Distributions in R

Shuhan Song

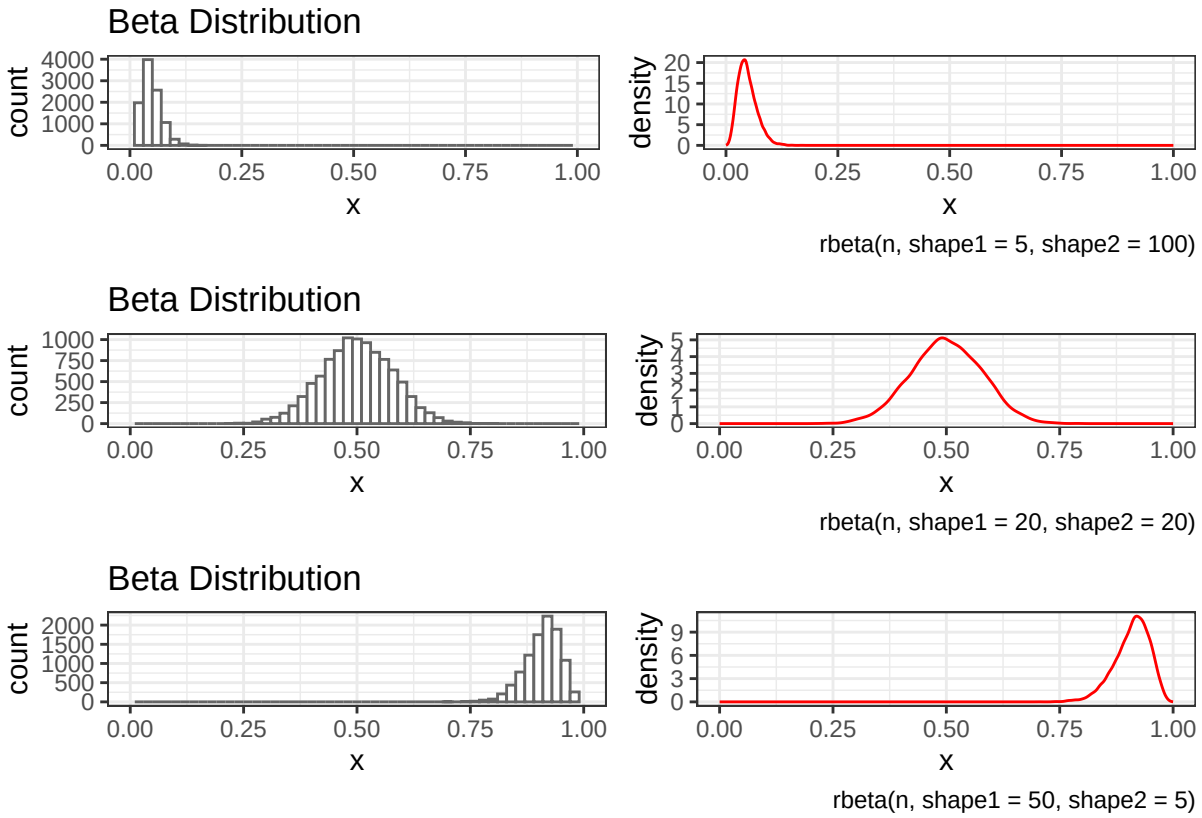
2022-12-27

Contents

Beta Distribution rbeta()

```
args(rbeta)
```

```
## function (n, shape1, shape2, ncp = 0)
## NULL
```



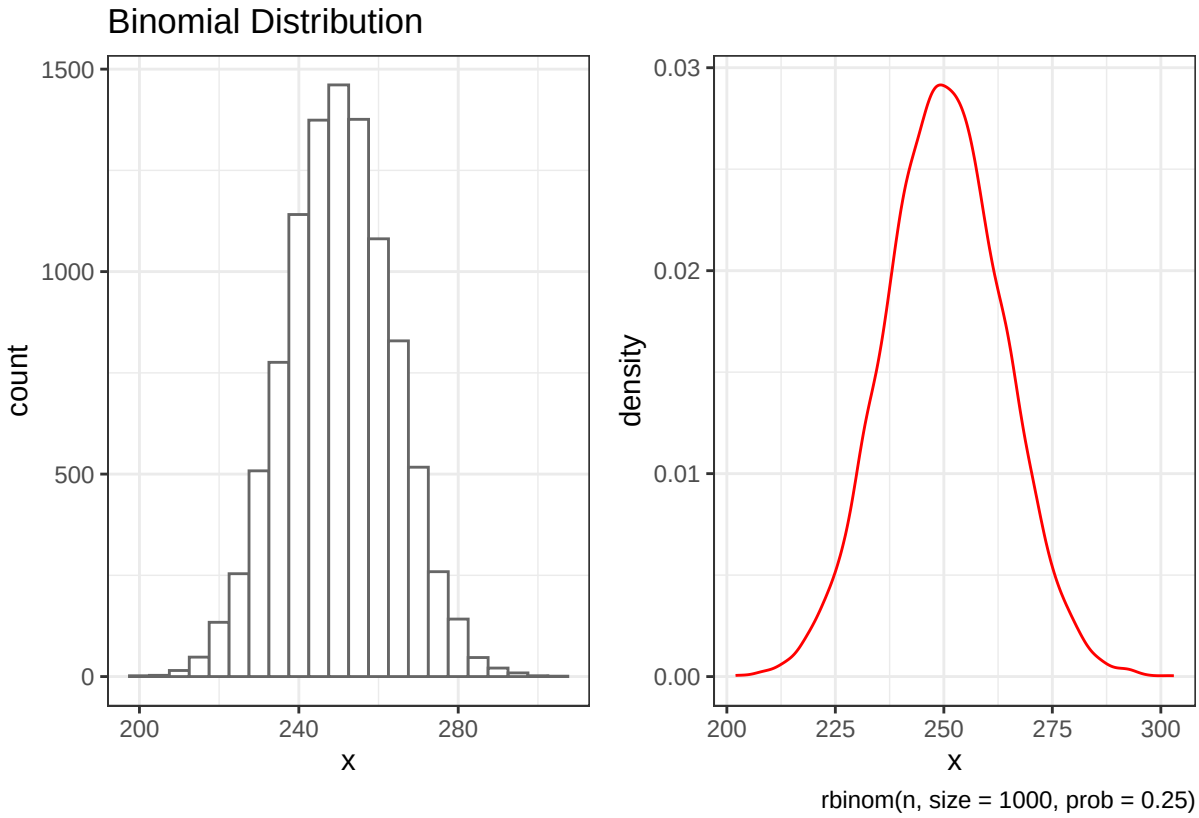
Binomial Distribution rbinom()

```
args(distribution_fcn_ls$rbinom)
```

```
## function (n, size, prob)
## NULL
```

