# HW1 Solution

Daniel Moore

March 12, 2024

## Table of contents



## 1 Purpose

The purpose of this solution is to demonstrate how Quarto should be used to prepare future homework solutions and reports. Additionally, this will show how simply these problems can be solved using `simmer`.

## 2 Question 3

Suppose that customers arrive in a bank at a Poisson rate of one per every 15 minutes, and that the service time is exponential at a rate of of one service per 12 minutes. WE assume that there is no limit on the number of customers that could be in the system at the same time.

Develop a simulation model using R and run the simmulation for durations of 2 hours and 24 hours. Estimate the following statistics:

a) The average number of customers in the system.
b) The averagge amount of time a customer spends in the system
c) Ceate an animation in R to illustrate the average number of customers in the system over the simulated 2-hours (120 mintutes) period

## 2.1 Approach

1. Load Packages
2. Model the system in `simmer`
3. Run the simulation
4. Extract required statistics

## 2.2 Loading Packages

```
# for plots, data manipulation, and output
library(tidyverse)
library(knitr)

# simulation package
library(simmer)
library(simmer.plot)
library(simmer.bricks)

# packages for making animations
library(gganimate)
library(gifski)
```

## 2.3 Building `simmer` Model

First we will define the parameters in a way that is easy and clear to be referenced later. Then we need to define the `trajectory`, the `resources`, and the `simmer` environment.

### 2.3.1 Bank Parameters

```r
bank_sim_time <- c(2*60, 24*60)

lambda_bank <- 1/15

arrival_bank <- function(n=1) {
  rexp(n = n, rate = lambda_bank)
}

mu_bank <- 1/12

service_bank <- function(n=1) {
  rexp(n = n, rate = mu_bank)
}
```

### 2.3.2 Customer Trajectory

```r
customer <- trajectory("customer") |>
  visit("teller", service_bank)
```

### 2.3.3 Bank Environment

```r
bank <- simmer("bank") |>
  add_resource("teller") |>
  add_generator("customer", customer, arrival_bank)
```

### 2.3.4 Run Simulation & Save Results

```r
reset(bank) |> run(bank_sim_time[1])
bank_resources_2 <- get_mon_resources(bank)
bank_arrivals_2 <- get_mon_arrivals(bank)
```

```r
reset(bank) |> run(bank_sim_time[2])
```

```
simmer environment: bank | now: 1440 | next: 1440.52490383319
{ Monitor: in memory }
```

```
{ Resource: teller | monitored: TRUE | server status: 1(1) | queue status: 12(Inf) }
{ Source: customer | monitored: 1 | n_generated: 123 }
```

```
bank_resources_24 <- get_mon_resources(bank)
bank_arrivals_24 <- get_mon_arrivals(bank)
```

### 2.3.5 Extract Statistics

```
resource_weighted_avg <- function(resource_data, item, sim_time) {
  resource_data <- resource_data |>
    arrange(time) |>
    mutate(time_diff = lead(time, default = sim_time) - time)

  return(sum(resource_data[[item]] * resource_data$time_diff) / sim_time)
}
```

```
L_bank_2hr <- resource_weighted_avg(
  bank_resources_2, "system", bank_sim_time[1])

L_bank_24hr <- resource_weighted_avg(
  bank_resources_2, "system", bank_sim_time[2])
```

```
W_bank_2hr <- mean(bank_arrivals_2$end_time - bank_arrivals_2$start_time)

W_bank_24hr <- mean(bank_arrivals_24$end_time - bank_arrivals_24$start_time)
```

The results are summarized in Table 1

```
bank_summary_df <- tibble(
  "Simulation Time (min)" = bank_sim_time,

  "Avg Customers in System, L" = round(
    c(L_bank_2hr, L_bank_24hr), digits = 2),

  "Avg Time in System, W (min)" = round(
    c(W_bank_2hr, W_bank_24hr), digits = 2)
)

kable(bank_summary_df)
```

Table 1: Bank Simulation Summary Table

| Simulation Time (min) | Avg Customers in System, L | Avg Time in System, W (min) |
|---|---|---|
| 120 | 0.82 | 18.94 |
| 1440 | 0.98 | 123.09 |

### 2.3.6 Plot

See Figure 1 and Figure 3 for static plots.

```
plot(bank_resources_2, items = "system") +
  labs(
    x = "time (min)",
    y = "customers",
    title = "average over time"
  )
```

