

CSCI 164 Final Project - Factory Simulation

Matt Doherty, Nicholas Fong, Julia Marcondes, Christina Matera

Project Description:

A factory holds 7 machines, each of which can be one of two varieties of machine. The first variety is a machine with a high throughput and quick production speed, but also more likely to break down and be out of service. When this first machine type breaks down, it is slow to repair, so it is out of commission for a long time, and the cost of repairs are higher.

The second variety is a machine with a low throughput and slow production speed, but also less likely to break down and be out of service. When this second machine type breaks down, it is quick to repair, so it is only out of commission for a short period of time, with a lower cost of repairs.

There is only one repair crew for each class of machine so only one machine of each class can be repaired at a time.

We will run a simulation to predict the most efficient combination of the first and second varieties in our factory of seven machines. We will run this simulation over a long period of time and will be able to report the frequency of extreme events (ex. three days out of year, all systems down), and also the overall rate of production for each combination of machines.

We need 2 Weibull distributions:

- 1 for how long machine is running
- 1 for how long it took to repair (high alpha, low beta)

State Variables:

We plan to record the information for the machines by creating two classes: one class for each of the two varieties of machines. The 7 machines will be recorded as combinations of instances of these two classes. The following state variable will be added as a member variable to each machine.

- Number of machines that are broken of each type
 - State of each machine (either broken or working)
 - Each machine will have a state variable associated with it, which will always be checked before the machine is able to operate. Whenever the state is recorded as "broken," we will take it out of commission. The repair subroutine will address

any broken machines (one at a time) and the machines will come back into production as soon as the given time of repair has elapsed.

- Repair Queue (one for each type of machine)
 - Seeing as how we can only repair one fast and one slow machine at a time, we will need two repair team queues so we can keep track of how many machines are currently being repaired, or in line to be repaired.

Distributions:

- Rate of production for each machine
 - Both the slow and fast machines will have a given throughput. This rate of production will be associated with each machine based on which variety it is (slow or fast). Each time we simulate a different combination of machine varieties, we will change the combination of machines, but during each individual simulation, this variable will not change. It is a fundamental property of the two different machines.
- Rate of repair for each machine
 - Slow machines and fast machines will have their own distributions to represent the time that each machine spent in the shop. At the end of each simulation, we will collect the average time that each type of machine spent out of the factory and analyze how much of a loss the factory experienced during the machines' absence.
- Repair cost for each machine
 - Each machine will have their own distribution dependent on the severity of their breakdowns. Fast machines will generally have costly repairs and slow machines will have reasonable prices. During each simulation, we will collect the costs for each machine and calculate the best net worth (total profit - total costs of repairs).

Counter Variables:

- Total products made
 - During each simulation, we will keep a tally of the total products made. This will represent the total income that the machines produced. This is a major key in analyzing the most efficient combination of machines.

- Cost of repairs
 - After each repair, we will collect the cost of each machine and sum them all. At the end of each simulation, we will subtract the total cost of repairs from the total income of each simulation to calculate the best profit.
- End of period profit (total products made - cost of repair)
 - The end-of-period profit will be the deciding factor to determine the best combination of fast and slow machines. We are able to calculate this because we will keep track of all products made and the total cost of repairs.
- Total wait time for repairing machines
 - This will help determine which machine is most efficient overall. Although slow machines produce less product, it may be more efficient because the wait time for fast machines may be much larger than the slow. This variable will account for this. While the end of period profit may seem like the universal scale of efficiency, it is worth noting if the machines have a extreme amount of down time that could impact operations. This could result in a situation that might have the best overall end profit, but having these extreme periods of low activity could make the situation suboptimal.
- Time of break down
 - We will keep track of the time of break down of each machine so can return the broken machine to the factory once it has been repaired.
- Time machine returned from repair
 - This will be the time that the machine broke down plus the time it took for its repair. At this time, we will flip the said machine's state from broken to working and return it to the factory.

System Parameters:

- Number of fast machines and slow machines
 - We will simulate each run with every combination of the fast and slow machines. At the end of each simulation, we will average the total products made, how many machines broke and for how long, and total costs and profits.

Event List:

- A machine breaks down:
 - Add machine to the appropriate repair queue

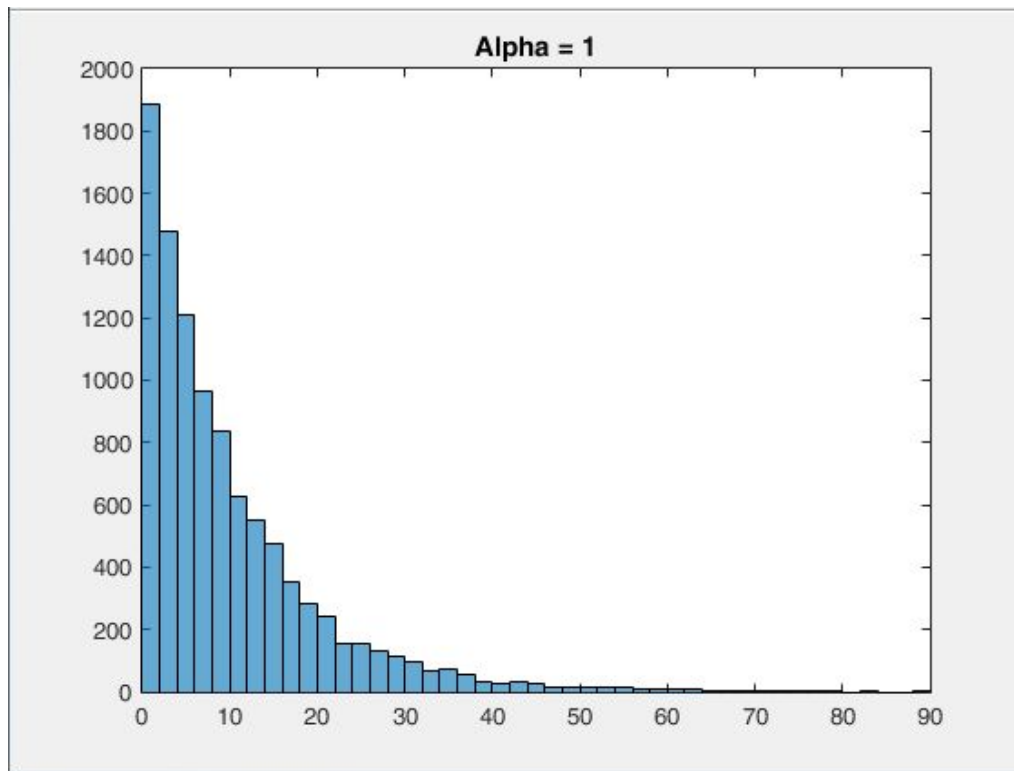
- Increase that queue by 1.
 - If the queue is 1 then generate a repair time, update counter variables
 - If queue is ≥ 2 then simply increase queue number
- Adjust production rate
- A machine is repaired:
 - Adjust production rate
 - Reduce queue by 1.
 - If new queue is >0 then generate a repair time, update counter variables
- A product is made (We might not need this if the rate of products being made is super high since the amount we truncate from just doing rate \times time is negligible)

