

# Financial Engineering Analytics: A Practice Manual Using R

*Alvin Chung*

*19/02/2019*

## Contents

<b>1</b>	<b>Introduction to Financial Analytics</b>	<b>2</b>
1.1	Analytics . . . . .	2
1.2	Chapter Outline . . . . .	3
1.3	Nomenclature . . . . .	4
<b>2</b>	<b>Macrofinancial Data Analysis</b>	<b>4</b>
2.1	Imagine This . . . . .	4
2.1.1	Working an example . . . . .	5
2.1.2	How will we proceed . . . . .	5
2.2	Building the Stylized Facts . . . . .	6
2.2.1	Implications . . . . .	11
2.2.2	Now for something really interesting..again . . . . .	13
2.3	Gettings Caught in the Cross-Current . . . . .	16
2.3.1	On to quantiles . . . . .	31
2.4	Time is on our Side . . . . .	33
2.4.1	Forecasting GNP . . . . .	36
2.4.2	Residual again . . . . .	41
2.5	Give it the Boot . . . . .	44
2.6	Summary . . . . .	46
<b>3</b>	<b>Term Structure and Splines</b>	<b>47</b>
3.1	The Bond . . . . .	47
3.1.1	A quick example . . . . .	48
3.1.2	What's a spline? . . . . .	48
3.1.3	Back to the Bond . . . . .	49
3.1.4	An example to clarify . . . . .	51
3.1.5	Rolling the integration . . . . .	52
3.2	Foward Rate Parameters . . . . .	52
<b>4</b>	<b>Market Risk</b>	<b>53</b>
4.1	What is Market Risk . . . . .	53
4.1.1	Try this exercise . . . . .	53
4.2	History Speaks . . . . .	54
4.3	Try this exercise . . . . .	58
4.4	Now to the Matter at Hand . . . . .	60
4.4.1	Example . . . . .	64
4.5	Carl Friedrich Guass, I Pressume... . . . .	66
4.6	Back to the Future . . . . .	69
4.7	Try this example . . . . .	71
4.8	Let's go to Extremes . . . . .	73
4.8.1	Example . . . . .	74
4.9	All Together Now . . . . .	75

<b>5</b>	<b>Credit Risk (Missing)</b>	<b>77</b>
<b>6</b>	<b>Operational Risk and Extreme Finance</b>	<b>77</b>
6.1	Imagine this . . . . .	77
6.2	What is Operational Risk . . . . .	77
6.2.1	Example . . . . .	78
6.3	How Much? . . . . .	78
6.3.1	Severity thresholds . . . . .	80
6.4	How Often . . . . .	81
6.5	How Much Potential Loss? . . . . .	83
6.5.1	Example . . . . .	83
6.6	We Have History . . . . .	86
6.7	Fire Losses . . . . .	86
6.8	Example . . . . .	88
6.9	Estimating the Extremes . . . . .	90
6.10	Example . . . . .	93
6.10.1	THE shape of things to come . . . . .	95
6.10.2	Example . . . . .	96
6.11	Summary . . . . .	100
<b>7</b>	<b>Measuring Volatility</b>	<b>101</b>
7.1	Introduction . . . . .	101
7.1.1	Example . . . . .	101
7.2	What is ALL the Fuss About? . . . . .	101
7.3	It is Fitting... . . . .	106
7.3.1	Example . . . . .	109
7.4	Simulate... again until Morale Improves... . . . .	116
7.4.1	Example . . . . .	123
7.5	Multivariate GARCH models . . . . .	125
7.6	Finishing . . . . .	132
7.7	Example . . . . .	135
<b>8</b>	<b>Aggregating Enterprise Risk</b>	<b>136</b>
8.1	Introduction: The Problem with Enterprise Risk . . . . .	136
8.2	Copula . . . . .	137
8.2.1	From scratch . . . . .	137
8.3	Sklar's in the house... . . . .	138
8.4	Analyze the results . . . . .	140
8.4.1	Example . . . . .	141
8.5	Risk Measures . . . . .	142
8.5.1	Example . . . . .	143
<b>9</b>	<b>Finished</b>	<b>146</b>
9.1	Congratulations . . . . .	146

# 1 Introduction to Financial Analytics

## 1.1 Analytics

By its very nature the science of data analytics is disruptive. That means, among many other things, that much attention should be paid to the scale and range of invalid, as yet not understood, outlying, and emerging trends. This is as true within the finance domain of knowledge as any other.

Throughout the book, we will learn the core of ideas of programming software development to implement financial analyses (functions, objects, data structures, flow control, input and output, debugging, logical design and abstraction) through writing code. We will learn how to setup stochastic simulations, manage data analyses, and employ numerical optimization algorithms, diagnose their limitations, and work with and filter large data sets. Since code is also an important form of communication among analysts, we will learn how to comment and organize code, as well as document work product.

## 1.2 Chapter Outline

Here is an outline of topics covered by chapter.

**2. R Warm-Ups for Finance.** R computations, data structures, financial, probability and statistics calculations, visualization. Documentation with **R Markdown**

**3. More R Warm-Ups.** Functions, loops, control bootstrapping, simulations and more visualization.

**4. Stylized Facts of Financial Markets.** Data from FRED, Yahoo, and other sources. Empirical characteristics of economic and financial time series. Bootstrapping confidence intervals.

**5. Term Structure of Interest Rates.** Bond pricing, forward and yield curves. Estimating Non-linear regression splines. Applications.

**6. Market Risk.** Quantile (i.e., Value at Risk) and coherent (i.e., Expected Shortfall) risk measures

**7. Credit Risk.** Hazard rate models, Markov transition probabilities Risk measures, Laplace simulation with FFT.

**8. Operational Risk and Extreme Finance.** Generate frequency and severity of operational loss distribution. Estimating operational risk distribution parameters. Simulating loss distributions.

**9. Measuring Volatility.** Measuring volatility. GARCH estimation. GARCH simulation. Measuring Value at Risk (VaR) and Expected Shortfall (ES).

**10. Portfolio Optimization.** Combining risk management with portfolio allocations. Optimizing allocations. Simulating the efficient frontier.

**11. Aggregating Enterprise Risk.** Enterprise risk management analytics and applications. Workflow to build an online application. Introduction to **Shiny** and **ShinyDashboard**. Building a simple app. Using **R Markdown** and **Shiny**.

### 1.3 Nomenclature

Symbol	Name	Usage
$\alpha$	alpha	significance level
$\beta$	beta	regression coefficient, scale parameter
$\gamma$	gamma	autocorrelation
$\Gamma$	Gamma	Gamma function
$\delta$	delta	delta
$\Delta$	Delta	change
$\epsilon$	epsilon 1	is element of
$\varepsilon$	epsilon 2	error term
$\xi$	ksi	shape parameter
$\lambda$	lambda	hazard rate
$\mu$	mu	mean of a distribution
$\nu$	nu	number of degrees of freedom
$\pi$	pi	probability of occurrence
$\rho$	rho	correlation coefficient
$\sigma$	sigma	standard deviation
$\Sigma$	sum sigma	arithmetic sum
$\tau$	tau	quantile

## 2 Macrofinancial Data Analysis

### 2.1 Imagine This

Your US-based company just landed a contract worth more than 20 percent of your company's current revenue in Spain. Now that everyone has recovered from this coup, your management wants you to

1. Retrieve and begin to analyze data about the Spanish economy
2. Compare the contrast Spanish stock market and government-issued debt value versus the United States and several other countries
3. Begin to generate economic scenarios based on political events that may, or may not, happen in Spain

Up to this point we had reviewed several ways to manipulate data in **R**. We then reviewed some basic finance and statistics concepts in **R**. We also got some idea of the financial analytics workflow.

1. What decision(s) are we making?
2. What are they key business questions we need to support this decision?
3. What data do we need?
4. What tools do we need to analyze the data?
5. How do we communicate answers to inform the decision?

### 2.1.1 Working an example

Let's use this workflow to motivate our work in this chapter.

1. Let's identify a decision at work (e.g., investment in new machine, financing a building, acquisition of customers, hiring talent, locating manufacturing).
2. For this decision we will list three business questions you need to inform the decision we chose.
3. Now we consider data we might need to answer one of those questions and choose from this set:
  - Macroeconomic data: GDP, inflation, wage, population
  - Financial data: stock market prices, bond prices, exchange rates, commodity prices

Here is the example using the scenario that started this chapter.

1. Our decision is **supply a new market segment**
  - Product: voltage devices with supporting software
  - Geography: Spain
  - Customers: Major buyers at Iberdrola, Repsol, and Endesa
2. We pose three business questions:
  - How would the performance of these companies affect the size and timing of orders?
  - How would the value of their product affect the value of our business with these companies?
  - We are a US functional currency firm, so how would we manage the repatriation of accounts receivable from Spain?
3. Some data and analysis to inform the decision could include
  - Customer stock prices: volatility and correlation
  - Oil prices: volatility
  - USD/EUR exchange rates: volatility
  - All together: correlation among these indicators

### 2.1.2 How will we proceed

This chapter will develop stylized facts of the market. These continue to be learned the hard way: financial data is not independent, it possesses volatile volatility, and has extremes.

- \* Financial stock, bonds, commodity have highly interdependent relationships
- \* Volatility is rarely constant and often has a structure (mean reversion) and is dependent on the past
- \* Past shocks persist and may or may not dampen (rock in a pool).
- \* Extreme events are likely to happen with other extreme events.
- \* Negative returns are more likely than positive returns (left skew).

## 2.2 Building the Stylized Facts

Examples from the 70s, 80s, and 90s have multiple intersecting global events influencing decision makers. We will load some computational help and some data from Brent, format dates, and create a time series object (package zoo will be needed)

```
library(fBasics)
library(evir)
library(qrmdata)
library(zoo)
data("OIL_Brent")
str(OIL_Brent)
```

```
## An 'xts' object on 1987-05-20/2015-12-28 containing:
##   Data: num [1:7258, 1] 18.6 18.4 18.6 18.6 18.6 ...
##   - attr(*, "dimnames")=List of 2
##     ..$ : NULL
##     ..$ : chr "OIL_Brent"
##   Indexed by objects of class: [Date] TZ: UTC
##   xts Attributes:
##   NULL
```

We will compute rates of change for Brent oil prices next.

```
Brent.price <- as.zoo(OIL_Brent)
str(Brent.price)
```

```
## 'zoo' series from 1987-05-20 to 2015-12-28
##   Data: num [1:7258, 1] 18.6 18.4 18.6 18.6 18.6 ...
##   - attr(*, "dimnames")=List of 2
##     ..$ : NULL
##     ..$ : chr "OIL_Brent"
##   Index: Date[1:7258], format: "1987-05-20" "1987-05-21" "1987-05-22" "1987-05-25" "1987-05-26" ...
```

```
Brent.return <- diff(log(Brent.price))[-1] * 100
colnames(Brent.return) <- "Brent.return"
head(Brent.return, n = 5)
```

```
##           Brent.return
## 1987-05-22    0.5405419
## 1987-05-25    0.2691792
## 1987-05-26    0.1611604
## 1987-05-27   -0.1611604
## 1987-05-28    0.0000000
```

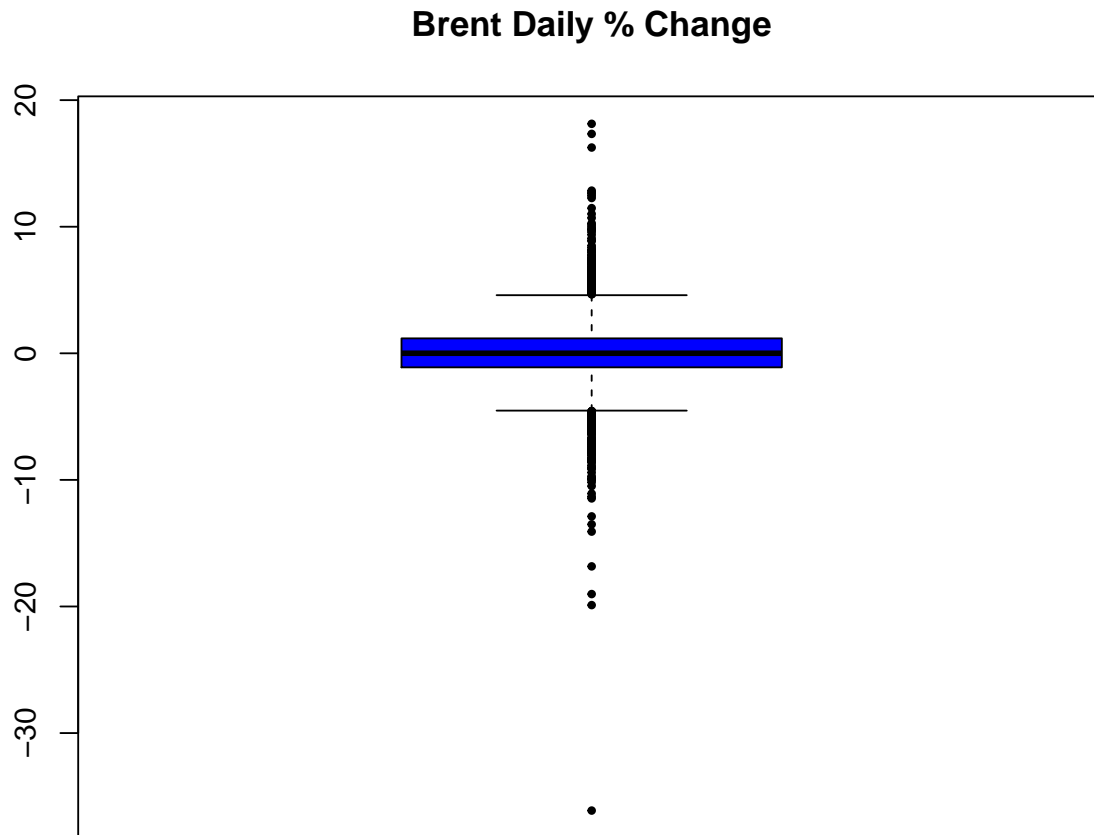
```
tail(Brent.return , n = 5)
```

```
##           Brent.return
## 2015-12-21   -3.9394831
## 2015-12-22   -0.2266290
## 2015-12-23    1.4919348
## 2015-12-24    3.9177726
## 2015-12-28   -0.3768511
```

Let's look at this data with box plots and autocorrelation functions. Box plots will show minimum to maximum with the mean in the middle of the box. Autocorrelation plots will reveal how persistent the returns are over time

We run these statements.

```
boxplot(as.vector(Brent.return), title = FALSE,  
        main = "Brent Daily % Change", col = "blue",  
        cex = 0.5, pch = 19)
```



```
skewness(Brent.return)
```

```
## [1] -0.6210447  
## attr("method")  
## [1] "moment"
```

```
kurtosis(Brent.return)
```

```
## [1] 14.62226  
## attr("method")  
## [1] "excess"
```

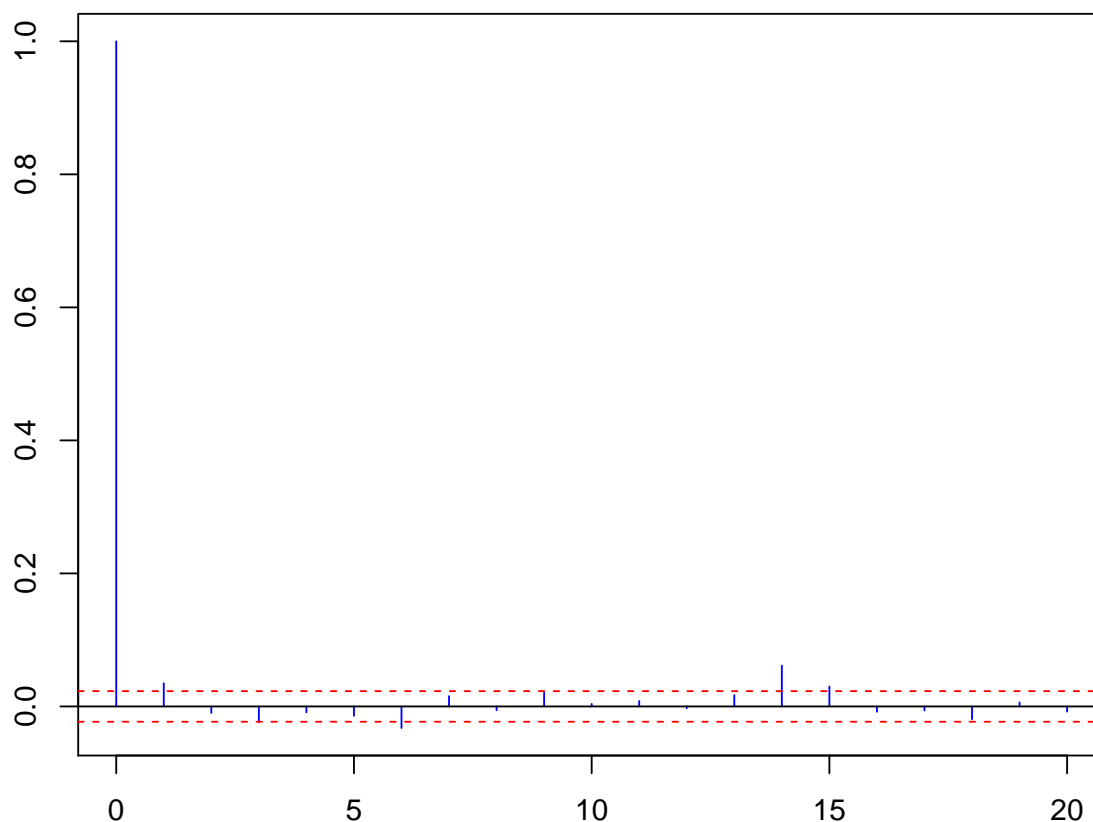
This time series plot shows lots of return clustering and spikes, especially negative ones. Performing some “eyeball econometrics” these clusters seem to occur around - The oil embargo of the ‘70s - The height of the new interest rate regime in Paul Volcker at the Fed - “Black Monday” stock market crash in 1987 - Gulf I - Baring and other derivative business collapses in the 90s’

2. Let's look at the likelihood of a positive versus negative return. We might want to review the skewness and kurtosis definition and range to help us.

Now to look at persistence:

```
acf(coredata(Brent.return), main = "Brent Daily Autocorrelogram",  
    lag.max = 20, ylab = "", xlab = "",  
    col = "blue", ci.col = "red")
```

**Brent Daily Autocorrelogram**



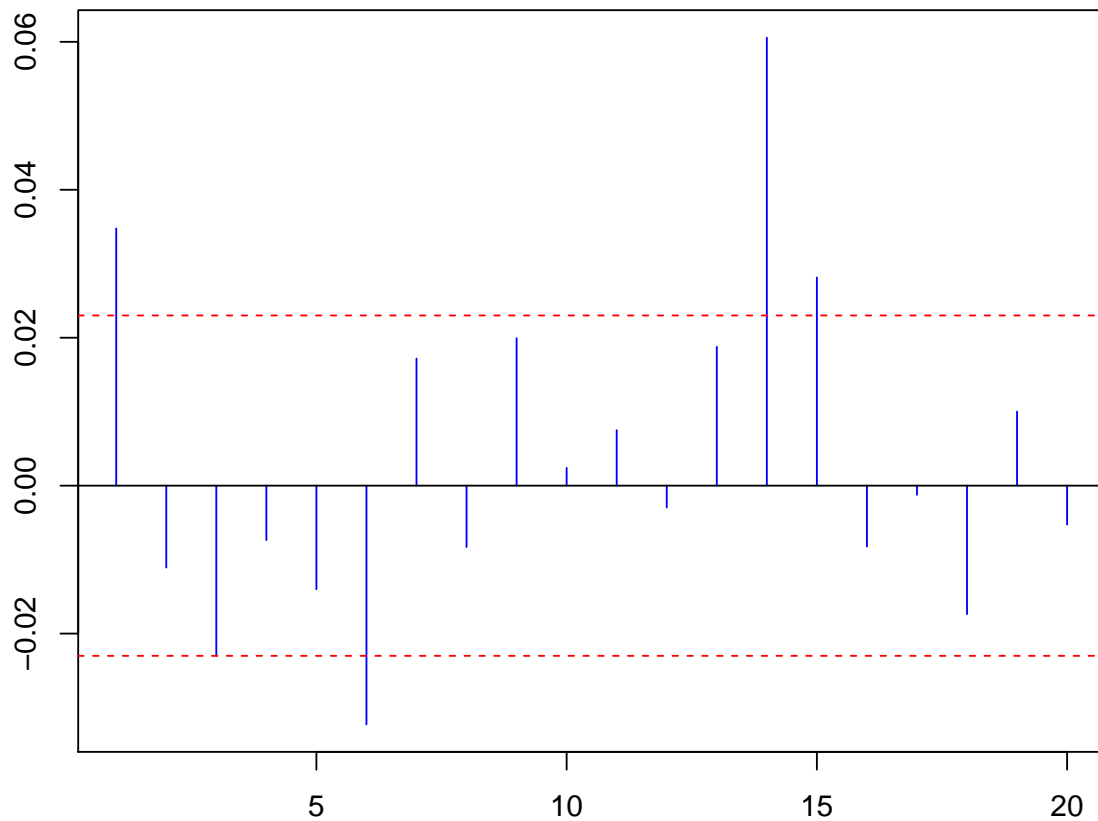
```
class(Brent.return)
```

```
## [1] "zoo"
```

```
pacf(coredata(Brent.return), main = "Brent Daily Partial Autocorrelogram",  
     lag.max = 20, ylab = "", xlab = "",  
     col = "blue",  
     ci.col = "red")
```



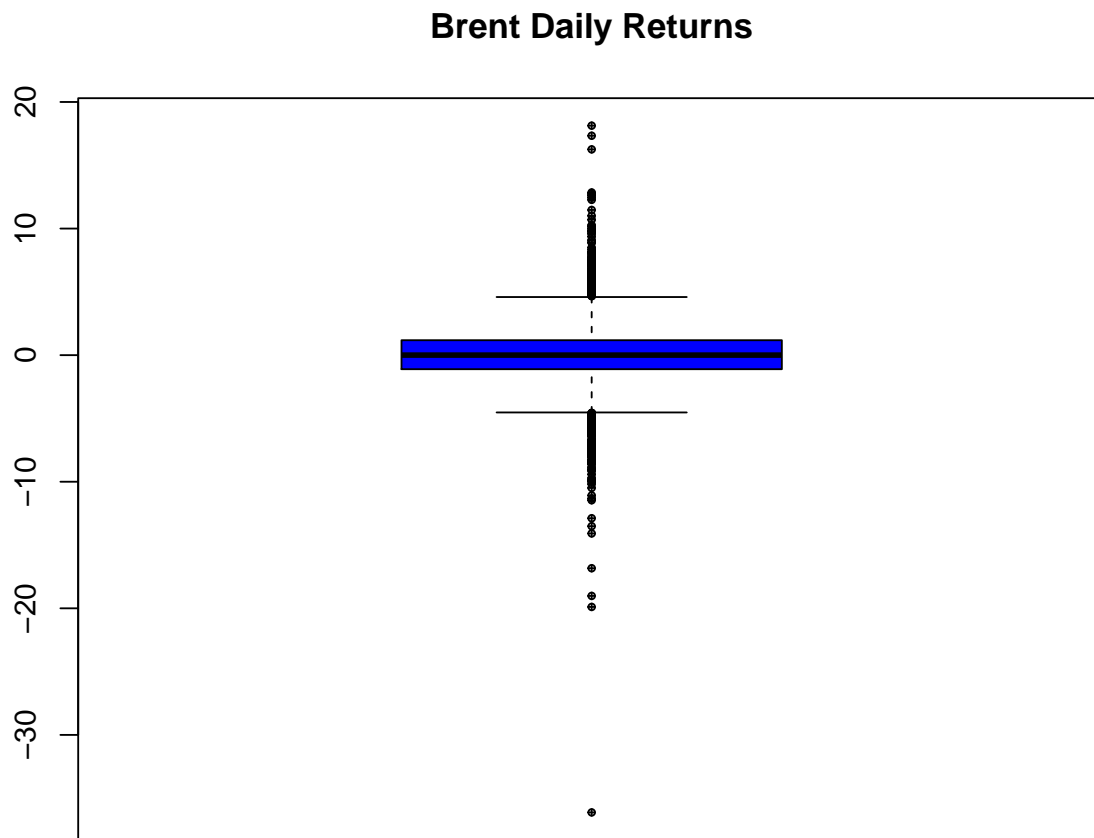
## Brent Daily Partial Autocorrelogram



Confidence interval are the red dashed lines. ACF at lag 6 means the correlation of current returns correlate with returns 6 trading days ago, including any correlation from trading day 1 to trading 6. PACF is simpler: it is the raw correlation between day 0 and day 6. ACF starts at lag 0 (today); PACF starts at lag 1 (yesterday).

Here is a first look:

```
boxplot(as.vector(Brent.return), title = FALSE,  
        main = "Brent Daily Returns", col = "blue",  
        cex = 0.5, pch = 10)
```



with some basic stats to back up the eyeball econometrics in the box plot

```
skewness(Brent.return)
```

```
## [1] -0.6210447  
## attr(,"method")  
## [1] "moment"
```

```
kurtosis(Brent.return)
```

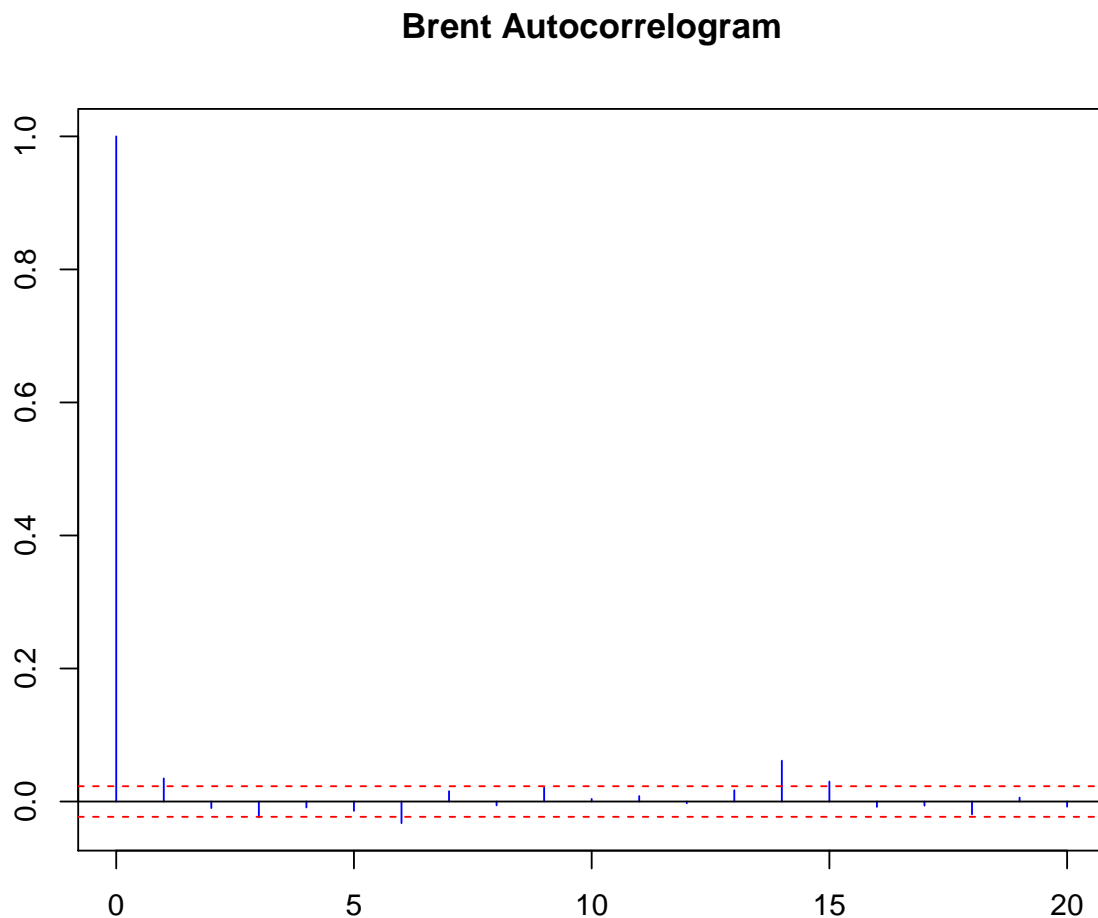
```
## [1] 14.62226  
## attr(,"method")  
## [1] "excess"
```

- A negative skew means there are more observations less than the median than greater.
- This high kurtosis means a pretty heavy tail, especially in negative returns. That means they have happened more often than positive returns.
- A preponderance of negative returns frequently happening spells trouble for anyone owning these assets.

### 2.2.1 Implications

- We should recommend that management budget for the body of the distribution from the mean and out to positive levels.
  - At the same time management should build a comprehensive playbook for the strong possibility that bad tail events frequently happen and might happen again (and why shouldn't they?)
3. Now for something really interesting

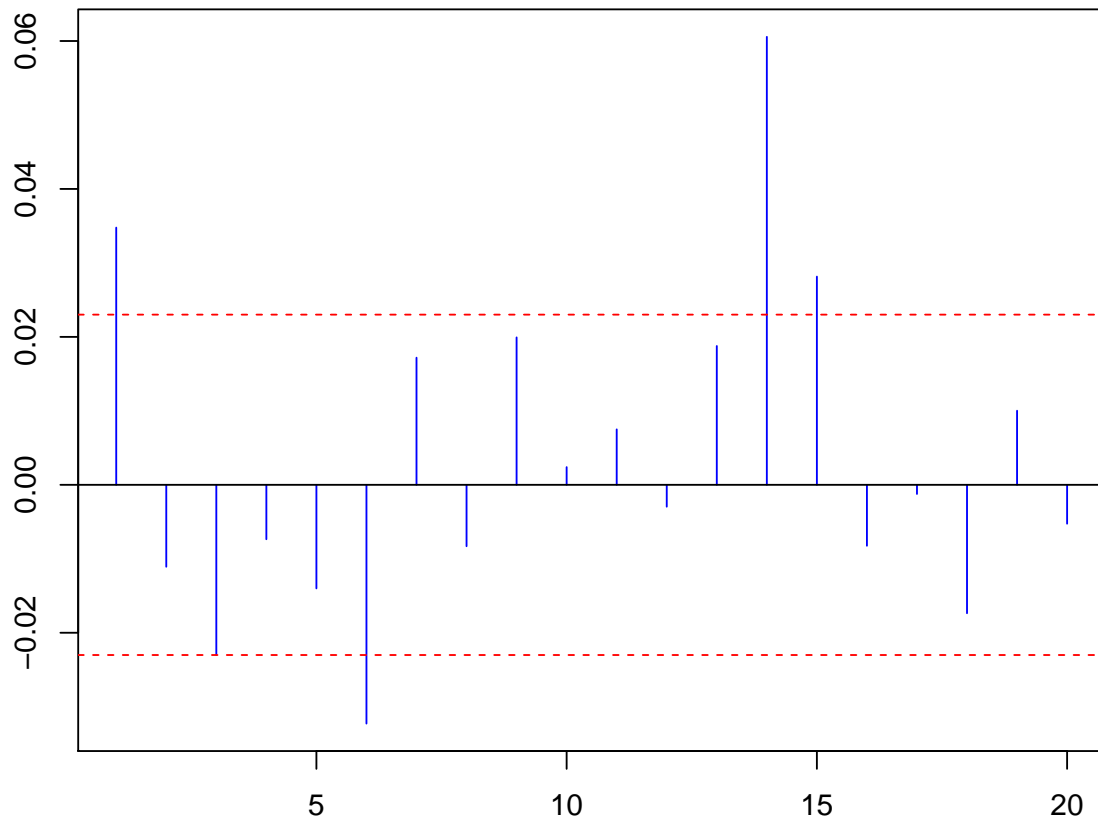
```
acf(coredata(Brent.return), main = "Brent Autocorrelogram",  
    lag.max = 20, ylab = "", xlab = "",  
    col = "blue",  
    ci.col = "red")
```



```
pacf(coredata(Brent.return),  
    main = "Brent Partial Autocorrelogram",  
    lag.max = 20,  
    ylab = "",  
    xlab = "",  
    col = "blue",
```

```
ci.col = "red"  
)
```

### Brent Partial Autocorrelogram



On average there are 5 days in the trading week and 20 in the trading month.

Some further thoughts:

- There seems to be positive weekly and negative monthly cycles.
- On a weekly basis negative rates (5 trading days ago) are followed by negative rates (today) and vice-versa with positive rates.
- On a monthly basis negative rates (20 days ago) are followed by positive rates (today).
- There is memory in the markets: positive correlation at least weekly up to a month ago reinforces the strong and frequently occurring negative rates (negative skew and leptokurtotic a.k.a heavy tails)
- Run the PACF for 60 days to see a 40-day negative correlation as well.

### 2.2.2 Now for something really interesting..again

Let's look just at the size of the Brent returns. The absolute value of the returns (think of oil and countries entering and leaving the EU!) can signal contagion, herd mentality, and simply very large margin calls (and the collateral to back it all up!) Let's run this code:

```
Brent.return.abs <- abs(Brent.return)
## Trading position size matters
Brent.return.tail <- tail(Brent.return.abs[order(Brent.return.abs)], 100)[1]
## Take just the first 100
## observations and pick the first

index <- which(Brent.return.abs > Brent.return.tail,
              arr.ind = TRUE)
## Build an index of those sizes that
## exceed the heavy tail threshold

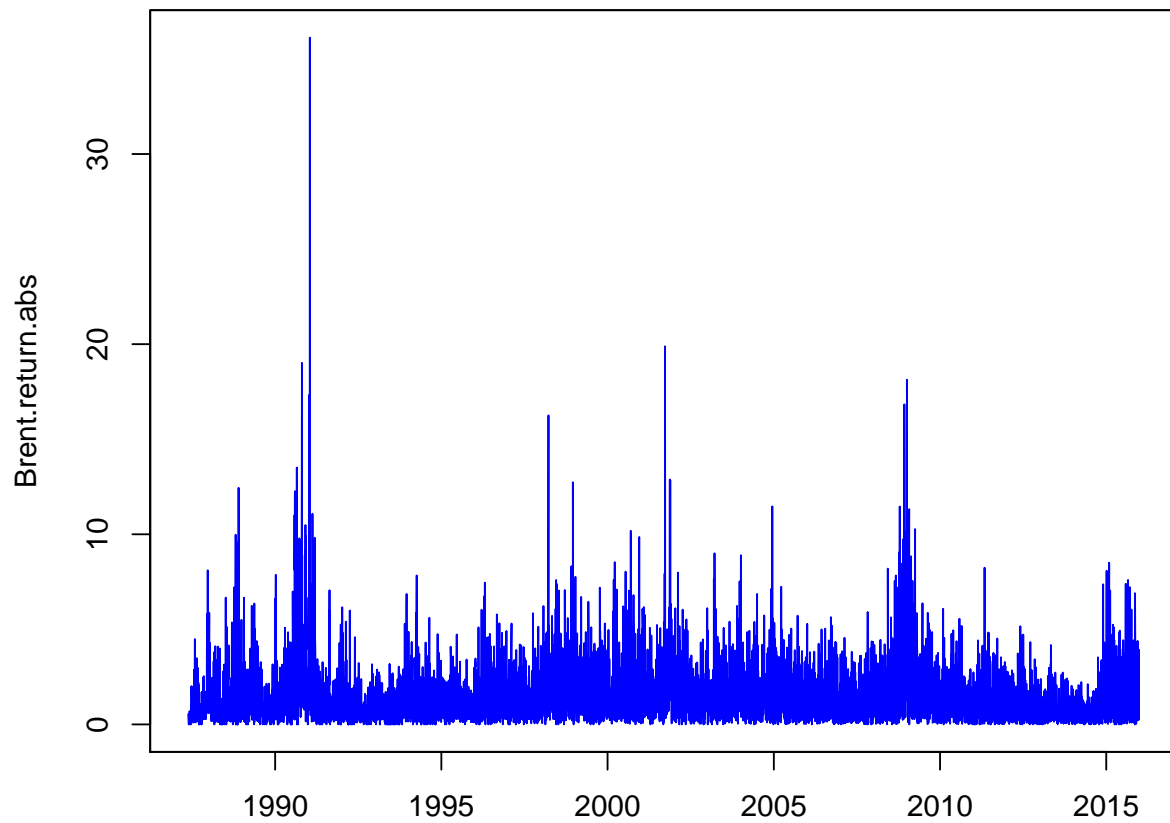
Brent.return.abs.tail <- timeSeries(rep(0, length(Brent.return)),
                                   charvec = time(Brent.return))
# just a lot of zeros we will fill up
# next

Brent.return.abs.tail[index, 1] <- Brent.return.abs[index]
```

What did we do? Let's run some plots next.

```
plot(Brent.return.abs, xlab = "", main = "Brent Daily Return Sizes",
     col = "blue")
```

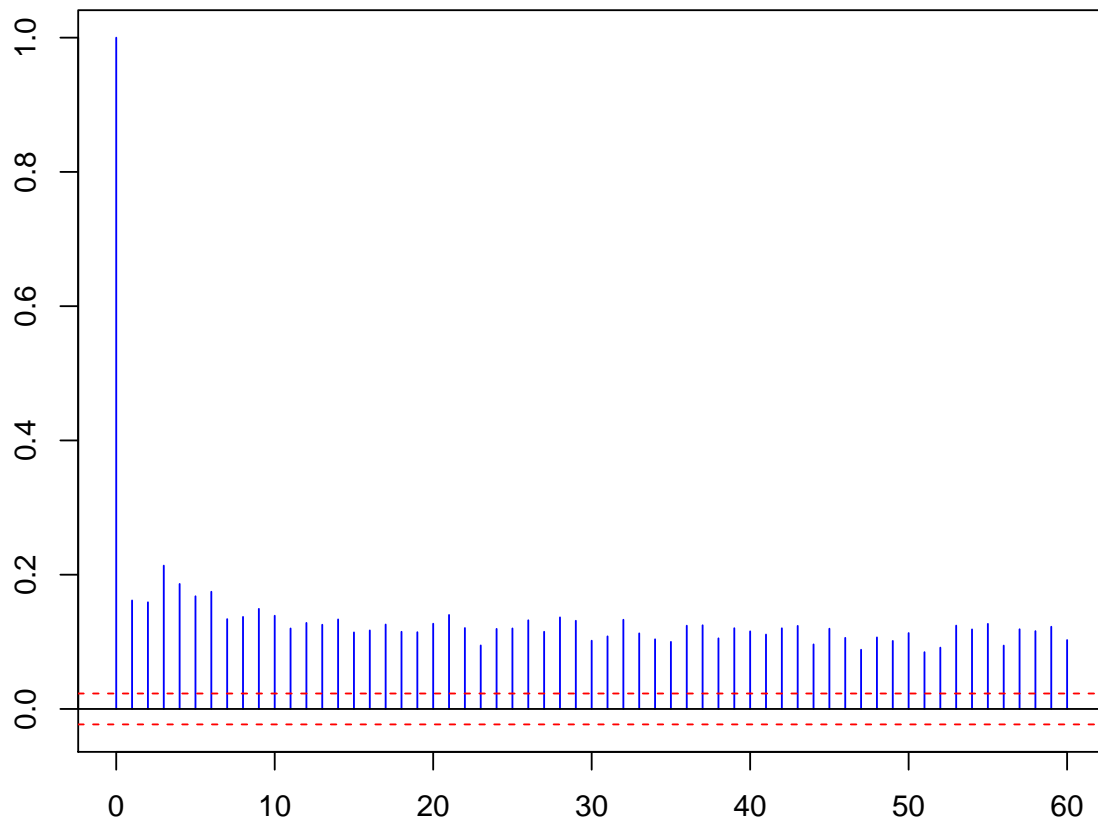
## Brent Daily Return Sizes



We see lots of return volatility - just in the pure size along. These are correlated with financial innovations from the 80s and 90s as well as Gulf 1, Gulf 2, Great Recession, and its 9/11 antecedents.

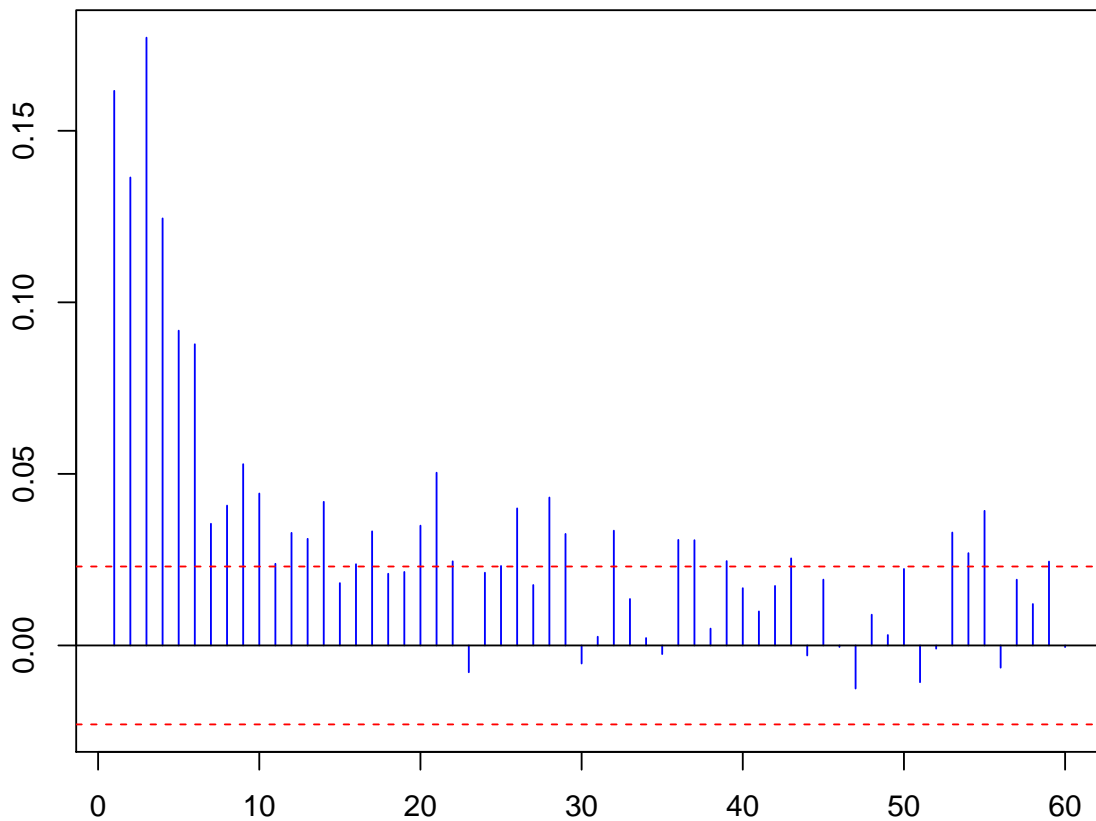
```
acf(coredata(Brent.return.abs), main = "Brent Autocorrelogram",  
    lag.max = 60, ylab = "", xlab = "",  
    col = "blue", ci.col = "red")
```

## Brent Autocorrelogram



```
pacf(coredata(Brent.return.abs), main = "Brent Partial Autocorrelogram", lag.max = 60, ylab = "", xlab = "Lag",  
     col = "blue", ci.col = "red")
```

## Brent Partial Autocorrelogram



There is *Volatility Clustering galore*. Strong persistent lags of absolute movement in returns evidenced by the ACF plot. There is evidence of dampening with after shocks past trading 10 days ago. Monthly volatility affects today's performance.

Some of this volatility arises from the way Brent is traded. It is lifted through well-heads in the North Sea. It then is scheduled for loading onto ships and loads are then bid, along with routes to destination. It takes about five days to load crude and another five to unload. At each partial loading and unloading, the cruid is re-priced. Then there is the voyage loag itself, where paper claims to wet crude create further pricing and volatility.

Next we explore the relationship among financial variables.

### 2.3 Gettings Caught in the Cross-Current

Now our job is to ask the really important questions around connectivity. Suppose we are banking our investment in certain sectors of an economy, with its GDP, financial capability, employment, exports and imports, and so on.

- How will we decide to contract for goods and services, segment vendors, segment customers, based on these interactions?



- How do we construct our portfolio of business opportunities?
- How do we identify insurgent and relational risks and build a playbook to manage these?
- How will changes in one sector's factors (say, finance, political will) affect factors in another?

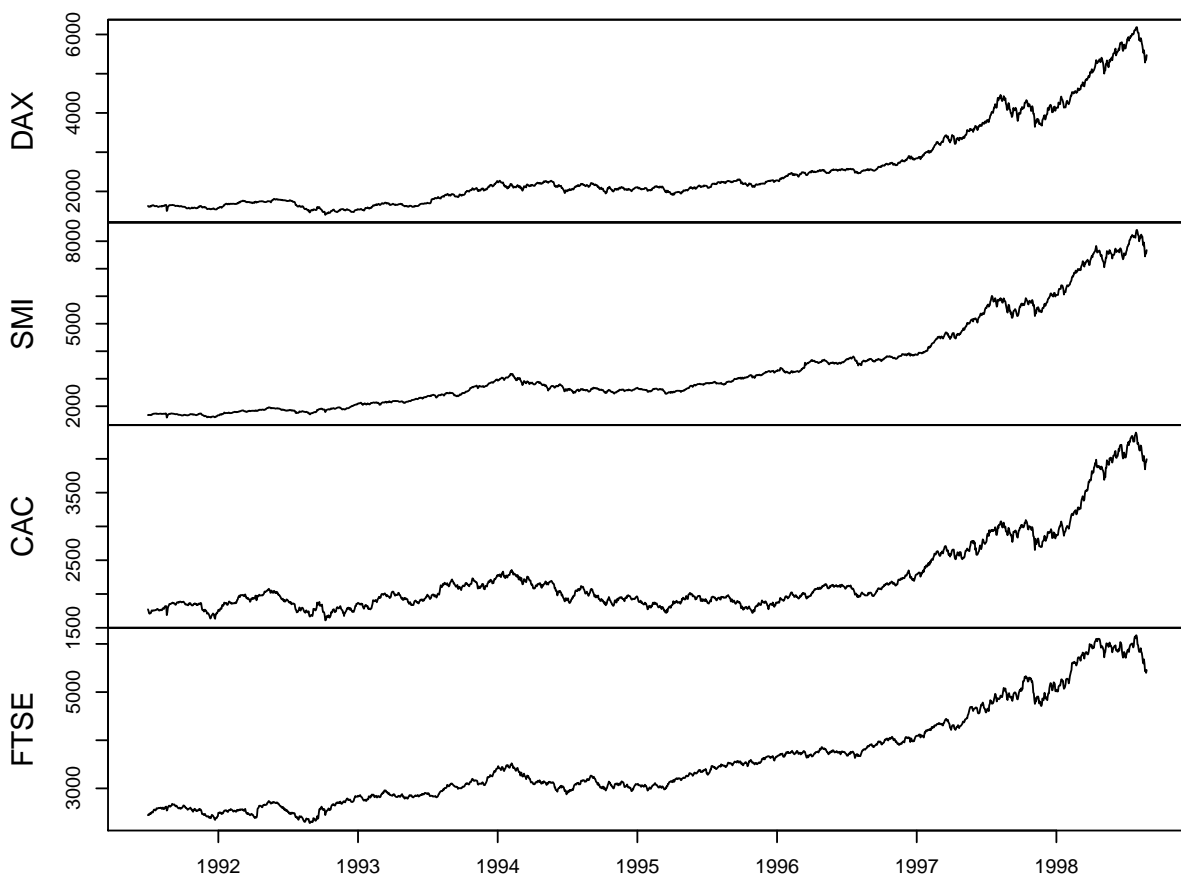
We will not stretch our univariate analysis a bit and look at *cross-correlations* to help us get the ground truth around these relationships, and *begin* to answer some of these business questions in a more specific context.

Let's load the `zoo` and `qrmdata` libraries first and look at the `EuroStoxx50` data set. Here we can imagine we are rebuilding our brand and footprint in the European Union and United Kingdom. Our customers might be the companies based in these countries as our target market.

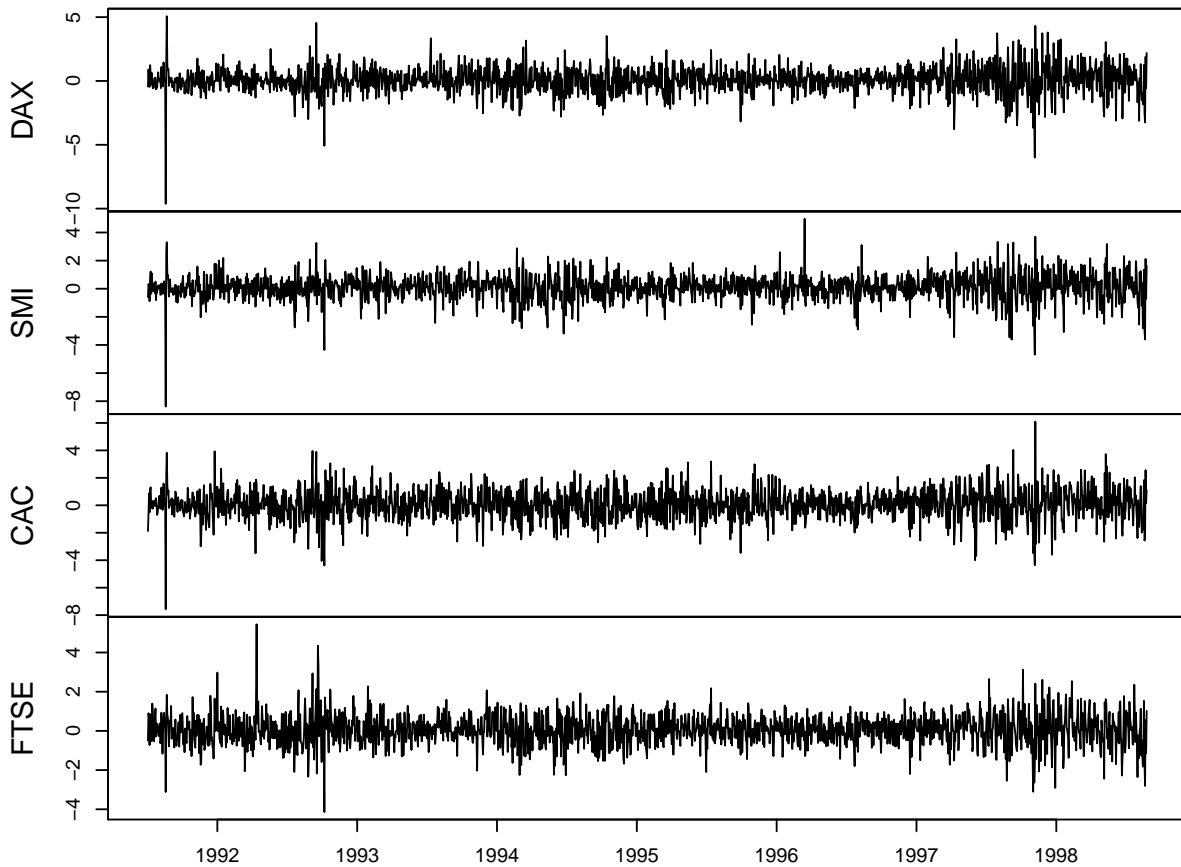
- The data: 4 stock exchange indices across Europe (and the United Kingdom)
- This will allow us to profile the forward capabilities of these companies across their economies.
- Again we will look at returns data using the `diff(log(data))[-1]`

```
library(zoo)
library(qrmdata)
library(xts)
data("EuStockMarkets")
EuStockMarkets.price <- as.zoo(EuStockMarkets)
EuStockMarkets.return <- diff(log(EuStockMarkets.price))[-1] * 100

plot(EuStockMarkets.price, xlab = "", main = "")
```



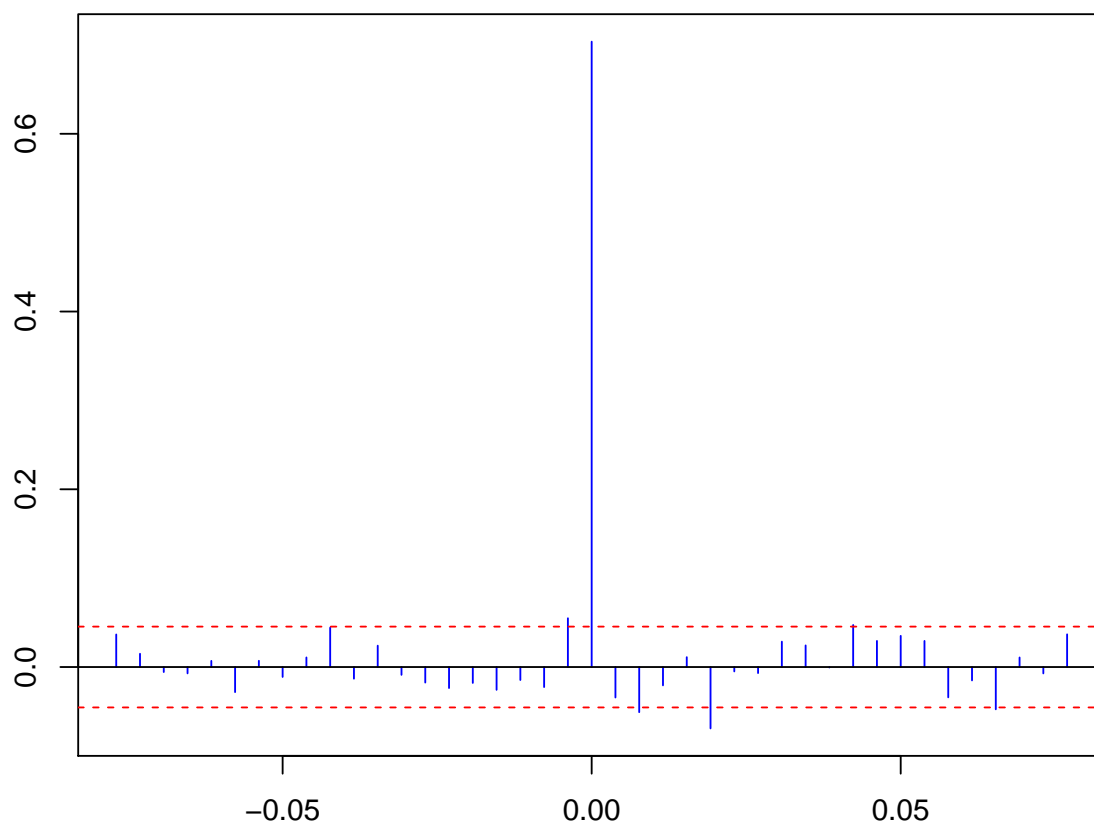
```
plot(EuStockMarkets.return, xlab = " ", main = " ")
```



Let's then look at cross-correlation among one pair of these indices to see how they are related across time (lags) for returns and the absolute value of returns. The function `ccf` will aid us tremendously.

