cvič 12

1. příklad – test. hypotézy, norm. veličina

a)

```
qnorm(0.95)
```

```
## [1] 1.644854
```

```
qnorm(0.975)
```

```
## [1] 1.959964
```

```
5-qnorm(0.975)
```

```
## [1] 3.040036
```

```
5+qnorm(0.975)
```

```
## [1] 6.959964
```

```
5-qnorm(0.95)
```

```
## [1] 3.355146
```

```
5+qnorm(0.95)
```

```
## [1] 6.644854
```

```
x = rnorm(10, 5, 1); x
```

```
##  [1] 5.680805 7.884434 7.355688 4.949182 5.407089 4.678862 4.380730 3.144230
##  [9] 5.348480 5.489184
```

```
mean(x)
```

```
## [1] 5.431868
```

1c)

```
u = 5 + 1.96/sqrt(10)
l = 5 - 1.96/sqrt(10)
l; u
```

```
## [1] 4.380194
```

```
## [1] 5.619806
```

```
pnorm(u,mean=4,sd=1/sqrt(10)) - pnorm(l,4,1/sqrt(10))
```

```
## [1] 0.1146278
```

```
pnorm((u-4)*sqrt(10)) - pnorm((l-4)*sqrt(10))
```

```
## [1] 0.1146278
```

```r
l2 = -10000
u2 = 5 + 1.64/sqrt(10)
l2; u2
```

```
## [1] -10000
```

```
## [1] 5.518614
```

```r
pnorm(u2,mean=4,sd=1/sqrt(10)) - pnorm(l2,4,1/sqrt(10))
```

```
## [1] 0.9999992
```

```r
pnorm((u2-4)*sqrt(10)) - pnorm((l2-4)*sqrt(10))
```

```
## [1] 0.9999992
```

```r
u3 = 10000
l3 = 5 - 1.64/sqrt(10)
l3; u3
```

```
## [1] 4.481386
```

```
## [1] 10000
```

```r
pnorm(u3,mean=4,sd=1/sqrt(10)) - pnorm(l3,4,1/sqrt(10))
```

```
## [1] 0.06396976
```

```r
pnorm((u3-4)*sqrt(10)) - pnorm((l3-4)*sqrt(10))
```

```
## [1] 0.06396976
```

```r
u = 5
pnorm(u,mean=4,sd=sqrt(10))
```

```
## [1] 0.6240852
```

```r
pnorm((u-4)/sqrt(10))
```

```
## [1] 0.6240852
```

2. příklad – chyba stroje

```r
qbinom(0.95,600,0.03)
```

```
## [1] 25
```

```r
pbinom(24:25,600,0.03)
```

```
## [1] 0.9347241 0.9578847
```

```r
p=0.03
600*p + sqrt(600*p*(1-p))*qnorm(0.95)
```

```
## [1] 24.87305
```

3. příklad – kostka

Vyřešíme dvěma způsoby (přímé dosazení do vzorce nebo použití funkce chisq.test). V obou případech samozřejmě vyjde totéž.

```r
N=50
kostka = sample(6,N, replace=T)
#kostka = c(3,5,1,5,...)
kostka
```

```
##  [1] 3 4 3 5 4 4 1 6 4 2 4 6 4 1 3 1 2 5 6 5 5 5 1 5 5 4 5 5 5 4 6 3 5 5 3 5 2 2
## [39] 3 3 6 6 2 2 5 2 1 6 5 5
```

```r
obs = rep(0,6); obs
```

```
## [1] 0 0 0 0 0 0
```

```r
for(i in 1:6){
  obs[i] = sum(kostka==i)
}
obs
```

```
## [1]  5  7  7  8 16  7
```

```r
sum(obs)
```

```
## [1] 50
```

```r
stopifnot(sum(obs)==N)

T = sum((obs-N/6)^2/(N/6)); T
```

```
## [1] 9.04
```

```r
1-pchisq(T,5)     # P(hodnota T nebo vyšší)
```

```
## [1] 0.1074793
```

```r
qchisq(0.95,5)
```

```
## [1] 11.0705
```

```r
chisq.test(obs)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  obs
## X-squared = 9.04, df = 5, p-value = 0.1075
```

