DataSet 1

Number 1

* Examples of plotted pictures

Output

A picture containing text

Description automatically generatedText

Description automatically generated with medium confidence

Number 2a)

# Make a Function

myfunction <- function(r){

pixel\_label\_0 <- r[,-1]

c <- c(1:length(pixel\_label\_0))

ave\_pix <- c()

for (x in c) {

ave\_pix <- append(ave\_pix, mean(pixel\_label\_0[,x]))

}

matrix\_ave\_pix <- matrix(ave\_pix, nrow=28, ncol=28)

image(matrix\_ave\_pix, axes=FALSE, col=grey.colors(256,start=0,end=1))

}

myfunction(label\_0)

myfunction(label\_1)

myfunction(label\_2)

myfunction(label\_3)

myfunction(label\_4)

myfunction(label\_5)

myfunction(label\_6)

myfunction(label\_7)

myfunction(label\_8)

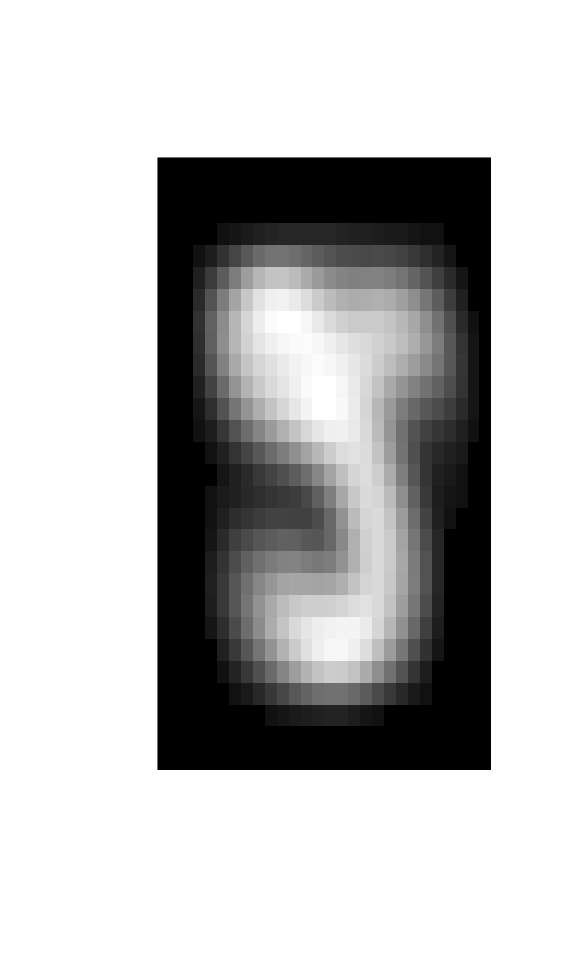
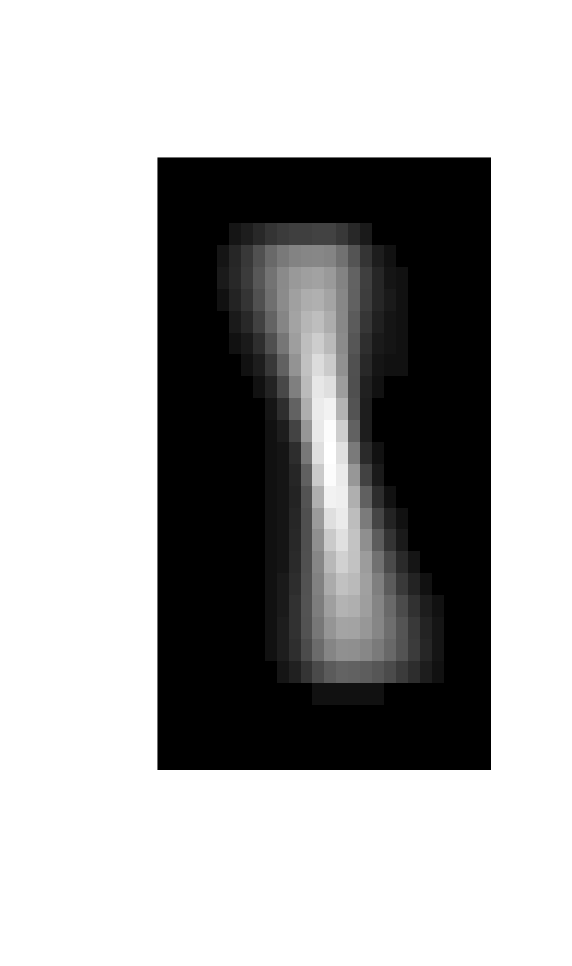
myfunction(label\_9)