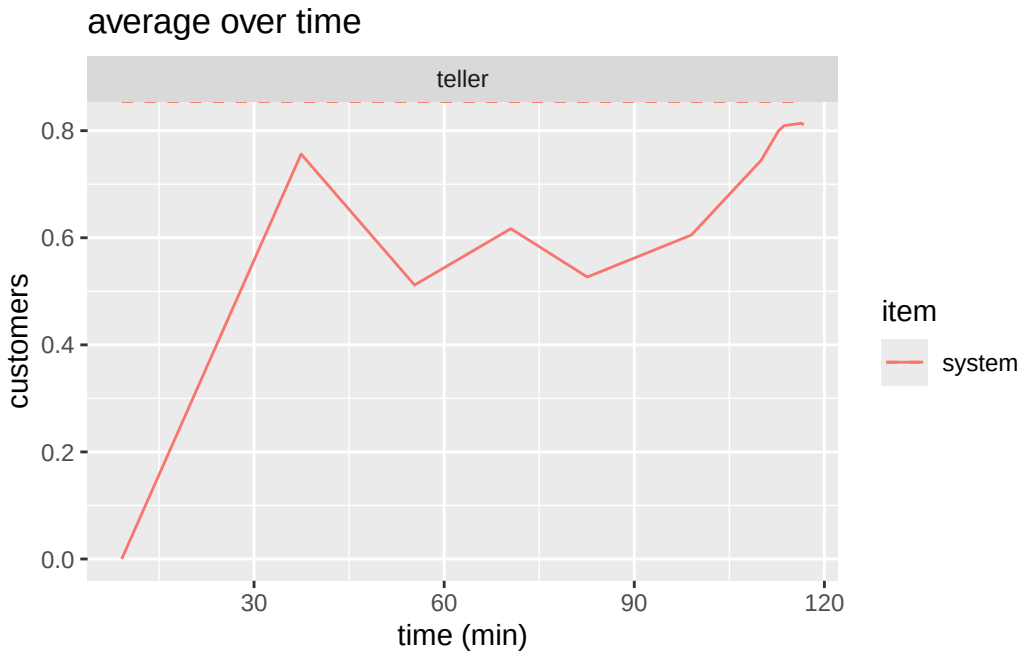


Figure 1: Bank System (2 hr)

```
plot(bank_resources_2, items = "system", steps = TRUE) +
  labs(
```

```
    x = "time (min)",
    y = "customers",
    title = "instantaneous"
  )
```

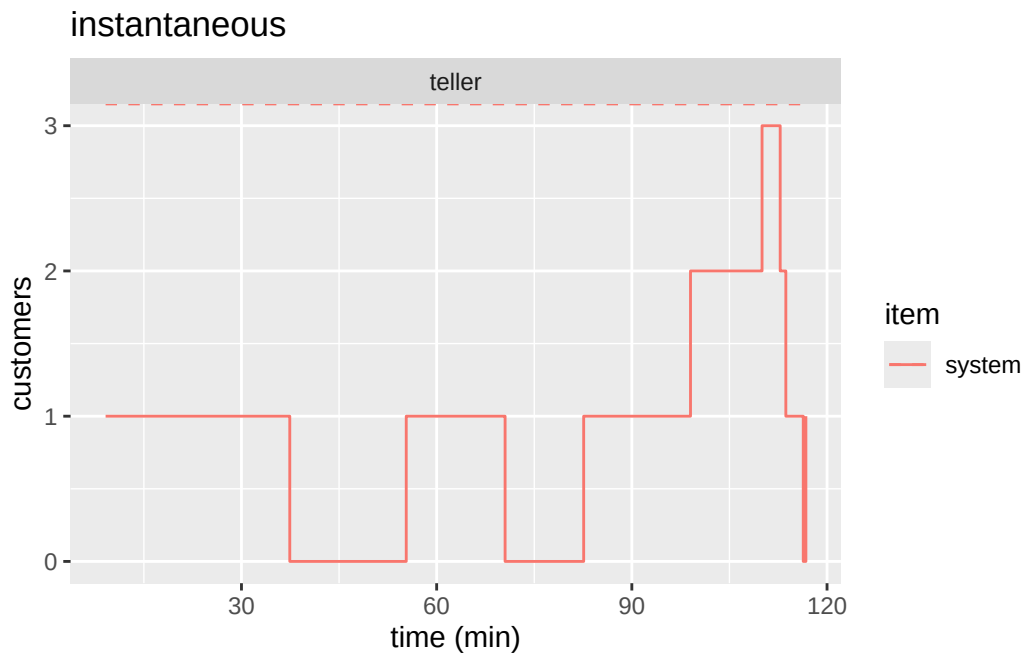


Figure 2: Bank System (2 hr)

```
plot(bank_resources_24, items = "system") +
  labs(
    x = "time (min)",
    y = "customers",
    title = "average over time"
  )
```

