

Analyzing the z-Tree Tutorial results in *R*

Stefan Palan

1 Data preparation

1. Clear *R*'s memory and load Oliver Kirchkamp's *z-Tree* to *R* import function:

```
# Initial cleanup
rm(list = ls()) #Clears memory

# Loads Oliver Kirchkamp's import function
install.packages("ztree") #Installs package
library("ztree") #Loads package
```

2. Read in z-Tree data and display the result:

```
# Imports z-Tree data
Data <- zTreeTables("151024_1136.xls", c("globals", "subjects", "session"))

## reading 151024_1136.xls ...
## Skipping: summary
## Doing: globals
## Doing: subjects
## Doing: session

# Displays imported data structure
str(Data)

## List of 3
## $ globals :'data.frame': 10 obs. of 6 variables:
## ..$ Date : chr [1:10] "151024_1136" "151024_1136" "151024_1136" "151024_1136" ...
## ..$ Treatment : num [1:10] 1 1 1 1 1 1 1 1 1 1
## ..$ Period : num [1:10] 1 2 3 4 5 6 7 8 9 10
## ..$ NumPeriods : num [1:10] 10 10 10 10 10 10 10 10 10 10
## ..$ RepeatTreatment: num [1:10] 0 0 0 0 0 0 0 0 0 0
## ..$ Timeout : num [1:10] 30 30 30 30 30 30 30 30 30 30
## $ subjects:'data.frame': 240 obs. of 15 variables:
## ..$ Date : chr [1:240] "151024_1136" "151024_1136" "151024_1136" "151024_1136" ...
## ..$ Treatment : num [1:240] 1 1 1 1 1 1 1 1 1 1 ...
## ..$ Period : num [1:240] 1 1 1 1 1 1 1 1 1 1 ...
## ..$ Subject : num [1:240] 1 2 3 4 5 6 7 8 9 10 ...
## ..$ Group : num [1:240] 1 1 1 1 2 2 2 2 3 3 ...
## ..$ Profit : num [1:240] 28.6 31.6 29.6 28.6 35.2 24.2 26.2 29.2 26.8 29.8 ...
## ..$ TotalProfit : num [1:240] 28.6 31.6 29.6 28.6 35.2 24.2 26.2 29.2 26.8 29.8 ...
## ..$ Participate : num [1:240] 1 1 1 1 1 1 1 1 1 1 ...
## ..$ EfficiencyFactor : num [1:240] 1.6 1.6 1.6 1.6 1.6 1.6 1.6 1.6 1.6 1.6 ...
## ..$ Endowment : num [1:240] 20 20 20 20 20 20 20 20 20 20 ...
## ..$ Contribution : num [1:240] 17 14 16 17 8 19 17 14 12 9 ...
## ..$ SumC : num [1:240] 64 64 64 64 58 58 58 58 47 47 ...
## ..$ N : num [1:240] 4 4 4 4 4 4 4 4 4 4 ...
## ..$ TimeOKContributionEntryOK : num [1:240] 23 -9 -19 -24 -29 -36 -43 -46 -50 -55 ...
## ..$ TimeContinueProfitDisplayOK: num [1:240] 0 NA NA NA NA NA NA NA NA NA ...
## $ session :'data.frame': 24 obs. of 9 variables:
```

```
## ..$ Date           : chr [1:24] "151024_1136" "151024_1136" "151024_1136" "151024_1136" ...
## ..$ Treatment      : num [1:24] 1 1 1 1 1 1 1 1 1 1 ...
## ..$ Subject        : num [1:24] 1 2 3 4 5 6 7 8 9 10 ...
## ..$ FinalProfit     : num [1:24] 18.1 18.1 18.2 19.2 17.9 ...
## ..$ ShowUpFee       : num [1:24] 10 10 10 10 10 10 10 10 10 10 ...
## ..$ ShowUpFeeInvested: num [1:24] 0 0 0 0 0 0 0 0 0 0 ...
## ..$ MoneyAdded      : num [1:24] 0 0 0 0 0 0 0 0 0 0 ...
## ..$ MoneyToPay      : num [1:24] 28.1 28.1 28.2 29.2 27.9 ...
## ..$ MoneyEarned     : num [1:24] 28.1 28.1 28.2 29.2 27.9 ...
```