# 0, 3, and 8 look the best because they are both distinguishable and in the right orientation

# 1 and 9 look the most blurry

Output

Icon

Description automatically generated with low confidenceA black and white photo of a person's face

Description automatically generated with medium confidenceA black and white photo of a person's face

Description automatically generated with low confidenceA black and white photo of a person's face

Description automatically generated with low confidenceA picture containing icon

Description automatically generatedA black and white photo of a person's face

Description automatically generated with low confidenceA black and white photo of a person's face

Description automatically generated with low confidenceA black and white photo of a person's face

Description automatically generated with low confidence

Number 2b)

# 0, 3, and 8 look the best because they are both distinguishable and in the right orientation

# 1, 2, and 9 look the most blurry

Number 3a)

> func\_each\_label(label\_0, "label\_0")

[1] 12843.17

the column number for label\_0 that has the highest variance is: 268 and the variance is 12843.17

> func\_each\_label(label\_1, "label\_1")

[1] 12997.2

the column number for label\_1 that has the highest variance is: 573 and the variance is 12997.2

> func\_each\_label(label\_2, "label\_2")

[1] 13051.45

the column number for label\_2 that has the highest variance is: 467 and the variance is 13051.45

> func\_each\_label(label\_3, "label\_3")

[1] 12539.43

the column number for label\_3 that has the highest variance is: 188 and the variance is 12539.43

> func\_each\_label(label\_4, "label\_4")

[1] 12629.21

the column number for label\_4 that has the highest variance is: 429 and the variance is 12629.21

> func\_each\_label(label\_5, "label\_5")

[1] 12562.46

the column number for label\_5 that has the highest variance is: 184 and the variance is 12562.46

> func\_each\_label(label\_6, "label\_6")

[1] 12495.36

the column number for label\_6 that has the highest variance is: 240 and the variance is 12495.36

> func\_each\_label(label\_7, "label\_7")

[1] 12615.84

the column number for label\_7 that has the highest variance is: 602 and the variance is 12615.84

> func\_each\_label(label\_8, "label\_8")

[1] 12270.6

the column number for label\_8 that has the highest variance is: 627 and the variance is 12270.6

> func\_each\_label(label\_9, "label\_9")

[1] 12445.65

the column number for label\_9 that has the highest variance is: 576 and the variance is 12445.65

3b) Column 467 in label\_2 and Column 573 in label\_1 had the highest variance, which could explain why their output was blurry (related to 2b)

3c) It doesn’t make a difference because the pixels with the lowest variance had 0 for everything - Tarabeth

3d) - Tarabeth

4) Tarabeth

Dataset 2

# Problem 1

# put file into variable

Mcountdata <- read.csv("/Users/sarahferaidoon/Desktop/Mnemiopsis\_count\_data.csv")

