Queueing Project: Dunkin' Donut Drive-through

Olusegun

2022-04-10

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
Project <- read.csv("/Users/shegeb/Documents/MY PROJECTS/probability_modeing_for_decision_making_bgsu/Du
Dukin <-Project[-c(55,56),-c(1,5:8)]
```

```
Dukin$Arrival.Time = as.numeric(as.duration(hms(Dukin$Arrival.Time)), "minutes")
Dukin$Entrance.into.Service = as.numeric(as.duration(hms(Dukin$Entrance.into.Service)), "minutes")
Dukin$Departure.Time = as.numeric(as.duration(hms(Dukin$Departure.Time)), "minutes")

Dukin$Queueing.Time = Dukin$Entrance.into.Service - Dukin$Arrival.Time
Dukin$Service.Time = Dukin$Departure.Time - Dukin$Entrance.into.Service
Dukin$Waiting.Time = Dukin$Departure.Time - Dukin$Arrival.Time
Dukin$Interarrival.Time = Dukin$Arrival.Time - lag(Dukin$Arrival.Time)
```

```
means <- data.frame(Mean = c(diff(range(Dukin$Arrival.Time))/54, apply(Dukin[, 4:6], 2, mean, na.rm = TR
  mutate(Rate = 1 / Mean)
```

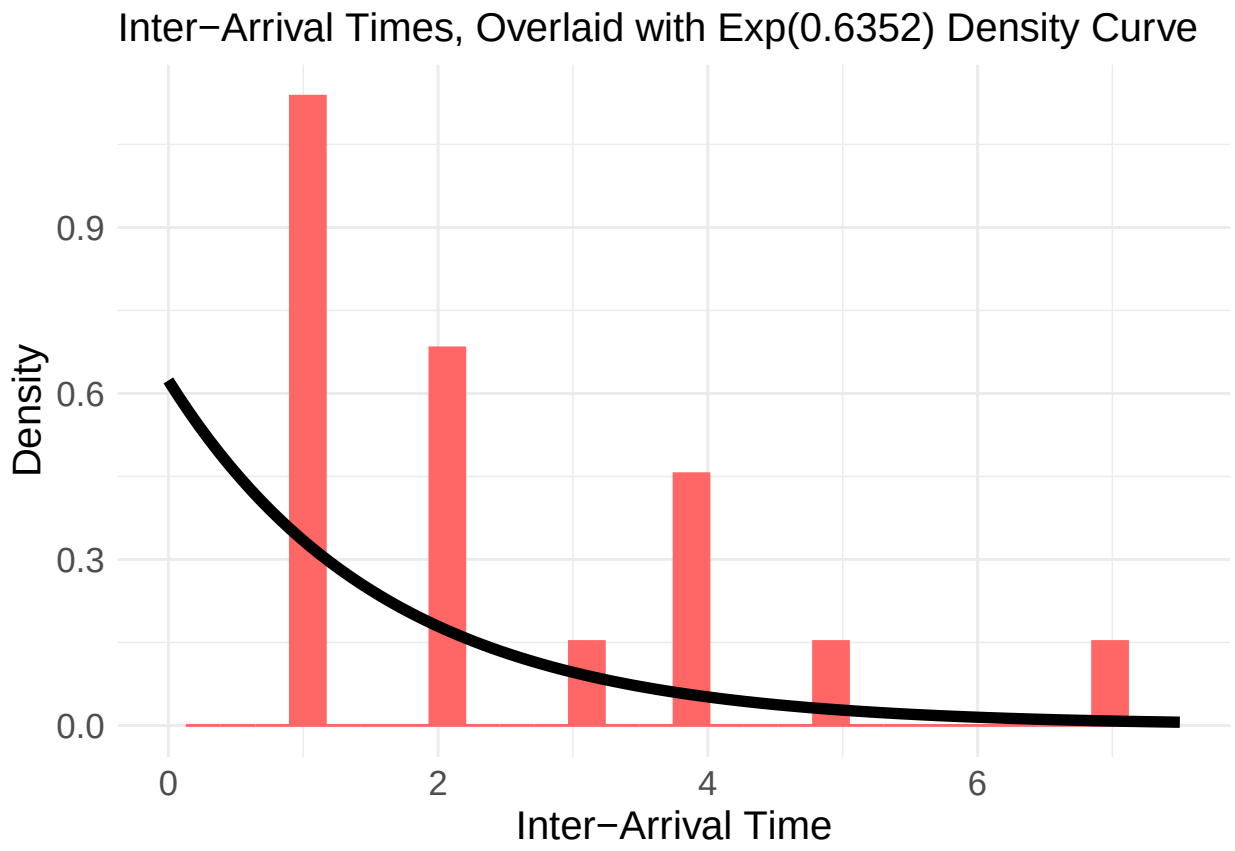
```r
rownames(means) <- c("Inter-arrival Time",
                     "Queueing Time",
                     "Service Time",
                     "Waiting Time")
colnames(means) = c("Mean (Mins)", "Rate")
means
```

