Module 6

This is a single, concatenated file, suitable for printing or saving as a PDF for offline viewing. Please note that some animations or images may not work.

Module 6 Study Guide and Deliverables

Topics:	**Lecture 11:** Risk Analytics in Practice **Lecture 12:** Course Summary & Lessons Learned
Readings:	**Case:** Expect the Unexpected: Risk Measurement and Management in Commercial Real Estate Craig Furfine Lectures 11 and 12 online content
Discussions:	**Module 6 Discussion**
Assignments:	None this week
Assessments:	None this week

Lecture 11

Learning Objectives

After you complete this lecture, you will be familiar with the following:

- Inventory control problems in a general context
- Portfolio optimization: risk versus return
- Choosing among projects with different revenue potentials and probability of success

An Inventory Control Example

In the previous lectures (in particular Lectures 5, 6, and 7), we have worked with the newsboy problem. The newsboy problem is a single-period decision problem in which the decision maker should decide how much to order in a given time period. It is one of the simplest inventory decision problems. In this lecture, we will look at a more general inventory problem where the goal is to find the optimal reorder point and order quantity and to determine the impact of these on the service level and the average inventory carried in the system.

Why do we care so much about the management of inventories? According to The Wall Street Journal, U.S. businesses recently had a total inventory worth about $885 billion dollars.1 As a lot of money is tied up in inventories, businesses want to manage their inventories as effectively as possible. Keeping a high level of inventory in stock adds cost to a business and the business runs the risk of not selling those items which may go obsolete in the near future. On the other hand, by keeping a low level of inventory, a business runs the risk of not being able to satisfy its demand, resulting in low service levels. Therefore, the goal of each business is to find the right level of inventory to maintain in order to minimize these risks.

In this section of the lecture, we will work with the following inventory management problem:

> Laura Tanner is the owner of Millennium Computer Corporation (MCC), a retail computer store in Austin, Texas. Competition in retail computer sales is fierce — in terms of both price and service. Laura is concerned about the number of stockouts occurring on a popular type of computer monitor. Stockouts are very costly to the business because when customers cannot buy this item at MCC, they simply buy it from a competing store, and MCC loses the sale. Laura measures the effects of stockouts on her business in terms of service level, or the percentage of total demand that can be satisfied from inventory.
>
> Laura has been following the policy of ordering 50 monitors whenever her daily ending inventory position (defined as ending inventory on hand plus outstanding orders) falls below her reorder point of 28 units. Laura places the order at the beginning of the next day. Orders are delivered at the beginning of the day, and therefore, can be used to satisfy demand on that day. For example, if the ending inventory position on day 2 is less than 28, Laura places the order at the beginning of day 3. If the actual time between order and delivery, or lead time, turns out to be four days, then the order arrives at the start of day 7. The current level of on-hand inventory is 50 units, and no orders are pending.

MCC sells an average of six monitors per day. However, the actual number sold on any given day can vary. By reviewing her sales records for the past several months, Laura determined that the actual daily demand for this monitor is a random variable that can be described by the following probability distribution:

Units Demanded	0	1	2	3	4	5	6	7	8	9	10
Probability	0.01	0.02	0.04	0.06	0.09	0.14	0.18	0.22	0.16	0.06	0.02

The manufacturer of this computer monitor is located in California. Although it takes an average of four days for MCC to receive an order from this company, Laura has determined that the lead time of a shipment of monitors is also a random variable that can be described by the following probability distribution:

Lead Time (days)	3	4	5
Probability	0.2	06	0.2

One way to guard against stockouts and improve the service level is to increase the reorder point for the item so that more inventory is on hand to meet the demand occurring during the lead time. However, holding costs are associated with keeping more inventory on hand. Laura wants to evaluate her current ordering policy for this item and determine if it might be possible to improve the service level without increasing the average amount of inventory on hand.

Source: Ragsdale, C. T., (2015). *Spreadsheet modeling and decision analysis: A practical introduction to business analytics* (7th edition). Cengage Learning, Chapter 12: Introduction to Simulation using Analytic Solver Platform, pp. 651–653.

