Using Python to Quantify Portfolio Diversification

Robin Warner

PyCon Canada 2018

THANK YOU DOUG & TEAGAN

Agenda

I. Background Theory

II. Data Set & Problem Space

III. Diversification Metrics

IV. Applications & Resources

I. Background Theory

What is an Investment?

An asset or item acquired with the goal of generating a profit over a given timeframe.

What is an Investment?

Invest Today

- Use today's expectations of tomorrow

Generate Profit

- Accumulate gain or loss

Consume Tomorrow

- Redeem the investment

What is required to invest?

Expectations of the future

(and the potential rewards available)

What is required to invest?

More specifically,

Expectations of future <u>**changes**</u> in the value of an investment

(i.e. the investment's **Expected Return**)

So what’s the problem?

The future is uncertain.

So what’s the problem?

Expected Returns ≠ Realized Returns

So, what is required to invest (again)?

Expectations of the future

(i.e. the investment's **Expected Return**)

...and a measure of how wrong we may be about the future

(i.e. the investment's **Risk**)

How to estimate an investment's characteristics

Expected Return = $\mathbf{E}(\boldsymbol{R_i})$

- Historical Average
- DCF Model
- Earnings Multiple
- Macro Economic Model
- ⋮

Risk = $\boldsymbol{\sigma_i}$

- Historical Volatility
- Historical Variance
- Historical Inter-quartile Range
- ⋮

How to measure an investment's characteristics

Expected Return = $\mathbf{E}(\boldsymbol{R_i})$

Historical Average

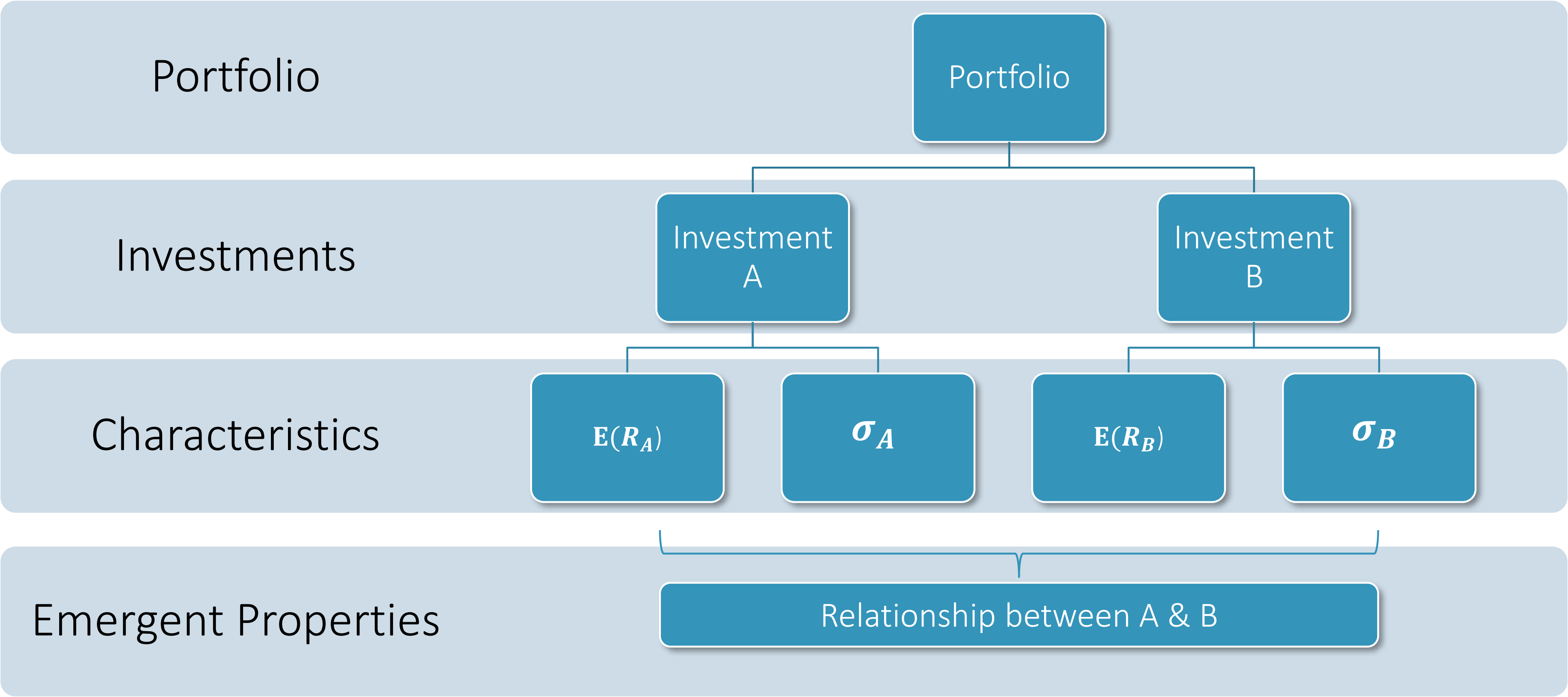
- 5 yr avg monthly return

Risk = $\boldsymbol{\sigma_i}$

Historical Volatility

- 5 yr avg monthly dispersion around the historical average

What is a Portfolio?



What is a Portfolio's Expected Return?

The Expected Return of a Portfolio:

$$\mathbf{E}\left(\boldsymbol{R}_{\boldsymbol{pf}}\right) = \sum_{\boldsymbol{i}} \boldsymbol{w}_{\boldsymbol{i}} \mathbf{E}(\boldsymbol{R}_{\boldsymbol{i}})$$

Where:

$\mathbf{E}\left(\boldsymbol{R}_{\boldsymbol{pf}}\right)$ – The expected return of the portfolio,
$\boldsymbol{w}_{\boldsymbol{i}}$ – The weights of each investment within the portfolio, and
$\mathbf{E}(\boldsymbol{R}_{\boldsymbol{i}})$– The expected return of each investment within the portfolio.

What is Portfolio Risk?

The Volatility of a Portfolio:

$$\sigma_{pf} = \sqrt{\sum_i w_i^2 \sigma_i^2 + \sum_i \sum_{j \neq i} w_i w_j \sigma_i \sigma_j \rho_{ij}}$$

Where:

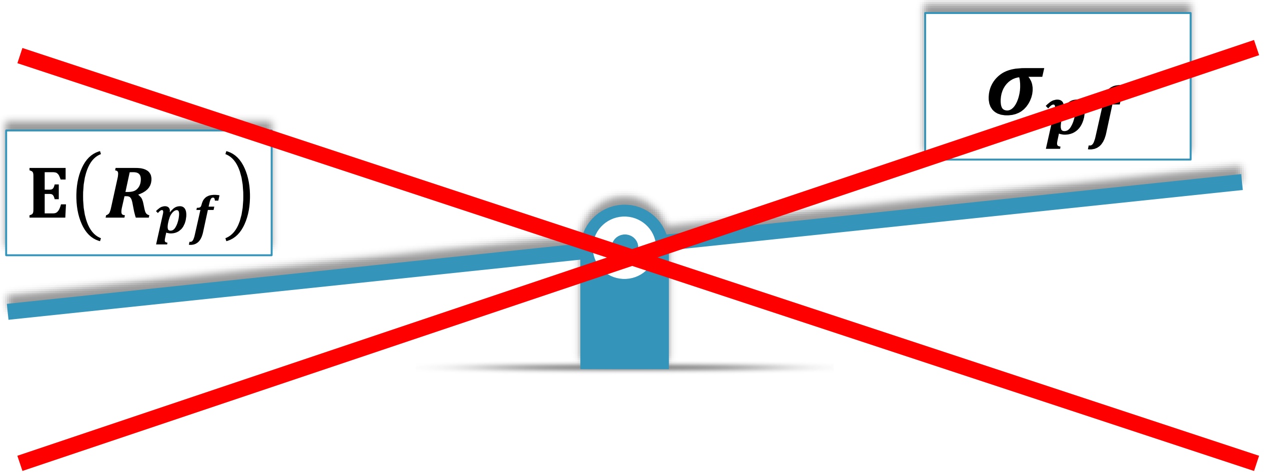
σ_{pf} – the volatility of the portfolio,

w_i , w_j –the weights of investments *i* and *j* respectively within the portfolio,

σ_i , σ_j – the volatilities of investments *i* and *j* respectively within the portfolio, and

ρ_{ij} – the correlation between investments *i's* and *j's* returns

How to manage both Risk & Reward?



Through Portfolio Efficiency

Portfolio Efficiency

A measure of how much reward a portfolio may produce for every unit of risk taken

Portfolio Efficiency

$$\boldsymbol{Risk\ Adjusted}\ \mathbf{E}(\boldsymbol{R}_{\mathbf{pf}}) = \mathbf{E}(\boldsymbol{R}_{\mathbf{pf}})/\boldsymbol{\sigma}_{\mathbf{pf}}$$

What is Portfolio Diversification?

Diversification is a risk management technique that aims to minimize the impact of any single investment's performance on that of the total portfolio.

Portfolio Efficiency & Diversification

How can one use diversification to achieve an efficient portfolio?

II. Data Set & Problem Space

Pandas & the Pandas DataFrame

What is Pandas?

- Popular Python library used for data manipulation and analysis
- Developed by Wes McKinney in the late 2000s
- Name derived from the term “panel data”
- Primary data structure known as the **DataFrame** object

Why use the DataFrame Object?

