Modele Regresji Raport 1

Paweł Matławski
album 249732

23 marca 2021

Spis treści

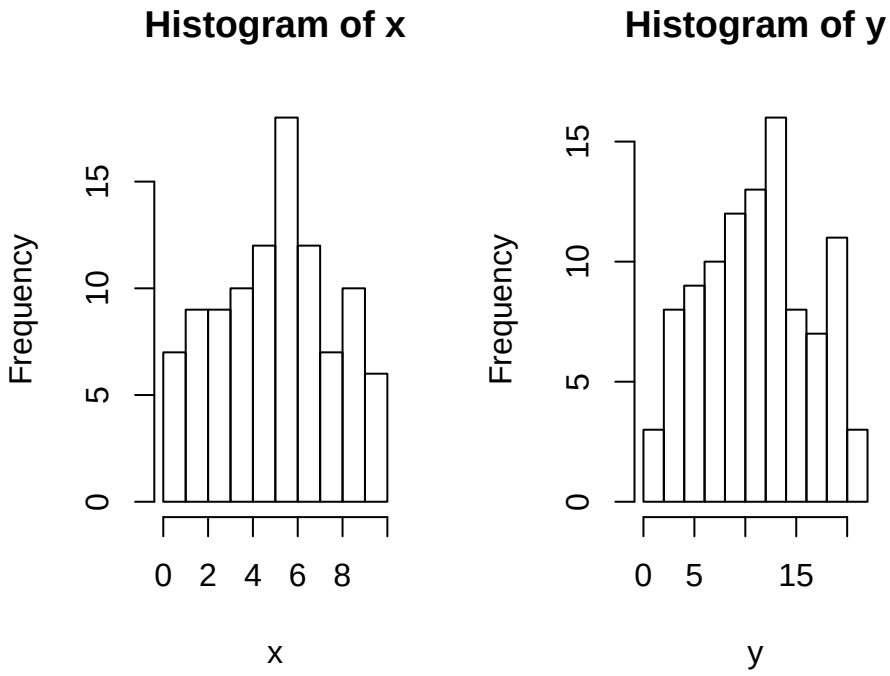
1 Lista nr 1

1.1 Zadanie nr 1

```
library(xtable)
df <- read.delim('lab1.txt', sep = "\t", dec = ",")
attach(df)

stats1 <- c(mean(x), var(x), sd(x), quantile(x, probs = 0.25), median(x), quantile(x, pr
            min(x), max(x))
stats2 <- c(mean(y), var(y), sd(y), quantile(y, probs = 0.25), median(y), quantile(y, pr
            min(y), max(y))
stats0 <- c(stats1, stats2)
stats0.matrix <- matrix(stats0, nrow = 2, ncol = length(stats1),
                        dimnames= list(c('x','y'),c("mean", "var", "sd", "1st quartile",
                                                    "median", "3rd quartile", "min", "max
                        , byrow = TRUE)
xtable(stats0.matrix)
```

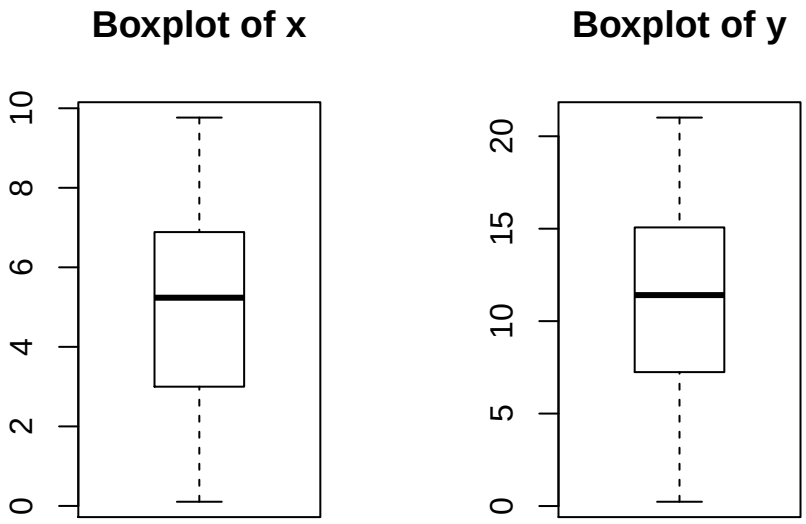
	mean	var	sd	1st quartile	median	3rd quartile	min	max
x	5.02	6.71	2.59	3.02	5.24	6.87	0.11	9.77
y	11.14	28.81	5.37	7.33	11.41	14.97	0.23	21.01

```r
par(mfrow = c(1,2))
hist(x)
hist(y)
```



