

Case Study: Emergency Room (DES)

Coco Choo

1. Business Problem Statement

The Emergency Room (ER) experiences a major challenge in having a high patient waiting time. This can potentially lead to longer flow times, patient dissatisfaction and inefficient resource utilization. Our goal of this study is to optimize the process to reduce the patient waiting times while maintaining an efficient doctor utilization rate and overall patient flow. A Discrete Event Simulation (DES) is performed for 20 continuous days to model the ER system to test various optimization strategies to improve service flow.

2. Decision Variables and Key Parameters

2.1 Decision Variables:

1. Doctor Capacity - Number of doctors available
2. Patient Prioritization – Various factors decide the prioritization of patients
3. Queue Structure – Single & shared queues for different types of patients
4. Treatment Time – Patient condition and treatment stage determines the treatment time

2.2 Key Parameters

1. Doctor – 2 doctors are available interchangeably
2. Patient Arrival Rate – The frequency of patient arrivals
3. Service Time Range
 - a. NIA (Need Immediate Attention) - 40 ± 30 min (initial), 30 ± 20 min (final)
 - b. CW (Can Wait) - 15 ± 10 min (initial), 10 ± 5 min (final)
4. Priority Level
 - a. NIA – Priority 3 (Highest) -> Priority 2 (Mid)
 - b. CW – Priority 1 (Lowest) -> Priority 2 (Mid)

