

R cheat sheet

Podstawy:

x<-1 #przypisywanie zmiennych (skrót LAlt + -)
10 %% 3 #reszta z dzielenia (modulo)
10 %/% 3 #dzielenie całkowite (bez ułamka)
sqrt(x) #pierwiastek
abs(x) #wartość bezwzględna
round(x, 2) #zaokrąglenie do podanej ilości miejsc po przecinku (tu do 2)

c("sth", "sth2") #tworzy wektor
sum(x) #funkcja sumy, do liczenia ile elementów spełnia dany warunek
length(x) #długość danych
nrow(x) #liczba wierszy
ncol(x) #liczba kolumn
names(x) #nazwy kolumn
rownames(x) #nazwy wierszy
head(x, 58) #pokazuje pierwsze 58 wartości z x
is.na(x) #sprawdza czy jakies wartosci sa puste

x > 0 | x < 10 #znaczek | to LUB
x > 0 & x < 10 #znaczek & I
!x == 11 #znaczek ! to zaprzeczenie

Trochę statystyki

mean(x) #średnia
median(x) #mediana
sd(x) #odchylenie
var(x) #wariancja
quantile(x, 0.75) #kwartyl (w miejsce 0.75 wybierasz który chcesz)
min(x) **max(x)**

Przydatne funkcje:

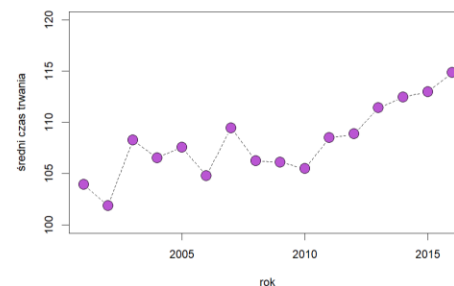
class(x) #wyświetla rodzaj danych
str(x) #nazwy kolumn, typy danych i początkowe dane z każdej kolumny
summary(x) #zlicza statystyki każdej z kolumn
seq(from = 1, to = 10, by = 2) #tworzy nam ciąg znaków (od ; do ;co ile)

table(x,y, useNA = "ifany") #sprawdzanie liczebności, parametr dodatkowo dokłada do tabeli wartości NA jeśli takowe występują
prop.table(table(x,y), marg = 1-3) #jak table tylko nie liczy ilości a zawartość procentową, argumentem margin można manipulować czy odnosimy się do wierszy czy kolumn
unique(x) - #pomija powtarzające się elementy
order(x, decreasing = TRUE) #sortuje nasze dane rosnąco, jednak zmienna decreasing powoduje sortowanie malejąco
cut(x, breaks = 10) #podział na przedziały (x jest wektorem), argument breaks wskazuje jak długie mają być
rnorm(100, mean = 5, sd = 2) #100 liczb z rozkładu normalnego

tapply(x,y,mean,na.rm=TRUE) #tworzy nam "tabelę" i liczy z niej w tym wypadku średnią (sd, var, itd też można użyć)
dim(x) #sprawdza wymiary ramki
x\$kolumna1 #odnosimy się do konkretnej kolumny w ramce danych
x[x\$kolumna1 > 6, "kolumna1"] #przed przecinkiem warunek dla wierszy, po przecinku które kolumny
colnames(x)[colnames(x) == "nazwa_kolumny"] <- "nowa_nazwa_kolumny" #zamienia nazwę konkretnej kolumny na inną

Przykłady:

