



**BSS**  
BIAŁOWIESKA SZKOŁA STATYSTYKI

# **Wprowadzenie do R**

# Czym jest R?



#Rozbudowane narzędzie statystyczne i graficzne

#Specyficzny język programowania:

- Język skupiony raczej wokół obiektów niż akcji
- Obiekty złożone z danych oraz funkcji

#Prosty sposób importu/eksportu danych

#Darmowy i otwarty

#Dostępny dla wielu systemów operacyjnych  
(Windows, Mac, Linux, FreeBSD itd.)

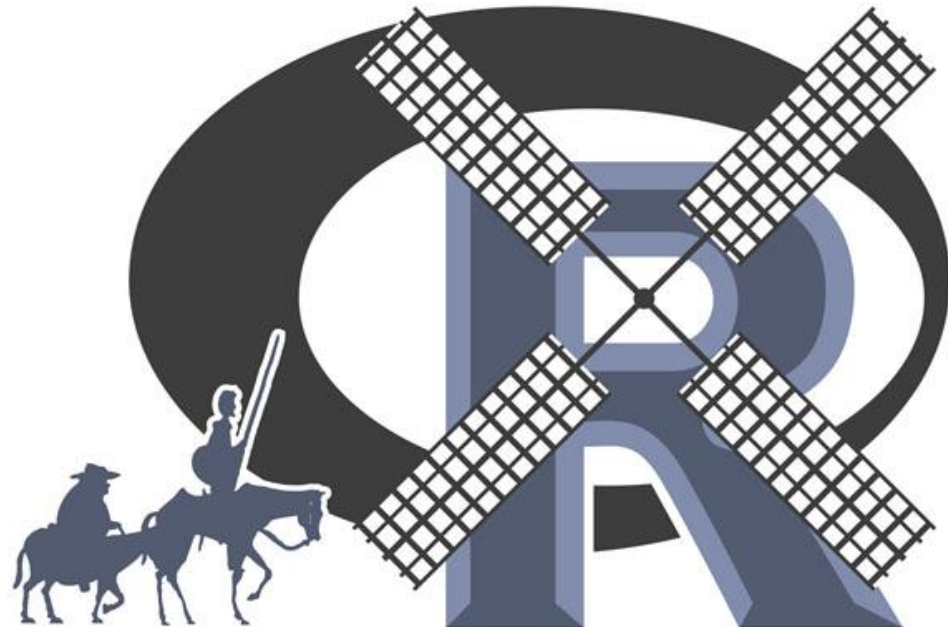
# Zalety

- #Praca z wieloma zestawami danych o różnej strukturze jednocześnie: listy, ramki danych, wektory, macierze itp.
- #Pojawianie się nowych funkcji i bibliotek, przez co R ciągle poszerzany
- #Budowa pętli, własnych funkcji w zależności od potrzeb
- #Doświadczalna interaktywność – programujesz tak długo, aż otrzymasz to, co potrzebujesz
- #Dostępne prawie każde narzędzie statystyczne
- #Do rozwiązania tego samego problemu można dojść wieloma drogami
- #Szeroko rozbudowana pomoc



# Wady

- #Bardzo stroma krzywa uczenia się
- #Nie jest programem typu „wskaż i kliknij”
- #Łatwo o pomyłki, które są trudne do zlokalizowania
- #Program metody „prób i błędów”



# Jak wygląda?

Lista obiektów istniejących w obrębie projektu

Edytor tekstu (skrypt)

The screenshot displays the RStudio interface with three main panes:

- Script Editor (Top Left):** Contains R code for data manipulation and plotting. A red arrow points to the `bez.plot` object definition.
- Environment Pane (Top Right):** Lists objects in the global environment. A red arrow points to the `bez.plot` object.
- Plot Window (Bottom Right):** Shows a scatter plot of PC1 vs PC2 with vectors and labels. A red arrow points to the plot area.

**Console (Bottom Left):** Shows the execution of the script, including the creation of `bez.plot`.

konsola

Okno grafiki

# Podstawowe funkcje

Funkcja	Działanie
<code>read.table</code>	Wczytuje dane
<code>colnames/rownames</code>	Nazwy kolumn/wierszy
<code>colSums/rowSums</code>	Sumy wartości w kolumnach/wierszach
<code>colMeans/rowMeans</code>	Średnie wartości z kolumn/wierszy
<code>c</code>	Łączy wartości w wektor
<code>cbind/rbind</code>	Skleja obiekty kolumnami/wierszami
<code>plot</code>	Graficzne przedstawienie danych
<code>abline</code>	Dodaje linię trendu
<code>mean/median</code>	Średnia/mediana
<code>sqrt</code>	Pierwiastek kwadratowy
<code>sd</code>	Odchylenie standardowe

## Funkcja

## Działanie

`t.test`

Test t Studenta

`wilcox.test`

Test U Manna-Whitneya

`chisq.test`

Test Chi-kwadrat

`cor.test`

Test korelacji

`lm`

Tworzy model liniowy z danych

`lme`

Tworzy mieszany model liniowy

`glm`

Tworzy zgeneralizowany model liniowy

`anova`

Analiza wariancji lub porównywanie modeli

`summary`

Podsumowanie modelowanych wyników

`str`

Struktura danych (z obiektów jakiego typu składa się ramka danych)

