Číselné charakteristiky náhodných vektorov
Regresná priamka

I. Jednoduchá dvojica dátových súborov

a) Nech vektor x predstavuje určité kontrolné body a vektor y hodnoty namerané v týchto bodoch. To znamená, že i-ta zložka vektora y sa viaže výlučne k i-tej zložke vektora x.

```
>> x=1:50+rand(1,50);
>> y=0.01*x.^2-3*x+7;
```

Vzájomnú väzbu vektorov x, y vyjadruje ich kovariancia. Hodnota kovariancie bude východiskom pre výpočet korelačného koeficientu a rovnice regresnej priamky.

```
>> cov(x,y)
ans =   1.0e+003 *

    0.2125   -0.5291
   -0.5291    1.3211
```

Príkaz cov(x,y) alebo cov(x,y,0) počíta korigovanú kovarianciu – tj. pracuje so vzorcom, v ktorom sa delí hodnotou (n-1). Nekorigovanú kovarianciu získame príkazom cov(x,y,1).

Matlab na príkaz cov odpovedá maticou. Hľadanú kovarianciu nájdeme mimo hlavnej uhlopriečky. Na hlavnej uhlopriečke sa nachádzajú variancie vektorov x a y. Overme si to priamym výpočtom kovariancie a variancií:

```
>> n=length(x); xm=mean(x); ym=mean(y); sxy=(x-xm)*(y-ym)'/(n-1)

sxy =
        -529.1250

>> vx=(x-xm)*(x-xm)'/(n-1)
                                    vx =
                                     212.5000
>> vy=(y-ym)*(y-ym)'/(n-1)
                                    vy =
                                     1.3211e+003
```

Výsledky sú potvrdené.

Korelačný koeficient môžeme počítať z hodnôt, ktoré už máme k dispozícii:

```
>> kk=sxy/sqrt(vx*vy)

kk =
      -0.9987
```

Matlab má však svoj vlastný príkaz na výpočet korelačného koeficientu:

```
>> corrcoef(x,y)
ans =
    1.0000   -0.9987
   -0.9987    1.0000
```

Výsledok je opäť v matici.

Maticová odpoveď Matlabu v príkazoch cov a corrcoef je trochu nepohodlná, pretože výsledok musíme ešte zo získanej matice „vybrať“. Prečo nám to Matlab tak komplikuje? Jednou z hlavných príčin je pohodlnosť pri výpočte korelácie a korelačného koeficientu pri väčšom počte vektorov. Jedným príkazom cov získame v matici prehľadne usporiadané korelácie každého vektora s každým.

Pristúpme ku kresleniu regresnej priamky závislosti y od x. Jej smernicu získame z matice sxy:

```
>> k=sxy(1,2)/sxy(1,1)
k =   -2.4900
```

Vzorec regresnej priamky zostavíme na základe poznatku, že musí prechádzať cez bod [xm, ym]:

$$y = ym + k*(x-xm)$$

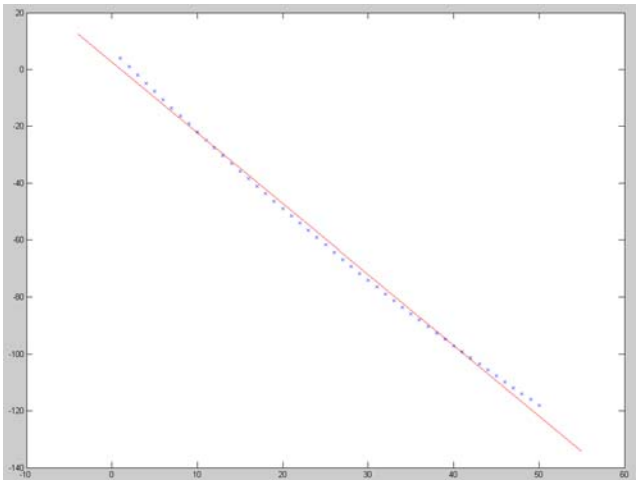
Kreslenie začneme vyznačením bodov daných vektormi x, y (tj. stanovíšť a nameraných hodnôt):

```
>> plot(x,y,'x'), hold on
```

Parametrom 'x' prikazujeme kresliť iba samotné body (nespájať ich čiarami). Vykreslené body majú tvar „voličského“ krížika. Príkaz hold on spôsobí, že ďalšie príkazy na kreslenie sa zrealizujú v existujúcom obrázku (inak by sa zmazal).

