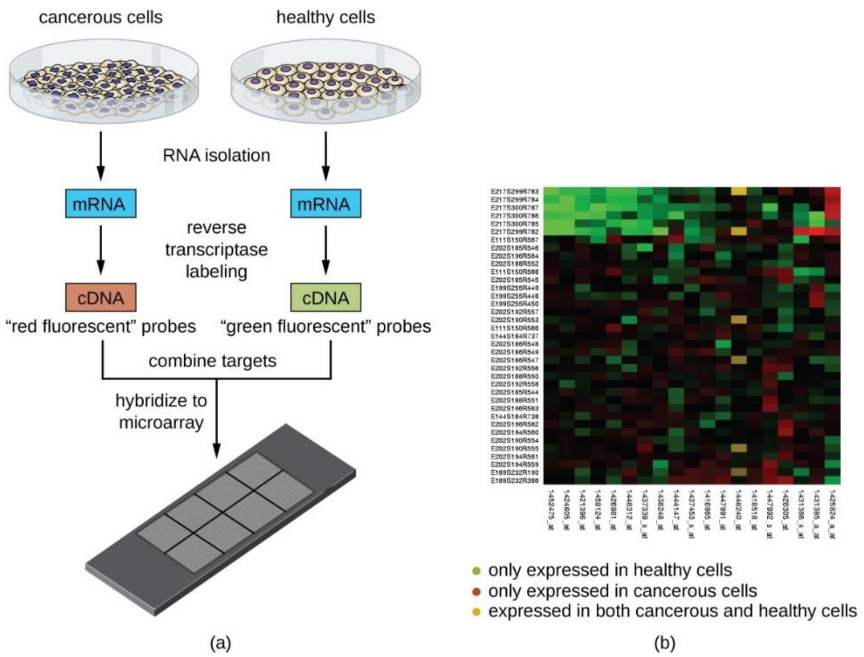
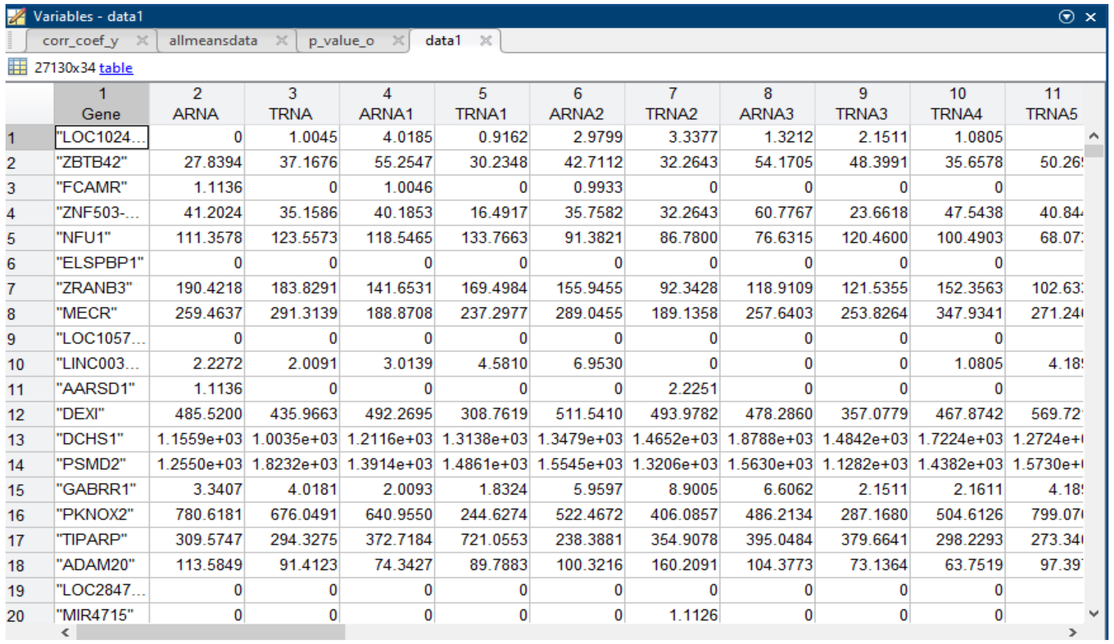
Introduction to Scientific Computation 113E



cancerous cells
healthy cells
RNA isolation
mRNA
mRNA
reverse transcriptase labeling
cDNA
cDNA
"red fluorescent" probes
"green fluorescent" probes
combine targets
hybridize to microarray
(a)
● only expressed in healthy cells
● only expressed in cancerous cells
● expressed in both cancerous and healthy cells
(b)



27130x34 table

	1 Gene	2 ARNA	3 TRNA	4 ARNA1	5 TRNA1	6 ARNA2	7 TRNA2	8 ARNA3	9 TRNA3	10 TRNA4	11 TRNA5
1	"LOC1024...	0	1.0045	4.0185	0.9162	2.9799	3.3377	1.3212	2.1511	1.0805	
2	"ZBTB42"	27.8394	37.1676	55.2547	30.2348	42.7112	32.2643	54.1705	48.3991	35.6578	50.26
3	"FCAMR"	1.1136	0	1.0046	0	0.9933	0	0	0	0	
4	"ZNF503-...	41.2024	35.1586	40.1853	16.4917	35.7582	32.2643	60.7767	23.6618	47.5438	40.84
5	"NFU1"	111.3578	123.5573	118.5465	133.7663	91.3821	86.7800	76.6315	120.4600	100.4903	68.07
6	"ELSPBP1"	0	0	0	0	0	0	0	0	0	
7	"ZRANB3"	190.4218	183.8291	141.6531	169.4984	155.9455	92.3428	118.9109	121.5355	152.3563	102.63
8	"MECR"	259.4637	291.3139	188.8708	237.2977	289.0455	189.1358	257.6403	253.8264	347.9341	271.24
9	"LOC1057...	0	0	0	0	0	0	0	0	0	
10	"LINC003...	2.2272	2.0091	3.0139	4.5810	6.9530	0	0	0	1.0805	4.18
11	"AARSD1"	1.1136	0	0	0	0	2.2251	0	0	0	
12	"DEXI"	485.5200	435.9663	492.2695	308.7619	511.5410	493.9782	478.2860	357.0779	467.8742	569.72
13	"DCHS1"	1.1559e+03	1.0035e+03	1.2116e+03	1.3138e+03	1.3479e+03	1.4652e+03	1.8788e+03	1.4842e+03	1.7224e+03	1.2724e+
14	"PSMD2"	1.2550e+03	1.8232e+03	1.3914e+03	1.4861e+03	1.5545e+03	1.3206e+03	1.5630e+03	1.1282e+03	1.4382e+03	1.5730e+
15	"GABRR1"	3.3407	4.0181	2.0093	1.8324	5.9597	8.9005	6.6062	2.1511	2.1611	4.18
16	"PKNOX2"	780.6181	676.0491	640.9550	244.6274	522.4672	406.0857	486.2134	287.1680	504.6126	799.07
17	"TIPARP"	309.5747	294.3275	372.7184	721.0553	238.3881	354.9078	395.0484	379.6641	298.2293	273.34
18	"ADAM20"	113.5849	91.4123	74.3427	89.7883	100.3216	160.2091	104.3773	73.1364	63.7519	97.39
19	"LOC2847...	0	0	0	0	0	0	0	0	0	
20	"MIR4715"	0	0	0	0	0	1.1126	0	0	0	

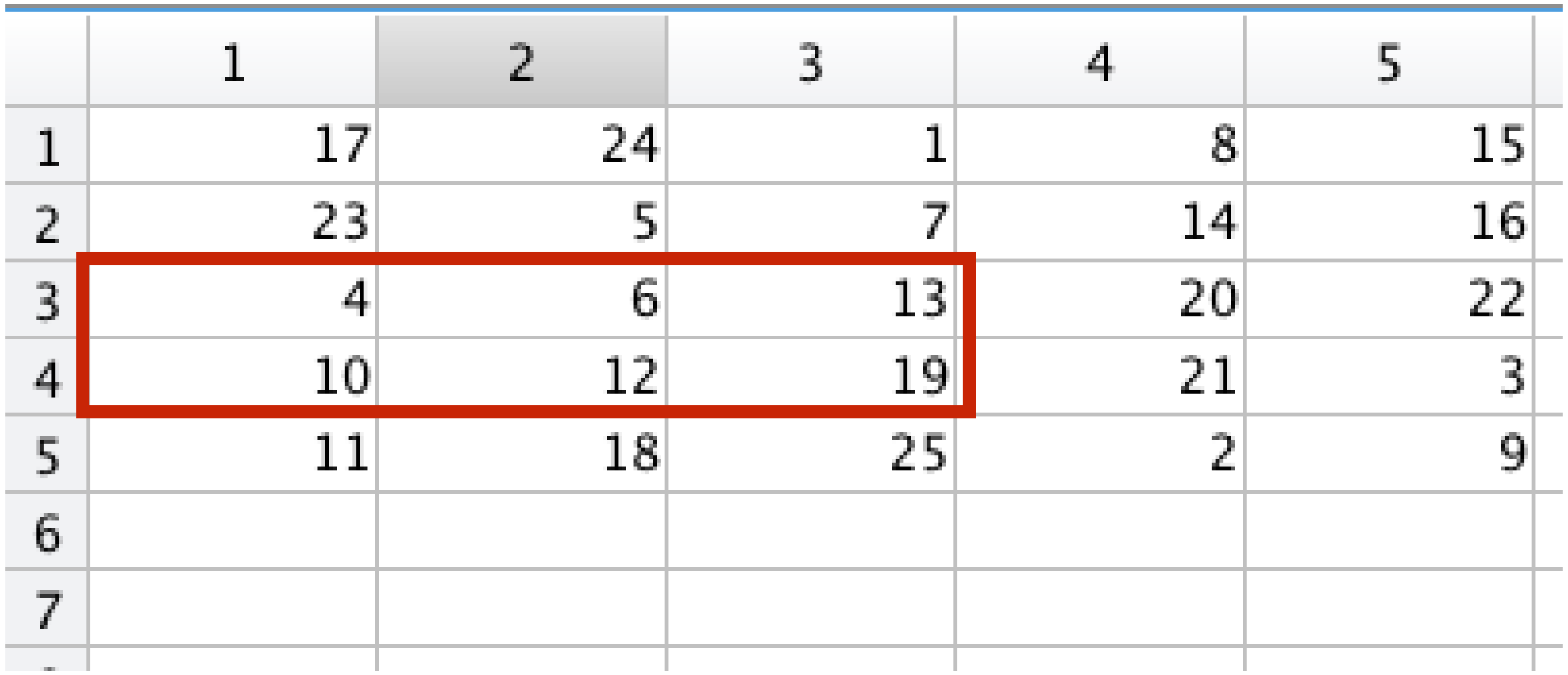
Assc. Prof. Halil Bayraktar

Lecture 3

from lecture 2

Column1 Column2

row 1
row 2



	1	2	3	4	5
1	17	24	1	8	15
2	23	5	7	14	16
3	4	6	13	20	22
4	10	12	19	21	3
5	11	18	25	2	9
6					
7					

Row Column

b=a([3,4],[1,2,3])

```
b =

   4    6   13
  10   12   19
```

from lecture 2

vector slicing

you can find a small section of array

```
b=examscore(1:10,1)


c=examscore(5:end,1)
d=examscore(1,:)
```

```
b =

78
73
64
80
80
76
60
89
86
76
```

```
c =

80
76
60
89
86
76
99
70
61
87
100
82
83
100
82
96
```

```
d =

78   88
```