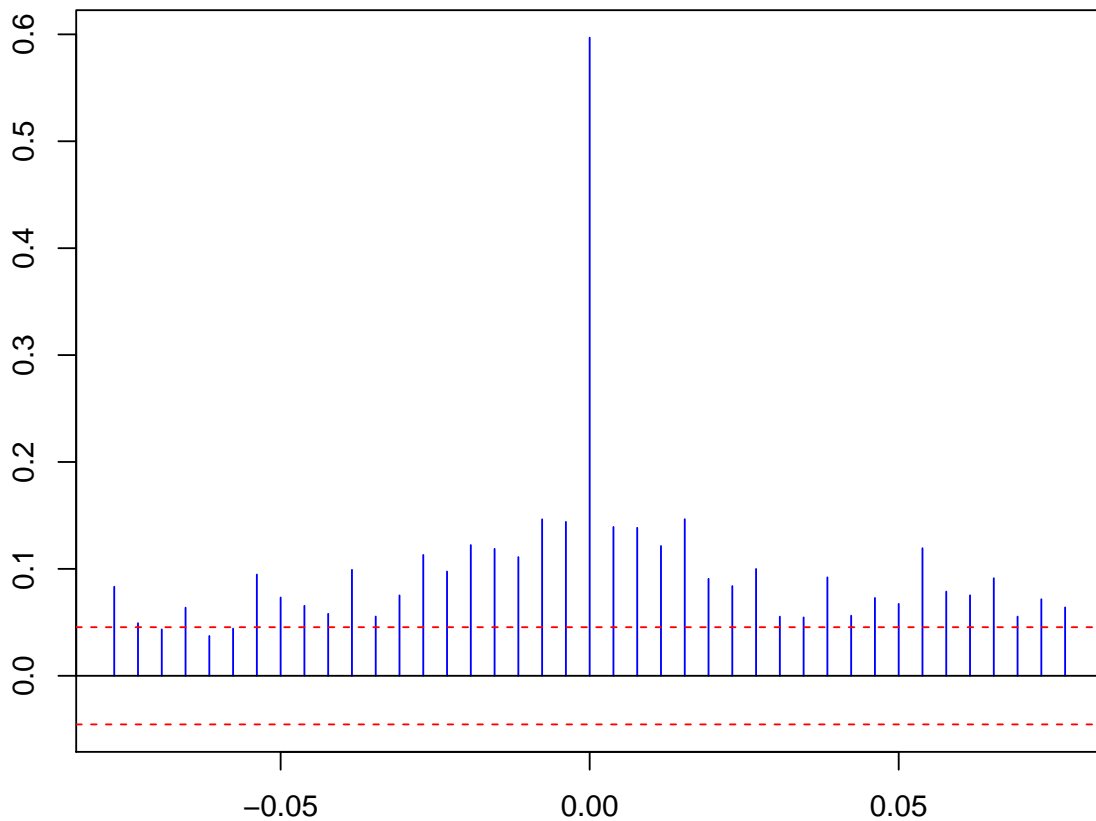
```
ccf(EuStockMarkets.return[,1], EuStockMarkets.return[,2], main = "Returns DAX vs. CAC", lag.max = 20, ylab =
```

## Returns DAX vs. CAC



```
ccf(abs(EuStockMarkets.return[,1]),abs(EuStockMarkets.return[,2]),main = "Absolute Returns DAX vs. CAC",)
```

## Absolute Returns DAX vs. CAC



We see some small raw correlations across time with raw returns. More revealing, we see volatility of correlation clustering using return sizes. We can conduct one more experiment: a rolling correlation using this function:

```
corr.rolling <-function(x) {  
  dim <-ncol(x)  
  corr.r <-cor(x)[lower.tri(diag(dim),  
                           diag =FALSE)]  
  return(corr.r)  
}
```

We then embed our rolling correlation function, `corr.rolling` into the function `rollapply`. The question we need to answer is: What is the history of correlations, and from the history, the pattern of correlation in the UK and EU stock markets? If there is a “history” with a “pattern”, then we have to manage the risk that conducting business in one country will definitely affect business in another. The implication is that bad things will be followed by more bad things more often than good things. The implication compounds a similar implication across markets.

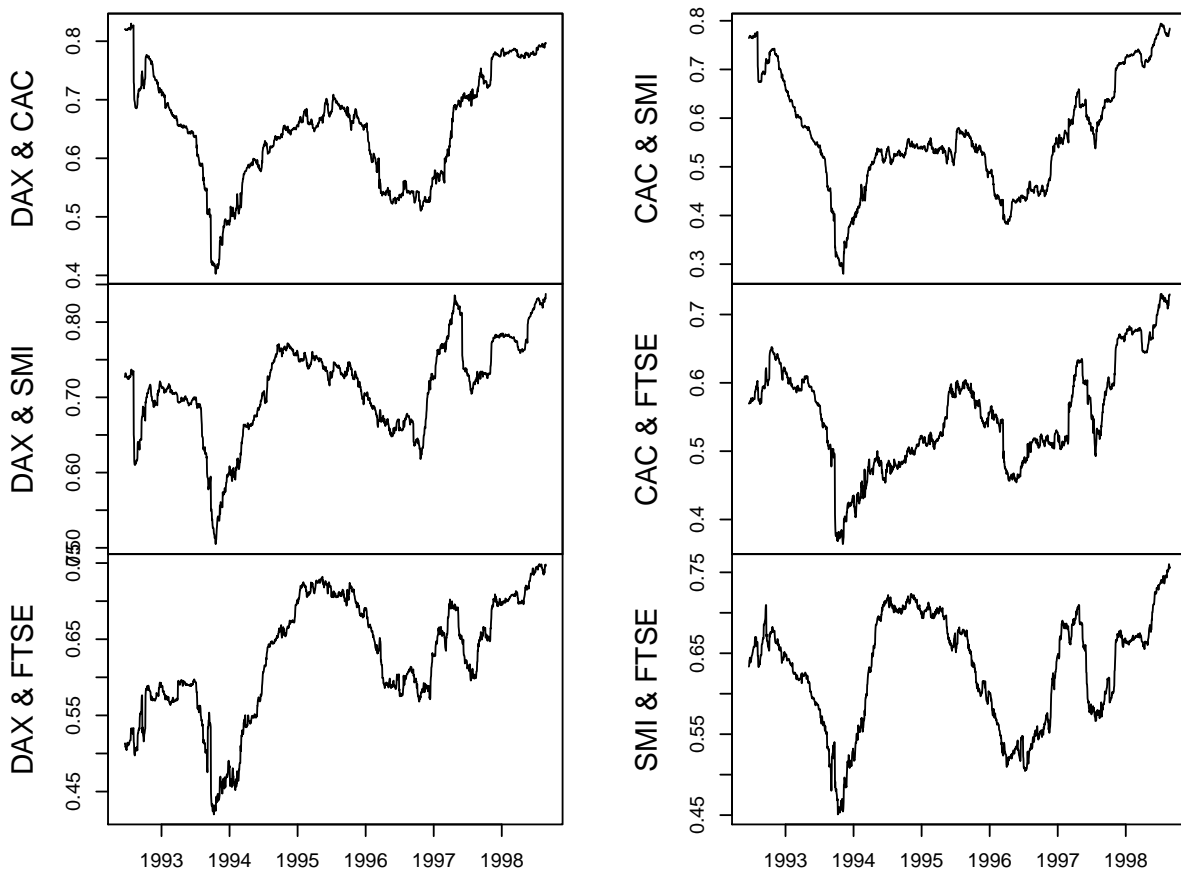
```
corr.returns <- rollapply(EuStockMarkets.return,  
                          width = 250,  
                          corr.rolling,  
                          align = "right",
```

```

                                by.column = FALSE)
colnames(corr.returns) <- c("DAX & CAC", "DAX & SMI", "DAX & FTSE"
                             , "CAC & SMI", "CAC & FTSE", "SMI & FTSE")

plot(corr.returns, xlab = "", main = "")

```



Again we observe the volatility clustering from bunching up of the absolute sizes of returns. Economic performance is certainly subject here to the same dynamics we saw for a single financial variable such as Brent.

Let's redo some of the work we just did using another set of techniques. This time we are using the "Fisher" transformation. Look up Fisher in Wikipedia and in your reference texts.

- How can the Fisher Transformation possibly help us answer our business questions?
- For three Spanish companies, Iberdrola, Endesa, and Repsol, replicate the Brent and EU stock market experiments above with absolute sizes and tails. Here we already have "series" covered.

First, the Fisher transformation is a smoothing routine that helps us stabilize the volatility of a variate. It does this by pulling some of the shockiness (i.e., outliers and aberrant noise) out of the original time series. In a phrase, it helps us see the forest (or the wood) for the trees.

We now replicate the Brent and EU stock exchange experiments. We again load some packages and get some data using `quantmod` `getSymbols` off the Madrid stock exchange to match our initial working example of Iberian companies on account. Then compute returns and merge into a master file.

```
library(xts)
library(qrmdata)
library(quantreg)
library(quantmod)
library(matrixStats)

tickers <- c("ELE.MC", "IBE.MC", "REP.MC")
getSymbols(tickers)

## [1] "ELE.MC" "IBE.MC" "REP.MC"

REP.r <- na.omit(diff(log(REP.MC[, 4]))[-1])
IBE.r <- na.omit(diff(log(IBE.MC[, 4]))[-1])
ELE.r <- na.omit(diff(log(ELE.MC[, 4]))[-1])

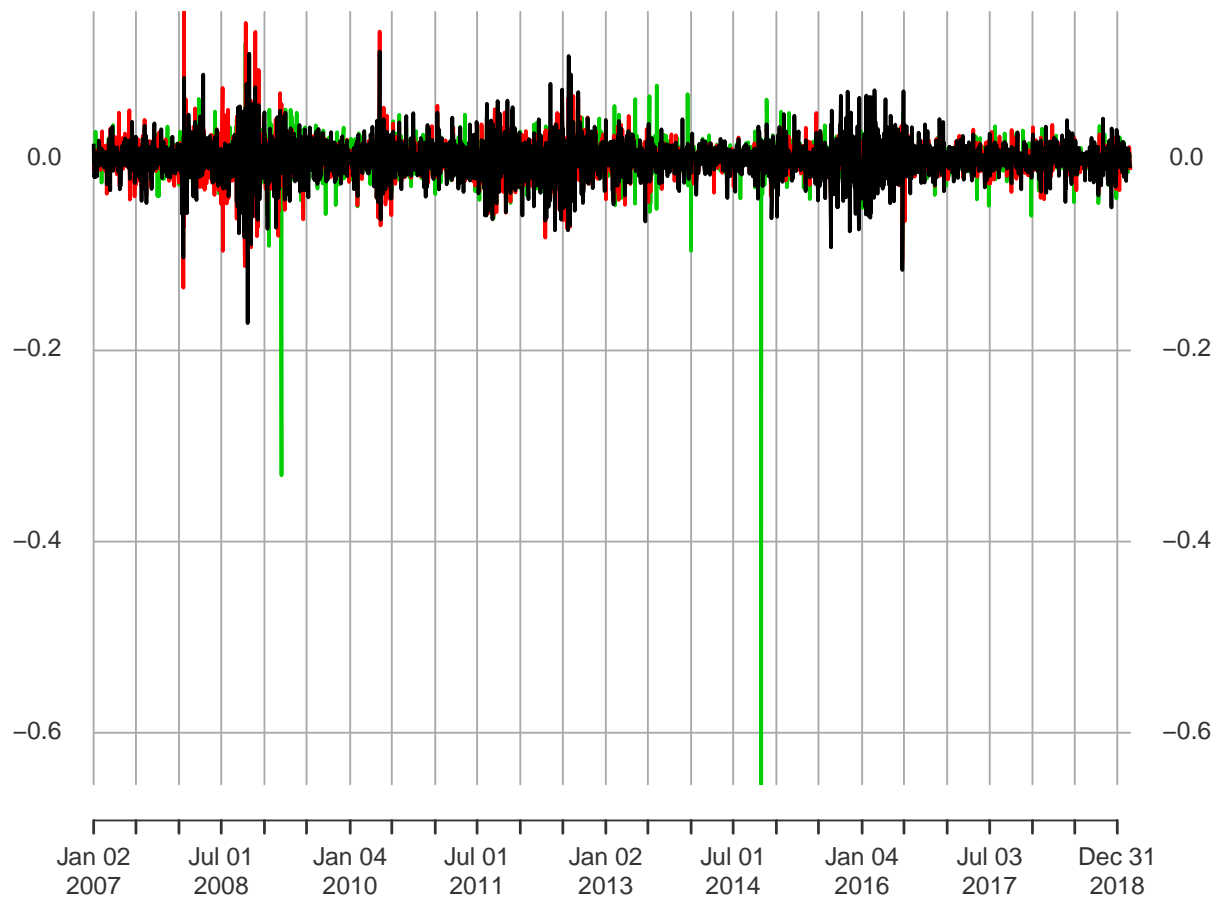
ALL.r <- na.omit(merge(REP = REP.r, IBE = IBE.r,
                      ELE = ELE.r, all = FALSE))
```

1. The persistence of returns
2. The importance of return size
3. Clustering of volatility

```
plot(ALL.r)
```

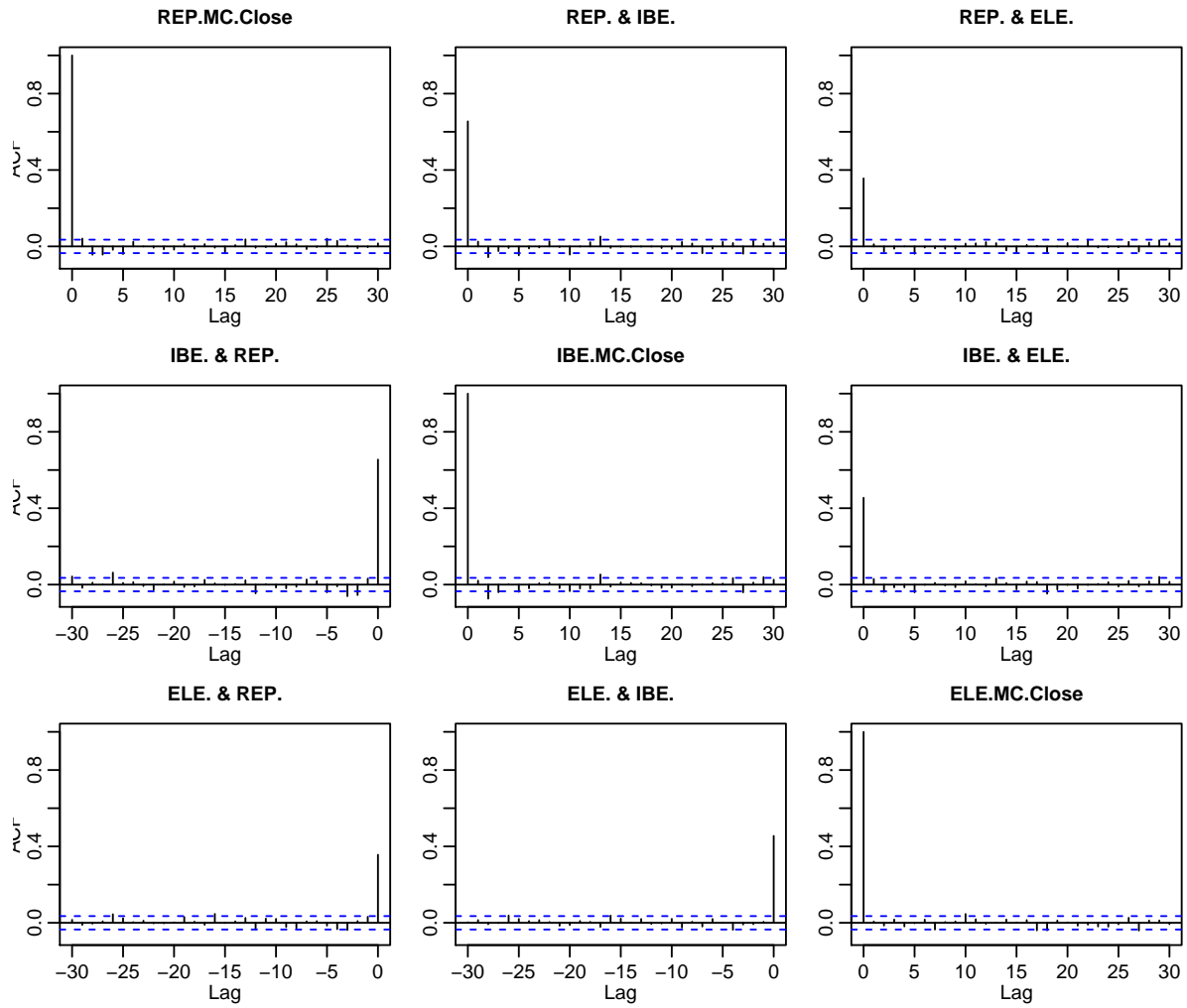
ALL.r

2007-01-02 / 2019-02-25

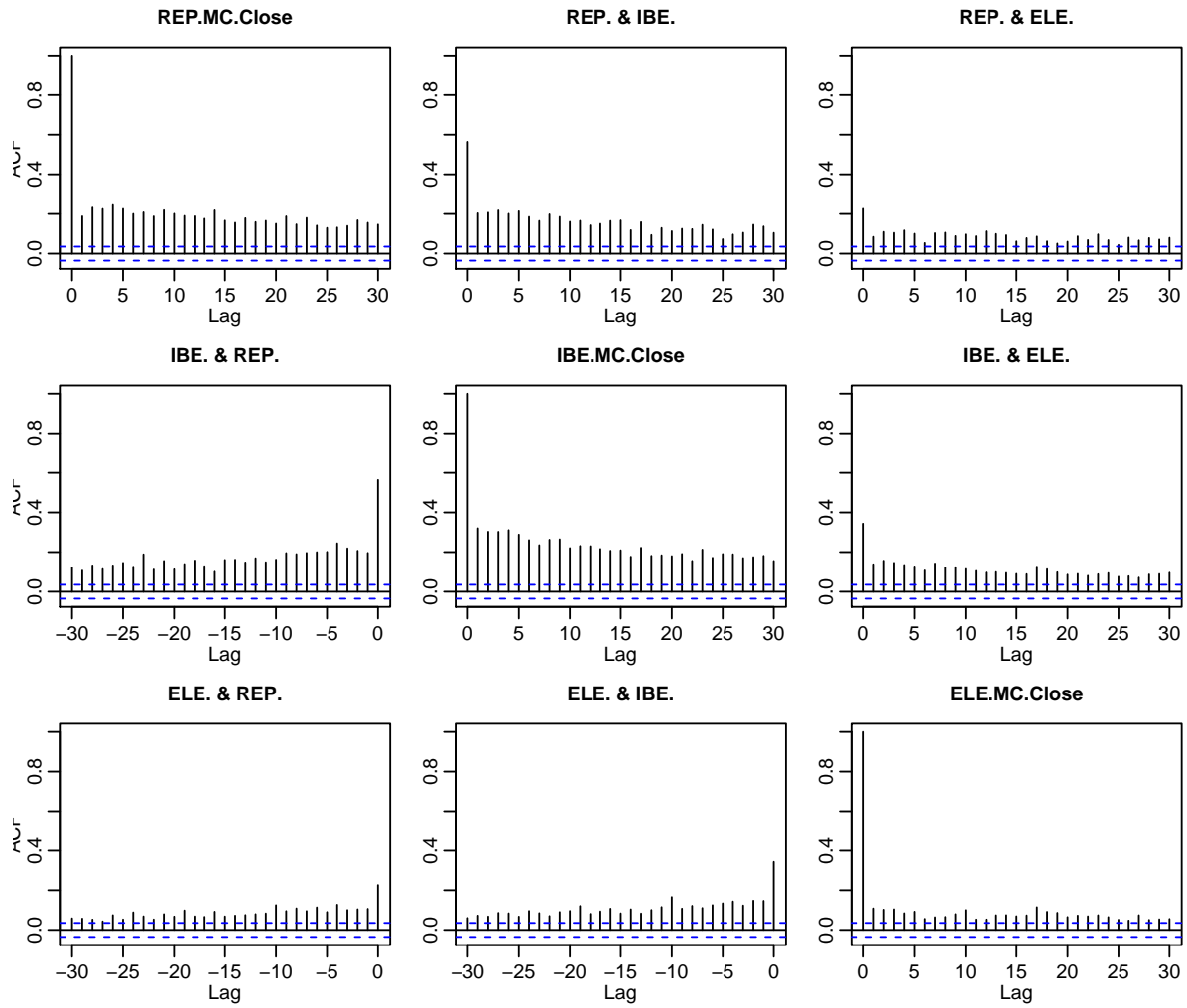


```
par(mfrow = c(2, 1))  
acf(ALL.r)
```

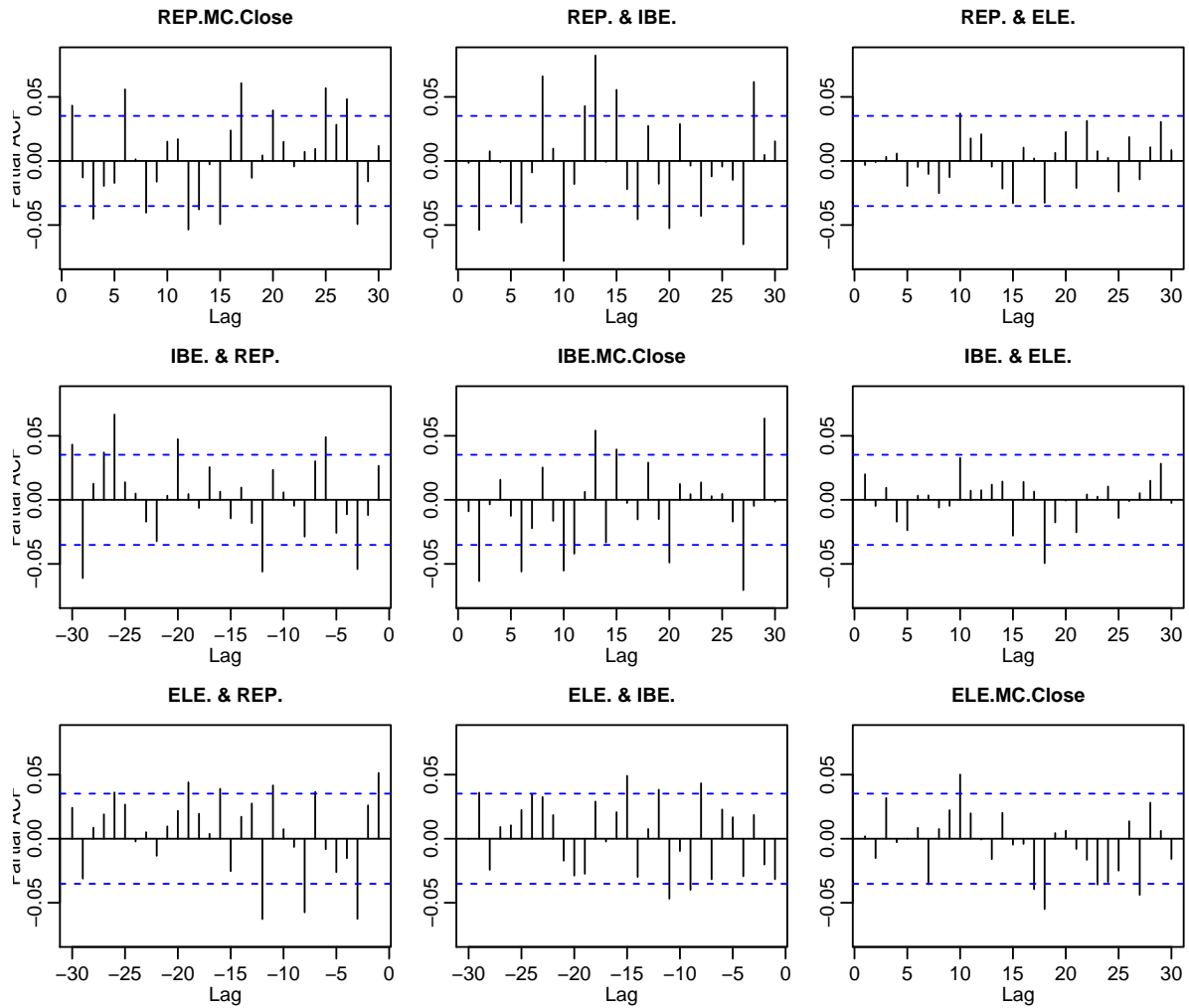




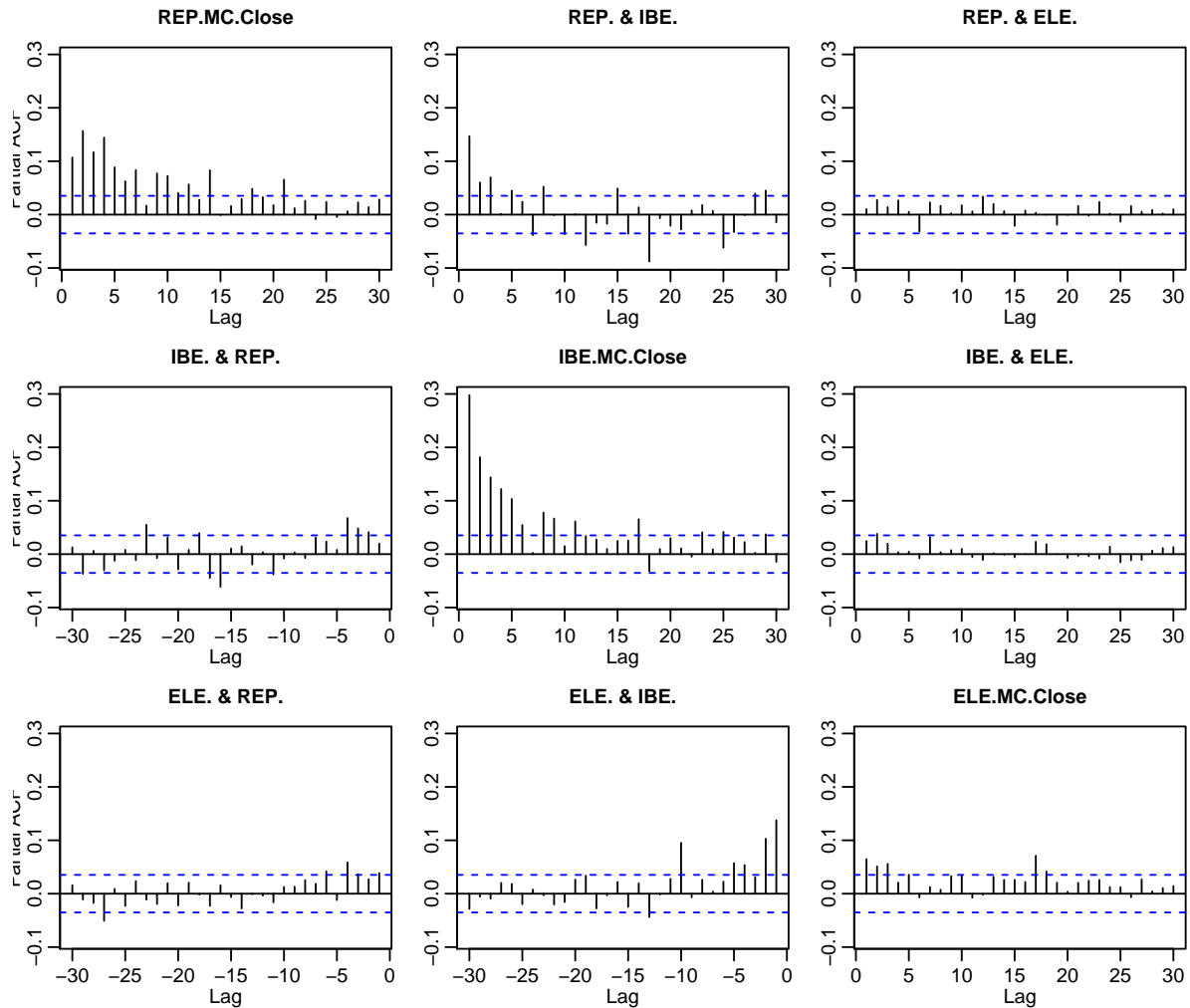
```
par(mfrow = c(2, 1))
acf(abs(ALL.r))
```



```
par(mfrow = c(2, 1))
pacf(ALL.r)
```



```
par(mfrow = c(2, 1))
pacf(abs(ALL.r))
```



Let's examine the correlation structure of markets where we can observe

1. The relationship between correlation and colatility
2. How quantile regression gets us to an understanding of high stress (high and low quantile) episodes

```
R.corr <- apply.monthly(ALL.r, FUN = cor)
R.vols <- apply.monthly(ALL.r, FUN = colSds) ## from MatrixStats
head(R.corr, 3)
```

```
##           [,1]      [,2]      [,3]      [,4] [,5]      [,6]
## 2007-01-31    1 0.3613554 -0.27540757 0.3613554    1 0.10413800
## 2007-02-28    1 0.5661814 -0.09855544 0.5661814    1 0.10760477
## 2007-03-30    1 0.4500982 -0.08874664 0.4500982    1 0.08538064
##           [,7]      [,8] [,9]
## 2007-01-31 -0.27540757 0.10413800    1
## 2007-02-28 -0.09855544 0.10760477    1
## 2007-03-30 -0.08874664 0.08538064    1
```

```
head(R.vols, 3)
```

```
##           REP.MC.Close IBE.MC.Close ELE.MC.Close
```

```
## 2007-01-31 0.009787963 0.007892759 0.009777426
## 2007-02-28 0.009181099 0.014571945 0.007674848
## 2007-03-30 0.015317331 0.012719792 0.010919155
```

```
R.corr.1 <- matrix(R.corr[1, ],
                  nrow = 3,
                  ncol = 3,
                  byrow = FALSE)
rownames(R.corr.1) <- tickers
colnames(R.corr.1) <- tickers
head(R.corr.1)
```

```
##           ELE.MC    IBE.MC    REP.MC
## ELE.MC  1.0000000 0.3613554 -0.2754076
## IBE.MC  0.3613554 1.0000000 0.1041380
## REP.MC -0.2754076 0.1041380 1.0000000
```

```
R.corr <- R.corr[, c(2, 3, 6)]
colnames(R.corr) <- c("ELE.IBE", "ELE.REP",
                     "IBE.REP")
colnames(R.vols) <- c("ELE.vols", "IBE.vols",
                     "REP.vols")
head(R.corr, 3)
```

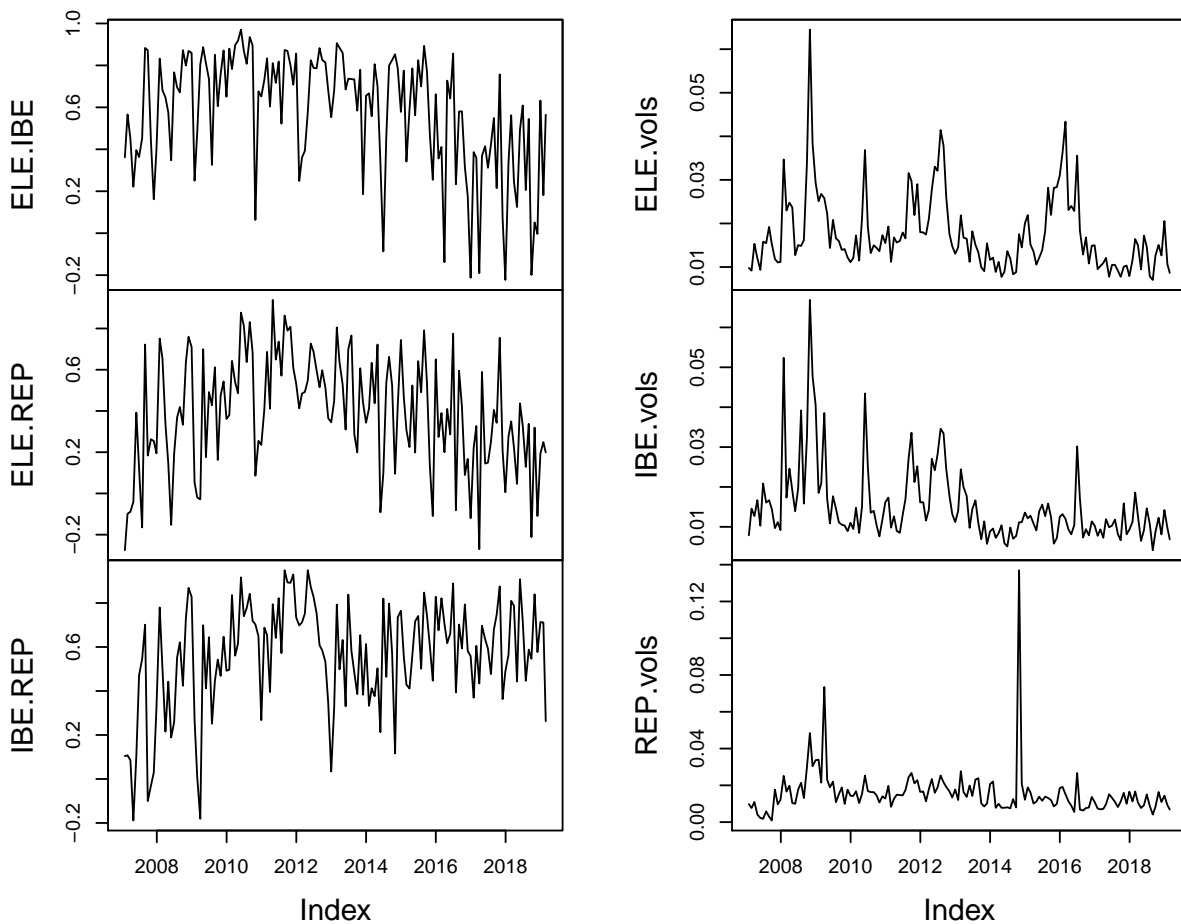
```
##           ELE.IBE    ELE.REP    IBE.REP
## 2007-01-31 0.3613554 -0.27540757 0.10413800
## 2007-02-28 0.5661814 -0.09855544 0.10760477
## 2007-03-30 0.4500982 -0.08874664 0.08538064
```

```
head(R.vols, 3)
```

```
##           ELE.vols    IBE.vols    REP.vols
## 2007-01-31 0.009787963 0.007892759 0.009777426
## 2007-02-28 0.009181099 0.014571945 0.007674848
## 2007-03-30 0.015317331 0.012719792 0.010919155
```

```
R.corr.vols <- merge(R.corr, R.vols)
plot.zoo(merge(R.corr.vols))
```

merge(R.corr.vols)



```
ELE.vols <- as.numeric(R.corr.vols[, "ELE.vols"])
```

```
IBE.vols <- as.numeric(R.vols[, "IBE.vols"])
```

```
REP.vols <- as.numeric(R.vols[, "REP.vols"])
```

```
length(ELE.vols)
```

```
## [1] 146
```

```
fisher <- function(r) {  
  0.5*log((1+r)/(1-r))  
}
```

```
rho.fisher <- matrix(fisher(as.numeric(R.corr.vols[,  
  1:3])),  
  nrow =length(ELE.vols),  
  ncol =3,  
  byrow =FALSE)
```

### 2.3.1 On to quantiles

Here is the quantile regression part of the package. Quantile regression finds the average relationship between dependent and independent variables just like ordinary least squares with one exception. Instead of centering the regression on the arithmetic mean of the dependent variable, quantile regression centers the regression on the specified quantile of the dependent variable. So instead of using the arithmetic average of the rolling correlations, we now use the 10th quantile, or the median, which is the 50th quantile as our reference. This makes great intuitive sense since we have already established that the series we deal with here are thick tailed, skewed, and certainly not normally distributed.

Here is how we use the `quantreg` package.

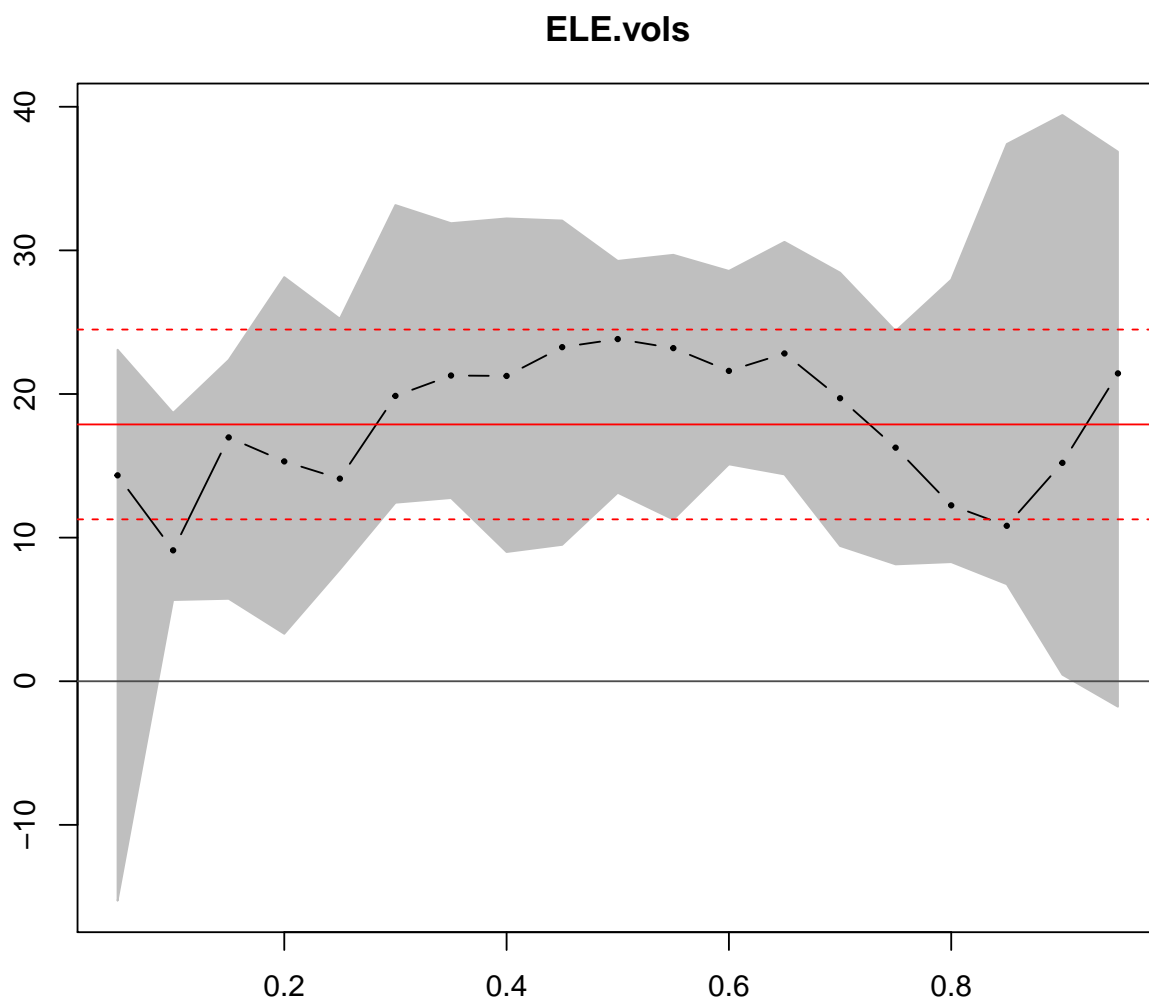
1. We set `taus` as the quantile of interest.
2. We run the quantile regression using the `quantreg` package and a call to the `rq` function.
3. We can overlay the quantile regression results onto the standard linear model regression.
4. We can sensitize our analysis with the range of upper and lower bounds on the parameter estimates of the relationship between correlation and volatility. This sensitivity analysis is really a confidence interval based on quantile regression.

```
library(quantreg)
taus <- seq(0.05, 0.95, 0.05)

fit.rq.ELE.IBE <- rq(rho.fisher[, 1] ~ ELE.vols, tau = taus)

fit.lm.ELE.IBE <- lm(rho.fisher[, 1] ~ ELE.vols)

plot(summary(fit.rq.ELE.IBE), parm = "ELE.vols")
```



Here we build the estimations and plot the upper and lower bounds.

```
taus1 <- c(0.05, 0.95) ## fit the confidence interval (CI)

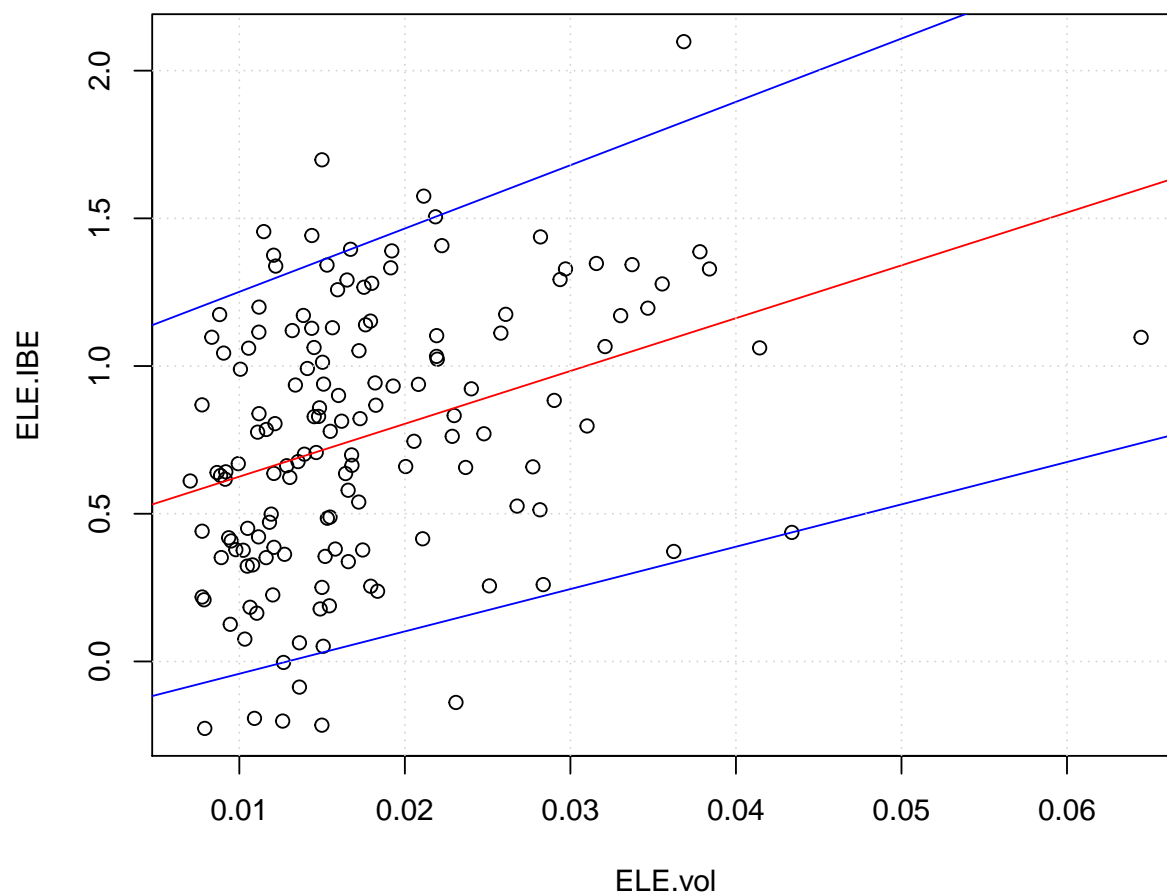
plot(ELE.vols, rho.fisher[, 1], xlab = "ELE.vol", ylab = "ELE.IBE")

abline(fit.lm.ELE.IBE, col = "red")

for (i in 1:length(taus1)){
  # these will be the CI
  abline(rq(rho.fisher[, 1] ~ ELE.vols,
            tau = taus1[i]), col = "blue")
}

grid()
```





Quantile regression helps us to see the upper and lower bounds. Relationships between high-stress periods and correlation are abundant. These markets simply reflect normal buying behaviors across many types of exchanges: buying food at Safeway or Whole Foods, buying collateral to insure a project, selling of illiquid assets.

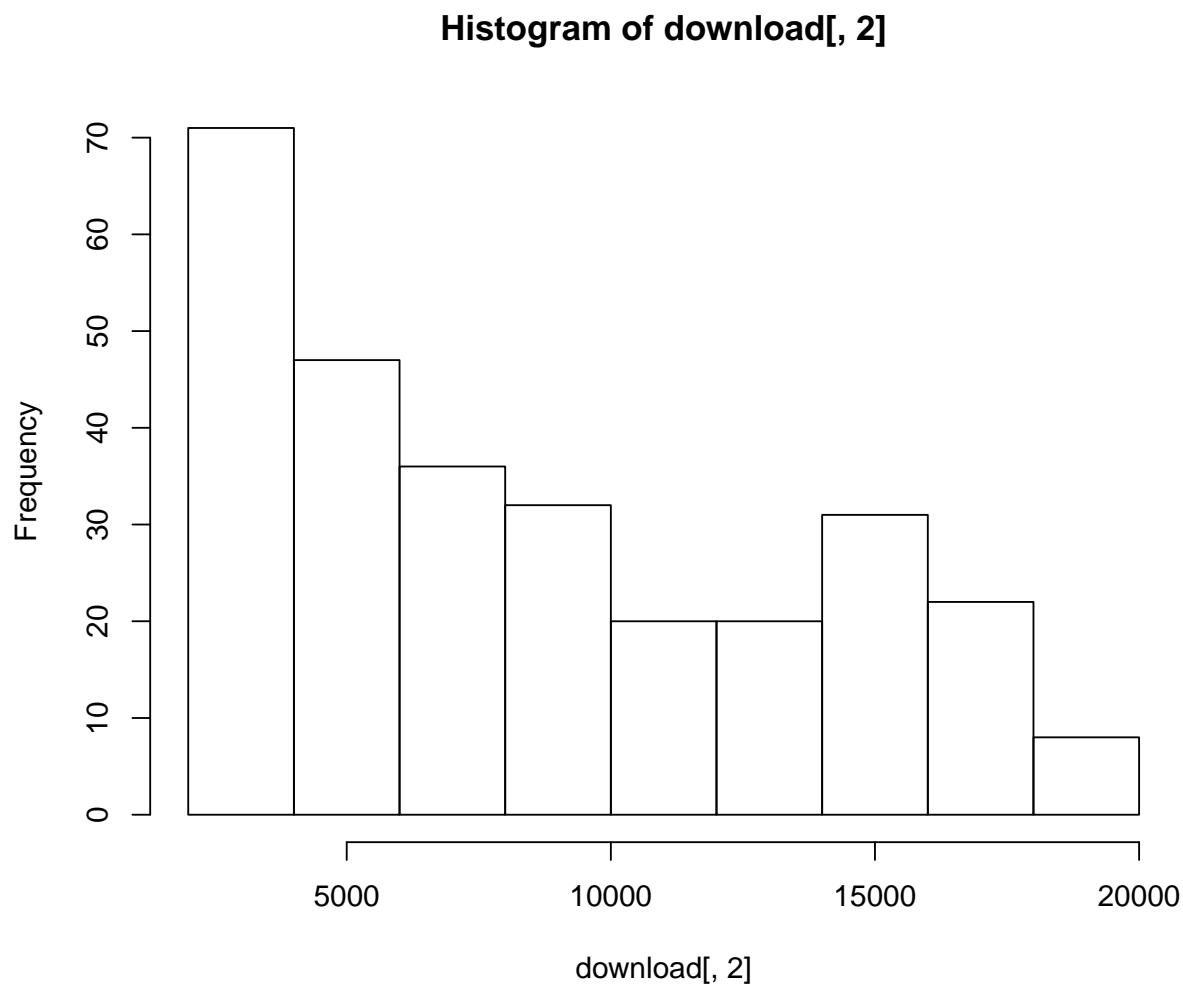
## 2.4 Time is on our Side

Off to another important variable, the level and growth rate of Gross National Product. Let's start with some US Cross National Product (GNP) data from the St.Louis Fed's open data website ("FRED").

```
name <- "GNP"
download <- read.csv("data/GNPC96.csv")
```

look at the data:

```
hist(download[, 2])
```



```
summary(download[, 2])
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      2037   4051    7026   8574   13246   18908
```

We then create a raw time series object using `ts` function where rownames are dates, select some data, and calculate growth rates. This will allow us to use plotting functions to index the dates in the data. Again, we make use of the `diff(log(data))` vector calculations.

```
GNP <- ts(download[1:85, 2], start = c(1995,1), freq = 4)
GNP.rate <- 100* diff(log(GNP))
str(GNP)
```

```
## Time-Series [1:85] from 1995 to 2016: 2046 2041 2037 2069 2101 ...
```

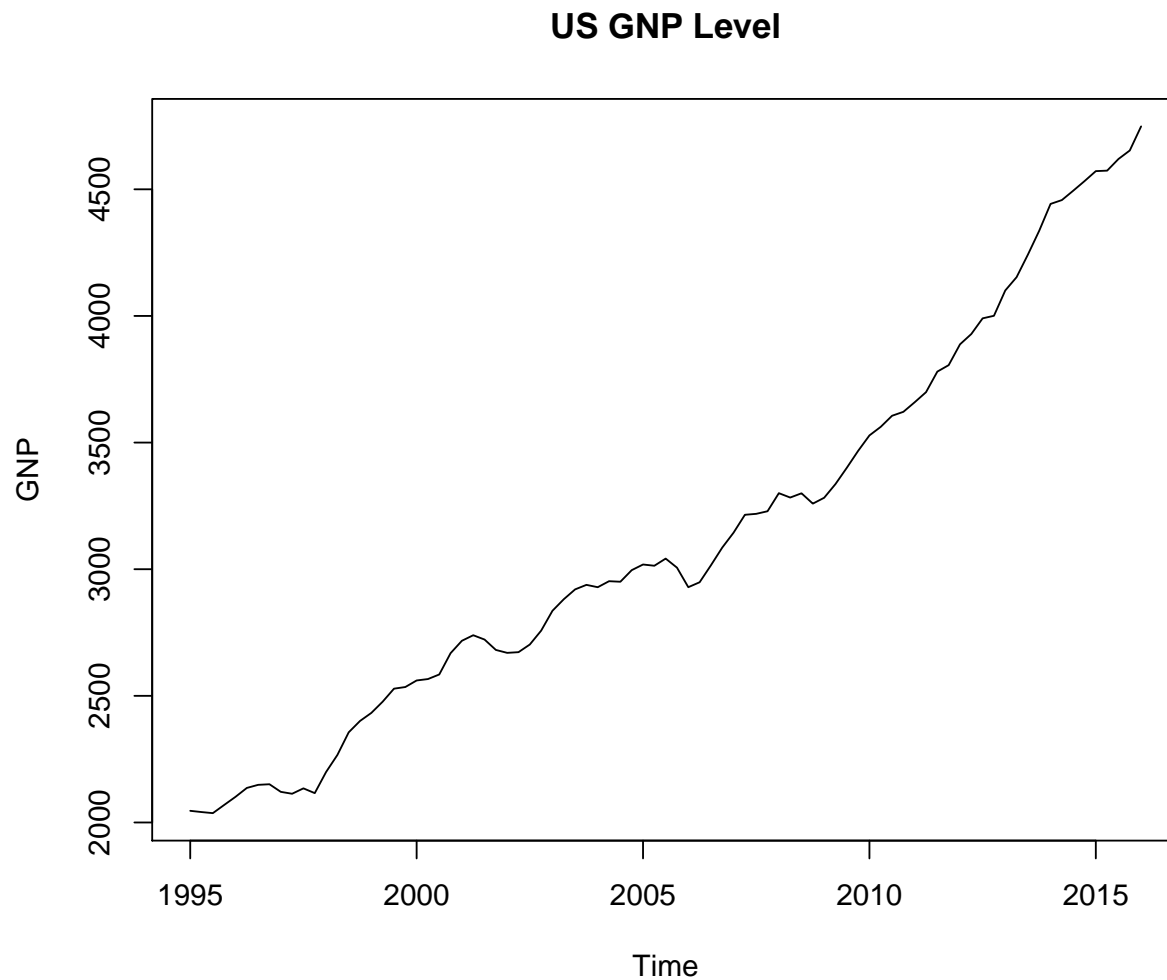
```
head(GNP)
```

```
## [1] 2046.020 2041.114 2036.864 2069.176 2101.218 2136.364
```

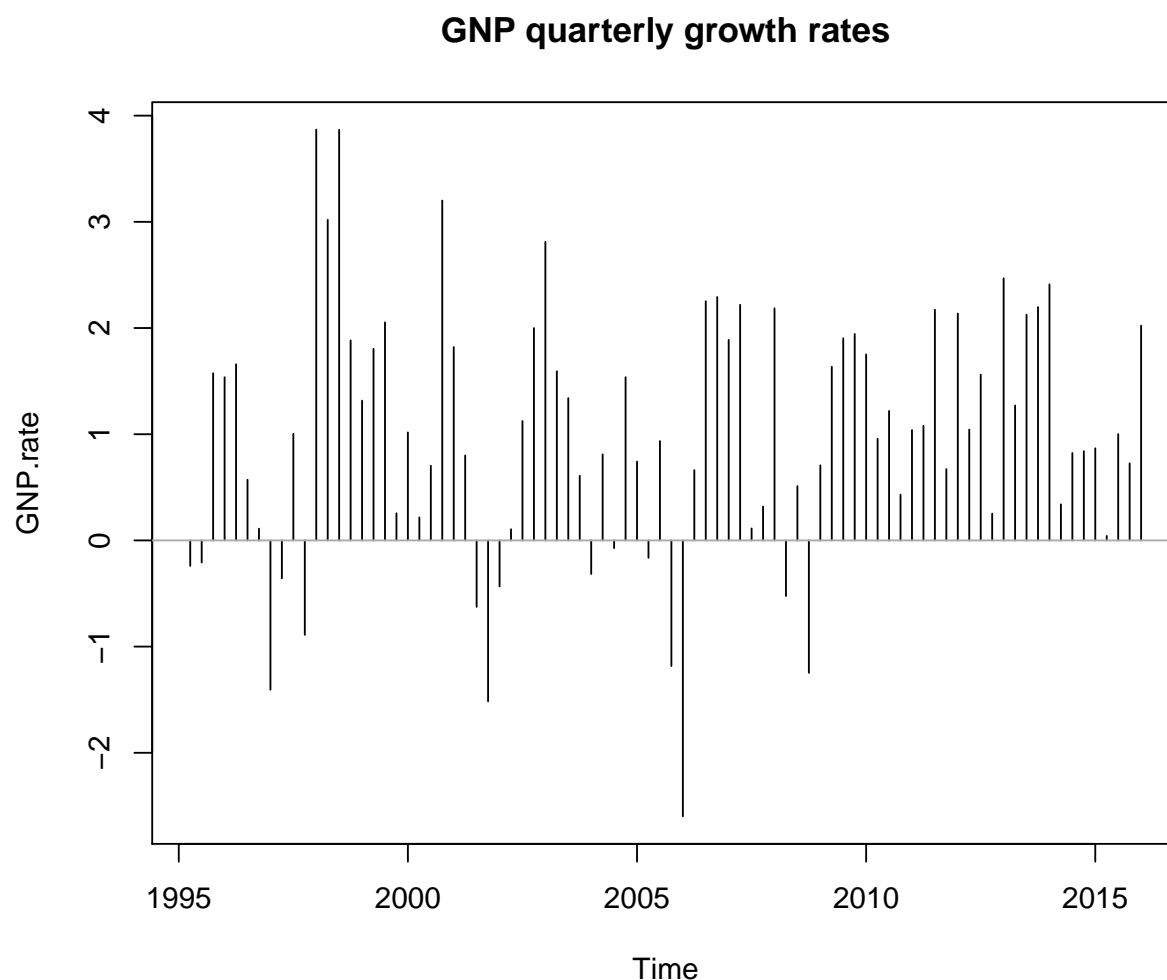
```
head(GNP.rate)
```

```
## [1] -0.2400705 -0.2084367 1.5739090 1.5366716 1.6588142 0.5711614
```

```
plot(GNP, main = "US GNP Level")
```



```
plot(GNP.rate, type = "h", main = "GNP quarterly growth rates")  
abline(h = 0, col = "darkgray")
```



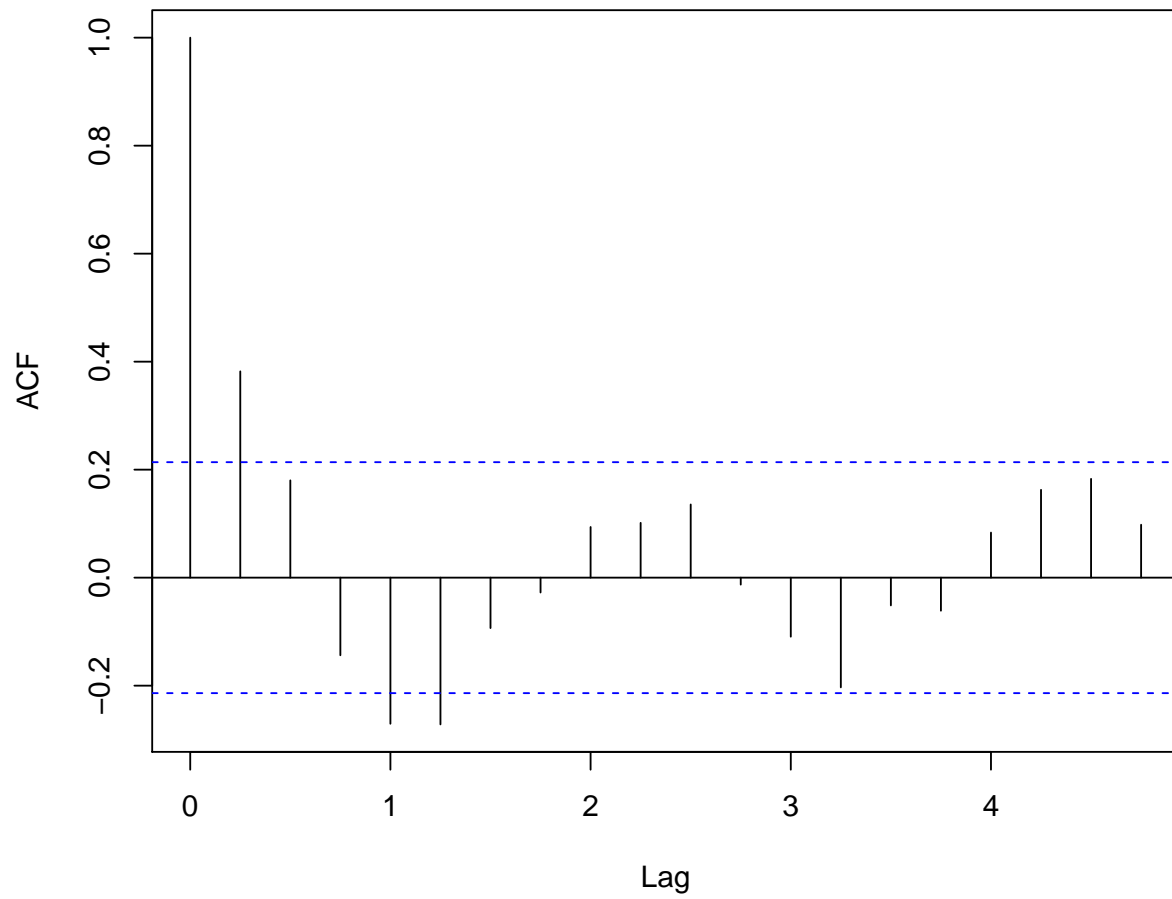
We see a phenomenon called “nonstationarity”. The probability distribution (think `hist()`) would seem to change over time (many versions of a `hist()`). This means that the standard deviation and mean change as well (and higher moments such as skewness and kurtosis). There is trend in the level and simply dampened sinusoidal in the rate. In a nutshell we observe several distributions mixed together in this series. This will occur again in the term structure of interest rates where we will use splines and their knots to get at parameterizing the various distributions lurking just beneath the ebb and flow of the data.

