

Linearna regresija

Avtor: Žiga Zaplotnik (vp. št.: 28031261)
Srednja Bela, 26. 4. 2009

1. naloga: določi parametra (k,n) najboljše premice za meritve iz vhodne datoteke »HitrostTokaOdFrekvence.txt«. Podane so napake hitrosti, zato lahko določiš tudi χ^2 .

a) Parametra **k** in **n**, če NE upoštevamo napak pri meritvah.

$$k = 1.008084$$
$$n = -2.095472$$

b) Parametra **k** in **n** ter vrednost χ^2 , če upoštevamo napake pri meritvah.

$$k = 0.978093$$
$$n = -1.938642$$
$$\chi^2 = 5.437883$$

Komentar: parametra **k** in **n** sem izračunal s pomočjo lastnih programov, napisanih v programskem jeziku C, χ^2 pa po formuli za najmanjšo vsoto kvadratov razdalj točk od premice. Vsi grafi so narisani v programu *Mathematica 7.0*. Spodaj je prikazano:

1. izpiska glavnih delov obeh programov,
2. primerjava grafov premic, narisanih za parametra **k** in **n** pri točkah a) in b) in
3. graf za parametre iz točke b) in točke (skupaj z njihovimi napakami) iz meritev.

1.)

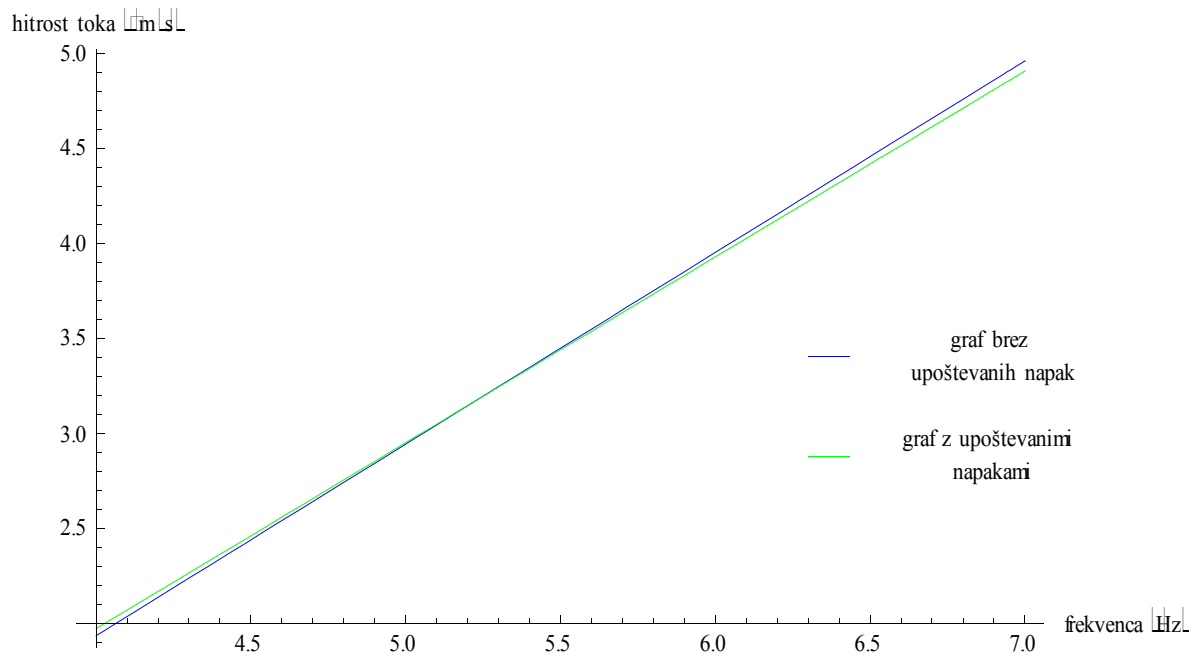
a)

```
for(int i=0; i<9; i++){
    sum1+=1;
    sumx+=x[i];
    sumy+=y[i];
    sumx2+=x[i]*x[i];
    sumxy+=x[i]*y[i];
}
k=(sum1*sumxy-sumx*sumy)/(sum1*sumx2-sumx*sumx);
n=(sumx2*sumy-sumx*sumxy)/(sum1*sumx2-sumx*sumx);
```

