Tehnici exploratorii multivariate

Tehnicile exploratorii multivariate reprezintă acele metode special proiectate pentru a descoperi pattern-uri ascunse în datele multidimensionale, cuprinzând, printre altele:

- *analiza factorială* (*Factor Analysis*),
- *analiza componentelor principale* (*Principal Components Analysis*),
- *analiza canonică* (*Canonical Analysis*) **şi**
- *analiza discriminant* (*Discriminant Analysis*).

Analiza factorială

Analiza factorială constă într-o varietate de tehnici statistice utilizate în scopul reprezentării unui set de variabile în funcţie de un număr mai redus de variabile ipotetice, numite *factori*.

Analiza factorială este utilizată în special pentru rezolvarea următoarelor două probleme:

- *Reducerea* numărului de atribute ale obiectelor în vederea măririi vitezei de procesare a datelor;
- *Detectarea* structurilor ascunse în relaţiile dintre date, în vederea clasificării atributelor obiectelor.

Un exemplu simplu pentru ilustrarea rolului de reducere a datelor și identificarea structurii relațiilor, obținut prin utilizarea analizei factoriale, este cel privitor la procesul de construire a tipologiei consumatorului standard al unui supermarket.

Astfel, în construirea bazei de date a consumatorilor pot apărea simultan atribute ca, de exemplu, venitul anual și impozitul anual. Deoarece cele două atribute sunt corelate prin formula de deducere a impozitului din venituri, este suficient numai unul, celălalt fiind redundant, deci poate fi îndepărtat fără pierdere de informație.

Datorită faptului că cele două atribute (variabile) sunt corelate, relaţia dintre ele este foarte bine rezumată de dreapta de regresie ce trece prin *norul* punctelor generate de perechile de date, putând fi folosită aşadar, pentru detectarea structurii (liniare) a relaţiei dintre ele.

În fapt, reducem astfel cele două variabile la un singur factor, acesta fiind o combinaţie liniară a celor două variabile iniţiale.

Pentru mai mult de două variabile, filosofia reducerii variabilelor la un singur factor rămâne aceeaşi.

Astfel, pentru cazul a trei variabile (corelate), de pildă, putem considera *dreapta* lor de regresie (dreaptă ce trece prin *norul* punctelor generate de tripletele de date, în *spaţiul* tri-dimensional creat de ele) şi astfel le reducem la un singur factor –o combinaţie liniară a lor.

Datele multivariate se pot *suprapune* uneori, în sensul că anumite grupuri pot fi dependente.

De exemplu la decatlon fiecare atlet concurează în 10 probe, dar mai multe probe pot fi considerate a fi de viteză, altele de forţă, etc. Astfel un concurent la 10 probe face de fapt dovada a 3 sau 4 abilităţi atletice, datorită legăturilor existente între anumite probe.

În concluzie, analiza factorială reprezintă acea metodologie statistică care rezumă variabilitatea dintre atributele date, privite ca variabile aleatoare, cu ajutorul unui număr restrâns de alte variabile -*factori*.

Atributele (variabilele) luate în consideraţie sunt exprimate prin combinaţii liniare ale factorilor la care se adaugă şi un termen desemnând *eroarea* modelului.

Analiza factorială este intens utilizată în diferite domenii ca: psihologie (e.g. psihometrie – C. Spearman), ştiinţele sociale, marketing, managementul producţiei.

1. exemplu

Presupunem că staff-ul unui lanţ de supermarket-uri doreşte să măsoare gradul de satisfacţie a cumpărătorilor relativ la serviciile oferite.
Pentru aceasta se consideră doi *factori* de estimare ai satisfacţiei:

(*a*) gradul de satisfacţie privind modul de servire a unui client şi
(*b*) gradul de satisfacţie privind calitatea produselor comercializate.

Pentru aceasta se efectuează un sondaj printre $N = 1000$ de clienţi fideli, care trebuie să dea răspuns unui chestionar cu $M = 10$ întrebări *cheie* care pot măsura gradul de satisfacţie a clienţilor.

Vom considera fiecare răspuns al unui client ca un *scor* referitor la chestiunea respectivă, acesta fiind considerat ca o variabilă *observată*. Deoarece clienţii au fost selectaţi aleator dintr-o *populaţie* numeroasă, se poate presupune că cele 10 răspunsuri (scoruri) reprezintă variabile aleatoare.

Să presupunem, de asemenea, că scorul mediu *per* client, *per* întrebare poate fi privit ca o combinaţie liniară a celor două tipuri de satisfacţii (factori de satisfacţie –variabile *neobservate*), de exemplu, pentru întrebarea nr k, $k = 1, 2,..., 10$, avem:

$\{7*satisfacţie\ serviciu + 5*satisfacţie\ produs\}$,

unde numerele 7 şi 5 se numesc *încărcările factorilor* (*factor loadings*) şi sunt identice pentru toţi clienţii.

Să remarcăm că pot exista încărcări diferite pentru chestiuni diferite, 7 şi 5 fiind încărcările factorilor relativ la chestiunea nr k.

Doi clienţi cu acelaşi grad de satisfacţie în cele două direcţii pot avea scoruri diferite la aceeaşi întrebare din chestionar, deoarece părerile individuale diferă faţă de medie, această diferenţă reprezentând *eroarea*.

Vom prezenta în continuare modelul matematic corespunzător analizei factoriale, adaptat exemplului de mai sus.

Astfel, pentru fiecare client i din cei N clienţi, cele M scoruri sunt date de ecuaţiile:

$$x_{1,i} = b_1 + a_{1,1} \cdot s_{1i} + a_{1,2} \cdot s_{2i} + \varepsilon_{1,i}, \; i = 1, 2, \ldots, N$$

$$x_{2,i} = b_2 + a_{2,1} \cdot s_{2i} + a_{2,2} \cdot s_{2i} + \varepsilon_{2,i}, \; i = 1, 2, \ldots, N$$

.....

$$x_{M,i} = b_M + a_{M,1} \cdot s_{Mi} + a_{M,2} \cdot s_{Mi} + \varepsilon_{M,i}, \; i = 1, 2, \ldots, N$$

unde:

- $x_{k,i}$ reprezintă scorul corespunzător întrebării nr k pentru clientul nr i;
- s_{1i} reprezintă gradul de satisfacţie privind *modul de servire* a clientului nr i;
- s_{2i} reprezintă gradul de satisfacţie privind *calitatea produselor comercializate* a clientului nr i;
- a_{kj} reprezintă încărcările factorilor pentru întrebarea #k corespunzătoare factorului j, $j = 1, 2$;
- ε_{ki} reprezintă eroarea (i.e. diferenţa între scorul clientului nr i pentru chestiunea nr k şi scorul mediu corespunzător chestiunii nr k pentru toţi clienţii ale căror satisfacţii referitoare la servicii şi produse sunt aceleaşi ca la clientul nr i);
- b_k sunt nişte constante aditive.

În limbaj matricial, ecuaţiile de mai sus se transpun în următoarea ecuaţie:

$$X = b + AS + \varepsilon ,$$

unde:

- X este matricea variabilelor aleatoare *observate*;
- b este vectorul constantelor *neobservate*;
- A este matricea încărcărilor factorilor (constante *neobservate*);
- S este o matrice de variabile aleatoare *neobservate*;
- ε este o matrice de variabile aleatoare *neobservate* (matricea *erorilor*).

Analiza factorială îşi propune estimarea matricei A a încărcărilor factorilor, a vectorului mediilor b şi a dispersiei erorilor ε.

Trebuie subliniată distincţia între filosofia din spatele tehnicilor de analiza factorială şi aplicarea efectivă a acestor tehnici pe date concrete.

Practic, analiza factorială se poate aplica numai utilizând programe specializate pe această topică.

În modelul de analiză factorială variabilele considerate depind de un număr mai mic de factori *latenţi*.

Fiecare asemenea factor afectează mai multe variabile si astfel aceşti factori sunt cunoscuţi sub numele de *factori comuni*.

Se presupune că fiecare variabilă este o combinaţie liniară de factori comuni, ai căror coeficienţi sunt cunoscuţi sub numele de *factori de încărcare* (saturaţie)

Fiecare variabilă are o componentă datorată variabilităţii aleatoare independente, cunoscută sub numele de *dispersia specifică*.

Fiecare variabilă are o componentă datorată variabilității aleatoare independente, cunoscută sub numele de *dispersia specifică*.

Analiză factorială presupune că matricea de covarianță a datelor este de forma:

SigmaX = Lambda*Lambda' + Psi

unde **Lambda** este matricea factorilor de încărcare iar matricea diagonală **Psi** are ca elemente dispersiile specifice.

In Matlab funcția **factoran** construiește modelul de analiză factorială utilizând metoda *maximum likelihood*.

2. Exemplu (Matlab)

120 de studenţi au de dat 5 examene, două de literatură, două de matematică şi unul interdisciplinar.
Unii studenţi dovedesc abilităţi atât literatură cât şi matematică, alţii au probleme la unul dintre cele două domenii.

Vrem să găsim factorii comuni ce influenţează notele la examene
Scopul acestei analize constând în a stabili dacă există dovezi cantitative că notele obținute de studenți la cele 5 examene sunt determinate doar de cele două tipuri de abilităţi menţionate.

Vom încărca baza de date şi apoi apelăm **factoran** impunând un model cu un singur factor comun:

```
>> load examgrades
>> [Loadings1,specVar1,T,stats] = factoran(grades,1)
Loadings1 =
    0.6021
    0.6686
    0.7704
    0.7204
    0.9153
specVar1 =
    0.6375
    0.5530
    0.4065
    0.4810
    0.1623
T =
     1
stats =
    loglike: -0.1046
        dfe: 5
      chisq: 12.1211
          p: 0.0332
```

Primele argumente returnate de factoran sunt factorii de încărcare - Loadings1 şi dispersiile specifice- specVar1.
Din factorii de încărcare se observă că modelul cu un singur factor comun atribuie ponderi positive mari tuturor celor 5 variabile, dar în special pentru cea de-a 5-a, examenul interdisciplinar.

O interpretare posibilă este că studentul trebuie privit în termenii de *capacitatea sa în ansamblu*, în acest sens măsura convenabilă fiind examenul interdisciplinar. Notele la celelalte examene, care au un grad de specificitate mai mare, depind de abilitatea generală a studentului dar şi de faptul că acesta este mai bine pregătit în domeniul respectiv, ceea ce explică şi valoarea mai mică a factorilor de încărcare.

Din estimarea dispersiilor specifice observăm că modelul indică faptul că nota obținută de un student la primele patru examene variază destul de mult în comparaţie cu nota obtinută la examenul interdisciplinar.

Dacă dispersia specifică este 1, atunci nu există o componentă a factorului comun în această variabilă, în timp ce dispersia specifică nulă indică faptul că variabilă este determinată în intregime de factorii comuni.

Rezultatele obţinute la cele 4 examene par a fi determinate de factorul comun în proporţie de cel mul 50%.

Valoarea p = 0.0332 din structura **stats** respinge ipoteza nulă a unui singur factor comun, deci suntem obligate să reconstruim modelul.

Incercăm să explicăm mai bine rezultatele examenelor.

Dacă folosim doi sau mai mulţi factori comuni (latenţi), putem roti factorii de încărcare, încercând astfel să dăm o interpretare mai simplă.

Deocamdată nu aplicăm rotaţia.

```
>>load examgrades
>> [Loadings2,specVar2,T,stats] = factoran(grades,2,'rotate','none')
Loadings2 =
    0.6289    0.3485
    0.6992    0.3287
    0.7785   -0.2069
    0.7246   -0.2070
    0.8963   -0.0473
```

```
specVar2 =
   0.4829
   0.4031
   0.3512
   0.4321
   0.1944
T =
    1     0
    0     1
stats =
   loglike: -0.0012
       dfe: 1
     chisq: 0.1422
         p: 0.7061
```

Din factorii de încărcare estimaţi observăm că primul factor comun aplică ponderi aproximativ egale tuturor variabilelor, în timp ce pentru al doilea primele două variabile contrastează cu următoarele două.

Putem interpreta aceşti factori comuni ca abilitate cantitativa vs abilitate calitativă, extinzând interpretarea ce am dat-o modelului cu un factor latent.

Din factorii de încărcare estimaţi observăm că primul factor comun aplică ponderi aproximativ egale tuturor variabilelor, în timp ce pentru al doilea primele două variabile contrastează cu următoarele două.

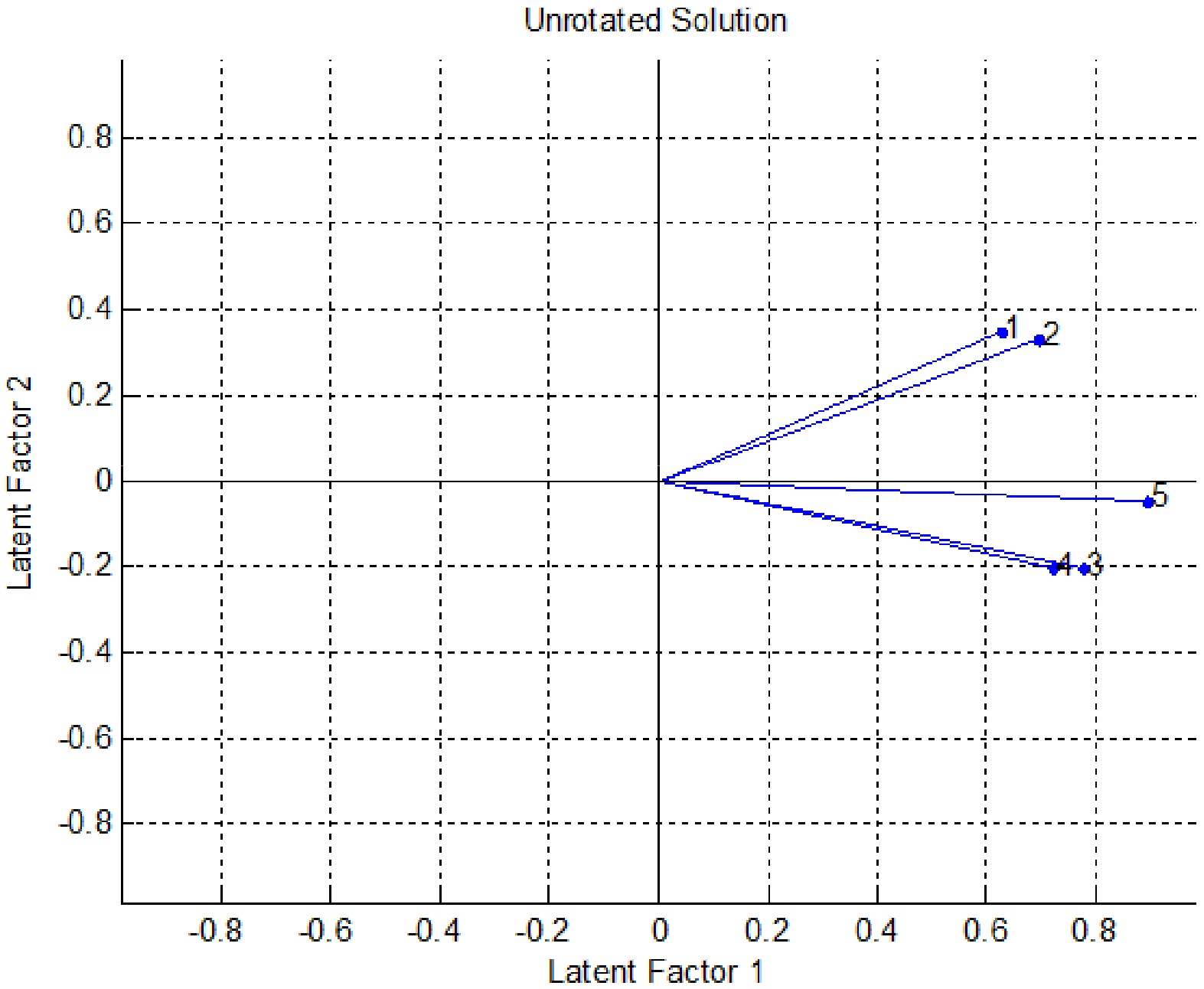
Putem interpreta aceşti factori comuni ca abilitate cantitativa vs abilitate calitativă, extinzând interpretarea ce am dat-o modelului cu un factor latent..

O ilustrare a acestei interpretări constă în desenarea unui grafic al variabilelor, în condiţiile în care fiecare factor de încărcare este o coordonată pe axele corespunzătoare celor 2 factori comuni.

Pentru cel de-al doilea factor primele două examene au ponderi positive, ceea ce sugerează că depind de abilitatea cantitativă, în timp ce celelalte două examene depind de abilitatea calitativă.

Examenul interdisciplinar are o pondere mică pentru al doilea factor.

```
>>biplot(Loadings2, 'varlabels',num2str((1:5)'));
>>title('Unrotated Solution');
>>xlabel('Latent Factor 1');
>>ylabel('Latent Factor 2');
```


Unrotated Solution
Latent Factor 2
Latent Factor 1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
2
3
4
5

Din dispersiile specifice estimate observăm că acest model cu doi factori comuni are variaţii mai mici decât modelul cu un factor comun.

Cea mai mică dispersie o au notele de la examenul interdisciplinar.

Structura **stats** arată că în acest model avem un grad de libertate. **dfe**: 1

Având un număr mic de variabile, nu putem construi un model care să aibă mai mult de doi factori comuni.

Se poate întâmpla să avem doar matricea de covarianţa care sintetizează datele. factoran acceptă şi o matrice de covarianţă sau de corelaţie, utilizând Xtype, iar rezultatul va fi acelaşi ca în cazul datelor brute.

```
>> load examgrades
>> Sigma = cov(grades)
Sigma =
   76.0419   31.6219   26.5967   29.4115   25.1093
   31.6219   42.7982   23.4537   24.2608   21.0167
   26.5967   23.4537   55.2184   38.7902   27.6386
   29.4115   24.2608   38.7902   73.9821   29.8406
   25.1093   21.0167   27.6386   29.8406   27.6554
```

```
>> [LoadingsCov,specVarCov] = ...
factoran(Sigma,2,'Xtype','cov','rotate','none')
LoadingsCov =
    0.6289    0.3485
    0.6992    0.3287
    0.7785   -0.2069
    0.7246   -0.2070
    0.8963   -0.0473
specVarCov =
    0.4829
    0.4031
    0.3512
    0.4321
    0.1944
```

Factorii de încărcare estimaţi de un model de analiză factorială pot da uneori ponderi mari la mai mulţi factori comuni, ceea ce face dificil de răspuns la întrebarea
ce reprezintă aceşti factori comuni (latenţi)?

Scopul rotaţiei factorului este de a găsi o soluţie pentru care fiecare variabilă are un număr mic de ponderi mari, ceea ce înseamnă că este afectată de un număr mic de factori, preferabil de unul singur.

Fiecare linie a matricei de încărcare poate fi considerată a reprezenta coordonatele unui punct intr-un spaţiu m-dimensional, caz în care fiecare factor comun corespiunde unei axe.
Rotaţia factorilor este echivalentă cu rotaţia acestor axe şi permite calcularea de noi factori de incărcare în sistemul de coordonate obţinut prin rotaţie.

Există mai multe metode în acest sens, unele păstrând axele ortogonale, altele schimbând unghiul dintre axe.

O rotaţie **varimax** este o schimbare de coordonate care maximizează suma dispersiilor pătratelor factorilor de încărcare, adică permite descrierea fiecărui subiect ca o combinaţie liniară a unui număr mic de funcţii de bază.

