Poligon Doświadczalny

Czyli co młody statystyk jest w stanie wyczytać z danych PISA2012?

Marcin Kosiński[1234], Marcin Kania[1], Marta Szczerbień[134],
Magda Waśniowska[1], Patrycja Wiśniewska[15]

mailkola@mimuw.edu.pl

6 marca 2014

[1]Wydział Matematyki, Informatyki i Mechaniki, Uniwersytet Warszawski
[2]Wydział Matematyki i Nauk Informacyjnych, Politechnika Warszawska
[3]Wydział Biologii, Uniwersytet Warszawski
[4]Wydział Fizyki, Uniwersytet Warszawski
[5]Wydział Nauk Ekonomicznych, Uniwersytet Warszawski

Streszczenie

Poniższy dokument przedstawia raport [na temat] stworzony przez członków Koła Zarządzania Projektami Statystycznymi Uniwersytetu Warszawskiego. Dane niezbędne do analizy zostały zaczerpnięte z badania `PISA2012` - Programme for International Student Assesment, przeprowadzanego przez Organisation for Economic Co-operation and Development (na pewno?). Raport jest w trakcie tworzenia. Każdy zainteresowany może dopisać swój rozdział bądź podrozdział i zostanie uwzględniony w spisie autorów w ostatecznej wynikowej wersji naszych eksperymentalnych potyczek ze zbiorem danych z `PISA2012`.

Autorzy

Marcin Kania - Rozdział 2 4

Marcin Kosiński - Rozdział 1 4

Marta Szczerbień - Rozdział 2

Magda Waśniowska - Rozdział 3 4

Patrcyja Wiśniewska - Rozdział 3 4

SPIS TREŚCI

ROZDZIAŁ 1

WCZYTANIE PISA2012 DO $\mathcal{R}$

1.1 Kwestionariusze osobowe uczniów

Poniżej mała instrukcja jak dokopać się do danych z PISA2012, aby działały w $\mathcal{R}$. Dane w formacie `.txt` pobieramy stąd. Następnie w systemie `SAS` tworzymy nowy program, którego 3 pierwsze linie można (ale nie trzeba) wpisać jak poniżej:

```
libname  MD "D:\PISA 2012";
filename STU "D:\PISA 2012\INT_STU12_DEC03.txt";
options nofmterr;
```

Kolejne linie w programie powinny byc przekopiowane z tego pliku. W tym momencie można już wywołać cały program w `SAS`, aby uzyskać pełną bazę danych `PISA2012`. Ponieważ baza zajmuje około 1,5 GB, ograniczymy się jedynie do danych dotyczących Polski, dzięki czemu program $\mathcal{R}$ będzia działał sprawniej na mniej pojemnym pliku. Posłuży do tego zapytanie `SQL`, które prezentuję poniżej:

```
proc sql;
create table POL as
select *
from Md.Stu
where CNT = 'POL'
;
```

Pomimo, że pierwsza kolumna bazy, z której wybieramy jedynie Polskę, ma widniejący podpis `Country code 3-character`, to jednak po wyświetleniu atrytbutów kolumny widać, że jej nazwa to `CNT`, a `Country code 3-character` to jedynie etykieta. Dodatkowo można w ten sposób odczytać informację o długości znaków w tej kolumnie, która wynosi 3, dlatego ostatecznie w zapytaniu `SQL` widnieje linia `where CNT = 'POL'`.

Tak pomniejszoną bazę danych eksportuję do formatu `.csv` (możliwe, że bezmyślnie), dzięki procedurze `export`. Wszystkie dotychczasowe komendy i operacja odbywały się w systemie `SAS`.

```
proc export data=Pol
   outfile='D:\PISA 2012\polska.csv'
   dbms=csv
   replace;
run;
```

Ostatecznie z pliku `.csv` można już "tradycyjnie" wczytać dane do pakietu $\mathcal{R}$, używając prostego polecenia `read.csv`.

```
POL <- read.csv("D:/PISA 2012/polska.csv", sep = ",", h = TRUE)
```

W rezulatacie wymiar bazdy danych, dotyczących jedynie Polski to:

```
dim(POL)

[1] 4607  634
```

A rozmiar, w bajtach:

```
file.info("D:/PISA 2012/polska.csv")$size

[1] 25376098
```

Dla porównania, cała baza danych `PISA2012` jeszcze w formacie `.txt`:

```
format(file.info("D:/PISA 2012/INT_STU12_DEC03.txt")$size, digits = 15)

[1] "1140901500"
```

Opisy poszczególnych kolumn można znaleźć w Codebook'u. Należy pamiętać, że powyższa baza danych dotyczyła jedynie kwestionariuszy wypełnianych przez uczniów.

