MFE Bootcamp

Tobias Ingebrigtsen

August 19, 2022

Abstract

These notes are written for part 2 of the UCLA Anderson Master of Financial Engineering Bootcamp. I have pulled together many concepts that will be useful for the MFE program and tried to make example code where I believe it might be useful. The material is gathered from lectures that I have taken myself, and some are my very own notes. If you find any typos (there are likely a lot of typos) or have any feedback, then please let me know.

TI

Contents

TI

1 Linear Algebra

Motivation: As quantitative economists, we often rely on linear algebra as a tool for solving simple and complex problems. For example, linear algebra allows us to find solutions to linear systems:

$$\begin{aligned} y_1 &= \phi_1 x_{11} + \phi_2 x_{12} + ... + \phi_k x_{1k}, \\ &\vdots \\ y_n &= \phi_1 x_{n1} + \phi_2 x_{n2} + ... + \phi_k x_{nk}, \end{aligned}$$

where we are interested in the unknowns ϕ_i given series of observables x_i and y_i. An example of a system of interest is the famous Capital Asset Pricing Model:

$$r_{i,t} - r^f = \alpha_i + \beta_i (r_{m,t} - r^f),$$

where $r_{i,t}$ is the return of asset i on time t, $r_{m,t}$ is the market return on time t, and r^f is the risk-free rate of interest. In this model we try to explain the excess return of all assets by each assets sensitivity to the market (more on this later). Important things to consider when we work with these problems:

- How many solutions (if any) exists?
- If there are no closed-form solution to this system, can we still approximate it?
- How do we compute the solution?

Linear algebra can also simplify notation. Consider the simple case of computing the variance of a portfolio of two assets: We know that the total variance of two variables, in this case stock 1 and stock 2, s_1 and s_2 can be computed as

$$\text{Var}(s_1 + s_2) = Var(s_1) + Var(s_2) + 2cov(s_1, s_2).$$

Now, let ω_1 and ω_2 be the portfolio weights of stock 1 and stock 2. The portfolio variance is now:

$$\text{Var}(\omega_1 s_1 + \omega_2 s_2) = \omega_1^2 Var(s_1) + \omega_2^2 Var(s_2) + 2\omega_1 \omega_2 cov(s_1, s_2).$$

For an n-asset portfolio this becomes

$$\text{Var}(\omega_1 s_1 + ... + \omega_n s_n) = \sum_{i=1}^{n} \sum_{j=1}^{n} \omega_i \omega_j Cov(s_i, s_j).$$

By using matrix notation we get

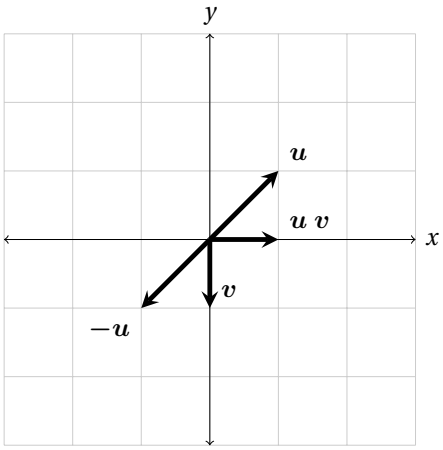
$$\text{Var}(\omega_1 s_1 + ... + \omega_n s_n) = \boldsymbol{\omega}' \boldsymbol{\Sigma} \boldsymbol{\omega},$$

where $\boldsymbol{\Sigma}$ is and $(n \times n)$ covariance matrix and $\boldsymbol{\omega}$ is a vector of weights.

TI

Vectors

A vector in R^n is an n-tuple of numbers and represent a direction and magnitude.

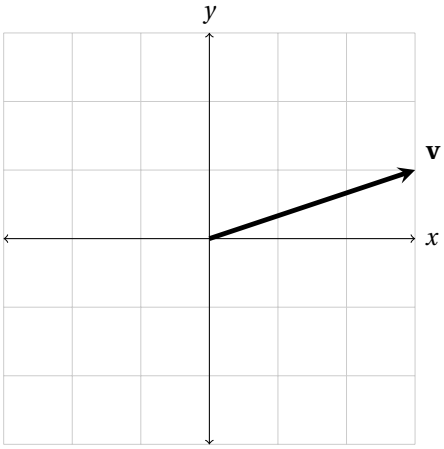


We can express the same vector in two ways:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \text{and} \quad \mathbf{x}^t = [x_1, x_2, ..., x_3].$$

Example: In R^2 :

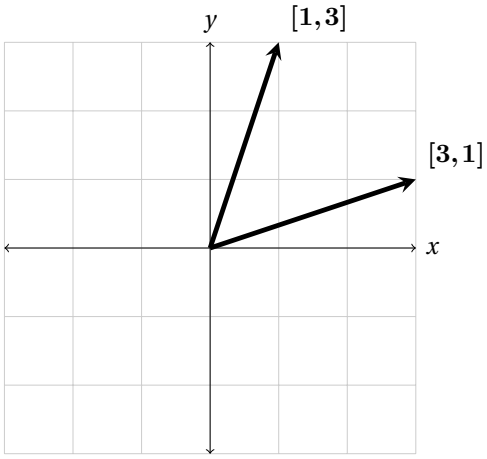
$$\mathbf{v} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$



The ordering matters:

$$\begin{bmatrix} 3 \\ 1 \end{bmatrix} \neq \begin{bmatrix} 1 \\ 3 \end{bmatrix}$$

TI



What can we do with vectors?

An n-vector, **u**, can be multiplied by a number c. This is called scalar multiplication and yields a new vector n-vector, $c\mathbf{u}$.

Example: (n=3) Let $\mathbf{u} = \begin{bmatrix} 2 \\ -1 \\ 0 \end{bmatrix}$ and $c = 3$. Then,

$$3\mathbf{u} = 3\begin{bmatrix} 2 \\ -1 \\ 0 \end{bmatrix} = \begin{bmatrix} 3\times 2 \\ 3\times(-1) \\ 3\times 0 \end{bmatrix} = \begin{bmatrix} 6 \\ -3 \\ 0 \end{bmatrix}.$$

We can also add and subtract to vectors **u**, **v**:

$$\mathbf{u}' + \mathbf{v}' = (u_1 + v_1, u_2 + v_2, +...+, u_n + v_n),$$
$$\mathbf{u}' - \mathbf{v}' = (u_1 - v_1, u_2 - v_2, -...-, u_n - v_n).$$

Definition 1.1: Vector Space

Formally, a vector space over $\mathbb{R}$ is a tuple $(V, +, \cdot)$ where V is a set, $+\colon V \times V \to V$ and $\cdot\colon \mathbb{R} \times V \to V$ such that

1. $x + y = y + x$ for all $x, y \in V$,
2. $(x + y) + z = x + (y + z)$ for all $x, y \in V$,
3. there exist $0 \in V$ such that $x + 0$=x for all $x \in V$,
4. for every $x \in V$, there exist $-x \in V$ such that $x + (-x) = 0$,
5. $1 \cdot x = x$ for all $x \in V$,
6. $(ab) \cdot x = a \cdot (b \cdot x)$ for all $a, b \in \mathbb{R}$ and $x \in V$,
7. $a \cdot (x + y) = a \cdot x + a \cdot y$ for all $a \in \mathbb{R}$ and $x, y \in V$.
8. $(a + b) \cdot x = a \cdot x + b \cdot x$ for all $a, b \in \mathbb{R}$ and $x \in V$

TI

Example: Let $n \in \mathbb{N}$. Let

$$V = \mathbb{R}^n = \{(x_1, ..., x_n) | x_1, ..., x_n \in \mathbb{R}\}.$$

Given $(x_1, ..., x_n), (y_1, ..., y_n) \in V$ and $a \in \mathbb{R}$, define $(x_1, ..., x_n) + (y_1, ..., y_n) \in V$ and $a \cdot (x_1, ..., x_n) \in V$ by

$$(x_1, ..., x_n) + (y_1, ..., y_n) = (x_1 + y_1, ..., x_n + y_n)$$

and

$$a \cdot (x_1, ..., x_n) = (ax_1, ..., ax_n).$$

Then $(V, +, \cdot)$ is a vector space.

1.1 Exercise

1. Let S be a nonempty set. Let $V = \{f : S \to \mathbb{R}\}$. Given $f, g \in V$ and $a \in \mathbb{R}$, define $f + g \in V$ and $a \times f \in V$ by

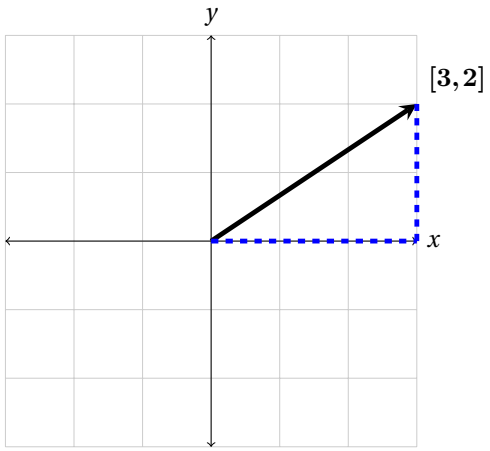
   $$(f + g)(s) = f(s) \quad \text{for } s \in S,$$

   and

   $$(a \cdot f)(s) = af(s) \quad \text{for } s \in S.$$

   Show that $(V, +, \cdot)$ is a vector space.

1.2 Length of a vector, inner product, and the Cauchy-Schwarz inequality



Lenght of a vector: This one is easy, we have a vector of length 2 and we remember the Pythagorean theorem:

$$c^2 = b^2 + a^2.$$

For a vector $\mathbf{x}$ of length n:

$$\|\mathbf{x}\| = \sqrt{\sum_{i=1}^{n} x_i^2}.$$

This is called the **Euclidean norm**. We can verify it for the example above where $\mathbf{x}' = (3, 2) : \|\mathbf{x}\| = \sqrt{3^2 + 2^2}$, which is equal to what we learned using the Pythagorean theorem. Also, for any two points $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, $\|\mathbf{x} - \mathbf{y}\|$ measures the distance between $\mathbf{x}$ and $\mathbf{y}$.

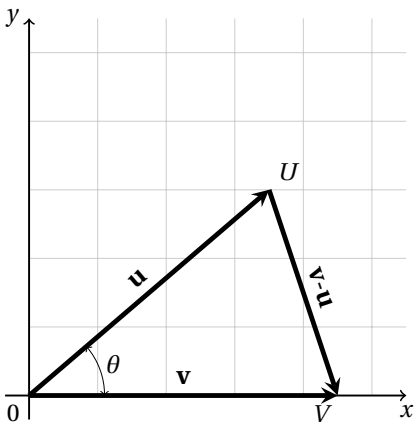
Inner product of two vectors $\mathbf{v}, \mathbf{u} \in \mathbb{R}^n$ is defined by $\mathbf{v} \cdot \mathbf{u} = \sum_{i=1}^{n} x_i y_i$. For example, let ω be a vector of portfolio weights and $\mathbf{p}$ be a price vector for every asset in the portfolio. The value of the portfolio is then given by $\omega \cdot \mathbf{p} = \sum_{i=1}^{n} \omega_i p_i$.

Cauchy-Schwarz inequality:

$$\mathbf{u} \cdot \mathbf{v} \leq \|\mathbf{u}\| \cdot \|\mathbf{v}\|$$

In fact,

$$\mathbf{u} \cdot \mathbf{v} = \|\mathbf{u}\| \cdot \|\mathbf{v}\| \cdot cos(\theta)$$



According to Pythagora's theorem, the angle θ between two vectors $\mathbf{u}$ and $\mathbf{v}$ is a right angle ($= 90°$) if and only if $(OU)^2 + (OB)^2 = (VU)^2$, or $\|\mathbf{u}\|^2 + \|\mathbf{u}\|^2 = \|\mathbf{v} - \mathbf{u}\|^2$. This implies that $\theta = 90°$ if and only if

$$\mathbf{v} \cdot \mathbf{v} + \mathbf{u} \cdot \mathbf{u} = (\mathbf{v} - \mathbf{u})(\mathbf{v} - \mathbf{u}) = \mathbf{u} \cdot \mathbf{u} + \mathbf{v} \cdot \mathbf{v} - \mathbf{v} \cdot \mathbf{u} - \mathbf{u} \cdot \mathbf{v}.$$

Because of symmetry, $\mathbf{u} \cdot \mathbf{v} = \mathbf{v} \cdot \mathbf{u}$, so we need $2\mathbf{u} \cdot \mathbf{v} = 0$ for the equality to hold, and so $\mathbf{u} \cdot \mathbf{v} = 0$. When the angle between two vectors $\mathbf{u}$ and $\mathbf{v}$ is $90°$, we say that they are **orthogonal.** If two vectors $\mathbf{u}$ and $\mathbf{v}$ are orthogonal we write $\mathbf{u} \perp \mathbf{v}$. Here we proved that two vectors in $\mathbb{R}^2$ or $\mathbb{R}^3$ are orthogonal if and only if their inner product is 0. For vectors in $\mathbb{R}^n$, we define orthogonality between $\mathbf{u}$ and $\mathbf{v}$ as

$$\mathbf{u} \perp \mathbf{v} \iff \mathbf{u} \cdot \mathbf{v} = 0.$$

Example: Suppose we have n observations of a commodity's price and quantity demanded $(p_1, d_1), (p_2, d_2)$, ..., (p_n, d_n). Define the means as

$$\overline{\mathbf{p}} = \frac{1}{n} \sum_{i=1}^{n} p_i, \quad \overline{\mathbf{d}} = \frac{1}{n} \sum_{i=1}^{n} d_i,$$

