```
Dobry chrześcijanin powinien wystrzegać się matematyków i tych wszystkich,
którzy tworzą puste proroctwa.
Istnieje niebezpieczeństwo, że matematycy zawarli przymierze z diabłem, aby
zgubić duszę człowieka i wtrącić go w odmęty piekieł. - św. Augustyn
```

Regresja ważona

Co, gdy nie ma stałej wariancji?

```
> strongx
   momentum energy crossx sd
1         4  0.345    367 17
2         6  0.287    311  9
3         8  0.251    295  9
4        10  0.225    268  7
5        12  0.207    253  7
6        15  0.186    239  6
7        20  0.161    220  6
8        30  0.132    213  6
9        75  0.084    193  5
10      150  0.060    192  5
```

crossx zależy liniowo od zmiennej energia (w jednostkach odwrotnych). Na każdym poziomie pędu (momentum) wielokrotnie zmierzono energię i crossx oraz obliczono jego odchylenie standardowe.

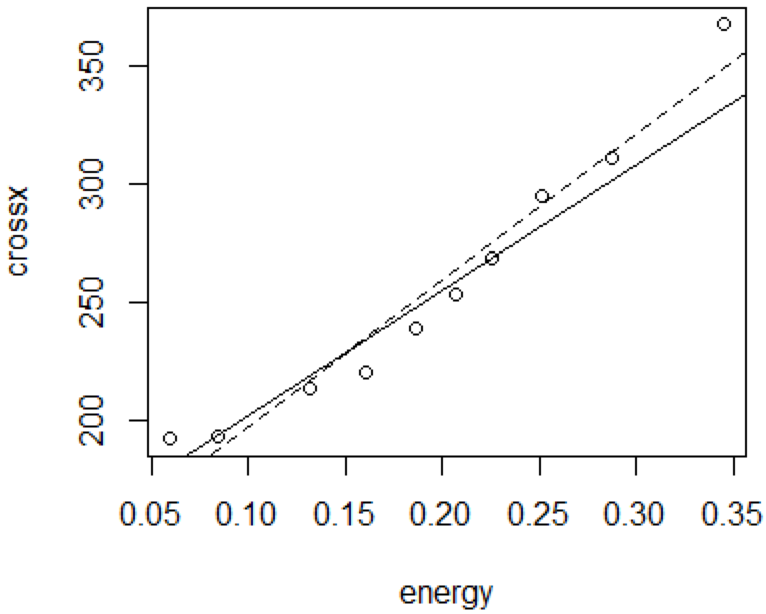
W takim przypadku stosuje się ważoną regresję liniową

```
> g <- lm(crossx ˜ energy, strongx, weights=sdˆ-2)
> summary(g)

Call:
lm(formula = crossx ~ energy, data = strongx, weights = sd^-2)

Weighted Residuals:
    Min      1Q  Median      3Q     Max
-2.3230 -0.8842  0.0000  1.3900  2.3353

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  148.473      8.079   18.38 7.91e-08 ***
energy       530.835     47.550   11.16 3.71e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.657 on 8 degrees of freedom
Multiple R-squared:  0.9397,    Adjusted R-squared:  0.9321
F-statistic: 124.6 on 1 and 8 DF,  p-value: 3.71e-06
```

