Classroom Simulation: Distributions of Sales and Queue Lengths at a Fast-Food Counter

Eric A. Suess, Bruce E. Trumbo, David R. Ahlberg

California State University, East Bay (Hayward)

Abstract: Lunch-hour sales at a fast food outlet are modeled and simulated using R. Concepts illustrated include: expectation, variance, and distribution of the sum of a random number of random variables; some relationships among binomial, Poisson, uniform, and exponential distributions; and average length of an M/M/s queue. Level suitable for upper-division probability students.

Keywords: Simulation; R; random sum of random variables; queue length; exponential interarrival times; teaching.

1. Simulation in the Classroom

The wide availability of ever faster computers and high quality, free software R has made feasible the use of meaningful simulations in upper-division probability courses. Depending on the level and interests of students, simulations can be intuition-building in-class demonstrations of basic principles or relatively simple programming projects, in which students are asked to make modifications in existing computer code.

The programming necessary to reproduce results and figures in the Sections 2–6 of this paper (normality of the sample total, convergence of binomial to Poisson, distribution of the sum of a random number of random variables, and exponential distribution of interarrival times of a Poisson Process) is very elementary, so some explanation of how simulations can be programmed in R is possible even for students with no background in programming. Results in Section 7 (simulating M/M/s queues) require somewhat more advanced programming. Results are presented here in a style we hope can easily be modified for in-class demonstrations. Some sample code for Section 7 is shown in the Appendix.

2. A Problem to Model: Fast Food Lunch

Suppose historical data show that a carry-out fast-food "restaurant" in the center of a city serves about $n = 225$ people for lunch between noon and 1 p.m. each weekday, and that the bill for a randomly chosen customer averages $\mu = \$5.00$ with a standard deviation of $\sigma = \$1.20$. So the average of daily lunchtime sales is about $n\mu = \$1125$.

However, on a particular day, the sales are only \$1050. So the relatively new restaurant manager wonders whether something has gone wrong. Has some change prompted the customers to order less? Have the cashiers contrived to pocket some of the money without ringing up sales? If detailed historical data were available, we could just look back to see how frequently sales have been as low as \$1050, but we only know the averages quoted above.

2. A First Model: Sum of *n* Normal Bills

We want to make a probability model to see whether \$1050, which is \$75 below average, is a suspiciously low figure. As a first model, we might say that the total of the sales on any one day is $S = X_1 + X_2 + ... + X_{225}$. So $E(S) = n\mu = 225(\$5) = \$1125$, as mentioned above. Moreover, assuming independence among customers, $SD(S) = \sigma\sqrt{n} = \$18$. If the bills of individual customers are normally distributed, then the total S is also normal, with

$$P\{S \leq 1050\} = P\{Z \leq -75/18 = -4.17\} = 0.000015,$$

where Z is standard normal. [In R, `pnorm(1050, 1125, 18)` returns 1.545430e–05.] The normal distribution is a fairly safe model here.

According to the Central Limit Theorem, the sum S of such a large number of individual sales should be approximately normal, even if the individual sales are not normal. So, based on these preliminary computations, the lunchtime receipts of only \$1050 for this particular day do seem suspiciously small. See Figure 1.