`head`

Wyświetla pierwsze 6 kolumn i wierszy z tabeli; funkcja przydatna do sprawdzenia, czy dane były prawidłowo wczytane

`t`

Transpozycja kolumn z wierszami



# Pierwsze kroki

#R jest typem języka obiektowego

```
x<-33
```

#x jest obiektem, strzałka to symbol przypisania liczby 33 do obiektu x

#Obiekt może zawierać więcej liczb lub znaków

#Pojedyncze wartości (liczbowe i nie tylko) można łączyć w bardziej złożone całości zwane obiektami

# Typy obiektów

## #Wektory (vectors)

- Najprostszy typ obiektu w R
- Funkcją tworzącą wektor jest `c`
- Ciągi liczb (integer), znaków (character) lub wartości logicznych (logical)

```
shrews.mass<-as.integer(c(26,29,41,24,28,56,74,35,68,95,  
45,67,89,35,67,88,75,34))
```

```
fur.color<-  
as.character(c('gray','gray','gray','gray','brown','brown',  
, 'brown','brown','brown','black','black','black',  
'black','black','black','gray','brown','brown'))
```

```
male<-  
as.logical(c('TRUE','TRUE','TRUE','TRUE','FALSE','FALSE',  
'FALSE','FALSE','FALSE','TRUE','TRUE','TRUE','TRUE',  
'FALSE','FALSE','FALSE','FALSE','FALSE'))
```

```
str(shrews.mass) #patrzemy, jaka jest struktura danych
> str(shrews.mass) int [1:18] 26 29 41 24 28 56 74 35 68
95 ...
```

```
str(fur.color)
> str(fur.color) chr [1:18] "gray" "gray" "gray" "gray"
"brown" "brown" "brown" "brown" "brown" "black" ...
```

```
str(male)
> str(male) logi [1:18] TRUE TRUE TRUE TRUE FALSE FALSE
...
```

```
> class(shrews.mass)
[1] "integer"
> class(fur.color)
[1] "character"
> class(male)
[1] "logical"
```

## #Listy (lists):

- podobnie jak wektory, listy to ciągi obiektów. Różnice pomiędzy wektorami a listami:

- W listach każdy element może być innego typu
- Możliwość odwoływania się do elementów listy za pomocą nazwy tego elementu i operatora `$`

```
ryjowki.list<-list(as.integer(shrews.mass), fur.color,  
male)
```

```
names(ryjowki.list)<-c("masa", "sierśc", "chłop")
```

```
> ryjowki.list  
$masa  
[1] 26 29 41 24 28 56 74 35 68 95 45 67 89 35 67 88 75 34  
  
$sierśc  
[1] "gray" "gray" "gray" "gray" "brown" "brown" "brown" "brown" "brown"  
[10] "black" "black" "black" "black" "black" "black" "gray" "brown" "brown"  
  
$chłop  
[1] TRUE TRUE TRUE TRUE FALSE FALSE FALSE FALSE FALSE TRUE TRUE TRUE  
[13] TRUE FALSE FALSE FALSE FALSE FALSE
```

#W poprzednim przykładzie nadaliśmy elementom listy nazwy. A co w przypadku, gdy tych nazw nie nadamy?

```
> ryjowki.list
[[1]]
 [1] 26 29 41 24 28 56 74 35 68 95 45 67 89 35 67 88 75 34

[[2]]
 [1] "gray" "gray" "gray" "gray" "brown" "brown" "brown" "brown" "brown"
[10] "black" "black" "black" "black" "black" "black" "black" "gray" "brown" "brown"

[[3]]
 [1] TRUE TRUE TRUE TRUE FALSE FALSE FALSE FALSE FALSE TRUE TRUE TRUE
[13] TRUE FALSE FALSE FALSE FALSE FALSE
```

#Wtedy, aby odwołać się do konkretnego elementu listy używamy atrybutu `[[n]]`, np. dla poziomu pierwszego:

```
> ryjowki.list[[1]]
 [1] 26 29 41 24 28 56 74 35 68 95 45 67 89 35 67 88 75 34
```

#Listy będziemy wykorzystywać przy analizach wielowymiarowych do ekstrakcji współrzędnych punktów w przestrzeni ordynacyjnej

## #as.factor:

- funkcja używana, gdy chcemy wektor zakodować jako czynnik (jako typ wyliczeniowy)
- przydatna, gdy jesteśmy pewni, że nasz ciąg liczb przyjmuje wartości dyskretne, np. 2, 4, 6, 8, 10, 12

```
> as.factor(shrews.mass)
[1] 26 29 41 24 28 56 74 35 68 95 45 67 89 35 67 88 75 34
Levels: 24 26 28 29 34 35 41 45 56 67 68 74 75 88 89 95
```

