

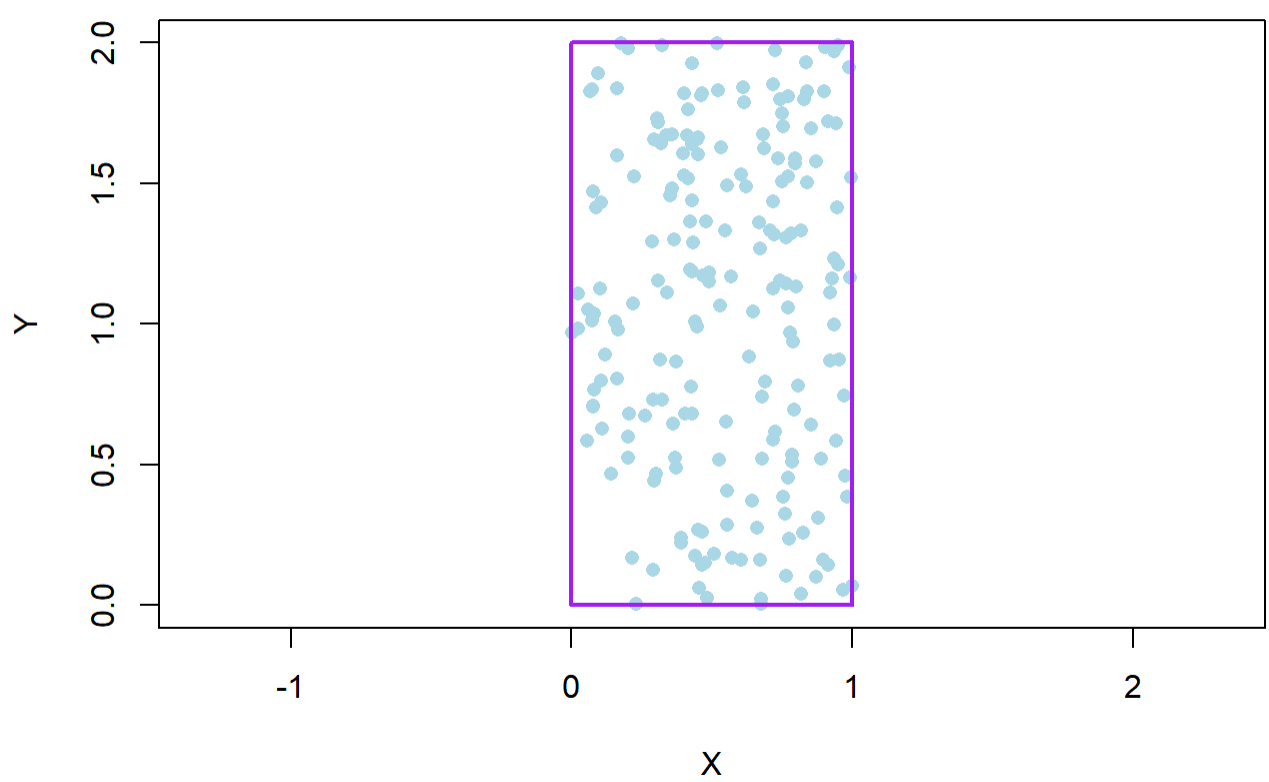
Cwiczenia_1

```
# Zadanie 1: Generowanie punktów w prostokącie
# Wygenerowanie 200 losowych współrzędnych x w zakresie [0, 1]
points_x <- runif(200, min = 0, max = 1)

# Wygenerowanie 200 losowych współrzędnych y w zakresie [0, 2]
points_y <- runif(200, min = 0, max = 2)

# Zadanie 2: Rysowanie punktów i prostokąta
# Wykres punktów z proporcjami osi 1:1
plot(points_x, points_y, xlab = "X", ylab = "Y", pch = 16, col = "lightblue", asp = 1,
      xlim = c(0, 1), ylim = c(0, 2))

# Rysowanie prostokąta o wymiarach [0,1] x [0,2]
rect(0, 0, 1, 2, border = "purple", lwd = 2)
```



```
# Zadanie 3: Generowanie punktów w okręgu
# Wygenerowanie 300 losowych wartości promienia r^2 w zakresie [0, 100]
r_squared <- runif(300, min = 0, max = 100)

# Wygenerowanie 300 losowych kątów w zakresie [0, 2π]
angles <- runif(300, min = 0, max = 2 * pi)

# Obliczenie współrzędnych x punktów na podstawie r i kąta
circle_x <- sqrt(r_squared) * cos(angles)

# Obliczenie współrzędnych y punktów na podstawie r i kąta
circle_y <- sqrt(r_squared) * sin(angles)

# Zadanie 4: Rysowanie okręgu z punktami
# Ładowanie biblioteki do analizy procesów punktowych
library(spatstat)

