# MEIC - Engenharia de Software – 2011/12
## Instructions for Program 2

**Read completely and clarify any question with your instructor before starting the work!**

## 1. Goals

Besides the goals of the first project (follow a process and gather data to understand and improve performance), the second project has also the goal of applying estimation techniques.

## 2. Requirements

Using the software development process SDP2 described below, write a program in Java to perform the following linear regression calculations:

- calculate the linear regression parameters $\beta_0$ and $\beta_1$ (coefficients of the linear regression line) and correlation coefficient $r$ for a set of $n$ pairs of data $(x_1, y_1)$, …, $(x_n, y_n)$;
- calculate the significance of this correlation;
- given a value $x_k$, calculate a corresponding estimate $y_k$ by linear regression: $y_k = \beta_0 + \beta_1 x_k$;
- calculate the 70% prediction interval (LPI - lower prediction interval and UPI – upper prediction interval) for that estimate.

Linear regression is used in one of the variants of the PROBE method. Table 1 contains historical estimated and actual data for 10 programs. For program 11, the developer has estimated a size of 386 added and modified LOC.

| Program Number | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Estimated Added and Modified Size (LOC) | 130 | 650 | 99 | 150 | 128 | 302 | 95 | 945 | 368 | 961 |
| Actual Added and Modified Size (LOC) | 186 | 699 | 132 | 272 | 291 | 331 | 199 | 1890 | 788 | 1601 |
| Actual Development Time (Hours) | 15.0 | 69.9 | 6.5 | 22.4 | 28.4 | 65.9 | 19.4 | 198.7 | 38.8 | 138.2 |

**Table 1**

Thoroughly test the program. At a minimum, run the following tests.

- Test 1: Perform the required calculations defined above using the estimated and actual added and modified size in Table 1. Use an estimated size of $x_k$ = 386 in producing the improved size estimate and prediction interval.
- Test 2: Perform the required calculations defined above using estimated added and modified size and actual development time in Table 1. Use an estimated size of $x_k$ = 386 in producing the time estimate and prediction interval.

Expected results are shown in Table 2.

| | $r$ | *tail area* | $\beta_0$ | $\beta_1$ | $y_k$ | UPI (70%) | LPI (70%) |
|---|---|---|---|---|---|---|---|
| Test 1 | 0.9545 | 1.775E-05 | -22.55 | 1.728 | 644.4 | 874.4 | 414.4 |
| Test 2 | 0.9333 | 7.982E-05 | -4.039 | 0.1681 | 60.86 | 88.42 | 33.30 |

**Table 2**

You must have different classes for: input/output; regression calculations; Student's t-distribution (required in the regression calculations). The relevant numerical methods and test cases for implementing the Student's *t*-distribution are already provided in the assignment package.

# 3. Mathematical background

## 3.1. Usage of linear regression

Linear regression is a way of optimally fitting a line to a set of data, as illustrated in Figure 1.

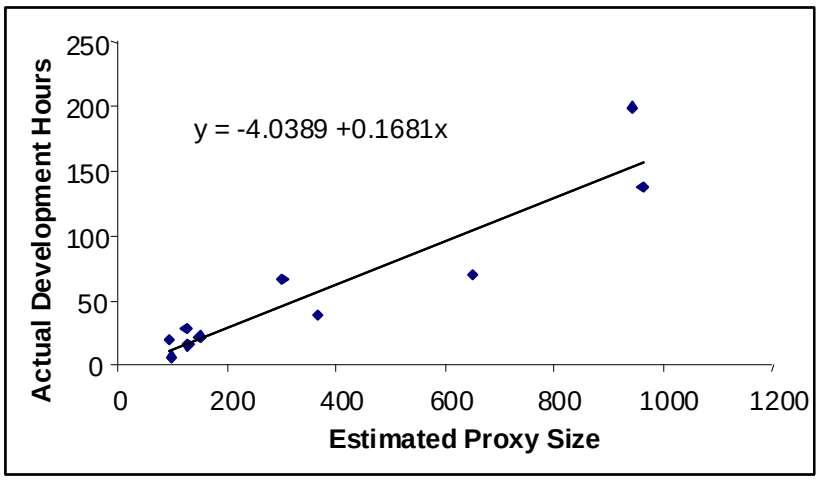



**Figure 1 (Data from test 2)**

The correlation coefficient *r* determines the degree to which two sets of numerical data are linearly correlated. The correlation coefficient *r* can range from -1 to +1.