- Easy to use table object
- Lots of built in functionality including missing data handling, charting, statistics, & hierarchical labeling
- Similar to R’s data.frame object

The Data

The Standard & Poor's 500 Index (i.e. S&P 500):

American stock market index
500 largest companies by market capitalization listed on the NYSE or NASDAQ
More or less market cap weighted
Rich historical data available

DATE	AFL.N	AES.N	ABT.N	ABMD.OQ	ATVI.OQ	ADBE.OQ	AMD.OQ	AMG.N	APD.N	ALK.N	ALB.N	ALXN.OQ	HON.N	ALL.N	ARNC.N	HES.N	AEE.N	AEP.N	AXP.N	AIG.N	ABC.N	AME.N
2013-01-31	-0.00113	[illegible]	[illegible]	[illegible]	[illegible]	0.003981	0.083333	0.105878	0.040551	0.070534	-0.01304	0.001919	0.075107	0.092855	0.018443	0.268127	0.05599	0.061153	0.023138	0.071671	0.050718	0.09103
2013-02-28	-0.05842	[illegible]	[illegible]	[illegible]	[illegible]	0.038858	-0.04231	0.01598	-0.0125	0.117469	0.061491	-0.07714	0.027266	0.048292	-0.03622	-0.00983	0.041615	0.03312	0.056793	0.004758	0.040335	0.020493
2013-03-31	0.041233	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.090042	0.036577
2013-04-30	0.046521	[illegible]	[illegible]	[illegible]	[illegible]	0.036084	0.105882	0.01374	-0.00186	-0.03627	-0.02031	0.063599	-0.02403	0.003872	-0.00209	0.00796	0.035123	0.057578	0.014082	0.066976	0.051895	-0.06112
2013-05-31	0.023145	[illegible]	[illegible]	[illegible]	[illegible]	-0.04814	0.41844	0.053443	0.085707	-0.0782	0.092571	-0.00469	0.066876	-0.02071	0	-0.06608	-0.06097	-0.10908	0.10671	0.073394	-0.00074	0.059936
[illegible]	[illegible]	[illegible]	-0.04881	-0.00046	-0.01178	0.061757	0.02	-0.00037	-0.03002	-0.08483	-0.06919	-0.05434	0.011281	-0.00249	-0.08002	-0.01365	0.011751	-0.0227	-0.01255	0.005398	0.032359	-0.0197
[illegible]	[illegible]	[illegible]	0.050172	0.163265	0.26087	0.037752	-0.07598	0.100098	0.186393	0.176538	-0.0045	0.260082	0.045887	0.059435	0.016487	0.119868	0.039779	0.03506	-0.01324	0.018121	0.043704	0.09409
[illegible]	[illegible]	[illegible]	-0.09009	-0.061	-0.09232	-0.03236	-0.13263	-0.03343	-0.05984	-0.07453	0.005806	-0.07287	-0.04109	-0.06002	-0.03132	0.005238	-0.05585	-0.07659	-0.02521	0.020875	-0.02317	-0.0726
[illegible]	[illegible]	[illegible]	-0.0042	-0.19023	0.021446	0.135301	0.16208	0.047728	0.043313	0.10597	0.009139	0.077951	0.043605	0.054883	0.054273	0.033267	0.030464	0.01285	0.050202	0.046707	0.073436	0.072227
[illegible]	[illegible]	[illegible]	0.101235	0.257472	-0.0018	0.043512	-0.12105	0.081034	0.02294	0.128393	0.051636	0.058454	0.044447	0.049654	0.14184	0.049909	0.038462	0.080507	0.083157	0.062102	0.069231	0.039331
2013-11-30	[illegible]	0.034067	0.04487	0.192244	0.034255	0.047601	0.08982	0.014232	-0.00169	0.100198	0.038072	0.012607	0.020524	0.022804	0.036451	-0.00086	-0.00912	0.004697	0.0489	-0.03679	0.079596	0.029061
2013-12-31	[illegible]	-0.00412	0.003666	-0.06471	0.036026	0.054597	0.063187	0.083046	0.027134	-0.05608	-0.07743	0.068755	0.032269	0.004975	0.106432	0.023049	0.008647	-0.0068	0.057459	0.026131	-0.00312	0.070093
2014-01-31	[illegible]	-0.03101	-0.04357	0.028048	-0.03926	-0.01152	-0.1137	-0.08134	-0.05942	0.077678	0.012463	0.19292	-0.00154	-0.06124	0.082811	-0.09048	0.04646	0.044288	-0.06293	-0.06053	-0.04395	-0.0617
2014-02-28	[illegible]	-0.02916	0.085106	0.025464	0.129597	0.159149	0.081633	-0.05616	0.153909	0.095599	0.028202	0.113841	0.035167	0.059766	0.019699	0.06014	0.067918	0.028478	0.07363	0.03774	0.009372	0.077297
2014-03-31	[illegible]	0.046154	-0.03193	-0.07627	0.056331	-0.04183	0.080863	0.063813	-0.01881	0.077101	0.006516	-0.13954	-0.01778	0.042757	0.096212	0.035612	0.01955	0.009163	-0.01369	0.004822	-0.03331	-0.03287
2014-04-30	[illegible]	0.011905	0.005972	-0.09025	-0.02104	-0.06161	0.01995	-0.00925	-0.01272	0.008144	0.009335	0.0399	0.001518	0.006539	0.046648	0.075772	0.00267	0.062179	-0.02888	0.062388	-0.00625	0.023888
2014-05-31	[illegible]	-0.02422	0.032783	-0.03757	0.038481	0.046199	-0.022	-0.04844	0.020801	0.046556	0.03207	0.051327	0.002707	0.023003	0.010565	0.024002	-0.04745	-0.00855	0.046552	0.017692	0.122737	0.006829
2014-06-30	[illegible]	0.102837	0.022244	0.102632	0.073147	0.121165	0.0475	0.089077	0.072131	-0.03453	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.009432	-0.00711	-0.01507
2014-07-31	[illegible]	-0.06045	0.029829	0.018298	0.003587	-0.04491	-0.06683	-0.02994	0.025902	-0.0749	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	-0.04764	0.058492	-0.06867
2014-08-31	[illegible]	0.039014	0.002849	0.017578	0.051832	0.04037	0.066496	0.059724	0.009509	0.0539	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.078492	0.006241	0.087287
2014-09-30	[illegible]	-0.06588	-0.01539	-0.04683	-0.11682	-0.03769	-0.18225	-0.0511	-0.02274	-0.06042	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	-0.03639	-0.00116	-0.05157
2014-10-31	[illegible]	-0.00776	0.048088	0.32058	-0.0404	0.013441	-0.17889	-0.00284	0.0344	0.222554	-0.00883	0.154022	0.032185	0.056705	0.041448	-0.10083	0.104618	0.11741	0.02753	-0.00833	0.104916	0.038638
2014-11-30	0	-0.01421	0.021106	0.083257	0.085213	0.05077	-0.00357	0.01902	0.068118	0.108961	0.011305	0.018499	0.030763	0.050887	0.031573	-0.14008	0.018186	-0.01354	0.02746	0.022961	0.066034	-0.02282
2014-12-31	[illegible]	-0.00721	0.011458	0.071509	-0.06928	-0.0133	-0.04301	0.042487	0.002783	0.012367	0.018462	-0.05064	0.008527	0.030814	-0.08668	0.012203	0.070053	0.055083	0.006709	0.02208	-0.00977	0.032771
2015-01-31	[illegible]	-0.11256	-0.00578	0.359432	0.037221	-0.03535	-0.03745	-0.03166	0.0096	0.13571	-0.19741	-0.00967	-0.02164	-0.00655	-0.00873	-0.08575	-0.01843	0.03442	-0.13274	-0.12748	0.054237	-0.08987
2015-02-28	[illegible]	0.061375	0.058311	0.174913	0.115789	0.127905	0.210117	0.053036	0.072278	-0.06218	0.172192	-0.01566	0.051337	0.011606	-0.05511	0.112461	-0.06338	-0.08327	0.011154	0.132187	0.081115	0.109395
2015-03-31	[illegible]	-0.00925	-0.02195	0.177496	-0.0253	-0.06523	-0.13826	-0.00758	-0.0311	0.039749	-0.06594	-0.0392	0.014972	0.008074	-0.12628	-0.09603	-0.00495	-0.0231	-0.04253	-0.00976	0.10617	-0.01129
2015-04-30	[illegible]	0.031128	0.001943	-0.11679	0.00396	0.028672	-0.15672	0.052845	-0.05191	-0.03203	0.129826	-0.02349	-0.03249	-0.02122	0.038541	0.133048	-0.02986	0.011022	-0.00858	0.027377	0.005542	-0.00228
2015-05-31	[illegible]	0.026415	0.046963	-0.05536	0.106924	0.039837	0.00885	-0.01092	0.023228	0.009054	0.007538	-0.03185	0.032486	-0.03359	-0.06859	-0.12198	-0.01734	-0.0102	0.029309	0.041215	-0.01522	0.025563
2015-06-30	-0.00032	-0.025	0.009877	0.100636	-0.04157	0.024276	0.052632	-0.02262	-0.06766	-0.00325	-0.08113	0.103333	-0.02143	-0.03639	-0.10779	-0.00948	-0.06339	-0.05898	-0.02509	0.054769	-0.05526	0.018973

~500 Stocks Columns

60 Date Rows

This will be our raw DataFrame used for the analysis

S&P 500 Monthly Returns

From 2013-01-31 to 2017-12-31

Analysis Setup & Process

Investment Universe

- A collection of investable instruments and their characteristics from which portfolios may be built.

Portfolio Generator

- A function with the ability to randomly sample an investment universe and produce portfolios.

Portfolio Universe

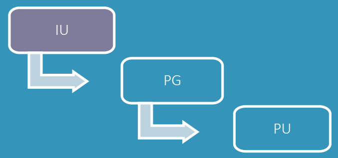
- A collection of portfolios and their characteristics on which analysis may be done.

```python
class InvestmentUniverse:
    def __init__(self, universe):

        self.universe = universe
        self.tickers = pd.DataFrame(data=universe.columns,
                                    index=pd.RangeIndex(1, universe.T.count()[0]+1),
                                    columns=['Ticker'])
        self.dates = pd.DataFrame(data=universe.index.values,
                                  index=pd.RangeIndex(1, universe.count()[0]+1),
                                  columns=['Date'])

        self.returns = universe

        self.expected_returns = pd.DataFrame(data=self.returns.mean(),
                                             columns=['ExpectedReturn'])

        self.variances = pd.DataFrame(data=self.returns.var(),
                                      columns=['Variance'])

        self.volatilities = pd.DataFrame(data=self.returns.std(),
                                         columns=['Volatility'])

        self.correl_mtx = self.returns.corr()

        self.covar_mtx = self.returns.cov()

        self.covar_mtx_diag = pd.DataFrame(data=np.diag(self.returns.var()),
                                           index=self.tickers['Ticker'].values,
                                           columns=self.tickers['Ticker'].values)

        self.covar_mtx_offdiag = self.covar_mtx - self.covar_mtx_diag

        self.volsumprod_mtx = self.volatilities.dot(self.volatilities.T)

    def get_ticker_count(self):
        return self.tickers.count()[0]

    def get_date_count(self):
        return self.dates.count()[0]
```

Investment Characteristics

Step1: Investment Universe

For this presentation, **universe** is the S&P 500 Monthly Returns DataFrame.



