Journal of Statistical Software

March 2011, Volume 39, Issue 7. http://www.jstatsoft.org/

An Open Source Approach for Modern Teaching Methods: The Interactive TGUI System

Gerlinde Dinges
Statistics Austria

Alexander Kowarik
Statistics Austria

Bernhard Meindl
Statistics Austria

Matthias Templ
Statistics Austria

Abstract

In order to facilitate teaching complex topics in an interactive way, the authors developed a computer-assisted teaching system, a graphical user interface named TGUI (Teaching Graphical User Interface). TGUI was introduced at the beginning of 2009 in the *Austrian Journal of Statistics* (Dinges and Templ 2009) as being an effective instrument to train and teach staff on mathematical and statistical topics. While the fundamental principles were retained, the current TGUI system has been undergone a complete redesign. The ultimate goal behind the reimplementation was to share the advantages of TGUI and provide teachers and people who need to hold training courses with a strong tool that can enrich their lectures with interactive features. The idea was to go a step beyond the current modular blended-learning systems (see, e.g., Da Rin 2003) or the related teaching techniques of classroom-voting (see, e.g., Cline 2006). In this paper the authors have attempted to exemplify basic idea and concept of TGUI by means of statistics seminars held at Statistics Austria. The powerful open source software R (R Development Core Team 2010a) is the backend for TGUI which can therefore be used to process even complex statistical contents. However, with specifically created contents the interactive TGUI system can be used to support a wide range of courses and topics. The open source R packages **TGUICore** and **TGUITeaching** are freely available from the Comprehensive R Archive Network at `http://CRAN.R-project.org/`.

Keywords: R, active learning, blended-learning, classroom-voting, computer-assisted teaching.

1. Introduction

Lecturers are constantly challenged to prepare and facilitate complex subjects and to deliver the data in a manner their students can both, understand and benefit through attendance and thus improve their knowledge base. In 2005, the staff of the methodological division at Statistics Austria were challenged with the task of preparing basic statistics courses offered as in-house employee training. The goal was to deliver training courses which would be open to everybody. The task was further exacerbated by the fact that everyone should be able to understand even complex content including statistical terms and issues, e.g., the central limit theorem or the basic concepts of robust statistics, through active participation. To put this into practice and to avoid excessive hours of classroom teaching, a computer-assisted teaching system, a graphical user interface called TGUI (Teaching Graphical User Interface), was developed. TGUI is both an environment that allows lecturers to enrich their courses with interactive features getting instant feedback on any given example or posed question, as well as students to be actively integrated without any requirements with respect to software or programming skills. Moreover, the TGUI system enables course instructors to interact with their students by periodically toggling between classroom teaching and active student contribution.

The starting point for the development of TGUI was the statistics courses at Statistics Austria. In this paper we begin with the main ideas and concepts which led to the development of the framework (Section 2). To outline how the system could be practically implemented, some examples from our current training courses are shown in Section 3. The interactive system of TGUI is freely available as the open source R package **TGUICore** which contains everything required to create interactive examples or to ask for student feedback. **TGUICore** was developed to facilitate the application of TGUI and to separate technically needed code from user-generated code. Section 4 outlines details of the technical implementation of **TGUICore**. Section 5 introduces **TGUITeaching**, a generic content R package working as a prototype to create new interactive training materials using the TGUI system. To conclude, a simple example explaining the creation of a custom package supported by TGUI is shown.

2. Motivation and underlying ideas

Nowadays students sometimes be subjected to hours of nonstop lectures in passive mode – that does not come up to modern teaching methods. Particularly, statistical topics often seem to be quite theoretical and abstract to students, and are therefore not quite easy to teach.

2.1. Main ideas and key concepts

In order to increase student interest and to improve understanding of the subject matter it is necessary to prepare theoretical and complex content in a self-explanatory way. This goal can be achieved by trying to visualize statistical concepts wherever possible. Furthermore, student attention can be raised by including student-related data directly (on-the-fly) via teaching-slides. The key idea and the main goal is to involve students in a lecture from the very beginning by using modern computer techniques that include interactive teaching elements and features. Based on previous courses covering statistical topics, the following six key concepts were identified to improve insight and course experience, and these are now taken into account when preparing for future training courses.

Virtual teaching book

Firstly one needs to define and structure the relevant topics and to compile user-friendly slides. Each course participant is provided with a screen for the course slides to be viewed during classroom teaching. In doing so, the participants general attention is increased, audio-vision problems are decreased and the course becomes more hands-on.

Maximizing Visualization

Visualization wherever possible, improves insight and understanding and can facilitate the teaching of theoretical and complex topics. On a statistics course, per say, the aim is not only to explain key terms of statistics but also to allow students actually "see" statistics in order to better understand the subject. To this end, an issue should always be visualized or at least accompanied by an easy to understand example in the virtual teaching book.

Self-motivation via student-generated data

It is imperative that students relate to the examples. When teaching statistical content this goal can therefore be achieved by using either real-life datasets with which course participants deal in their job, or by using data that has been entered by the students during the course. A example could be to allow students to flip a coin, this would lead to an introductory example for randomness. Another possibility would be asking students questions about current affairs, the outcome of an election for example, to support the visualization of a simple distribution and increasing interest.

Active participation from all students

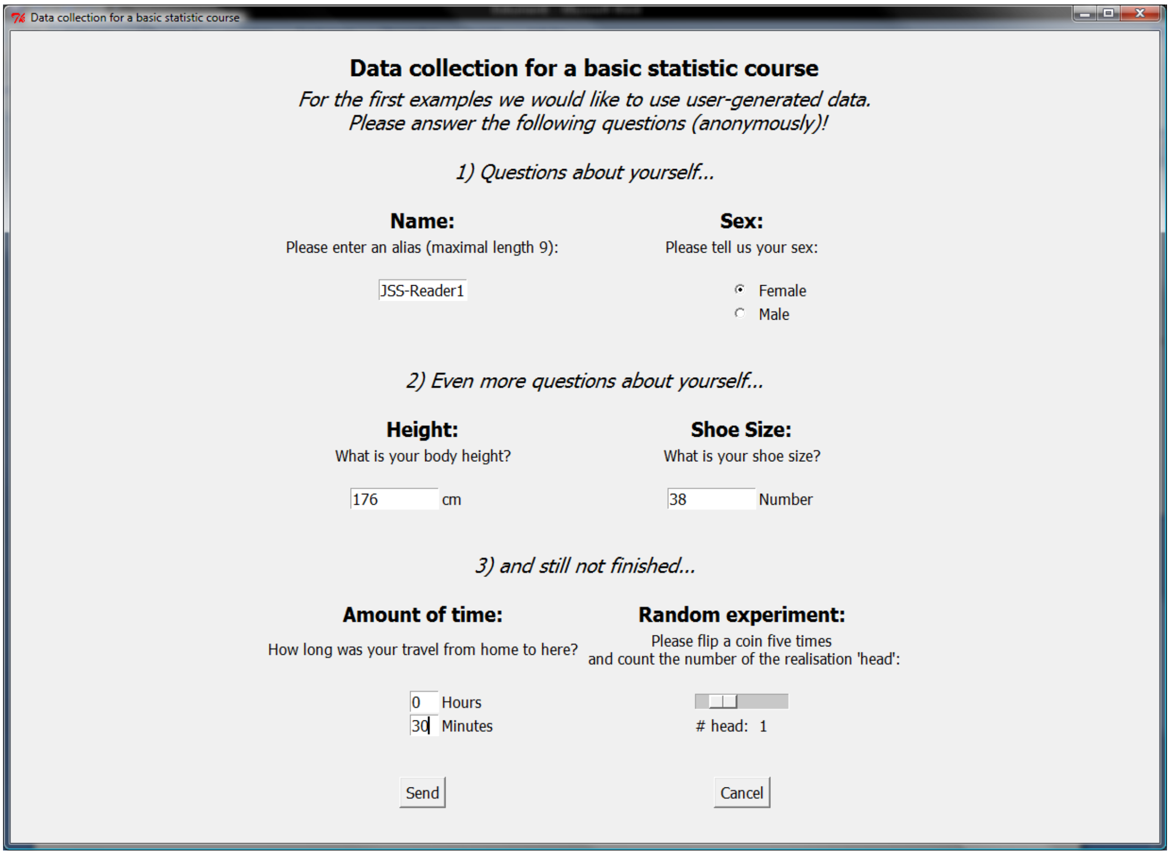
A hybrid form of classroom teaching and practical applications is commonly accepted as best teaching practice (see, e.g., Brown 2001; Ward 2004; Blakesley Lindsay 2004). Active Learning methods (see, e.g., Silberman 1996; Cline and Lomen 2010) which include students actively participating through teamwork, role-playing, case studies or written exercises are effective, but these prove time consuming. The objective is to motivate the students to actively participate in the lecturing process therefore avoiding mechanical classroom teaching methods. The challenge is that the realization of participation is not only time-consuming but in some cases have proved to be simply impossible because of a lack of feedback or missing skills such as the knowledge of one common software language. If these skills were mandatory for active participation, a percentage of potential course participants would be excluded in the first place.

