Homework 1

DATA604 Simulation and Modeling

Daniel Dittenhafer

February 2, 2016

1.1

Name several entities, attributes, activities, events, and state variables for the following systems.

(a) A cafeteria

Entities

- Serving Line
- Food Servers
- Tables

Attributes

- Number of Food Servers
- Number of seats per table
- Rate of serving for Food Servers
- Time range for eating the meal

Activities

- Waiting in line
- Being served by a Food Server
- Waiting for a table to eat
- Eating at a table

Events

- Arrival of new person in serving line to be served
- Person leaving serving line
- Person waiting for seat at table to eat
- Person finishing eating and leaving table

State Variables

- Number of people eating at tables
- Number of people waiting in line to be served

(b) A grocery store

Entities

- Checkout lanes

Attributes

- Max number of items allowed in checkout lane
- Rate of checkout for cashier

Activities

- Customer shopping in the grocery store
- Customer checking out (paying for goods)

Events

- Arrival of customer at grocery store
- Arrival of customer at checkout lane
- Customer completing checkout
- Customer departing store without purchasing anything

State Variables

- Number of customers in grocery store
- Number of customers in checkout lane lines

(c) A laundromat

Entities

- Washing machines
- Drying machines

Attributes

- Washing machine run time
- Drying machine run time
- Ratio of washing machine to drying machine capacity

Activities

- Washing clothes
- Drying clothes
- Loading washing machine
- Transfering from washing to drying machine
- Unloading from drying machine

Events

- Washing maching cycle starts
- Washing machine cycle stops
- Dryer cycle starts
- Dryer cycle stops

State Variables

- Number of busy washing machines
- Number of busy dryers

(d) A fast-food restaurant

Entities

- Cashiers
- Back-cooks (i.e. burger flippers)
- Fryers

Attributes

- Burgers per burger flipper
- Orders of frys per Fryer
- Cashier busy or not

Activities

- Cooking a burger
- Making french frys
- Cashier taking order, accepting payment

Events

- Order in
- Order ready for pickup
- French fries done cooking

State Variables

- Number of orders pending
- Number of burgers being cooked
- Orders of french frys cooked/ready for serving
- Number of bugers being ready for serving

(e) A hospital emergency room

Entities

- Doctors
- Beds
- Patients
- Admitting staff

Attributes

- Patients per Doctor

Activities

- Patient admitted
- Doctor take care of patient
- Patient discharged

Events

- Patient arrives
- Patient admitted
- Doctor discharges patient

State Variables

- Beds empty
- Patients awaiting addmission
- Paitents awaiting discharge

(f) A taxicab company with 10 taxis

Entities

- Taxis
- Dispatcher
- Customers

Attributes

- Taxi has customer
- Taxi enroute to customer
- Customer waiting for taxi

Activities

- Enroute to customer
- Transporting customer

Events

- Picking up custommer
- Dropping off customer

State Variables

- Taxis with customers
- Customers waiting for available taxi

(g) An automobile assembly line

Entities

- Parts
- Assembly machines
- Workers

Attributes

- Parts inventory
- Assembly machine rate of production
- Worker rate of production

Activities

- Machine assembling car
- Worker assembling car
- Staging parts for use by Machine or Worker

Events

- Car assembly started
- Car assembly completed
- Parts depleted
- Car assembly by machine X completed
- Car assembly by worker Y completed

State Variables

- Cars on assembly line
- Parts inventory level
- Workers out sick/vacation
- Machines broken down

2.1

Consider the following continuously operating job shop. Interarrival times of jobs are distributed as follows:

Time Between Arrivals (hours)	Probability
0	0.23
1	0.37
2	0.28
3	0.12

Processing times for jobs are normally distributed, with mean 50 minutes, and standard deviation 8 minutes. Construct a simulation table and perform a simulation for 10 new customers. Assume that, when the simulation starts, there is one job being processed (scheduled to be completed in 25 minutes) and there is one job with a 50-minute processing time in the queue.

```
# Create a data frame of the pre-existing jobs
existingJobs <- data.frame(customer=c(-2, -1),
                           iaHrs=c(0,0),
                           iaMins=c(0,0),
                           arrivalMins=c(0,0),
                           svcTimeMins=c(25, 50),
                           timeSvcBegin=c(0, 25),
                           queueWaitMins=c(0,25),
                           timeSvcEnd=c(25, 75),
                           timeInSystem=c(25,75))
# Create a data frame of the new customers and their jobs
newJobs <- data.frame(customer=seq(1, 10),
                      iaHrs=c(0, sample(seq(0, 3),
                                        size=9,
                                        prob=c(.23, .37, .28, .12),
                                        replace=TRUE)),
                      iaMins=rep(NA, 10),
                      arrivalMins=rep(0, 10),
                      svcTimeMins=rnorm(10, mean=50, sd=8),
                      timeSvcBegin=rep(0, 10),
```

```
                      queueWaitMins=rep(0, 10),
                      timeSvcEnd=rep(0, 10),
                      timeInSystem=rep(0, 10))
# Convert from interarrival hours to minutes and
# determine overall arrival times
newJobs$iaMins <- newJobs$iaHrs * 60
newJobs$arrivalMins <- cumsum(newJobs$iaMins)
# Join the existing and new jobs into one table
simTable <- rbind(existingJobs, newJobs)
# Loop over the rows the compute the various activity and clock times
for(i in seq(3, nrow(simTable)))
{
  simTable[i,]$timeSvcBegin <- max(simTable[i,]$arrivalMins, simTable[i-1,]$timeSvcEnd)
  simTable[i,]$queueWaitMins <- simTable[i,]$timeSvcBegin - simTable[i,]$arrivalMins
  simTable[i,]$timeSvcEnd <- simTable[i,]$timeSvcBegin + simTable[i,]$svcTimeMins
  simTable[i,]$timeInSystem <- simTable[i,]$timeSvcEnd - simTable[i,]$arrivalMins
}
# Show the table
kable(simTable)
```

customer	iaHrs	iaMins	arrivalMins	svcTimeMins	timeSvcBegin	queueWaitMins	timeSvcEnd	timeInSystem
-2	0	0	0	25.00000	0.0000	0.00000	25.0000	25.00000
-1	0	0	0	50.00000	25.0000	25.00000	75.0000	75.00000
1	0	0	0	49.88872	75.0000	75.00000	124.8887	124.88872
2	3	180	180	44.90948	180.0000	0.00000	224.9095	44.90948
3	0	0	180	57.70883	224.9095	44.90948	282.6183	102.61831
4	1	60	240	49.44490	282.6183	42.61831	332.0632	92.06321
5	1	60	300	55.89174	332.0632	32.06321	387.9550	87.95496
6	2	120	420	46.70007	420.0000	0.00000	466.7001	46.70007
7	1	60	480	44.79027	480.0000	0.00000	524.7903	44.79027
8	1	60	540	64.67686	540.0000	0.00000	604.6769	64.67686
9	0	0	540	35.05879	604.6769	64.67686	639.7356	99.73565
10	2	120	660	50.96674	660.0000	0.00000	710.9667	50.96674

(a) What was the average time in the queue for the 10 new jobs? The average time in the queue for the 10 new jobs is computed below:

```
mean(simTable[seq(3, 12),]$queueWaitMins)
```

```
## [1] 25.92679
```

(b) What was the average processing time of the 10 new jobs? The average processing time is computed below:

```
mean(newJobs$svcTimeMins)
```

```
## [1] 50.00364
```

(c) What was the maximum time in the system for the 10 new jobs? The maximum time in the system for the 10 new jobs is computed below:

```
max(simTable[seq(3, 12),]$timeInSystem)
```

```
## [1] 124.8887
```

