R Exercises

Gabriel Crone

2024-04-10

This document compiles my solutions from the R exercises from a book by Grinberg & Reed's "Programming Exercises for R."

Part 1: Vectors

1.

```
# a)
1:20
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
```

```
# b)
20:1
```

```
##  [1] 20 19 18 17 16 15 14 13 12 11 10  9  8  7  6  5  4  3  2  1
```

```
# c)
c(1:20, 19:1) # OR c(1:19, 20:1)
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 19 18 17 16 15
## [26] 14 13 12 11 10  9  8  7  6  5  4  3  2  1
```

```
# d)
tmp <- c(4, 6, 3)

# e)
rep(tmp, 10)
```

```
##  [1] 4 6 3 4 6 3 4 6 3 4 6 3 4 6 3 4 6 3 4 6 3 4 6 3 4 6 3 4 6 3
```

```
# f)
c(rep(tmp, 10), 4)
```

```
##  [1] 4 6 3 4 6 3 4 6 3 4 6 3 4 6 3 4 6 3 4 6 3 4 6 3 4 6 3 4 6 3 4
```

```
# g)
c(
  rep(4,10),
  rep(6, 20),
  rep(3, 30)
)
```

```
##  [1] 4 4 4 4 4 4 4 4 4 4 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 3 3 3 3 3 3 3 3
## [39] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
```

2.

```
seq <- seq(3, 6, 0.1)
exp(seq) * cos(seq)
```

```
##  [1] -19.884531 -22.178753 -24.490697 -26.773182 -28.969238 -31.011186
##  [7] -32.819775 -34.303360 -35.357194 -35.862834 -35.687732 -34.685042
## [13] -32.693695 -29.538816 -25.032529 -18.975233 -11.157417  -1.362099
## [19]  10.632038  25.046705  42.099201  61.996630  84.929067 111.061586
## [25] 140.525075 173.405776 209.733494 249.468441 292.486707 338.564378
## [31] 387.360340
```

3.

```
# a)
exp1 <- seq(3,36,3)
exp2 <- seq(1, 34, 3)
(0.1^exp1)*(0.2^exp2)
```

```
##  [1] 2.000000e-04 1.600000e-09 1.280000e-14 1.024000e-19 8.192000e-25
##  [6] 6.553600e-30 5.242880e-35 4.194304e-40 3.355443e-45 2.684355e-50
## [11] 2.147484e-55 1.717987e-60
```

```
# b)
num <- 2^(1:25)
denom <- 1:25
num/denom
```

```
##  [1] 2.000000e+00 2.000000e+00 2.666667e+00 4.000000e+00 6.400000e+00
##  [6] 1.066667e+01 1.828571e+01 3.200000e+01 5.688889e+01 1.024000e+02
## [11] 1.861818e+02 3.413333e+02 6.301538e+02 1.170286e+03 2.184533e+03
## [16] 4.096000e+03 7.710118e+03 1.456356e+04 2.759411e+04 5.242880e+04
## [21] 9.986438e+04 1.906502e+05 3.647221e+05 6.990507e+05 1.342177e+06
```

4.

Note that `i` as a name for index cannot be used, since it's already an identified object in R.

```r
# a)
index <- 10:100
sum(index^3 + 4*index^2)
```

```
## [1] 26852735
```

```r
# OR as for loop
vec <- c()
for(i in 10:100) {
  vec[i - 9] <- i^3 + 4*i^2
}
sum(vec)
```

```
## [1] 26852735
```

```r
# b)
index <- 1:25
sum(
  ((2^index)/index) + (3^index)/(index^2)
)
```

```
## [1] 2129170437
```

```r
# OR with for loop
vec <- c()
for(i in 1:25) {
  vec[i] <- ((2^i)/i) + ((3^i)/(i^2))
}
sum(vec)
```

```
## [1] 2129170437
```

