Week 8 Assignment - DOE and Probability Distributions

Omer Farooq (EDx ID: mfarooq4)

03/10/2020

Table of Contents

[QUESTION 12.1 1](#_Toc34716991)

[QUESTION 12.2 1](#_Toc34716992)

[QUESTION 13.1 3](#_Toc34716993)

[QUESTION 13.2 3](#_Toc34716994)

# QUESTION 12.1

**Describe a situation or problem from your job, everyday life, current events, etc., for which a design of experiments approach would be appropriate.**

At my job at the T-Mobile HQ in the Seattle area, my team helps build analytics products and analysis for our network supply chain team. The Supply chain team manages the planning, procurement and logistics of getting the right equipment (radios, antennas etc.) to the right locations so that T-Mobile’s network could get built or improved. As part of our process, every week, we have to allocate the material from the available inventory to the short-list of high-priority projects from across regions in the US. The short-listed projects are almost always more than the available inventory. This would be a good design of experience candidate where several combinations of projects from different regions could be tested to make sure 1) the inventory consumption is maximized, 2) the regions get the equal share of projects and 3) the right mix of programs (new sites/towers, enhancements etc.) are picked.

There could be hundreds of combinations from the short-listed projects every week and a factorial design would be good tool to use based on different factors related to the projects (program project is part of, location, material need etc.).

# QUESTION 12.2

**To determine the value of 10 different yes/no features to the market value of a house (large yard, solar roof, etc.), a real estate agent plans to survey 50 potential buyers, showing a fictitious house with different combinations of features. To reduce the survey size, the agent wants to show just 16 fictitious houses. Use R’s FrF2 function (in the FrF2 package) to find a fractional factorial design for this experiment: what set of features should each of the 16 fictitious houses have? Note: the output of FrF2 is “1” (include) or “-1” (don’t include) for each feature.**

The FrF2 function in the FrF2 package (<https://www.rdocumentation.org/packages/FrF2/versions/2.1/topics/FrF2>) provides an easy ability to do fractional fractorial design. I first built a string of 10 possible features of the house and then used the FrF2 function to run a fractional factorial design.

library(FrF2)  
  
set.seed(101)  
  
features = c("openconcept", "steelappliances", "2cargarage", "largeyard", "solarroof","basement", "goodschools", "bigyard", "gym", "woodfloors")  
  
FrF2(nruns = 16, factor.names = features)

## openconcept steelappliances X2cargarage largeyard solarroof basement  
## 1 -1 -1 -1 1 1 1  
## 2 1 1 1 1 1 1  
## 3 1 -1 1 1 -1 1  
## 4 -1 1 1 -1 -1 -1  
## 5 -1 -1 -1 -1 1 1  
## 6 1 -1 -1 1 -1 -1  
## 7 1 -1 1 -1 -1 1  
## 8 -1 1 -1 -1 -1 1  
## 9 1 1 1 -1 1 1  
## 10 -1 1 1 1 -1 -1  
## 11 1 1 -1 1 1 -1  
## 12 -1 -1 1 1 1 -1  
## 13 -1 -1 1 -1 1 -1  
## 14 1 -1 -1 -1 -1 -1  
## 15 1 1 -1 -1 1 -1  
## 16 -1 1 -1 1 -1 1  
## goodschools bigyard gym woodfloors  
## 1 1 -1 1 -1  
## 2 1 1 1 1  
## 3 -1 1 -1 -1  
## 4 1 1 -1 1  
## 5 1 1 -1 1  
## 6 1 1 1 1  
## 7 -1 -1 1 1  
## 8 -1 1 1 -1  
## 9 1 -1 -1 -1  
## 10 1 -1 1 -1  
## 11 -1 1 -1 -1  
## 12 -1 -1 -1 1  
## 13 -1 1 1 -1  
## 14 1 -1 -1 -1  
## 15 -1 -1 1 1  
## 16 -1 -1 -1 1  
## class=design, type= FrF2

The output of the function provided different 16 designs based on 10 factors for the house.

