Homework 1

Max Wagner

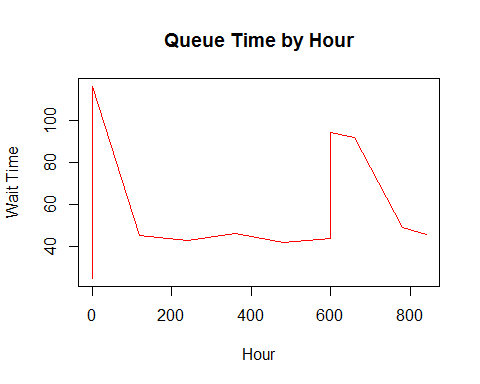
February 14, 2015

#### 1.1

1. system: cafe, entities: customers, attributes: checking-account balance, activities: buying lunch, events: arrival/departure/eating, state variables: number of cashiers/number of customers
2. system: grocery store, entities: customers/cashiers, attributes: checking-account balance, activities: buying groceries, events: checking out, state variables: number of cashiers/number of customers
3. 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
4. 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
5. system: ER, entities: doctors/patients, attributes: injuries, activities: healing injuries, events: taking medicine, state variables: number of doctors/number of patients/empty rooms
6. 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
7. 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

########################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")  
#################################################################

1. Average time in the queue:

mean(wt)

## [1] 59.80794

1. Average processing time:

mean(se-sb)

## [1] 45.3498

1. Max time in the system:

max(wt)

## [1] 116.029

#### 2.2

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

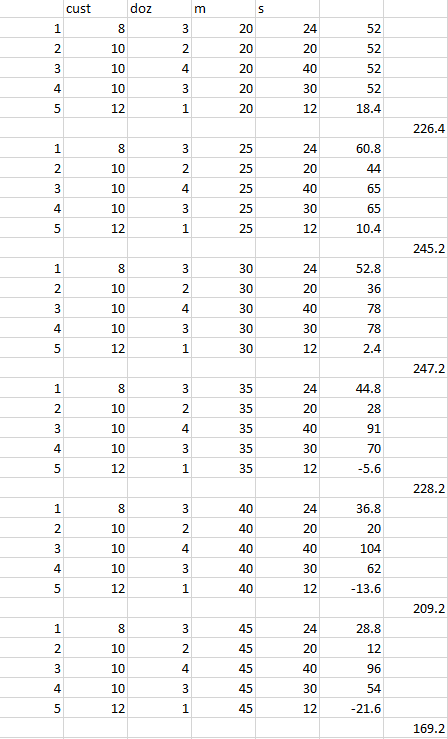
## [1] 20.4

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

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

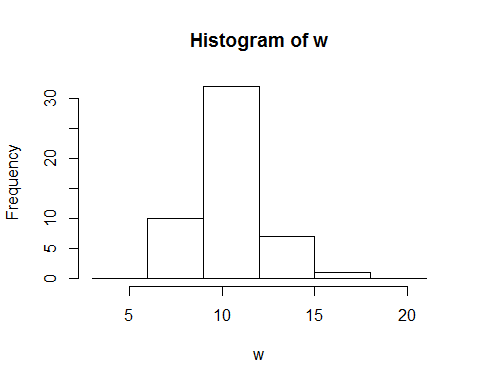


It's hard to say with only 5 days of simulation, but from the above excel sheet we can see that making 30-35 bagels seems to be optimal.

#### 2.4

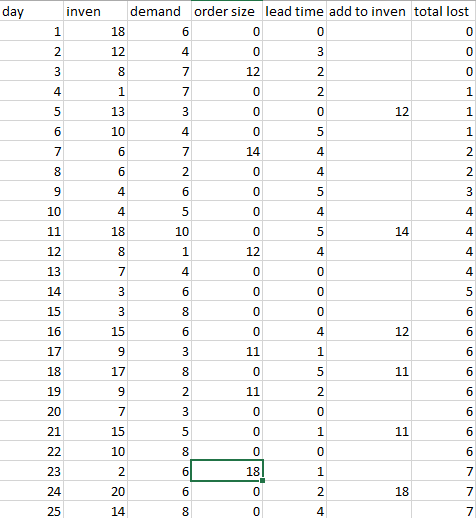
#### 2.5

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

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



There was 6 lost orders according to the above spreadsheet

#### 2.8