

# Sławacka\_Weronika5

2023-11-15

#zadanie1

```
# a) Bez wprowadzania kodu do konsoli wskaż która z odpowiedzi jest poprawna.
# if (numer < 10) {
# if (numer < 5) {
# result <- "bardzo niski"
# } else {
# result <- "niski"
# }
# } else if (numer < 100) {
# result <- "sredni"
# } else {
# result <- "duzy"
# }
# print(result)
# 1. Jeśli numer będzie równy 6, w konsoli pojawi się „niski”.
# 2. Jeśli numer będzie równy 100, pojawi się „sredni”.
# 3. Jeśli numer będzie równy 0, nic się nie wyświetli.
# 4. Jeśli numer będzie równy 2500, R wyświetli błąd.

# Poprawna jest odpowiedź 1.
```

```
# b) Napisz kod który sprawdzi, które elementy wektora są parzyste, a które
# nieparzyste.
```

```
v <- c(1,7, 8,12,14,19);
for(i in 1:length(v)){
  if(v[i]%%2){
    cat(i, ". element wektora wynosi ", v[i]," i jest nieparzysty \n");
  }else {
    cat(i, ". element wektora wynosi ", v[i]," i jest parzysty \n");
  }
}
```

```
## 1 . element wektora wynosi 1 i jest nieparzysty
## 2 . element wektora wynosi 7 i jest nieparzysty
## 3 . element wektora wynosi 8 i jest parzysty
## 4 . element wektora wynosi 12 i jest parzysty
## 5 . element wektora wynosi 14 i jest parzysty
## 6 . element wektora wynosi 19 i jest nieparzysty
```

#zadanie2

*# a) Napisz pętlę while zaczynając od x=0. Pętla wyświetla liczby z krokiem 1  
# do wartości 20, ale omija numer 12.*

```
x <- 0;
while(x<20){
  x <- x+1;
  if(x==12){
    next;
  }
  print(x);
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
## [1] 11
## [1] 13
## [1] 14
## [1] 15
## [1] 16
## [1] 17
## [1] 18
## [1] 19
## [1] 20
```

*# b) Utwórz pętlę while z instrukcją warunkową. W pierwszym kroku przypisz  
# do obiektu i wartość 1. Dopóki i<=10, oraz jest podzielne przez 3: wyświetl  
# w konsoli wartość i przemnożoną przez 3. Gdy i będzie podzielne przez 8:  
# wyświetl w konsoli komunikat 'Koniec' oraz przerwij działanie pętli. Na koniec  
# nadpisz i o 1.*

```
i <- 1;
while(i<=10){
  if(i%%3==0){
    print(i*3);
  } else if (i%%8==0) {
    print("Koniec");
    break;
  }
  i <- i + 1;
}
```

```
## [1] 9
## [1] 18
## [1] "Koniec"
```

```
#zadanie3
```

```
# a) Napisz 2 pętle for które będą wyświetlać zmienną Displacement (disp)
# z mtcars.
```

```
mtcars;
```

```
##           mpg cyl  disp  hp drat   wt  qsec vs am gear carb
## Mazda RX4      21.0   6  160.0  110 3.90 2.620 16.46 0  1    4    4
## Mazda RX4 Wag  21.0   6  160.0  110 3.90 2.875 17.02 0  1    4    4
## Datsun 710     22.8   4  108.0   93 3.85 2.320 18.61 1  1    4    1
## Hornet 4 Drive  21.4   6  258.0  110 3.08 3.215 19.44 1  0    3    1
## Hornet Sportabout 18.7   8  360.0  175 3.15 3.440 17.02 0  0    3    2
## Valiant        18.1   6  225.0  105 2.76 3.460 20.22 1  0    3    1
## Duster 360     14.3   8  360.0  245 3.21 3.570 15.84 0  0    3    4
## Merc 240D      24.4   4  146.7   62 3.69 3.190 20.00 1  0    4    2
## Merc 230       22.8   4  140.8   95 3.92 3.150 22.90 1  0    4    2
## Merc 280       19.2   6  167.6  123 3.92 3.440 18.30 1  0    4    4
## Merc 280C      17.8   6  167.6  123 3.92 3.440 18.90 1  0    4    4
## Merc 450SE     16.4   8  275.8  180 3.07 4.070 17.40 0  0    3    3
## Merc 450SL     17.3   8  275.8  180 3.07 3.730 17.60 0  0    3    3
## Merc 450SLC    15.2   8  275.8  180 3.07 3.780 18.00 0  0    3    3
## Cadillac Fleetwood 10.4   8  472.0  205 2.93 5.250 17.98 0  0    3    4
## Lincoln Continental 10.4   8  460.0  215 3.00 5.424 17.82 0  0    3    4
## Chrysler Imperial 14.7   8  440.0  230 3.23 5.345 17.42 0  0    3    4
## Fiat 128       32.4   4   78.7   66 4.08 2.200 19.47 1  1    4    1
## Honda Civic    30.4   4   75.7   52 4.93 1.615 18.52 1  1    4    2
## Toyota Corolla 33.9   4   71.1   65 4.22 1.835 19.90 1  1    4    1
## Toyota Corona  21.5   4  120.1   97 3.70 2.465 20.01 1  0    3    1
## Dodge Challenger 15.5   8  318.0  150 2.76 3.520 16.87 0  0    3    2
## AMC Javelin    15.2   8  304.0  150 3.15 3.435 17.30 0  0    3    2
## Camaro Z28     13.3   8  350.0  245 3.73 3.840 15.41 0  0    3    4
## Pontiac Firebird 19.2   8  400.0  175 3.08 3.845 17.05 0  0    3    2
## Fiat X1-9      27.3   4   79.0   66 4.08 1.935 18.90 1  1    4    1
## Porsche 914-2  26.0   4  120.3   91 4.43 2.140 16.70 0  1    5    2
## Lotus Europa   30.4   4   95.1  113 3.77 1.513 16.90 1  1    5    2
## Ford Pantera L  15.8   8  351.0  264 4.22 3.170 14.50 0  1    5    4
## Ferrari Dino   19.7   6  145.0  175 3.62 2.770 15.50 0  1    5    6
## Maserati Bora   15.0   8  301.0  335 3.54 3.570 14.60 0  1    5    8
## Volvo 142E     21.4   4  121.0  109 4.11 2.780 18.60 1  1    4    2
```

```
# Pętla 1: będzie wyświetlać tylko obserwacje większe lub równe 160 dla
# zmiennej disp.
```

```
for(i in 1:length(mtcars$disp)){
  if(mtcars$disp[i] >= 160){
    print(mtcars$disp[i]);
  }
}
```

```
## [1] 160
## [1] 160
## [1] 258
```

```
## [1] 360
## [1] 225
## [1] 360
## [1] 167.6
## [1] 167.6
## [1] 275.8
## [1] 275.8
## [1] 275.8
## [1] 472
## [1] 460
## [1] 440
## [1] 318
## [1] 304
## [1] 350
## [1] 400
## [1] 351
## [1] 301
```

*# Pętla 2: wypisuje wartości i zatrzyma się gdy obserwacja będzie niższa niż 160 dla zmiennej disp.*

```
for(i in 1:length(mtcars$disp)){
  if(mtcars$disp[i]<160){
    break;
  }
  print(mtcars$disp[i]);
}
```

```
## [1] 160
## [1] 160
```

*# b) Użyj zbioru danych rivers (dostępny bezpośrednio w R). Napisz 1 pętlę która:  
# Dla rzek krótszych niż 500, wyświetli się komunikat „krotka rzeka”  
# Dla rzek dłuższych niż 1000, wyświetli się „długa rzeka”.  
# Dla rzek pomiędzy, wyświetli się oryginalna wartość długości rzeki.*

```
rivers;
```

```
## [1] 735 320 325 392 524 450 1459 135 465 600 330 336 280 315 870
## [16] 906 202 329 290 1000 600 505 1450 840 1243 890 350 407 286 280
## [31] 525 720 390 250 327 230 265 850 210 630 260 230 360 730 600
## [46] 306 390 420 291 710 340 217 281 352 259 250 470 680 570 350
## [61] 300 560 900 625 332 2348 1171 3710 2315 2533 780 280 410 460 260
## [76] 255 431 350 760 618 338 981 1306 500 696 605 250 411 1054 735
## [91] 233 435 490 310 460 383 375 1270 545 445 1885 380 300 380 377
## [106] 425 276 210 800 420 350 360 538 1100 1205 314 237 610 360 540
## [121] 1038 424 310 300 444 301 268 620 215 652 900 525 246 360 529
## [136] 500 720 270 430 671 1770
```

```
for(i in 1:length(rivers)){
  if(rivers[i]<500){
    print("krotka rzeka");
  }
```

```

}
else if(rivers[i]>1000){
    print("dluga rzeka");
}
else print(rivers[i]);
}

