Experimental Methods in Computer Science

(*Metodologias Experimentais em Informática)*

Henrique Madeira

Master in Informatics Engineering
Departamento de Engenharia Informática
Faculdade de Ciências e Tecnologia da Universidade de Coimbra
2023/2024

1

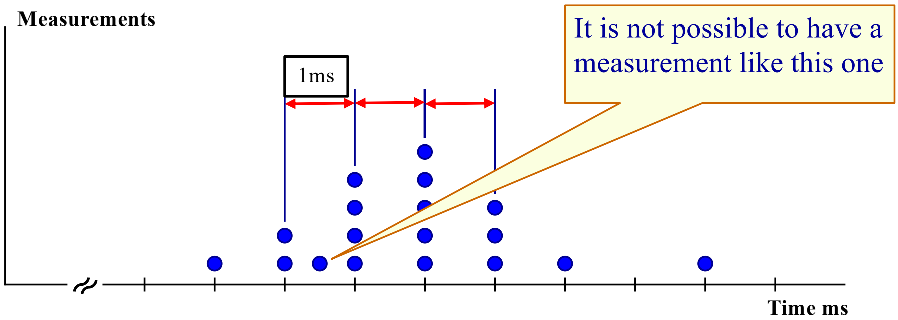
Measurements and confidence intervals

2

Resolution

Resolution of the measuring instrument: the smallest difference between measurements provided by a measuring device

Example: measuring execution time of a program in milliseconds



Henrique Madeira, DEI-FCTUC, 2018-2023

Experimental Methods in Computer Science, Master in Informatics Engineering , DEI-FCTUC, 2023/2024 3

3

Uncertainty

Uncertainty of the measurement: if we repeat a measurement, we will get slightly different results. Reflects the lack of **precision** of the measurement

Two types of uncertainties (leading to errors):

- **Random uncertainties**
  Variations in the measurements that occur without a predictable pattern.
- **Systematic uncertainties**
  Variations that consistently cause the measured value to be smaller or larger that the exact value.

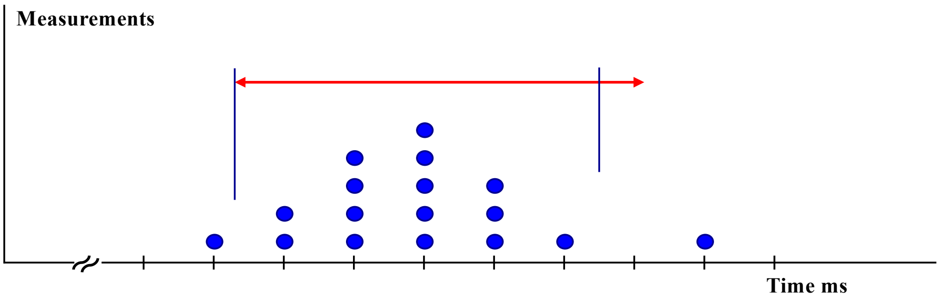
Henrique Madeira, DEI-FCTUC, 2018-2023

Experimental Methods in Computer Science, Master in Informatics Engineering , DEI-FCTUC, 2023/2024 4

4

Random uncertainties

- Occur without a predictable pattern.
- Can be reduced but never eliminated.
- Must be statistically analyzed and reported in the measurement process.



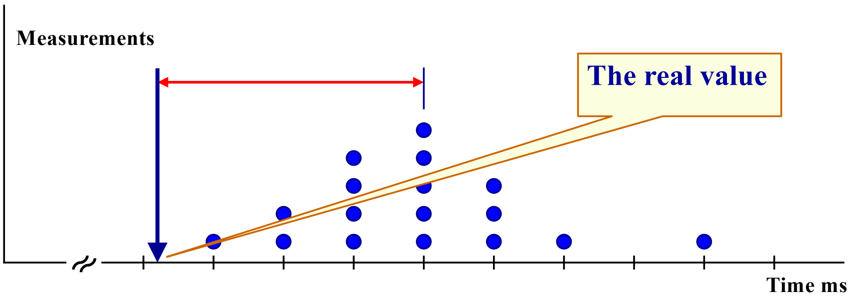
Henrique Madeira, DEI-FCTUC, 2018-2023

Experimental Methods in Computer Science, Master in Informatics Engineering , DEI-FCTUC, 2023/2024 5

5

Systematic uncertainties

- Systematic deviations from the real value
- Due to many possible causes (e.g., inaccurate measuring tools, miscalibration, tool reaction time/delay, etc)
- Once identified, can be eliminated (it is one of the steps in experiment design)



Henrique Madeira, DEI-FCTUC, 2018-2023

Experimental Methods in Computer Science, Master in Informatics Engineering , DEI-FCTUC, 2023/2024 6