Interactive checks and instant feedback for everybody

The main purpose of interactive checks is to finalize the current topic or to motivate and introduce the next one. Instant feedback provides the lecturer the opportunity to determine whether the course group understood a topic thoroughly or if further explanation is needed. Interactive checks may be seen as occasional brain calisthenics. In the case of suitable examples being prepared in advance, students would then have room to train and strengthen their knowledge about the taught topic.

Anonymous (textual) feedback-facility for everybody

Avoiding the often patronizing prompt "Any further questions?", the idea of interactive checks is enhanced by an anonymous textual feedback-facility. Moreover, the course instructors can formulate their own pertinent questions beforehand, ensuring that important key issues are raised and revised. Questions for improving future courses can also be included.



Data collection for a basic statistic course

Data collection for a basic statistic course

For the first examples we would like to use user-generated data.
Please answer the following questions (anonymously)!

1) Questions about yourself...

Name:
Please enter an alias (maximal length 9):
JSS-Reader1

Sex:
Please tell us your sex:
● Female
○ Male

2) Even more questions about yourself...

Height:
What is your body height?
176 cm

Shoe Size:
What is your shoe size?
38 Number

3) and still not finished...

Amount of time:
How long was your travel from home to here?
0 Hours
30 Minutes

Random experiment:
Please flip a coin five times
and count the number of the realisation 'head':
# head: 1

Figure 1: Example for a TGUI data input mask.

2.2. Putting ideas into practice

To include all the above mentioned concepts a computer-assisted teaching system with a graphical user interface is required. For this reason TGUI – an open source approach for modern teaching techniques – has been developed.

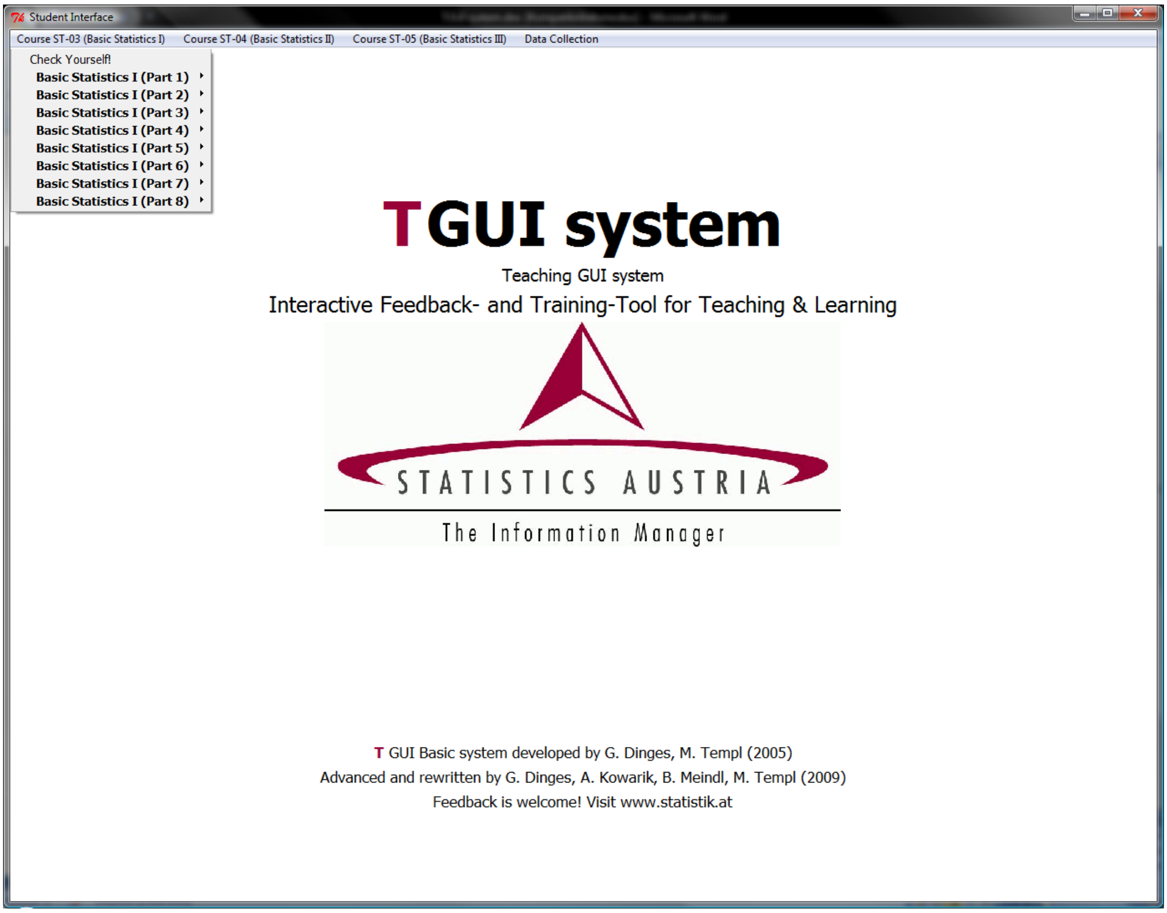
When supporting a course using a system like TGUI there is initially a need for some technology input, because one requires an adequate computer network to be in place and the aforesaid software must be available (see Section 5).

In addition, both suitable and relevant examples must be prepared in advance (for a library see e.g., Zullo and Cline 2010) and embedded in the system.

To allow for data input each participant must be provided with a personal computer or laptop which would in turn be linked to a network used for the course. However, during classroom teaching the course slides should be displayed on all screens so that students do not get lost in the world-wide-web while the teacher is trying to explain a topic.

2.3. Benefits of TGUI

At the conclusion stage of a topic the lecturer's prompt for comments, suggestions and feedback is often answered by very few – if any – students. Ensuring active participation of


Student Interface
Course ST-03 (Basic Statistics I)
Course ST-04 (Basic Statistics II)
Course ST-05 (Basic Statistics III)
Data Collection
Check Yourself!
Basic Statistics I (Part 1)
Basic Statistics I (Part 2)
Basic Statistics I (Part 3)
Basic Statistics I (Part 4)
Basic Statistics I (Part 5)
Basic Statistics I (Part 6)
Basic Statistics I (Part 7)
Basic Statistics I (Part 8)
STATISTICS AUSTRIA
The Information Manager

TGUI system

Teaching GUI system

Interactive Feedback- and Training-Tool for Teaching & Learning

T GUI Basic system developed by G. Dinges, M. Templ (2005)
Advanced and rewritten by G. Dinges, A. Kowarik, B. Meindl, M. Templ (2009)
Feedback is welcome! Visit www.statistik.at

Figure 2: TGUI main menu adjusted for courses at Statistics Austria.