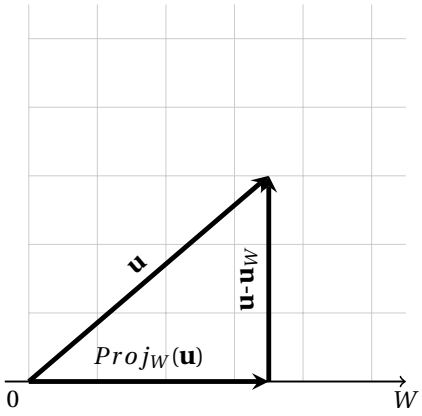
and

$$\mathbf{v} = (p_1 - \overline{\mathbf{p}}, p_2 - \overline{\mathbf{p}}, ..., p_n - \overline{\mathbf{p}}), \quad \mathbf{u} = (d_1 - \overline{\mathbf{d}}, d_2 - \overline{\mathbf{d}}, ..., d_n - \overline{\mathbf{d}}).$$

Then the **correlation coefficient** ρ can be computed as

$$\rho = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{v}\| \cdot \|\mathbf{u}\|} = cos(\theta).$$

Projection



$\mathbf{u}_W$ is such that

1. $\mathbf{u}_W$ is parallell with W: There exist a scalar c such that $\mathbf{u}_W = cW$,
2. $\mathbf{u}_w \perp (\mathbf{u} - \mathbf{u}_W)$

Now we can find c, i.e. find $\mathbf{u}_w$. 2. means that

$$\begin{aligned} \mathbf{u}_W \cdot (\mathbf{u} - \mathbf{u}_W) &= 0, \\ \mathbf{u}_W \cdot \mathbf{u} - \mathbf{u}_W \cdot \mathbf{u}_W &= 0, \quad \text{(Distributive law (VS 7))} \\ \mathbf{u}_W \cdot \mathbf{u} &= \|\mathbf{u}_W\|^2. \quad (*) \end{aligned}$$

From 1. we have $\mathbf{u}_W = cW$. Insert $(*)$ into 1. to get $cW \cdot \mathbf{u} = \|cW\|^2 = c^2\|W\|^2$. Solve for c:

$$c = \frac{W \cdot \mathbf{u}}{\|W\|^2}.$$

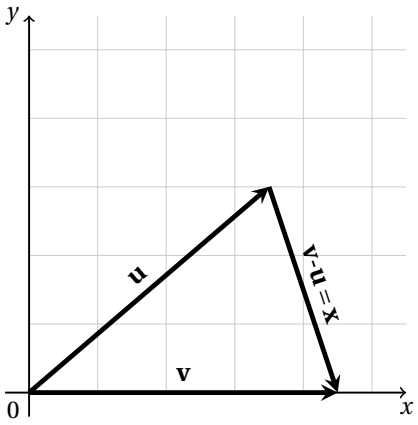
Conclusion:

$$Proj_W(\mathbf{u}) = \mathbf{u}_W = \frac{W \cdot \mathbf{u}}{\|W\|^2} \cdot W.$$

Example: Linear regression as a projection (later).

Linear independence of vectors, basis and linear subspace.

We have already seen that we can create a new vector **x** by two other vectors:



A natural question then becomes: how many other vectors can we create by linear combinations of **u** and **v**?

Definition 1.2: Span

Assume $\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_m$ are n-vectors. Then, span$(\mathbf{v}_1, ..., \mathbf{v}_m)$ are all the vectors that are linear combinations of $\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_m$ $(c_1\mathbf{v}_1, c_2\mathbf{v}_2, ..., c_m\mathbf{v}_m)$.

Example: (m=1) span$(\mathbf{v}_1)$ are all vectors $c_1\mathbf{v}_1$ where $c_1 \in \mathbb{R}$.

Exercise: (m=3) $\mathbf{v}_1 = \begin{bmatrix}1\\0\\0\end{bmatrix}, \mathbf{v}_2 = \begin{bmatrix}0\\1\\0\end{bmatrix}, \mathbf{v}_3 = \begin{bmatrix}2\\3\\0\end{bmatrix}$. What is span$(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$?

Definition 1.3: Linear dependence

The vectors $\{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_m\}$ are **linearly dependent** if one or more of the vectors can be written as a linear combination of the others.

Example: $\mathbf{v}_1 = \begin{bmatrix}1\\0\\0\end{bmatrix}, \mathbf{v}_2 = \begin{bmatrix}0\\1\\0\end{bmatrix}, \mathbf{v}_3 = \begin{bmatrix}2\\3\\0\end{bmatrix}$ are linear dependent since $\mathbf{v}_2 = \frac{1}{3}\mathbf{v}_3 - \frac{2}{3}\mathbf{v}_1$.

Definition 1.4: Basis

If the vectors $\{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_m\}$ are not linearly dependent we say that they are **linearly independent**. Then the vectors $\{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_m\}$ form a **basis** for span$(\{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_m\})$.

Example: Consider the vectors $\mathbf{v}_1 = \begin{bmatrix}1\\0\\0\end{bmatrix}, \mathbf{v}_2 = \begin{bmatrix}0\\1\\0\end{bmatrix}, \mathbf{v}_3 = \begin{bmatrix}2\\3\\0\end{bmatrix}$. $\{\mathbf{v}_1, \mathbf{v}_2\}$ are linearly independent (why?) therefore forms a basis for the xy-plane. Note: $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ spans $\mathbb{R}^2$ but is not a basis since they are linearly dependent.

Definition 1.5: Dimension

The number of vectors is in the basis is called the dimension of the linear subspace.

Example: Consider the vectors $\mathbf{v}_1 = \begin{bmatrix}1\\0\\0\end{bmatrix}, \mathbf{v}_2 = \begin{bmatrix}0\\1\\0\end{bmatrix}, \mathbf{v}_3 = \begin{bmatrix}2\\3\\0\end{bmatrix}$. The dimension of span$(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$ is 2.

Linear systems

$$\begin{aligned} 2x_1 - 4x_2 + 2x_3 &= 12, \\ 3x_1 - 5x_2 - 2x_3 &= 5, \\ -4x_1 + 7x_2 + 4x_3 &= -2. \end{aligned}$$

TI

Write $\mathbf{u}_1 = \begin{bmatrix} 2 \\ 3 \\ -4 \end{bmatrix}, \mathbf{u}_2 = \begin{bmatrix} -4 \\ -5 \\ 7 \end{bmatrix}, \mathbf{u}_3 = \begin{bmatrix} 2 \\ -2 \\ 4 \end{bmatrix}, \mathbf{v} = \begin{bmatrix} 12 \\ 5 \\ -2 \end{bmatrix}$. The linear system now becomes

$$x_1\mathbf{u}_1 + x_2\mathbf{u}_2 + x_3\mathbf{u}_3 = \mathbf{v}.$$

Question: Does the above system have a solution? (i.e. is $\mathbf{v}$ in $\text{span}(\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3)$?) → Gaussian Elimination.

Definition 1.6: Gaussian elimination

- a method for solving all linear systems.

1. Write the linear system as a matrix
2. From this matrix, create a new matrix in echelon form using elementary row operations
3. Translate the new matrix to a simpler linear system.
4. Solve this system using backwards substitution.

Definition 1.7: Echelon form

1. 0-rows are in the bottom rows of the matrix
2. Every pivot (leading coefficient) is longer to the right than the pivots in the rows above it (a pivot is the first number in a row that is non-zero.)

Definition 1.8: Free variables

The variables that corresponds to columns without pivots (but not the last column).

Example:

$$\begin{bmatrix} 6 & 4 & 0 & 1 \\ 0 & 4 & 3 & 3 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The above matrix is in echelon form, where the blue numbers are **pivots** and the variable corresponding to the third column is free.

Exercise: Write the linear system

$$\begin{aligned} 2x_1 - 4x_2 + 2x_3 &= 12, \\ 3x_1 - 5x_2 - 2x_3 &= 5, \\ -4x_1 + 7x_2 + 4x_3 &= -2, \end{aligned}$$

in matrix form and solve it using Gaussian elimination.

TI

1.3 The solutions

Every linear system has either

1. No solution $\iff$ pivot in the last column.
2. A unique solution $\iff$ pivot in all columns except the last.
3. Infinitely many solutions $\iff$ no pivot in last column and at least one free variable.

1.4 Rank of a matrix

The rank of a matrix is the number of pivots in the echelon form.

Example:

$$\mathbf{A} = \begin{bmatrix} 2 & 4 & -1 & 3 \\ -4 & -3 & 9 & 5 \\ 6 & 12 & -3 & 9 \end{bmatrix} \sim \begin{bmatrix} 2 & 4 & -1 & 3 \\ 0 & 5 & 7 & 11 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Here we have to pivots, so rk(A)=2.

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 5 \\ 0 & 0 & 1 & 9 \end{bmatrix} \sim \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Here we have three pivots, so rk(B)=3.

```
from numpy.linalg import matrix_rank
A = np.matrix('2 4 -1 3;-4 -3 9 5;6 12 -3 9')

A
Out[10]:
matrix([[ 2,  4, -1,  3],
        [-4, -3,  9,  5],
        [ 6, 12, -3,  9]])

matrix_rank(A)
Out[11]: 2

B = np.eye(3)

B
Out[16]:
array([[1., 0., 0.],
       [0., 1., 0.],
       [0., 0., 1.]])

matrix_rank(B)
Out[17]: 3
```

1.5 Applications

Let $\{\mathbf{u}_1, \mathbf{u}_2, ...\mathbf{u}_m\}$ be n-vectors.

1. The vectors are linearly independent $\iff x_1\mathbf{u}_1 + x_2\mathbf{u}_2 + ... + x_m\mathbf{u}_m = 0$ only has the trivial solution $(x_1 = x_2 = ... = x_m = 0)$. [a system on the form $x_1\mathbf{u}_1 + x_2\mathbf{u}_2 + ... + x_m\mathbf{u}_m = 0$ is called a homogeneous system]
2. The vectors corresponding to pivot positions in the echelon form gives a basis for span$(\mathbf{u}_1, \mathbf{u}_2, ...\mathbf{u}_m)$

Matrices

Definition 1.9: Transpose

The transpose of an (m×n)-matrix A gives an (n×m) matrix A' where the first column in A becomes the first row of A', etc.

Example: $A = \begin{bmatrix} 1 & 0 & -3 \\ 2 & 7 & 5, \end{bmatrix}, \quad A' = \begin{bmatrix} 1 & 2 \\ 0 & 7 \\ -3 & 5 \end{bmatrix}.$

```
import numpy as np

A = np.matrix('1 0 -3; 2 7 5')


Out[4]:
matrix([[ 1,  0, -3],
        [ 2,  7,  5]])

A.T
Out[5]:
matrix([[ 1,  2],
        [ 0,  7],
        [-3,  5]])
```

What can we do with matrices? Almost the same operations as with scalars, however **the order is important. Example:**

$$\begin{aligned} A(B+C) &= AB + AC, \\ (A+B)^2 &= A^2 + AB + BA + B^2, \\ (A')' &= A, \\ c(A+B) &= cA + cB, \quad etc. \end{aligned}$$

Important: AB is not necessarily equal to BA:

TI

```
import numpy as np
A = np.matrix(np.random.randint(0,10, size=(3, 3)))
B = np.matrix(np.random.randint(0,10, size=(3, 3)))

A
Out[28]:
matrix([[0, 0, 1],
        [3, 0, 4],
        [5, 7, 9]])

B
Out[29]:
matrix([[8, 0, 8],
        [4, 5, 9],
        [3, 0, 4]])

C = np.matmul(A,B)
D = np.matmul(B,A)
C
Out[32]:
matrix([[  3,   0,   4],
        [ 36,   0,  40],
        [ 95,  35, 139]])

D
Out[33]:
matrix([[ 40,  56,  80],
        [ 60,  63, 105],
        [ 20,  28,  39]])
```

Linear systems using matrix notation (m equations, n unknowns)

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1,\\
a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2,\\
&\vdots\\
a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &= b_m.
\end{aligned}
$$

Write

$$
A = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ \vdots & & \ddots & \vdots \\ a_{11} & a_{m2} & \ldots & a_{mn} \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}.
$$

We can now write the linear system above as

$$
A\mathbf{x} = \boldsymbol{b}.
$$

Definition 1.10: The identity matrix

Let $A = (a_{ij})$ be a **quadratic** $(n \times n)$ matrix where $a_{ij} = 1$ for $i = j$ and 0 otherwise:

$$I = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

This is called the **identity matrix**. We then have that

$$AI = A = IA$$

Example:

```
import numpy as np
A = np.matrix(np.random.randint(0,10, size=(3, 3)))
I = np.eye(3)
A
Out[36]:
matrix([[5, 8, 7],
        [7, 1, 2],
        [5, 2, 1]])

I
Out[37]:
array([[1., 0., 0.],
       [0., 1., 0.],
       [0., 0., 1.]])

np.matmul(A,I)
Out[38]:
matrix([[5., 8., 7.],
        [7., 1., 2.],
        [5., 2., 1.]])

np.matmul(I,A)
Out[39]:
matrix([[5., 8., 7.],
        [7., 1., 2.],
        [5., 2., 1.]])
```