6

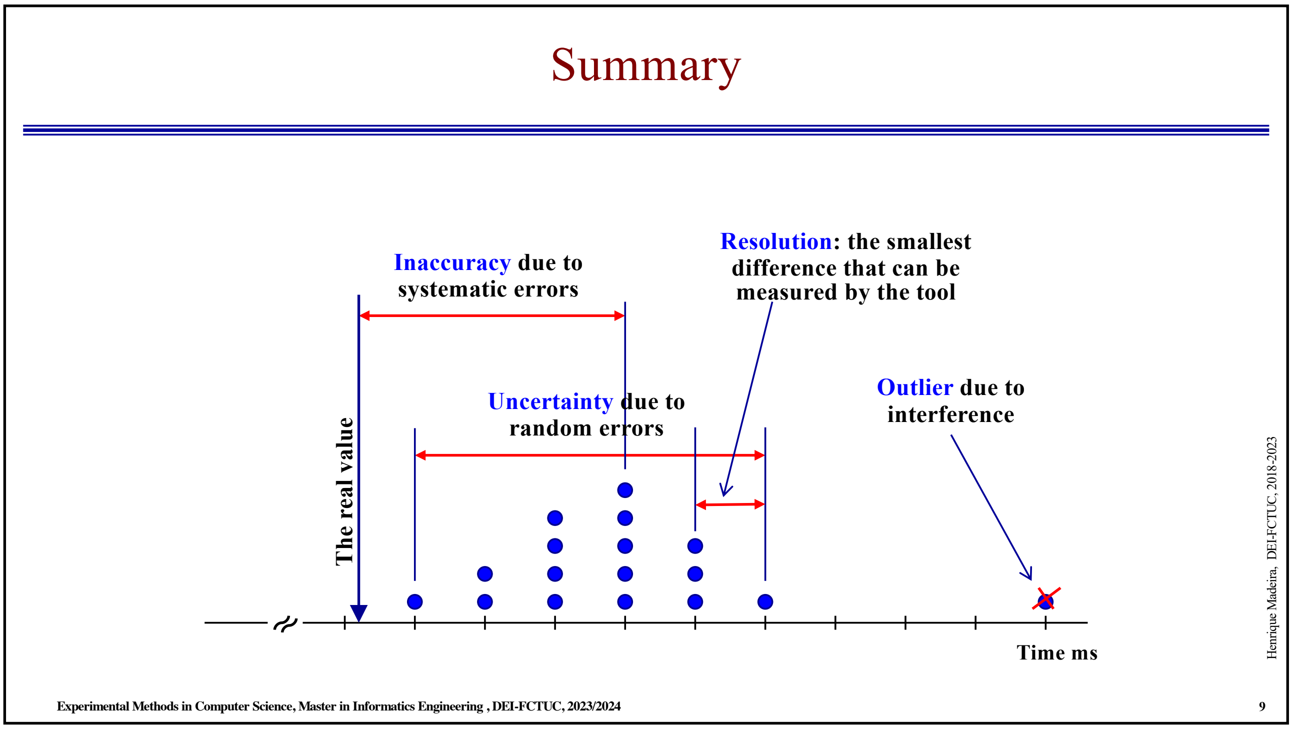
Systematic uncertainties: special cases

- **Warm-up:** the first measurement could be different from subsequent ones
- **Ramp-up:** it is necessary a set of measurements to reach stable values
- **Hysteresis**: the outcome depends on previous measurements (history)

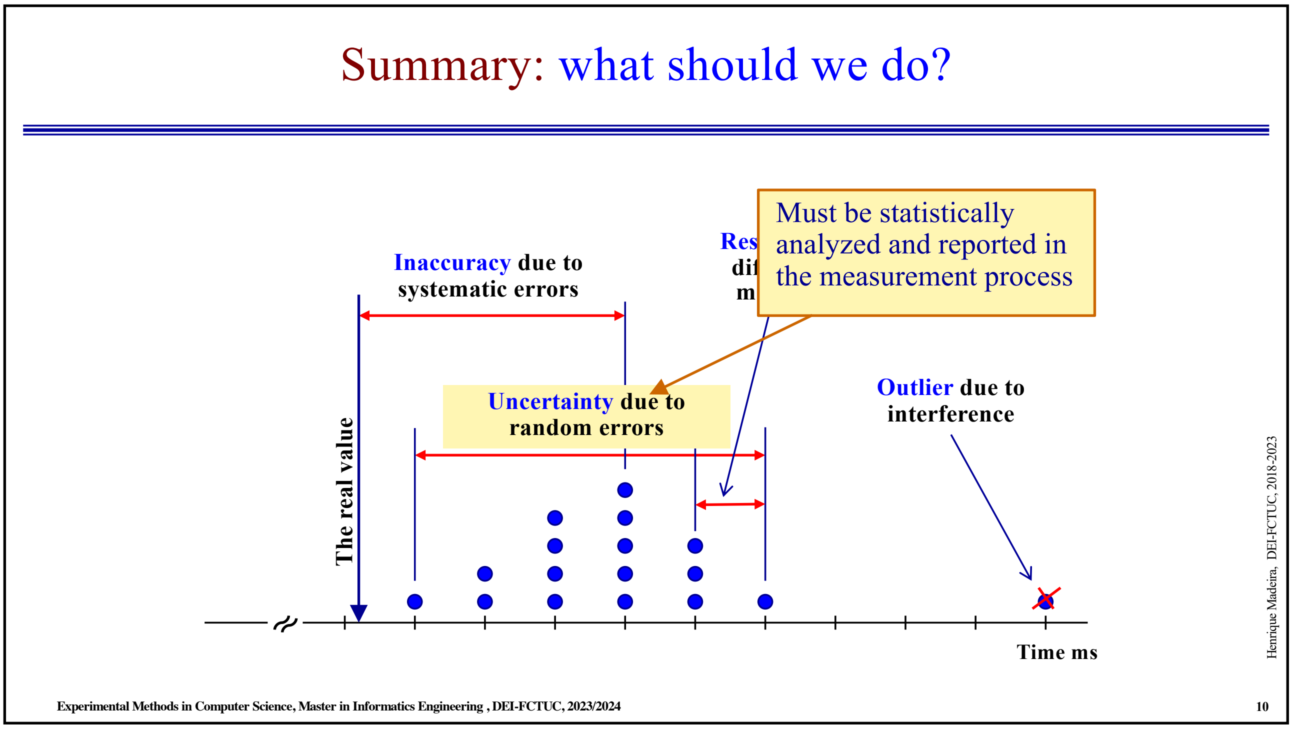
Variability

The variability in the measurements can result from two different sources:

- **Precision limitations of the measuring instrument.**
  - Even if the experiment conditions were totally stable, the different measurements would show slightly different values.
- **Changes in the conditions of the measurements (experiment environment, handling techniques, etc.).**
  - For examples, small changes in the load of a computer, cache state, available network bandwidth, etc., in the different measurements.
  - Quite often, the small changes in the experiment environment are analyzed statistically as random uncertainties.
  - Extreme cases lead to outliers (that should be ignored)

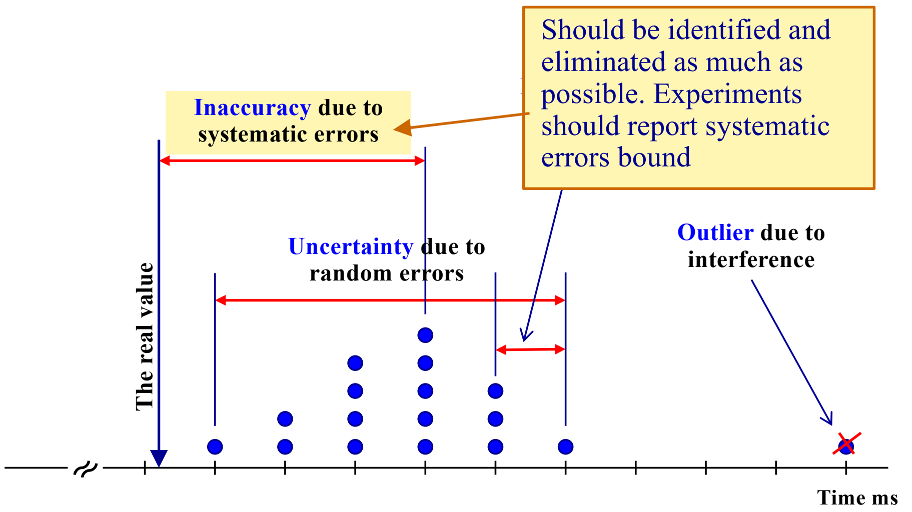

Summary
Inaccuracy due to systematic errors
Resolution: the smallest difference that can be measured by the tool
Uncertainty due to random errors
Outlier due to interference
The real value
Time ms
Henrique Madeira, DEI-FCTUC, 2018-2023
Experimental Methods in Computer Science, Master in Informatics Engineering , DEI-FCTUC, 2023/2024
9