Este o schemă de rotaţie ortogonală prin care cât mai puţine variabile au ponderi mari, restul ponderilor fiind aproape nule.

factoran efectuează implicit rotaţia **varimax**.

```
>> [LoadingsVM,specVarVM,rotationVM] = factoran(grades,2)
LoadingsVM =
    0.2782    0.6631
    0.3456    0.6910
    0.7395    0.3194
    0.6972    0.2860
    0.7332    0.5177
specVarVM =
    0.4829
    0.4031
    0.3512
    0.4321
    0.1944
rotationVM =
    0.7854    0.6190
   -0.6190    0.7854
```

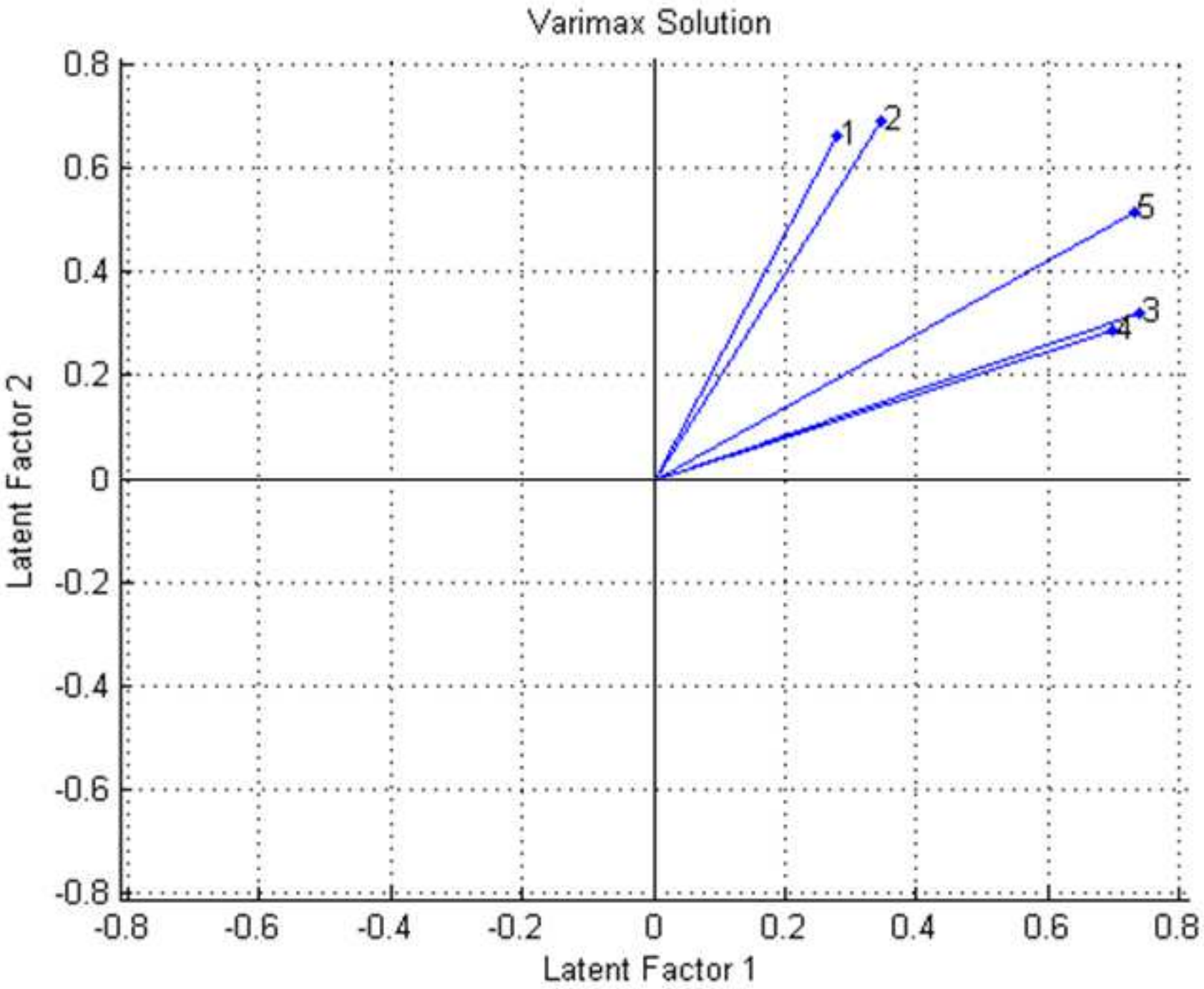
Varimax roteşte axele, păstrând mărimea unghiurilor .

```
rotationVM'*rotationVM
ans =
    1.0000    0.0000
    0.0000    1.0000
```

In statistică se foloseşte un tip de grafic numit **biplot**, care permite ca informaţiile asupra eşantioanelor şi variabilelor dintr-o matrice cu date să fie reprezentate grafic.

Astfel un biplot a celor 5 variabile cu factorii comuni rotiţi ne arată efectul rotaţiei.

```
>> biplot(LoadingsVM, 'varlabels',num2str((1:5)'));
>> title('Varimax Solution');
>> xlabel('Latent Factor 1'); ylabel('Latent Factor 2');
```


Varimax Solution
Latent Factor 2
Latent Factor 1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
2
3
4
5

Varimax a rotit rigid axele pentru a face factorii de incărcare să tindă spre 1 sau 0.

Primele două examene sunt mai apropiate de axa celui de-al doilea factor comun, în timp ce examenele 3 şi 4 sunt apropiate de axa primului factor comun, iar al 5-lea examen este într-o poziţie intermediară.

Aceşti factori obţinuţi prin rotaţie pot fi interpretaţi ca abilitate cantitativă si abilitatea calitativă.

Aceasţa rotaţie ortogonală nu a dus la rezultatul dorit, deoarece niciuna dintre variabile nu se află foarte aproape de axele factorilor comuni.

Rotația **promax** este o rotație non-ortogonală, folosită pentru baze mari de date, fiind foarte rapidă

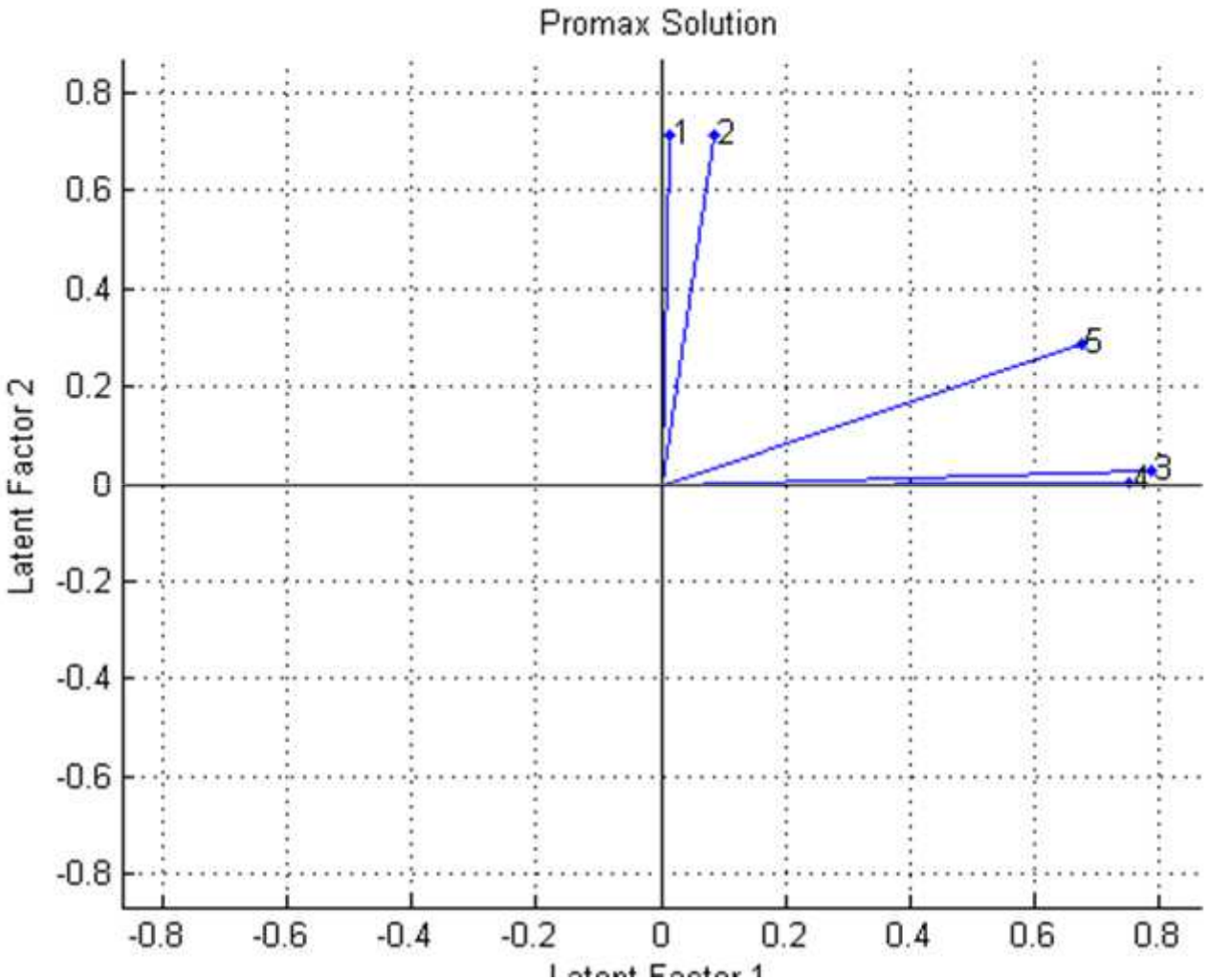
```
>> [LoadingsPM,specVarPM,rotationPM] = ...
factoran(grades,2,'rotate','promax')
LoadingsPM =
    0.0123    0.7104
    0.0848    0.7112
    0.7876    0.0254
    0.7506    0.0044
    0.6759    0.2846
specVarPM =
    0.4829
    0.4031
    0.3512
    0.4321
    0.1944
rotationPM =
    0.6901    0.3882
   -1.2100    1.3378
```

Verificăm faptul că rotaţia promax nu este ortogonală

```
>> rotationPM'*rotationPM
ans =
    1.9405   -1.3509
   -1.3509    1.9405
```

Pentru a vedea efectul rotaţiei **promax** vom desena un biplot având ca axe noii factori comuni obţinuţi în urma rotaţiei:

```
>> biplot(LoadingsPM, 'varlabels',num2str((1:5)'));
>> title('Promax Solution');
>> xlabel('Latent Factor 1'); ylabel('Latent Factor 2');
```


Promax Solution
Latent Factor 2
Latent Factor 1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
2
3
4
5

Promax a făcut o rotaţie a axelor care nu este rigidă şi a realizat o structură mai simplă. Primele doua examene sunt foarte apropiate de cea de-a 2-a axă, examenele 3 şi 4 sunt apropiate de prima axă, examenul interdisciplinar este într-o poziţie intermediară.

Interpretarea factorilor comuni ca abilitate cantitativă, respectiv abilitate calitativă este mai precisă în acest caz.

Pe baza rezultatelor obţinute se pot clasifica noi observaţii.

De exemplu dacă luăm în considerare modelul cu doi factori comuni şi înterpretarea dată de factorii obţinuţi cu rotaţia **promax** putem prognoza ce notă va obţine un student la examenele de literatură.
Datele constau în notele brute obţinute la examene şi astfel factoran returnează predicţii asupra valorii factorilor comuni obţinuţi prin rotaţie pentru fiecare student.

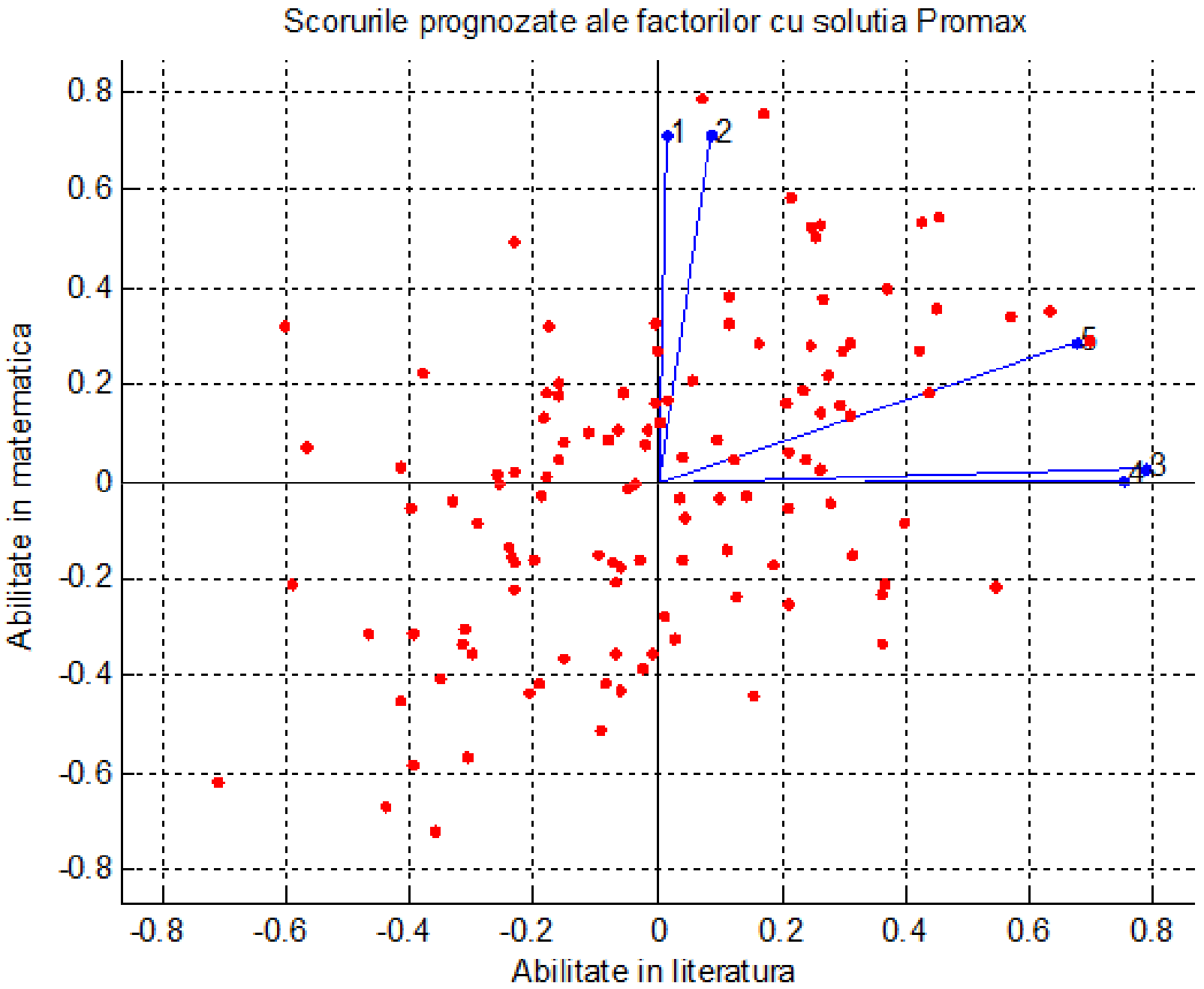
```
>> [Loadings,specVar,rotation,stats,preds] = ...
factoran(grades,2,'rotate','promax','maxit',200)
Loadings =
    0.0123    0.7104
    0.0848    0.7112
    0.7876    0.0254
    0.7506    0.0044
    0.6759    0.2846
specVar =
    0.4829
    0.4031
    0.3512
    0.4321
    0.1944
rotation =
    0.6901    0.3882
   -1.2100    1.3378
stats =
    loglike: -0.0012
        dfe: 1
      chisq: 0.1422
          p: 0.7061
```

```
preds =
-0.9101   -0.6480
   -1.2154   -1.1593
   -0.0265    1.2764
    1.0663    0.8611
   -0.1453   -0.0112
    0.2764    3.0551
   -2.7595   -2.3845
    0.4355    1.4836
.....................................
```

```
>> biplot(Loadings, 'varlabels',num2str((1:5)'), 'Scores',preds)
>> title('Predicted Factor Scores for Promax Solution');
>> xlabel('Ability In Literature'); ylabel('Ability In Mathematics');
```

```
preds =
-0.9101   -0.6480
   -1.2154   -1.1593
   -0.0265    1.2764
    1.0663    0.8611
   -0.1453   -0.0112
    0.2764    3.0551
   -2.7595   -2.3845
    0.4355    1.4836
.....................................
```

```
>> biplot(Loadings, 'varlabels',num2str((1:5)'), 'Scores',preds)
>>  title('Scorurile prognozate ale factorilor cu solutia Promax');
>> xlabel('Abilitate in literatura'); ylabel('Abilitate in matematica');
```


Scorurile prognozate ale factorilor cu solutia Promax
Abilitate in matematica
Abilitate in literatura
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
2
3
4
5

Graficul ilustrează implementarea modelului, prezentând variabilele iniţiale (vectori) şi scorurile prognozate pentru fiecare subiect (puncte).

Se sugerează astfel că dacă unii studenţi au note bune la un anumit tip de examene, dar nu şi la celălalt (al 2-lea şi al 4-lea cadran), marea majoritate au note bune sau note proaste la ambele tipuri de examene (primul şi al 3-lea cadran).

3. Exemplu (Matlab)

Prezentăm baza de date stockreturns, existentă în Matlab, care reprezintă schimbările procentuale în preţurile acţiunilor a 10 companii, schimbări înregistrate timp de 100 de săptămâni. Primele patru companii sunt specializate în tehnologii de vârf, următoarele trei sunt finaciare şi ultimele trei sunt specializate în vânzări cu amănuntul.

Este natural ca preţurile acţiunilor din companiile ce fac parte din acelaşi sector de activitate, să varieze împreună în cazul schimbării condiţiilor economice.

Vrem să demonstrăm cantitativ cu ajutorul analizei factoriale că toate companiile din acelaşi sector au experienţe similare în ceea ce priveşte preţurile acţiunilor, săptămână de săptămână.

Vom încărca baza de date şi, pentru a ne face o părere despre natura datelor, prezentăm observaţiile pe primele patru săptămâni:

```
>>load stockreturns
>> stocks(1:4,:)
ans =
   0.4478    0.0673    0.1503    2.9914   -0.2077    0.4560   - 0.7811
0.2052   -0.0014    1.4646
   0.9881    1.5816   -0.6113    2.6204    0.5358    0.5899    1.2077
1.0658   -0.5516    0.2027
   0.8746    0.7562   -2.1800   -0.3624    1.3719    0.9187    1.1147
1.5450    1.5231    0.3478
   0.7144   -0.6898    0.2538    2.2094    0.5801    1.8370    0.9197
0.4222   -0.4172   -1.0093
```

Suntem interesaţi de un model de analiză factorială cu 3 factori comuni şi o soluţie în care să nu intervină rotaţia, cerinţe pe care le vom specifica când apelăm factoran. Ni se vor returna încărcările estimate ale factorilor într-o matrice în care fiecare linie corespunde uneia dintre cele 10 tipuri de acţiuni şi fiecare coloană corespunde unui factor comun.

```
>> [Loadings,specificVar,T,stats] = ...
  factoran(stocks,3,'rotate','none');
>>Loadings
Loadings =
       0.8885    0.2367   -0.2354
       0.7126    0.3862    0.0034
       0.3351    0.2784   -0.0211
       0.3088    0.1113   -0.1905
       0.6277   -0.6643    0.1478
       0.4726   -0.6383    0.0133
       0.1133   -0.5416    0.0322
       0.6403    0.1669    0.4960
       0.2363    0.5293    0.5770
       0.1105    0.1680    0.5524
```

Interpretarea factorilor în acest caz este destul de dificilă deoarece cele mai multe acţiuni au coeficienţi relativ mari pentru doi sau chiar trei factori comuni. Structura matricei de încărcare s-ar putea simplifica în urma rotaţiei factorilor şi, astfel, am putea să avem o interpretare mai bună a acestora.

```
>>specificVar
specificVar =
    0.0991
    0.3431
    0.8097
    0.8559
    0.1429
    0.3691
    0.6928
    0.3162
    0.3311
    0.6544
>>stats.p
ans =
    0.8144
```

Din estimarea dispersiilor specifice, observăm că în cazul preţului unor anumite acţiuni dispersia este apropiată de 1, ceea ce înseamnă că în această variabilă nu intervine factorul comun. În cazul preţului primei acţiunii respectiv preţului celei de-a 5-a acţiuni, dispersia specifică este apropiată de zero, ceea ce arată că variabila respectivă este determinată de factorul comun.
Nivelul de semnificaţie p returnat de stats nu respinge ipoteza nulă corespunzătoare celor 3 factori comuni.

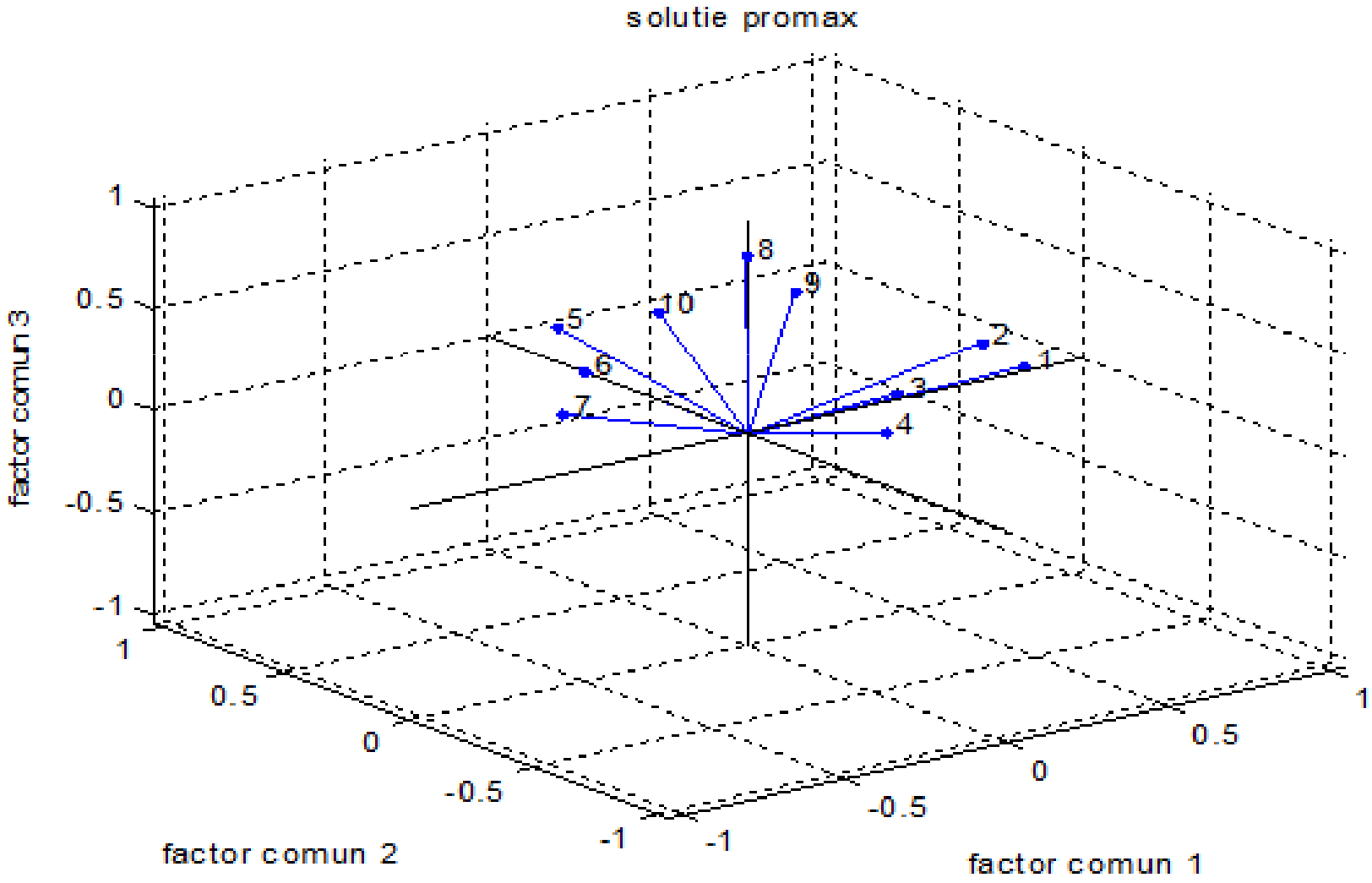
Suntem interesaţi acum de a găsi o parametrizare în care fiecare variabilă să aibă un număr mic de încărcari cu valoare mare, ceea ce ar însemna că fiecare variabilă este influenţată de un număr mic de factori comuni, de preferinţă unul singur, scop în care vom utiliza rotaţia factorilor.

