Raport - lista nr 2

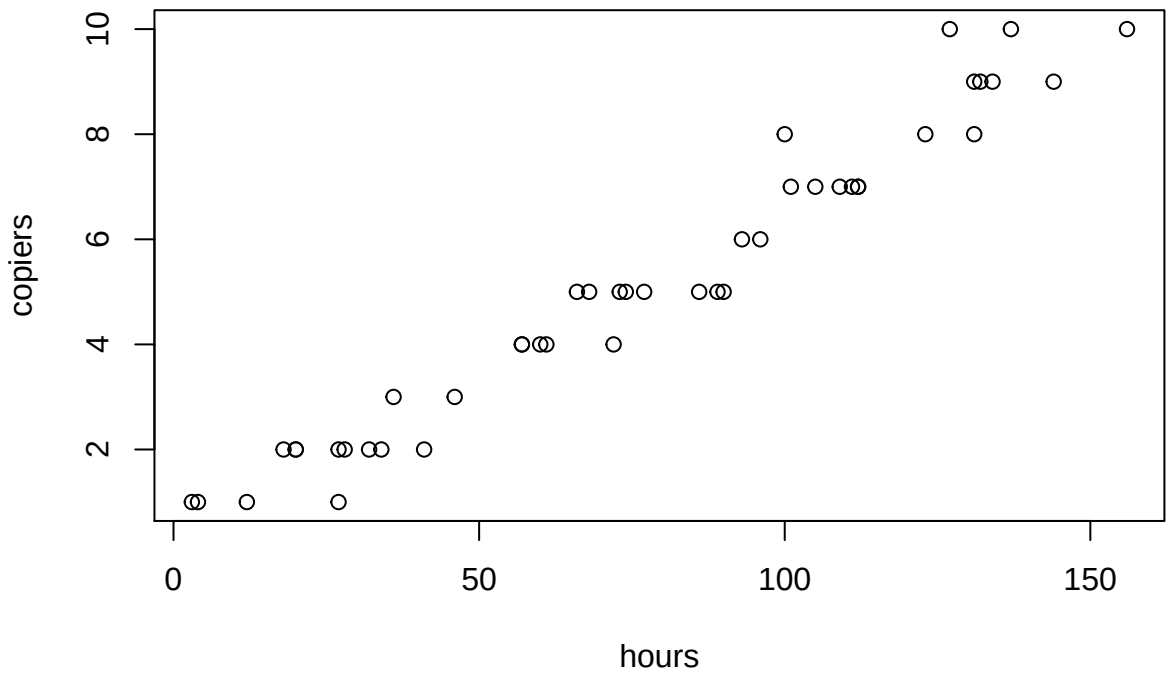
Erwin Jasic

17 listopada 2020

```
dane_lab2 <- read.table('dane_lab2.txt')
data <- data.frame(dane_lab2)
colnames(data) <- c('hours', 'copiers')
```

Zadanie 1

```
plot(data)
```



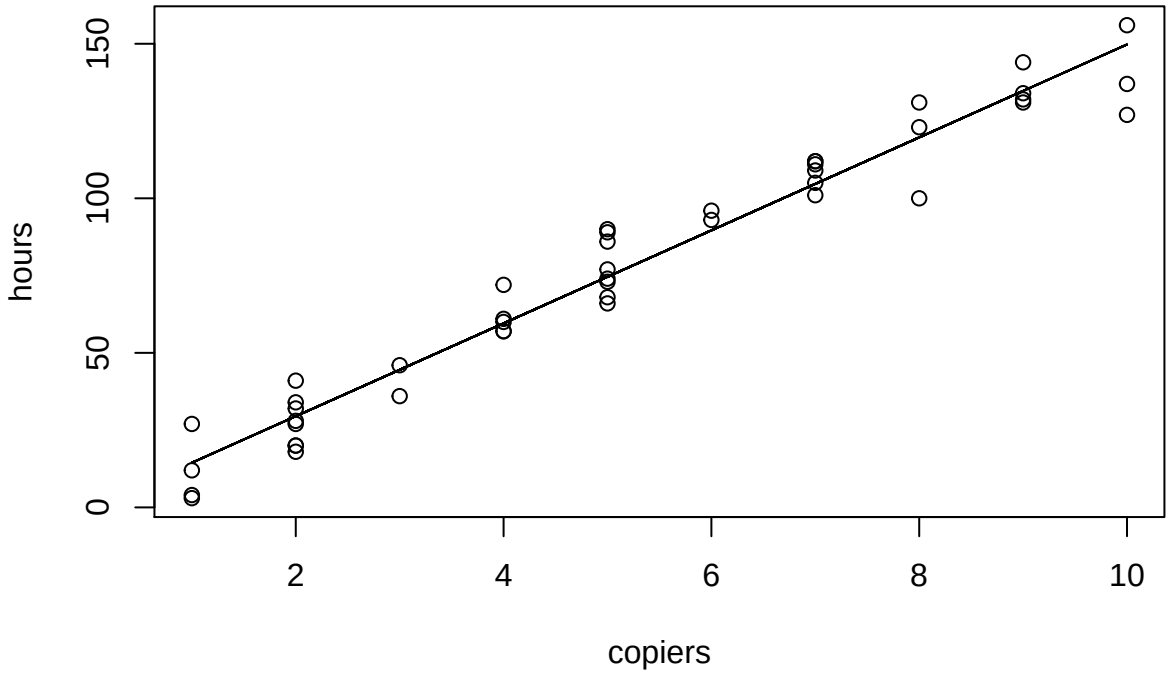
Widzimy zależność liniową, jaka zachodzi miedzy hours a copiers.

