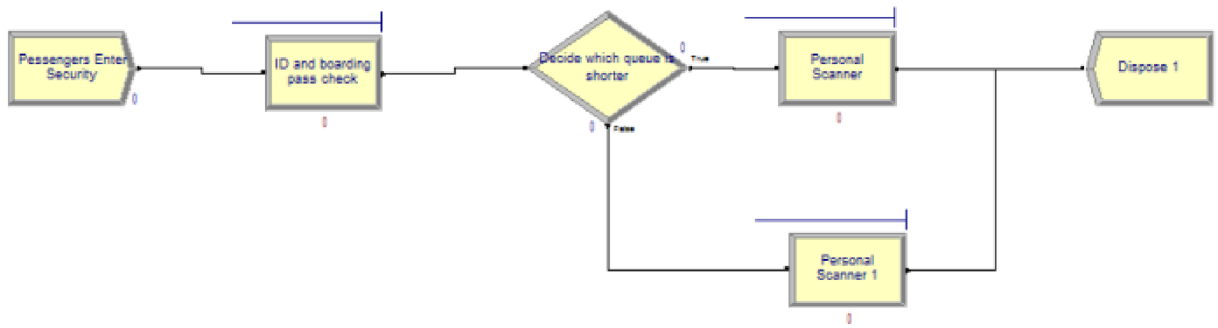
HW 6

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 $\lambda_1 = 5$ per minute (i.e., mean interarrival rate $\mu_1 = 0.2$ minutes) to the ID/boarding-pass check queue, where there are several servers who each have exponential service time with mean rate $\mu_2 = 0.75$ 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 $\lambda_1 = 50$ to simulate a busier airport.]



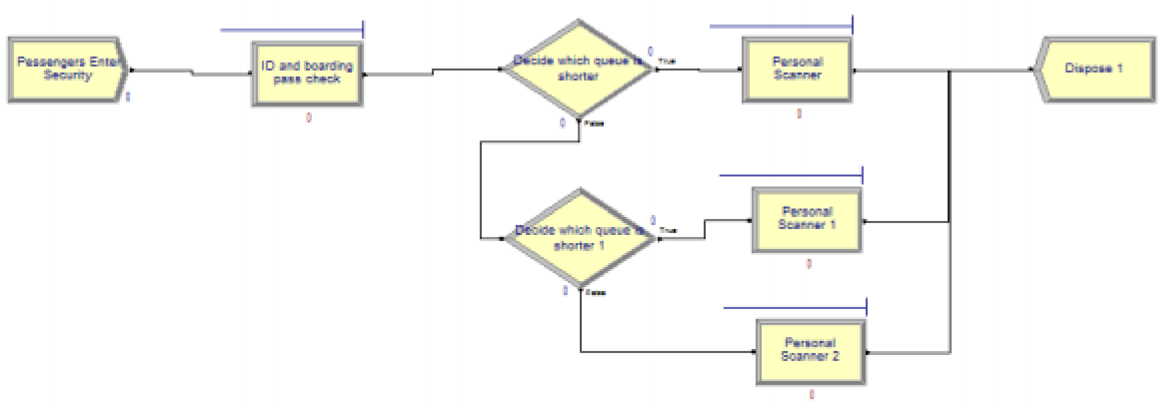
Wait Time

	Average
Entity 1	0.4396

Waiting Time

	Average
ID and boarding pass check.Queue	0.1300
Personal Scanner 1.Queue	0.3108
Personal Scanner.Queue	0.3114

Adding 3 servers at ID & boarding pass check, 2 personal scanner lines with one server each will give a Wait time of 26 minutes which is above 15 min threshold.



Wait Time	Average
Entity 1	0.3156

Waiting Time	Average
ID and boarding pass check.Queue	0.2636
Personal Scanner 1.Queue	0.05120456
Personal Scanner 2.Queue	0.04611909
Personal Scanner.Queue	0.05924787

Wait time = 18.936 with one extra scanner

Wait Time	Average
Entity 1	0.2756

Waiting Time	Average
ID and boarding pass check.Queue	0.04396864
Personal Scanner 1.Queue	0.2308
Personal Scanner 2.Queue	0.2287
Personal Scanner.Queue	0.2385

Wait time = 16.536 with two extra scanners

Wait Time	Average
Entity 1	0.1088

Waiting Time	Average
ID and boarding pass check.Queue	0.07457095
Personal Scanner 1.Queue	0.03570406
Personal Scanner 2.Queue	0.02793915
Personal Scanner 3.Queue	0.02387695
Personal Scanner.Queue	0.04553027

Wait time = 6.528 with three extra scanners

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.

```
#Read the data
data <- read.table("14.1breast-cancer-wisconsin.dataSummer2018.txt", stringsAsFactors = FALSE,
header = FALSE, sep=",")

# Try to find the missing value
for (i in 2:11){
  print(paste0("V",i))
  print(table(data[,i]))
}
```

