Analiza Danych
Podstawy Statystyczne
(ADPS)

Laboratorium 1

Organizacja zajęć

- 4 bloki zajęć laboratoryjnych (2 x 1,5h każdy).
- Każde zajęcia podlegają ocenie (0-5 pkt).
- Sala 10 lub CS102.
- Adresy e-mail prowadzących:

  K.Jedrzejewski.1@elka.pw.edu.pl

  M.Rupniewski@elka.pw.edu.pl

Pakiet R

- R jest pakietem (środowiskiem) i językiem programowania przeznaczonym do zaawansowanych obliczeń statystycznych.
- Bezpłatny.
- Platformy: Windows, Linux, Unix, MacOS.
- Język R jest językiem interpretowanym, a nie kompilowanym.
- Źródła: http://www.r-project.org
- Pakiety.
- CRAN (*Comprehensive R Archive Network*).

RStudio

- Zintegrowane środowisko programistyczne (IDE) do R, zawierające m.in.
  - konsolę,
  - edytor wspomagający pisanie kodu (syntax-highlighting) i debuggowanie kodu,
  - okno podglądu danych i historii,
  - okno pomocy, podglądu plików, zarządzania pakietami R, wyświetlania rysunków.
- Źródła: https://www.rstudio.com

RStudio

RStudio

r2d2.ii.pw.edu.pl:8787

File Edit Code View Plots Session Build Debug Tools Help

kjedrze1

Project: (None)

Lab1.R × df_KGHM ×

Source on Save | Run | Source

```
#########################################
# Przykład 1
#########################################

rm(list = ls()) # kasowanie danych

cat("\014") # czyszczenie konsoli

#dir.create('~/ADPS')
#dir.create('~/ADPS/Lab1')
setwd('~/ADPS/Lab1')


unzip('mstall.zip', 'KGHM.mst')
df_KGHM = read.csv('KGHM.mst')

names(df_KGHM) = c('ticker', 'date', 'open', 'high', 'low', 'close', 'vol')

#Wykres kursu otwarcia w zależności od daty:
  df_KGHM$date = as.Date.character(df_KGHM$date, format = '%Y%m%d')
plot(open ~ date, df_KGHM,
     type = 'l', col = 'blue',
     xlab = 'Data', ylab = 'Kurs otwarcia [zł]',
     main = 'KGHM' )
grid()

#Wykres procentowych zmian kursu otwarcia:
  df_KGHM$open_ch= with(df_KGHM, c(NA, 100*diff(open)/ open[1:length(open) -1]))
plot(open_ch ~ date, df_KGHM,
     type = 'l', col = 'blue',
     xlab = 'Data', ylab = 'Procentowa zmiana kursu otwarcia [%]',
     main = 'KGHM' )
grid()
```

10:1 (Top Level) R Script

Console ~/ADPS/Lab1/

```
> #dir.create('~/ADPS')
> #dir.create('~/ADPS/Lab1')
> setwd('~/ADPS/Lab1')
>
>
> unzip('mstall.zip', 'KGHM.mst')
> df_KGHM = read.csv('KGHM.mst')
>
> names(df_KGHM) = c('ticker', 'date', 'open', 'high', 'low', 'close', 'vol')
>
> #Wykres kursu otwarcia w zależności od daty:
>   df_KGHM$date = as.Date.character(df_KGHM$date, format = '%Y%m%d')
> plot(open ~ date, df_KGHM,
+      type = 'l', col = 'blue',
+      xlab = 'Data', ylab = 'Kurs otwarcia [zł]',
+      main = 'KGHM' )
> grid()
>
```