# QUESTION 13.1

**For each of the following distributions, give an example of data that you would expect to follow this distribution (besides the examples already discussed in class).**

* **a. Binomial** Binomial distribution is simply the probability of success or failure in an experiment that is repeated multiple times. Lottery is a good example. You either win a lottery or you don’t. Let’s say if probability of winning on the lottery ticket is 0.30 and 20 lottery tickets are purchases, binomial distribution can tell us the probability of winning at least 1 (or 2 or 3 etc.) lottery tickets.
* **b. Geometric** Geometric distribution represents number of failures (or successes) before the first success (or failure) in a series of trials. The number days before an accidents or incidents happens at a manufacturing facility would follow a geometric distribution. It would help answer that how many days without an accident (successes) before an accident (failure) happens.
* **c. Poisson** A Poisson distribution describes the probability discrete events ina time period where the average time between events is known, but the exact timing of events is random. Number of visitors to any website like Google etc. would follow poisson distribution.
* **d. Exponential** The time between the events mentioned above in Poisson process are described by the exponential distribution. The time between the visitors to a given website from the example above might follow an exponential distribution.
* **e. Weibull** Weibull distribution helps with time to an event (success or failure). For example, life of a cell phone (or time before the user might have to switch cell phones due to cell phone completely wearing out) might follow a weibull distribution with k>1 given as failure rate would be expected to increase over time as the device ages.

# QUESTION 13.2

**In this problem you, can simulate a simplified airport security system at a busy airport. Passengers arrive according to a Poisson distribution with λ1 = 5 per minute (i.e., mean interarrival rate 1 = 0.2 minutes) to the ID/boarding-pass check queue, where there are several servers who each have exponential service time with mean rate 2 = 0.75 minutes. [Hint: model them as one block that has more than one resource.] After that, the passengers are assigned to the shortest of the several personal-check queues, where they go through the personal scanner (time is uniformly distributed between 0.5 minutes and 1 minute).**

**Use the Arena software (PC users) or Python with SimPy (PC or Mac users) to build a simulation of the system, and then vary the number of ID/boarding-pass checkers and personal-check queues to determine how many are needed to keep average wait times below 15 minutes. [If you’re using SimPy, or if you have access to a non-student version of Arena, you can use λ1 = 50 to simulate a busier airport.]**

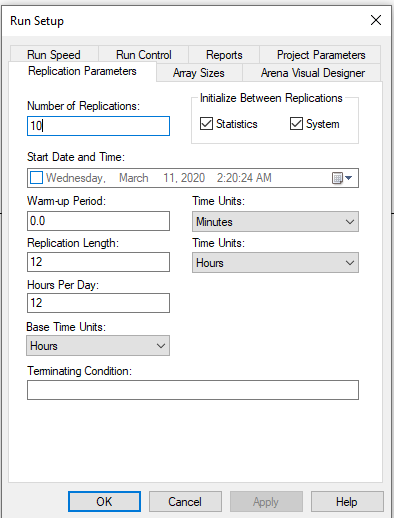
**ARENA SOFTWARE**

First, I used Arena Software to build the simulation and checked multiple scenarios. Summary of my scenarios is below:

1. 1 ID checking process with 1 resource (4 capacity) and 1 scanner (with 4 capacity)
2. 1 ID checking process with 1 resources (4 capacity) and 2 scanners (with 1 resource ea with 2 capacity) with decision logic to route passenger to shortest scanner queue.
3. And I continued increasing ID checkers and scanners to see when wait time became lower than 15 mins.

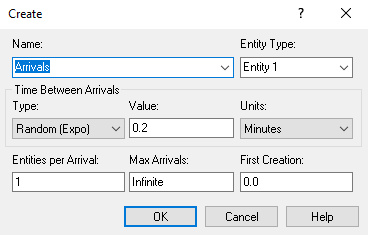
**Run Set-up:**

I used the set-up below for all simulation runs.