5.

```r
# a)
paste("label", 1:30)
```

```
##  [1] "label 1"  "label 2"  "label 3"  "label 4"  "label 5"  "label 6" 
##  [7] "label 7"  "label 8"  "label 9"  "label 10" "label 11" "label 12"
## [13] "label 13" "label 14" "label 15" "label 16" "label 17" "label 18"
## [19] "label 19" "label 20" "label 21" "label 22" "label 23" "label 24"
## [25] "label 25" "label 26" "label 27" "label 28" "label 29" "label 30"
```

```r
# b)
paste0("fn", 1:30)
```

```
##  [1] "fn1"  "fn2"  "fn3"  "fn4"  "fn5"  "fn6"  "fn7"  "fn8"  "fn9"  "fn10"
## [11] "fn11" "fn12" "fn13" "fn14" "fn15" "fn16" "fn17" "fn18" "fn19" "fn20"
## [21] "fn21" "fn22" "fn23" "fn24" "fn25" "fn26" "fn27" "fn28" "fn29" "fn30"
```

6.

```r
# Setup
set.seed(50)
xVec <- sample(0:999, 250, replace=T)
yVec <- sample(0:999, 250, replace=T)

# a)
yVec[-1]-xVec[-length(xVec)]
```

```
##   [1] -359  692 -724   40 -626 -719 -809  527  -89 -829  248  144 -749 -352 -220
##  [16] -249  387 -492   85 -106  303  -97 -436  146  282 -206 -385  -96 -567 -757
##  [31]  287  277 -562  292  -89  -93 -847 -822 -203  679  309 -199 -273    4  -47
##  [46]  142  122  414 -602 -304 -674   -8 -662 -168 -349  -63 -221  115    1 -600
##  [61] -382 -487    2  375   19 -113 -634  107   60   47  214 -325  -49 -290  169
##  [76]  290 -624  457 -408  581 -189  204  -80  409  209 -410  461   37 -127  185
##  [91]  382 -446   44  -56 -270 -598 -378 -155  134 -187  109  316 -139  158  305
## [106]  -39 -119  182  441 -403 -107  615  614 -378 -464   31 -385  665  674 -217
## [121] -279 -406  -45 -489 -350 -451  -18  660  504   -6   60 -130 -379 -302 -219
## [136]  -21  438  129 -201 -275  131  694  -96 -176  117 -113  887 -439 -126 -148
## [151]  392 -158  444 -291  232  -12 -274  477 -510  336 -759 -363 -195 -220  160
## [166] -308 -333  302 -183  227  -12  428  665 -301   -8  222  -50 -444 -425 -650
## [181] -424  318  154  238 -727   71  472  908  265  654 -644 -754  657 -382 -313
## [196]  910 -381  394 -596  602  397 -572  378 -274 -271  601 -791 -378 -461   39
## [211]  163 -118 -332 -170  -94  262 -474  566 -273 -366 -400  374   42  100  135
## [226]  609 -527  580 -219  128 -524  620 -206  410 -280  -66  -50  252  279   48
## [241] -595  -59 -623  247  514   62 -102  475  287
```

```r
# b)
sin(yVec[-length(yVec)])/cos(xVec[-1])
```

```
##   [1]  -0.251172387  -1.714506221  -1.020534796   1.903299501   4.119691628
##   [6]   0.474814101   0.989633106   1.157740470  -2.882161424   1.394557843
##  [11]   3.580223541  -0.664006287   0.847013135  -1.095954285   0.469790086
##  [16]  -0.544605141  -1.253304942  -0.353697082  -1.357919064  -1.271051804
##  [21]   0.009344711 -10.576799813  -0.674196809  -2.087577737 -32.238631575
##  [26]  -4.966988605  -0.490160550  -0.043753175  -1.766774293  -2.456887042
##  [31]  -1.034891510  -0.118782199   0.036938122  -0.173612680   0.851649079
##  [36]  -0.353501116  -1.024989992   0.589100687   4.018304788  -1.004929483
##  [41]   0.977012057   0.952499185  -0.553614203   1.115864382   1.254898524
##  [46]   1.155274507  -1.032061497   0.906767318  -0.945760017   2.658291106
##  [51]  -1.254932608  -0.547511538   3.928248333  -3.673653649  -0.127796877
##  [56]  -3.226308105  -0.894400173   0.701414603   2.128023689  -0.414254810
##  [61]  -0.917829147   0.558286542   0.000000000  -0.564183939   0.946409854
##  [66]   1.407199013   0.800206658  -1.146531865   1.181638746   0.495187680
##  [71]   1.446378081  -1.004546310  -0.648760582  -0.909533924   2.364794213
##  [76]  -1.067542299   0.735761986  -1.085930368   1.046185000  -0.235199131
##  [81]   5.984225913   0.811049413   1.134803403   0.652427664  -0.915236455
##  [86]   2.180704313   1.832739489  -0.433287348  -2.250658142  -1.675524524
##  [91]  -1.553949156  -1.249207462  12.092176300   0.808459235   0.832313214
##  [96]   0.903245844   1.066231524   5.294081295  25.183138328  -1.825927150
## [101]   0.246533436   0.874187672   1.092740398  -0.722495528   0.766922081
## [106]   1.339270617  -1.106082115   2.130306844   0.799377104   6.004793950
## [111]   8.758581778  -0.981032578   1.256557471  -2.048399633   1.195677807
## [116]  -1.040371386   0.644501924  -3.483330739 -13.236374600   1.549790230
## [121]  31.620598675   1.501371199  -0.815852894  -0.286937344   2.086096713
## [126]   0.169201819  -0.944412409   1.404263380   1.918061130  -6.647769454
## [131]   1.097542414  22.775427858   1.581242582   0.596376118   1.286025755
## [136]  -0.274884286   1.792271243   0.929571073  -1.398639309   0.180473893
## [141]  -0.861690054  -3.817208262   1.406954246  -0.116266112   0.911847938
## [146]   1.720472285  -0.351853469  -0.140708190 102.843164547  -0.814695119
## [151]  -4.198084329  -0.796216596  -1.483961597  -0.913007027   2.209820017
## [156]  -0.874704929  -7.182835580  -0.171459039   1.010581535  -0.060316505
## [161]  -5.934493823  -0.194382721   2.788122775   7.857328199   9.813569870
## [166]  -1.041969001 -17.395410126   0.604907292  -1.027075437  -0.256821966
## [171]   0.380364801  -0.898464055  -0.755130341  -4.425456175   0.969617473
## [176]   0.292765308  -1.130607542   0.904382713  -0.969592200  -0.069832291
## [181]   0.000000000  -0.629251004   3.078476507   0.433631660   1.711984983
## [186]  -3.189462891   0.221313374  -1.594268883  -0.158597479   0.218596389
## [191]   0.984958329  -2.227888183   0.233630793  -2.913722008   0.773617325
## [196]   1.026041041   9.839208858  -1.672630213  -0.370798565   1.312355946
## [201]  -0.889637266  -1.473979138  -0.618321180  -0.117513292   0.237786925
## [206]   0.876611998   1.328547955  -0.982203606  -0.061533125   1.004160645
## [211]  -1.266243000   0.534185156   0.571802673  -0.622297532   0.580410706
## [216]  -1.482925131   1.095346759   1.701972216  -0.976847974  -1.309841610
## [221]   1.053044730   0.277103276   0.934757395  -1.112172484  -0.326263919
## [226]   0.416934946  -1.171403611  -1.014835503  -0.549469340  -0.741222996
## [231]   1.877938004  -0.074048223  -0.021532438 -11.511078788  -1.389914300
## [236]   2.423540052   0.791040396  -6.445834186  -0.523259399  -1.446820579
## [241]  -0.734282899  -1.420238075  -0.820938658  -0.949002149   1.452579737
## [246]  -2.219943144  -1.114115747   0.747424267  -0.919986986
```

```
# c)
# Shortcut to using base R is using head and
# -number, where number is number of values
# to "snip" off from the end.
head(xVec, -2) + 2*head(xVec, -1)[-1] + -1*xVec[-(1:2)]
```

```
##   [1] -432 1554  145  937 1811 2133  855  883 2241 1428 -681 1225 2238  842  885
##  [16]  391 1031 1982 1288  798  416 1618 1932 1055 2127 1157 1459  956 2323 1263
##  [31] -426 1014  880 -276  863 1761 2374 1379  344  434 1629 1203 2177  682   58
##  [46] 1226  815 1005 1830 2686  320 1871  700 1189 1356 2095  818 -840 1403 1790
##  [61] 1343  564  803  303 1559 2033 1126  711 1608  614  976 2108 1821  473 -281
##  [76] 1586  473 1755  263 1559   -3 1558 1209  -98  846 1131  848  778  350 -508
##  [91] 1253   37 1380 1064  799 1816 2216  816 1445 1179  931 1474  394 1219  556
## [106] 1333 1798  406 1122 1175  636  135  676 1807 1514 1387  531  154  478  408
## [121] 2011 1071 1427 1472 2076 1747  369 -343 1252  997  400  790 1398 2042  776
## [136]  689 1589  547 1510  934 -289  978 1202 -593 1834  228 1515 1275  452  979
## [151] 1210  804 1813  794  435 1688  948  914 1012 1244 1965 1456 1683  798 1463
## [166] 2673  608 1735  204 1100  713  462  352 1109 2028  401 1380 1531 1223 1660
## [181] 1373  632 -148 1321 2084 1650  112  559  147  552 2410  782  703 2550  354
## [196] 1500  368 1147  673 -268 1355 1719  506 2061  279  813 1833 1976 2057 2043
## [211] 1030 1081  963  808 -241 1331  488  470 1993 1412  454 1210 2389  835 -187
## [226] 1162 -177 2083  286 1363  478  423  308 1123   43 1458  964 1250  750 1493
## [241] 2121 2040  760  761  445 1787 1083 -306
```

```
# d)
exp(-xVec[-1]) / (head(xVec, -1) + 10)
