

# One and Two Samples Using Only an R Function \*

Ying-Ying Zhang  
Chongqing University

---

## Abstract

We create an R function `one_two_sample()` which deals with one and two (normal) samples. For one normal sample  $\mathbf{x}$ , the function reports descriptive statistics, plot, interval estimations and hypothesis testings of the means and variances of  $\mathbf{x}$ . For one abnormal sample  $\mathbf{x}$ , the function reports descriptive statistics, plot, two sided interval estimation of the mean of  $\mathbf{x}$ . For two normal samples  $\mathbf{x}$  and  $\mathbf{y}$ , the function reports descriptive statistics, plot, interval estimations and hypothesis testings of the means and variances of  $\mathbf{x}$  and  $\mathbf{y}$ , respectively. It also reports interval estimations and hypothesis testings of the difference of the means of  $\mathbf{x}$  and  $\mathbf{y}$  and the ratio of the variances of  $\mathbf{x}$  and  $\mathbf{y}$ , tests whether  $\mathbf{x}$  and  $\mathbf{y}$  are from the same population, finds the correlation coefficient of  $\mathbf{x}$  and  $\mathbf{y}$  if they have the same length. The function is in a created R package **OneTwoSamples** which is available on CRAN.

*Keywords:* one and two samples, interval estimation, hypothesis testing, mean, variance, R.

---

## 1. Introduction

R software (R Development Core Team 2013) has become more and more popular among researchers due to its freeness, handy and powerful programming language, coherent statistical analysis tools, superior statistical charting and many other advantages. The extensive data from industrial productions, financial economics, medical experiments and many other aspects, may result in one or two samples. First, we are interested in whether it is or they are normal. For one normal sample  $\mathbf{x}$ , we are further interested in its descriptive statistics, plots (the histogram, the empirical cumulative distribution function (ECDF), the QQ plot), interval estimations and hypothesis testings of the means and variances of  $\mathbf{x}$ . For two normal samples  $\mathbf{x}$  and  $\mathbf{y}$ , except for the descriptive statistics, plots, interval estimations and hypothesis testings of the means and variances of  $\mathbf{x}$  and  $\mathbf{y}$ , respectively. We are also interested in interval estimations and hypothesis testings of the difference of the means of  $\mathbf{x}$  and  $\mathbf{y}$  and the ratio of the variances of  $\mathbf{x}$  and  $\mathbf{y}$ , whether  $\mathbf{x}$  and  $\mathbf{y}$  are from the same population, and the correlation coefficient of  $\mathbf{x}$  and  $\mathbf{y}$  if they have the same length. All these interested information can be obtained by implementing one R function `one_two_sample()`, which is in a created R package **OneTwoSamples** available on CRAN (Zhang 2013).

Statistical inferences are main contents of mathematical statistics. Parametric estimation and hypothesis testing are two classical methods widely used in statistical inferences. They are treated in many statistics textbooks (Casella and Berger 2002; DeCoursey 2003; Freedman *et al.* 2007; McClave *et al.* 2008; Ross 2009; Soong 2004; Walpole *et al.* 2011; Xue and

---

\*The research was supported by Natural Science Foundation Project of CQ CSTC CSTC2011BB0058.

Table 1: The functions used in interval estimations and hypothesis testings of the means and variances of one and two normal samples.

mu	one sample			two samples			
	functions	sigma known	sigma unknown	functions	sigma1,sigma2 known	sigma1,sigma2 (=)unknown	sigma1,sigma2 (!=)unknown
interval estimation	<code>interval_estimate1()</code> (two sided)	✓	✓	<code>interval_estimate2()</code> (two sided)	✓	✓	✓
	<code>interval_estimate4()</code>	✓	✓	<code>interval_estimate5()</code>	✓	✓	✓
	<code>t.test()</code>	✗	✓	<code>t.test()</code>	✗	✓	✓
hypothesis testing	<code>mean_test1()</code>	✓	✓	<code>mean_test2()</code>	✓	✓	✓
	<code>t.test()</code>	✗	✓	<code>t.test()</code>	✗	✓	✓
sigma	functions	mu known	mu unknown	functions	mu1 & mu2 known	mu1 or mu2 unknown	
interval estimation	<code>interval_var1()</code> (two sided)	✓	✓	<code>interval_var2()</code> (two sided)	✓	✓	
	<code>interval_var3()</code>	✓	✓	<code>interval_var4()</code>	✓	✓	
				<code>var.test()</code>	✗	✓	
hypothesis testing	<code>var_test1()</code>	✓	✓	<code>var_test2()</code>	✓	✓	
				<code>var.test()</code>	✗	✓	