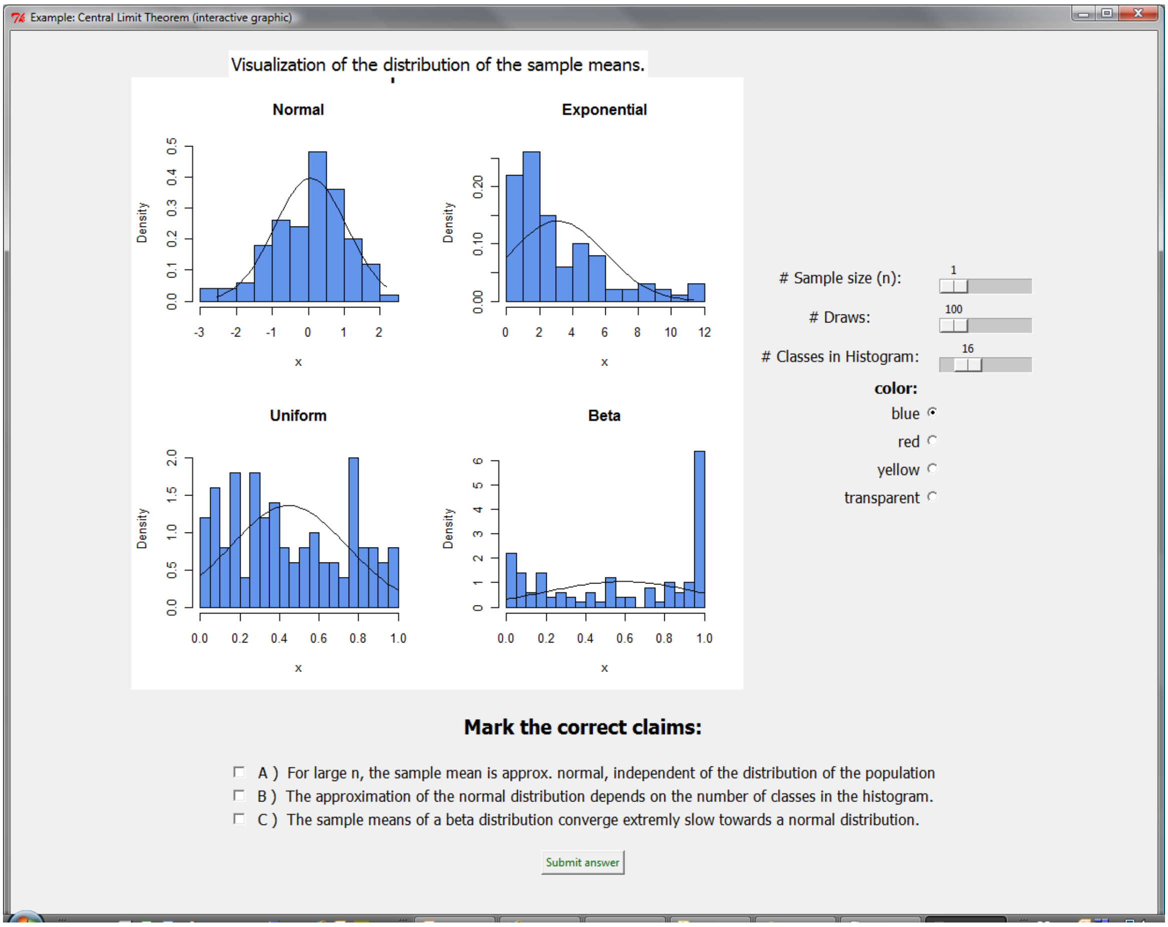
everyone is not easily achieved, but essential for better understanding of complex content.

TGUI supports the principle of game-based-learning and facilitates participation of all students without burden.

The benefit of TGUI is that this modern teaching approach turns monologue into dialogue. This is a major advantage for teachers because instant feedback helps to assess whether the content has been understood. It gives the possibility to re-explain the most important terms and contents directly. Course instructors have additional information to improve the quality of their courses by monitoring the students' understanding.

The main advantages for the course participants are that they can deepen their knowledge playfully. Due to the anonymity, students can give feedback to the course instructor without the need of "standing up" and provide input without the fear of public embarrassment. Both, students and teachers, can benefit from the increased interaction.

Over the past five years using TGUI we observed improvements in teaching methods and learning outcomes. We started in 2005 with test sessions in one of our courses and were overwhelmed with the positive reaction and improved experience of teaching. Therefore we expanded the application of TGUI, step by step, in all of our basic statistics courses and



Example: Central Limit Theorem (interactive graphic)

Visualization of the distribution of the sample means.

Sample size (n): 1

Draws: 100

Classes in Histogram: 16

color:

blue ●
red ○
yellow ○
transparent ○

Mark the correct claims:

□ A) For large n, the sample mean is approx. normal, independent of the distribution of the population
□ B) The approximation of the normal distribution depends on the number of classes in the histogram.
□ C) The sample means of a beta distribution converge extremly slow towards a normal distribution.

Figure 3: Interactive graph using sliders.

increased the number of interactive checks in each course. To periodically interrupt a lecture with TGUI is valuable because students can refocus and practice. Using TGUI at 20 minutes intervals proved to be the optimum – as perceived by the course participants.

3. How TGUIcan be applied: A course example

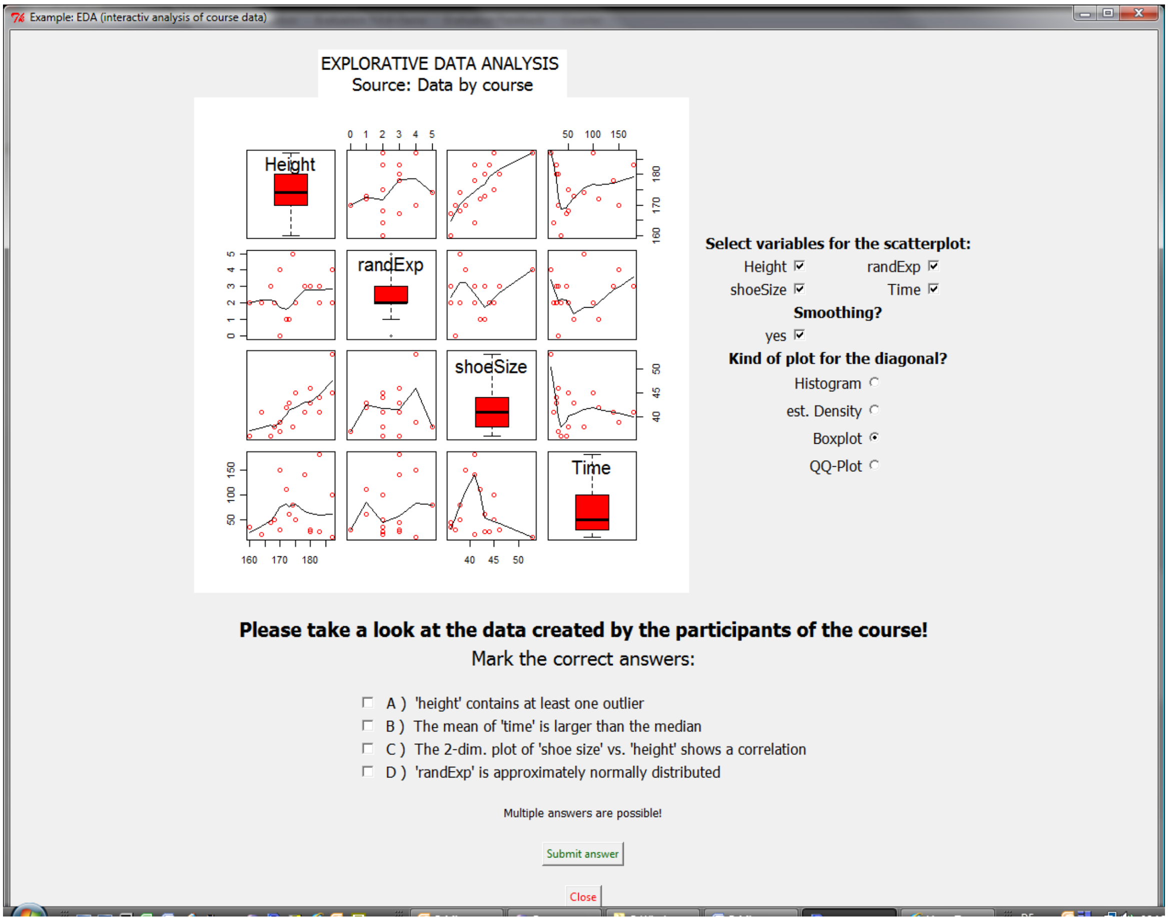
The key idea when preparing courses on basic statistics was to avoid students being lulled or inundated with knowledge by course instructors.