Rysunek 1: Histograms

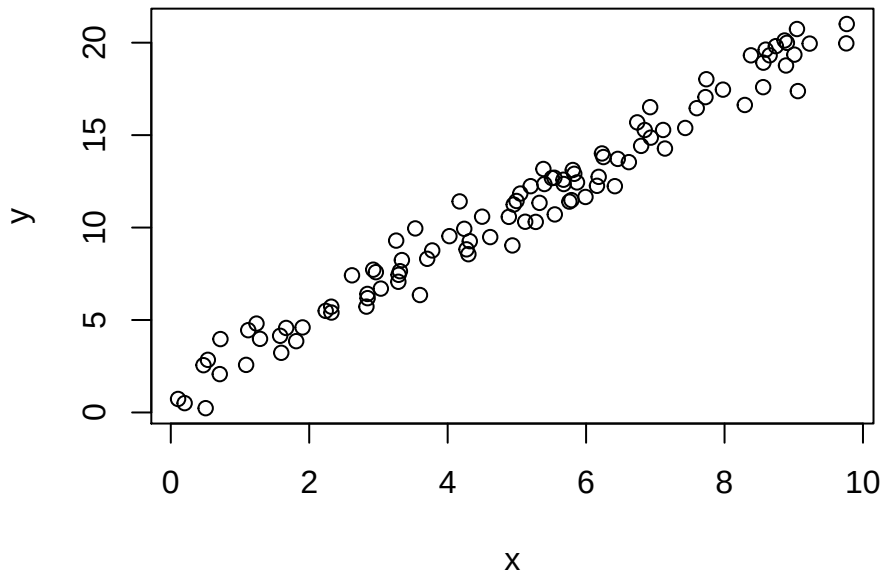
```r
par(mfrow = c(1,2))
boxplot(x, main = 'Boxplot of x')
boxplot(y, main = 'Boxplot of y')
```



Rysunek 2: Boxplots

1.2 Zadanie nr 2

```
plot(x, y)
```



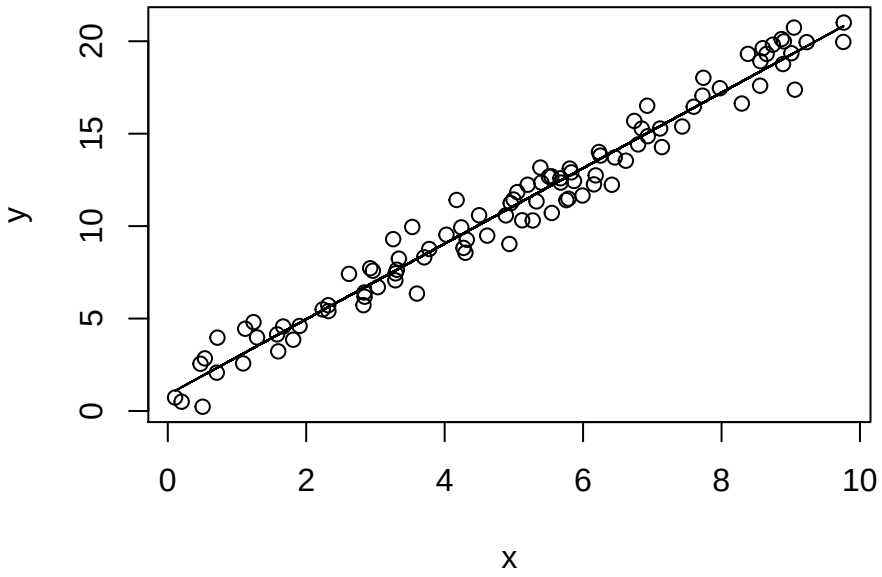

```
cor.test(x, y, method = 'pearson')

##
##  Pearson's product-moment correlation
##
## data:  x and y
## t = 57.125, df = 98, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9782133 0.9901124
## sample estimates:
##       cor
## 0.9853143
```

Z wykresu rozproszenia możemy wyczytać, że chmura punktów ma charakter liniowy. W związku z tym i z faktem, że współczynnik korelacji wynosi około 0.985 , możemy wykorzystać model regresji liniowej ($y = \beta_0 + \beta_1 * x + \epsilon$) do opisania zależności pomiędzy zmienną x i zmienną y.