Chen 2007; Yang *et al.* 2004). It is well known that the R built-in function `t.test()` can implement the interval estimation and hypothesis testing of one and two normal populations' mean. However, `t.test()` can neither accomplish those of the one normal population's mean when the population's variance is known, nor accomplish those of the two normal populations' mean when the populations' variances are known. Another R built-in function, `var.test()`, can implement the interval estimation and hypothesis testing of two normal populations' variances. However, `var.test()` can neither accomplish those of the one normal population's variance, nor accomplish those of the two normal populations' variances when the populations' means are known. Xue and Chen (2007) write twelve functions to implement all the interval estimations and hypothesis testings of the means and variances of one and two normal populations. See Table 1. In the table, the functions with blue text are superior to others since they still work when `mu` or `sigma` is known. '✓' denotes the function can handle this case, while '✗' indicates it can not. Most of the functions can compute both one and two sided interval estimation and hypothesis testing except for those marked with 'two sided'. The functions listed above are applicable for normal sample(s). As for an abnormal sample, `interval_estimate3()` can implement the two sided interval estimation of the mean no matter the variance is known or not.

However, it is burdensome to remember and apply the functions in Table 1 in a flexible way. Zhang and Wei (2013) integrate them into one R function `IntervalEstimate_TestOfHypothesis()`. Users only need to input the sample(s) and set the parameters (`test`, `mu`, `sigma`, `var.equal`, `ratio`, `side`, `alpha`) as needed. It is convenient for users who merely care about the interval estimation and hypothesis testing of the mean or variance. The function `one_two_sample()` differs from `IntervalEstimate_TestOfHypothesis()` in many ways. See Table 2.

## 2. R function: `one_two_sample()`

The function `one_two_sample()` deals with one or two (normal) samples. In this section, we will introduce the program flowchart and usage of `one_two_sample()` in detail.

Table 2: Differences between two functions.

	<code>one_two_sample()</code>	<code>IntervalEstimate_TestOfHypothesis()</code>
Orientation	Deal with one or two (normal) samples. Report descriptive statistics, plots, interval estimations and hypothesis testings of the means and variances of one or two normal samples. For two samples, test whether $x$ and $y$ are from the same population, find the correlation coefficient of $x$ and $y$ if they have the same length.	Implement interval estimation and hypothesis testing of the mean or variance of one or two normal samples.
Outputs of interval estimation and hypothesis testing	One normal sample: Interval estimation and hypothesis testing of $\mu$ AND $\sigma$ .  Two normal samples: Interval estimation and hypothesis testing of $\mu$ AND $\sigma$ of $x$ and $y$ , respectively. Interval estimations and hypothesis testings of the difference of the means of $x$ and $y$ AND the ratio of the variances of $x$ and $y$ .	One normal sample: Interval estimation and hypothesis testing of $\mu$ OR $\sigma$ .  Two normal samples: Interval estimation and hypothesis testing of the difference of the means of $x$ and $y$ OR the ratio of the variances of $x$ and $y$ .
Call functions of interval estimation and hypothesis testing	Directly call the following functions according to the input parameters: <code>interval_estimate4()</code> , <code>interval_estimate5()</code> , <code>mean_test1()</code> , <code>mean_test2()</code> , <code>interval_var3()</code> , <code>interval_var4()</code> , <code>var_test1()</code> , <code>var_test2()</code> , <code>t.test()</code> , <code>var.test()</code> .	Make up the following four functions, and call them according to the input parameters: <code>single_mean()</code> , <code>single_var()</code> , <code>double_mean()</code> , <code>double_var()</code> .
Availability	An R package <b>OneTwoSamples</b> available on CRAN.	Through email to the author.

## 2.1. Program flowchart

To make the structure of the R function `one_two_sample()` easy to understand, we draw a program flowchart with Microsoft Office Visio 2007. See Figure 1.

## 2.2. Usage

The usage of `one_two_sample()` is as follows:

```
one_two_sample(x, y = NULL, mu = c(Inf, Inf), sigma = c(-1, -1),
              var.equal = FALSE, ratio = 1, side=0, alpha=0.05)
```

**x**: a (non-empty) numeric vector of sample data values.

**y**: an optional non-empty numeric vector of sample data values. The default is `NULL`, i.e., there is only one sample. In this case, we can also use the function `one_sample()`.

**mu**: if `y = NULL`, i.e., there is only one sample. See the argument `mu` in `one_sample()`. For two normal samples `x` and `y`, `mu` plays one role: the population means. However, `mu` is used in two places: one is the two sided or one sided interval estimation of  $\sigma_1^2 / \sigma_2^2$  of two normal samples, another is the two sided or one sided hypothesis testing of  $\sigma_1^2$  and  $\sigma_2^2$  of two normal samples. When `mu` is known, input it, and the function computes the interval endpoints (or the F value) using an F distribution with degree of freedom (`n1`, `n2`). When it is unknown, ignore it, and the function computes the interval endpoints (or the F value) using an F distribution with degree of freedom (`n1 - 1`, `n2 - 1`).