The goal was clear – get the entire course group actively involved in training sessions from the onset.

3.1. Data input made available using TGUI

One major step was to include input from participants directly into slides, graphics, tables and – generally speaking – in the complete course notes.

To do so, TGUI provides a mask for data input. Course participants can be asked to enter



Select variables for the scatterplot:
Height ☑ randExp ☑
shoeSize ☑ Time ☑
Smoothing?
yes ☑
Kind of plot for the diagonal?
Histogram ○
est. Density ○
Boxplot ●
QQ-Plot ○

Please take a look at the data created by the participants of the course!
Mark the correct answers:

- □ A) 'height' contains at least one outlier
- □ B) The mean of 'time' is larger than the median
- □ C) The 2-dim. plot of 'shoe size' vs. 'height' shows a correlation
- □ D) 'randExp' is approximately normally distributed

Multiple answers are possible!

Figure 4: Example using student input-data.

a nickname and course-related information such as coin flipping results, for example (see Figure 1).

With pre-written course slides it is possible to access the data entered by the course group and to generate outputs and tables based on students data *on the fly*. This is realized by using `Sweave` (Leisch 2002a,b) and LaTeX (see, e.g., Lamport 1994), which provide a possibility to dynamically link the R code in LaTeX-slides, so that R code can be embedded in a LaTeX-document. This makes this approach very flexible. For example the R package **exams** (Zeileis and Grün 2009) uses the combination of `Sweave` and LaTeX to automatically create unique exams for students.

However, the main task of TGUI is to allow the creation and realization of interactive examples in a simple manner and to support the interaction of students and lecturers.

3.2. Examples made available using TGUI

When lecturing a fundamental subject, participants should always have the opportunity to recapitulate and practice the subject matter.

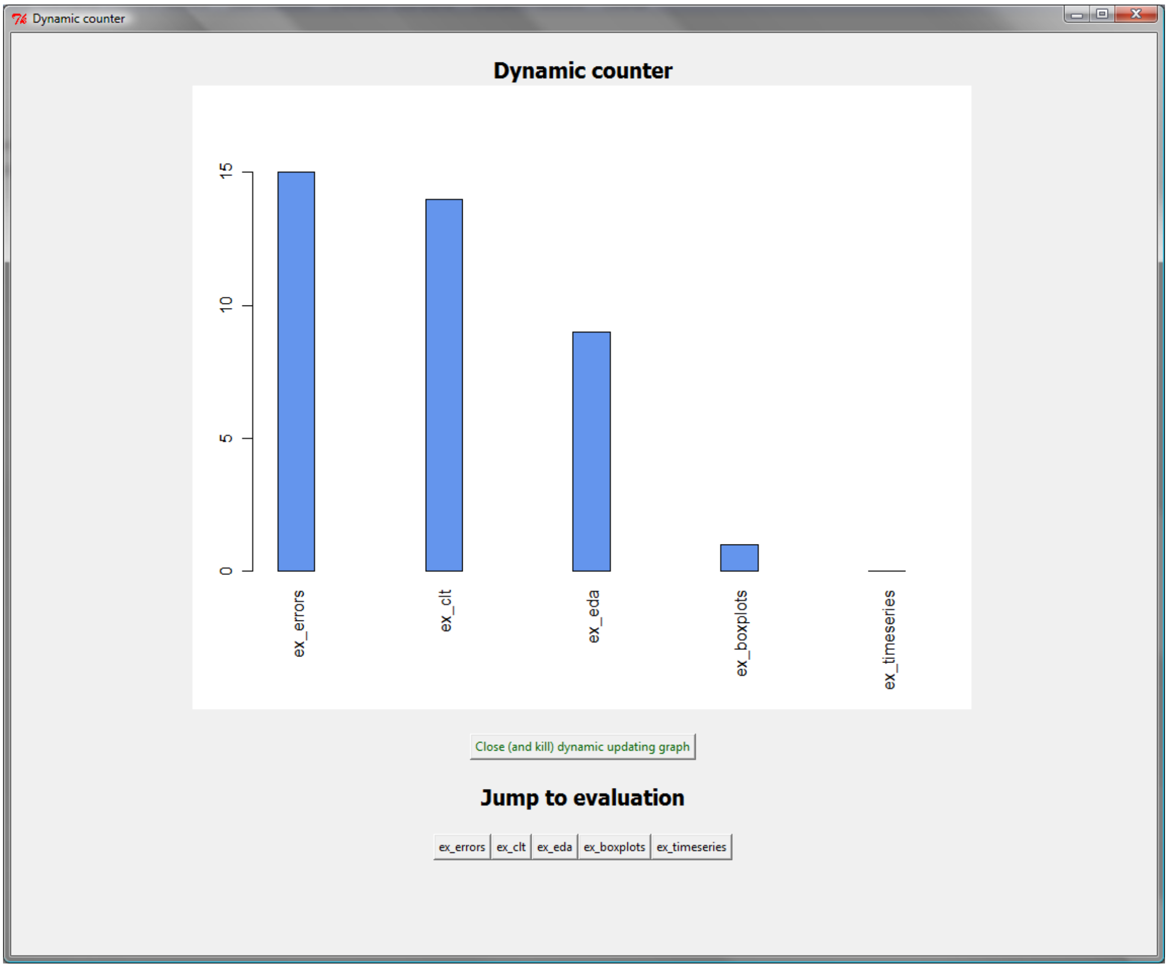


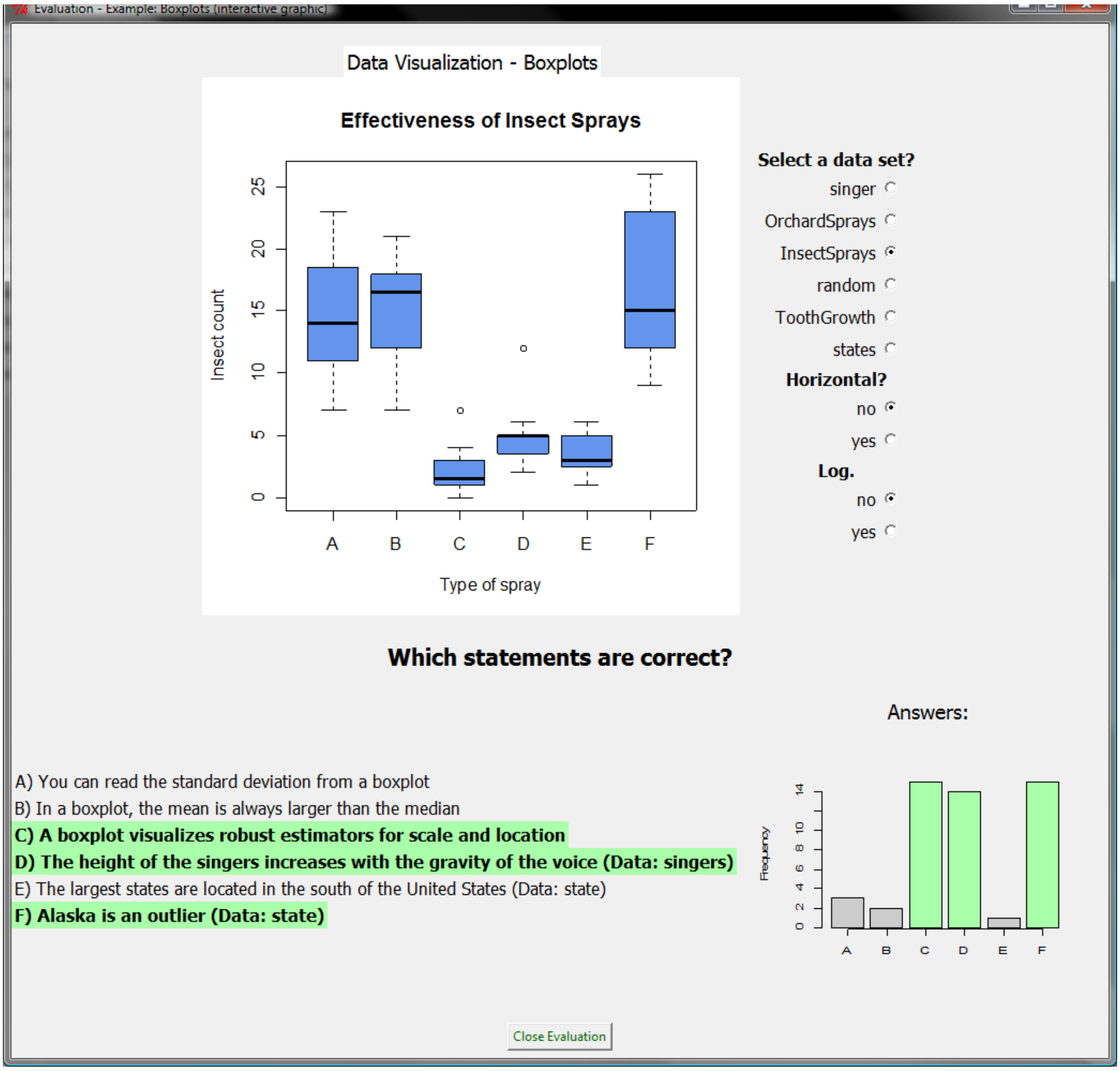
Figure 5: Dynamic counter. One notes that the height of the bar on the very left reflects that 15 course participants had already completed this check, for example, but only one student has finalized the fourth (bar, counted from the left hand side).