```

```

## [1] 735
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] 524
## [1] "krotka rzeka"
## [1] "dluga rzeka"
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] 600
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] 870
## [1] 906
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] 1000
## [1] 600
## [1] 505
## [1] "dluga rzeka"
## [1] 840
## [1] "dluga rzeka"
## [1] 890
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] 525
## [1] 720
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] 850
## [1] "krotka rzeka"
## [1] 630
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] 730
## [1] 600
## [1] "krotka rzeka"

```

```

## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] 710
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] 680
## [1] 570
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] 560
## [1] 900
## [1] 625
## [1] "krotka rzeka"
## [1] "dluga rzeka"
## [1] "dluga rzeka"
## [1] "dluga rzeka"
## [1] "dluga rzeka"
## [1] "dluga rzeka"
## [1] 780
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] 760
## [1] 618
## [1] "krotka rzeka"
## [1] 981
## [1] "dluga rzeka"
## [1] 500
## [1] 696
## [1] 605
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] "dluga rzeka"
## [1] 735
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] "dluga rzeka"
## [1] 545
## [1] "krotka rzeka"

```

```
## [1] "dluga rzeka"
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] 800
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] 538
## [1] "dluga rzeka"
## [1] "dluga rzeka"
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] 610
## [1] "krotka rzeka"
## [1] 540
## [1] "dluga rzeka"
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] 620
## [1] "krotka rzeka"
## [1] 652
## [1] 900
## [1] 525
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] 529
## [1] 500
## [1] 720
## [1] "krotka rzeka"
## [1] "krotka rzeka"
## [1] 671
## [1] "dluga rzeka"
```

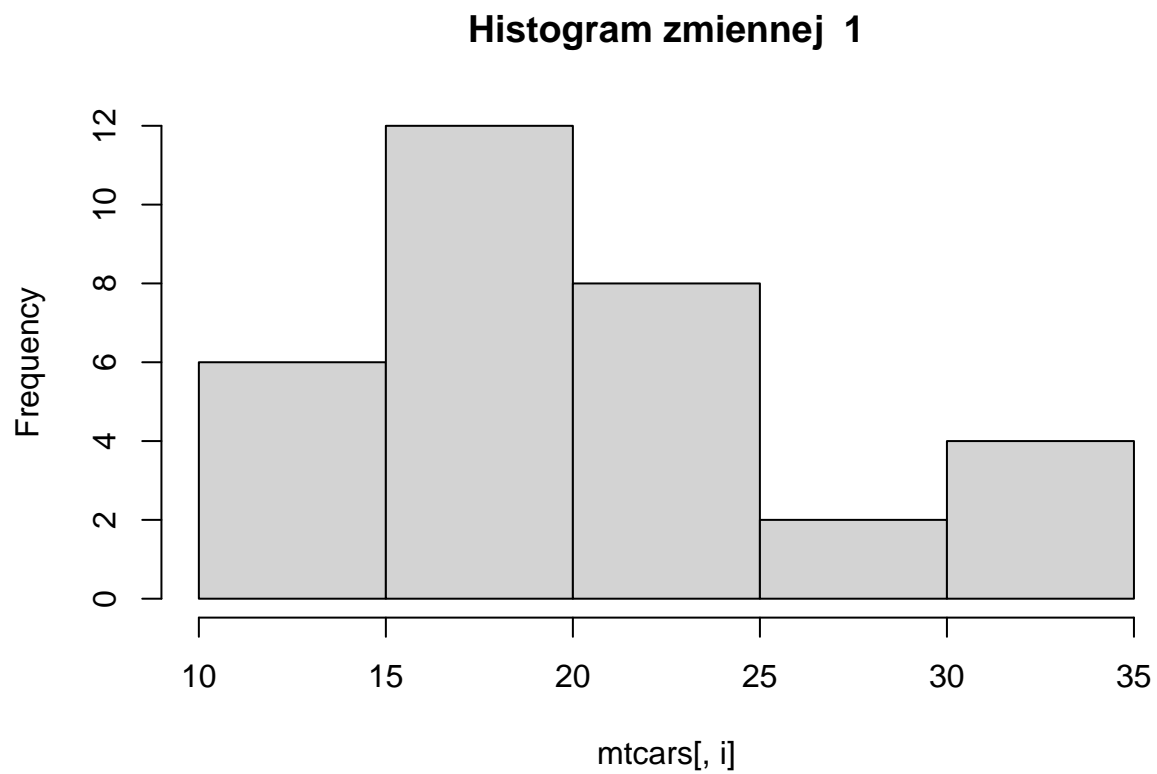
```
# c) Użyj instrukcji for aby dla wszystkich zmiennych numerycznych z mtcars
# utworzyć 2 wykresy:
# Histogram: hist(nazwa_zmiennej)
# wykres ramka-wąsy boxplot(nazwa_zmiennej)

str(mtcars);
```

```
## 'data.frame':   32 obs. of  11 variables:
## $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
## $ cyl : num   6  6  4  6  8  6  8  4  4  6 ...
## $ disp: num  160 160 108 258 360 ...
## $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
```

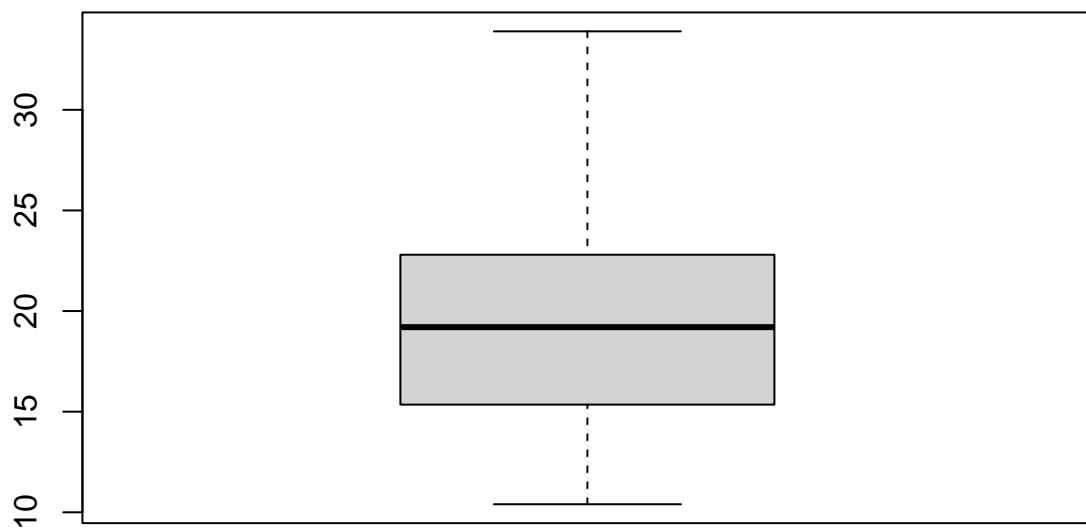
```
## $ drat: num 3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
## $ wt : num 2.62 2.88 2.32 3.21 3.44 ...
## $ qsec: num 16.5 17 18.6 19.4 17 ...
## $ vs : num 0 0 1 1 0 1 0 1 1 1 ...
## $ am : num 1 1 1 0 0 0 0 0 0 0 ...
## $ gear: num 4 4 4 3 3 3 3 4 4 4 ...
## $ carb: num 4 4 1 1 2 1 4 2 2 4 ...
```

```
for(i in 1:ncol(mtcars)){
  hist(mtcars[,i], main=paste("Histogram zmiennej ", i));
  boxplot(mtcars[,i], main=paste("Wykres zmiennej ", i));
}
```