Definition 1.11: Invertability of a matrix

An $(n \times n)$ matrix A is invertible if there exist an $(n \times n)$ matrix B such that

$$AB = I = BA$$

If A is invertible, B is unique such that $AB = I = BA$.

```
import numpy as np
A = np.matrix(np.random.randint(0,10, size=(3, 3)))
B = np.linalg.inv(A)

A
Out[41]:
matrix([[1, 6, 1],
        [8, 7, 6],
        [4, 1, 7]])

B
Out[42]:
matrix([[-0.25443787,  0.24260355, -0.17159763],
        [ 0.18934911, -0.01775148, -0.01183432],
        [ 0.1183432 , -0.13609467,  0.24260355]])

np.matmul(A,B)
Out[43]:
matrix([[ 1.00000000e+00, -2.77555756e-17,  0.00000000e+00],
        [ 2.22044605e-16,  1.00000000e+00,  5.55111512e-17],
        [ 5.55111512e-17, -2.77555756e-17,  1.00000000e+00]])

np.matmul(B,A)
Out[44]:
matrix([[ 1.00000000e+00,  1.38777878e-16,  0.00000000e+00],
        [-6.93889390e-18,  1.00000000e+00, -5.20417043e-18],
        [-1.11022302e-16, -5.55111512e-17,  1.00000000e+00]])
```

The result above is very useful in economics since it allows us to solve linear systems very easily. Remember that we can write a linear system on the form

$$A\mathbf{x} = \boldsymbol{b}.$$

We might ask ourselves, what **x** will solve the system above? That is, we want to isolate **x** on the left side of the equation. Let B be the inverse of A (we often write A^{-1}), then:

$$\begin{aligned} A\mathbf{x} &= \boldsymbol{b}, \\ BA\mathbf{x} &= B\boldsymbol{b}, \\ I\mathbf{x} &= B\boldsymbol{b}. \end{aligned}$$

$\Rightarrow$ a very easy way to solve a system!

Definition 1.12: Determinants

For every (n×n) matrix A there exist a number det(A) (often $|A|$) so that the following is true:

1. det(AB) = det(A)· det(B)
2. det(A') = det(A)

Theorem: A invertible $\iff$ det(A)$\neq 0$.

If the determinant of A is not zero, then we say that is **nonsingular**.

Useful results: for an (n×n) matrix A where det(A)= 0:

1. The columns of A are dependent vectors in $\mathbb{R}^n$
2. The rows of A are dependent vectors in $\mathbb{R}^n$
3. A is not invertible

Definition 1.13: Eigenvalues and eigenvectors

The (n×n) matrix A has the eigenvalue λ (a number) if the equation

$$A\mathbf{x} = \lambda\mathbf{x}$$

has a non-trivial solution (i.e. $\mathbf{x} = \boldsymbol{b} \neq \mathbf{0}$). Then the vector $\boldsymbol{b}$ is an eigenvector to A with eigenvalue λ.

Example:

$$A = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}, \quad \text{then} \quad A\mathbf{x} = 3\mathbf{x},$$

so $\lambda = 3$ is an eigenvalue to A and all 2-vectors are eigenvectors to A with eigenvalue 3.

Example:

$$A = \begin{bmatrix} 4 & 2 \\ -3 & -1 \end{bmatrix}.$$

For what number λ does the system

$$A\mathbf{x} = \lambda\mathbf{x}$$

have non-trivial solutions?

$$\text{i.e.} \quad \begin{bmatrix} 4 & 2 \\ -3 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \lambda x_1 \\ \lambda x_2 \end{bmatrix}$$

$$\text{i.e.} \quad \begin{cases} 4x_1 + 2x_2 = \lambda x_1 \\ -3x_1 - x_2 = \lambda x_2 \end{cases}$$

$$\text{i.e.} \quad \begin{cases} (4-\lambda)x_1 + 2x_2 = 0 \\ -3x_1 - (1+\lambda)x_2 = 0 \end{cases}$$

$$\text{i.e.} \quad \underbrace{\begin{bmatrix} 4-\lambda & 2 \\ -3 & -(1+\lambda) \end{bmatrix}}_{B} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Rightarrow \text{ a homogeneous system!}$$

Generally: A homogeneous system $B\mathbf{x} = \mathbf{0}$ has non-trivial solutions $\iff$ det(B)= 0. **Question:** Why? Therefore,

$$|B| = 0 \iff (4-\lambda)(-(1+\lambda)) - 2 \cdot (-3) = \lambda^2 - 3\lambda + 2 = 0.$$

We solve this second-order equation to get

$$\lambda = \frac{-(-3) \pm \sqrt{(-3)^2 - 4 \cdot 2}}{2} = \begin{cases} 1 \\ 2 \end{cases}$$

So $\lambda = 1$ and $\lambda = 2$ are the eigenvalues of A.

Exercise: Find the eigenvectors of A.

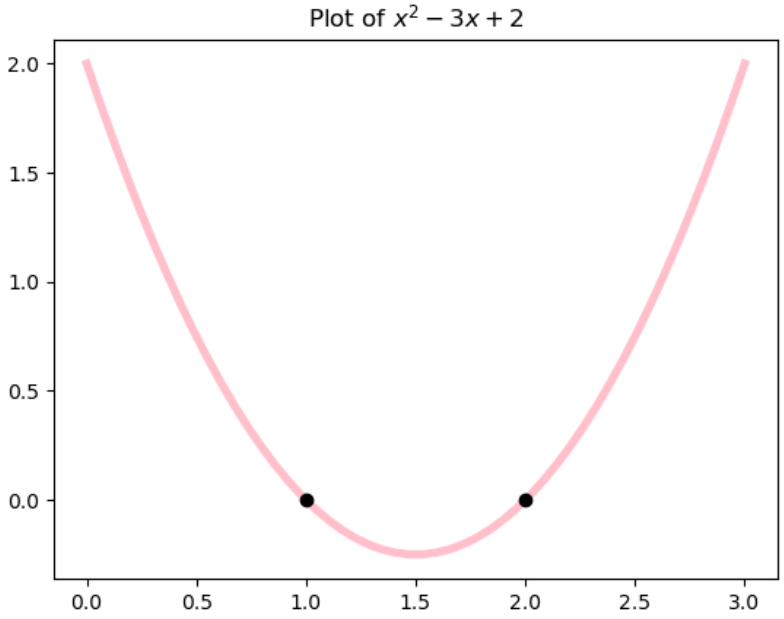


Figure 1: The roots are given by the black dots

1.5.1 Sidenote: Root-finding algorithms

In the previous example we solved a second-order equation of the form

$$x^2 - 3x + 2 = 0.$$

Although these can be straight-forward computed by hand, it's easy to make mistakes (and who solves problems by hand these days anyways?). Moreover, this becomes **very** complicated as the degree of the polynomial increase. A useful tool is therefore a root-finder. A root-finding algorithm is an algorithm for finding zeroes ("roots") of continuous functions.

```
import numpy as np

# Find the roots of f(x) = x^2 -3x +2
coeff = [1,-3,2]
np.roots(coeff)

Out[7]: array([2., 1.])
# The roots are 2 and 1.

# The figure
import matplotlib.pyplot as plt
def f(x):
    return (x**2 -3*x +2 )

x = np.linspace(0,3,50)
y = f(x)

plt.figure()
```

```
plt.plot(x,y,c='pink',linewidth=4, markevery=[1,2])
plt.plot(1,0,"—o",c='black')
plt.plot(2,0,"—o",c='black')
plt.title('Plot of $x^2—3x +2$')
```

1.6 Quadratic forms

Quadratic forms are polynomials where all terms are of degree two. Examples:

1. $Q_1(\mathbf{x}) = 3x_1^2 + 8x_1x_2 + 5x^2$
2. $Q_2(\mathbf{x}) = x_1^2 - 3x_2^2 + 5x_3^2 + 7x_2x_3 + 9x_1x_3 - 11x_1x_2$

For every quadratic form $Q(\mathbf{x})$ in n variables there exist a symmetrix $(n \times n)$ matrix A such that

$$Q(\mathbf{x}) = \mathbf{x}'A\mathbf{x}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \text{so } \mathbf{x}' = [x_1, x_2, \ldots, x_n].$$

Example:

$$A = \begin{bmatrix} 3 & 4 \\ 4 & 5 \end{bmatrix}.$$

$$\mathbf{x}'A\mathbf{x} = [x_1, x_2] \begin{bmatrix} 3 & 4 \\ 4 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 3x_1^2 + 8x_1x_2 + 5x_2^2 = Q(\mathbf{x}).$$

1.7 Definiteness of quadratic forms

Let $Q(\mathbf{x})$ be a quadratic form with symmetric matrix A.

Definition 1.14: Definiteness

$Q(\mathbf{x})$ is positive semidefinite definite if $Q(\mathbf{x}) \geq 0$ for all values of $\mathbf{x}$ (and positive definite if $Q(\mathbf{x}) > 0$ for all $x \neq 0$.)

$Q(\mathbf{x})$ is negative semidefinite definite if $Q(\mathbf{x}) \leq 0$ for all values of $\mathbf{x}$ (and negative definite if $Q(\mathbf{x}) < 0$ for all $x \neq 0$.)

If $Q(\mathbf{x})$ is neither positive or negative semidefinite, then it is indefinite.

Example:

$$Q(\mathbf{x}) = 3x_1^2 + 5x_2^2 > \forall \mathbf{x} \neq \mathbf{0},$$

i.e. $Q(\mathbf{x})$ is positive definite (and positive semidefinite).

What about $Q(\mathbf{x}) = 3x_1^2 + 8x_1x_2 + 5x_2^2$? $8x_1x_2$ is negative if x_1 and x_2 have opposite signs - it's not clear when we have crossterms in $Q(\mathbf{x})$.

Theorem Let $\lambda_1, \lambda_2, ..., \lambda_n$ be the eigenvalues of A. Then,
$Q(\mathbf{x})$

$$\begin{aligned} \text{positive semidefinite} &\iff \text{all } \lambda_i \geq 0 \\ \text{positive definite} &\iff \text{all } \lambda_i > 0 \\ \text{negative semdefinite} &\iff \text{all } \lambda_i \leq 0 \\ \text{negative definite} &\iff \text{all } \lambda_i < 0 \\ \text{indefinite} &\iff \lambda_i < 0 \text{ and } \lambda_j > 0 \text{ for some } i, j. \end{aligned}$$

2 Optimization

We are often faced with optimization problems in economics. A classic example is consumption and investment decisions, where an agent can choose to either consume some (or even all) of her wealth, and invest the remaining in assets which she can use to consume next period. This gives rise to yet another problem; what portfolio should she invest in? Should it be equal-weighted in risky stocks? Should she perhaps allocate some of her wealth to risk-free T-bills? These questions can be answered using optimization.

Definition 2.1: Extreme points

If $f(x)$ has domain D then

$$c \in D \text{ is a } \textbf{maximum point} \text{ for } f \iff f(x) \leq f(c) \text{ for all } x \in D,$$
$$d \in D \text{ is a } \textbf{minimum point} \text{ for } f \iff f(x) \geq f(d) \text{ for all } x \in D.$$

If the value of f at c is strictly larger than at any other point in D, then c is a **strict maximum point**. Similarly,d is a **strict minimum point** if $f(x) > f(d)$ for all $x \in D$, $x \neq 0$.

Note: If f is any function with domain D: $f(x) \leq f(c)$ for all x in D iff $-f(x) \geq -f(c)$ for all x in D. Implication: c maximizes f in D iff c minimizes $-f$ in D. This is sometimes useful in numerical optimization if the optimizer only minimize.

Example: find the extreme points for $g(x) = 3 - (x-2)^2$.

Solution:

Since $(x-2)^2 \geq 0$ for all x, it follows that $f(x) \leq 3$ for all x. But $f(x) = 3$ when $(x-2)^2 = 0$ at $x = 2$. Therefore, $x = 2$ is a **maximum point** for f. Because $f(x) \to -\infty$ as $x \to \pm\infty$, f has no minimum.

Example: find the extreme points for $f(x) = \sqrt{x-5} - 100, \quad x \geq 5$.

Solution:

Because $\sqrt{x-5}$ is ≥ 0 for all $x \geq 5$, it follows that $f(x) \geq -100$ for all $x \geq 5$. Since $f(5) = -100$, we conclude that $x = 5$ is a **minimum point**. Since $f(x) \to \infty$ as $x \to \infty$, f has no maximum.

Definition 2.2: Derivative of a function

The derivative of a function f at point a, denoted by $f'(a)$, is given by the formula

$$f'(a) = \lim_{h \to 0} \frac{f(a+h) - f(a)}{h}.$$

$f'(a)$ = slope of the tangent to the curve $y = f(x)$ at the point $(a, f(a))$.