By using TGUI a course instructor is able to activate pre-constructed examples for a given topic during a lecture. To do so, TGUI provides two GUIs (Graphical User Interface), one for course instructors and one for students.

Students' view

Using the TGUI main menu (see Figure 2) displayed in full-screen mode, course participants can complete the activated examples by interacting with the plotted graphs using e.g., sliders – Figure 3 (allowing one to change parameters in a sampling experiment).

In Figure 4 we see an interactive exercise facilitating a better understanding of Exploratory Data Analysis, based on data entered by the students during the course (see Section 3.1). Students can manipulate the controls of the interactive graphic; thus answering multiple choice questions can be easier.


Evaluation - Example: Boxplots (interactive graphic)
Data Visualization - Boxplots
Effectiveness of Insect Sprays
Insect count
0
5
10
15
20
25
A
B
C
D
E
F
Type of spray
Answers:
Frequency
0
2
4
6
8
10
14
A
B
C
D
E
F
Close Evaluation

Select a data set?
singer ○
OrchardSprays ○
InsectSprays ●
random ○
ToothGrowth ○
states ○

Horizontal?
no ●
yes ○

Log.
no ●
yes ○

Which statements are correct?

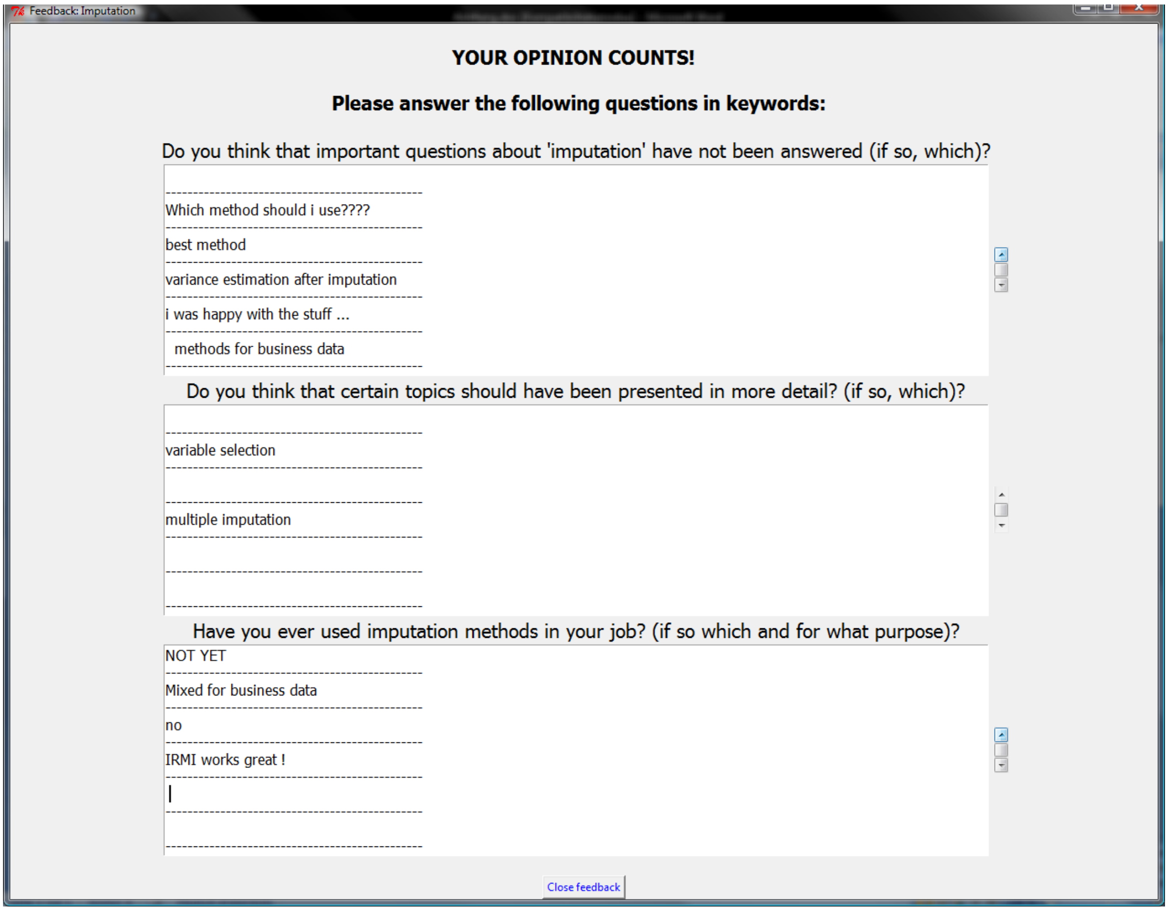
A) You can read the standard deviation from a boxplot
B) In a boxplot, the mean is always larger than the median
C) A boxplot visualizes robust estimators for scale and location
D) The height of the singers increases with the gravity of the voice (Data: singers)
E) The largest states are located in the south of the United States (Data: state)
F) Alaska is an outlier (Data: state)

Figure 6: Evaluation view with answers correctly marked and the distribution of participants answers.

Dynamic counting of finalized questions

To help the course instructors manage their time allocation per topic and allowing time for the group to finish working on given examples and to decide when the reviewing of results and examples can be started, a *dynamic counter* (shown in Figure 5) automatically runs in the background of TGUI, while the course group is working on the examples.

Once the interactive checks of a topic are completed, the course instructor can switch from *student view* to *trainer view* – Figure 6.



YOUR OPINION COUNTS!

Please answer the following questions in keywords:

Do you think that important questions about 'imputation' have not been answered (if so, which)?

Which method should i use????

best method

variance estimation after imputation

i was happy with the stuff ...

methods for business data

Do you think that certain topics should have been presented in more detail? (if so, which)?

variable selection

multiple imputation

Have you ever used imputation methods in your job? (if so which and for what purpose)?

NOT YET

Mixed for business data

no

IRMI works great !

Figure 7: Evaluation view of textual student feedback.

Trainers' view

When the students have completed the examples, the course instructor can recapitulate each example from the trainer interface of TGUI, either by displaying the *trainer view* with a beamer for the whole course group, or – ideally, circumstances permitting that the equipment exists in the classroom – he can switch the *trainer view* to individual screens. In the *trainer view*, not only the example but also the correct answer to the question (highlighted, Figure 6) as well as the distribution of the answers given by the course participants (bottom right in Figure 6) is shown. It is emphasized that especially the distribution of the answers helps the course instructor to ascertain if an example has been understood in general or if more explanation is needed on the topic.