**#Ramki danych (data.frame) :**

- Podstawowy typ obiektów w modelowaniu i przedstawianiu graficznemu danych.
- Kojarzona z tabelaryczną/macierzową strukturą , której elementy w każdej kolumnie są tego samego typu, ale mogą różnić się pomiędzy kolumnami.
- Dlatego można ją traktować jako listę wektorów o tej samej długości, a każdy z nich odpowiada jednej kolumnie (w listach niekoniecznie)

response	predictor	colour	response
1.23	A	black	1.23
2.45	A	black	2.45
1.11	A	black	NA
2.35	A	black	2.35
5.78	B	blue	NA
4.32	B	blue	4.32
3.31	B	blue	3.31
6.98	B	blue	6.98

```
ryjowki<-as.data.frame(ryjowki.list) #sklejenie wektorów  
kolumnami i zapisanie ich jako ramkę danych
```

```
> ryjowki  
  shrews.mass fur.color male  
1         26      gray  TRUE  
2         29      gray  TRUE  
3         41      gray  TRUE  
4         24      gray  TRUE  
5         28     brown FALSE  
6         56     brown FALSE  
7         74     brown FALSE  
8         35     brown FALSE  
9         68     brown FALSE  
10        95     black  TRUE  
11        45     black  TRUE  
12        67     black  TRUE  
13        89     black  TRUE  
14        35     black FALSE  
15        67     black FALSE  
16        88      gray FALSE  
17        75     brown FALSE  
18        34     brown FALSE
```



# Podstawowe działania na ramkach danych

Funkcja	Działanie
<code>[ , 1]</code>	Pierwsza kolumna z tabeli
<code>[ , c(1:3)]</code>	Kolumny od 1 do 3
<code>[ , c(1, 3)]</code>	Kolumna 1 i 3
<code>[ , -1]</code>	Usuwanie pierwszej kolumny
<code>[ , -c(1, 3)]</code>	Usuwanie kolumny pierwszej i trzeciej
<code>[1, ]</code>	Pierwszy wiersz
<code>[c(1:3) , ]</code>	Wiersze od 1 do 3
<code>[c(1, 3) , ]</code>	Wiersz 1 i 3
<code>[-1, ]</code>	Usuwanie pierwszego wiersza
<code>[-c(1, 3) , ]</code>	Usuwanie pierwszego i trzeciego wiersza

# Tabela wąska vs. tabela szeroka

```
> epi.all
      species site freq
1    Alyxoria varia s.l. A1h 1
2    Arthonia arthonioides A1h 1
3    Arthonia spadicea A1h 1
4    Arthonia vinosa A1h 1
5    Calicium adpersum A1h 1
6    chaenotheca chrysocephala A1h 1
7    chaenotheca furfuracea A1h 1
8    chaenotheca trichialis A1h 1
9    chrysothrix candelaris A1h 1
10   Cladonia fimbriata A1h 1
11   Evernia prunastri A1h 1
12   Lecanora argentata s.l. A1h 1
13   Lecidella elaeochroma A1h 1
14   Loxospora elatina A1h 1
15   Melanelixia glabrata A1h 1
16   Opegrapha niveoatra A1h 1
17   Opegrapha vermicellifera A1h 1
18   Parmelia sulcata s.l. A1h 1
19   Pertusaria amara A1h 1
20   Pertusaria coccodes A1h 1
21   Porina aenea A1h 1
22   Usnea dasypoga A1h 1
23   Cladonia coniocraea A1h 2
24   Graphis scripta s.l. A1h 2
25   Hypogymnia physodes A1h 2
26   Lepraria incana s.l. A1h 3
27   Pertusaria leioplaca A1h 2
28   Phlyctis argena A1h 2
29   Pyrenula nitida A1h 2
30   Pyrenula nitidella A1h 2
31   Thelotrema lepadinum A1h 2
32   Zwackhia viridis A1h 2
33   Alyxoria varia s.l. A10h 1
34   Arthonia arthonioides A10h 1
35   Arthonia spadicea A10h 1
36   chaenotheca furfuracea A10h 1
37   Evernia prunastri A10h 1
38   Felipes leucopellaeus A10h 1
39   Fellhanera gyrophorica A10h 1
40   Lecanora argentata s.l. A10h 1
41   Lecanora thysanophora A10h 1
42   Loxospora elatina A10h 1
```

```
> epi
      A10h A11h A1h A2h A3h A4h A5h A6h A7h A8h A9h B10h B11h B1h B2h B3h
Acr.gem  0    2    0    1    0    1    0    0    0    0    1    0    0    1    0    0
Aly.var  1    0    1    2    2    2    2    1    1    1    2    3    1    1    1    2
Ani.bif  0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0

      B4h B5h B6h B7h B8h B9h C10h C11h C1h C2h C3h C4h C5h C6h C7h C8h C9h
Acr.gem  0    1    0    0    1    1    2    0    0    0    0    0    0    1    0    1    1
Aly.var  2    0    0    1    3    2    2    2    0    0    2    1    0    2    1    1    2
Ani.bif  0    0    0    0    0    0    1    0    0    0    0    0    0    0    0    0    0

      D10h D11h D1h D2h D3h D4h D5h D6h D7h D8h D9h E10h E11h E1h E2h E3h
Acr.gem  0    0    0    0    1    0    0    0    0    0    1    0    0    0    0    0
Aly.var  1    1    0    0    2    1    2    1    0    2    2    0    0    0    0    1
Ani.bif  0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0

      E4h E5h E6h E7h E8h E9h F10h F11h F1h F2h F3h F4h F5h F6h F7h F8h F9h
Acr.gem  1    0    0    0    0    0    1    1    0    0    0    1    0    0    0    1    0
Aly.var  1    2    1    1    2    0    0    2    0    0    0    2    1    0    1    1    1
Ani.bif  0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0

      G10h G11h G1h G2h G3h G4h G5h G6h G7h G8h G9h H10h H11h H1h H2h H3h
Acr.gem  0    0    0    0    0    0    1    0    0    0    0    0    0    0    0    0
Aly.var  3    3    0    0    0    0    0    0    1    3    2    0    0    0    0    0
Ani.bif  0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0

      H4h H5h H6h H7h H8h H9h I10h I11h I1h I2h I3h I4h I5h I6h I7h I8h I9h
Acr.gem  0    0    0    1    0    0    0    0    0    0    0    0    0    0    0    0    0
Aly.var  0    0    0    3    3    0    0    0    0    0    0    1    1    3    2    2    1
Ani.bif  0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
```