9


Summary: what should we do?
Must be statistically analyzed and reported in the measurement process
Inaccuracy due to systematic errors
Uncertainty due to random errors
Outlier due to interference
The real value
Time ms
Henrique Madeira, DEI-FCTUC, 2018-2023
Experimental Methods in Computer Science, Master in Informatics Engineering , DEI-FCTUC, 2023/2024
10

10

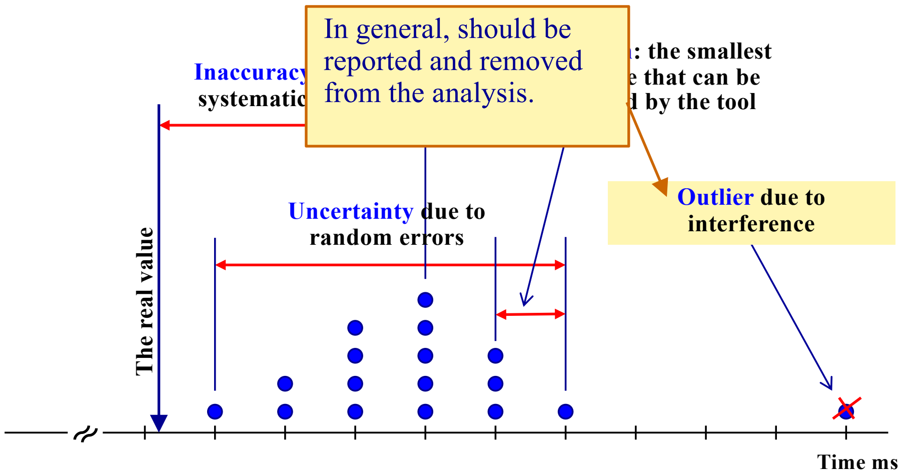
Summary: what should we do?



Experimental Methods in Computer Science, Master in Informatics Engineering , DEI-FCTUC, 2023/2024 11

11

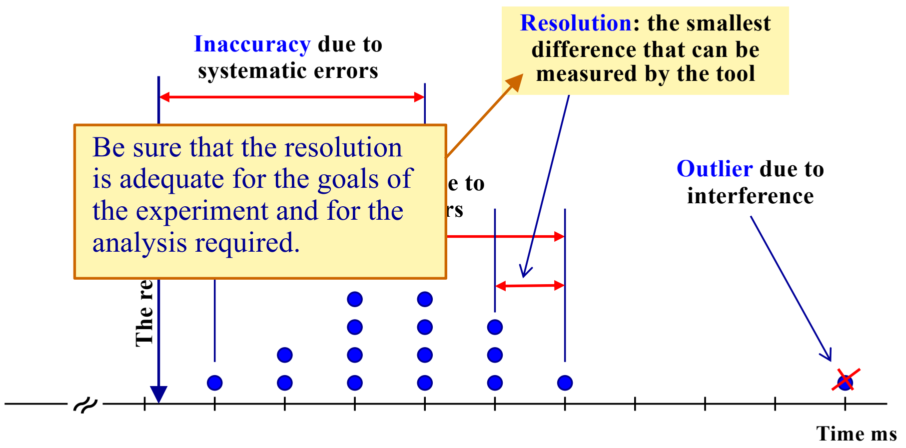
Summary: what should we do?



Experimental Methods in Computer Science, Master in Informatics Engineering , DEI-FCTUC, 2023/2024 12

12

Summary: what should we do?



Henrique Madeira, DEI-FCTUC, 2018-2023

Experimental Methods in Computer Science, Master in Informatics Engineering , DEI-FCTUC, 2023/2024 13

13

Example: measuring time in a computer

Several time sources:

- **Timer interrupts**
  Cause periodic CPU interrupts and run the clock interrupt handler that keeps the system time (human readable). Reasonably accurate. Maximal resolution is microseconds
- **Time stamp counter**
  A special register that counts the cycles since the machine was booted. Depends on CPU clock rate, which may change, e.g., to save energy in laptops. May drift, depending on temperature. Nanosecond resolution (but need many processor cycle to take a reading)
- **Time server and NTP (Network Time Protocol)**
  Gets time from a standard source, for clock synchronization in a network. May lead to a jump in time, forward or backwards…
- **Other** (system specific)…

Henrique Madeira, DEI-FCTUC, 2018-2023

Experimental Methods in Computer Science, Master in Informatics Engineering , DEI-FCTUC, 2023/2024 14

14

Example: Linux gettimeofday()

- Updated notion of real time as per the external source
- Synchronize with the time stamp counter
- When called, read the current time stamp counter and extrapolate from the previous clock interrupt
- Combines different timing sources
- Report result in microsecond resolution

Example: simple measurement

Goals:

- Measurement of some computer activity or operation. For examples, sorting a given number of items
- Done from user level
- With no specialized and/or external tools

Example: simple measurement (cont.)

Alternative 1:

```
t1 = gettimeofday();
<operation being measured>
t2 = gettimeofday();
print ”execution time was ", t2 – t1, "\n";
```

- Potential problems:
  - Inaccuracy due to the measurement overhead
  - The error is highly relevant if the execution time of the “operation being measured” is of similar range as the execution time of gettimeofday();

Example: simple measurement (cont.)

Alternative 2 – multiple measurements + buffering:

```
for (i=0; i<N; i++) {
        t1 = gettimeofday();
        <operation being measured>
        t2 = gettimeofday();
        time[i] = t2 - t1;
}
print ”average execution time is", avg(time[0.. N-1]), "\n”;
```

- Pros & cons:
  - The average is good (for large enough N)
  - Avoids the overhead of printing, which is normally heavy
  - May have resolution problems if the execution time of the “operation being measured” is of similar range as the execution time of gettimeofday();

Example: simple measurement (cont.)

Alternative 3 – multiple executions of the task:

```
t1 = gettimeofday();
for (i=0; i<N; i++) {
        <operation being measured>
}
t2 = gettimeofday();
print ”average execution time is", (t2 – t1)/N, "\n”;
```

- Problems:
  - Need to subtract the loop overhead
  - Running an empty loop to find the loop overhead may not work…
    Depends on the compiler optimization settings, scope of variables, etc.

Example: simple measurement (cont.)

Alternative 4 – multiple executions + unrolling:

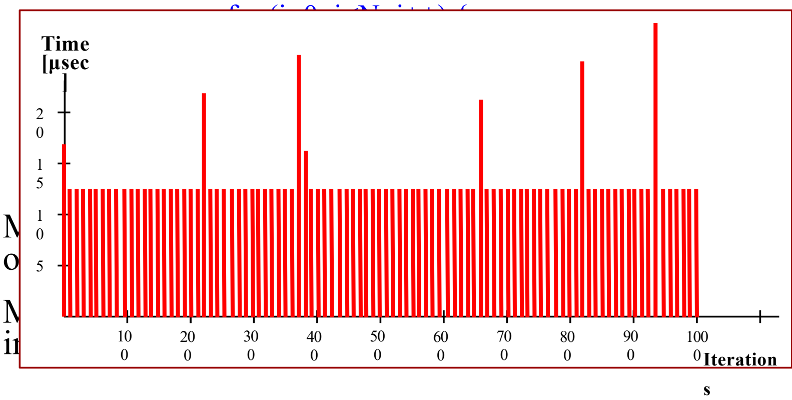
```
t1 = gettimeofday();
for (i=0; i<N/3; i++) {
        <operation being measured>
        <operation being measured>
        <operation being measured>
}
t2 = gettimeofday();
print ”average execution time is", (t2 – t1)/N, "\n”;
```

- May solve the problem when the execution time of the operation being measured is at similar range as the execution time of gettimeofday();
- N should be big enough to pass resolution limit and average out random errors.

