Experimental Methods
in Computer Science (and Informatics Engineering)
2017-2018

Solution of some training exercises

Important note: The solutions provided here for some of the proposed exercises must be used by the students only after a serious attempt to solving the exercises (without looking at the solution). It is obviously much easier to understand a proposed solution than to come to solution by yourself, as it is required in the written exam.

Training exercises 1

3) One of the first steps in the design of an experiment is to define the problem statement (or research question). Give a concrete example of problem statement (i.e., provide the actual sentence that express the problem statement) and briefly explain the experiment context related to each problem statement.

A good (i.e., relevant) problem statement should be focused enough to allow the clear identification of the variables of the problem but, at the same time, should be sufficiently open to allow different hypothesis to answer the problem/question. For example:

Is the time necessary to sort a given number of items in an array mainly dependent on the number of items to be sorted or the size of the items plays a major role as well?

The context of this research question is performance of sorting algorithms and programs.

4) For the example of problem statement provided in your answer to the previous question, indicate the following:

a) Dependent variable(s)

- Sorting time

b) Independent variables

- Size of the array
- Size of each element of the array (or average size)
- Sorting algorithm
- Programming language
- Discuss other: computer, operating system, etc.

c) Examples of possible levels for the independent variables in the experiments

- Size of the array → 100, 10.000, 100.000, 1,000.000, 100.000.000
- Size of each element of the array (or average size) → 1, 2, 3, 5, 10, 20, 50, 100, 200
- Sorting algorithm → Quick sort, merge sort, shell sort
- Programming language → C, Java, Python

d) Hypothesis that could be tested (indicate if the hypothesis is directional or non-directional).

H0 – The size of the items to be sorted have no impact on the sorting time
H1 – The size of the items to be sorted has impact on the sorting time

5) An engineer conducts a hypothesis test and concludes that his hypothesis is correct. Explain why this conclusion is never an appropriate decision in hypothesis testing.

The goal of hypothesis testing is to prove that the null hypothesis is not true (i.e., to reject the null hypothesis), with a given level of confidence. It is not possible to prove that the null hypothesis is correct; you can only prove that it should be rejected.

7) Outliers obtained in the measurements should be reported but, in general, are removed from the analysis. Explain what should be taken into account in the decision of ignoring or not the outliers.

Outliers can be caused by random variability in the measurements, by errors in the design and implementation of the experiments, by features of the experimental setup, or may simply indicate something scientifically and technically interesting. The most important thing is to determine if the outliers represent bad data (and in that case, they should be removed from the analysis) oy represent something relevant (although unusual). The analyst should try to systematically identify all the possible outliers causes that represent bad data. If outliers are not caused by any error or variability in the measurements, then it's fair to analyze the outliers as they may represent some relevant aspect.

8) Explain what should be done to deal with the two types of measurement uncertainties: random and systematic uncertainties.

Random uncertainties: treated statistically, using averages, variance and calculating confidence intervals.

Systematic uncertainties: should be understood and removed before analysis.

9) Suppose you are the data center administrator of a big organization and you are about to decide if your organization should accept the new cooling system that was recently installed in the data center. As the cooling system is very expensive, the contract defines precisely the conditions that should be met, concerning the temperature inside the server racks. Quoting the contract: "The cooling system assures that the temperature in the racks is always in the range of 16.00 ± 0.80 Celsius degrees, with a confidence of 95%". In order to be sure that the cooling system is operating under the values defined in the contract, you decided to measure the temperature in the racks using a high precision electronic thermometer. To assure representative measurements, you took 100 measurements, including a variety of server loading scenarios and covering the 24 hours of the day. The results obtained show an average temperature of 16.245 Celsius with a standard deviation of

2.234. Do you think the conditions defined by the contract are met and your organization should accept the cooling system as is?

Considering that the measurements are independent (which is the case, as one measurement is not dependent on previous measurements) and that the measurements follow an approximately normal distribution, the formula to obtain the confidence interval is:

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

where $\bar{x}$ is the mean (arithmetic average) of the samples, z is confidence coefficient obtained form the Z table, s is the standard deviation of the samples and n is the number of samples. That is:

n = 100
z = 1.96 for 95% confidence interval (see the normal table)
s = 2.234
$\bar{x}$ = 16.245

Doing the calculation, the confidence interval is:

$$z * s / \sqrt{n} = 1.96 * 2.234/\sqrt{100} = 0.44$$

The confidence interval for the measured temperature, with 95% of confidence, is

Minimum value: 16.245 – 0.44 = 15.81
Maximum value: 16.245 + 0.44 = 16.68

Based on these results, the conclusion is that the cooling system is operating under the specifications defined in the contract.

