COPENHAGEN BUSINESS SCHOOL, 2012

Allocation of Risk Capital

Aleksandrs Smilgins

Dansk titel:	Allokering af Risikokapital
CPR:	XXXXXX-XXXX
Studium:	Cand. Merc. (Mat.)
Afleveringsdato:	31-08-2012
Vejleders navn:	Jens Leth Hougaard
Institut:	Institut for Produktion og Erhvervsøkonomi
Opgavens type:	Kandidatafhandling
Antal anslag:	173561

Underskrift:

Acknowledgements

The writing of this thesis has been a significant academic challenge. Without the guidance and support of several individuals certain parts of this study would not have been possible. I owe my deepest gratitude to the following people:

- Jens Leth Hougaard, who agreed to be my supervisor. His knowledge and support inspired and motivated me.
- Ove Sten Jensen and Lars Mayland Nielsen who gave me an opportunity to write this thesis at Danmarks Nationalbank. My colleagues and the environment as a whole motivated me. The provided computer with a powerful processor made the simulations in chapter 6 possible.
- Wolfram Mathematica Student Support Forum, Matlab Central Answers and Wilmott Quantitative Finance Forum which gave the answers to my questions.

Abstract

En finansiel institution er tit interesseret i at allokere den totale risikokapital ud på de underafdelinger, subporteføljer, business lines eller andre typer af underenheder den består af. Interessen skyldes den diversifikationseffekt der finder sted i institutionen: typisk er summen af ”risici” i de forskellige underenheder mindre end ”risikoen” i hele den finansielle institution. En fair allokering af risikoen kan bruges til bestemmelse af kapitalkrav og til præstationsevaluering (fx forventet resultat i forhold til allokeret risiko) af den enkelte underenhed. I opgaven antages at, der eksisterer en finansiel institution, hvis samlede portefølje består af flere forskellige subporteføljer, som er eksponeret over for "market risk". Opgaven besvarer følgende spørgsmål:

- Hvilke metoder kan benyttes til målingen af risici (de såkaldte risikomål)?
- Hvilke metoder kan benyttes til allokeringen af den samlede risiko ud på de enkelte underenheder (de såkaldte allokeringsregler)?
- Hvilke metoder virker bedst i hvilke situationer?

I opgaven bruges to velkendte risikomål: Value-at-Risk og Expected Shortfall. Til at allokere den samlede risiko ud på underenhederne, bruges resultaterne fra spilteori og omkostningsfordeling. Valg af sådanne teorier er oplagte, da en underafdeling får gavn af at være en del af den samlede institution og samtidig gavner de andre ved at bringe ekstra diversifikationseffekt ind i institutionen. På den måde forbindes de enkelte subporteføljer med spillerne. I alt otte forskellige allokeringsregler bliver præsenteret, hvor spillerne først antages at være udelelige og derefter fuldt delelige. Desværre viser det sig, at der ikke findes en allokeringsregel der opfylder alle ønskede egenskaber. Et større simulationsstudie, der udregner core-compatibility ratios af udvalgte allokeringsregler viser, at det desværre også er et praktisk problem. I praksis er det derfor nødvendigt at vælge en allokeringsregel, der bedst passer til den aktuelle problemstilling. Det er dog ikke længere nødvendigt at tænke på kompleksiteten af udregningerne, da det vedlagte MatLab program kan anvende alle otte allokeringsregler på vilkårlige problemer. Det velkendte trade-off mellem teoretiske egenskaber og praktisk enkelthed er dermed væk.

Table of contents

Introduction

This thesis has a title ”Allocation of Risk Capital”. To understand the topic covered in this thesis, it is important to understand the different parts of the title separately. First, what is “risk”? There are many ambiguous and inconsistent meanings attached to the definition of "risk", which both leads to widespread confusion and means that very different approaches to risk management are taken in different fields. In general, risk means the potential that a chosen activity will lead to an undesirable outcome, such as a loss or even bankruptcy. A company or a financial unit may have many types of risk. Examples could be:

- **Market risk**: the risk due to the change in value of the market risk factors. The associated market risks are interest rate risk, currency risk, equity risk, commodity risk.
- **Credit risk**: risk that the borrower will not meet its payment obligations and will default on any type of debt.
- **Liquidity risk**: risk of not being able to trade an asset or security quickly enough to prevent a loss.
- **Operational risk**: risk arising from people, systems and processes. Examples could be fraud (e.g. rogue traders), technology failures, external events (e.g. earthquakes, terrorist attacks), political risks (sudden change in policies in a certain country), etc.

Risk management is the identification and assessment of risks followed by application of resources to control and monitor the potential unfortunate events or to maximize the realization of opportunities. Once the risks are identified, the risk manager seeks to minimize the impact of negative events. Hence, inadequate risk management can result in severe consequences for companies and individuals. For example, loose credit risk management was one of the reasons for the recession that began in 2008.
In this thesis only market risk will be in focus, i.e. risk that the value of a portfolio will decrease due to the fluctuations in prices. Market risk, as well as other types of risk, is very difficult to measure, since it is unobservable in contrast to prices, returns, value of a portfolio, etc. Risk can be measured by tools called “risk measures” designed to measure risks from the perspective of present. In general there is a clear relationship between the risk and return of an investment: the more risk you take, the more return you expect. Risk management occurs anytime an investor quantifies the potential losses in an investment and then takes an appropriate action given the investment objectives and risk tolerance. As insurance against the uncertainty of the portfolio values in future a company would usually hold an amount of investments with extremely low risk, which acts as a buffer to reduce the effects of potential unexpected losses. This buffer is called the **risk capital** of the company. “Risk Capital” is thus the amount of riskless capital which added to the firm’s assets, ensures that the value of the company in future is acceptable to the management, chief risk officer, regulator or others. In case of an external regulator a specific amount of risk capital is no longer an option, but an amount required in order to be able to hold a risky position. From a financial perspective, holding an amount of money in very low risk is seen as a burden, because of the very low return on this amount of money, which instead could be invested in something else with much higher return.
Usually a company has several subdivisions, subportfolios, business lines or other types of subunits. If these subunits acted independently from each other, then each subunit would be required to hold its own specific amount of risk capital, which would cover potential losses of that specific subunit. But because these subunits stand together by being a part of the same company, there is a diversification benefit. The

underlying intuition is that the risks of different subunits are usually not perfectly correlated. For example in case with different subportfolios some hedge potential may arise. Thus the amount of risk capital required for the whole company is usually smaller than the sum of the amounts of risk capital that would be required to be withheld by the subunits separately. The question is how the diversification benefit should be shared between the subunits? Another side of the same coin is to ask how the amount of risk capital required to be withheld by the company should be shared between the different subunits? This is the main question that is covered in this thesis, hence the name “Allocation of Risk Capital”.

A proper allocation of risk capital can be used not only for the allocation of capital requirements, but also for performance evaluation. Knowing both the profit generated by a certain subunit *and* the risk taken by that subunit allows for a much wiser comparison than only knowing profits. For example, if a certain subunit generates a moderate profit, but at the same time hedges some other subunits, its value for the company is very significant. A properly allocated risk capital would be low for that subunit and by using some sort of RORAC approach of the form $\frac{\text{Expected profit}}{\text{Allocated risk capital}}$ the management would easily see the importance of that subunit. On the other hand, without allocation of the risk capital, management would only be able to see the subunit’s individual expected profit compared to its individual risk. In this case, the diversification benefit that is brought to the company by that subunit would not be taken into consideration.

In this thesis we consider a company with a number of subunits. Each subunit has its own subportfolio. The portfolio of the whole company is called the grandportfolio and consists of all the subportfolios in the company. We measure the risk of the grandportfolio by one of the risk measures and seek to allocate the risk capital to the different subportfolios as fair as possible. To allocate the risk capital properly, we will use results from Game Theory and Cost Sharing Theory. Use of these theories in this case is obvious, because each subunit benefits from being a part of the company and at the same time gives some diversifying benefit to others by being a part of the company. The challenge is finding a stable and equitable way of sharing the overall diversification benefit to the subportfolios. So, the questions we seek to answer are:

- Which methods are there for measuring risk?
- Which methods are there for allocating a given risk measurement value across the subportfolios?
- In which circumstances different methods are more appropriate?

Usually the methods with the best theoretical properties are more advanced and require a lot more effort to be applied on real-life situations. Unlike Finance, Game Theory is an area which today is rarely combined with Computer Science. Nevertheless, in this thesis all included methods of calculating and allocating risk were programmed in MatLab. This thesis shows that Game Theory and Computer Science can be an excellent combination, which allows applying the advanced theory on real-life situations.

This thesis is structured as follows. First, the basics of the risk management are reviewed in chapter 2. Chapter 3 presents the necessary basics of Game Theory and Cost Sharing Theory. The different methods of allocating risk that are presented in chapter 3 were programmed in Matlab and the programming part itself is explained in chapter 4. Chapter 5 covers different examples where the difference between the methods of allocating risk is presented. Finally, chapter 6 presents some simulation results, which show core compatibility ratios (defined in chapter 2) of selected methods of allocating risk.

1. Notation and assumptions

As previously stated in the introduction, we consider a financial institution that has n subunits. Throughout this thesis we will use the following notation:

- $N = \{1,..., n\}$ denotes the set of subunits which have their own subportfolios.
- $\Omega = \{\omega_1,..., \omega_m\}$ denotes the set of states in the world.
- $\pi(\omega) > 0$ denotes the probability that the state $\omega \in \Omega$ occurs.
- $X_i(\omega)$ denotes the value of subportfolio *i* in state $\omega \in \Omega$.
- $X = \{X_i,..., X_n\}$ denotes the profit of the *n* subportfolios.
- *V* denotes the set of random variables on Ω.

Explanation: each subunit has a portfolio which consists of *n* subportfolios indexed by *1,…,n*. Subportfolio *i* generates a random profit X_i at a given future date, so the total profit is given by $\sum_{i=1}^{n} X_i$. Negative X_i means a loss. The institution as a whole has one grandportfolio which consists of all subportfolios in the institution.

Assumptions about the subportfolios:

- Stationarity requirement: for example 1% fluctuation in returns is equally likely to occur at any point in time
- Day-to-day fluctuations in returns are independent: for example a decline of a subportfolio on one day of x % has no predictive power regarding returns on the next day.
- Non-negativity requirement: subportfolios cannot attain negative values.
- All subportfolios are constant over time: the amounts invested in each asset do not change.
- There are no dividends

Even though some of the above mentioned assumptions are more realistic than others, they are all very common in the academic literature.

2. Risk measures

In this chapter we review the basics of risk management. Future risky business or investment is unpredictable by definition, so we have to predict the future outcomes. Firstly, we have to specify the time horizon, i.e. a future time at which we are interested to measure risk. The time horizon varies a lot among different agents – it can vary from a couple of hours for a trader to several years for a pension fund. Nevertheless, companies are often interested in measuring the risk of the end-of-year financial result. Once the time horizon is specified, we need to estimate the distribution of the portfolio value at that specific time in future. Different stakeholders are interested in different parts of the distribution. Investors and management are mainly interested in the middle part of the distribution focusing on the likelihood of making a profit as well as a loss. On the other hand, for risk management purposes, we are mostly interested in fluctuations and the left tail. Although fluctuations are annoying, the real fear is downside potential. An event may occur that might cause the capital to fall far enough to interfere with the company's ability to continue normal business. Hence rating agencies and regulators will be mainly focused on extreme downside in the left tail where the existence of the company is in doubt.

In practice the underlying distribution of market prices and returns is unknown, which makes the problem of risk comparisons much harder. Even though today there are many models, including very complicated stochastic models, which are able to estimate the probability density function (PDF) of returns, practically it is impossible to accurately identify the distribution of financial returns. The task of measuring financial risk is further complicated by the fact that financial risk cannot be measured directly, but has to be inferred from the behavior of observed market prices. Risk is a latent variable, i.e. it cannot be measured in the same manner as weight can be measured by a scale. At the end of the trading day the daily return is known while the day's risk is unknown. All we can say is that the risk might be high if prices have fluctuated during the day.

When measuring the risk, we are interested in the changes in the value of the portfolio. These changes are measured by returns.

Definition 2.1. Returns

Usually in literature two types of returns are used. Let P_t *denote the value of a certain portfolio at time t.*

Simple returns (Arithmetic returns):

$$R_t = R_{t-1,t} = \frac{P_t - P_{t-1}}{P_{t-1}} = \frac{P_t}{P_{t-1}} - 1$$

j-period returns:

$$R_{t-j,t} = (1 + R_{t-1,t})(1 + R_{t-2,t-1})\ldots(1 + R_{t-j,t-j+1}) - 1 = \frac{P_t}{P_{t-j}} - 1$$

Continuously compounded returns (Geometric returns):

$$r_t = r_{t-1,t} = \ln(\frac{P_t}{P_{t-1}}) = \ln(P_t) - \ln(P_{t-1}) = \ln(1 + R_{t-1,t})$$

j-period returns:

$$r_{t-j,t} = \ln(\frac{P_t}{P_{t-j}}) = \ln(1 + R_{t-j,t}) = \ln(1 + R_{t-1,t}) + \ldots + \ln(1 + R_{t-j,t-j+1}) = r_{t-1,t} + \ldots + r_{t-j,t-j+1}$$

Note that for small returns and short time horizons geometric returns and arithmetical returns are almost equal. Which method is the right one to choose? Both methods have their advantages. Geometric returns are often used in financial modeling due to their ability to easily move between time periods. For example, if $r_{t-1,t}$ and $r_{t-2,t-1}$ are normally distributed then $r_{t-2,t}$, which is the sum of two normals, is also normally distributed and has a closed form solution. However, when using arithmetic returns, the distribution of the two period return would be a product of two normal random variables. Hence there is little we can say analytically in closed form on the distribution of two period arithmetic return. The advantage of arithmetic returns comes into play when combining assets into portfolios. If R_t^k and w^k are the 1-period arithmetic return of asset *k* and fraction of wealth invested in asset k respectively, then $R_t^{port} = \sum_{k=1}^{K} R_t^k w^k$. However, using similar notation for geometric returns, $r_t^{port} \neq \sum_{k=1}^{K} r_t^k w^k$. Since in this thesis we do not need to move between time periods, but instead need to combine different subportfolios into portfolios, we will use arithmetic returns to measure the risk of the portfolios.

One of the ways of measuring the risk of returns is by assuming a certain distribution of returns, such as normal or Student-t. For example, in finance geometric returns are often assumed to be normal, which is the same as assuming that the distribution of future prices is lognormal. For example, this assumption is the key to simplicity and elegance of the famous Black-Scholes option pricing formula. Note that returns usually are not normally distributed, since the real distributions of returns usually have much fatter tails than the normal distribution, especially the left tail. However, the normality assumption simplifies the model and has many desired characteristics, such as returns can be infinitely low or high (normal distribution), while prices cannot be negative (lognormal distribution). This has been widely recognized:

"As you well know, the biggest problems we now have with the whole evolution of risk is the fat-tail problem, which is really creating very large conceptual difficulties. Because as we all know, the assumption of normality enables us to drop off the huge amount of complexity in our equations …

Because once you start putting in non-normality assumptions, which is unfortunately what characterizes the real world, then these issues become extremely difficult"

Alan Greenspan (1997)

Another way of measuring the risk is by using historical distribution which has to be estimated from historical data. For a portfolio of n assets the sample data might be given as n-dimensional points:

$$(P_1^1, ..., P_1^n), ..., (P_N^1, ..., P_N^n)$$

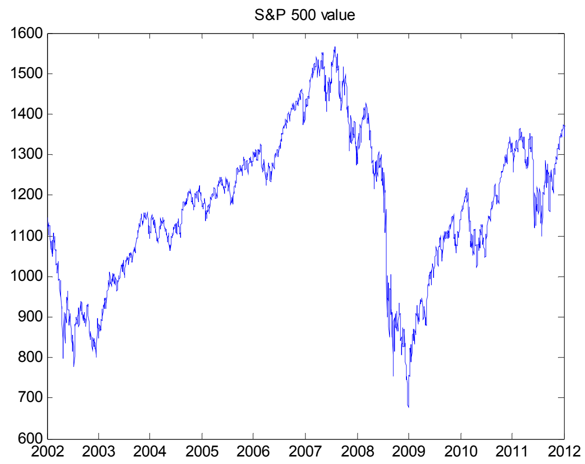
where P_t^i denotes the value (price) of asset *i* at time *t*. When using the historical distribution one assumes that the return in future will simply be one of the observed returns in history. Historical distribution is used in circumstances when the distribution of portfolio value either cannot be calculated analytically or when one wants to avoid making any distribution assumptions. Hence historical distribution can have shapes very different from normal or Student-t distributions. For example, it can be skewed, have several vertices, fatter tails, etc. Note that historical distribution is discrete in contrast to normal or Student-t distributions which are continuous.

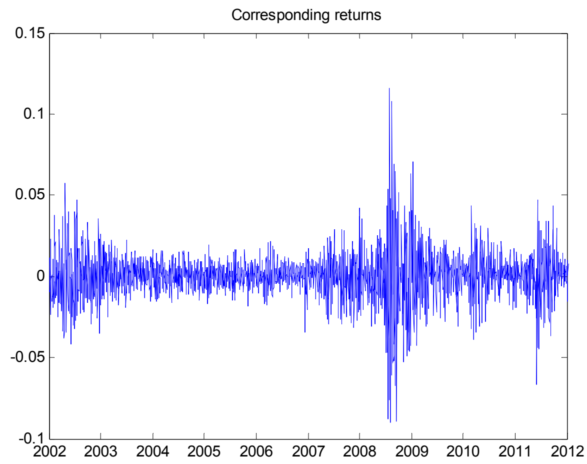
Example 2.1

Assume that we create a new portfolio by investing 100$ in only S&P500 and want to estimate the PDF of the portfolio value in 1 day (tomorrow).

Historical distribution

We decide to base our estimation on the last, say, 10 years of historical data. In Matlab we download daily data of S&P500 close prices from Yahoo Finance for the last 10 years and store it in a price vector. In this example we get 2520 price observations. We create a new vector consisting of corresponding 1-day arithmetic returns to the historical prices by using the formula from definition 2.1. In this example we get 2519 historical returns.

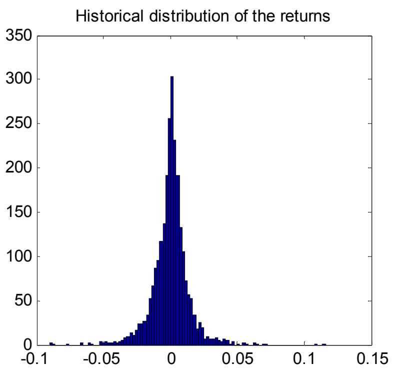


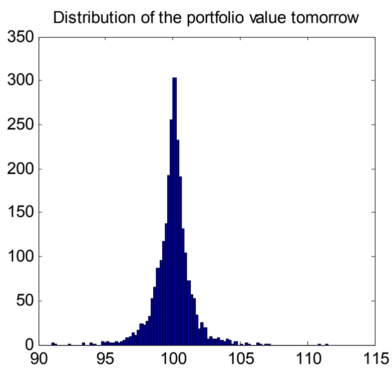


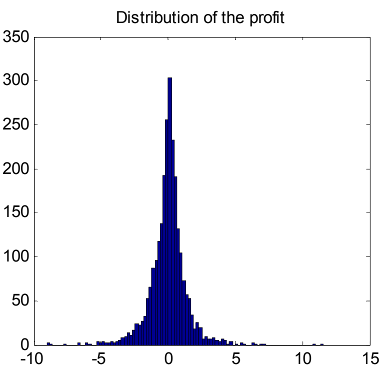
If we assume that from today to tomorrow we will get one of the returns stored in our return vector (with equal probability), then the corresponding vector of possible profits tomorrow can easily be calculated by multiplying the invested amount, 100$, by the possible returns. The corresponding vector of possible values of the portfolio tomorrow is the profit plus the invested amount. We do this in Matlab (arithmetic returns are stored in the variable "ret") :

```
InvAmount=100;
P1=InvAmount*(1+ret);
Profit=InvAmount*ret;
```

Note that, because "ret" is an array (vector), "P1" and "Profit" are arrays too.
The PDFs of possible returns, the portfolio value and possible profit are illustrated below:





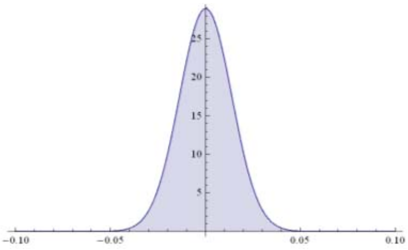


Note that measuring the risk of the portfolio value, measuring the risk of the corresponding returns and measuring the risk of the profit of the portfolio are different sides of the same coin.

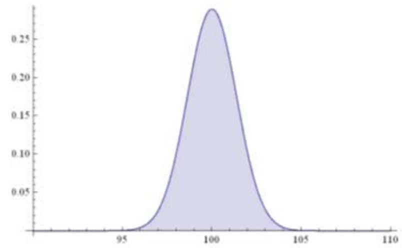
Normal distribution assumption

*Here we can assume that the daily returns are normally distributed. We can use the same historical data to calculate the mean and the standard deviation (SD) of the returns. In this example the mean is 0.000177 and the SD is 0.0138. Hence, the returns from today to tomorrow is by our assumption normally distributed with mean 0.000177 and SD 0.0138; the possible profit is normally distributed with mean 0.000177*100=0.0177 and SD 0.0138*100=1.38; the portfolio value tomorrow is by our assumption normally distributed with mean 100*(1+0.000177)=100.0177 and SD 1.38.*

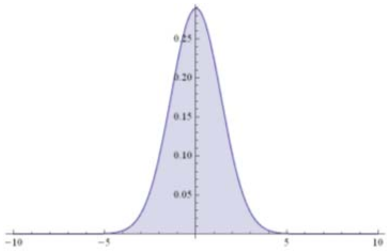
Distribution of the returns



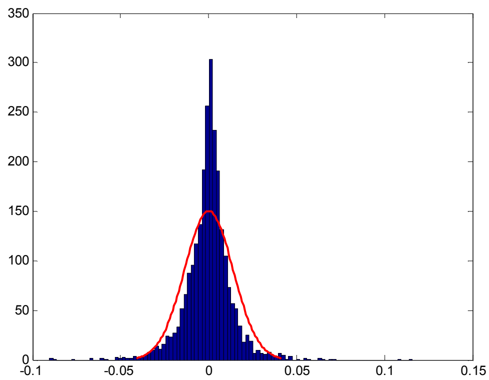
Distribution of the portfolio value

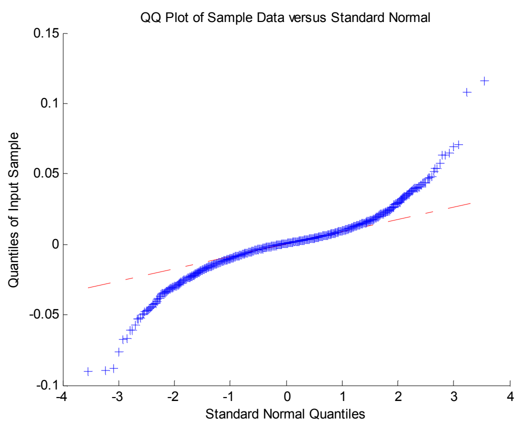


Distribution of the profit



How reasonable is the assumption about normally distributed returns? A histogram of historical returns with a fitted normal distribution on top and a QQ plot might help to answer this question:



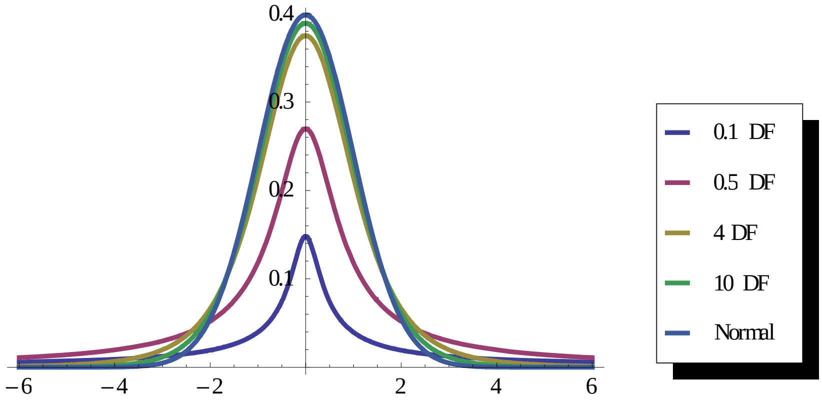


Additionally we calculate the skewness and kurtosis of the historical returns to be 0,0237 and 11,5190 respectively. Recall that for any normal distribution skewness is 0 (symmetric) while kurtosis is 3. As you can see, the historical distribution of returns is not normally distributed. The historical distribution has a higher degree of “peakedness” than a normal distribution. At the same time the historical distribution has much fatter tails, as indicated by the QQ plot.

There are more sophisticated methods for checking whether data is normally distributed, such as Jarque-Beta test or Kolmogorov-Smirnov test. However, these will not be used or demonstrated in this thesis.

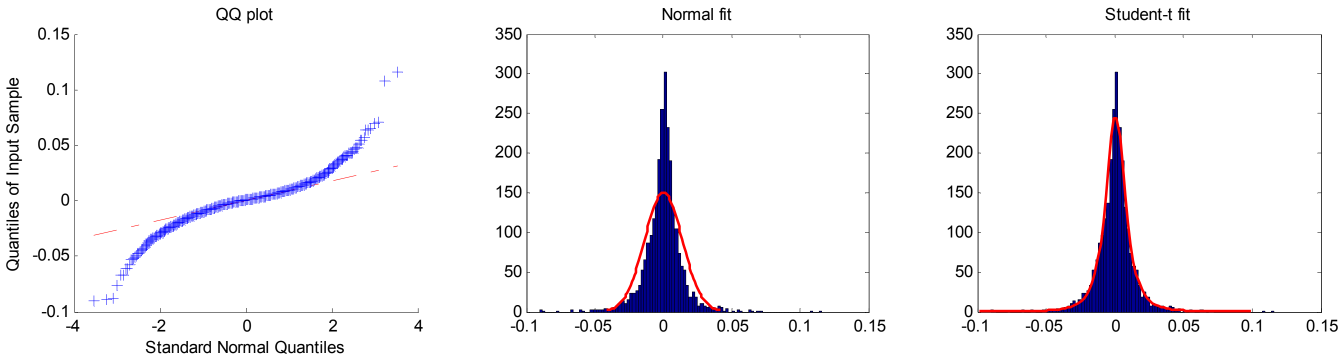
▪

The previous example shows that arithmetic returns are not normally distributed due to fatter tails. An obvious question to ask is whether returns can be assumed to be Student-t distributed. Recall that a Student-t distribution with v degrees of freedom has the variance $\frac{v}{v-2}$. The parameter v indicates how fat the tails are and when $v = \infty$ the Student-t distribution becomes normal. The variance is only defined for $v > 2$.



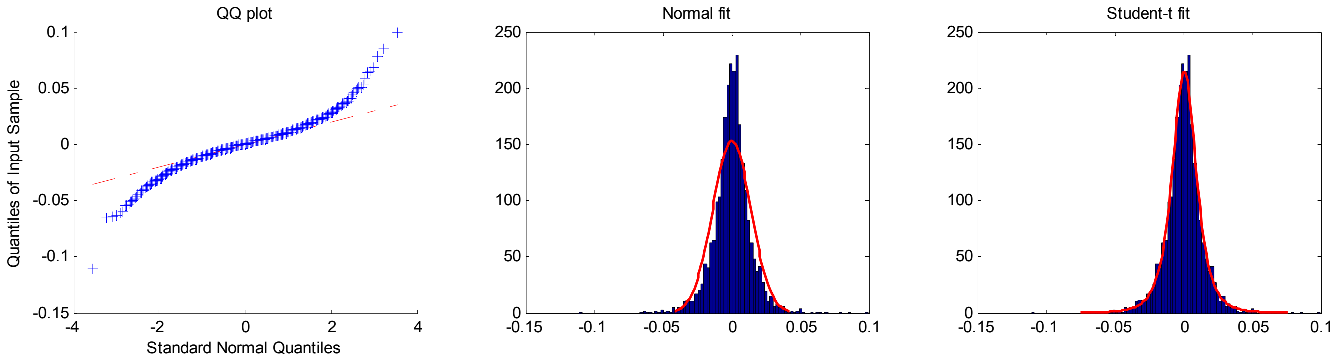
Example 2.1, cont.

