IE306-02

Assignment-3 Group1
2009400189-2015400183-2015400132

Model Logic & Assumptions:

In this project, we are given the task of simulating a cinema that has three theaters each of which shows a different movie. We are initially provided with a sample model that works for two movies and our task is to improve it so that it could simulate the case where the cinema shows three different movies in different theaters. Another task is to convert the model into a model that has a personnel for each movie that gives the customers their tickets after possibly waiting in a specific queue for the movie they fancy to watch. Now, below, we will be explaining the logic behind our model for both cases.

1-Counter Model:

Before going into the details, it is important to note that we wanted the customers who would desire to watch a movie that is already sold out before they come to the cinema to quit the system immediately. For this reason, the model initially checks whether the movie that customers desire to watch is sold out or not. And if so, customers are disposed of.

After this process, customers get in the queue if the counter is busy. If the counter is not busy, then the customer seizes the counter. According to the movie number, the customer follows a specific path. For each customer, there are 3 attributes each of which indicates whether the customer has attempted to get tickets from a movie. Thus, after seizing the counter, the corresponding attribute is marked so that the model knows that the customer has already attempted to watch that movie. This is useful in the case where the movie initially selected is not sold out but there are also not enough tickets since the customer would like to select another movie to watch.

Afterwards, the model checks whether there are enough tickets for the movie selected by the customer. If so, after a delay of one minute, the number of tickets left for that movie is updated and the number of tickets sold for that movie is recorded. And then, the model checks if the movie is sold out or not. If it is sold out, then all the customers waiting for that movie are forced to leave the system after recording the number of those customers. At the end, the customer is disposed of after releasing the counter. Going a few steps back, if the number of tickets for the movie is not enough for the customer, then the customer randomly selects a movie that has not been attempted yet. If there is a movie that has not been attempted, the customer tries to buy tickets without releasing the counter. And if there are no movies attempted, then the customer is disposed of after releasing the counter.

3-Counter Model:

In the 3-Counter Model, there is a personnel assigned to each movie, resulting in three different queues for movies. The general idea is still the same, however there are slight differences in the model. Obviously, there is a queue for each movie. After determining the choice of the customer, the model puts the customer in the corresponding queue. Another difference is in the case where the customer is supposed to choose another movie when there are not enough tickets for the movie initially chosen and yet the movie is not sold out. In this case the customer releases the counter after choosing another movie and gets in the queue of the movie chosen. Other than these differences, the model is the same as the 1-Counter model.

1) This section provides the input analysis of the sample data. Before modelling a simulation in Arena, we must determine the statistical distribution of entities arriving into our system. Using R Studio, we have applied Kolmogorov-Smirnov Test to fit the data into a distribution. The following script includes the required tests, whereby we test if the sample data is fit into the exponential distribution.

```r
library(readxl)
require(vcd)
require(MASS)
data <- read_excel("interarrival_times-Assn3-2020.xls",)
View(data)

day1 = as.vector(data$'Day1')
View(day1)

arrival_mean = mean(day1)
print(arrival_mean)

fit1 <- fitdistr(day1, "exponential")
ks.test(day1, "pexp", fit1$estimate)
```

```
        One-sample Kolmogorov-Smirnov test

data:  day1
D = 0.069555, p-value = 0.7186
alternative hypothesis: two-sided
```

The size of sample data provided is 100. The Kolmogorov-Smirnov for testing goodness of fit results in a test value of D=0.069555 where degree of freedom is over 35. In Table A.8 in reference book, the critical value is approximated as 0.136 by the given formula. The test yields a fail to reject conclusion, acceptance of null hypothesis, which suggests that the sample data fits into the desired distribution, which is the exponential distribution with a mean of sample mean, 1.744706 minutes.

2) The model is converted into a 3-movie case in the way expressed in section Model Logic & Assumptions above.

