ISYE 6051 : Homework 9
3/24/2021

Table of Contents

12.1 Design of Experiments (DoE)

DoE is used to evaluate factors where the value of the associated parameter can be controlled. It is a cause-and-effect design to determine if 1 or more parameters are controlled what will be the response. There are also factors that cannot be controlled that impact the response but, these also should be recognized as best possible to understand their impact. One hypothesis that could be tested for is that patients that have time slots with a 15-minute interval cadence are moved through the total arrive to leave process within 30 minutes.

The real-life example that I would create a design for is determining the length of time from arrival at a vaccine site to departure for patients. Patients may be concerned with wait times and avoid scheduling a vaccine if they believe that they will not be able to stay for extended periods of time.

The controllable factors that I would look at are:

1) ***Arrival time of the participant***
    a. ***Early arrival***
    b. ***On-time arrival***
    c. ***> 10 minutes late***
2) ***Number of check-in staff available***
3) ***Number of medical staff available to administer the vaccine***
4) ***Appointment cadence***
    a. ***Patients have specific time slots in 15-minute intervals***
    b. ***Patients have specific time slots in 30-minute intervals***
    c. ***Patients are asked to arrive within a 2-hour window***
    d. ***Patients are asked to arrive within a 4-hour window.***

There can be uncontrolled factors such as the time it takes to actually administer the vaccine to a patient based on the patient's fears (e.g., will the patient hold still for the shot) and whether a patient has an adverse reaction causing them to need to stay longer for observation.

12.2 Fractional Factoral Design

This solution called for creating an experimental design subset of 50 houses (16 or 32%) and evaluate 10 factors for use in determining market value. The FrF2 functional was used to create a list for relators. In this solution, for ease of reading, "yes" and "no" were explicity defined instead of 1 and -1.

CODE:

```
rm(list = ls())
set.seed(42)

#library updates
install.packages("FrF2")
require(FrF2)

# There are 10 different yes/no factors features) to determine
# the response(market price) of a home.

HouseTypes <- FrF2(nruns = 16, nfactors = 10,
                   factor.names = c("SkyLight","GameRoom","Office","TheaterRoom",
                                    "Pool","Sauna","HalfBath","Gym","Landscaping",
                                    "DogRun"),
                   default.levels = c("yes","no"))
HouseTypes
```

OUTPUT:

```
HouseTypes
```

	Skylight	GameRoom	Office	TheaterRoom	Pool	Sauna	HalfBath	Gym	Landscaping	DogRun
1	no	no	no	yes	no	no	no	yes	yes	yes
2	yes	yes	no	yes	no	yes	yes	no	no	no
3	yes	yes	no	yes	no	yes	yes	no	no	yes
4	no	yes	yes	no	yes	yes	no	no	no	no
5	no	yes	no	no	yes	no	yes	no	yes	yes
6	yes	yes	yes	yes	no	no	no	no	yes	no
7	yes	no	yes	yes	yes	no	yes	no	no	yes
8	no	yes	no	yes	yes	no	yes	yes	no	no
9	no	no	no	no	no	no	no	no	no	no
10	yes	yes	no	no	no	yes	yes	yes	yes	no
11	yes	yes	yes	no	no	no	no	yes	no	yes
12	yes	no	yes	no	yes	no	yes	yes	yes	no
13	yes	no	no	yes	yes	yes	no	no	yes	no
14	no	no	yes	no	no	yes	yes	no	yes	yes
15	no	yes	yes	yes	yes	yes	no	yes	yes	yes