Na vykreslenie priamky nám stačia dva body, ktoré potom spojíme červenou (parameter 'r'):

```
>> xe=[min(x)-5, max(x)+5]; ye=ym + k*(xe-xm); plot(xe,ye, 'r')
```



b) Nakreslíme (červenou) regresnú priamku závislosti y od x pre viac rozptýlené hodnoty (x necháme pôvodné):

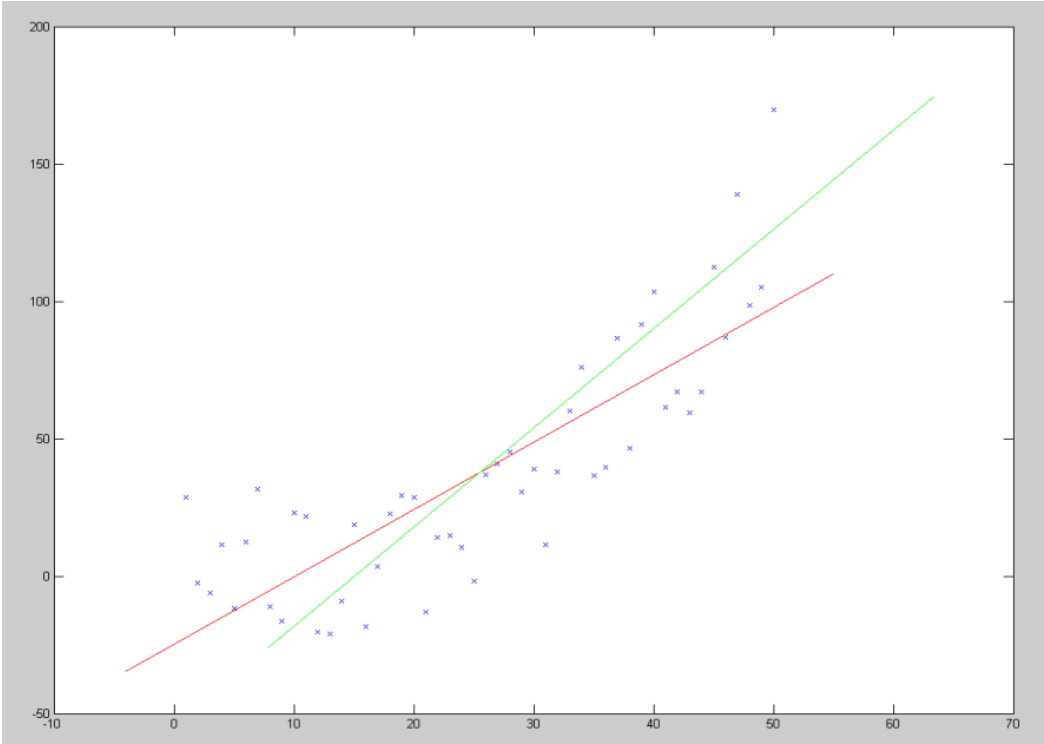
```
>> y=0.1*x.^2-3*x+70*rand(size(x));
>> sxy=cov(x,y); k=sxy(1,2)/sxy(1,1); plot(x,y,'x'), hold on
>> xm=mean(x); ym=mean(y);
>> xe=[min(x)-5, max(x)+5]; ye=ym + k*(xe-xm);
>> plot(xe,ye, 'r')
```

Tu musíme zdôrazniť, že ide o závislosť **y od x**. Opačnú závislosť, x od y, modeluje **iná** regresná priamka, ktorú získame štandardným postupom:

```
>> ki=sxy(1,2)/sxy(2,2); ye=[min(y)-5, max(y)+5]; xe=xm + ki*(ye-ym);
```

Kresliť chceme do pôvodného obrázku, preto musíme nechať pôvodné poradie xe a ye. Na odlíšenie od prvej priamky zvolíme zelenú farbu:

```
>> plot(xe,ye, 'g')
```



Priamky nie sú totožné. Pri hľadaní regresnej priamky preto treba starostlivo rozlišovať medzi primárnou a závislou veličinou.

Otázka na záver: aké sú súradnice bodu, v ktorom sa obe priamky pretínajú?