Totéž stručněji.

```r
kostka = sample(6, 100, replace=T)
obs=table(kostka); obs
```

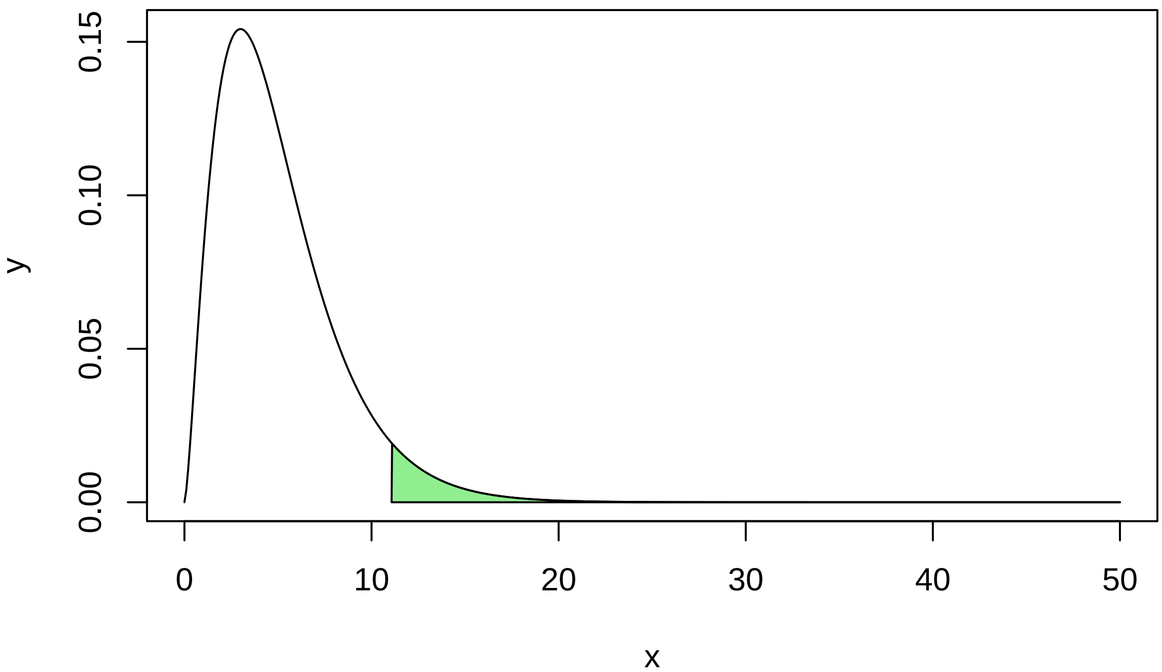
```
## kostka
##  1  2  3  4  5  6
## 17 21 15 12 13 22
```

```r
chisq.test(obs)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  obs
## X-squared = 5.12, df = 5, p-value = 0.4014
```

Pro ilustraci: graf hustoty chi-square rozdělení s pěti stupni volnosti. Zelená plocha odpovídá 5 %, neboli souřadnice jejího začátku je kvantilová funkce toho rozdělení v 0.95, což je cca 11.

```
x = seq(0,50,.1)
y = dchisq(x,5)
plot(x,y, type='l')
h=qchisq(0.95,5)
polygon(c(x[x>=h], max(x), h), c(y[x>=h], 0, 0), col="light green")
```



Pro kontrolu změříme pravděpodobnost chyby prvního druhu:

```
h = qchisq(0.95,5); h
```

```
## [1] 11.0705
```

```
fail = 0
for (i in 1:10^4){
  if (chisq.test(table(sample(6, 100, replace=T)))$statistic > h) {
    fail = fail+1
  }
}
fail/10^4
```

```
## [1] 0.0467
```