The missing is within vector v7. I verified this by going to the table and filtering by ?.

```
[1] "V7"

  ?   1  10   2   3   4   5   6   7   8   9
 16 402 132  30  28  19  30   4   8  21   9
```

show the missing data
data[which(data$V7 =="?"),]

```
         V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11
24  1057013  8  4  5  1  2  ?  7  3   1   4
41  1096800  6  6  6  9  6  ?  7  8   1   2
140 1183246  1  1  1  1  1  ?  2  1   1   2
146 1184840  1  1  3  1  2  ?  2  1   1   2
159 1193683  1  1  2  1  3  ?  1  1   1   2
165 1197510  5  1  1  1  2  ?  3  1   1   2
236 1241232  3  1  4  1  2  ?  3  1   1   2
250  169356  3  1  1  1  2  ?  3  1   1   2
276  432809  3  1  3  1  2  ?  2  1   1   2
293  563649  8  8  8  1  2  ?  6 10   1   4
295  606140  1  1  1  1  2  ?  2  1   1   2
298   61634  5  4  3  1  2  ?  2  3   1   2
316  704168  4  6  5  6  7  ?  4  9   1   2
322  733639  3  1  1  1  2  ?  3  1   1   2
412 1238464  1  1  1  1  1  ?  2  1   1   2
618 1057067  1  1  1  1  1  ?  1  1   1   2
```

#Which rows in vector has the ?
missing<-which(data$V7 =="?", arr.ind = TRUE)

# The mode of v7 is 1. Use 1 to impute missing data.
mode_v7 <-as.numeric(getmode(data[-missing,"V7"]))

2. Use regression to impute values for the missing data.

# Not to include the response variable in regression imputation
datam <- data[-missing, 2:10]
datam$V7 <- as.integer(datam$V7)

Linear regression Imputation
model <- lm(V7~V2+V3+V4+V5+V6+V7+V8+V9+V10, datam)
summary(model)

```
Call:
lm(formula = V7 ~ V2 + V3 + V4 + V5 + V6 + V7 + V8 + V9 + V10,
    data = datam)

Residuals:
    Min      1Q  Median      3Q     Max
-9.7316 -0.9426 -0.3002  0.6725  8.6998

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.616652   0.194975  -3.163  0.00163 **
V2           0.230156   0.041691   5.521 4.83e-08 ***
V3          -0.067980   0.076170  -0.892  0.37246
V4           0.340442   0.073420   4.637 4.25e-06 ***
V5           0.339705   0.045919   7.398 4.13e-13 ***
V6           0.090392   0.062541   1.445  0.14883
V8           0.320577   0.059047   5.429 7.91e-08 ***
V9           0.007293   0.044486   0.164  0.86983
V10         -0.075230   0.059331  -1.268  0.20524
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.274 on 674 degrees of freedom
Multiple R-squared:  0.615,     Adjusted R-squared:  0.6104
F-statistic: 134.6 on 8 and 674 DF,  p-value: < 2.2e-16
```

Step(model)

```
Coefficients:
(Intercept)           V2           V4           V5           V8
    -0.5360       0.2262       0.3173       0.3323       0.3238
```

#v2, v4, v5, v8 are important variables to predict v7.
model2<- lm(V7~V2+V4+V5+V8, datam)
summary(model2)

```
Call:
lm(formula = V7 ~ V2 + V4 + V5 + V8, data = datam)

Residuals:
    Min      1Q  Median      3Q     Max 
-9.8115 -0.9531 -0.3111  0.6678  8.6889 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.53601    0.17514  -3.060   0.0023 ** 
V2           0.22617    0.04121   5.488 5.75e-08 ***
V4           0.31729    0.05086   6.239 7.76e-10 ***
V5           0.33227    0.04431   7.499 2.03e-13 ***
V8           0.32378    0.05606   5.775 1.17e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.274 on 678 degrees of freedom
Multiple R-squared:  0.6129,	Adjusted R-squared:  0.6107 
F-statistic: 268.4 on 4 and 678 DF,  p-value: < 2.2e-16
```

#All variables are important use model 2.
# predict the values to impute in V7
V7v <- predict(model2, newdata = data[missing,])
data_reg_imp <- data
data_reg_imp[missing,]$V7 <-V7v
data_reg_imp$V7 <- as.numeric(data_reg_imp$V7)

```
       24        41       140       146       159       165       236       250       276 
5.4585352 7.9816106 0.9872832 1.6218560 0.9807851 2.2157441 2.7152652 1.7634059 2.0741942 
      293       295       298       316       322       412       618 
6.0866099 0.9872832 2.5265324 5.2438347 1.7634059 0.9872832 0.6634986 
```

3. Use regression with perturbation to impute values for the missing data.
V7_h <- rnorm(length(missing), V7v, sd(V7v))
V7_h

```
 [1]  3.1253997  9.1739362  1.1192448 -0.6122775 -2.4170019  2.2766128  4.6494576  1.1139752  1.7010815
[10]  8.5017041  4.8511467  0.5388254  4.0268305  3.5838691  3.2906390 -2.0443688
```

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?

I can use optimization to allocate my time spent on physical fitness to maximize weight loss.

Variables:

Xi = # of hours spent on physical activity per day

Yi = #weight measurement in pounds each day

Constraints:

Total Xi per day should be less than or equal to 2. (this limit is set to avoid injuries)

Objective function:

Max sigma Xi per day to max Yi per day