Computer Simulations and Risk Assessment – Lecture 5

Fall 2019

Brandeis International Business School

Course Information - Schedule

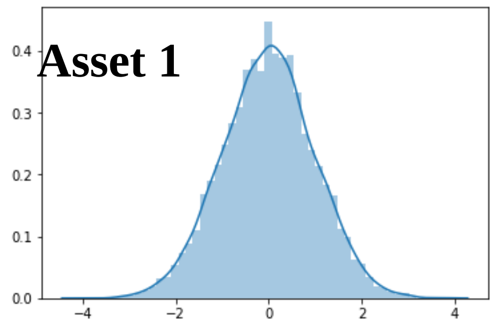
Class Date	Text Chapters
Aug. 30, 2019 – L1	• Course Introduction/Python Installation • Introduction to Quantitative Finance Career • Python basics
Sep. 6, 2019 – L2	• Advanced Python Topics
Sep. 13, 2019 – L3	• Advanced Python Topics
Sep. 20, 2019 – L4	• Sourcing and handling Data • Stylized financial data analysis using Python
Sep. 27, 2019 – L5	• Value at Risk
Oct. 4, 2019 – L6	• Conditional Value at Risk (Expected Shortfall) + Mid-term Review
Oct. 11, 2019	• Mid-term
Oct. 18, 2019 – L7	• Modeling Volatility I
Oct. 25, 2019 – L8	• Modeling Volatility II
Nov. 1, 2019 – L9	• Practical application case Studies I
Nov. 8, 2019 – L10	• Practical application case Studies II
Nov. 15, 2019 – L11	• Back Testing + Conditional risk prediction
Nov. 22, 2019 – L12	• Research project presentation
Dec. 6, 2019 – L13	• Final Review

Risk Measures and Value at Risk Basics

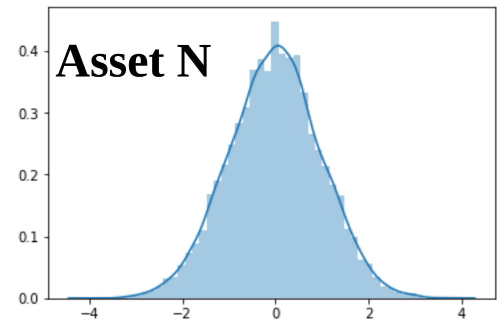
- **Introduction to risk measures**
- **Introduction to Value-at-Risk (VaR)**
- **VaR calculations**
- **Interpreting and analyzing VaR**

Foundation of a Modern Risk System

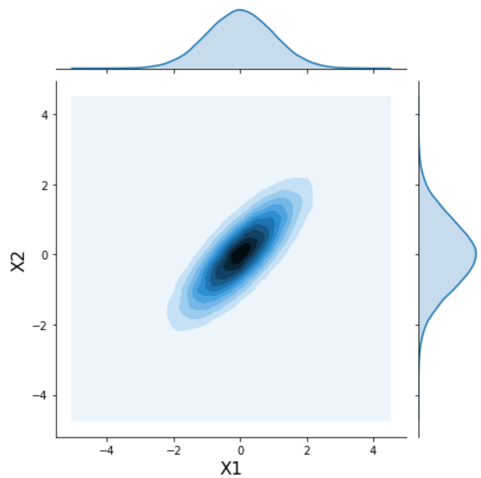
1. For each asset (risk factor), develop assumptions about distribution of its returns



....



2. Incorporate assumptions about correlations of these returns



3. Combining weights (exposure to risk factors)

$$\mathbf{W}'\boldsymbol{\Sigma}\mathbf{W}$$

here **W** is the weights vector and **Σ** is the covariance matrix

Introduction to Risk Measures

- There are different risk measures for different finance applications
  - Volatility – standard deviation of fund returns
  - Covariance – Measure that combines correlation and volatilities
  - Tracking Error - standard deviation of fund vs. benchmark relative returns (relative risk)
  - Value at risk – E.g., there is 5% probability that your portfolio's value will go down at least $x or more (absolute risk) - $x here is the VaR at 5% threshold
  - Expected shortfall/ Conditional VaR – how much can your portfolio go down on average under the 5% worst possible scenarios (absolute risk)
- Value at risk (VaR) and Expected Shortfall (ES) are measures that focus on tail risks
- How do these measures relate to the risk management framework?

Mean, Standard Deviation, Covariance

- Portfolio return mean and standard deviation

$$\mu_k = \frac{1}{N}\sum_{t=1}^{N} R_{t,k}$$

$$\sigma_k = \sqrt{\frac{1}{N-1}\sum_{t=1}^{N}\left(R_{t,k} - \mu_k\right)^2}$$ (Also called volatility）

- Correlation and Covariance

$$\rho = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}}$$

$$cov = \sigma_{ij} = \frac{1}{N-1}\sum_{i=1}^{N}\left(x_{ij} - \overline{x_j}\right)\left(x_{ik} - \overline{x_k}\right)$$