16	yes	no	no	no	yes	yes	no	yes	no	yes

class=design, type= FrF2

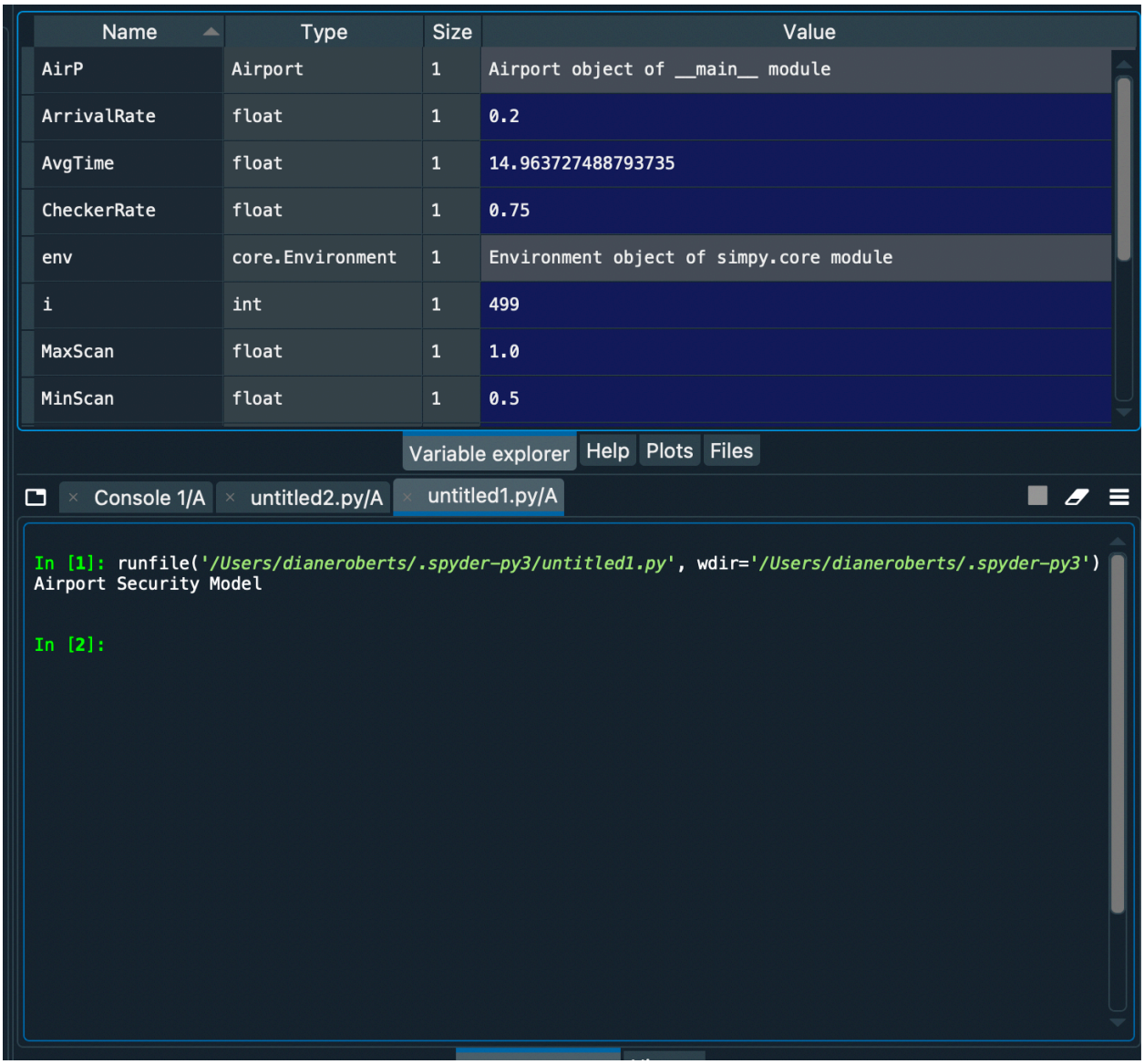
13.1 Distribution Data Examples

a. ***Binomial – In Binomial distribution the determination is based on the number of trials needed before a desired response is achieved.***
    a. ***Data example – 2 runners will run a 100-yard dash 6 times. Define the probability (number of times) that runner A will win a trial.***
b. ***Geometric – In Geometric distribution the probability is determine based on the number of individual data points needing to be tested before achieving the desired outcome.***
    a. ***Probability of the number of dogs that I will need to view in a kennel on a given day before finding one that has a coat that is at least 75% white.***
c. ***Poisson – Poisson distribution determines the probability of the number of a particular item in a pool of data generally related to time, geography, etc.***
    a. ***Determine the probability of sighting a Monarch butterfly during one day in September.***
d. ***Exponential – Exponential distribution determines the probability distribution for the time between events.***
    a. ***Probability for the amount of time it takes for a customer to be seated at a restaurant during the peak hours of 7-9pm on a Friday evening.***
e. ***Weibull – Weibull distribution determines the amount of time between failures. This is generally used for reliability studies.***
    a. ***Probability on the amount of time that it will take for a wind turbine to fail after installation and between repairs.***

