Statistical Distributions

- Density, cumulative distribution function, quantile function and random variate generation for many standard probability distributions are available in the stats package.

  - dxxx : functions for the density/mass function,
  - pxxx : cumulative distribution function
  - qxxx : quantile function
  - rxxx : random variable generation.

2.3 pxxx

Cumulative distribution function (lower tail probability)

Example : Standard normal distribution

- `pnorm(value-of-x-axis)` or `pnorm(quantile)`

- `pnorm(0) = 0.5` (the area under the standard normal curve to the left of zero).

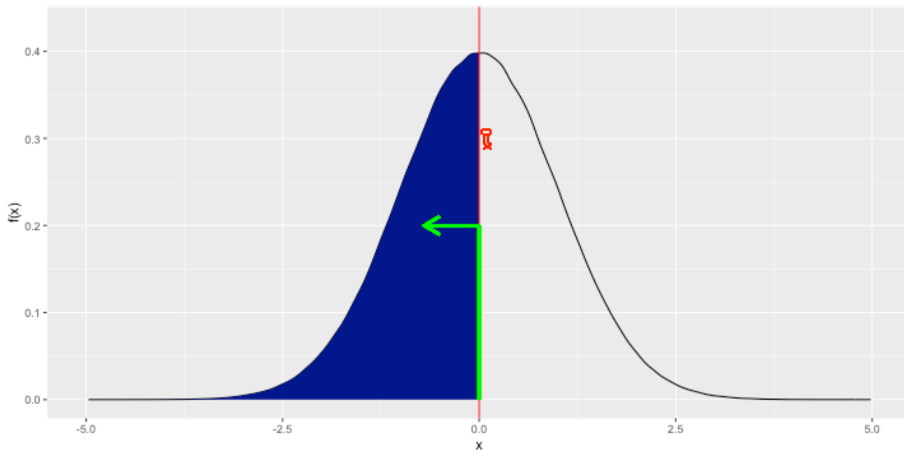


Figure 2.1: Standard normal distribution

- `pnorm(1.281552) = 0.9000` (the area under the standard normal curve to the left of 1.281).

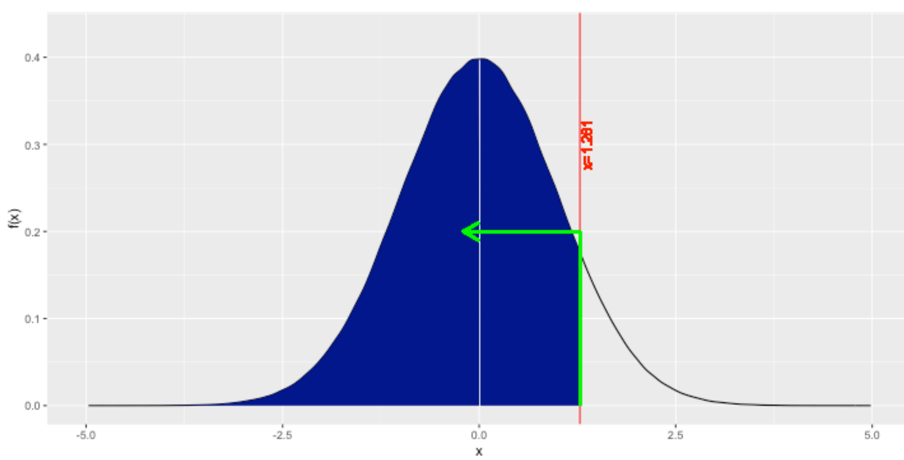


Figure 2.2: Standard normal distribution

- The `pnorm` function also takes the argument `lower.tail`. If `lower.tail` is set equal to `FALSE` then `pnorm` returns the upper tail probability (*the integral from q to ∞ of the pdf*) of the normal distribution.

- Note that

```
pnorm(1.281552)
```

```
## [1] 0.9000001
```

```
pnorm(1.281552, lower.tail = TRUE)
```

```
## [1] 0.9000001
```

```
pnorm(1.281552, lower.tail = FALSE)
```

```
## [1] 0.09999992
```

```
1-pnorm(1.281552, lower.tail = TRUE)
```

```
## [1] 0.09999992
```

2.4 qxxx

Example : Standard normal distribution

The `qnorm` function is simply the inverse of the cdf, which you can also think of as the inverse of `pnorm`!