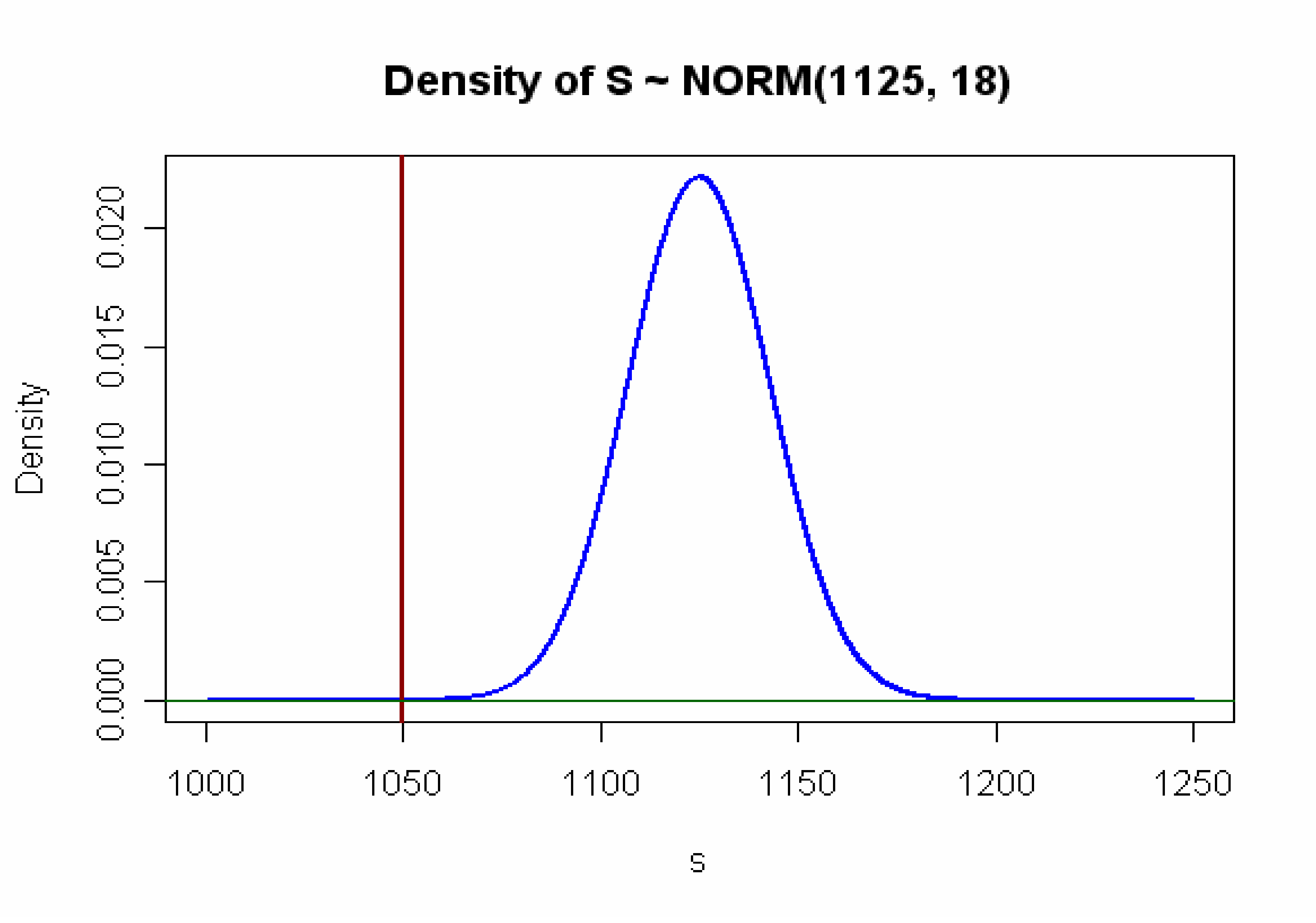

Figure 1. If S ~ NORM(1125, 18), then P{S ≤ 1050} is nearly 0, the area under the curve to the left of the vertical red line.

3. Modeling a Random Number of Customers

The analysis of Section 2 assumes there are *exactly* 225 lunchtime customers every day. However, a more realistic model would take into account that the number of customers on a given day is also random. So let N be a random variable with $E(N) = 225$. Now the total of the receipts varies not only because each customer spends a random amount of money, but also because the number of customers is random.

What is a reasonable distribution for N? Here is one approach. Suppose there are $M = 22{,}500$ people in the downtown area each day who are potential customers. Also suppose that each of them has 1 chance in 100 of having lunch at our particular restaurant on a given day. Then the random number N of our actual customers is binomial with $M = 22{,}500$ trials and success probability $p = 1/100$. This gives $E(N) = np = 225$. The vertical green bars in Figure 2 show probabilities from this binomial distribution.

Of course, the population size 22,500 and the success probability 1/100 are both only very rough guesses, but it turns out that *any* binomial random variable S based on a large population size M, a small success probability p, and mean $E(S) = Mp = 225$ has nearly the same distribution—the Poisson distribution with mean 225. Some values of this Poisson distribution are shown as small blue circles in Figure 2.

Clearly, the distributions BINOM(22,500, 1/100) and POIS(225) are not much different. For a numerical comparison, Figure 3 shows some cumulative probabilities for the three random variables $V_1 \sim$ BINOM(22,500, .01), $V_2 \sim$ BINOM(45,000, .005), and $W \sim$ POIS(225). All three distributions have mean 225. Cumulative probabilities agree to two or three decimal places.