3. Calculate summary variables:

```
# Calculates summary variables and displays them for verification purposes
NumSubjects <- print(max(Data$subjects$Subject))

## [1] 24

NumPeriods <- print(max(Data$subjects$Period))

## [1] 10
```

2 Descriptive statistics

1. Take a first peek at the data:

```
# Prints subjects' contributions in the first period
Data$subjects$Contribution[Data$subjects$Period == 1]

## [1] 17 14 16 17 8 19 17 14 12 9 12 14 3 7 5 18 11 12 14 5 9 20 20
## [24] 14

# Prints subjects' contributions in the last period
Data$subjects$Contribution[Data$subjects$Period == NumPeriods]

## [1] 9 8 4 10 11 12 18 3 4 11 12 14 8 9 10 8 10 11 19 17 5 2 4
## [24] 7

# Prints mean contribution over all periods and subjects
mean(Data$subjects$Contribution)

## [1] 10.52917
```

2. Create a *list* containing 1) mean and standard deviation of contributions within each period, and 2) mean and standard deviation of contributions for each subject:

```
### Calculates descriptive statistics

Summary <- list(ByPeriod = matrix(NA, ncol = 2, nrow = NumPeriods + 1), BySubject = matrix(NA,
  ncol = 2, nrow = NumSubjects + 1))

# By period
dimnames(Summary$ByPeriod) <- list(c(1:NumPeriods, "Total"), c("Mean", "StdDev"))
Summary$ByPeriod[1:NumPeriods, "Mean"] <- aggregate(Data$subjects$Contribution,
```

```

by = list(Data$subjects$Period), mean)[, 2] #Contribution mean
Summary$ByPeriod[1:NumPeriods, "StdDev"] <- aggregate(Data$subjects$Contribution,
by = list(Data$subjects$Period), sd)[, 2] #Contribution standard deviation

# By subject
dimnames(Summary$BySubject) <- list(c(1:NumSubjects, "Total"), c("Mean", "StdDev"))
Summary$BySubject[1:NumSubjects, "Mean"] <- aggregate(Data$subjects$Contribution,
by = list(Data$subjects$Subject), mean)[, 2] #Contribution mean
Summary$BySubject[1:NumSubjects, "StdDev"] <- aggregate(Data$subjects$Contribution,
by = list(Data$subjects$Subject), sd)[, 2] #Contribution standard deviation

# Adds totals
Summary$ByPeriod["Total", "Mean"] <- Summary$BySubject["Total", "Mean"] <- mean(Data$subjects$Contribution)
Summary$ByPeriod["Total", "StdDev"] <- Summary$BySubject["Total", "StdDev"] <- sd(Data$subjects$Contribution)

### Displays summary data
Summary$BySubject

##           Mean    StdDev
## 1      10.90000  5.762908
## 2      10.90000  4.581363
## 3      10.70000  6.236986
## 4       9.30000  3.433495
## 5      12.10000  4.332051
## 6      11.90000  5.820462
## 7       8.90000  6.367452
## 8      11.40000  5.815688
## 9      11.70000  5.888784
## 10     9.60000  3.405877
## 11     11.80000  5.473167
## 12     11.90000  5.404730
## 13     11.00000  5.715476
## 14       7.00000  5.249339
## 15     11.60000  4.325634
## 16     10.80000  5.513620
## 17       8.30000  3.743142
## 18     12.40000  5.420127
## 19     14.20000  5.613476
## 20     12.40000  4.765618
## 21       7.30000  5.812821
## 22       9.00000  5.981453
## 23       6.60000  5.103376
## 24     11.00000  6.359595
## Total  10.52917  5.394013

Summary$ByPeriod

##           Mean    StdDev
## 1      12.791667  4.881012
## 2      11.458333  5.332258
## 3      11.458333  4.624878
## 4       9.958333  5.864256
## 5       8.958333  6.160739
## 6      12.000000  5.373041
## 7       9.000000  5.793550
## 8       9.750000  4.971659
## 9      10.500000  5.618293
## 10      9.416667  4.577038
## Total  10.529167  5.394013

