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 λ_1 = 5 per minute (i.e., mean inter arrival rate μ_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 μ_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 λ_1 = 50 to simulate a busier airport.]

Answer 13.2

I created the simulation using ARENA software. (ARENA file - week6_13.2.doe)

1. Use the Basic Process -> Create block to simulate passenger arriving to airport security system.

2. Use Process block for ID/boarding pass check queue. Set the inter arrival rate of 0.2minutes.

3. Use the resource spreadsheet to add resources.

Resource - Basic Process

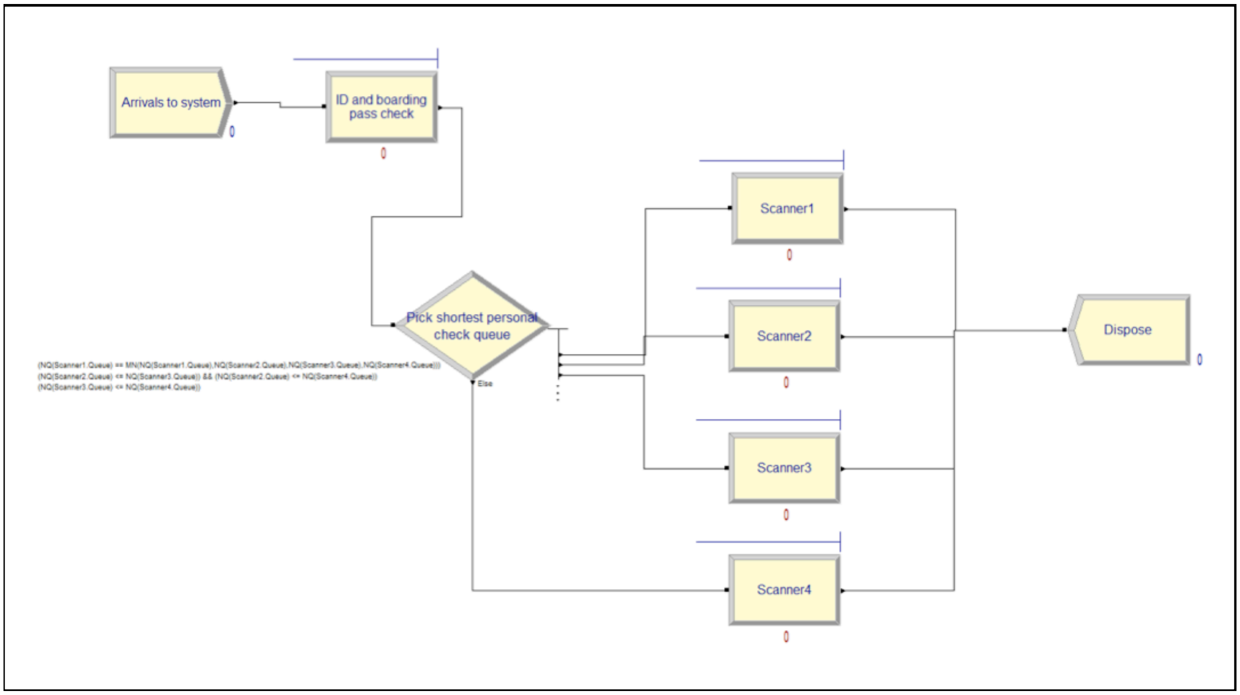
	Name	Type	Capacity	Busy / Hour	Idle / Hour	Per Use	StateSet Name	Failures	Report Statistics
1	Scanner 1	Fixed Capacity	1	0.0	0.0	0.0		0 rows	☑
2	Resource 1	Fixed Capacity	4	0.0	0.0	0.0		0 rows	☑
3	Scanner 2	Fixed Capacity	1	0.0	0.0	0.0		0 rows	☑
4	Scanner 3	Fixed Capacity	1	0.0	0.0	0.0		0 rows	☑
5	Scanner 4	Fixed Capacity	1	0.0	0.0	0.0		0 rows	☑

Double-click here to add a new row.