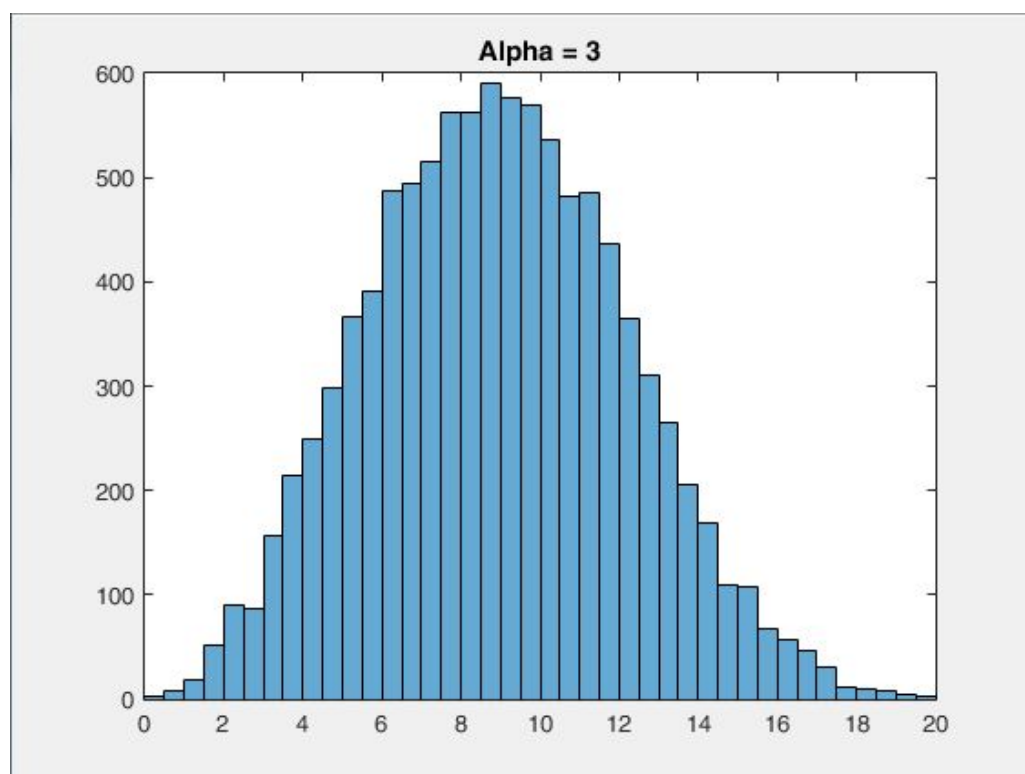
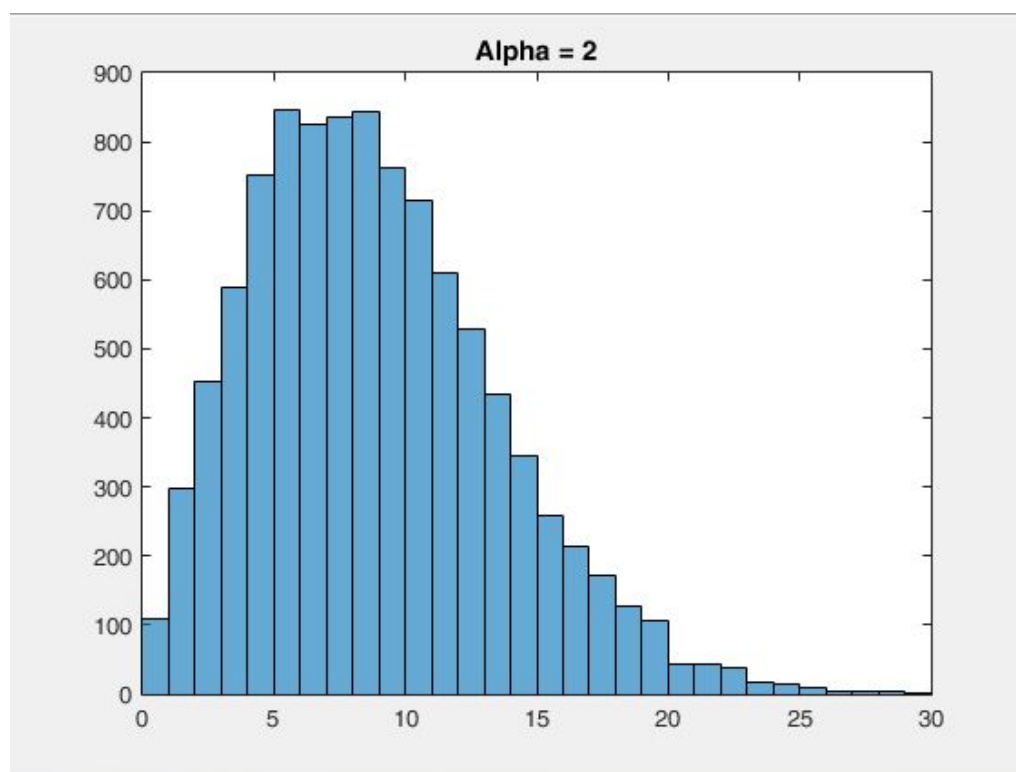
Search Strategy:

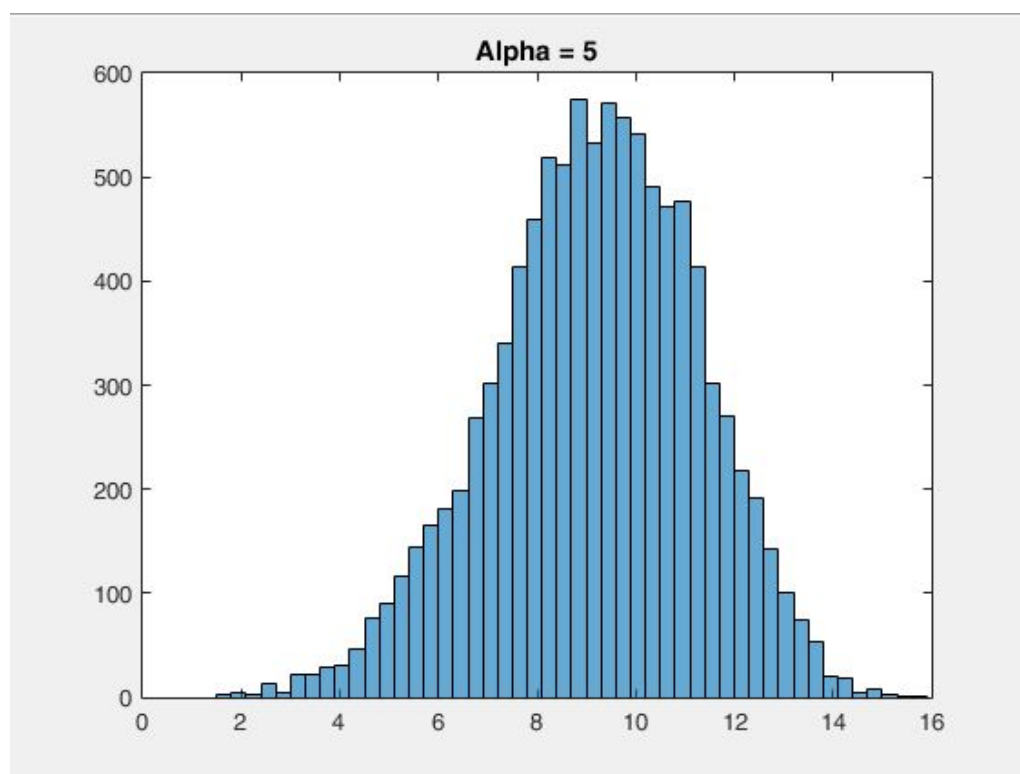
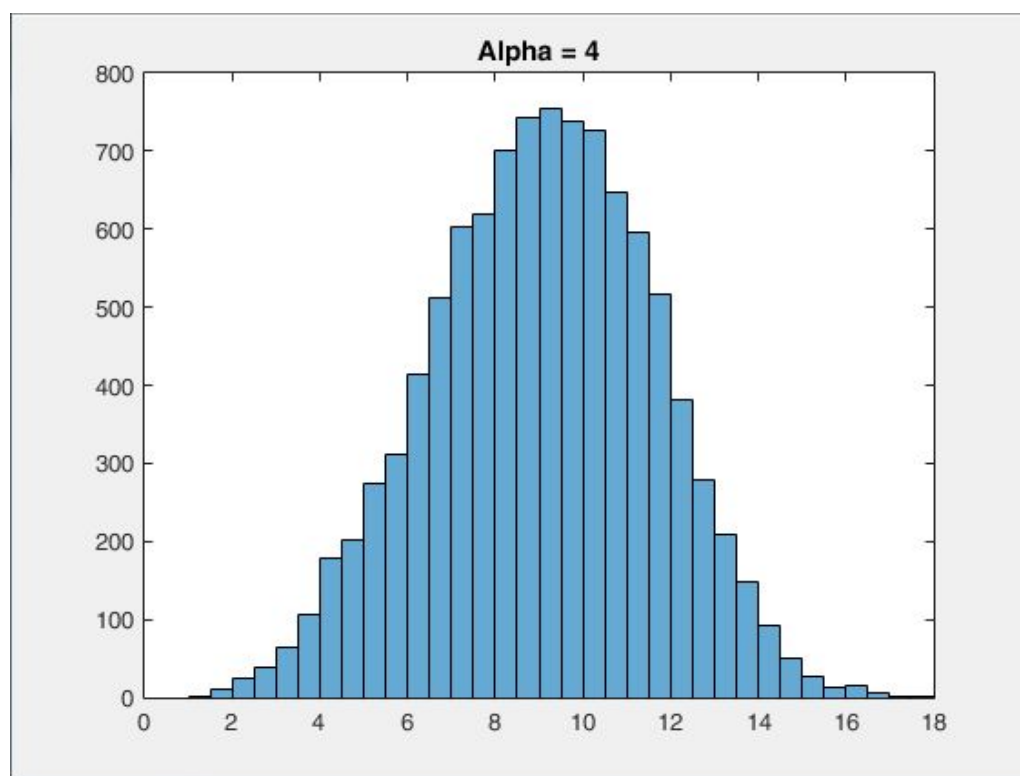
- Perform many trials each combination of n fast machines and $(7-n)$ slow machines with $0 \leq n \leq 7$, averaging the results to find the combination that optimizes profit and efficiency, by taking into account:
 - Cost of each machine
 - Cost of repairs
 - Revenue from products (just a function of how many products are made)
 - Machine down time (Only indirectly affects profits so this might not be needed, but it might be relevant if we want a more consistent factory)

