Experimental Design and Data Analysis, Lecture 2

Eduard Belitser

VU Amsterdam

Lecture Overview

1. recap distributions
2. bootstrap confidence intervals
3. statistical tests
4. bootstrap tests

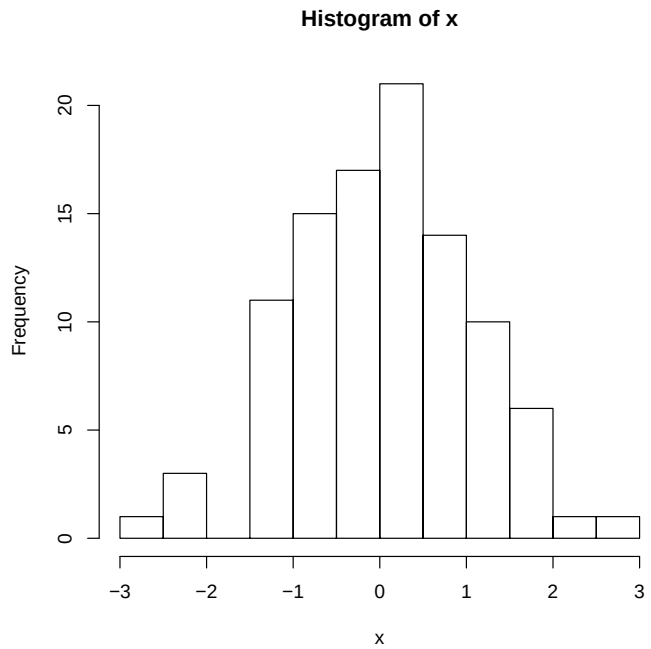
recap distributions

Recap — histogram

The histogram corresponding to numerical measurements $x_1, x_2, ..., x_N$ is a barplot, where the area of the bar over an interval (a, b) corresponds to the fraction

$$\frac{1}{N}\#\big(1 \le n \le N : a \le x_n \le b\big).$$

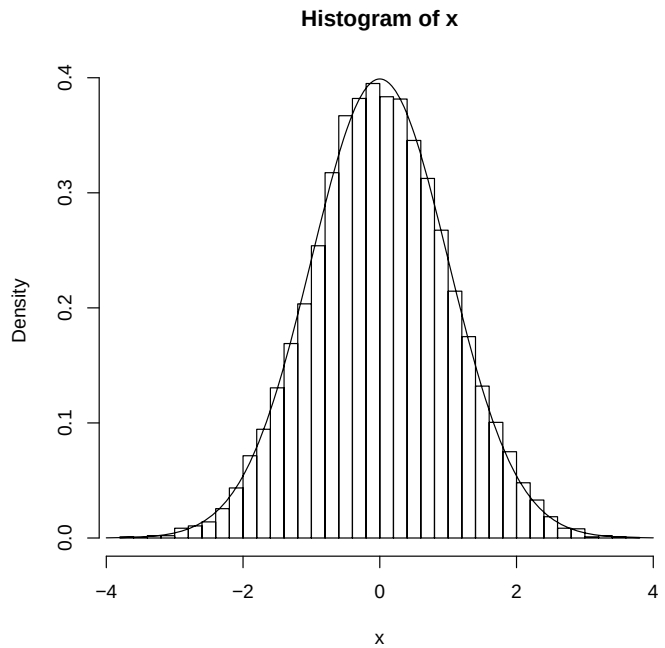
```
> x=rnorm(100)
> hist(x)
```



Recap — population distribution

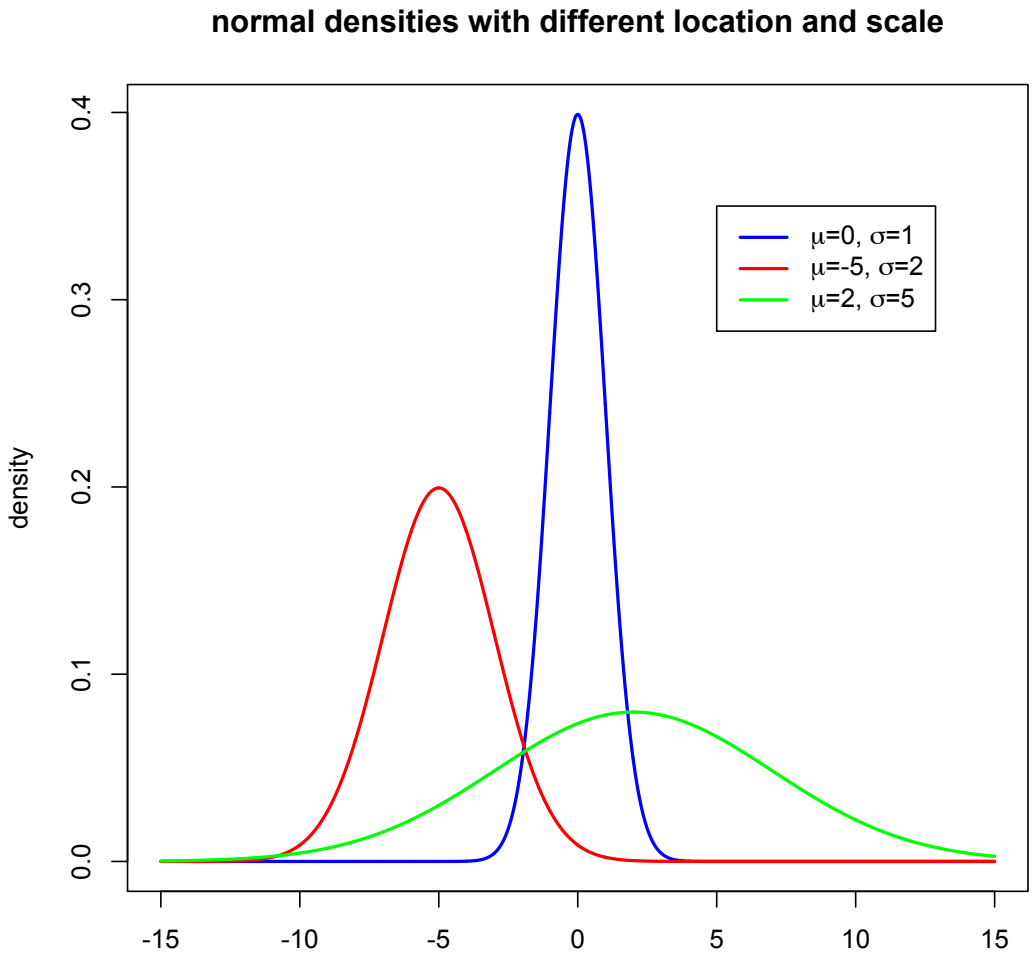
A population curve or population density is a (smoothed) histogram of a population of values.

A population can be an actual population, e.g. the heights of all men in the Netherlands. It can also be the (imaginary) infinite number of outcomes obtained by repeating an experiment over and over.