```

```
##   [1]  1.204242e-07  0.000000e+00  1.810357e-44  0.000000e+00  0.000000e+00
##   [6]  0.000000e+00 4.554744e-166 6.364504e-310  0.000000e+00 2.216790e-121
##  [11]  1.473287e-10  0.000000e+00  0.000000e+00 1.761903e-123 4.875841e-209
##  [16] 1.228359e-150  0.000000e+00  0.000000e+00 2.632953e-194 8.791668e-186
##  [21] 2.957234e-212  0.000000e+00  0.000000e+00 1.242932e-274  0.000000e+00
##  [26] 3.798215e-209  0.000000e+00 2.659252e-250  0.000000e+00  9.156951e-73
##  [31]  3.635629e-08 1.486750e-266  2.515946e-97  5.302491e-73  0.000000e+00
##  [36]  0.000000e+00  0.000000e+00 6.667706e-161 7.175243e-112 4.173352e-228
##  [41]  0.000000e+00 3.371825e-261  0.000000e+00  5.294413e-41 7.447700e-144
##  [46] 1.055107e-297 3.617615e-201  0.000000e+00  0.000000e+00  0.000000e+00
##  [51]  7.104957e-82  0.000000e+00 3.344205e-116  0.000000e+00 1.577990e-288
##  [56]  0.000000e+00  3.136056e-29  7.598385e-24  0.000000e+00 1.118155e-281
##  [61] 4.849243e-215 4.072293e-121 4.021236e-205 2.979220e-176  0.000000e+00
##  [66]  0.000000e+00 1.578479e-192 4.352046e-220 5.403389e-317 4.434427e-149
##  [71]  0.000000e+00  0.000000e+00 7.702245e-221  2.942717e-44  4.070550e-96
##  [76]  0.000000e+00 1.960926e-107  0.000000e+00  8.309216e-49  0.000000e+00
##  [81]  1.698799e-27  0.000000e+00 7.343299e-105  2.126099e-57 3.504066e-255
##  [86] 5.064610e-195 1.445624e-147 7.116919e-115  1.100181e-32  2.711706e-21
##  [91] 5.926211e-290  4.603199e-53  0.000000e+00 9.427948e-195 6.364717e-294
##  [96]  0.000000e+00  0.000000e+00 7.625728e-171  0.000000e+00 9.824078e-208
## [101] 7.583353e-232 4.006788e-260 3.946542e-105 6.408844e-292 2.183232e-153
## [106]  0.000000e+00 1.487365e-268  1.692497e-99 1.208239e-283 2.365668e-173
## [111] 3.313994e-113 1.695058e-116 4.154916e-281  0.000000e+00 3.549757e-248
## [116] 1.211651e-210  7.148063e-60  2.450240e-92 1.443070e-171 3.278673e-222
## [121]  0.000000e+00 2.073755e-207  0.000000e+00  0.000000e+00  0.000000e+00
## [126] 4.118041e-199  9.235553e-21  2.396726e-72 3.885028e-309 4.034170e-142
## [131] 1.647834e-153 1.243431e-268  0.000000e+00  0.000000e+00 1.293524e-122
## [136] 3.920730e-238 3.840864e-293 9.145122e-128 9.929487e-304  4.640297e-73
## [141]  7.844793e-37 2.029344e-265 3.800454e-138  3.206837e-13  0.000000e+00
## [146]  1.716175e-41  0.000000e+00 6.741383e-161 1.076850e-152 5.532183e-263
## [151] 3.774784e-247 5.664716e-225  0.000000e+00 1.247700e-112 8.923511e-214
## [156]  0.000000e+00 6.989130e-164 1.852419e-253 4.124315e-267  0.000000e+00
## [161]  0.000000e+00 2.024308e-286  0.000000e+00 6.792761e-220  0.000000e+00
## [166]  0.000000e+00  8.756904e-96  0.000000e+00  4.374177e-17 4.075868e-278
## [171]  7.893133e-91 9.210971e-144 2.536816e-171  0.000000e+00  0.000000e+00
## [176] 1.147966e-115  0.000000e+00 4.220391e-285 7.768924e-286 2.074705e-318
## [181] 1.106982e-194 3.894643e-104 3.076703e-121  0.000000e+00  0.000000e+00
## [186] 7.431362e-194  3.099987e-14 1.449462e-165  5.460289e-98 6.858588e-292
## [191]  0.000000e+00 5.329236e-111 1.286986e-310  0.000000e+00  1.374861e-37
## [196]  0.000000e+00  6.521471e-48 1.748565e-262  4.341399e-69 1.615574e-101
## [201]  0.000000e+00 4.571403e-272 6.717264e-173  0.000000e+00  9.322170e-53
## [206]  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
## [211] 5.015282e-176 1.327335e-229 6.273082e-159 2.738194e-122  9.678820e-46
## [216] 1.524753e-311  1.149232e-85 4.899630e-264  0.000000e+00 3.967651e-199
## [221] 5.138059e-181  0.000000e+00  0.000000e+00  5.861523e-38  8.061906e-68
## [226] 9.665031e-249  1.331189e-55  0.000000e+00  5.013803e-05 3.776034e-307
## [231]  1.476945e-23 3.805934e-140 7.443384e-115 2.738260e-230  1.259697e-81
## [236]  0.000000e+00 3.049294e-178 1.161061e-298 6.344190e-226  0.000000e+00
## [241]  0.000000e+00 4.655339e-302 6.395985e-117 3.705172e-198 3.083071e-176
## [246]  0.000000e+00  2.853509e-95  1.943904e-63  0.000000e+00
```

