LOAD SYSTEM

To a better understanding of how traffic flows in a system, we developed a simple simulation of a random system and observed it for a specific time to explain important components of the system. To do this exercise we use R codding to obtain the information necessary for the system load.

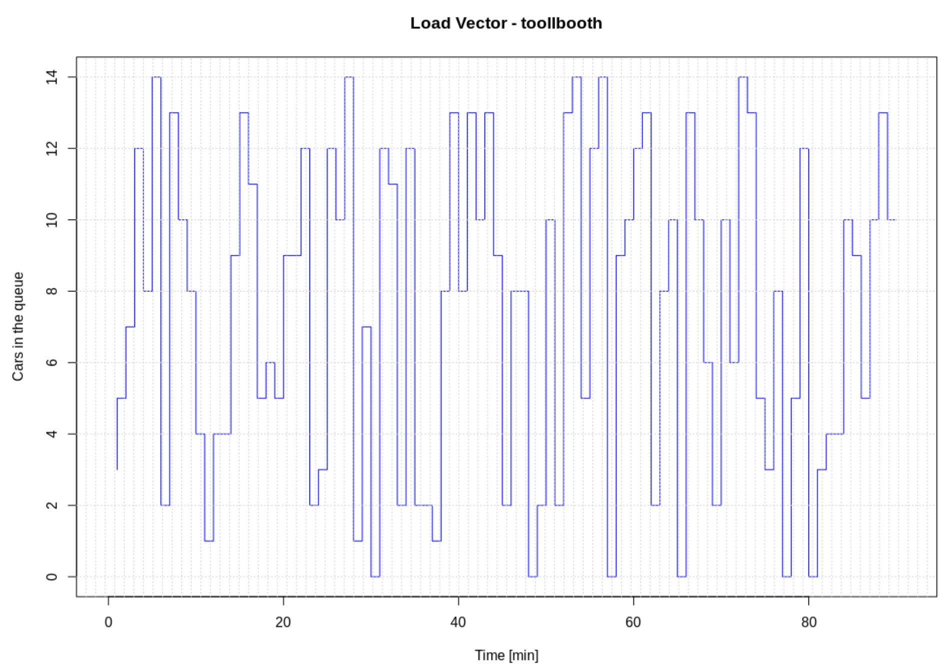
For this exercise we propose a tollbooth in a motorway where the tollbooth is our server and car arriving/departing are our jobs in the system.

VECTOR

To analyze our system, we should create a load vector. This vector contained the information about how many cars there are in the queue of the tollbooth waiting to be processes in a specific moment during the time of observation, this time was set as 90 minutes.

Using R it has been created a random vector with 90 values (time of observation, once per minute) and a maximum value of 15 for the length queue. This vector represents the load in our system depends on number of arrival/departures in each time of representation.

```
> # Create random vector (System Load)
> load_vector <- floor(runif(90, min = 0, max = 15))
```



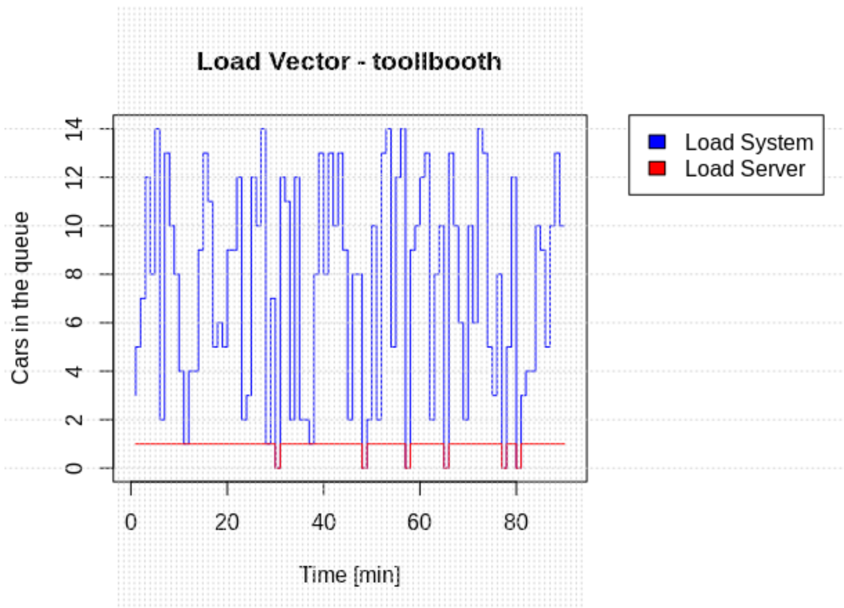
AVERAGE LOAD SERVER

Once we created our load vector we can start to compute relevant information about the system, the first value to compute is **Average Load in Server**.

As we know, our server only can process one car at time. So, to compute the average of load in the server we should kwon how many times the tollbooth has at least one car in the queue.