Tracking Error

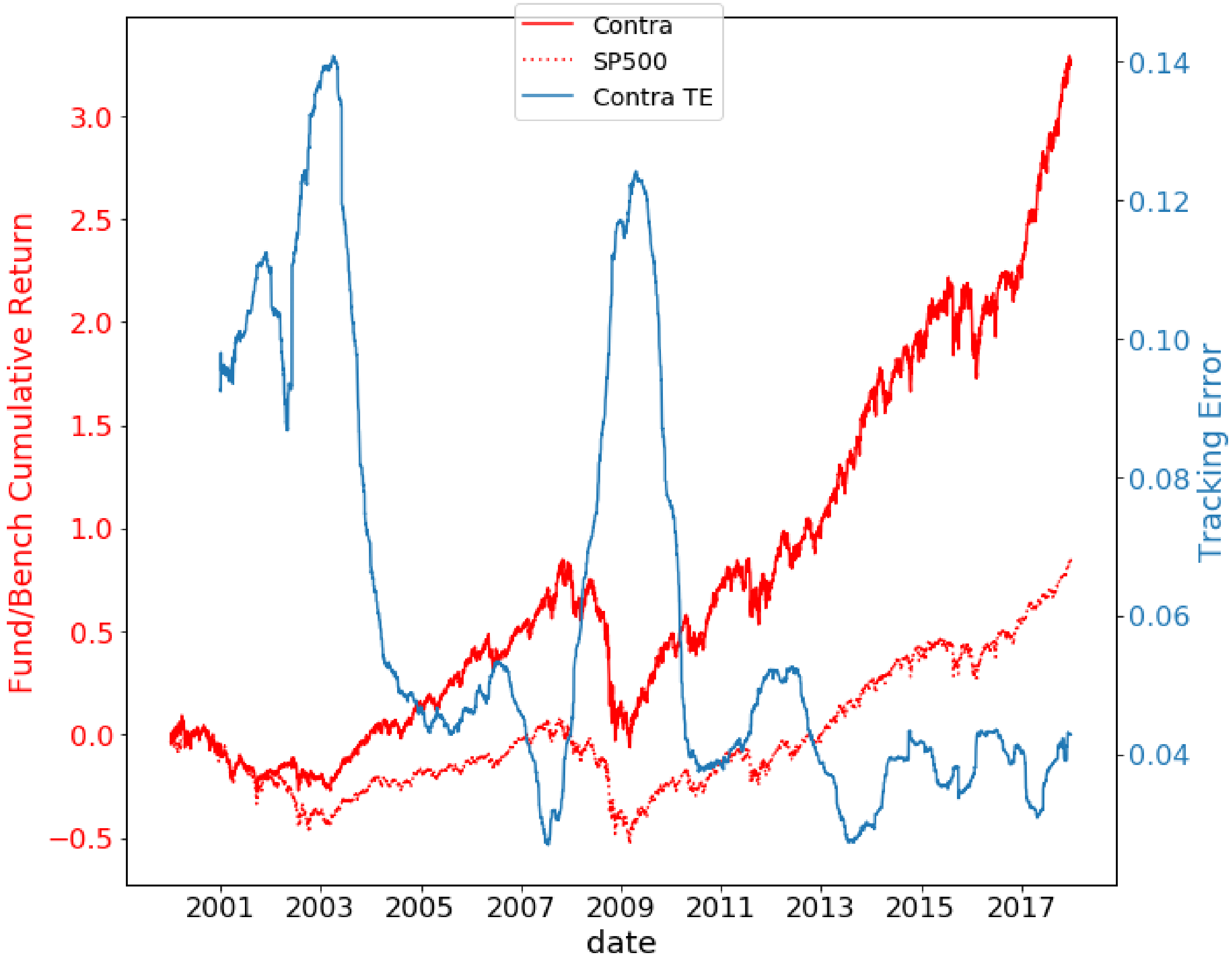
- Tracking error is a measure of active risk
- It measures the magnitude of deviation of a fund's returns vs. its benchmark:

$$Tracking\ Error = std(\ r_{fund} - r_{benchmark}) = \sqrt{\frac{1}{N-1}\sum_{t=1}^{N}\left(R_{t,fund} - r_{t,benchmark}\right)^2}$$

- The higher the tracking error, the more 'active' a fund is vs. its benchmark
- A pure index fund will have a tracking error of close to zero as it completely replicates the price behavior of the benchmark

Tracking Error Example – Fidelity Contra Fund

- Tracking error is the standard deviation of relative (active) returns
- TE high during down market, normally at 4%
- See L5_TrackingError.py for details



Value-at-Risk (VaR)

- Value-at-Risk, or VaR(p), is the loss on a portfolio/position such that there is a probability p of losses equaling or exceeding this magnitude in a given trading period specified
- p is also called confidence level
- Or in other words, there is a (1-p) probability of loses being less than VaR(p)
- If $Q = P_t - P_{t-1}$ is the P&L for the portfolio during the specified period, where P_t is the portfolio value at t, then we have

$$\Pr[(Q \leq -VaR(p)] = p$$

- If f_q is the P/L density function then we have VaR(p) is the magnitude of loss such that

$$p = \int_{-\infty}^{-VaR(p)} f_q\,(x)dx$$

Value-at-Risk (VaR)

- VaR was developed by JPMorgan to measure risk of trading desks and loan books through the RiskMetrics software system
- VaR is very commonly used by investment banks and hedge funds to measure absolute risk levels of their business and portfolios
- One of the most important risk measures to understand and know how to calculate/apply for quantitative finance professionals
- Important properties of risk metrics: magnitude + probability + horizon!

