Example: data analysis from the TGPP experiment

Jacek Matulewski
version: October 24, 2021 (translation: June 11, 2021)

Best performed in a notebook in R Studio

The course of the experiment:

The experiment involved children with autism spectrum disorder. The goal was the goal was to test effectiveness of the training with mobile app improving daily functioning skills. It consisted of three stages: pre-tests, cognitive training (in the form of a therapeutic game), and post-tests. The participants were divided into two groups: test and control (not participating in the training).

Cf. TGPP (two-groups, pretest-posttest) experiment design

Input data:

The input data are the anonymised results of the "I can/I can't" questionnaire diagnosing children's skills of everyday functioning and elements of social communication. The same questionnaire was filled in by the children and the parents (the parents described the children's skills). The data is stored in four text files (ANSI encoding) in CSV format (children-pretests, children-post-tests, parents-pretests, parents-post-tests), in which the respondents are in the columns (anonymised, codes of respondents) and their answers to subsequent questions of the survey are in rows (on a scale from 1 - "no" to 4 - "yes"). In a separate file there is the allocation of the respondents to groups (the first letter of the respondent's code does not necessary prove that the participant belongs to it).

Not all survey questions are related to training. Therefore, we will only select questions from 12 to 15, from 23 to 26, 35 to 43, question 45, and from 48 to 63.

The data is not complete. Questionnaires for some children are missing. Also, some questions were omitted by some respondents.

Files:

AnkietyPotrafie_Pretest_Dzieci.csv – results of the questionnaire completed by children, pretest
AnkietyPotrafie_Posttest_Dzieci.csv – results of the questionnaire completed by children, posttest
AnkietyPotrafie_Pretest_Rodzice.csv – results of the questionnaire completed by the parents, pretest
AnkietyPotrafie_Posttest_Rodzice.csv – results of the questionnaire completed by the parents, posttest
Grupy.csv – assignments of children to groups

Research questions:

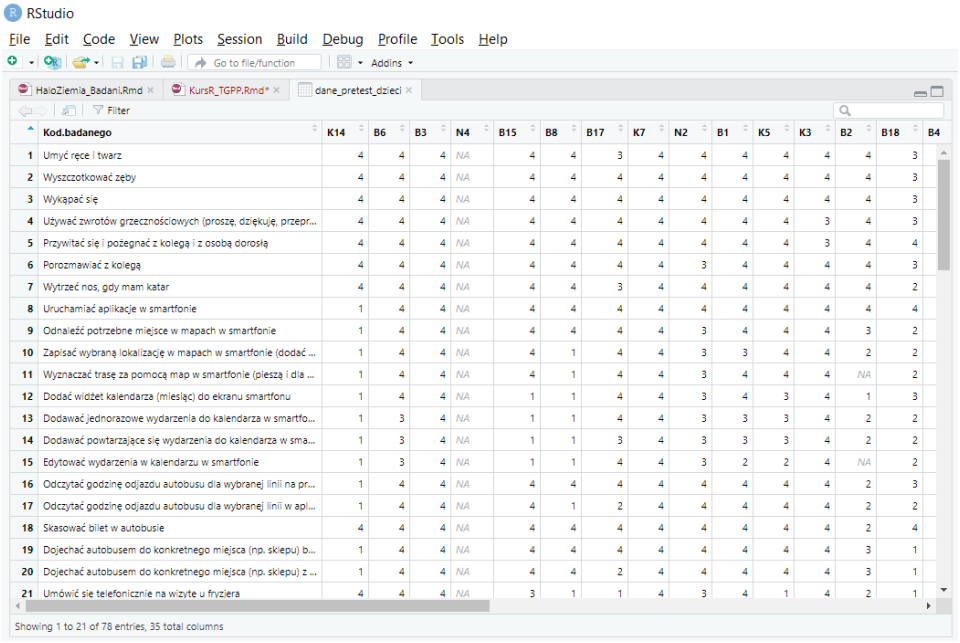
1. Has the level of skills measured by the questionnaire changed more in the test group than in the control group?
2. How much is the self-assessment of children's skills different from that of their parents?