- Results near +1 imply a strong positive relationship; when *x* increases, so does *y*.
- Results near -1 imply a strong negative relationship; when *x* increases, *y* decreases.
- Results near 0 imply no relationship.

Correlation is used to judge the quality of the linear relation in various historical process data that are used for planning. For this purpose, we examine the value of $r^2$.

| If $r^2$ is | the relationship is |
|---|---|
| $.9 \le r^2$ | predictive; use it with high confidence |
| $.7 \le r^2 < .9$ | strong and can be used for planning |
| $.5 \le r^2 < .7$ | adequate for planning but use with caution |
| $r^2 < .5$ | not reliable for planning purposes |

## 3.2. Regression and correlation calculations

The regression parameters $\beta_0$ and $\beta_1$ and the correlation coefficient *r* are calculated as follows:

$$\beta_1 = \frac{\left(\sum_{i=1}^{n} x_i y_i\right) - \left(n x_{avg} y_{avg}\right)}{\left(\sum_{i=1}^{n} x_i^2\right) - \left(n x_{avg}^2\right)} \qquad \beta_0 = y_{avg} - \beta_1 x_{avg}$$

$$r = \frac{n\left(\sum_{i=1}^{n} x_i y_i\right) - \left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} y_i\right)}{\sqrt{\left[n\left(\sum_{i=1}^{n} x_i^2\right) - \left(\sum_{i=1}^{n} x_i\right)^2\right]\left[n\left(\sum_{i=1}^{n} y_i^2\right) - \left(\sum_{i=1}^{n} y_i\right)^2\right]}}$$

where

- $x_{avg}$ is the average of the *x* values
- $y_{avg}$ is the average of the *y* values

### 3.3. Significance

The significance test determines the likelihood that a strong correlation is random (occurs by chance), and is therefore of no practical significance. For example, a data set with only two points will always have an $r^2 = 1$, but this correlation is not statistically significant.

The procedure for calculating the correlation significance is as follows.

1. Compute the value of $x$ given by
$$x = \frac{|r|\sqrt{n-2}}{\sqrt{1-r^2}}$$
where
   - $r$ is the correlation
   - $n$ is the number of data points
2. Calculate the probability $p = F_{n-2}(x)$, where $F_d(x)$ is the cumulative distribution function of the Student's $t$-distribution with $d$ degrees of freedom, that is, the integral of the probability density function $f(x)$ between $-\infty$ and $x$ (see section 3.5).
3. Calculate the tail area as *1-p*. (The area under the curve of $f(x)$ between x and $+\infty$).

A tail area ≤ 0.05 is considered as strong evidence that there is a relationship.
A tail area ≥ 0.2 indicates a relationship that is due to chance.

### 3.4. Prediction interval

The prediction interval provides a likely range around the estimate.

- A 70% prediction interval gives the range within which 70% of the estimates will fall.
- It is not a forecast, only an expectation.
- It only applies if the estimate behaves like the historical data.

It is calculated from the same data used to calculate the regression parameters.

To calculate the prediction interval, use the following steps.

1. Calculate the *Range* for a 70% interval.
2. Calculate the Upper Prediction Interval (UPI) as $y_k$ + *Range*.
3. Calculate the Lower Prediction Interval (LPI) as $y_k$ – *Range*.