```
##                     Mean (Mins)      Rate
## Inter-arrival Time    1.574074 0.6352941
## Queueing Time         1.851852 0.5400000
## Service Time          5.185185 0.1928571
## Waiting Time          7.037037 0.1421053
```

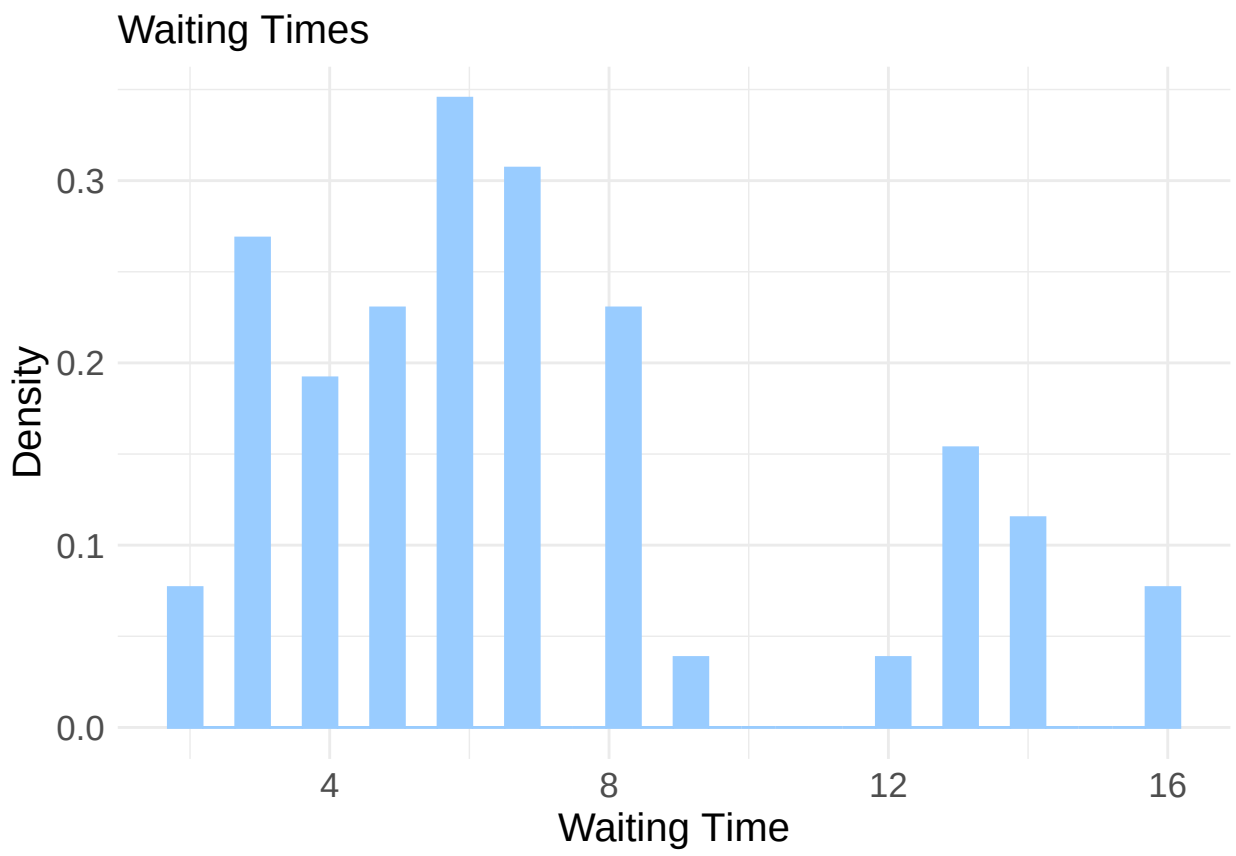
```r
Dukin[-1, ] %>%
  ggplot(aes(Interarrival.Time)) +
  geom_histogram(aes(y = after_stat(density)), bins = 30, fill = "#ff6666", color = "#ff6666") +
  xlim(0,7.5) +
  stat_function(
    fun = dexp,
    args = list(rate = 1 / mean(Dukin$Interarrival.Time, na.rm = TRUE)),
    lwd = 2) +
  theme_minimal() +
  theme(axis.title.x = element_text(size = 15),
        axis.text.x = element_text(size = 12.5),
        axis.title.y = element_text(size = 15),
        axis.text.y = element_text(size = 12.5),
        plot.title = element_text(size = 15)) +
  labs(title = "Inter-Arrival Times, Overlaid with Exp(0.6352) Density Curve",
       x = "Inter-Arrival Time",
       y = "Density")
```