4. příklad – emaily

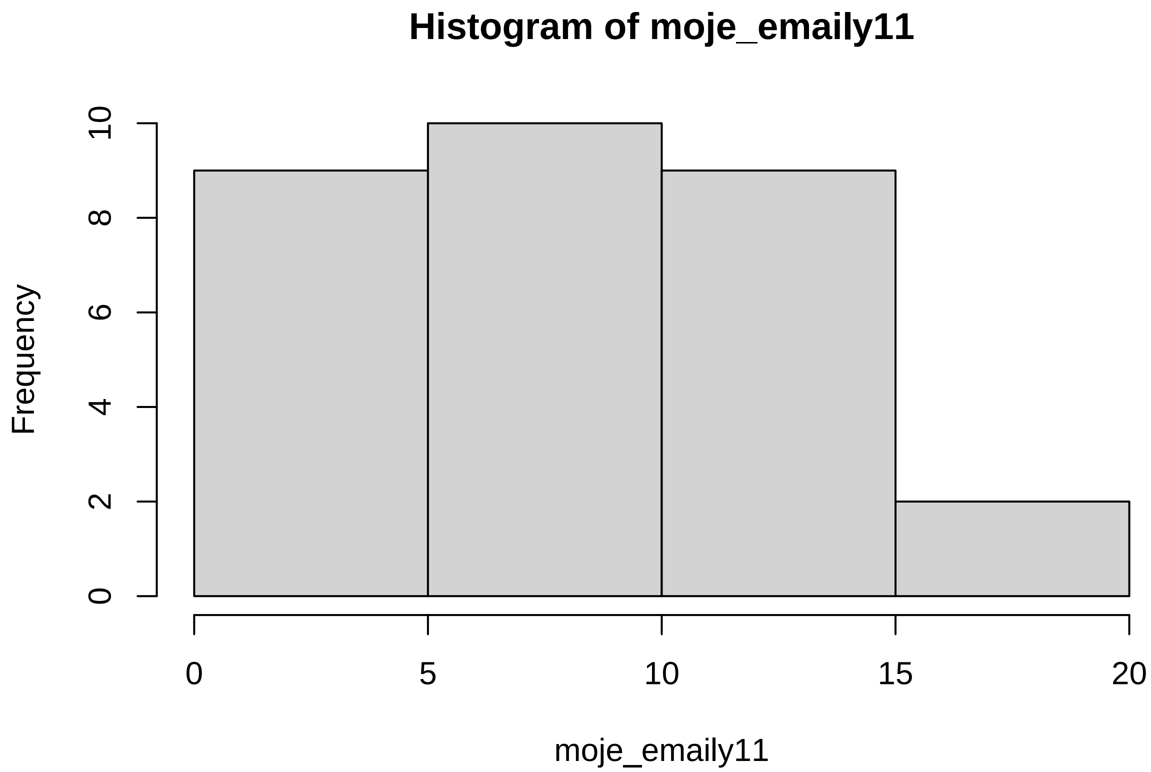
```
moje_emaily11 = c(0,6,14,8,8,9,3,3,12,12,15,7,15,2,5,13,5,17,15,11,9,2,16,8,9,11,6,2,2,9)
#moje_emaily12 = c(13,14,3,8,5,4,12,22,8,4,5,3)
mean(moje_emaily11)
```

```
## [1] 8.466667
```

```
var(moje_emaily11)
```

```
## [1] 23.63678
```

```
hist(moje_emaily11)
```