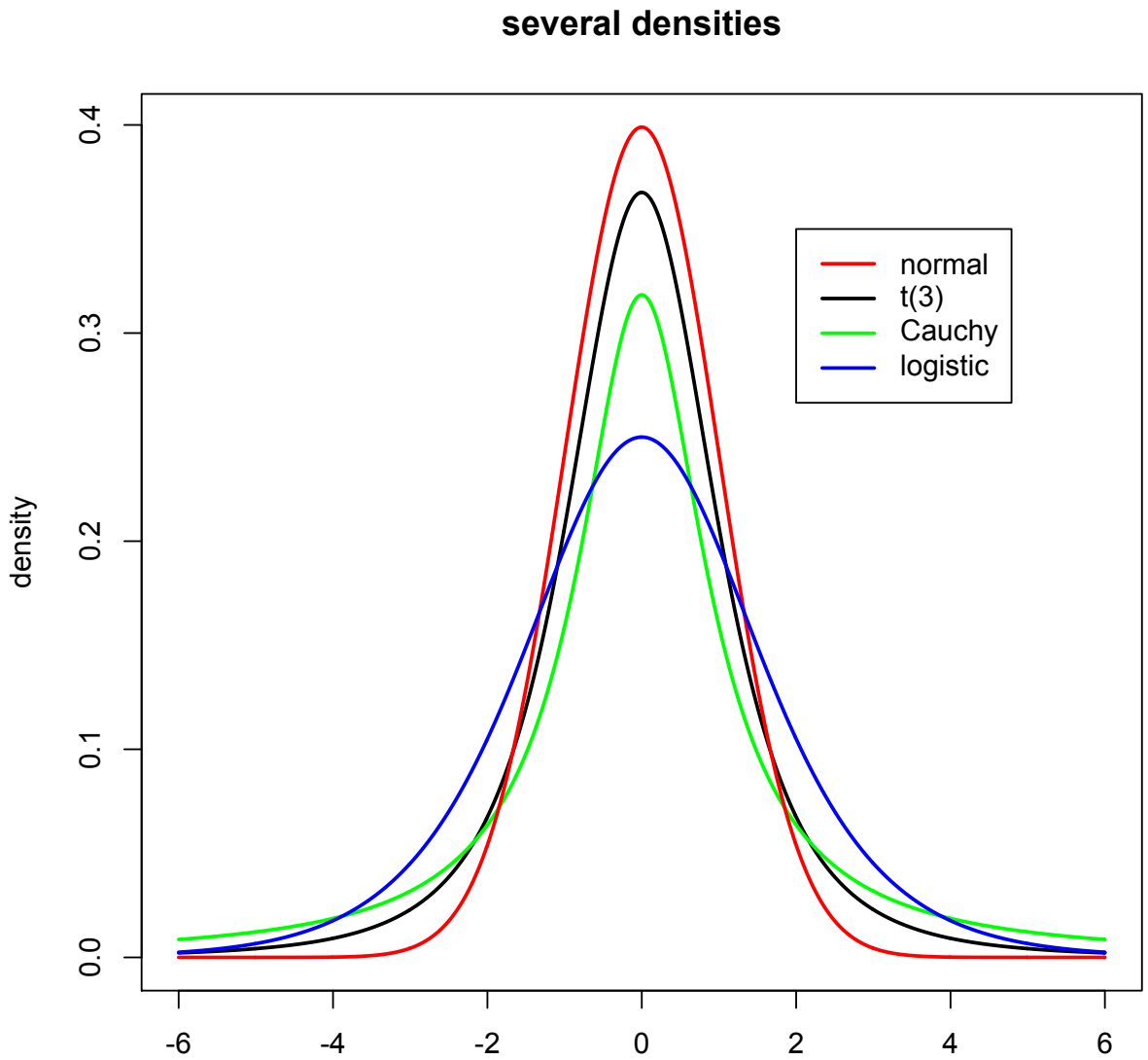


Recap — normal density

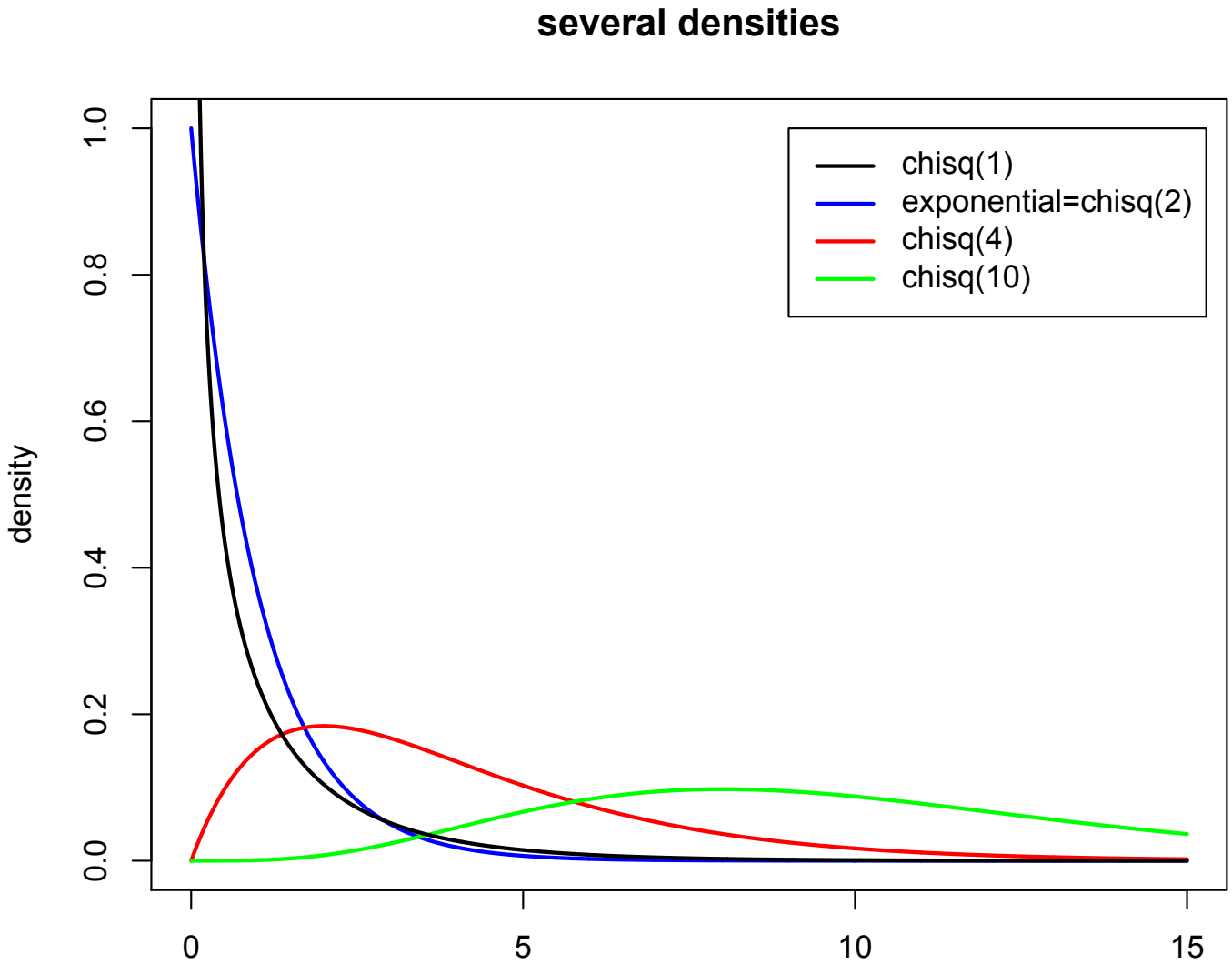
The normal density curve is given by the function $x \mapsto \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}(x-\mu)^2/\sigma^2}$. The parameters μ and σ are the location and scale.



Other symmetric population curves

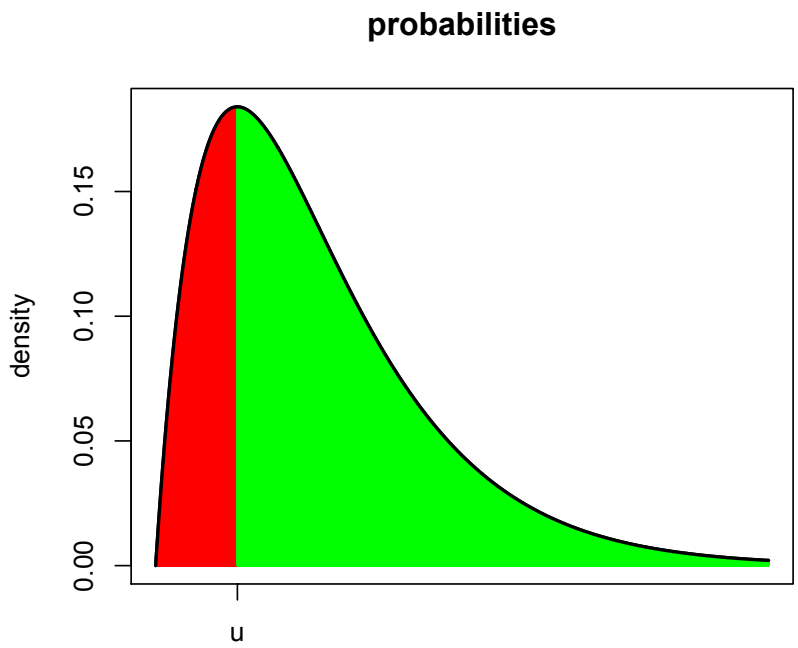


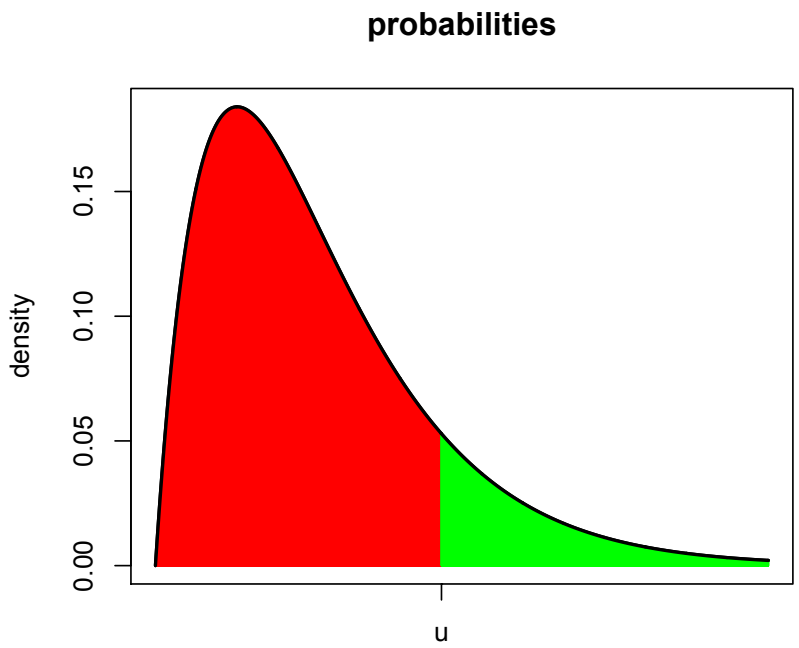
Asymmetric population curves



About population curves

If a random variable X is distributed according to a density curve, the probability $\mathrm{P}(X \leq u)$ is the (red) area under the density curve left of u. Likewise, $\mathrm{P}(X \geq u)$ is the (green) area under the density curve right of u.





bootstrap confidence intervals

Confidence interval for normal data

A point estimate for an unknown parameter μ is the outcome of some estimating statistic.

