

# Generating Mathematical Exercises for E-learning Systems Using R and QTI Import Format

Eckhard Liebscher <sup>1\*</sup>, Ben Michael <sup>2</sup>

<sup>1</sup> University of Applied Sciences Merseburg, GERMANY

<sup>2</sup> Optimax Energy Leipzig, GERMANY

\* CORRESPONDENCE: ✉ [eckhard.liebscher@hs-merseburg.de](mailto:eckhard.liebscher@hs-merseburg.de)

## ABSTRACT

In the paper a software solution is presented for the generation of instances of exercises. The focus is on education in mathematics but the approach can be adopted in other fields such as computer science and natural science. The present solution uses the R environment for the generation process and LaTeX for the content presentation. In the established system, we realized the separation of computations and design which has the advantage that it is easier to find errors and shortcomings.

**Keywords:** mathematics exercises, item generation, R

## INTRODUCTION

Generating a large number of exercises automatically for the training of topics within mathematics education is a challenging task. In this paper, we provide a software solution to generate xml-files according to the QTI standard. These files contain exercises to be imported into a learning management system (LMS). In our implementation we used the Ilias LMS. Many LMSs such as Ilias have an xml/QTI input filter and an xml/QTI output interface available. In the established system, the files with exercise items are prepared for the input via this interface. The generation process is done within the R environment. Similar solutions are only rarely to be found in the literature. In this paper, we explain how to write R scripts which can be deployed to generate as many instances of a specific type of exercises as desired. Here the focus is on exercises in mathematics but the ideas can be used for exercise generation in similar areas such as natural sciences and computer science. The specific aims of mathematics education are studied in detail in Maron (2016). The importance of using modern information technologies in mathematics education is worked out in Mendoza and Mendoza (2018).

In modelling the structure of exercises, we follow the ideas of the approach by Gierl and colleagues. We refer to Gierl et al. (2015) and some other papers of these authors. There they describe thoroughly the development of an item model and its usage in the generation of a large number of items. In Section 2 we refine the item model and adapt it to mathematical exercises.

In our established system, we separate computations and the content of exercise items including the design of the exercises. The computations are done using the R system and a special R script. This approach has the advantage that we can do experiments in R until reaching the final version without changing the text content of the exercises. Since the R script is strongly adapted to the specific aim, there is no need for a special R package. The text content and the design of the exercises are defined in a document using elements of LaTeX (LaTeX-like file). Since a LaTeX file containing all instances is one of the output files of the R script, a pdf-file

---

**Article History:** Received 17 January 2019 ♦ Revised 9 May 2019 ♦ Accepted 9 May 2019

© 2019 by the authors; licensee Modestum Ltd., UK. Open Access terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>) apply. The license permits unrestricted use, distribution, and reproduction in any medium, on the condition that users give exact credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if they made any changes.

can be provided using the LaTeX system. This pdf file can be used to check all instances and to find out errors and shortcomings. Problematic instances of exercises can be identified and then simply be removed from the file. An R script and a LaTeX-like file are the input files into our system which has to be prepared by the user.

Another approach is connected with the R package exams for the automatic generation of examination exercises, see Grün and Zeileis (2009), Zeileis, Umlauf and Leisch (2014). In contrast to our approach, the idea of these authors is to mix parts of R script for computations and LaTeX document parts to design the exercises. Moreover, our approach of external generation of exercises differs from solutions inside the LMS. In some LMSs, there is an exercise type STACK available where an online generation of the exercises using random numbers is realised.

Two roles are important concerning the usage of the e-learning system: the learner and the teacher. The teacher establishes the exercises and provides the correct results. The implementation of exercises in e-learning systems requires a time-consuming effort for the teacher. One aim of the proposed system is to support the process of the establishment of questions. If these files with the content of assessments are generated automatically, then it leads to a substantial reduction of the effort for the teacher.

The task for the learner is to train in the solving of mathematical exercises in the framework of lessons. The learner wants to get a good preparation for the examinations. In most situations, it is important to include some intermediate checks of results imported into the system. If the learner is only able to solve the exercise partially then he or she obtains valuable information about the degree to what he or she achieved the correct solution. From a psychological point of view, it stimulates the learner to a larger amount to pursue the perfect solution. There is a strong impact of the quality of the feedback on the students' motivation for further learning. This was shown in studies reported in Rakoczy et al. (2008). The role of the teacher in engaging students to e-learning is examined in Yengin et al. (2010). It should be pointed out that the usage of the developed system offers only one possibility to improve the mathematical skills. We cannot expect that e-learning contributes to all facets of learning outcomes. E-learning cannot substitute completely or in big parts traditional learning techniques.

