

Podstawowe typy - zadania

1.1

A. liczba dodatnich elementów

```
sum(x > 0)
```

$x > 0$ zwraca wektor logiczny, więc sum zwróci liczbę wystąpień wartości TRUE (bo wektor logiczny przed sumowaniem zostanie niejawnie przekonwertowany na wektor liczbowy, FALSE przejdzie na 0, a TRUE na 1)

B. podzielne przez 3

```
sum(x[round(x) %% 3 == 0])
```

C. odległość od 8

```
abs(x-8)
```

D. normalizacja: <https://www.wikiwand.com/pl/Interpolacja liniowa>

```
y0 <- -1; y1 <- 1
```

```
x0 <- min(x); x1 <- max(x)
```

```
y0 + ((y1-y0)/(x1-x0))*(x-x0)
```

E. średnia wartość kwadratów liczb >5 lub <2

```
mean( (x[x > 5 | x < 2])^2 )
```

F. parzysta/nieparzysta

```
c("parzysta","nieparzysta")[x %% 2 +1]
```

#druga opcja z tworzeniem i uzupełnianiem wektora:

```
y <- character(length(x))
```

```
y[x%%2==0] <- "parzysta"
```

```
y[x%%2==1] <- "nieparzysta"
```

G. średnia

```
sum(x)/length(x)
```

H. wariancja

```
sum((x-mean(x))^2)/(length(x)-1)
```

I. min maks

```
sort(x)[c(1,length(x))]
```

1.2 kalkulator

```
#musimy dodać 1 do x aby przesunąć liczby, pierwszym  
# elementem wektora jest 0 a nie 1  
cat("\n",top[x+1],"\n",mid[x+1],"\n",bot[x+1])
```

1.3 unikatowe

```
#przeczytaj w dokumentacji jak działa funkcja rle  
rle(sort(x))$values
```

1.4 korelacja

```
#ważna jest kolejność potęgowania i sumowania w mianowniku!  
sum((x-mean(x))*(y-mean(y))) /  
(sqrt(sum((x-mean(x))^2)) * sqrt(sum((y-mean(y))^2)))
```

1.5

```
# jeśli różnica dwóch kolejnych elementów jest równa 0 to oznacza,  
# że te elementy są sobie równe  
which(diff(x) == 0)
```

1.6

```
# do wektora dodajemy ten sam wektor,ale odwrócony (dzięki funkcji  
# rev) a następnie ograniczamy go do połowy elementów  
(x + rev(x))[1:(length(x)/2)]  
# Nawias podczas indeksowania jest istotny. Jeśli nie wiesz dlaczego  
# porównaj działanie poniższych operacji i zastanów się nad kolejnością  
# działań  
1:10/2  
1:(10/2)
```

1.7

```
# do wektora x na miejsca gdzie jest NA wpisujemy średnia z wektora x  
# policzoną BEZ wartości NA - dzięki ustawieniu parametru  
# na.rm na TRUE  
x[is.na(x)] <- mean(x, na.rm=TRUE)
```

1.8

#mnożymy pi przez kolejne potęgi 10 (1, 10, 100, 1000 itd),
#pozbywamy się części ułamkowej (przez funkcję floor) i na koniec
#dzielimy modulo przez 10 aby z liczby kilkucyfrowej wydobyć jednostki
`floor(pi*10^(0:10))%%10`

1.9 Leibniz

#dzięki regule zawijania możemy mnożyć dwuelementowy wektor
c(1,-1) z dowolnie długim (w tym wypadku 1000 elementów)
`4*sum(c(1, -1) / (2* (0:999) + 1))`

1.10

#losujemy punkty (x,y) - pierwszy element z wektora x i pierwszy z y
to jeden punkt, x[2] i y[2] to drugi itd
`x <- runif(1000,-1,1); y <- runif(1000,-1,1)`
`mean(sqrt(x^2+y^2) < 1)*4` #sprawdzamy które punkty są w kole
`[1] 3.144`

1.11

#tworzymy trzelementową listę
`list(x[x<0], x[x==0], x[x>0])`