

به نام خدا

تمرین دوم

مبانی داده کاوی

ماتریس عدم شباهت - فاصله

و پیش پردازش/نرمال سازی

نام و نام خانوادگی: علی رضائی نژاد

شماره دانشجویی: ۹۶۰۱۸۴۱۵۶

مشخصه درس: ۹۱۳۵۱

نام استاد: خانم امینه امینی

Euclidean Distance Matrix

ماتریس فاصله اقلیدسی

در R با استفاده از تابع از پیش تعریف شده‌ی `dist (stats::dist)` می‌توان انواع فواصل را به صورت زیر بدست آورد.

```
dist(Dataset, method = "euclidean", diag = TRUE,  
upper = FALSE, p = 2)
```

```
stats::dist(Dataset, method = "euclidean", ...)
```

 یا:

”**method**” بیانگر نوع فاصله‌ایست که نیاز به محاسبه‌ی آن داریم، به عنوان مثال می‌توان آن را معادل `manhattan` یا `minkowski` قرار داد.

با ”**diag**” می‌توان وضعیت چاپ قطر (که مقداری همواره برابر با صفر دارد) را تعیین کرد. همانطور که در تصویر صفحه‌ی بعد مشاهده می‌شود، `TRUE` به معنی چاپ قطر می‌باشد.

به صورت مشابه، ”**upper**” تعیین‌گر چاپ یا عدم چاپ مثلث بالای قطر در خروجی است. به دلیل پرهیز از مقادیر تکراری آن را معادل `FALSE` قرار دادم.

”**p**” نشان‌دهنده‌ی توان (**Power**) فاصله‌ی `minkowski` است (معادل `h` در فرمول زیر). نیازی به تعیین آن نبود زیرا با تعیین فاصله اقلیدسی که خود بیانگر فاصله‌ی `minkowski` با توان دو می‌باشد، آن را مشخص کرده بودیم.

$$d(i, j) = \sqrt[h]{|x_{i1} - x_{j1}|^h + |x_{i2} - x_{j2}|^h + \dots + |x_{ip} - x_{jp}|^h}$$

```
In [1]: seeds=read.csv("D:\\Daneshga\\T8\\Amini\\Datasets\\Iris Alternatives\\Wp
library(philentropy)
```

```
In [19]: dist(seeds, method = "euclidean", diag = TRUE, upper = FALSE, p = 2)
#distance(seeds, method = "euclidean")
#stats::dist(seeds, method = "euclidean")
#getDistMethods()
```

	1	2	3	4	5	6
1	0.000000					
2	1.666862	0.000000				
3	2.472342	2.119240	0.000000			
4	3.492546	2.663945	1.192453	0.000000		
5	4.198030	3.315009	3.216621	2.914056	0.000000	
6	5.142158	4.300418	3.017545	2.106382	2.473047	0.000000
7	6.193021	5.631328	4.164457	3.468298	3.381299	1.594181
8	7.160034	6.297926	5.008112	4.043040	3.984601	2.035891
9	8.175159	7.417128	6.744734	6.068848	4.193958	4.137637
10	9.098030	8.262679	7.508823	6.739415	5.071225	4.710253
11	10.266836	9.685587	8.327494	7.582333	6.861730	5.545797
12	11.108344	10.070358	9.061377	8.028799	7.381166	6.057831
13	12.248056	11.452964	10.093071	9.167421	8.791843	7.187623
14	13.147705	12.247451	11.023753	10.043195	9.533934	8.054638
15	14.132028	13.202414	12.018550	11.025927	10.463534	9.040028
16	15.165207	14.362112	13.089686	12.181585	11.498584	10.151356
17	16.380656	15.631318	14.234382	13.338979	12.864399	11.361689
18	17.020807	16.033348	15.124201	14.164456	13.013675	12.116987

به دلیل بالا بودن شمار سطر و ستون‌های ماتریس (۲۱۰ در ۲۱۰) این تابع هر ۶ ستون را پیش رفته و ماتریس را تا آن مرحله چاپ می‌کند، ادامه ماتریس (۶ ستون بعدی) در پایان دویست و دهمین ردیف قابل مشاهده است. که به دلیل طول بسیار زیاد این خروجی، تصویر کامل آن را قرار ندادم.

```

In [1]: seeds=read.csv("D:\\Daneshga\\T8\\Amini\\Datasets\\Iris Alternatives\\Wheat Seeds Dataset\\Final Refi
library(philentropy)

In [21]: dist(seeds$Area, method = "euclidean", diag = TRUE, upper = FALSE, p = 2)
#distance(seeds, method = "euclidean")
#stats::dist(seeds, method = "euclidean")
#getDistMethods()

```

	1	2	3	4	5	6	7	8	9	10	11	12
1	0.00											
2	0.38	0.00										
3	0.97	0.59	0.00									
4	1.42	1.04	0.45	0.00								
5	0.88	1.26	1.85	2.30	0.00							
6	0.88	0.50	0.09	0.54	1.76	0.00						
7	0.57	0.19	0.40	0.85	1.45	0.31	0.00					
8	1.15	0.77	0.18	0.27	2.03	0.27	0.58	0.00				
9	1.37	1.75	2.34	2.79	0.49	2.25	1.94	2.52	0.00			
10	1.18	1.56	2.15	2.60	0.30	2.06	1.75	2.33	0.19	0.00		
11	0.00	0.38	0.97	1.42	0.88	0.88	0.57	1.15	1.37	1.18	0.00	
12	1.23	0.85	0.26	0.19	2.11	0.35	0.66	0.08	2.60	2.41	1.23	0.00
13	1.37	0.99	0.40	0.05	2.25	0.49	0.80	0.22	2.74	2.55	1.37	0.14
14	1.48	1.10	0.51	0.06	2.36	0.60	0.91	0.33	2.85	2.66	1.48	0.25
15	1.52	1.14	0.55	0.10	2.40	0.64	0.95	0.37	2.89	2.70	1.52	0.29
16	0.67	0.29	0.30	0.75	1.55	0.21	0.10	0.48	2.04	1.85	0.67	0.56
17	1.27	0.89	0.30	0.15	2.15	0.39	0.70	0.12	2.64	2.45	1.27	0.04
18	0.43	0.81	1.40	1.85	0.45	1.31	1.00	1.58	0.94	0.75	0.43	1.66

از طرفی با استفاده از کتابخانه‌ی **philentropy** علاوه بر خروجی خلاصه شده و مرتب‌تر، توانایی محاسبه ده‌ها نوع فاصله دیگر نیز در اختیار ما قرار می‌گیرد.

با فراخوانی تابع **getDistMethods()** انواع فواصلی که این کتابخانه برای محاسبه آنان تابع تعریف کرده را می‌توان مشاهده کرد.

```

In [1]: seeds=read.csv("D:\\Daneshga\\T8\\Amini\\Datasets\\Iris Alternatives\\Wheat Seeds Dataset\\Final Refined CSV.csv")
library(philentropy)

In [20]: #dist(seeds, method = "euclidean", diag = TRUE, upper = FALSE, p = 2)
#distance(seeds, method = "euclidean")
#stats::dist(seeds, method = "euclidean")
getDistMethods()

```

'euclidean' · 'manhattan' · 'minkowski' · 'chebyshev' · 'sorensen' · 'gower' · 'soergel' · 'kulczynski_d' · 'canberra' · 'lorentzian' · 'intersection' · 'non-intersection' · 'wavehedges' · 'czekanowski' · 'motyka' · 'kulczynski_s' · 'tanimoto' · 'ruzicka' · 'inner_product' · 'harmonic_mean' · 'cosine' · 'hassebrook' · 'jaccard' · 'dice' · 'fidelity' · 'battacharyya' · 'hellinger' · 'matusita' · 'squared_chord' · 'squared_euclidean' · 'pearson' · 'neyman' · 'squared_chi' · 'prob_symm' · 'divergence' · 'clark' · 'additive_symm' · 'kullback-leibler' · 'jeffreys' · 'k_divergence' · 'topsoe' · 'jensen-shannon' · 'jensen_difference' · 'taneja' · 'kumar-johnson' · 'avg'

با اینکود کردن این کتابخانه میتوان از تابع `distance` آن به صورت زیر استفاده کرد.

```
library(philentropy)
distance(Dataset, method = "euclidean")
```

دایکومنتیشن این کتابخانه از لینک زیر قابل مطالعه است (تابع `distance` در صفحه ۱۲ شرح داده شده است).

[philentropy.pdf \(r-project.org\)](https://r-project.org/doc/philentropy.pdf)

مثالی با استفاده از تمام آرگومان‌های این تابع:

```
distance( Dataset,
method = "euclidean",
p = NULL,
test.na = TRUE,
unit = "log",
est.prob = NULL,
use.row.names = FALSE,
as.dist.obj = FALSE,
diag = FALSE,
upper = FALSE,
mute.message = FALSE )
```

همانطور که ملاحظه می‌شود، آرگومان‌های `diag`، `upper` و `p` نیز در این تابع تحت همان عنوان، قابل مشخص کردن هستند.

```
In [1]: seeds=read.csv("D:\\Daneshga\\T8\\Amini\\Datasets\\Iris Alternatives\\Wheat Seeds Dataset\\Final Refined CSV.csv")
library(phyloentropy)
```

```
In [27]: distance(seeds, method = "euclidean")
```

Metric: 'euclidean'; comparing: 210 vectors.

