Assignment 3: Discrete Events

Jonathan De Los Santos

4/27/2021

Contents

1 Problem 1: Queuing Systems

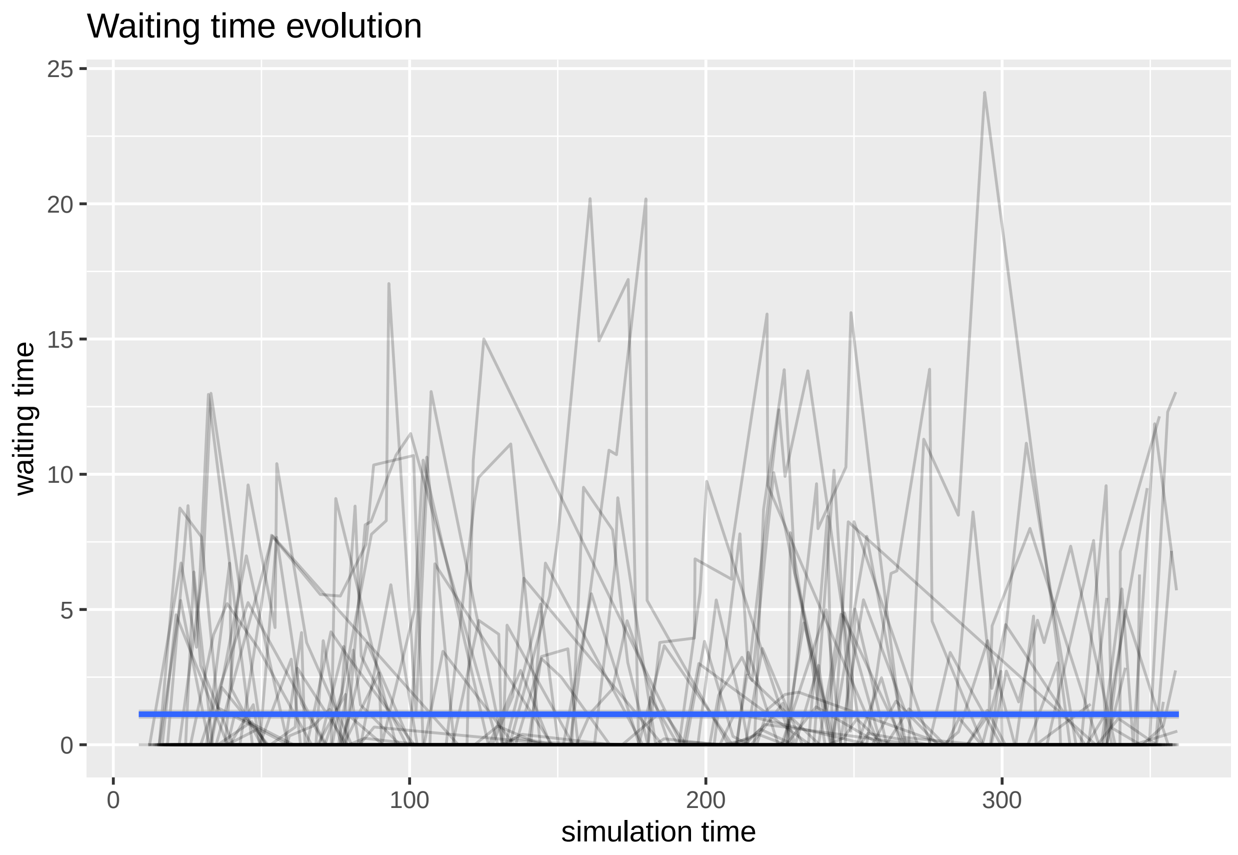
The time between incoming phone calls to customer service is Exponential with the mean of 14 minutes. Each call typically takes min=5, max=20, and most likely = 10 minutes, (Triangularly distributed). Simulate the system based on the following scenarios (run each scenario for 6 hours, with 50 replications).

1.1 a) Several Counters, Single Queue

There are two operators, and the system has only a single queue. Report the histogram and mean for the average waiting time.