II. Dvojrozmerné diskrétne rozdelenie

Náhodné veličiny X a Y nadobúdajú nasledujúce hodnoty:

```
>> x= -5:4:99; y=1:0.1:7;
```

Pravdepodobnosť jednotlivých hodnôt [xi,yi] určuje funkcia (1./(abs(15*y-x)+0.1)/q.. Vyrobme tabuľku hodnôt pre všetky dvojice [xi,yi] a nájdime hodnotu q.

Predbežne urobíme tabuľku hodnôt bez koeficientu q. Predstavme si kartézsky súčin vektorov x a y. Do matíc X a Y umiestnime zvlášť prvú a druhú zložku všetkých dvojíc kart. súčinu:

```
>> X=x'*ones(size(y)); Y=ones(size(x'))*y;
```

(Pre lepšie pochopenie si pozrite hodnoty X a Y v „array editore“.)

Teraz dosadíme do F na jedenkrát kartézsky súčin [X, Y]:

```
>> f=inline('1./(abs(15*y-x)+0.1)');
>> fxy=f(X,Y);
```

Poznámka. Maticu fxy môžeme vyrobiť aj dvojicou cyklov:

```
>> for i=1:length(x), for j=1:length(y), fxy(i,j)=f(x(i),y(j)); end, end
```

Aby bola fxy správne definovaná, musí mať súčet 1. Hodnota q teda bude súčtom predbežnej fxy:

```
>> q=sum(sum(fxy))
>> fxy=fxy/q;
```

Poznámka. Vnútorný príkaz *sum* najprv urobil súčty stĺpcov v matici, výsledkom bol riadkový vektor. Vonkajší príkaz *sum* potom sčítal zložky tohto vektora.

Na výpočet číselných charakteristík daného rozdelenia nám už nepomôžu jednoduché príkazy Matlabu, ktoré sme využívali v prvej časti.

Nájdime najprv pravdepodobnostné funkcie samostatných náhodných veličín X a Y:

```
>> fx=sum(fxy,2); fy=sum(fxy, 1);
```

Vektor fx (stĺpec) získame súčtom riadkov fxy (parameter 2 znamená voľbu sčitovania cez riadky), podobne fy (riadok) získame súčtom cez stĺpce (volíme parametrom 1 alebo žiadnym parametrom).

Stredná hodnota je skalárnym súčinom hodnôt vektora a hodnôt jeho pravdepodobnostnej funkcie:

```
>> EX=x*fx, EY=y*fy'

EX =    5.555431068107227e+001
EY =    3.822465130865539e+000
```

Pre varianciu platí: $\mathrm{var}(X) = E(x^2) - (E(x))^2$, kde $E(x^2) = \Sigma_i x_i^2 fx(x_i)$.

Výpočet v Matlabe:

```
>> varx=(x.^2)*fx - EX^2, vary = (y.^2)*fy' - EY^2

varx =    7.291936953583586e+002
vary =    2.985970190200526e+000
```

Pre kovarianciu platí

$$\mathrm{cov}(X,Y) = E(X.Y) - E(X).E(Y),$$

kde

$$E(X.Y) = \Sigma_{ij} fxy_{ij} x_i y_j = x*fxy*y'$$

Výpočet v Matlabe:

```
>> covxy = x*fxy*y' - EX*EY

covxy =    3.808736377137220e+001
```

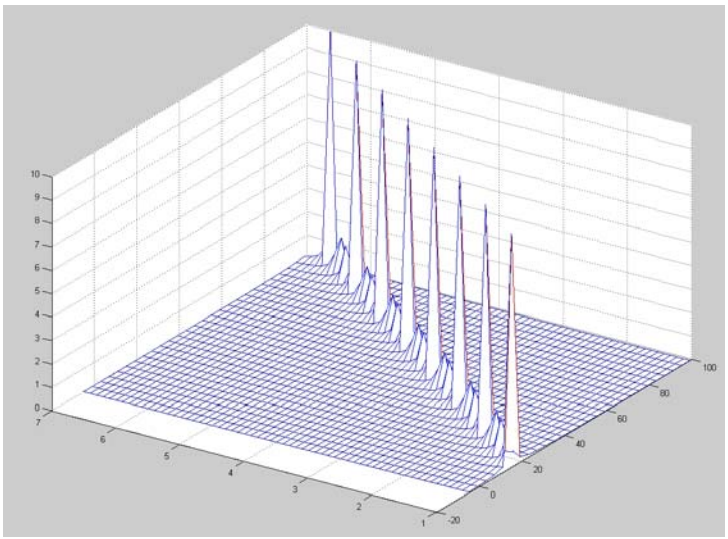
Korelačný koeficient:

```
>> kkxy = covxy / sqrt(varx*vary)

kkxy =  8.162378458268031e-001
```

Korelačný koeficient je cca. 0,82. To znamená, že má zmysel hľadať zobrazenie lineárnej väzby regresnou priamkou.