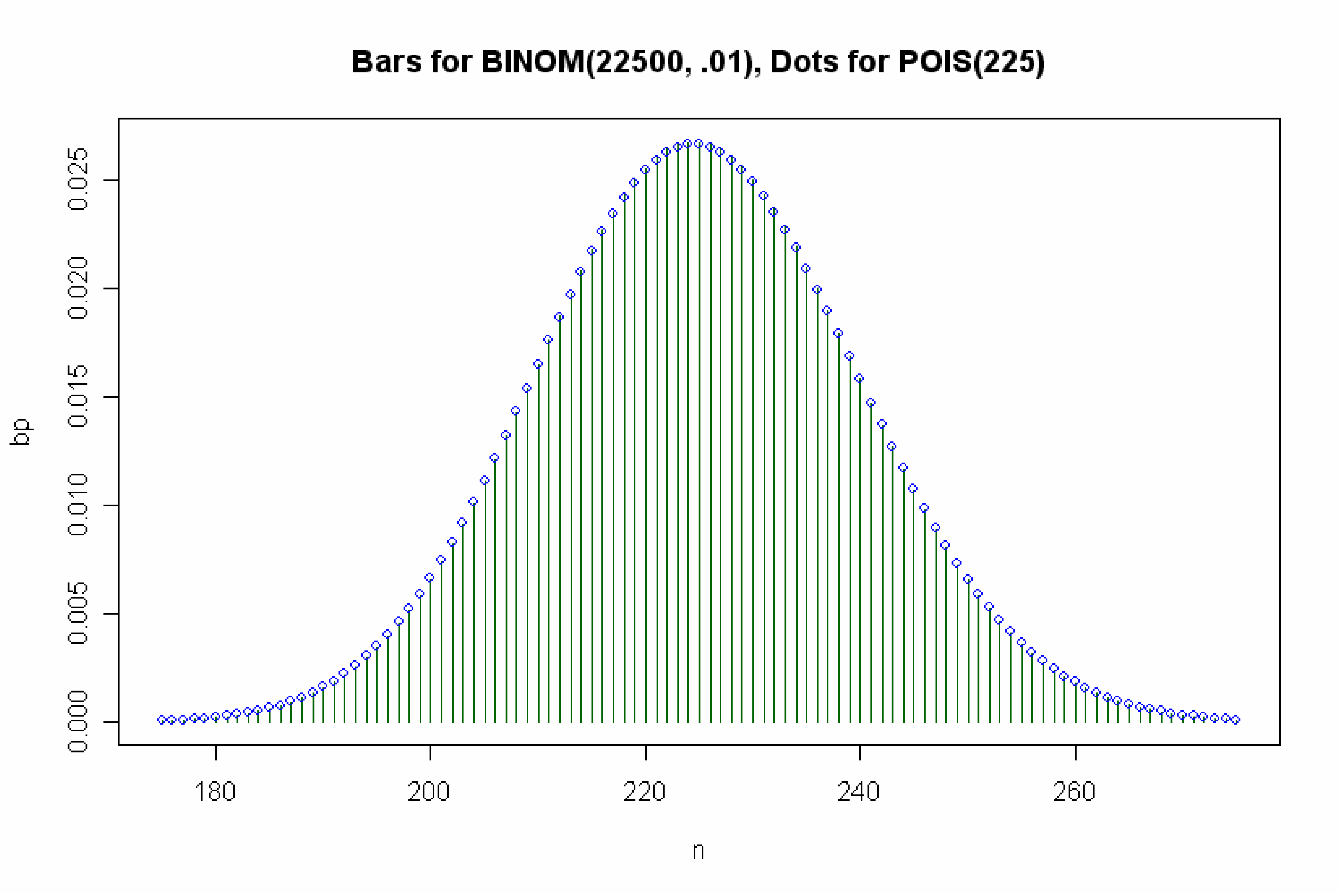

Figure 2. Nearly equal distributions: Vertical bars show BINOM(22,500, 0.01) and blue circles show POIS(225).

k	P{$V_1 \le k$}	P{$V_2 \le k$}	P{$W \le k$}
200	0.048	0.049	0.049
205	0.094	0.095	0.095
210	0.166	0.166	0.167
215	0.264	0.265	0.265
220	0.385	0.386	0.386
225	0.518	0.518	0.518
230	0.647	0.647	0.647
235	0.761	0.760	0.760
240	0.850	0.850	0.849
245	0.914	0.913	0.913
250	0.954	0.954	0.954

Figure 3. Numerical illustration that binomial distributions with large *M*, small *p*, and *Mp* = 225 are nearly POIS(225).

R code for making Figure 3 is essentially as follows:

```
k = seq(200, 250, by=5)
CDF1 = pbinom(k, 22500, .01)
CDF2 = pbinom(k, 45000, .005)
CDF3 = ppois(k, 225)
round(cbind(k, CDF1, CDF2, CDF3), 3)
```

More generally, POIS(225), the Poisson distribution with mean 225, is the limit of binomial distributions as *M* becomes infinite and *p* approaches 0 in such a way that *Mp* = 225 remains constant.

This illustrates that the details about M and p do not matter much as long as we believe that the number N of luncheon customers on a particular day results from a large number M of independent decisions, each with probability p of choosing to eat lunch at our restaurant. So we assume $N \sim$ POIS(225).

4. A Random Sum of Random Variables

Now we investigate the model in which total lunchtime sales on a randomly chosen day is the sum of a random number N of random variables. $S = X_1 + X_2 + \ldots + X_N$, where $E(N) = 225$ and X_i are distributed with mean \$5.00 and standard deviation \$1.20, and the X_i are independent of each other and of N.

From arguments involving conditional probability, we have

$$E(S) = E(N)E(X) = 225(\$5.00) = \$1125$$

as before. But now

$$V(S) = E(N)V(X) + V(N)[E(X)]^2 = 5949, \text{ so that } SD(S) = \$77.13.$$

The formula for the mean and the first term in the formula for the variance are just as they would be if N were a constant value with $V(N) = 0$. The second term in the variance expresses the increased variability that comes from having a random number N rather than a fixed number n of customers.

Taken alone, these results for the mean and standard deviation of total sales S say nothing about the probability $P(S \leq 1050)$. But, assuming N to be Poisson, and the X_i to be normal, we can get this probability from simulation.

Figure 4 shows the R code used in our simulation. In this code we assumed that $N \sim$ POIS(225) and $X_i \sim$ NORM(5, 1.20) and we simulated receipts for ten thousand lunch hours. The histogram in Figure 5 shows the simulated distribution of S. From this distribution we see that about 17% of the simulated daily receipts were $1050 or smaller. So while the day that raised the manager's suspicions was certainly below average, it is not really a very unusual day when we take into account that the number of customers can vary randomly from one day to the next.

```
mu.n = 225;  mu.x = 5;  sd.x = 1.2
m = 10000;  s = numeric(m)
for (i in 1:m)
   {
   n = rpois(1, mu.n)
   s[i] = sum(rnorm(n, mu.x, sd.x))
   }
mean(s)  # simulated E(S)
sd(s)    # simulated SD(S)
mean(s <= 1050)  # simulated P{X ≤ 1050}
hist(s, prob=T)  # histogram only
```

Figure 4. R code used to simulate P{*S* ≤ 1050} = 0.17 and make the histogram in Figure 5. On each of 10,000 randomly chosen days, we first simulate the number *N* of customers, then their bills, and finally the total receipts for the day.