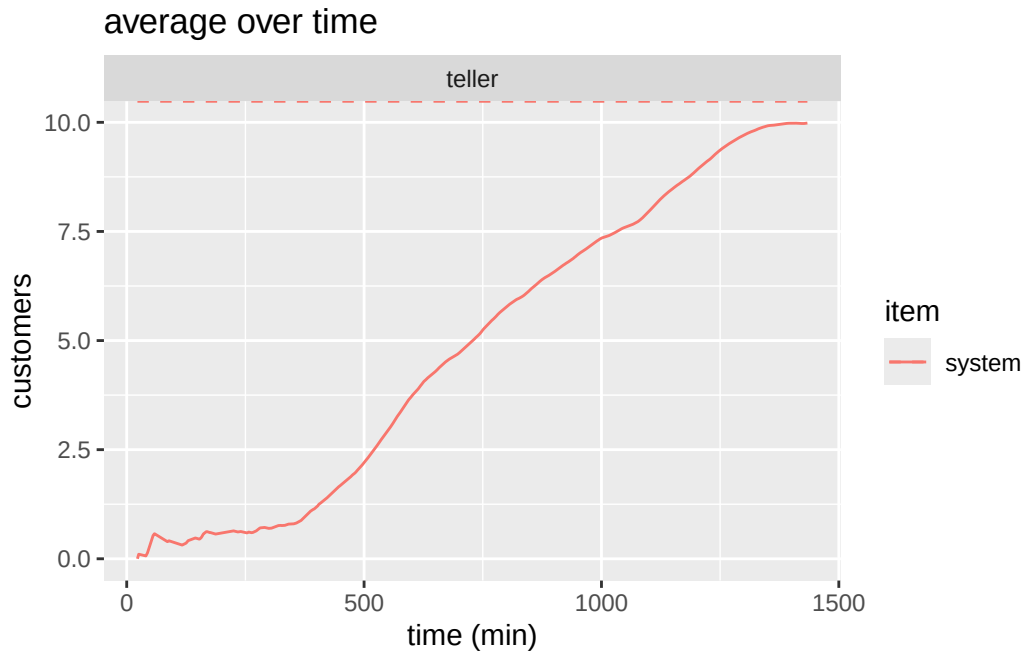


Figure 3: Bank System (24 hr)

```
plot(bank_resources_24, items = "system", steps = TRUE) +
  labs(
    x = "time (min)",
    y = "customers",
    title = "instantaneous"
  )
```

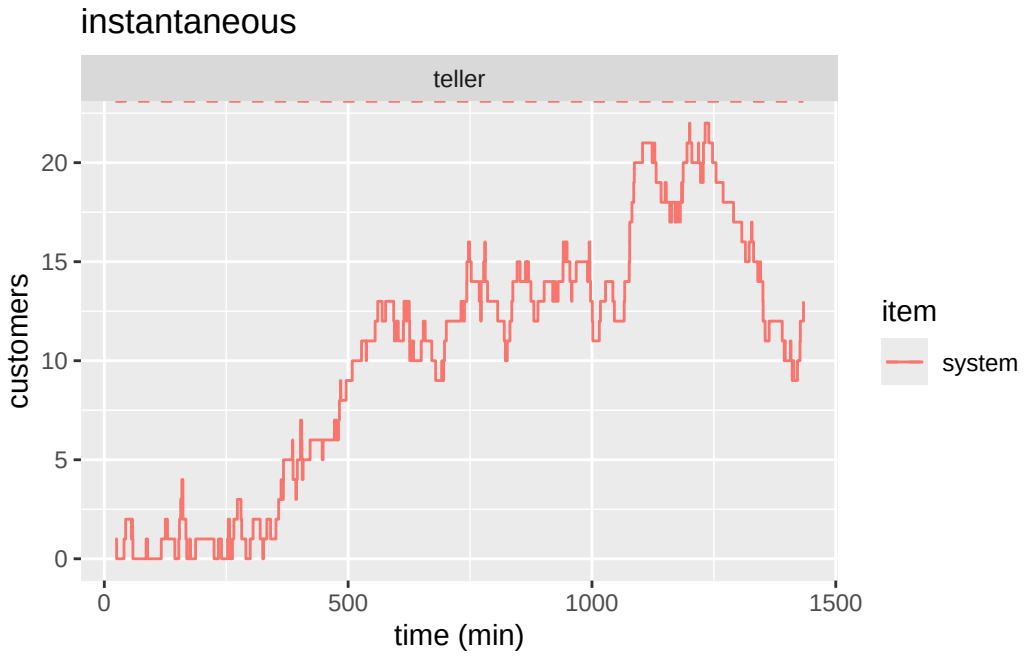


Figure 4: Bank System (24 hr)

### 2.3.7 Animation

We are tasked with animating the *average* number of customers in the system. This means we do not want the step-plot which gives the instantaneous number of people in the system. We have already generated this static plot in Figure 1. `simmer.plot` already handled getting the time-weighted average. All we need to do is animate it which is shown and explained in the code below.

> **Note**
>
> As I've pointed out before, you cannot insert a ".gif" into a pdf. You need to set the cell `eval` option to `false` as shown below. Run the cell by clicking the "Run Current Chunk" in `RStudio`.

```
```{r, animate-2hr-average}
#| eval: false
#| warning: false

p <- plot(bank_resources_2, "teller", metric="usage", item="system") + # <1>
    labs( # <2>
      x="Time (min)",
```
```

```
      y="Number Customers"
    ) +
  transition_reveal(bank_resources_2$time) # <3>

anim_save("bank_2hr_anim.gif", p) # <4>
```

① Assign our plot to a variable
② Adjust x and y label
③ Specify what part of the plot should be animated
④ Save the animation.