3. Mathematical / Graphical Model for Solution

3.1 Mathematical Formulation

3.1.1 Patient Arrival Distribution

Given the unknown probability distribution of Emergency Room arrivals, we conducted input modeling to identify the distribution that best represents the data. Assuming the data is discrete, we began by creating three visualizations to gain initial insights.

```

library(simmer)
data <- read.csv("C:\\Coco\\TTU\\Spring 2025\\Spring I\\Simulation & Optimization\\Group
Application\\Case6_emergency-room.csv")

InterArrival = data$interArrival

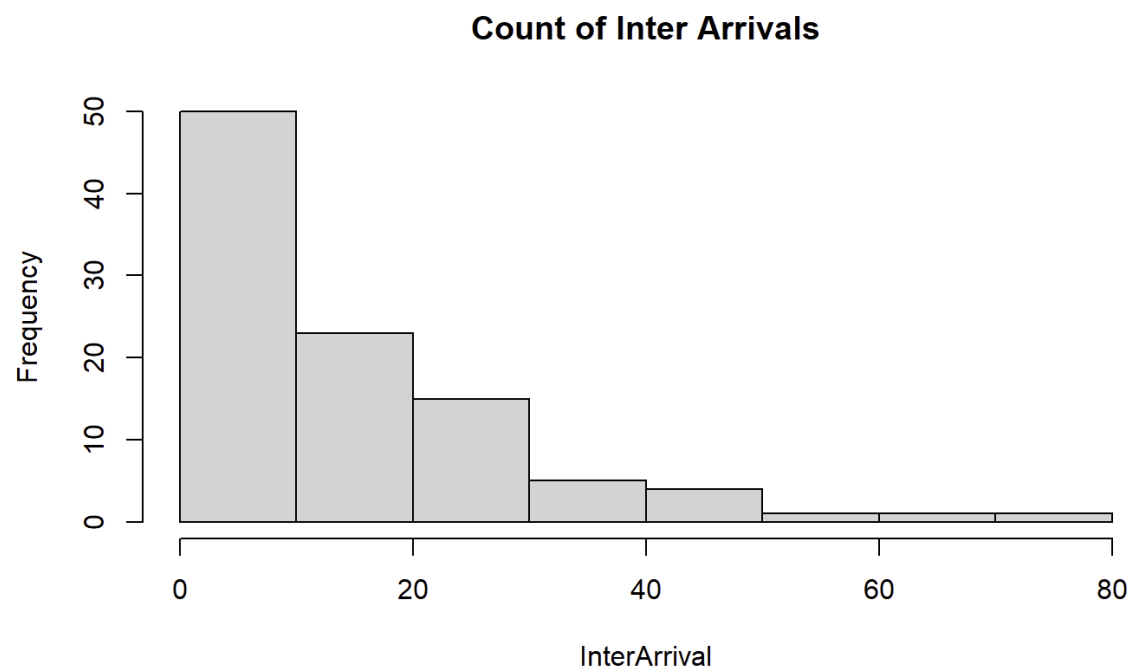
# Data is discrete
# Visualize data
hist(InterArrival, main="Count of Inter Arrivals")
# The data distribution is right skewed and unimodal

# Time Series Plot
plot(InterArrival, type="b", main="Count of Inter Arrivals", ylab = "Count")
# There is no noticeable trend in the time-series plot

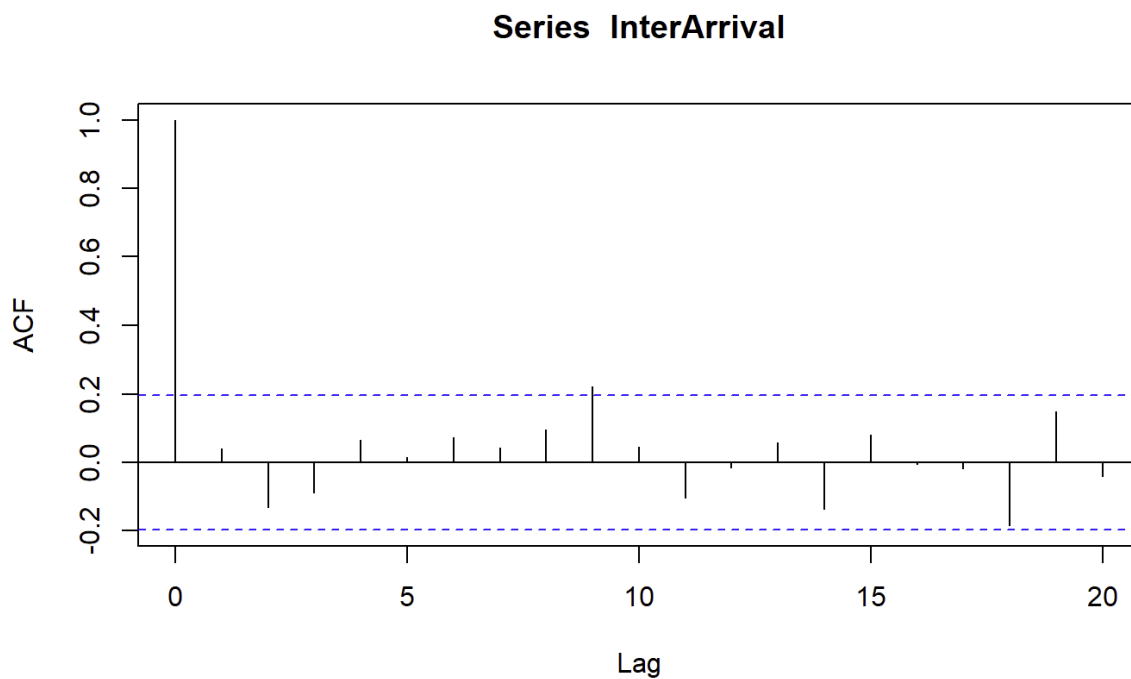
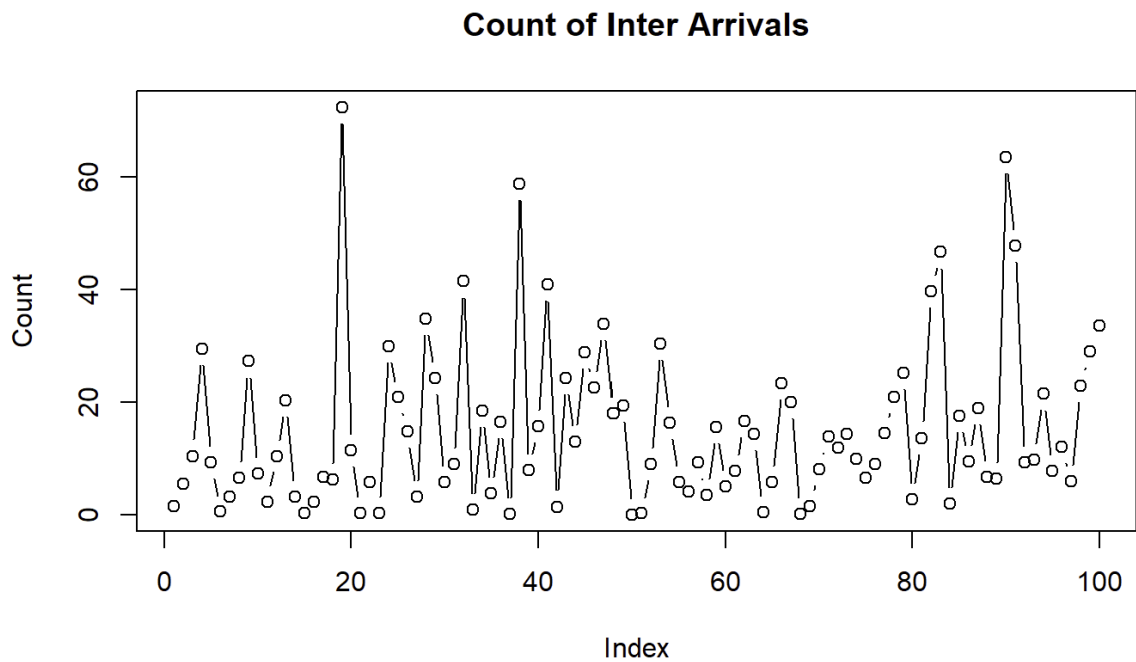
# Autocorrelation Plot
acf(InterArrival)
# The autocorrelation plot shows no significant correlation for the observation number (all lags are
well within the confidence band except for one instance, without following a pattern). Therefore, the
data appears to be stationary (IID).