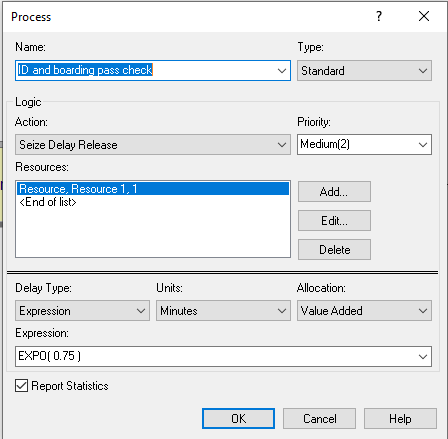


**Simulation Set-up**

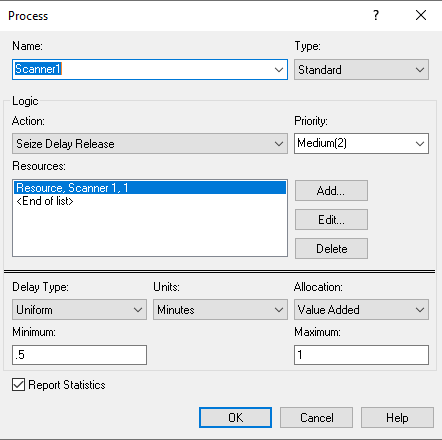
System Arrival



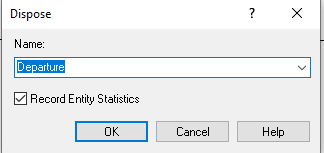
ID Checker Process (did not add resources to the process, rather increased the resource’s capacity to get passed free software limit – this guidance was provided in the office hours on Mar 9th)



Scanner process



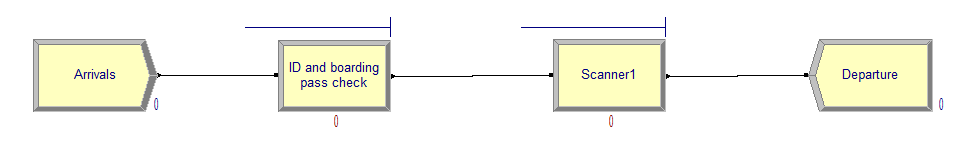
Leave System

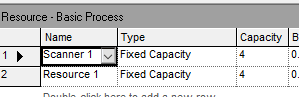


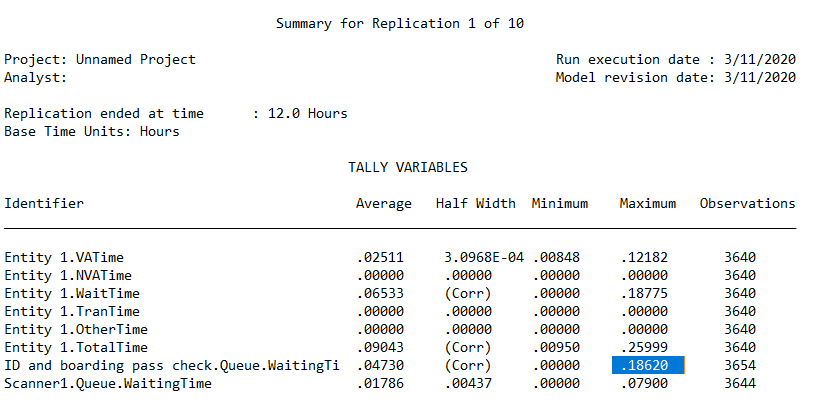
**Output**

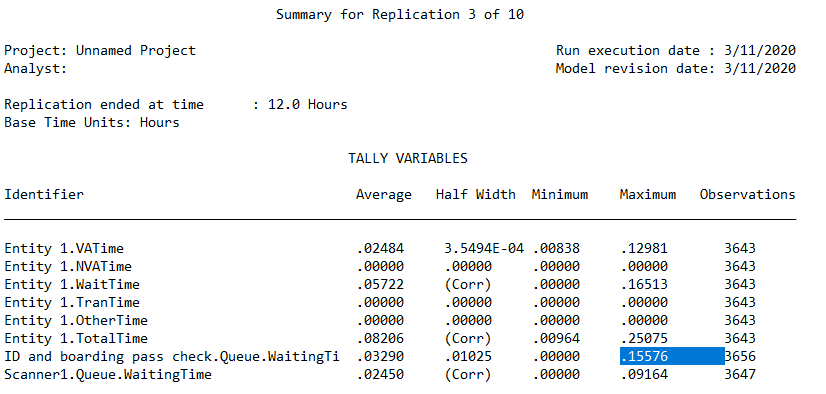
* **1 Scanner**

The image of the simulation model and some screenshots from output are shown. Several replications had over 15 mins of wait times. I used capacity of 4 ea for both ID checking and scanner process resources.