A matrix: 210 × 210 of type dbl

	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	...	v201	v202	v203
v1	0.000000	1.666862	2.472342	3.492546	4.198030	5.142158	6.193021	7.160034	8.175159	9.098030	...	200.0631	201.0345	202.0729
v2	1.666862	0.000000	2.119240	2.663945	3.315009	4.300418	5.631328	6.297926	7.417128	8.262679	...	199.0794	200.0312	201.0768
v3	2.472342	2.119240	0.000000	1.192453	3.216621	3.017545	4.164457	5.008112	6.744734	7.508823	...	198.0402	199.0189	200.0443
v4	3.492546	2.663945	1.192453	0.000000	2.914056	2.106382	3.468298	4.043040	6.068848	6.739415	...	197.0430	198.0150	199.0407
v5	4.198030	3.315009	3.216621	2.914056	0.000000	2.473047	3.381299	3.984601	4.193958	5.071225	...	196.0970	197.0521	198.1064
v6	5.142158	4.300418	3.017545	2.106382	2.473047	0.000000	1.594181	2.035891	4.137637	4.710253	...	195.0456	196.0204	197.0492
v7	6.193021	5.631328	4.164457	3.468298	3.381299	1.594181	0.000000	1.530974	3.436589	3.939163	...	194.0366	195.0299	196.0519
v8	7.160034	6.297926	5.008112	4.043040	3.984601	2.035891	1.530974	0.000000	3.291239	3.438948	...	193.0395	194.0184	195.0430
v9	8.175159	7.417128	6.744734	6.068848	4.193958	4.137637	3.436589	3.291239	0.000000	1.110817	...	192.1029	193.0696	194.1233
v10	9.098030	8.262679	7.508823	6.739415	5.071225	4.710253	3.939163	3.438948	1.110817	0.000000	...	191.0968	192.0615	193.1143
v11	10.266836	9.685587	8.327494	7.582333	6.861730	5.545797	4.171639	3.804043	3.604949	3.053563	...	190.0409	191.0496	192.0686
v12	11.108344	10.070358	9.061377	8.028799	7.381166	6.057831	5.394828	4.121522	4.332395	3.410884	...	189.0567	190.0189	191.0524
v13	12.248056	11.452964	10.093071	8.791843	7.187623	6.105023	5.175734	5.577936	4.691202	188.0240	189.0239	190.0352
v14	13.147705	12.247451	11.023753	10.043195	9.533934	8.054638	7.096026	6.028433	6.141555	5.155238	...	187.0319	188.0179	189.0363
v15	14.132028	13.202414	12.018550	11.025927	10.463534	9.040028	8.105965	7.018783	6.977952	5.955947	...	186.0344	187.0169	188.0369
v16	15.165207	14.362112	13.089686	12.181585	11.498584	10.151356	9.036346	8.156088	7.801735	6.793626	...	185.0310	186.0332	187.0494
v17	16.380656	15.631318	14.234382	13.338979	12.864399	11.361689	10.200360	9.362389	9.268783	8.297328	...	184.0191	185.0397	186.0400
v18	17.020807	16.033348	15.124201	14.164456	13.013675	12.116987	11.229991	10.207342	9.140721	8.081733	...	183.0875	184.0440	185.0953
v19	18.043827	17.028151	16.034533	15.036656	14.118888	13.028298	12.160774	11.064700	10.372592	9.294879	...	182.0662	183.0256	184.0661
v20	19.314602	18.421911	17.140599	16.152989	15.709167	14.210983	13.200166	12.177362	12.078760	11.055991	...	181.0167	182.0203	183.0197
v21	20.054287	19.127665	18.015423	17.037841	16.229791	15.020447	14.020750	13.013435	12.371342	11.333944	...	180.0391	181.0238	182.0475
v22	21.046538	20.089189	19.007860	18.017054	17.193251	16.007338	15.040143	14.003643	13.342248	12.299222	...	179.0435	180.0212	181.0479
v23	22.058652	21.029165	20.177482	19.196342	18.012090	17.167285	16.297943	15.251436	14.117749	13.082497	...	178.1150	179.0553	180.1182
v24	23.302436	22.229667	21.177583	20.114326	19.527835	18.209097	17.392507	16.210125	15.908963	14.854782	...	177.0582	178.0147	179.0337
v25	24.006394	23.015555	22.047239	21.060242	20.041688	19.034690	18.096859	17.064933	16.126436	15.088964	...	176.0756	177.0354	178.0808
v26	25.054520	24.047583	23.184350	22.211767	21.007236	20.172855	19.261472	18.247143	17.057307	16.039567	...	175.1228	176.0649	177.1306
v27	26.151183	25.195212	24.047563	23.045104	22.356210	21.070881	20.089927	19.049010	18.531905	17.488800	...	174.0259	175.0162	176.0259
v28	27.151150	26.149061	25.055482	24.031906	23.327127	22.070769	21.139742	20.055895	19.525879	18.478502	...	173.0377	174.0126	175.0283
v29	28.038063	27.069875	26.003180	25.010039	24.142978	23.004310	22.026335	21.000763	20.236706	19.199828	...	172.0438	173.0215	174.0485
v30	29.099860	28.155926	27.029675	26.038403	25.262365	24.044599	23.039458	22.027444	21.365066	20.332553	...	171.0273	172.0204	173.0333
...
v181	180.0855	179.0970	178.0535	177.0525	176.1266	175.0601	174.0564	173.0524	172.1411	171.1321	...	20.038889	21.21627	22.021230
v182	181.0605	180.0755	179.0370	178.0390	177.0952	176.0422	175.0351	174.0361	173.1031	172.0963	...	19.006688	20.186925	21.075738
v183	182.0632	181.0767	180.0386	179.0396	178.0986	177.0437	176.0380	175.0374	174.1079	173.1008	...	18.012228	19.184864	20.056209
v184	183.0812	182.0953	181.0516	180.0519	179.1215	178.0579	177.0516	176.0504	175.1325	174.1246	...	17.022888	18.270618	19.047050
v185	184.0724	183.0970	182.0511	181.0573	180.1103	179.0573	178.0410	177.0506	176.1103	175.1057	...	16.032105	17.462676	18.251075
v186	185.0648	184.0717	183.0396	182.0370	181.0994	180.0438	179.0441	178.0377	177.1129	176.1052	...	15.093450	16.151837	17.022388
v187	186.0685	185.0819	184.0437	183.0439	182.1057	181.0487	180.0429	179.0419	178.1143	177.1077	...	14.030773	15.272204	16.065371
v188	187.0851	186.0901	185.0535	184.0493	183.1237	182.0590	181.0611	180.0517	179.1418	178.1323	...	13.166875	14.253745	15.004582
v189	188.1415	187.1689	186.1037	185.1086	184.1950	183.1142	182.0955	181.1037	180.2010	179.1936	...	12.246243	14.098575	14.425584
v190	189.1071	188.1156	187.0699	186.0667	185.1510	184.0773	183.0762	182.0687	181.1694	180.1593	...	11.212414	12.507675	13.050355
v191	190.0997	189.1126	188.0658	187.0647	186.1435	185.0729	184.0676	183.0644	182.1568	181.1482	...	10.130250	11.583075	12.080615
v192	191.0725	190.0770	189.0440	188.0404	187.1072	186.0488	185.0513	184.0424	183.1240	182.1151	...	9.210442	10.256661	11.003131
v193	192.0553	191.0588	190.0318	189.0287	188.0845	187.0354	186.0387	185.0303	184.0997	183.0916	...	8.244326	9.144252	10.044916
v194	193.0920	192.1012	191.0598	190.0572	189.1331	188.0660	187.0639	186.0580	185.1477	184.1391	...	7.233426	8.636978	9.050771
v195	194.0528	193.0603	192.0303	191.0292	190.0828	189.0343	188.0337	187.0291	186.0948	185.0873	...	6.156215	7.268690	8.080012
v196	195.0499	194.0634	193.0295	192.0319	191.0794	190.0341	189.0279	188.0291	187.0869	186.0804	...	5.057286	6.536014	7.279626
v197	196.0584	195.0758	194.0372	193.0409	192.0908	191.0424	190.0325	189.0368	188.0958	187.0898	...	4.023488	5.941371	6.435961
v198	197.0379	196.0519	195.0229	194.0263	193.0633	192.0264	191.0197	190.0226	189.0678	188.0626	...	3.281938	4.748824	5.618733
v199	198.0348	197.0392	196.0196	195.0181	194.0578	193.0216	192.0239	191.0183	190.0685	189.0623	...	2.979829	3.256892	4.457009
v200	199.0337	198.0340	197.0176	196.0151	195.0534	194.0195	193.0257	192.0170	191.0683	190.0607	...	2.869685	2.075098	3.681258
v201	200.0631	199.0794	198.0402	197.0430	196.0970	195.0456	194.0366	193.0395	192.1029	191.0968	...	0.000000	3.365706	2.846212
v202	201.0345	200.0312	199.0189	198.0150	197.0521	196.0204	195.0299	194.0184	193.0696	192.0615	...	3.365706	0.000000	2.600023
v203	202.0729	201.0768	200.0443	199.0407	198.1064	197.0492	196.0519	195.0430	194.1233	193.1143	...	2.846212	2.600023	0.000000
v204	203.1278	202.1624	201.1001	200.1103	199.1739	198.1098	197.0839	196.1012	195.1704	194.1659	...	4.245231	6.481667	4.830085
v205	204.0431	203.0497	202.0241	201.0233	200.0692	199.0273	198.0273	197.0230	196.0800	195.0733	...	4.291367	3.446724	2.467105
v206	205.0457	204.0496	203.0252	202.0231	201.0714	200.0284	199.0309	198.0241	197.0849	196.0773	...	5.340477	4.251998	3.236417
v207	206.0709	205.0770	204.0437	203.0409	202.1046	201.0485	200.0489	199.0421	198.1189	197.1108	...	6.244794	5.619297	4.018889
v208	207.1137	206.1480	205.0902	204.1011	203.1555	202.0990	201.0731	200.0915	199.1501	198.1465	...	7.606817	8.529153	6.959002
v209	208.0501	207.0535	206.0289	205.0261	204.0771	203.0321	202.0350	201.0274	200.0907	199.0833	...	8.239041	7.183534	6.079149
v210	209.0648	208.0811	207.0419	206.0446	205.0984	204.0473	203.0380	202.0411	201.1038	200.0979	...	9.002489	8.687224	7.300074