from lecture 2

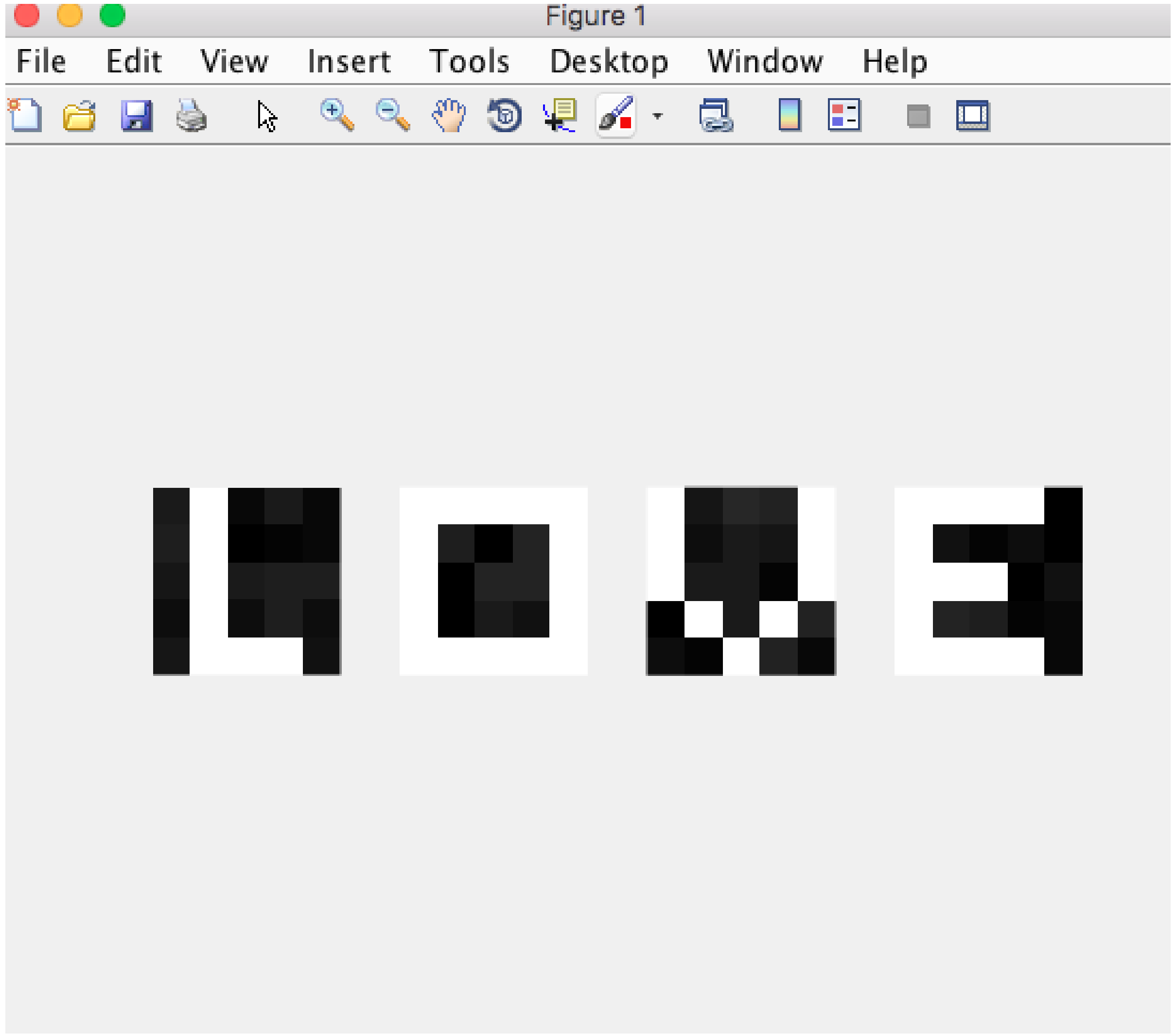
1D Arrays,

2D Arrays

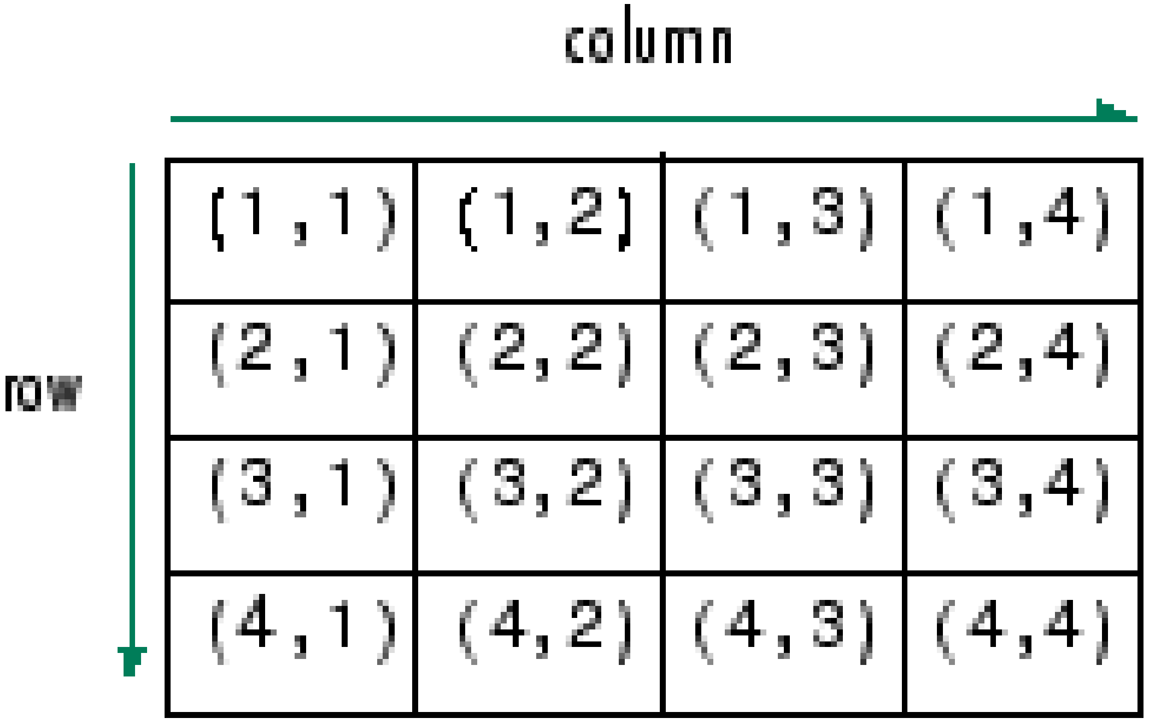
3D Arrays

Array Operation (+,-,x,/)

Mean, median, std, max, min, rand, Randi



An array having more than two dimensions is called a multidimensional array



(1,1)	(1,2)	(1,3)	(1,4)
(2,1)	(2,2)	(2,3)	(2,4)
(3,1)	(3,2)	(3,3)	(3,4)
(4,1)	(4,2)	(4,3)	(4,4)

```
x=[1 2;3 4]

x =

 1     2
 3     4
```

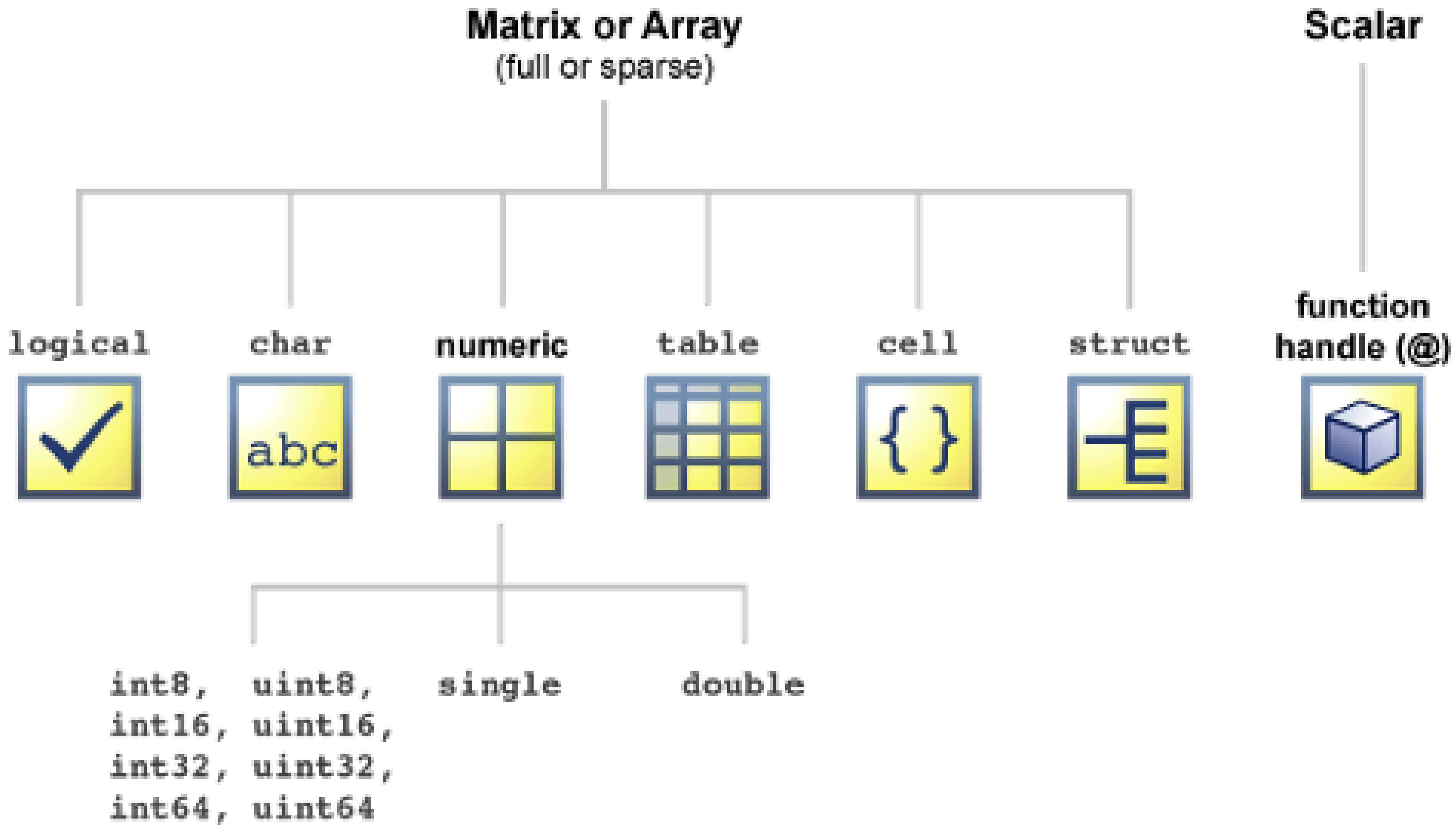
Exam scores

2 columns

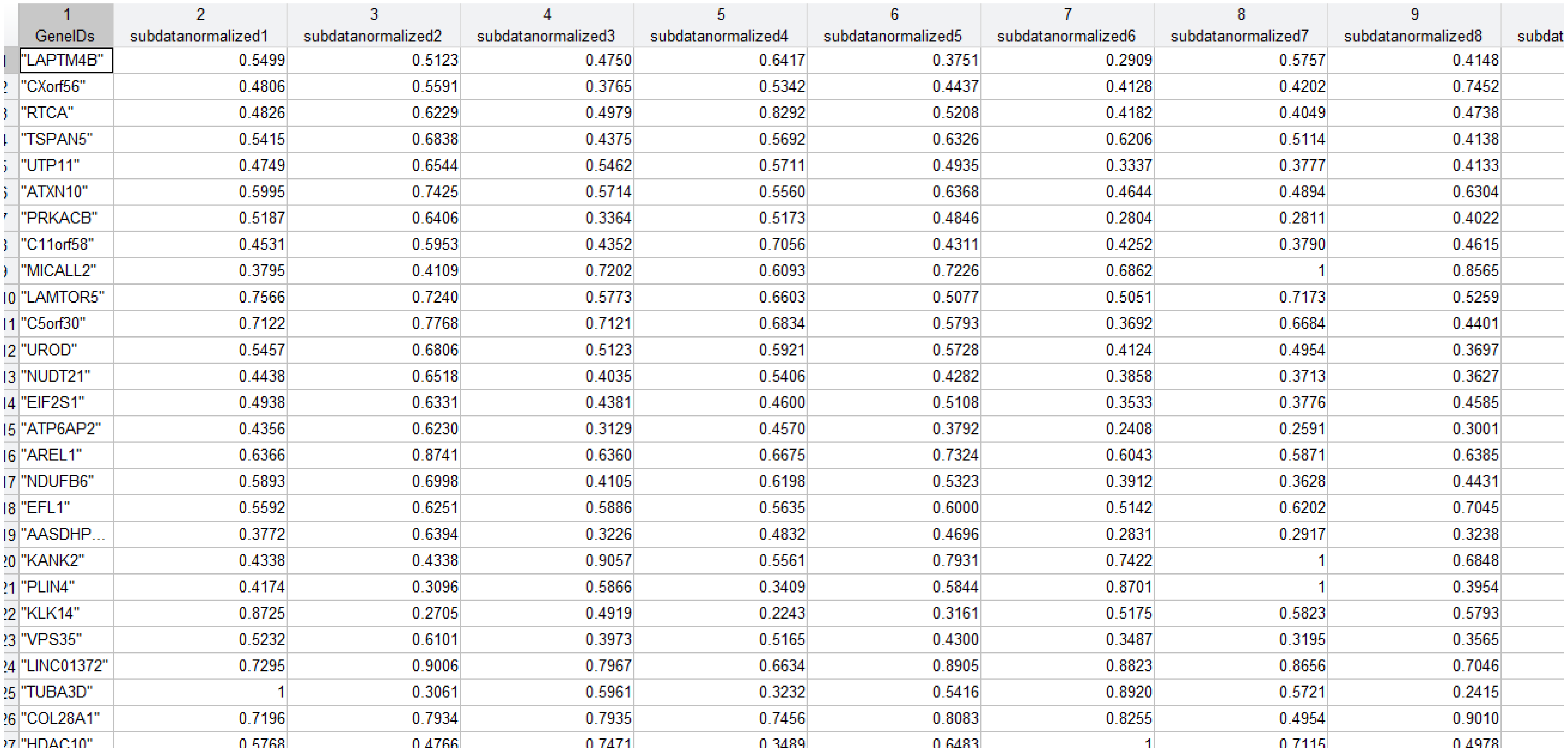
20 rows

MD1	MD2
71	61
86	88
71	97
72	70
82	100
66	96
99	70
69	91
63	99
62	70
99	90
77	83
69	100
92	75
67	71
65	74
65	80
84	96
70	62
94	65

Array types


Matrix or Array
(full or sparse)
Scalar
logical
char
abc
numeric
table
cell
{ }
struct
function handle (@)
int8, int16, int32, int64,
uint8, uint16, uint32, uint64
single
double