#### 2.4.1 Forecasting GNP

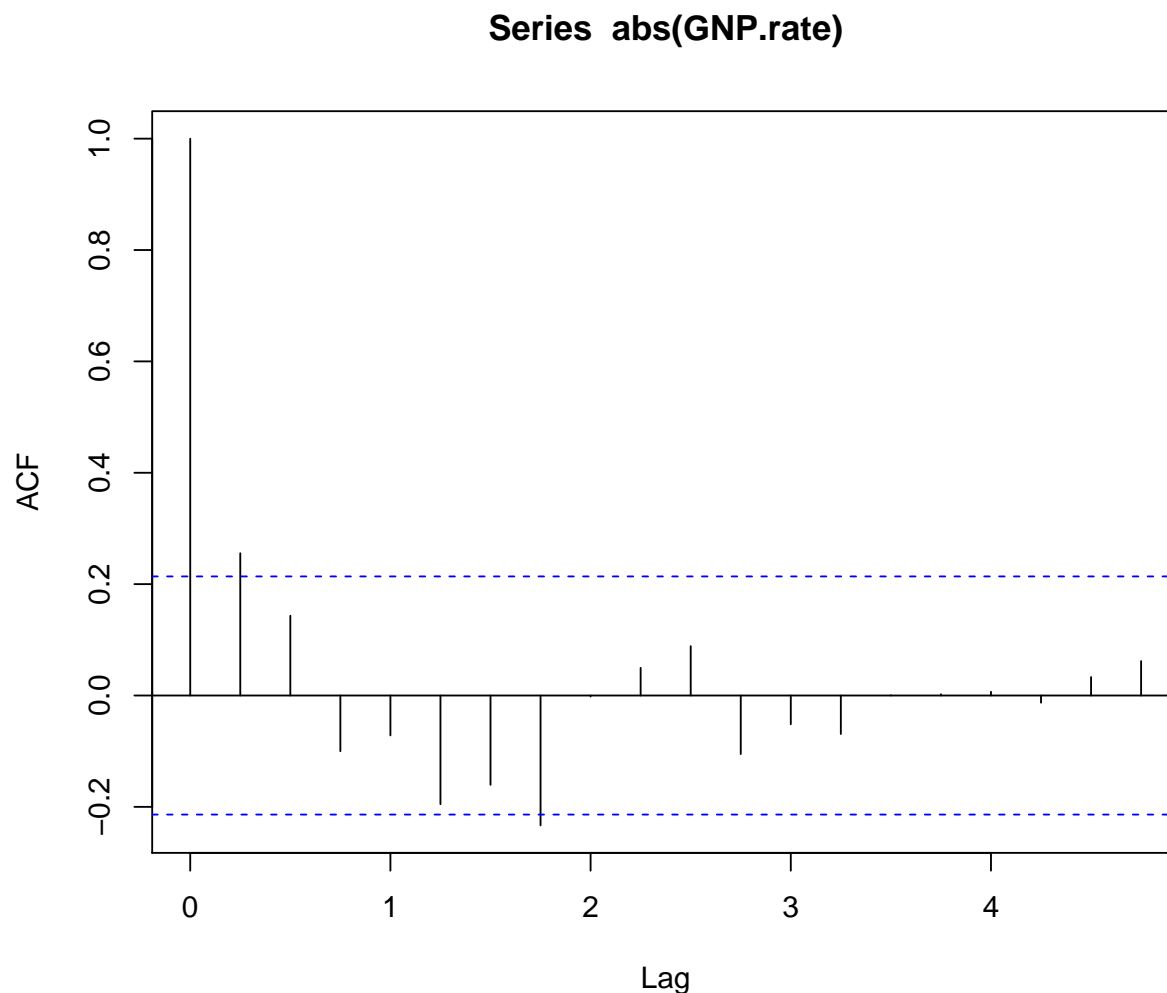
As always let’s look at ACF and PACF:

```
par(mfrow = c(2, 1))  
acf(GNP.rate)
```

### Series GNP.rate



```
acf(abs(GNP.rate))
```



```
par(mfrow = c(1, 1)) # default setting
```

What do we think is going on? There are several significant autocorrelations within the last 4 quarters. Partial autocorrelation also indicates some possible relationship 8 quarters back.

Let's use R's time series estimation and prediction tool **arima**. In this world we think there is a regression that looks like this:

$$x_t = \alpha_0 + \alpha_1 x_{t-1} + \dots + \alpha_p x_{t-p} + b_1 \epsilon_{t-1} + \dots + b_q \epsilon_{t-q}$$

where  $x_t$  is a first,  $d = 1$ , differenced level of a variable, here GNP. There are  $p$  lags of the rate itself and  $q$  lags of residuals. We officially call this an **Autoregressive Integrated Moving Average** process of order  $(p, d, q)$ , or **ARIMA**( $p, d, q$ ) for short.

Estimation is quick and easy.

```
fit.rate <- arima(GNP.rate, order = c(2, 1, 1))
```

The order is 2 lags of rate, 1 further difference (already differenced once when we calculated `diff(log(GNP))`), and 1 lag of residuals. Let's diagnose the results with `tsdiag()`. What are the results?

```
fit.rate
```

```
##
## Call:
## arima(x = GNP.rate, order = c(2, 1, 1))
##
## Coefficients:
##          ar1      ar2      ma1
##      0.3829  0.0542 -1.0000
## s.e.  0.1106  0.1109   0.0381
##
## sigma^2 estimated as 1.202:  log likelihood = -127.16,  aic = 262.31
```

Let's take out the moving average term and compare:

```
fit.rate.2 <- arima(GNP.rate, order = c(2, 0, 0))
fit.rate.2
```

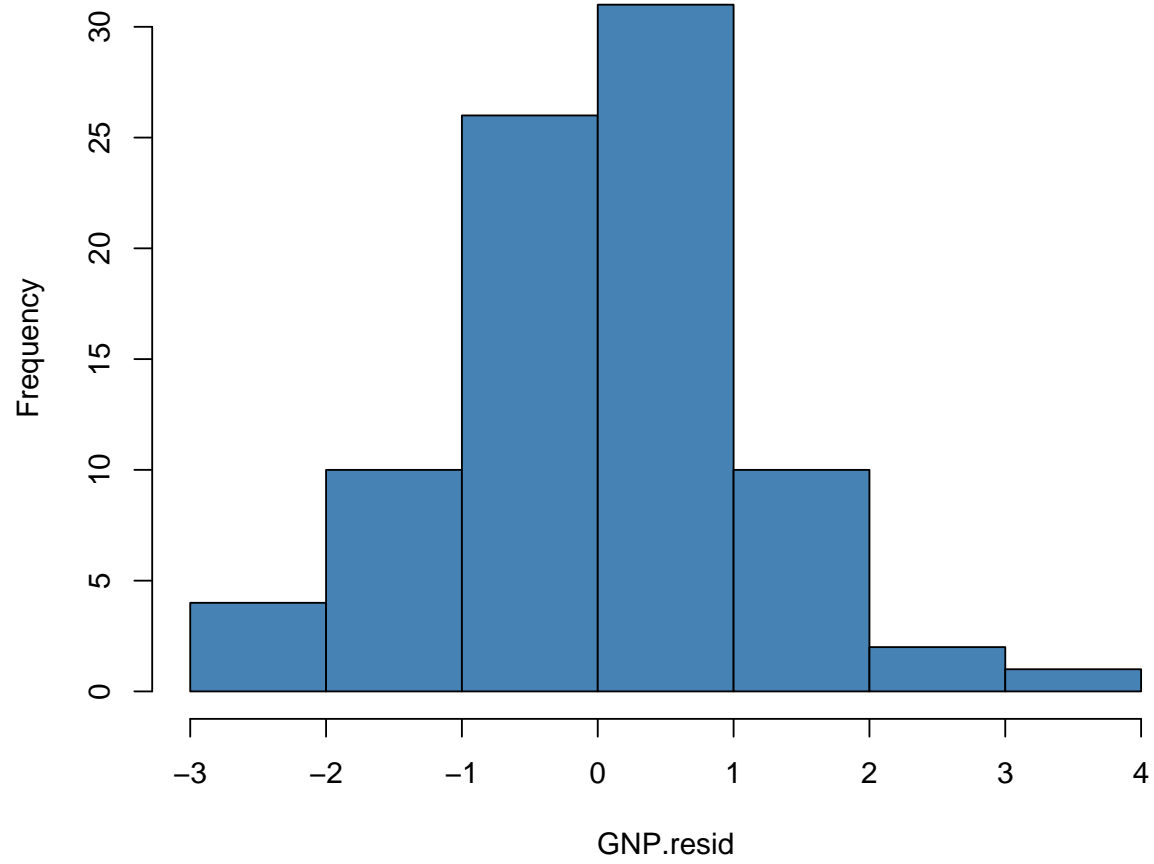
```
##
## Call:
## arima(x = GNP.rate, order = c(2, 0, 0))
##
## Coefficients:
##          ar1      ar2  intercept
##      0.3700  0.0415    0.9992
## s.e.  0.1092  0.1095    0.2002
##
## sigma^2 estimated as 1.188:  log likelihood = -126.49,  aic = 260.98
```

```
GNP.resid <- residuals(fit.rate.2)
```

The qqnorm function plots actual quantiles against theoretical normal distributions of the quantiles. A line through the scatterplot will reveal deviations of actual quantiles from the normal ones. Those deviations are the key to understanding tail behavior, and thus the potential influence of outliers, on our understanding of the data.

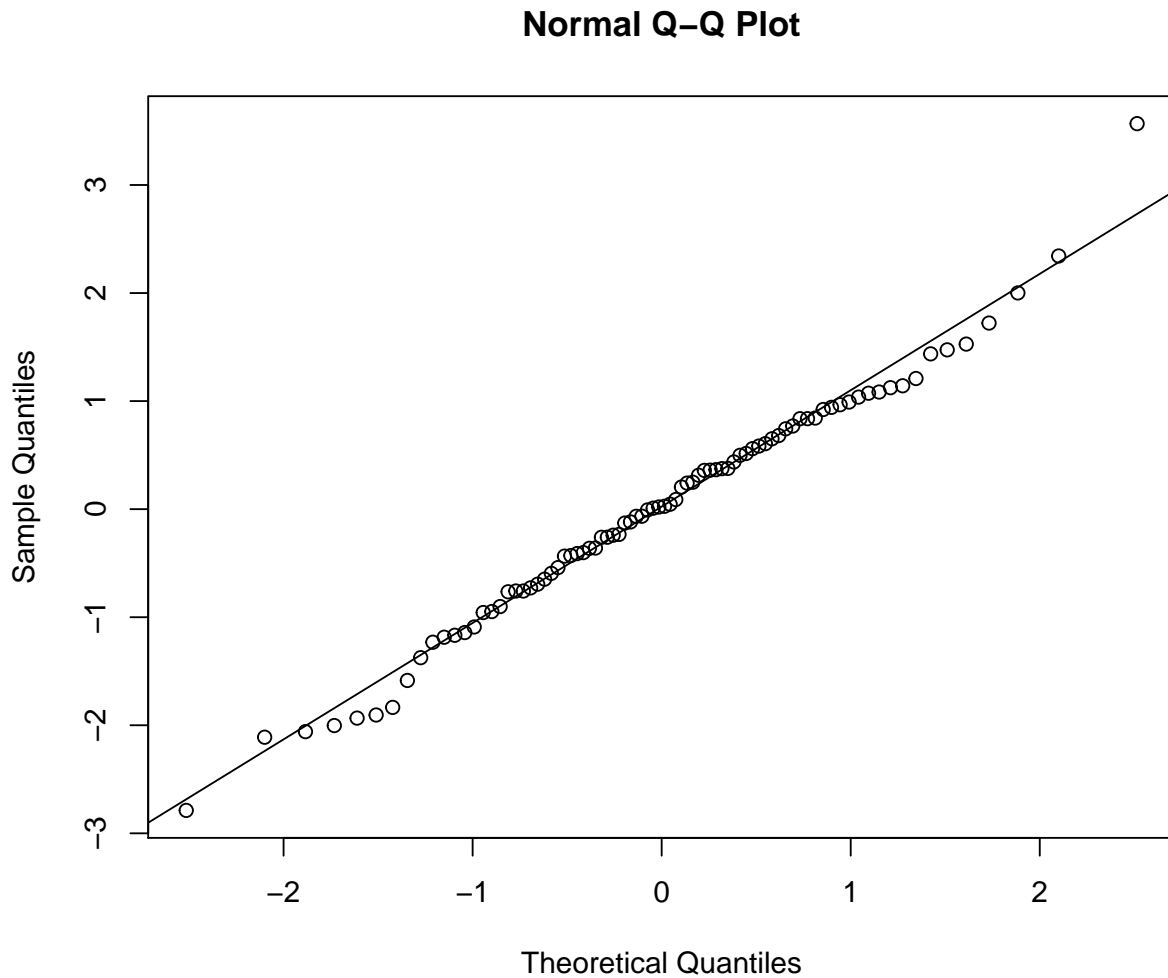
```
GNP.resid <- fit.rate.2$residuals
hist(GNP.resid, col = "steelblue")
```

**Histogram of GNP.resid**



```
qqnorm(GNP.resid)  
qqline(GNP.resid)
```





Some ways to interpret the qq-chart include

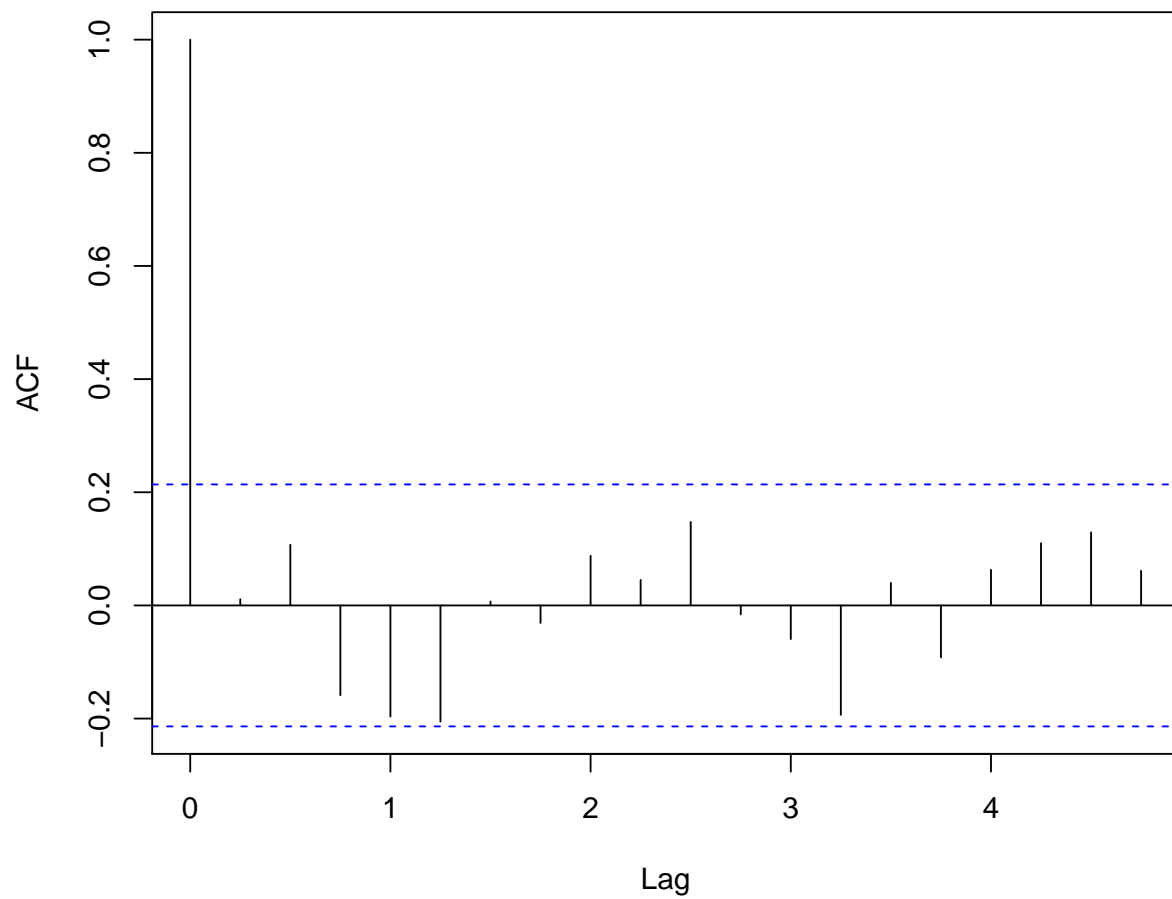
1. The diagonal line is the normal distribution quantile line.
2. Deviations of actual quantiles from the normal quantile line mean nonnormal.
3. Especially deviations at either (or both) end of the line spell thick tails and lots more “shape” than the normal distribution allows.

#### 2.4.2 Residual again

How can we begin to diagnose the GNP residuals? Let’s use the ACF and the `moments` package to calculate **skewness** and **kurtosis**. We find that the series is very thick tailed and serially correlated as evidenced by the usual statistical suspects. But no volatility clustering.

```
acf(GNP.resid)
```

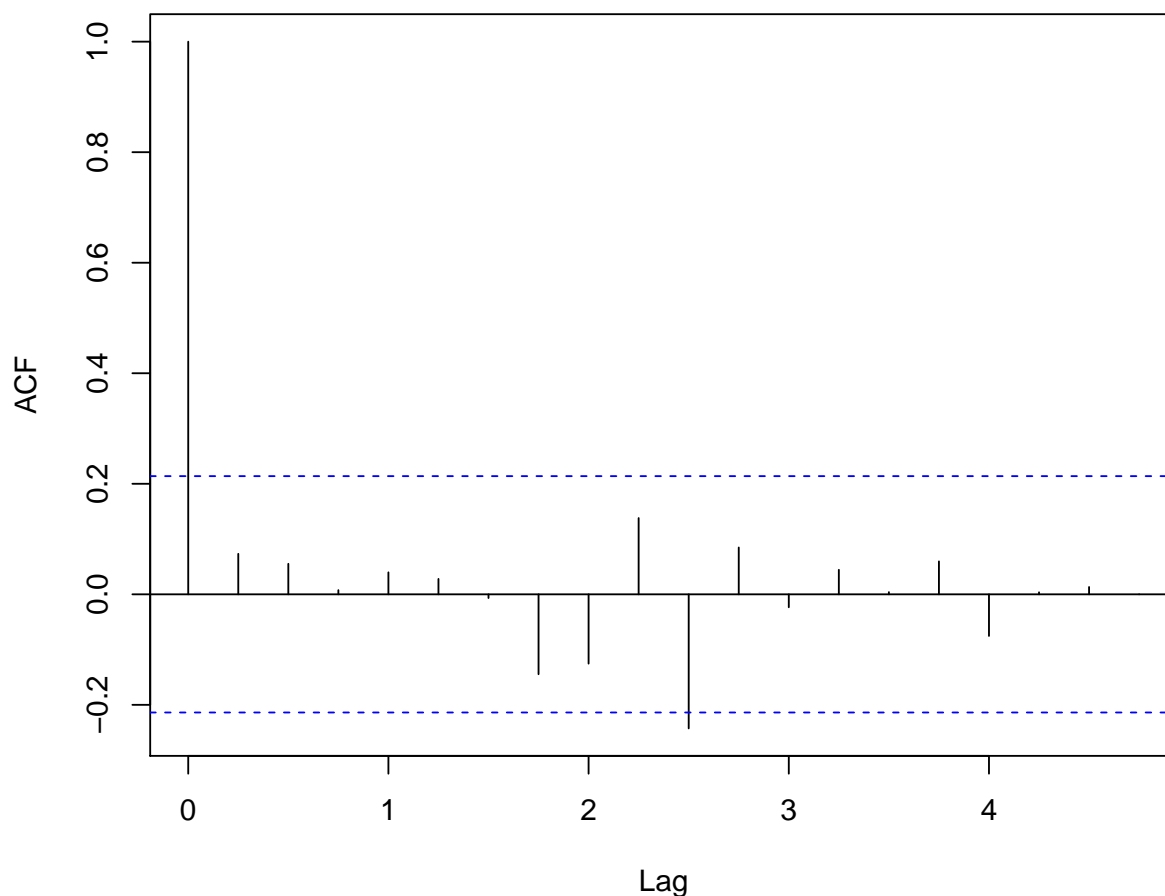
### Series GNP.resid



Now let's look at the absolute values of growth (i.e., GNP growth sizes). This will help us understand the time series aspects of the volatility of the GNP residuals

```
acf(abs(GNP.resid))
```

### Series abs(GNP.resid)



...and compute tail statistics.

```
library(moments)
skewness(GNP.resid)
```

```
## [1] 0.06218325
```

```
kurtosis(GNP.resid)
```

```
## [1] 3.574021
```

The residuals are positively skewed and not so thick tailed, as the normal distribution has by definition a kurtosis equal to 3.00. Now for the forecast.

```
(GNP.pred <- predict(fit.rate, n.ahead = 8))
```

```
## $pred
```

```
##      Qtr1      Qtr2      Qtr3      Qtr4
```

```
## 2016      1.375848 1.198497 1.095574
```

```
## 2017 1.046558 1.022214 1.010238 1.004334
```

```
## 2018 1.001425
```

```
##
```

```
## $se
```

```
##           Qtr1      Qtr2      Qtr3      Qtr4
## 2016           1.102806 1.185428 1.209155
## 2017 1.215722 1.217821 1.218576 1.218880
## 2018 1.219014
```

Now for something really interesting, yet another rendering of the notorious Efficient Market Hypothesis.

## 2.5 Give it the Boot

Our goal is to infer the significance of a statistical relationship among variates. However, we do not have access to, or a “formula” does not exist, that allows us to compute the sample standard deviation of the mean estimator.

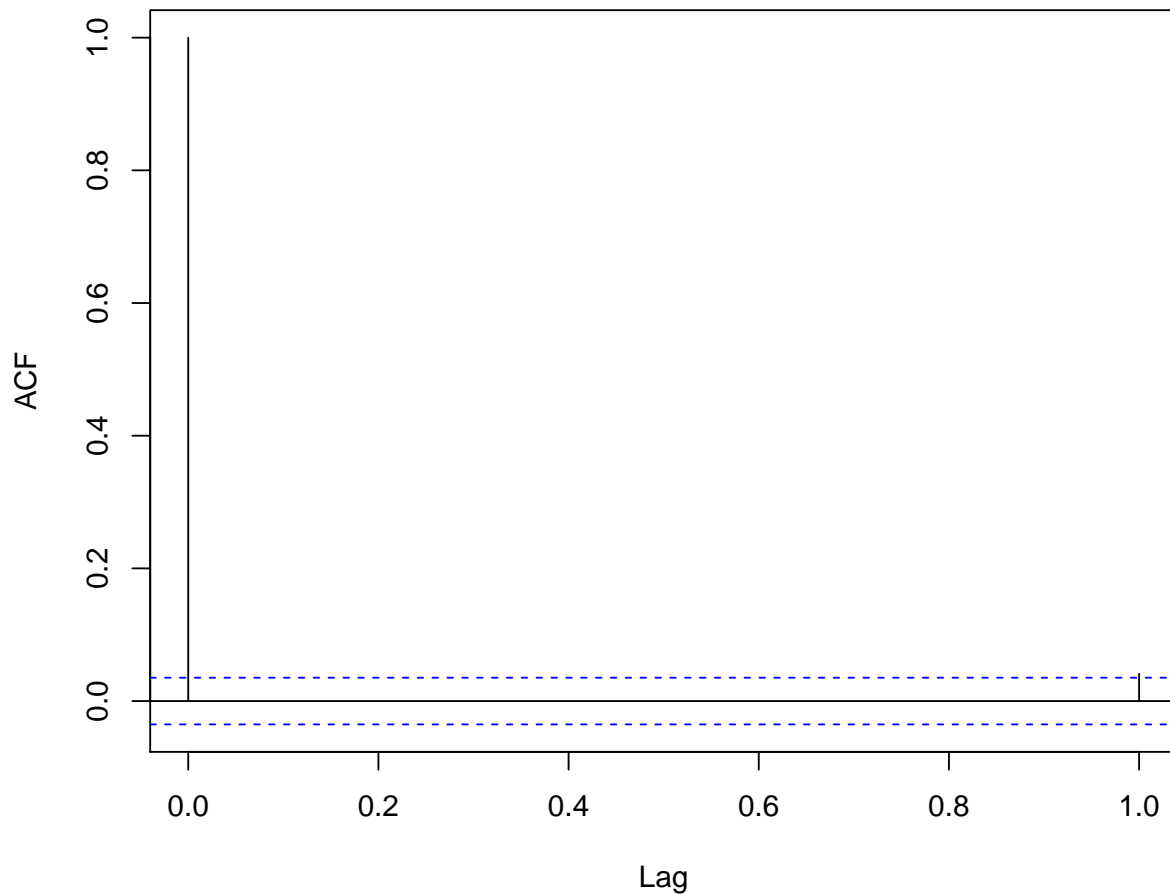
- The context is just how dependent is today’s stock return on yesterday’s?
- We want to use the distribution of real-world return data, without needing assumptions about normality.
- The null hypothesis  $H_0$  is lack of dependence (o.e., an efficient market). The alternative hypothesis  $H_1$  is that today’s returns depends on past returns, on average.

Our strategy is to change the data repeatedly, and re-estimate a relationship. The data is sampled using the `replicate` function, and the sample ACF is computed. This gives us the distribution of the coefficient of the ACF under the null hypotheses,  $H_0$ : independence, while using the empirical distribution of the returns data.

Let’s use the Repsol returns and pull the 1st auto correlation from the sample with this simple code,

```
acf(REP.r, 1)
```

## Series REP.r



There is not much for us to see, barely a blip, but there is a correlation over the 95% line. Let's further test this idea.

- We obtain 2500 draws from the distribution of the first autocorrelation using the `replicate` function.
- We operate under the null hypothesis of independence, assuming rational markets (i.e, rational markets is a “maintained hypothesis”)

```
set.seed(1016)

# Sample has 3100 rows, we want to sample 2500 without replacement
nrow(REP.r)

## [1] 3103

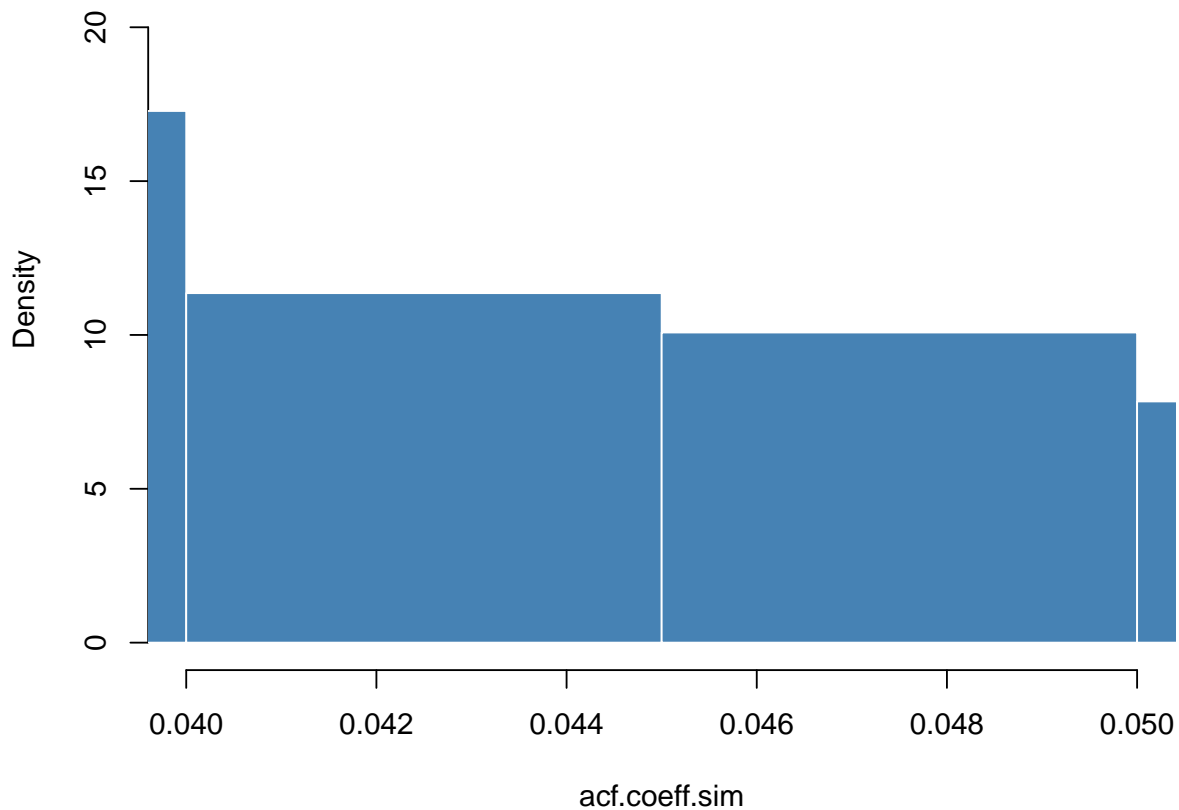
# Replicate
acf.coeff.sim <- replicate(2500, acf(sample(REP.r,
                                           size = 2500, replace = FALSE), lag = 2,
                                           plot = FALSE)$acf[2])

# Summary
summary(acf.coeff.sim)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.04374 0.01179 0.02466 0.02478 0.03705 0.09441
```

```
hist(acf.coeff.sim, probability = TRUE,
     breaks = "FD", xlim = c(0.04, 0.05),
     col = "steelblue", border = "white"
    )
```

**Histogram of acf.coeff.sim**



## 2.6 Summary

We explored time series data using ACF, PACF, and CCF. We characterized several stylized facts of financial returns and inferred behaviour using the rolling correlation regression on volatility. We then supplemented the ordinary least square regression confidence intervals using the entire distribution of the data with quantile regression. We also built using bootstrapping techniques we simulated coefficient inference to check the efficient market hypothesis. This along with the quantile regression technique, allows us to examine risk tolerance from an inference point of view.

### 3 Term Structure and Splines

Our organization is about to expand its operations into the European Union (EU). To do that we will have to raise several hundred million dollars of collateral to back employees, customers and our supply chain in the EU. Negative interest rates abound in the markets as political events and central bankers vie for control of the euro and its relationship to major currencies. Our task is to help the Chief Financial Officer understand how the term structure of interest rates might change and thus impact the amount and pricing of the collateral our organization is about to raise.

We might ask ourselves:

1. What is the term structure of interest rates?
2. Why would the term structure matter to the CFO?

“Term” refers to the maturity of a debt instrument, the “loan” we get to buy a house. “Term structure” is the schedule of interest rates posted for each maturity. The “term structure” is also known as the “yield curve”. If the maturity on your loan (or your company’s bond) rises, a higher yield might just convince investors (like you) to wait longer for a return of the principal they lend to you. Higher yields mean also that you would have to price the bond at a higher coupon rate. This means more income is paid out to lenders and bondholders than to management, staff, and shareholders. The consequence is cash flow volatility.

Our objective is to conceive a model of bond prices that reflect the underlying dynamic of the term structure of interest rates. Such a model requires us to formulate forward rates of return on bonds of various maturities, and average these rates across the time to maturity in a yield calculation. We will then need to understand how to interpolate and extrapolate rates across maturities. With that tool in hand, we can price bonds, collaterals, and build discount rates to evaluate future expected cash flows.

To do all of this we will employ “regression splines” to interpolate and extrapolate values of forward rates from the term structure. The regression spline will reflect a financial model of the term structure applied to the estimation of actual term structures. We will

#### 3.1 The Bond

Our analysis begins with understanding cash flow and valuation of a simple bond. A typical bond is a financial instrument that pays fixed cash flows for the maturity of the bond with a repayment of the principal (notional or face value) of the bond at maturity. In symbols,

$$V = \sum_{t=0}^{mT} \frac{cP}{(1 + y/m)^t} + \frac{P}{(1 + y/m)^{mT}}$$

Where  $V$  is the present value of the bond,  $T$  is the number of years to maturity,  $m$  is the number of periods cash flows occur per year,  $y$  is the bond yield per year,  $c$  is the coupon rate per year, and  $P$  is the bond’s principal (notional or face value).

Using the idea of an annuity that pays  $(c/m)P$  per period for  $mT$  periods and  $P$  at maturity period  $mT$  we get a nice formula for the present value sum of coupon payments:

$$V = (c/m)P \left( \frac{1}{y/m} - \frac{1}{(y/m)(1 + y/m)^{mT}} \right) + \frac{P}{(1 + y/m)^{mT}}$$

From a financial engineering point of view this formulation is the same as constructing a position that is

- long a perpetuity that pays a unit of currency starting the next compounding period and
- short another perpetuity that starts to pay a unit of currency at the maturity of the bond plus one compounding period.

Our typical bond pays coupons twice a year, so  $m = 2$ . If the bond's maturity in years is 10 years, then  $mT = 10 \times 2 = 20$  compounding periods. We will assume that there is no accrual of interest as of the bond's valuation date for the moment.

```
c <- 0.05
P <- 100
y <- c(0.04, 0.05, 0.06)
m <- 2
T <- 10

(V <- (c/m)*P*(1/(y/m)-1/((y/m)*(1+(y/m))^(m*T)))+P/(1+(y/m))^(m*T))

## [1] 108.17572 100.00000 92.56126
```

### 3.1.1 A quick example

1. If the coupon rate is greater than the yield, why is the price greater than par value?
2. Negative interest rates abound, so set  $y \leftarrow c(-0.02, -0.01, 0.00, 0.01, 0.02)$  and recalculate the potential bond values.

One answers might be:

The bond pays out at a rate greater than what is required in the market. Buyers pay more than par to make up the difference.

Here are more results. We assign variables to assumptions in the question and the calculate the

```
c <- 0.05
P <- 100
y <- c(-0.02, -0.01, 0, 0.01, 0.02)
m <- 2
T <- 10
(V <- (c/m)*P*(1/(y/m)-1/((y/m)*(1+(y/m))^(m*T)))+P/(1+(y/m))^(m*T))

## [1] 177.9215 163.2689      NaN 137.9748 127.0683
```

Why a NaN? Because we are dividing by 0! By the way, these are some hefty premia.

The yield in this model is an average of the forward rates the market used to price future cash flow **during future periods of time**. How can we incorporate a changing forward rate into a model of bond prices? Our strategy is to use the past as a guide for the future and calibrate rates to a curve of rates versus maturities. By using a regression spline model of the term structure we can develop rates at any maturity. Then we can use the interpolated (even extrapolated) rates as building blocks in the valuation of bonds.

### 3.1.2 What's a spline?

A *spline* is a function that is constructed piece-wise from polynomial functions. But imagine the polynomials are pieces of the term structure of interest rates. Each term structure segment is marked off by a set of price and maturity pairs. Whole sections can be marked off by a *knot* at a location in the term structure paired data. Knots are most commonly placed at quantiles to put more knots where data is clustered close together. A different polynomial function is estimated for each range and domain of data between each knot; this is the spline. Now we will use some finance to build a polynomial regression spline out of US Treasury zero-coupon data.

In general a polynomial is an expression that looks like this:



$$f(x) = \alpha_0 x^0 + \alpha_1 x^1 + \alpha_2 x^2 + \dots + \alpha_p x^p,$$

where the  $\alpha$ 's are constant coefficients, and  $x^0 = 1$  and  $x^1 = x$ .

- If  $p = 0$ , then we have a *constant* function.
- if  $p = 1$ , we have a *linear* function.
- $p = 2$ , we have a *quadratic* function.
- $p = 3$ , we have a *cubic* function, and so on...

In our term structure work, We will be using a *cubic* function to build a regression spline.

### 3.1.3 Back to the Bond

Suppose investors give an issuer today the present value, the bond “price”  $P_t$  of receiving back the 100% of the face value of a zero-coupon bond at maturity year (or fraction thereof),  $T$ .

The price of a zero coupon bond, quoted in terms of the percentage of face value, is this expression for discrete computing at rate  $y_t^d$  (say, monthly you get a percentage of last month's balance):

$$P_T = \frac{100}{(1 + y^d(T))^T}$$

Suppose, with the help of Jacob Bernoulli and Leonhardt Euler, we run this experiment:

1. Pay today at the beginning of the year  $P$  to receive \$1 at the end of one year AND interest  $y$  is compounded only once in that year. Thus at the end of the year we receive  $P + yP$ . But by contract this amount is \$1, so that:

$$P + yP = P(1 + y) = 1$$

and solving for  $P$

$$P = \frac{1}{(1 + y)^1}$$

where we raised  $(1 + y)$  to the first power to emphasize that one compound interest period was modeled.

2. Now suppose that we can receive interest twice a year at an annual interest rate of  $y$ , then at the end of the first half of the year we receive half of the interest  $y/2$  so that we have in our contract account

$$P(1 + \frac{y}{2})$$

We then can let this stay in the account this amount will also earn  $y/2$  interest to the end of the contract year so that

$$P(1 + \frac{y}{2}) + \frac{y}{2} [P(1 + \frac{y}{2})] = P(1 + \frac{y}{2})(1 + \frac{y}{2}) = P(1 + \frac{y}{2})^2$$

Setting

$$P(1 + \frac{y}{2})^2 = 1 \text{ dollar}$$

we can solve for  $P$ , the present value of receiving \$1 at the end of the contract year when we have two compounded periods or

$$P = \frac{1}{(1 + \frac{y}{2})^2}$$

3. We can, again with Jacob Bernoulli's help, more generally state for  $m$  compounding periods.

$$P = \frac{1}{(1 + \frac{y}{m})^m}$$

4. Now let's suppose  $y = 100$  percent interest  $m = 365 \times 24 \times 60 = 525600$  compounding periods (minute by minute compounding), then

$$P = \frac{1}{(1 + \frac{1}{525600})^{525600}}$$

```
(m <- 365 * 24 * 60)
```

```
## [1] 525600
```

```
(y <- 1)
```

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

```
(P <- 1/(1 + (y/m))^m)
```

```
## [1] 0.3678798
```

This translates into continuous compounding (you get a percentage of the last nanosecond's balance at the end of this nanosecond... as the nanoseconds get ever smaller) as...

$$P_T(\theta) = 100 \exp(-y_T T)$$

This expression is the present value of receiving a cash flow of 100% of face value at maturity. If the bond has coupons, we can consider each of the coupon payments as a mini-zero bond. Taking the sum of the present value of each of the mini-zeros give us the value of the bond, now seen as a portfolio of mini-zeros.

The yield  $y_t$  the rate from date 0 to date  $T$ , maturity. it covers the stream of rates for each intermediate maturity from 0 to  $T$ . Suppose, we define forward rate  $r(t, \theta)$ , where each  $t$  is one of the intermediate maturity dates between time 0 and maturity  $T$ , and  $\theta$  contains all of the information we need about the shape of  $r$  across maturities. We can estimate the forward curve from bond prices  $P(T)$  of the  $T$ th maturity with

$$-\frac{\Delta \log(P(T_i))}{\Delta T_i} = -\frac{\log(P(T_i)) - \log(P(T_{i-1}))}{T_i - T_{i-1}}$$

The  $\Delta$  stands for the difference in one price or maturity  $i$  and the previous price and maturity  $i - 1$ . The  $\log()$  function is the natural logarithm. An example will follow.

The *yield* is then the *average of the forward rates* from date 0 to date  $T$  of a zero bond. We can use the integral, which amounts to a cumulative sum, to compute this average:

$$y_T(\theta) = T^{-1} \int_0^T r(t, \theta) dt$$

The numerator is the cumulative sum of the forward rates for each maturity up to the last maturity  $T$ . In this expression,  $r dt$  is the forward rate across a small movement in maturity. The denominator is the number of maturity years  $T$ .

### 3.1.4 An example to clarify

Load these lines of R into the RStudio console:

```
maturity <- c(1, 5, 10, 20, 30) # in years
price <- c(99, 98, 96, 93, 89) # in percentage of face value
```

A. now let's experiment on these zero-coupon prices with their respective maturities:

1. Calculate the ``log(price)/100``. Then find the forward rates using the following

```
```r
(forward <- -diff(log(price))/ diff(maturity))
```

## [1] 0.002538093 0.004123857 0.003174870 0.004396312
```

2. Compare ``log(price)`` with ``price``.

3. What does the forward rate formula indicate? What would we use it for?

B. Find the yield-to-maturity curve and recover the bond prices using

```
(forward.initial <- -log(price[1]/100))

## [1] 0.01005034
(forward.integrate <- c(forward.initial,
                       forward.initial + cumsum(forward * diff(maturity))))

## [1] 0.01005034 0.02020271 0.04082199 0.07257069 0.11653382
# a rolling integration of rates
# across maturities
(price <- 100 * exp(-forward.integrate))

## [1] 99 98 96 93 89
# present value of receiving 100% of
# face value
```

1. What is the interpretation of the `forward.integrate` vector?

2. What happened to the first bond price?

3. Did we recover the original prices?

Some results follow.

For question A we ran the forward rates. Let's run the natural logarithm of price:

```
(forward <- -diff(log(price/100)/diff(maturity)))

## [1] 1.527958e-03 4.165799e-05 3.174870e-03 2.187638e-02
(-log(price/100))

## [1] 0.01005034 0.02020271 0.04082199 0.07257069 0.11653382
```

`-log(price/100)` seems to give us the yield-to-maturity directly. But do look at `-diff(log(price))` next:

```
(-diff(log(price/100)))
```

```
## [1] 0.01015237 0.02061929 0.03174870 0.04396312
```

These look like rates, because they are. They are the continuous time version of a percentage change, or growth, rate from one maturity to another maturity. We can use these numerical results to motivate an interpretation: interest rates are simply rates of change of present values relative to how much time they cover in maturity.

Now let's calculate

```
(-diff(price/100)/(price[-length(price)/100]))
```

```
## [1] 0.0001020408 0.0002083333 0.0003225806 0.0004494382
```

These are discrete percentage changes that are similar, but not quite the same, as the continuous (using `log()`) version.. Note the use of the indexing of price to eliminate the last price, since what we want to compute is:

$$\frac{P(T_i) - P(T_{i-1})}{P(T_{i-1})}$$

### 3.1.5 Rolling the integration

Running the code for question B we get:

```
(forward.initial <- -log(price[1]/100))
```

```
## [1] 0.01005034
```

```
(forward.integrate <- c(forward.initial,
                        forward.initial + cumsum(forward * diff(maturity))))
```

```
## [1] 0.01005034 0.01616217 0.01637046 0.04811915 0.26688300
```

```
# a rolling integration of rates
```

```
# across maturities
```

```
(price <- 100 * exp(-forward.integrate))
```

```
## [1] 99.00000 98.39677 98.37628 95.30202 76.57627
```

```
# present value of receiving 100% of  
# face value
```

The rolling “integration” is an accumulative process of adding more forward rates as the maturity advances, thus a cumulative sum or in R a `cumsum` is deployed.

1. Yields are the accumulation of forward rates. Thus the use of the cumulative sum as a discrete version
2. The first forward rate is just the discount rate on the 1-year maturity bond stored in `forward.initial`
3. All bond prices are recovered by inverting the process and producing forwards from prices and converting

## 3.2 Forward Rate Parameters

Now we attend to the therefore mysterious  $\theta$ , embedded in the forward rate sequence  $r$ . We suppose the forward rate is composed of a long-term constant,  $\theta_0$ ; a slope term with parameter  $\theta_1$ ; a more shapely, even “humped” term with parameter  $\theta_2$ ; and so on for  $p$  polynomial terms:

$$r(t, \theta) = \theta_0 + \theta_1 t + \theta_2 t^2 + \cdots + \theta_p t^p$$

Stopped at Page 134

## 4 Market Risk

Suppose a division in our company buys electricity to make steel. We know of two very volatile factors in this process:

- \* 1. The price of steel at the revenue end, and the other is
- \* 2. The price of electricity at the cost end.

To model the joint volatility of the power-steel spread We can use electricity and steel prices straight from the commodity markets. We can also use stock market indexes or company stock prices from electricity producers and transmitters and from a steel products company to proxy for commodities. Using company proxies gives us a broader view of these commodities than just the traded pure play in them.

In this chapter we will

1. Measure risks using historical and parametric approaches
2. Interpret results relative to business decisions
3. Visualize market risk

### 4.1 What is Market Risk

Market risk for financial markets is the impact of unanticipated price changes on the value of an organization's position in instruments, commodities, and other contracts. In commodity markets there is sometimes considered a more physical type of market risk called volumetric risk that relates to the delivery of the commodity to a buyer. This risk might be triggered by a complex contract such as a CDO or a spark spread tolling agreement in power and energy markets. Here we will assume that volumetric changes are physical in the sense that a electricity system operator governs a physical process such as idling an electric generation plant.

A "position" is the physical holding of an asset or commodity, such as a share of Apple stock, or an ounce of gold. A long position means that the market participant possesses a positive amount of the asset or commodity. A short position means that the market participant does not possess an asset or commodity. The "position" is considered exogenous to the price stochastic process. This implies that changes in position do not affect the liquidity of the market relative to that position.

#### 4.1.1 Try this exercise

Suppose you are in charge of the project to manage the contracting for electricity and steel at your speciality steel company. Your company and industry have traditionally used tolling agreements to manage the steel-power spread, both from a power input and a steel output point of view.

1. Look up a tolling agreement and summarize its main components.
2. What are the input and output decisions in this kind of agreement.

Here are some results

1. In the electric power to steel tolling agreement a steel buyer supplies power to a steel plant and receives from the plant supplier an amount of steel based on an assumed power-to-steel transformation rate at an agreed cost.

- \* Prices of power and steel

- \* Transformation rate

- \* Agree cost

2. Decisions include

- \* Buying an amount of power (MWh)

- \* Selling an amount of steel (tons)

- \* Scheduling plant operations

Decisions will thus depend on the prices of electricity and steel, the customer and vendor segments served, the technology that determines the steel (tons) / Power (MWh) transformation rate, start-up, idle, and shut-down timing and costs and overall plant production costs.

## 4.2 History Speaks

To get the basic idea of risk measures across we develop the *value at risk* and *expected shortfall* metrics from the historical simulated distributions of risk factors. Given these risk factors we combine them into a portfolio and calculate their losses. Finally with the *loss* distribution in hand we can compute the risk measures.

We will use a purely, non-parametric historical simulation approach in this section. This means that we will not need to compute means, standard deviations, correlations, or other statistical estimators, also known as parameters.

First we need to get some data. We will use throughout these computations several libraries:

1. `mvtnorm` builds multivariate normal (Gaussian) simulations and

2. `QRM` estimates Student-t and generalized pareto distribution (GPD) simulation.