An important requirement is that the concrete instance of an exercise fetched from the LMS should vary from fetch to fetch. Otherwise the learner can memorize the solutions or can get the solution from other sources. This does not lead to a gain of knowledge and skills. Here we generate randomly the parameters of the exercises. The teacher has to provide the range for the parameter values prior to implementation.

Another important issue is the usability of the developed system (cf. the paper by Masemola and De Villiers (2006)). Since the focus of this paper is on other aspects, we decided to conduct only a small survey among the students of our university. Furthermore, we analyzed the assessment results of one course of our university. The outcomes of these studies were used to validate the developed system.

The present paper is structured as follows: Section 2 deals with the exercises and its structure. The software and its usage are described in Section 3. The example discussed in Section 3 can be regarded as a template for other applications. Special exercises were prepared for the Ilias LMS in our university. The results of a case study among the students of our university are discussed in Section 4. The **Appendix** at the end contains the corresponding documents for an example introduced in Section 2.

## EXERCISES IN THE E-LEARNING SYSTEM - THE ITEM MODEL

In an e-learning system, exercises are deployed to assess the progress in acquisition of knowledge and skills the learner achieved. Moreover, the learner gets knowledge about the point at which he fails in getting the solution of the exercise. In this section we develop further the item model given by other authors, see Gierl and Lai (2012) for example. Concerning detailed accounts on item models, we refer to Bejar et al. (2003). Technically, the exercises consist of the following components which are parts of the item model:

- (1) title
- (2) metadata: authors, topic, pedagogical aims
- (3) text of the exercise
- (4) text of a correct solution
- (5) description of the gaps
- (6) parameters and their domains, conditions on the parameters, generation distribution

(7) computation formulas or algorithm for the generation

(8) conditions on intermediate results

The item model can be used for a formalized preparation of the specific exercise template. To prepare a specific type of exercises, we have to fill all the points of the item model with content. Considering a certain exercise type, the points (3), (4) and (5) contain the contents and the design of the exercise whereas information about the generation of the specific parameter values are included in (6)-(8). So contents and generation are separated from one another.

In (3) and (4) the parameters appear encoded in a specific predefined form. These encoded parameters will be replaced by numbers, texts or formulas in the generation process. The gaps realize the input of the results into the LMS. Practically, there are input masks where the learner has to enter the specific results. Then the LMS checks the result and provides a feedback for the result. We assume that the checking process is fully done by the LMS without any option of interruption for the author (as it is done in Ilias).

In our approach, the generation of the specific exercise instances is done by external programs. The parameter values are generated according to the generation distribution which is the uniform distribution in most cases. For the generated parameter values, the conditions in (6) have to be checked. These conditions have to ensure that the exercise is interesting enough and not too simple. After computing all the results, the conditions on the intermediate results have to be checked. One requirement is that the intermediate results should have an easy-to-handle form. Numerical values with a lot of digits or huge numbers, long texts or complicated formulas should be avoided.

For mathematical exercises, the following types of result queries are used and available in the system under focus:

- numerical gaps
- text gaps
- single choice gaps

Other types are also available but not relevant in our context. The LMS checks the validity of the entered reply. In case of numerical gaps, the input value has to belong to a narrow interval around the true result. This interval is to be provided by the teacher. For text gaps, the system simply checks whether the input matches the correct results.

In the following, we want to explain how to apply the approach for exercises from probability theory. Now we give the example for this explanation:

**Example exercise:** We want to generate exercises for the computation of probabilities. The learning goal is to be able to evaluate probabilities in the model of a normal distribution. The learner should be able to calculate probabilities for intervals by using appropriate tools like a table or a computer for the computation of the distribution function of a standard normal distribution. The text of the exercise is as follows:

In Mr. Chickill's farm the weight of chicken aged 10 weeks is normally distributed with expectation of  $a$  kg. The variance of the weight is  $b$ .

- a) How large is the probability for the weight being between  $c$  and  $d$  kg?
- b) Evaluate the probability that the weight is at least  $e$  kg.

The quantities  $a, b, c, d, e$  are the parameters of the exercise which will be replaced in the processing. The conditions on the parameters are as follows:  $a, b > 0, 0 < c < d, c \neq e, d \neq e, a \neq e$ . The text of the correct solution depending on the parameters can be found in the Appendix.

