Homework 1

Max Wagner

February 14, 2015

1.1

a. system: cafe, entities: customers, attributes: checking-account balance, activities: buying lunch, events: arrival/departure/eating, state variables: number of cashiers/number of customers

b. system: grocery store, entities: customers/cashiers, attributes: checking-account balance, activities: buying groceries, events: checking out, state variables: number of cashiers/number of customers

c. system: laundromat, entities: machines/customers, attributes: cost/money, activities: doing laundry, events: start/stop cycle, state variables: number of machines/number of customers/machine status

d. system: fast food restaurant, entities: customers/cashiers, attributes: money, activities: buying fast-food, eating, events: checking out, state variables: number of cashiers/number of customers/number of empty tables

e. system: ER, entities: doctors/patients, attributes: injuries, activities: healing injuries, events: taking medicine, state variables: number of doctors/number of patients/empty rooms

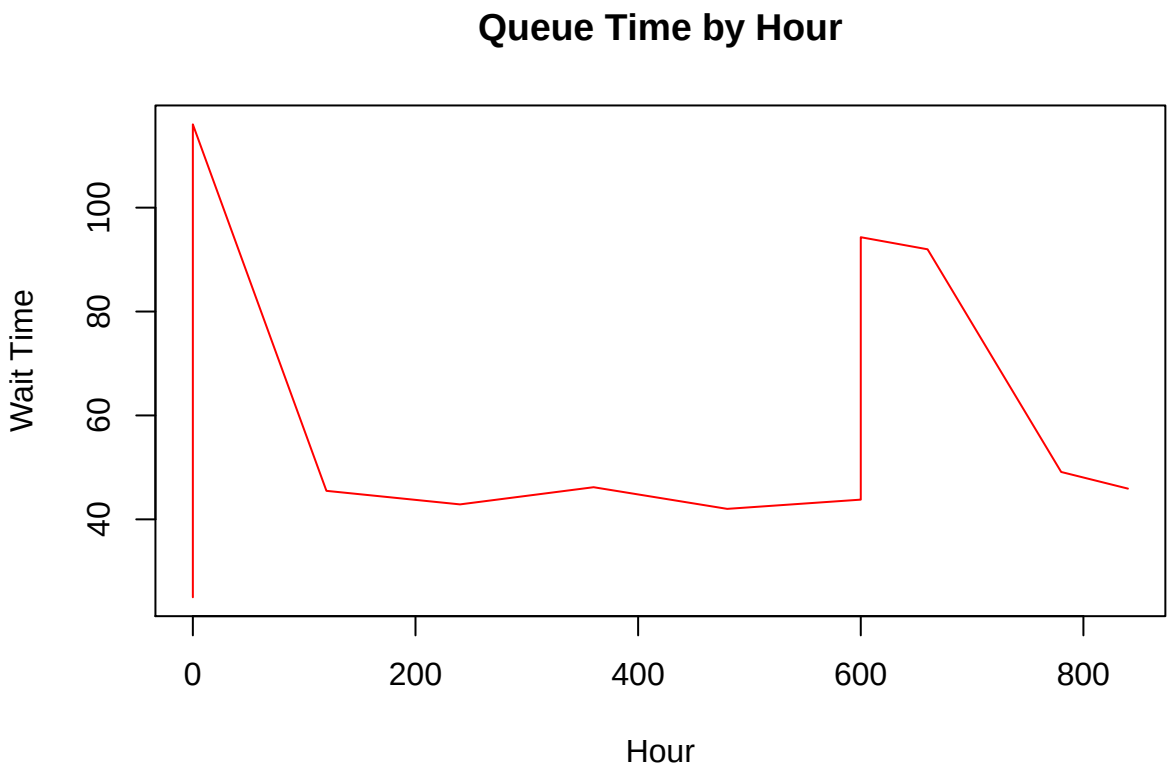
f. system: taxi company, entities: customers/taxis/drivers, attributes: origin/destination, activities: riding in a taxi, events: arriving/leaving, state variables: number of fairs in an hour

g. system: auto assembly line, entities: machines, attributes: speed/accuracy/precision, activities: making a door, events: welding/bolting, state variables: number of doors made

