R

`#---1.Wygeneruj liczby całkowite od 10 do 20 i zachowaj je w wektorze x

x = 10:20

x

#---2.Wygeneruj 4 powtórzenia sekwencji liczb (3, 5 ,7).

rep(c(3,5,7),4)

#\_lub

x = c(3,5,7)

rep(x,4)

seq(1, 9, by = 2)

#---3.Wygeneruj sekwencję zawierającą osiem czwórek, następnie

#siedem szóstek i na koniec dziewięć

#trójek. Zapisz ją w macierzy M o sześciu wierszach i czterech kolumnach.

?matrix()

l = c(rep(4,8),rep(6,7),rep(3,9))

l

m = matrix(l,nrow=7)

m

#---4.Wygeneruj wszystkie liczby nieparzyste z przedziału [1, 100].

seq(1, 100, by = 2)

#---5.Utwórz wektor zawierający jedną jedynkę,

#dwie dwójki,..., dziewięć dziewiątek.

rep(c(1,2,3,4,5,6,7,8,9),c(1,2,3,4,5,6,7,8,9))

#---6.Skonstruuj wektor x używając poniższego kodu:

x <- c(NA, 3, 14, NA, 33, 17, NA, 41)

x

#• Zlicz liczbę braków.

sum(is.na(x))

#• Wyznacz średnią arytmetyczną nie biorąc braków pod uwagę.

mean(x)

mean(x,na.rm = TRUE)

#• Usuń braki w danych.

x<-x[!is.na(x)]

x

#• Zastąp braki liczbą 11.

x[is.na(x)] =11

x

#---7.Utwórz ramkę danych zawierającą

#jedynie klientów o dodatnich dochodach (INCOME) i

#ujemnej całkowitej wartości

#netto (NETWORTH). Ilu jest takich klientów?

library("UsingR")

cfb

head(cfb)

attach(cfb)

data.frame(cfb)

a = cfb$INCOME>0

sum(a)

b = cfb$NETWORTH <0

sum(b)

data.frame(a)

data.frame(b)

#---8.Sprawdź, które zmienne w zbiorze danych Cars93 (sprzedaż samochodów w USA w roku

#1993) z pakietu MASS są czynnikami. Wyznacz w postaci tabeli liczbę samochodów dla miejsca

#pochodzenia (zmienna Origin) oraz rodzaju samochodu (zmienna Type).

library("MASS")

Cars93

colnames(Cars93)

factor(Cars93)

is.factor("Make")

data.frame(Cars93)

?table()

sapply(Cars93,is.factor)

attach(Cars93)

table(Origin,Type)

#---9.Dla zbioru danych Cars93 wyświetl informacje

#funkcją summary dla każdego poziomu zmiennej Type.

summary(Cars93)

sapply(split(Cars93,Cars93$Type),summary)

#---10.. Dla zbioru danych Insurance z pakietu MASS, zawierającego informacje

#na temat roszczeń ubezpieczeniowych samochodów w 1973 roku, skonstruuj

#tabelę zawierającą liczbę roszczeń

#(zmienna Claims) z podziałem na wiek (zmienna Age) oraz typ samochodu (zmienna Group)

Insurance

attach(Insurance)

tapply(Insurance$Claims,list(Insurance$Group,Insurance$Age),sum)

with(Insurance,tapply(Claims,list(Group,Age)),sum)

#---11.Ze zbioru mtcars utwórz ramkę mtcars6, która zawiera informacje jedynie o samochodach

#z 6 cylindrami.

mtcars

mtcars[mtcars$cyl==6,]

#---12.Ze zbioru Cars93 z pakietu MASS, utwórz ramkę samochody,

#która zawiera informacje jedynie

#o małych oraz sportowych samochodach.

filter <- Cars93$Type == 'Small'| Cars93$Type == 'Sporty'

Cars93[filter,]

filter

#---13. Zastosuj funkcję sapply do funkcji is.factor na każdej kolumnie zbioru tinting (wpływ

#zabrudzenia szyb na jakość widzenia) z pakietu DAAG. Dla każdej kolumny będącej czynnikiem