- `qnorm(probability)`
- `qnorm(0.5) = 0` (0 is the 50th percentile of the standard normal distribution)

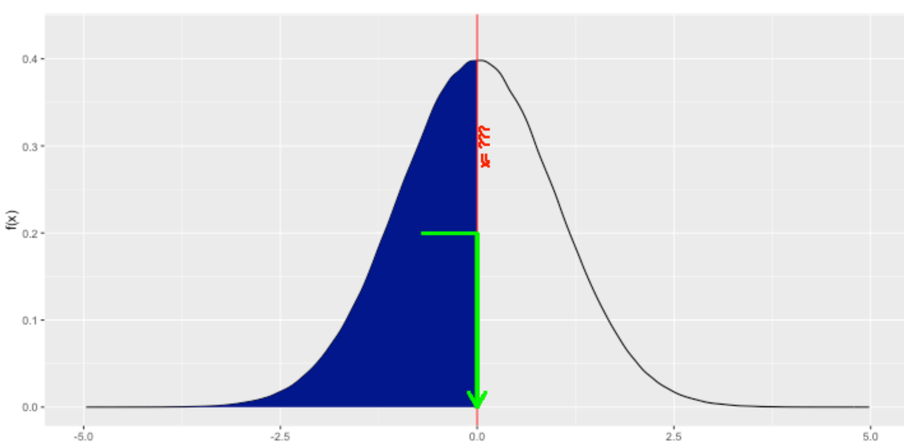


Figure 2.3: Standard normal distribution

```r
qnorm(0.5)
```

```
## [1] 0
```

```r
qnorm(0.9)
```

```
## [1] 1.281552
```

```r
qnorm(0.1, lower.tail = FALSE)
```

```
## [1] 1.281552
```

```r
qnorm(0.9,  lower.tail = FALSE)
```

```
## [1] -1.281552
```

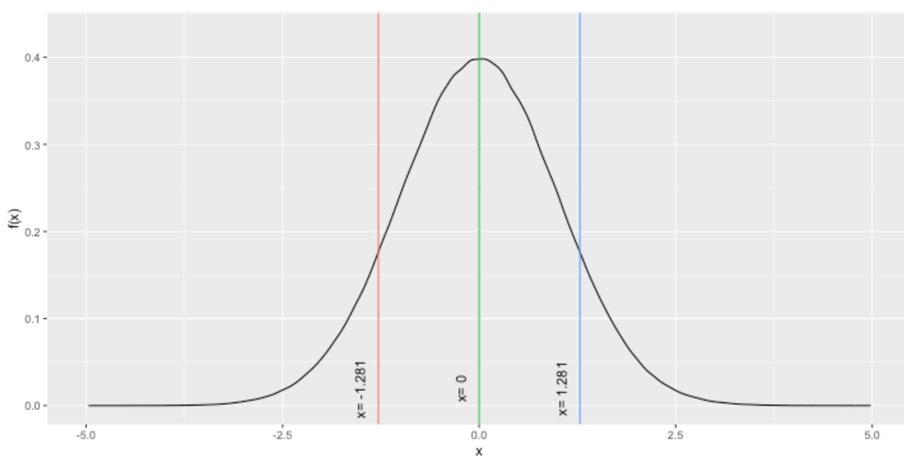


Figure 2.4: Standard normal distribution

2.5 dxxx

Example : Standard normal distribution

The function `dnorm` returns the value of the probability density function for the normal distribution given parameters for x, μ, and σ.