#Jeżeli tabela ma wiele kolumn i nie mieści się w oknie konsoli, wówczas R przerzuca niemieszczące się kolumny do następnego wiersza i kontynuuje ich wyświetlanie

# Łączenie wąskich tabel w szerokie

epiphytes.hist

```
> epiphytes.hist
```

	species	site	freq
1	Alyxoria varia s.l.	A1h	1
2	Arthonia arthonioides	A1h	1
3	Arthonia spadicea	A1h	1
4	Arthonia vinosa	A1h	1
5	Calicium adpersum	A1h	1
6	Chaenotheca chrysocephala	A1h	1
7	chaenotheca furfuracea	A1h	1
8	chaenotheca trichialis	A1h	1
9	Chrysothrix candelaris	A1h	1
10	Cladonia fimbriata	A1h	1
11	Evernia prunastri	A1h	1
12	Lecanora argentata s.l.	A1h	1
13	Lecidella elaeochroma	A1h	1
14	Loxospora elatina	A1h	1
15	Melanelixia glabrata	A1h	1
16	Opegrapha niveoatra	A1h	1
17	Opegrapha vermicellifera	A1h	1
18	Parmelia sulcata s.l.	A1h	1
19	Pertusaria amara	A1h	1
20	Pertusaria coccodes	A1h	1
21	Porina aenea	A1h	1
22	Usnea dasypoga	A1h	1
23	Cladonia coniocraea	A1h	2
24	Graphis scripta s.l.	A1h	2
25	Hypogymnia physodes	A1h	2
26	Lepraria incana s.l.	A1h	3
27	Pertusaria leioplaca	A1h	2
28	Phlyctis argena	A1h	2
29	Pyrenula nitida	A1h	2
30	Pyrenula nitidella	A1h	2
31	Thelotrema lepadinum	A1h	2
32	Zwackhia viridis	A1h	2
33	Alyxoria varia s.l.	A10h	1
34	Arthonia arthonioides	A10h	1
35	Arthonia spadicea	A10h	1

epiphytes.new

```
> epiphytes.new
```

	species	site	freq
1	Acrocordia gemmata	A1	1
2	Alyxoria varia s.l.	A1	1
3	Arthonia arthonioides	A1	1
4	Arthonia radiata	A1	1
5	Arthothelium ruanum	A1	1
6	Chaenotheca brachypoda	A1	1
7	Chaenotheca chrysocephala	A1	1
8	chaenotheca ferruginea	A1	1
9	Chaenotheca trichialis	A1	1
10	Cladonia digitata	A1	1
11	Inoderma byssaceum	A1	1
12	Lecanora carpinea s.l.	A1	1
13	Lecanora chlarotera	A1	1
14	Melanelixia glabrata	A1	1
15	Micarea prasina s.l.	A1	1
16	Microcalicium disseminatum	A1	1
17	Opegrapha vermicellifera	A1	1
18	Pertusaria coccodes	A1	1
19	Pertusaria ophthalmiza	A1	1
20	Pertusaria pertusa	A1	1
21	Pyrenula nitidella	A1	1
22	Ramalina farinacea	A1	1
23	Reichlingia leopoldii	A1	1
24	Arthonia spadicea	A1	2
25	Buellia griseovirens	A1	2
26	Chrysothrix candelaris	A1	2
27	Cladonia coniocraea	A1	2
28	Coenogonium pineti	A1	2
29	Fellhanera gyrophorica	A1	3
30	Graphis scripta s.l.	A1	3
31	Lecanora argentata s.l.	A1	3
32	Lecanora thysanophora	A1	3
33	Lecidella elaeochroma	A1	2
34	Lepraria finkii s.l.	A1	3
35	Lepraria incana s.l.	A1	3

```
#import danych
epiphytes.hist<-read.csv('epiphytes.hist.csv', sep=';')
epiphytes.new<-read.csv('epiphytes.new.csv', sep=';')

#sprawdzenie, czy dobrze wczytaliśmy dane
head(epiphytes.hist)
head(epiphytes.new)

#sklejenie tabel wierszami
epi.all<-rbind(epiphytes.hist,epiphytes.new)
```

```
#załadowujemy bibliotekę
library(reshape2)

#przekształcamy tabelę wąską do tabeli szerokiej
epi.cast<-as.data.frame(dcast(epi.all, species~site,
value.var='freq', drop=FALSE))

#eksportujemy tabelę z R do pliku .csv
write.table(epi.cast, 'epi.casting.csv', sep=';')

#nową tabelę można znowu wgrać do R
epi<-read.csv('epi.cast.csv', sep=';')
```