```

3. Plot contributions over subjects and over time:

```
# Plots contributions by subject
barplot(Summary$BySubject[1:NumSubjects, "Mean"], main = "Mean contribution per period, by subject",
        xlab = "Subject", ylab = "Mean contribution (points)")
```

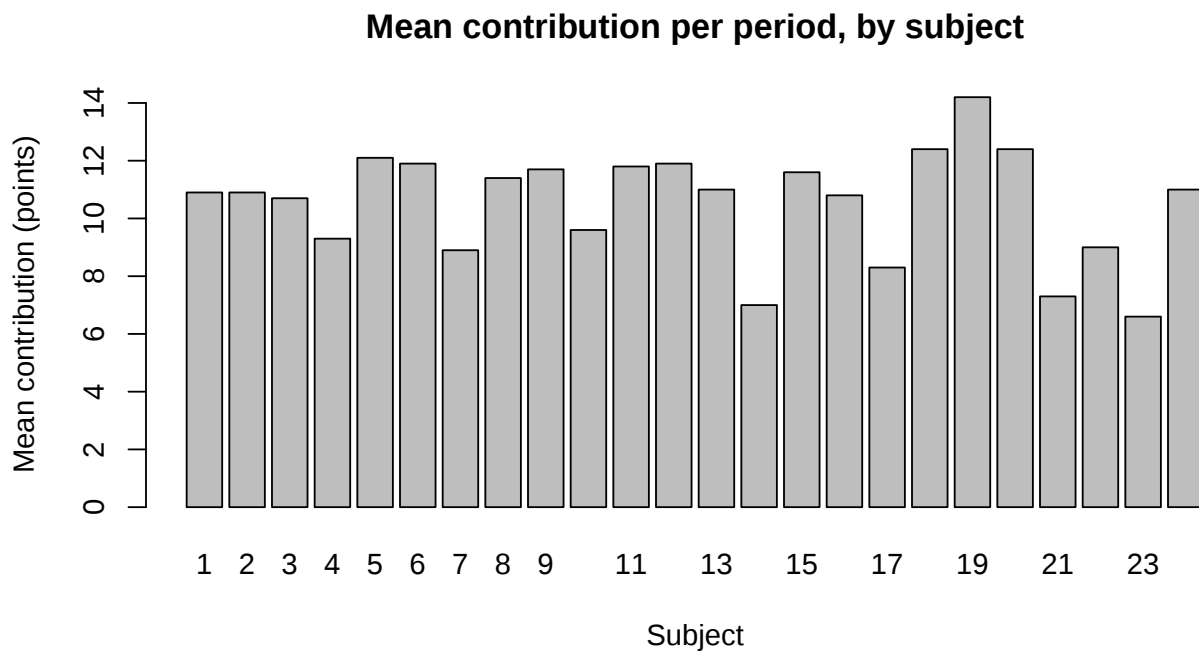


Figure 1: Contributions per subject

```
# Plots contributions over time
plot(Summary$ByPeriod[1:NumPeriods, "Mean"], t = "l", lwd = 2, ylim = c(0, max(Summary$ByPeriod[1:NumPeriods, "Mean"]) * 1.1), main = "Development of contributions over time", xlab = "Period",
      ylab = "Contribution (points)")
```