Więcej na ten temat można znaleźć na stronie PISA2012.

1.2 Wyniki uzyskane przez uczniów w Polsce

Podobne kroki wykonuję się, aby wgrać do pakietu $\mathcal{R}$ wyniki uzyskane przez Polskich szesnatolatków. Dane w formacie `.txt` pobieramy stąd. Przy użyciu tych samych komend w `SAS`, tworzę plik o rozszerzeniu `.csv` zawierający wyniki. Następnie wgrywam je do $\mathcal{R}$ i łączę z poprzednią ramką danych (być może bezmyślnie).

```
Wyn <- read.csv("D:/PISA 2012/wyniki.csv", sep = ",", h = TRUE)
polo <- merge(POL, Wyn)
```

Następnie by można było ewentualnie przetransportować połączone dane używam poniższych komend do zapisu scalonej bazy danych w formacie `.txt` i `.csv`

```
write.csv(polo, "D:/PISA 2012/polaczone.csv")
write.table(polo, "D:/PISA 2012/polaczone.txt", sep = "\t")
```

Obecnie dane można wczytać poleceniami:

```
ponowne <- read.table("D:/PISA 2012/polaczone.txt", sep = "\t", header = TRUE)
ponowne2 <- read.csv("D:/PISA 2012/polaczone.csv", sep = ",", h = TRUE)
```

Zbiór danych wczytanych z pliku `.csv` zawiera na początku jedną dodatkową kolumnę zawierającą liczbę porządkową danego gimnazjalisty.

```
dim(ponowne)

[1] 4607  843

dim(ponowne2)

[1] 4607  844
```

ROZDZIAŁ 2

PROCENTOWE PORÓWNANIE ODPOWIEDZI DZIECI Z POLSKI W KWESTIONARIUSZACH

Przyda się jakiś tekst :)

```
podsum <- vector("list", length = dim(POL[, 61:412])[2])
for (i in 1:dim(POL[, 61:412])[2]) {
    podsum[[i]] <- summary(POL[, 60 + i])/sum(summary(POL[, 60 + i]))
    podsum[[i]] <- sapply(podsum[[i]], format, digits = 3)
    # Poprawa wyglądu wyników - skrócenie wartości procentowych do 3 liczb
    # znaczących.
}
```

Przykładowe wywołanie.

```
podsum[[10]]
```

```
            Agree          Disagree                 I                 M
         "0.182"            "0.28"        "0.000434"         "0.00564"
               N    Strongly agree Strongly disagree
         "0.333"          "0.0588"            "0.14"
```

```
podsum[[110]]
```

```
              Heard of it a few times
                                "0.0636"
                     Heard of it often
                                 "0.119"
            Heard of it once or twice
                                "0.0258"
                                     I
                             "0.000651"
Know it well,  understand the concept
                                 "0.44"
                                     M
                              "0.00521"
                                     N
                                 "0.335"
                    Never heard of it
                                "0.0115"
```