## #tabela szeroka

[illegible]

#transpozycja kolumn z wierszami  
epi .t<-t(epi)

> epi.t														
	Acr.gem	Aly.var	Ani.bif	Art.art	Art.atr	Art.did	Art.rad	Art.spa	Art.vin	Ath.rua	Ath.spe	Bac.arc	Bac.bec	
A10h	0	1	0	1	0	1	0	0	0	0	0	0	0	
A11h	2	0	0	1	0	0	0	1	0	0	0	0	1	
A1h	0	1	0	1	0	1	1	0	0	0	0	0	0	
A2h	1	2	0	2	0	1	0	1	1	0	0	0	0	
A3h	0	2	0	0	1	0	0	0	0	0	0	0	1	
A4h	1	2	0	2	0	1	0	1	0	0	0	0	0	
	Bac.lau	Bac.rub	Bct.dry	Bia.glo	Bry.fus	Bry.imp	Bue.dis	Bue.eru	Bue.gri	Bue.sch	Cal.ads	Cal.gla	Cal.sal	
A10h	0	0	0	0	0	0	0	0	0	0	0	0	0	
A11h	0	0	0	0	0	1	0	0	0	0	0	0	1	
A1h	0	0	0	0	0	0	1	0	0	0	0	0	0	
A2h	0	0	0	0	0	0	0	0	0	0	0	0	0	
A3h	0	0	0	0	0	0	0	0	0	1	0	0	0	
A4h	1	0	0	0	0	1	0	0	1	1	0	0	0	
	Cal.vir	Clp.pyr	Can.xan	Crb.ant	Cet.cet	Cha.bra	Cha.bru	Cha.chl	Cha.chr	Cha.fer	Cha.fur	Cha.gra	Cha.pha	
A10h	0	0	2	0	1	0	0	0	2	0	0	0	0	
A11h	0	0	1	0	0	0	0	0	3	0	0	2	0	
A1h	0	0	1	0	1	0	0	1	1	0	0	2	0	
A2h	0	0	1	0	1	0	0	0	2	0	1	2	0	
A3h	0	0	2	0	1	0	0	1	2	0	0	2	0	
A4h	0	0	1	0	1	0	0	1	2	0	0	2	1	
	Cha.tri	Cha.xyl	Chr.can	Cl.a.cen	Cl.a.chl	Cl.a.con	Cl.a.dig	Cl.a.fim	Cl.a.flo	Cl.a.gla	Cl.a.mac	Cl.a.par	Cl.a.squ	
A10h	0	0	0	0	0	0	1	1	1	0	2	0	3	
A11h	0	0	0	0	0	0	0	0	1	0	3	0	2	
A1h	1	0	0	0	0	0	1	0	0	0	2	0	2	
A2h	0	0	0	0	0	0	0	1	1	0	3	0	3	
A3h	0	0	0	0	0	0	1	0	0	0	3	0	3	
A4h	0	0	0	0	0	0	0	0	1	0	3	0	3	