Building a R Model of the Inventory Control Problem

Note that we are looking at the 30 days inventory activity in this model. The R Code to build the model is shown below. The sim_inv function simulates the inventory based on the assumptions we used.

```
days <- c(1:30)
b_inventory <- 50
b_inventory[2:length(days)+5] <- 0
demand <- 0
demand_sim <-
function(){sample(x=c(0:10),size=1,prob=c(0.01,0.0
2,0.04,   0.06,     0.09,     0.14,     0.18,
     0.22,     0.16,     0.06,     0.02))}
lead_time_sim <-
function(){sample(x=c(3,4,5),size=1,prob=c(0.2,0.6
,0.2))}
orders_recieved <- rep(0,length(days)+5)
rop <- 28 #Reorder Point
moq <- 50 # Order Quantity
orders <- rep(0,length(days)+5)
sim_inv <- function(){
stock_out <- 0
days <- c(1:30)
b_inventory <- 50
b_inventory[2:length(days)+5] <- 0
  demand <- 0
unfulfiled_demand <- 0

for (i in 2:length(days)){
rdemand <- demand_sim()
rld_time <- lead_time_sim()
demand[i] <- rdemand
b_inventory[i] <- b_inventory[i-1] +
orders_recieved[i] - min(rdemand,(b_inventory[i-1]
+ orders_recieved[i]))

if(b_inventory[i-
1]+sum(orders_recieved[i:length(days)]) < rop) {
   a. orders_recieved[i+rld_time] <- moq +
      orders_recieved[i+rld_time] #In case orders
      are already arriving on that day
   b. orders[i] <-1
     }
if(rdemand > b_inventory[i-1]+orders_recieved[i]){
   a. stock_out <- stock_out+1
   b. unfulfiled_demand <-
      unfulfiled_demand+rdemand-(b_inventory[i-
      1]+orders_recieved[i])
     }

   }
service_level <- 1-
 (unfulfiled_demand/sum(demand[1:length(days)]))
avg_inv <- mean(b_inventory)
return(c(service_level,avg_inv))
}
```

Simulating the Model

Now, let us perform 5,000 trials of this model using the ASP. Figure 11.1 shows the frequency distribution of the service level obtained as a result of 5,000 replications and Figure 11.2 shows the frequency distribution of the average inventory level obtained as a result of 5,000 replications. We observe that with a reorder point of 28 and order quantity 50 units, the average service level is about 96% with a minimum of 82% and a maximum of 100%. The average inventory level turns out to be about 22 units with a minimum of 18 units and a maximum of 30 units.