2.1 The code below sets up the sheet for the problem.

```
########################Initial Parameters#######################
set.seed(1234)
n=12  #should be 12 by the problem...
#################################################################

##################Set Up Interarrival Times######################
ia=rep(0,n) #initialize array for interarrival time (pmf)
s=c(rep(0,230),rep(60,370),rep(120,280),rep(180,120)) #pmf
ia[1]=0
ia[2]=0
ia[3:n]=sample(s,n-3+1,replace=T)
#################################################################

#######################Calculate Arrival Time####################
ac=rep(0,n) #initialize array for arrival time (arrival clock)
ac[1:2]=0
for (i in 3:n){ac[i]=ac[i-1]+ia[i]}
#################################################################

#######################Calculate Service Time####################
```

```
st=rnorm(n,50,8) #normal distrribution
sb=rep(0,n)  #time service begins
se=rep(0,n)  #time service ends
sb[1]=0
sb[2]=25
se[1]=25
se[2]=75
for (i in 3:n){sb[i]=max(sb[i-1]+st[i-1], ac[i])}
se[3:n]=sb[3:n]+st[3:n]
#########################################################################

#######################Calculate OtherTimes#####################
wt=se-ac  #wait time
it=rep(0,n)
it[1]=0
for (i in 2:n){it[i]=max(0,ac[i]-se[i-1])}
a=seq(1:n)
#########################################################################


########################Make it Pretty#############################
mydata=data.frame(cbind(a,ia,ac,st,sb,wt,se,it))
plot(wt~ac, main="Queue Time by Hour", col="red", xlab="Hour",
ylab="Wait Time", type="l")
```



```
write.csv(mydata, "output.csv")
#########################################################################
```

a. Average time in the queue:

```
mean(wt)
```

```
## [1] 59.80794
```

b. Average processing time:

```
mean(se-sb)
```

```
## [1] 45.3498
```

c. Max time in the system:

```
max(wt)
```

```
## [1] 116.029
```

2.2 A little prework below lets us know that we should bake at least 20 dozen a day.

```
nc <- .35 * 8 + .30 * 10 + .25 * 12 + .10 * 14 # number of customers
do <- .4 * 1 + .3 * 2 + .2 * 3 + .1 * 4 # dozens ordered
tot <- nc * do;tot
```

```
## [1] 20.4
```

profit = rev from sales - cost of bagels + rev from grocery sales - cost of bagels

profit = 8.4S - 5.8Q + 4.2(Q-S) - 5.8(Q-S)

```
set.seed(1234)
n=5
cust=rep(0,n)
s=c(rep(8,35),rep(10,30),rep(12,25),rep(14,10)) #customers
cust[1:n]=sample(s,n,replace=T)

doz=rep(0,n)
s=c(rep(1,4),rep(2,3),rep(3,2),rep(4,1)) #dozens
doz[1:n]=sample(s,n,replace=T)

q <- cust*doz #total dozens
profit <- 8.4-5.8
profitG <- 8.4/2 - 5.8
```

	cust	doz	m	s		
1	8	3	20	24	52	
2	10	2	20	20	52	
3	10	4	20	40	52	
4	10	3	20	30	52	
5	12	1	20	12	18.4	
						226.4
1	8	3	25	24	60.8	
2	10	2	25	20	44	
3	10	4	25	40	65	
4	10	3	25	30	65	
5	12	1	25	12	10.4	
						245.2
1	8	3	30	24	52.8	
2	10	2	30	20	36	
3	10	4	30	40	78	
4	10	3	30	30	78	
5	12	1	30	12	2.4	
						247.2
1	8	3	35	24	44.8	
2	10	2	35	20	28	
3	10	4	35	40	91	
4	10	3	35	30	70	
5	12	1	35	12	-5.6	
						228.2
1	8	3	40	24	36.8	
2	10	2	40	20	20	
3	10	4	40	40	104	
4	10	3	40	30	62	
5	12	1	40	12	-13.6	
						209.2
1	8	3	45	24	28.8	
2	10	2	45	20	12	
3	10	4	45	40	96	
4	10	3	45	30	54	
5	12	1	45	12	-21.6	
						169.2

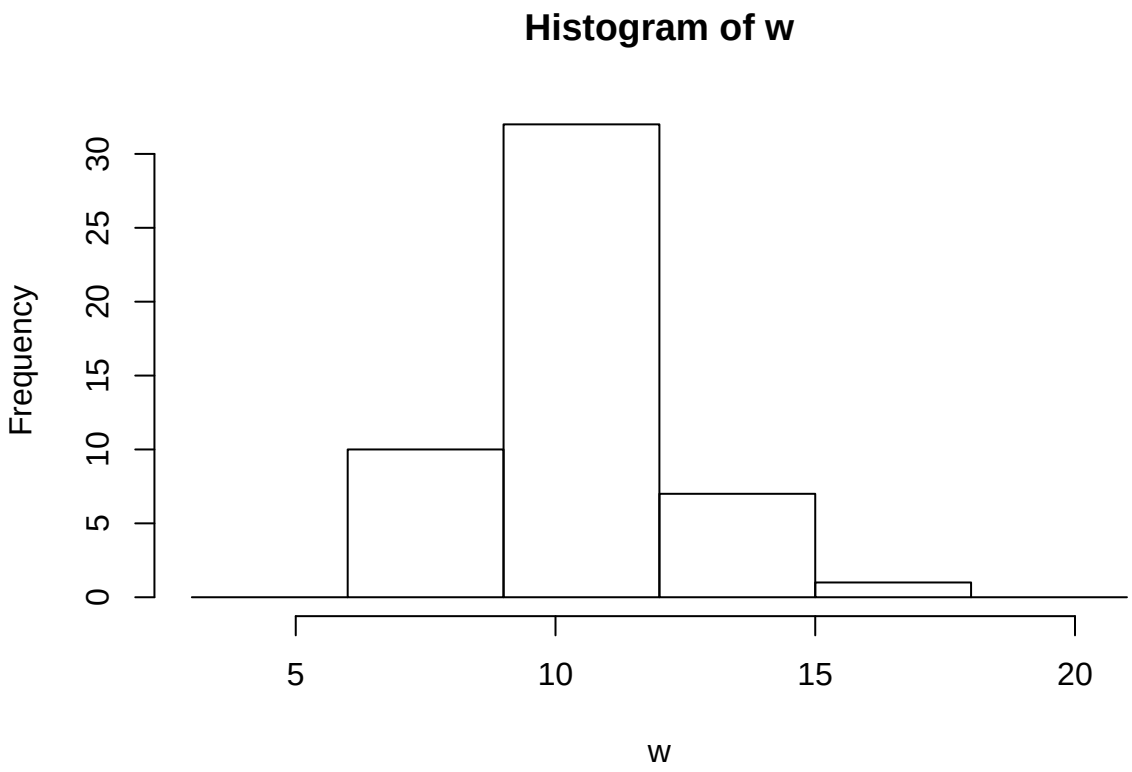
It's hard to say with only 5 days of simulation, but from the above excel sheet we can see that making 30-35 bagels seems to be optimal.