#określ poziomy. Które czynniki są uporządkowane (is.ordered)?

library("DAAG")

tinting

sapply(tinting[which(sapply(tinting,is.factor)==TRUE)],levels)

list1

#---14. Wykonaj poniższe polecenie tworzące listę list1:

#list1 <- list(observationA = c(1:5, 7:3), observationB = matrix(1:6, nrow = 2 ))

#Korzystając z grupy funkcji apply() wyznacz liczbę unikatowych wartości w każdym elemencie

#listy.

list1 <- list(observationA = c(1:5, 7:3), observationB = matrix(1:6, nrow = 2 ))

sapply(list1, unique)

sapply(list1, length)

#---15.Wprowadź dane do R poleceniem

x <- c(1, 8, 2, 6, 3, 8, 5, 5, 5, 5)

#Oblicz:

#• sumę wszystkich elementów,

sum(x)

#• logarytm dziesiętny wszystkich elementów,

log10(x)

#• różnicę pomiędzy największym i najmniejszym elementem wektora x.

max(x)

min(x)

różnica = max(x)-min(x)

różnica

#---16. Skonstruuj macierz diagonalną o wymiarze

#4 o elementach 4, 1, 2, 3 na przekątnej.

diagonal<-c(4,1,23)

diag(diagonal)

#---17.Utwórz następującą macierz

#oraz oblicz jej transpozycję i odwrotność.

#Pomnóż macierz przez jej odwrotność.

m = rbind(c(1,2,3),c(4,2,1),c(2,3,0))

m

inv=solve(m)

inv%\*%m

#---18. Walec o wysokości h oraz promieniu podstawy r ma objętość V = πr2h oraz pole powierzchni

P = 2πr(r + h). Dla długości promienia 1:5 oraz wysokości 4:8 oblicz odpowiednie objętości i

pola powierzchni. Skonstruuj ramkę danych o kolumnach: r, h, V, P.

r = c(1:5)

h = c(4:8)

P = 2\*pi\*r\*(r+h)

V = pi\*r^2\*h

P

V

data <- cbind.date.frame(r,h,P,V)

#---19.Znajdź 8 największych lądów (kontynenty, wyspy)

#świata (zbiór danych islands).

isl = data.frame(sort(island,decreasing = TRUE))

#---20.W pakiecie schoolmath znajduje się zbiór danych primlist, który zawiera liczby pierwsze

#pomiędzy 1 a 9 999 999. Znajdź największą liczbę pierwszą mniejszą od 1000. Ile jest liczb

#pierwszych większych od 100 a mniejszych od 500?

install.packages('schoolmath')

library(schoolmath)

primlist[primlist<1000]

max(primlist[primlist<1000])

length(primlist[primlist>100 & primlist < 500 ])

#---21.Utwórz wektor liczb naturalnych od 1 do 1000, a następnie zamień liczby parzyste na ich

#odwrotności.

vec = c(1:1000)

vec

even = vec[vec %% 2 == 0]

even

inv\_even = even^(-1)

inv\_even

#---22.Utwórz wektor liczb naturalnych od 1 do 1000, a następnie zamień liczby parzyste na ich

#odwrotności.

library(MASS)

hills

rownames(hills[hills$climb ==2100, ])

row.names(hills)[hills$climb == 2100]

#---23.Zmienna ftv w zbiorze danych birthwt, z pakietu MASS, zawiera liczbę wizyt matek u lekarza

#w pierwszym trymestrze ciąży. Przekształć ją do czynnika o trzech poziomach 0, 1 oraz 2 lub

#więcej (użyj funkcji factor oraz levels).

df = data.frame(birthwt)

df

df$ftv[df$ftv > 2] = 2

df$ftv

df$ftv = factor(df$ftv,

levels = c(0,1,2),

labels = c('zero','one','two or more'))

df

#---24W pewnych sytuacjach przydatna może się okazać tzw. kategoryzacja zmiennych, czyli inny

#podział na kategorie niżby wynikał z danych. Wygeneruj 100 obserwacji, które są odpowiedziami

#na pytania ankiety, każda odpowiedź może przyjąć jedną z wartości: ’a’, ’b’, ’c’, ’d’, ’e’. Dokonaj

#kategoryzacji w taki sposób, aby kategoria 1 obejmowała odpowiedzi ’a’ i ’b’, 2 odpowiedzi ’c’ i

#’d’ oraz 3 odpowiedź ’e’.