EXAMPLE Suppose we have a sample $X_1, \ldots, X_N$ from a normal population with unknown population mean μ. We can estimate μ using the estimating statistic $\overline{X}$. The point estimate for μ is $\hat{\mu} = \overline{X}$.

A confidence interval for an unknown parameter μ is an interval around the point estimate. It contains the unknown parameter with e.g. 95 % confidence. The length of this interval is based upon the distribution of the estimating statistic.

EXAMPLE (cont'd) The confidence interval for μ with 95% confidence is the interval $[\overline{X} - m, \overline{X} + m]$, where $m = 2s/\sqrt{N}$. This margin m is based on the normality of the sample.

Confidence interval for normal data

A point estimate for an unknown parameter μ is the outcome of some estimating statistic.

EXAMPLE Suppose we have a sample $X_1, \ldots, X_N$ from a normal population with unknown population mean μ. We can estimate μ using the estimating statistic $\overline{X}$. The point estimate for μ is $\hat{\mu} = \overline{X}$.

A confidence interval for an unknown parameter μ is an interval around the point estimate. It contains the unknown parameter with e.g. 95 % confidence. The length of this interval is based upon the distribution of the estimating statistic.

EXAMPLE (cont'd) The confidence interval for μ with 95% confidence is the interval $[\overline{X} - m, \overline{X} + m]$, where $m = 2s/\sqrt{N}$. This margin m is based on the normality of the sample.

Confidence interval for normal data

A point estimate for an unknown parameter μ is the outcome of some estimating statistic.

EXAMPLE Suppose we have a sample $X_1, \ldots, X_N$ from a normal population with unknown population mean μ. We can estimate μ using the estimating statistic $\overline{X}$. The point estimate for μ is $\hat{\mu} = \overline{X}$.

A confidence interval for an unknown parameter μ is an interval around the point estimate. It contains the unknown parameter with e.g. 95 % confidence. The length of this interval is based upon the distribution of the estimating statistic.

EXAMPLE (cont'd) The confidence interval for μ with 95% confidence is the interval $[\overline{X} - m, \overline{X} + m]$, where $m = 2s/\sqrt{N}$. This margin m is based on the normality of the sample.

Confidence interval for normal data

A point estimate for an unknown parameter μ is the outcome of some estimating statistic.

EXAMPLE Suppose we have a sample $X_1, \ldots, X_N$ from a normal population with unknown population mean μ. We can estimate μ using the estimating statistic $\overline{X}$. The point estimate for μ is $\hat{\mu} = \overline{X}$.

A confidence interval for an unknown parameter μ is an interval around the point estimate. It contains the unknown parameter with e.g. 95 % confidence. The length of this interval is based upon the distribution of the estimating statistic.

EXAMPLE (cont'd) The confidence interval for μ with 95% confidence is the interval $[\overline{X} - m, \overline{X} + m]$, where $m = 2s/\sqrt{N}$. This margin m is based on the normality of the sample.

Confidence interval for nonnormal data

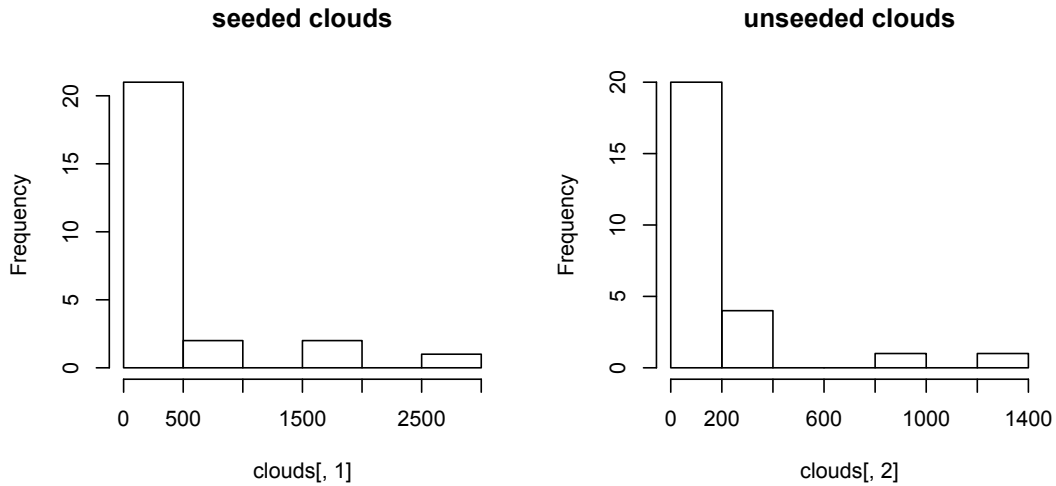
If we have a (small) sample from an unknown distribution we cannot make confidence intervals, because we have no information about the size of the margin m.

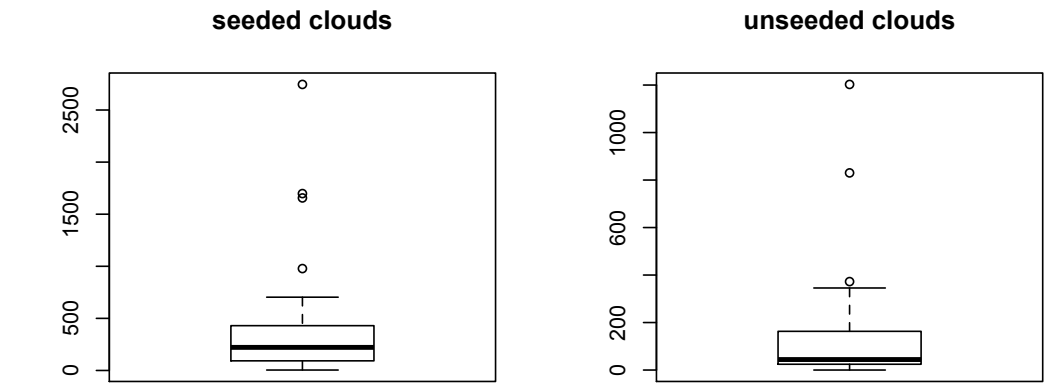
EXAMPLE

Estimate the population mean of the two clouds data sets:

```
> c1=clouds[,1] #   seeded
> c2=clouds[,2] # unseeded
> T1=mean(c1)
> T2=mean(c2)
> T1
[1] 441.9846
> T2
[1] 164.5619
```

How to set up confidence intervals?