10) Consider the scenario described in the previous question but now suppose that you could only take 15 measurements. Explain what is different in this case. Consider both the arguments of the data center administrator (your perspective) and the vending representative of the cooling system.

The calculation is the same as for the previous exercise with the differences that confidence coefficient is now obtained from a T table (T Student distribution) and we consider the df (degree of freedom) of n-1 instead of n samples. The t value for df = 14 is 2.1448 (in a T table for upper tail probability, you will have to look at $\alpha = 0.05/2 = 0.025$, as the confidence interval consider two tails.
Doing the calculation, the confidence interval is:

$$t * s / \sqrt{n} = 2.1448 * 2.234/\sqrt{15} = 1.24$$

The confidence interval for the measured temperature, with 95% of confidence and only 15 measurements, is

Minimum value: 16.245 – 1.24 = 15.01
Maximum value: 16.245 + 1.28 = 17.48

Based on these results, the conclusion is that the cooling system is NOT operating under the specifications defined in the contract, with 95% of confidence. Obviously, the problem here is that due to the small number of measurements, the level of confidence is smaller. The vendor could argue that you would need at least 30 measurements to take a solid decision.

11) Consider the following problem statement: the number of software bugs found in the tests of program units developed by programmers is dependent on the average number of sleeping hours of the programmers. Assume that you have the detailed specifications of a set of program units to be developed, and consider that the program units include units of

high, medium and low complexity. Additionally, you have comprehensive unit test suits to test each program unit.

In these circumstances, describe how you would organize an experiment to answer the proposed problem statement. Your answer should be as complete as possible, focusing on the experiment design steps (obviously, it does not make sense to speculate about the experiment results and conclusions), and indicate the dependent and independent variables, the levels you would consider for the independent variables, the hypothesis under evaluation and the hypothesis testing technique you would use. Also describe, very briefly, the experimental setup and take into account in your answer to the whole question that the experiment deals with people (the programmers).

The answer should follow the main steps involved in the design of an experiment. As the problem statement is given, we start by identifying the variables.

Variables:
Dependent variable:
- Number of bugs detected by the test suite (this is what we want to observe).

Independent variables:
- Average number of sleeping hours
- Complexity of the program units

Levels:
After defining the variables, an important step is to select the levels for the independent variables, as these levels represent the set points of the experiments. The levels could be:
- Average number of sleeping hours (2, 4, 6, 8, 10 sleeping hours per day)
- Complexity of the program units (small, medium, and high complexity[1])

The two independent variables mentioned above are the most obvious ones (and would be enough for a correct answer). However, there are other independent variables that could be referred. For example:
- Expertise of the programmer (novice, qualified, expert)
- Programming language (Java, C++, Python)

Hypothesis:
As the hypothesis is a tentative answer to the problem, there are several alternatives. In any case, one of the most obvious hypotheses is:
H0 – The number of bugs done by programmers does not depend on the average sleeping hours of the programmers.
H1 – The number of bugs done by programmers increase when the average sleeping hours of the programmers increases.

Note that the definition of the hypothesis includes both the null hypotheses (H0) and the alternate hypothesis (H1).

Considering that the number of measurements/samples in experiments with people is normally limited, the appropriate hypothesis testing technique would be two-sample T test, applied successively to the measurements obtained with the different levels of the independent variables, or ANOVA (one-way or even two-ways ANOVA, depending on the number of independent variables and levels used in the experiment). Do not forget that with ANOVA you can only determine whether the number of sleeping has effect on the number of bugs or

[1] The complexity of program units can be measured using classic software complexity metrics such as McCabe's cyclomatic complexity or Halsteads metrics (if you are not familiar with programming complexity metrics have a look at: http://en.wikipedia.org/wiki/Programming_complexity).

not (i.e., you cannot test the case in which you assume that more sleeping hours will lead to less bugs). Another aspect that is worth mentioning is that the two samples can be treated as independent samples. However, the experiment would be much better if you could use a two-dependent sample setup. In this case, the measurement would be taken considering the same programmer and different sleeping times, and the hypothesis testing technique would be the two-dependent sample T test.

