

# ΕΞΟΡΥΞΗ ΔΕΔΟΜΕΝΩΝ

## Εργασία 2

ΠΑΝΤΕΛΕΗΜΩΝ ΠΡΩΙΟΣ

ice18390023

7ο Εξάμηνο

ice18390023@uniwa.gr

Τμήμα Τρίτης 9:00-13:00



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ  
UNIVERSITY OF WEST ATTICA

**Υπεύθυνοι καθηγητές**

ΠΑΡΙΣ ΜΑΣΤΟΡΟΚΩΣΤΑΣ

Τμήμα Μηχανικών και Πληροφορικής Υπολογιστών  
18 Δεκεμβρίου 2021

## Περιεχόμενα

<b>1 Προεπεξεργασία δεδομένων</b>	<b>1</b>
1.1 Υπολογισμός παραδείγματος 1.7 . . . . .	1
1.2 Υπολογισμός παραδείγματος 1.8 . . . . .	5
<b>2 Μέτρα απόστασης</b>	<b>10</b>
2.1 Ευκλείδεια απόσταση και των τεσσάρων στηλών . . . . .	10
2.2 Υπολογισμός αποστάσεων για συνδυασμό στηλών . . . . .	12

## Κώδικες

1.1 Script παραδείγματος 1.7 . . . . .	3
1.2 Linear Transformation . . . . .	5
1.3 Z-score Transformation . . . . .	5
1.4 Script παραδείγματος 1.8 των δεδομένων data . . . . .	6
1.5 Script παραδείγματος 1.8 των δεδομένων iris . . . . .	6
2.1 Script υπολογισμού αποστάσεων . . . . .	11
2.2 Script υπολογισμού αποστάσεων για διάφορους συνδυασμούς χαρακτηριστικών	21

## Κατάλογος σχημάτων

1.1 Καθαρισμός δεδομένων με διαγραφή γραμμών . . . . .	2
1.2 Καθαρισμός δεδομένων με διαγραφή στηλών . . . . .	2
1.3 Καθαρισμός δεδομένων με αντικατάσταση του μηδενός . . . . .	3
1.4 Εμφάνιση των 2 πρώτων στηλών των δεδομένων με καθαρισμό αντικατάστασης σε μέση τιμή . . . . .	3
1.5 Μετασχηματισμένα δεδομένα . . . . .	6
1.6 Μήκος σέπαλου . . . . .	8
1.7 Πλάτος σέπαλου . . . . .	8
1.8 Μήκος πετάλου . . . . .	9
1.9 Πλάτος πετάλου . . . . .	9
2.1 Ευκλείδεια απόσταση της πρώτης παρατήρησης . . . . .	10
2.2 Euclidean, City block και Mahalanobis αποστάσεις της πρώτης παρατήρησης	11
2.3 Αποστάσεις της πρώτης παρατήρησης για πρώτο και δεύτερο χαρακτηριστικό	12
2.4 Αποστάσεις της πρώτης παρατήρησης για πρώτο και δεύτερο χαρακτηριστικό σε σύγκριση και με τα τέσσερα . . . . .	13
2.5 Αποστάσεις της πρώτης παρατήρησης για πρώτο και τρίτο χαρακτηριστικό	14
2.6 Αποστάσεις της πρώτης παρατήρησης για πρώτο και τρίτο χαρακτηριστικό σε σύγκριση και με τα τέσσερα . . . . .	15
2.7 Αποστάσεις της πρώτης παρατήρησης για τρίτο και τέταρτο χαρακτηριστικό	16
2.8 Αποστάσεις της πρώτης παρατήρησης για τρίτο και τέταρτο χαρακτηριστικό σε σύγκριση και με τα τέσσερα . . . . .	17

2.9	Αποστάσεις της πρώτης παρατήρησης για πρώτο, δεύτερο και τρίτο χαρακτηριστικό . . . . .	18
2.10	Αποστάσεις της πρώτης παρατήρησης για πρώτο, δεύτερο και τρίτο χαρακτηριστικό σε σύγκριση και με τα τέσσερα . . . . .	19
2.11	Αποστάσεις της πρώτης παρατήρησης για πρώτο, δεύτερο και τέταρτο χαρακτηριστικό . . . . .	20
2.12	Αποστάσεις της πρώτης παρατήρησης για πρώτο και τέταρτο χαρακτηριστικό σε σύγκριση και με τα τέσσερα . . . . .	21

## 1 Προεπεξεργασία δεδομένων

Τα δεδομένα χρειάζονται πριν την επεξεργασία τους πρέπει να **καθαριστούν**. Για το παράδειγμα 1.7 έχει χρησιμοποιηθεί ο τυχαίος πίνακας με NaNs `irisV`, ο οποίος βρίσκεται στο αρχείο `matlab.mat`.

### 1.1 Υπολογισμός παραδείγματος 1.7

Το script 1.1 υλοποιεί:

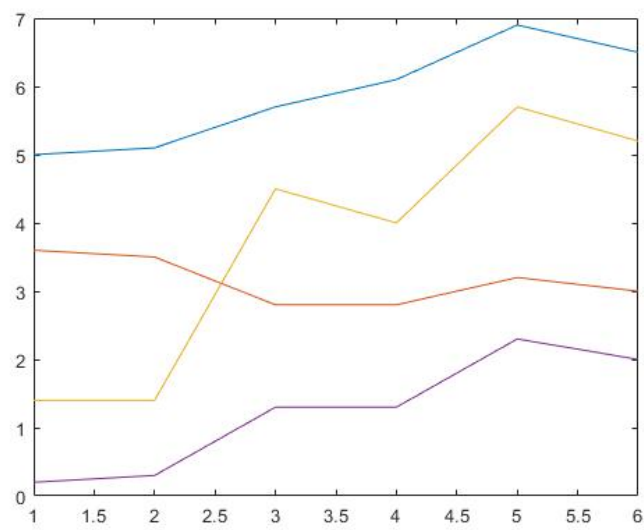
1. διαγραφή των γραμμών με NaNs
2. διαγραφή των στηλών με NaNs
3. αντικατάσταση NaNs με 0
4. αντικατάσταση των NaNs με την μέση τιμή της στήλης

Επειδή, η πιθανότητα αντικατάστασης των αρχικών τιμών με κάποια NaN ήταν αρκετά μεγάλη, οι γραμμές που παρέμειναν μετά την διαγραφή ήταν ελάχιστες όπως φαίνεται στην εικόνα 1.1.