The calculated average wait time is short in this scenario, only 1.13 minutes. The short wait time is likely due to the exponential distribution of incoming calls being higher than the mode of the call lengths. This is supported by the histogram which does not show any meaningful increase in waiting time over the duration of the simulation.

```r
#install.packages('simmer')
library(simmer)
library(triangle)

set.seed(123)

call <-
  trajectory("Call Routing") %>%
  seize("operator", 1) %>%
  timeout(function() rtriangle(1, 5, 20, 10)) %>%
  release("operator", 1)

#install.packages('simmer.plot')
library(simmer.plot)

set.seed(123)

envs <- lapply(1:50, function(i) {
  simmer("Service") %>%
    add_resource("operator", 2) %>%
    add_generator("call", call, function() rexp(1, 1/14)) %>%
    run(360)
  })

resources <- get_mon_resources(envs)

arrivals <- get_mon_arrivals(envs)

plot(arrivals, metric = "waiting_time")
```



```
waitTime = (arrivals$end_time - arrivals$start_time) - arrivals$activity_time
mean(waitTime)
```

```
## [1] 1.126063
```

1.2 b) Multiple Operators, Independent Queues

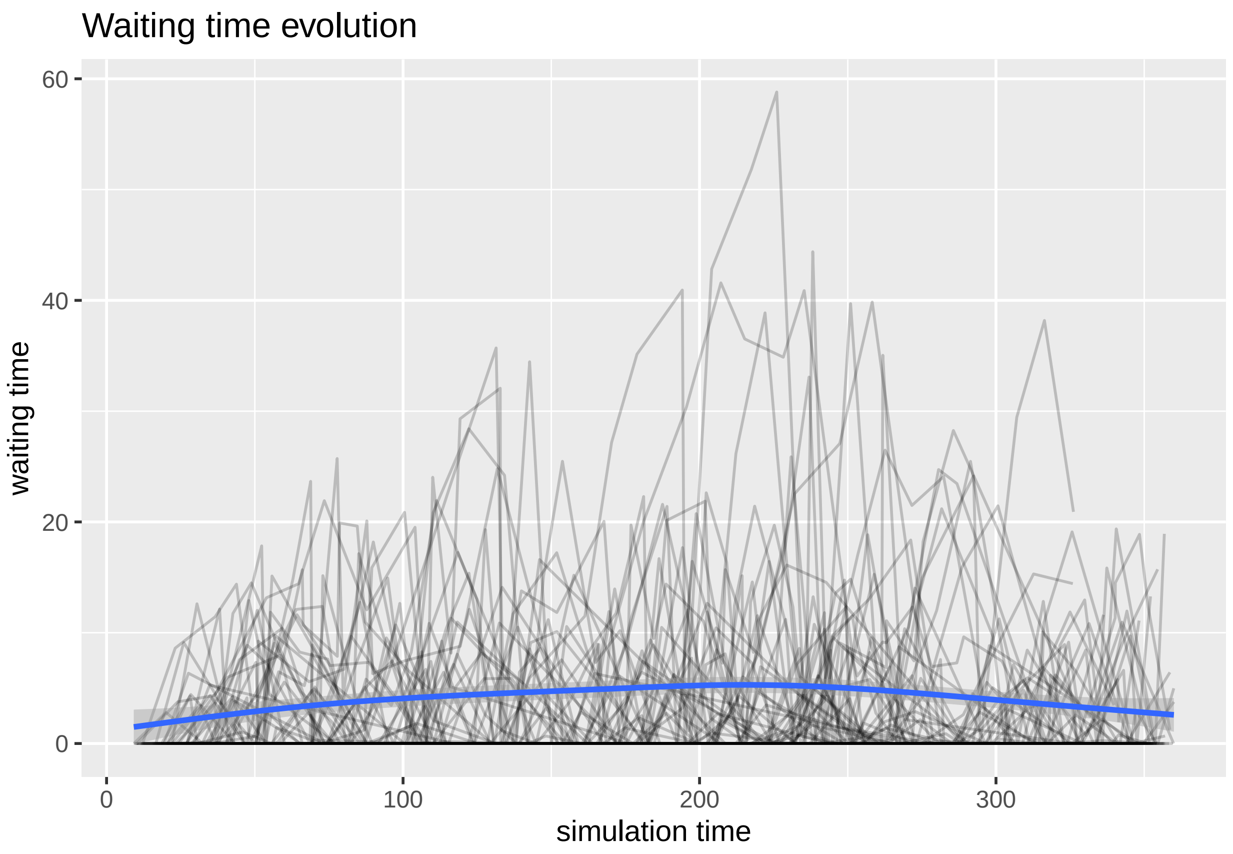
> There are two operators, and each operator has its own queue. Queue routing is random. Report the histogram and mean for the average waiting time.

