# Przykład: analiza danych z eksperymentu TGPP

Jacek Matulewski
wersja z 24 października 2021, poprawki z 11 czerwca 2022

*Najlepiej wykonywać w notebooku w R Studio*

**Przebieg eksperymentu:**

Eksperyment obejmował dzieci z zaburzeniami ze spektrum autyzmu. Celem było sprawdzenie skuteczności aplikacji mobilnej wspierającej trening poprawiający umiejętności funkcjonowania codziennego (TFC). Składał się z trzech etapów: pretesty, trening poznawczy (w postaci gry terapeutycznej), posttesty. Uczestnicy byli podzieleni na dwie grupy: badana i kontrolna (nie biorąca udziału w treningu).
Por. projekt eksperymentu TGPP (z ang. *two-groups, pretest-posttest*)

**Dane wejściowe:**

Dane wejściowe to zanonimizowane wyniki ankiety „Potrafię/Nie potrafię" diagnozującej umiejętności dzieci w zakresie funkcjonowania codziennego i elementów komunikacji społecznej. Taka sama ankieta była wypełniana przez dziecko i przez rodzica (rodzic opisywał umiejętności dzieci). Dane umieszczone są w czterech plikach tekstowych (kodowanie ANSI) w formacie CSV (dzieci-pretesty, dzieci-posttesty, rodzice-pretesty, rodzice-posttesty), w których w kolumnach są osoby badane (zanonimizowane, kody badanych), a w wierszach odpowiedzi na kolejne pytania ankiety (w skali od 1 – „nie" do 4 – „tak"). W osobnym pliku jest przydział badanych do grup (o przynależności nie świadczy pierwsza litera kodu badanego).

Nie wszystkie pytania ankiety są związane z treningiem. Dlatego wybierzemy jedynie pytania od 12 do 15, od 23 do 26, od 35 do 43, pytanie 45 oraz od 48 do 63.

Dane nie są kompletne. Brakuje całych ankiet dla niektórych dzieci. Również niektóre pytania zostały pominięte przez niektórych respondentów.

Pliki:

*AnkietyPotrafie_Pretest_Dzieci.csv* – wyniki ankiety wypełnionej przez dzieci, pretest
*AnkietyPotrafie_Posttest_Dzieci.csv* – wyniki ankiety wypełnionej przez dzieci, posttest
*AnkietyPotrafie_Pretest_Rodzice.csv* – wyniki ankiety wypełnionej przez rodziców, pretest
*AnkietyPotrafie_Posttest_Rodzice.csv* – wyniki ankiety wypełnionej przez rodziców, posttest
*Grupy.csv* – przydziały dzieci do grup

**Pytania badawcze:**

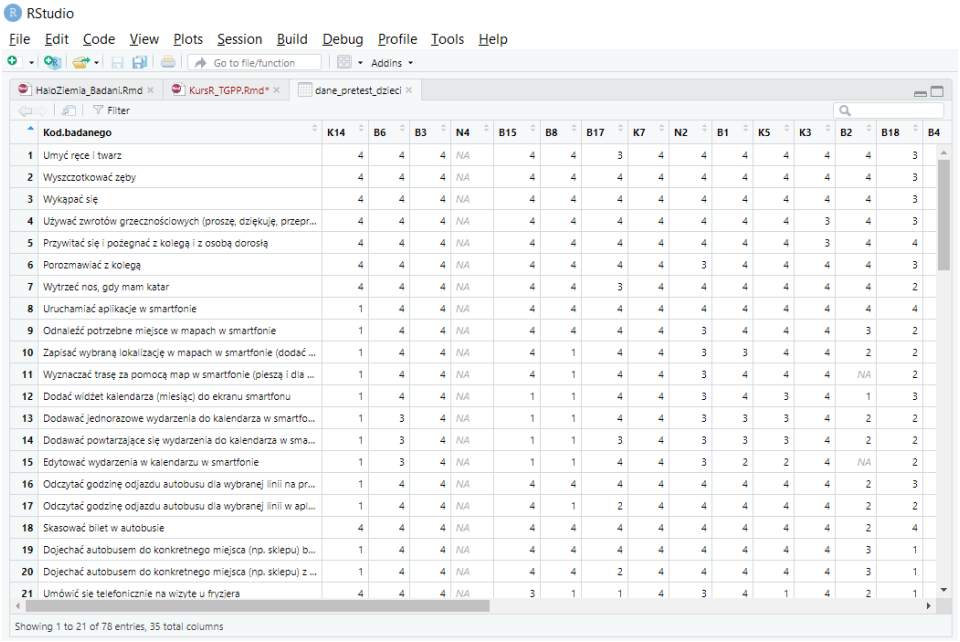
1. Czy poziom umiejętności mierzony ankietą zmieniły się u dzieci z grupy badanej bardziej niż w grupie kontrolnej?
2. Jak bardzo różni się obraz umiejętności dzieci i ich rodziców?

**Analiza:**

1. Wczytywanie, weryfikacja i porządkowanie danych

Wczytywanie danych do obiektów `data.frame`

```
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

Sprawdzamy ile wierszy mają wczytane tabele. Wszystkie powinny mieć 78 wierszy (tyle jest pytań w ankiecie).

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

Wybieramy pytania, które będziemy analizować. Po tym dane powinny mieć po 34 wiersze.