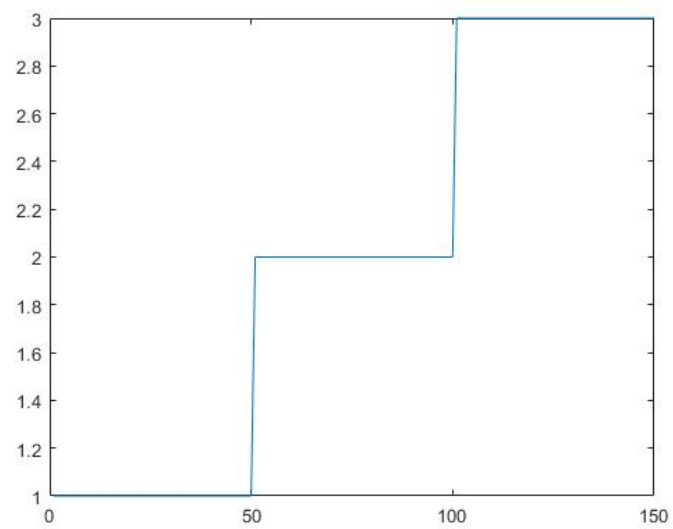
Στις 4 στήλες, πάντα θα υπάρχει τουλάχιστον ένα NaN, έτσι ώστε αυτός ο τρόπος να είναι άχρηστος. Βλέπουμε στο σχήμα 1.2, πως μόνο η στήλη 5 εμφανίζεται, η οποία δεν πειράχθηκε με NaN, όμως δεν χρησιμεύει κάπου.

Η αντικατάσταση με τα ελλιπή δεδομένα σε 0, είναι γεμάτα θόρυβο με αποτέλεσμα να μην μπορεί να βγει κάποιο πόρισμα (εικ. 1.3).

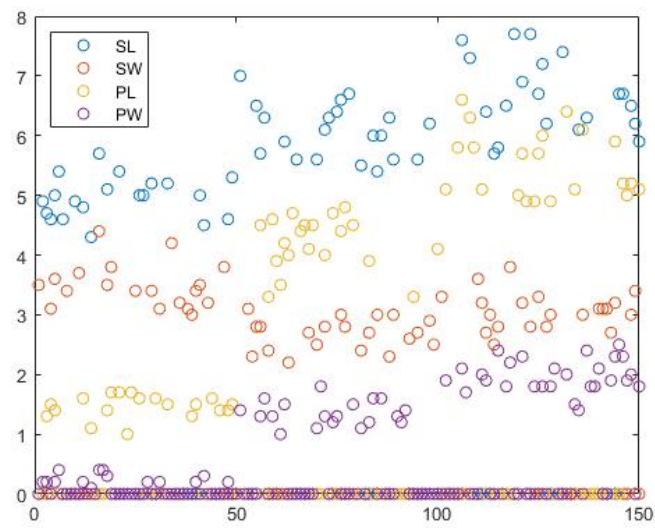
Η αντικατάσταση των NaN με την μέση τιμή της στήλης, είναι ποίο όμορφη οπτικά και ταυτόχρονα χρήσιμη διότι δεν αχρηστεύονται οι υπόλοιπες στήλες. Στο σχήμα 1.4 ο άξονας X αντιστοιχεί στο μήκος πετάλου και ο άξονας Y στο πλάτος πετάλου.



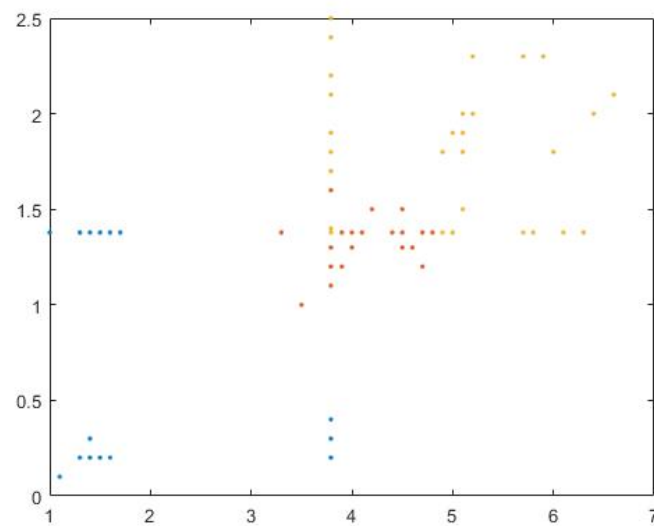
Σχήμα 1.1: Καθαρισμός δεδομένων με διαγραφή γραμμών



Σχήμα 1.2: Καθαρισμός δεδομένων με διαγραφή στηλών



Σχήμα 1.3: Καθαρισμός δεδομένων με αντικατάσταση του μηδενός



Σχήμα 1.4: Εμφάνιση των 2 πρώτων στηλών των δεδομένων με καθαρισμό αντικατάστασης σε μέση τιμή

```
1 load iris.dat
2
3 p=0.60;
4 irisV=iris;
```

```
5 [ro,~]=size(iris);
6 r1=randperm(ro);
7 irisV(r1(1:p*ro),1)=NaN;
8 r1=randperm(ro);
9 irisV(r1(1:p*ro),2)=NaN;
10 r1=randperm(ro);
11 irisV(r1(1:p*ro),3)=NaN;
12 r1=randperm(ro);
13 irisV(r1(1:p*ro),4)=NaN;
14
15 % 1) delete rows
16 data = irisV;
17 data(any(isnan(data),2),:) = [];
18 figure();
19 plot(data(:,1:4))
20
21 % 2) delete columns
22 data = irisV;
23 data(:,any(isnan(data),1)) = [];
24 figure();
25 plot(data)
26
27 % 3) NaN -> 0
28 data = irisV;
29 notNaN = ~isnan(data);
30 data(~notNaN) = 0;
31 figure();
32 plot(data(:,1:4),'o');
33 legend('SL','SW','PL','PW','Location','NorthWest')
34
35 % 4) NaN -> mean of column
36 data = irisV;
37 notNaN = ~isnan(data);
38 data(~notNaN) = 0;
39 totalNo = sum(notNaN);
40 columnTot = sum(data);
41 colMean = columnTot./totalNo;
42
43 for i = 1:length(colMean)
44     data(find(notNaN(:,i)==0),i)=colMean(i);
45 end
46
47 figure();
48 plot([data(1:50,3),data(51:100,3),data(101:150,3)],...
49      [data(1:50,4),data(51:100,4),data(101:150,4)],'.')
```

Κώδικας 1.1: Script παραδείγματος 1.7

## 1.2 Υπολογισμός παραδείγματος 1.8

Η συνάρτηση linear transform (κώδικας 1.2), μετασχηματίζει τα δεδομένα σαν την min-max, όπου ως νέο max είναι το 1 και ως νέο min είναι το 0 και για αυτό παραλείπονται. Ακόμα, η επόμενη συνάρτηση που μετασχηματίζει τα δεδομένα, είναι η z-score (κώδικας 1.3), η οποία επειδή ακολουθεί την κανονική κατανομή συνήθως οι τιμές τις διακυμαίνονται μεταξύ του -3 έως 3 και με σπάνιες περιπτώσεις σε μεγαλύτερα νούμερα κατά απόλυτη τιμή.

```
1 function yV = LinearTransform(xV)
2 xV = xV(:);
3 xmin = min(xV);
4 xmax = max(xV);
5 d = xmax - xmin;
6 yV = (xV - xmin) / d;
7 end
```