```

The histogram revealed that the data distribution is right-skewed and unimodal. To assess whether the data follows an identically and independently distributed (IID) process, we analyzed the time series and autocorrelation plots.



The time series plot indicated no discernible trend or pattern, while the autocorrelation plot showed no significant correlation across observations, with all lags falling within the confidence band. Based on these observations, the data appears to be stationary.



Next, we hypothesized two potential distributions: Negative binomial and Poisson. By comparing the log-likelihood values, AIC and BIC, we determined that the Negative Binomial distribution provided a better fit. However, goodness-of-fit testing yielded a p-value of approximately $8.42e^{-7}$, which is below the 0.05 significance threshold. This result indicates that the data does not follow Negative Binomial distribution, suggesting the need to explore continuous distributions.

```

# Hypothesizing distributions
library(fitdistrplus)
descdist(InterArrival, discrete = TRUE)

# We first evaluate Negative Binomial
fit.nbinom = fitdist(InterArrival, "nbinom")
summary(fit.nbinom)

# Then, we evaluate Poisson
fit.pois = fitdist(InterArrival, "pois")
summary(fit.pois)
# nbinom has bigger loglikelihood value. Based on AIC and BIC, negative binomial is better

```

For continuous distribution, we tested Gamma, Weibull and Normal distributions. Based on the log-likelihood values, AIC and BIC, the Gamma and Weibull distributions provided the best fit. Goodness-of-fit testing of the Gamma distribution resulted in a p-value of approximately 0.17, exceeding the 0.05 threshold. This confirms that the inter-arrival times are well-modeled by a Gamma distribution. The estimated shape and rate parameters for the Gamma distribution are as follows: shape = 0.9638 and rate = 0.0639.

```

# If we assume data is continuous
descdist(InterArrival, discrete = FALSE)

# Gamma
fit.gamma <- fitdist(InterArrival, "gamma")
summary(fit.gamma)

# Weibull
fit.weibull <- fitdist(InterArrival, "weibull")
summary(fit.weibull)

# Normal
fit.norm <- fitdist(InterArrival, "norm")
summary(fit.norm)
# Gamma and Weibull have the best likelihood and AIC & BIC

# Plot
plot(fit.gamma)
plot(fit.weibull)
# Will check Gamma distribution

##{r}
summary(fit.gamma)
# shape = 0.9638
# rate = 0.0639

