AMS-511 Foundations of Quantitative Finance

Fall 2020 — Lecture 03 — 2020-09-16 Wednesday

Portfolio optimization
FinancialData[ ] function

Robert J. Frey, Research Professor
Stony Brook University, Applied Mathematics and Statistics

Robert.Frey@StonyBrook.edu
http://www.ams.sunysb.edu/~frey

Asset Return

Asset return, the relative increase in wealth over a specified time interval from owning an asset, is a fundamental measure of investment performance (See http://en.wikipedia.org/wiki/Rate_of_return).

In addition to owning an asset, holding a long position, it is also possible to create what is known as a short position, where the change in wealth from holding a short is the negative of the corresponding long position.

Rate of Return

A rate of return is much like an interest rate. Assuming that there are no other cash flows involved, the rate of return $r_{t-\Delta,t}$ over a period from $t-\Delta$ to t given starting and ending prices $S_{t-\Delta}$ and S_t is

$$S_t = S_{t-\Delta}(1 + r_{t-\Delta,t}) \;\Rightarrow\; r_{t-\Delta,t} = \frac{S_t - S_{t-\Delta}}{S_{t-\Delta}}$$

More generally, a rate of return from an activity can be thought of as a change in wealth w from one point in time $t-\Delta$ to t.

$$w_t = w_{t-\Delta}(1 + r_{t,t+\Delta}) \;\Rightarrow\; r_{t-\Delta,t} = \frac{w_t - w_{t-\Delta}}{w_{t-\Delta}}$$

Normally, when speaking of the single period of return we simplify the notation such that $r_t = r_{t-1,t}$.

$$r_t = \frac{w_t - w_{t-1}}{w_{t-1}}$$

Example — Dividend Payment

A stock with a price of S_t trades ex-dividend on t . The dividend payment is D_t. The rate of return on the stock is

$$r_t = \frac{w_t - w_{t-1}}{w_{t-1}} = \frac{(S_t + D_t) - S_{t-1}}{S_{t-1}}$$

Example — Stock Split

At time t a stock with a price of S_t experiences a s:1 stock split effective t. The rate of return on the stock is

$$r_t = \frac{w_t - w_{t-1}}{w_{t-1}} = \frac{s \times S_t - S_{t-1}}{S_{t-1}}$$

Log Returns

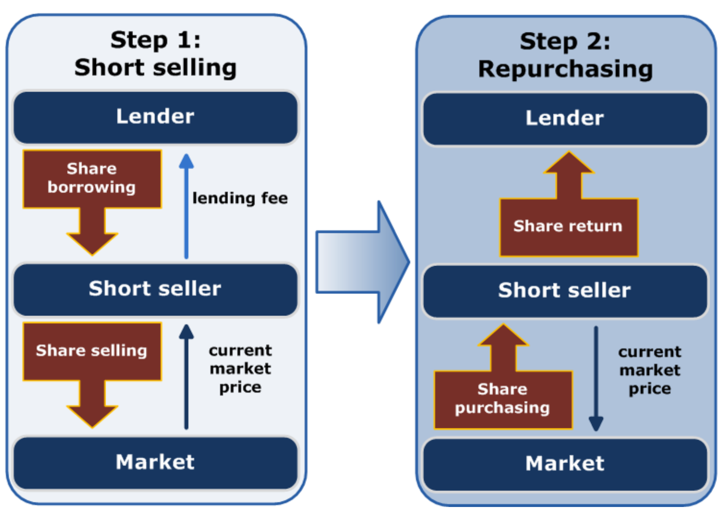
Sometimes we speak of log return, R_t, which is not the logarithm of return but the logarithm of the wealth ratio.

$$R_t = \text{Log}\left[\frac{w_t}{w_{t-1}}\right] = \text{Log}[1 + r_t]$$

$$R_1 + R_2 + \ldots + R_n = \text{Log}[(1 + r_1)(1 + r_2)\ldots(1 + r_n)]$$

Short Sales

Short selling (http://en.wikipedia.org/wiki/Short_sales) is the sale of an asset that one does not own. This is done by borrowing the asset from someone who does own it and then selling it in the market. At some future point the asset must be bought back in the open market to repay the loan by replacing the asset. Any cash flows experienced by the asset, e.g., dividend payments, that occur during the period of the loan must be replaced by the short seller. If the price of the asset has dropped then buying it back costs less than the original proceeds from selling it. Thus, the short seller makes a profit if the price has declined and experiences a loss if the price has risen.

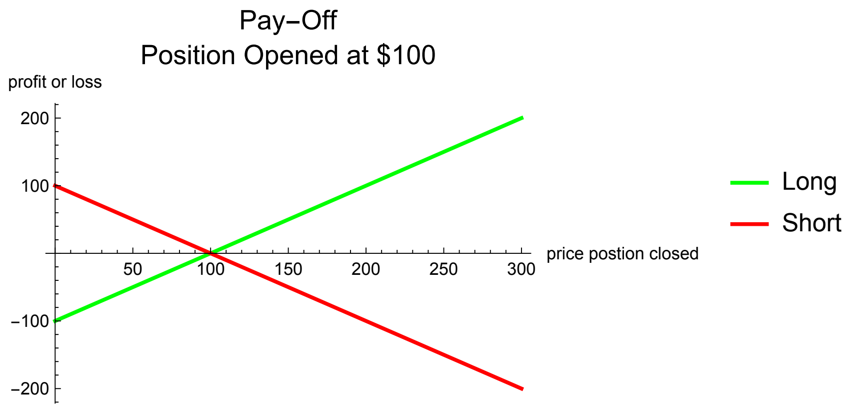


Wikipedia, "Short (finance)", 2014,

Shorting can be dangerous. If you buy an asset, then typically the most one can lose is the value of the asset. If one shorts an asset, there is no upper bound to the amount that can be lost. Here is an example of the profit and loss function for a non-dividend paying stock purchased or shorted at $100:

In[296]:=
```
Plot[{x - 100, 100 - x}, {x, 0, 300},
 PlotLabel →
  Style[Column[{"Pay-Off", "Position Opened at $100"}, Center], FontSize → 14],
 PlotStyle → {{Thick, Green}, {Thick, Red}},
 AxesLabel → {"price postion closed", "profit or loss"},
 PlotLegends → {"Long", "Short"}
]
```

Out[296]=



⚠ Note that the maximum loss of a long position is limited to the original value of the asset, but the maximum gain is unbounded. The maximum loss for a short position is unbounded, but the maximum gain is limited to the value of the original position.

Also, closing out a short position requires that shares be purchased in the market in order to return them to the original holder from whom the stock was borrowed. A short sequeeze results when there are insufficient shares available in the market to accomplish this. Thus, it is sometimes extremely difficult and costly, even impossible, to close out a short, and this tends to occur precisely when you are most desperate to do so!

Remember:

☺ LONG: Unbounded Gain, Bounded Loss — Subject to normal market liquidity.

☹ SHORT: Bounded Gain, Unbounded Loss — Subject to "short squeeze".

Distribution of Portfolio Returns

A common working assumption is that the distribution of asset returns follows a multivariate Normal distribution. This provides a framework in which the reward and risk of individual assets and of portfolios of those assets can be modeled.

http://en.wikipedia.org/wiki/Multivariate_normal_distribution

Multivariate Normal Distribution

Let the n-dimensional random variable $\mathbf{X} = (X_1, X_2, \ldots, X_n)^T$ follows a *multivariate Normal distribution*. This distribution is determined by two parameters:

- The *mean vector* $\boldsymbol{\mu}$ is an n vector

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{pmatrix}$$

$$\mu_i = \mathrm{E}[X_i]$$

- The *covariance matrix* $\boldsymbol{\Sigma}$ is an $n \times n$ matrix

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_{1,1} & \cdots & \sigma_{1,n} \\ \vdots & \ddots & \vdots \\ \sigma_{n,1} & \cdots & \sigma_{n,n} \end{pmatrix}$$

$$\sigma_{i,j} = \mathrm{Cov}[X_i, X_j] = E[(\mathrm{X}_i - E[\mathrm{X}_i])\,(\mathrm{X}_j - E[\mathrm{X}_j])]$$

Note from the definition of covariance that the covariance matrix Σ is symmetric, i.e., $\sigma_{i,j} = \sigma_{j,i}$ and that $\sigma_{i,i} = \sigma_i^{\,2} = \mathrm{Var}[X_i]$. Valid covariance matrices are also positive semi-definite, i.e.,

$$\mathbf{x}^T\,\boldsymbol{\Sigma}\,\mathbf{x} \geq \mathbf{0}, \quad \forall\, \boldsymbol{x} \neq \mathbf{0} \in \mathbb{R}^n$$

If the covariance matrix is of full rank, then the above inequality is strict and the matrix is said to be *positive definite*.

- The *correlation matrix* is formed by dividing each covariance term by the product of the associated standard deviations and is a measure of the relative strength of the relationship between the components of $\mathbf{X}$

$$\mathbf{C} = \begin{pmatrix} \rho_{1,1} & \cdots & \rho_{1,n} \\ \vdots & \ddots & \vdots \\ \rho_{n,1} & \cdots & \rho_{n,n} \end{pmatrix}$$

$$\rho_{i,j} = \frac{\sigma_{i,j}}{\sigma_i\,\sigma_j} \quad \Longrightarrow \quad \sigma_{i,j} = \rho_{i,j}\,\sigma_i\,\sigma_j$$

$$\rho_{i,i} = \frac{\sigma_i^{\,2}}{\sigma_i\,\sigma_i} \quad \Longrightarrow \quad \rho_{i,i} = 1$$

In our work here we will assume that the covariance matrix is of full rank and, thus, the PDF of a multivariate Normal distribution can be expressed as

$$\phi(\mathbf{x}) = \frac{1}{\sqrt{2\,\pi\,|\,\boldsymbol{\Sigma}\,|}}\,e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\,\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}$$

Example - Bivariate Normal Distribution

Consider the following mean and covariance parameters for a bivariate Normal distribution, i.e., a multivariate Normal of dimension 2.

$$\mu = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} 1 & -\frac{1}{3} \\ -\frac{1}{3} & 1 \end{pmatrix}$$