Fiecare linie din matricea încărcărilor poate fi considerată ca reprezentând coordonatele unui punct din spaţiul tri-dimensional, situaţie în care fiecare factor comun ar corespunde unei axe de coordonate, rotaţia factorilor fiind echivalentă rotaţiei acestor axe. Vom folosi rotaţia *promax,* care este cea mai utilizată metodă de rotaţie oblică:

```
>> [LoadingsPM,specVarPM]= factoran(stocks,3,'rotate','promax');
>>LoadingsPM
LoadingsPM =
    0.9452    0.1214   -0.0617
    0.7064   -0.0178    0.2058
    0.3885   -0.0994    0.0975
    0.4162   -0.0148   -0.1298
    0.1021    0.9019    0.0768
    0.0873    0.7709   -0.0821
   -0.1616    0.5320   -0.0888
    0.2169    0.2844    0.6635
    0.0016   -0.1881    0.7849
   -0.2289    0.0636    0.6475
```

Într-adevăr, am creat o structură mai simplă a încărcarii factorilor, marea majoritate a acţiunilor având o încărcare mare doar pe un factor comun. Utilizând factorii comuni drept axe de coordonate, vizualizăm fiecare acţiune, folosind biplot:

```
>>biplot(LoadingsPM,'varlabels',num2str((1:10)'));
```

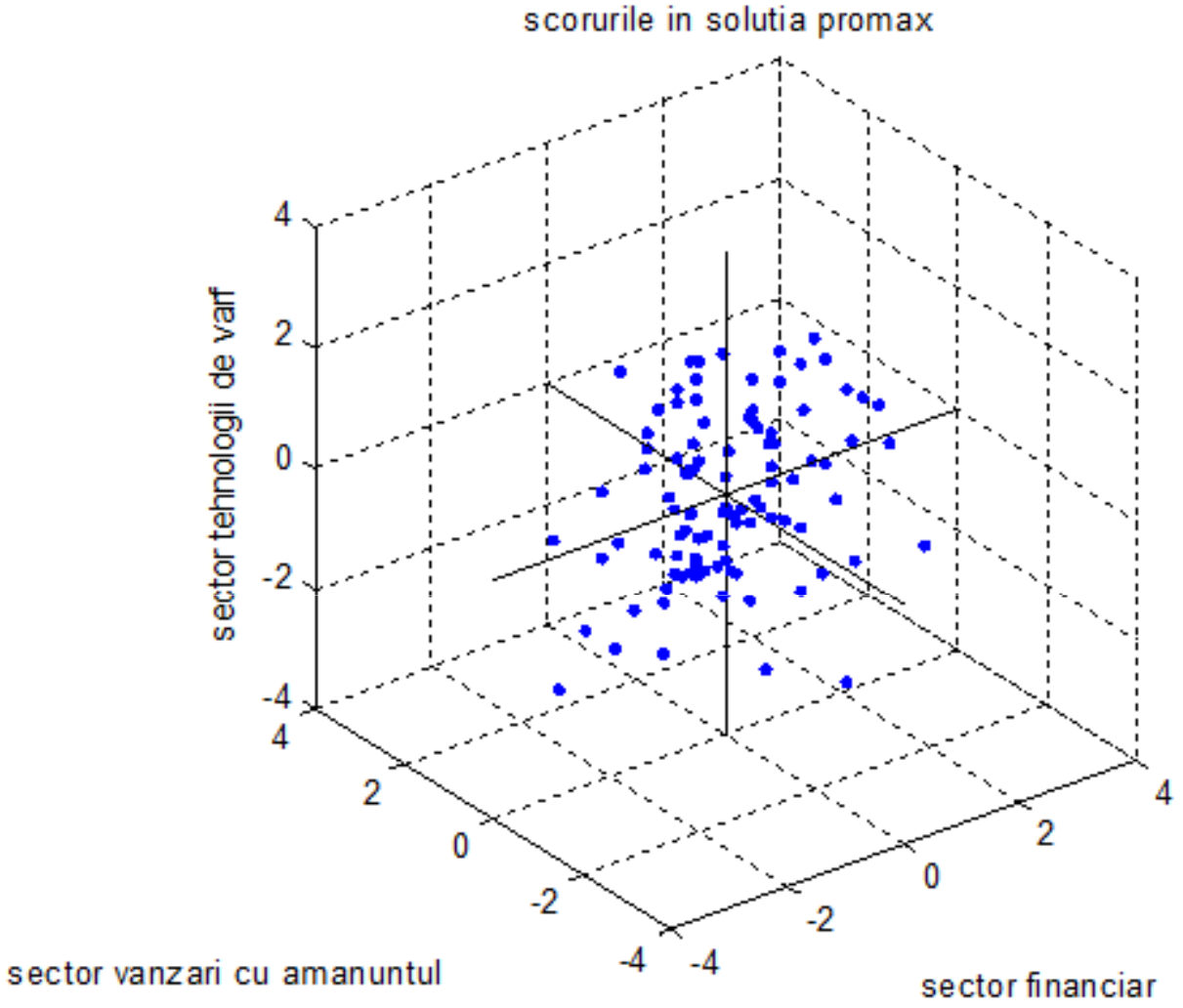

solutie promax
factor comun 3
1
0.5
0
-0.5
-1
factor comun 2
1
0.5
0
-0.5
-1
factor comun 1
1
0.5
0
-0.5
-1
1
2
3
4
5
6
7
8
9
10

Fiecare acţiune depinde de un singur factor comun şi, astfel, putem descrie fiecare factor în funcţie de acţiunile pe care le afectează. Observăm acţiunile căror companii sunt apropiate de anumite axe şi, astfel, putem spune că axa primului factor reprezintă sectorul financiar, a doua corespunde sectorul vânzărilor cu amănuntul şi a treia reprezintă sectorul tehnologiilor de vârf.

Dacă acceptăm modelul cu trei factori şi interpretarea factorilor obţinuţi prin rotaţie, ne propunem acum să clasificăm fiecare săptămână în funcţie de cât de favorabilă a fost celor trei sectoare de acţiuni, bazându-ne pe datele obţinute. Funcţia factoran poate să returneze estimările valorii fiecăruia dintre cei 3 factori comuni obținut în urma rotației pe fiecare saptămână.

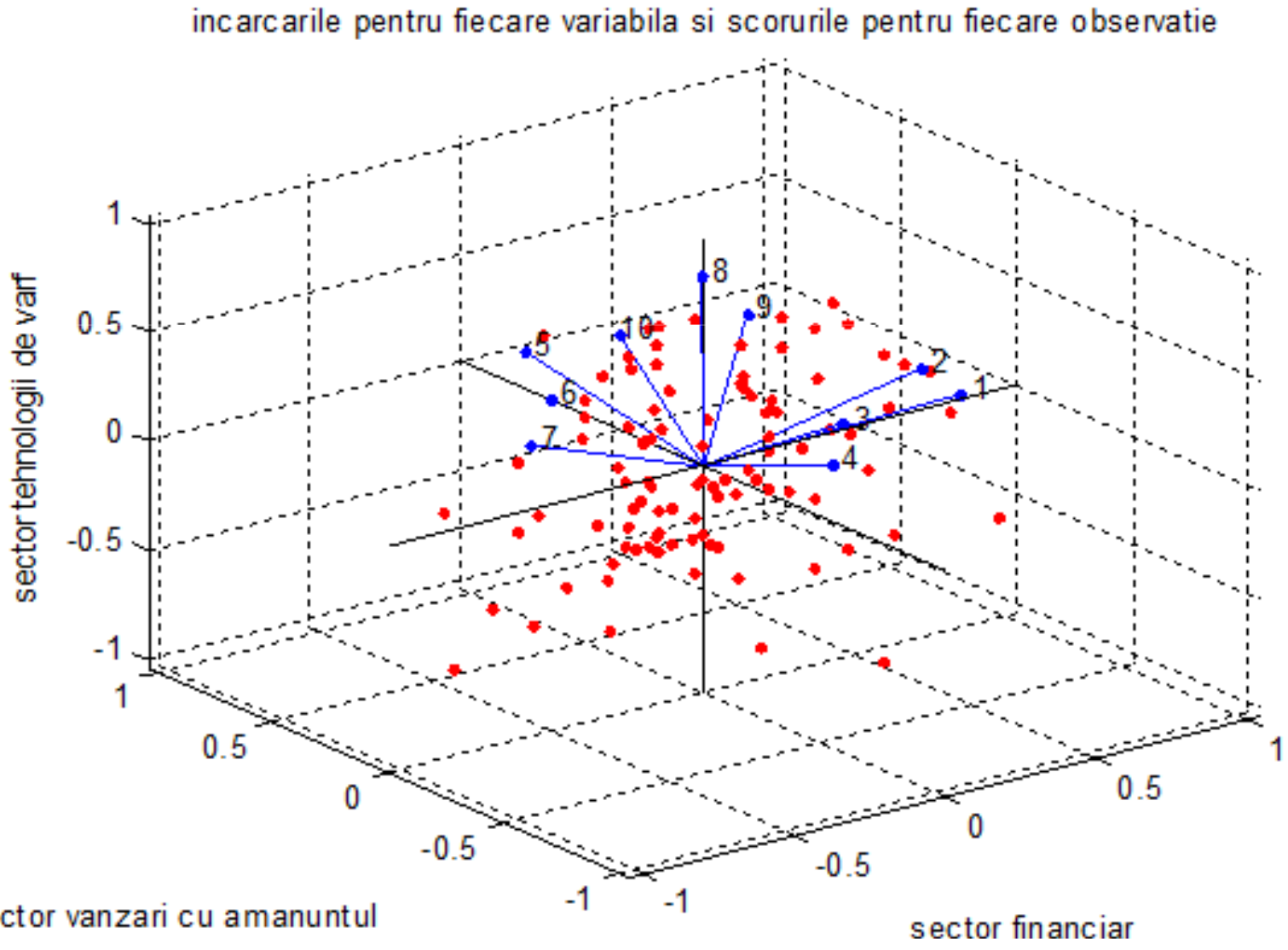
Putem desena scorurile estimate pentru a vedea, în fiecare saptămână, cum sunt afectate acțiunile din fiecare sector de activitate:

```
>> [LoadingsPM,specVarPM,TPM,stats,F] = ...
  factoran(stocks, 3,'rotate','promax');
>>plot3(F(:,1),F(:,2),F(:,3),'b.')
>>line([-4 4 NaN 0 0 NaN 0 0], [0 0 NaN -4 4 NaN 0 0],...
     [0 0 NaN 0 0 NaN -4 4], 'Color','black')
>>grid on
>>axis square
```


scorurile in solutia promax
sector tehnologii de varf
4
2
0
-2
-4
sector vanzari cu amanuntul
sector financiar

Vom vizualiza în acelaşi sistem de coordonate atât încărcările pentru fiecare variabilă cât şi scorurile pentru fiecare observaţie. Fiind un model, cu 3 factori comuni vizualizarea prin biplot este tridimensională

```
>>biplot(LoadingsPM,'scores',F,'varlabels',num2str((1:10)'))
```

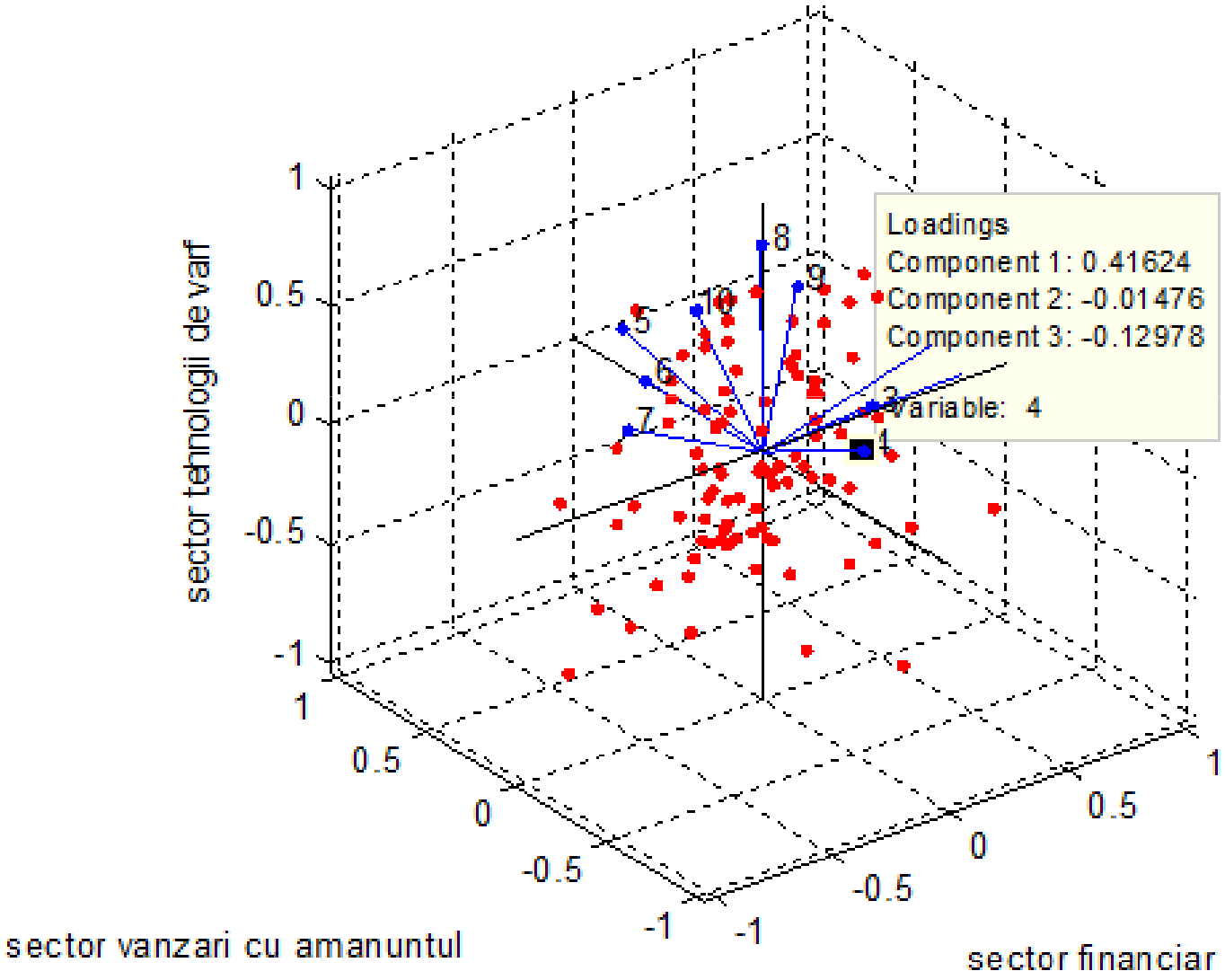


Fiecare dintre cele 10 acţiuni este prezentată printr-un vector, direcţia şi norma acestuia arătând modul său de dependenţă de factorii comuni în urma rotaţiei *promax*:

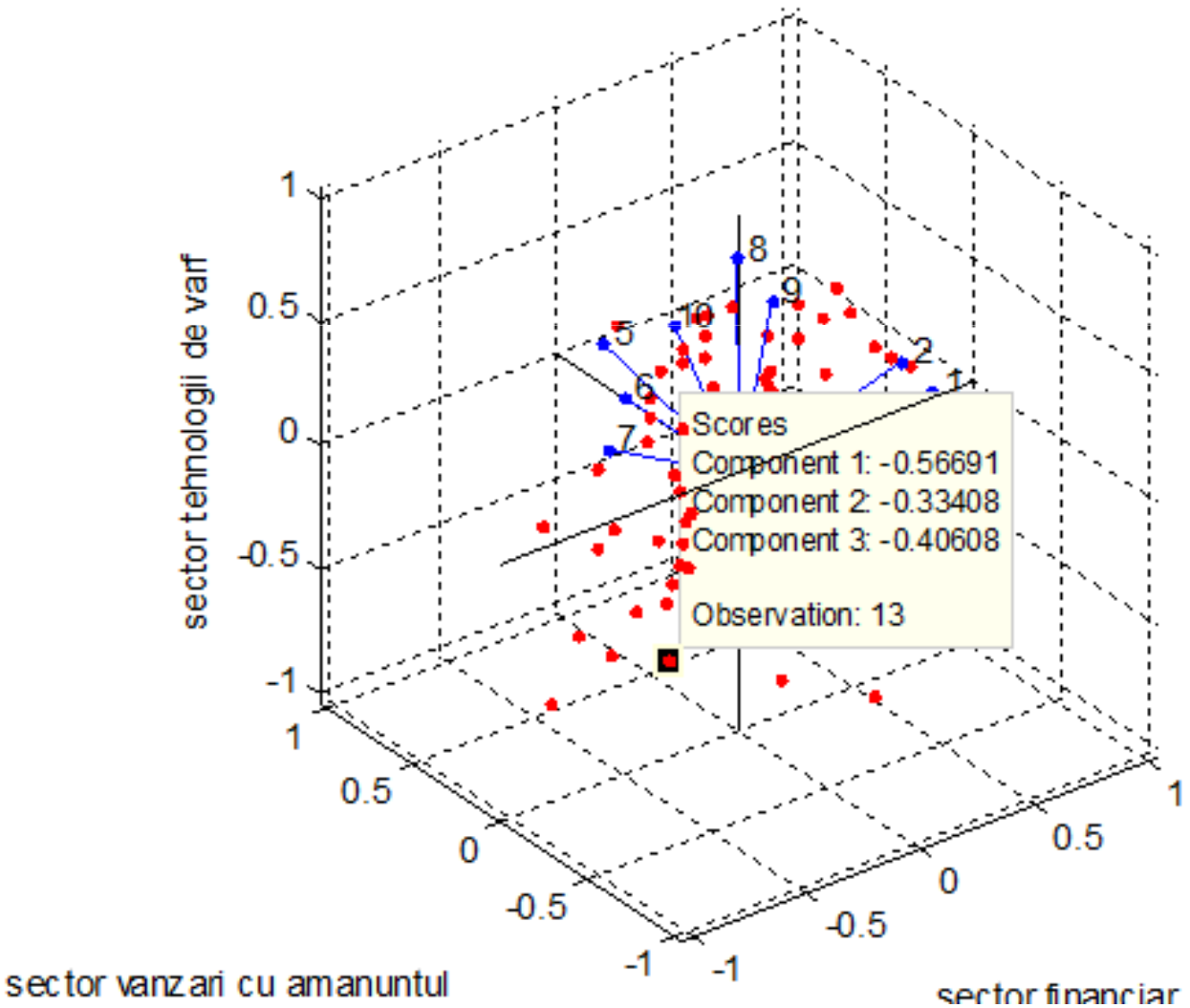
- Dependenţa acţiunilor companiilor 1,2,3,4 de primul factor corespunde celor patru vectori direcţionaţi aproximativ de-alungul axei de coordonate, constituită de primul factor, care este interpretat ca fiind efectul sectorului tehnologiilor de vârf.
- Dependenţa acţiunilor companiilor 5,6,7 de al doilea factor, care este interpretat ca efectul sectorului vânzarilor cu amănuntul, corespunde vectorilor ce au aproximativ direcţia axei corespunzătoare celui de al doilea factor comun.
- Analog dependenţa acţiunilor companiilor 8,9,10 de cel de-al treilea factor comun este prezentată de vectori ce au aproximativ direcţia axei corespunzătoare celui de al treilea factor comun.