Figure 11.1: Frequency distribution of the service-level

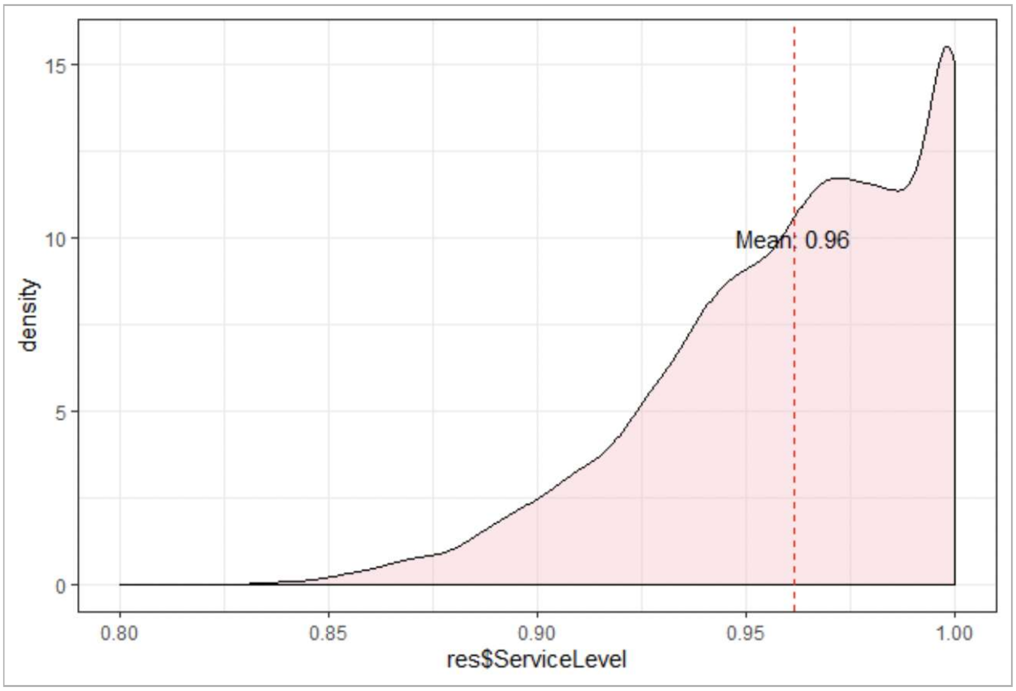
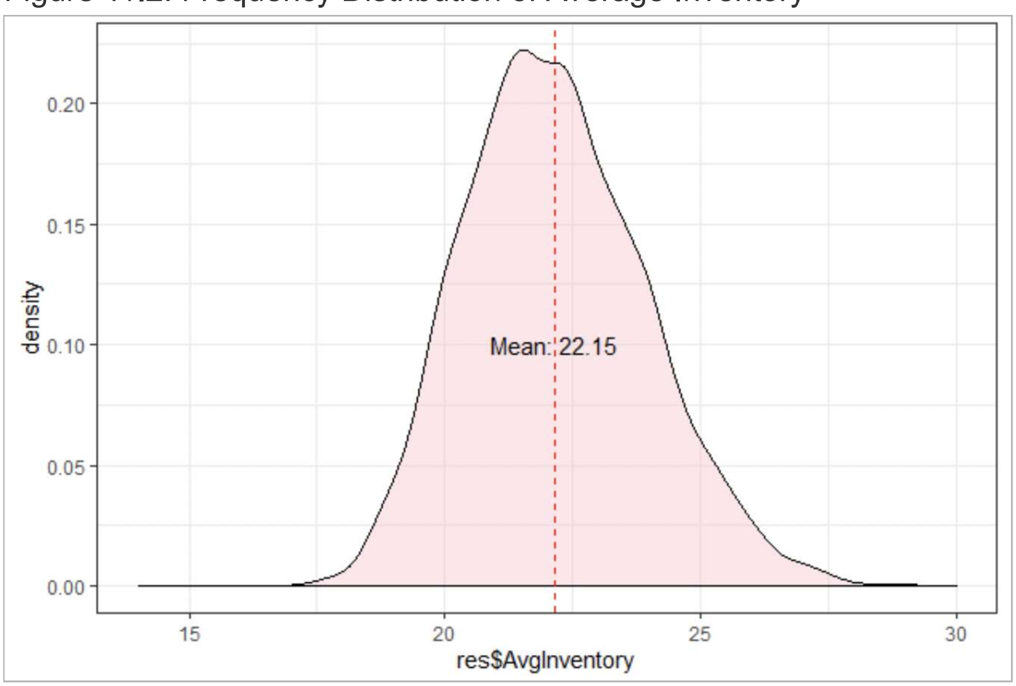


Figure 11.2: Frequency Distribution of Average Inventory



Optimizing the Model

Laura is now interested in finding the reorder point and order quantity that will provide her an average service level of 98% while keeping the average inventory level as low as possible.

Source: Ragsdale, C. T., (2015). *Spreadsheet modeling and decision analysis: A practical introduction to business analytics* (7th edition). Cengage Learning, p. 658.

As you may have noticed, this is a simulation optimization problem. We have already built the simulation model. Now, we will build the optimization model. Before we start working on the optimization model, let us add the following two cells to our spreadsheet:

Let us first define the objective, variables, and the constraints of our model:

Objective: We would like to minimize the average inventory level.

Variables: We would like to find values for the reorder point and the order quantity as a result of solving this optimization problem.

Constraints: The main constraint in this model is the service level constraint; i.e., we would like to keep the service level at least at the level of 98%. In addition, recall that simulation optimization works best if we put bounds on the variables. To this end, let us assume that Laura is considering values between 1 and 70 for both the reorder point and the order quantity. We also need to indicate that these decision variables take only integer values.

