Computer Simulations and Risk Assessment – Lecture 4

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

Lecture 4 – Stylized Facts About Financial Data

Stylized Facts of Financial Data

- Autocorrelations
- Fat tails
- Volatility persistence/clusters
- Nonlinear/Extreme dependence

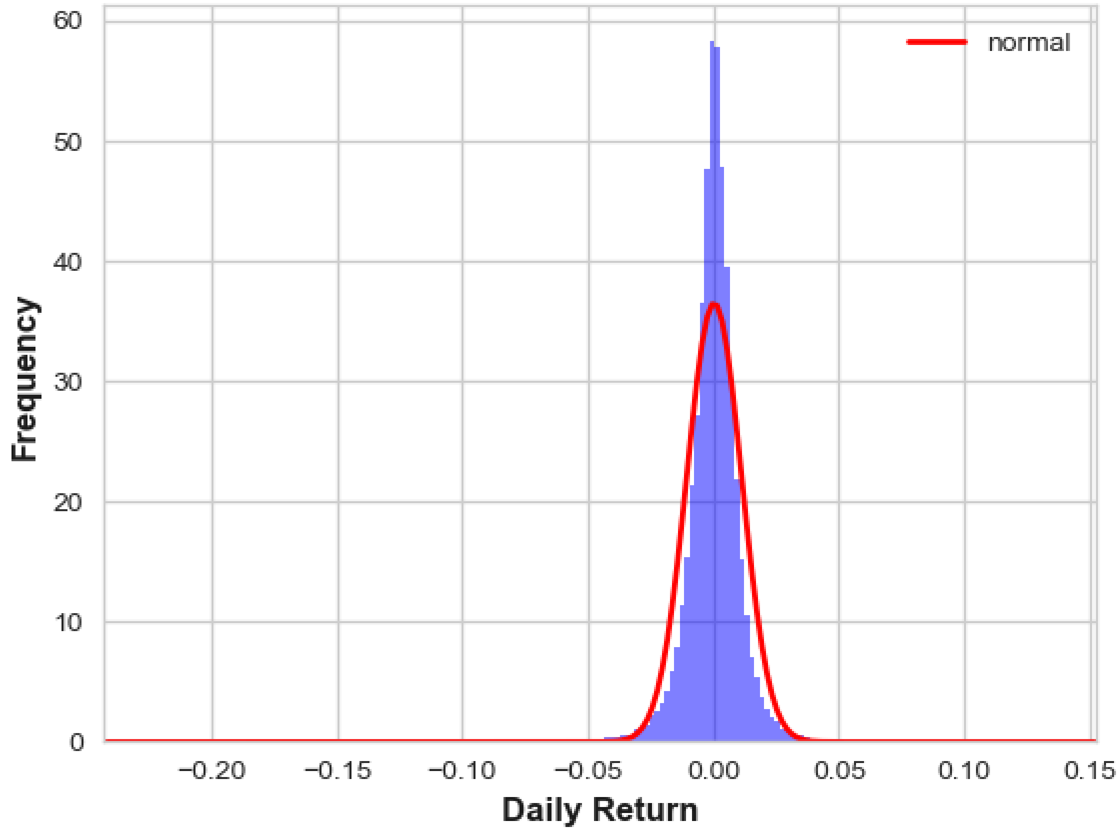
Introduction

- What is risk management about?
  - What is the distribution of my future profit/loss
  - E.g., there is a 5% probability my portfolio will lose 10% of more in value in the coming 12 month
  - It is about predicting what happens in the future, not the past
- How do we achieve that?
  - Make assumptions of assets/factors return distribution in the future
  - Combine at the portfolio level with consideration of correlation
- Today's class provide
  - A quick overview of statistical properties of returns and volatilities of returns
  - This review provides a theoretical foundation for some of the models we will cover the rest of the classes

Assumptions of asset returns/correlation

- How do we go about making assumptions about future asset returns/correlations?
  - Study the history
  - Unconditional/Conditional distribution/Correlation
  - Study current economic situation and predict what happens next
    - Are we going to experience similar environment like 2008/2009 in the coming two years?
    - Are we going to experience something we have never seen before?
    - Apply the corresponding forward-looking distribution/correlation

Dow Daily Return Density and Normal Fit



- Normal distribution not a good fit

QQ (Quantile-Quantile) Plot

- QQ plot is a diagnostic test of distributions
- It plots sample quantiles (historical data quantiles) versus corresponding ones from assumed theoretical distributions, e.g., normal.
- A quantile is the fraction of data below the given value
- If the points in the q-q plot depart meaningfully from a straight line, then the sample is not likely to follow the theoretical distribution being compared with
- Python function: stats.probplot()