**sigma**: if `y = NULL`, i.e., there is only one sample. See the argument `sigma` in `one_sample()`. For two normal samples `x` and `y`, `sigma` plays one role: the population standard deviations. However, `sigma` is used in two places: one is the two sided or one sided interval estimation of  $\mu_1 - \mu_2$  of two normal samples, another is the two sided or one sided hypothesis testing of  $\mu_1$  and  $\mu_2$  of two normal samples. When the standard deviations are known, input it, then the function computes the interval endpoints using normal population; when the standard deviations are unknown, ignore it, now we need to consider whether the two populations have equal variances. See `var.equal` below.

**var.equal**: a logical variable indicating whether to treat the two variances as being equal. If `TRUE` then the pooled variance is used to estimate the variance otherwise the Welch (or Satterthwaite) approximation to the degrees of freedom is used.

**ratio**: the hypothesized ratio of the population variances of `x` and `y`. It is used in `var.test(x, y, ratio = ratio, ...)`, i.e., when computing the interval estimation and hypothesis testing of  $\sigma_1^2 / \sigma_2^2$  when `mu1` or `mu2` is unknown.

**side**: if `y = NULL`, i.e., there is only one sample. See the argument `side` in `one_sample()`. For two normal samples `x` and `y`, `sigma` is used in four places: interval estimation of  $\mu_1 - \mu_2$ , hypothesis testing of  $\mu_1$  and  $\mu_2$ , interval estimation of  $\sigma_1^2 / \sigma_2^2$ , hypothesis testing of  $\sigma_1^2$  and  $\sigma_2^2$ . In interval estimation of  $\mu_1 - \mu_2$  or  $\sigma_1^2 / \sigma_2^2$ , `side` is a parameter used to control whether to compute two sided or one sided interval estimation. When computing the one sided upper limit, input `side = -1` (or a number  $< 0$ ); when computing the one sided lower limit, input `side = 1` (or a number  $> 0$ ); when computing the two sided limits, input `side = 0` (default). In hypothesis testing of  $\mu_1$  and  $\mu_2$  or  $\sigma_1^2$  and  $\sigma_2^2$ , `side` is a parameter used to control two sided or one sided hypothesis testing. When inputting `side = 0` (default), the function computes two sided hypothesis testing, and  $H_1: \mu_1 \neq \mu_2$  or  $H_1: \sigma_1^2 \neq \sigma_2^2$ ; when inputting

