerability analysis
- Historical data
- VaR and ES again

What is operational risk?

This is a third major category of risk that includes everything from the possibility of process failure, to fraud, to natural (and our own homemade) disaster, errors, omissions, in other words, having lots of really bad days.

How do we measure this risk?

In a nutshell it is frequency times severity.

- Here we think of the probability of loss as how often a loss could occur, its likelihood.
- Infused into likelihood are opinions about “how often,” “how fast,” “how long” (before detection, response, correction), “how remote” (or “imminent”), among other potential suspects.

- We think of severity as the monetary loss itself.
- Beyond a threshold the loss focuses on “extreme” value methods of dealing with distributions.

Typically experience in most organizations, there is no good time series or cross-sections of data on which to rely.

- For example, the emerging and imminent risks of botnets wreaking havoc in our computer operating systems have really only a little loss data available for analysis.
- The problem with this data is that only certain aspects of the data are disclosed, they are supremely idiosyncratic, that is, applicable to highly specialized environments.
- Where we do have some data, we have established ways of working with severity distributions that make sense.

Try this

For example, suppose management, after much discussion and challenge by subject matter experts, determine that

- ① The median loss due to a vulnerability of a known cyber breach is \$60 million,
- ② With an average (or median) frequency of 25%,
- ③ All in a 12-month cycle.
- ④ Management also thinks that the variation in their estimates is about \$20 million.

We sometimes call this the “volatility of the estimate” as it expresses the degree of managerial uncertainty about the size of the loss.

- ① How would you quantify this risk?
- ② Consider a way to manage this risk. List at least 5 components you might deploy.

Thinking...

Quantify cyber? Yes

- 1 Develop a cyber risk taxonomy, For example look at NIST and ISO for guidance.
- 2 Develop clear definitions of frequency and severity in terms of how often and how much. Calibrate to a metric like operational income.
- 3 Develop range of operational scenarios and score their frequency and severity of impact on operating income.

Manage? Certainly...

- 4 Aggregate and report ranges of median and quantile potential loss
- 5 Develop mitigation scenarios and their value. Subtract this value from potential loss and call it retained risk.
- 6 Given your corporate risk tolerance (you do have one, right?), determine if more mitigation needs to be done.

Behind managerial scenes, we can model the loss with

- A gamma severity function allows skew and “heavy” tails, specified by shape, α , and scale, β , parameters.
- Gamma is especially useful for time-sensitive losses.

We can specify these parameters using the mean, μ , and standard deviation, σ of the random severities, X , as

$$\beta = \mu/\sigma^2,$$

and

$$\alpha = \mu\beta$$

and this is the same as

$$\alpha = \mu^2/\sigma^2$$

The distribution itself is defined as

$$f(x; \alpha, \beta) = \frac{\beta^\alpha x^{\alpha-1} e^{-x\beta}}{\Gamma(\alpha)}$$

Where

$$\Gamma(x) = \int_0^\infty x^{t-1} e^{-x} dx$$

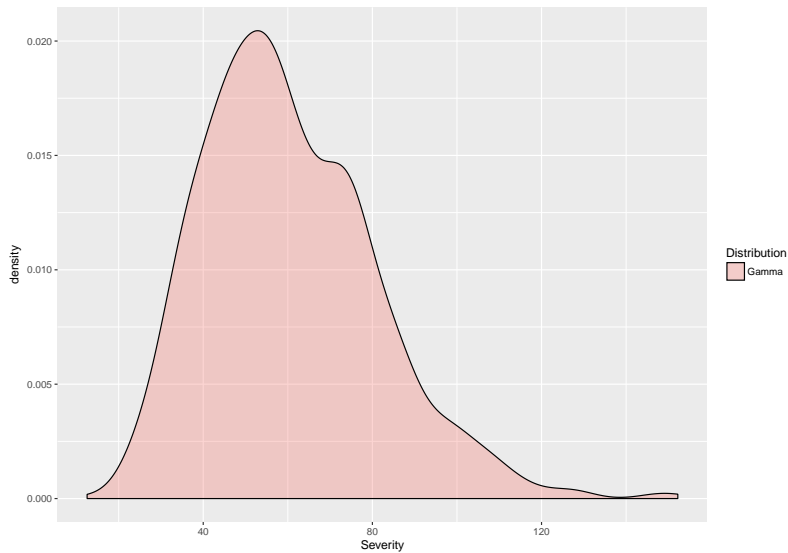
Enough of the math, although very useful for term structure interpolations, and transforming beasts of expressions into something tractable. Let's finally implement into R.

```
set.seed(1004)
n.sim <- 1000
mu <- 60 # the mean
sigma <- 20 # management's view of how certain they think their estimates are
sigma.sq <- sigma^2
beta <- mu/sigma.sq
alpha <- beta * mu
severity <- rgamma(n.sim, alpha, beta)
summary(severity)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    12.54   45.44   57.48   60.30   73.47  152.23
```

The distribution is dispersed from a low of 15 to a high of over 120 million dollars. We will check with management that this low and high (and intermediate levels) are reasonable. Let's graph the distribution.

```
library(ggplot2)
gamma.sev <- data.frame(Severity = severity,
  Distribution = rep("Gamma", each = n.sim))
ggplot(gamma.sev, aes(x = Severity, fill = Distribution)) +
  geom_density(alpha = 0.3)
```



Try this at home...