We compare the normal fit with the Student-t fit:

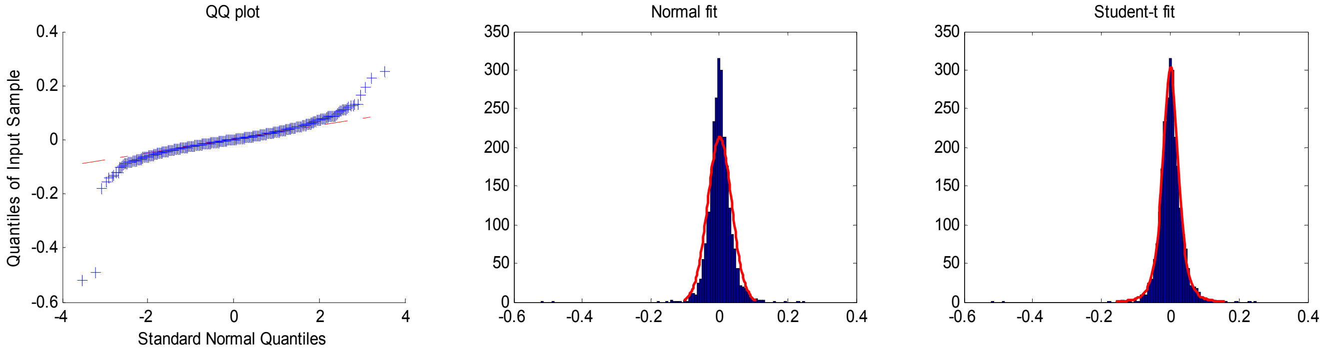


It is obvious that Student-t fit is better than normal. The additional examples below also based on 10 years of data show that this is the case not only with S&P500:

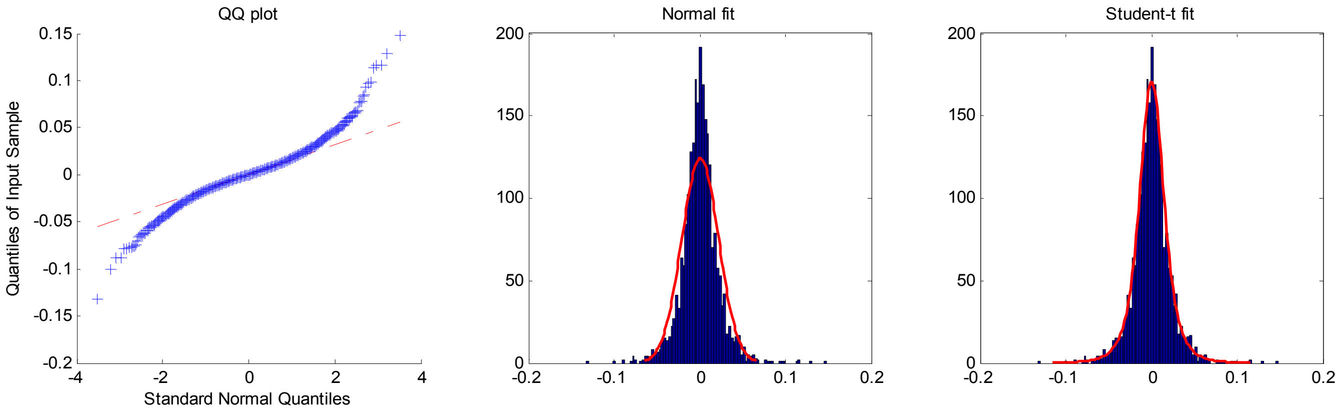
OMX20



GOLD



BMW.DE



Even when we have the distribution of returns, the comparison of different portfolios would be challenging. The most common approach to the problem of comparing the risk of different portfolios having different distributions is to employ a risk measure that represents the risk of a portfolio as a single number that is comparable across portfolios.

Definition 2.2 (see for instance Daníelsson 2011 [2])
A risk measure ρ is a mathematical method for computing risk. It measures the risk of a portfolio from the perspective of the present.
◊

Risk measure ρ is a mapping from a set of random numbers to the real numbers, i.e. we get one specific number that represents the risk of the portfolio. That specific number is called risk measurement.

Definition 2.3 (see for instance Daníelsson 2011 [2])
Risk measurement is a number that captures risk. It is obtained by applying data to a risk measure.
◊

Before introducing different risk measures, we introduce the desired properties, which a risk measure should satisfy. These represent a set of common sense rules. The failure to comply with these rules puts into question the suitability of the method for measuring risk.

Definition 2.4 *(Artzner et al [7]): A risk measure* $\rho : V \to R$ *is deemed to be* **coherent** *if it satisfies the following properties:*

Subadditivity*: for all* $Y_1, Y_2 \in V$

$$\rho(Y_1 + Y_2) \le \rho(Y_1) + \rho(Y_2)$$

Monotonicity*: for all* $Y_1, Y_2 \in V$ *where* $Y_1(\omega) \le Y_2(\omega)$ *for all* $\omega \in \Omega$

$$\rho(Y_1) \ge \rho(Y_2)$$

Positive Homogeneity*: for all* $Y \in V$ *and c>0*

$$\rho(cY) = c\rho(Y)$$

Translation invariance*: for all* $Y \in V$ *and* $c \in R$

$$\rho(Y + c) = \rho(Y) - c$$

◊

Explanation: **Subadditivity** requires that a merger should not create extra risk. When portfolios are combined together there may or may not be some diversification benefit. However, combining portfolios together should never be disadvantageous. **Monotonicity** states that if one portfolio is always worth more than another, it cannot be riskier. **Positive Homogeneity** requires that scaling a portfolio by a constant will change the risk by the same proportion. For example, if you double your portfolio then you double the risk. Also when changing the currency being used, the risk should be changed by the exchange rate. Positive Homogeneity can also be seen as a special case of Subadditivity: when combining two perfectly correlated subportfolios (no diversification effect) the total risk should be the sum of the risks of these subportfolios. Finally, **Translation Invariance** states that adding a risk free portfolio to an existing portfolio acts like an insurance. For example, when adding some non-variable riskless investment the result will be improved by this amount in all outcomes.

Choosing a coherent risk measure is very important, since any measure which does not satisfy some of the above mentioned axioms will produce paradoxical results of some kind giving a wrong assessment of relative risks. In chapter 3 methods of allocating risk will be introduced. These also have to satisfy a number of properties to be coherent. A crucial point to note is that an allocation method will not be coherent unless the risk measure chosen is coherent.

2.1 Risk measure alternatives

This section includes the general results from Daníelsson 2011 [2], my own examples and my own illustrations in Matlab and Mathematica.

Volatility

In classical Markowitz portfolio theory risk is measured by volatility (standard deviations). Standard Deviation measures the spread of the distribution, so it is based on the whole distribution rather than just on a part of it. However, volatility's ability to describe the shape of a distribution is limited, since it is possible to construct starkly different distributions which have the same SD, but would elicit strong variations in views regarding perceived "risk". SD is sufficient as a risk measure only when returns are normally distributed, since all statistical properties of the normal distribution is captured by mean and variance. The use of volatility as a risk measure can lead to misleading conclusions when applying to non-normal distributions of returns. However, the description of a distribution can be enhanced by looking at higher moments. It is also well-known that SD is the square root of variance, which is the 2nd central moment. The skewness and kurtosis (“peakidness”) can be measured by 3rd and 4th moments respectively. SD is not a coherent risk measure, since it violates monotonicity requirement as illustrated in the following very simple example.

Example 2.1.1

Consider two portfolios with the following sets of financial outcomes:

A: Equal likelihood of loss or profit of 10 (mean 0, SD $\sqrt{(-10)^2 \cdot 0.5 + 10^2 \cdot 0.5} = \sqrt{200} \approx 14.14$ *)*

B: Always a loss of 10 (mean -10, SD 0)

Even though portfolio A has either better or the same outcome (never worse), it is more risky from the perspective of SD. This is a violation of monotonicity.

▪

Although the significance of volatility is diminished if applied to non-standard distributions (heavy tailed, skewed, etc.), it is frequently used in finance.

Value-at-Risk (VaR)

The most common risk measure after volatility is Value-at-Risk. Even though VaR has some flaws (discussed below) it has remained the risk measure of choice in the financial industry. The reason for this is the best balance between theoretical properties and ease of implementation and backtesting. VaR is a point measure, which only focuses on a single point in the left tail of the distribution. It calculates the maximum loss expected on an investment, over a given time period and given a specified degree of confidence denoted by 1-α .

Definition 2.1.1 Value-at-Risk (VaR)

The loss on a trading portfolio such that there is a probability α of losses equaling or exceeding VaR(α) in a given trading period and a (1- α) probability of losses being lower than the VaR(α).

Let:

Q be the profit on an invested portfolio: $Q = P_t - P_{t-1}$

$f_q(x)$ *be the probability density function for the profit on the portfolio (PDF)*

$F_q(x)$ *be the cumulative density function for the profit of the portfolio (CDF)*

Continuous VaR:

VaR(α) is given by:

$$\Pr(Q \leq -VaR(\alpha)) = \alpha = \int_{-\infty}^{-Var(\alpha)} f_q(x)dx = F_q(-VaR(\alpha))$$

Discrete VaR:

VaR(α) is the ***smallest*** *value such that the probability that your losses exceed - VaR(α) is less than α :* $\Pr(Q < -VaR(\alpha)) < \alpha$

Exception: If there is a level, VaR(α) where $\Pr(Q \leq -VaR(\alpha)) = \alpha$ *then this inequality defines VaR.*

◊

Explanation: $VaR(\alpha)$ is a percentile on the distribution of profit, i.e. the value such that probability that your losses equal or exceed -$VaR(\alpha)$ is α . The losses will exceed the VaR level in α fraction of periods. Note that we use a minus sign because VaR is defined as a loss. Hence when VaR is a negative number the worst case scenario is actually a gain.

Example 2.1, cont.

We can now calculate VaR(0.05) and VaR(0.01) of our portfolio.

Parameters:

- *Time horizon: 1 day*
- *Degree of confidence: first α =0.05 and then α =0.01*

Historical distribution:

We simply find the 5% and 1% percentiles in our return vector. These numbers multiplied by our invested amount are VaR(5%) and VaR(1%) respectively. This can be calculated in Matlab:

```
VaR5=-InvAmount*quantile(ret,0.05)
VaR1=-InvAmount*quantile(ret,0.01)
```

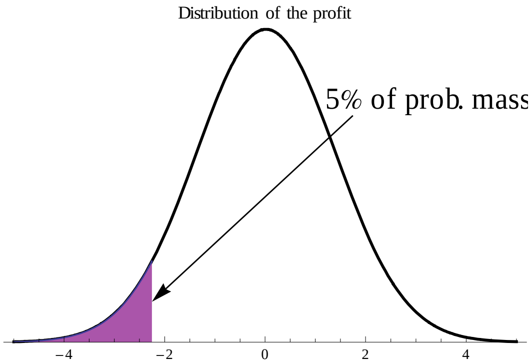
Results: VaR(5%) = 2.1512 and VaR(1%)=4.1536.

Explanation: If we invest 100$ today, we do not expect our portfolio to lose more than 2.1512$ (we do not expect our portfolio value tomorrow to be below 100-2.1512=97.8488$), since this only happens in 5% of *the outcomes. Hence VaR(5%) is the maximal possible loss at 5% level. As expected, VaR(1%) is higher than VaR(5%) since the degree of confidence level has increased.*

Normal distribution assumption:

We previously found that under this assumption the distribution of returns is normally distributed with the mean 0.000177 and the SD is 0.0138. VaR can be calculated by using the inverse normal function:

```
29 -    m=mean(ret)
30 -    SD=std(ret)
           ...

49 -    VaR5n=-InvAmount*norminv(0.05,m,SD)
50 -    VaR1n=-InvAmount*norminv(0.01,m,SD)
```

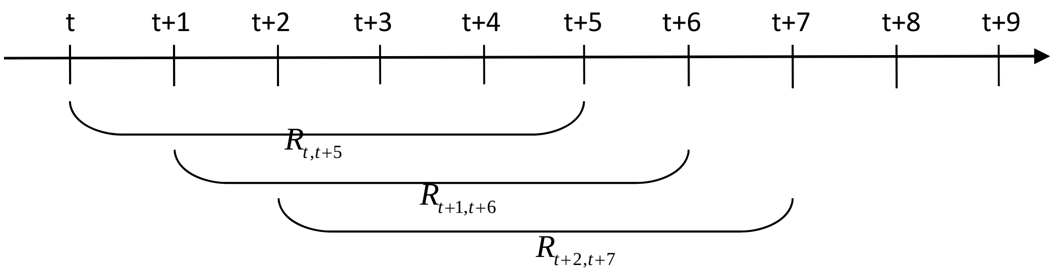


Results: VaR(5%) = 2.2533 and VaR(1%)=3.1942.

As you can see, both methods more or less agree on VaR(5%), however VaR(1%) is quite different. As previously stated, the historical distribution has much fatter tails and larger kurtosis, so when we focus on more extreme events, the normal distribution underestimates the risk (loss of 3.1942 vs. 4.1536). However, when focusing on less extreme events, normal distribution will overestimate the risk. For example historical VaR(10%) is 1.4088 , while normal VaR (10%) is 1.7517.

▪

What should one do if the time horizon was longer than 1 day, say 5 days? One way is to download weekly data instead of daily data. Another way is to calculate the historical 5-day returns out of historical data:



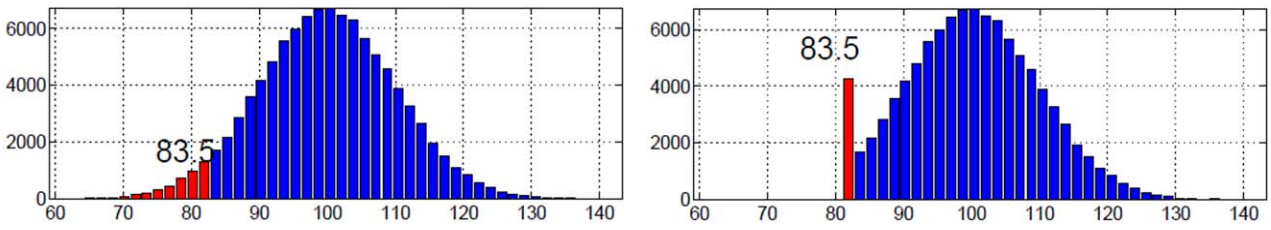
When using geometric returns and assuming normality, one can easily move between the time periods. When the holding period increases, the mean grows at rate T while volatility grows at rate $\sqrt{T}$. When time horizons are small, the mean does not play such an important role and price movements are mainly caused by volatility, so in practice mean is often ignored when time horizon is small. Hence, when using geometric returns, ignoring the mean and assuming normality and independence of returns over time, VaR can easily be scaled. For example, the 5-day VaR would be: $Var^{5-day}(5\%) = \sqrt{5}Var^{1-day}(5\%)$.

A common misunderstanding among many commentators is that VaR implies normality of returns. As demonstrated in previous example, this is not true: we can use any distribution provided the mean is

defined. Also note that under normality assumption VaR provides the same information as SD, since in that case VaR is simply a constant multiplied by SD.
VaR has several advantages. It is relatively easy to calculate and to explain. One does not need to know or be able to estimate more than a single point of the probability distribution. Most people understand VaR or at least think they do. On the other hand, because VaR is only a percentile, it has some critical drawbacks. Firstly, it completely ignores the risk in the left tail. For example, VaR(5%) states that for 95 days out of 100 downward price movements are expected to be less than the VaR and for 5 days out of 100 are expected to exceed VaR. As a consequence, VaR(5%) is incapable of capturing the risk of extreme movements that have a probability of less than 5%. Next, consider the following example:

Example 2.1.2

Assume that the value of equity in 1 month is normally distributed with mean 100 and SD 10.
Now consider two portfolios: a straight equity position (portfolio A) and the same equity position, but hedged with a put option which puts a floor on the losses (portfolio B). Assume that the floor is below $VaR(\alpha)$ *percentile cutoff. If we ignore the cost of the option, VaR will be the same in both cases, while the risk is very different.*



▪

Next, VaR is not subadditive, as demonstrated in the following example:

Example 2.1.3

Consider two similar uncorrelated portfolios:

Payoff	Prob.
+10	96%
-100	4%

VaR(5%) of each portfolio is equal to -10 (negative VaR is a gain). When combining these two portfolios together, we get the following payoffs in different states:

Payoff, A	Payoff, B	Total payoff:	Prob	Cum. Prob
-100	-100	-200	0.16%	0.16%
10	-100	-90	3.84%	4.00%
-100	10	-90	3.84%	7.84%
10	10	20	92.16%	100.00%

VaR of a combined portfolio is equal to 90 (loss).

▪

The example illustrates that from VaR's point of view, the combined portfolio is more risky than the sum of the individual risks of the portfolios, which is a violation of the Subadditivity, i.e. VaR is not a coherent risk measure. This is a serious theoretical problem. If the capital requirement of each subunit is dimensioned on its own risk, the regulator should be confident that also the overall company capital should be an adequate one. Only with subadditivity one can measure the benefits of diversification of adding new subportfolios or business lines. The question is whether subadditivity is also a practical problem, or just a theoretical problem with violation in only specific cases. We will answer this question in chapter 6. Note that VaR is subadditive under the normal distribution because then VaR is proportional to volatility.

Expected Shortfall (ES)[1]

To overcome the problem of lack of subadditivity in the VaR a family of alternative risk measures has been proposed. Instead of looking at a single point in the distribution, one could summarize the entire left tail as a single number by asking a question: "What is the expected result given the result is beyond a given threshold?". Expected Shortfall is one of these risk measures where the "given threshold" is VaR.

Definition 2.1.2 Expected Shortfall

The expected loss given that VaR loss is exceeded.

$$ES = -E(Q \mid Q \le -VaR(\alpha))$$

For continuous distributions:

$$ES = -E(Q \mid Q \le -VaR(\alpha)) = \frac{1}{\alpha} \int_{-\infty}^{-VaR(\alpha)} x f_q(x) dx$$

◊

Explanation*:* Given the significance level 1-α , ES results the average loss in α % of the worst cases (while VaR shows the best outcome among them). In literature there are different definitions of Expected Shortfall, however, the general idea is still the same.

Because Expected Shortfall is based on the whole left tail rather than just a single point, it gives a much richer description of risk than VaR. Additionally, in contrast to VaR, ES is coherent for continuous distributions as demonstrated by Artzner et al. 1999 [7]. However, ES can violate subadditivity in very specific cases with discrete distributions. Hence lack of subadditivity of ES is not a practical, but only theoretical problem. This will also be illustrated in chapter 6.

[1] Other names: Conditional Value-at-Risk (CVaR), Average Value-at-Risk (AVaR), Tail Value-at-Risk (TVaR), Expected Tail Loss (ETL).

Example 2.1, cont.

When using the historical distribution, we can get the expected loss given that VaR is exceeded by taking the mean of the observed returns which are below or equal to the VaR level. Then we multiply this mean return given that VaR is exceeded by the invested amount and flip sign (loss):

```
ES5=-InvAmount*mean(ret(ret<=quantile(ret,0.05)))
ES1=-InvAmount*mean(ret(ret<=quantile(ret,0.01)))
```

Results: ES(5%)=3.3440 and ES(1%)=5.6222.

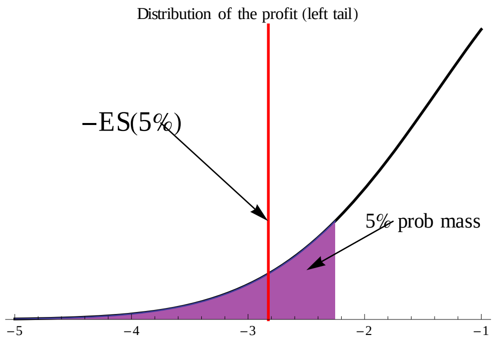
When assuming normality of returns, we can directly apply the formula from Definition 3.

$$ES(5\%) = -\frac{1}{0.05}\int_{-\infty}^{-2.2533} x f_q(x)dx \qquad \textit{and} \qquad ES(1\%) = -\frac{1}{0.01}\int_{-\infty}^{-3.1942} x f_q(x)dx$$

where $f_q(x)$ *is a normal distribution with mean 0.0177 and SD 1.38.*

This can be calculated in Mathematica:

```
In[35]:= InvAmount = 100;

         -InvAmount *
          NIntegrate[PDF[NormalDistribution[0.000177, 0.0138], x] * x,
            {x, -Infinity, InverseCDF[NormalDistribution[0.000177, 0.0138], 0.05]}] / 0.05

Out[34]= 2.82884

         -InvAmount *
          NIntegrate[PDF[NormalDistribution[0.000177, 0.0138], x] * x,
            {x, -Infinity, InverseCDF[NormalDistribution[0.000177, 0.0138], 0.01]}] / 0.01

Out[36]= 3.6603
```



Results: ES(5%)=2.8288 and ES(1%)=3.6603

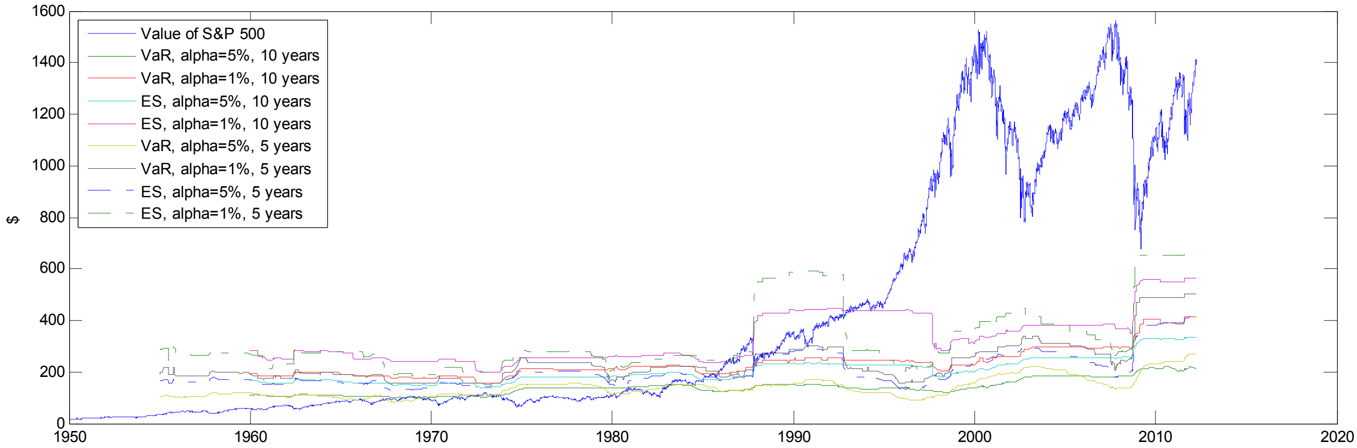
Note that, when assuming normality, the risk is underestimated due to the fatter tails in the historical distribution.

▪

ES is a coherent risk measure that shares many advantages with VaR. Any institution having a VaR based risk management system could implement ES without much additional effort. However, in practice the vast majority of financial institutions employ VaR instead of ES. The reasons could be:

- ES has a higher degree of uncertainty than VaR. Calculation of ES consists of two steps: ascertaining VaR and obtaining expectations of tail observations. This means that there are at least two sources of error.
- ES is much harder to backtest, since it requires estimates of the tail expectation to compare with ES forecast. Hence, in backtesting, ES can only be compared with the output from a model while VaR can be compared with actual observations. For example when backtesting 1-day VaR(5%) one can simply count how many times VaR level was exceeded. If the model works properly, VaR level should be exceeded in 5% of observations.
- “The expected result every x years” can be difficult to explain to public.

The difference between the risk measurements when using VaR and ES is illustrated below. Assume that in 1950 we invest 10000$ in S&P 500 and wait until today. Below historical values of daily Value-at-Risk and Expected Shortfall with different values of α and estimation windows are illustrated:



As expected, with a given estimation window and value of α ES risk measurement is higher than VaR risk measurement. Also, as α decreases, risk measurement increases (higher degree of confidence).

An obvious question to ask is which values of α and estimation windows should be used in practice. There is no final answer to this question and one will have to decide on the applied method in every single situation, considering which method fits best the actual setting. In general one shouldn't calculate VaR or ES with an extremely large degree of confidence (small α). The reason for that is the small number of such extreme events in the history; hence there is not enough data to estimate the risk properly, so the result will be imprecise. While VaR completely ignores these very rare observations in the left tail, ES is based on these observations. Regarding the length of the estimation window there is neither a final answer. On one hand, longer estimation window will include more extreme events. On the other hand, if the window is too long, the risk would be based on outdated data and the world might have changed significantly from that time. One of the main extensions of the simple VaR and ES models are VaR and ES with changing volatility. When looking at the historical returns, for example on page 7, we see that there are volatility clusters. So, in periods with low volatility large either positive or negative returns are unlikely and vice versa. Models with changing volatility put more weight on the recent data. These models among others include Moving Average Models, ARCH models and GARCH models. Even though these models significantly improve the risk estimates, they are not included in this thesis. The reason is that our main focus lies on the allocation of the risk once it is estimated.

2.2 Risk measures for more than one asset

When combining different assets into a portfolio, one has to take into the consideration the dependence structure between different assets. Understanding the dependence between different assets can be a difficult task, since there can be any kind of dependence structures. Correlation plays a crucial role in financial theory, even though it has significant shortcomings. Firstly, correlation is only defined for random variables with finite variances only. This property can cause problems when we work with heavy tailed distributions. Secondly, correlation only measures linear dependencies, so if two random variables X and Y are independent, then $Corr(X,Y)=0$. However the converse is false, since X and Y can have other dependence structures than linear. Also $Corr(X,Y) \approx 0$ does not necessary mean weak dependence between X and Y. Finally, correlation is not invariant under nonlinear strictly increasing transformations, i.e. $Corr(t_1(X), t_2(Y)) \neq Corr(X,Y)$ for t_1, t_2 strictly increasing functions.

As in case with one asset, we can either use the historical distribution or we can assume a specific distribution when measuring the risk of the portfolio.

Historical distribution

As previously stated, arithmetic returns have a very convenient advantage: the return of a portfolio is simply the weighted sum of the returns of individual assets. To apply a risk measure to a portfolio of several assets, one has to calculate the returns of the portfolio in all historical states: $R_t^{port} = \sum_{k=1}^{K} R_t^k w^k$ where R_t^k is the return of asset k at time t, w^k is the fraction of the wealth invested in asset k, i.e. $\sum w^k = 1$

Example 2.2.1

*Assume that on a specific day the return of S&P500 was equal to 1,2% and on the same day the return of Dow Jones Industrial Average Index was equal to -0.5%. Assume also that 25% of the invested amount was invested in S&P500 while the rest was invested in Dow Jones. The return of the portfolio on that specific day was 0,25*1,2%+(1-0,25)*(-0.5)=-0,075%.*

▪

Once we have the historical returns of the portfolio, the risk of the portfolio can be measured exactly as in the case with one asset. The main advantage of this method is that one does not have to take the dependence structure into consideration, since it is automatically built in the historical returns of the portfolio. On the other hand, the historical distribution may not have enough extreme events to predict the possible extreme events in future. For example, during the extreme events correlations are observed to rise in tails.

Specific distribution assumptions

When not using the historical distribution, one has to model dependence, so if there is an advanced deep dependence structure, it may not be captured by the assumptions.

When assuming normality of the returns, the returns of the portfolio will be a weighted sum of normally distributed variables, hence also normally distributed.

Let:

- $\mu = \{\mu_1, \mu_2, ..., \mu_K\}^T$ be the vector of the expected returns of individual assets
- Σ be the Covariance matrix of the returns
- $w = \{w_1, w_2, ..., w_K\}^T$ be the vector of weights, where elements sum up to 1.

From classical Markowitz theory we know that the returns of the portfolio at a specific time horizon will be normally distributed with mean $w \cdot \mu^T$ and variance $w^T \Sigma w$.

When assuming other distributions than normal, the task can get much more complicated. Often one will have to use transformations and Monte Carlo simulations.

3. Risk Allocation from the perspective of Game Theory

In the previous chapter we investigated different risk measures, which transform risk into a single number. Now recall our institution with several subunits. Once the risk measure is specified, each subunit can measure the risk of its own portfolio. We call the individual risk of a subunit for **stand-alone risk.** However, when several divisions create a coalition by combining their subportfolios into a single grandportfolio there might be a diversification benefit, i.e. the risk of the coalition might be smaller than the sum of the stand-alone risks in the coalition. Recall that if the risk measure is coherent, the risk of the grandportfolio cannot be riskier than the sum of the risks of the subportfolios (subadditivity requirement). Hence, in rare cases when risks of subportoflios are completely independent, there is no benefit, however in most circumstances there will be a diversification benefit because of the correlations among the subportfolios. The question is how this benefit should be shared among the subunits. The situation can be seen from the perspective of Game Theory, which is the study of situations where players adopt various strategies to best attain their individual goals. More precisely, we will use the results from Cooperative Game Theory where groups of players may enforce cooperative behaviour; hence the game is a competition between coalitions of players, rather than between individual players. In our case each subunit is a player who has its stand-alone risk. Each subunit has to put reserves aside which will be used in extreme bad events to save the company from default. The more risky is your portfolio, the more reserves you have to put aside. These reserves cannot be invested, so each subunit gets much lower profits on its investment. Hence, each subunit wants to get allocated as small share of the total risk capital as possible. The situation is in a way similar to cost sharing problems, where the goal of each player is to minimize its cost and the strategies consist of accepting or not to take part in different coalitions, including the coalition of all players. The difference is that while general cost sharing problems can have almost any structure, risk allocation problems cannot, because the risks of different coalitions are calculated by applying a certain risk measure to either historical data or a specific distribution. That is why, in this section, we first review the important results from Cooperative Game Theory and Cost Sharing theory. Then we translate these results to risk allocation problems and seek to understand the link between the chosen risk measure and the corresponding risk allocation problem.

The first section focuses on a situation where players are atomic, i.e. indivisible. By assuming atomic players, each player can either be a part of the coalition as a whole or not be in the coalition at all. In the second section the theory is expanded to the case with fractional players, where they become continuous and have a "scalable" presence. In our example each division has an amount of money invested in its own portfolio. By assuming atomic players we assume that each division can either be completely out of the coalition or completely in the coalition with its portfolio and its fixed amount of money invested in this portfolio. On the other hand, by assuming fractional players we allow scaling each division's amount of money invested in its portfolio when taking part in the coalition. For example, we could say that a coalition could consist of 60% of portfolio A and 40% of portfolio B.

3.1 Allocation to atomic players

This section will focus on allocation rules where players are atomic (indivisible). First we review the important results from the theory of Cost Sharing (taken from [1] and [4]). Then we will translate it to the risk allocation problems.

