UDACITY

AI FOR TRADING: TERM I

OFFLINE INSTRUCTIONS

Another poster:
Zips all the files in the workspace so you can download them all at the same time:

```
zip -r workspace.zip ./*
```

Zips just data:

```
zip -r data.zip ../../data/*
```

Rama Krishna B:
1)First Download the data/module* folder. This depends on your exercise. Try something line " !tar -cvf my.tar /data/ " and download and extract my.tar to your local workstation.

2)Change os.environ['ZIPLINE_ROOT'] variable in your python notebook. Set it to appropriate place based on where you extracted your data folder from the my.tar. Look at the change in my iPython notebook and do something similar. One easy way to do is to print the existing os.environ['ZIPLINE_ROOT'] and work accordingly.

3)If there is any reference to any file in the data folder, change the path accordingly.

4)Look at https://colab.research.google.com/drive/1Dj1fLiEA122N95vjgfLNOSRda5IbplIH That is an exercise file from from one of our lessons.

To download twits data:
run the following in your notebook:
Copy the data to the workspake dir with: !cp -r ../../data/ ../workspace/
Zip the data using: !zip -r data.zip data/
— I used json.dump() once the data was already loaded into notebook

```
with open(‘file_name.json', 'w') as f:
    json.dump(twits, f)
```

MODULE 1: MOMENTUM TRADING

Lesson 7: Distribution of Returns and Prices

Xxxxxxxxxx

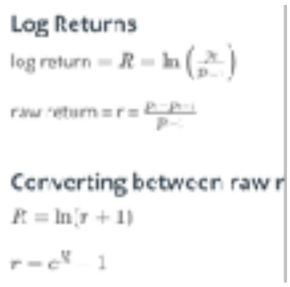
Sebastian, this is kind of a specific question: How long do you think before we can do away with hand-tuning of hyperparameters for Neural Networks?

Gohar: Immediately. I use a procedure named "fiddle" what all of my Stanford PhD students use all the time. It's simply parameter wiggling automated. When you hand tune, you can auto-tune using your own hand tuning method

Log Returns

Let's summarize what we just learned. These are some generally accepted reasons that quantitative analysts use log returns:

1. Log returns can be interpreted as continuously compounded returns.
2. Log returns are time-additive. The multi-period log return is simply the sum of single period log returns.
3. The use of log returns prevents security prices from becoming negative in models of security returns.
4. For many purposes, log returns of a security can be reasonably modeled as distributed according to a normal distribution.
5. When returns and log returns are small (their absolute values are much less than 1), their values are approximately equal.
6. Logarithms can help make an algorithm more numerically stable.

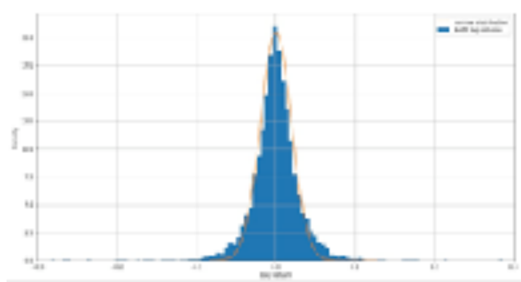


Stepping back, it may not be immediately obvious why all these attributes are benefits. Don't worry about this. As you progress in this course and beyond, you will see more applications of returns and log returns in trading strategies and algorithms and you'll be able to better appreciate why they are used.

Distribution of Returns and Prices

An important first step is to think of stock prices and returns as *random variables*, i.e. outcomes of random phenomena, that take on values as described by *distributions*. Distributions allow us to summarize the behavior of random variables. So, what are the distributions of returns and prices?

If we calculate returns and log returns on stock price and plot them, we'll see something like the graphs below (price (lhs-top); returns (lhs-bottom)) and distribution of returns (rhs):



The tails of the histogram clearly lie above the tails of the normal distribution. We call these "fat tails". In general, the normal distribution can be a reasonable approximation for short-term returns and log returns for some applications. However, many analyses have shown that the data do not conform perfectly to a normal distribution, and often deviate significantly in the tails. The significance of this is that the normal distribution predicts fewer extreme events than are actually observed. The conversation about the best model for the distribution of returns has been going on for at least the past century. The best model will depend on exactly what your analysis seeks to achieve.

Normality and Long-Term Investments

Based on historical data, it may be reasonable to consider short-term returns as approximately normally distributed for some purposes. However, even if short-term returns are normally distributed, long-term returns cannot be.

If $r_1 = \frac{p_1 - p_0}{p_0}$ and $r_2 = \frac{p_2 - p_1}{p_1}$ are normally distributed, the sum of these, r_1 + r_2, would be normally distributed. But the two-period return is not the sum of the one-period returns.

The product (1 + r_1)(1 + r_2) shown left is not normal, and becomes noticeably less normal as the product grows over time.

Two-period return = $\frac{p_2 - p_0}{p_0}$

$$\frac{p_2 - p_0}{p_0} + 1 = \frac{p_2 - p_0}{p_0} + \frac{p_0}{p_0} = \frac{p_2}{p_0}$$

$$= \frac{p_1}{p_0} \times \frac{p_2}{p_1}$$

$$= (1 + r_1)(1 + r_2)$$

DISTRIBUTION OF LOG RETURNS

So long-term prices and cumulative returns can be modeled as approximately lognormally distributed because they are *products* of independently, identically distributed (IID) random variables. On the other hand, **log returns** *sum* over time. Therefore,

if $R_1 = \ln\left(\frac{p_1}{p_0}\right)$ and $R_2 = \ln\left(\frac{p_2}{p_1}\right)$ are normal, their sum, the two-period log return, is also normal.

Even if they are not normal, as long as they are IID, their long-term sum will be approximately normal, thanks to the Central Limit Theorem. This is one reason why using log returns can be convenient for modeling purposes.

Project 1: Momentum

Reviewer Notes | Comments

Student Hub | Knowledge Discussions

PROJECT 1, FUNCTION 'PORTFOLIO_RETURNS': CAN'T UNDERSTAND WHY WE NEED TO DIVIDE THE RESULT BY N_STOCKS
Can anyone explain, why in the project we need to divide the final results by 'n_stocks' in the function 'portfolio_returns'?
If we multiply lookahead returns by the binary df's, we get returns of each stock for each date already. Why divide by n_stock?
Or is it just some convention to use in the next cell ('View Data'), where to get the average returns for each day we just sum up all the tickers' returns from the function 'portfolio_returns'?
If this is the case, then all this is not really intuitive. I think in this case there should not be this division by 'n_stocks' in 'portfolio_returns', but instead it should be implemented in the next cell in 'plot_returns' by dividing the transposed sum by 2*top_bottom_n.

The reason why *n_stocks* is provided is hinted at by this comment in the project: **"*This makes it easier to compute a portfolio's returns as the simple arithmetic average of the individual stock returns*."** Since you have a finite amount of money to invest in a given series of stocks, if you divide your total money into *n_stocks* worth of different bets evenly, then your *total return* is the **average** of the *individual returns* of those *n_stocks. — NickD*

I think I understood the division by `n_stocks`. While it may seem confusing that the *portfolio_returns* dataframe contains the stock returns divided by the number of stocks, that lets us calculating an average when we then do the sum by date (see *expected_portfolio_returns_by_date*). The *expected_portfolio_return_by date* is in fact calculated as the sum of the transpose of *expected_portfolio_returns*. But because the *expected_portfolio_returns* already contains the return from each of the stocks we invested in, divided by the number of stocks, doing a sum by date of the transpose dataframe is equivalent to calculating the mean.
In notation, that is equivalent to calculate the mean as:

(x/n + y/n + z/n) instead of (x + y + z)/n

It took me a bit to understand, but it is clear for me now. — GiacomoS

PREVIOUS RETURN VS LOOKAHEAD RETURNS VS CURRENT RETURNS
in this project we're calculating the previous return, and we're using them to rank the stock. Next, we use lookahead returns. But why aren't we using anywhere the current results?

Eg: let's say we are in July, we use June EOM (previous) returns to choose which stock to sell or to buy, and then August EOM ones (lookahead) for the returns, Shouldn't we use July EOM (current) returns instead?

Or, viceversa, use July EOM to choose stocks, and August EOM to calculate actual returns?

It seems to me we're missing a month somewhere…

Although we are devising a strategy using a known historical data set, our strategy will ultimately be used in real-time. That means if we are in a given month, we only know what the return was previously. So if we are in July, we don't know the current month's return yet. It hasn't happened.

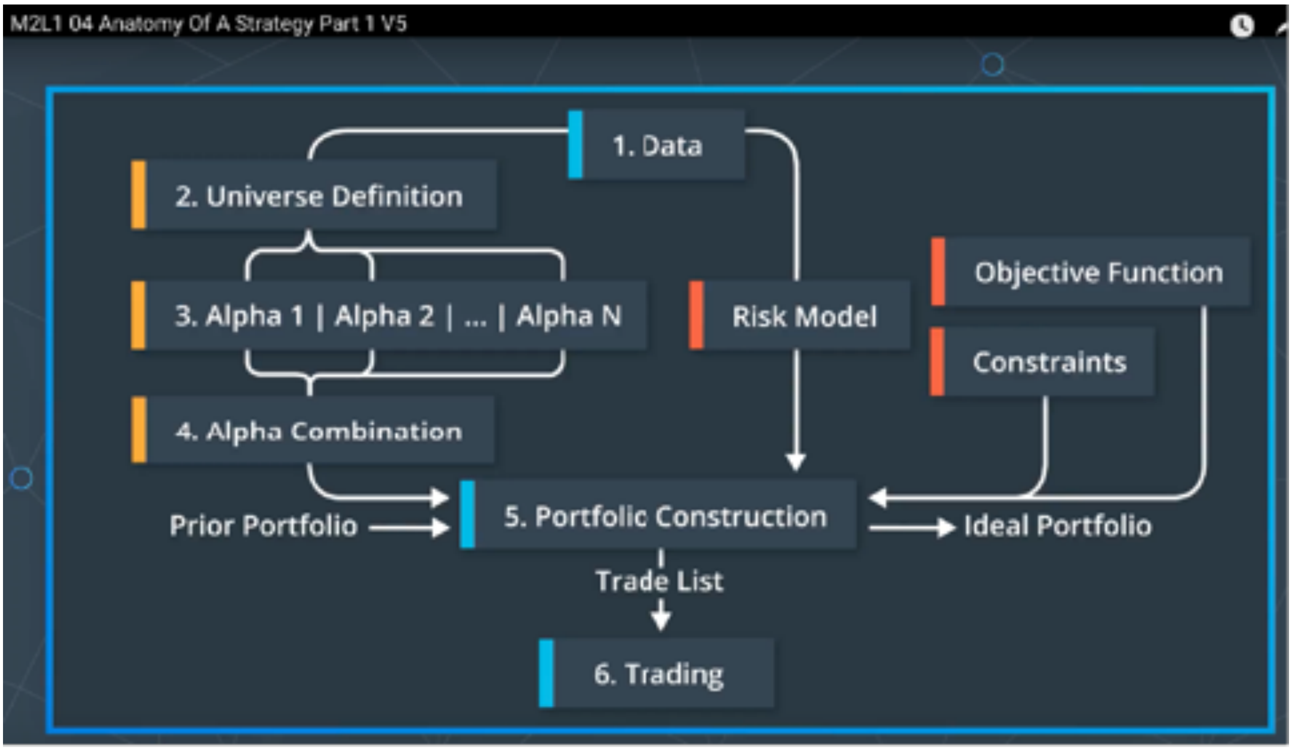
All we can do is look at the previous month (June) and see which stocks performed well for our decision.

Afterwards, to evaluate the strategy we have to check the return for the stock we bought. Return is P(t) - P(t-1). Here P(t-1) is July and therefore we need to find P(t) which is the following month August. Hence the term “lookahead”. — Thomas K

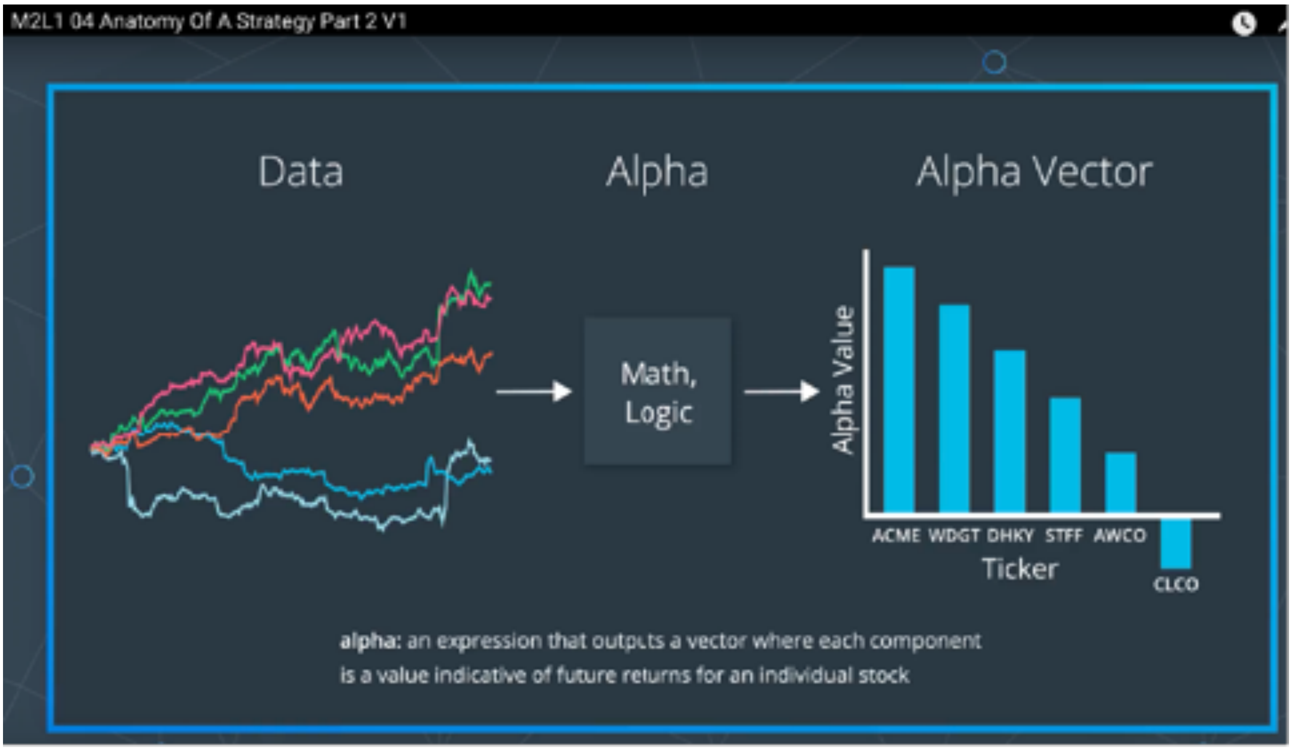
MODULE 2: BREAKOUT STRATEGY

Lesson 10: Quant Workflow

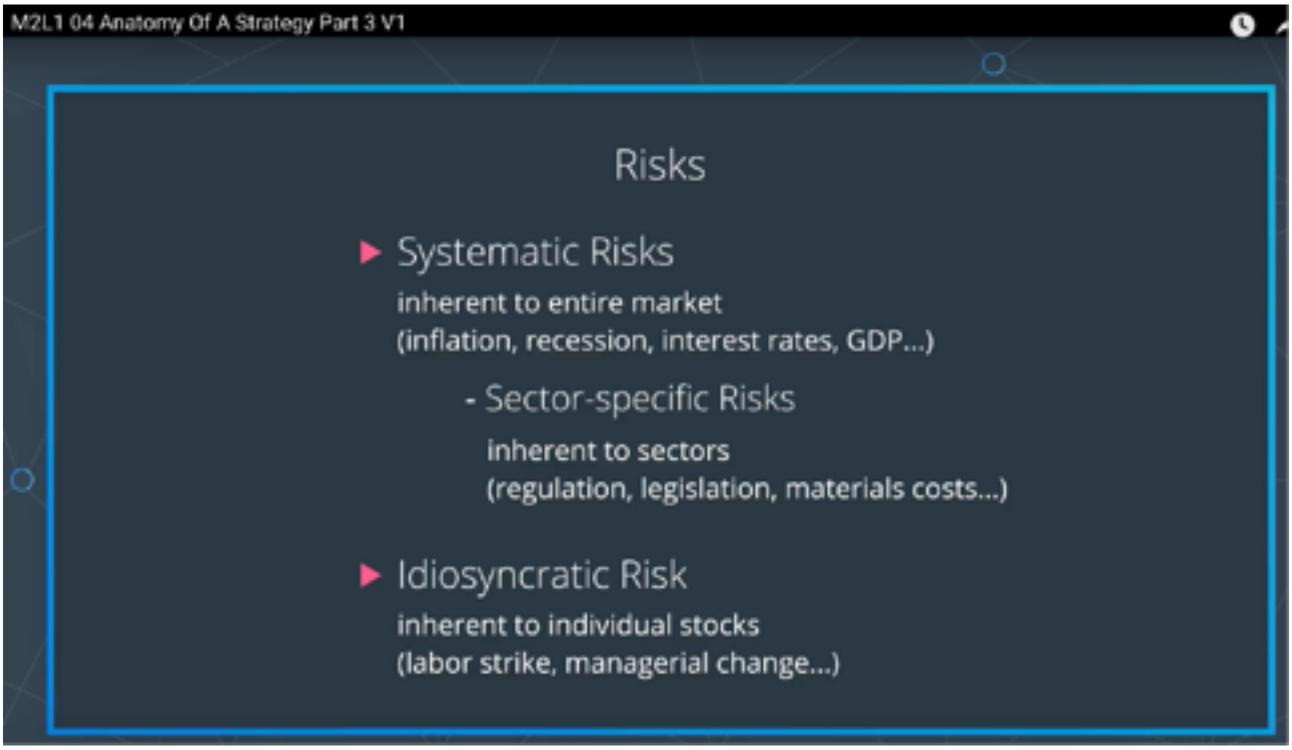
Cross-Sectional Strategy



Alpha Vectors

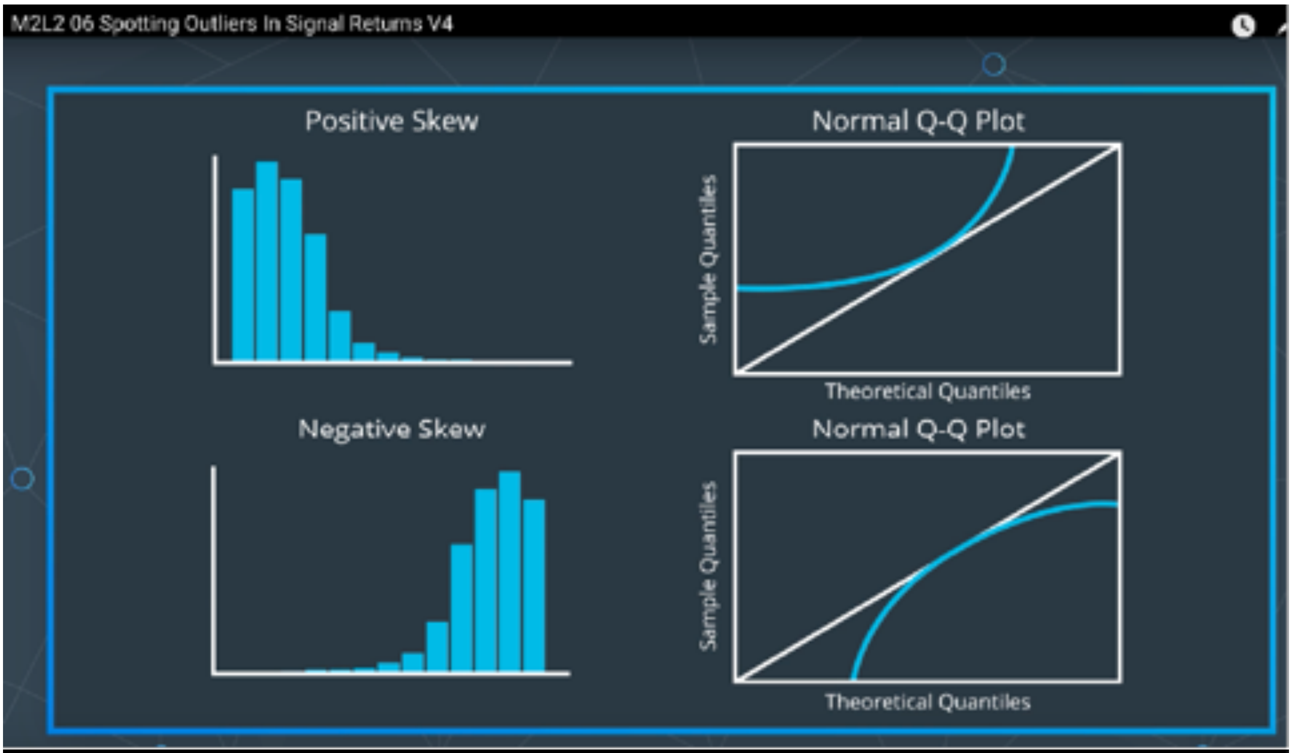


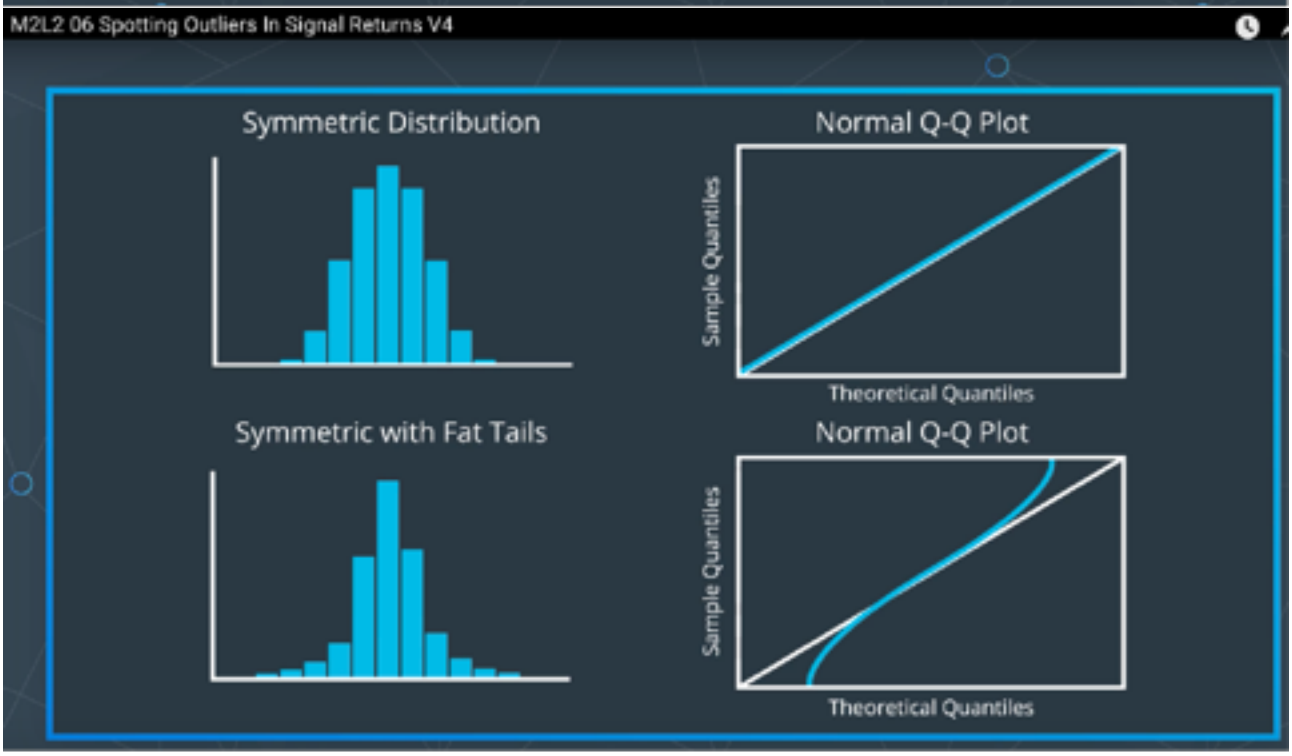
Risk


M2L1 04 Anatomy Of A Strategy Part 3 V1
Risks
Systematic Risks
inherent to entire market
(inflation, recession, interest rates, GDP...)
- Sector-specific Risks
inherent to sectors
(regulation, legislation, materials costs...)
Idiosyncratic Risk
inherent to individual stocks
(labor strike, managerial change...)