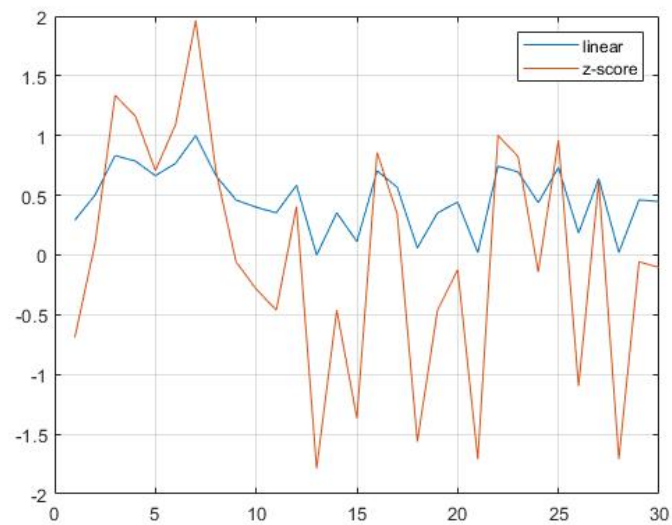
Κώδικας 1.2: Linear Transformation

```
1 function yV = zscoreTransform(xV)
2 xV = xV(:);
3 mx = mean(xV);
4 xsd = std(xV);
5 yV = (xV - mx) / xsd;
6 end
```

Κώδικας 1.3: Z-score Transformation

Τα κανονικοποιημένα δεδομένα (εικ. 1.5), έχουν την ίδια ακριβώς γραφική παράσταση μεταξύ τους και σε σύγκριση με τα πρωταρχικά δεδομένα. Η διαφορά τους όμως είναι οι τιμές του άξονα Y. Επίσης, στον γραμμικό μετασχηματισμό, είναι πολύ εύκολο να αναγνωρίσεις την μεγαλύτερη και την μικρότερη τιμή, διότι σίγουρα θα είναι 1 και 0 αντίστοιχα.





Σχήμα 1.5: Μετασχηματισμένα δεδομένα

```

1 data = [
2
3     -0.3999    -0.2625    -1.0106
4         0.069     0.2573     0.6145
5         0.8156    -1.0565     0.5077
6         0.7119    -0.2625    -0.0708
7         0.4376    -0.8051     0.5913
8         0.6686     0.5287    -0.6436
9         1.1908     0.2193     0.3803
10        0.4376    -0.9219    -1.0091
11       -0.0198    -0.2625    -0.0195
12       -0.1567    -0.0592    -0.0482];
13
14 linearT = LinearTransform(data);
15 zscoreT = zscoreTransform(data);
16
17 plot(linearT)
18 hold on;
19 plot(zscoreT)
20 legend('linear','z-score')
21 grid on;

```

Κώδικας 1.4: Script παραδείγματος 1.8 των δεδομένων data

Το script 1.5 χρησιμοποιεί τους μετασχηματισμούς linear και z-score, στις πρώτες τέσσερις στήλες των δεδομένων iris. Με τον z-score μετασχηματισμό, είναι ποίο εύκολο να διακρίνουμε **ακραία δεδομένα**, όπως το δέκατο έκτο (16) για πλάτος πετάλου.

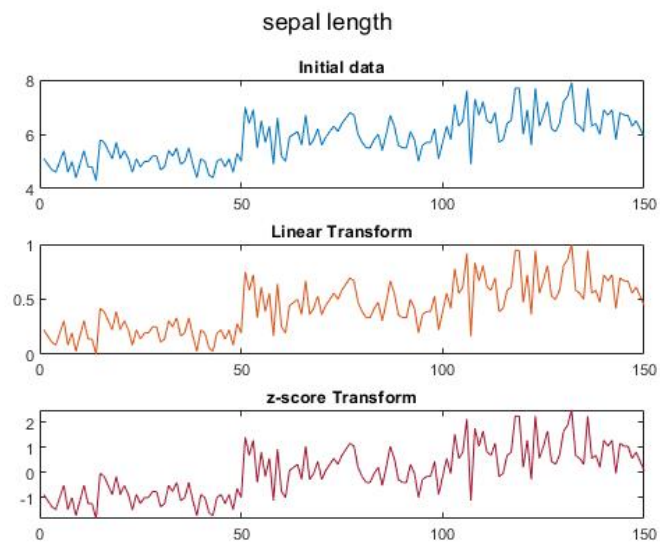
```

1 load iris.dat

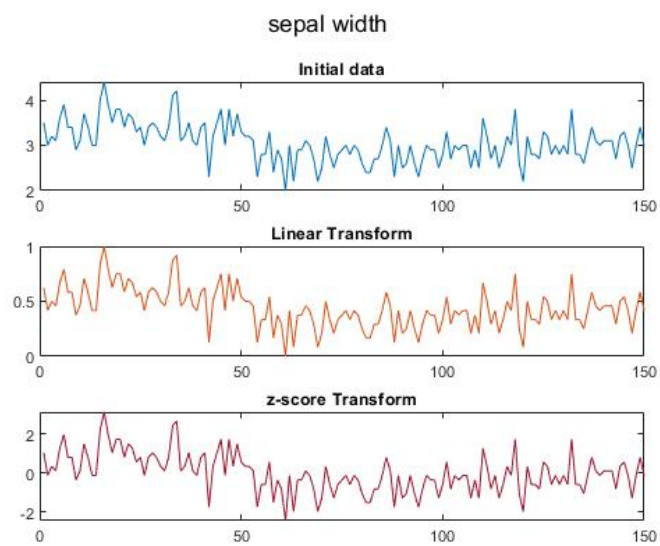
```

```
2
3 linearDataset = [LinearTransform(iris(:,1)),...
4   LinearTransform(iris(:,2)),...
5   LinearTransform(iris(:,3)),...
6   LinearTransform(iris(:,4))];
7
8
9 zscoreDataset = [zscoreTransform(iris(:,1)),...
10  zscoreTransform(iris(:,2)),...
11  zscoreTransform(iris(:,3)),...
12  zscoreTransform(iris(:,4))];
13
14 characteristics = {'sepal length','sepal width',...
15   'petal length', 'petal width'};
16
17 for i=1:4
18   figure();
19   subplot(3,1,1);
20   plot(iris(:,i));
21   title('Initial data');
22
23   subplot(3,1,2);
24   plot(linearDataset(:,i),'color','#D95319');
25   title('Linear Transform');
26
27   subplot(3,1,3);
28   plot(zscoreDataset(:,i),'color','#A2142F');
29   title('z-score Transform');
30
31   sgtitle(characteristics{i})
32 end
```

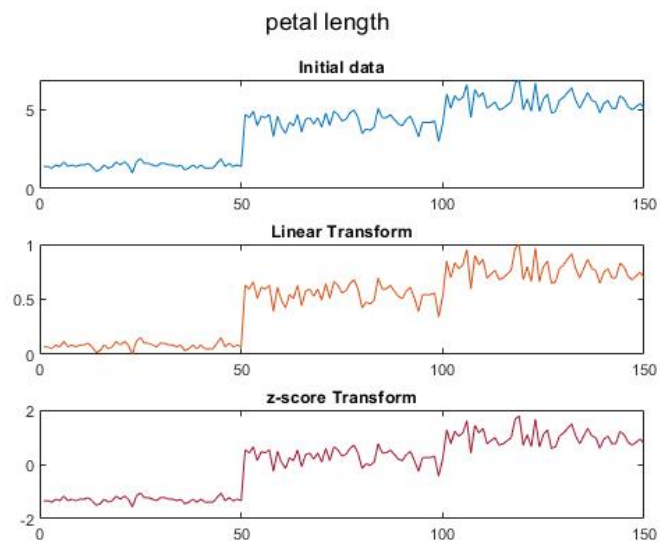
Κώδικας 1.5: Script παραδείγματος 1.8 των δεδομένων iris