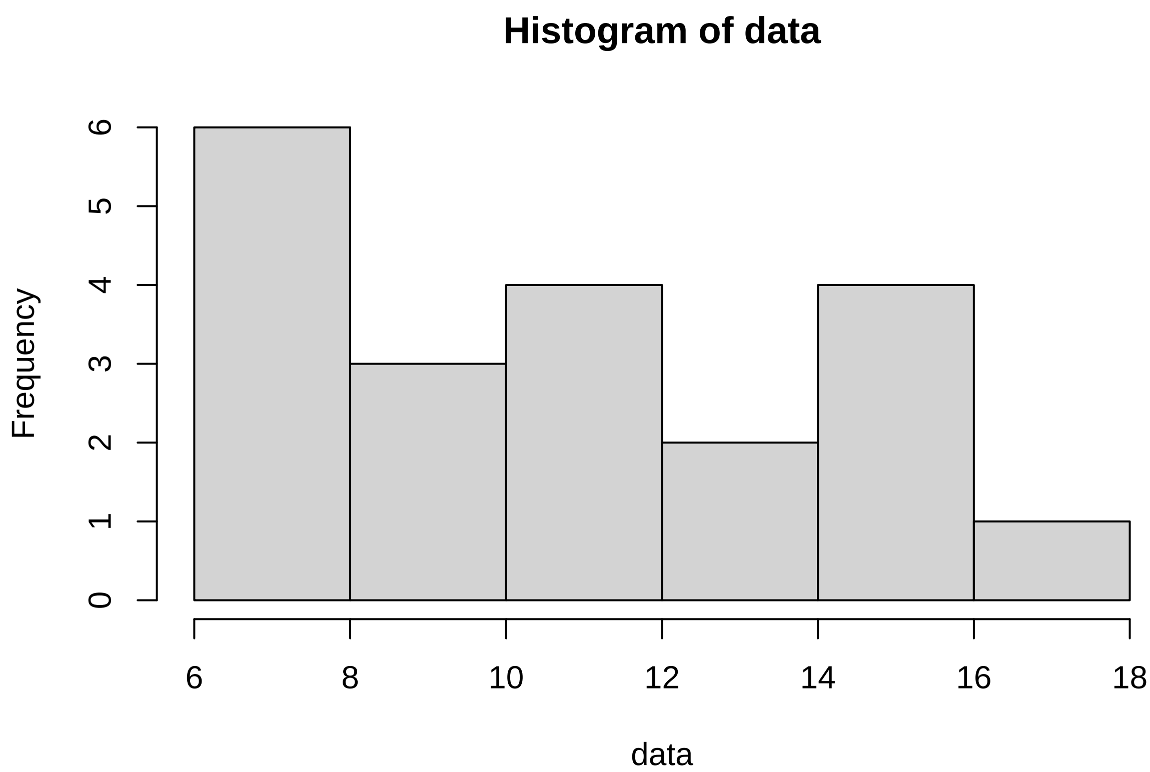

```
day = 1:30
data = moje_emaily11[day %% 7!=1 & day %%7 != 0 & day!=17]
lambda = mean(data); lambda
```

```
## [1] 11.05
```

```
var(data)
```

```
## [1] 12.05
```

```
hist(data)
```




```
#data = moje_emaily12[day %% 7!=5 & day %%7 != 6 & day <= length(moje_emaily12)]
#data
#mean(data)
#var(data)
```

```
n = 15
bins = 0:n
day = 1:30
data = moje_emaily11[day %% 7!=1 & day %%7 != 0 & day!=17]
lambda = mean(data); lambda
```

```
## [1] 11.05
```

```
var(data)
```

```
## [1] 12.05
```

```
p = dpois(bins,lambda)
p[n+1] = 1-ppois(n-1,lambda)
p
```

```
##  [1] 1.588715e-05 1.755530e-04 9.699303e-04 3.572577e-03 9.869243e-03
##  [6] 2.181103e-02 4.016864e-02 6.340907e-02 8.758378e-02 1.075334e-01
## [11] 1.188244e-01 1.193645e-01 1.099148e-01 9.342762e-02 7.374108e-02
## [16] 1.496184e-01
```

```
sum(p)
```