Stats – Quantile

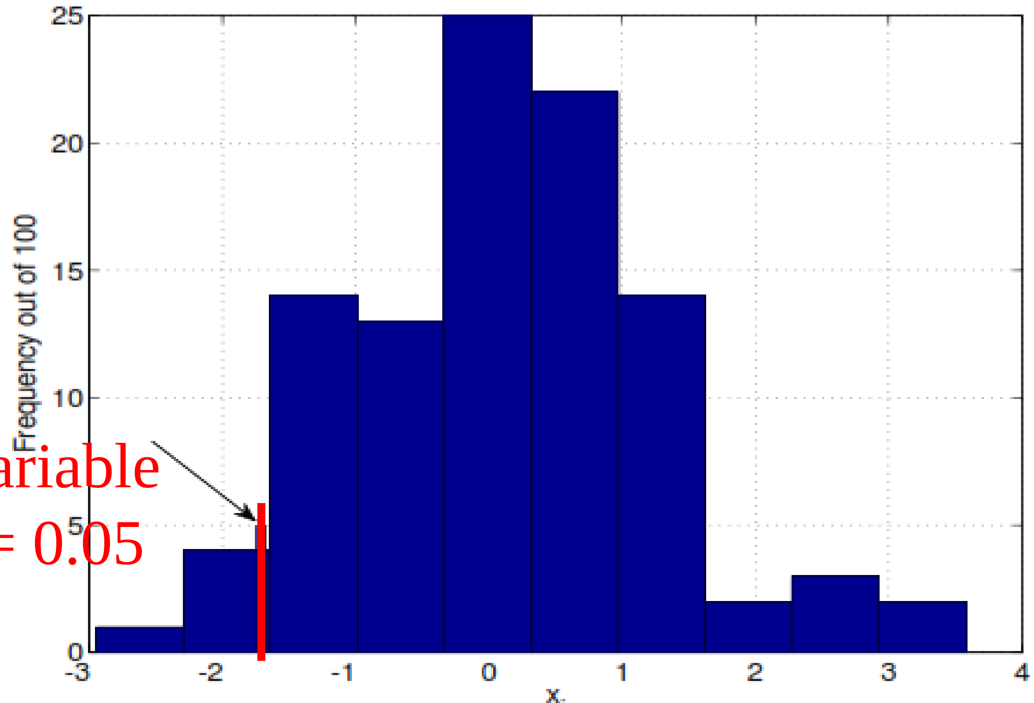
- Quantile α

$$q_{\alpha}: \Pr(X<q_{\alpha})= \alpha$$

e.g., $q_{0.5}$=median of sample

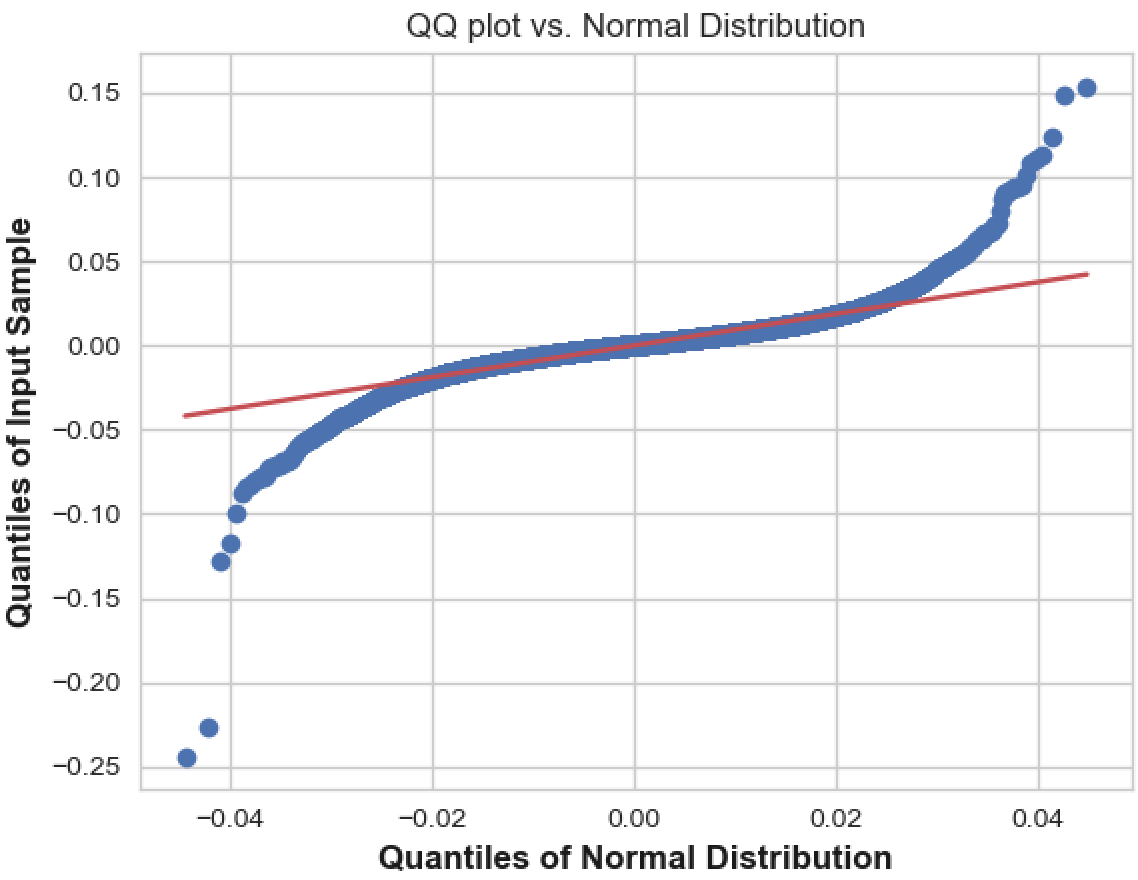
for standard normal $q_{0.5}$=0, $q_{0.95}$=-1.64, or 50th and 95th percentile, is 0 and 1.64 standard deviations from the mean

- Quantile 0.05, $q_{0.05}$ for historical data:



$q_{0.05}$ is the x variable that make cdf = 0.05

QQ Plot of Daily Returns with Normal Distribution



- Is the Dow return normally distributed per the QQ plot?

QQ Plot – how can we produce the curve?

See sample code: L4_StylizedFinExcelData.py

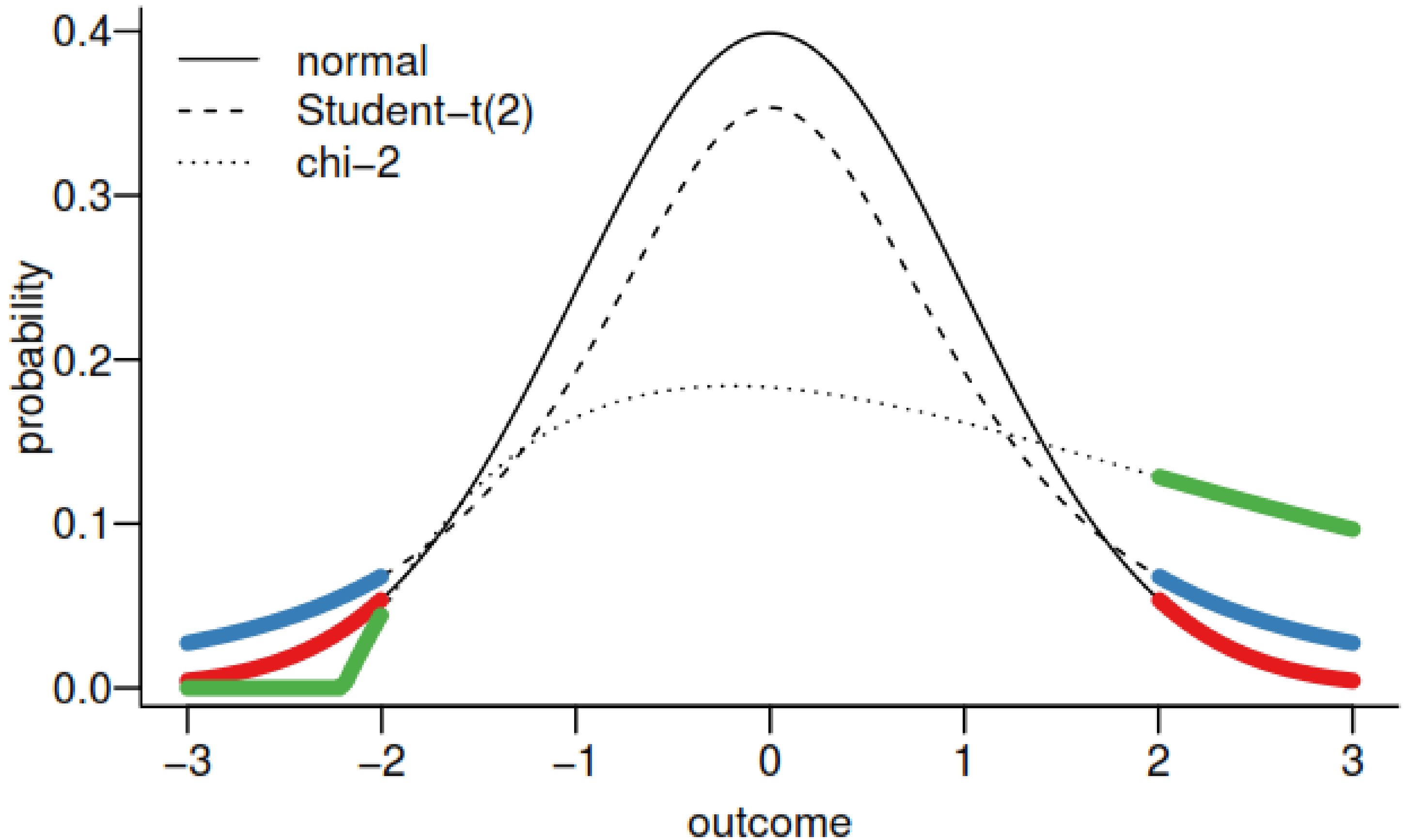
1. Sort the data: *Ret_Dow_sorted = np.sort(Ret_Dow)*
2. Take the minimum sorted returns: *Ret_Dow_min = Ret_Dow_sorted[0]*
3. Calculate the empirical cumulative distribution function (cdf) for this return data point: *cdf_Ret_Dow_min = 1/T*
4. To have the same cdf from step 3, if data is normally distributed, what would be the return be? *normZscore = norm.ppf(cdf_Ret_Dow_min,loc=mu, scale=sigma)*
5. Print out the results: *print('x,y value of the first data point on the QQ plot is',normZscore, Ret_Dow_min)*

