STA240 Final Project

Anthony Zhao, Abby Li, William Yan

Scenario 1

Customer Arrival

Poisson process (rate = λ)

- T_k: Arrival time of the kth customer
- W_k: Time between the $k-1$th arrival and the kth arrival

$$W_k = T_k - T_{k-1}.$$

$W_k \sim Pois(\lambda)$

where lambda = 5

Service Time

$S_k \sim Exp(\lambda)$

where lambda = 6

Arrival Times

```
# simulating the arrival times of customers throughout the day

# Poisson process (lambda = 5)
# Tk= arrival time of the kth customer
# Wk= time between the k-1th customer arrival and the kth customer arrival where Wk ~ Pois(la
```

```r
# set parameter
lambdaA <- 5 # in units: customers per hour
hours <- 12 # operating hours: 10am to 10pm
total_time <- hours*60 # operating hours in minutes
lambdaA <- 5/60 # customers per minute
# converting to minutes because our lambda is low, and we can can get greater precision in ar

n <- ceiling(lambdaA*total_time) # max number of customers the store can have throughout the

# generate W1,..,Wn (calculating the time between the arrival times of 2 customers)
W_sample <- rexp(n, rate= lambdaA)

# calculate T or the arrival times by summing together the Wi arrival times

T_sample <- numeric(n)

for(i in 1:n) {
  T_sample[i] <- sum(W_sample[1:i])
}

# all possible arrival times of customers throughout the day (X minutes after opening)

# however, the store is only open for 12 hours or 720 minutes so we must get rid of the value

arrival_times <- T_sample[T_sample <= total_time]

arrival_times
```

```
 [1]  28.02106  28.67609  40.01211  61.39682  77.95015  88.15381  91.73525
 [8] 112.74095 146.80043 147.75133 148.64691 162.29404 167.94815 198.77323
[15] 199.25159 200.97746 202.43130 203.05280 209.64057 217.68064 228.18873
[22] 249.20487 264.00761 273.23794 273.98955 282.14972 311.20089 325.93528
[29] 341.81748 347.31328 359.53838 382.38080 384.96448 390.18433 394.08461
[36] 396.48476 399.46880 442.95552 452.00275 454.88584 485.87243 498.12153
[43] 499.12880 508.76798 526.50091 531.93787 536.54344 537.85425 541.00846
[50] 551.57833 552.52801 552.97726 555.45951 579.77999 583.34149 593.04277
[57] 613.39125 616.25780 631.16137 636.28821
```

Arrival Times Analysis

```r
length(arrival_times) # 55 customers
```

```
[1] 60
```

```r
mean(arrival_times) # 323 mins, 5.4 hours
```

```
[1] 346.9912
```

```r
min(arrival_times) # 10 mins
```

```
[1] 28.02106
```

```r
max(arrival_times) # 711 mins, 11.8 hours
```

```
[1] 636.2882
```

In this simulation, the number of customers that will be arriving within the operating hours is 55, with the first customer arriving 10 minutes after opening (~10:10 am) and the last customer arriving 11.8 hours into the workday (~9:48 pm).

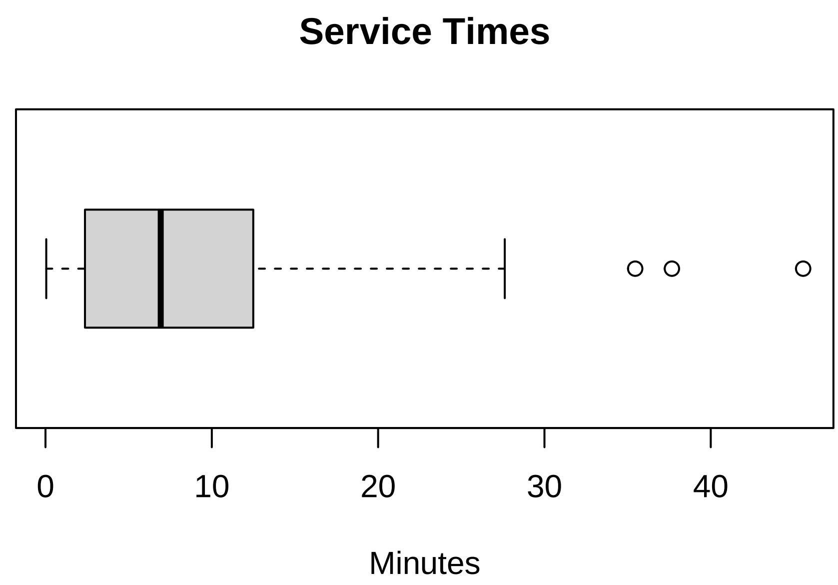
Serving Times

```r
# given the output from above, simulate the serving times of customers before they leave

# notice that service time is modeled by exp(6)
lambdaS <- 6 # customers per hour
lambdaS <- 6/60 # customers per minute

# simulate customer's service time
# n= only simulating the service time for those where T_sample <= total_time
service_times <- rexp(length(arrival_times), rate= lambdaS)

service_times #these are the serving times for each arriving customer before they leave
```

```
 [1] 12.50798299  2.16021842  0.54837804  2.70822262  8.63208515  7.65463441
 [7]  0.04705043  6.75998895  4.46667010  3.04786467  0.96959724  3.30503666
[13] 18.42674388  7.21754568 21.59389938  8.06538386  1.82727380  8.34763619
[19]  7.00876507 11.84066924  2.58866606 35.45590866  3.86566940  2.10351181
[25]  8.99132880 19.00686493  1.97679752 10.42984302  6.91371934  0.76155582
[31]  4.23562444 21.63867438  1.92852190  0.11005027 19.30486709  3.73027794
[37]  0.04608486 11.20239452  0.44774704 19.92360990 45.55483509 11.84561811
[43]  4.52898790  1.63431367  4.70470978  6.94309534 13.48672184 21.69659272
[49] 37.66368439  5.90035166  6.66307666 27.61374974  0.46106524  7.96501183
[55] 14.24711082  0.18864998 10.64783112 12.47828118 18.77132164  3.18826067
```

Serving Times Analysis

```r
boxplot(service_times, horizontal= TRUE, main= "Service Times", xlab= "Minutes")
```




```r
mean(service_times) # 8.5 mins
```

```
[1] 9.466344
```

The average service time is 8.5 minutes, with the data skewed right, consistent with an exponential distribution. This indicates that service times tend to lower.