Analysis:

1. Reading, verification and ordering of data

Loading data into `data.frame` objects
(pl. dzieci - children, rodzice - parents)

```
rm(list = ls())
dane_pretest_dzieci <- read.csv("AnkietyPotrafie_Pretest_Dzieci.csv",
       sep = ";", encoding = "ASCII", stringsAsFactors = FALSE)
dane_pretest_rodzice <- read.csv("AnkietyPotrafie_Pretest_Rodzice.csv",
       sep = ";", encoding = "ASCII", stringsAsFactors = FALSE)
dane_posttest_dzieci <- read.csv("AnkietyPotrafie_Posttest_Dzieci.csv",
       sep = ";", encoding = "ASCII", stringsAsFactors = FALSE)
dane_posttest_rodzice <- read.csv("AnkietyPotrafie_Posttest_Rodzice.csv",
       sep = ";", encoding = "ASCII", stringsAsFactors = FALSE)
```



	Kod.badanego	K14	B6	B3	N4	B15	B8	B17	K7	N2	B1	K5	K3	B2	B18	B4
1	Umyć ręce i twarz	4	4	4	NA	4	4	3	4	4	4	4	4	4	3	
2	Wyszczotkować zęby	4	4	4	NA	4	4	4	4	4	4	4	4	4	3	
3	Wykąpać się	4	4	4	NA	4	4	4	4	4	4	4	4	4	3	
4	Używać zwrotów grzecznościowych (proszę, dziękuję, przepr...	4	4	4	NA	4	4	4	4	4	4	4	3	4	3	
5	Przywitać się i pożegnać z kolegą i z osobą dorosłą	4	4	4	NA	4	4	4	4	4	4	4	3	4	4	
6	Porozmawiać z kolegą	4	4	4	NA	4	4	4	4	3	4	4	4	4	3	
7	Wytrzeć nos, gdy mam katar	4	4	4	NA	4	4	3	4	4	4	4	4	4	2	
8	Uruchamiać aplikacje w smartfonie	1	4	4	NA	4	4	4	4	4	4	4	4	4	4	
9	Odnaleźć potrzebne miejsce w mapach w smartfonie	1	4	4	NA	4	4	4	4	3	4	4	4	3	2	
10	Zapisać wybraną lokalizację w mapach w smartfonie (dodać ...	1	4	4	NA	4	1	4	4	3	3	4	4	2	2	
11	Wyznaczać trasę za pomocą map w smartfonie (pieszą i dla ...	1	4	4	NA	4	1	4	4	3	4	4	4	NA	2	
12	Dodać widżet kalendarza (miesiąc) do ekranu smartfonu	1	4	4	NA	1	1	4	4	3	4	3	4	1	3	
13	Dodawać jednorazowe wydarzenia do kalendarza w smartfo...	1	3	4	NA	1	1	4	4	3	3	3	4	2	2	
14	Dodawać powtarzające się wydarzenia do kalendarza w sma...	1	3	4	NA	1	1	3	4	3	3	3	4	2	2	
15	Edytować wydarzenia w kalendarzu w smartfonie	1	3	4	NA	1	1	4	4	3	2	2	4	NA	2	
16	Odczytać godzinę odjazdu autobusu dla wybranej linii na pr...	1	4	4	NA	4	4	4	4	4	4	4	4	2	3	
17	Odczytać godzinę odjazdu autobusu dla wybranej linii w apl...	1	4	4	NA	4	1	2	4	4	4	4	4	2	2	
18	Skasować bilet w autobusie	4	4	4	NA	4	4	4	4	4	4	4	4	2	4	
19	Dojechać autobusem do konkretnego miejsca (np. sklepu) b...	1	4	4	NA	4	4	4	4	4	4	4	4	3	1	
20	Dojechać autobusem do konkretnego miejsca (np. sklepu) z ...	1	4	4	NA	4	4	2	4	4	4	4	4	3	1	
21	Umówić się telefonicznie na wizytę u fryzjera	4	4	4	NA	3	1	1	4	3	4	1	4	2	1	

