# Cumulative density function (CDF)
# Percent point function (PPF)
# in Excel, R and Python

For distribution functions commonly used in inferential statistics (confidence intervals, tests) : Normal, Student, Chi-Squared, Fisher-Snedecor.

Ricco Rakotomalala

# Calculation of CDF and PPF in inferential statistics

Calculations of the quantiles and cumulative distribution functions values are required in inferential statistics, when constructing confidence intervals or for the implementation of hypothesis tests, especially for the calculation of the p-value.

Functions available in different tools allow us to obtain these values. We do not longer need to use statistical tables.

- Via Excel statistical functions (new functions are available from Excel 2010)
- Via R's statistical functions provided by the "stats" package (directly accessible)
- Via Python's statistical functions provided by the "scipy" package

```
import scipy.stats as stats
```

# NORMAL DISTRIBUTION

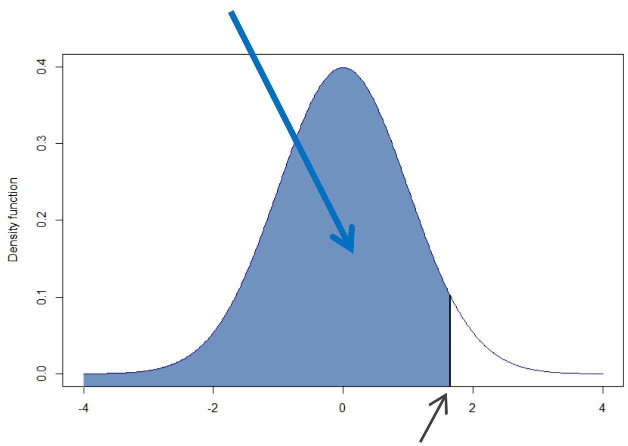
CDF of the **standard normal distribution (μ = 0 and σ = 1)**.

Probability of less than **x = 1.65** is equal to 0.9505285

Probability density function

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

**0.9505285**




**x=1.65**

EXCEL

**TRUE** for the CDF. If FALSE, we have the value of the density function. Required.

NORM.DIST(1.65, 0 , 1 , TRUE)

(μ = 0) and (σ = 1). Required settings.

NORM.S.DIST(1.65,TRUE) For the standard normal distribution.

R

pnorm(1.65, mean = 0, sd = 1, lower.tail = TRUE)

(μ = 0) and (σ = 1). Default.

TRUE: probabilities are $]-\infty\,;\,q\,]$. Default.

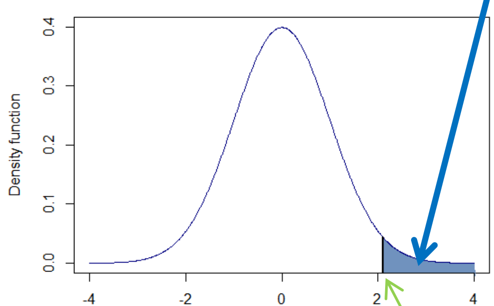
Python

stats.norm.cdf(1.65, loc = 0, scale = 1)

(μ = 0) and (σ = 1). Default.

Calculation of the p-value for the **standard normal distribution** in a right tailed test. The probability of more than **z = 2.1** is equal to 0.01786442

**p-value = 0.01786442**




**z=2.1**

## EXCEL

1- NORM.S.DIST(2.1, TRUE)

## R

1 - pnorm(2.1)

pnorm(2.1, lower.tail = FALSE)