The detailed sample solutions are an important element of our approach. They enable the student to comprehend the solution techniques. Moreover, the students can find starting points for studying the material of the course and material from other sources to improve their theoretical skills. These skills are related to specific concepts, facts, formulas, relationships etc. of the considered topic. A good knowledge of them helps in turn students in solving the exercises of the considered topic. This learning process can be supported by providing helpful links in the implementation in the LMS.

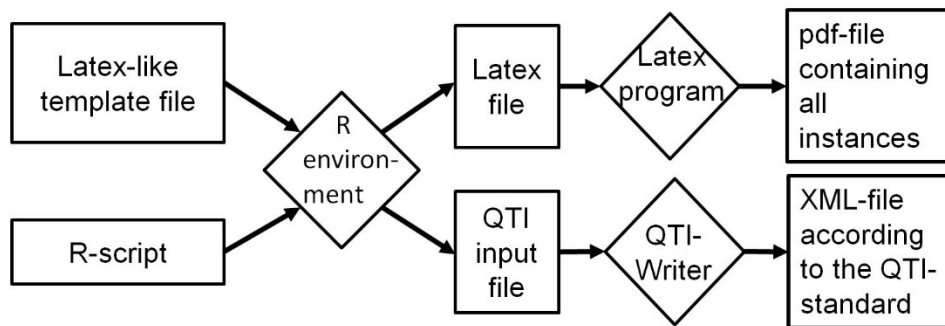


Figure 1. Structure of the system

## DESCRIPTION OF THE ESTABLISHED SYSTEM

### Structure of the System

Figure 1 shows the structure of the system.

Two input files have to be supplied by the teacher: the LaTeX-like template file and the R script which are considered in the next subsections. The xml/QTI files are generated by using first the R environment and secondly a java program. If the user is interested in a pdf-file containing all generated exercises and solutions then he or she can input the file into a LaTeX editor and environment (for example TexStudio or Texnic center). A special framework file is provided in our system. This file together with the LaTeX file containing the exercise items is compiled using a LaTeX compiler, pdf2latex for example. Eventually, we obtain the required pdf-file with all exercises.

### LaTeX-like Template File

This file provides the pattern for the exercises to generate. It has the following special structure where input fields begin and end with dots "...". The description of the input is given in italics.

```

\begin{question}
\authorq{... author name ...}
\titleq{... title of the exercise...}
\descriptionq{... specific description of the exercise ...}
\begin{textq}
... text of the exercise ...
\end{textq}
\gap{...text...}{... value ...}{... lower limit ...}{... upper limit ...}{... points ...}{... width ...}

```

➡ within braces: *text* before the input mask, exact *value*, *lower and upper limit* of the interval for a positive assessment, *points* in case of a positive assessment, *width* of the input mask

```

\gapstext{... text ...}{... input text ...}{... points ...}{... width ...}

```

➡ within braces: *text* before the input mask, correct *input text* for a positive assessment, *points* for the assessment, *width* of the input mask

```

\begin{solution}
... solution of the exercise ...
\end{solution}
\end{question}

```

The author of the exercise has to replace the fields within ..., the dots included, by the corresponding string. Strings starting with @ are connected with the R script. In the R script, specific variable names (parameter)

are declared, typically  $a, b, c, \dots$  ( $@a, @b, \dots$  in the LaTeX-like template file). The actual parameter value varies from one individually generated exercise to the other. They are generated by random number generation with a specific distribution defined in the R script. Typically the uniform distribution is used. In addition, the identifiers  $@rr1, @rr2, \dots$  are reserved for the results in the solution. The identifiers  $@rl1, @ru1, @rl2, \dots$  are connected with the lower and upper interval limits. The numerical result is assessed by the LMS as correct if the input result is within these interval limits. The “text of the exercise” has to be inserted as simple text (not in LaTeX, HTML tags allowed) including two special parts:

- \newline commands for the typesetting,
- LaTeX parts in MathJax style.

MathJax is a cross-browser JavaScript library for displaying mathematical formulas. MathJax is supported by a lot of LMS, for example Ilias.

