Ćwiczenia 3

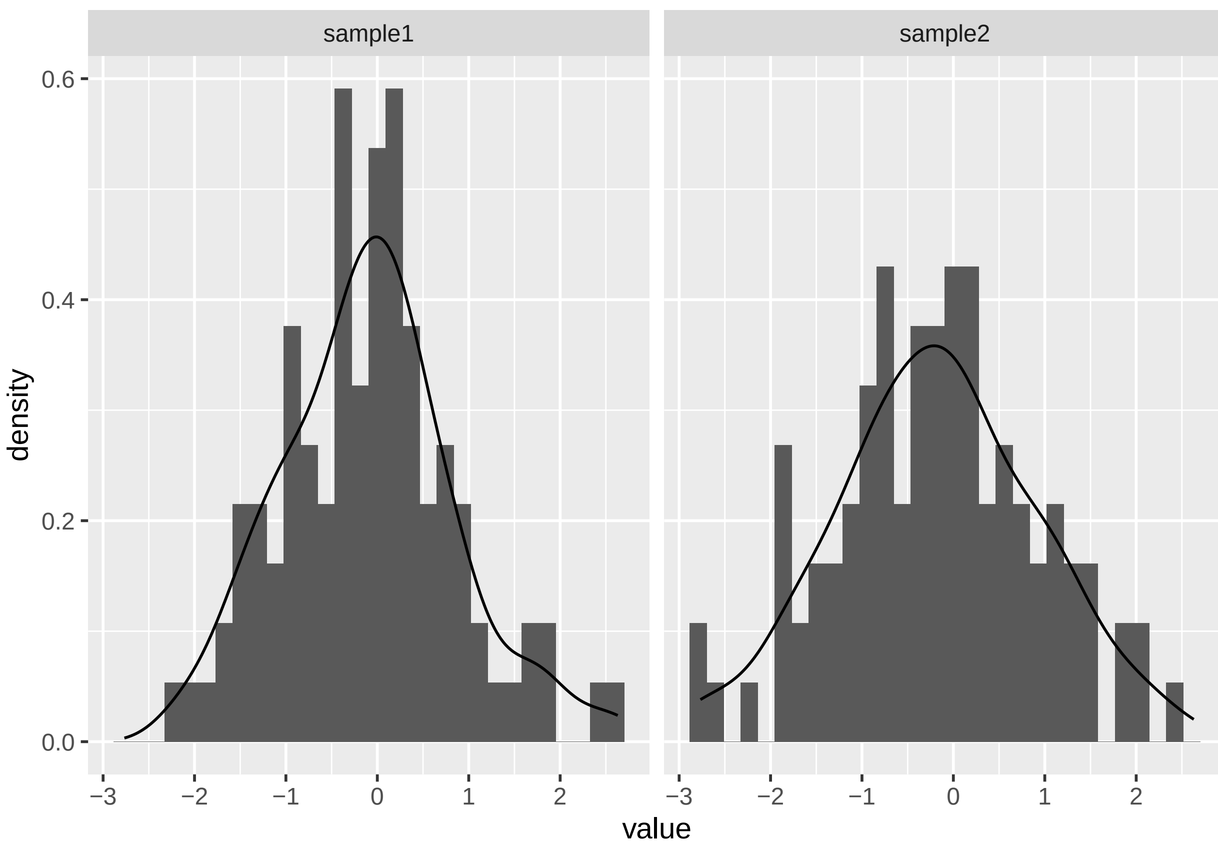
2023-10-23

Zadanie 1

```
df <- data.frame(sample1 = rnorm(100),
                 sample2 = rnorm(100))

df |>
  pivot_longer(cols = c("sample1", "sample2")) |>
  ggplot(aes(x = value)) +
  facet_wrap(~ name) +
  geom_histogram(aes(y = after_stat(density))) +
  geom_density()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Zadanie 2