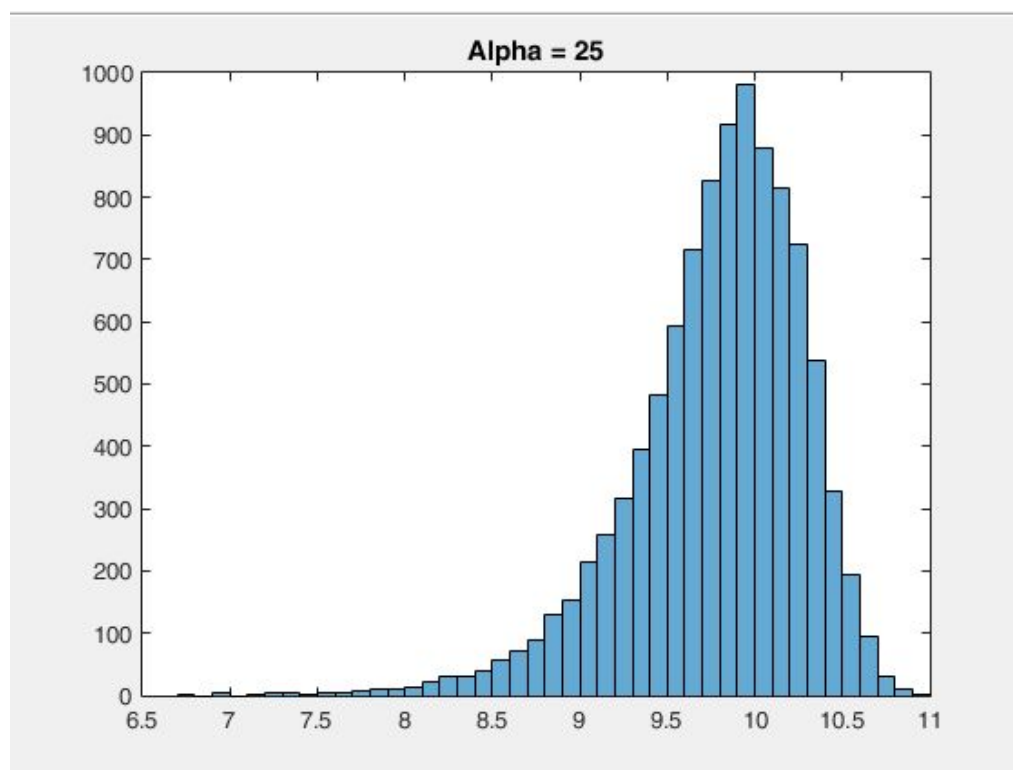
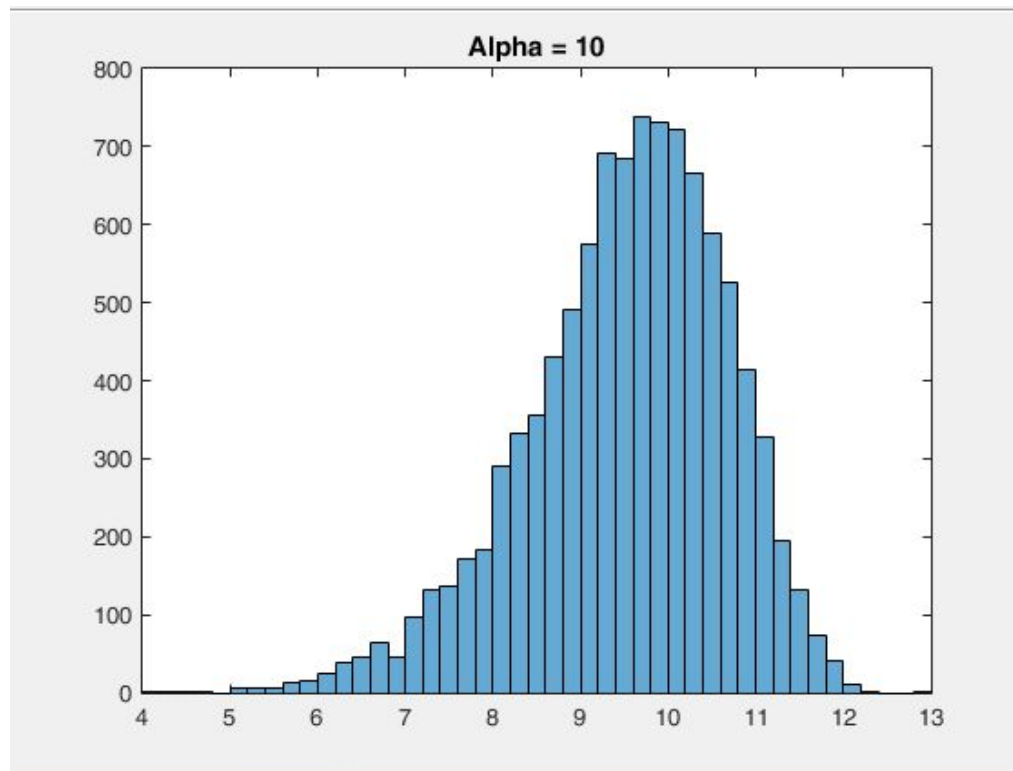
Brief look at Weibull distributions by the numbers:

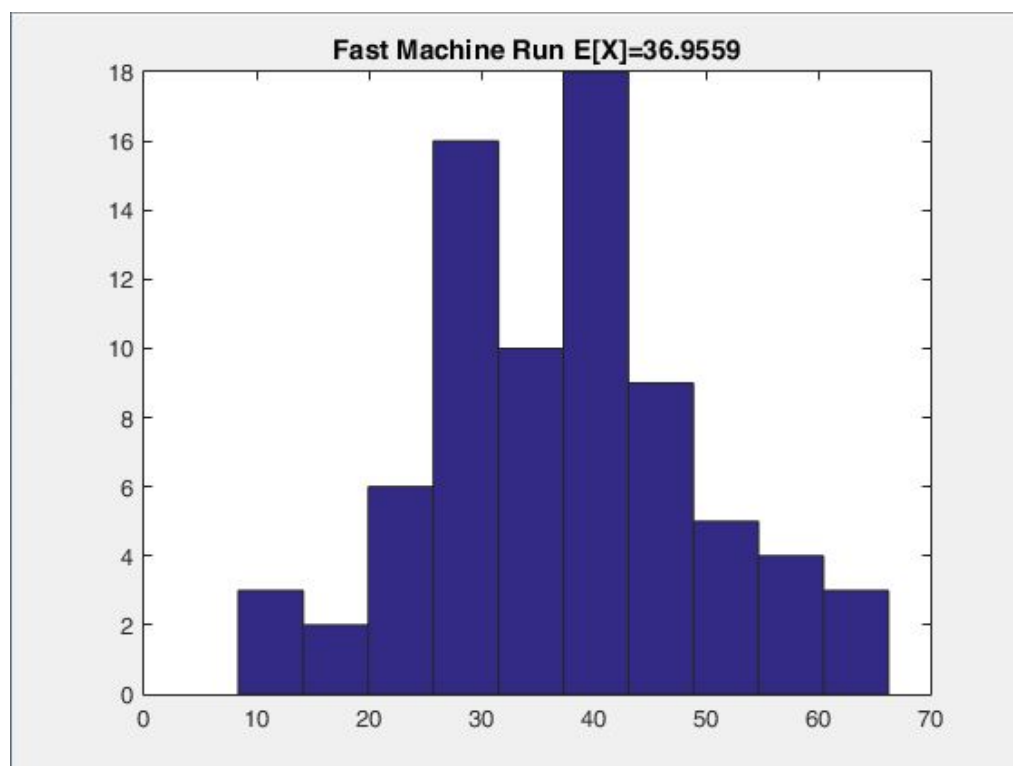
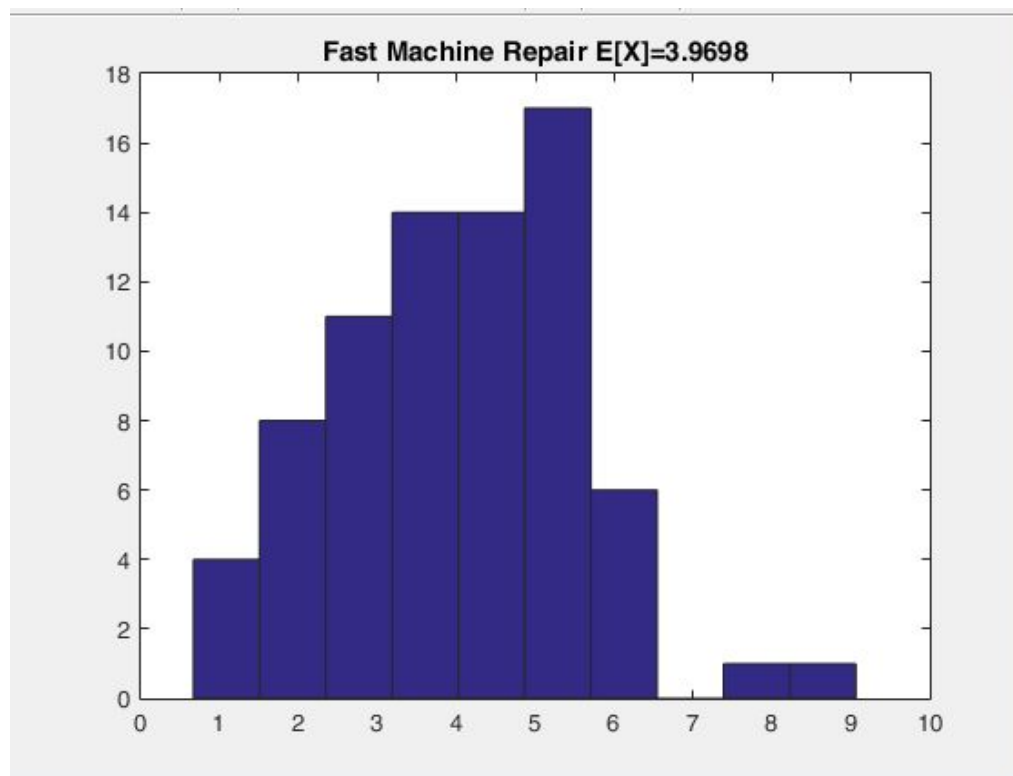
Beta seemed to strongly determine where the distributions were most “concentrated” with a direct link to the value. For example, a $\beta=10$ led to the peak of the distribution being basically at $x=10$. With this considered I only am going to post the variations in α because the differences are more significant. Beta has been set to 10 for all the graphs