Environment History

Import Dataset | List

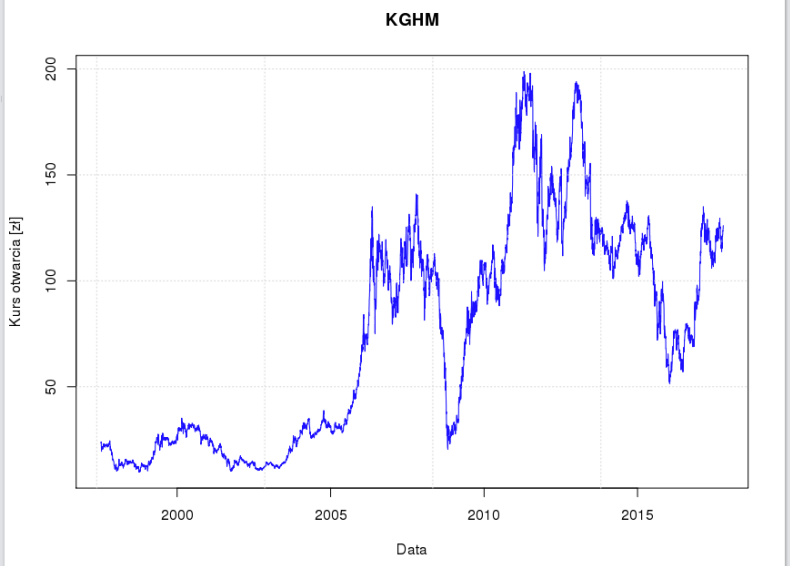
Global Environment

Data

df_KGHM	5078 obs. of 7 variables

Files Plots Packages Help Viewer

Zoom | Export | Publish



Środowisko

- Pomoc
  - **?**nazwa_funkcji, **help**(nazwa_funkcji) - np. ?c, help(c)
  - **??**fraza, np. ??predict
  - example(nazwa_funkcji), np. example(plot)
  - zakładka Help w RStudio
- Wywołanie funkcji
  - nazwa_funkcji(arg1, arg2, arg3 = wartość)
- Znak komentarza **#**
- Przydatne funkcje
  - **ls**(), **rm**(), rm(list = ls()), **print**('napis'), **print**(dane)
- Katalog roboczy
  - **getwd**(), **setwd**('nazwa_katalogu'), **dir**()

Typy danych

- Numeryczny
  - 2.345
  - 3.5e-15
- Znakowy
  - ’a’, 'abc', \n, \t
- Logiczny
  - TRUE\T, FALSE\F
- Zespolony
  - x = 2 + 3i **Mod**(x), **Arg**(x), **Re**(x), **Im**(x)

Struktury danych

- Wektor (ang. *vector*)
- Czynnik (ang. *factor*)
- Ramka (ang. *data frame*)
- Lista (ang. *list*)
- Tablica (ang. *array*)

Wektor

- Tworzenie
  - x = **c**(1,4,6,-3), x <- **c**(1,5,6,-3), x = **c**('bdb', 'db', 'dst', 'bdb')
  - x = **c**(TRUE,FALSE,TRUE, TRUE) ; x = c(T,F,T,T)
- Indeksowanie wektorów
  - y = x[3], y = x[2:4], y = x[**c**(1,3)]
- Operator :
  - x = 1:10, x = 10:1
- seq(), rep()
  - x = **seq**(0, 5, **by** = 0.25), x = **seq**(5, 0.5, **length** = 10)
  - x = **rep**(**c**(1,2,3), 4), x = **rep**(**c**(1,2,3), **each** = 4)
- Deklarowanie
  - x = c(), w = vector('logical', 10)

Czynnik

- Czynnik jest strukturą przechowującą oprócz szeregu danych informacje o powtórzeniach takich samych wartości oraz zbiorze unikalnych wartości.
- factor()
  - faktor1 = **factor**(c(2,3,4), levels = 1:5)
  - oceny = **factor**(**c**('bdb', 'dst', 'db', 'dst', 'bdb', 'ndst', 'db'))
- levels()
  - **levels**(oceny)
- table()
  - **table**(oceny)

Tablica

- Tablica jest wektorem zawierającym dodatkowe dane określające uporządkowanie elementów w tablicy.
- Tworzenie tablic
  - x = 1:20; **dim**(x) = c(4,5)
  - x = **matrix**(1:20, 4, 5)
  - x = **array**(1:20, c(4,5))
- Nazwy kolumn/wierszy
  - **dimnames**(x) = **list**(letters[1:4], LETTERS[1:5])
- **attributes**(x), **summary**(x)
- Łączenie macierzy/wektorów rbind(), cbind()
  - x= **rbind**(1:3, 4:6); y = **cbind**(1:3, 4:6)
- Indeksowanie
  - x[2, 3], x[1:2, 2:3], x[2,], x[,3]