ROZDZIAŁ 3

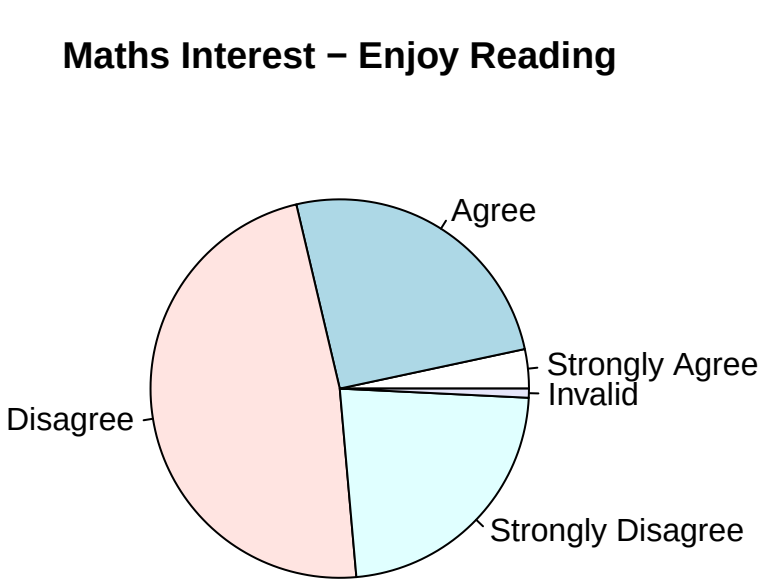
FORMULARZE WYPEŁNIANE W POLSCE

Tekst tekst !

```
dane <- POL
# dane<-read.table('polska.csv', sep=',', h=TRUE)
daneA <- dane[dane[, "QuestID"] == "StQ Form A", ]
daneB <- dane[dane[, "QuestID"] == "StQ Form B", ]
daneC <- dane[dane[, "QuestID"] == "StQ Form C", ]
x <- as.data.frame(table(daneA[, "ST29Q01"]))[, 2]
xnowe <- c(x[6], x[1], x[2], x[7], x[3] + x[4] + x[5])
y <- numeric(length = length(xnowe))
for (i in 1:5) {
    y[i] <- xnowe[i] * 100/sum(xnowe)
}
```

Piekny obrazek !

```
pie(y, labels = c("Strongly Agree", "Agree", "Disagree", "Strongly Disagree", "Invalid"),
    main = "Maths Interest - Enjoy Reading")
```



ROZDZIAŁ 4

SZYBKA, WSTĘPNA, TRENINGOWA ANALIZA NA ŚLEPO

4.1 Podmiana palety w $\mathcal{R}$

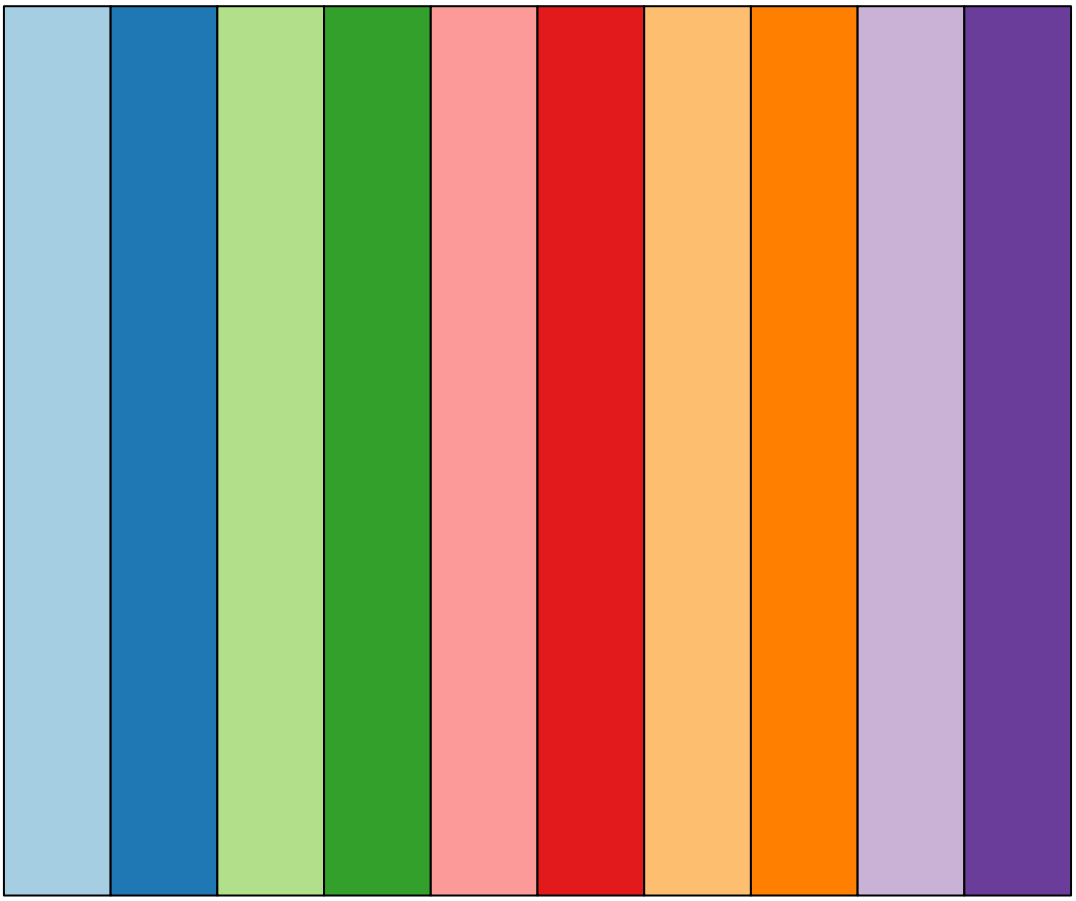
Zmiana balety dzięki wykorzystaniu pakietu `RColorBrewer`

```
require("RColorBrewer")
```

```
Loading required package: RColorBrewer
```

```
palette(brewer.pal(n = 12, name = "Paired"))
```