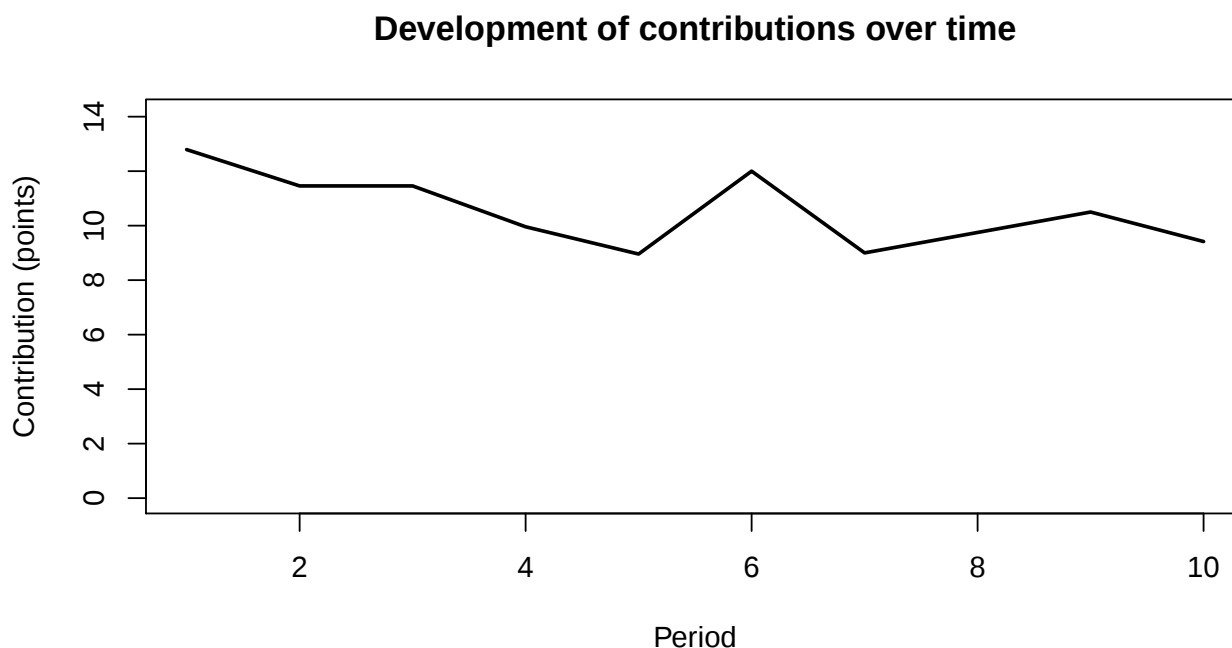


Figure 2: Development of contributions over time

3 Inferential statistics

1. Regress contributions on period and display the results:

```
### Performs regression
M1 <- lm(Contribution ~ Period, data = as.data.frame(Data$subjects))
summary(M1)

##
## Call:
## lm(formula = Contribution ~ Period, data = as.data.frame(Data$subjects))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.9424  -3.6924  -0.0292   4.1775  10.4351
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  12.0444     0.7455  16.155  <2e-16 ***
## Period       -0.2755     0.1202  -2.293   0.0227 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.347 on 238 degrees of freedom
## Multiple R-squared:  0.02161, Adjusted R-squared:  0.0175
## F-statistic: 5.257 on 1 and 238 DF, p-value: 0.02273
```