### The R Script

The teacher has to write an R script for the evaluations. The R script is executed to compute the values for the @-identifiers in the exercise template (LaTeX-like file). All identifiers beginning with @ such as  $@a, @b, @rr1, @rr2, \dots$  are replaced by numbers or strings in the LaTeX-like file. The R program should contain the implementation for the random generation of concrete values for the exercise variables  $@a, @b, @c, \dots$  and for the evaluations of the results including intermediate results of the sample solution. These results represented by variables  $@rr1, @rr2, \dots, @rl1, @ru1, @rl2, \dots$ . For the parameters the range for the random choice is to be defined. For establishing the script, a template and useful R functions are available. By random generation of the parameters, we obtain a large number of diverse instances of a certain exercise type. For the example of Section 2, the specific files are given in the Appendix.

### The QTI-input File

The QTI-input file is a LaTeX-like file which differs in two features from the LaTeX file: The text parts are given without LaTeX commands, the mathematical parts are given in LaTeX/MathJax style. Typical delimiters for the MathJax parts are  $\$, \backslash($  and  $\backslash)$ ,  $\backslash[$  and  $\backslash]$ . The QTI-writer program takes the QTI-input file and generates the xml-file.

### The XML File Following QTI Format

The IMS Question and Test Interoperability specification (QTI) defines a standard format for assessments at the interface between the e-learning system and the teacher. In our system, the QTI-formatted file contains the e-assessment exercises including a correct solution. In our framework, the criticism about this file type (cf. Piotrowski (2011)) is only partially relevant since xml/QTI-files appear only in an intermediate stage of the preparing process. Checks of intermediate results and evaluative feedbacks are supported by the learning management system. The corresponding elements have to be implemented in the QTI-file generation.

## A CASE STUDY FOR VALIDATING THE SYSTEM

### Design and Methods

At the University of Merseburg, the developed system was offered for use to students of the Bachelor course of Business Administration. Exercises from probability theory and statistics were implemented in the e-learning system Ilias. At most 85 students trained the solution of exercises in the basic course of Statistics during one semester. Each student could conduct an unlimited number of runs for each test. The collected assessment data were analyzed to validate the developed system.

The test system under consideration comprised three exercise tests: one about “computation of probabilities” (77 participants), one about “discrete random variables” (85 participants) and one on the topic of “continuous random variables” (83 participants). When entering an exercise test, the student had to solve three exercises. The achieved assessment results were recorded in an excel file. They include the achieved number of assessment points for each trial, separately for each student and for each exercise. We computed simple descriptive statistics of the assessment points. Some results are surveyed in the following section.