4. Choose the decision block to assign passenger to the shortest of personal check queues.

5. Once the passenger chooses a Scanner, action for scanner is Seize Delay Release, as only one passenger can use scanner at a time.

6. As last step, passenger exits the system. Use Dispose block for this.



Airport Security System Simulation using Arena

Run the Simulation

After creating Process flow, configure simulation parameters.

Number of replications - 100

Replication length - 1

Hours per day - 24

Set the batch run (no animation), to get output faster.

Review report using Reports -> Category Overview

5:50:04PM **Category Overview** June 26, 2018

Values Across All Replications

Unnamed Project

Replications: 100 Time Units: Hours

Entity

Time

VA Time	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
Entity 1	0.02491463	0.00	0.02313381	0.02661764	0.00839401	0.1905

NVA Time	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
Entity 1	0.00	0.00	0.00	0.00	0.00	0.00

Wait Time	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
Entity 1	0.03549699	0.00	0.01572224	0.08869949	0.00	0.1657

Transfer Time	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
Entity 1	0.00	0.00	0.00	0.00	0.00	0.00

Other Time	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
Entity 1	0.00	0.00	0.00	0.00	0.00	0.00

Total Time	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
Entity 1	0.06041163	0.00	0.03892563	0.1147	0.00883921	0.2818

Other

Conclusion: Based on the 100 replications above, average wait time is less than 15 minutes. To keep the average wait time below 15 minutes, 4 boarding pass checkers and 4 personal check queues are required (as chosen in setup of simulation).

Question 14.1

The breast cancer data set breast-cancer-wisconsin.data.txt from http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/ (description at http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29) has missing values.

1. Use the mean/mode imputation method to impute values for the missing data. 2. Use regression to impute values for the missing data.

Use regression with perturbation to impute values for the missing data.
4. (Optional) Compare the results and quality of classification models (e.g., SVM, KNN) build using

(1) the data sets from questions 1,2,3;
(2) the data that remains after data points with missing values are removed; and (3) the data set when a binary variable is introduced to indicate missing values.

Answer 14.1

Checking the data points, Bare Nuclei (V7), has missing data.

Row indexes with missing values are:

```
24,  41, 140, 146, 159, 165, 236, 250, 276, 293, 295, 298, 316, 322, 412, 618
```

1. Using the mean function and rounding it off, because the attribute information is integer. I got the missing value as 3.

2. Using step wise linear regression, I used factors V2, V3, V4, V5 and V9 to compute the missing values of V7.

```
Following are the values for missing data calculated based on regression

4,  4,  2,  3,  2,  3,  3,  2,  3,  6,  2,  3,  4,  2,  2,  2
```

3. To impute missing values using regression with perturbation, I used random normal distribution with standard deviation.

> Following are the values for missing data calculated based on regression
>
> 3, 3, 1, 1, 4, 3, 1, 4, 4, 4, 3, 4, 6, 3, 4, 1

4. Using classification model knn, I get the following results for different datasets.

knn classification model	Using mean for imputation	Using regression for imputation	Using regression with perturbation for imputation	Removing rows with missing values
k=1	0.957	0.962	**0.968**	0.951
k=2	0.957	0.962	**0.968**	0.951
k=3	0.919	0.914	0.928	0.927
k=4	0.919	0.914	0.928	0.927
k=5	0.919	0.914	0.928	0.917

It doesn't look like there is significant difference in accuracy of the model using the above 4 ways. Using regression with perturbation mechanism to impute missing values, seems to be slightly better with this dataset.

Question 15.1

Describe a situation or problem from your job, everyday life, current events, etc., for which optimization would be appropriate. What data would you need?

Answer 15.1

In my organization we have been trying to build agile model for release of applications. There are more than 80 different applications which need to be released within a span of maximum 7 days. Goal is to minimize the number of days to complete the release cycle. Some of the applications can be grouped and released together. Some of them have sequential dependency on others, meaning Application A should be released before Application B can be released. An application is ready to be released only when it has successfully completed User acceptance testing. Data needed would be:

- List of applications
- Dependencies
- Acceptance testing completed (Y/N)
- Grouping of applications