

## Week 5: Market Risk

Copyright 2016, William G. Foote, all rights reserved

# Imagine this

For our working example we will consider an organization that uses electricity to produce steel.

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

- 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.

## 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.
- 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.
- ASSUMPTION: 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.

# Try this...

- 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 points 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.

Thinking...

- 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
  - Amount of power (MWh)
  - Amount of steel (tons)
  - Steel (tons) / Power (MWh) transformation rate
  - Start-up, idle, and shut-down timing and costs
  - Plant production costs and optimal scheduling

**Whitman**  
SCHOOL *of* MANAGEMENT  
**SYRACUSE UNIVERSITY**

---

**MBA@SYRACUSE**



## Outcomes this week

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

**Whitman**  
SCHOOL *of* MANAGEMENT  
**SYRACUSE UNIVERSITY**

---

**MBA@SYRACUSE**

# 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 factors we combine them into a portfolio and calculate their losses.
- Finally with the loss distribution in hand we can compute the risk measures.
- This approach is nonparametric.

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

Read in the csv file from the working directory. This file contains dates and several risk factors: RWE will stand in for electricity inputs and THYSSEN for steel outputs.

```
## Download the data
data.all <- read.csv("data/eurostock.csv",
  stringsAsFactors = FALSE)
# This will convert string dates to
# date objects below
str(data.all) # Check the structure and look for dates
```

```
## '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 ...
## $ DRESDBER.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 ...
```

The way we do that is to make the dates into row names so that dates are the index for the two risk factors. 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-
```

```
rownames(data.all) <- date
head(data.all)
```

```
##           X ALLIANZ.HLDG.  COMMERZBANK  DRESDNER.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      37.88 15.85 12.71      23.66      23.40 210.11
```

and...

```
## 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
```

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 job 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
```

# Try this

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 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.

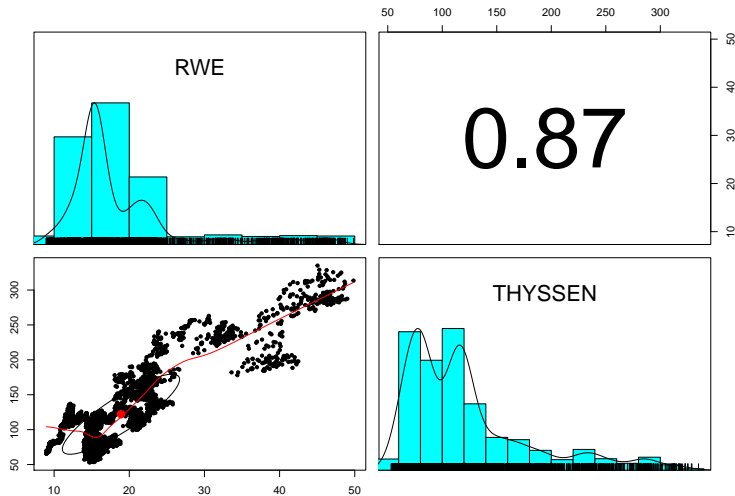
```
## Use scatter plots of the two price series along with their histograms to examine
library(psych)
pairs.panels(price)
price.rownames <- rownames(price)
plot(as.Date(price.rownames), price[, "THYSSEN"], type="l",
     main="Thyssen stock price data", # title
     xlab="Date t", # x-axis label
     ylab=expression(Stock-price-price[t])) # y-axis label
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
```



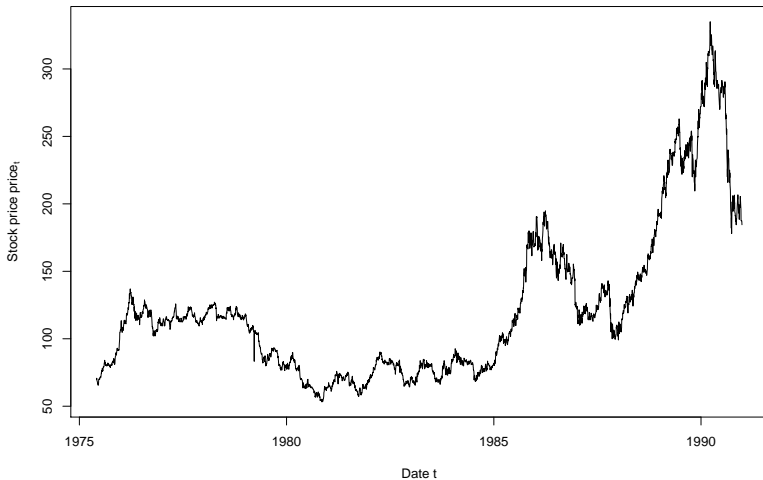
Thinking...

