Selection of optimal power plant generation mixes

Dr. Barbara Glensk

FCN | Future Energy Consumer Needs and Behavior

E.ON Energy Research Center

RWTH AACHEN UNIVERSITY

Presentation outline

1. Economic viability of energy investments
2. Definition of portfolio selection criterion for power generation assets
3. Mean-variance portfolio selection model for power generation assets
4. Calculation procedure
5. Example: Python code for simulation of present values
6. Example: Python code for portfolio optimization

1. Economic viability of energy investments

- **Capital budgeting and profitability** accounting are necessary for assessing the economic viability of energy investments
- The methodology for energy investments does not differ fundamentally from other applications, but there exist some **unique characteristics of investment in energy technologies**, e.g. long planning, construction, and operation periods that make the result of an investment decision strongly dependent on the discounting of future cash flows
- Financial appraisal techniques **require a forecast of future flows of costs and revenues** over the lifetime of the investment
- In this context, only the **costs and revenues directly associated with the considered investments** should be taken into account

NOTE: e.g. in the decision process of retrofitting a power plant, or analyzing the operation of existing power plant the initial construction (investment) cost of the plant is irrelevant

1. Economic viability of energy investments

The **most important cost components** are:

- Costs for fuel and emission rights, depending on output-related fuel efficiency
- Accelerated degradation of the installation due to thermal stress resulting from temperature change in boilers and pipes
- Fuel losses during start-up and shut-off periods

Marginal unit cost = the sum of these costs divided by the additional production

Marginal cost describes the economic impact on the plant operation

- Investment outlay, personnel and administration costs are relevant for the evaluation of long-term investment decisions

1. Economic viability of energy investments

The **revenues** $\boldsymbol{R_{t,i}}$ per period t generated by the investment i **depend on**:

- Installed capacity $\boldsymbol{Cap_i}$ of power plant (technology) i [in MW of electricity]
- The capacity factor $\boldsymbol{F_{cap,i}}$ for power plant (technology) i specifying the average expected percentage of annual full load operation [in %]
- The average expected price of energy sales at time t $\boldsymbol{P_{el,t}}$ [in €/MWh]

$$R_{t,i} = P_{el,t} \cdot Q_{t,i}$$

$$Q_{t,i} = Cap_i \cdot F_{cap,i} \cdot 8760$$

1. Economic viability of energy investments

Expected future annual costs, can be divided into **variable** and **fixed cost**

- **Variable cost c_{var}** includes the cost of intermediate inputs such as annual expenses for raw materials, fuels, emission rights, waste disposal, and to some extent also wages
- **Fixed cost c_{fixe}** amounts to the annualized investment expenditures (Inv)

Net Present Value (NPV) is given by:

$$NPV = -Inv + \sum_{t=1}^{T} \frac{(P_{el,t} - c_{var,t}) \cdot Q_t}{(1+r)^t}$$

where r is the discount rate

An investment with a positive NPV will be profitable and investment with a negative NPV will be not profitable

2. Definition of portfolio selection criterion for power generation assets

Definition of present value as portfolio selection criterion for power generation assets

(Net) Present Value of the i technology per installed capacity (€/MW):

$$\boldsymbol{PV_i = \sum_{t=1}^{T} \frac{CF_{t,i}}{(1+WACC)^t}} / Cap_i \qquad \textbf{(1)}$$

where

- Cap_i … installed capacity of power plant (technology) i [MW]
- $WACC$ … Weighted Average Cost of Capital (discount rate) [%]
- t … lifetime [a]
- $CF_{t,i}$ … annual cash flow of technology i [€] at time t given as:

$$\boldsymbol{CF_{t,i} = R_{t,i} - C\_fuel_{t,i} - C\_CO_{2_{t,i}} - C\_OM_{t,i} - \delta_{t,i}} \qquad \textbf{(2)}$$