The resulting optimization code is as given below. We will explain the intricacies of the code during the class. But please note that the following is a simplified version of the real model. We did not impose integer constraints and we reduced the number of trials to 200 to ensure that the program runs on a reasonably powered computer. In the real world scenario, these constraints play a critical role in achieving causal solutions and hence we will need significant computational resources to achieve a more optimal solution.

```
n_trials <- 200
sim_inv <- function(rop,moq){
  stock_out <- 0
  days <- c(1:30)
  b_inventory <- 50
  b_inventory[2:length(days)+5] <- 0
  demand <- 0
  unfulfiled_demand <- 0
  for (i in 2:length(days)){
    rdemand <- demand_sim()
    rld_time <- lead_time_sim()
    demand[i] <- rdemand
    b_inventory[i] <- b_inventory[i-1] + orders_recieved[i] - min(rdemand,(b_inventory[i-1] + orders_recieved[i]))

    if((b_inventory[i-1]+sum(orders_recieved[i:length(days)]))[1] < rop){
      orders_recieved[i+rld_time] <- moq + orders_recieved[i+rld_time] #In case orders are already arriving on that day
      orders[i] <-1
    }
    if(rdemand > b_inventory[i-1]+orders_recieved[i]){
      stock_out <- stock_out+1
      unfulfiled_demand <- unfulfiled_demand+rdemand-(b_inventory[i-1]+orders_recieved[i])
    }
  }
  service_level <- 1-(unfulfiled_demand/sum(demand[1:length(days)]))
  avg_inv <- mean(b_inventory)
  return(c(service_level,avg_inv))
}
```

We will use the DEOptim function to Optimize our model. Since we are minimizing the average value of the inventory while maintaining 98% service level, let us add the following simulation optimization routine:

```
sim_inv_opt <- function(parms){
rop <- parms[1]
moq <- parms[2]
res <- data.frame(ncol=2,nrow=0)
colnames(res) <- c("ServiceLevel","AvgInventory")
for (i in 1:n_trials) {
res <- rbind(res,sim_inv(rop,moq))
  }
if(mean(res$ServiceLevel)<0.98)
    return (Inf)
return(mean(res$AvgInventory))
}

DEoptim(sim_inv_opt,lower = c(15,3), upper = c(40,200))
```

After optimizing this, we see that our best estimate is having a reorder point at 35 and ordering quantity at about 7. The average inventory is about 11 units This makes sense intuitively by looking at the demand distribution. We often have single digit demands so it makes sense to order less every time but order sooner than the stock reaches 28 (increase the buffer). However, this does not include the logistics and shipping cost attached to it. We can make the problem even more realistic if we added the logistic cost. But let us leave it to a day when you are building your own startup to compete with Amazon.

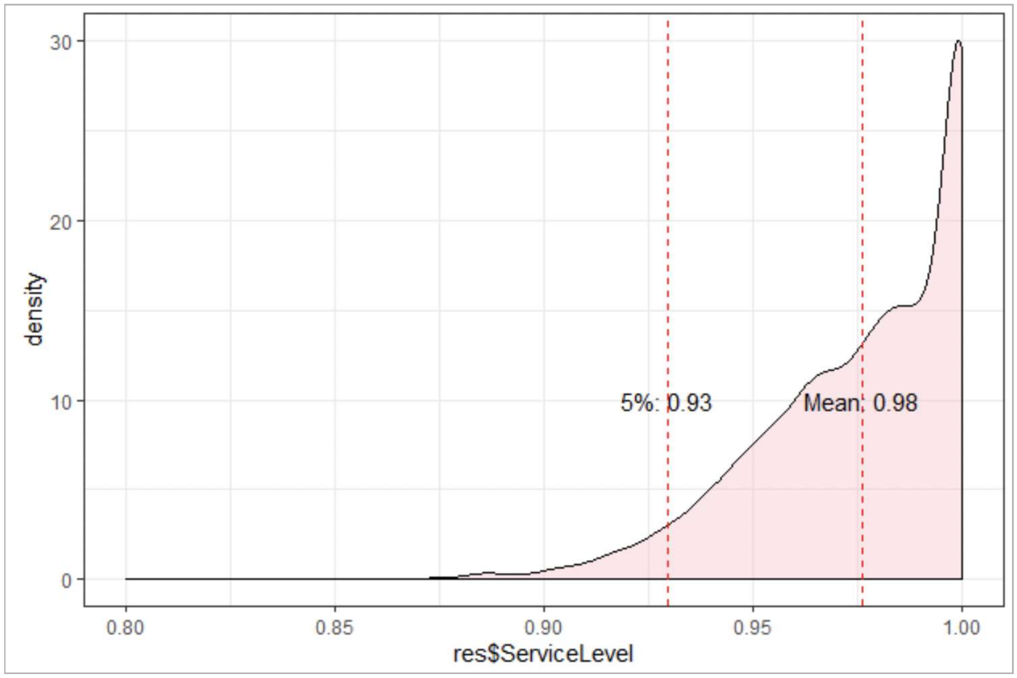
What is important to recognize here is that with the optimal reorder point of 35 and optimal order quantity of 7, MCC is able to keep the service level at 98% and reduce its average inventory level from 22 units to around 11 units, about a 50% reduction.