#przekształcenie tabeli szerokiej do wąskiej  
epi.melt<-melt(epi.t.id, value.name='freq')

```
> epi.melt
  Var1 Var2 freq
1  A10h Acr.gem 0
2  A11h Acr.gem 2
3   A1h Acr.gem 0
4  A2h Acr.gem 1
5  A3h Acr.gem 0
6  A4h Acr.gem 1
7  A5h Acr.gem 0
8  A6h Acr.gem 0
9  A7h Acr.gem 0
10 A8h Acr.gem 0
11 A9h Acr.gem 1
12 B10h Acr.gem 0
13 B11h Acr.gem 0
14  B1h Acr.gem 1
15  B2h Acr.gem 0
16  B3h Acr.gem 0
17  B4h Acr.gem 0
18  B5h Acr.gem 1
19  B6h Acr.gem 0
20  B7h Acr.gem 0
21  B8h Acr.gem 1
22  B9h Acr.gem 1
23 C10h Acr.gem 2
24 C11h Acr.gem 0
25  C1h Acr.gem 0
26  C2h Acr.gem 0
27  C3h Acr.gem 0
28  C4h Acr.gem 0
```



```
#zmiana nazw kolumn
```

```
colnames(epi.melt)<-c("site", "species", "frequency")
```

```
> epi.melt
  site species frequency
1  A10h  Acr.gem         0
2  A11h  Acr.gem         2
3   A1h  Acr.gem         0
4   A2h  Acr.gem         1
5   A3h  Acr.gem         0
6   A4h  Acr.gem         1
7   A5h  Acr.gem         0
```

```
#tworzenie obiektu z danymi przynależności gatunku do siedliska
forest<-rep(c("dec.forest", "con.forest", "mix.forest"),49)
species<-colnames(epi.t.id)
habitat<-as.data.frame(cbind(species, forest))
```

```
> habitat
  species forest
1  Acr.gem dec.forest
2  Aly.var con.forest
3  Ani.bif mix.forest
4  Art.art dec.forest
5  Art.atr con.forest
6  Art.did mix.forest
7  Art.rad dec.forest
8  Art.spa con.forest
9  Art.vin mix.forest
10 Ath.rua dec.forest
11 Ath.spe con.forest
12 Bac.arc mix.forest
13 Bac.bec dec.forest
14 Bac.lau con.forest
15 Bac.rub mix.forest
16 Bct.dry dec.forest
17 Bia.glo con.forest
18 Bry.fus mix.forest
19 Bry.imp dec.forest
20 Bue.dis con.forest
```

```
#dołączanie do tabeli wąskiej kolumny z tabeli szerokiej  
epi.join<-left_join(epi.melt, habitat, by='species')
```

```
> epi.join  
  site species frequency    forest  
1  A10h Acr.gem         0 dec.forest  
2  A11h Acr.gem         2 dec.forest  
3   A1h Acr.gem         0 dec.forest  
4  A2h Acr.gem         1 dec.forest  
5  A3h Acr.gem         0 dec.forest  
6  A4h Acr.gem         1 dec.forest  
7  A5h Acr.gem         0 dec.forest  
8  A6h Acr.gem         0 dec.forest  
9  A7h Acr.gem         0 dec.forest  
10 A8h Acr.gem         0 dec.forest  
11 A9h Acr.gem         1 dec.forest  
12 B10h Acr.gem         0 dec.forest  
13 B11h Acr.gem         0 dec.forest  
14 B1h Acr.gem         1 dec.forest  
15 B2h Acr.gem         0 dec.forest  
16 B3h Acr.gem         0 dec.forest  
17 B4h Acr.gem         0 dec.forest  
18 B5h Acr.gem         1 dec.forest  
19 B6h Acr.gem         0 dec.forest  
20 B7h Acr.gem         0 dec.forest  
21 B8h Acr.gem         1 dec.forest  
22 B9h Acr.gem         1 dec.forest  
23 C10h Acr.gem         2 dec.forest  
24 C11h Acr.gem         0 dec.forest  
25 C1h  Acr.gem         0 dec.forest
```

#funkcja `left_join` bardzo przydatna, gdy przy analizie kolejnych zbiorów danych pracujemy z cechami tych samych gatunków.

#Wtedy nie trzeba ręcznie do każdego gatunku doklejać jego cechy z poprzedniej bazy danych, tylko z bazy poprzedniej można wartości tych cech wkleić automatycznie w nową bazę.

#OSZCZĘDNOŚĆ CZASU :

- Ręczne przekształcanie tabeli wąskiej w szeroką (np. w przypadku, gdy mielibyśmy do czynienia z setkami powierzchni badawczych i setkami obserwacji) zajęłoby tygodnie, jak nie miesiące
- To samo tyczy się ręcznego doklejania cech tych samych gatunków do wąskich/szerokich tabel

#R robi to w ułamek sekundy...

# Działania na tabeli przy zastosowaniu funkcji z pakietu dplyr

change.per.summit.merged

```
> change.per.summit.merged
```

	Access	Tourism	BCdist	LoId	Lnew	ToId	Tnew	MoId	Mnew	NoId	Nnew	RoId	Rnew
1	2	1	0.3658537	4.632653	4.742424	1.714286	1.575758	3.295918	3.196970	2.553191	2.363636	3.683673	3.227273
2	3	1	0.5303030	4.589286	4.657895	1.937500	1.907895	3.098214	3.065789	2.764151	2.687500	4.172727	4.085526
3	4	2	0.6821192	4.218750	3.862069	2.148438	2.465517	3.320312	3.304598	2.629032	2.988235	3.571429	3.252874
4	2	2	0.5405405	4.632653	4.920000	1.775510	1.500000	3.285714	3.160000	2.622222	2.229167	3.530612	2.920000
5	4	2	0.6725146	4.220000	3.842975	2.500000	2.834711	2.960000	3.148760	2.867347	3.150000	4.430000	4.024793
6	4	1	0.5260116	4.587302	4.518182	1.968254	1.950000	3.055556	3.090909	2.762295	2.740566	4.137097	3.972727
7	4	1	0.9629630	4.652174	4.725806	1.739130	1.709677	3.326087	3.048387	2.195652	2.266667	2.869565	3.161290
8	3	1	0.4301676	4.301205	4.359375	2.259036	2.234375	3.259036	3.130208	2.814815	2.803191	3.855422	3.911458
9	3	2	0.3517241	4.441176	4.446602	2.029412	1.864078	3.251337	3.174757	2.795455	2.773196	3.856757	3.828431
10	1	3	0.5229358	4.476190	4.739130	1.809524	1.500000	3.277778	3.217391	2.557377	2.125000	3.317460	3.065217
11	1	1	0.2285714	4.650602	4.745614	1.656627	1.508772	3.307229	3.289474	2.416667	2.277778	3.343373	3.254386
12	4	2	0.4032258	4.416129	4.338710	1.954839	2.016129	3.241935	3.284946	2.791946	2.877778	3.810458	3.875000
13	3	1	0.5000000	4.329670	4.315603	2.170330	2.148936	3.159341	3.198582	2.857143	2.912409	4.027473	3.946429
14	2	3	0.5238095	4.379562	4.541096	2.135036	2.095890	3.175182	3.130137	2.766917	2.794521	4.103704	4.075342