Zadanie 2

```
regres <- lm(hours~copiers, data)
regres
```

```
##
## Call:
## lm(formula = hours ~ copiers, data = data)
##
## Coefficients:
## (Intercept)      copiers
##     -0.5802      15.0352
```

```
pred <- predict.lm(regres)
plot(hours~copiers, data)
lines(pred~copiers, data)
```




```
confint(regres)
```

```
##                 2.5 %    97.5 %
## (Intercept) -6.234843  5.074529
## copiers     14.061010 16.009486
```

```
summary.lm(regres)
```

```
##
## Call:
## lm(formula = hours ~ copiers, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -22.7723  -3.7371   0.3334   6.3334  15.4039
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.5802     2.8039  -0.207    0.837
## copiers      15.0352     0.4831  31.123   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.914 on 43 degrees of freedom
## Multiple R-squared:  0.9575, Adjusted R-squared:  0.9565
## F-statistic: 968.7 on 1 and 43 DF,  p-value: < 2.2e-16
```

Aby estymować parametry modelu liniowego

$$Y = \beta_1 X + \beta_0 + \epsilon$$

wyznaczamy estymatory największej wiarogodności

$$b_1 = \frac{\sum_{i=1}^{n}(X_i - X)(Y_i - Y)}{\sum_{i=1}^{n}(X_i - X)^2}$$

$$b_0 = Y - b_1 X$$

Następnie wyznaczamy przedział ufności dla beta1 mianowicie:

$$b_1 \pm t_c s(b_1)$$

, gdzie

$$t_c = t(1 - \frac{\alpha}{2}, n - 2)$$

$$s^2(b_1) = \frac{s^2}{\sum_{i=1}^{n}(X_i - X)^2}$$

$$s^2 = \frac{\sum_{i=1}^{n}(Y_i - Y)^2}{n - 2}$$

Korzystając z funkcji w R przedział dla b1 wynosi [14.061010; 16.009486]
W ostatnim podpunkcie tego zadania wykonujemy test istotnośći dla beta1:
Hipoteza zerowa: beta1 = 0
Hipoteza alternatywna: beta1 jest różna od 0.
Statystyka testowa

$$t = \frac{b_1 - 0}{s(b_1)}$$

Obszar odrzucenia hipotezy zerowej:

$$t_c \leq |t|$$

Widzimy, korzystając z funkcji summary.lm, że p-wartość wynosi praktycznie 0, zatem oznacza to, że na dowolnym poziomie istotności odrzucamy Hipotezę zerową. Zachodzi liniowa zależność.

Zadanie 3

```
predict(regres, data.frame(copiers = 11))
```

```
##        1
## 164.8076
```

```
predict(regres, data.frame(copiers = 11), interval = 'confidence')
```

```
##        fit      lwr      upr
## 1 164.8076 158.4754 171.1397
```

Wzór na estymacje średniego czasu pracy:

$$E(y_h) = m_h = \beta_0 + \beta_1 X_h$$

$$M_h = b_0 + b_1 X_h$$

Estymacja przedziałowa:

$$M_h \pm t_c s(M_h)$$

, gdzie

$$s^2(M_h) = s^2(\frac{1}{n} + \frac{(X_h - X)^2}{\sum_{i=1}^{n}(X_i - X)^2})$$

Zadanie 4

```
predict(regres, data.frame(copiers = 11), interval = 'prediction')
```

```
##        fit      lwr     upr
## 1 164.8076 145.7491 183.866
```