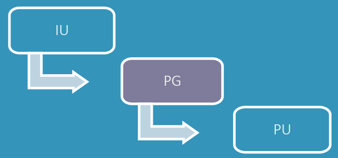
```python
def portfolio_generator(investment_universe,
                        position_list,
                        number_of_simulations):

    row_index = investment_universe.tickers.index
    columns = tuple(itertools.product(position_list, range(1, number_of_simulations+1)))
    row_index_names = ['TickerID']
    columns_names = ['NumOfPosn', 'SimID']

    portfolio_generation_mtx = blank_data_frame(len(row_index),
                                                len(columns),
                                                row_index,
                                                columns,
                                                row_index_names,
                                                columns_names)

    for j in range(len(position_list)):
        for i in range(1, number_of_simulations+1, 1):
            pf_sample = list(random.sample(list(row_index), position_list[j]))
            for k in range(len(pf_sample)):
                portfolio_generation_mtx[position_list[j],i][pf_sample[k]] = 1

    portfolio_generation_mtx.index = investment_universe.tickers['Ticker']

    return portfolio_generation_mtx
```

Random sampler

Step2: Portfolio Generator

For this presentation, **position_list** is set as [1,2,5,10,20,30,40,50,100,150, 200,300,400,481] and **number_of_simulations** is set at 400.

Therefore the total number of portfolios generated is 5600.

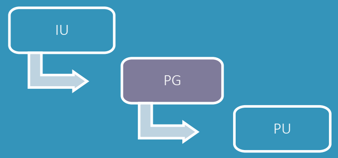


5600 Portfolio Columns

~500 Stocks Rows

NumOfPosn	1	2	5	10	20	30	40	50	100	150	200	300	400	481	1	2	5	10	20	30	40	50
SimID					1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2
Ticker																						
AFL.N					0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0
AES.N					0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
ABT.N					0	0	0	0	1	0	1	0	1	1	0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	1	0	0	1	1	1	0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	1	0
		0	0	0	0	0	0	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0
APD.N	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	0	0	0	0	0	0	0
ALK.N	0	0	1	0	0	0	0	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0
ALB.N	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0	0	0	0	1	0	0
ALXN.OQ	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	1	0
HON.N	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0
ALL.N	0	0	0	0	0	0	1	1	1	0	1	1	0	1	0	0	0	0	0	0	0	0
ARNC.N	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
HES.N	0	0	0	0	0	0	0	1	0	1	0	1	1	1	0	0	0	0	0	0	0	0
AEE.N	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
AEP.N	0	0	0	0	0	0	0	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0
AXP.N	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	1
AIG.N	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
ABC.N	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1
AME.N	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0
AMGN.OQ	0	0	0	0	0	0	0	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0
APH.N	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
APC.N	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	1	0
ADI.OQ	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	1	0	0
AON.N	0	0	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0
APA.N	0	0	0	0	0	0	0	0	1	0	1	0	1	1	0	0	0	0	0	0	0	0
AIV.N	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1	1	0
AAPL.OQ	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0	0	0	0	0	0	0	0

Portfolio Generation Matrix



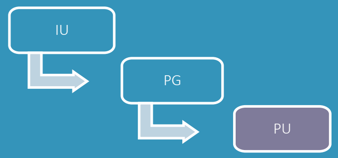
```python
class PortfolioUniverse:
    def __init__(self,
                 investment_universe,
                 position_list,
                 number_of_simulations):

        self.investment_universe = investment_universe
        self.position_list = position_list
        self.number_of_simulations = number_of_simulations
        self.portfolio_generation_mtx = portfolio_generator(self.investment_universe,
                                                            self.position_list,
                                                            self.number_of_simulations)

        def get_number_of_different_positions(self):
            return len(self.position_list)

        self.number_of_different_positions = get_number_of_different_positions

        def get_number_of_portfolios(self):
            return self.number_of_different_positions*self.number_of_simulations

        self.number_of_portfolios = get_number_of_portfolios(self)

        self.weights = self.portfolio_generation_mtx/self.portfolio_generation_mtx.sum()

        def get_pf_returns(self):
            pf_returns = blank_data_frame(self.investment_universe.get_date_count(),
                                          self.number_of_portfolios,
                                          self.investment_universe.dates['Date'].values,
                                          self.portfolio_generation_mtx.columns,
                                          ['Date'],
                                          self.portfolio_generation_mtx.columns.names)

            for j in self.weights.columns.levels[0]:
                for i in self.weights.columns.levels[1]:
                    pf_returns[j,i] = self.weights[j,i].dot(self.investment_universe.returns.T)

            return pf_returns

        self.returns = get_pf_returns(self)
```

Portfolio Weights

Step 3: Portfolio Universe

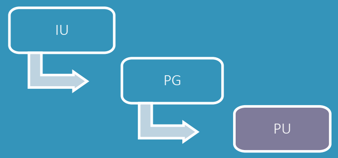
Note: Each investment held by a portfolio within the portfolio universe is equally weighted.

This means that their respective weight equals 1/n where n represents the number of investments within the portfolio.



```python
def portfolio_expected_return(pf_weight, investment_universe):
    return pf_weight.dot(investment_universe.expected_returns)

def portfolio_variance(pf_weight, investment_universe):
    return pf_weight.dot(investment_universe.covar_mtx.dot(pf_weight))

def portfolio_volatility(pf_weight, investment_universe):
    return portfolio_variance(pf_weight, investment_universe)**0.5

def portfolio_riskadj_expected_return(pf_weight, investment_universe):
    return portfolio_expected_return(pf_weight, investment_universe)/portfolio_volatility(pf_weight, investment_universe)

def portfolio_variance_term1(pf_weight, investment_universe):
    return pf_weight.dot(investment_universe.covar_mtx_diag.dot(pf_weight))

def portfolio_variance_term2(pf_weight, investment_universe):
    return pf_weight.dot(investment_universe.covar_mtx_offdiag.dot(pf_weight))

def portfolio_undiversified_variance(pf_weight, investment_universe):
    return pf_weight.dot(investment_universe.volsumprod_mtx.dot(pf_weight))

def portfolio_undiversified_volatility(pf_weight, investment_universe):
    return portfolio_undiversified_variance(pf_weight, investment_universe)**0.5
```

Portfolio Metrics

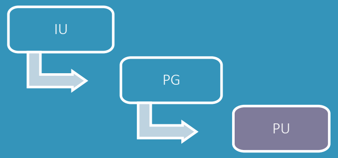


SimID	NumOfPosn	expected_returns	volatilities	riskadj_expected_returns	variances	variances_term1	variances_term2	undiversified_variances	undiversified_volatilities
1	1	0.01502	0.04583	0.32773	0.00210	0.00210	0.00000	0.00210	0.04583
1	2	0.01158	0.04916	0.23562	0.00242	0.00211	0.00031	0.00363	0.06026
1	5	0.01630	0.04228	0.38541	0.00179	0.00102	0.00077	0.00472	0.06871
1	10	0.02124	0.05092	[illegible]	[illegible]	[illegible]	0.00145	0.00840	0.09165
[illegible]	[illegible]	[illegible]	0.03134	[illegible]	[illegible]	[illegible]	0.00076	0.00409	0.06394
[illegible]	[illegible]	[illegible]	0.02617	[illegible]	[illegible]	[illegible]	0.00057	0.00329	0.05732
[illegible]	[illegible]	[illegible]	0.02769	[illegible]	[illegible]	[illegible]	0.00064	0.00434	0.06590
[illegible]	[illegible]	[illegible]	0.03041	0.44383	0.00092	0.00009	0.00084	0.00421	0.06491
1	[illegible]	0.01354	0.02980	0.45429	0.00089	0.00005	0.00084	0.00416	0.06450
1	[illegible]	0.01355	0.02989	0.45314	0.00089	0.00003	0.00086	0.00428	0.06546
1	[illegible]	0.01328	0.02871	0.46272	0.00082	0.00002	0.00080	0.00397	0.06303
1	[illegible]	0.01279	0.02768	0.46221	0.00077	0.00002	0.00075	0.00407	0.06383
1	[illegible]	0.01330	0.02882	0.46133	0.00083	0.00001	0.00082	0.00416	0.06451
1	[illegible]	0.01351	0.02870	0.47076	0.00082	0.00001	0.00081	0.00415	0.06445
2	[illegible]	0.00178	0.06007	0.02966	0.00361	0.00361	0.00000	0.00361	0.06007
2	[illegible]	-0.00091	0.04623	-0.01961	0.00214	0.00215	-0.00001	0.00422	0.06493
2	[illegible]	0.01952	0.03395	0.57478	0.00115	0.00076	0.00039	0.00373	0.06111
2	[illegible]	0.01367	0.03501	0.39043	0.00123	0.00049	0.00073	0.00435	0.06594
2	[illegible]	0.01593	0.03170	0.50238	0.00101	0.00022	0.00078	0.00409	0.06398
2	[illegible]	0.01397	0.03009	0.46437	0.00091	0.00013	0.00077	0.00381	0.06170
2	[illegible]	0.01312	0.03285	0.39932	0.00108	0.00015	0.00093	0.00477	0.06906
2	[illegible]	0.01418	0.02733	0.51867	0.00075	0.00011	0.00064	0.00431	0.06566
2	[illegible]	0.01285	0.02824	0.45498	0.00080	0.00005	0.00075	0.00430	0.06560
2	[illegible]	0.01400	0.02870	0.48789	0.00082	0.00003	0.00079	0.00434	0.06585
2	[illegible]	0.01293	0.02852	0.45331	0.00081	0.00002	0.00079	0.00392	0.06260
2	[illegible]	0.01373	0.02922	0.46999	0.00085	0.00002	0.00084	0.00421	0.06485
2	[illegible]	0.01345	0.02876	0.46767	0.00083	0.00001	0.00082	0.00412	0.06419
2	481	0.01351	0.02870	0.47076	0.00082	0.00001	0.00081	0.00415	0.06445
3	1	0.01093	0.06113	0.17875	0.00374	0.00374	0.00000	0.00374	0.06113
3	2	0.01724	0.04492	0.38381	0.00202	0.00177	0.00025	0.00354	0.05948
3	5	0.01300	0.04105	0.31659	0.00169	0.00089	0.00079	0.00430	0.06561
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