```

3.1.2 Patient Type Distribution

Before developing the discrete event simulation model, we calculated the distribution of patient types, which revealed the following proportions:

- CW patients: 82%
- NIA patients: 18%

```

##{r}
# Goodness of fit on Gamma distribution
g <- gofstat(fit.gamma)
g$chisqpvalue

# H0: Data is distributed by Gamma distribution
# Test is NOT rejected because p-value > 0.05
# InterArrival is distributed by Gamma distribution

```

```
[1] 0.170187
```

```

```{r}
Calculate probability distribution of each type
count <- table(data$type)
prop.table(count)
CW = 0.82
NIA = 0.18
```

```

3.2 ER Simulation and Performance Analysis

To evaluate system performance, we simulated ER operations with two doctors and then tested the impact of increasing the number of doctors to seven. The model was run 20 times for each scenario to ensure statistical robustness.

```

```{r}
patient <- trajectory("patients' path") %>%

branch(option = function() sample(1:2, 1, prob = c(0.82, 0.18), replace=T), continue = c(T,T),

Set up a trajectory for NIA patients
trajectory("NIA Patient") %>%
Highest Priority: 3
set_attribute("priority", 3) %>%
set_prioritization(c(5, 7, T)) %>%
seize("doctor", 1) %>%
timeout(function() runif(1, 10, 70)) %>%
release("doctor", 1) %>%

Mid Priority: 2
set_attribute("priority", 2) %>%
set_prioritization(c(4, 7, T)) %>%
seize("doctor", 1) %>%
timeout(function() runif(1, 10, 50)) %>%
release("doctor", 1),

Set up a trajectory for CW patients
trajectory("CW Patient") %>%
Lowest Priority: 1
set_attribute("priority", 1) %>%
set_prioritization(c(2, 7, T)) %>%
seize("doctor", 1) %>%
timeout(function() runif(1, 5, 25)) %>%
release("doctor", 1) %>%

Mid Priority: 2
set_attribute("priority", 2) %>%
set_prioritization(c(4, 7, T)) %>%
seize("doctor", 1) %>%
timeout(function() runif(1, 5, 15)) %>%
release("doctor", 1)
)

set.seed(123)
envs <- lapply(1:20, function(i) {
 env <- simmer("Emergency Room")
 env %>%
 add_resource("doctor", 2) %>%
 add_generator("patient", patient, function() rgamma(1, shape = 0.9638, rate = 0.0639), mon = 2) %>%
 run(24*60)
})
```

```

3.2.1 Impact on Average Waiting Time

The average waiting times for CW and NIA patients in both scenarios are summarized below:

| Scenario | CW Patients | NIA Patients |
|-----------------------|--------------|--------------|
| With 2 Doctors | 12.1 minutes | 292 minutes |
| With 7 Doctors | 4.03 minutes | 2.28 minutes |

The results indicate that with only two doctors, NIA patients experience significantly longer wait times despite being higher-priority cases. However, after increasing the number of doctors to seven, waiting times dropped significantly for both patient