Value-at-Risk (VaR) Examples

- The prop trading desk's derivative book has a daily VaR(5%) of $10M. This means the prop desk is expected to lose $10M or more once every 20 days
- Hedge fund XYZ's holdings have a value of $1B and a daily Var(1%) = $15M. This means the hedge fund is expected to lose $15M or more once every 100 days
- Three things to specify
  - Confidence level
  - Holding period length
  - VaR value
- Instead of measured in dollar amount, VaR can also be measured in percent returns. E.g., Bank A's trading desk has a monthly VaR (at 5% confidence level) of -5%

How to calculate VaR? – A simple example

- Estimate monthly VaR at 5% confidence level
- Monthly Return Data: Total of 40 numbers
- See Python code L5_pdfcdf.py

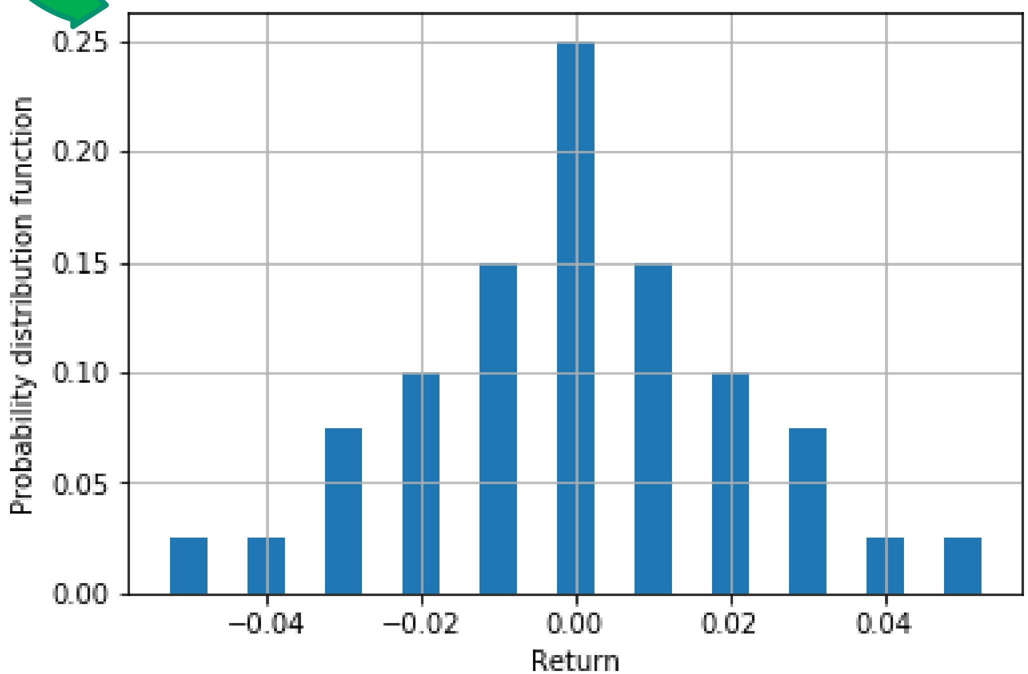
```
ReturnSample = np.array([-0.05,
                         -0.04,
                         -0.03, -0.03, -0.03,
                         -0.02, -0.02,-0.02, -0.02,
                         -0.01, -0.01, -0.01, -0.01, -0.01, -0.01,
                         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
                         0.01, 0.01, 0.01, 0.01, 0.01,0.01,
                         0.02, 0.02,0.02, 0.02,
                         0.03, 0.03,0.03,
                         0.04,
                         0.05
                         ])
```

Calculate VaR Step 1: Develop the pdf

```
ReturnSample = np.array([-0.05,
                         -0.04,
                         -0.03, -0.03, -0.03,
                         -0.02, -0.02,-0.02, -0.02,
                         -0.01, -0.01, -0.01, -0.01, -0.01, -0.01,
                         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
                         0.01, 0.01, 0.01, 0.01, 0.01,0.01,
                         0.02, 0.02,0.02, 0.02,
                         0.03, 0.03,0.03,
                         0.04,
                         0.05
                         ])
```

Step 1: Develop probability density function (pdf)

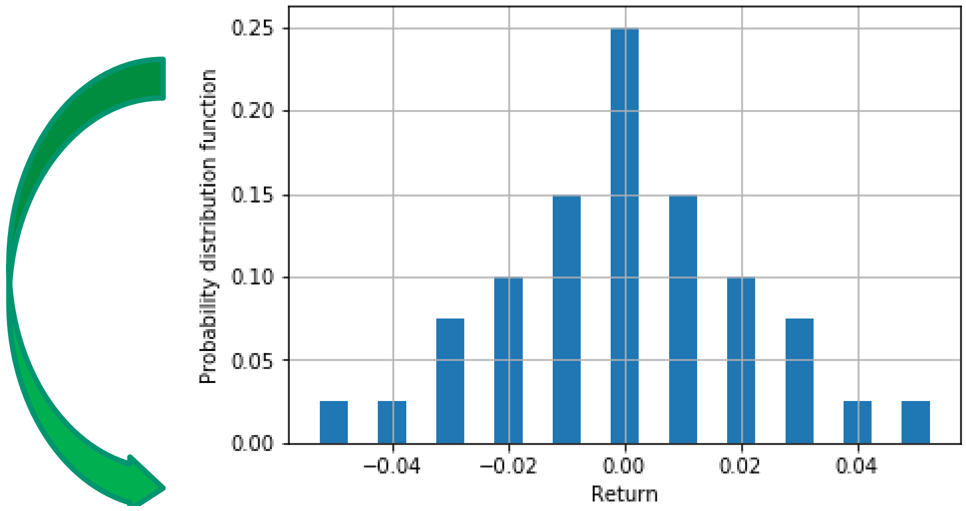
- Probability of each return: 1/40=2.5%
- -0.05 return appeared once, has a pdf of 1/40=2.5%
- 0.0 return appeared 10 times, has a pdf of 10/40 = 25%