3. We will hold off on these parametric approaches till later and start with historical simulation.

4. The `psych` library helps us to explore the interactions among data through scatterplots and histograms.

5. The `ggplot2` library allows us to build complex visualizations that will aid the generation of further insights.

We read in the csv file from the working directory. This file contains dates and several risk factors. In this setup we will use RWE stock prices will stand in for electricity price risk factor input and THYSSEN stock prices for the steel price risk factor.

```
# Download the data
data.all <- read.csv("./data/eurostock.csv", stringsAsFactors = FALSE)

# This will convert string dates to
## date objects below

str(data.all)
```

```
## 'data.frame':   6147 obs. of  24 variables:
## $ X              : chr  "1973-01-01" "1973-01-02" "1973-01-03" "1973-01-04" ...
```

```
## $ ALLIANZ.HLDG. : num 156 156 161 162 164 ...
## $ COMMERZBANK : num 147 147 149 152 152 ...
## $ DRESHDNER.BANK : num 18.4 18.4 18.8 18.9 18.9 ...
## $ BMW : num 104 109 110 111 109 ...
## $ SCHERING : num 36.9 37.4 37.8 37.9 37.4 ...
## $ BASF : num 15 15.4 15.6 15.8 15.8 ...
## $ BAYER : num 12.2 11.9 12.1 12.7 12.7 ...
## $ BAYERISCHE.VBK. : num 23.5 22.9 23.4 23.7 23.9 ...
## $ BAYER.HYPBK. : num 23.4 23.2 23.3 23.5 23.4 ...
## $ DEGUSSA : num 203 207 208 210 214 ...
## $ DEUTSCHE.BANK : num 22.3 22.5 22.9 23 23.3 ...
## $ CONTINENTAL : num 8.54 8.83 8.78 8.83 8.73 8.82 8.74 8.73 8.74 8.74 ...
## $ VOLKSWAGEN : num 134 140 145 144 140 ...
## $ DAIMLER.BENZ : num 17 17.6 17.8 17.8 17.7 ...
## $ HOECHST : num 13.8 13.8 14.2 14.3 14.2 ...
## $ SIEMENS : num 20.8 21.1 21.3 21.4 21.5 ...
## $ KARSTADT : num 360 360 362 369 368 ...
## $ LINDE : num 136 137 140 142 144 ...
## $ THYSSEN : num 67.5 68.4 67.5 71.6 71.2 ...
## $ MANNESMANN : num 85 86.5 87.8 88.7 88.6 ...
## $ MAN : num 118 119 125 125 127 ...
## $ RWE : num 11.7 11.9 12 11.9 12 ...
## $ INDEX : num 536 545 552 556 557 ...
```

The next thing we must do is transform the data set into a time series object. The way we do that is to make the dates into row names so that dates are the index for the two riskfactors. Making dates an index allows us to easily filter the data.

```
str(row.names <- data.all$X) ## We find that the first field X contains dates

## chr [1:6147] "1973-01-01" "1973-01-02" "1973-01-03" "1973-01-04" ...
date <- as.Date(row.names) ## convert string dates to date objects
str(date) ## Always look at structure to be sure

## Date[1:6147], format: "1973-01-01" "1973-01-02" "1973-01-03" "1973-01-04" "1973-01-05" ...
rownames(data.all) <- date
head(data.all)
```

```
##           X ALLIANZ.HLDG. COMMERZBANK DRESHDNER.BANK    BMW
## 1973-01-01 1973-01-01      155.51      147.41      18.40 103.97
## 1973-01-02 1973-01-02      155.51      147.41      18.40 109.05
## 1973-01-03 1973-01-03      160.58      149.14      18.80 109.83
## 1973-01-04 1973-01-04      162.27      152.05      18.91 110.81
## 1973-01-05 1973-01-05      164.30      152.05      18.89 109.44
## 1973-01-08 1973-01-08      164.30      152.25      18.99 109.05
##           SCHERING  BASF  BAYER  BAYERISCHE.VBK.  BAYER.HYPBK.  DEGUSSA
## 1973-01-01      36.88 14.96 12.24      23.47      23.40 203.46
## 1973-01-02      37.44 15.43 11.95      22.92      23.22 206.85
## 1973-01-03      37.79 15.61 12.10      23.45      23.34 208.21
## 1973-01-04      37.86 15.85 12.71      23.66      23.49 210.11
## 1973-01-05      37.44 15.75 12.74      23.87      23.40 214.31
## 1973-01-08      37.79 15.80 12.74      24.07      23.46 216.68
##           DEUTSCHE.BANK  CONTINENTAL  VOLKSWAGEN  DAIMLER.BENZ  HOECHST
## 1973-01-01      22.29      8.54      134.06      16.97      13.77
## 1973-01-02      22.50      8.83      140.00      17.59      13.77
```

```
## 1973-01-03      22.86      8.78      144.53      17.79      14.22
## 1973-01-04      23.04      8.83      144.04      17.81      14.32
## 1973-01-05      23.29      8.73      139.92      17.73      14.23
## 1973-01-08      23.18      8.82      143.77      17.70      14.19
##      SIEMENS KARSTADT LINDE THYSSEN MANNESMANN      MAN      RWE      INDEX
## 1973-01-01      20.76      359.55      135.95      67.47      84.97      117.92      11.68      536.36
## 1973-01-02      21.06      359.96      136.89      68.41      86.51      118.78      11.87      545.43
## 1973-01-03      21.29      361.99      139.59      67.47      87.75      124.95      12.03      552.46
## 1973-01-04      21.44      369.32      142.21      71.62      88.71      124.95      11.95      556.14
## 1973-01-05      21.48      368.50      143.71      71.24      88.63      127.29      12.03      557.44
## 1973-01-08      21.48      366.88      143.77      70.77      89.01      125.34      11.91      555.51
```

```
tail(data.all) ## And always look at data
```

```
##      X ALLIANZ.HLDG. COMMERZBANK DRESDNER.BANK      BMW
## 1996-07-16 1996-07-16      2550      323.0      38.15      843.5
## 1996-07-17 1996-07-17      2572      331.0      38.35      845.2
## 1996-07-18 1996-07-18      2619      335.0      39.60      844.0
## 1996-07-19 1996-07-19      2678      336.8      39.50      847.5
## 1996-07-22 1996-07-22      2632      336.8      39.00      844.0
## 1996-07-23 1996-07-23      2622      337.5      39.20      844.0
##      SCHERING BASF BAYER BAYERISCHE.VBK. BAYER.HYPBK. DEGUSSA
## 1996-07-16      101.0      41.00      50.45      47.45      40.20      498.0
## 1996-07-17      102.5      41.87      50.92      48.08      40.55      503.2
## 1996-07-18      101.2      41.86      52.00      49.05      41.48      507.5
## 1996-07-19      102.9      42.10      51.85      49.48      41.92      506.0
## 1996-07-22      101.8      40.70      50.60      49.40      41.40      501.0
## 1996-07-23      102.0      40.15      50.25      49.88      41.55      499.0
##      DEUTSCHE.BANK CONTINENTAL VOLKSWAGEN DAIMLER.BENZ HOECHST
## 1996-07-16      72.10      23.00      531.00      78.45      49.85
## 1996-07-17      72.86      23.63      539.00      79.30      50.30
## 1996-07-18      74.30      24.11      528.50      78.00      50.50
## 1996-07-19      74.90      24.18      531.00      78.25      50.70
## 1996-07-22      73.60      24.10      522.25      77.48      49.20
## 1996-07-23      73.70      24.15      515.00      77.35      48.35
##      SIEMENS KARSTADT LINDE THYSSEN MANNESMANN      MAN      RWE      INDEX
## 1996-07-16      78.75      544.0      923      274.0      536.0      373.0      54.20      2469.79
## 1996-07-17      78.79      554.0      925      273.1      542.0      374.5      54.40      2497.19
## 1996-07-18      77.61      543.0      920      271.0      536.7      369.0      55.00      2506.22
## 1996-07-19      77.24      543.0      932      271.9      535.3      369.5      54.33      2520.19
## 1996-07-22      76.49      540.0      931      268.1      529.5      364.0      52.90      2482.40
## 1996-07-23      76.90      539.5      935      265.5      530.5      360.0      53.15      2475.07
```

With this machinery in hand we can subset the data by starting and ending date as well as the choice of RWE and THYSSEN

```
# Subset the data using a start and
# end date

start.date <- "1975-06-02"
end.date <- "1990-12-30"
## First column looks for filtered
## dates, second and third columns
## pull out prices

price <- data.all[start.date <= date & date <= end.date, c("RWE", "THYSSEN")]
```



```
## We add a check to ensure that price
## is a matrix and that ncol will work

if(!is.matrix(price)) {
  price <- rbind(price, deparse.level = 0L)
}
str(price)

## 'data.frame': 4065 obs. of 2 variables:
## $ RWE : num 8.96 9.2 9.16 9.2 9.36 9.24 9.12 9.08 9.04 8.99 ...
## $ THYSSEN: num 69.8 70.8 69.8 68.9 68.8 ...
```

```
head(price) ## show the beginning
```

```
##           RWE THYSSEN
## 1975-06-02 8.96   69.82
## 1975-06-03 9.20   70.77
## 1975-06-04 9.16   69.82
## 1975-06-05 9.20   68.88
## 1975-06-06 9.36   68.79
## 1975-06-09 9.24   67.94
```

```
tail(price) ## and the en
```

```
##           RWE THYSSEN
## 1990-12-21 36.36   187.5
## 1990-12-24 36.36   187.5
## 1990-12-25 36.36   187.5
## 1990-12-26 36.36   187.5
## 1990-12-27 36.28   186.5
## 1990-12-28 35.75   184.5
```

The code before the `str`, `head`, and `tail` filters the price data by start and end dates. We could also perform this head and tail work using the following code.

```
(end.idx <- dim(price)[1])
```

```
## [1] 4065
```

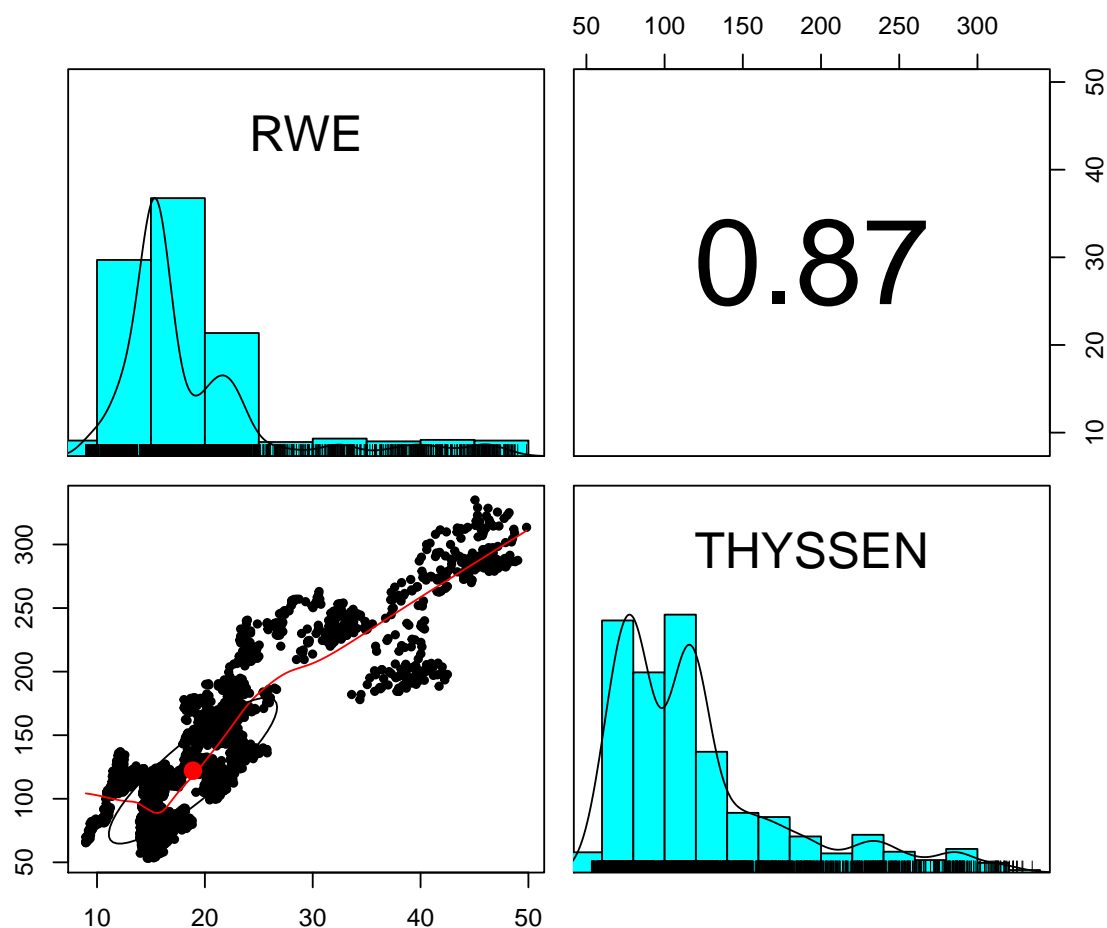
```
(price.2 <- rbind(price[1:5, ], price[(end.idx - 4):end.idx, ]))
```

```
##           RWE THYSSEN
## 1975-06-02 8.96   69.82
## 1975-06-03 9.20   70.77
## 1975-06-04 9.16   69.82
## 1975-06-05 9.20   68.88
## 1975-06-06 9.36   68.79
## 1990-12-24 36.36  187.50
## 1990-12-25 36.36  187.50
## 1990-12-26 36.36  187.50
## 1990-12-27 36.28  186.50
## 1990-12-28 35.75  184.50
```

### 4.3 Try this exercise

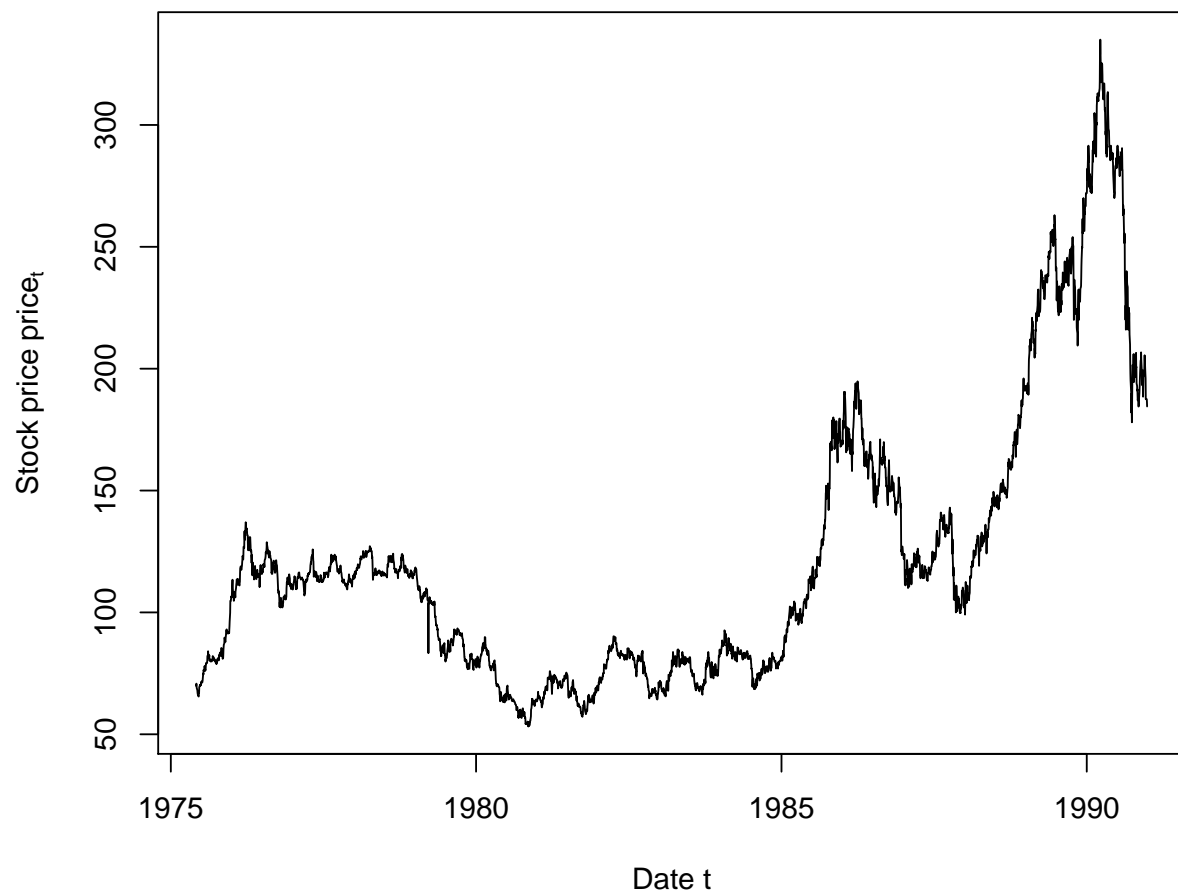
Now let's really explore this data. The library `psych` has a prefabricated scatter plot histogram matrix we can use. With this composite plot we can examine historical relationships between the two risk factors as well as the shape of the risk factors themselves. We can also use this device to look at dependent simulations. After the scatter plots, we then look at the time series plots of the two factors.

```
# Use scatter plots of the two price series along with their histograms
# to examine the relationship
# between REW and THYSSEN
library(psych)
pairs.panels(price)
```



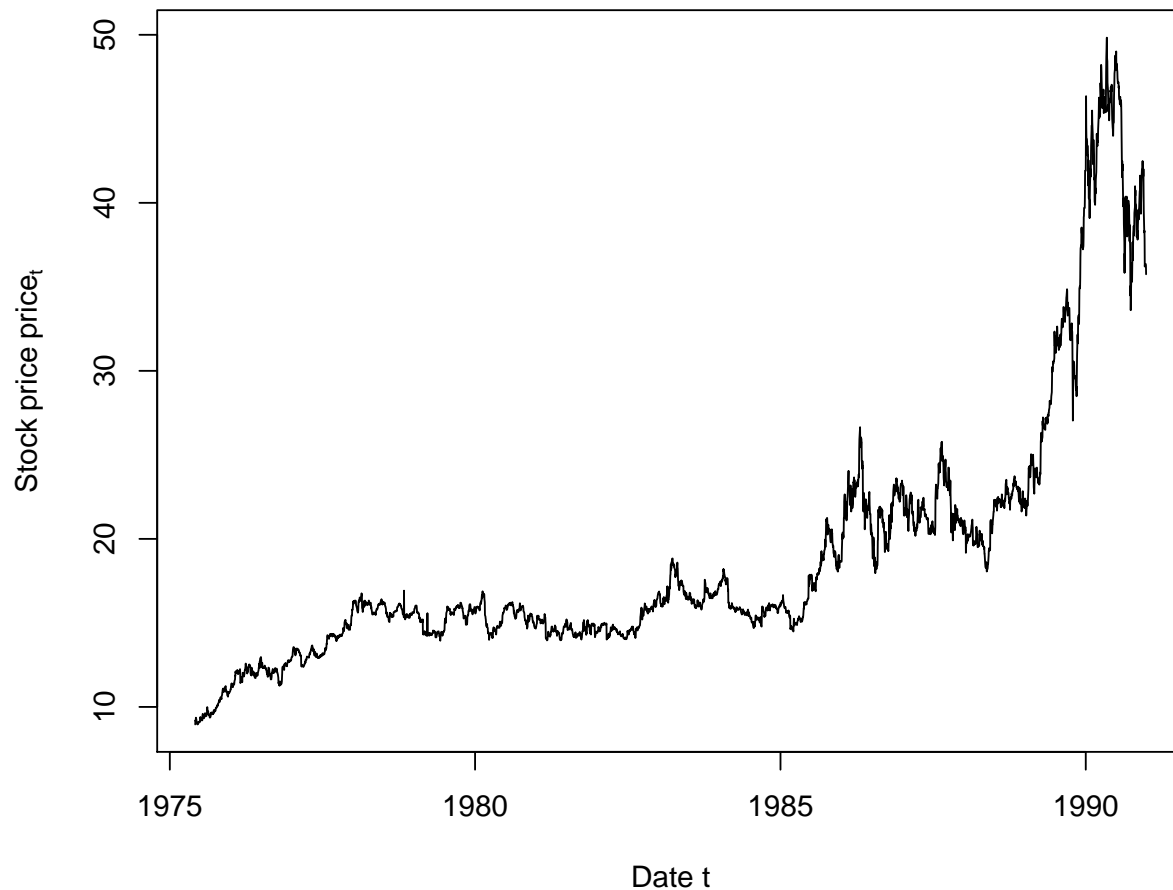
```
price.rownames <- rownames(price)
plot(as.Date(price.rownames), price[, "THYSSEN"],
     type = "l",
     main = "Thyssen stock price data", ## title
     xlab = "Date t", ## x-axis label
     ylab = expression(Stock ~ price ~ price[t]) ## y-axis label
)
```

### Thyssen stock price data



```
plot(as.Date(price.rownames), price[, "RWE"],  
     type = "l",  
     main = "RWE stock price data", ## title  
     xlab = "Date t", ## x-axis label  
     ylab = expression(Stock ~ price ~ price[t]) ## y-axis label  
)
```

## RWE stock price data



The `pairs.panel` plot displays a matrix of interaction between RWE and THYSSEN. Price levels are interesting but, as we have seen, are not stable predictors. Let's transform them into returns next.

### 4.4 Now to the Matter at Hand

Now to the matter at hand: *value at risk* and *expected shortfall*. These two measures are based on the quantile of losses attributable to risk factors. Value at risk is the quantile at an  $\alpha$  level of tolerance. Expected shortfall is the mean of the distribution beyond the value at risk threshold.

TO get losses attributable to market risk factors we compute log price differences (also called log price relatives). These can be interpreted as returns, or simply as percentage changes, in the risk factor prices (superior regressions require transformation towards stationary processes).

A plot lets us examine the results.

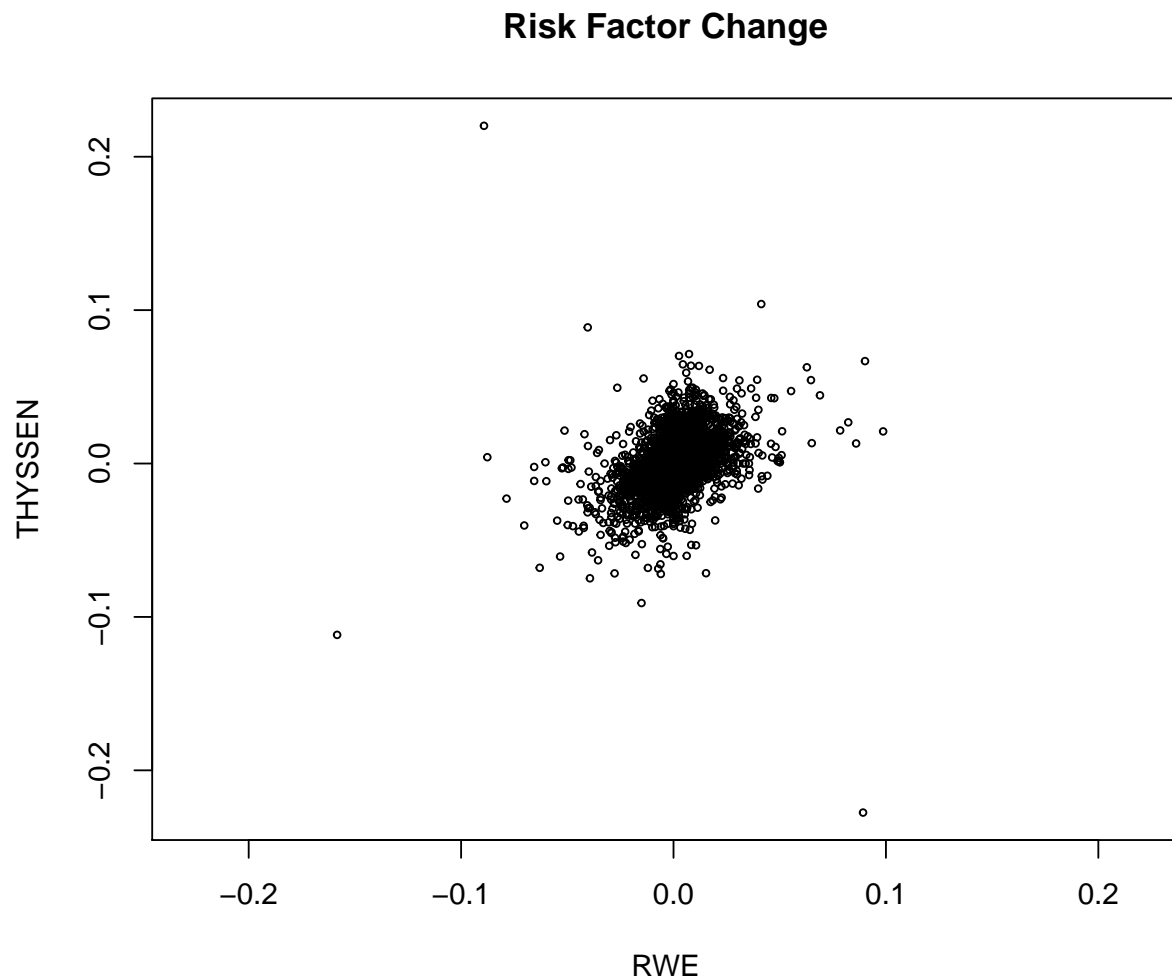
```
# Here we can compute two items  
# together: log price differences  
# and their range (to bound a plot)  
  
return.range <- range(return.series <- apply(log(price)
```

```

, 2, diff)) ## compute log-returns and range
return.range

## [1] -0.2275282  0.2201375
plot(return.series, xlim = return.range,
      ylim = return.range,
      main = "Risk Factor Change",
      cex = 0.5
    )

```



Using the returns we can now compute loss. Weights are defined as the value of the positions in each risk factor. We can compute this as the notional times the last price. Remember we are talking about an input, electricity, and an output, steel. We form the margin:

$$Margin = price_{steel} \times tons - price_{power} \times [(rate_{MWh/tons} \times tons)],$$

where the last term is the power to steel transformation rate that converts power price dollars per MWh to dollars per tons.

We convert prices to share prices and tons to equivalent values in terms of the number of shares. The naturally short position in power is equivalent to the negative number of shares (in the square brackets). The naturally long position in steel is equivalent to a positive number of shares. By naturally short we mean that power is an input, incurs a cost, and is demanded by the plant, and supplied by a third party. By naturally long we mean that steel is an output, earns a revenue, and demanded by a third party.

```
## Get last prices
price.last <- as.numeric(tail(price, n = 1))

## Specify the positions
position.rf <- c(-30, 10)

## And compute the position weights
w <- position.rf * price.last

# Fan these across the length and
## breadth of the risk factor series
weights.rf <- matrix(w, nrow = nrow(return.series),
                     ncol = ncol(return.series), byrow = TRUE)

# We need to compute exp(x) - 1 for
# very small x: expm1 accomplishes
# this

loss.rf <- -rowSums(expm1(return.series) * weights.rf)

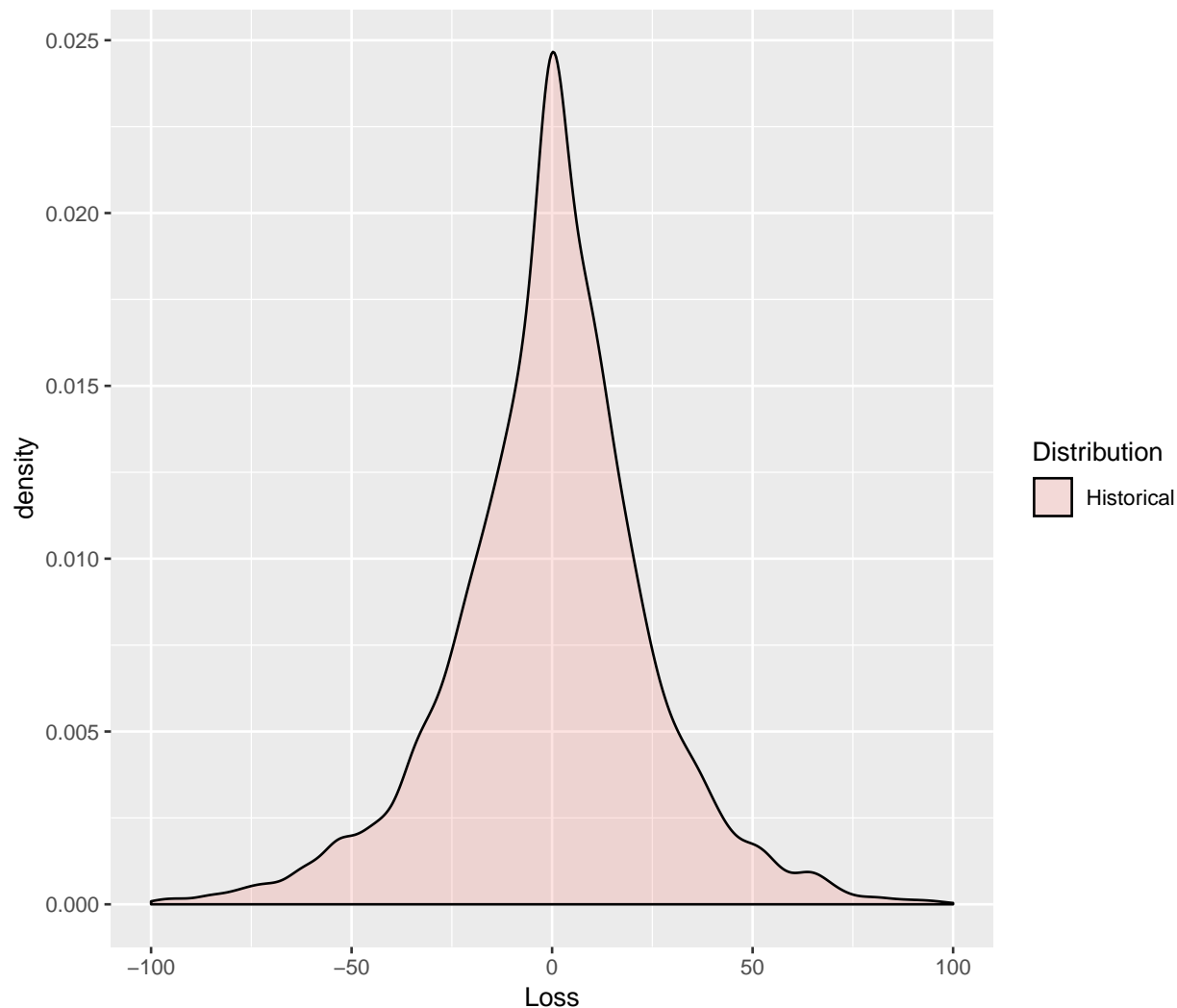
summary(loss.rf)

##      Min.   1st Qu.   Median     Mean   3rd Qu.    Max.
## -545.8825 -13.3084   0.0000  -0.2467  13.1330  475.5576

loss.rf.df <- data.frame(Loss = loss.rf,
                        Distribution = rep("Historical",
                                          each = length(loss.rf)))

library(ggplot2)

ggplot(loss.rf.df, aes(x = Loss, fill = Distribution)) + geom_density(alpha = 0.2) + xlim(-100, 100)
```



The plos reveals some interesting deep and shallow outliers. The distribution is definately very peaked. We use the base function `expm1` that computes the natural exponent of returns all minus 1.

$$e^r - 1$$

Some of these returns, or percentage price changes if you will, are very close to zero. High precision arithmetic is needed to get acurate calculations. The function `expm1` does this well.

Now we get to estimating value at risk (VaR) and expected shortfall (ES). WE set the tolerance level  $\alpha$ , for example, equal to 95%. This would mean that decision maker would not tolerate loss in more than 5% of all risk scenarios.

We define the VaR as the quantile for probability  $\alpha \in (0, 1)$  as

$$\text{VaR}_\alpha(X) = \inf\{x \in R : F(x) \geq \alpha\},$$

which means find the greatest lower bound of loss  $x$  (what the symbol *inf* = *infimum* means in English), such that the cumulative probability of  $x$  is greater than or equal to  $\alpha$ .

Using the  $\text{VaR}_\alpha$  definition we can also define ES as

$$ES_\alpha = E[X|X \geq VaR_\alpha]$$

Where  $ES$  is the “expected shortfall” and  $E$  is the expectation operator, also known as the “mean”. Again, in English, the expected shortfall is the average of all losses greater than the loss at a  $VaR$  associated with probability  $\alpha$ , and  $ES \geq VaR$ .

#### 4.4.1 Example

First the computation of VaR and ES:

```
# Simple Value at Risk
alpha.tolerance <- 0.99
(VaR.hist <- quantile(loss.rf, probs = alpha.tolerance,
                      names = FALSE))
```

```
## [1] 67.43459
```

```
# Just as simple Expected shortfall
(ES.hist <- mean(loss.rf[loss.rf > VaR.hist]))
```

```
## [1] 97.97649
```

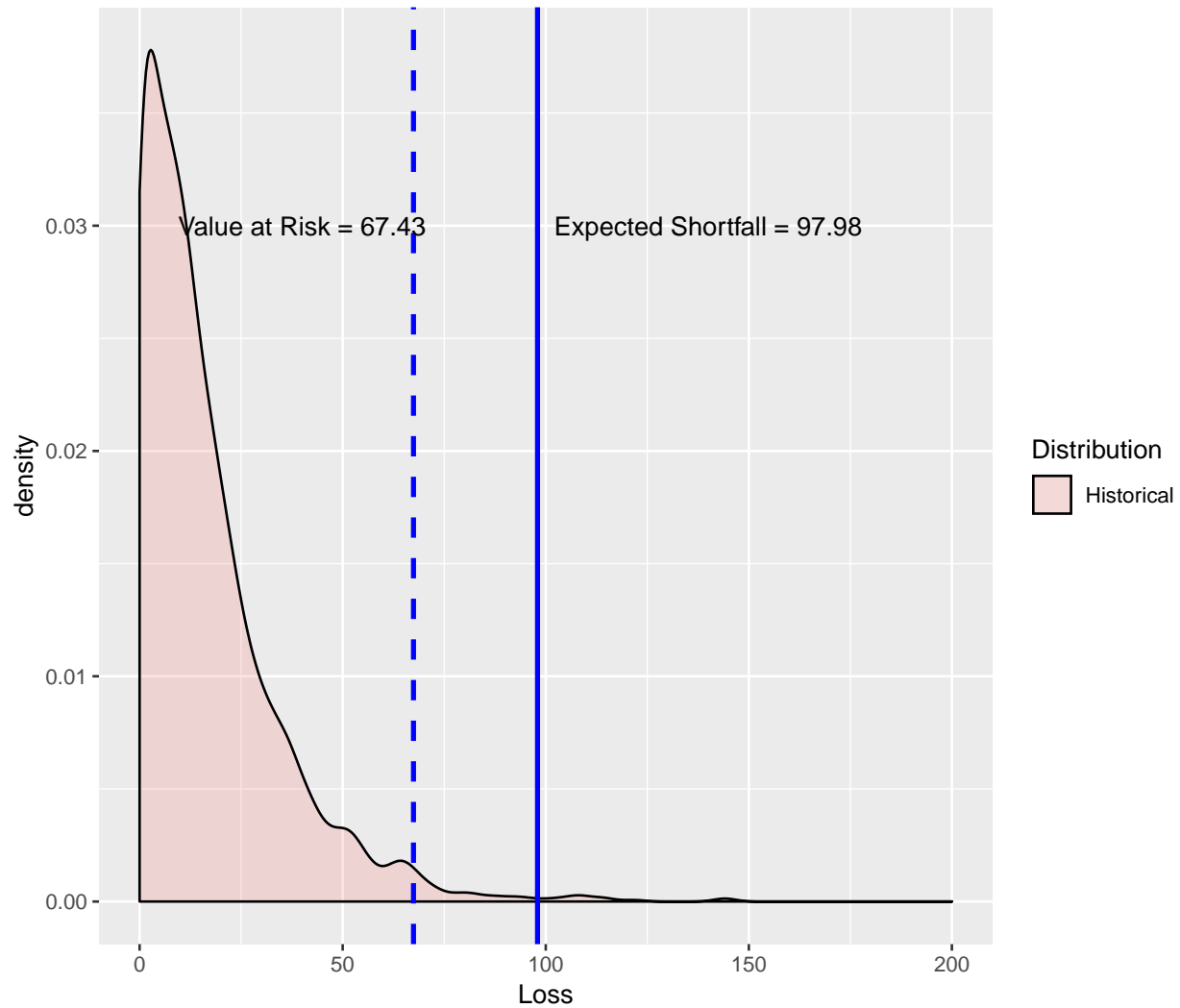
Next we set up the text and plotting environment.

```
VaR.text <- paste("Value at Risk =", round(VaR.hist, 2))

ES.text <- paste("Expected Shortfall =", round(ES.hist, 2))

ggplot(loss.rf.df, aes(x = Loss, fill = Distribution)) + geom_density(alpha = 0.2) +
  geom_vline(aes(xintercept = VaR.hist), linetype = "dashed", size = 1, color = "blue") +
  geom_vline(aes(xintercept = ES.hist), size = 1, color = "blue") +
  xlim(0, 200) +
  annotate("text", x = 40, y = 0.03, label = VaR.text) +
  annotate("text", x = 140, y = 0.03, label = ES.text)
```



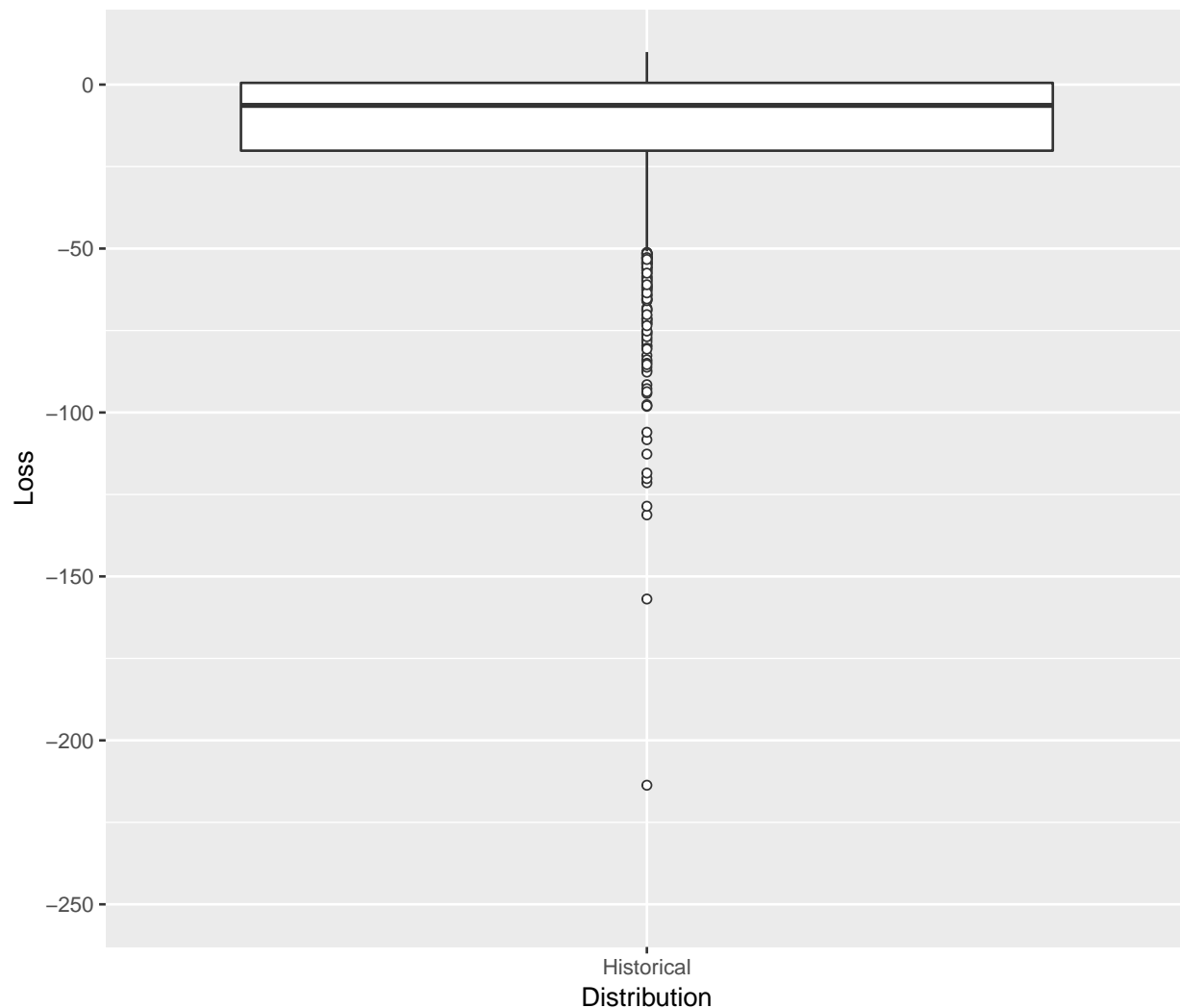


We see that ES is much bigger than VaR but also much less than the maximum historical loss.

One note: VaR is computed as a pre-event indicator beyond a loss of 0 in this example. Many applications of this metric center loss at the median loss. Thus, loss would be computed as gross loss minus the median (50th percentile of loss).

A box plot might also help us visualize the results without resorting to a probability distribution function.

```
ggplot(loss.rf.df, aes(x = Distribution,
                       y = Loss)) +
  geom_boxplot(outlier.size = 1.5, outlier.shape = 21) +
  ylim(-250, 10)
```



This box plot might look better with more than one distribution. So far we simply let history speak for itself. We didn't assume anything at all about the shape of the data. We just used the empirical record as the shape. In what follows let's start to put some different shapes into the loss potential of our tolling agreement.

#### 4.5 Carl Friedrich Gauss, I Presume...

What we just did was the classic historical simulation technique for computing tail risk measures. Historical simulation is a "nonparametric" technique, since there is no estimation of parameters conditional on a distribution. Only history, unadorned, informs risk measurement. Now we shift gears into the parametric work of Gauss: Gaussian, Generalized Pareto and as an exercise Gossett's (Student's) distributions.

Carl Friedrich Gauss is often credited with the discovery of the normal distribution. So we tack his name onto the normal distribution. This distribution has a crucial role in quantitative risk and finance. It is often the basis for most derivative pricing models and for simulation of risk factors in general. It does not exhibit thick tails, and definitely is not skewed or peaked. This distribution definitely does not describe volatility clustering we observe in most financial and commodity time series. Nevertheless, it is otherwise ubiquitous, if only as a benchmark (like "perfect competition" or "efficient markets").

With just a little of math here, we can define the Gaussian (normal) distribution function. If  $x$  is a uniformly distributed random variable, then

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$$

is the probability density function of the normally distributed  $x$  with mean  $\mu$  and standard deviation  $\sigma$ .

“Halfway” between the normal Gaussian distribution and Student’s  $t$  is the chi-square,  $\chi^2$ , distribution. We define  $\chi^2$  as the distribution of the sum of the square normal distribution variables  $x$  with density function and  $k$  degrees of freedom for  $x > 0$ :

$$f(x) = \frac{x^{(k/2-1)}e^{-x/2}}{2^{k/2}\Gamma\left(\frac{k}{2}\right)}$$

and 0 otherwise. The “degrees of freedom” are the number of normal distributions used to create a chi-square variate.

Now on to Student’s  $t$  distribution which is defined in terms of the Gaussian and chi-square distribution as the ratio of a Gaussian random variate to the square root of a chi-squared random variate. Student (a pseudonym for William Sealy Gossett) will have thicker tails but also the same symmetry as the normal curve.

Here is a quick comparison of the standard Gaussian and the Student’s  $t$  distribution. The functions `rnorm` and `rt` generate Gaussian and Student’s  $t$  variates, respectively. The function `qnorm` and `qt` compute the distance from the mean (probability = 50%) for a given probability here stored in `alpha.tolerance`.

```
library(mvtnorm) ## Allows us to generate Gaussian and Student-t variates
library(ggplot2)
set.seed(1016)

n.sim <- 1000
z <- rnorm(n.sim)
t <- rt(n.sim, df = 5)
alpha.tolerance <- 0.95
(z.threshold <- qnorm(alpha.tolerance))
```

```
## [1] 1.644854
```

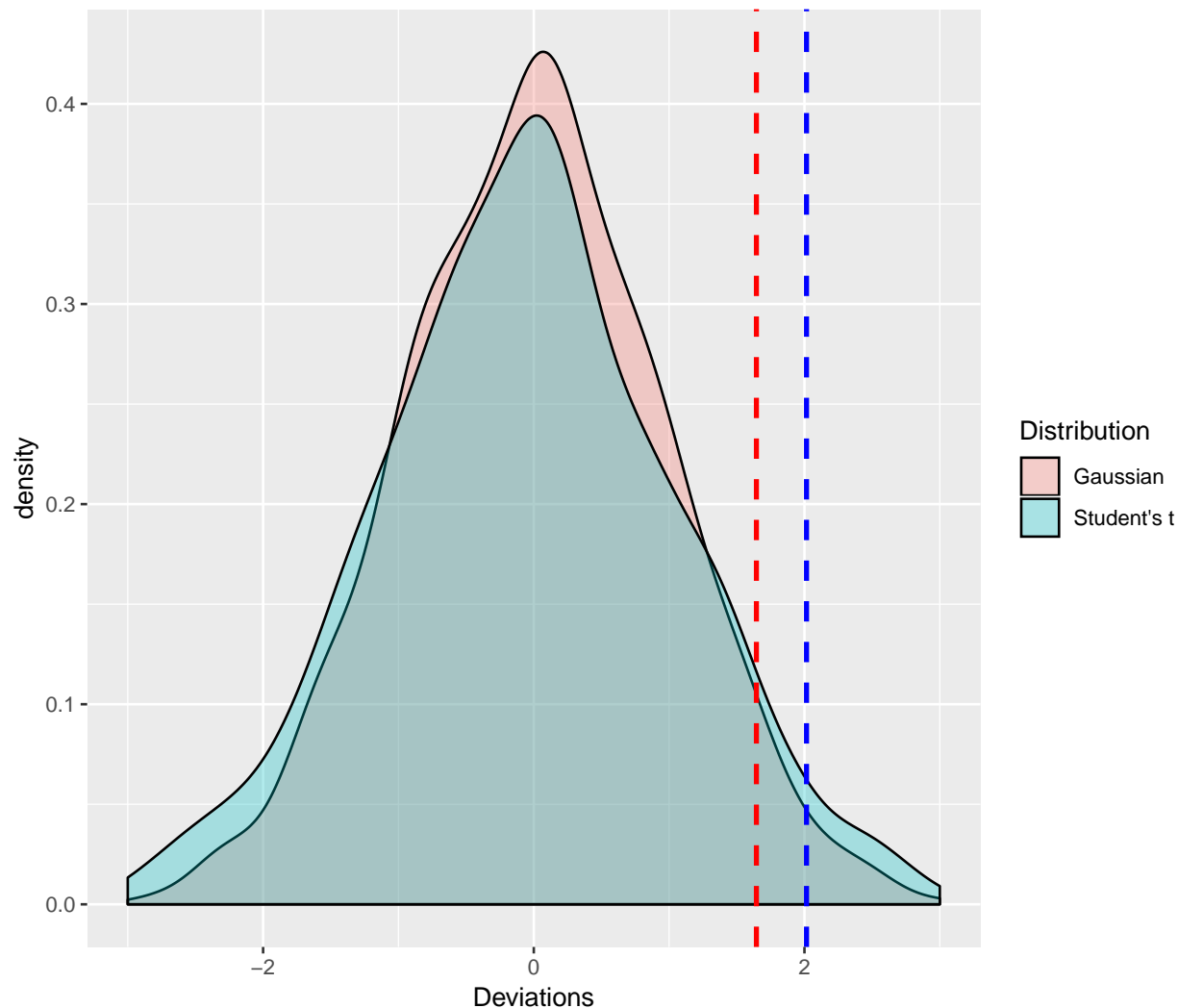
```
(t.threshold <- qt(alpha.tolerance, df = 5))
```

```
## [1] 2.015048
```

We now make a data frame and plot with `ggplot`:

```
zt.df <- data.frame(Deviations = c(z, t), Distribution = rep(c("Gaussian", "Student's t"), each = n.sim))

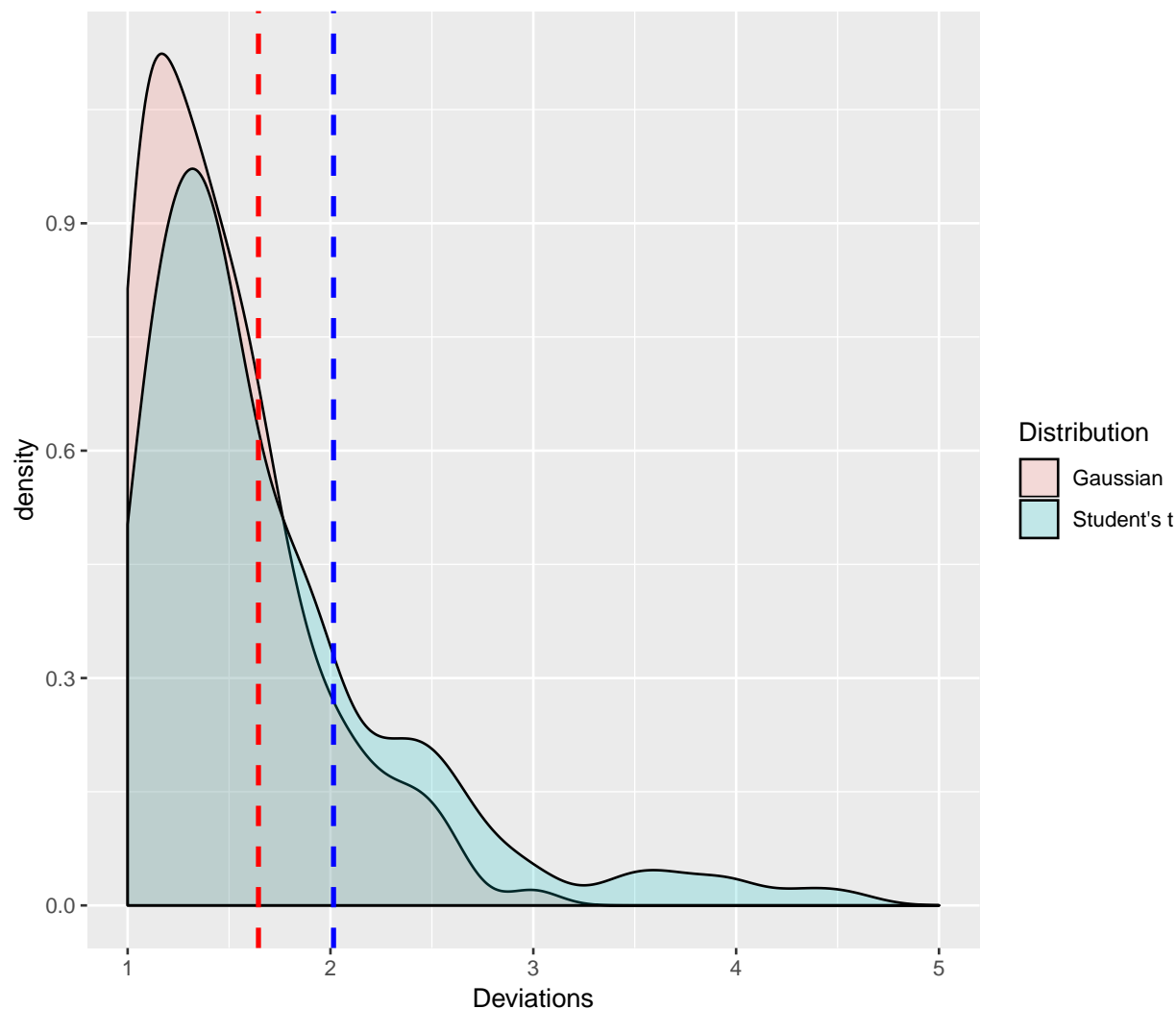
ggplot(zt.df, aes(x = Deviations, fill = Distribution)) +
  geom_density(alpha = 0.3) + geom_vline(aes(xintercept = z.threshold),
    color = "red",
    linetype = "dashed",
    size = 1
  ) +
  geom_vline(aes(xintercept = t.threshold),
    color = "blue",
    linetype = "dashed",
    size = 1) + xlim(-3, 3)
```



We see the two distributions are nearly the same in appearance. But the Student's t tail is indeed thicker in the tail as the blue density overtakes the red density. This is numerically evident as the `t.threshold` is > than the `z.threshold` for a cumulative probability of 95% the 95th quantile.

let's zoom in on the right tail of the distribution with the `xlim` facet.

```
ggplot(zf.df, aes(x = Deviations, fill = Distribution)) +
  geom_density(alpha = 0.2) +
  geom_vline(aes(xintercept = z.threshold), color = "red", linetype = "dashed", size = 1) +
  geom_vline(aes(xintercept = t.threshold), color = "blue", linetype = "dashed", size = 1) +
  xlim(1, 5)
```



Interesting digression! But not really too far off the mark. The thresholds are the same with two standard risk measures, scaled for particular risk factors and positions. We have simulated two different *values at risk*.

## 4.6 Back to the Future

Let's remember where the returns (as changes) in each risk factor come from. Also, we will extract the last price for use below.

```
# Again computing returns as changes
# in the risk factors

return.series <- apply(log(price), 2, diff) ## compute the risk-factor change

price.last <- as.numeric(tail(price, n = 1)) ## reserve last price
```

Again to emphasize what constitutes this data, we specify the notional exposure. These are number of shares of stock, number of 1 dollars million contracts of futures, or volumetric contract sizes, e.g., MMBtus or boe. All of these work for us given the that price is dimensioned relative to the notional dimension.

So if the risk factors are oil and natural gas prices, then we should use a common volumetric equivalent such

as Btu (energy content) or boe (barrel of oil equivalent for volume). Position weights are then calculated as position times the last available price.

First, we can set the weights directly and a little more simply than before since we do not need to simulate historically.

```
## Specify the positions
positions.rf <- c(-30, 10) ## As before
## And compute the position weights
## directly again as before
```

```
(w <- position.rf * price.last)
```

```
## [1] -1072.5 1845.0
```

Second, we estimate the mean vector and the variance-covariance matrix, the two major inputs to the simulation of normal risk factor changes. Here we use a purely parametric approach.

```
mu.hat <- colMeans(return.series) ## Mean vector mu; estimated = hat
Sigma.hat <- var(return.series) ## Variance-covariance matrix Sigma
(loss.mean <- -sum(w * mu.hat))
```

```
## [1] -0.07596846
```

```
(loss.stdev <- sqrt(t(w) %*% Sigma.hat %*% w))
```

```
## [1]
```

```
## [1,] 28.4431
```

```
## Standard deviation of losses
```

Third we set the level of risk tolerance  $\alpha$ . Then let's calculate VaR and ES:

```
# Compute VaR and ES and return
alpha.tolerance <- 0.95
q.alpha <- qnorm(alpha.tolerance)
(VaR.varcov <- loss.mean + loss.stdev * q.alpha)
```

```
## [1]
```

```
## [1,] 46.70877
```

```
(ES.varcov <- loss.mean + loss.stdev * dnorm(q.alpha) / (1 - alpha.tolerance))
```

```
## [1]
```

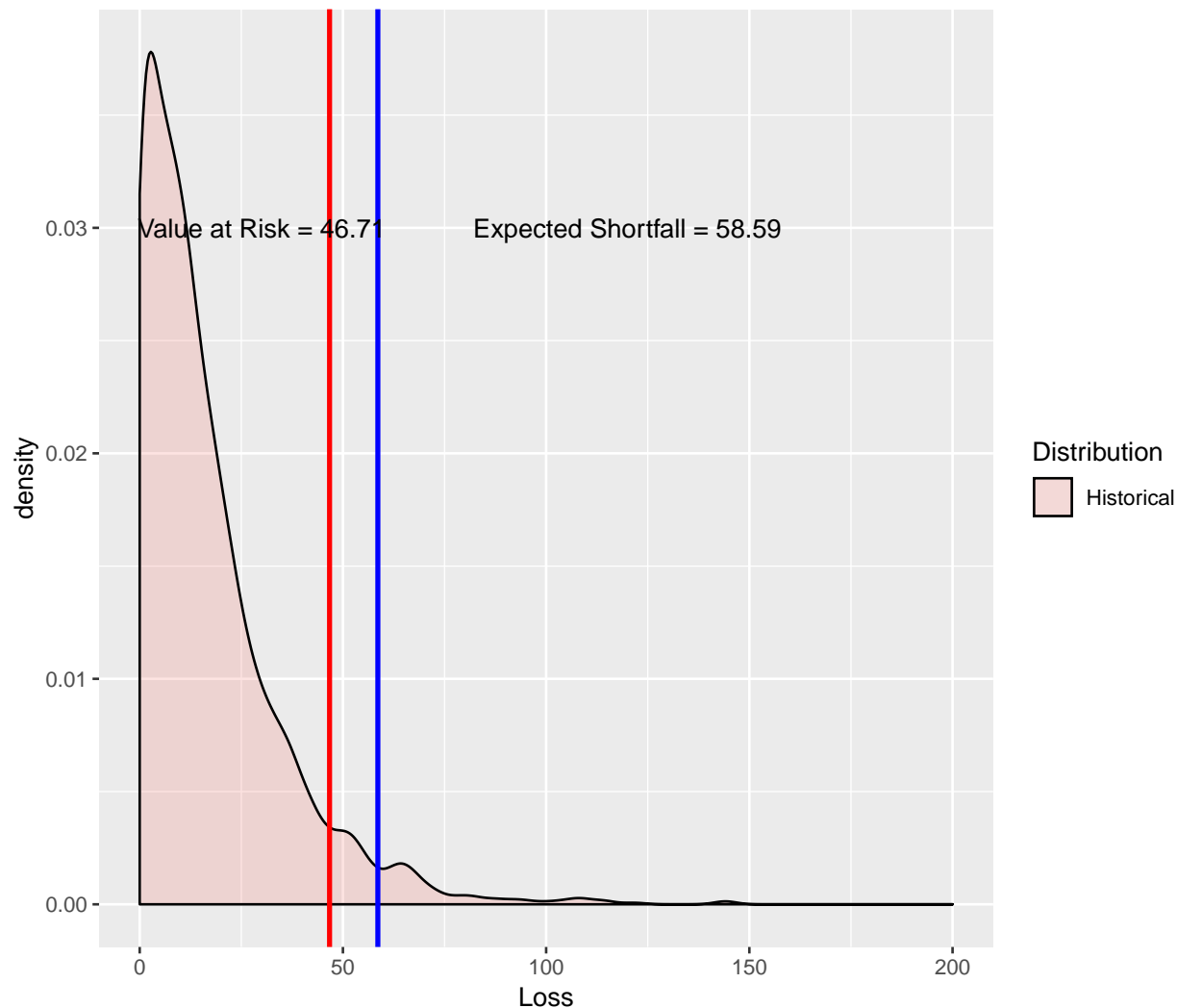
```
## [1,] 58.59398
```

and plot

```
VaR.text <- paste("Value at Risk =",
                  round(VaR.varcov, 2))
```

```
ES.text <- paste("Expected Shortfall =",
                 round(ES.varcov, 2))
```

```
ggplot(loss.rf.df, aes(x = Loss, fill = Distribution)) + geom_density(alpha = 0.2) +
  geom_vline(aes(xintercept = VaR.varcov), colour = "red", size = 1) +
  geom_vline(aes(xintercept = ES.varcov), colour = "blue", size = 1) +
  xlim(0, 200) +
  annotate("text", x = 30, y = 0.03, label = VaR.text) +
  annotate("text", x = 120, y = 0.03, label = ES.text)
```



## 4.7 Try this example

Suppose it takes less electricity to make steel than we thought above. We can model this by changing the position to  $(-20, 10)$ . Let's redo steps, 1, 2 and 3 (this begs for a function).