- `dnorm(0) == 1/sqrt(2*pi)`

If $Z \sim N(0,1)$, then

$$\phi_Z(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2}; \;\; -\infty < z < \infty$$

$$\phi_Z(0) = \frac{1}{\sqrt{2\pi}} = 0.3989423$$

```
dnorm(0)
```

```
## [1] 0.3989423
```

```
1/sqrt(2*pi)
```

```
## [1] 0.3989423
```

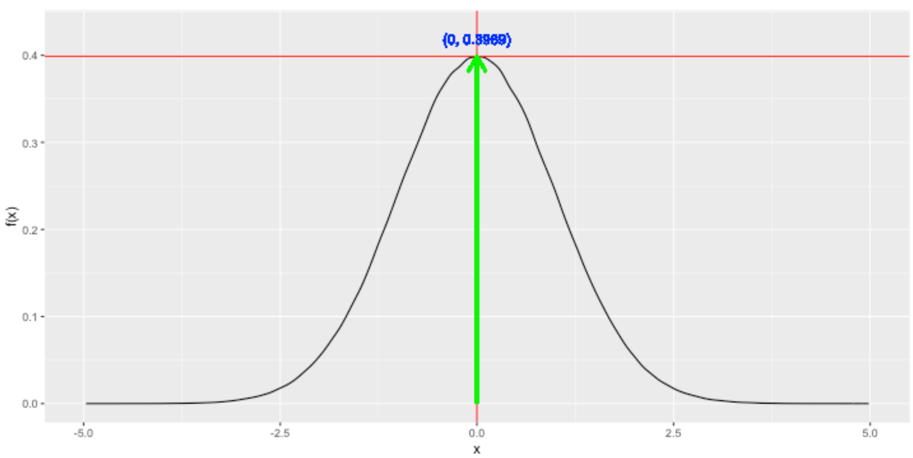


Figure 2.5: Standard normal distribution

$$\phi_Z(1) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}} = 0.3989423$$

```
dnorm(1)
```

```
## [1] 0.2419707
```

```
(1/sqrt(2*pi)) * exp(-1/2)
```

```
## [1] 0.2419707
```

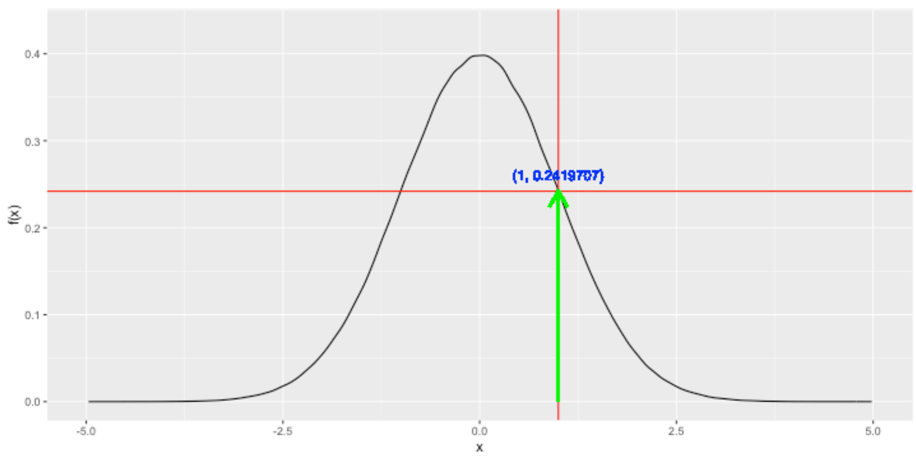


Figure 2.6: Standard normal distribution

2.6 rxxx

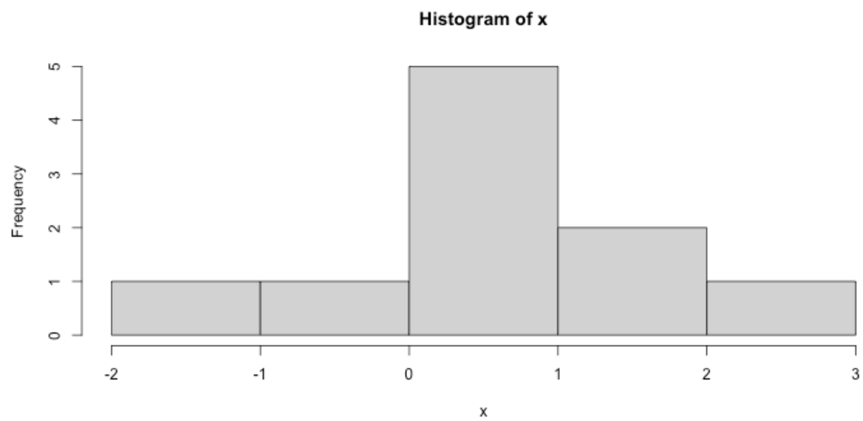
Example : Standard normal distribution

`rnorm(100)` generates 100 random deviates from a standard normal distribution.