Definition 3.1.1

An **allocation problem**[2] or a **coalitional game** *(N,c) consists of:*

- a finite set N of n players with $|N| = n$

- a cost function c: $2^N \to R$ *is defined on all subsets of N satisfying* $c(\{\varnothing\}) = 0$ *. It associates a real number c(S) to each subset S of N (called a coalition).*

◊

The allocation problem is to allocate the cost of the grandcoalition N between the players. In our case, we will use the theory to allocate the amount of risk $\rho(X)$ between the subunits of the institution. We denote the set of possible coalitions by ζ . Since there are 2^n possible subsets of N, there are 2^n possible coalitions (or $2^n - 1$ possible coalitions if disregarding the empty coalition Ø). Only coalitions that benefit every member have a potential of being formed, i.e. where the members of the coalition are obligated to pay less than they would on their own. One of the desired properties of a cost sharing problem is subadditivity.

Definition 3.1.2

An allocation problem is deemed to be **subadditive** *if*

$c(S \cup T) \le c(S) + c(T)$ *for all subsets S and T of N with empty intersection.*

◊

Subadditivity is a desired property, because it requires that the costs connected with cooperation should always be smaller than adding up the costs of separate activities. An even stronger property is concavity.

Definition 3.1.3

An allocation problem is deemed to be **concave** *if*

$c(S \cup T) + c(S \cap T) \le c(S) + c(T)$ *for all subsets S and T of N where* $S \ne T$ *.*

◊

When an allocation problem is concave, players have even more incentive to form coalitions.

Note the subadditivity is a special case of concavity where $c(S \cap T) = \varnothing$. Hence, if an allocation rule is concave, then it is also subadditive. In general, if one has to confirm that the game is concave, one has to check $\frac{(2^n - 1)(2^n - 2)}{2}$ possibilities (own calculation). As n increases, the numbers of possibilities increases rapidly.

---

[2] General name: a **cost sharing problem** or **cost sharing game**

n	Number of coalitions	Number of cases to check for subadditivity[3]	Number of cases to check for concavity
2	3	1	3
3	7	6	21
4	15	25	105
5	31	90	465
6	63	301	1.953
7	127	966	8.001
8	255	3.025	32.385
9	511	9.330	130.305
10	1.023	28.501	522.753
11	2.047	86.526	2.094.081
12	4.095	261.625	8.382.465
13	8.191	788.970	33.542.145
14	16.383	2.375.101	134.193.153
15	32.767	7.141.686	536.821.761

For large values of n the problem becomes enormous, so subadditivity and concavity checking can be impossible, unless you have a computer program which can check all the possibilities, or the game it constructed in a way so that it is obvious to see that there is at least one violation. For example, if there is at least one coalition where cooperation is disadvantageous for any player or for any smaller coalition of players, then the game is obviously not subadditive and hence not concave. An example with 3 players where we check if the game is subadditive and concave can be found in Appendix 1.
Once the allocation problem is specified, the costs of the grandcoalition N can be shared among the players by using an allocation rule.

Definition 3.1.4

Let A be the set of risk capital allocation problems: pairs (N, ρ).

An **allocation rule** *is a function* $\Pi : A \to R^n$ *that maps each allocation problem (N, ρ) into a unique allocation:*

$$\Pi : (N,c) \to \begin{bmatrix} \Pi_1(N,c) \\ \Pi_2(N,c) \\ \cdot \\ \cdot \\ \cdot \\ \Pi_n(N,c) \end{bmatrix} = \begin{bmatrix} K_1 \\ K_2 \\ \cdot \\ \cdot \\ \cdot \\ K_n \end{bmatrix} \quad \textit{such that } \sum_{i \in N} K_i = c(N)$$

◊

Explanation: player *i* gets allocated a share K_i of the total costs.

[3] Simulation approach. See the code in Appendix 3.

In fact, we require that the costs allocated to different players sum up to the total cost of the grandcoalition N, i.e. $\sum_{i \in N} K_i = c(N)$. In Game Theory this requirement is called **Pareto-optimality**[4].

When deciding whether to take part in the coalition or not, each player compares its cost in the coalition to its stand-alone cost. In Game Theory a player would threaten to leave the coalition if he is allocated a share of the total cost that is higher that his own individual cost. Similar threats may come from coalitions of players, i.e. some players could threat to leave the coalition and create their own subcoalition if each player would get allocated a lower or the same cost in the new subcoalition. The set of allocations where no player has an incentive to leave the coalition is called the core.

Definition 3.1.5

Let $K \in R^n$ *be a cost allocation related to the allocation problem (N,c). Then*

$$core(N,c) = \{K \in R^n \left| \sum_{i \in N} K_i = c(N), \sum_{i \in S} K_i \le c(S) \forall S \in N \right.\}$$

◊

Explanation: No coalition would reduce its costs if it left the grand coalition N.

Note that core compatibility implies Pareto-efficiency, since it requires that all the risk of the main unit should be allocated. In general a core of an allocation problem is either empty (does not exist, i.e. no solution can satisfy all inequalities at the same time) or a compact and convex subset of R^n.

[4] Also called Efficiency in the literature

Example 3.1.1

Consider the following cost allocation problem with 4 players.

Cost allocation problem | Core conditions | Core conditions (an alternative formulation)

Coalition S	Cost c(S)
{1}	15
{2}	14
{3}	16
{4}	15
{1,2}	23
{1,3}	22
{1,4}	21
{2,3}	25
{2,4}	23
{3,4}	24
{1,2,3}	28
{1,2,4}	29
{1,3,4}	27
{2,3,4}	29
{1,2,3,4}	32

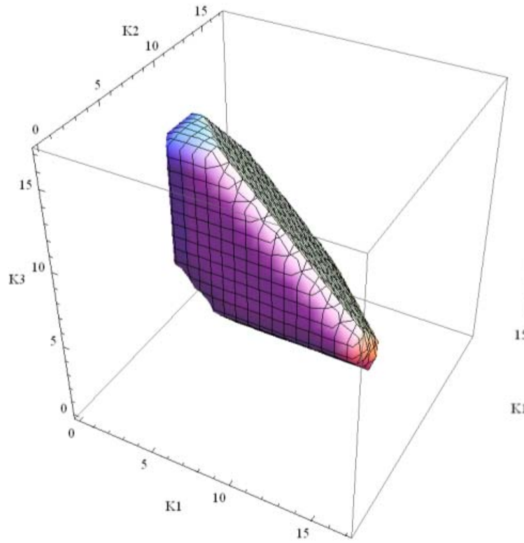
Core conditions:

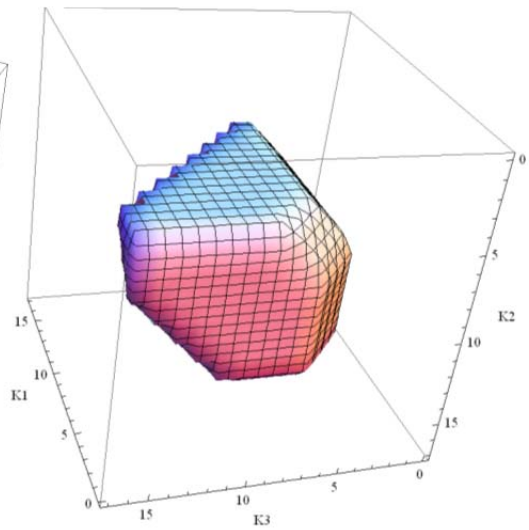
$$K_1 \le 15$$
$$K_2 \le 14$$
$$K_3 \le 16$$
$$K_4 \le 15$$
$$K_1 + K_2 \le 23$$
$$K_1 + K_3 \le 22$$
$$K_1 + K_4 \le 21$$
$$K_2 + K_3 \le 25$$
$$K_2 + K_4 \le 23$$
$$K_3 + K_4 \le 24$$
$$K_1 + K_2 + K_3 \le 28$$
$$K_1 + K_2 + K_4 \le 29$$
$$K_1 + K_3 + K_4 \le 27$$
$$K_2 + K_3 + K_4 \le 29$$
$$K_1 + K_2 + K_3 + K_4 = 32$$

Core conditions (an alternative formulation):

$$K_1 \le 15$$
$$K_2 \le 14$$
$$K_3 \le 16$$
$$32 - K_1 - K_2 - K_3 \le 15$$
$$K_1 + K_2 \le 23$$
$$K_1 + K_3 \le 22$$
$$32 - K_2 + K_3 \le 21$$
$$K_2 + K_3 \le 25$$
$$32 - K_1 + K_3 \le 23$$
$$32 - K_1 + K_2 \le 24$$
$$K_1 + K_2 + K_3 \le 28$$
$$32 - K_3 \le 29$$
$$32 - K_2 \le 27$$
$$32 - K_1 \le 29$$

The allocation problem is concave, as illustrated in the appendix 2, and hence also subadditive. Since the cost shares of the grandcoalition N has to sum up to 32, we can reformulate the core conditions by substituting $K_4 = 32 - K_1 - K_2 - K_3$ *into all inequalities, as done above to the right. The core of the game can be illustrated graphically in Mathematica (two different angles):*





▪

A non-empty core is an important property of an allocation problem, since it ensures a solution where all agents are willing to cooperate. We now consider the properties which ensure the non-empty core.
Let for each coalition $S \in \zeta$, a(S) be the membership vector in S, where $a_i(S) = 1$ if $i \in S$ and $a_i(S) = 1$ otherwise. Let $(\lambda^S)_{S\in\zeta} \in R_+$ denote the collection of numbers $\lambda^S \in R_+$ for all $S \in \zeta$.

Definition 3.1.6
A ***balanced collection of weights*** *is a collection of numbers* $(\lambda^S)_{S\in\zeta} \in R_+$ *such that* $\sum_{S\in\zeta} \lambda^S a(S) = a(N)$ *. A game is deemed to be* ***balanced*** *if* $\sum_{S\in\zeta} \lambda^S c(S) \geq c(N)$ *for all balanced collection of weights.*

◊

Theorem 3.1.1 (Bondareva 1963, Shapley 1967 [28])
A coalitional game has a non-empty core if and only if it is balanced.
◊

The intuition here is as follows. If each player can distribute one unit of its "working time" to different coalitions and if each coalition is active during a fraction λ^S of a unit of time, then the players cannot reach smaller costs than c(N), which is the costs of the grandcoalition. In this case everybody is willing to cooperate and form the grandcoalition N. In fact, a concave allocation problem is balanced, though the converse is not true in general. Hence the following theorem:

Theorem 3.1.2 (Shapley 1971 [29])
Let (N,c) be a concave cost allocation problem. Then the core is non-empty, i.e. $core(N,c) \neq \emptyset$
◊

Risk allocation problems

Since now we have summarized the necessary results of Cooperative Game Theory, we are ready to model risk capital allocation problems as coalitional games. We associate the subportfolios of an institution with players of the game and the risk measure ρ with the cost function c:

$$c(S) = \rho(\sum_{i\in S} X_i) \text{ for } S \subseteq N$$

Note that ρ can be any risk measures, and not only the ones illustrated in chapter 2. Thus the problem is to share the total risk capital of the firm, namely $K = \rho(X)$. Note that if the risk measure ρ is coherent and thus subadditive in the sense $\rho(Y_1 + Y_2) \leq \rho(Y_1) + \rho(Y_2)$ of Definition 2.4, the allocation problem with cost function c is subadditive in the sense $c(S \cup T) \leq c(S) + c(T)$ of Definition 3.1.2. Hence, if the risk measure used is coherent, the cooperation in the corresponding allocation problem is never disadvantageous. In fact, we can also show that the corresponding coalitional game is balanced:

Theorem 3.1.3 (Denault 2001 [14])
If a risk capital allocation problem is modelled as a coalitional game whose cost function c is defined with a coherent risk measure ρ*, then the core is non-empty.*

Proof [14]: *Let* $0 \le \lambda_S \le 1$ *for* $S \subseteq \zeta$ *and* $\sum_{S \in \zeta} \lambda^S a(S) = a(N)$. *Then*

$$\sum_{S\in\zeta} \lambda_S c(S) = \sum_{S\in\zeta} \rho\left(\sum_{i\in S} \lambda_S X_i \right) \ge \rho\left(\sum_{S\in\zeta}\left(\sum_{i\in S} \lambda_S X_i \right)\right) = \rho\left(\sum_{i\in N}\left(\sum_{i\in S, S\in\zeta} \lambda_S X_i \right)\right) = c(N)$$

By Theorem 3.1.1 the allocation problem has a non-empty core.
◊

This is a very important result, because it states that if we choose a coherent risk measure, we are sure that the total risk can be allocated in such a way that all subunit are satisfied by being a part of the company. Whether there are other conditions on the risk measures that ensure balancedness of the corresponding game is an open question.
In chapter 2 we investigated different risk measures. An important property of a risk measure is coherence, which is a set of desired axioms the failure to comply with which put into question the suitability of the method for measuring risk. As discussed earlier, there are also several desired properties an allocation rule should satisfy. Denault (2001) [14] defines coherence of an allocation rule by stating three axioms a coherent allocation rule should satisfy.

Definition 3.1.7 (Denault 2001 [14])
An allocation rule Π *is deemed to be* ***coherent*** *if for every allocation problem* (N, ρ)*, the allocation* Π (N, ρ) *satisfies the three properties:*

No undercut[5]

$$\forall M \subseteq N, \sum_{i\in M} K_i \le \rho(\sum_{i\in M} X_i)$$

Symmetry[6]
If by joining any subset $M \subseteq N \setminus \{i, j\}$*, portfolios i and j both make the same contribution to the risk capital, then* $K_i = K_j$.

Riskless allocation
If player i has a riskless portfolio then:

$$K_i = \rho(k) = -k$$

◊

No undercut states that the solution should be in the core of the game. Even though, in our case, a subunit cannot walk away from the company, we still use the same approach to share the risk fairly. If the allocation problem is not mapped to the elements of the core, there is at least one subunit which would

[5] Also called Core Compatibility in the literature
[6] Also called Equal Treatment Property in the literature

think that the solution is unfair and would like to block the coalition, even though it will be forced to cooperate. Because we seek to avoid such situations, core compatibility or no undercut is an important property of an allocation rule. **Symmetry** requires that subunits having the very same characteristics are treated the same way. It also ensures that the capital allocated to the elements should only depend on its contribution to the risk of the portfolio and nothing else**. Riskless allocation** ensures that a riskless portfolio gets allocated exactly its risk measure which will be negative. It also means that if a portfolio increases in cash position then its allocated capital should decrease by the same amount.

In addition to coherence we also want an allocation rule to satisfy two additional properties. The first one is strong monotonicity:

Definition 3.1.8 (see for instance Balog et al (2011) [11])
An allocation rule is satisfies ***Strong Monotonicity*** *if for an arbitrary subunit* $j \in N$ *for all* $S \subseteq N$ *the inequalities*

$$\rho(\sum_{i \in S \cup \{j\}} X_i) - \rho(\sum_{i \in S} X_i) \leq \tilde{\rho}(\sum_{i \in S \cup \{j\}} X_i) - \tilde{\rho}(\sum_{i \in S} X_i)$$

imply that the player j's share when using ρ *as a risk measure is not larger than when using* $\tilde{\rho}$ *as a risk measure.*
◊

Strong Monotonicity requires that if the contribution of risk of subunit to all the coalitions of subunits is not decreasing, then its allocated risk capital should not decrease either. Hence the players are not motivated to increase their contribution of costs (in our case risk). Example 3.1.2 on page 38 illustrates this point.

The second property, that more relate to the manipulation rather than the fairness of the allocation itself once the game is defined are the so-called "No advantageous splitting and merging". The name speaks for itself: no subunits should be able to manipulate the results in the favorable direction for themselves by either merging into one subunit or by splitting into several subunits. This property should be considered when there is a possibility of merging or splitting in the company. For example, a department in a company that deals with three departments, say, Canada, USA and Mexico, could suggest to merge into one large department "North America". The management should be aware of the fact that the incentive might be to get allocated less risk capital in total if an allocation rule used does not satisfy "No advantageous splitting and merging".
Of course, we would like to find an allocation rule which satisfies all of the desired properties. However, this is not possible, which we will demonstrate later in theorem 3.1.8.
The impossibility of the existence of an ideal allocation rule is a relevant practical problem. Different allocation rules will have their own advantages and disadvantages and one will have to decide on the applied method in every single situation, considering which method fits best the actual setting.

3.1.1 Allocation rule alternatives

This section introduces some of the most common risk allocation rules. Since these rules are going to be presented without the specification of the risk measure, they can be easily applied to different kinds of risk measures. We totally introduce seven different allocation rules: Activity, Incremental, Beta, Cost-Gap, Nucleolus, Shapley and Lorenz. The first two allocation rules are the obvious allocation rules, both to understand and to calculate. The Beta method is an allocation rule that is adopted from financial theory. Finally, the last four allocation rules are adopted from Game Theory. These are designed to share risk as fairly as possible, which makes them more complicated than the previous allocation rules. Because the definition of "fair" differs in every allocation rule, these allocation rules use very different approaches. We illustrate the allocation rules by allocating costs from example 3.1.1. This is done for illustration purposes only and the risks in an arbitrary risk allocation problem, obtained by applying a certain risk measure, can be allocated in exactly the same manner. Only the calculations of player 1's share will be illustrated in the examples.

Activity based method (Hamlen et al. 1977 [18])

The total risk is allocated to the subunits in proportion to their own risk.

$$K_i^{AB} = \frac{\rho(X_i)}{\sum_{j \in N} \rho(X_j)} \rho(X_N)$$

The significant advantage of this method is its simplicity. On the other hand, a very serious drawback of this method is that it does not consider the dependence structure between subunits, i.e. it does not reward those subunits that are hedging the others.

Example 3.1.1, cont.

For player 1 we have:

$$K_1^{AB} = \frac{15}{15+14+16+15} \cdot 32 = 8$$

The solution:

$$K^{AB} = (8; 7\frac{7}{15}; 8\frac{8}{15}; 8)$$

Incremental method (see for instance Jorion 2007 [21])

This method allocates the risk in proportion to every subunit's individual contribution of risk to the main unit.

$$K_i^{Inc} = \frac{\rho(X_N) - \rho(X_{N \setminus \{i\}})}{\sum_{j \in N} \rho(X_N) - \rho(X_{N \setminus \{j\}})} \rho(X_N)$$

Example 3.1.1, cont.

For player 1 we have:

$$K_1^{Inc} = \frac{32-29}{32*4-(29+27+29+28)} \cdot 32 = 6,4$$

The solution:

$$K^{Inc} = (6.4; 10\frac{2}{3}; 6.4; 8\frac{8}{15})$$

Beta method (see for instance Panjer 2002 [23])

Let $Cov(X_i, X_N)$ denote the covariance of the subunit *i* and the grandportfolio. Additionally let $\sigma^2(X)$ denote the variance of the total portfolio. It is well known that the beta of unit i is calculated as $\beta_i = \frac{Cov(X_i, X_N)}{\sigma^2(X)}$. The Beta method allocates:

$$K_i^B = \frac{\beta_i}{\sum_{j \in N} \beta_j} \rho(X_N)$$

Beta method is still simple and it takes some of the dependence structure into consideration. Note that Beta method is the only method where one only has to measure the risk of the grandcoalition N. Even stand alone risks are not necessary. On the other hand, one additionally has to calculate the Variance-Covariance matrix, which is not required by any other method. Note that K_i^B is not always defined. For example, when all the portfolios are riskless with the same return in all cases, all betas are 0 and the term $\frac{\beta_i}{\sum_{j \in N} \beta_j}$ is not defined. Because beta method is relatively obvious and the Variance-Covariance matrix is not provided, we will not apply the method on our example.

Cost gap method[7] (Driessen and Tijs 1986 [33])

For a given allocation problem $\rho(X_N) - \rho(X_{N \setminus \{i\}})$ denotes the marginal cost of agent *i* joining the main unit. This can be interpreted as the lower bound for agent *i*'s share. For any coalition $S \subseteq N$ let $g(S) = \rho(X_S) - \sum_{i \in N} (\rho(X_N) - \rho(X_{N \setminus \{i\}}))$ denote the cost gap related to S, i.e. the difference between the stand alone cost of S and the total lower bound of cost shares.

Define for each player:

$$\gamma_i = \min_{\emptyset \neq S \subseteq N, i \in S} \left| g(S) \right| = \min_{\emptyset \neq S \subseteq N, i \in S} \left| \rho(X_S) - \sum_{i \in N} (\rho(X_N) - \rho(X_{N \setminus \{i\}})) \right|$$

It can be argued that if $i \in S$, then agent i should never get allocated more than $\rho(X_N) - \rho(X_{N \setminus \{j\}}) + g(S)$. Hence a natural upper bound for agent i is $\rho(X_N) - \rho(X_{N \setminus \{j\}}) + \gamma_i$.

[7] Also called the τ value in the academic literature.

The cost gap method itself is defined as:

$$K_i^{CG} = \begin{cases} \rho(X_N) - \rho(X_{N\backslash\{i\}}) & \text{if } \sum_{j\in N}\gamma_j = 0 \\ \rho(X_N) - \rho(X_{N\backslash\{i\}}) + \dfrac{\gamma_i}{\sum_{j\in N}\gamma_j}\left(\rho(X_N) - \sum_{i\in N}(\rho(X_N) - \rho(X_{N\backslash\{i\}}))\right) & \text{otherwise} \end{cases}$$

The intuition of this method is as follows. If $\sum_{j\in N}\gamma_j$ =0 then $\gamma_i = 0 \forall i \in N$. In this rare case $\sum_{j\in N}\rho(X_N) - \rho(X_{N\backslash\{j\}})$ =$\rho(X_N)$. If this is not the case, then there is a remaining share that still needs to get allocated, namely $\rho(X_N) - \sum_{j\in N}\rho(X_N) - \rho(X_{N\backslash\{j\}}) > 0$. It gets shared in proportion with minimal cost-gaps of the players.

Example 3.1.1, cont.

To allocate the risk using Cost Gap method, we first have to find γ_i *for player i. For player 1 we have:*

Coalitions S	$\rho(S)$	$\sum_{j\in N}(\rho(X_N) - \rho(X_{N\backslash\{j\}}))$	$\rho(S)$ - $\sum_{j\in N}(\rho(X_N) - \rho(X_{N\backslash\{j\}}))$
{1}	15	3	12
{2}	14	5	9
{3}	16	3	13
{4}	15	4	11
{1,2}	23	8	15
{1,3}	22	6	16
{1,4}	21	7	14
{2,3}	25	8	17
{2,4}	23	9	14
{3,4}	24	7	17
{1,2,3}	28	11	17
{1,2,4}	29	12	17
{1,3,4}	27	10	17
{2,3,4}	29	12	17
{1,2,3,4}	32	15	17

The numbers in the last column which relate to the coalitions where player 1 is present are marked with orange. The smallest number among the orange numbers is 12, hence $\gamma_1 = 12$. In similar way we can find γ_i for all players:

$$\gamma = \begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \gamma_4 \end{pmatrix} = \begin{pmatrix} 12 \\ 9 \\ 13 \\ 11 \end{pmatrix}$$

Now, since $\sum_{j\in N}\gamma_j = 12+9+13+11 = 45 \neq 0$*, the allocated risk to player 1 is given by (2 decimals):*

$$32-29+\frac{12}{45}\left(32-(32\cdot 4-(28+29+27+29))\right)\approx 7.53$$

The solution:

$$K_i^{CG} = (7.53; 8.4; 7.91; 8.16)$$

Nucleolus method (Schmeidler 1969 [26])

Consider a given allocation problem (N,c) and let x be a specific allocation, i.e. $\sum_{i=1}^{n} x_i = c(N)$. Given (N,c) and x let for all $S \subset N$

$$e_S(x) = c(S) - \sum_{i\in S} x_i$$

be the gains of coalition S from a cooperative action given the particular allocation x. Note that $e_S(x)$ is positive for all S if and only if x lies in the core of the game. One type of fairness requirement is to favouring the worst off part, given the allocation x, i.e. to maximize the minimal gain. Since *e* indicates the gain from cooperation given x, we want to change the allocation x in such a way, that we maximize the value *e* for the worst off coalition S.

Denote by $e(x)\in R^{2^n-2}$ the vector of gains given the allocation x and let $\Theta : R^{2^n-2} \to R^{2^n-2}$ map vector elements in increasing order. Formally, the Nucleolus allocation is the allocation which lexicographically maximizes $\Theta(e(x))$:

$$K^{Nuc} = \{x \in I(N,c) \,|\, \Theta(e(x)) \succ_{lex} \Theta(e(y)), \forall y \in I(N,c)\}$$

where $I(N,c) = \{x \in R^n \,|\, \sum_{i=1}^{n} x_i = c(N)\}$ and $\succ_{lex}$ is the lexicographic ordering[8]

Computation of the Nucleolus can be done by solving a series of LP-problems. First solve:

$$\begin{aligned} &\text{Max} && \varepsilon \\ &\text{S.t.} && e_S(x) \geq \varepsilon, S \in N \\ & && x \in I(N,c) \end{aligned}$$

Let ε_1 be the optimal solution and let B_1 be the set of coalitions S for which the coalitional gain is ε_1 for all optimal solutions (x, ε_1). If the optimal solution is unique, it is the Nucleolus. If there are multiple solutions, solve the following problem:

[8] $x \succ_{lex} y$ if $x_1 > y_1$ or ($x_1 = y_1$ and $x_2 > y_2$) or ($x_1 = y_1$ and $x_2 = y_2$ and $x_3 > y_3$), etc.

$$\begin{array}{ll} \text{Max} & \varepsilon \\ & e_S(x) \geq \varepsilon, S \in N \\ \text{S.t.} & e_S(x) = \varepsilon_1 \forall S \in B_1 \\ & x \in I(N,c) \end{array}$$

Let ε_2 be the optimal solution and let B_2 be the set of coalitions S for which the coalitional gain is ε_2 for all optimal solutions (x, ε_2). Continue to the next LP problem as long as there are multiple optimal solutions. The next LP problem is defined by adding to the k'th problem the following restrictions:

$$e_S(x) = \varepsilon_k \forall S \in B_k \text{ where } B_k = \{S \mid e_S(x) = \varepsilon_k \text{ for all optimal solutions (x, } \varepsilon_k)\}.$$

A unique Nucleolus allocation is obtained after solving at most n-1 such LP problems. The main problem with Nucleolus is its practical application. When the number of players becomes large, the LP problems also become large and complicated to solve.

Example 3.1.1, cont.

The idea behind Nucleolus can be illustrated in Excel using the Solver function. We choose an arbitrary allocation, say (5,6,7,14). Given that specific allocation, we calculate the costs allocated to all possible coalitions and the corresponding gains of the coalitions, $e_S(x)$ *. This is illustrated below.*

	A	B	C	D	E	F	G	H	I	J	K	L
1	player 1	player 2	player 3	player 4	Sum			Min gain:				
2	5	6	7	14	32			1				
3												
4	Coalitions				c(S)		Costs allocated to each player				Sum	Gain of the coalition
5	1	0	0	0	15		5	0	0	0	5	10
6	0	1	0	0	14		0	6	0	0	6	8
7	0	0	1	0	16		0	0	7	0	7	9
8	0	0	0	1	15		0	0	0	14	14	1
9	1	1	0	0	23		5	6	0	0	11	12
10	1	0	1	0	22		5	0	7	0	12	10
11	1	0	0	1	21		5	0	0	14	19	2
12	0	1	1	0	25		0	6	7	0	13	12
13	0	1	0	1	23		0	6	0	14	20	3
14	0	0	1	1	24		0	0	7	14	21	3
15	1	1	1	0	28		5	6	7	0	18	10
16	1	1	0	1	29		5	6	0	14	25	4
17	1	0	1	1	27		5	0	7	14	26	1
18	0	1	1	1	29		0	6	7	14	27	2

In the orange cell we define the smallest gain of all coalitions (min. value in column L). We ask solver to maximize that minimal gain (orange cell) by changing the allocation (yellow cells) subject to c(N)=32, i.e. the yellow cells have to sum up to 32. The result:

	A	B	C	D	E	F	G	H	I	J	K	L
1	player 1	player 2	player 3	player 4	Sum			Min gain:				
2	7,25	9,25	7,25	8,25	32			4,25				
3												
4	Coalitions				c(S)		Costs allocated to each player				Sum	Gain of the coalition
5	1	0	0	0	15		7,25	0	0	0	7,25	7,75
6	0	1	0	0	14		0	9,25	0	0	9,25	4,75
7	0	0	1	0	16		0	0	7,25	0	7,25	8,75
8	0	0	0	1	15		0	0	0	8,25	8,25	6,75
9	1	1	0	0	23		7,25	9,25	0	0	16,5	6,5
10	1	0	1	0	22		7,25	0	7,25	0	14,5	7,5
11	1	0	0	1	21		7,25	0	0	8,25	15,5	5,5
12	0	1	1	0	25		0	9,25	7,25	0	16,5	8,5
13	0	1	0	1	23		0	9,25	0	8,25	17,5	5,5
14	0	0	1	1	24		0	0	7,25	8,25	15,5	8,5
15	1	1	1	0	28		7,25	9,25	7,25	0	23,75	4,25
16	1	1	0	1	29		7,25	9,25	0	8,25	24,75	4,25
17	1	0	1	1	27		7,25	0	7,25	8,25	22,75	4,25
18	0	1	1	1	29		0	9,25	7,25	8,25	24,75	4,25

As you can see, the maximal possible minimal gain is 4,25, which is achieved by all four 3-player coalitions. The minimal gain cannot be improved, and hence the Nucleolus of the game is:

$$K^{Nuc} = (7\frac{1}{4};9\frac{1}{4};7\frac{1}{4};8\frac{1}{4})$$

Shapley method (Shapley 1953 [27])

This method uses an entirely different approach and is well known in Game Theory. Instead of minimizing the maximum dissatisfaction as in case with Nucleolus, we could allocate an amount proportional to the loss each coalition derives from having a specific player as a member. The risk associated with any coalition of players S is given by $\rho(S)$ and consequently it is possible to determine the marginal cost of letting agent *i* join an arbitrary coalition S, where $i \notin S$. The Shapley method allocates the risk by taking a weighted average of these marginal costs caused by all possible coalitions:

$$K_i^{Sh} = \sum_{S \in \varsigma_i} \frac{(|S|-1)! \cdot (|N|-|S|)!}{|N|!} (\rho(X) - \rho(X_{S \setminus \{i\}}))$$

where $|S|$ denotes the number of elements in S and ς_i represents all coalitions of N that contain *i*.