Predykcja modelu regresji liniowej (u nas regres) opisuje wzór:

$$y_h = \beta_0 + \beta_1 x_h + \epsilon$$

, gdzie epsilon pochodzi z rozkładu normalnego ze średnią 0 oraz z wariancja sigma do kwadratu. Przedział ufności dla predykcji:

$$Y_h \pm t_c \sigma(Y_h)$$

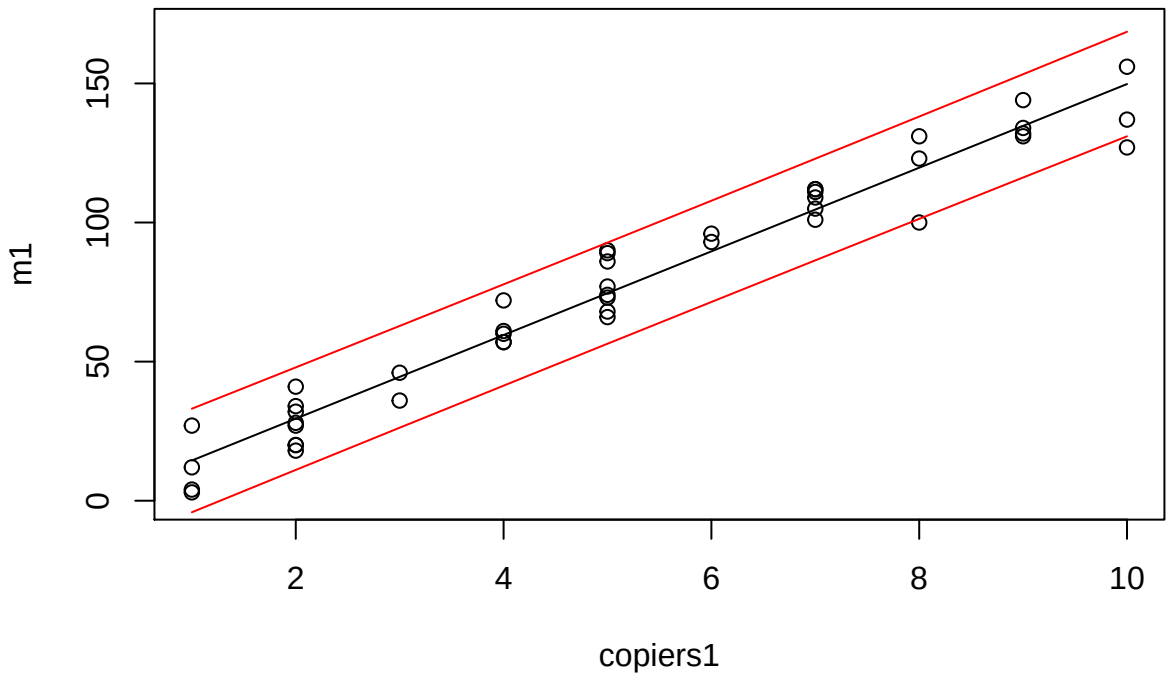
$$\sigma^2(Y_h) = \sigma^2(y_h - M_h)$$

Zadanie 5

```
pred2 <- predict(regres, interval = 'prediction')
```

```
## Warning in predict.lm(regres, interval = "prediction"): predictions on current data refer to _future_
```

```
pred2down <- sort(pred2[, 2])
pred2up <- sort(pred2[, 3])

v <- predict(regres, se.fit=TRUE)
m <- v$fit
copiers1 <- sort(data$copiers)
hours1 <- sort(data$hours)
m1 <- sort(m)
plot(copiers1, m1, type = 'l', ylim = c(0,170))
points(data$copiers,data$hours)
points(copiers1,pred2up,type='l',col='red')
points(copiers1,pred2down,type='l',col='red')
```



Widzimy, że na wykresie jedynie 2 obserwacje leżą poza czerwonymi liniami. To jest zgodne z naszą intuicją, ponieważ w naszym przedziale powinno znajdować się około 95% wszystkich obserwacji. Wszystkich obserwacji mamy 45 zatem 2/45 to około 4.4%, więc wszystko się zgadza.