Nakreslíme si jednotlivé body [x, y] v rovine (tvoria uzly rovnomernej siete), pričom si farebne zvýrazníme ich pravdepodobnostnú „váhu“. Na to slúži príkaz *mesh* (maticu fxy vkladáme transponovanú – taká je syntax):

```
>> mesh(x,y,fxy')
```

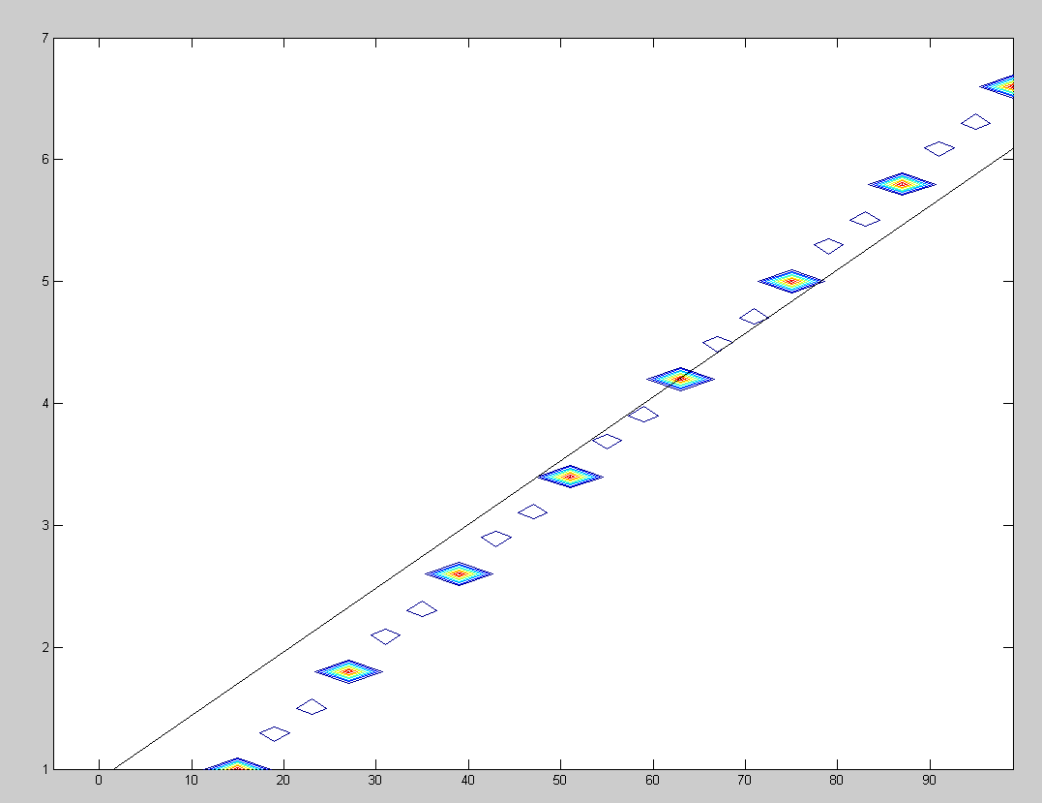


Regresná priamka je dvojrozmerná záležitosť, preto bude vhodné si urobiť „pohľad zhora“ alebo pôdorys:

```
>> contour(x,y,fxy'), hold on
```

Do tohto obrázku zakreslíme regresnú priamku y = EY + (covxy/varx)*(x-EX).

```
>> xk=[min(x)-5, max(x)+5]; yk=EY+(covxy/varx)*(xk-EX); plot(xk,yk,'k')
```



V obrázku vidíme len miesta, ktoré majú najvyššiu pravdepodobnosť. Preto prekvapuje, že priamka neprechádza priamo cez ne. Svoju úlohu tu zohrávajú totiž pravdepodobnostné váhy aj tých bodov, ktoré sa v obrázku stratili v bielej farbe.

