Matthias Templ

Institute of Data Analysis and Process Design, ZHAW

Statistics at Universities of Applied Sciences (UAS)
Olten, January 26, 2017

Game of Lazybones
Computer-Assisted Exercises with Automatised Feedback and Evaluation

The teaching- and feedback-system

tguishiny
Popularity of blended learning

- Increasing importance of “Statistics in education and training”
- Increasing interest in blended and digital learning

<table>
<thead>
<tr>
<th>Sub-Classification</th>
<th>Year Range</th>
<th>Number of articles</th>
</tr>
</thead>
<tbody>
<tr>
<td>First attempts</td>
<td>1999-2002</td>
<td>125</td>
</tr>
<tr>
<td>Definition period</td>
<td>2003-2006</td>
<td>1200</td>
</tr>
<tr>
<td>Popularity period</td>
<td>2007-2009</td>
<td>1460</td>
</tr>
<tr>
<td>Recently</td>
<td>2010-2012</td>
<td>1660</td>
</tr>
</tbody>
</table>

from Güzera and Caner (2014)

A lot of tools are available, but do they fit to our needs?
Popularity of blended learning

- Increasing importance of “Statistics in education and training”
- Increasing interest in blended and digital learning

<table>
<thead>
<tr>
<th>Sub-Classification</th>
<th>Year Range</th>
<th>Number of articles</th>
</tr>
</thead>
<tbody>
<tr>
<td>First attempts</td>
<td>1999-2002</td>
<td>125</td>
</tr>
<tr>
<td>Definition period</td>
<td>2003-2006</td>
<td>1200</td>
</tr>
<tr>
<td>Popularity period</td>
<td>2007-2009</td>
<td>1460</td>
</tr>
<tr>
<td>Recently</td>
<td>2010-2012</td>
<td>1660</td>
</tr>
</tbody>
</table>

from Güzera and Caner (2014)

A lot of tools are available, but do they fit to our needs?
Introduction

We extend existing concepts.

▶ **New possibilities** and **new ideas** make it possible to improve these systems in order to maximize the effectiveness of teaching.

▶ The aims is both,
  ▶ a systematic **implementation** of teaching concepts in software using **modern interactive tools** and
  ▶ to invent **new ways** to teach

Content: the **tguishiny**, a digital learning tool developed by us for feedback-based interactive teaching
Introduction

We extend existing concepts.

- **New possibilities** and **new ideas** make it possible to improve these systems in order to maximize the effectiveness of teaching.

- The aims is both,
  - a systematic **implementation** of teaching concepts in software using **modern interactive tools** and
  - to invent **new ways** to teach

**Content:** the **tguishiny**, a digital learning tool developed by us for feedback-based interactive teaching
Elements of computer-assisted feedback learning

**Exercises:** Allows to repeat and present topics and to execute exercises

**Visualisation:** Interactive graphics and animations to understand complex topics and relationships

**Integration of student surveys:** Questionnaires integrated - resulting data can be integrated into exercises

**Interactive feedback:** Client to server - everything done by the students, saved in a data base and summaries are presented to the teacher

**Gaming:** Interactive features supports a gaming character and rewards are also possible

**Monitoring:** Which student has finished which exercise? Forces lazybones to be active during the class.
Elements of computer-assisted feedback learning

**Exercises:** Allows to repeat and present topics and to execute exercises

**Visualisation:** Interactive graphics and animations to understand complex topics and relationships

**Integration of student surveys:** Questionnaires integrated - resulting data can be integrated into exercises

**Interactive feedback:** Client to server - everything done by the students, saved in a data base and summaries are presented to the teacher

**Gaming:** Interactive features supports a gaming character and rewards are also possible

**Monitoring:** Which student has finished which exercise? Forces lazybones to be active during the class.
Elements of computer-assisted feedback learning

**Exercises:** Allows to repeat and present topics and to execute exercises

**Visualisation:** Interactive graphics and animations to understand complex topics and relationships

**Integration of student surveys:** Questionnaires integrated - resulting data can be integrated into exercises

**Interactive feedback:** Client to server - everything done by the students, saved in a data base and summaries are presented to the teacher

**Gaming:** Interactive features supports a gaming character and rewards are also possible

**Monitoring:** Which student has finished which exercise? Forces lazybones to be active during the class.
Elements of computer-assisted feedback learning

**Exercises**: Allows to repeat and present topics and to execute exercises

**Visualisation**: Interactive graphics and animations to understand complex topics and relationships

**Integration of student surveys**: Questionnaires integrated - resulting data can be integrated into exercises

**Interactive feedback**: Client to server - everything done by the students, saved in a data base and summaries are presented to the teacher

**Gaming**: Interactive features supports a gaming character and rewards are also possible

**Monitoring**: Which student has finished which exercise? Forces lazybones to be active during the class.
**Exercises**: Allows to repeat and present topics and to execute exercises