Showing 1 to 21 of 78 entries, 35 total columns

We check how many rows have the loaded tables. They should all have 78 lines (this is the number of questions in the survey).

```
nrow(dane_pretest_dzieci)
nrow(dane_pretest_rodzice)
nrow(dane_posttest_dzieci)
nrow(dane_posttest_rodzice)
```

We select questions that we will analyze. After that, the data should be 34 lines long.

```
pytania_z_gry <- c(12:15, 23:26, 35:43, 45, 48:63)
pytania_z_gry
dane_pretest_dzieci <- dane_pretest_dzieci[pytania_z_gry,]
dane_pretest_rodzice <- dane_pretest_rodzice[pytania_z_gry,]
dane_posttest_dzieci <- dane_posttest_dzieci[pytania_z_gry,]
dane_posttest_rodzice <- dane_posttest_rodzice[pytania_z_gry,]
```

There are no questionnaires from some respondents (entire columns with NA values). Let's remove such columns.
(pl. usuń – remove, puste kolumny – empty columns)

```
usunPusteKolumny <- function(df)
{
  puste_kolumny <- apply(df=="" | is.na(df),2,all)
  #is.na in case it doesn't even have a header
  puste_kolumny
  result <- df[,!puste_kolumny]
  return(result)
}

dane_pretest_dzieci <- usunPusteKolumny(dane_pretest_dzieci)
dane_pretest_rodzice <- usunPusteKolumny(dane_pretest_rodzice)
dane_posttest_dzieci <- usunPusteKolumny(dane_posttest_dzieci)
dane_posttest_rodzice <- usunPusteKolumny(dane_posttest_rodzice)
```

Since we were reading data as strings (without transforming it to factors), there is a value of NA in empty cells (if `stringAsFactors = TRUE` would was set, there would be empty strings).

```
#one can also use the na.strings=c("","NA") parameter when reading
dane_pretest_dzieci[dane_pretest_dzieci == ""] <- NA
dane_pretest_rodzice[dane_pretest_rodzice == ""] <- NA
dane_posttest_dzieci[dane_posttest_dzieci == ""] <- NA
dane_posttest_rodzice[dane_posttest_rodzice == ""] <- NA
```

To convert data from strings to numbers, we get rid of the first column, which are the questions of the "I can/I can't" questionnaire.

```
dane_pretest_dzieci_n = as.data.frame(
      sapply(dane_pretest_dzieci[,-c(1)], as.numeric))
dane_pretest_rodzice_n = as.data.frame(
      sapply(dane_pretest_rodzice[,-c(1)], as.numeric))
dane_posttest_dzieci_n = as.data.frame(
      sapply(dane_posttest_dzieci[,-c(1)], as.numeric))
dane_posttest_rodzice_n = as.data.frame(
      sapply(dane_posttest_rodzice[,-c(1)], as.numeric))
```

Check whether there is any data that does not fall within the range from 1 to 4, e.g. for pre-tests:

```
min(dane_pretest_dzieci_n, na.rm = TRUE)
max(dane_pretest_dzieci_n, na.rm = TRUE)
```

Calculation of average values for individual subjects (columns), omitting empty cells:

```
dane_pretest_dzieci_means <- colMeans(dane_pretest_dzieci_n, na.rm = TRUE)
dane_pretest_rodzice_means <- colMeans(dane_pretest_rodzice_n, na.rm = TRUE)
dane_posttest_dzieci_means <- colMeans(dane_posttest_dzieci_n, na.rm = TRUE)
dane_posttest_rodzice_means <- colMeans(dane_posttest_rodzice_n, na.rm = TRUE)
dane_pretest_dzieci_means
dane_pretest_rodzice_means
dane_posttest_dzieci_means
dane_posttest_rodzice_means
```