tt[tt\$Age >= 70 & !is.na(tt\$Age), c("Name", "Sex", "Pclass", "Age")] #wyświetl nazwiska, płeć, klasę i wiek osób mających przynajmniej 70 lat
prop.table(table(tt\$Survived, tt\$Pclass), margin=1)*100 #dla osób, które przeżyły i nie przeżyły rozkład procentowy klas, którymi podróżowali
prop.table(table(tt\$Survived, complete.cases(tt)), margin=2)*100 #jaki procent osób przeżył i nie przeżył wśród tych osób, o których mamy wszystkie informacje oraz tych, których rekordy mają braki
head(tt[order(tt\$Age), c("Name", "Age", "Sex", "Survived")], n=10) #Wyświetl imiona, wiek, płeć oraz informacje o przeżyciu 10 najmłodszych osób posortowanych rosnąco wg wieku
head(sort(tt\$Age, decreasing = TRUE), n=20) #Wyświetl w postaci wektora wiek 20 najstarszych osób posortowany malejąco
#Wyświetl średni wiek osób, którzy w imieniu mają ciąg znaków "Mr."
tt[grepl("Rev.", tt\$Name, fixed = TRUE), c("Name", "Age", "Pclass", "Survived")]
#Wśród pasażerów było 6 pastorów. Można ich zidentyfikować po tytule "Rev."
Wyświetl ich nazwiska, wiek, klasę oraz informację, czy przeżyli.
bikes\$season <- factor(bikes\$season, levels = 1:4, labels = c("winter", "spring", "summer", "autumn")) #Kolumnę season zamień na factor z wartościami "winter", "spring", "summer", "autumn"
bikes\$yr <- factor(bikes\$yr, levels = c(0,1), labels = c(2011, 2012))
#Kolumnę yr zamień na factor z wartościami 2011 i 2012.



plot(names(tapply(movies\$duration, movies\$year, mean)), tapply(movies\$duration, movies\$year, mean), ylab='średni czas trwania', xlab='rok', ylim=c(100,120), type='b', lty=2, pch=21, bg='mediumorchid', cex=2)

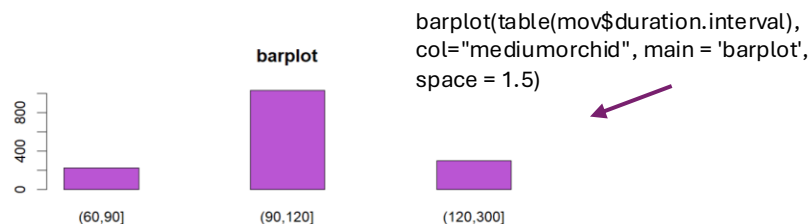
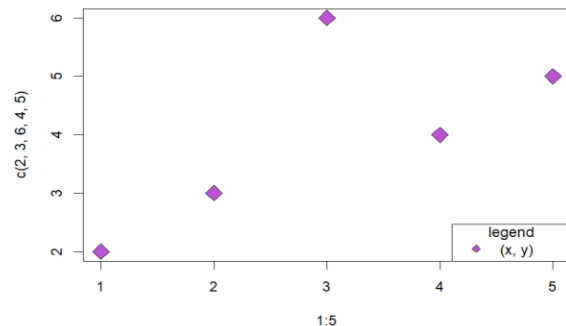
WAŻNE!

read.csv("data/x.csv", sep = ",")
#zapisuje nasz plik jako zmienną, separator wybieramy na podstawie tego co jest między naszymi danymi w tabeli
read.csv2('./data/movies.csv') #to samo tylko przecinek domyślnie jako separator dziesiętny
rbind(pokemon1_4, pokemon5_6) #łączy dane np z dwóch plików na podstawie wierszy
cbind(x, y) #to samo tylko dla kolumn (zamiast wierszy)

WYKRESY

```
plot(1:5, c(2, 3, 6, 4, 5), pch = 23, cex = 2, bg = "purple")
#wykres (podany ksztalt, wielkosc i wypelnienie)
legend("bottomright", "(x, y)", pch=23, pt.bg = "purple",
title= "legend") #tworzy legende do wykresu
```

```
abline(h = 0) #dodaje linie pozioma w
y=0 do istniejacego juz wykrsu
abline(v = 3) #dodaje linie pionowa w x=3
do istniejacego juz wykresu
boxplot(x ~ y, horizontal = TRUE)
#tworzy wykres pudełkowy, jesli wychodzi
dziwny to zamień zmienne kolejnoscia,
horizontal go odwraca na bok
barplot(x,y) # wykres słupkowy
hist(x,y) #histogram
```



```
barplot(table(mov$duration.interval),
col="mediumorchid", main = 'barplot',
space = 1.5)
```

Plot - rodzaje

type
what type of plot should be drawn. Possible types are

- "p" for points,
- "l" for lines,
- "b" for both,
- "c" for the lines part alone of "b",
- "o" for both 'overplotted',
- "h" for 'histogram' like (or 'high-density') vertical lines,
- "s" for stair steps,
- "S" for other steps, see 'Details' below,
- "n" for no plotting.

main
an overall title for the plot: see [title](#).