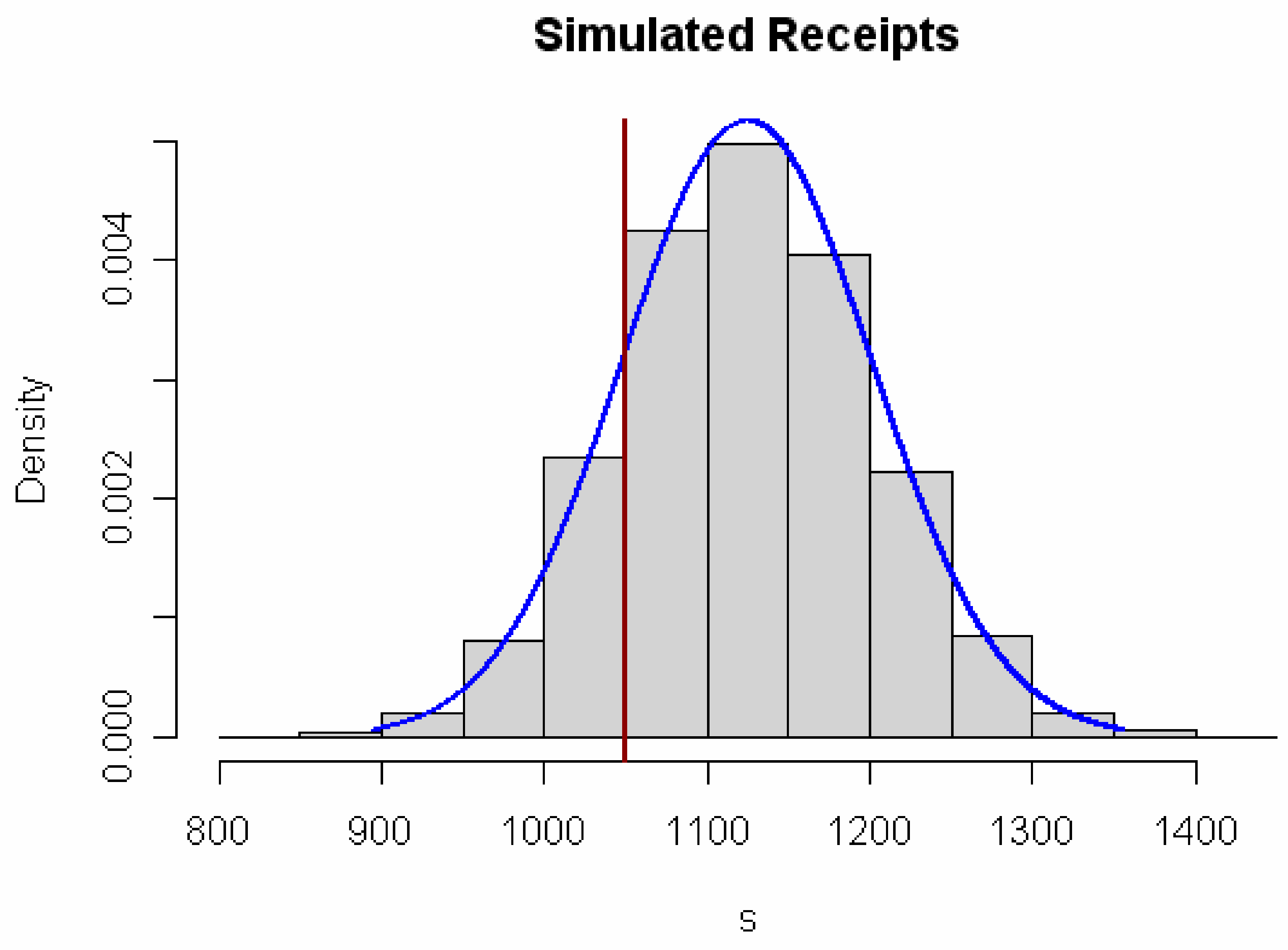

Figure 5. Simulated histogram of *S* using normally distributed customer bills, compared with a normal curve based on the theoretical mean and standard deviation of *S*. The areas to the left of the vertical red line (in the histogram and under the curve) are both about 17%.

It is always well to validate a simulation to the extent possible. One run of our simulation gave $E(S) \approx 1124.67$ and $SD(S) \approx \$78.08$, which are close to the theoretical values. Guessing that the resulting mixture of the 10,000 normal distributions might be approximately NORM(1125, 77.13), we superimposed this normal density on the histogram in Figure 5. If S has this normal distribution then $P(S \leq 1050) = 0.1654$, so in this case a *normal approximation* based on the known mean and standard deviation gives serviceable results.

5. A Discrete Model for Individual Purchases

The manager thinks he might get a more realistic model by using information from a couple of days of register tapes on actual amounts of customer bills. Because of the prices charged for various luncheon items, suppose it turns out that almost all of the bills are in one of the amounts \$3.18, \$3.64, \$4.33, \$4.99, \$6.03, or \$7.50, and that these amounts occur roughly in the proportions 10%, 10%, 20%, 30%, 20% and 10%, respectively. Accordingly, customer bills still have mean \$5.00 and standard deviation \$1.20. Of course, now the distribution is highly discrete instead of normal. But the theoretical values $E(S) = \$1125$ and $SD(S) = \$77.13$, which do not depend on the shape of the distribution of the X_i, remain valid.

Simulating ten thousand lunchtimes with this discrete model gives the approximate value $P\{S \leq 1050\} = 0.16\%$. Again here, values of $E(S)$ and $SD(S)$ are in good agreement with the theoretical values; our simulated results were 1125.85 and 76.47 respectively. Also, a histogram of the simulated distribution is difficult to distinguish from that of Figure 5. The only change in the R code for this simulation is to use the line

```
s[i] = sum(sample(x, n, repl=T, prob=w))
```

inside the loop, where `x` is the vector of six prices and `w` is the vector of their relative frequencies.

6. Issues in Probability Modeling

So far, we have had some reasonable success in modeling the distribution of total receipts S. With very few exceptions, useful probability models are based on assumptions that are not likely to be exactly correct. But with some caution and some luck we can hope to get serviceable approximations.