Add a threshold or two based on value at risk (VaR) and expected shortfall ES.

```
alpha.tolerance <- 0.95
(VaR.sev <- quantile(severity, alpha.tolerance))
(ES.sev <- mean(gamma.sev$Severity[gamma.sev$Severity >
  VaR.sev]))
```

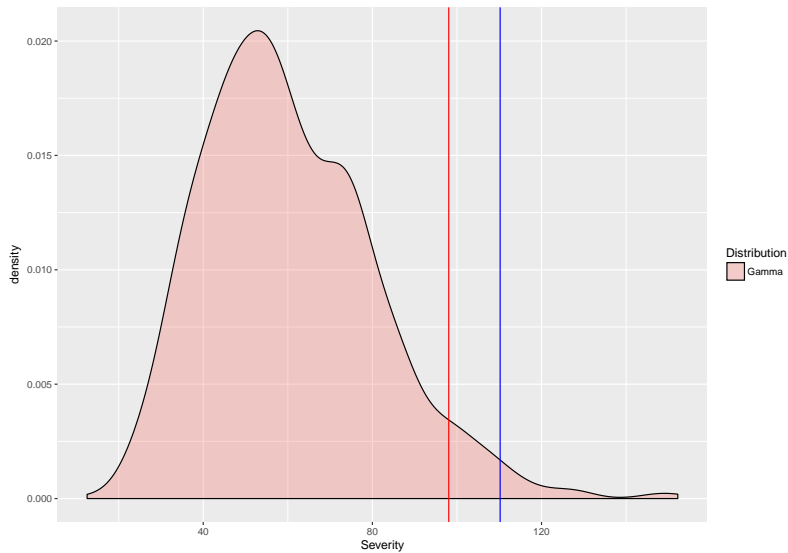
Plot them onto the distribution using the `geom_vline` feature in `ggplot2`. What is the relationship between VaR and ES?

```
ggplot(gamma.sev, aes(x = Severity, fill = Distribution)) +
  geom_density(alpha = 0.3) + geom_vline(xintercept = VaR.sev,
  color = "red") + geom_vline(xintercept = ES.sev,
  color = "blue")
```

Result

```
##          95%  
## 98.05779
```

```
## [1] 110.2134
```



What is the relationship between VaR and ES?

As it should be, the VaR is less than the ES. For this risk of potential cyber breach vulnerability, management might think of setting aside capital.

How much?

- Not so fast! We have to consider the likelihood of such a risk event.

How often?

We usually model the frequency of a loss event with a poisson distribution.

- This is a very basic counting distribution defined by the rate of arrival of an event, λ in a unit of time. This rate is just what we struggled to estimate for credit rating transitions.
- Again the math, for completeness sake: we define the probability of n risk events arriving at a rate of λ :

$$Probability[n; \lambda] = \frac{\lambda^n e^{-\lambda}}{n!}.$$

Management might try to estimate the count of events in a unit of time using hazard rate models (same λ). Here we simulate management's view with $\lambda = 0.25$. Our interpretation of this rating is that management believes that a vulnerability event will happen in the next 12 months once every 3 months (that is four times a year).

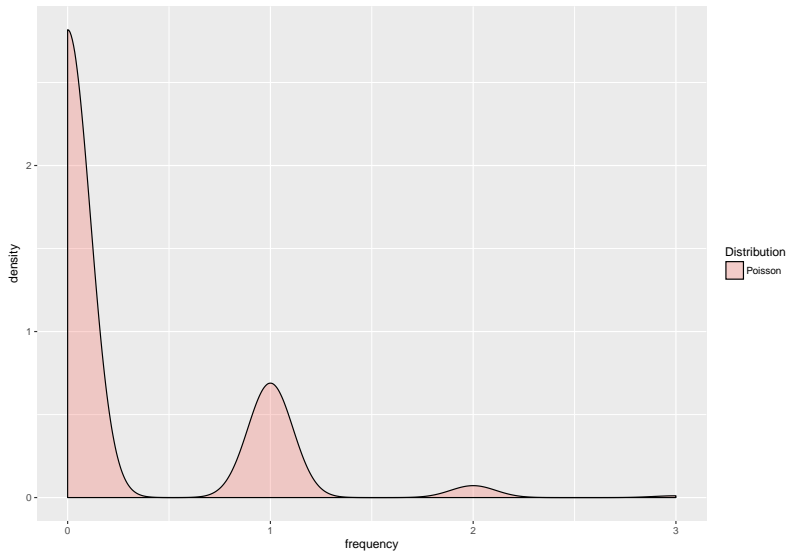
We simulate as follows:

```
lambda <- 0.25  
frequency <- rpois(n.sim, lambda)  
summary(frequency)
```

##	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
##	0.000	0.000	0.000	0.241	0.000	3.000

summary() is not very informative so we also plot this distribution using this code:

```
poisson.freq <- data.frame(Frequency = frequency,  
  Distribution = rep("Poisson", each = n.sim))  
ggplot(poisson.freq, aes(x = frequency,  
  fill = Distribution)) + geom_density(alpha = 0.3)
```

This shows a smooth version of the discrete event count: more likely zero, then one event, then two, then three events, all of which divides the 12 months into 4, 3-month event intervals, with the last interval beyond the third month.