One can check the mean values of individual scores and their standard deviations:

```
mean(dane_pretest_dzieci_means);sd(dane_pretest_dzieci_means)
mean(dane_pretest_rodzice_means);sd(dane_pretest_rodzice_means)
mean(dane_posttest_dzieci_means);sd(dane_posttest_dzieci_means)
mean(dane_posttest_rodzice_means);sd(dane_posttest_rodzice_means)
```

In this way, we can check "by eye" whether the mean increased in the test group and did not change in the control group.

Let us assign the subjects to the test and control groups:
(pl. przydział do grup - assignment to groups, grupa kontrolna i badana - control and test group)

```
przydzial_grup <- read.csv("Grupy.csv", sep = ";", encoding = "ASCII",
stringsAsFactors = FALSE)
przydzial_grup
grupa_B <- przydzial_grup[przydzial_grup$Grupa == "B",]$Kod.badanego
grupa_K <- przydzial_grup[przydzial_grup$Grupa == "K",]$Kod.badanego
grupa_B
grupa_K
```

We split the data into four sets:
The blank data records appear because there are people present in the Group.csv file who have not returned the polls.

```
#pretest-children
names(dane_pretest_dzieci_means)
dane_pretest_dzieci_means_B <- dane_pretest_dzieci_means[grupa_B]
dane_pretest_dzieci_means_B <-
       dane_pretest_dzieci_means_B[!is.na(dane_pretest_dzieci_means_B)]

dane_pretest_dzieci_means_K <- dane_pretest_dzieci_means[grupa_K]
dane_pretest_dzieci_means_K <-
       dane_pretest_dzieci_means_K[!is.na(dane_pretest_dzieci_means_K)]


#posttest-children
names(dane_pretest_dzieci_means)
dane_posttest_dzieci_means_B <- dane_posttest_dzieci_means[grupa_B]
dane_posttest_dzieci_means_B <-
       dane_posttest_dzieci_means_B[!is.na(dane_posttest_dzieci_means_B)]

dane_posttest_dzieci_means_K <- dane_posttest_dzieci_means[grupa_K]
dane_posttest_dzieci_means_K <-
       dane_posttest_dzieci_means_K[!is.na(dane_posttest_dzieci_means_K)]

#pretest-parents
names(dane_pretest_rodzice_means)
dane_pretest_rodzice_means_B <- dane_pretest_rodzice_means[grupa_B]
```

```
dane_pretest_rodzice_means_B <-
      dane_pretest_rodzice_means_B[!is.na(dane_pretest_rodzice_means_B)]

dane_pretest_rodzice_means_K <- dane_pretest_rodzice_means[grupa_K]
dane_pretest_rodzice_means_K <-
dane_pretest_rodzice_means_K[!is.na(dane_pretest_rodzice_means_K)]

#posttest-parents
names(dane_pretest_rodzice_means)
dane_posttest_rodzice_means_B <- dane_posttest_rodzice_means[grupa_B]
dane_posttest_rodzice_means_B <-
      dane_posttest_rodzice_means_B[!is.na(dane_posttest_rodzice_means_B)]

dane_posttest_rodzice_means_K <- dane_posttest_rodzice_means[grupa_K]
dane_posttest_rodzice_means_K <-
      dane_posttest_rodzice_means_K[!is.na(dane_posttest_rodzice_means_K)]
```

Dodatkowe pytanie: How many participants have returned the fully completed questionnaires.

How to deal with missing answers to some of the survey questions? We simply omitted them when calculating the average. An alternative solution would be to assign them the value 1.