Comparing tails: Dow vs. Normal

- The table below compares the probabilities of Dow generating certain negative daily returns, relative to those from a normal distribution with the same mean and standard deviation

	Probability	
Returns Below this Level	Dow Realized Return	Normal Distribution with same mean and std
-1%	**0.113**	**0.1742**
-2%	**0.0301**	**0.03196**
-3%	**0.0109**	**0.00282**
-5%	**0.00233**	**0.0000021**

Tails of Common Distributions



Jarque-Bera (JB) test for Normality

- JB test for normality – skewness and kurtosis of the returns follow a chi square distribution
$$\frac{T}{6}Skewness^2 + \frac{T}{24}(Kurtosis - 3)^2 \sim \chi^2(2)$$
- If p-value is low, then reject null hypothesis of the distribution is normally distributed
- Python code: *h, p = stats.jarque_bera(Ret_Dow)*
  - If p<5%, reject null hypothesis or return is NOT normally distributed
  - E.g., p=0.001 in this case, so Dow return is NOT normally distributed

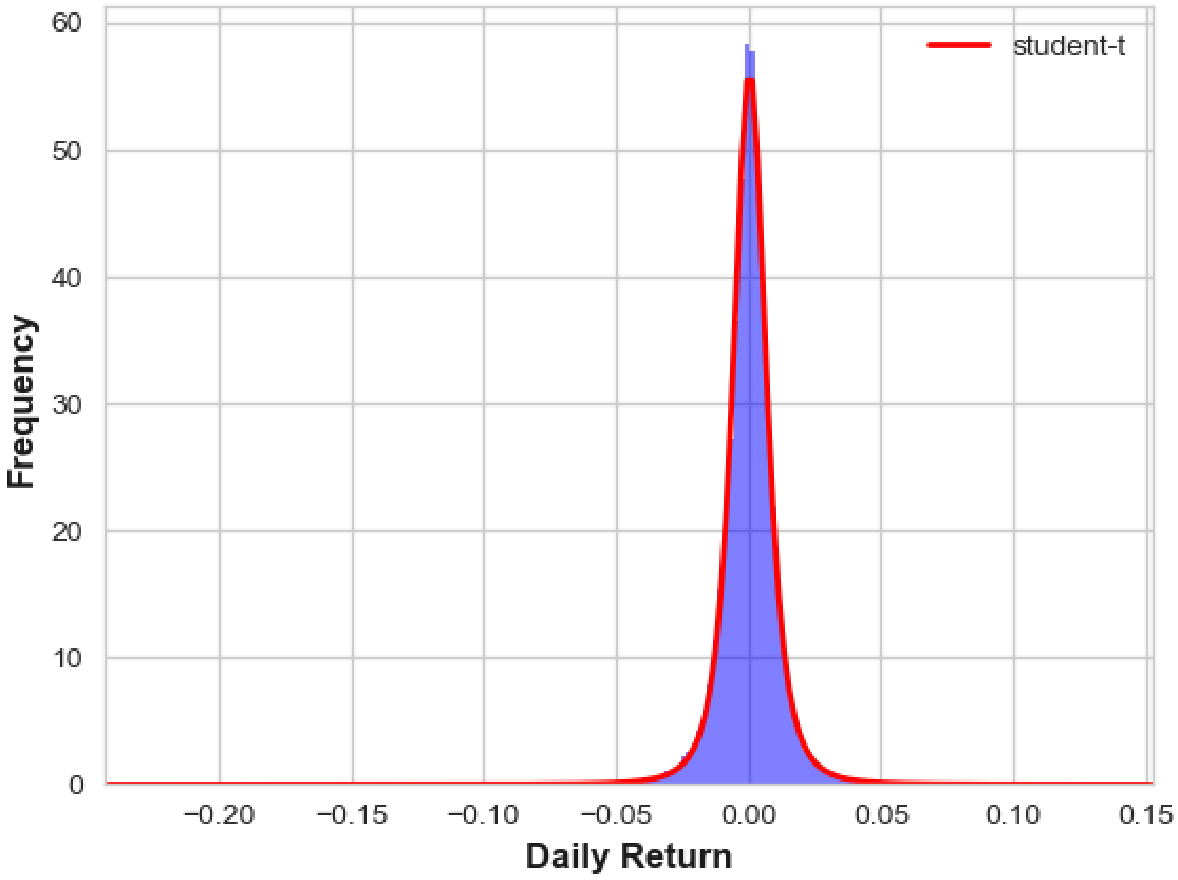
Some Memorable Tail Risk Quotes

- "as you well know, the biggest problems we now have with the whole evolution of risk is the fat-tail problem, which is really creating very large conceptual difficulties. Because as we all know, the assumption of normality enables us to drop off the huge amount of complexity in our equations.." Alan Greenspan (1997)

Some Memorable Tail Risk Quotes

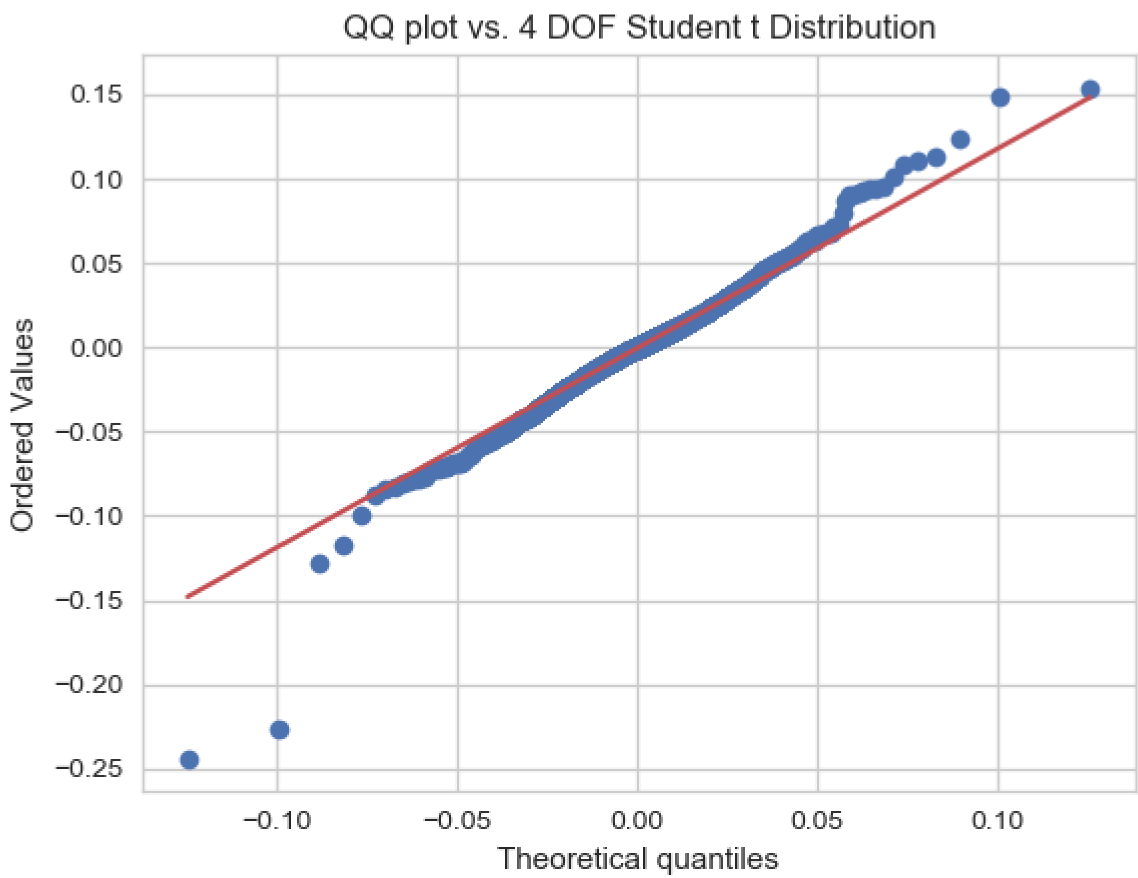
- "On 13 August 2007, The Financial Times reported David Viniar (former CFO of Goldman Sachs)'s explanation of why two large hedge funds managed by Goldman Sachs had both lost over a quarter of their value in a week, requiring the injection of $3 billion to support them. Viniar ascribed the events to a series of exceptional events: "We were seeing things that were 25 standard deviation moves, several days in a row".This has since been used to illustrate the problems of inappropriate mathematical models in finance, especially those based on the assumption of **Normality**
- **25 Sigma under normal distribution has a probability of 3×10^{-138}. The age of the universe is estimated to be 5×10^{12} days only**