The formula for calculating the prediction range for a 70% interval is

$$Range = t(n-2, 0.5+\frac{0.70}{2})\,\sigma\sqrt{1+\frac{1}{n}+\frac{(x_k - x_{avg})^2}{\sum_{i=1}^{n}(x_i - x_{avg})^2}}$$

where

- $x_i$ is your historical data
- $n$ is the number of historical data points
- $t(d,p)$ is the value of $x$ such that $F_d(x) = p$ (i.e., such that the probability of the Student's $t$-distribution with $d$ degrees of freedom between $-\infty$ and $x$ is $p$).

The formula for calculating the standard deviation term is

$$\sigma = \sqrt{\left(\frac{1}{n-2}\right)\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2}$$

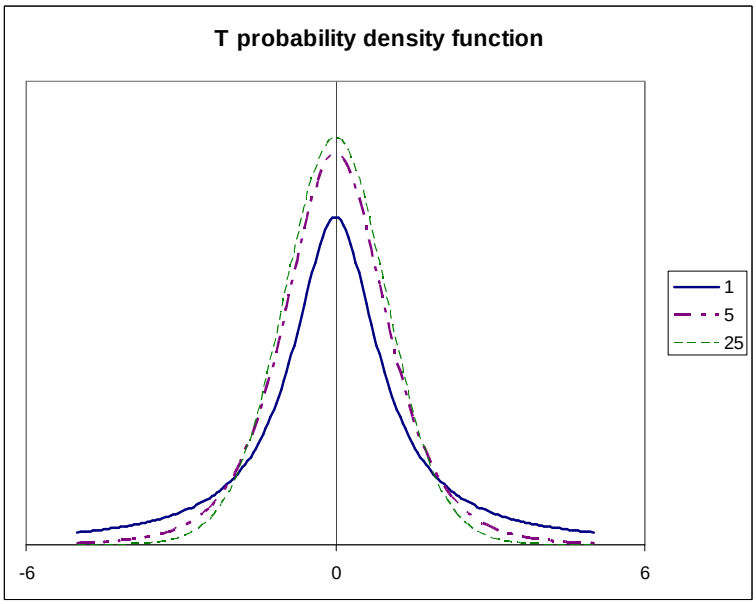
where

- $x_i$, $y_i$ is your historical data
- $n$ is the number of historical data points

### 3.5. The Student's *t*-distribution

The t distribution is a very important statistical tool. It is used instead of the normal distribution when the true value of the population variance is not known and must be estimated from a sample. Assume X is a random variable normally distributed with known mean μ and unknown variance $\sigma^2$. Let $\overline{X}$ and *S* be the mean and standard deviation of a sample $X_1, \ldots X_n$. Then, the "t" statistics $t = \frac{\overline{X} - \mu}{S/\sqrt{n}}$ has a *t*-distribution with *d=n-1* degrees of freedom.

The shape of the t distribution is dependent on the number of points in the sample. As *n* gets large, the t distribution approaches the normal distribution. For lower values, it has a lower central "hump" and fatter "tails."




In software estimation, the t distribution can be used in two ways: to test the significance of a linear correlation (the probability of not happening by chance); and to calculate a prediction interval around an estimate calculated by linear regression.

The *t*-distribution has the following probability density function:

$$f(x) = \frac{\Gamma\left(\frac{d+1}{2}\right)}{(d*\pi)^{1/2}\Gamma\left(\frac{d}{2}\right)}\left(1+\frac{x^2}{d}\right)^{-(d+1)/2}$$

where

- *d* = degrees of freedom
- Γ is the gamma function

By numerically integrating this function (e.g., with the Simpson method), one can compute the cumulative distribution function $F(x) \triangleq P(X \leq x) = \int_{-\infty}^{x} f(x)dx$, where *X* represents a random variable with Student's *t*-distribution. Reciprocally, notice that *F'(x) = f(x).*

To determine the value *x* such that *F(x)=p (i.e., P(X≤x)=p),* for a given *p*, one can use the Newton-Rapson method to find the root of *g(x)=F(x)-p*.