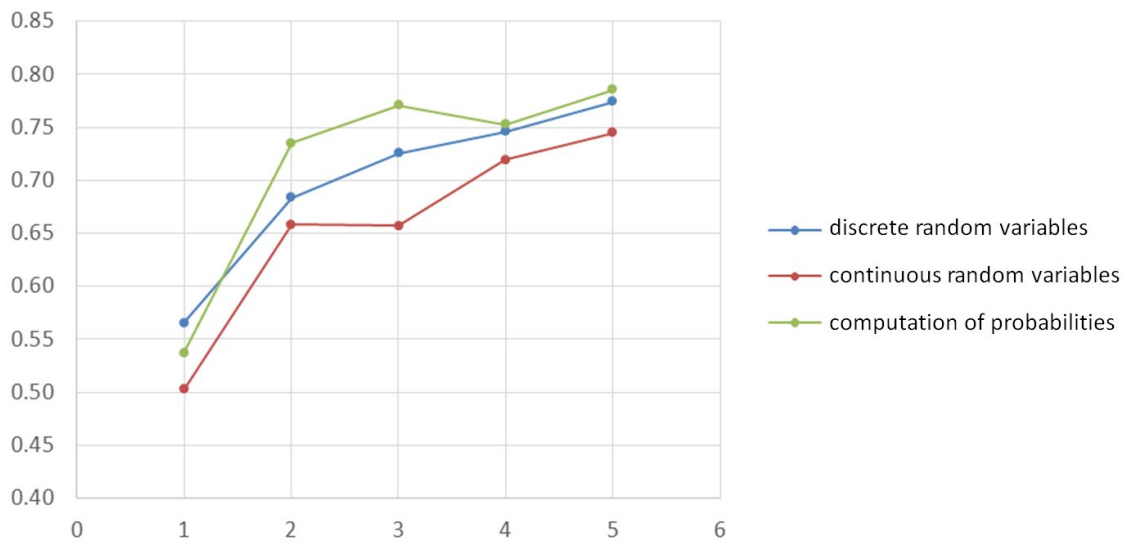


Figure 2. Average fraction of achieved points versus the trial number

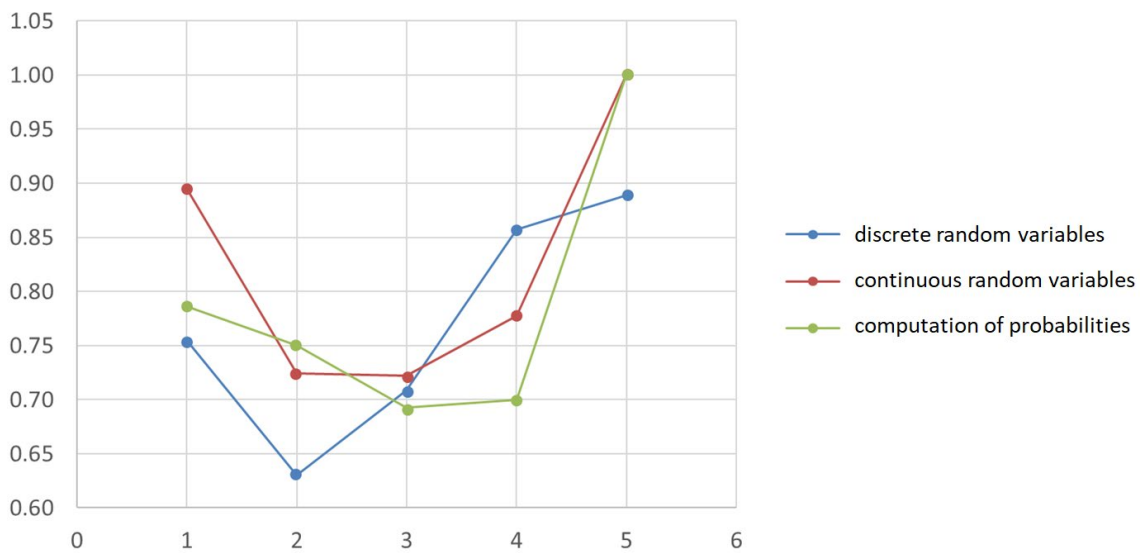


Figure 3. Average gain of a trial in comparison to the previous one

### Results

First of all, we give some results from the analysis of the assessment data. It is to be seen in **Figure 2** that the average of points achieved in the assessment increases with the number of trained exercises. However, the slope is much smaller for later trials because of the smaller improvement potential. Next we want to discuss the success of one training step. This success can be measured by the conditional probability that the learner achieved more points in the succeeding trial than in the previous one, given there are two or more points left to the maximum number of points in the previous trial. Summarizing the results (see **Figure 3**), some features can be recognized from **Figure 3**. First, there is a success at the beginning. In the middle, the success rate decreases. In later trials we see an effect of longer training. It means that hard and long training improves significantly the students' skills.

In the framework of the study, we also analyzed the usage and the usability. We give only some short impressions about the results. Among other aspects, 85% of the students recommended the tests. 67% of the students intend to use the test regularly. For 89% of the students, the usage of the test system was easy and

did not cause trouble. We observed that students who used our system achieved better results in examinations in average than others. Detailed investigations should be carried out to ascertain the generality of this observation. Altogether, we can conclude that in our university, the system is successfully applied to improve the students' skills in mathematics.

## CONCLUSIONS

The system under consideration is able to support the teacher in generating a large amount of exercises. In the described solution the formulas for the computations of the results are separated from the presentation of the exercises in the LMS. It has the advantage that the presentation of the exercises is more transparent. The details in the R implementation do not confuse and distract the user. He or she can easily experiment with the R script before coming up with the final version which realises the computations perfectly. The teacher can concentrate on the preparation of the instances for the exercises.

In summary, the following features distinguish our system from other approaches, and they define the particular advantages of our system:

- separated files for mathematical computations and content including design (advantageous for finding errors),
- sample solutions of the exercises can be generated as well,
- a template for the R script including helpful R functions is available to support the process of implementation,
- the system is practically used in a Ilias learning management system for education in mathematics.

In the future, there is a lot research work to be done to develop a better support of writing the R scripts and the LaTeX-like input files. Moreover, we want to extend the approach to other learning management systems such as Moodle.

## Disclosure statement

No potential conflict of interest was reported by the authors.