3.3. Anonymous feedback made available using TGUI

At the end of either a topic or a course session, it is possible to ask for textual student feedback anonymous via the facility shown in Figure 7. This feedback feature is especially useful as the course instructor has an additional tool for optimizing time-management. If time is still left, the teacher may respond to the student feedback in more detail. It should be noted

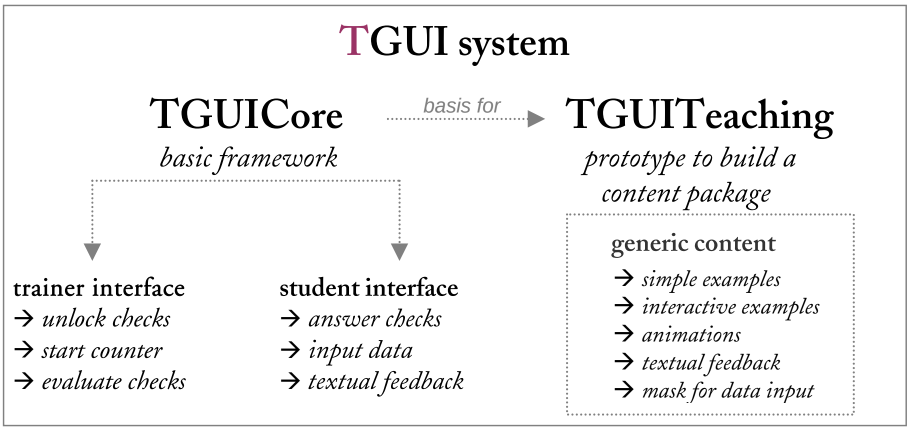


Figure 8: The TGUI system is based on the R package **TGUICore** (displayed on the left figure) which provides a trainer and a student interface. Customized content can be generated using **TGUITeaching**.

that it is even possible to pre-specify some frequently asked or important questions.

The examples shown in this section are a short excerpt of the quite extensive material created by our trainer team using the R package **TGUICore** to support basic statistics courses held at Statistics Austria. The selected excerpt should point out how TGUI can be used in practice and give an impression of its features.

The examples presented in Figure 1–7 were obtained using R 2.10.0 and **TGUICore**. They can be accessed with the freely available R package **TGUITeaching** (see Section 5), which includes all examples shown in this section along with some other generic examples. With reasonably good R skills it is quite easy to create powerful teaching examples (e.g., conducting bootstrapping or simulations) in a relatively short period of time. The R package **TeachingDemos** (Snow 2010) provides different functions which can be used to demonstrate statistical concepts supported by TGUI. However, the generic examples included in the R package **TGUITeaching** can also easily be modified, extended, changed or removed without serious programming skills (see Section 5).

4. Implementation of the R package TGUICore

So what is the R package **TGUICore**? **TGUICore** provides a student interface and a trainer interface and contains all facilities needed to easily create interactive examples and checks, or to ask for student feedback (see Figure 8 and Section 3). **TGUICore** creates the opportunity to share the possibilities of TGUI with other teachers and course instructors, because it is open source and distributed over the web. A second publicly available R package is **TGUITeaching** (presented in Section 5), which is a prototype for creating TGUI content for courses supported by TGUI. Some generic examples are included in **TGUITeaching** to show how exercises, interactive plots and animations are generated and integrated.

4.1. Software

TGUI was written using the **tcltk** framework (Hobbs, Jones, and Welch 2003) and by using R (R Development Core Team 2010a). The R package **tkrplot** (Tierney 2010) was used to create interactive graphics and animations. Furthermore, the idea for using sliders in interactive graphics and content was borrowed from the R package **TeachingDemos** (Snow 2010).

4.2. Structure

Examples supporting various courses can simply be generated by using only a few functions provided by the **TGUICore** package. For details on the functions, the corresponding help files available in the R package should be utilized (`http://CRAN.R-project.org/package=TGUICore`).

A fullscreen Tcl/Tk window for an example as shown in Figures 3, 4, 6 and 7 has to be created by calling the function `OpenWindow()`. Optionally it is possible to set a window title as well as to include an image when calling the function. Furthermore, `OpenWindow()` returns a window with an empty frame in which additional widgets can be embedded; such as the interactive scatterplot in Figure 4. Widgets that can be placed into the returned frame include single choice questions, multiple choice questions, animations and interactive graphics. It is often useful to provide varying insights in an example. It is therefore possible to show different graphics and output in the evaluation view; e.g., important data could be highlighted in a plot while discussing an example or the correct answers may be shown. One is also able to include a block to obtain instant, anonymous feedback from course participants into a specified frame or – when a training session is evaluated – the collected feedback can be shown.

In the following paragraph the authors list all the possible widgets that could be included. Please note that a detailed description of all possible input parameters for each function can be found in the R help-files in the **TGUICore** package. In order to simplify the data provided, the following functions are listed without input parameters.

- `InteractivePlot()` is used to specify interactive plots with possible specification of sliders, radio- or checkbuttons to allow course participants adapt graphics and to gain additional insight during this process. Optionally the alignment and the relative size of the graph can be adjusted.

- `SingleChoice()` is used to create single choice questions with an optional image or plot in a frame.

- `SingleChoiceGrid()` is used to includes a block of several single choice questions. It is possible (as for `SingleChoice()`) to improve the layout of the size and alignment of the widget.

- `MultipleChoice()` is applied to include multiple choice questions. Another option would be an image or a plot function can be specified which is placed above the questions and the possible answers by default. As the widgets discussed before, it is also possible to adjust the layout and the alignment of a multiple choice question block.

- The `Feedback()` function makes it possible to ask course participants for instant, anonymous textual feedback or even their opinions to pre-specified questions. Optionally, an image can be specified which is placed just above the question block.

The functions SingleChoice(), SingleChoiceGrid(), MultipleChoice() and Feedback() automatically provide the possibility to show the corresponding evaluation of the example by setting the parameter Answer to TRUE when calling the function. If Answer equals TRUE, the given feedback is shown for function Feedback() (as seen in Figure 7) while the distribution of the given answers is presented along with the correct ones for the remaining functions. All functions return a tcltk frame. These objects need to be displayed using the tcltk command tkgrid. Additional auxiliary functions (such as setFont() or showPDF()) help modify the given font, for example.

To create a font object for tcltk we use the small wrapper function setFont(), which creates a font definition for the font family defined in family with options italic and bold. The font size can be controlled by adjusting the parameter size (see the R helpfiles in the **TGUICore** package). By changing the function setFont() it is possible to change all fonts used in the TGUI.

The function showPDF() opens a PDF-File with name filename in an appropriate viewer (Acrobat Reader on Windows, shell command open under Unix-type OS).

The following functions read and write objects from specific R environments specified by **TGUICore**, TGUIenv and adminTGUIenv: Tassign(x, value), Tget(x), Texists(x), Trm(x), aTassign(x, value), aTget(x), aTexists(x), aTrm(x).

They can also be used to test if an object with name x exists in the environment as well as to remove objects. The first set of functions, without the starting "a", is used to write in an environment called TGUIenv, which is cleared, when a task is finished. The second set of functions, with the starting "a", uses the environment adminTGUIenv, which exists throughout the TGUI session and should not be cleared. In the adminTGUIenv environment, important information such as paths and global parameters are saved in order to minimize interference within the workspace.

4.3. Design

The GUI of TGUI is designed to produce a uniform appearance. Every window is full-screen, and the information needed for one example should fit on one screen. It should be mentioned that at this point the GUI is optimized for a screen resolution of 1280x1024. In fact there are two separate GUIs, one for students and one for teachers. The GUI for students includes the possibility to open all unlocked exercises, the data collection and the textual feedback feature. The GUI for teachers has additional features such as the evaluation functions. Moreover, a dynamic counter (counting participants, who have already finished an exercise) and the possibility to lock/unlock each exercise are two recently added features. During previous courses we experienced that these features are very useful.

Planned improvements

At the moment TGUI relies on the use of flat text files to store answers and additional information. When modifying a shared file on a network drive in very specific settings and conditions problems like "race conditions" and "access rights" could occur. Applying TGUI in the way we do – as a "communication switch" between teachers and students – it would not be a problem if this unlikely case occurs. Nevertheless, the next advanced TGUI version will be ameliorated by using a database based approach to store the necessary information.