Probabilities are $[z\ ;\ +\infty[$

## Python

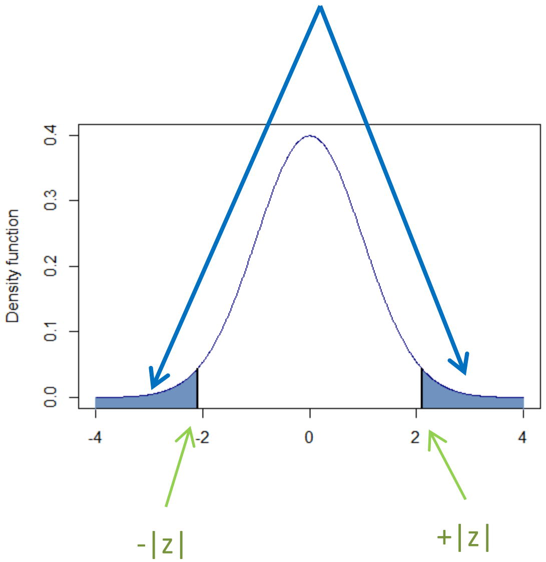
1 - stats.norm.cdf(2.1)

stats.norm.sf(2.1)

sf = 1 - cdf

Calculation of the p-value for **the standard normal distribution** in a two-tailed test. The probability of more than z = 2.1 in absolute value is equal to 0.03572884

**p-value = 2 * 0.01786442 = 0.03572884**




EXCEL

2*(1- NORM.S.DIST(2.1, TRUE))

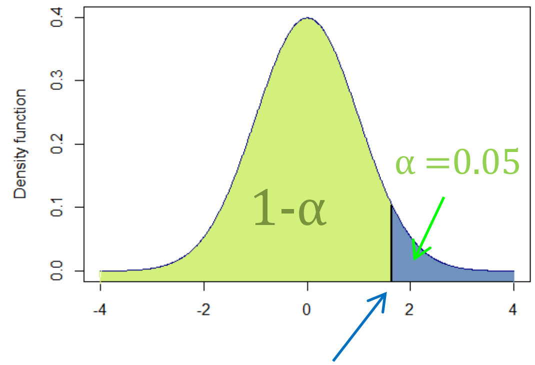
R

2 * pnorm(2.1, lower.tail = FALSE)

Python

2 * (1 - stats.norm.cdf(2.1))

PPF (q) of **the standard normal distribution** for the probability $(1-\alpha) = 0.95$




**q = 1.644854**

## EXCEL

NORM.INV(0.95, 0, 1)

NORM.S.INV(0.95)

---

## R

qnorm(**0.95**,mean=0,sd = 1,lower.tail=TRUE)

qnorm(**0.05**,mean=0,sd=1,lower.tail=FALSE)

---

## Python

stats.norm.ppf(**0.95**, loc =0, scale = 1)

# Generating random numbers from standard normal distribution

$$\mathcal{N}(\mu=0,\sigma=1)$$

RAND() returns an evenly distributed random real number greater than or equal to 0 and less than 1.

## EXCEL

```
NORM.S.INV(RAND())
```

---

## R

```
rnorm(n=1,mean=0,sd = 1)
```

Number of values to return. If (n > 1), we obtain a vector of values. Required.

---

## Python

Initialization of the generator. If random_state = integer, the values obtained are reproductible. Optional.

```
stats.norm.rvs(loc=0,scale=1, size=1, random_state = none)
```

Number of values to return. If (size > 1), we obtain a vector of values. Optional.

Approximations of the standard normal cumulative distribution function. Some “basic” formulas for (x > 0)

$$\Phi_1(x) = 1 - \frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi}}\left(\frac{0.4361836}{1+0.33267x} + \frac{-0.1201676}{(1+0.33267x)^2} + \frac{0.9772980}{(1+0.33267x)^3}\right)$$

(https://fr.wikipedia.org/wiki/Loi_normale)

$$\Phi_2(x) = 0.5 + \frac{1}{2}\left\{1 - \frac{1}{30}\left[7e^{-\frac{x^2}{2}} + 16e^{-x^2(2-\sqrt{2})} + \left(7 + \frac{1}{4}\pi x^2\right)e^{-x^2}\right]\right\}^{\frac{1}{2}}$$

(http://mathworld.wolfram.com/NormalDistributionFunction.html)

⇨ $\Phi_1(1.65) = 0.9494966$

$\Phi_2(1.65) = 0.9505364$

(Excel, R and Python → 0.9505285)