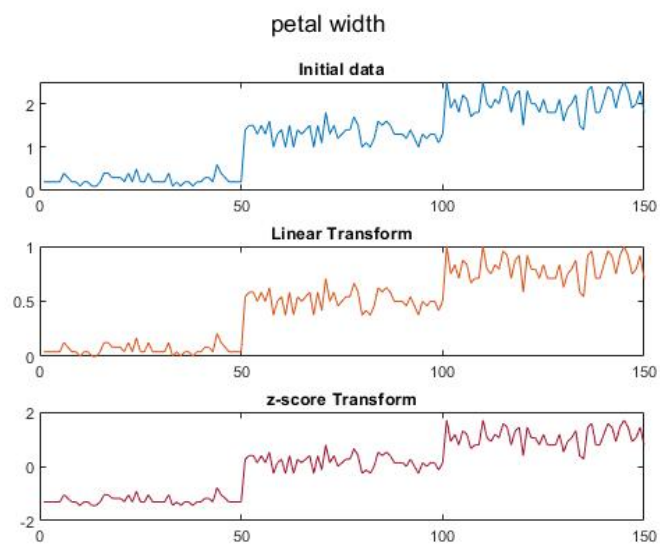
Σχήμα 1.6: Μήκος σέπαλου



Σχήμα 1.7: Πλάτος σέπαλου



Σχήμα 1.8: Μήκος πετάλου



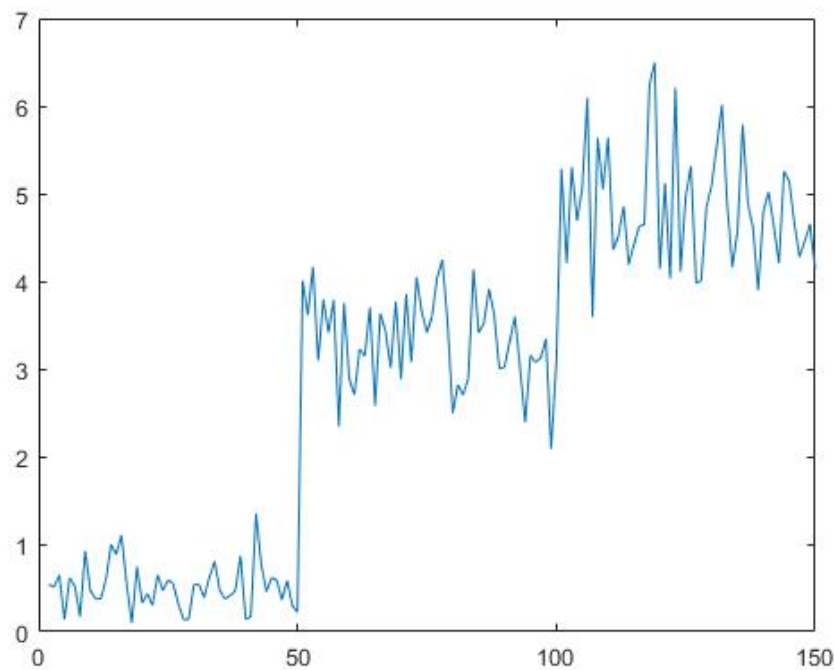
Σχήμα 1.9: Πλάτος πεταλου

## 2 Μέτρα απόστασης

Υπολογισμός των αποστάσεων των δεδομένων Iris για Euclidean, Mahalanobis και City block.

### 2.1 Ευκλείδεια απόσταση και των τεσσάρων στηλών

Με την χρήση του script 2.1, υπολογίζουμε την ευκλείδεια απόσταση και για τα τέσσερα χαρακτηριστικά, εμφανίζουμε την γραφική παράσταση της απόστασης της πρώτης γραμμής σε σύγκριση με τις υπόλοιπες.

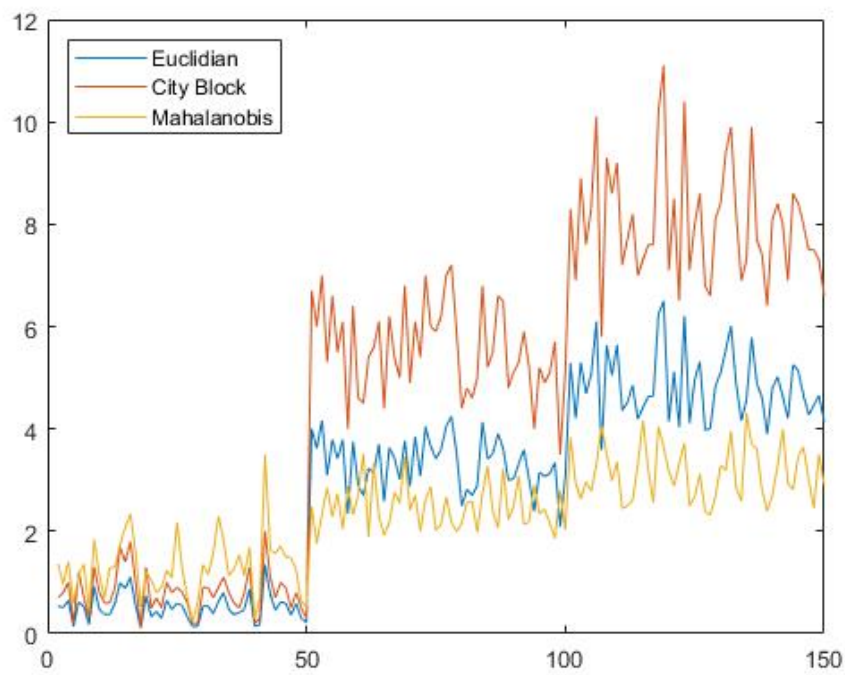


Σχήμα 2.1: Ευκλείδεια απόσταση της πρώτης παρατήρησης

Εν συνεχεία, το script 2.1 υπολογίζει για απόσταση City block και για Mahalanobis.

Από τα σχήματα 2.6 και 2.8, διαπιστώνουμε πως είναι αρκετά μόνο 2 χαρακτηριστικά, έτσι ώστε να βγάλουμε ένα αντίστοιχο πόρισμα. Τα πρώτα 49 στοιχεία είναι σχεδόν εκμηδενισμένα και στα δύο σχήματα για τις μετρικές euclidian και city block, ενώ για τα υπόλοιπα 100 είναι εμφανές η διαφορά μεταξύ τους.

Στο συγκεκριμένο παράδειγμα, η χειρότερη περίπτωση επιλογής αποστάσεων, είναι η Mahalanobis. Ο καλύτερος τρόπος υπολογισμού απόστασης, είναι από την ευκλείδεια. Ωστόσο, η city block είναι και αυτή αρκετά καλή για το συγκεκριμένο παράδειγμα, όμως όχι τόσο όσο η ευκλείδεια. Με την ευκλείδεια, είναι ποίο εμφανείς η διαφορά του τύπου λουλουδιού.