1.3 Zadanie nr 3

Korzystając ze wzoru na $\hat{\beta}_1 = r\frac{s_y}{s_x}$ i $\hat{\beta}_0 = \bar{y} - \hat{\beta}_1\bar{x}$, gdzie r jest estymowanym współczynnikiem korelacji Pearsona, .

```
beta_1 = cor(x, y) * sd(y) / sd(x)
beta_1

## [1] 2.042517

beta_0 = mean(y) - (beta_1 * mean(x))
beta_0

## [1] 0.8798193

f <- function(x)
{
  return(beta_0 + beta_1 * x)
}
plot(x, y)
lines(x, f(x))
```



Wartości parametrów $\hat{\beta}_1$ i $\hat{\beta}_0$ wynoszą odpowiednio: 2.042517 i 0.8798193. Na powyższym wykresie dopasowałem estymowaną funkcję $y = \hat{\beta}_0 + \hat{\beta}_1 x$ regresji liniowej do wykresu rozrzutu.

1.4 Zadanie nr 4

Znajdźmy wartość estymatora $\hat{\sigma}^2 := \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n-2}$ parametru σ^2.

```
n = length(y)
sigma2 = sum((y - (beta_0 + beta_1 * x))^2)/(n-2)
sigma2

## [1] 0.8486302
```

1.5 Zadanie nr 5

Na poziomie istotności $\alpha = 0.05$ zweryfikujemy hipotezę $H_0 : \beta_0 = 0$, przy hipotezie alternatywnej $H_1 : \beta_1 \neq 0$.

```
alpha = 0.05
SE2 = sigma2 / sum((x - mean(x))^2)
T_stat = beta_1 / sqrt(SE2)
t_student = qt(p = 1-(0.05/2), df=n-2)
abs(T_stat)>=t_student

## [1] TRUE

Htest <- lm(y~x, data = df)
summary(Htest)

##
## Call:
## lm(formula = y ~ x, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.00391 -0.68670  0.05062  0.55173  2.01107
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.87982    0.20178    4.36 3.21e-05 ***
## x            2.04252    0.03576   57.12  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9212 on 98 degrees of freedom
## Multiple R-squared:  0.9708,Adjusted R-squared:  0.9705
## F-statistic:  3263 on 1 and 98 DF,  p-value: < 2.2e-16
```

Na podstawie oszacowania statystyki $|T| \geq t_{\alpha/2,n-2}$ (gdzie $T := \frac{\hat{\beta}_1}{SE_{\beta_1}}$) i tego, że p-value dla modelu liniowego dopasowanego do naszych danych wynosi $p \leq 2.2e - 16$ możemy odrzucić hipotezę zerową ($H_0 : \beta_0 = 0$) oraz uznać, że rozpatrywany w tym przykładzie model regresji liniowej ma sens.

1.6 Zadanie nr 6

Na poziomie 0.99 skonstruujemy przedziały ufności dla parametru β_1.

```
alpha2 = 0.99
SE = sqrt(SE2)
beta_1_L = beta_1 + (qt(p = alpha2/2, df = n-2) * SE)
beta_1_U = beta_1 - (qt(p = alpha2/2, df = n-2) * SE)
beta_1_L

## [1] 2.042068

beta_1_U

## [1] 2.042967
```

Przedział ufności dla parametru β_1 wygląda następująco: (2.042068, 2.042967).

1.7 Zadanie nr 7

Obliczymy prognozowaną przez model wartość $\hat{Y}(x_0)$ dla $x_0 = 1$, a następnie wyznaczymy przedziały ufności na poziomie 0.99 dla naszej prognozowanej wartości $\hat{Y}(x_0)$.