2d data
Microarray data cell type vs gene expression



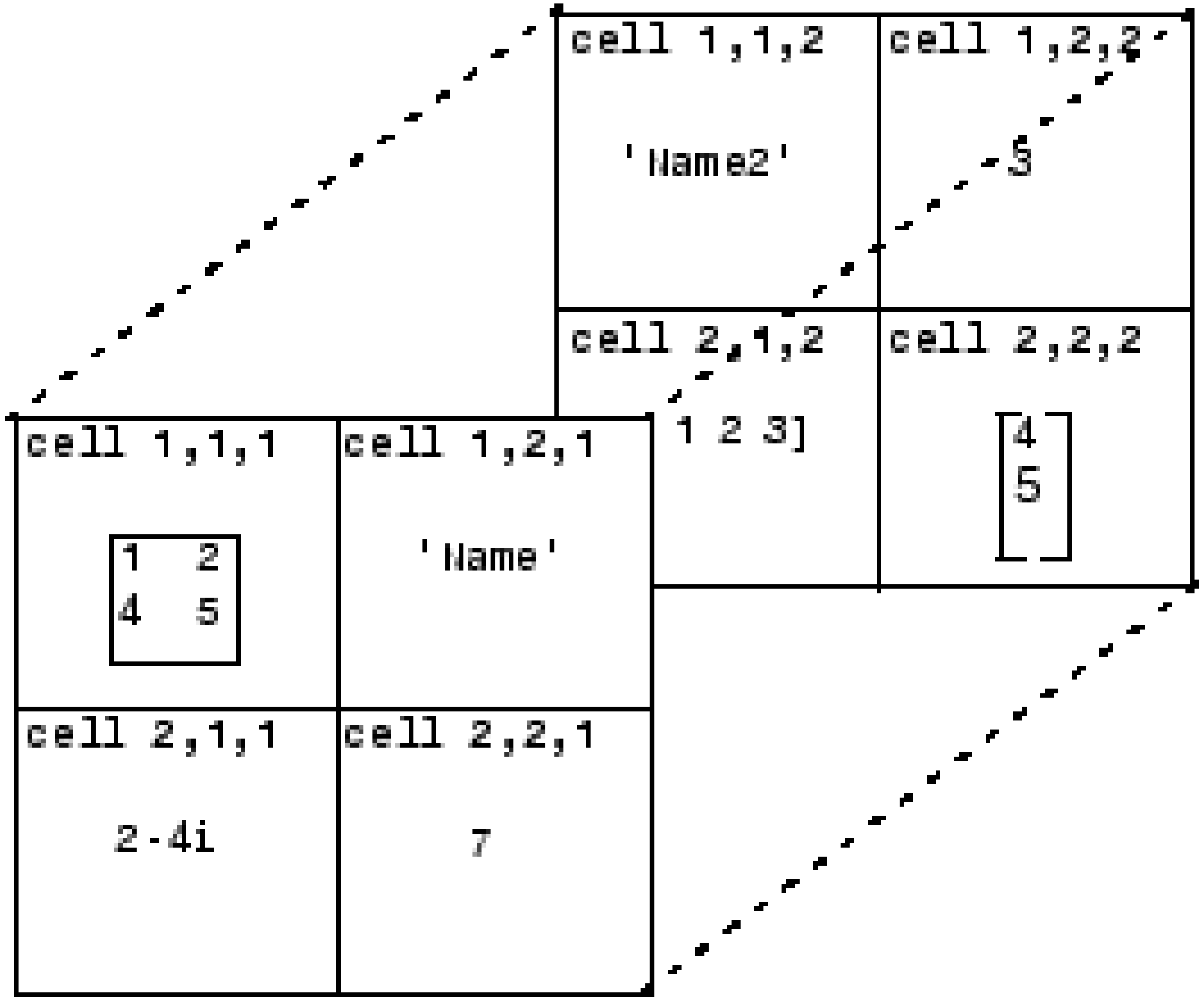
	1	2	3	4	5	6	7	8	9	
	GeneIDs	subdatanormalized1	subdatanormalized2	subdatanormalized3	subdatanormalized4	subdatanormalized5	subdatanormalized6	subdatanormalized7	subdatanormalized8	subdat
1	"LAPTM4B"	0.5499	0.5123	0.4750	0.6417	0.3751	0.2909	0.5757	0.4148	
2	"CXorf56"	0.4806	0.5591	0.3765	0.5342	0.4437	0.4128	0.4202	0.7452	
3	"RTCA"	0.4826	0.6229	0.4979	0.8292	0.5208	0.4182	0.4049	0.4738	
4	"TSPAN5"	0.5415	0.6838	0.4375	0.5692	0.6326	0.6206	0.5114	0.4138	
5	"UTP11"	0.4749	0.6544	0.5462	0.5711	0.4935	0.3337	0.3777	0.4133	
6	"ATXN10"	0.5995	0.7425	0.5714	0.5560	0.6368	0.4644	0.4894	0.6304	
7	"PRKACB"	0.5187	0.6406	0.3364	0.5173	0.4846	0.2804	0.2811	0.4022	
8	"C11orf58"	0.4531	0.5953	0.4352	0.7056	0.4311	0.4252	0.3790	0.4615	
9	"MICALL2"	0.3795	0.4109	0.7202	0.6093	0.7226	0.6862	1	0.8565	
10	"LAMTOR5"	0.7566	0.7240	0.5773	0.6603	0.5077	0.5051	0.7173	0.5259	
11	"C5orf30"	0.7122	0.7768	0.7121	0.6834	0.5793	0.3692	0.6684	0.4401	
12	"UROD"	0.5457	0.6806	0.5123	0.5921	0.5728	0.4124	0.4954	0.3697	
13	"NUDT21"	0.4438	0.6518	0.4035	0.5406	0.4282	0.3858	0.3713	0.3627	
14	"EIF2S1"	0.4938	0.6331	0.4381	0.4600	0.5108	0.3533	0.3776	0.4585	
15	"ATP6AP2"	0.4356	0.6230	0.3129	0.4570	0.3792	0.2408	0.2591	0.3001	
16	"AREL1"	0.6366	0.8741	0.6360	0.6675	0.7324	0.6043	0.5871	0.6385	
17	"NDUFB6"	0.5893	0.6998	0.4105	0.6198	0.5323	0.3912	0.3628	0.4431	
18	"EFL1"	0.5592	0.6251	0.5886	0.5635	0.6000	0.5142	0.6202	0.7045	
19	"AASDHP...	0.3772	0.6394	0.3226	0.4832	0.4696	0.2831	0.2917	0.3238	
20	"KANK2"	0.4338	0.4338	0.9057	0.5561	0.7931	0.7422	1	0.6848	
21	"PLIN4"	0.4174	0.3096	0.5866	0.3409	0.5844	0.8701	1	0.3954	
22	"KLK14"	0.8725	0.2705	0.4919	0.2243	0.3161	0.5175	0.5823	0.5793	
23	"VPS35"	0.5232	0.6101	0.3973	0.5165	0.4300	0.3487	0.3195	0.3565	
24	"LINC01372"	0.7295	0.9006	0.7967	0.6634	0.8905	0.8823	0.8656	0.7046	
25	"TUBA3D"	1	0.3061	0.5961	0.3232	0.5416	0.8920	0.5721	0.2415	
26	"COL28A1"	0.7196	0.7934	0.7935	0.7456	0.8083	0.8255	0.4954	0.9010	
27	"HDAC10"	0.5768	0.4766	0.7471	0.3489	0.6483	1	0.7115	0.4978	

27140 genes and 30 people, how to analyze data?

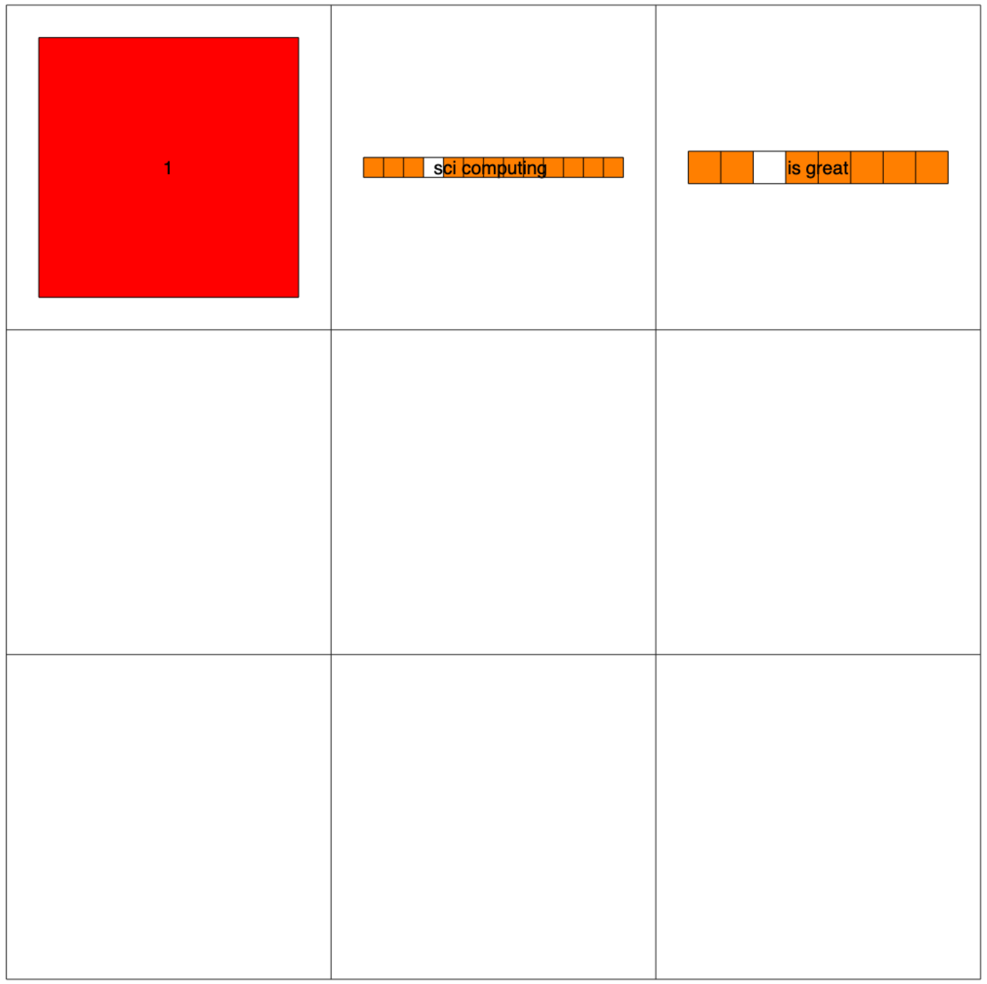
The dataset related with Alzheimer's disease was downloaded from Gene Expression Omnibus (http://www.ncbi.nlm.nih.gov/geo/): GSE104704.

Cell Arrays

Cell arrays are arrays of cells where each cell stores an array. Within a cell, elements must be the same type (because cells store arrays), but two cells may have different types.



```
y{1,1}=([1])
y{1,2}='sci computing'
y{1,3}='is great'
```



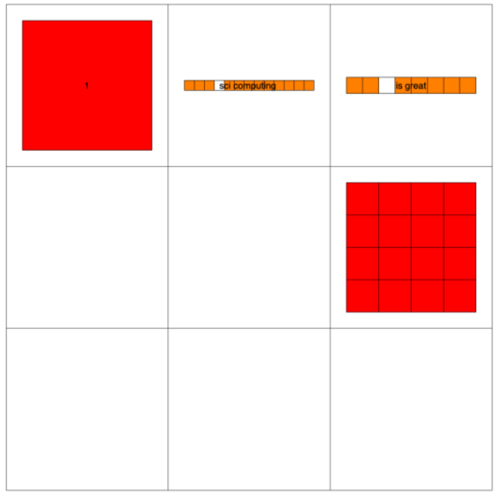
Cell Arrays

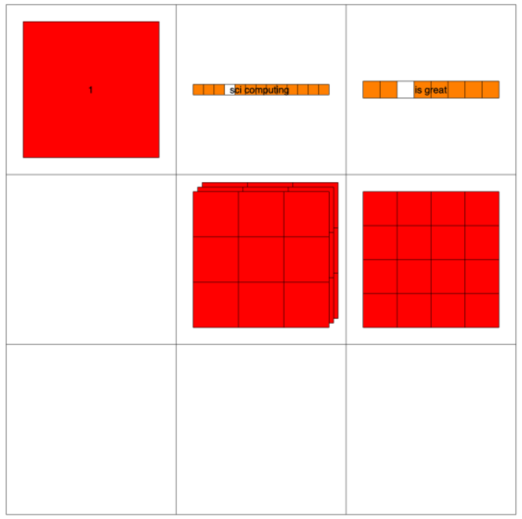
cellplot(C)

Add 3D array into a cell

```
y{2,3}=randi(4,4)
```

```
y{2,2}=ones(3,3,3)
```





```
C = {1,rand(1,10,2),'sci computing';
         "matlab",[3,4],{11; 22; 33}}
```

```
cellplot(C)
```

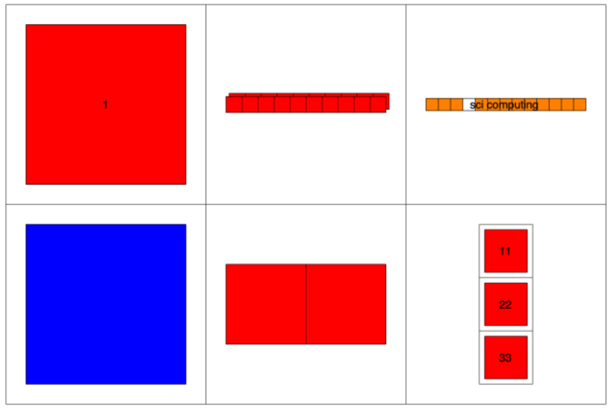
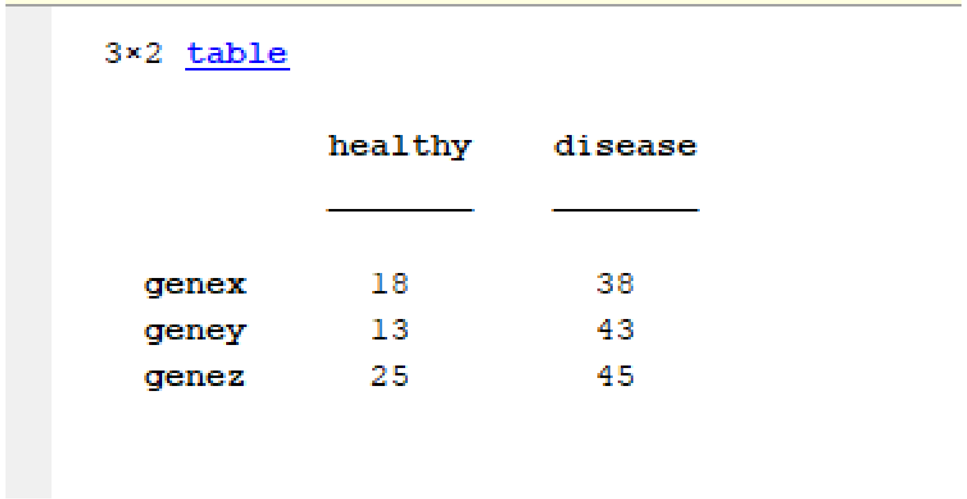


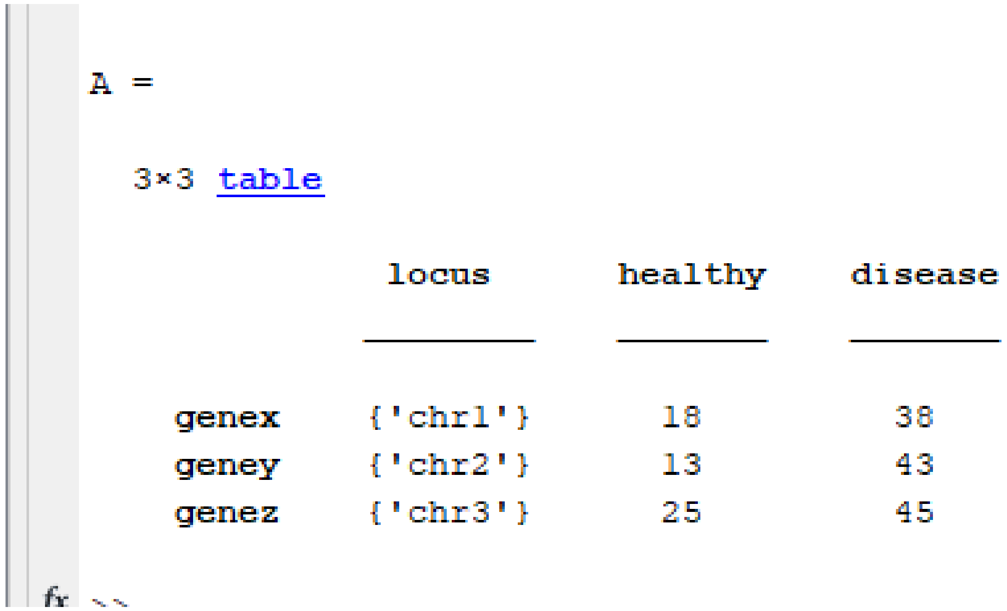
Table array

```
A = table([18;13;25],[38;43;45],...
                                    'VariableNames',{'healthy' 'disease'},...
                                    'RowNames',{'genex' 'geney' 'genez' })
```