size: number of trails

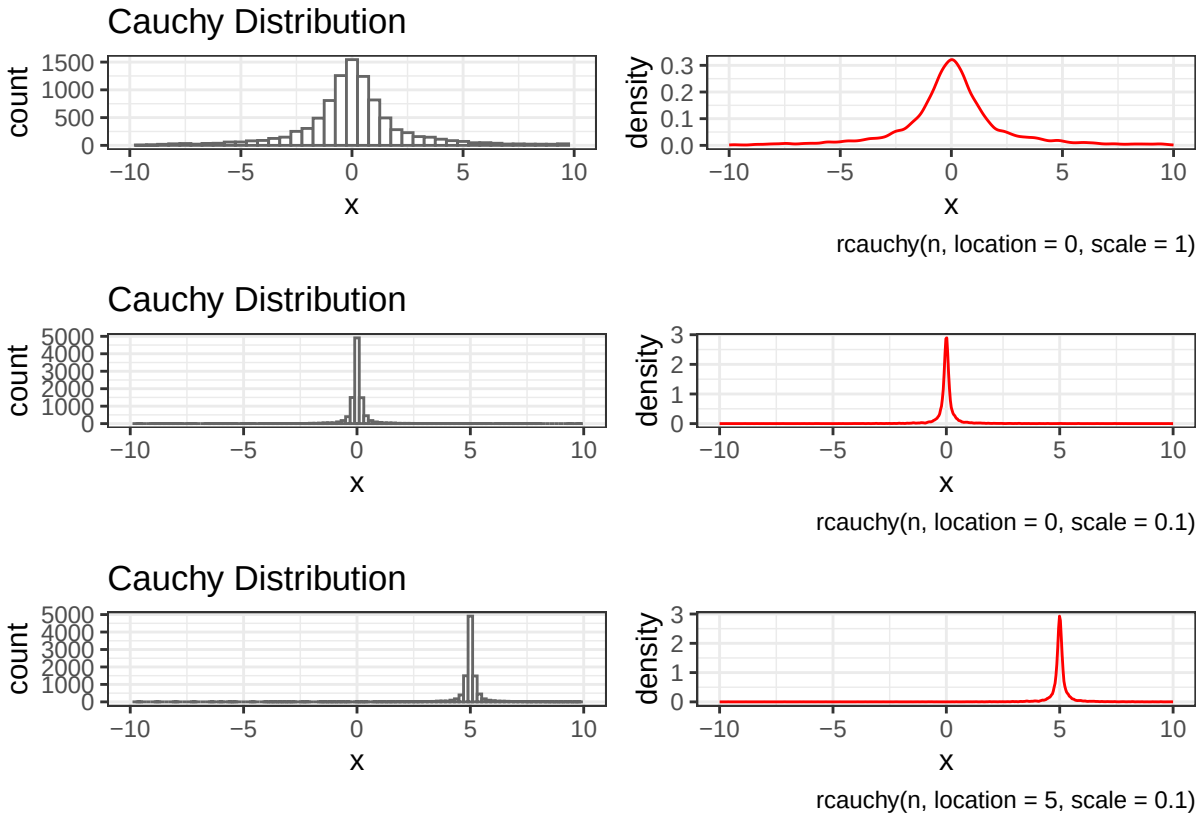
prob: probability of success on each trail



Cauchy Distribution rcauchy()

```
args(distribution_fcn_ls$rcauchy)
```

```
## function (n, location = 0, scale = 1)
## NULL
```



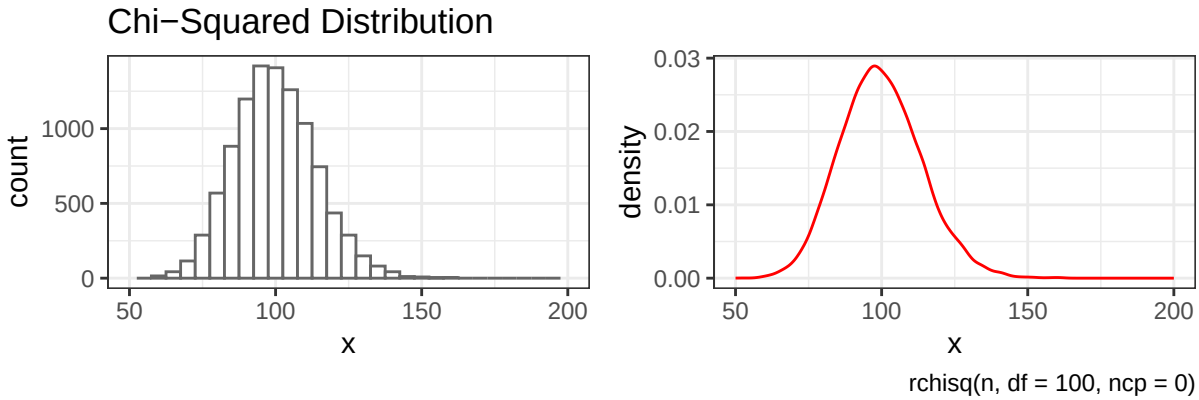
Chi-Squared Distribution rchisq()