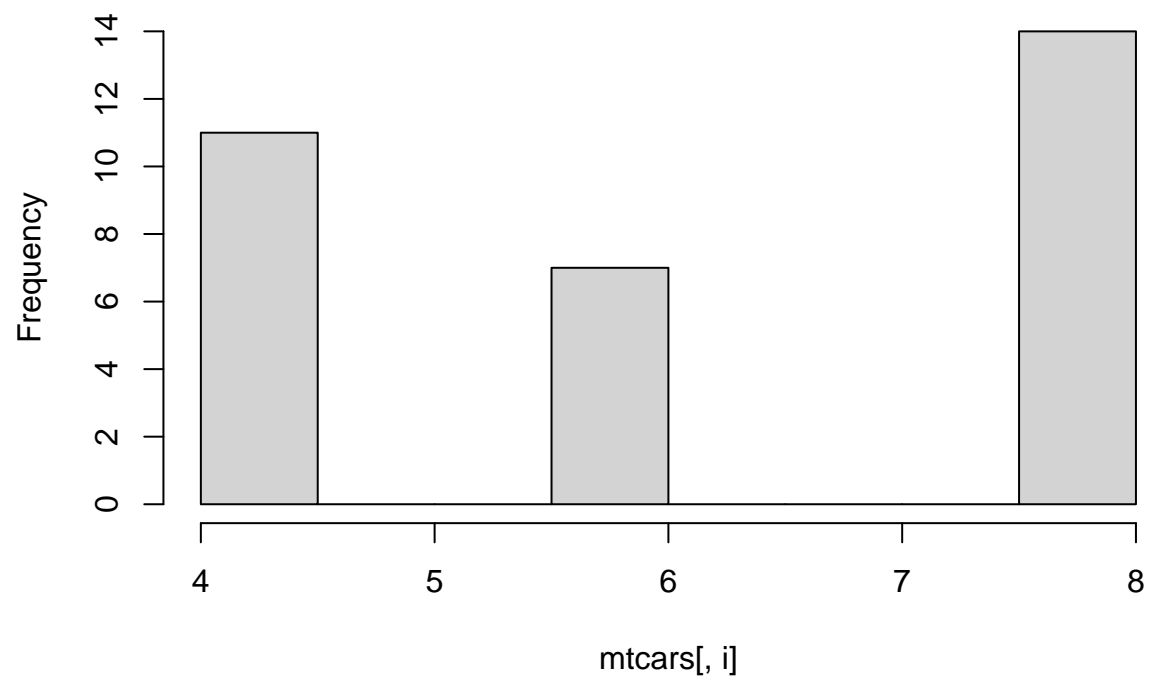




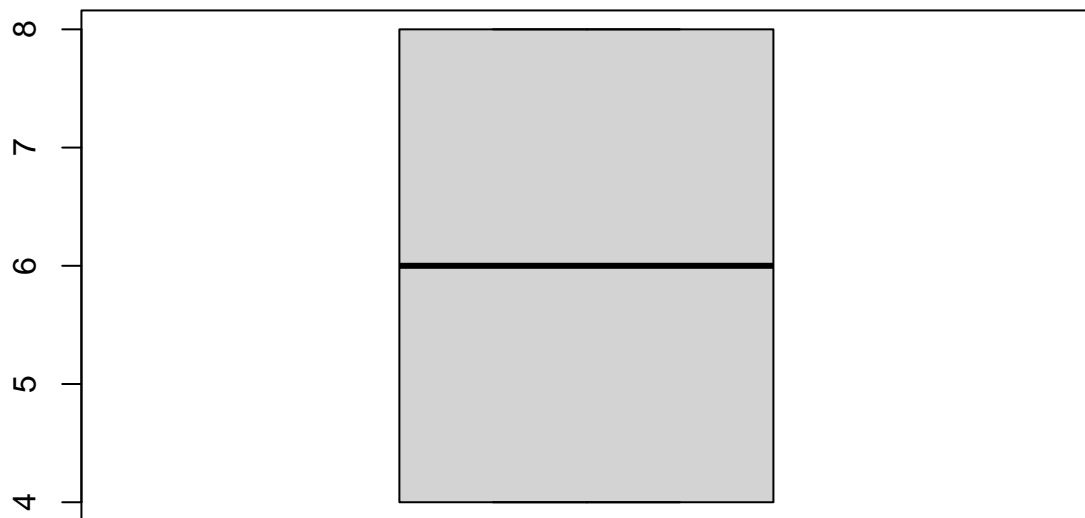
**Wykres zmiennej 1**



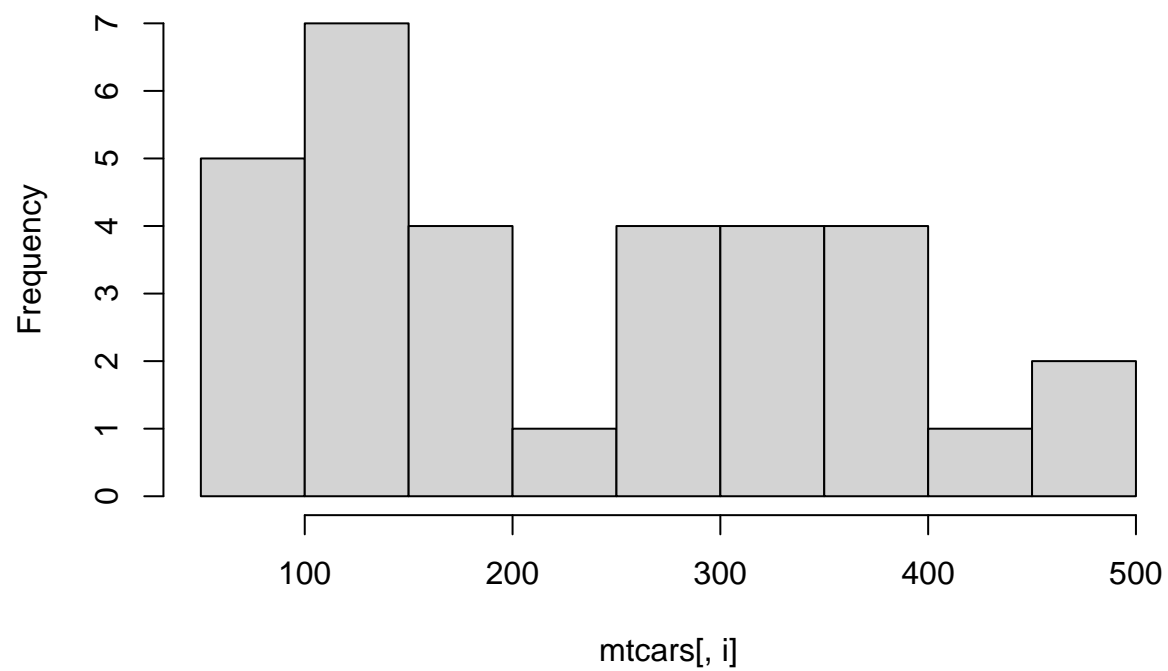
**Histogram zmiennej 2**



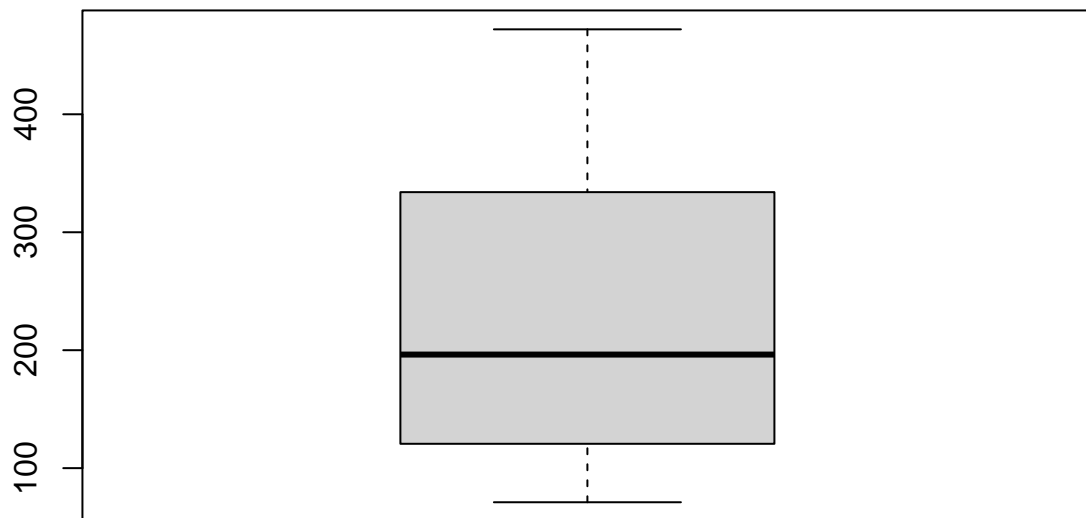
**Wykres zmiennej 2**



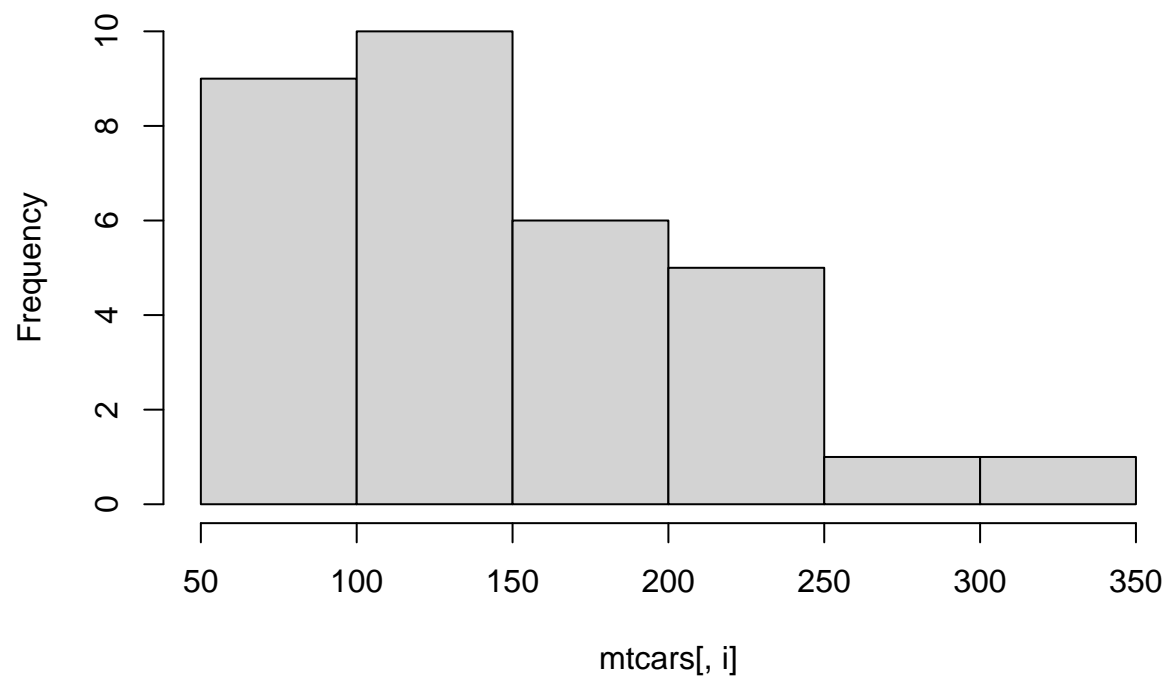
**Histogram zmiennej 3**



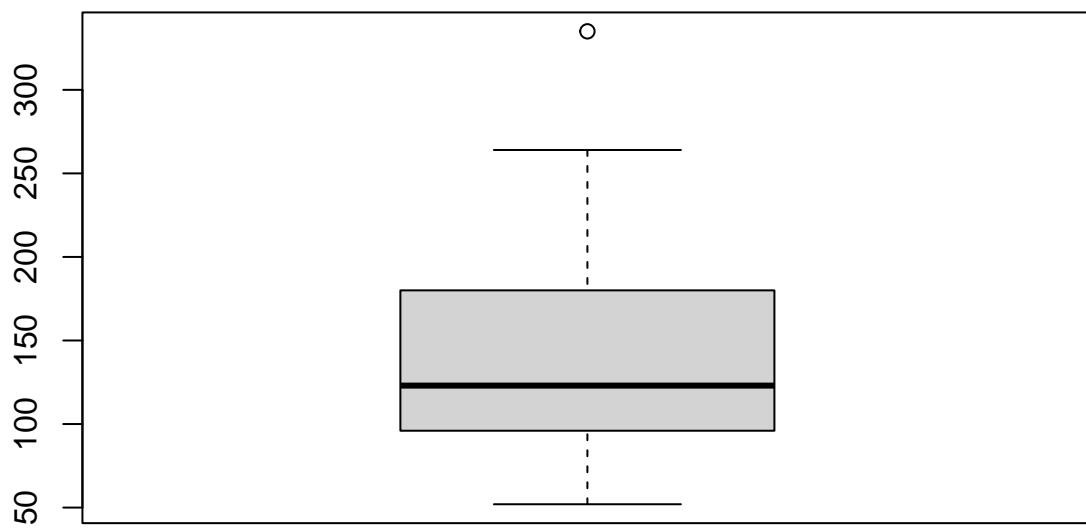
**Wykres zmiennej 3**



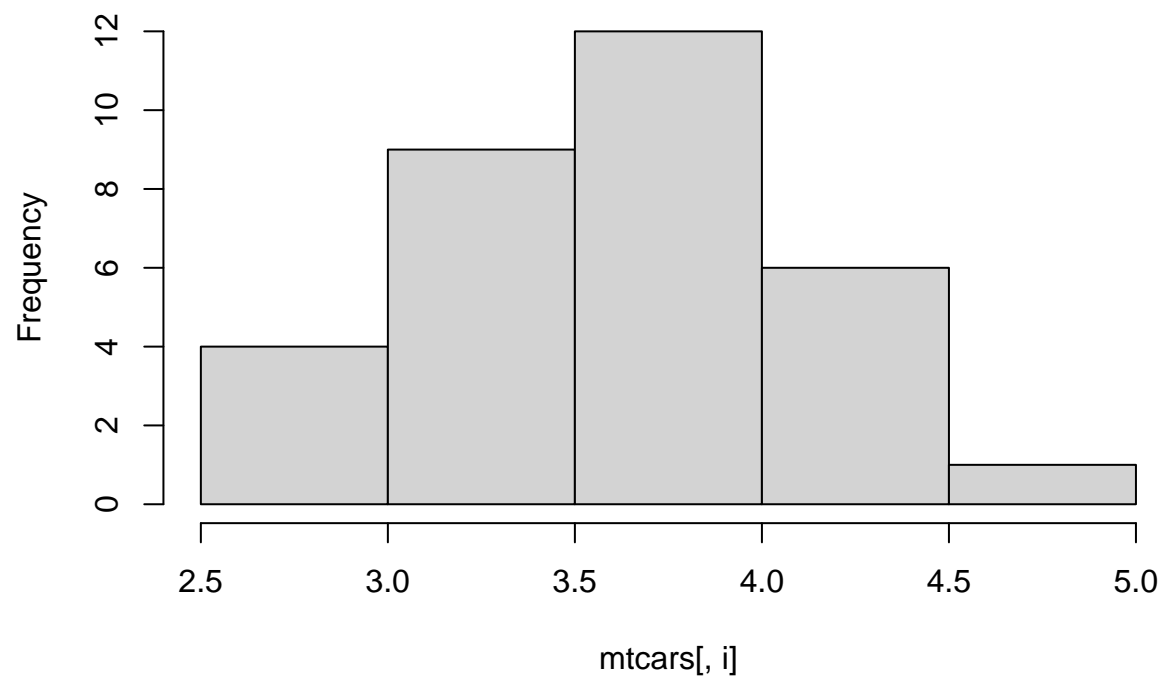
**Histogram zmiennej 4**



**Wykres zmiennej 4**

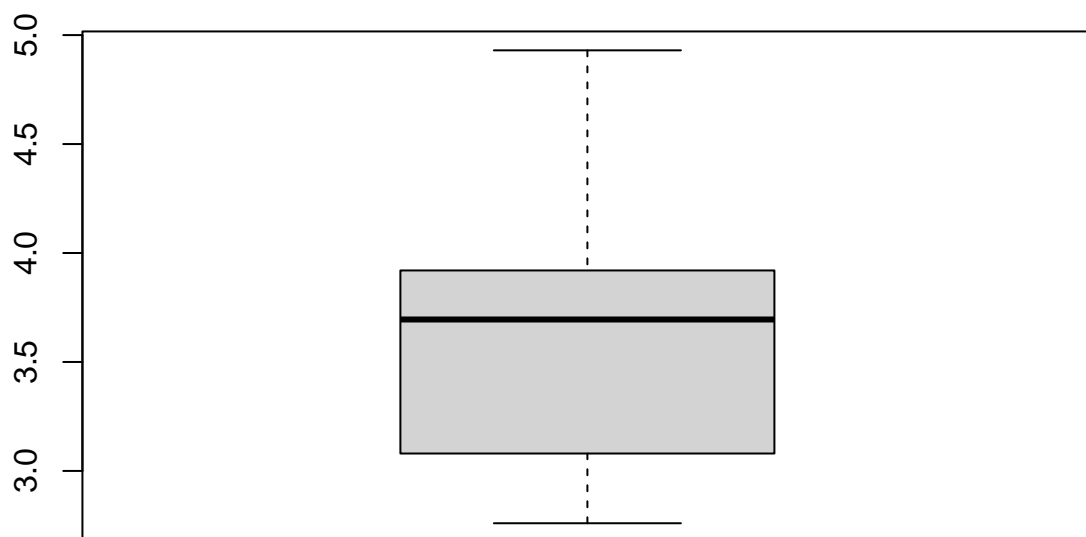


**Histogram zmiennej 5**

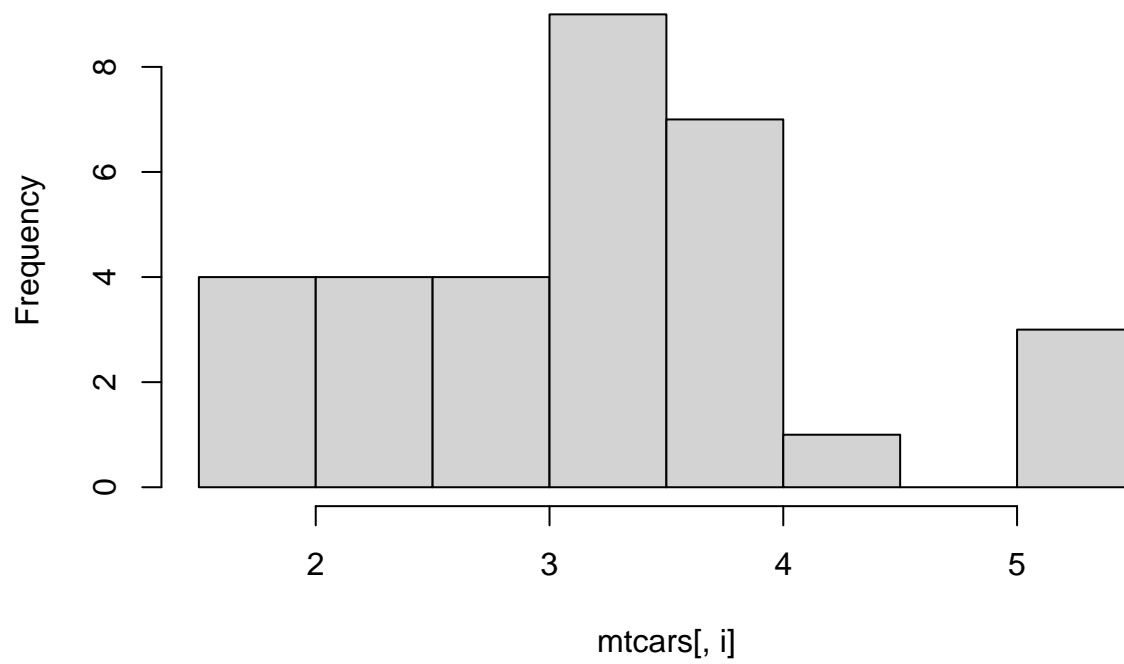




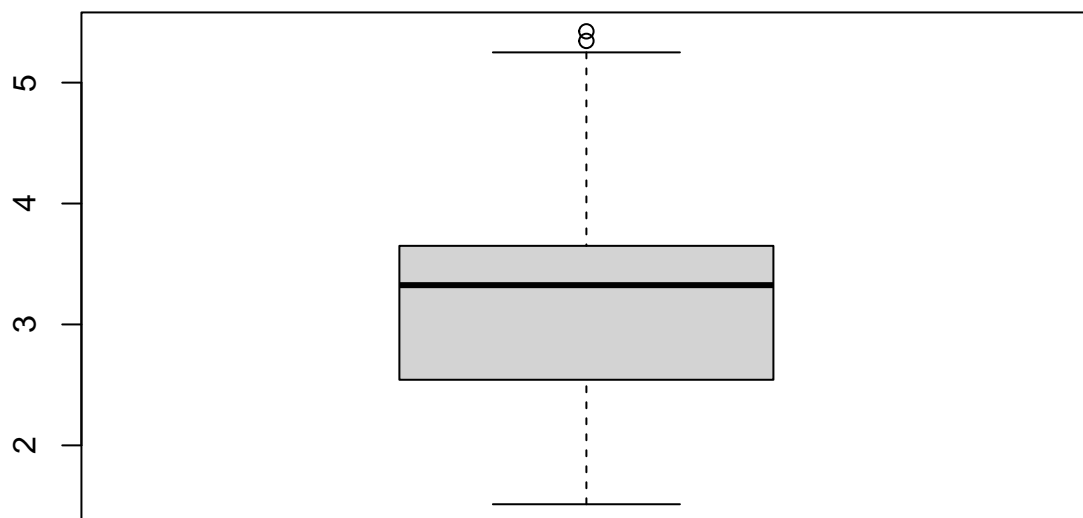
**Wykres zmiennej 5**



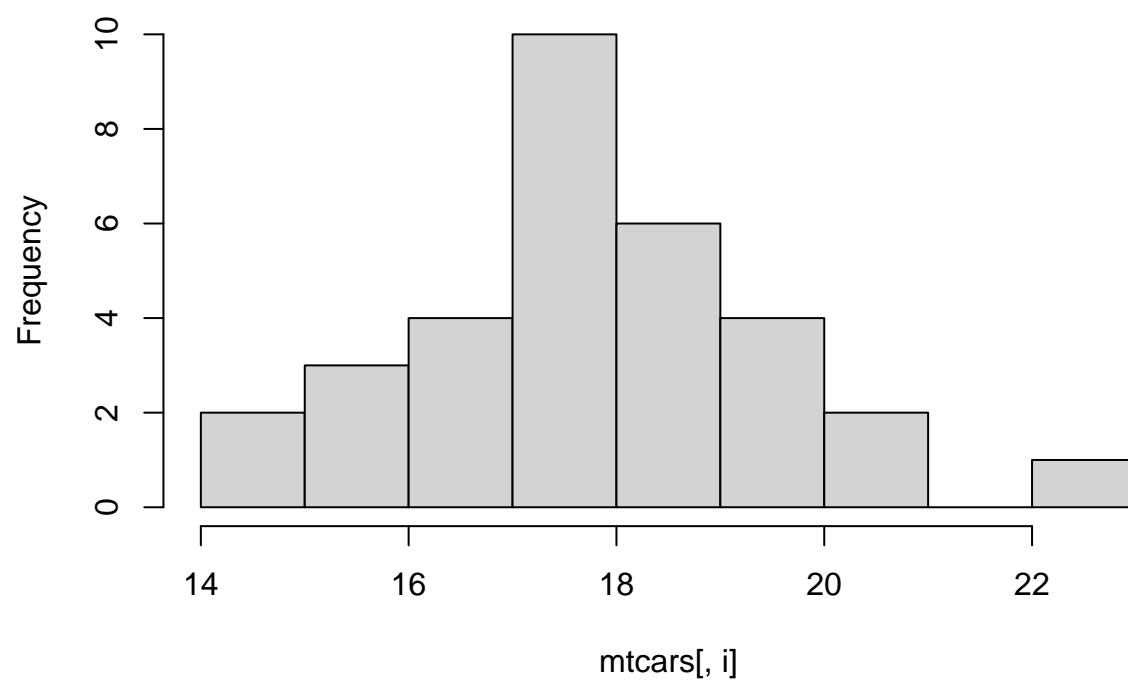
**Histogram zmiennej 6**



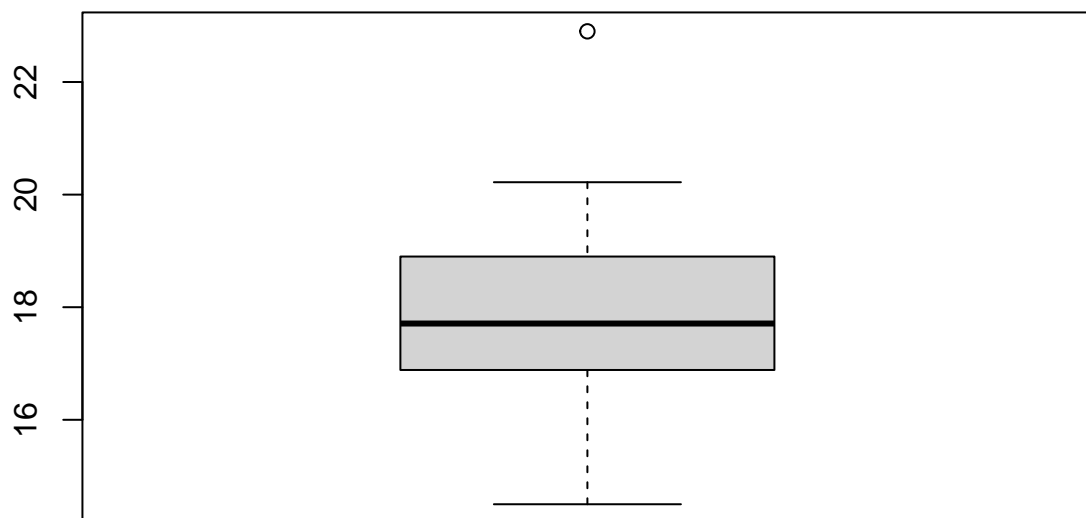
**Wykres zmiennej 6**



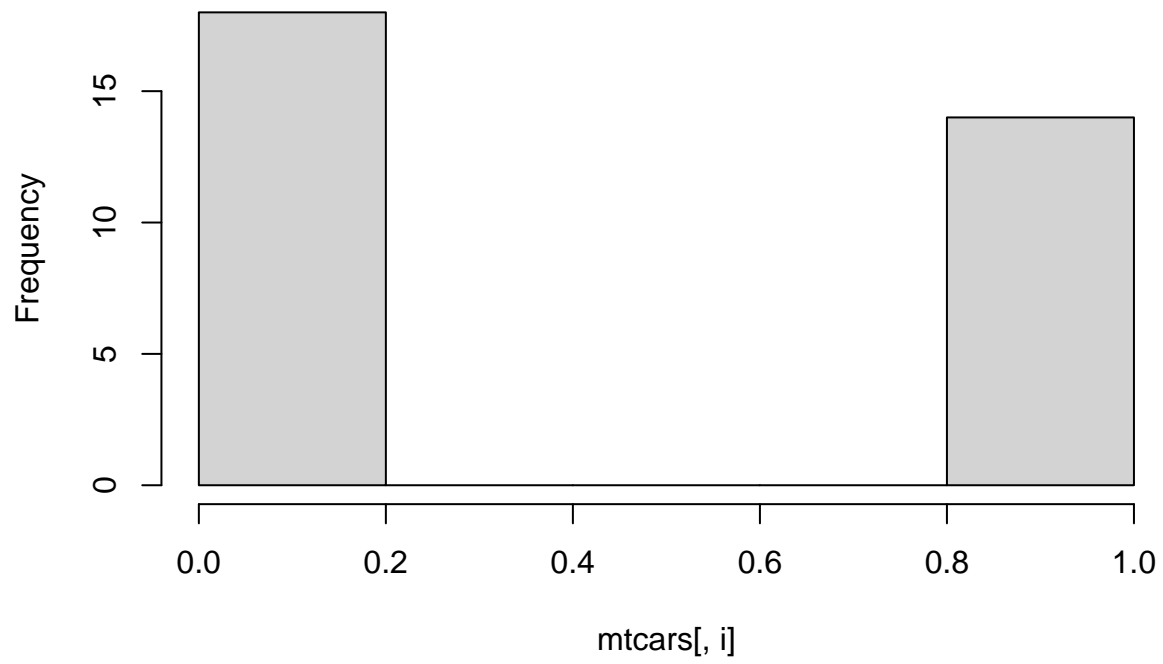
**Histogram zmiennej 7**



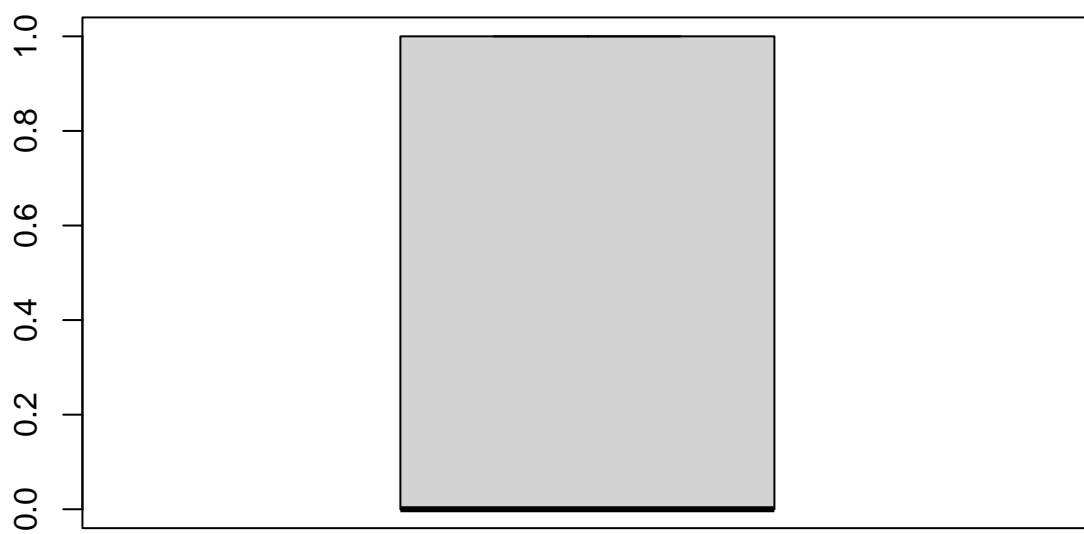