# Results

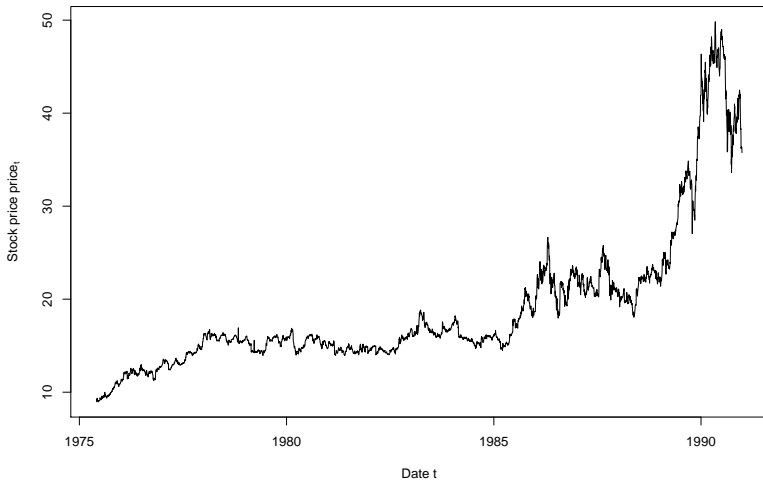
First the pairs plot that displays a matrix of interactions between RWE and THYSSEN:



Thyssen stock price data



RWE stock price data



**Whitman**  
SCHOOL *of* MANAGEMENT  
**SYRACUSE UNIVERSITY**

---

**MBA@SYRACUSE**

Prices are interesting but, as we have seen, are not stable predictors. Let's transform them to returns next.

**Whitman**  
SCHOOL *of* MANAGEMENT  
**SYRACUSE UNIVERSITY**

---

**MBA@SYRACUSE**

# Now to the matter at hand

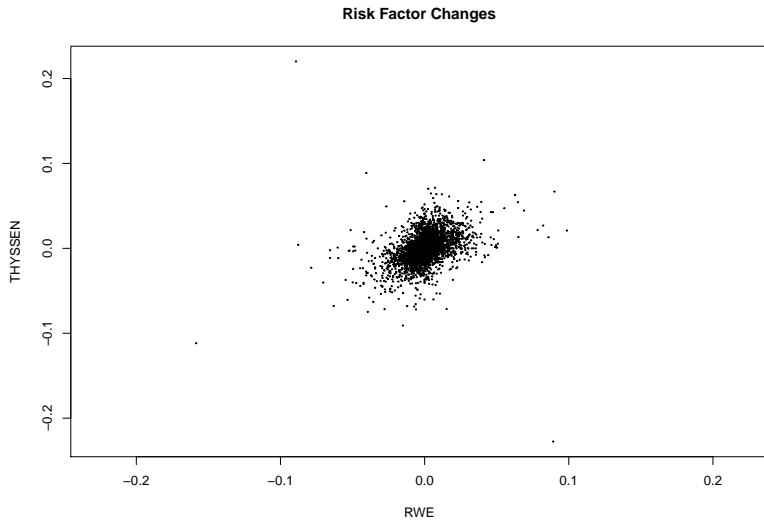
Now to the matter at hand: *value at risk* and *expected shortfall*. These two measures are based on the quantiles of losses attributable to risk factors. Value at risk is the quantile at an  $\alpha$  level of tolerance. Expected shortfall is the mean of the distribution beyond the value at risk threshold.