Explanation: $\rho(X) - \rho(X_{S\setminus\{i\}})$ shows how the risk changes when player *i* joins the coalition S. The coalition S can be formed in (S-1)! different ways. After player *i* joins the coalition S, there are (N-S)! ways the grand coalition N can be formed:

$$\underbrace{(1)(2)...(|S|-2)(|S|-1)}_{|S|-1 \text{ arrive}} \qquad \underbrace{(i)}_{\text{i arrives}} \qquad \underbrace{(|N|-|S|)(|N|-|S|-1)...(2)(1)}_{\text{remaining players arrive}}$$

Hence there are $(|S|-1)! \cdot (|N|-|S|)!$ ways player i can join the grand coalition N, joining S first. Since the grand coalition N can be formed is N! different ways, the term $\frac{(|S|-1)! \cdot (|N|-|S|)!}{|N|!}$ can be seen as the probability that the grand coalition N is formed in a specific way. Hence, the Shapley value is the expected risk contribution of player *i*.

Example 3.1.1, cont.

We find all the possible coalitions where S is present and calculate the contribution of player 1 by checking how the cost changes when player 1 leaves the group:

$S \cup i$	c($S \cup i$)	S	c(S)	c($S \cup i$)-c(S)	$\frac{(\lvert S\rvert-1)!\cdot(\lvert N\rvert-\lvert S\rvert)!}{\lvert N\rvert!}$	column 5 * column 6
{1}	15	{Ø}	0	15	0,25	3,75
{1,2}	23	{2}	14	9	0,083333	0,75
{1,3}	22	{3}	16	6	0,083333	0,5
{1,4}	21	{4}	15	6	0,083333	0,5
{1,2,3}	28	{2,3}	25	3	0,083333	0,25
{1,2,4}	29	{2,4}	23	6	0,083333	0,5
{1,3,4}	27	{3,4}	24	3	0,083333	0,25
{1,2,3,4}	32	{2,3,4}	29	3	0,25	0,75
					Sum	**7,25**

Player 1 gets allocated a cost of 7,25. The costs allocated to other players can be calculated in the similar way. The result: $K_i^{Sh} = (7\frac{1}{4};8\frac{2}{3};8\frac{1}{6};7\frac{11}{12})$

▪

A significant advantage of this method is that it very deeply investigates the dependence structure and allocates the costs based on this information. However, Shapley method has some drawbacks. Firstly, it becomes fairly complicated when the number of players becomes large, since the calculation is based on all possible coalitions. Secondly, the Shapley value does not always satisfy Core Compatibility. However, there are situations when it does:

Theorem 3.1.4 (Shapley [29])

If a coalitional game (N,c) is concave, its core contains the Shapley value.

◊

Theorem 3.1.5 (Aubin [8])

If for all coalitions S, $\lvert S\rvert \geq 2$,

$$\sum_{T \subseteq S} (-1)^{\lvert S\rvert - \lvert T\rvert} c(T) \leq 0$$

then the core contains Shapley value.

◊

Lorenz method

Fairness can also be interpreted as distributional equality in the sense of Lorenz domination. Here we try to split the risk between the players as equally as possible, given that the solution has to be in the core of the game. We define the set of Lorenz maxima to be

$$L(N,c) = \{x \in core(N,c) \big| x \succ_{LD} y, \forall y \in core(N,c)\}.$$

Note that L(N,c) need not to be unique. From the definition it is obvious that the Lorenz allocation is either the equal split, if the equal split is in the core of the game, or a point on the edge of the core otherwise. If the game has no core, Lorenz allocation is not defined.

Theorem 3.1.6 (Dutta and Ray 1989 [16])

Uniqueness of $K^L = L(N,c)$ *is only guaranteed for concave allocation problems.*

◊

For concave allocation problems, the unique Lorenz allocation can be found by Dutta and Ray's algorithm. In general, one of the Lorenz solutions can be found by solving the following maximization problem:

Max $\quad z = u(x_1) + ... + u(x_n)$

S.t $\quad \sum_{i \in S} x_i \le c(S)$ for every $S \subset N$

where u is a strongly concave function.

Example 3.1.1, cont.

Since the equal split solution is in the core of the game, it is the Lorenz solution:

$K^L = (8;8;8;8)$

We could also solve the maximization problem from above with a strongly concave u, say $u(t) = -t^2$*. This would give the same solution, namely (8,8,8,8).*

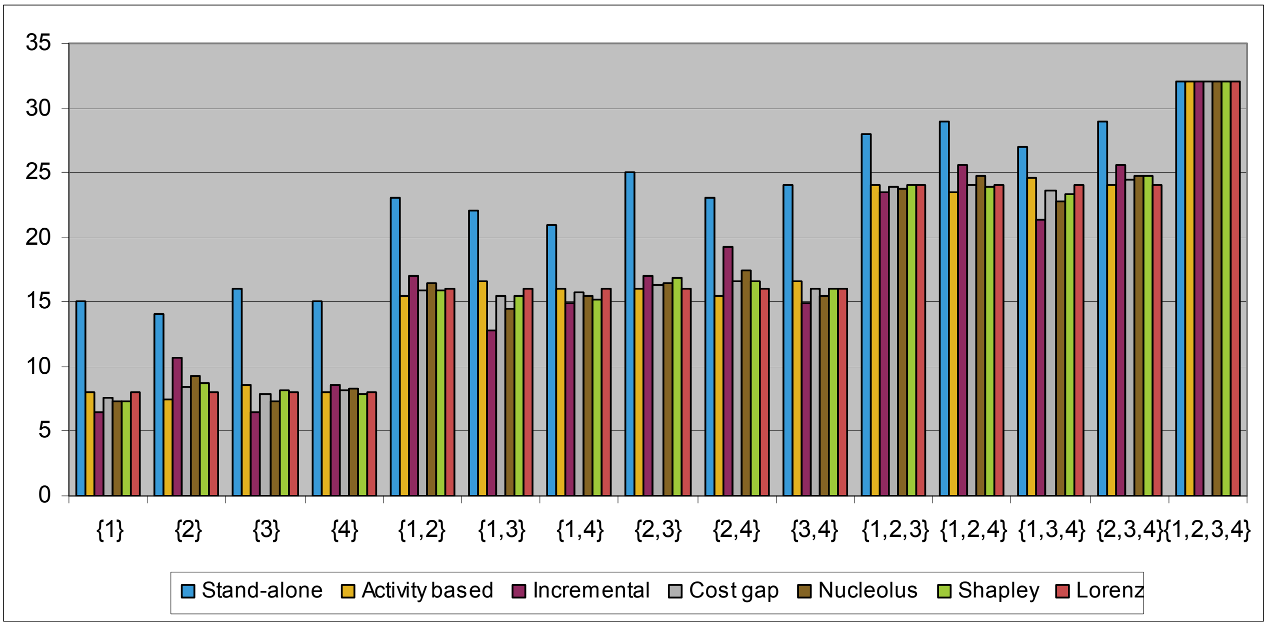
3.1.2 Choosing an allocation rule

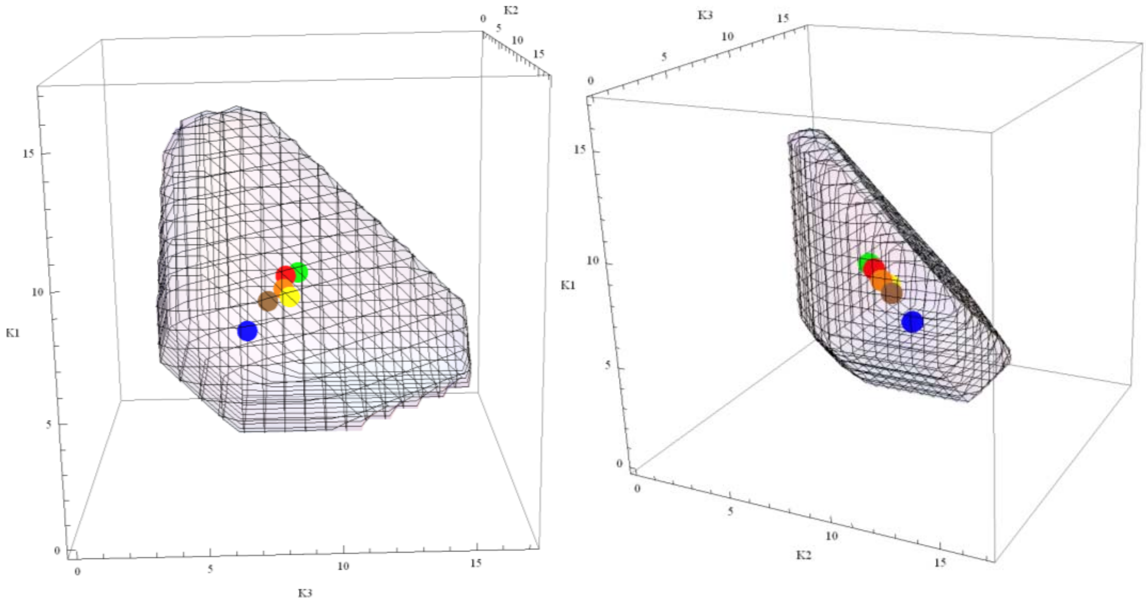
Example 3.1.1, cont.

We have considered 6 different allocation rules, which gave us 6 possible solutions (2 decimals):

	Activity	**Incremental**	**Cost-gap**	**Nucleolus**	**Shapley**	**Lorenz**
Player 1	8	6.4	7.53	7.25	7.25	8
Player 2	7.47	10.67	8.4	9.25	8.67	8
Player 3	8.53	6.4	7.91	7.25	8.17	8
Player 4	8	8.53	8.16	8.25	7.92	8
Sum	**32**	**32**	**32**	**32**	**32**	**32**

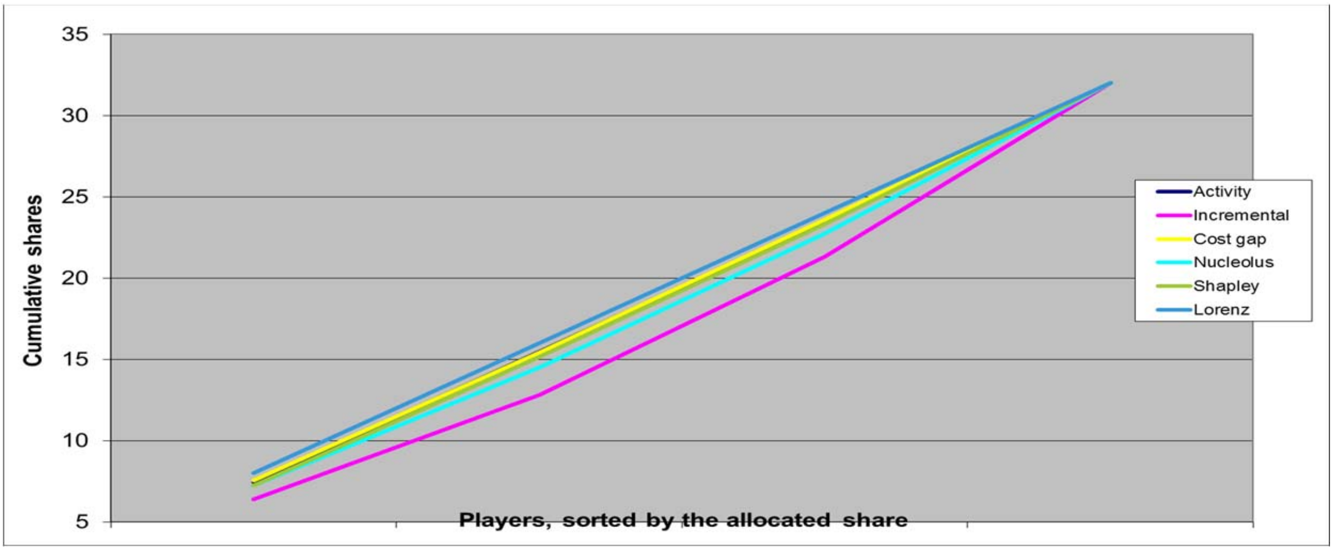
Even though not all allocation rules are core-compatible in general, they all are in this example as illustrated below:





Activity Incremental Cost-Gap Nucleolus Shapley Lorenz

Lorenz domination



Explanation: A straight line means that the risks are shared equally. The less distant a line is from the straight line, the closer is the allocation to the equal split. As expected, the Lorenz solution is the solution which divides costs most equally between the players.

The question is which allocation rules are better than the others. The answer to this question is not obvious and depends on the actual setting.

Properties of the different allocation rules are summarized in the following table:

	Coherence				
Method/Axiom	**No undercut**	**Symmetry**	**Riskless allocation**	**Strong monotonicity**	**No advantageous splitting and merging**
Activity based	No	Yes	No	No	Yes
Beta	No	No	No	No	Yes
Incremental	No	Yes	No	No	No
Cost gap	No	Yes	Yes	No	No
Nucleolus	Yes	Yes	Yes	No	No
Shapley	No	Yes	Yes	Yes	No
Lorenz	Yes	Yes	No	No	No

We see that Shapley and Nucleolus are the best allocation rules, as they satisfy three out of five desired properties.

The blue parts of the table above are results taken from Balog et al 2011 [11], where authors provide proofs and several examples. For instance they show that Incremental method does not satisfy Strong Monotonicity by the following example (here we also calculate the Shapley allocation):

Example 3.1.2

Consider the following situation with two players where the risk measure used is Expected Shortfall with $\alpha = 50\%$ *:*

	X_1	X_2	$X_1 + X_2$
$\omega_1 = 0,5$	-2	-9	-11
$\omega_2 = 0,5$	-9	0	-9
Risk measurement	**9**	**9**	**11**

Coalition S	Cost c(S)
{1}	9
{2}	9
{1,2}	11

The Incremental and Shapley both allocate 11/2=5,5 to each player. Now assume that the situation changes in the following way:

	X_1	X_2	$X_1 + X_2$
$\omega_1 = 0,5$	-2	-7	-9
$\omega_2 = 0,5$	-9	0	-9
Risk measurement	**9**	**7**	**9**

Coalition S	Cost c(S)
{1}	9
{2}	7
{1,2}	9

The Incremental method allocates $\frac{9-7}{(9-9)+(9-7)}9 = 9$ *to player 1 and 0 to player 2. So, even though the expression* $\rho(X_1 + X_2) - \rho(X_2)$ *is smaller in the second situation, player 1 gets allocated more in the second situation. This is a violation of Strong Monotonicity. Note that the Shapley method, which is the only method that satisfies Strong Monotonicity, allocates 5,5 to player 1 and 3,5 to player 2. One could argue that Shapley allocation is fair: the extra benefit in the second case is created solely by player 2, so player 2 should get allocated less, while the share of player 1 should remain unchanged.*

▪

The rest of the table is filled in by me by using the following arguments. Example 5.4 shows that Activity, Incremental and Beta do not satisfy Riskless allocation (RA). In fact, in this case, Beta will always allocate 0 to the player with riskless portfolio, since the covariance between the riskless portfolio and the grandportfolio is 0 in this case. A well-known fact in game theory is that Shapley satisfies Dummy player property, which implies RA. Nucleolus and Lorenz also satisfy RA, since the solutions are always in core of the game, so the player with riskless portfolio will not be able to get allocated more than his stand-alone cost. At the same time he will not be able to get allocated less, as this would worsen the excess and equality of the allocation. When using Cost-Gap method a riskless portfolio would have $\gamma_i = 0$ and would get allocated the contribution to the main unit, which is exactly its stand-alone cost.

That Activity method satisfies No advantageous splitting and Merging is a well-known fact in game theory. See for instance [1]. That Beta method satisfies this property is clear from the definition: the covariance when merging or splitting the Invested Amount does not change the returns, so the covariance is still the same. Example 5.3 shows that the other methods do not satisfy this property.

Theorem 3.1.7 (Young 1985 [37])

Let ϑ be a solution on the class of totally balanced games. Then solution ϑ satisfies Pareto Optimality, Symmetry and Strong Monotonicity if and only if ϑ is the Shapley solution.

◊

Because the Shapley solution is not core-compatible, we can also conclude the following:

Theorem 3.1.8 (Young 1985 [37], Csóka and Pintér 2010 [13])[9]

Considering a general coherent risk measure there is no allocation rule meeting the conditions of No Undercut, Strong Monotonicity and Symmetry at the same time.

◊

The Lorenz allocation is always core-compatible, so No-undercut is satisfied by definition. According to theorem 3.1.8 Lorenz can thus not satisfy Strong Monotonicity at the same time. Symmetry is also satisfied, as two players with the same contributions would have the same conditions when solving the LP problem. The table is thus filled in completely.

Theorem 3.1.7 makes it clear that if the Shapley method satisfied No Undercut, it would an outstanding allocation rule. Recall that the Shapley method actually does result in core-compatible allocations when theorem 3.1.5 is satisfied or when the allocation problem is concave. Unfortunately, the primer is in no way implied by the coherence of the risk measure ρ. Additionally the practical value of the theorem is low, since given a specific allocation problem it might be a difficult task to check if the conditions of theorem 3.1.5 are satisfied. Whether there exist any conditions on the risk measure that ensure the conditions provided in theorem 3.1.5 is an open question. Unfortunately the latter, concavity, is usually not satisfied either. In fact, with a coherent risk measure the game is concave only when the risks of subunits are perfectly positively correlated, i.e. no diversification benefit.

Theorem 3.1.9

Let ρ be a coherent risk measure. If the corresponding risk allocation problem is concave, then the risks of the subunits are perfectly positively correlated.

***Proof**: Consider any three portfolios with random expected profits at a specific time in future denoted by X,Y and Z respectively. The concavity of the risk allocation problem implies*

$$\rho(X+Z)+\rho(Y+Z)\geq\rho(X+Y+Z)+\rho(Z)$$

but also

$$\begin{aligned}\rho(X+Z)+\rho(Y+Z)&=\rho(X+(Y+Z)-Y)+\rho(Y+Z)\\&\leq\rho(X+(Y+Z))+\rho((Y+Z)-Y)\\&=\rho(X+Y+Z)+\rho(Z)\end{aligned}$$

This implies that

$$\rho(X+Z)+\rho(Y+Z)=\rho(X+Y+Z)+\rho(Z)$$

[9] Csóka and Pintér (2010) [13] proved the theorem on the class of totally balanced games

By taking Z=0 we obtain the additivity of ρ :

$$\rho(X)+\rho(Y)=\rho(X+Y)$$

Given a coherent risk measure ρ this is only possible when risks of the players are perfectly positively correlated.

◊

In order to make the theorem above, I took one of Denault's 2001 [14] theorems with the corresponding proof and made an amendment. The following theorem is invented by me:

Theorem 3.1.10

Let ρ be a coherent risk measure. If the corresponding risk allocation problem is concave, then the core consists of one element only, namely the solution where all players get allocated their stand-alone costs.

Proof: *from theorem 3.1.9 we know that the risks of different subportfolios are perfectly positively correlated. Consider $\hat{n}$ independent subportfolios denoted by $PF_1, PF_2, ..., PF_{\hat{n}}$. Let $\hat{N}$ denote the set of all these subportfolios. Because all subportoflios are perfectly positively correlated, the following is true:*

$$\rho(PF_1+PF_2+...PF_{\hat{k}})=\rho(PF_1)+\rho(PF_1)+...+\rho(PF_{\hat{k}}) \qquad (*)$$

where $\hat{k}$ is an integer satisfying $2 \le \hat{k} \le \hat{n}$.

When translated to the risk allocation problem (*) implies:

$$c(\{1\}\cup\{2\}\cup...\cup\{\hat{k}\})=c(\{1\})+c(\{2\})+...+c(\{\hat{k}\})$$

Consider two coalitions $\tilde{S}$ and $\overline{S}$ that each consists of some of the portfolios $PF_1, PF_2, ..., PF_{\hat{n}}$ with an empty intersection, i.e. $\tilde{S}\cap\overline{S}=\emptyset$. Because coalitions of portfolios $\tilde{S}$ and $\overline{S}$ are also perfectly positively correlated, the following is true:

$$\rho(\sum_{i\in\tilde{S}}PF_i+\sum_{j\in\overline{S}}PF_j)=\rho(\sum_{i\in\tilde{S}}PF_i)+\rho(\sum_{j\in\overline{S}}PF_j) \qquad (**)$$

When translated to risk allocation games (**) implies:

$$c(\tilde{S}\cup\overline{S})=c(\tilde{S})+c(\overline{S})$$

Let $\hat{K}$ be a share that needs to be allocated, i.e. $c(\hat{N})=\hat{K}$. Let ϑ be a solution that satisfies Pareto Optimality and let $\hat{K}_i$ denote the share allocated to player i , i.e. $\sum_i \hat{K}_i=\hat{K}$.

(*) and (**) imply that

$$\hat{K}=\sum_i \hat{K}_i = c(\hat{N})=c(\{1\})+c(\{2\})+...+c(\{\hat{n}\})$$

and

$$\sum_{i,i\neq j}\hat{K}_i = c(N\setminus\{j\})=c(\{1\})+c(\{2\})+...+c(\{\hat{n}\})-c(\{j\})$$

Assume that player j gets allocated a share that is larger than his stand-alone costs, i.e. $\hat{K}_j > c(\{j\})$. Then the solution is obviously not core-compatible.

Instead assume that player j gets allocated a share that is smaller than his stand-alone costs, i.e. $\hat{K}_j < c(\{j\})$ *. This would imply that:*

$$c(\{1\}) + c(\{2\}) + \ldots + c(\{\hat{n}\}) - c(\{j\}) < \underbrace{c(\{1\}) + c(\{2\}) + \ldots + c(\{\hat{n}\})}_{=K} - \hat{K}_j$$

$$\Leftrightarrow \sum_{i, i \neq j} c(\{i\}) < \sum_{i, i \neq j} \hat{K}_i$$

That is, the core conditions are again violated.

Assume that player j gets allocated exactly his stand-alone costs, i.e. $\hat{K}_j = c(\{j\})$ *. Then:*

$$c(\{1\} \cup \{2\} \cup \ldots \cup \{\hat{k}\}) = c(\{1\}) + c(\{2\}) + \ldots + c(\{\hat{k}\}) = \hat{K}_1 + \hat{K}_2 + \ldots + \hat{K}_{\hat{k}}$$

The core conditions are now satisfied.

◊

Note that because the risk allocation problem is not concave unless the risks are perfectly positively correlated, the uniqueness of Lorenz allocation is not guaranteed.

The last theorems imply that, unless a risk allocation has a specific structure it is very difficult, if not impossible, to predict whether the Shapley method will result in a core-compatible solution. The question is whether there are other unknown conditions that imply core-compatibiliy of the Shapley method. We do not answer this question in this thesis. However, we note that knowing how often the Shapley method results in core-compatible allocations could be a good start. That is why in chapter 6 we present the results from a large simulation work, which shows the core-compatibility ratios of selected allocation rules.

3.2. Allocation to fractional players

In the previous section players were indivisible. This assumption is not a natural one, as we could consider fractions of portfolios as well as coalitions involving fractions of portfolios. This section examines a variant of the allocation problem where players have a scalable presence. In the academic literature such games are called “fuzzy games” or simply “games with fractional players”. The results in this section are taken from Aumann and Shapley 1974 [5] and Denault 2001 [14].

Let $\lambda \in R_+^n$ denote the “level of presence” of each of n players in a coalition. The vector λ can thus be used to represent a specific coalition of parts of players.

Definition 3.2.1

A coalitional game with fractional players (N, Λ, r) *consists of*

- *a finite set N of players with* $|N| = n$.
- *a positive vector* $\Lambda \in R_+^n$*, each element representing for one of the n players his full involvement or capacity.*
- *a real-valued cost function r:* $R_+^n \to R$*,* $\lambda \mapsto r(\lambda)$ *such that r(0)=0.*

◊

In our case the vector Λ represents the “size” of each portfolio, i.e. the amount of money that is invested in each portfolio. In case with allocation to different business lines Λ would represent the volumes of these business lines. The ratio $\frac{\lambda_i}{\Lambda_i}$ denotes the “presence level” for player *i*. Note that this is an extension of the case from the previous section, where each player could have either no or full presence. So, in case with atomic players λ_i can either be 0, if player *i* is not a member of the coalition, or Λ_i if player *i* is a member of the coalition (full presence level). When expanding the theory to fractional players, we let λ_i take other values as well.

The cost function r can be identified with a risk measure ρ through

$$r(\lambda) = \rho\left(\sum_{i \in N} \frac{\lambda_i}{\Lambda_i} X_i\right) \qquad \text{so that } r(\Lambda) = \rho(N)$$

By extension we also call $r(\lambda)$ a risk measure. The expression $\frac{X_i}{\Lambda_i}$ is the per-unit profit of portfolio *i* at a specific time in future. Since X_i denotes the profit at a specific time in future with an amount of money Λ_i invested in it, the expression $\frac{X_i}{\Lambda_i}$ simply denotes the return at the very same specific time in future. As stated on page 8, measuring the risk of return and the profit are two sides of the same coin.

The definition of coherent risk measure (Definition 2.4) is adapted as follows.

Definition 3.2.2 (Denault 2001 [14])
*A **risk measure** r is **coherent** if it satisfies the following axioms:*

***Subadditivity**: for all* λ^*, λ^{**} *in* R^n*,*

$$r(\lambda^* + \lambda^{**}) \le r(\lambda^*) + r(\lambda^{**})$$

***Monotonicity**: for all* λ^*, λ^{**} *in* R^n

$$\sum_{i \in N} \frac{\lambda_i^*}{\Lambda_i} X_i \le \sum_{i \in N} \frac{\lambda_i^{**}}{\Lambda_i} X_i \Rightarrow r(\lambda^*) \ge r(\lambda^{**})$$

***Degree one homogeneity**: for all* $\lambda \in R^n$ *and* $\gamma \in R_+$

$$r(\gamma\lambda) = \gamma r(\lambda)$$

***Translation invariance**[10]: for all* $\lambda \in R^n$ *, if portfolio k is riskless then*

$$r(\lambda) = r\left(\begin{bmatrix}\lambda_1 & \lambda_2 & . & . & . & \lambda_{k-1} & 0 & \lambda_{k+1} & . & . & . & \lambda_{n-1} & \lambda_n\end{bmatrix}^t\right) - \frac{\lambda_k}{\Lambda_k} X_k$$

◊

Note that r is coherent if and only if ρ is coherent. An allocation rule is called a fuzzy value when translated into the allocation problem with fractional players.

Definition 3.2.3 (Denault 2001 [14])
Let $k \in R^n$ *be the per unit allocation of risk capital to each portfolio. Thus the capital allocated to portfolio i is given by* $K_i = k_i \Lambda_i$.
A fuzzy value is a mapping assigning to each coalitional game with fractional player a unique per-unit allocation vector:

$$\phi : (N, \Lambda, r) \rightarrow \begin{bmatrix} \phi_1(N,\Lambda,r) \\ \phi_2(N,\Lambda,r) \\ . \\ . \\ . \\ \phi_n(N,\Lambda,r) \end{bmatrix} = \begin{bmatrix} k_1 \\ k_2 \\ . \\ . \\ . \\ k_n \end{bmatrix} \qquad \text{with} \qquad \Lambda^t k = r(\Lambda)$$

◊

[10] Denault assumes that the n'th subportfolio is riskless. I extended his definition of Translation Invariance, so that any subportfolio can be riskless.

Definition 3.2.4 (Denault 2001 [14])

Let r be a coherent risk measure. A fuzzy value

$$\phi : (N, \Lambda, r) \rightarrow k \in R^n$$

is coherent if it satisfies the following properties and if k is an element of the fuzzy core (defined below):

***Aggregation invariance**: Suppose the risk measures r and* $\overline{r}$ *satisfy* $r(\lambda) = \overline{r}(\Gamma\lambda)$ *for some mxn matrix* Γ *and all* λ *such that* $0 \leq \lambda \leq \Lambda$. *Then*

$$\phi(N, \Lambda, r) = \Gamma^t \phi(N, \Gamma\Lambda, r)$$

***Continuity**: The mapping* ϕ *is continuous over the normed vector space* M^n *of continuously differentiable functions r:* $R_+^n \rightarrow R$ *that vanish at the origin.*

***Non-negativity under r non-decreasing**: if r is non-decreasing in the sense that* $r(\lambda) \leq r(\lambda^*)$ *whenever* $0 \leq \lambda \leq \lambda^* \leq \Lambda$ *then*

$$\phi(N, \Lambda, r) \geq 0$$

***Dummy player allocation**: If i is a dummy player in the sense that* $r(\lambda) - r(\lambda^*) = (\lambda_i - \lambda_i^*)\dfrac{\rho(X_i)}{\Lambda_i}$ *whenever* $0 \leq \lambda \leq \Lambda$ *and* $\lambda^* = \lambda$ *except in the i'th element, then*

$$k_i = \frac{\rho(X_i)}{\Lambda_i}$$

***Fuzzy core**: The allocation* $\phi(N, \Lambda, r)$ *belongs to the fuzzy core of the game* (N, Λ, r) *if for all* λ *such that* $0 \leq \lambda \leq \Lambda$,

$$\lambda^t \phi(N, \Lambda, r) \leq r(\lambda) \qquad \text{as well as} \qquad \Lambda^t \phi(N, \Lambda, r) = r(\Lambda)$$

◊

The properties required of a coherent fuzzy value can be justified in the same manner as in case with atomic players. Aggregation invariance requires that equivalent risks should receive equivalent allocation, i.e. this property is akin to the symmetry property. Continuity ensures that similar risk measures yield similar allocations. Non-negativity under non-decreasing risk measures enforces to allocate more when there is more risk. The dummy player property is akin to the riskless allocation. Finally, the fuzzy core is the natural extension of the core: no players, neither any fuzzy coalitions of players has any incentives to split off from the grand coalition. It is thus obvious that the core is a part of the fuzzy core.