**Wykres zmiennej 7**



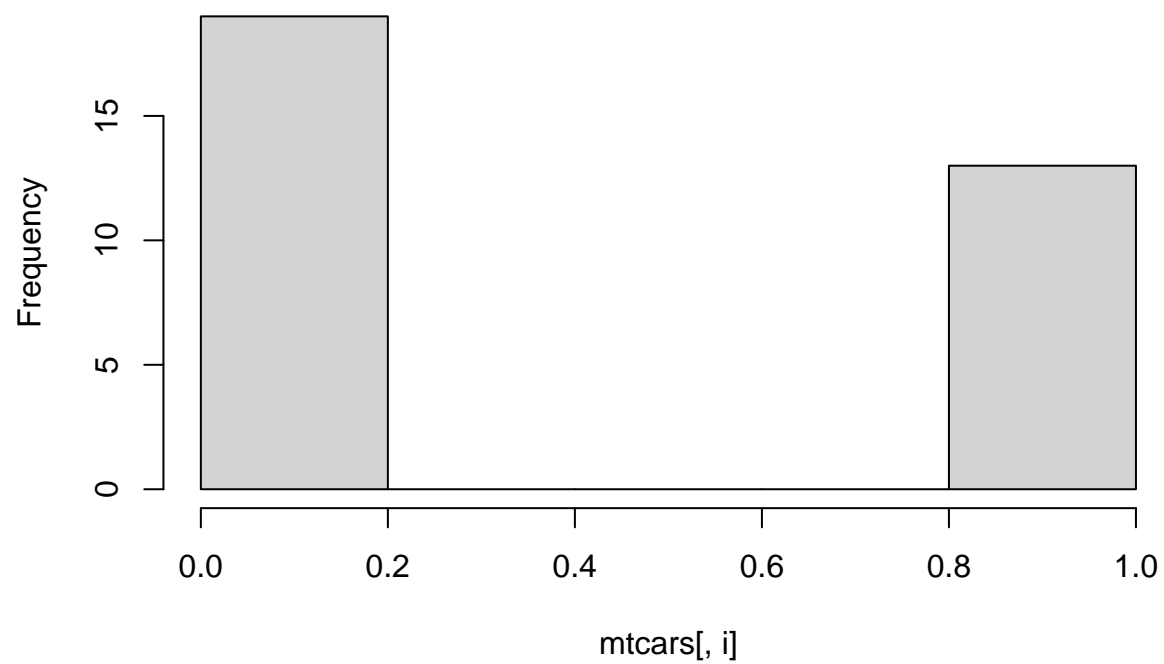
**Histogram zmiennej 8**



**Wykres zmiennej 8**

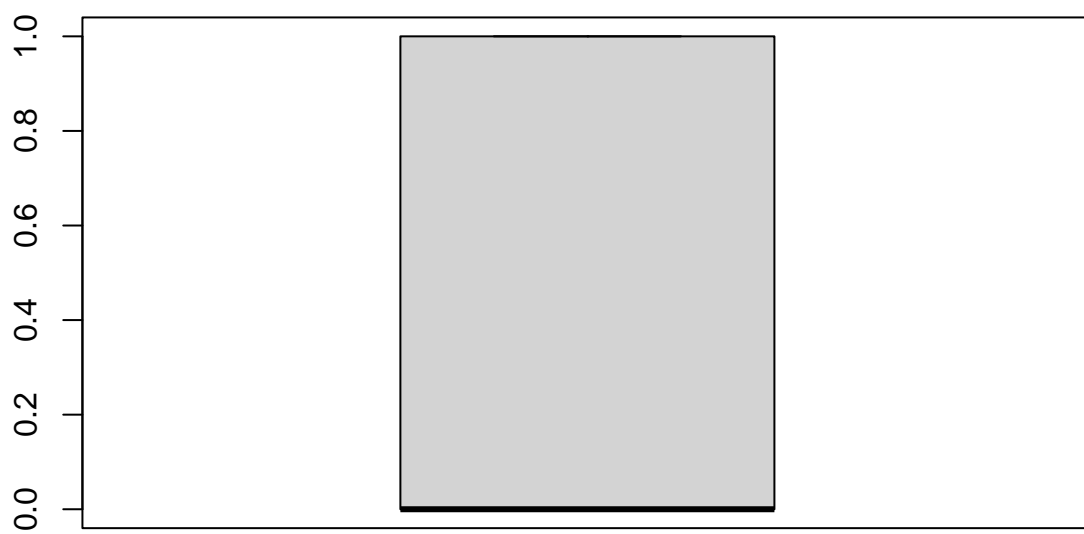


**Histogram zmiennej 9**

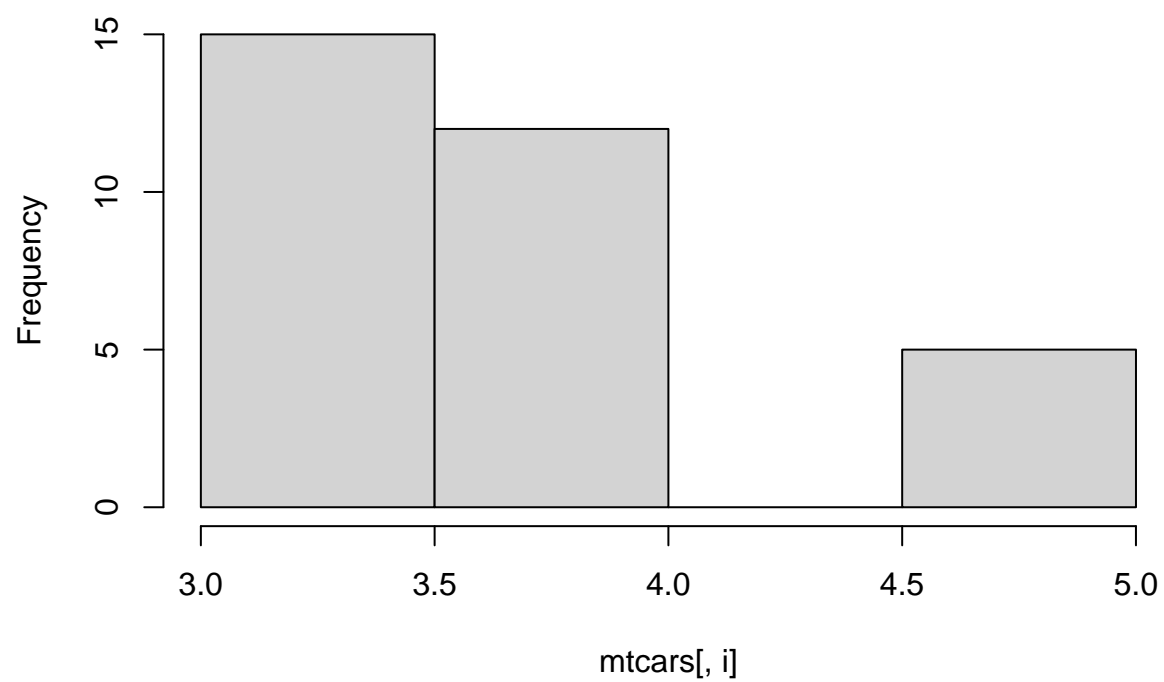




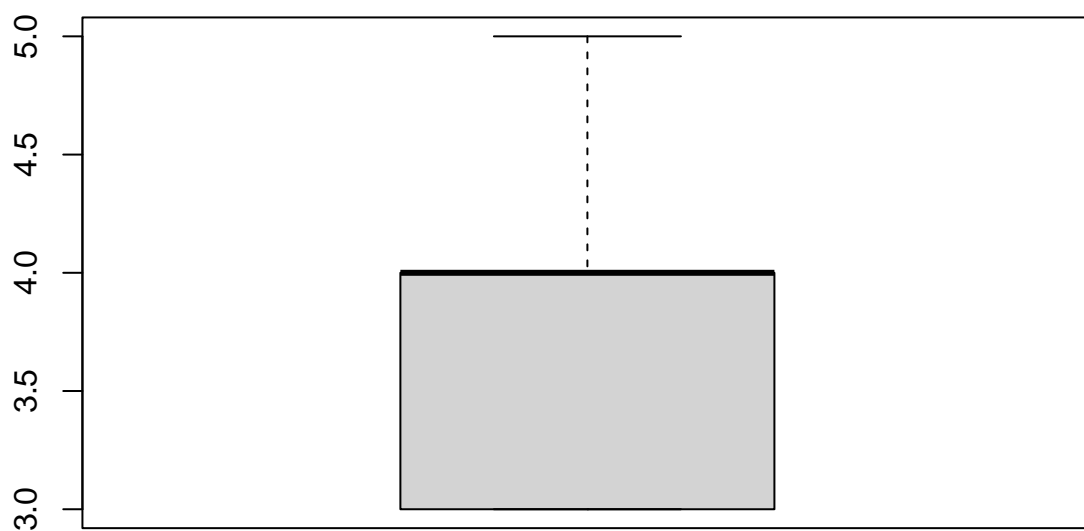
**Wykres zmiennej 9**



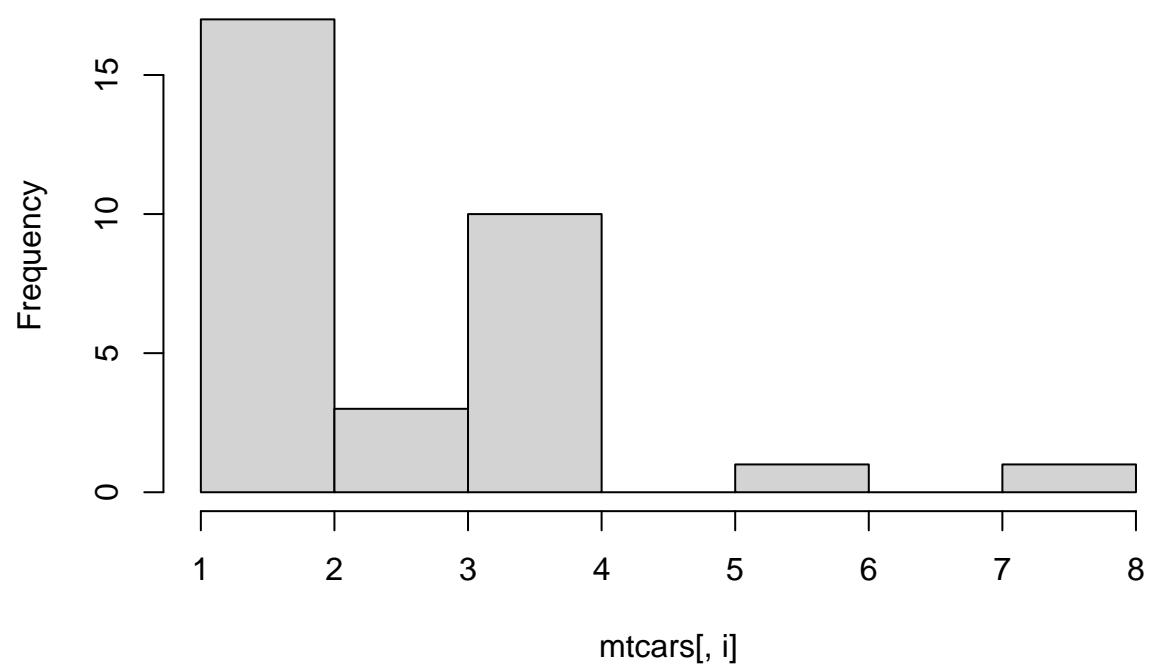
**Histogram zmiennej 10**



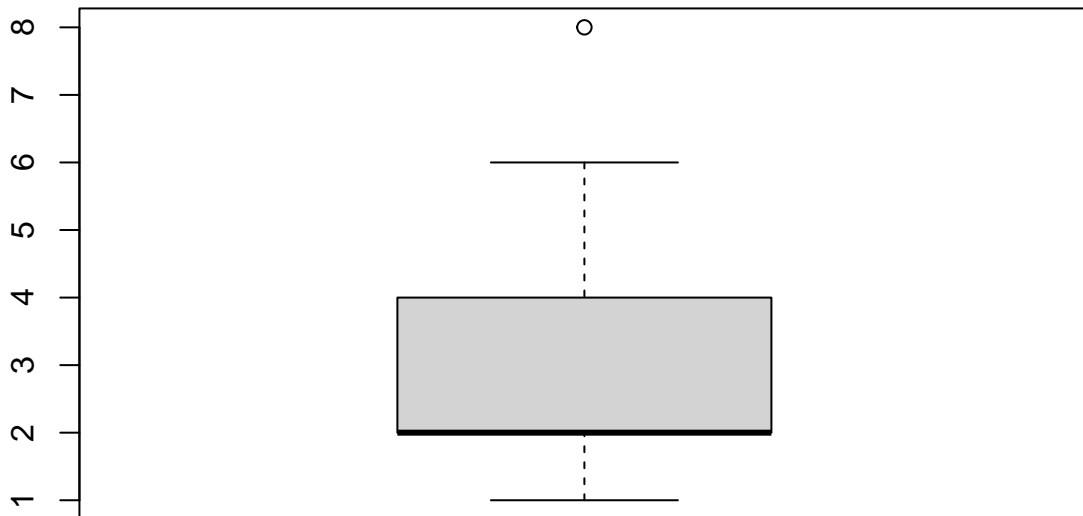
**Wykres zmiennej 10**



**Histogram zmiennej 11**



## Wykres zmiennej 11



#zadanie4

*# a) sprawdź ?sapply a następnie stwórz wektor v <- c(1,2,3,4) i oblicz  
# pierwiastek dla każdego elementu wektora*

```
?sapply;
```

```
## uruchamianie serwera httpd dla pomocy ... wykonano
```

```
v <- c(1,2,3,4);  
sapply(v, sqrt);
```

```
## [1] 1.000000 1.414214 1.732051 2.000000
```

*# b) Zdefiniuj macierz składającą się z 10 wierszy i 3 kolumn o wartościach  