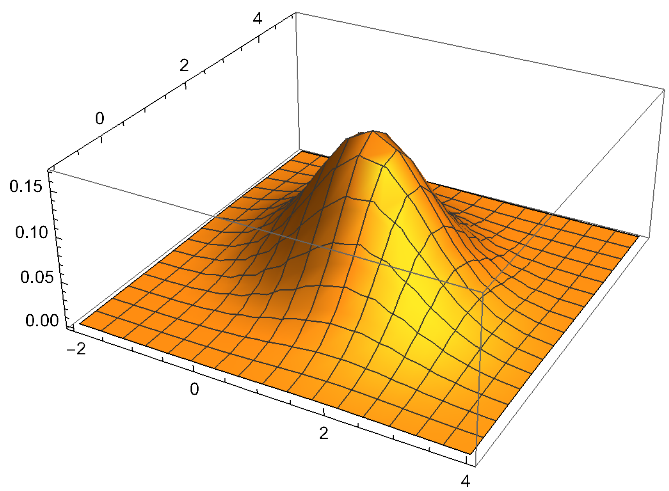
In[297]:=
```
PDF[NormalDistribution[μ, σ], x]
```

Out[297]=
$$\frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi}\,\sigma}$$

In[298]:=
```
PDF[GammaDistribution[α, β], τ]
```

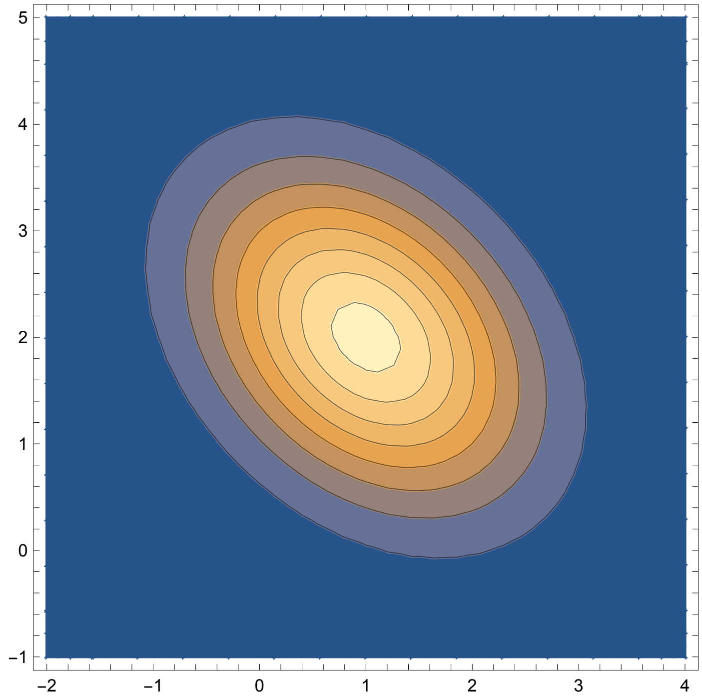
Out[298]=
$$\begin{cases} \frac{e^{-\frac{\tau}{\beta}}\beta^{-\alpha}\tau^{-1+\alpha}}{\text{Gamma}[\alpha]} & \tau > 0 \\ 0 & \text{True} \end{cases}$$

In[299]:=
```
vnM = {1, 2};
mnQ = {{1, -1/3}, {-1/3, 1}};
Plot3D[
 PDF[MultinormalDistribution[vnM, mnQ], {x, y}],
 {x, -2, 4}, {y, -1, 5},
 PlotRange → All
]
```

Out[301]=


In[302]:=
```
ContourPlot[
 PDF[MultinormalDistribution[vnM, mnQ], {x, y}],
 {x, -2, 4}, {y, -1, 5}
]
```

Out[302]=



Portfolio Mean and Variance

Let the vector $\mathbf{x}_{\mathcal{P}}$ represent the allocation of capital to assets of a portfolio $\mathcal{P}$, i.e., x_i = postion of asset i.

- Portfolio Mean

$$\mu_{\mathcal{P}} = \boldsymbol{\mu}^T \mathbf{x}_{\mathcal{P}}$$

- Portfolio Variance

$$\sigma_{\mathcal{P}}^2 = \mathbf{x}_{\mathcal{P}}^T \, \boldsymbol{\Sigma} \, \mathbf{x}_{\mathcal{P}}$$

Unless it is necessary to distinguish between different portfolios, we will normally drop the portfolio subscript $\mathcal{P}$.

Markowitz's Modern Portfolio Theory: Mean-Variance Portfolios

Modern Portfolio Theory or MPT is widely used as the basis for constructing portfolios. It assumes that asset returns can be (approximately) modeled using a multivariate Normal distribution and further assumes that risk is measured by a portfolio's return variance (or standard deviation) and reward is measured by a portfolio's expected return.

Despite concerns about the reasonableness of these assumptions, MPT provides a framework for selecting portfolios based on an optimal trade-off of risk and reward. MPT leads an investor to allocate capital across based on the overall characteristics of the portfolio rather than allocating capital to individual assets on a case-by-case basis without considering how the performance of individual assets may interact.

See http://en.wikipedia.org/wiki/Modern_portfolio_theory.

Monte Carlo Simulation

We'll simulate a large number of feasible portfolios and examine their behavior in $\{\sigma, \mu\}$-space.

In[303]:=
```
MatrixForm[KroneckerProduct[{a, b, c}, {a, b, c}]]
```

Out[303]//MatrixForm=

$$\begin{pmatrix} 36 & 6\,b & 6\,c \\ 6\,b & b^2 & b\,c \\ 6\,c & b\,c & c^2 \end{pmatrix}$$

In[304]:=
```
vnMean = {0.05, 0.08, 0.10};
vnSigma = {0.07, 0.09, 0.10};
mnCor = {{1, 0.4, 0.6}, {0.4, 1, 0.4}, {0.6, 0.4, 1}};
mnCov = KroneckerProduct[vnSigma, vnSigma] mnCor;
Print["μ = ", MatrixForm[vnMean]]
Print["Σ = ", MatrixForm[mnCov]]
```

$$\mu = \begin{pmatrix} 0.05 \\ 0.08 \\ 0.1 \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} 0.0049 & 0.00252 & 0.0042 \\ 0.00252 & 0.0081 & 0.0036 \\ 0.0042 & 0.0036 & 0.01 \end{pmatrix}$$

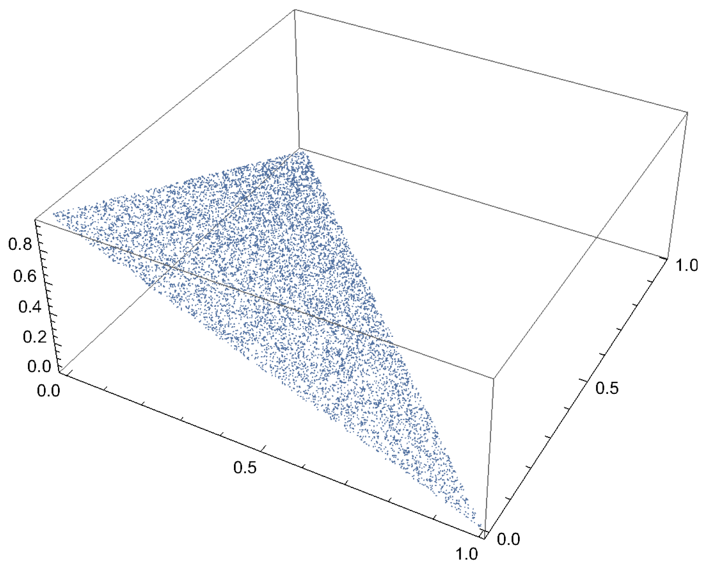
In generating random feasible portfolios we will assume that short positions are not permitted, i.e., $x_i \geq 0$. Our simulation will produce 10,000 cases for analysis.

In[310]:=
```
xRandomSimplexDirichlet[d_] :=
 Append[#, 1 - Total[#]] &[RandomVariate[DirichletDistribution[Array[1 &, d]]]]
```

In[311]:=
```
xRandomSimplexDirichlet[d_, n_] := Block[
   {α},
   α = Array[1 &, d];
   Append[#, 1 - Total[#]] & /@ RandomVariate[DirichletDistribution[α], n]
  ];
```

In[312]:=
```
mnSimplexExample = xRandomSimplexDirichlet[3, 10 000];
ListPointPlot3D[mnSimplexExample, PlotStyle → {PointSize → Tiny}]
```

Out[313]=



In[314]:=
```
iD = Length[vnMean];
iSimLength = 10 000;
xRandomPortfolio[n_] := #/Total[#] &[RandomVariate[UniformDistribution[], n]];
xPortSdevMean[x_, {μ_, Σ_}] := {√x.Σ.x , μ.x};
```

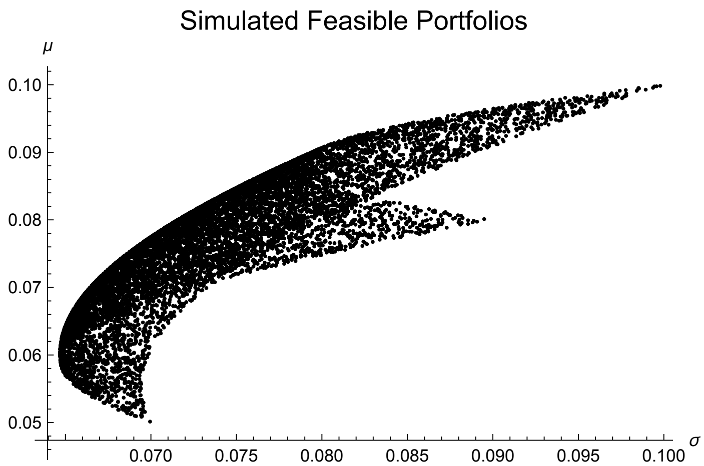
In[318]:=
```
vvnSdevMeanPts =
  Table[xPortSdevMean[xRandomSimplexDirichlet[iD], {vnMean, mnCov}], {iSimLength}];
```

In[319]:=
```
vvnSdevMeanPts⟦1 ;; 5⟧
```

Out[319]=
```
{{0.0649318, 0.0580984}, {0.0705546, 0.0725835},
 {0.0772442, 0.0743367}, {0.0780565, 0.0883256}, {0.0774673, 0.0830697}}
```