## Warning: pakiet 'spatstat' został zbudowany w wersji R 4.4.1

## Ładowanie wymaganego pakietu: spatstat.data

## Warning: pakiet 'spatstat.data' został zbudowany w wersji R 4.4.2

## Ładowanie wymaganego pakietu: spatstat.univar

## Warning: pakiet 'spatstat.univar' został zbudowany w wersji R 4.4.2

## spatstat.univar 3.1-1

## Ładowanie wymaganego pakietu: spatstat.geom

## Warning: pakiet 'spatstat.geom' został zbudowany w wersji R 4.4.1

## spatstat.geom 3.3-3

## Ładowanie wymaganego pakietu: spatstat.random

## Warning: pakiet 'spatstat.random' został zbudowany w wersji R 4.4.1

## spatstat.random 3.3-2

## Ładowanie wymaganego pakietu: spatstat.explore

## Warning: pakiet 'spatstat.explore' został zbudowany w wersji R 4.4.1

## Ładowanie wymaganego pakietu: nlme

## spatstat.explore 3.3-2

## Ładowanie wymaganego pakietu: spatstat.model

## Warning: pakiet 'spatstat.model' został zbudowany w wersji R 4.4.1

## Ładowanie wymaganego pakietu: rpart

## spatstat.model 3.3-2

## Ładowanie wymaganego pakietu: spatstat.linnet

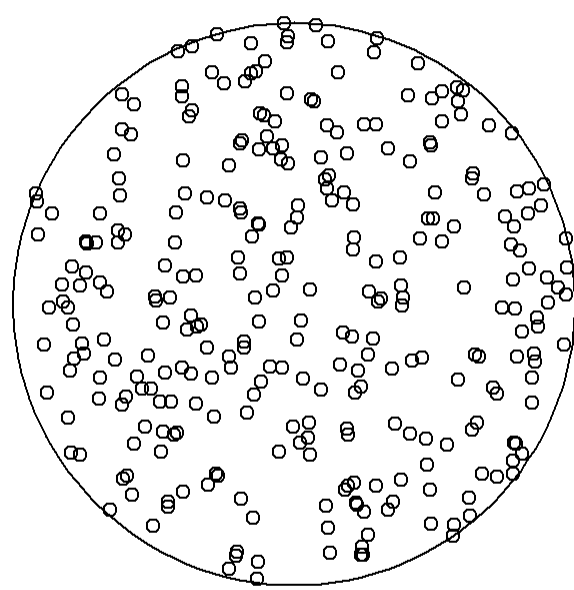
## Warning: pakiet 'spatstat.linnet' został zbudowany w wersji R 4.4.1

## spatstat.linnet 3.2-2

##
## spatstat 3.2-1
## For an introduction to spatstat, type 'beginner'

# Rysowanie okręgu o promieniu 10
plot(disc(radius = 10, centre = c(0, 0)), main = "")

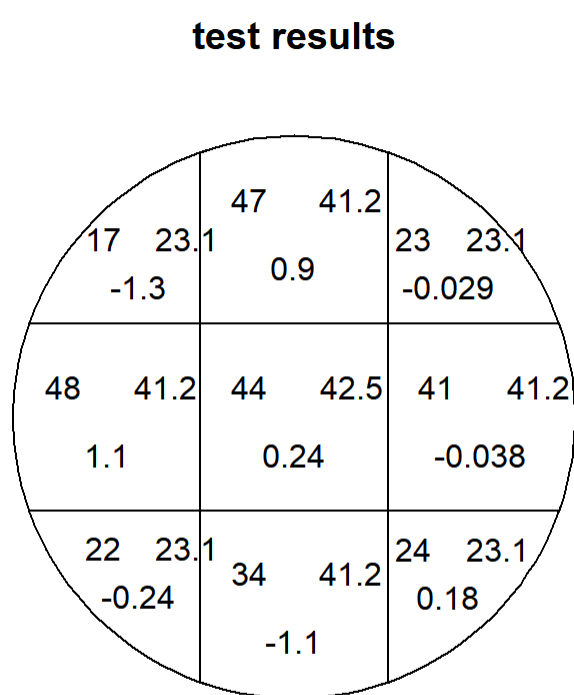