Confidence interval for nonnormal data

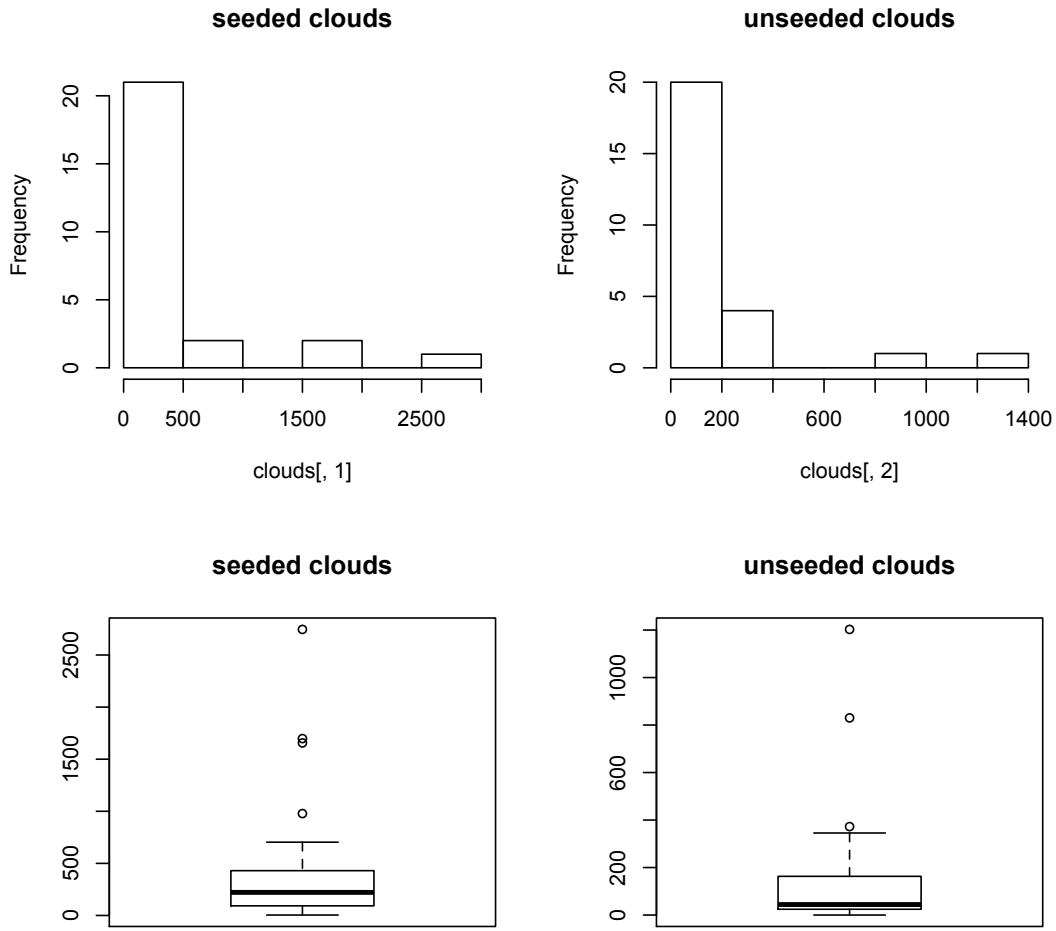
If we have a (small) sample from an unknown distribution we cannot make confidence intervals, because we have no information about the size of the margin m.

EXAMPLE

Estimate the population mean of the two clouds data sets:

```
> c1=clouds[,1] #   seeded
> c2=clouds[,2] # unseeded
> T1=mean(c1)
> T2=mean(c2)
> T1
[1] 441.9846
> T2
[1] 164.5619
```

How to set up confidence intervals?



Confidence interval for nonnormal data

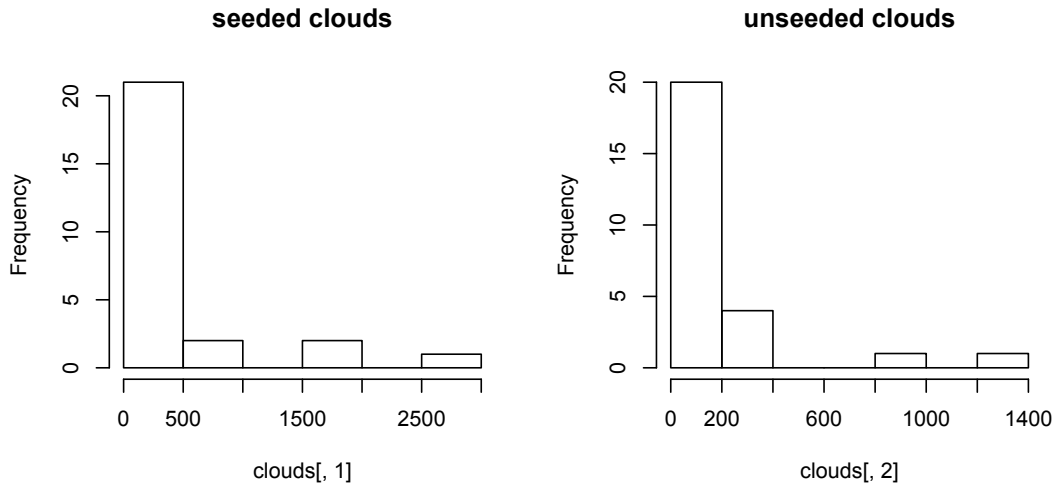
If we have a (small) sample from an unknown distribution we cannot make confidence intervals, because we have no information about the size of the margin m.

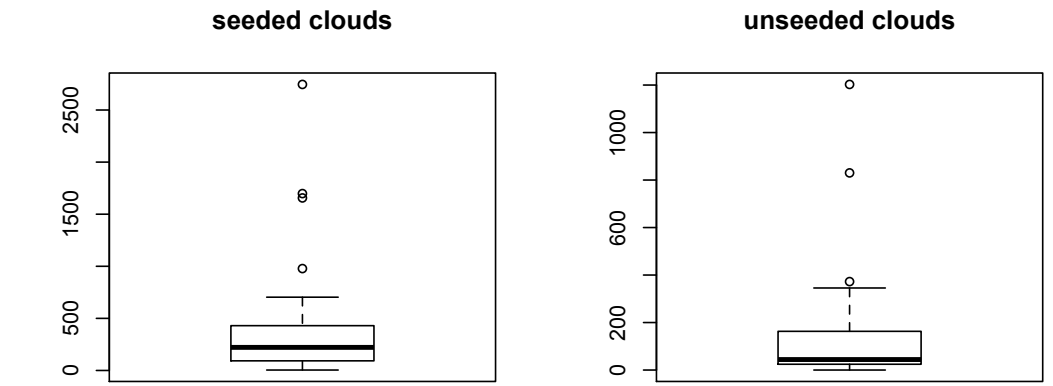
EXAMPLE

Estimate the population mean of the two clouds data sets:

```
> c1=clouds[,1] #   seeded
> c2=clouds[,2] # unseeded
> T1=mean(c1)
> T2=mean(c2)
> T1
[1] 441.9846
> T2
[1] 164.5619
```

How to set up confidence intervals?





Bootstrap confidence interval

A bootstrap confidence interval uses simulation to find the distribution of the estimating statistic. The left and right margins for the confidence interval are found from this simulated distribtution.

Denote the data sample as $X_1, \ldots, X_N$ and the estimating statistic as $T = T(X_1, \ldots, X_N)$. The bootstrap method estimates the distribution of T by a sample of representative values $T_1^*, \ldots, T_B^*$ with B large.

The formula for the bootstrap confidence interval with confidence $1 - 2\alpha$ is

$$[2T(X_1, \ldots, X_N) - T^*_{(1-\alpha)}, 2T(X_1, \ldots, X_N) - T^*_{(\alpha)}]$$

where $T^*_{(\alpha)}$ is the T^*-value such that $\alpha \times 100\%$ of the T^*-values are lower than $T^*_{(\alpha)}$.

Bootstrap confidence interval

A bootstrap confidence interval uses simulation to find the distribution of the estimating statistic. The left and right margins for the confidence interval are found from this simulated distribtution.

Denote the data sample as $X_1, \ldots, X_N$ and the estimating statistic as $T = T(X_1, \ldots, X_N)$. The bootstrap method estimates the distribution of T by a sample of representative values $T_1^*, \ldots, T_B^*$ with B large.

The formula for the bootstrap confidence interval with confidence $1 - 2\alpha$ is

$$[2T(X_1, \ldots, X_N) - T^*_{(1-\alpha)}, 2T(X_1, \ldots, X_N) - T^*_{(\alpha)}]$$

where $T^*_{(\alpha)}$ is the T^*-value such that $\alpha \times 100\%$ of the T^*-values are lower than $T^*_{(\alpha)}$.

Bootstrap confidence interval

A bootstrap confidence interval uses simulation to find the distribution of the estimating statistic. The left and right margins for the confidence interval are found from this simulated distribtution.

Denote the data sample as $X_1, \ldots, X_N$ and the estimating statistic as $T = T(X_1, \ldots, X_N)$. The bootstrap method estimates the distribution of T by a sample of representative values $T_1^*, \ldots, T_B^*$ with B large.

The formula for the bootstrap confidence interval with confidence $1 - 2\alpha$ is

$$[2T(X_1, \ldots, X_N) - T^*_{(1-\alpha)}, 2T(X_1, \ldots, X_N) - T^*_{(\alpha)}]$$

where $T^*_{(\alpha)}$ is the T^*-value such that $\alpha \times 100\%$ of the T^*-values are lower than $T^*_{(\alpha)}$.

T^*-values

The generation of the T^* values is as follows.

Repeat B times ($i = 1, \ldots, B$)

- generate a surrogate data set $X_1^*, \ldots, X_N^*$ by sampling N values from the original data set $X_1, \ldots, X_N$ with replacement
- compute $T_i^* = T(X_1^*, \ldots, X_N^*)$ for the surrogate sample

In the first step we sample from the data that we have. In this step some data points X_i may be chosen more than once amongst the X^*-values, whereas other data points X_i may not be chosen at all. We do not introduce any new X-values. We only determine new T^*-values.