Manhattan Distance Matrix

ماتریس فاصله منهتن

```
In [1]: seeds=read.csv("D:\\Daneshga\\T8\\Amini\\Datasets\\Iris Alternatives\\wheat See  
library(philentropy)
```

```
In [33]: #distance(seeds, method = "manhattan")  
dist(seeds, method = "manhattan", diag = TRUE, upper = FALSE)
```

	1	2	3	4	5	6	7	8
1	0.0000							
2	3.3571	0.0000						
3	5.1240	4.1729	0.0000					
4	6.3035	5.3524	2.1445	0.0000				
5	6.3284	5.5913	7.0376	6.1489	0.0000			
6	7.4161	6.5070	3.7079	3.2934	5.3963	0.0000		
7	8.5479	8.1852	6.4561	6.5156	6.6465	3.2222	0.0000	
8	9.9621	9.1410	5.5439	5.2434	7.9503	2.7100	3.2722	0.0000
9	11.2747	12.2204	12.3413	11.4368	6.8677	8.6834	7.8152	7.2294
10	11.4860	12.2769	12.5260	11.6215	6.8304	8.8681	8.0161	7.4141
11	12.5464	13.8055	12.6164	12.6759	10.7568	9.3825	6.1603	7.4325
12	14.0776	12.2535	10.7964	9.4559	11.0808	7.3685	8.2603	5.2545
13	16.8910	15.9819	12.1470	11.2265	14.8854	9.5951	8.3431	6.9731
14	16.9679	16.0692	12.4221	11.5216	14.9865	9.7762	9.3280	7.1542
15	17.8984	17.0027	13.2676	12.3471	15.9200	10.7157	10.6695	8.0937
16	19.0943	18.1432	15.0897	15.1168	17.0241	12.2382	10.6524	10.4722
17	22.4943	21.5432	17.3703	16.4908	20.3159	15.0782	14.0524	12.5522
18	18.7888	17.8937	18.7948	17.9093	14.2444	15.1567	14.7369	13.7107
19	20.9713	19.1122	17.8633	17.0038	17.7189	14.6772	15.2794	13.3872
20	25.7994	24.9105	20.9714	19.8969	23.8278	18.4315	17.2693	15.8775
21	22.7356	22.4947	19.7856	19.5851	20.7660	16.7317	15.4235	14.3417
22	23.5862	23.2209	20.1858	19.8633	21.7002	17.1099	16.8497	14.6439
23	24.6327	22.9736	25.1031	24.2052	19.1235	21.4526	20.8708	20.0066
24	29.8996	27.4087	26.1216	24.1671	26.3160	23.0237	24.0995	20.9357
25	25.0773	24.4764	25.1033	24.2828	22.4557	21.3954	20.8572	19.9214
26	27.8479	26.7368	28.8941	27.9896	22.1385	25.2362	24.3780	23.7822
27	31.5279	30.6390	26.7299	25.5894	29.5563	24.1780	23.4238	21.6060
28	32.0626	31.1737	27.7426	26.2121	30.0910	24.7127	25.6965	22.5327
29	30.8490	30.1039	26.9270	26.6265	28.8554	23.9331	23.9651	21.3831


```
In [1]: seeds-read.csv("D:\\Daneshga\\T8\\Amini\\Datasets\\Iris Alternatives\\Wheat Seeds Dataset\\Final Refined CSV.csv")
library(phylentropy)
```

```
In [35]: distance(seeds, method = "manhattan")
#dist(seeds$Perimeter, method = "manhattan", diag = TRUE, upper = FALSE)
```

Metric: 'manhattan'; comparing: 210 vectors.