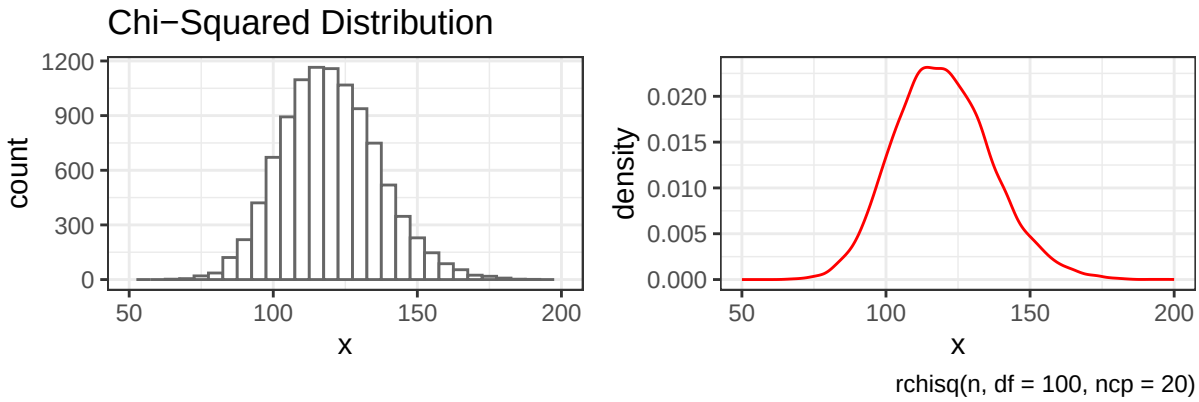
```
args(distribution_fcn_ls$rchisq)
```

```
## function (n, df, ncp = 0)
## NULL
```

df: degree of freedom

ncp: non-centrality parameter (non-negative)



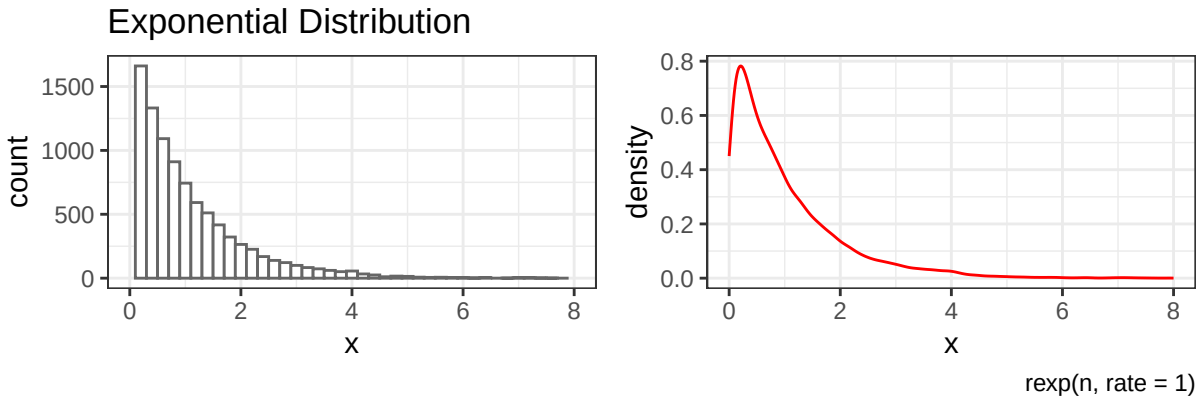


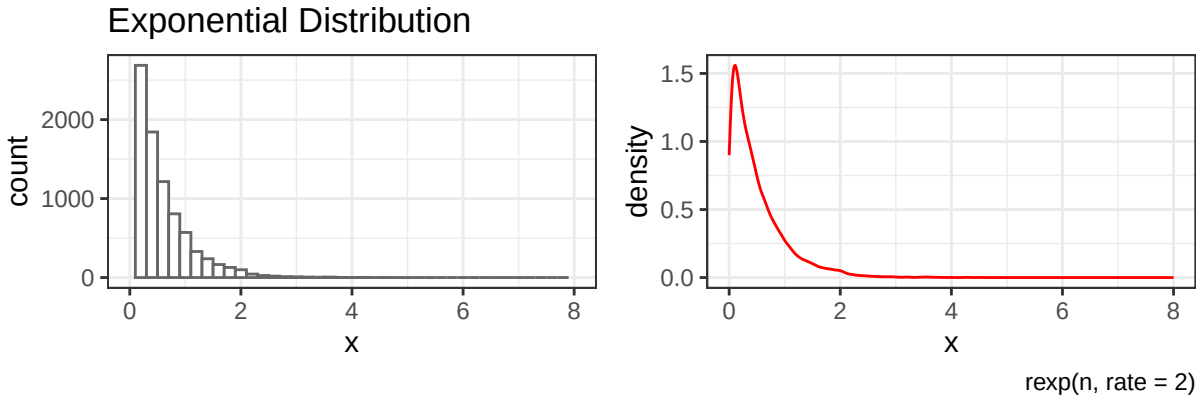
Exponential Distribution rexp()

```
args(distribution_fcn_ls$rexp)
```

```
## function (n, rate = 1)
## NULL
```