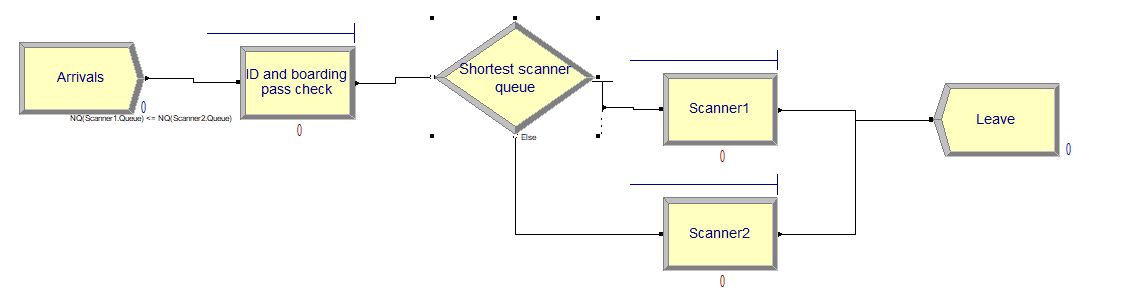


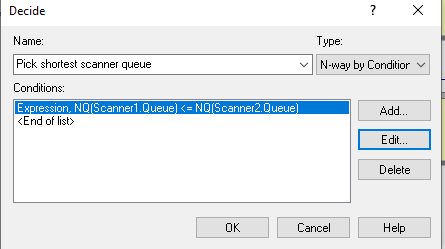


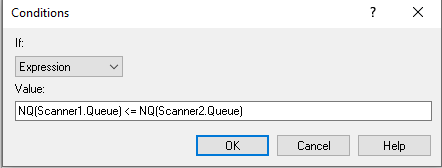
Plus 2 other replications had >15mins wait times.

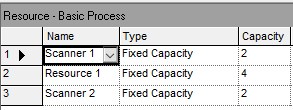
* **2 Scanners**

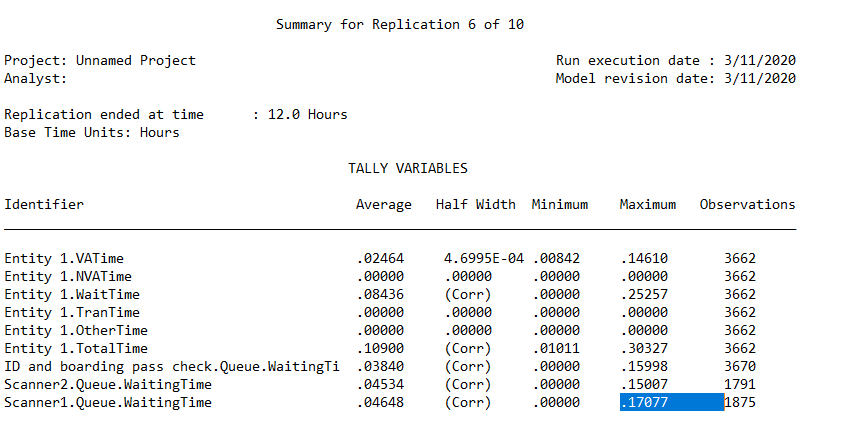
This option also showed a few >15 min wait times adding the max wait times for ID checker and scanners. Even though the avg wait times were within 15 min, a few >15 mins max values make it a less optimum option. Thus, I ran another option with 3 scanners next. I used capacity of 4 for the ID checker and 2 ea for scanners in this run to get past 150 entities limit.