This example retains both operators, but now each has their own queue to which callers are randomly assigned. The average wait time is now 4.12 minutes, and interestingly the histogram reveals a slight spike which may more appropriately viewed as a hill or bunny slope. Larger simulations may reveal whether this is a recurring pattern or an anomaly that reverts to the mean over time.

```
set.seed(123)
call <-
  trajectory("Call Routing") %>%
  select(c("operator1", "operator2"), policy = "random") %>%
  seize_selected() %>%
  timeout(function() rtriangle(1, 5, 20, 10)) %>%
  release_selected

set.seed(123)
```

```r
envs2 <- lapply(1:50, function(i) {
  simmer("Service") %>%
    add_resource("operator1", 1) %>%
    add_resource("operator2", 1) %>%
    add_generator("call", call, function() rexp(1, 1/14)) %>%
    run(360) })

arrivals3 <- get_mon_arrivals(envs2)
plot(arrivals3, metric = "waiting_time")
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



```r
waitTime2 = (arrivals3$end_time - arrivals3$start_time) - arrivals3$activity_time
mean(waitTime2)
```

```
## [1] 4.108974
```

1.3 c) Comparison

Compare the average waiting time of part a and b. Is there any meaningful difference? Explain.

The simulated single queue wait times are much shorter with a significantly narrower confidence interval visualized in the box plot. This phenomenon can be explained by the extra queue and random sorting.

Rather than callers being routed to the next available resource, they are arbitrarily waiting for a specific resourced to be freed.

This creates waste that accounts for the difference in average wait time, as well as greater variation in those times accounting for the greater standard error.

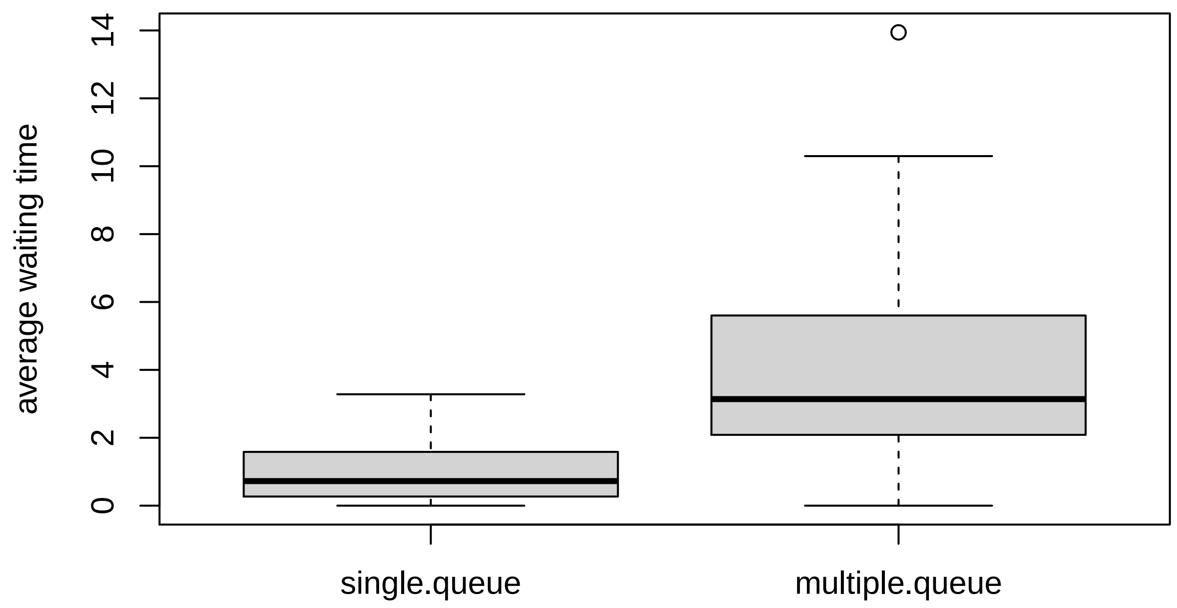
```r
# Single Queue
arrivals2 = cbind(arrivals, waitTime)
xbar = aggregate(arrivals2$waitTime, by = list(arrivals2$replication), FUN = mean)
quantile(xbar$x, c(0.025 , 0.975))
```