Other Measures of Risk

This section is inspired from Ragsdale, C. T., (2015). *Spreadsheet modeling and decision analysis: A practical introduction to business analytics* (7th edition). Cengage Learning, pp. 666–667.

We identified an inventory policy that provides a 98% service level on average. What is the downside risk associated with this? To answer this question, let us look at the frequency chart associated with the average service level in the model (Figure 11.3).

Figure 11.3: Frequency chart of the average service level



What we observe in Figure 11.3 is that while the service level of 98% is satisfied on average, there is a 5% chance that the service level will be less than 93%. This means that if Laura uses a reorder point of 36 and an order quantity of 7 then on a given month there is a 5% chance that her actual service level will be less than her desired service level of 93%.

Question: How can we make sure that Laura's goal of achieving 98% service level is guaranteed each month?

Answer: With the help of a quantile (chance) constraint.

With the quantile constraint, we can specify in our model the percentage of times we would like the service level to be below 98%. For instance, let us assume that Laura wants only a 5% chance of observing a service level below 98% in any given month. In other words, 95% of the time, she wants the service level to be at least 98%. Now change the code to add this constraint instead of the average constraint and what do you observe?

A Portfolio Optimization Problem

In this section of the module, we will work with the following portfolio optimization problem:

In recent years, a fundamental shift occurred in power plant asset ownership. Traditionally, a single regulated utility would own a given power plant. Today, more and more power plants are owned by merchant generators that provide power to a competitive wholesale marketplace. This makes it possible for an investor to buy, for example, 10% of 10 different generating assets rather than 100% of a single power plant. As a result, nontraditional power plant owners have emerged in the form of investment groups, private equity funds, and energy hedge funds.

The McDaniel Group is a private investment company in Richmond, Virginia, that currently has a total of $1 billion that it wants to invest in power-generation assets. Five different types of investments are possible: natural gas, oil, coal, nuclear, and wind-powered plants. The following table summarizes the megawatts (MW) of generation capacity that can be purchased per each $1 million investment in the various types of power plants.

Fuel Type	Gas	Coal	Oil	Nuclear	Wind
MWs	2.0	1.2	3.5	1.0	0.5

The return on each type of investment varies randomly and is determined primarily by fluctuations in fuel prices and spot price (or current market value) of electricity. Assume that McDaniel Group analyzed historical data to determine that the return per MW produced by each type of plant can be modeled as normally distributed random variables with the following means and standard deviations.

	Gas	Coal	Oil	Nuclear	Wind
Mean	16%	12%	10%	9%	8%
Std Dev	12%	6%	4%	3%	1%

Additionally, while analyzing the historical data on operating costs, it was observed that many of the returns are correlated. For example, when the returns from plants fueled by natural gas are high (due to low gas prices), returns from plants fueled by coal and oil tend to be low. So there is a negative correlation between the returns from gas plants and the returns from coal and oil plants. The following table summarizes all the pairwise correlations between the returns from different types of power plants.