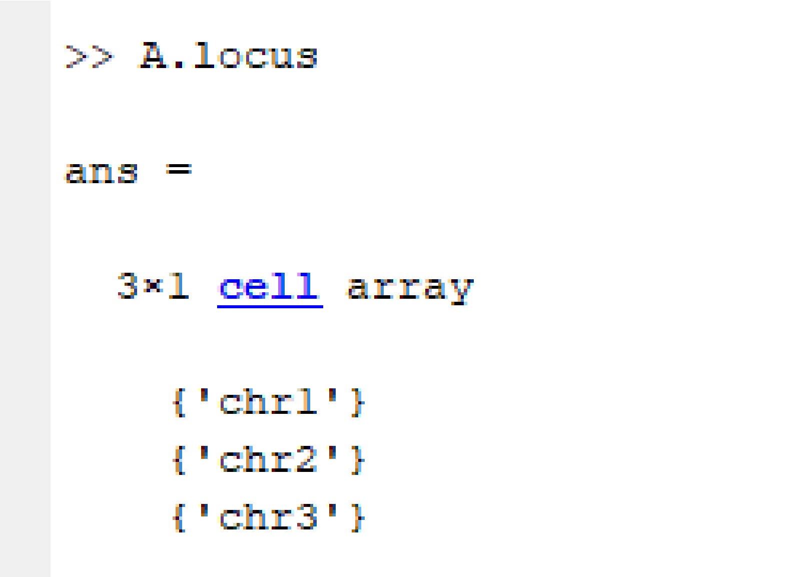
```
  3×2 table

              healthy    disease
              _______    _______

    genex       18         38
    geney       13         43
    genez       25         45
```

```
A = table({'chr1';'chr2';'chr3'}, [18;13;25],[38;43;45],...
    'VariableNames',{'locus','healthy' 'disease'},...
    'RowNames',{'genex' 'geney' 'genez' })
```

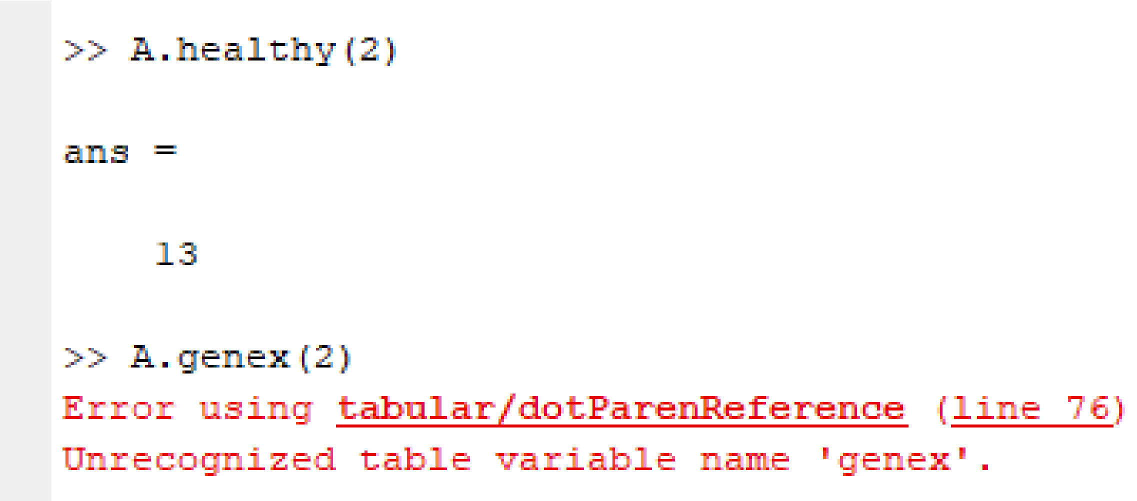


```
A =

  3×3 table

              locus       healthy    disease
             ________     _______    _______

    genex    {'chr1'}       18         38
    geney    {'chr2'}       13         43
    genez    {'chr3'}       25         45

fx >>
```

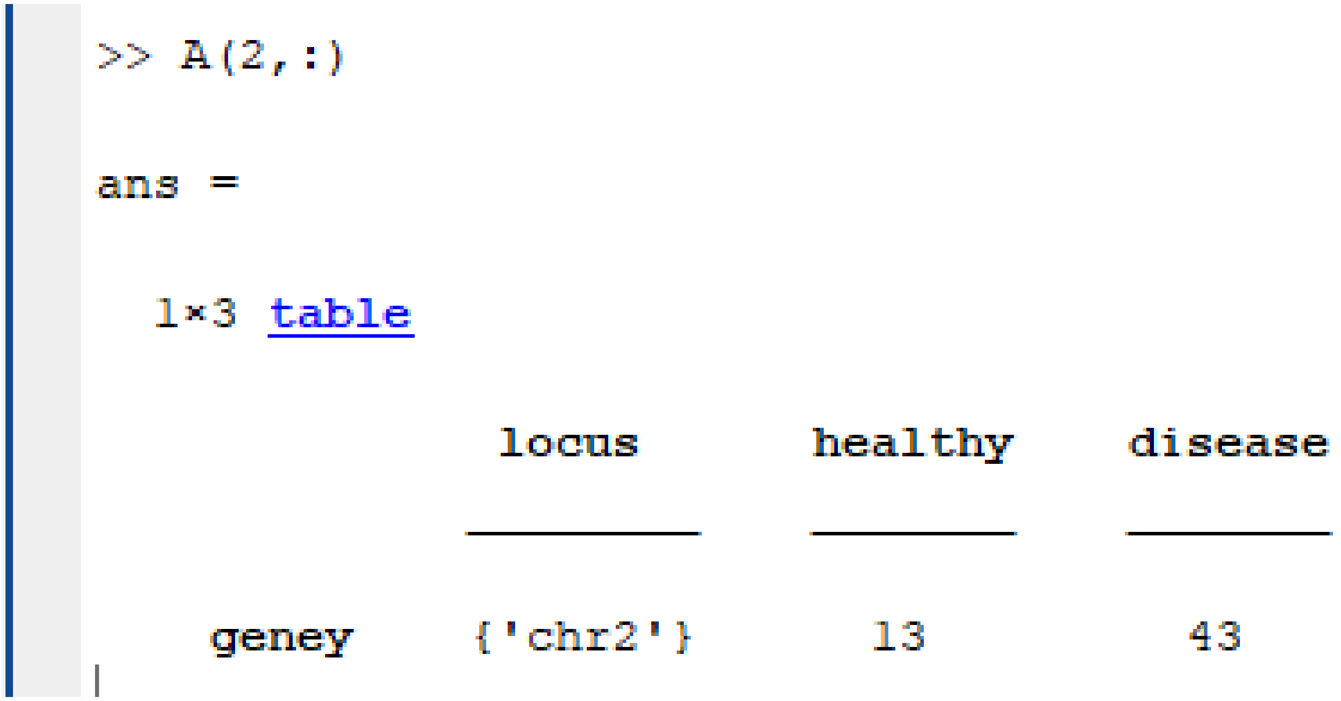
Table splicing



```
>> A.locus

ans =

  3×1 cell array

    {'chr1'}
    {'chr2'}
    {'chr3'}
```



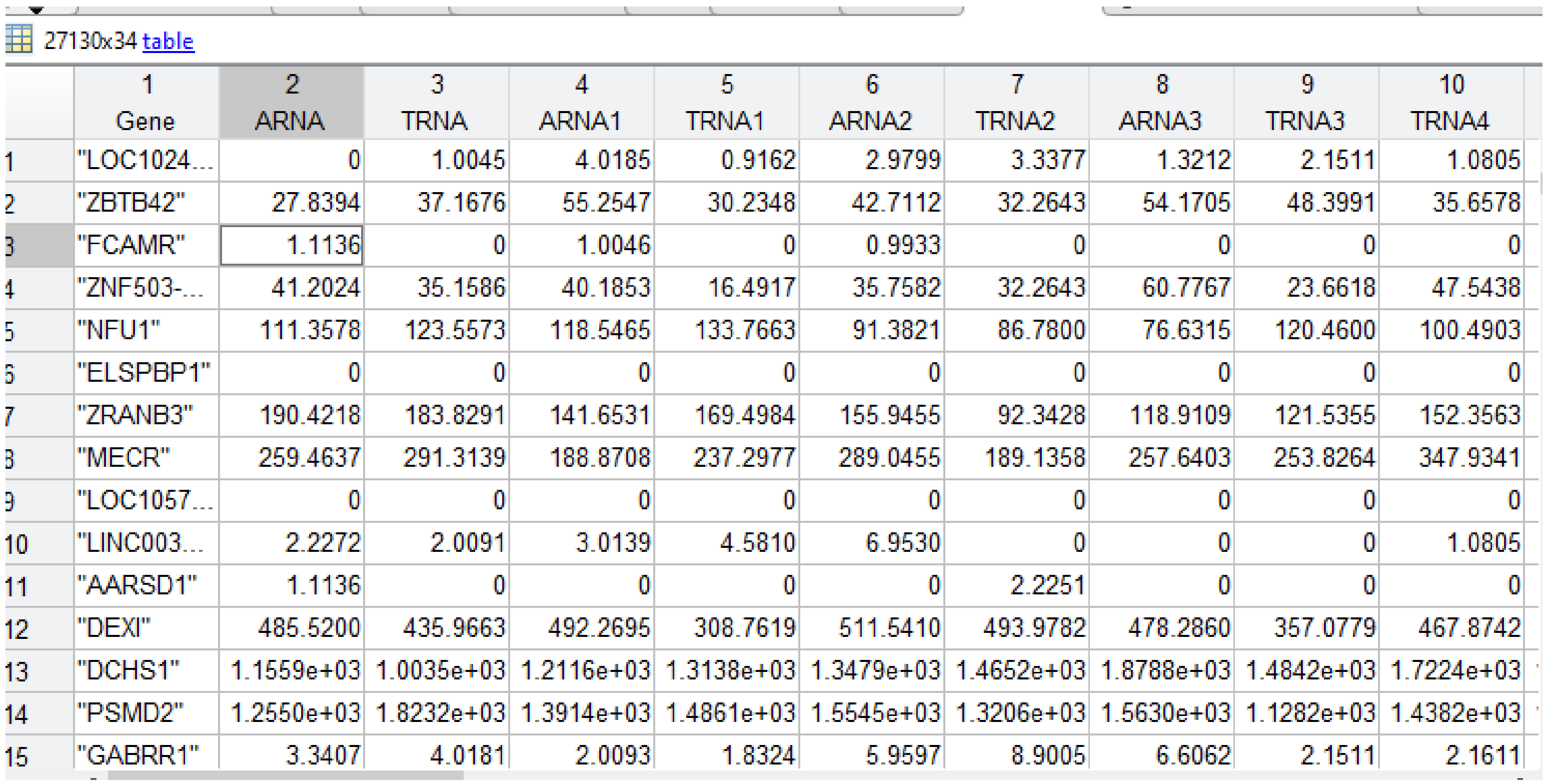
```
>> A.healthy(2)

ans =

    13

>> A.genex(2)
Error using tabular/dotParenReference (line 76)
Unrecognized table variable name 'genex'.
```



```
>> A(2,:)

ans =

  1×3 table

               locus      healthy    disease
             ________    _______    _______

    geney    {'chr2'}      13         43
```

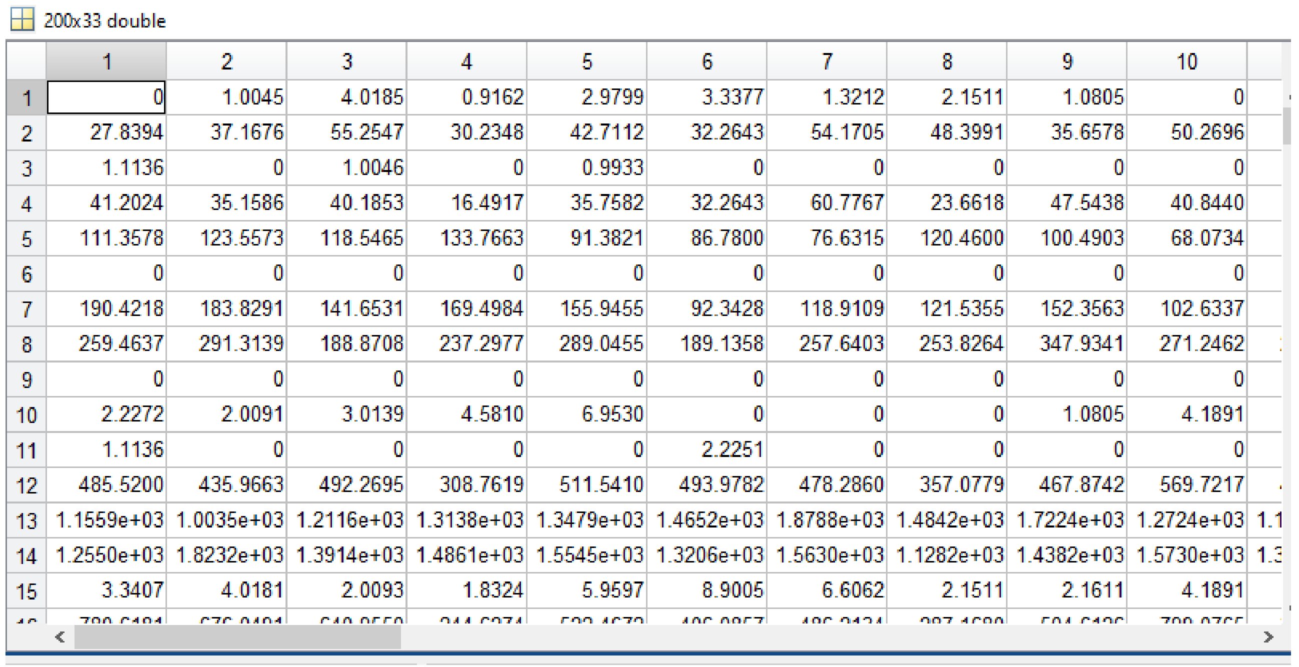
Example data1

- Most microarrays data are the table format
- subdata1(1:100,:)=table2array(data1(1:100,2:34));