```
mean(abs(rnorm(100, 100, 10) - 100) < 2 * 10)
```

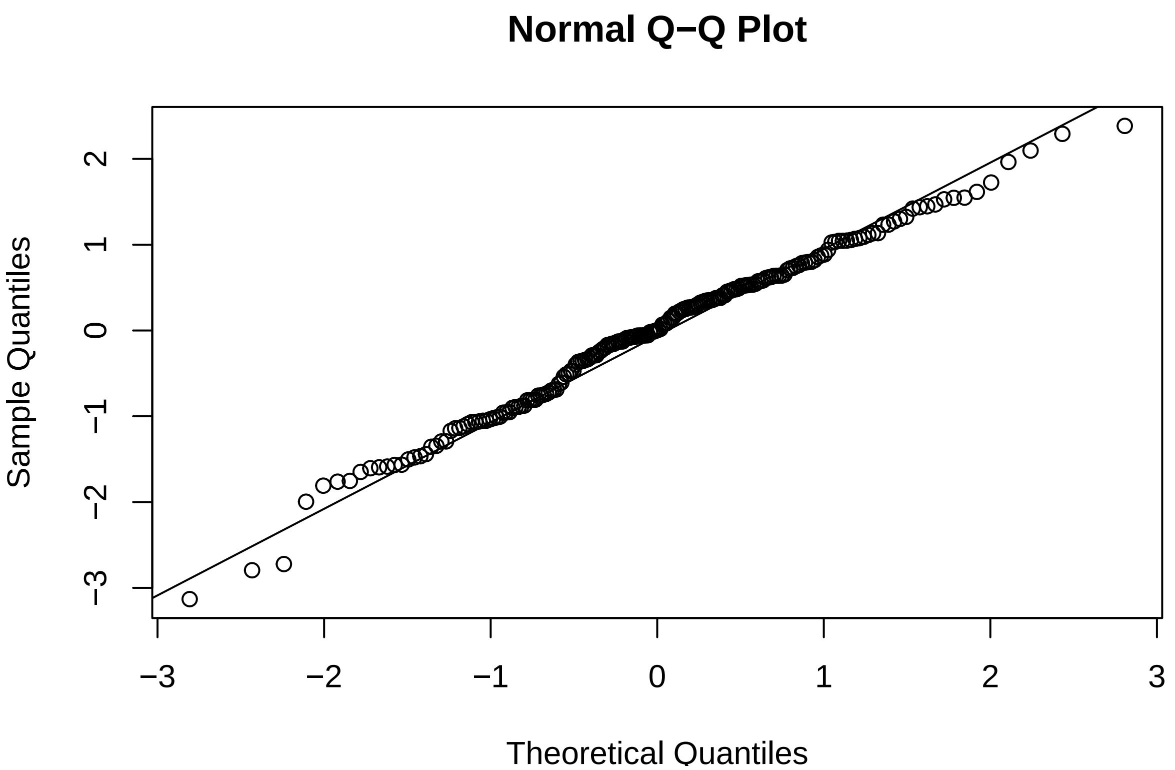
```
## [1] 0.96
```

Zadanie 3

Wykres normalności

```r
x <- rnorm(200)