Zadanie 6

```
n <- 40
sigma2 <- 120
SSX <- 1000
alpha <- 0.05
df <- n - 2
tc <- qt(1 - alpha/2, df)

sigma2beta1 <- sigma2/SSX
beta1 <- 1

delta <- beta1/sqrt(sigma2beta1)

prob1 <- function(delta) {
  pt(tc, df, delta)
}
prob2 <- function(delta) {
  pt(-tc, df, delta)
}

power <- 1 - prob1(delta) + prob2(delta)
```

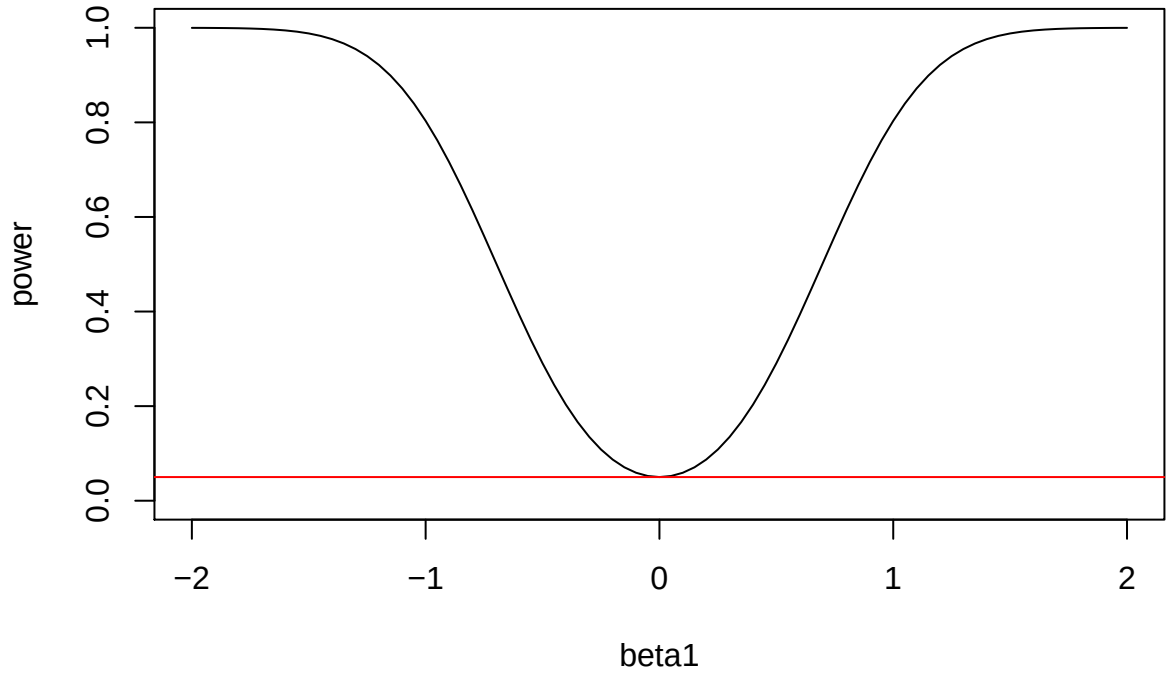
```
power
```

```
## [1] 0.8032105
```

```
beta1 <- seq(from = -2.0, to = 2.0, by = 0.05)
delta <- beta1/sqrt(sigma2beta1)
prob1 <- function(delta) {
  pt(tc, df, delta)
}
prob2 <- function(delta) {
  pt(-tc, df, delta)
}
power <- 1 - prob1(delta) + prob2(delta)
```

```
## Warning in pt(tc, df, delta): pełna precyzja może nie zostać osiągnięta w
## 'pnt{final}'

## Warning in pt(tc, df, delta): pełna precyzja może nie zostać osiągnięta w
## 'pnt{final}'

## Warning in pt(tc, df, delta): pełna precyzja może nie zostać osiągnięta w
## 'pnt{final}'

## Warning in pt(tc, df, delta): pełna precyzja może nie zostać osiągnięta w
## 'pnt{final}'

## Warning in pt(tc, df, delta): pełna precyzja może nie zostać osiągnięta w
## 'pnt{final}'

## Warning in pt(tc, df, delta): pełna precyzja może nie zostać osiągnięta w
## 'pnt{final}'

## Warning in pt(tc, df, delta): pełna precyzja może nie zostać osiągnięta w
## 'pnt{final}'

## Warning in pt(tc, df, delta): pełna precyzja może nie zostać osiągnięta w
## 'pnt{final}'

## Warning in pt(tc, df, delta): pełna precyzja może nie zostać osiągnięta w
## 'pnt{final}'
```

```
plot(beta1, power, type = 'l', ylim = c(0, 1))
abline(h = 0.05, col = 'red')
```



Podobne zadanie było robione na wykładzie. Tutaj robimy podobne obliczenia, ale dla innych danych.