Definition 2.3: Necessary first-order condition

Suppose that a function f is differentiable in an interval I and that c is an interior point for I. For $x = c$ to be a maximum or minimum point for f in I, a necessary condition is that it is a

stationary point for f, i.e. that $x = c$ satisfies the equation

$$f'(x) = 0.$$

We call this the **first-order condition.**

Definition 2.4: First-derivative test for max/min

If $f'(x) \geq 0$ for $x \leq c$ and $f'(x) \leq 0$ for $x \geq c$, then c is a maximum point for f. If $f'(x) \leq 0$ for $x \leq c$ and $f'(x) \geq 0$ for $x \geq c$, then c is a minimum point for f.

Exercise

Consider the function f defined for all x by

$$f(x) = e^{2x} - 5e^{x} + 4$$

(a) Find the zeros of $f(x)$ and compute its derivative $f'(x)$.
(b) Find the intervals where f increases and decreases, and determine possible extreme points and values.
(c) Examine the limit of $f(x)$ as $x \to -\infty$. Sketch the graph of f.
(d) Repeat (a)-(c) on the computer using Python. Try to avoid pre-programmed routines but rather write your own functions.

Definition 2.5: Second-order condition

Suppose that f is a $\mathscr{C}^2$ function (zeroth, first, and second derivative of f are continuous) and x be an interior point of I. Then:

(a) $f'(x) = 0$ and $f''(x) < 0 \Rightarrow$ c = x is a **strict** local maximum point
(b) $f'(x) = 0$ and $f''(x) > 0 \Rightarrow$ c = x is a **strict** local minimum point
(c) $f'(x) = 0$ and $f''(x) = 0 \Rightarrow$?

Exercise

(a) Classify the stationary points of

$$f(x) = 1/9x^3 - 1/6x^2 - 2/3x + 1$$

by using the second-derivative test.
(b) Find a two examples that satisfies (c) while being different local extreme points.

Definition 2.6: Extreme Value Theorem

Suppose that f is a continuous function over a closed and bounded interval $[a, b]$. Then there exist a point d in $[a, b]$ where f has a minimum, and a point c in $[a, b]$ where f has a maximum, so that

$$f(d) \leq f(x) \leq f(c) \quad \text{for all } x \text{ in } [a, b].$$

This result is useful in **constrained optimization problems** where we consider optimization

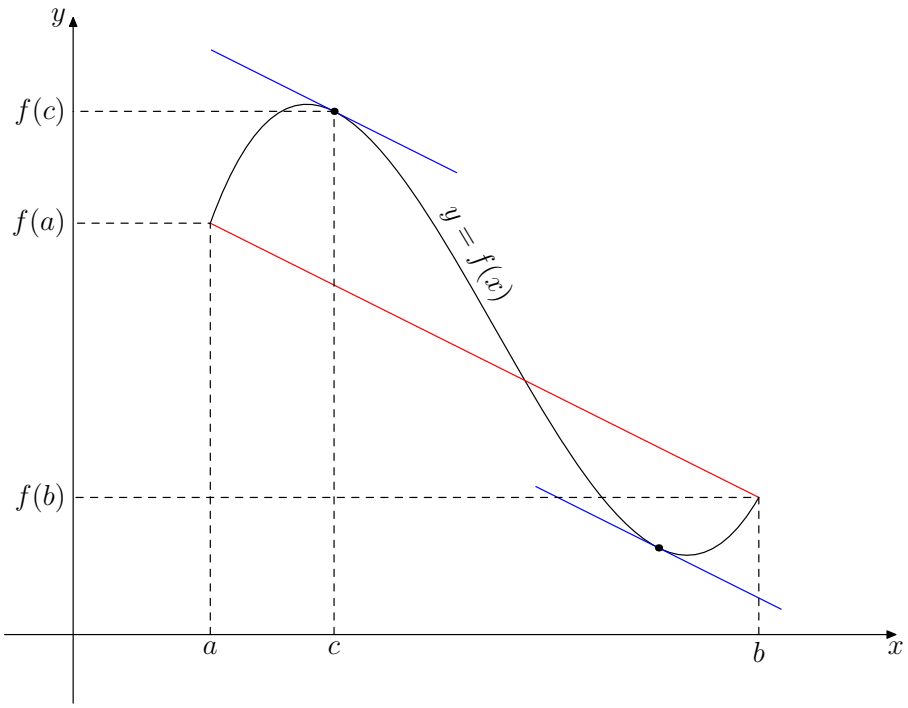
problems over closed and bounded intervals.

Definition 2.7: The Mean Value Theorem

If f is continuous in the closed bounded interval $[a,b]$, and differentiable in the open interval (a,b), then there exists at least one interior point y in (a,b) such that

$$f'(y) = \frac{f(b) - f(a)}{b - a}.$$

"Proof":



2.1 Stationary points of quadratic functions

A second-order function:

$$\begin{aligned} f(\mathbf{x}) = & \underbrace{Q(\mathbf{x})}_{\text{quadr. form}} + \underbrace{L(\mathbf{x})}_{\text{linear form}} + \underbrace{c}_{\text{constant}} \\ = & \mathbf{x}' A\mathbf{x} + B\mathbf{x} + c. \end{aligned}$$

Then,

$$\frac{\partial f}{\partial \mathbf{x}} = \begin{bmatrix} \partial f/\partial x_1 \\ \vdots \\ \partial f/\partial x_n \end{bmatrix} = 2A\mathbf{x} + B'.$$

Example:

$$f(x_1, x_2) = x_1^2 - 4x_1x_2 + 3x_2^2 + 7x_1 - 8x_2 + 5$$

which can be written as

$$[x_1, x_2] \begin{bmatrix} 1 & -2 \\ -2 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [7, -8] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + 5.$$

$$\begin{aligned} \frac{\partial f}{\partial x_1} &= 2x_1 - 4x_2 + 7, \\ \frac{\partial f}{\partial x_2} &= 6x_2 - 4x_1 - 8, \\ &\text{so} \\ \frac{\partial f}{\partial \mathbf{x}} &= \begin{bmatrix} 2x_1 - 4x_2 + 7 \\ 6x_2 - 4x_1 - 8 \end{bmatrix}. \end{aligned}$$

$$2A\mathbf{x} = \begin{bmatrix} 2x_1 - 4x_2 \\ -4x_1 + 6x_2 \end{bmatrix}, \quad B' = \begin{bmatrix} 7 \\ -8 \end{bmatrix},$$

so

$$\frac{\partial f}{\partial \mathbf{x}} = 2A\mathbf{x} + B'.$$

2.2 Classification of stationary points

$$\frac{\partial f}{\mathbf{x}} = 0 \quad \text{i.e.} \quad 2A\mathbf{x} + B' = 0$$

which is the linear system

$$A\mathbf{x} = -\frac{1}{2}B'.$$

$$\begin{aligned} f \text{ convex} &\iff A \text{ positive semidefinite} \\ f \text{ concave} &\iff A \text{ negative semidefinite} \end{aligned}$$

If f convex:	All stationary points are global minimum.
If f concave:	All stationary points are global maximum.
If f indefinite:	All stationary points are saddle points.

3 Linear Regression

We are often interested in modeling relationships. For example, what is the relationship between returns and risk (what is risk?), what is the relationship between prices and quantities demanded, or does past returns inform us about future return dynamics? One of the workhorses in modeling relationships is linear regression.

3.1 Regression vs correlations

Why do we need regressions when we can easily compute correlations? Correlation between two variables x and y is the degree of linear association between them. This means that we treat the two variables in a completely symmetric way and we do not imply that the are any causal effect (i.e. that changes in x *causes* changes in y, or vice-versa.), but we rather say that there is evidence of a linear relationship, i.e. that they are related to an extent given by the correlation coefficient. When thinking in terms of regressions, the dependent variable (usally denoted by the vector $\boldsymbol{y}$) is treated as random and the independent variable(s) (usally denoted by the matrix X) is treated as fixed. This allows us to model relationships in a more meaningful way.

3.2 The Linear Regression Model

$$y_i = x_i'\beta + \varepsilon_i \quad i = 1, ..., N$$

x_i is a vector of observations on k independent variables. Usually, we include an intercept, so

$$x_i^t = [1, x_{1,i}, ..., x_{k,i}].$$

The conditional mean function, $r(x)$, then maps $\mathbb{R}^k \to \mathbb{R}$. Take the conditional expectation of both sides of the model:

$$\mathbb{E}[y|x] = \mathbb{E}[x'\beta + \varepsilon|x] = x'\beta + \mathbb{E}[\varepsilon|x].$$

If the conditional expectation of the error term, $\mathbb{E}[\varepsilon|x]$, is zero then we can interpret the linear term as the regression function, i.e. $r(x) = x'\beta$. If we include a constant term in the regressor matrix then this exogeneity condition will always be met.

We usually include another assumption, the homoskedasticity assumption:

$$\mathbb{V}[\varepsilon|x] = \sigma^2 < \infty.$$

Homoskedasticity refers to a constant variance (is this a reasonable assumption in financial time series?).

3.3 The Gauss-Markov Theorem

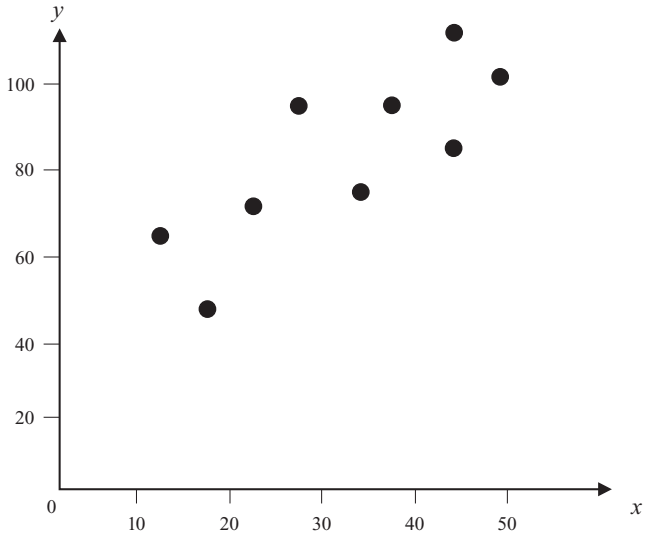
If the following assumptions about the residuals are true:

1. $\mathbb{E}[\varepsilon_i] = 0$,
2. $\mathbb{V}[\varepsilon_i] = \sigma^2 < \infty$,
3. $\text{Cov}(\varepsilon_i, \varepsilon_j) = 0, \forall i \neq j$,

then the OLS estimator is the Best Linear Unbiased Estimator (BLUE).

TI

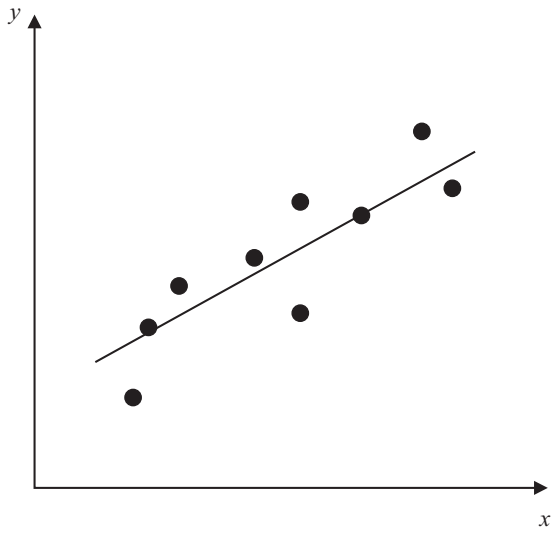
3.4 Estimation of β: Ordinary Least Squares - Univariate case



Let's say we are interested in estimating the following model:

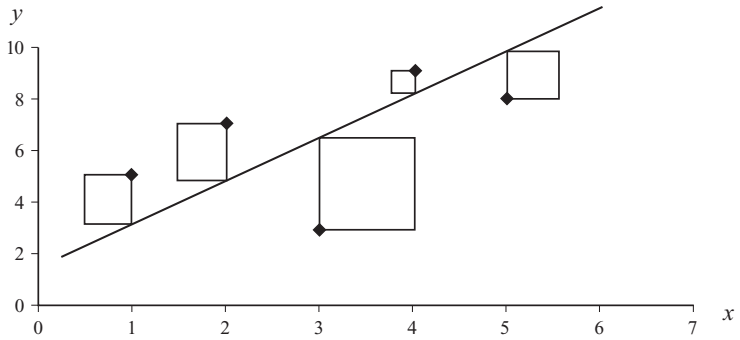
$$y_i = \alpha + x_i\beta + \varepsilon_i.$$

What this means is that we are looking for the estimates of α and β that best captures the linear relationship between the two variables:



The method of Ordinary Least Squares (OLS) entails taking each vertical distance from the point to the line, squaring it and then minimizing the sum of the squares:

TI



Write the fitted values of y_i as:

$$\hat{y}_i = \hat{\alpha} + \hat{\beta} x_i.$$

The residual, ε_i, is then defined as

$$\varepsilon_i = y_i - \hat{y}_i = y_i - \hat{\alpha} - \hat{\beta} x_i.$$

The problem of finding the $\hat{\alpha}$ and $\hat{\beta}$ that minimizes the sum of the squares can be written as

$$\underset{\alpha,\beta}{\arg\min} \sum_{i=i}^{N} \varepsilon_i^2 = (y_i - \hat{\alpha} - \hat{\beta} x_i)^2.$$