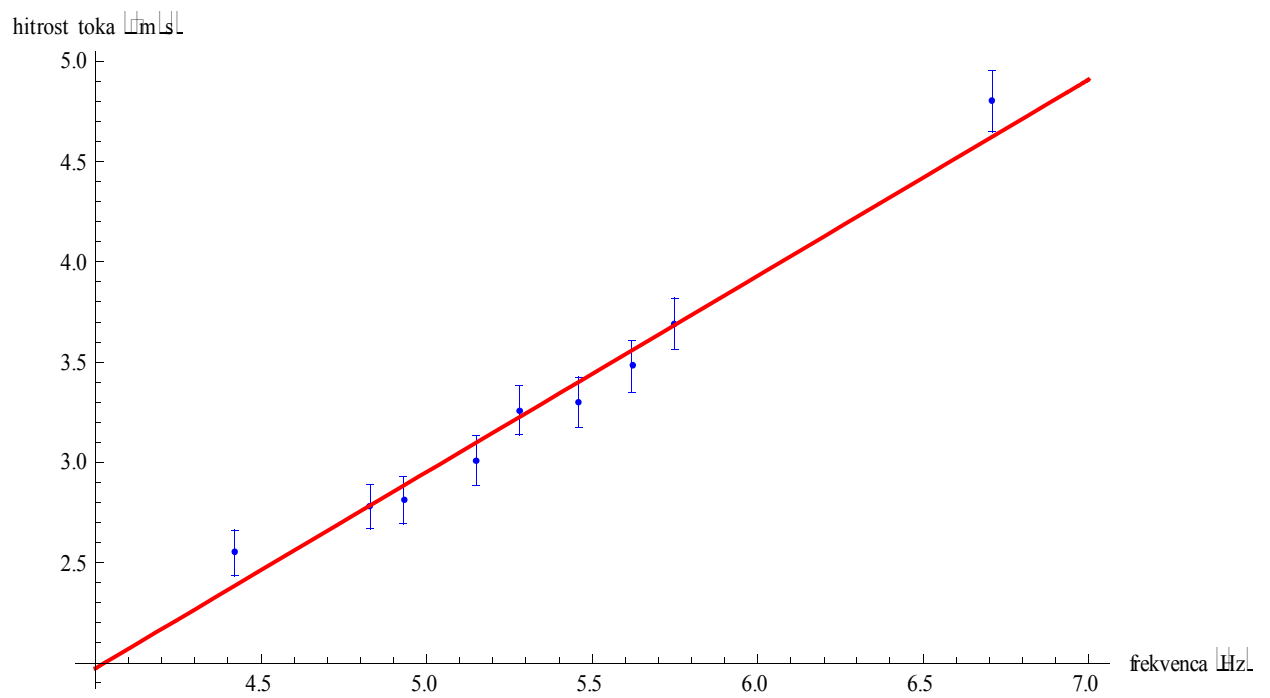
b)

```
for(int i=0; i<9; i++){
    sum1+=1/(napaka[i]*napaka[i]);
    sumx+=x[i]/(napaka[i]*napaka[i]);
    sumy+=y[i]/(napaka[i]*napaka[i]);
    sumx2+=x[i]*x[i]/(napaka[i]*napaka[i]);
    sumxy+=x[i]*y[i]/(napaka[i]*napaka[i]);
}
k=(sum1*sumxy-sumx*sumy)/(sum1*sumx2-sumx*sumx);
n=(sumx2*sumy-sumx*sumxy)/(sum1*sumx2-sumx*sumx);
```