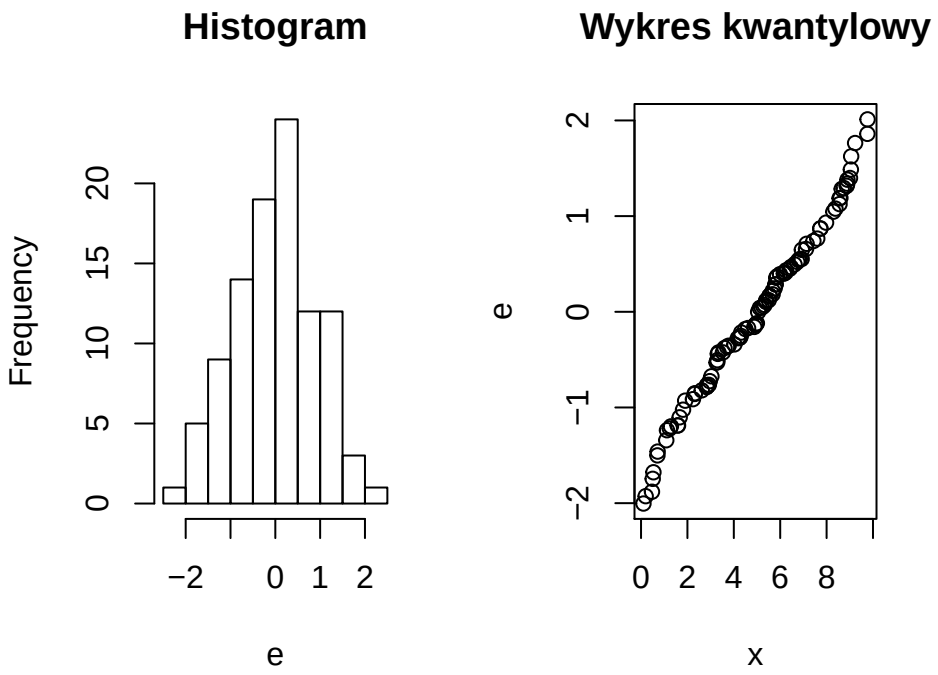
```
x_0 = 1
Y_0 = f(x_0)
Y_0

## [1] 2.922337

SE2_Y_0 = sigma2*(1+ (1/n) + ((x_0-mean(x))^2/sum(x-mean(x)^2)))
SE_Y_0 = sqrt(SE2_Y_0)
Y_0_L = Y_0 + qt(p = 0.99/2, df = n-2) * SE_Y_0
Y_0_U = Y_0 - qt(p = 0.99/2, df = n-2) * SE_Y_0
Y_0_L

## [1] 2.91075

Y_0_U

## [1] 2.933924
```

$\hat{Y}(1) = 2.922337$ i przedział ufności na poziomie 0.99: (2.91075, 2.933924).

1.8 Zadanie nr 8

Narysujmy histogram i wykres kwantylowy dla rezyduów naszego modelu.

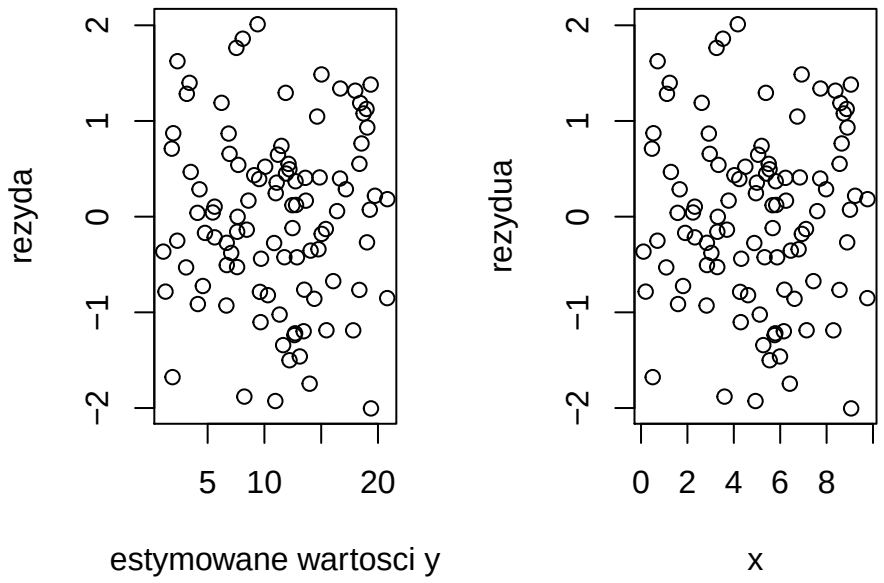
```
e = y - f(x)
par(mfrow = c(1,2))
hist(e, main = "Histogram")
qqplot(x, e, plot.it = TRUE, main = "Wykres kwantylowy")
```