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. 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
plot(return.series, xlim = return.range,
      ylim = return.range, main = "Risk Factor Changes",
      cex = 0.2)
```



```
## [1] -0.2275282  0.2201375
```



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:

$$\text{Margin} = \text{price}_{\text{steel}} \times \text{tons} - \text{price}_{\text{power}} \times [(\text{rate}_{\text{MWh/tons}} \times \text{tons})],$$

where the last term is the power to steel transformation rate that converts power prices \$ per MWh to \$ per ton.

We convert prices to share prices and tons to equivalent values in terms of the number of shares. The position in power is equivalent to a (negative) number of shares (in the square brackets). The position in steel is equivalent to a (positive) number of shares.

```

# 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)

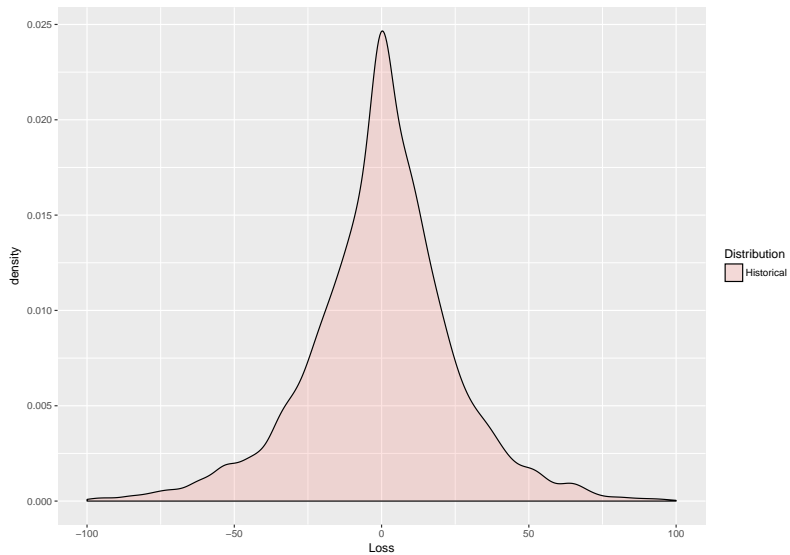
```

```

##      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 plot reveals some interesting deep and shallow outliers. The distribution is definitely 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 accurate calculations. The function `expm1` does this well.

Now we can 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 a 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

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

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  $VaR_\alpha$  definition we can also define  $ES$  as

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

where  $ES$  is “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  $ES \geq VaR$ .



# Try this

- 1 Run the following lines of code.
- 2 Look up `quantile` in R and see that it matches the calculation for `VaR.hist`.
- 3 ES is calculated off of VaR, but only looking for losses greater than VaR.
- 4 Look closely at the text annotations we can achieve in `ggplot2`.

Here is the code:

```
## Simple Value at Risk
alpha.tolerance <- 0.99
(VaR.hist <- quantile(loss.rf, probs = alpha.tolerance,
  names = FALSE))
## Just as simple Expected shortfall
(ES.hist <- mean(loss.rf[loss.rf > VaR.hist]))
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)
```

Thinking...

# Results

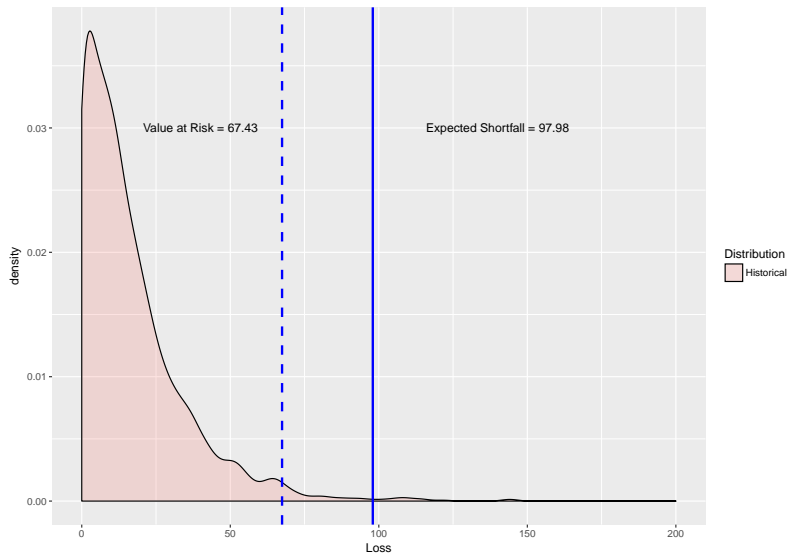
First the computations of  $VaR$  and  $ES$ :

```
## 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]))
```