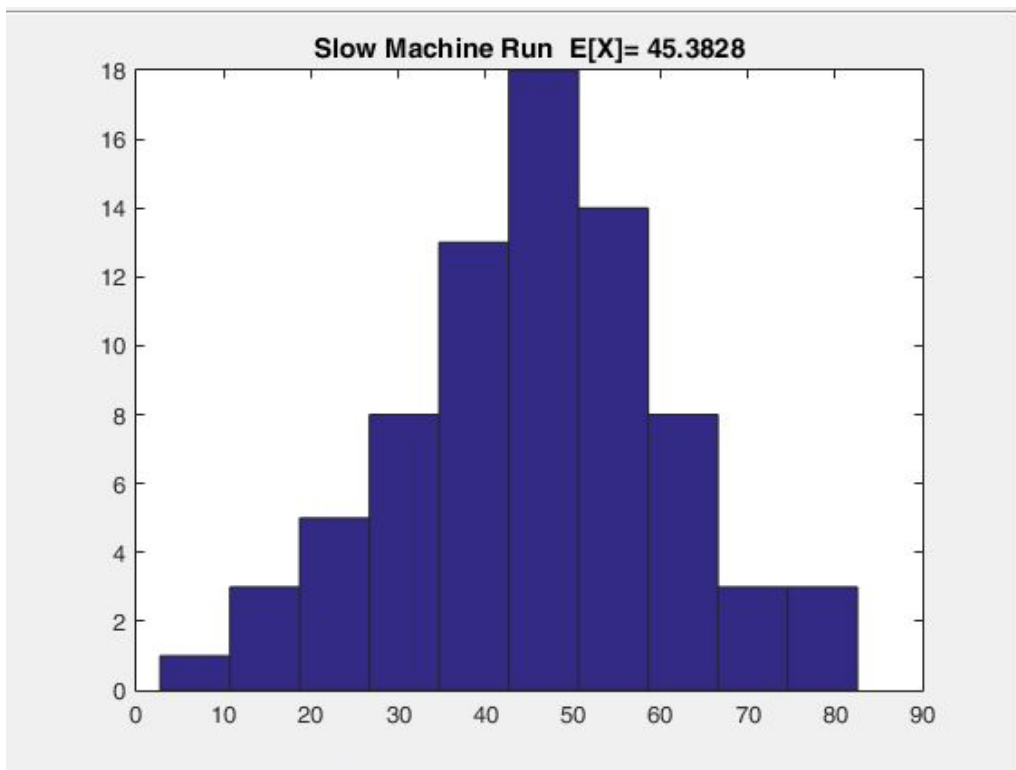
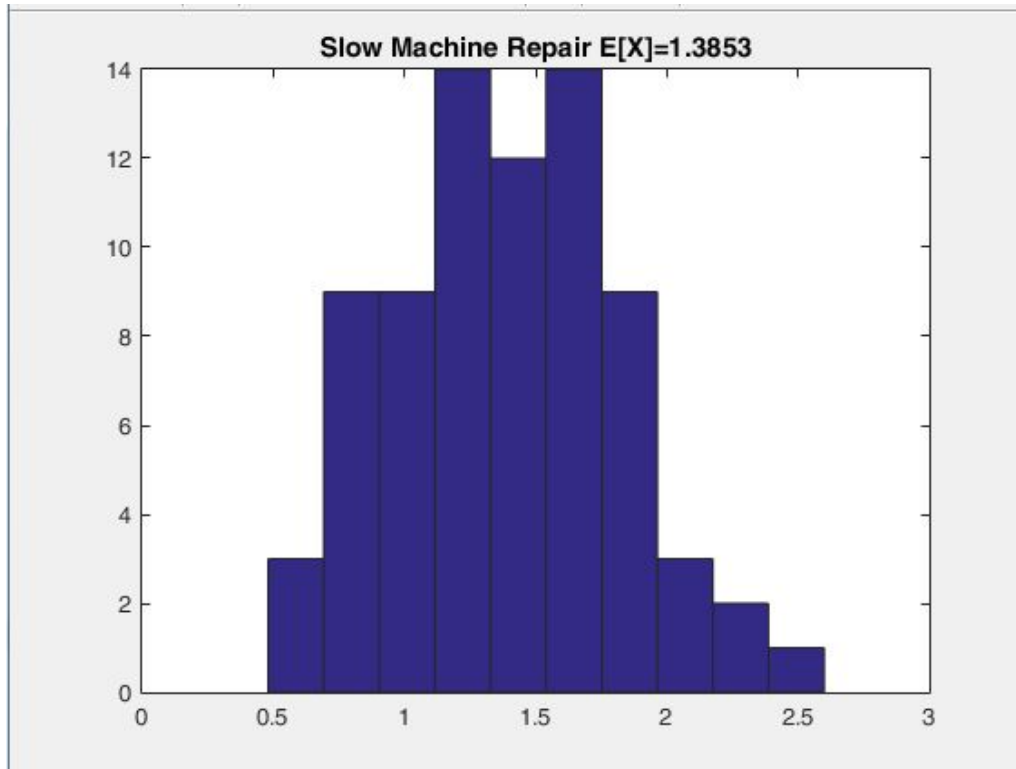