Lesson 11: Spotting Outliers in Signal Returns

QQ Plots





Lesson 12: Regression

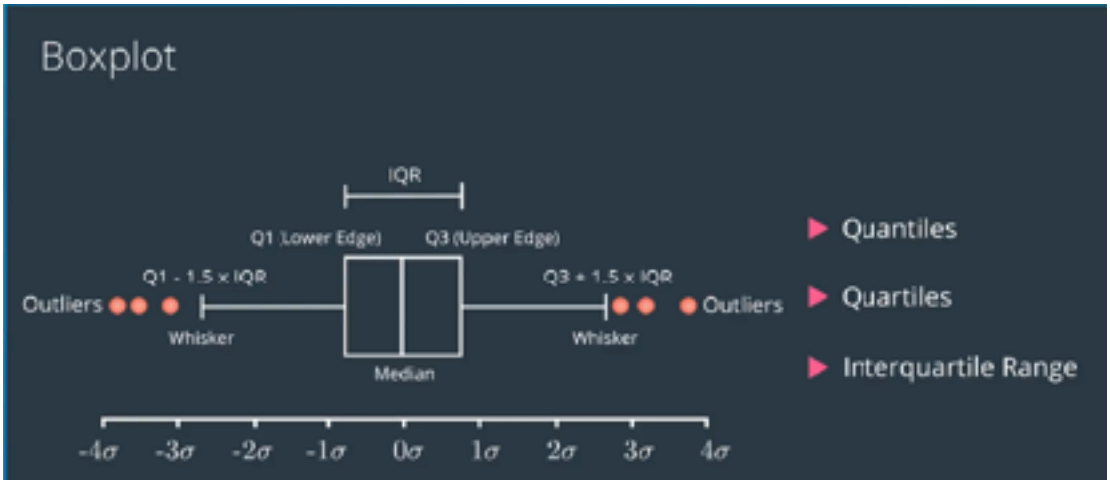
Distributions

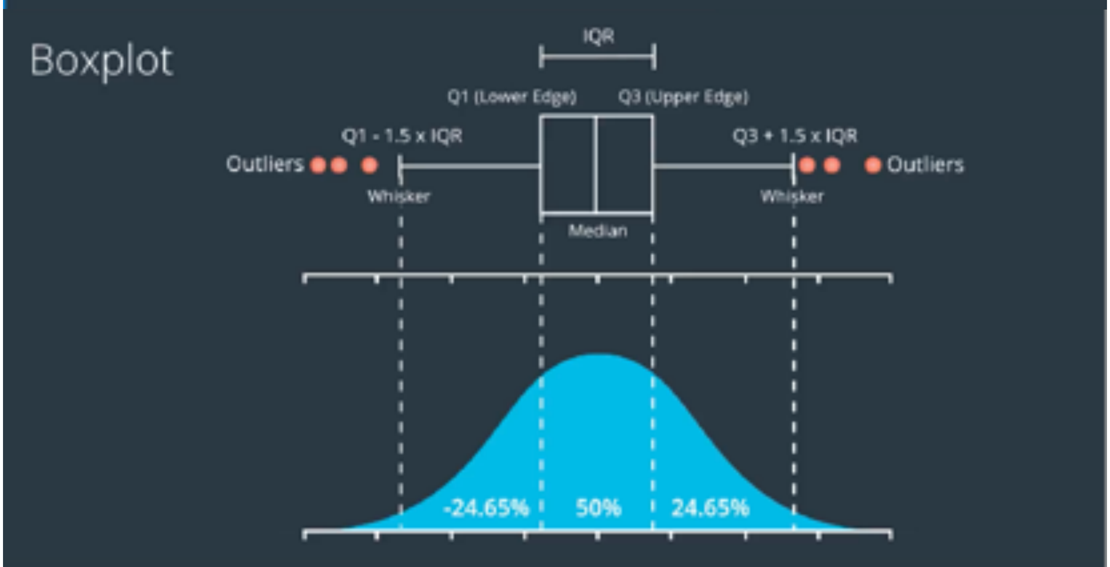
- If not normal, test for validity could erroneously endorse a model that is not valid
  - Types: normal, log-normal, exponential, uniform
- Random variable —> normally distributed within some constraints

PARAMETERS
- Formula for prob density function (PDF) has parameters to change its shape
- Notation: X ~ D means a random variable ‘X’ follows a probability distribution ‘D’
  - P(x|D) = p(x) reads: probability of x given D
  - So: P(2|N) = p(2) = a number between 0 and 1 <— the probability
  - ’N’ is for normal distribution
- The parameters are:
  - μ = mean
  - σ = standard deviation

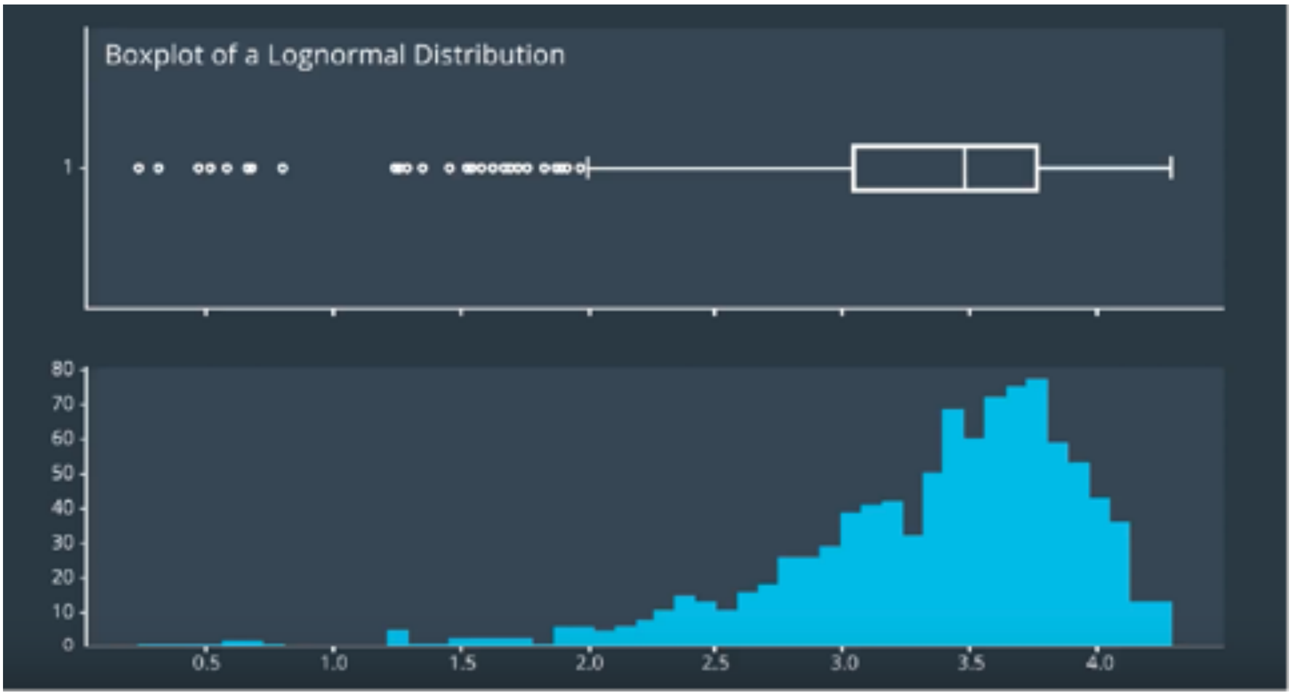
Testing for normality

A histogram lets us check if a distribution is symmetric/skewed, and if it has fat tails. QQ plots help us compare any two distributions, so they can be used to compare distributions other than the normal distribution. A box whisker plot lets us check for symmetry around the mean. A QQ-plot lets us compare our data distribution with a normal distribution (or any other theoretical "ideal" distribution). “goodness of fit”





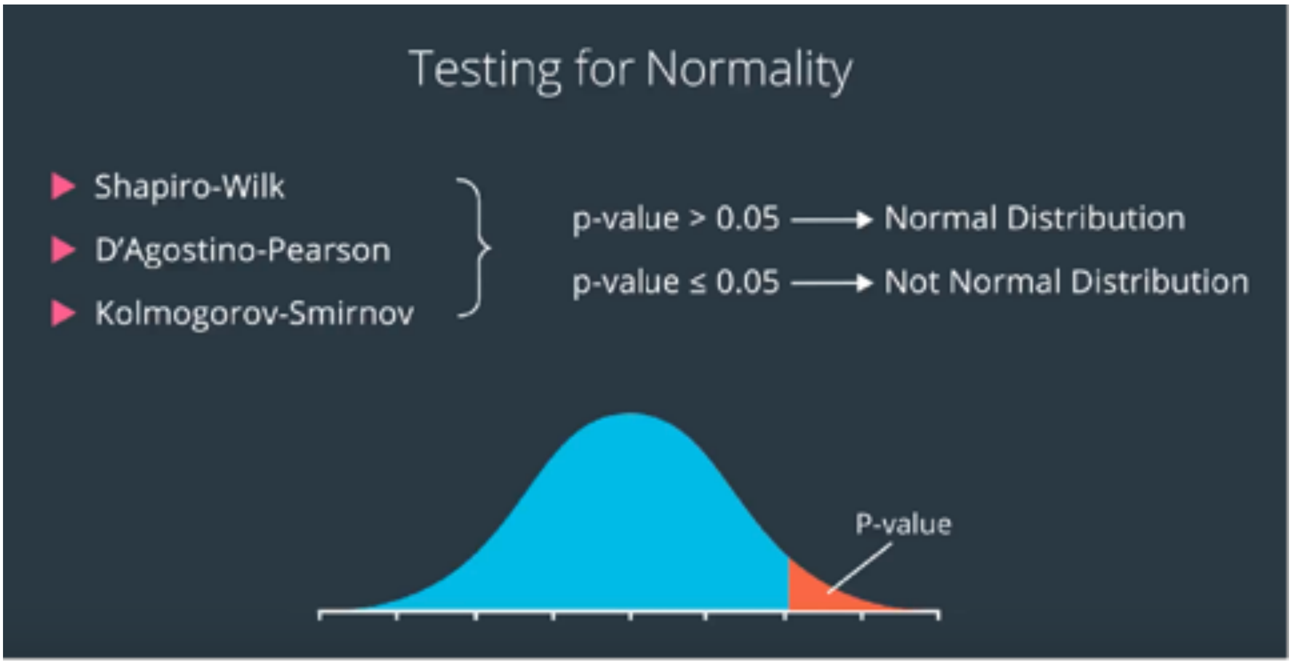
Boxplot of a Normal Distribution



HYPOTHESIS TESTS
There are three hypothesis tests that can be used to decide if a data distribution is normal. These are the **Shapiro-Wilk** test, the **D'Agostino-Pearson** test, and the **Kolmogorov-Smirnov** test. Each of these produce a p-value, and <u>if the p-value is small enough, say 0.05 or less, we can say with a 95% confidence that the data is not normally distributed</u>. Shapiro-Wilk tends to perform better in a broader set of cases compared to the D'Agostino-Pearson test. In part, this is because the D'Agostino-Pearson test is used to look for skewness and kurtosis that do not match a normal distribution, so there are some odd non-normal distributions for which it doesn't detect non-normality, where the Sharpiro-Wilk would give the correct answer.

The Kolmogorov Smirnov test can be used to compare distributions other than the normal distribution, so it's similar to the QQ plot in its generality.

To do a normality test, we would first <u>rescale the data (demean)</u> (subtract the mean and divide by its standard deviation), then compare the rescaled data distribution with the standard normal distribution (which has a mean of zero and standard deviation of 1). In general, the Shapiro-Wilk test tends to be a better test than the Kolmogorov Smirnov test, but not in all cases.



HETEROSKEDASTICITY
One of the assumptions of linear regression is that its input data are homoscedastic. A visual way to check if the our data is homoscedastic is a scatter plot (like the one we saw in the video).
If our data is heteroscedastic, a linear regression estimate of the coefficients may be less accurate (further from the actual value), and we may get a smaller p-value than should be expected, which means we may assume (incorrectly) that we have an accurate estimate of the regression coefficient, and assume that it's statistically significant when it's not.

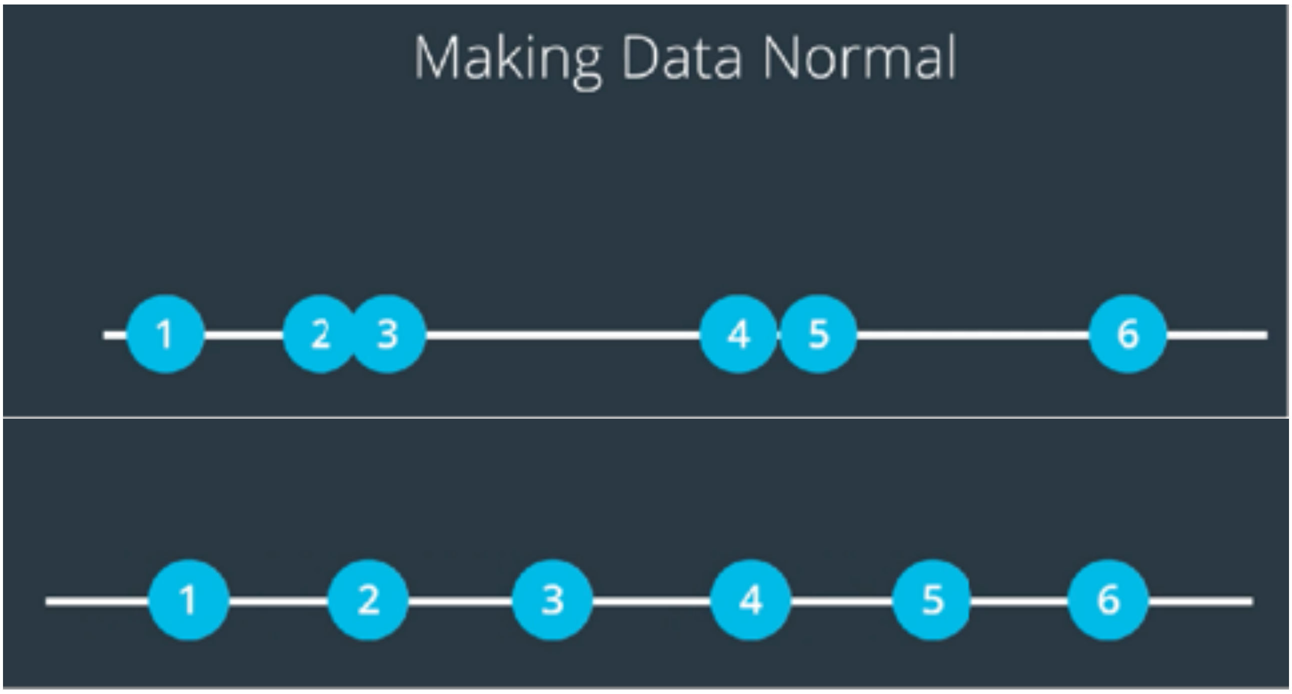
Breusch-Pagan Test
Note, we'll cover the Breusch-Pagan test for heteroscedasticity in more detail after we learn about regression.

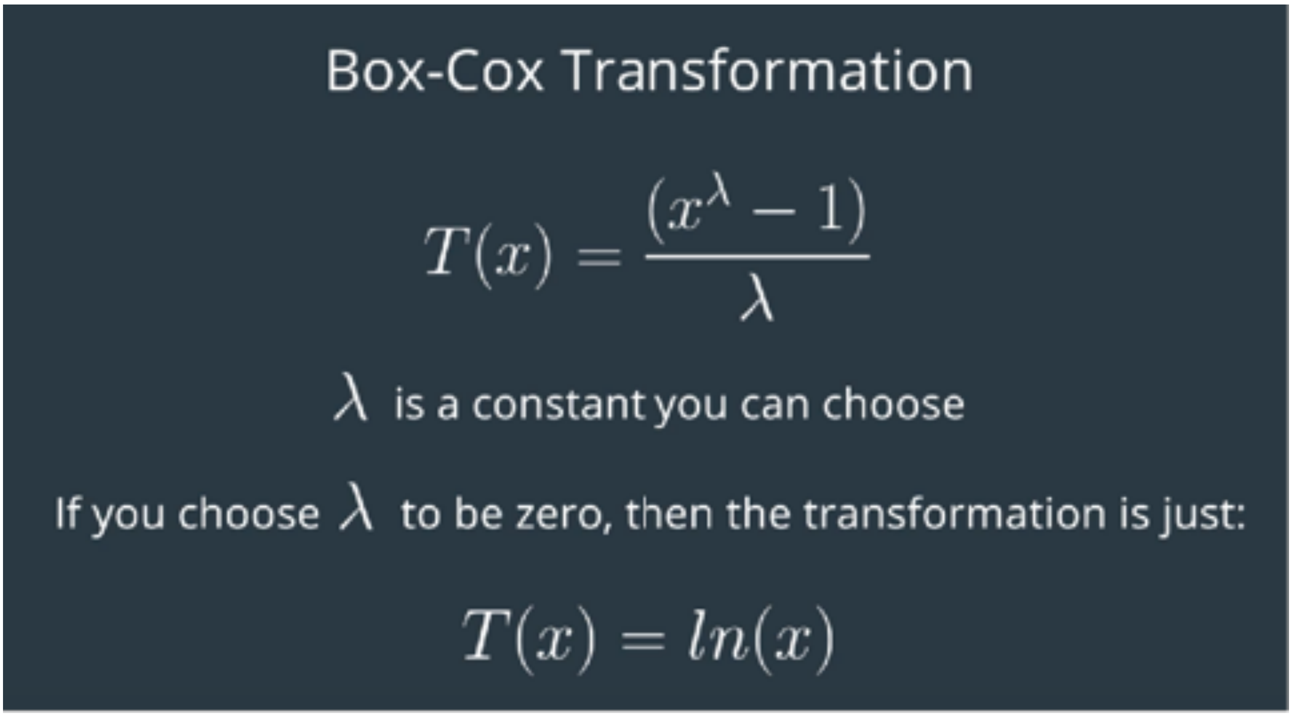
- P-value < 0.05 ==> hetroscedastic
- P-value > 0.05 ==> homoscedastic

Stationary data: mean, variance and covariance stay consistent (stationary) over time

Transforming Data
Make normal by apply log function to it
Make data homoscedastic, can apply the time difference ==> just convert to log returns!!

BOX-COX TRANSFORMATION
A monotonic transformation (evens out the spacing, but maintains order)





Linear Regression

Residuals (ie: error terms) = y_actual = y_pred

- If the residuals are normal distributed (ie: μ=0 and σ is constant), the the residuals can be considered as random (noise)
  - If not, the model is probably biased

EVALUATING THE MODEL

R-squared:

- Ranges from 0 to 1; 1 means all variation of dependent variables is explained by the independent variables

Adjusted R-squared: better metric

- Helps to find the minimum number of independent variables that are most relevant

F-test:

- Checks whether the model co-efficients and intercepts (weights) are non-zero and therefore meaningful
  - P-value <= 0.05 then can assume weights are not zero, meaning the model does describe a meaningful relationship

BREUSCH-PAGAN TEST FOR HETEROSCEDASTICITY (REVISITED)

The Breusch-Pagan test is one of many tests for homoscedasticity/heteroscedasticity. It takes the residuals from a regression, and checks if they are dependent upon the independent variables that we fed into the regression.

The test does this by performing a second regression of the residuals against the independent variables, and checking if the coefficients from that second regression are statistically significant (non-zero).

If the coefficients of this second regression are significant, then the residuals depend upon the independent variables. If the residuals depend upon the independent variables, then it means that the variance of the data depends on the independent variables. In other words, the data is likely heteroscedastic.

So if the p-value of the Breusch-Pagan test is ≤ 0.05, we can assume with a 95% confidence that the distribution is heteroscedastic (not homoscedastic).

BREUSCH-PAGAN TEST IN PYTHON

In Python, we can use the **statsmodels.stats.diagnostic.het_breuschpagan(resid, exog_het)** function to test for heteroscedasticity. We input the residuals from the regression of the dependent variable against the independent variables. We also input the independent variables that may affect the variance of the data. The function outputs a p-value.

Multivariate Linear Regression

Multiple regression: predicting one y with multiple x's
Multivariate regression: predicting multiple y's with multiple x's

- aka multivariate multiple regression?? Video made it look like the same thing…

REGRESSION IN TRADING

Difficult but worthwhile to learn

Lesson 13: Time Series Modeling

Autoregressive Models (AR)

AR models assume that past period returns provide some insight into the next period returns

Prior periods are on a rolling window basis, called lag: AR Lag 2 shown below

Should check different AR(p) with R^2

If using multiple time series, can use different AR(p) for each

To account for possible interdependence between variable, use Vector AR Model
- Will help in pairs trading later

Moving Average Models (MA)

$$y_t = \mu + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \ldots + \theta_q \epsilon_{t-q}$$

AUTOCORRELATION

Want to use the periods with high positive of negative variations.

Advanced Time Series Models

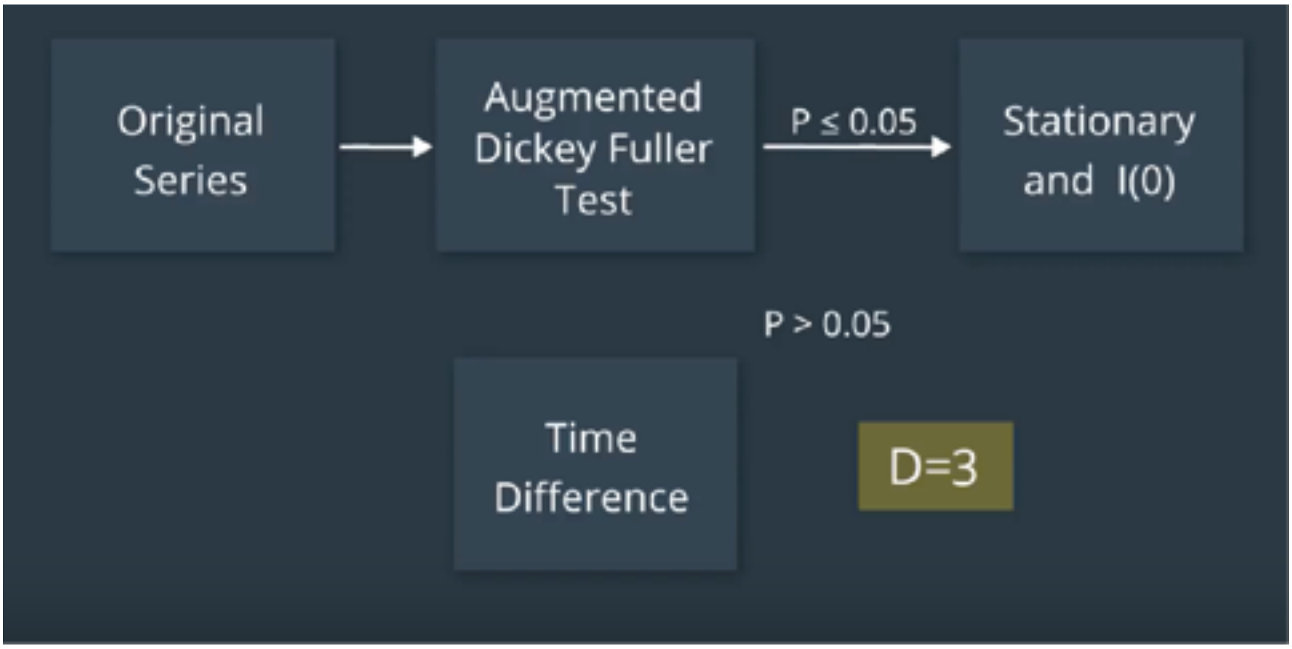
AUTOREGRESSIVE MOVING AVERAGE (ARMA)
AR and MA models tend to capture different relationships, hence combing them can be useful

- ARMA(p, q) :   y ~. AR(p): + MA(q):

$$y_t = \alpha + B_1 y_{t-1} + B_2 y_{t-2} + \dots$$
$$+ \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \dots$$

AUTOREGRESSIVE INTEGRATED MOVING AVERAGE (ARIMA)
Used in pair trading.

Use augmented Dickey Fuller test to see if data is stationary



SEASONAL ADJUSTMENTS USING ARIMA (SARIMA)
Time series data tends to have seasonal patterns. For instance, natural gas prices may increase during winter months, when it's used for heating homes. Similarly, it may also increase during peak summer months, when natural gas generators are used to produce the extra electricity that is used for air conditioning. Retail sales also has expected increases during the holiday shopping season, such as Black Friday in the US (November), and Singles' Day in China (also in November). Stocks may potentially have seasonal patterns as well. One has to do with writing off losses in order to minimize taxes. Funds and individual investors have unrealized capital gains or losses when the stock price increases or decreases from the price at which they bought the stock. Those capital gains or losses become "realized capital gains" or "realized capital losses" when they sell the stock. At the end of the tax year (which may be December, but not necessarily), an investor may decide to sell their underperforming stocks in order to realize capital losses, which may potentially reduce their taxes. Then, at the start of the next tax year, they may buy back the same stocks in order to maintain their original portfolio. This is sometimes referred to as the "January effect."

Removing seasonal effects can help to make the resulting time series stationary, and therefore more useful when feeding into an autoregressive moving average model.

To remove seasonality, we can take the difference between each data point and another data point one year prior. We'll refer to this as the "seasonal difference". For instance, if you have monthly data, take the difference between August 2018 and August 2017, and do the same for the rest of your data. **It's common to take the "first difference"** either before or after taking the seasonal difference. If we took the "first difference" from the original time series, this would be taking August 2018 and subtracting July 2018. Next, to take the seasonal difference of the first difference, this would mean taking the difference between (August 2018 - July 2018) and (August 2017 - July 2017).

You can check if the resulting time series is stationary, and if so, run this stationary series through an autoregressive moving average model.

Side Note
Kendall Lo, one of the subject matter experts of our course, recommends this book: "Way of the Turtle: The Secret Methods that Turned Ordinary People into Legendary Traders". The book is about how a successful investor trained his students (his "turtles") to follow his trend-following trading strategy. The book illustrates the concepts of using trading signals, back-testing, position sizing, and risk management. The story is also summarized in this article **Turtle Trading: A Market Legend**

Filters

KALMAN FILTERS
Single state represents the past, so no need to look at the earlier time periods or try to decide what the best lag is; handles noisy data well

KALMAN FILTERS FOR PAIRS TRADING
One way Kalman Filters are used in trading is for choosing the hedge ratio in pairs trading. We will get into pairs trading and hedge ratios in lesson 13 of this module, but for now, imagine that there's a magic number that you can estimate from a model, such as a regression model, based on time series data of two stocks.

Every day when you get another data point, you can run another regression and get an updated estimate for this number. So do you take the most recent number every time? Do you take a moving average? If so, how many days will you average together? Will you give each day the same weight when taking the average?