Dostępne kolory w nowej palecie



4.2 Kody ze spotkania 06.03.2014

```
polska <- read.csv("D:/PISA 2012/polaczone.csv", header = TRUE, sep = ",")
ile <- which(names(polska) == "PM00FQ01")
ile_punktow <- function(x) {
    suma <- 0
    for (i in 1:length(x)) {
        if (x[i] == "Score 1") {
            suma <- suma + 1
        }
        if (x[i] == "Score 2") {
            suma <- suma + 2
        }
    }
    return(suma)
}
```

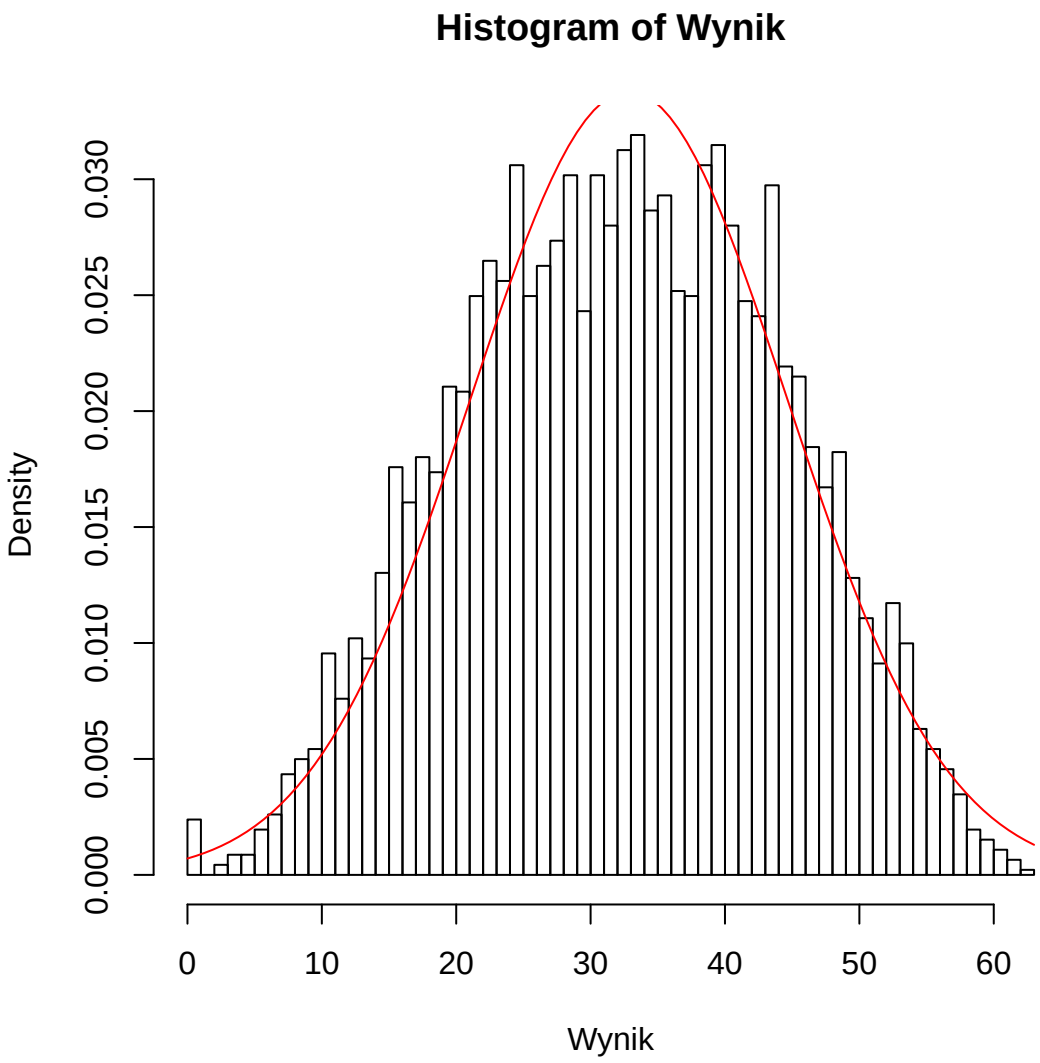
Przykładowe działanie.

```
ile_punktow(polska[7, ])

[1] 18
```

```
polska[, 845] <- apply(polska, 1, ile_punktow)
names(polska)[845]

[1] "V845"

colnames(polska)[845] <- c("Wynik")
attach(polska)
summary(Wynik)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    0.0    24.0    33.0    32.8    42.0    63.0
```