First we can set the weights directly a little more simply than before since we do not need to simulate history.

```
## Specify the positions
position.rf <- c(-20, 10)

## And compute the position weights
# directly again as before

(w <- position.rf * price.last)
```

```
## [1] -715 1845
```

Second, estimate the mean vector and the variance-covariance matrix, the two major inputs to the simulation of normal risk factor changes. Here we use a purely parametric approach.

```
mu.hat <- colMeans(return.series) ## Mean vector mu; estimated = hat
Sigma.hat <- var(return.series) ## Variance-covariance matrix Sigma
(loss.mean <- sum(w * mu.hat)) ## Mean loss
```

```
## [1] 0.1976962
```

```
(loss.stdev <- sqrt(t(w) %*% Sigma.hat %*%
  w )) ## Standard deviation of loss
```

```
##           [,1]
## [1,] 28.53755
```

Third, set the level of risk tolerance  $\alpha$ . Then calculate VaR and ES.

```
# Compute VaR and ES and return
alpha.tolerance <- 0.95
q.alpha <- qnorm(alpha.tolerance)
(VaR.varcov <- loss.mean + loss.stdev * q.alpha)
```

```
##           [,1]
## [1,] 47.13779
```

```
(ES.varcov <- loss.mean + loss.stdev * dnorm(q.alpha)/(1 - alpha.tolerance))
```

```
##           [,1]
## [1,] 59.06247
```

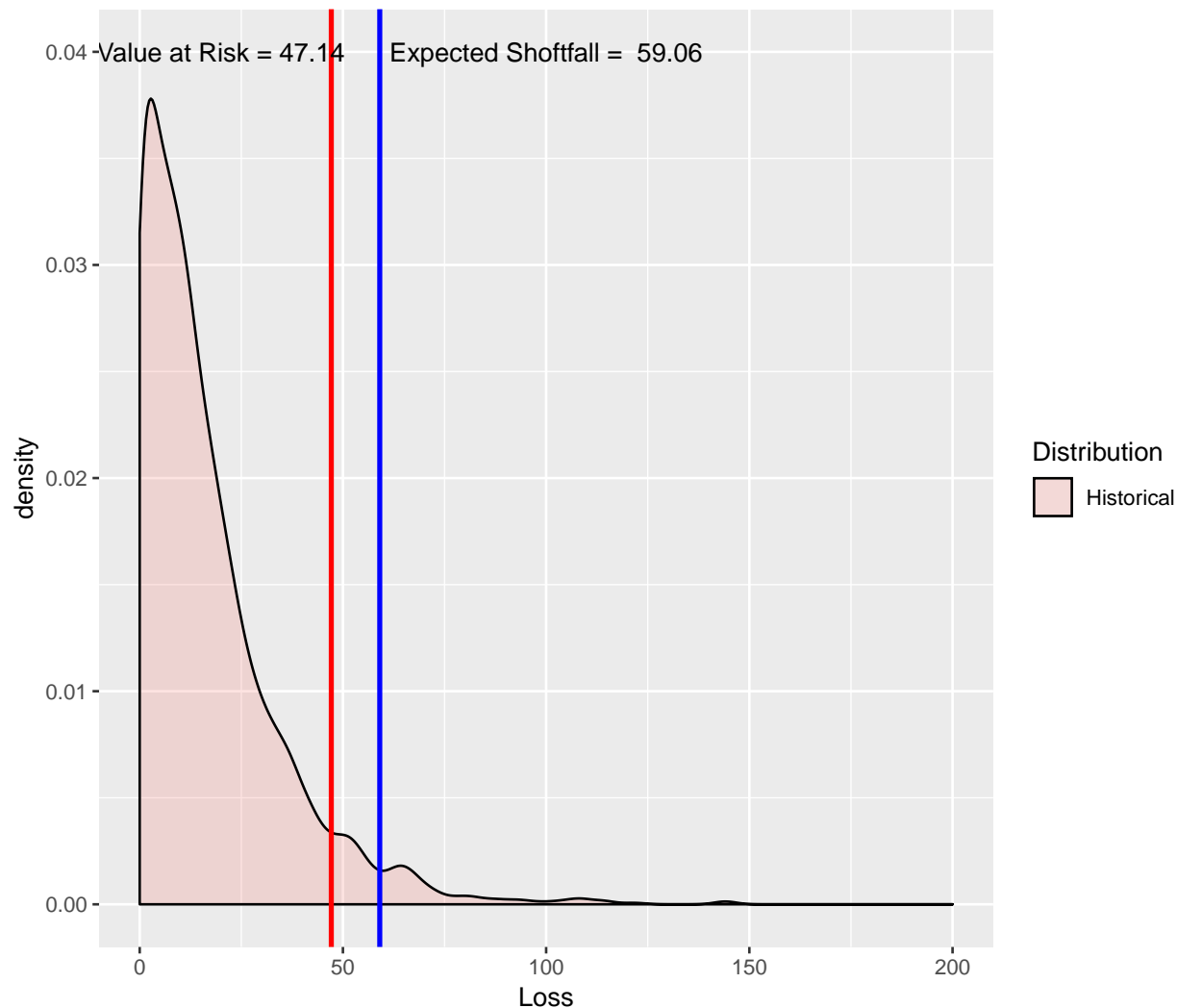
... and plot

```
VaR.text <- paste("Value at Risk =", round(VaR.varcov, 2))
```

```
ES.text <- paste("Expected Shoftfall = ", round(ES.varcov, 2))
```

```
ggplot(loss.rf.df, aes( x = Loss, fill = Distribution)) +
  geom_density(alpha = 0.2) +
  geom_vline(aes(xintercept = VaR.varcov), colour = "red", size = 1) +
  geom_vline(aes(xintercept = ES.varcov), color = "blue", size = 1) +
  xlim(0, 200) + annotate("text",
    x = 20,
    y = 0.04,
    label = VaR.text) +
  annotate("text",
    x = 100,
    y = 0.04,
    label = ES.text)
```





Aesthetics may overtake us here as we really should change the `x` and `y` `annotate` coordinates to fit on the graph properly.

So ends the story of the main method used for years and embodied in the famous 4:15pm risk report at JP Morgan. Also, we remember the loss that we simulate here is an operating income loss which after taxes and other adjustments, and say, a one year horizon, means a loss of additions to retained earnings. Book equity drops and so will market capitalization on average.

## 4.8 Let's go to Extremes

All along we have been stylizing financial returns, including commodities and exchanges, as skewed and with thick tails. We next go on to investigate these tails further using an extreme tail distribution called the Generalized Pareto Distribution (GPD). For very high thresholds, such as value at risk and expected shortfall, GPD not only well describes behavior in excess of the threshold, but 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 markets well exceeded all Gaussian and Student's thresholds.

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

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

and when the shape parameter  $\xi = 0$ , the GPD becomes the exponential distribution dependent only on the scale parameter  $\beta$ :

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

There is one reason for GPD's notoriety. 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 rate events begin. We will come back to this property when we look at operational loss data in a few chapters.

Let's use the `QRM` library to help us find the optimal fit of losses to the parameters. The `fit.GPD` function will do this for us.

```
library("QRM")

u <- quantile(loss.rf, alpha.tolerance,
              names = FALSE)

fit <- fit.GPD(loss.rf, threshold = u) ## Fit GDP to the excesses
(xi.hat <- fit$par.ests[["xi"]]) ## fitted xi

## [1] 0.1928437
(beta.hat <- fit$par.ests[["beta"]]) ## fitted beta

## [1] 15.89524

Now for the closed form (no random variate simulation!)

## Pull out the losses over the
## threshold and compute excess over
## the threshold

loss.excess <- loss.rf[loss.rf > u] - u ## Compute the excesses over u

n.relative.excess <- length(loss.excess) / length(loss.rf) ## = N_u/n
(VaR.gpd <- u + (beta.hat/xi.hat) * (((1 - alpha.tolerance)/n.relative.excess)^(-xi.hat) - 1))

## [1] 40.40925
(ES.gpd <- (VaR.gpd + beta.hat - xi.hat * u) / (1 - xi.hat))

## [1] 60.11707
```

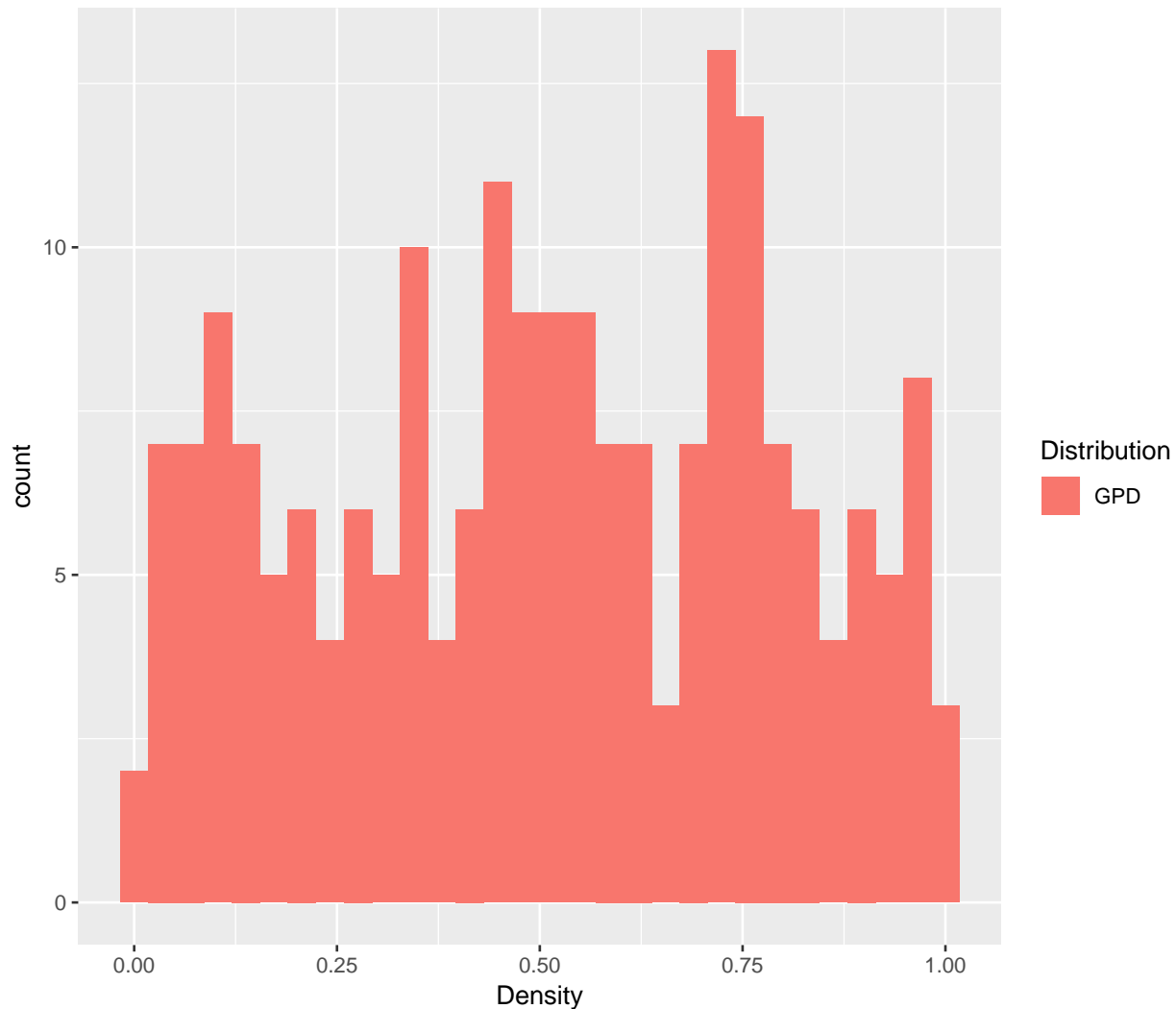
#### 4.8.1 Example

How good a fit to the data have we found? This plot should look roughly uniform since the GPD excess loss function is a linear function of thresholds  $u$ .

```
gpd.density <- pGPD(loss.excess , xi = xi.hat, beta = beta.hat)

gpd.density.df <- data.frame(Density = gpd.density,
                             Distribution = rep("GPD", each = length(gpd.density))) # This should be U[

ggplot(gpd.density.df, aes(x = Density, fill = Distribution)) + geom_histogram()
```



And it does look “uniform” enough (in a statistical sort of way as we perform eyeball econometrics again!).

## 4.9 All Together Now

Let’s graph the historical simulation, variance-covariance and GPD results together.

```
loss.plot <- ggplot(loss.rf.df, aes(x = Loss, fill = Distribution)) + geom_density(alpha = 0.2)

loss.plot <- loss.plot + geom_vline(aes(xintercept = VaR.varcov), color = "red",
                                   linetype = "dashed",
                                   size = 1)

loss.plot <- loss.plot + geom_vline(aes(xintercept = ES.varcov), colour = "blue",
```

```

linetype = "dashed", size = 1)

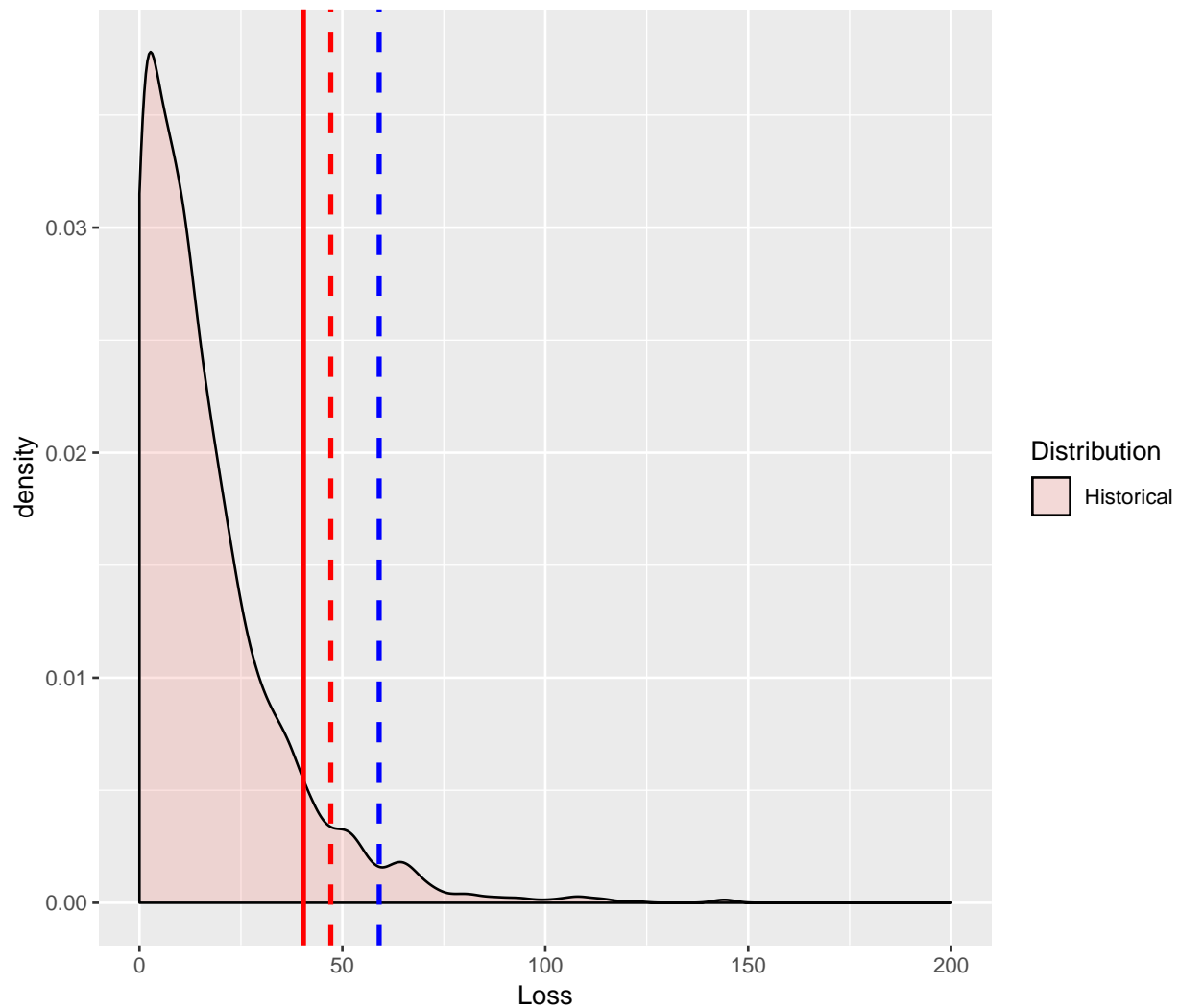
loss.plot <- loss.plot + geom_vline(aes(xintercept = VaR.gpd),
                                   colour = "red",
                                   size = 1)

loss <- loss.plot + geom_vline(aes(xintercept = ES.gpd), colour = "blue",
                              size = 1)

loss.plot <- loss.plot + xlim(0,200)

loss.plot

```



That was a lot. We will need our “mean over excess” knowledge when we get to operational risk. Actually, we will be able to apply that to these risk measures for any kind of risk. But we will save ourselves for operational risk later. Someone might even annotate the graph...

## 5 Credit Risk (Missing)

## 6 Operational Risk and Extreme Finance

### 6.1 Imagine this

Remember that company we just acquired? Not only is customer creditworthiness apt to cost us another \$80 million, but our walk-through of distribution, call-center, and production facilities had a raft of negatively impacting issues with health and safety, environmental, and intellectual property all located in places rife with fraud and corruption.

Now only that, but (5 years ago!) hurricane damage has still not been rectified. Our major insurance carrier has also just pulled out of underwriting your fire-related losses. To seemingly top it all off, three Bp's of regional manufacturing have just been indicted for padding vendor accounts for kickbacks. To add insult to injury, employee turnover rates are over 40%

Previously we studied the stylized facts of financial variables, market risk, and credit risk. A common theme runs through this data and outcomes: thick tails, high Value at Risk and Expected Shortfall. Rare events can have strong and persistent effects. Then there is the probability of contagion that allows one bad market to infect other markets.

Now we need to consider more formally the impact of extreme, but rare events on financial performance. To do this we will start off with modeling potential loss in two dimensions, frequency of loss and severity. Frequency of losses, modeled with a Poisson process. The Poisson process is at the heart of Markov credit risks. Each transition probability stems from an average hazard rate of defaults in a unit of time and a segment of borrowers. Here we will use a simplified sampling of frequencies of events drawn from a Poisson process similar to the one we used in credit risk measurement.

Severity is defined as currency based loss. In credit this is the unrecoverable face value, accrued interest, and administrative expenses of a defaulted credit instrument such as a loan. In operational risk this loss could be the value of customers lost due to a cyber breach, or suppliers failing to deliver critical goods and services resulting in unmet demand, or reputation with investors leading to a flight of capital. We will typically use gamma or generalized Pareto distributions to help us model severity. In some equity markets, the log-normal distribution is prevailing practice.

- Mean Excess Loss from reliability and vulnerability analysis
- Historical data
- VaR and ES again

### 6.2 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.

In a nutshell we measure operational risk using **frequency** and **severity**. Again there is analogy to credit risk where the frequency is the probability of default and severity is recoverable exposure. 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.

On the other hand we think of **severity** as the monetary loss itself. This loss is beyond the **threshold**, and thus the focus on “extreme” value methods of dealing with distributions. The “extreme” will be the tail of the distribution where probably but rare events with large magnitudes roam.

In typical experience in most organizations, there is no good time series or cross-section of data on which to rely for measurements of operational risk. For example, the emerging and immanent 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, and often seem far too much an estimate of enterprise-wide loss. Where we do have some data, we have established ways of working with severity distributions that make will sense of the shape of the data.

### 6.2.1 Example

Suppose management, after much discussion and challenge by subject matter experts, determines 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 we quantify this risk?
2. How would we manage this risk?

Here are some steps management might take to measure and mitigate this operational risk:

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.
4. Aggregate and report ranges of median and quantile potential loss.

This last step puts a “value” on operational risk scenarios using a rate of error that, if exceeded, would put the organization at ongoing risk that management is not willing to tolerate. This “significance level” in data analytics is often called “alpha” or  $\alpha$ . An  $\alpha = 5$  percent indicates that management, in a time frame such as one year, is not willing to be wrong more than 1 in 20 times about its experience of operational risk. Management believes that are 95 percent ( $1 - \alpha$ ) confident in their measurements and management of risk. It is this sort of thinking that is at the heart of statistically based confidence and hypothesis testing.

## 6.3 How Much?

Managers can use several probability distributions to express their preferences for risk of loss. A popular distribution is the **gamma** severity function. This function allows for skewness and “heavy” tails. Skewness means that loss is attenuated and spread away from the mean or media, stretching occurrences further into the tail of the loss distribution.

the gamma distribution is fully specified by shape,  $\alpha$  and scale,  $\beta$ , parameters. The shape parameter is a dimensionless number that affects the size and skewness of the tail. The scale parameter is a measure of central tendency “scaled” by the standard deviation of the loss distribution. Risk specialists find this distribution to be especially useful for time-sensitive losses.

We can specify the  $\alpha$  shape and  $\beta$  scale parameters using the mean,  $\mu$ , and standard deviation,  $\sigma$  of the random severity,  $X$  as

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

and

$$\alpha = \mu\beta$$

and this is the same as

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

Here  $\beta$  will have dimension of  $LOSS^{-1}$ , while  $\alpha$  is dimensionless. The probability that a loss will have exactly  $X$ , conditional on  $\alpha$  and  $\beta$  is

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

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

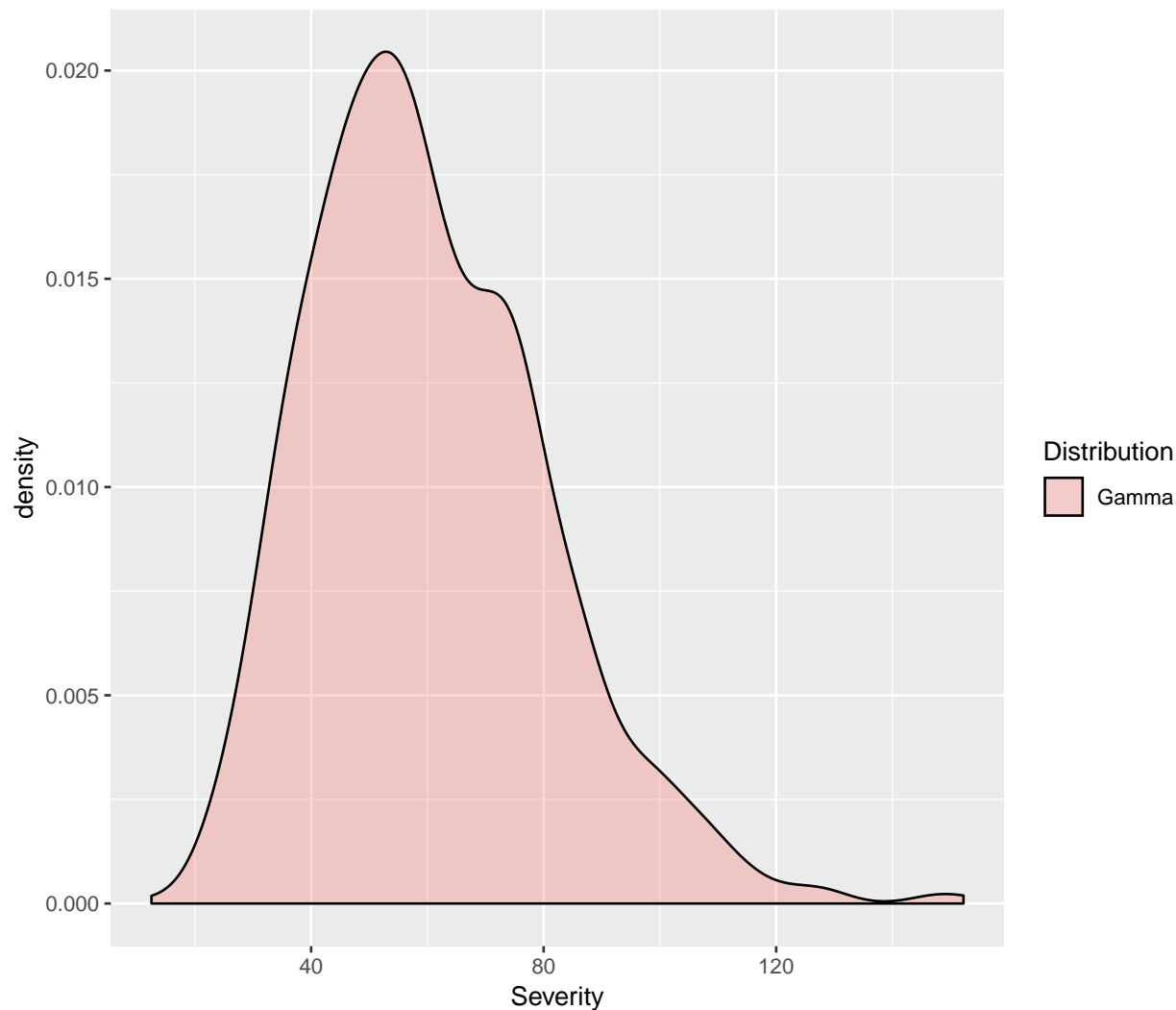
mu <- 60 ## The mean
sigma <- 20 ## management's view of how certain they think their estimates are
sigma.sq <- sigma^2
beta <- mu/sigma.sq
alpha <- beta * mu
severity <- rgamma(n.sim, alpha, beta)
summary(severity)
```

```
##      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 check with management that this low and high (and intermediate levels) are reasonable. Let's graph the distribution. Here we form a data frame in `gamma.sev` to create factor values for the `ggplot` fill.

```
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)
```



### 6.3.1 Severity thresholds

We can add thresholds based on Value at Risk (VaR) and expected shortfall ES. Here we calculate `VaR.sev` as the quantile of a confidence level of  $1 - \alpha$ .

```
alpha.tolerance <- 0.05
(VaR.sev <- quantile(severity, 1 - alpha.tolerance))

##      95%
## 98.05779

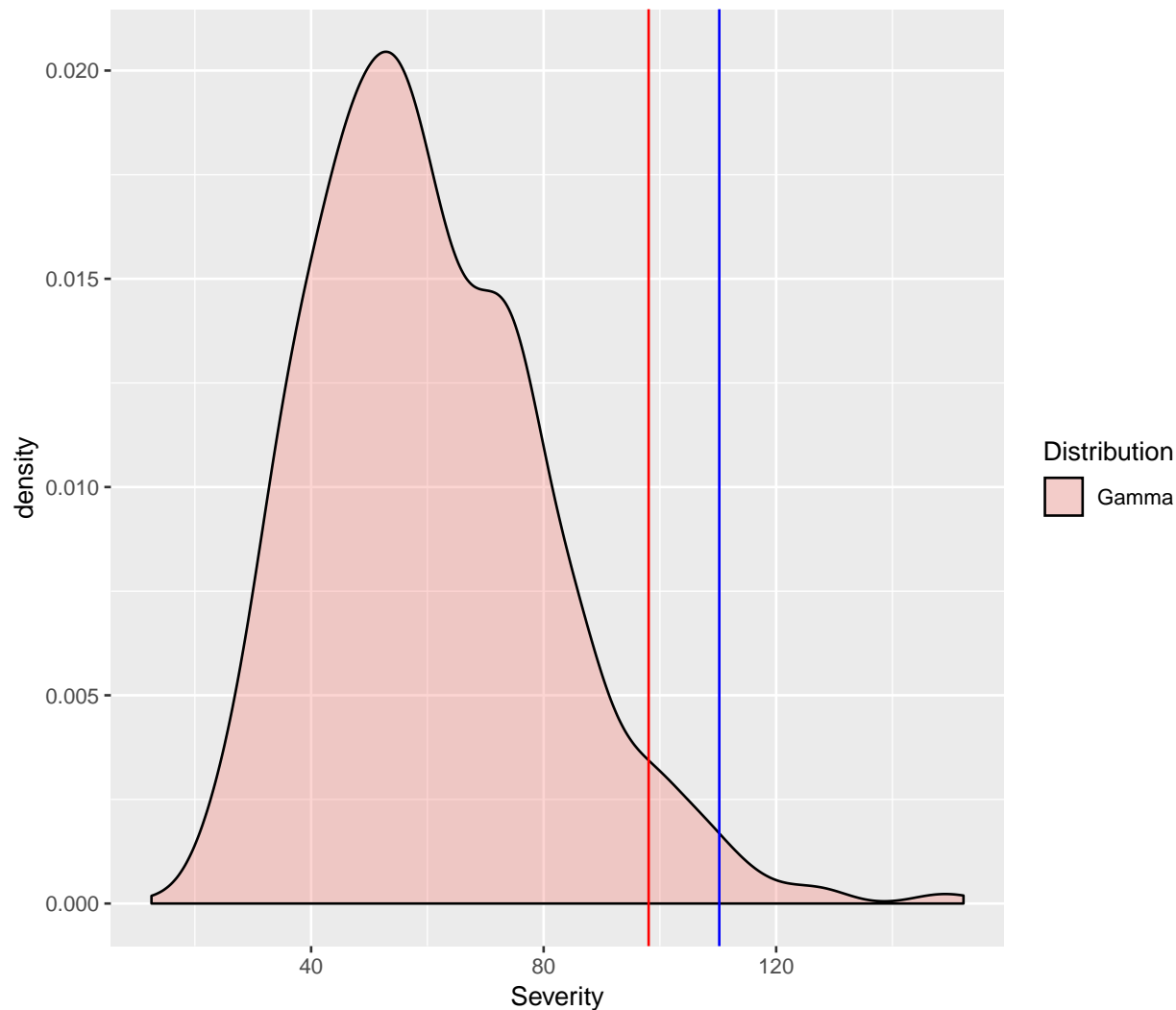
(ES.sev <- mean(gamma.sev$Severity[gamma.sev$Severity > VaR.sev]))

## [1] 110.2134
```

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")
```





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 proportional to `ES.sev - quantile(severity, 0.50)`, that is the difference between the expected shortfall and the median potential loss. The modifier “proportional” needs some further specifications. We have to consider the likelihood of such a risk event.

## 6.4 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.  $\lambda$  in a unit of time. This rate is just what we struggled to estimate for credit rating transitions.

Again for completeness sake and as a reference, we define the probability of exactly  $x$  risk events arriving at a rate of  $\lambda$  as

$$\text{Probability}[x|\lambda] = \frac{\lambda^x e^{-\lambda}}{x!}$$

Management might try to estimate the count of events in a unit of time using hazard rate models (same  $\lambda$  as

in credit migration transitions). 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 once every four months).

We simulate by using `n.sim` trials drawn with the `rpois()` function conditional on  $\lambda = 0.25$ .

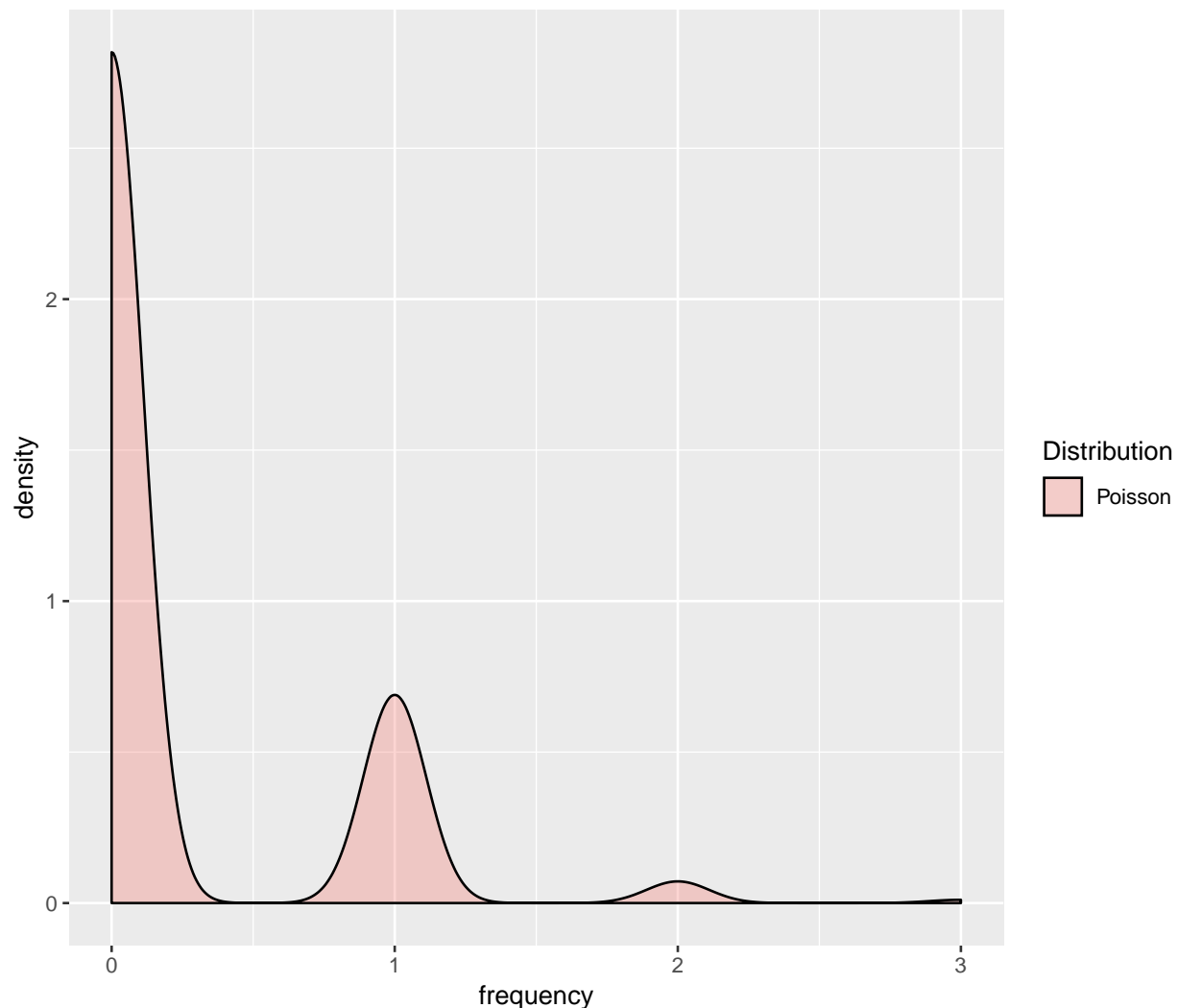
```
n.sim <- 1000
lambda <- 0.25
frequency <- rpois(n.sim, lambda)
summary(frequency)
```

```
##      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 four, 3-month event intervals.

## 6.5 How Much Potential Loss?

Now we get to the nub of the matter, how much potential 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 many `poisson` frequency scenarios.

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) 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 data. Then calculate the risk measures for the potential loss as

```
alpha.tolerance = 0.95
```

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

(VaR.loss.rf <- quantile(loss.rf$Loss,
                        1 - alpha.tolerance))
```

```
## 5%
## 6
```

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

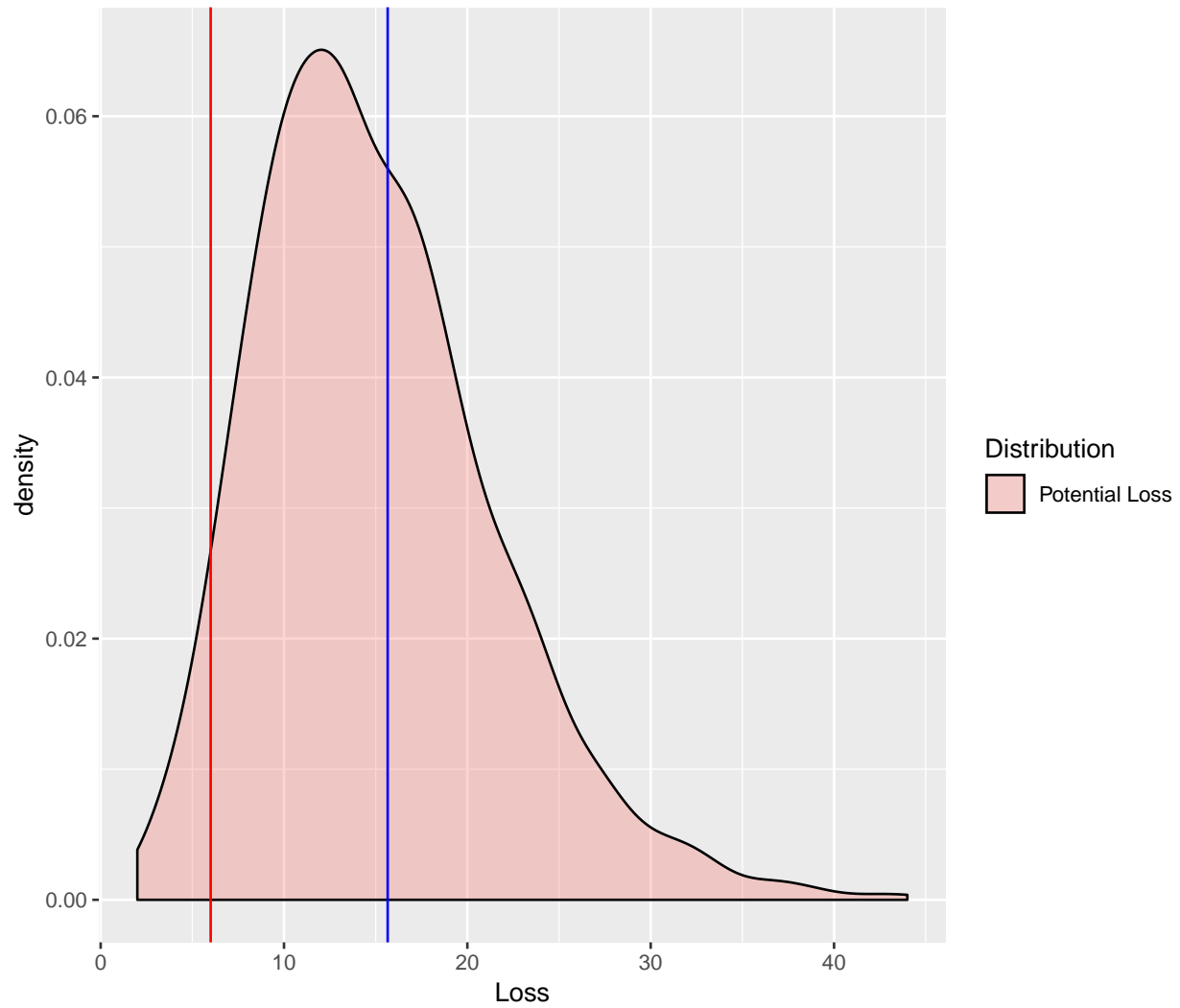
```
## [1] 15.65462
```

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

### 6.5.1 Example

1. Plot the loss fucniton and risk measures with

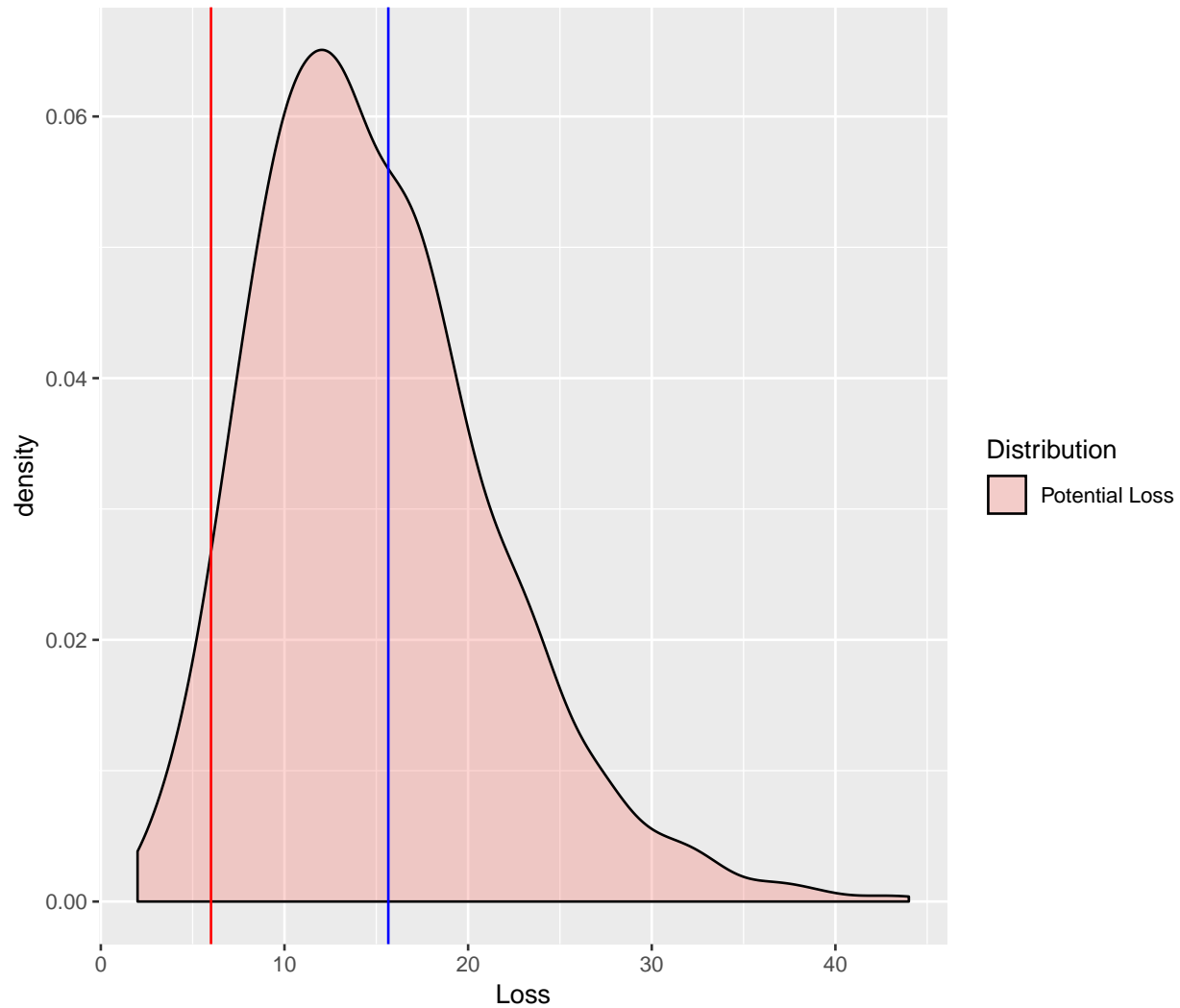
```
ggplot(loss.rf, aes(x = Loss, fill = Distribution)) + geom_density(alpha = 0.3) +
  geom_vline(xintercept = VaR.loss.rf, color = "red") + geom_vline(xintercept = ES.loss.rf, color = "blue")
```



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

Some results using this code:

```
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 the organization can decide on the difference between the VaR (or ES) and the median loss.

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

```
(VaR.loss.rf <- quantile(loss.rf$Loss, 1 - alpha.tolerance))
```

```
## 5%
```

```
## 6
```

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

```
## [1] 15.65462
```

```
(Reserve.loss.rf <- ES.loss.rf - quantile(loss.rf$Loss, 0.5))
```

```
## 50%
```

```
## 1.654623
```

If this were a bank, then managers would have calculated the capital requirement (Basel III)

## 6.6 We Have History

Now suppose we have some history of losses. In this context, we use an extreme tail distribution called the Generalized Pareto Distribution (GPD) to estimate historical loss parameters. This distribution models the tails of any other distribution. We would have split a `poisson-gamma` loss distribution into two parts, a body and a tail. The tail would begin at a specified threshold.

The GPD is especially well-known for a single property: for very high thresholds, GPD not only well describes the behavior in excess of the threshold, but the mean excess over the threshold is linear in the threshold. From this property 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 in distressed markets.

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

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

and when the shape parameter  $\xi = 0$ , the GPD becomes the exponential distribution dependent only on the scale parameter  $\beta$ :

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

Now for the infamous property. If  $u$  is an upper (very high) threshold, then the excess of the 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. We will use as a threshold to exceedance that begins to make the excesses linear.

## 6.7 Fire Losses

Now to get some data. The first thing we do is load the well researched Danish fire claims data set in millions of Danish Kroner collected from 1980 to 1990 as a time series object. we will plot the Mean Excesses (of thresholds). These are simply the mean of  $e(u)$ , a function of the parameters of the GPD. We will follow very closely the code and methodology used in the `qrmtutorial`

```
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
```

```
head(danish, n = 3)
```

```
## GMT
##           FIRE.LOSSES
## 1980-01-03    1.683748
## 1980-01-04    2.093704
## 1980-01-05    1.732581
```

```
tail(danish, n = 3)
```

```
## GMT
##           FIRE.LOSSES
## 1990-12-30    4.867987
## 1990-12-30    1.072607
## 1990-12-31    4.125413
```

```
## Set Danish to a numeric series
## object if a time series
```

```
if (is.timeSeries(danish)) danish <- series(danish)
danish <- as.numeric(danish)
```

We review the dataset.

```
summary(danish)
```

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

Our next task is to sort out the losses and rank order unique losses with this function.

```
n.danish <- length(danish)
## Sort and rank orer 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 use the `rank.series` 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 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. Let's spell this out again:

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.

## 6.8 Example

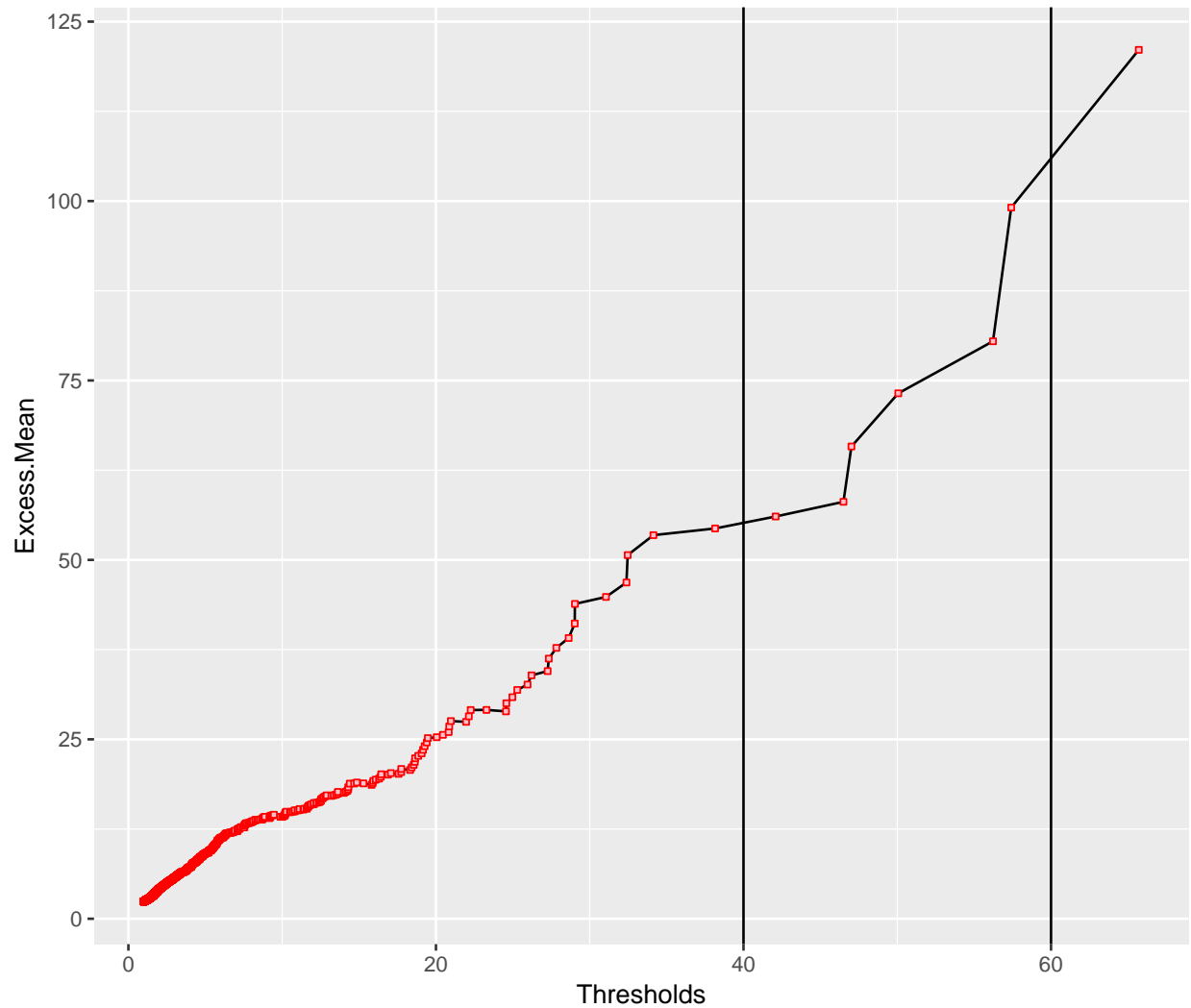
```
library(ggplot2)

omit.points <- 3
excess.mean.df <- data.frame(Excess.Mean = excess.mean[1:(n.points - omit.points)], Thresholds = 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 3 mean excess data points.

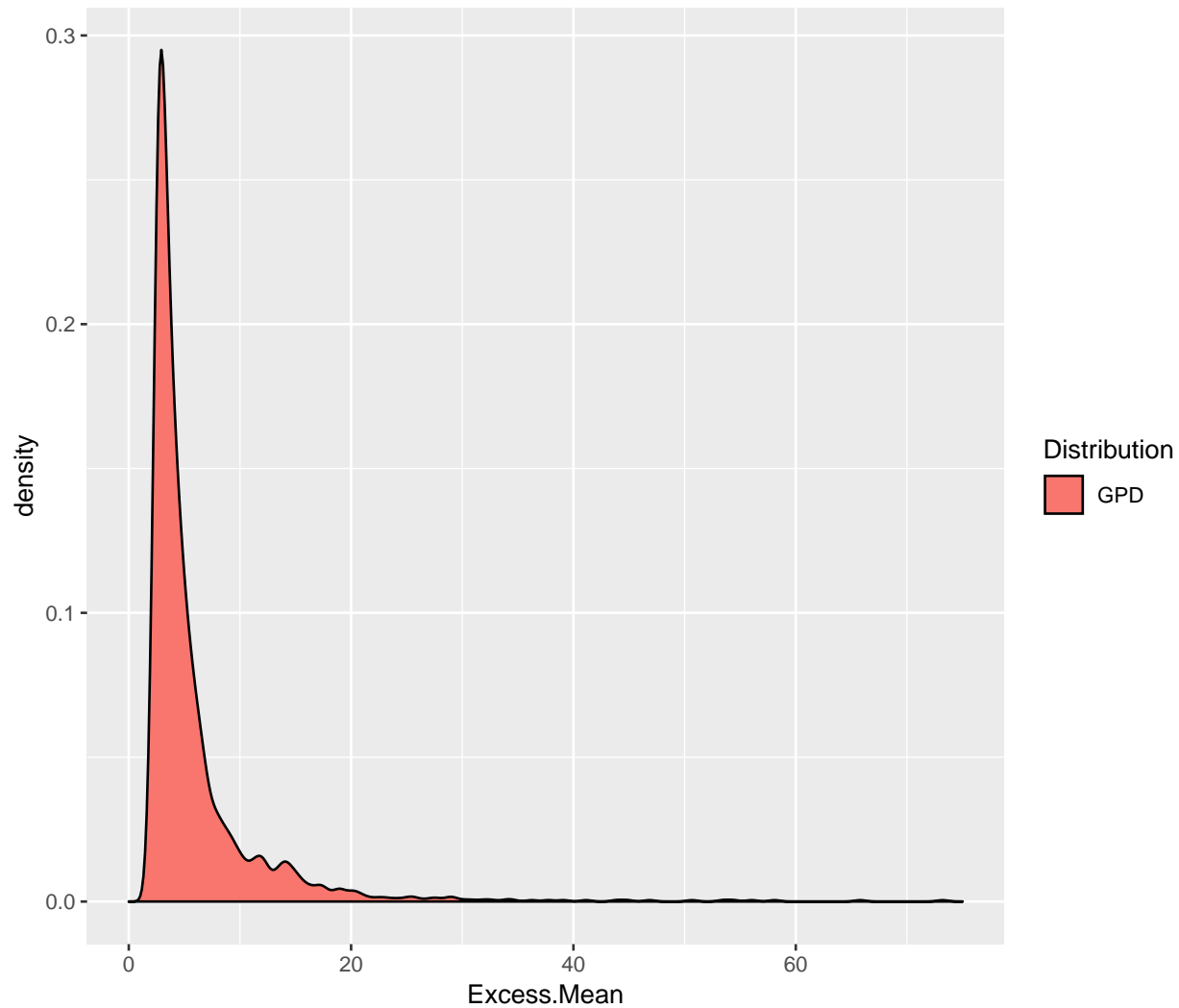
```
## Mean Excess plot
ggplot(excess.mean.df, aes(x = Thresholds,
                           y = Excess.Mean)) + geom_line() +
  geom_point(size = 1, shape = 22, colour = "red", fill = "pink") +
  geom_vline(xintercept = 40) + geom_vline(xintercept = 60)
```





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 rare events we might want to mitigate. We might look at the vertical line at 40 and 60 million kroner to possibly design retention and limits for a contract to insure against experiencing loss in this region.

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



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.

## 6.9 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.

```
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.se[["xi"]]) ## fitted xi

## [1] 0.1351274
```

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

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

Let's put these estimates to good use. Here are the closed form calculations for value at risk and expected shortfall (no random variate simulations!) using the formulae in chapter 5:

```
## 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))
```

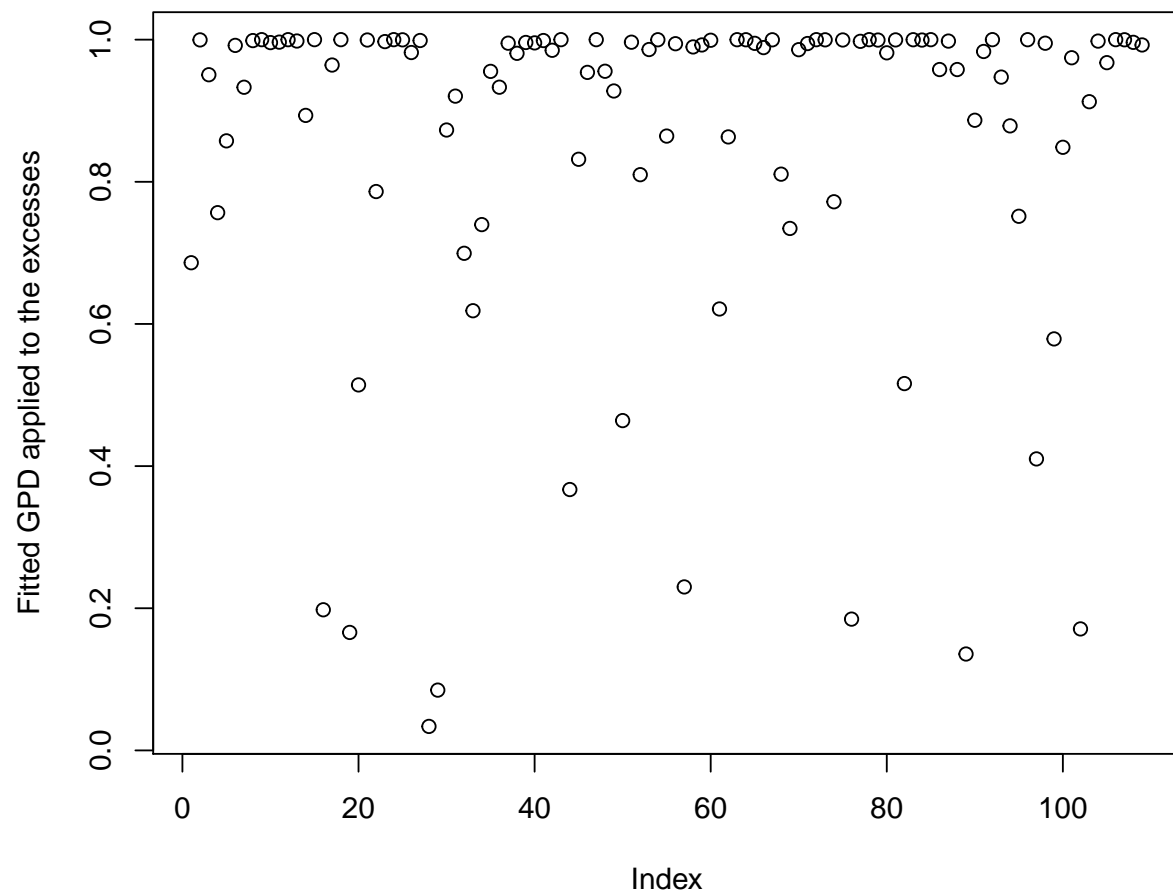
```
## [1] 11.27284
```

Using these risk measures we can begin to have a discussion around the size of reserves, the mitigation of losses, and the impact of this risk on budgeting for the allocation of risk tolerance relative to other risks.

How good a fit? Let's run this code next.