Mcountdata

Output:  
Gene aboral1 aboral2 aboral3 aboral4 oral1 oral2 oral3 oral4

1 ML000110a 69 175 141 139 108 146 133 63

2 ML000111a 0 0 0 0 0 1 0 0

3 ML000112a 1 10 8 3 2 13 6 1

4 ML000113a 383 546 402 471 290 190 282 317

5 ML000114a 188 214 257 230 289 215 162 128

6 ML000115a 493 455 540 501 413 403 419 452

7 ML000116a 404 462 464 362 516 336 285 336

8 ML000117a 266 361 301 273 396 277 239 277

9 ML000118a 177 158 162 153 164 131 107 136

10 ML000119a 382 339 362 295 254 310 259 308

11 ML00011a 37 26 33 29 24 46 34 26

12 ML000120a 227 225 250 141 333 241 130 169

13 ML000121a 385 294 398 213 385 351 188 270

14 ML000122a 352 336 336 283 442 300 245 276

15 ML000123a 1353 1232 1534 1162 1919 1272 976 1130

16 ML000124a 882 1694 1025 1001 979 834 655 849

17 ML000125a 5 0 1 0 30 16 9 12

18 ML000126a 295 306 244 231 237 208 214 253

19 ML000127a 490 810 854 923 451 619 708 412

20 ML000128a 0 0 0 0 0 0 0 0

21 ML000129a 397 207 133 124 373 240 284 367

22 ML00012a 240 598 418 37 264 468 625 259

23 ML000130a 32 56 22 42 49 40 34 67

24 ML000131a 26 62 66 58 69 23 56 50

25 ML000132a 308 850 686 840 2946 4839 4990 5979

26 ML00013a 89 154 127 63 80 45 63 103

27 ML00014a 0 0 0 0 0 0 0 0

28 ML00015a 1 0 2 0 1 1 4 3

29 ML00016a 469 619 573 505 1542 775 1094 917

30 ML00017a 52713 57824 59132 60254 59242 47001 48346 47841

31 ML00018a 32 8 21 1 65 22 10 46

32 ML00019a 122 401 312 266 297 271 273 107

33 ML00021a 148 163 182 157 280 156 105 118

34 ML00022a 186 154 218 182 188 169 147 149

35 ML00023a 357 492 472 515 349 428 435 503

36 ML000310a 626 645 804 675 674 412 337 475

37 ML000311a 951 891 892 863 769 780 524 606

38 ML000312a 242 223 262 233 238 273 182 236

39 ML000313a 1962 2071 2205 1672 3630 2049 1382 1730

40 ML000314a 4875 4087 4765 4996 3122 2269 2096 2356

41 ML000315a 22 26 14 10 14 7 7 7

42 ML000316a 1058 952 943 1022 802 743 572 615

43 ML000317a 0 0 0 0 0 0 0 0

44 ML00031a 288 337 522 437 161 183 328 212

45 ML00032a 0 0 0 0 0 0 0 0

46 ML00033a 297 441 539 468 378 227 269 305

47 ML00034a 429 483 419 281 320 403 226 387

48 ML00035a 128 164 136 64 113 101 43 104

49 ML00036a 204 200 200 116 188 114 110 170

50 ML00037a 0 0 0 0 0 0 0 0

51 ML00038a 227 187 361 242 497 254 166 185

52 ML00039a 0 0 0 0 0 0 0 0

53 ML00041a 125 174 172 185 100 101 129 122

54 ML00042a 0 0 2 1 0 2 1 0

55 ML00051a 32 153 74 61 31 16 13 17

56 ML00052a 351 1215 758 536 360 182 204 221

57 ML000610a 3 2 2 3 1 6 2 8

58 ML00061a 878 839 889 829 869 768 689 789

59 ML00062a 1857 1787 1972 1737 2901 1680 1332 1599

60 ML00063a 2034 2257 2348 2792 1170 1023 823 968

61 ML00064a 1 3 1 2 8 5 3 7

62 ML00065a 129 107 119 110 38 62 51 72

63 ML00066a 1949 2276 2608 2327 1232 1740 1793 2061

64 ML00067a 84 115 96 94 55 69 58 82

65 ML00068a 0 0 0 0 0 0 0 0

66 ML00069a 0 0 0 0 0 0 0 0

67 ML000710a 162 74 102 71 123 90 58 74

68 ML000711a 71 87 85 60 142 57 67 76

69 ML000712a 511 471 537 449 585 416 377 382

70 ML000713a 401 644 582 604 552 319 380 347

71 ML000714a 616 524 585 442 656 353 452 562

72 ML000715a 51 19 28 96 274 785 662 1134

73 ML000716a 572 533 763 519 600 519 297 427

74 ML000717a 864 737 750 629 626 677 543 635

75 ML000718a 0 0 0 0 0 0 0 0

76 ML000719a 3462 2546 3537 2662 3827 2944 2731 2842

77 ML00071a 15 27 37 22 16 29 11 21