Lista

- Lista (list) jest uporządkowanym zbiorem elementów różnego typu.
  - Lista = **list**('Jan', 'Kowalski', 1990, 'Warszawa', 'TRUE')
  - Lista = **list**(imie = 'Jan', nazwisko = 'Kowalski', rok_ur = 1990, zam = 'Warszawa', stud = 'TRUE')
- Wybór z listy
  - Lista$nazwisko
- Dodawanie
  - Lista$imie[2] = 'Jakub'; Lista$nazwisko[2] = 'Nowak'; …
  - Lista2 = **list**(imie = c('Jan','Piotr'), nazwisko = c('Kowalski', 'Nowak'), rok_ur = c(1991,1995), zam = c('Warszawa','Poznan'), stud = c(T,F))
  - Lista2$rok_ur[1]
  - Lista2[[4]][2]

Ramka

- Ramka (data frame) jest macierzą, w której poszczególne kolumny mogą zawierać wartości różnego typu.
  - ramka = **data.frame**(LETTERS[1:6], seq(10, 60, by = 10), seq(10, 60, by = 10) > 35)
  - **names**(ramka) = **c**('Litera', 'Punkty', 'Punkty > 35')
- Wybór z ramki
  - ramka[3,]; ramka[,2]
  - ramka$Punkty
  - ramka$Litera[2]
  - ramka$'Punkty > 35'

Wykresy

- Podstawowe funkcje
  - x = seq(0, 2*pi, by = 0.01); y = sin(x)
  - **plot**(x, y, type = 'l', xlab = 'opis x', ylab = 'opis y', main = 'tytul')
  - **plot**(y ~ x, type = 'l')
  - **plot**(kolumna_1 ~ kolumna3, nazwa_ramki)
  - **points**(), **lines**() - kolejne dane/linie na wykresie
  - **hist**(**rnorm**(1000))
  - **pie**(1:6, labels = LETTERS[1:6])
- Pożyteczne funkcje
  - **grid**()
  - **legend**()

Programowanie w R

- Instrukcja warunkowa
- Pętle
- Funkcje
- Skrypty

Instrukcja warunkowa

- **if**(warunek) {wyrażenie}
  - x = 3; y = 2
  - if(x>y) {
    paste('x =', x, 'jest wieksze od y =', y) }
- **if**(warunek) {wyrażenie1} **else** {wyrażenie2}
- **ifelse**(warunek, yes, no)
  - ifelse(x > y,
    { paste('x =', x, 'jest wieksze od y =', y) },
    { paste('x =', x, 'jest mniejsze lub rowne y =', y) })
- **switch**()
- Operatory logiczne:
  - &, |, !, xor(x,y), ==, !=, <, >, <=, <=, isTRUE(x), &&, ||.

Pętle

- **for**(zmienna **in** zbiór_wartości) {wyrażenie}
  - for(k in 1:10) {
  - print(k)
  - }
- **while**(warunek) {wyrażenie}
- **repeat** {wyrażenie}
- **break**, **next**

- **with**(dane, wyrażenie, …)
- **apply**(), **replicate**()

Funkcje

- nazwa_funkcji = **function**(arg1, arg2, arg3 = wartość) {ciało funkcji}
- Zwracane wartości - ostatnia linia
- **return**()
- **stop**()
- **warning**()

Wczytywanie i zapisywanie danych

- Pobieranie z plików i zapisywanie:
  - dane = **scan**('c:/plik.txt')
  - dane = **read.table**('plik.txt', **header** = T)
  - dane = **read.csv**('Zeszyt1.csv', **sep** = '; ', **header** = T, **dec** = ',')
  - **write**(x, 'plik.txt')
  - **write.table**(dane, **file** = 'plik.txt'), **write.csv**()
- Zmiana nazw kolumn, wierszy
  - **names**(dane), **colnames**(dane), **rownames**(dane)
- Edycja/zmiana danych
  - **edit**(dane)
  - mechanizmy Rstudio.