Dow Daily Return Density and Student-t Fit



- Student-t distribution a much better fit

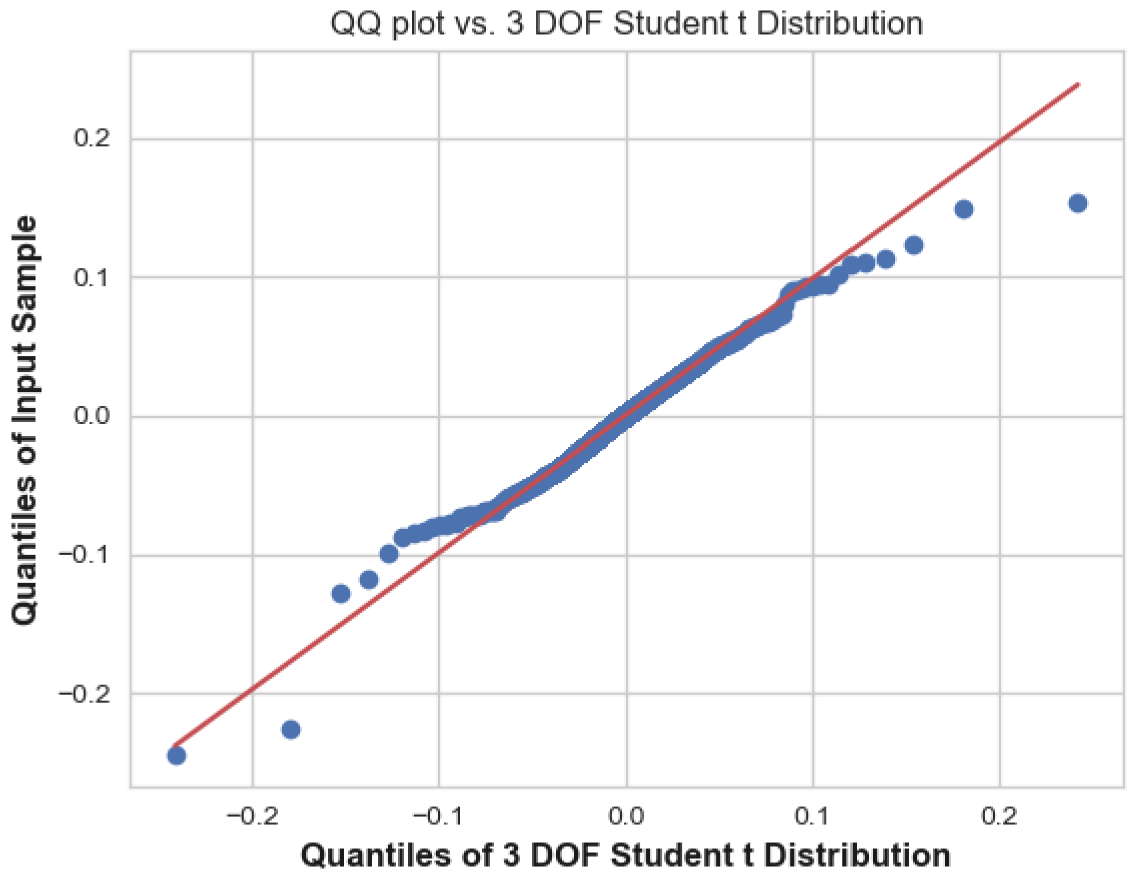
QQ plot with 4 degree of freedom Student-t



- Student-t distribution with 4 dof a better fit

Brandeis

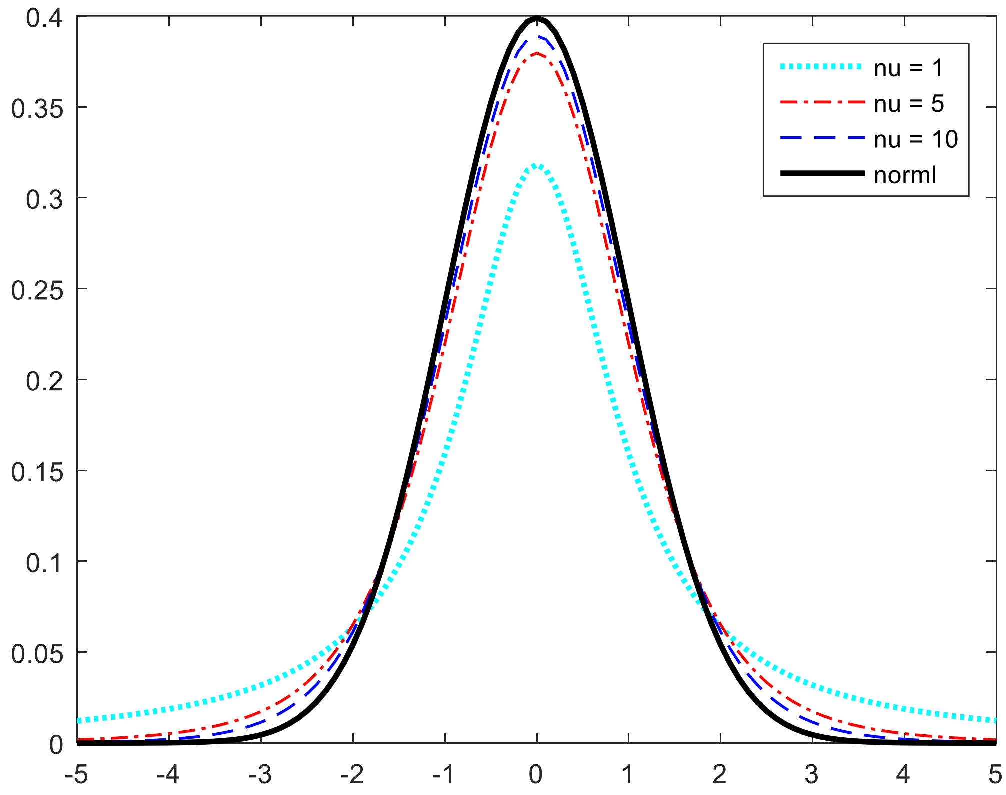
QQ plot with 3 degree of freedom Student-t



- Student-t distribution with 3 dof a much better fit (fatter tail)

Student t Distribution

- Produce fatter tail relative to normal distribution
- Decided by only one parameter: ν. When $\nu \rightarrow \infty$, approach normal
- Mean=0, for $\nu>1$, otherwise undefined