rate: vector of rates





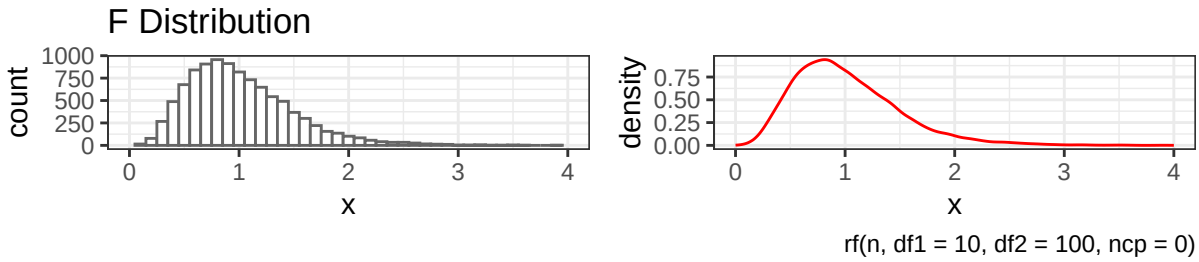
F Distribution rf()

```
args(distribution_fcn_ls$rf)
```

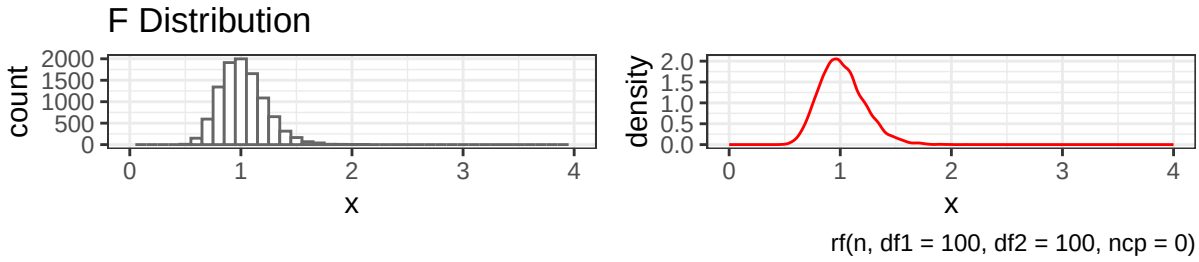
```
## function (n, df1, df2, ncp)
## NULL
```

df1, df2: degrees of freedom. Inf is allowed.

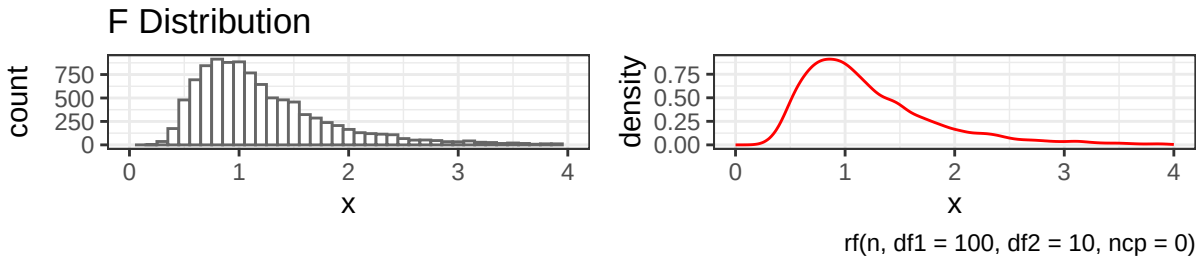
ncp: non-centrality parameter. If omitted the central F is assumed.



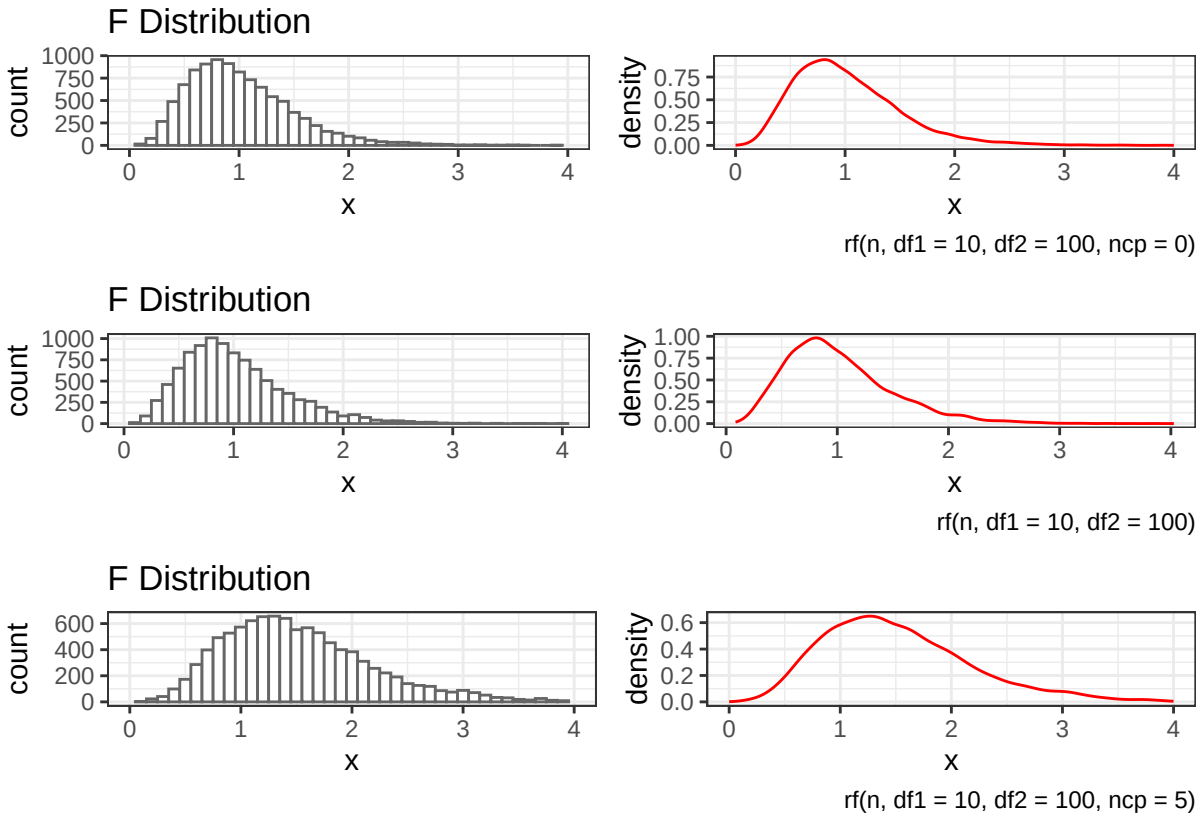
rf(n, df1 = 10, df2 = 100, ncp = 0)