Example: simple measurement (cont.)

Alternative 5 – double look to catch outliers:

for (r=0; r<REP; r++) {
t1 = gettimeofday();



n”;

- M
o
- M
ii

Henrique Madeira, DEI-FCTUC, 2018-2023

21

Example: simple measurement (cont.)

Alternative 5 – double look to catch outliers:

```
for (r=0; r<REP; r++) {
        t1 = gettimeofday();
        for (i=0; i<N; i++) {
                <operation being measured>
        }
        t2 = gettimeofday();
        print ”average execution time is", (t2 – t1)/N, "\n”;
}
```

- Multiple measurements in the double loop will catch outliers.
- May be frequent when the operation being measured involves disk, database or network accesses.

Henrique Madeira, DEI-FCTUC, 2018-2023

22

Example: simple measurement (cont.)

Alternative 5 – double look to catch outliers:

```
for (r=0; r<REP; r++) {
"\n";
}
```

There are many other alternatives to improve the accuracy of measuring time in computers

- Multiple measurements in the double loop will catch outliers.
- May be frequent when the operation being measured involves disk, database or network accesses.

Henrique Madeira, DEI-FCTUC, 2018-2023

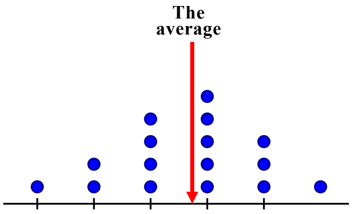
Confidence intervals (basics)

- When we perform multiple measurements of the same thing, we can calculate confidence intervals
- Assume measurements are samples from a (normal) distribution (real value + random error)
- Characterize the distribution dispersion
- Find the range that includes the desired mass of the probability density (e.g. 90%)

Henrique Madeira, DEI-FCTUC, 2018-2023

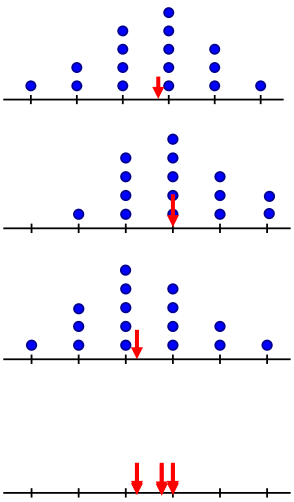
Confidence intervals

- Assume a set of measurements come from a normal distribution (real value + random error)
- This set has an average, which is an estimate of the real value
- If we repeat this with different samples, we will get a slightly different average



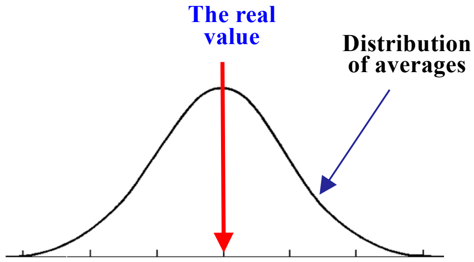
Confidence intervals (cont.)

- Multiple sets of samples induce multiple samples from the distribution of averages
- The distribution of averages is narrower than the base distribution
- So it gives a tighter estimate of the real value



Confidence intervals (cont.)

- **Assumption**: the averages reflect a true value plus some random error/noise
- Thus, the averages are distributed around the true value



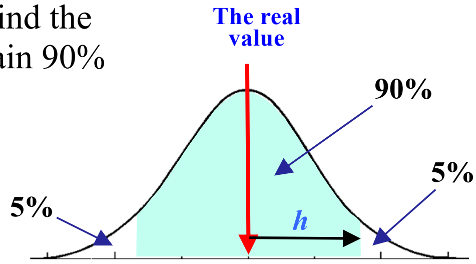
Henrique Madeira, DEI-FCTUC, 2018-2023

Experimental Methods in Computer Science, Master in Informatics Engineering , DEI-FCTUC, 2023/2024 27

27

Confidence intervals (cont.)

- **Assumption**: the averages reflect a true value plus some random error/noise
- Thus, the averages are distributed around the true value
- Given the distribution, we can find the range ***h*** that is expected to contain 90% of the averages (90% is just an example)



Henrique Madeira, DEI-FCTUC, 2018-2023

Experimental Methods in Computer Science, Master in Informatics Engineering , DEI-FCTUC, 2023/2024 28

28

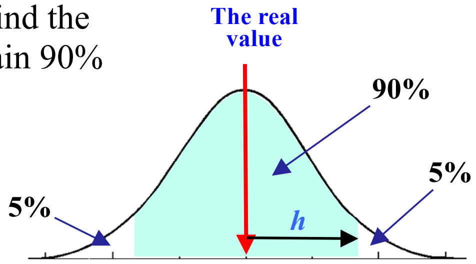
Confidence intervals (cont.)

- Assumption: the averages reflect a true ... noise
- ... ted
- ... find the ... tain 90% ... example)

For 90% of the averages, the true value is within *h*

or

the range average ± *h* has probability 0.9 to include the real value



Henrique Madeira, DEI-FCTUC, 2018-2023

Calculate confidence intervals

- Let μ denote the real mean of the base distribution
- Let $\bar{x}$ denote the average of n samples
- If the base distribution is normal, then the averages have a ***t*** distribution or **Z** distribution when sample is large (n ≥ 30)
- Let α denote the acceptable uncertainty (imply that the level of confidence is 1 – α) and define the half-width as

$$h = t_{n-1,1-\alpha/2} s_{\bar{x}}$$

Then

$$p(|\bar{x} - \mu| < h) = 1 - \alpha$$

Henrique Madeira, DEI-FCTUC, 2018-2023

Calculate confidence intervals

- Let μ den
- Let $\overline{x}$ den
- If the base distributio
- Let α den of confidence is 1 ... define the half-width as

- $t_{n-1,1-\alpha/2}$ comes from tables (**t** tables for n ≤ 30)
  - *n* is the number of samples
  - *n – 1* degrees of freedom (for the **t** Student distrib.)

  (for $n \geq 30$ use the z table, normal distribution)
- $s_{\overline{x}}$ is the standard deviation of the averages. Assuming the base samples are independent, this can be calculated as $s/\sqrt{n}$, where s is the standard deviation of the samples

$$h = t_{n-1,1-\alpha/2} s_{\overline{x}}$$

Then

$$p(|\overline{x} - \mu| < h) = 1 - \alpha$$

Henrique Madeira, DEI-FCTUC, 2018-2023

31

Calculate confidence intervals (cont.)