**Visualisation**: Interactive graphics and animations to understand complex topics and relationships

**Integration of student surveys**: Questionnaires integrated - resulting data can be integrated into exercises

**Interactive feedback**: Client to server - everything done by the students, saved in a data base and summaries are presented to the teacher

**Gaming**: Interactive features supports a gaming character and rewards are also possible

**Monitoring**: Which student has finished which exercise? Forces lazybones to be active during the class.
Elements of computer-assisted feedback learning

**Exercises**: Allows to repeat and present topics and to execute exercises

**Visualisation**: Interactive graphics and animations to understand complex topics and relationships

**Integration of student surveys**: Questionnaires integrated - resulting data can be integrated into exercises

**Interactive feedback**: Client to server - everything done by the students, saved in a data base and summaries are presented to the teacher

**Gaming**: Interactive features supports a gaming character and rewards are also possible

**Monitoring**: Which student has finished which exercise? Forces lazybones to be active during the class.
Elements of computer-assisted feedback learning

**Exercises:** Allows to repeat and present topics and to execute exercises

**Visualisation:** Interactive graphics and animations to understand complex topics and relationships

**Integration of student surveys:** Questionnaires integrated - resulting data can be integrated into exercises

**Interactive feedback:** Client to server - everything done by the students, saved in a data base and summaries are presented to the teacher

**Gaming:** Interactive features supports a gaming character and rewards are also possible

**Monitoring:** Which student has finished which exercise? Forces lazybones to be active during the class.
This is cannot happen with tguishiny. Students are usually motivated to use tguishiny, but if not - we can track their activity in solving exercises in real-time.
Monitoring

This is cannot happen with *tguishiny*. Students are usually motivated to use *tguishiny* but if not - we can track their activity in solving exercises in real-time.

Matthias Templ (IDP)
Technical implementation - History

Teaching with TGUI (Trainings GUI) and developments

- **from 2005 to 2009** first tcl/tk version used at all courses in Statistics Austria

- **from 2009 to 2011** the system is provided via R packages (GUI in Gtk2)
  - Info: [http://www.jstatsoft.org/v39/i07](http://www.jstatsoft.org/v39/i07) (Dinges, Kowarik, Meindl, and Templ 2011b)

- **from 2011 to 2017** online version via RApache, demo version TGUI\textsubscript{online} as showcase online
  - Info: [http://www.statistik.at/TguiOnline](http://www.statistik.at/TguiOnline) (Dinges, Kowarik, Meindl, and Templ 2011a)

- **from 2017 to 2018** a new project from ZHAW/SoE helps to modernize the teaching environment and to implement new ideas.
The new implementation: **tguishiny**

### Technical issues

- **R** package **tguishiny**.
- Strictly objekt-orientierted programming of exercise types (using R6 reference classes and R modules)
- Web-application with **R** package **shiny**. Advantages: easy to write web-applications, \( \LaTeX \) (over MathJax), JavaScript, HTML, d3, R, markdown, ... can be used.
- Installations on a server or locally on the PC
Basic features of the system

- **Surveys:** possible to collect information about the course participants through questionnaires and integration of the results into exercises
- **(interactive) Exercises:** different kind of exercise classes (next slide)
- **Feedback-tool:** tracks and stores all activities from the students (mouse clicks, answers, R code, ...) to predefined tasks
- **Evaluation:** e.g. visualization of the distribution of answers for exercises
- **Dynamic counter:** for each unlocked exercise it counts, how many students have solved which exercises
- **R:** integration of R for exercises with R
Basic features of the system

- **Surveys**: possible to collect information about the course participants through questionnaires and integration of the results into exercises

- **(interactive) Exercises**: different kind of exercise classes (next slide)

- **Feedback-tool**: tracks and stores all activities from the students (mouse clicks, answers, R code, ...) to predefined tasks

- **Evaluation**: e.g. visualization of the distribution of answers for exercises

- **Dynamic counter**: for each unlocked exercise it counts, how many students have solved which exercises

- **R**: integration of R for exercises with R
Basic features of the system

- **Surveys**: possible to collect information about the course participants through questionnaires and integration of the results into exercises
- **(interactive) Exercises**: different kind of exercise classes (next slide)
- **Feedback-tool**: tracks and stores all activities from the students (mouse clicks, answers, R code, ...) to predefined tasks
- **Evaluation**: e.g. visualization of the distribution of answers for exercises
- **Dynamic counter**: for each unlocked exercise it counts, how many students have solved which exercises
- **R**: integration of R for exercises with R
Basic features of the system