3) Running the `*BaseModel.doe*` for a single counter gives the results as:

Avg time before the movie is sold out

The Clone Wars: $56.2333 \pm 15.91 \Rightarrow 40.3233 < \mu < 72,1433$

Decalogue: $44.0667 \pm 17,22 \Rightarrow 26,8467 < \mu < 61.2867$

Vavien: $49.4000 \pm 16.96 \Rightarrow 32,44 < \mu < 66,36$

Avg number of people reneged

The Clone Wars: $0.3333 \pm 0.27 \Rightarrow 0.0633 < \mu < 0.6033$

Decalogue: $0.4000 \pm 0.45 \Rightarrow 0 < \mu < 0.85$

Vavien: $0.4333 \pm 0.44 \Rightarrow 0 < \mu < 0.8733$

Utilization of the seller is the instantaneous utilization of the counter:

Counter 1: $0.4928 \pm 0.03 \Rightarrow 0.4628 < x < 0.5228$

4) Running the `*AlternativeModel.doe*` for three counter gives the results as:

Avg time before the movie is sold out

The Clone Wars: $46.3000 \pm 16.03 \Rightarrow 30.27 < \mu < 62.33$

Decalogue: $48.0667 \pm 17.62 \Rightarrow 30.4467 < \mu < 65.6867$

Vavien: $51.6333 \pm 16.55 \Rightarrow 35.0833 < \mu < 68.1833$

Avg number of people reneged

The Clone Wars: $0.06666 \pm 0.09 \Rightarrow 0 < \mu < 0.15666$

Decalogue: $0.06666 \pm 0.09 \Rightarrow 0 < \mu < 0.15666$

Vavien: $0.1667 \pm 0.17 \Rightarrow 0 < \mu < 0.3367$

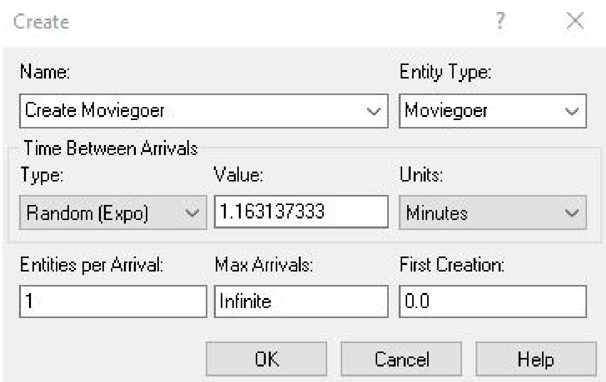
Utilization of the seller is the instantaneous utilization of the counters:

Counter 1: $0.1680 \pm 0.02 \Rightarrow 0.148 < x < 0.188$

Counter 2: $0.1684 \pm 0.02 \Rightarrow 0.1484 < x < 0.1884$

Counter 3: $0.1731 \pm 0.02 \Rightarrow 0.1531 < x < 0.1931$

5) Our initial assumption for the time between arrival times was exponentially distributed with 1.744706 minute average. Increasing its rate by 50% gives the new mean value as 1.163137333.



Running the `*BaseModel.doe*` for a **single counter** gives the results as:

Avg time before the movie is sold out

The Clone Wars: $30.2333 \pm 11.13 \Rightarrow 19.1033 < \mu < 41.3633$

Decalogue: $39.7667 \pm 13.25 \Rightarrow 26.5167 < \mu < 53.0167$

Vavien: $38.9 \pm 11.63 \Rightarrow 27.27 < \mu < 50.53$

Avg number of people reneged

The Clone Wars: $0.6667 \pm 0.47 \Rightarrow 0.1967 < \mu < 1.1367$

Decalogue: $1.800 \pm 1.58 \Rightarrow 0.22 < \mu < 3.38$

Vavien: $1.0333 \pm 0.82 \Rightarrow 0.2133 < \mu < 1.8533$

Utilization of the seller is the instantaneous utilization of the counter:

Counter 1: $0.5947 \pm 0.06 \Rightarrow 0.5347 < \rho < 0.6547$

Running the `*AlternativeModel.doe*` for **three counter** gives the results as:

Avg time before the movie is sold out

The Clone Wars: $30.8667 \pm 10.69 \Rightarrow 20.1767 < \mu < 41.5567$

Decalogue: $41.5 \pm 12.87 \Rightarrow 28.63 < \mu < 54.37$

Vavien: $30.1333 \pm 11.04 \Rightarrow 19.0933 < \mu < 41.1733$

Avg number of people reneged

The Clone Wars: $0.2333 \pm 0.19 \Rightarrow 0.0433 < \mu < 0.4233$

Decalogue: $0.3000 \pm 0.30 \Rightarrow 0 < \mu < 0.60$

Vavien: $0.1000 \pm 0.15 \Rightarrow 0 < \mu < 0.25$

Utilization of the seller is the instantaneous utilization of the counters:

Counter 1: $0.1964 \pm 0.03 \Rightarrow 0.1664 < \rho < 0.2264$

Counter 2: $0.1984 \pm 0.03 \Rightarrow 0.1684 < \rho < 0.2284$

Counter 3: $0.2151 \pm 0.04 \Rightarrow 0.1751 < \rho < 0.2551$

6) In this section, the simulation length is adjusted into 60 minutes with interarrival mean of 1.163137333 (as we did in **Task 5**), and both models are run for 30 repetitions.

Running the `*BaseModel.doe*` for a **single counter** gives the results as:

Avg time before the movie is sold out

The Clone Wars: $20.5333 \pm 9.64 \Rightarrow 10.8933 < \mu < 30.1733$

Decalogue: $10.0 \pm 7.68 \Rightarrow 2.32 < \mu < 17.68$

Vavien: $13.7667 \pm 8.71 \Rightarrow 5.0567 < \mu < 22.4767$

Avg number of people reneged

The Clone Wars: $0.5667 \pm 0.48 \Rightarrow 0.0867 < \mu < 1.0467$

Decalogue: $0.3333 \pm 0.41 \Rightarrow 0.00 < \mu < 0.7433$

Vavien: $0.2667 \pm 0.31 \Rightarrow 0.2133 < \mu < 0.5767$

Utilization of the seller is the instantaneous utilization of the counter:

Counter 1: $0.7977 \pm 0.03 \Rightarrow 0.7677 < \rho < 0.8277$

Running the `*AlternativeModel.doe*` for **three counter** gives the results as:

Avg time before the movie is sold out

The Clone Wars: $19.4667 \pm 9.17 \Rightarrow 10.2967 < \mu < 28.6367$

Decalogue: $10.0 \pm 7.70 \Rightarrow 2.3 < \mu < 17.70$

Vavien: $14.7333 \pm 8.60 \Rightarrow 6.1333 < \mu < 23.3333$

Avg number of people reneged

The Clone Wars: $0.1333 \pm 0.13 \Rightarrow 0.0033 < \mu < 0.2633$

Decalogue: $0.06666667 \pm 0.09 \Rightarrow 0 < \mu < 0.1567$

Vavien: $0.00 \pm 0.00 \Rightarrow 0.00 < \mu < 0.00$

Utilization of the seller is the instantaneous utilization of the counters:

Counter 1: $0.2798 \pm 0.02 \Rightarrow 0.2598 < \rho < 0.2998$

Counter 2: $0.2729 \pm 0.02 \Rightarrow 0.2529 < \rho < 0.2929$

Counter 3: $0.2873 \pm 0.02 \Rightarrow 0.2673 < \rho < 0.3073$

Let us continue with further analysis on how the ticket purchases are affected by the change in simulation length, keeping the arrival rate constant (**the rate used in Task 5)**.

Let us first analyze the 1-counter case by the following reports:

1. Simulation Length = 120

Count	Average	Half Width	Minimum Average	Maximum Average
CustomersReturned	29.5667	6,74	0.00	61.0000
Mov1SoldoutTime	30.2333	11,13	0.00	76.0000
Mov2SoldoutTime	39.7667	13,25	0.00	108.00
Mov3SoldoutTime	38.9000	11,63	0.00	80.0000
numTicketsSold1	48.5333	0,19	48.0000	49.0000
numTicketsSold2	48.6000	0,19	48.0000	49.0000
numTicketsSold3	48.6333	0,18	48.0000	49.0000
Record Reneging Movie1 Customers	0.6667	0,47	0.00	6.0000
Record Reneging Movie2 Customers	1.8000	1,58	0.00	18.0000
Record Reneging Movie3 Customers	1.0333	0,82	0.00	11.0000