How much likely?

Now we get to the rub of the matter.

- How much potentially can we lose?
- This means we need to combine frequency with severity.
- We need to “convolve” the frequency and severity distributions together to form one loss distribution.
- “To convolve” means that for each simulated cyber dollar loss scenario we see whether the scenario occurs at all using the many poisson frequency scenarios.

...and this convolution can be accomplished in the following code:

```
loss <- rpois(n.sim, severity * lambda)
summary(loss)
```

##	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
##	2.00	10.00	14.00	15.02	19.00	44.00

- 1 This code takes each of the `gamma` distributed severities (`n.sim` of them) and asks how often one of them will occur (`lambda`).
- 2 It then simulates the frequency of these risk events scaled by the `gamma` severities of the loss size.
- 3 The result is a *convolution* of all `n.sim` of the severities with all `n.sim` of the frequencies. We are literally combining each scenario with every other scenario.

Let's visualize our handiwork and frame up the data. Then we will calculate the risk measures for the potential loss as:

```
loss.rf <- data.frame(Loss = loss, Distribution = rep("Potential Loss",  
  each = n.sim))  
(VaR.loss.rf <- quantile(loss.rf$Loss,  
  alpha.tolerance))
```

```
## 95%  
## 27
```

```
(ES.loss.rf <- mean(loss.rf$Loss[loss.rf$Loss >  
  VaR.loss.rf]))
```

```
## [1] 32.29268
```

Again VaR is a quantile and ES is the mean of a filter on the convolved losses.

Try this one too...

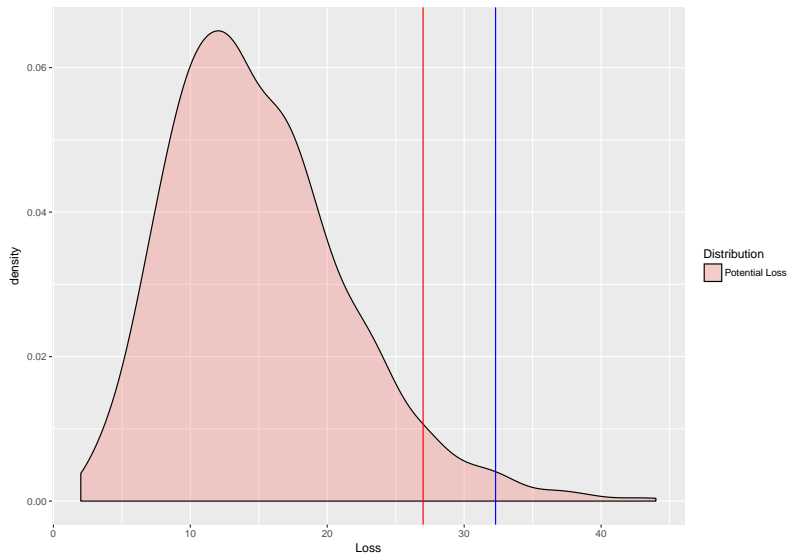
- 1 Plot the loss function and risk measures with:

```
ggplot(loss.rf, aes(x = Loss, fill = Distribution)) +  
  geom_density(alpha = 0.3) + geom_vline(xintercept = VaR.loss.rf,  
    color = "red") + geom_vline(xintercept = ES.loss.rf,  
    color = "blue")
```

- 2 What would you advise management about how much capital might be needed to underwrite these losses?

Result

```
ggplot(loss.rf, aes(x = Loss, fill = Distribution)) +  
  geom_density(alpha = 0.3) + geom_vline(xintercept = VaR.loss.rf,  
    color = "red") + geom_vline(xintercept = ES.loss.rf,  
    color = "blue")
```



- Management should consider different risk tolerances first.
- Then management should decide on the difference between the VaR (or ES) and the median loss.
- If this were a bank, then we would have calculated the capital requirement (almost Basle III).

We have history

Now suppose we have some history. - We use an extreme tail distribution called the Generalized Pareto Distribution (GPD) to estimate historical loss parameters. - This distribution of the tails of any other distribution is well-known for a single property: for very high thresholds, GPD not only well describes the behavior in excess of the threshold, but also (here is the property) the mean excess over the threshold is linear in the threshold. - From this we get more intuition around the use of expected shortfall as a coherent risk measure. - In recent years we well exceeded all Gaussian and Student's t thresholds.

For a random variate x , this distribution is defined for the scale parameter β and shape parameter $\xi \geq 0$ as:

$$g(x; \xi \geq 0) = 1 - (1 + x\xi/\beta)^{-1/\xi}.$$

When the shape parameter $\xi = 0$, the GPD becomes the exponential distribution dependent only on the scale parameter β :

$$g(x; \xi = 0) = 1 - \exp(-x/\beta).$$

Now for the infamous property. If u is an upper (very high) threshold, then the excess of threshold function for the GPD is

$$e(u) = \frac{\beta + \xi u}{1 - \xi}$$

This simple measure is linear in thresholds. It will allow us to visualize where rare events begin (see McNeil, Embrechts, and Frei (2015, chapter 5)). We will use as a threshold the exceedance that begins to make the excesses linear.