Random Number Generators in Python

- scipy.stats.norm.rvs(loc=0, scale=1, size=1) – generate Multivariate normal random numbers
- scipy.stats.norm.pdf(x, loc=0, scale=1) – generate the probability distribution function for a normal distribution
- scipy.stats.t.rvs(df, loc=0, scale=1, size=1) – generate Multivariate t random numbers, with degree of freedom df, mean of loc=0, and standard deviation of scale =1
- scipy.stats.t.pdf(x, df, loc=0, scale=1) – generate Multivariate t random numbers, with degree of freedom df, mean of loc=0, and standard deviation of scale =1

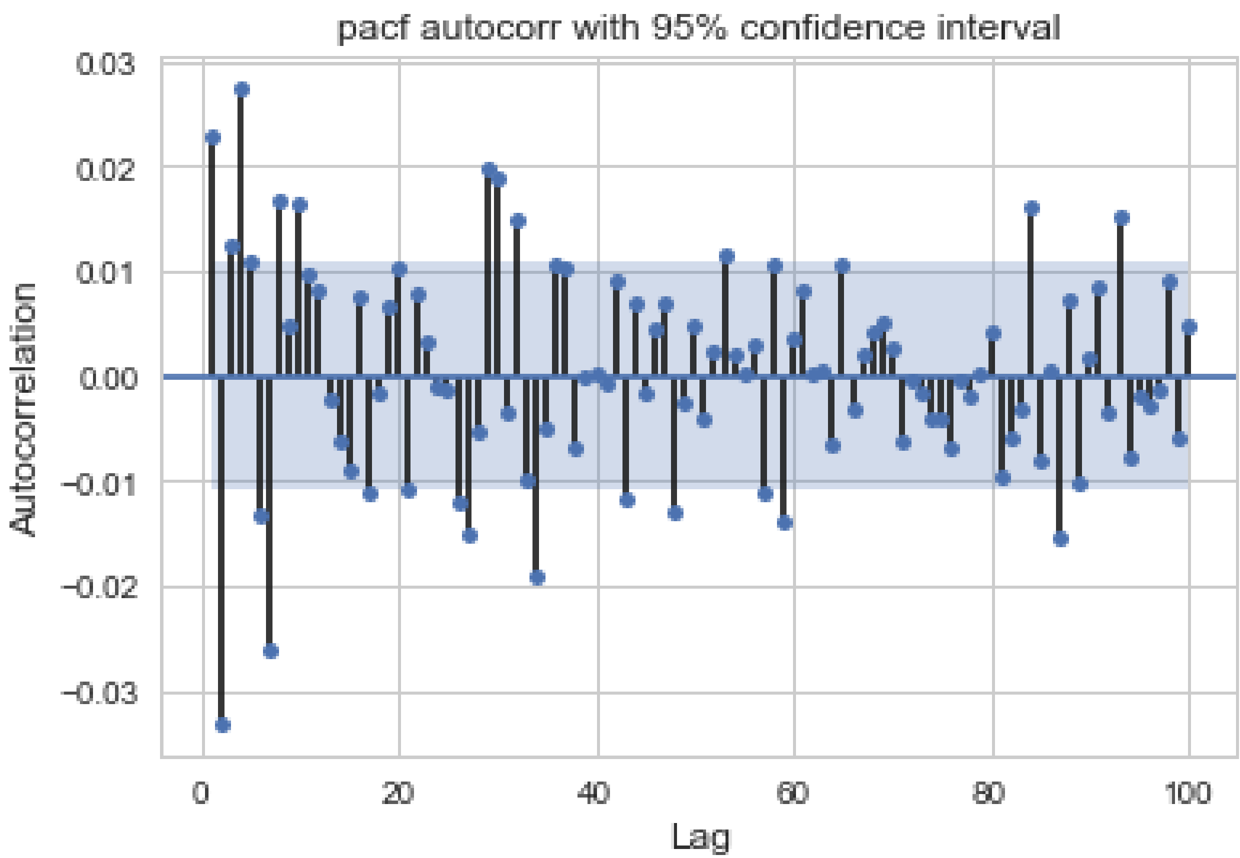
Autocorrelation

- Autocorrelations measure how returns from one period are correlated with returns from previous periods (as oppose to cross-security correlation)
- If autocorrelations are statistically significant, there is evidence for predictability
- The coefficients of an autocorrelation function (ACF) give the correlation between returns and its lags

$$\beta_i = Corr(x_t, x_{t-i})$$

- Significance of autocorrelation coefficients can be tested by using the Ljung-Box (LB) test or the Engle LM test

Dow Daily Return Autocorrelations



- Dow Jones Industrials daily return
- Autocorrelation of returns very low and statistically marginally significant

Statistical Test of Significance of Autocorrelation

- Ljung-Box (LB) test:
  - the hypothesis: data in the time series are independent of each other, or the autocorrelation of the time series is not different from zero
  - Applies to time series that can be assumed homoscedastic (**homogeneity of variance)**
  - Python function: acorr_ljungbox()
- Engle LM test:
  - Applies to time series that can be assumed heteroskedastic
  - More appropriate for most of financial return time series
  - Python function: het_arch() (Note this is test of residuals, not returns

Statistical Test of Significance of Autocorrelation

- Python function: het_arch() (Note this is test of residuals, not returns

1. Estimate the best fitting autoregressive model AR(q) $y_t = a_0 + a_1 y_{t-1} + \cdots + a_q y_{t-q} + \epsilon_t = a_0 + \sum_{i=1} a_i y_{t-i} + \epsilon_t$.

2. Obtain the squares of the error $\hat{\epsilon}^2$ and regress them on a constant and q lagged values:

$$\hat{\epsilon}_t^2 = \hat{\alpha}_0 + \sum_{i=1}^{q} \hat{\alpha}_i \hat{\epsilon}_{t-i}^2$$

where q is the length of ARCH lags.

3. The null hypothesis is that, in the absence of ARCH components, we have $\alpha_i = 0$ for all $i = 1, \cdots, q$. The alternative hypothesis is that, in the presence of ARCH components, at least one of the estimated α_i coefficients must be significant. In a sample of T residuals under the null hypothesis of no ARCH errors,