2.2

A baker is trying to figure out how many dozens of bagels to bake each day. The probability distribution of the number of bagel customers is as follows:

Customer/Day	8	10	12	14
Probability	0.35	0.30	0.25	0.10

Customers order 1,2,3 or 4 dozen bagels according to the following probability distribution:

Dozen Ordered/Customer	1	2	3	4
Probability	0.4	0.3	0.2	0.1

Bagels sell for $8.40 per dozen. They cost $5.80 per dozen to make. All bagels not sold at the end of the day are sold at half price to a local grocery store. Based on 5 days of simulation, how many dozen (to the nearest 5 dozen) bagels should be baked each day?

```r
# Function to define a simulation at a specified
# level of dozens of bagels produced.
bakersProfit <- function(bagelsMade)
{
  simDays <- 5
  revPerDoz <- 8.40
  costPerDoz <- 5.80
  simTable <- data.frame(day=seq(1, simDays),
                       customers=sample(c(8,10,12,14),
                                        size=simDays,
                                        prob=c(0.35, 0.30, 0.25, 0.10),
                                        replace=TRUE),
                       dozenOrdered=rep(NA, simDays),
                       revenue=rep(NA, simDays),
                       lostProfit=rep(NA, simDays),
                       salvage=rep(NA, simDays),
                       dailyCost=rep(NA, simDays),
                       dailyProfit=rep(NA, simDays))

  for(i in seq(1, nrow(simTable)))
  {
    bagelsOrdered <- sample(c(1,2,3,4),
                           size=simTable[i,]$customers,
                           prob=c(0.4, 0.3, 0.2, 0.1),
                           replace=TRUE)

    simTable[i,]$dozenOrdered <- sum(bagelsOrdered)

    simTable[i,]$revenue <- min(simTable[i,]$dozenOrdered, bagelsMade) * revPerDoz
    simTable[i,]$lostProfit <- max(simTable[i,]$dozenOrdered - bagelsMade, 0) * (revPerDoz - costPerDoz)
    simTable[i,]$salvage <- max(bagelsMade - simTable[i,]$dozenOrdered, 0) * (revPerDoz / 2)
    simTable[i,]$dailyCost <- bagelsMade * costPerDoz
    simTable[i,]$dailyProfit <- simTable[i,]$revenue + simTable[i,]$salvage - simTable[i,]$dailyCost
  }

  return(simTable)
}
# Loop over a range of dozens of bagels (0, 5, 10, etc)
dozens <- seq(0, 30, by=5)
profitTable <- data.frame(dozPerDay=c(), fiveDayProfit=c())
```

```
for(d in dozens)
{
  # Run the simulation for the given level of production
  simTable <- bakersProfit(d)
  profitTable <- rbind(profitTable, cbind(dozPerDay=d, fiveDayProfit=sum(simTable$dailyProfit)))
  #print(paste(d, " dozen/day: 5 day profit is ", profitTable[profitTable$dozPerDay==d,]$fiveDayProfit, ".", s
}
```

The following table shows the profit associated with various levels of production:

dozPerDay	fiveDayProfit
0	0.0
5	65.0
10	130.0
15	186.6
20	251.6
25	199.0
30	222.0

The following table shows the details of the simulation for the maximum profit shown above (20 dozen bagels/day):

day	customers	dozenOrdered	revenue	lostProfit	salvage	dailyCost	dailyProfit
1	10	19	159.6	0.0	4.2	116	47.8
2	12	28	168.0	20.8	0.0	116	52.0
3	10	15	126.0	0.0	21.0	116	31.0
4	12	26	168.0	15.6	0.0	116	52.0
5	8	19	159.6	0.0	4.2	116	47.8

2.4

Smalltown Taxi operates one vehicle during the 9:00 A.M. to 5:00 P.M. period. Currently, consideration is being given to the addition of a second vehicle to the fleet. The demand for taxis follows the distribution shown:

Time Between Calls (minutes)	15	20	25	30	35
Probability	0.14	0.22	0.43	0.17	0.04

The distribution of time to complete a service is as follows:

Service Time (minutes)	5	15	25	35	45
Probability	0.12	0.35	0.43	0.06	0.04

Simulate 5 individual days of operation of the current system and of the system with an additional taxicab. Compare the two systems with respect to the waiting times of the customers and any other measures that might shed light on the situation.

```
singleTaxiDailyCalls <- function(callsPerDay, maxDailyMinutes)
{
  # Create a data frame of the new customers and their jobs
  simTable <- data.frame(customer=seq(1, callsPerDay),
                         iaMins=sample(seq(15, 35, by=5),
                                       size=callsPerDay,
                                       prob=c(0.14, 0.22, 0.43, 0.17, 0.04),
                                       replace=TRUE),
                         arrivalMins=rep(0, callsPerDay),
```

```r
                         svcTimeMins=sample(seq(5, 45, by=10),
                                            size=callsPerDay,
                                            prob=c(0.12, 0.35, 0.43, 0.06, 0.04),
                                            replace=TRUE),
                         timeSvcBegin=rep(0, callsPerDay),
                         queueWaitMins=rep(0, callsPerDay),
                         timeSvcEnd=rep(0, callsPerDay),
                         timeInSystem=rep(0, callsPerDay))

  # Determine overall arrival times
  simTable$arrivalMins <- cumsum(simTable$iaMins)
  # Loop over the rows the compute the various activity and clock times
  for(i in seq(1, nrow(simTable)))
  {
    if(i == 1)
    {
      simTable[i,]$timeSvcBegin <- simTable[i,]$arrivalMins

    }
    else
    {
      simTable[i,]$timeSvcBegin <- max(simTable[i,]$arrivalMins, simTable[i-1,]$timeSvcEnd)
    }
    simTable[i,]$queueWaitMins <- simTable[i,]$timeSvcBegin - simTable[i,]$arrivalMins
    simTable[i,]$timeSvcEnd <- simTable[i,]$timeSvcBegin + simTable[i,]$svcTimeMins
    simTable[i,]$timeInSystem <- simTable[i,]$timeSvcEnd - simTable[i,]$arrivalMins
  }

  return(simTable)
}

oneDayOneTaxi <- singleTaxiDailyCalls(32)

# Show the table
kable(oneDayOneTaxi)
```

customer	iaMins	arrivalMins	svcTimeMins	timeSvcBegin	queueWaitMins	timeSvcEnd	timeInSystem
1	20	20	5	20	0	25	5
2	25	45	25	45	0	70	25
3	25	70	25	70	0	95	25
4	20	90	35	95	5	130	40
5	20	110	5	130	20	135	25
6	30	140	15	140	0	155	15
7	30	170	15	170	0	185	15
8	20	190	5	190	0	195	5
9	25	215	25	215	0	240	25
10	20	235	5	240	5	245	10
11	20	255	15	255	0	270	15
12	25	280	15	280	0	295	15
13	20	300	25	300	0	325	25
14	15	315	5	325	10	330	15
15	15	330	25	330	0	355	25
16	20	350	15	355	5	370	20
17	25	375	25	375	0	400	25
18	15	390	35	400	10	435	45
19	25	415	25	435	20	460	45
20	15	430	25	460	30	485	55
21	25	455	25	485	30	510	55
22	30	485	25	510	25	535	50

customer	iaMins	arrivalMins	svcTimeMins	timeSvcBegin	queueWaitMins	timeSvcEnd	timeInSystem
23	25	510	15	535	25	550	40
24	20	530	25	550	20	575	45
25	25	555	5	575	20	580	25
26	25	580	15	580	0	595	15
27	25	605	15	605	0	620	15
28	20	625	15	625	0	640	15
29	15	640	15	640	0	655	15
30	15	655	25	655	0	680	25
31	25	680	25	680	0	705	25
32	30	710	15	710	0	725	15