The first thing we will do is load the also-famous Danish fire claims data set in millions of Danish Kroner collected from 1980 to 1990 as a time series object.

Then we will plot the mean excesses. These are simply the mean of $e(u)$, a function of the parameters of the GPD, including the thresholds u .

```
library(QRM)
# Load Danish fire loss data and look
# at structure and content
data(danish)
str(danish)
```

```
## Time Series:
## Name:          object
## Data Matrix:
## Dimension:     2167 1
## Column Names:  FIRE.LOSSES
## Row Names:     1980-01-03 ... 1990-12-31
## Positions:
## Start:         1980-01-03
## End:           1990-12-31
## With:
## Format:        %Y-%m-%d
## FinCenter:     GMT
## Units:         FIRE.LOSSES
## Title:         Time Series Object
## Documentation:  Wed Mar 21 09:48:58 2012
```

```
summary(danish)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      1.000   1.321   1.778   3.385   2.967 263.250
```

Sort out the losses and rank order unique losses with this function.

```
n.danish <- length(danish)
# Sort and rank order unique losses
rank.series <- function(x, na.last = TRUE) {
  ranks <- sort.list(sort.list(x, na.last = na.last))
  if (is.na(na.last))
    x <- x[!is.na(x)]
  for (i in unique(x[duplicated(x)])) {
    which <- x == i & !is.na(x)
    ranks[which] <- max(ranks[which])
  }
  ranks
}
```

Now we will use this function to create the mean excess function point by point through the sorted series of loss data. In the end we want to cumulatively sum data in a series of successive thresholds.

```
danish.sorted <- sort(danish)  # From low to high
# Create sequence of high to low of
# indices for successive thresholds
n.excess <- unique(floor(length(danish.sorted) -
  rank.series(danish.sorted)))
points <- unique(danish.sorted) # Just the unique losses
n.points <- length(points)
# Take out last index and the last
# data point
n.excess <- n.excess[-n.points]
points <- points[-n.points]
# Compute cumulative sum series of
# losses
excess <- cumsum(rev(danish.sorted))[n.excess] -
  n.excess * points
excess.mean <- excess/n.excess # Finally the mean excess loss series
```


So much happened here.

- 1 Sort the data.
- 2 Construct indices for successive thresholds.
- 3 With just unique losses throw out the last data point, and its index.
- 4 Now for the whole point of this exercise: calculate the cumulative sum of losses by threshold (the `[n.excess]` operator) in excess of the cumulative threshold `n.excess*points`.
- 5 Then take the average to get the mean excess loss series.

“PHEW!”

Try this...

- 1 Build a data frame with components `Excess.Mean`, `Thresholds`, `Distribution` type.

```
library(ggplot2)
omit.points <- 3
excess.mean.df <- data.frame(Excess.Mean = excess.mean[1:(n.points -
  omit.points)], Thresholds = points[1:(n.points -
  omit.points)], Distribution = rep("GPD",
  each = length(excess.mean[1:(n.points -
  omit.points)])))
```

- 2 Plot the mean excess series against the sorted points. In this case we will omit the last three mean excess data points.

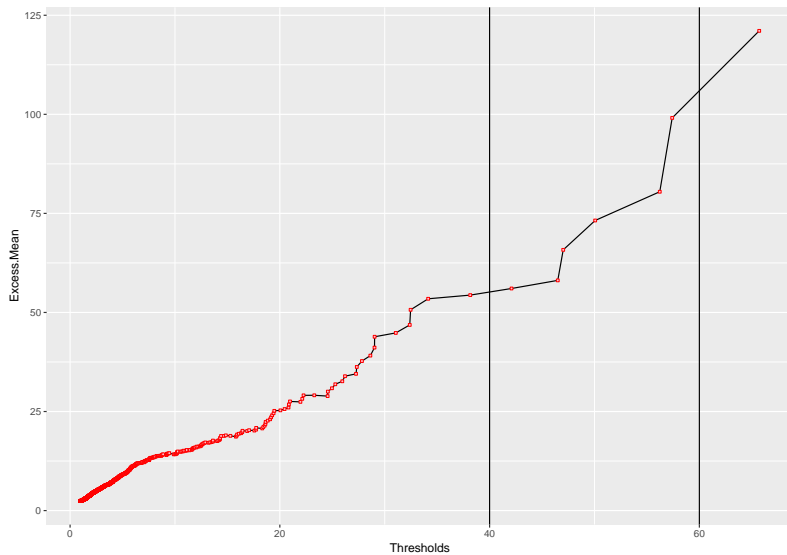
```
# Mean Excess Plot
ggplot(excess.mean.df, aes(x = Thresholds,
  y = Excess.Mean)) + geom_line() +
  geom_point(size = 1, shape = 22,
    colour = "red", fill = "pink") +
  geom_vline(xintercept = 40) + geom_vline(xintercept = 60)

# Plot density
ggplot(excess.mean.df, aes(x = Excess.Mean,
  fill = Distribution)) + geom_density() +
  xlim(0, 75)
```

Interpret for other humanoids...