```
##          2.5%         97.5%
## -1.416645e-15  3.054537e+00
```

```r
# Multiple Queue
arrivals4 = cbind(arrivals3, waitTime2)
xbar2 = aggregate(arrivals4$waitTime2, by = list(arrivals4$replication), FUN = mean)
quantile(xbar2$x, c(0.025 , 0.975))
```

```
##        2.5%       97.5%
##  0.07056625 10.13033321
```

```r
# Comparison
boxplot(xbar$x,
        xbar2$x,
        names = c("single.queue", "multiple.queue"),
        ylab = "average waiting time")
```



2 Problem 2: Queuing with Variable Service Time

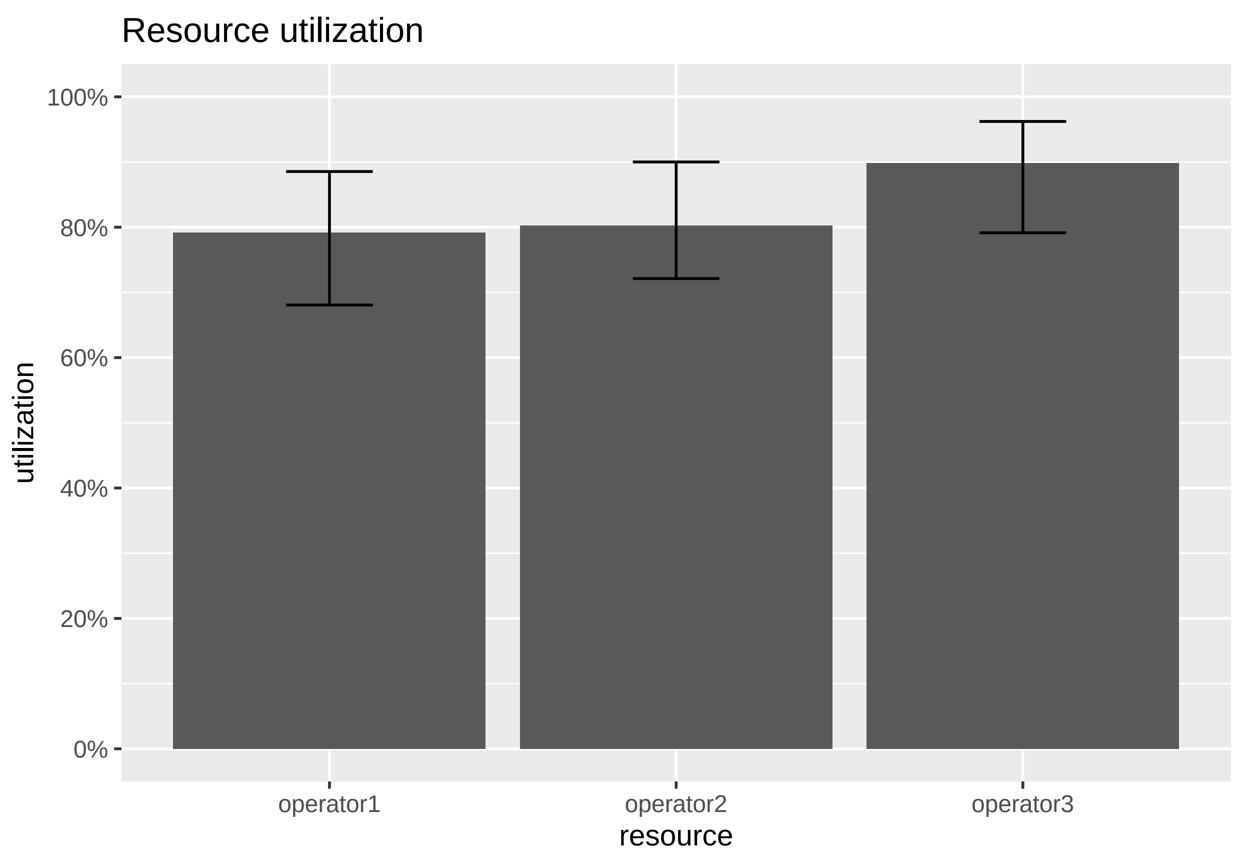
Suppose the time between incoming phone calls to customer service is Exponential with the mean of 5 minutes. And, suppose there are 3 operators and each operator has its own queue. Phone calls will be directed to queues randomly. Operators 1 and 2 respond to calls typically between min=5 to max=20 minutes (Uniformly distributed). The third operator is chattier, and their service takes more time, which is Normally distributed with a mean of 15 and a standard deviation of 3 minutes. Run your model for 6 hours, with 50 replications.

2.1 a) Utilization Plot

Discuss what you learned from the plot from each operator.

The first two operators are utilized equally as their response distributions are the same. The third caller with their longer service distribution has a higher utilization indicated by the plot.