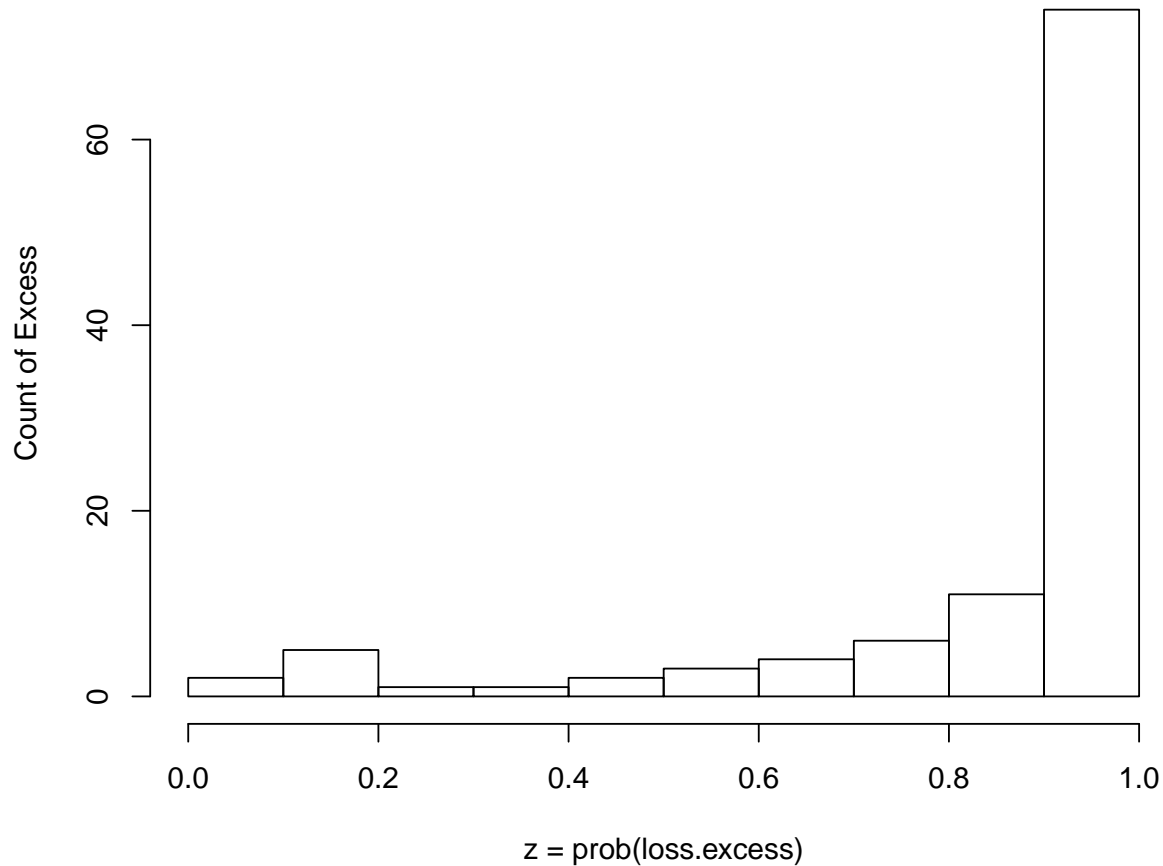
```
z <- pGPD(loss.excess, xi = xi.hat.danish, beta = beta.hat.danish) ## should be U[0,1]
```

```
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 it is as most of the excess are in the tail

### 6.10 Example

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

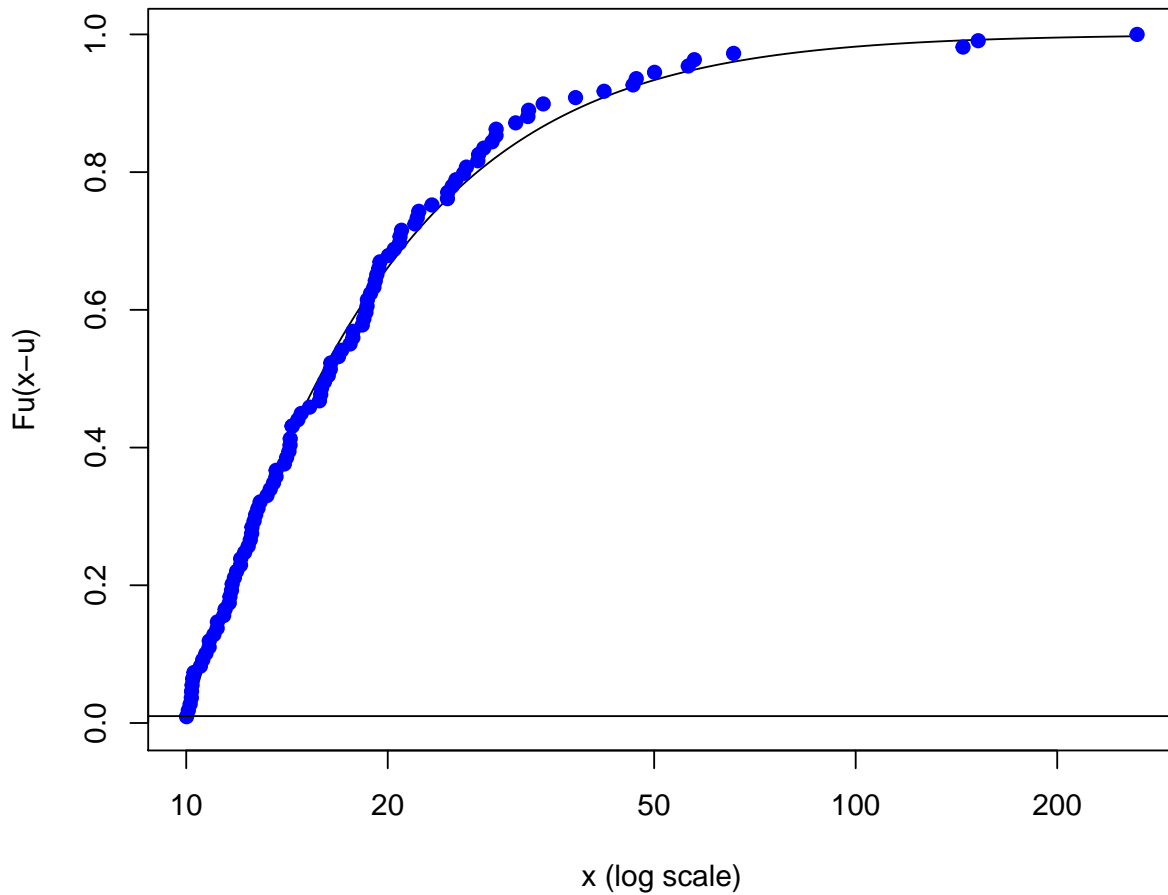
```
## Fit GPD model above 10

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
```

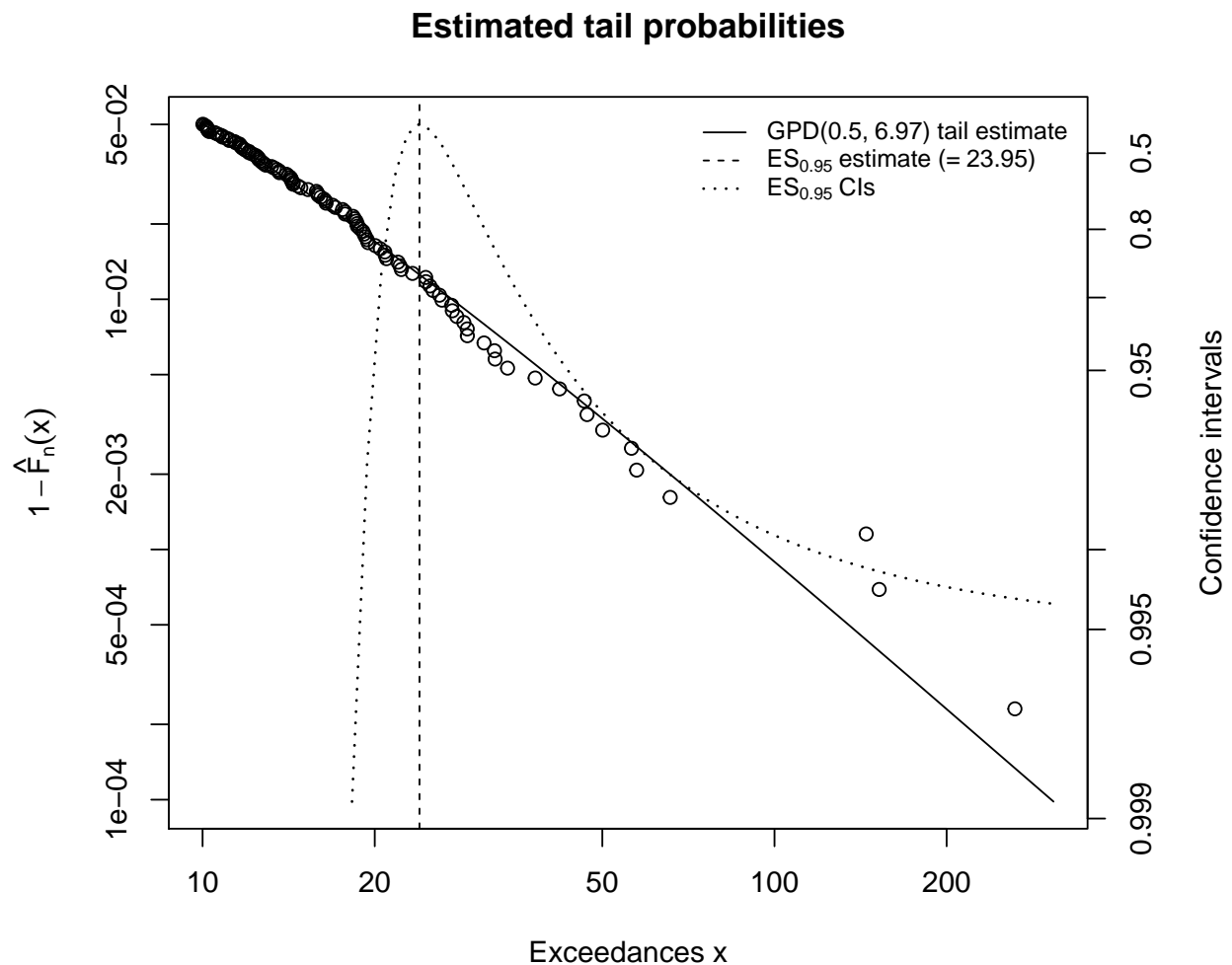
Now let's look at the whole picture with these exceedances. On the vertical axis we have the probabilities that you are in the tail ... the extreme loss of this operation. On the vertical axis we have the exceedances (loss over the threshold  $u$ ). The horizontal line is our risk tolerance (this would be  $h$  percent of the time we don't want to see loss...)

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



Let's use the `shorRM` function in the `QRM` package to calculate the expected short-fall confidence intervals. The initial `alpha.tolerance` is `1 - h` risk tolerance. We are only looking at the expected shortfall and could also have specified value at risk. What we really want to see is the range of ES as we wander into the deep end of the pool with high exceedances.

```
alpha.tolerance <- 0.05
showRM(fit.danish, alpha = 1 - alpha.tolerance, RM = "ES", method = "BFGS")
```



CI stands for “confidence interval” and we answer this question:

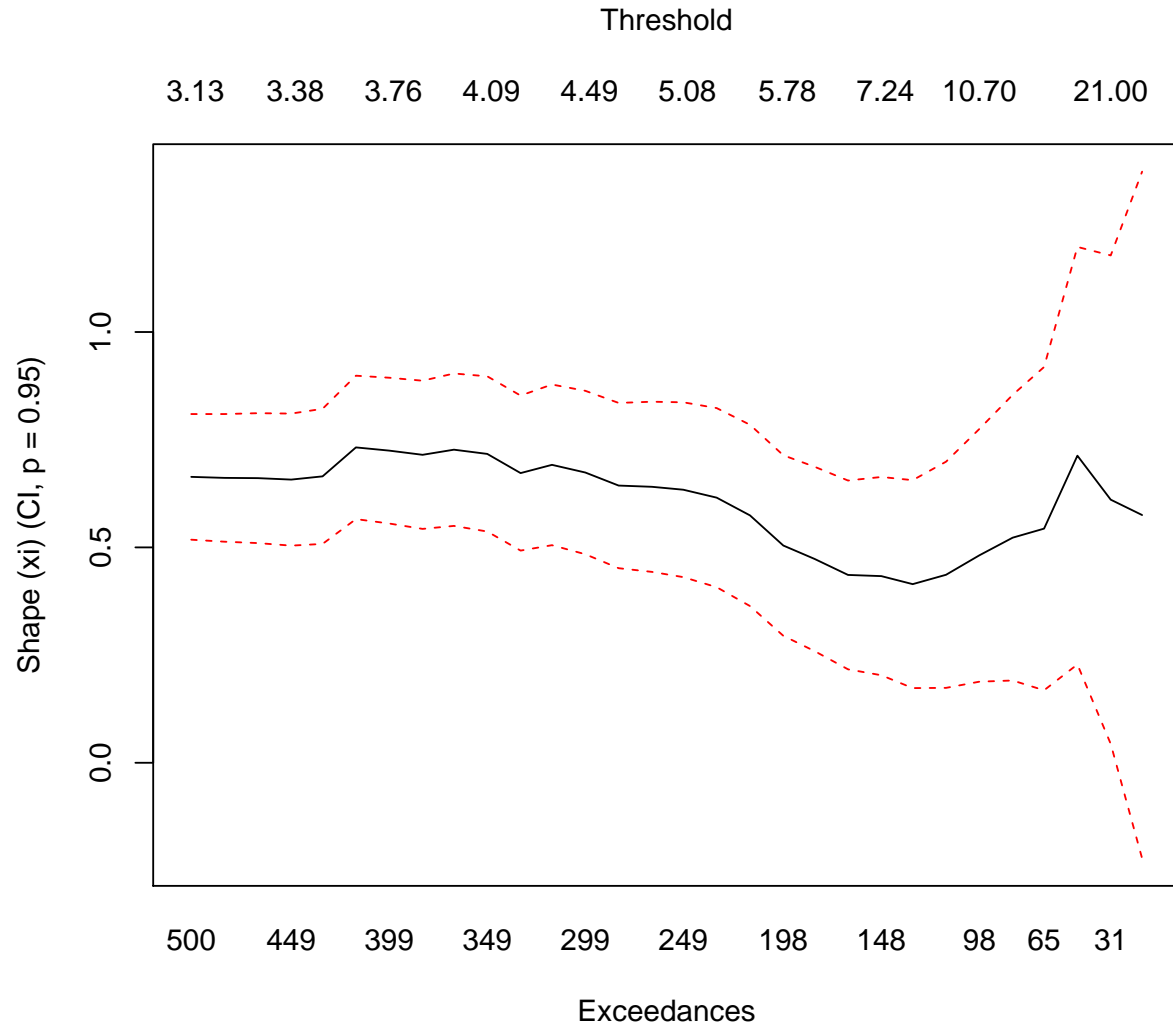
For each level of tolerance for loss (vertical axis) what is the practical lower and upper limits of the expected shortfall of 23.95 million?

Practically speaking how much risk would managers be willing to retain (lower bound) and how much would you underwrite (upper - lower bounds)?

#### 6.10.1 The shape of things to come

Let's now plot the GPD shape parameter as a function of the changing threshold.

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



What does  $\xi$  tell us? Mainly information about the relationship between our risk measures. The ratio of VaR to ES is  $(1 - \xi)^{-1}$  if  $0 \leq \xi \leq 1$ . How bent the tail will be: higher  $\xi$  means a heavier tail, and a higher frequency of very large losses.

Again the upper and lower bounds help us diagnose what is happening to our exceedances. The Middle line is the shape parameter at various thresholds and their corresponding exceedances. Dashed red lines are the “river banks” bounding the upper and lower edges of the tail (by threshold) distribution of the estimated risk measure. Fairly stable?

### 6.10.2 Example

We can run this code to throw the two popular risk measures together

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

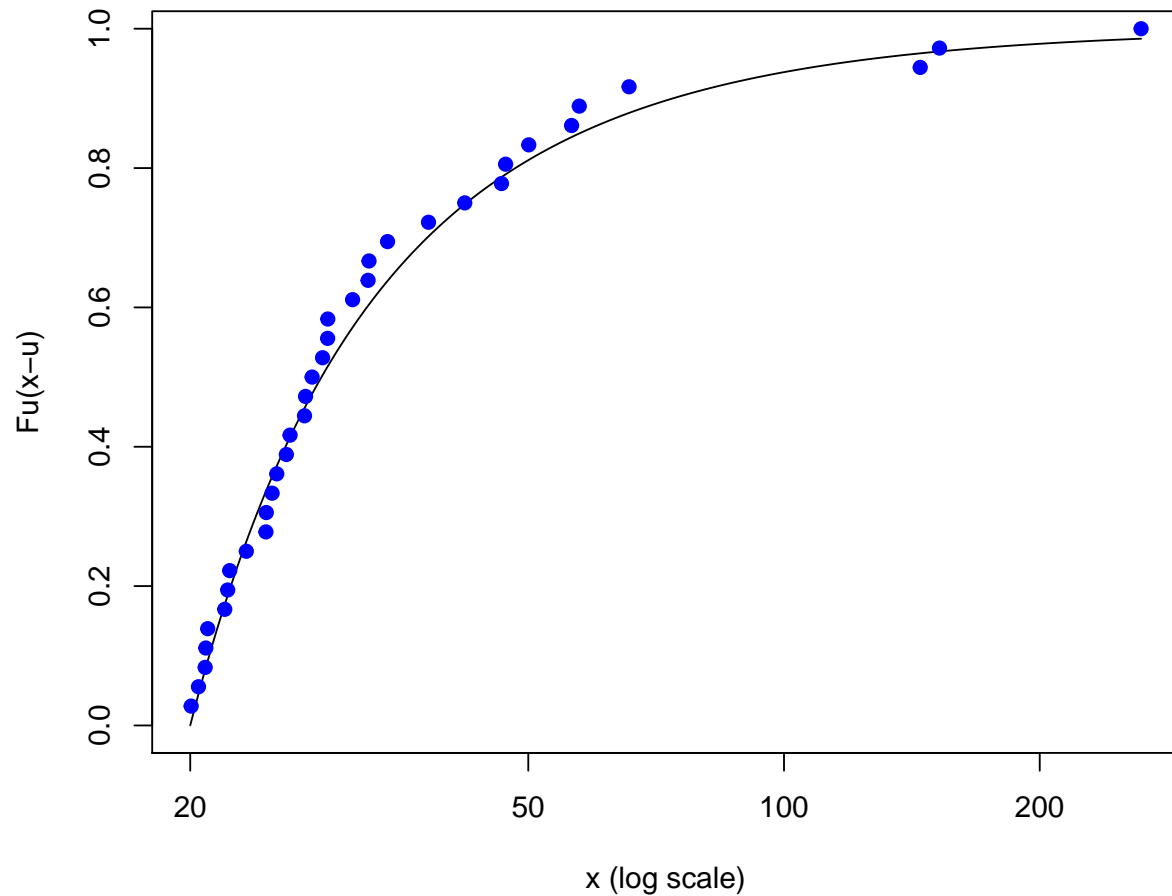
```
##          xi          beta
## 0.6844366 9.6341385
```



```
mod2$par.ses
```

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

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



Let's interpret across various threshold regions:

1. From 20 to about 40, fairly dense and linear
2. From about 40 to 60, less dense and a more bent slope ( $\xi$  is bigger than for lower threshold)
3. Big and less frequent outliers from about 50 to well over 200

Some managerial implications to spike the discussion could include:

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
```

Here `df` is the degree of freedom for 2 estimated parameters. The values are very low, meaning there is a very small chance that the estimates are zero. Various risk measures and tail plots can elucidate more interpretation. Here we can use two confidence levels.

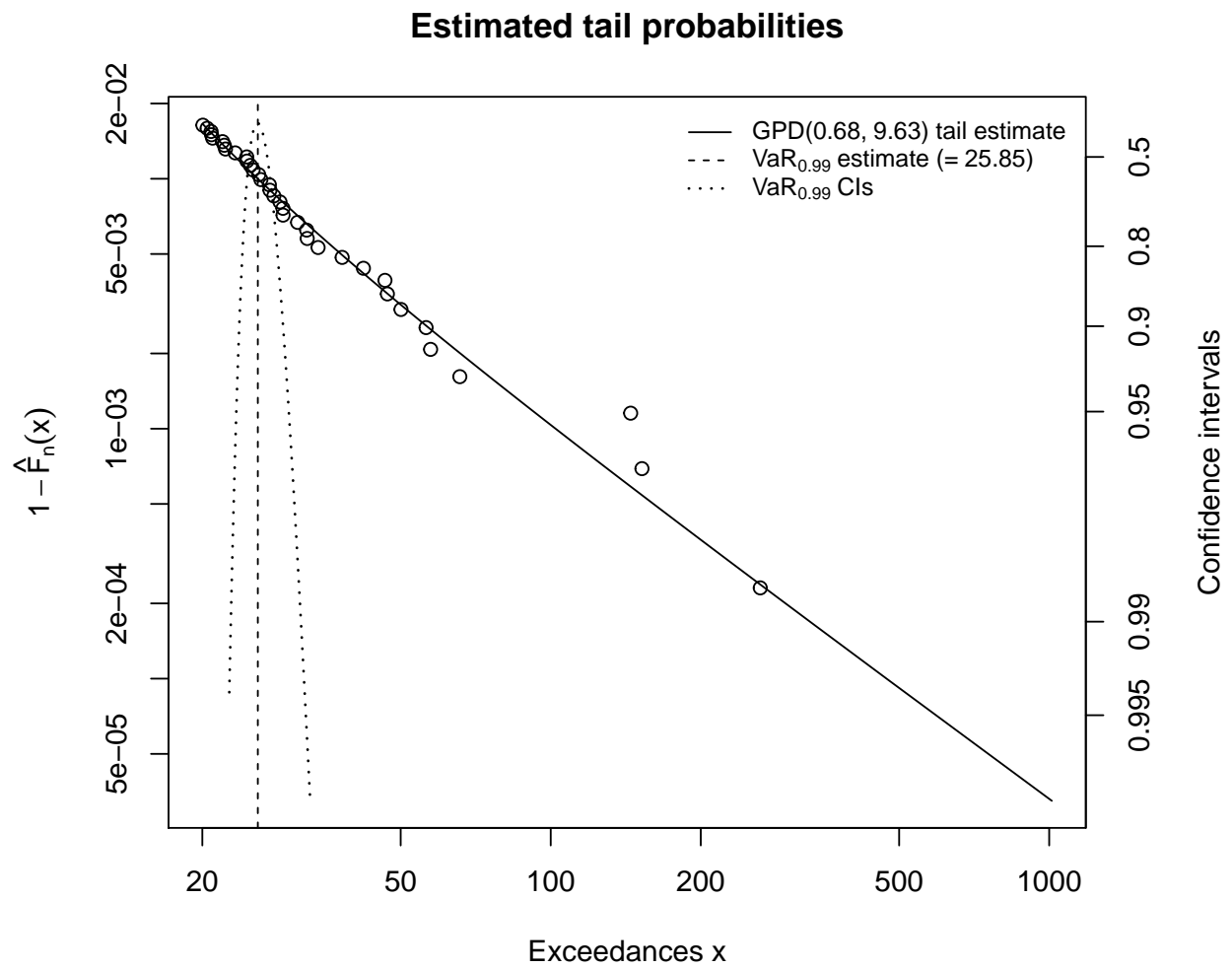
```
(RMs2 <- RiskMeasures(mod2, c(0.99, 0.995)))
```

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

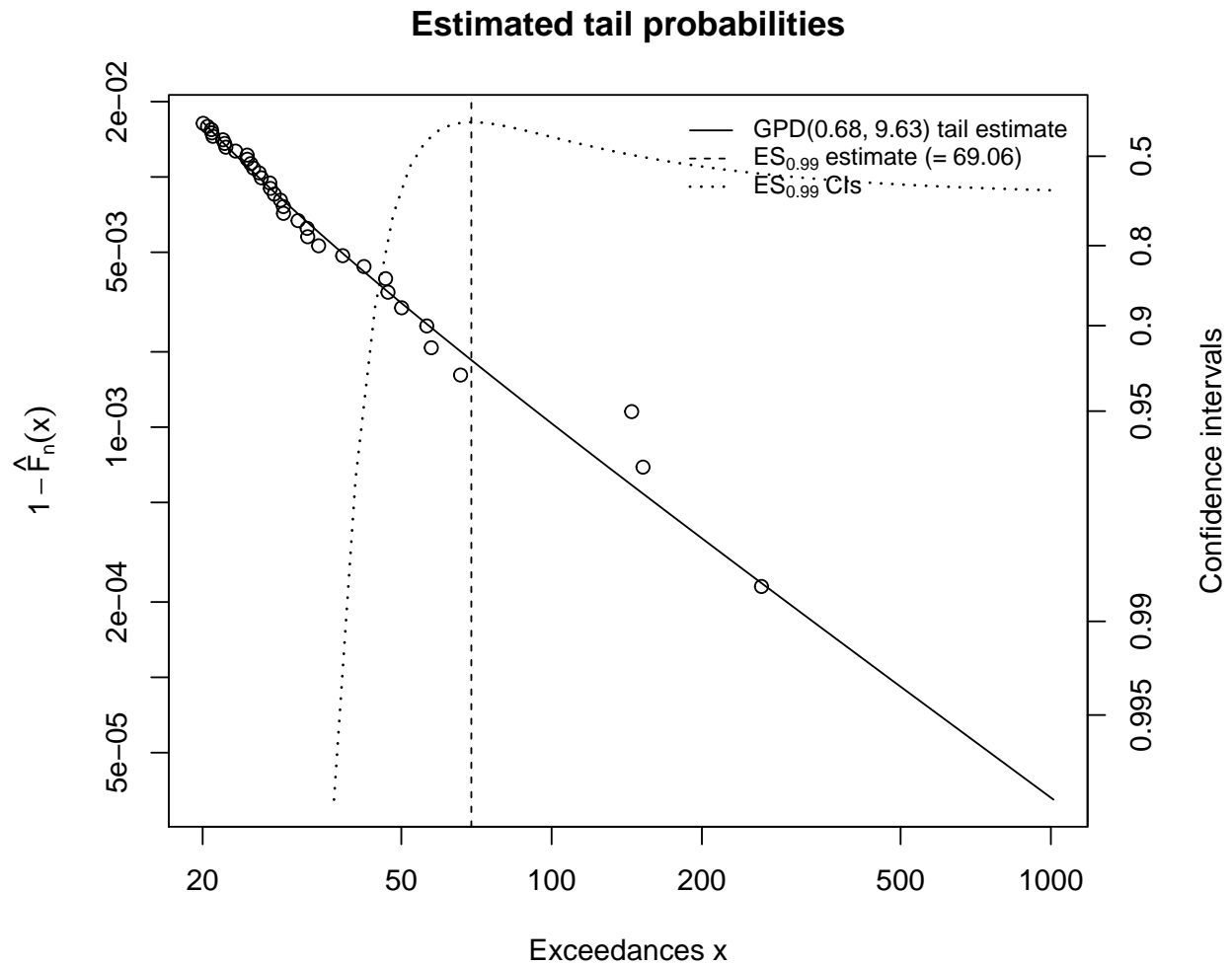
```
RMs2
```

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

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



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



## 6.11 Summary

Operational risk is “extreme” finance with (of course) extreme value distributions, methods from reliability and vulnerability analysis thrown in for a good measure, and numerous regulatory capital regulations. We just built both simulation and estimation models that produced data driven risk thresholds and an operational nature. That nature means:

- Heavy tail loss severity distributions,
- Often with frequencies that may vary considerably over time, and
- Random losses over time

Application of robust risk measures is an ongoing exercise especially as we learn more about loss and its impact on operations and strategy.

## 7 Measuring Volatility

### 7.1 Introduction

- Your company owns several facilities for the manufacture and distribution of polysyllabic acid, a compound used in several industries (and of course quite fictional!).
- Inputs to the manufacturing and distribution processes include various petroleum products and natural gas. Price swings can be quite volatile, and you know that Brent crude exhibits volatility clustering.
- Working capital management is always a challenge, and recent movements in the dollar against the euro and pound sterling have impacted cash conversion severely.

8 Your CFO wants to hedge the input price risk using futures and over-the-counter (OTC) instruments.

The board is concerned because the company has missed its earnings estimate for the fifth straight quarter in the row. Shareholders are not pleased. The culprits seem to be a volatile cost of goods sold coupled with large swings in some revenue segments in the United Kingdom (UK) and the European Union (EU). Your CFO has handed you the job of organizing the company's efforts to understand the limits your exposure can extend. The analysis will be used to develop policy guidelines for managing customers and vendor relationships.

#### 7.1.1 Example

1. What are the key business questions you should ask about energy pricing patterns?
2. What systematic approach might you use to manage input volatility?

Here are some ideas.

1. Key business questions might be
  - Which input prices and exchange rates are more volatile than others and when?
  - Are price movements correlated?
  - In times of market stress how volatile can they be?
  - Are there hedging instruments we can deploy or third parties we can use to mitigate pricing risk?
2. Managing volatility
  - Set up a monitoring system

### 7.2 What is ALL the Fuss About?

We have already looked at volatility clustering. ARCH models are one way to model this phenomenon

*ARCH* stands for

- \* Autoregressive (lags in volatility)
- \* Conditional (any new values depend on others)
- \* Heteroscedasticity (Greek for varying volatility, here time-varying)

These models are especially useful for financial time series that exhibit periods of large return movements alongside intermittent periods of relative calm price changes.

An experiment is definitely in order.

The AR+ARCH model can be specified starting with  $z(t)$  standard normal variables and initial (we will overwrite this in the simulation) volatility series  $\sigma(t)^2 = z(t)^2$ . We then condition these variates with the  $\epsilon(t) = (\sigma(t)^2)^{1/2} z(t)$ . Then we first compute for each date  $t = 1 \dots n$ ,

$$\epsilon(t) = (\sigma^2)^{1/2} z(t)$$

Then, using this conditional error term we compute the autoregression (with lag 1 and centered at the mean  $\mu$ )

$$y(t) = \mu + \phi(y(t-1) - \mu) + \epsilon(t)$$

Now we are ready to compute the new variance term.

```
n <- 10500 ## lots of trials
z <- rnorm(n) ## sample standard normal distribution variates
e <- z ## store variates
y <- z ## store again in a different place
sig2 <- z^2 ## create volatility series
omega <- 1 ## base variance
alpha <- 0.55 ## Markov dependence on previous variance
phi <- 0.8 ## mMarkov dependence on previous period
mu <- 0.1 ## average return
omega/(1-alpha) ; sqrt(omega/(1-alpha))

## [1] 2.222222
## [1] 1.490712

set.seed("1012")
for (t in 2:n) ## Because of lag start at second date
{
  e[t] <- sqrt(sig2[t])*z[t] # 1. e is conditional on sig
  y[t] <- mu + phi*(y[t-1]-mu) + e[t] ## 2. generate returns
  sig2[t+1] <- omega + alpha * e[t]^2 ## 3. generate new sigma^2 to feed 1
}
```

A plot is a little than instructure

```
par(mfrow = c(2,4))
plot(z[10001:n], type = "l", xlab = "t", ylab = expression(epsilon), main = "1. Simple noise")

plot(sqrt(sig2[10000:n]), type = "l", xlab = "t", ylab = expression(sigma[t]), main = "2. Conditional sigma")

plot(e[10001:n], type = "l", xlab = "t", ylab = "a", main = "3. ARCH")

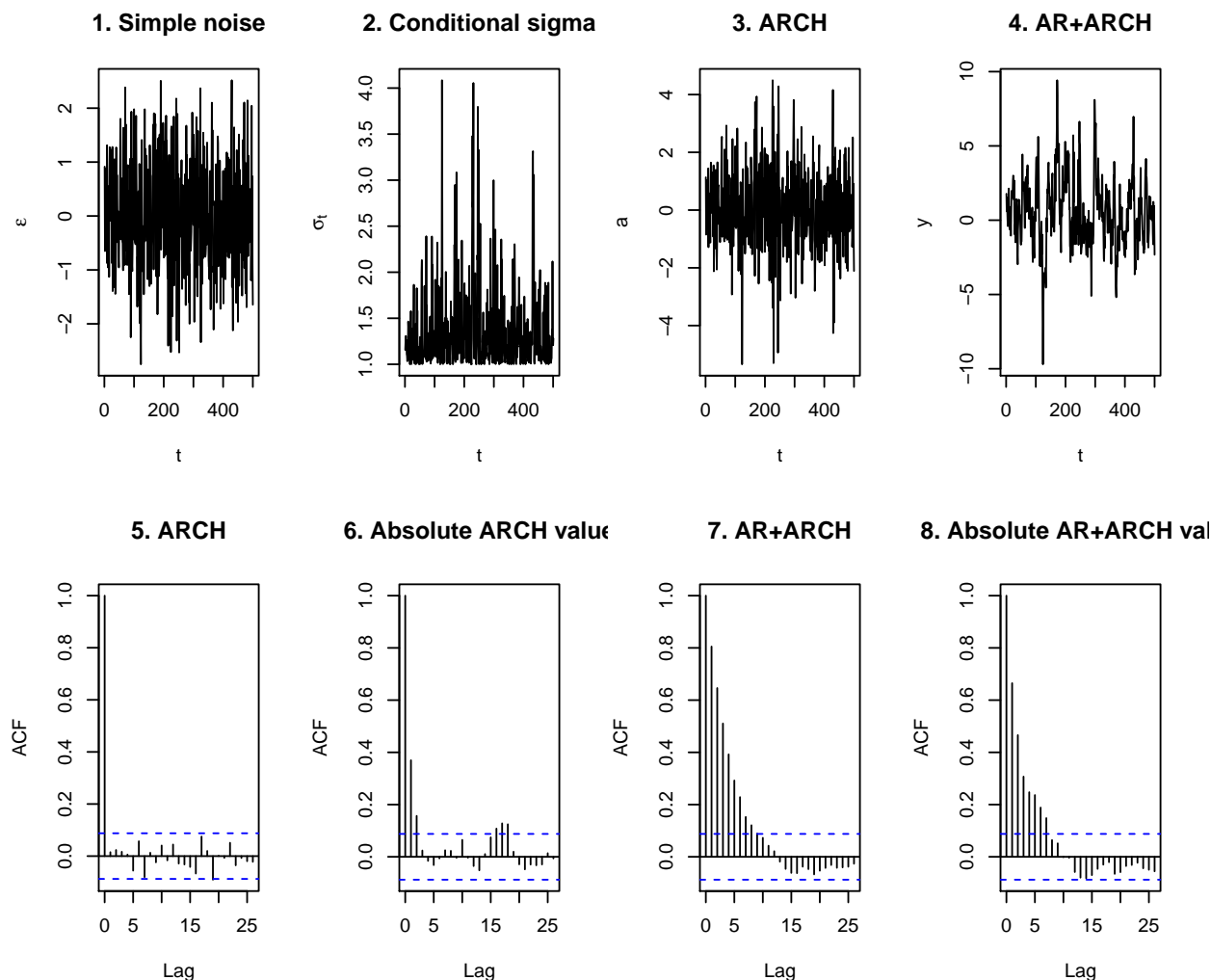
plot(y[10001:n], type = "l", xlab = "t", ylab = "y", main = "4. AR+ARCH")

acf(e[10001:n], main = "5. ARCH")

acf(abs(e[10001:n]), main = "6. Absolute ARCH value")

acf(y[10001:n], main = "7. AR+ARCH")

acf(abs(y[10001:n]), main = "8. Absolute AR+ARCH value")
```



What do we see?

1. Large outlying peaks in the conditional standard deviation
2. Showing up as well in the ARCH plot
3. AR adds the clustering as returns attempt to revert to the long run mean of  $\mu = 10$  percent.
4. Patterns reminiscent of clustering occur with thick and numerous lags in the acf plots. There is persistence of large movement both up and down.

Why does it matter?

- Revenue received from customer contracts priced with volatility clustering will act like the prices: when in a downward spiral, that spiral amplifies more than when prices try to trend upward.
- The same will happen with the value of inventory and the costs of inputs.
- All of this adds up to volatile EBITDA (Earning Before Interest and Tax adding in non-cash Depreciation and Amortization), missed earnings targets, shareholders selling, the stock price dropping and equity-based compensation falling.

We have more than one way to estimate the parameters of the AR-ARCH process. Essentially we are running yet another “regression”. Let’s first load some data to tackle the CFO’s question around exposures in the UK,

EU, and in the oil market.

```
library(rugarch)
library(qrmdata)
library(xts)
# Currency Data
data("EUR_USD")
data("GBP_USD")
# Brent Oil data (FRED)
data("OIL_Brent")

data.1 <- na.omit(merge(EUR_USD, GBP_USD, OIL_Brent))

P <- data.1
R <- na.omit(diff(log(P)) * 100)
names.R <- c("EUR.USD", "GBP.USD", "OIL.Brent")
colnames(R) <- names.R
Brent.p <- data.1[, 3]
Brent.r <- R[, 3] ## Pull out just the Brent Prices
```

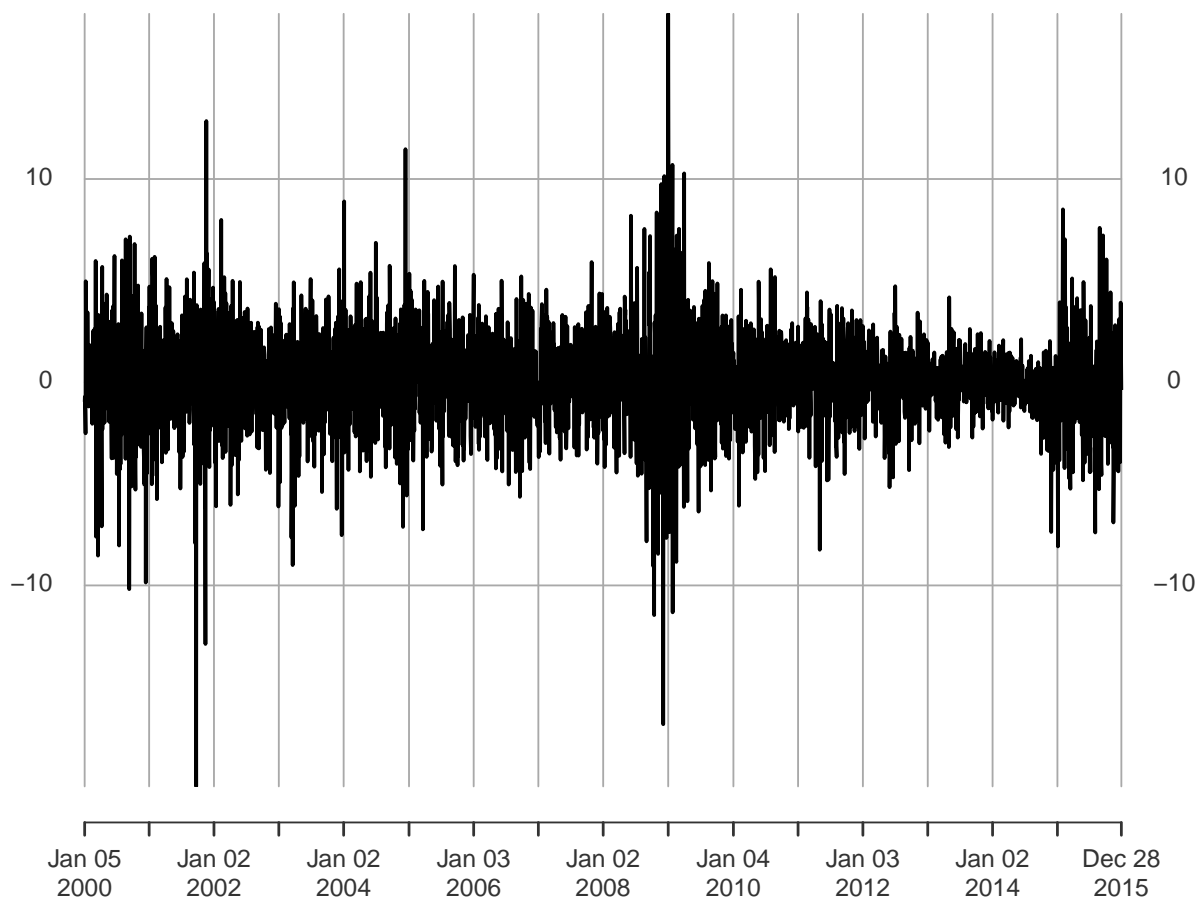
Then we plot the data, transformations, and autocorrelations.

```
plot(Brent.r)
```



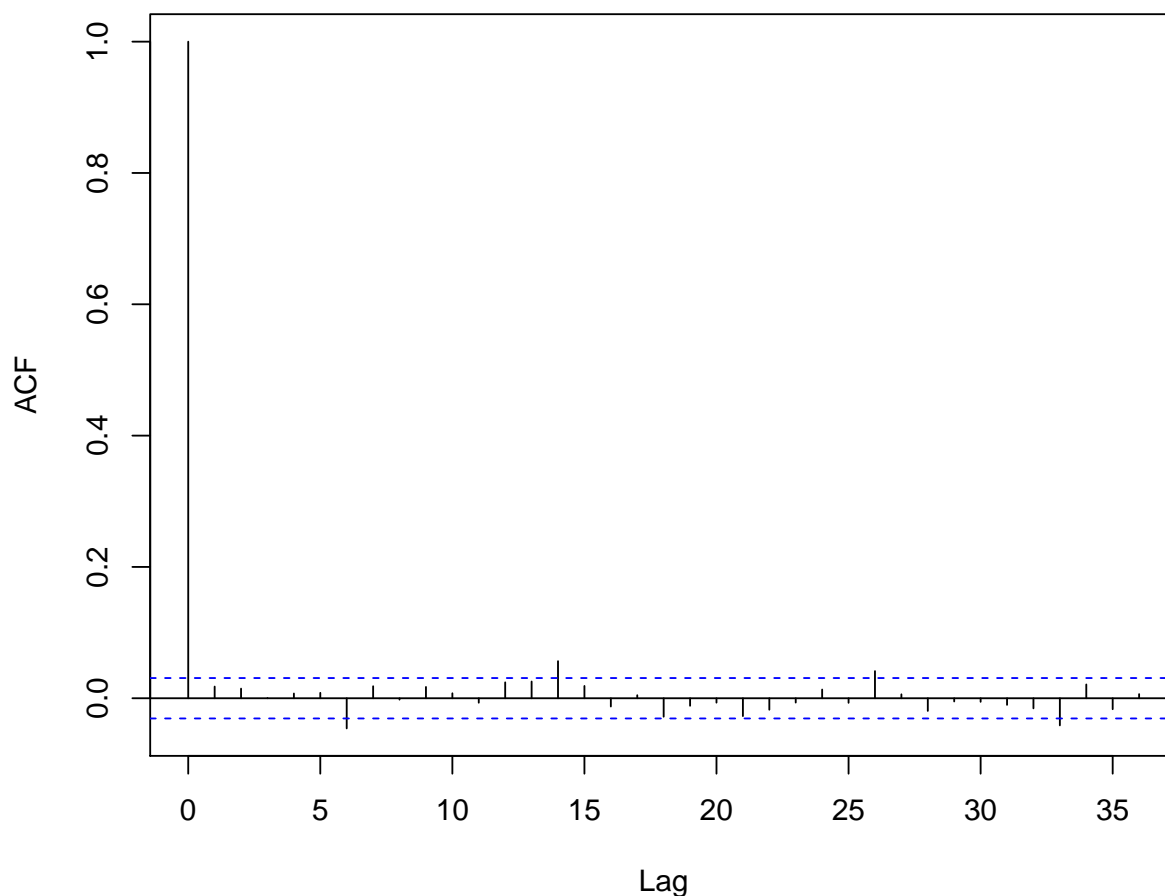
Brent.r

2000-01-05 / 2015-12-28



```
acf(Brent.r)
```

## Series Brent.r



```
Box.test(Brent.r, type = c("Ljung-Box"))
```

```
##  
## Box-Ljung test  
##  
## data: Brent.r  
## X-squared = 1.2847, df = 1, p-value = 0.257
```

The p-value is small enough to more than reject the null hypothesis that the 14-day lag is not significantly different from zero

### 7.3 It is Fitting...

Our first mechanical task is to specify the ARMA-GARCH model. First we specify.

1. Use the `ugarchspec` function to specify a plain vanilla `sGarch` model.
2. `garchOrder = c(1, 1)` means we are using the first lags of residuals squared and variance or (with  $\omega$ , “omega”, the average variance,  $\sigma_t^2$ ), here with brent returns.

$$\sigma_t^2 = \omega + \alpha_1 \varepsilon_{t-1}^2 + \beta_{t-1} \sigma_{t-1}^2$$

3. Use `armaORDER = c(1, 0)` to specify the mean Brent return model with long run average  $\mu$

$$r_t = \mu + \psi_1 y_{t-1} + \psi_2 \varepsilon_{t-1}$$

4. Include `means` as in the equations above.
5. Specify the distribution as `norm` for normally distributed innovations  $\epsilon_t$ . WE will also compare this fit with the `std` Student's t-distribution innovations using the Akaike Information Criterion (AIC).
6. Fit the data to the model using `ugarchfit`

```
AR.GARCH.Brent.Norm.spec <- ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1, 1)), m
fit.Brent.norm <- ugarchfit(spec = AR.GARCH.Brent.Norm.spec, data = Brent.r)
```

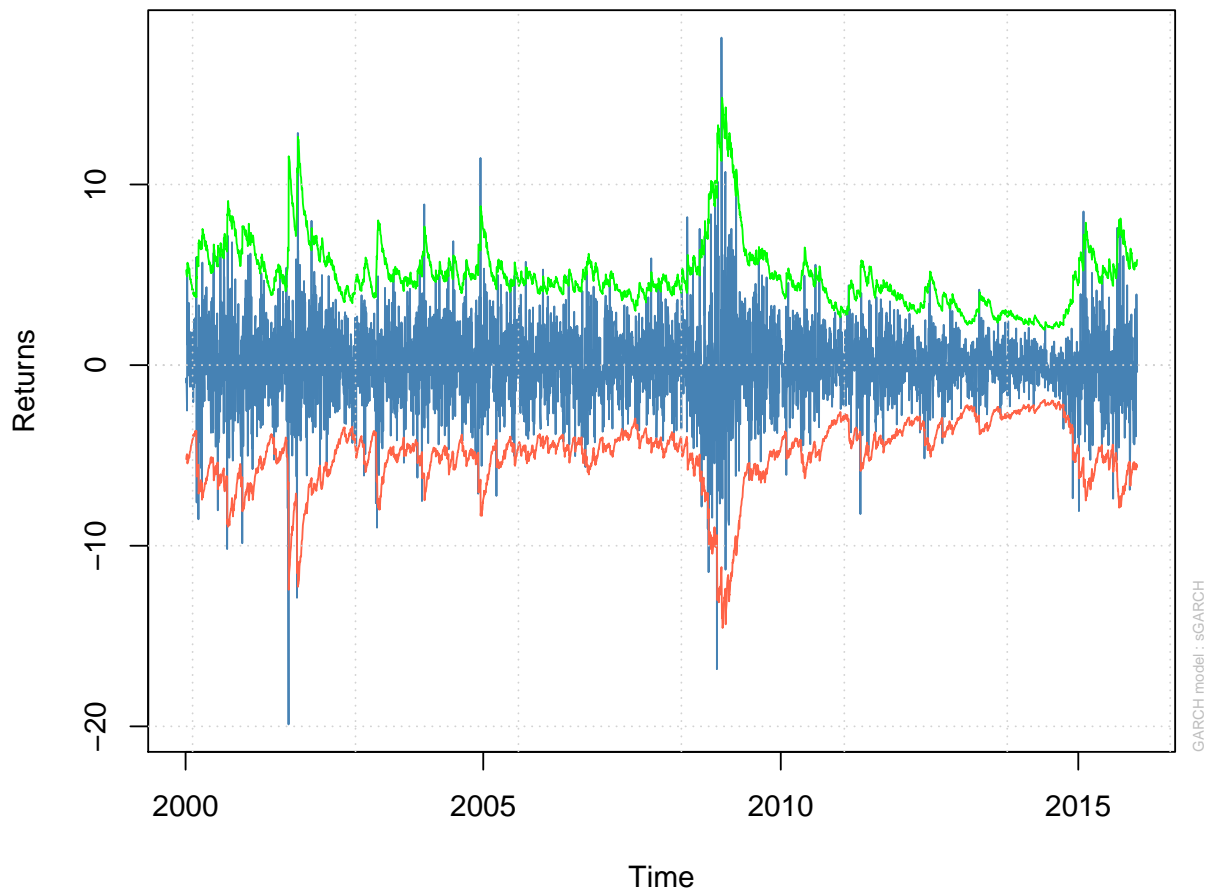
Let's look at the conditional quantiles from this model, otherwise known as the VaR limits, nominally set at 99 percent.

```
## First the series with conditional
## quantiles

plot(fit.Brent.norm, which = 2)

##
## please wait...calculating quantiles...
```

Series with with 1% VaR Limits



Now let's generate a panel of plots.

```
par(mfrow = c(2, 2))
## acf of absolute data - shows serial
## correlation

plot(fit.Brent.norm, which = 6)
## QQPlot of data - shows
## leptokurtosis of standardized
## residuals - normal assumptions not
## supported

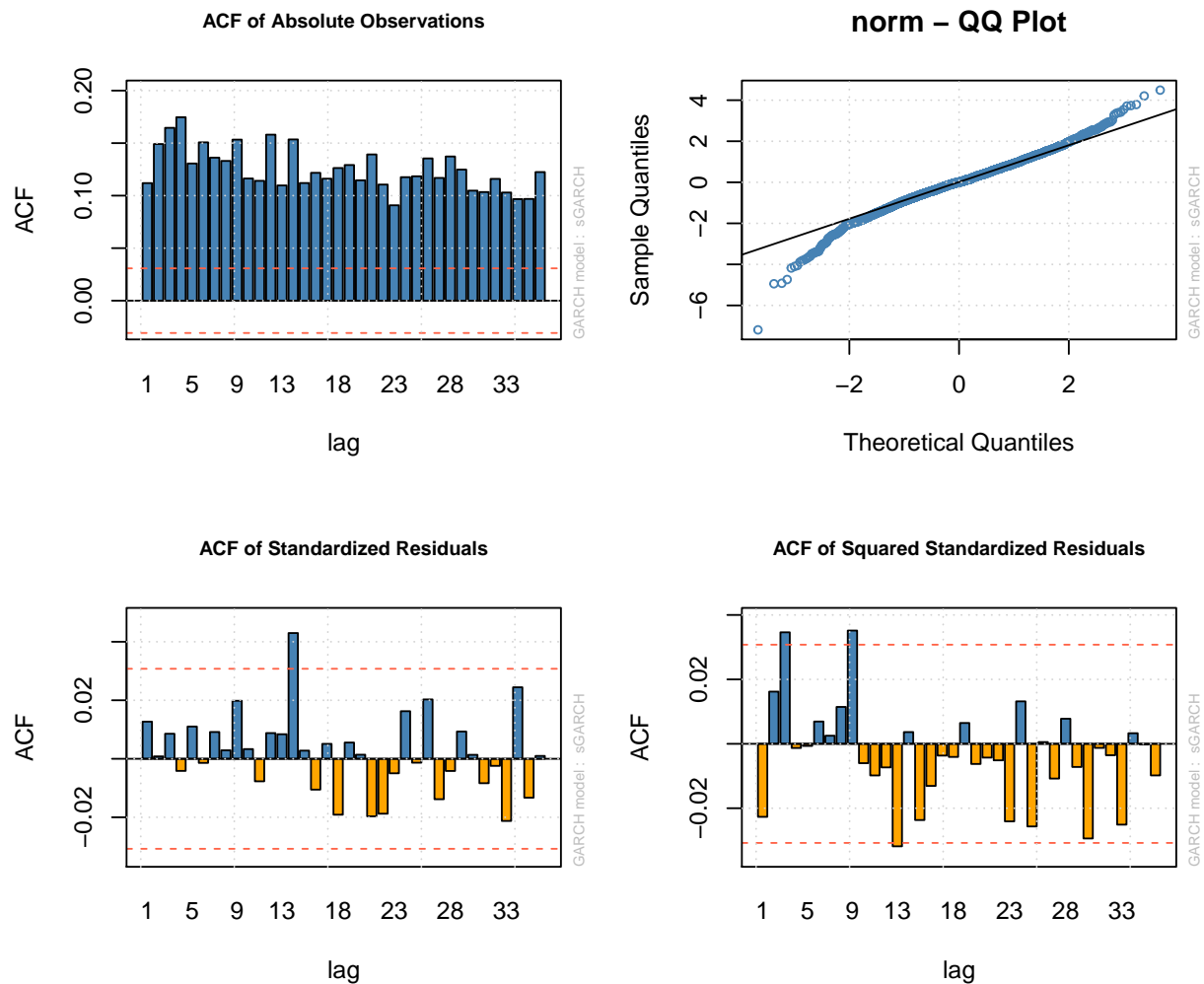
plot(fit.Brent.norm, which = 9)
## acf of standardized residuals -
## shows AR dynamics do a reasonable
## job of explaining conditional mean

plot(fit.Brent.norm, which = 10)

## acf of squared standardized
```

```
## residuals - shows GARCH dynamics do
## a reasonable job of explaining
## conditional sd
```

```
plot(fit.Brent.norm, which = 11)
```



### 7.3.1 Example

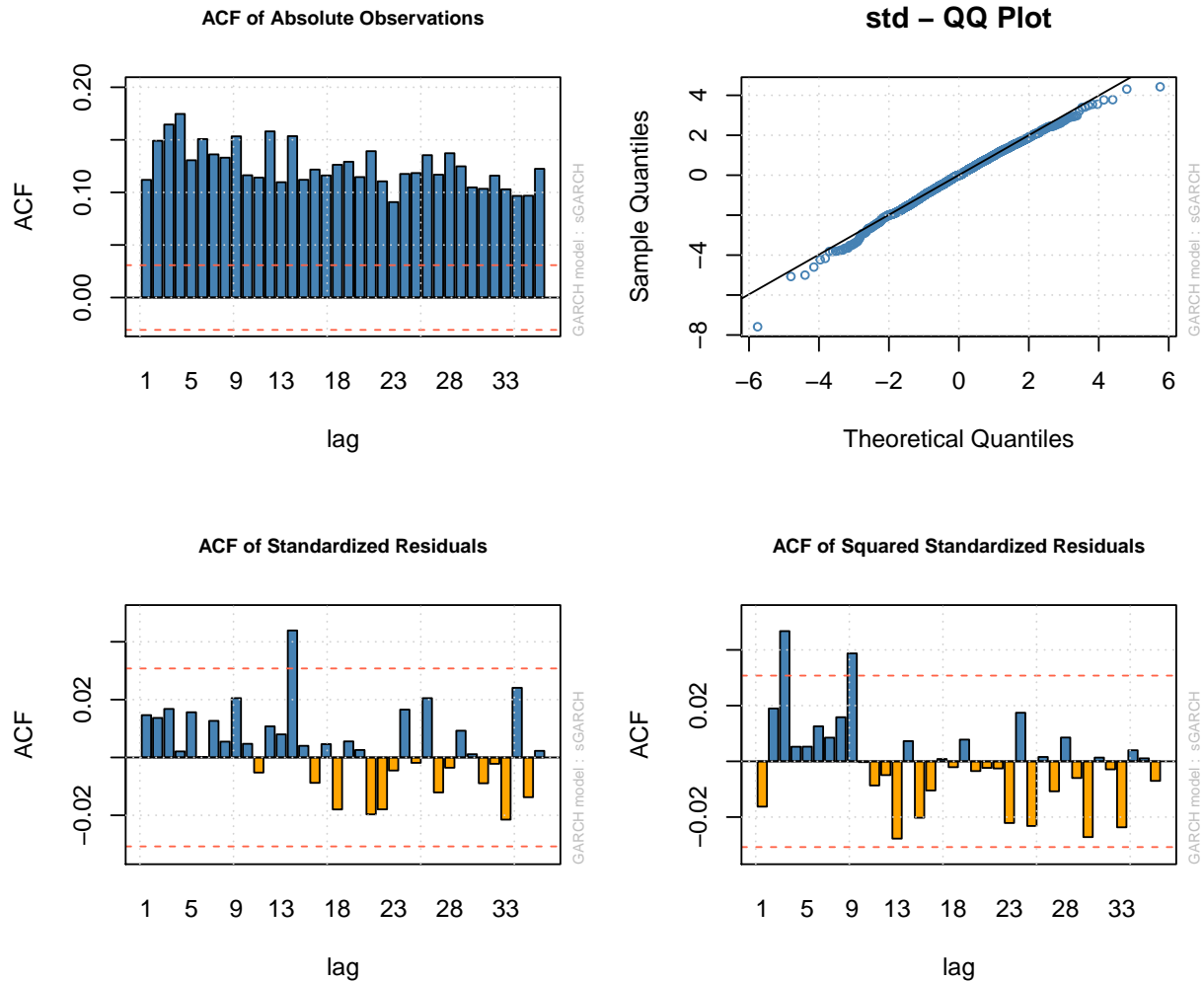
Let's redo the GARCH estimation now using the possibly more realistic thicker tails of the Student t-distribution for the  $\epsilon$  innovations. Here we just replace `norm` with `std` in the `distribution.model =` statement in the `ugarchspec` function.

```
## Fit an AR(1)-GARCH(1,1) model with
## student innovations
```

```
AR.GARCH.Brent.T.spec <- ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1,1)), mean.model =
```

```
fit.Brent.t <- ugarchfit(spec = AR.GARCH.Brent.T.spec, data = Brent.r)
```

```
par(mfrow =c(2,2))
plot(fit.Brent.t,which = 6)
plot(fit.Brent.t,which =9)
plot(fit.Brent.t,which =10)
plot(fit.Brent.t,which =11)
```



```
par(mfrow =c(1,1))
```

Here are some results

- 1.ACF of absolute observations indicates much volatility clustering.
- 2.These are significantly dampened by the AR-ARCH estimation with almost bounded standardized residuals (residual / standard error).
- 3.More evidence of this comes from the ACF of the squared standardized residuals.
- 4.It appears that this AR-GARCH specification and Student's t-distributed innovations captures most of the movement in volatility for Brent.