2. Add regression line to Figure 2:

```
# Adds regression line
abline(M1, col = 2, xpd = FALSE)
```

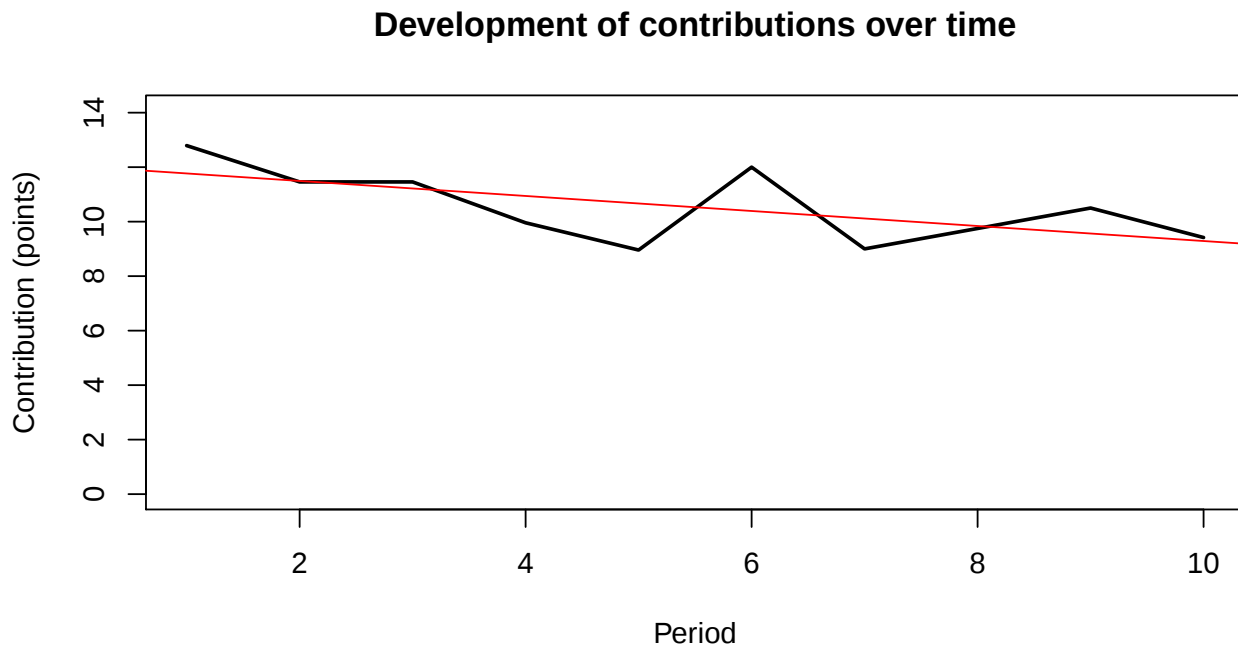


Figure 3: Development of contributions over time including linear fit

3. Test for significance of change in contributions between first and last periods:

```
# Tests for equality of contributions in first and last period
t.test(Data$subjects$Contribution[Data$subjects$Period == 1], Data$subjects$Contribution[Data$subjects$Period == NumPeriods], paired = TRUE)

##
## Paired t-test
##
## data: Data$subjects$Contribution[Data$subjects$Period == 1] and Data$subjects$Contribution[Data$subjects$Period == NumPeriods]
## t = 2.254, df = 23, p-value = 0.03403
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.2774836 6.4725164
## sample estimates:
## mean of the differences
##                3.375

wilcox.test(Data$subjects$Contribution[Data$subjects$Period == 1], Data$subjects$Contribution[Data$subjects$Period == NumPeriods], paired = TRUE)

## Warning in wilcox.test.default(Data$subjects$Contribution[Data$subjects$Period == 1], Data$subjects$Contribution[Data$subjects$Period == NumPeriods]): cannot
## compute exact p-value with ties
## Warning in wilcox.test.default(Data$subjects$Contribution[Data$subjects$Period == 1], Data$subjects$Contribution[Data$subjects$Period == NumPeriods]): cannot
## compute exact p-value with zeroes

##
## Wilcoxon signed rank test with continuity correction
##
## data: Data$subjects$Contribution[Data$subjects$Period == 1] and Data$subjects$Contribution[Data$subjects$Period == NumPeriods]
## V = 189.5, p-value = 0.04224
## alternative hypothesis: true location shift is not equal to 0
```