3.2.1 The Aumann-Shapley method[11]

The Shapley method was extended by Aumann and Shapley to games with fractional players in their book "Values of non-atomic games" [5]. Since the players have a scalable presence, there are no atoms, i.e. smallest entities that could be called players, hence the name "non-atomic games". In the book the Aumann-Shapley value is defined as:

$$\phi_i^{AS}(N,\Lambda,r) = k_i^{AS} = \int_0^1 \frac{\partial r}{\partial \lambda_i}(\overline{\gamma}\Lambda)d\overline{\gamma}$$

Explanation: The per-unit cost k_i^{AS} is an average of the marginal costs of portfolio *i* as the level of activity increases uniformly for all subportfolios from 0 to Λ.

When translated into risk allocation games when a coherent risk measure is used, the Aumann-Shapley value can be simplified. Consider the following results from standard calculus:

Definition 3.2.1.1. Homogeneity

A real n-variables function F is homogeneous of degree k if for all h>0 the following equation holds:

$$F(hx) = h^k F(x)$$

◊

Theorem 3.2.1.1 Euler's theorem

If F is a real n-variables homogeneous function of degree k, then

$$x_1\frac{\partial F(x)}{\partial x_1} + x_2\frac{\partial F(x)}{\partial x_2} + ... + x_n\frac{\partial F(x)}{\partial x_n} = kF(x)$$

◊

Theorem 3.2.1.2

If F is a homogeneous function of degree k then $\frac{\partial F(x)}{\partial x_i}$ *is homogeneous of degree (k-1).*

◊

Recall that both Expected Shortfall and Value-at-Risk satisfy Positive Homogeneity from definition 2.4 or Degree one Homogeneity from definition 3.2.2, which is also implied by any other coherent risk measure. In fact, the axiom Positive Homogeneity from Definition 2.4 simply means that the risk measure is homogeneous of degree 1. As a result the Aumann-Shapley value in case of Risk allocation problems with a risk measure that satisfies is homogeneous of degree 1 can be simplified to:

$$k_i^{AS} = \int_0^1 \frac{\partial r}{\partial \lambda_i}(\overline{\gamma}\Lambda)d\overline{\gamma} = \frac{\partial r(\Lambda)}{\partial \lambda_i}$$

Note that the per-unit allocation vector is the gradient of the mapping r evaluated at the full presence level:

$$k^{AS} = \nabla r(\Lambda)$$

[11] In finance this method is sometimes called "allocation by the gradient method" or "Euler allocation".

This gradient is called “Aumann-Shapley prices”. The amount of risk capital allocated to each player is given by:

$$K_i^{AS} = k_i^{AS} \Lambda_i$$

The intuition behind these results is demonstrated in the following example:

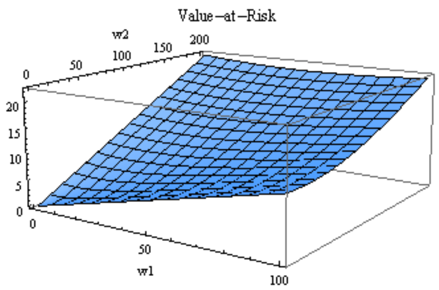
Example 3.2.1.1

Assume that a company has two independent portfolios whose returns at a specific time in future are normally distributed with mean 0.05, standard deviation 0.1 and mean 0.02, standard deviation 0.08 respectively. Since the portfolios are independent, the profit of the grand portfolio will also be normally distributed. This can be calculated in Mathematica:

```
In[230]:= a = NormalDistribution[0.05, 0.1];

In[231]:= b = NormalDistribution[0.02, 0.08];

In[232]:= m = w1 * Mean[a] + w2 * Mean[b]
Out[232]= 0.05 w1 + 0.02 w2

In[233]:= sd = Sqrt[w1^2 * StandardDeviation[a]^2 + w2^2 * StandardDeviation[b]^2]
Out[233]= √(0.01 w1^2 + 0.0064 w2^2)

In[234]:= portfolio = NormalDistribution[m, sd]
Out[234]= NormalDistribution[0.05 w1 + 0.02 w2, √(0.01 w1^2 + 0.0064 w2^2)]
```

Once we have the distribution of the profit of the grandportfolio at the specific time in future, we can apply a risk measure, such as Value-at-Risk. Assume that $\alpha = 5\%$.

```
VaR = -Quantile[portfolio, alpha]
Out[235]= -0.05 w1 - 0.02 w2 + 1.64485 √(0.01 w1^2 + 0.0064 w2^2)

In[238]:= Plot3D[VaR, {w1, 0, 100}, {w2, 0, 200}, AxesLabel → Automatic, PlotLabel → "Value-at-Risk"]

Out[238]=
```



The last graph plots VaR of the portfolio measured in $ as a function of the two amounts invested in the two portfolios. Now, assume that the amounts invested in the portfolios are 100 $ and 200 $ respectively. Hence $\Lambda = \begin{pmatrix} 100 \\ 200 \end{pmatrix}$ *and VaR is 22.035 (right upper point on the graph above). This amount of risk has to be shared between the two portfolios. To calculate* $k_i^{AS} = \int_0^1 \frac{\partial r}{\partial \lambda_i}(\overline{\gamma}\Lambda) d\overline{\gamma}$ *for i=1 and i=2 we only need the combinations* *of w1 and w2 that satisfy* $\begin{pmatrix} w1 \\ w2 \end{pmatrix} = \overline{\gamma}\Lambda = \begin{pmatrix} \overline{\gamma}100 \\ \overline{\gamma}200 \end{pmatrix}$ *for all* $0 \le \overline{\gamma} \le 1$, *i.e. the diagonal line on the plot above from (0,0) to (100,200). When we only use these combinations of w1 and w2, we get:*

```
In[247]:= w1 = gamma * 100;

In[248]:= w2 = gamma * 200;
          VaR = -Quantile[portfolio, alpha]

Out[249]= -9. gamma + 31.035 Sqrt[gamma^2]
```

That is, VaR is a ***linear*** *function of gamma (in our case gamma is not negative and the square root cancels out the power of two) and thus the first derivative is a constant. Hence the marginal contributions of each portfolio to the risk of the grandportfolio are the same for all* $0 \le \overline{\gamma} \le 1$ *and we can simply calculate the marginal contributions at one point, for example (100,200), which is the full capacity. We clear the values of w1,w2, recalculate the general expression of VaR, differentiate it with respect to w1 and later w2 and evaluate the result at w1=100, w2=200:*

```
In[282]:= VaR = -Quantile[portfolio, alpha]

Out[282]= -0.05 w1 - 0.02 w2 + 1.64485 Sqrt[0.01 w1^2 + 0.0064 w2^2]

In[283]:= D[VaR, w1]

Out[283]= -0.05 + 0.822427 (0. + 0.02 w1) / Sqrt[0.01 w1^2 + 0.0064 w2^2]

In[284]:= K1[w1_, w2_] := -0.05` + 0.8224268134757362` (0.` + 0.020000000000000004` w1) / Sqrt[0.010000000000000002` w1^2 + 0.0064` w2^2]

In[285]:= K1[100, 200] * 100

Out[285]= 3.71771

In[286]:= D[VaR, w2]

Out[286]= -0.02 + 0.822427 (0. + 0.0128 w2) / Sqrt[0.01 w1^2 + 0.0064 w2^2]

In[279]:= K2[w1_, w2_] := -0.02` + 0.8224268134757362` (0.` + 0.0128` w2) / Sqrt[0.010000000000000002` w1^2 + 0.0064` w2^2]

In[280]:= K2[100, 200] * 200

Out[280]= 18.3173
```

That is, player 1 gets allocated 3.72 $ and player 2 gets allocated 18.32 $ (two decimals). Note that both players are willing to cooperate, because the stand-alone risks are 11.45 and 22.32 respectively:

```
In[289]:= VaR1 = -Quantile[a, alpha] * 100

Out[289]= 11.4485

In[290]:= VaR2 = -Quantile[b, alpha] * 200

Out[290]= 22.3177
```

▪

When allowing fractional portfolios the Aumann-Shapley method is the right way to go. Firstly because it is coherent:

Theorem 3.2.1.3 (Denault 2001 [14])
If (N, Λ, r) is a game with fractional players, with r a coherent cost function that is differentiable at Λ, then the Aumann-Shapley value is a coherent fuzzy value.
◊

Secondly, because there is no alternative method that is coherent:

Theorem 3.2.1.3 (Aubin 1979 [8])
If the cost function r is positively homogeneous, then the fuzzy core is non-empty, convex and compact. If furthermore r is differentiable at Λ, then the core consists of a single vector, the gradient $k^{AS} = \nabla r(\Lambda)$
◊

In general, when using Value-at-Risk as a risk measure, it can be shown that the Aumann-Shapley is given by (see for instance McNeil et al 2005 [3]):

$$K_i = -E(X_i \mid X_N = q_\alpha(X_N))$$

where $q_\alpha(X_N)$ is the α percentile of X_N.
In words this means that one finds the certain percentile of X_N and then takes the value of X_i in that specific case. There might be situations where this approach is not fair in case with historical distribution, as subportfolios are only evaluated on one specific event in the history. If a subportfolio had a very bad return that day, the amount of risk capital allocated will be large.

When instead using the Expected Shortfall as a risk measure, it can be shown that the Aumann-Shapley value is again of a shortfall type, namely (see for instance Denault 2001 [14]):

$$K_i = -E(X_i \mid X_N \leq q_\alpha(X_N))$$

where $q_\alpha(X_N)$ is the α percentile of X_N.

The definition of the Aumann-Shapley makes it clear that this method satisfies “No advantageous splitting and merging”. However, one of the drawbacks of the Aumann-Shapley value is that it requires differentiability of r at Λ. While this is not a problem in case with continuous distribution of returns, such as normal or Student-t, it can cause problems when using the historical distribution. This is illustrated in the following example:

Example 3.2.1.2

Consider two portfolios and 4 states of the world at a specific time in future with given probabilities:

Prob	X1	X2	X1+X2
0,1	-60	-6	-66
0,1	0	-60	-60
0,1	-30	y	-30+y
0,7	15	40	55

Assume that we want to apply Expected Shortfall with $\alpha = 20\%$. *This gives us the following result:*

Case	$\rho(X1+X2)$	K_1^{AS}	K_2^{AS}
y>-30	63	30	33
-36<y<-30	48-y/2	45	3-y/2
y<-36	48-y/2	45	3-y/2
y=-30	63	NA	NA
y=-36	66	NA	NA

Note that the Aumann-Shapley value is not defined for $y \in \{-30, -36\}$. For example, consider the case when y=30. Then

Prob	X1	X2	X1+X2
0,1	-60	-6	-66
0,1	0	-60	-60
0,1	-30	-30	-60
0,7	15	40	55

In this case it is not a problem to measure the risk by either VaR or ES, since the 20% percentile is -60. However, there are two cases where X1+X2 = -60. Since we need to check how the result changes when there is a small increase in size of one portfolio, we do not know which of the two cases the calculations should be based on.

▪

4. Implementation of the allocation rules in Matlab

In the previous chapter several allocation rules were presented and their properties were described in detail. When the number of subunits increases, the number of possible coalitions increases as well and the allocation problem becomes larger and more complex. Some of the allocation methods are based on costs/risks of a few coalitions. For example, to allocate risks by using the Activity method, one only has to know the stand-alone risks and the risk of the grandportfolio N. Even with relatively large n the problem remains manageable and never gets too complex. On the other hand, some of the allocation methods are complicated to calculate even for moderate n, since they are based on many or all possible coalitions. For example, in the academic literature authors often argue that the significant drawback of the Shapley method is that it is almost impossible to calculate for large n because the problem becomes too large.

"Note that this requires the evaluation of c for each of the 2^n possible coalitions, unless the problem has some specific structure. Depending on what c is, this task may become impossibly long, even for moderate n."

Denault 2001 [14]

Theorists are mainly interested in the technical properties of the allocation rules. Their examples are usually small, where the number of players rarely gets over 3 or 4. On the other hand, in practice, an allocation rule often needs to be applied to large allocation problems with large number of players/subunits/business lines, etc. In order be useful, an allocation problem should not only satisfy as many of the desired theoretical properties as possible, but it should also be manageable practically, i.e. it should be possible to apply the allocation rule to a specific allocation problem without too much effort. The latter is very important, since without the ease of implementation the theoretically best, but practically complicated allocation rules, will rarely be implemented in practise. Often there is a trade-off between the theoretical properties and the ease of implementation of the allocation rules. Theoretically best allocation rules are usually complex (Shapley, Nucleolus), while practically obvious allocation rules usually are theoretically unsatisfactory in their performance (Incremental). With today's technological progress, computer programs are the obvious candidates to solution to the problem. Allocating risks/costs by a computer program, would satisfy both sides: theoretically best allocation rules can be applied as easily as practically obvious allocation rules: by pressing the "Run" button.

This section will mainly focus on the implementation of the allocation rules described earlier. All the allocation rules were programmed in Matlab as functions and can be used to allocate risk/costs of an arbitrary allocation problem. The next section briefly describes how the Matlab program works. The code itself can be found in Appendix 3 and on the attached CD-Rom. The code works properly in Matlab R2011b.

4.1 Brief description of the program

With n players there are $2^n - 1$ possible coalitions with corresponding costs/risks. The coalitions can be identified by a ($2^n - 1$ x n) matrix with 0 and 1 elements only. The columns correspond to players and columns correspond to coalitions. Elements inside the matrix indicate whether a specific coalition contains a specific player (1 if yes and 0 if no). In case with n=3 the matrix would have the following structure:

Coalition matrix

1	0	0
0	1	0
0	0	1
1	1	0
1	0	1
0	1	1
1	1	1

Corresponding coalitions

{1}
{2}
{3}
{1,2}
{1,3}
{2,3}
{1,2,3}

In our example I made a Matlab function **S=coalitions(n)**, which takes n as input and gives the coalition matrix S as output. By calling the function with input n=4 we get the following output:

```
>> S=coalitions(4)

S =

     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1
     1     1     0     0
     1     0     1     0
     1     0     0     1
     0     1     1     0
     0     1     0     1
     0     0     1     1
     1     1     1     0
     1     1     0     1
     1     0     1     1
     0     1     1     1
     1     1     1     1
```

Apart from being able to control all the possible coalitions by a single matrix, the method has another significant advantage. Since it is a matrix, it can be multiplied with other matrices to achieve different results. For example, we can continue with the game from example 3.1.1. We define the constraints of different coalitions:

```
ES =

    15
    14
    16
    15
    23
    22
    21
    25
    23
    24
    28
    29
    27
    29
    32
```

Note that the number of rows in the coalition matrix S and in the constraint vector ES is the same (here $2^4 - 1$=15). The constraint of coalition *k* is the k'th element in the vector ES, i.e. we can use the same index.

Once we have defined the allocation problem, it is obvious to ask whether the problem has a non-empty core. One way is to check whether the problem is balanced. However, since Matlab has an ability to deal with linear programming, there is an easier way. We can ask the computer to solve the following linear program:

Max $z = x_1 + ... + x_n$

Subject to $\sum_{i \in S} x_i \leq c(S)$ for every $S \subset N$

The core of the game is not empty if and only if the linear program has a maximum, say, z^*, and $z^* \geq c(N)$. A Matlab function "linprog" can find a minimum of a regular LP-problem. So, our problem has to be translated into Matlab format:

Min $z = -x_1 - ... - x_n$

Subject to $S \cdot [x_1 \quad . \quad . \quad . \quad x_n]^T \leq ES$

If the solution is found and if –z* is greater than the risk of the grandcoalition N, then the core is non-empty. This is the approach that is used in my Matlab function **CE=core_existence(S,rm)** where

- S is the coalition matrix (2^n-1 rows and n columns)
- rm is the vector of risks of different coalitions (length 2^n-1)
- CE is the output variable which can take values:
  - 3 if the problem has a non-empty core
  - 2 if the core is empty
  - 0 if the number of iterations exceeded options.MaxIter
  - -2 if no feasible point was found
  - -3 if the problem is unbounded
  - -4 if NaN value was encountered during execution of the algorithm
  - -5 if both primal and dual problems are infeasible
  - -7 if search direction became too small and no further progress could be made

Values 3 and 2 are returned if the LP problem was solved. If the LP problem could not be solved, the code seeks to rerun the problem two more times. If the optimum is not found after 3 trials, the error code is returned instead (a non-positive number). Note that the optimum does exist, so if one gets one of the errors, such as -3, then something went wrong when solving the LP problem, as the LP problem is not unbounded. One should check if the problem is defined correctly in Matlab.

If we take a specific allocation problem, say, the Nucleolus value, the risks allocated to different coalition can be calculated in the following way:

```
>>  allocation=[7.25,9.25,7.25,8.25]';
temp=S*allocation

temp =

    7.2500
    9.2500
    7.2500
    8.2500
   16.5000
   14.5000
   15.5000
   16.5000
   17.5000
   15.5000
   23.7500
   24.7500
   22.7500
   24.7500
   32.0000
```

Now, when having the vector temp one can easily check if the solution is core compatible by checking if any all elements in the vector temp - ES are greater than 0. If this is the case, the core compatibility is violated. I made a Matlab function **CC=core_compatibility (S,cond,allocation,print)** which uses this approach.
The input variables are:

- S: the coalition matrix (2^n-1 rows and n columns)
- cond: a vector that contains the risks/costs of coalitions (length 2^n-1)
- allocation: the allocation that needs to be checked for core compatibility (length: n)
- print: a binary variable which states whether the function should print how many core conditions are violated (print=1 prints and print=0 does not print). If the solution is core compatible, then the function does not print any comments.

The output variable:

- CC a binary output variable which states whether the solution is in the core (CC=1 when the solution is core compatible and CC=0 when the solution is not core compatible).

Recall that allocation is Nucleolus solution. Define a new allocation that is not core compatible:

```
>> allocation2=[20,10,2,0]
```

Core compatible allocation, print=1

```
>> CC=core_compatibility(S,ES,allocation,1)

ans =

The allocation is core compatible


CC =

     1
```

Core compatible allocation, print=0

```
>> CC=core_compatibility(S,ES,allocation,0)

CC =

     1
```

Not core compatible allocation, print=1

```
>> CC=core_compatibility(S,ES,allocation2,1)

ans =

Core compatibility is violated in that many conditions:


ans =

     4


CC =

     0
```

Not core compatible allocation, print=0

```
>> CC=core_compatibility(S,ES,allocation2,0)

CC =

     0
```

Once we have all the possible coalitions, we need to measure the risks of all the possible coalitions in order to set the problem up as an allocation problem. As previously stated in chapter 2, we can either use the historical distribution or we can make a specific distribution assumption of returns, such as normal. These two cases are considered separately in the next two sections.

4.1.1 Historical distribution

To calculate the risks of all possible coalitions from the historical data by applying a specific risk measure, one needs the following data:

- The coalition matrix S: can be generated by my function **S=coalitions(n)**.
- The matrix with historical returns. The historical prices can be downloaded from Yahoo Finance by using the function **hist_stock_data**. By using this approach one can create a historical return matrix “ret” which consists of a number of rows and n columns, where rows correspond to specific dates in the past and columns correspond to the values of a specific player's portfolio. So, element (i,j) states the return of the portfolio of player j at date i.
- The vector with amounts invested in each portfolio: The required input from the user.
- Long/short position: Long position is assumed by default. If the position is short, multiply the amount invested in the portfolio by -1, so that it becomes negative.

To estimate the risk of the portfolio of a specific coalition by using historical returns one has to do the following:

1. **Take a specific coalition by extracting a specific row from the coalition matrix S.**

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1
1	1	0	0
1	0	1	0
1	0	0	1
0	1	1	0
0	1	0	1
0	0	1	1
1	1	1	0
1	1	0	1
1	0	1	1
0	1	1	1
1	1	1	1

2. **Multiply the extracted vector with the vector which stores amounts invested in each subportfolio to get the amounts invested in different portfolios by that specific coalition.**

Invested amounts:	100	200	300	100
Coalition:	1	0	1	0
Amounts invested by the coalition:	100	0	300	0

3. **Multiply the output vector from step 2 with the matrix which stores historical returns to get the vector of historical profits of that specific coalition:**

Returns of the four different subportfolios at specific dates in past

Profits measured in $ of the portfolio of the coalition at specific dates in the past

```
ret =
```

Dates

Subportfolio 1	Subportfolio 2	Subportfolio 3	Subportfolio 4	Profit
-0.0296	0.1536	0.0078	-0.0465	-0.6136
-0.0231	0.1175	0.0078	-0.0293	0.0173
-0.0343	0.0918	0.0038	-0.0095	-2.2758
0.0299	-0.0677	0.0023	0.0382	3.6815
0.0200	-0.0785	0.0054	0.0311	3.6065
0.0327	-0.0620	-0.0114	0.0386	-0.1498
0.0035	-0.0275	0.0077	-0.0162	2.6589
-0.0053	0.0487	0.0267	0.0073	7.4826
-0.0217	-0.0332	-0.0007	-0.0291	-2.3906
0.0400	-0.0966	-0.0179	0.0563	-1.3524
0.0116	-0.0905	0.0341	0.0012	11.3832
-0.0016	-0.0211	-0.0037	0.0046	-1.2580
0.0236	-0.0069	-0.0074	0.0400	0.1553
-0.0140	0.0343	0.0148	-0.0185	3.0486
…	…	…	…	…
-0.0019	-0.0289	0.0103	-0.0025	2.8906
-0.0072	0.0364	-0.1238	0.0028	-37.8580
0.0031	-0.0548	0.0079	0.0003	2.6950
0.0139	-0.0378	0.0033	0.0181	2.3741
-0.0028	0.0919	-0.0258	-0.0021	-8.0092
-0.0049	-0.0064	-0.0258	-0.0101	-8.2244
-0.0016	0.0006	0.0101	-0.0022	2.8751
0.0037	0.0013	0.0016	0.0044	0.8480

Subportfolios

The first element in the output vector, -0.6136 =-0.0296*100+0.1536*0+0.0078*300-0.0465*0, states that on the first historical date the profit of the portfolio of coalition {1,3} was -0,6136 USD, i.e. a loss of 61.36 cents. Note that even though only players 1 and 3 are present in the coalition, we can still use the same total return matrix, which contains the historical returns of all four players. The returns of the absent players get automatically multiplied by 0.

4. **Apply a risk measure to the output vector from step 3, as demonstrated previously in chapter 2, and store the output number.**

For example, calculate a certain percentile and flip sing if using VaR or calculate the minus mean of the historical profits which are below the –VaR level if using ES.

4.1.2 Specific distribution assumption

From chapter 2 we know that when we combine assets whose returns are normally distributed, the portfolio that consists of these assets will also be normally distributed with deterministic mean and variance. However, the task gets more complicated when working with other distributions such as Student-t. One of the ways to solve this problem is by Monte Carlo simulation. Using this approach we ask Matlab to simulate a large number of returns, say 100.000 simulations. Once the numbers are simulated, we can use the same approach as with historical distribution.

To make a simulation the following input is needed:

- A vector of length n, which contains Means of the returns of the n subportfolios
- A vector of length n, which contains Standard Deviations of the returns the n subportfolios
- A n x n correlation matrix

Note that the information for the last two bullet points can be extracted from the covariance matrix Σ. First recall the properties of a covariance matrix:

- It is symmetric so that $\Sigma^T = \Sigma$
- The diagonal elements contain the variances and satisfy $\Sigma_{i,i} \geq 0$
- It is positive semi-definite so that $x^T \Sigma x \geq 0$ for all $x \in R^n$

Standard Deviations are simply the square roots of the diagonal elements and $Corr(X_1, X_2) = \dfrac{Cov(X_1, X_2)}{\sqrt{Var(X_1)Var(X_2)}}$

Matlab can generate a series of **independent** random variables from specific distributions, such as normal or Stundet-t. However, our task is to generate a series of n random variables from a specific distribution **with a specific correlation structure**. The problem can be solved by using the Cholesky decomposition method. The purpose of the method is to make the uncorrelated random numbers, generated for the purpose of Monte Carlo simulation, correlated using the underlying volatilities and correlation matrix of the assets.

Generating correlated random variables with Cholesky decomposition

Our goal is to generate $X = (X_1, ..., X_n)$ where $X \sim N(0, \Sigma)$.

Matlab can generate independent $Z_i \sim N(0,1)$ for i=1,…,n.

Assume that we have a vector of $\breve{n}$ positive constants: $\hat{c} = (\hat{c}_1, ..., \hat{c}_{\breve{n}})$ where $\hat{c}_i > 0$ for i=1,…, $\breve{n}$.

Because our generated variables are independent, i.e. $Cov(Z_i, Z_j) = 0$ for all $i \neq j$, and because of the well-known variance rules (a is a constant):

$$Var(aX) = a^2 Var(X)$$
$$Var(X + Y) = Var(X) + Var(Y) + 2Cov(X, Y)$$

the following is true:

$$\hat{c}_1 Z_1 + \ldots + \hat{c}_{\tilde{n}} Z_{\tilde{n}} \sim N(0,\sigma^2) \text{ where } \sigma^2 = \hat{c}_1^{\,2} + \ldots + \hat{c}_{\tilde{n}}^{\,2} .$$

That is, as stated previously, a linear combination of normal random variables is normal.

More generally, let C be a (m x m) symmetric matrix and let $Z = (Z_1, \ldots, Z_m)^T$. Then

$$C^T Z \sim N(0, C^T C)$$

so, our problem reduces to finding C, such that $C^T C = \Sigma$.

Now, recall that a covariance matrix is a positive semi-definite. The Cholesky method can be applied if and only if the covariance matrix is positive definite, i.e. all eigenvalues of the covariance matrix must be positive. In practical terms a covariance matrix can be positive definite, but this implies that no subportfolio can have a perfect correlation with another subportfolio, i.e. no non-diagonal elements in the correlation matrix are equal to -1 or 1. In practise, no portfolios are perfectly correlated, apart from the cases where two players have the same position in exactly the same portfolio (correlation of 1) or the reverse position (i.e. long vs. short) in exactly the same portfolio (correlation of -1).

A well-known fact from linear algebra is that any symmetric positive-definite matrix, M, may be written as

$$M = U^T D U$$

where U is an upper triangular matrix and D is a diagonal matrix with positive diagonal elements. When no portfolios are perfectly positively or negatively correlated, our covariance matrix is positive definite and we may therefore write

$$\Sigma = U^T D U = (U^T \sqrt{D})(\sqrt{D} U) = (\sqrt{D} U)^T (\sqrt{D} U)$$

The matrix $C = \sqrt{D} U$ satisfies $C^T C = \Sigma$.

How this information is relevant when working with Matlab is illustrated below.

Cholesky decomposition in Matlab

Matlab has a built-in function **chol** which computes the Cholesky decomposition of a symmetric positive-definite matrix. The approach is illustrated by an example, where we generate a 100.000 x 3 matrix where numbers in columns are 3 normally distributed variables with a given covariance structure.

```
>> Sigma=[1 0.5 0.5; 0.5 2 0.3 ; 0.5 0.3 1.5]

Sigma =

    1.0000    0.5000    0.5000
    0.5000    2.0000    0.3000
    0.5000    0.3000    1.5000
```

Define the correlation matrix

```
>> C=chol(Sigma)

C =

    1.0000    0.5000    0.5000
         0    1.3229    0.0378
         0         0    1.1174
```

Make the Cholesky decomposition

```
>> tic
Z=randn(3,100000);
X=C'*Z;
cov(X')
toc

ans =

    1.0012    0.5022    0.4988
    0.5022    1.9914    0.2939
    0.4988    0.2939    1.4961

Elapsed time is 0.028955 seconds.
```

Generate a random 100000x3 matrix with independent numbers that all follow standard normal distribution. Multiply the transposed C by this matrix. In fact X is a 100000x3 matrix where elements are normally distributed.

We see that the 3 columns in matrix X have correlation structure defined in the very beginning of the code.

Of course, when the number of simulations increases, the precision of the result also increases, but so does the computing time. The same approach with 1.000.000 and 10.000.000 simulations:

```
>> tic
Z=randn(3,1000000);
X=C'*Z;
cov(X')
toc

ans =

    1.0005    0.4993    0.5013
    0.4993    2.0003    0.3027
    0.5013    0.3027    1.4987

Elapsed time is 0.243735 seconds.
```

```
>> tic
Z=randn(3,10000000);
X=C'*Z;
cov(X')
toc

ans =

    1.0006    0.5003    0.5006
    0.5003    1.9996    0.3001
    0.5006    0.3001    1.5008

Elapsed time is 1.632826 seconds.
```

4.2 The allocation rules

All eight allocation rules were programmed as functions in Matlab. The methods are fully invented by me. The code can be found in Appendix 3 or on the CD-Rom. The table below summarizes which approaches are used when programming the allocation rules. The following input variables are used:

- S: $2^n - 1$ x n coalition matrix
- rm: vector of length $2^n - 1$: stand-alone risks of all the possible coalitions
- InvAmount: vector of length n: amounts invested in the different subportfolios.
- ret matrix of historical returns with n columns
- alpha the confidence level of Expected Shortfall or Value-at-Risk

	Input	**Method**
Activity	S,rm	Reference to specific elements of the input data
Incremental	S,rm	Reference to specific elements of the input data
Beta	ret, InvAmount, rm	Calculation of covariances and betas
Cost_gap	S,rm	Reference to specific elements of the input data
Nucleolus	S,rm	Solving a series of LP problems
Shapley	S,rm	Reference to specific elements of the input data
Lorenz	S,rm	Solving a LP-problem
Aumann-Shapley	S, ret, InvAmount, alpha	Calculating mean of the specific historical data

When calculating Nucleolus or Lorenz, the allocation is a solution obtained from solving at least one LP problem. As a consequence, the result might be imprecise to a certain degree depending on which algorithm is used for solving these LP problems. The approaches used for Nucleolus and Lorenz are quite different. In case with Nucleolus the Matlab function linprog is used, while the function quadprog is used in case with Lorenz.

linprog

$$\min_x f^T x \text{ such that } \begin{cases} A \cdot x \le b, \\ Aeq \cdot x = beq, \\ lb \le x \le ub. \end{cases}$$

quadprog

x = quadprog(H,f) returns a vector x that minimizes 1/2*x'*H*x + f'*x. H must be positive definite for the problem to have a finite minimum.

x = quadprog(H,f,A,b) minimizes 1/2*x'*H*x + f'*x subject to the restrictions A*x ≤ b. A is a matrix of doubles, and b is a vector of doubles.

x = quadprog(H,f,A,b,Aeq,beq) solves the preceding problem subject to the additional restrictions Aeq*x = beq. Aeq is a matrix of doubles, and beq is a vector of doubles. If no inequalities exist, set A = [] and b = [].