- Let μ de
- Let $\overline{x}$ de
- If the ba distribut
- Let α de ... of confidence is 1 – α) and define the ... as

The confidence interval:

$$p(|\overline{x} - \mu| < h) = 1 - \alpha$$

with a certainty of 1-α, the distance between a sample of the average $\overline{x}$ and the true mean μ is less than *h*

If we repeat this many times, and each time we draw a segment of $\pm h$ around $\overline{x}$, then in 1-α of the cases this segment will include μ

$$h = t_{n-1,1-\alpha/2} s_{\overline{x}}$$

Then

$$p(|\overline{x} - \mu| < h) = 1 - \alpha$$

Henrique Madeira, DEI-FCTUC, 2018-2023

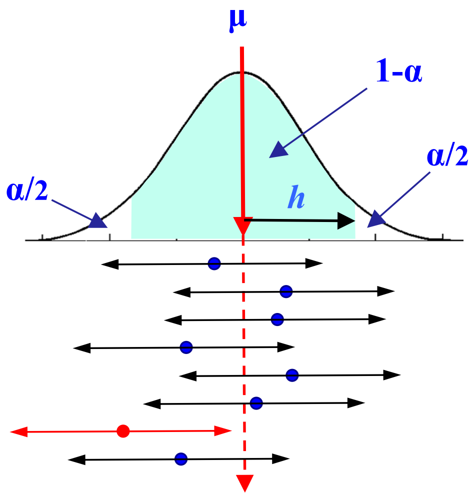
32

Calculate confidence intervals (cont.)

With a certainty of $1\text{-}\alpha$ the distance between a <u>sample</u> of the average $\bar{x}$ and the true mean μ is less than h

or

If we repeat a measurement many times, and each time we draw a segment of $\pm h$ around $\bar{x}$, then in $1\text{-}\alpha$ of the cases this segment will include μ



Henrique Madeira, DEI-FCTUC, 2018-2023

33

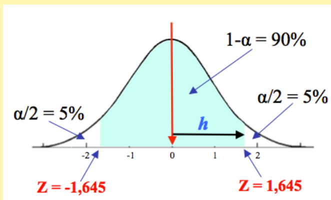
Calculate confidence intervals (cont.)

In practice, assuming the base samples are independent, the formula is:

$\bar{x} \pm t * s/\sqrt{n}$ (for $n \leq 30$, use t table with df = n-1)

or

$\bar{x} \pm z * s/\sqrt{n}$ (for n ≥ 30, use z table for standard normal distribution)



Where:

- s is the standard deviation of the n samples
- For example, for α = 0.1 the value z = 1,645. It represents the point in the axis where the area under the standard normal curve is 1 – α (i.e., 90% for α = 0.1)

Henrique Madeira, DEI-FCTUC, 2018-2023

34

Calculate confidence intervals (cont.)

Assumptions:

- The base samples come from a normal distribution
  If not, but have a finite variance, the averages will still be normal, but this will require a larger *n*
- Base samples are independent
  If not, maybe using larger batches will reduce the correlation between them
- If the number of samples is small ($n \leq 30$) we assume a **t** Student distribution

Henrique Madeira, DEI-FCTUC, 2018-2023

Experimental Methods in Computer Science, Master in Informatics Engineering , DEI-FCTUC, 2023/2024 35

35

Calculate confidence intervals (cont.)

Assumptions:

- The b…
  If not… will
  requi…
- Base…
  If not… n
- If the…
  Stud…

In practice, before computing confidence intervals:

- Clean up the data first
- Remove outliers that indicate interference or spurious measurements. For example:
  - remove top and bottom measurements;
  - look at the data and decide outliers to be removed
- Remove warm-up and history effects

Henrique Madeira, DEI-FCTUC, 2018-2023

Experimental Methods in Computer Science, Master in Informatics Engineering , DEI-FCTUC, 2023/2024 36

36

How to find the value Z?

Example: what is the confidence coefficient Z for α = 5%? (two-tailed test)

1. Subtract α from 1
   1 – 0.05 = 0.95

2. Divide result by 2 (because it is two-tailed)
   0.95/2 = 0.475

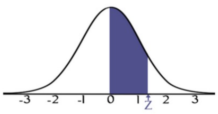
3. Look at the z-table and locate the results from Step 2 (0.475) in the table.
   The closest value for the coefficient Z is at the intersection of row 1.9 and the column of 0.06. Adding up these two values comes that Z = 1,96 for α = 5%

Henrique Madeira, DEI-FCTUC, 2018-2023

Experimental Methods in Computer Science, Master in Informatics Engineering , DEI-FCTUC, 2023/2024 37

37

How to f



STANDARD NORMAL TABLE (Z)

Entries in the table give the area under the curve between the mean and *z* standard deviations above the mean. For example, for *z* = 1.25 the area under the curve between the mean (0) and *z* is 0.3944.

