Module 4

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 4 Study Guide and Deliverables

Topics: **Lecture 7:** Enterprise Decision Making under Uncertainty – I
Lecture 8: Enterprise Decision Making under Uncertainty – II

Readings: Lectures 7 and 8 online content

Discussions: **Module 4 Discussion**

Assignments: **Tutorial:** Optimization libraries in R

Assignment 4: Individual Assignment covering Lecture 7 and Lecture 8.

Assessments: **Quiz 4**

Lecture 7

Learning Objectives

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

- Performing what-if analysis with interactive simulation
- Performing multiple parameterized solutions using R
- The notion of simulation optimization

- Solving optimization problems with uncertainty via simulation optimization

An Airline Revenue Management Model

In this module, we will work with the following airline revenue management model. Our goal is to show the use of multiple parameterized solutions and simulation optimization to solve this problem, but we start with building a simulation model of the problem.

Consider a hypothetical flight from San Francisco to Seattle, where the price for a ticket is $200. The plane can hold no more than 100 passengers. Usually, some passengers who have purchased a ticket are "no-shows". To protect against those no-shows the airline would like to sell more than 100 tickets for each flight. Basically, the airline would like to utilize each seat available as much as possible because after the plane takes off, there is no way to use these seats and it is lost revenue for the airline. On the other hand, if the airline sells more tickets and more passengers than expected show-up, then some passengers will not be able to be seated even though they paid the price of the ticket. To protect the customers, federal regulations require that any ticketed customer who is unable to board the plane due to overbooking is entitled to a compensation of 125% of the ticket value paid by the customer. Any no-show customer is refunded 50% of the ticket value paid by the customer.

The airline would like to know how much revenue it will generate from each flight, less refunds for no-shows and compensation for "bumped" passengers. The airline would further want to know how many tickets to sell to maximize its revenue.

Source: Frontline Solvers Optimization and Simulation User Guide, chapter "Examples: Simulation and Risk Analysis" section "An Airline Revenue Management Model" p. 106.

A simple model of the problem parameters is shown in figure 7.1.

Figure 7.1: Models the Problem

Ticket_Price	penalty	refund	capacity	sales	no_shows
200	250	100	100	To be Simulated	Uncertain

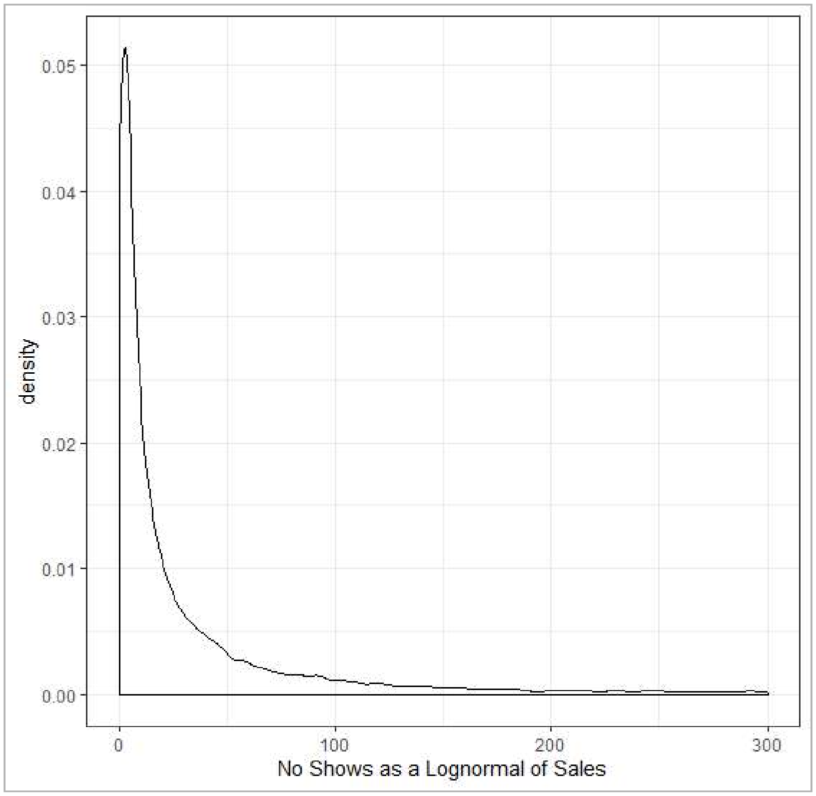
Simple Formulation

The number of no-shows is the uncertain variable in this model. What is important to realize here is that the number of no-shows actually depends on the number of tickets sold. The common practice is to model the number of no-shows with a lognormal distribution. We assume that the number of no-shows is lognormally distributed with a mean of 0.1 × number of tickets sold and standard deviation 0.06 × number of tickets sold. This is to make sure that the number of no-shows is an integer value.

R Tip

To model a Lognormal distribution in R, we use the rlnorm function. However, for a lognormal distribution the parameters are also logarithmic, meaning we have to use log(μ) and log(σ). Also, note that the lognormal distribution has a long tail to the right. So we limit the maximum no shows to the number of tickets sold.

Figure 7.2: Lognormal Modeling of No Shows



- In the base case scenario, let us assume the number of tickets sold is 110. In this scenario we are only interested in the effect of randomness in the no shows on the profitability.
- The number of passengers showing up is basically the number of tickets sold minus the number of no-shows.
- The number of overbooked tickets (bumps) is the maximum of zero and the number of customers showing up minus the flight capacity = AX(0, (passengers – capacity)).
- The total revenue is calculated as follows:

  Total Revenue = Number of tickets sold × Ticket price

  –(Refund to no-shows × Ticket price × Number of no-shows)

  –(Overbooking compensation × Ticket price × Number of overbooked customers)

- We can summarize the simulation just by looking at the mean which would give us the average or Expected scenario.