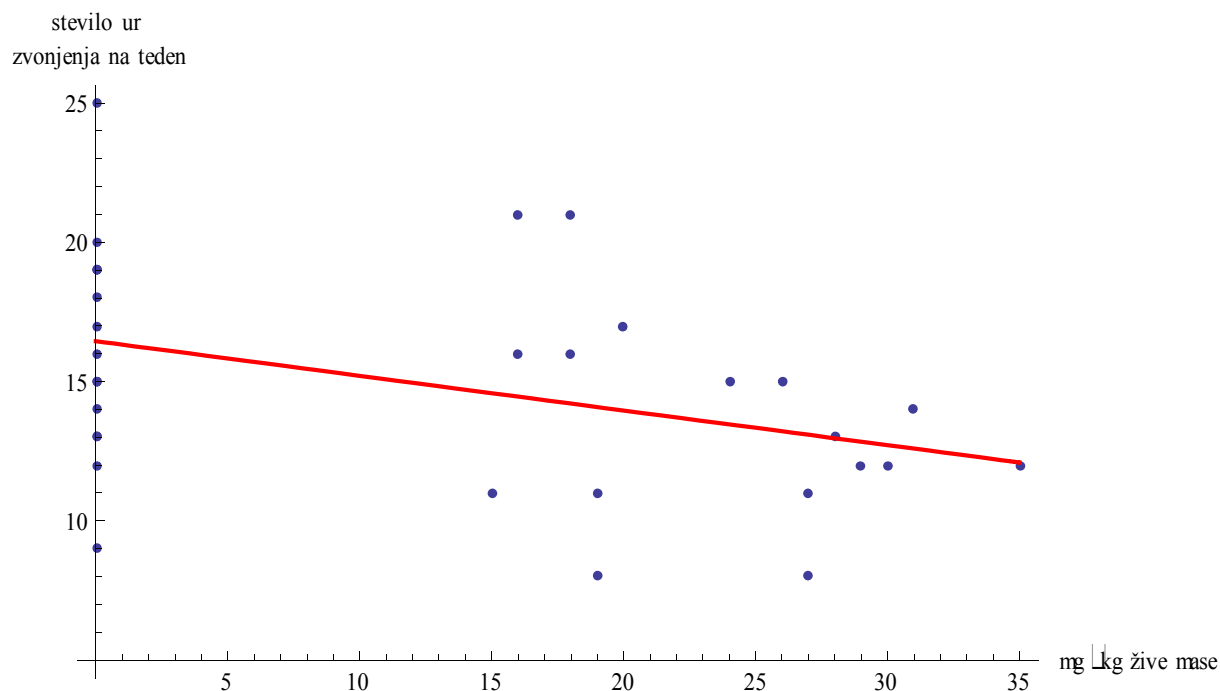
2.)



3.)



2. naloga: skozi oblak podatkov iz datoteke »Tintin.dat« potegni najboljšo premico. Uporabiš lahko kar korelacijske rezultate iz naloge 6.2.



Parametra najboljše premice:

$k = -0.1245$

$n = 16.4546$

Komentar: graf sem naredil s pomočjo programa *Mathematica 7.0*. Podatke za izračun najboljše premice pa sem priredil s pomočjo lastnega programa, napisanega v programskem jeziku C.