Fiecare dintre cele 100 de observaţii este reprezentată de un punct şi locul unde se afla acesta indică scorul fiecărei observaţii pentru cei 3 factori comuni. De *exemplu* punctele ce se află în vârful acestui grafic au cele mai mari scoruri pentru factorul corespunzător tehnologiilor de vârf.
Folosind *Data Cursor* din Tools, putem afla amănunte despre elementele acestei reprezentări grafice:

- Dând click pe un vector (acțiune) putem afla încărcările pentru fiecare factor.



- Dând click pe un punct (observaţie), putem citi scorurile acelei observaţii pentru fiecare factor.



Analiza componentelor principale

Analiza componentelor principale (*Principal components analysis* -PCA) poate fi privită ca o tehnică de analiză factorială, atunci când este luată în considerație dispersia totală a datelor.

În principiu, analiza componentelor principale are ca scop reducerea numărului de variabile utilizate inițial, luând în considerație un număr mai mic de variabile *reprezentative* și necorelate. Se obține astfel o clasificare a variabilelor și cazurilor.

Să considerăm că dorim să cumpărăm un anumit produs dintr-un magazin şi, pentru început, suntem interesaţi doar de două caracteristici ale sale, A şi B.
În acest caz putem considera *norul* de împrăştiere al punctelor generate de perechile de date corespunzătoare celor două atribute, după care vom considera dreapta care traversează centrul *norului* de puncte (în particular, centroidul *norului*), deci dreapta lor de regresie care este reprezentativă pentru cele două atribute.

Să presupunem acum că luăm în consideraţie încă o caracteristică a produsului, notată C. Dacă în acest caz vom lua în consideraţie doar perechile de regresii între cele trei atribute, nu obţinem o alegere satisfăcătoare, deoarece nu avem o imagine de ansamblu asupra tuturor celor trei atribute.
Avem nevoie de ceva care să *însumeze* toate cele trei atribute deodată.

Problema se complică şi mai mult dacă luăm în consideraţie un număr şi mai mare de atribute, în acest caz având nevoie, în loc de perechi de regresii, de un *scor* care să caracterizeze obiectul respectiv.

Din punct de vedere geometric, acest scor poate fi generat de o dreaptă sau drepte (axele factor) care să treacă prin centroidul *norului* de puncte generat de *tuplele* de date.
Astfel, plecând de la spaţiul iniţial al datelor, se consideră un subspaţiu generat de un set de axe noi, numite *axe factor* (*factor axes*), subspaţiu în care este proiectat spaţiul iniţial.

În principiu, tehnica PCA caută dreapta care se *potriveşte* cel mai bine *norului* de puncte din spaţiul vectorial al instanţelor şi atributelor. Matematic vorbind, considerând p atribute şi q instanţe, tehnica PCA de identificarea a factorilor se referă la diagonalizarea matricei simetrice reprezentând matricea corelaţiilor (matricea covarianţelor).

Reamintim că, deoarece covarianţa se calculează doar pentru perechi de variabile statistice, în cazul a trei variabile X, Y şi Z, matricea covarianţelor este dată de:

$$Cov(X,Y,Z) = \begin{pmatrix} \text{cov}(X,X) & \text{cov}(X,Y) & \text{cov}(X,Z) \\ \text{cov}(Y,X) & \text{cov}(Y,Y) & \text{cov}(Y,Z) \\ \text{cov}(Z,X) & \text{cov}(Z,Y) & \text{cov}(Z,Z) \end{pmatrix},$$

pentru cazul a n variabile se va proceda analog.

În cazul de față, dacă matricea standardizată (adică centrată în raport cu mediile respective) X reprezintă datele corespunzătoare celor q instanțe și p atribute, atunci $X^t \cdot X$ reprezintă matricea covarianțelor și problema care se pune este diagonalizarea acesteia.

Rezultatul va fi un nou set de variabile – *componentele principale* – care sunt combinații liniare ale atributelor inițiale și sunt necorelate.
Se obține astfel un spațiu de dimensiune mai mică, în care se proiectează instanțele și atributele și care păstrează maximum din variabilitatea datelor.
Schematic, PCA poate fi rezumată în următorii pași:

- ⇨ identificarea vectorilor proprii ai matricei covarianțelor;
- ⇨ construirea noului spațiu generat de vectorii proprii.

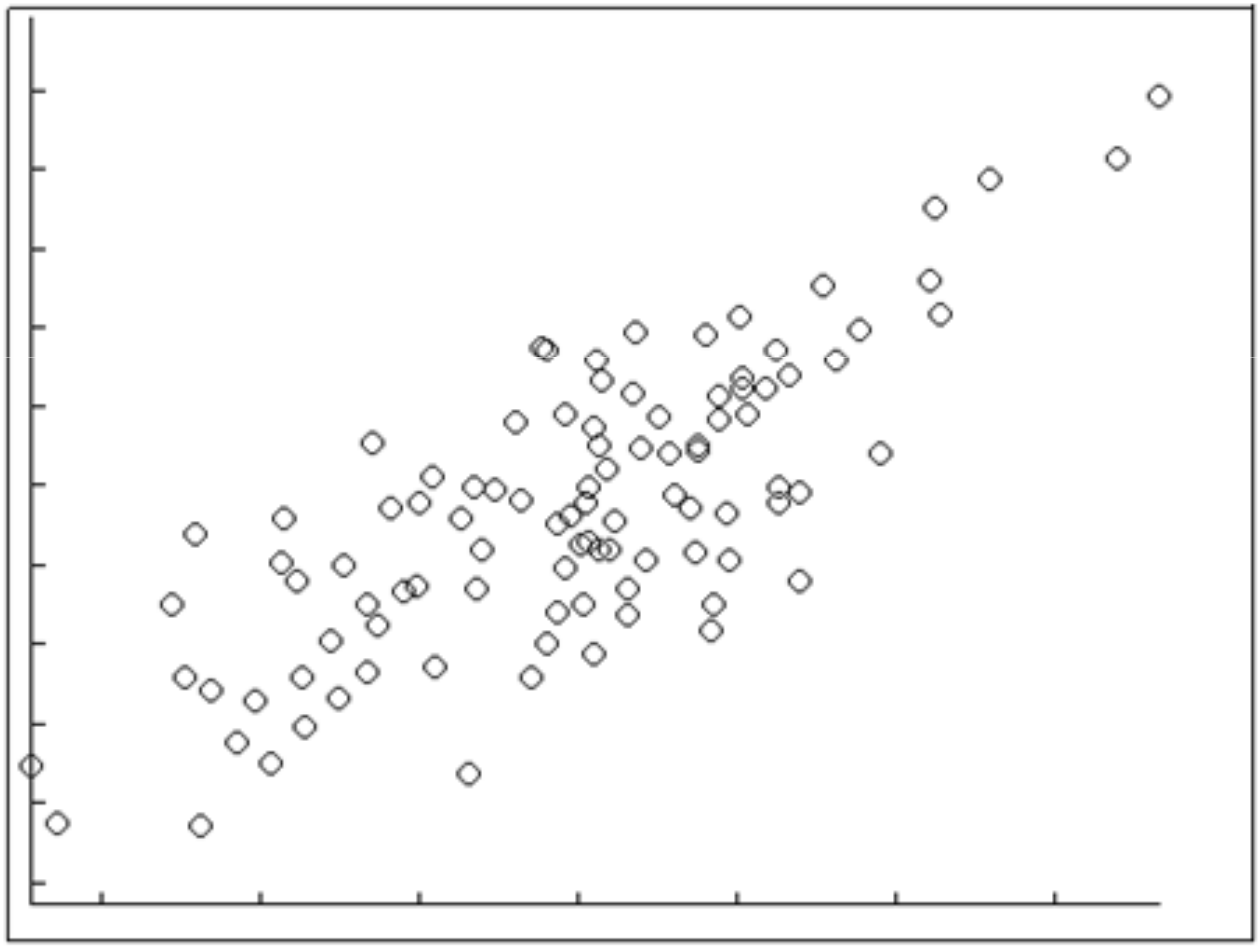
Vom ilustra schematic, în figurile următoare, paşii unei analize PCA.

- prima componenta principală reprezintă combinaţia variabilelor care *explică* cea mai mare dispersie a datelor.
- a doua componentă principală *explică* următoarea dispersie maximă a datelor, fiind independentă de prima

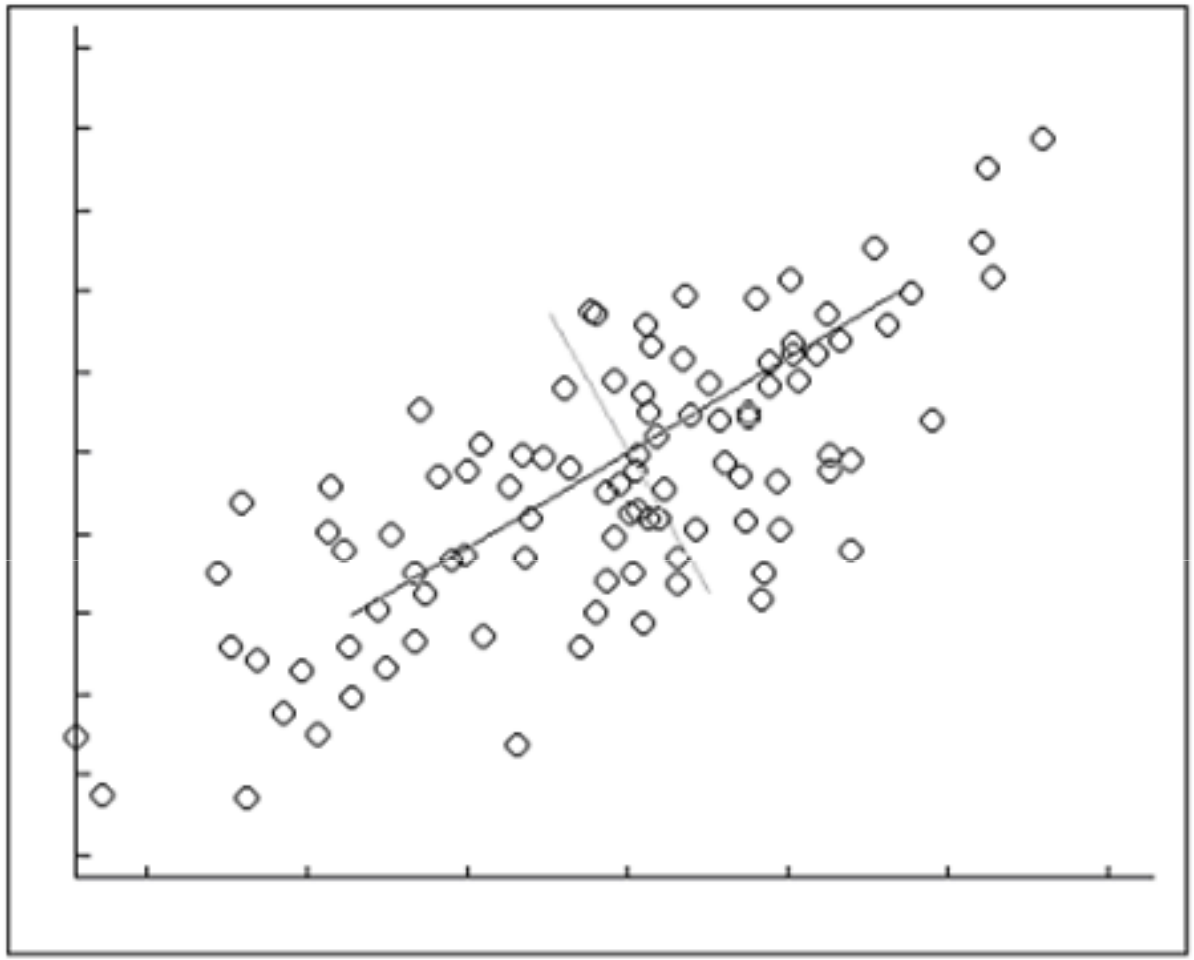
ş.a.m.d.,

Putem considera, în principiu, atâtea componente principale câte variabile există.

Prezentăm setul de date sub forma *norului* lor de împrăştiere.

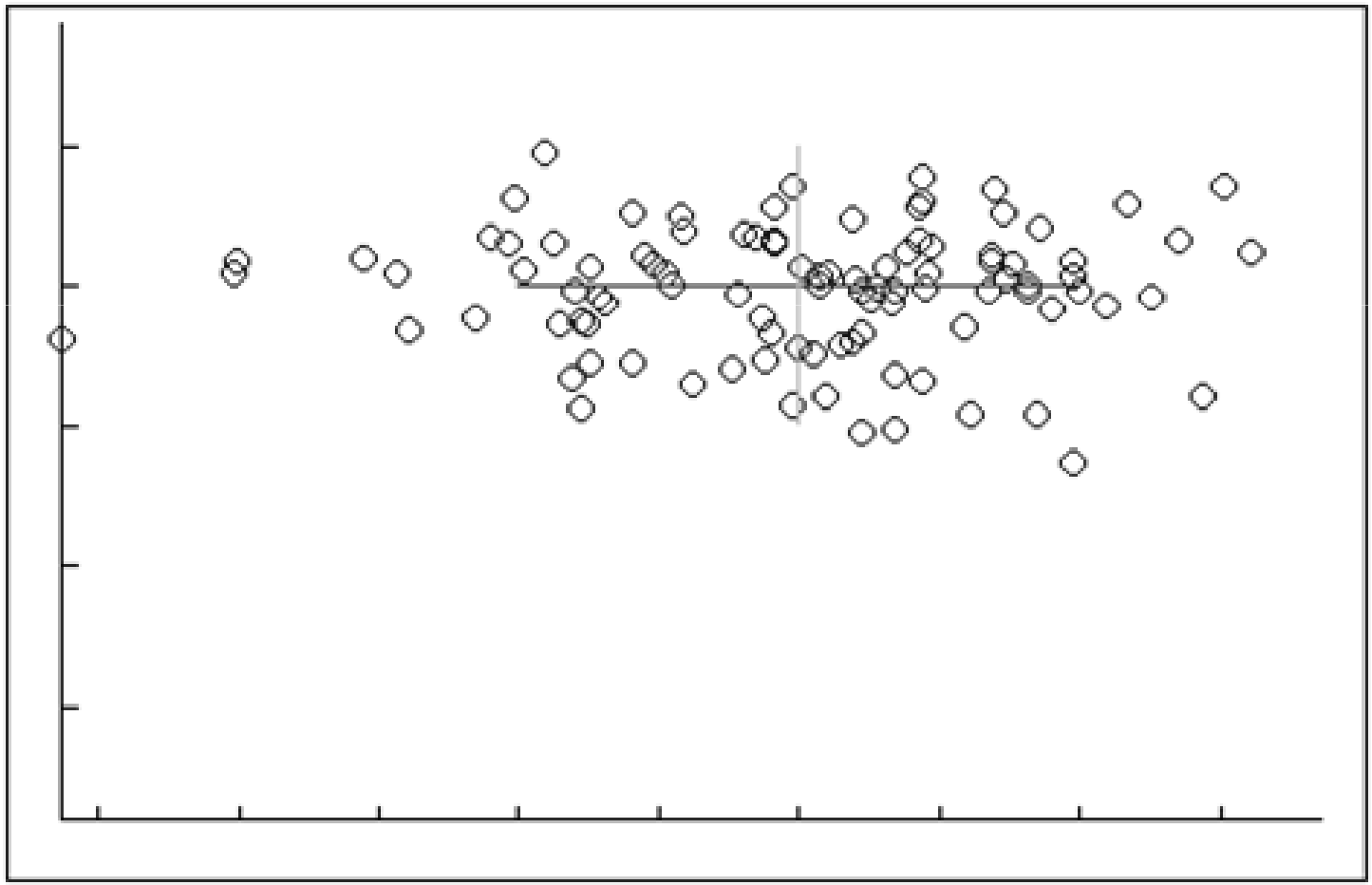


După calcularea matricei covarianţelor şi a celor două componente principale, figura următoare ilustrează noua configuraţie a spaţiului datelor.



Dreapta oblică (cea pronunţată), care trece prin centrul *norului* punctelor, explicând cea mai mare dispersie, reprezintă prima componentă principală. A doua componentă principală este perpendiculară pe prima (independentă de aceasta) şi explică restul de dispersie.

În final, prin înmulțirea datelor inițiale cu componentele principale, datele vor fi rotite, astfel încât componentele principale formează axele noului spațiu, așa cum se observă în figura de mai jos.



În statistica multivariată, una dintre cele mai dificile probleme constă în

În statistica multivariată, una dintre cele mai dificile probleme constă în
vizualizarea datelor de mai multe variabile.

In Matlab funcţia **plot** descrie graficul relaţiei existente între două variabile, în timp ce funcţiile **plot3** şi **surf** sunt folosite pentru spaţiul tridimensional.

In mulţimile de date cu mai multe variabile, uneori grupuri de caracteristici variază împreună. Astfel măsurând o singură variabilă din grup, care este considerată forţa motrice a sistemului, ştim că întregul sistem are aceeaşi variaţie.

Dacă putem măsura toate caracteristicile, ne putem da seama care sunt redundante, putând simplifica astefl problema înlocuind grupul de caracteristici cu una singură, considerată o reprezentatntă a grupului.

Analiza componentelor principale este metoda ce face această simplificare, generând o nouă mulțime de variabile, numite componente principale. Fiecare componentă principală este o combinație liniară de variabilele inițiale.

Componentele proincipale sunt ortogonale doua câte două, formând o bază ortogonală a spațiului datelor.

Prima componentă principală este o axă unică în spațiu. Valorile obținute prin proiectarea fiecărei observații pe această axă formează o nouă variabilă (caracteristică), a cărei dispersie este cea mai mare dintre toate alegerile posibile a acestei prime axe.

A doua componentă principală este o altă axă a spațiului, perpendiculară pe prima. Proiectarea observațiilor pe această axă generează o nouă variabilă a cărei dispersie este cea mai mare dintre toate alegerile posibile a acestei a doua axe.

Mulţimea completă a componentelor principale este la fel de mare ca mulţime iniţială. Dacă suma dispersiilor primelor componente principale reprezintă 80% din dispersia totală a datelor iniţiale, ne rezumăm la aceste prime componente principale.

Prin examinarea graficelor acestor noi variabile, se înţelege mai bine care sunt forţele motrice generate de datele iniţiale.

In Matlab funcţia princomp returnează componentele principale, în cazul în care avem valorile datelor iniţiale

Daca avem matricea corelaţiilor datelor sau covarianţelor funcţia pcacov va face analiza componentelor principale.

4. exemplu

Considerăm baza de date ce prezintă 9 indici diferiţi ai calităţii vieţii în 329 oraşe din SUA: clima, situaţia locativă, sănătate, criminalitate, transport, educaţie, artă, recreere şi economie.

Este bine ca indicele să fie cât mai mare. De exemplu dacă criminalitatea are un indice mare, această înseamnă ca rata criminalităţii este mică.

Vom încărca baza de date şi vom da comanda whos, care generează un tabel cu informaţii despre variabilele din bază:

```
>> load cities
>> whos
  Name          Size         Bytes  Class
  categories     9x14          252  char array
  names        329x43        28294  char array
  ratings       329x9        23688  double array
```

Această bază de date conține trei variabile:

- **categories**, un vector coloană ce conține numele indicilor.
- **names**, un vector coloană ce conține numele clor 329 de orașe.
- **Ratings** (evaluări), o matrice cu 329 linii și 9 coloane.