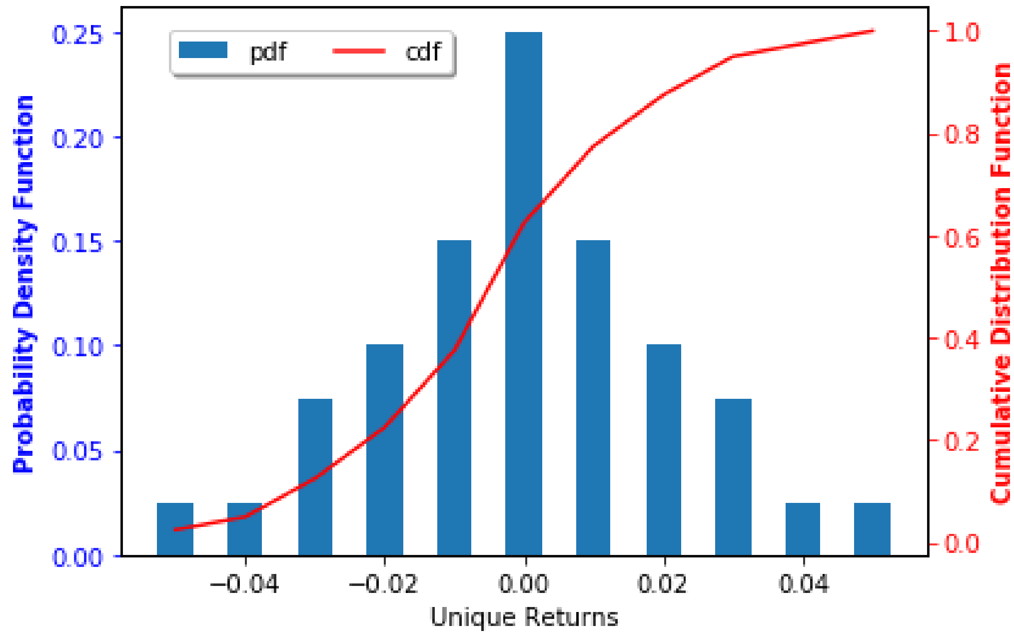


Calculate VaR Step 2: Convert pdf into cdf

Step 2: Convert pdf into cdf

- cdf (x) = probability (X≤ x)
- $cdf(x) = \int_{-\infty}^{x} pdf(t)dt$

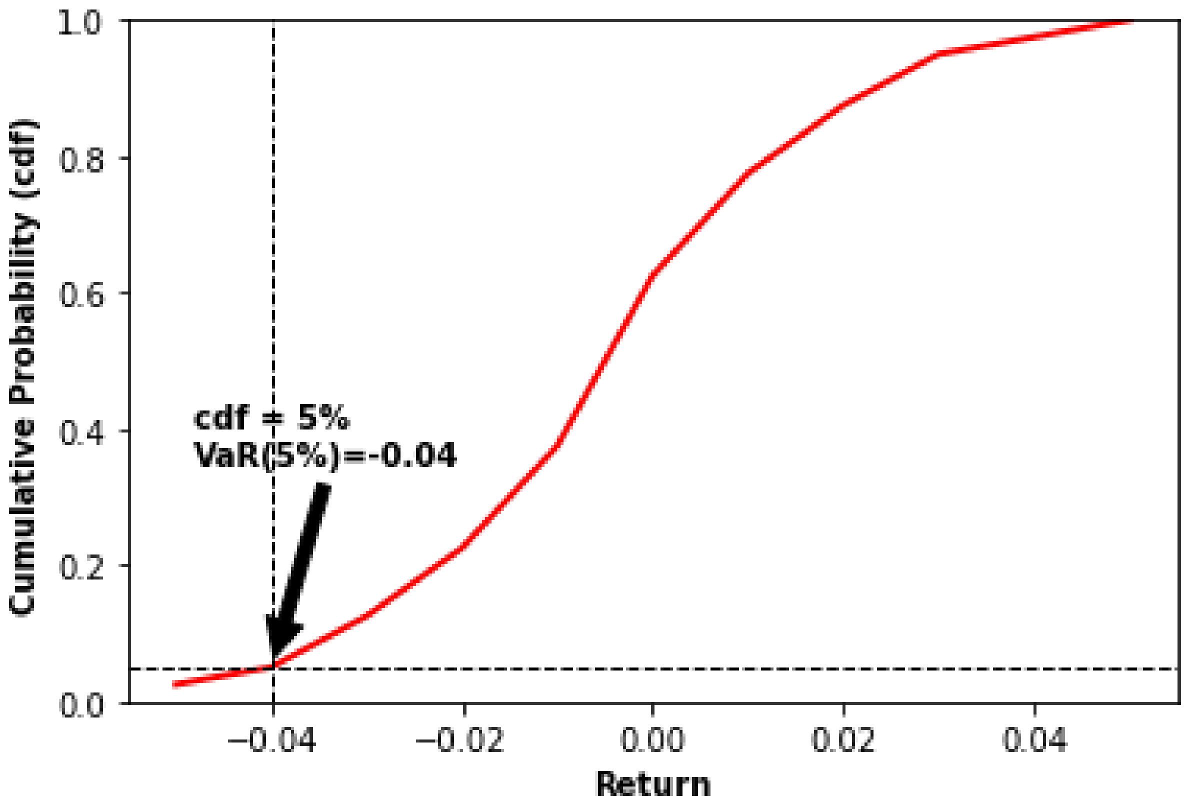




Calculate VaR Step 3: Find VaR on the cdf curve

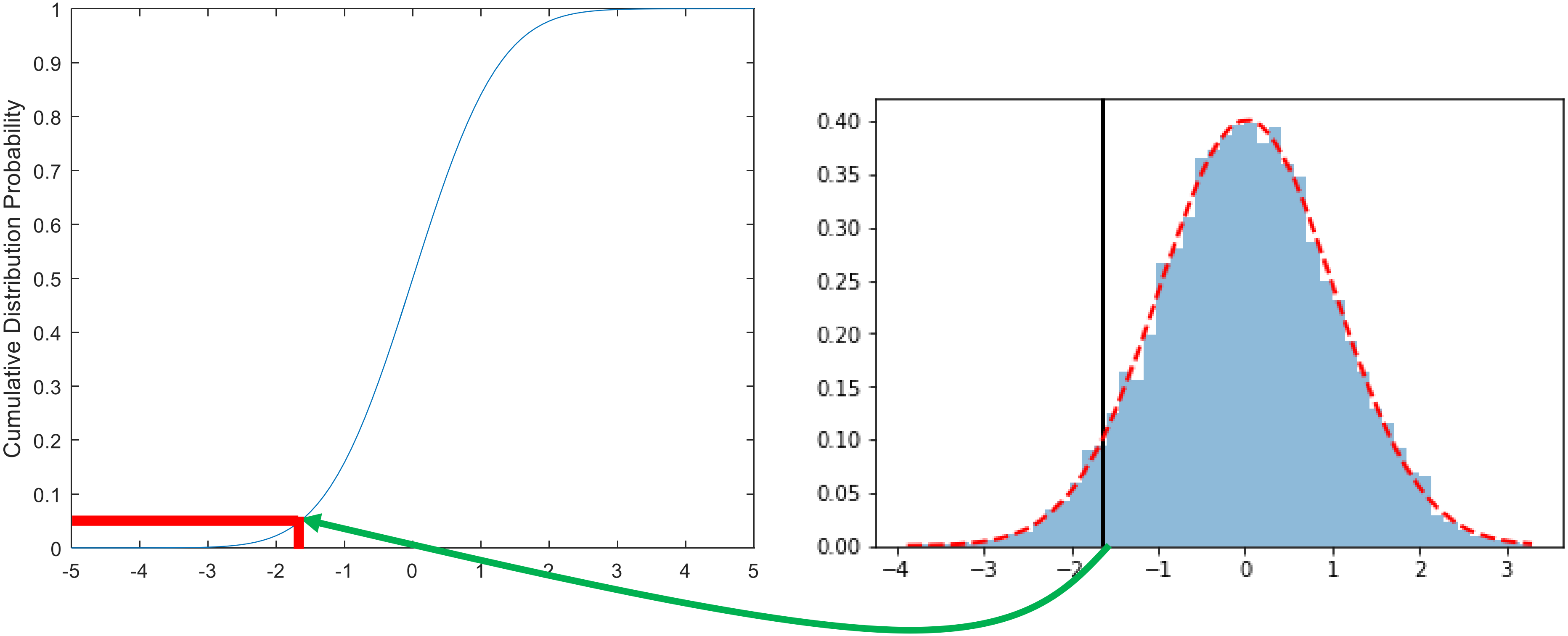
Step 3: Identify VaR from the cdf curve

- VaR is the return (x axis) on the cdf curve where cdf = VaR (y axis) confidence level