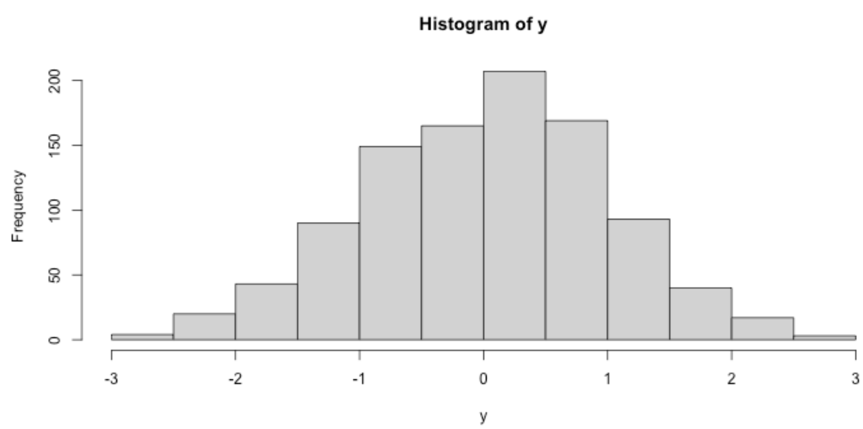
```
x <- rnorm(10)
x
```

```
##  [1] -1.0080707  1.3549394 -0.4689749  1.4681936  0.4425564  0.1462031
##  [7]  0.1715031  0.5925072  2.7647493  0.6192188
```

```
hist(x)
```




```r
y<- rnorm(1000)
hist(y)
```



2.7 Distributions in R

Distribution	R name	Additional arguments
beta	`beta`	`shape1, shape 2, ncp`
binomial	`binom`	`size, prob`
Cauchy	`cauchy`	`location, scale`
chi-squared	`chisq`	`df, ncp`
exponential	`exp`	`rate`
F	`f`	`df1, df2, ncp`

Distribution	R name	Additional arguments
gamma	`gamma`	`shape, scale`
geometric	`geom`	`prob`
hypergeometric	`hyper`	`m, n, k`
log-normal	`lnorm`	`meanlog, sdlog`
logistic	`logis`	`location, scale`
negative binomial	`nbinom`	`size, prob`
normal	`norm`	`mean, sd`
Poisson	`pois`	`lambda`
signed rank	`signrank`	`n`
Student's t	`t`	`df, ncp`
uniform	`unif`	`min, max`
Weibull	`weibull`	`shape, scale`
Wilcoxon	`wilcox`	`m, n`