2.4 Some of the excel for day 1 of 5 with 1 taxi. All 5 days in total lead to an average wait time of 9.95 minutes per customer, and a average idle time of 5.36 minutes between customers. I am unsure how to do the 2 cab simulation.

day	call	intertime	call time	wait time	service time	start service	end service	idle time
1	1	25	9:25:00 AM	0.00	25	9:25:00 AM	9:50:00 AM	25.00
	2	20	9:45:00 AM	5.00	15	9:50:00 AM	10:05:00 AM	0.00
	3	25	10:10:00 AM	0.00	15	10:10:00 AM	10:25:00 AM	5.00
	4	20	10:30:00 AM	0.00	5	10:30:00 AM	10:35:00 AM	5.00
	5	25	10:55:00 AM	0.00	35	10:55:00 AM	11:30:00 AM	20.00
	6	35	11:30:00 AM	0.00	15	11:30:00 AM	11:45:00 AM	0.00
	7	25	11:55:00 AM	0.00	35	11:55:00 AM	12:30:00 PM	10.00
	8	20	12:15:00 PM	15.00	15	12:30:00 PM	12:45:00 PM	0.00
	9	20	12:35:00 PM	10.00	25	12:45:00 PM	1:10:00 PM	0.00
	10	25	1:00:00 PM	10.00	45	1:10:00 PM	1:55:00 PM	0.00
	11	35	1:35:00 PM	20.00	15	1:55:00 PM	2:10:00 PM	0.00
	12	25	2:00:00 PM	10.00	15	2:10:00 PM	2:25:00 PM	0.00
	13	25	2:25:00 PM	0.00	5	2:25:00 PM	2:30:00 PM	0.00
	14	25	2:50:00 PM	0.00	25	2:50:00 PM	3:15:00 PM	20.00
	15	35	3:25:00 PM	0.00	35	3:25:00 PM	4:00:00 PM	10.00
	16	30	3:55:00 PM	5.00	5	4:00:00 PM	4:05:00 PM	0.00
	17	25	4:20:00 PM	0.00	15	4:20:00 PM	4:35:00 PM	15.00
	18	25	4:45:00 PM	0.00	25	4:45:00 PM	5:10:00 PM	10.00

2.5 The equation is summarized below:

```
x <- rnorm(50, 100, sqrt(100))
y <- rnorm(50, 300, sqrt(225))
z <- rnorm(50, 40, sqrt(64))
w <- (x + y) / z
hist(w, breaks = seq(3,21,by=3))
```



2.7 There were 7 lost orders according to the below spreadsheet, which comes out to 1.4 orders lost per week.

```
data <- cbind(1:25, round(rnorm(25,5,1.5)))
write.csv(data, "output.csv")
```

day	inven	demand	order size	lead time	add to inven	total lost
1	18	6	0	0		0
2	12	4	0	3		0
3	8	7	12	2		0
4	1	7	0	2		1
5	13	3	0	0	12	1
6	10	4	0	5		1
7	6	7	14	4		2
8	6	2	0	4		2
9	4	6	0	5		3
10	4	5	0	4		4
11	18	10	0	5	14	4
12	8	1	12	4		4
13	7	4	0	0		4
14	3	6	0	0		5
15	3	8	0	0		6
16	15	6	0	4	12	6
17	9	3	11	1		6
18	17	8	0	5	11	6
19	9	2	11	2		6
20	7	3	0	0		6
21	15	5	0	1	11	6
22	10	8	0	0		6
23	2	6	18	1		7
24	20	6	0	2	18	7
25	14	8	0	4		7

2.8