- We do not know which binomial model would be the right one for the distribution of N. But we have seen that the particular choice doesn't matter because any reasonable binomial model is nearly Poisson.

- We don't know whether the discrete model, the normal model, or some kind of model we have not thought to consider is the best one for the individual customer bills X_i. However, mainly because of the Central Limit Theorem, we have seen that two very different choices give about the same result.

There are some additional modeling issues we have not yet discussed that do not turn out to be worrisome.

> First, it is possible that groups of customers may arrive and place very similar orders. Even so, it is probably OK to assume that the X_i are essentially independent.
>
> Second, because our fast food operation is organized so that most orders—from the cheapest to the most expensive—are placed simply as numbers from a menu, it is probably safe to assume expensive orders do not slow the traffic enough to influence the number N of people served during the lunch hour. So it also seems safe to assume that N is independent of the X_i.

There are, nevertheless, other worrisome issues to consider. If there are about 225 customers per hour, that means on average one customer arrives every 16 seconds. Suppose there are two servers (order takers), each of whom handles a customer in exactly 30 seconds. Then, if customers are "considerate" enough to arrive at equally spaced 16-second intervals, everyone will get waited on with little or no delay. Thus N will be determined by customer demand as we have modeled it—and not on whether the system gets overloaded so that some potential customers can't be served during the lunch hour.

Because such an orderly arrival pattern seems unlikely in the real world, we wonder what kind of a probability model would reasonably describe customer arrivals. One simple and fairly reasonable assumption is that customers arrive independently and according to a uniform distribution within the 60 minute lunch hour from noon to 1 p.m.

A simple simulation illustrates this distribution of arrival times. First, we used the distribution POIS(225) to simulate $N = 218$ customers arriving during a particular lunch hour. Then we simulated 218 independent arrival times T_i according to the distribution UNIF(0, 60). Not surprisingly, a histogram (not shown) of these arrival times has bars of approximately equal heights.

A beginning student might guess that the intervals W_i between these arrivals would also be roughly uniform, but that guess is far from the truth. A continuation of our simulation shows these interarrival times to be distributed exponentially. With an average rate 225 per 60 minutes or $\lambda_{\text{arr}} = 225/60 = 3.75$ arrivals per minute, the *mean* time between arrivals is $1/3.75 = 0.267$ minutes or 16 seconds.

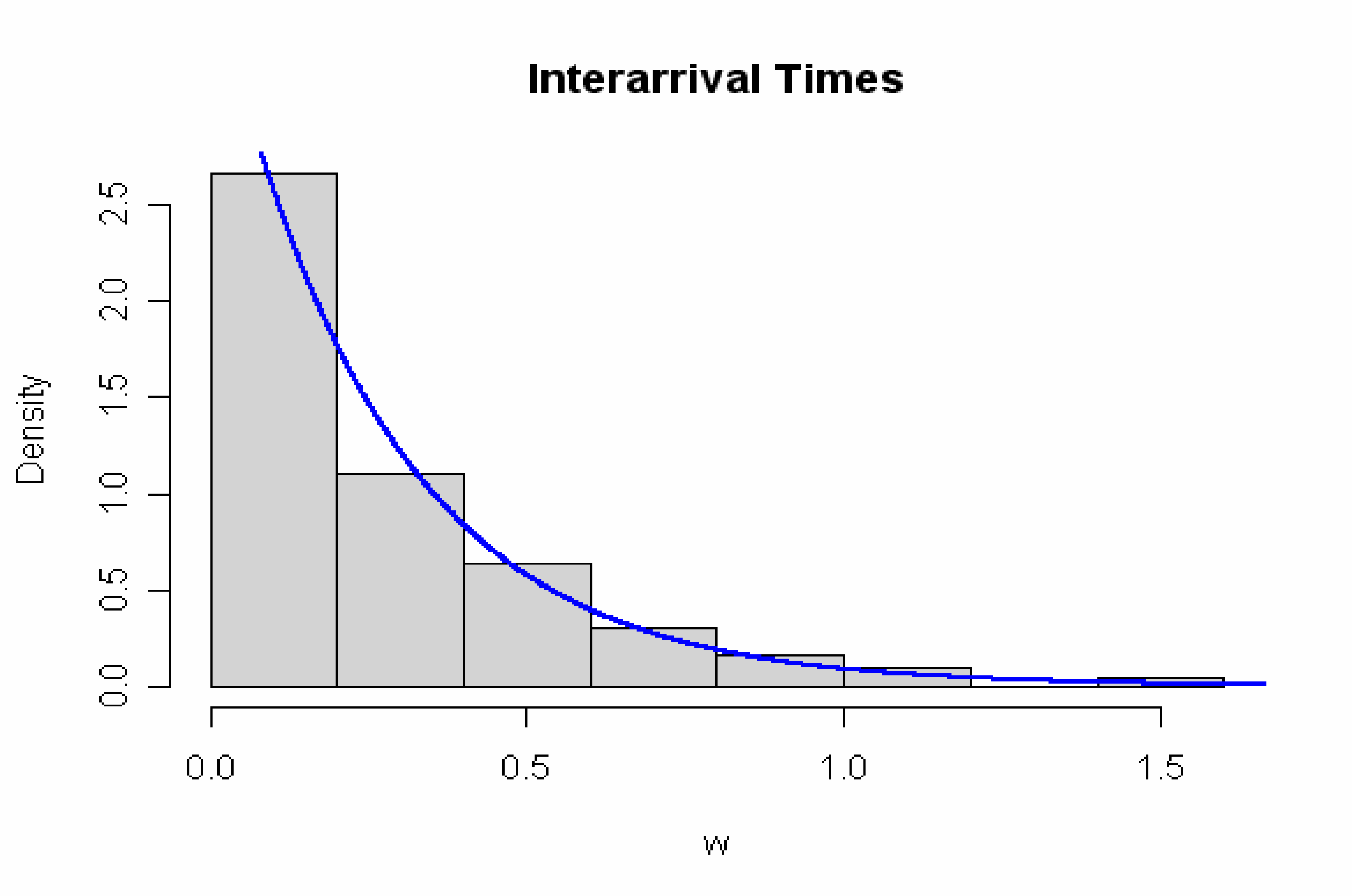

Figure 6. Exponentially distributed interarrival times between successive customers where *N* = 218 arrival times are uniformly distributed on (0, 60).