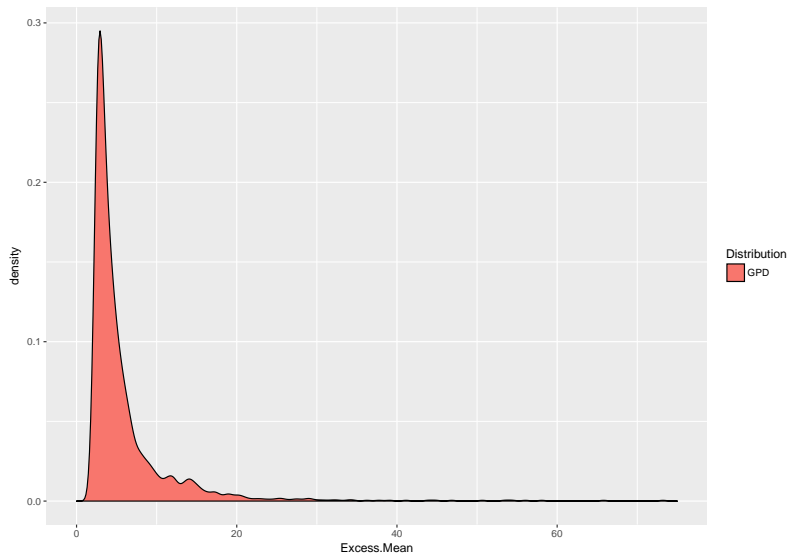
Interpretation

We use the `[1:(n.points - omit.points)]` throughout to subset the mean excess series, the thresholds, and an indicator of the type of distribution.



This plot shows that the mean of any excess over each and every point grows linearly with each threshold.

- As the mean excesses get larger, they also become more sparse, almost like outliers.
- They are the rare events we might want to mitigate.
- Look at the vertical lines at 40 and 60 million kroner to possibly design retention and limits for a contract to insure against experiencing loss in this region.



This density plot confirms the pattern of the mean excess plot.

- The mean excess distribution is understood by engineers as the mean residual life of a physical component.
- In insurance on the other hand, if the random variable X is a model of insurance losses, like the danish data set, then
- the conditional mean $E(X - u|X > u)$ is the expected claim payment per loss given that the loss has exceeded the deductible of u .
- In this interpretation, the conditional mean $E(X - t|X > t)$ is called the mean excess loss function.

Estimating the extremes

Now we borrow some Generalized Pareto Distribution code from our previous work in market risk and apply it to the danish data set.

```
# library(QRM)
alpha.tolerance <- 0.95
u <- quantile(danish, alpha.tolerance,
  names = FALSE)
fit.danish <- fit.GPD(danish, threshold = u) # Fit GPD to the excesses
(xi.hat.danish <- fit.danish$par.ses[["xi"]]) # fitted xi
```

```
## [1] 0.1351274
```

```
(beta.hat.danish <- fit.danish$par.ses[["beta"]]) # fitted beta
```

```
## [1] 1.117815
```

Okay, not much meaning here except we proved to ourselves we can run some R. Let's put these estimates to good use...

Here are the closed form calculations for value at risk and expected shortfall (no random variate simulation!) using McNeil, Embrechts, Frey (2015, chapter 5) formulae:

```
# Pull out the losses over the  
# threshold and compute excess over  
# the threshold  
loss.excess <- danish[danish > u] - u # compute the excesses over u  
n.relative.excess <- length(loss.excess)/length(danish) # = N_u/n  
(VaR.gpd <- u + (beta.hat.danish/xi.hat.danish) *  
  (((1 - alpha.tolerance)/n.relative.excess)^(-xi.hat.danish) -  
    1))
```

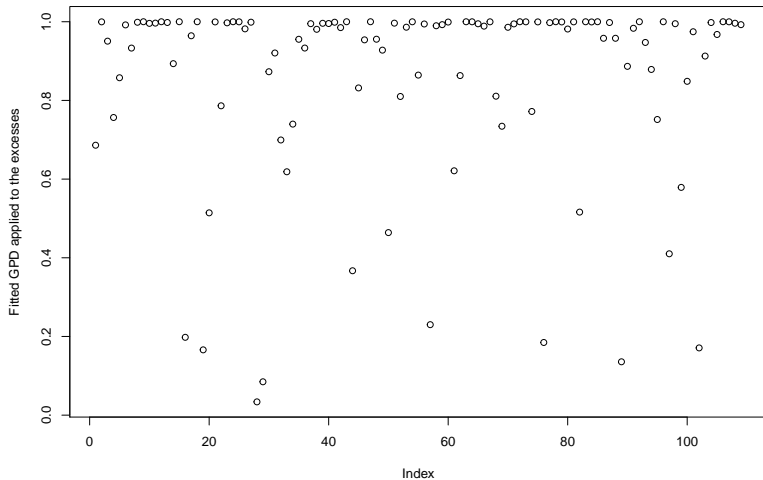
```
## [1] 9.979336
```

```
(ES.gpd <- (VaR.gpd + beta.hat.danish -  
  xi.hat.danish * u)/(1 - xi.hat.danish))
```

```
## [1] 11.27284
```