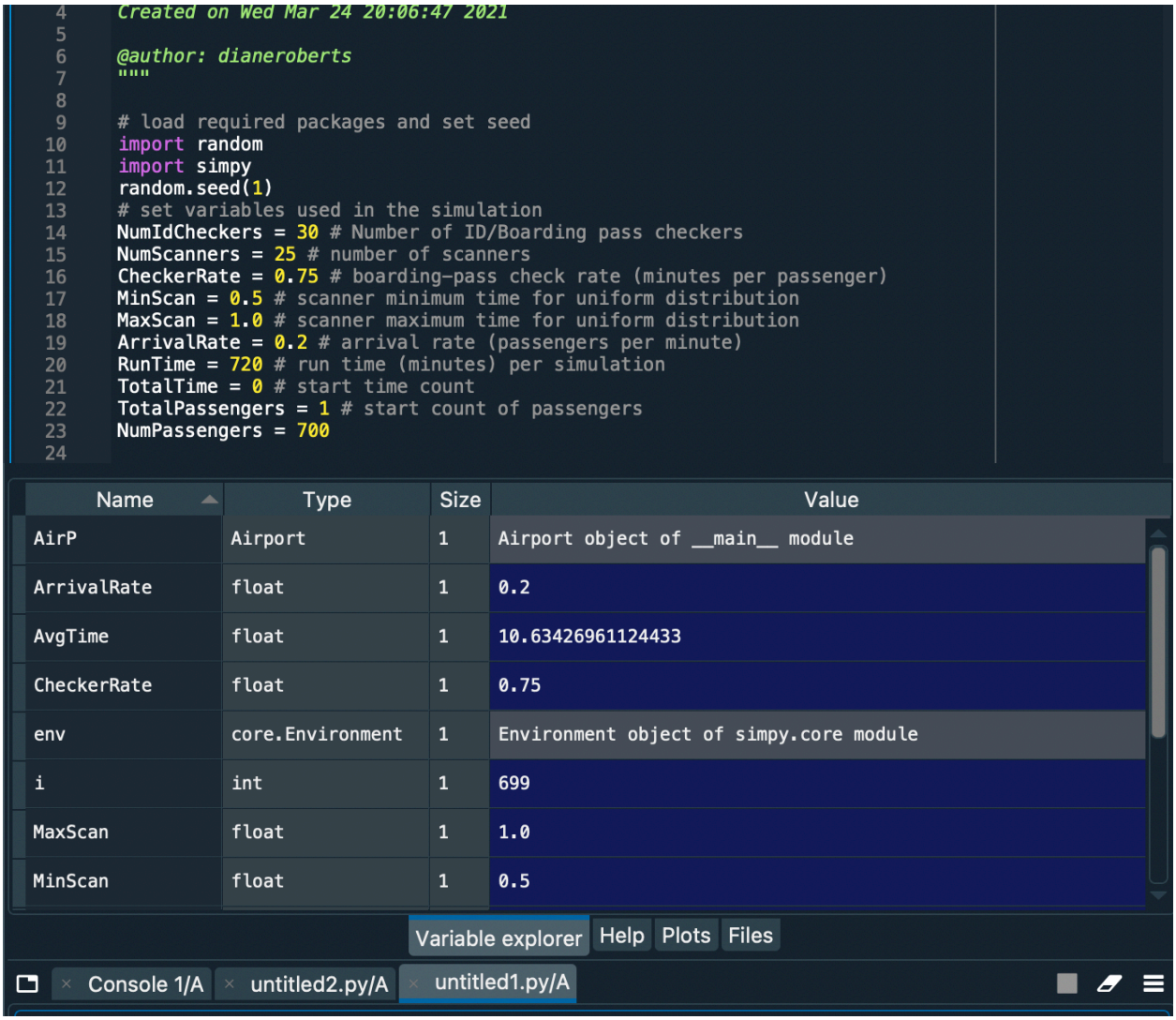
13.2 Simulation

Using Python I tried a few variations to get the maximum number of customers through within the 15 minute wait time.

```
7   """
8
9   # load required packages and set seed
10  import random
11  import simpy
12  random.seed(1)
13  # set variables used in the simulation
14  NumIdCheckers = 9 # Number of ID/Boarding pass checkers
15  NumScanners = 13 # number of scanners
16  CheckerRate = 0.75 # boarding-pass check rate (minutes per passenger)
17  MinScan = 0.5 # scanner minimum time for uniform distribution
18  MaxScan = 1.0 # scanner maximum time for uniform distribution
19  ArrivalRate = 0.2 # arrival rate (passengers per minute)
20  RunTime = 1000 # run time (minutes) per simulation
21  TotalTime = 0 # start time count
22  TotalPassengers = 1 # start count of passengers
23  NumPassengers = 500
24
```


Name
Type
Size
Value
AirP
Airport
1
Airport object of __main__ module
ArrivalRate
float
1
0.2
AvgTime
float
1
14.963727488793735
CheckerRate
float
1
0.75
env
core.Environment
1
Environment object of simpy.core module
i
int
1
499
MaxScan
float
1
1.0
MinScan
float
1
0.5
Variable explorer
Help
Plots
Files
Console 1/A
untitled2.py/A
untitled1.py/A
In [1]: runfile('/Users/dianeroberts/.spyder-py3/untitled1.py', wdir='/Users/dianeroberts/.spyder-py3')
Airport Security Model
In [2]:

Using more scanners and boarding pass checkers allowed more passengers through the process. However, what this simulation does not account for is determination of the maximum cost that is acceptable for checkers and additional scanners.



```
4   Created on Wed Mar 24 20:06:47 2021
5
6   @author: dianeroberts
7   """
8
9   # load required packages and set seed
10  import random
11  import simpy
12  random.seed(1)
13  # set variables used in the simulation
14  NumIdCheckers = 30 # Number of ID/Boarding pass checkers
15  NumScanners = 25 # number of scanners