To find the interarrival times for our simulated lunch hour, we sort the arrival times T_i in order from earliest to latest, and take differences to find the N interarrival times W_i (including the first, which we take to start at time 0). In R, three simple statements do the entire simulation:

```
n = rpois(225);  t = runif(n, 0, 60)
w = diff(sort(c(0, t)))
```

In one simulated lunch hour, the average interarrival time was 0.27 minutes. Figure 6 shows the histogram of these interarrival times along with the density function of the exponential distribution with rate 3.75 customers arriving per minute, which fits the histogram very well.

7. Queueing Models

In this case the choice to use an exponential model for service times is motivated, at least in part, by convenience. If we make this choice, then our system of arrivals and departures follows a much-studied probability model known as the ***M/M/2 queue***.

A characteristic of most queues is an interplay between the rate of arrivals and the rate at which customers get served and depart. In our queueing model, the arrival rate is 3.75 per minute and the capacity for service is 2 servers times $\lambda_{\text{srv}} = 2$ per minute, or 4 customers per minute altogether. Because $2\lambda_{\text{srv}} = 4 > \lambda_{\text{arr}} = 3.75$ our model reaches a steady state in which the arrivals and departures are in balance. (When the capacity is less than the arrival rate, the number of people waiting to be served increases relentlessly to infinity.)

Assuming steady state conditions during the lunch hour, standard formulas for an M/M/2 queueing system give the following *average* values: 225 customers arrive, 225 customers are served, 15.5 customers are waiting in line or being served at any given instant, and a customer's wait until being served is 4 minutes and 8 seconds (4.13 minutes).

However, under these conditions where the system is operating at almost full capacity (with $2\lambda_{\text{srv}}$ only a little larger than λ_{arv}) the actual number waiting in line can occasionally get very large. The servers are almost always busy so that only about 9% of arriving customers get served immediately without having to wait in line.

Figure 7 shows ten simulated hours under these conditions. The vertical axis plots the number of customers "in the system," including the 0, 1, or 2 people being served as well as any who are waiting in line. In the real world, one wonders whether some customers might be lost to nearby restaurants when arriving customers see 20 or more people already in the system. (Separate simulation runs in which *all* such customers are assumed lost show the average number of customers lost in this way to be between 4 and 5 per hour. This is a much more stable simulation because a few lost arrivals help the queue to decrease in size just when it most needs this help.)