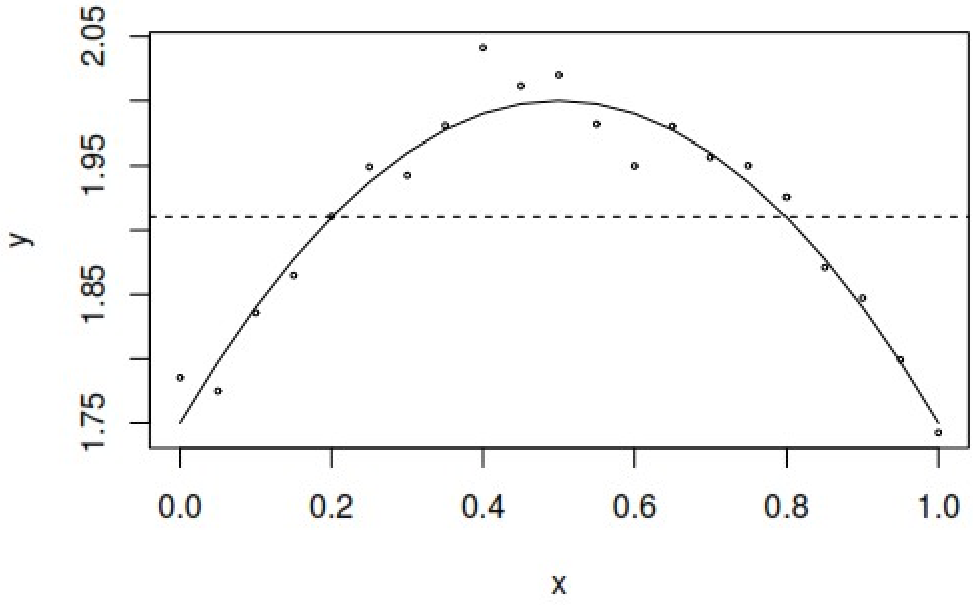
Tu prawdziwe $\sigma^2=1$ (dużo powtórzeń, więc wariancje są dobrze oszacowane)

Gdyby nie stosować regresji ważonej, to

```
> gu <- lm(crossx ~ energy, strongx)
> summary(gu)

Call:
lm(formula = crossx ~ energy, data = strongx)

Residuals:
    Min      1Q  Median      3Q     Max 
-14.773  -9.319  -2.829   5.571  19.817 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   135.00      10.08    13.4 9.21e-07 ***
energy        619.71      47.68    13.0 1.16e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

Residual standard error: 12.69 on 8 degrees of freedom
Multiple R-squared:  0.9548,    Adjusted R-squared:  0.9491 
F-statistic: 168.9 on 1 and 8 DF,  p-value: 1.165e-06


> plot(crossx ~ energy, data=strongx)
> abline(g)
> abline(gu,lty=2)
```



Testowanie dopasowania



Dobre dopasowanie jest wtedy, gdy błąd średniokwadratowy jest nieobciążonym estymatorem σ^2.