z	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0.0	0.0000	0.0040	0.0080	0.0120	0.0160	0.0190	0.0239	0.0279	0.0319	0.0359
0.1	0.0398	0.0438	0.0478	0.0517	0.0557	0.0596	0.0636	0.0675	0.0714	0.0753
0.2	0.0793	0.0832	0.0871	0.0910	0.0948	0.0987	0.1026	0.1064	0.1103	0.1141
0.3	0.1179	0.1217	0.1255	0.1293	0.1331	0.1368	0.1406	0.1443	0.1480	0.1517
0.4	0.1554	0.1591	0.1628	0.1664	0.1700	0.1736	0.1772	0.1808	0.1844	0.1879
0.5	0.1915	0.1950	0.1985	0.2019	0.2054	0.2088	0.2123	0.2157	0.2190	0.2224
0.6	0.2257	0.2291	0.2324	0.2357	0.2389	0.2422	0.2454	0.2486	0.2517	0.2549
0.7	0.2580	0.2611	0.2642	0.2673	0.2704	0.2734	0.2764	0.2794	0.2823	0.2852
0.8	0.2881	0.2910	0.2939	0.2969	0.2995	0.3023	0.3051	0.3078	0.3106	0.3133
0.9	0.3159	0.3186	0.3212	0.3238	0.3264	0.3289	0.3315	0.3340	0.3365	0.3389
1.0	0.3413	0.3438	0.3461	0.3485	0.3508	0.3513	0.3554	0.3577	0.3529	0.3621
1.1	0.3643	0.3665	0.3686	0.3708	0.3729	0.3749	0.3770	0.3790	0.3810	0.3830
1.2	0.3849	0.3869	0.3888	0.3907	0.3925	0.3944	0.3962	0.3980	0.3997	0.4015
1.3	0.4032	0.4049	0.4066	0.4082	0.4099	0.4115	0.4131	0.4147	0.4162	0.4177
1.4	0.4192	0.4207	0.4222	0.4236	0.4251	0.4265	0.4279	0.4292	0.4306	0.4319
1.5	0.4332	0.4345	0.4357	0.4370	0.4382	0.4394	0.4406	0.4418	0.4429	0.4441
1.6	0.4452	0.4463	0.4474	0.4484	0.4495	0.4505	0.4515	0.4525	0.4535	0.4545
1.7	0.4554	0.4564	0.4573	0.4582	0.4591	0.4599	0.4608	0.4616	0.4625	0.4633
1.8	0.4641	0.4649	0.4656	0.4664	0.4671	0.4678	0.4686	0.4693	0.4699	0.4706
1.9	0.4713	0.4719	0.4726	0.4732	0.4738	0.47[illegible]	0.4750	0.4756	0.4761	0.4767
2.0	0.4772	0.4778	0.4783	0.4788	0.4793	0.4798	0.4803	0.4808	0.4812	0.4817
2.1	0.4821	0.4826	0.4830	0.4834	0.4838	0.4842	0.4846	0.4850	0.4854	0.4857
2.2	0.4861	0.4864	0.4868	0.4871	0.4875	0.4878	0.4881	0.4884	0.4887	0.4890
2.3	0.4893	0.4896	0.4898	0.4901	0.4904	0.4906	0.4909	0.4911	0.4913	0.4916
2.4	0.4918	0.4920	0.4922	0.4925	0.4927	0.4929	0.4931	0.4932	0.4934	0.4936
2.5	0.4938	0.4940	0.4941	0.4943	0.4945	0.4946	0.4948	0.4949	0.4951	0.4952
2.6	0.4953	0.4955	0.4956	0.4957	0.4959	0.4960	0.4961	0.4962	0.4963	0.4964
2.7	0.4965	0.4966	0.4967	0.4968	0.4969	0.4970	0.4971	0.4972	0.4973	0.4974
2.8	0.4974	0.4975	0.4976	0.4977	0.4977	0.4978	0.4979	0.4979	0.4980	0.4981
2.9	0.4981	0.4982	0.4982	0.4983	0.4984	0.4984	0.4985	0.4985	0.4986	0.4986
3.0	0.4987	0.4987	0.4987	0.4988	0.4988	0.4989	0.4989	0.4989	0.4990	0.4990
3.1	0.4990	0.4991	0.4991	0.4991	0.4992	0.4992	0.4992	0.4992	0.4993	0.4993
3.2	0.4993	0.4993	0.4994	0.4994	0.4994	0.4994	0.4994	0.4995	0.4995	0.4995
3.3	0.4995	0.4995	0.4995	0.4996	0.4996	0.4996	0.4996	0.4996	0.4996	0.4997
3.4	0.4997	0.4997	0.4997	0.4997	0.4997	0.4997	0.4997	0.4997	0.4997	0.4998

Example: what is the conf (two-tailed test)

1. Subtract α from 1
   1 – 0.05 = 0.95

2. Divide result by 2 (becaus
   0.95/2 = 0.475

3. Look at the z-table and lo the table.
   The closest value for the co and the column of 0.06. Add 1,96 for α = 5%

Henrique Madeira, DEI-FCTUC, 2018-2023

Experimental Methods in Computer Science, Master in Informatics Engineering , DEI-FCTUC, 202 38

38

Common confidence levels and values of Z

Confidence Level	Z
0.70	1.04
0.75	1.15
0.80	1.28
0.85	1.44
0.90	1.645
0.91	1.70
0.92	1.75
0.93	1.81
0.94	1.88
0.95	1.96
0.96	2.05
0.97	2.17
0.98	2.33
0.99	2.575

Henrique Madeira, DEI-FCTUC, 2018-2023

Example of confidence intervals computation

Assume you are measuring the execution time of a given program. You repeat the program execution with different loads and in different moments, in the same computer.

$$\bar{x} \pm z * s/\sqrt{n}$$

Exec. Time (msec)	
2711	2634
2673	3275
3533	2580
2867	3353
3392	2950
2864	3452
3274	3449
3322	2542
2884	2419
3569	3538
3484	3290
3198	3290
2879	3290
3281	3290
3347	3290
2960	3290

	90%	99%
n of samples	32	32
Z	1.65	2.575
S (std dev)	330.51	330.51
average	3130.31	3130.31
Confidence interval	96.11	150.45
Exec. time minimum	3034.20	2979.86
Exec. time maximum	3226.42	3280.76

Execution time (95%) = 3130.31 ± 96.11
Execution time (99%) = 3130.31 ± 150.45

Henrique Madeira, DEI-FCTUC, 2018-2023

Example of confidence intervals computation

Assume you are measuring the execution
time c
progr
differ

Confidence interval (CI):

For small samples use the Student's ***t*** distribution. That is, for $n \leq 30$ use ***t*** table with $df = n-1$. The standard deviation ***s*** is taken from the ***n*** samples

$$\bar{x} \pm t * s/\sqrt{n}$$

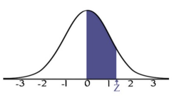
For large samples use the standard normal distribution. That is, for $n > 30$ use the ***z*** table. The standard deviation ***s*** is taken from the ***n*** samples

$$\bar{x} \pm z * s/\sqrt{n}$$

2960	3290

Henrique Madeira, DEI-FCTUC, 2018-2023

Examples of statistic table



STANDARD NORMAL TABLE (Z)

Entries in the table give the area under the curve between the mean and z standard deviations above the mean. For example, for $z = 1.25$ the area under the curve between the mean (0) and z is 0.3944.