2. Definition of portfolio selection criterion for power generation assets

$$\boldsymbol{CF_{t,i} = R_{t,i} - C\_fuel_{t,i} - C\_CO_{2_{t,i}} - C\_OM_{t,i} - \delta_{t,i}}$$

where

- $R_{t,i}$ … revenues from energy production at time t [€] given as:

$$R_{t,i} = P_{el,t} \cdot Q_{t,i} \qquad \textbf{(3)}$$

where

- $P_{el,t}$ … electricity price at time t [€/MWh]
- $Q_{t,i}$ … electricity generation of technology i at time t [MWh] given as:

$$Q_{t,i} = Cap_i \cdot F_{cap,i} \cdot 8760 \qquad \textbf{(4)}$$

where

- $F_{cap,i}$ … capacity factor for power plant (technology) i [%]

2. Definition of portfolio selection criterion for power generation assets

$$\boldsymbol{CF_{t,i} = R_{t,i} - C\_fuel_{t,i} - C\_CO_{2_{t,i}} - C\_OM_{t,i} - \delta_{t,i}}$$

where

- $C\_feul_{t,i}$ … fuel costs at time t [€] given as:

$$C\_fuel_{t,i} = P_{fuel,t} \cdot F_{consum,t} \tag{5}$$

where

- $P_{fuel,t}$ … fuel price at time t [€/MWh or €/tonne] (depends on the fuel)
- $F_{consum,t}$ … fuel consumption at time t given as:

$$F_{consum,t} = Q_{t,i} \cdot \frac{1}{HV_i \cdot \eta_i} \cdot Con_{factor} \tag{6}$$

where

- η_i … net thermal efficiency of technology i [%]
- HV_i … heating value of technology i
- Con_{factor} … conversion factor

2. Definition of portfolio selection criterion for power generation assets

$$\boldsymbol{CF_{t,i} = R_{t,i} - C\_fuel_{t,i} - C\_CO_{2_{t,i}} - C\_OM_{t,i} - \delta_{t,i}}$$

where

- $C\_CO_{2_{t,i}}$ … carbon dioxide costs at time t [€] given as:

$$C\_CO_{2_{t,i}} = (Q_{t,i} \cdot SF_{CO_2,i}) \cdot P_{CO_2,t} \qquad \textbf{(7)}$$

where

- $SF_{CO_2,i}$ … specific $\mathrm{CO_2}$ emission factor for technology i [t $\mathrm{CO_2}$/MWh]
- $P_{CO_2,t}$ … $\mathrm{CO_2}$ price at time t [€/t $\mathrm{CO_2}$]

2. Definition of portfolio selection criterion for power generation assets

$$\boldsymbol{CF_{t,i} = R_{t,i} - C\_fuel_{t,i} - C\_CO_{2_{t,i}} - C\_OM_{t,i} - \delta_{t,i}}$$

where

- $C\_OM_{t,i}$ … operation and maintenance costs at time t [€] given as:

$$C\_OM_{t,i} = Fixed\_OM_{t,i} \cdot Cap_i + Var\_OM_{t,i} \cdot Q_{t,i} \qquad \textbf{(8)}$$

where

- $Fixed\_OM_{t,i}$ … fixed operation and maintenance costs at time t for technology i [€/MW]
- $Var\_OM_{t,i}$ … variable operation and maintenance costs at time t for technology i (excl. fuel costs) [€/MWh]

2. Definition of portfolio selection criterion for power generation assets

$$\boxed{\boldsymbol{CF_{t,i} = R_{t,i} - C\_fuel_{t,i} - C\_CO_{2_{t,i}} - C\_OM_{t,i} - \delta_{t,i}}}$$

where

- $\delta_{t,i}$ … depreciation (depreciation time = lifetime) at time t for technology i [€] given as:

$$\delta_{t,i} = \frac{Invest.costs_i \cdot Cap_i}{lifetime_i} \tag{9}$$

where

- $Invest.costs_i$ … investment costs for technology i for one unit of installed capacity [€/MW]

3. Mean-variance portfolio selection model for power generation assets

Mean-Variance (MV) portfolio selection model (Markowitz, 1952)

$$\sum_{i=1}^{n} E(PV_i)x_i \rightarrow max \qquad \textbf{(10)}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{n} x_i x_j cov_{ij} \rightarrow min \qquad \textbf{(11)}$$

$$\text{subject to:} \quad \sum_{i=1}^{n} x_i = 1 \quad \text{and} \quad 0 \leq x_i \leq x_{i,max}$$

where: x_i … share of technology i,

$E(PV_i)$ … expected return of (net) present value of technology i,

$cov_{i,j}$ … covariance between expected return of technology i and j

Step 1
Simulation of inputs data for portfolio analysis

- Simulation of mean value and standard deviation of $E(PV_i)$ per installed unit of capacity for each technology considered into portfolio analysis

