# Importing z-Tree data into R

Oliver Kirchkamp[*]

25th November 2018

The software z-Tree is used for thousands of economic experiments worldwide. z-Tree stores experimental data in a way that minimises the risk of losing data. However, it may be cumbersome to manually read this data into a statistical package. The purpose of the R-package `zTree` is to make the process of importing data from z-Tree into the statistical package R transparent, reproducible and simple.

Keywords: z-Tree. Reproducible research.
JEL: C55, C81, C91.

## 1 Introduction

The software z-Tree (Fischbacher, 2007) is used for several thousands of economic experiments worldwide. The R-package `zTree` imports the raw data from z-Tree directly into the statistical software R. z-Tree stores its raw data in a way that minimises the risk of losing data. z-Tree also provides functions that, with a few mouse-clicks for each session, translate z-Tree data into several files that are more accessible to statistical software. However, these steps have to be carried out again and again for each file and for each session. The more of these manual steps must be carried out, the more likely it is that files are forgotten or misplaced. Furthermore, the more manual steps the researcher has to carry out, the more cumbersome the documentation of methods for reproducible research becomes (Koenker and Zeileis, 2009).

It is the aim of the R-package `zTree` to make importing raw data from z-Tree into R transparent, reproducible and simple.

## 2 The package

### 2.1 z-Tree-files

Although z-Tree uses the `.xls` extension for its files, these files do not at all follow the structure of a Microsoft Excel file. Instead, z-Tree writes a sequence of tab delimited tables into one single file. In each period of a session of an experiment several tables are appended to the same

[*] University Jena, School of Economics, Carl-Zeiss-Str. 3, 07737 Jena, oliver@kirchkamp.de, +49 3641 943240.

file. Since each of these tables contain a different number of columns and since, furthermore, different columns in different tables refer to different variables, reading the file with standard methods (such as `read.csv` in R) fails.

The R-package `zTree` provides commands to transparently read these tables into R, separate the different tables and collect data from different sessions. For Stata, Kan Takeuchi has written an `ado` file to read data from z-Tree.[1]

## 2.2 Features of the package

The R-package `zTree` provides functions to import data for several sessions of an experiment with a single command. The resulting object (a list) contains all selected tables from all sessions.

The package also reads manipulated files (i.e. files that have already been opened in a spreadsheet program, changed, and saved back to disk), although with a warning.[2]

## 2.3 Examples

We load the package and obtain a vector of filenames. Here we assume that there are two `.xls` files in the current directory.

```
library(zTree)
( myFilenames <- list.files(pattern="*.xls") )
##  [1] "160215_0810.xls" "160215_0949.xls"
```

Now `myFilenames` is a vector of filenames. Each name refers to one experimental session (here we have only two sessions). We read all sessions with the following command:

```
allData <- zTreeTables(myFilenames)
##  reading  160215_0810.xls ...
##  Skipping: summary
##  Doing: globals
##  Doing: subjects
##  *** 160215_0949.xls is file 2 / 2 ***
##  reading  160215_0949.xls ...
##  Skipping: summary
##  Doing: globals
##  Doing: subjects
```

The option `zTree.silent` can be used to reduce verbosity. By default `zTreeTables` reads only `globals` and `subjects`. The parameter `tables` can be used to select any other tables.

In our example `allData` contains (as a list of data frames) all desired tables from all sessions of the experiment. Here is a subset of the `globals` tables:

[1] `https://sites.google.com/view/takekan/research/ztree2stata`

[2] The spreadsheet editor does not remember the underlying table structure and saves all z-Tree tables as if they had the same number of columns.

```
allData[["globals"]][,1:5]
```

| Date | Treatment | Period | NumPeriods | RepeatTreatment |
|---|---:|---:|---:|---:|
| 160215_0810 | 1 | 1 | 1 | 0 |
| 160215_0949 | 1 | 1 | 1 | 0 |

And here is a subset of the `subjects` tables:

```
allData[["subjects"]][1:4,1:7]
```

| Date | Treatment | Period | Subject | Group | Profit | TotalProfit |
|---|---:|---:|---:|---:|---:|---:|
| 160215_0810 | 1 | 1 | 1 | 1 | 6.00 | 6.00 |
| 160215_0810 | 1 | 1 | 2 | 1 | 10.00 | 10.00 |
| 160215_0810 | 1 | 1 | 3 | 1 | 8.00 | 8.00 |
| 160215_0810 | 1 | 1 | 4 | 1 | 7.50 | 7.50 |

**Importing questionnaires** z-Tree stores information from the questionnaire at the end of the experiment in a table with the extension `.sbj`.

```
( myQuestFiles <- list.files(pattern="*.sbj") )
##  [1] "160215_0810.sbj" "160215_0949.sbj"
```

Now `myQuestFiles` is a vector of filenames. Each name refers to the subjects from one session of the experiment (here we have only two sessions). While the `.xls` table stores subject specific information as one line per subject, the `.sbj` table uses one column per subject. The function `zTreeSbj` reads a vector of files and transposes them to one line per subject. We read subjects from all sessions with the following command:

```
allQuest <- zTreeSbj(myQuestFiles)
##  reading  160215_0810.sbj ...
##  reading  160215_0949.sbj ...
```

The data frame `allQuest` contains all subjects from all experiments. Here is a subset:

```
allQuest[1:4,1:5]
```

| Date | Subject | client | Age | Gender |
|---|---:|---|---:|---|
| 160215_0810 | 1 | client01 | 25 | male |
| 160215_0810 | 2 | client02 | 24 | female |
| 160215_0810 | 3 | client03 | 23 | female |
| 160215_0810 | 4 | client04 | 23 | female |

We can easily merge data from the questionnaire with the `subjects` table:

```
subjectsJoined <- merge(allData[["subjects"]],allQuest)
```

Here is a subset:

```
subjectsJoined[1:4,c("Date","Subject","Group","Profit","Age","Gender")]
```

| Date | Subject | Group | Profit | Age | Gender |
|---|---:|---:|---:|---:|---|
| 160215_0810 | 1 | 1 | 6.00 | 25 | male |
| 160215_0810 | 2 | 1 | 10.00 | 24 | female |
| 160215_0810 | 3 | 1 | 8.00 | 23 | female |
| 160215_0810 | 4 | 1 | 7.50 | 23 | female |

## 3 Summary

The package will hopefully simplify the work of researchers who deal with a larger number of sessions of a z-Tree experiment. Perhaps more importantly, the package should simplify the documentation of the analysis. Clearly, the package solves a problem that could as well be solved with a number of mouse-clicks followed by a number of import and merge commands. Documenting mouse-clicks is tricky. Documenting one line of code should be easier. The R-package `zTree` is available at CRAN: `https://cran.r-project.org/package=zTree` and free to use under GPLv3.

## References

Fischbacher, U. (2007). z-tree: Zurich toolbox for ready-made economic experiments. *Experimental Economics*, 10(2):171–178.

Koenker, R. and Zeileis, A. (2009). On reproducible econometric research. *Journal of Applied Econometrics*, 24(5):833–847.

R Development Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.