2.8 Exercise

2.8.1 Normal

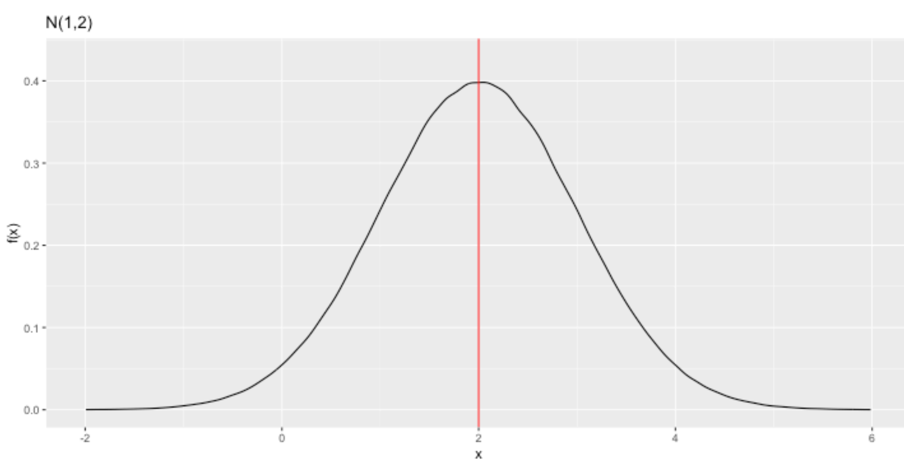


Figure 2.7: Standard normal distribution

```
# ?dnorm - Help page

dnorm(2, mean = 2, sd = 1, log = FALSE)
pnorm(2, mean = 2, sd = 1, lower.tail = TRUE, log.p = FALSE)
qnorm(0.5, mean = 2, sd = 1, lower.tail = TRUE, log.p = FALSE)
rnorm(n=10, mean = 2, sd = 1)
```

```
# ?dnorm - Help page

dnorm(2, mean = 2, sd = 1, log = FALSE)
```

```
## [1] 0.3989423
```

```
pnorm(2, mean = 2, sd = 1, lower.tail = TRUE, log.p = FALSE)
```

```
## [1] 0.5
```

```
qnorm(0.5, mean = 2, sd = 1, lower.tail = TRUE, log.p = FALSE)
```

```
## [1] 2
```

```
rnorm(n=10, mean = 2, sd = 1)
```

```
##  [1] 0.9919293 3.3549394 1.5310251 3.4681936 2.4425564 2.1462031 2.1715031
##  [8] 2.5925072 4.7647493 2.6192188
```

2.8.2 Gamma

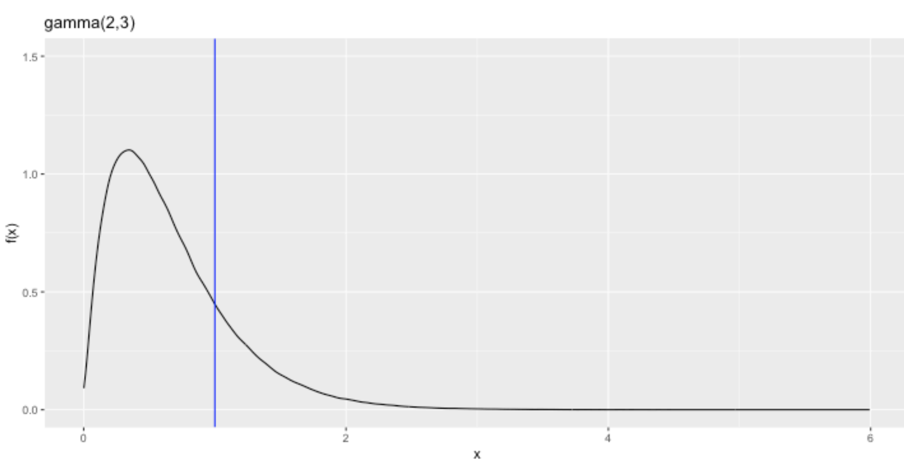


Figure 2.8: Standard normal distribution

```r
dgamma(1, shape =2, scale = 1/3, log = FALSE)
pgamma(1, shape =2, scale = 1/3, lower.tail = TRUE,
       log.p = FALSE)
qgamma(0.8, shape = 2,  scale = 1/3, lower.tail = TRUE,
       log.p = FALSE)
rgamma(10, shape =2 , scale = 1/3)
```

```r
dgamma(1, shape =2, scale = 1/3, log = FALSE)
```

```
## [1] 0.4480836
```

```r
pgamma(1, shape =2, scale = 1/3, lower.tail = TRUE,
       log.p = FALSE)
```

```
## [1] 0.8008517
```

```r
qgamma(0.8, shape = 2,  scale = 1/3, lower.tail = TRUE,
       log.p = FALSE)
```

```
## [1] 0.9981028
```

```r
rgamma(10, shape =2 , scale = 1/3)
```

```
##  [1] 0.3117027 0.5663228 1.8567817 0.1403567 0.7776307 0.8252981 1.4063366
##  [8] 0.7292312 0.9434481 0.9001169
```