Variabila **categories** are următoarele valori:

```
>>categories
categories =
   climate
   housing
   health
   crime
   transportation
   education
   arts
   recreation
   economics
```

Primele 10 componente ale lui **names** sunt:

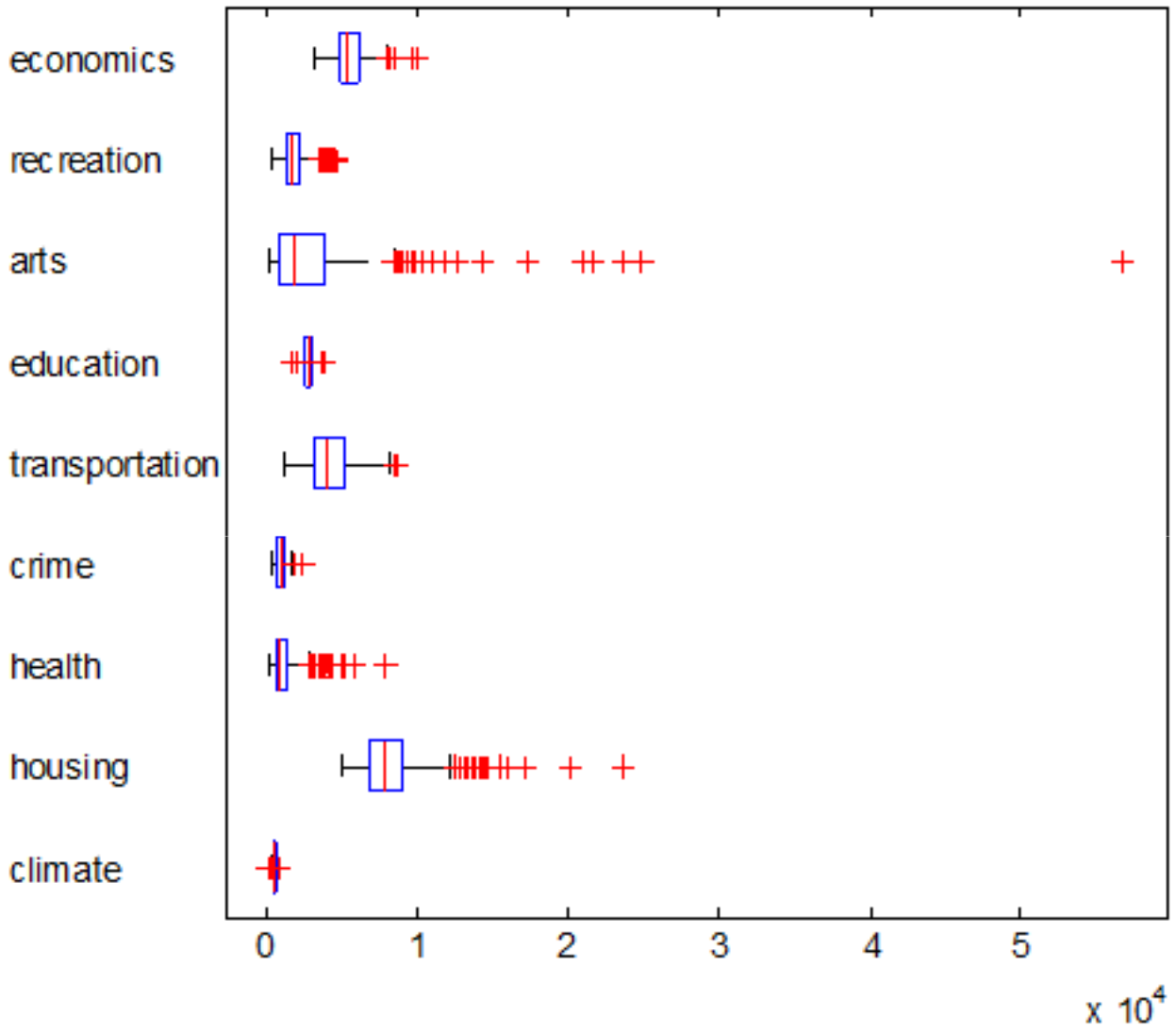
```
>> first10= names(1:10,:)
first10 =
Abilene, TX
Akron, OH
Albany, GA
Albany-Troy, NY
Albuquerque, NM
Alexandria, LA
Allentown,Bethlehem, PA-NJ
Alton, Granite City, IL
Altoona, PA
Amarillo, TX
```

Pentru o primă impresie asupra datelor vom face o diagramă de tip **boxplot**:

boxplot(X) desenează o diagramă pentru fiecare coloană a matricei X, diagramă ce oferă informații privind tendința centrală și forma distribuției studiate.

O diagramă de tip boxplot reflectă grafic rezumarea prin cele 5 valori ale unei distribuții: valoarea minimă, prima quartila, mediana, a treia quartila și valoarea maximă. Va prezenta deasemeni și valorile aberante (outliers) .

```
>>boxplot(ratings,'orientation','horizontal','labels',categories)
```



Din desen se observă că în evaluările privitoare la artă şi situaţia locativă este
mai mare variabilitate decât în cazul climei sau ratei criminalităţii.

De multe ori, dacă variabilele au aceeaşi unitate de măsură calculăm componentele principale din datele brute.

Dacă variabilele au unităţi diferite de măsură sau dispersia diferitelor coloane este semnificativă (ca în acest caz) este necesară *standardizarea* datelor.

Putem standardiza datele împărţind fiecare coloană prin deviaţia sa standard.

```
>> stdr = std(ratings)
stdr =
  1.0e+003 *
    0.1208    2.3853    1.0030    0.3572    1.4512    0.3208    4.6423    0.8079    1.0845
```

B = repmat(A,m,n) crează o matrice ce are $m \times n$ celule în care se află copii ale matricei A .

```
>> sr = ratings./repmat(stdr,329,1)
sr =
   4.3126   2.5993   0.2363   2.5843   2.7777   8.5943   0.2145   1.7391   7.0385
   4.7596   3.4118   1.6510   2.4807   3.3648   7.5999   1.1985   3.2579   4.0112
   3.8739   3.0768   0.6162   2.7159   1.7441   7.9802   0.0511   1.0633   4.8411
   3.9401   3.3154   1.4267   1.7079   4.7430  10.5956   1.0027   2.0015   5.4073
   5.4549   3.5187   1.8475   4.1523   4.5191   9.4329   0.9685   3.2331   5.2809
   4.3043   2.4396   0.6381   2.0355   1.6841   9.2645   0.0719   1.2601   4.8448
   4.6272   3.4747   0.6191   1.4392   1.9853   9.8007   0.5026   1.3826   4.7000
.....................................
```

Acum putem găsi componentele principale, apelând funcția princomp:

```
>> [coefs,scores,variances,t2] = princomp(sr)
```

coefs reprezinţa coeficienţii combinaţiilor liniare formate din variabilele (caracteristicile) de bază, care generează componentele principale, coeficienţi ce se întâlnesc şi sub denumirea de *încărcări*.

Coeficfienţii vectorilor ce constituie primele trei componente principale sunt:

```
c3 = coefs(:,1:3)
c3 =
    0.2064    0.2178   -0.6900
    0.3565    0.2506   -0.2082
    0.4602   -0.2995   -0.0073
    0.2813    0.3553    0.1851
    0.3512   -0.1796    0.1464
    0.2753   -0.4834    0.2297
    0.4631   -0.1948   -0.0265
    0.3279    0.3845   -0.0509
    0.1354    0.4713    0.6073
```

In prima componentă principală, căreia îi corespunde prima coloană, cei mai mari coeficienţi corespund caracteristicilor sănătate şi arte. Deoarece toți coeficienţii din prima coloană au acelaşi semn putem considera că este o medie ponderată a caracteristicilor iniţiale.

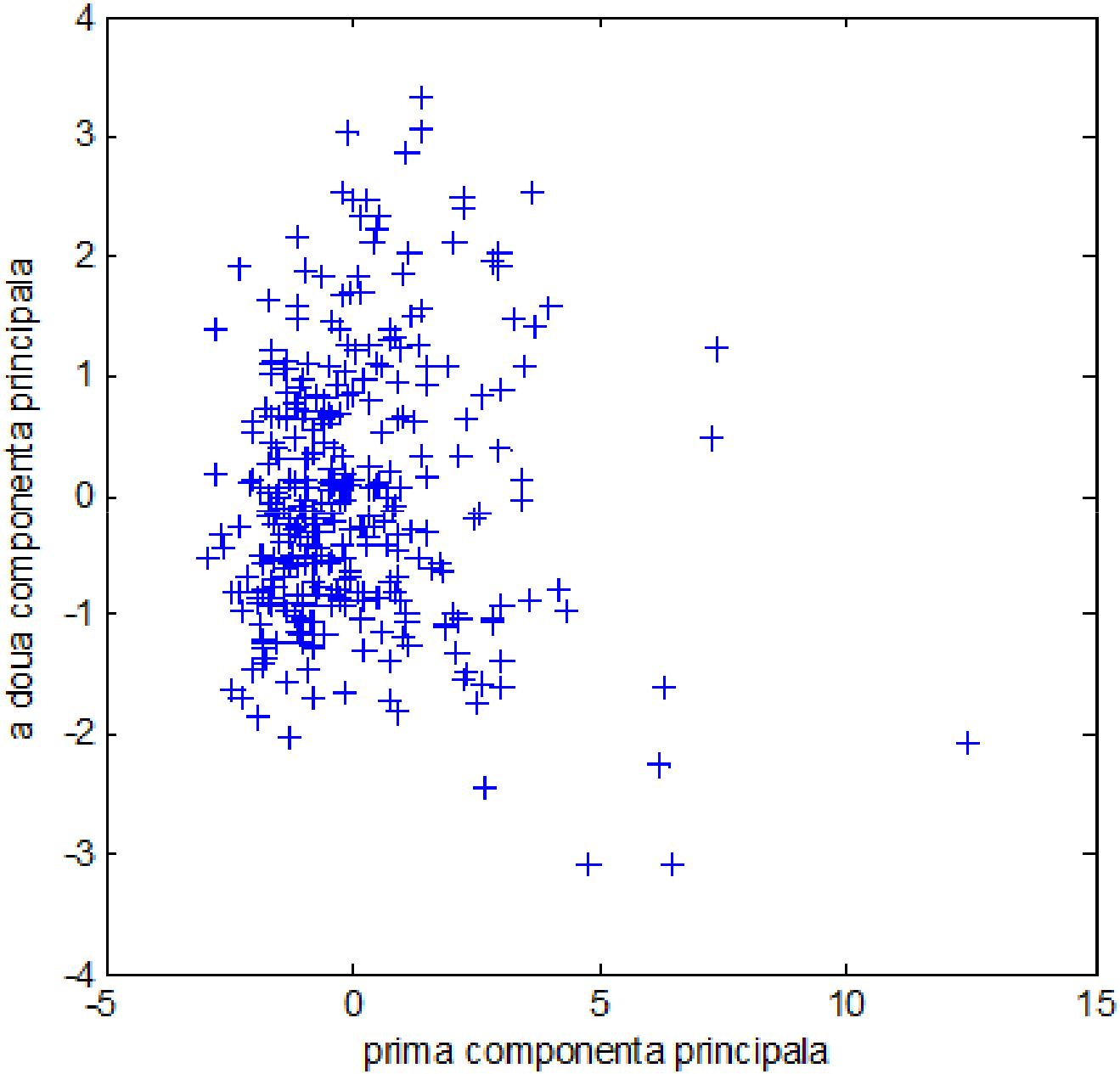
Componentele principale au norma egală cu unitatea și sunt ortogonale.

```
I = c3'*c3
I =
    1.0000   -0.0000   -0.0000
   -0.0000    1.0000   -0.0000
   -0.0000   -0.0000    1.0000
```

scores conţine conţine coordonatele datelor iniţiale în noul sistem de coordonate, definit de componentele principale.

Un grafic al primelor două coloane din **scores** arată datele evaluate proiectate pe primele două componente

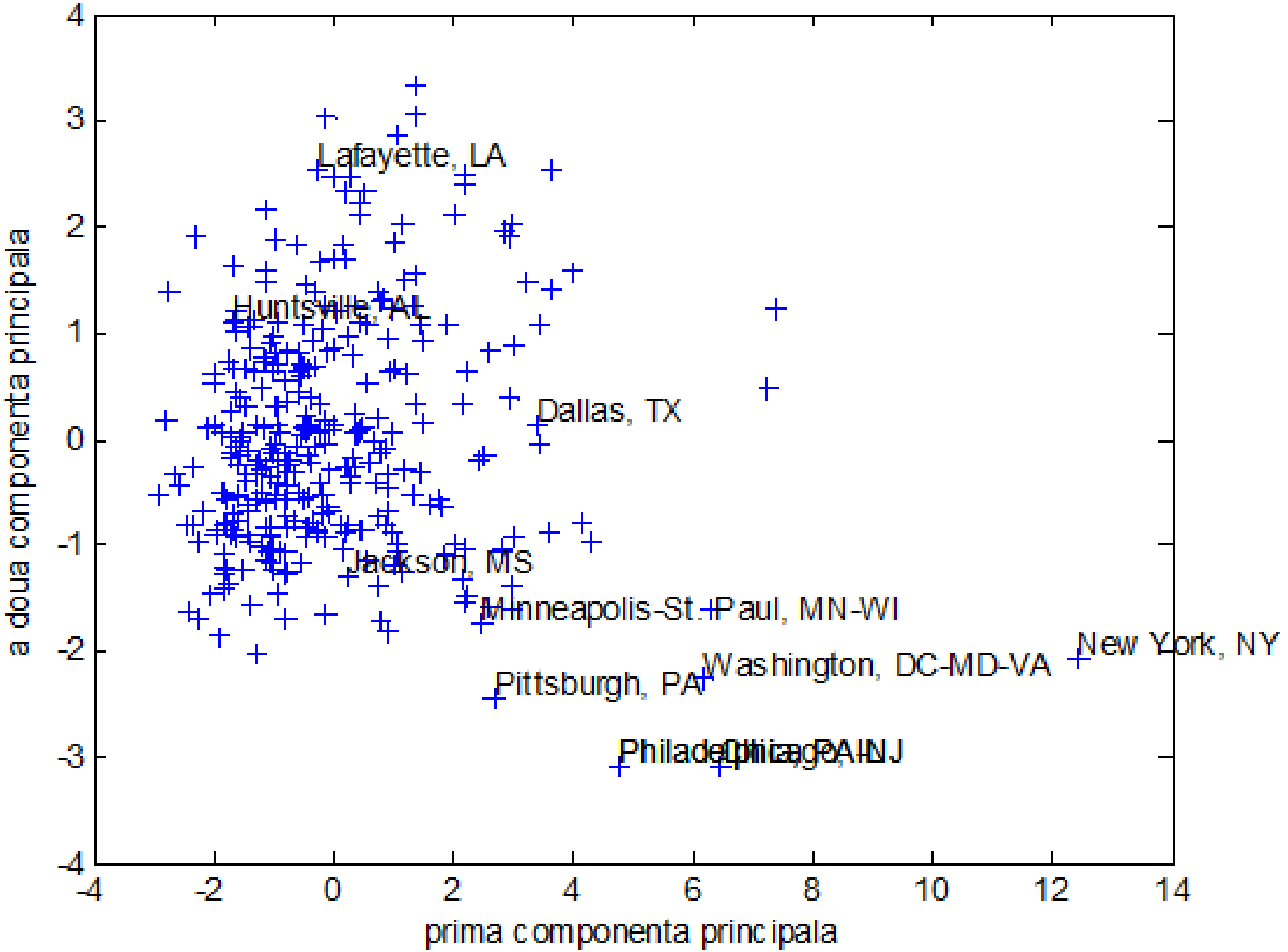
```
>> plot(scores(:,1),scores(:,2),'+')
>> xlabel('prima componenta principala')
>> ylabel('a doua componenta principala')
```


4
3
2
1
0
-1
-2
-3
-4
-5
0
5
10
15
a doua componenta principala
prima componenta principala

Să observăm punctele aberante(outliers)

Este posibil a desena un grafic tridimensional utilizând cele trei coloane din **scores**, dar este preferabilă crearea graficelor bidimensionale, care sunt mai uşor de înţeles.

Funcţia **gname** este folosită pentru a identifica puncte din graficul prezentat mai sus. De exemplu **gname(names)** va arăta numele oraşelor prin mişcarea cursorului şi dând click pe fiecare punct.


Lafayette, LA
Huntsville, AL
Dallas, TX
Jackson, MS
Minneapolis-St. Paul, MN-WI
New York, NY
Washington, DC-MD-VA
Pittsburgh, PA
Philadelphia, PA-NJ
a doua componenta principala
prima componenta principala
4
3
2
1
0
-1
-2
-3
-4
-4
-2
0
2
4
6
8
10
12
14

variances este un vector ce conţine dispersiile, fiecare coloană din **scores** are dispersia egală cu elementul corespiunzător din **variances**.

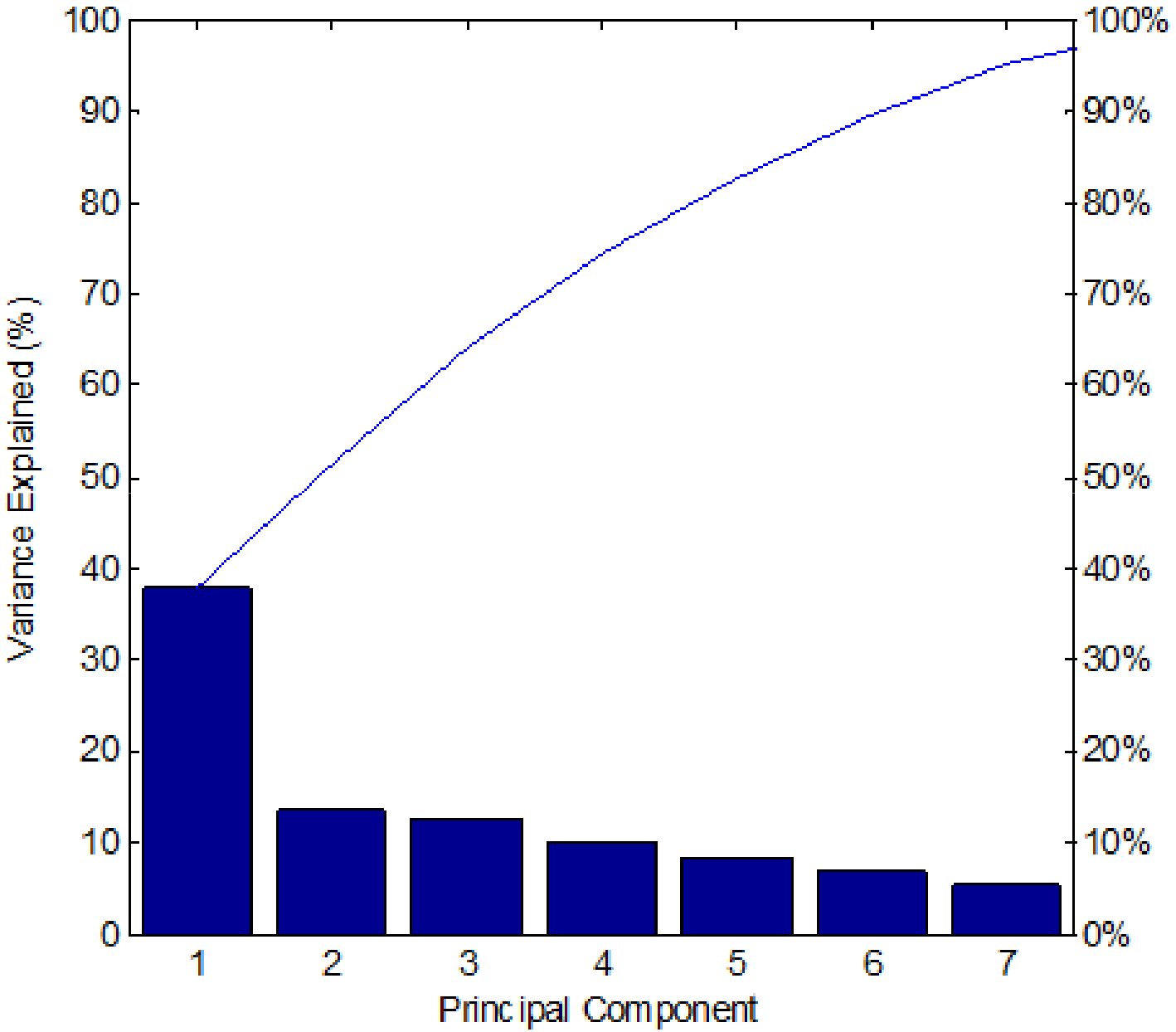
```
variances
variances =
    3.4083
    1.2140
    1.1415
    0.9209
    0.7533
    0.6306
    0.4930
    0.3180
```

Putem calcula procentajul variabilității totale, explicată de fiecare componentă principală.

```
percent_explained = 100*variances/sum(variances)
percent_explained =
   37.8699
   13.4886
   12.6831
   10.2324
    8.3698
    7.0062
    5.4783
    3.5338
    1.3378
```

Utilizând funcţia **pareto** vom obţine un grafic cu variabilitatea în procente a fiecărei componente principale.

```
>>pareto(percent_explained)
>>xlabel('Principal Component')
>>ylabel('Variance Explained (%)')
```


100
90
80
70
60
50
40
30
20
10
0
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Variance Explained (%)
1
2
3
4
5
6
7
Principal Component

Intre prima şi a doua componentă există o mare diferența de variabilitate. Totuşi prima componentă explică mai puțin de 40% din dispersie, ceea ce înseamnă că e nevoie să luăm în considerare mai multe componente principale.

Observăm că primele trei componente principale explică două treimi din variabilitatea totală, așă că înaite de vizualizarea datelor este bine să reducem dimensiunea.

t2 este notația pentru Hotelling's T^2, o măsură statistică a distanței multivariate dintre fiecare observație și centroidul mulțimii datelor.

Este o manieră analitică de a găsi punctele extreme din baza de date.

```
[st2, index] = sort(t2,'descend'); % sortare in ordine descrescatoare
extreme = index(1)
extreme =
   213
names(extreme,:)
ans =
   New York, NY
```

Nu este surprinzător că rezultatele pentru New York sunt cele mai îndepărtate de medie.

t2 este notaţia pentru Hotelling's T^2, o măsură statistică a distanţei multivariate dintre fiecare observaţie şi centroidul mulţimii datelor.