Using a logical comparation in R we obtained when our system is vacancy and when it is busy.

```
1. > #Logical condition to evaluate when the system is vancacy/busy
2. > bin_load_svr <- load_vector !=0
```



Knowing when the system has a load, now we can compute the average load in server just summarizing how many times our system is busy dived by the time of observation. With R we compute this value with the function "mean ()" and to a better understanding, we used the function "fraction ()" to keep the result in a simplified fraction.

```
1. > #Compute of average of load in the server
2. > avg_load_svr <- fractions(mean(bin_load_svr))
3. > avg_load_svr
4. [1] 14/15
```

AVERAGE LOAD QUEUE

After computing the average of the server, we continuous to compute average load in the queue, to obtain this value we should count how many cars are in the queues but without the car which is being processed.

So, in R we only made a difference between the vector of load server and the vector load system, that means to discount one car from the queue when the tollbooth is busy because there is a car which is being processes.

```
> #Compute of length of queues in the vector
> queue_length <- load_vector - bin_load_svr
```

Once we have the difference between load vector and load server vector, we use this new vector to compute the average of queue length. Basically, we only have to summarize the vector of queue length and dived by the time of observation. In R we used the function "mean ()" to calculate the average and "fractions ()" to keep the result in a simply fraction.

```
> #Compute of average length queue
> avg_queue <- fractions(mean(queue_length))

> avg_queue
[1] 292/45
```

AVERAGE LOAD SYSTEM

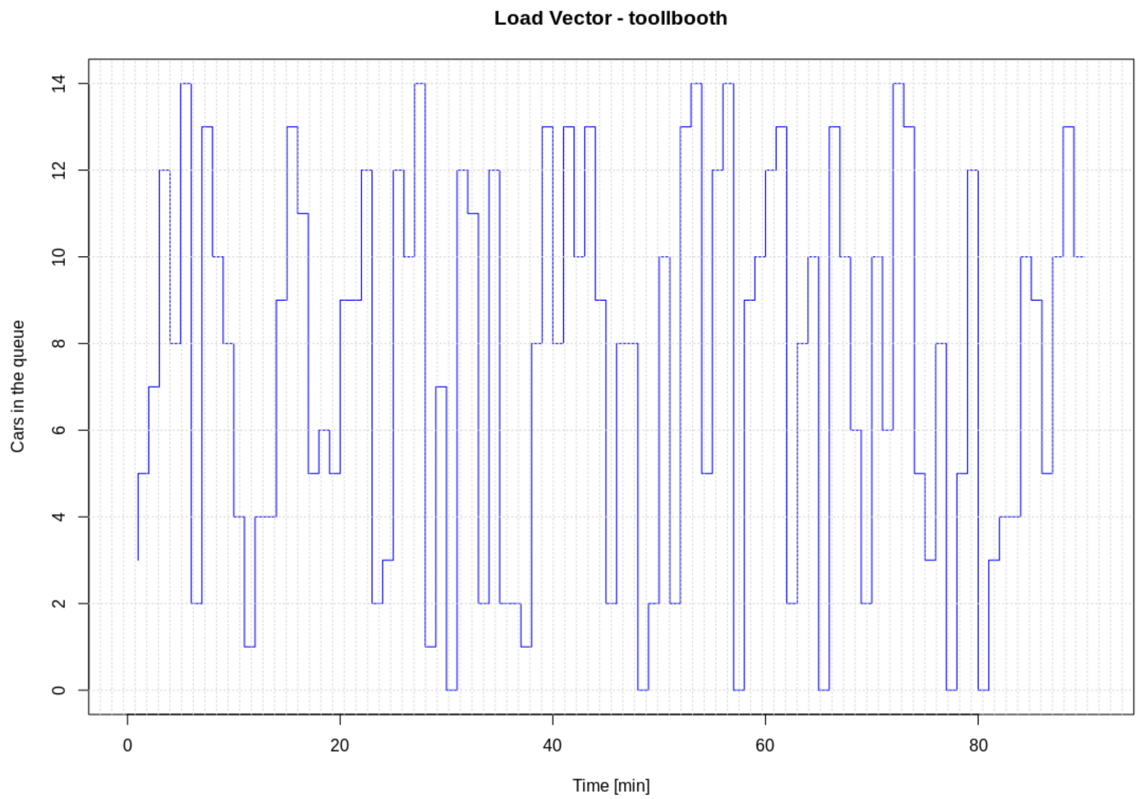
To have an entire vision of our system we have to compute average of load in the system, this average must have all the entire load during the time of observation.

To compute the average of load in the entire system we have to count how many cars there were during the all process and divide by the time of observation.

In R we used the function "mean ()" to calculate the average and "fractions ()" to keep the result in a simply fraction.

```
> #Compute of cars in the system during the time of observation
> avg_cars_fraction <- fractions(mean(load_vector))

> avg_cars_fraction
[1] 334/45
```


Load Vector - toollbooth
Cars in the queue
0
2
4
6
8
10
12
14
0
20
40
60
80
Time [min]