rf(n, df1 = 100, df2 = 100, ncp = 0)



rf(n, df1 = 100, df2 = 10, ncp = 0)



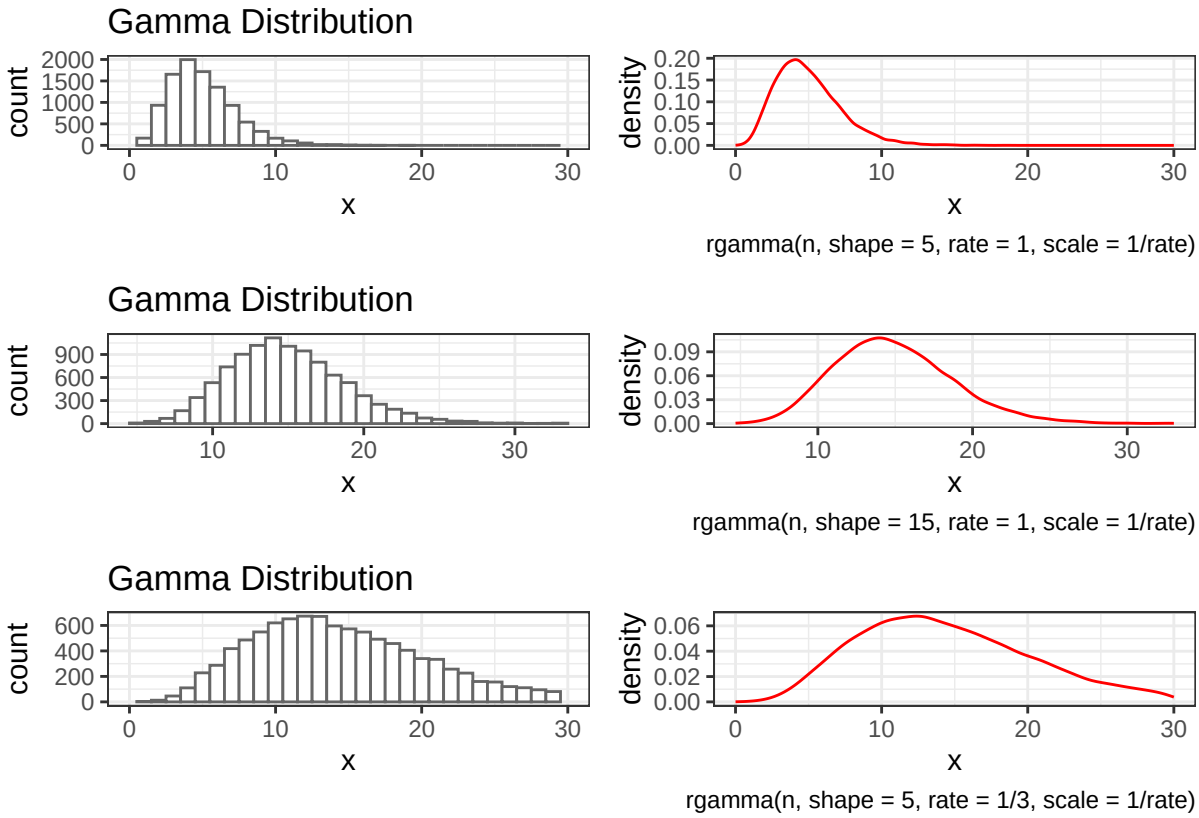
kk## Gamma Distribution rgamma()

```
args(distribution_fcn_ls$rgamma)
```

```
## function (n, shape, rate = 1, scale = 1/rate)
## NULL
```

shape, scale: shape and scale parameters. Must be positive, `scale` strictly.