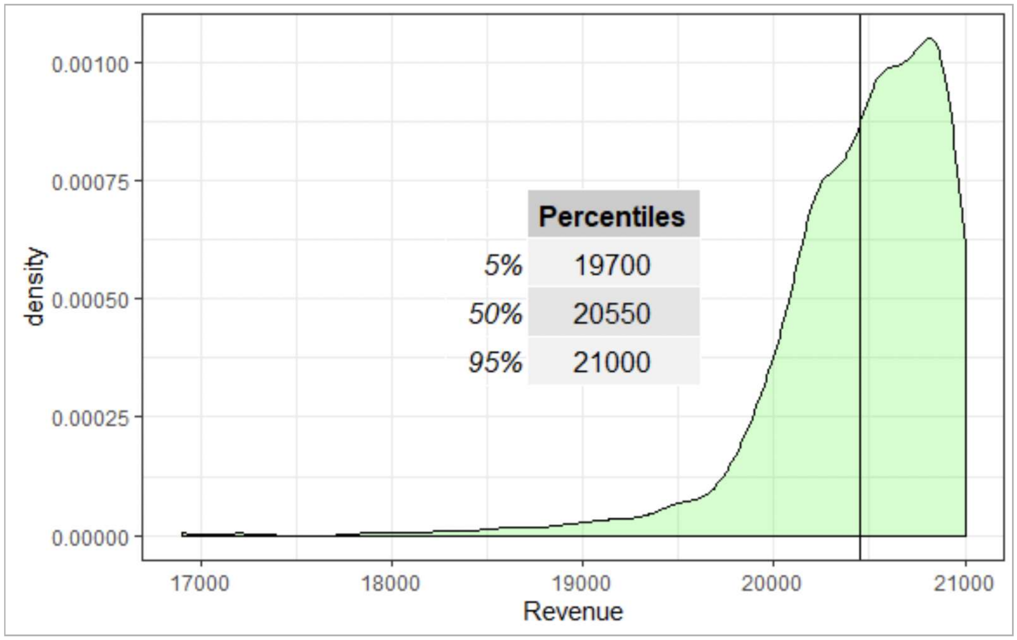
Simulation

To simulate the problem, we will use the following code.

```
no_shows <- sapply(round(rlnorm(n_trials,
log(mean*sales),log(sd*sales))),min,sales)
passengers <- sales-no_shows
bumps <- sapply((passengers -cap), max,0)
revenue <- sales*tkt_price - no_shows*refund -
bumps*penalty
```

Once the simulation is complete, we can plot the density of the simulated revenues. Figure 7.3 shows the density plot. As we can see in the 'Gandalf hat' shaped picture below, the revenue is spread with a wide range between $6,000 to ~$30,000. The wide scenarios are all stemming from one uncertainty in the model: the number of no shows. This is a good example of what risk means to a business, as such a wide range of possibilities locks up significant capital and capacity within the enterprise. If we just looked at the average case scenario, the mean would tell us the revenue on average would be around $19,000. Also, from the percentile table we can see that the median revenue is about $19,800 as well, and at the 95% confidence interval the minimum revenue is $11,000 (5% on the left). It does not tell us how certain we are going to be or how often we are going to be close to the average. But with simulation, we are now in a position to understand the range better.

Figure 7.3: Simulating the airline revenue management model



The above simulation just gave us an understanding of what the range might be. It does not, however, instruct us on what decisions we can make because of it. In the next section, we will do just that. We will extend the process now to understand how many tickets the airline should sell.

We have just answered the question of how much revenue the airline will generate from each flight under the assumptions given in the problem. Now, we can explore the question of how many tickets to sell to maximize the expected net revenue. We will answer this question with the help of the following two methods:

- Interactive simulation & multiple parameterized simulations
- Simulation optimization.

Interactive and Multiple Parameterized Simulation

While this is not the perfect technique, interactive simulation allows us to try out various scenarios to understand how sale of tickets affects profitability. To perform an interactive simulation, we will set up a few scenarios and run. We then randomly change the value of the number of tickets sold, perform a new simulation of 1000 Monte Carlo trials for each new value of the number of tickets sold, and update the expected net revenue accordingly. To make things simpler, let us try sales figures from 80 all the way to 180. And as we perform simulation for each sales figure we will plot the ‘Gandalf hat’ for the entire simulation.

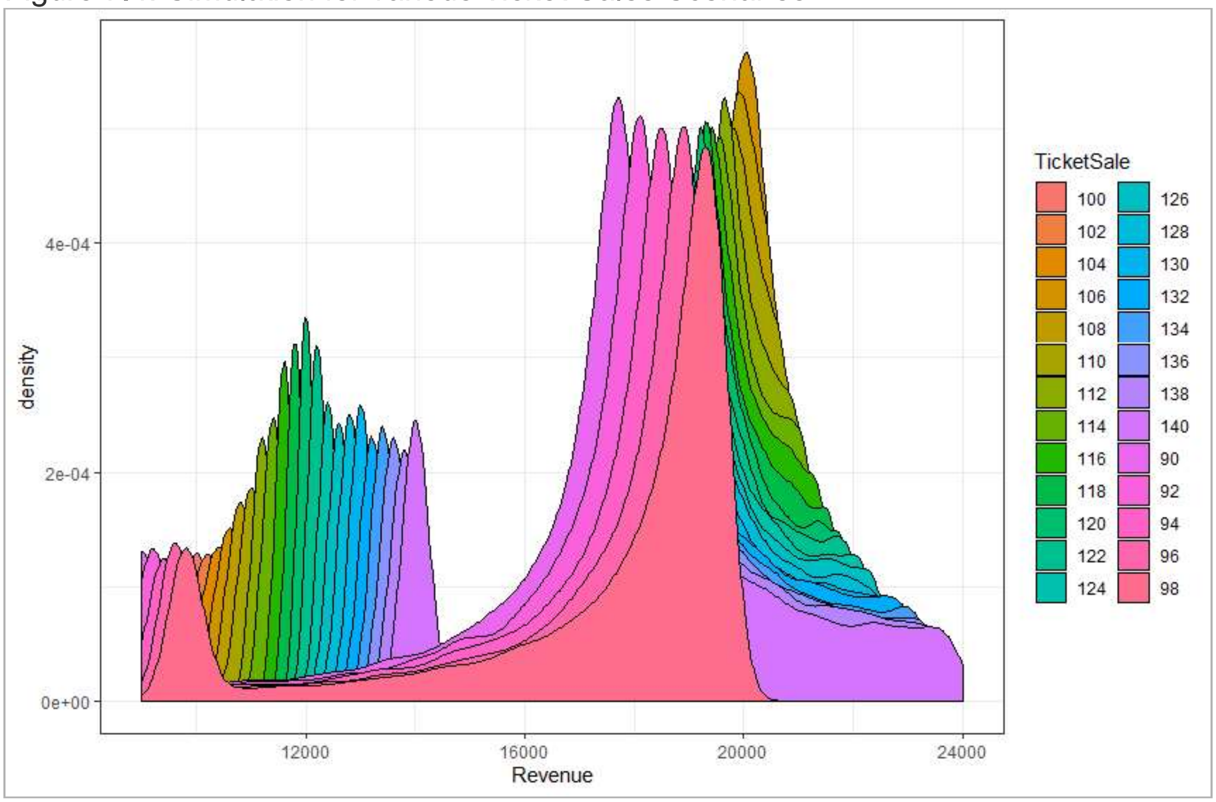
```
sales <- seq (80,180,1)
```

Question

What do you think we can infer from the resulting plot?