5. TGUITeaching: How to create TGUI content

As starting point to create custom content, **TGUITeaching** can be used as a prototype since it consists of all necessary functions as well as some examples to build a custom package. In this section, it is shown how one can support a course with TGUI by modifiying an example from **TGUITeaching**.

5.1. Preparations

The first step is to download and install **TGUICore** and **TGUITeaching** within R. Both TGUI packages as well as R itself can be obtained from CRAN (`http://CRAN.R-project.org/`).

After the TGUI packages have been installed, **TGUITeaching** can be loaded into R with the command `library("TGUITeaching")`. Now, one can have a look at the embedded generic examples (including all the examples shown in Section 3) and may choose a suitable exercise which should be modified later.

In order to customize a generic example from **TGUITeaching** to enhance one's own course, the next step is to download **TGUITeaching** from CRAN as a source package. After extracting the source code, all the included examples can be found in the subfolder R.

5.2. How to create an exercise

Hypothetically, the course instructor chooses to modify the "Errors and Pitfalls" example and wants to change the underlying graph, the question and the possible answers. The source code of the example is

```
`ex_errors` <- function(eval = FALSE) {
  exerciseTitle <- "Exercise: errors and pitfalls"
  q1 <- "In a well known Austrian newspaper ..."
  q2 <- c("'62 PERCENT OF ALL MALES DID NOT MAKE THE GRADE!'",
    "What do you think is correct?")
  labs <- c("Headline and graph are completely correct and ...",
    "The headline matches the graph ...",
    "The headline is completely wrong ...")
  notice <- "It is possible to choose multiple answers!"
  ex <- "ex_errors"

  PiePlot <- function() {
    par(bg = "white")
    pie(c(62,38), labels = c("Male", "Female"), init.angle = 90,
      main = "Students who did not make the grade (classified by sex)",
      cex = 1.4, col = c("skyblue", "pink"), sub = "Source: www.ifas.jku.at")
    text(x = c(-0.45, 0.34), y = c(0.091, 0.20),
    label = c("62 %", "38 %"), cex = 1.8)
  }

  if(eval == FALSE) {
    OpenWindow(title = exerciseTitle)
    MultipleChoice(frame = MainFrame, err1, question1 = q1, question2 = q2,
      labels = labs, plotFunction = PiePlot, note = notice, filename = ex)
    tkgrid(err1)
  } else {
    OpenWindow(title = paste("Evaluation - ", exerciseTitle, sep = ""))
    MultipleChoice(frame = MainFrame, err1, question1 = q1, question2 = q2,
      labels = labs, plotFunction = PiePlot, filename = ex, Answer = TRUE)
    tkgrid(err1)
  }
}
```

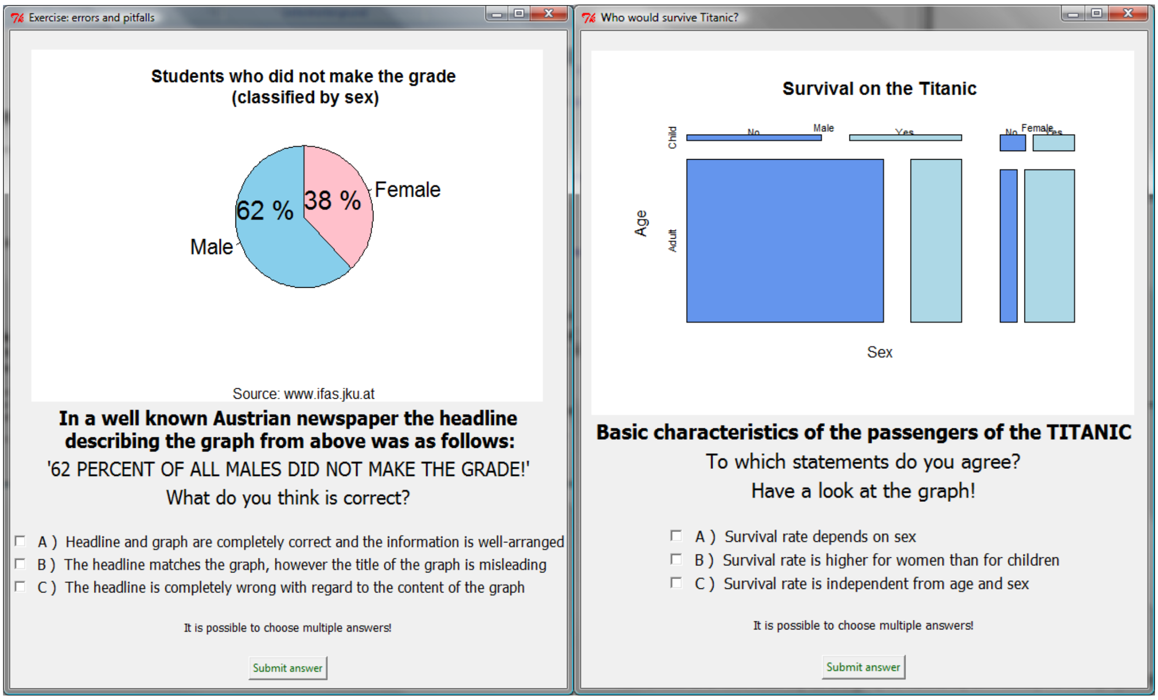
Changing the example can be done as follows. The first ten lines in the code above are adjusted by changing the function name and the desired formulation of the question along with the corresponding answers.

The `PiePlot` function given in the original example is replaced by a new one which produces the right graph for the new exercise. Finally, the name of the new plot function has to be changed in the `plotFunction` argument. The complete adjusted code is

```
`MyExercise` <- function(eval = FALSE) {
  exerciseTitle <- "Who would survive Titanic?"
  q1 <- "Basic characteristics of the passengers of the TITANIC"
  q2 <- c("To which statements do you agree?", "Have a look at the graph!")
  labs <- c("Survival rate depends on sex",
        "Survival rate is higher for woman than for children",
        "Survival rate is independent from age and sex")
  notice <- "It is possible to choose multiple answers!"
  ex <- "MyExercise"

  MosaicPlot <- function() {
    par(bg = "white")
    require("stats")
    mosaicplot(~ Sex + Age + Survived, main = "Survival on the Titanic",
      data = Titanic, color = c("cornflowerblue", "lightblue", "grey"))
    }

  if(eval == FALSE) {
    OpenWindow(title = exerciseTitle)
    MultipleChoice(frame = MainFrame, err1, question1 = q1, question2 = q2,
      labels = labs, plotFunction = MosaicPlot, note = notice, filename = ex)
    tkgrid(err1)
  } else {
    OpenWindow(title = paste("Evaluation - ", exerciseTitle, sep = ""))
    MultipleChoice(frame = MainFrame, err1, question1 = q1, question2 = q2,
      labels = labs, plotFunction = MosaicPlot, filename = ex, Answer = TRUE)
    tkgrid(err1)
  }
}
```



Exercise: errors and pifalls

Students who did not make the grade
(classified by sex)
62 %
38 %
Female
Male

Source: www.ifas.jku.at

In a well known Austrian newspaper the headline describing the graph from above was as follows:

'62 PERCENT OF ALL MALES DID NOT MAKE THE GRADE!'

What do you think is correct?

□ A) Headline and graph are completely correct and the information is well-arranged
□ B) The headline matches the graph, however the title of the graph is misleading
□ C) The headline is completely wrong with regard to the content of the graph

It is possible to choose multiple answers!

Basic characteristics of the passengers of the TITANIC

To which statements do you agree?

Have a look at the graph!

□ A) Survival rate depends on sex
□ B) Survival rate is higher for women than for children
□ C) Survival rate is independent from age and sex

It is possible to choose multiple answers!