Input		Output
▪ Technical and economic data for power plants considered ▪ Electricity, fuel and CO_2 prices given as stochastic distribution ▪ EEG prices (if necessary)	**Monte Carlo Simulation of PV_i values in Python**	▪ Present values PV_i/installed unit of capacity for each technology ▪ Mean value and standard deviation of PV_i/install capacity for each technology ▪ Correlation matrix of PV_i/installed unit of capacity for all technologies ▪ Variance-covariance matrix for all technologies

Step 2
Mean-variance portfolio optimization

- Selection of the optimal portfolio structure

Input		Output
▪ Present values PV_i/installed unit of capacity for each technology	**Portfolio optimization in Python**	▪ Graphical presentation of all portfolios and efficient frontier ▪ Portfolio structure in table form with the share of each technology from efficient frontier

5. Example: Python code for simulation of present values

Initialization of libraries and functions needed for the calculation procedure Step 1

```
import numpy as np
import matplotlib.pyplot as plt
import csv
```

Definition of all parameters needed for calculations and Monte Carlo Simulation

- Prices:

```
# Define the parameters of the normal distributions in the global scope
mean_el, std_el = 50.00, 10.00 # electricity_price €/MWh
mean_CO2, std_CO2 = 12.00, 4.0 # CO2 price €/tCO2
mean_coal, std_coal = 45.00, 8.50 # hard coal €/t SKE
mean_lignite, std_lignite = 60.00, 4.00 # lignite €/t
```

- Other parameters:

```
# T: Lifetime for each technology [a]
T = np.array([40, 100, 40, 20])
# t: rest operation time = Lifetime [T] - opertion year [a]
t = np.array([15, 25, 10, 5])
# Weighted average cost of capital
WACC = np.array([0.092, 0.092, 0.092, 0.092])
# Common ratio: Discountrate based on WACC
common_ratio = (1 + WACC)**-1
```

5. Example: Python code for simulation of present values

$$\boldsymbol{CF_{t,i} = R_{t,i} - C\_fuel_{t,i} - C\_CO_{2_{t,i}} - C\_OM_{t,i} - \delta_{t,i}}$$

$$R_{t,i} = P_{el,t} \cdot Q_{t,i}$$

$$Q_{t,i} = Cap_i \cdot F_{cap,i} \cdot 8760$$

Define the capacity factor and installed capacity to calculate electricity generation of technology *i:*

```
# F_Cap: Capcity factor of power plant i [%]
F_Cap = np.array([0.60, 0.05, 0.82, 0.20])
# Cap: Installed capcity of power plant i [MW]
Cap = np.array([1200.00, 500.00, 700.00, 25.00])

# Q_el: Electricity generation of technology [MWh]
Q_el = F_Cap * Cap * 8760
```

Define the function for the calculation of revenues from electricity generation – Eq. (3):

```
def Revenues_electricity_sales(Pel, Q_el):
    result = Pel * Q_el
    return result
```

5. Example: Python code for simulation of present values

$$\boxed{\boldsymbol{CF_{t,i} = R_{t,i} - C\_fuel_{t,i} - C\_CO_{2\,t,i} - C\_OM_{t,i} - \delta_{t,i}}}$$

$$C\_fuel_{t,i} = P_{fuel,t} \cdot F_{consum,t}$$

$$F_{consum,t} = Q_{t,i} \cdot \frac{1}{HV_i \cdot \eta_i} \cdot Con_{factor}$$

Define all parameters needed for fuel consumption:

```
# HV: Heating value
HV = np.array([8.06, np.inf, 4.17, np.inf])
# Net thermal effiency [/]
eta = np.array([0.44, 0.80, 0.39, 1.00])
# Con_factor: Converrion factors for different technologies
Con_factor = np.array([0.97, 1, 1, 1])

# F_consum: fuel consumption, different units depend on the technology
F_consum = (Q_el / (HV * eta)) * Con_factor
```