Figure 7.4 shows the simulation of the various sales scenarios and if we look at it closely, we notice that better scenarios seem to be closer to selling 110-120 tickets. Purely by eyeballing we also notice there are two peaks for each scenario, as we saw in the earlier picture. The taller peak (Peak1) has a higher probability; the second peak (Peak2) is less likely but still significant. The best scenarios are those in which Peak 1 is as tall as possible, and as far to the right as possible (more revenue), and also in which Peak2 is as far to the right as possible. Statistically speaking we can also talk about average case, i.e., mean of the distribution, but this gives us a better insight of what the downside can be (Peak2).

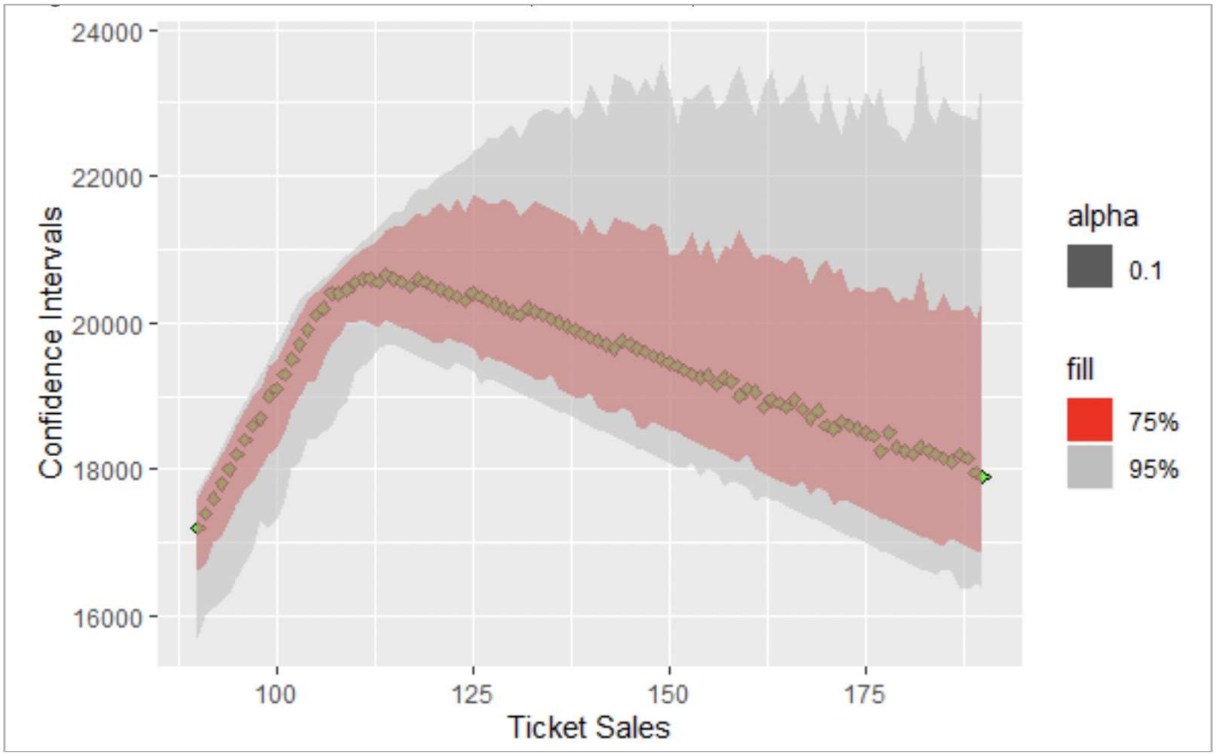
Figure 7.4: Simulation for various Ticket Sales Scenarios



The Trend Chart

It is also useful to look at the confidence chart or the trend chart to see what the 95% and 75% confidence intervals look like across each of the simulation. R has the quantiles command that allows us to calculate the confidence intervals (quantile) as we saw in the previous modules. The 95% confidence interval obviously is a much wider band as can be seen by the gray band. The 75% band, the one in red in the center, is a lot narrower. The line at the center is the median of the simulations for each ticket sales information.

Figure 7.5: Confidence Interval Chart (Trend Chart)



We can also see that the maximum median profit seems to be closer to 106 tickets. Bear in mind that this is a simulation based on our input parameters as well as the penalty/refund structures involved. Also, since this is a probabilistic simulation every one of your answers might be slightly different.

The use of multiple parameterized solutions has allowed us to examine the results of the simulations run with different parameters (i.e., number of tickets sold). From the simulation we can directly answer the question 'How many tickets should the airline sell to maximize expected net revenue?' with a method called simulation optimization. In the rest of this module, we will cover this method.

Note

It is a good idea at this point to review the tutorial on optimization in R. There are several optimization methods that we might use in the following section. Broadly speaking we will first identify if the optimization problem has constraints or not. In this lecture we will mostly cover linear optimization problems, but in finance we will typically talk about quadratic optimization (non-linear) and will need a different set of tools in R to solve those problems. You can think of optimization as a smart algorithm that knows how to search for the optimal value in an efficient manner.

Simulation Optimization

The problem we are trying to solve (i.e., the maximization of the expected revenue) is indeed an optimization problem. But what distinguishes this problem from conventional optimization problems is that it includes uncertainty. Therefore, the name "simulation" is in the title of the approach used to solve this problem.

Airline Revenue Simulation Optimization

We follow the following steps to perform the simulation optimization of the airline revenue management problem. Building a simulation optimization model requires that we build an optimization model and a simulation model. We have already built the simulation model for this problem. Therefore, the following steps will build the optimization model. In Lecture 8, we will build a simulation optimization model from scratch.

'For this exercise we are going to use the optimize function in R. The optimize function takes in arguments the objective function, in our case it is the revenue written as an R function and the search space for the sales of tickets. One approach is to define the **revenue_est** as a function of sales, and returns expected profit. Then we can use the optimize function to optimize the revenue.

```
revenue_est <- function(sales){
  no_shows <- sapply(round(rlnorm(n_trials,
                log(mean*sales),log(sd*sales))),min, sales)
  passengers <- sales-no_shows
  bumps <- sapply((passengers -cap), max,0)

  return (mean(sales*tkt_price - no_shows*refund -
          bumps*penalty))
}
opt <- optimize(revenue_est, interval = c(80,300),maximum =
               TRUE)
```

Question

Do you think this is in line with our uncertainty analysis that we have been doing so far?

