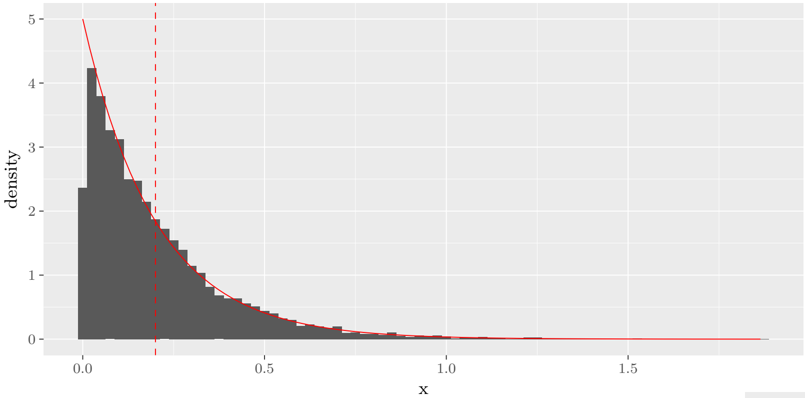
Stochastic effects in queueing models

Consider the Exponential(5) distribution with mean $1/5 = 0.2$ and standard deviation $\sqrt{(1/5)^2} = 1/5 = 0.2$ and hence coefficient of variation 1.

```
library(ggplot2)

set.seed(2018); lambda<-5; df<-data.frame(x=rexp(10000,lambda))
ggplot(df, aes(x)) +
   geom_histogram(aes(y=..density..), binwidth=0.025) +
   stat_function(fun=function(x) dexp(x,rate=lambda), colour='red') +
   geom_vline(xintercept=1/lambda, colour='red', linetype='dashed')
```



R package: simmer

With the **R** package **simmer** we can easily (and efficiently) simulate a M/M/1 queue as well as many other queueing systems and networks.

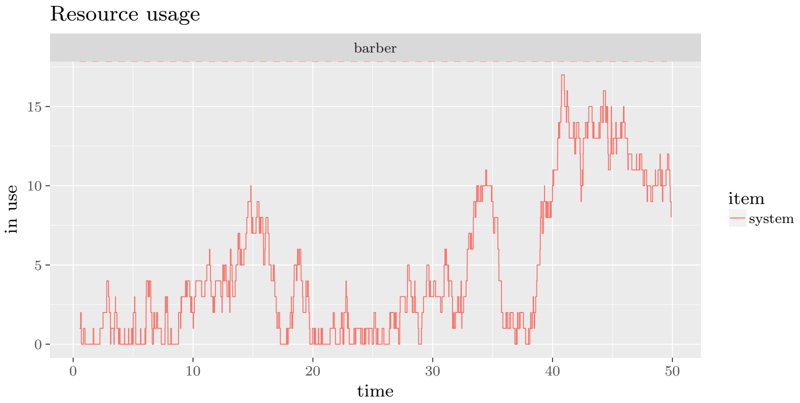
```r
# r-simmer.org

library(simmer); library(simmer.plot)

simulate_barber_mm1 <- function(duration) {
  lambda <- 5; mu <- 6

  barber_shop <- trajectory() %>%
    seize("barber", amount=1) %>%
    timeout(function() rexp(1,mu) ) %>%
    release("barber", amount=1)

  model <- simmer() %>%
    add_resource("barber", capacity=1, queue_size=Inf) %>%
    add_generator("customer", barber_shop, function() rexp(1,lambda)) %>%
    run(until=duration)
}
```

Simulation of M/M/1 model for Barber shop

```
barber_mm1_env <- simulate_barber_mm1(50)

head(get_mon_resources(barber_mm1_env))

##   resource      time server queue capacity queue_size system limit
## 1   barber 0.1120626      1     0        1        Inf      1   Inf
## 2   barber 0.1681311      0     0        1        Inf      0   Inf
## 3   barber 0.4261049      1     0        1        Inf      1   Inf
## 4   barber 0.5893962      0     0        1        Inf      0   Inf
## 5   barber 0.8137872      1     0        1        Inf      1   Inf
## 6   barber 1.2603662      0     0        1        Inf      0   Inf
##   replication
## 1           1
## 2           1
## 3           1
## 4           1
## 5           1
## 6           1

head(get_mon_arrivals(barber_mm1_env))

##        name start_time  end_time activity_time finished replication
## 1 customer0  0.1120626 0.1681311    0.05606846     TRUE           1
## 2 customer1  0.4261049 0.5893962    0.16329128     TRUE           1
## 3 customer2  0.8137872 1.2603662    0.44657897     TRUE           1
## 4 customer3  1.5030851 1.9256380    0.42255296     TRUE           1
## 5 customer4  1.5522889 1.9512939    0.02565582     TRUE           1
## 6 customer5  1.8220153 2.0959025    0.14460863     TRUE           1
```