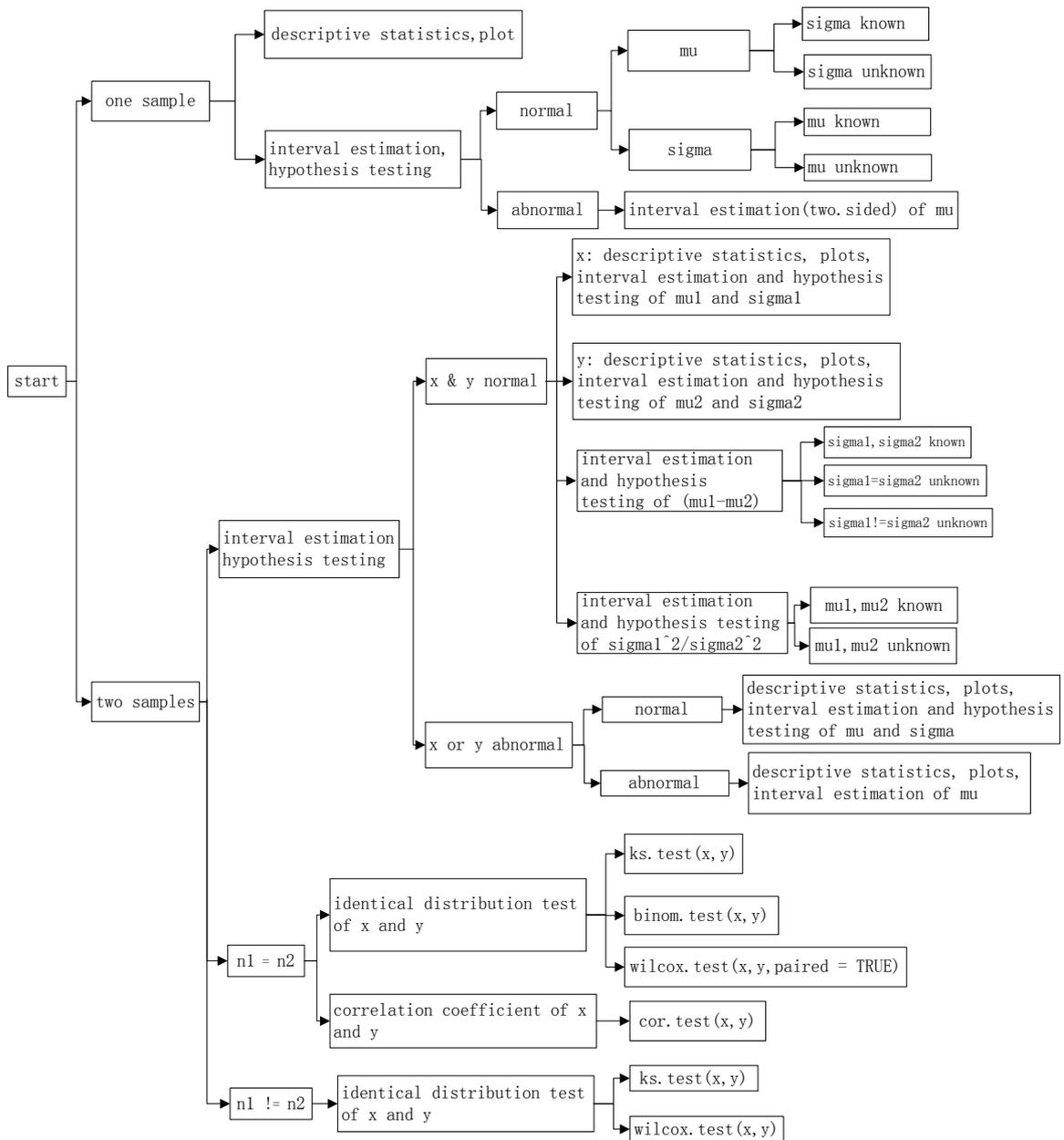


Figure 1: The program flowchart of `one_two_sample()`.

`side = -1` (or a number  $< 0$ ), the function computes one sided hypothesis testing, and  $H_1: \mu_1 < \mu_2$  or  $H_1: \sigma_1^2 < \sigma_2^2$ ; when inputting `side = 1` (or a number  $> 0$ ), the function computes one sided hypothesis testing, and  $H_1: \mu_1 > \mu_2$  or  $H_1: \sigma_1^2 > \sigma_2^2$ .

`alpha`: the significance level, a real number in  $[0, 1]$ . Default to 0.05.  $1 - \alpha$  is the degree of confidence.

In Table 3, we further illustrate the usage of `one_two_sample()` by examples. All the examples are implemented in `'tests_OneTwoSamples.R'` in the `'tests'` folder of the package **OneTwoSamples**. In the table, each cell is divided into two parts. The upper part illustrates the usage of input parameters, and the lower part lists the functions called by `one_two_sample()`.

### 2.3. Practical application

As mentioned in Table 2, `one_two_sample()` call other functions according to the input parameters. Thus the validity of `one_two_sample()` relies on those functions. In this section, we apply the function `one_two_sample()` to a dataset `women` in the `datasets` package.

To use the function `one_two_sample()`, we should first: `library('OneTwoSamples')`. Note: the outputs explanations of a specific function can be obtained through the help page, for example, `'?data_outline'`, `'?t.test()'`.

```
> ## generate samples x and y
> x = women$height; x

[1] 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72

> y = women$weight; y

[1] 115 117 120 123 126 129 132 135 139 142 146 150 154 159 164

> ## operate on one sample
> ## one_two_sample(x) is equivalent to one_sample(x)
> one_two_sample(x)

quantile of x
 0% 25% 50% 75% 100%
58.0 61.5 65.0 68.5 72.0
data_outline of x
  N Mean Var std_dev Median std_mean CV CSS USS R R1 Skewness
1 15 65 20 4.472136 65 1.154701 6.880209 280 63655 14 7 0
  Kurtosis
1 -1.2

Shapiro-Wilk normality test

data: x
W = 0.96359, p-value = 0.7545
```

Table 3: The usage of `one_two_sample()`.