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

```
freq = bins*0
freq = rep(0,n+1)
for(i in bins){ freq[i+1] = sum(data==i) }
```

```r
freq[n+1] = sum(data>=n)
freq
```

```
##  [1] 0 0 0 0 0 0 2 1 3 3 0 2 2 1 1 5
```

```r
N = sum(freq); N
```

```
## [1] 20
```

```r
length(data)
```

```
## [1] 20
```

```r
stopifnot(length(data)==sum(freq))

T = sum((freq-p*N)^2/(p*N))
T
```

```
## [1] 8.153163
```

```r
options(digits = 8)
1-pchisq(T,n)
```

```
## [1] 0.91749538
```

```r
qchisq(0.95,n)
```

```
## [1] 24.99579
```

```r
chisq.test(x=freq,p=p)
```

```
## Warning in chisq.test(x = freq, p = p): Chi-squared approximation may be
## incorrect
```

```
##
##  Chi-squared test for given probabilities
##
## data:  freq
## X-squared = 8.15316, df = 15, p-value = 0.9175
```

Ilustrace toho, jak data sdružovat do menšího počtu přihrádek.

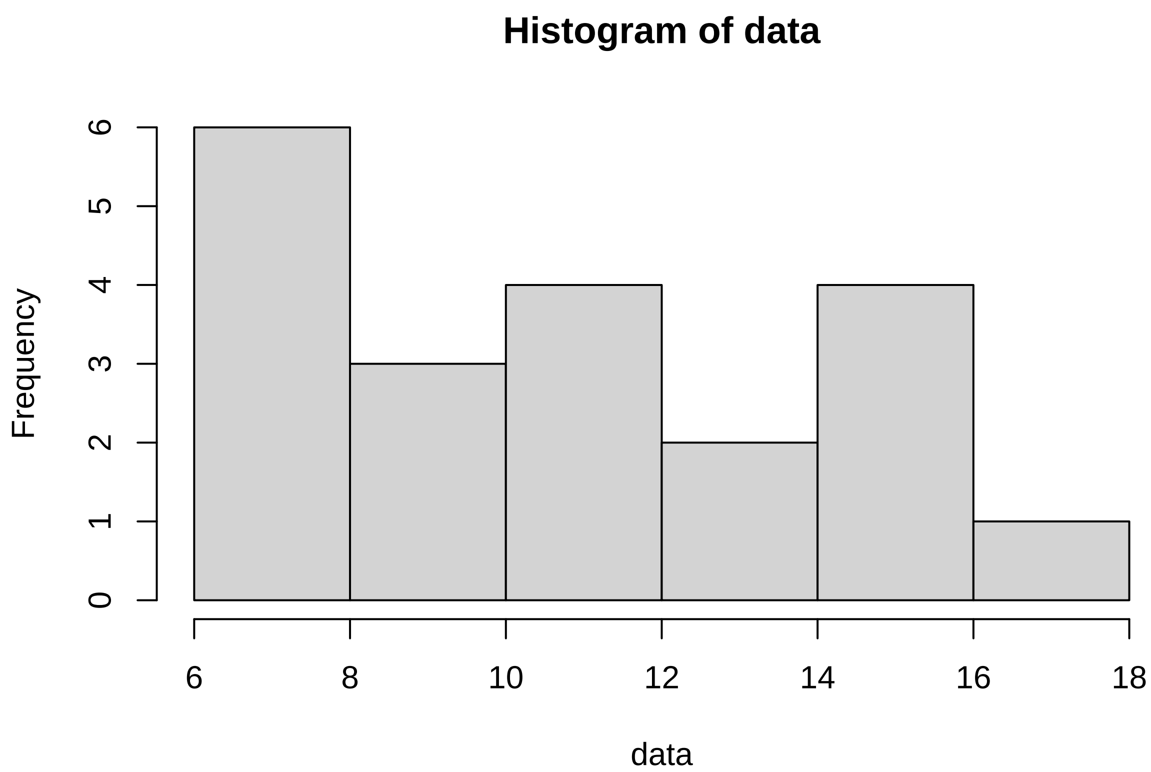
```r
bin_ends = c(-Inf,8,10,12,Inf)  # hraniční body intervalů

day = 1:30
data = moje_emaily11[day %% 7!=1 & day %%7 != 0 & day!=17]
lambda = mean(data); lambda
```

```
## [1] 11.05
```

```r
var(data)
```

```
## [1] 12.05
```

```r
hist(data)
```




```
p = diff(ppois(bin_ends,lambda))
sum(p)
```