Although very complicated, linprog can be used to calculate the Nucleolus allocation by giving the input variables the following structure (lb and ub undefined):

$$f = \begin{bmatrix} \underbrace{0,...,0}_{n \text{ zeros}}, -1, 0 \end{bmatrix}, A = \begin{bmatrix} \overline{S} & \overline{0} & \overline{0} \\ \overline{S} & \overline{1} & -rm1 \end{bmatrix}, b = \begin{bmatrix} rm1 \\ \overline{0} \end{bmatrix}, Aeq = \begin{bmatrix} 1,...,1,0,0 \\ \underbrace{0,...,0}_{length\ n}, 0, 1 \end{bmatrix}, beq = \begin{bmatrix} rm2 \\ 1 \end{bmatrix}$$

where $\overline{0}$ and $\overline{1}$ are column vectors of length $2^n - 2$ with 0 and 1 elements respectively; rm1 is a column vector of length $2^n - 2$ that contains the core conditions of all the possible coalitions disregarding the empty and the grand coalition; rm2 is the total risk that needs to be shared.

Example 3.1.1, cont.

Here the linprog function would need the following input to solve the Nucleolus:

$f =$ 0 0 0 0 -1 0

A						*b*
1	0	0	0	0	0	15
0	1	0	0	0	0	14
0	0	1	0	0	0	16
0	0	0	1	0	0	15
1	1	0	0	0	0	23
1	0	1	0	0	0	22
1	0	0	1	0	0	21
0	1	1	0	0	0	25
0	1	0	1	0	0	23
0	0	1	1	0	0	24
1	1	1	0	0	0	28
1	1	0	1	0	0	29
1	0	1	1	0	0	27
0	1	1	1	0	0	29
1	0	0	0	1	-15	0
0	1	0	0	1	-14	0
0	0	1	0	1	-16	0
0	0	0	1	1	-15	0
1	1	0	0	1	-23	0
1	0	1	0	1	-22	0
1	0	0	1	1	-21	0
0	1	1	0	1	-25	0
0	1	0	1	1	-23	0
0	0	1	1	1	-24	0
1	1	1	0	1	-28	0
1	1	0	1	1	-29	0
1	0	1	1	1	-27	0
0	1	1	1	1	-29	0

Aeq						*beq*
1	1	1	1	0	0	32
0	0	0	0	0	1	1

Ensures that the solution sums up to the total risk and that the last variable is equal to 1

The first half of the matrix is simply the core conditions.

The second half defines the excesses for all the coalitions. For example, the last row states that

$$x_2 + x_3 + x_4 + \varepsilon - \underbrace{c(\{2,3,4\})}_{=29} \le 0 \Leftrightarrow c(\{2,3,4\}) - \sum_{i=2,3,4} x_i \ge \varepsilon$$

The ε itself is maximized by minimizing $-\varepsilon$.

The code then solves next LP problems by moving some of the conditions from A (inequalities) to Aeq (equalities). The algorithm stops when the solution is unchanged after solving the new LP problem.

▪

In case with Lorenz, the input variable should have the following structure:

H is a nxn diagonal matrix with -1 elements in the diagonal; f empty; $A = \overline{S}$; b=rm1; *Aeq* is a vector of length n with 1 elements only; *beq* is the total risk that has to be shared.

Example 3.1.1, cont.

Here the Lorenz function would minimize the following function:

$$\frac{1}{2}\begin{pmatrix} x_1 & x_2 & x_3 & x_4 \end{pmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \frac{1}{2}\left(x_1^2 + x_2^2 + x_3^2 + x_4^2\right)$$

which gives the same result, as maximizing $-\frac{1}{2}\left(x_1^2 + x_2^2 + x_3^2 + x_4^2\right)$, which is obviously concave.

▪

Note that while the first seven functions would work with any risk measures, the Aumann-Shapley function can only calculate the risk allocation for Expected Shortfall and Value-at-Risk, which both are generated as output. The reason is simple: in the first case one first has to set up the coalitional game by using a certain risk measure. Then the allocation rules get this game as input. In case with Aumann-Shapley one has to calculate how the risk measurement changes with a slight change in the amount invested in a certain portfolio. Thus the risk measure itself should be programmed into the function.

5. Examples

This chapter includes several examples which illustrate strengths and weaknesses of the different allocation rules. The programming approach is explained in chapter 4 and the historical data is downloaded from Yahoo Finance. Consider a company which has a grandportfolio that consists of several subportfolios. The challenge is to allocate the total risk capital to all the subportfolios. In all the following examples we set α to 5% level.

Example 5.1

Assume that there are 10 subunits that all invest in Microsoft. The invested amounts are (100,200,300,400,500,600,700,800,900,1000). We use 4 years of historical data to estimate VaR and ES. By specifying the invested amounts and the Yahoo-Finance ticker codes in Matlab, we get the following result by pressing the run button:

Different allocation rules in the following order:

Activity, Incremental, Beta, Cost-Gap, Nucleolus, Shapley, Lorenz, Aumann-Shapley

```
Solution_ES =

    4.9823    4.9823    4.9823    4.9823    4.9823    4.9823    4.9823    4.9823
    9.9647    9.9647    9.9647    9.9647    9.9647    9.9647    9.9647    9.9647
   14.9470   14.9470   14.9470   14.9470   14.9470   14.9470   14.9470   14.9470
   19.9294   19.9294   19.9294   19.9294   19.9294   19.9294   19.9294   19.9294
   24.9117   24.9117   24.9117   24.9117   24.9117   24.9117   24.9117   24.9117
   29.8941   29.8941   29.8941   29.8941   29.8941   29.8941   29.8941   29.8941
   34.8764   34.8764   34.8764   34.8764   34.8764   34.8764   34.8764   34.8764
   39.8587   39.8587   39.8587   39.8587   39.8587   39.8587   39.8587   39.8587
   44.8411   44.8411   44.8411   44.8411   44.8411   44.8411   44.8411   44.8411
   49.8234   49.8234   49.8234   49.8234   49.8234   49.8234   49.8234   49.8234
```

Risk allocated to specific subunits: row i is the risk allocated to player i.

```
ans =

sum ES


ans =

  274.0288  274.0288  274.0288  274.0288  274.0288  274.0288  274.0288  274.0288
```

Controls that all the allocated shares sum up to the total risk

```
Solution_VAR =

    3.2659    3.2659    3.2659    3.2659    3.2659    3.2659    3.2659    3.2659
    6.5319    6.5319    6.5319    6.5319    6.5319    6.5319    6.5319    6.5319
    9.7978    9.7978    9.7978    9.7978    9.7978    9.7978    9.7978    9.7978
   13.0638   13.0638   13.0638   13.0638   13.0638   13.0638   13.0638   13.0638
   16.3297   16.3297   16.3297   16.3297   16.3297   16.3297   16.3297   16.3297
   19.5956   19.5956   19.5956   19.5956   19.5956   19.5956   19.5956   19.5956
   22.8616   22.8616   22.8616   22.8616   22.8616   22.8616   22.8616   22.8616
   26.1275   26.1275   26.1275   26.1275   26.1275   26.1275   26.1275   26.1275
   29.3935   29.3935   29.3935   29.3935   29.3935   29.3935   29.3935   29.3935
   32.6594   32.6594   32.6594   32.6594   32.6594   32.6594   32.6594   32.6594


ans =

sum VaR


ans =

  179.6267  179.6267  179.6267  179.6267  179.6267  179.6267  179.6267  179.6267
```

Note that this problem is relatively large, since the number of possible coalitions in this case is 1023. Nevertheless, the program easily computes the allocations by using different allocation rules. We see that according to Expected Shortfall the risk is 274.0288, while 179.6267 when using Value-at-Risk. As expected, Expected Shortfall, gives a larger risk measurement than Value-at-Risk, since it is the expected value given that VaR level is violated. In this example all the subunits have exactly the same portfolio with different invested amounts. It is thus obvious that in this case the only fair allocation would be allocating the risk in proportion to the invested amounts. All eight allocation rules capture this and allocate the risk measurement number in this way. For example, because player 2's invested amount is twice as large as player 1's, player 2 gets allocated twice as much as player 1. Note that the correlation matrix in this case is a 10x10 matrix with 1 elements only, i.e. risks of the subunits are perfectly positively correlated. According to theorem 3.1.10 the core only consists of only one element, namely where each subdisivion gets allocated its stand-alone risk. All eight allocations are thus in the core of the allocation problem.

Example 5.2

Consider a company with 3 different subunits that invest 100$ in BMW, Volkswagen and Porsche respectively. We use 2 years of historical data to calculate ES and VaR. The result is as follows:

Order of the allocation rules:
Activity, Incremental, Beta, Cost-Gap, Nucleolus, Shapley, Lorenz, Aumann-Shapley

```
Solution_ES =

    4.5299    4.5673    4.4500    4.5639    4.5981    4.5360    4.1807    4.4227
    4.1558    4.1099    4.2195    4.1423    4.1794    4.1135    3.7621    3.8280
    6.1066    6.1150    6.1228    6.0861    6.0148    6.1428    6.8495    6.5416


sum_ES =

   14.7923   14.7923   14.7923   14.7923   14.7923   14.7923   14.7923   14.7923


Solution_VAR =

    3.2177    3.3275    3.2932    3.2706    3.2837    3.2658    3.3900   -2.0387
    3.2290    2.8732    3.1226    3.0575    2.9916    3.0643    2.6272   -3.6667
    4.5001    4.7461    4.5311    4.6187    4.6716    4.6168    4.9296   16.6522


sum_VaR =

   10.9468   10.9468   10.9468   10.9468   10.9468   10.9468   10.9468   10.9468
```

What we see is that the different allocation rules more or less agree on how the risk should be allocated. In this example the subportfolios are highly positively correlated. The intuition is that the prices of these stocks are partly driven by common European car manufacturer's factor. Because all three subunits have a long position in European car manufacturer's stocks, the correlation between risks of different subunits is high and the diversification benefit is thus low. Hence there is not much benefit that needs to be divided between the subunits. So, in this situation there is no need to apply one of the complicated allocation rules. As previously stated, Aumann-Shapley can give non-sense results when using VaR as a risk measure. Here we see what can happen: Why should player 1 and 2 get allocated negative shares? Player 3 gets allocated too large amount of risk capital. The solution is very different from the very same case when using ES, where Aumann-Shapley agrees with the other methods. The reason is simple: in that specific case where the grandportfolio has the 5% percentile, subportfolio 3 actually had a relatively bad return. The allocation is only based on that specific case, so player 3 gets allocated very large amount of risk capital. On the other hand, in case with ES, the result is based on the many cases where the grandportfolio had worse outcomes than the 5% percentile. Here the Aumann-Shapley approach is adequate.

Assume that the company gets its fourth subunit, which invests $\widehat{W}$ in a short Peugeot position. First assume that $\widehat{W}$ = 50. The situation now is as follows:

	Player 1	**Player 2**	**Player 3**	**Player 4**
Portfolio	BMW	Volkswagen	Porsche	Peugeot
Invested amount	100	100	100	$\widehat{W}$ =50
Long/short	Long	Long	Long	Short

Because we expect high correlation between the different subunits, we expect that addition of subunit 4 will reduce the total risk of the company. When the first three subportfolios increase in value, the fourth subportfolio decreases in value and vice verca. So, the subunit 4 should get allocated very low amount of risk capital. By rerunning the program the following results get generated:

Order of the allocation rules:
Activity, Incremental, Beta, Cost-Gap, Nucleolus, Shapley, Lorenz, Aumann-Shapley

```
Solution_ES =

    3.2473    4.5595    4.2097    4.3088    4.3651    3.9362    3.9179    4.3207
    2.9791    4.3272    4.0405    4.0380    4.1655    3.6618    3.7183    3.9017
    4.4099    6.2464    5.9483    5.8836    5.8146    5.6047    5.3674    6.3545
    2.0751   -2.4217   -1.4871   -1.5191   -1.6337   -0.4913   -0.2922   -1.8655


sum_ES =

   12.7114   12.7114   12.7114   12.7114   12.7114   12.7114   12.7114   12.7114


Solution_VAR =

    2.3980    3.5317    3.2871    3.5485    3.4833    3.0815    3.4853    2.3014
    2.4064    3.2956    3.1550    3.3056    3.1345    2.9375    3.2424    2.3008
    3.3537    4.0910    4.6447    4.1505    4.2125    4.1018    4.0606    4.1100
    1.7674   -0.9928   -1.1612   -1.0791   -0.8847   -0.1953   -0.8628    1.2133


sum_VaR =

    9.9255    9.9255    9.9255    9.9255    9.9456    9.9255    9.9255    9.9255
```

As expected, the total risk is now reduced (for example 12.7114 vs. 14.7923 when using ES) . As previously stated, the Activity method only takes the stand-alone risks into consideration, so in the first column player 4 not surprisingly gets the largest amount of risk capital among all eight allocation rules. The other allocation rules suggest paying player 4 for being a part of the grandcoalition.

What happens when $\widehat{W}$, the amount of money invested in the portfolio of player 4, increases? The results are calculated below:

$\hat{W} = 100$

```
Solution_ES =

    2.4818    4.9708    3.7079    3.9896    4.2398    3.2713    3.2246    3.6951
    2.2768    4.8747    3.6022    3.7835    3.9492    3.0599    3.1622    3.5053
    3.3703    6.8376    5.4052    5.4521    5.2692    4.7950    4.4356    5.8230
    3.1718   -5.3824   -1.4145   -1.9245   -2.1575    0.1745    0.4784   -1.7226

sum_ES =

   11.3007   11.3007   11.3007   11.3007   11.3007   11.3007   11.3007   11.3007

Solution_VAR =

    1.7066    3.9126    2.7305    2.9923    3.0650    2.4418    2.6053    4.1457
    1.7126    3.6385    2.6527    2.8919    2.9763    2.3212    2.4228    3.4514
    2.3868    4.7129    3.9804    3.8922    3.5994    3.4446    3.1382    4.7248
    2.5158   -3.9421   -1.0417   -1.4546   -1.3188    0.1143    0.1556   -4.0000

sum_VaR =

    8.3218    8.3218    8.3218    8.3218    8.3218    8.3218    8.3218    8.3218
```

$\hat{W} = 200$

```
Solution_ES =

    1.9653   -6.8494    2.0202    2.6660    2.6489    2.1424    2.8651    1.4729
    1.8030   -8.3166    2.0632    2.5314    2.6927    1.9747    2.8651    1.4421
    2.6689  -16.0542    3.3297    3.9049    3.4336    3.3740    2.8651    3.5469
    5.0234   42.6807    4.0474    2.3582    2.6852    3.9695    2.8651    4.9986

sum_ES =

   11.4605   11.4605   11.4605   11.4605   11.4605   11.4605   11.4605   11.4605

Solution_VAR =

    1.3137   11.9213    1.4705    1.7470    1.7025    1.3082    2.0856    0.9408
    1.3183   52.7016    1.5018    2.0603    2.1435    1.4365    2.0856   -0.2483
    1.8373  126.7659    2.4237    3.2733    3.3372    2.6603    2.0856   -0.6002
    3.8730 -183.0465    2.9462    1.2617    1.1591    2.9372    2.0856    8.2500

sum_VaR =

    8.3423    8.3423    8.3423    8.3423    8.3423    8.3423    8.3423    8.3423
```

$\hat{W} = 300$

```
Solution_ES =

    2.1788   10.7773    0.4110    2.0631    2.0886    1.6878    3.8726   -0.0337
    1.9989   11.1932    0.5644    1.8488    2.0132    1.4771    3.8726    0.0362
    2.9589    4.5323    1.2305    3.1414    2.6859    2.7926    3.8726    1.0289
    8.3540  -11.0122   13.2846    8.4374    8.7030    9.5331    3.8726   14.4592

sum_ES =

   15.4906   15.4906   15.4906   15.4906   15.4906   15.4906   15.4906   15.4906

Solution_VAR =

    1.3869    7.2011    0.2880    1.2832    1.3847    0.8947    2.7130    3.0291
    1.3918    6.8362    0.3954    1.3097    1.3987    0.8476    2.7130    4.4444
    1.9397   -4.0620    0.8621    2.5911    2.7594    2.1198    2.7130    2.6157
    6.1335    0.8767    9.3066    5.6679    5.3092    6.9899    2.7130    0.7627

sum_VaR =

   10.8520   10.8520   10.8520   10.8520   10.8520   10.8520   10.8520   10.8520
```

$\hat{W} = 500$

```
Solution_ES =

    2.7465   -6.6824   -1.2481    1.5596    1.6049    1.3114    4.5974   -1.3368
    2.5197   -6.9759   -1.0146    1.3181    1.4831    1.1061    4.5974   -1.3909
    3.7299   -6.1237   -1.0623    2.5114    2.0272    2.2945    5.5975   -0.4051
   17.5512   46.3294   29.8724   21.1583   21.4321   21.8354   11.7551   29.6801

sum_ES =

   26.5474   26.5474   26.5474   26.5474   26.5474   26.5474   26.5474   26.5474

Solution_VAR =

    1.8105  -10.9364   -0.9169    0.7367    0.8171    0.5741    3.3036   -4.0068
    1.8168  -10.4926   -0.7454    0.7824    0.8544    0.6252    3.3036   -3.2015
    2.5321   -8.1726   -0.7805    1.6675    1.4279    1.5359    4.3397    0.7327
   13.3442   49.1052   21.9464   16.3170   16.4043   16.7685    8.5568   25.9792

sum_VaR =

   19.5036   19.5036   19.5036   19.5036   19.5036   19.5036   19.5036   19.5036
```

We see that as the amount invested in the portfolio of player 4 increases, player 4 gets allocated more and more risk capital. We also see that first the total risk measurement gets smaller and smaller as $\widehat{W}$ rises, but then begins to increase. The reason for that is that when $\widehat{W}$ is small, player 4 benefits the company, since it hedges the other subunits. However, when $\widehat{W}$ gets large, player 4 becomes the main risk source of the company. So, the other units hedge subunit 4 instead of the opposite.

Note the non-sense allocation provided by the Incremental method when $\widehat{W}$ =200. It suggests the other players to pay player 4 an insane amount for being part of the grandcoalition. In a real life situation subunits 1, 2 and 3 would try to block the cooperation, as they are required to pay and amount that is much larger than their stand-alone risks. This is a good example of the fact that solely information about contributions to the main unit can be misleading in some cases.

Lorenz allocation seeks to allocate the risk capital as equally as possible given that the solution should be in the core. The equal split is the core-allocation in case with $\widehat{W}$ =200 and $\widehat{W}$ =300. The question is whether Lorenz allocation is fair in these two cases. If player 4 gets invested 100$ more in his portfolio, one could argue that he should also be responsible for the major part of the extra risk capital. However, in case with Lorenz all players equally share the extra risk created by solely player 4.

Example 5.3

Now consider the situation from the previous example where $\widehat{W}$ =200. Assume that subunit 4 gets divided into two different subunits, where the portfolio is the same and the invested amount is equally shared, i.e. 100$ in each subunit of the initial subunit 4. The historical estimation window is still 2 years.

Before:

	Player 1	**Player 2**	**Player 3**	**Player 4**
Portfolio	BMW	Volkswagen	Porsche	Peugeot
Invested amount	100	100	100	200
Long/short	Long	Long	Long	Short

After:

	Player 1	**Player 2**	**Player 3**	**Player 4**	**Player 5**
Portfolio	BMW	Volkswagen	Porsche	Peugeot	Peugeot
Invested amount	100	100	100	100	100
Long/short	Long	Long	Long	Short	Short

The result:

Order of the allocation rules:
Activity, Incremental, Beta, Cost-Gap, Nucleolus, Shapley, Lorenz, Aumann-Shapley

```
Solution_ES =

    1.9653    2.2229    2.0202    2.2670    1.5412    1.9220    2.2857    1.4729
    1.8030    2.6990    2.0632    2.1791    1.5931    1.8033    2.2857    1.4421
    2.6689    5.2101    3.3297    3.2855    2.7522    3.2016    2.3016    3.5469
    2.5117    0.6643    2.0237    1.8645    2.7870    2.2668    2.2937    2.4993
    2.5117    0.6643    2.0237    1.8645    2.7870    2.2668    2.2937    2.4993


sum_ES =

   11.4605   11.4605   11.4605   11.4605   11.4605   11.4605   11.4605   11.4605


Solution_VAR =

    1.3137    0.5119    1.4705    1.5056    1.5010    1.2711    1.6347    0.9408
    1.3183    2.2632    1.5018    1.8597    1.3185    1.3443    1.6347   -0.2483
    1.8373    5.4437    2.4237    2.8478    2.2936    2.4147    1.8037   -0.6002
    1.9365    0.0618    1.4731    1.0646    1.6147    1.6561    1.6347    4.1250
    1.9365    0.0618    1.4731    1.0646    1.6147    1.6561    1.6347    4.1250


sum_VaR =

    8.3423    8.3423    8.3423    8.3423    8.3423    8.3423    8.3423    8.3423
```

We calculate how much player 4 and 5 (the initial player 4) get allocated together:

```
>> Solution_ES(4,:)+Solution_ES(5,:)

ans =

    5.0234    1.3285    4.0474    3.7289    5.5740    4.5336    4.5874    4.9986

>> Solution_VAR(4,:)+Solution_VAR(5,:)

ans =

    3.8730    0.1235    2.9462    2.1292    3.2293    3.3122    3.2693    8.2500
```

As discussed earlier, the Activity, Beta and Aumann-Shapley are the only allocation rules where the splitting and merging have no effect on the allocation. This is exactly what we see in this example (the numbers should be compared to the case $\widehat{W}$ = 200 on page 66). The other allocation rules allocate more risk capital to players 4 and 5 than to the initial player 4. Hence, merging for player 4 and 5 is advantageous when using Incremental, Beta, Cost Gap, Nucleolus, Shapley and Lorenz.

Example 5.4

Consider a new case with 5 players and the same invested amount of 100. Assume that the fifth subunit invests 100 $ in a riskless portfolio, where the profit is 1%, i.e. the profit is 1$ in all outcomes.

	Player 1	Player 2	Player 3	Player 4	Player 5
Portfolio	S&P 500	Oil	Microsoft	Google	Riskless
Invested amount	100	100	100	100	100
Long/short	Long	Long	Long	Short	-

We use 5 years of historical data and get the following result:

Order of the allocation rules:
Activity, Incremental, Beta, Cost-Gap, Nucleolus, Shapley, Lorenz, Aumann-Shapley

```
Solution_ES =

    2.0178    4.0948    2.6449    3.2976    2.8015    2.6389    2.7910    3.3337
    2.8815    6.0960    4.4682    4.8137    4.4962    4.3507    4.1551    4.9328
    2.3010    4.3181    3.1786    3.6123    3.1684    3.0610    2.9433    3.5688
    2.6079   -3.7236   -0.9736   -1.4054   -0.1479    0.2677    0.4288   -1.5171
   -0.4900   -1.4671         0   -1.0000   -1.0000   -1.0000   -1.0000   -1.0000


sum_ES =

    9.3182    9.3182    9.3182    9.3182    9.3182    9.3182    9.3182    9.3182


Solution_VAR =

    1.1864    3.4122    1.5003    1.8562    1.5760    1.5659    1.4048    1.0068
    1.8288    5.9915    2.5345    3.0157    2.9645    2.8666    2.2367    5.1661
    1.3140    3.5972    1.8030    2.0174    1.8010    1.7716    1.7380    0.2328
    1.4006   -4.9929   -0.5522   -0.6039   -0.0559    0.0814    0.9061   -0.1202
   -0.4443   -2.7225         0   -1.0000   -1.0000   -1.0000   -1.0000   -1.0000


sum_VaR =

    5.2855    5.2855    5.2855    5.2855    5.2855    5.2855    5.2855    5.2855
```

We see that Cost Gap, Nucleolus, Shapley, Lorenz and Aumann-Shapley satisfy the riskless allocation, which is not surprising considering the results from the table on page 37. Note that Beta allocates no risk capital to player 5. This will always be the case, as the covariance of the riskless portfolio and the grandportfolio will always be 0.

6. The core-compatibility ratios of the allocation rules

As previously stated, many of the allocation rules do not satisfy "no-undercut" axiom, i.e. they are not core compatible. The goal of this section is to find out in what extent the core compatibility is violated by different allocation rules. We are mainly interested in the Shapley method, since this is the only method that satisfies Pareto Optimality, Strong Monotonicity and Symmetry at the same time. The simulation results presented in this section can thus give an answer to the level of impossibility of fair risk allocation in practice. Additionally, the simulation results will show the core compatibility ratio of all the mentioned methods, except Aumann-Shapley, Lorenz and Nucleolus which are always core compatible.

We simulate the arithmetic returns of a company consisting of a number of subunits. The returns are assumed to have a Student-t distribution. Recall that a Student-t distribution with a high value of degrees of freedom is very close to normal distribution. The standard deviation of the returns of each subunit is random and is assumed to be uniformly distributed between 0,005 and 0,8. We are interested in checking how the core compatibility changes when input variables vary. We will focus on the following input variables:

- Number of players: from 3 to 7
- Degree of confidence of a risk measure, 1-α : 1%, 5%, 9%, 13%
- Degrees of freedom of the distribution of returns (tail thickness): 3, 5, 7, 9

We use the Monte Carlo approach described in section 4.1.2 with some small adjustments. The program has the following steps:

1. Generate a random correlation matrix.

First generate a n x n lower triangular matrix with random elements uniformly distributed between -1 and 1. For illustration purpose, we give this matrix a name, say A:

```
 0.5155        0        0        0
 0.4863  -0.2155        0        0
 0.3110  -0.6576   0.4121        0
-0.9363  -0.4462  -0.9077  -0.8057
```

Note that the probability of an element being exactly 1 or -1 is zero. However, since Matlab only works with a certain number of decimals, the probability is extremely low, but still positive. The code takes this problem into account by generating another number if 1 or -1 is generated.

Calculate $B = AA^T$ to obtain symmetry.

```
 0.2657   0.2507   0.1603  -0.4827
 0.2507   0.2829   0.2930  -0.3591
 0.1603   0.2930   0.6990  -0.3718
-0.4827  -0.3591  -0.3718   2.5488
```

Normalize matrix B by making the following transformation of each element (Matlab):

```
C(i,j)=B(i,j)/(sqrt(B(i,i))*sqrt(B(j,j)));
```

where *i* is a row index and *j* is a column index.

In fact, matrix C is a random correlation matrix (positive definite):

1.0000	0.9142	0.3719	-0.5865
0.9142	1.0000	0.6588	-0.4229
0.3719	0.6588	1.0000	-0.2785
-0.5865	-0.4229	-0.2785	1.0000

2. Simulate Standard Deviations of returns.
We assume that the Standard Deviations of returns are uniformly distributed between 0,005 and 0,8. The Standard Devations of different subportfolios are independent.

SD:

0.0066	0.0247	0.0222	0.0394

3. Simulate weights invested in each subportfolio.
First simulate a n-dimensional vector with random elements between 0 and 1. Then normalize the vector, so that the elements sum up to 1:

Random vector:	The same vector normalized (elements sum up to 1)
0.9572	0.4014
0.4854	0.2035
0.8003	0.3356
0.1419	0.0595

4. Simulate a series of returns with the correlation structure from step 1 and with v degrees of freedom.
Make a Cholesky decomposition of the correlation matrix from step 1 and simulate the return series as described in section 4.1.2 . By default 10.000 x n returns are simulated.

5. Scale the return series to obtain the Standard Deviation structure from step 2.
We know that each of the simulated return series has a variance $\frac{v}{v-2}$. To obtain the standard deviations generated in step 2, we divide the simulated numbers by $\sqrt{\frac{v}{v-2}}$ and multiply them by the standard deviations from step 2.

6. Set up the allocation problem.
Calculate Value-at-Risk and Expected Shortfall with different values of α.
Create the coalition matrix S by using my function **S=coalitions(n)**.

7. Check if the core exists.

Use my function **CC=core_existence(S,rm)** for different values of α. If the optimal solution to the LP problem can't be found, rerun the LP problem at most 2 times. If the solution is still not found, return the error code.