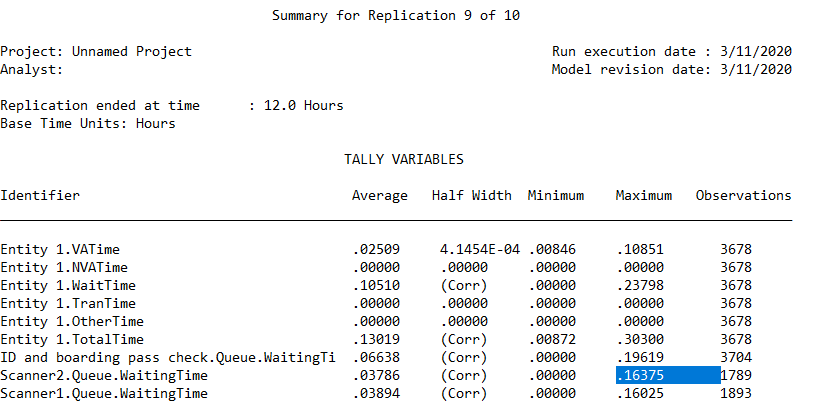
****

****

****

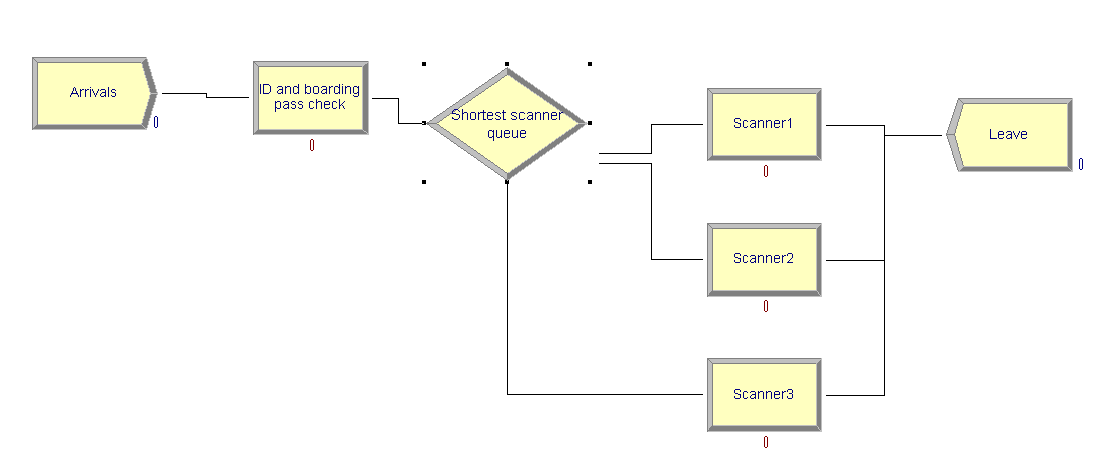
****

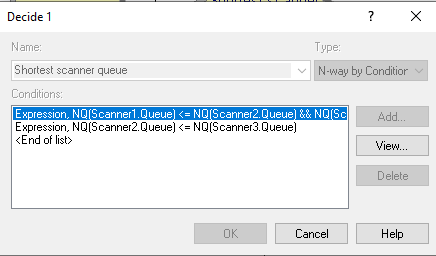
****

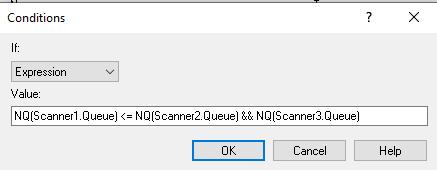
****

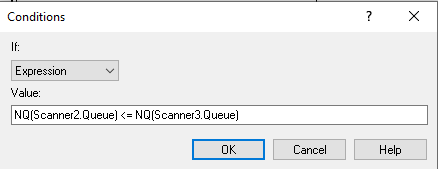
* **3 Scanners**

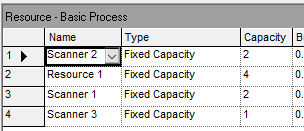
I ran this simulation with 3 scanners. I used capacity of 4 for the ID checker and 2 ea for 2 of the scanners and 1 for the 3rd one to get past 150 entities limit.











The average wait times were all less than 15 mins for this run. Details output file is attached to this assignment.

Thus, final simulation reveals that 1 ID/boarding pass check post with capacity of 4 and 3 scanners gets us lower than 15 min wait times on average.

Finally, I tried Poisson lambda = 50 (50 arrivals per minute or 0.02 mean interarrival rate) on my final model and even with increasing the capacity many folds, I could not make the model to run due to 150 entities limit on the trial version (I am not an OMSA student yet so could not get that license). Logic says that increasing arrival rate 10 folds will increase the number of ID/boarding pass and scanners needed many folds as well (probably to 35-40).