Let L denote the objective function, $\sum_{i=i}^{N} \varepsilon_i^2$. FOC:

$$\frac{\partial L}{d\alpha} = \sum_{i=1}^{N} -2(y_i - \hat{\alpha} - \hat{\beta} x_i) = 0 \tag{1}$$

$$\frac{\partial L}{d\beta} = \sum_{i=1}^{N} -2x_i(y_i - \hat{\alpha} - \hat{\beta} x_i) = 0 \tag{2}$$

From (1):

$$\begin{aligned}
\sum_{i=1}^{N} -2(y_i - \hat{\alpha} - \hat{\beta} x_i) &= 0 \\
\sum_{i=1}^{N} (y_i - \hat{\alpha} - \hat{\beta} x_i) &= 0 \\
N\overline{y} - N\hat{\alpha} - N\hat{\beta}\overline{x} &= 0 \\
\hat{\alpha} &= \overline{y} - \beta\overline{x}.
\end{aligned}$$

Now, insert this into (2):

$$\sum_{i=1}^{N} -2x_i(y_i - \hat{\alpha} - \hat{\beta}x_i) = 0$$
$$\sum_{i=1}^{N} -2x_i(y_i - (\overline{y} - \hat{\beta}\overline{x}) - \hat{\beta}x_i) = 0$$
$$\sum_{i=1}^{N} -2x_i(y_i - \overline{y} + \hat{\beta}\overline{x} - \hat{\beta}x_i) = 0$$
$$\sum_{i=1}^{N} x_i(y_i - \overline{y} + \hat{\beta}\overline{x} - \hat{\beta}x_i) = 0$$
$$\sum_{i=1}^{N} x_i y_i - \overline{y}x_i + \hat{\beta}\overline{x}x_i - \hat{\beta}x_i^2 = 0$$
$$\sum_{i=1}^{N} x_i y_i - \overline{y}\sum_{i=1}^{N} x_i + \hat{\beta}\overline{x}\sum_{i=1}^{N} x_i - \hat{\beta}\sum_{i=1}^{N} x_i^2 = 0$$
$$\hat{\beta} = \frac{\sum_{i=1}^{N} x_i y_i - N\overline{xy}}{\sum_{i=1}^{N} x_i^2 - N\overline{x}^2}$$
$$\hat{\beta} = \frac{\sum_{i=1}^{N}(x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{N}(x_i - \overline{x})^2}.$$

We see that that the slope coefficient is the equivalent to the sample covariance between x and y divided by the sample variance of x. We also see from the equation for $\hat{\alpha}$ that the regression line will go through the mean of the observations, i.e. that the point $(\overline{y}, \overline{x})$ lies on the regression line.

3.5 Estimation of β: Ordinary Least Squares - General case

We can write the model as

$$\mathbf{y} = \boldsymbol{X}\boldsymbol{\beta} + \varepsilon,$$

where **y** is a vector of the dependent variable, X is an (N×k) matrix of the independent variables and a constant column, and ε is a vector of the residuals. $\boldsymbol{\beta}$ is the vector of estimates that we are interested in estimating. The objective is still to minimize the sum of the squared residuals:

$$\arg\min_{\boldsymbol{\beta}} \varepsilon'\varepsilon = (\mathbf{y} - \boldsymbol{X}\boldsymbol{\beta})'(\mathbf{y} - \boldsymbol{X}\boldsymbol{\beta}) = \mathbf{y}'\mathbf{y} - 2\mathbf{y}'\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\beta}'\boldsymbol{X}'\boldsymbol{X}\boldsymbol{\beta}.$$

FOC:

$$\frac{\partial \varepsilon'\varepsilon}{\partial \boldsymbol{\beta}} = -2\boldsymbol{X}'\mathbf{y} + 2\boldsymbol{X}'\boldsymbol{X}\boldsymbol{\beta} = 0$$
$$\boldsymbol{X}'\boldsymbol{X}\boldsymbol{\beta} = \boldsymbol{X}'\mathbf{y}.$$

Now we need to isolate $\boldsymbol{\beta}$. We do this by using the inverse trick:

$$A^{-1}A = I.$$

$$\begin{aligned} \boldsymbol{X}'\boldsymbol{X}\boldsymbol{\beta} &= \boldsymbol{X}'\mathbf{y} \\ I\boldsymbol{\beta} &= (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\mathbf{y} \\ \boldsymbol{\beta} &= (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\mathbf{y}. \end{aligned}$$

3.6 Sampling Properties of OLS

Let $\boldsymbol{b}$ denote the least squares estimator. Then,

$$\begin{aligned}
\boldsymbol{b} &= (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\mathbf{y}, \\
\boldsymbol{b} &= (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\left(\boldsymbol{X}\boldsymbol{\beta}+\boldsymbol{\varepsilon}\right), \\
\boldsymbol{b} &= (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{X}\boldsymbol{\beta}+(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{\varepsilon}, \\
\boldsymbol{b} &= \boldsymbol{\beta}+\left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1}\boldsymbol{X}'\boldsymbol{\varepsilon}
\end{aligned}$$

Taking the conditional expectation gives us

$$\begin{aligned}
\mathbb{E}[\boldsymbol{b}|\boldsymbol{X}] &= \boldsymbol{\beta}+\mathbb{E}\left[\left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1}\boldsymbol{X}'\boldsymbol{\varepsilon}|\boldsymbol{X}\right], \\
&= \boldsymbol{\beta}+\left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1}\mathbb{E}\left[\boldsymbol{X}'\boldsymbol{\varepsilon}|\boldsymbol{X}\right], \\
&= \boldsymbol{\beta}+\left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1}\boldsymbol{X}'\mathbb{E}\left[\boldsymbol{\varepsilon}|\boldsymbol{X}\right], \\
&= \boldsymbol{\beta},
\end{aligned}$$

due to the exogeneity condition. Thus, the Least Squares estimator is unbiased. This also holds unconditionally since $\mathbb{E}[b] = \mathbb{E}_X[\mathbb{E}[b|X]] = \mathbb{E}_X[\beta] = \beta$ by the Law of Iterated Expectations. What is the uncertainty associated with the estimates?

$$\begin{aligned}
\mathbb{V}(\boldsymbol{b}) &= \mathbb{E}\left[(\boldsymbol{b}-\boldsymbol{\beta})^2|\boldsymbol{X}\right], \\
&= \mathbb{E}\left[(\boldsymbol{\beta}+\left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1}\boldsymbol{X}'\boldsymbol{\varepsilon}-\boldsymbol{\beta})^2|\boldsymbol{X}\right], \\
&= \mathbb{E}\left[\left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1}\boldsymbol{X}'\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}'\boldsymbol{X}\left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1}|\boldsymbol{X}\right], \\
&= \left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1}\boldsymbol{X}'\mathbb{E}\left[\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}'|\boldsymbol{X}\right]\boldsymbol{X}\left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1}, \\
&= \left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1}\boldsymbol{X}'\mathbb{V}(\boldsymbol{\varepsilon})\boldsymbol{X}\left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1}, \\
&= \left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1}\boldsymbol{X}'\Lambda\boldsymbol{X}\left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1},
\end{aligned}$$

where Λ is the covariance matrix of the residuals. If we assume that the errors are homoskedastic and not autocorrelated, then Λ is simply $I\sigma^2$, thus:

$$\mathbb{V}(\boldsymbol{b}) = \sigma^2\left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1}.$$

We are seeking a sample estimator $\boldsymbol{b}$ of the true population coefficient $\boldsymbol{\beta}$. Thus, if we are approaching the population, $\boldsymbol{b}$ should approach $\boldsymbol{\beta}$:

$$\lim_{n\to\infty}\mathbb{P}(|\boldsymbol{b}-\boldsymbol{\beta}|\geq\delta) = 0 \quad \text{for every } \delta > 0.$$

Equivalently:

$$\lim_{n\to\infty}\mathbb{V}(\boldsymbol{b}) = \sigma^2\lim_{n\to\infty}\left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1} = 0.$$

This property is called consistency. We can formally show this:

$$\begin{aligned}
b &= \beta+\left(X'X\right)^{-1}X'\varepsilon, \\
&= \beta+\left(\frac{X'X}{n}\right)^{-1}\frac{X'\varepsilon}{n}.
\end{aligned}$$

TI

Taking the probability limit on both sides gives us

$$\begin{aligned}\text{plim}(b) &= \beta + \text{plim}\left(\left(\frac{X'X}{n}\right)^{-1}\frac{X'\varepsilon}{n}\right),\\ &= \beta + \text{plim}\left(\left(\frac{X'X}{n}\right)^{-1}\right)\text{plim}\left(\frac{X'\varepsilon}{n}\right).\end{aligned}$$

First, we know that

$$\text{plim}\left(\left(\frac{X'X}{n}\right)^{-1}\right) = \Omega,$$

which is a positive definite matrix. Next,

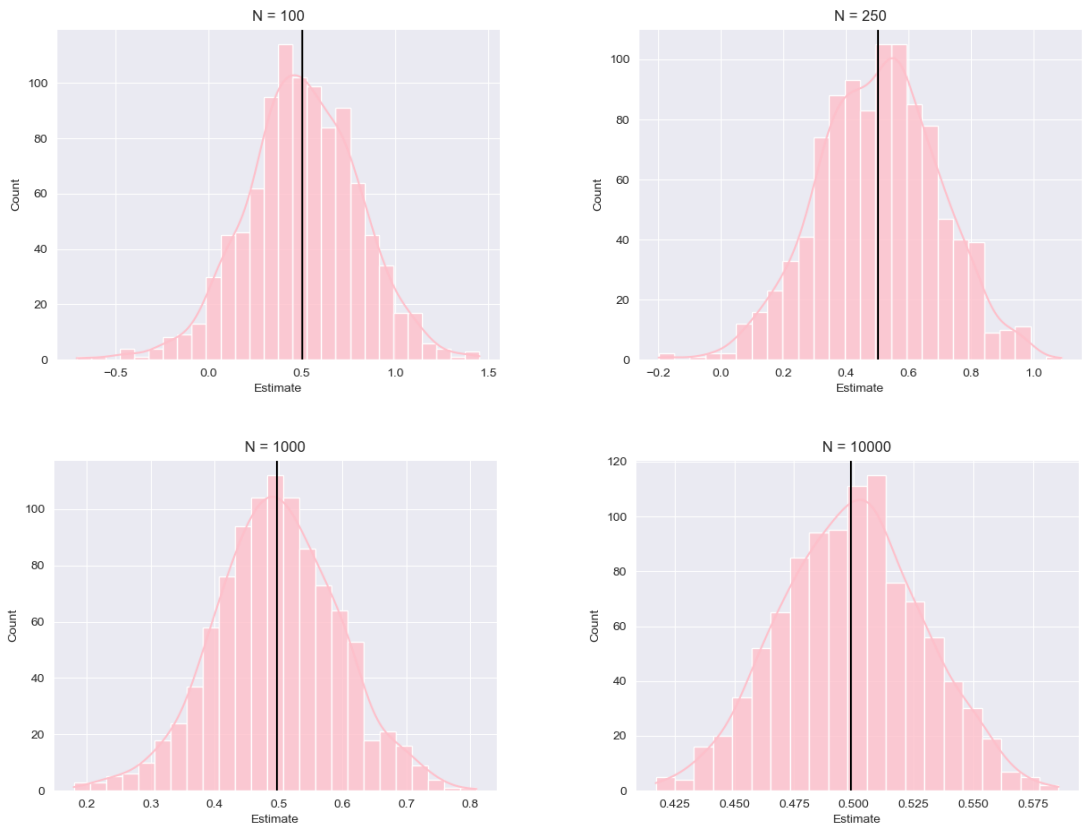
$$\text{plim}\left(\frac{X'\varepsilon}{n}\right) = 0,$$

which follows from the fact that $\mathbb{E}[\varepsilon|X] = 0$ which implies $\mathbb{E}[\varepsilon x] = 0$. This is because $\frac{X'\varepsilon}{n} = n^{-1}\sum_{i=1}^{n} x_i\varepsilon_i$. This leads us to the desired result:

$$\begin{aligned}\text{plim}(b) &= \beta + \text{plim}\left(\left(\frac{X'X}{n}\right)^{-1}\right)\text{plim}\left(\frac{X'\varepsilon}{n}\right),\\ &= \beta + \Omega \times 0 = \beta.\end{aligned}$$

$\Rightarrow b$ is consistent!