78 ML000720a 2097 2938 2239 2129 1703 1255 1391 1409

79 ML00072a 3212 3547 3212 2977 3596 2733 2422 2213

80 ML00073a 439 605 670 582 142 275 345 318

81 ML00074a 1075 1330 1273 1104 1008 858 784 849

82 ML00075a 401 494 498 420 220 333 252 303

83 ML00076a 34 0 2 0 3 11 1 13

84 ML00077a 39 53 68 37 47 28 23 29

85 ML00078a 0 0 0 0 0 0 0 0

86 ML00079a 517 229 327 279 383 373 201 280

87 ML00081a 279 345 276 149 208 296 190 228

88 ML00082a 88 62 142 181 163 91 159 63

89 ML00083a 1 1 2 3 34 33 32 44

90 ML00084a 180 270 263 255 268 213 254 205

91 ML00085a 1 0 0 2 35 52 106 29

92 ML00086a 774 792 907 737 796 715 662 873

93 ML00087a 0 0 0 0 0 0 0 0

94 ML00088a 0 0 0 0 0 0 0 0

95 ML00089a 217 236 116 139 281 170 112 210

96 ML00091a 16 5 6 20 9 18 8 12

97 ML001010a 678 622 659 529 858 568 494 564

98 ML001011a 19 18 39 1 2 11 0 9

99 ML00101a 157 226 174 164 235 149 160 127

100 ML00102a 672 7 144 0 536 322 24 220

101 ML00103a 0 0 0 0 0 0 0 0

102 ML00104a 330 229 269 264 250 244 134 197

103 ML00105a 549 513 458 495 536 408 393 444

104 ML00106a 2 11 0 9 8 11 5 4

105 ML00107a 487 506 546 523 513 510 328 336

106 ML00108a 1523 1493 1844 1313 1991 1208 1092 1275

107 ML00109a 902 874 998 803 704 588 478 670

108 ML001110a 17151 17236 17159 18129 14732 12573 11963 11931

109 ML001111a 0 0 0 0 0 0 0 0

110 ML001112a 2044 1553 1861 1640 1614 1153 747 1145

111 ML001113a 1 0 1 0 0 0 0 0

[ reached 'max' / getOption("max.print") -- omitted 16437 rows ]

Mcoldata <- read.csv("/Users/sarahferaidoon/Desktop/Mnemiopsis\_col\_data.csv")

Mcoldata

Output:

Sample type condition

1 aboral-1 Mleidyi aboral

2 aboral-2 Mleidyi aboral

3 aboral-3 Mleidyi aboral

4 aboral-4 Mleidyi aboral

5 oral-1 Mleidyi oral

6 oral-2 Mleidyi oral

7 oral-3 Mleidyi oral

8 oral-4 Mleidyi oral

Mcountdata$avg <- rowMeans(Mcountdata[,2:9])

topgenes <- Mcountdata[order(Mcountdata$avg,decreasing = TRUE), 1]

head(topgenes,5)

# >[1] "ML20395a" "ML26358a" "ML46651a" "ML020045a" "ML00017a"

# ML20395a has molecular function: nucleotide binding, translation elongation factor activity, GTPase activity, GTP binding; Cellular Component: ribosome; Biologica process: nematode larval development, gamete generation, positive regulation of growth rate, locomotion, embryo development ending in birth or egg hatching, regulation of translational elongation

# ML26358a has molecular function : nucleotide binding, ATPbinding; Cellular Componend: cytoskeleton, cytoplasm; Biological Process: cytoskeleton organization, actin cytoskeleton organization, cytokinesis, embryo development ending in birth or egg hatching

# ML46651a has cellular component: membrane attack complex

# ML020045a molecular function: nucleotide binding, GTPase activity, GTP binding; Cellular Component: microtubule, cytoskeleton, cytoplasm, protein complex; Biological process: microtubule-based process, microtubule-based movement, protein polymerization

# ML00017a

Problem 2

# Yes, the top 5 genes will be different if done on a per column basis

# aboral1

head(Mcountdata[order(Mcountdata$aboral1,decreasing=TRUE),1],5)

# [1] "ML46651a" "ML20395a" "ML020045a" "ML174731a" "ML26358a"

# ML174731a is new, missing ML00017a

# aboral2

head(Mcountdata[order(Mcountdata$aboral2,decreasing=TRUE),1],5)

