# # Programowanie w R – kurs podstawowy

## **# Zajęcia 7. Operacje logiczne**

# **---------------------Czyszczenie danych**

# Pobranie danych

future500=read.csv("http://www.kuligowska.com/uouw/future500.csv",sep=";",dec=",", header=T)

# Usuńmy słowo “Dollars” z kolumny Expenses

# Usuńmy znaczek “%” z kolumny Growth

# Usuńmy znaczek “$” z kolumny Revenue

# **---------------------Równość**

# Równość sprawdzamy „==”

# Nierówność sprawdzamy „!=”

# **---------------------Mniejszy i większy**

# > < >= <=

# **---------------------Porównanie wektorów**

# można porównać dwa wektory tej samej długości, każdy element z każdym

# można porównać wektor do liczby

# **---------------------Porównanie macierzy**

# **---------------------Operatory & oraz | czyli i oraz lub**

# operator koniunkcji &

# operator alternatywy |

# na wektorach

# na macierzach

**# Zadania**

# Z podanych 4 wektorów utwórz ramkę danych o nazwie „Studenci”.

nazwisko <- c("Burdzy", "Dyrkacz", "Grzeszczak", "Kowalczyk", "Lubicki", "Nowak", "Nowicki", "Sachowicz", "Walczak", "Wrotek")

typ <- c("dzienne", "dzienne", "zaoczne", "dzienne", "zaoczne", "zaoczne", "dzienne", "zaoczne", "zaoczne", "dzienne")

test1 <- c(42, 85, 49, 76, 73, 36, 66, 91, 55, 89)

test2 <- c(52, 65, 68, 25, 83, 18, 66, 94, 40, 75)