Which model? - Use the Akaike Information Criterion (AIC) to measure information leakage from a model. - AIC measures the amount of information used in a model specified by a log likelihood function. - Likelihood:

probability that you will observe the Brent returns given the parameters estimated by (in this case) the GARCH(1,1) model with normal or t-distributed innovations. - Smallest information leakage (smallest AIC) is the model for us.

Compute 5. Using normally distributed innovations produces a model with  $AIC = 4.2471.6$ . Using Student's t-distributed innovations produces a model with  $AIC = 4.2062$ . 7.GARCH(1,1) with Student's t-distributed innovations is more likely to have less information leakage than the GARCH(1,1) with normally distributed innovations.

Here are some common results we can pull from the fit model:

```
coef(fit.Brent.t)
```

```
##          mu          ar1          omega          alpha1          beta1          shape
## 0.04021501 0.01726091 0.01088057 0.03815145 0.96074931 7.04099749
```

Coefficients include:

- `mu` is the long run average Brent return.
- `ar1` is the impact of one day lagged return on today's return.
- `omega` is the long run variance of Brent return.
- `alpha1` is the impact of lagged squared variance on today's return.
- `beta1` is the impact of lagged squared residuals on today's Brent return.
- `shape` is the degrees of freedom of the Student's t-distribution and the bigger this is, the thicker the tail.

Let's plot the star of this show: time-varying volatility.

```
coef(fit.Brent.t)
```

```
##          mu          ar1          omega          alpha1          beta1          shape
## 0.04021501 0.01726091 0.01088057 0.03815145 0.96074931 7.04099749
```

```
plot(sigma(fit.Brent.t))
```

`sigma(fit.Brent.t)`

2000-01-05 / 2015-12-28



Here's the other reason for going through this exercise: we can look at any Brent volatility range we like.

```
plot(quantile(fit.Brent.t, 0.99))
```



quantile(fit.Brent.t, 0.99)

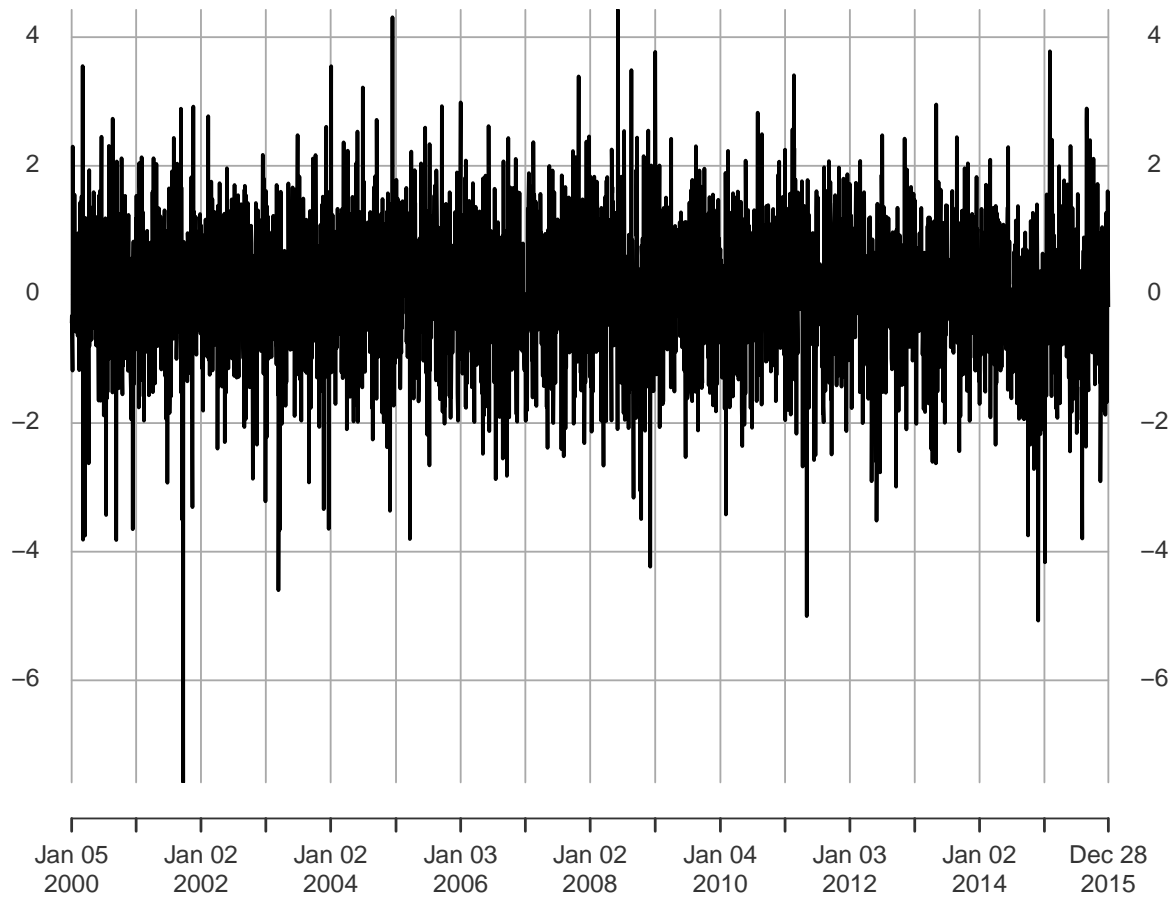
2000-01-05 / 2015-12-28



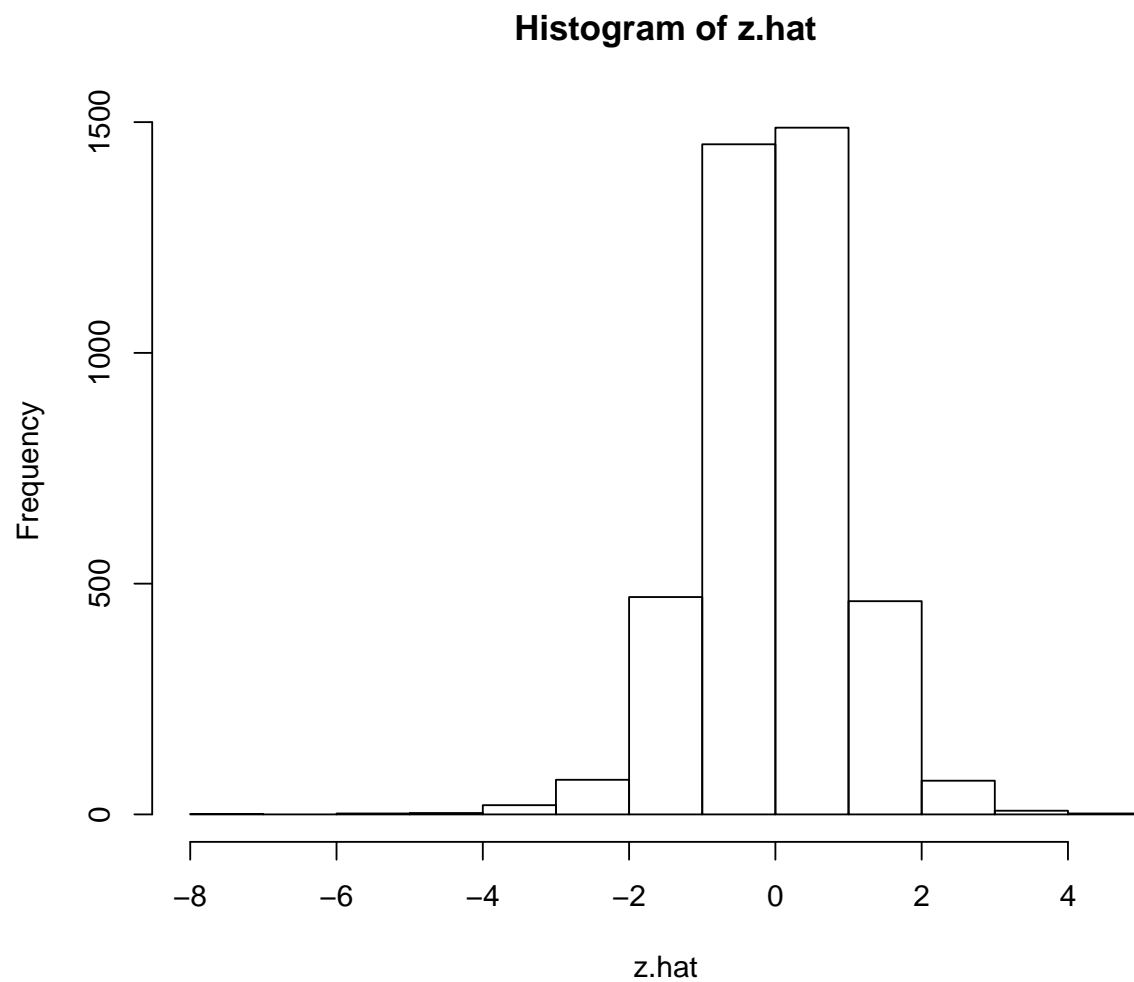
```
z.hat <- residuals(fit.Brent.t, standardize = TRUE)
plot(z.hat)
```

**z.hat**

2000-01-05 / 2015-12-28



```
hist(z.hat)
```



```
mean(z.hat)
```

```
## [1] -0.01813252
```

```
var(z.hat)
```

```
##          [,1]
```

```
## [1,] 1.000743
```

```
library(moments)
```

```
skewness(z.hat)
```

```
##          X
```

```
## -0.3208621
```

```
kurtosis(z.hat)
```

```
##          X
```

```
## 5.051197
```

```
shapiro.test(as.numeric(z.hat))
```

```
##
```

```
## Shapiro-Wilk normality test
##
## data:  as.numeric(z.hat)
## W = 0.98439, p-value < 2.2e-16
```

```
jarque.test(as.numeric(z.hat))
```

```
##
## Jarque-Bera Normality Test
##
## data:  as.numeric(z.hat)
## JB = 780.84, p-value < 2.2e-16
## alternative hypothesis: greater
```

What do we see?

- Left skewed.
- Thick tailed
- Potentially large losses can occur with ever larger losses in the offing
- More negative than positive
- Both standard tests indicate rejection of the null hypothesis that the series is normally distributed.

## 7.4 Simulate... again until Morale Improves...

1. Specify the AR-GARCH process using the parameters from the fit.Brent.t results
2. Generate 2000 paths

```
require(rugarch)
```

```
GARCHspec <- ugarchspec(variance.model = list(model = "sGARCH",
                                              garchOrder = c(1, 1)),
                        mean.model = list(armaOrder = c(1, 0), include.mean = TRUE),
                        distribution.model = "std",
                        fixed.pars = list(mu = 0.04, ar1 = 0.0173,
                                          omega = 0.0109, alpha1 = 0.0382,
                                          beta1 = 0.9607, shape = 7.0377))
```

```
GARCHspec
```

```
##
## *-----*
## *      GARCH Model Spec      *
## *-----*
##
## Conditional Variance Dynamics
## -----
## GARCH Model      : sGARCH(1,1)
## Variance Targeting : FALSE
##
## Conditional Mean Dynamics
## -----
## Mean Model      : ARFIMA(1,0,0)
## Include Mean    : TRUE
## GARCH-in-Mean   : FALSE
##
```

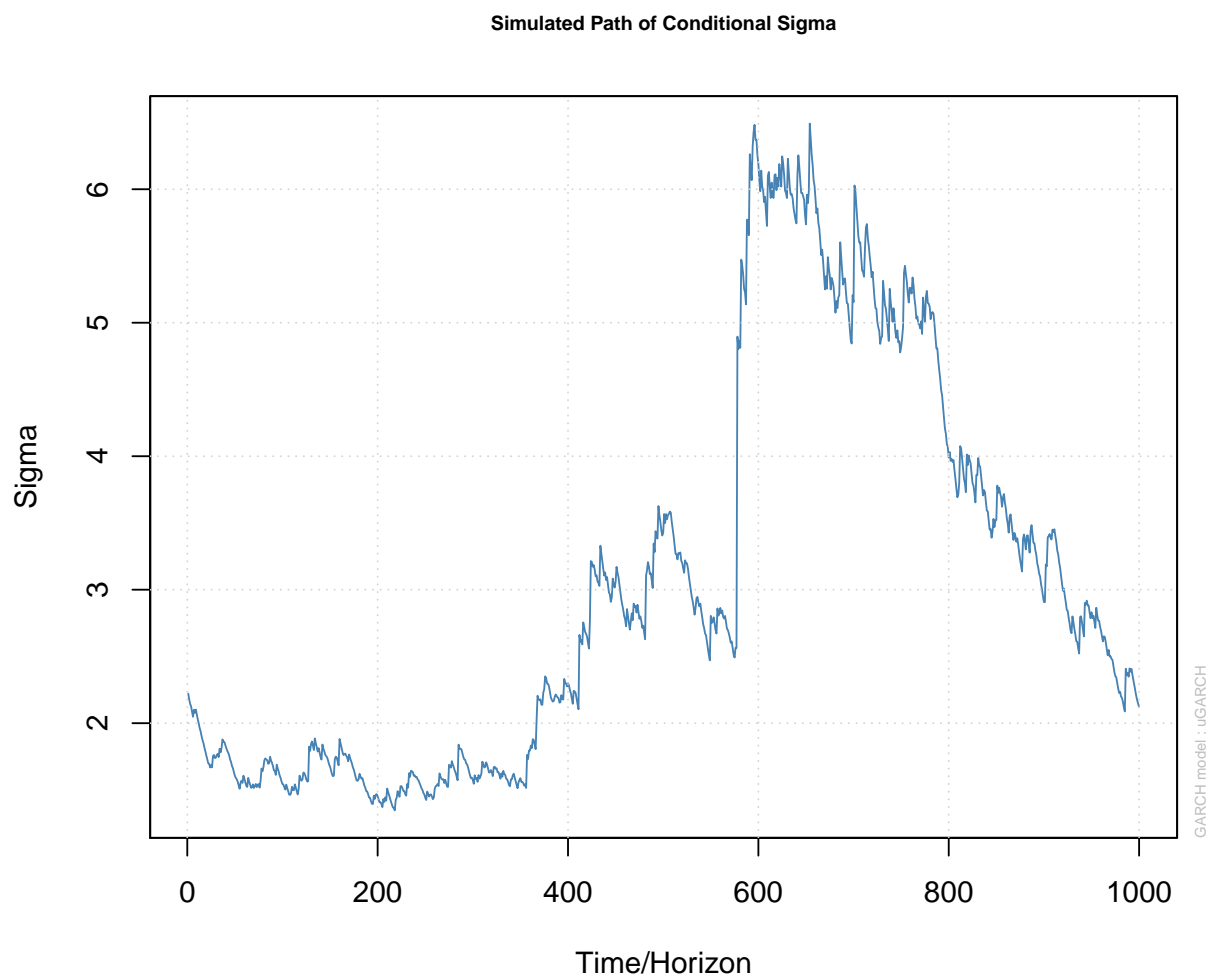
```
## Conditional Distribution
## -----
## Distribution : std
## Includes Skew : FALSE
## Includes Shape : TRUE
## Includes Lambda : FALSE

## Generate two realizations of length
## 2000

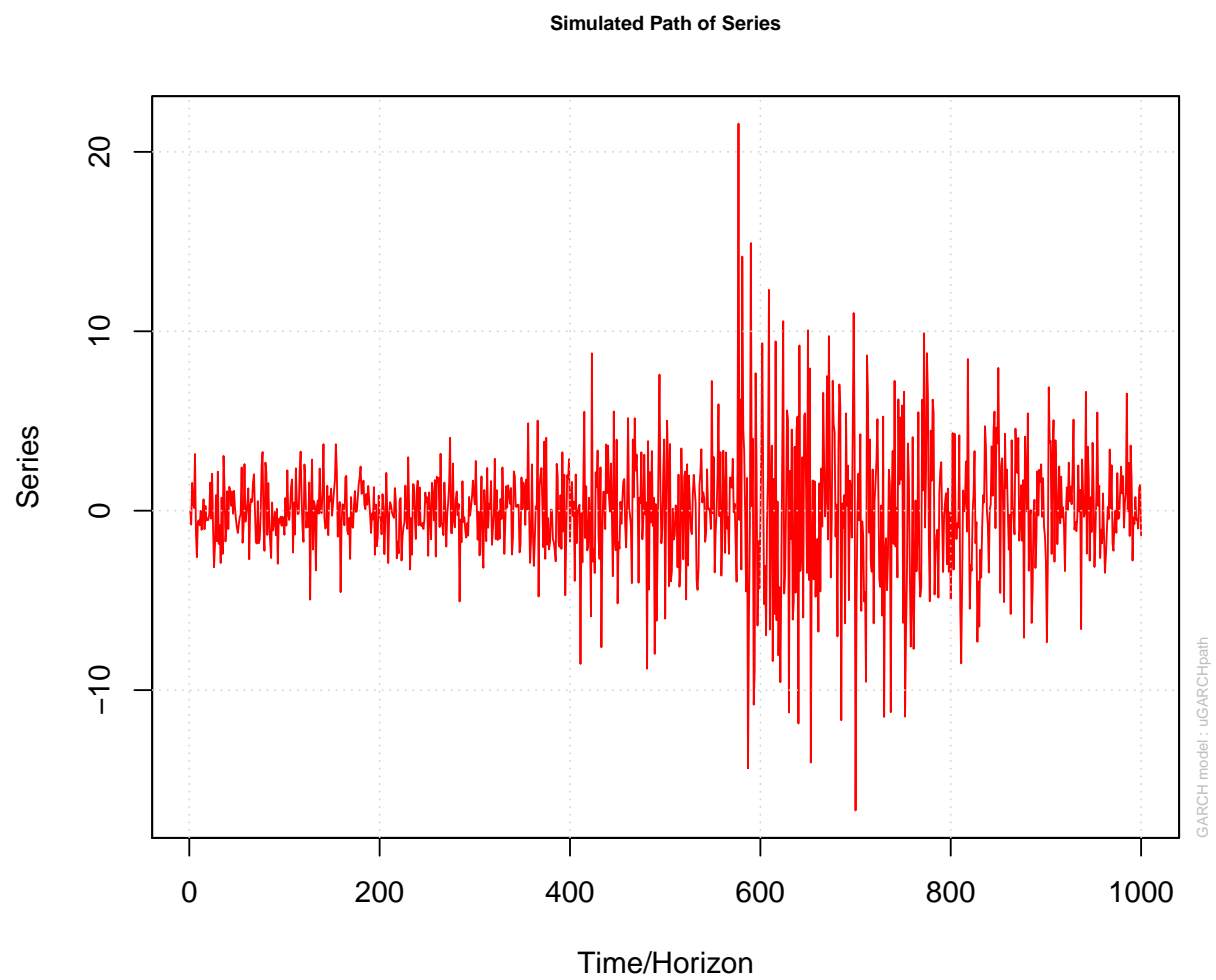
path <- ugarchpath(GARCHspec, nsim = 2000, n.start = 50, m.sim = 2)
```

There is a special plotting function for “uGARCHpath” objects.

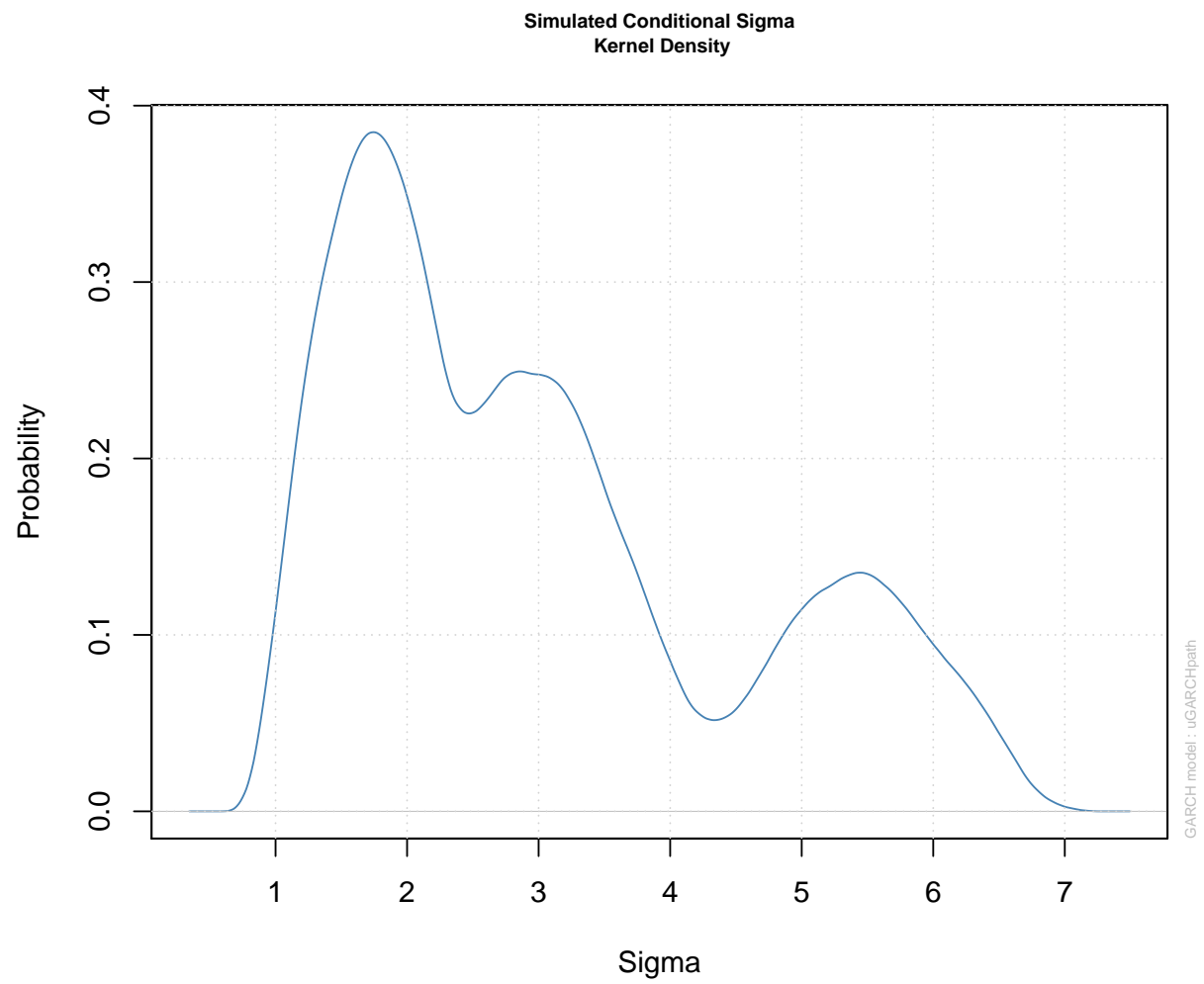
```
plot(path, which = 1)
```



```
plot(path, which = 2)
```



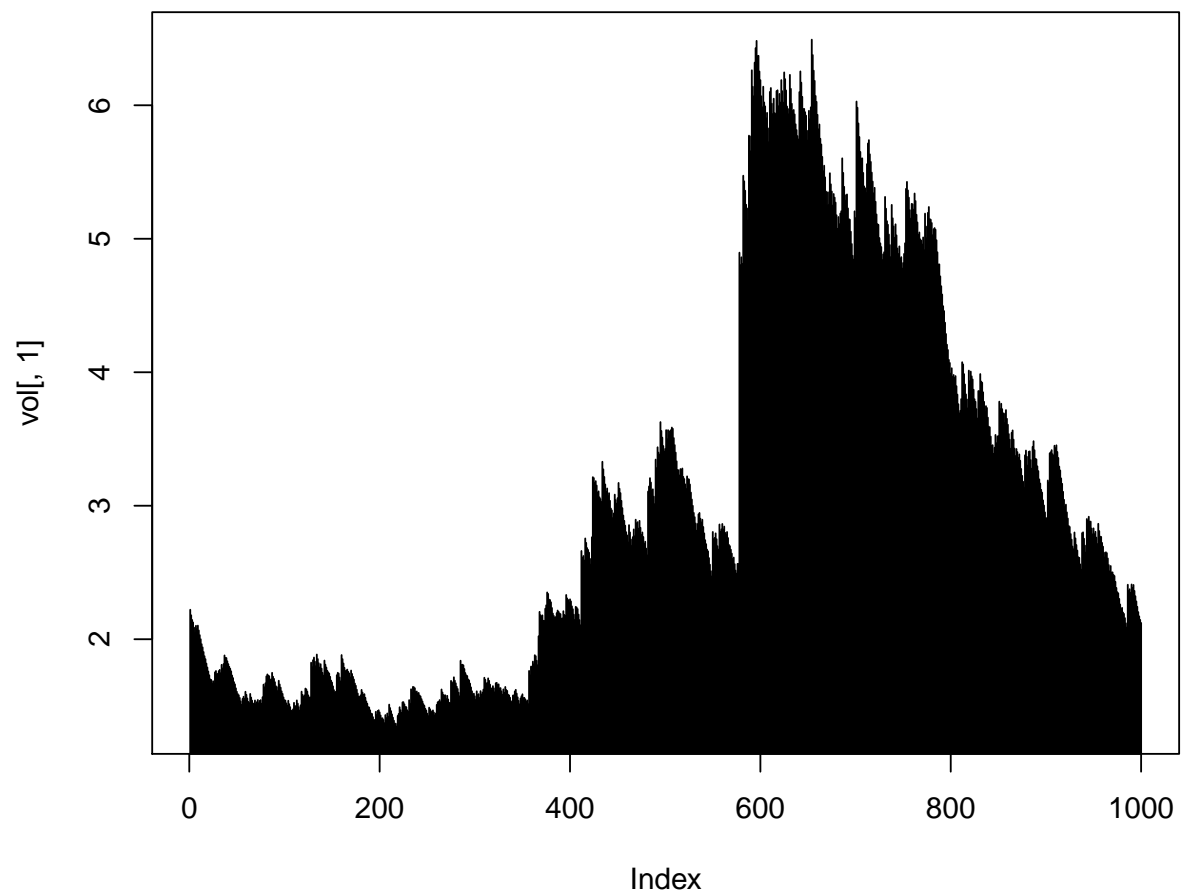
```
plot(path, which = 3)
```



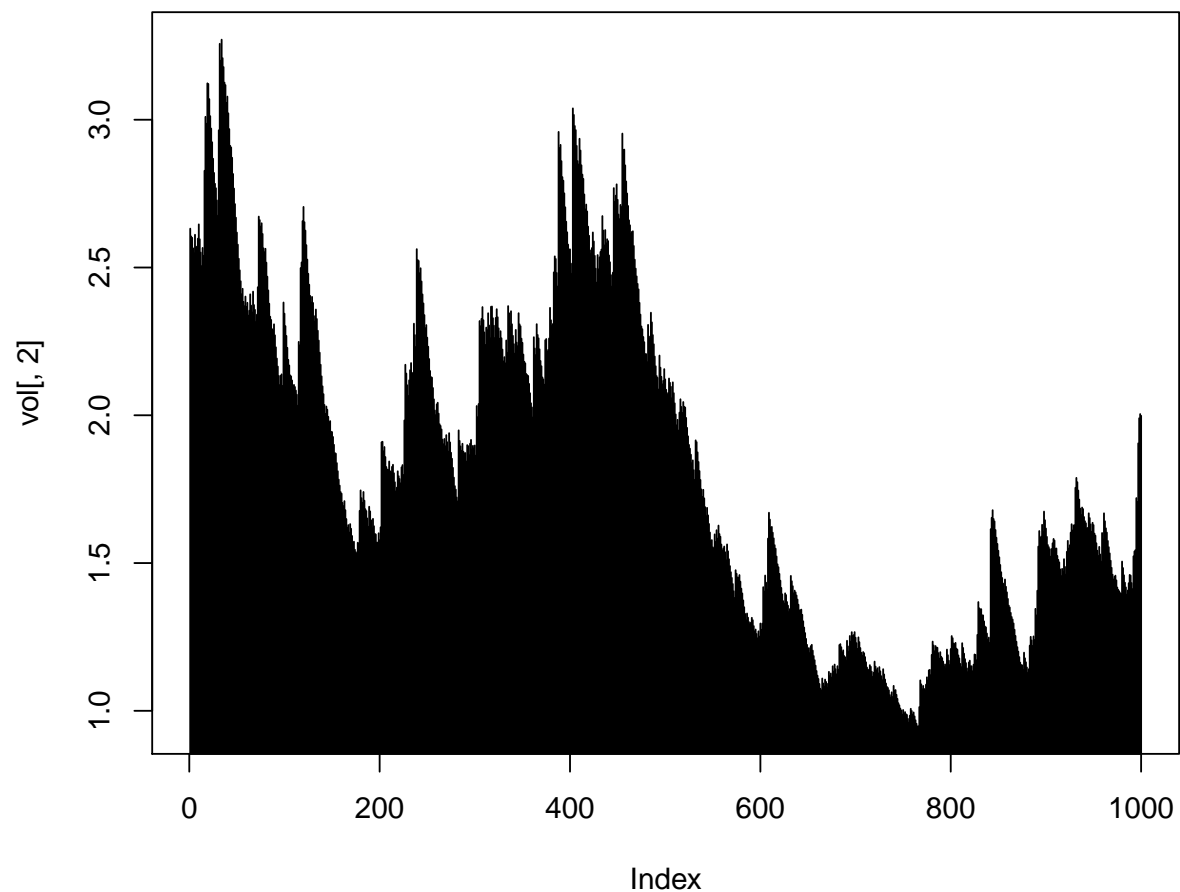
```
plot(path, which = 4)
```







```
plot(vol[, 2], type = "h")
```



Here is a little more background on the classes used

```
series <- path@path
## series is a simple list
class(series)
```

```
## [1] "list"
```

```
names(series)
```

```
## [1] "sigmaSim" "seriesSim" "residSim"
```

```
## the actual simulated data are in
## the matrix/vector called
## 'seriesSim'
```

```
X <- series$seriesSim
head(X)
```

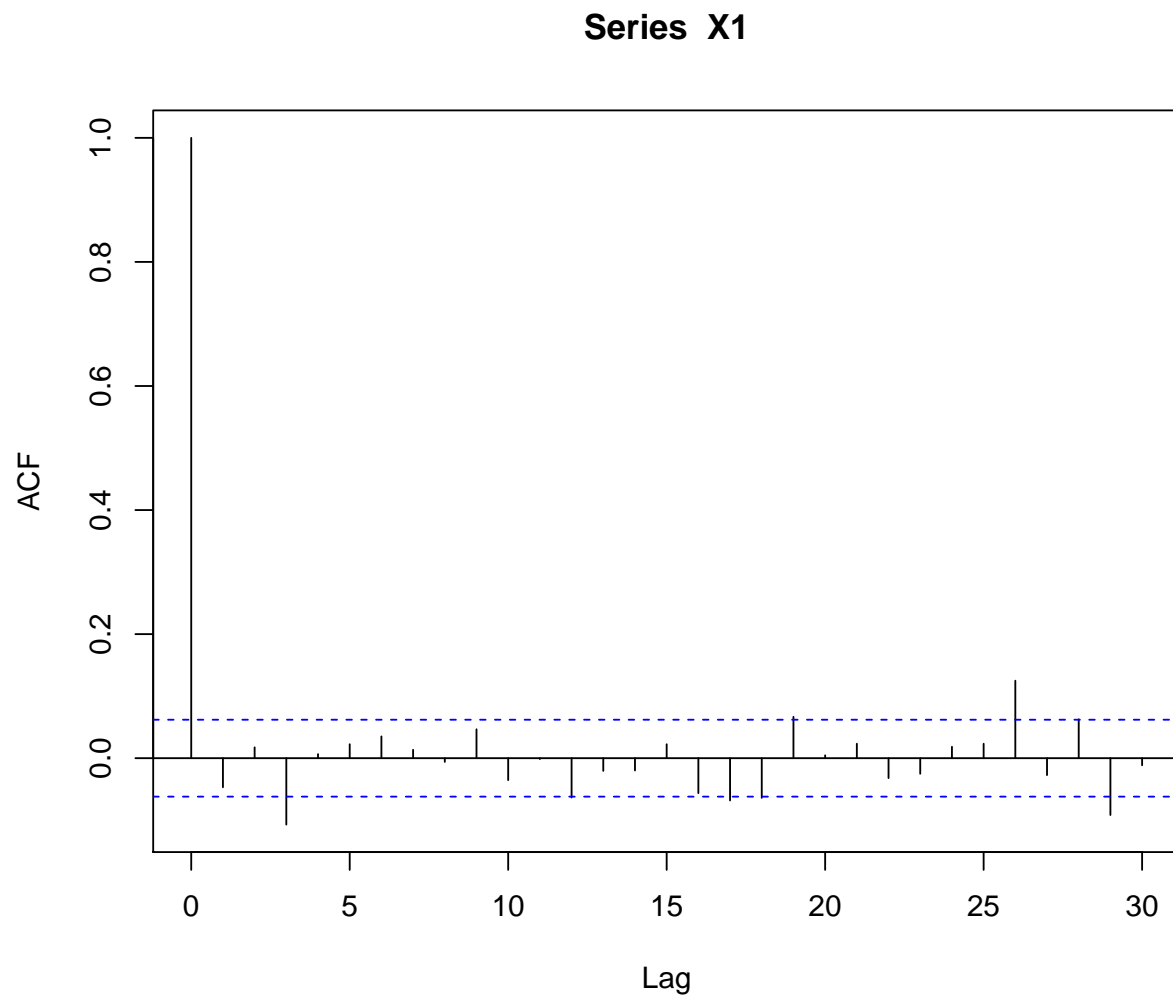
```
##           [,1]      [,2]
## [1,] -0.006481361  1.777493
## [2,] -0.773492065  2.535944
```

```
## [3,] 1.548749563 1.446420
## [4,] 0.387488720 -2.027726
## [5,] 0.151340652 -3.834220
## [6,] 3.161365645 0.954118
```

### 7.4.1 Example

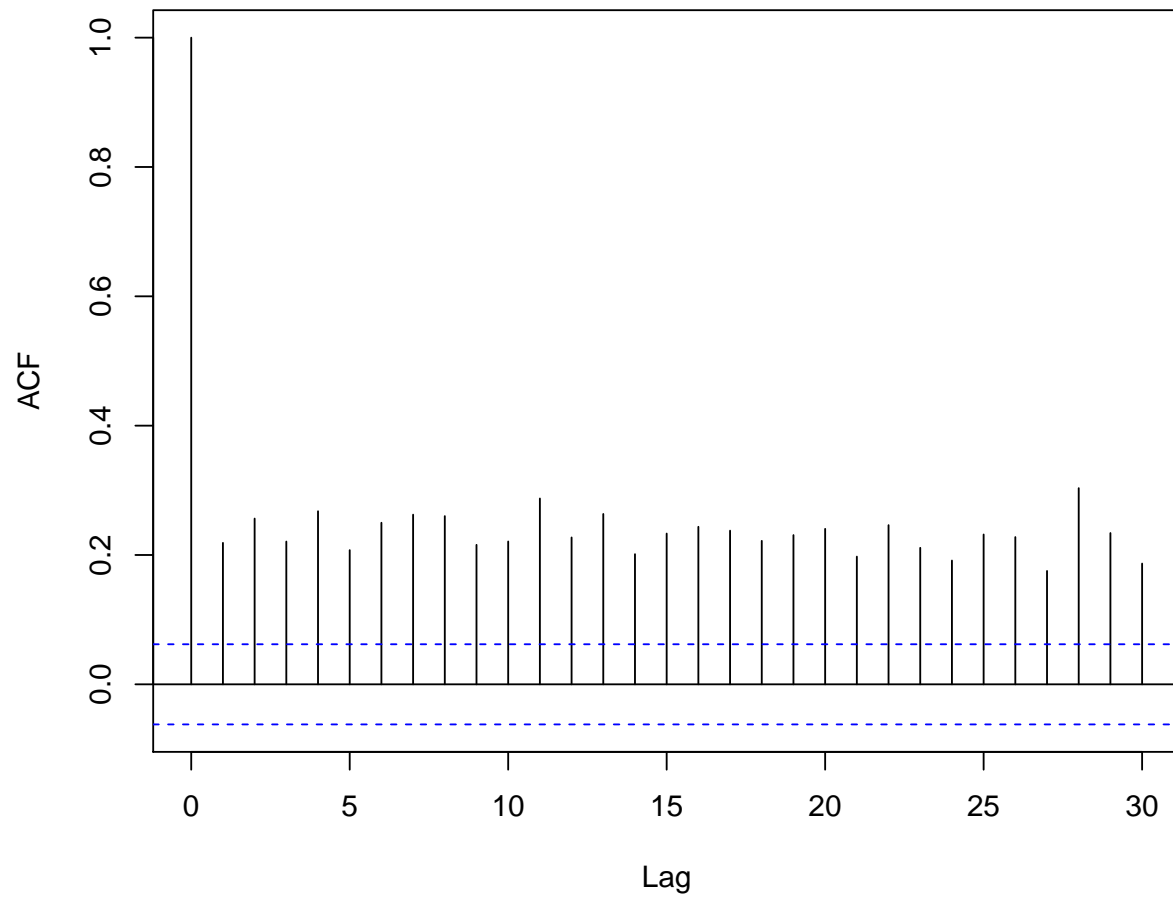
Does the simulated series conform to stylized facts?

```
X1 <- X[, 1]
acf(X1)
```

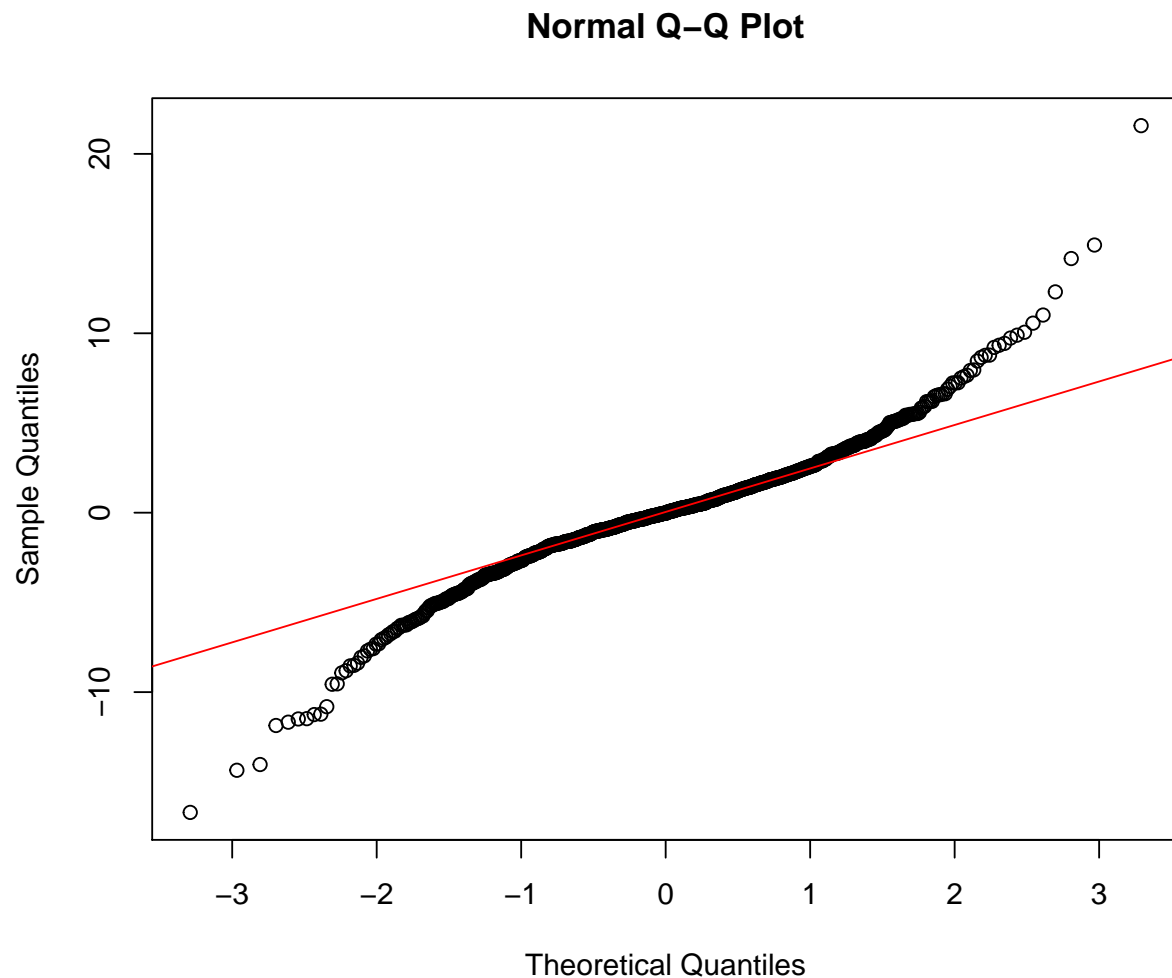


```
acf(abs(X1))
```

### Series abs(X1)



```
qqnorm(X1)  
qqline(X1, col = 2)
```



```
shapiro.test(X1)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  X1
## W = 0.94808, p-value < 2.2e-16
```

## 7.5 Multivariate GARCH models

We go from univariate GARCH to multivariate GARCH... and use the most recent techniques to make it into the fray:

- The Dynamic Conditional Correlation of Nobel Laureate Robert Engle.
- In the GARCH model we just did, individual assets following their own univariate GARCH process: they now have time-varying volatilities.
- Engle figured out how to make the correlation among asset return also time-varying.

Why? - What if we have a portfolio, like the accounts receivable that might face variations in exchange rate

and in Brent oil. - We would need to know the joint volatilities and dependencies of these three factors as they contribute to overall accounts receivable volatility - We would use these conditional variances at least to model option prices on instruments to manage currency and commodity risk.

```
library("rmgarch")
garch11.spec <- ugarchspec(mean.model = list(armaOrder = c(0,0)),
                           variance.model = list(garchOrder = c(1, 1),
                                                  model = "sGARCH"),
                           distribution.model = "std")

dcc.garch11.spec = dccspec(uspec = multispec(replicate(3, garch11.spec)), dccOrder = c(1, 1),
                           distribution = "mvt")
```

Look at dcc.garch11.spec

```
dcc.garch11.spec
```

```
##
## *-----*
## *      DCC GARCH Spec      *
## *-----*
## Model          : DCC(1,1)
## Estimation     : 2-step
## Distribution    : mvt
## No. Parameters : 21
## No. Series     : 3
```

Now for the fit (takes more than 27.39 seconds on my laptop...)

```
dcc.fit <- dccfit(dcc.garch11.spec, data = R)
```

```
dcc.fit
```

```
##
## *-----*
## *      DCC GARCH Fit      *
## *-----*
##
## Distribution    : mvt
## Model          : DCC(1,1)
## No. Parameters  : 21
## [VAR GARCH DCC UncQ] : [0+15+3+3]
## No. Series     : 3
## No. Obs.       : 4057
## Log-Likelihood  : -12820.82
## Av.Log-Likelihood : -3.16
##
## Optimal Parameters
## -----
##              Estimate Std. Error   t value Pr(>|t|)
## [EUR.USD].mu    0.006996   0.007195   0.97238 0.330863
## [EUR.USD].omega 0.000540   0.000288   1.87540 0.060738
## [EUR.USD].alpha1 0.036643   0.001590  23.04981 0.000000
## [EUR.USD].beta1 0.962357   0.000397 2426.48600 0.000000
## [EUR.USD].shape 9.344066   1.192113   7.83824 0.000000
## [GBP.USD].mu    0.006423   0.006386   1.00581 0.314509
## [GBP.USD].omega 0.000873   0.000327   2.67328 0.007511
```

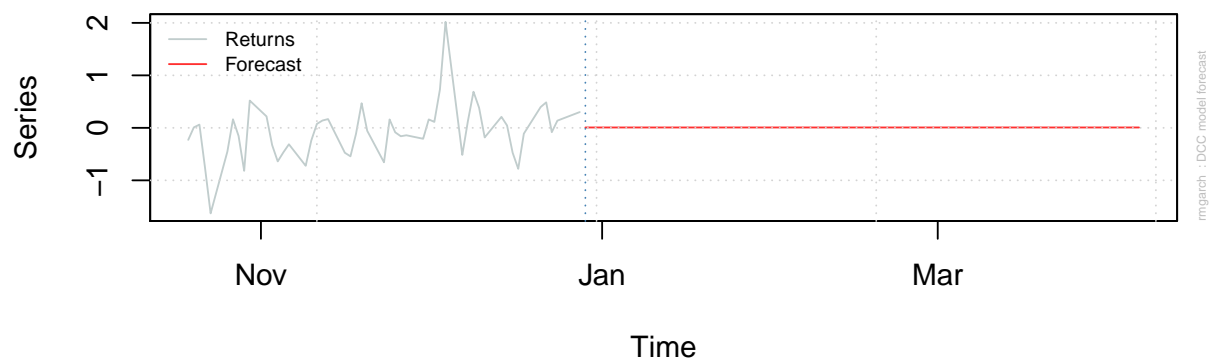
```
## [GBP.USD].alpha1    0.038292    0.002217    17.27051 0.000000
## [GBP.USD].beta1     0.958482    0.000555 1727.97104 0.000000
## [GBP.USD].shape    10.480971    1.534191    6.83160 0.000000
## [OIL.Brent].mu      0.040477    0.026697    1.51618 0.129474
## [OIL.Brent].omega   0.010780    0.004342    2.48275 0.013037
## [OIL.Brent].alpha1  0.037977    0.001940   19.57669 0.000000
## [OIL.Brent].beta1   0.960934    0.000453 2119.91309 0.000000
## [OIL.Brent].shape   7.040593    0.729750    9.64795 0.000000
## [Joint]dccal        0.009915    0.002821    3.51463 0.000440
## [Joint]dccb1        0.987616    0.004387   225.14806 0.000000
## [Joint]mshape       9.732815    0.653415   14.89531 0.000000
##
## Information Criteria
## -----
##
## Akaike          6.3307
## Bayes           6.3633
## Shibata         6.3306
## Hannan-Quinn    6.3423
##
##
## Elapsed time : 5.151176
```

- The mean models of each series (EUR.USD, GPB.USD, OIL.Brent) are overwhelmed by the preponderance of time-varying volatility, correlation, correlation, and shape (degrees of freedom since we used the Student's t-distribution).
- The joint conditional covariance (relative of correlation) parameters are also significantly different from zero.