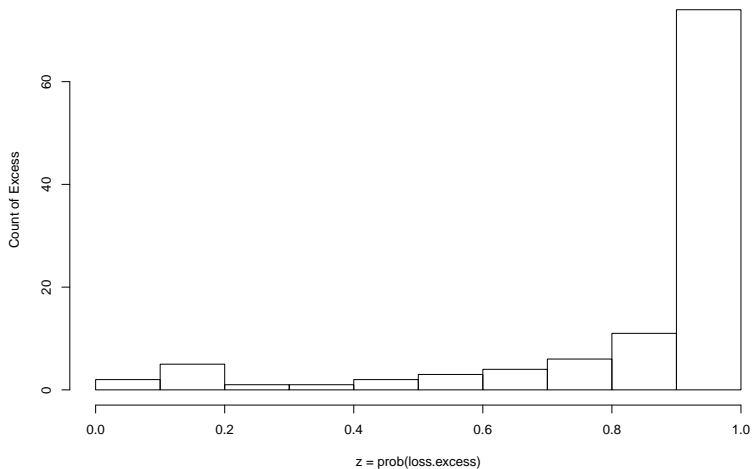
Next: How good a fit? Let's see...

```
z <- pGPD(loss.excess, xi = xi.hat.danish,  
  beta = beta.hat.danish) # should be U[0,1]  
plot(z, ylab = "Fitted GPD applied to the excesses") # looks fine
```



```
hist(z, xlab = "z = prob(loss.excess)",  
     ylab = "Count of Excess", main = "GPD Goodness of Fit")
```

GPD Goodness of Fit



And most of the excesses are in the tail!

Try this...

Let's try other thresholds to sensitize ourselves to the GPD fit and coefficients.

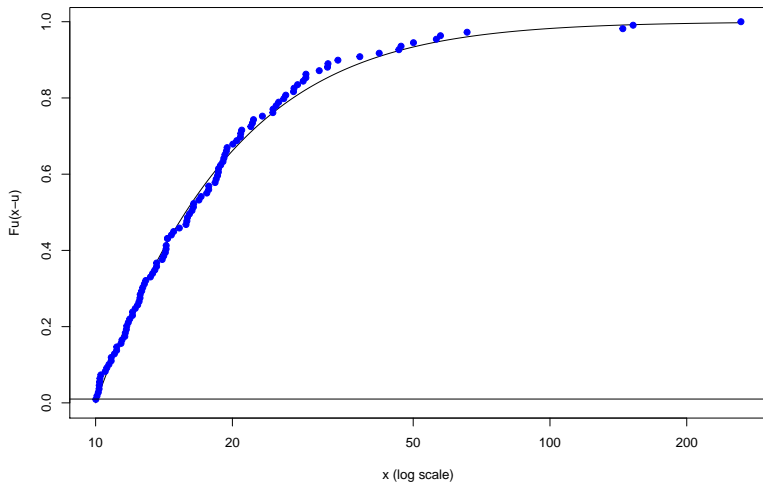
```
# Fit GPD model above 10
u <- 10
fit.danish <- fit.GPD(danish, threshold = u)
(RM.danish <- RiskMeasures(fit.danish,
  c(0.99, 0.995)))
```

```
##           p quantile      sfall
## [1,] 0.990 27.28640 58.22848
## [2,] 0.995 40.16646 83.83326
```


Now let's look at the whole picture with these exceedances.

```
plotFittedGPDvsEmpiricalExcesses(danish,  
    threshold = u)  
abline(h = 0.01)
```

- On the vertical axis we have the probabilities that you are in the tail... the extreme loss of this operation.
- On the vertical axis we have the exceedances (loss over the threshold u).
- The horizontal line is our risk tolerance (this would be h percent of the time we don't want to see a loss...).

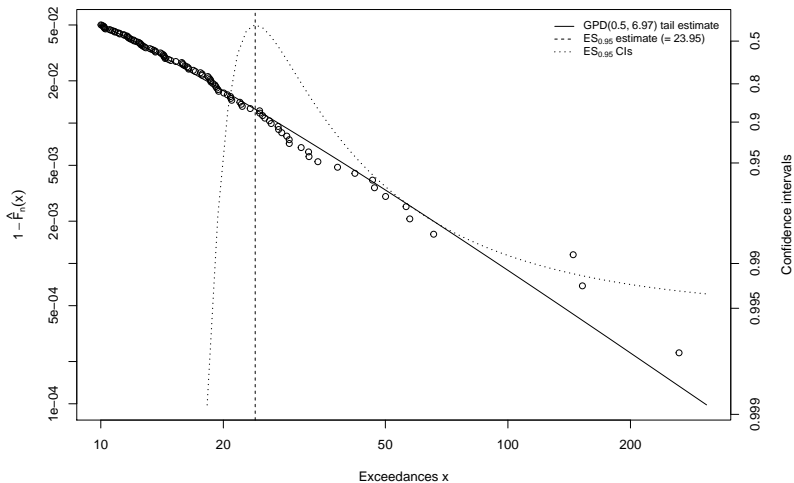


Let's really jazz up this picture ...

```
showRM(fit.danish, alpha = alpha.tolerance,  
       RM = "ES", method = "BFGS")
```

- Original `alpha.tolerance` is $1 - h$ risk tolerance.
- We are only looking at the expected shortfall.
- What we really want to see is the range of ES as we wander into the deep end of the pool with high exceedances.
- BFGS is described (all too technically) here:
https://en.wikipedia.org/wiki/Broyden%E2%80%93Fletcher%E2%80%93Goldfarb-Shanno_algorithm