# od 1 do 30 ułożone kolumnami. Następnie zsumuj elementy w każdym wierszu.*

```
m1 <- matrix(1:30, nrow=10, ncol=3, byrow=FALSE);  
m1;
```

```
##      [,1] [,2] [,3]  
## [1,]    1    11   21  
## [2,]    2    12   22  
## [3,]    3    13   23  
## [4,]    4    14   24  
## [5,]    5    15   25
```

```
## [6,]    6   16   26
## [7,]    7   17   27
## [8,]    8   18   28
## [9,]    9   19   29
## [10,]   10   20   30
```

```
apply(m1, 1, sum)
```

```
## [1] 33 36 39 42 45 48 51 54 57 60
```

```
# c) Utwórz listę:
```

```
# $item1
```

```
# 1 2 3 4 5
```

```
# $item2
```

```
# 4 12 20 28 36
```

```
# $item3
```

```
# 1 3 5 7 9
```

```
lista <- list(item1=c(1, 2, 3, 4, 5), item2=c(4, 12, 20, 28, 36), item3=c(1, 3, 5, 7, 9));
lista;
```

```
## $item1
```

```
## [1] 1 2 3 4 5
```

```
##
```

```
## $item2
```

```
## [1] 4 12 20 28 36
```

```
##
```

```
## $item3
```

```
## [1] 1 3 5 7 9
```

```
# Następnie oblicz wartość średnią elementów dla poszczególnych obiektów.
```

```
lapply(lista, mean);
```

```
## $item1
```

```
## [1] 3
```

```
##
```

```
## $item2
```

```
## [1] 20
```

```
##
```

```
## $item3
```

```
## [1] 5
```

```
# d) Dla każdej ze zmiennych ilościowych z mtcars oblicz statystyki opisowe  
# i zapisz jako elementy listy.
```

```
lista_statystyk <- lapply(mtcars, summary);
```

```
lista_statystyk;
```

```
## $mpg
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
```

```
##   10.40   15.43   19.20   20.09   22.80   33.90
```

```
##
```

```
## $cyl
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   4.000   4.000   6.000   6.188   8.000   8.000
##
## $disp
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   71.1   120.8   196.3   230.7   326.0   472.0