Σχήμα 2.2: Euclidean, City block και Mahalanobis αποστάσεις της πρώτης παρατήρησης

```

1 load iris.dat
2
3 data = iris(:,1:4);
4
5 % 2.α
6 D1 = pdist(data);
7 Z1 = squareform(D1);
8
9 % 2.β
10 figure();
11 plot([2:150],Z1(1,2:150));
12
13 % 2.γ
14 D2 = pdist(data,'cityblock');
15 Z2 = squareform(D2);
16
17 C = cov(data);
18 D3 = pdist(data,'mahalanobis',C);
19 Z3 = squareform(D3);
20
21 figure();
22 plot([2:150],Z1(1,2:150));
23 hold on;

```

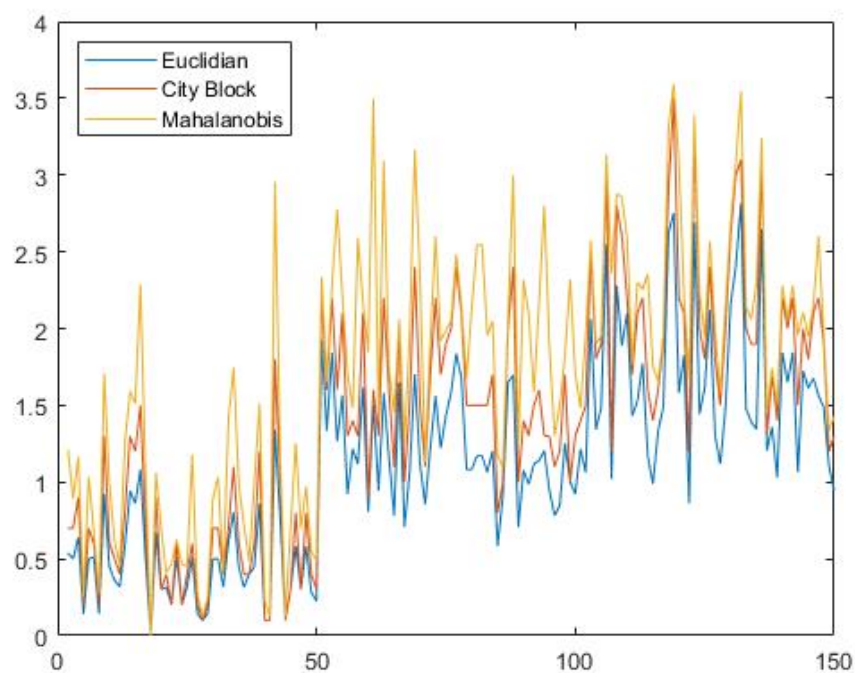
```
24 plot([2:150],Z2(1,2:150));  
25 hold on;  
26 plot([2:150],Z3(1,2:150));  
27 legend('Euclidian','City Block','Mahalanobis',...  
28 'Location','NorthWest');
```

Κώδικας 2.1: Script υπολογισμού αποστάσεων

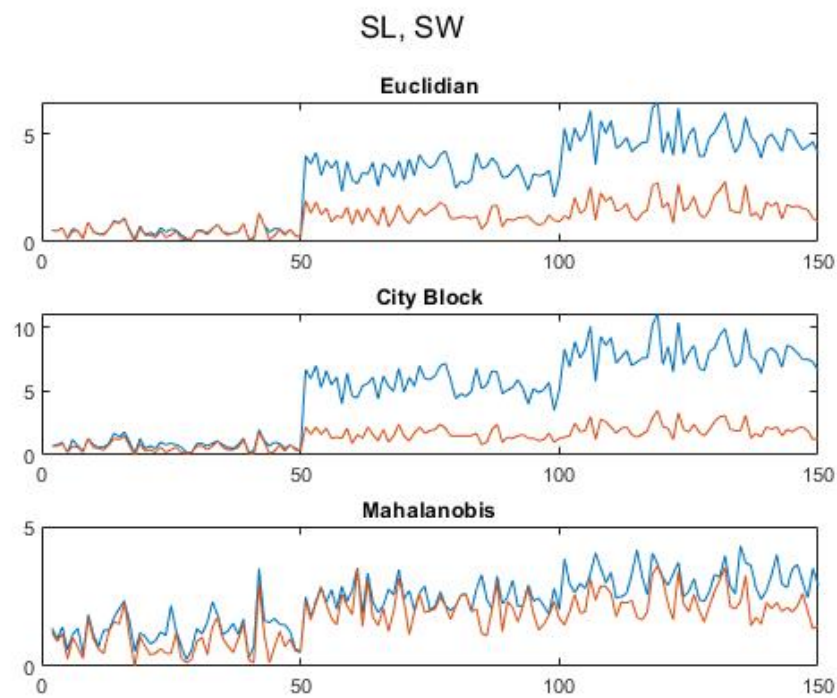
## 2.2 Υπολογισμός αποστάσεων για συνδυασμό στηλών

Με την χρήση του script 2.2 υπολογίζουμε τους συνδυασμούς:

- πρώτο και δεύτερο χαρακτηριστικό
- πρώτο και τρίτο χαρακτηριστικό
- τρίτο και τέταρτο χαρακτηριστικό
- πρώτο, δεύτερο και τρίτο χαρακτηριστικό
- πρώτο, δεύτερο και τέταρτο χαρακτηριστικό

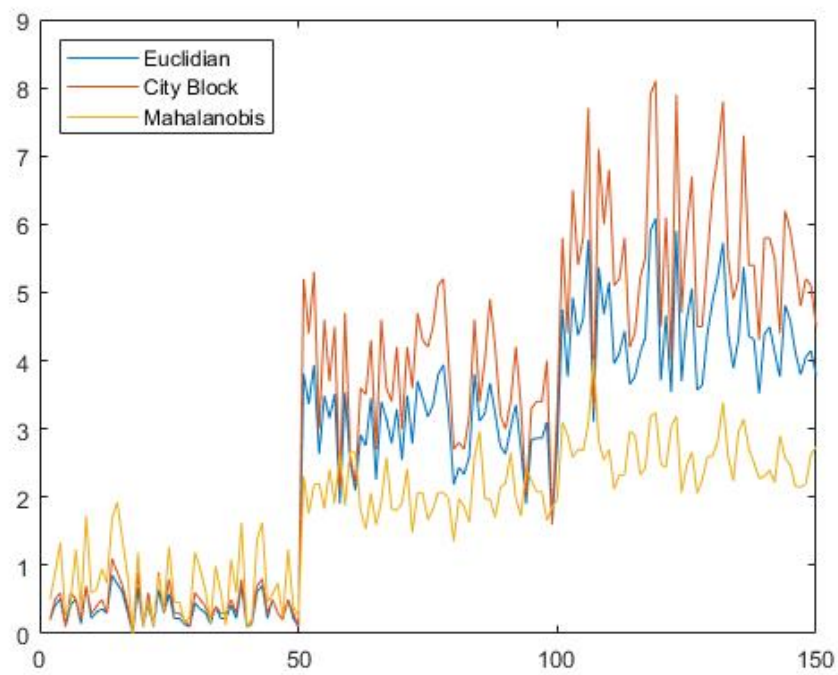


Σχήμα 2.3: Αποστάσεις της πρώτης παρατήρησης για πρώτο και δεύτερο χαρακτηριστικό