2. Checking the normality of distributions and analysis of variance
   (we will use questionnaires filled in by parents as an example)

```
#==== PARENTS ====
#prettest-posttest - repeated measurement -> dependent tests
#B-K - independent trials

#COMPARISON OF GROUPS B AND K IN PRETEST (independent tests)
length(dane_pretest_rodzice_means_B)
length(dane_pretest_rodzice_means_K)
shapiro.test(dane_pretest_rodzice_means_B)$p.value > 0.05
shapiro.test(dane_pretest_rodzice_means_K)$p.value > 0.05
#both distributions are normal
t.test(dane_pretest_rodzice_means_B,dane_pretest_rodzice_means_K,paired = FALSE)
#Result: p = 0.75 - samples are not different
#                 - groups B and K are similar, random division
```

```
        Welch Two Sample t-test

data:  dane_pretest_rodzice_means_B and dane_pretest_rodzice_means_K
t = -0.32477, df = 26.954, p-value = 0.7479
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.4895859  0.3557883
sample estimates:
mean of x mean of y
 2.675347  2.742246
```

```
#Comparison of groups B and K in the post-test
length(dane_posttest_rodzice_means_B)
length(dane_posttest_rodzice_means_K)
shapiro.test(dane_posttest_rodzice_means_B)$p.value > 0.05
shapiro.test(dane_posttest_rodzice_means_K)$p.value > 0.05
#group K distribution is normal and group B distribution is not
wilcox.test(dane_posttest_rodzice_means_B,dane_posttest_rodzice_means_K,paired =
FALSE)
# In the post-test, the groups differ significantly from each other
#(if 0.05023 is considered as 0.05)
```

```
        Wilcoxon rank sum test with continuity correction

data:  dane_posttest_rodzice_means_B and dane_posttest_rodzice_means_K
W = 170, p-value = 0.05023
alternative hypothesis: true location shift is not equal to 0
```

```
# Comparison of pre-tests and post-tests for both groups (dependent samples)

#one need to find a common subset
grupa_B_wspolna_rodzice <-
       intersect(names(dane_pretest_rodzice_means_B),
                 names(dane_posttest_rodzice_means_B))
```
#Important! Do not use the same variable in different places in the notebook, because running different fragments may produce random results.
```
grupa_B_wspolna_rodzice
length(grupa_B_wspolna_rodzice)
wilcox.test(dane_pretest_rodzice_means_B[grupa_B_wspolna_rodzice],dane_posttest_rod
zice_means_B[grupa_B_wspolna_rodzice],paired = TRUE)
```
#The results of the test group differ significantly in the pre-test and post-test

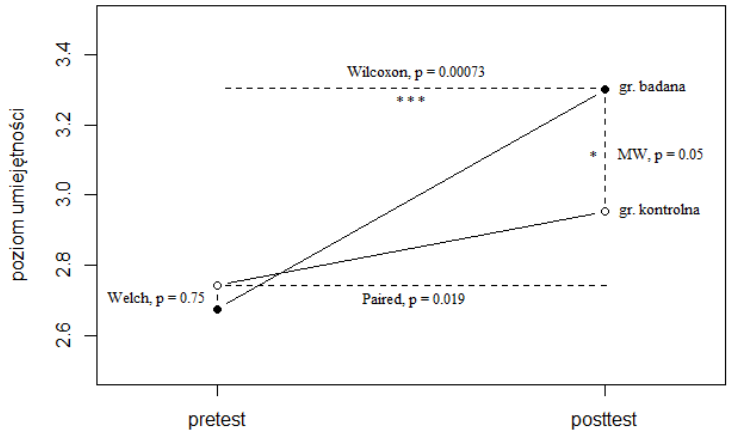
```
        Wilcoxon signed rank test with continuity correction

data:  dane_pretest_rodzice_means_B[grupa_B_wspolna_rodzice] and dane_posttest_rodzice_means_B[grupa_B_wspolna_rodzice]
V = 0, p-value = 0.0007247
alternative hypothesis: true location shift is not equal to 0
```

```
grupa_K_wspolna_rodzice <-
      intersect(names(dane_pretest_rodzice_means_K),
                names(dane_posttest_rodzice_means_K))
grupa_K_wspolna_rodzice
length(grupa_K_wspolna_rodzice)
t.test(dane_pretest_rodzice_means_K[grupa_K_wspolna_rodzice],dane_posttest_rodzice_
means_K[grupa_K_wspolna_rodzice],paired = TRUE)
```
#Group K results do not differ significantly in the pre-test and post-test

```
        Paired t-test

data:  dane_pretest_rodzice_means_K[grupa_K_wspolna_rodzice] and dane_posttest_rodzice_means_K[grupa_K_wspolna_rodzice]
t = -2.6314, df = 14, p-value = 0.01973
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.43937324 -0.04476223
sample estimates:
mean of the differences
           -0.2420677
```