categories.

```
## {r}
x1 <- get_mon_arrivals(envs)
x2 <- get_mon_attributes(envs)

all <- merge(x1, x2, by= c("name", "replication"), all= T)
head(all)

priority1 <- na.omit(subset(all, all$value ==1 ))
priority3 <- na.omit(subset(all, all$value ==3))

# Compare average waiting time between NIA and CW patients
priority1.waiting <- (priority1$end_time - priority1$start_time) - priority1$activity_time

priority3.waiting <- (priority3$end_time - priority3$start_time) - priority3$activity_time

# Average waiting time for CW patients
mean(priority1.waiting)
## Before suggestion: 12.1 minutes

# Average waiting time for NIA patients
mean(priority3.waiting)
## Before suggestion: 292 minutes

# Average waiting time per replication for each type
priority1.waiting.rep <- aggregate(priority1.waiting, by = list(priority1$replication), mean)

priority3.waiting.rep <- aggregate(priority3.waiting, by = list(priority3$replication), mean)

## {r}
x1 <- get_mon_arrivals(envs)
x2 <- get_mon_attributes(envs)

all <- merge(x1, x2, by= c("name", "replication"), all= T)
head(all)

priority1 <- na.omit(subset(all, all$value ==1 ))
priority3 <- na.omit(subset(all, all$value ==3))

# Compare average waiting time between NIA and CW patients
priority1.waiting <- (priority1$end_time - priority1$start_time) - priority1$activity_time

priority3.waiting <- (priority3$end_time - priority3$start_time) - priority3$activity_time

# Average waiting time for CW patients
mean(priority1.waiting)
## After suggestion (Add more doctors: 7): 4.03 minutes

# Average waiting time for NIA patients
mean(priority3.waiting)
## After suggestion (Add more doctors: 7): 2.28 minutes

# Average waiting time per replication for each type
priority1.waiting.rep <- aggregate(priority1.waiting, by = list(priority1$replication), mean)

priority3.waiting.rep <- aggregate(priority3.waiting, by = list(priority3$replication), mean)

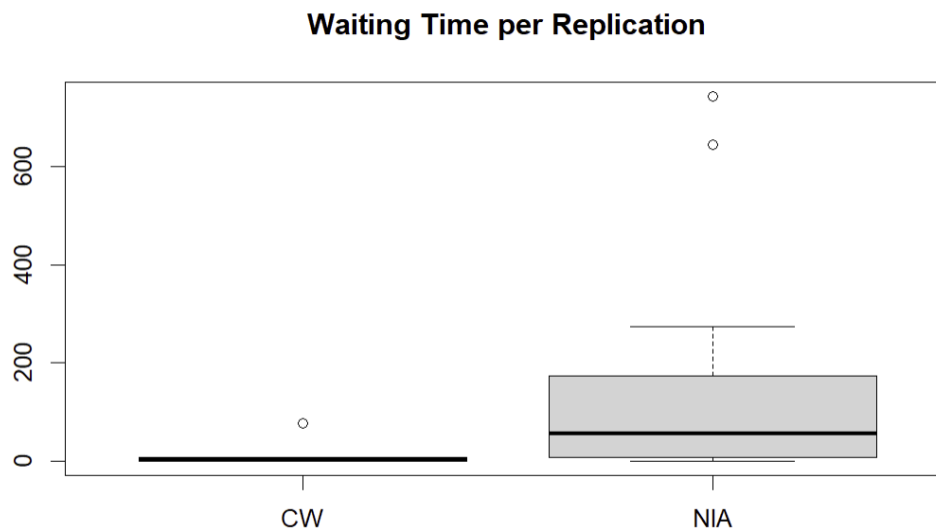
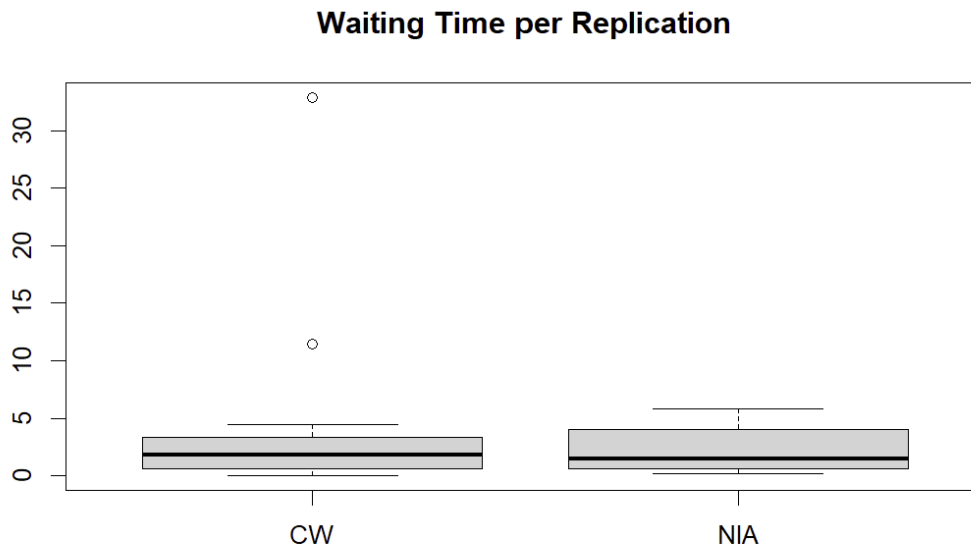
boxplot(priority1.waiting.rep$x, priority3.waiting.rep$x, names = c("CW", "NIA"), main = "Waiting Time per Replication")
```