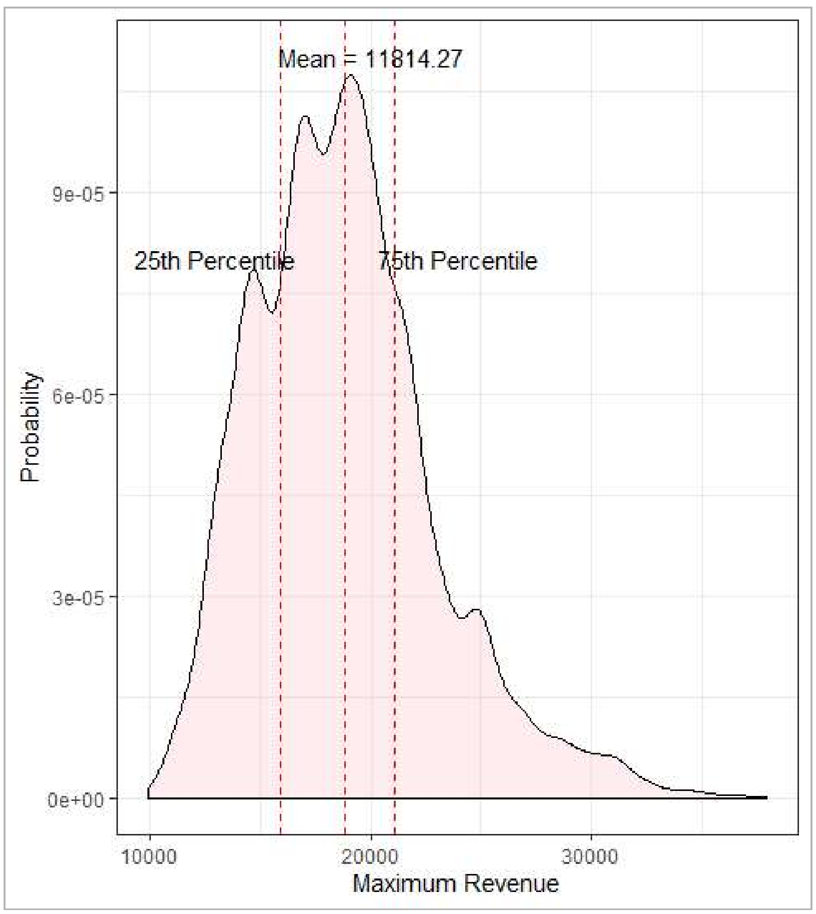
The approach we took above is a common misconception on how to run simulation optimization. This is also called the (f)law of averages. In the above scenario, we are optimizing the revenue over the uncertainty of no shows on the average case. We would, however, like to optimize the revenue for each possible sales scenario and then look at the overall objective.

So we are going to modify our function to get results of optimization for every sales number:

```
for (i in 1:n_trial) {
    rev_fn1 <- function(sales){
      no_shows <- sapply(round(rlnorm(1,
log(mean*sales),log(sd*sales))),min,sales)
      passengers <- sales-no_shows
      bumps <- sapply((passengers -cap), max,0)
      sales*tkt_price - no_shows*refund - bumps*penalty
    }
    opt <- optimize(rev_fn1, maximum = TRUE, interval =
c(80,300))
    opt_max[i] <- opt$maximum
    opt_obj[i] <- opt$objective
}
```

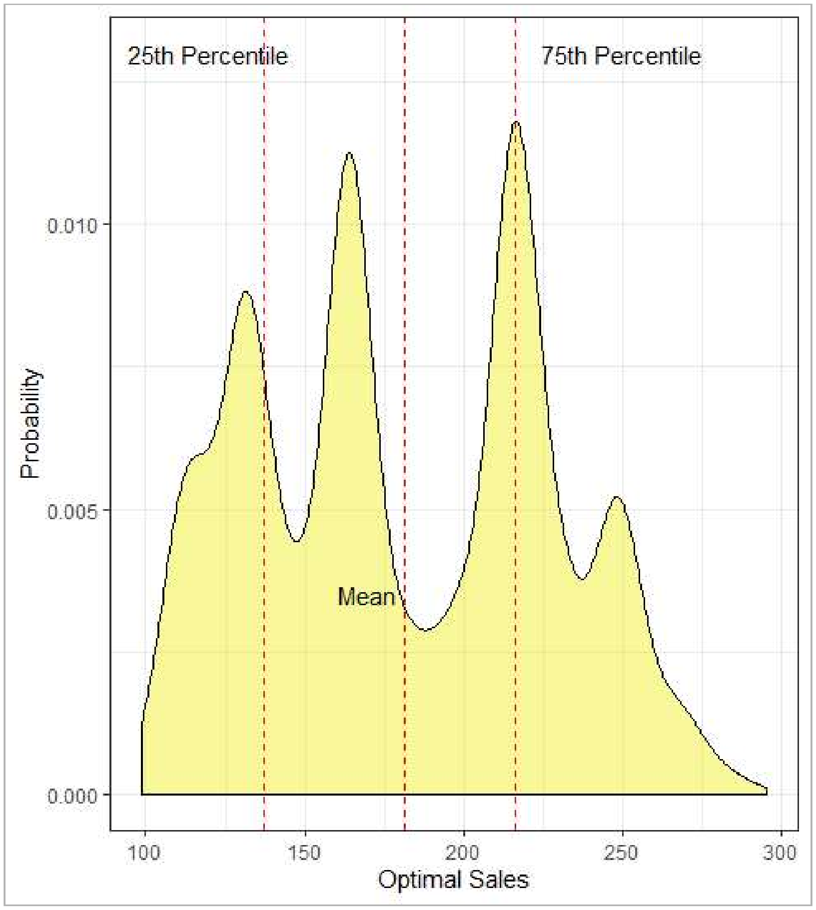
The results are presented in Figure 7.6. The average revenue is around $19,000.

Figure 7.6: Scenarios of optimized Revenues for various uncertainties in No Shows



So with simulation optimization, we did not have to vary any parameter and make subjective judgments. We were able to obtain the optimal solution (the solution that maximizes the expected net revenue in this case) in seconds. If we now look at the range of sales that went into generating the optimized scenarios.