- **Surveys:** possible to collect information about the course participants through questionnaires and integration of the results into exercises
- **(interactive) Exercises:** different kind of exercise classes (next slide)
- **Feedback-tool:** tracks and stores all activities from the students (mouse clicks, answers, R code, ...) to predefined tasks
- **Evaluation:** e.g. visualization of the distribution of answers for exercises
- **Dynamic counter:** for each unlocked exercise it counts, how many students have solved which exercises
- **R:** integration of R for exercises with R
Basic features of the system

- **Surveys:** possible to collect information about the course participants through questionnaires and integration of the results into exercises
- **(interactive) Exercises:** different kind of exercise classes (next slide)
- **Feedback-tool:** tracks and stores all activities from the students (mouse clicks, answers, R code, ...) to predefined tasks
- **Evaluation:** e.g. visualization of the distribution of answers for exercises
- **Dynamic counter:** for each unlocked exercise it counts, how many students have solved which exercises
- **R:** integration of R for exercises with R
Basic features of the system

- **Surveys**: possible to collect information about the course participants through questionnaires and integration of the results into exercises
- **(interactive) Exercises**: different kind of exercise classes (next slide)
- **Feedback-tool**: tracks and stores all activities from the students (mouse clicks, answers, R code, ...) to predefined tasks
- **Evaluation**: e.g. visualization of the distribution of answers for exercises
- **Dynamic counter**: for each unlocked exercise it counts, how many students have solved which exercises
- **R**: integration of R for exercises with R
## Important question classes for exercises

<table>
<thead>
<tr>
<th>question class</th>
<th>details</th>
<th>user task</th>
<th>evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>(ShinyQuestion)</td>
<td>(parent class)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>McQuestion</td>
<td>multiple choice</td>
<td>choice of answer(s)</td>
<td>distribution of given answers</td>
</tr>
<tr>
<td>RQuestion</td>
<td>R exercises</td>
<td>R code</td>
<td>evaluation of resulting R object</td>
</tr>
<tr>
<td>DfQuestion</td>
<td>data manipulation tasks</td>
<td>R code</td>
<td>evaluation of resulting object or code lines</td>
</tr>
<tr>
<td>PlotQuestion</td>
<td>plotting task</td>
<td>produce a plot</td>
<td>evaluation of plot</td>
</tr>
<tr>
<td>LmQuestion</td>
<td>exercises for linear models</td>
<td>estimating a linear model</td>
<td>evaluation of the resulting object</td>
</tr>
<tr>
<td>MarkdownQuestion</td>
<td>combines previous question classes</td>
<td>depends on class</td>
<td>depends on question class</td>
</tr>
</tbody>
</table>
# Important question classes for exercises

<table>
<thead>
<tr>
<th>question class</th>
<th>details</th>
<th>user task</th>
<th>evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>(ShinyQuestion)</td>
<td>(parent class)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>McQuestion</td>
<td>multiple choice</td>
<td>choice of answer(s)</td>
<td>distribution of given answers</td>
</tr>
<tr>
<td>RQuestion</td>
<td>R exercises</td>
<td>R code</td>
<td>evaluation of resulting R object</td>
</tr>
<tr>
<td>DfQuestion</td>
<td>data manipulation tasks</td>
<td>R code</td>
<td>evaluation of resulting object or code lines</td>
</tr>
<tr>
<td>PlotQuestion</td>
<td>plotting task</td>
<td>produce a plot</td>
<td>evaluation of plot</td>
</tr>
<tr>
<td>LmQuestion</td>
<td>exercises for linear models</td>
<td>estimating a linear model</td>
<td>evaluation of the resulting object</td>
</tr>
<tr>
<td>MarkdownQuestion</td>
<td>combines previous question classes</td>
<td>depends on class</td>
<td>depends on question class</td>
</tr>
</tbody>
</table>

Matthias Templ (IDP)
A brief demonstration of **tguishiny**

Some notes:

- **tguishiny** also runs on ZHAW and Statistics Austria servers, whereby teachers and students have access and everybody make the exercises on the given server.
- Here we show the local version.
- The server version has severeral benefits, e.g. full control of the R installation.
- To run smoothly on a server, RStudio’s shiny server must be installed.
- For using it in the class, always the server version is used.
Key features of tguishiny

▶ **Students view:**
  ▶ view on exercises unlocked by the teacher
  ▶ summary statistics: *my* performance in comparison to other students (work-in-progress)

▶ **Teachers view:**
  ▶ Lock/unlock of exercises or questionaires
  ▶ Link to evaluations for each exercise
  ▶ Counter that shows the progress of the group (how many students have completed which exercise)

▶ **Under the hood:**
  ▶ collection of all actions from students in a data base
  ▶ any evaluation is thus supported, even gamification and any statistics presented to students and teachers possible
  ▶ user management and access rights
  ▶ could be in principle be used also for automated exams (and correction)
Key features of **tguishiny**

- **Students view:**
  - view on exercises unlocked by the teacher
  - summary statistics: *my* performance in comparison to other students *(work-in-progress)*