A matrix: 210 × 210 of type dbl

	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	...	v201	v202	v203	v204	v210
v1	0.0000	3.3571	5.1240	6.3035	6.3284	7.4161	8.5479	9.9621	11.2747	11.4860	...	210.5831	208.6467	213.6810	216.2624	213.101
v2	3.3571	0.0000	4.1729	5.3524	5.5913	6.5070	8.1852	9.1410	12.2204	12.2769	...	209.8722	207.7376	212.7921	215.4413	212.301
v3	5.1240	4.1729	0.0000	2.1445	7.0376	3.7079	6.4561	5.5439	12.3413	12.5260	...	206.0171	204.3853	208.6810	211.5736	208.441
v4	6.3035	5.3524	2.1445	0.0000	6.1489	3.2934	6.5156	5.2434	11.4368	11.6215	...	204.9426	202.4772	207.6065	210.4991	207.371
v5	6.3284	5.5913	7.0376	6.1489	0.0000	5.3963	6.6465	7.9503	6.8677	6.8304	...	208.6115	206.6217	211.7094	214.2580	211.131
v6	7.4161	6.5070	3.7079	3.2934	5.3963	0.0000	3.2222	2.7100	8.6834	8.8681	...	203.3932	201.5546	206.3131	208.9497	205.821
v7	8.5479	8.1852	6.4561	6.5156	6.6465	3.2222	0.0000	3.2722	7.8152	8.0161	...	202.0530	202.6708	205.1509	207.7145	204.571
v8	9.9621	9.1410	5.5439	5.2434	7.9503	2.7100	3.2722	0.0000	7.2294	7.4141	...	200.7512	199.4846	203.7591	206.3077	203.181
v9	11.2747	12.2204	12.3413	11.4368	6.8677	8.6834	7.8152	7.2294	0.0000	2.0373	...	205.8578	203.9140	208.9557	211.5297	208.371
v10	11.4860	12.2769	12.5260	11.6215	6.8304	8.8681	8.0161	7.4141	2.0373	0.0000	...	204.0691	202.0987	207.1670	209.7156	206.581
v11	12.5464	13.8055	12.6164	12.6759	10.7568	9.3825	6.1603	7.4325	7.6131	5.8244	...	198.2447	200.7971	202.3266	203.9268	202.011
v12	14.0776	12.2535	10.7964	9.4559	11.0808	7.3685	8.2603	5.2545	8.9829	7.0424	...	197.6187	195.5771	200.6286	203.1928	200.041
v13	16.8910	15.9819	12.1470	11.2265	14.8854	9.5951	8.3431	6.9731	12.1583	10.3430	...	194.3401	195.1417	197.0040	199.8966	196.901
v14	16.9679	16.0692	12.4221	11.5216	14.9865	9.7762	9.3280	7.1542	12.2352	10.4441	...	193.9710	193.3508	196.7229	199.5505	196.401
v15	17.8984	17.0027	13.2676	12.3471	15.9200	10.7157	10.6695	8.0937	13.1663	11.3776	...	193.1315	192.0543	195.7954	198.7140	195.561
v16	19.0943	18.1432	15.0897	15.1168	17.0241	12.2382	10.6524	10.4722	14.3196	12.5043	...	192.1154	193.3676	195.0473	197.6719	195.071
v17	22.4943	21.5432	17.3703	16.4908	20.3159	15.0782	14.0524	12.5522	17.6196	15.8043	...	189.0534	192.2036	192.8833	194.5379	194.081
v18	18.7888	17.8937	18.7948	17.9093	14.2444	15.1567	14.7369	13.7107	12.5281	10.4908	...	194.3719	192.3821	197.4698	200.0184	196.891
v19	20.9713	19.1122	17.8633	17.0038	17.7189	14.6772	15.2794	13.3872	15.5706	13.6113	...	191.7364	189.1186	194.3723	197.2649	194.131
v20	25.7994	24.9105	20.9714	19.8969	23.8278	18.4315	17.2693	15.8775	21.0741	19.2854	...	185.0457	186.6281	187.9836	190.7238	187.841
v21	22.7356	22.4947	19.7856	19.5851	20.7660	16.7317	15.4235	14.3417	18.0103	16.2216	...	187.8525	187.4923	190.9646	193.5520	190.361
v22	23.5862	23.2209	20.1858	19.8633	21.7002	17.1099	16.8497	14.8439	18.8585	17.0698	...	186.9993	185.8365	190.0972	192.6762	189.511
v23	24.6327	22.9736	25.1031	24.2052	19.1235	21.4526	20.8708	20.0066	17.8720	15.8347	...	190.6678	188.6780	193.7657	196.3143	193.181
v24	29.8996	27.4087	26.1216	24.1671	26.3160	23.0237	24.0995	20.9357	24.8123	22.8816	...	183.8295	182.1263	185.3966	190.1400	186.211
v25	25.0773	24.4764	25.1033	24.2828	22.4557	21.3954	20.8572	19.9214	19.9380	18.0073	...	186.5058	184.4920	189.5203	192.1077	188.931
v26	27.8479	26.7368	28.8941	27.9896	22.1385	25.2362	24.3780	23.7822	19.7212	17.7701	...	188.4310	186.4668	191.5289	194.0825	190.951
v27	31.5279	30.6390	26.7299	25.5894	29.5563	24.1780	23.4238	21.6060	26.8026	25.0139	...	179.4952	179.4966	182.1669	185.2283	181.921
v28	32.0626	31.1737	27.7426	26.2121	30.0910	24.7127	25.6965	22.5327	27.3373	25.5486	...	178.9475	177.3793	181.6416	184.7610	181.311
v29	30.8490	30.1039	26.9270	26.6265	28.8554	23.9331	23.9651	21.3831	26.1163	24.3130	...	179.7701	178.7057	182.8540	185.4134	182.271
v30	33.5676	32.9607	29.5556	28.5751	31.5960	26.7417	25.1475	24.0317	28.8423	27.0536	...	177.0165	177.7023	180.1286	182.7680	179.531
:	:	:	:	:	:	:	:	:	:	:	...	:	:	:	:	:
v181	192.0960	191.2071	187.0900	186.0155	190.1244	184.7281	183.5659	182.1741	187.3707	185.5820	...	22.5429	25.8747	23.4970	28.8644	26.991
v182	190.8914	190.3845	186.5294	185.4549	188.9198	183.9055	182.3613	181.2635	186.1661	184.3774	...	19.7517	23.7861	24.6616	25.9998	24.461
v183	192.1871	191.7022	187.8471	186.7726	190.2155	185.2232	183.6570	182.5812	187.4618	185.6731	...	19.1020	23.0118	23.1261	25.5855	23.471
v184	194.5835	194.0526	190.1975	189.1230	192.6119	187.5736	186.0534	184.9316	189.8582	188.0695	...	18.7064	23.0182	21.4345	25.1179	23.751
v185	194.0709	193.7100	190.2629	189.1224	192.1893	187.4490	185.5428	184.7390	189.1536	187.3549	...	18.1438	22.7896	24.0979	22.5473	23.761
v186	195.1892	194.7523	191.0412	189.9007	193.2316	188.2733	186.6591	185.6313	190.4639	188.6752	...	17.9881	20.2279	18.8642	24.4716	20.761
v187	196.6462	196.2853	192.6742	191.5337	194.7646	189.8603	188.1181	187.1643	191.6569	189.8682	...	15.9691	20.1209	19.2772	22.5126	20.391
v188	198.9578	198.0689	194.2138	193.1393	196.9862	191.5899	190.4277	189.0359	194.2325	192.4438	...	16.9607	18.9945	15.8508	23.2822	19.851
v189	202.7926	201.9057	198.0506	196.9761	200.8210	195.4267	194.2625	192.8707	198.0673	196.2786	...	16.2095	20.8333	17.8516	18.4270	21.691
v190	202.6752	201.7863	197.6692	196.5547	200.7036	195.3073	194.1451	192.7533	197.9499	196.1612	...	15.0939	18.1799	14.9942	21.4076	19.581
v191	203.0660	202.3551	198.5000	197.4255	201.0944	195.8761	194.5359	193.2341	198.3407	196.5520	...	12.6309	17.1967	14.0530	19.0424	18.051
v192	202.1477	201.3488	197.4937	196.4192	200.1761	194.8698	193.6176	192.2278	197.4224	195.8337	...	12.5386	14.2804	11.5687	18.8621	15.041
v193	201.6725	201.1186	197.2635	196.1890	199.6839	194.6396	193.1254	191.9976	196.9398	195.1415	...	11.0336	12.1322	12.0255	17.4079	13.501
v194	205.5234	204.9005	201.0454	199.9709	203.5518	198.4215	196.9933	195.7795	200.7981	199.0094	...	10.2663	14.0101	10.9264	16.6778	14.591
v195	203.7101	202.9332	199.0781	198.0036	201.7385	196.4542	195.1800	193.8122	198.9848	197.1961	...	7.8470	10.1548	10.1411	14.2585	10.731
v196	204.5920	203.6829	199.7340	198.6595	202.5904	197.1941	196.0441	194.6401	199.8593	198.0480	...	6.4011	9.2187	10.7630	11.8884	10.711
v197	206.1086	205.2085	201.3534	200.2789	204.1218	198.7295	197.5633	196.1715	201.3759	199.5794	...	4.6857	9.0661	10.4516	10.3118	10.221
v198	205.1689	204.5298	200.7251	199.5926	203.1695	198.0432	196.6210	195.4012	200.4362	198.6271	...	5.4420	8.2318	9.7779	11.0935	9.461
v199	205.6829	205.3220	201.7049	200.5644	203.8013	198.8910	197.7148	196.2010	200.9356	199.1469	...	5.1038	5.5916	8.0599	11.1673	7.551
v200	206.8294	205.9203	201.7826	200.7099	204.8070	199.4133	199.7975	196.8673	202.0967	200.2814	...	4.6465	2.8733	6.9024	10.0710	7.101
v201	210.5831	209.8722	206.0171	204.9426	206.6115	203.3932	202.0530	200.7512	205.8578	204.0691	...	0.0000	5.2998	5.9541	6.5435	5.681
v202	208.6467	207.7376	204.3853	202.4772	206.6217	201.5546	202.6708	199.4846	203.9140	202.0987	...	5.2998	0.0000	5.3037	8.7843	5.761
v203	213.6810	212.7921	208.6810	207.6065	211.7094	206.3131	205.1509	203.7591	208.9557	207.1670	...	5.9541	5.3037	0.0000	8.2614	4.601
v204	216.2624	215.4413	211.5736	210.4991	214.2580	208.9497	207.7145	206.3077	211.5297	209.7156	...	6.5435	8.7843	8.2614	0.0000	6.111
v205	213.1023	212.3034	208.4483	207.3738	211.1307	205.8244	204.5722	203.1824	208.3770	206.5883	...	5.6852	5.7610	4.6013	6.1107	0.001
v206	214.4343	213.5308	209.5037	208.4292	212.4481	207.0518	205.8896	204.4978	209.7016	207.9057	...	7.5534	6.3824	5.2613	7.8401	1.971
v207	217.4709	216.6760	212.8209	211.7464	215.4993	210.1970	208.9408	207.5550	212.7456	210.9569	...	9.1818	9.7056	4.8219	9.5093	4.371
v208	219.1223	218.4132	214.5437	213.4692	217.1161	211.9198	210.5744	209.2778	214.3896	212.5743	...	11.1814	13.5544	13.1213	5.1419	8.801
v209	217.6859	216.9730	213.1179	212.0434	215.7143	210.4940	209.1558	207.8520	212.9606	211.1719	...	10.8508	10.0726	8.0269	11.2603	5.311
v210	219.8936	219.2187	215.3636	214.2891	217.9220	212.7397	211.3635	210.0977	215.1683</							


```
In [34]: #distance(seeds, method = "manhattan")
dist(seeds$Perimeter, method = "manhattan", diag = TRUE, upper = FALSE)
```