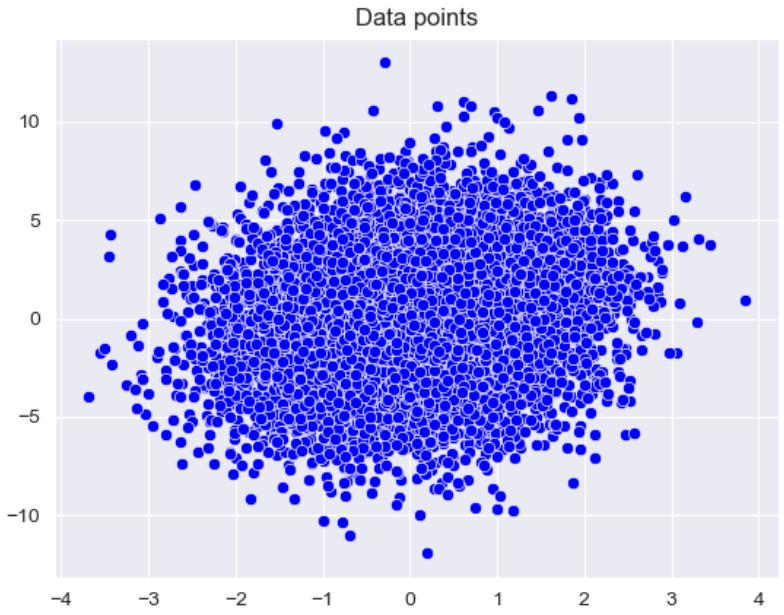
3.7 Estimator uncertainty



TI

The plots above might appear to be very similar at first sight - they are all sort of symmetric and are centered around the same mean value (0.5). However, if we take a look at the spread in each plot, we see that the upper left plot (N=100) includes a range of values between -0.7 and 1.5, but the lower right plot has is spread over over the range 0.425-0.575. These are all estimates from a simulated series of data. The data can be seen in figure 18.

Figure 2: Plot of the data



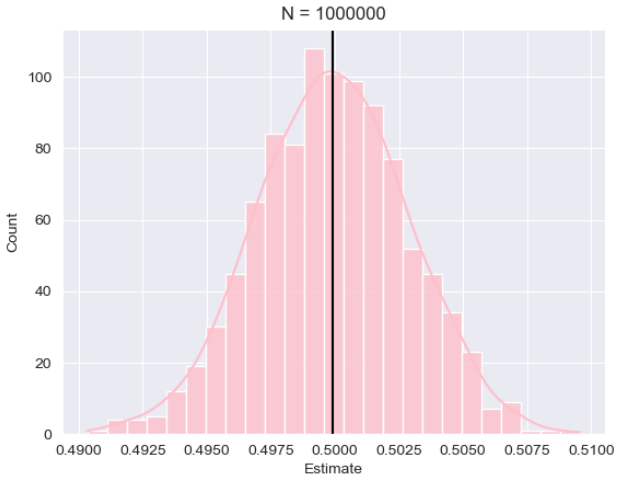
In this example, N refers to the sample length. As N increase we see that also the spread decrease. Put differently, as N increase, the uncertainty about the estimate we have computed decrease. The data generating process in this case was

$$y_i = 0.2 + 0.5 \times x_i + 3 \times u_i,$$

where $x_i \sim \mathcal{N}(0,1), u_i \sim \mathcal{N}(0,1)$. So therefore, the estimates seems to be unbiased and consistent since (i) the estimates are centered around the true value (0.5) (ii) the variance of the estimate appears to go towards zero as the sample size increase. This begs the question: how can we evaluate the precision of our estimate?

TI

Figure 3: Estimates when the sample length is reeaallly long.




```
import numpy                as np
import pandas               as pd
import seaborn              as sns
import matplotlib.pyplot    as plt
import statsmodels.api      as sm

N1 = 100
N2 = 250
N3 = 1000
N4 = 10000
N5 = 1000000

N = N4

alpha = 0.2
beta  = 0.5

epsilon = np.random.randn(N)

x = np.random.randn(N)

y = alpha + x*beta + epsilon

X = sm.add_constant(x)

# OLS
XX = np.matmul(X.T,X)

beta_hat = np.matmul(np.linalg.inv(XX),np.matmul(X.T,y))
var_beta = np.var(epsilon)*np.linalg.inv(XX)

# Sampling properties of beta_hat
b = [0]*1000
```

```
a = [0]*1000

for ii in range(1000):
    epsilon = np.random.randn(N)

    x = np.random.randn(N)

    y = alpha + x*beta + epsilon*3


    X = sm.add_constant(x)

    # OLS
    XX = np.matmul(X.T,X)

    a[ii] = np.matmul(np.linalg.inv(XX),np.matmul(X.T,y))[0]
    b[ii] = np.matmul(np.linalg.inv(XX),np.matmul(X.T,y))[1]
    ab = a+b

t1           = ['a']*1000
t2           = ['b']*1000
t            = t1+t2
data         = pd.DataFrame({'Estimate':ab})
data['type'] = t

# Plot histogram
plt.figure()
ax=sns.histplot(data[data.type=='b'].Estimate,kde=True,color='pink',
alpha=0.8)
ax.set(title   = 'N = '+str(N))
plt.axvline(data[data.type=='b'].Estimate.mean(), color='black')
ax.get_figure().savefig('plot'+str(N) + '.png')


# Scatter plot of data points
plt.figure()
ax=sns.scatterplot(x     = x,
                   y     = y,
                   color = "blue",
                  )
ax.set(title   = "Data points")
ax.get_figure().savefig('plot_data_'+str(N) + '.png')
```

3.8 Hypothesis testing - The test of significance approach

When we have an estimate, $\boldsymbol{b}$, when can easily compute the standard errors of this estimate, SE($\boldsymbol{b}$):

$$SE(\boldsymbol{b}) = \sqrt{\mathbb{V}(\boldsymbol{b})} = \sqrt{\sigma^2(\boldsymbol{X}'\boldsymbol{X})^{-1}}.$$

The diagonal elements of this matrix corresponds to each estimate. Element (1,1) is the standard error of the intercept, element (2,2) is the standard error of the first slope estimate, etc. When doing a

test of significance, we usually form some type of hypothesis. A typical hypothesis formulation could be that, under the null hypothesis, the true population parameter is zero. An alternative hypothesis could be that the true population parameter is different from zero:

$$\begin{array}{lll} H_0 & \beta = 0 & \text{(Null hypothesis)} \\ H_1 & \beta \neq 0 & \text{(Alternative hypothesis)} \end{array}$$

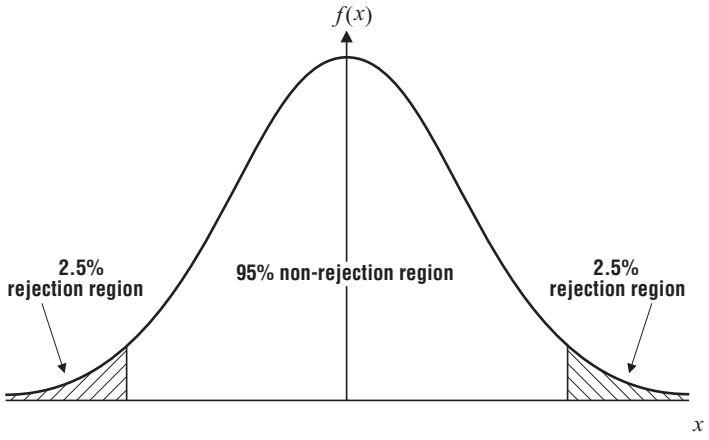
Then, we need to provide the distribution of the test statistic under the null. It is important to keep in mind that the test statistic is indeed a random variable, so we need it to have a well defined distribution so that we can define the rejection region. In our case, our test statistic will follow a t-ditribution with (n-k) degrees for freedom. The test statistic can be computed the following way:

$$\text{test statistic} = \frac{\hat{\beta} - \beta^*}{\text{SE}(\hat{\beta})} \sim t_{\alpha, n-k},$$

where β^* is the parameter value under the null.

We need to define the significance level, α, of our test. What is a "correct" significance level to use? The common value is 5%, however it is debatable if that is always appropriate. For a two-sided test, the rejection regions can be illustrated in the following figure 17.

Figure 4: The rejection regions for a two-sided test with a significance level of 5%.



Look at how the test statistic is computed, it is a ratio of the distance between the null and the computed value divided by the standard error. Since we know the distribution of the test statistic, we can make inferences about, given our null hypothesis, how "likely" it is to observe the estimate that we have computed. If the probability is less that the significance level we have chosen, we reject our null hypothesis. Be careful: this does not mean that we "accept" our alternative hypothesis. Similarly, if we fail to reject the null it means just that - we never accept the null hypothesis. Sometimes it is useful to think about the assumptions that we make for the validity of the test (e.g. Gauss-Markov Assumption + normality of errors).

4 Constrained Optimization

In the previous section we were focusing on optimizing functions over an unconstrained domain. However, usually we have constraints. In economics, an obvious constraint is a budget constraint; the agent might only have ten dollars to spend on either coconuts or bananas which has prices \$1 and \$1.50 respectively. Now the problem becomes much more interesting since the questions we can answer by imposing these restrictions becomes much closer to the ones we are trying to solve in real-life. Another (and perhaps more relevant) examples for people in quantitative finance is that we sometimes want to find optimal portfolios. A common restriction is that all the portfolio weights must sum to one. Perhaps we do not want to allow shorting? Well, then we add another constraint where all weights must be weakly positive.

4.1 The Lagrange Multiplier Method

Assume that we have a consumer that chooses how much of her available income m she will spend on a good x which costs p, and how much she will spend on other goods y. Her budget constraint is $xp + y = m$. Her preference for consuming x and y is given by her utility function $u(x, y)$. Mathematically we can now represent her problem as

$$\max u(x, y) \quad \text{subject to} \quad px + y = m.$$

This is a **constrained maximization problem**. In general, we write

$$\max(\min) \; f(x, y) \quad \text{subject to} \quad g(x, y) = c.$$

Definition 4.1: Lagrange Multiplier Method

To find the only possible solutions of the problem

$$\text{maximize (minimize)} \quad f(x, y) \quad \text{subject to} \quad g(x, y) = c$$

proceed as follows:

1. Write down the Lagrangian

   $$\mathscr{L}(x, y) = f(x, y) - \lambda(g(x, y) - c)$$

   where is a constant.

2. Differentiate $\mathscr{L}$ w.r.t. x and y and equate the partial derivatives to 0.

3. The two equations in (2.), together with the constraint, yield the following three equations:

   $$\begin{aligned} \mathscr{L}_1'(x, y) &= f_1'(x, y) - \lambda g_1'(x, y) = 0 \\ \mathscr{L}_2'(x, y) &= f_2'(x, y) - \lambda g_2'(x, y) = 0 \\ g(x, y) &= c. \end{aligned}$$

4. Solve these three equations simultaneously for the three unknowns x, y, and . These triplets (x, y, λ) are the solution candidates, at least one of which solves the problem (if it has a solution).

Example: A consumer has the utlity function $U(x,y) = xy$ and faces the budget constraint $2x+y = 100$. Find the only solution candidate to the consumer demand problem

$$\max xy \text{ subject to } 2x + y = 100.$$

The Lagrangian becomes

$$\mathscr{L}(x,y) = xy - \lambda(2x + y - 100).$$

FOC:

$$\begin{aligned}\mathscr{L}_1'(x,y) &= y - 2\lambda = 0,\\ \mathscr{L}_2'(x,y) &= x - \lambda = 0,\\ 2x + y &= 100.\end{aligned}$$

From the first two solutions we have that $y = 2\lambda$ and $x = \lambda$. So $y = 2x$. Inserting this into the constraint yields $2x + 2x = 100$. So $x = 25$ and $y = 50$, implying that $\lambda = x = 25$.

4.2 Intepreting the Lagrange Multiplier

Consider again the problem

$$\text{maximize (minimize)} \quad f(x,y) \quad \text{subject to} \quad g(x,y) = c.$$

Suppose x^* and y^* are the values of x and y that solve this problem. In general, x^* and x^* depend on c. We assume that $x^* = x^*(c)$ and $y^* = y^*(c)$ are differentiable functions of c. The associated value of $f(x,y)$ is then also a function of c, with

$$f^*(c) = f(x^*(c), y^*(c)).$$

Here $f^*(c)$ is called the (optimal) value function for the problem. The associated value of the Lagrange multiplier also depends on c, in general. Provided that a certain regularity conditions are satisfied, we have the result that

$$\frac{df^*(c)}{dc} = \lambda(c).$$

We therefore have the following result:

Definition 4.2: Lagrange Multiplier

The Lagrange multiplier $\lambda = \lambda(c)$ is the rate at which the optimal value of the objective function changes with respect to changes in the contraint constant c. In particular, if dc is a small change in c, then

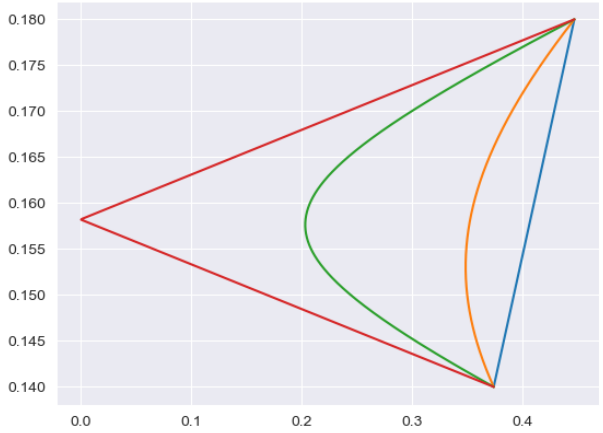
$$f^*(c + dc) - f^*(c) \approx \lambda(c)dc.$$

4.3 Application: Portfolio Theory

Figure 5: God himself

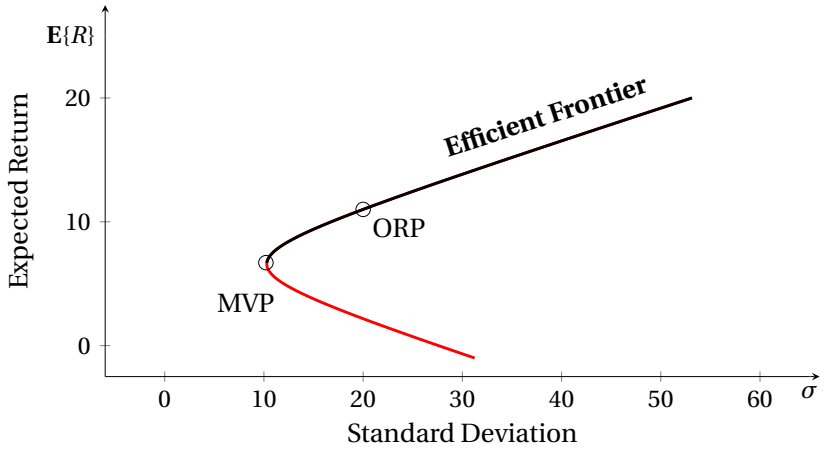
When we are investing we are faced with the problem of deciding how to allocate our limited wealth into assets. In his seminal 1952 paper, Nobel Laureate Harry Markowitz introduced modern portfolio theory. He argued that we should not pay much attention to each assets return-risk characteristics (expected return and standard deviation), but rather how inclusion of assets affects the overall riskiness and return of the portfolio.