Przykład 1 – przygotowanie katalogu

- W katalogu domowym utwórz katalog *ADPS/Lab1*
  - dir.create('~/ADPS')
  - dir.create('~/ADPS/Lab1')
- Zmień katalog roboczy w Rstudio na *ADPS/Lab1*
  - setwd('~/ADPS/Lab1')
- Powyższe operacje można wykonać korzystając z zakładki *Files*
  - *New Folder*
  - *More -> Set As Working Directory*

Przykład 1 – pobranie danych

- Znajdź i pobierz dane historyczne spółek giełdowych z portalu www.bossa.pl
  - zakładka *Notowania & wykresy*
  - *Dane do programów AT -> Metastock -> Wszystkie grupy GPW -> baza danych w formacie tekstowym -> mstall.zip*
- Zapisz plik *mstall.zip* do katalogu *ADPS/Lab1*.
- Rozpakuj w tym katalogu dane spółki KGHM
  - unzip('mstall.zip', 'KGHM.mst')

Przykład 1 – wczytanie danych

- Wczytaj dane z pliku KGHM.mst do środowiska R
  - df_KGHM = read.csv('KGHM.mst')
- Obejrzyj dane dotyczące spółki KGHM, zwróć uwagę na nazwy kolumn.
- Zmień nazwy kolumn:
  - names(df_KGHM) = c('ticker', 'date', 'open', 'high', 'low', 'close', 'vol')

Przykład 1 – wykres kursu w czasie

- Wykres kursu otwarcia w zależności od daty:
  - df_KGHM$date = as.Date.character(df_KGHM$date, format = '%Y%m%d')
  - plot(open ~ date, df_KGHM,

    type = 'l', col = 'blue',

    xlab = 'Data', ylab = 'Kurs otwarcia [zł]',

    main = 'KGHM')
  - grid()

Przykład 1 – procentowe zmiany kursu

- Wykres procentowych zmian kursu otwarcia:
  - df_KGHM$open_ch= with(df_KGHM,
    c(NA, 100*diff(open)/open[1:length(open) -1]))
  - plot(open_ch ~ date, df_KGHM,

    type = 'l', col = 'blue',

    xlab = 'Data', ylab = 'Procentowa zmiana kursu otwarcia [%]',

    main = 'KGHM')
  - grid()

Przykład 1 – histogram

- Histogram procentowych zmian kursu otwarcia:
  - hist(df_KGHM$open_ch,
    breaks = 50, prob = T,
    xlab = 'Zmiana kursu otwarcia [%] ',
    ylab = 'Częstość występowania',
    main = 'Histogram procentowych zmian kursu KGHM')
  - grid()

Przykład 1 – FGP

- Wartość średnia oraz odchylenie standardowe zmian kursu otwarcia:
  - m = mean(df_KGHM$open_ch, na.rm = T)
  - s = sd(df_KGHM$open_ch, na.rm = T)
- Dorysowanie do histogramu wykresu gęstości rozkładu normalnego o obliczonych parametrach:
  - curve(dnorm(x, mean = m, sd = s), add = T, col = 'red', -10, 10)

Przykład 1 – wykres pudełkowy

- Wykres pudełkowy procentowych zmian kursu otwarcia:
  - boxplot(df_KGHM $open_ch,

    col = 'green',

    xlab = 'KGHM', ylab = 'Zmiana kursu otwarcia [%] ',

    main = 'KGHM')
  - grid()

Przykład 2 – pobranie danych

- Znajdź i pobierz dane historyczne dotyczące katastrof lotniczych ze strony https://opendata.socrata.com:
  - https://opendata.socrata.com/Government/Airplane-Crashes-and-Fatalities-Since-1908/q2te-8cvq
- Zapisz plik *Airplane_Crashes_and_Fatalities_Since_1908.csv* do katalogu *ADPS/Lab1*
- Wczytaj dane do środowiska R:
  - kat = read.csv('Airplane_Crashes_and_Fatalities_Since_1908.csv')
- Obejrzyj dane, zwróć uwagę na puste pola.