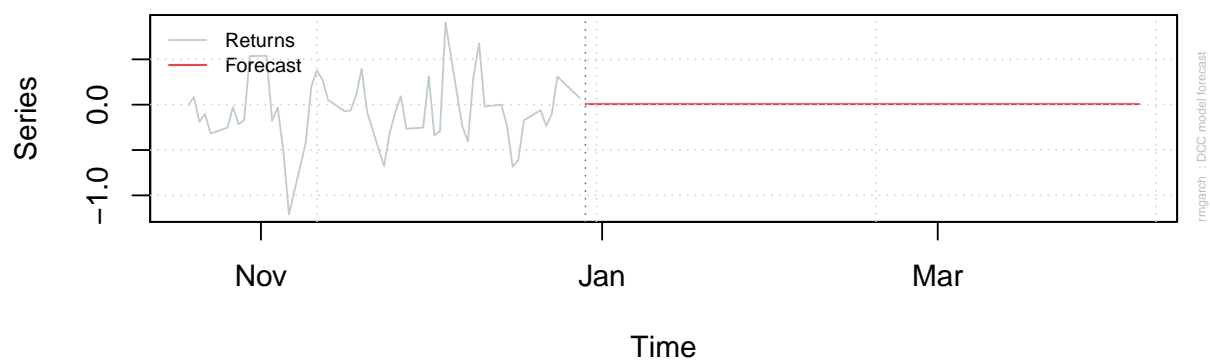
Using all of the information from the fit, we now forecast. These are the numbers we would use to simulate hedging instruments or portfolio VaR or ES. Let's plot the time-varying sigma first.

```
dcc.fcst <- dccforecast(dcc.fit, n.ahead = 100)
plot(dcc.fcst, which = 1)
```

## DCC Series Unconditional Forecast EUR.USD



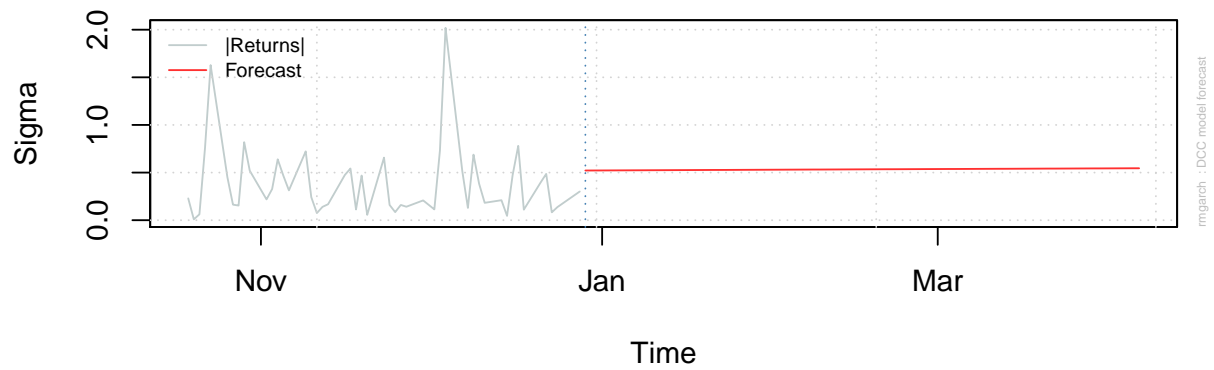
## GBP.USD



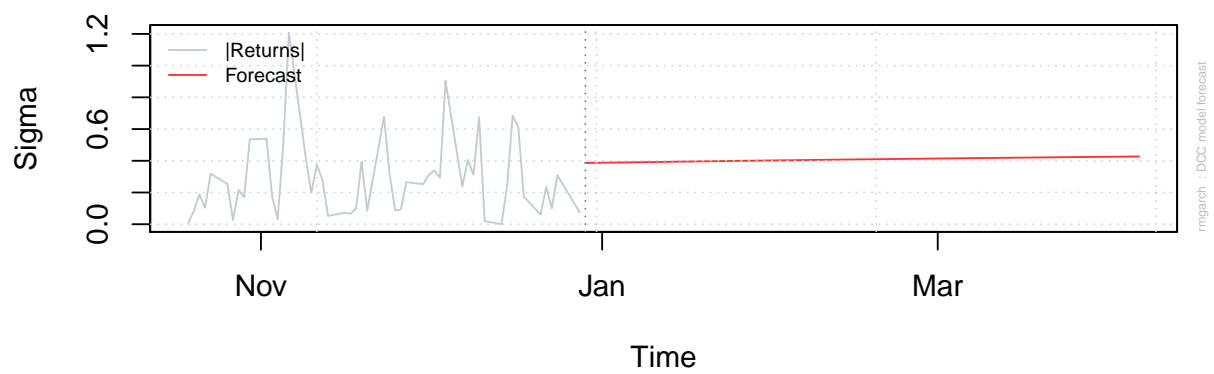
```
plot(dcc.fcst, which = 2)
```



### DCC Sigma Unconditional Forecast EUR.USD

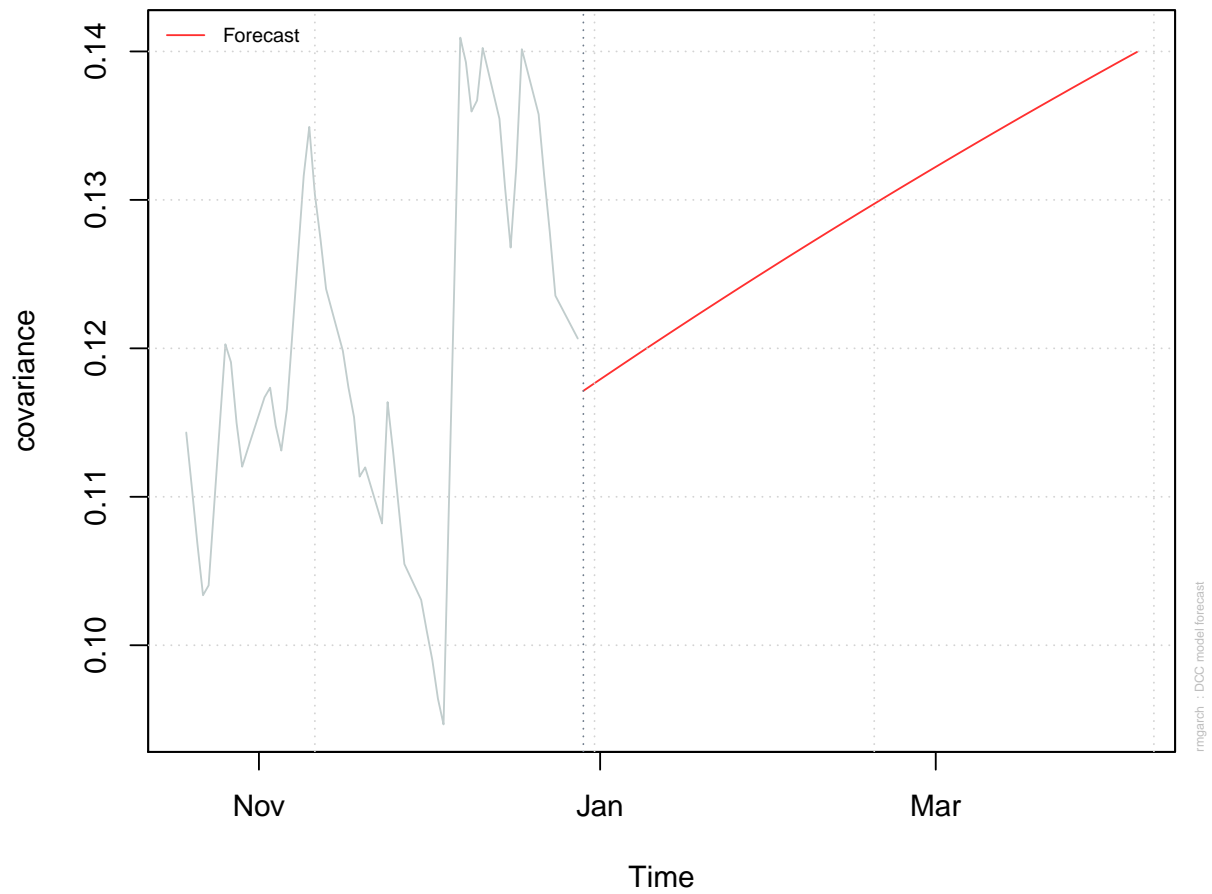


### GBP.USD



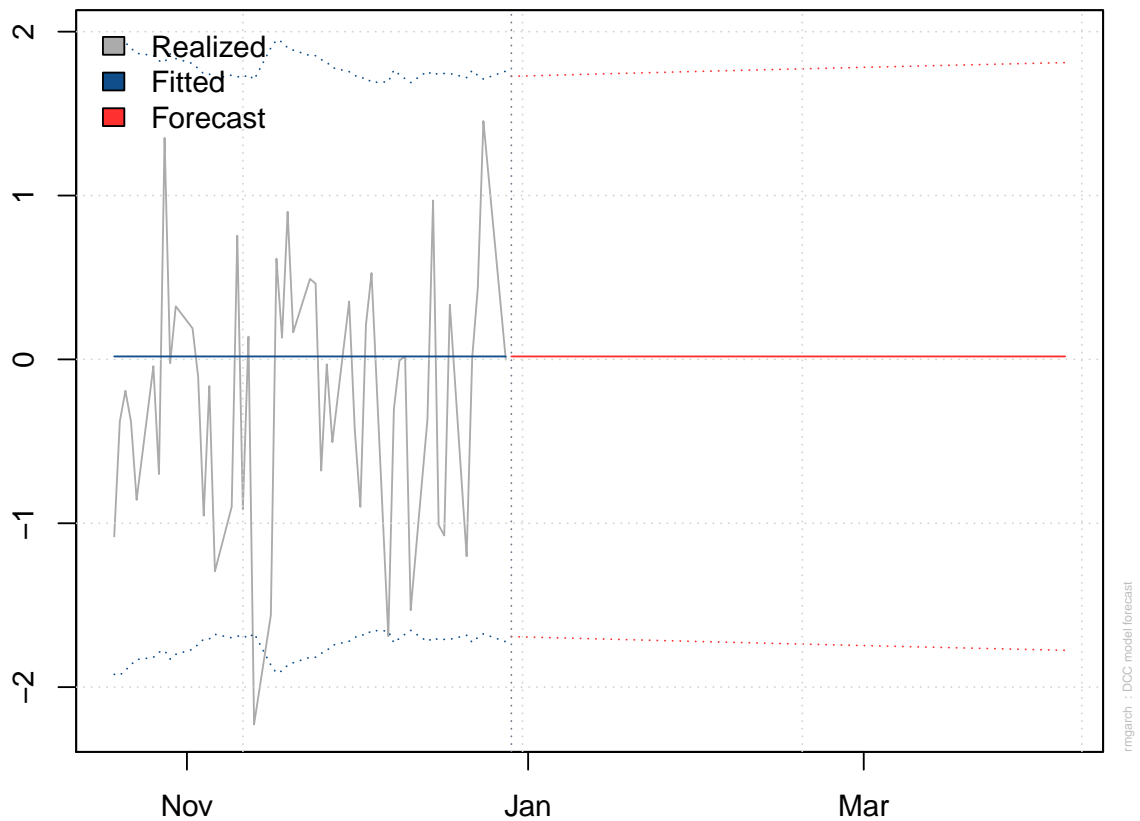
```
plot(dcc.fcst, which = 3)
```

## DCC Unconditional Covariance Forecast GBP.USD–EUR.USD



```
plot(dcc.fcst, which = 5)
```

### EW Forecast Portfolio with Unconditional 2.5% VaR Limits (based on Conditional Weighted Density)



```

dcc.residuals <- residuals(dcc.fit)
(Brent.dcc.var <- quantile(dcc.residuals$OIL.Brent, c(0.01, 0.05, 0.5, 0.95, 0.99)))

```

```

##          1%          5%          50%          95%          99%
## -6.13726785 -3.67712869 -0.00443754  3.39131486  5.89699481

```

```

(GBP.dcc.var <- quantile(dcc.residuals$GBP.USD, c(0.01, 0.05, 0.5, 0.95, 0.99)))

```

```

##          1%          5%          50%          95%          99%
## -1.339311287 -0.823506918 -0.000326409  0.765973270  1.246595209

```

```

(EUR.dcc.var <- quantile(dcc.residuals$EUR.USD, c(0.01, 0.05, 0.5, 0.95, 0.99)))

```

```

##          1%          5%          50%          95%          99%
## -1.520666396 -0.980794376  0.006889539  0.904772045  1.493169076

```

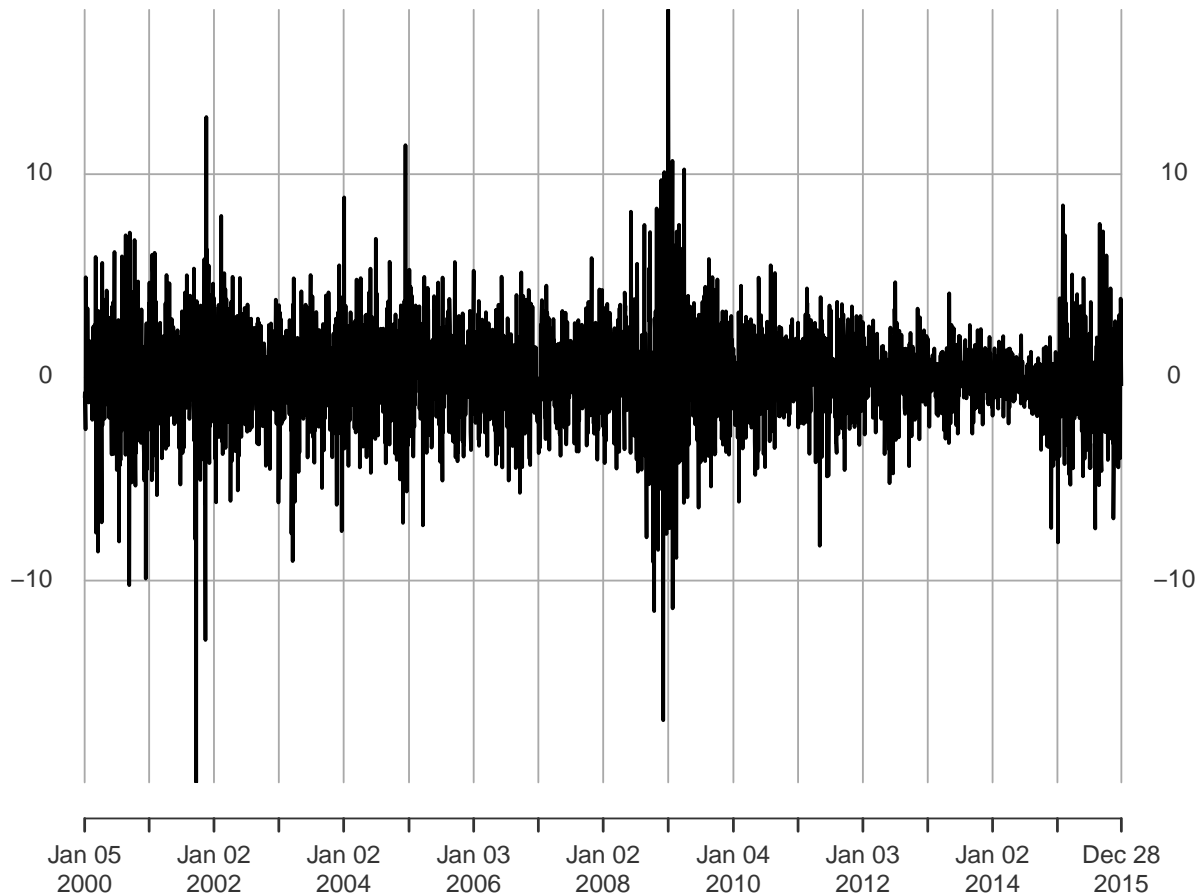
```

plot(dcc.residuals$OIL.Brent)

```

dcc.residuals\$OIL.Brent

2000-01-05 / 2015-12-28



What do we see?

1. A bit more heavily weighted in the negative part of the distributions.
2. Exchange rates are about the same as one another in this profile.
3. Brent is shocky at best: large moves either way.
4. If you use Brent contingent inputs (cost) in your production process you are naturally short Brent and would experience losses at the rate of 500% about 1% of the time. (99% CI)

## 7.6 Finishing

Back to Brent. Let's refit using the new volatility models and innovation distribution to capture asymmetry and thick tails.

```
Brent.spec <- ugarchspec(variance.model = list(model = "gjrGARCH",  
                                              garchOrder = c(1, 1)),  
                        mean.model = list(arimaOrder = c(1, 1),  
                                         include.mean = TRUE),  
                        distribution.model = "nig")
```

Here we experiment with a new distribution: the negative inverse gamma. Thick tails abound...

```
fit.Brent <- ugarchfit(spec = Brent.spec,
                      data = R$OIL.Brent,
                      solver.control = list(trace = 0))
```

Another 10 seconds (or so) of our lives will fit this models.

```
fit.Brent

##
## *-----*
## *          GARCH Model Fit          *
## *-----*
##
## Conditional Variance Dynamics
## -----
## GARCH Model   : gjrGARCH(1,1)
## Mean Model    : ARFIMA(1,0,1)
## Distribution   : nig
##
## Optimal Parameters
## -----
##      Estimate  Std. Error    t value Pr(>|t|)
## mu      -0.039926    0.027750 -1.4388e+00 0.150213
## ar1      0.996066    0.001902  5.2383e+02 0.000000
## ma1     -0.989715    0.000005 -1.8808e+05 0.000000
## omega    0.006344    0.003427  1.8509e+00 0.064181
## alpha1   0.009668    0.003844  2.5153e+00 0.011894
## beta1    0.968204    0.001237  7.8285e+02 0.000000
## gamma1   0.042778    0.007189  5.9502e+00 0.000000
## skew    -0.120148    0.032059 -3.7477e+00 0.000178
## shape    2.363709    0.351747  6.7199e+00 0.000000
##
## Robust Standard Errors:
##      Estimate  Std. Error    t value Pr(>|t|)
## mu      -0.039926    0.030725 -1.2995e+00 0.193784
## ar1      0.996066    0.002110  4.7215e+02 0.000000
## ma1     -0.989715    0.000005 -1.8395e+05 0.000000
## omega    0.006344    0.003389  1.8721e+00 0.061199
## alpha1   0.009668    0.004571  2.1153e+00 0.034407
## beta1    0.968204    0.000352  2.7482e+03 0.000000
## gamma1   0.042778    0.008515  5.0240e+00 0.000001
## skew    -0.120148    0.033158 -3.6235e+00 0.000291
## shape    2.363709    0.406328  5.8172e+00 0.000000
##
## LogLikelihood : -8508.439
##
## Information Criteria
## -----
##
## Akaike          4.1989
## Bayes           4.2129
## Shibata         4.1989
## Hannan-Quinn    4.2038
##
```

```

## Weighted Ljung-Box Test on Standardized Residuals
## -----
##               statistic p-value
## Lag[1]                1.856  0.1730
## Lag[2*(p+q)+(p+q)-1] [5]    2.196  0.9090
## Lag[4*(p+q)+(p+q)-1] [9]    2.659  0.9354
## d.o.f=2
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## -----
##               statistic p-value
## Lag[1]                0.5107 0.47484
## Lag[2*(p+q)+(p+q)-1] [5]    9.3918 0.01317
## Lag[4*(p+q)+(p+q)-1] [9]   13.2751 0.00921
## d.o.f=2
##
## Weighted ARCH LM Tests
## -----
##           Statistic Shape Scale P-Value
## ARCH Lag[3]      10.26 0.500 2.000 0.001361
## ARCH Lag[5]      10.41 1.440 1.667 0.005217
## ARCH Lag[7]      11.06 2.315 1.543 0.010374
##
## Nyblom stability test
## -----
## Joint Statistic:  2.528
## Individual Statistics:
## mu      0.90796
## ar1     0.06995
## ma1     0.06315
## omega   0.70757
## alpha1  0.22117
## beta1   0.28123
## gamma1  0.17735
## skew    0.25143
## shape   0.16525
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:      2.1 2.32 2.82
## Individual Statistic:  0.35 0.47 0.75
##
## Sign Bias Test
## -----
##           t-value      prob sig
## Sign Bias      1.1464 0.25169
## Negative Sign Bias 0.7507 0.45287
## Positive Sign Bias 1.8417 0.06559  *
## Joint Effect    9.7941 0.02040  **
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## -----
## group statistic p-value(g-1)

```

```
## 1    20    27.75    0.08841
## 2    30    46.52    0.02081
## 3    40    58.52    0.02302
## 4    50    69.71    0.02750
##
##
## Elapsed time : 3.148171
```

## 7.7 Example

Let's run this code and recall what the `evir` package does for us.

```
library("evir")
Brent.resid <- abs(residuals(fit.Brent))
gpdfit.Brent <- gpd(Brent.resid, threshold = quantile(Brent.r, 0.9))
(Brent.risk <- riskmeasures(gpdfit.Brent, c(0.9, 0.95, 0.975, 0.99, 0.999)))
```

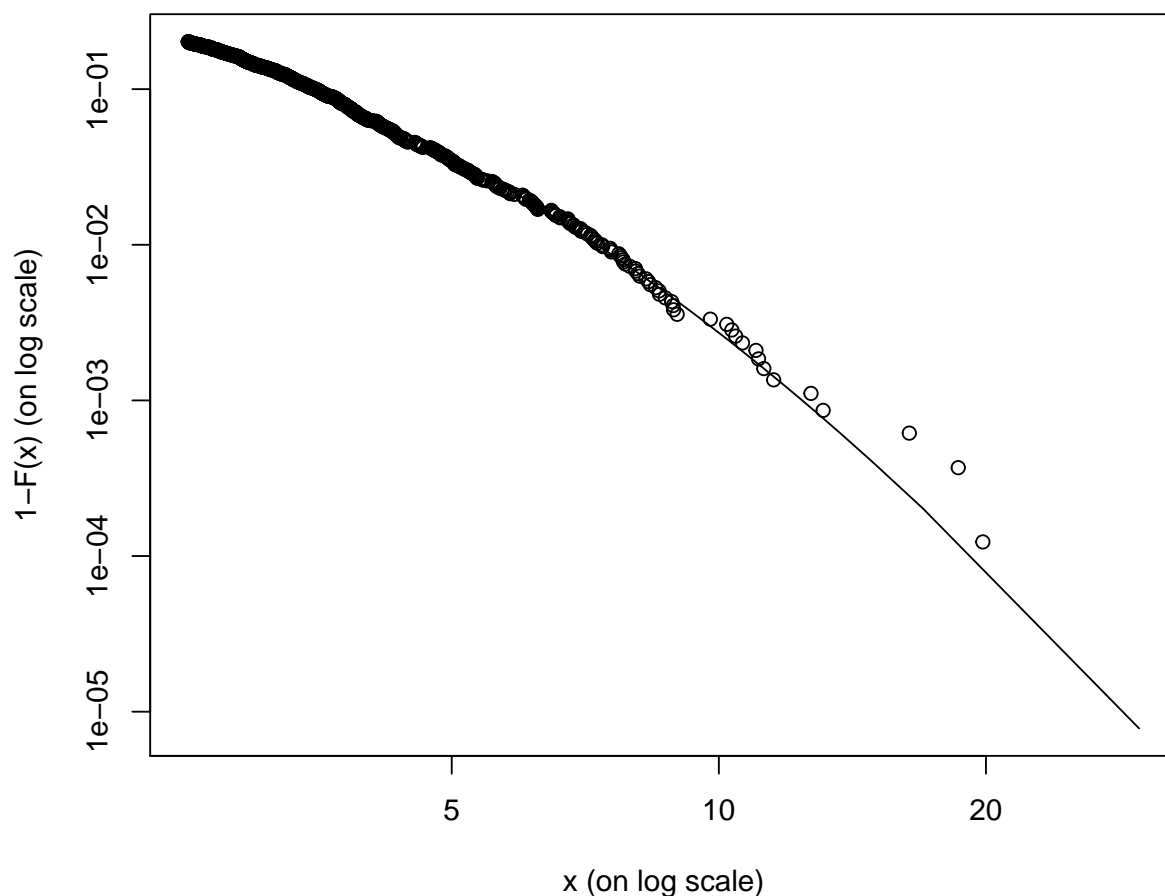
```
##           p quantile      sfall
## [1,] 0.900  3.478470  5.110313
## [2,] 0.950  4.509210  6.293452
## [3,] 0.975  5.636212  7.587085
## [4,] 0.990  7.289151  9.484419
## [5,] 0.999 12.415542 15.368767
```

What does this mean?

1.  $1 - p$  gives us levels of tolerance.
2. `quantile` gives us the value at risk (VaR)
3. `sfall` reports the expected short fall (ES)

From the `evir` package here is the tail lot

```
tailplot(gpdfit.Brent)
```



What does this mean?

- The results show much thick and volatile tail activity event with the **AR-GARCH** treatment.
- We could well go back to market and operational risk section to understand mean excess value (beyond thresholds) and the confidence interval for VaR and ES.
- For accounts recivable mitigation strategies might be to excess risk hedge provided through reinsurance and total return swaps.
- Credit risk analysis of customers is critical: frequent updates of Brent exposed customers will help to detect early on problems that might admit of some solutions.

## 8 Aggregating Enterprise Risk

### 8.1 Introduction: The Problem with Enterprise Risk

Corporate policy dictates that mangement must assess risks to equity annually and whether a circumstance dictates. Such a circumstance is an IPO.



Management knows of at least three material risks:

- Customers defect so there is uncertainty in revenue growth.
- Suppliers stop competing on price, quantity, and quality so there is uncertainty in variable expense.
- There are major compliance breaches which impact fixed expense.

No one knows much about these risks from history because this company is the first in its market to product a very innovative product. One question management knows *someone* will ask is how likely is it that the net operating margin will fall below, say, indicated earnings of \$400 million. (Venture risk)

## 8.2 Copula

Our problem is:

1. We have three major risk factors and each has their own distribution.
2. We also know wthat they are somehow correlated.
3. How can we aggregate the three into one risk measure that is tangible, and preserves the correlation?

### 8.2.1 From scratch

Our first task is to generate multivariate normal variates that are correlated with one another. Here we relate three standard normal random variables together. A standard normal random variable has a mean ,  $\mu = 0$ , and variance,  $\sigma^2 = 1$ . The variable `sigma` in the code below is the *correlation* matrix.

```
library("mvtnorm")
set.seed(1016)

n.risks <- 3 ## Number of risk factors
m <- n.risks

n.sim <- 1000

sigma <- matrix(c(1, 0.4, 0.2, 0.4, 1, -0.8, 0.2, -0.8, 1), nrow = 3)

z <- rmvnorm(n.sim, mean = rep(0, nrow(sigma)), sigma = sigma, method = "svd")

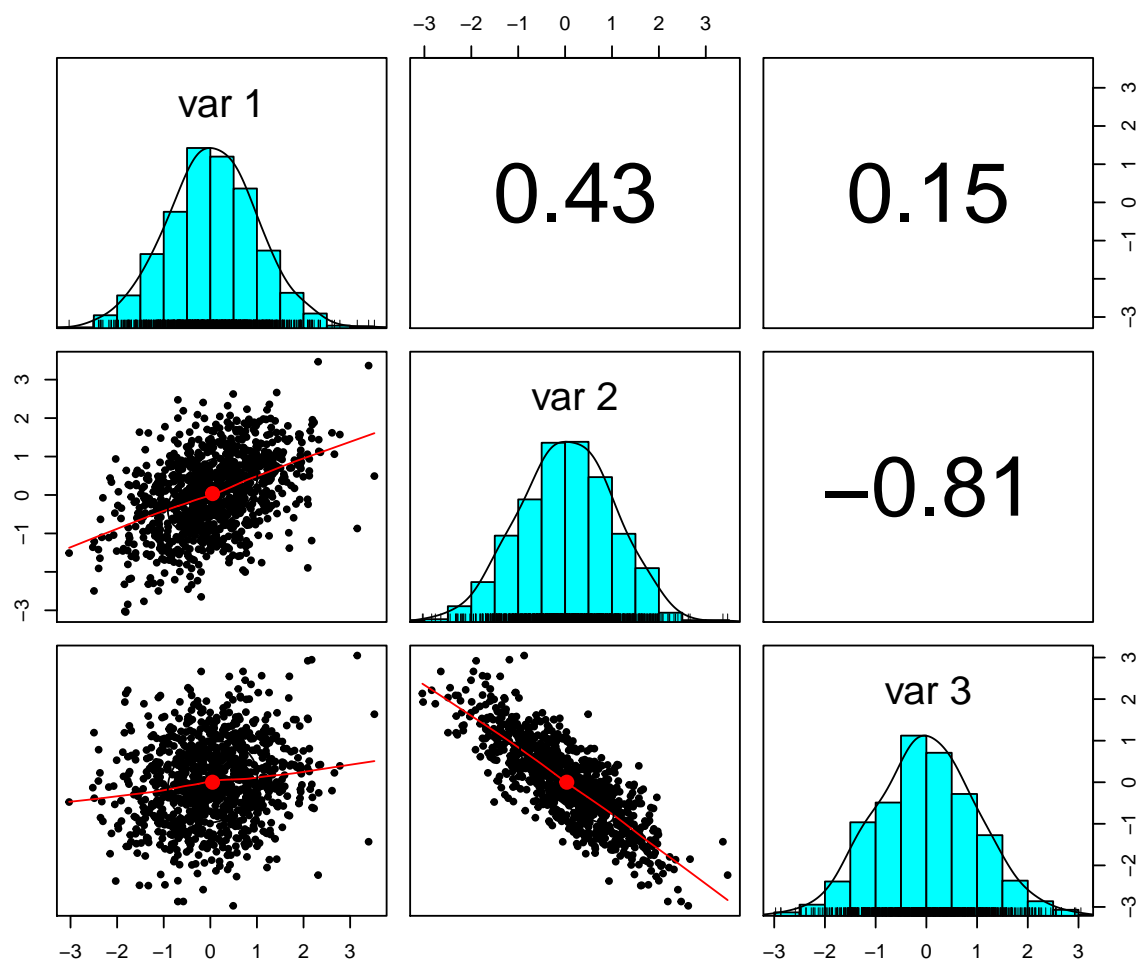
library(psych)
cor(z, method = "spearman")

##           [,1]      [,2]      [,3]
## [1,] 1.0000000 0.4080956 0.1388115
## [2,] 0.4080956 1.0000000 -0.8006036
## [3,] 0.1388115 -0.8006036 1.0000000

cor(z, method = "pearson")

##           [,1]      [,2]      [,3]
## [1,] 1.0000000 0.4256940 0.1518530
## [2,] 0.425694  1.0000000 -0.8110413
## [3,] 0.151853 -0.8110413 1.0000000

pairs.panels(z)
```



### 8.3 Sklar's in the house...

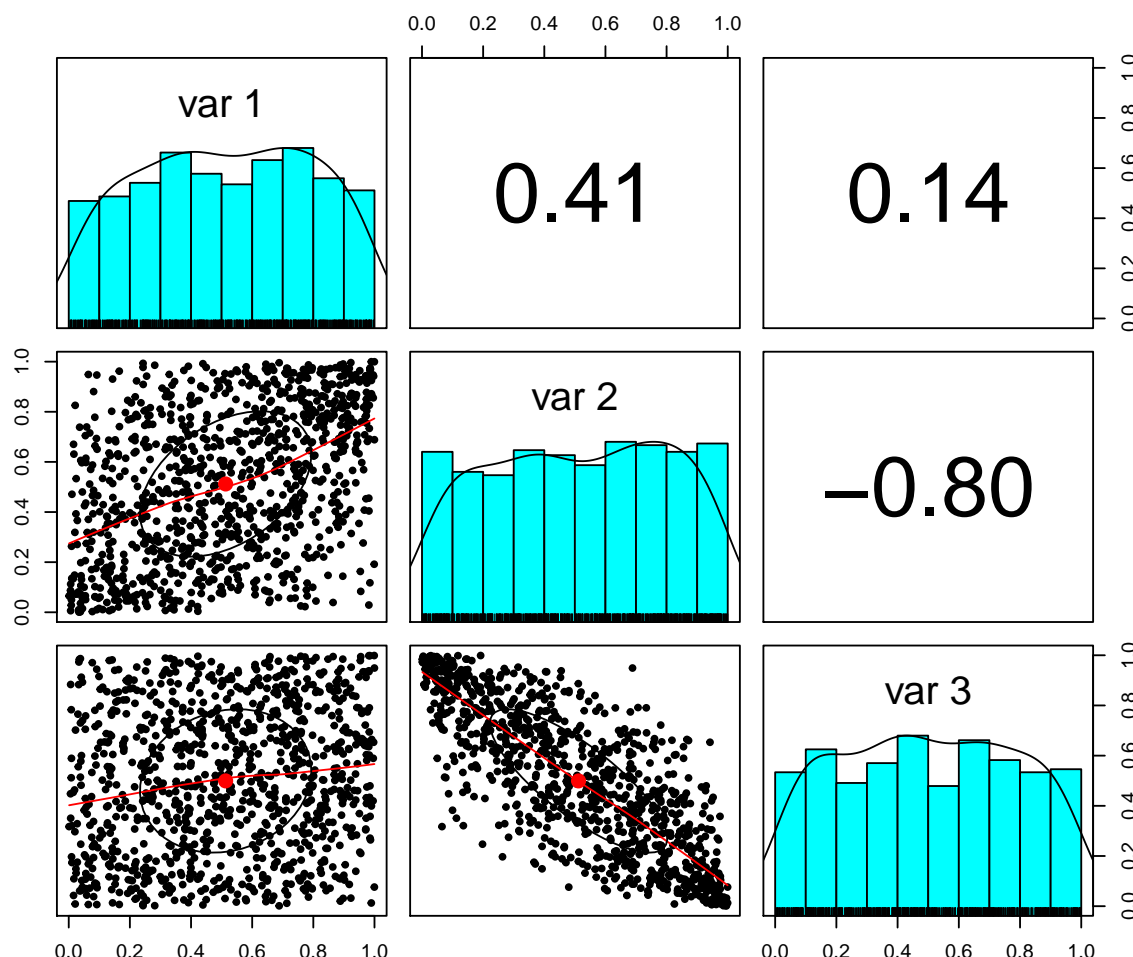
Next we use a result from mathematical probability called

**Sklar's Theorem (1959)**

- If  $x$  is a random variable with distribution  $F$ ,
- the  $F(x)$  is uniformly distributed in the interval  $[0,1]$ .

Let's translate this idea into R and look at the resulting interactions.

```
require(psych)
u <- pnorm(z)
pairs.panels(u)
```



We see that the Gaussian (normal) distribution has been reshaped into a uniform distribution just as Sklar predicted. The idea around this theorem is the same as around the number 1. We can multiply and real number by one and get the real number back. This is the identical operation. (Too funny, must apologize to mathematicians and statisticians). In somewhat analogous way, the uniform distribution serves a role as an distribution identity operator. When we operate on the uniformly distributed random numbers with a distribution, we get back that distribution. But in this case the identity distribution has structure in it (correlations) that the new distribution inherits.

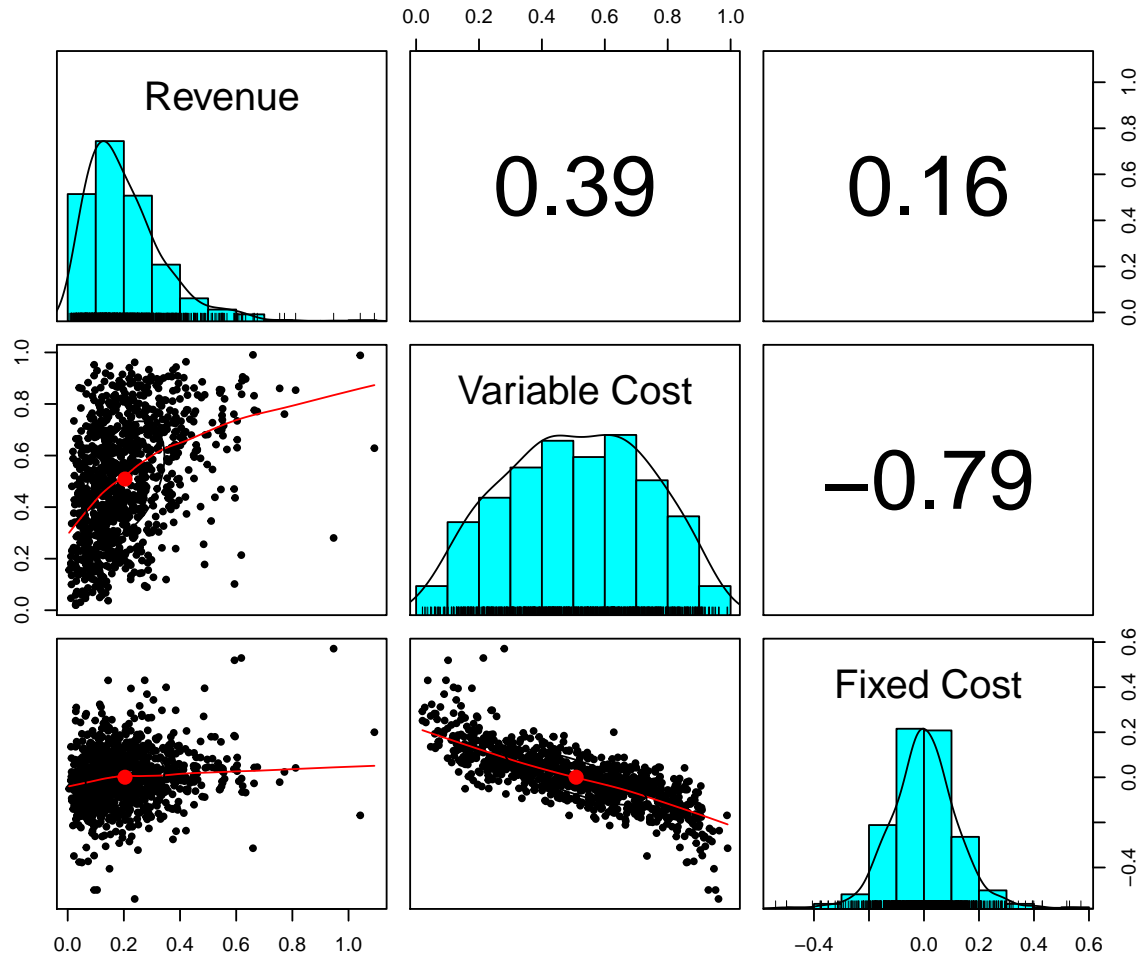
Now, we only need to select the marginal probabilities of the risks we are assessing and apply them to the dependently related  $u$  variates. Suppose the marginal probability distributions for revenue growth is  $\gamma$ , variable expense ratio is  $\beta$ , and the fixed expense ratio is Student's  $t$  distribution with parameters:

```
x1 <- qgamma(u[, 1], shape = 2, scale = 1)
x2 <- qbeta(u[, 2], 2, 2)
x3 <- qt(u[, 3], df = 5)
```

Nice outliers! Starting from a multivariate normal distribution we create dependent uniform variates. Using the dependent uniform variates we created dependent distributions of our choosing

```
factors.df <- cbind(x1/10, x2, x3/10)
```

```
colnames(factors.df) <- c("Revenue", "Variable Cost", "Fixed Cost")
pairs.panels(factors.df)
```



```
cor(factors.df, method = "spearman")
```

|               | Revenue   | Variable Cost | Fixed Cost |
|---------------|-----------|---------------|------------|
| Revenue       | 1.0000000 | 0.4080956     | 0.1388115  |
| Variable Cost | 0.4080956 | 1.0000000     | -0.8006036 |
| Fixed Cost    | 0.1388115 | -0.8006036    | 1.0000000  |

## 8.4 Analyze the results

Now to use all of this simulation to project revenue, expense, and margin.

```
revenue <- 1000 * (1 + factors.df[, 1])
variable.cost <- revenue * factors.df[, 2]
fixed.cost <- revenue * factors.df[, 3]
```

```
total.cost <- variable.cost + fixed.cost

operating.margin <- revenue - variable.cost - fixed.cost

analysis <- cbind(revenue, total.cost, operating.margin)

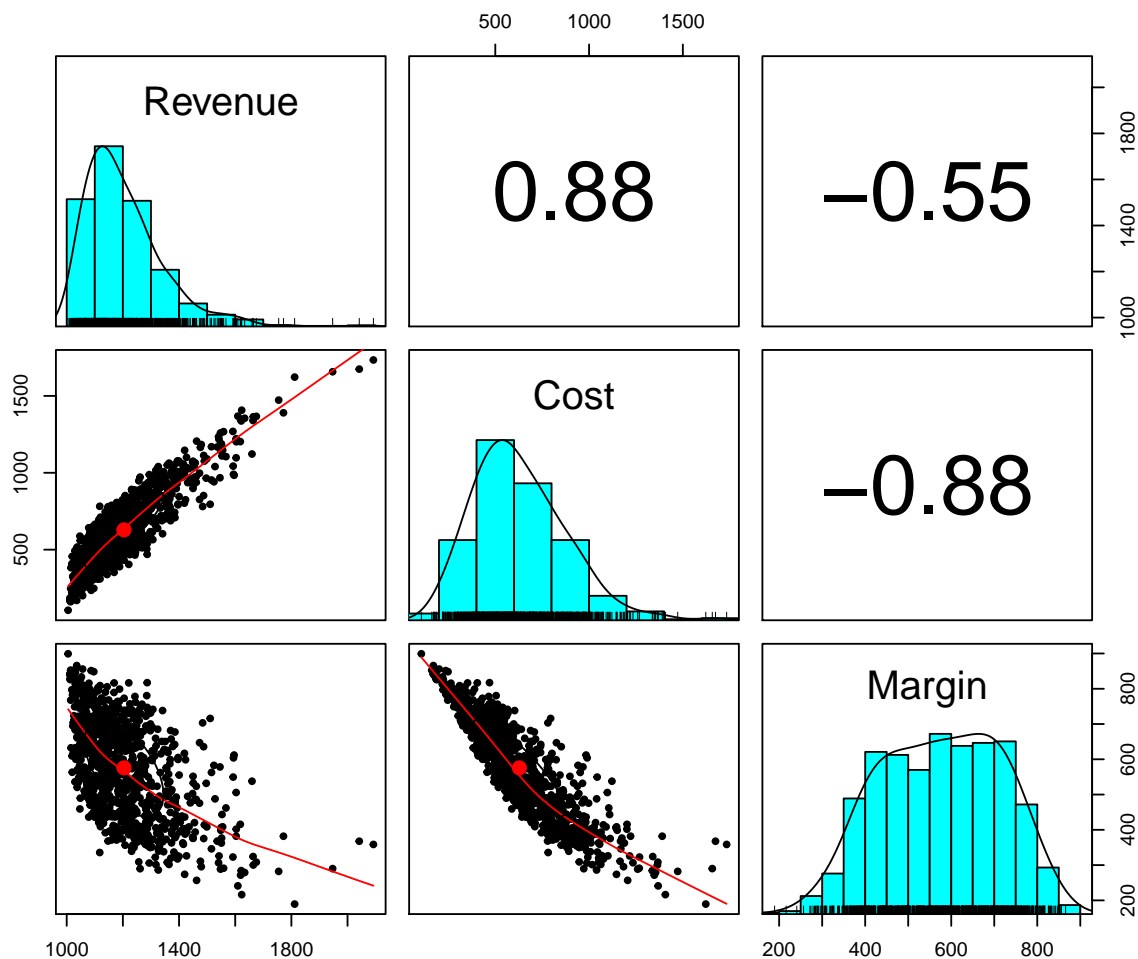
colnames(analysis) <- c("Revenue", "Cost", "Margin")
```

#### 8.4.1 Example

Run `pairs.panels` using the `analysis` data frame. What do you see?

Here's the result.

```
pairs.panels(analysis)
```



What do we see?

1. Variable and fixed cost aggregate into a distribution that is right-skewed.

2. Margin has a high density across a broad range of potential outcomes.
3. An increase (decrease) in cost will probably result in an increase (decrease) in revenue.
4. Revenue and margin also seem to be counter cyclical, a non-intuitive result, but one that makes sense only by looking at the negative correlation between cost and margin.

## 8.5 Risk Measures

We are not yet done. The whole point of this analysis is to get consistent and coherent measures of risk to a consumer of the analysis, namely, the decision maker who is the CFO in this case. We define the value at risk,  $VaR$ , as the  $\alpha$  quantile of the performance metric of interest. Higher  $\alpha$  means lower risk tolerance. Here is the relationship:

$$Q(x, \alpha) = F(x; \text{Prob}[X] > \alpha)$$

The metric  $x$  in this case is margin. Expected Shortfall,  $ES$ , is then the mean of the margin beyond  $VaR$ . The parameter  $\alpha$  is the level of organizational risk tolerance. If  $\alpha = 0.99$ , then the organization would want risk capital to cover the potential loss  $VaR$ , and more conservatively,  $ES$ . The organization is even more conservative the higher the  $\alpha$ .

We purloin the R code from the market risk material here:

```
## Simple Value at Risk
expected.margin <- 400
## Center margin loss on expected
## margin
loss.rf <- -(expected.margin - operating.margin)
## Assign metric of interest to
## reusable code
summary(loss.rf)

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -210.55  63.05  179.77  175.72  290.42  499.09

## Always review a key variable's
## content
alpha.tolerance <- 0.99

## Very intolerant! Remember that
## putting a variable assignment in
## parentheses also prints the result

(VaR.hat <- quantile(loss.rf, probs = alpha.tolerance, names = FALSE))

## [1] 436.8785

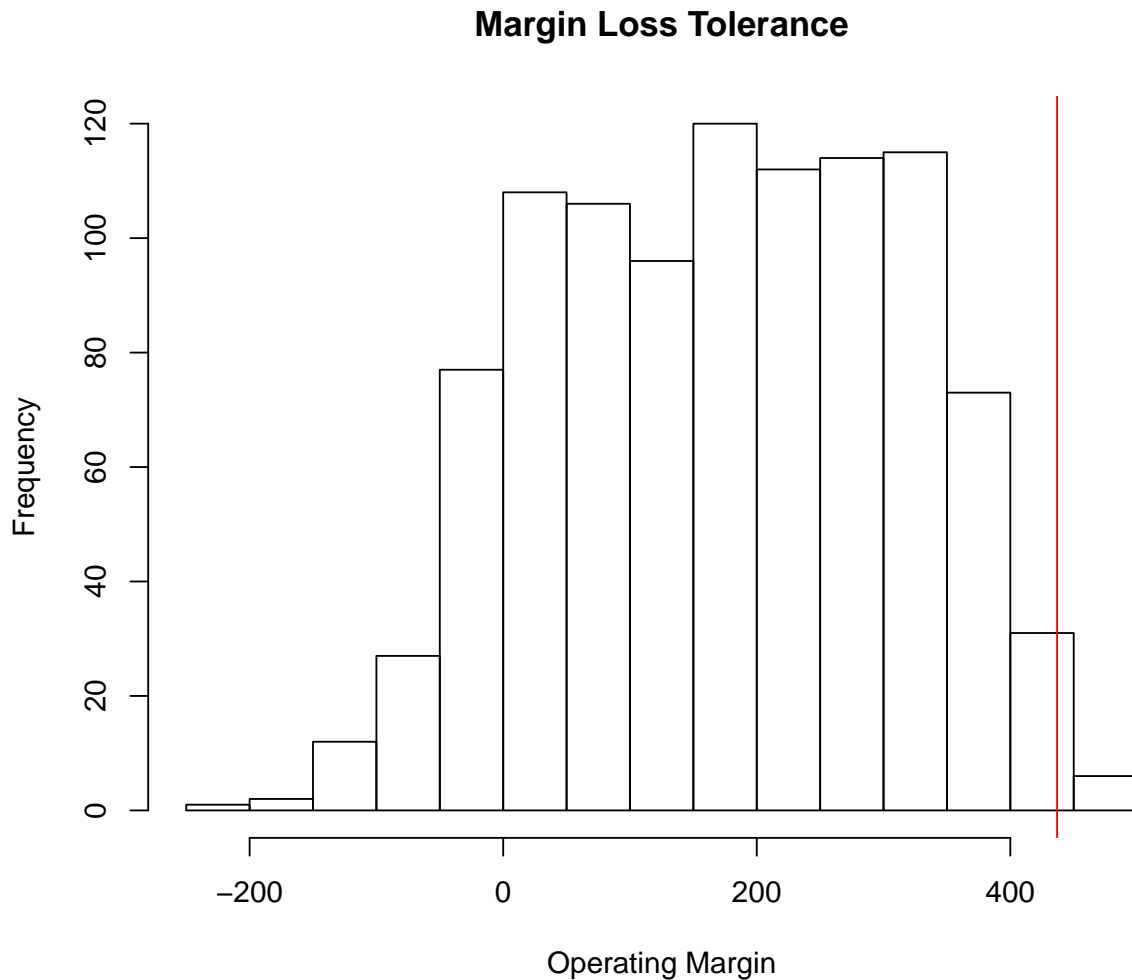
### Just a simple Expected shortfall
(ES.hat <- mean(loss.rf[loss.rf > VaR.hat]))

## [1] 455.7716
```

Let's plot the results

```
hist(loss.rf, xlab = "Operating Margin",
      ylab = "Frequency", main = "Margin Loss Tolerance")

abline(v = VaR.hat, col = "red")
```



Sklar provides us with a way to join together any set of distributions. It transforms correlated variates into uniform distribution. The uniform distribution takes on the role of the number 1 in algebra. Anything multiplied by 1 returns itself. In a very loose way, the uniform distribution is the identity distribution, just like one is the identity term in algebra. So that whenever we operate on the uniform distribution we get back the same distribution - but this time with correlation.

The rub is the starting point. Here we used the Gaussian (normal) distribution. This is not a very thick tailed distribution, and it can be shown that extreme events are not dependent on one another using this distribution. This is NOT a useful feature ultimately. SO analyst use thickly tailed distributions such as the Student-t and the generalized Pareto distribution (GPD) to get dependency far out into the tails. This is nearly perfect for risk managers and decision makers.

#### 8.5.1 Example

Let's use this R code to modify the copula making machine we just built. Instead of `rmvnorm` we will use `rmvt` to generate the correlated risk factors. The is called a **t-copula**.

```
library(mvtnorm)
library(psych)
```

```

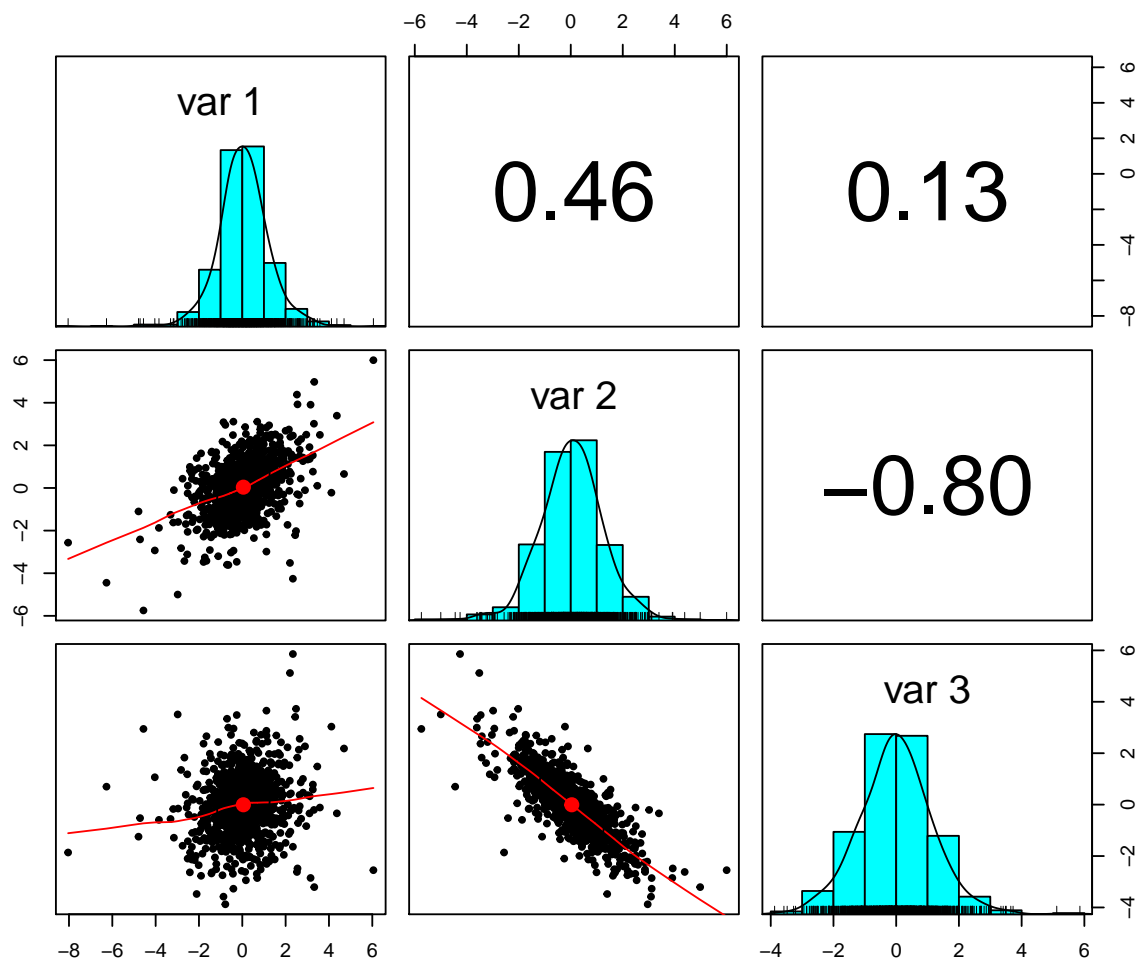
set.seed(1016)

n.risk <- 3
m <- n.risk
n.sim <- 1000
sigma <- matrix(c(1, 0.4, 0.2, 0.4, 1, -0.8, 0.2, -0.8, 1), nrow = m)
z <- rmvt(n.sim, delta = rep(0, nrow(sigma)),
          sigma = sigma, df = 6, type = "shifted")

```

Here are the results of our experiment. Let's go through the paces. First we look at the  $z$  variates we simulated using the multivariate Student's  $t$ -distribution.

```
pairs.panels(z)
```



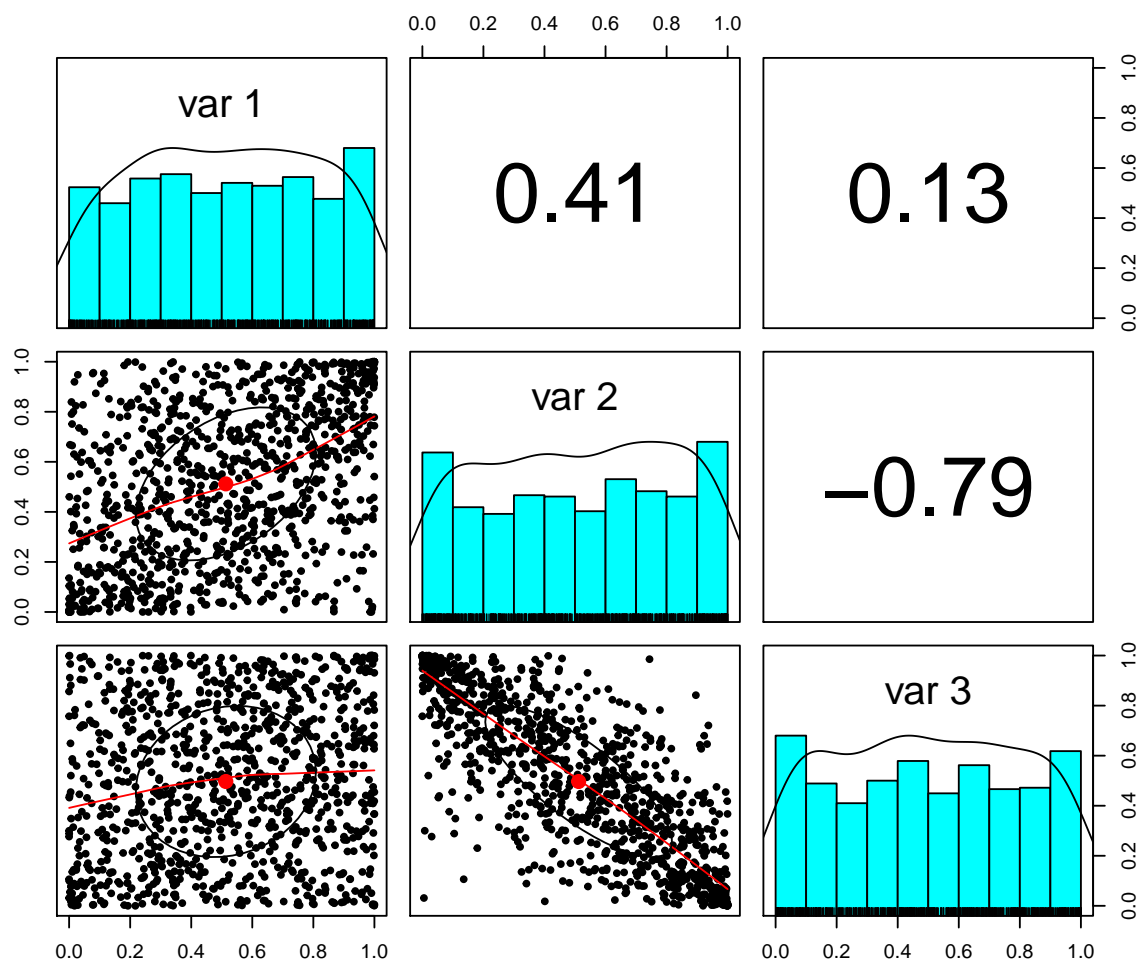
We then run the uniform distribution generator (with correlation structure).

```

require(psych)
u <- pnorm(z)
pairs.panels(u)

```





Now, we only need to select the marginal probabilities of the risks we are assessing and apply them to the dependently related 'u' variates. Again suppose the marginal probability for revenue growth is `gamma`, for these variables expense ratio is `beta`, and fixed expense ratio is Student's t distribution with these parameters:

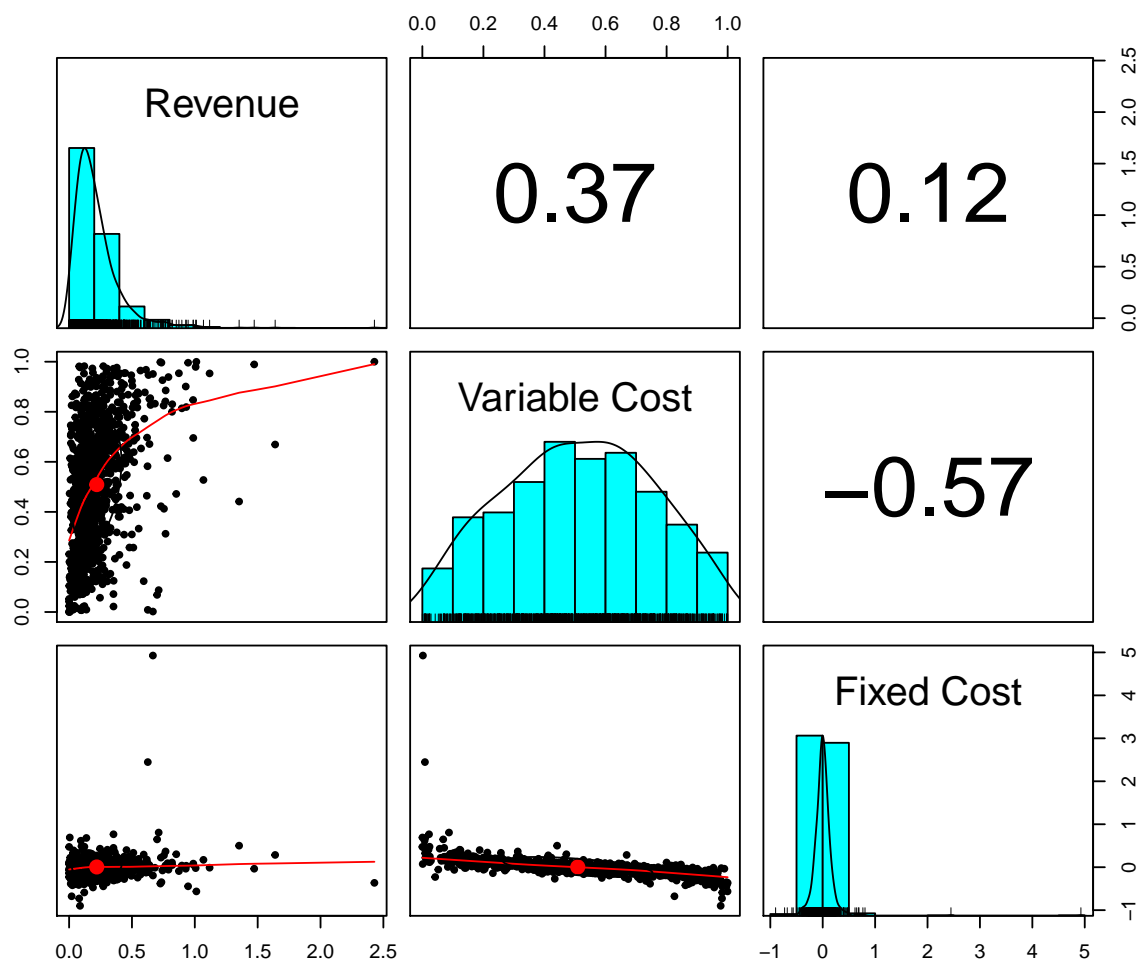
```
x1 <- qgamma(u[, 1], shape = 2, scale = 1)
x2 <- qbeta(u[, 2], 2, 2)
x3 <- qt(u[, 3], df = 6)
```

Starting from a multivariate Student's t-distribution we created dependent uniform variates. Using the dependent uniform variates we created dependent distributions of our choosing.

Next, we combine the series into a data frame and review the scatterplot matrix.

```
factors.df <- cbind(x1/10, x2, x3/10)
colnames(factors.df) <- c("Revenue", "Variable Cost", "Fixed Cost")

pairs.panels(factors.df)
```



```
cor(factors.df, meth = "spearman")
```

```
##           Revenue Variable Cost Fixed Cost
## Revenue      1.0000000      0.4067919  0.1248762
## Variable Cost 0.4067919      1.0000000 -0.7944976
## Fixed Cost    0.1248762     -0.7944976  1.0000000
```

## 9 Finished

### 9.1 Congratulations