This procedure yields $T_1^*, \ldots, T_B^*$.

(This bootstrap procedure is called the empirical bootstrap.)

T^*-values

The generation of the T^* values is as follows.

Repeat B times ($i = 1, \ldots, B$)

- generate a surrogate data set $X_1^*, \ldots, X_N^*$ by sampling N values from the original data set $X_1, \ldots, X_N$ with replacement
- compute $T_i^* = T(X_1^*, \ldots, X_N^*)$ for the surrogate sample

In the first step we sample from the data that we have. In this step some data points X_i may be chosen more than once amongst the X^*-values, whereas other data points X_i may not be chosen at all. We do not introduce any new X-values. We only determine new T^*-values.

This procedure yields $T_1^*, \ldots, T_B^*$.

(This bootstrap procedure is called the empirical bootstrap.)

Estimation in R

EXAMPLE (cont'd) For the seeded clouds data (c1) we determine this interval:

```
> B=1000
> Tstar=numeric(B)
> for(i in 1:B)
+ {
+  Xstar=sample(c1,replace=TRUE)
+  Tstar[i]=mean(Xstar)
+ }
> Tstar25=quantile(Tstar,0.025)
> Tstar975=quantile(Tstar,0.975)
> sum(Tstar<Tstar25)
[1] 25
> c(2*T1-Tstar975,2*T1-Tstar25)
176.8857 668.9462
```

generate $X_1^*, \ldots, X_N^*$
compute T_i^*

determine $T^*_{(\alpha)}$
determine $T^*_{(1-\alpha)}$

The 95% bootstrap confidence interval for the population mean of seeded clouds is [177, 669] around its mean $T1 = 442$.

Similarly, we find for unseeded clouds the interval [42, 254] around its mean $T2 = 165$.

Estimation in R

EXAMPLE (cont'd) For the seeded clouds data (c1) we determine this interval:

```
> B=1000
> Tstar=numeric(B)
> for(i in 1:B)
+ {
+  Xstar=sample(c1,replace=TRUE)
+  Tstar[i]=mean(Xstar)
+ }
> Tstar25=quantile(Tstar,0.025)
> Tstar975=quantile(Tstar,0.975)
> sum(Tstar<Tstar25)
[1] 25
> c(2*T1-Tstar975,2*T1-Tstar25)
176.8857 668.9462
```

generate $X_1^*, \ldots, X_N^*$
compute T_i^*

determine $T^*_{(\alpha)}$
determine $T^*_{(1-\alpha)}$

The 95% bootstrap confidence interval for the population mean of seeded clouds is [177, 669] around its mean $T1 = 442$.

Similarly, we find for unseeded clouds the interval [42, 254] around its mean $T2 = 165$.

Estimation in R (2)

The smaller a confidence interval (with fixed confidence) the more accurate our estimation is. These two intervals are very large, because the mean is influenced by the large values in the data set. That is, the estimating statistic $\overline{X}$ is not robust against outliers.

A robust estimator for location is $median(X)$, which is the estimating statistic for the population median. For the clouds data, the median is smaller than the mean, because of the large outliers.

The 95% bootstrap confidence interval for the population median of seeded clouds is [139, 326] (cf. [177, 669] for population mean).
Similarly, we find for unseeded clouds the interval [-20, 62] (cf. [42, 254] for population mean).

For both data sets: the confidence interval for the median is shorter and contains lower values. This confirms that the median is more robust than the mean.

Estimation in R (2)

The smaller a confidence interval (with fixed confidence) the more accurate our estimation is. These two intervals are very large, because the mean is influenced by the large values in the data set. That is, the estimating statistic $\overline{X}$ is not robust against outliers.

A robust estimator for location is $median(X)$, which is the estimating statistic for the population median. For the clouds data, the median is smaller than the mean, because of the large outliers.

The 95% bootstrap confidence interval for the population median of seeded clouds is [139, 326] (cf. [177, 669] for population mean).
Similarly, we find for unseeded clouds the interval [-20, 62] (cf. [42, 254] for population mean).

For both data sets: the confidence interval for the median is shorter and contains lower values. This confirms that the median is more robust than the mean.

Estimation in R (2)

The smaller a confidence interval (with fixed confidence) the more accurate our estimation is. These two intervals are very large, because the mean is influenced by the large values in the data set. That is, the estimating statistic $\overline{X}$ is not robust against outliers.

A robust estimator for location is $median(X)$, which is the estimating statistic for the population median. For the clouds data, the median is smaller than the mean, because of the large outliers.

The 95% bootstrap confidence interval for the population median of seeded clouds is [139, 326] (cf. [177, 669] for population mean).
Similarly, we find for unseeded clouds the interval [-20, 62] (cf. [42, 254] for population mean).

For both data sets: the confidence interval for the median is shorter and contains lower values. This confirms that the median is more robust than the mean.

Estimation in R (2)

The smaller a confidence interval (with fixed confidence) the more accurate our estimation is. These two intervals are very large, because the mean is influenced by the large values in the data set. That is, the estimating statistic $\overline{X}$ is not robust against outliers.

A robust estimator for location is $median(X)$, which is the estimating statistic for the population median. For the clouds data, the median is smaller than the mean, because of the large outliers.

The 95% bootstrap confidence interval for the population median of seeded clouds is [139, 326] (cf. [177, 669] for population mean).
Similarly, we find for unseeded clouds the interval [-20, 62] (cf. [42, 254] for population mean).

For both data sets: the confidence interval for the median is shorter and contains lower values. This confirms that the median is more robust than the mean.

Estimation in R (2)

The smaller a confidence interval (with fixed confidence) the more accurate our estimation is. These two intervals are very large, because the mean is influenced by the large values in the data set. That is, the estimating statistic $\overline{X}$ is not robust against outliers.

A robust estimator for location is $median(X)$, which is the estimating statistic for the population median. For the clouds data, the median is smaller than the mean, because of the large outliers.

The 95% bootstrap confidence interval for the population median of seeded clouds is [139, 326] (cf. [177, 669] for population mean).
Similarly, we find for unseeded clouds the interval [-20, 62] (cf. [42, 254] for population mean).

For both data sets: the confidence interval for the median is shorter and contains lower values. This confirms that the median is more robust than the mean.

Bootstrap confidence intervals — discussion

- Repeating the computation of a bootstrap confidence interval will always yield a different interval. The variation in the intervals is due to the size of B. Enlarging B will reduce the variation.
- Whereas the bootstrap interval is for a population parameter, the computed interval depends on the precise data values in the sample $X_1, \ldots, X_N$. In case these values are somewhat extreme in the population (this can happen with probability > 0) then the bootstrap interval will be a bit off as well. We cannot correct for this, because our only information is the sample.

statistical tests

Recap — statistical test

A statistical test chooses between two possibilities: the null hypothesis H_0 and the alternative hypothesis H_1.

Statistical tests are typically not perfect, but make two types of errors:
- Error of the first kind rejecting H_0 while it is true.
- Error of the second kind not rejecting H_0 while it is false.

Tests are constructed to have small probability of an error of the first kind ($<$ 5%). The error of the second kind depends (among others) on the amount of data. Because of this asymmetry we either reject H_0 (and accept H_1) or do not reject H_0 (and treat the analysis as inconclusive).

Recap — statistical test

A statistical test chooses between two possibilities: the null hypothesis H_0 and the alternative hypothesis H_1.

Statistical tests are typically not perfect, but make two types of errors:
- Error of the first kind rejecting H_0 while it is true.
- Error of the second kind not rejecting H_0 while it is false.

Tests are constructed to have small probability of an error of the first kind ($<$ 5%). The error of the second kind depends (among others) on the amount of data. Because of this asymmetry we either reject H_0 (and accept H_1) or do not reject H_0 (and treat the analysis as inconclusive).

Test statistic (1)

A statistical test uses a test statistic. In this test statistic relevant information from the data about the validity of H_0 is quantified.

EXAMPLE The t-test is for testing the population mean μ of a normal population, $H_0 : \mu = \mu_0$. Given a sample $X_1, \ldots, X_N$ from the population, the test statistic is

$$T = \frac{\overline{X}_N - \mu_0}{S_N}.$$

When T is very different from 0, we reject H_0.

The critical value for T that acts as border between rejecting and not rejecting H_0 is based on the distribution of T under H_0. For the t-test this distribution is the t_{N-1}-distribution.

Test statistic (1)

A statistical test uses a test statistic. In this test statistic relevant information from the data about the validity of H_0 is quantified.

EXAMPLE The t-test is for testing the population mean μ of a normal population, $H_0 : \mu = \mu_0$. Given a sample $X_1, \ldots, X_N$ from the population, the test statistic is

$$T = \frac{\overline{X}_N - \mu_0}{S_N}.$$

When T is very different from 0, we reject H_0.