- The cdf curve establish a relationship between a return and cumulative probability for returns lower than that particular threshold
- In the chart below, the probability of return being equal or less than -0.04 is 5%
- <u>Inverse cdf function</u>: given the cdf (e.g. 5%) what is corresponding return/portfolio value?



Relationship Between PDF and CDF

Cumulative distribution function (pdf) / Probability density function (pdf)



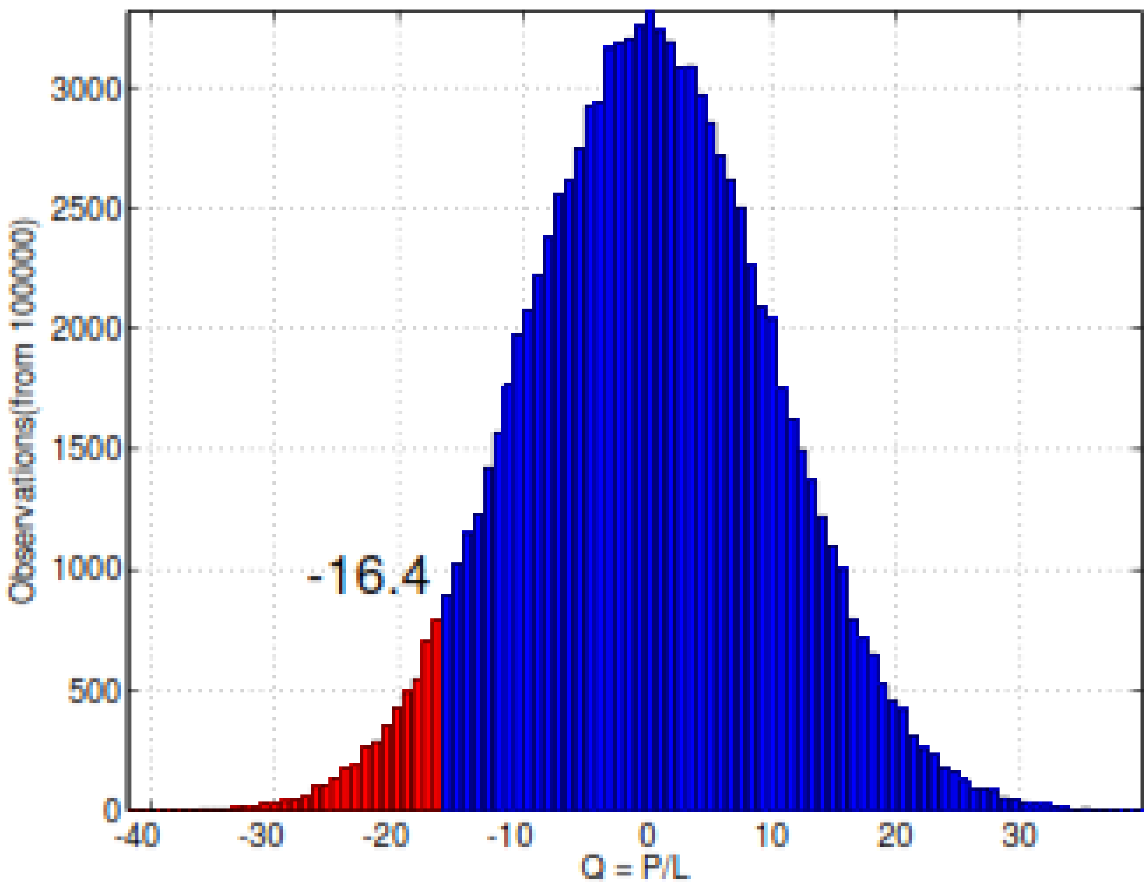
When X Var=-1.65 the cumulative distribution function = 0.05