- Matlab test results for S&P500

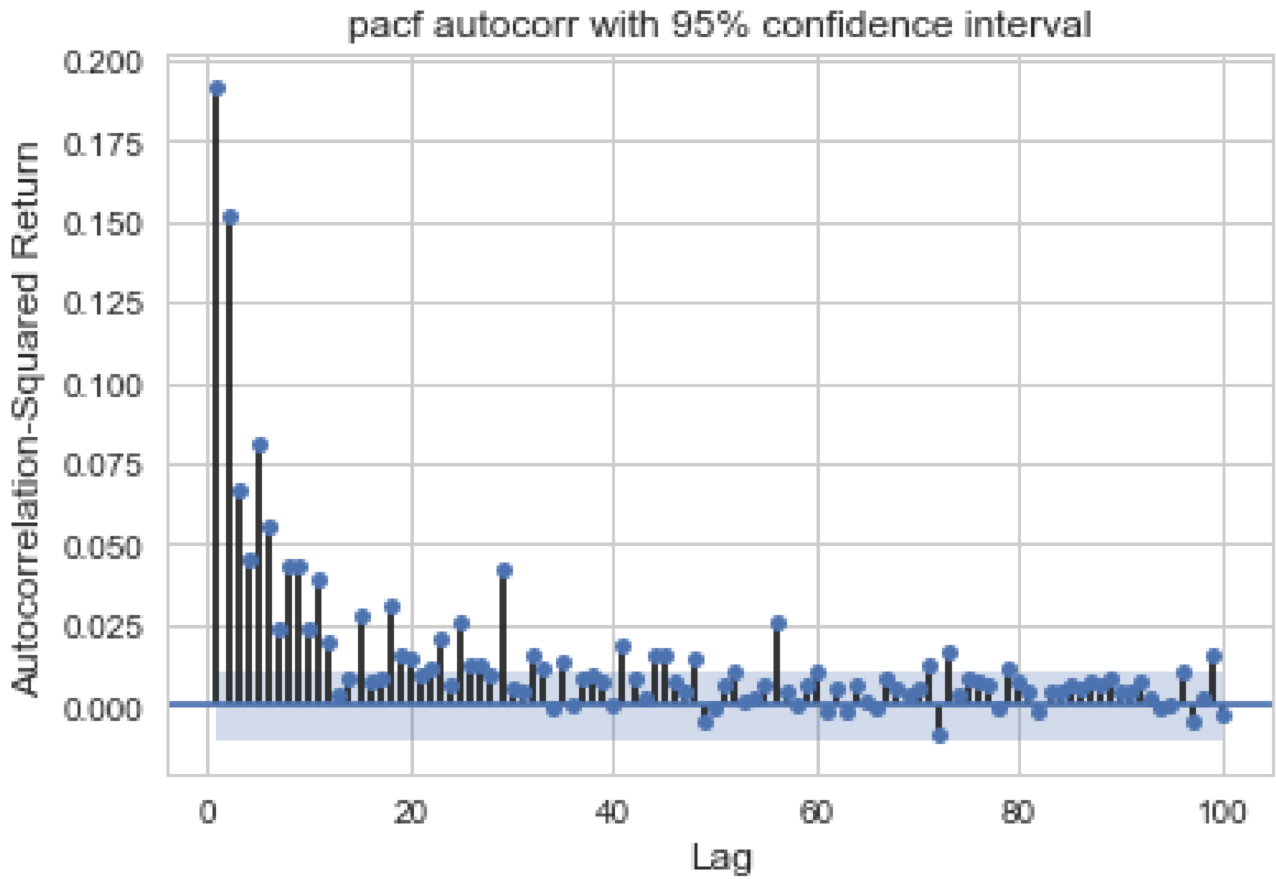
[Lmstat, pval_lm] = lmtest1(ret_Stock,Lags);

Lmstat = 3.4467; pval_lm =0.0634

Autocorrelation

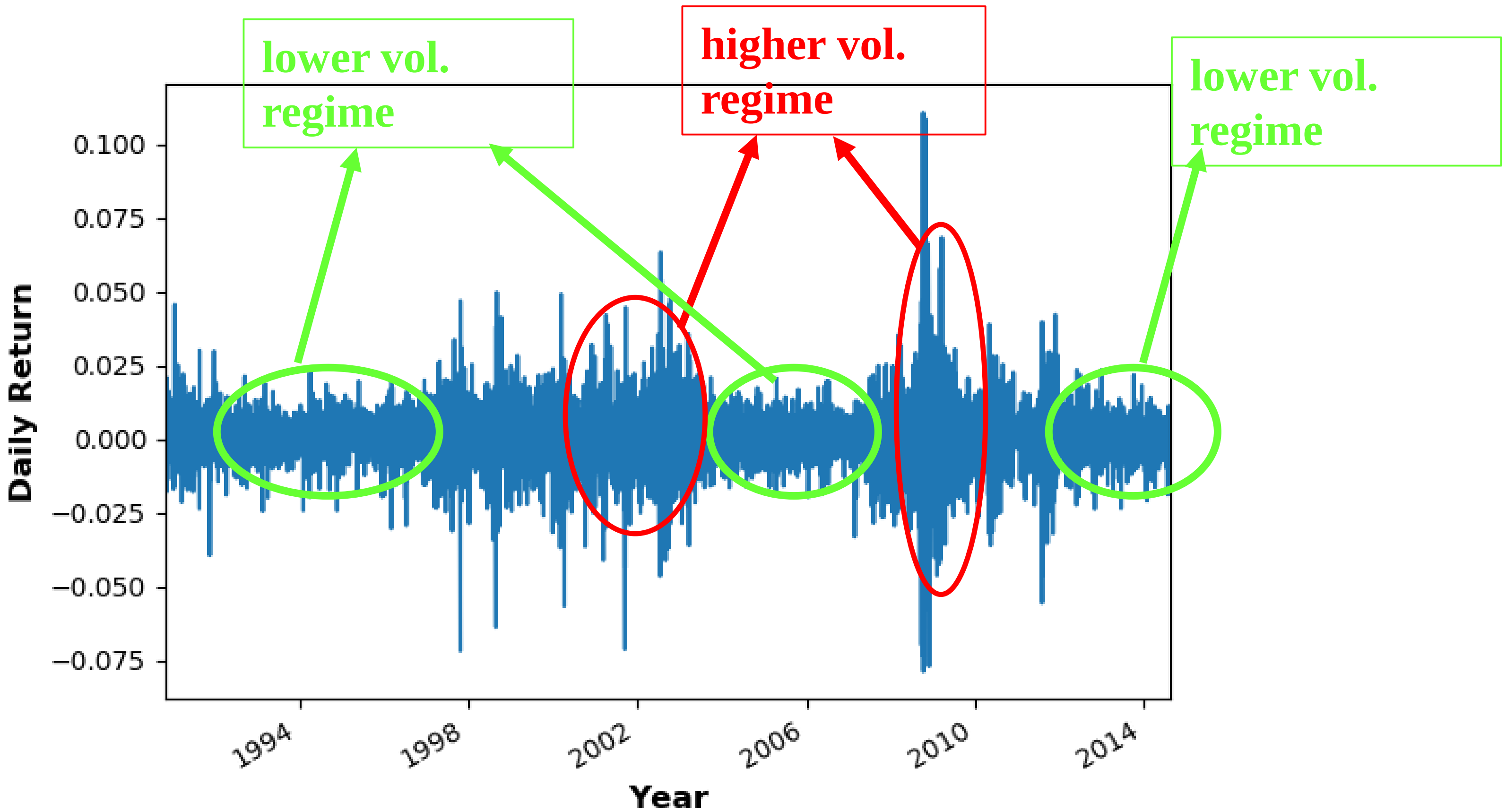
- Significance of autocorrelation coefficients can be tested by using the Ljung-Box (LB) test or the Engle LM test
  - *t0, p0 = tsd.acorr_ljungbox(Ret_Dow, lags=10), where t0 and p0 are the tstats and pvalue of the tests*
- Null hypothesis is that return data has no significant autocorrelation.
- High tstat or low pvalue suggests rejection the null hypothesis, which means data has statistically significant autocorrelation