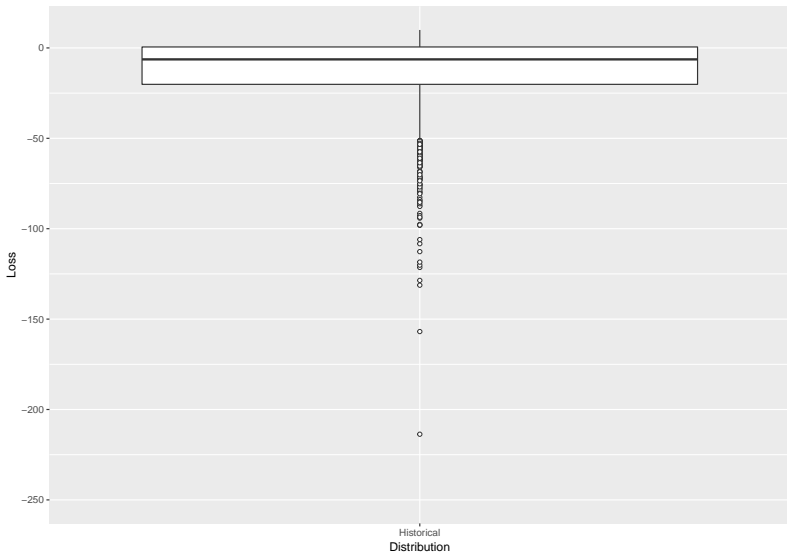
```
## [1] 97.97649
```



- 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)
```





**Whitman**  
SCHOOL *of* MANAGEMENT  
**SYRACUSE UNIVERSITY**

---

**MBA@SYRACUSE**

- This box plot might look better with more than one distribution.
- Next we start to put some different shapes into the loss potential of our tolling agreement.

**Whitman**  
SCHOOL *of* MANAGEMENT  
**SYRACUSE UNIVERSITY**

---

**MBA@SYRACUSE**

# 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  $t$ ) distributions.

- Carl Friedrich Gauss is often credited with the discovery of the normal distribution. So we tack his name often enough to 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 is not exhibit thick tails, and definitely 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 squared normal random variables  $x$  with density function and  $k$  degrees of freedom for  $x > 0$ :

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

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 distributions 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. (Lookup this curve in Wikipedia among other references.)



- Here is a quick comparison of the standard Gaussian and the Student's t distributions.
- The functions `rnorm` and `rt` generate Gaussian and Student's t variates, respectively.
- The functions `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))
```