## Notes on contributors

**Eckhard Liebscher** – University of Applied Sciences Merseburg, Germany.

**Ben Michael** – Optimax Energy Leipzig, Germany.

## REFERENCES

- Bejar, I. I., Lawless, R. R., Morley, M. E., Wagner, M. E., Bennett, R. E., & Revuelta, J. (2003). A Feasibility Study of On-the-Fly Item Generation in Adaptive Testing. *Journal of Technology, Learning, and Assessment*, 2(3), 3-28. <https://doi.org/10.1002/j.2333-8504.2002.tb01890.x>
- Gierl, M. J., & Lai, H. (2012). The Role of Item Models in Automatic Item Generation. *International Journal of Testing*, 12, 273-298. <https://doi.org/10.1080/15305058.2011.635830>
- Gierl, M. J., Lai, H., Hogan, J. B., & Matovinovic, D. (2015). A Method for Generating Educational Test Items that are Aligned to the Common Core State Standards. *Journal of Applied Testing Technology*, 16(1), 1-18.
- Grün, B., & Zeileis, A. (2009). Automatic Generation of Exams in R. *Journal of Statistical Software*, 29(10), 1-14. <https://doi.org/10.18637/jss.v029.i10>
- Maron, A. I. (2016). Priorities of teaching mathematics in universities. *IEJME–International Electronic Journal of Mathematics Education*, 11(9), pp. 3339-3350
- Masemola, T., & De Villiers, R. (2006). Towards a framework for usability testing of interactive e-learning applications in cognitive domains, illustrated by a case study. *Proceedings of the 2006 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on IT Research in Developing Countries*, 187-197. <https://doi.org/10.1145/1216262.1216283>

- Mendoza, D. J., & Mendoza, D. I. (2018). Information and Communication Technologies as a Didactic Tool for the Construction of Meaningful Learning in the Area of Mathematics. *IEJME–International Electronic Journal of Mathematics Education*, 13(3), 261-271. <https://doi.org/10.12973/iejme/3907>
- Piotrowski, M. (2011). QTI: A Failed E-Learning Standard? In *Handbook of Research on E-Learning Standards and Interoperability: Frameworks and Issues*. IGI Global Publisher. Editors: Fotis Lazarinis, Steve Green, Elaine Pearson. <https://doi.org/10.4018/978-1-61692-789-9.ch004>
- Rakoczy, K., Klieme, E., Bürgermeister, A., & Harks, B. (2008). The Interplay Between Student Evaluation and Instruction. *Journal of Psychology*, 216(2), 111–124. <https://doi.org/10.1027/0044-3409.216.2.111>
- van der Kleij, F. M., Eggen, T. J.H.M., Timmers, C. F., & Veldkamp, B. P. (2012). Effects of feedback in a computer-based assessment for learning. *Computers & Education*, 58(1), 263-272. <https://doi.org/10.1016/j.compedu.2011.07.020>
- Yengin, I., Karahoca, D., Karahoca, A., & Yücel, A. (2010). *Procedia Social and Behavioral Sciences*, 2, 5775-5787. <https://doi.org/10.1016/j.sbspro.2010.03.942>
- Zeileis, A., Umlauf, N., & Leisch, F. (2014). Flexible Generation of E-Learning Exams in R: Moodle Quizzes, OLAT Assessments, and Beyond. *Journal of Statistical Software*, 58(1), 1-36. <https://doi.org/10.18637/jss.v058.i01>

## APPENDIX

### Example exercise - input file:

```

\begin{question}
\authorq{Eckhard Liebscher}
\titleq{continuous distributions}
\descriptionq{Computing probabilities of a continuous distribution}
\begin{textq}
In Mr. Chickill’s farm the weight of chicken aged 10 weeks is normally distributed
with expectation of  $\$@a\$$  kg. The variance of the weight is  $\$@b\$$ . \newline
a) How large is the probability for the weight being between  $\$@c\$$  and  $\$@d\$$  kg?
\newline
b) Evaluate the probability that the weight is at least  $\$@e\$$  kg.
\end{textq}
\gap{a} standard deviation}{@rr1}{@rl1}{@ru1}{1}{8}
\gap{ probability}{@rr6}{@rl6}{@ru6}{2}{8}
\gap{b} probability}{@rr9}{@rl9}{@ru9}{1}{8}
\begin{solution}
 $X \sim \mathcal{N}(@a,@b)$  weight of the chicken \newline
 $\sigma = \sqrt{@b} = @rr1$  is the standard deviation \newline
 $\left[ \begin{array}{l} \text{a) } P(@c \leq X \leq @d) = \Phi \left( \frac{@d - @a}{\sigma} \right) - \Phi \left( \frac{@c - @a}{\sigma} \right) \\ \text{b) } P(X \geq @e) = 1 - \Phi \left( \frac{@e - @a}{\sigma} \right) = 1 - \Phi (@rr7) = 1 - @rr8 = @rr9 \end{array} \right]$ 
\end{solution}
\end{question}