Autocorrelation of Squared Returns



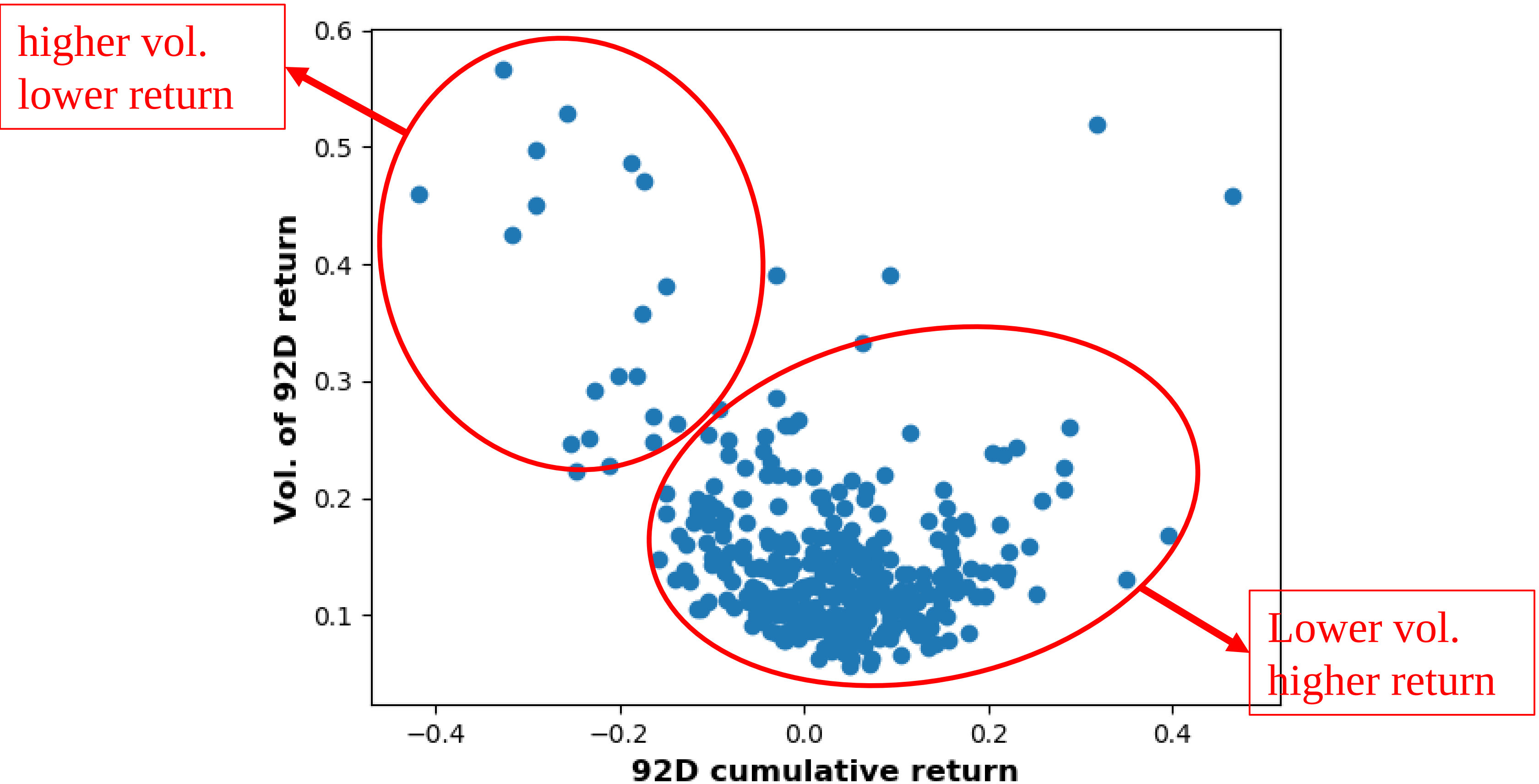
- Autocorrelation of Squared returns is statistically significant
- Volatility ~ return^2

Persistence of Volatility



- US equity returns exhibits higher/lower vol. regimes
- The regimes persist before switching to a different one

Volatility Clustering – US Equities Returns



- Volatilities demonstrate clustering behavior
- Volatilities tend to negatively correlate with returns

Nonlinear Dependence

- Nonlinear dependence: dependence between different returns changes according to market conditions
  - Returns are more correlated in volatile markets
  - Returns are more correlated in down markets
  - Cross asset/security correlation can approach 1 during market stress
- If returns were jointly normal, correlations would decrease for extreme event, but empirical evidence shows the opposite
- Assumption of linear dependence does not hold when it matters most

Example of Nonlinear Dependence

- Correlation of Microsoft, Morgan Stanley Goldman Sachs and Citigroup are higher during the financial crisis
- Based on daily returns

May 5, 1999 - June 12, 2015

	MSFT	MS	GS
MS	46%		
GS	46%	81%	
C	37%	65%	63%

August 1, 2007 - August 15, 2007

	MSFT	MS	GS
MS	93%		
GS	82%	94%	
C	87%	93%	92%

Exceedance Correlation

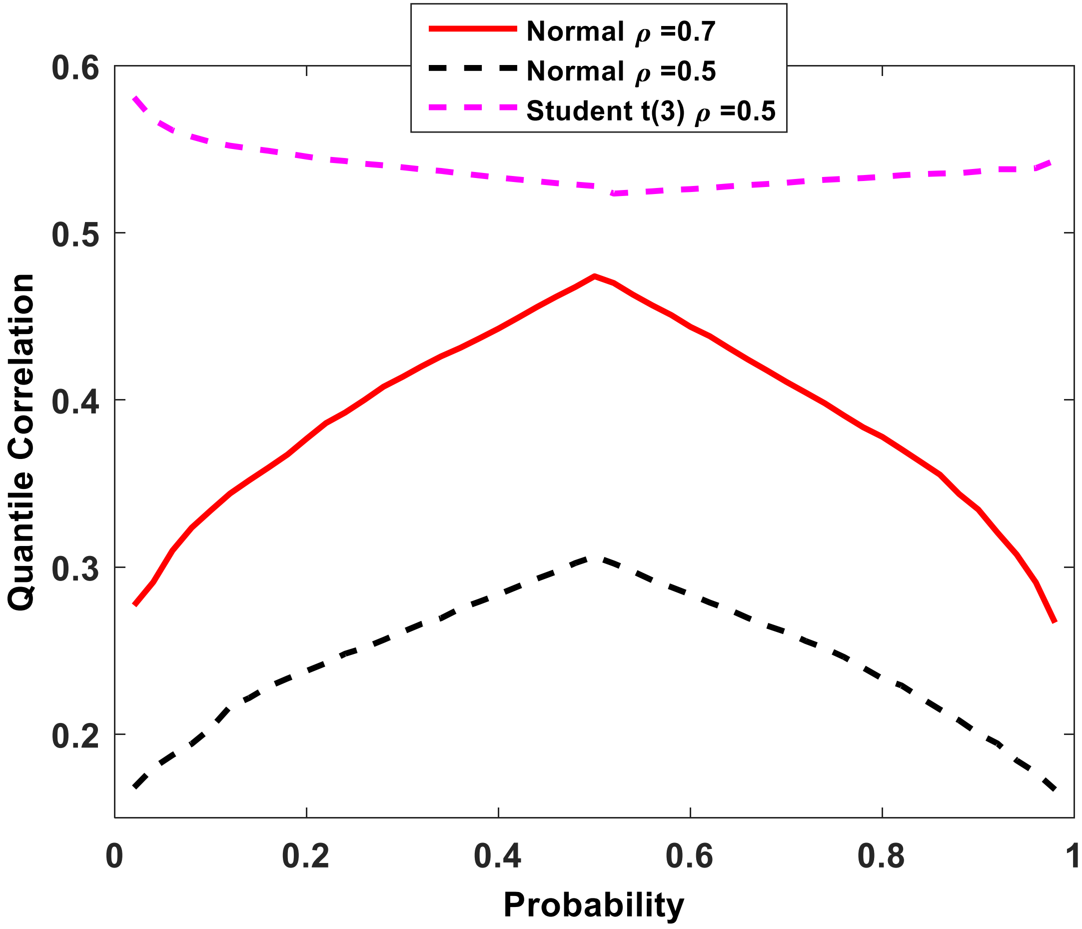
- Exceedance correlations show the correlations of (standardized) stock returns X and Y as being conditional on exceeding some threshold

$$\tilde{\rho}(p) = \begin{cases} Corr[X,Y|X \leq Q_X(p)\ and\ Y \leq Q_Y(p)]\ for\ p \leq 0.5 \\ Corr[X,Y|X > Q_X(p)\ and\ Y > Q_Y(p)] for\ p > 0.5 \end{cases}$$

