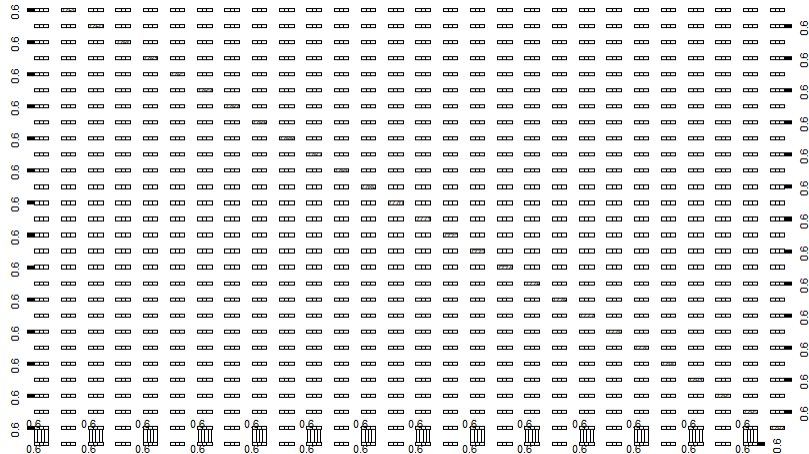


Khan Inan

Train.csv

This table is the plot of row 1 of the data frame

```
plot(train[1, 2:28])
par(new=TRUE)
plot(train[1, 29:56])
par(new=TRUE)
plot(train[1, 57:84])
par(new=TRUE)
plot(train[1, 85:112])
par(new=TRUE)
plot(train[1, 113:140])
par(new=TRUE)
plot(train[1, 141:168])
par(new=TRUE)
plot(train[1, 169:196])
par(new=TRUE)
plot(train[1, 197:224])
par(new=TRUE)
plot(train[1, 225:252])
par(new=TRUE)
plot(train[1, 253:280])
par(new=TRUE)
plot(train[1, 281:308])
par(new=TRUE)
plot(train[1, 309:336])
par(new=TRUE)
plot(train[1, 337:364])
par(new=TRUE)
plot(train[1, 365:392])
par(new=TRUE)
plot(train[1, 393:420])
par(new=TRUE)
plot(train[1, 421:448])
par(new=TRUE)
plot(train[1, 476:504])
par(new=TRUE)
plot(train[1, 505:532])
par(new=TRUE)
plot(train[1, 533:560])
par(new=TRUE)
plot(train[1, 561:588])
par(new=TRUE)
```



Here on the left is my code for the trains CSV file. And so as you can see I have 28 lines of code each in increments/multiples of 28 (28,56,84) etc . And this ends up with a 28x28 square of values with 784 total values. And in between I included par(new=TRUE) because I felt this was the best way to plot all these values on the same graph instead of 28 separate graphs

```

plot(train[1, 589:616])
par(new=TRUE)
plot(train[1, 617:644])
par(new=TRUE)
plot(train[1, 645:672])
par(new=TRUE)
plot(train[1, 673:700])
par(new=TRUE)
plot(train[1, 701:728])
par(new=TRUE)
plot(train[1, 729:756])
par(new=TRUE)
plot(train[1, 757:784])

```

2.

```

trainmatrix <- apply(as.matrix.noquote(train),
                     2,
                     as.numeric)

```

#here I had to change the non-numeric values within the train csv to numeric because the colmeans function only accepts numeric values

```

plot(colMeans(trainmatrix))

```

#this function plots the new matrix created from the train csv after all values were converted to numeric



This was the plot that I got and as we can see it is very similar to the digit label

1.

Filtering used for the genes with highest average expression

2747	ML02153a	815	999	900	1025	800	752	801	708
3546	ML03234a	1213	999	910	901	1359	652	880	986
14317	ML26798a	890	999	1134	1032	846	723	785	723
226	ML002127a	633	998	649	618	855	547	525	512
852	ML00571a	693	998	797	562	811	602	438	644

2.

Top 5 genes Filtered on a per column basis are shown below. As we can see it does vary

10617	ML14112a	999	1396	1253	1227	1242	919	994	919
14684	ML28206a	998	971	1032	1115	729	851	718	768
1573	ML01164a	998	851	1004	3024	19	2	15	33
3913	ML03521a	998	1369	1167	1452	1104	1141	1396	1046
6766	ML071318a	998	1261	1028	964	1877	1111	857	913

V1

14317	ML26798a	890	999	1134	1032	846	723	785	723
2747	ML02153a	815	999	900	1025	800	752	801	708
3546	ML03234a	1213	999	910	901	1359	652	880	986
1331	ML00955a	919	998	845	843	953	1009	659	791
4961	ML04822a	802	998	843	397	843	601	292	669

V2

6560	ML06971a	746	913	999	802	702	704	674	695
5982	ML062210a	681	854	999	1105	477	510	513	590
108	ML00109a	902	874	998	803	704	588	478	670
7159	ML07512a	1023	832	998	1032	647	535	348	546
4656	ML04524a	1257	1064	998	1027	1650	839	879	1040

V3

7239	ML076034a	761	795	854	999	870	686	744	785
4374	ML04201a	1111	1256	1219	999	2759	1112	778	1071
14092	ML25766a	978	1217	1140	998	716	882	697	787
13148	ML21627a	987	1174	1125	998	1300	1028	902	1007
6031	ML06272a	670	791	794	997	652	333	347	330

V4

13476	ML22162a	819	668	813	809	999	697	614	605	V5
14546	ML27698a	588	832	862	747	999	700	628	496	
7222	ML076019a	1048	525	668	578	999	838	618	826	
15284	ML329912a	2614	2287	2601	2437	998	1175	817	996	
7578	ML08065a	5402	5736	6021	5298	9970	4353	4109	4986	
2213	ML017310a	12212	12648	11774	10146	11354	9993	11459	11150	V6
15617	ML358826a	1484	2175	2040	1923	904	999	1154	1169	
319	ML002624a	830	883	835	819	1314	999	886	1058	
14851	ML296220a	1064	957	1079	1033	1254	999	782	858	
5632	ML05674a	1318	1217	1351	1151	718	998	893	845	

12777	ML205622a	1440	1368	1271	984	1411	911	999	1221	V7
7462	ML078935a	1047	958	1012	895	938	846	999	1116	
10154	ML13044a	2003	2635	2950	2410	2235	1324	999	1433	
9537	ML11613a	1290	1358	1374	1369	1558	1124	999	1056	
15458	ML34751a	1407	1676	1537	1285	624	1065	999	930	
14401	ML271531a	1413	1046	1108	1003	1063	1049	725	999	V8
9650	ML11975a	353	709	540	590	1774	778	1024	999	
4971	ML04841a	1613	641	830	492	1456	1269	474	998	
15284	ML329912a	2614	2287	2601	2437	998	1175	817	996	
15349	ML33825a	1107	3398	2097	2898	502	930	709	996	

3.

v1 Length:16549 Class :character Mode :character	v2 Length:16549 Class :character Mode :character	v3 Length:16549 Class :character Mode :character	v4 Length:16549 Class :character Mode :character	v5 Length:16549 Class :character Mode :character	v6 Length:16549 Class :character Mode :character
v7 Length:16549 Class :character Mode :character	v8 Length:16549 Class :character Mode :character	v9 Length:16549 Class :character Mode :character			

STANDARD DEV. of 8 columns

```
1 0.1739635
2 1.0524619
3 -0.1749438
4 0.7143595
5 0.8107692
6 1.6739663
7 1.1187899
8 0.3872657
```

Code for standard deviation

```
apply(Mnemiopsis_count_data, sd)
```