# STUDENT'S T-DISTRIBUTION

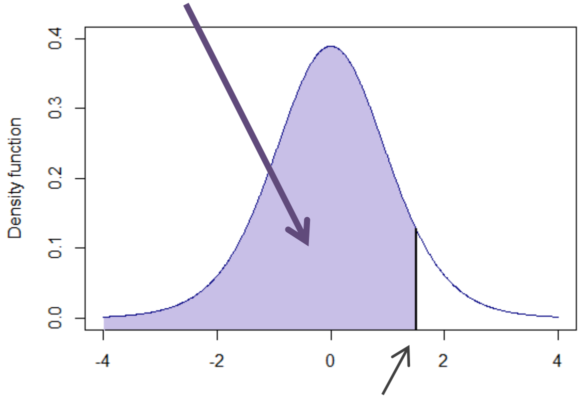
CDF of Student's t-distribution with k (k > 0) degrees of freedom.

Probability of less than t = 1.5 with k = 10.

Probability density function

$$f_k(t) = \frac{1}{\sqrt{k\pi}} \frac{\Gamma\left(\frac{k+1}{2}\right)}{\Gamma\left(\frac{k}{2}\right)} \left(1 + \frac{t^2}{k}\right)^{-\frac{k+1}{2}}$$

0.9177463




t = 1.5

EXCEL

TRUE, cumulative distribution function. If FALSE, returns the probability density function. Required

```
T.DIST(1.5,10,TRUE)
1 - T.DIST.RT(1.5,10)
```

We can use also the probability of more than t = 1.5

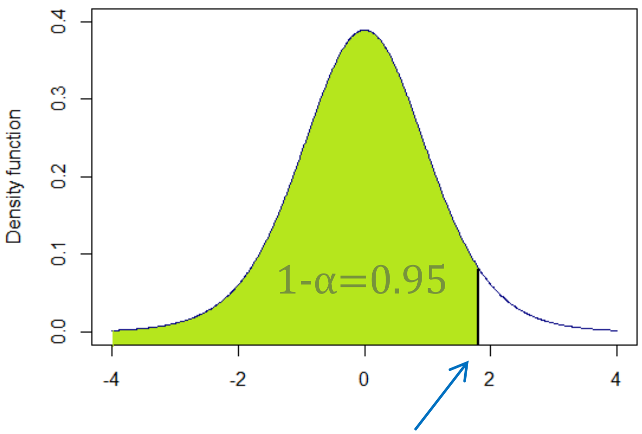
R

```
pt(1.5,df=10,lower.tail=TRUE)
1 - pt(1.5,df=10,lower.tail=FALSE)
```

Python

```
stats.t.cdf(1.5,df=10)
```

PPF (q) of the Student's t-distribution with k = 10 degrees of freedom for the probability (1 – α) = 0.95




**q = 1.812461**

EXCEL

```
T.INV(0.95,10)
```

---

R

```
qt(0.95,df=10,lower.tail=TRUE)
qt(0.05,df=10,lower.tail=FALSE)
```

---

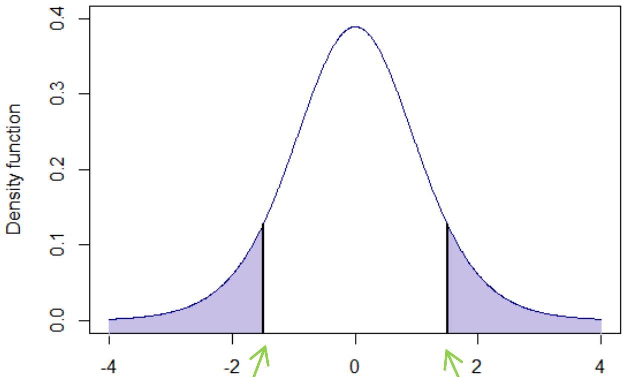
Python

```
stats.t.ppf(0.95,df=10)
```

CDF and PPF for two-tailed Student's t-distribution.

EXCEL provides two specific functions.