All of these kinds of decisions are meant to smooth an estimate of a number that is based on noisy data. The Kalman Filter is designed to provide this estimate based on both past information and new observations. So instead of taking a moving average of this estimate, we can use a Kalman Filter.

The Kalman Filter takes the time series of two stocks, and generate its "smoothed" estimate for this magic number at each new time period. Kalman Filters are often used in control systems for vehicles such as cars, planes, rockets, and robots. They're similar to the application in pairs trading because they take noisy indirect measurements at each new time period in order to estimate state variables (location, direction, speed) of a system .

Kalman Filters are not used in this module's project, but if you want to learn more (if you've finished the project early and have time), please check out Udacity's free Linear Algebra course: Lesson 4: Matrices and Transformation of State

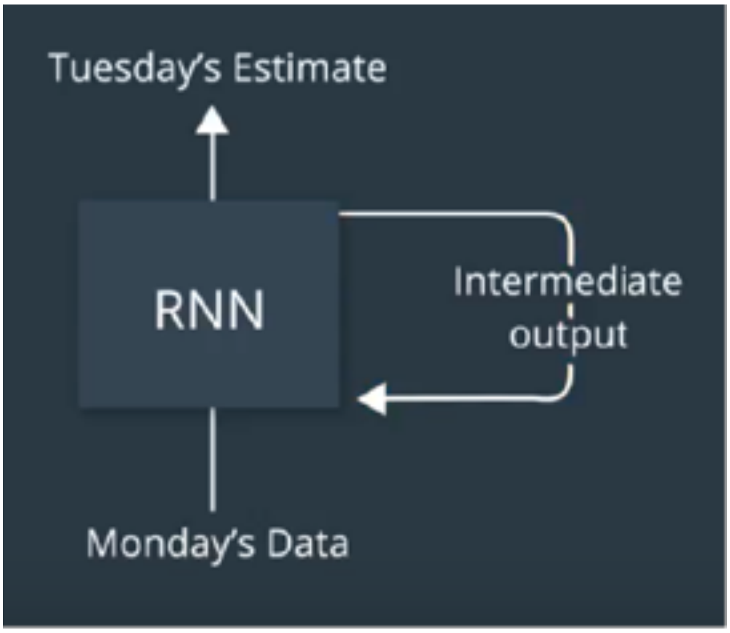
PARTICLE FILTERS
Genetic algorithm: uses natural selection to improve predictions
- Do not assume data is normally distributed
- Do not need linear relationships so can fit non-linear better

For an introduction to particle filters, please check out Sebastian Thrun's lesson in the free "Intro to Artificial Intelligence" course:
lesson 16 "HMMs and Filters: Node 18 "Particle Filters"

Recurrent Neural Networks

Recurrent Neural Networks are useful for processing sequence data in general, and are often used for Natural Language Processing. We will cover recurrent neural networks and natural language processing in term 2 of this program.

The intermediate output fed back into the NN allows the network to remember previous data (previous cell state)



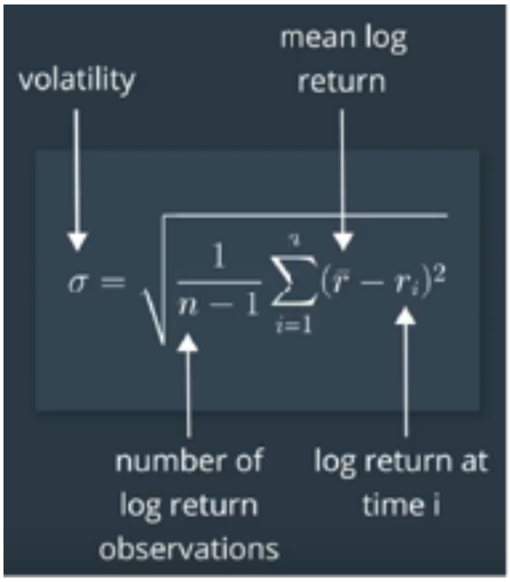
Lesson 14: Volatilty

Historical & Annualized Volatility

Volatility: important for
- Measuring risk
- Defining position sizes
- Designing alpha factors
- Pricing options
- Trading volatility directly

Formula is for a specific time frequency
Like daily, but can be annualized

$\sigma_{\text{year}} = \sqrt{12}\sigma_{\text{month}}$
$\sigma_{\text{year}} = \sqrt{52}\sigma_{\text{week}}$
$\sigma_{\text{year}} = \sqrt{252}\sigma_{\text{day}}$



Exponentially Weighted Moving Average

When working with daily log returns, the mean is small enough to ignore. Also, n-1 can just be n for large enough data series, so the volatility formula simplifies to the average of squared log returns.

$$\sigma_t^2 = \frac{1}{n-1}\sum_{i=1}^{n}(\overset{0}{\cancel{\bar{r}}} - r_{t-i})^2$$

Below is the formula for then finding the exponential weighted moving average of the variance. λ is the weight, so the denominator is the sum of the weights to provide the weighted average.

$$\sigma_t^2 = \frac{r_{t-1}^2 + \lambda r_{t-2}^2 + \lambda^2 r_{t-3}^2 + \ldots + \lambda^{n-1} r_{t-n}^2}{1 + \lambda + \lambda^2 + \lambda^3 + \ldots + \lambda^{n-1}}$$

λ is constant between 0 and 1 that define how quickly the weights on older data should decrease. A high value of λ (close to 1) will cause older data to matter relatively more in the calculation. A low value of λ will mean the most recent data matters more, making the moving average more volatile.

PANDAS
Pandas provides built-in **exponentially weighted moving window** functions with the `.ewm` method. Consider using `.ewm().mean()`, and be sure to properly specify the `alpha` parameter (hint: it is related to, but not equal to λ). Note that `.ewm().std()` and `.ewm().var()` implement `ewmvar(x) = ewma(x**2) - ewma(x)**2`, which is slightly different than what you'll want to implement for this problem.

Forecasting Volatility

The general thinking is that it is easier to predict volatility than price.

ARCH: AUTOREGRESSIVE CONDITIONALLY HETEROSCEDASTIC

Autoregressive simply means that the current value relates to recent values. Heteroscedastic means the variable may have different magnitudes of variability (variance) at different times. Conditionally refers to a constraint placed on the heteroscedastic property to be conditionally dependent on the previous values/values of the variable.

Parameter α_0 can be thought of as the baseline variance. Subsequent α's as the weight for the contribution to the model of the previous time period's log returns:

ARCH (1) $$\mathrm{Var}(r_t|r_{t-1}) = \alpha_0 + \alpha_1 r_{t-1}^2$$

ARCH (m) $$\mathrm{Var}(r_t|r_{t-1}, r_{t-2}, \ldots, r_{t-m}) = \alpha_0 + \alpha_1 r_{t-1}^2 + \alpha_2 r_{t-2}^2 + \cdots + \alpha_m r_{t-m}^2$$

https://en.wikipedia.org/wiki/Autoregressive_conditional_heteroskedasticity

In econometrics, the **autoregressive conditional heteroskedasticity** (**ARCH**) model is a statistical model for time series data that describes the variance of the current error term or innovation as a function of the actual sizes of the previous time periods' error terms;[1] often the variance is related to the squares of the previous innovations. The ARCH model is appropriate when the error variance in a time series follows an autoregressive (AR) model; if an autoregressive moving average model (ARMA) model is assumed for the error variance, the model is a **generalized autoregressive conditional heteroskedasticity** (**GARCH**) model.[2] For forecasting, combining ARIMA and ARCH models could be considered. For instance, a hybrid ARIMA-ARCH model was examined for shipping freight rate forecast.[3]

ARCH models are commonly employed in modeling financial time series that exhibit time-varying volatility and volatility clustering, i.e. periods of swings interspersed with periods of relative calm. ARCH-type models are sometimes considered to be in the family of stochastic volatility models, although this is strictly incorrect since at time *t* the volatility is completely pre-determined (deterministic) given previous values.

GARCH: GENERAL AUTOREGRESSIVE CONDITIONALLY HETEROSCEDASTIC

If an autoregressive moving average model (ARMA) model is assumed for the error variance, the model is a generalized autoregressive conditional heteroskedasticity (GARCH) model.

GARCH(m,n) $$\begin{aligned}\sigma_t^2 &= Var(y_t|y_{t-1}, \ldots y_{t-m}, \sigma_{t-1}^2, \ldots \sigma_{t-n}^2) \\ &= \alpha_0 + \alpha_1 y_{t-1}^2 + \alpha_2 y_{t-2}^2 + \ldots + \alpha_m y_{t-m}^2 \\ &+ \beta_1 \sigma_{t-1}^2 + \beta_2 \sigma_{t-2}^2 + \ldots + \beta_n \sigma_{t-n}^2\end{aligned}$$

This means GARCH looks at previous returns as well as previously modeled variance by adding in the modeled variance terms σ^2:

where: m = # of log return terms; n = # of variance terms

Generally, when testing for heteroskedasticity in econometric models, the best test is the White test. However, when dealing with time series data, this means to test for ARCH and GARCH errors.

Exponentially weighted moving average (EWMA) is an alternative model in a separate class of exponential smoothing models. As an alternative to GARCH modelling it has some attractive properties such as a greater weight upon more recent observations, but also drawbacks such as an arbitrary decay factor that introduces subjectivity into the estimation.

USES
Can assess whether some strategies are correlated to changes in volatility or work better in higher of lower volatility environments.

Using Volatility for Equity Trading

USES OF VOLATILITY

- Limit risk — filter universe to exclude high volatility stocks
- Mean reversion - low vol stock that has a sudden price and volatility spike
- Low volatility stocks actually outperform LT, which is counter-intuitive to higher risk/higher return concept
  - Tortoise and the hare phenomenon - ETFs exist for low vol stocks
- Normalize signals by standardizing by risk, ie: σ
- Determine position size in a given strategy:

$$\frac{R}{\sigma \times M \times \text{LastClose}} = \text{PositionSize}$$

- R = \$ amount the trader is willing to lose if an M-sigma event occurs against his or her position
- σ = annualized volatility of the security or strategy in question
- M = trader-defined integer
- LastClose = last closing price of the security
- PositionSize = the number of shares to trade

Lesson 15: Pairs Trading and Mean Reversion

Mean Reversion

DRIFT AND VOLATILITY

Drift: the LT average volatility of a stock (first term)
Volatility term: randomness (second term)

$$dp_t = p_t\mu dt + p_t\sigma\epsilon\sqrt{dt}$$

Drift and Volatility Model (optional)

The drift and volatility model is also called a Brownian Motion model, and is a type of stochastic volatility model. First, let's discuss how this relates to the finance industry. Stochastic volatility models are fundamental building blocks for estimating the price of options (calls, puts, swaps) and also bonds. Before creating a model of an option (like a call option, for instance), we first want a model for the movement of its underlying asset (the stock price itself). The movement of the stock price is what the drift and volatility model (brownian motion model) attempts to describe.

The word Brownian Motion refers to the movements of molecules suspended in fluid, since this model was first used in physics and later adapted for finance. So it helps to imagine the stock price as a small particle, drifting through a glass of water, while it's being bumped around by other particles and molecules. The word "stochastic" is another word for "random". Stochastic volatility models attempt to represent the movement of a stock price when the volatility of its movements is random. Stochastic volatility models were used to improve upon the work of Black, Scholes and Merton, who came up with the first formula for pricing options.

Now let's revisit the drift and volatility model and describe what it means.

The left is referring to the differential of the stock price at time t. This type of equation is called a differential equation, since it describes the change over time of some process, rather than the specific state (stock price) of that series.

The term 2nd is the drift term. First, notice that it depends on the value of the stock price at time t. This means that if we compare the movements of two stocks, one that's priced at $2 per share, and another that's priced at $1000 per share, the series with the larger price per share is expected to drift (change) more in absolute dollar amounts compared to the other stock. The μ term is the expected return of the stock (think average return). Think of the expected return as the expected percent change over a period of time. We usually estimate the expected future return based on historical returns. So if a stock is expected to have a larger percent change per day compared to another, we'd also expect it to drift more (change more) compared to the other stock. This term also includes *dt*, which is the change in time (how much time has passed). If we watched a stock over a period of day versus over one month, we would expect it to drift more over a month, as more time has passed.

Now let's look at the volatility term. Think of this as the random, bouncy part of the stock movement. This term includes the stock price. It also includes the standard deviation of the stock σ_t, which is a function of time. This is why this model is a type of stochastic volatility model, because it allows for a non-constant volatility that varies over time. If a stock series has higher volatility, this will result in a larger overall movement in stock price (a higher dp_t). The ϵ is a white noise term, which means it's a random number with a mean of zero and standard deviation of one. The white noise accounts for movements in the stock price that are not accounted for by the model. Finally, there's the square root of the change in time. Note that the product $\epsilon\sqrt{(dt)}$ is usually written as dW_t and named a Wiener process.

Back to Mean Reversion

Okay, stepping back a bit to relate this to mean reversion. The drift and volatility model is a way to describe phenomena that we observe in real life, such as stock prices. The model assumes that there is a constant drift term with some added randomness, so we can expect that a series will bounce around, but still revert back to its long-term mean.

Pairs Trading

SPREAD AND HEDGE RATIO

Hedge ratio can simply be calculated using a price ratio: ie stock B price / stock A price.

Better is using regression: B = βA + α, which uses more price history

The spread = B_{actual_price} - $B_{estimate}$; where $B_{estimate}$ is the above regression hedge ratio (basically β)

> Note that with pairs trading, we analyze the original stock price series, and do not convert them to returns or log returns. We'll get into the details shortly, but let's just look at an example. Let's say $stock_A$ is \$2 per share, and $stock_B$ is \$3 per share. If we figured out that we can trade these pairs together, we may go long $stock_A$ and short $stock_B$. But how much do we long $stock_A$ and short $stock_B$? What if we long 3 shares of $stock_A$ and short 2 shares of $stock_B$? This is nice, because $shares_A \times price_A - shares_B \times price_B$ gives us $3 \times \$2 - 2 \times \$3$, or zero. Doing pairs trading analysis with the stock price series instead of returns lets us decide how many shares of each stock to long or short, since our goal will be to have the same dollar amount in our long position as in our short position.

> We can perform a regression where $stock_2$ is the dependent variable, and $stock_1$ is the independent variable (it doesn't matter which you choose to be x or y). Then the regression coefficient, which is our hedge ratio, is effectively $\frac{w_1}{w_2}$. You can see how multiplying $stock_1$ by $\frac{w_1}{w_2}$ is similar to multiply $stock_1$ by w_1 and $stock_2$ with w_2; in either case, we're weighting each stock so that their linear combination produces a stationary series.

COINTEGRATION

Stock prices are *integrated* of order 1: log price series: I(1) <== notation is 'capital i' (1)

- If we take the time difference of order 1, we get a stationary series (ie: of order I(0)

Cointegration

A way to think about whether two stocks' time series are cointegrated is to see if some linear combination of their time series forms a stationary series. In other words, let's say $stock_1$ and $stock_2$ are non-stationary, but $w_1 * stock_1 + w_2 * stock_2$ is a stationary series.

More generally, given y_t ~ I(1) and x_t ~ I(1):

1. Multiply x by the hedge ratio: $y_t \sim \alpha + \beta x_t$
2. Spread = $y_t - (\alpha + \beta x_t)$

If the spread is stationary, then it is integrated of order 0 (ie: I(0)) and x and y are cointegrated

- The hedge ratio is called the coefficient of cointegration

Note that cointegration is not the same as correlation. We are interested in the level of cointegration when seeking pairs of stocks to trade.

TWO-STEP ENGLE-GRANGER TEST

1. Get the hedge ratio from a linear regression
2. Calculate the spread and check is spread is stationary (meaning cointegrated)
   - Use Augmented Dickey-Fuller Test to see if stationary
     - If p-value <= 0.05, assume spread is stationary, therefore cointegrated

AUGMENTED DICKEY-FULLER TEST

To check if two series are cointegrated, we can use the Augmented Dickey Fuller (ADF) Test. First, let's get some intuition to see what the ADF test is doing. It's trying to determine if the linear combination of the two series, (which is also a time series) is stationary.

A series is stationary when its mean and covariance are constant, and also when the autocorrelation between one time period and another only depends on the time duration between them, and not the specific point in time of each observation.

If you could represent a series as an AR(1) model: $y_t = \beta y_{t-1} + \epsilon_t$

let's think about what happens if the β is greater than one. We can imagine putting in a value for y_{t-1} to get an estimate for y_t; then for the next day, we'll use that value as y_{t-1} to put into the model and estimate the new y_t. We'd end up having a series that trends in one direction, so its mean is not constant, and therefore it is not stationary.

Next, if we had a β equal to one, then $y_t = y_t - 1 + \epsilon_t$. We call this special case a random walk, and it means that the current price is equal to the previous price plus some white noise. Even though the mean of this series is constant, its covariance between one time period and another depends upon the point in time of the observations, so it is also not stationary.

Finally, if we had a β of less than one, then we notice that y_t depends upon less than 100% of the value of its previous value y_t - 1, with some added random noise ϵ_t . The series doesn't trend in a particular direction. Its variance is also constant, and its covariance between any two data points doesn't depend on the point in time of the data point. You can think of the series like a bouncing rubber ball that's being tapped lightly by random raindrops. Without the rain, the bouncing ball would have smaller and smaller bounces, and eventually stop bouncing. With random raindrops falling on the ball, some raindrops would make the ball bounce more, others would make the ball bounce less. So overall, the ball maintains a constant bounce height over time.

So conceptually, the Augmented Dickey Fuller Test is a hypothesis test for which the null hypothesis is that a series is a random walk (its β is equal to one), and so the null hypothesis assumes that the series is not stationary. The alternate hypothesis is that β is less than one, and therefore it's a stationary series. So if the ADF produces a p-value of 0.05 or less, we can say with a 95% confidence level that the series is stationary.

Clustering

Can use this ML model to find candidate stock pairs (rather than just using obvious sector segmentation which is too common / popular). The inputs to the model would be the stock time series themselves. After identifying pairs, then still want to test for cointegration.

Trading Pairs of Stocks

Go long or short the spread. Short when the spread widens (meaning short the stock that has gone up more and buy the laggard) and vice versa for going long the spread.

Use the z-score to define how standard deviations the spread is from its mean.

Train the model on a training set, do intermediate checks using the validation set (parameter tuning etc.), then check against test set (do not tune using test set as you introduce a lookahead bias leak).

VARIATIONS OF PAIRS TRADING OR MEAN REVERSION TRADING

Note that it's also possible to extend pairs trading to more than two stocks. We can identify multiple pairs and include these pairs in the same portfolio. We can also analyze stocks that are in the same industry. If we grouped the stocks within the same industry into a virtual portfolio and calculated the return of that industry, this portfolio return would represent the general expected movement of all stocks within the industry. Then, for each individual stock series, we can calculate the spread between its return and the portfolio return. We can assume that stocks within the same industry may revert towards the industry average. So when the spread between the single stock and the industry changes significantly, we can use that as a signal to buy or sell.

COINTEGRATION WITH 2 OR MORE STOCKS

Generalizing the 2-stock pairs trading method

We can extend cointegration from two stocks to three stocks using a method called the Johansen test. First let's see an example of how this works with two stocks.
The Johansen test gives us coefficients that we can multiply to each of the two stock series, so that a linear combination produces a number, and we can use it the same way we used the spread in the prior pairs trading method.

$w_1 * stock_1 + w_2 * stock_2 = spread$

In other words, if the first stock series moves up significantly relative to the second stock, we can see this by an increase in the "spread" beyond its historical average. We will assume that the spread will revert down towards its historical average, so we'll short the first stock that is relatively high, and long the second stock that is relatively low.

So far, this looks pretty much like what you did before, except instead of computing a hedge ratio to multiply to one stock, the Johansen test gives you one coefficient to multiply to each of the two stock series.

Extending to 3 stocks (optional)

Now let's extend this concept to three stocks. If we analyze three stock series with the Johansen, we can determine whether all three stocks together have a cointegrated relationship, and that a linear combination of all three form a stationary series. Note that for the purpose of cointegration trading we use the original price series, and do not convert them to log returns. The Johansen test also lets us decide whether only two series are needed to form a stationary series, but for now, let's assume that we find a trio of stocks that are cointegrated.

The Johansen gives us three coefficients, one for each stock series. We take the linear combination to get a spread.

$w_1 * stock_1 + w_2 * stock_2 + w_3 * stock_3 = spread$

We get the historical average of the spread. Then we check if the spread deviates significantly from that average. For example, let's say the spread increases significantly. So we check whether each of the three individual series moved up or down significantly to result in the change in spread. We short the series that are relatively high, and long the series that are relatively low. To determine how much to long or short, we again use the weights that are given by the Johansen

test (w_1, w_2, w_3).

For example, let's say the spread has gotten larger. Let's also pretend that w_1 is 0.5, w_2 is 0.3, and

w_3 is -0.1. Notice that the weights do not need to sum to 1. We'll long or short the number of shares for each stock in these proportions. So for instance, if we traded 5 shares of $stock_1$, we'll trade 3 shares of $stock_2$, and one share of $stock_3$.

If we notice that $stock_1$ is higher than normal, $stock_2$ is lower than normal, and $stock_3$ is lower than normal, then let's see whether we long or short a stock, and by how much.

Since $stock_1$ is higher than usual (relative to the others), we short 5 shares of $stock_1$ because we expect it should revert by decreasing relative to the others.

Since $stock_2$ is lower than normal, we long it by 3 shares, because we expect it to revert by increasing relative to the others.

Since $stock_3$ is lower than normal, so we also long it by 1 share but notice that w_3 is a negative number (-0.1). Whenever we see a negative weight, it means we change a buy to a sell, or change a sell to a buy. So we long a -1 shares, which is actually shorting 1 share.

DETAILS OF THE JOHANSEN TEST (VERY OPTIONAL)
So you may be wondering how we get these coefficients, and also how we check whether three stocks have a cointegrated relationship. For a closer look, let's introduce a bit of math. Recall from the lesson on time series, that a vector autoregression attempts to describe a stock's current value based on not only its prior values, but also the prior values of other stocks. Let's use two stocks as an example: Note, I'm using the variable names "IBM" and "GE" to refer to the log returns of these stocks. The \mu refers to a historical average for each stock's time series. The "e" refers to an error term for each stock.

*** there is a lot more here, but tired of formatting formulas!!

MODULE 3: SMART BETA AND PORTFOLIO OPTIMIZATION

Lesson 17: Stocks, Indices, Funds

Tracking Error

Tracking Error measures how the returns of a portfolio differ from the returns of its benchmark. To operationalize the definition of tracking error, we first take the portfolio's daily returns minus the benchmark's daily returns. This daily difference is referred to as the excess return and also the active return. Next, we take the sample standard deviation of the active return. Finally, we annualize the daily sample standard deviation by multiplying by the square root of 252, since there are 252 trading days in a year.

The formula is:

$$ExcessReturn_{portfolio} = return_{portfolio} - return_{benchmark}$$
$$DailyTrackingError = SampleStandardDeviation(ExcessReturn_{portfolio})$$
$$AnnualizedTrackingError = \sqrt[2]{252} * DailyTrackingError$$

In Summary:

$$TE = \sqrt[2]{252} * SampleStdev(return_{portfolio} - return_{benchmark})$$

Expense Ratios

AUM: Assets Under Management Gross Expense Ratio: Expenses / AUM Net Expense Ratio: (Expenses - Discounts) / AUM

Lesson 18: ETF's

Create / Redeem Process

Authorized Participants (APs) and ETF Sponsors partner together to make the ETF system work. We can think of APs as the intermediaries between investors and the ETF Sponsor. Unlike mutual funds or hedge funds, ETF Sponsors don't take cash to invest, nor do they deal directly with investors. ETF Sponsors take a portfolio of stocks instead of cash, and they trade with APs instead of with investors. ETF Sponsors and APs create ETF shares with the "create process".

The "create process" involves the following steps:

1. The Authorized Participant buys stocks and bundles them in the same proportions as defined by the ETF Sponsor.
2. The AP gives these stocks to the ETF Sponsor.
3. The ETF Sponsor creates ETF shares and gives these to the AP.
4. The AP sells the ETF shares to investors.

The redeem process involves the following steps

1. The AP buys ETF shares from investors in the stock market.
2. The AP trades these ETF shares with the ETF Sponsor in exchange for the original stocks.
3. The AP sells these stocks on the stock exchange.

ETF sponsors can charge more competitive (lower) fees in part because their transactions can be more tax efficient. For an ETF Sponsor, recall that when it enters a create or redeem process, stocks and ETF shares are being exchanged, and not cash. Also, the dollar value of these assets being exchanged are more or less equal.

Lesson 19: Portfolio Risk and Return

Portfolio Variance

We start with this:

$$\sigma_P^2 = \sum_i p(i)[r_P - E(r_P)]^2$$

Let's plug in what we know.

$$\sigma_P^2 = \sum_i p(i)[x_A r_A - x_A E(r_A) + x_B r_B - x_B E(r_B)]^2$$

$$= \sum_i p(i)[x_A(r_A - E(r_A)) + x_B(r_B - E(r_B))]^2$$

Then we square everything in the brackets:

$$= \sum_i p(i)[x_A^2(r_A - E(r_A))^2 + x_B^2(r_B - E(r_B))^2 + 2x_A x_B(r_A - E(r_A))(r_B - E(r_B))]$$

Whew, let's stop for a breather.