Experimental setup/scenario

Very briefly, as requested… Considering the simple scenario with only two independent variables, the experiment setup consists of a group of programmers that develop a set of program units of different complexity levels. Each programmer develops all the program units involved in the experiment (including low, medium and high complexity program units), covering all the possibilities for the levels defined for the sleeping hours (at least one program unit of each complexity level should be developed in each level of sleeping hours). In order to full control the experiment, the sleeping hours of the programmers need to be controlled using a simple alarm clock, in order to achieve the levels defined for the variable "average number of sleeping hours". When a programmer declares that a given program unit is finished and ready for testing, the program is tested using the test suit (mentioned in the question). The number of bugs detected (and corrected until the test suit tells that the program unit is free of bugs) is recorded. After performing all the individual experiments with each programmer and gather the results, the analysis will use the selected hypothesis testing technique (T test or ANOVA) to draw a conclusion (retain or reject the hypothesis).

13) In hypothesis testing, there are two types of errors: type II errors (false negatives) and type I errors (false positive). Explain the differences between these two errors and compare the consequences of each one.

The type I error (false positive) is when you reject the null hypothesis but the null hypothesis is true and type II error (false negative) is when you retain the null hypothesis but the null hypothesis is false.

Type II errors are equivalent to doing nothing. That is, instead of rejecting the null hypothesis (the correct decision) you retain it. For example, if you are optimizing the configuration of a server to improve response time, it means that you keep the current configuration, although the alternative configuration is better. In any case, you can do additional tests and improve the server configuration later on.

The consequences of type I error are much more severe, as rejecting a true null hypothesis means that you change something in the system under evaluation but that change is bad. In the example of tuning up the response time of a server, in a type I error you reject the current configuration (null hypothesis) and accept a new configuration, but this one is worse than the previous configuration.

14) Code inspections (a specific form of software inspections proposed by Michael Fagan more than four decades ago) is a technique to find bugs in the software under development in order to produce high quality software. Very briefly, in a code inspection a group of inspectors (experienced programmers) examine the code of a given code module (in general, short blocs of code with less than 100 lines of code) in order to find possible bugs. Obviously, the inspectors are not the authors of the code under inspection.

The goal is to perform an experiment to evaluate the effectiveness of code inspection in finding bugs. The idea is to use a set of code modules that have been very well tested and are considered totally correct (i.e., with no bugs) and insert a small number of bugs (3 or 4 for bugs per module). The inspectors will perform the code inspection on these modules and the goal is to check if they can detect the bugs that have been inserted in the modules.

In each individual code inspection, the inspector indicates the number of bugs found in the module. As the person that is managing the experiment knows exactly the bugs that have been inserted in each module, it is possible to confirm the quality of the code inspection done by each inspector. In practice, the result of each individual inspection is the number of bugs found and the number of mistakes done by the inspector. The mistakes are of two types: bugs that are wrongly identified (i.e., the code is correct but the inspector thinks there is a bug; this is a false positive) and real bugs that are not identified by the inspector (i.e., he looked at the code but did not find anything wrong; this is a false negative). These are the results that will be used to evaluate the effectiveness of the code inspections.

Assume that the modules (blocks of code) available for the experiment are written in C, C++ and Java, and that the authors of the modules have inserted the bugs in the code, to assure that the bugs are realistic (i.e., not obvious bugs, very easy to find). Additionally, consider that the group of inspectors available for the experiment fill a questioner with some demographic data (age, sex, etc.) and state the number of years of experience as programmer/code inspector.

In these circumstances, and considering the problem statement of knowing if code inspections are effective in finding bugs, indicate the following elements of the experiment:

a) Hypothesis

There are several possibilities for the definition of hypothesis. The most obvious would be to set a given assumed goal for the performance of the code inspectors and test if the hypothesis can be rejected. For example:

H0 – The code inspectors can detect in average up to 70% of the bugs existing in the inspected code. That is, $\mu \leq 70\%$.

H1 – The code inspectors can detect more than to 70% of the bugs existing in the inspected code. That is, $\mu > 70\%$.

But many other hypotheses could be used in this experiment. For example:

H0 – The probability of bug detection is the same, no matter the programing language in which the inspected module is written (C, C++ or Java).

H1 – The probability of bug detection is depended on the programing language in which the inspected module is written.

b) Variables (dependent and independent) and levels used for the independent variables, justifying your choices

Dependent variables:
- Number of bugs detected
- Number of false positives
- Number of false negatives

The choice of the dependent variable or variables depends on the needs and goals of the experiment. You could just focus in one of these dependent variables or you could study all of them.

Independent variables:
- Programming language → levels: C, C++, Java
- Experience of the inspectors → novice, knowledgeable, expert

Many other possible independent variables could be considered. For examples, variables related to demographic aspects of the inspectors (age or sex), variables related to the duration

of the inspection (to assess the impact of lack of concentration after some time), variables related to the metrics of the modules (lines of code, complexity, etc.).