Figure 6: The effect of different combination of two assets (correlations: blue =1, orange = 0.5, green = -0.5, and red = -1)



```
w_1   = np.linspace(0,1,500)
w_2   = 1-w_1
var_1 = 0.2
var_2 = 0.14
r_1   = .18
r_2   = .14
rhos  = [1,0.5,-0.5,-1]
def var(weight1,weight2,var1,var2,corr):
    std1 = var1**(.5)
    std2 = var2**(.5)
    return(weight1**2*var1 + weight2**2*var2 + \
                  2*corr*std1*std2*weight2*weight1)
for rho in rhos:
    p_ret = [0]*len(w_1)
    p_var = [0]*len(w_1)
    for ind in range(len(w_1)):
        p_ret[ind] = w_1[ind]*r_1+w_2[ind]*r_2
        p_var[ind] = var(w_1[ind],w_2[ind],var_1,var_2,rho)**.5
    plt.plot(p_var,p_ret, '-')
```

First we will tackle the problem of finding the of the optimal portfolios. We will be interested in two portfolios: (i) The minimum variance portfolio (ii) the optimal risky portfolio. Graphically:



The question is then, how can we find these two portfolios?

4.4 Minimum Variance Portfolio

The minimum variance portfolio is the portfolio that minimize the portfolio with the least variance among all feasible portfolios. Mathematically, we are searching for the weights, ω_i, that minimize the variance of the portfolio p. Since we are constructing a portfolio, we want the weights of this portfolio to sum to one. This gives rise to the following constrained optimization problem:

$$\underset{\omega_i}{\arg\min}\ \sigma_p^2 \quad \text{subject to } \sum \omega_i = 1.$$

Again, remember how we defined portfolio variance:

$$\sigma^2 = \boldsymbol{\omega}'\Sigma\boldsymbol{\omega},$$

where $\boldsymbol{\omega}$ is a vector of weights and Σ is a covariance matrix. We see that this is a simple optimization problem with a quadratic form:

$$\underset{\omega}{\arg\min}\ \ \boldsymbol{\omega}'\Sigma\boldsymbol{\omega} \quad \text{subject to } \boldsymbol{\omega}'\mathbf{1} = 1.$$

Example: Consider the case with two assets, 1 and 2. We want to find the weights, ω_1 and ω_2, that minimizes the portfolio variance. The problem becomes:

$$\underset{\omega_1,\omega_2}{\arg\min}\ \ \omega_1^2 Var(s_1) + \omega_2^2 Var(s_2) + 2\omega_1\omega_2 cov(s_1,s_2) \quad \text{subject to } \omega_1 + \omega_2 = 1.$$

From the constraint we get that $\omega_2 = 1-\omega_1$. We rewrite the problem as

$$\underset{\omega_1}{\arg\min}\ \ \omega_1^2 Var(s_1) + (1-\omega_1)^2 Var(s_2) + 2\omega_1(1-\omega_1)cov(s_1,s_2).$$

FOC:

$$2\omega_1 Var(s_1) + 2Var(s_2) - 2\omega_1 Var(s_2) + 2cov(s_1,s_2) - 4\omega_1 cov(s_1,s_2) = 0.$$

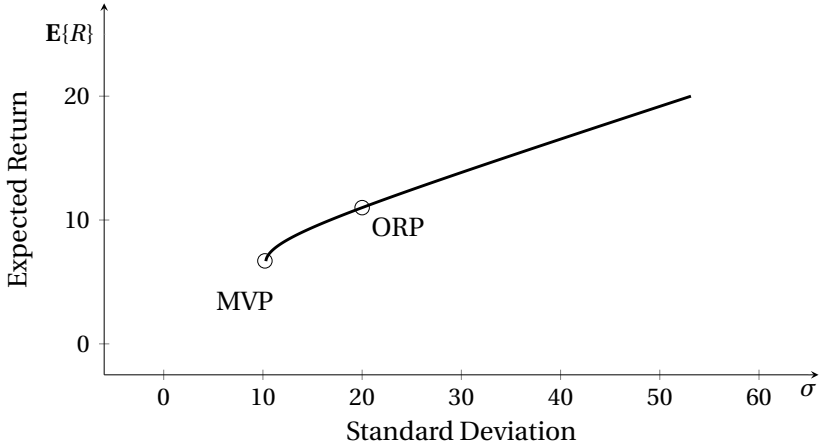
Rearranging yields

$$\omega_1^* = \frac{Var(s_2) - cov(s_1,s_2)}{Var(s_1) + Var(s_2) - 2cov(s_1,s_2)}.$$

and consequently, $\omega_2^* = 1-\omega_1^*$. How do we know that this actually solves the minimization problem? Since this is a quadratic function, it's sufficient to check the definiteness of the covariance matrix, Σ. Since any covariance matrix is positive semi-definite, we know that the function if convex and that any optimum is a global minimum.

4.5 Optimal Risky Portfolio

We are now interested in the Optimal Risky Portfolio, which is defined as the portfolio that maximizes the expected risk-return tradeoff.



If we look at the efficient frontier, we see that it is constructed by the portfolios that, for a given exptected return, minimize the risk of the portfolio. Clearly, the minimum variance portfolio will be one of these portfolios. However, among these portfolios there exists a portfolio that gives us a better deal in terms of return per unit of risk than any of the other portfolios. We call this the Optimal Risky Portfolio.

For any portfolio we can define the Sharpe Ratio, named after Nobel Laureate William F. Sharpe as the ratio of the excess return of the portfolio divided by its standard deviation:

$$S_p = \frac{\mathbb{E}[r_p] - r^f}{\sigma_p}.$$

The Optimal Risky Portfolio is then the portfolio that maximize this ratio. The problem of finding this portfolio becomes

$$\arg\max_{\omega} \; S_p \quad \text{subject to} \quad \omega'\mathbf{1} = 1,$$

or

$$\arg\max_{\omega} \; \omega' \boldsymbol{R} \left(\sqrt{\omega' \Sigma \omega}\right)^{-1} \quad \text{subject to} \quad \omega'\mathbf{1} = 1,$$

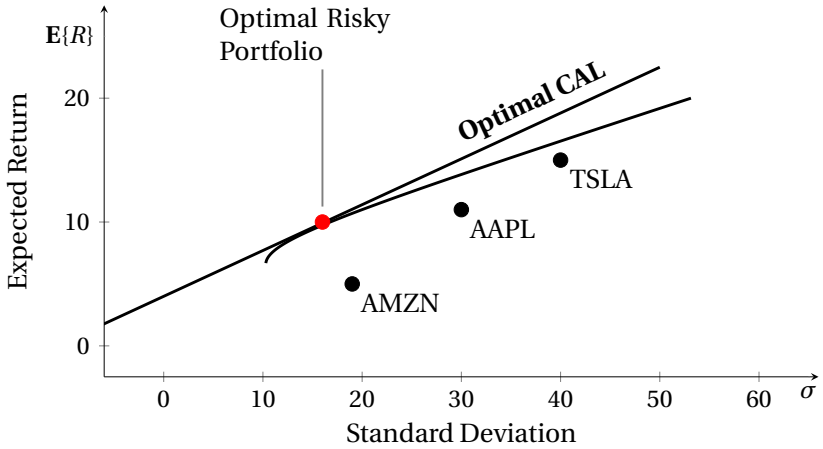
where $\boldsymbol{R}$ is a vector of excess returns for each asset.

Example: For the 2-asset problem, the optimal weights can be computed as

$$\omega_1^* = \frac{\left(\mathbb{E}[s_1] - r^f\right) Var(s_2) - \left(\mathbb{E}[s_2] - r^f\right) cov(s_1, s_2)}{\left(\mathbb{E}[s_1] - r^f\right) Var(s_2) + \left(\mathbb{E}[s_2] - r^f\right) Var(s_1) - \left(\mathbb{E}[s_1] - r^f + \mathbb{E}[s_2] - r^f\right) cov(s_1, s_2)},$$
$$\omega_2^* = 1 - \omega_1^*.$$

4.6 Asset allocation with risky assets and T-bill

Now that we have established the optimal portfolio, we can talk about optimal capital allocation. In mean-variance analysis, every investor actually agrees on what this portfolio is. Why? Because the mean-variance investor only cares about the first two moments, the expected return and the variance. If we are all in the same investment universe, then mechanically we should end up with the same optimal portfolio. If we introduce a risk-free asset (usually a T-bill), we can construct a line that is tangent to the efficient frontier.

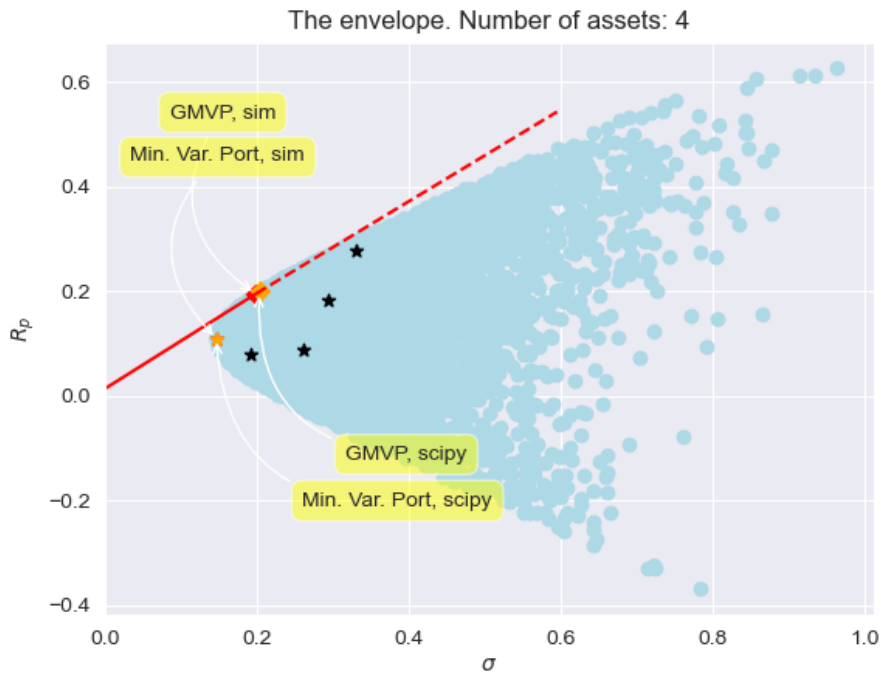


In fact, the exercise of finding the optimal risky portfolio can be viewed as the problem of finding the capital allocation line with the steepest gradient. Moreover, now there exists only one efficient risky portfolio. As an example, it does not any longer make sense to say that the minimum variance portfolio is efficient. Why? Because we can do better. By investing in T-bills and the optimal portfolio, we can generate a return that is higher than the minimum variance portfolio, but with the same amount of risk. Therefore, the problem of an investor can be decomposed into two steps: (i) finding the optimal risky portfolio (ii) decide on the allocation between the risk-free asset and this optimal portfolio. This is called the two-fund separation theorem. Notice that the capital allocation line extends further than the optimal portfolio. This is because can **lend** at the risk-free rate (**short T-bills**) and invest the proceeds in the optimal portfolio. The weight in the risk-free asset will then become negative and the weight in the optimal portfolio will be greater than one, but the sum of the two will still be one.

4.7 Numerical approach to find the portfolios

How can we solve the portfolio construction problem in Python? Two ways are by simulation and by a numerical optimizer.