<b>One normal sample</b>	sigma known	sigma unknown
mu known	Example 1: x, mu =, sigma =, side = 0, alpha = 0.05	Example 3: x, mu =, side = 0, alpha = 0.05
	<code>interval_estimate4()</code> , <code>mean_test1()</code> , <code>interval_var3()</code> , <code>var_test1()</code>	<code>t.test()</code> , <code>interval_var3()</code> , <code>var_test1()</code>
mu unknown	Example 2: x, sigma =, side = 0, alpha = 0.05	Example 4: x, side = 0, alpha = 0.05
	<code>interval_estimate4()</code> , <code>mean_test1()</code> , <code>interval_var3()</code> , <code>var_test1()</code>	<code>t.test()</code> , <code>interval_var3()</code> , <code>var_test1()</code>
<b>One abnormal sample</b>	Example 5: x, sigma =, alpha = 0.05	Example 6: x, alpha = 0.05
	<code>interval_estimate3()</code>	<code>interval_estimate3()</code>
<b>Two normal samples</b>	mu1, mu2 known	mu1, mu2 unknown
sigma1, sigma2 known	Example 7: x, y, mu = c(), sigma = c(), side = 0, alpha = 0.05	Example 10: x, y, ratio = 1, sigma = c(), side = 0, alpha = 0.05
	<code>interval_estimate5()</code> , <code>mean_test2()</code> , <code>interval_var4()</code> , <code>var_test2()</code>	<code>interval_estimate5()</code> , <code>mean_test2()</code> , <code>var.test()</code>
sigma1 = sigma2 unknown	Example 8: x, y, mu = c(), var.equal = TRUE, side = 0, alpha = 0.05	Example 11: x, y, ratio = 1, var.equal = TRUE, side = 0, alpha = 0.05
	<code>t.test()</code> , <code>interval_var4()</code> , <code>var_test2()</code>	<code>t.test()</code> , <code>var.test()</code>
sigma1 != sigma2 unknown	Example 9: x, y, mu = c(), side = 0, alpha = 0.05	Example 12: x, y, ratio = 1, side = 0, alpha = 0.05
	<code>t.test()</code> , <code>interval_var4()</code> , <code>var_test2()</code>	<code>t.test()</code> , <code>var.test()</code>

The data is from the normal population.

The data is from the normal population.

Interval estimation and test of hypothesis of mu  
 Interval estimation and test of hypothesis: t.test()  
 H0:  $\mu = 0$       H1:  $\mu \neq 0$

#### One Sample t-test

data: x  
 t = 56.292, df = 14, p-value < 2.2e-16  
 alternative hypothesis: true mean is not equal to 0  
 95 percent confidence interval:  
 62.52341 67.47659  
 sample estimates:  
 mean of x  
 65

Interval estimation and test of hypothesis of sigma  
 Interval estimation: interval\_var3()  
 var df      a      b  
 1 20 14 10.72019 49.74483  
 Test of hypothesis: var\_test1()  
 H0:  $\sigma^2 = 1$       H1:  $\sigma^2 \neq 1$   
 var df chisq2 P\_value  
 1 20 14 280 0

> ## one\_two\_sample(y) is equivalent to one\_sample(y)  
 > one\_two\_sample(y)

quantile of y  
 0% 25% 50% 75% 100%  
 115.0 124.5 135.0 148.0 164.0  
 data\_outline of y  

	N	Mean	Var	std_dev	Median	std_mean	CV	CSS	USS	R
1	15	136.7333	240.2095	15.49869	135	4.001746	11.33498	3362.933	283803	49

R1 Skewness Kurtosis  
 1 23.5 0.2814297 -1.040715

#### Shapiro-Wilk normality test

data: y  
 W = 0.96036, p-value = 0.6986

The data is from the normal population.

The data is from the normal population.

Interval estimation and test of hypothesis of mu  
 Interval estimation and test of hypothesis: `t.test()`  
 H0:  $\mu = 0$       H1:  $\mu \neq 0$