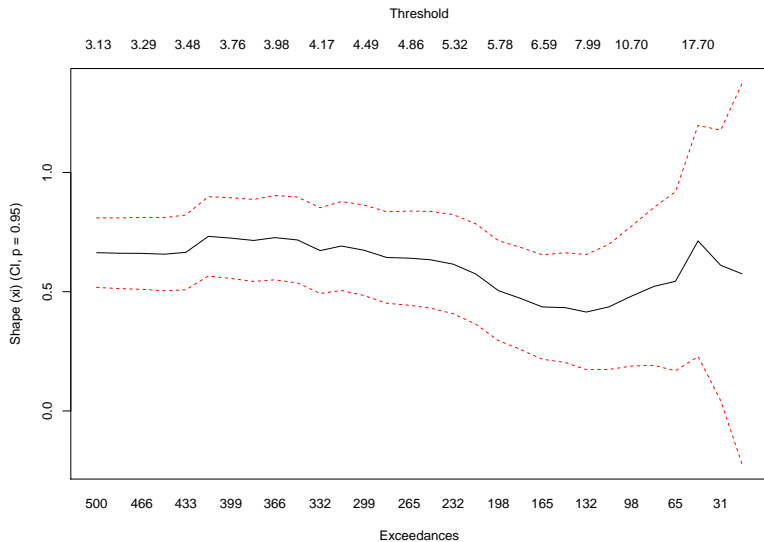
Estimated tail probabilities



CI stands for “confidence interval” and we answer these questions:

- For each level of tolerance for loss (vertical axis) what is the practical lower and upper limit on the expected shortfall of 23.95 million?
- Practically speaking how much risk would be you be willing to retain (lower bound) and how much would you underwrite (upper - lower bounds)?

```
# Effect of changing threshold on xi  
xiplot(danish)
```



What does ξ tell us?

- 1 Information about the relationship between our risk measures
- 2 The ratio of VaR to 'ES' is $(1 - \xi)^{-1}$ if $0 \leq \xi \leq 1$ and
- 3 How bent the tail will be: higher ξ means heavier tail

Again the upper and lower bounds help us diagnose what is happening to our exceedances

- Middle line is the shape parameter at various thresholds and their corresponding exceedances.
- Dashed red lines are the “river banks.”
- Fairly stable?

Now try this...

```
# Fit GPD model above 20
mod2 <- fit.GPD(danish, threshold = 20)
mod2$par.ests
mod2$par.ses
mod2$par.ests/mod2$par.ses
plotFittedGPDvsEmpiricalExcesses(danish,
  threshold = 20)
(RMs2 <- RiskMeasures(mod2, c(0.99, 0.995)))
RMs2
plotTail(mod2)
showRM(mod2, alpha = 0.99, RM = "VaR",
  method = "BFGS")
showRM(mod2, alpha = 0.99, RM = "ES",
  method = "BFGS")
```

We are just doubling the threshold.

- 1 Interpret the fit with `mod2$par.ests/mod2$par.ses`
- 2 Interpret the empirical excesses for this threshold.
- 3 Compute the risk measures.
- 4 Relate risk tolerance to exceedance.
- 5 Compare value at risk with expected shortfall.

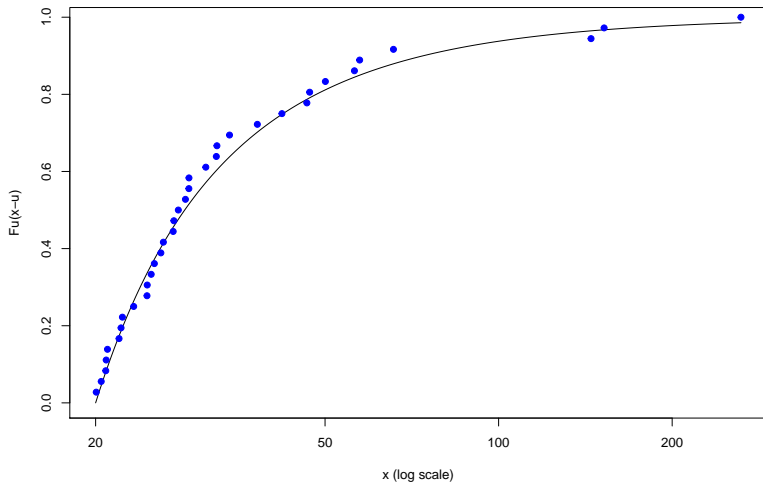
Thinking...

Result

Run this chunk:

```
plotFittedGPDvsEmpiricalExcesses(danish,  
    threshold = 20)
```

This is the cumulative probability of an exceedance over a threshold, here 20.



Various regions

- ① From 20 to about 40, fairly dense and linear
- ② From about 40 to 60, less dense and a more bent slope (ξ is bigger than for lower threshold)
- ③ Big and less frequent outliers from about 60 to well over 200.

Implications

- ① Budget for loss in region 1.
- ② Buy insurance for region 2.
- ③ Consider some loss reserves for region 3.


```
mod2 <- fit.GPD(danish, threshold = 20)
mod2$par.ests
```

```
##          xi          beta
## 0.6844366 9.6341385
```

```
mod2$par.ses
```

```
##          xi          beta
## 0.2752081 2.8976652
```

```
(t.value <- mod2$par.ests/mod2$par.ses)
```

```
##          xi          beta
## 2.486978 3.324794
```

The ratio of the parameter estimates to the standard errors of those estimates gives up an idea of the rejection of the hypothesis that the parameters are no different than zero.