Define the function for the calculation of the costs of fuel consumption – Eq. (5):

```
def fuel_cost(P_coal, P_lignite, F_consum):
    result = np.array([F_consum[0] * P_coal, F_consum[1]*1,
                       F_consum[2] * P_lignite, F_consum[3]*1])
    return result
```

5. Example: Python code for simulation of present values

$$\boxed{CF_{t,i} = R_{t,i} - C\_fuel_{t,i} - C\_CO_{2_{t,i}} - C\_OM_{t,i} - \delta_{t,i}}$$

$$C\_CO_{2_{t,i}} = (Q_{t,i} \cdot SF_{CO_2,i}) \cdot P_{CO_2,t}$$

Define all parameters needed for CO_2 costs calculation:

```
# SF_CO2: Specific CO2 emission factor for technology [t CO2/MWh]
SF_CO2 = np.array([0.917, 0, 0.929, 0])
# CO2_emission: Level of CO2 emission for each technology [t CO2]
CO2_emission = Q_el * SF_CO2
```

Define the function for the calculation of the CO_2 costs – Eq. (7):

```
def carbon_cost(P_CO2, CO2_emission):
    return P_CO2 * CO2_emission
```

5. Example: Python code for simulation of present values

$$\boldsymbol{CF_{t,i} = R_{t,i} - C\_fuel_{t,i} - C\_CO_{2_{t,i}} - C\_OM_{t,i} - \delta_{t,i}}$$

$$C\_OM_{t,i} = Fixed\_OM_{t,i} \cdot Cap_i + Var\_OM_{t,i} \cdot Q_{t,i}$$

Define all parameters needed and function for O&M costs calculation – Eq. (8):

```
# Fixed_OM: Fixed operation and maintence costs [€/MW]
Fixed_OM =  np.array([12500, 10500, 40000, 20000])
#Var_OM: Variable operation and maintence costs [€/MWh]
Var_OM = np.array([0, 0, 0, 0])

# C_OM: Operation and maintanance costs [€]
C_OM = Fixed_OM * Cap + Var_OM * Q_el
```

5. Example: Python code for simulation of present values

$$\boxed{CF_{t,i} = R_{t,i} - C\_fuel_{t,i} - C\_CO_{2_{t,i}} - C\_OM_{t,i} - \delta_{t,i}}$$

$$\delta_{t,i} = \frac{Invest.costs_i \cdot Cap_i}{lifetime_i}$$

Define all parameters needed for depreciation calculation – Eq. (9):

```
# Invest_costs: Investment costs for technologies for one unit installed capacity [€/MW]
Invest_costs = np.array([800000, 1500000, 900000, 900000])
# delta: Depreciation for each technology [€]
delta = (Invest_costs * Cap)/T
```

5. Example: Python code for simulation of present values

(Net) Present Value of the i technology per unit of installed capacity (€/kW):

$$\boldsymbol{PV_i = \sum_{t=1}^{T} \frac{CF_{t,i}}{(1+WACC)^t}} / Cap_i$$

Present value is given as cash flow per installed capacity [in kW]

Define the function for the calculation of present value – Eq. (1):

```
# Slide 3: Calculation of present value = sum of discounted cash flow
def sum_DCF(CF, common_ratio, t):
    return ((CF * (1 - np.power(common_ratio, t)) / (1 - common_ratio)) / (Cap * 1000)) / 1000
```

Over 1,000 € for the simplicity by the results presentation

5. Example: Python code for simulation of present values

Simulation of discounted cash flow

```
# Define the number of iteration for simulation, number of assets
# and create an empty matrix to store the results
num_runs = 100000
num_assets = 4
PV_matrix = np.empty((num_runs, num_assets))
```

```
# Set the random seed to ensure reproducibility
np.random.seed(123)
```

```
# Run the program and record the results
for i in range(num_runs):
    P_el = np.random.normal(mean_el, std_el)
    P_CO2 = np.random.normal(mean_CO2, std_CO2)
    P_coal = np.random.normal(mean_coal, std_coal)
    P_lignite = np.random.normal(mean_lignite, std_lignite)
    R_energysales = Revenues_electricity_sales(P_el, Q_el)
    C_fuel = fuel_cost(P_coal, P_lignite, F_consum)
    C_CO2 = carbon_cost(P_CO2, CO2_emission)
    CF = R_energysales - C_fuel - C_CO2 - C_OM - delta
    PV = sum_DCF(CF, common_ratio, t)
    PV_matrix[i, :] = PV
```