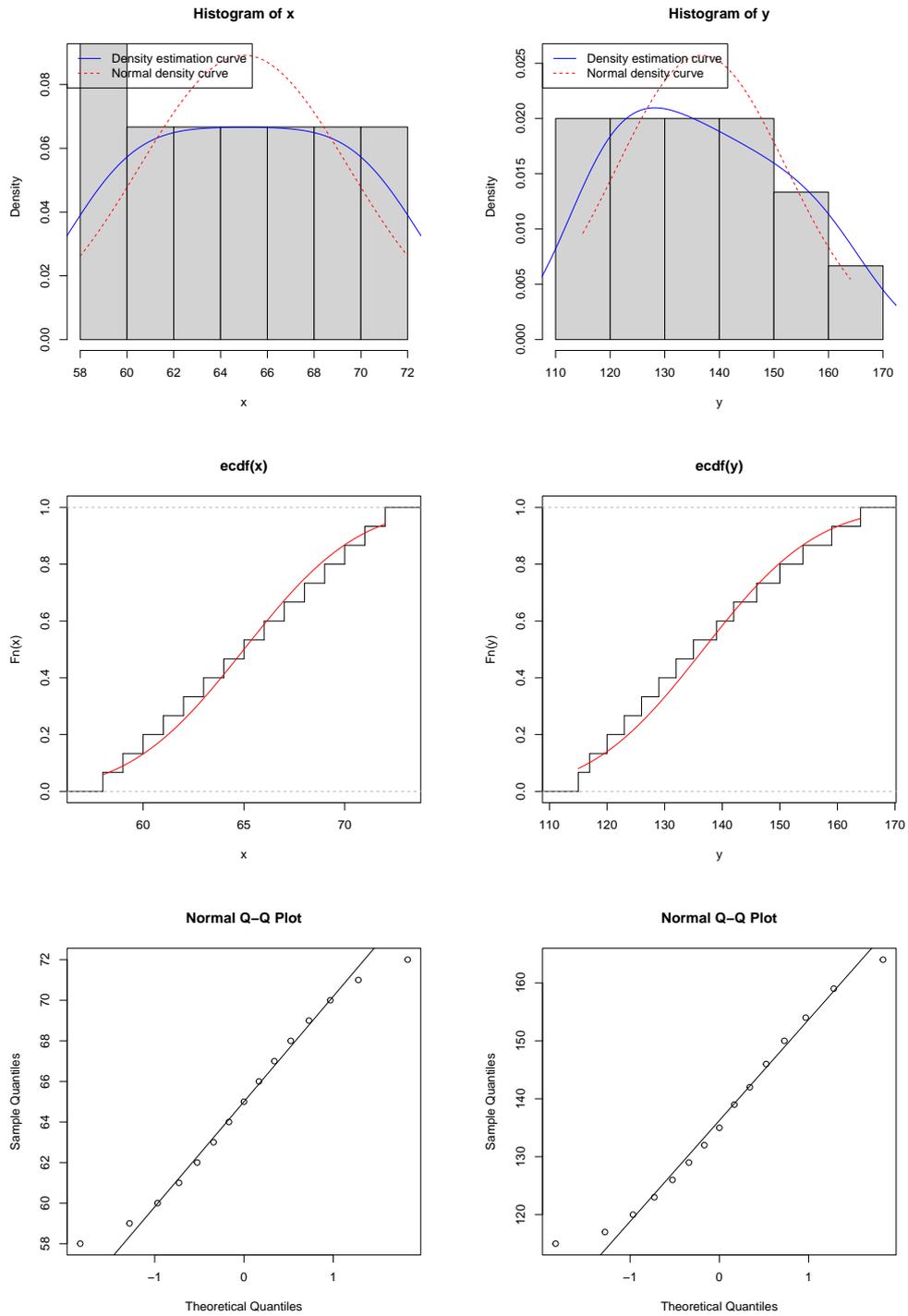
#### One Sample t-test

```
data: y
t = 34.168, df = 14, p-value = 6.907e-15
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 128.1504 145.3162
sample estimates:
mean of x
 136.7333
```

```
Interval estimation and test of hypothesis of sigma
Interval estimation: interval_var3()
      var df      a      b
1 240.2095 14 128.7545 597.459
Test of hypothesis: var_test1()
H0: sigma2 = 1      H1: sigma2 != 1
      var df  chisq2 P_value
1 240.2095 14 3362.933      0
```

Illustration: The outputs of `one_two_sample(x)` and `one_two_sample(y)` are listed above. For `x`, first the function reports descriptive statistics (the quantile of `x` and the `data_outline` of `x`). Then in Shapiro-Wilk normality test,  $p\text{-value} = 0.7545 > 0.05$ , so the data `x` is from the normal population. After that, the 3 plots in the left column of Figure 2 show the histogram, the ECDF, and the QQ plot of `x`, respectively. The 3 plots all indicate that the data `x` is from the normal population, in agree with the result of Shapiro-Wilk normality test. Finally, the function displays interval estimations and hypothesis testings of the means and variances of `x`. The interval estimation and hypothesis testing of `mu` call the function `t.test()`. We find that the 95 percent confidence interval of `mu` is  $[62.52341, 67.47659]$ , the  $p\text{-value} < 2.2e-16 < 0.05$ , so reject H0:  $\mu = 0$  and accept H1:  $\mu \neq 0$ . The interval estimation of `sigma` calls the function `interval_var3()`. We find that the 95 percent confidence interval of `sigma` is  $[10.72019, 49.74483]$ . The hypothesis testing of `sigma` calls the function `var_test1()`. We find that the  $P\text{-value} = 0 < 0.05$ , so reject H0:  $\sigma^2 = 1$  and accept H1:  $\sigma^2 \neq 1$ . The explanations of the outputs of `one_two_sample(y)` are omitted.

```
> ## operate on two samples
```

Figure 2: Histogram, ECDF, and QQ plot of  $x$  and  $y$ .

```
> one_two_sample(x, y)
```

```
Interval estimation and test of hypothesis
```

```
Shapiro-Wilk normality test
```

```
data: x
W = 0.96359, p-value = 0.7545
```

```
Shapiro-Wilk normality test
```

```
data: y
W = 0.96036, p-value = 0.6986
```

```
x and y are both from the normal populations.
```

```
x: descriptive statistics, plot, interval estimation and test of hypothesis
quantile of x
```

```
0% 25% 50% 75% 100%
58.0 61.5 65.0 68.5 72.0
```

```
data_outline of x
```