Rysunek 3: Histogram i wykres kwantylowy dla rezyduów

Następnie narysujmy wykres rozproszenia dla $(\hat{yi}, e)$, gdzie i=1,2,...,n oraz dla (x_i, e), gdzie i=1,2,...,n.

```
par(mfrow = c(1,2))
plot(f(x), e, xlab = "estymowane wartości y", ylab = "rezyda")
plot(x, e, xlab = "x", ylab = "rezydua")
```



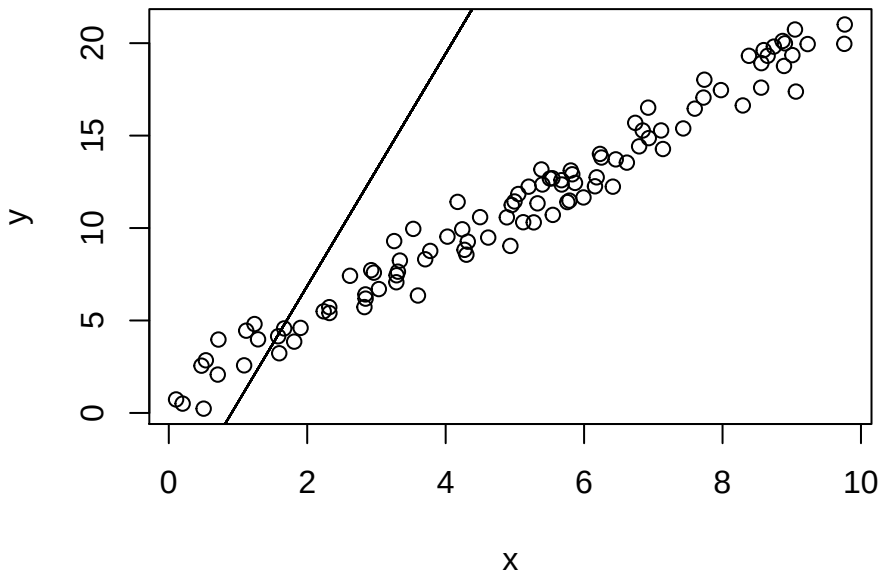
Rysunek 4: Wykresy rozproszenia

Na podstawie powyższych rysunków możemy stwierdzić, że nie ma korelacji ani pomiędzy estymowanymi wartościami y i rezyduami, ani pomiędzy wartościami x i rezyduami. W takim razie nic nie wskazuje, aby któreś z założeń modelu regresji nie było spełnione.

1.9 Zadanie nr 9

Zamieńmy ostatnią obserwację w naszych danych z (0.6546, 10.4989) na (0.6546, 1000.4989).

```
df_2 <- read.delim('lab1.txt', sep = "\t", dec = ",")
df_2$y[100] <- df_2$y[100]*100
summary(lm(df_2$y~df_2$x, data = df_2))

##
## Call:
## lm(formula = df_2$y ~ df_2$x, data = df_2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
##  -35.67  -22.36  -15.41   -5.92 1448.52
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -5.706     32.302  -0.177    0.860
## df_2$x         6.285      5.724   1.098    0.275
##
## Residual standard error: 147.5 on 98 degrees of freedom
## Multiple R-squared:  0.01215,Adjusted R-squared:  0.002073
## F-statistic: 1.206 on 1 and 98 DF,  p-value: 0.2749

beta_0_2 = -5.706
beta_1_2 = 6.285
plot(x, y)
lines(x, beta_0_2 + beta_1_2*x)
```



Parametry β_0 i β_1 dla zmodyfikowanego modelu wynoszą odpowiednio: $\beta_0 = -5.706$ i $\beta_1 = 6.285$. Na podstawie tego i wykresu przedstawiającego doposowanie modelu regresji liniowej możemy stwierdzić, że obserwacja jest odstająca i wpływowa.