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0.00														
2	0.27	0.00													
3	0.75	0.48	0.00												
4	0.90	0.63	0.15	0.00											
5	0.15	0.42	0.90	1.05	0.00										
6	0.63	0.36	0.12	0.27	0.78	0.00									
7	0.35	0.08	0.40	0.55	0.50	0.28	0.00								
8	0.74	0.47	0.01	0.16	0.89	0.11	0.39	0.00							
9	0.62	0.89	1.37	1.52	0.47	1.25	0.97	1.36	0.00						
10	0.41	0.68	1.16	1.31	0.26	1.04	0.76	1.15	0.21	0.00					
11	0.01	0.28	0.76	0.91	0.14	0.64	0.36	0.75	0.61	0.40	0.00				
12	0.68	0.41	0.07	0.22	0.83	0.05	0.33	0.06	1.30	1.09	0.69	0.00			
13	0.82	0.55	0.07	0.08	0.97	0.19	0.47	0.08	1.44	1.23	0.83	0.14	0.00		
14	0.78	0.51	0.03	0.12	0.93	0.15	0.43	0.04	1.40	1.19	0.79	0.10	0.04	0.00	
15	0.79	0.52	0.04	0.11	0.94	0.16	0.44	0.05	1.41	1.20	0.80	0.11	0.03	0.01	0.00
16	0.56	0.29	0.19	0.34	0.71	0.07	0.21	0.18	1.18	0.97	0.57	0.12	0.26	0.22	0.23
17	1.01	0.74	0.26	0.11	1.16	0.38	0.66	0.27	1.63	1.42	1.02	0.33	0.19	0.23	0.22
18	0.09	0.18	0.66	0.81	0.24	0.54	0.26	0.65	0.71	0.50	0.10	0.59	0.73	0.69	0.70
19	0.63	0.36	0.12	0.27	0.78	0.00	0.28	0.11	1.25	1.04	0.64	0.05	0.19	0.15	0.16
20	1.27	1.00	0.52	0.37	1.42	0.64	0.92	0.53	1.89	1.68	1.28	0.59	0.45	0.49	0.48
21	0.44	0.17	0.31	0.46	0.59	0.19	0.09	0.30	1.06	0.85	0.45	0.24	0.38	0.34	0.35
22	0.58	0.31	0.17	0.32	0.73	0.05	0.23	0.16	1.20	0.99	0.59	0.10	0.24	0.20	0.21
23	0.06	0.33	0.81	0.96	0.09	0.69	0.41	0.80	0.56	0.35	0.05	0.74	0.88	0.84	0.85
24	1.61	1.34	0.86	0.71	1.76	0.98	1.26	0.87	2.23	2.02	1.62	0.93	0.79	0.83	0.82
25	0.08	0.19	0.67	0.82	0.23	0.55	0.27	0.66	0.70	0.49	0.09	0.60	0.74	0.70	0.71
26	0.32	0.59	1.07	1.22	0.17	0.95	0.67	1.06	0.30	0.09	0.31	1.00	1.14	1.10	1.11
27	1.08	0.81	0.33	0.18	1.23	0.45	0.73	0.34	1.70	1.49	1.09	0.40	0.26	0.30	0.29
28	1.17	0.90	0.42	0.27	1.32	0.54	0.82	0.43	1.79	1.58	1.18	0.49	0.35	0.39	0.38
29	0.66	0.39	0.09	0.24	0.81	0.03	0.31	0.08	1.28	1.07	0.67	0.02	0.16	0.12	0.13
30	0.82	0.55	0.07	0.08	0.97	0.19	0.47	0.08	1.44	1.23	0.83	0.14	0.00	0.04	0.03
31	1.02	0.75	0.27	0.12	1.17	0.39	0.67	0.28	1.64	1.43	1.03	0.34	0.20	0.24	0.23
32	0.10	0.37	0.85	1.00	0.05	0.73	0.45	0.84	0.52	0.31	0.09	0.78	0.92	0.88	0.89
33	0.43	0.16	0.32	0.47	0.58	0.20	0.08	0.31	1.05	0.84	0.44	0.25	0.39	0.35	0.36
34	0.67	0.40	0.08	0.23	0.82	0.04	0.32	0.07	1.29	1.08	0.68	0.01	0.15	0.11	0.12

```
In [37]: #distance(seeds, method = "manhattan")
#dist(seeds$Perimeter, method = "manhattan", diag = TRUE, upper = FALSE)
manhattan_dist <- function(a, b){
  dist <- abs(a-b)
  dist <- sum(dist)
  return(dist)
}
manhattan_dist(seeds$Area, seeds$Perimeter)
```

286.41

استفاده از توابع euclidean و manhattan کتابخانهی philentropy

euclidean(P, Q, testNA)

manhattan(P, Q, testNA)

که در آن P و Q دو وکتوری هستند که فاصله‌ی آن‌ها را می‌خواهیم به دست آوریم. مقدار بولین testNA به منزله چک کردن برای مقادیر missnig می‌باشد.

(برای نصب کتابخانه philentropy از فرمان زیر در ترمینال مامبا استفاده شد.)

```
mamba install -c conda-forge r-philentropy
```

```
In [1]: seeds=read.csv("D:\\Daneshga\\T8\\Amini\\Datasets\\Iris Alternatives\\Wheat Seeds Dataset\\  
library(philentropy)
```

```
In [40]: #distance(seeds, method = "manhattan")  
#dist(seeds$Perimeter, method = "manhattan", diag = TRUE, upper = FALSE)  
manhattan(seeds$Area, seeds$Perimeter, testNA = FALSE)
```

286.41

```
In [1]: seeds=read.csv("D:\\Daneshga\\T8\\Amini\\Datasets\\Iris Alternatives\\Wheat Seeds Dataset\\  
library(philentropy)
```

```
In [43]: #distance(seeds, method = "manhattan")  
#dist(seeds$Perimeter, method = "manhattan", diag = TRUE, upper = FALSE)  
euclidean(seeds$Compactness, seeds$Perimeter, testNA = FALSE)
```

199.241758580901

Supremum Distance

فاصله‌ی سوپریمم

متد maximum در تابع از پیش تعریف شده‌ی `dist(stats::dist)` فاصله‌ی سوپریمم را برای ما محاسبه می‌کند.

```
dist(Dataset, method = "maximum", diag = TRUE,  
upper = FALSE)
```

```
stats::dist(Dataset, method = "maximum", ...)
```

یا:

```
In [1]: seeds=read.csv("D:\\Daneshga\\T8\\Amini\\Datasets\\Iris Alternatives\\Wheat Seeds Dat  
library(philentropy)
```

```
In [44]: #distance(seeds, method = "manhattan")  
dist(seeds$Compactness, method = "maximum", diag = TRUE, upper = FALSE)  
#euclidean(seeds$Compactness, seeds$Perimeter, testNA = FALSE)
```

	1	2	3	4	5	6	7	8	9	10
1	0.0000									
2	0.0101	0.0000								
3	0.0340	0.0239	0.0000							
4	0.0245	0.0144	0.0095	0.0000						
5	0.0324	0.0223	0.0016	0.0079	0.0000					
6	0.0241	0.0140	0.0099	0.0004	0.0083	0.0000				
7	0.0089	0.0012	0.0251	0.0156	0.0235	0.0152	0.0000			
8	0.0201	0.0100	0.0139	0.0044	0.0123	0.0040	0.0112	0.0000		
9	0.0037	0.0064	0.0303	0.0208	0.0287	0.0204	0.0052	0.0164	0.0000	
10	0.0170	0.0069	0.0170	0.0075	0.0154	0.0071	0.0081	0.0031	0.0133	0.0000
11	0.0014	0.0115	0.0354	0.0259	0.0338	0.0255	0.0103	0.0215	0.0051	0.0184
12	0.0086	0.0015	0.0254	0.0159	0.0238	0.0155	0.0003	0.0115	0.0049	0.0084
13	0.0170	0.0069	0.0170	0.0075	0.0154	0.0071	0.0081	0.0031	0.0133	0.0000
14	0.0049	0.0052	0.0291	0.0196	0.0275	0.0192	0.0040	0.0152	0.0012	0.0121
15	0.0034	0.0067	0.0306	0.0211	0.0290	0.0207	0.0055	0.0167	0.0003	0.0136
16	0.0283	0.0182	0.0057	0.0038	0.0041	0.0042	0.0194	0.0082	0.0246	0.0113
17	0.0473	0.0372	0.0133	0.0228	0.0149	0.0232	0.0384	0.0272	0.0436	0.0303
18	0.0348	0.0247	0.0008	0.0103	0.0024	0.0107	0.0259	0.0147	0.0311	0.0178
19	0.0443	0.0342	0.0103	0.0198	0.0119	0.0202	0.0354	0.0242	0.0406	0.0273
20	0.0024	0.0125	0.0364	0.0269	0.0348	0.0265	0.0113	0.0225	0.0061	0.0194
21	0.0126	0.0227	0.0466	0.0371	0.0450	0.0367	0.0215	0.0327	0.0163	0.0296
22	0.0012	0.0089	0.0328	0.0233	0.0312	0.0229	0.0077	0.0189	0.0025	0.0158
23	0.0278	0.0177	0.0062	0.0033	0.0046	0.0037	0.0189	0.0077	0.0241	0.0108
24	0.0046	0.0147	0.0386	0.0291	0.0370	0.0287	0.0135	0.0247	0.0083	0.0216
25	0.0053	0.0154	0.0393	0.0298	0.0377	0.0294	0.0142	0.0254	0.0090	0.0223
26	0.0139	0.0038	0.0201	0.0106	0.0185	0.0102	0.0050	0.0062	0.0102	0.0031
27	0.0069	0.0170	0.0409	0.0314	0.0393	0.0310	0.0158	0.0270	0.0106	0.0239
28	0.0116	0.0217	0.0406	0.0301	0.0470	0.0307	0.0135	0.0217	0.0102	0.0216

Jaccard Distance

فاصله‌ی جاکارد

```
In [1]: seeds <- read.csv("D:\\Daneshga\\T8\\Amini\\Datasets\\Iris Alternatives\\Wheat Seeds Dataset\\Final Refined CSV.csv")
library(philentropy)
```

```
In [5]: jaccard(seeds$Length.of.Kernel, seeds$Width.of.Kernel, testNA = FALSE)
0.23465386829523
```

```
In [3]: distance(seeds, method = "jaccard")
```

Metric: 'jaccard'; comparing: 210 vectors.