Training exercises 2

1) The following summary table shows the results of the execution time of a package of benchmark programs compiled with four different compilers, named as A, B, C and D for experiment purposes. Show whether the compilers used have any effect on the execution time of the compiled programs or not considering 95% of confidence.

Compiler	Number of runs	Exec. time mean	Exec. time standard deviation
A	8	4.35	2.25
B	11	3.79	1.63
C	10	4.91	1.89
D	9	6.33	2.21

1 - State the hypothesis:

H_0 – The compiler **is not relevant** for the execution time.

H_1 – The compiler **is relevant** for the execution time.

2 - Compute the test statistic using ANOVA:

Using the formulas for the sum of squares (see slides used in the lectures) and the elements of the one-way ANOVA table, and making the calculations, the ANOVA summary table is:

Formulas:

$$GM = \frac{\sum n_i \bar{x}_i}{\sum n_i} \qquad SS_{(W)} = \sum df\, s^2 \qquad SS_{(B)} = \sum n_A (\bar{X}_A - \bar{X}_G)^2$$

Table after calculations:

SOURCE	SS	df	MS	F
Between	34.030	3	11.343	2.895
Within	133.228	34	3.918	
TOTAL	167.258	37		

The critical value for α = 5% and $F_{3,34}$ (df_1 = 3 for the numerator and df_2 = 34 for the denominator) can be obtained from a F table. Looking at $F_{3,34}$ in a F table for α = 5%, we see that for df_1 = 3 the table does not show a line with df_2 = 34. But we can observe that the table shows df_2 = 30 and df_2 = 40 with the following values of F:

- $F_{3,30}$ → Fc = 2.9223
- $F_{3,40}$ → Fc = 2.8387

The critical value for α = 5% and $F_{3,34}$ is between these two values. Unfortunately, the value of F = 2.895 we obtained in a) is also inside this interval, so the only conclusion we can draw is that the value of F = 2.895 is very close to the critical value. We would need a more detailed F table to find the accurate critical value for $F_{3,34}$.

3 - Calculate the p value:

As you know, if the value of F obtained from the ANOVA table is greater than the critical value, it means that the probability p is lower than alpha (α = 5% in this exercise) and we should reject the null hypothesis. With the results obtained from the ANOVA table (and without access to a calculator that gives the accurate value of p for $F_{3,34}$) we would need to use the F table for α = 5% and the F table for α = 2.5% to obtain an approximate p value. It is worth noting that most of the F tables (including the ones available at Inforestudante) are organized to find the critical value in a F distribution and not to find the p value for a

given value of F. In any case, there are tables that provide for each $F_{df1, df2}$ the values for different probabilities, which simplifies finding the value of p for a given $F_{df1, df2}$.

4 - Make a decision

In a written exam, I would expect that students could provide a clear characterization of the situation for an exercise like this, showing the problem of lack of detail in the F table to provide conclusions in this borderline case. The answer would be considered as 100% correct if it shows that the student fully understands the problem and the solution.

In order to provide a complete answer, using a more detailed table or an online calculator (e.g., https://www.easycalculation.com/statistics/f-test-p-value.php), the value of p for $F_{3,34}$ is 4.93%. This means that we should reject the null hypothesis. As we can see, the value is very close to 5%.

2) A Bank has developed a new database system with the goal of improving the speed of typical web banking transactions done by the Bank customers. There are several types of transactions (e.g., viewing account balances, funds transfers between the customer's accounts, viewing transactions in a given period indicated by the customer, etc.) The response time of each transaction depends on several factors related to both the queries issued by the customer and the load of the system. In order to assess the new database response, the Bank asked its Information Systems Department to perform an experiment to test the speed of the new database and compare the speed with the current database. The experiment includes a set of representative accounts with typical transactions profiles to assure that the measurements of the speed in both databases (the new one and the already existing) are meaningful.

a) State the different elements of the experiment (problem definition, variables, levels, hypothesis,…) and justify your choices.

Problem
Is the new database faster than the old one for a balanced mix of transactions?

Dependent variable
Number of transactions per minute.

Independent variables

- Database system → Levels: old systems; new system
- Transaction profiles → Levels: Profile 1 (e.g., viewing intensive, Profile 2 (e.g., transfer intensive), etc.
- Other independent variable not relevant for the study, given the way the problem is described: Server (machine), Operating System, DBMS tuning, etc.