	Gas	Coal	Oil	Nuclear	Wind
Gas	1	-0.49	-0.31	0.16	0.12
Coal	-0.49	1	-0.41	0.11	0.07
Oil	-0.31	-0.41	1	0.13	0.09

Nuclear	0.16	0.11	0.13	1	0.04
Wind	0.12	0.07	0.09	0.04	1

The McDaniel Group would like to determine the portfolios that determine the maximum expected return at a variety of different risk levels.

Source: Ragsdale, C. T., (2015). *Spreadsheet modeling and decision analysis: A practical introduction to business analytics* (7th edition). Cengage Learning, pp. 673–674.

Building a R Model of the Portfolio Optimization Problem

Portfolio optimization is a very widely used construct across finance, supply chain or project management. In this case, we are minimizing risk while maximizing return. This is often referred to as Mean Variance Optimization. Even though we are doing a simulation optimization, there is a closed form solution once we have identified our random variables to be normally distributed. It is called Quadratic Optimization. We will not go into great detail on that topic, and will leave it to a finance class. However, what we show here will apply to situations where the normality assumption is violated, which is often the case. The data from the problem statement can be summarized in the table below.

PlantType	MWperM	AmountInvested	MWPurchased	MeanReturn	SDReturn
Gas	2.0	200	400	0.16	0.12
Coal	1.2	200	240	0.12	0.06
Oil	3.5	200	700	0.10	0.04
Nuclear	1.0	200	200	0.09	0.03
Wind	0.5	200	100	0.08	0.01

The mean and standard deviation of returns indicate that we are going to use normal distribution to simulate the returns. Also, we have the information about the correlation matrix

V1	V2	V3	V4	V5
1.00	-0.49	-0.31	0.16	0.12
-0.49	1.00	-0.41	0.11	0.07
-0.31	-0.41	1.00	0.13	0.09
0.16	0.11	0.13	1.00	0.04
0.12	0.07	0.09	0.04	1.00

Generating Correlated Random Normal Variables

In the previous modules we generated random variables correlated with each other using NORTARA library in R. But for a normal distribution as we are discussing in this case, we can generate a correlated set using the mvrnorm function part of the MASS package.

For the mvrnorm function to work, we will need the covariance matrix. From statistics we know that covariance and correlation are related to each other as follows:

$$cov(x, y) = \frac{corr(x, y)}{\sigma x * \sigma y}$$

We will accomplish the same using the following code:

```
corr_matrix <- matrix(c(1,-0.49,-0.31,0.16,0.12,-0.49,1,-0.41,0.11,0.07,-0.31, -0.41, 1,0.13,0.09,0.16,0.11,0.13,1,0.04,0.12,0.07,0.09,0.04,1),5,5)
##To use MVNorm we will need Covariance Matrix########
##Cov(x,y) = Corr(x,y)*sigmax*sigmay
cov_mat <- matrix(nrow=5,ncol=5)
for (row in 1: nrow(corr_matrix)){
  for (col in 1: ncol(corr_matrix)){
      cov_mat[row,col] <- corr_matrix[row,col]*model$SDReturn[row]*model$SDReturn[col]
    }
}
mvrnorm(10,model$MeanReturn,cov_mat)
```

What is interesting in a portfolio optimization problem is that we are only interested in the returns of each investment irrespective of how much they are priced at each. Let us use the DEOptim package again for the portfolio optimization problem. Like we discussed before, DEOptim is a very powerful optimizer with a wide variety of applications.

The Optimization Model

The following code is the optimization model of this problem. In particular, we would like to maximize the mean average return with the following constraints:

- The sum of the allocated ratios should add up to 1 (% of weights).
- The amount invested in each plant should be bigger than or equal to 0 (weights cannot be negative).
- The standard deviation of the weighted average return should be less than the allowable standard deviation should be under a threshold, we let the standard deviation take values between 2% and 12%.

```
threshold_sd <- 0.02
n_trials <- 500
port_optim <-function(weights){
  if(sum(weights)==0)
    weights <- weights+1e-4
  parms <- weights/sum(weights)
  pf_return <- weights %*% t(mvrnorm(n_trials,model$MeanReturn,cov_mat))
  pf_sd <- sd(pf_return)

# Risk Constraint
  if(pf_sd > threshold_sd) return (Inf)
  return (-mean(pf_return))
}
optimum <- DEoptim(port_optim,rep(0,5),rep(1,5))
allocation <- optimum$optim$bestmem/sum(optimum$optim$bestmem)
names(allocation) <- model$PlantType
```