5600 Portfolio Rows

Portfolio Metrics

Portfolio Metrics



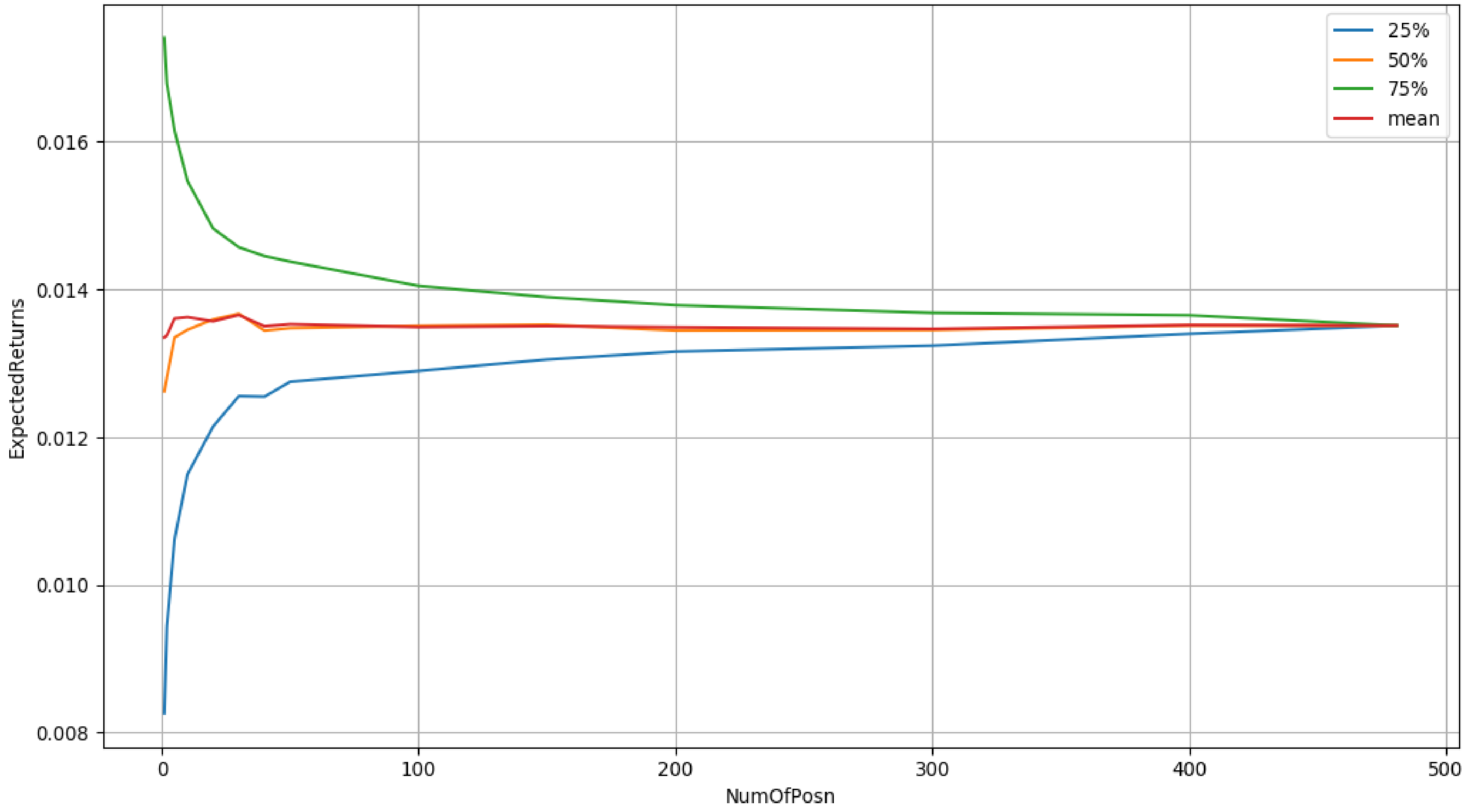
Recap: Portfolio Efficiency & Diversification

How can one use diversification to achieve an efficient portfolio?

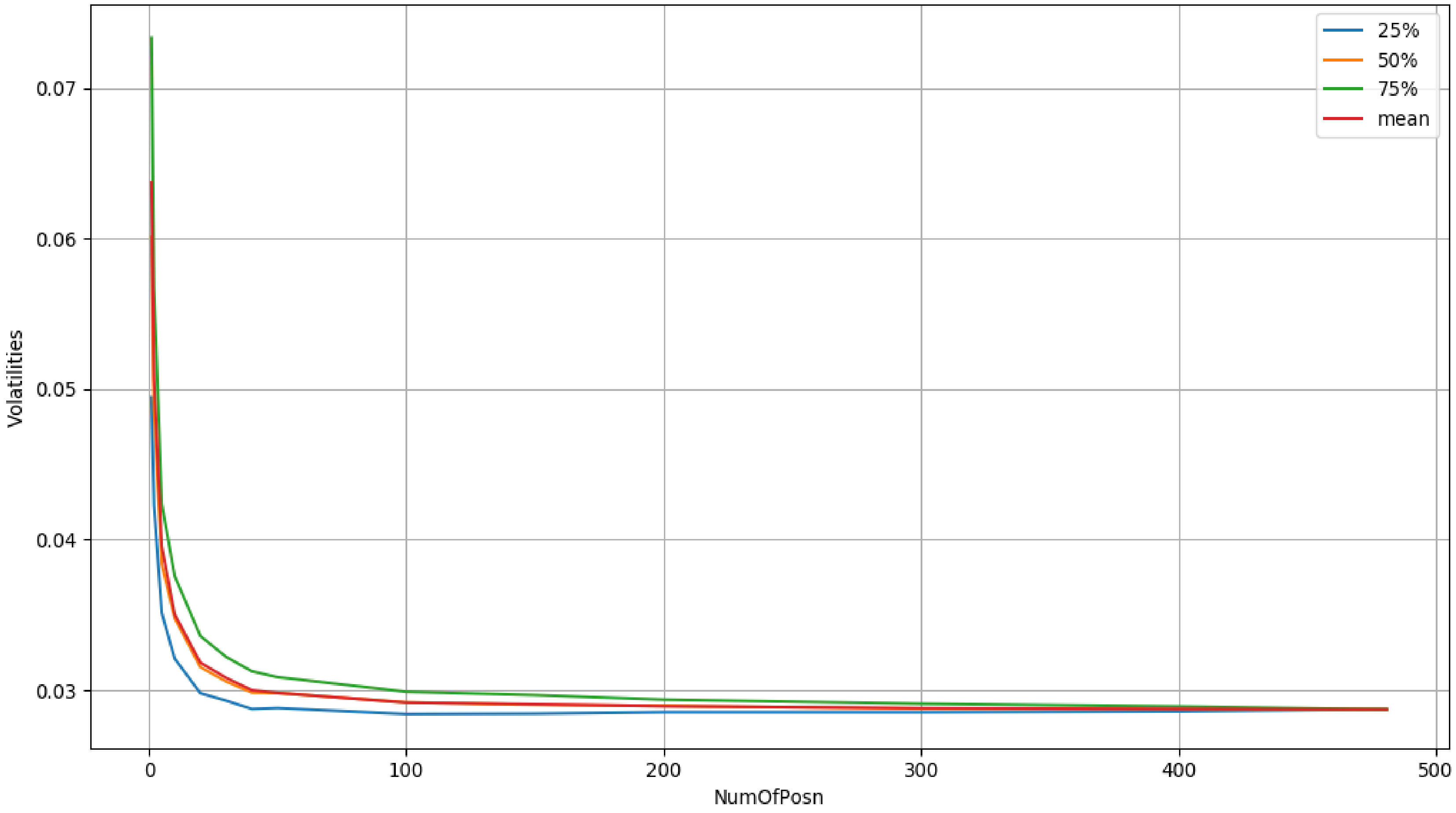
III. Diversification Metrics

Portfolio Efficiency & Diversification

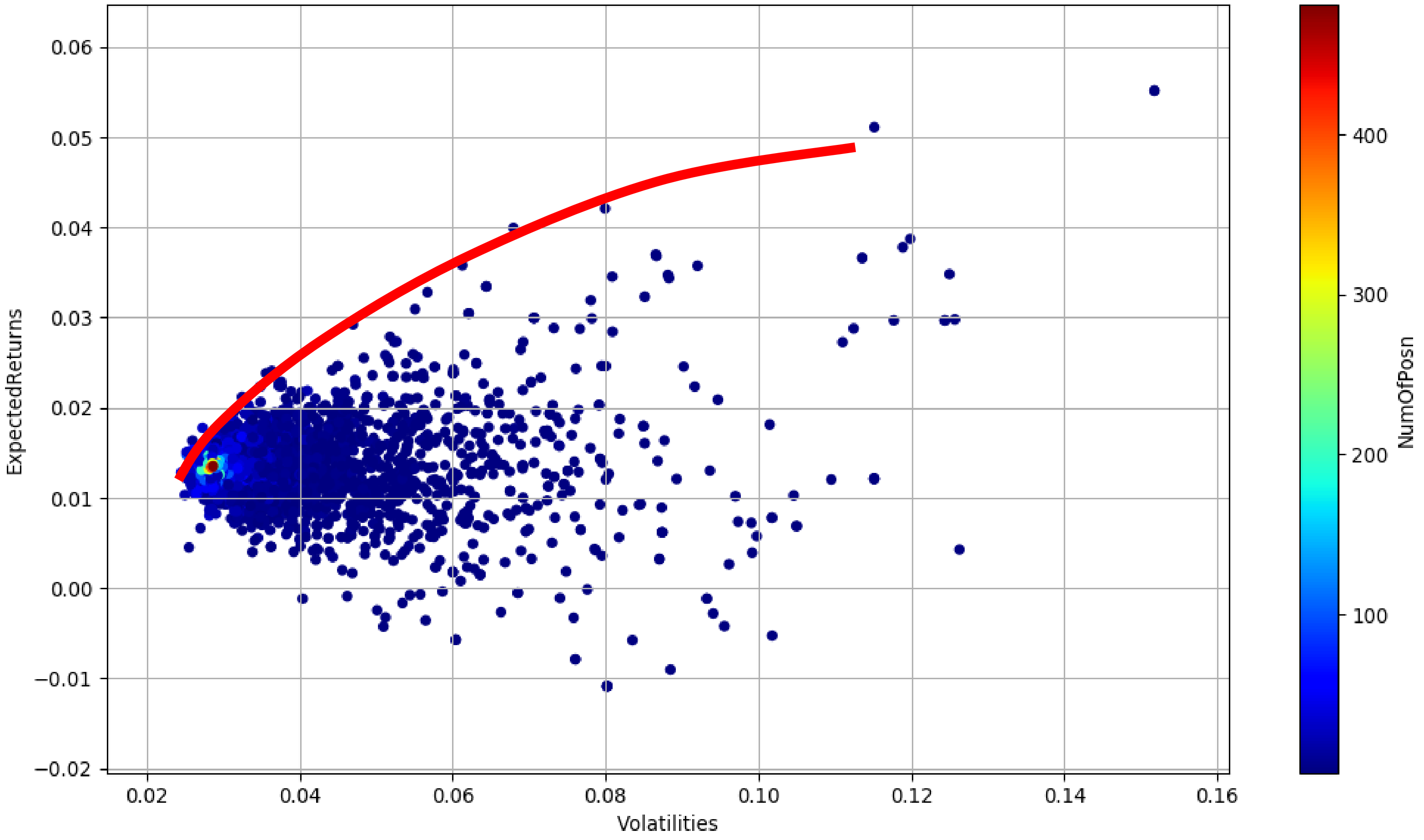
What happens to a portfolio's risk & return profile when you increase its number of investments?