Figure 7.7: Scenarios of Ticket Sales for various uncertainties in No Shows



Looking at Figure 7.7 we notice that there are three solutions that result in maximum revenue, with a sale of 225 being the best-case solution most of the time. Please note that this is a function of the no shows, penalty clauses, and refund policies. This also gives us some insights on why airlines often over-book tickets and by a big margin.

Simulation Optimization: Further Examples

Our goal in this section is to illustrate the use of simulation optimization as a technique to solve optimization problems involving uncertainty that we have covered in our previous modules:

- Newsvendor Problem from Lecture 5.

- Zappos Problem from Lecture 6.

Newsvendor Problem

Suppose that the retail store anticipates at least 40 wool socks will be sold during Valentine's day but the actual amount is uncertain. Each wool sock cost $12 and it will sell for $18. After Valentine's day, the unsold socks will be discounted 50% and are eventually sold.

The question is how many wool socks should the retail store purchase from the wholesaler in the face of uncertain demand?

Figure 7.8: Historical Demand for woolen socks

	V1
1	42
2	45
3	46
4	40
5	43
6	43
7	46
8	42
9	44
10	43
11	47
12	41
13	41
14	45
15	51
16	43
17	45
18	42
19	44
20	48

We will be using the model that uses the method of resampling to represent the demand. It is important to note that we could have used other methods that were covered in Lecture 5 to represent the demand data, however we chose the resampling method for the sake of simplicity. Simulation optimization would work with any other method that is used to represent the demand data.

```
prof_tbl <- c()
supply_tbl <- c()
for (i in 1:n_trials){
  k = 0
  profit_est <- function(supply){
      #We are resampling the demand from the past history
      demand <- sample(sales_tab,size=1)
      sales <- min(demand, supply)
      profit <- sales*price + max((supply -
demand),0)*discount*price - supply*cost
      return(profit)
    }
    temp<- optimize(profit_est,c(30,100),maximum = TRUE)
    prof_tbl[i] <- temp$objective
    supply_tbl[i] <- temp$maximum
  }
```

The code should be straightforward after solving the airline revenue problem. The graphs in Figure 7.9 and 7.10 below show the value of the estimated profit as well as the estimated stock quantity needed to achieve the optimized profit.

Figure 7.9: Profit Simulation for the Newsvendor Problem

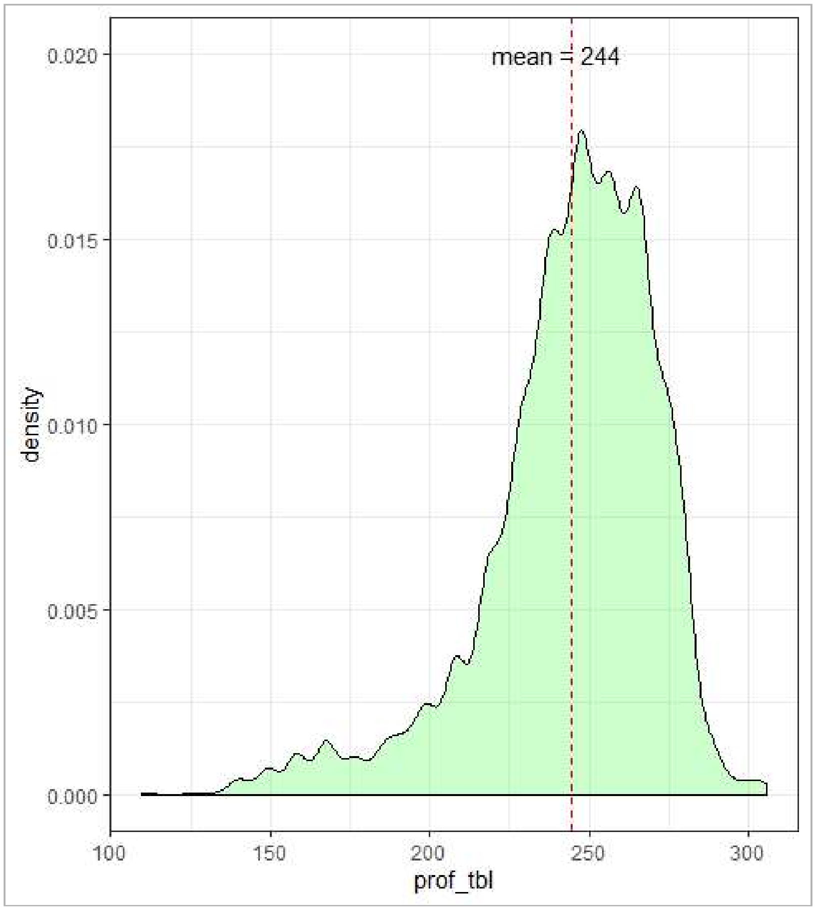
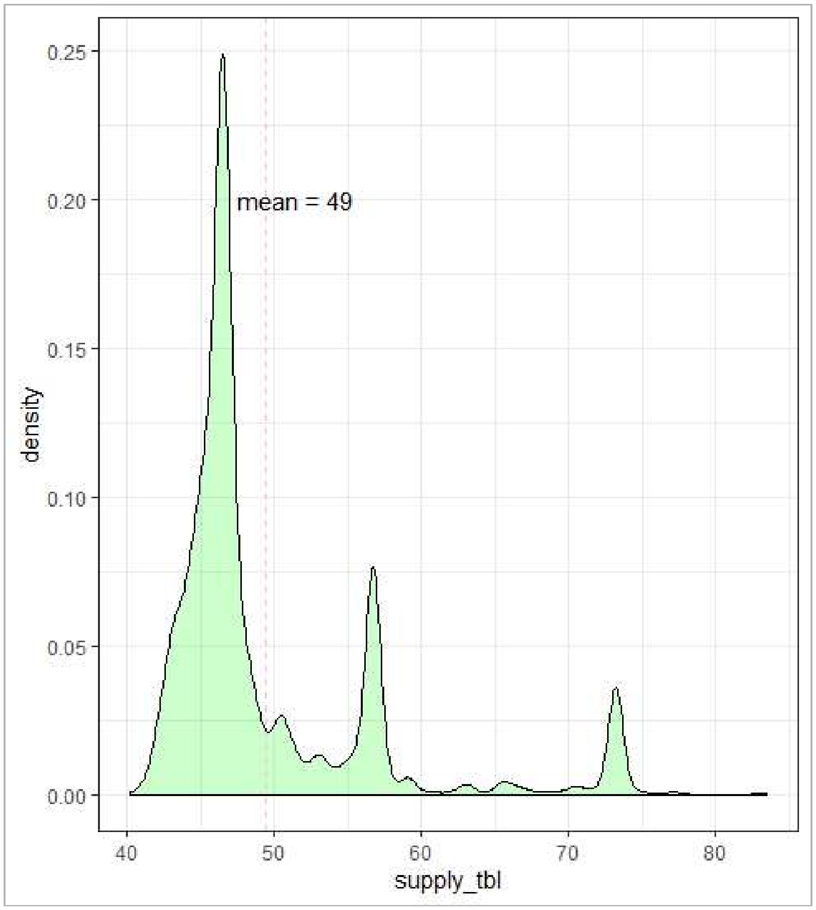