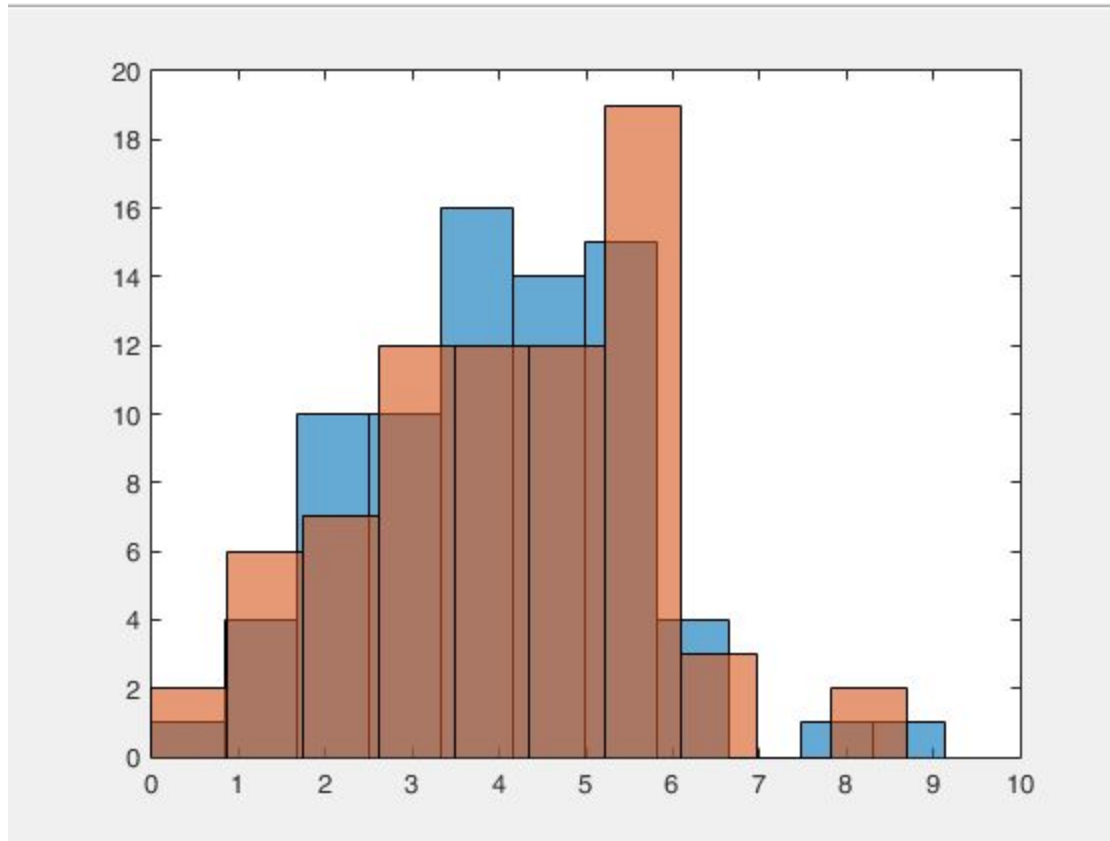






FMRepair: ==> Alpha=3.1, Beta=4.5

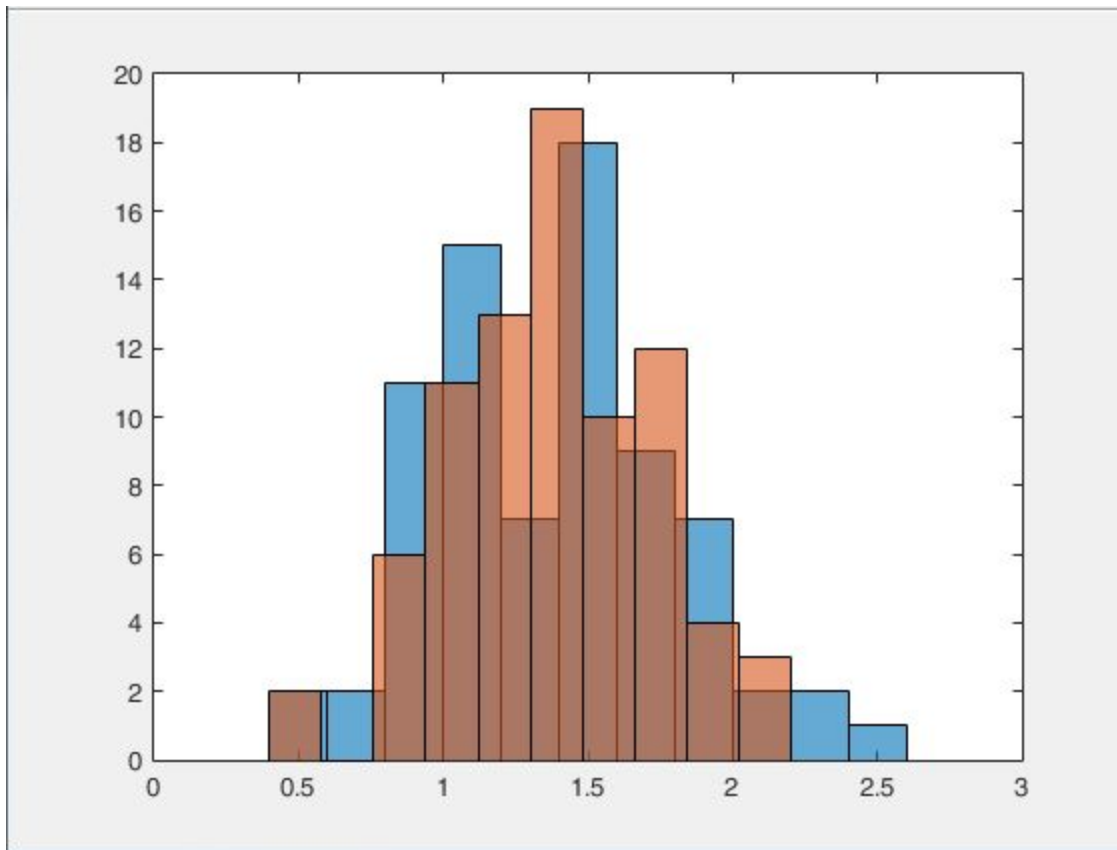
The blue is the data we were provided, the orange is the weibull



We also determined the Mean of the Weibull function with the corresponding alpha and beta.