# [1] "ML20395a" "ML46651a" "ML26358a" "ML01482a" "ML034334a"

# "ML01482a" & "ML034334a" are new, missing ML00017a and ML020045a

# aboral3

head(Mcountdata[order(Mcountdata$aboral3,decreasing=TRUE),1],5)

# [1] "ML20395a" "ML01482a" "ML26358a" "ML46651a" "ML034334a"

# "ML01482a" & "ML034334a" are new

# aboral4

head(Mcountdata[order(Mcountdata$aboral4,decreasing=TRUE),1],5)

# [1] "ML01482a" "ML20395a" "ML034334a" "ML46651a" "ML034336a"

# "ML01482a" & "ML034334a" & "ML034336a" are new

# oral1

head(Mcountdata[order(Mcountdata$oral1,decreasing=TRUE),1],5)

# [1] "ML20395a" "ML020045a" "ML04011a" "ML26358a" "ML00017a"

# ML04011a is new

# oral2

head(Mcountdata[order(Mcountdata$oral2,decreasing=TRUE),1],5)

# [1] "ML20395a" "ML020045a" "ML04011a" "ML00017a" "ML26358a"

# ML04011a is new

# oral3

head(Mcountdata[order(Mcountdata$oral3,decreasing=TRUE),1],5)

# [1] "ML20395a" "ML004510a" "ML26358a" "ML00017a" "ML04011a"

# "ML004510a" is new

# oral4

head(Mcountdata[order(Mcountdata$oral4,decreasing=TRUE),1],5)

# [1] "ML20395a" "ML004510a" "ML46651a" "ML020045a" "ML00017a"

# "ML004510a" is new

Problem 3

# from columns 2-9 in Mcountdata, put the colmeans into a list

MCDmean <- c(colMeans(Mcountdata[,2:9]))

MCDmean

Output:

aboral1 aboral2 aboral3 aboral4 oral1 oral2 oral3 oral4

524.0979 580.5219 581.2736 560.0897 551.6403 428.9934 419.6067 457.4317

# apply sd method to columns 2 - 9 of Mcountdata

MCDsd <- sapply(Mcountdata[,2:9],sd)

MCDsd

Output:

aboral1 aboral2 aboral3 aboral4 oral1 oral2 oral3 oral4

2281.937 2665.179 2451.040 2687.429 2362.584 1631.392 1726.889 1912.523

# put all columns in Mcountdata in variable

scale <- Mcountdata[,1:9]

for (i in 2:9){# for every item in the column

c <- (MCDmean[1])/(MCDmean[i-1]) # create scaling factor of meancol/meancolumn-1

scale[,i]<- c \* scale[i] # multiply column by this scaling factor

}

scale

colMeans(scale[,2:9])

Output

aboral1 aboral2 aboral3 aboral4 oral1 oral2 oral3 oral4

524.0979 524.0979 524.0979 524.0979 524.0979 524.0979 524.0979 524.0979

# Problem 4

cor(scale[,2:9])

# aboral-aboral and oral-oral correlations are closer than aboral-oral

Output

aboral1 aboral2 aboral3 aboral4 oral1 oral2 oral3 oral4

aboral1 1.0000000 0.8471946 0.8873340 0.7951286 0.8386773 0.8527215 0.7762130 0.8500432

aboral2 0.8471946 1.0000000 0.9720700 0.9747975 0.7403459 0.7430881 0.8011097 0.7501215

aboral3 0.8873340 0.9720700 1.0000000 0.9491527 0.8257897 0.8260390 0.8427193 0.8014047

aboral4 0.7951286 0.9747975 0.9491527 1.0000000 0.6726462 0.6811715 0.7641900 0.6955056

oral1 0.8386773 0.7403459 0.8257897 0.6726462 1.0000000 0.9586231 0.8905611 0.9020024

oral2 0.8527215 0.7430881 0.8260390 0.6811715 0.9586231 1.0000000 0.9308689 0.9420304

oral3 0.7762130 0.8011097 0.8427193 0.7641900 0.8905611 0.9308689 1.0000000 0.9491639

oral4 0.8500432 0.7501215 0.8014047 0.6955056 0.9020024 0.9420304 0.9491639 1.0000000

# Problem 5

# I would break the data into thirds somehow