Let us start the optimizer with a threshold of 2% and then vary to six different values equally spaced between 2% and 12%; i.e., 2%, 4%, 6%, 8%, 10%, and 12%. Note that we are returning negative of mean because by default the DEOptim package minimizes the objective function. We will minimize negative of mean return, or maximize the mean return.

For a 2% and 12% threshold, we notice that the optimizer found as an allocation in $M as shown in the table. We can notice that as the risk appetite goes higher, the optimizer allocates more to gas. (Why?)

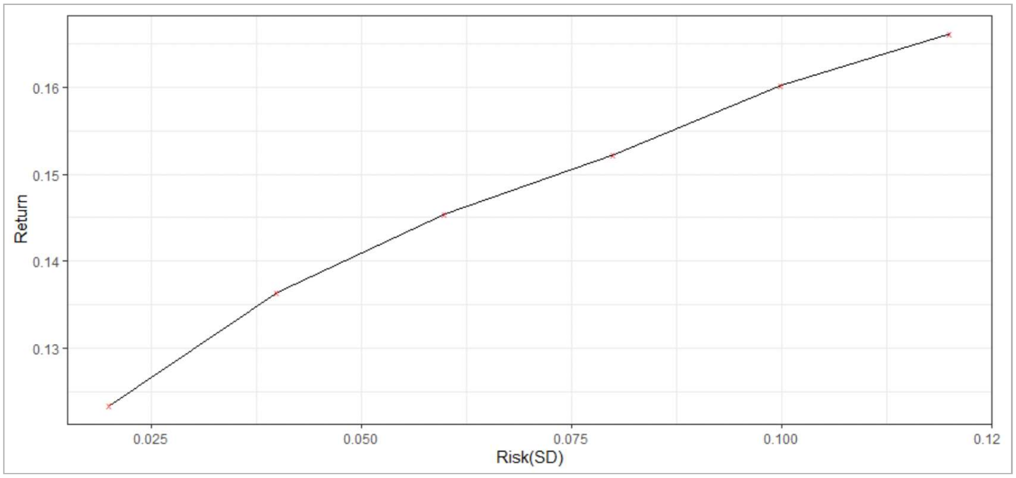
	V1
Gas	238.538596
Coal	404.864873
Oil	352.528083
Nuclear	2.535818
Wind	1.532630

	V1
Gas	975.8636847
Coal	3.4723845
Oil	0.1581358
Nuclear	0.7194602
Wind	19.7863348

Analyzing the Results

In order to better visualize the results, it is important to look at a chart, which summarizes the maximum weighted average return found for each of the six optimizations.

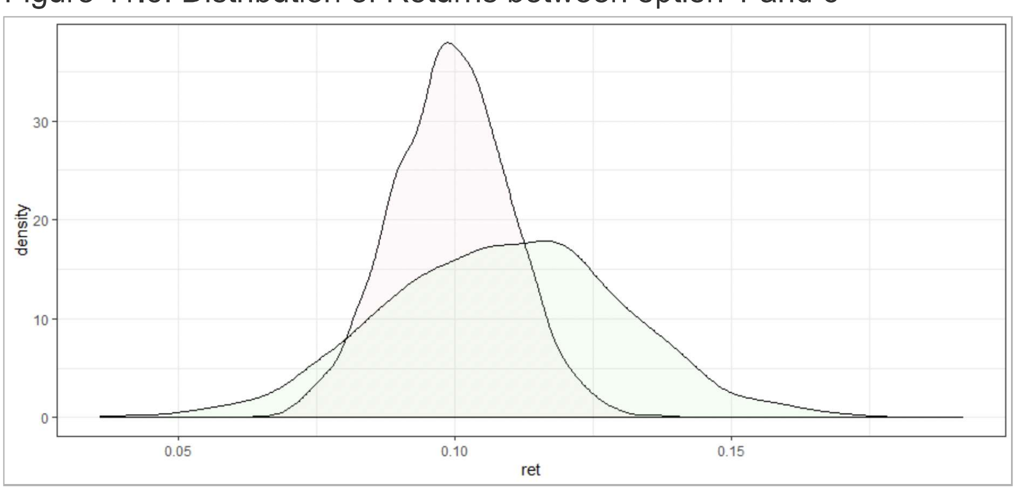
Figure 11.4: Risk Return Tradeoff Curve



So, we conclude that high levels of return are associated with high levels of risk (i.e., standard deviation).