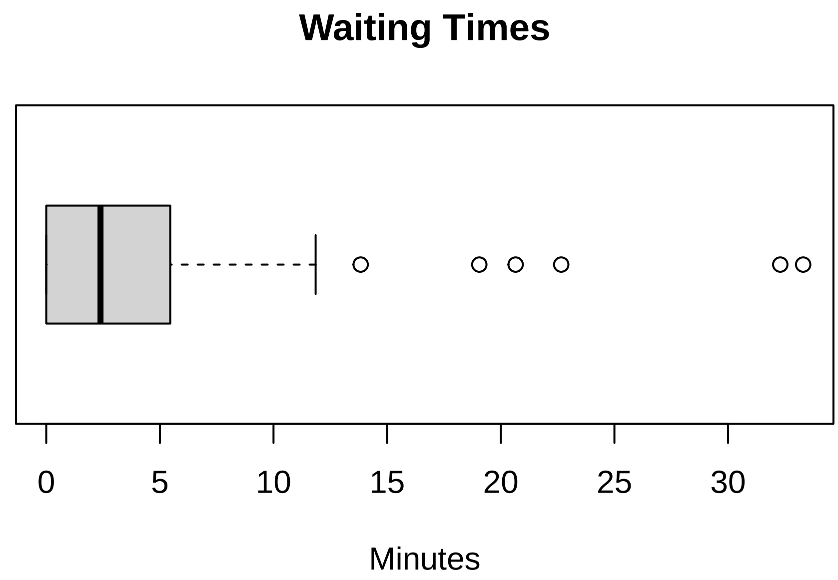
Waiting Times

```
# determining waiting times

# for each observation (customer), calculate when the service begins and when it ends
# serving ends = service begins + service time
# service begins: either when the customer walks in, or when the previous customer leaves (as

# compare this to the arrival time
# if arrival time > time service ends then wait time = 0
# but if arrival time < service time ends then wait time = time service ends- arrival time

# variable initialization
waiting_times <- numeric(length(arrival_times))  # generating times for each customer
service_start <- numeric(length(arrival_times))
service_end <- numeric(length(arrival_times))
current_end <- numeric(0) # service end time for current customer (i)

# iterate over each customer
for (i in 1:length(arrival_times)) {

  # only includes observations where service time > arrival time => which means there is a wa
  # gets rid of observations where service < arrival time => 0 wait time
  if (length(current_end) > 0) {
    current_end <- current_end[current_end > arrival_times[i]]
  }

 if (length(current_end) == 0) {
   # scenario 1: if there is no waiting time, service starts at the customer arrival
    service_start[i] <- arrival_times[i]
  } else {
    # scenario 2: if there is a waiting time, service starts at the end of the previous custo
    service_start[i] <- min(current_end)
  }

  # update the service end time for current customer by adding when service starts and how lo
  service_end[i] <- service_start[i] + service_times[i]

  # add this service end time to current end services
  current_end <- c(current_end , service_end[i])

  # update waiting time
```

```r
  waiting_times[i] <- service_start[i] - arrival_times[i]
}


simulation_results <- data.frame(
  customer = 1:length(arrival_times),
  arrival_time = arrival_times,
  service_length = service_times,
  service_start = service_start,
  service_end = service_end,
  waiting_time = waiting_times
)

print(head(simulation_results, 15)) # printing first 15 customers
```

```
   customer arrival_time service_length service_start service_end waiting_time
1         1     28.02106    12.50798299      28.02106    40.52904    0.0000000
2         2     28.67609     2.16021842      40.52904    42.68926   11.8529566
3         3     40.01211     0.54837804      40.52904    41.07742    0.5169289
4         4     61.39682     2.70822262      61.39682    64.10504    0.0000000
5         5     77.95015     8.63208515      77.95015    86.58224    0.0000000
6         6     88.15381     7.65463441      88.15381    95.80844    0.0000000
7         7     91.73525     0.04705043      95.80844    95.85549    4.0731864
8         8    112.74095     6.75998895     112.74095   119.50094    0.0000000
9         9    146.80043     4.46667010     146.80043   151.26710    0.0000000
10       10    147.75133     3.04786467     151.26710   154.31497    3.5157757
11       11    148.64691     0.96959724     151.26710   152.23670    2.6201975
12       12    162.29404     3.30503666     162.29404   165.59908    0.0000000
13       13    167.94815    18.42674388     167.94815   186.37489    0.0000000
14       14    198.77323     7.21754568     198.77323   205.99078    0.0000000
15       15    199.25159    21.59389938     205.99078   227.58467    6.7391876
```

```r
boxplot(waiting_times, horizontal= TRUE, main= "Waiting Times", xlab= "Minutes")
```




```
mean(waiting_times)
```

```
[1] 4.946483
```

Waiting times tend to be short, if not zero, and on average, the waiting time is around 5 minutes.