Figure 7.10: Stock Order Quantity for the Newsvendor Problem

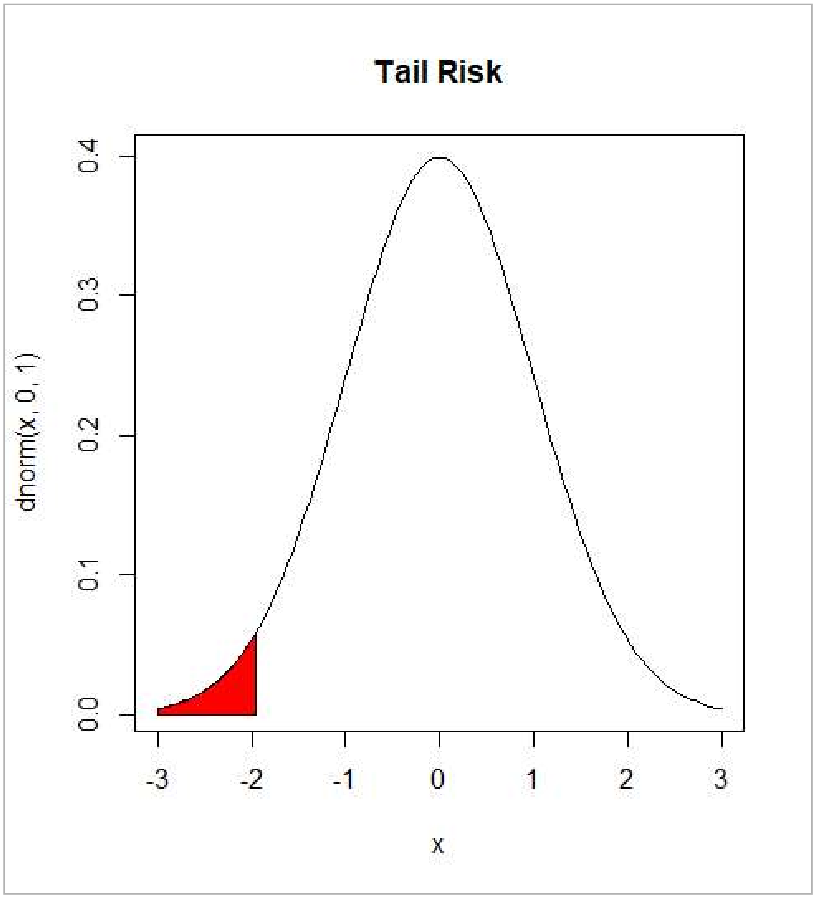


These are problems where we are faced with a single parameter that needs to be optimized based on the inherent uncertainty within the system. Looking at Figure 7.9, we can make intelligent estimates at what is the most likely amount of stock we need to order to arrive at the optimized profit. We can now envisage problems in the real world where usually there are more parameters that determine the outcome of the business. Further, in both the newsvendor and the airline revenue problem we did not impose any additional constraints on the solution. We will now look again at the Zappos problem we discussed earlier, and see if we can find an optimal solution across multiple parameters along with imposing some constraints to achieve them.

Minimizing Tail Risk

The real value in performing stochastic optimization is not just understanding how the objective function can vary according to uncertainty but actually making decisions to minimize or maximize probabilistic outcomes. One common phenomenon that most of you are familiar with called tail risk. In many business scenarios we want to minimize the tail risk if it is a cost function. If it is a profitability curve then we might want to maximize the area in red, that guarantees for instance, if we want to cap our worst-case outcomes.

Figure 7.11: Tail Risk of an uncertain Model



Let us modify our newsvendor model to now calculate tail risk at 5%. Here the tr is a variable where we are calculating the tail risk. We want to put a lower limit on what the tail risk should be, or minimize the red shaded area. So we want to optimize how much we order that minimizes tail risk. However, it is also important to know the trade-off in such decisions. You can always minimize the tail risk by not taking any risk at all, which defeats the purpose of simulation, or for that matter, decision making. Such a trade-off is often referred to as utility function. So if you look at the code, we added a constraint to include only results that fulfil the minimum profitability criteria. We will cover constrained optimization in the next section and explain how to solve multi-parameter problems with constraints.

```
profit_est <- function(supply){
  profit <- c()
  for (i in 1:n_trials){
    #We are resampling the demand from the past history
    demand <- sample(sales_tab,size=1)
    sales <- min(demand, supply)
    profit[i] <- sales*price + max((supply -
demand),0)*discount*price - supply*cost
  }
  if(mean(profit) < 200)
    return (Inf)
  tr <- 0
  for (k in min(profit):quantile(profit,0.05)[[1]]){
    tr <- tr + sum(profit == k)*k
  }
  return (tr)
}
```

Zappos Problem

Our goal is to solve the Zappos problem that was introduced in Lecture 6 using simulation optimization. We will use the task pane to develop the simulation optimization model.

Zappos is a retailer of women's clothing, accessories, and beauty products. Suppose that the regional purchasing manager for Zappos is in the process of determining the order quantities for a new product line of women's sleepwear for the upcoming holiday season. The new product line consists of four types of pajamas: cotton, flannel, silk, and velour. We are given the following data for this problem.

Table 7.1: Available data for Zappos problem

	Cost	Selling Price	Discount Price
Cotton Pajama	$25	$30	$10
Flannel Pajama	$25	$40	$10

Silk Pajama	$35	$60	$30
Velour Pajama	$30	$55	$20

The cotton and the flannel pajamas are similar to other products that Zappos has sold in the past. The marketing department has compiled representative samples of cotton and flannel pajama demand as shown in Table 7.1.