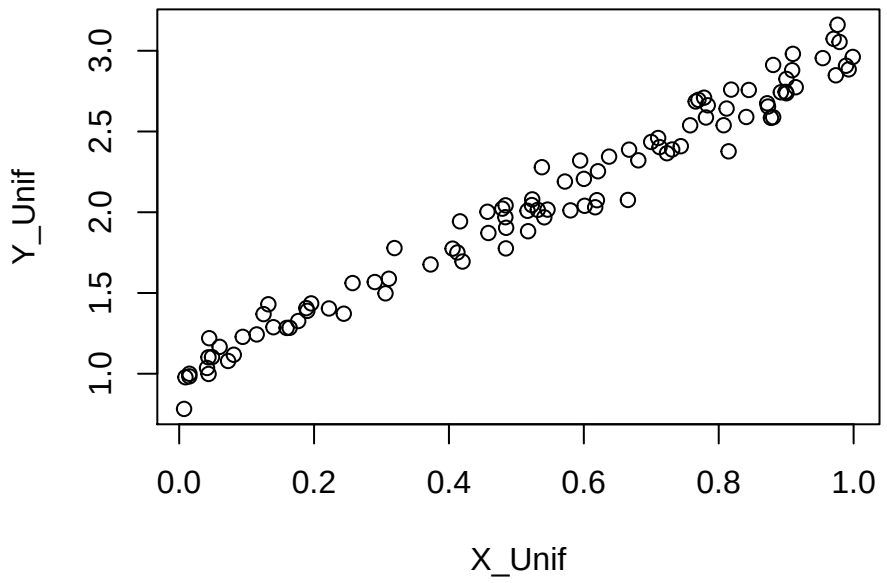
1.10 Zadanie nr 10

Generujemy 100 wartości x z rozkładu jednostajnego $U(0, 1)$, oraz wartości ϵ z rozkładu normalnego $N(0, \sigma^2)$. Teraz wygeneryjemy wartości y dla modelu liniowego o parametrach $\beta_0 = 1$, $\beta_1 = 2$ oraz $\sigma = 0.1$.

```
X_Unif <- runif(n, min = 0, max = 1)
beta_0_Unif = 1
beta_1_Unif = 2
SD = 0.1
E_Unif <- rnorm(n, mean = 0, sd = SD)
Y_Unif <- beta_0_Unif + (beta_1_Unif*X_Unif)+E_Unif
```

Następnie wykonajmy wykres rozproszenia, aby zwizualizować dopasowany model i wyznaczmy estymatory najmniejszych kwadratów $(\hat{\beta}_0, \hat{\beta}_1)$ parametrów (β_0, β_1).

```
plot(X_Unif, Y_Unif)
```

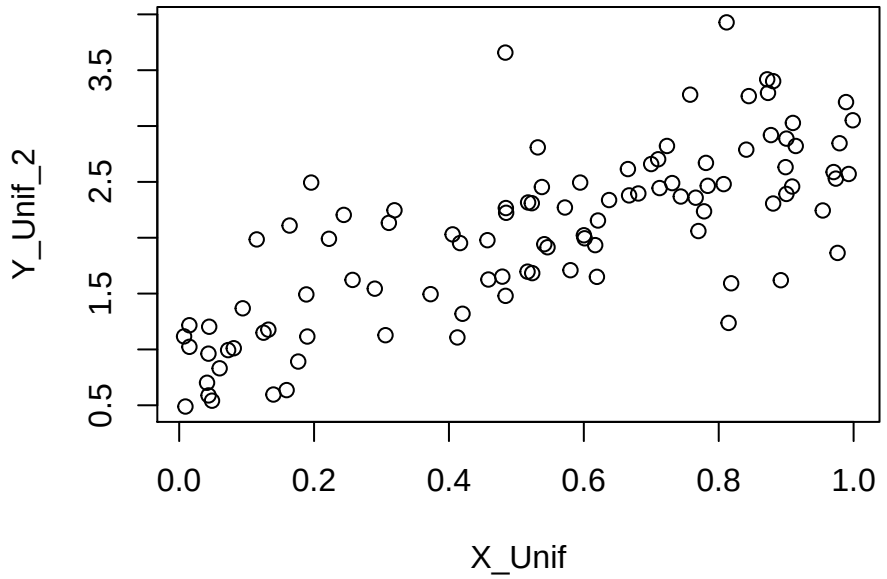


```r
summary(lm(Y_Unif~X_Unif))

##
## Call:
## lm(formula = Y_Unif ~ X_Unif)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.24368 -0.06271 -0.01100  0.06311  0.21636
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.98166    0.02053   47.81   <2e-16 ***
## X_Unif       2.01083    0.03325   60.48   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.102 on 98 degrees of freedom
## Multiple R-squared:  0.9739,Adjusted R-squared:  0.9736
## F-statistic:  3658 on 1 and 98 DF,  p-value: < 2.2e-16
```

Estymatory wynoszą $\hat{\beta}_0 = 0.98589$, $\hat{\beta}_1 = 2.01482$, czyli są bardzo zbliżone rzeczywistym wartościom, które wynoszą odpowiednio:$\beta_0 = 1, \beta_1 = 2$.