27130x34 table

	1 Gene	2 ARNA	3 TRNA	4 ARNA1	5 TRNA1	6 ARNA2	7 TRNA2	8 ARNA3	9 TRNA3	10 TRNA4
1	"LOC1024...	0	1.0045	4.0185	0.9162	2.9799	3.3377	1.3212	2.1511	1.0805
2	"ZBTB42"	27.8394	37.1676	55.2547	30.2348	42.7112	32.2643	54.1705	48.3991	35.6578
3	"FCAMR"	1.1136	0	1.0046	0	0.9933	0	0	0	0
4	"ZNF503-...	41.2024	35.1586	40.1853	16.4917	35.7582	32.2643	60.7767	23.6618	47.5438
5	"NFU1"	111.3578	123.5573	118.5465	133.7663	91.3821	86.7800	76.6315	120.4600	100.4903
6	"ELSPBP1"	0	0	0	0	0	0	0	0	0
7	"ZRANB3"	190.4218	183.8291	141.6531	169.4984	155.9455	92.3428	118.9109	121.5355	152.3563
8	"MECR"	259.4637	291.3139	188.8708	237.2977	289.0455	189.1358	257.6403	253.8264	347.9341
9	"LOC1057...	0	0	0	0	0	0	0	0	0
10	"LINC003...	2.2272	2.0091	3.0139	4.5810	6.9530	0	0	0	1.0805
11	"AARSD1"	1.1136	0	0	0	0	2.2251	0	0	0
12	"DEXI"	485.5200	435.9663	492.2695	308.7619	511.5410	493.9782	478.2860	357.0779	467.8742
13	"DCHS1"	1.1559e+03	1.0035e+03	1.2116e+03	1.3138e+03	1.3479e+03	1.4652e+03	1.8788e+03	1.4842e+03	1.7224e+03
14	"PSMD2"	1.2550e+03	1.8232e+03	1.3914e+03	1.4861e+03	1.5545e+03	1.3206e+03	1.5630e+03	1.1282e+03	1.4382e+03
15	"GABRR1"	3.3407	4.0181	2.0093	1.8324	5.9597	8.9005	6.6062	2.1511	2.1611

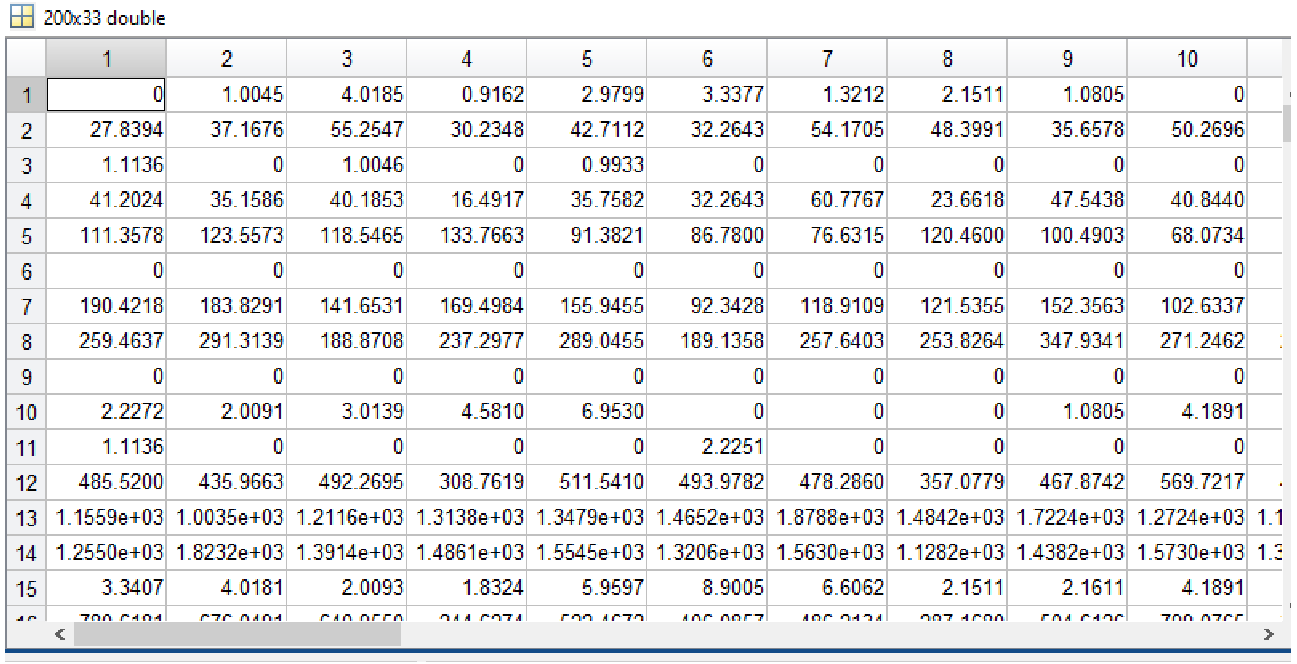


200x33 double

	1	2	3	4	5	6	7	8	9	10
1	0	1.0045	4.0185	0.9162	2.9799	3.3377	1.3212	2.1511	1.0805	0
2	27.8394	37.1676	55.2547	30.2348	42.7112	32.2643	54.1705	48.3991	35.6578	50.2696
3	1.1136	0	1.0046	0	0.9933	0	0	0	0	0
4	41.2024	35.1586	40.1853	16.4917	35.7582	32.2643	60.7767	23.6618	47.5438	40.8440
5	111.3578	123.5573	118.5465	133.7663	91.3821	86.7800	76.6315	120.4600	100.4903	68.0734
6	0	0	0	0	0	0	0	0	0	0
7	190.4218	183.8291	141.6531	169.4984	155.9455	92.3428	118.9109	121.5355	152.3563	102.6337
8	259.4637	291.3139	188.8708	237.2977	289.0455	189.1358	257.6403	253.8264	347.9341	271.2462
9	0	0	0	0	0	0	0	0	0	0
10	2.2272	2.0091	3.0139	4.5810	6.9530	0	0	0	1.0805	4.1891
11	1.1136	0	0	0	0	2.2251	0	0	0	0
12	485.5200	435.9663	492.2695	308.7619	511.5410	493.9782	478.2860	357.0779	467.8742	569.7217
13	1.1559e+03	1.0035e+03	1.2116e+03	1.3138e+03	1.3479e+03	1.4652e+03	1.8788e+03	1.4842e+03	1.7224e+03	1.2724e+03
14	1.2550e+03	1.8232e+03	1.3914e+03	1.4861e+03	1.5545e+03	1.3206e+03	1.5630e+03	1.1282e+03	1.4382e+03	1.5730e+03
15	3.3407	4.0181	2.0093	1.8324	5.9597	8.9005	6.6062	2.1511	2.1611	4.1891

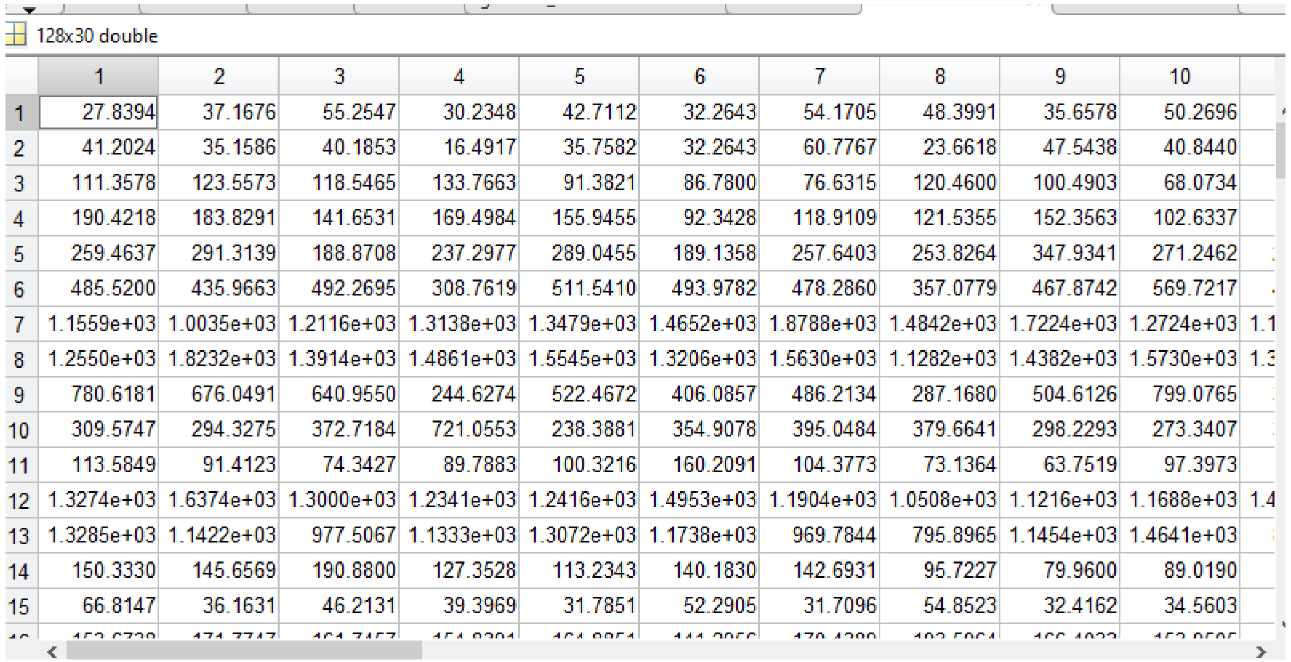
How to remove missing data?

subcleandata1= rmmissing(subdata1)



200x33 double

	1	2	3	4	5	6	7	8	9	10
1	0	1.0045	4.0185	0.9162	2.9799	3.3377	1.3212	2.1511	1.0805	0
2	27.8394	37.1676	55.2547	30.2348	42.7112	32.2643	54.1705	48.3991	35.6578	50.2696
3	1.1136	0	1.0046	0	0.9933	0	0	0	0	0
4	41.2024	35.1586	40.1853	16.4917	35.7582	32.2643	60.7767	23.6618	47.5438	40.8440
5	111.3578	123.5573	118.5465	133.7663	91.3821	86.7800	76.6315	120.4600	100.4903	68.0734
6	0	0	0	0	0	0	0	0	0	0
7	190.4218	183.8291	141.6531	169.4984	155.9455	92.3428	118.9109	121.5355	152.3563	102.6337
8	259.4637	291.3139	188.8708	237.2977	289.0455	189.1358	257.6403	253.8264	347.9341	271.2462
9	0	0	0	0	0	0	0	0	0	0
10	2.2272	2.0091	3.0139	4.5810	6.9530	0	0	0	1.0805	4.1891
11	1.1136	0	0	0	0	2.2251	0	0	0	0
12	485.5200	435.9663	492.2695	308.7619	511.5410	493.9782	478.2860	357.0779	467.8742	569.7217
13	1.1559e+03	1.0035e+03	1.2116e+03	1.3138e+03	1.3479e+03	1.4652e+03	1.8788e+03	1.4842e+03	1.7224e+03	1.2724e+03
14	1.2550e+03	1.8232e+03	1.3914e+03	1.4861e+03	1.5545e+03	1.3206e+03	1.5630e+03	1.1282e+03	1.4382e+03	1.5730e+03
15	3.3407	4.0181	2.0093	1.8324	5.9597	8.9005	6.6062	2.1511	2.1611	4.1891