Mmmmk. So now, we do the same thing we did in the derivation of the portfolio mean. Instead of putting everything into one big sum, we break the big sum up into sub-sums, and pull out the weights, which aren't indexed by i.

$$= x_A^2 \sum_i p(i)(r_A - E(r_A))^2 + x_B^2 \sum_i (r_B - E(r_B))^2 + 2x_A x_B \sum_i (r_A - E(r_A))(r_B - E(r_B))$$

And now, if we look closely, we can see the result already. In the first two terms, the sums are just the individual asset variances. The third term is where the magic happens. That sum simply equals the covariance.

$$= x_A^2 \sigma_A^2 + x_B^2 \sigma_B^2 + 2x_A x_B \mathrm{Cov}(r_A, r_B)$$

Not quite...be sure you have framed the calculation appropriately as

$$\sigma_P^2 = \begin{bmatrix} x_A & x_B & x_C \end{bmatrix} \begin{bmatrix} \mathrm{Cov}(r_A, r_A) & \mathrm{Cov}(r_A, r_B) & \mathrm{Cov}(r_A, r_C) \\ \mathrm{Cov}(r_B, r_A) & \mathrm{Cov}(r_B, r_B) & \mathrm{Cov}(r_B, r_C) \\ \mathrm{Cov}(r_C, r_A) & \mathrm{Cov}(r_C, r_B) & \mathrm{Cov}(r_C, r_C) \end{bmatrix} \begin{bmatrix} x_A \\ x_B \\ x_C \end{bmatrix}$$

and go back and check your calculation.

Reducing Risk with Imperfectly Correlated Stocks

We just noted that when the correlation is less than one $(\rho < 1)$, the portfolio standard deviation is less than the weighted average of the individual standard deviations:
$\sigma_{p,\rho<1} < x_A\sigma_A + x_B\sigma_B$.
Let's walk through this together to see how this helps us as investors.

First, we notice that if the standard deviation of a portfolio is less than the standard deviation of another, then the variance of the first portfolio is also less than that of the second.
$\sigma_{p1} < \sigma_{p2} \Leftrightarrow \sigma_{p1}^2 < \sigma_{p2}^2$

So let's compare the variance of a portfolio where correlation is +1, and compare it to another portfolio where correlation is less than 1 (let's just say 0.9).

$\sigma_{p,\rho=1.0}^2 = x_A^2\sigma_A^2 + x_B^2\sigma_B^2 + 2x_Ax_B\sigma_A\sigma_B\rho_{rArB}$
where $\rho_{rArB} = 1$

Versus
$\sigma_{p,\rho=0.9} = x_A^2\sigma_A^2 + x_B^2\sigma_B^2 + 2x_Ax_B\sigma_A\sigma_B\rho_{rArB}$
where $\rho_{rArB} = 0.9$

If we cancel all of the identical terms in both equations, we can compare the third term in each:
$2x_Ax_B\sigma_A\sigma_B \times 1 > 2x_Ax_B\sigma_A\sigma_B \times 0.9$. Or more simply: 1 > 0.9

So we can show that the variance of the imperfectly correlated portfolio is less than the variance of the perfectly correlated one.
$\sigma_{p,\rho=1.0}^2 = (x_A\sigma_A + x_B\sigma_B)^2 = x_A^2\sigma_A^2 + x_B^2\sigma_B^2 + 2x_Ax_B\sigma_A\sigma_B \times 1$
$> x_A^2\sigma_A^2 + x_B^2\sigma_B^2 + 2x_Ax_B\sigma_A\sigma_B \times 0.9 = \sigma_{p,\rho=-0.9}^2$

In other words: $\sigma_{p,\rho=1.0}^2 > \sigma_{p,\rho=0.9}^2$
which implies that $\sigma_{p,\rho=1.0} > \sigma_{p,\rho=0.9}$

The nice benefit of putting two stocks into a portfolio is that, as long as they're not perfectly correlated, we'll end up with a portfolio whose risk is less than the the weighted sum of the individual risks. A key benefit of portfolio diversification is that it helps us to reduce risk!

Covariance Matrix and Quadratic Forms

COVARIANCE MATRIX

Let's take a moment to learn a compact way to represent the portfolio variance using matrices and vectors.

Remember that the portfolio variance we calculated for our two-stock portfolio was:

$$\sigma_P^2 = x_A^2\sigma_A^2 + x_B^2\sigma_B^2 + 2x_Ax_B\mathrm{Cov}(r_A, r_B).$$

But

$$\sigma_A^2 = \mathrm{Cov}(r_A, r_A),$$

so

$$\sigma_P^2 = x_A^2\mathrm{Cov}(r_A, r_A) + x_B^2\mathrm{Cov}(r_B, r_B) + 2x_Ax_B\mathrm{Cov}(r_A, r_B).$$

This expression now has a nice parallel structure. If we create a matrix called the covariance matrix,

$$\mathbf{P} = \begin{bmatrix} \mathrm{Cov}(r_A, r_A) & \mathrm{Cov}(r_A, r_B) \\ \mathrm{Cov}(r_B, r_A) & \mathrm{Cov}(r_B, r_B) \end{bmatrix},$$

and a vector of weights:

$$\mathbf{x} = \begin{bmatrix} x_A \\ x_B \end{bmatrix},$$

and do the following matrix multiplication:

$$\begin{bmatrix} x_A & x_B \end{bmatrix} \begin{bmatrix} \mathrm{Cov}(r_A, r_A) & \mathrm{Cov}(r_A, r_B) \\ \mathrm{Cov}(r_B, r_A) & \mathrm{Cov}(r_B, r_B) \end{bmatrix} \begin{bmatrix} x_A \\ x_B \end{bmatrix}$$

$$= \begin{bmatrix} x_A & x_B \end{bmatrix} \begin{bmatrix} x_A\mathrm{Cov}(r_A, r_A) + x_B\mathrm{Cov}(r_A, r_B) \\ x_A\mathrm{Cov}(r_B, r_A) + x_B\mathrm{Cov}(r_B, r_B) \end{bmatrix}$$

$$= x_Ax_A\mathrm{Cov}(r_A, r_A) + x_Ax_B\mathrm{Cov}(r_A, r_B) + x_Ax_B\mathrm{Cov}(r_A, r_B) + x_Bx_B\mathrm{Cov}(r_B, r_B)$$

$$= x_A^2\sigma_A^2 + x_B^2\sigma_B^2 + 2x_Ax_B\mathrm{Cov}(r_A, r_B),$$

we see that we recover the expression for σ_P^2.

So:

$$\sigma_P^2 = \mathbf{x}^{\mathrm{T}}\mathbf{P}\mathbf{x}$$

QUADRATIC FORMS

A polynomial where the sums of the exponents of the variables in each term equals 2 is called a quadratic form. An example:

$$4x^2 - 2xy + 3y^2$$

The portfolio variance is an example of a quadratic form (remember, x_A and x_B are the variables here):

$$\sigma_P^2 = x_A^2\sigma_A^2 + x_B^2\sigma_B^2 + 2x_Ax_B\mathrm{Cov}(r_A, r_B)$$

A quadratic form can always been written as $\mathbf{x}^{\mathrm{T}}\mathbf{P}\mathbf{x}$, where P is a symmetric matrix.

Calculate a Covariance Matrix

Remember how we defined the covariance matrix:

$$\mathbf{P} = \begin{bmatrix} \mathrm{Cov}(r_A, r_A) & \mathrm{Cov}(r_A, r_B) \\ \mathrm{Cov}(r_B, r_A) & \mathrm{Cov}(r_B, r_B) \end{bmatrix}.$$

And covariance is

$$\mathrm{Cov}(r_A, r_B) = \mathrm{E}[(r_A - \bar{r}_A)(r_B - \bar{r}_B)].$$

If r_A and r_B are discrete vectors of values, that is, they can take on the values (r_{Ai}, r_{Bi}) for $i = 1, \ldots, n$, with equal probabilities 1/n, then the covariance can be equivalently written,

$$= \frac{1}{n-1} \sum_{i=1}^{n} (r_{Ai} - \bar{r}_A)(r_{Bi} - \bar{r}_B).$$

We use $n - 1$ in the denominator of the constant for the same reason that we use $n - 1$ in the denominator of the constant out front in the sample standard deviation —because we have a sample, and we want to calculate an *unbiased* estimate of the population covariance.

But if $\bar{r}_A = \bar{r}_B = 0$, then the covariance equals

$$= \frac{1}{n-1} \sum_{i=1}^{n} r_{Ai} r_{Bi}.$$

In matrix notation, this equals

$$\frac{1}{n-1} \mathbf{r}_A^{\mathrm{T}} \mathbf{r}_B.$$

Therefore, if $\mathbf{r}$ is a matrix that contains the vectors $\mathbf{r}_A$ and $\mathbf{r}_B$ as its columns,

$$\mathbf{r} = \begin{bmatrix} \vdots & \vdots \\ \mathbf{r}_A & \mathbf{r}_B \\ \vdots & \vdots \end{bmatrix},$$

then

$$\mathbf{r}^{\mathrm{T}}\mathbf{r} = \begin{bmatrix} \cdots & \mathbf{r}_A & \cdots \\ \cdots & \mathbf{r}_B & \cdots \end{bmatrix} \begin{bmatrix} \vdots & \vdots \\ \mathbf{r}_A & \mathbf{r}_B \\ \vdots & \vdots \end{bmatrix} = \begin{bmatrix} \mathbf{r}_A^{\mathrm{T}}\mathbf{r}_A & \mathbf{r}_A^{\mathrm{T}}\mathbf{r}_B \\ \mathbf{r}_B^{\mathrm{T}}\mathbf{r}_A & \mathbf{r}_B^{\mathrm{T}}\mathbf{r}_B \end{bmatrix}.$$

So if each vector of observations in your data matrix has mean 0, you can calculate the covariance matrix as:

$$\frac{1}{n-1} \mathbf{r}^{\mathrm{T}}\mathbf{r}$$

Sharpe Ratio

The Sharpe ratio is the **ratio of reward to volatility**. It's a popular way to look at the performance of an asset relative to its risk.

$\text{Sharpe Ratio} = \frac{r_{\text{risky portfolio}} - r_{\text{risk free}}}{\sigma_{\text{excess return}}}$

The numerator of the Sharpe ratio is called the *excess return*, *differential return* as well as the *risk premium*. It's called "excess return" because this is the return in excess of the risk-free rate. It's also called the "risk premium", because this represents the premium that investors should be rewarded with for taking on risk.

The denominator is the volatility of the excess return.

How do you calculate this? The *risk premium* (which we'll denote with _D_) equals the portfolio return minus risk free rate over a period of time:

$D_t = r_{\text{portfolio}, t} - r_{\text{risk free}, t}$

Then, calculate the mean and standard deviation of D_t over the historical period from $t = 1$ to T:

$D_{average} = \frac{1}{t} \sum_{t=1}^{T} D_t \qquad \sigma_D = \sqrt{\frac{\sum_{t=1}^{T} (D_t - D_{average})^2}{T-1}}$

$\text{Sharpe Ratio} = \frac{D_{average}}{\sigma D}$

As we saw previously, the Sharpe Ratio is the slope of the *Capital Market Line*.

The Sharpe Ratio allows us to compare stocks of different returns, because the Sharpe ratio adjusts the returns by their level of risk.

[Note that if you do not see some fractions displaying as expected, please try to zoom in with your browser]

To annualize daily risk premium $(r_p - r_f)$, we add the daily return 252 times, or more simply multiply by 252. $D_{year} = 252 \times D_{day}$

To annualize the daily standard deviation, let's first annualize the daily variance. To annualize daily variance, we add σ_D^2 252 times, or more simply multiply it by 252.
$\sigma_{D,year}^2 = 252 \times \sigma_{D,day}^2$

The standard deviation is the square root of the variance, which is
$\sqrt[2]{252 \times \sigma_D^2}$.
or just $\sqrt[2]{252} \times \sigma_D^2$

In other words:
$\sigma_{D,year} = \sqrt[2]{252} \times \sigma_{D,day}$

If we combine the annualization factors of the numerator and denominator, this becomes:
$\frac{252}{\sqrt[2]{252}}$
which simplifies to $\sqrt[2]{252}$

So to convert the Sharpe ratio from daily to annual, we multiply by $\sqrt[2]{252}$.

Therefore:
$\text{Sharpe Ratio}_{year} = \sqrt{252}\ \text{Sharpe Ratio}_{day}$

Other Risk Measures

SEMI-DEVIATION

If you were given two stocks, one that continued to increase by 10% every day, and one that decreased by 10% every day, would you intuitively think that one stock was more risky than the other? Standard deviation measures of risk would give these two stocks the same level of risk, but you might think that investors are more worried about down-side risk (when stocks decline), rather than upside risk. The the motivation for semi-deviation measure of risk is to measure downside risk specifically, rather than any kind of volatility.

Semi-deviation is calculated in a similar way as standard deviation, except it only includes observations that are less than the mean.

$$SemiDeviation = \sum_{t=1}^{n} (\mu - r_i)^2 \times I_{r_i < \mu}$$

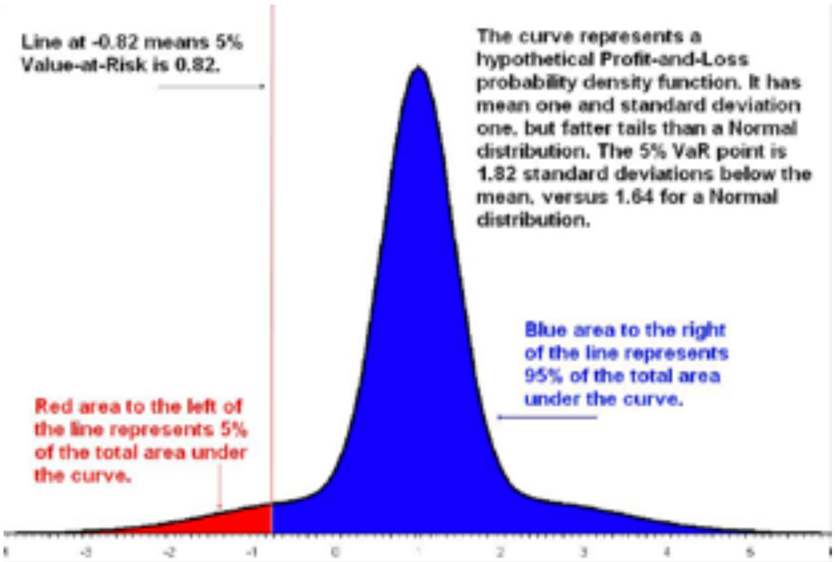
where $I_{r_i<\mu}$ equals 1 when $r_i < \mu$, and 0 otherwise.

VALUE-AT-RISK (VAR)

VaR, or value-at-risk is a portfolio risk measure. Risk managers at investment firms and investment banks calculate VaR to estimate how much money a portfolio manager's fund may potentially lose over a certain time period. Corporations also estimate their own VaR to decide how much cash they should hold to avoid bankruptcy during a worst case scenario.

VaR is defined as the maximum dollar amount expected to be lost over a given time horizon at a predefined confidence level. For example, if the 95% one month VaR is $1 million, there is 95% confidence that the portfolio will not lose more than $1 million next month. Another way to describe the VaR is that there is a 5% chance of losing $1 million or more next month. The methods for calculating VaR are beyond the scope of this lesson, but if you ever become a risk manager, or ever work with a risk manager, you'll probably see Value-at-Risk quite a bit.

For a visual representation of VaR, we can look at a data distribution that represents the rate of return of a stock. If we color in the area in the left tail that represents 5% of the distribution, the rate of return represented by that point on the horizontal axis is the rate of return that may occur in the 5% worst case scenario. To convert that to a VaR, we multiply that rate of return by the amount of capital that is exposed to risk. For a portfolio, it would be the amount of dollars invested in that particular stock.



As an example, let's say we invested $10 million in a stock. We estimate the mean and standard deviation of the stock's returns and model it with a distribution function (it might be a normal distribution, but there are other models). Then we find the rate of return that defines 5% of the distribution to its left, in the left tail. Let's say that rate of return is -20%. We multiply that rate of return by the amount that we're exposed to, which is -0.20 x $10m = -$2m. So the VaR on any given day is $2 million. In other words, we may plan some hedging strategies or hold enough cash to help us handle the possibility of losing $2 million on stock A on any given day. For more detail, and an image of the distribution, check out Wikipedia's page on **Value-at-Risk**

Capital Asset Pricing Model

CAPM
In addition to the Capital Market Line, we will further introduce another important concept: the Capital Asset Pricing Model which is also called CAPM.

The CAPM is a model that describes the relationship between systematic risk and expected return for assets. The CAPM assumes that the excess return of a stock is determined by the market return and the stock's relationship with the market's movement. It is the foundation of the more advanced multi-factor models used by portfolio managers for portfolio construction.

To recap: the systematic risk, or market risk, is undiversifiable risk that's inherent to the entire market. In contrast, the idiosyncratic risk is the asset-specific risk.

Now the CAPM. For a stock, the return of stock i equals the return of the risk free asset plus β times the difference between the market return and the risk free return. β equals the covariance of stock i and the market divided by the variance of the market.

$$r_i - r_f = \beta \times (r_m - r_f)$$

r_i = stock return
r_f = risk free rate
r_m = market return
$\beta_i = \frac{cov(r_i, r_m)}{\sigma_m^2}$

β describes which direction and by how much a stock or portfolio moves relative to the market. For example, if a stock has a β of 1, this indicates that if the market's excess return is 5%, the stock's excess return would also be 5%. If a stock has a β of 1.1, this indicates that if the market's excess return is 5%, the stock's excess return would be 1.1 × 5%, or 5.5%.

COMPENSATING INVESTORS FOR RISK

Generally speaking, investors need to be compensated in two ways: time value of money and risk. The time value of money is represented by the risk free return. This is the compensation to investors for putting down investments over a period of time. β times $r_m - r_f$ represents the risk exposure to the market. It is the additional excess return the investor would require for taking on the given market exposure, β. $r_m - r_f$ is the risk premium, and β reflects the exposure of an asset to the overall market risk.

When the β_i for stock i equals 1, stock i moves up and down with the same magnitude as the market. When β_i is greater than 1, stock i moves up and down more than the market. In contrast, when β_i is less than 1, stock i moves up and down less than the market.

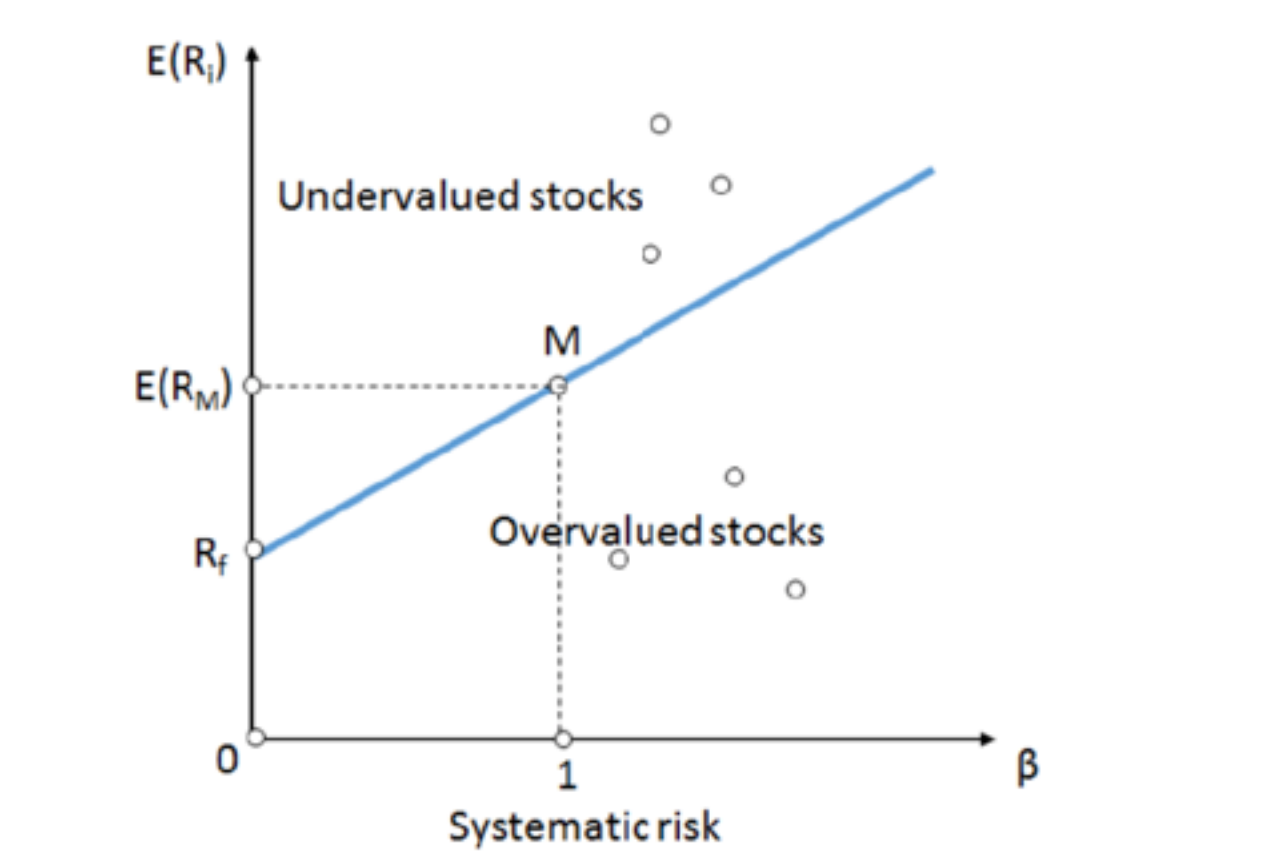
Let's look at a simple example. If the risk free return is 2%, β_i of stock i equals 1.2 and the market return is 10%. The return of stock i equals 2% + 1.2 × (10% - 2%) = 11.6.

r_f = 2%
β_i = 1.2
r_m = 10%
r_i = 2% + 1.2 × (10% - 2%) = 11.6%

SECURITY MARKET LINE
The Security Market Line is the graphical representation of CAPM and it represents the relation between the risk and return of stocks. Please note that it is different from the capital market line. The y-axis is expected returns but the x-axis is beta. (You may recall that for the capital market line that we learned earlier, the x-axis was standard deviation of a portfolio.) As beta increases, the level of risk increases. Hence, the investors demand higher returns to compensate risk.

The Security Market Line is commonly used to evaluate if a stock should be included in a portfolio. At time points when the stock is above the security market line, it is considered "undervalued" because the stock offers a greater return against its systematic risk. In contrast, when the stock is below the line, it is considered overvalued because the expected return does not overcome the inherent risk.

The SML is also used to compare similar securities with approximately similar returns or similar risks.



Lesson 20: Portfolio Optimization

What is Optimization?

Second-order condition

In the previous problem, we cheated a little. We knew the shape of the function, and we knew its orientation from our plot, so when we found the point where the derivative equaled 0, we knew we had found the minimum. However, in general, points where the derivative equals 0 could be minima, maxima, or saddle points. To distinguish between these cases, we need to check the function's curvature around the point in question. We do this using the second derivative of the function. For a function of one variable, the rule is:

- If $\frac{d^2y}{dx^2}(x_0) < 0$ then f has a local maximum at x_0.
- If $\frac{d^2y}{dx^2}(x_0) > 0$ then f has a local minimum at x_0.
- If $\frac{d^2y}{dx^2}(x_0) = 0$, the test is inconclusive.

So in the case of the function above, we have

$$\frac{dy}{dx} = 2(x-1)$$

so,

$$\frac{d^2y}{dx^2} = 2$$

The second derivative is positive for all x, which means the function's slope is increasing everywhere, hence the function's curvature is upward everywhere. So we can be confident that the point we found is a minimum.

For a function of two variables, the rule must change a bit. We construct the matrix of second-order partial derivatives:

$$H(x,y) = \begin{pmatrix} f_{xx}(x,y) & f_{xy}(x,y) \\ f_{yx}(x,y) & f_{yy}(x,y) \end{pmatrix}$$

This is called the *Hessian* matrix $H(x,y)$.

Recall that the *determinant* of a matrix,

$$\det\begin{pmatrix} a & b \\ c & d \end{pmatrix} = ad - bc.$$

If the first-order partial derivatives are 0 at the point (a,b), i.e. $f_x(a,b) = f_y(a,b) = 0$, then, we apply the following rule:

- If $\det(H)(a,b) > 0$ and $f_{xx}(a,b) > 0$ then (a,b) is a local minimum of f.
- If $\det(H)(a,b) > 0$ and $f_{xx}(a,b) < 0$ then (a,b) is a local maximum of f.
- If $\det(H)(a,b) < 0$ then (a,b) is a saddle point of f.
- If $\det(H)(a,b) = 0$ then the second derivative test is inconclusive, and the point (a,b) could be any of a minimum, maximum or saddle point.

2-Asset Portfolio Optimization

Derivation of Optimal Weights on a Two-Asset Portfolio

So how do we set up the portfolio optimization problem? In general, we know that we want high returns and low variance of returns, and that the weights on each asset in our portfolio should sum to 1.

Let's again consider a portfolio a portfolio with 2 assets in it, Stock A and Stock B. We want to solve for the weight on each asset, x_A and x_B.

Our objective function for this problem is the expression for the portfolio variance:

$$(1)\ \sigma_P^2 = x_A^2\sigma_A^2 + x_B^2\sigma_B^2 + 2x_Ax_B\sigma_A\sigma_B\rho_{r_Ar_B}.$$

This is what we will seek to *minimize*.

Our only constraint is:

$$(2)\ x_A + x_B = 1$$

It turns out that this is a problem we can solve analytically. If we substitute (2) into (1), we will get a function of a single variable:

$$\sigma_P^2 = x_A^2\sigma_A^2 + (1 - x_A)^2\sigma_B^2 + 2x_A(1 - x_A)\sigma_A\sigma_B\rho_{r_Ar_B}$$

Now, let's take a breather and look at a plot of σ_P^2 as a function of x_A, to get a sense of what the function looks like. Let's assume some values for the variables σ_A, σ_B, and $\rho_{r_Ar_B}$.