# Utwórz nową kolumnę o nazwie „wynik”,

# która będzie zawierać średni wynik osiągnięty przez studenta z obu testów

# Utwórz wektor logiczny „zaliczenie” przyjmujący wartość PRAWDA,

# jeśli średni wynik studenta z obu testów jest większy niż 50 punktów

# i z żadnego z testów nie otrzymał wyniku gorszego niż 30 punktów

# Dodaj wektor „zaliczenie” do ramki „Studenci” jako nową kolumnę „zaliczenie”

# Zrób to w jednej linijce – połącz dwa poprzednie polecenia

# (bez tworzenia wektora „zaliczenie”)

**# -------------------Filtrowanie wierszy ramki danych – wektor logiczny**

# Z naszej ramki danych zawierającej wyniki testów chcielibyśmy wybrać

# tylko tych studentów, którzy zaliczyli zajęcia

# Możemy wybrać wszystkie dane/kolumny dla studentów z pozytywnym wynikiem egzaminu

# Możemy też wybrać studentów z **negatywnym** wynikiem egzaminu

**# -------------------Filtrowanie wierszy ramki danych**

# Wyświetl dane tylko dla osoby o nazwisku Nowak

# Wyświetl dane tylko dla osób ze studiów dziennych

# Wyświetl dane dla osób, które średnio z obu testów zdobyły więcej niż 80 pkt

# Wyświetl dane dla osób, które zdobyły więcej niż 50 pkt, ale nie więcej niż 60

# Wyświetl dane dla osób, które chociaż z jednego testu zdobyły co najmniej 50 pkt

# Ile jest łącznie takich osób? Użyj funkcji sum() do zliczenia liczby

# obserwacji spełniających powyższy warunek

# Wyświetl dane dla osób o nazwisku Burdzy i Lubicki

# Wyświetl dane dla pozostałych osób poza Burdzy i Lubicki

**# -------------------Filtrowanie wierszy i kolumn ramki danych**

# Użyjmy wektora “zaliczenie” do wyświetlenia nazwisk studentów

# z pozytywnym wynikiem egzaminu

# Użyjmy wektora “zaliczenie” do wyświetlenia nazwisk

# i wyniku studentów z pozytywnym wynikiem egzaminu

**# Zadanie**

# Wyświetl średni wynik z testów tylko dla osoby o nazwisku Wrotek

# Wyświetl wyniki z obu testów tylko dla osoby o nazwisku Wrotek

# Wyświetl nazwiska i wyniki z obu testów dla wszystkich osób poza Nowakiem

**# -------------------Filtrowanie wierszy ramki danych – tekst a faktory**

# Wyświetl średni wynik z testów tylko dla osób ze studiów dziennych – to jest OK

# Wyświetl dane dla osób o nazwiskach od litery G - nie da się, bo to faktor

# Musimy najpierw zmienić typ danych na tekstowy

**# ------------------- Funkcja subset – wybór wierszy**

# Pierwszy argument funkcji to zbiór danych, który chcemy filtrować.

# Drugi argument to warunek, który ma być spełniony.

# Użyjmy funkcji **subset()**do wybrania studentów

# z pozytywnym wynikiem egzaminu

# Użyjmy funkcji **subset()**do wybrania studentów

# z negatywnym wynikiem egzaminu

# Wyświetl dane dla studentów z wynikiem wyższym niż 70 pkt

# Wyświetl dane dla studentów studiów dziennych

**# ------------------- Funkcja subset – wybór kolumn**

# Wyświetlmy tylko kolumnę z wynikami z pierwszego testu

# Wyświetlmy kolumny z wynikami z obu testów

# Wyświetlmy kolumny z wynikami z obu testów i wynikiem średnim dla studentów, którzy zdali

**# -------------------Funkcja split (dla faktorów)**

# Wyświetlmy osobno dane dla osób ze studiów dziennych i zaocznych

# Wyświetlmy osobno dane dla osób ze studiów zaocznych

**# -------------------Sortowanie kolumn**

# Dodajemy jedną dodatkową kolumnę

# Przesuwamy ósmą kolumnę LP na pierwsze miejsce

# Chcemy zamienić pierwszą kolumnę z drugą

# Przesuwamy ósmą kolumnę na drugie miejsce

#-----------------------

# Zadanie 1. Zaimportuj dwie ramki danych i połącz je w jedną ramkę Kraje

kraje1=read.csv("http://www.kuligowska.com/uouw/kraje1.csv", sep=";", dec=",", header=T)

colnames(kraje1)=c("Panstwo","Kod","PKB","Populacja","Dostep\_do\_Internetu","Wzrost\_populacji")

kraje3=read.csv("http://www.kuligowska.com/uouw/kraje3.csv", sep=";", dec=",", header=T)

# Zadanie 2. Usuń zbędną kolumnę, która się powtarza

# Zadanie 3. Dostosuj nazwy kolumn, aby wszystkie były po polsku

# Zadanie 4. Dodaj kolumnę LP z liczbami porządkowymi

# Zadanie 5. Przenieś kolumnę LP na początek tabeli (nadpisz zmiany)

# Zadanie 6. Przenieś kolumnę Państwo na drugie miejsce w tabeli (nadpisz zmiany)

# Zadanie 7. Posegreguj tabelę wg Państw rosnąco

# Zadanie 8. Utwórz nową kolumnę PKB\_os z PKB na mieszkańca

# Zadanie 9. Usuń kolumnę PKB

# Zadanie 10. Posegreguj tabelę wg PKB\_os malejąco

# Zadanie 11. Zmień typ zmiennej Państwo na tekstową

# Zadanie 12. Wyświetl kraje na literę R

# Zadanie 13. Wyświetl dane dla Polski

# Zadanie 14. Wyświetl dane dla krajów, w których PKB na mieszkańca jest nie większy niż 1000

# Zadanie 15. Ile jest takich krajów?

# Zadanie 16. Wyświetl wszystkie kategorie zmiennej Region

# Zadanie 17. Wyświetl dane dla krajów, w których PKB na mieszkańca jest niższy od 1000 i nie leżą w Afryce Subsaharyjskiej

# Zadanie 18. Używając funkcji split wyświetl dane dla Afryki Subsaharyjskiej

# Zadanie 19. Używając funkcji subset wyświetl dane dla krajów, w których mniej niż 20% mieszkańców ma dostęp do internetu

# Zadanie 20. Używając funkcji subset wyświetl nazwy państw, regionu i dostępu do internetu dla krajów, w których więcej niż 80% populacji mieszka w miastach

# Zadanie 21. Wyświetl to samo co w poprzednim ćwiczeniu, ale dla krajów, w których mniej niż 30% populacji mieszka w miastach