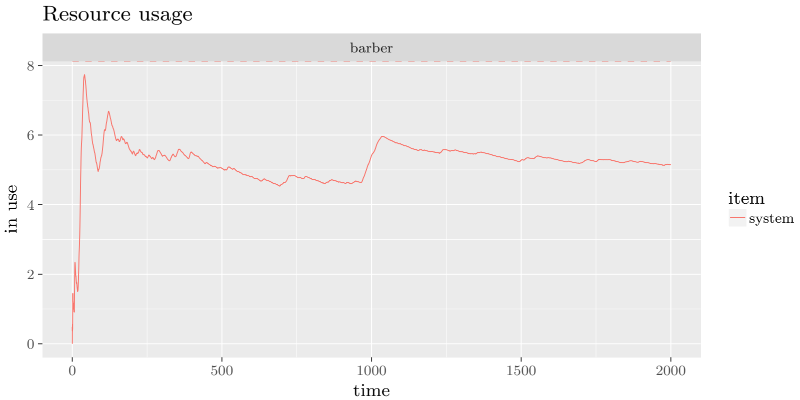
Simulation results

```
barber_mm1_env <- simulate_barber_mm1(50)

plot(get_mon_resources(barber_mm1_env), items="system", steps=TRUE)
```



Simulation results (ctd)

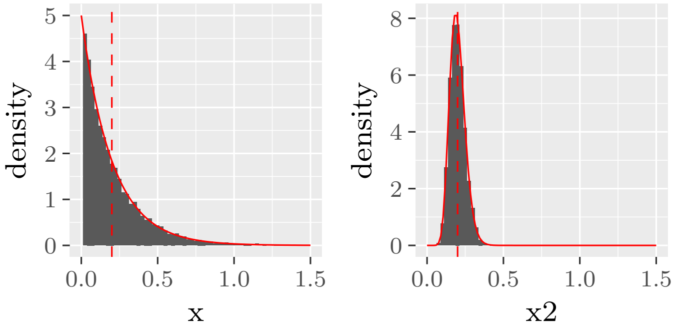
```
barber_mm1_env <- simulate_barber_mm1(2000)

plot(get_mon_resources(barber_mm1_env), items="system")
```



Erlang distribution

Consider an Erlang distribution with $r = 16$ stages each an independent Exponential($r\lambda$) distribution with mean $r(1/(r\lambda)) = 1/\lambda$ and standard deviation $\sqrt{r(1/(r\lambda))^2} = 1/(\sqrt{r}\lambda)$ and hence coefficient of variation $1/\sqrt{r} = 1/\sqrt{16} = 1/4 = 0.25 < 1$.

```
lambda<-5; r <- 16; df<-data.frame(x=rexp(10000,lambda), x2=rgamma(10000,shape=r,rate=r*lambda))
ggplot(df,aes(x)) + geom_histogram(aes(y=..density..),binwidth=0.025) +
   stat_function(fun=function(x) dexp(x,rate=lambda), colour='red') +
   geom_vline(xintercept=1/lambda, colour='red', linetype='dashed') + xlim(0,1.5)
ggplot(df,aes(x2)) + geom_histogram(aes(y=..density..),binwidth=0.025) +
   stat_function(fun=function(x) dgamma(x,shape=r,rate=r*lambda), colour='red') +
   geom_vline(xintercept=1/lambda, colour='red', linetype='dashed') + xlim(0,1.5)
```



Simulation of $E_{16}/E_{16}/1$ queue

```r
# r-simmer.org
simulate_barber_e16e161 <- function(duration) {
  lambda <- 5; mu <- 6; r <- 16

  barber_shop <- trajectory() %>%
    seize("barber", amount=1) %>%
    timeout(function() rgamma(1, shape=r, rate=r*mu) ) %>%
    release("barber", amount=1)

  model <- simmer() %>%
    add_resource("barber", capacity=1, queue_size=Inf) %>%
    add_generator("customer", barber_shop, function()  rgamma(1, shape=r, rate=r*lambda)) %>%
    run(until=duration)
}
```

Simulation results for $E_{16}/E_{16}/1$ queue

```
barber_e16e161_env <- simulate_barber_e16e161(50)

plot(get_mon_resources(barber_e16e161_env), items="system", steps=TRUE)
```