# Nanoszenie wygenerowanych punktów na wykres
points(circle_x, circle_y)
```



```
# Zadanie 5: Test losowości punktów
# Utworzenie planarnego wzorca punktowego z punktów w okręgu o promieniu 10
planar_point_pattern <- ppp(circle_x, circle_y, window = disc(radius = 10, centre = c(0, 0)))

# Wykonanie testu losowości na 3x3 podobozarach
test_result <- quadrat.test(planar_point_pattern, nx = 3, ny = 3)

# Wyświetlenie wyników testu na wykresie
plot(test_result, main = "test results")
```



```
# Wyświetlenie wartości p-value testu
cat("Wartość p-value:", test_result$p.value, "\n")

## Wartość p-value: 0.4757801
```

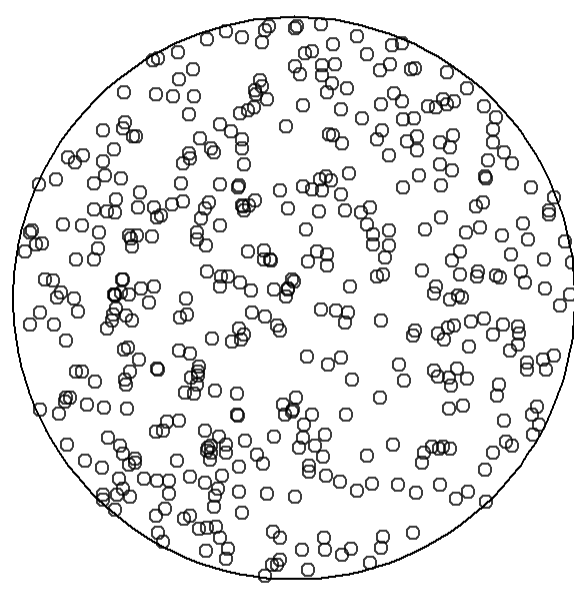
```
# Zadanie 6: Proces punktowy Poissona
# Obliczenie intensywności lambda na podstawie liczby punktów i powierzchni koła
lambda <- 500 / (pi * 10^2)

# Utworzenie okna w kształcie okręgu o promieniu 10
wzorzec <- disc(radius = 10, centre = c(0, 0))

# Wygenerowanie jednorodnego procesu Poissona w zadanym oknie
poisson_process <- rpoispp(lambda, win = wzorzec)

# Rysowanie procesu punktowego Poissona
plot(poisson_process)
```

poisson_process



```
# Liczenie liczby punktów w procesie
number_of_points <- npoints(poisson_process)

# Wyświetlenie liczby punktów
cat("Liczba punktów:", number_of_points, "\n")

## Liczba punktów: 469
```