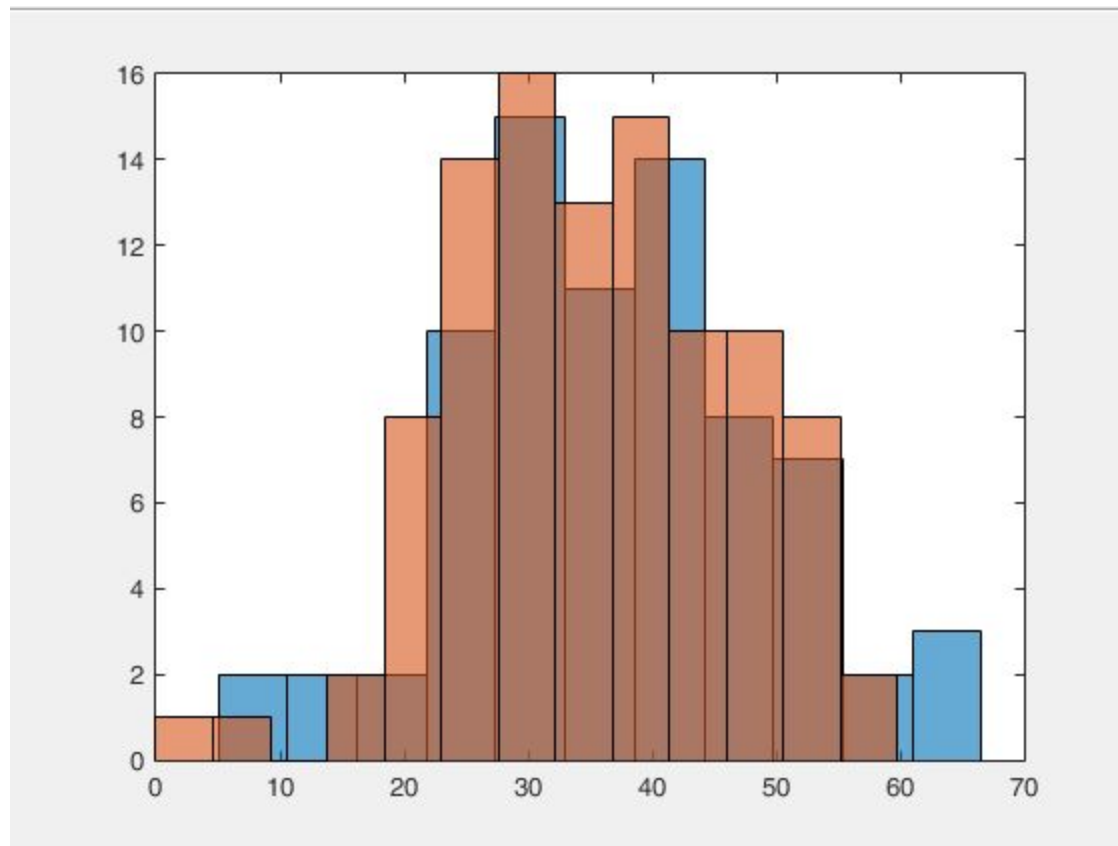
Mean = 4.024372682600938

SMRepair : Alpha = 5.5 Beta= 1.5



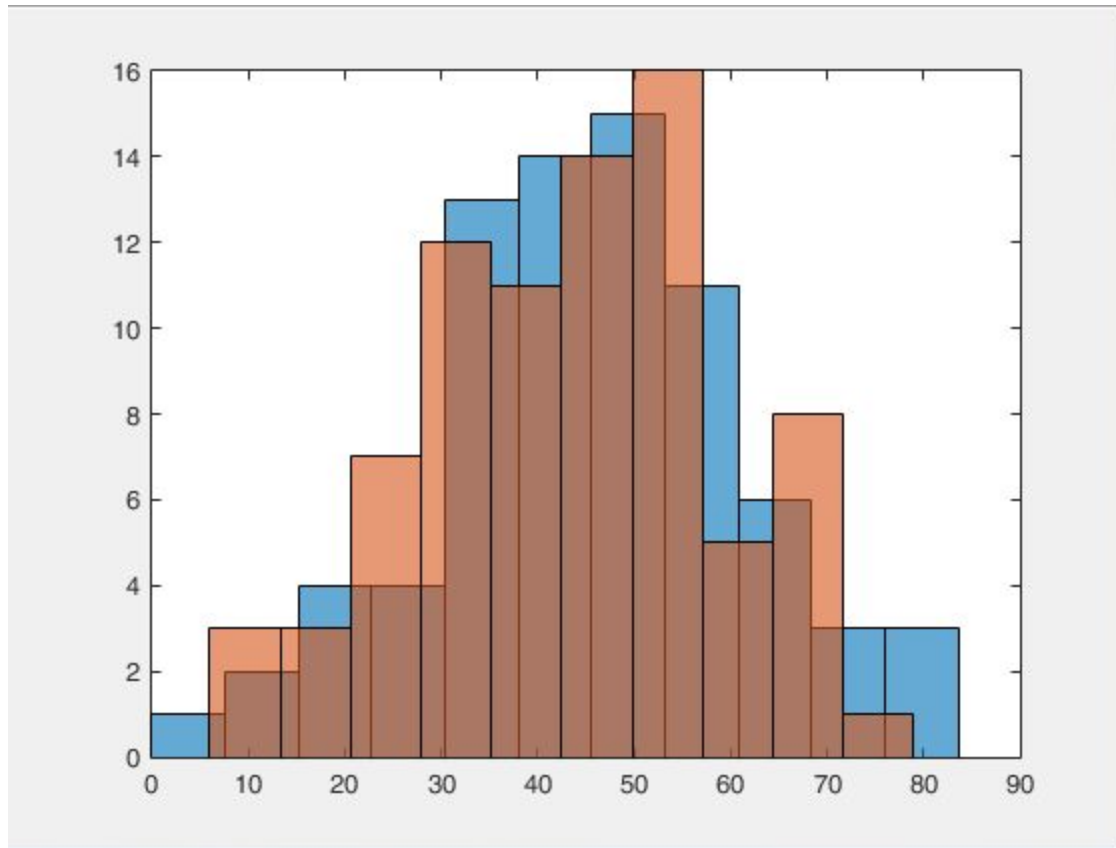
Mean = 1.384802137415484

FMRun : Alpha=3.64 Beta=41



Mean = 36.967496888114745

SMRun : Alpha = 3.1 Beta=50.8



Mean = 45.430696061361694

Final report

Here at Make CSCI 164 Great Again Inc., our biggest goal is to get the most cash money in our customer's pockets as possible! We do extensive research and use the latest technology to make sure your plans run as smoothly and efficiently as you need them to.

This is where you come in, AppleBee Co.! You presented us with your factory dilemma and we are happy to say we have come up with a solution so you and your company can pump out as many honey apple pies as machine-ly possible.