1. σ^2 jest znane

$$\frac{\hat{\sigma}^2}{\sigma^2} \sim \frac{\chi^2_{n-p}}{(n-p)}$$

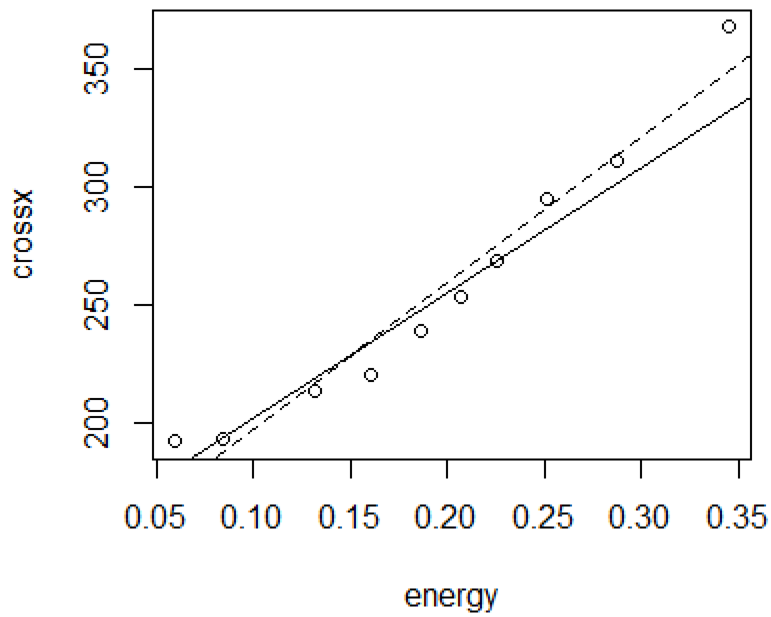
Testem dopasowania będzie

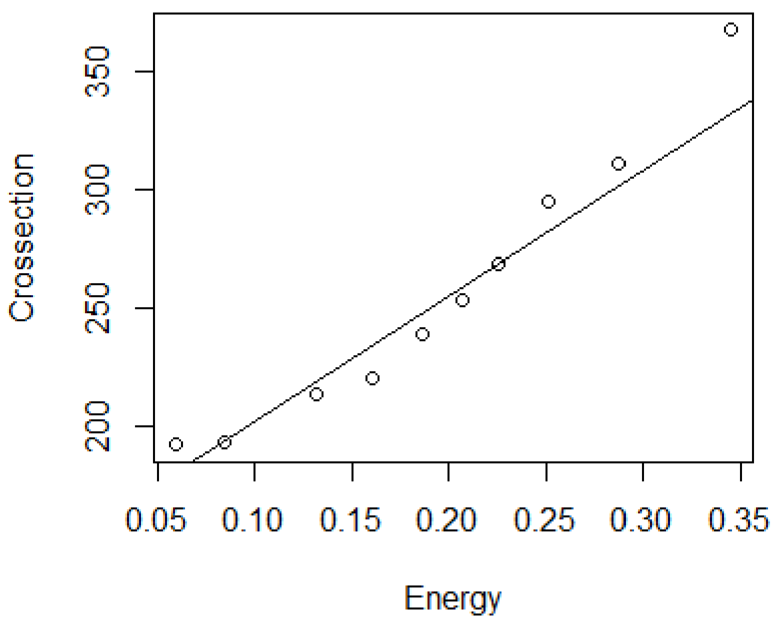
$$\frac{(n-p)\hat{\sigma}^2}{\sigma^2} > \chi^{2\;(1-\alpha)}_{n-p}$$

```
> g <- lm(crossx ˜ energy, strongx, weights=sdˆ-2)
> summary(g)

Call:
lm(formula = crossx ~ energy, data = strongx, weights = sd^-2)

Weighted Residuals:
    Min      1Q  Median      3Q     Max 
-2.3230 -0.8842  0.0000  1.3900  2.3353 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  148.473      8.079   18.38 7.91e-08 ***
energy       530.835     47.550   11.16 3.71e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1 

Residual standard error: 1.657 on 8 degrees of freedom
Multiple R-squared:  0.9397,    Adjusted R-squared:  0.9321 
F-statistic: 124.6 on 1 and 8 DF,  p-value: 3.71e-06
```

Tu prawdziwe σ^2=1

Czy to jest dobre dopasowanie?





> 1.66^2*8
[1] 22.045
> 1-pchisq(22.045,8)
[1] 0.0048332
Czyli jest złe dopasowanie!

> g2 <- lm(crossx ˜ energy + I(energyˆ2), weights=sdˆ-2, strongx)

```
> summary(g2)

Call:
lm(formula = crossx ~ energy + I(energy^2), data = strongx, weights = sd^-2)

Weighted Residuals:
     Min       1Q   Median       3Q      Max 
-0.89928 -0.43508  0.01374  0.37999  1.14238 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  183.8305     6.4591  28.461  1.7e-08 ***
energy         0.9709    85.3688   0.011 0.991243    
I(energy^2) 1597.5047   250.5869   6.375 0.000376 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1 

Residual standard error: 0.6788 on 7 degrees of freedom
Multiple R-squared:  0.9911,    Adjusted R-squared:  0.9886 
F-statistic: 391.4 on 2 and 7 DF,  p-value: 6.554e-08
```