II. Dvojrozmerné spojité rozdelenie

Opäť sa budeme venovať príkladu z predošlého odseku. Tentoraz však náhodné veličiny X a Y budú z reálnych intervalov [-5, 99], [1, 7].

Pravdepodobnostnú funkciu nahradí rovnako definovaaná funkcia hustoty (1./(abs(15*y-x)+0.1)/q.

```
>> f=inline('1./(abs(15*y-x)+0.1)');
```

Aby sme zistili hodnotu q, musíme f integrovať cez obdĺžnik [-5, 99] × [1, 7].

```
>> q=dblquad(f, -5, 99, 1, 7)
>> f=inline('1./(abs(15*y-x)+0.1)/69.18959134809417');
```

Funkciu hustoty veličiny X získame integrovaním f podľa y cez interval [1, 7]. Pomocou nej potom analogicky ako v druhom odseku počítame strednú hodnotu a varianciu. Nebudeme pritom funkciu hustoty zvášť vyjadrovať – pri výpočte strednej hodnoty a variancie si vždy v rámci dvojného integrálu funkciu hustoty vyrobíme:

```
>> fxe=inline('x./(abs(15*y-x)+0.1)/69.18959134809417')
>> EX=dblquad(fxe,-5, 99, 1, 7)

        EX =   54.94600124248422

>> fxv=inline('x.^2./(abs(15*y-x)+0.1)/69.18959134809417')
>> varx=dblquad(fxv,-5, 99, 1, 7) - EX^2

                varx =     6.586493430971359e+002
```

Podobne postupujeme v prípade Y:

```
>> fye=inline('y./(abs(15*y-x)+0.1)/69.18959134809417')
>> EY=dblquad(fye,-5, 99, 1, 7) -EY^2

                EY =   3.81060179268186

>> fyv=inline('y.^2./(abs(15*y-x)+0.1)/69.18959134809417')
>> vary=dblquad(fyv,-5, 99, 1, 7)-EY^2

                vary =   2.66618371428309
```

Porovnávajte získané výsledky s hodnotami z predošlého odseku! Mali by vychádzať rádovo zhodné.

Podobne počítame kovarianciu a korelačný koeficient:

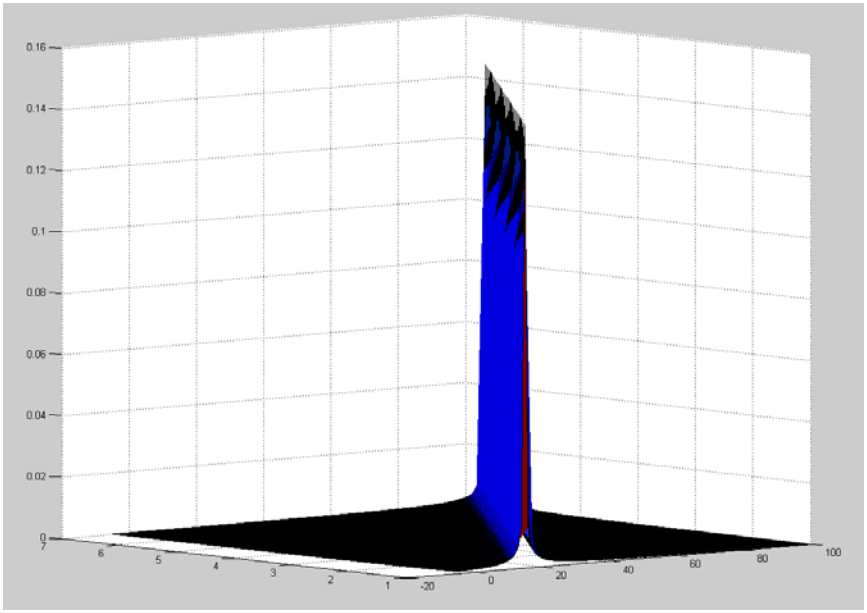
```
>> fcxy=inline('x.*y./(abs(15*y-x)+0.1)/69.18959134809417')
>> covxy=dblquad(fcxy,-5, 99, 1, 7)-EX*EY

                covxy =  31.89079755000236

>> kkxy=covxy/sqrt(varx*vary)

                kkxy =   0.76101499362854
```