Expected Return vs. Number of Positions in the Portfolio



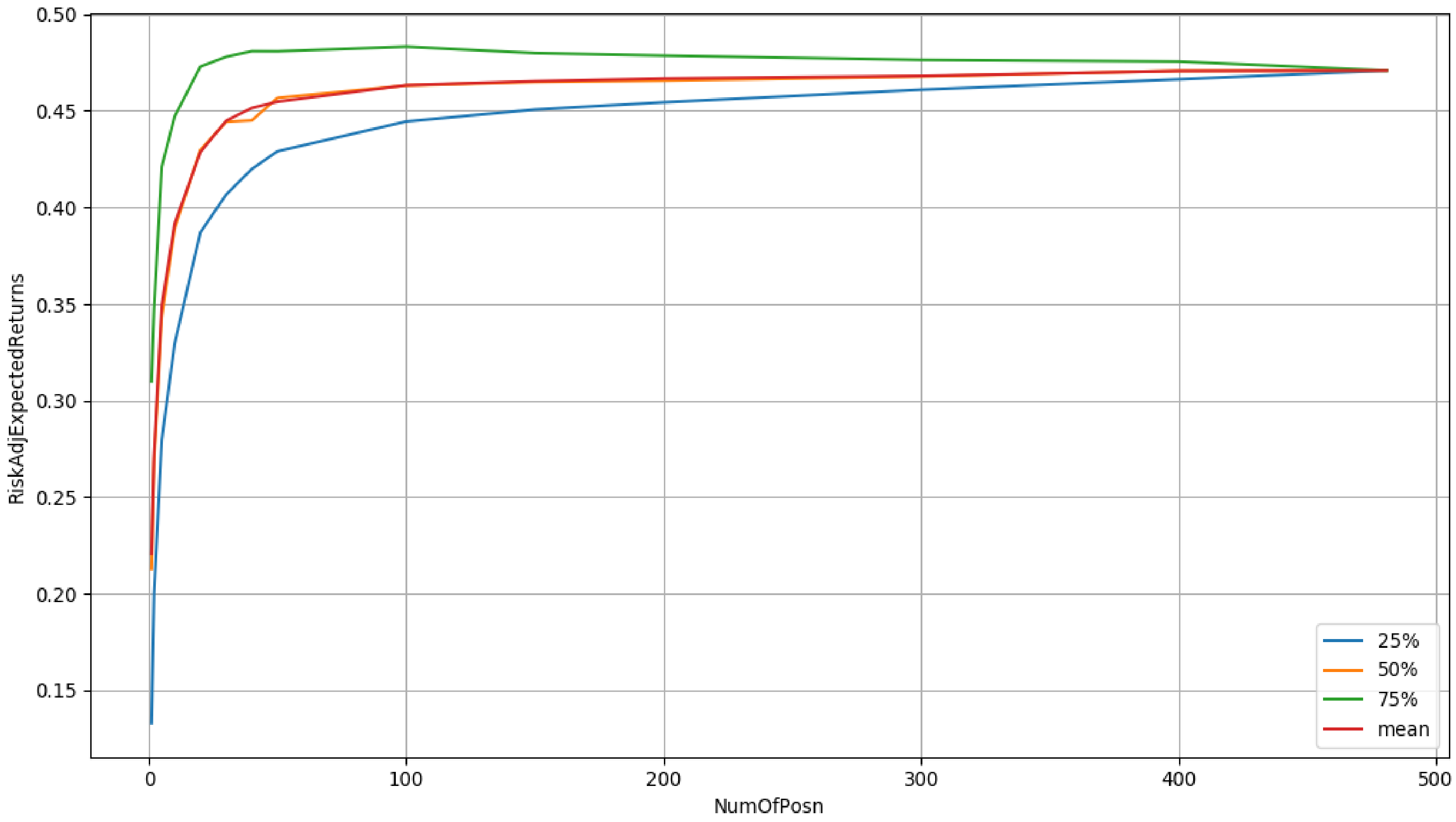
Volatility vs. Number of Positions in the Portfolio



Expected Return vs. Volatility

Portfolio Efficiency & Diversification

What about a portfolio's efficiency as you increase its number of investments?



Risk Adjusted Expected Return vs. Number of Positions in the Portfolio

Two Observations:

1. As the number of investment ↑ → Portfolio efficiency ↑

2. However, it does so at a diminishing marginal rate…

How can diversification be used to target optimal efficiency?

$$\boldsymbol{Risk\ Adjusted}\ \mathbf{E}(\boldsymbol{R}_{\mathbf{pf}}) = \boxed{\mathbf{E}(\boldsymbol{R}_{\mathbf{pf}})}\ /\ \boxed{\boldsymbol{\sigma}_{\mathbf{pf}}}$$

Unaffected

Lowered

How can diversification be used to target optimal efficiency?

$$\sigma_{pf}^2 = \boxed{\sum_i w_i^2 \sigma_i^2} + \sum_i \sum_{j \neq i} w_i w_j \sigma_i \sigma_j \boxed{\rho_{ij}}$$

No relationship between investments

“Emergent Properties”

Diversification Metrics

1. Diversification Ratio

2. Average Correlation

3. Diversification Index

1. Portfolio Diversification Ratio

$$\sigma_{pf} = \sqrt{\sum_i w_i^2 \sigma_i^2 + \sum_i \sum_{j \neq i} w_i w_j \sigma_i \sigma_j \boxed{\rho_{ij}}}$$

Set equal to 1

1. Portfolio Diversification Ratio

$$\boldsymbol{\sigma_{pf} = \sum_i w_i \sigma_i}$$

1. Portfolio Diversification Ratio

$$Undiversified\ PF\ Volatility = \sum_i w_i \sigma_j$$

$$PF\ Diversification\ Ratio = [Undiversified\ PF\ Volatility] / [PF\ Volatility]$$

```
def portfolio_diversification_ratio(pf_volatility, pf_undiversified_volatility):
    return pf_undiversified_volatility/pf_volatility
```

1. Portfolio Diversification Ratio

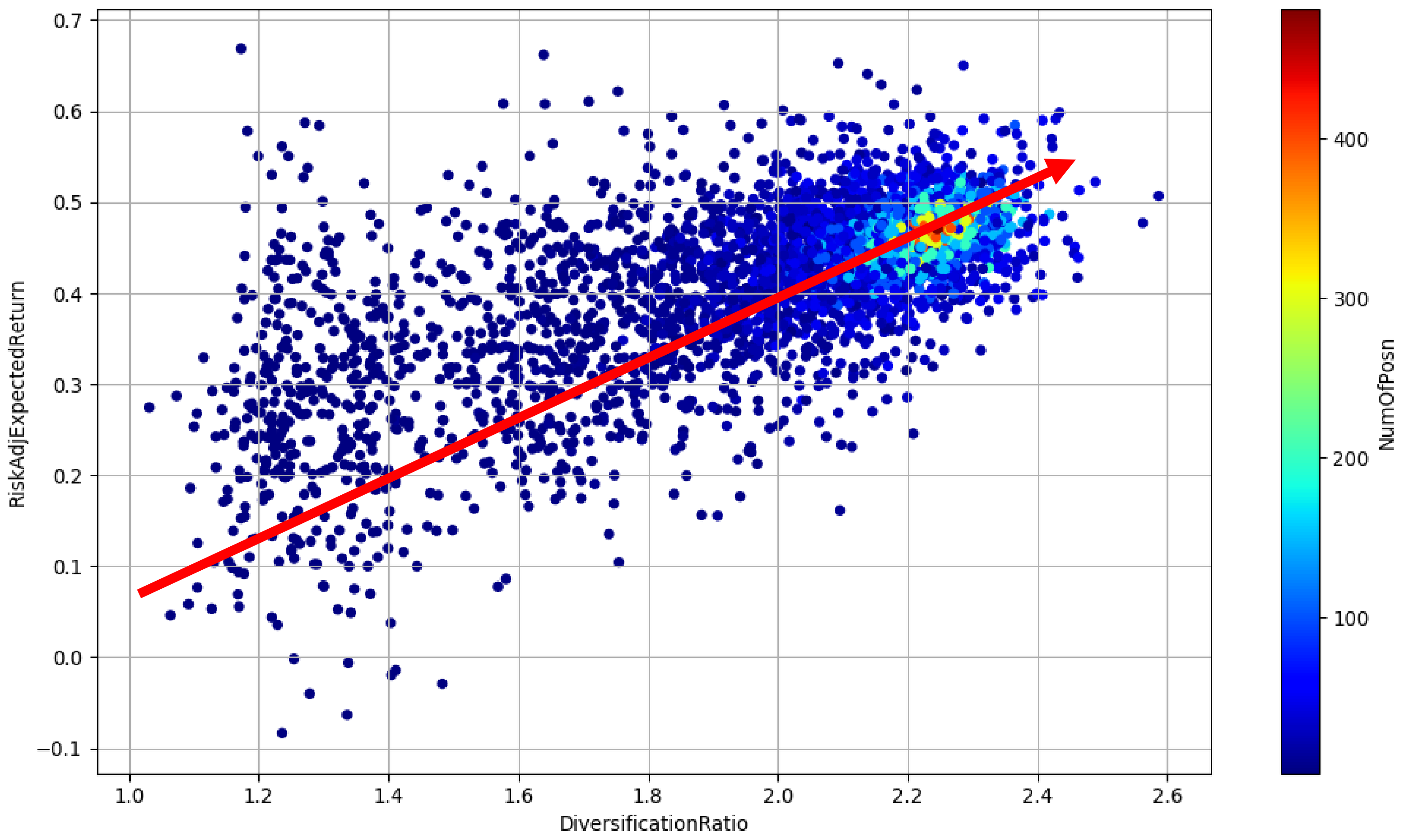
$$\boldsymbol{Diversification\ Ratio \cong 1}$$