The critical value for T that acts as border between rejecting and not rejecting H_0 is based on the distribution of T under H_0. For the t-test this distribution is the t_{N-1}-distribution.

Test statistic (2)

The test statistic is not unique. For the same H_0 we can choose different test statistics.

EXAMPLE For testing $H_0 : \mu = 0$ we can as well use the sign test. Given a sample $X_1, \ldots, X_N$ from the population, the test statistic for the sign test is

$$T = \#(X_i < 0).$$

When T is very different from $N/2$ we reject H_0. For this test the critical value comes from the bin$(N, \frac{1}{2})$-distribution, the distribution of number of heads in throwing N times a fair coin.

In general performing a test requires a test statistic and its distribution under H_0. Without this distribution we do not know when to reject, and when not to.

Test statistic (2)

The test statistic is not unique. For the same H_0 we can choose different test statistics.

EXAMPLE For testing $H_0 : \mu = 0$ we can as well use the sign test. Given a sample $X_1, \ldots, X_N$ from the population, the test statistic for the sign test is

$$T = \#(X_i < 0).$$

When T is very different from $N/2$ we reject H_0. For this test the critical value comes from the bin$(N, \frac{1}{2})$-distribution, the distribution of number of heads in throwing N times a fair coin.

In general performing a test requires a test statistic and its distribution under H_0. Without this distribution we do not know when to reject, and when not to.

Choosing a test statistic

Different test statistics can yield different statistical power of the test.

The power of a test is 1 minus the probability of an error of the second kind. In other words, the power is the probability of correctly rejecting H_0 (that is, when H_0 is not true). Apart from the test statistic, also the sample size influences the statistical power: higher sample sizes yield higher power.

Tests with high statistical power are preferred, keeping the level of the test (probability of an error of the first kind, often taken at 5%) fixed.

(The power of a test is specified for each possibility under H_1. E.g. if $H_0 : \mu = 0$ then the power can be calculated in each value $\mu \neq 0$. A *good* test (that is, a test based on a *good* test statistic) has high power in all these nonzero μ-values, relative to other tests.)

Choosing a test statistic

Different test statistics can yield different statistical power of the test.

The power of a test is 1 minus the probability of an error of the second kind. In other words, the power is the probability of correctly rejecting H_0 (that is, when H_0 is not true). Apart from the test statistic, also the sample size influences the statistical power: higher sample sizes yield higher power.

Tests with high statistical power are preferred, keeping the level of the test (probability of an error of the first kind, often taken at 5%) fixed.

(The power of a test is specified for each possibility under H_1. E.g. if $H_0 : \mu = 0$ then the power can be calculated in each value $\mu \neq 0$. A *good* test (that is, a test based on a *good* test statistic) has high power in all these nonzero μ-values, relative to other tests.)

Choosing a test statistic

Different test statistics can yield different statistical power of the test.

The power of a test is 1 minus the probability of an error of the second kind. In other words, the power is the probability of correctly rejecting H_0 (that is, when H_0 is not true). Apart from the test statistic, also the sample size influences the statistical power: higher sample sizes yield higher power.

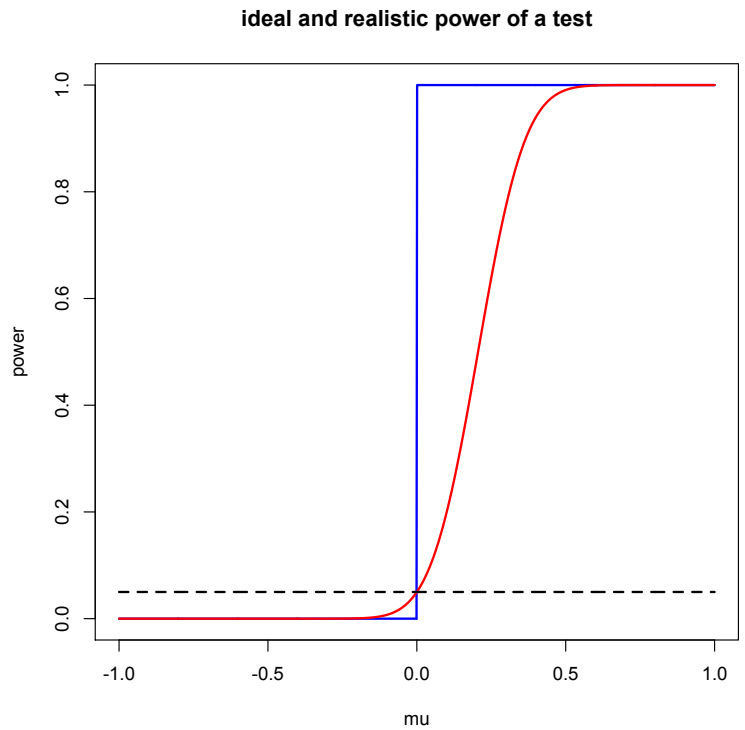
Tests with high statistical power are preferred, keeping the level of the test (probability of an error of the first kind, often taken at 5%) fixed.

(The power of a test is specified for each possibility under H_1. E.g. if $H_0 : \mu = 0$ then the power can be calculated in each value $\mu \neq 0$. A *good* test (that is, a test based on a *good* test statistic) has high power in all these nonzero μ-values, relative to other tests.)

Ideal test and realistic test

The ideal test would make no errors, e.g.

- never falsely reject (no error of type I)
- always reject when H_1 is true (no error of type II)



The power of the ideal test and a realistic test for $H_0 : \mu \leq 0$. The dashed line is the level of the test, here 0.05 (the probability of type I error).

Comparing powers

Assume we have a normal sample and test $H_0 : \mu = 0$ using the t-test and the sign test. We can compare the power in $\mu = 0.5$ of the two tests by simulation.

```
> B=1000
> n=50
> psign=numeric(B)   ## will contain p-values of sign test
> pttest=numeric(B)  ## will contain p-values of t-test
> for(i in 1:B) {
+  x=rnorm(n,mean=0.5,sd=1) ## generate data under H1 with mu=0.5
+  pttest[i]=t.test(x)[[3]]                    ## extract p-value
+  psign[i]=binom.test(sum(x>0),n,p=0.5)[[3]]    ## extract p-value
+ }
> sum(psign<0.05)/B
[1] 0.746
> sum(pttest<0.05)/B
[1] 0.937
```

The power in $\mu = 0.5$ for the t-test (0.937) is higher than for the sign test (0.746) when we reject for p-values smaller than the level 0.05. Why?

Hence, for normally distributed data the t-test has better performance than the sign test.

Comparing powers

Assume we have a normal sample and test $H_0 : \mu = 0$ using the t-test and the sign test. We can compare the power in $\mu = 0.5$ of the two tests by simulation.

```
> B=1000
> n=50
> psign=numeric(B)   ## will contain p-values of sign test
> pttest=numeric(B)  ## will contain p-values of t-test
> for(i in 1:B) {
+  x=rnorm(n,mean=0.5,sd=1) ## generate data under H1 with mu=0.5
+  pttest[i]=t.test(x)[[3]]                    ## extract p-value
+  psign[i]=binom.test(sum(x>0),n,p=0.5)[[3]]    ## extract p-value
+ }
> sum(psign<0.05)/B
[1] 0.746
> sum(pttest<0.05)/B
[1] 0.937
```

The power in $\mu = 0.5$ for the t-test (0.937) is higher than for the sign test (0.746) when we reject for p-values smaller than the level 0.05. Why?

Hence, for normally distributed data the t-test has better performance than the sign test.

Comparing powers

Assume we have a normal sample and test $H_0 : \mu = 0$ using the t-test and the sign test. We can compare the power in $\mu = 0.5$ of the two tests by simulation.

```
> B=1000
> n=50
> psign=numeric(B)   ## will contain p-values of sign test
> pttest=numeric(B)  ## will contain p-values of t-test
> for(i in 1:B) {
+  x=rnorm(n,mean=0.5,sd=1) ## generate data under H1 with mu=0.5
+  pttest[i]=t.test(x)[[3]]                    ## extract p-value
+  psign[i]=binom.test(sum(x>0),n,p=0.5)[[3]]    ## extract p-value
+ }
> sum(psign<0.05)/B
[1] 0.746
> sum(pttest<0.05)/B
[1] 0.937
```

The power in $\mu = 0.5$ for the t-test (0.937) is higher than for the sign test (0.746) when we reject for p-values smaller than the level 0.05. Why?

Hence, for normally distributed data the t-test has better performance than the sign test.