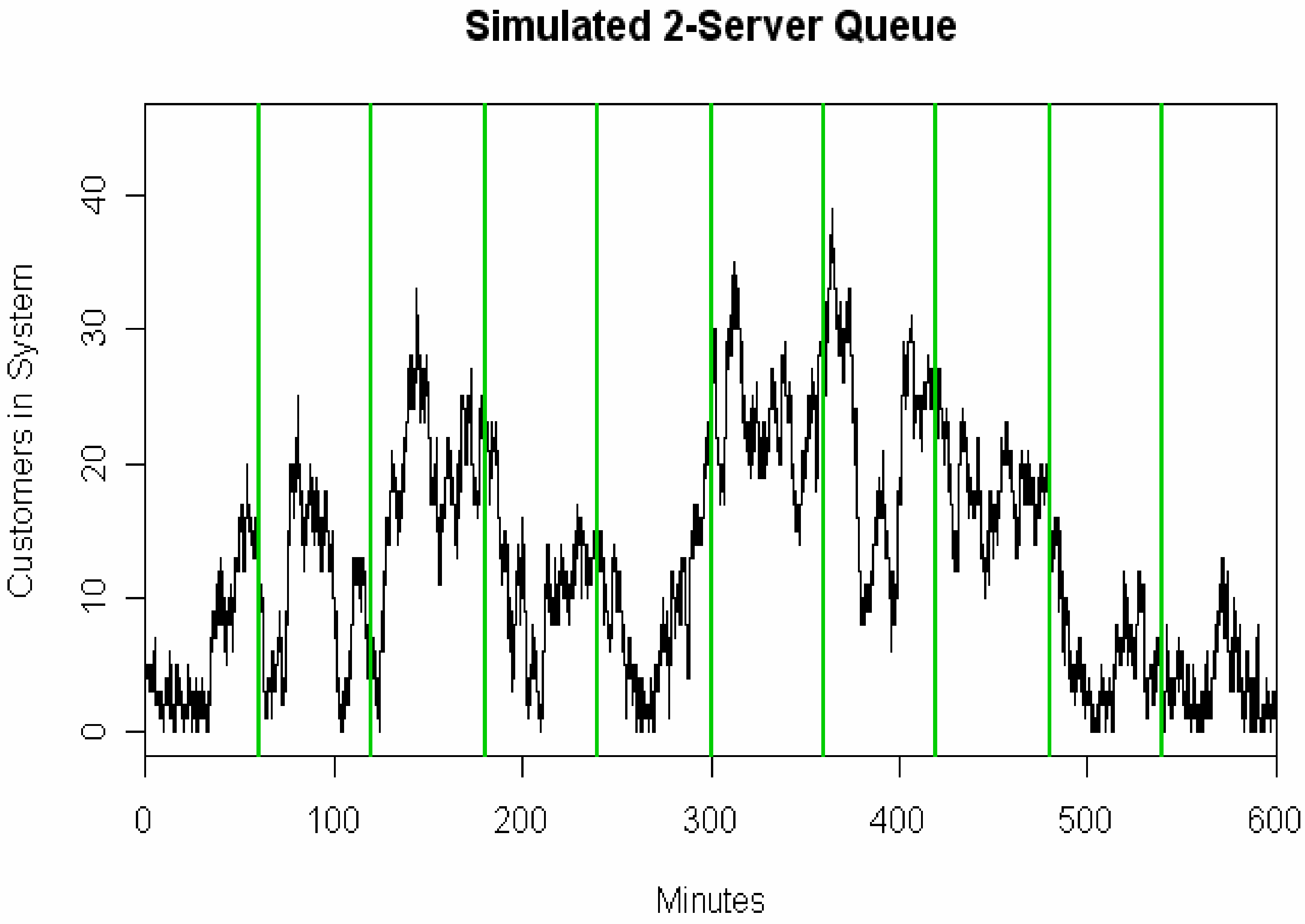

Figure 7. Simulation over a ten hour period of an M/M/2 queue with λ_{arr} = 3.75/min and 2λ_{srv} = 4/min.

R code for the simulation in Figure 7 is shown in the Appendix. Briefly, the method of simulation is to model a potential arrival event as exponential with rate λ_{arr} and a “competing” potential departure event as exponential with rate $b\lambda_{\mathrm{srv}}$, where b is the number of servers currently busy. The one of these two events that occurs first “wins” and not only determines the time interval between events, but also whether the length of the queue increases or decreases by 1 at the end of that interval.

In contrast to the two-server queue above, suppose there are *three* servers, each with service rate $\lambda_{\mathrm{srv}} = 2$. (Maybe the manager helps serve during the lunch hour.) Then the average number of arrivals and departures per hour is still 225. But the average number of customers in the system decreases to 2.52 and the average time before being served is only about 40 seconds. This system has a lot of excess capacity. About 61% of the arriving customers are served immediately without having to wait in line. Figure 8 shows a simulated ten-hour period for this system.

Going back to our original problem of accounting for the day with low sales, we see that, in theory, neither of these queueing systems shows enough modification of the flow of served customers to make much of a difference in total lunch hour sales. Either way, the number of customers served during the lunch hour is still approximately distributed as POIS(225).

In practice, the two-server queue shown in Figure 7—with almost no excess capacity—might sometimes result in an unpleasant lunch hour for servers and customers alike. Here any little departure from the assumptions of the model (such as time out to replace the roll of paper in one of the cash registers or a brief restroom break for one of the servers) might lead to an extremely long waiting line and perhaps a noticeable loss of business.