Hypothesis
H_0 – The speed of the new database system is not higher than the old one.
H_1 – The new database system is faster.

b) In very few cases it was observed the response time of the transactions was several times larger than the average response time. Explain the choices you have to deal with this problem.

1 – Try to understand the source of these unexpected measurements and decide whether they should be considered as part of the nominal response, or should be considered an outlier and ignored in the data analysis (but should be reported anyway).

2 – If they are an outlier (and ignored in the data analysis), it is important to try to identify if the outliers may represent something rare but relevant for the system.

3) A company has developed a new word-searching algorithm in files in smartphones and wants to test the speed of the new algorithm in three operating systems (Android, iOS, and Ubuntu Touch) and in two smartphones (named A and B for the experiment purposes). After a set of measurements, the data collected originated the following ANOVA table.

	SS	df	MS	F	P
Factor OS	23.333	2	11.6667	17.5	1.384e-06
Factor Smartphone	1.667	1	1.667	2.5	0.1197
Interaction	23.333	2	11.6667	17.5	1.384e-06
Within (error)	36.000	54	0.6667		

Based on this table, what conclusions can you take considering a significance level of 5%.

First, we need to make it clear the hypothesis under test. It must be something like this:

- **H_0**
  H_{0a} – The OS has no impact on the speed of the searching algorithm
  H_{0b} – The smartphone has no impact on the speed of the searching algorithm
  H_{0c} – There is no interaction between OSs and smartphone with impact on the speed of the searching algorithm
- **H_1** – Either the OS or the smartphone have impact on the speed of the searching algorithm or there is interaction between OSs and smartphone

Based on the ANOVA table, we can conclude the following:

- The influence of the OS is highly significant (p is very small) → reject **H_0**
- The influence of the smartphone is not significant (p is 11.97%)
- The interaction is highly significant (p is very small) → reject **H_0**

→ Globally, we should reject **H_0**

4) In the design of a large-scale experiment involving virtualized infrastructures such as the ones that support the Cloud you have identified a very large number of variables. You are aware of the fact that a large number of variables complicates the experiments and increases the whole experimental cost. Because of that, you are reanalysing the elements of your experiment to simplify (and to focus) the experiment as much as possible. In particular, should the following two variables (A and B) be considered in the experiment (as independent variables) or can you use just one of them? Justify your answer.

Variable A	Variable B
2.34	3.36
3.98	2.23
7.89	-2.98
12.13	-7.56
13.45	-7.98

The solution is to verify if the two variables are correlated. If they have a strong correlation then we can use only one of them as independent variable, as the conclusions we can extract from the experiments with one variable will be similar to the conclusion that can be drown with the other.

In order to verify if the two variables are correlated we can use the Pearson's test and calculate the Pearson's coefficient *r*.

The formula for the Pearson's coefficient is:

$$r = \frac{n\sum XY - (\sum X)(\sum Y)}{\sqrt{[n\sum X^2 - (\sum X)^2]\,[n\sum Y^2 - (\sum Y)^2]}}$$

In order to facilitate the calculation, we build the following table with the intermediate results:

	A (X)	B (Y)	XY	X^2	Y^2
	2.34	3.36	7.86	5.48	11.28
	3.98	2.23	8.88	15.84	4.97
	7.89	-2.98	-23.51	62.25	8.88
	12.13	-7.56	-91.70	147.14	57.15
	13.45	-7.98	-107.33	180.90	63.68
Sum	**39.79**	**-12.93**	**-205.81**	**411.61**	**145.96**

Doing the calculations, the result is r = - 0.995. The conclusion is that there is a strong negative correlation between the two variables.

However, in order to exclude one of the variables, we must test the hypothesis that A and B are correlated with a given significance level, for example 95%. The test statistic in this case is:

$$t = \frac{r}{\sqrt{\frac{1-(r)^2}{n-2}}}$$

1 – Hypothesis

H_0 – A and B are not correlated with a high level of significance (e.g., 95%).
H_1 – A and B are correlated

2 – Compute the test statistic

t = - 0.995/SQR((1- (- 0.995)^2)/(5-2)) = - 0.995/0,057662813 = -17.25548839

3 – Obtain p value

Obtain the p value for t = -17.255 for df = 5-2 = 3
→ we need the probability for t_3= -17.255

Searching the T table, we can see that p = 0.05% for t_3 = 12.92. This means that the probability for t_3= -17.255 is even lower.

4 – Make a decision

Since p is very low (lower than 0.05%), we can reject H_0
We can use only one variable, either A or B