8. Allocate the costs by selected allocations methods.

9. Check if the solutions from step 8 are core-compatible. Save the result in a binary variable.

10. Repeat the steps 1-9 a number of times (10 000 by default).

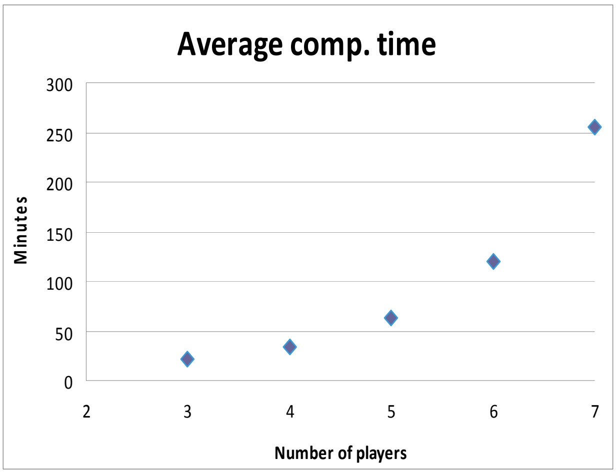
11. Repeat steps 1-10 for different values of v (degrees of freedom) and n (number of players).

6.1 Simulation results

It took more than 32 hours to run the program on the following machine:

Computer:
Intel(R) Core(TM)
i5-2400S CPU @ 2.50GHz
2.49 GHz, 3,16 GB RAM

```
Elapsed time is 117305.593056 seconds.
```



The long computation time is not surprising, since the problem itself is very demanding. While changes in the values of α or degrees of freedom don't have any significant effects on the computation time, the number of players does, as the size of the problem itself gets affected directly. A larger number of players means larger Cholesky matrix, more required simulations and more possible coalitions.

In order to evaluate the core compatibility of the allocation rules we first need to check in how many cases the core itself existed. If the core didn't exist, then it would not be fair to blame the allocation rules for not being able to make a core-compatible solution. Because we use a Student-t distribution and the smallest value of v is 3, the variance is always defined and VaR and ES should be coherent. However, because we use a Monte-Carlo approach the distribution becomes discrete and subadditivity might be a problem with a small number of simulations. As the number of simulations increases, the number of cases where subadditivity is violated by either VaR or ES decreases.

The results show that both Expected Shortfall and Value-at-Risk had a few cases where the function **CC=core_existence(S,rm)** returned an error code. This means that Matlab could not find an optimal solution in these cases, i.e. we do not know if the core existed or not. However, when we only focus on the cases where the optimal solution was found, we can see that sometimes Value-at-Risk did result in an allocation problem with an empty core.

Number of cases where the core of the game was empty when using Value-at-Risk as a risk measure:

	3 degrees of freedom			
	alpha=1%	alpha=5%	alpha=9%	alpha=13%
3 players	0,0446	0,0233	0,0203	0,0176
4 players	0,0477	0,0206	0,0176	0,0142
5 players	0,0541	0,0211	0,0157	0,0153
6 players	0,062	0,02	0,0164	0,0159
7 players	0,0635	0,0178	0,016	0,0151

	5 degrees of freedom			
	alpha=1%	alpha=5%	alpha=9%	alpha=13%
3 players	0,0433	0,0208	0,0196	0,0198
4 players	0,0362	0,0203	0,0165	0,0166
5 players	0,0434	0,0181	0,0142	0,0167
6 players	0,0382	0,0139	0,0131	0,0125
7 players	0,0453	0,0159	0,0142	0,0145

	7 degrees of freedom			
	alpha=1%	alpha=5%	alpha=9%	alpha=13%
3 players	0,0328	0,0226	0,0181	0,0192
4 players	0,0348	0,0173	0,0158	0,015
5 players	0,0341	0,0163	0,014	0,0154
6 players	0,031	0,0139	0,0153	0,0148
7 players	0,033	0,0158	0,0121	0,0142

	9 degrees of freedom			
	alpha=1%	alpha=5%	alpha=9%	alpha=13%
3 players	0,0353	0,0214	0,0208	0,0188
4 players	0,0353	0,0188	0,0185	0,017
5 players	0,0334	0,0152	0,0136	0,0163
6 players	0,0305	0,0139	0,014	0,0138
7 players	0,0319	0,0125	0,0127	0,0116

We see that the numbers are relatively low (below 5%), which means that core is empty in a relatively few cases. We also see that the numbers are higher when alpha is 1%. Since VaR is just a percentile, low alpha means that the risk is based on a few numbers far away in the left tail and hence the result is not surprising. The larger number of simulations might improve (lower) the numbers in the table above.
When using the Expected Shortfall as a risk measure, there were no generated allocation problems with an empty core.

The two tables below show the core-compatibility ratios of the five allocation rules for different values of alpha (the degree of confidence of the two risk measures), ν (degrees of freedom of the Student-t distribution) and n (number of players). Note that the VaR numbers are adjusted for the cases when the game had an empty core. For example, if a certain allocation rule had a core-compatibility ratio of 0.5 and the core did not exist in 5% of the cases, then the core compatibility ratio of that allocation rule gets corrected to $\frac{0.5}{(1-0.05)} \approx 0.5263$.

Note that because the simulation approach is used it can be difficult to say whether some differences are caused by small errors or whether there is an explanation to those differences. Further investigation of this can be a great topic for further research. With a more powerful computer, the simulation could be rerun with a larger number of simulations.

Core compatibility simulation results. Risk measure: Expected Shortfall

		3 degrees of freedom				5 degrees of freedom				7 degrees of freedom				9 degrees of freedom			
		alpha=1%	**alpha=5%**	**alpha=9%**	**alpha=13%**	**alpha=1%**	**alpha=5%**	**alpha=9%**	**alpha=13%**	**alpha=1%**	**alpha=5%**	**alpha=9%**	**alpha=13%**	**alpha=1%**	**alpha=5%**	**alpha=9%**	**alpha=13%**
	Activity	0,3665	0,3736	0,3767	0,3739	0,3755	0,3774	0,3748	0,377	0,3863	0,3817	0,378	0,3842	0,3753	0,3787	0,3818	0,3789
	Incremental	0,1915	0,1979	0,2012	0,2044	0,1958	0,2001	0,1969	0,2031	0,1916	0,1976	0,1995	0,2007	0,1967	0,2028	0,2012	0,2008
3 players	**Cost-Gap**	0,7493	0,7524	0,751	0,7564	0,7541	0,7556	0,7495	0,7588	0,7533	0,7461	0,7472	0,7464	0,7547	0,7493	0,747	0,7552
	Beta	0,4903	0,6255	0,6252	0,5982	0,4742	0,6229	0,6798	0,6932	0,5038	0,6358	0,6844	0,7061	0,5151	0,6389	0,6765	0,7033
	Shapley	0,6747	0,6898	0,6917	0,6954	0,6648	0,6843	0,692	0,6926	0,676	0,6925	0,6963	0,6978	0,6723	0,682	0,6904	0,6872
	Activity	0,1945	0,1912	0,1904	0,1896	0,1884	0,1886	0,1907	0,1888	0,1932	0,19	0,1883	0,1881	0,19	0,1927	0,1872	0,1914
	Incremental	0,0793	0,0845	0,0848	0,0839	0,079	0,0809	0,0824	0,0821	0,0773	0,0799	0,0795	0,0814	0,0772	0,0809	0,0819	0,0861
4 players	**Cost-Gap**	0,7136	0,7109	0,7157	0,71	0,7112	0,7161	0,71	0,7116	0,7066	0,7203	0,7169	0,7232	0,7166	0,7204	0,7207	0,725
	Beta	0,3779	0,5435	0,5578	0,524	0,3772	0,561	0,6488	0,6484	0,3962	0,5688	0,6482	0,6593	0,4312	0,5844	0,6482	0,665
	Shapley	0,4792	0,4896	0,4844	0,4889	0,4692	0,4854	0,4907	0,4864	0,4727	0,4849	0,4841	0,4917	0,4704	0,4801	0,4857	0,4861
	Activity	0,099	0,0995	0,0986	0,0985	0,1033	0,0999	0,0998	0,0997	0,1069	0,1016	0,1009	0,1003	0,1032	0,0981	0,1012	0,1003
	Incremental	0,0295	0,0342	0,0331	0,0349	0,0322	0,0339	0,0353	0,0351	0,0323	0,0339	0,0358	0,0371	0,0317	0,0325	0,0349	0,0351
5 players	**Cost-Gap**	0,6415	0,6497	0,6617	0,6553	0,6343	0,6467	0,6465	0,6527	0,6435	0,6472	0,6596	0,6567	0,6506	0,6593	0,6467	0,6533
	Beta	0,2952	0,4725	0,4904	0,4575	0,2993	0,4967	0,5857	0,6082	0,3281	0,5114	0,5844	0,6188	0,3685	0,5331	0,592	0,6269
	Shapley	0,2655	0,2766	0,274	0,2795	0,2573	0,2718	0,2678	0,2695	0,2718	0,2734	0,272	0,2799	0,269	0,276	0,2767	0,2709
	Activity	0,0588	0,053	0,0563	0,0578	0,063	0,0615	0,058	0,0594	0,0556	0,0546	0,0563	0,0567	0,0579	0,0551	0,0555	0,0558
	Incremental	0,0138	0,015	0,0148	0,0151	0,0142	0,0139	0,0162	0,0176	0,0138	0,0158	0,0172	0,0178	0,0138	0,0154	0,0157	0,0171
6 players	**Cost-Gap**	0,5683	0,5869	0,5967	0,592	0,5675	0,5844	0,5851	0,5905	0,5727	0,5763	0,5861	0,5815	0,5724	0,5927	0,5869	0,5889
	Beta	0,2262	0,417	0,4381	0,4033	0,2355	0,4393	0,5336	0,5604	0,2676	0,4624	0,5494	0,5758	0,3086	0,4789	0,5475	0,5822
	Shapley	0,2738	0,292	0,2918	0,2916	0,2739	0,2882	0,2897	0,2863	0,2716	0,2829	0,2799	0,2835	0,2833	0,289	0,2871	0,2884
	Activity	0,0296	0,0302	0,0302	0,028	0,0284	0,0294	0,0297	0,0289	0,0316	0,0311	0,0319	0,0301	0,0335	0,0326	0,0333	0,0316
	Incremental	0,0059	0,0063	0,0076	0,0071	0,0057	0,0075	0,008	0,0084	0,0061	0,0068	0,0087	0,0081	0,0052	0,0071	0,0072	0,0075
7 players	**Cost-Gap**	0,4969	0,5271	0,5278	0,53	0,5023	0,5239	0,5209	0,5237	0,5092	0,5228	0,5306	0,5286	0,5095	0,5236	0,5281	0,5364
	Beta	0,1736	0,3586	0,3928	0,3681	0,1917	0,3945	0,5028	0,5193	0,2207	0,4104	0,4968	0,5364	0,2595	0,4284	0,5015	0,5317
	Shapley	0,1984	0,2099	0,2058	0,2069	0,1982	0,2067	0,2069	0,2083	0,2048	0,2128	0,2111	0,2162	0,2037	0,2084	0,2102	0,215

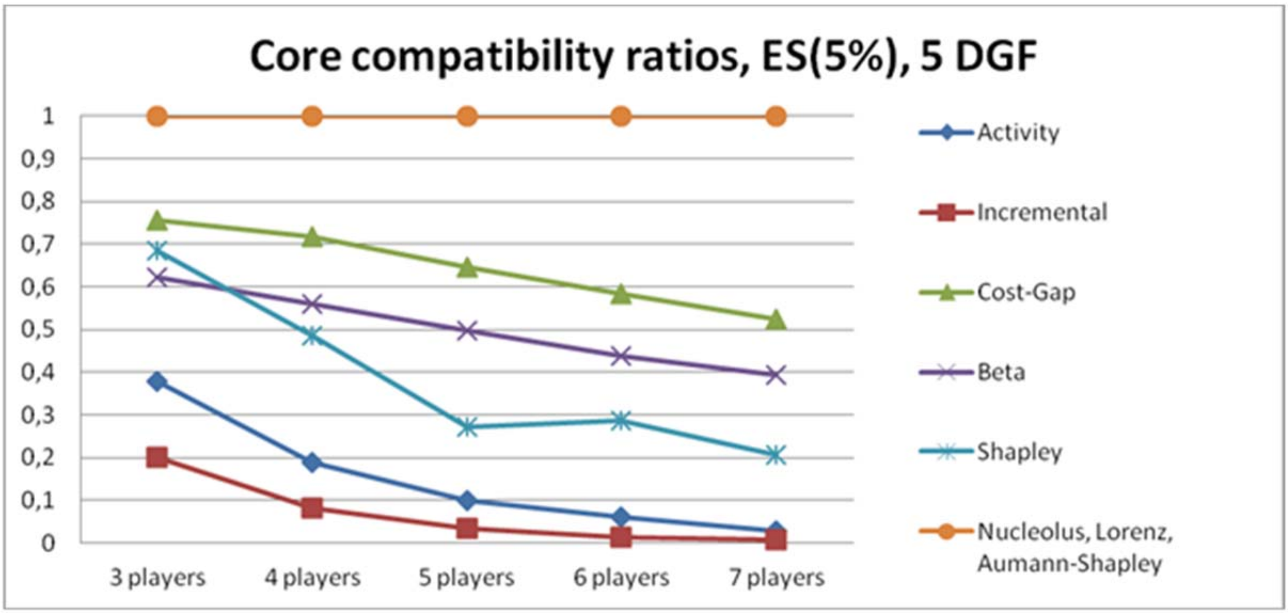
Core compatibility simulation results. Risk measure: Value-at-Risk (adjusted for cases with an empty core)

		3 degrees of freedom				5 degrees of freedom				7 degrees of freedom				9 degrees of freedom			
		alpha=1%	alpha=5%	alpha=9%	alpha=13%	alpha=1%	alpha=5%	alpha=9%	alpha=13%	alpha=1%	alpha=5%	alpha=9%	alpha=13%	alpha=1%	alpha=5%	alpha=9%	alpha=13%
	Activity	0,3888	0,3990	0,3848	0,3861	0,3885	0,3924	0,3921	0,3893	0,3787	0,4015	0,4023	0,3956	0,3861	0,4007	0,3996	0,3939
	Incremental	0,1907	0,2121	0,2124	0,2150	0,1944	0,2024	0,2069	0,2081	0,1935	0,2073	0,2085	0,2090	0,1993	0,2050	0,2147	0,2129
3 players	Cost-Gap	0,7489	0,7905	0,7895	0,7928	0,7553	0,7836	0,7871	0,7871	0,7571	0,7936	0,7925	0,7881	0,7513	0,7870	0,7828	0,7860
	Beta	0,5093	0,5181	0,4664	0,4269	0,5221	0,5862	0,5628	0,5423	0,5331	0,5999	0,5798	0,5689	0,5408	0,5898	0,5996	0,5818
	Shapley	0,7021	0,7143	0,7074	0,7061	0,7006	0,7068	0,7089	0,7073	0,7018	0,7161	0,7092	0,7157	0,6948	0,7064	0,7060	0,7100
	Activity	0,2040	0,1960	0,1899	0,1883	0,1955	0,1974	0,1952	0,1954	0,1933	0,1920	0,1943	0,1918	0,1964	0,1977	0,1981	0,1939
	Incremental	0,0701	0,0799	0,0879	0,0839	0,0728	0,0800	0,0840	0,0837	0,0741	0,0816	0,0849	0,0809	0,0761	0,0807	0,0817	0,0820
4 players	Cost-Gap	0,7070	0,7497	0,7566	0,7400	0,7069	0,7540	0,7555	0,7529	0,7071	0,7554	0,7499	0,7549	0,7056	0,7462	0,7481	0,7597
	Beta	0,3938	0,4186	0,3735	0,3284	0,4234	0,4925	0,4752	0,4451	0,4317	0,4984	0,4895	0,4780	0,4382	0,4958	0,5047	0,4903
	Shapley	0,4949	0,5068	0,5007	0,4919	0,4847	0,4981	0,4998	0,4973	0,4896	0,5018	0,4973	0,4943	0,4898	0,4933	0,4989	0,4966
	Activity	0,1004	0,1000	0,0951	0,0931	0,1018	0,1005	0,1019	0,1019	0,1062	0,1036	0,1025	0,1034	0,0986	0,1003	0,1009	0,0998
	Incremental	0,0259	0,0314	0,0320	0,0341	0,0284	0,0332	0,0320	0,0311	0,0287	0,0334	0,0342	0,0351	0,0306	0,0336	0,0338	0,0351
5 players	Cost-Gap	0,6404	0,6923	0,6830	0,6929	0,6424	0,6884	0,6894	0,6868	0,6428	0,6781	0,6816	0,6811	0,6461	0,6896	0,6837	0,6831
	Beta	0,3087	0,3261	0,2836	0,2616	0,3383	0,4108	0,3931	0,3732	0,3483	0,4156	0,4075	0,3923	0,3460	0,4140	0,4270	0,4072
	Shapley	0,2802	0,2850	0,2813	0,2829	0,2709	0,2813	0,2849	0,2876	0,2757	0,2937	0,2891	0,2886	0,2767	0,2921	0,2909	0,2878
	Activity	0,0603	0,0594	0,0551	0,0568	0,0605	0,0585	0,0565	0,0575	0,0558	0,0533	0,0542	0,0541	0,0557	0,0560	0,0542	0,0568
	Incremental	0,0103	0,0133	0,0135	0,0132	0,0117	0,0146	0,0155	0,0148	0,0120	0,0143	0,0138	0,0150	0,0119	0,0124	0,0152	0,0144
6 players	Cost-Gap	0,5711	0,6222	0,6224	0,6241	0,5784	0,6164	0,6157	0,6172	0,5724	0,6120	0,6221	0,6163	0,5783	0,6245	0,6227	0,6173
	Beta	0,2325	0,2638	0,2269	0,1954	0,2618	0,3301	0,3127	0,2994	0,2747	0,3529	0,3507	0,3262	0,2892	0,3497	0,3525	0,3355
	Shapley	0,2891	0,2854	0,2825	0,2742	0,2725	0,2863	0,2823	0,2831	0,2736	0,2852	0,2802	0,2830	0,2819	0,2839	0,2861	0,2859
	Activity	0,0298	0,0304	0,0285	0,0266	0,0302	0,0284	0,0285	0,0287	0,0315	0,0307	0,0305	0,0296	0,0315	0,0340	0,0318	0,0334
	Incremental	0,0046	0,0054	0,0062	0,0048	0,0038	0,0060	0,0065	0,0068	0,0040	0,0060	0,0052	0,0062	0,0037	0,0058	0,0064	0,0067
7 players	Cost-Gap	0,5034	0,5553	0,5554	0,5521	0,5012	0,5483	0,5509	0,5586	0,5034	0,5536	0,5513	0,5570	0,5117	0,5484	0,5515	0,5499
	Beta	0,1816	0,2108	0,1864	0,1607	0,2109	0,2758	0,2596	0,2443	0,2198	0,2835	0,2736	0,2623	0,2247	0,2833	0,2843	0,2796
	Shapley	0,2014	0,1992	0,2022	0,1963	0,1923	0,1983	0,1983	0,2022	0,2024	0,2018	0,2022	0,2071	0,2031	0,2110	0,2068	0,2095

The following can be concluded:

- The Beta method is the only method that seems to be sensitive to changes in alpha. When alpha is set at 1% level, the core compatibity ratio of the Beta method is lower. On the other hand, the core compatibility of the other methods seem to be unaffected by the changes in the level of alpha.
- The core compatibility ratio of all allocation methods but the Beta method seems to be unaffected by the thickness of the tails (degrees of freedom).
- The results when using Expected Shortfall as a risk measure are in most cases very close to results when using Value-at-Risk.
- The performance of different allocation methods worsens as the number of players rises

The clear pattern that we see is that the core compatibility of the allocation rules decreases significantly as the number of players increases. This is not surprising, as the core itself shrinks when the number of players increases (more inequalities have to be satisfied).



Not surprisingly the more advanced game-theoretic methods have a better performance from the perspective of core compatibility. The most surprising fact in this graph is that such an "exotic" allocation rule as Cost-Gap method has a relatively high core compatibility ratio, for example compared to the very popular Shapley method.

To answer the main question of this section we could argue that the performance of the Shapley method measured by core compatibility ratio is not satisfying, especially for many players. Since the Shapley method is the only method that satisfies Symmetry and Strong Monotonicity, its low ratio of the Core Compatibility means that in practical applications we will have to give up one of the properties.

Conclusion

This thesis demonstrates the basics of the risk management, of the Cooperative Game Theory and creates the link between these two separate areas in order to allocate risk capital to the different subunits. Two different risk measures were used, namely Value-at-Risk and Expected Shortfall. Despite the fact that the primer is the risk measure of choice in the financial industry, the latter has better theoretical properties. The main issue with VaR is lack of Subadditivity, which might be a practical problem. If the capital requirement of each subunit is dimensioned on its own risk, the regulator should be confident that also the overall company capital should be an adequate one. Lack of Subadditivity may thus result in non-sense results when combining portfolios together, which later may result in non-sense allocations of the risk capital. The simulation in chapter 6 showed that when using VaR as a risk measure, the core does not exist in 1% to 5% of the cases depending on the number of players and choice of α. On the other hand, even though Expected Shortfall doesn't always satisfy Subadditivity when the distribution is discrete, there were no cases in the simulation, where the core didn't exist, which makes lack of Subadditivity of ES only a theoretical problem. This is very important, as the allocation rules can only be coherent when the risk measure used is coherent.

The thesis shows that when the risk measure used is coherent, the corresponding risk allocation problem has a non-empty core. Hence, there will always be at least one allocation where the subunits are willing to cooperate and benefit by being a part of the same company. Eight different allocation rules were presented, namely Activity, Incremental, Beta, Cost Gap, Nucleolus, Shapley, Lorenz and Aumann-Shapley. Unfortunately, the perfect allocation rule was not found, as it does not exist. Among others, there is the well-known trade-off between Core Compatibility and Strong Monotonicity. As such, one will have to choose which properties are most important for the actual setting and select an allocation method based on this information.

Different allocation rules use very different approaches, but the main idea is to share the diversification benefit fairly and/or easily. However, if there is no or not much benefit, as in case with high positive correlation, there is no reason for a company to spend money on applying one of the advanced allocation rules, because for example, Activity method, would be able to share the risk capital fairly enough. On the other hand, if there is a subunit which is very advantageous for the other subunits, the Activity method should not be used, because the benefit that particular subunit brings by being a part of the coalition will be ignored.

When there is a good possibility of splitting and/or merging the subunits, one should consider the Activity, Beta and Aumann-Shapley, as these are the only methods where merging and splitting is never advantageous from the perspective of the allocated risk capital. One should be aware of using the Incremental method. Even though it is obvious and often gives fair allocations, sometimes it may result in extreme non-sense allocations, as was demonstrated in specific examples in this thesis. The only property it satisfies is Symmetry, which is satisfied by most other allocation rules. Additionally, Incremental's very low core compatibility ratio makes it one the worst allocation rules presented in this thesis.

Lorenz method should not be used in situations where the amount invested in portfolios can easily be scaled up and down, as it will seek to allocate the extra risk created by larger invested amount in a certain subportfolio as equally as possible between all the subportfolios. Additionally, because the risk allocation

problem when using a coherent risk measure is not concave, unless the case with no diversification benefit, the Lorenz allocation is not always unique.
Beta method uses a very alternative approach to allocation of the risk capital. It requires calculation of the covariances rather than stand alone risks. The method satisfies only one of the desired properties and its core compatibility ratio is not satisfying, especially for many players. The Beta method will always allocate 0 risk capital to any subunit with a riskless portfolio, regardless of the amount invested in that portfolio. The method can be chosen if the covariances are much easier to calculate and the company can thus save a lot of money on calculations. Otherwise, choosing a method from Game Theory would be a better choice.
The best allocation rules when assuming atomic players are Shapley and Nucleolus, as they have the best theoretical properties. The main problem with Nucleolus is its practical application, especially when the problem is large, as the LP problems might be difficult to solve, even for a computer. Different programs use different approach to find the optimum of a LP problem. But since there is no perfect algorithm, there may be some small errors when finding the optimum and hence small errors in the Nucleolus allocation itself. On the other hand, because Shapley is calculated by referring to specific elements of the input data, the allocation is always exact, no matter how large the problem is. Additionally the Shapley method is the only method that satisfies Strong Monotonicity, Symmetry and Pareto Optimality at the same time. However, the main problem with Shapley is lack of core-compatibility even when used on problems with non-empty core. In order to check if it is also a practical problem, a large simulation work was done to see to what degree the core is violated by the different allocation rules that are not core-compatible. Unfortunately, the performance of the Shapley method was not satisfying, especially for many players. As previously stated, this means that a perfect allocation rule does not exist and in practice one will have to give up some of the desired axioms. The simulation also showed that such an "exotic" allocation rule as Cost-Gap in practice shows relatively high core-compatibility ratio, for example compared to the very famous Shapley.
When players are portfolios, assumption about fractional players becomes natural. Here Aumann-Shapley method was considered as the only candidate. Fortunately, it turned out that in case with a positively homogeneous risk measure, the Aumann-Shapley calculation simplifies significantly. Additionally, when the risk measure used is coherent, the fuzzy core consists of one element only, namely the Aumann-Shapley allocation, which makes it the only coherent allocation rule when assuming fractional players. However, note that Aumann-Shapley is not recommended when the risk measure used is Value-at-Risk, as the allocation will be based on one specific event in the history, namely where the grandportfolio has its certain percentile. On the other hand, Aumann-Shapley in combination with Expected Shortfall showed a good performance. However, one should remember that in very rare situations Aumann-Shapley allocation might not be defined when using historical distribution.
Finally it is important to mention that that all eight allocation rules were programmed in Matlab. This means that the risk capital can be allocated even when the problem gets large. Therefore the trade-off between the theoretical properties and the practical application, which gets a lot of attention in the academic literature, is no longer relevant. All the allocation rules can be applied equally simply by pressing the "Run" button. In practice one will be able to choose the allocation rule whose theoretical properties best fits the actual setting without speculation about the practical application.

References

Books

[1] Jens Leth Hougaard (2009). An introduction to Allocation Rules.

[2] Jón Daníelsson (2011). Financial Risk Forecasting.

[3] Alexander J. McNeil, Rüdiger Frey, Paul Embrechts (2005). Quantitative Risk Management.

[4] E.N.Barron (2008). Game Theory: An introduction

[5] R.J. Aumann, L.S. Shapley (1974). Values of Non-Atomic Games.

Articles

[6] Acerbi, C., & Tasche, D. (2002). Expected shortfall: A natural coherent alternative to value at risk. *Economic Notes, 31*(2), 379-388.

[7] Artzner, P., Delbaen, F., Eber, J. M., & Heath, D. (1999). Coherent measures of risk. *Mathematical Finance, 9*(3), 203-228.

[8] Aubin, J. P. (1979). Mathematical methods of game and economic theory. *NORTH-HOLLAND PUBL.CO., N.Y., 1979, 620.*

[9] Aubin, J. P. (1981). Cooperative fuzzy games. *Mathematics of Operations Research,* 1-13.

[10] Balog, D. (2011). Capital allocation in financial institutions: the Euler method

[11] Balog, D., Bátyi, T. L., Csóka, P., & Pintér, M. (2011) Properties of risk capital allocation methods.

[12] Csóka, P., Herings, P., & Kóczy, L. Á. (2009). Stable allocations of risk. *Games and Economic Behavior, 67*(1), 266-276.

[13] Csóka, P., & Pintér, M. (2010). On the impossibility of fair risk allocation.

[14] Denault, M. (2001). Coherent allocation of risk capital. *Journal of Risk, 4*, 1-34.

[15] Dhaene, J., Tsanakas, A., Valdez, E., & Vanduffel, S. (2005). Optimal capital allocation principles. *9th International Congress on Insurance: Mathematics and Economics,* pp. 6-8.

[16] Dutta, B., & Ray, D. (1989). A concept of egalitarianism under participation constraints. *Econometrica: Journal of the Econometric Society,* 615-635.

[17] Gulick, G., & Norde, H. (2011). Fuzzy cores and fuzzy balancedness. *Discussion Paper,*

[18] Hamlen, S. S., Hamlen Jr, W. A., & Tschirhart, J. T. (1977). The use of core theory in evaluating joint cost allocation schemes. *Accounting Review,* 616-627.

[19] Herings, P., Csóka, P., & Koczy, L. (2007). Stable allocations of risk.

[20] Hesselager, O., Andersson, U., & Insurance, T. (2002). Risk sharing and capital allocation. *Denmark, Tryg Insurance.*

[21] Jorion, P. (2007). Value at risk: The new benchmark for managing financial risk.

[22] Kalkbrener, M. (2005). An axiomatic approach to capital allocation. *Mathematical Finance, 15*(3), 425-437.

[23] Panjer, H. H. (2001). *Measureement of risk, solvency requirements and allocation of capital within financial conglomerates* University of Waterloo, Institute of Insurance and Pension Research.

[24] Patrik, G., Bernegger, S., & Rüegg, M. B. (1999). THE USE OF RISK ADJUSTED CAPITAL TO SUPPORT BUSINESS DECISION MAKING.

[25] Paul Kaye (2005). A Guide To Risk Management, Capital Allocation And Related Decision Support Issues.

[26] Schmeidler, D. (1969). The nucleolus of a characteristic function game. *SIAM Journal on Applied Mathematics, 17*(6), 1163-1170.

[27] Shapley, L.S. (1953). A value for n-person games. *Ann Math Stud 28:307-318.*

[28] Shapley, L. S. (1967). On balanced sets and cores. *Naval Research Logistics Quarterly, 14*(4), 453-460.

[29] Shapley, L. S. (1971). Cores of convex games. *International Journal of Game Theory, 1*(1), 11-26.

[30] Tasche, D. (1999). Risk contributions and performance measurement. *Report of the Lehrstuhl Für Mathematische Statistik, TU München,*

[31] Tasche, D. (2002). Expected shortfall and beyond. *Journal of Banking & Finance, 26*(7), 1519-1533.