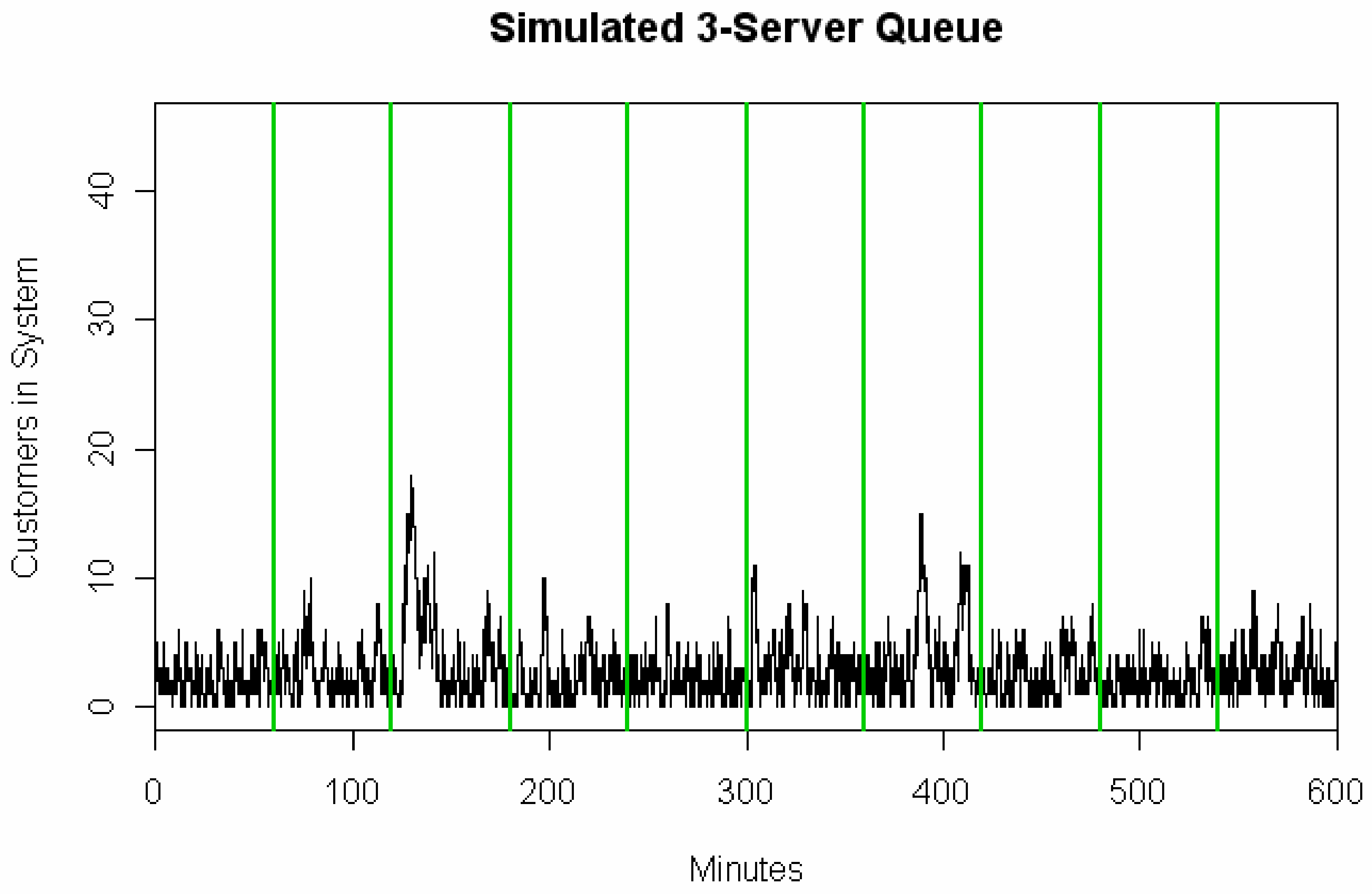

Figure 8. Simulation over a ten hour period of an M/M/3 queue with λ_{arr} = 3.75/min and 3λ_{srv} = 6/min. The average number of people in the system is smaller than in Figure 7.

Queueing theory is used in a variety of applications, particularly in industrial engineering, communications, and the design of networks, hardware, and software for computers. Formulas for the behaviour of the standard models we have considered here are available in many books on stochastic processes and operations research. However, even slight departures from standard models can lead to systems that are very difficult or impossible to solve analytically. Consequently, simulation methods are extensively used in practical queueing applications.

References

Roe Goodman: *Introduction to stochastic models,* Sections 5.4 and 11.6, Benjamin / Cummings (1988).

James H. Higgins and Sallie Keller-McNulty, *Concepts in probability and stochastic modeling,* Sections 7.9 and 8.4, Duxbury (1995).

Sheldon M. Ross: *Introduction to probability models,* 6th edition, Sections 3.4 and 8.8. Academic Press (1997).

Appendix: Annotated R Code for Section 7
(Simulation of an M/M/s Queue, Including Figures 6 and 7)

```
## Queue for a Fast Food Counter   For JSM 2007 Poster
## Version: July 2007
## B. Trumbo and D. Ahlberg
##Comments to: bruce.trumbo@csueastbay.edu

set.seed(12)         # seed used for Fig 6

# M/M/s Queue Parameters
s = 2            # number of servers (2 for Fig 6, 3 for Fig 7)
lam.a = 3.75     # customer arrival rate
lam.s = 2            # service rate (each server)

# Simulation set up
m = 500000       # events
w = numeric(m)   # interarrival times
n = numeric(m)       # number in system
b = numeric(m)   # number being served = number of busy servers
a = logical(m)       # TRUE if event is Arrival, FALSE if Departure
n[1] = 5             # number in system at start (arbitrary)
b[1] = min(n[1],s)   # number in service at start (DO NOT CHANGE)
```

```
# Simulation: One event (arrival or departure) per loop
for (i in 2:m)
    {
    x = rexp(1, lam.a) # potential time to next arrival
    y = Inf            # pot. time to next dep. ('Inf'inity if queue empty)
    if (b[i-1] > 0) y = rexp(1, b[i-1]*lam.s)  # dep. rate depends on b
    #if (n[i-1] >= 20) x = Inf  # If used, limits number in system to 20
    w[i-1] = min(x, y)      # actual time to next event
                            # Categorize event as arrival or departure
    if (x <= y) {arr = T}
    else {arr = F}
    a[i] = arr
    # Track number n in system; number b in service
    if (arr)                  n[i] = n[i-1] + 1
    if (arr & (b[i-1] < s))   b[i] = b[i-1] + 1
    if (arr & (b[i-1] == s))  b[i] = b[i-1]
    if (!arr & (n[i-1] > 0))  n[i] = n[i-1] - 1
    if (!arr & (n[i-1] <= s)) b[i] = b[i-1] - 1
    if (!arr & (n[i-1] > s))  b[i] = b[i-1]
    }
```

```
# Numerical Output
cbind(w, a, n, b)[1:20,]                      # Summary of first 20 events
avg.l = sum(w*n)/sum(w);  avg.l               # Avg number in system
sd.l = sqrt(sum(w*n^2)/sum(w)- avg.l^2); sd.l # SD number in system
avg.w = avg.l/lam.a;  avg.w                   # Avg time in system

q = n - b                                     # Number in queue
avg.lq = sum(w*q)/sum(w); avg.lq              # Avg number in queue
sd.lq = sqrt(sum(w*q^2)/sum(w) - avg.lq^2 ); sd.lq   # SD number in queue
avg.wq = avg.lq/lam.a; avg.wq                 # Avg time in queue
sum(w[b==0])/sum(w)                           # % of time all servers idle
sum(w[b<s])/sum(w)                            # % of time cust. served
immed.

# Graphical Output
plot(cumsum(w),n,type="s",xlim=c(0,600), ylim=c(0, 45), xaxs="i",
  main=paste("Simulated ",s,"-Server Queue", sep=""),
  ylab="Customers in System", xlab="Minutes")
abline(v=seq(60, 599, by=60), col="green3", lwd=2)
```