rate: an alternative way specify the scale

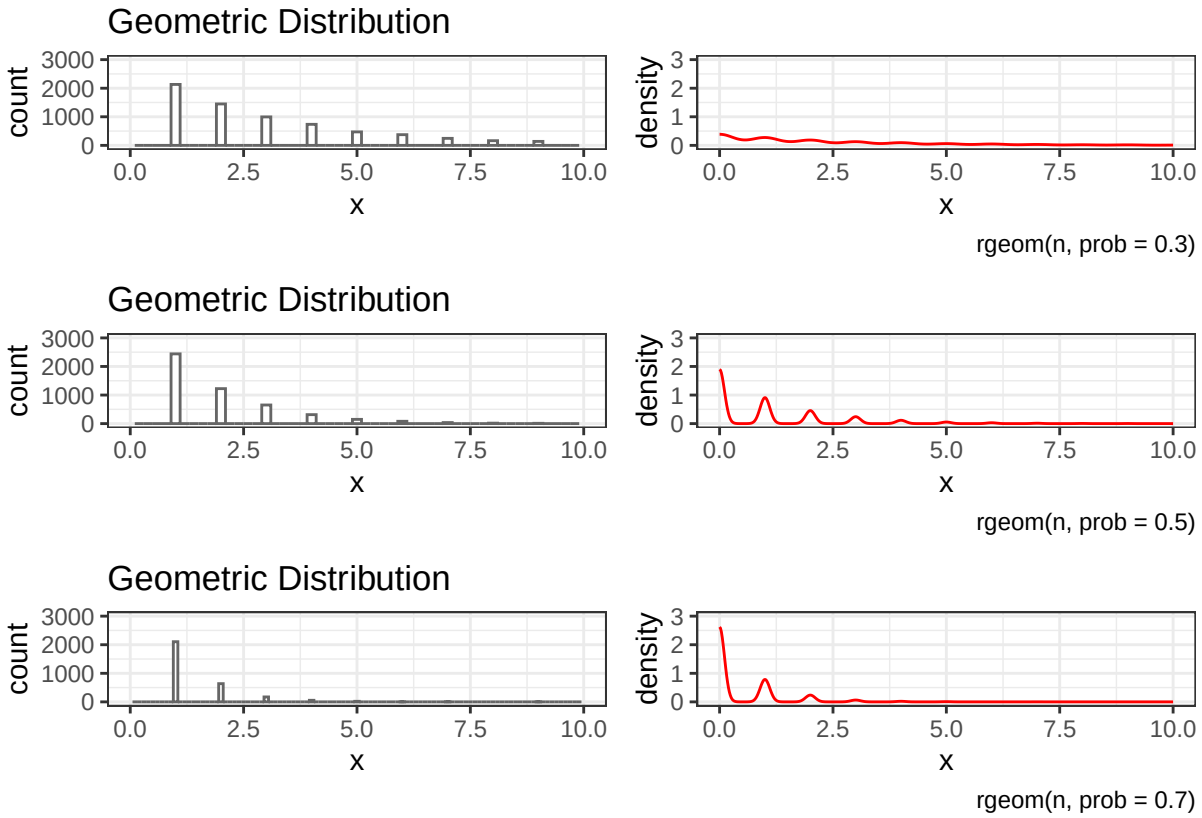


Geometric Distribution rgeom()

```
args(distribution_fcn_ls$rgeom)
```

```
## function (n, prob)
## NULL
```

prob: probability of success in each trial. $0 <$ prob $<= 1$.



Hypergeometric Distribution rhyper()

```
args(distribution_fcn_ls$rhyper)
```

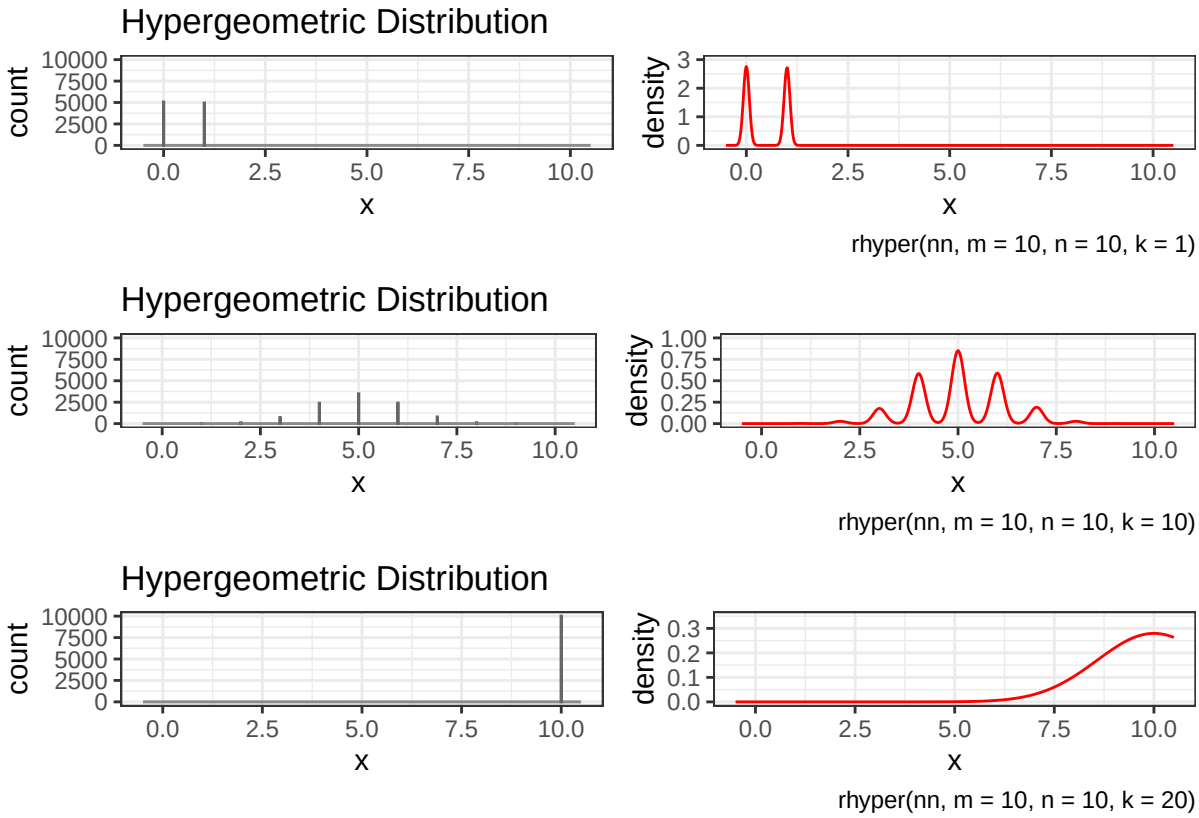
```
## function (nn, m, n, k)
## NULL
```