Comparing powers

Assume we have a normal sample and test $H_0 : \mu = 0$ using the t-test and the sign test. We can compare the power in $\mu = 0.5$ of the two tests by simulation.

```
> B=1000
> n=50
> psign=numeric(B)   ## will contain p-values of sign test
> pttest=numeric(B)  ## will contain p-values of t-test
> for(i in 1:B) {
+  x=rnorm(n,mean=0.5,sd=1) ## generate data under H1 with mu=0.5
+  pttest[i]=t.test(x)[[3]]                    ## extract p-value
+  psign[i]=binom.test(sum(x>0),n,p=0.5)[[3]]    ## extract p-value
+ }
> sum(psign<0.05)/B
[1] 0.746
> sum(pttest<0.05)/B
[1] 0.937
```

The power in $\mu = 0.5$ for the t-test (0.937) is higher than for the sign test (0.746) when we reject for p-values smaller than the level 0.05. Why?

Hence, for normally distributed data the t-test has better performance than the sign test.

bootstrap tests

Idea

Suppose we are given

- a sample $X_1, \ldots, X_N$
- a null hypothesis H_0 stating some claim about the population distribution
- a (sensible) test statistic $T = T(X_1, \ldots, X_N)$

but we lack

- the distribution of T under H_0.

In such a case, we cannot perform the test, because we do not have a critical value for T, that acts as border between rejecting and not rejecting H_0.

For this situation we can use a bootstrap test. It uses simulation to find (an estimate of) the distribution of T under H_0.

(For a bootstrap test we cannot use a standard R-command — we have to program it ourselves.)

Idea

Suppose we are given

- a sample $X_1, \ldots, X_N$
- a null hypothesis H_0 stating some claim about the population distribution
- a (sensible) test statistic $T = T(X_1, \ldots, X_N)$

but we lack

- the distribution of T under H_0.

In such a case, we cannot perform the test, because we do not have a critical value for T, that acts as border between rejecting and not rejecting H_0.

For this situation we can use a bootstrap test. It uses simulation to find (an estimate of) the distribution of T under H_0.

(For a bootstrap test we cannot use a standard R-command — we have to program it ourselves.)

Idea

Suppose we are given

- a sample $X_1, \ldots, X_N$
- a null hypothesis H_0 stating some claim about the population distribution
- a (sensible) test statistic $T = T(X_1, \ldots, X_N)$

but we lack

- the distribution of T under H_0.

In such a case, we cannot perform the test, because we do not have a critical value for T, that acts as border between rejecting and not rejecting H_0.

For this situation we can use a bootstrap test. It uses simulation to find (an estimate of) the distribution of T under H_0.

(For a bootstrap test we cannot use a standard R-command — we have to program it ourselves.)

Set up of a bootstrap test

Given our sample $X_1, \ldots, X_N$, we can compute the test statistic $T = T(X_1, \ldots, X_N)$ based on our sample.

Simulating the distribution of T under H_0 in the bootstrap fashion means generate a bunch of surrogate T-values $(T_1^*, \ldots, T_B^*)$ that are representative values for T under H_0.

The simulation set up is

- repeat B times $(i = 1, \ldots, B)$
  1. generate a surrogate data sample $X_1^*, \ldots, X_N^*$ (same sample size as original data set) according to H_0
  2. compute the test statistic $T_i^* = T(X_1^*, \ldots, X_N^*)$ for the surrogate sample
- compare the T-value of the original data to the surrogate T^*-values and determine a p-value.

(By simulating the unknown distribution we make an estimation error. This error can be made arbitrarily small by choosing B large enough.)

Set up of a bootstrap test

Given our sample $X_1, \ldots, X_N$, we can compute the test statistic $T = T(X_1, \ldots, X_N)$ based on our sample.

Simulating the distribution of T under H_0 in the bootstrap fashion means generate a bunch of surrogate T-values $(T_1^*, \ldots, T_B^*)$ that are representative values for T under H_0.

The simulation set up is

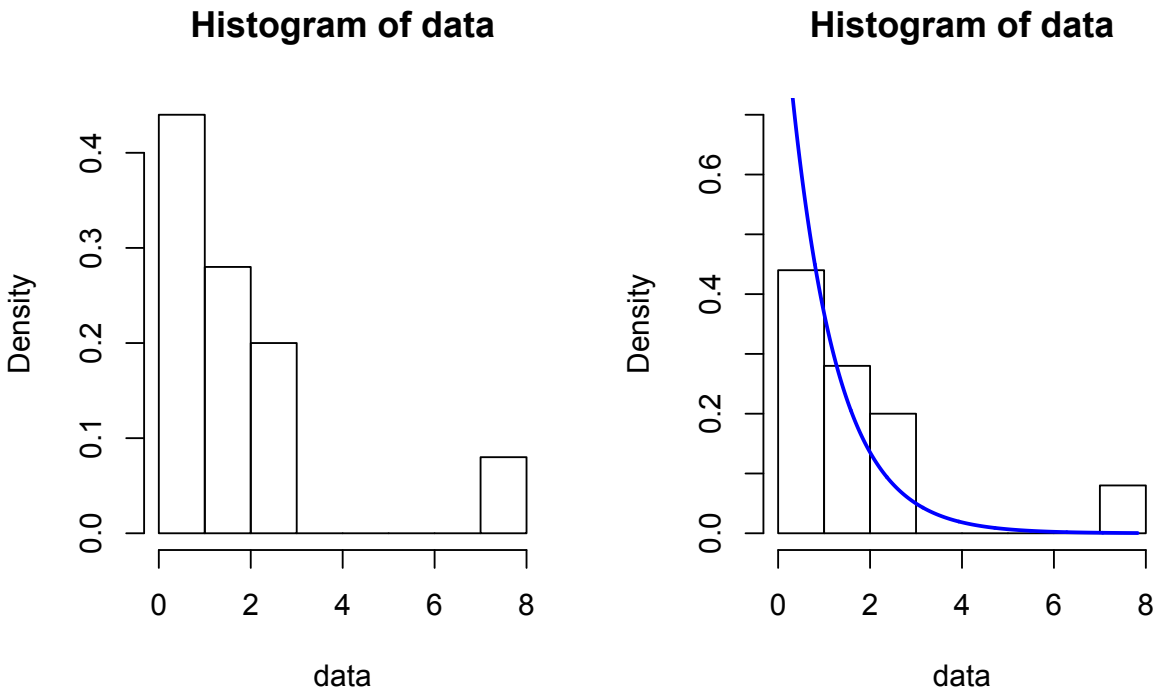
- repeat B times $(i = 1, \ldots, B)$
  1. generate a surrogate data sample $X_1^*, \ldots, X_N^*$ (same sample size as original data set) according to H_0
  2. compute the test statistic $T_i^* = T(X_1^*, \ldots, X_N^*)$ for the surrogate sample
- compare the T-value of the original data to the surrogate T^*-values and determine a p-value.

(By simulating the unknown distribution we make an estimation error. This error can be made arbitrarily small by choosing B large enough.)

Bootstrap test — implementation in R (1)

We wish to test $H_0 : X_i \sim \exp(1)$, i.i.d. $i = 1 \ldots, N$, i.e. the data are a random sample from the standard exponential distribution.

```
> hist(data,prob=T)
> hist(data,prob=T,ylim=c(0,0.7))
> x=seq(0,max(data),length=1000)
> lines(x,dexp(x),type="l",col="blue",lwd=2)
```