A matrix: 210 × 210 of type dbl

	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	...	v201
v1	0.000000000	0.005332761	0.011966779	0.024010691	0.03104986	0.049158911	0.067956661	0.091567326	0.104042684	0.127245899	...	0.9834979
v2	0.005332761	0.000000000	0.009029603	0.014398371	0.01991981	0.035474738	0.057818891	0.073250907	0.088182709	0.108254304	...	0.9788748
v3	0.011966779	0.009029603	0.000000000	0.002973864	0.01917953	0.018040427	0.032759624	0.048020941	0.074918932	0.092004024	...	0.9739318
v4	0.024010691	0.014398371	0.002973864	0.000000000	0.01594017	0.008934682	0.023091793	0.031918733	0.061686344	0.075506562	...	0.9690280
v5	0.031049860	0.019919811	0.019179529	0.015940165	0.000000000	0.011034061	0.019798366	0.027916271	0.027299799	0.039654401	...	0.9628483
v6	0.049158911	0.035474738	0.018040427	0.008934682	0.01103406	0.000000000	0.004715716	0.007838800	0.028067690	0.036213365	...	0.9583037
v7	0.067956661	0.057818891	0.032759624	0.023091793	0.01979837	0.004715716	0.000000000	0.004274013	0.018810673	0.024635352	...	0.9526320
v8	0.091567326	0.073250907	0.048020941	0.031918733	0.02791627	0.007838800	0.004274013	0.000000000	0.017573742	0.019165752	...	0.9477570
v9	0.104042684	0.088182709	0.074918932	0.061686344	0.02729980	0.028067690	0.018810673	0.017573742	0.000000000	0.001797484	...	0.9411116
v10	0.127245899	0.108254304	0.092004024	0.075506562	0.03965440	0.036213365	0.024635352	0.019165752	0.001797484	0.000000000	...	0.9358760
v11	0.161444920	0.147846805	0.113611427	0.096035650	0.07240246	0.050583157	0.027983301	0.023743537	0.019015352	0.013676620	...	0.9305737
v12	0.194859331	0.166339571	0.139610501	0.112402038	0.08712099	0.063015625	0.048579967	0.029183453	0.028539497	0.017781675	...	0.9260953
v13	0.227181053	0.205494053	0.166515456	0.140761333	0.11882044	0.085589140	0.060412128	0.044591732	0.045916097	0.032689146	...	0.9203200
v14	0.253469763	0.227977543	0.192335216	0.163825494	0.13616150	0.104566543	0.079457203	0.059020833	0.054574302	0.038755676	...	0.9148208
v15	0.281980662	0.255219082	0.220145477	0.190285236	0.15874372	0.127351050	0.100431769	0.077573286	0.068630645	0.050457137	...	0.9092077
v16	0.303226776	0.281020331	0.243156680	0.215884590	0.17961151	0.149905030	0.117105738	0.098006511	0.081088204	0.062080604	...	0.9030104
v17	0.342629509	0.322111696	0.279164910	0.251437844	0.21824758	0.182852805	0.145726905	0.126152731	0.111766106	0.090550546	...	0.8974504
v18	0.344399066	0.316475084	0.290934159	0.261796592	0.20970732	0.192627555	0.163275665	0.138833737	0.102537246	0.081044782	...	0.8913071
v19	0.382475438	0.353245312	0.324414971	0.293177286	0.24468737	0.221654675	0.190723524	0.162869156	0.131371337	0.106850857	...	0.8859692
v20	0.431139677	0.406019953	0.366717070	0.334979092	0.29779098	0.261537884	0.223717307	0.196429877	0.176112513	0.149656837	...	0.8804967
v21	0.431743851	0.406173933	0.373817849	0.343424968	0.29639925	0.269976418	0.233551256	0.207250489	0.173195944	0.147389369	...	0.8739432
v22	0.457047144	0.430761859	0.399729576	0.369003613	0.32059653	0.295113625	0.258880186	0.231315372	0.194886369	0.168035486	...	0.8680836
v23	0.464706743	0.436849820	0.414227679	0.384759243	0.32686636	0.312333789	0.279724466	0.252253179	0.203342396	0.177127028	...	0.8616638
v24	0.533621983	0.503952045	0.474171039	0.440688500	0.39586184	0.366655309	0.332300205	0.298773750	0.266595775	0.236430910	...	0.8571820
v25	0.512116046	0.486220241	0.460413872	0.431175035	0.37730126	0.358448504	0.322997052	0.295699913	0.249112383	0.221489912	...	0.8496078
v26	0.522155605	0.496436832	0.474585812	0.446599402	0.38863109	0.375325348	0.341368142	0.315054390	0.261993253	0.235192376	...	0.8431221
v27	0.575025194	0.550956003	0.520027371	0.490550766	0.44472959	0.418137256	0.379984187	0.352012986	0.313401946	0.283937379	...	0.8379479
v28	0.596396451	0.571350433	0.542647956	0.512931737	0.46664126	0.441438498	0.404824306	0.375707274	0.335960628	0.306209106	...	0.8319434
v29	0.596609878	0.573200819	0.546087812	0.518204493	0.46908320	0.447847715	0.411092622	0.384461644	0.339545668	0.310867722	...	0.8250354
v30	0.618896934	0.596672569	0.568270778	0.540863927	0.49452864	0.471428883	0.433942541	0.407918869	0.365228876	0.336531179	...	0.8188325
:	:	:	:	:	:	:	:	:	:	:	:	:
v181	0.9810700	0.9759422	0.9704563	0.9649969	0.9580758	0.9530242	0.9466805	0.9412381	0.9337749	0.9279189	...	0.0107963088 0.
v182	0.9804697	0.9753868	0.9699564	0.9645429	0.9575704	0.9526167	0.9462931	0.9409083	0.9333762	0.9275664	...	0.0096651213 0.
v183	0.9808089	0.9757437	0.9703441	0.9649542	0.9580445	0.9531032	0.9468258	0.9414608	0.9340010	0.9282227	...	0.0086433776 0.
v184	0.9813577	0.9763224	0.9709200	0.9655559	0.9587506	0.9537845	0.9475418	0.9422019	0.9348599	0.9291104	...	0.0076878674 0.
v185	0.9805251	0.9755723	0.9701852	0.9648825	0.9580343	0.9531479	0.9468886	0.9416382	0.9342259	0.9285271	...	0.0067826831 0.
v186	0.9816700	0.9766461	0.9713618	0.9660386	0.9592835	0.9544064	0.9482748	0.9429620	0.9356833	0.9299981	...	0.0059903458 0.
v187	0.9815844	0.9766204	0.9713303	0.9660505	0.9593381	0.9544751	0.9483467	0.9430925	0.9358491	0.9302025	...	0.0051520255 0.
v188	0.9825530	0.9775699	0.9723152	0.9670397	0.9604377	0.9555559	0.9495069	0.9442337	0.9371353	0.9315042	...	0.0045193914 0.
v189	0.9823599	0.9775226	0.9721433	0.9669453	0.9604557	0.9555093	0.9493841	0.9442363	0.9372311	0.9316445	...	0.0038891538 0.
v190	0.9831459	0.9782314	0.9729821	0.9677668	0.9613041	0.9564184	0.9504230	0.9452148	0.9382717	0.9326990	...	0.0032478059 0.
v191	0.9829222	0.9780547	0.9728254	0.9676484	0.9611993	0.9563515	0.9503682	0.9452105	0.9382706	0.9327368	...	0.0026380877 0.
v192	0.9829796	0.9780910	0.9729601	0.9677953	0.9613232	0.9565613	0.9506544	0.9454917	0.9385388	0.9330324	...	0.0021706818 0.
v193	0.9828072	0.9779371	0.9728620	0.9677264	0.9612379	0.9565452	0.9506769	0.9455427	0.9385712	0.9330986	...	0.0017306403 0.
v194	0.9834061	0.9785905	0.9734665	0.9683604	0.9620185	0.9572531	0.9513897	0.9462958	0.9394790	0.9340343	...	0.0013263130 0.
v195	0.9828816	0.9780794	0.9730384	0.9679653	0.9615473	0.9568988	0.9510769	0.9460146	0.9391160	0.9337054	...	0.0009557170 0.
v196	0.9826542	0.9779054	0.9728691	0.9678394	0.9614317	0.9568192	0.9510013	0.9459951	0.9391013	0.9337245	...	0.0006416724 0.

محاسبه ماتریس عدم شباهت با کتابخانه‌ی cluster

با تابع daisy که در کتابخانه‌ی cluster تعریف شده نیز می‌توان به صورت زیر ماتریس عدم شباهت را ترسیم کرد.

```
library(cluster)
daisy(Dataset, metric = c("euclidean", "manhattan"),
stand = FALSE)
```

”metric“ بیانگر نوع فاصله‌ایست که نیاز به محاسبه‌ی آن داریم، به عنوان مثال می‌توان آن را معادل euclidean، manhattan یا gower قرار داد. gower برای مواقعی استفاده می‌شود که داده به طور کامل عددی نباشد.