	N	Mean	Var	std_dev	Median	std_mean	CV	CSS	USS	R	R1	Skewness
1	15	65	20	4.472136	65	1.154701	6.880209	280	63655	14	7	0

Kurtosis

1	-1.2
---	------

```
Shapiro-Wilk normality test
```

```
data: x
W = 0.96359, p-value = 0.7545
```

```
The data is from the normal population.
```

```
The data is from the normal population.
```

```
Interval estimation and test of hypothesis of mu
Interval estimation and test of hypothesis: t.test()
H0: mu = 0      H1: mu != 0
```

```
One Sample t-test
```

```
data: x
t = 56.292, df = 14, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
```

95 percent confidence interval:

62.52341 67.47659

sample estimates:

mean of x

65

Interval estimation and test of hypothesis of sigma

Interval estimation: interval\_var3()

var	df	a	b
1	20	14	10.72019 49.74483

Test of hypothesis: var\_test1()

H0: sigma2 = 1 H1: sigma2 != 1

var	df	chisq2	P_value
1	20	14	280 0

y: descriptive statistics, plot, interval estimation and test of hypothesis

quantile of y

%	25%	50%	75%	100%
115.0	124.5	135.0	148.0	164.0

data\_outline of y

	N	Mean	Var	std_dev	Median	std_mean	CV	CSS	USS	R
1	15	136.7333	240.2095	15.49869	135	4.001746	11.33498	3362.933	283803	49

	R1	Skewness	Kurtosis
1	23.5	0.2814297	-1.040715

Shapiro-Wilk normality test

data: y

W = 0.96036, p-value = 0.6986

The data is from the normal population.

The data is from the normal population.

Interval estimation and test of hypothesis of mu

Interval estimation and test of hypothesis: t.test()

H0: mu = 0 H1: mu != 0

One Sample t-test

data: y

t = 34.168, df = 14, p-value = 6.907e-15

alternative hypothesis: true mean is not equal to 0

95 percent confidence interval:

128.1504 145.3162

sample estimates:

mean of x  
136.7333

Interval estimation and test of hypothesis of sigma

Interval estimation: interval\_var3()

	var	df	a	b
1	240.2095	14	128.7545	597.459

Test of hypothesis: var\_test1()

H0: sigma2 = 1      H1: sigma2 != 1

	var	df	chisq2	P_value
1	240.2095	14	3362.933	0

Interval estimation and test of hypothesis of mu1-mu2

Interval estimation and test of hypothesis: t.test()

Welch Two Sample t-test

data: x and y

t = -17.223, df = 16.315, p-value = 6.826e-12

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

-80.54891 -62.91775

sample estimates:

mean of x	mean of y
65.0000	136.7333

Interval estimation and test of hypothesis of sigma1^2/sigma2^2

Interval estimation and test of hypothesis: var.test()

F test to compare two variances

data: x and y

F = 0.083261, num df = 14, denom df = 14, p-value = 3.586e-05

alternative hypothesis: true ratio of variances is not equal to 1

95 percent confidence interval:

0.02795306 0.24799912

sample estimates:

ratio of variances
0.08326065

n1 == n2

Test whether x and y are from the same population

H0: x and y are from the same population (without significant difference)  
`ks.test(x,y)`

Exact two-sample Kolmogorov-Smirnov test

data: x and y  
 D = 1, p-value = 1.289e-08  
 alternative hypothesis: two-sided

`binom.test(sum(x<y), length(x))`

Exact binomial test

data: `sum(x < y)` and `length(x)`  
 number of successes = 15, number of trials = 15, p-value = 6.104e-05  
 alternative hypothesis: true probability of success is not equal to 0.5  
 95 percent confidence interval:  
 0.7819806 1.0000000  
 sample estimates:  
 probability of success  
 1

`wilcox.test(x, y, alternative = alternative, paired = TRUE)`

Wilcoxon signed rank exact test

data: x and y  
 V = 0, p-value = 6.104e-05  
 alternative hypothesis: true location shift is not equal to 0

Find the correlation coefficient of x and y  
 H0:  $\rho = 0$  (x, y uncorrelated)

Pearson's product-moment correlation

data: x and y  
 t = 37.855, df = 13, p-value = 1.091e-14  
 alternative hypothesis: true correlation is not equal to 0  
 95 percent confidence interval:  
 0.9860970 0.9985447  
 sample estimates:  
 cor  
 0.9954948

Kendall's rank correlation tau

```

data:  x and y
T = 105, p-value = 1.529e-12
alternative hypothesis: true tau is not equal to 0
sample estimates:
tau
  1

```

Spearman's rank correlation rho

```

data:  x and y
S = 1.2434e-13, p-value < 2.2e-16
alternative hypothesis: true rho is not equal to 0
sample estimates:
rho
  1