```r
library(simmer)
call <-
  trajectory("Call Routing") %>%

  branch(option = function() sample(1:3, 1, replace = T), continue = c(T,T,T),
        trajectory("A") %>%
        seize("operator1", 1) %>%
        timeout(function() runif(1, 5, 20)) %>%
        release("operator1", 1),

        trajectory("B") %>%
        seize("operator2", 1) %>%
        timeout(function() runif(1, 5, 20)) %>%
        release("operator2", 1),

        trajectory("C") %>%
        seize("operator3", 1) %>%
        timeout(function() rnorm(1, 15, 3)) %>%
        release("operator3", 1)
  )


set.seed(123)
envs <- lapply(1:50, function(i) {
  simmer("Service") %>%
    add_resource("operator1", 1) %>%
    add_resource("operator2", 1) %>%
    add_resource("operator3", 1) %>%
    add_generator("call", call, function() rexp(1, 1/5)) %>%
    run(360)
})

resources <- get_mon_resources(envs)
plot(resources, metric = "utilization")
```



2.2 b) Waiting Time

Report a waiting time for each operator. Discuss your results.

Again, the first two operators are similar in wait time. The third operator's is much higher due to their higher utilization and overall call lengths.

```r
arrivals <- get_mon_arrivals(envs, per_resource = T)

waitTime = (arrivals$end_time - arrivals$start_time) - arrivals$activity_time
arrivals2 = cbind(arrivals, waitTime)
aggregate(arrivals2$waitTime, by = list(arrivals2$resource), FUN=mean)
```

```
##     Group.1        x
## 1 operator1 15.97817
## 2 operator2 18.48356
## 3 operator3 24.76400
```

2.3 c) Confidence Interval

Find a 95% confidence interval for the average waiting time for each operator.

The confidence interval of the average wait time for the third operator is wider than the other two as expected, but the difference in error is not as large as the wait times themselves

```
arrivals2 <- cbind(arrivals, waitTime)

Operator1 <- na.omit(subset(arrivals2, arrivals2$resource == "operator1"))
round(quantile(Operator1$waitTime, c(.025, .975)), 3)
```

```
##   2.5%  97.5%
##  0.000 63.282
```

```
Operator2 <- na.omit(subset(arrivals2, arrivals2$resource == "operator2"))
round(quantile(Operator2$waitTime, c(.025, .975)), 3)
```