#dodawanie, odejmowanie, mnożenie i dzielenie wartości pomiędzy kolumnami

```
library(dplyr)
```

```
summitssummary<-change.per.summits.merged%>%
```

```
  group_by(Access)%>%
```

```
  mutate(changeL=Lnew-Lo1d, changeT=Tnew+To1d,  
changeM=Mnew*Mo1d, changeN=Nnew/No1d,  
changeR=Rnew+Ro1d)%>%
```

```
  select(-(Lo1d:Rnew))
```

```
> summitssummary  
# A tibble: 14 x 8  
# Groups:   Access [4]  
   Access Tourism BCdist changeL changeT changeM changeN changer  
   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>  
1       2       1 0.366 0.110      3.29    10.5    0.926    6.91  
2       3       1 0.530 0.0686      3.85     9.50    0.972    8.26  
3       4       2 0.682 -0.357      4.61    11.0    1.14     6.82  
4       2       2 0.541 0.287      3.28    10.4    0.850    6.45  
5       4       2 0.673 -0.377      5.33     9.32    1.10     8.45  
6       4       1 0.526 -0.0691      3.92     9.44    0.992    8.11  
7       4       1 0.963 0.0736      3.45    10.1    1.03     6.03  
8       3       1 0.430 0.0582      4.49    10.2    0.996    7.77  
9       3       2 0.352 0.00543      3.89    10.3    0.992    7.69  
10      1       3 0.523 0.263      3.31    10.5    0.831    6.38  
11      1       1 0.229 0.0950      3.17    10.9    0.943    6.60  
12      4       2 0.403 -0.0774      3.97    10.6    1.03     7.69  
13      3       1 0.5    -0.0141      4.32    10.1    1.02     7.97  
14      2       3 0.524 0.162      4.23     9.94    1.01     8.18
```

#wyłanianie z tabeli konkretnych wartości, grupowanie i podsumowywanie danych: chcemy np. znaleźć dla każdego roku liczbę dni ze średnią roczną temperaturą powietrza >5 stopni Celsjusza

```
> meteo.kasprowy
```

	station	date	temp	day	month	year
1	Kasprowy wierch	01.01.1966	-7.0	1	1	1966
2	Kasprowy wierch	02.01.1966	-6.1	2	1	1966
3	Kasprowy wierch	03.01.1966	-7.4	3	1	1966
4	Kasprowy wierch	04.01.1966	-12.0	4	1	1966
5	Kasprowy wierch	05.01.1966	-19.0	5	1	1966
6	Kasprowy wierch	06.01.1966	-20.2	6	1	1966
7	Kasprowy wierch	07.01.1966	-17.2	7	1	1966
8	Kasprowy wierch	08.01.1966	-13.9	8	1	1966
9	Kasprowy wierch	09.01.1966	-15.9	9	1	1966
10	Kasprowy wierch	10.01.1966	-16.4	10	1	1966
11	Kasprowy wierch	11.01.1966	-17.3	11	1	1966
12	Kasprowy wierch	12.01.1966	-12.0	12	1	1966
13	Kasprowy wierch	13.01.1966	-15.8	13	1	1966
14	Kasprowy wierch	14.01.1966	-17.2	14	1	1966
15	Kasprowy wierch	15.01.1966	-15.3	15	1	1966
16	Kasprowy wierch	16.01.1966	-13.3	16	1	1966
17	Kasprowy wierch	17.01.1966	-11.2	17	1	1966
18	Kasprowy wierch	18.01.1966	-11.5	18	1	1966
19	Kasprowy wierch	19.01.1966	-7.6	19	1	1966
20	Kasprowy wierch	20.01.1966	-8.6	20	1	1966

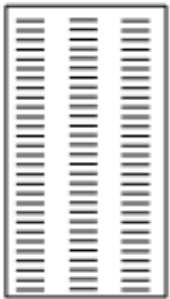
```
meteo2<-meteo%>%filter(temp>5)%>%group_by(year,  
station)%>%summarise(length=n(), mean.temp=mean(temp))
```

```
> meteo2  
# A tibble: 66 x 4  
# Groups:   year [66]  
   year station length mean.temp  
   <int> <fct>      <int>      <dbl>  
1  1951 Kasprowy wierch    275      -2.49  
2  1952 Kasprowy wierch    275      -4.80  
3  1953 Kasprowy wierch    249      -3.98  
4  1954 Kasprowy wierch    272      -4.50  
5  1955 Kasprowy wierch    271      -4.66  
6  1956 Kasprowy wierch    278      -5.29  
7  1957 Kasprowy wierch    269      -2.98  
8  1958 Kasprowy wierch    258      -4.18  
9  1959 Kasprowy wierch    288      -2.41  
10 1960 Kasprowy wierch    287      -2.98  
# ... with 56 more rows
```