# 4. SDP2 Process scripts

Before starting program 2, review the top-level process script below. Also, ensure that you have all of the required inputs. Changes to SDP1 are highlighted in ***bold italic***.

## 4.1. SDP2 Process Script

| **Purpose** | To guide the development of module-level programs |
|---|---|
| **Entry Criteria** | □ Problem description [1]<br>□ Project Plan Summary form [2]<br>□ Time and Defect Recording logs 2<br>□ Defect Type and Size Counting Standard 1<br>□ ***Size Estimating template*** 2<br>□ ***Historical size and time data*** [3]<br>□ ***Size counting tool*** [4] |

| **Step** | **Activities** | **Description** |
|---|---|---|
| 1 | Planning (see detailed script) | - Produce or obtain a requirements statement.<br>- ***Produce a program conceptual design.***<br>- ***Complete the Size Estimating template.***<br>- ***Use historical productivity to estimate the required development time.***<br>- Complete the Time Recording log. |
| 2 | Development (see detailed script) | - Design the program.<br>- Implement the design.<br>- Test the program, and fix and log all defects found.<br>- Complete the Time Recording log. |
| 3 | Postmortem (see detailed script) | - Complete the Project Plan Summary form with actual time, defect, and size data. |

| **Exit Criteria** | □ A thoroughly tested program<br>□ Completed Project Plan Summary form with estimated and actual data<br>□ ***Completed Size Estimating template***<br>□ Completed Time and Defect Recording logs |
|---|---|

## 4.2. SDP2 Planning Script

| **Purpose** | To guide the SDP2 planning process |
|---|---|
| **Entry Criteria** | □ Problem description<br>□ Project Plan Summary form<br>□ Time Recording log |

| **Step** | **Activities** | **Description** |
|---|---|---|
| 1 | Program Requirements | □ Produce or obtain a requirements statement for the program.<br>□ Ensure that the requirements statement is clear and unambiguous.<br>□ Resolve any questions. |
| 2 | Size Estimate | □ ***Produce a program conceptual design.***<br>□ ***Complete the Size Estimating template.***<br>□ ***The total estimated added and modified size in the Project Plan Summary form is calculated automatically.*** |
| 3 | Time Estimate | □ ***Enter your estimated productivity in the Project Plan Summary form, based on your historical productivity from previous project(s).***<br>□ ***The estimated development time in the Project Plan Summary form is calculated automatically.*** |

| **Exit Criteria** | □ Documented requirements statement<br>□ Completed Project Plan Summary form with estimated size and time data<br>□ ***Completed Size Estimating template***<br>□ Completed Time Recording log |
|---|---|

Verify that you have met all of the exit criteria for the planning phase, **then have an instructor review your plan**. After your plan has been reviewed, proceed to the development phase.

[1] In this document.
[2] In the accompanying Excel workbook.
[3] In this case, use the actual size and time from program 1 to obtain the historical productivity in LOC/hour.
[4] Developed in program 1.

### 4.3. SDP2 Development Script *(same as in program 1)*

### 4.4. SDP1 Postmortem Script

| | |
|---|---|
| **Purpose** | To guide the SDP2 postmortem process |
| **Entry Criteria** | □ Problem description and requirements statement<br>□ Project Plan Summary form with development time data<br>□ Completed Time and Defect Recording logs<br>□ A tested and running program. |

| Step | Activities | Description |
|---|---|---|
| 1 | Defects | □ Review the Defect Recording log to verify that all of the defects found in each phase were recorded.<br>□ Using your best recollection, record any omitted defects.<br>□ Check that the data on every defect in the Defect Recording log are accurate and complete.<br>□ Verify that the numbers of defects injected and removed per phase are reasonable and correct.<br>□ Using your best recollection, correct any missing or incorrect defect data. |
| 2 | Size | □ ***Use your LOCCounter program to determine the added and modified size of each source file in your program, in LOC (except test code).***<br>□ ***Enter the actual sizes in the Size Estimating template.***<br>□ ***The actual size in the Project Plan Summary form is calculated automatically.*** |
| 3 | Time | □ Review the completed Time Recording log for errors or omissions.<br>□ Using your best recollection, correct any missing or incomplete time data. |

| | |
|---|---|
| **Exit Criteria** | □ A thoroughly tested program<br>□ Completed Project Plan Summary form<br>□ Completed Time and Defect Recording logs |