```
##   2.5%  97.5%
##  0.000 74.303
```

```
Operator3 <- na.omit(subset(arrivals2, arrivals2$resource == "operator3"))
round(quantile(Operator3$waitTime, c(.025, .975)), 3)
```

```
##   2.5%  97.5%
##  0.000 77.333
```

3 Problem 3: Machine/Operator Queues

Consider a manufacturing system comprising two different machines and two operators:

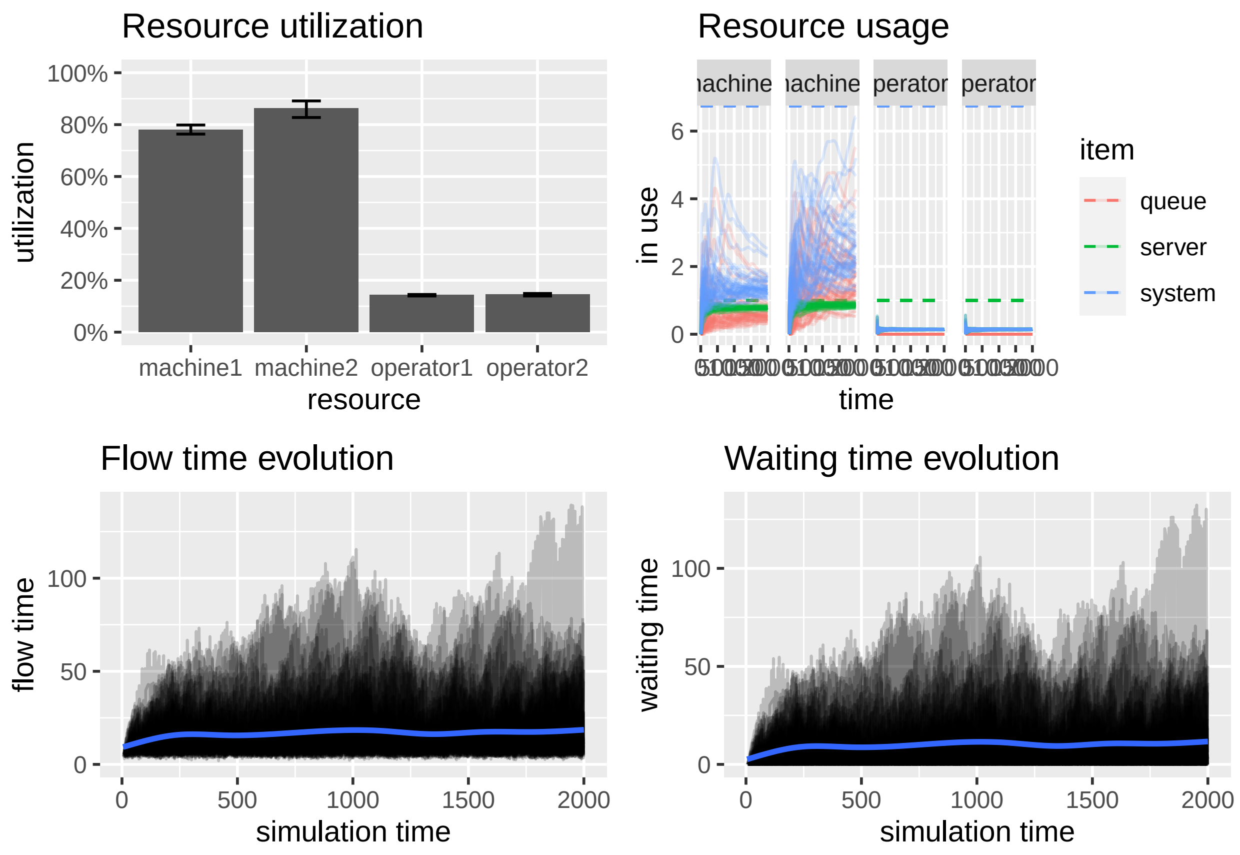
- Each operator is assigned to run a single machine.
- Parts arrive with a Truncated-Normally distributed interarrival time with a min of 0.5 minutes, mean of 4 minutes, and stdev of 2 minutes. The arriving parts are one of two types:
  - Sixty percent of the arriving parts are Type 1 and are processed on Machine 1. These parts require the assigned operator for a 1-minute setup operation.
  - The remaining 40% of the parts are Type 2 and are processed on Machine 2. These parts require the assigned operator for a 1.5-minute setup operation.
- You might have NAs in your waiting time, because some people start the service but do not finish it before the end of simulation clock.
- The service times (excluding the setup time) are Normally distributed with a mean of 4.5 minutes and a standard deviation of 1 minute for Type 1 parts and a mean of 7.5 minutes and a standard deviation of 1.5 minutes for Type 2 parts.
- Run your model for 2,000 minutes, with 50 replications.

3.1 a) Utilization, Usage, Flow Time, and Waiting Plots

Machine 1 has the advantage is setup time and services parts with a lower mean and standard deviation in their distribution relative to Machine 2. The utilization and usage plots demonstrate the difference, especially regarding queue and system times.

The flow and waiting times settle on their means over time, indicating that this process does not become "backed up" over time.