Prices as stochastic parameter used in calculation and defined before

Functions define before and used by the simulation

Cash flow defined with Eq. (2)

Simulated present values for each technology needed for step 2

Simulation of discounted cash flow

Calculation of the outputs for Step 1:

```
# Calculate the outputs
E_PV = mean_values = np.mean(PV_matrix, axis=0)
PV_std_dev = np.std(PV_matrix, axis=0)
correlation_matrix = np.corrcoef(PV_matrix, rowvar=False)
var_cov_matrix = np.cov(PV_matrix, rowvar=False)
```

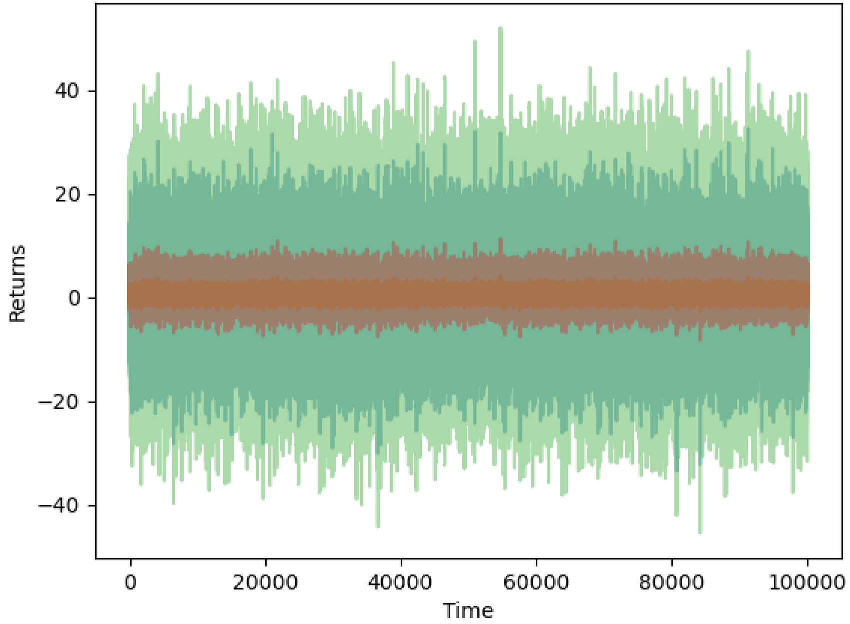
How to save the outputs:

```
# Save the outputs to CSV files
np.savetxt('Present Values .csv', PV_matrix, delimiter=',')
np.savetxt('Expected Present Values .csv', E_PV, delimiter=',')
np.savetxt('Standard deviation.csv', PV_std_dev, delimiter=',')
np.savetxt('Correlation matrix.csv', correlation_matrix, delimiter=',')
np.savetxt('Variance-covariance.csv', var_cov_matrix, delimiter=',')
```

Simulation of discounted cash flow

How to plot the outputs:

```
# Ploting of simulated present value data for each technology
fig = plt.figure()
plt.plot(PV_matrix, alpha=.4);
plt.xlabel('Time')
plt.ylabel('Returns')
plt.show()
```



6. Example: Python code for portfolio optimization

Initialization of libraries and functions needed for the calculation procedure Step 2

```
import numpy as np
import matplotlib.pyplot as plt
import cvxopt as opt
from cvxopt import blas, solvers
import pandas as pd
import plotly
import cufflinks
```

Initialization of the generator for the simulation of the rate of return

```
## Generator for simulation
np.random.seed(123)
```