16  CheckerRate = 0.75 # boarding-pass check rate (minutes per passenger)
17  MinScan = 0.5 # scanner minimum time for uniform distribution
18  MaxScan = 1.0 # scanner maximum time for uniform distribution
19  ArrivalRate = 0.2 # arrival rate (passengers per minute)
20  RunTime = 720 # run time (minutes) per simulation
21  TotalTime = 0 # start time count
22  TotalPassengers = 1 # start count of passengers
23  NumPassengers = 700
24
```

Name	Type	Size	Value
AirP	Airport	1	Airport object of __main__ module
ArrivalRate	float	1	0.2
AvgTime	float	1	10.63426961124433
CheckerRate	float	1	0.75
env	core.Environment	1	Environment object of simpy.core module
i	int	1	699
MaxScan	float	1	1.0
MinScan	float	1	0.5

Variable explorer | Help | Plots | Files

Console 1/A | untitled2.py/A | untitled1.py/A

Likewise, the desired wait time is not achieved with even small reductions in scanners and checkers.

```
7   """
8
9   # load required packages and set seed
10  import random
11  import simpy
12  random.seed(1)
13  # set variables used in the simulation
14  NumIdCheckers = 7 # Number of ID/Boarding pass checkers
15  NumScanners = 10 # number of scanners
16  CheckerRate = 0.75 # boarding-pass check rate (minutes per passenger)
17  MinScan = 0.5 # scanner minimum time for uniform distribution
18  MaxScan = 1.0 # scanner maximum time for uniform distribution
19  ArrivalRate = 0.2 # arrival rate (passengers per minute)
20  RunTime = 1000 # run time (minutes) per simulation
21  TotalTime = 0 # start time count
22  TotalPassengers = 1 # start count of passengers
23  NumPassengers = 500
```

Name	Type	Size	Value
AirP	Airport	1	Airport object of __main__ module
ArrivalRate	float	1	0.2
AvgTime	float	1	19.452710367803448
CheckerRate	float	1	0.75
env	core.Environment	1	Environment object of simpy.core module
i	int	1	499
MaxScan	float	1	1.0
MinScan	float	1	0.5

Variable explorer | Help | Plots | Files

Console 1/A | untitled2.py/A | untitled1.py/A