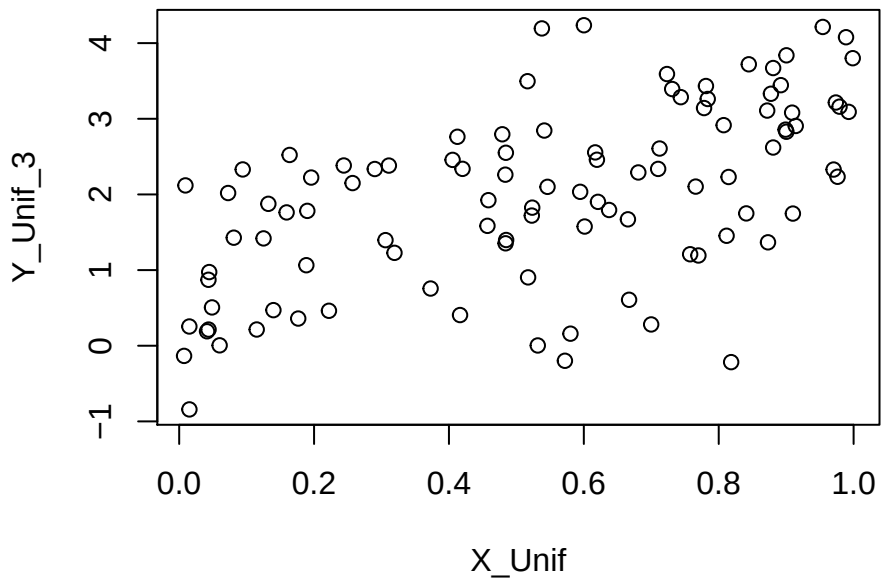
Powtórzmy teraz obliczenia dla $\sigma = 0.5$, a następnie dla $\sigma = 1$

```
SD_2 = 0.5
E_Unif_2 <- rnorm(n, mean = 0, sd = SD_2)
Y_Unif_2 <- beta_0_Unif + (beta_1_Unif*X_Unif)+E_Unif_2
plot(X_Unif, Y_Unif_2)
```




```
summary(lm(Y_Unif_2~X_Unif))

##
## Call:
## lm(formula = Y_Unif_2 ~ X_Unif)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.33394 -0.31064 -0.03545  0.28000  1.71306
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.0288     0.1006   10.23   <2e-16 ***
## X_Unif        1.8935     0.1628   11.63   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4994 on 98 degrees of freedom
## Multiple R-squared:  0.5798,Adjusted R-squared:  0.5756
## F-statistic: 135.2 on 1 and 98 DF,  p-value: < 2.2e-16
```

```r
SD_3 = 1
E_Unif_3 <- rnorm(n, mean = 0, sd = SD_3)
Y_Unif_3 <- beta_0_Unif + (beta_1_Unif*X_Unif)+E_Unif_3
plot(X_Unif, Y_Unif_3)
```




```r
summary(lm(Y_Unif_3~X_Unif))
```

```
##
## Call:
## lm(formula = Y_Unif_3 ~ X_Unif)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.83173 -0.62806  0.07151  0.77073  2.19694
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.8136     0.1955   4.162 6.78e-05 ***
## X_Unif        2.2011     0.3165   6.954 4.00e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9709 on 98 degrees of freedom
## Multiple R-squared:  0.3304,Adjusted R-squared:  0.3236
## F-statistic: 48.35 on 1 and 98 DF,  p-value: 4.005e-10
```

Precyzja estymatorów zdecydowanie maleje, co widać ewidetnie na wykresach rozproszenia. Współczynnik R^2 wzraz ze wzrostem parametru σ^2 zdecydowanie maleje.