```



**parameter:**  $a \in [1.1, 1.5]$ , step size in the grid 0.01

$b \in [0.02, 0.05]$ , step size in the grid 0.001

$\sigma = \sqrt{b}$  standard deviation

$c \in [a - 2.6\sigma, a - 0.2\sigma]$ , step size in the grid 0.01

$d \in [a + 0.2\sigma, a + 2.6\sigma]$ , step size in the grid 0.01

$e \in [a - 2.6\sigma, a + 2.6\sigma]$ , step size in the grid 0.01

**conditions:**  $c \neq e$ ;  $d \neq e$ ;  $a \neq e$ ;

### Generated LaTeX file (one exercise)

```

\begin{question}
\authorq{Eckhard Liebscher}
\titleq{continuous distributions}
\descriptionq{Computing probabilities of a continuous distribution}
\begin{textq}
In Mr. Chickill's farm the weight of chicken aged 10 weeks is normally distributed
with expectation of $1.31$ kg. The variance of the weight is $0.021$. \newline
a) How large is the probability for the weight being between $1.16$ and $1.5$
kg? \newline
b) Evaluate the probability that the weight is at least $1.46$ kg.
\end{textq}
\gap{a} standard deviation){0.14491}{0.14481}{0.14501}{1}{8}
\gap{ probability){0.75478}{0.75378}{0.75578}{2}{8}
\gap{b) probability){0.15031}{0.14931}{0.15131}{1}{8}
\begin{solution}
 $X \sim \mathcal{N}(1.31, 0.021)$  weight of the chicken \newline
 $\sigma = \sqrt{0.021} = 0.14491$  is the standard deviation \newline
\[\begin{align*} \text{a) } P(1.16 \leq X \leq 1.5) &= \Phi\left(\frac{1.5-1.31}{\sigma}\right) - \Phi\left(\frac{1.16-1.31}{\sigma}\right) \\ &= \Phi(1.3111) - \Phi(-1.0351) = 0.90509 - 0.15031 = 0.75478 \\ \text{b) } P(X \geq 1.46) &= 1 - \Phi\left(\frac{1.46-1.31}{\sigma}\right) = 1 - \Phi(1.0351) = 1 - 0.84969 = 0.15031 \end{align*}]
\end{solution}

```

### LaTeX output

Author: Eckhard Liebscher

Title: continuous distributions

Description: Computing probabilities of a continuous distribution

In Mr. Chickill's farm the weight of chicken aged 10 weeks is normally distributed with expectation of 1.31 kg. The variance of the weight is 0.021.

a) How large is the probability for the weight being between 1.16 and 1.5 kg?

b) Evaluate the probability that the weight is at least 1.46 kg.

Gap

Label: a) standard deviation, Value: 0.14491, lower interval limit: 0.14481, upper interval limit: 0.14501,

Points: 1, Gap width: 8

Gap

Label: probability, Value: 0.75478, lower interval limit: 0.75378, upper interval limit: 0.75578,

Points: 2, Gap width: 8

Gap

Label: b) probability, Value: 0.15031, lower interval limit: 0.14931, upper interval limit: 0.15131,

Points: 1, Gap width: 8

Solution:  $X \sim N(1.31; 0.021)$  weight of the chicken

$\sigma = \sqrt{0.021} = 0.14491$  is the standard deviation

a)

$$P(1.16 \leq X \leq 1.5) = \Phi\left(\frac{1.5-1.31}{\sigma}\right) - \Phi\left(\frac{1.16-1.31}{\sigma}\right)$$

$$= \Phi(1.3111) - \Phi(-1.0351)$$

$$= 0.90509 - 0.15031 = 0.75478$$

b)

$$P(X \geq 1.46) = 1 - \Phi\left(\frac{1.46-1.31}{\sigma}\right)$$

$$= 1 - \Phi(1.0351)$$

$$= 1 - 0.84969 = 0.15031$$

<http://www.iejme.com>