Zadanie 7

```
set.seed(1)
ex <- MASS::mvrnorm(1, rep(0, 200), Sigma = 1/200 * diag(200))

make_y_a <- function(x) {
  epsilon <- MASS::mvrnorm(mu = rep(0, 200), Sigma = diag(200))
  Y <- 5 + epsilon
  Y
}
ys_a <- lapply(1:1000, make_y_a)


make_y_b <- function(x) {
  epsilon <- stats::rexp(200, 1)
  Y <- 5 + epsilon
  Y
}
ys_b <- lapply(1:1000, make_y_b)


make_y_c <- function(x) {
  epsilon <- MASS::mvrnorm(mu = rep(0, 200), Sigma = diag(200))
  Y <- 5 + 1.5 * x + epsilon
```

```
  Y
}
ys_c <- lapply(1:1000, make_y_c)


make_y_d <- function(x) {
  epsilon <- stats::rexp(200, 1)
  Y <- 5 + 1.5 * x + epsilon
  Y
}
ys_d <- lapply(1:1000, make_y_d)

test_beta_1 <- function(Y, X) {
  hat_beta_1 <- (sum((X - mean(X)) * (Y - mean(Y))) / sum((X - mean(X)) ^ 2))
  alpha <- 0.05
  n <- length(X)
  tc <- qt(1 - alpha/2, n - 2)
  s_sqrt <- sum((Y - mean(Y)) ^ 2) / (n - 2)
  s_sqrt_hat_beta_1 <- s_sqrt / sum((X - mean(X)) ^ 2)
  t <- hat_beta_1 / sqrt(s_sqrt_hat_beta_1)
  return(abs(t) > tc)
}

beta_test_a <- lapply(X = ys_a, FUN = test_beta_1, ex)
sum(unlist(beta_test_a)) / 1000
```

```
## [1] 0.046
```

```
beta_test_b <- lapply(X = ys_b, FUN = test_beta_1, ex)
sum(unlist(beta_test_b)) / 1000
```

```
## [1] 0.042
```

```
beta_test_c <- lapply(X = ys_c, FUN = test_beta_1, ex)
sum(unlist(beta_test_c)) / 1000
```

```
## [1] 0.05
```

```
beta_test_d <- lapply(X = ys_d, FUN = test_beta_1, ex)
sum(unlist(beta_test_d)) / 1000
```

```
## [1] 0.043
```

```
power <- function(beta1) {
  n <- 1000
  sig2 <- 1
  SSX <- sum((ex - mean(ex)) ^ 2)
  df <- n - 2
  alpha <- 0.05
  sig2b1 <- sig2 / SSX
  tc <- qt(1 - alpha/2, df)
  delta <- beta1 / sqrt(sig2b1)
  prob1 <- function(delta) {
    pt(tc, df, delta)
  }
  prob2 <- function(delta) {
```

```
    pt(-tc, df, delta)
  }
  power <- 1 - prob1(delta) + prob2(delta)
  return(power)
}
power(1.5)
```

```
## [1] 0.2843568
```

W zadaniu 7 widzimy, że wyniki dla beta_test a-d są do siebie dość mocno zbliżone i są to wyniki na poziomie od 0.042 do 0.05.

Zadanie 8

```
n <- 20
b0 <- 1
b1 <- 3
s <- 4
alpha <- 0.05
sb1 <- 1
s2b1 <- sb1^2
tc <- qt(1 - alpha/2, n - 2)
tc
```

```
## [1] 2.100922
```

```
a_begin <- b1 - tc * sb1
a_end <- b1 + tc * sb1
a_result <- c(a_begin, a_end)
a_result
```

```
## [1] 0.899078 5.100922
```

```
c_begin <- 16 - tc * (1 + sb1)
c_end <- 16 + tc * (1 + sb1)
c_result <- c(c_begin, c_end)
c_result
```

```
## [1] 11.79816 20.20184
```

Z podpunktu a) widzimy, że nasz przedział nie zawiera 0 co oznacza, że odrzucamy hipotezę zerową czyli mamy zależność miedzy X a Y. W podpunkcie c) wyliczamy przedział predykcyjny. Tak jak się spodziewaliśmy, ten przedział jest większy od przedziału ufności.