# 5. SDP2 Process forms and standards

### 5.1. Project Plan Summary

Student(s) _____ Class _____
Program _____ Language _____
Instructor _____ Start Date _____
End Date _____

| Parameter | Estimated Value | Actual Value |
|---|---|---|
| ***Added and Modified Size (LOC)*** | | |
| Time (minutes) | | |
| ***Productivity (LOC/Hour)*** | | |

| Phase | Time in Phase | Defects Injected | Defects Removed |
|---|---|---|---|
| PLAN - Planning | | | |
| DLD - Detailed Design | | | |
| CODE - Code | | | |
| UT - Unit Test | | | |
| PM - Postmortem | | | |
| **Total** | | | |

**Instructions *(for changed/new elements)***

| | |
|---|---|
| **Estimated *Added and Modified* Size** | - Shows the estimated ***added and modified*** size of your program, in lines of code, excluding test code (see Size Counting Standard).<br>- ***It is calculated automatically based on the estimates in the Size Estimating template.*** |
| **Estimated Time** | - ***Shows the estimated total development time, in minutes.***<br>- ***It is calculated automatically based on the estimated size and productivity.*** |
| **Actual Size** | - ***Shows the actual added and modified program size.***<br>- ***It is calculated automatically based on the actual sizes in the Size Estimating template.*** |

| | |
|---|---|
| **Actual Time** | - Actual development time; calculated automatically based on the Time Recording Log. |
| ***Estimated Productivity*** | - ***Enter your estimated productivity for this project, in LOC/hour, based on your historical productivity.*** |
| ***Actual Productivity*** | - ***Shows the actual productivity for this project.*** |

## 5.2. Time Recording Log *(same as in program 1)*

## 5.3. Defect Recording Log *(same as in program 1)*

## 5.4. Size Estimating Template

| New Parts | | | | | |
|---|---|---|---|---|---|
| **Part name (class or part of class)** | **Part type** | **Num. Items (methods)** | **Items' relative size** | **Estimated part size (LOC)** | **Actual size (LOC)** |
| | | | | | |

| Base/Reused Parts | | | | |
|---|---|---|---|---|
| **Part name (class / file)** | **Initial size (LOC)** | **Estimated added and modified size (LOC)** | **Actual added and modified size (LOC)** | **Final size (LOC)** |
| | | | | |

### Instructions

| | |
|---|---|
| **Purpose** | Use this form to make size estimates. |
| **General** | - A part could be a class, module, component, product, or system.<br>- Where parts have a substructure of methods, procedures, functions, or similar elements, these lowest-level elements are called items. |
| **New Parts** | If you plan to add newly developed parts<br>- enter the part name, type, number of items (or methods), and relative size<br>- for each part, get the size per item from the appropriate relative size table, multiply this value by the number of items, and enter in estimated size (this is performed automatically)<br>- in the postmortem phase, measure and enter the actual size of each newly developed part |
| **Base/Reused Parts** | If this is a modification or enhancement of an existing product or you plan to include reused parts<br>- measure and enter the initial size of each part you want to reuse or modify<br>- estimate and enter the modified and added size to the base code<br>- in the postmortem phase, measure and enter the actual added and modified size and the final size of each part |

## 5.5. Defect Type Standard *(same as in program 1)*

## 5.6. Size Counting Standard *(same as in program 1)*

# 6. General instructions

## 6.1. Assignment package

The assignment package (ESOF-Prog2.zip) contains the following files:
- ESOF-Prog2.pdf - this document, with the assignment instructions;
- ESOF-Prog2.xls - Excel workbook for recording data about your work;
- Function.java, TestTStudentDistribution.java, GammaFunction.java, SimpsonMethod.java, NewtonRapsonMethod.java –source code to reuse (unchanged) in the implementation of the Student's t-distribution calculations.