We gave our highly skilled team of computer simulators the rates and costs you provided us and they have successfully created an algorithm and computer program that represents how your factory would run with every combination of fast and slow machines. The team made a variable run time so their program can show you how your factory would perform if it ran for one hour or millions of hours. After many trials, we are confident we have found an accurate representation of your factory because the numbers remain reasonable and constant whether closing time is 10 or 10,000,000. Sample run outputs have been attached to the end of this report.

Our team developed these simulations to include a variety of pertinent parameters. Our goal was to test combinations of fast and slow varieties of your seven machines, so this proportion of fast and slow machines was the primary parameter our simulations were based on. We also included the distribution parameters Appleby Co. provided to us, such as the rate of production, likelihood of breakdown, and repair costs for each machine. We kept track of the number of total profit made, the total cost of repairs, the net profit, and the total wait time for repairing machines, as well as the time each machine exits and enters production.

While the end of period profit numbers provide a very clear pattern, we decided to do a more in-depth analysis and discovered an interesting trend among the number of machines. When comparing the net profit between the first case, 0 fast machines and 7 slow machines, and the second case, 1 fast machine and 6 slow machines, we see a profit gain of 7.664% but also a 27.347% increases in total down time. These differences are smaller in the comparison of the final cases with profit increasing by 3.339% and down time increasing by 22.212%. It is currently unclear as to how this decreasing trend will continue past the restraints of 7 total machines. The decrease may bottom out at around 0% since the number of slow machines will get trivially low, or the fast machines could be unstable in a way that is difficult to see with only 7 machines.

It is also worth noting the comparison between the first and last cases, of 0 fast machines to all fast machines, in which we see a 45.180% increase in profit but a 441.381% increase in down time. This is where the subjective business analysis comes into play rather than the objective numerical analysis. If your business model is that of bulk orders placed in large intervals so you can have a large block of time to fulfill your orders, this down time could be negligible and the increase in end of period profit would be the only really important factor so the 7 fast machines is the clear winner. On the other hand, if your business model depends more on constant availability, for example in the case of multiple small orders that need to be filled continuously, the increases in down time could potentially lead to missed deadlines which could lead to lost customers, hurting your sales. In this case it might be optimal to take a more conservative route that includes some slow machines.

We here at Make CSCI 164 Great Again Inc. are all about profit. It is clear from the simulated outputs that your repair times and net profit continue to grow as more fast machines are added. We have compared the repair times for different combinations of machines¹ and discovered that although slow machines are being repaired for a fraction of the time, they also produce a relatively smaller fraction of the revenue. Since the fast machines' production rates are so high, they more than cover their repair costs. So it is our opinion that AppleBee Co. would benefit most from buying all 7 fast machines. Our team strongly believes these machines are well worth the investment.

¹ - Net Profit between 6 & 7 fast machines: $1.76417e+08 / 1.70717e+08 = 1.0333885904742$
- Down time between 6 & 7 fast machines: $1.00743e+06 / 824331 = 1.2221182995665$
- Net Profit between 0 & 1 fast machines : $1.30829e+08 / 1.21516e+08 = 1.0766401132361$
- Down time between 0 & 1 fast machines: $290663 / 228245 = 1.2734692983416$
-Net Profit between 0 & 7 fast machines: $1.76417e+08 / 1.21516e+08 = 1.4518005859310$
-Down time between 0 & 7 fast machines : $1.00743e+06 / 228245 = 4.4138097220092$

Sample Run Outputs

100 hours:

With 0 fast machines and 7 slow machines, over a time length of 100, we had

- Total Costs = 34.9522
- Total Revenue = 12510.6
- Net Profit = 12475.6
- Machine Down Time = 20.5909

With 1 fast machines and 6 slow machines, over a time length of 100, we had

- Total Costs = 68.1177
- Total Revenue = 13365.8
- Net Profit = 13297.6
- Machine Down Time = 22.1746

With 2 fast machines and 5 slow machines, over a time length of 100, we had

- Total Costs = 145.79
- Total Revenue = 14319.1
- Net Profit = 14173.4
- Machine Down Time = 31.3039

With 3 fast machines and 4 slow machines, over a time length of 100, we had

- Total Costs = 135.138
- Total Revenue = 16092.7
- Net Profit = 15957.6
- Machine Down Time = 29.283

With 4 fast machines and 3 slow machines, over a time length of 100, we had

- Total Costs = 106.462
- Total Revenue = 16304.2
- Net Profit = 16197.8
- Machine Down Time = 45.335

With 5 fast machines and 2 slow machines, over a time length of 100, we had

- Total Costs = 255.674
- Total Revenue = 16776.6
- Net Profit = 16521
- Machine Down Time = 65.9012

With 6 fast machines and 1 slow machines, over a time length of 100, we had

- Total Costs = 208.963
- Total Revenue = 19878.5
- Net Profit = 19669.5
- Machine Down Time = 78.3785

With 7 fast machines and 0 slow machines, over a time length of 100, we had

- Total Costs = 273.497
- Total Revenue = 18207.3
- Net Profit = 17933.8
- Machine Down Time = 107.577

1,000,000 hours

With 0 fast machines and 7 slow machines, over a time length of 1e+06, we had

- Total Costs = 377089
- Total Revenue = 1.21893e+08
- Net Profit = 1.21516e+08
- Machine Down Time = 228245

With 1 fast machines and 6 slow machines, over a time length of 1e+06, we had

- Total Costs = 759534
- Total Revenue = 1.31589e+08
- Net Profit = 1.30829e+08

-Machine Down Time = 290663

With 2 fast machines and 5 slow machines, over a time length of 1e+06, we had

-Total Costs = 1.16101e+06

-Total Revenue = 1.40991e+08

-Net Profit = 1.3983e+08

-Machine Down Time = 363511

With 3 fast machines and 4 slow machines, over a time length of 1e+06, we had

-Total Costs = 1.57603e+06

-Total Revenue = 1.49984e+08

-Net Profit = 1.48408e+08

-Machine Down Time = 450014

With 4 fast machines and 3 slow machines, over a time length of 1e+06, we had

-Total Costs = 2.01016e+06

-Total Revenue = 1.58512e+08

-Net Profit = 1.56502e+08

-Machine Down Time = 552940

With 5 fast machines and 2 slow machines, over a time length of 1e+06, we had

-Total Costs = 2.45573e+06

-TotalRevenue = 1.66519e+08

-Net Profit = 1.64063e+08

-Machine Down Time = 673352

With 6 fast machines and 1 slow machines, over a time length of 1e+06, we had

-Total Costs = 2.90861e+06

-Total Revenue = 1.73626e+08

-Net Profit = 1.70717e+08

-Machine Down Time = 824331

With 7 fast machines and 0 slow machines, over a time length of $1e+06$, we had

-Total Costs = $3.36194e+06$

-Total Revenue = $1.79779e+08$

-Net Profit = $1.76417e+08$

-Machine Down Time = $1.00743e+06$