> 0.679^2*7

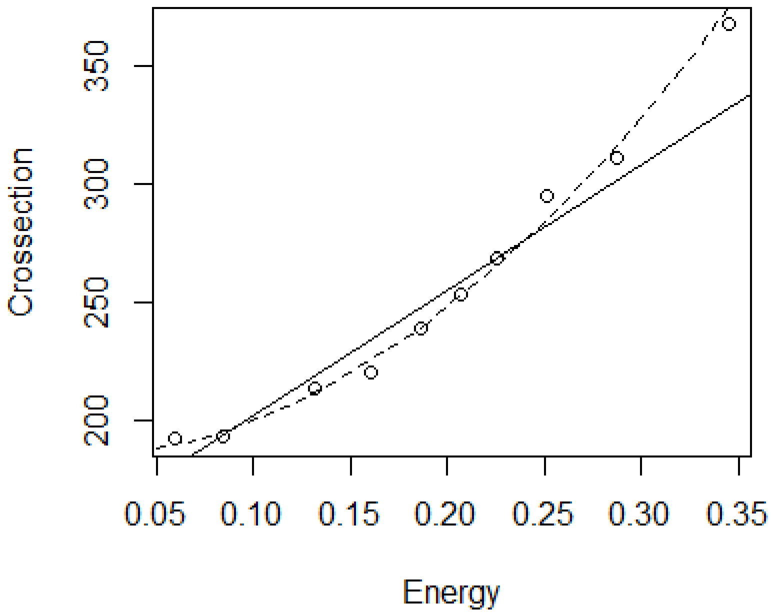
[1] 3.2273

> 1-pchisq(3.32273,7)

[1] 0.85363

Tu jest dobre dopasowanie

```
> x <- seq(0.05,0.35,by=0.01)
> lines(x,g2$coef[1]+g2$coef[2]*x+g2$coef[3]*x^2,lty=2)
```



2. σ^2 jest nieznane

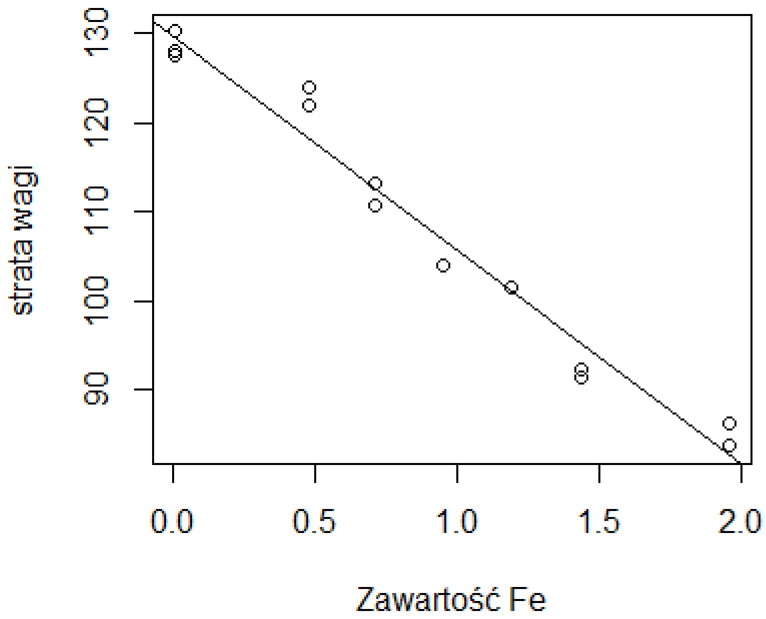
Możliwe wtedy gdy mamy powtórzone x dla tego samego y.

```
      Fe  loss
1  0.01 127.6
2  0.48 124.0
3  0.71 110.8
4  0.95 103.9
5  1.19 101.5
6  0.01 130.1
7  0.48 122.0
8  1.44  92.3
9  0.71 113.1
10 1.96  83.7
11 0.01 128.0
12 1.44  91.4
13 1.96  86.2
```