Nakreslíme si priestorový graf funkcie hustoty. Na to bude potrebné si diskretizovať intervaly možných hodnôt X a Y.[1] Sieť hodnôt funkcie hustoty vypočítame rovnako ako v predošlom odseku:

```
>> >> x= -5:1:99; y=1:0.1:7;
>> X=x'*ones(size(y)); Y=ones(size(x'))*y;
>> fxy=f(X,Y);
```

Miesto siete, ktorá je iba pomôcku, si necháme vykresliť celú plochu (surface):

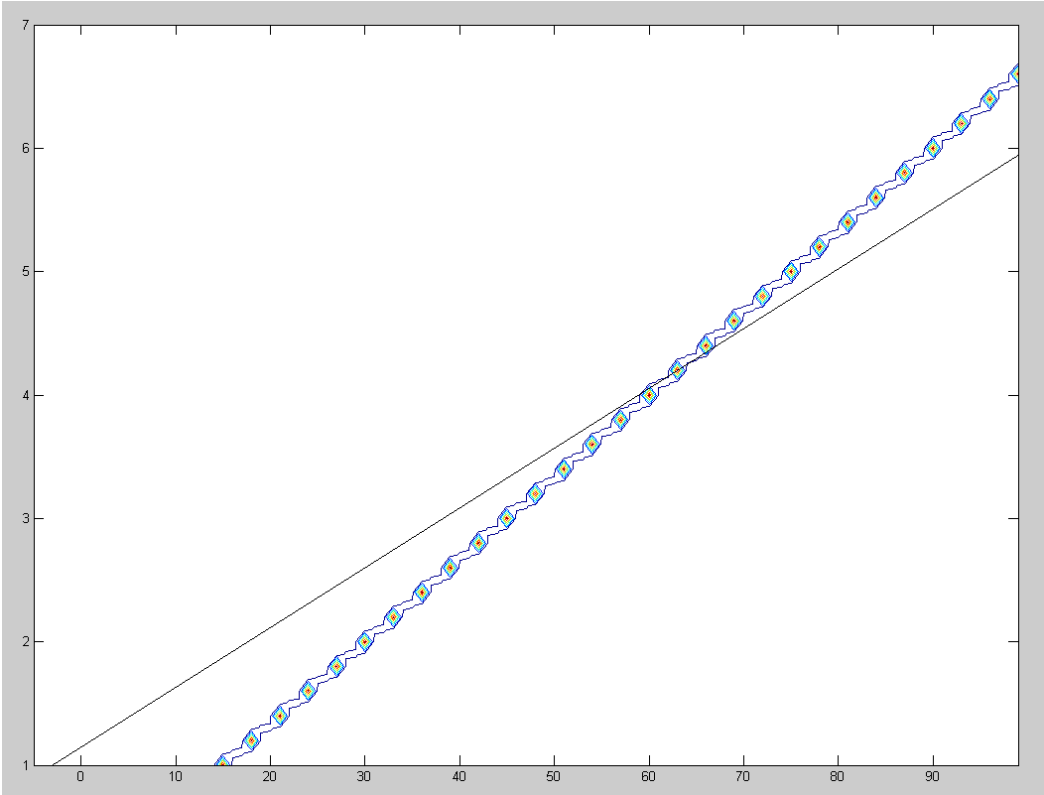
```
>> surf(x,y,fxy')
```



[1] Hustotu diskretizácie treba voliť obozretne – ak je príliš veľká, výpočet bude veľmi dlhý a výsledkom bude tak hustý nákres, že sa všetky farby zlejú do čiernej. Naša voľba je niekde na hranici únosnosti.

Koreláciu opäť znázorníme regresnou priamkou položenou na farebnú mapu:

```
>> xk=[min(x)-5, max(x)+5]; yk=EY+(covxy/varx)*(xk-EX); plot(xk,yk,'k')
```



Opäť vidíme regresnú priamku trochu nakrivo – príčiny sú rovnaké ako v predošlom prípade. Na farebnej mape sa objavili iba najvýraznejšie pravdepodobnosti, avšak aj biela farba predstavuje určitú váhu, ktorá zaváži pri výpočte smernice priamky.