[32] Tasche, D. (2007). Euler allocation: Theory and practice. *Preprint, Fitch Ratings, London,*

[33] Tijs, S. H., & Driessen, T. S. H. (1986). Game theory and cost allocation problems. *Management Science,* , 1015-1028.

[34] Tsanakas, A. (2009). To split or not to split: Capital allocation with convex risk measures. *Insurance: Mathematics and Economics, 44*(2), 268-277.

[35] Tsanakas, A., & Barnett, C. (2003). Risk capital allocation and cooperative pricing of insurance liabilities. *Insurance: Mathematics and Economics, 33*(2), 239-254.

[36] Van Gulick, G., De Waegenaere, A., & Norde, H. (2011). Excess based allocation of risk capital. *Insurance: Mathematics and Economics,*

[37] Young, H. P. (1985). Monotonic solutions of cooperative games. *International Journal of Game Theory, 14*(2), 65-72.

Notes

[38] *George Foulds (2011). Portfolio Optimisation and Student's t Distribution.*

[39] Martin Haugh (2004). The Monte Carlo Framework, Examples from Finance and Generating Correlated Random Variables. *Columbia University.*

[40] Edward Neuman. Linear Programming with Matlab. *Southern Illinois University at Carbondale.*

Picture on the front page is taken from https://dentaldirectsales.com.

Appendix 1. Example with Subadditivity and Concavity checking

Consider the following coalitional game with 3 players (n=3):

Coalition S	**Cost c(S)**
{1}	5
{2}	6
{3}	2
{1,2}	9
{1,3}	6
{2,3}	5
{1,2,3}	10

As expected, there are $2^3 - 1 = 7$ *possible coalitions.*

Is the game subadditive? To answer this question, we have to check all possible coalitions S and T with an empty intersection:

S	T	c(S)	c(T)	$S \cup T$	$c(S \cup T)$	*c(S)+c(T)*	Is subadditivity violated (is column 6 greater than column 7)?
{1}	{2}	5	6	{1,2}	9	11	no
{1}	{3}	5	2	{1,3}	6	7	no
{2}	{3}	6	2	{2,3}	5	8	no
{1,2}	{3}	9	2	{1,2,3}	10	11	no
{1,3}	{2}	6	6	{1,2,3}	10	12	no
{2,3}	{1}	5	5	{1,2,3}	10	10	no

Answer: yes, the game is subadditive.

Is the game concave? To answer this question, we have to check all possible coalitions S and T.

S	T	c(S)	c(T)	$S \cup T$	c($S \cup T$)	$S \cap T$	c($S \cap T$)	c($S \cup T$)+ c($S \cap T$)	c(S)+c(T)	Is concavity violated (is column 6 greater than column 7)?
{1}	{2}	5	6	{1,2}	9	{Ø}	0	9	11	no
{1}	{3}	5	2	{1,3}	6	{Ø}	0	6	7	no
{1}	{1,2}	5	9	{1,2}	9	{1}	5	14	14	no
{1}	{1,3}	5	6	{1,3}	6	{1}	5	11	11	no
{1}	{2,3}	5	5	{1,2,3}	10	{Ø}	0	10	10	no
{1}	{1,2,3}	5	10	{1,2,3}	10	{Ø}	0	10	15	no
{2}	{3}	6	2	{2,3}	5	{Ø}	0	5	8	no
{2}	{1,2}	6	9	{1,2}	9	{2}	6	15	15	no
{2}	{1,3}	6	6	{1,2,3}	10	{Ø}	0	10	12	no
{2}	{2,3}	6	5	{2,3}	5	{2}	6	11	11	no
{2}	{1,2,3}	6	10	{1,2,3}	10	{2}	6	16	16	no
{3}	{1,2}	2	9	{1,2,3}	10	{Ø}	0	10	11	no
{3}	{1,3}	2	6	{1,3}	6	{3}	2	8	8	no
{3}	{2,3}	2	5	{2,3}	5	{3}	2	7	7	no
{3}	{1,2,3}	2	10	{1,2,3}	10	{3}	2	12	12	no
{1,2}	{1,3}	9	6	{1,2,3}	10	{1}	5	15	15	no
{1,2}	{2,3}	9	5	{1,2,3}	10	{2}	6	16	14	yes
{1,2}	{1,2,3}	9	10	{1,2,3}	10	{1,2}	9	19	19	no
{1,3}	{2,3}	6	5	{1,2,3}	10	{3}	2	12	11	yes
{1,3}	{1,2,3}	6	10	{1,2,3}	10	{1,3}	6	16	16	no
{2,3}	{1,2,3}	5	10	{1,2,3}	10	{2,3}	5	15	15	no

Answer: no, the game is not concave, since concavity is violated for some sets of S and T.
The core of the game is defined as follows:

$$K_1 \leq 5$$
$$K_2 \leq 6$$
$$K_3 \leq 2$$
$$K_1 + K_2 \leq 9$$
$$K_1 + K_3 \leq 6$$
$$K_2 + K_3 \leq 5$$
$$K_1 + K_2 + K_3 = 10$$

One can check that the above linear inequalities and the linear equality have a unique solution. Hence, in this case the core consists of only one element, namely $(K_1, K_2, K_3) = (5,4,1)$.

Appendix 2. Example 3.1.1 - Concavity

S	T	c(S)	c(T)	$S \cup T$	c($S \cup T$)	$S \cap T$	c($S \cap T$)	c($S \cap T$)+ c($S \cup T$)	c(S)+c(T)	Violation of concavity?
{1}	{2}	15	14	{12}	23	{Ø}	0	23	29	no
{1}	{3}	15	16	{13}	22	{Ø}	0	22	31	no
{1}	{4}	15	15	{14}	21	{Ø}	0	21	30	no
{1}	{12}	15	23	{12}	23	{1}	15	38	38	no
{1}	{13}	15	22	{13}	22	{1}	15	37	37	no
{1}	{14}	15	21	{14}	21	{1}	15	36	36	no
{1}	{23}	15	25	{123}	28	{Ø}	0	28	40	no
{1}	{24}	15	23	{124}	29	{Ø}	0	29	38	no
{1}	{34}	15	24	{134}	27	{Ø}	0	27	39	no
{1}	{123}	15	28	{123}	28	{1}	15	43	43	no
{1}	{124}	15	29	{124}	29	{1}	15	44	44	no
{1}	{134}	15	27	{134}	27	{1}	15	42	42	no
{1}	{234}	15	29	{1234}	32	{Ø}	0	32	44	no
{1}	{1234}	15	32	{1234}	32	{1}	15	47	47	no
{2}	{3}	14	16	{23}	25	{Ø}	0	25	30	no
{2}	{4}	14	15	{24}	23	{Ø}	0	23	29	no
{2}	{12}	14	23	{12}	23	{2}	14	37	37	no
{2}	{13}	14	22	{123}	28	{Ø}	0	28	36	no
{2}	{14}	14	21	{124}	29	{Ø}	0	29	35	no
{2}	{23}	14	25	{23}	25	{2}	14	39	39	no
{2}	{24}	14	23	{24}	23	{2}	14	37	37	no
{2}	{34}	14	24	{234}	29	{Ø}	0	29	38	no
{2}	{123}	14	28	{123}	28	{2}	14	42	42	no
{2}	{124}	14	29	{124}	29	{2}	14	43	43	no
{2}	{134}	14	27	{1234}	32	{Ø}	0	32	41	no
{2}	{234}	14	29	{234}	29	{2}	14	43	43	no
{2}	{1234}	14	32	{1234}	32	{2}	14	46	46	no
{3}	{4}	16	15	{34}	24	{Ø}	0	24	31	no
{3}	{12}	16	23	{123}	28	{Ø}	0	28	39	no
{3}	{13}	16	22	{13}	22	{3}	16	38	38	no
{3}	{14}	16	21	{134}	27	{Ø}	0	27	37	no
{3}	{23}	16	25	{23}	25	{3}	16	41	41	no
{3}	{24}	16	23	{234}	29	{Ø}	0	29	39	no
{3}	{34}	16	24	{34}	24	{3}	16	40	40	no
{3}	{123}	16	28	{123}	28	{3}	16	44	44	no
{3}	{124}	16	29	{1234}	32	{Ø}	0	32	45	no
{3}	{134}	16	27	{134}	27	{3}	16	43	43	no
{3}	{234}	16	29	{234}	29	{3}	16	45	45	no
{3}	{1234}	16	32	{1234}	32	{3}	16	48	48	no
{4}	{12}	15	23	{124}	29	{Ø}	0	29	38	no
{4}	{13}	15	22	{134}	27	{Ø}	0	27	37	no
{4}	{14}	15	21	{14}	21	{4}	15	36	36	no
{4}	{23}	15	25	{234}	29	{Ø}	0	29	40	no

{4}	{24}	15	23	{24}	23	{4}	15	38	38	no
{4}	{34}	15	24	{34}	24	{4}	15	39	39	no
{4}	{123}	15	28	{1234}	32	{Ø}	0	32	43	no
{4}	{124}	15	29	{124}	29	{4}	15	44	44	no
{4}	{134}	15	27	{134}	27	{4}	15	42	42	no
{4}	{234}	15	29	{234}	29	{4}	15	44	44	no
{4}	{1234}	15	32	{1234}	32	{Ø}	0	32	47	no
{12}	{13}	23	22	{123}	28	{1}	15	43	45	no
{12}	{14}	23	21	{124}	29	{1}	15	44	44	no
{12}	{23}	23	25	{123}	28	{2}	14	42	48	no
{12}	{24}	23	23	{124}	29	{2}	14	43	46	no
{12}	{34}	23	24	{1234}	32	{Ø}	0	32	47	no
{12}	{123}	23	28	{123}	28	{12}	23	51	51	no
{12}	{124}	23	29	{124}	29	{12}	23	52	52	no
{12}	{134}	23	27	{1234}	32	{1}	15	47	50	no
{12}	{234}	23	29	{1234}	32	{2}	14	46	52	no
{12}	{1234}	23	32	{1234}	32	{12}	23	55	55	no
{13}	{14}	22	21	{134}	27	{1}	15	42	43	no
{13}	{23}	22	25	{123}	28	{3}	16	44	47	no
{13}	{24}	22	23	{1234}	32	{Ø}	0	32	45	no
{13}	{34}	22	24	{134}	27	{3}	16	43	46	no
{13}	{123}	22	28	{123}	28	{13}	22	50	50	no
{13}	{124}	22	29	{1234}	32	{1}	15	47	51	no
{13}	{134}	22	27	{134}	27	{13}	22	49	49	no
{13}	{234}	22	29	{123}	28	{3}	16	44	51	no
{13}	{1234}	22	32	{1234}	32	{13}	22	54	54	no
{14}	{23}	21	25	{1234}	32	{Ø}	0	32	46	no
{14}	{24}	21	23	{124}	29	{4}	15	44	44	no
{14}	{34}	21	24	{134}	27	{4}	15	42	45	no
{14}	{123}	21	28	{1234}	32	{1}	15	47	49	no
{14}	{124}	21	29	{124}	29	{14}	21	50	50	no
{14}	{134}	21	27	{134}	27	{14}	21	48	48	no
{14}	{234}	21	29	{1234}	32	{4}	15	47	50	no
{14}	{1234}	21	32	{1234}	32	{14}	21	53	53	no
{23}	{24}	25	23	{234}	29	{2}	14	43	48	no
{23}	{34}	25	24	{234}	29	{3}	16	45	49	no
{23}	{123}	25	28	{123}	28	{23}	25	53	53	no
{23}	{124}	25	29	{1234}	32	{2}	14	46	54	no
{23}	{134}	25	27	{1234}	32	{3}	16	48	52	no
{23}	{234}	25	29	{234}	29	{23}	25	54	54	no
{23}	{1234}	25	32	{1234}	32	{23}	25	57	57	no
{24}	{34}	23	24	{234}	29	{4}	15	44	47	no
{24}	{123}	23	28	{1234}	32	{2}	14	46	51	no
{24}	{124}	23	29	{124}	29	{24}	23	52	52	no
{24}	{134}	23	27	{1234}	32	{Ø}	0	32	50	no
{24}	{234}	23	29	{234}	29	{24}	23	52	52	no
{24}	{1234}	23	32	{1234}	32	{24}	23	55	55	no
{34}	{123}	24	28	{1234}	32	{3}	16	48	52	no
{34}	{124}	24	29	{1234}	32	{4}	15	47	53	no
{34}	{134}	24	27	{134}	27	{34}	24	51	51	no

{34}	{234}	24	29	{234}	29	{34}	24	53	53	no
{34}	{1234}	24	32	{1234}	32	{34}	24	56	56	no
{123}	{124}	28	29	{1234}	32	{12}	23	55	57	no
{123}	{134}	28	27	{1234}	32	{13}	22	54	55	no
{123}	{234}	28	29	{1234}	32	{23}	25	57	57	no
{123}	{1234}	28	32	{1234}	32	{123}	28	60	60	no
{124}	{134}	29	27	{1234}	32	{14}	21	53	56	no
{124}	{234}	29	29	{1234}	32	{24}	23	55	58	no
{124}	{1234}	29	32	{1234}	32	{124}	29	61	61	no
{134}	{234}	27	29	{1234}	32	{34}	24	56	56	no
{134}	{1234}	27	32	{1234}	32	{134}	27	59	59	no
{234}	{1234}	29	32	{1234}	32	{234}	29	61	61	no

Appendix 3. Matlab code

```
function S=coalitions(n)

dummy=1;
sj2=0;
%n = 6;
S=zeros(2^n-1,n);
rk = 0;

for k = 1: n - 1
    rk = rk + 1;
    for t = 1 : k
        S(rk, t) = 1;
    end

    stop1 = 0;
    while (stop1 == 0)
        rk = rk + 1;
        q = n + 1;

        while(dummy==1)
            q = q - 1;
            if S(rk - 1, q) == 1
                sj = q;
                break
            end
        end

        S(rk, :)=S(rk-1,:);

        if sj2 == 0
            S(rk, sj) = 0;
            S(rk, sj + 1) = 1;

        else

            S(rk, sj2) = 0;
            S(rk, sj2 + 1) = 1;

            for u = 1 : antal_ja
                S(rk, n + 1 - u) = 0;
            end

            for u = 1 :antal_ja
                S(rk, sj2 + u + 1) = 1;
            end
        end
```

```
 %are all combinations with k players found?
        stop1 = 1;
        antal_ja = 0;
        for t = n-2 + 2: -1: n + 1 - k
            if S(rk, t) == 0;
                stop1 = 0;
                break
            end
            antal_ja = antal_ja + 1;
        end

        sj2 = 0;

        if stop1 == 0 && antal_ja > 0
            t = n - antal_ja + 1;

            while(dummy==1)
                t = t - 1;
                if S(rk, t) == 1
                    sj2 = t;
                    break
                end
            end
        end
    end
end

%'the last coalition
for p = 1: n
    S(rk + 1, p) = 1;
end
```

```
function CC=core_compatibility(S,cond,allocation,print)

temp=cond-S*allocation;

if abs(temp(end))<0.00001
    temp(end)=0;
end


if count(temp<0)>0
    if print==1
        'Core compatibility is violated in that many conditions:'
        count(cond-S*allocation<0)
    end

    CC=0;
else
    if print==1
        'The allocation is core compatible'
    end
    CC=1;
end
```

```
function CE=core_existence(S,rm)

n=length(S(1,:));

f=ones(1,n);
f=f.*(-1);

options=optimset('display','off');

[x,fval,exitflag]=linprog(f,S(1:end-1,:),rm(1:end-1),[],[],[],[],[],options);
%[x,fval,exitflag]=linprog(f,S,rm);

-fval
rm(end);

if exitflag==1
    if -fval>=rm(end)
        CE=3;
    else
        CE=2;
    end

else
    CE=exitflag;
end
```

Code used in chapter 5. Examples

```
%Input:
%assets={'MSFT','MSFT','MSFT','MSFT','MSFT','MSFT','MSFT','MSFT','MSFT','MSFT'};
%InvAmount=[100,200,300,400,500,600,700,800,900,1000];

assets={'BMW.DE','VOW.DE','PAH3.DE','PEU.DE'};
InvAmount=[100,100,100,-100];

n=length(assets);

start_date = '01042010';
end_date = '01042012';

alpha=0.05;
%--Calculate returns--------

Dates=(datenum(str2num(start_date(5:8)), ...
    str2num(start_date(3:4)),str2num(start_date(1:2))):...
    datenum(str2num(end_date(5:8)),str2num(end_date(3:4)),...
    str2num(end_date(1:2))))';

for i=1:n

    stocks = hist_stock_data(start_date, end_date, char(assets(i)));
    % matlab date conversion
    stocks.Date;
    stocks.Date=datenum(stocks.Date);
    % reverse vectors old -> new
    stocks.Date = stocks.Date(end:-1:1);
    stocks.Close = stocks.Close(end:-1:1);
    stocks.Open = stocks.Open(end:-1:1);
    stocks.High = stocks.High(end:-1:1);
    stocks.Low = stocks.Low(end:-1:1);
    stocks.Volume = stocks.Volume(end:-1:1);
    stocks.AdjClose = stocks.AdjClose(end:-1:1);

    %Delete irrelevant dates and merge data
    if i==1
        Dates=Dates(ismember(Dates,stocks.Date,'rows'),:);
        Prices=horzcat(Dates,stocks.Close);
    else
        stocks.Close=stocks.Close(ismember(stocks.Date,Prices(:,1),'rows'),:);

        Prices=Prices(ismember(Prices(:,1),stocks.Date,'rows'),:);
        Prices=horzcat(Prices,stocks.Close);
    end
end

Prices;
ret = Prices(2:end,2:n+1)./Prices(1:end-1,2:n+1)-1;
```

```
'length'
length(ret)

corr(ret)
%----Create all possible coalitions

S=coalitions(n);
VAR=zeros(2^n-1,1);
ES=zeros(2^n-1,1);

for i=1:2^n-1
    S1=S(i,:);
    XS=ret*(S1.*InvAmount)';
    VAR(i)=-quantile(XS,alpha);
    ES(i)=-mean(XS(XS<=quantile(XS,alpha)));
end


Activity_ES=activity(S,ES);
Incremental_ES=incremental(S,ES);
Cost_gap_ES=cost_gap(S,ES);
Beta_ES=beta(ret,InvAmount,ES);
Shapley_ES=shapley(S,ES);
Nucleolus_ES=nucleolus(S,ES);
Lorenz_ES=lorenz(S,ES);

Activity_VAR=activity(S,VAR);
Incremental_VAR=incremental(S,VAR);
Cost_gap_VAR=cost_gap(S,VAR);
Beta_VAR=beta(ret,InvAmount,VAR);
Shapley_VAR=shapley(S,VAR);
Nucleolus_VAR=nucleolus(S,VAR);
Lorenz_VAR=lorenz(S,VAR);

[AS_VAR,AS_ES]=as(S,ret,InvAmount, alpha);


Solution_ES=[Activity_ES';Incremental_ES';Beta_ES';Cost_gap_ES';Nucleolus_ES';Shapl
ey_ES';Lorenz_ES'; AS_ES']'
sum_ES= sum(Solution_ES)


Solution_VAR=[Activity_VAR';Incremental_VAR';Beta_VAR';Cost_gap_VAR';Nucleolus_VAR'
;Shapley_VAR';Lorenz_VAR';AS_VAR']'
sum_VaR=sum(Solution_VAR)
```

Allocation Rules

```
function Activity=activity(S,rm)
n=length(S(1,:));
Activity=rm(1:n)/sum(rm(1:n)).*rm(end);
```

```
function Incremental=incremental(S,rm)
n=length(S(1,:));

for i=1:n
    Incremental(i)=rm(end)-rm(end-i);
end

Incremental=(Incremental/sum(Incremental).*rm(end))';
```

```
function Beta=beta(ret,InvAmount,rm)
n=length(ret(1,:));

COV=zeros(n,1);

for i=1:n
    COV1=cov(InvAmount(i)*ret(:,i),ret*InvAmount');
    COV(i)=COV1(1,2);
end

COV;

B=COV/var(ret*InvAmount');

Beta=B/sum(B)*rm(end);
```

```
function Cost_gap=cost_gap(S,rm)
n=length(S(1,:));
temp1=zeros(2^n-1,1);
gamma1=zeros(n,1);
Cost_gap=zeros(n,1);

for i=1:length(S(:,1))  %coalitions
    temp=0;
    for j=1:n           %players
        if S(i,j)==1
            temp=temp+rm(end)-rm(end-j);
        end
    end
    temp1(i)=temp;
end

temp2=rm-temp1;

for i=1:n
    index=find(S(:,i)>0);
    gamma1(i)=min(temp2(index));
end


for i=1:n
    Cost_gap(i)=rm(end)-rm(end-i);
end

if sum(gamma1)~=0
    for i=1:n
        Cost_gap(i)=Cost_gap(i)+gamma1(i)/sum(gamma1)*(rm(end)-temp1(end));
    end
end
```

```matlab
function Nucleolus=nucleolus(S,rm)
prec=0.00001;

n=length(S(1,:));

temp1=zeros(2^n-1,1);
temp2=zeros(1,n+2);
temp2(end)=1;
S_eq=ones(1,n+2);
S_eq(end-1:end)=0;
S_eq=vertcat(S_eq,temp2);
S_ineq=horzcat(S,temp1);
S_ineq=S_ineq(1:end-1,:);

S_ineq2=S_ineq;

S_ineq2(:,end)=1;

S_ineq=vertcat(S_ineq,S_ineq2);

S_ineq=horzcat(S_ineq,vertcat(zeros(2^n-2,1),-rm(1:end-1)));

b_eq=vertcat(rm(end),[1]);

f=zeros(1,n+2);
f(end-1)=-1;

b_ineq=vertcat(rm(1:end-1),zeros(2^n-2,1));

options=optimset('display','off','MaxIter',1000);

% f
% S_ineq
% b_ineq
% S_eq
% b_eq
[x,fval,exitflag]=linprog(f,S_ineq,b_ineq,S_eq,b_eq,[],[],[],options);

% x

stop1=0;

while stop1==0
    q=rm-S*x(1:end-2);
    x_copy=x;

    index1=find(abs(q+fval)<prec);

    add_eq=S(index1,:);
    add_eq(:,end+1)=1;
    add_eq=horzcat(add_eq,-rm(index1));
```

```
    S_eq=vertcat(S_eq,add_eq);

    add_b_eq=zeros(length(index1),1);

    b_eq=vertcat(b_eq,add_b_eq);

    [x,fval,exitflag]=linprog(f,S_ineq,b_ineq,S_eq,b_eq,[],[],[],options);

    if abs(x-x_copy)<prec
        stop1=1;
    end
end

Nucleolus=x(1:end-2);
```

```
function Shapley=shapley(S,rm)

n=length(S(1,:));
Shapley=zeros(n,1);
num_players=sum(S')';
weight=factorial(num_players-1).*factorial(n-num_players)./factorial(n);


for i=1:n
    rows=find(S(:,i)==1);
    S2=S(rows,:);
    S2(:,i)=0;

    without_i=zeros(length(S2),1);

    for j=1:length(S2)
        [~,indx]=ismember(S2(j,:),S,'rows');
        if indx==0
             without_i(j)=0;
        else
            without_i(j)=rm(indx);
        end
    end

    Shapley(i)=weight(rows)'*(rm(rows)-without_i);
end
```

```
function Lorenz=lorenz(S,rm);

n=length(S(1,:));

H=diag(ones(1,n));
f=[];
A=S(1:end-1,:);
b=rm(1:end-1);
Aeq=S(end,:);
beq=rm(end);

options=optimset('display','off','Algorithm','active-set');
Lorenz = quadprog(H,f,A,b,Aeq,beq,[],[],[],options);
```

```
function [AS_VAR,AS_ES]=as(S,ret,InvAmount, alpha);

n=length(S(1,:));
profit=ret*InvAmount';

indexVaR=find(profit==quantile(profit,alpha));

if length (indexVaR)==1
    AS_VAR=-ret(indexVaR,:)'.*InvAmount';

    indexES=find(profit<=quantile(profit,alpha));
    AS_ES=-mean(ret(indexES,:))'.*InvAmount';
else
    AS_VAR=zeros(n,1);
    AS_ES=zeros(n,1);
end
```

Code used for the simulation in chapter 6

```
%Random correlation matrix
tic
players=[3:7];
alpha=[0.01 0.05 0.09 0.13];
degr_fr=[3 5 7 9];

nsim=10000;
nsim2=10000;

%Core existence:
CE_ES=zeros(nsim2,length(alpha),length(players),length(degr_fr));
CE_VAR=zeros(nsim2,length(alpha),length(players),length(degr_fr));

%Core compatibility:
CC_Activity_ES=zeros(nsim2,length(alpha),length(players),length(degr_fr));
CC_Incremental_ES=zeros(nsim2,length(alpha),length(players),length(degr_fr));
CC_Cost_gap_ES=zeros(nsim2,length(alpha),length(players),length(degr_fr));
CC_Beta_ES=zeros(nsim2,length(alpha),length(players),length(degr_fr));
CC_Shapley_ES=zeros(nsim2,length(alpha),length(players),length(degr_fr));

CC_Activity_VAR=zeros(nsim2,length(alpha),length(players),length(degr_fr));
CC_Incremental_VAR=zeros(nsim2,length(alpha),length(players),length(degr_fr));
CC_Cost_gap_VAR=zeros(nsim2,length(alpha),length(players),length(degr_fr));
CC_Beta_VAR=zeros(nsim2,length(alpha),length(players),length(degr_fr));
CC_Shapley_VAR=zeros(nsim2,length(alpha),length(players),length(degr_fr));

for v=1:length(degr_fr)
    'Degrees of freedom'
    degr_fr(v)
    toc
    for p=1:length(players)
        n=players(p)
        toc

        for sim_i=1:nsim2;
            Sigma=zeros(n,n);
            for i=1: n
                for j=1: i
                    Sigma(i,j)=random('unif',-1,1,1,1);
                    if abs(Sigma(i,j))==1
                        Sigma(i,j)=random('unif',-1,1,1,1);
                    end
                end
            end

            Sigma=Sigma*Sigma';

            Sigma2=zeros(n,n);

            for i=1: n
```

```
    for j=1:i
        Sigma2(i,j)=Sigma(i,j)/(sqrt(Sigma(i,i))*sqrt(Sigma(j,j)));
        Sigma2(j,i)=Sigma2(i,j);
    end
end

Sigma2;

%Random Standard Deviation
SD=random('unif',0.005,0.8,1,n);

%Random weights
 W=rand(n,1);
 W=(W/sum(W))';

C=chol(Sigma2);

%Z=randn(n,nsim);
Z=random('t',degr_fr(v),n,nsim);
X=C'*Z;
X=X';
%corr(X)-Sigma2;

ret=zeros(nsim,n);
for i=1:nsim
    ret(i,:)=X(i,:)./sqrt(degr_fr(v)/(degr_fr(v)-2)).*SD;
end


S=coalitions(n);
VAR=zeros(2^n-1,1);
ES=zeros(2^n-1,1);

for i=1:2^n-1
    S1=S(i,:);
    retS=ret*(S1.*W)';

    for j=1:length(alpha);
        VAR(i,j)=-quantile(retS,alpha(j));
        ES(i,j)=-mean(retS(retS<=quantile(retS,alpha(j))));
    end
end

for j=1:length(alpha)

    CE_ES(sim_i,j,p,v)=core_existence(S,ES(:,j));
    CE_VAR(sim_i,j,p,v)=core_existence(S,VAR(:,j));


    Activity_ES=activity(S,ES(:,j));
    Incremental_ES=incremental(S,ES(:,j));
    Cost_gap_ES=cost_gap(S,ES(:,j));
```

```
                Beta_ES=beta(ret,W,ES(:,j));
                Shapley_ES=shapley(S,ES(:,j));


CC_Activity_ES(sim_i,j,p,v)=core_compatibility(S,ES(:,j),Activity_ES,0);
CC_Incremental_ES(sim_i,j,p,v)=core_compatibility(S,ES(:,j),Incremental_ES,0);
CC_Cost_gap_ES(sim_i,j,p,v)=core_compatibility(S,ES(:,j),Cost_gap_ES,0);
CC_Beta_ES(sim_i,j,p,v)=core_compatibility(S,ES(:,j),Beta_ES,0);
CC_Shapley_ES(sim_i,j,p,v)=core_compatibility(S,ES(:,j),Shapley_ES,0);


                Activity_VAR=activity(S,VAR(:,j));
                Incremental_VAR=incremental(S,VAR(:,j));
                Cost_gap_VAR=cost_gap(S,VAR(:,j));
                Beta_VAR=beta(ret,W,VAR(:,j));
                Shapley_VAR=shapley(S,VAR(:,j));


CC_Activity_VAR(sim_i,j,p,v)=core_compatibility(S,VAR(:,j),Activity_VAR,0);
CC_Incremental_VAR(sim_i,j,p,v)=core_compatibility(S,VAR(:,j),Incremental_VAR,0);
CC_Cost_gap_VAR(sim_i,j,p,v)=core_compatibility(S,VAR(:,j),Cost_gap_VAR,0);
CC_Beta_VAR(sim_i,j,p,v)=core_compatibility(S,VAR(:,j),Beta_VAR,0);
CC_Shapley_VAR(sim_i,j,p,v)=core_compatibility(S,VAR(:,j),Shapley_VAR,0);
            end
        end
    end
end

toc
```

Code used for concavity checking (the table on page 22)

```
result=zeros(14,1);

for n=2:15
    S=coalitions(n);
    SUM=0;
    for i=1:length(S)
       rk=S(i,:);
       rk2=find(rk==1);
       S2=S(:,rk2);

       if size(S2,2)==1
           SUM=SUM+count(S2==0);
       else
           SUM=SUM+count(sum(S2')==0);
       End

    end
    SUM=SUM/2;
    result(n-1,1)=SUM;
end

result
```