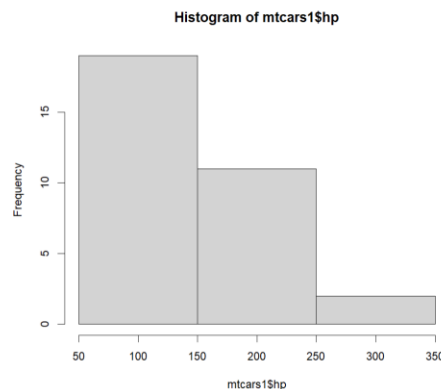
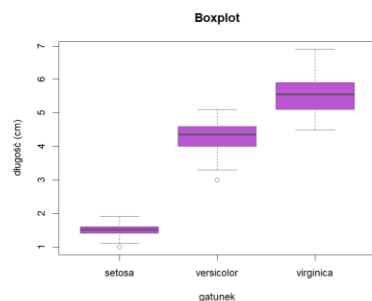
sub
a subtitle for the plot: see [title](#).

xlab
a title for the x axis: see [title](#).

ylab
a title for the y axis: see [title](#).

asp
the y/x aspect ratio, see [plot.window](#).

```
hist(mtcars1$hwp, breaks =
c(50, 150, 250, 350))
```



```
boxplot(iris$Petal.Length ~ iris$Species,
col = "mediumorchid", border = "#555555",
xlab = "gatunek", ylab = "długość (cm)",
main = "Boxplot")
```

```
grep("przykładowy tekst", x, value = TRUE) #pokazuje na
ktorej pozycji (o ile jest) znajduje sie nasz fragament tekstu w
tabeli, parametr value pozwala nam wyswietlic ten tekst
grepl("przykładowy tekst", x) #tworzy wektor logiczny
zwracajacy TRUE jesli fragament tekstu znajduje sie na jakiejś
pozycji
gsub("t", "r", x) # wszystkie "t" zamienione na "r"
rep("sth", times = 5) #powtarza określoną
zmienną x razy (tutaj 5)
paste0("Tom", "goat") #łączy podane teksty w jeden
nchar(x) #liczy ilość znaków
ifelse(x > 0, 1, 0) #sprawdzamy warunek, jesli jest spełniony,
to dostajemy wartość 1, w innym wypadku 0
as.factor(x) #zmienna zachowuje sie jak factor
as.integer(x) #zmienna zachowuje sie jak liczba
complete.cases(x) #zwraca TRUE gdy nie ma braków (na)
unlist(strsplit("a.b.c", "[.]")) #służy do rozłączenia ("a" "b" "c")
```

```
grep("^t", x, value = TRUE) # początek na "t"
grep("ek$", x, value = TRUE) # na końcu "ek"
grep("[0-9]", x, value = TRUE) # ciąg zawiera dowolną cyfrę
w dowolnym miejscu
ifelse(grepl("[0-9]", x), "zawiera liczbę", "nie zawiera
liczby") #sprawdza warunek i wyświetla wektor z
odpowiednimi napisami do każdego elementu
gsub("&", "and", grep("&", imdb$title, value = TRUE))
#wyświetla tytuły filmów, które oryginalnie zawierają symbol
"&", ale zamień ten symbol na wyraz "and"
x <- c("a", "aaa", "aa")
nchar(x) #wynik: 1 3 2
tt.complete <- tt[complete.cases(tt), ] #wybieramy tylko
kompletne rekordy
unlist(strsplit(im$Actors, ',')) #dzieli aktorów na
podstawie znaku ","
```

Jeśli nie wiesz jak coś działa - zaznacz daną funkcję i kliknij klawisz **F1** lub wpisz przed nią znak **?** i zatwierdź (CTRL + ENTER)

Powodzenia! Aleks Korzeniowski, Dominika Włodarz

KODY

PRZYKŁADY