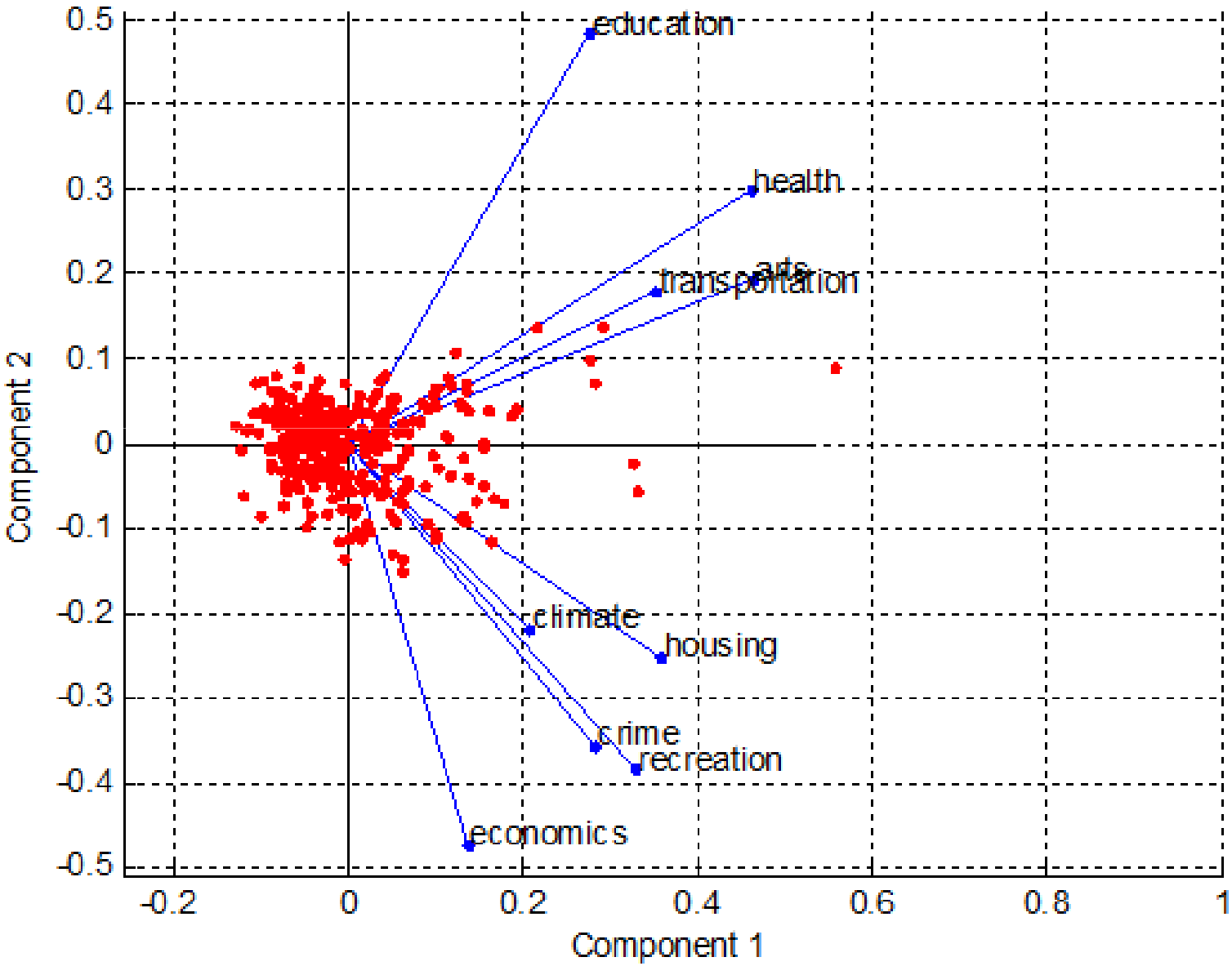
Este o manieră analitică de a găsi punctele extreme din baza de date.

```
>> [st2, index] = sort(t2,'descend'); % sortare in ordine descrescatoare
>>extreme = index(1)
extreme =
   213
>>names(extreme,:)
ans =
   New York, NY
```

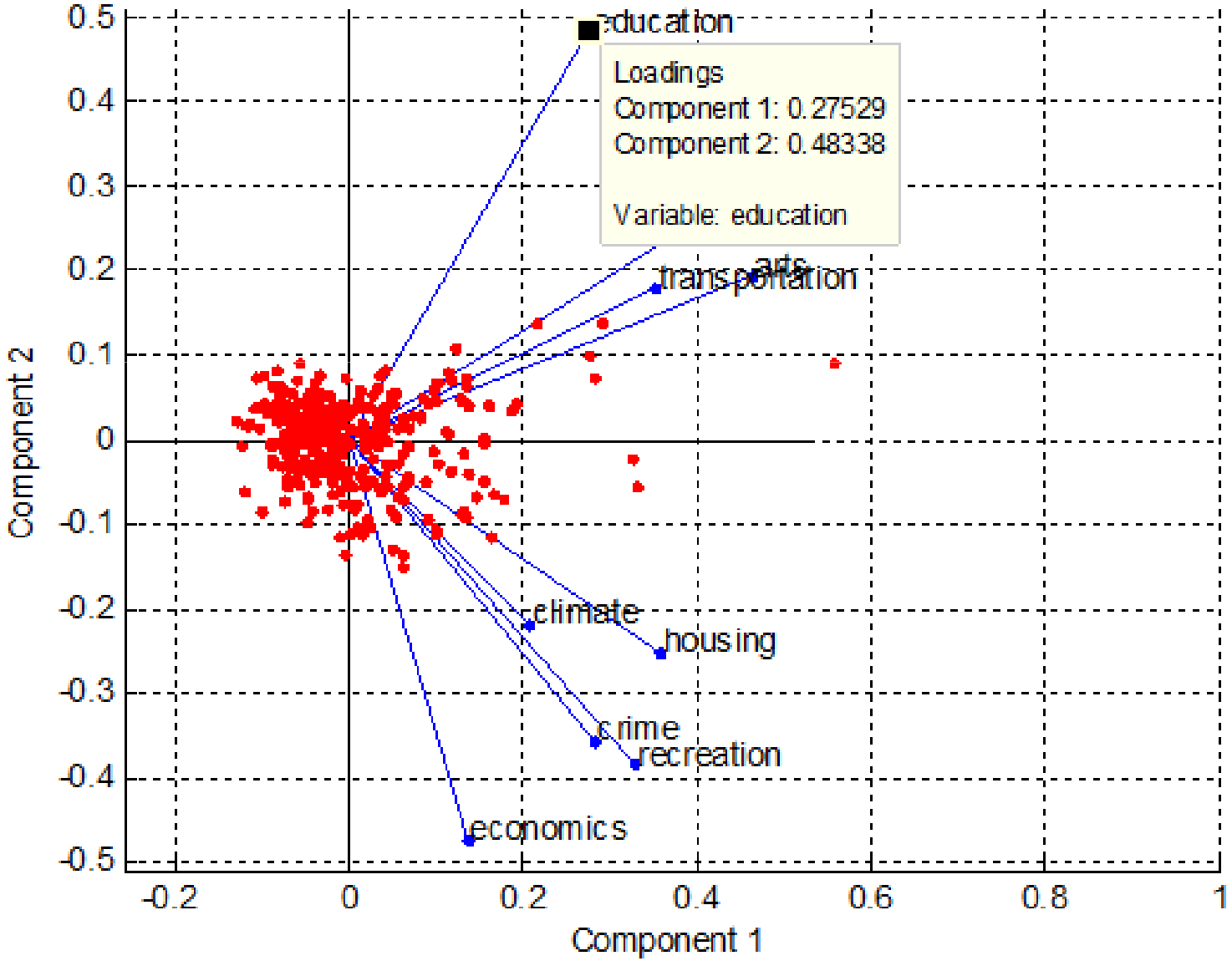
Nu este surprinzător că rezultatele pentru New York sunt cele mai îndepărtate de medie.

Vom reprezenta grafic rezultatele din analiza componentelor principale şi vom eticheta fiecare variabilă (caracteristică).

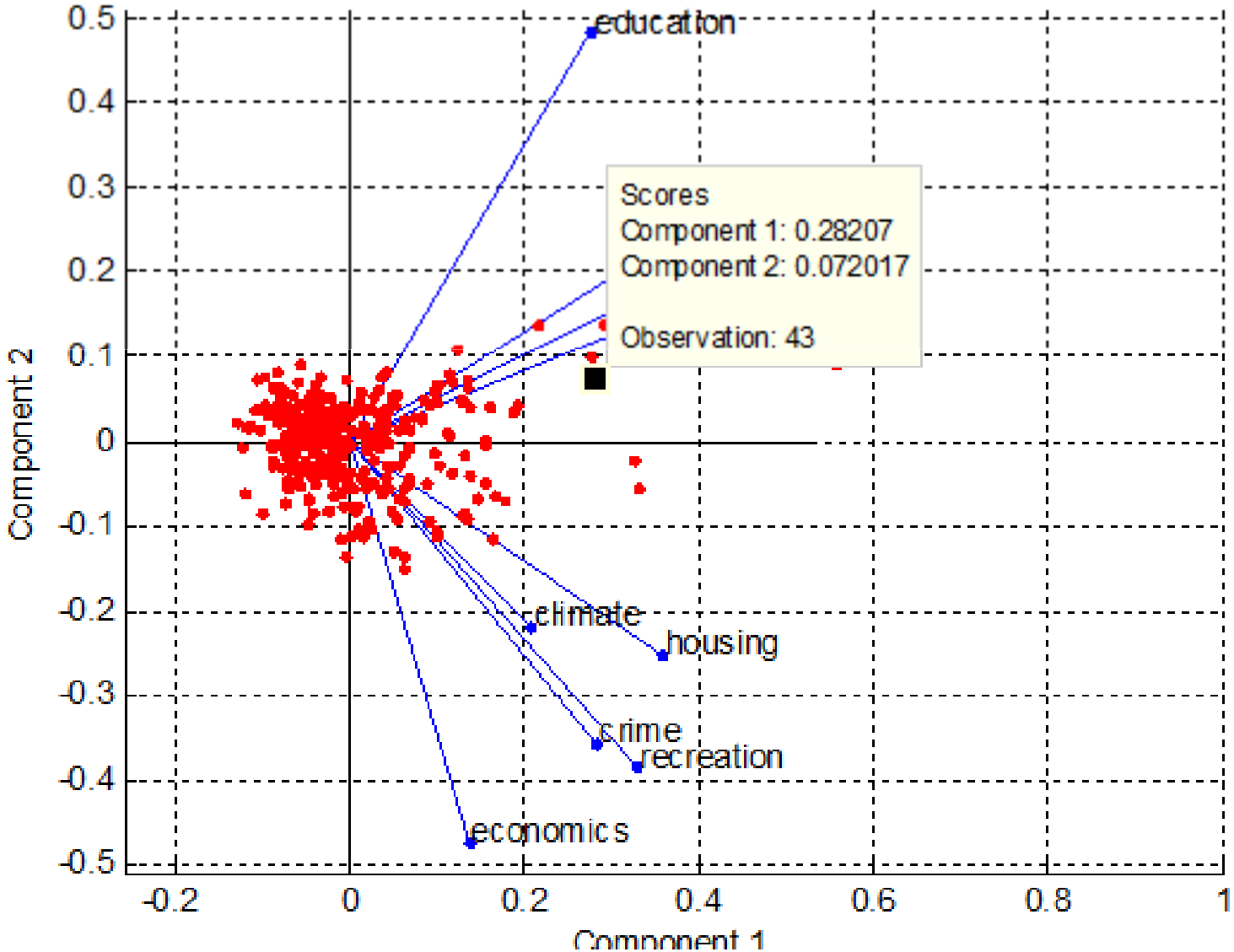
```
>>biplot(coefs(:,1:2), 'scores',scores(:,1:2),'varlabels',categories);
>>axis([-.26 1 -.51 .51]);
```


education
health
arts
transportation
climate
housing
crime
recreation
economics
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-0.2
0
0.2
0.4
0.6
0.8
1
Component 2
Component 1

Utilizând **Data Cursor**, din meniul **Tools** putem identifica elementele acestui grafic. Făcând click pe un vector ce reprezintă o variabilă putem citi coeficienții acestei variabile pentru fiecare componenă principală.

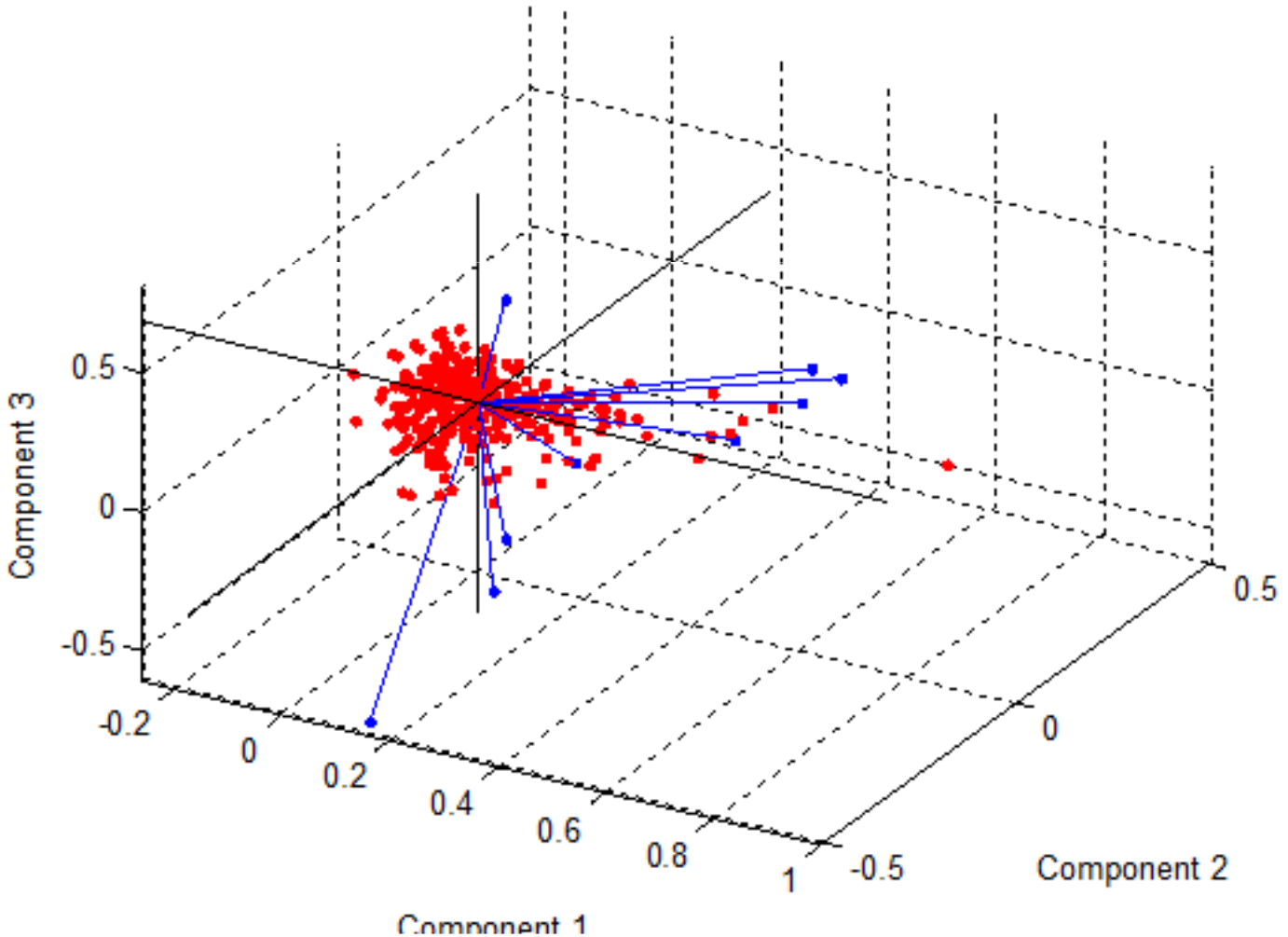


Făcând click pe un punct (o observatie-un oraş) putem citi scorurile observaţiei corespunzătoare fiecărei componente principale.



Putem construi un biplot în spaţiu tridimensional, lucru util dacă primele doua componente principale nu explică suficient dispersia datelor.

```
>>biplot(coefs(:,1:3), 'scores',scores(:,1:3),'obslabels',names);
axis([-.26 1 -.51 .51 -.61 .81]);
view([30 40]);
```



În încheiere, să amintim că PCA mai este cunoscută şi ca *transformarea Hotteling* (*Hotelling transform*) sau *transformarea Karhunen-Loève* (*Karhunen-Loève transform* –KLT). Pentru amănunte privind calculele şi principalii algoritmi utilizaţi (e.g. *metoda covarianţei*, *metoda corelaţiei*) cititorul este îndemnat să consulte [102] sau materialele „*A tutorial on Principal Components Analysis*", L. Smith URL: (http://csnet.otago.ac.nz/cosc 453/ student _ tutorials/principal_components.pdf),

Comparaţie între analiza factorială şi analiza componentelor principale

Anumiţi algoritmi de analiza factorială implică analiza componentelor principale.

Ambele tehnici reduc dimensiunea, în sensul că o mulţime mare de variabile (caracteristici) observate poate fi redusă la o mulţime mai mică de noi variabile

De cele mai multe ori dau rezultate asemănătoare

Dacă dorim să aproximăm datele într-un spaţiu de o dimensiune mai mică , de exemplu dorim să le vizualizăm, folosim **analiza componentelor principale.**

Dacă avem nevoie de un model care să explice corelaţiile intre date folosim **analiza factorială.**

Analiza canonică

Să presupunem că staff-ul unui supermarket este interesat să analizeze gradul de satisfacţie a clienţilor săi relativ la modul de servire.
Pentru aceasta, clienţii sunt rugaţi să completeze un chestionar cu un anumit număr de întrebări relative la satisfacţia privind modul de servire. În acelaşi timp, ei sunt de asemenea rugaţi să răspundă la alt set de întrebări, care se referă la măsurarea gradului de satisfacţie în alte domenii decât al servirii (calitatea produselor, diversitatea produselor etc.).

Problema care se pune este aceea de a identifica posibile conexiuni între satisfacţia faţă de servire şi satisfacţia faţă de alte aspecte ale activităţii supermarketului.

In cazul regresiei (liniare) simple sau multiple aveam de a face cu un set format dintr-una sau mai multe variabile predictoare, explicative, şi de o variabilă dependentă (criteriu) care era determinată de celelalte.
Aşa după cum uşor se observă, este cazul în care variabila dependentă ar fi reprezentată doar de gradul de satisfacţie faţă de modul de servire.

În cazul de faţă există însă şi alte variabile dependente (calitatea produselor, diversitatea produselor etc.), astfel încât suntem în situaţia să avem un set de variabile explicative şi un set de variabile dependente în loc de o singură variabilă dependentă.

În acest caz, generalizând tehnica regresiei liniare multiple, suntem interesaţi să corelăm un set de variabile dependente, fiecare din ele fiind ponderată, cu un set de variabile predictoare, de asemenea ponderate.

Formal, având un set de variabile explicative $\{X_1, X_2,..., X_q\}$ și un set de variabile dependente $\{Y_1, X_2,..., Y_p\}$, trebuie identificată ecuația:

$$a_1.Y_1+a_2.Y_2+...+a_p.Y_p = b_1.X_1 +b_2.X_2+...+b_q.X_q,$$

care stabileşte legătura dintre cele două seturi de variabile.

Din punct de vedere computațional, există programe specializate pentru rezolvarea acestei probleme (e.g. *Statistica –Multivariate exploratory techniques*, *SPSS, Matlab- Statistical Tools*).

Analiza canonică in Matlab

[A,B] = canoncorr(X,Y) calculează coeficienţii canonici ai matricelor de date X (de dimensiune $n \times d_1$) şi Y (de dimensiune $n \times d_2$).

Matricele X şi Y au acelaşi bunăr de observaţii (linii), numărul atributelor (variabilelor- coloane) fiind diferit.

A şi B sunt matrice de dimensiuni $d_1 \times d$ respectiv $d_2 \times d$, unde

$$d = \min(rang(X), rang(Y))$$

Coloanele matricelor A şi B sunt scalate astfel încât matricele de covarianţă ale variabilelor canonice să fie matrice identitate.

[A,B,r] = canoncorr(X,Y) returnează şi un vector r de dimensiune d ce conţine corelaţiile canonice, adică elementul j a lui r este coeficientul de corelaţie între coloanele j din U şi V.

Matricele U şi V sunt calculate conform formulelor:

U = (X-repmat(mean(X),N,1))*A, V = (Y-repmat(mean(Y),N,1))*B

unde N este dimensiunea lui X

[A,B,r,U,V] = canoncorr(X,Y) returnează în plus matricele U şi V.