The silk pajama's design is different from all of Zappos' other existing products, therefore the marketing department does not believe that we should use past data to model the demand of silk pajamas. However, the marketing department has conducted extensive surveys to estimate the likely demand scenarios for silk pajamas and obtained the following information:

Table 7.2: Demand scenarios for silk pajama demand

Values	Demand
5,000	0.10
10,000	0.40
15,000	0.20
20,000	0.25
25,000	0.05

In addition, Zappos estimates a maximum silk pajama demand of 30,000 and a minimum possible demand of 0.

The velour pajama is a new product being introduced this season. Therefore, we cannot use past data to model it and the only information we have is the minimum, most likely, and maximum values of velour pajama demand, which are 2500 units, 10,000 units, and 25,000 units, respectively.

An initial order quantity of 10,000 of each type of pajama has been proposed, and we evaluated this decision with a simulation model in Lecture 6. Our goal in this lecture is to answer the question of what order quantity for each pajama type is needed to maximize the profit.

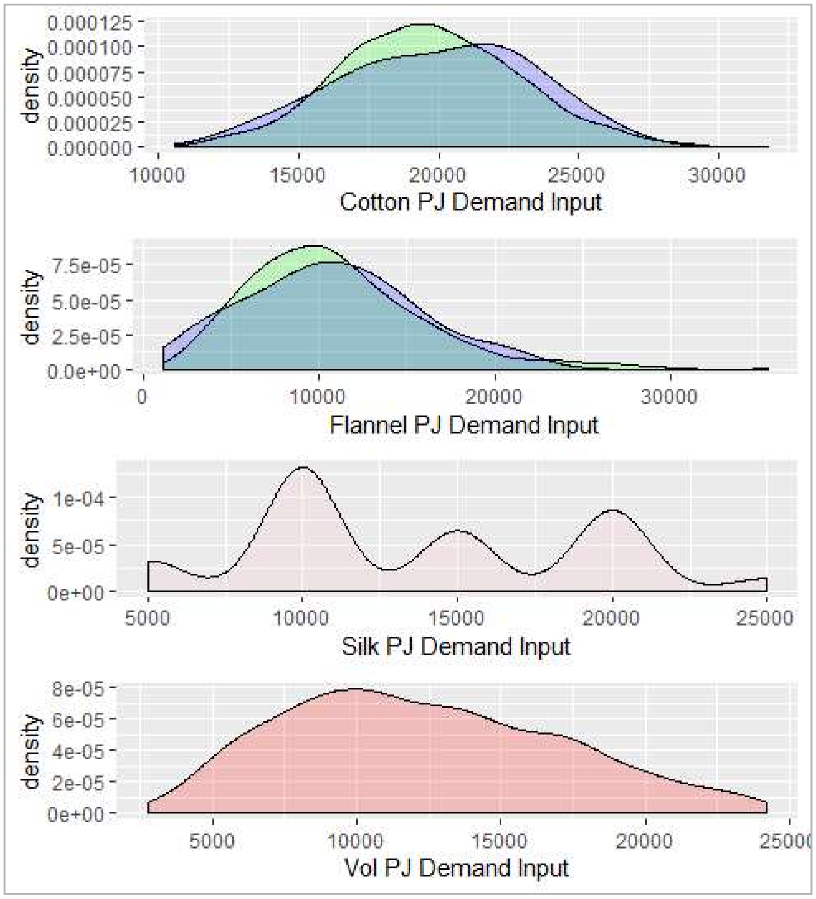
In addition, we are provided with the following information (business constraints):

1. Zappos wants to spend no more than \$1,150,000 on the wholesale procurement of the pajamas.
2. Management wants to order at least 1000 units of each type of pajama and does not want to order more than 20,000 units of any single pajama type.

Source: Adapted from Camm et al., Essentials of Business Analytics, 2015, Cengage Learning, pp. 518-524.

Like before, let us model the inputs of the simulation. For the cotton pajamas, and the flannel pajamas let us continue to fit a beta distribution based on historical data. Figure 7.12 shows the demand distributions for various clothing. The green shade represents the original data, and blue represents the fitted distribution. For silk pajamas, since the demand is unknown, we will use the probability distribution as suggested in the problem and resample according to the distribution. This is done in R using the sample function. Finally, for velour pajamas let us fit a triangular distribution.

Figure 7.12: Input modeling of the Demand for clothing types



For the optimization of this problem we will use a powerful optimization package in R called DEOptim. It is not a linear solver, like the previous problem, but an evolutionary solver. The advantage of this solver is that it can be used for solving non-linear problems as well. The formulation of this problem, however, is slightly different than the usual optimization setup. One of the additional nuances for this problem is that we want to be able to take into account the constraint that the supplier does not want to spend more than $1,150,000 on clothing in total.

```
1  for (k in 1:1000){
2  for (i in 1:nrow(zappos_model)){demands[i] <-
   zappos_model$demand[[i]]()}
3  profit_fn <-function(production){
4  profit <- 0
5  cost <- 0
6  for (i in 1:nrow(zappos_model)){
7  demand <- demands[i]
8  cost <- cost + production[i]*zappos_model$cost[i]
9  profit = profit-zappos_model$cost[i]*production[i] +
   zappos_model$sales_price[i] *
   min(demand,production[i]) +
10 zappos_model$clearance_price[i]*max((production[i]-
   demand),0)
11 }
12 if (cost > 1150000) return (Inf)
13 return (-profit) # We are using the negative of profit
   because the function minimizes the data
14 }
15 temp <- DEoptim(fn =
   profit_fn,lower=c(1000,1000,1000,1000),upper=c(20000,2
   0000,20000,20000))
16 opt_parms[k] <- list(temp$optim$bestmem)
17 opt_res[k] <- temp$optim$bestval
18 }
```

We are going to simulate this over 1000 times, and get the objective function. Please note that this optimization is going to take much longer than the previous scenarios. For DEOptim package, the constraints are defined within the function. For instance, if you look at line 12 of the code above, you will notice that we are returning Inf value when the constraint is not met. This is a way of telling the optimizer to not search for values that fail the constraint. Another point to note is DEOptim minimizes the objective function. To maximize, we then negate the objective function, meaning we include a negative sign in front of the profit.

The results of the simulation are shown in the figures below.