Figure 7: Using numerical approaches to find the portfolios



We see from the figure above that the two approaches yields approximately the same solution. The simulation approach works by drawing random weights and then constructing portfolios based on these. The numerical optimizer takes an objective function and then optimize.

```
import requests
import sys
import pandas              as pd
import pandas_datareader   as web
import numpy               as np
import matplotlib.pyplot   as plt
import datetime            as dt
```

```
import yfinance          as yf

from bs4             import BeautifulSoup
from scipy.optimize import minimize


### Define functions ###
def objective_gmvp(w, ret ,vCov, rf):
    '''
    Objective function for global mean variance portfolio
    Inputs:
     - w    : vector of weights
     - ret  : vector of returns
     - vCov : covariance matrix
     - rf   : risk-free rate

    Output:
        - -1s_p : The Sharpe ratio times negative one.
    '''
    var = np.matmul(np.matmul(w, vCov) ,np.transpose(w))
    std  = var**0.5
    s_p  = (np.matmul(w, ret)-rf)/std
    return -1*s_p


def objective_var(w,vCov):
    '''
    Objective function for minimum variance portfolio.
    Inputs:
     - w   : vector of weights
     - ret : vector of returns

    Output:
    - var_p : The portfolio variance.
    '''
    var_p = np.matmul(np.matmul(w, vCov) ,np.transpose(w))
    return var_p

def constraint(w):
    '''
    Constraint. We want our weights to sum to one.
    Input:
     - w   : vector of weights

    Output:
     - The sum of the weight vector minus one.
    '''
    return sum(w) - 1


def get_ticks():
    # Get all SP500 tickers and industries
    URL = "https://en.wikipedia.org/wiki/List_of_S%26P_500_companies"

    res       = requests.get(URL).text
    soup      = BeautifulSoup(res,'lxml')
    comp_list = []
    ind_list  = []
    for items in soup.find('table', class_='wikitable').find_all('tr')[1::1]:
        row = items.find_all(['th','td'])
        try:
            comp_list.append(row[0].a.text)
            ind_list.append(row[3].text)
        except: continue
    ind_list = [ind.split('\n')[0] for ind in ind_list ]
    return comp_list, ind_list

################################################################################

tickers      = ['AAPL','MSFT','GOOGL','GLD']
data         = pd.DataFrame()
start        = dt.datetime(2005, 1, 1)
end          = dt.datetime(2018,1,1)
data         = yf.download(tickers, start=start, end=end).Close
data.columns = tickers
ret_daily    = np.log(data / data.shift(1))
ret_mean_an  = ret_daily.mean()*252
vcov         = ret_daily.cov()*252
print('.'*100)
print('\n Summary statistics \n')
print('.'*100)
print(' \n Annualized mean returns: \n')
print(ret_mean_an)
print('\n Covariance matrix of returns: \n')
print(vcov)
print('.'*100)


port_std = []
port_ret = []
simLen   = 10000
print('\nProgress:')
print('.'*100 + '\n')
for ii in range(1,simLen+1):
    weights   = np.random.normal(0,1,len(tickers))
    weights  /= weights.sum()
    if np.any(weights>=3) or np.any(weights<=-3): continue
    port_var  = np.matmul(np.matmul(weights,vcov),np.transpose(weights))
    port_std  = np.append(port_std,port_var**0.5)
    port_ret  = np.append(port_ret,np.matmul(weights,ret_mean_an))
    if (ii % 100==0 and ii != simLen+1):
        b=('Finished with iteration ' + str(ii) + ' of ' + str(len(range(1,simLen+1))))
```

```
        sys.stdout.write('\r'+b)
    if (ii == simLen): sys.stdout.write('\r'+'-'*30+' Done! '+'-'*30+'\n')

## Identify mean-variance portfolio through simulation ##
risk_free    = web.get_data_fred('TB3MS',start=end,end=end)/100
s_p          = (port_ret-risk_free['TB3MS'][0])/port_std
s_p_m_ind    = np.argmax(s_p)
min_var_ind = np.argmin(port_std)

### Identify GMVP by constrained optimization ###
rf = risk_free.values[0].tolist()
rf = rf[0]

# Initial guesses
x0   = np.ones(len(weights))
x0  /= x0.sum()

# Run optimization routine, find GMVP
b    = (-3,3)
bnds = (b,)*len(weights)
con  = {'type': 'eq', 'fun': constraint}
gmvp = minimize(objective_gmvp,x0,args=(ret_mean_an,vcov,rf),constraints=con,
                bounds=bnds,options={'disp': True})
var_gmvp = np.matmul(np.matmul(gmvp.x,vcov),np.transpose(gmvp.x))
ret_gmvp = np.matmul(gmvp.x,ret_mean_an)
std_gmvp = var_gmvp**0.5
s_p_gmvp = (ret_gmvp-rf)/std_gmvp
print('.'*100)
print('\nGlobal Mean Variance portfolio:\n')
print('Return = ' + str(ret_gmvp) + '\n')
print('Standard deviation = ' +  str(std_gmvp) + '\n')
print('Sharpe ratio = ' + str(s_p_gmvp) + '\n')
print('.'*100)


# Run optimization routine, find MVP
minvar = minimize(objective_var,x0,args=(vcov),constraints=con,
                  options={'disp': True})
ret_minvar = np.matmul(minvar.x,ret_mean_an)
std_minvar = minvar.fun**0.5
print('.'*100)
print('\nMinimum variance portfolio:\n')
print('Return = ' + str(ret_minvar) + '\n')
print('Standard deviation = ' + str(std_minvar) + '\n')
print('.'*100)

fig, ax = plt.subplots()
ax.scatter(port_std, port_ret, c='lightblue')
ax.scatter(port_std[s_p_m_ind], port_ret[s_p_m_ind], c='red',marker='D')
ax.scatter(port_std[min_var_ind], port_ret[min_var_ind], c='red',marker='*')
ax.scatter(std_gmvp, ret_gmvp, c='orange',marker='D')
ax.scatter(std_minvar, ret_minvar, c='orange',marker='*')
ax.scatter(np.diag(vcov**0.5), ret_mean_an, c='black',marker='*')
ax.plot([0,port_std[s_p_m_ind]], [risk_free['TB3MS'][0],port_ret[s_p_m_ind]],c='r')
ax.plot([port_std[s_p_m_ind],port_std[s_p_m_ind]*3], [port_ret[s_p_m_ind],\
        (port_ret[s_p_m_ind]-risk_free['TB3MS'][0])*3],c='r',linestyle='--')
plt.xlim(left=0)
ii=1
for port, x, y in zip(['GMVP, scipy','GMVP, sim','Min. Var. Port, scipy','Min. Var. Port, sim'],
                      [std_gmvp,port_std[s_p_m_ind],std_minvar,port_std[min_var_ind]],
                      [ret_gmvp,port_ret[s_p_m_ind],ret_minvar, port_ret[min_var_ind]]):
    plt.annotate(
        port, xy=(x, y), xytext=(-40*((-1)**(ii)), 80*((-1)**(ii))),
        textcoords='offset points', ha='left', va='bottom',
        bbox=dict(boxstyle='round,pad=0.5', fc='yellow', alpha=0.5),
        arrowprops=dict(arrowstyle = '->', connectionstyle='arc3,rad=' + str((-1)**(ii)*0.5)))
    ii = ii+1
plt.title("The envelope. Number of assets: " + str(len(tickers)))
plt.xlabel(r'$\sigma$')
plt.ylabel(r'$R_p$')
plt.show()
#plt.savefig('plot1.pdf')


### What if we add more assets? ###

sp_tickers,_ = get_ticks()

## Fetch 20 random tickers ##
magic_indicies = np.random.randint(0,500,20)
ext_tickers    = [sp_tickers[ind] for ind in magic_indicies]
data_new       = yf.download(ext_tickers, start=start, end=end).Close

ret_daily_new    = np.log(data_new / data_new.shift(1))
ret_mean_an_new  = ret_daily_new.mean()*252
vcov_new         = ret_daily_new.cov()*252
print('.'*100)
print('\n Summary statistics \n')
print('.'*100)
print('\n Annualized mean returns: \n')
print(ret_mean_an_new)
print('\n Covariance matrix of returns: \n')
print(vcov_new)
print('.'*100)



port_std_new = []
port_ret_new = []
simLen    = 50000
print('\nProgress:')
print('.'*100 + '\n')
```

```
for ii in range(1,simLen+1):
    weights    = np.random.normal(0,1,len(data_new.columns))
    weights   /= weights.sum()
    if np.any(weights>=3) or np.any(weights<=-3): continue
    port_var_new = np.matmul(np.matmul(weights,vcov_new),np.transpose(weights))
    port_std_new = np.append(port_std_new,port_var_new**0.5)
    port_ret_new = np.append(port_ret_new,np.matmul(weights,ret_mean_an_new))
    if (ii % 100==0 and ii != simLen):
        b=('Finished with iteration ' + str(ii) + ' of ' + str(len(range(1,simLen+1))))
        sys.stdout.write('\r'+b)
    if (ii == simLen): sys.stdout.write('\r'+'-'*30+' Done! '+'-'*30+'\n')

## Identify mean-variance portfolio through simulation ##
risk_free         = web.get_data_fred('TB3MS',start=end,end=end)/100
s_p_new           = (port_ret_new-risk_free['TB3MS'][0])/port_std_new
s_p_m_ind_new     = np.argmax(s_p_new)
min_var_ind_new = np.argmin(port_std_new)

# Initial guesses
x0    = np.ones(len(data_new.columns))
x0   /= x0.sum()

# Run optimization routine, find GMVP
b         = (-3,3)
bnds      = (b,)*len(data_new.columns)
con       = {'type': 'eq', 'fun': constraint}
gmvp_new = minimize(objective_gmvp,x0,args=(ret_mean_an_new,vcov_new,rf),constraints=con,
                    bounds=bnds,options={'disp': True})
ret_gmvp_new = np.matmul(gmvp_new.x,ret_mean_an_new)
std_gmvp_new = objective_var(gmvp_new.x,vcov_new)**0.5

# Run optimization routine, find MVP
minvar_new = minimize(objective_var,x0,args=(vcov_new),constraints=con,
                      options={'disp': True})
ret_minvar_new = np.matmul(minvar_new.x,ret_mean_an_new)
std_minvar_new = minvar_new.fun**0.5


fig, ax = plt.subplots()
ax.scatter(port_std_new, port_ret_new, c='lightblue')
ax.scatter(port_std_new[s_p_m_ind_new], port_ret_new[s_p_m_ind_new], c='red',marker='D')
ax.scatter(port_std_new[min_var_ind_new], port_ret_new[min_var_ind_new], c='red',marker='*')
ax.scatter(std_gmvp_new, ret_gmvp_new, c='orange',marker='D')
ax.scatter(std_minvar_new, ret_minvar_new, c='orange',marker='*')
ax.plot([0,port_std_new[s_p_m_ind_new]], [risk_free['TB3MS'][0],port_ret_new[s_p_m_ind_new]],c='r')
ax.scatter(np.diag(vcov_new**0.5), ret_mean_an_new, c='black',marker='*')
ii=1
for port, x, y in zip(['GMVP, scipy','GMVP, sim','Min. Var. Port, scipy','Min. Var. Port, sim'],
                      [std_gmvp_new,port_std_new[s_p_m_ind_new],std_minvar_new,port_std_new[min_var_ind_new]],
                      [ret_gmvp_new,port_ret_new[s_p_m_ind_new],ret_minvar_new, port_ret_new[min_var_ind_new]]):
    plt.annotate(
        port, xy=(x, y), xytext=(-40*((-1)**(ii)), 80*((-1)**(ii))),
        textcoords='offset points', ha='left', va='bottom',
        bbox=dict(boxstyle='round,pad=0.5', fc='yellow', alpha=0.5),
        arrowprops=dict(arrowstyle = '->', connectionstyle='arc3,rad=' + str((-1)**(ii)*0.5)))
    ii = ii+1
ax.plot([port_std_new[s_p_m_ind_new],port_std_new[s_p_m_ind_new]*3],\
        [port_ret_new[s_p_m_ind_new],(port_ret_new[s_p_m_ind_new]-risk_free['TB3MS'][0])*3],c='r',linestyle='--')
ax.plot([port_std[s_p_m_ind],port_std[s_p_m_ind]*3], [port_ret[s_p_m_ind],\
         (port_ret[s_p_m_ind]-risk_free['TB3MS'][0])*3],c='g',linestyle='--')
plt.xlim(left=0,right=2)
plt.title("The envelope. Number of assets: " + str(len(data_new.columns)))
plt.xlabel(r'$\sigma$')
plt.ylabel(r'$R_p$')
plt.show()
#plt.savefig('plot2.pdf')


### Figure with both envelopes, for comparison
fig, ax = plt.subplots()
ax.scatter(port_std, port_ret, c='red')
ax.scatter(port_std_new, port_ret_new, c='lightblue' ,alpha=0.6)
ax.legend(['Number of assets: '+ str(len(tickers)),'Number of assets: '+ str(len(data_new.columns))])
plt.xlim(left=0,right=1)
plt.title('The envelope.')
plt.xlabel(r'$\sigma$')
plt.ylabel(r'$R_p$')
plt.show()
#plt.savefig('comparison.pdf')
```