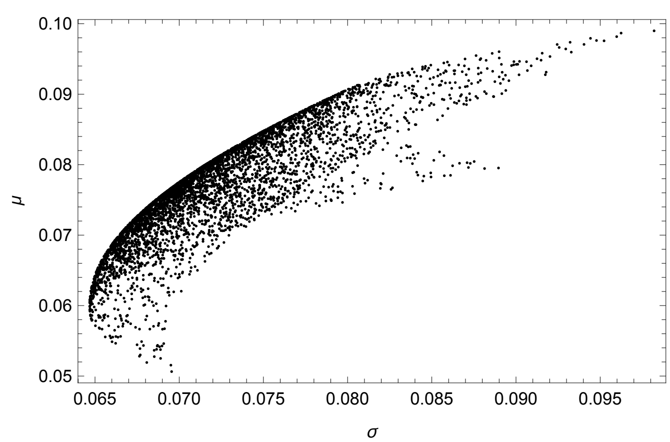
In[320]:=
```
ListPlot[
 vvnSdevMeanPts,
 PlotStyle → {PointSize → Small, Black},
 PlotLabel → Style["Simulated Feasible Portfolios", FontSize → 14],
 AxesLabel → {"σ", "μ"}
]
```

Out[320]=



Selecting a Set of Optimal Porfolios

Accepting for the moment that our definition of reward is the expected value of return and of risk the standard deviation of return, then clearly we wish to maximize a portfolio's expected return and minimize its standard deviation. While these conditions do not allow us to select a single optimal point from our feasible set they do allow us to restrict our solution to a set of values which are *Pareto optimal*. A solution that is subject to multiple objectives is Pareto optimal when increasing the value of one objective can only be accomplished by decreasing the value of another objective.



Click on image above to flip through different views.

The set of Pareto optimal mean-standard deviation solutions are called *efficient*. The line of efficient mean-standard deviation solutions, shown in red above, is called the *efficient frontier*, and the portfolios associated with that set are called *efficient portfolios*.

Markowitz Portfolio Analysis - Analytical Solution for a Simple Case

A portfolio optimization (See http://en.wikipedia.org/wiki/Portfolio_optimization) involves the solution of a mathematical program that seeks to create a portfolio that balances minimizing risk and maximizing reward subject to constraints reflecting the financial condition and policies of the portfolio holder.

Although the solution of a Markowitz Mean-Variance Portfolio Optimization must in general be solved numerically (See http://en.wikipedia.org/wiki/Quadratic_program), there are special cases which do admit an analytic solution. Such cases provide useful insights into the geometry of more general problems.

$$\mathcal{M} = \min\{\text{RISK} - \lambda \times \text{REWARD} \mid \text{PORFOLIO} \in \text{FEASIBLE SET}\}$$

$$\mathcal{M} = \min_{x}\left\{\frac{1}{2}\mathbf{x}^T\,\Sigma\,\mathbf{x} - \lambda\,\boldsymbol{\mu}^T\,\mathbf{x} \,\middle|\, \mathbf{x} \in \mathcal{S}\right\}$$

Mathematical Development

One form of the mean-variance (Markowitz) model is

$$\mathcal{M} = \min_{x}\left\{\frac{1}{2}\mathbf{x}^T\,\Sigma\,\mathbf{x} - \lambda\,\boldsymbol{\mu}^T\,\mathbf{x} \,\middle|\, \mathbf{1}^T\,\mathbf{x} = 1\right\}$$

As $\lambda \to 0$, $\mathcal{M}$ approaches a simple QP representing a minimum variance portfolio:

$$\min_{x}\left\{\frac{1}{2}\mathbf{x}^T\,\Sigma\,\mathbf{x} - \lambda\,\boldsymbol{\mu}^T\,\mathbf{x} \,\middle|\, \mathbf{1}^T\,\mathbf{x} = 1\right\} \xrightarrow{\lambda\to 0} \min_{x}\left\{\frac{1}{2}\mathbf{x}^T\,\Sigma\,\mathbf{x} \,\middle|\, \mathbf{1}^T\,\mathbf{x} = 1\right\}$$

As $\lambda\to\infty$ $\mathcal{M}$ approaches a LP representing a maximum return portfolio:

$$\min_{x}\left\{\frac{1}{2}\mathbf{x}^T\,\Sigma\,\mathbf{x} - \lambda\,\boldsymbol{\mu}^T\,\mathbf{x} \,\middle|\, \mathbf{1}^T\,\mathbf{x} = 1\right\} \xrightarrow{\lambda\to\infty} \max_{x}\left\{\boldsymbol{\mu}^T\,\mathbf{x} \,\middle|\, \mathbf{1}^T\,\mathbf{x} = 1\right\}$$

Here the vector $\mathbf{x}$ represents the proportional allocation of our capital; thus, the $\mathbb{1}^T\,\mathbf{x} = 1$ constraint. There is no restriction on the sign of $\mathbf{x}$. This means we are able to short positions, if necessary.

The Lagrangian (See http://en.wikipedia.org/wiki/Lagrange_multiplier) with multiplier ζ to price-out the capital constraint is

$$\mathcal{L}(\mathcal{M}) = \frac{1}{2}\mathbf{x}^T\,\Sigma\,\mathbf{x} - \lambda\,\boldsymbol{\mu}^T\,\mathbf{x} - \zeta\left(\mathbf{1}^T\,\mathbf{x} - 1\right)$$

Necessary conditions for optimality is that the gradient is equal to zero

$$\nabla\mathcal{L}(\mathcal{M}) = \Sigma\ \mathbf{x} - \lambda\,\boldsymbol{\mu} - \zeta\,\mathbf{1} = \mathbf{0}$$

and the constraints—here we have just the capital constraint—are satisfied.

We assume the covariance matrix is of full rank and, hence, positive definite, and this means that the objective function is convex. Thus, the solution of the above is also sufficient to ensure global optimality.

Solving for $\mathbf{x}$ the solution is

$$\mathbf{x} = \lambda\,\Sigma^{-1}\,\mu + \zeta\,\Sigma^{-1}\,\mathbf{1}$$

Assume that λ is held fixed, what is the value of ζ? In order for $\mathbf{x}$ to be an efficient portfolio, we know that ζ must be chosen so that the elements of $\mathbf{x}$ sum to 1.

$$1 = \mathbf{1}^T\,\mathbf{x} = \mathbf{1}^T\left(\lambda\,\Sigma^{-1}\,\mu + \zeta\,\Sigma^{-1}\,\mathbf{1}\right) \Rightarrow \zeta = \frac{1 - \lambda\,\mathbf{1}^T\,\Sigma^{-1}\,\mu}{\mathbf{1}^T\,\Sigma^{-1}\,\mathbf{1}}$$

Substituting ζ back into the solution for $\mathbf{x}$ we get

$$\mathbf{x} = \lambda\,\Sigma^{-1}\,\mu + \zeta\,\Sigma^{-1}\,\mathbf{1} = \lambda\,\Sigma^{-1}\,\mu + \left(\frac{1 - \lambda\,\mathbf{1}^T\,\Sigma^{-1}\,\mu}{\mathbf{1}^T\,\Sigma^{-1}\,\mathbf{1}}\right)\Sigma^{-1}\,\mathbf{1}$$

This result gives the efficient frontier parameterized by λ. For $\lambda = 0$, we ignore the issue of return entirely and the solution realized is the minimum variance portfolio. (Rewrite the above solution with $\lambda = 0$ to see what this looks like.).

As $\lambda \to \infty$, we place less and less emphasis on risk and more on just maximizing return. If shorts are permitted, as is the case in the above program, then additional capital will be raised by shorting lower return assets to invest in higher return ones.

Example

We'll reuse the same problem that we simulated above.

In[321]:=
```
? LinearSolve
```

Out[321]=

Symbol

LinearSolve[m, b] finds an x that solves the matrix equation $m.x == b$.
LinearSolve[m] generates a LinearSolveFunction[…] that can be applied repeatedly to different b.

In[322]:=
```
? Inverse
```

Out[322]=

Symbol

Inverse[m] gives the inverse of a square matrix m.

In[323]:=
```
xOptimalPortfolio[mnCov_, vnMean_, λ_] := Module[
   {mi, v1, vμ},
   mi = Inverse[mnCov];
   v1 = Total /@ mi;
   vμ = mi.vnMean;
   λ vμ + ((1 - λ Total[vμ]) / Total[v1]) v1
  ];
```

In[324]:=
```
xOptimalPortfolio[mnCor, vnMean, 0.03]
```

Out[324]=
```
{0.29805, 0.40015, 0.3018}
```

In[325]:=
```
xOptimalPortfolio[mnCor, vnMean, 0.1]
```

Out[325]=
```
{0.2935, 0.4005, 0.306}
```

In[326]:=
```
xOptimalPortfolio[mnCor, vnMean, 0.]
```

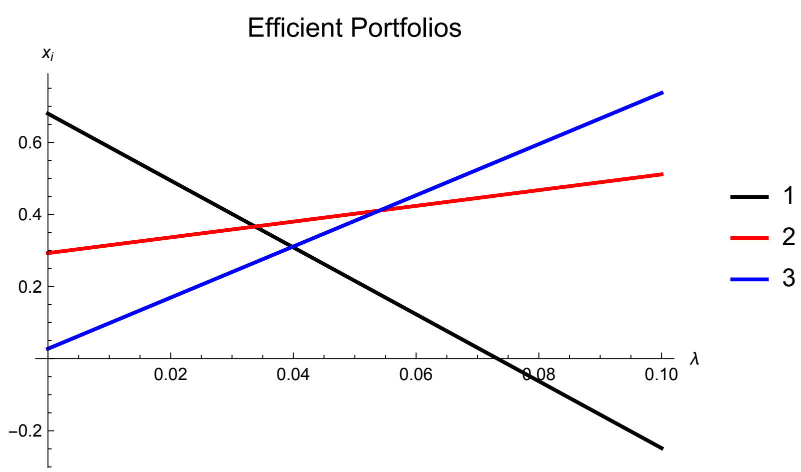
Out[326]=
```
{0.3, 0.4, 0.3}
```

In[327]:=
```
vuEfficientPortfolios = Table[
   {λ, xOptimalPortfolio[mnCov, vnMean, λ]},
   {λ, 0., 0.1, 0.01}
  ]
```