qqnorm(x)
qqline(x)
```



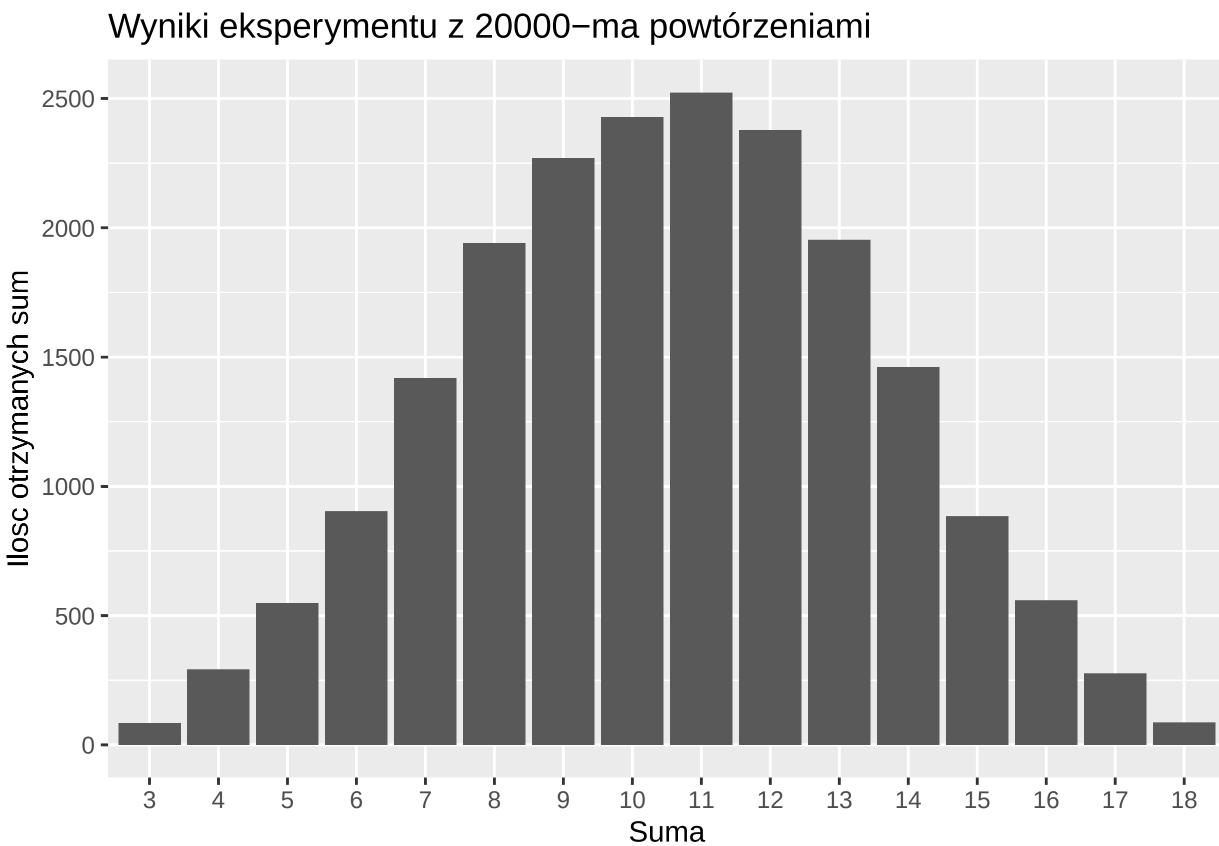

```r
data.frame(
  row.names = paste0(1:3, c(" - sigma")),
  oczekiwane = c(.68, .95, .998),
  zaobserwowane = sapply(1:3, function(t) mean(abs(x) < t))
)
```

```
##           oczekiwane zaobserwowane
## 1 - sigma      0.680         0.675
## 2 - sigma      0.950         0.970
## 3 - sigma      0.998         0.995
```

Zadanie 4

```r
fn <- function(throws = 1L) {
  data.frame(x = apply(as.matrix(1:3),
        function (t) sample(1:6, size = throws, replace = TRUE),
        MARGIN = 1) |>
    as.matrix() |>
    rowSums() |> as.factor()) |>
    ggplot(aes(x = x)) +
```

```r
    geom_bar() +
    ylab("Ilość otrzymanych sum") +
    xlab("Suma") +
    ggtitle(paste0("Wyniki eksperymentu z ", throws, ifelse(throws == 1L, "-nym",
                                                            "-ma"),
                   " powtórzeniami"))
}
fn(20000)
```



Zadania 5

Wartości teorytyczne $\mathbb{E}X = .99$ oraz $\mathrm{var}(X) = .99 \cdot .01 = 0.0099$

```r
fn <- function(n = 500L) {
  xx <- mean(rbinom(n = n, size = 1, prob = .99))
  data.frame("Średnia" = c(xx, .99),
             "Wariancja" = c(xx * (1 - xx), .0099),
             row.names = c("Zaobserwowane", "Teorytyczne"))
}
fn()
```

```
##               Średnia Wariancja
## Zaobserwowane   0.996  0.003984
## Teorytyczne     0.990  0.009900
```

Zadanie 6

Prawdopodobieństwo na podstawie: $\mathbb{P}\left(|X-\mu| \geq .8\right) = \mathbb{P}\left(\frac{|X-\mu|}{\sigma} \geq \frac{.8}{\sigma}\right) = \mathbb{P}(|\mathcal{N}_{0,1}| \geq \frac{.8}{\sigma}) = 1 - \Phi\left(\frac{.8}{\sigma}\right) + \Phi\left(-\frac{.8}{\sigma}\right) = 2\Phi\left(-\frac{.8}{\sigma}\right)$