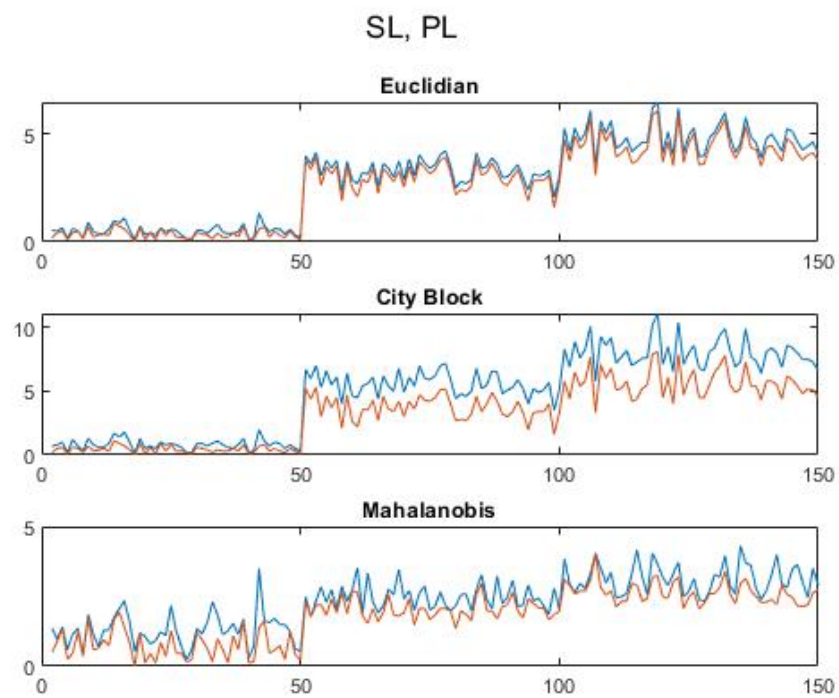


Σχήμα 2.4: Αποστάσεις της πρώτης παρατήρησης για πρώτο και δεύτερο χαρακτηριστικό σε σύγκριση και με τα τέσσερα

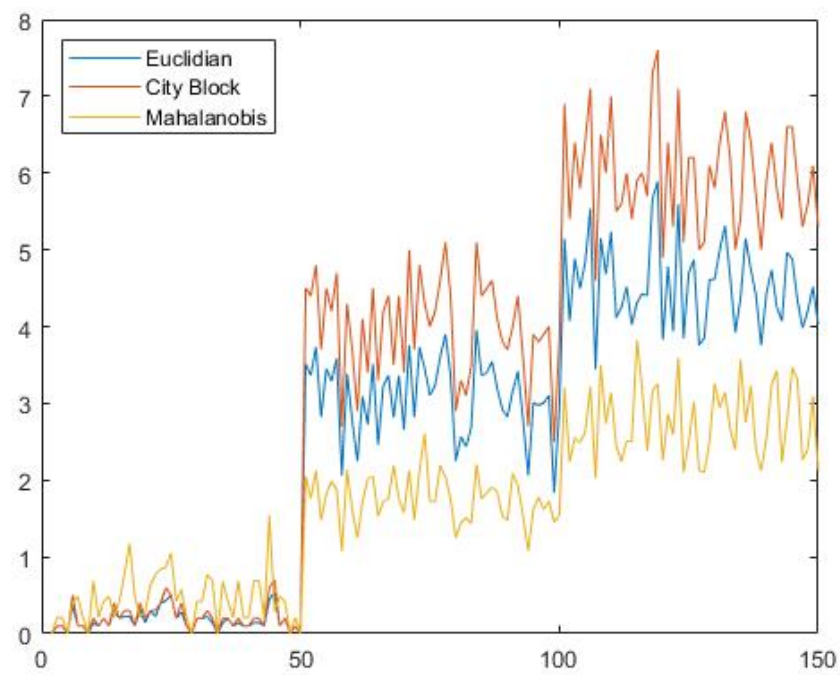




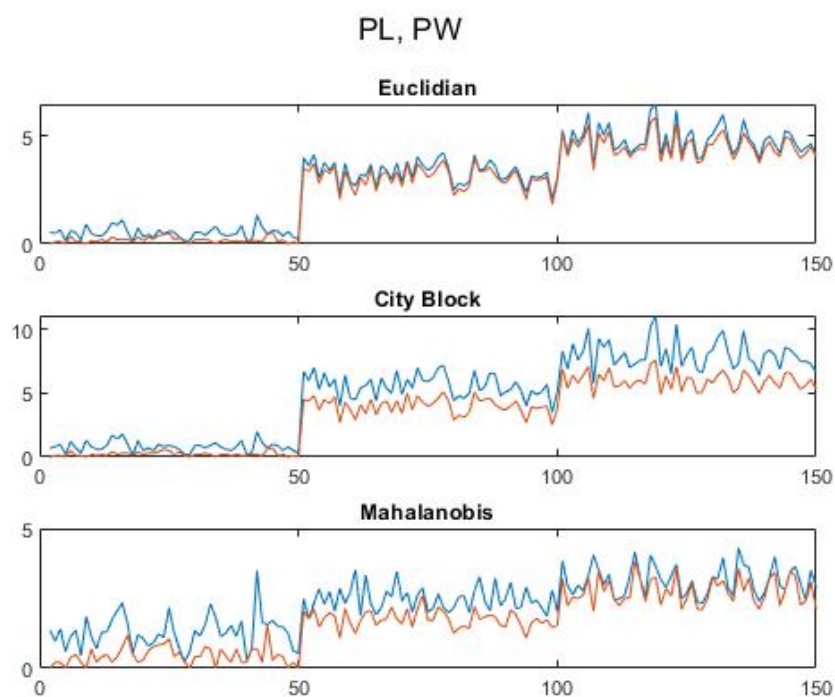
Σχήμα 2.5: Αποστάσεις της πρώτης παρατήρησης για πρώτο και τρίτο χαρακτηριστικό



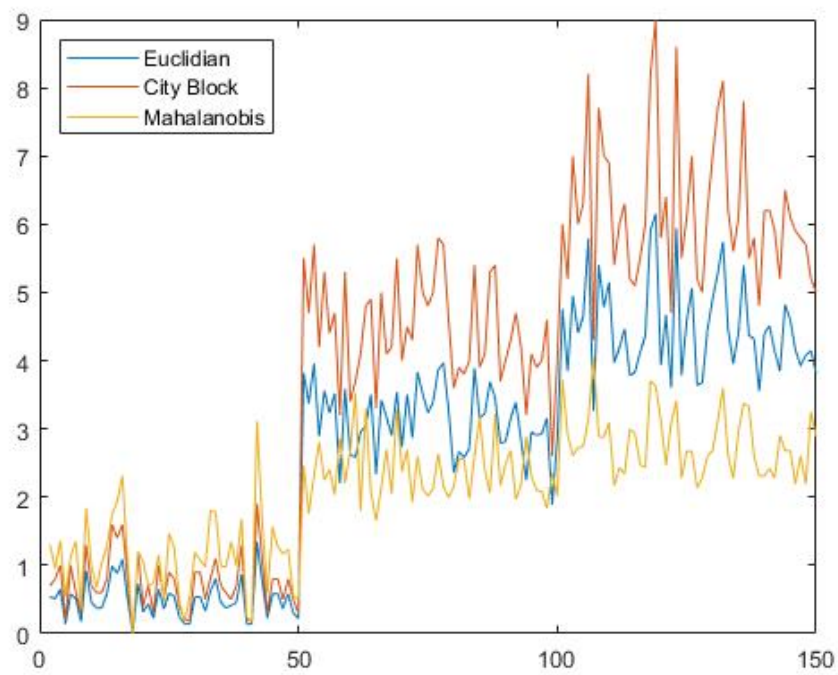
Σχήμα 2.6: Αποστάσεις της πρώτης παρατήρησης για πρώτο και τρίτο χαρακτηριστικό σε σύγκριση και με τα τέσσερα