2. Simulation Length = 60

Count	Average	Half Width	Minimum Average	Maximum Average
CustomersReturned	1.8333	0,86	0.00	9.0000
Mov1SoldoutTime	20.5333	9,64	0.00	59.0000
Mov2SoldoutTime	10.0000	7,68	0.00	59.0000
Mov3SoldoutTime	13.7667	8,71	0.00	59.0000
numTicketsSold1	46.0333	1,89	29.0000	49.0000
numTicketsSold2	44.6000	1,89	34.0000	49.0000
numTicketsSold3	44.7000	2,09	26.0000	49.0000
Record Reneging Movie1 Customers	0.5667	0,48	0.00	6.0000
Record Reneging Movie2 Customers	0.3333	0,41	0.00	5.0000
Record Reneging Movie3 Customers	0.2667	0,31	0.00	4.0000

The counters numTicketsSoldX is used to keep track of final values of sold tickets per repetitions. We observe that the number of tickets sold decreases in average for all movies.

Let’s turn our attention into 3-counter case by the following reports:

1. Simulation Length = 120

Count	Average	Half Width	Minimum Average	Maximum Average
CustomersReturned	31.6000	6,55	0.00	62.0000
Mov1SoldoutTime	30.8667	10,69	0.00	76.0000
Mov2SoldoutTime	41.5000	12,87	0.00	104.00
Mov3SoldoutTime	30.1333	11,04	0.00	77.0000
numTicketsSold1	48.5667	0,19	48.0000	49.0000
numTicketsSold2	48.6333	0,18	48.0000	49.0000
numTicketsSold3	48.5333	0,19	48.0000	49.0000
Record Reneging Movie1 Customers	0.2333	0,19	0.00	2.0000
Record Reneging Movie2 Customers	0.3000	0,30	0.00	4.0000
Record Reneging Movie3 Customers	0.1000	0,15	0.00	2.0000

2. Simulation Length = 60

Count	Average	Half Width	Minimum Average	Maximum Average
CustomersReturned	2.5333	1,09	0.00	10.0000
Mov1SoldoutTime	19.4667	9,17	0.00	57.0000
Mov2SoldoutTime	10.0000	7,70	0.00	59.0000
Mov3SoldoutTime	14.7333	8,60	0.00	59.0000
numTicketsSold1	46.5333	1,56	33.0000	49.0000
numTicketsSold2	45.6667	1,67	34.0000	49.0000
numTicketsSold3	45.9000	1,76	28.0000	49.0000
Record Reneging Movie1 Customers	0.1333	0,13	0.00	1.0000
Record Reneging Movie2 Customers	0.06666667	0,09	0.00	1.0000
Record Reneging Movie3 Customers	0.00	0,00	0.00	0.00

Just as we observed in 1-counter case, the average number of tickets sold for all movies decrease in 3-counter case as well.

In conclusion, decreasing the number of minutes counter(s) work might be beneficial if the hourly wage of counter(s) can compensate the decrease in the sale profits. Otherwise, keeping the counters open for 120 minutes will be more beneficial, since the average number of tickets sold increases. The problem here depends on the hourly employee expenses for the cinema.

Output Analysis and Comments:

A general output analysis can be conducted as follows. Keeping the interarrival rate constant (in terms of given sample), increasing the number of counters result in a decrease in average counter(s) utilization and average number of people reneged from the queue. The empirical observations are actually sensible, since an increase in the number of counters will yield a lower waiting time in queue per customers, and a lower average number of customers in queue.

Let us now consider the increase in arrival rate. We observe that the average time before a movie is sold out decreases, whereas the average number of reneged people and counter(s) utilization increases, keeping the simulation length constant. This empirical results are expected as well. Since we are increasing the interarrival rate, the number of people waiting in queue(s) and average waiting time in queue(s) will increase. The movies will also be soldout much quicker than previous, since the arrival rate increases.

The output analysis is correlated with our theoretical expectations, which is stated by comparing the empirical results into each other to see how the output is affected as the model parameters vary.