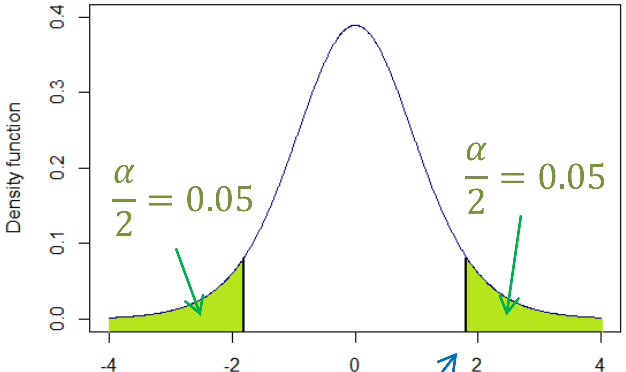
**p-value = 2 * 0.08225366 = 0.16450733**




-|1.5|  +|1.5|

ABS() "absolute value" function. Essential if the test statistic takes a negative value.

T.DIST.2T(ABS(1.5),10)




$\frac{\alpha}{2} = 0.05$  $\frac{\alpha}{2} = 0.05$

**q = 1.812461**

T.INV.2T(0.1,10)

$\alpha = 0.1$

# CHI-SQUARED DISTRIBUTION

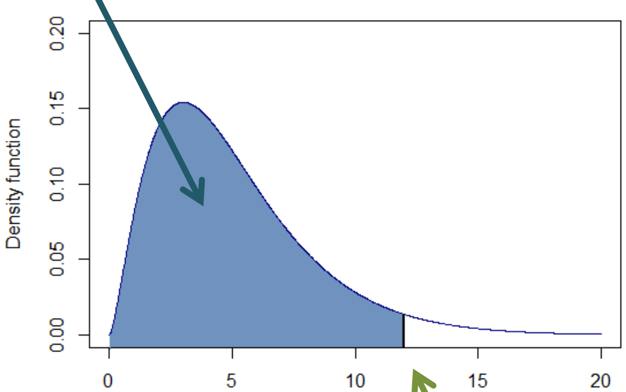
CDF of the CHI-SQUARED distribution with k (k > 0) degrees of freedom.

Probability of less than t = 12.0 with k = 5.

Probability density function of $\chi^2$

$$f_k(t) = \frac{1}{2^{\frac{k}{2}}\,\Gamma\left(\frac{k}{2}\right)} t^{\frac{k}{2}-1} e^{-\frac{t}{2}}$$

**0.9652122**




**t = 12.0**

TRUE, cumulative distribution function. If FALSE, returns the probability density function. Required

## EXCEL

CHISQ.DIST(12.0,5,TRUE)

1 – CHISQ.DIST.RT(12.0,5)

We can use also the probability of more than t = 12.0

## R

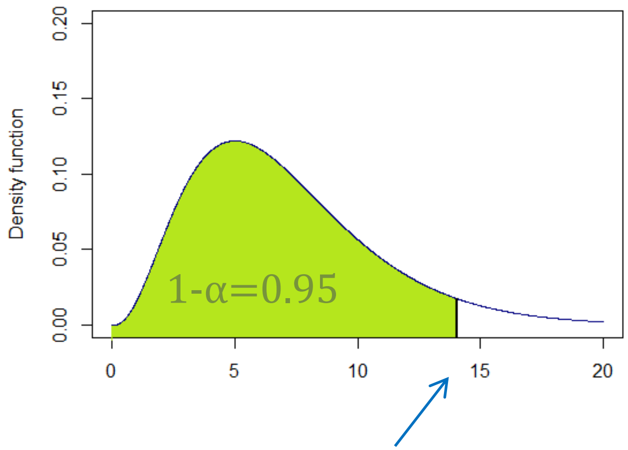
pchisq(12.0,df=5)

1 - pchisq(12.0,df=5,lower.tail=FALSE)

## Python

stats.chi2.cdf(12.0,df=5)

PPF (q) of the chi-squared distribution with k = 7 degrees of freedom for the probability (1 – α) = 0.95




**q = 14.06714**

## EXCEL

CHISQ.INV (0.95,7)

CHISQ.INV.RT (0.05,7)

---

## R

```
qchsiq(0.95,df=7)
qchisq(0.05,df=7,lower.tail=FALSE)
```

---

## Python

```
stats.chi2.ppf(0.95, df=7)
```

# FISHER-SNEDECOR DISTRIBUTION

CDF of F-distribution with d1 (d1 > 0) and d2 (d2 > 0) degrees of freedom.