Out[327]=
```
{{0., {0.679656, 0.292846, 0.0274976}},
 {0.01, {0.586886, 0.314669, 0.0984444}}, {0.02, {0.494116, 0.336493, 0.169391}},
 {0.03, {0.401346, 0.358316, 0.240338}}, {0.04, {0.308575, 0.38014, 0.311285}},
 {0.05, {0.215805, 0.401963, 0.382232}}, {0.06, {0.123035, 0.423787, 0.453178}},
 {0.07, {0.0302646, 0.44561, 0.524125}}, {0.08, {-0.0625057, 0.467434, 0.595072}},
 {0.09, {-0.155276, 0.489257, 0.666019}}, {0.1, {-0.248046, 0.511081, 0.736966}}}
```

In[328]:=
```
ListLinePlot[
 Transpose[{First /@ vuEfficientPortfolios, #}] & /@
  Transpose[Last /@ vuEfficientPortfolios],
 PlotStyle → {{Black, Thick}, {Red, Thick}, {Blue, Thick}},
 PlotLabel → Style["Efficient Portfolios", FontSize → 14],
 AxesLabel → {"λ", "xᵢ"},
 PlotLegends → (ToString /@ Range[Length[vnMean]])
]
```

Out[328]=



In[329]:=
```
vuEfficientFrontier = Transpose[{First /@ vuEfficientPortfolios,
   xPortSdevMean[#, {vnMean, mnCov}] & /@ (Last /@ vuEfficientPortfolios)}]
```

Out[329]=
```
{{0., {0.0646821, 0.0601603}},
 {0.01, {0.0650061, 0.0643623}}, {0.02, {0.0659686, 0.0685643}},
 {0.03, {0.0675423, 0.0727664}}, {0.04, {0.0696858, 0.0769684}},
 {0.05, {0.0723484, 0.0811705}}, {0.06, {0.0754753, 0.0853725}},
 {0.07, {0.0790113, 0.0895746}}, {0.08, {0.0829041, 0.0937766}},
 {0.09, {0.0871059, 0.0979787}}, {0.1, {0.0915741, 0.102181}}}
```

In[330]:=
```
? Tooltip
```

Out[330]=

Symbol

Tooltip[*expr*, *label*] displays *label* as a tooltip while the mouse pointer is in the area where *expr* is displayed.

In[331]:=
```
Tooltip["This is a sentence.",
 Style["And a short one at that!", FontSize → 72, FontColor → Red]]
```

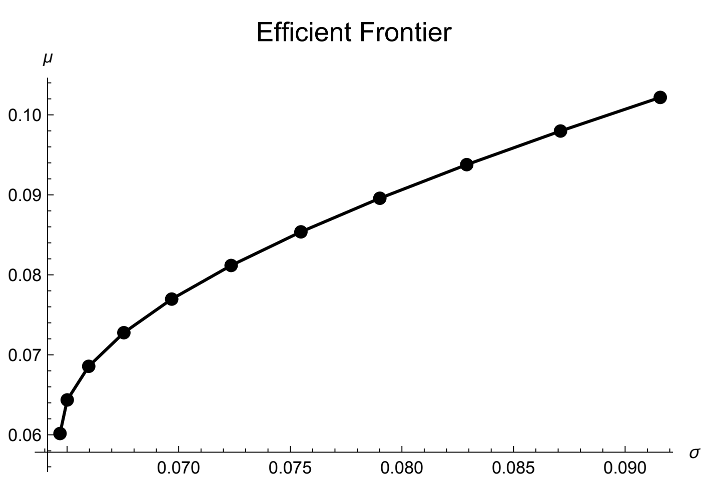
Out[331]=
```
This is a sentence.
```

In[332]:=
```
MapThread[f, {{a, b, c}, {x, y, z}}]
```

Out[332]=
```
{f[6, x], f[b, y], f[c, z]}
```

In[333]:=
```
ListPlot[
 MapThread[Tooltip, {Last /@ vuEfficientFrontier, vuEfficientPortfolios}],
 PlotStyle → {PointSize[Large], Black},
 PlotLabel → Style["Efficient Frontier", FontSize → 14],
 Joined → True, Mesh → All,
 AxesLabel → {"σ", "μ"}
]
```

Out[333]=



Markowitz Portfolio Analysis - An Extended Example with No Short Positions

The introduction of no-short constraints means we must rely on a non-linear program solver. An extended example using *Mathematica*'s NMinimize[ ] function is illustrated.

Mathematical Development

Consider the following problem. We modify the program above to replace the trade-off parameter λ with a target return constraint and to add no-short constraints:

$$\mathcal{M} = \min_{x} \left\{ \frac{1}{2} \mathbf{x}^T \Sigma \mathbf{x} \;\middle|\; \mathcal{S} = \left\{ \mu^T \mathbf{x} \geq \tau \;\&\&\; \mathbb{1}^T \mathbf{x} = 1 \;\&\&\; \mathbf{x} \geq \mathbf{0} \right\} \right\}$$

Example

Again, we'll reuse our simple 3-asset mean and covariance problem.

Mathematica has a number of built-in functions to support optimization. We'll use one of them here, NMinimize[ ], to solve for a the efficient portfolio corresponding to a particular target return.

NMinimize[ ] requires that we specify a vector containing the objective function and constraint set and a list of decision variables over which the optimization is run. The decision variables are the portfolio allocations $\mathbf{x} = \{x_{1,\ldots,} x_i, \ldots, x_n\}$.

In[334]:=
```
? NMinimize
```

Out[334]=

Symbol

NMinimize[f, x] minimizes f numerically with respect to x.
NMinimize[f, {x, y, …}] minimizes f numerically with respect to x, y, ….
NMinimize[{f, *cons*}, {x, y, …}] minimizes f numerically subject to the constraints *cons*.
NMinimize[…, $x \in$ *reg*] constrains x to be in the region *reg*.

In[335]:=
```
x = {x1, x2, x3};
```

The objective function is the portfolio variance divided by 2.

In[336]:=
$$\frac{\texttt{x.mnCov.x}}{2}$$

Out[336]=
$$\frac{1}{2} \left(x2 \left(0.00252\, x1 + 0.0081\, x2 + 0.0036\, x3\right) + x1 \left(0.0049\, x1 + 0.00252\, x2 + 0.0042\, x3\right) + \left(0.0042\, x1 + 0.0036\, x2 + 0.01\, x3\right) x3\right)$$

The constraints are expressed as a conjunction of the individual constraints. The total capital constraint is appended to the list of non-negativity constraints, producing a vector of logical conditions.

In[337]:=
```
Append[# ≥ 0 & /@ x, Total[x] == 1]
```
Out[337]=
```
{x1 ≥ 0, x2 ≥ 0, x3 ≥ 0, x1 + x2 + x3 == 1}
```

In[338]:=
```
FullForm[%]
```
Out[338]//FullForm=
```
List[GreaterEqual[x1, 0], GreaterEqual[x2, 0],
 GreaterEqual[x3, 0], Equal[Plus[x1, x2, x3], 1]]
```

This list is then converted into a conjunction by using Apply[ ] (in the short infix form @@) to replace the Head of the List with the logical operation And.

In[339]:=
```
And @@ Append[# ≥ 0 & /@ x, Total[x] == 1]
```
Out[339]=
```
x1 ≥ 0 && x2 ≥ 0 && x3 ≥ 0 && x1 + x2 + x3 == 1
```

In[340]:=
```
FullForm[%]
```
Out[340]//FullForm=
```
And[GreaterEqual[x1, 0], GreaterEqual[x2, 0],
 GreaterEqual[x3, 0], Equal[Plus[x1, x2, x3], 1]]
```

Next we put these elements together into a NMinimize[ ] call. The value returned is the value of the objective function and the optimal solution as a vector of rules. Note that we just have the portfolio variance in the objective function with no constraints associated with return. Thus, the value returned here is the minimum variance portfolio.

In[341]:=
```
NMinimize[{x.mnCov.x/2, And @@ Join[{Total[x] == 1}, # ≥ 0 & /@ x]}, x]
```
Out[341]=
```
{0.00209189, {x1 → 0.679656, x2 → 0.292846, x3 → 0.0274976}}
```

In[342]:=
```
x /. Last[%]
```
Out[342]=
```
{0.679656, 0.292846, 0.0274976}
```

We can extract the solution and ensure that we produce a numeric vector with the decision variables in the desired order. Here the order in the rule vector happens to be the same as we might desire, but you can't count on that. You need to use the approach below to ensure that the values are in the order you want.

In[343]:=
```
vnMinVar =
 x /. Last[NMinimize[{x.mnCov.x/2, And @@ Join[{Total[x] == 1}, # ≥ 0 & /@ x]}, x]]
```
Out[343]=
```
{0.679656, 0.292846, 0.0274976}
```

The return associated with the minimum variance porfolio is the minimum target return.

In[344]:=
```
nMuMinVar = vnMean.vnMinVar
```
Out[344]=
```
0.0601603
```

Given the no short condition, the maximum target return is simply the asset with the highest return. We can compute that without complication; however, if there were a more complex set of constraints, then we would need to construct an optimization, a linear program, that would maximize portfolio return subject to the portfolio constraints.

In[345]:=
```
nMuMax = Max[vnMean]
```
Out[345]=
```
0.1
```

We can adapt the above to return any point on the efficient frontier by adding the target return constraint to $\mathcal{S}$. We use the variable nTarget to represent the target return in a given optimization.

```
In[346]:= And @@ Join[{vnMean.x ≥ nTarget, Total[x] == 1}, # ≥ 0 & /@ x]

Out[346]= 0.05 x1 + 0.08 x2 + 0.1 x3 ≥ nTarget && x1 + x2 + x3 == 1 && x1 ≥ 0 && x2 ≥ 0 && x3 ≥ 0
```

Here is a function that will produce the efficient frontier given a target return and a vector containing the parameters of a multivariate Normal distribution.