m: the number of white balls in the urn

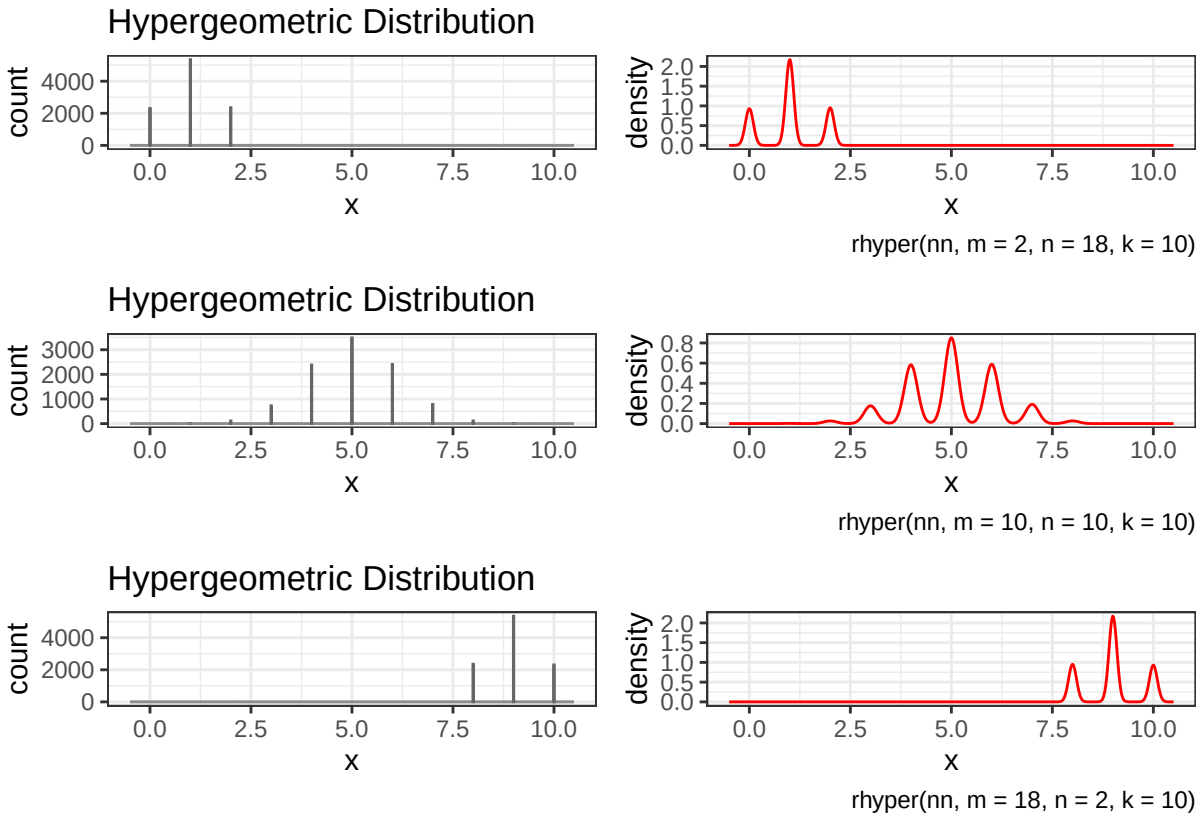
n: the number of black balls in the urn

k: the number of balls drawn from the urn. 0 <= k <= m+n

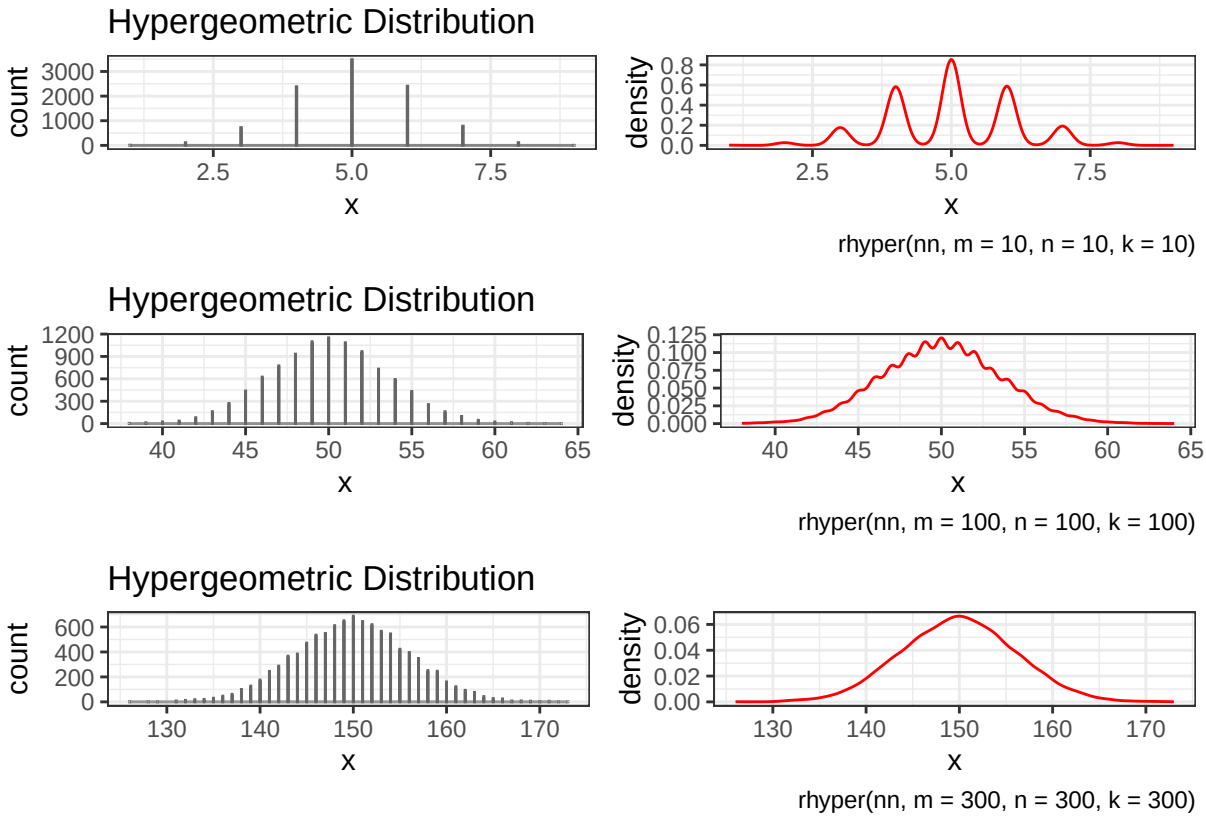
1. changing k



2. changing m/n ratio



3. changing m, n, k magnitude

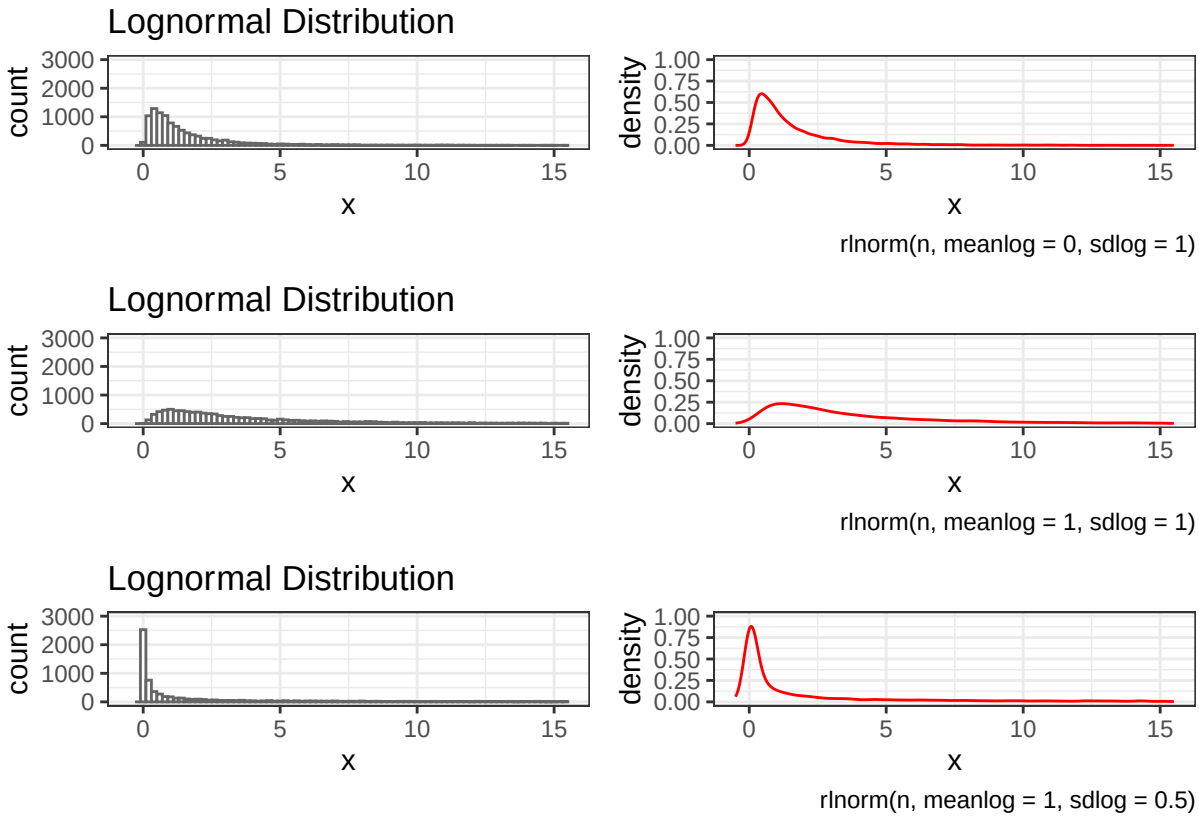


Lognormal Distribution rlnorm()

```
args(distribution_fcn_ls$rlnorm)
```

```
## function (n, meanlog = 0, sdlog = 1)
## NULL
```

meanlog, sdlog: mean and standard deviation of the distribution on the log scale with default values of 0 and 1 respectively.


Lognormal Distribution
count
3000
2000
1000
0
0
5
10
15
x
density
1.00
0.75
0.50
0.25
0.00
0
5
10
15
x
rlnorm(n, meanlog = 0, sdlog = 1)
Lognormal Distribution
count
3000
2000
1000
0
0
5
10
15
x
density
1.00
0.75
0.50
0.25
0.00
0
5
10
15
x
rlnorm(n, meanlog = 1, sdlog = 1)
Lognormal Distribution
count
3000
2000
1000
0
0
5
10
15
x
density
1.00
0.75
0.50
0.25
0.00
0
5
10
15
x
rlnorm(n, meanlog = 1, sdlog = 0.5)