```

Illustration: The outputs of `one_two_sample(x, y)` are listed above. The explanations for the former parts of the outputs of `one_two_sample(x, y)` are omitted since they have been listed in the outputs of `one_two_sample(x)` and `one_two_sample(y)`. The interval estimation and hypothesis testing of  $\mu_1 - \mu_2$  call the function `t.test()`. We find that the 95 percent confidence interval of  $\mu_1 - \mu_2$  is  $[-80.54891, -62.91775]$ , the p-value =  $6.826e-12 < 0.05$ , so reject  $H_0: \mu_1 = \mu_2$  and accept  $H_1: \mu_1 \neq \mu_2$ . The interval estimation and hypothesis testing of  $\sigma_1^2 / \sigma_2^2$  call the function `var.test()`. We find that the 95 percent confidence interval of  $\sigma_1^2 / \sigma_2^2$  is  $[0.02795306, 0.24799912]$ , the p-value =  $3.586e-05 < 0.05$ , so reject  $H_0: \sigma_1^2 = \sigma_2^2$  and accept  $H_1: \sigma_1^2 \neq \sigma_2^2$ . We obtain  $n_1 == n_2$ , i.e.,  $x$  and  $y$  have the same length. Three functions `ks.test()`, `binom.test()`, and `wilcox.test()` are used to test whether  $x$  and  $y$  are from the same population. Three p-values are all less than 0.05, so reject  $H_0: x$  and  $y$  are from the same population. The function `cor.test(x, y, method = c('pearson', 'kendall', 'spearman'))` is used to find the correlation coefficient of  $x$  and  $y$ . Three p-values are all less than 0.05, so reject  $H_0: \rho = 0$  ( $x, y$  uncorrelated). Thus  $x$  and  $y$  are correlated. In fact,  $x$  and  $y$  have nearly 1 correlation.

### 3. Conclusions

The function `one_two_sample()` can deal with one and two (normal) samples. For one normal sample  $x$ , the function reports descriptive statistics, plot, interval estimations and hypothesis testings of the means and variances of  $x$ . For one abnormal sample  $x$ , the function reports descriptive statistics, plot, two sided interval estimation of the mean of  $x$ . For two normal samples  $x$  and  $y$ , the function reports descriptive statistics, plot, interval estimations and hypothesis testings of the means and variances of  $x$  and  $y$ , respectively. It also reports interval estimations and hypothesis testings of the difference of the means of  $x$  and  $y$  and the ratio of the variances of  $x$  and  $y$ , tests whether  $x$  and  $y$  are from the same population, finds the correlation coefficient of  $x$  and  $y$  if they have the same length. The function is in

a created R package **OneTwoSamples** which is available on CRAN. In addition, the usage of arguments of `one_two_sample()` is straightforward. It will simplify the users' operations of dealing with one and two (normal) samples to a great extent.

## Acknowledgements

## References

- Casella G, Berger RL (2002). *Statistical Inference*. Duxbury, United States. 2nd edition.
- DeCoursey W (2003). *Statistics and Probability for Engineering Applications with MS Excel*. Newnes, New York.
- Freedman D, Pisani R, Purves R (2007). *Statistics*. W. W. Norton & Company, New York. 4th edition.
- McClave JT, Benson PG, Sincich T (2008). *A First Course in Business Statistics*. Prentice Hall, New Jersey. 8th edition.
- R Development Core Team (2013). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL <http://www.R-project.org/>.
- Ross S (2009). *Introduction to Probability and Statistics for Engineers and Scientists*. Elsevier Academic Press, USA. 4th edition.
- Soong TT (2004). *Fundamentals of Probability and Statistics for Engineers*. John Wiley & Sons, USA.
- Walpole RE, Myers RH, Myers SL, Ye KE (2011). *Probability and Statistics for Engineers and Scientists*. Pearson, New York. 9th edition.
- Xue Y, Chen LP (2007). *Statistical Modeling and R Software*. Tsinghua University Press, Beijing. This is a Chinese book.
- Yang H, Liu QS, Zhong B (2004). *Mathematical Statistics*. China Higher Education Press, Beijing. This is a Chinese book.
- Zhang YY (2013). **OneTwoSamples**: Deal with One and Two (Normal) Samples. R package version 1.0-3, URL <http://CRAN.R-project.org/package=OneTwoSamples>.
- Zhang YY, Wei Y (2013). "Implement All the Interval Estimations and Hypothesis Testings of the Means and Variances of Normal Populations in One R Function." *Statistics and Decision*. This is a Chinese journal, accepted for publication.

**Affiliation:**

Ying-Ying Zhang

Department of Statistics and Actuarial Science

College of Mathematics and Statistics

Chongqing University

Chongqing, China

E-mail: [robertzhangyying@qq.com](mailto:robertzhangyying@qq.com)

URL: <http://baike.baidu.com/view/7694173.htm>