Boxplot Analysis of Waiting Times

To further assess the impact of increasing the number of doctors, boxplots were generated to compare the distribution of waiting times for CW and NIA patients before and after adjustments.

- Before increasing the number of doctors, the NIA waiting time boxplot showed a wide spread, with several extreme outliers, indicating high variance and inefficiency in patient handling.
- After increasing the number of doctors to seven, the waiting time distribution became more compact, with significantly reduced outliers for NIA patients, demonstrating improved system efficiency and better prioritization of high-urgency cases.

The median waiting time for NIA patients decreased drastically, aligning with the expected priority-based queuing principles.



3.2.2 Impact on Average Flow Time

In addition to waiting times, we examined the average flow time - the total duration a patients spends in the system, including both service time and waiting time. The following table compares the results:

| Scenario | CW Patients | NIA Patients |
|-----------------------|--------------|--------------|
| With 2 Doctors | 35 minutes | 357 minutes |
| With 7 Doctors | 29.2 minutes | 72.1 minutes |

The increase in doctor availability led to a drastic reduction in patient flow time, especially for NIA patients. This ensures that high-priority patients receive timely care, improving overall system efficiency.

```

{r}
# Average flow time of each type of patient (BEFORE suggestion)
priority1.flowtime = (priority1$end_time-priority1$start_time)

priority3.flowtime = (priority3$end_time-priority3$start_time)

# Average flow time of CW Patient
mean(priority1.flowtime, na.rm = T)
## [1] 35

# Average flow time of NIA Patient
mean(priority3.flowtime, na.rm = T)
## [1] 357

```

```

{r}
# Average flow time of each type of patient (AFTER suggestion)
priority1.flowtime = (priority1$end_time-priority1$start_time)

priority3.flowtime = (priority3$end_time-priority3$start_time)

# Average flow time of CW Patient
mean(priority1.flowtime, na.rm = T)
## [1] 29.2

# Average flow time of NIA Patient
mean(priority3.flowtime, na.rm = T)
## [1] 72.1

```

3.2.3 Resource Utilization and Optimization

Boxplot Analysis of Doctor Utilization

Boxplots were used to visualize doctor utilization rates before and after increasing staffing levels:

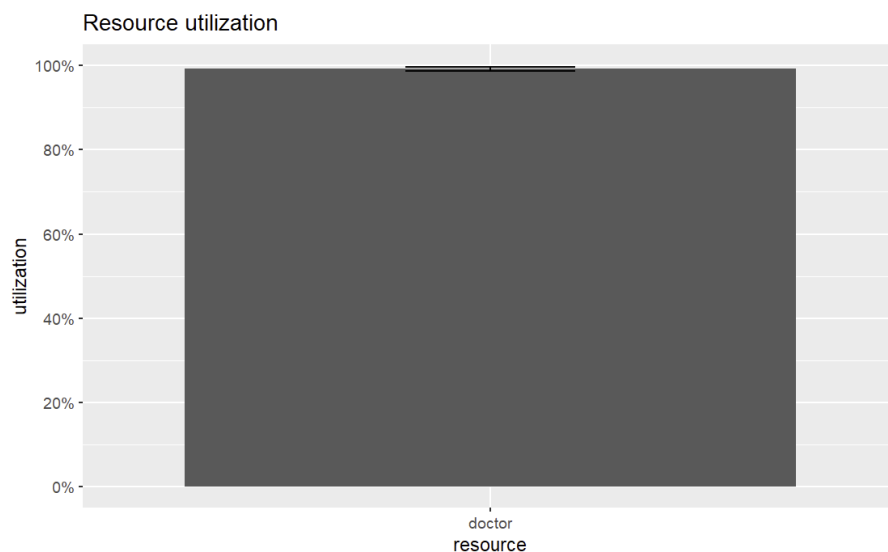
- With two doctors, utilization rates were extremely high, with some periods showing near 100% utilization, leading to burnout and inefficiency.
- With seven doctors, the boxplot distribution became more balanced, showing lower variance in workload and a more sustainable utilization pattern.

This suggests that increasing the number of doctors not only reduced patient wait times but also balanced workload distribution, preventing excessive doctor fatigue.

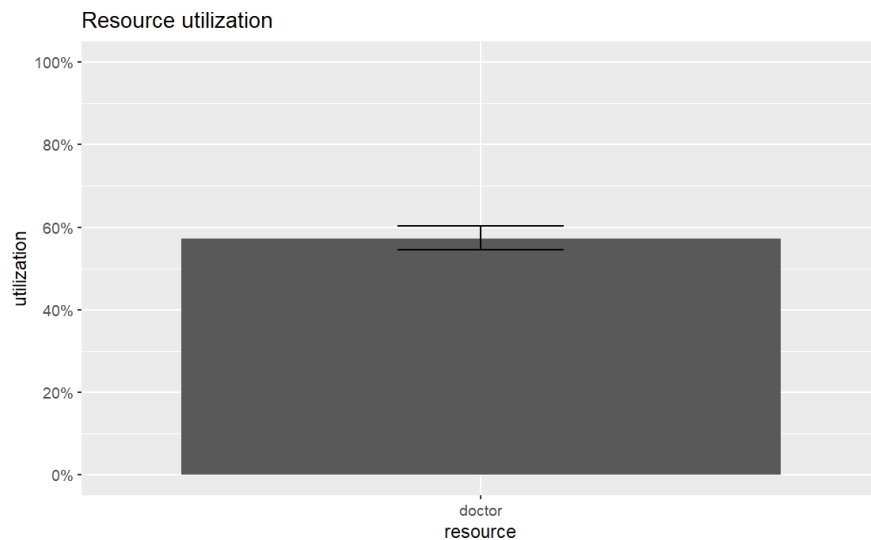
```

{r}
# Utilization of doctors BEFORE suggestion
library(simmer.plot)
options(digits = 3)
resources <- get_mon_resources(envs)
plot(resources, metric = "utilization")

```




```
{r}
# Utilization of doctors AFTER suggestion
resources <- get_mon_resources(envs)
plot(resources, metric = "utilization")
```



The results show that:

- With the two doctors, resource allocation was inefficient, resulting in excessive patient wait times and overburdened doctors.
- With seven doctors, approximately 60% of the time, all doctors were actively engaged, indicating an optimal balance between resource availability and patient demand.

4. Conclusion

The simulation results demonstrate that increasing the number of doctors from two to seven significantly improves efficiency by:

- Reducing patient waiting times, particularly for high-priority (NIA) patients.
- Lowering total flow time, ensuring faster patient processing.
- Optimizing doctor utilization, avoiding excessive idle time while maintaining adequate staffing levels.

Further improvements can be achieved by:

- Implementing dynamic doctor allocation strategies to adjust staffing based on real-time demand.
- Refining patient prioritization criteria to ensure that high-urgency cases receive immediate attention.
- Redistributing workloads among available doctors to enhance operational efficiency.

By adopting these measures, the ER can sustain long-term improvements in patient care, resource efficiency, and overall healthcare outcomes.