Area under the red line from –infinity to -1.65= 0.05 = confidence level for VaR

VaR Calculation Methods

- Analytical: Assume return/risk factors are normally distributed
  - Advantages: Simple and fast
  - Disadvantages: linear assumption on value, normality assumptions on drivers
- Historical: Use historical data as the sample
  - Advantages: Simple and fast, no need to make distribution assumption, captures tails naturally(if data is rich)
  - Disadvantages: Data may be limited
- Monte-Carlo: Random Numbers based on distribution assumption
  - Advantages: Can generate a lots of data, easy to calculate confidence bands and standard errors
  - Disadvantages: Depends on distribution assumptions

Analytical Random Normal: VaR(5%), Q is N(μ=0, σ=10)



Normal distribution： X variable with 5% cdf= μ -1.65* σ

Normal Return VaR

- $R_{t+1} = \frac{P_{t+1}}{P_t} - 1$, returns for time t to t+1 (Ignoring dividends here)
- $profit\ and\ loss\ Q_{t+1} = P_{t+1} - P_t = P_t(1 + R_{t+1}) - P_t = P_t R_{t+1}$
- Two ways to think about VaR
  - In dollar amount, as how much money you will lose
  - Or in percentage returns, as how many percent the value of your portfolio will go down
- 1st way: If $F_P\ is\ cdf\ for\ P_{t+1}$, we have $R^* = F_P^{-1}(p)$, where F_P^{-1} is inverse cdf. Then VaR can be calculated as

$$VaR(p) = R^* - P_t$$

  - In Python, the function to do this is scipy.stats.norm.ppf imported from the scipy package
- 2nd way: If $F_R\ is\ cdf\ for\ R_{t+1}$, we have $R^* = F_R^{-1}(p)$, where F_R^{-1} is inverse cdf. Then VaR can be calculated as

$$VaR(p) = -PR^*$$

VaR Example 1 – Analytical Approach

1. Simple Python code to do this, given normal distribution (this is the analytical approach we talked about previously)
2. Filename: L5_VaR.py

```
############### Analytical VAR example#################
PortValue_Current = 100
mu=100
sigma = 10
p=0.05

#  ----- Analytical Approach to calculate VaR ------
PortValue_At_p = stats.norm.ppf(p,mu,sigma) # Value at Risk following the equation
Loss_At_p_Analytical = PortValue_At_p - PortValue_Current
```

VaR Example 1 – Monte-Carlo Approach

- Simple Python code to do this, using Monte-Carlo Simulations
- Filename: L5_VaR.py
- Similar to what we did in our first example at the beginning:
  1. Generate random number for portfolio value or returns based on distribution assumption (e.g., normal or student t)
  2. Sort return (or portfolio value) data on ascending order
  3. Assuming each data point is associated with the probability of 1/N, where N is the total number of data points
  4. The cutoff point associated with the confidence level is identified, which also identifies the VaR

```python
#  ------ Monte-Carlo Approach ------
N = 10000
# generate 10000 normally distributed random numbers
PortValue_MC = np.random.normal(mu, sigma, N)
# or stats.norm.rvs(mu, sigma, N)
PortValue_MC_Sorted = np.sort(PortValue_MC)
ID_At_p = round(p*N)-1 # matlab start from 1, python start from 0
PortValue_At_p_MC = PortValue_MC_Sorted[ID_At_p]
# First way
Loss_At_p_MC = PortValue_At_p_MC - PortValue_Current
```

Historical VaR

- Historical VaR: The approach uses quantiles from historical data for VaR calculations (Remember the first example?)
- No need to make assumptions about distribution
- Naturally reflect the “fat tails”– to the extent that the data set is rich enough
- The key to implementation:
  - Find the quantiles of the data set (for returns, portfolio values)
  - Can be done through sorting