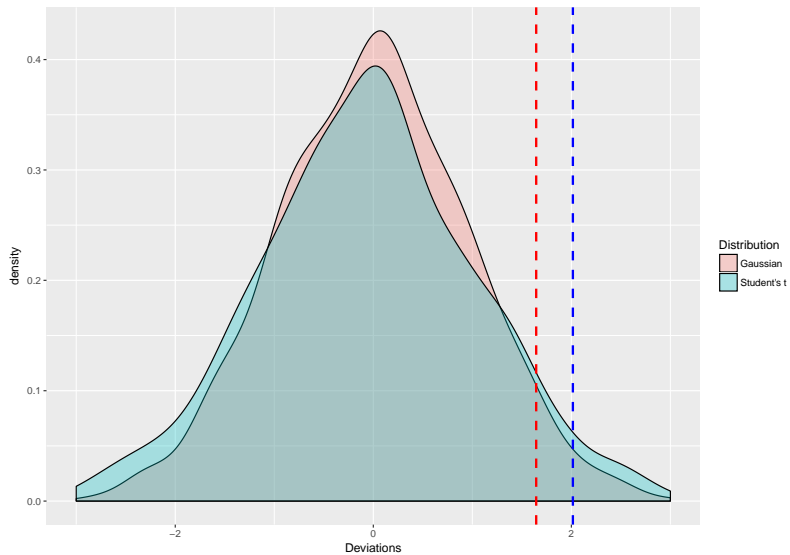
```
## [1] 1.644854
```

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

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

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)
```



- The `ggplots2` library allows us to control several aspects of the histogram including fill, borders, vertical lines, colors, and line types and thickness.
- The plot requires a data frame where we have indicated the type of distribution using a replication of character strings.
- 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 *t* density overtakes the red *z* density. - This is numerically evident as the `t.threshold` is  $>$  than the `z.threshold` for a cumulative probability of 95%, the 95th quantile.

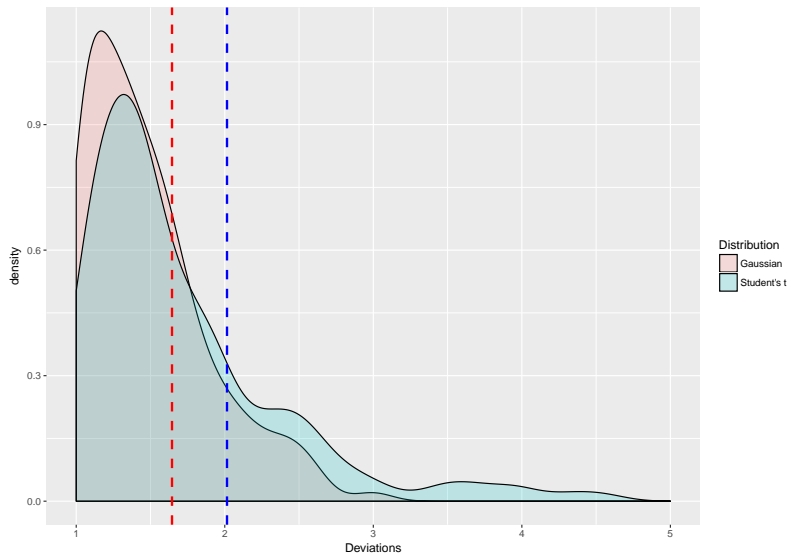
# Try this...

Let's zoom in on the right tail with the `xlim` facet.

```
ggplot(zt.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)
```

Thinking...

# Results



## Interesting digression!

- But not really not 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*.



**Whitman**  
SCHOOL *of* MANAGEMENT

**SYRACUSE UNIVERSITY**

---

**MBA@SYRACUSE**

# Back to the future

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

```
## Again computing returns as changes
## in the risk factors
return.series <- apply(log(price), 2,
  diff) # compute risk-factor changes
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 if stock, number of \$1 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  
position.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*, 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] -0.07596846
```

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

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

*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)
```

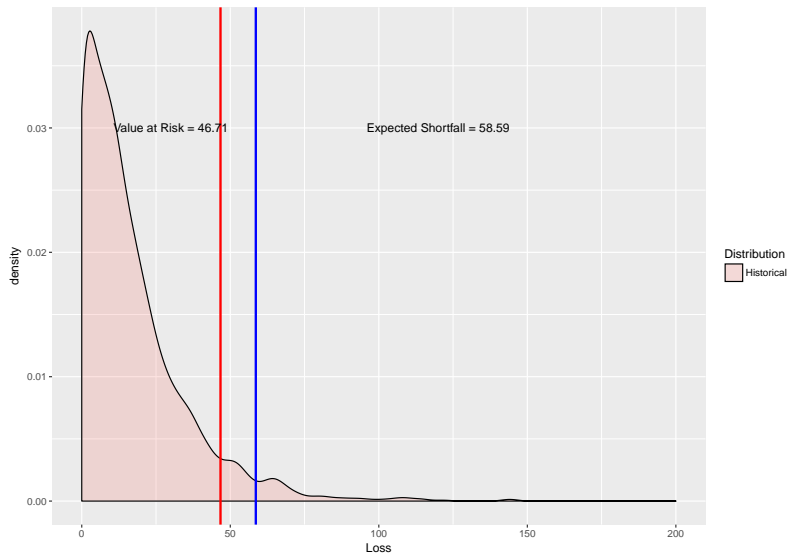
```
##           [,1]
## [1,] 46.70877
```

```
(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)
```



## Try this...

Suppose it takes less electricity to make steel. Model this by changing the positions to  $(-20, 10)$ . Redo steps 1, 2, and 3. What happened?