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

```
freq = bin_ends*0
for(i in 1:length(bin_ends)){
  freq[i] = sum(data<=bin_ends[i])
}
freq = diff(freq)
freq
```

```
## [1] 6 3 4 7
```

```
N = sum(freq); N
```

```
## [1] 20
```

```
length(data)
```

```
## [1] 20
```

```
T = sum((freq-p*N)^2/(p*N))
T
```

```
## [1] 1.120553
```

```
1-pchisq(T,length(freq)-1)
```

```
## [1] 0.77211498
```

```
qchisq(.95,length(freq)-1)
```

```
## [1] 7.8147279
```

```
chisq.test(x=freq,p=p)
```

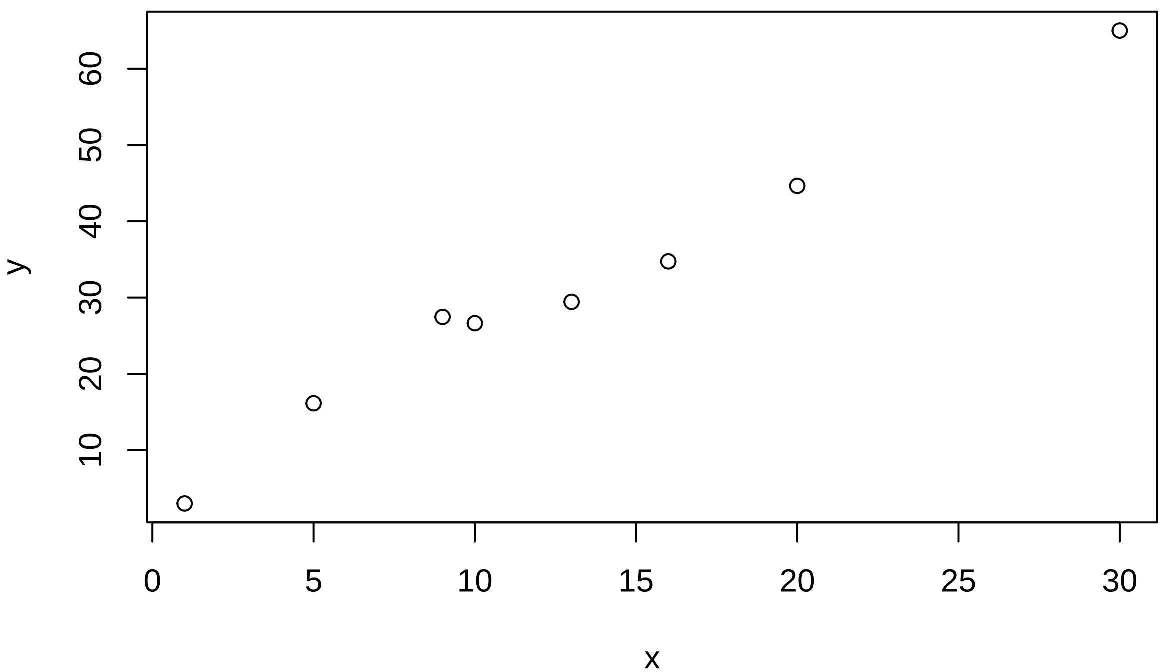
```
## Warning in chisq.test(x = freq, p = p): Chi-squared approximation may be
## incorrect
```

```
##
##  Chi-squared test for given probabilities
##
## data:  freq
## X-squared = 1.12055, df = 3, p-value = 0.77211
```

5. příklad – regrese

V zadání byla řečena data pro x a y. Zde vidíme (v zakomentované části) i jak byla data vyrobena: k ideálnímu vzorci přičteme náhodný šum. Můžeme pak dobře sledovat, jak se spočtené řešení bude lišit od "ideálu".

```
x = c(1,5,9,10,13,16,20,30)
#y = c(6.1982, 12.9892, 23.8005, 23.8891, 30.0391, 35.7535, 49.0685, 63.1825)
y = 2*x+4 + rnorm(length(x),0,3)
#y = 2*x+4 + rnorm(1,0,3)
plot(x,y)
```




```
xm = mean(x)
ym = mean(y)
a = sum((x-xm)*(y-ym))/sum((x-xm)^2); a
```

```
## [1] 2.0110944
```

```
cov(x,y)/var(x)
```