Historical VaR Python Example

- Methods used:
  - Analytical (use estimated mean and standard deviation)
  - Historical
- Comparison of results from these two different methods
  - VaR(5%) historical is less than Normal
  - VaR(1%) historical is greater than Normal
  - Greater extreme tail risk

```
5% VaR Historical data   1.4469
1% VaR Historical data   2.8286
5% VaR Normal   1.5971
1% VaR Normal   2.2741
```

```
#
#---------Historical VaR Method --------------------
# load historical data
mat = spio.loadmat('retUS.mat',
                   squeeze_me=True)
# ret are daily returns with dividends
ret1 = mat['ret'] # indicate one day return (just a name)

# vector of portfolio value for 100 held for 1 day
#    all possible outcomes from the sample
port1day = 100*(ret1+1)
# find historical var using direct percentile on portfolio values
#    report the oposite of the loss
var05Hist = 100-np.percentile(port1day,5)
print("5% VaR Historical data {0:8.4f}".format(var05Hist) )

# find historical var using direct percentile on portfolio values
var01Hist = 100-np.percentile(port1day,1)
print("1% VaR Historical data {0:8.4f}".format(var01Hist) )

```

P vs (1-p)

- VaR(p) can be reported/quoted two ways:
  1. There is a probability of p that the portfolio loss will be at least VaR(p), or
  2. We are confident with 1-p confidence level that the portfolio loss will not exceed VaR(p)

Sign of VaR

- We have been talking about VaR using positive numbers in the context of "the probability the loss on the portfolio is at least $10M is 5%. Note $10M here is positive
- This is ok because we added 'loss' before it. It is not critical but you should be aware of it
- Theoretically, it is possible a portfolio may never lose value, in this case it is not a loss we are talking about. But this rarely happens in financial market

Caution: Consistency about data frequency

- If VaR and ES calculation is for a future time period of 1Day/1M/1Yr, make sure
  - In applying the analytical method, the mean, standard deviation calculation is calculated for the same time duration, i.e., 1Day/1M/1Y mean and standard deviation
  - In applying historical and Monte Carlo methods, the return data used is converted to cover the same time duration
- It is generally the best practice to use data of the same frequency, i.e., weekly data for weekly VaR/ES
- What about calculating VaR, ES for 12M in the future?
  - using weekly and monthly data? Mean return will be the same but Std of return will be different! No right answer!
  - Try rolling return calculations (e.g., Mth 1:12 gives first annual data, 2:13 gives second annual data)

Monte-Carlo VaR vs. Historical VaR

- Historical VaR: The approach uses quantiles from historical data for VaR calculations
- Monte-Carlo VaR:
  – Decide to use certain statistical distributions to represent the returns/values of the portfolio
  – Estimation of the necessary parameters needed to define the distributions (most of the time this is done using historical data)
- Very often Monte-Carlo VaR and Historical VaR are closely linked through the usage of the same set of historical data
- Monte-Carlo approach is sometimes redundant because Historical VaR can be used instead

VaR Assuming Student-t Distributed Returns

- Instead of normal distribution to calculate analytical or Monte Carlo VaR, we can also use other distribution assumptions such as student-t
- P_t=100
- $R_{t+1} \sim t(4), \mu = 0.12, \sigma^2 = 0.2^2, v = 4$
- Return corresponding to VaR(5%):