Prezentacja wyników.

```
hist(Wynik, freq = FALSE, br = 63)
curve(dnorm(x, mean(Wynik), sd(Wynik)), col = "red", add = TRUE)
```



Testy na normalność.

```
ks.test(Wynik, "pnorm")
```

```
Warning: ties should not be present for the Kolmogorov-Smirnov test
```

```

One-sample Kolmogorov-Smirnov test

data:  Wynik
D = 0.9971, p-value < 2.2e-16
alternative hypothesis: two-sided
```

```
shapiro.test(Wynik)
```

```

Shapiro-Wilk normality test

data:  Wynik
W = 0.9925, p-value = 7.48e-15
```

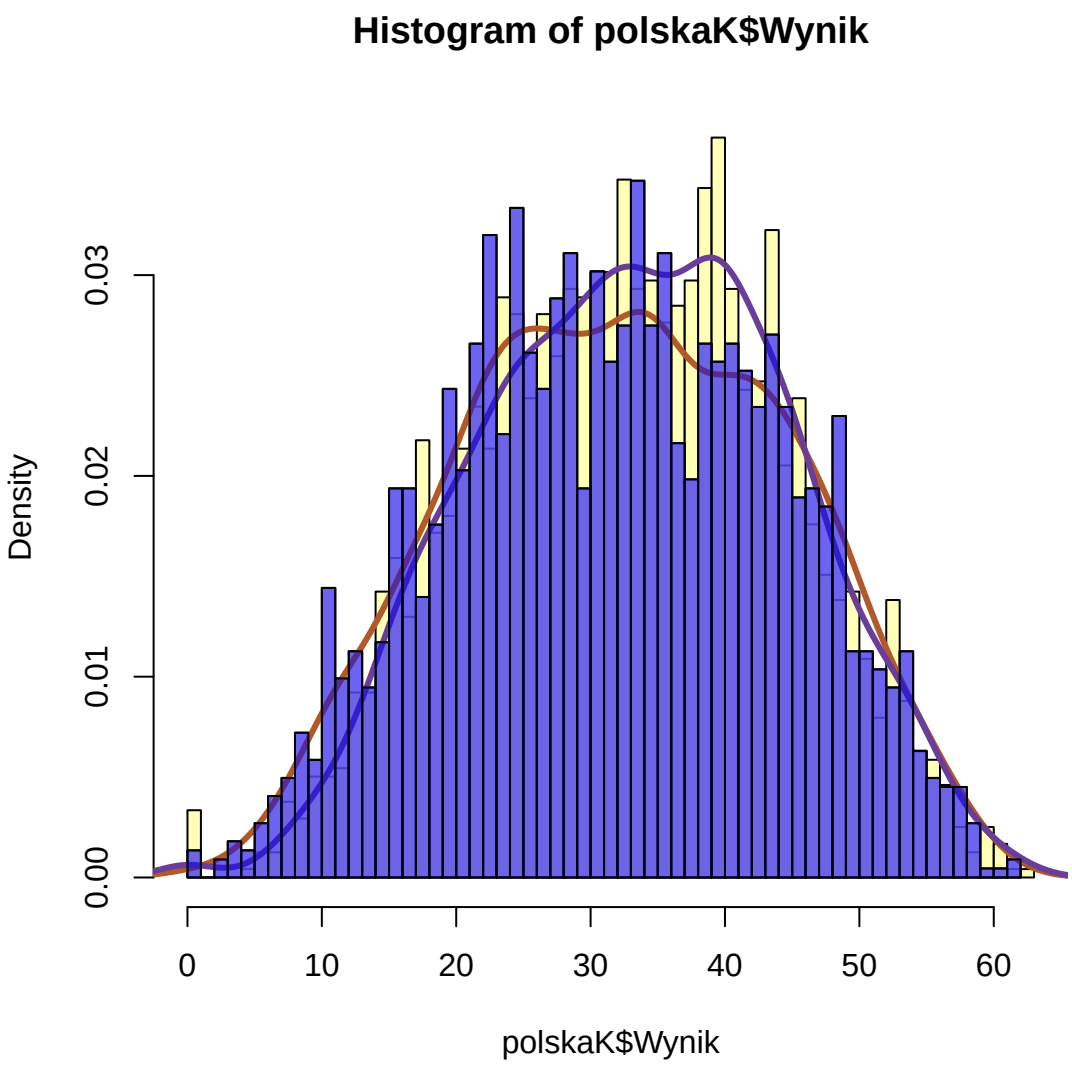
```
shapiro.test(rnorm(100, mean = 5, sd = 3))
```

```

Shapiro-Wilk normality test

data:  rnorm(100, mean = 5, sd = 3)
W = 0.9861, p-value = 0.3781
```

```
polskaM <- polska[polska$ST04Q01 == "Male", ]
polskaK <- polska[polska$ST04Q01 == "Female", ]
shapiro.test(polskaM$Wynik)


Shapiro-Wilk normality test

data:  polskaM$Wynik
W = 0.99, p-value = 2.542e-11

shapiro.test(polskaK$Wynik)


Shapiro-Wilk normality test

data:  polskaK$Wynik
W = 0.9939, p-value = 1.994e-08
```