#Wskazówka: Wykorzystaj funkcję recode z pakietu car

install.packages("car")

library(car)

?recode

?sample

x= sample(c('a','b','c','d','e'), size = 100, replace = TRUE)

x

table(x)

y = recode(x,"c('a','b') = 1;

c('c','d') = 2;

else = 3")

y

table(y)

#---25

df = data.frame(Pima.tr2)

df

sum(is.na(df$bp) == TRUE)

#---26

#########################

######zestaw 3 & 4

#1.1. Dla zbioru danych LakeHuron (poziomy jeziora (w stopach) Huron (lata 1875-1972)):

#• narysuj jak zmieniał się poziom wody w czasie,

#• zastosuj funkcję identify() aby oznaczyć punkty odpowiadające latom o najwyższym

#oraz najniższym poziomie wody.

LakeHuron

plot(LakeHuron)

min\_ind = which(LakeHuron == min(LakeHuron))

max\_ind = which(LakeHuron == max(LakeHuron))

min\_ind

max\_ind

plot(LakeHuron)

identify(c(min\_ind + 1874, max\_ind + 1874),

c(LakeHuron[min\_ind], LakeHuron[max\_ind]),

label = c('min', 'max'))

#---2.W zbiorze danych emissions (emisja CO2 a poziom PKB (26 państw)) z pakietu UsingR istnieje obserwacja odstająca. Znajdź ją używając funkcji identify(), a następnie narysuj wykres,

#bez tej obserwacji. Zamiast punktów powinny być nazwy krajów. Użyj skali logarytmicznej dla

#obu osi.

library('UsingR')

emissions

plot(emissions$CO2 , emissions$GDP)

identify(emissions$CO2,

emissions$GDP,

label = rownames(emissions[1])

)

df = emissions[-1, ]

df

plot(log10(df$CO2), lod10(df$GDP))

text(log10(df$CO2), log10(df$GDP), labelst = rownames(df[1]))

#3.Dla danych 60, 85, 72, 59, 37, 75, 93, 7, 98, 63, 41, 90, 5, 17, 97

#wykonaj wykres typu „łodyga– liście”

stem(c(60, 85, 72, 59, 37, 75, 93, 7, 98, 63, 41, 90, 5, 17, 97), scale = 10)

stem

#4.Zbadaj rozkład długości głów (hdlngth) ze zbioru possum (cechy morfologiczne oposów) z

#pakietu DAAG. Porównaj następujące wykresy: histogram, wykres typu „łodyga – liście” oraz

#wykres gęstości. Jakie są wady i zalety stosowania tych metod wizualizacji danych?

#library (DAAG)

possum

hist(possum$hdlngth, probability = T)

lines(density(possum$hdlngth))

stem(possum$hdlngth)

#5.Dla zbioru chips (grubość chipsów) z pakietu UsingR

#wykonaj wykres pudełkowy w zależności od miejsca pomiarów (powinno być 8 wykresów na jednym diagramie). Co można na ich

podstawie powiedzieć o średniej i wariancji?

#6.Wykonaj wykres pudełkowy (w zależności od żywienia)

#dla zbioru chicken (waga kurczaków

w zależności od metody żywienia) z pakietu UsingR. Co można powiedzieć na temat średnich

wag na podstawie tego wykresu?

boxplot(chips)

chips

mean(chips$wafer11)

var(chips$wafer11)

mean(chips$wafer12)

var(chips$wafer12)

mean(chips$wafer13)

var(chips$wafer13)