Portfolio has little diversification properties

1. Portfolio Diversification Ratio

$$\boldsymbol{Diversification\ Ratio > 1}$$

Portfolio benefits from diversification among its investments



Risk Adjusted Expected Return vs. Diversification Ratio

1. DR
2. AC
3. DI

2. Portfolio Average Correlation

$$\sigma_{pf} = \sqrt{\sum_i w_i^2 \sigma_i^2 + \sum_i \sum_{j \neq i} w_i w_j \sigma_i \sigma_j \boxed{\rho_{ij}}}$$

Solve for single rho

2. Portfolio Average Correlation

$$PF\ Average\ Correlation = [PF\ Variance\ - PF\ Variance\ Term\ 1] / [Undiversified\ PF\ Variance\ - PF\ Variance\ Term\ 1]$$

```python
def portfolio_avg_correlation(pf_size, pf_variance, pf_variance_term1, pf_undiversified_variance):
    if pf_size == 1:
        portfolio_avg_correlation = 1
    else:
        portfolio_avg_correlation = (pf_variance - pf_variance_term1)/(pf_undiversified_variance - pf_variance_term1)

    return portfolio_avg_correlation
```

2. Portfolio Average Correlation

$$|\boldsymbol{Average\ Correlation}| \cong \mathbf{1}$$

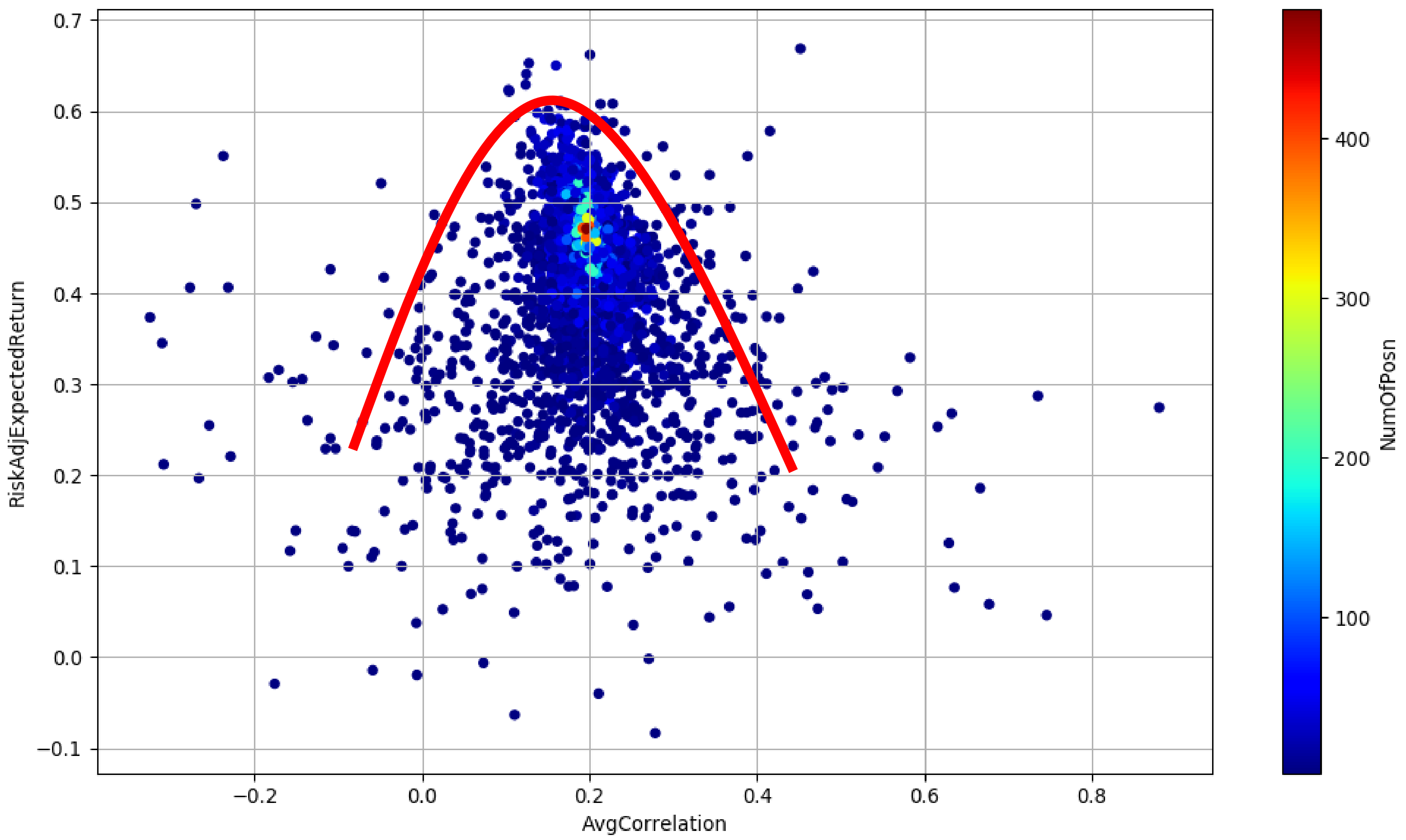
Portfolio has little diversification properties

2. Portfolio Average Correlation

$$|\boldsymbol{Average\ Correlation}| \cong \mathbf{0}$$

Portfolio benefits from diversification among its investments

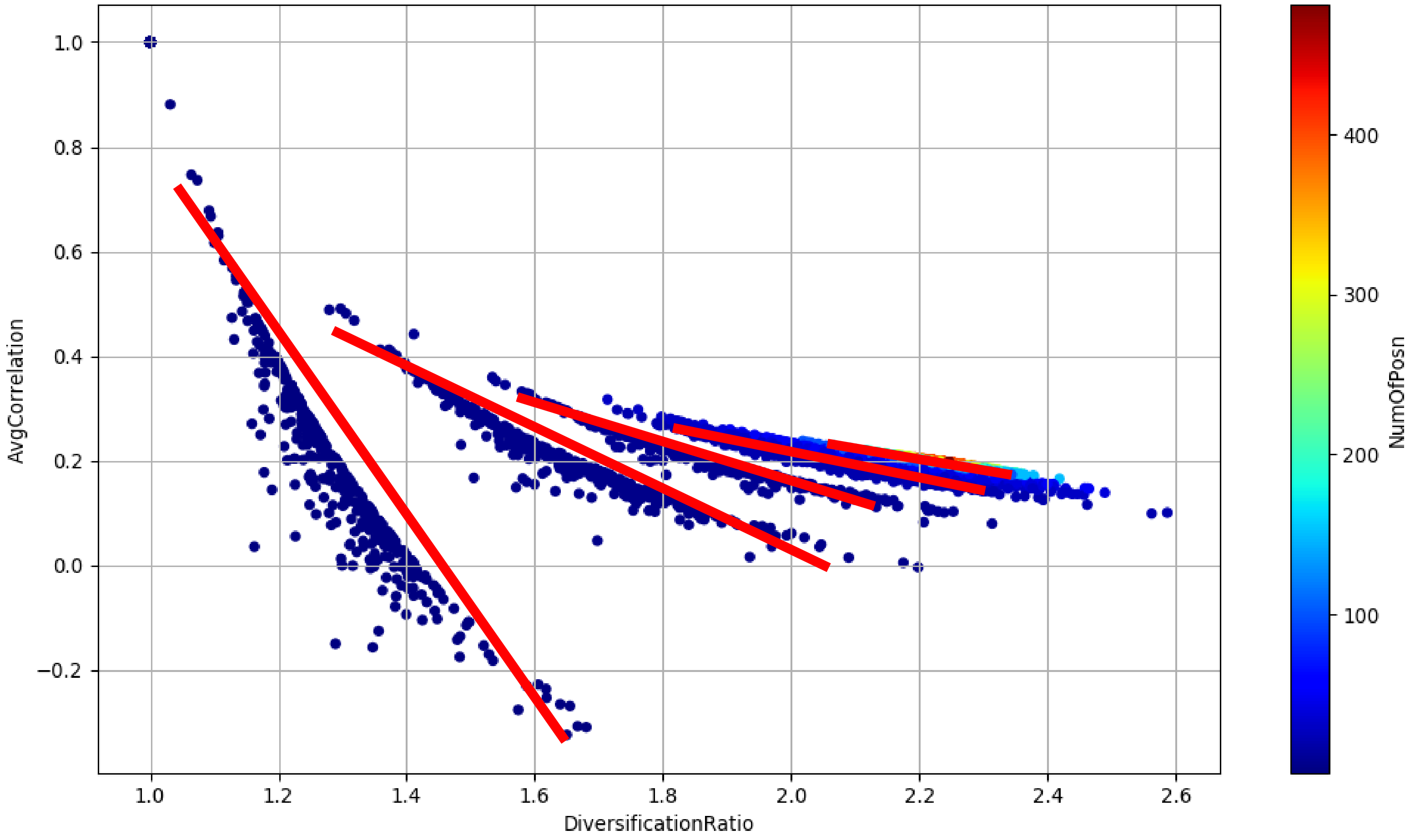
1. DR
2. AC
3. DI



Risk Adjusted Expected Return vs. Average Correlation

Average Correlation vs. Diversification Ratio

Do they tell the same story?