- **Teachers view:**
  - Lock/unlock of exercises or questionnaires
  - Link to evaluations for each exercise
  - Counter that shows the progress of the group (how many students have completed which exercise)

- **Under the hood:**
  - collection of all actions from students in a data base
  - any evaluation is thus supported, even gamification and any statistics presented to students and teachers possible
  - user management and access rights
  - could be in principle be used also for automated exams (and correction)
Key features of tguishiny

- **Students view:**
  - view on exercises unlocked by the teacher
  - summary statistics: *my* performance in comparison to other students (work-in-progress)

- **Teachers view:**
  - Lock/unlock of exercises or questionnaires
  - Link to evaluations for each exercise
  - Counter that shows the progress of the group (how many students have completed which exercise)

- **Under the hood:**
  - collection of all actions from students in a database
  - any evaluation is thus supported, even gamification and any statistics presented to students and teachers possible
  - user management and access rights
  - could be in principle be used also for automated exams (and correction)
Writing your own exercises

Core system: approx. 8000 lines of efficient R code in order

- to easily create new exercises
- to automatically bind them to the web-interface
- to provide questionnaires and feedback
- to deal with multi-user issues
- to integrate R for related questions
Writing your own exercises

- With the current version, one needs to have minimal R knowledge.
- In future versions this should be possible online by point-and-click and simple text input without any R knowledge

First we want to start R and the tguishiny package. Each question type is documented.

```r
library("tguishiny")
?ShinyQuestion
?McQuestion
?RQuestion
?MarkdownQuestion
...
```

Hint: Use show_interactive() to test new questions
Writing your own exercises

- With the current version, one needs to have minimal \texttt{R} knowledge.
- In future versions this should be possible online by point-and-click and simple text input without any R knowledge.

First we want to start \texttt{R} and the \texttt{tguishiny} package. Each question type is documented.

```
library("tguishiny")
?ShinyQuestion
?McQuestion
?RQuestion
?MarkdownQuestion
...
```

Hint: Use \texttt{show\_interactive()} to test new questions.
After writing questions

tguiApp(questions = 'path/to/questions',
        db_path = 'path/to/database.db')

By default the working path is used. You can place questions and
databases separated from tguishiny.

# questions, exercises and database within tguishiny:
tguiApp()

# ZHAW course:
tguiApp(questions = '..',tguicoursesrepo/ZHAW/"

# Statistics Austria course ST03:
tguiApp(questions = '..',tguicoursesrepo/ST03/"

...
After writing questions

tguiApp(questions = 'path/to/questions',
       db_path = 'path/to/database.db')

By default the working path is used. You can place questions and databases separated from tguishiny.

# questions, exercises and data base within tguishiny:
tguiApp()

# ZHAW course:
tguiApp(questions = "../tguicoursesrepo/ZHAW/")

# Statistics Austria course ST03:
tguiApp(questions = "../tguicoursesrepo/ST03/")
...

Matthias Templ (IDP)
Experience with TGUI online and tguishiny

- Forced automatised feedback (from everybody) essential for classes > 8-10 students
- Linking individual data from students with exercises makes students more interested
- Dynamic counter essential
- Through on-the-fly evaluations, the teacher has full control if students have understood the topics
- Students have been very positive about the tool
Summary & outlook

Actual situation

▶ basic **programming** of **tguishiny** is more or less done
▶ system is running on server and local

Things to be done with additional funds

▶ **tguishiny** can be in principle used for automated exams, but some security issues must be solved
▶ any kind of gamification can be implemented, because all necessary data are stored. If so, **tguishiny** can be relatively straightforward re-written using dashboards to present figures, smileys, statistics to students permanently.
▶ more complex evaluations, comparison of students, Rasch models to evaluate the difficulty of questions, etc.
Summary & outlook

Actual situation

- basic **programming** of **tguishiny** is more or less done
- system is running on server and local

Things to be done with additional funds

- **tguishiny** can be in principle used for automated exams, but some security issues must be solved
- any kind of gamification can be implemented, because all necessary data are stored. If so, **tguishiny** can be relatively straightforward re-written using dashboards to present figures, smileys, statistics to students permanently.
- more complex evaluations, comparison of students, Rasch models to evaluate the difficulty of questions, etc.
Teaching interactively
with the teaching and feedback system
tguishiny

▷ Many thanks to SoE Lehre (ZHAW) for the grant “Digitale Lehrformen”

▷ Many thanks to my students Gregor De Cillia (TU Wien) for his excellent contribution to the R code, Tamara Ganz and Stevan Ljubomirovic (ZHAW) for transferring many examples to tguishiny. Thanks to Bernhard Meindl (Statistics Austria) for helpful discussions and contributions.

your FEEDBACK is not forced but welcome