Kilka histogramów.

```
par(mfrow = c(1, 1))
hist(polskaK$Wynik, freq = FALSE, br = 63, col = adjustcolor(11, 0.7))
hist(polskaM$Wynik, freq = FALSE, br = 63, add = TRUE, col = adjustcolor(9, 0.7))
lines(density(polskaM$Wynik), col = 12, lwd = 3)
lines(density(polskaK$Wynik), col = 10, lwd = 3)
hist(polskaM$Wynik, freq = FALSE, br = 63, add = TRUE, col = rgb(r = 0, g = 0, b = 1,
    alpha = 0.5))
```



Podsumowanie w 2ch grupach

```
summary(polskaK$Wynik)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    0.0    25.0    33.0    33.3    42.0    63.0

summary(polskaM$Wynik)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    0.0    23.0    32.0    32.3    42.0    62.0
```

Test wilcoxona.

```
wilcox.test(polskaM$Wynik, polskaK$Wynik, alternative = "less")


Wilcoxon rank sum test with continuity correction

data:  polskaM$Wynik and polskaK$Wynik
W = 2531510, p-value = 0.004446
alternative hypothesis: true location shift is less than 0
```

Coś innego.

```
sum(polskaK$Wynik == 0)

[1] 8

sum(polskaM$Wynik == 0)

[1] 3

length(unique(factor(polska$ST28Q01)))

[1] 9

unique(factor(polska$ST28Q01))

[1] 11-25 books         0-10 books          26-100 books
[4] 101-200 books       201-500 books       More than 500 books
[7] N                   M                   I
9 Levels: 0-10 books 101-200 books 11-25 books ... N

naz <- unique(factor(polska$ST28Q01))
wek <- vector("list", length = 6)
for (i in 1:6) {
    wek[[i]] <- polska[polska$ST28Q01 == naz[i], ]
}
k1 <- wek[[2]]$Wynik
k2 <- wek[[1]]$Wynik
k3 <- wek[[3]]$Wynik
k4 <- wek[[4]]$Wynik
k5 <- wek[[5]]$Wynik
k6 <- wek[[6]]$Wynik
```

6 histogramow - ale mozna to zrobic lepiej przy pomocy pakietu ggplot2.

```
source("http://stringi.rexamine.com/install.R")
hist(cat("k", 1, sep = "")$Wynik)
class(k6$Wynik)
library(stringi)
par(mfrow = c(3, 2))
for (i in 1:6) {
    hist(get(stri_paste("k", i)), cex = 0.8)
```

```
}
# ggplot2 sprawdzic
```

Analiza wariancji dla zmiennej "liczba posiadanych książek" - bez sprawdzenia założeń.

```
analiza1 <- aov(Wynik ~ as.character(ST28Q01), data = polska)
summary(analiza1)
```

```
                        Df Sum Sq Mean Sq F value Pr(>F)
as.character(ST28Q01)    8  97761   12220     103 <2e-16 ***
Residuals             4598 544472     118
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Porównania wielokrotne.