If, $\sigma_A = 0.1$, $\sigma_B = 0.05$, and $\rho_{r_Ar_B} = 0.25$, then the plot looks like this:

Cool, it's a parabola. So now we see that we are looking for the bottom of that parabola again.

So, since this is a problem in only one variable, x_A, let's take the derivative with respect to x_A, and solve for the value of x_A that makes the resulting expression equal 0:

$$\frac{\mathrm{d}(\sigma_P^2)}{\mathrm{d}x_A} = 0 = 2x_A\sigma_A^2 - 2\sigma_B^2(1 - x_A) + 2\sigma_A\sigma_B\rho_{r_Ar_B}[-x_A + (1 - x_A)]$$

$$0 = 2x_A\sigma_A^2 + 2\sigma_B^2x_A - 2\sigma_B^2 + 2\sigma_A\sigma_B\rho_{r_Ar_B}[1 - 2x_A]$$

$$x_A[2\sigma_A^2 + 2\sigma_B^2 - 4\sigma_A\sigma_B\rho_{r_Ar_B}] - 2\sigma_B^2 + 2\sigma_A\sigma_B\rho_{r_Ar_B} = 0$$

$$x_A[\sigma_A^2 + \sigma_B^2 - 2\sigma_A\sigma_B\rho_{r_Ar_B}] = \sigma_B^2 - \sigma_A\sigma_B\rho_{r_Ar_B}$$

$$x_A = \frac{\sigma_B^2 - \sigma_A\sigma_B\rho_{r_Ar_B}}{\sigma_A^2 + \sigma_B^2 - 2\sigma_A\sigma_B\rho_{r_Ar_B}}$$

From the constraint condition, (2), we have:

$$x_B = 1 - x_A$$

Now we know the portfolio weights. You can see that they are only dependent on the standard deviations of Stock A and B, and their covariance. If we wanted to know the expected portfolio mean, we only have to remember that it is the weighted sum of the individual portfolio means:

$$\mu_P = \mu_Ax_A + \mu_Bx_B$$

To be sure our solution is a *minimum*, we should check the second-order condition. We will leave this as an exercise to you.

Formulating Portfolio Optimization Problems

So far, we've discussed one way to formulate a portfolio optimization problem. We learned to set the portfolio variance as the objective function, while imposing the constraint that the portfolio weights should sum to 1. However, in practice you may *frame* the problem a little differently. Let's talk about some of the different ways to *set up* a portfolio optimization problem.

Common Constraints

There are several common constraints that show up in these problems. Earlier, we were allowing our portfolio weights to be negative or positive, as long as they summed to 1. If a weight turned out to be negative, we would consider the absolute value of that number to be the size of the *short* position to take on that asset. If your strategy does not allow you to take short positions, your portfolio weights will all need to be positive numbers. In order to enforce this in the optimization problem, you would add the constraint that every x_i in the $\mathbf{x}$ vector is *positive*.

no short selling: $0 \leq x_i \leq 1, \qquad i = 1, 2, \ldots, n$

You may choose to impose constraints that would limit your portfolio allocations in individual sectors, such as technology or energy. You could do this by limiting the sum of weights for assets in each sector.

sector limits:

$$x_{\text{biotech1}} + x_{\text{biotech2}} + x_{\text{biotech3}} \leq M, \qquad M = \text{percent of portfolio to invest in bi}$$

If your optimization objective seeks to minimize portfolio variance, you might also incorporate into the problem a goal for the total portfolio return. You can do this by adding a constraint on the portfolio return.

constraint on portfolio return:

$$\mathbf{x}^T\mu \geq r_{\min}, \qquad r_{\min} = \text{minimum acceptable portfolio return}$$

Maximizing Portfolio Return

We can also flip the problem around by maximizing returns instead of minimizing variance. Instead of minimizing variance, it often makes sense to impose a constraint on the variance in order to manage risk. Then you could maximize mean returns, which is equivalent to minimizing the negative mean returns. This makes sense when your employer has told you, "I want the best return possible, but you must limit your losses to p percent!"

objective: $\text{minimize}: \quad -\mathbf{x}^T\mu$

constraint: $\mathbf{x}^T\mathbf{P}\mathbf{x} \leq p, \quad p = \text{maximum permissible portfolio variance}$

Maximizing Portfolio Return *And* Minimizing Portfolio Variance

Indeed, you could also create an objective function that both maximizes returns and minimizes variance, and controls the tradeoff between the two goals with a parameter, b. In this case, you have two terms in your objective function, one representing the portfolio mean, and one representing the portfolio variance, and the variance term is multiplied by b.

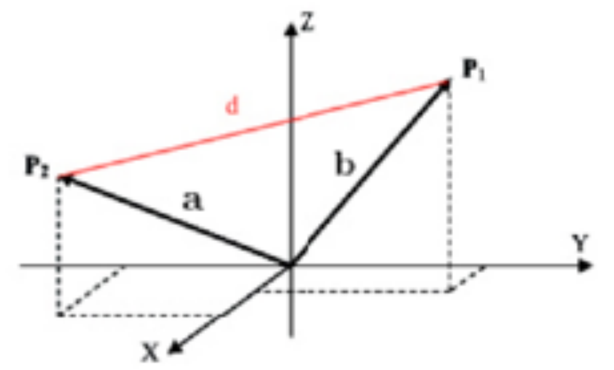
How does one determine the parameter b? Well, it's very dependent on the individual and the situation, and depends on the level of risk aversion appropriate. It basically represents how much percent return you are willing to give up for each unit of variance you take on.

objective: $\quad \text{minimize}: \quad -\mathbf{x}^{\mathrm{T}}\mu + b\mathbf{x}^{\mathrm{T}}\mathbf{P}\mathbf{x}, \quad b = \text{tradeoff parameter}$

A Math Note: the L2-Norm

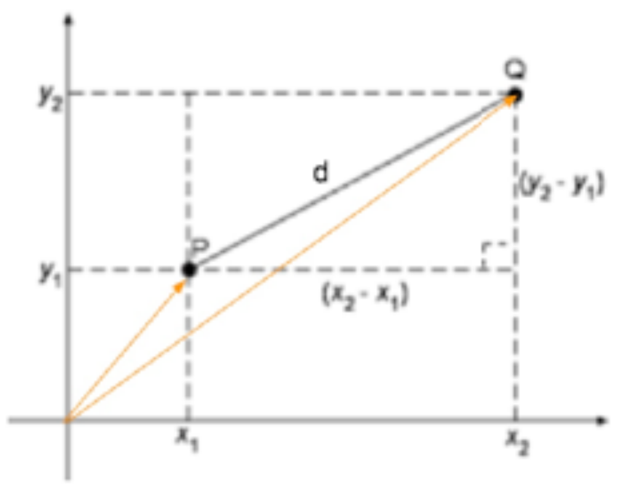
There's another way to formulate an optimization objective that relies on a new piece of notation, so I'll just take a moment to explain that now. Say we just want to minimize the difference between two quantities. Then we need a measure of the difference, but generalized into many dimensions. For portfolio optimization problems, each dimension is an asset in the portfolio. When we want to measure the distance between two vectors, we use something called the Euclidean norm or L2-norm. This is just the square root of the squared differences of each of the vectors' components. We write it with double bars and a 2 subscript.

$$d = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2 + (a_z - b_z)^2} = \|\mathbf{a} - \mathbf{b}\|_2$$



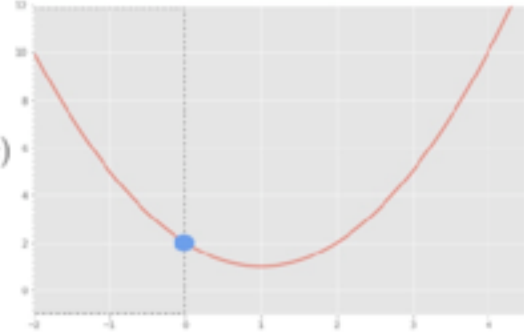
Note that this reduces to the familiar Pythagorean theorem in 2 dimensions.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$



Minimizing Distance to a Set of Target Weights

Back to portfolio optimization! One way to formulate an optimization problem is to use the L2 norm and minimize the difference between your weights and a set of predefined target portfolio weights $\mathbf{x}^*$. the weights as close as possible to the set of target weights constraints. As an example, these target weights might be v $f(x)$ proportional to future returns for each asset, in other word



objective:

$$\text{minimize}: \quad \|\mathbf{x} - \mathbf{x}^*\|_2, \quad \mathbf{x}^* = \text{a set of target}$$

Tracking an Index

What if you want to minimize portfolio variance, but have the portfolio track an index at the same time? In this case, you would want terms in your objective function representing both portfolio variance and the relationship between your portfolio weights and the index weights, $\mathbf{q}$. There are a few ways to set this up, but one intuitive way is to simply minimize the difference between your portfolio weights and the weights on the assets in the index, and minimize portfolio variance at the same time. The tradeoff between these goals would be determined by a parameter, λ.

objective:

$$\text{minimize}: \quad \mathbf{x}^T\mathbf{P}\mathbf{x} + \lambda\,\|\mathbf{x} - \mathbf{q}\|_2, \quad \mathbf{q} = \text{a set of index weights}, \quad \lambda = \text{a tradeoff parameter}$$

Leverage Ratio

$$\frac{\sum_{i=1}^{N} |x_i|}{Notional}$$

Example portfolio: \$1m Notional	Leverage ratio: sum of positions/ Notional
\$1m long positions	$\$1m/\$1m = 1$
Borrow to have \$2m long positions	$\$2m/\$1m = 2$
\$1m long, \$1m short positions	$\$2m/\$1m = 2$
\$2m long, \$2m short positions	$\$4m/\$1m = 4$

What is cvxpy?

`cvxpy` is a Python package for solving convex optimization problems. It allows you to express the problem in a human-readable way, calls a solver, and unpacks the results.

Solve the simple example we referred to in the lectures, optimizing the objective function $(x-1)^2 + 1$ subject to the constraint $x <= 0$:

```
import cvxpy as cvx
import numpy as np

x = cvx.Variable(1)
objective = cvx.Minimize((x - 1)**2 + 1)
constraints = [x <= 0]
problem = cvx.Problem(objective, constraints)
result = problem.solve()
print('Optimal value of x: {:.6f}'.format(x.value[0]))
print('Optimal value of the objective: {:.6f}'.format(problem.value))
```

```
Optimal value of x: -0.000000
Optimal value of the objective: 2.000000
```

HOW TO USE `CVXPY`

Import: First, you need to import the package:`import cvxpy as cvx`

Steps: Optimization problems involve finding the values of a *variable* that minimize an *objective function* under a set of *constraints* on the range of possible values the variable can take. So we need to use `cvxpy` to declare the *variable*, *objective function* and *constraints*, and then solve the problem

Optimization variable: Use `cvx.Variable()` to declare an optimization variable. For portfolio optimization, this will be $\mathbf{x}$, the vector of weights on the assets. Use the argument to declare the size of the variable; e.g. `x = cvx.Variable(2)` declares that $\mathbf{x}$ is a vector of length 2. In general, variables can be scalars, vectors, or matrices.

Objective function: Use `cvx.Minimize()` to declare the objective function. For example, if the objective function is $(\mathbf{x} - \mathbf{y})^2$, you would declare it to be: `objective = cvx.Minimize((x - y)**2)`.

Constraints: You must specify the problem constraints with a list of expressions. For example, if the constraints are $\mathbf{x} + \mathbf{y} = 1$ and $\mathbf{x} - \mathbf{y} \geq 1$ you would create the list: `constraints = [x + y == 1, x - y >= 1]`. Equality and inequality constraints are elementwise, whether they involve scalars, vectors, or matrices. For example, together the constraints `0 <= x` and `x <= 1` mean that every entry of $\mathbf{x}$ is between 0 and 1. You cannot construct inequalities with < and >. Strict inequalities don't make sense in a real world setting. Also, you cannot chain constraints together, e.g., `0 <= x <= 1` or `x == y == 2`.

Quadratic form: Use `cvx.quad_form()` to create a quadratic form. For example, if you want to minimize portfolio variance, and you have a covariance matrix $\mathbf{P}$, the quantity `cvx.quad_form(x, P)` represents the quadratic form $\mathbf{x}^{\mathrm{T}}\mathbf{P}\mathbf{x}$, the portfolio variance.

Norm: Use `cvx.norm()` to create a norm term. For example, to minimize the distance between $\mathbf{x}$ and another vector, $\mathbf{b}$, i.e. $\|\mathbf{x} - \mathbf{b}\|_2$, create a term in the objective function `cvx.norm(x-b, 2)`. The second argument specifies the type of norm; for an L2-norm, use the argument 2.

Constants: Constants are the quantities in objective or constraint expressions that are not Variables. You can use your numeric library of choice to construct matrix and vector constants.

For instance, if `x` is a `cvxpy` `Variable` in the expression `A*x + b`, `A` and `b` could be Numpy ndarrays, Numpy matrices, or SciPy sparse matrices. `A` and `b` could even be different types.

Optimization problem: The core step in using `cvxpy` to solve an optimization problem is to specify the problem. Remember that an optimization problem involves minimizing an *objective function*, under some *constraints*, so to specify the problem, you need both of these. Use `cvx.Problem()` to declare the optimization problem. For example, `problem = cvx.Problem(objective, constraints)`, where `objective` and `constraints` are quantities you've defined earlier. Problems are immutable. This means that you cannot modify a problem's objective or constraints after you have created it. If you find yourself wanting to add a constraint to an existing problem, you should instead create a new problem.

Solve: Use `problem.solve()` to run the optimization solver.

Status: Use `problem.status` to access the status of the problem and check whether it has been determined to be unfeasible or unbounded.

Results: Use `problem.value` to access the optimal value of the objective function. Use e.g. `x.value` to access the optimal value of the optimization variable.

Rebalancing a Portfolio

TURNOVER

Take the absolute difference in weight between two time periods for each asset:

$$|\mathbf{x}_{t_1} - \mathbf{x}_{t_2}| = \left| \begin{bmatrix} x_{t_1,1} \\ x_{t_1,2} \\ \vdots \\ x_{t_1,n} \end{bmatrix} - \begin{bmatrix} x_{t_2,1} \\ x_{t_2,2} \\ \vdots \\ x_{t_2,n} \end{bmatrix} \right|$$

And then sum these over each asset:

$$\text{turnover} = |x_{t_1,1} - x_{t_2,1}| + |x_{t_1,2} - x_{t_2,2}| + \cdots + |x_{t_1,n} - x_{t_2,n}|$$

And then annualize the turnover:

$$\frac{\text{sum total turnover}}{\text{num total rebalancing events}} \times \text{num rebalancing events per year}$$

MODULE 4: ALPHA RESEARCH & FACTOR MODELING

Lesson 22: Factors

Offline Instructions

Rama Krishna B:
1)First Download the data/module* folder. This depends on your exercise. Try something line " !tar -cvf my.tar /data/ " and download and extract my.tar to your local workstation.

2)Change os.environ['ZIPLINE_ROOT'] variable in your python notebook. Set it to appropriate place based on where you extracted your data folder from the my.tar. Look at the change in my iPython notebook and do something similar.One easy way to do is to print the existing os.environ['ZIPLINE_ROOT'] and work accordingly.

3)If there is any reference to any file in the data folder, change the path accordingly.

4)Look at https://colab.research.google.com/drive/1Dj1fLiEA122N95vjgfLNOSRda5IbplIH
That is an exercise file from from one of our lessons.

To download twits data:
— I used json.dump() once the data was already loaded into notebook

```
with open('file_name.json', 'w') as f:
    json.dump(twits, f)
```

Another poster:
Zips all the files in the workspace so you can download them all at the same time:

```
zip -r workspace.zip ./*
```

Zips just data:

```
zip -r data.zip ../../data/*
```

Examples of a Factor

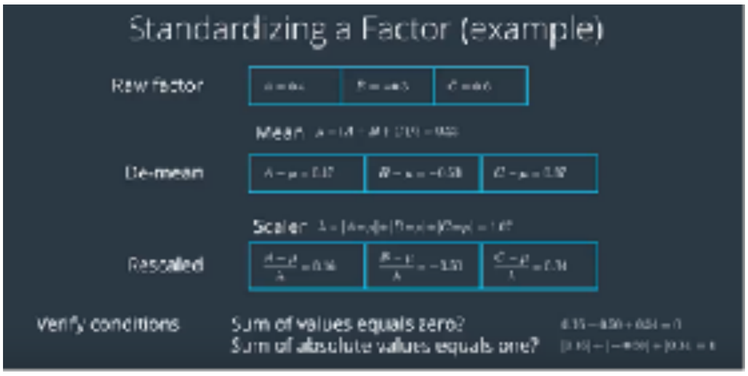
Alpha factors = drivers of mean returns
Risk factors = drivers of volatility

Factors can be: momentum, fundamentals, sentiment….

Formal definition:
- a list of numerical values, one for each stock, potentially predictive of an aspect of the performance of these stocks in the future

Standardized Factor

To standardize, have to both de-mean (so sum of weights = 0) and re-scale (so sum of absolute values equals 1 by finding the scalar, which is the sum of the absolute de-meaned values)



Leverage Ratio

The leverage ratio is the sum of the magnitudes of all positions, divided by the notional. The leverage ratio gives a sense of how much risk a portfolio is taking, because taking more positions magnifies both gains and losses. To standardize a factor, we divide by the sum of the magnitudes (sum of the absolute value of the positions), so that this rescaled vector's sum of magnitudes is equal to one. This makes different factors more comparable, because it's as if you're comparing different portfolios but each with the same amount of money placed on their positions.

STANDARDIZING A FACTOR

To make a factor dollar neutral, subtract its mean from each position.

To make the factor have a leverage ratio of one, divide by the sum of the absolute value of the positions.

11. Quiz:
- The sum of the de-meaned weights is always 0
- The sum of the re-scaled weights is always 0
- The absolute sum of the re-scaled weights is always 1
- The sum of the re-scaled short positions is always -0.5

Zipline Pipeline

Zipline uses **Data Bundles** to make it easy to use different data sources. A data bundle is a collection of pricing data, adjustment data, and an asset database.

Zipline's ingestion process will start by downloading the data or by loading data files from your local machine. It will then pass the data to a set of writer objects that converts the original data to Zipline's internal format (`bcolz` for pricing data, and `SQLite` for split/merger/dividend data) that hs been optimized for speed. This new data is written to a standard location that Zipline can find. By default, the new data is written to a subdirectory of `ZIPLINE_ROOT/data/<bundle>`, where `<bundle>` is the name given to the bundle ingested and the subdirectory is named with the current date.

In this notebook, we will be using stock data from **Quotemedia**. In the Udacity Workspace you will find that the stock data from Quotemedia has already been ingested into Zipline. Therefore, in the code below we will use Zipline's `bundles.load()` function to load our previously ingested stock data from Quotemedia. In order to use the `bundles.load()` function we first need to do a couple of things. First, we need to specify the name of the bundle previously ingested. In this case, the name of the Quotemedia data bundle is `eod-quotemedia`:

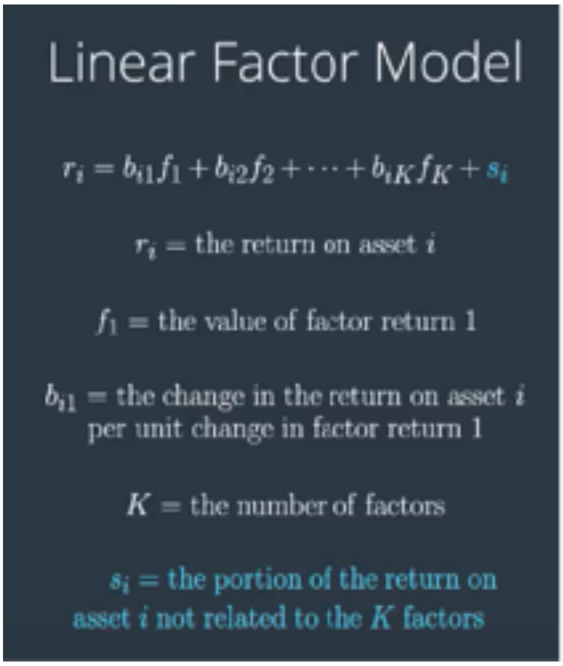
Second, we need to register the data bundle and its ingest function with Zipline, using the `bundles.register()` function. The ingest function is responsible for loading the data into memory and passing it to a set of writer objects provided by Zipline to convert the data to Zipline's internal format. Since the original Quotemedia data was contained in `.csv` files, we will use the `csvdir_equities()` function to generate the ingest function for our Quotemedia data bundle. In addition, since Quotemedia's `.csv` files contained daily stock data, we will set the time frame for our ingest function, to `daily`.

Lesson 23: Factor Model and Types of Factors

Linear Factor Model

Weighted contribution (the stock's exposure to) of multiple factors to a stock's return, plus an unexplained portion

- Linear regression is one way to build a linear factor model, but need to take into account latent or unobserved random variables



eg: how to assess if size is a factor in stock returns, ie: smaller companies provide greater returns?

What we would want to do is generate a single size-factor return vector time series

- This would be an example of a latent variable, something nebulous and hard to directly measure
- So we do so by creating a theoretical portfolio of long small caps, short large caps every day
- This time series of the portfolio's daily returns is then the latent variable

TERMINOLOGY

The terminology used to describe factor models varies widely. Here are some common phrases used to refer to the components of the model.

Factor returns (the fkS) may be:

- macro-economic variables
- returns on pre-specified portfolios,
- returns on zero-investment strategies (long and short positions of equal value) giving maximum exposure to fundamental or macro-economic factors,
- returns on benchmark portfolios representing asset classes,
- or something else.

The bjb coefficients may be called:

- factor exposures,
- factor sensitivities,
- factor loadings,
- factor betas,
- asset exposures

- style
- or something else.

The ei term may be called:

- idiosyncratic return,
- security-specific return,
- non-factor return,
- residual return,
- selection return
- or something else.

Factor Model Assumptions

- The residual return is assumed to be uncorrelated to each of the factor returns
- The residual return of one asset is uncorrelated with the residual return of any other asset
- So the only correlations to returns are to the factors

Covariance Matrix Using a Factor Model

We need to derive the covariance matrix of returns of a factor model (factor returns, exposure returns, and residual returns)

- Need the return variance in order to describe and control the portfolio variance and optimization

First, standardize returns distribution by subtracting the mean so mean becomes zero

Organize:

- returns **r** into a vector of N assets (N x 1)
- factor exposures **B** into a matrix of N x K assets and factors
  - Represents the sensitivity of returns to a given factor
- **f** is a vector of random variables, each representing the value of a factor return (K x 1)
- **s** is a vector of the residuals specific to each asset (N x 1)

$$r_i = b_{i1}f_1 + b_{i2}f_2 + \cdots + b_{iK}f_K + s_i$$

$$\mathbf{r} = \begin{bmatrix} r_1 \\ \vdots \\ r_N \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} B_{1,1} & \cdots & B_{1,K} \\ \vdots & \ddots & \vdots \\ B_{N,1} & \cdots & B_{N,K} \end{bmatrix} \qquad \mathbf{f} = \begin{bmatrix} f_1 \\ \vdots \\ f_K \end{bmatrix} \qquad \mathbf{s} = \begin{bmatrix} s_1 \\ \vdots \\ s_N \end{bmatrix}$$

$$N = number\ of\ companies$$

$$K = number\ of\ factors$$

Covariance formula below

$$\mathrm{Cov}(X,Y) = \frac{1}{n}\sum_{i=1}^{n}(X_i - \cancel{\bar{X}})(Y_i - \cancel{\bar{Y}})$$

but when means are zero, expectation value E simplifies to their product:

E(XY)

$$\mathrm{E}(\mathbf{A}) = \begin{bmatrix} \mathrm{E}(A_{11}) & \cdots & \mathrm{E}(A_{1p}) \\ \vdots & \ddots & \vdots \\ \mathrm{E}(A_{n1}) & \cdots & \mathrm{E}(A_{np}) \end{bmatrix}$$

The expectation value of a matrix is just the each:

$$\begin{bmatrix} r_1 \\ \vdots \\ r_N \end{bmatrix} \begin{bmatrix} r_1 & \cdots & r_N \end{bmatrix} = \begin{bmatrix} r_1 r_1 & \cdots & r_1 r_N \\ \vdots & \ddots & \vdots \\ r_N r_1 & \cdots & r_N r_N \end{bmatrix} = \mathbf{r}\mathbf{r}^T$$

And the matrix of products is just **r rTranspose**:

$$\begin{bmatrix} & \cdots & \cdots & \\ \vdots & \ddots & & \vdots \\ \vdots & \mathrm{Cov}(r_i, r_j) & \ddots & \vdots \\ & \cdots & \cdots & \end{bmatrix} = \mathrm{E}[\mathbf{r}\mathbf{r}^T]$$

So the covariance matrix is written as the expectation value of **r rTranspose**:

This expands and then, with some cancelling for the assumptions on non-correlation of residual returns to factor returns, to this:

- The **B** matrix can be pulled out of the expectation formula because it is not a random variable, it is fixed

$$\begin{bmatrix} & \cdots & \cdots & \\ \vdots & \ddots & & \vdots \\ \vdots & \mathrm{Cov}(r_i, r_j) & \ddots & \vdots \\ & \cdots & \cdots & \end{bmatrix} = \mathrm{E}[\mathbf{r}\mathbf{r}^T]$$

$$\mathrm{E}(\mathbf{r}\mathbf{r}^T) = \mathrm{E}[(\mathbf{B}\mathbf{f} + \mathbf{s})(\mathbf{B}\mathbf{f} + \mathbf{s})^T]$$

$$\mathrm{E}(\mathbf{r}\mathbf{r}^T) = \mathrm{E}[(\mathbf{B}\mathbf{f} + \mathbf{s})((\mathbf{B}\mathbf{f})^T + \mathbf{s}^T)]$$

$$\mathrm{E}(\mathbf{r}\mathbf{r}^T) = \mathrm{E}[\mathbf{B}\mathbf{f}(\mathbf{B}\mathbf{f})^T + \mathbf{B}\mathbf{f}\mathbf{s}^T + \mathbf{s}(\mathbf{B}\mathbf{f})^T + \mathbf{s}\mathbf{s}^T]$$

$$\mathrm{E}(\mathbf{r}\mathbf{r}^T) = \mathrm{E}[\mathbf{B}\mathbf{f}\mathbf{f}^T\mathbf{B}^T + \mathbf{B}\mathbf{f}\mathbf{s}^T + \mathbf{s}\mathbf{f}^T\mathbf{B}^T + \mathbf{s}\mathbf{s}^T]$$

$$\mathrm{E}(\mathbf{r}\mathbf{r}^T) = \mathbf{B}\mathrm{E}[\mathbf{f}\mathbf{f}^T]\mathbf{B}^T + \cancel{\mathbf{B}\mathrm{E}[\mathbf{f}\mathbf{s}^T]} + \cancel{\mathrm{E}[\mathbf{s}\mathbf{f}^T]\mathbf{B}^T} + \mathrm{E}[\mathbf{s}\mathbf{s}^T]$$

$$\mathrm{E}(\mathbf{r}\mathbf{r}^T) = \mathbf{B}\mathrm{E}[\mathbf{f}\mathbf{f}^T]\mathbf{B}^T + \mathrm{E}[\mathbf{s}\mathbf{s}^T]$$

$$\mathrm{E}(\mathbf{r}\mathbf{r}^T) = \mathbf{B}\mathbf{F}\mathbf{B}^T + \mathrm{E}[\mathbf{s}\mathbf{s}^T]$$

$$\mathbf{S} = \begin{bmatrix} S_{11} & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & S_{NN} \end{bmatrix}$$

$$\mathrm{E}(\mathbf{r}\mathbf{r}^T) = \mathbf{B}\mathbf{F}\mathbf{B}^T + \mathbf{S}$$

Factor Model in Quant Finance

Quants deviate from the history of factor model uses in modeling time series returns explicitly. They use factor modeling instead for devising and testing trading strategies and portfolio optimization, which means there are several simplifications made.

$$\text{Why } \mathrm{Var}(\beta_{i,1} f_1) \rightarrow \beta_{i,1}^2 \mathrm{Var}(f_1) \text{ ?}$$

$$\mathrm{Var}(f_1) = \frac{1}{2}[(f_{1,1} - \mu)^2 + (f_{1,2} - \mu)^2]$$

$$\mathrm{Var}(\beta_{i,1} f_1) = \frac{1}{2}[(\beta_{i,1} f_{1,1} - \beta_{i,1}\mu)^2 + (\beta_{i,1} f_{1,2} - \beta_{i,1}\mu)^2]$$

$$\mathrm{Var}(\beta_{i,1} f_1) = \frac{1}{2}[(\beta_{i,1}(f_{1,1} - \mu))^2 + (\beta_{i,1}(f_{1,2} - \mu))^2]$$

$$\mathrm{Var}(\beta_{i,1} f_1) = \frac{1}{2}[\beta_{i,1}^2(f_{1,1} - \mu)^2 + \beta_{i,2}^2(f_{1,2} - \mu)^2]$$

$$\mathrm{Var}(\beta_{i,1} f_1) = \beta_{i,1}^2 \times \underbrace{\frac{1}{2}[(f_{1,1} - \mu)^2 + (f_{1,2} - \mu)^2]}_{\mathrm{Var}(f_1)}$$

A portfolio with weights **x** has a portfolio factor exposure of $\mathbf{B}^T \mathbf{x}$:

$$\mathbf{B}^T\mathbf{x} = \begin{bmatrix} B_{1,1} & \cdots & B_{N,1} \\ B_{1,2} & \cdots & B_{N,2} \\ \vdots & \ddots & \vdots \\ B_{1,K-1} & \cdots & B_{N,K-1} \\ B_{1,K} & \cdots & B_{N,K} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} \mathbf{B}_1 \cdot \mathbf{x} \\ \mathbf{B}_2 \cdot \mathbf{x} \\ \vdots \\ \mathbf{B}_{K-1} \cdot \mathbf{x} \\ \mathbf{B}_K \cdot \mathbf{x} \end{bmatrix}$$

There are two groups of factors, those that relate to returns (ie: alpha factors), and those to volatility (ie: risk). So we want to constrain the volatility factors. Therefore, we drop the alpha factors from **B** so we only constrain the risk factors. The matrix **B** has become the risk loading matrix only. In this case then, F is the matrix of factors that have large influence on variance across all stocks. And s is the variance that's leftover. Only the risk factors are included in **F**, while any variance from the alpha factors shows up in **s**:

$$\mathbf{E(rr^T) = BFB^T + s}$$

The remaining alpha factors add to the objective function in optimization. Later on, we combine these into a single vector and optimize the portfolio weights to this.

Risk Factors v. Alpha Factors

In general, **risk factors** are significant contributors to the **variance** of asset returns, and less predictive of the mean of returns. Risk factors are identified to control risk. One way to do control an asset's exposure to a risk factor is to hold an equal amount long as short. For instance, a dollar neutral portfolio with equal amounts long and short is controlling for risks that the overall market may move up or down.

In general, risk factors that are significant in describing the **mean** of asset returns can be candidates for **alpha factors**. Alpha factors are used to give some indication of whether each stock in the portfolio may have positive expected returns or negative expected returns. For example, a former alpha factor was the market capitalization of a stock. Small cap stocks tend to have higher future returns compared to large cap stocks.

CHARACTERISTICS OF RISK FACTORS AND ALPHA FACTORS

Usually, we'd choose 20 to 60 risk factors that describe overall stock variance as much as possible. So risk factors as a whole account for more of the overall movement of stocks.

- eg: sector, country, interest rates….

On the other hand, there are fewer alpha factors and they contribute to smaller movements of stocks, which is okay, because we seek to identify these alpha factors because they give some indication of the direction of expected returns, even if they're small compared to risk factors.

- eg: book-to-market X idiosyncratic volatility, trajectory of stock returns over time

An important reason why it's important to identify risk factors and then neutralize a portfolio's exposure to risk factors is that if we didn't, the asset movements due to risk factors would overwhelm the movements that are due to the alpha factors.

Risk factors are well-known by the investment community, so investors will track those factors when optimizing their portfolios. This also means that it's unlikely that any one investor can gain a competitive advantage (higher than normal returns) using risk factors.

HOW AN ALPHA FACTOR BECOMES A RISK FACTOR
They get discovered / well-known, then become diffused. Among quants, you may hear the joke that "your alpha factor is my risk factor," since it's up to each fund to decide whether to use a factor to control risk or to drive returns.

Various Factor Type Examples

MOMENTUM OR REVERSAL FACTORS
Momentum = continuation of trend; could simply be expressed as + annual return
Reversal = mean reversion;

PRICE-VOLUME FACTORS
ie: technicals, net buying / selling, short interest,

FUNDAMENTAL RATIOS
P/E can lead to infinite large number problem when earnings tiny, so using earnings-to-price (earnings yield) is preferred

Also, book-to-price is preferred??

EVENT-DRIVEN
Natural disasters, government changes, interest rate changes, M&A, index add/delete
Earnings announcements, product announcements …

NLP
Sector Classification using 10-Ks
https://www.winton.com/research/systematic-methods-for-classifying-equities

Intro to NLP
See the extracurricular content for an introduction to Natural Language Processing.

10-K
We will use NLP to analyze 10-Ks in term 2. https://www.sec.gov/fast-answers/answersreada10khtm.html

Some companies that are aggregating alternate data
https://www.buildfax.com/
http://edi.om/
https://www.thinknum.com/
https://orbitalinsight.com/

Lesson 24: Risk Factor Models

Variance is the measure of risk. For a 2 stock portfolio the formula is the var weighted by the stock positions, plus 2 times each of the weights and the covariance fo the stocks returns:

$$\mathrm{Var}(r_p) = x_1^2 Var(r_1) + x_2^2 \mathrm{Var}(r_2) + 2x_1x_2\mathrm{Cov}(r_1, r_2)$$

But this does not scale, succumbs to the curse of dimensionality, so unable to calculate the measure of historical portfolio risk in real or near-real time (not sure if that is the right reason…). Instead we need to use the 'risk factor model' approach.

FACTOR MODEL OF RETURN
An asset's return can be modeled as the contribution of returns from a set of **factors**, plus the **specific return** (the part that isn't explained by the factors).

$$r_i = \overbrace{\sum_{k=1}^{K} (\beta_{i,k} \times f_k)}^{\text{Common Return}} + \underbrace{s_i}_{\text{Specific Return}}$$

Factor Model of Portfolio Return

FACTOR EXPOSURE OF PORTFOLIO
The portfolio exposure to a single factor is simply the weighted average of each of the portfolio's stocks exposures to that factor. Repeat to get portfolio exposure for each factor:

Factor Exposures of each stock to factor "k"

$\beta_{1,k}, \beta_{2,k}, \ldots \beta_{N,k}$

Stock weights in portfolio

$x_1, x_2 \ldots x_N$

Factor exposure of portfolio to factor k?

$\beta_{p,k} = (x_1 \times \beta_{1,k}) + (x_2 \times \beta_{2,k}) + \ldots + (x_N \times \beta_{N,k})$

Contributions of factors to Portfolio return

$$r_p = (\beta_{p,1} \times f_1) + (\beta_{p,2} \times f_2) + \ldots + (\beta_{p,k} \times f_k) + \ldots$$

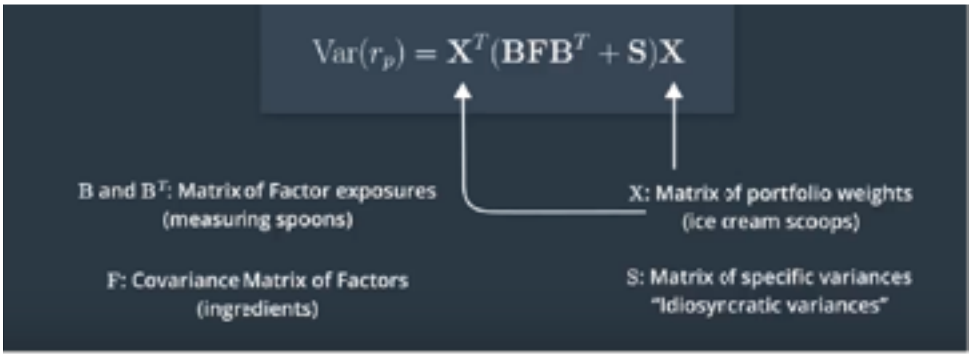
$$\beta_{p,1} = \sum_{i=1}^{N} (x_i \times \beta_{i,1}) \qquad \beta_{p,2} = \sum_{i=1}^{N} (x_i \times \beta_{i,2}) \qquad \beta_{p,k} = \sum_{i=1}^{N} (x_i \times \beta_{i,k})$$

And then just account for the portfolio's specific return:

$$r_p = \underbrace{\sum_{k=1}^{K} (\beta_{p,k} \times f_k)}_{\text{Contribution of Factors}} + \underbrace{s_p}_{\text{Specific return}} \qquad s_p = \sum_{i=1}^{N} (x_i \times s_{i,k})$$

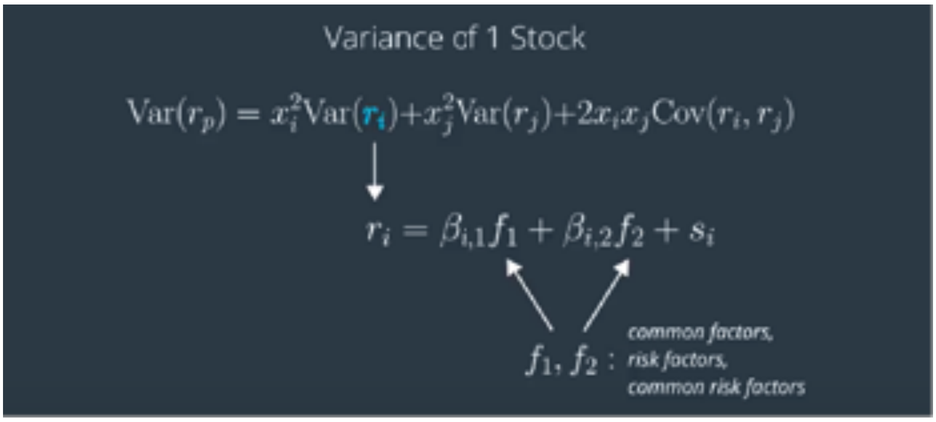
Factor Model of Portfolio Variance

Similar approach as in calculating portfolio factor returns



DO EXERCISES 5 AND 8….

VARIANCE OF ONE STOCK
Each stock's return input to calculate portfolio variance is itself a formula calculated from its weighted exposures to the various factors:



Then you find the variance of r_i as:

- The last line moves the constants (factor exposures) outside (since they are constant, they don't vary)
  - For the 1st two, they need to be squared as the variance is squared (second image)

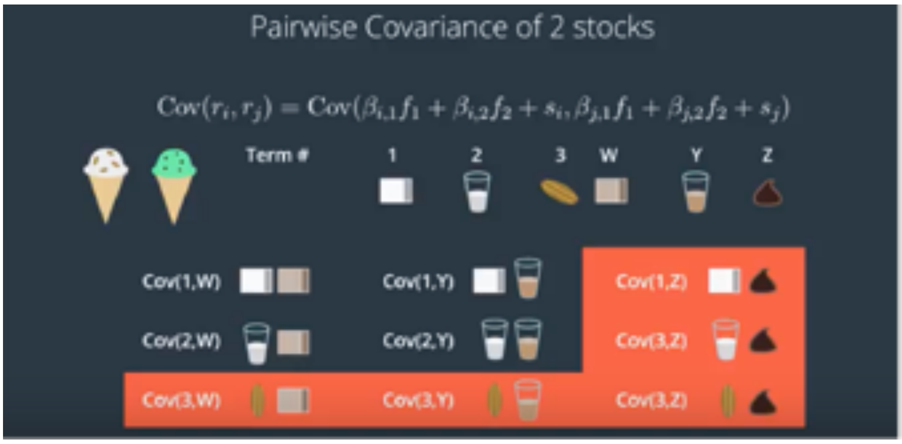
$$\mathrm{Var}(r_i) = \mathrm{Var}(\beta_{i,1}f_1 + \beta_{i,2}f_2 + s_i)$$

$$\mathrm{Var}(r_i) = \mathrm{Var}(\beta_{i,1}f_1) + \mathrm{Var}(\beta_{i,2}f_2) + 2\mathrm{Cov}(\beta_{i,1}f_1, \beta_{i,2}f_2) + Var(s_i)$$

$$\mathrm{Var}(r_i) = \beta_{i,1}^2\mathrm{Var}(f_1) + \beta_{i,2}^2\mathrm{Var}(f_2) + 2\beta_{i,1}\beta_{i,2}\mathrm{Cov}(f_1, f_2) + \mathrm{Var}(s_i)$$

PAIRWISE COVARIANCE OF TWO STOCKS
The covariance of two stocks can be written as the sum of the covariances of the factors. In this example we have two factors, so we have four covariance terms. If we were using three factors to describe the asset returns, there would be three times three or nine covariance terms.

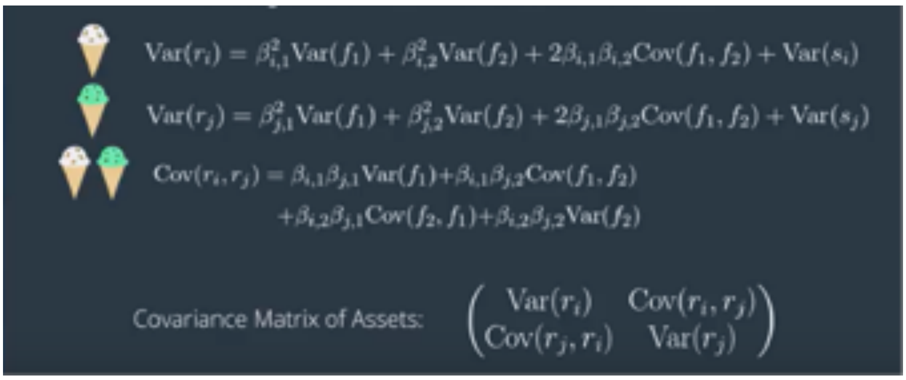
- The covariances of any of the specific returns are assumed to not be correlated so can be excluded



- Highlighted is the contribution of factor 1 to stock *i*, and factor 1 to stock *j*:

The covariance of a factor with itself is just that factor's variance. We can also move the constant exposures outside to simplify the formulas as follows:

So now we have the building blocks for the Covariance Matrix of Assets:



The other way to do this would be to use the time-series of each stock to calculate the variance and covariances, but this doesn’t scale well, so using the factor-based approach above is much better.

PORTFOLIO VARIANCE WITH MATRIX NOTATION

$$\mathrm{Var}(r_p) = x_i^2\mathrm{Var}(r_i)+x_j^2\mathrm{Var}(r_j)+2x_ix_j\mathrm{Cov}(r_i,r_j)$$

=

$$\mathrm{Var}(r_p) = \mathbf{X}^T(\mathbf{B}\mathbf{F}\mathbf{B}^T + \mathbf{S})\mathbf{X}$$

$$\mathbf{F} = \begin{pmatrix} \mathrm{Var}(f_1) & \mathrm{Cov}(f_1,f_2) \\ \mathrm{Cov}(f_2,f_1) & \mathrm{Var}(f_2) \end{pmatrix}$$

$$\mathbf{B} = \begin{pmatrix} \beta_{i,1}, \beta_{i,2} \\ \beta_{j,1}, \beta_{j,2} \end{pmatrix} \quad \mathbf{B}^T = \begin{pmatrix} \beta_{i,1}, \beta_{j,1} \\ \beta_{i,2}, \beta_{j,2} \end{pmatrix}$$

$$\mathbf{S} = \begin{pmatrix} \mathrm{Var}(s_i) & 0 \\ 0 & \mathrm{Var}(s_j) \end{pmatrix}$$

$$\mathbf{X} = \begin{pmatrix} x_i \\ x_j \end{pmatrix} \quad \mathbf{X}^T = \begin{pmatrix} x_i & x_j \end{pmatrix}$$

Portfolio Variance Matrix Notation (putting the pieces together)

$$\mathrm{Var}(r_p) = \begin{pmatrix} x_i & x_j \end{pmatrix}\left(\begin{pmatrix} \beta_{i,1}, \beta_{i,2} \\ \beta_{j,1}, \beta_{j,2} \end{pmatrix}\begin{pmatrix} \mathrm{Var}(f_1) & \mathrm{Cov}(f_1,f_2) \\ \mathrm{Cov}(f_2,f_1) & \mathrm{Var}(f_2) \end{pmatrix}\begin{pmatrix} \beta_{i,1}, \beta_{j,1} \\ \beta_{i,2}, \beta_{j,2} \end{pmatrix} + \begin{pmatrix} \mathrm{Var}(s_i) & 0 \\ 0 & \mathrm{Var}(s_j) \end{pmatrix}\right)\begin{pmatrix} x_i \\ x_j \end{pmatrix}$$

TYPES OF RISK MODELS

- Time Series Risk Models: CAPM and Fama French 3 Factor Model
- Cross Sectional Risk Models
- PCA Risk Models

Lesson 25: Time Series and Cross Sectional Risk Models

Time Series Models

FACTOR VARIANCE

Covariance Matrix of Factors
We'll collect a time series that represents the chosen factor. In this case, our factor is "**market excess return**", so we can use an index (such as the S&P500) and subtract a time series that represents the **risk-free rate**, such as the three-month US Treasury Bill rate. Calculating the variance of this market excess return helps us fill in the covariance matrix of factors.

So:

$$\mathbf{Var(r_p) \;=\; X^T\,(BFB^T + S)\;X}$$

Where:

F^T = var(r_{market} - r_{rfr})

B = col vector of exposure weights for each stock to the single factor

S = matrix of specific returns

$$S = \begin{pmatrix} Var(s_i) & 0 \\ 0 & Var(s_j) \end{pmatrix}$$

FACTOR EXPOSURE
We can **use regression to calculate the factor exposures** in a time series model. We'll use the **asset's excess return as the dependent "y" variable**, and the **factor return** (in this case, market excess return) **as the independent "x" variable**. The estimated coefficient from the regression is an estimate of the asset's "exposure" to that factor.

CAPM

$$r_i - r_f = \underbrace{\beta_{i,1}}_{\text{Market exposure: factor exposure}} \times \underbrace{(r_m - r_f)}_{f_m} + c_i$$

Estimate the factor exposures!
One way: regression!

SPECIFIC RETURN
To get the specific return, take the difference between the actual return and estimated return.

Specific return at time "t"

$$s_{i,t} = \underbrace{(r_{i,t} - r_{f,t})}_{\text{Actual return}} - \underbrace{(\beta_{i,1} \times f_{m,t} + c_i)}_{\text{Estimated return}}$$

TIME SERIES RISK MODEL

The covariance matrix of assets can be considered the plug-and-play component of various portfolio models. This is just multiplied by the transpose of portfolio weights and the portfolio weights, which can then be optimized.

$$\mathrm{Var}(r_p) = \mathbf{X}^T \underbrace{(\mathbf{BFB}^T + \mathbf{S})}_{\text{Covariance matrix of assets}} \mathbf{X}$$

Fama French Models (Time Series)

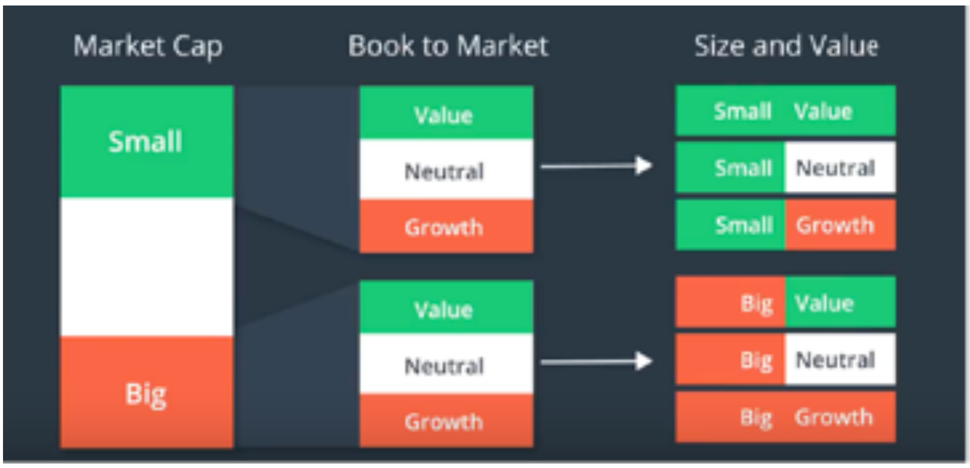
THEORETICAL PORTFOLIOS
If we had a hypothesis that a particular attribute affects returns, we could try to quantify that using a theoretical portfolio. It's common to use a dollar neutral theoretical portfolio, which means we'd have an equal dollar amount long as short. We'd go long on assets that have more of a particular trait that may improve returns, and go short on assets that have less of that trait. This is a way to create a "measurement" of how that attribute influences returns.

SMB (SMALL MINUS BIG)
To create a theoretical portfolio representing size, we could go long the bottom 10th percentile of stocks by market cap (long small cap stocks) and go short stocks above the 90th percentile (go short the large cap stocks). We could assume an equal dollar amount invested in each stock. In the above example, we are dividing by 2 to take the average return of going long small cap stocks and going short large cap stocks.

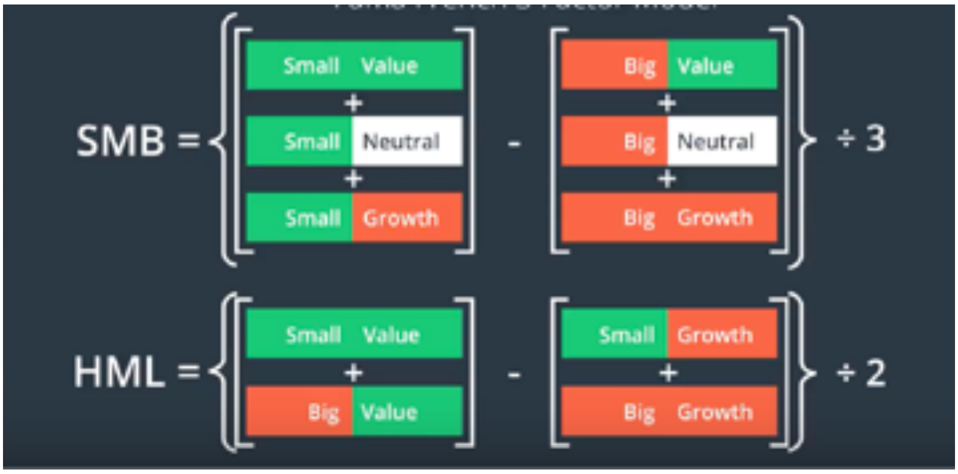
It's also common to compute the spread between two portfolios. One portfolio contains the small cap stocks, and the other portfolio contains the large cap stocks. In this case, we'd just take the difference between the returns of the two portfolios.

VALUE (HML - HIGH MINUS LOW)
Sort universe by book-to-market value. High book/MV are the value stocks and low refers to the growth portfolio.

FAMA AND FRENCH SMB AND HML: 3 FACTOR MODEL
Sort market by size, take top small caps, then sort by value / neutral / growth. Do the same for large caps. Will now have 6 portfolio groupings.