با ”stand“ می‌توان وضعیت استاندارد کردن داده (با تفریق میانگین از آن و تقسیم آن بر انحراف معیار میانگین) را تعیین کرد. همانطور که در تصویر صفحه‌ی بعد مشاهده می‌شود، این آرگومان بولین می‌باشد.

```
In [48]: seeds=read.csv("D:\\Daneshga\\T8\\Amini\\Datasets\\Iris Alternatives\\Wheat Seeds Dataset")
library(cluster)
```

```
In [51]: daisy(seeds, metric = c("euclidean"),
stand = FALSE)
```

Dissimilarities :

	1	2	3	4	5	6
2	1.666862					
3	2.472342	2.119240				
4	3.492546	2.663945	1.192453			
5	4.198030	3.315009	3.216621	2.914056		
6	5.142158	4.300418	3.017545	2.106382	2.473047	
7	6.193021	5.631328	4.164457	3.468298	3.381299	1.594181
8	7.160034	6.297926	5.008112	4.043040	3.984601	2.035891
9	8.175159	7.417128	6.744734	6.068848	4.193958	4.137637
10	9.098030	8.262679	7.508823	6.739415	5.071225	4.710253
11	10.266836	9.685587	8.327494	7.582333	6.861730	5.545797
12	11.108344	10.070358	9.061377	8.028799	7.381166	6.057831
13	12.248056	11.452964	10.093071	9.167421	8.791843	7.187623
14	13.147705	12.247451	11.023753	10.043195	9.533934	8.054638
15	14.132028	13.202414	12.018550	11.025927	10.463534	9.040028
16	15.165207	14.362112	13.089686	12.181585	11.498584	10.151356

نرمال سازی

یکی از کتابخانه‌هایی که در R برای پیش‌پردازش به کمک ما می‌آید، نام **caret** دارد. البته برای استفاده از این کتابخانه پیام نیاز به دو لایبرری دیگر (**ggplot2** و **lattice**) را نیز دریافت کردم که از قبل نصب بودند و همان‌گونه که مشاهده می‌شود فقط آنان را فرا خواندم.

(برای نصب کتابخانه **caret** از فرمان زیر در ترمینال مامبا استفاده شد.)

```
mamba install -c r r-caret
```

نرمال سازی Min-Max

```
In [1]: seeds=read.csv("D:\\Daneshga\\T8\\Amini\\Datasets\\Iris Alternatives\\Wheat Seeds Dataset")
library(ggplot2)
library(lattice)
library(caret)
```

```
In [4]: preproc2 <- preProcess(seeds, method=c("range"))
norm2 <- predict(preproc2, seeds)

#summary(norm2)
cols <- c("Area", "Perimeter", "Compactness", "Length.of.Kernel", "Width.of.Kernel", "Asymmetry.Coefficient", "Length.of.Kernel.Groove")
summary(norm2[cols])
```

Area	Perimeter	Compactness	Length.of.Kernel
Min. :0.0000	Min. :0.0000	Min. :0.0000	Min. :0.0000
1st Qu.:0.1586	1st Qu.:0.2149	1st Qu.:0.4428	1st Qu.:0.2045
Median :0.3555	Median :0.3946	Median :0.5930	Median :0.3516
Mean :0.4020	Mean :0.4441	Mean :0.5708	Mean :0.4108
3rd Qu.:0.6341	3rd Qu.:0.6829	3rd Qu.:0.7230	3rd Qu.:0.6085
Max. :1.0000	Max. :1.0000	Max. :1.0000	Max. :1.0000
Width.of.Kernel	Asymmetry.Coefficient	Length.of.Kernel.Groove	
Min. :0.0000	Min. :0.0000	Min. :0.0000	
1st Qu.:0.2238	1st Qu.:0.2336	1st Qu.:0.2590	
Median :0.4326	Median :0.3685	Median :0.3466	
Mean :0.4480	Mean :0.3816	Mean :0.4378	
3rd Qu.:0.6641	3rd Qu.:0.5206	3rd Qu.:0.6686	
Max. :1.0000	Max. :1.0000	Max. :1.0000	

مشاهده می‌شود که با تابع `preProcess` (تعریف شده در `caret`) نرمال‌سازی صورت گرفت و همچنین در نهایت مانند قسمت اول تمرین ۱ برای دستور `summary` تعداد ستون‌ها را محدود کردم تا مقادیری که نرمال‌سازی برای آنان بی‌معنی است (مانند ستون شماره ردیف‌ها و برچسب) نرمال نشوند.

کتابخانه `heatmaply`

در تهیه این گزارش کار، با کتابخانه جالبی آشنا شدم که فراتر از یک عکس، یک نمودار در لحظه (`Real-Time`) را با نرمال‌سازی تحویل می‌دهد. با `hover` کردن موس روی هر قسمت نمودار، داده مربوط به آن نقطه نمایش داده می‌شود. مانند `caret`، برای استفاده از این کتابخانه پیام نیاز به دو لایبرری دیگر (`ggplot2` و `plotly`) را نیز دریافت کردم که از قبل نصب بودند و همان‌گونه که مشاهده می‌شود فقط آنان را فرا خواندم.

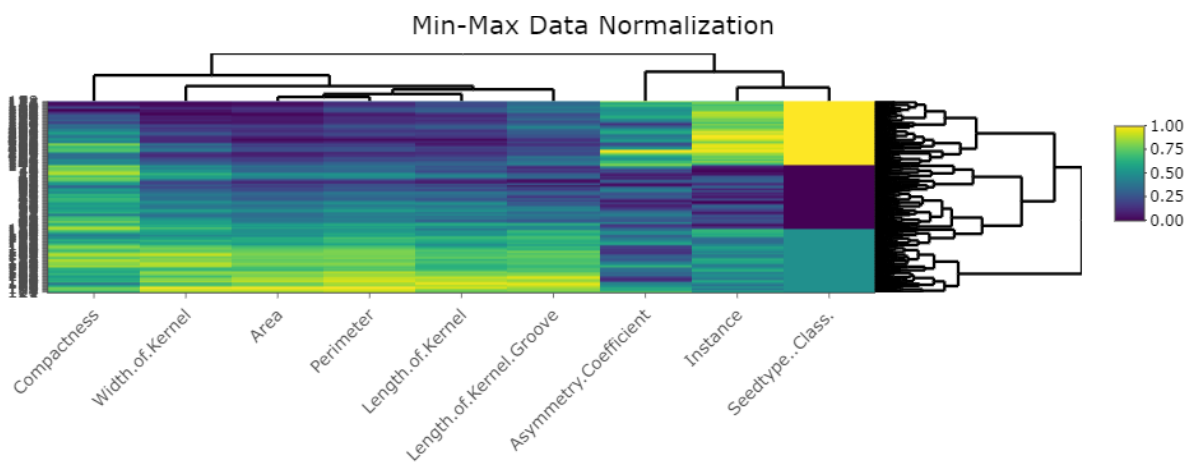
(برای نصب کتابخانه `heatmaply` از فرمان زیر در ترمینال مامبا استفاده شد.)

```
mamba install -c conda-forge r-heatmaply
```

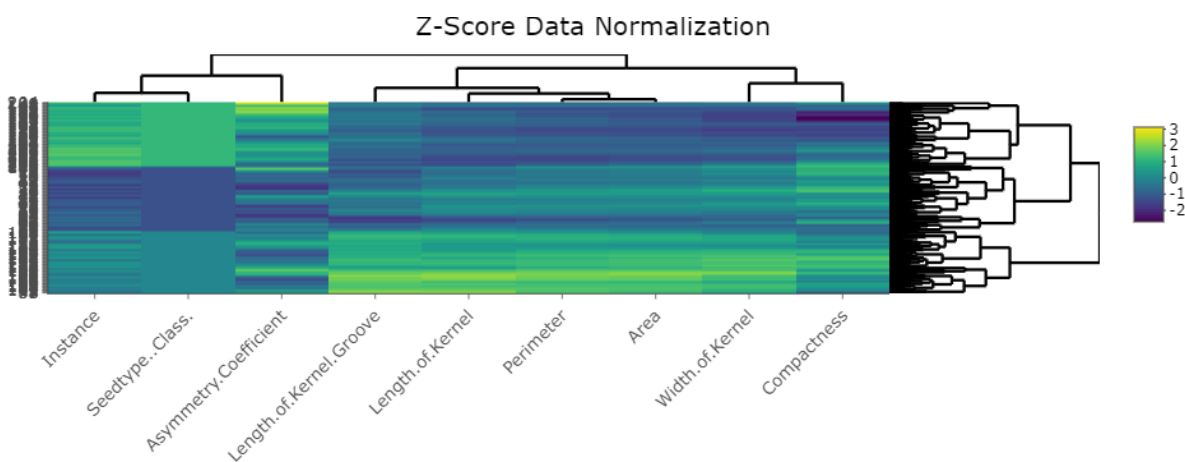
```
library(ggplot2)
library(plotly)
library(heatmaply)

heatmaply(
  normalize(seeds),
  main = "Min-Max Data Normalization"
)
```


نرمال سازی Min-Max با کتابخانه heatmaply



نرمال سازی Z-Score با کتابخانه heatmaply



تابع `znorm` از کتابخانه‌ی `dprep` خروجی بهتری برای نرمال سازی **Z-Score** به ما می‌داد اما هرچه تلاش کردم موفق به نصب آن بر روی `mamba` نشدم.