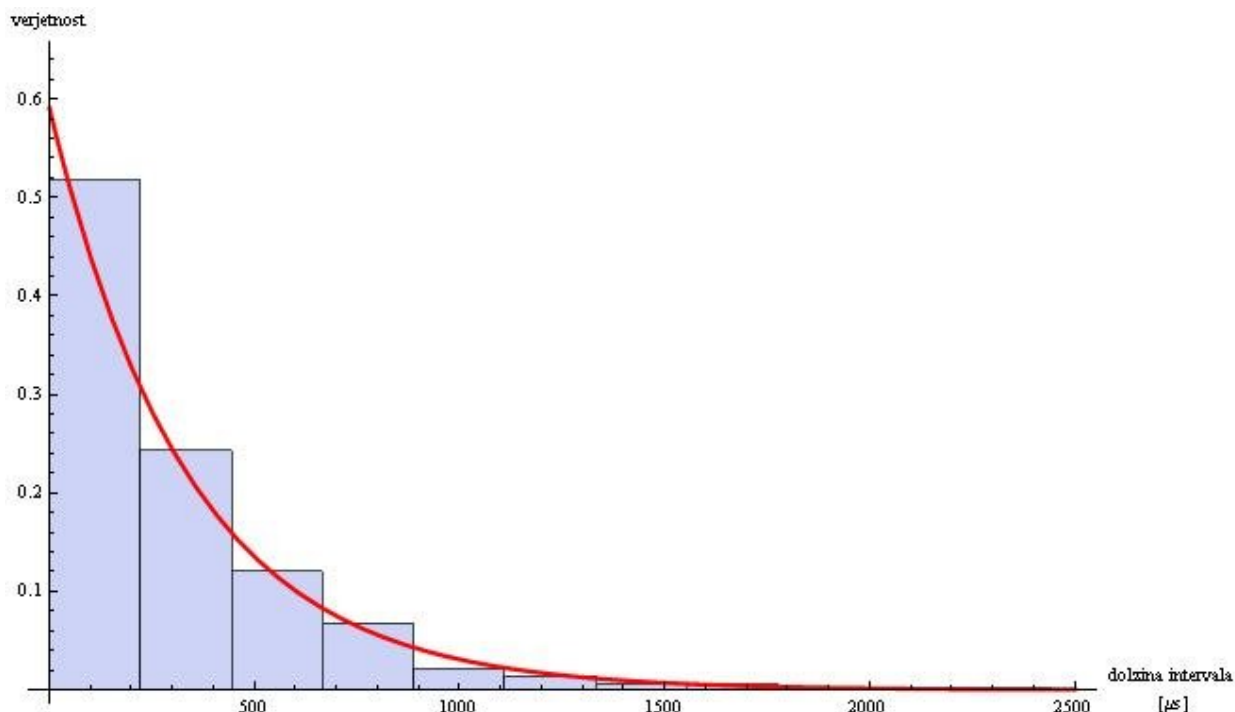
Spodaj je naveden izpisek programa, s katerim sem najprej izračunal povprečji ter standardna odklona za odmerek in čas zvonjenja, nato pa sem uporabil že izračunan korelacijski koeficient iz prejšnje naloge in izračunal še naklonski koeficient premice ter začetno vrednost.

```
for(int i=0; i<32; i++){
    sumzvon+=zvon[i];
    sumodm+=odm[i];
}
zvonpov=sumzvon/32;
odmpov=sumodm/32;

for(int i=0; i<32; i++){
    sumzvon2+=(zvon[i]-zvonpov)*(zvon[i]-zvonpov);
    sumodm2+=(odm[i]-odmpov)*(odm[i]-odmpov);
}
odmsigma=sqrt(sumodm2/32);
zvonsigma=sqrt(sumzvon2/32);

double r=-0.39409;
double k= r*zvonsigma/odmsigma;
double n= zvonpov-k*odmpov; //y=kx+n (enačba za premico)
```

3. naloga: skozi točke v histogramu podatkov »Interval.dat« potegni najboljšo eksponentno funkcijo $w = Ae^{-\lambda x}$.



a) Parametra k in n premice v grafu $y = \ln(w)$ v odvisnosti od x :

$$k = -0.002950$$

$$n = -0.526686$$

b) Parametra λ in A v grafu eksponentne funkcije:

$$\lambda_1 = 0.002950$$

$$A = 0.590559$$

c) Recipročna povprečna vrednost histograma za 11 predalčkov:

$$\lambda_2 = 1/322.24 = 0.003106$$

Komentar: graf eksponentne funkcije ter histogram sem narisal v programu *Mathematica 7.0*. Za primer sem vzel 11 predalčkov. Podatke za graf pa sem preračunal s pomočjo lastnega programa, napisanega v programskem jeziku *C*. Vrednost λ se precej dobro ujema z recipročno povprečno vrednostjo histograma, do majhne razlike pride le zaradi predalčenja. Spodaj je izpisek celotnega programa, ki izračuna vrednosti λ in A v odvisnosti od željenega števila predalčkov.

```

#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <string.h>
#define MAX 1500
int funkcija(char *vhodna, char *izhodna,int st);

int main(void){
    printf("Vnesi stevilo predalckov: ");
    //največ 11, sicer je vsaj en predalček v histogramu enak 0
    int st;
    scanf("%d", &st);
    printf("%d", funkcija("Interval.dat","Interval_izhodna.txt",st));
    return 0;
}

int funkcija(char *vhodna, char *izhodna,int st){
    FILE *vh=fopen(vhodna, "rt");
    FILE *iz=fopen(izhodna, "wt");

    double interval[MAX],meja_bina[MAX],bin[MAX],binsize;

    for(int i=0; i<999; i++){
        fscanf(vh,"%lf", &interval[i]);
    }
    binsize=2441.3/st;    //Max[podatki] = 2441.3

    for(int j=0; j<=st; j++){
        meja_bina[j]=j*binsize;
    }
    for(int j=0; j<st; j++){
        bin[j]=0;
    }

    for(int i=0; i<999; i++){
        for(int j=1; j<=st; j++){
            if(interval[i]>=meja_bina[j-1] && interval[i]<meja_bina[j])
                bin[j-1]+=1;
        }
    }
    double srednja_vrednost_bina[MAX],y[MAX];
    for(int j=0; j<st; j++){
        bin[j]/=999;
        srednja_vrednost_bina[j]=(meja_bina[j+1]+meja_bina[j])/2;
        if(bin[j]!=0) y[j]=log(bin[j]);
        else y[j]=0;
    }

    for(int j=0; j<st; j++){
        fprintf(iz, "%lf    %lf\n", srednja_vrednost_bina[j], bin[j]);
    }

    // sedaj moramo rešiti sistem dveh enačb z dvema neznankama:
    //  $w=A \cdot e^{-mx} \rightarrow \ln(w)=\ln(A)-mx$ 
    //  $y = \ln(w)$ ,  $k = -m$ ,  $n = \ln(A)$ .
    // Po teoriji verjetnosti mora biti koeficient m enak recipročni
    // povprečni vrednosti histograma.

    double suml=0, sumx=0, sumy=0, sumx2=0, sumxy=0;
    double k,n,m,A;