## #macierze (matrix)

- struktura do przechowywania wartości tego samego typu w postaci tabelarycznej (liczbowe lub znakowe, nigdy oba jednocześnie).
- Inaczej, jest to seria wektorów w systemie kolumn i wierszy.
- Matrix może mieć różną liczbę kolumn i wierszy, ale każdy wiersz musi mieć taką samą liczbę kolumn, a kolumna taką samą liczbę wierszy.
- Jeżeli nie znamy jakiejś wartości, wpisujemy **NA**



	A1	A2	A6	B1	B4	B5	B6	C2	C3	C4	C5
A1	0.0000000	0.2755102	0.3106796	0.4187192	0.3554502	0.3665158	0.3580247	0.4009217	0.2735849	0.3446809	0.3333333
A2	0.2755102	0.0000000	0.3689320	0.4187192	0.3364929	0.3755656	0.3580247	0.3548387	0.2924528	0.3531915	0.3939394
A6	0.3106796	0.3689320	0.0000000	0.3051643	0.3122172	0.3073593	0.3604651	0.3127753	0.2702703	0.2653061	0.2884615
B1	0.4187192	0.4187192	0.3051643	0.0000000	0.3027523	0.2894737	0.4556213	0.3125000	0.3881279	0.3223140	0.3853659
B4	0.3554502	0.3364929	0.3122172	0.3027523	0.0000000	0.3305085	0.4463277	0.3534483	0.3215859	0.3040000	0.3802817
B5	0.3665158	0.3755656	0.3073593	0.2894737	0.3305085	0.0000000	0.3903743	0.3140496	0.3586498	0.2615385	0.2914798
B6	0.3580247	0.3580247	0.3604651	0.4556213	0.4463277	0.3903743	0.0000000	0.4535519	0.3820225	0.3930348	0.3658537
C2	0.4009217	0.3548387	0.3127753	0.3125000	0.3534483	0.3140496	0.4535519	0.0000000	0.2875536	0.2421875	0.3515982
C3	0.2735849	0.2924528	0.2702703	0.3881279	0.3215859	0.3586498	0.3820225	0.2875536	0.0000000	0.2191235	0.3644860
C4	0.3446809	0.3531915	0.2653061	0.3223140	0.3040000	0.2615385	0.3930348	0.2421875	0.2191235	0.0000000	0.3248945
C5	0.3333333	0.3939394	0.2884615	0.3853659	0.3802817	0.2914798	0.3658537	0.3515982	0.3644860	0.3248945	0.0000000
C7	0.4218009	0.4502370	0.3303167	0.3394495	0.4336283	0.3220339	0.4124294	0.4051724	0.3920705	0.3600000	0.3239437
	C7	D11	D3	D4	D5	E10	E11	E2	E3	E4	E5
A1	0.4218009	0.4040404	0.3744076	0.4297872	0.4382979	0.3951220	0.4773869	0.5642458	0.5263158	0.4187192	0.3518519
A2	0.4502370	0.3838384	0.3459716	0.3957447	0.4553191	0.4048780	0.4874372	0.5195531	0.5052632	0.3990148	0.4074074
A6	0.3303167	0.3557692	0.3122172	0.3551020	0.3551020	0.3860465	0.3875598	0.4920635	0.4200000	0.3521127	0.3185841
B1	0.3394495	0.3658537	0.3119266	0.3140496	0.2975207	0.2830189	0.3203883	0.3225806	0.2588832	0.2666667	0.3183857
B4	0.4336283	0.3896714	0.3628319	0.3360000	0.4000000	0.4090909	0.3925234	0.4948454	0.4146341	0.3577982	0.4112554
B5	0.3220339	0.3542601	0.2966102	0.2923077	0.2538462	0.2695652	0.3125000	0.4215686	0.3395349	0.3421053	0.2946058
B6	0.4124294	0.4024390	0.4237288	0.4726368	0.4925373	0.4035088	0.4787879	0.4896552	0.4743590	0.4437870	0.4395604
C2	0.4051724	0.3424658	0.2155172	0.2812500	0.3046875	0.3539823	0.3363636	0.3400000	0.2985782	0.2410714	0.2827004
C3	0.3920705	0.3457944	0.3127753	0.3386454	0.4103586	0.3484163	0.3953488	0.4871795	0.4660194	0.3333333	0.3189655
C4	0.3600000	0.3502110	0.2880000	0.2846715	0.3211679	0.3114754	0.3445378	0.4128440	0.3973799	0.3223140	0.3254902
C5	0.3239437	0.3600000	0.3239437	0.3248945	0.3080169	0.3333333	0.4129353	0.4143646	0.3750000	0.3463415	0.3119266
C7	0.0000000	0.3708920	0.3628319	0.3120000	0.3200000	0.3090909	0.3738318	0.4226804	0.3853659	0.3577982	0.3073593

# Generalna rada

Na każdym etapie pracy z danymi, tzn. wczytanie, doklejanie/odejmowanie wierszy/kolumn, transpozycja, przekształcanie tabeli wąskich w szerokie itd., używając funkcji: `head`, `str`, `summary`, `class` i `dim`, należy sprawdzać, czy dane zostały wczytane/przekształcone prawidłowo



**BSS**  
BIAŁOWIESKA SZKOŁA STATYSTYKI