```
2*pnorm(-.8 / .4)
```

```
## [1] 0.04550026
```

symulacja:

```
xx <- abs(rnorm(mean = 4.8, n = 100000, sd = .4) - 4.8) > .8

prop.test(sum(xx), n = 100000, p = 2*pnorm(-.8 / .4))
```

```
##
##  1-sample proportions test with continuity correction
##
## data:  sum(xx) out of 1e+05, null probability 2 * pnorm(-0.8/0.4)
## X-squared = 0.025513, df = 1, p-value = 0.8731
## alternative hypothesis: true p is not equal to 0.04550026
## 95 percent confidence interval:
##  0.04411228 0.04670278
## sample estimates:
##       p
## 0.04539
```

Oczekujemy:

```
50*2*pnorm(-.8 / .4)
```

```
## [1] 2.275013
```

Odrzucanie:

```
50 * 2 * (1 - pnorm(abs(c("4" = 4, "6" = 6) - 4.8) / .4)) < .5
```

```
##     4     6
## FALSE  TRUE
```

Zadanie 7

Według testu grubbsa 34 jest outlierem a według testu dixona nie ma podstaw do odrzucenia hipotezy zerowej o tym, że 34 nie jest outlierem

```
wyniki <- c(12, 34, 22, 14, 22, 17, 24, 22, 18, 14, 18, 12)

print(dixon.test(wyniki))
```

```
##
##  Dixon test for outliers
##
## data:  wyniki
## Q = 0.54545, p-value = 0.1007
## alternative hypothesis: highest value 34 is an outlier
```

```
print(grubbs.test(wyniki))
```

```
##
##  Grubbs test for one outlier
```

```
##
## data:  wyniki
## G = 2.3833, U = 0.4367, p-value = 0.0295
## alternative hypothesis: highest value 34 is an outlier
```

Kyterium Chauveneta też sugeruje odrzucenie:

```r
print(length(wyniki) * 2 * (1 - pnorm(abs(max(wyniki) - mean(wyniki)) / sd(wyniki))))
```

```
## [1] 0.2059207
```

```r
length(wyniki) * 2 * (1 - pnorm(abs(max(wyniki) - mean(wyniki)) / sd(wyniki))) < .5
```