```r
library(simmer)
library(truncnorm)
library(gridExtra)
library(simmer.plot)

part <- trajectory("Part Path" ) %>%

branch(option = function() sample(1:2, 1, prob = c(0.6, 0.4), replace=T), continue = c(T),

  trajectory("A") %>%
    set_attribute("type", 1) %>%
    seize("machine1", 1) %>%
    seize("operator1", 1) %>%
    timeout(1) %>%
    release("operator1", 1) %>%
    timeout(function() rnorm(1, 4.5, 1)) %>%
    release("machine1", 1),

  trajectory("B") %>%
    set_attribute("type", 2) %>%
    seize("machine2", 1) %>%
    seize("operator2", 1) %>%
    timeout(1.5) %>%
    release("operator2", 1)%>%
    timeout(function() rnorm(1, 7.5, 1.5)) %>%
    release("machine2", 1)
)

envs <- lapply(1:50, function(i) {
  simmer("Man") %>%
    add_resource("operator1", 1) %>%
    add_resource("operator2", 1) %>%
    add_resource("machine1", 1) %>%
    add_resource("machine2", 1) %>%
    add_generator("part", part, function() rtruncnorm(1, a=0.5, mean=4, sd=2), mon = 2) %>%
    run(2000)
  })

resources = get_mon_resources(envs)
arrivals = get_mon_arrivals(envs)

p1 = plot(resources, metric = "utilization")
p2 = plot(resources, metric = "usage")
p3 = plot(arrivals, metric = "flow_time")
p4 = plot(arrivals, metric = "waiting_time")
grid.arrange(p1,p2,p3,p4)
```



3.2 b) Flow Time per Part

In this simulation, the flow times for Type 2 parts (used by Machine 2) are greater by a factor of 2.5x.

```
x1 <- get_mon_arrivals(envs)
x2 <- get_mon_attributes(envs)
merged <- merge(x1, x2, by = c("name", "replication"), all = T)

TypeA <- subset(merged, merged$value == 1)
TypeB <- subset(merged, merged$value == 2)

typeA.flowTime = (TypeA$end_time-TypeA$start_time)
cat("Flow time for Type 1 parts:", mean(typeA.flowTime, na.rm = T), "\n")
```

```
## Flow time for Type 1 parts: 9.969793
```

```
typeB.flowTime = (TypeB$end_time-TypeB$start_time)
cat("Flow time for Type 2 parts:", mean(typeB.flowTime, na.rm = T))
```