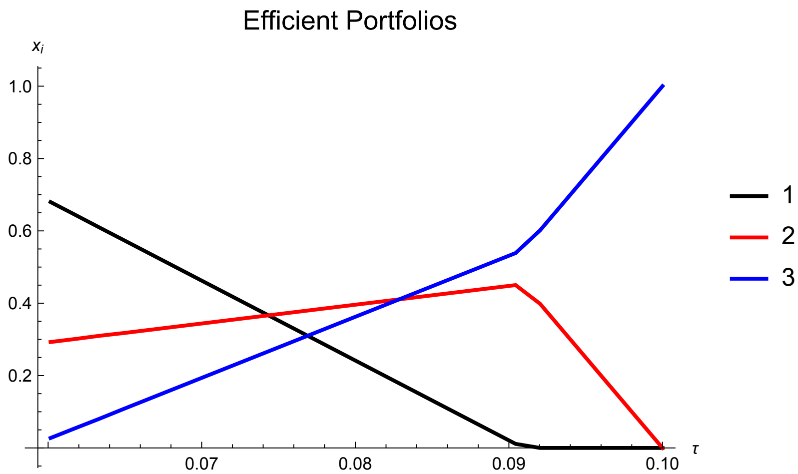
```
In[347]:= xEfficientPortfolioNoShorts[nTarget_, {vnMean_, mnCov_}] := x /. Last[NMinimize[
            {x.mnCov.x/2, And @@ Join[{vnMean.x ≥ nTarget, Total[x] == 1}, # ≥ 0 & /@ x]}, x]];

In[348]:= xEfficientPortfolioNoShorts[0.08, {vnMean, mnCov}]

Out[348]= {0.241638, 0.395904, 0.362458}
```

In[349]:=
```
Off[NMinimize::incst]
vuEfficientPortfoliosNoShort = Table[
  {nTarget, xEfficientPortfolioNoShorts[nTarget, {vnMean, mnCov}]},
  {nTarget, nMuMinVar, nMuMax, (nMuMax - nMuMinVar)/25}
 ]
On[NMinimize::incst]
```

Out[350]=
```
{{0.0601603, {0.679656, 0.292846, 0.0274976}},
 {0.0617538, {0.644541, 0.301134, 0.0543253}},
 {0.0633474, {0.609182, 0.310031, 0.080787}},
 {0.064941, {0.574104, 0.317688, 0.108208}},
 {0.0665346, {0.538921, 0.325967, 0.135112}},
 {0.0681282, {0.503739, 0.334242, 0.162019}},
 {0.0697218, {0.468557, 0.342519, 0.188925}},
 {0.0713154, {0.433374, 0.350796, 0.21583}},
 {0.072909, {0.398191, 0.359073, 0.242736}},
 {0.0745026, {0.363009, 0.36735, 0.269641}},
 {0.0760962, {0.327826, 0.375627, 0.296547}},
 {0.0776897, {0.292643, 0.383904, 0.323452}},
 {0.0792833, {0.257461, 0.392182, 0.350358}},
 {0.0808769, {0.222278, 0.400459, 0.377263}},
 {0.0824705, {0.187095, 0.408736, 0.404169}},
 {0.0840641, {0.151913, 0.417013, 0.431074}},
 {0.0856577, {0.11673, 0.42529, 0.45798}},
 {0.0872513, {0.0815474, 0.433567, 0.484885}},
 {0.0888449, {0.0463647, 0.441845, 0.511791}},
 {0.0904385, {0.011182, 0.450122, 0.538696}},
 {0.0920321, {6.93276 × 10^-9, 0.398399, 0.601601}},
 {0.0936256, {6.81709 × 10^-9, 0.31872, 0.68128}},
 {0.0952192, {6.41767 × 10^-9, 0.239041, 0.760959}},
 {0.0968128, {-2.13425 × 10^-9, 0.15937, 0.84063}},
 {0.0984064, {7.26689 × 10^-9, 0.0796803, 0.92032}},
 {0.1, {-1.27579 × 10^-9, 3.93052 × 10^-6, 0.999996}}}
```

In[352]:=
```
ListLinePlot[
  Transpose[{First /@ vuEfficientPortfoliosNoShort, #}] & /@
   Transpose[Last /@ vuEfficientPortfoliosNoShort],
  PlotStyle → {{Black, Thick}, {Red, Thick}, {Blue, Thick}},
  PlotLabel → Style["Efficient Portfolios", FontSize → 14],
  AxesLabel → {"τ", "xᵢ"},
  PlotLegends → (ToString /@ Range[Length[vnMean]])
 ]
```

Out[352]=

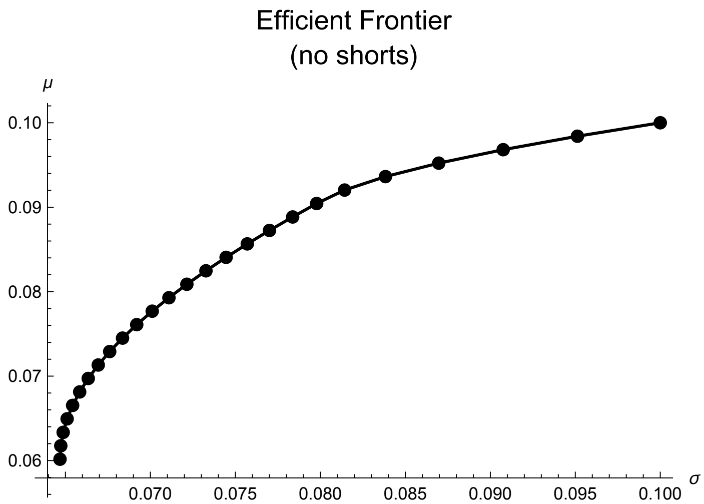


In[353]:=
```
vuEfficientFrontierNoShort = Transpose[{First /@ vuEfficientPortfoliosNoShort,
    xPortSdevMean[#, {vnMean, mnCov}] & /@ (Last /@ vuEfficientPortfoliosNoShort)}]
```

Out[353]=
```
{{0.0601603, {0.0646821, 0.0601603}}, {0.0617538, {0.0647286, 0.0617503}},
 {0.0633474, {0.0648679, 0.0633403}}, {0.064941, {0.0651012, 0.064941}},
 {0.0665346, {0.0654253, 0.0665346}}, {0.0681282, {0.0658397, 0.0681282}},
 {0.0697218, {0.0663426, 0.0697218}}, {0.0713154, {0.0669321, 0.0713154}},
 {0.072909, {0.067606, 0.072909}}, {0.0745026, {0.0683616, 0.0745026}},
 {0.0760962, {0.0691963, 0.0760962}}, {0.0776897, {0.0701074, 0.0776897}},
 {0.0792833, {0.0710918, 0.0792833}}, {0.0808769, {0.0721466, 0.0808769}},
 {0.0824705, {0.0732688, 0.0824705}}, {0.0840641, {0.0744552, 0.0840641}},
 {0.0856577, {0.0757029, 0.0856577}}, {0.0872513, {0.0770089, 0.0872513}},
 {0.0888449, {0.0783702, 0.0888449}}, {0.0904385, {0.0797841, 0.0904385}},
 {0.0920321, {0.0814283, 0.092032}}, {0.0936256, {0.083831, 0.0936256}},
 {0.0952192, {0.0869661, 0.0952192}}, {0.0968128, {0.0907574, 0.0968126}},
 {0.0984064, {0.0951278, 0.0984064}}, {0.1, {0.0999997, 0.0999999}}}
```

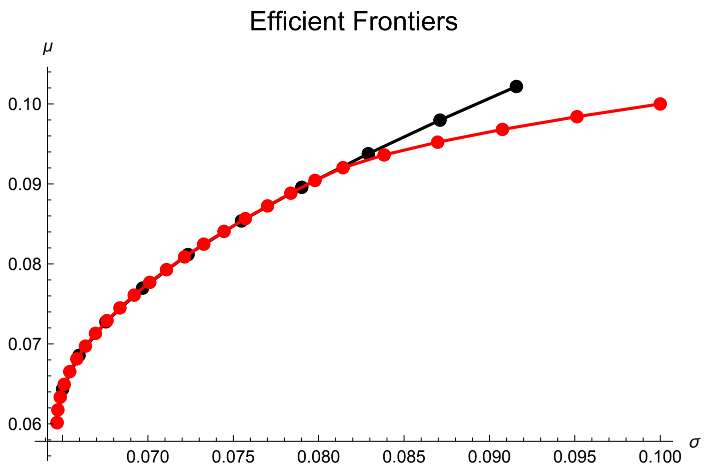
In[354]:=
```
ListPlot[
 MapThread[Tooltip,
  {Last /@ vuEfficientFrontierNoShort, vuEfficientPortfoliosNoShort}],
 PlotStyle → {PointSize[Large], Black},
 PlotLabel →
  Style[Column[{"Efficient Frontier", "(no shorts)"}, Center], FontSize → 14],
 Joined → True, Mesh → All,
 AxesLabel → {"σ", "μ"}
]
```

Out[354]=



In[355]:=
```
Show[
 ListPlot[
  MapThread[Tooltip, {Last /@ vuEfficientFrontier, vuEfficientPortfolios}],
  PlotStyle → {PointSize[Large], Black},
  PlotLabel → Style["Efficient Frontiers", FontSize → 14],
  Joined → True, Mesh → All,
  AxesLabel → {"σ", "μ"}
 ],
 ListPlot[
  MapThread[Tooltip,
   {Last /@ vuEfficientFrontierNoShort, vuEfficientPortfoliosNoShort}],
  PlotStyle → {PointSize[Large], Red},
  PlotLabel →
   Style[Column[{"Efficient Frontier", "(no shorts)"}, Center], FontSize → 14],
  Joined → True, Mesh → All
 ],
 PlotRange → All
]
```

Out[355]=



Tangent Portfolio

Within the framework of Markowitz's Mean-Variance Portfolio Optimization, a concept called the tangent portfolio provides us with a rationale for selecting a particular point on the efficient frontier of risky assets.

In general, determining the tangent portfolio must be done numerically; however, for certain special cases we can directly solve for the tangent portfolio by including a risk-free asset and then observing that the tangent portfolio is that portfolio which has a zero position in cash.

Leverage: Using Saving or Borrowing to Control Risk and Reward

To the set of risky assets consider adding the possibility of investing in a risk-free asset with rate of return r_f. We will call holdings in the risk-free asset the *cash position*. Assume that we can either lend (a positive cash position) or borrow (a negative cash position) at this rate. Let $\mathcal{P}$ be an efficient portfolio of risky assets whose proportional positions $\mathbf{x}_{\mathcal{P}}$ have a total capital of 1.