The SMB big portfolio takes 3 parts small value / neutral / growth minus 3 parts large value / neutral / growth. The HML just takes 2 part small value / big value minus 2 parts small growth / big growth. And since we want equal weighted of SMB and HML, we divide SMB by 3 and HML by 2 and combine them together:



$$SMB = \frac{1}{3}((r_{s,v} + r_{s,n} + r_{s,g}) - (r_{b,v} + r_{b,n} + r_{b,g}))$$

$$HML = \frac{1}{2}((r_{s,v} + r_{b,v}) - (r_{s,g} + r_{b,g}))$$

FAMA FRENCH RISK MODEL

Use the cov matrix of assets to fill in from before:

$$\mathrm{Var}(r_p) = \mathbf{X}^T\underbrace{(\mathbf{BFB}^T + \mathbf{S})}_{\text{Covariance matrix of assets}}\mathbf{X}$$

F will be a 3 x 3 covariance matrix of the 3 factors. The first column has the market factors, the second has the size factros, and the third has the value factors:

$$F = \begin{pmatrix} Var(f_m) & Cov(f_m, f_s) & Cov(f_m, f_v) \\ Cov(f_s, f_m) & Var(f_s) & Cov(f_s, f_v) \\ Cov(f_v, f_m) & Cov(f_v, f_s) & Var(f_v) \end{pmatrix}$$

So we need time series of each of the 3 factor portfolios:

$$\begin{array}{ll} f_m & r_m - r_f \\ f_s & SMB = \frac{1}{3}((r_{s,v} + r_{s,n} + r_{s,g}) - (r_{b,v} + r_{b,n} + r_{b,g})) \\ f_v & HML = \frac{1}{2}((r_{s,v} + r_{b,v}) - (r_{s,g} + r_{b,g})) \end{array}$$

Then we create the factor exposure matrix, the size or which depends on the number of stocks. Here we assume only 2 stocks:

$$B = \begin{pmatrix} \beta_{i,m} & \beta_{i,s} & \beta_{i,v} \\ \beta_{j,m} & \beta_{j,s} & \beta_{j,v} \end{pmatrix} \quad B^T = \begin{pmatrix} \beta_{i,m} & \beta_{j,m} \\ \beta_{i,s} & \beta_{j,s} \\ \beta_{i,v} & \beta_{j,v} \end{pmatrix}$$

And we calculate our estimate of these factor exposures using multiple regression for each stock *i, j*:

$$r_i = (\beta_{i,m} \times f_m) + (\beta_{i,s} \times f_s) + (\beta_{i,v} \times f_v)$$
$$r_j = (\beta_{j,m} \times f_m) + (\beta_{j,s} \times f_s) + (\beta_{j,v} \times f_v)$$

The specific return matrix S has the variance of non-factor returns and is calculated as:

$$\mathbf{S} = \begin{pmatrix} \mathrm{Var}(s_i) & 0 \\ 0 & \mathrm{Var}(s_j) \end{pmatrix}$$

$$s_i = r_{i,actual} - r_{i,estimated}$$
$$r_{i,estimated} = (\beta_{i,m} \times f_m) + (\beta_{i,s} \times f_s) + (\beta_{i,v} \times f_v)$$

$$s_j = r_{j,actual} - r_{j,estimated}$$
$$r_{j,estimated} = (\beta_{j,m} \times f_m) + (\beta_{j,s} \times f_s) + (\beta_{j,v} \times f_v)$$

Cross Sectional Model

A cross-section means that we use multiple stocks for a single time period in a calculation. In contrast, a time series is looking at a single stock over multiple time periods.

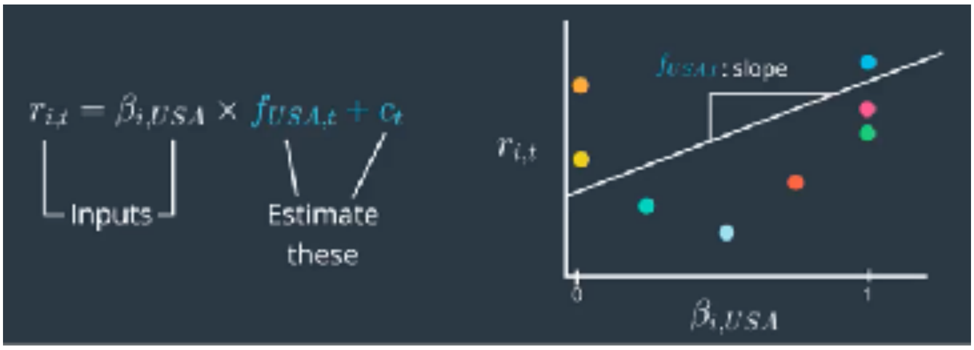
A cross-sectional model calculates the factor exposure first, and then uses that information to estimate the factor return.

CATEGORICAL FACTORS (ONE-HOT ENCODING)
When handling categorical variables, we can make each unique value within a category be its own variable. In this example, the country variable becomes "country_usa", "country_india", "country_brazil" etc. Then assign a value to each of these variables to represent how "exposed" the company is to each country. With one-hot encoding, the country exposed would get 1, the rest zero (or as a decimal between 0 and 1 to represent percentage exposure).

ESTIMATING FACTOR RETURN
If we collect a cross-section of multiple stocks for a single time period, then we'll have pairs of stock returns and factor exposures. We can use regression to estimate the factor return for that single time period. Then repeat over multiple time periods to get a time series of factor returns.

1. Estimate $f_{USA,t}$ for a single time period t using multiple stocks $i = 1, \dots N$
   - Get the returns
   - Each stock return will have a factor associated with it
     - The factor exposure (ie: to USA) will range between 0 and 1 for each stock

So, given the stock returns and the factor exposures, we then estimate the factor returns and specific returns. We can use regression to estimate the factor return for that single time period:

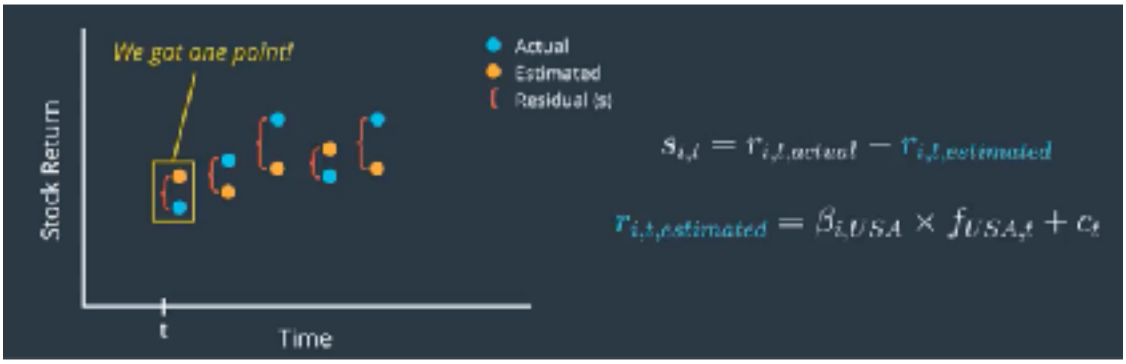


Then repeat this for each time period and we can get the time series factor return f_{USA} and then calculate the variance to plug into the covariance matrix of assets.

ESTIMATING SPECIFIC RETURN

For each time period, we can calculate the specific return as the difference between actual stock return and estimated stock return (using the estimated factor return). Do this for multiple time periods to get a time series of specific return.

The estimate single time period return for a stock is calculated by the factor exposure (country) multiplied by the factor (country) return plus the plus the intercept term, all at time *t*.



Repeat for all time periods to get the specific return time series for a given stock and use it to calculate the variance to plug into the specific return variance matrix

$$S = \begin{pmatrix} Var(s_i) & 0 \\ 0 & Var(s_j) \end{pmatrix}$$

Now we have everything required for the blue highlights:

$$\mathrm{Var}(r_p) = \mathbf{X}^T(\mathbf{BFB}^T + \mathbf{S})\mathbf{X}$$

Fundamental Factors

In a cross-sectional risk model, the fundamental data calculated on a company, based on its financials, can be used as the factor exposure of that company, to that factor. We can use regression on a cross-section of stocks to estimate the factor return.

Example, book-to-value is a fundamental factor, updated quarterly
Market cap, which is updated daily

- In a cross-sectional, we use these as factor exposures (rather than factor returns as in time-series approach)

We will estimate the factor returns using regression:

$$r_i = (\beta_{i,v} \times f_v) + (\beta_{i,s} \times f_s) + c_i$$

1. Get the factor exposures of each stock to each factor (ie: book-MCAP, MCAP)
2. Get each stocks return for one time period: r_i
3. Regress the stocks returns against the factor exposures.
4. Repeat to get a time series for the factor returns
5. Calculate the variance and covars:
6. Calculate the specific returns by subtracting the estimated stock returns using the chosen factors

$$r_i = (\beta_{i,v} \times f_v) + (\beta_{i,s} \times f_s) + c_i$$

$$\mathrm{Var}(f_v),\ \mathrm{Var}(f_s);\ \mathrm{Cov}(f_v, f_s)$$

In practice, institutional investors usually purchase a risk model that uses the cross-cross-sectional approach, while academic research papers typically use the time-series approach.

There is also a third approach — using principal component analysis.

Lesson 26: Risk Factor Models with PCA

Bases

BASES AS LANGUAGES
- i-hat; j-hat are the bases
- Vectors can be written as linear combinations of the basis vectors

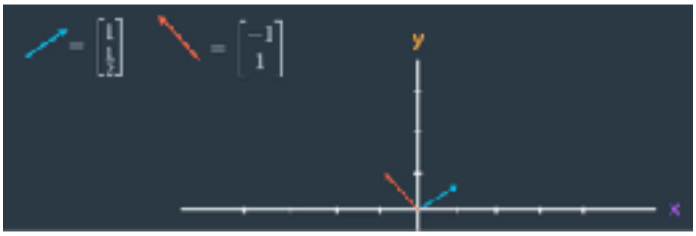
A set of vectors is a basis for a space if:
1. No vector in the set is a linear combination of the others
2. Every vector in the space (like the 2D plane) can be written as a linear combination of the set of vectors

Can express this linear combination of the basis vectors; as this:

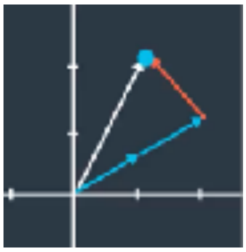
$$1 \times \begin{bmatrix}1\\0\end{bmatrix} + 2 \times \begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}1\\2\end{bmatrix}$$

$$\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\begin{bmatrix}1\\2\end{bmatrix} = \begin{bmatrix}1\\2\end{bmatrix}$$

TRANSLATING BETWEEN BASES
eg: old to new
- top left are the new ‘prime basis expressed in the language of our original/normal basis
- How to express our [1, 2] vector in the new basis language?



- we know we need 2 copies of the blue vector and one of the red vector to represent our original vector [1, 2] (as plotted):



  - And as in a linear combination:

$$2 \times \begin{bmatrix}1\\\frac{1}{2}\end{bmatrix} + 1 \times \begin{bmatrix}-1\\1\end{bmatrix} = \begin{bmatrix}1\\2\end{bmatrix}$$

- So we can see how the translation must work
  - Write the new basis vectors in the language of the old basis language
  - Then the formula for building the basis in the new language (the 2 x and 1x above) gives us the the vector in our old basis language

$$\underbrace{\begin{bmatrix}1 & -1\\\frac{1}{2} & 1\end{bmatrix}}_{\text{New basis vectors written in old basis language}}\underbrace{\begin{bmatrix}2\\1\end{bmatrix}}_{\text{Our vector in new basis language}} = \underbrace{\begin{bmatrix}1\\2\end{bmatrix}}_{\text{Our vector in old basis language}}$$

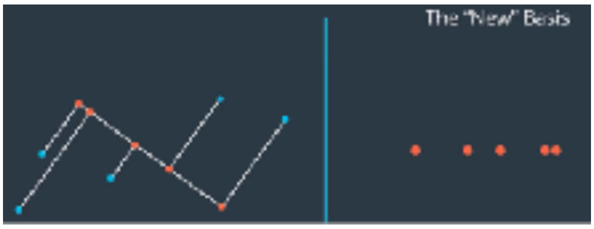
- Conversely, to go the other way, we would have to find out how to write the i-hat and j-hat in the language of the new basis

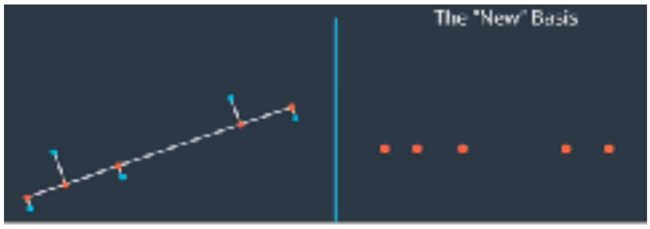
$$\begin{array}{cc} \underbrace{\begin{matrix}?\\?\end{matrix}}_{\text{How to write }\hat{i}\text{ in new basis}} & \underbrace{\begin{matrix}?\\?\end{matrix}}_{\text{How to write }\hat{j}\text{ in new basis}} \end{array} \overbrace{\begin{bmatrix}1\\2\end{bmatrix}}^{\text{Our vector in old basis language}} = \underbrace{\begin{bmatrix}2\\1\end{bmatrix}}_{\text{Our vector in new basis language}}$$

- It turns out that the matrix to translate from the new basis to the old basis is the inverse of the new basis vectors written in the old basis language
  - So inverse of $\begin{bmatrix} 1 & -1 \\ 1/2 & 1 \end{bmatrix}$
  - Which is $\begin{bmatrix} 2/3 & 2/3 \\ -1/3 & 2/3 \end{bmatrix}$

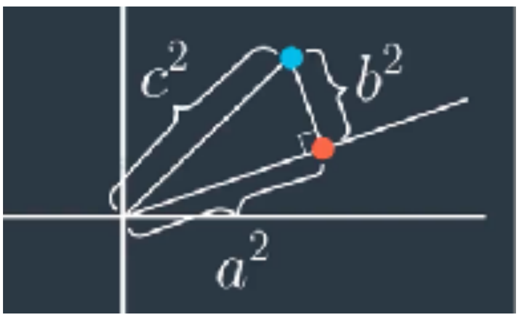
PCA: The Core Idea

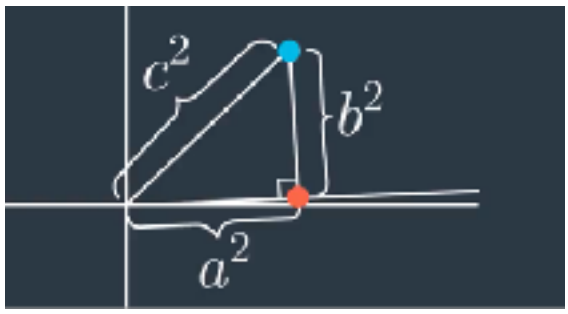
PCA is a series of calculations that gives us a new, special basis for our data.
- The aim is to choose a new basis where the data points along the first dimension are the most spread out
  - ie: maximizes the variance between the points projected onto the first plane
- The effect of this is that it ends up minimizing the distance of the projections to the plane
  - Known as 'minimizing reconstruction error'
- So, you start with your set of data (the blue dots), then add a line and orthogonally project the dots onto the line, then optimize for maximum variance or spread of the project points on the line





- The squared distance from the origin to the projection (a^2) and the squared distance from the projection to the original point (b^2) equals the squared distance from the origin to the point (c^2)
  - So when you change the orientation of one of the lines, if one of the distance increases, another must decrease
    - the orientation of the line chosen by PCA is the one the maximizes the squared distances along the line for all points while simultaneously minimizing the projection distances for all points





This is how we find the first basis direction. The next basis direction must be perpendicular (orthogonal) to the first line.
- If we working in more the 2D, then it would also aim to maximize the variance of the projected points along that dimension, and so on, until you have as many new dimensions as started with
  - By new dimensions, must just mean new bases

PCA: Detailed Steps

1. Make sure the data are centered around zero
   - Means you need to subtract the means from each dimension, eg: for 2D, subtract the x means from the x points, and subtract the y means from the y points
   - Called mean centering or mean normalizing the data

2. D

… these are the 'writing it down' videos …

The Principal Components

So far have discussed in terms of geometric space. But also represent column vectors of say individual returns of a stock over time. The new vectors / PCs under the new basis system may or may not represent something that makes sense in the real world. This refers to whether or not the PCs are interpretable. **(principal component = eigenvector)**

HOW TO USE
In practice we don't end up using all dimensions of the PCs. Instead we use the ones that explain the most, starting with the first dimension. We also express the PC positions along the line only and not their position orthogonal to the line. But if we have found a basis system that minimizes those orthogonal distances, we don't lose much information.

One way to decide on how many PCs to use is by their variance. The lower the variance, the lower the information, so those lowest variance PCs can be dropped.

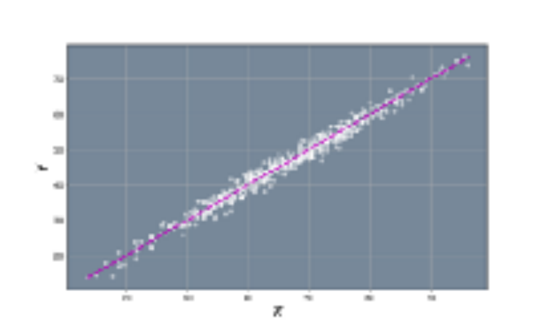
The sum of the distance from the origin to each data point (ie: along the hypotenuse) is the total variance, and this is the same in the original basis system and the new basis system.

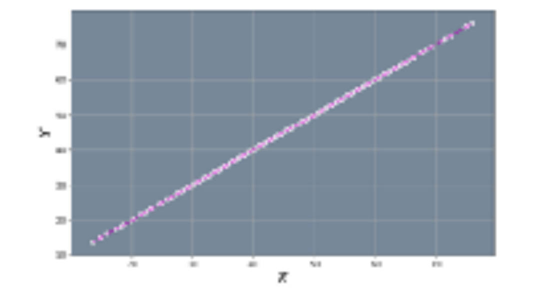
Each successive PC dimension's total variance is less the one before it. So we can reduce the dimensionality of data set by dropping those with the least variance but still retain most of the information and make the problem more manageable.

In short, dimensionality reduction is the process of reducing the number of variables used to explain your data.

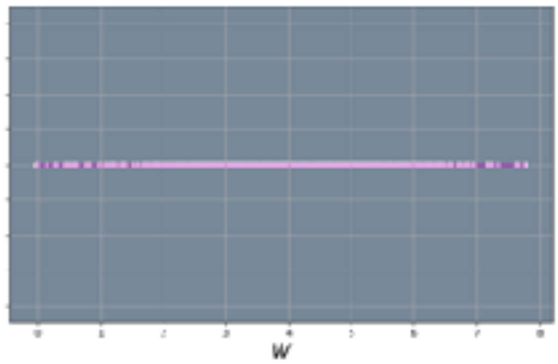
The Jupyter PCA Toy Problem

HAS THE BEST EXPLANATION:
We can see that this 2-Dimesnional data is described by two variables, X and Y. However, notice that all the data points lie close to a straight line (LHS). We can see that most of the variation in the data occurs along this particular purple line. This means, that we could explain most of the variation of the data by only looking at how the data is distributed along this particular line. Therefore, we could reduce the data from 2D to 1D data by projecting the data points onto this straight line (RHS).





This will reduce the number of variables needed to describe the data from 2 to 1 since you only need one number to specify a data point's position on a straight line. Therefore, the 2 variables that describe the 2D plot will be replaced by a new single variable that encodes the 1D linear relation.



It is important to note, that this new variable and dimension don't need to have any particular meaning attached to them. For example, in the original 2D plot, X and Y may represent stock returns, however, when we perform dimensionality reduction, the new variables and dimensions don't need to have any such meaning attach to them. The new variables and dimensions are just abstract tools that allow us to express the data in a more compact form. While in some cases these new variables and dimensions may represent a real-world quantities, it is not necessary that they do.

PCA IN BRIEF

How do we find the best straight line to project our data onto? We define the best line as the line such that the sum of the squares of the distances of the data points to their projected counterparts is minimized. It is important to note, that these projected distances are orthogonal to the straight line, not vertical as in linear regression. Also, we refer to the distances from the data points to their projected counterparts as *projection errors*.

In general, for N-Dimensional data, PCA will find the lower dimensional surface on which to project the data so as to minimize the projection error. The lower dimensional surface is going to be determined by a set of vectors $v^1, v^2, \dots, v^k$ where k is the dimension of the lower dimensional surface, with $k < N$. Therefore, what the PCA algorithm is really doing is finding the vectors that determine the lower dimensional surface that minimizes the projection error.We also define the first principal component to be the eigenvector corresponding to the largest eigenvalue of X; the second principal component as the eigenvector corresponding to the second largest eigenvalue of X, and so on. If $v^1, v^2, \dots, v^N$ is the set of eigenvectors for X, then the principal components of X will be determined by the subset $v^1, v^2, \dots, v^k$ for some chose value of $k < N$.

SCIKIT-LEARN PCA

Scikit-Learn's `PCA()` class uses a technique called **Singular Value Decomposition (SVD)** to compute the eigenvectors and eigenvalues of a given set of data. Given a matrix X of shape *(M, N)*, the SVD algorithm consists of factorizing X into 3 matrices U, S, V such that $X = USV$.

The shape of the U and V matrices depends on the implementation of the SVD algorithm. When using Scikit-Learn's `PCA()` class, the U and V matrices have dimensions (M, P), and (P, N), respectevely, where P = min(M, N). The V matrix contains the eigenvectors of X as rows and the S matrix is a diagonal (P, P) matrix and contains the eigenvalues of X arranged in decreasing order, *i.e.* the largest eigenvalue will be the element S_{11}, the second largest eigenvalue will be the element S_{22}, and so on. The eigenvectors in V are arranged such that the first row of V holds the eigenvector correspoding to the eigenvalue in S_{11}, the second row of V will hold the eigenvector corresponding to eigenvalue in S_{22}, and so on.

Once the eigenvectors and eigenvalues have been calculated using SVD, the next step in dimensionality reduction using PCA is to choose the size of the dimension we are going to project our data onto. The size of this dimension is determined by k, which tells us the number of principal components we want to use.

After we have set the parameters of our PCA algorithm, we now have to pass the data to the `PCA()` class. This is done via the `.fit()` method. It returns an array containing the principal components in the attribute `.components_`, and the corresponding eigenvalues in a 1D array in the attribute `.singular_values_`. Other attributes of the `PCA()` class include `.explained_variance_ratio_` which gives the percentage of variance explained by each of the principal components.

Typically, you choose *k* such that anywhere from 80% to 99% of the variance of the original data is retained. You check the percentage of the variance of your data that is explained for a given value of *k* using the `.explained_variance_ratio_` attribute. You sum the variance of the *k* components to see if that variance meets your threshold.

Once we find the vectors (principal component) that span our lower dimensional space, the next part of the dimensionality redcution algorithm is to find the projected values of our data onto that space. We can use the `.transform()` method from the `PCA()` class to apply dimensionalty reduction and project our data points onto the lower dimensional space.

PCA as a Factor Model

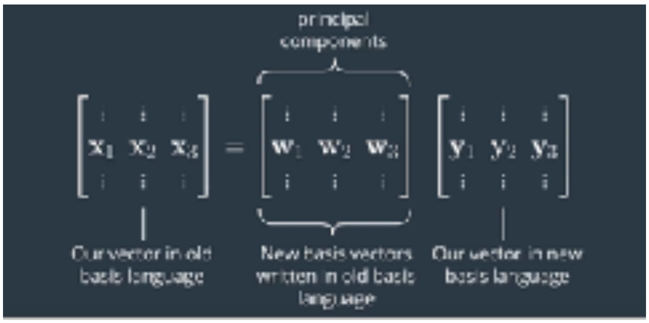
Involves a lot of matrix multiplication, so important to keep track of the various matrix dimensions.

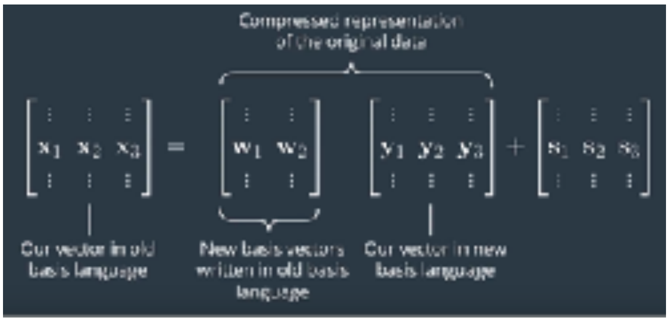
r = Bf + s

Where:

- ***r*** returns = (N companies, T time points)
- ***B*** factor exposures = (N companies, K factors)
- ***f*** factor returns = (K factors, T time points)
- **S** idiosyncratic / systematic returns = (N companies, T time points)

If we run PCA and keep all dimensions, we get LHS below, and RHS is the compressed representation of the data if we reduce dimensionality:





The RHS compressed representation looks just like our factor model r = Bf + s. The matrix **W** of new basis vectors expressed in the old basis language *is* the factor exposure matrix **B**. The matrix **Y** in the new basis coordinates is the matrix **f** of factor returns.