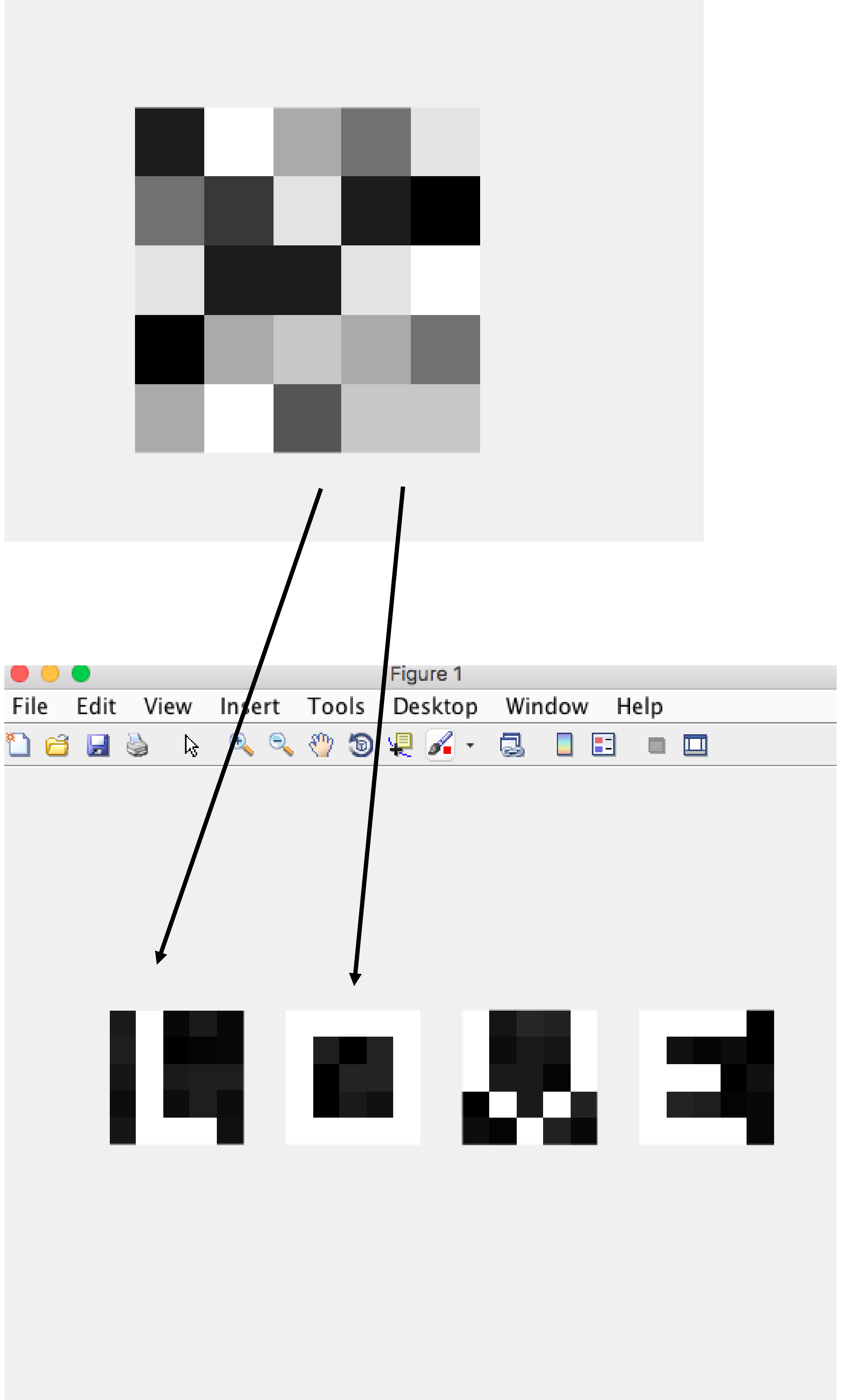


128x30 double

	1	2	3	4	5	6	7	8	9	10
1	27.8394	37.1676	55.2547	30.2348	42.7112	32.2643	54.1705	48.3991	35.6578	50.2696
2	41.2024	35.1586	40.1853	16.4917	35.7582	32.2643	60.7767	23.6618	47.5438	40.8440
3	111.3578	123.5573	118.5465	133.7663	91.3821	86.7800	76.6315	120.4600	100.4903	68.0734
4	190.4218	183.8291	141.6531	169.4984	155.9455	92.3428	118.9109	121.5355	152.3563	102.6337
5	259.4637	291.3139	188.8708	237.2977	289.0455	189.1358	257.6403	253.8264	347.9341	271.2462
6	485.5200	435.9663	492.2695	308.7619	511.5410	493.9782	478.2860	357.0779	467.8742	569.7217
7	1.1559e+03	1.0035e+03	1.2116e+03	1.3138e+03	1.3479e+03	1.4652e+03	1.8788e+03	1.4842e+03	1.7224e+03	1.2724e+03
8	1.2550e+03	1.8232e+03	1.3914e+03	1.4861e+03	1.5545e+03	1.3206e+03	1.5630e+03	1.1282e+03	1.4382e+03	1.5730e+03
9	780.6181	676.0491	640.9550	244.6274	522.4672	406.0857	486.2134	287.1680	504.6126	799.0765
10	309.5747	294.3275	372.7184	721.0553	238.3881	354.9078	395.0484	379.6641	298.2293	273.3407
11	113.5849	91.4123	74.3427	89.7883	100.3216	160.2091	104.3773	73.1364	63.7519	97.3973
12	1.3274e+03	1.6374e+03	1.3000e+03	1.2341e+03	1.2416e+03	1.4953e+03	1.1904e+03	1.0508e+03	1.1216e+03	1.1688e+03
13	1.3285e+03	1.1422e+03	977.5067	1.1333e+03	1.3072e+03	1.1738e+03	969.7844	795.8965	1.1454e+03	1.4641e+03
14	150.3330	145.6569	190.8800	127.3528	113.2343	140.1830	142.6931	95.7227	79.9600	89.0190
15	66.8147	36.1631	46.2131	39.3969	31.7851	52.2905	31.7096	54.8523	32.4162	34.5603

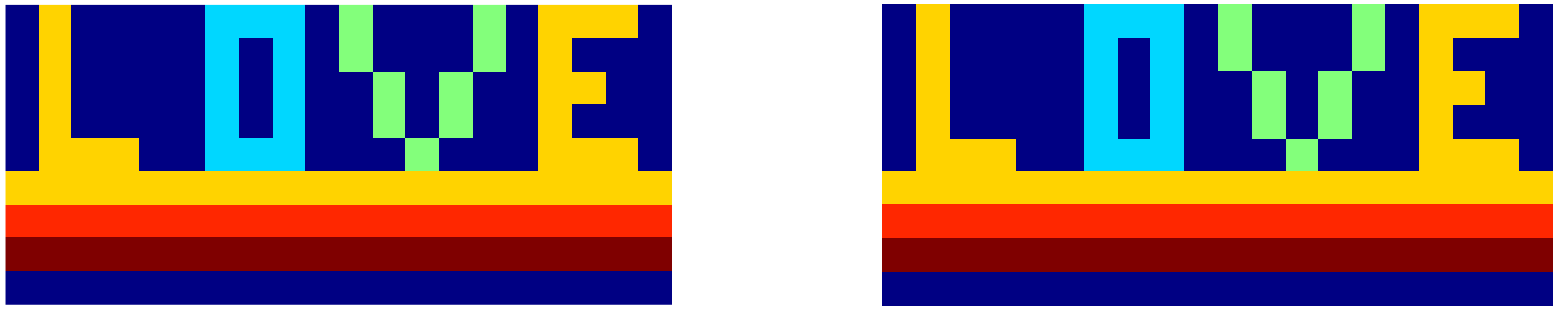
There are many different ways to remove missing data from an array we will learn them after we learn for loops and decision control (if).
Rmmmissing is a build in function

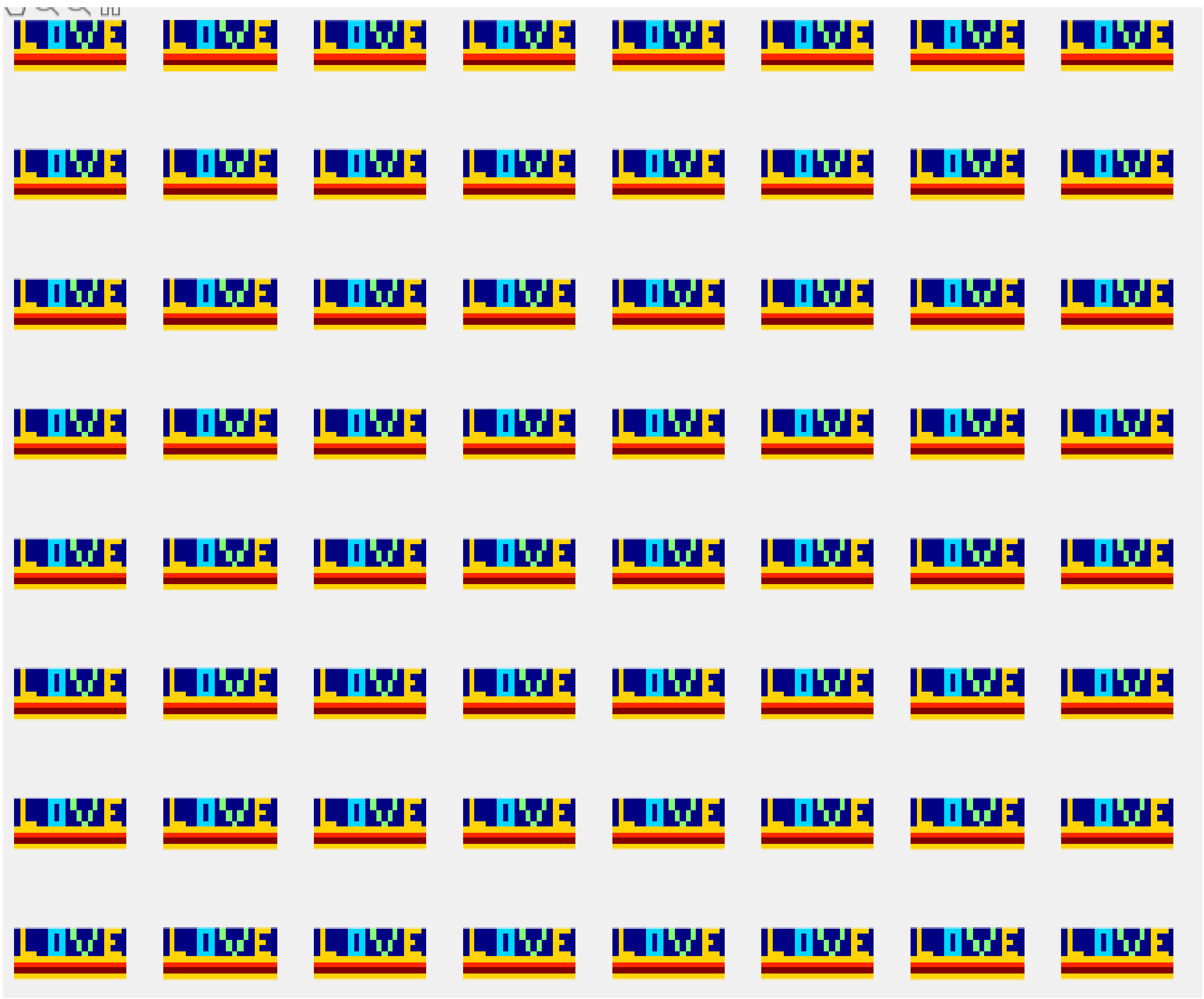
How can we imagine Arrays in 2D?

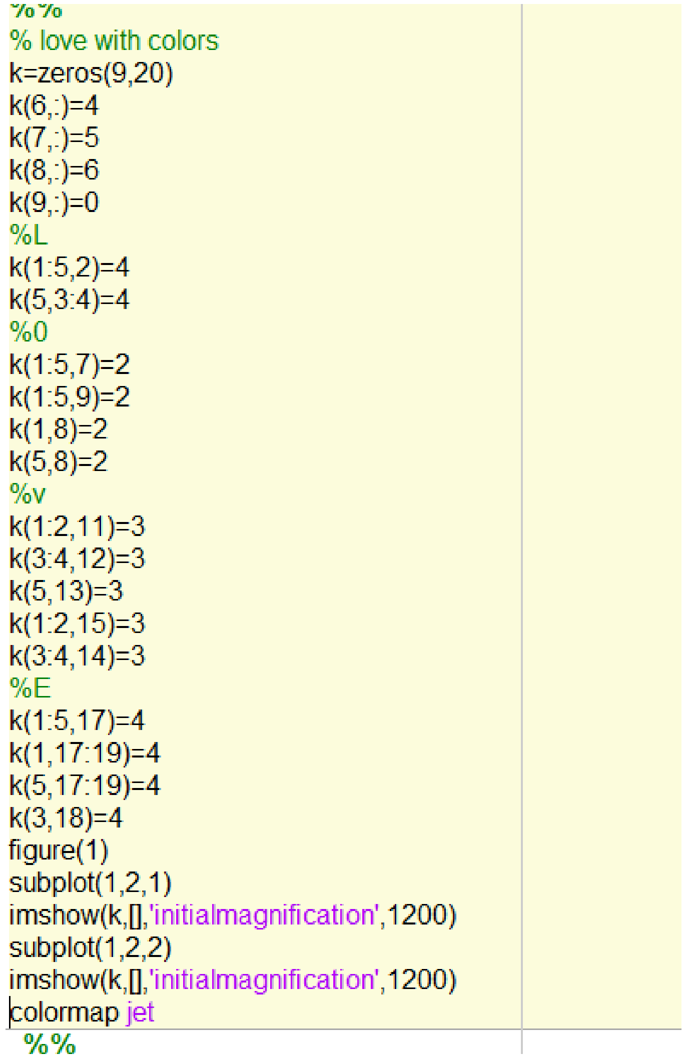


```
%print LOVE
y=randi(10,5,5)
y(:,2)=60
y(5,2:4)=60
figure(1)
subplot(1,4,1)
imshow(rot90(y,4),[],'initialmagnification',600)
z=randi(10,5,5)
z(:,1)=60
z(1,:)=60
z(5,:)=60
z(:,5)=60
subplot(1,4,2)
imshow(z,[],'initialmagnification',600)
k=randi(10,5,5)
k(1:3,1)=60
k(4,2)=60
k(5,3)=60
k(4,4)=60
k(1:3,5)=60
subplot(1,4,3)
imshow(k,[],'initialmagnification',600)
t=randi(10,5,5)
t(1,1:4)=60
t(:,1)=60
t(5,1:4)=60
t(3,1:3)=60
subplot(1,4,4)
imshow(t,[],'initialmagnification',600)
```

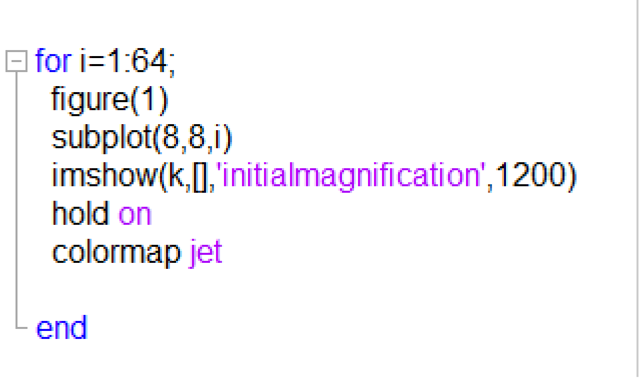
How can we imagine Arrays in 2D? Can we print Love with many colors? Can we print mant of them?