```
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,]
```

Brakuje ankiet od niektórych badanych (całe kolumny z wartościami NA). Usuńmy takie kolumny.

```
usunPusteKolumny <- function(df)
{
  puste_kolumny <- apply(df=="" | is.na(df),2,all)
  #is.na w przypadku gdy nie ma nawet nagłówka
  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)
```

Ponieważ wczytywaliśmy dane jako łańcuchy (bez zmieniania ich na czynniki), w pustych komórkach jest wartość NA (przy ustawieniu `stringAsFactors = TRUE` były by tam puste łańcuchy).

```
#można też przy czytaniu użyć parametru na.strings=c("","NA")
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
```

Aby skonwertować dane z łańcuchów do liczb, pozbywamy się pierwszej kolumny, w której są pytania ankiety „Potrafię/Nie potrafię”.

```
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))
```

Sprawdzamy, czy nie ma danych, które nie należą do zakresu od 1 do 4 np. dla pretestów:

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

Obliczenie średnich wartości dla poszczególnych badanych (kolumn) z pominięciem pustym komórek:

```
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
```

Można sprawdzić średnie wartości wyników poszczególnych osób i ich odchylenia standardowe:

```
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)
```


Możemy w ten sposób „na oko” sprawdzić, czy średnia podniosła się w grupie badanej i nie zmieniała w grupie kontrolnej.

Przydzielmy badanych do grupy badanej i kontrolnej:

```
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
```

Rozdzielamy dane na cztery zbiory:
*Puste dane biorą się z tego, że w pliku Grupy.csv są osoby, które nie oddały ankiet.*

```
#pretest-dzieci
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-dzieci
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-rodzice
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-rodzice
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:** Ile uczestników ma wszystkie wypełnione ankiety.

Jak traktować braki odpowiedzi na niektóre pytania w ankiecie? My je po prostu pominęliśmy obliczając średnią. Alternatywnym rozwiązaniem byłoby przypisanie im wartości 1.

2. Sprawdzenie normalności rozkładów i analiza wariancji
   (dla przykładu użyjemy ankiet wypełnianych przez rodziców)

```
#==== RODZICE ====
#prettest-posttest - powtórzony pomiar -> próby zależne
#B-K - próby niezależne

#PORÓWNANIE GRUP B i K W PRETEŚCIE (próby niezależne)
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
#oba rozkłady są normalne
t.test(dane_pretest_rodzice_means_B,dane_pretest_rodzice_means_K,paired = FALSE)
#Wynik: p = 0.75 - próby nie są różne – grupa B i K są podobne, podział losowy
```

```
        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
```

```
#Porównanie grup B i K w poststeście
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
#rozkład grupy K jest normalny, a grupy B - nie
wilcox.test(dane_posttest_rodzice_means_B,dane_posttest_rodzice_means_K,paired =
FALSE)
#W poststeście grupy różnią się od siebie istotnie (jeżeli 0.05023 uznać za 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
```

```
#Porównanie pretestów i posttestów dla obu grup (próby zależne)

#trzeba znaleźć wspólny podzbiór
grupa_B_wspolna_rodzice <-
       intersect(names(dane_pretest_rodzice_means_B),
                 names(dane_posttest_rodzice_means_B))
```
**#Ważne! Nie używać tej samej zmiennej w różnych miejscach notatnika, bo wówczas uruchamianie różnych fragmentów może dać przypadkowe wyniki.**
```
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)
```
**#Wyniki grupy badanej różnią się istotnie w preteście i postteście**

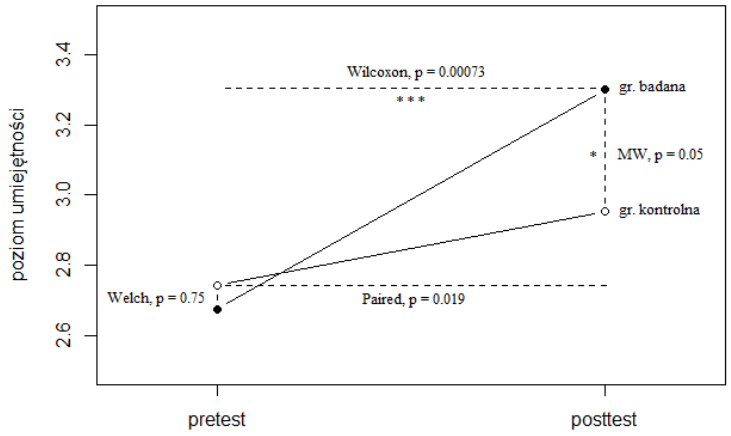
```
        Wilcoxon signed rank test with continuity correction

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)
```
**#Wyniki grupy K nie różnią się istotnie w preteście i postteście**

```
        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
```



3. Porównanie wyników ankiet wypełnionych przez rodziców i dzieci

```
#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)
```

**#We wszystkich przypadkach oceny są istotnie różne poza grupą badaną w posttestach**
(uzgodnienie ocen w wyniku interwencji?)

4. Wykres

```
#Wykres r`odzice
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))
```