Figure 9: Original and customized example.

course name	**name of a course chapter**	**function name**	**evaluation** no/yes (0/1)	**exercise title**	**right answer**
TGUI-Demo	TGUI-Demo (Simple Question)	ex_errors()	1	errors and pifalls	3
MyCourse	MyExercise	MyExercise()	1	survival on the Titanic	1,2

Figure 10: The six column names (in bold) of the `contents.csv` file, the content related to the question given in `ex_errors` and the new content created in `MyExercise`.

In Figure 9, the original example is shown on the left hand side and the customized example is shown on the right hand side.

5.3. How to add an exercise to the TGUImenu: The `contents.csv` file

In order to build the modified **TGUITeaching** package it is necessary to change the file `contents.csv` in the directory `inst/etc` of the package. It has to be changed according to the details of the exercises that have been altered or added. Each exercise corresponds to a line in the file. The entries of the table are shown in Figure 10.

To structure a course in the main menu, different exercises can be listed under the same main menu entry, e.g., the same course name or the same course chapter. For instance, as suggested in Section 2.3, a lecturing process could be broken up periodically with student contributions by conducting a chapter including two or three exercises.

5.4. Building modified packages

In the directory `R` where all R files should be placed, the file `zzz.R` is located. This file contains the function `.first.lib()`. This function is called when the package is loaded. The first part of the function consists of the definition of names, titles and the output directory. This part can and should be adjusted according to user needs and the local setup. For instance, `pathGUI` has to be set to the path of the shared network drive in a classroom.

As soon as all needed steps are completed, the customized package can be built from the modified source (for details on the building of packages, see R Development Core Team 2010b).

When the modified package has been installed on the classroom computers, it can be loaded from within R. The *student view* of the TGUI can then be started by the command `runGUI()`. The *trainer view* can be started with `adminGUI()`.

6. Conclusion

The TGUI system, which we introduced in this paper, is designed to enhance and support modern teaching methods by facilitating interaction between teachers and students. While blended-learning systems are often expensive, TGUI is free. Furthermore the source code is publicly available. Once implemented, the TGUI system is relatively user friendly and an in depth method, a means of understanding on a lectured topic is achieved. With the inclusion of all fundamental ideas described by the authors, this open source approach in modern teaching goes a step beyond common personal response systems (see, e.g., d'Inverno, Davis, and White 2003; Elliott 2003) and many typical blended-learning systems (see, e.g., Graham and Bonk 2004).

TGUI is applicable to courses where personal computers have access to a network. However, with R as backend, TGUI is especially useful for teaching statistical content. Since enterprises and universities often have the possibility to use computer labs for training, TGUI may be applied in many fields and situations. How invaluable TGUI is will depend on course instructors skills and ingenuity to create entertaining but instructive examples for supporting a lecture topic. An emphasis is placed on asking good questions.

TGUI was developed as a tool to support Statistik Austria's in-house employee-training. There arose a need for understandable and enjoyable ways to teach statistical topics. The goal was therefore to have an environment that – once implemented – supported and enriched a course with user-friendly facilities, without the pre-requisite of software skills on the part of the course participants. There was no need to implement features such as "holding exams with TGUI," "facilitating the package building process for non-R users" or "optimizing TGUI for Linux and OS X." However looking into the future – and given that demand and resources exist – the authors would consider generalizing the system to function in other network environments and facilitating the TGUI content building process for non-R users (e.g., with a GUI for the package creator).

The motivation for the introduction of TGUI was to share the advantages of TGUI. With this in mind the R package **TGUICore**, which supplies all necessary functions to create interactive checks and collect responses to present them to the whole course group, as well as the R package **TGUITeaching**, providing guidance to building one's own R packages, are freely available for download at CRAN.

Acknowledgments

The authors would like to thank two anonymous reviewers and editor Jan de Leeuw for their valuable suggestions and input in improving this paper, and their constructive comments on the code. A special thanks goes to Maria Lambrakis for the very helpful support to improve the manuscript. The authors are also grateful to Statistics Austria and its management, who gave them the opportunity to break new ground and modernize in teaching statistics. And most importantly, to our students and course participants we say *merci*, for the positive initial feedback and constructive support that finally encouraged us to enhance the teaching system TGUI.

References

Blakesley Lindsay E (2004). "The Best of Both Worlds: Teaching a Hybrid Course." *Academic Exchange Quarterly*, **8**(4).

Brown DG (2001). "Hybrid Courses Are Best." *Syllabus: New Dimensions in Education Technology*, **15**(1), 22.

Cline KS (2006). "Classroom Voting in Mathematics." *Mathematics Teacher*, **100**(2), 100.

Cline KS, Lomen D (2010). "Classroom Voting: Active Learning in Differential Equations." *CODEE Journal*. URL http://www.codee.org/ref/CJ09-0158.

Da Rin D (2003). *Vom E-Learning zum Blended Learning*. Ph.D. thesis, Universität Luzern. URL http://edoc.zhbluzern.ch/unilu/ediss/unilu_diss_2005_001_darin_fulltext.pdf.

Dinges G, Templ M (2009). "Motivation zur Statistik – Computergestützt Lernen in der Statistik Austria." *Austrian Journal of Statistics*, **38**(1), 3–16. URL http://www.stat.tugraz.at/AJS/ausg091/091DingesTempl.pdf.

d'Inverno R, Davis H, White S (2003). "Using a Personal Response System for Promoting Student Interaction." *Teaching Mathematics and its Applications*, **22**(4), 163–169.

Elliott C (2003). "Using a Personal Response System in Economics Teaching." *International Review of Economics Education*, **1**(1). URL http://www.economicsnetwork.ac.uk/iree/i1/elliott.htm.

Graham CR, Bonk CJ (2004). *Handbook of Blended Learning: Global Perspectives, Local Designs*. Pfeiffer Publishing.

Hobbs J, Jones K, Welch BB (2003). *Practical Programming in Tcl and Tk*. 4th edition. Prentice Hall PTR, New York.

Lamport L (1994). *LaTeX: A Document Preparation System: User's Guide and Reference*. 2nd edition. Addison-Wesley, Reading, Massachusetts.

Leisch F (2002a). "Dynamic Generation of Statistical Reports Using Literate Data Analysis." In W Härdle, B Rönz (eds.), *COMPSTAT 2002 – Proceedings in Computational Statistics*, pp. 575–580. Physica-Verlag, Heidelberg.

Leisch F (2002b). "Sweave, Part I: Mixing R and LaTeX." *R News*, **2**(3), 28–31. URL http://CRAN.R-project.org/doc/Rnews/.

R Development Core Team (2010a). *An Introduction to R: A Programming Environment for Data Analysis and Graphics*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-12-7, URL http://www.R-project.org/.

R Development Core Team (2010b). *Writing R Extensions*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Silberman M (1996). *Active Learning: 101 Strategies to Teach Any Subject*. Allyn & Bacon.

Snow G (2010). ***TeachingDemos**: Demonstrations for Teaching and Learning*. R package version 2.7, URL http://CRAN.R-project.org/package=TeachingDemos.

Tierney L (2010). ***tkrplot**: Tk R Plot*. R package version 0.0-19, URL http://CRAN.R-project.org/package=tkrplot.

Ward B (2004). "The Best of Both Worlds: A Hybrid Statistics Course." *Journal of Statistics Education*, **12**(3). URL http://www.amstat.org/publications/jse/v12n3/ward.html.

Zeileis A, Grün B (2009). "Automatic Generation of Exams in R." *Journal of Statistical Software*, **29**(10), 1–14. URL http://www.jstatsoft.org/v29/i10/.

Zullo H, Cline K (2010). "MathQUEST/MathVote: Resources for Clickers and Classroom Voting in Collegiate Mathematics." URL http://mathquest.carroll.edu/.

Affiliation:

Gerlinde Dinges, Alexander Kowarik, Bernhard Meindl, Matthias Templ
Department of Methodology
Registers, Classifications and Methods Division
Statistics Austria
Guglgasse 13
1110 Vienna, Austria
E-mail: gerlinde.dinges@statistik.gv.at, alexander.kowarik@statistik.gv.at
bernhard.meindl@statistik.gv.at, matthias.templ@statistik.gv.at

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 39, Issue 7 *Submitted:* 2009-11-19
March 2011 *Accepted:* 2010-04-23