5. exemplu

Baza carbig (Matlab) conţine atributele a 406 maşini din perioada 1970-1980, caracteristicile acestora fiind: cilindree, puterea motorului, greutate, acceleraţie, consumul de carburant MPG (câte mile face autovehicolul cu un galon de combustibil).
Prezentăm doar primele 14 maşini (observaţii) cu cele 5 caracteristici ale lor

```
>>load carbig
>> X = [Displacement Horsepower Weight Acceleration MPG]
X =
  1.0e+003 *
    0.3070    0.1300    3.5040    0.0120    0.0180
    0.3500    0.1650    3.6930    0.0115    0.0150
    0.3180    0.1500    3.4360    0.0110    0.0180
    0.3040    0.1500    3.4330    0.0120    0.0160
    0.3020    0.1400    3.4490    0.0105    0.0170
    0.4290    0.1980    4.3410    0.0100    0.0150
    0.4540    0.2200    4.3540    0.0090    0.0140
    0.4400    0.2150    4.3120    0.0085    0.0140
    0.4550    0.2250    4.4250    0.0100    0.0140
    0.3900    0.1900    3.8500    0.0085    0.0150
    0.1330    0.1150    3.0900    0.0175       NaN
    0.3500    0.1650    4.1420    0.0115       NaN
    0.3510    0.1530    4.0340    0.0110       NaN
    0.3830    0.1750    4.1660    0.0105       NaN
.....
```

In baza de date anumite valori ale caracteristicilor sunt NaN, probabil date cenzurate şi astfel e nevoie să rezolvăm această problemă, în aşa fel încât când apelăm canoncorr să nu avem în matrice NaN.

```
>> nans = sum(isnan(X),2)
nans =

     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     1
     1
     1
     1
.....
```

```
>> [A B r U V] = canoncorr(X(~nans,1:3),X(~nans,4:5))
A =
    0.0025    0.0048
    0.0202    0.0409
   -0.0000   -0.0027
B =
   -0.1666   -0.3637
   -0.0916    0.1078
r =
    0.8782    0.6328
U =
    0.7843    0.1737
    1.5940    1.3053
    1.2174    1.2266
    1.1824    1.1677
    0.9751    0.7061
    2.4421    1.2987
    2.9486    2.2835
    2.8136    2.1244
    3.0503    2.3028
    2.1951    2.0992
    1.7808    2.0156
    1.4701    1.2778
    1.4146    0.7486
    3.0834    5.8868
   -0.3800    0.8444
```

```
>> [A B r U V] = canoncorr(X(~nans,1:3),X(~nans,4:5))
A =
    0.0025    0.0048
    0.0202    0.0409
   -0.0000   -0.0027
B =
   -0.1666   -0.3637
   -0.0916    0.1078
r =
    0.8782    0.6328
U =
    0.7843    0.1737
    1.5940    1.3053
    1.2174    1.2266
    1.1824    1.1677
    0.9751    0.7061
    2.4421    1.2987
    2.9486    2.2835
    2.8136    2.1244
    3.0503    2.3028
    2.1951    2.0992
    1.7808    2.0156
    1.4701    1.2778
    1.4146    0.7486
    3.0834    5.8868
```

V =

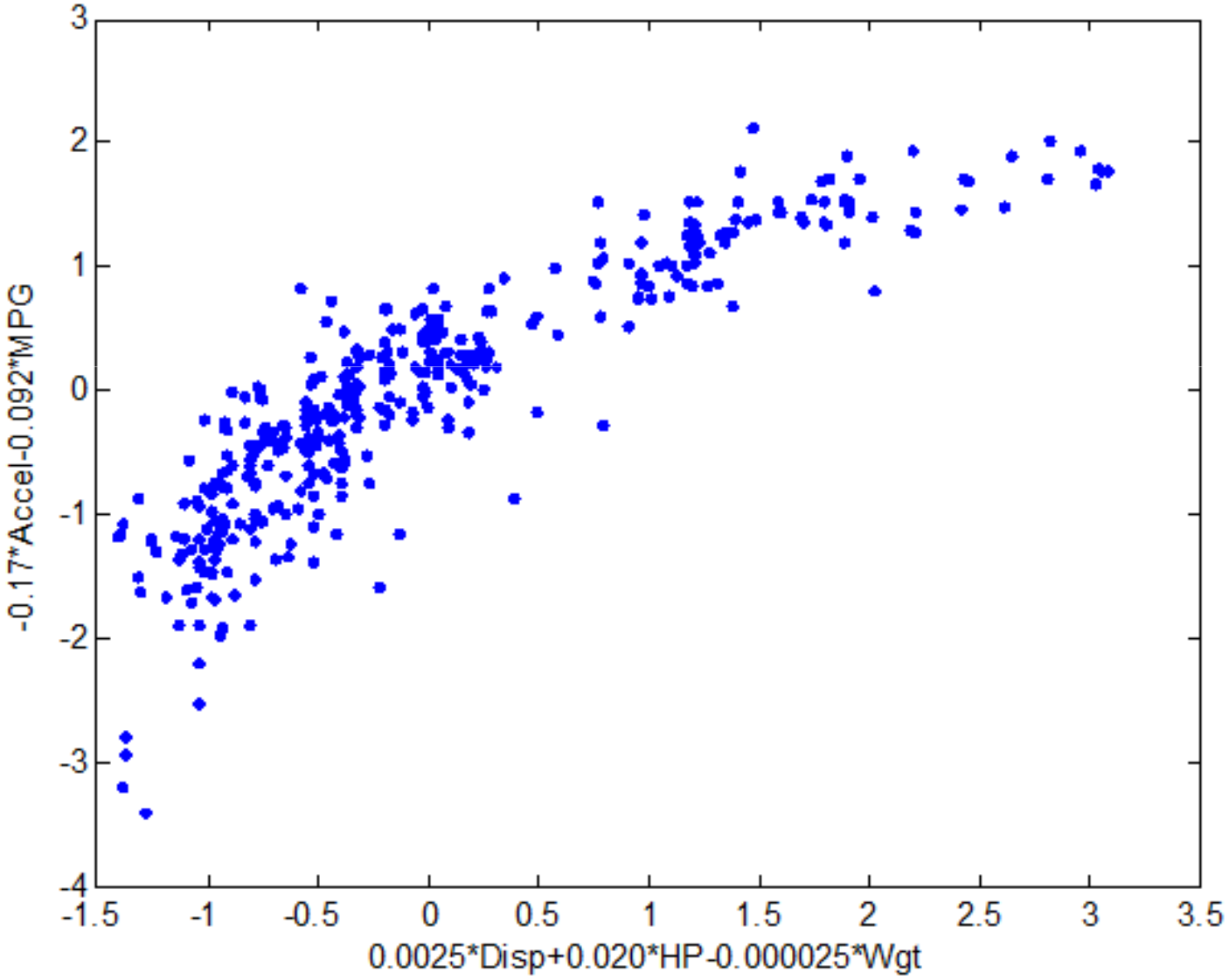
1.0886	0.7011
1.4466	0.5596
1.2553	1.0649
1.2717	0.4856
1.4301	1.1389
1.6965	1.1052
1.9547	1.3612
2.0380	1.5431
1.7881	0.9975
1.9465	1.6509
1.6965	1.1052
2.1213	1.7249
1.7798	1.2871
1.7881	0.9975
0.0395	0.2566
0.1393	-0.1408

……

Am fi fost tentaţi să lucrăm mai simplu, dar în acest caz nu se puteau calcula coeficienţii

```
>> [A B r U V] = canoncorr(X(:,1:3),X(:,4:5))
Warning: X is not full rank.
```

```
plot(U(:,1),V(:,1),'.')
xlabel('0.0025*Disp+0.020*HP-0.000025*Wgt')
ylabel('-0.17*Accel-0.092*MPG')
```



Analiza discriminant

Analiza discriminant reprezintă o metodă de clasificare a unor obiecte în anumite clase pe baza analizei unui set de variabile predictoare –input-uri. Modelul se bazează, în principiu, pe un set de observaţii pentru care se cunosc *a priori* clasele, formând setul de antrenament.

Pe baza antrenamentului, se construieşte un set de funcţii discriminant, de forma:

$$L_i = b_1 \cdot X_1 + b_2 \cdot X_2 + ... + b_n \cdot X_n + c,\ i = 1, 2, ..., k,$$

unde X_1, X_2,..., X_n sunt variabilele predictoare (care discriminează între clase), b_1, b_2,..., b_n reprezintă *coeficienţii discriminant*, iar c este o constantă.

Utilizând datele din mulţimea de antrenament analiza discriminant estimează parametrii funcţiilor discriminant, care sunt funcţii de n variabile(variabilele predictor). Aceste funcţii detrmină frontierele dintre diferite clase în spaţiul predictorilor.

Fiecare funcţie discriminant L_i corespunde unei clase Ω_i , $i = 1, 2,..., k$, în care trebuie să partiţionăm observaţiile.

O nouă instanţă va fi clasificată în acea categorie pentru care funcţia discriminat corespunzătoare ia valoarea maximă.

Ca domenii de aplicaţii concrete pentru analiza discriminat amintim:

- recunoaşterea feţelor (*face recognition*),
- marketing (distincţia între clienţi, managementul produselor etc.),
- medicina
- etc.,

menţionând existenţa programelor specializate necesare.

exemple

Să presupunem că pentru stabilirea clasei unui uragan avem la dispoziţie mai multe măsurători relative la diferite caracteristici meteorologice premergătoare declanşării uraganului (variabilele predictive discriminatorii).
Studiul pe care îl efectuăm îşi propune să stabilească care variabile sunt cele mai bune predictoare ale clasei uraganului, deci care variabile fac efectiv distincţie (*discriminare*) între diferitele categorii de uragane.

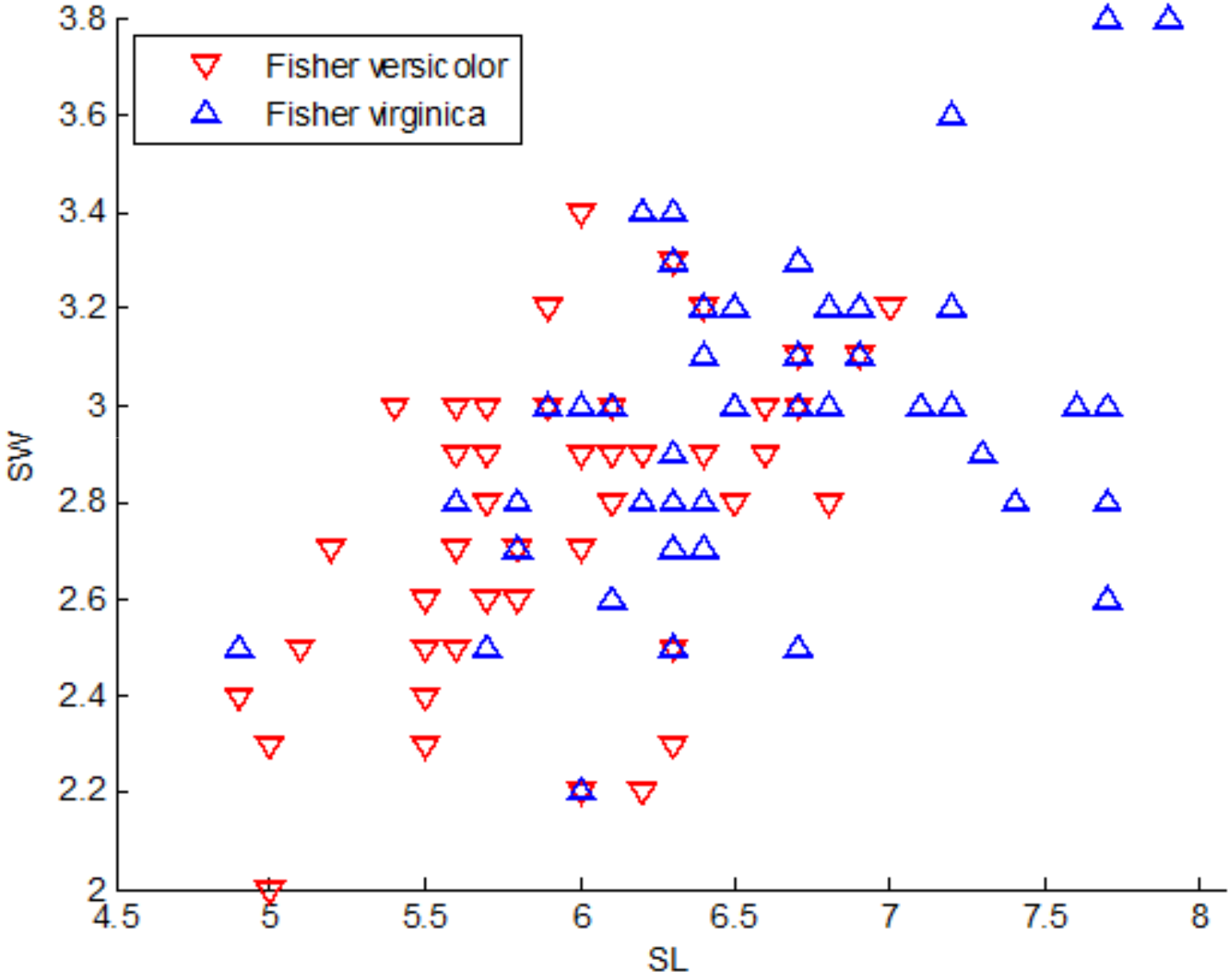
Analog, în comerţ putem analiza ce caracteristici (variabile discriminatorii) fac diferenţa în ceea ce priveşte raţiunea pentru care un cumpărător alege dintre mai multe categorii de produse unul anume.

În domeniul medical, de asemenea, un doctor este interesat ce caracteristici pot determina modul în care un pacient se poate vindeca complet, parţial sau deloc.

6. exemplu

Ca mulţime de antrenament, considerăm valorile sepalelor pentru irişii versicolor şi virginica, din baza de date a lui Fisher.
După încarcarea bazei de date, notăm cu SL, respectiv SW dimensiunile sepalelor şi cu group clasele cărora le aparţin florile de la nr 51 până la sfârşit (150).
Desenăm în plan punctele ale căror coordonate sunt dimensiunile sepalelor, cu roşu, respectiv albastru în funcţie de clasă căreia îi aparţin:

```
>>load fisheriris
>>SL = meas(51:end,1);
>>SW = meas(51:end,2);
>>group = species(51:end);
>>h1 = gscatter(SL,SW,group,'rb','v^',[],'off');
>>set(h1,'LineWidth',2)
>>legend('Fisher versicolor','Fisher virginica','Location','NW')
```


Fisher versicolor
Fisher virginica
SW
SL
3.8
3.6
3.4
3.2
3
2.8
2.6
2.4
2.2
2
4.5
5
5.5
6
6.5
7
7.5
8

class = classify(sample,training,group,type) clasifică fiecare linie din matricea datelor eşantionului –*sample* (fiecare observaţie) într-unul din grupurile (clasele) mulţimii de antrenament- *training*.

sample şi training sunt matrice cu acelaşi număr de coloane, în timp ce group este o variabilă ce etichetează elementele din mulţimea de antrenament. Fiecare observaţie din mulţimea de antrenament aparţine unei anumite clase

training şi group au acelaşi număr de linii.

type specifică tipul funcţiei discriminant: l 'linear' , 'quadratic', 'mahalanobis'.

classify consideră NaN ca valori lipsă şi ignoră respectivele linii ale mulţimii de antrenament.

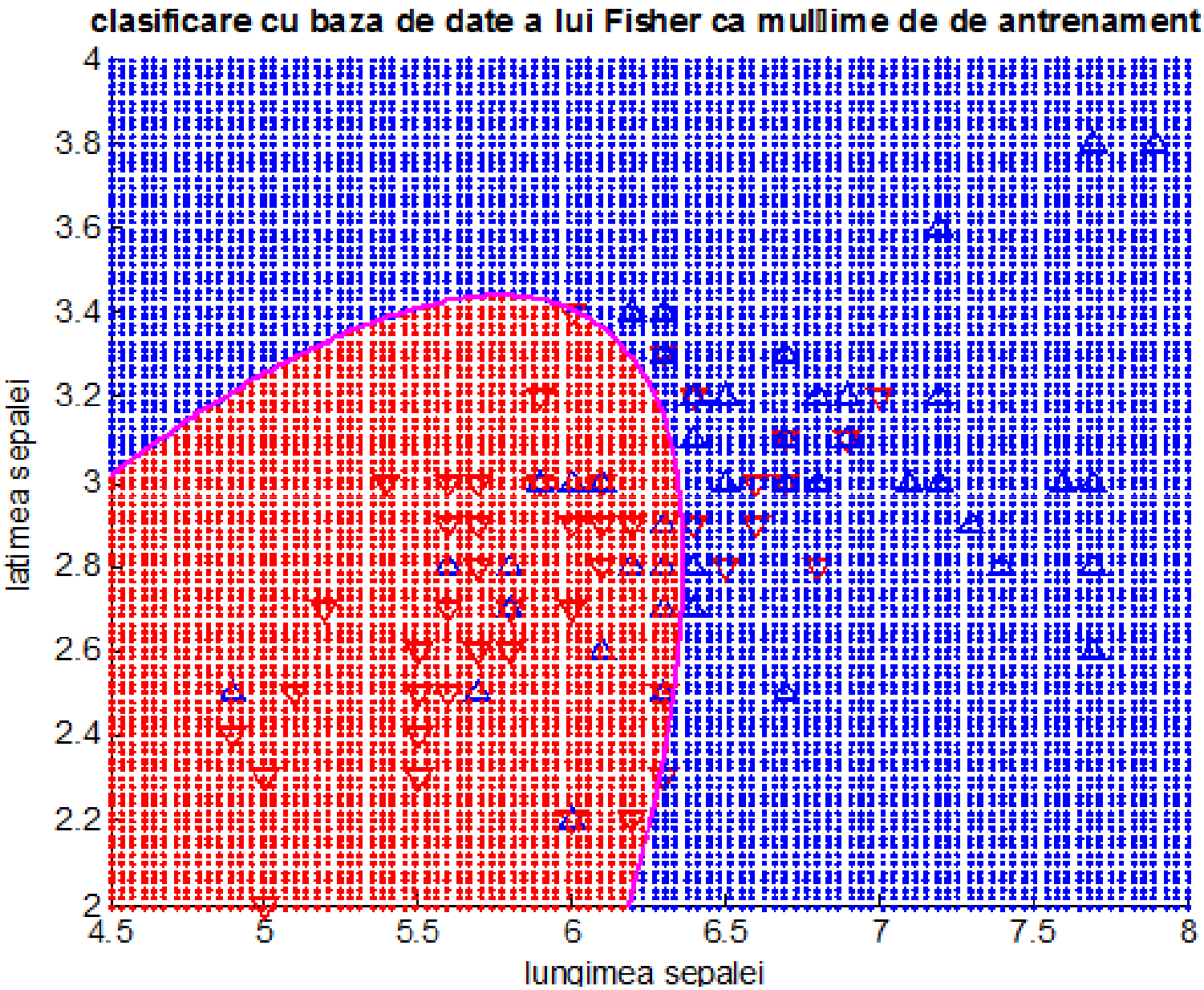
Outputul class îndică clasa căreia îi aparţine fiecare observaţie.