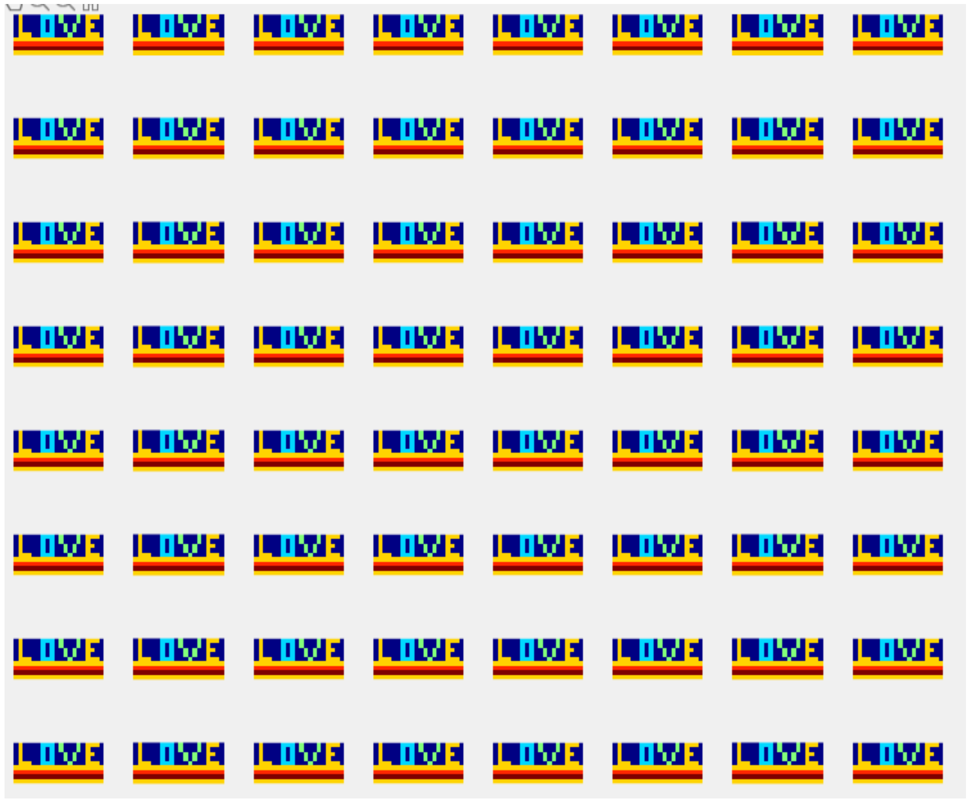


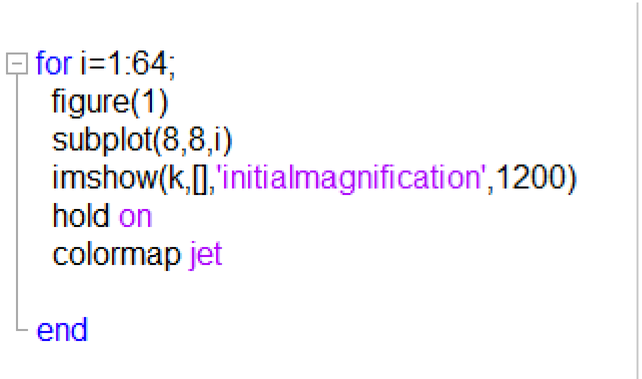
```
%%
% love with colors
k=zeros(9,20)
k(6,:)=4
k(7,:)=5
k(8,:)=6
k(9,:)=0
%L
k(1:5,2)=4
k(5,3:4)=4
%0
k(1:5,7)=2
k(1:5,9)=2
k(1,8)=2
k(5,8)=2
%v
k(1:2,11)=3
k(3:4,12)=3
k(5,13)=3
k(1:2,15)=3
k(3:4,14)=3
%E
k(1:5,17)=4
k(1,17:19)=4
k(5,17:19)=4
k(3,18)=4
figure(1)
subplot(1,2,1)
imshow(k,[],'initialmagnification',1200)
subplot(1,2,2)
imshow(k,[],'initialmagnification',1200)
colormap jet
 %%
```



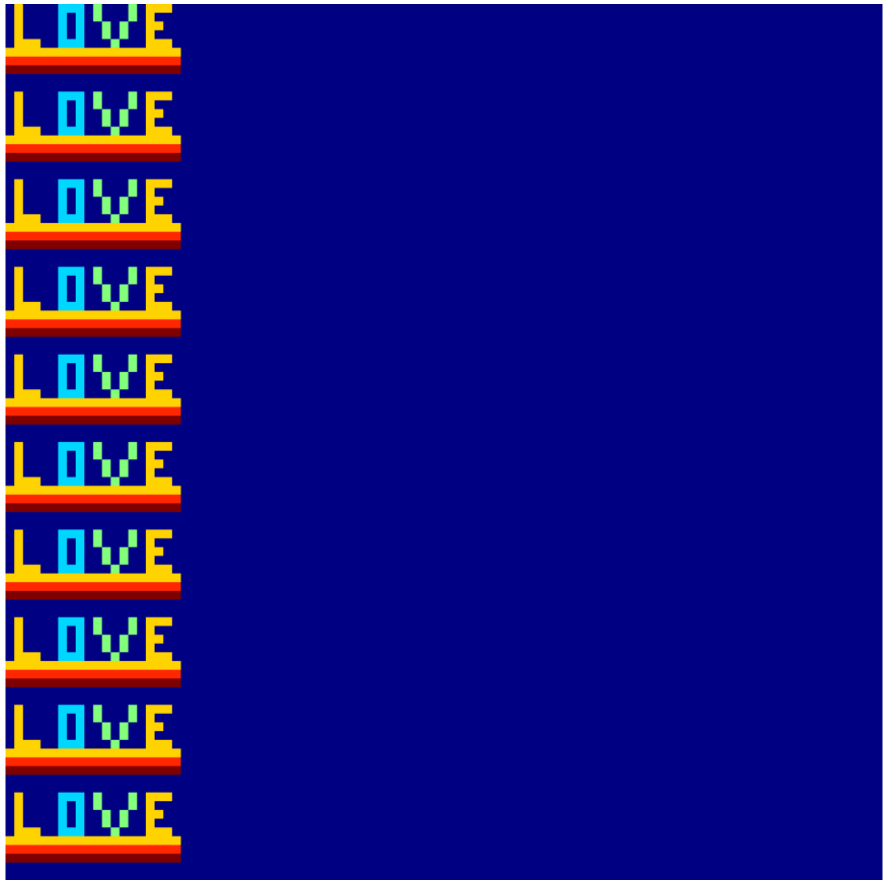
```
for i=1:64;
 figure(1)
 subplot(8,8,i)
 imshow(k,[],'initialmagnification',1200)
 hold on
 colormap jet

end
```

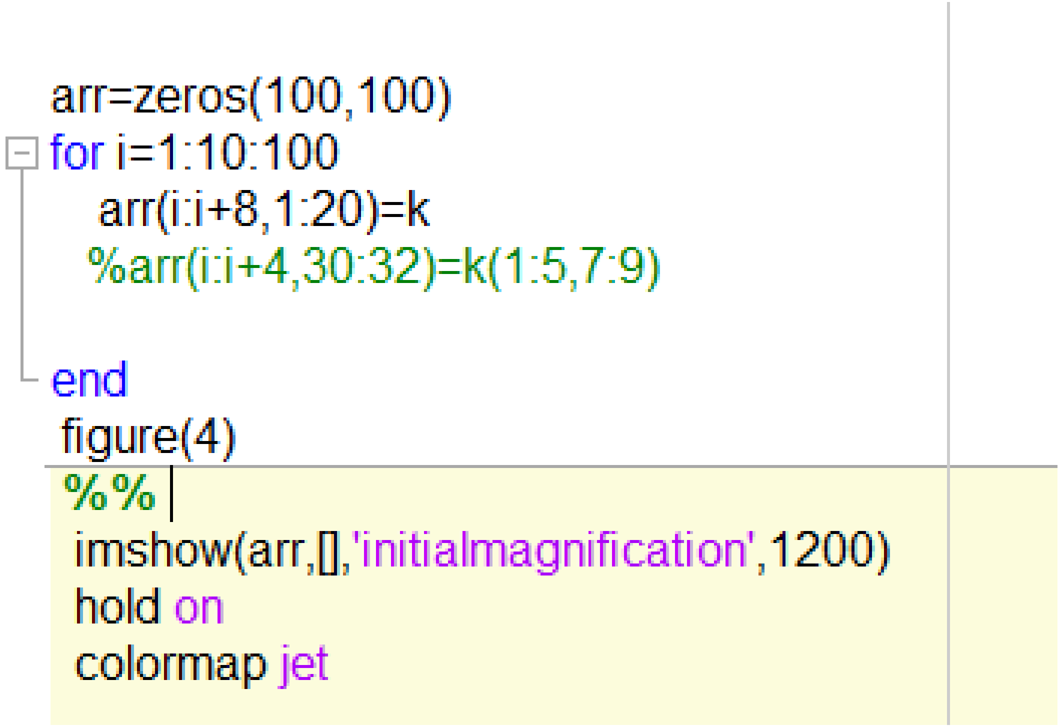
Can we organize arrays with different ways?





```
for i=1:64;
  figure(1)
  subplot(8,8,i)
  imshow(k,[],'initialmagnification',1200)
  hold on
  colormap jet

end
```





```
arr=zeros(100,100)
for i=1:10:100
    arr(i:i+8,1:20)=k
   %arr(i:i+4,30:32)=k(1:5,7:9)

end
 figure(4)
 %%
  imshow(arr,[],'initialmagnification',1200)
  hold on
  colormap jet
```

Sorting data

80	82
91	73
89	71
74	87
68	73
86	100
77	81
82	98
94	81
65	100
62	89
74	60
84	99
90	79
76	98
78	87
80	81
76	76
97	67
63	99

Finding and selecting elements in a matrix

```
z=magic(4)
z([1,2],[3,4])
z(1,1)
z(2)
```

```
z =

    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
```

```
ans =

     3    13
    10     8
```

```
ans =

    16
```

```
ans =

     5
```

```
z =

                      16     2     3    13
Index number 2         5    11    10     8
                       9     7     6    12
                       4    14    15     1
```

Rounding the elements of an array to the nearest integer

```
x=1.5001
floor(x)
ceil(x)
```

```
ans =

     1

ans =

     2
```

```
x=rand(3)
ceil(x)
```

```
x =

    0.4254    0.9915    0.1293
    0.9842    0.7764    0.7471
    0.9800    0.3136    0.6842
```

```
ans =

     1     1     1
     1     1     1
     1     1     1
```

```
round(x)
```

```
x =

    3.3468    3.0285    1.0143
    2.2344    3.6283    3.4876
    2.2025    1.8237    1.0050
```

```
ans =

     3     3     1
     2     4     3
     2     2     1
```

Sorting rows

```
examscores =

    94    60
    65    88
    80    82
   100    77
    67    81
    95    70
    62    97
    65    88
    76    74
    60    65
```

```
sortrows(examscores)

ans =

    60    65
    62    97
    65    88
    65    88
    67    81
    76    74
    80    82
    94    60
    95    70
   100    77
```

```
sortrows(examscores,2)

ans =

    94    60
    60    65
    95    70
    76    74
   100    77
    67    81
    80    82
    65    88
    65    88
    62    97
```

Find an information in an array

returns the row and column indices of non-zero entries in a matrix.

```
1   2   2
4   6   9
1   10  9
```

```
[row,col,v]=find(x>7)

ans =

6
8
9
```

```
[row,col,v]=find(x>7)

row =
3
2
3
col =
2
3
3
v =
3×1 logical array
1
1
1
```

Data sorting

sort the elements of each column in a particular order.

```
examscores =

98    76    71    83    70    85    89    83    71    63
```

```
sort(x,'ascend')          ans =

    63    70    71    71    76    83    83    85    89    98
```

```
sort(x,'descend')         ans =

    98    89    85    83    83    76    71    71    70    63
```

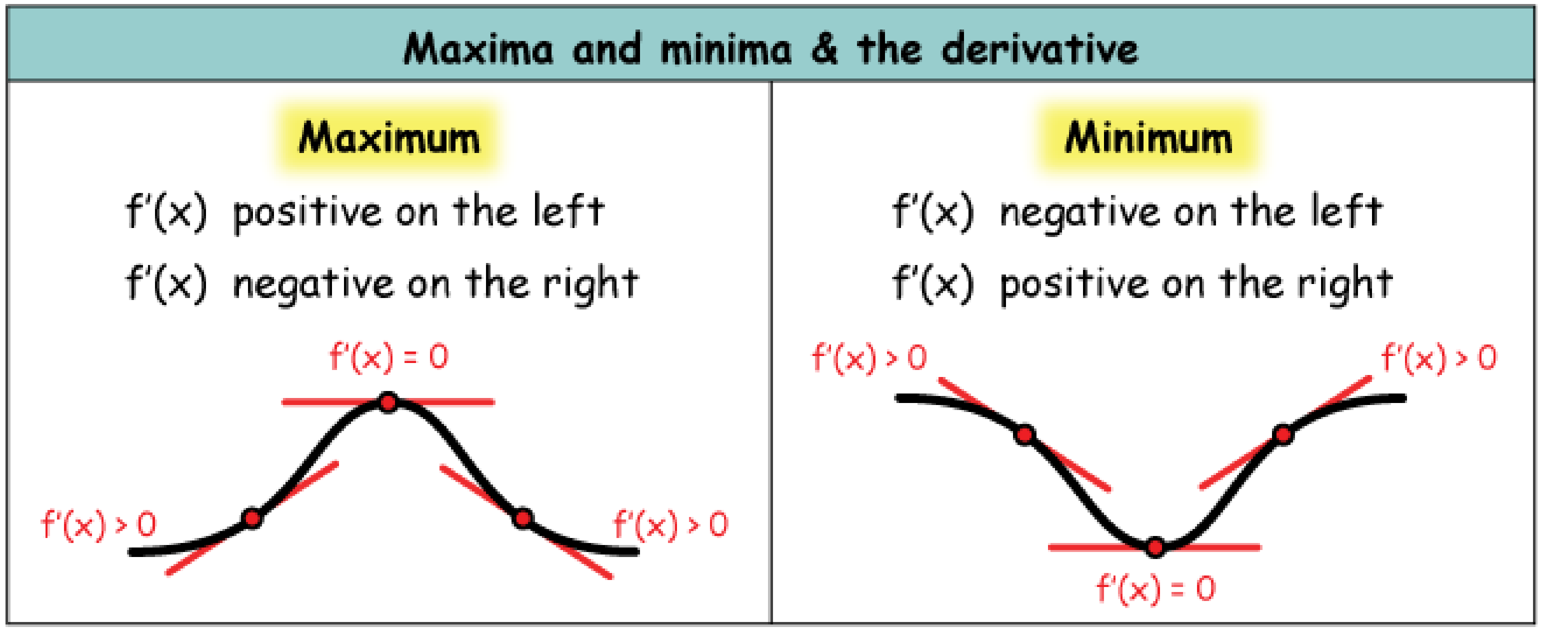
Finding anomaly in the data. This is an harder problem for teaching the computer to find the outliers.

a=[5,8,3,6,7,200, 10, 12, 295, 34, 250]

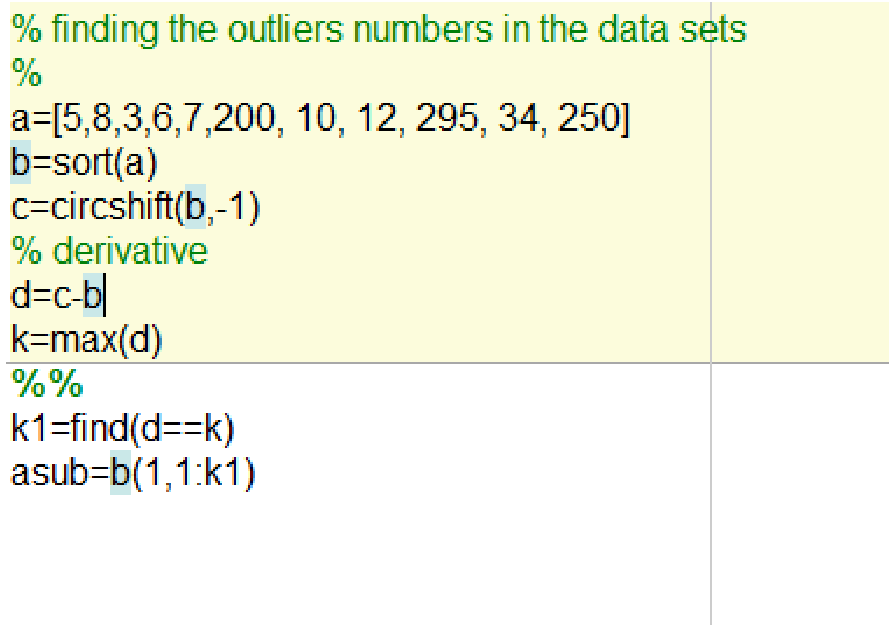
b = 3 5 6 7 8 10 12 34 200 250 295

You should take the first derivative of the function. How can you take the first derivative with matlab (circshift)?