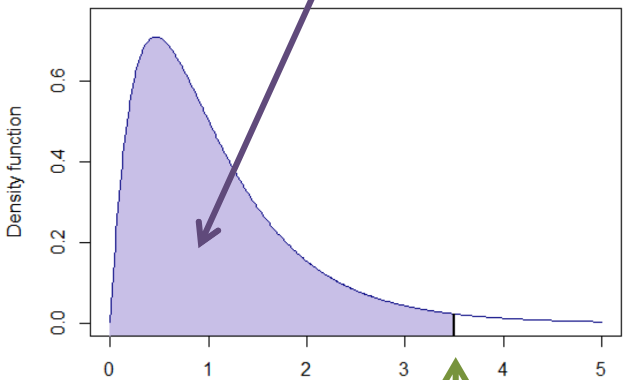
Probability of less than x = 3.5 with (d1 = 4, d2 = 26).

Probability density function

$$f(x) = \frac{\left(\frac{d_1 x}{d_1 x + d_2}\right)^{\frac{d_1}{2}} \left(1 - \frac{d_1 x}{d_1 x + d_2}\right)^{\frac{d_2}{2}}}{x \, \mathrm{B}\left(\frac{d_1}{2}, \frac{d_2}{2}\right)}$$

B() is the beta function

0.97948051




x = 3.5

TRUE, cumulative distribution function. If FALSE, returns the probability density function. Required

EXCEL

F.DIST(3.5,4,26,TRUE)

1 - F.DIST.RT(3.5,4,26)

We can use also the probability of more than x = 3.5

R

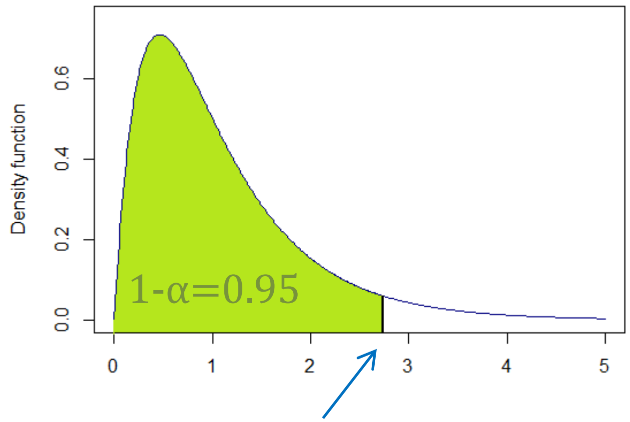
pf(3.5,df1=4,df2=26)

1 - pf(3.5,df1=4,df2=26,lower.tail=FALSE)

Python

stats.f.cdf(3.5,dfn=4,dfd=26)

PPF (q) of the F-Distribution with (d1 = 4, d2 = 26) degrees of freedom for the probability (1 – α) = 0.95




**q = 2.742594**

## EXCEL

F.INV(0.95,4,26)

F.INV.RT(0.05,4,26)

---

## R

qf(0.95,df1=4,df2=26)

qf(0.05,df1=4,df2=26,lower.tail=FALSE)

---

## Python

stats.f.ppf(0.95,dfn=4,dfd=26)

# References

# References

Scipy.org – Statistical functions (scipy.stats)

https://docs.scipy.org/doc/scipy/reference/stats.html

Microsoft – Excel Statistical Functions

https://support.office.com/en-us/article/Statistical-functions-reference-624DAC86-A375-4435-BC25-76D659719FFD

R Tutorial – Basic Probability Distributions

http://www.cyclismo.org/tutorial/R/probability.html