##
## $hp
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   52.0    96.5   123.0   146.7   180.0   335.0
##
## $drat
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.760   3.080   3.695   3.597   3.920   4.930
##
## $wt
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.513   2.581   3.325   3.217   3.610   5.424
##
## $qsec
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   14.50   16.89   17.71   17.85   18.90   22.90
##
## $vs
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.0000   0.0000   0.0000   0.4375   1.0000   1.0000
##
## $am
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.0000   0.0000   0.0000   0.4062   1.0000   1.0000
##
## $gear
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.000   3.000   4.000   3.688   4.000   5.000
##
## $carb
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.000   2.000   2.812   4.000   8.000
```

#zadanie5

```
# Utwórz pętlę while sprawdzającą prędkość auta i wyświetlającą informacje
# o prędkości. Jeśli prędkość będzie wyższa niż 80 lub niższa niż 30, wtedy
# pętla powinna się zakończyć. Jeśli prędkość jest wyższa niż 48 to powinien
# pojawić się komunikat „zwolnij mocno”, a od prędkości odejmij 11, w innych
# przypadkach powinien pojawić się komunikat zwolnij, a prędkość jest redukowana
# o 6. Prędkość początkowa wynosi 64.
```

```
predkosc <- 64;
while(predkosc<=80 & predkosc>30){
  cat("Twoja predkosc to ", predkosc,"\n");
  if(predkosc>48){
    cat("zwolnij mocno\n");
```

```
    predkosc <- predkosc-11;
  }else {
    cat("zwolnij\n");
    predkosc <- predkosc-6;
  }
}
```

```
## Twoja predkosc to 64
## zwolnij mocno
## Twoja predkosc to 53
## zwolnij mocno
## Twoja predkosc to 42
## zwolnij
## Twoja predkosc to 36
## zwolnij
```