Vom clasifica o rețea de valori utilizând **classify**

```
>> [X,Y] = meshgrid(linspace(4.5,8),linspace(2,4));
>>X = X(:); Y = Y(:);
>> [C,err,P,logp,coeff] = classify([X Y],[SL SW],group,'quadratic');
```

Vizualizăm clasificarea:

```
>>hold on;
>>gscatter(X,Y,C,'rb','.',1,'off');
>>K = coeff(1,2).const;
>>L = coeff(1,2).linear;
>>Q = coeff(1,2).quadratic;
>>f = sprintf('0 = %g+%g*x+%g*y+%g*x^2+%g*x.*y+%g*y.^2',...
K,L,Q(1,1),Q(1,2)+Q(2,1),Q(2,2));
>>h2 = ezplot(f,[4.5 8 2 4]);
>>set(h2,'Color','m','LineWidth',2)
>>axis([4.5 8 2 4])
xlabel('lungimea sepalei')
ylabel('latimea sepalei')
      title('{\bf clasificare cu baza de date a lui Fisher ca mulţime de
      antrenament}')
```


clasificare cu baza de date a lui Fisher ca mulțime de de antrenament
lățimea sepalei
lungimea sepalei
4
3.8
3.6
3.4
3.2
3
2.8
2.6
2.4
2.2
2
4.5
5
5.5
6
6.5
7
7.5
8

Detectarea anomaliilor

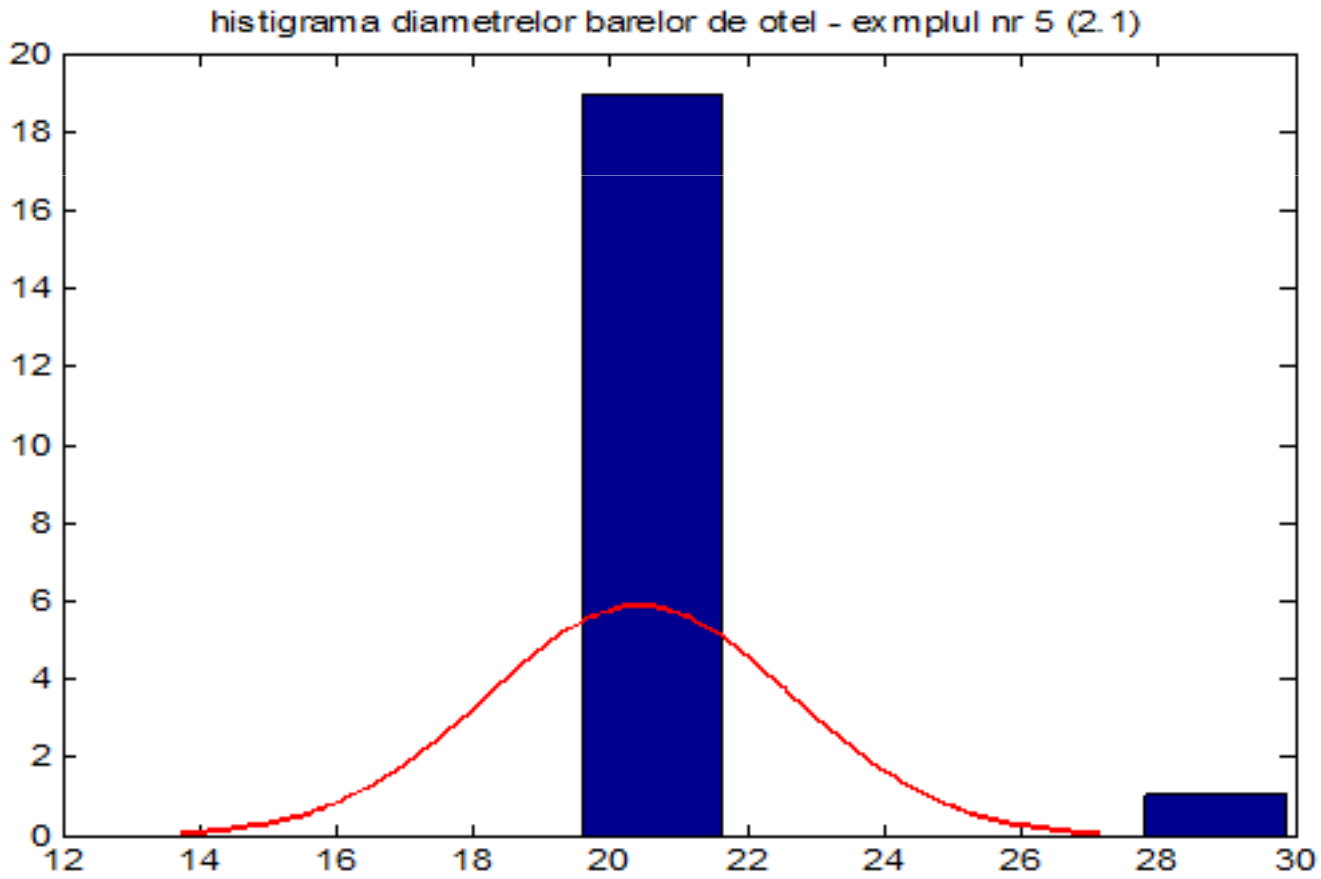
Termenul de *anomalie/valoare extremă/valoare excepţională* este definit în Statistică ca fiind acea valoare care se găseşte „foarte departe” de restul datelor, fiind o *singularitate* (un punct *izolat*) a setului de date. Repartiția gaussiană (normală) presupune existența valorilor extreme, dar în număr redus, a se vedea cazul extremităților *clopotului* lui Gauss. Așadar existența acestor valori în date este normală, ele indicând de cele mai multe ori fie date eronate în sine, fie date colectate eronat, fie chiar date corecte, generate în mod natural de fenomenul în cauză, de exemplu cazurile de nanism sau gifantism la oameni.

Exemple

- În exemplul nr. 5 din Noţiuni introductive, privitor la înălţimea băieţilor dintr-o clasă de gimnaziu, înălţimea băiatului care suferea de o uşoară formă de nanism este o valoare extremă.
- În exemplul nr. 4 din acest capitol privitor la cele 329 de oraşe din SUA, New-York era o valoare extremă, componentele sale fiind cele mai îndepărtate de medie, confirmând percepţia oamenilor despre acest oraş „atipic".
- Repartiţiile negaussiene, asimetrice pot prezenta astfel de valori (poziţionate la *coada* curbei), fiind necesară examinarea repartiţiei datelor, pentru a hotărî dacă anomaliile vor fi menţinute sau îndepărtate.

- În exemplul nr. 7 din Noţiuni introductive privind diametrele barelor de oţel dintr-un eşantion aleator ales, există un extrem- acea bară cu diametrul de 29.9 evident un rebut. Desenând histograma corespunzătoare, vedem că reprezentarea grafică a datelor este foarte utilă pentru identificarea valorilor excepţionale.

```
>> X3=[19.9 19.8 20.1 19.9 19.7 20.1 20 19.6 19.7 20.1 20.4 20
19.9 19.8 20.2 20 19.8 19.6 29.9 20.3]
```

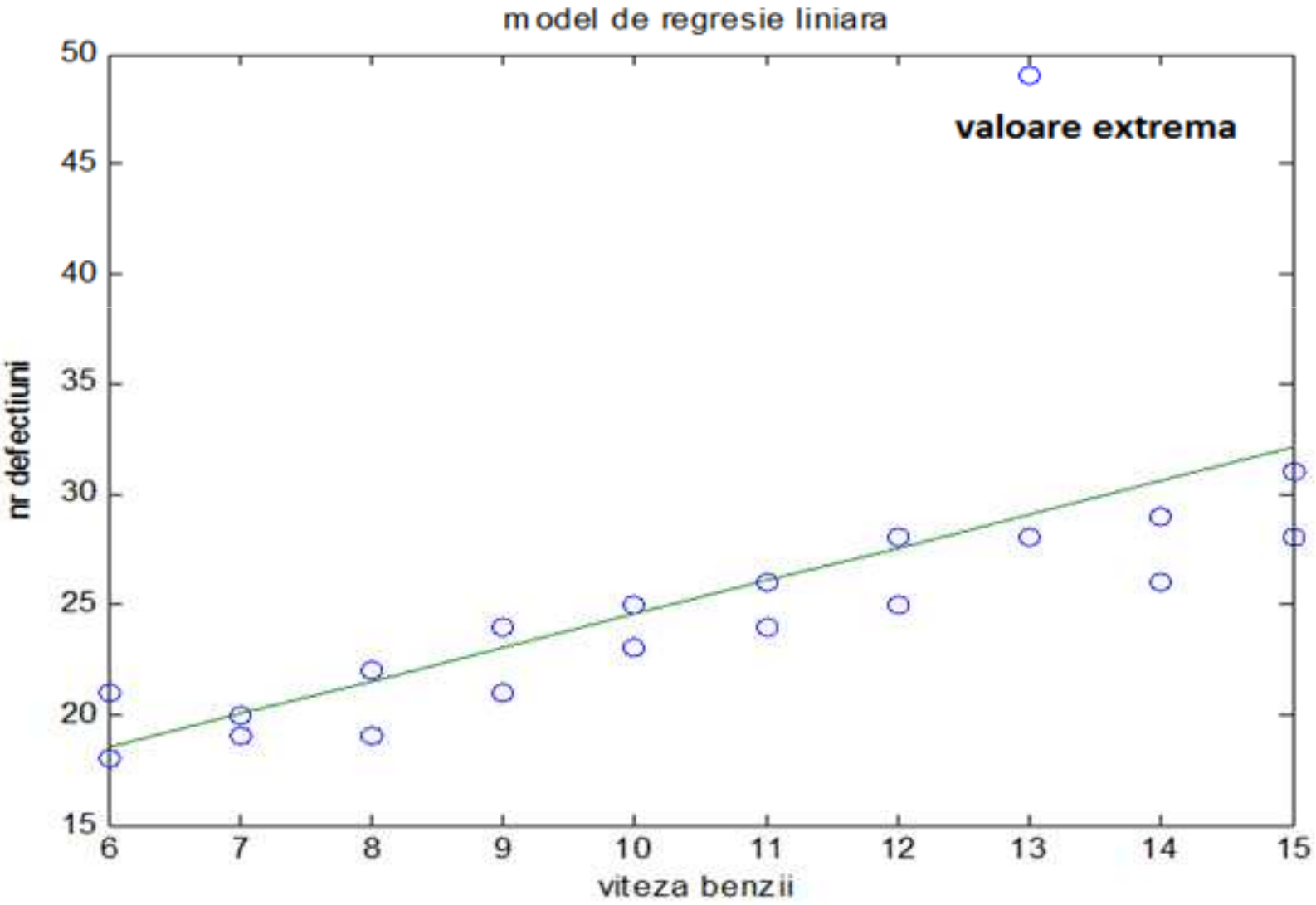


7. exemplu

- Prezentăm acum un model regresiv liniar cu o valoare extremă. Astfel într-o fabrică s-a constatat că viteza (m/min) benzii de lucru afectează numărul de defecţiuni descoperite în timpul verificării. Următorul tabel prezintă diferite viteze ale benzii şi numărul defectelor găsite.

viteza	nr defecţiuni		viteza	nr defecţiuni
6	21		11	24
6	18		11	26
7	19		12	25
7	20		12	28
8	22		13	49
8	19		13	28
9	21		14	26
9	24		14	29
10	23		15	28
10	25		15	31

```
>> X=[6 6 7 7 8 8 9 9 10 10 11 11 12 12 13 13 14 14 15 15];
>> Y=[21 18 19 20 22 19 21 24 23 25 24 26 25 28 49 28 26 29 28 31];
>> A = [ones(size(X)); X]';
>> a = regress(Y',A);
>> plot(X,Y,'o',X,a(1)+a(2)*X)
```



Apariția acestor valori nespecifice/extreme în date poate influența semnificativ estimarea diferiților parametrii ai modelului. Estimațiile și modelele care nu sunt influențate semnificativ de existența anomaliilor se numesc *robuste*.

Exemplu

Mediana este mai robustă decât media la existența anomaliilor (a se vedea exemplul nr. 3 modificat din paragraful 1.2)

Plecând de la valoarea medianei, se pot defini diferite tipuri de anomalii. De *exemplu*, dacă valoarea anomaliei este mai mare decât triplul medianei și mai mică decât de 5 ori mărimea acesteia, anomalia este de tipul I. În cazul datelor multidimensionale, aceste comparații se pot face pe fiecare coordonată.

Detectarea anomaliilor, privite şi prin prisma tehnicilor de Data Mining are numeroase aplicaţii, cum ar fi: detectarea fraudelor cu carduri bancare, detecția intruziunilor în diferite tipuri de rețelele, detecția defecțiunilor diferitelor sisteme, procesarea sunetului şi imaginilor, criptografie etc.

Procesul de identificare a anomaliilor în date este o tehnică nesupervizată, în care se presupune că marea majoritate a datelor au valori *normale*. Acest proces are două etape. Pentru început, se construieşte profilul pattern-ului *normal* al datelor, profil care este utilizat în a doua etapă pentru detectarea anomaliilor pe baza măsurării diferenţei față de normalitate.

Ca tehnici utilizate în acest proces, amintim metode grafice, metode statistice, metode bazate pe măsurarea distanței și metode bazate pe modele.

Dintre metodele *grafice* utilizate în detectarea anomaliilor, cele mai utilizate sunt:

- Metoda *box-plot*

- Metoda diagramelor de împrăștiere (*scatterplot*)

- Metoda acoperirii convexe (*convex hull*). Reamintim că în matematică acoperirea convexă a unei mulțimi X este definită ca fiind cea mai mică mulțime convexă ce include mulțimea X, sau ca fiind mulțimea tuturor combinațiilor convexe de puncte din X.

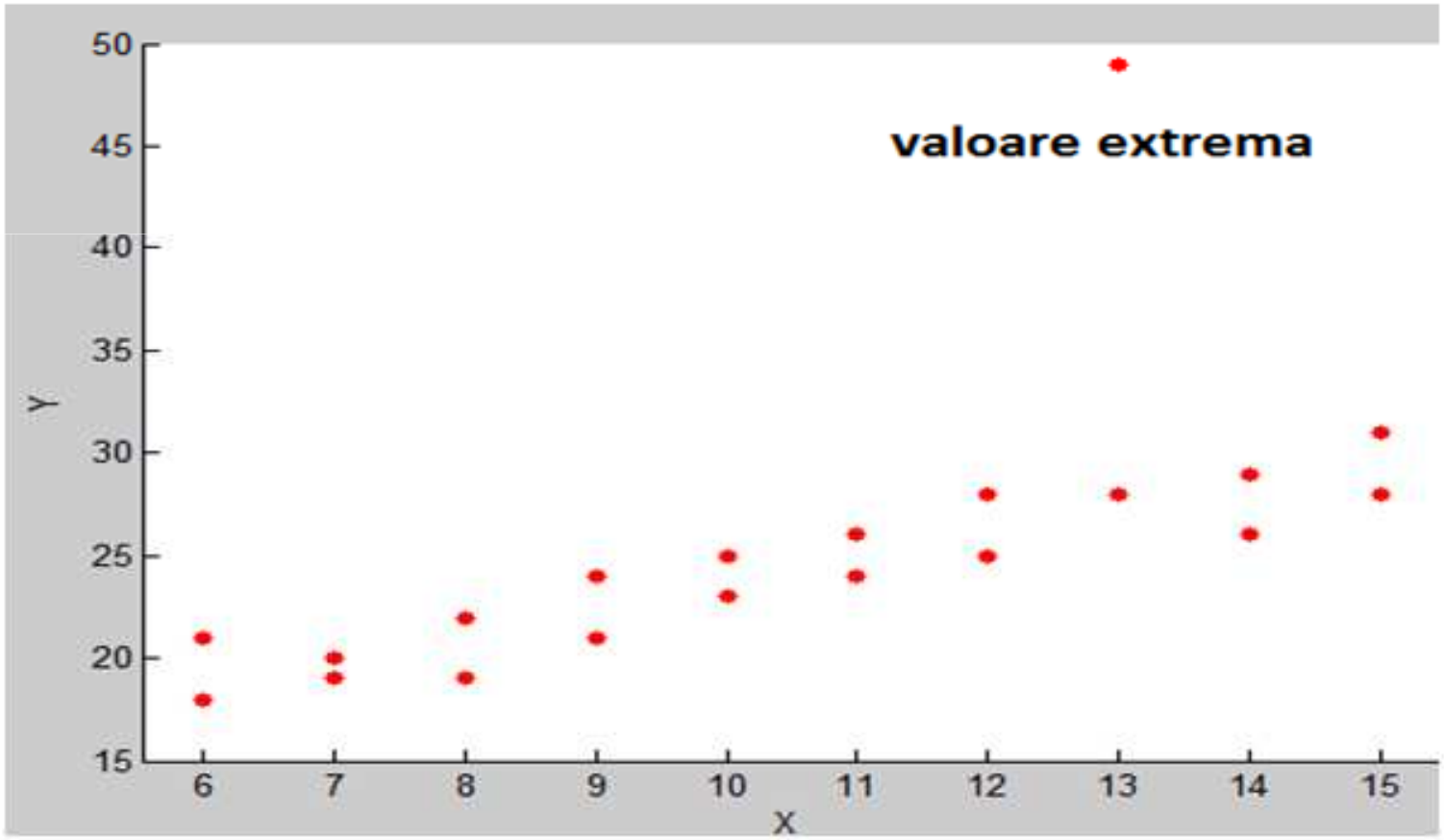
exemplu (box-plot)

Reluăm exemplul nr. 7 din Noţiuni introductive, privitor la diametrele barelor de oţel dintr-un eşantion aleator ales, vectorul X3 având drept componente diametrele barelor respective:

```
>>X3=[19.9 19.8 20.1 19.9 19.7 20.1 20 19.6 19.7 20.1 20.4 20 19.9
19.8 20.2 20 19.8 19.6 29.9 20.3];
>>boxplot(X3)
```

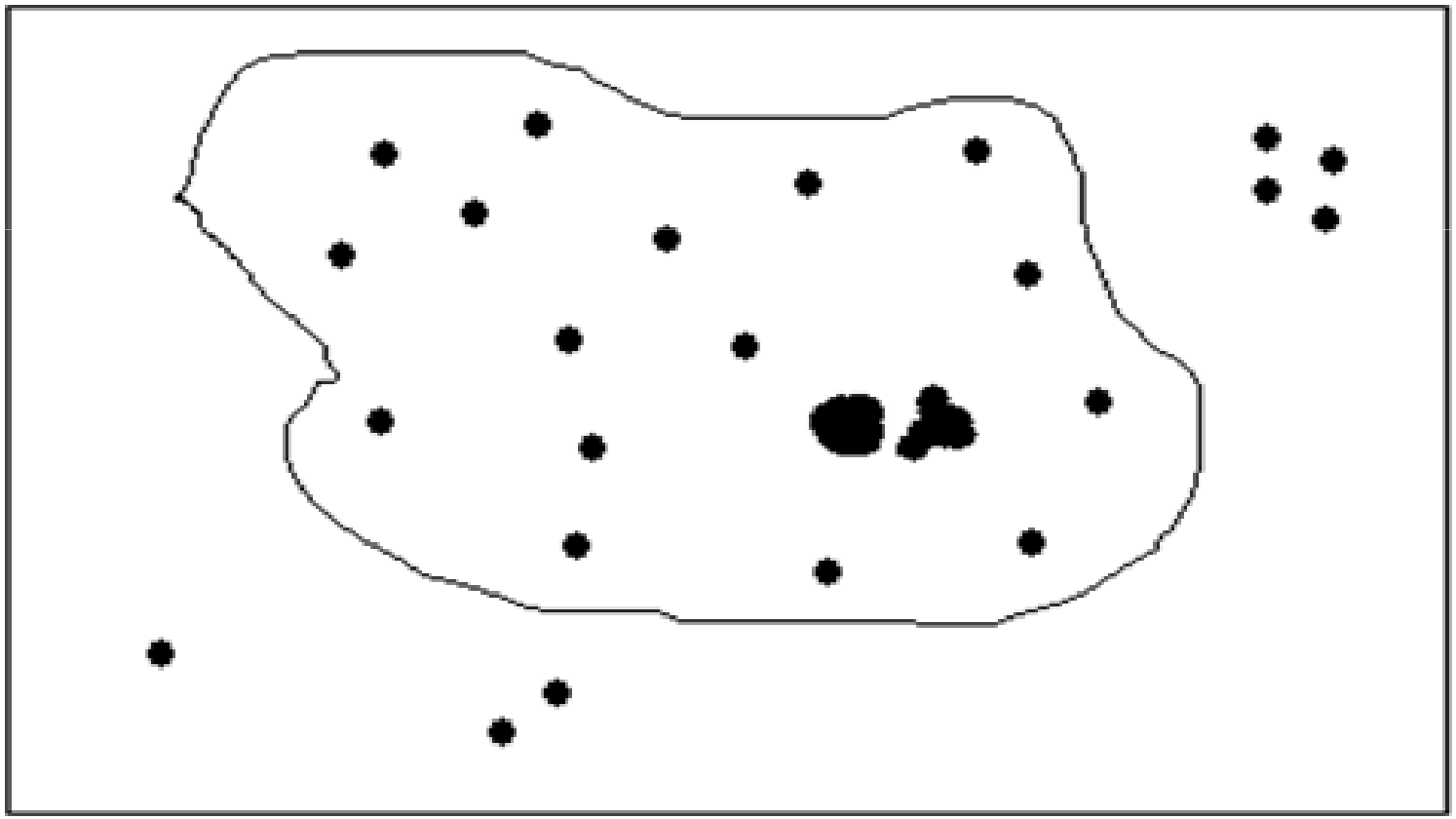
exemplu (diagrama de împrăştiere)

Prezentăm diagrama de împrăştiere în cazul exemplului anterior, ce are ca date viteza benzii de lucru şi numărul corespunzător de defecţiuni.



Exemplu (acoperirea convexa)

Desenul următor ilustrează metoda:



În acest caz, singura problemă notabilă este existența unor anomalii în interiorul acoperirii convexe (cele două *note* negre din interior)

Metodele grafice au avantajul faptului că sunt sugestive, dar, pe de altă parte, datorită subiectivității lor privind interpretarea imaginilor, pot duce la erori în detectarea valorilor anormale.

În utilizarea metodelor *statistice* se pleacă de la ipoteza că există un anumit model al repartiţiei datelor, aşa numitul pattern al datelor tipice, după care se utilizează diferite teste statistice pentru identificarea valorilor anormale în raport cu acest model, teste privind tipul repartiţiei, parametrii repartiţiei, intervalul de încredere.

Ţinând seama de rezultatul analizării pattern-ului datelor, pe baza testării statistice se pot identifica anomaliile/valorile extreme.

În cazul metodelor bazate pe *măsurarea distanţelor*, datele sunt reprezentate ca vectori aparţinând unui anumit spaţiu liniar normat. Prezentăm foarte succint două metode clasice de detectare a anomaliilor:

- Metoda *k-nearest-neighbor*. (*k*-NN) clasifică un obiect pe baza celor mai apropiate (*k*) obiecte din vecinătate.
- Metoda *clusteringului* se bazează pe divizarea setului de date în clustere (mănunchiuri) de date, pe baza similarităţii dintre eleşi identificarea anomaliilor -valori atipice, prin evidenţierea poziţiei lor singulare faţă de clusterele formate din valori tipice.

Metodele bazate pe *modele* utilizează tehnici Data Mining pentru identificarea anomaliilor în marea masă a datelor tipice fenomenului considerat. Astfel, se construieşte un model de *clasificare* pe baza unui număr suficient de mare de date, atât *tipice* cât şi *atipice*, cu două clase: A- date *normale* şi B- *anomalii*. După antrenarea sa pe date cunoscute, acest model se aplică la date noi, pentru detectarea eventualelor anomalii.