(pl. poziom umiejętności – skill level, grupa badana – test group, grupa kontrolna – control group)

3. Comparing the results of the questionnaires completed by parents and children

```
#PRETEST, B
grupa_B_wspolna_pretest_dr <-
     intersect(names(dane_pretest_dzieci_means_B),
               names(dane_pretest_rodzice_means_B))
grupa_B_wspolna_pretest_dr
length(grupa_B_wspolna_pretest_dr)
t.test(dane_pretest_dzieci_means_B[grupa_B_wspolna_pretest_dr],
       dane_pretest_rodzice_means_B[grupa_B_wspolna_pretest_dr], paired = TRUE)

#PRETEST, K
grupa_K_wspolna_pretest_dr <-
     intersect(names(dane_pretest_dzieci_means_K),
               names(dane_pretest_rodzice_means_K))
grupa_K_wspolna_pretest_dr
length(grupa_K_wspolna_pretest_dr)
t.test(dane_pretest_dzieci_means_K[grupa_K_wspolna_pretest_dr],
       dane_pretest_rodzice_means_K[grupa_K_wspolna_pretest_dr], paired = TRUE)



#POSTTEST, B
grupa_B_wspolna_posttest_dr <-
     intersect(names(dane_posttest_dzieci_means_B),
               names(dane_posttest_rodzice_means_B))
grupa_B_wspolna_posttest_dr
length(grupa_B_wspolna_posttest_dr)
wilcox.test(dane_posttest_dzieci_means_B[grupa_B_wspolna_posttest_dr],
            dane_posttest_rodzice_means_B[grupa_B_wspolna_posttest_dr],
            paired = TRUE)

#POSTTEST, K
grupa_K_wspolna_posttest_dr <-
     intersect(names(dane_posttest_dzieci_means_K),
               names(dane_posttest_rodzice_means_K))
grupa_K_wspolna_posttest_dr
length(grupa_K_wspolna_posttest_dr)
t.test(dane_posttest_dzieci_means_K[grupa_K_wspolna_posttest_dr],
       dane_posttest_rodzice_means_K[grupa_K_wspolna_posttest_dr], paired = TRUE)
```

 # In all cases, the ratings are significantly different outside the test group in the post-tests
(reconciliation of assessments as a result of the intervention?)

4. Plots

```
#Parents plot
plotDaneB <- c("pretest" = mean(dane_pretest_rodzice_means_B),
               "posttest" = mean(dane_posttest_rodzice_means_B))
plotDaneB
names(plotDaneB)

plotDaneK <- c("pretest" = mean(dane_pretest_rodzice_means_K),
               "posttest" = mean(dane_posttest_rodzice_means_K))
plotDaneK
names(plotDaneK)

par(oma = c(1, 0, 0, 0), mar = c(3, 7, 2, 2))
plot(plotDaneB, type="b", xlab = "", ylab = "poziom umiejętności",
     xlim=c(0.75,2.25), ylim=c(2.5,3.5), xaxt="n", pch=19)
points(plotDaneK, type = "b", pch=1)
axis(1, at=1:2, tick=1, labels=names(plotDaneB))
```