**library(tibble)**

**library(tidyr)**

**library(dplyr)**

**airquality**

**airquality %>%**

**select(Temp, Month, Day) %>%**

**as\_tibble() -> data.set**

**view(data.set)**

**wide = pivot\_wider(data.set, names\_form = 'Month', values\_from = 'Temp')**

**wide**

**long = pivot\_longer(wide, cols=2:6,names\_to='Month',values\_to = 'Temp')**

**long**

**?unite**

**day\_month = unite(long, Day, Month, col = 'Date', sep='/')**

**day\_month**

**day\_month\_sep = separate(day\_month, col=Date,**

**into = c('Day','Month'),sep='/')**

**day\_month\_sep**

**set.seed(10)**

**sample(nrow(data.set),**

**5, replace = FALSE)**

**data.set[sample(nrow(data.set),**

**5, replace = FALSE), 'Temp'] = NA**

**data.set**

**sum(is.na(data.set))**

**na\_index = which(is.na(data.set))**

**data.set$Temp[na\_idex] = 'Unknow'**

**data.set = fill(date.set, Temp)**

**install.packages('PogromcyDanych')**

**library(PogromcyDanych)**

**df = auta2012**

**sapply(df, is.factor)**

**summarise(df, rows = n(),**

**'nr of models', n\_distinct(Marka))**

**x= group\_by(df, Marka)**

**y= summarise(x, df = n())**

**y**

**count(df, Marka)**

**numb = summarise(count(df, Mark),max(n))[1,]**

**y[y$df == numb,]**

**df$Rodzaj.paliwa == 'benzyna'**

**nrow(df[df$Rodzaj.paliwa == 'benzyna'])**

**nrow(filter(df, Rodzaj.paliwa == 'benzyna'))/nrow(df)**

**nrow(filter(df,Cena.w.PLN>2000 ))**

**nrow(filter(df,Pojemnosc.skokowa>=1500 ))**

**nrow(filter(df,Pojemnosc.skokowa>=1500 )) / nrow(df)**

**nrow(filter(df,Cena.w.PLN>2000 &**

**Rodzaj.paliwa == 'olej napedowy (diesel)' ))**

**df %>%**

**filter(Marka == 'Volkswagen') %>%**

**group\_by(Rodzaj.paliwa) %>%**

**count()**

**install.packages('PogromcyDanych')**

**library(PogromcyDanych)**

**library(tibble)**

**library(tidyr)**

**library(dplyr)**

**df = auta2012**

**view(df)**

**df %>%**

**filter(Marka == 'Volkswagen') %>%**

**group\_by(Rodzaj.paliwa) %>%**

**count()**

**mean(df$Cena.w.PLN[df$Marka == 'Volkswagen'])**

**mean(df$Przebieg.w.km[df$Marka == 'Volkswagen'], na.rm = TRUE)**

**summarise(filter(df, Marka == 'Volkswagen'),**

**`evrydz cena` = mean(Cena.w.PLN),**

**`evrydz przebieg` = mean(Przebieg.w.km, na.rm = TRUE))**

**df %>%**

**filter(Marka == 'Volkswagen') %>%**

**group\_by(Model) %>%**

**count()**

**summarise(group\_by(df, Model),**

**`average price` = mean(Cena.w.PLN , na.rm = TRUE)) -> pom1**

**view(pom1)**

**df %>%**

**filter(Marka == 'Toyota', Model == 'Corolla') %>%**

**select(Cena) %>%**

**quantile(na.rm = TRUE)**

**df %>%**

**filter(Marka == 'Toyota') %>%**

**group\_by(Model) %>%**

**summarise(`srednia cena` = mean(Cena.w.PLN)) %>%**

**arrange(desc(srednia.cena))**

**df %>%**

**filter(Marka == 'Volkswagen', Model = 'Passat',Rok.produkcji = 2006, Cena.w.PLN<35000) %>%**

**count()**

**df %>%**

**filter(Rok.produkcji = 2007) %>%**

**group\_by(Marka) %>%**

**summarise(`ilosc` = n()) %>%**

**arrange(ilosc)->pom2**

**rlang::last\_error()**

**install.packages('AER')**

**library(AER)**

**glimpse(Fertility)**

**?Fertility**

**df\_fert <- glimpse(Fertility)**

**glimpse(df\_fert)**

**view(df\_fert)**

**slice(select(df\_fert, age, work), 35:50)**

**slice(select(df\_fert, nrow(df\_fert)), nrow(df\_fert))**