```
## [1] TRUE
```

Zadanie 8

```r
forest <- predict(isolation.forest(trees, ntrees = 1000), trees)
sort(forest, decreasing = TRUE)[1:5]
```

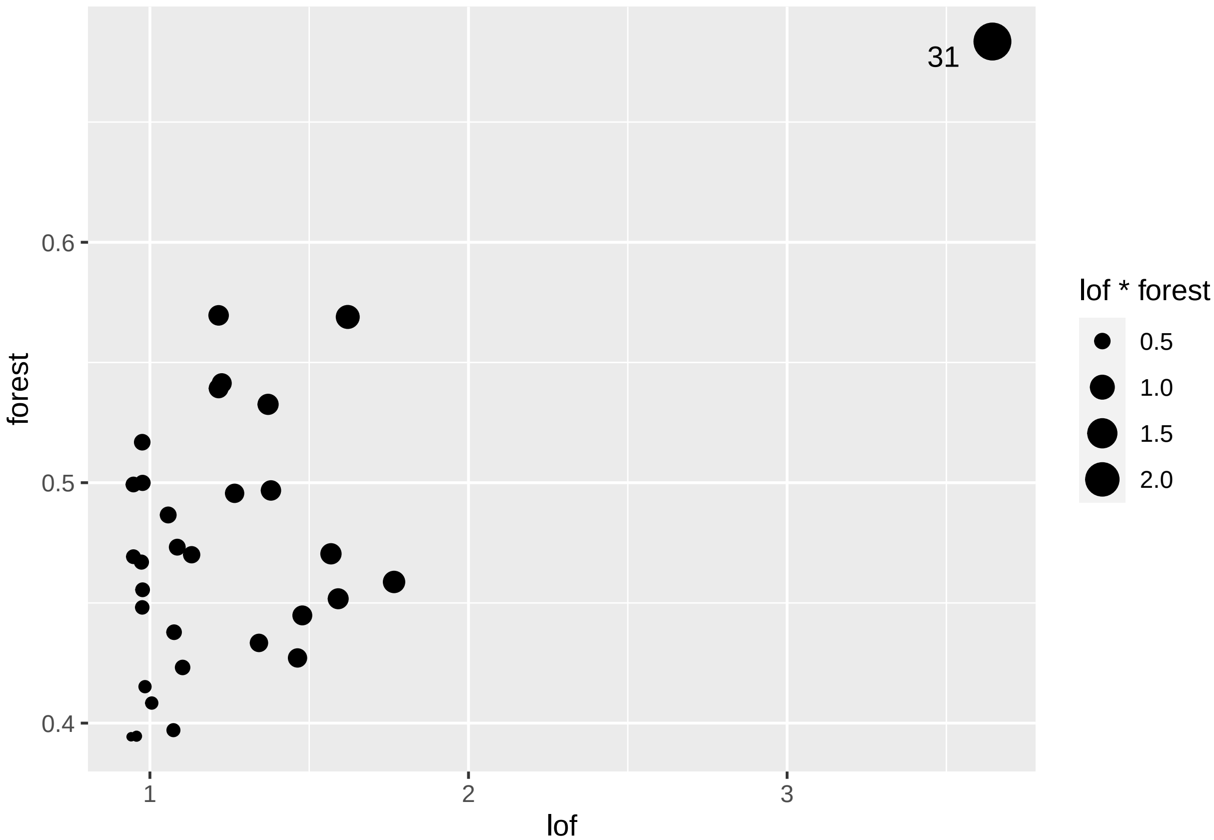
```
##        31         3        20         1         2
## 0.6834886 0.5695981 0.5689373 0.5413775 0.5392385
```

```r
xx <- lof(trees)
names(xx) <- 1:31
sort(xx, decreasing = TRUE)
```

```
##        31        19        20        14         5         4        11         6
## 3.6445516 1.7663248 1.6210132 1.5909972 1.5683008 1.4784099 1.4633124 1.3798939
##        18         9         7         1         2         3        25        16
## 1.3709020 1.3422896 1.2658520 1.2254345 1.2156568 1.2156568 1.1309854 1.1026153
##        26        21        10        27         8        15        23        24
## 1.0858870 1.0758357 1.0737739 1.0573879 1.0056643 0.9845838 0.9770081 0.9770081
##        17        22        29        13        28        30        12
## 0.9759299 0.9759299 0.9733871 0.9580263 0.9482168 0.9482168 0.9410482
```

```r
tibble(forest = forest, lof = xx) |>
  ggplot(aes(x = lof, y = forest)) +
  geom_point(aes(size = lof * forest)) +
  geom_text(vjust = 1.2, hjust = 2, aes(label = 31),
            data = tibble(forest = forest, lof = xx) |>
              top_n(1))
```

```
## Selecting by lof
```



Zadanie 9

```
df <- trees
df[df$Height == 80, "Height"] <- NA
tail(df)
```

```
##    Girth Height Volume
## 26  17.3     81   55.4
## 27  17.5     82   55.7
## 28  17.9     NA   58.3
## 29  18.0     NA   51.5
## 30  18.0     NA   51.0
## 31  20.6     87   77.0
```

`lm` wypada słabo bo jest mało danych

```
model <- lm(Height ~ ., data = df |>
  filter(!is.na(Height)))

df1 <- cbind(
  "imputacja" = predict(model, df |> filter(is.na(Height))),
  df |> filter(is.na(Height))
)
```

```
model <- mice(df)
```

```
##
##  iter imp variable
##   1   1  Height
```

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

```
df2 <- complete(model) |>
  subset(is.na(df$Height))
```

```
c("mice" = mean((df2$Height - 80) ^ 2),
  "lm"   = mean((df1[,1] - 80) ^ 2))
```

```
##     mice       lm
## 25.20000 12.74768
```