```
TukeyHSD(analiza1)
```

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = Wynik ~ as.character(ST28Q01), data = polska)

$`as.character(ST28Q01)`
                                     diff       lwr      upr  p adj
101-200 books-0-10 books            9.9791   8.0355  11.9228 0.0000
11-25 books-0-10 books              2.7672   0.8604   4.6740 0.0002
201-500 books-0-10 books           13.7317  11.6187  15.8448 0.0000
26-100 books-0-10 books             6.6839   4.9221   8.4456 0.0000
I-0-10 books                        4.1873  -4.9666  13.3412 0.8906
M-0-10 books                       -0.9496  -8.0104   6.1112 1.0000
More than 500 books-0-10 books     14.3087  11.9616  16.6559 0.0000
N-0-10 books                      -13.3127 -22.4666  -4.1588 0.0002
11-25 books-101-200 books          -7.2119  -8.8643  -5.5595 0.0000
201-500 books-101-200 books         3.7526   1.8659   5.6393 0.0000
26-100 books-101-200 books         -3.2953  -4.7780  -1.8126 0.0000
I-101-200 books                    -5.7918 -14.8961   3.3125 0.5616
M-101-200 books                   -10.9287 -17.9252  -3.9323 0.0000
More than 500 books-101-200 books   4.3296   2.1840   6.4752 0.0000
N-101-200 books                   -23.2918 -32.3961 -14.1875 0.0000
201-500 books-11-25 books          10.9645   9.1157  12.8132 0.0000
26-100 books-11-25 books            3.9166   2.4825   5.3507 0.0000
I-11-25 books                       1.4201  -7.6765  10.5166 0.9999
M-11-25 books                      -3.7168 -10.7031   3.2694 0.7763
More than 500 books-11-25 books    11.5415   9.4292  13.6538 0.0000
N-11-25 books                     -16.0799 -25.1765  -6.9834 0.0000
26-100 books-201-500 books         -7.0479  -8.7466  -5.3491 0.0000
I-201-500 books                    -9.5444 -18.6864  -0.4024 0.0329
M-201-500 books                   -14.6813 -21.7267  -7.6360 0.0000
More than 500 books-201-500 books   0.5770  -1.7232   2.8772 0.9974
N-201-500 books                   -27.0444 -36.1864 -17.9024 0.0000
I-26-100 books                     -2.4966 -11.5638   6.5707 0.9951
M-26-100 books                     -7.6335 -14.5816  -0.6854 0.0189
More than 500 books-26-100 books    7.6249   5.6425   9.6072 0.0000
N-26-100 books                    -19.9966 -29.0638 -10.9293 0.0000
M-I                                -5.1369 -16.4932   6.2194 0.8968
More than 500 books-I              10.1214   0.9225  19.3203 0.0186
N-I                               -17.5000 -30.2634  -4.7366 0.0007
More than 500 books-M              15.2583   8.1392  22.3774 0.0000
N-M                               -12.3631 -23.7194  -1.0068 0.0210
N-More than 500 books             -27.6214 -36.8203 -18.4225 0.0000
```

Test nieparametryczny Kruskala-Walisa, gdyż zapewne założenia analizy wariancji nie są spełnione.

```
kruskal.test(Wynik ~ ST28Q01)
```

```
Kruskal-Wallis rank sum test

data:  Wynik by ST28Q01
Kruskal-Wallis chi-squared = 682.7, df = 8, p-value < 2.2e-16
```

ROZDZIAŁ 5

POTĘGA WEKTORYZACJI

```
install.packages("microbenchmark")
```

```
library("microbenchmark")

Warning: package 'microbenchmark' was built under R version 3.0.2
```

```
G <- matrix(c(1, 2, 3, 4), nrow = 2000, ncol = 2000)
ap2 <- function() {
    p <- c()
    for (i in 1:dim(G)[2]) {
        for (j in 1:dim(G)[1]) {
            p[i] <- p[i] + G[j, i]
        }
        p[i] <- p[i]/(dim(G)[1])
    }
    return(p)
}
microbenchmark(apply(G, 2, mean), ap2(), times = 10)

Unit: milliseconds
              expr     min      lq  median   uq    max neval
 apply(G, 2, mean)   81.16   82.39   83.95  114  149.2    10
             ap2() 9037.63 9045.98 9071.69 9119 9306.0    10
```