$$R^* = stats.t.ppf(p, v, \mu, \sigma) = -0.181,$$

$$\text{e.g., scipy.stats.t.ppf}(5\%, 4, 0.12, 0.2)$$

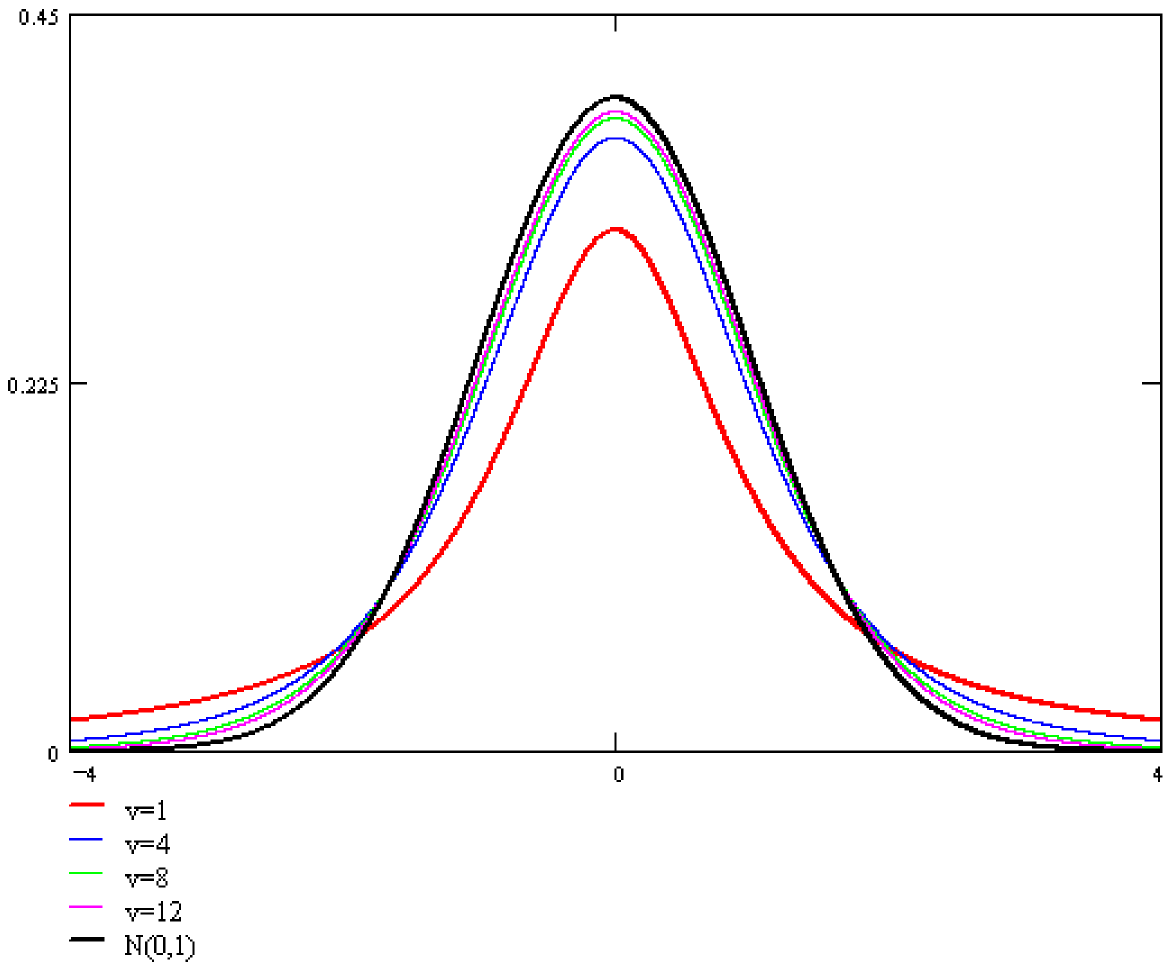
$$VaR(p) = -P_t R^*,$$

$$e.g., -100 * \text{scipy.stats.t.ppf}(5\%, 4, 0.12, 0.2) = 20.9$$

- scipy.stats.t.ppf(), is the function for inverse student-t cdf

Student-t Distributed Returns

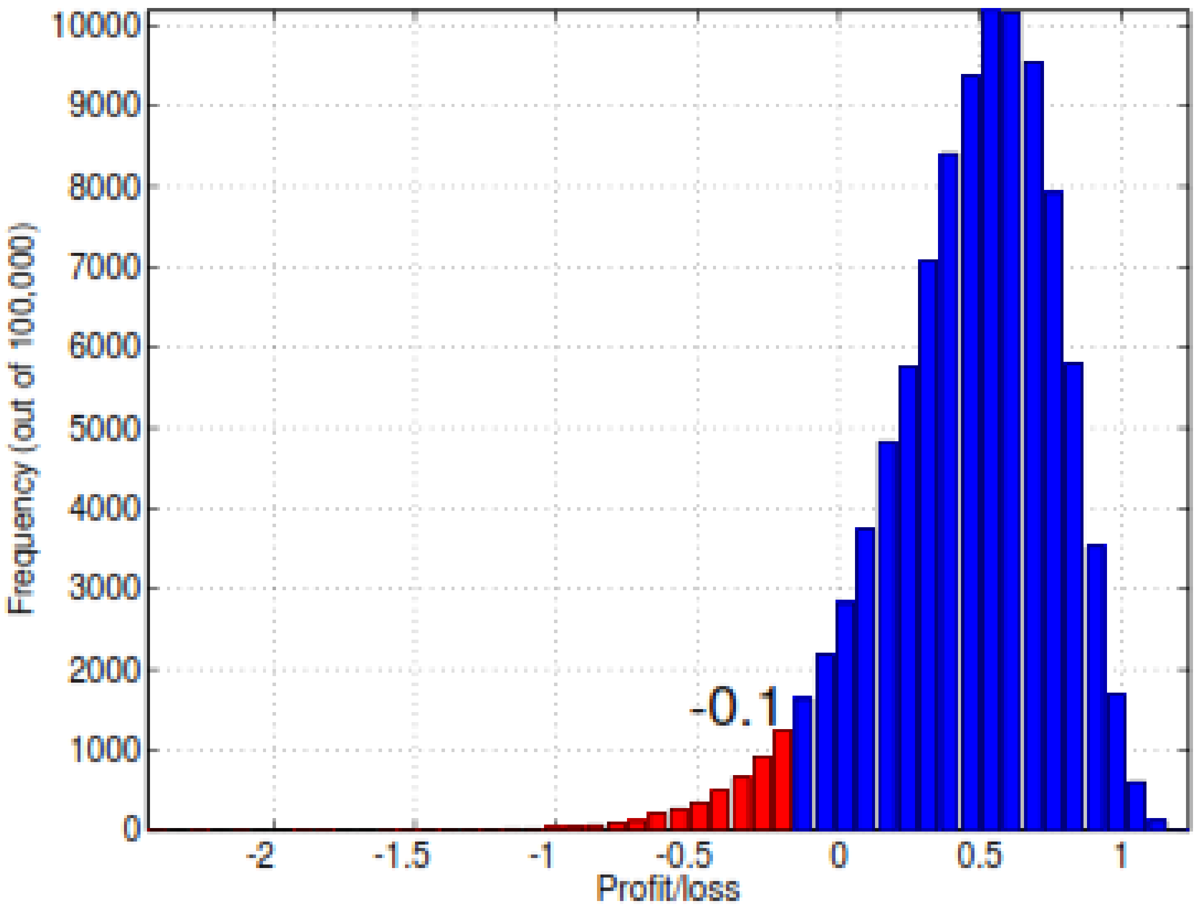
Student-t distribution: the lower the degree of freedom, the fatter the tail



Issues for VaR: It Doesn't Tell the Whole Story

- VaR is just a quantile marking a threshold point on the distribution
- VaR does not tell us anything about left tail beyond it
- Might ignore a lot of risk if the tail is fat or irregular
- What is the right probability level to use? 5%, 1%. It is not always the smaller the probability the better

VaR Ignores the Left Tail



- Doesn't tell us the magnitudes of loss and associated probabilities left of the cutoff point associated with VaR

Solution: Expected Shortfall

- CVaR (Expected shortfall): Mean loss given that VaR loss is exceeded – the mean of returns that that less than VaR, or the weighted average of areas under the cyan line