The factor returns are calculated by multiplying the transpose of the original factor exposures by the returns (since the basis is orthonormal, the inverse of **B** is just its transpose):

f = B^T * r

Remember, for the risk model, we need to calculate the factor covariance. Since factor returns are now projections onto the PCs, they are orthogonal (orthonormal) and so the covariance matrix is a diagonal matrix. Since returns are daily, we need to annualize, multiply by 252.

$$\mathbf{F} = \frac{1}{T-1}\mathbf{f}\,\mathbf{f}^T$$

$$\mathbf{F} = 252\begin{pmatrix} F_{1,1} & 0 & \cdots & 0 \\ 0 & F_{2,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & F_{K,K} \end{pmatrix}$$

Lastly, we need to calculate the idiosyncratic or specific risk matrix **s = r - Bf**. To calculate the specific risk matrix, calculate the covariance covariance matrix of the residuals and set the off-diagonals to zero.

$$\mathbf{S} = \frac{1}{T-1}\mathbf{s}\,\mathbf{s}^T$$

$$\mathbf{S} = \begin{pmatrix} S_{1,1} & 0 & \cdots & 0 \\ 0 & S_{2,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & S_{K,K} \end{pmatrix}$$

Lesson 27: Alpha Factors

DEFINITION OF KEYWORDS
- Alpha model - is an algorithm that transforms data into numbers associated with each stock per time step.
- Alpha value - refers to a single value for a single stock, for a single time period.
  - Positive number = buy
  - Negative number = short
- Alpha vector - has a number for each stock, and the number is proportional to the amount of money we wish to allocate for each stock.
  - ie: it is the list of alpha values for each stock for a single time period (row vector)
  - Each number is proportional to amount invested in each stock
- Alpha factor - a time series of alpha vectors (over multiple time periods).
  - ie: the dataframe
- Raw alpha factor - a version of an alpha factor before additional processing.
- Stock universe - set of stocks under consideration for the portfolio.

Researching Alphas from Academic Papers

Should not expect to get stong production ready models — they are likely already diffused in the market

REASONS TO STUDY ACADEMIC PAPERS:
1. Spur idea generation
2. Baseline for comparison to our own models
3. New methods and techniques
4. New data sources or novel ways to work with data

Open source repositories are free but not peer reviewed (but this means they are often available more timely as peer review takes time.

THINGS TO CHECK / LOOK OUT FOR:
- Use of citation to find other related works and validate claims
- Focus on papers that you can replicate (eg: you can get the data etc)
- Check if methodology is practical, does it have prohibitively high turnover? transaction costs? unrealistic liquidity (slippage)? etc.

SOURCES
1. arxiv: https://arxiv.org/archive/q-fin is searchable
2. SSRN: https://papers.ssrn.com/sol3/topten/topTenResults.cfm?groupingtype=2&groupingId=203

Controlling for Risk with Alpha Factors

The aim is for our portfolio to be neutral to common risk factors. Portfolio optimization neutralizes exposure to common risk factors, but we should not wait to rely solely on portfolio optimization to neutralize these exposures. It’s best to consider obvious common risks, even at the alpha factor research stage.

MOST SIGNIFICANT RISK FACTORS:

1. **Market Risk**
    - It is controlled by the definition of an alpha factor — the sum of the alpha values is zero
        - *Remember*: alpha value is a single buy/short value for a single stock, for a single time period (ie: the investment weight)
    - An important assumption is that the betas (exposures of stocks to the markets) are all one (which is more or less true in aggregate, hence the assumption at individual level)
    - **Dollar neutral** (market neutral): subtract the mean from each alpha value in the vector so that its values sum to zero

2. **Sector Risk**
    - Neutralize sector exposure in basically the same way:
        - Subtract the sector mean from each weight so that the sum of the short weights in the sector are equal in magnitude to the sum of the long weights in the sector
            - ie: together they sum to zero
        - Again, it is assumed that each stocks beta to a sector is 1
    - eg: AAPL and Alphabet (GOOG)
        - Raw alphas: AAPL = 0.33; GOOG = 2.31
        - Sector mean: 1.32
        - Sector neutral:
            - AAPL: 0.33 - 1.32 = -0.99
            - GGOG: 2.31 - 1.32 = 0.99
        - Repeat this for all sectors in the universe

**** first neutralize by market, then by sector**

Ranking

Turnover: We have to take into account somehow that our portfolio weights, as determined by the alpha factor, would be changing daily and result in too high turnover.

Outliers: We also have to deal with outliers on any given day that might signal to buy a high exposure to a stock which may not be warranted. Clipping extreme large/negative data points can guard against overweighting outliers.
- Winsorizing: clipping at the 95th and 5th percentiles
- Another method would be to just set max position size limits

Ranking: eg: if both AAPL and GOOG alpha factors increase slightly by same amount, this would normally mean to buy more of both, but rank vs each other doesn't change, so this seems like unnecessary trading.
- Ranking is a way to handle this problem, ie: if AAPL alpha vector jumps while other 2 (rhs) each increase slightly, the ranks haven't changed
- So helps deal with small unimportant changes (noise) and large outlier changes

RANKING IN ZIPLINE

Explore the rank function
The Returns class inherits from zipline.pipeline.factors.factor.
The documentation for rank is located here and is also pasted below:

rank(method='ordinal', ascending=True, mask=sentinel('NotSpecified'), groupby=sentinel('NotSpecified'))[source] Construct a new Factor representing the sorted rank of each column within each row.

Parameters:
method (str, {'ordinal', 'min', 'max', 'dense', 'average'}) – The method used to assign ranks to tied elements. See scipy.stats.rankdata for a full description of the semantics for each ranking method. Default is 'ordinal'.

ascending (bool, optional) – Whether to return sorted rank in ascending or descending order. Default is True.

mask (zipline.pipeline.Filter, optional) – A Filter representing assets to consider when computing ranks. If mask is supplied, ranks are computed ignoring any asset/date pairs for which mask produces a value of False.

groupby (zipline.pipeline.Classifier, optional) – A classifier defining partitions over which to perform ranking.

Returns:
ranks – A new factor that will compute the ranking of the data produced by self.

Return type:
zipline.pipeline.factors.Rank

By looking at the documentation, and the link to **scipy.stats.rankdata** (also pasted below), which option for parameter `method` would we choose if we want unique ranks associated with each stock, even when the values are tied?

Note *When the documentation refers to "tied" values, it means instances where there are two alpha values for two different assets that are the same number, so there are different ways to handle the "tied" values when converting those values into ranks.*

'average': The average of the ranks that would have been assigned to all the tied values is assigned to each value.

'min': The minimum of the ranks that would have been assigned to all the tied values is assigned to each value. (This is also referred to as "competition" ranking.)

'max': The maximum of the ranks that would have been assigned to all the tied values is assigned to each value.

'dense': Like 'min', but the rank of the next highest element is assigned the rank immediately after those assigned to the tied elements.

'ordinal': All values are given a distinct rank, corresponding to the order that the values occur in a.

Z-Scoring

Subtract the mean from the numerator and divide by the standard deviation:

```
(row - row.mean()) / row.stddev()
```

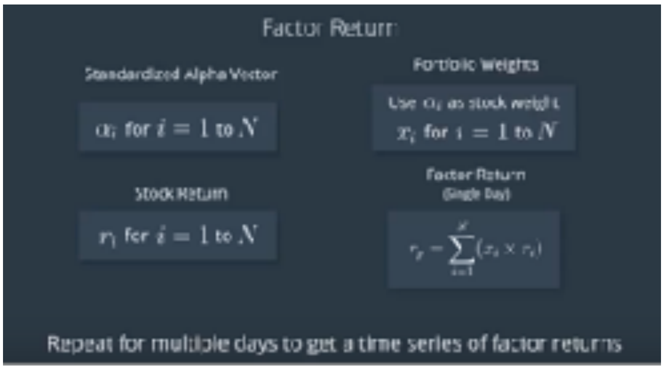
If comparing two factors from, say, different sized universes (eg: 100 stocks and 500 stocks), ranks will not be comparable, but z-scores will be.

However, z-scoring is not robust against outlier and noise like ranking is. So the best approach is to first conduct ranking, then z-scoring.

Documentation:
https://www.zipline.io/appendix.html#zipline.pipeline.factors.Factor.zscore

Evaluation Metrics

FACTOR RETURNS



UNIVERSE CONSTRUCTION RULE
Must not use a static list of stocks as universe. Must use universe as was available at the time in order to avoid look-ahead bias (ie: so not to exclude companies that went bankrupt etc. = survivorship bias).

One way to do this is to simply use the constituents of an index as provided historically by the index manager such as S&P500.

SHARPE RATIO

RANKED INFORMATION COEFFICIENT (RANK IC)
Tells us whether the ranks of our alpha values are correlated with our factor returns. So if a high factor weight ended up delivering a high return, its rank IC would be high.

	ABC	XYZ
Raw Alpha Vector	0.85	0.4
Ranks of Alpha Vectors	2	1
Returns	5%	-5%
Ranks of Returns	2	1

Rank IC = 1

<u>Calc</u>: rank the alpha factor for each time period; rank forward asset returns; if alpha factor rank and return rank for a stock are the same, then rank IC = 1, etc.
- Its called the Spearman Rank Correlation (different from the Pearson Correlation)

Pearson: cov of 2 variables rescaled by product of stdev of those 2 variables $\rho(X,Y) = \frac{\mathrm{Cov}(X,Y)}{\mathrm{Stdev}(X) \times \mathrm{Stdev}(Y)}$ or: $\rho(X,Y) = \sqrt[2]{R^2}$

Spearman: same except variable are converted to ranks first: $r_s(X,Y) = \frac{\mathrm{Cov}(r(X), r(Y))}{\mathrm{Stdev}(r(X)) \times \mathrm{Stdev}(r(Y))}$

INFORMATION RATIO
Same as Sharpe ratio except for active returns portion only, whereas SR is for all return components
- So IR is: **IR = np.sqrt(252) * mean(s) / stdev(s)**

*where: **s** = specific (aka idiosyncratic) returns*

FUNDAMENTAL LAW OF ACTIVE MANAGEMENT (GRINOLD'S LAW)
How to achieve high Sharpe ratios:

IR = IC * np.sqrt(B)

where: IR: information ratio
IC: information coefficient = skill
B: breadth = number of independent trading opportunities per year

When first coming to quant finance, people typically stop there. They spend all their time trying to increase the quality of a narrow set of forecasts. Of course this is important, but I'll let you in on something well known in the industry, but not so when known outside: the IC for even the best quants in the world is relatively not very high. In fact, **you might be surprised to learn that the count of *profitable trades as a ratio to all trades* (a measure similar in spirit to IC) by the best, well-known quants is typically *just a bit higher than 50%***. And if that's surprising, I'll tell you something else which might shock you: in general, **great discretionary investors have the advantage over great quants when it comes to IC. Quants however, have an advantage when it comes to breadth.**

Real World Constraints

LIQUIDITY
Bid-ask spreads can serve as a proxy to indicate liquidity. Less-liquid stocks wide spreads can really damage returns.

TRANSACTION COSTS
Market impact is usually larger than commissions.

TURNOVER
def: value of trades / portfolio value

At alpha research stage, turnover can be defined as the change in portfolio weights:

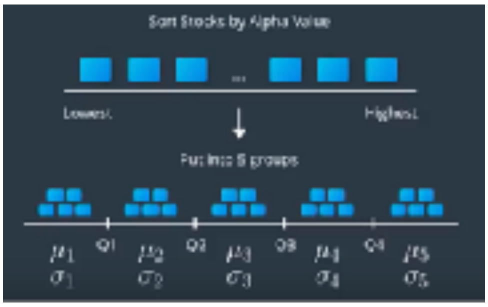
$$|x_{t1} - x_{t2}| = \left| \begin{pmatrix} x_{t1,1} \\ x_{t1,2} \\ \vdots \\ x_{t1,n} \end{pmatrix} - \begin{pmatrix} x_{t2,1} \\ x_{t2,2} \\ \vdots \\ x_{t2,n} \end{pmatrix} \right|$$

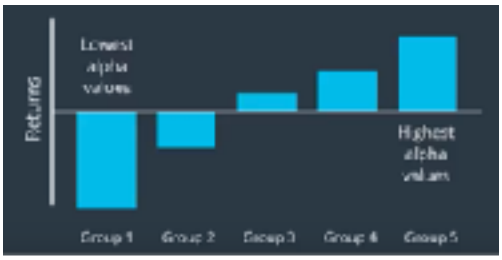
Factor Rank Autocorrelation
Another way to calculate portfolio turnover. When the ranks of the portfolio weights don't change much from day to day, the correlation to the next day's weights don't change much either, which means the factor rank autocorrelation is close to 1 and the turnover is low.

** this is different than rank ic
- Rank IC measures whether an alpha factor is correlated to future returns
- **FRA measures how stable factor ranks are from day-to-day**

Quantile Analysis
Want to see monotonic increase from lowest to highest, else could indicate factors are not reliable.

Academic papers separate raw factor returns into quantiles without ranking them first and are generally more interested in the tails to detect a broadly applicable





market phenomenon, and are less interested in applying the findings as a trading strategy.
- This means the findings may only apply to a small subset of stocks in the universe

The goal of a practitioner is to generate trade decisions for each stock in the portfolio, not just a small subset in the tails.

TRANSFER COEFFICIENT
Recall that we like to control for risk within the alpha vector during the alpha research stage so as to prevent significant changes in the portfolio weights after optimization.

Significant changes in weights after optimization affect the performance of the alpha vector. Remember, the alpha vector is the expectation of returns for a given stock. So if they change dramatically during optimization, the expected returns will be different.

The **transfer coefficient** measures how closely the optimized weights match the alpha vector weights.

transfer coefficient = corr(alpha vector, port weights X)

CONDITIONAL FACTORS (VID 28)
Multiplying 2 or more factors together (whenever you see AND in describing a string of factors)

Lesson 28: Alpha Factor Research Methods

Overnight Returns

Paper: Overnight Returns and Firm-Specific Investor Sentiment Abstract Authors:
We examine the suitability of using overnight returns *(def: from close to open)* to measure firm-specific investor sentiment by analyzing whether they possess characteristics expected of a sentiment measure. We document short-term overnight return persistence *(=momentum)*, consistent with existing evidence of short- term persistence in share demand of sentiment-influenced investors. We find that short-term persistence is stronger for harder-to-value firms, consistent with existing evidence that sentiment plays a larger role for such firms. We show that stocks with high (low) overnight returns underperform (outperform) over the longer-term, consistent with prior evidence of temporary sentiment-driven mispricing *(=mean-reversion)*. Overall, our evidence supports using overnight returns to measure firm-specific sentiment.

Notes
p 2, I: The recent work of Berkman, Koch, Tuttle, and Zhang (2012) suggests that a stock's overnight (close-to-open) return can serve as a measure of firm-level sentiment.

p 3, I: Specifically, Berkman et al. (2012) find that attention-generating events (high absolute returns or strong net buying by retail investors) on one day lead to higher demand by individual investors, concentrated near the open of the next trading day...This creates temporary price pressure at the open, resulting in elevated overnight returns that are reversed during the trading day.

p 3, I: We conduct three sets of analyses. In the first we test for short-run persistence in overnight returns. The basis for expecting this from a measure of sentiment is the evidence in Barber et al. (2009) that the order imbalances of retail investors, who are the investors most likely to exhibit sentiment, persist for periods extending over several weeks...In the third analysis we examine whether stocks with high overnight returns underperform those with low overnight returns over the long term.

Methodology for calculating weekly return:
Average daily returns over past 5 days * 5
(instead of just summing a given week's worth of daily returns)

— handles missing data (shortened trading weeks due to holidays etc) so that all weeks are comparable to each other

POTENTIAL FACTORS FROM PAPER
- Calculate overnight returns
- Aggregate weekly overnight returns
- Use weekly decile information for 1,2, 3, 4 weeks to test for Momentum
- Overweight(underweight) stocks with higher(lower) weekly overnight returns

Formation Process for Momentum Winners

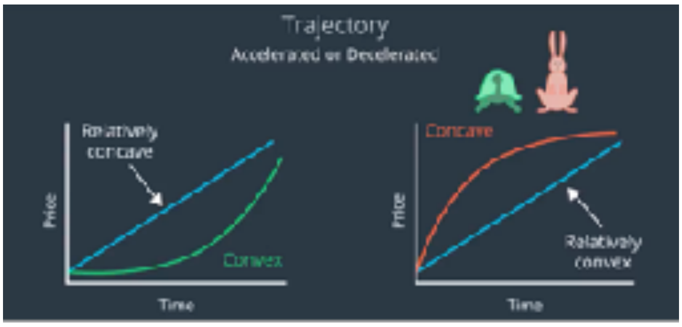
Abstract:
Previous studies have focused on which stocks are winners or losers but have paid little attention to the formation process of past returns. This paper develops a model showing that past returns

and the formation process of past returns have a joint effect on future expected returns. The empirical evidence shows that the zero-investment portfolio, including stocks with specific patterns of historical prices, improves monthly momentum profit by 59%. Overall, the process of how one stock becomes a winner or loser can further distinguish the best and worst stocks in a group of winners or losers.

Notes
p. 3: Intermediate-term (3–12 months) momentum has been documented by Jegadeesh and Titman (1993, 2001, hereafter JT), while short-term (weekly) and long-term (3–5 years) reversals have been documented by Lehmann (1990) and Jegadeesh (1990) and by DeBondt and Thaler (1985), respectively. Various models and theories have been proposed to explain the coexistence of intermediate-term momentum and long-term reversal. However, most studies have focused primarily on which stocks are winners or losers; they have paid little attention to how those stocks become winners or losers. This paper develops a model to analyze whether the movement of historical prices is related to future expected returns.

p. 4: This paper captures the idea that past returns and the formation process of past returns have a joint effect on future expected returns. We argue that how one stock becomes a winner or loser—that is, the movement of historical prices—plays an important role in momentum investing. Using a polynomial quadratic model to approximate the nonlinear pattern of historical prices, the model shows that as long as two stocks share the same return over the past n-month, the future expected return of the stock whose historical prices are convex shaped is not lower than one whose historical prices are concave shaped. In other words, when there are two winner (or loser) stocks, the one with convex-shaped historical prices will possess higher future expected returns than the one with concave-shaped historical prices.

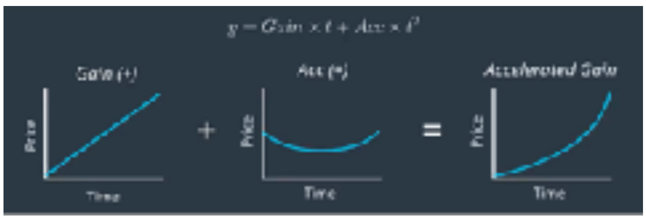
p. 4: To test the model empirically, we regress previous daily prices in the ranking period on an ordinal time variable and the square of the ordinal time variable for each stock. The coefficient of the square of the ordinal time variable is denoted as gamma.



Formula for convexity/concavity:

$$y = ax + bx^2 \quad \text{or as in the paper} \quad y = beta*x + gamma*x^2$$

In discussion of the coefficients, instructor calls beta ‘gain’ and gamma ‘acceleration’ for ease of understanding.



CREATING A JOINT FACTOR
Since we are deciding on long/short weights based on the combination of ‘gain’ and ‘acceleration’ (ie: momentum <u>and</u> convexity), we need to calculate these as a single resulting factor.

First, rank ‘gain’ and ‘acceleration’, then multiply the two together. A large number = a large long; a small number = large short

Expected Skewness and Momentum

Abstract:
Motivated by the time-series insights of Daniel and Moskowitz (2016), we investigate the link between expected skewness and momentum in the cross-section. The alpha of skewness-enhanced (-weakened) momentum is about twice (half) as large as the traditional alpha. These findings are driven by the short leg. Portfolio sorts, Fama-MacBeth regressions, and the market reaction to earnings announcements suggest that expected skewness is an important determinant of momentum. Due to the simplicity of the approach, its economic magnitude, its existence among large stocks, and the success of risk management, the results are difficult to reconcile with the efficient market hypothesis.

Notes
p1: In this paper, we comprehensively explore a new dimension in firm-level momentum profitability. More precisely, we document a strong relation between expected idiosyncratic skewness and cross-sectional momentum profits, in particular with respect to past loser stocks. The impact of skewness is economically large, statistically highly significant, holds among large firms, in international markets, and after controlling for a large set of firm characteristics previously linked to momentum profitability (e.g., past returns, idiosyncratic volatility, continuously arriving information, credit ratings).

p2: Based on this thought, momentum should be particularly pronounced if losers (winners) have a strong (weak) positive skew. Conversely, high (low) positive skewness on the winner (loser) leg is expected to reduce the profitability of momentum.

p3: As a **proxy for expected skewness**, our baseline analysis relies on the measure proposed by Bali et al. (2011) because of its simplicity, its economic persuasiveness, and its ability to predict realized skewness. It is **calculated as the maximum daily return during the preceding month**.

GENERAL COMMENTS ON ALPHA FACTORS
The aim is to try to capture signals (often imperceptible to humans):

1. Across many stocks
2. On a relative basis
3. And persistent across time

It is not the aim to get high conviction in any one particular stock. As per Grinold’s Law (of active fund management), our edge on a single stock should be low. Using breadth, we only need a signal that is marginally better than 50% correct and apply it across many stocks to generate a good Sharpe ratio.

SKEWNESS
Refers to the asymmetry in a distribution. Negative(positive) skewness is when it has a longer tail on the left(right), and its mean is to the left(right) of the median (ie: the majority of observations).

The first moment of a distribution is the mean, second is the variance, third is skewness, fourth is kurtosis (which measures the tails — stock returns typically exhibit excess kurtosis).

<u>Research paper definition/calc for skewness</u>: max daily return over past 20 trading days. More precisely, it is a measure of positive skew (because we have not measured the other tail with a min daily return).

Remember, the alpha vector is a relative ranking exercise.

4 scenario examples from the paper:

1. Momentum (+) and Skew (+): “weakened momentum”
   - +ve skew dampens outlook (mean reversion after jump)
2. Momentum (+) and Skew (less +): “enhanced momentum”
   - less +ve skew may indicate future returns may be stronger
3. Momentum (-) and Skew (+): “enhanced momentum”
   - people overly optimistic on a stock (-ve momo) which then suddenly bounces, but negative momentum reasserts itself
   - Is where the papers finds the biggest impact of skewness on momentum
4. Momentum (-) and Skew (less +): “weakened momentum”
   - Less positive skew has dampening affect on momentum, hence ‘weakened’

CONDITIONAL FACTOR CREATION
Use ranked momentum * ranked skew as a combination. Since skew is being used as mean reversion though, need to rank in reverse order.

Arbitrage Asymmetry and Idiosyncratic Volatility

Abstract
Many investors purchase stock but are reluctant or unable to sell short. Combining this arbitrage asymmetry with the arbitrage risk represented by idiosyncratic volatility (IVOL) explains the negative relation between IVOL and average return. The IVOL-return relation is negative among overpriced stocks but positive among underpriced stocks, with mispricing determined by combining 11 return anomalies. Consistent with arbitrage asymmetry, the negative relation among overpriced stocks is stronger, especially for stocks less easily shorted, so the overall IVOL-return relation is negative. Further supporting our explanation, high investor sentiment weakens the positive relation among underpriced stocks and, especially, strengthens the negative relation among overpriced stocks.

Notes
Here we use the famous value factor which is described in many places (e.g., Value and Momentum Everywhere **http://pages.stern.nyu.edu/~lpederse/papers/ValMomEverywhere.pdf)**: p. 8 "For individual stocks, we use the common value signal of the ratio of the book value of equity to market value of equity, or book-to-market ratio,"; in this study we use the Sharadar ratio Price/Book. We use this as an anomaly to create a refined factor as described in the "Arbitrage Asymmetry and the Idiosyncratic Volatility Puzzle". p. 2 This study presents an explanation for the observed negative relation be- tween IVOL and expected return. We start with the principle that [idiosyncratic volatility] IVOL rep- resents risk that deters arbitrage and the resulting reduction of mispricing. In keeping with previous literature, we refer to risk that deters arbitrage as arbi- trage risk.3 We then combine this familiar concept with what we term arbitrage asymmetry: many investors who would buy a stock they see as underpriced are reluctant or unable to short a stock they see as overpriced.

p. 2 Combining the effects of arbitrage risk and arbitrage asymmetry implies the observed negative relation between IVOL and expected return. To see this, first note that stocks with greater IVOL, and thus greater arbitrage risk, should be more susceptible to mispricing that is not eliminated by arbitrageurs [my emphasis added]. Among overpriced stocks, the IVOL effect in expected return should therefore be negative—those with the highest IVOL should be the most overpriced.

p. 11 We compute individual stock IVOL, following Ang et al. (2006), as the stan- dard deviation of the most recent month's daily benchmark-adjusted returns.

Points:

- Arbitrage supports efficient markets
- Volatility may limit arb activities
- Idiosyncratic vol may be more useful than total vol
- Combine IVOL and value factors together as an alpha factor

Just like returns, volatility (risk) can be broken into systematic and idiosyncratic (specific) components. Idiosyncratic risk may be a more useful measure of arbitrage risk (ie: the risk a losing money on an arb trade). This is because market participants seek to neutralize common market factor risks, leaving them bearing only idiosyncratic risk. Therefore it make more sense to only consider the idiosyncratic risks. It can be measured as the std deviation of the residual return (return not explained by market factors — paper uses Fama/French model of market, size, value). **This is IVOL** and is a proxy for arbitrage risk.

FUNDAMENTALS AND IVOL

Fundamental data is less responsive due to infrequent data points, so is often used as a conditioning factor along with price driven data, or to use as an alpha factor bu with low weight and high weight given to more responsive alphas. Since high IVOL indicates high idiosyncratic risk, can use as a conditioning factor with fundamental data, since IVOL by itself doesn't indicate to go in the long or short direction. The conditioning factor doesn't need to be an alpha factor on its own.

GENERALIZING THE VOLATILITY FACTOR

Try other approaches to academic paper, such as instead of using the Fama French risk model, try PCA. Or try combining IVOL with other factors.

Lesson 29: Advanced Portfolio Optimization

Need to add….