

Skalarni produkt in korelacija - 6. domača naloga

Mateja Gosenca

20. 4. 2009

0.1 1. naloga

Pri prvi nalogi sem določila korelacijski koeficient zveze med frekvenco rotorja in hitrostjo toka. To sem naredila s pomočjo programa, napisanega v c:

```
#include <stdio.h>
#include <math.h>

int main(void) {
    FILE *fin, *fout;
    //fout = fopen("HitrostTokaOdFrekvenceRez.dat", "w");

    float a, b, c;
    int n=0;
    float skalab=0;
    float R;
    float r;
    float a_pov=0;
    float b_pov=0;
    float a_sigma;
    float b_sigma;
    float vsotaA=0;
    float vsotaB=0;
    float vsotaA_2=0;
    float vsotaB_2=0;
    int s=0;

    //izracun povrpecja
    fin = fopen("HitrostTokaOdFrekvence.dat", "r");
    while(fscanf(fin, "%f %f %f", &a, &b, &c) == 3){
        vsotaA+=a;
        vsotaB+=b;
        s++;
    }
    fclose(fin);
    a_pov=vsotaA/s;
    b_pov=vsotaB/s;
```

```

//izracun sigme (2 prehoda)
    fin = fopen("HitrostTokaOdFrekvence.dat", "r");
s=0;
a=0;
b=0;
    while(fscanf(fin, "%f %f %f", &a, &b, &c) == 3){
vsotaA_2+=(a-a_pov)*(a-a_pov);
vsotaB_2+=(b-b_pov)*(b-b_pov);
s++;
}
fclose(fin);
a_sigma=sqrt(vsotaA_2/s);
b_sigma=sqrt(vsotaB_2/s);

//izracun skalarnega produkta
    fin = fopen("HitrostTokaOdFrekvence.dat", "r");
    while(fscanf(fin, "%f %f %f", &a, &b, &c) == 3){
skalab+=a*b;
n++;
}
fclose(fin);
r=skalab/n;

R=(r-a_pov*b_pov)/(a_sigma*b_sigma);

printf("R=%f\n r=%f\n a_pov=%f\n b_pov=%f\n a_sigma=%f\n b_sigma=%f\n skalab=%f\n", R, r,

//fclose(fout);
    return 0;
}

```

Kot rezultat sem dobila Korelacijski koeficient $R=0.98797$. Ker je ta koeficient precej visok sklepam, da sta količini zelo linearno odvisni.

0.2 2. naloga

Za reševanje 2. naloge sem uporabila isti program in tako dobila $R=-0.39409$. Vidimo, da je ta rezultat po absolutni vrednosti manjši, ker količini nista tako očitno linearno povezani.

0.3 3. naloga

Za lažjo predstavo sem najprej narisala diagram, ki prikazuje spreminjanje temperature v odvisnosti od časa in je viden na sliki (1). Nato sem program iz 1. naloge malo popravila in tako dobila

```

#include <stdio.h>
#include <math.h>

```

```

int main(void) {
    FILE *fin, *fout;
    fout = fopen("HitrostToka0dFrekvenceRez.dat", "w");

    int i;
    float a, b, c, e[590], f[590];
    int n=0, s=0, l=0;
    float skalab=0;
    float R=0;
    float r=0;
    float a_pov=0;
    float b_pov=0;
    float a_sigma;
    float b_sigma;
    float vsotaA=0;
    float vsotaB=0;
    float vsotaA_2=0;
    float vsotaB_2=0;

    //izracun povrpecja
    fin = fopen("Beton.dat", "r");
    while(fscanf(fin, "%f %f %f", &c, &a, &b) == 3){
        vsotaA+=a;
        vsotaB+=b;
        s++;
    }
    fclose(fin);
    a_pov=vsotaA/s;
    b_pov=vsotaB/s;

    //izracun sigme (2 prehoda)
    fin = fopen("Beton.dat", "r");
    a=0;
    b=0;
    vsotaA_2=0;
    vsotaB_2=0;
    while(fscanf(fin, "%f %f %f", &c, &a, &b) == 3){
        vsotaA_2+=(a-a_pov)*(a-a_pov);
        vsotaB_2+=(b-b_pov)*(b-b_pov);
        l++;
    }
    fclose(fin);
    a_sigma=sqrt(vsotaA_2/l);
    b_sigma=sqrt(vsotaB_2/l);
}

```

```

//izracun skalarnega produkta
fin = fopen("Beton.dat", "r");

    while(fscanf(fin, "%f %f %f", &c, &a, &b) == 3){
e[n]=a;
f[n]=b;
n++;
}
fclose(fin);

for(i=0; i<30; i++){

skalab=0;
for(n=0; n<(591-i); n++){
skalab+=(f[n]*e[n+i]);
}

R=((skalab/(591-i))-a_pov*b_pov)/(a_sigma*b_sigma);
fprintf(fout, "%d %f\n", i, R);
}

fclose(fout);
return 0;
}

```

S pomočjo tega programa sem narisala nov graf, ki prikazuje korelacijsko funkcijo

$$K(y) = (a(x), b(x + y)) \quad (1)$$

Iz grafa (2) oziroma iz izračunanih podatkov sem določila efektivno zakasnitev in dobila 5 časovnih enot, kar ustreza približno 72 minutam.

0.4 4. naloga

Pri 4. je bilo potrebno iz avtokorelacijske funkcije deklinacije čim bolj natančno določiti periodo Luninega tira. Postopala sem podobno kot pri 3. nalogi, le da tukaj ni šlo za primerjavo dveh funkcij, temveč ene same z njenimi faznimi premiki. Kot rezultat sem dobila periodo 27.3 dni.