We then combine $\mathcal{P}$ with the risk-free asset by scaling $\mathbf{x}_{\mathcal{P}}$ by a factor $0 \leq \phi$ (i.e., $\mathbf{x}_{\mathcal{P}(\phi)} = \phi\, \mathbf{x}_{\mathcal{P}}$) and holding a cash position of $(1-\phi)$ forming a new portfolio $\mathcal{P}(\phi)$. Thus, the total capital of 1 is preserved. We will call ϕ the *leverage factor*. The relationship between the standard deviations and means of $\mathcal{P}$ and $\mathcal{P}(\phi)$ is

$$\begin{pmatrix} \sigma_{\mathcal{P}} \\ \mu_{\mathcal{P}} \end{pmatrix} = \begin{pmatrix} \sqrt{\mathbf{x}_{\mathcal{P}}{}^T\, \mathbf{\Sigma}\, \mathbf{x}_{\mathcal{P}}} \\ \boldsymbol{\mu}^T\, \mathbf{x}_{\mathcal{P}} \end{pmatrix}$$

with the mean and standard deviation of $\mathcal{P}(\phi)$ being computed most simply by observing that

$$\begin{pmatrix} \sigma_{\mathcal{P}(\phi)} \\ \mu_{\mathcal{P}(\phi)} \end{pmatrix} = (1-\phi) \begin{pmatrix} 0 \\ r_f \end{pmatrix} + \phi \begin{pmatrix} \sigma_{\mathcal{P}} \\ \mu_{\mathcal{P}} \end{pmatrix}$$

To summarize, we can find ourselves in the following possible states:

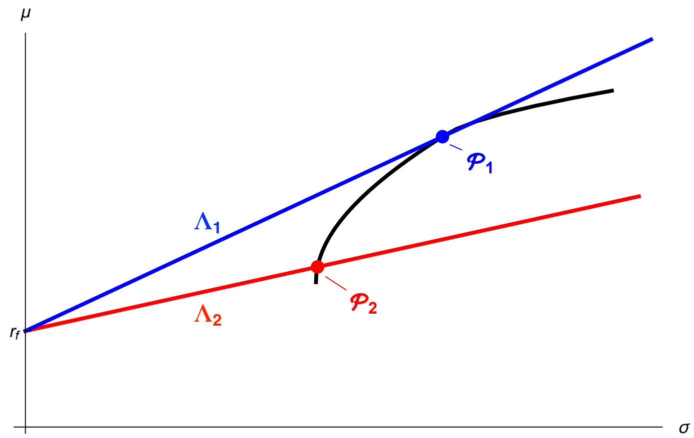
- $\phi = 0$ We have an all-cash position of 1 and invest nothing in $\mathcal{P}$.
- $0 < \phi < 1$ We keep $(1-\phi)$ in cash and receive a rate of return of r_f. The remaining ϕ of capital is invested proportionally in the risky portfolio.
- $\phi = 1$ We have no cash position; thus, $\mathcal{P}(1) = \mathcal{P}$.
- $1 < \phi$ We borrow $(\phi - 1)$ of cash paying an interest rate of r_f. We then invest ϕ proportionally into the risky portfolio.

Common nomenclature is to call situations where $0 \leq \phi \leq 1$ (where there is no borrowing) an *unleveraged* position and $1 < \phi$ (where borrowing is used to create a position greater than our capital) a *leveraged* position. See http://en.wikipedia.org/wiki/Leverage_(finance) for a broader description of financial leverage.

The Tangent Portfolio: Maximizing the Ratio of Reward to Risk

The equation above for $\mathcal{P}(\phi)$, representing it in standard deviation-mean space, is a parametric equation for a straight line $\Lambda(\phi \mid \mathbf{x}_{\mathcal{P}}, \boldsymbol{\mu}, \mathbf{\Sigma}, r_f)$ which we will call the *portfolio line*. It's intercept is r_f and its slope is $(\mu_{\mathcal{P}} - r_f)/\sigma_{\mathcal{P}}$. Thus, portfolio lines radiate out from r_f on the μ-axis and intersect the associated point(s) on the efficient frontier.

Consider two portfolios $\mathcal{P}_1$ and $\mathcal{P}_2$ and their associated lines Λ_1 and Λ_2 such that Λ_1 has the steeper slope, the standard deviation-mean combinations represented by Λ_1 dominate those of Λ_2; *i.e.*, for any portfolio in Λ_2 there exists portfolios in Λ_1 with higher reward at the same or lower risk (or lower risk at the same or higher reward).



Thus, we wish to select the portfolio line whose portfolios dominate all others. This is the one with the steepest slope. Given the concave shape of the efficient frontier, the portfolio line that is tangent to it produces the desired optimal Λ. The portfolio associated with the point of tangency on the efficient frontier is called the *tangent portfolio*.

Again, stressing that we are viewing risk and reward solely in terms of standard deviation and mean, ***we have no reason to hold any other portfolio of risky assets other than the tangent portfolio***.

If we desire less risk or more return than the tangent portfolio, we achieve it by lending or borrowing cash and investing less or more capital into the tangent portfolio. Any other choice involves an unnecessary sacrifice of risk or reward or both and is, therefore, sub-optimal.

The ratio $(\mu_{\mathcal{P}} - r_f)/\sigma_{\mathcal{P}}$, the slope of the portfolio line, is usually called the *Sharpe ratio* and is a common statistic used to evaluate both individual assets and portfolios (See http://en.wikipedia.org/wiki/Sharpe_ratio). ***It tells us the "price" of reward over the risk-free baseline in units of risk.*** The tangent portfolio is, of course, that efficient portfolio whose Sharpe ratio is maximal.

Originally developed by William Sharpe, the Sharpe ratio between two investment choices A and B was defined more generally as $\mu[r_A - r_B]/\sigma[r_A - r_B]$, the mean divided by the standard deviation of the difference of the two return streams. The form $(\mu - r_f)/\sigma$ has, however, become what people almost always mean when they talk about the Sharpe ratio.

Analytic Solution of the Tangent Portfolio

We will include the risk-free asset into the portfolio optimization problem where we seek to minimize variance subject to a target return. However, we replace the capital constraint with a computation that directly computes the amount of cash that is loaned or borrowed.

$$\mathcal{M} = \min_{x} \left\{ \frac{1}{2} \mathbf{x}^T \, \Sigma \, \mathbf{x} \;\middle|\; \left(\boldsymbol{\mu}^T \mathbf{x} + \left(1 - \mathbf{1}^T \mathbf{x}\right) r_f \right) = \tau \right\}$$

The portfolio of risky assets is $\mathbf{x}$ which represent a capital investment of $\mathbf{1}^T \mathbf{x}$. The amount loaned or borrowed is, therefore, $\left(1 - \mathbf{1}^T \mathbf{x}\right)$. Thus, total capital is constrained implicitly by the target return constraint to be 1.

This problem can be solved, as previously, by pricing out the constraint, taking the gradient, setting it to $\mathbf{0}$, and then solving for $\mathbf{x}$.

$$\mathcal{L}(\mathcal{M}) = \frac{1}{2} \mathbf{x}^T \, \Sigma \, \mathbf{x} - \lambda \left(\boldsymbol{\mu}^T \mathbf{x} + \left(1 - \mathbf{1}^T \mathbf{x}\right) r_f \right)$$

$$\nabla \mathcal{L}(\mathcal{M}) = \Sigma \, \mathbf{x} - \lambda \left(\boldsymbol{\mu} - \mathbf{1} \, r_f \right) = \mathbf{0}$$

$$\mathbf{x} = \lambda\, \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu} - \mathbf{1}\, r_f)$$

We now solve for the value of the Lagrange multiplier λ that results in no cash being invested or borrowed, i.e., a portfolio $\mathbf{x}$ of risky assets whose total capital is 1. Once the appropriate value of λ is determined, it is substituted back into the portfolio solution.

$$1 = \mathbf{1}^T \mathbf{x} = \lambda\, \mathbf{1}^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu} - \mathbf{1}\, r_f) \implies \lambda = \frac{1}{\mathbf{1}^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu} - \mathbf{1}\, r_f)}$$

Let $\mathcal{T}$ denote the tangent portfolio so the capital allocation of risky assets is

$$\mathbf{x}_{\mathcal{T}} = \frac{\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu} - \mathbf{1}\, r_f)}{\mathbf{1}^T \boldsymbol{\Sigma}^{-1}\left(\boldsymbol{\mu} - \mathbf{1}\, r_f\right)}$$

and the cash position is, by construction, 0.

A more straightforward approach is to compute $\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu} - \mathbf{1}\, r_f)$ and then normalize the result to unity; as a *Mathematica* function that is

```
In[356]:= xTangentPortfolio[mnCov_, vnMean_, nRiskFree_] :=
  #/Total[#] &[Inverse[mnCov].(vnMean - nRiskFree)];
```

Example

We will continue to use the mean vector and covariance matrix of our original example. Recently, the risk-free rate has been near 0, so the value of r_f chosen is 0.5%.

```
In[357]:= nRiskFree = 0.005;
```

The tangent portfolio allocations $\mathbf{x}_{\mathcal{T}}$ are first computed using the xTangentPortfolio[ ] function defined above.

```
In[358]:= vnTangentPortfolio = xTangentPortfolio[mnCov, vnMean, nRiskFree]

Out[358]= {-0.0239846, 0.458372, 0.565613}
```

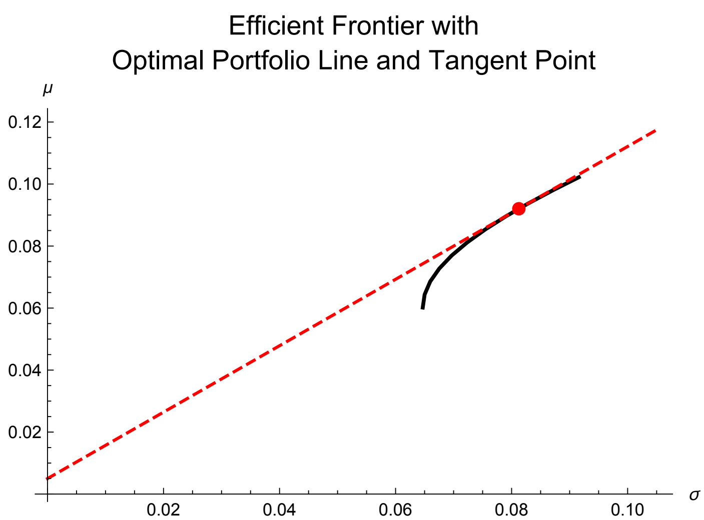
The standard deviation-mean point $\{\sigma_{\mathcal{T}},\ \mu_{\mathcal{T}}\}$ is

```
In[359]:= vnTangentPoint =
  {√vnTangentPortfolio.mnCov.vnTangentPortfolio , vnMean.vnTangentPortfolio}

Out[359]= {0.0812475, 0.0920318}
```

Using Show[ ] to combine Graphics[ ] objects, we plot the efficient frontier, the tangent point and the tangent portfolio line. The Tooltip[ ] function is used to annotate the tangent point with its value and its associated allocations.