Protocol: 1. sort the data
2. take the first derivative
3. Find the max and its index number
4. Use the index number and find the subdata?

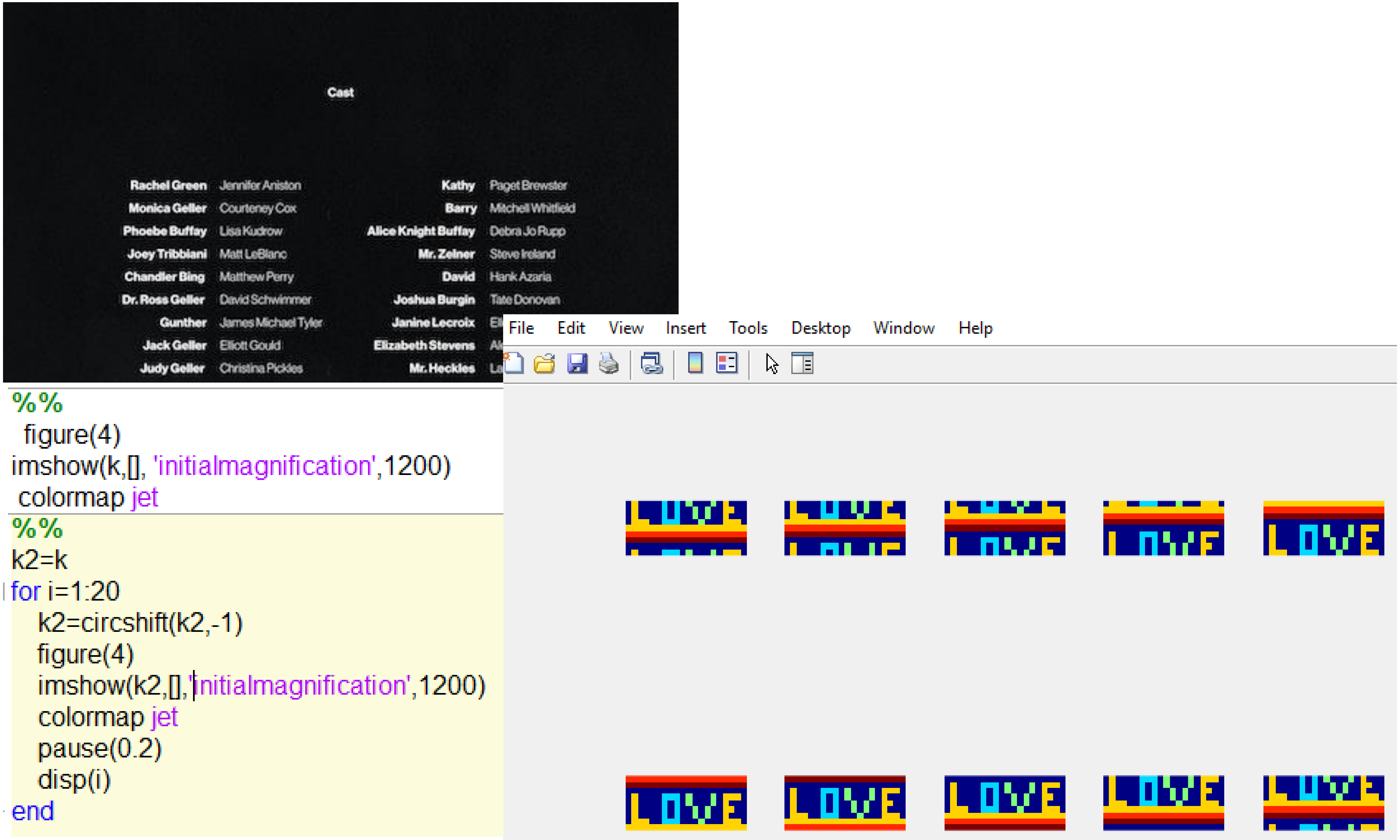


Maxima and minima & the derivative
Maximum
f′(x) positive on the left
f′(x) negative on the right
f′(x) = 0
f′(x) > 0
f′(x) > 0
Minimum
f′(x) negative on the left
f′(x) positive on the right
f′(x) > 0
f′(x) > 0
f′(x) = 0



```
% finding the outliers numbers in the data sets
%
a=[5,8,3,6,7,200, 10, 12, 295, 34, 250]
b=sort(a)
c=circshift(b,-1)
% derivative
d=c-b
k=max(d)
%%
k1=find(d==k)
asub=b(1,1:k1)
```

Lets do fun with circshift, rolling the writings: Design an animation for film credits

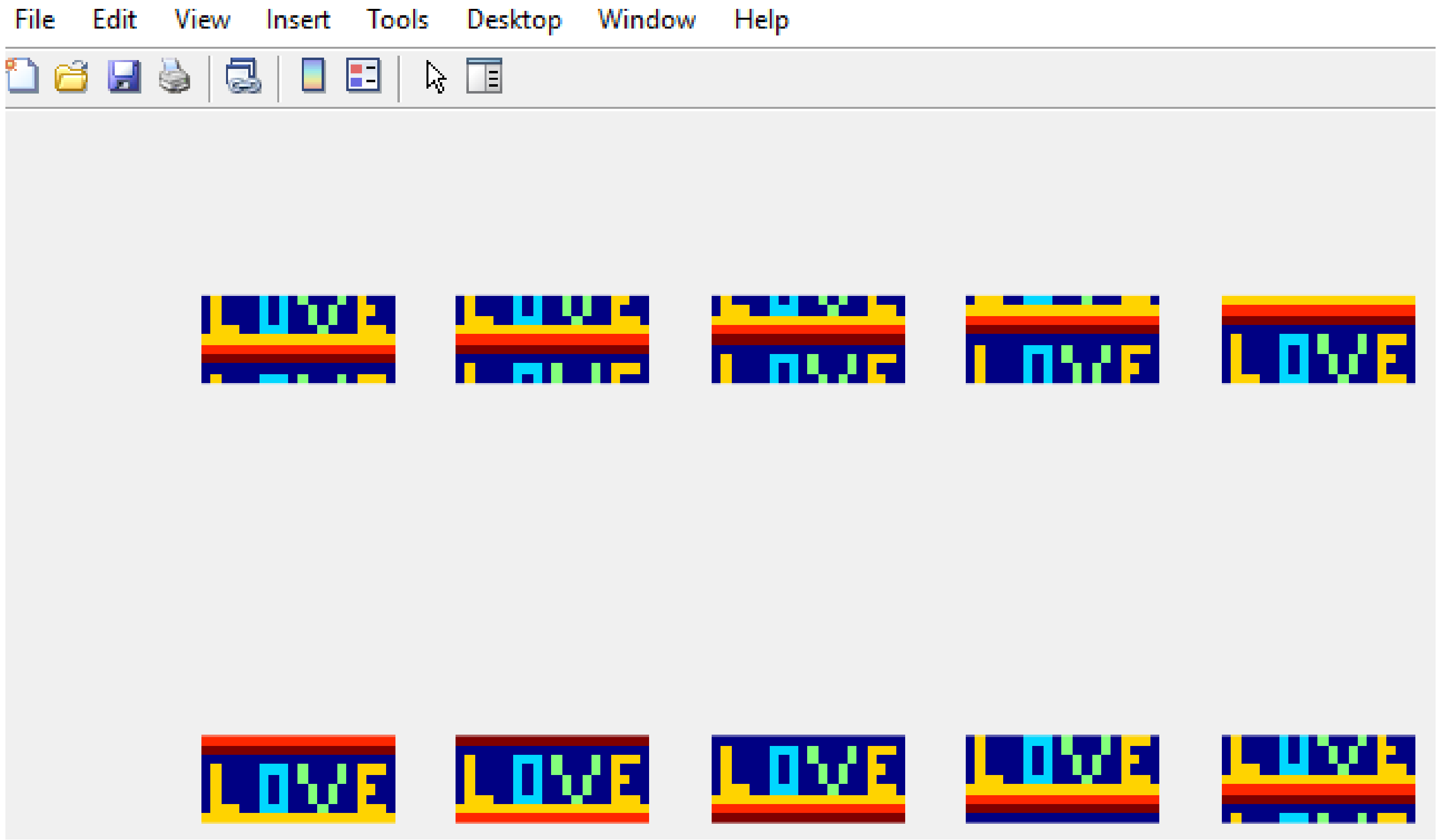


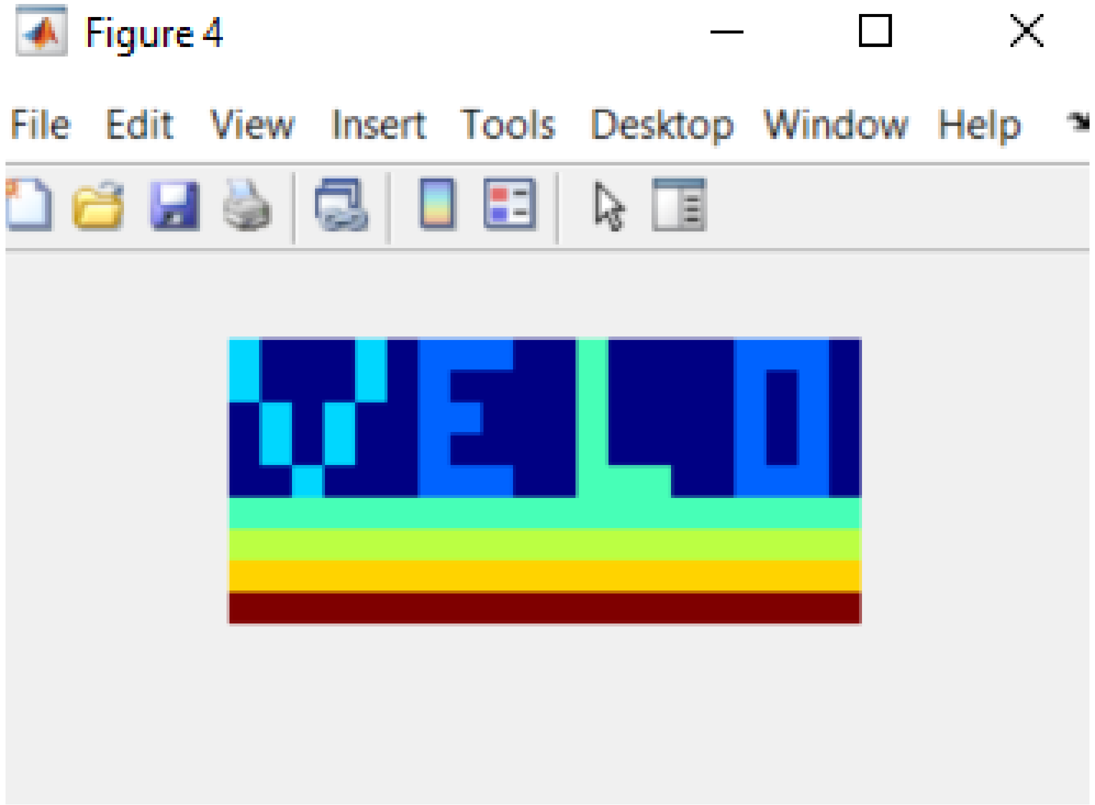
Cast

Rachel Green	Jennifer Aniston	Kathy	Paget Brewster
Monica Geller	Courteney Cox	Barry	Mitchell Whitfield
Phoebe Buffay	Lisa Kudrow	Alice Knight Buffay	Debra Jo Rupp
Joey Tribbiani	Matt LeBlanc	Mr. Zelner	Steve Ireland
Chandler Bing	Matthew Perry	David	Hank Azaria
Dr. Ross Geller	David Schwimmer	Joshua Burgin	Tate Donovan
Gunther	James Michael Tyler	Janine Lecroix	El
Jack Geller	Elliott Gould	Elizabeth Stevens	Al
Judy Geller	Christina Pickles	Mr. Heckles	La

```
%%
 figure(4)
imshow(k,[], 'initialmagnification',1200)
 colormap jet
%%
k2=k
for i=1:20
   k2=circshift(k2,-1)
   figure(4)
   imshow(k2,[],'initialmagnification',1200)
   colormap jet
   pause(0.2)
   disp(i)
end
```

Circle data in rows and columns





Compare genes and find unmatched nucleotides

```
geneA='AAAATAGTAGATGATGATGATGTCCATATAT'

geneB='AAAATATGTAATTGTATGGATGTCCATATAT'

[row,col,v]=find(geneA~=geneB)
```

Reshaping a Matrix

The number of rows and columns in a matrix can be changed provided the total number of elements remains the same.

```
a=randi([1,10],3,3)
 b=reshape(a,9,1)
```

```
2   2   8
7   3   2
1   8   3
```

```
2
7
1
2
3
8
8
2
3
```

```
b=reshape(a,1,9)
```

```
 8   5   7
10  8   10
9   9   10
```

```
8   10 9   5   8   9   7   10 10
```