In R we can do this:

```
(p.value = dt(t.value, df = length(danish) -  
2))
```

```
##          xi          beta  
## 0.018158771 0.001604948
```

... where `df` is the degrees of freedom for 2 estimated parameters.

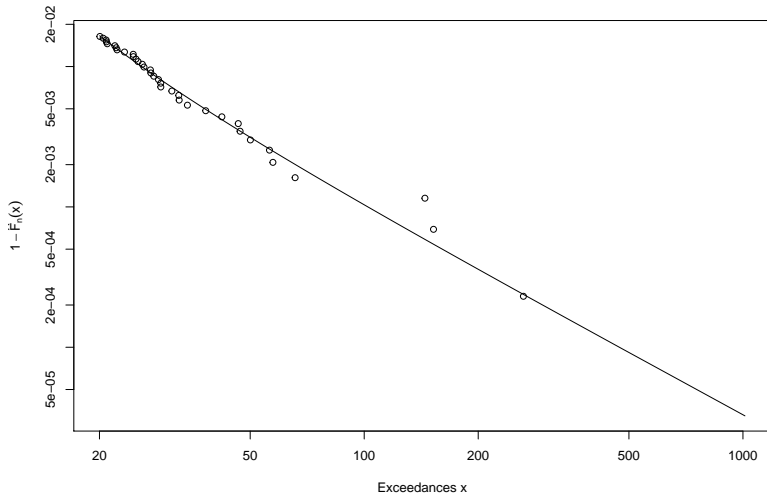
- p.values are very low.
- This means there is a very small chance that the estimates are zero.

```
(RMs2 <- RiskMeasures(mod2, c(0.99, 0.995)))
```

```
##           p quantile      sfall
## [1,] 0.990 25.84720 69.05935
## [2,] 0.995 37.94207 107.38721
```

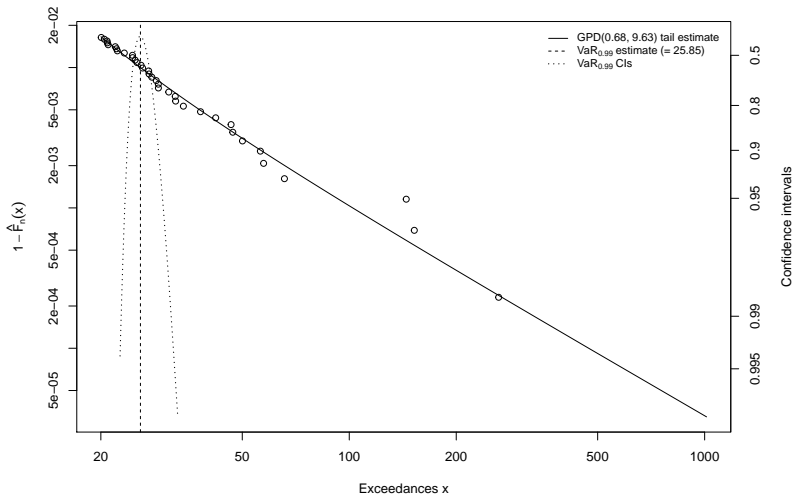
```
plotTail(mod2)
```

Estimated tail probabilities

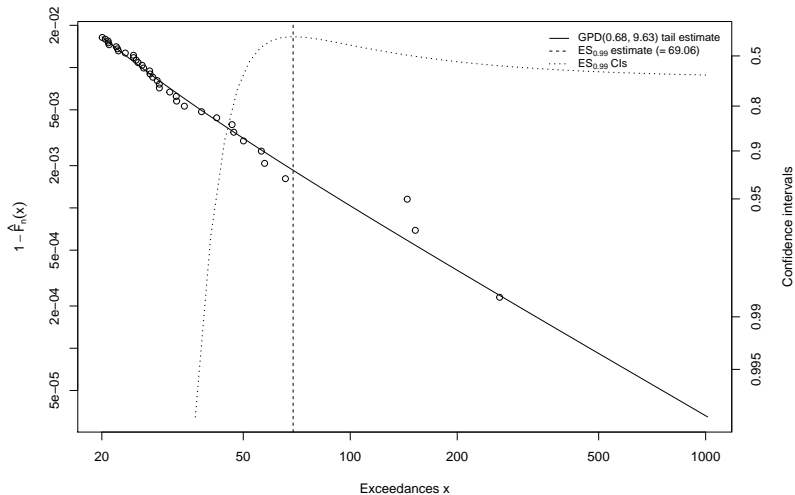


```
showRM(mod2, alpha = 0.99, RM = "VaR",  
       method = "BFGS")  
showRM(mod2, alpha = 0.99, RM = "ES",  
       method = "BFGS")
```

Estimated tail probabilities



Estimated tail probabilities



The wrap

- We learned a lot of operational maths: extreme distributions, methods from reliability and vulnerability analysis.
- We just built both simulation and estimation models that produced data driven risk thresholds, again.
- We also learned some more R and definitely some more finance.