[znorm: Z-score normalization in dprep: Data Pre-Processing and Visualization Functions for Classification \(rdrr.io\)](https://rdrr.io/rdrr.io/)

Z-Score نرمال سازی

```
In [4]: seeds <- read.csv("D:\\Daneshga\\T8\\Amini\\Datasets\\Iris Alternatives\\Wheat Seeds Dataset\\Final Refined CSV.csv")
cols <- c("Area", "Perimeter", "Compactness", "Length.of.Kernel", "Width.of.Kernel", "Asymmetry.Coefficient", "Length.of.Kernel.Groove")
scale(seeds[cols], center = TRUE, scale = TRUE)
```

A matrix: 210 × 7 of type dbl

Area	Perimeter	Compactness	Length.of.Kernel	Width.of.Kernel	Asymmetry.Coefficient	Length.of.Kernel.Groove
0.14175904	0.214948819	6.045733e-05	0.30349301	0.141364035	-0.98380096	-0.3826631
0.01116136	0.008204153	4.274938e-01	-0.16822270	0.196961591	-1.78390358	-0.9198156
-0.19160873	-0.359341919	1.438945e+00	-0.76181710	0.207551602	-0.66588820	-1.1863572
-0.34626388	-0.474200066	1.036904e+00	-0.68733567	0.318746714	-0.95852756	-1.2270506
0.44419577	0.329806966	1.371233e+00	0.06650665	0.803239702	-1.55976843	-0.4742231
-0.16067770	-0.267455401	1.019976e+00	-0.54740087	0.141364035	-0.82351440	-0.9198156
-0.05413749	-0.053053525	3.767096e-01	-0.14790958	0.001046394	-0.07595385	-0.3846977
-0.25347079	-0.351684709	8.506951e-01	-0.47066243	0.114889009	-0.66522311	-0.8302902
0.61259805	0.689695828	1.566449e-01	0.95802676	0.546431943	-1.10418215	0.9541143
0.54729921	0.528894422	7.195027e-01	0.57659157	0.652332050	-1.15140351	0.2541883
0.14175904	0.222606029	-5.918773e-02	0.19289937	-0.043961151	0.56053676	-0.1914042
-0.28096504	-0.305741450	3.640136e-01	-0.43003620	-0.152508761	-1.31900605	-0.8282555
-0.32907997	-0.412942387	7.195027e-01	-0.42777918	-0.157803766	0.19008193	-1.3633734
-0.36688456	-0.382313548	2.074291e-01	-0.33749867	-0.271646381	-0.37524411	-1.0907278
-0.38063169	-0.389970758	1.439489e-01	-0.33072763	-0.382841493	-0.51092236	-1.1863572
-0.08850530	-0.213854932	1.197720e+00	-0.62639632	0.196961591	0.32243474	-1.2758826
-0.29471216	-0.558429374	2.001803e+00	-1.15002332	0.329336725	1.02011358	-1.2758826
0.28954062	0.146033930	1.472801e+00	-0.22916205	0.676159574	-1.39748661	-0.7366954
-0.05070070	-0.267455401	1.874842e+00	-0.95592021	0.549079446	-1.28575158	-1.5444589
-0.73118336	-0.757516830	-1.015079e-01	-0.90852294	-0.554929166	0.26723231	-1.0052717
-0.23628688	-0.121968414	-5.331732e-01	0.06650665	-0.343128953	-0.41780983	-0.4721885
-0.25347079	-0.229169351	5.084461e-02	-0.24496114	-0.239876349	-0.67320418	-0.3846977
0.35483947	0.260892078	1.176560e+00	-0.02377387	0.657627055	-1.95210471	-0.6451353
-0.95113735	-1.017861964	-1.946121e-01	-1.19516358	-0.854096967	-1.51986307	-0.9096423
0.05583951	0.153691140	-2.242362e-01	0.36217534	-0.036018643	-1.26978943	-0.8282555
0.46137968	0.459979533	5.883103e-01	0.46148391	0.429941826	-1.86038887	-0.2056469
-0.62807993	-0.612029843	-2.919484e-01	-0.52708775	-0.615821727	-0.21761790	-1.1863572
-0.72430980	-0.680944731	-6.178135e-01	-0.52708775	-0.801146914	-0.79558065	-1.0968318
-0.25347079	-0.290427030	4.655819e-01	-0.19756387	-0.099558707	-0.62930828	-0.7529728
-0.48029834	-0.412942387	-4.485329e-01	-0.25398919	-0.512569123	-0.11253377	-0.6329273
:	:	:	:	:	:	:
-1.1814017	-1.2322638	-0.63474150	-1.2154767	-1.28034490	0.83588380	-1.1863572
-0.8205397	-0.8800322	-0.01686759	-0.8859528	-0.63964925	0.85583648	-0.5311939
-0.9133328	-0.9183182	-0.55433326	-0.8769248	-0.92557954	0.76937485	-0.5088125
-1.0989189	-1.1403773	-0.57126131	-1.1748505	-1.08178220	1.00348634	-0.5556099
-0.6727581	-0.6043726	-0.71514975	-0.3013865	-0.61582173	1.65261366	-0.1873349
-1.1298500	-0.9566043	-2.16673024	-0.5993122	-1.52391514	0.24062873	-0.4599805
-1.0439304	-0.8494033	-2.16673024	-0.4864615	-1.43654755	0.79664352	-0.1140868
-1.3532407	-1.3471220	-1.43035996	-1.2199907	-1.54509516	0.31844420	-0.9198156
-1.2432637	-1.3318076	-0.49085306	-1.2177337	-1.15855977	2.54316844	-0.9177809
-1.4632177	-1.6457532	-0.26232435	-1.6465662	-1.24857486	0.84785541	-1.2494319
-1.3463672	-1.3471220	-1.35418373	-1.3147853	-1.43390005	1.12918825	-0.7387301
-1.2295166	-1.3011787	-0.62204547	-1.2132197	-1.20356732	0.18941684	-0.8282555
-1.0233098	-1.1786634	0.35978157	-1.1206822	-0.80908942	-0.06863787	-0.5617139
-1.3841718	-1.3241504	-1.92127348	-1.0123455	-1.66423278	0.76671450	-0.6492047
-0.9408270	-0.9872331	-0.30041247	-0.8859528	-0.75084436	0.28718500	-0.8058741
-0.7036891	-0.8340889	0.63486242	-1.0574858	-0.35107146	0.78001629	-1.0052717

نرمال سازی Decimal Scaling

کتابخانه‌ی **dprep** در این مورد نیز تابع ساده‌ای تحت عنوان **decscale** را برای نرمال سازی در اختیار ما قرار می‌دهد. که به هیچ طریقی آن را برای **mamba** نتوانستم نصب کنم. طبق تحقیق من این پکیج دیگر پشتیبانی نمی‌شود و از بیشتر ریپازیتوری‌ها حذف شده‌است. گرچه فرآیند نرمال سازی **decimal scaling** به خودی خود ساده است اما دانش کافی برای نوشتن تابعی در **R** را برای آن نداشتم و از میان دیگر دانشجویان کسی را پیدا نکردم که جوابی برایش داشته باشد، معذرت می‌خواهم.

```
In [2]: install.packages("dprep")
```

```
Warning message:
"package 'dprep' is not available for this version of R

A version of this package for your version of R might be available elsewhere,
see the ideas at
https://cran.r-project.org/doc/manuals/r-patched/R-admin.html#Installing-packages"
```

ماتریس فاصله اقلیدسی با پایتون

از آن جا که پایتون را از قبل نصب داشتیم و به عنوان کنجکاوی اقدام به محاسبه ماتریس عدم شباهت در محیط آن کردم، از کتابخانه‌ی `sklearn` برای تابع `euclidean_distances` و از کتابخانه‌ی `pandas` (که تحت عنوان `pd` آن را ایمپورت کردم) برای بارگذاری فایل `seeds.csv` دیتابیس استفاده شده است.

```
In [1]: from sklearn.metrics.pairwise import euclidean_distances
```

```
In [4]: import pandas as pd
seeds = pd.read_csv (r'D:\\Daneshga\\T8\\Amini\\Datasets\\Iris Alternatives\\Wheat Seeds Datas
euclidean_distances(seeds)
```

```
Out[4]: array([[ 0.          ,  1.66686202,  2.4723418 , ..., 207.11369863,
                208.05012156, 209.06479846],
               [ 1.66686202,  0.          ,  2.11924001, ..., 206.14804895,
                207.05348786, 208.08109398],
               [ 2.4723418 ,  2.11924001,  0.          , ..., 205.09017564,
                206.02888224, 207.04189552],
               ...,
               [207.11369863, 206.14804895, 205.09017564, ..., 0.          ,
                5.04620456,  3.48584882],
               [208.05012156, 207.05348786, 206.02888224, ..., 5.04620456,
                0.          ,  2.32600853],
               [209.06479846, 208.08109398, 207.04189552, ..., 3.48584882,
                2.32600853,  0.          ]])
```

خلاصه‌ای از فرم دیتاست و مفهوم Compactness:

	A	P	C	L	W	AC	LG	Class
1	15.26	14.84	0.871	5.763	3.312	2.221	5.22	1
2	14.88	14.57	0.8811	5.554	3.333	1.018	4.956	1
3	14.29	14.09	0.905	5.291	3.337	2.699	4.825	1
.
.
.
210	12.3	13.34	0.8684	5.243	2.974	5.637	5.063	3

1. A = Area مساحت
2. P = Perimeter محیط
3. C = Compactness دنسیتی/چگالی کل دانه
4. L = Length of Kernel طول هسته
5. W = Width of Kernel عرض هسته
6. AC = Asymmetry Coefficient ضریب عدم تقارن
7. LG = Length of Kernel Groove طول گروو دانه

Compactness به این صورت تعریف شده است:

$$C = \frac{4\pi A}{P^2}$$

$\pi = 3.14$ محیط p = perimeter (میلی متر) مساحت a = Area (میلی متر مربع)