Average Correlation vs. Diversification Ratio

1. DR
2. AC
3. DI

3. Portfolio Diversification Index

The Volatility of a Portfolio in matrix form:

$$\sigma_{pf} = \sqrt{w'\boxed{\Sigma}w}$$

Decompose into independent factors

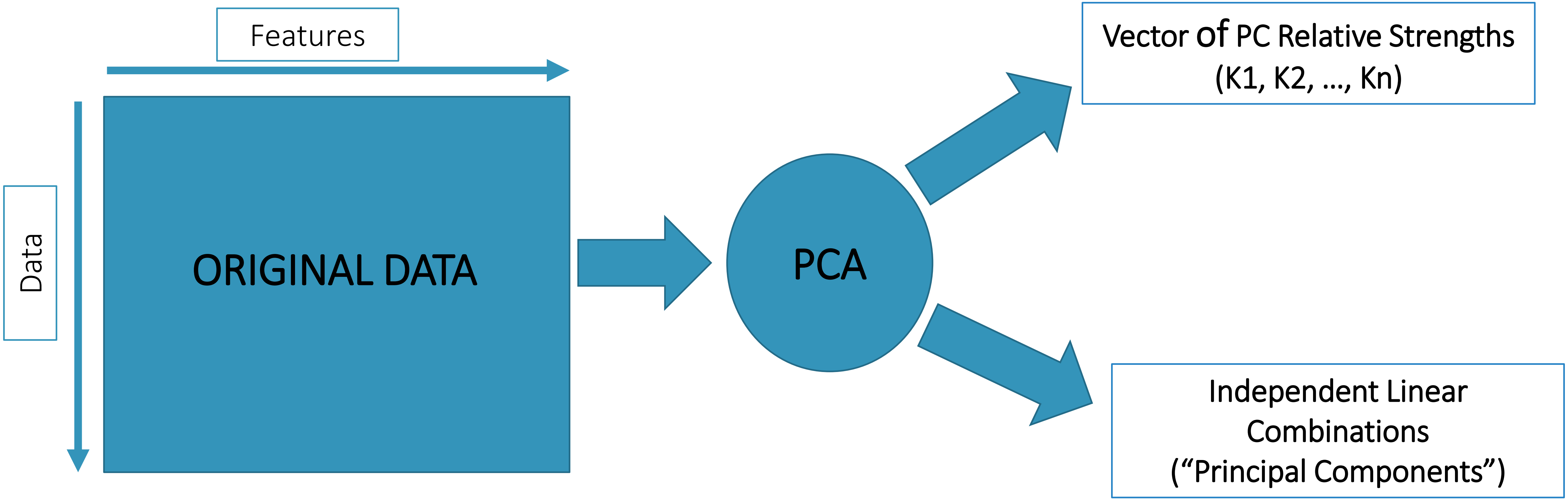
Where:

σ^2_{pf} – represents the variance of the portfolio,

w – represents the portfolio weight vector, and

Σ – represents the covariance matrix of the portfolio's investments

Principal Component Analysis Primer



```python
def get_pf_covar_mtxs(self):
    pf_covar_mtxs = dict()

    for j in self.weights.columns.levels[0]:
        for i in self.weights.columns.levels[1]:
            pf_vector = pd.DataFrame(self.portfolio_generation_mtx[j,i])
            pf_covar_mtxs[j,i] = pf_vector.dot(pf_vector.T) * self.investment_universe.covar_mtx

    return pf_covar_mtxs

self.covar_mtxs = get_pf_covar_mtxs(self)

def portfolio_eigenvalue_explained_variance(pf_covar_mtx):
    pca = PCA()
    return pca.fit(pf_covar_mtx).explained_variance_ratio_

def get_pf_eigenvalue_explvars(self):
    pf_eigenvalue_explvars = pd.DataFrame()

    for j in self.weights.columns.levels[0]:
        for i in self.weights.columns.levels[1]:
            pf_vector = pd.DataFrame(data=portfolio_eigenvalue_explained_variance(self.covar_mtxs[j,i]),
                                     index=pd.RangeIndex(1,len(self.covar_mtxs[j,i].index)+1),
                                     columns=pd.MultiIndex.from_tuples([(j,i)],names=('NumOfPosn','SimID')))

            pf_eigenvalue_explvars = pd.concat([pf_vector, pf_eigenvalue_explvars], axis=1)

    return pf_eigenvalue_explvars
```

SKLearn PCA object

PCA Explained Variance Ratio

3. Portfolio Diversification Index

$$\boldsymbol{PF\ Diversification\ Index = 2\sum_i i\,K_i - 1}$$

Where:

$\boldsymbol{K_i}$– represents weighted contribution of each eigenvalue on the system's variance (i.e. PC's relative strength)

```
def portfolio_diversification_index(pf_eigval_explained_var):
    return (2*sum(pf_eigval_explained_var*pf_eigval_explained_var.index))-1
```

3. Portfolio Diversification Index

$$\boldsymbol{PF\ Diversification\ Index\ \cong 1}$$

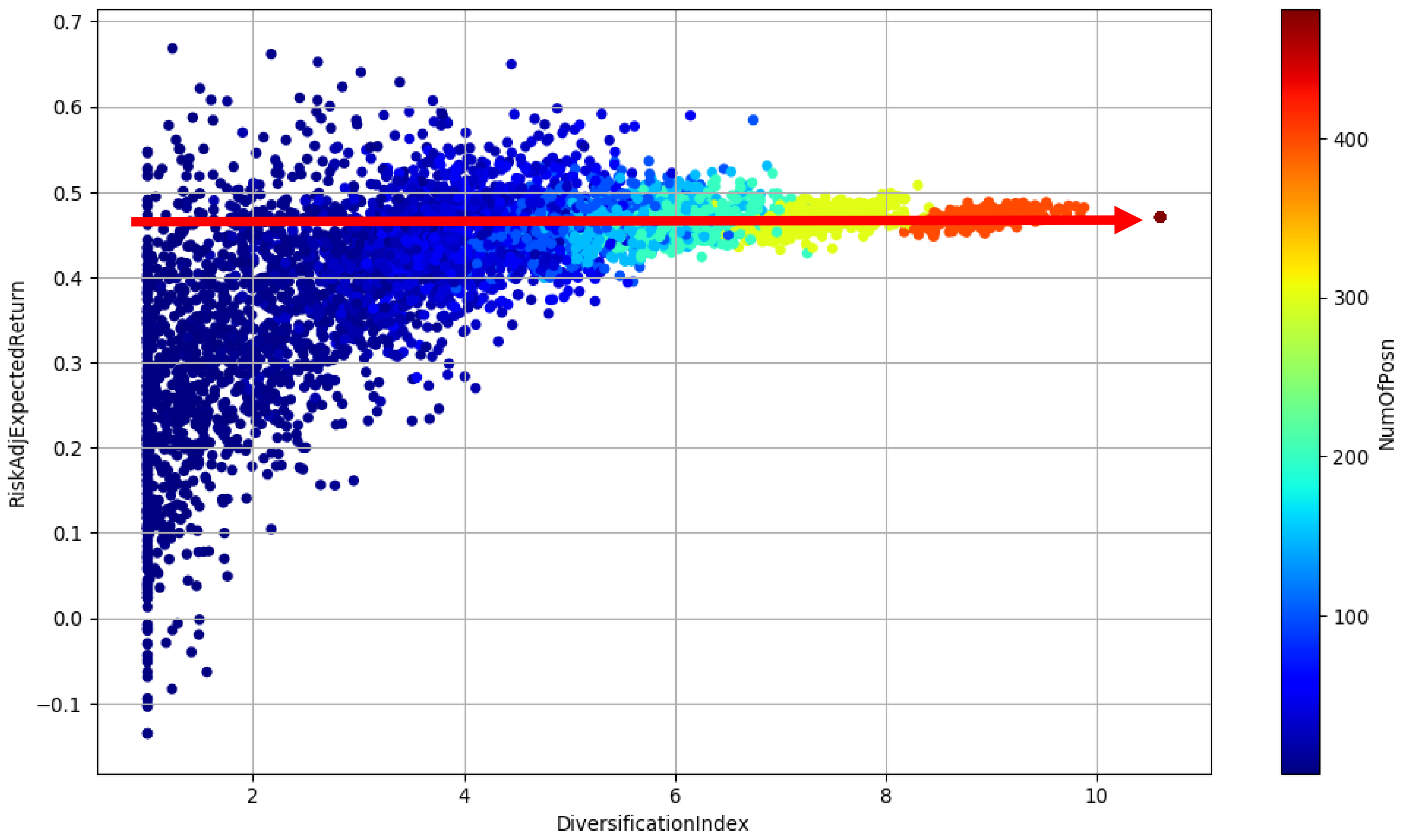
Portfolio has no diversification benefits