loss – strata w mg na skutek korozji

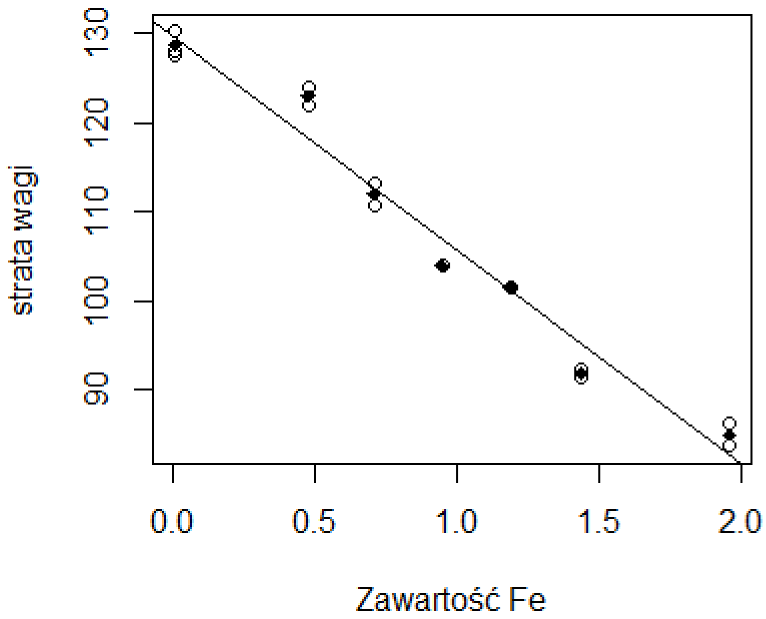
Fe – zawartość żelaza w sopie miedzioniklu [%]

```
> g <- lm(loss ~ Fe, data=corrosion)
> summary(g)

Call:
lm(formula = loss ~ Fe, data = corrosion)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.7980 -1.9464  0.2971  0.9924  5.7429 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  129.787      1.403   92.52  < 2e-16 ***
Fe           -24.020      1.280  -18.77 1.06e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

Residual standard error: 3.058 on 11 degrees of freedom
Multiple R-squared:  0.9697,    Adjusted R-squared:  0.967 
F-statistic: 352.3 on 1 and 11 DF,  p-value: 1.055e-09
```



Potraktujmy to zadanie jako model analizy wariancji z grupam, wyznaczonymi przez poziom Fe:

```
> factor(corrosion$Fe)
 [1] 0.01 0.48 0.71 0.95 1.19 0.01 0.48 1.44 0.71 1.96 0.01 1.44 1.96

Levels: 0.01 0.48 0.71 0.95 1.19 1.44 1.96

ga <- lm(loss ~ factor(Fe), data=corrosion)
summary(ga)
Call:
lm(formula = loss ~ factor(Fe), data = corrosion)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.2500 -0.9667  0.0000  1.0000  1.5333 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)     128.567      0.809 158.914 4.19e-12 ***
factor(Fe)0.48   -5.567      1.279  -4.352  0.00481 ** 
factor(Fe)0.71  -16.617      1.279 -12.990 1.28e-05 ***
factor(Fe)0.95  -24.667      1.618 -15.245 5.03e-06 ***
factor(Fe)1.19  -27.067      1.618 -16.728 2.91e-06 ***
factor(Fe)1.44  -36.717      1.279 -28.703 1.18e-07 ***
factor(Fe)1.96  -43.617      1.279 -34.097 4.24e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1 

Residual standard error: 1.401 on 6 degrees of freedom
Multiple R-squared:  0.9965,    Adjusted R-squared:  0.9931 
F-statistic: 287.3 on 6 and 6 DF,  p-value: 4.152e-07


points(corrosion$Fe,ga$fit,pch=18)
```



Porównanie obu modeli

```
> anova(g,ga)
Analysis of Variance Table

Model 1: loss ~ Fe
Model 2: loss ~ factor(Fe)
  Res.Df     RSS Df Sum of Sq      F   Pr(>F)
1     11 102.850
2      6  11.782  5    91.069 9.2756 0.008623 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Model 1 jest istotnie inny od Modelu 2
[Rysunek 5.4]