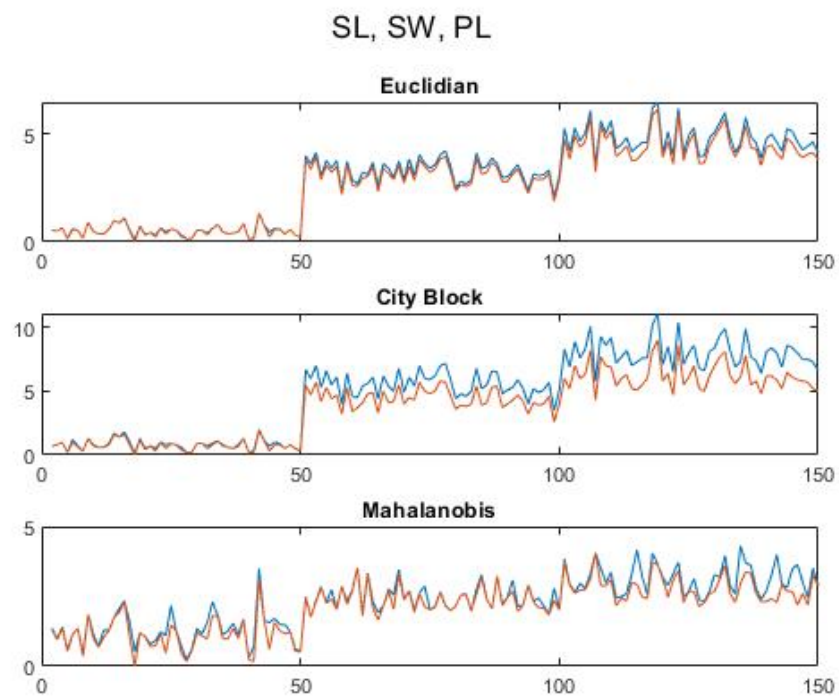
Σχήμα 2.7: Αποστάσεις της πρώτης παρατήρησης για τρίτο και τέταρτο χαρακτηριστικό



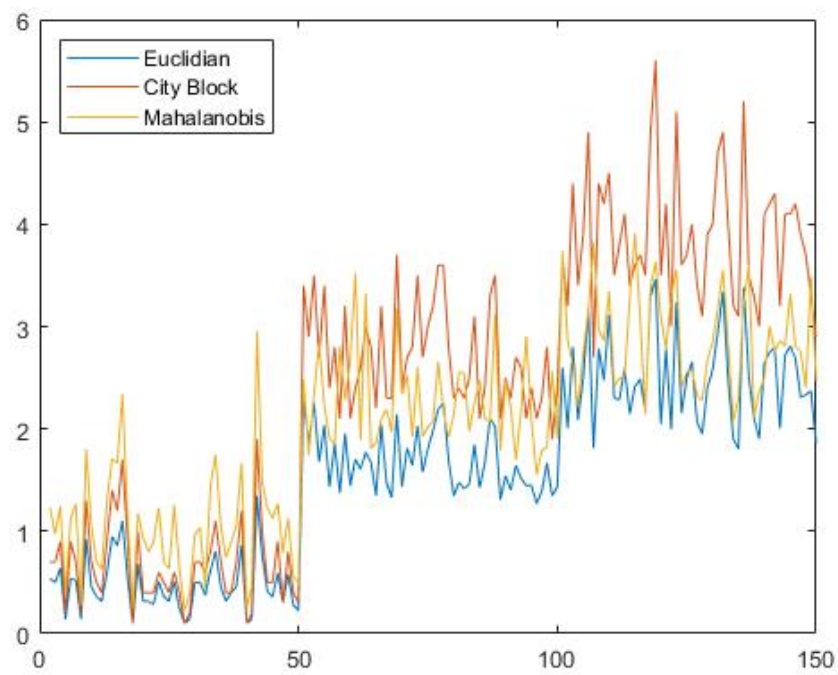
Σχήμα 2.8: Αποστάσεις της πρώτης παρατήρησης για τρίτο και τέταρτο χαρακτηριστικό σε σύγκριση και με τα τέσσερα



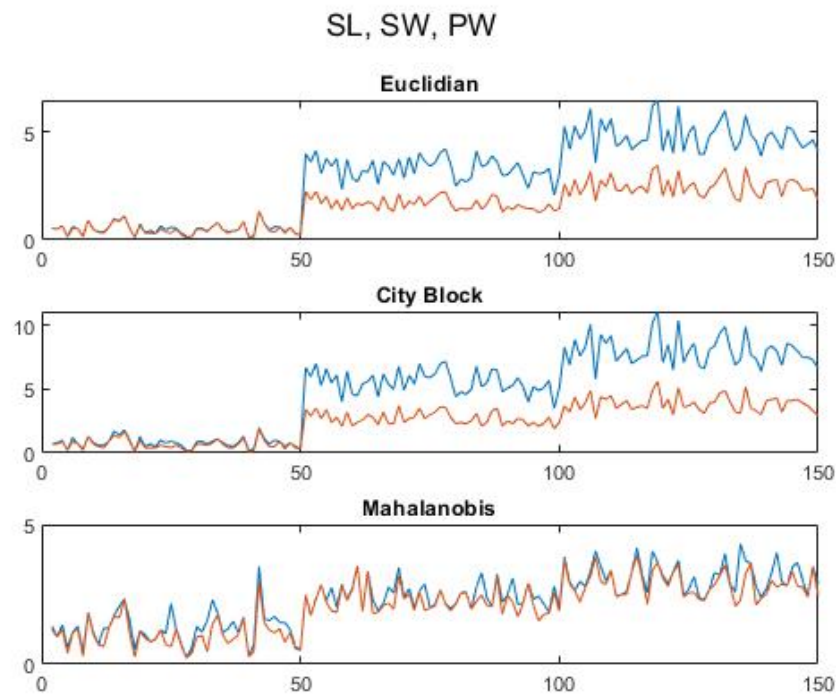
Σχήμα 2.9: Αποστάσεις της πρώτης παρατήρησης για πρώτο, δεύτερο και τρίτο χαρακτηριστικό



Σχήμα 2.10: Αποστάσεις της πρώτης παρατήρησης για πρώτο, δεύτερο και τρίτο χαρακτηριστικό σε σύγκριση και με τα τέσσερα