## 6.2. Work alone or in pairs

You can work alone or in pairs. In case you work as a pair, the pair should be the same for the 4 assignments, and you must do the complete work together, on the same tasks (you cannot split tasks); you should also record the time as of a single person. [5]

## 6.3. Follow the process

For each process script,
- ensure that you have all of the required inputs, according to the entry conditions;
- start a new Time Recording Log entry, with the start date and time;
- execute the process steps;
- check that the exit criteria are met;
- record the stop date and time in the Time Recording Log.

In the case of the Development script, create separate time records for Design, Code and Test.

## 6.4. Review your assignment

Use the attached grading checklist to check that your assignment is correct before you submit it.

By completing this assignment you gain **0,5 values out of 20**. ***In this assignment, you get 100% only if you meet all the grading checklist items in the first submission.***

## 6.5. Submit your assignment

When you've completed your review, submit your assignment in a zip file, through an email to the instructor with "**ESOF-Prog2**" in the subject, with the following information:
- workbook with your process data
- source program listing
- test results (***JUnit code with expected values, or printout of program execution results).***

In the email identify the elements of the group by their code (ei0XXX).
You should complete and submit this assignment at most **one week after the class**.

## 6.6. Suggestions

Keep your programs simple. You will learn as much from developing small programs as from large ones. If you are not sure about something, ask your instructor for clarification.

***Software is not a solo business, so you do not have to work alone. You must, however, produce your own estimates, designs, code, and completed forms and reports.***

[5] Discuss with your instructor any exceptions.

# 7. Grading checklist

| Legend |
|---|
| √ - O.K. |
| X – Not Ok |

| | **Assignment Package** | **Comments** |
|---|---|---|
| | Excel workbook | |
| | Source program listing | |
| | Test results | |

| | **Program and Test Results** | **Comments** |
|---|---|---|
| | The program appears to be workable. | |
| | ***The program is appropriately structured.*** | |
| | All required tests have been run. | |
| | The actual output is correct for each test. | |

| | **Time Log** | **Comments** |
|---|---|---|
| | Time data are entered for all process phases. | |
| | Process phases are sequenced appropriately. | |
| | Time data are entered against the appropriate process phase. | |
| | Time data are complete and reasonable. | |
| | Times were recorded as the work was done. | |

| | **Defect Log** | **Comments** |
|---|---|---|
| | Every defect has all the required data. | |
| | Defects were injected before removed. | |
| | Defects are adequately described. | |
| | Defect types are consistent with description. | |
| | Defect types are assigned consistently. | |

| | **Size Estimating Template** | **Comments** |
|---|---|---|
| | ***The estimated size data are complete and reasonable.*** | |
| | ***A suitable number of new parts are identified.*** | |
| | ***A suitable number of base/reused parts are identified.*** | |
| | ***The actual sizes have been entered correctly.*** | |

| | **Planning Summary** | **Comments** |
|---|---|---|
| | ***The header information is complete and correct.*** | |
| | ***Estimated productivity has been entered.*** | |

| | **Consistency Checks** | **Comments** |
|---|---|---|
| | Defects removed are consistent with time in phase and prog. size. | |
| | Total test defect fix times are less than test time. | |
| | Defect dates and phases are consistent with the time log. | |

| | **General** | **Comments** |
|---|---|---|
| | Followed the defined process. | |
| | Complete, consistent and accurate process data collected. | |
| | The student did his or her own work. | |
| | ***Historical data are used in planning the work.*** | |