Przykład 2 – liczba wypadków w latach

- Dodanie do danych kolumny z rokiem:
  - kat$Year = strftime(as.Date(kat$Date, '%m/%d/%Y'), '%Y')
- Wykres liczby wypadków w danym roku:
  - plot(table(kat$Year),

    type = 'h', col = 'blue',

    xlab = 'Rok', ylab = 'Liczba katastrof',

    main = 'Liczba katastrof w roku')
  - grid()

Przykład 2 – liczba ofiar w latach

- Agregacja danych po latach:
  - Ofiary_agr = aggregate(Fatalities ~ Year, kat, FUN = sum)
- Wykres:
  - plot(Ofiary_agr,

    type = 'h', col = 'blue',

    xlab = 'Rok', ylab = 'Lczba ofiar',

    main = 'Liczba ofiar katastrof w roku')
  - grid()

Generowanie liczb losowych

- Funkcje dot. rozkładów, przedrostki: d, p, q, r:
  - **d*rozkl***() – funkcja gęstości prawdop. dla rozkładu ***rozkl***,
  - **p*rozkl*** () – dystrybuanta,
  - **q*rozkl*** () – kwantyl / dystrybuanta odwrotna,
  - **r*rozkl*** () – generator liczb pseudolosowych,
  - ***rozkl*** = np. binom, pois, geom, unif, norm, exp, chisq, t, beta, gamma,….
- Podstawowe wskaźniki:
  - **mean**(), **sd**(), **var**(), **median**(), **quantile**()
- Empiryczna dystrybuanta: **ecdf**()
- Losowanie ze zbioru - **sample**()
  - **sample**(1:6, 20, replace = T)

Przykład 3

- Generacja *1000* próbek z rozkładu normalnego *N(2,9)*
  - proba = rnorm(1000, mean = 2, sd = 3)
- Wartości parametrów z próby
  - m = mean(proba); s = sd(proba)
- Histogram i gęstość prawdopodobieństwa
  - hist(proba, breaks = 20, prob = T)
  - curve(dnorm(x, mean = 2, sd = 3), add = T, col = 'red', -15, 15)
  - grid()

Przykład 3

- Dystrybuanta empiryczna i teoretyczna:
  - plot(ecdf(proba))
  - curve(pnorm(x, mean = 2, sd = 2), add = T, col = 'red', -15, 15)
- Wykres pudełkowy:
  - boxplot(proba)
  - grid()
- Teoretyczne i empiryczne wartości kwantyli dla *0.25*, *0.5* i *0.75*.
  - qnorm(c(0.25, 0.5, 0.75), mean = 2, sd = 3)
  - quantile(proba, c(0.25, 0.5, 0.75))

Zadanie 1

- Dla wybranych dwóch spółek
  - sporządź wykresy procentowych zmian kursów zamknięcia w zależności od daty,
  - wykreśl i porównaj histogramy procentowych zmian kursów zamknięcia,
  - wykonaj jeden wspólny rysunek z wykresami pudełkowymi zmian kursów zamknięcia.

Zadanie 2

- Sporządź wykres liczby katastrof lotniczych w poszczególnych
  - miesiącach,
  - dniach,
  - dniach tygodnia (*weekdays()*).
- Narysuj jak w kolejnych latach zmieniały się:
  - liczba osób, które przeżyły katastrofy,
  - odsetek osób (w procentach), które przeżyły katastrofy.

Zadanie 3

- Dla 2 różnych zestawów parametrów rozkładu dwumianowego:
  - *Binom(20,0.2)*,
  - *Binom(20, 0.8)*

  narysuj funkcje gęstości prawdopodobieństwa (zastosuj opcję *type* = *'h'* w funkcji *plot*) i dystrybuanty (zastosuj opcję *type* = *'h'* w funkcji *plot*).
- Dla rozkładu dwumianowego *Binom(20, 0.8)* wygeneruj trzy próby losowe składające się z *M = 100, 1000 i 10000* próbek. Dla poszczególnych prób wykreśl empiryczne funkcje gęstości prawdopodobieństwa i dystrybuanty. Oblicz empiryczne wartości średnie i wariancje. Porównaj je z wartościami teoretycznymi.

Zadanie 4

- Wygeneruj *K* = *500* realizacji (powtórzeń) prób losowych o długości *M* = *100* z rozkładu *Binom(20, 0.8).*

  Dla wszystkich realizacji oblicz wartości średnie i wariancje. Następnie narysuj histogramy wartości średnich i wariancji.

  Porównaj otrzymane wyniki z wartościami teoretycznymi.

  Powtórz eksperymenty dla *M* = *1000* i *M* = *10000*.

  Wskazówka:

  mm = replicate(500, mean(rbinom(M, 20, 0.8)))

Dziękuję za uwagę!

Pytania ?