Note that if we invest purely in coal plants, we will have an expected return of 12% with a standard deviation of 6% (as given in the problem description). By making use of correlations we are able to extract returns beyond what an individual element is able to give us. You could also use a similar technique in the stock market, but the challenge is not only coming up with good optimization problem. The underlying assumptions, like how we assumed the returns to be normal, are not always valid. Also, let us look at how uncertain our option 1 and option 6 returns are.

Figure 11.5: Distribution of Returns between option 1 and 6



The pink graph is the distribution of returns for option 1 and green is the distribution of returns for option 6.

Test Yourself Question

Question: Now, what should McDaniel do?

Suggested Answer: It depends on McDaniel's preference for risk. If they prefer lower levels of risk, they should choose the first portfolio, obtained as a result of solving the first optimization problem. If they can accept higher levels of risk, they should choose the sixth portfolio, obtained as a result of solving the sixth optimization problem. Or they may choose a risk level in between the first and the sixth portfolio. The real benefit of this analysis is that it helps McDaniel select a portfolio that provides the maximum return for the desired level of risk or the minimum level of risk for the desired level of return.[1]

Individual Exercise

Please review Section 12.16 of the book Ragsdale, C. T., Spreadsheet Modeling & Decision Analysis: A Practical Introduction to Business Analytics, 7e, Cengage Learning, Chapter 12: Introduction to Simulation Using Analytic Solver Platform, pp. 668-673. This part of the book focuses on a project selection problem, where the decision maker is trying to choose among 8 projects to invest with different initial costs, probability of success and uncertain revenue potential. Please try to solve this problem on your own first. Then, review the analysis presented in the book carefully.

Lecture 11 Footnotes

[1] Ragsdale, C. T., Spreadsheet Modeling & Decision Analysis: A Practical Introduction to Business Analytics, 7e, Cengage Learning, p. 651.

Lecture 11 References

Adapted from Ragsdale, C. T., (2015). *Spreadsheet modeling and decision analysis: A practical introduction to business analytics* (7th edition). Cengage Learning, p. 656.

Adaptd from Ragsdale, C. T., (2015). *Spreadsheet modeling and decision analysis: A practical introduction to business analytics* (7th edition). Cengage Learning, p. 676.

Adapted from Ragsdale, C. T., (2015). *Spreadsheet modeling and decision analysis: A practical introduction to business analytics* (7th edition). Cengage Learning, pp. 678–679.

Ragsdale, C. T., (2015). *Spreadsheet modeling and decision analysis: A practical introduction to business analytics* (7th edition). Cengage Learning

Ragsdale, C. T., (2015). *Spreadsheet modeling and decision analysis: A practical introduction to business analytics* (7th edition). Cengage Learning, p. 654.

Ragsdale, C. T., (2015). *Spreadsheet modeling and decision analysis: A practical introduction to business analytics* (7th edition). Cengage Learning, p. 656.

Ragsdale, C. T., (2015). *Spreadsheet modeling and decision analysis: A practical introduction to business analytics* (7th edition). Cengage Learning, p. 676.

Ragsdale, C. T., (2015). *Spreadsheet modeling and decision analysis: A practical introduction to business analytics* (7th edition). Cengage Learning, p. 678.

Ragsdale, C. T., (2015). *Spreadsheet modeling and decision analysis: A practical introduction to business analytics* (7th edition). Cengage Learning, p. 679.

Lecture 11 Resources

The spreadsheet model for the inventory control problem

The spreadsheet model for the portfolio optimization problem

Boston University Metropolitan College