3. Portfolio Diversification Index

$$\boldsymbol{PF\ Diversification\ Index\ \cong N}$$

Portfolio is ideally diversified

1. DR
2. AC
3. DI



Risk Adjusted Expected Return vs. Diversification Index

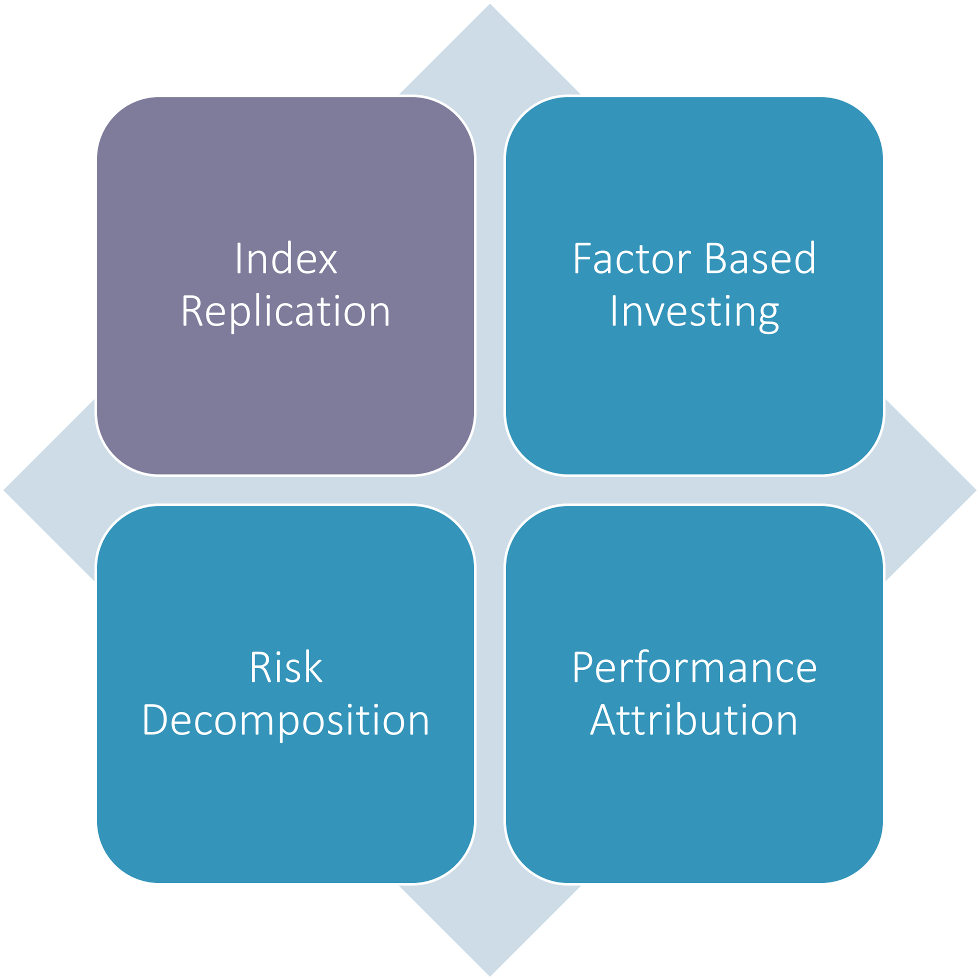
Recap: Diversification Metrics

1. Diversification Ratio
2. Average Correlation
3. Diversification Index

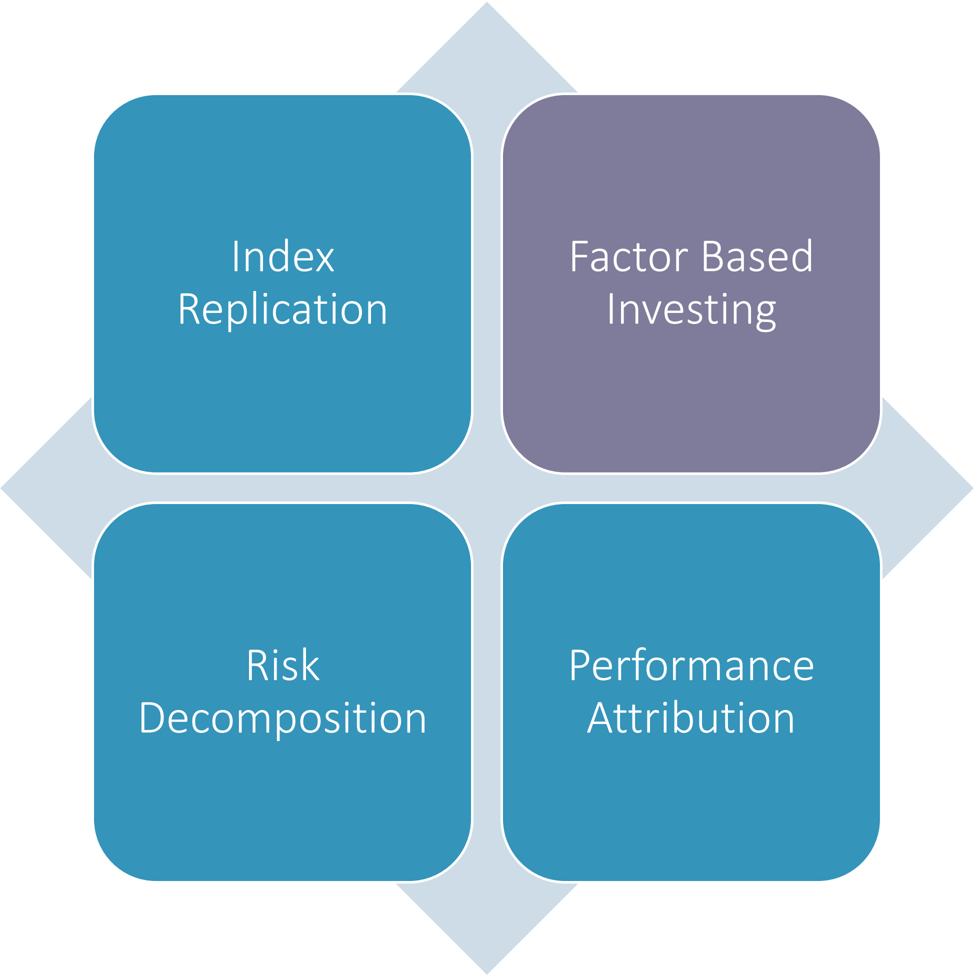
IV. Applications & Resources

Applications

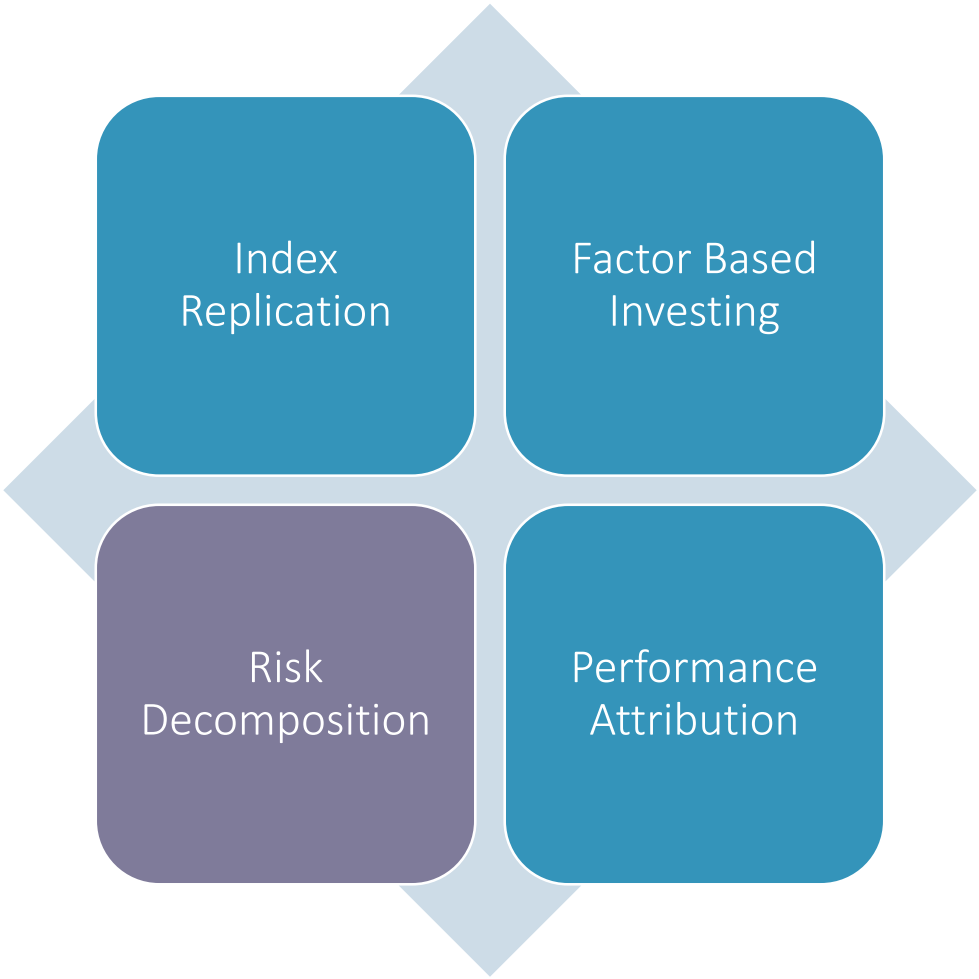
Applications



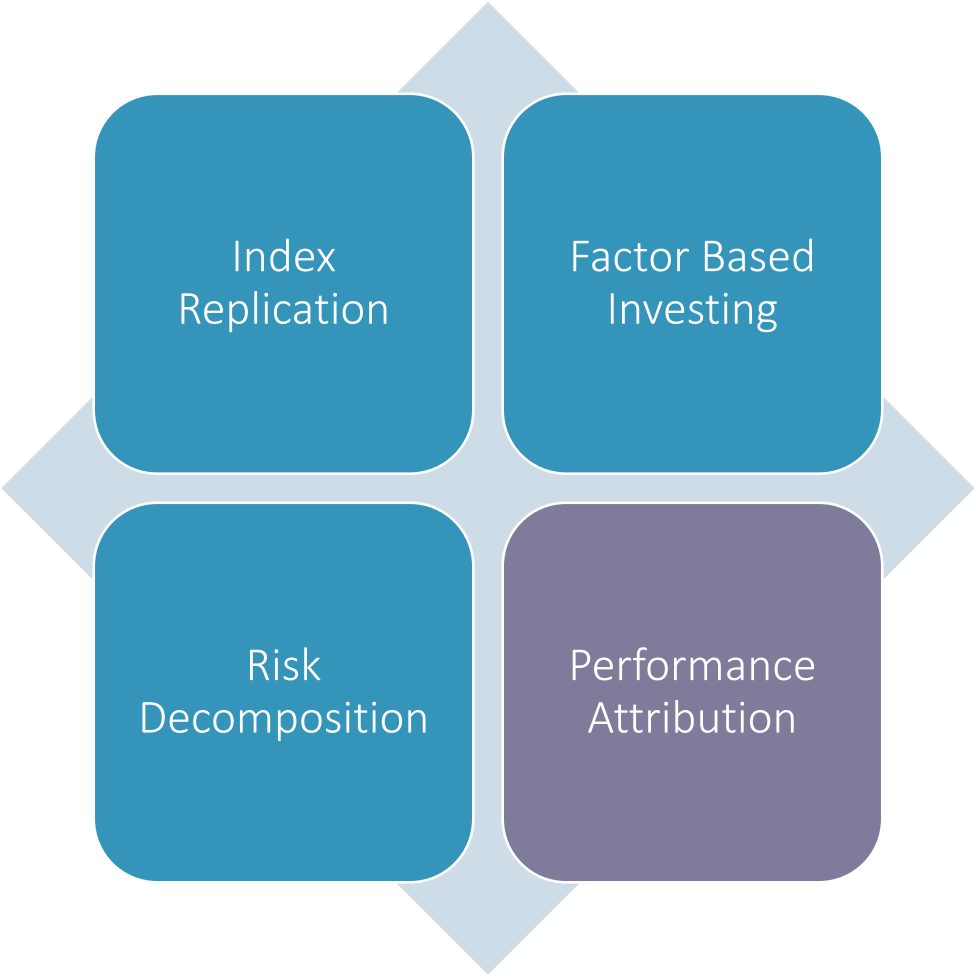
Applications



Applications



Applications



Resources

PYTHON LIBRARIES

- Pandas
- scikit-learn
- matplotlib

FINANCIAL LITERATURE

A Portfolio Diversification Index
- Rudin & Morgan
- Journal of Portfolio Management, Winter 2016

Towards Maximum Diversification
- Choueifaty & Coignard
- Journal of Portfolio Management, Fall 2008

Effective Number of Bets
- Attilio Meucci
- 2009a

THANK YOU!

Email: robin.warner@utam.utoronto.ca