z	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0.0	0.0000	0.0040	0.0080	0.0120	0.0160	0.0190	0.0239	0.0279	0.0319	0.0359
0.1	0.0398	0.0438	0.0478	0.0517	0.0557	0.0596	0.0636	0.0675	0.0714	0.0753
0.2	0.0793	0.0832	0.0871	0.0910	0.0948	0.0987	0.1026	0.1064	0.1103	0.1141
0.3	0.1179	0.1217	0.1255	0.1293	0.1331	0.1368	0.1406	0.1443	0.1480	0.1517
0.4	0.1554	0.1591	0.1628	0.1664	0.1700	0.1736	0.1772	0.1808	0.1844	0.1879
0.5	0.1915	0.1950	0.1985	0.2019	0.2054	0.2088	0.2123	0.2157	0.2190	0.2224
0.6	0.2257	0.2291	0.2324	0.2357	0.2389	0.2422	0.2454	0.2486	0.2517	0.2549
0.7	0.2580	0.2611	0.2642	0.2673	0.2704	0.2734	0.2764	0.2794	0.2823	0.2852
0.8	0.2881	0.2910	0.2939	0.2969	0.2995	0.3023	0.3051	0.3078	0.3106	0.3133
0.9	0.3159	0.3186	0.3212	0.3238	0.3264	0.3289	0.3315	0.3340	0.3365	0.3389
1.0	0.3413	0.3438	0.3461	0.3485	0.3508	0.3513	0.3554	0.3577	0.3529	0.3621
1.1	0.3643	0.3665	0.3686	0.3708	0.3729	0.3749	0.3770	0.3790	0.3810	0.3830
1.2	0.3849	0.3869	0.3888	0.3907	0.3925	0.3944	0.3962	0.3980	0.3997	0.4015
1.3	0.4032	0.4049	0.4066	0.4082	0.4099	0.4115	0.4131	0.4147	0.4162	0.4177
1.4	0.4192	0.4207	0.4222	0.4236	0.4251	0.4265	0.4279	0.4292	0.4306	0.4319
1.5	0.4332	0.4345	0.4357	0.4370	0.4382	0.4394	0.4406	0.4418	0.4429	0.4441
1.6	0.4452	0.4463	0.4474	0.4484	0.4495	0.4505	0.4515	0.4525	0.4535	0.4545
1.7	0.4554	0.4564	0.4573	0.4582	0.4591	0.4599	0.4608	0.4616	0.4625	0.4633
1.8	0.4641	0.4649	0.4656	0.4664	0.4671	0.4678	0.4686	0.4693	0.4699	0.4706
1.9	0.4713	0.4719	0.4726	0.4732	0.4738	0.4744	0.4750	0.4756	0.4761	0.4767
2.0	0.4772	0.4778	0.4783	0.4788	0.4793	0.4798	0.4803	0.4808	0.4812	0.4817
2.1	0.4821	0.4826	0.4830	0.4834	0.4838	0.4842	0.4846	0.4850	0.4854	0.4857
2.2	0.4861	0.4864	0.4868	0.4871	0.4875	0.4878	0.4881	0.4884	0.4887	0.4890
2.3	0.4893	0.4896	0.4898	0.4901	0.4904	0.4906	0.4909	0.4911	0.4913	0.4916
2.4	0.4918	0.4920	0.4922	0.4925	0.4927	0.4929	0.4931	0.4932	0.4934	0.4936
2.5	0.4938	0.4940	0.4941	0.4943	0.4945	0.4946	0.4948	0.4949	0.4951	0.4952
2.6	0.4953	0.4955	0.4956	0.4957	0.4959	0.4960	0.4961	0.4962	0.4963	0.4964
2.7	0.4965	0.4966	0.4967	0.4968	0.4969	0.4970	0.4971	0.4972	0.4973	0.4974
2.8	0.4974	0.4975	0.4976	0.4977	0.4977	0.4978	0.4979	0.4979	0.4980	0.4981
2.9	0.4981	0.4982	0.4982	0.4983	0.4984	0.4984	0.4985	0.4985	0.4986	0.4986
3.0	0.4987	0.4987	0.4987	0.4988	0.4988	0.4989	0.4989	0.4989	0.4990	0.4990
3.1	0.4990	0.4991	0.4991	0.4991	0.4992	0.4992	0.4992	0.4992	0.4993	0.4993
3.2	0.4993	0.4993	0.4994	0.4994	0.4994	0.4994	0.4994	0.4995	0.4995	0.4995
3.3	0.4995	0.4995	0.4995	0.4996	0.4996	0.4996	0.4996	0.4996	0.4996	0.4997
3.4	0.4997	0.4997	0.4997	0.4997	0.4997	0.4997	0.4997	0.4997	0.4997	0.4998

Tables T-11

Table entry for p and C is the critical value t^* with probability p lying to its right and probability C lying between $-t^*$ and t^*.

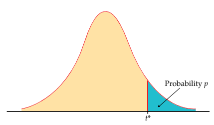


TABLE D

t distribution critical values

df	.25	.20	.15	.10	.05	.025	.02	.01	.005	.0025	.001	.0005
	Upper-tail probability p											
1	1.000	1.376	1.963	3.078	6.314	12.71	15.89	31.82	63.66	127.3	318.3	636.6
2	0.816	1.061	1.386	1.886	2.920	4.303	4.849	6.965	9.925	14.09	22.33	31.60
3	0.765	0.978	1.250	1.638	2.353	3.182	3.482	4.541	5.841	7.453	10.21	12.92
4	0.741	0.941	1.190	1.533	2.132	2.776	2.999	3.747	4.604	5.598	7.173	8.610
5	0.727	0.920	1.156	1.476	2.015	2.571	2.757	3.365	4.032	4.773	5.893	6.869
6	0.718	0.906	1.134	1.440	1.943	2.447	2.612	3.143	3.707	4.317	5.208	5.959
7	0.711	0.896	1.119	1.415	1.895	2.365	2.517	2.998	3.499	4.029	4.785	5.408
8	0.706	0.889	1.108	1.397	1.860	2.306	2.449	2.896	3.355	3.833	4.501	5.041
9	0.703	0.883	1.100	1.383	1.833	2.262	2.398	2.821	3.250	3.690	4.297	4.781
10	0.700	0.879	1.093	1.372	1.812	2.228	2.359	2.764	3.169	3.581	4.144	4.587
11	0.697	0.876	1.088	1.363	1.796	2.201	2.328	2.718	3.106	3.497	4.025	4.437
12	0.695	0.873	1.083	1.356	1.782	2.179	2.303	2.681	3.055	3.428	3.930	4.318
13	0.694	0.870	1.079	1.350	1.771	2.160	2.282	2.650	3.012	3.372	3.852	4.221
14	0.692	0.868	1.076	1.345	1.761	2.145	2.264	2.624	2.977	3.326	3.787	4.140
15	0.691	0.866	1.074	1.341	1.753	2.131	2.249	2.602	2.947	3.286	3.733	4.073
16	0.690	0.865	1.071	1.337	1.746	2.120	2.235	2.583	2.921	3.252	3.686	4.015
17	0.689	0.863	1.069	1.333	1.740	2.110	2.224	2.567	2.898	3.222	3.646	3.965
18	0.688	0.862	1.067	1.330	1.734	2.101	2.214	2.552	2.878	3.197	3.611	3.922
19	0.688	0.861	1.066	1.328	1.729	2.093	2.205	2.539	2.861	3.174	3.579	3.883
20	0.687	0.860	1.064	1.325	1.725	2.086	2.197	2.528	2.845	3.153	3.552	3.850
21	0.686	0.859	1.063	1.323	1.721	2.080	2.189	2.518	2.831	3.135	3.527	3.819
22	0.686	0.858	1.061	1.321	1.717	2.074	2.183	2.508	2.819	3.119	3.505	3.792
23	0.685	0.858	1.060	1.319	1.714	2.069	2.177	2.500	2.807	3.104	3.485	3.768
24	0.685	0.857	1.059	1.318	1.711	2.064	2.172	2.492	2.797	3.091	3.467	3.745
25	0.684	0.856	1.058	1.316	1.708	2.060	2.167	2.485	2.787	3.078	3.450	3.725
26	0.684	0.856	1.058	1.315	1.706	2.056	2.162	2.479	2.779	3.067	3.435	3.707
27	0.684	0.855	1.057	1.314	1.703	2.052	2.158	2.473	2.771	3.057	3.421	3.690
28	0.683	0.855	1.056	1.313	1.701	2.048	2.154	2.467	2.763	3.047	3.408	3.674
29	0.683	0.854	1.055	1.311	1.699	2.045	2.150	2.462	2.756	3.038	3.396	3.659
30	0.683	0.854	1.055	1.310	1.697	2.042	2.147	2.457	2.750	3.030	3.385	3.646
40	0.681	0.851	1.050	1.303	1.684	2.021	2.123	2.423	2.704	2.971	3.307	3.551
50	0.679	0.849	1.047	1.299	1.676	2.009	2.109	2.403	2.678	2.937	3.261	3.496
60	0.679	0.848	1.045	1.296	1.671	2.000	2.099	2.390	2.660	2.915	3.232	3.460
80	0.678	0.846	1.043	1.292	1.664	1.990	2.088	2.374	2.639	2.887	3.195	3.416
100	0.677	0.845	1.042	1.290	1.660	1.984	2.081	2.364	2.626	2.871	3.174	3.390
1000	0.675	0.842	1.037	1.282	1.646	1.962	2.056	2.330	2.581	2.813	3.098	3.300
z^*	0.674	0.841	1.036	1.282	1.645	1.960	2.054	2.326	2.576	2.807	3.091	3.291
	50%	60%	70%	80%	90%	95%	96%	98%	99%	99.5%	99.8%	99.9%
	Confidence level C											