```
## Warning: Removed 2 rows containing non-finite values (`stat_bin()`).
```

```
## Warning: Removed 2 rows containing missing values (`geom_bar()`).
```

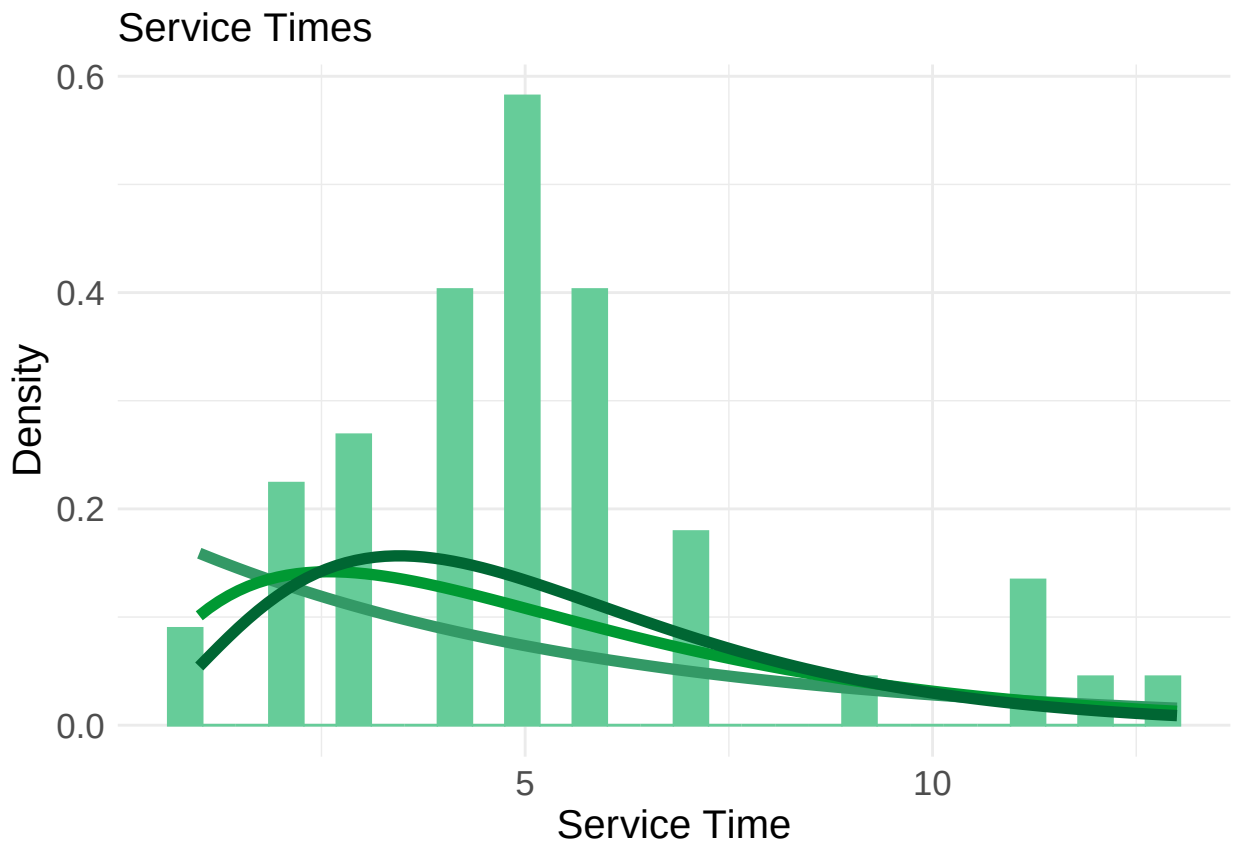



```
Dukin %>%
  ggplot(aes(Waiting.Time)) +
  geom_histogram(aes(y = after_stat(density)), bins = 30, fill = "#99ccff", color = "#99ccff") +
  theme_minimal() +
  theme(axis.title.x = element_text(size = 15),
        axis.text.x = element_text(size = 12.5),
        axis.title.y = element_text(size = 15),
        axis.text.y = element_text(size = 12.5),
        plot.title = element_text(size = 15)) +
  labs(title = "Waiting Times",
       x = "Waiting Time",
       y = "Density")
```




```
Dukin %>%
  ggplot(aes(Service.Time)) +
  geom_histogram(aes(y = after_stat(density)), bins = 30, fill = "#66cc99", color = "#66cc99") +
  stat_function(
    fun = dexp,
    args = list(rate = 1 / mean(Dukin$Service.Time)),
    lwd = 2,
    col = "#339966") +
  stat_function(
    fun = dgamma,
    args = list(shape = 2, rate = 2 / mean(Dukin$Service.Time)),
    lwd = 2,
    col = "#009933") +
  stat_function(
    fun = dgamma,
    args = list(shape = 3, rate = 3 / mean(Dukin$Service.Time)),
    lwd = 2,
    col = "#006633") +
  theme_minimal() +
  theme(axis.title.x = element_text(size = 15),
        axis.text.x = element_text(size = 12.5),
        axis.title.y = element_text(size = 15),
        axis.text.y = element_text(size = 12.5),
        plot.title = element_text(size = 15)) +
  labs(title = "Service Times",
       x = "Service Time",
```

```
       y = "Density")
```




```
#arrival and service rate per network node
(lambda = 1 / 1.574074)
```

```
## [1] 0.6352941
```

```
(mu = 4 / mean(Dukin$Service.Time))
```

```
## [1] 0.7714286
```

```
#expected time spent at each node
(w = 1 / (mu - lambda))
```

```
## [1] 7.345681
```

```
#expected time spent in the entire network
4 * w
```

```
## [1] 29.38272
```