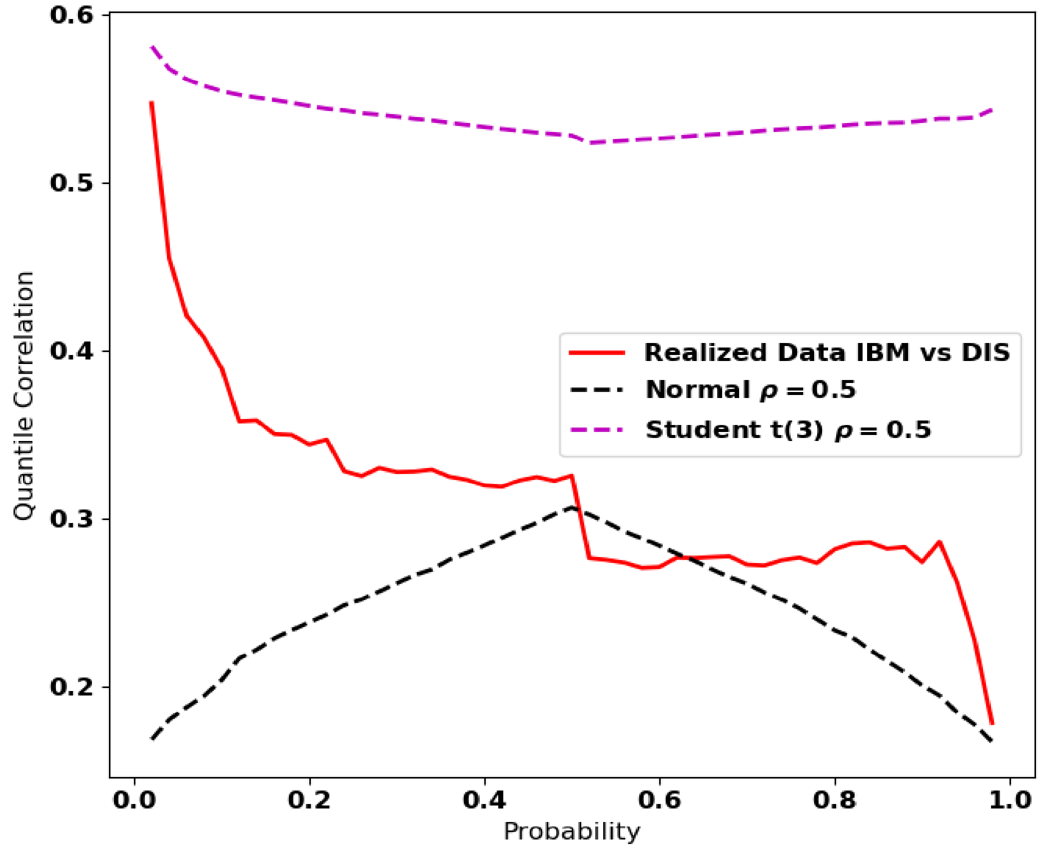
where $Q_X(p)$ and $Q_Y(p)$ are the p-th quantiles of X (or Y) given a distributional assumption

- Can be used to detect nonlinear dependence

Exceedance Correlation Plot



Exceedance Correlation Plot



- IBM and Disney return are almost perfectly correlated at the left tail (when returns are very negative)
- Data period: Jan. 1986 to June 2015

Brandeis

Stylized Facts about Returns

- Returns are not normally distributed
- Realized return distributions often have fat left tails
- Risk management in finance is very often concerned about forecasting the tail risk of a portfolio
- Methodologies such as Value at Risk (VaR), Conditional Value at Risk (CVaR) are measures currently used in the industry to measure tail risks
- We will spend half of the time in Session II to discuss these topics in details

Stylized Facts about Volatility

- Historical return data exhibits persistent periods of high and low volatility in returns
- Existence of volatility clusters/regimes: high vol. regimes will persist until it changes into low vol. regimes
- Autocorrelation is statistically significant
- Risk calculation methods such as exponential moving average take into considerations of these observations
- We will provide more details on these volatility forecasting methods when we discuss risk forecasting

Stylized Facts about Correlation

- Correlation is not stationary
- Correlation goes up when the market is volatile and going down
- Correlation approaches 1 for assets in the same class during market stress

Getting Security level data from Yahoo Finance

- Get data for the given name from the yahoo finance website

```
data = pdr.get_data_yahoo(
        tickers = ["SPY", "IWM", "..."], # tickers list (single tickers accepts a string as well)
        start = "2017-01-01", # start date (YYYY-MM-DD / datetime.datetime object)
            # (optional, defaults is 1950-01-01
        end = "2017-04-30", # end date (YYYY-MM-DD / datetime.datetime object)
            # (optional, defaults is Today)
        as_panel = False,  # return a multi-index dataframe
            # (optional, default is Panel, which is deprecated)
        group_by = 'ticker', # group by ticker (to access via data['SPY'])
            # (optional, default is 'column')
        auto_adjust = True, # adjust all OHLC automatically
            # (optional, default is False)
        actions = True,  # download dividend + stock splits data
            # (optional, default is None)
            # options are:
            #   - True (returns history + actions)
            #   - 'only' (actions only)
      threads = 10  # How may threads to use?
```

- More details at https://pypi.org/project/fix-yahoo-finance/

Two functions to help you get data from Yahoo

- def getDataBatch(tickers, startdate, enddate): get daily adjusted closing price data for a list of stocks/indices

- def getReturns(tickers, start_dt, end_dt, freq='monthly’): get monthly adjusted closing price data for a list of stocks/indices

Appendix

Data Definition

- P_t price of security/index at time t
- Arithmetic Return $R_t = \frac{P_t - P_{t-1}}{P_{t-1}}$ (here price is adjusted for income)
- Continuously compounded return:

$$r_t = \log(P_t/P_{t-1}) = \log P_t - \log P_{t-1}$$

- Unconditional and conditional standard deviation: σ, σ_t

Continuously Compounded Returns

- One period continuously compounded Return

$$r_t = \log(P_t/P_{t-1}) = \log P_t - \log P_{t-1}$$
$$= \log(1 + R_t)$$

- n period continuously compounded returns:

$$r_t(n) = \log(1 + R_t(n))$$
$$= \log\,(1+R_t)(1+R_{t-1})\,\ldots(1+R_{t-n+1})$$
$$= \log\,(1+R_t) + \log\,(1+R_{t-1}) + \ \ldots\ + \log(1+R_{t-n+1})$$
$$= \log P_t - \log P_{t-1} + \log P_{t-1} - \log P_{t-2} + \ldots \log P_{t-n+1} - \log P_{t-n}$$
$$= \log\,(P_t) - \log\,(P_{t-n})$$
$$= \log\,(P_t/P_{t-n})$$

Connection between Returns

- For short horizons and when return magnitude is small

$$r_t \cong R_t$$

- E.g:

  $R_t = 2\%$, $r_t = \log(1 + R_t)$=log(1+2%) =1.98%

  $R_t = 50\%$, $r_t = \log(1 + R_t)$=log(1+50%) =40.55%

Portfolio Returns

- If

$$R_t = arithmetic\ return\ of\ asset\ k$$
$$r_{t,k} = continuous\ return\ of\ asset\ k$$
$$w_k = weights\ of\ portfolio\ in\ asset\ k$$

- Portfolio return

$$R_{t,port} = \sum_{k=1}^{N} w_k R_{t,k}$$
$$r_{t,port} \neq \sum_{k=1}^{N} w_k r_{t,k}$$

Mean, Standard Deviation, Covariance and Tracking Error

- 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

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

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

- Tracking error

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