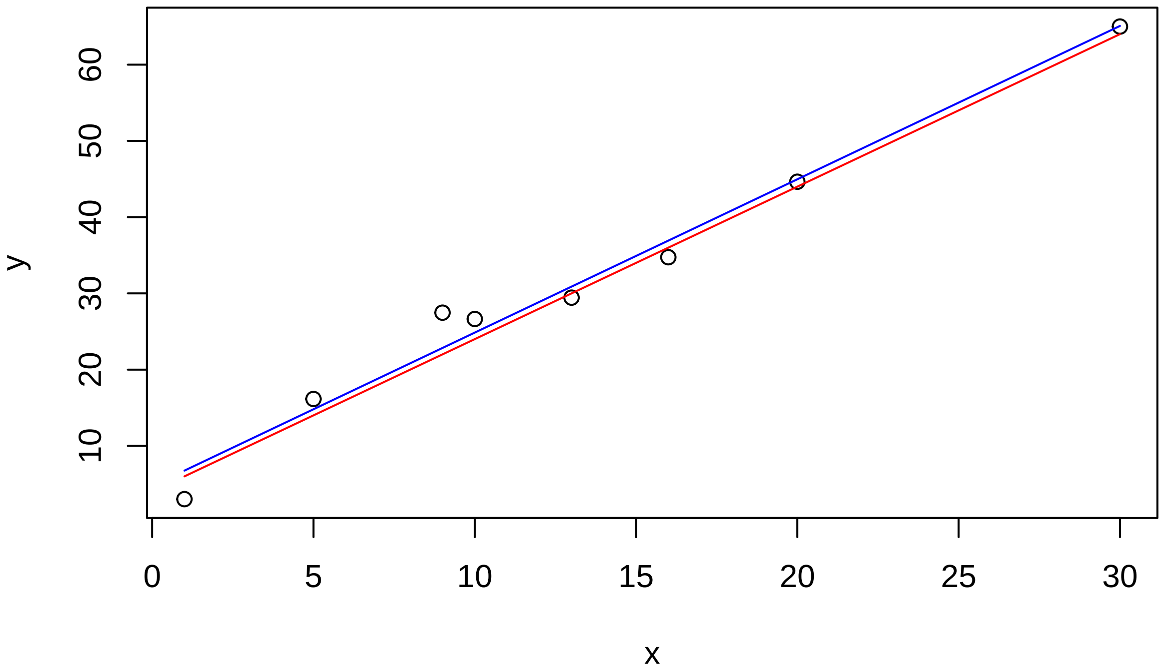
```
## [1] 2.0110944
```

```
b = ym - a*xm; b
```

```
## [1] 4.7448249
```

Červená je ta původní přímka, před přičtením šumu.

```
plot(x,y)
lines(x,a*x+b, col="blue")
lines(x,2*x+4, col="red")
```



Knihovní funkce na regresi ("linear model"). Umí např. i závislost na více proměnných (tak lze např. hledat aproximující polynom).

```
relation <- lm(y~x)

summary(relation)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.73838 -1.62973 -0.20156  1.46086  4.62685
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.74482    1.81286  2.6173  0.03973 *
## x            2.01109    0.11666 17.2395 2.44e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.8094 on 6 degrees of freedom
## Multiple R-squared:  0.98021,    Adjusted R-squared:  0.97691
## F-statistic:  297.2 on 1 and 6 DF,  p-value: 2.4398e-06
```

```r
relation$coefficients
```

```
## (Intercept)           x
##   4.7448249   2.0110944
```

```r
res = y-(a*x+b); res
```

```
## [1] -3.738378193  1.351714336  4.626851371  1.788315630 -1.446768103
## [6] -2.178621705 -0.321432056 -0.081681279
```