In[360]:=
```
Show[
 ListLinePlot[
  Last /@ vuEfficientFrontier,
  PlotStyle → {Black, Thick},
  AxesOrigin → {0, 0},
  PlotLabel → Style[Column[{"Efficient Frontier with",
      "Optimal Portfolio Line and Tangent Point"}, Center], FontSize → 14],
  AxesLabel → {"σ", "μ"}
 ],
 ParametricPlot[(1 - λ) {0, nRiskFree} + λ vnTangentPoint,
  {λ, 0, 1.3}, PlotStyle → {Red, Dashed}],
 Graphics[{Red, PointSize[Large], Tooltip[Point[vnTangentPoint],
    MatrixForm /@ {vnTangentPoint, vnTangentPortfolio}]}],
 PlotRange → All
]
```

Out[360]=



Are the Portfolios $\mathcal{P}(\phi)$ “Equivalent” in Risk-Reward Terms?

MPT is founded on the notion that portfolios can be evaluated by equating risk with the standard deviation of its return and reward with the mean of its return. It also assumes that we can both lend and borrow at the risk-free rate. Thus, we can focus on maximizing the Sharpe ratio of our portfolio. The portfolio line associated with the tangent portfolio contains portfolios all of which share this maximum Sharpe ratio.

- How are the risks of a portfolio that permits short positions different from one that does not?
- How are the risks of a portfolio that permits leverage different from one that does not?
- How does the analysis which produces the tangent line change if lending and borrowing rates are different?
- How reasonable is MPT’s assumption that portfolios with the same Sharpe ratio are, in a sense, equivalent?
- What other measures of risk might be relevant?
- How would you construct a measure of risk?

The FinancialData[ ] Function

With V6 *Mathematica* added several functions which access curated data made available through Wolfram Research over the internet.

One relevant data facility relevant for us here is the FinancialData[ ] function.

In[361]:=
```
? FinancialData
```

Out[361]=

Symbol

FinancialData["*name*"] gives the last known price or value for the financial entity specified by "*name*".
FinancialData["*name*", *start*] gives a list of dates
and daily closing values for "*name*" from *start* until the current date.
FinancialData["*name*", {*start*, *end*}] gives a list of dates and daily closing values for dates from *start* to *end*.
FinancialData["*name*", {*start*, *end*, *period*}] gives a list
of dates and prices for the specified periods lying between *start* and *end*.
FinancialData["*name*", "*prop*"] gives the value of the specified property for the financial entity "*name*".
FinancialData["*name*", "*prop*", {*start*, *end*, …}] gives a
list of dates and values of a property for a sequence of dates or periods.

For a given symbol (not *Mathematica* symbol but ticker symbol represented as a string) the following properties are available:

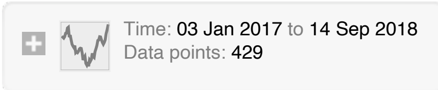
In[362]:=
```
FinancialData["IBM", "Properties"]
```

Out[362]=
```
{AdjustedClose, AdjustedHigh, AdjustedLow, AdjustedOHLC, AdjustedOHLCV,
 AdjustedOpen, AdjustedRange, Ask, AskSize, Average200Day, Average50Day,
 AverageVolume3Month, Bid, BidSize, BookValuePerShare, Change, Change200Day,
 Change50Day, ChangeHigh52Week, ChangeLow52Week, CIK, Close, Company,
 CumulativeFractionalChange, CumulativeReturn, Dividend, DividendPerShare,
 DividendYield, EarningsPerShare, EarningsYield, EBITDA, Exchange,
 FloatShares, FractionalChange, FractionalChange200Day, FractionalChange50Day,
 FractionalChangeHigh52Week, FractionalChangeLow52Week, High, High52Week,
 IPODate, Last, LastTradeSize, LatestTrade, Lookup, Low, Low52Week,
 MarketCap, MIC, Name, OHLC, OHLCV, Open, PERatio, Price, PriceToBookRatio,
 PriceToSalesRatio, Range, Range52Week, RawClose, RawHigh, RawLow, RawOHLC,
 RawOHLCV, RawOpen, RawRange, RawVolume, Return, Sector, SICCode, StandardName,
 Symbol, Volatility20Day, Volatility250Day, Volatility50Day, Volume, Website}
```

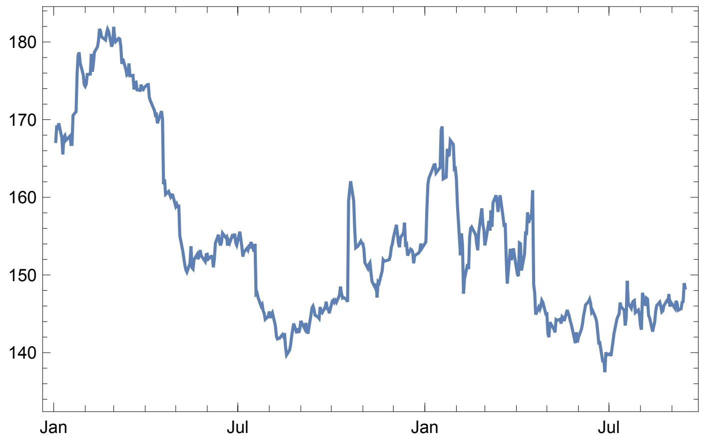
In[363]:=
```
FinancialData["IBM"]
```

Out[363]=
```
$124.27
```

In[364]:=
```
mIBM = FinancialData["IBM", {{2017, 1, 1}, {2018, 09, 14}}];
Short[mIBM, 20]
```

Out[365]//Short= TimeSeries[]

In[366]:=
```
DateListPlot[mIBM]
```

Out[366]=


In[367]:=
```
FinancialData["IBM", "MarketCap"]
```

Out[367]= $\$1.10732 \times 10^{11}$

In[368]:=

In its simplest form it reports the last known price given the ticker symbol

In[369]:=
```
FinancialData["IBM"]
FinancialData["IBM", "Close"]
```

Out[369]= $124.27

Out[370]= $124.27

Here are the open, high, low, close and volume for the day:

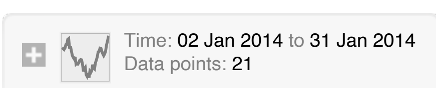
In[371]:=
```
FinancialData["IBM", "OHLCV", {{2018, 09, 14}}]
```

Out[371]= { $148.85 , $149.30 , $147.78 , $148.33 , 3 452 144 shares }

Historic data can be accessed given the ticker symbol and a date range. Only trading days are normally reported. The data field returned here is the price adjusted for splits and dividends price.

The data for a specific range of dates:

In[372]:=
```
FinancialData["IBM", "Close", {{2014, 1, 1}, {2014, 1, 31}}]
```

Out[372]= TimeSeries[]

The data for the period from a specific date to today:

In[373]:=
```
FinancialData["IBM", {2018, 9, 1}]
```

Out[373]=
```
TimeSeries[ Time: 04 Sep 2018 to 11 Dec 2020
            Data points: 573 ]
```

The data for a single date:

In[374]:=
```
FinancialData["IBM", {{2014, 2, 21}}]
```

Out[374]=
```
$182.79
```

If you request data for a specific date which is not a trading day, then a Missing[ ] value is returned:

In[375]:=
```
? Missing
```

Out[375]=

Symbol

Missing[] represents data that is missing.
Missing["*reason*"] specifies a reason for the data's being missing.
Missing["*reason*", *expr*] associates the expression *expr* with the missing data.

In[376]:=
```
FinancialData["IBM", {{2018, 7, 4}}]
```

Out[376]=
```
$139.57
```

However, if you specific a range that includes dates with valid data, the dates without data are simply left out of the return (note the holiday and weekend dates are not included):

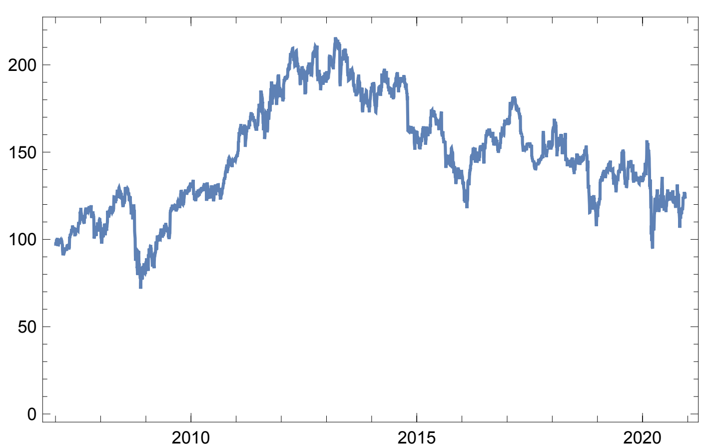
In[377]:=
```
FinancialData["IBM", {{2018, 7, 1}, {2018, 7, 14}}]
```

Out[377]=
```
TimeSeries[ Time: 02 Jul 2018 to 13 Jul 2018
            Data points: 9 ]
```

The DateListPlot[] function can be used to plot a list of {date, datum} pairs, the same format returned by calls to FinancialData[].

In[378]:=
```
DateListPlot[FinancialData["IBM", {2007, 1, 1}], Joined → True, PlotRange → All]
```

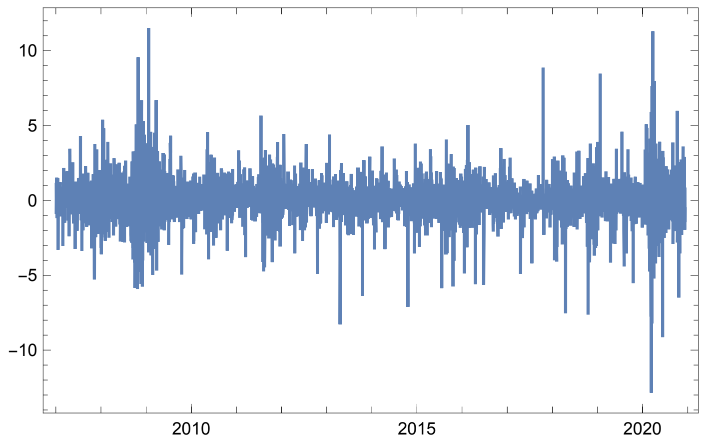
Out[378]=



Other properties are available. Here are the returns and volumes.