Figure 7.13: Input Estimates for Optimized Profit

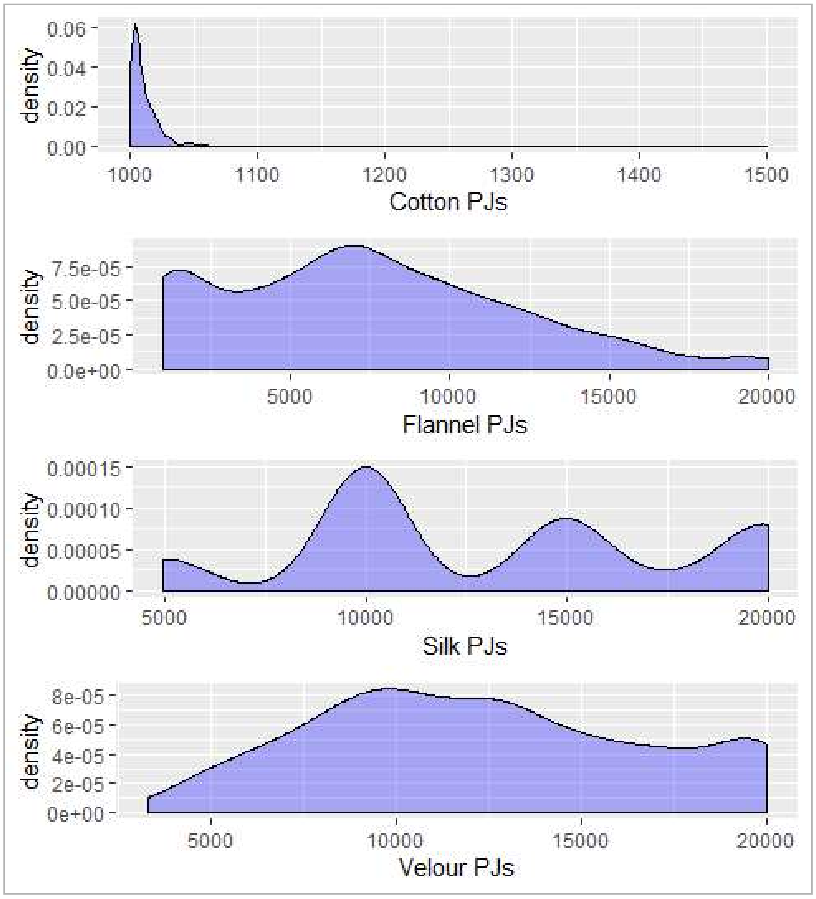
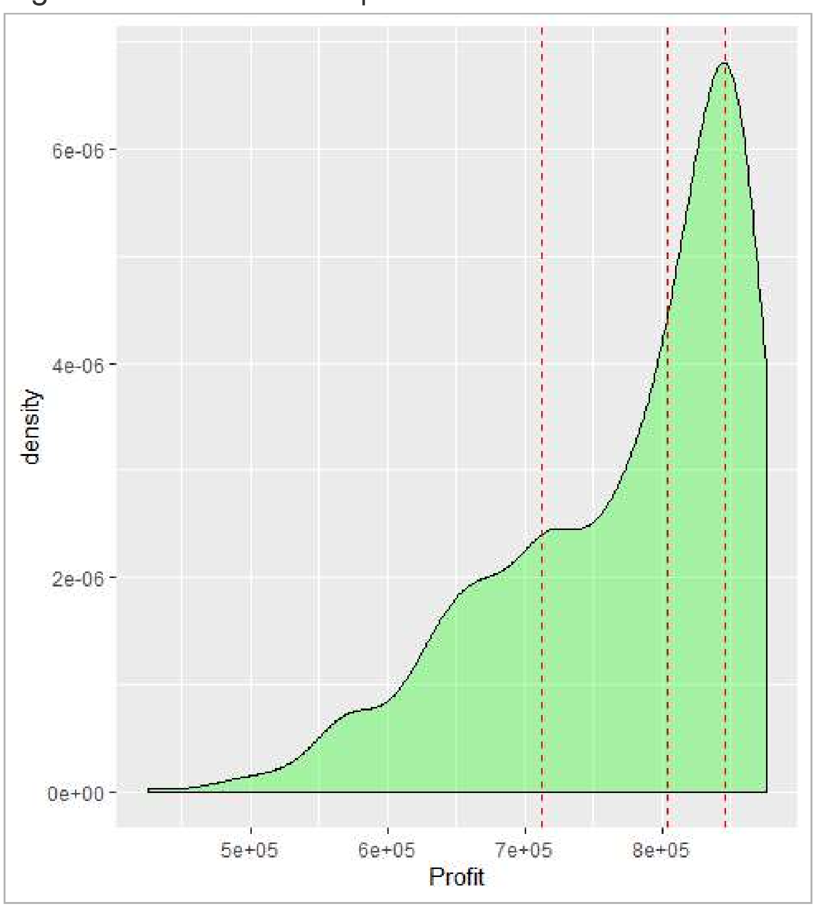


Figure 7.14: Simulated Optimized Profit



We would ideally like to simulate more but given the computational constraints of our laptops we limited the number of solutions and the iterations. After simulation, we learned the average estimate of profit is $772,817 and the median values for production for cotton, flannel, silk and velour pajamas are 1963, 7018, 10,002, and 11,922 respectively. Note that you might get different answers due to inherent randomness but the numbers should still be in that proportion.

Question

Based on this simulation, what is the optimal quantity of each item that needs to be purchased? Hint: Take into account that this is a simulation across four uncertain parameters. How would you modify the code to arrive at the optimal ordering quantities for each of the types? Note: This is a computationally heavy exercise. If your laptop is not able to run this code, we will demonstrate it in the class using a powerful machine.

Lecture 7 References

Camm, J. D., Cochran, J. J., Fry, M. J., Ohlmann, J. W., Anderson, D. R., Sweeney, D. J., & Williams, T. A. (2015). *Essentials of business analytics* (1st edition). Cengage Learning.

Frontline Solvers Optimization and Simulation User Guide, Chapter "Examples: Simulation and Risk Analysis," Section "An Airline Revenue Management Model," p. 107.

Lecture 7 Resources

[Airline revenue management model]

[Multiple parameterized solutions for airline revenue management model]

[Simulation optimization model for the airline revenue management problem]

[Simulation optimization model for the newsboy problem]

[Simulation optimization model for Zappos problem]

Lecture 7 Summary Questions

1. What is the goal of "multiple parameterized simulations"?
2. What is simulation optimization? What type of business problems are best solved with simulation optimization?
3. Please comment on the following statement: *"Simulation optimization works best when we bound decision variable(s)."*
4. Why is it a good idea to solve a simulation optimization problem several times before making a final decision?

Boston University Metropolitan College