Henrique Madeira, DEI-FCTUC, 2018-2023

Inferences for proportion

- Very often in computer/software experiments the dependent variable has only two possible outcomes. For example:
  - Error detected **or** error not detected
  - Vulnerability detected **or** vulnerability not detected
  - Silent data corruption **or** no silent data corruption (either the corruption was detected or there was no corruption at all)
  - System crashed **or** system did not crash
  - Robust behavior of web service **or** non robust behavior
  - Test case succeed **or** test case failed
  - Message arrived within specified timeframe **or** arrived outside the specified timeframe
  - Safety behavior **or** non safety behavior
  - Etc, etc, etc

Henrique Madeira, DEI-FCTUC, 2018-2023

Experimental Methods in Computer Science, Master in Informatics Engineering , DEI-FCTUC, 2023/2024 43

43

Inferences for proportion

- Very often in computer/software experiments the dependent variable has only two possible outcomes. For example:
  - Error detected **or** error not detected
  - Vulnerability detected **or** vulnerability not detected
  - Silent data corruption **or** no silent data corruption (either the corruption was detected or there was no corruption at all)
  - System crashed **or** system did not crash
  - Robust behavior of web service **or** non robust behavior
  - Test case succeed **or** test case failed
  - Message arrived within specified timeframe **or** arrived outside the specified

The dependent variable is binary (two mutually exclusive outcomes). We can assume that a binomial distribution is a good approximation for these cases

  - Etc, etc, etc

Henrique Madeira, DEI-FCTUC, 2018-2023

Experimental Methods in Computer Science, Master in Informatics Engineering , DEI-FCTUC, 2023/2024 44

44

Binomial model

A binomial variable has the following properties:

- The variable is binary; it can take only one of two possible values.
- The variable is observed a known number of times (called **n**).
  - Each observation is often called a trial.
  - The number of times that the outcome of interest (e.g., error detection) is observed is **x**. It is often called the number of "**successes**" (in observing the outcome of interest).
- The probability that the outcome of interest occurs is the same for each trial.
- The trials are independent and the outcome of one trial does not affect the outcome of the any other trial.

Henrique Madeira, DEI-FCTUC, 2018-2023

Experimental Methods in Computer Science, Master in Informatics Engineering , DEI-FCTUC, 2023/2024 45

45

Binomial model

A binomial variable has the following properties:

- The variable is binary; it can take only one of two possible values.
- The variable is observed a known number of times (called **n**).
  - Each observation is often called a trial.
  - The number of times that the outcome of interest (e.g., error detection) is observed is **x**. It is often called the number of "**successes**" (in observing the outcome of interest).
- The probability that the outcome of interest occurs is the same for each trial.
- The trials are independent and the outcome of one trial does not affect the outcome of the any other trial.

These two bullets deserve some discussion.
Impact on the experiments to assure validity.

Henrique Madeira, DEI-FCTUC, 2018-2023

Experimental Methods in Computer Science, Master in Informatics Engineering , DEI-FCTUC, 2023/2024 46

46

Sampling distribution of the sample proportion

- Sample Proportion: $\hat{p} = \dfrac{x}{n}$ (**sample** = set of trials)
- $\hat{p}$ is the proportion of the sample with the outcome of interest. It is an estimate of the population proportion ***p***
- $\hat{p}$ varies from sample to sample in a random way
- For **large *n*** of samples the sampling distribution can be considered as a **normal distribution**. But the large number of samples should:
  - include a number of successes and non successes larger or equal to 10 (i.e., $np \geq 10$ **and** $n(1-p) \geq 10$);
  - be at least 20 times smaller than the population (i.e., population should be much larger)
- Consequently, we assume that the mean of the sampling distribution is approximately equal to the true population proportion ***p*.**

Henrique Madeira, DEI-FCTUC, 2018-2023

47

Sampling distribution of the sample proportion

- Sample Proportion: $\hat{p} = \dfrac{x}{n}$ (**sample** = set of trials)
- $\hat{p}$ is the proportion of the sample with the outcome of interest. It is an estimate of the population proportion ***p***
- $\hat{p}$ varies from sample to sample in a random way
- For **large *n*** of samples the sampling distribution can be considered as a **normal distribution**. But the large number of samples should:
  - include a number of successes and non successes larger or equal to 10 (i.e., $np \geq 10$ **and** $n(1-p) \geq 10$);
  - be at least 20 times smaller than the population (i.e., population should be much larger)
- Consequently, we assume that the mean of the sampling distribution is approximately equal to the true population proportion ***p*.**

Could be a problem in computer dependability and security experiments. Why?

Generally not a problem in computer experiments. Why?

Henrique Madeira, DEI-FCTUC, 2018-2023

48

Confidence Intervals (CI) for population proportion

Considering that for larger samples the sampling distribution of the sample proportion is approximately normal:

- The standard error (SE) of sample proportion is given by

$$SE(\hat{p}) = \sqrt{\frac{p(1-p)}{n}}$$

- The confidence interval (CI) for population proportion is

$$\hat{p} \pm z\sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \qquad \hat{p} \pm t\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

Hypothesis Testing

Hypothesis testing slides are mainly based on chapter 8 of the book "Essentials of Social Statistics for a Diverse Society" Second Edition by Anna Leon-Guerrero, Chava Frankfort-Nachmias , SAGE Publications, Inc, 2010.