Definition of random weights for portfolio assets

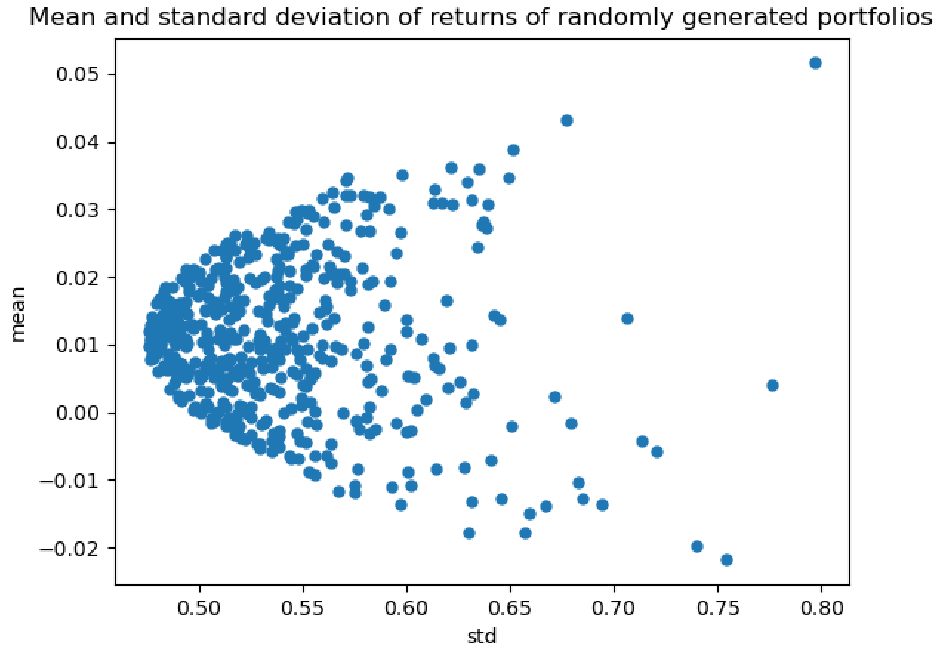
```
# Produces n random weights that sum to 1
def rand_weights(n):
    k = np.random.rand(n)
    return k / sum(k)
```

Random portfolios for simulated returns of assets

```
# Returns the mean and standard deviation of returns for a random portfolio
def random_portfolio(returns):
    p = np.asmatrix(np.mean(returns, axis=1))
    w = np.asmatrix(rand_weights(returns.shape[0]))
    C = np.asmatrix(np.cov(returns))

    mu = w * p.T
    sigma = np.sqrt(w * C * w.T)

    # This recursion reduces outliers to keep plots pretty
    if sigma > 2:
        return random_portfolio(returns)
    return mu, sigma

n_portfolios = 500
means, stds = np.column_stack([
    random_portfolio(return_vec)
    for _ in range(n_portfolios)])
```

Plotting of randomly generated portfolios (mean and standard deviation of returns)

```
fig = plt.figure()
plt.plot(stds, means, 'o', markersize=5)
plt.xlabel('std')
plt.ylabel('mean')
plt.title('Mean and standard deviation of returns of randomly generated portfolios')
plt.show()
```



E.ON Energy Research Center
RWTH AACHEN UNIVERSITY

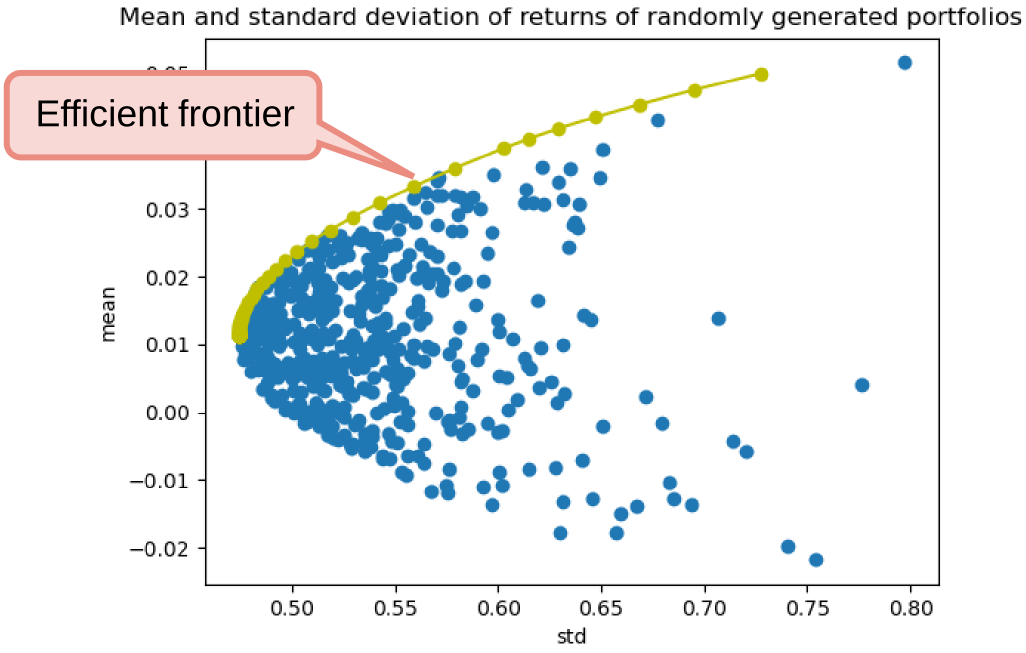
6. Example: Python code for portfolio optimization