Thinking...

# Results

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

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

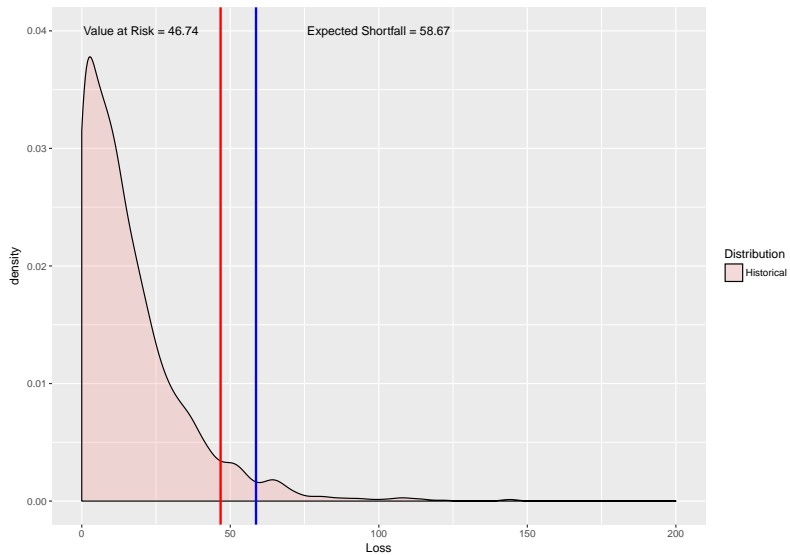
```
##           [,1]
## [1,] 46.7424
```

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

```
##           [,1]
## [1,] 58.66708
```

... 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 = 20, y = 0.04,  
    label = VaR.text) + annotate("text",  
  x = 100, y = 0.04, label = ES.text)
```



Aesthetics: change the x and y annotate coordinates to fit on the graph properly.

**Whitman**  
SCHOOL *of* MANAGEMENT  
**SYRACUSE UNIVERSITY**

---

**MBA@SYRACUSE**



- So ends the story of the main method used for years and embodied in the famous 4:15 (pm, that is) report at JP Morgan.
- Also 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.

**Whitman**  
SCHOOL *of* MANAGEMENT  
**SYRACUSE UNIVERSITY**

---

**MBA@SYRACUSE**

# Let's go to extremes

- All along we have been stylizing financial returns, including commodities and exchange rates, as skewed and with thick tails.
- We next go on to an extreme tail distribution called the Generalized Pareto Distribution (GPD).
- For very high thresholds, 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 we well exceeded all Gaussian and Student's  $t$  thresholds.

For a random variate  $x$ , this distribution is defined for the shape parameters  $\xi \geq 0$  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 one reason for GPD's notoriety. . . \*\*\* ## The notorious property: - If  $u$  is an upper (very high) threshold, then the excess of threshold function for the GPD is

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

- This simple measure is *linear* in thresholds. - It will allow us to visualize where rare events begin (see McNeil, Embrechts, and Frei (2015, chapter 5)). - We will come back to this property when we look at operational loss data in a later segment.