```

```

for(int j=0; j<st; j++){
    sum1+=1;
    sumx+=srednja_vrednost_bina[j];
    sumy+=y[j];
    sumx2+=((srednja_vrednost_bina[j])*(srednja_vrednost_bina[j]));
    sumxy+=srednja_vrednost_bina[j]*y[j];
}

k=(sum1*sumxy-sumx*sumy)/(sum1*sumx2-sumx*sumx);
n=(sumx2*sumy-sumx*sumxy)/(sum1*sumx2-sumx*sumx);

fprintf(iz, "\nk=%lf\nn=%lf\n", k, n);

m=-k;
A=exp(n);

fprintf(iz, "\nA=%lf    lambda=%lf", A, m);

return 0;
}

```

4. naloga: Teorija kemijske kinetike napove za sigmoidno krivuljo iz podatkov "Adrenalin.dat" (naloga 1.1) naslednjo odvisnost $F / F_{\max} = c / (a + c)$, kjer pomeni a koncentracijo s polovičnim maksimalnim učinkom. Določi koeficienta F_{\max} in a . Pretvori v linearno zvezo – ena pot je uvedba recipročnih spremenljivk $1 / F$ in $1 / c$, druga pa je uvedba spremenljivke c / F .

$$\frac{F}{F_{\max}} = \frac{c}{a+c} \rightarrow \frac{a+c}{F_{\max}} = \frac{c}{F} = m \rightarrow$$

$$\rightarrow \frac{a}{F_{\max}} + \frac{c}{F_{\max}} = \frac{c}{F} \rightarrow \frac{a}{F_{\max}} + \frac{1}{F_{\max}} \cdot c = \frac{c}{F}$$

$$\rightarrow y = \frac{c}{F}$$

$$\rightarrow x = c$$

$$\rightarrow k = \frac{1}{F_{\max}} \rightarrow F_{\max} = \frac{1}{k}$$

$$\rightarrow n = \frac{a}{F_{\max}} \rightarrow a = n \cdot F_{\max} = n \cdot \frac{1}{k}$$

$$k = 0.009757$$

$$n = 0.221796$$

$$F_{\max} = \frac{1}{k} = 102.49 \text{ enot}$$

$$a = n \cdot \frac{1}{k} = 22.73 \frac{\mu\text{g}}{\text{L}}$$

Komentar: za novo spremenljivko sem vzel $\frac{c}{F} = m$. Koeficienta k in n sem izračunal s

pomočjo lastnega programa, napisanega v programskem jeziku C, iz njiju pa sem nato izračunal F_{\max} in a . Pri tem sem izvzel 1. in 2. točko, ki precej odstopata od grafa, s tem pa se izognemo tudi deljenju z 0.

Spodaj je prikazano:

1. izpisek glavnega dela programa,
2. graf funkcije za vrednosti F_{\max} in a (modro), primerjan z vrednostmi meritev (rdeče).

1.)

```
for(int i=2; i<8; i++){
    sum1+=1;
    sumx+=c[i];
    sumy+=c[i]/F_Fmax[i];
    sumx2+=c[i]*c[i];
    sumxy+=c[i]*c[i]/F_Fmax[i];
}
k=(sum1*sumxy-sumx*sumy)/(sum1*sumx2-sumx*sumx);
n=(sumx2*sumy-sumx*sumxy)/(sum1*sumx2-sumx*sumx);
```

2.)