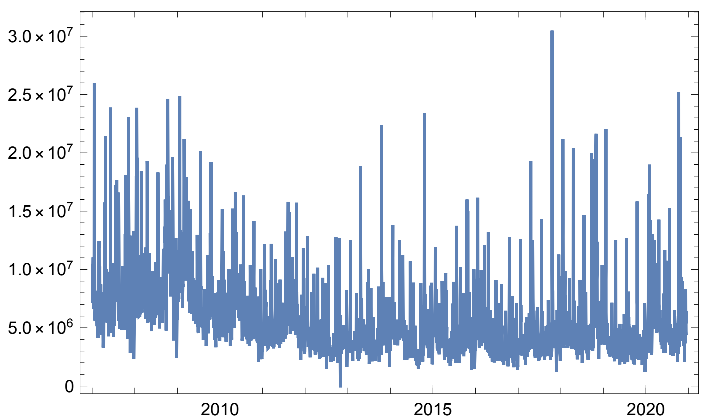
In[379]:=
```
DateListPlot[FinancialData["IBM", "Return", {2007, 1, 1}],
 Filling → 0, Joined → True, PlotRange → All]
```

Out[379]=



In[380]:=
```
DateListPlot[FinancialData["IBM", "Volume", {2007, 1, 1}],
 Joined → True, PlotRange → All]
```

Out[380]=



The composition of most important indices is also available; for example, the Dow Jones Industrial Average has the ticker symbol "^DJI".

In[381]:=
```
vsDjiTickers = FinancialData["^DJI", "Members"]
```

Out[381]=
```
{NYSE:MMM, NYSE:AXP, NASDAQ:AMGN, NASDAQ:AAPL, NYSE:BA, NYSE:CAT, NYSE:CVX,
 NASDAQ:CSCO, NYSE:KO, NYSE:DIS, NYSE:DOW, NYSE:GS, NYSE:HD, NYSE:HON, NASDAQ:INTC,
 NYSE:IBM, NYSE:JNJ, NYSE:JPM, NYSE:MCD, NYSE:MRK, NASDAQ:MSFT, NYSE:NKE,
 NYSE:PG, NYSE:CRM, NYSE:TRV, NYSE:UNH, NYSE:VZ, NYSE:V, NASDAQ:WBA, NYSE:WMT}
```

In[382]:=
```
vsDjiNames = FinancialData[#, "Name"] & /@ vsDjiTickers
```

Out[382]=
```
{3M, American Express, Amgen, Apple, Boeing, Caterpillar, Chevron, Cisco,
 Coca-Cola, Disney, Dow, Goldman Sachs, Home Depot, Honeywell, Intel, IBM,
 Johnson & Johnson, JPMorgan Chase, McDonald's, Merck & Co., Microsoft,
 Nike, Procter & Gamble, Salesforce.com, Travelers, UnitedHealth,
 Verizon Communications, Visa, Walgreens Boots Alliance Inc, Wal-Mart Stores}
```

In[383]:=
```
vnDjiMarketCaps = FinancialData[#, "MarketCap"] & /@ vsDjiTickers
```

Out[383]= $\{\$1.00379\times10^{11}, \$9.68094\times10^{10}, \$1.32385\times10^{11}, \$2.08119\times10^{12}, \$1.30028\times10^{11}, \$9.74008\times10^{10}, \$1.77967\times10^{11}, \$1.87275\times10^{11}, \$2.29268\times10^{11}, \$3.18138\times10^{11}, \$4.00546\times10^{10}, \$8.25728\times10^{10}, \$2.84815\times10^{11}, \$1.50603\times10^{11}, \$2.03794\times10^{11}, \$1.10732\times10^{11}, \$4.02647\times10^{11}, \$3.64443\times10^{11}, \$1.54804\times10^{11}, \$2.09892\times10^{11}, \$1.61235\times10^{12}, \$2.1571\times10^{11}, \$3.37105\times10^{11}, \$2.03514\times10^{11}, \$3.4184\times10^{10}, \$3.19819\times10^{11}, \$2.49734\times10^{11}, \$4.54637\times10^{11}, \$3.58778\times10^{10}, \$4.15905\times10^{11}\}$

Here's a quick report on the market capitalization on the stocks in the Dow Jones Industrial Average using the data above:

In[384]:=
```
TableForm[
 {vsDjiTickers, vsDjiNames, vnDjiMarketCaps}ᵀ,
 TableHeadings → {Automatic, {"Ticker", "Name", "MarketCap"}}
]
```

Out[384]//TableForm=

	Ticker	Name	MarketCap
1	NYSE:MMM	3M	$\$1.00379 \times 10^{11}$
2	NYSE:AXP	American Express	$\$9.68094 \times 10^{10}$
3	NASDAQ:AMGN	Amgen	$\$1.32385 \times 10^{11}$
4	NASDAQ:AAPL	Apple	$\$2.08119 \times 10^{12}$
5	NYSE:BA	Boeing	$\$1.30028 \times 10^{11}$
6	NYSE:CAT	Caterpillar	$\$9.74008 \times 10^{10}$
7	NYSE:CVX	Chevron	$\$1.77967 \times 10^{11}$
8	NASDAQ:CSCO	Cisco	$\$1.87275 \times 10^{11}$
9	NYSE:KO	Coca-Cola	$\$2.29268 \times 10^{11}$
10	NYSE:DIS	Disney	$\$3.18138 \times 10^{11}$
11	NYSE:DOW	Dow	$\$4.00546 \times 10^{10}$
12	NYSE:GS	Goldman Sachs	$\$8.25728 \times 10^{10}$
13	NYSE:HD	Home Depot	$\$2.84815 \times 10^{11}$
14	NYSE:HON	Honeywell	$\$1.50603 \times 10^{11}$
15	NASDAQ:INTC	Intel	$\$2.03794 \times 10^{11}$
16	NYSE:IBM	IBM	$\$1.10732 \times 10^{11}$
17	NYSE:JNJ	Johnson & Johnson	$\$4.02647 \times 10^{11}$
18	NYSE:JPM	JPMorgan Chase	$\$3.64443 \times 10^{11}$
19	NYSE:MCD	McDonald's	$\$1.54804 \times 10^{11}$
20	NYSE:MRK	Merck & Co.	$\$2.09892 \times 10^{11}$
21	NASDAQ:MSFT	Microsoft	$\$1.61235 \times 10^{12}$
22	NYSE:NKE	Nike	$\$2.1571 \times 10^{11}$
23	NYSE:PG	Procter & Gamble	$\$3.37105 \times 10^{11}$
24	NYSE:CRM	Salesforce.com	$\$2.03514 \times 10^{11}$
25	NYSE:TRV	Travelers	$\$3.4184 \times 10^{10}$
26	NYSE:UNH	UnitedHealth	$\$3.19819 \times 10^{11}$
27	NYSE:VZ	Verizon Communications	$\$2.49734 \times 10^{11}$
28	NYSE:V	Visa	$\$4.54637 \times 10^{11}$
29	NASDAQ:WBA	Walgreens Boots Alliance Inc	$\$3.58778 \times 10^{10}$
30	NYSE:WMT	Wal-Mart Stores	$\$4.15905 \times 10^{11}$

An index itself has its own data. Another common stock index is the S&P 500 which has the ticker symbol “^GSPC”. Plots of the value of the index and it return from 1990 to date are:

In[385]:=
```
tsGSPC = FinancialData["^GSPC", {1950, 1, 1}];
```

In[386]:=
```
Export[FileNameJoin[{NotebookDirectory[], "tsGSPC.m"}], tsGSPC]
```

Out[386]=
```
/Volumes/Files/Programming/Foundations of Quantitative Finance/Lecture 3/tsGSPC.m
```

In[387]:=
```
Head[tsGSPC]
```

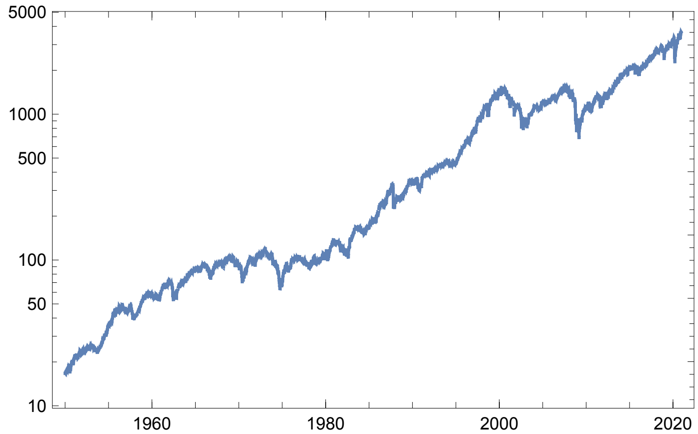
Out[387]=
```
TemporalData
```

In[388]:=
```
tsGSPC["FirstDate"]
```

Out[388]= Tue 3 Jan 1950 00:00:00 GMT−5.

In[389]:=
```
DateListLogPlot[FinancialData["^GSPC", {1950, 1, 1}], PlotRange → All]
```

Out[389]=



There are also a number of functions which rely on the functionality of FinancialData[ ]. An example is TradingChart[ ].

In[390]:=
```
? TradingChart
```

Out[390]=

Symbol

TradingChart[{{$date_1$, {$open_1$, $high_1$, low_1, $close_1$, $volume_1$}}, …}]
makes a chart showing prices and volume for each date.

TradingChart[{"*name*", *daterange*}] makes a financial chart for the financial entity "*name*" over the *daterange*.

TradingChart[{…}, {ind_1, ind_2, …}] makes a financial chart with indicators ind_1, ind_2, ….

In[391]:=
```
TradingChart[FinancialData["AAPL", "OHLCV", {2018, 1, 1}], Appearance → "OHLC"]
```

Out[391]=