```
library(QRM)
u <- quantile(loss.rf, alpha.tolerance,
              names = FALSE)
fit <- fit.GPD(loss.rf, threshold = u) # Fit GPD 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!) using the McNeil, Embrechts, and Frei (2015, chapter 5) formulae:

```
# 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
```

# Try this...

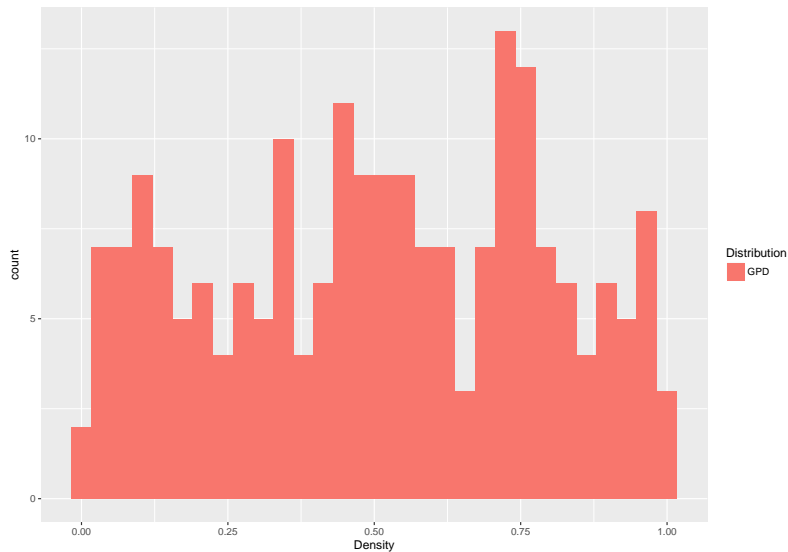
How good a fit? This plot should look roughly uniform since the GPD excess loss function is a linear function of thresholds  $u$ .

```
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[0,1]  
ggplot(gpd.density.df, aes(x = Density,  
  fill = Distribution)) + geom_histogram()
```

Thinking...



# Result



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

**Whitman**  
SCHOOL *of* MANAGEMENT  
**SYRACUSE UNIVERSITY**

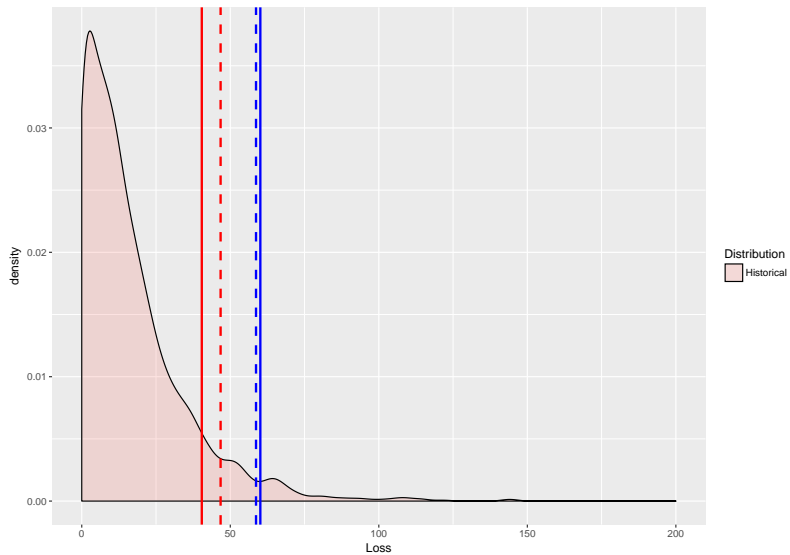
---

**MBA@SYRACUSE**

# 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),
  colour = "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.plot <- 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. . .

**Whitman**  
SCHOOL *of* MANAGEMENT  
**SYRACUSE UNIVERSITY**

---

**MBA@SYRACUSE**

# What have we done this week?

- More and More R, Finance, Risk, Statistics, and Probability
- Filtering
- Math to R translation
- Graphics
- Normal and GPD distributions
- VaR and ES
- Loss distributions and mean over loss



# To Prepare for the live session:

## List these:

- 1 What are the top 3 key learnings for you?
- 2 What pieces are still a mystery?
- 3 What parts would you like more practice on?
- 4 Review the assignment. What questions do you have about the assignment for the live session?

Thanks!

**Whitman**  
SCHOOL *of* MANAGEMENT  
**SYRACUSE UNIVERSITY**

---

**MBA@SYRACUSE**