Optimal portfolios for simulated returns of assets

```python
def optimal_portfolio(returns):
    n = len(returns)
    returns = np.asmatrix(returns)

    N = 100
    ## mus vector produces a series of expected return values $\mu$
    ## in a non-linear and more appropriate way
    mus = [10**(5.0 * t/N - 1.0) for t in range(N)]
    #print ('mus')
    #print(mus)

    # Convert to cvxopt matrices
    S = opt.matrix(np.cov(returns))
    pbar = opt.matrix(np.mean(returns, axis=1))

    # Create constraint matrices
    G = -opt.matrix(np.eye(n))   # negative n x n identity matrix
    h = opt.matrix(0.0, (n ,1))
    A = opt.matrix(1.0, (1, n))
    b = opt.matrix(1.0)

    # Calculate efficient frontier weights using quadratic programming
    portfolios = [solvers.qp(mu*S, -pbar, G, h, A, b)['x']
                  for mu in mus]
    ## CALCULATE RISKS AND RETURNS FOR FRONTIER
    returns = [blas.dot(pbar, x) for x in portfolios]
    risks = [np.sqrt(blas.dot(x, S*x)) for x in portfolios]
    shares = [np.asarray(x) for x in portfolios]

    return shares, returns, risks

weights, returns, risks = optimal_portfolio(return_vec)
```

Plotting of optimal portfolios (mean and standard deviation of returns) and efficient frontier

```
fig = plt.figure()
plt.plot(stds, means, 'o')
plt.ylabel('mean')
plt.xlabel('std')
plt.title('Mean and standard deviation of returns of randomly generated portfolios')
plt.plot(risks, returns, 'y-o')
plt.show()
```



6. Example: Python code for portfolio optimization

How to save the outputs:

```
# Save the outputs to CSV files
np.savetxt('Portfolio_returns.csv', returns, delimiter=',')
np.savetxt('Portfolio_risk.csv', risks, delimiter=',')
np.savetxt('Portfolio_shares.csv', weights, delimiter=',')
```

How to save the plotted graphs:

```
plt.savefig("Efficient_frontier.png", dpi=None, format='png')
```

Licensing

Case study "Selection of optimal power plant generation mix"

Chair of Energy Economics and Management
Institute for Future Energy Consumer Needs and Behavior
Prof. Dr. Reinhard Madlener, Dr. Barbara Glensk, Qinghan Yu M.Sc.
RWTH Aachen University April 2023

Participating universities:

Funded by:

Ministerium für Kultur und Wissenschaft des Landes Nordrhein-Westfalen

Contact

E.ON Energy Research Center,
Institute for Future Energy Consumer Needs and Behavior (FCN)
Mathieustraße 10
52074 Aachen
Germany

Dr. Barbara Glensk
T +49 241 80 49833
F +49 241 80 49829
bglensk@eonerc.rwth-aachen.de

Further Information:

http://www.eonerc.rwth-aachen.de/fcn

FCN | Future Energy Consumer Needs and Behavior
E.ON Energy Research Center
RWTH AACHEN UNIVERSITY