```
## Flow time for Type 2 parts: 26.55248
```

4 Problem 4: Leaving, Balking, and Reneging

Consider an Urgent Care (UC) comprising three doctors and one nurse to serve patients during a day.

- On a typical day, the interarrival time is exponentially distributed with a mean of 6 minutes.
- 25% of patients are high-priority, and the remaining are low-priority.
- Upon arrival into UC, the patients are triaged by a nurse into one of the two types of patients. The service time for triage is distributed by Triangular distribution with min = 3, max = 10, and the most likely value = 5 minutes.
  - Then, the patients wait in the waiting room and get called to visit doctors on a first-come-first-served basis. If more than 12 people are waiting for service, an arriving patient will exit before being triaged.
- Finally, low priority patients may depart if they have to wait longer than 20±5 minutes (Uniformly distributed) after triage.
- The doctor priority service time distributions are given as follows (in minutes):
- High Normal(mean = 40, sd = 5)
- Low Gamma(shape = 15, rate = 1)

Assuming that the UC opens for 8 hours, simulate the process for 50 replications.

4.1 a) Average Flow Time per Patient Type

In this simulation, the average flow times for low and high priority patients are very close at 53.1 and 54.5 minutes respectively. The slightly smaller flow time for low priority patients is likely explained by their renege rate which is not a factor for high priority patients.

```r
library(simmer)
library(triangle)

set.seed(12)

envs <- lapply(1:50, function(i) {
  env <- simmer("Clinic") %>%
    add_resource("nurse", 1) %>%
    add_resource("doctor", 3)

  patient <- trajectory("Patient Path") %>%

    branch(function()
      sample(1:2, size=1, replace=TRUE, prob=c(0.25,0.75)), continue=c(T,T),

      # High Priority
      trajectory("High Priority") %>%
        set_attribute("priority", 2) %>%
        set_prioritization(c(5, 7, T)) %>%

        leave(prob = function()
          ifelse((get_queue_count(env, "nurse")+ get_queue_count(env, "doctor"))>12, 1, 0)) %>%
        seize("nurse", 1) %>%
        timeout(function() rtriangle(1, 3, 10, 5)) %>%
        release("nurse", 1) %>%
        seize("doctor", 1) %>%
        timeout(function() rnorm(1, 40, 5)) %>%
        release("doctor", 1),

      # Low Priority
      trajectory("Low Priority") %>%
        set_attribute("priority", 1) %>%
        set_prioritization(c(3, 7, T)) %>%

        leave(prob = function()
          ifelse((get_queue_count(env, "nurse")+ get_queue_count(env, "doctor"))>12, 1, 0)) %>%
        seize("nurse", 1) %>%
        timeout(function() rtriangle(1, 3, 10, 5)) %>%
        release("nurse", 1) %>%

        # Renege for wait time 15-25
        renege_in(function() runif(1, 15, 25), out = trajectory()) %>%
        seize("doctor", 1) %>%
        renege_abort() %>%
        timeout(function() rgamma(1, 15, 1)) %>%
        release("doctor", 1)
    )
  env %>%
    add_generator("patient", patient, function() rexp(1, 1/6), mon = 2)

  env %>%
    run(480)
})
```

```r
x1 <- get_mon_arrivals(envs)
x2 <- get_mon_attributes(envs)
all <- merge(x1, x2, by=c("name", "replication"), all = T)
all <- na.omit(all)

# Low Priority
Type1 <- subset(all, all$value == 1)
type1.flowTime = (Type1$end_time-Type1$start_time)

# High Priority
Type2 <- subset(all, all$value == 2)
type2.flowTime = (Type2$end_time-Type2$start_time)

cat("Low Priority Average Flow Time: ", mean(type1.flowTime, na.rm = T), "\n")
```

```
## Low Priority Average Flow Time:  53.14423
```

```r
cat("High Priority Average Flow Time: ", mean(type2.flowTime, na.rm = T))
```

```
## High Priority Average Flow Time:  54.50496
```

4.2 (b) Patient Balk Probability

Balking occurs when a patient refuses to join a queue that is too long, meaning in this scenario it is evaluated prior to triage.

We are told that patients of all priorities will balk if the queue is greater than 12, and in our simulation the probability of low-priority patients balking is 2.4%.

```r
cat("Probability of low-priority patient balk: ", mean(all$activity_time==0 & all$value==1))
```

```
## Probability of low-priority patient balk:  0.02412716
```