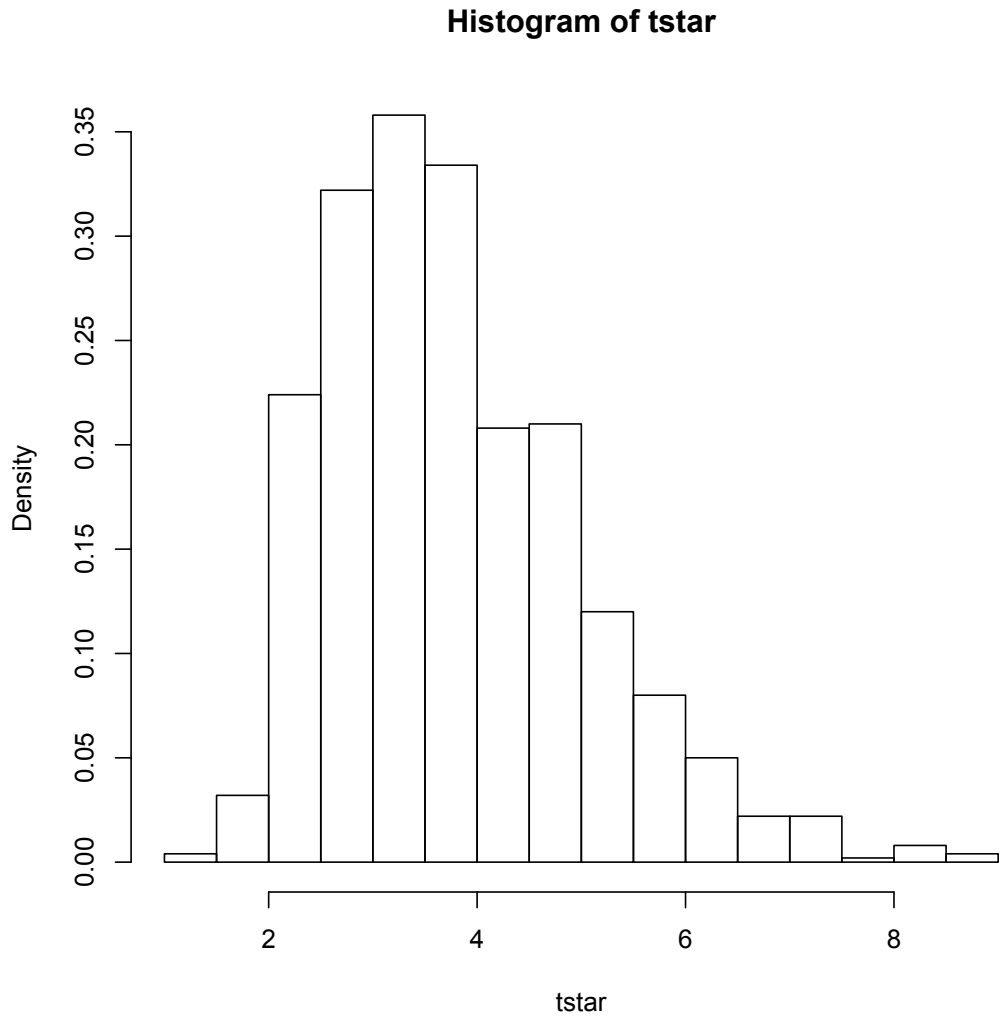


Bootstrap test — implementation in R (2)

We use as test statistic the maximum of the sample:
$T(X_1, \ldots, X_N) = max(X_1, \ldots, X_N)$.

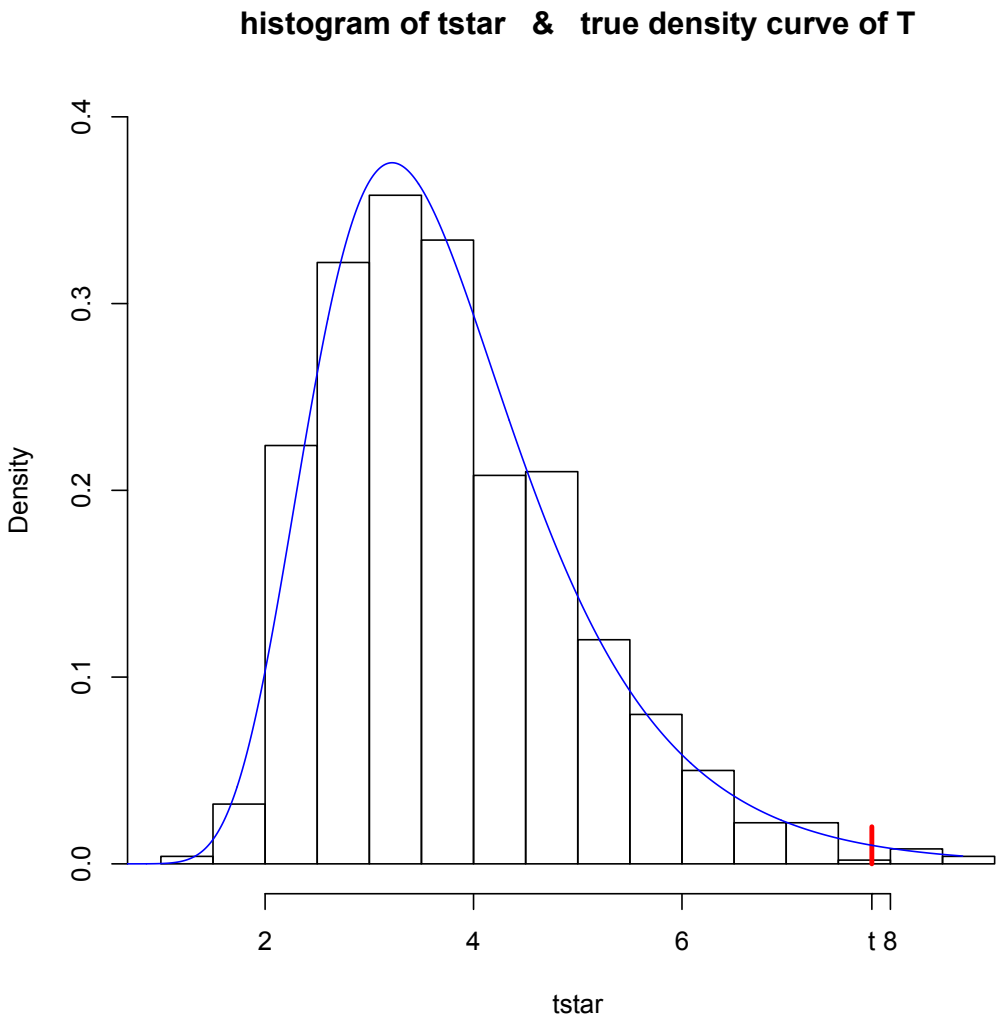
```
> t=max(data)
> t
[1] 7.821847

> B=1000
> tstar=numeric(B)
> n=length(data)
> for (i in 1:B){
+  xstar=rexp(n,1)
+  tstar[i]=max(xstar)
+ }
> hist(tstar,prob=T)
```



Bootstrap test — p-value in R (1)

The p-value is found by considering the proportion of T^*-values exceeding the T-value of the data.



Bootstrap test — p-value in R (2)

The R-code for the *p*-value:

```
> pl=sum(tstar<t)/B
> pr=sum(tstar>t)/B
> p=2*min(pl,pr)
> pl;pr;p
[1] 0.994
[1] 0.006
[1] 0.012
```

The *p*-value is 0.012 and H_0 is rejected.
The R-code for the histogram in the previous slide:

```
> hist(tstar,prob=T,ylim=c(0,0.4),
+ main="histogram of tstar & true density curve of T")
> densmaxexp=function(x,n) n*exp(-x)*(1-exp(-x))^(n-1)
> lines(rep(t,2),seq(0,2*densmaxexp(t,n),length=2),
+ type="l", col="red", lwd=3)
> axis(1,t,expression(paste("t") ) )
> u=seq(0,max(tstar),length=1000)
> lines(u,densmaxexp(u,n),type="l",col="blue")
```

Bootstrap test — discussion

- The resulting p-value depends on the exact T^*-values. Hence, it is recommended to repeat a bootstrap test a few times to see whether the p-value is stable. When B is too small, there is a lot of variation in the p-value. In that case B should be increased. In most cases $B = 1000$ is adequate.
- A bootstrap test can be performed with any test statistic. E.g. in the example taking `min` as a test statistic yields a bootstrap p-value of about 0.19 (check this yourselves!) and does not lead to rejecting H_0.
- The difference between the simulation of T^*-values for bootstrap confidence intervals and bootstrap tests is in the way the $X_1^*, \ldots, X_N^*$ are generated. For confidence intervals you choose X_i^* from your sample, whereas for tests you generate X_i^* according to H_0.

Bootstrap test — discussion

- The resulting p-value depends on the exact T^*-values. Hence, it is recommended to repeat a bootstrap test a few times to see whether the p-value is stable. When B is too small, there is a lot of variation in the p-value. In that case B should be increased. In most cases $B = 1000$ is adequate.
- A bootstrap test can be performed with any test statistic. E.g. in the example taking `min` as a test statistic yields a bootstrap p-value of about 0.19 (check this yourselves!) and does not lead to rejecting H_0.
- The difference between the simulation of T^*-values for bootstrap confidence intervals and bootstrap tests is in the way the $X_1^*, \ldots, X_N^*$ are generated. For confidence intervals you choose X_i^* from your sample, whereas for tests you generate X_i^* according to H_0.

Bootstrap test — discussion

- The resulting p-value depends on the exact T^*-values. Hence, it is recommended to repeat a bootstrap test a few times to see whether the p-value is stable. When B is too small, there is a lot of variation in the p-value. In that case B should be increased. In most cases $B = 1000$ is adequate.
- A bootstrap test can be performed with any test statistic. E.g. in the example taking `min` as a test statistic yields a bootstrap p-value of about 0.19 (check this yourselves!) and does not lead to rejecting H_0.
- The difference between the simulation of T^*-values for bootstrap confidence intervals and bootstrap tests is in the way the $X_1^*, \ldots, X_N^*$ are generated. For confidence intervals you choose X_i^* from your sample, whereas for tests you generate X_i^* according to H_0.

to finish

To wrap up

Today we saw:

1. recap distributions
2. bootstrap confidence intervals
3. statistical tests
4. bootstrap tests

Next time: 1 sample tests, 2 sample tests