Σχήμα 2.11: Αποστάσεις της πρώτης παρατήρησης για πρώτο, δεύτερο και τέταρτο χαρακτηριστικό



Σχήμα 2.12: Αποστάσεις της πρώτης παρατήρησης για πρώτο και τέταρτο χαρακτηριστικό σε σύγκριση και με τα τέσσερα

```

1 load iris.dat
2
3 data = iris(:,1:4);
4
5 D1 = pdist(data);
6 Z1 = squareform(D1);
7
8 D2 = pdist(data,'cityblock');
9 Z2 = squareform(D2);
10
11 C = cov(data);
12 D3 = pdist(data,'mahalanobis',C);
13 Z3 = squareform(D3);
14
15 % 2.δ
16
17 data = iris(:,1:2);
18
19 D11 = pdist(data);
20 Z11 = squareform(D11);
21

```



```
22 D22 = pdist(data,'cityblock');
23 Z22 = squareform(D22);
24
25 C = cov(data);
26 D33 = pdist(data,'mahalanobis',C);
27 Z33 = squareform(D33);
28
29 figure();
30 plot([2:150],Z11(1,2:150));
31 hold on;
32 plot([2:150],Z22(1,2:150));
33 hold on;
34 plot([2:150],Z33(1,2:150));
35 legend('Euclidian','City Block','Mahalanobis',...
36        'Location','NorthWest');
37
38 figure();
39 subplot(3,1,1);
40 plot([2:150],Z1(1,2:150));
41 hold on;
42 plot([2:150],Z11(1,2:150));
43 title('Euclidian');
44
45 subplot(3,1,2);
46 plot([2:150],Z2(1,2:150));
47 hold on;
48 plot([2:150],Z22(1,2:150));
49 title('City Block');
50
51 subplot(3,1,3);
52 plot([2:150],Z3(1,2:150));
53 hold on;
54 plot([2:150],Z33(1,2:150));
55 title('Mahalanobis');
56
57 sgtitle('SL, SW');
58
59 % 2.ε
60
61 % 1 3
62 data = [iris(:,1),iris(:,3)];
63
64 D11 = pdist(data);
65 Z11 = squareform(D11);
66
67 D22 = pdist(data,'cityblock');
68 Z22 = squareform(D22);
69
70 C = cov(data);
```

```
71 D33 = pdist(data,'mahalanobis',C);
72 Z33 = squareform(D33);
73
74 figure();
75 plot([2:150],Z11(1,2:150));
76 hold on;
77 plot([2:150],Z22(1,2:150));
78 hold on;
79 plot([2:150],Z33(1,2:150));
80 legend('Euclidian','City Block','Mahalanobis',...
81       'Location','NorthWest');
82
83 figure();
84 subplot(3,1,1);
85 plot([2:150],Z1(1,2:150));
86 hold on;
87 plot([2:150],Z11(1,2:150));
88 title('Euclidian');
89
90 subplot(3,1,2);
91 plot([2:150],Z2(1,2:150));
92 hold on;
93 plot([2:150],Z22(1,2:150));
94 title('City Block');
95
96 subplot(3,1,3);
97 plot([2:150],Z3(1,2:150));
98 hold on;
99 plot([2:150],Z33(1,2:150));
100 title('Mahalanobis');
101
102 sgtitle('SL, PL');
103
104
105 % 3 4
106 data = iris(:,3:4);
107
108 D11 = pdist(data);
109 Z11 = squareform(D11);
110
111 D22 = pdist(data,'cityblock');
112 Z22 = squareform(D22);
113
114 C = cov(data);
115 D33 = pdist(data,'mahalanobis',C);
116 Z33 = squareform(D33);
117
118 figure();
119 plot([2:150],Z11(1,2:150));
```

```

120 hold on;
121 plot([2:150],Z22(1,2:150));
122 hold on;
123 plot([2:150],Z33(1,2:150));
124 legend('Euclidian','City Block','Mahalanobis',...
125 'Location','NorthWest');
126
127 figure();
128 subplot(3,1,1);
129 plot([2:150],Z1(1,2:150));
130 hold on;
131 plot([2:150],Z11(1,2:150));
132 title('Euclidian');
133
134 subplot(3,1,2);
135 plot([2:150],Z2(1,2:150));
136 hold on;
137 plot([2:150],Z22(1,2:150));
138 title('City Block');
139
140 subplot(3,1,3);
141 plot([2:150],Z3(1,2:150));
142 hold on;
143 plot([2:150],Z33(1,2:150));
144 title('Mahalanobis');
145
146 sgtitle('PL, PW');
147
148 % 1 2 3
149 data = iris(:,1:3);
150
151 D11 = pdist(data);
152 Z11 = squareform(D11);
153
154 D22 = pdist(data,'cityblock');
155 Z22 = squareform(D22);
156
157 C = cov(data);
158 D33 = pdist(data,'mahalanobis',C);
159 Z33 = squareform(D33);
160
161 figure();
162 plot([2:150],Z11(1,2:150));
163 hold on;
164 plot([2:150],Z22(1,2:150));
165 hold on;
166 plot([2:150],Z33(1,2:150));
167 legend('Euclidian','City Block','Mahalanobis',...
168 'Location','NorthWest');

```

```
169
170 figure();
171 subplot(3,1,1);
172 plot([2:150],Z1(1,2:150));
173 hold on;
174 plot([2:150],Z11(1,2:150));
175 title('Euclidian');
176
177 subplot(3,1,2);
178 plot([2:150],Z2(1,2:150));
179 hold on;
180 plot([2:150],Z22(1,2:150));
181 title('City Block');
182
183 subplot(3,1,3);
184 plot([2:150],Z3(1,2:150));
185 hold on;
186 plot([2:150],Z33(1,2:150));
187 title('Mahalanobis');
188
189 sgtitle('SL, SW, PL');
190
191 % 1 2 4
192 data = [iris(:,1:2),iris(:,4)];
193
194 D11 = pdist(data);
195 Z11 = squareform(D11);
196
197 D22 = pdist(data,'cityblock');
198 Z22 = squareform(D22);
199
200 C = cov(data);
201 D33 = pdist(data,'mahalanobis',C);
202 Z33 = squareform(D33);
203
204 figure();
205 plot([2:150],Z11(1,2:150));
206 hold on;
207 plot([2:150],Z22(1,2:150));
208 hold on;
209 plot([2:150],Z33(1,2:150));
210 legend('Euclidian','City Block','Mahalanobis',...
211        'Location','NorthWest');
212
213 figure();
214 subplot(3,1,1);
215 plot([2:150],Z1(1,2:150));
216 hold on;
217 plot([2:150],Z11(1,2:150));
```

```
218 title('Euclidian');
219
220 subplot(3,1,2);
221 plot([2:150],Z2(1,2:150));
222 hold on;
223 plot([2:150],Z22(1,2:150));
224 title('City Block');
225
226 subplot(3,1,3);
227 plot([2:150],Z3(1,2:150));
228 hold on;
229 plot([2:150],Z33(1,2:150));
230 title('Mahalanobis');
231
232 sgtitle('SL, SW, PW');
```

Κώδικας 2.2: Script υπολογισμού αποστάσεων για διάφορους συνδυασμούς χαρακτηριστικών