7.

Tip 1: `which({condition})` returns the position values of the vector for which the condition specified is TRUE. With those values, need to index them in with `[]`.

Tip 2: The `order()` function returns the indices of an ordered vector, allowing you to use indices in rearranging of other vector (e.g., in part g).

```
# a)
yVec[which(yVec > 600)]
```

```
##  [1] 702 901 617 726 915 723 941 906 782 681 721 929 827 653 839 800 869 692 840
## [20] 845 769 866 696 685 788 642 902 797 601 656 842 970 680 792 662 868 875 795
## [39] 880 700 665 699 979 796 772 836 974 990 954 846 943 658 655 628 623 629 989
## [58] 738 992 758 870 910 933 641 872 904 647 988 753 624 996 621 714 965 920 755
## [77] 783 856 927 759 700 764 666 667 790 654 959 868 963 698 686
```

```
# b)
which(yVec > 600)
```

```
##  [1]   3   9  10  18  20  22  25  26  27  29  37  41  42  43  45  48  49  51  65
## [20]  67  71  74  79  81  84  85  88  95  98  99 103 106 108 109 110 113 114 119
## [39] 120 129 130 131 138 139 143 147 148 152 154 159 161 166 167 168 172 173 174
## [58] 176 177 183 187 188 189 190 191 194 196 197 201 202 204 206 207 211 212 219
## [77] 223 224 225 227 229 230 233 235 238 239 240 243 246 248 249
```

```
# c)
xVec[which(yVec > 600)]
```

```
##  [1] 819 706 903 761 439 481 624 988 473 568 926 518 852 593  86 455 773 935 398
## [20] 755 335 500 810 755 233 125 332 440 811 385 591 345 610 221 646 261 640 206
## [39] 388 161 705 319 667 286 605  87 895 561 777 576 778 963 961 212 201 324 387
## [58] 770 258 232 438  25 376 218 665 708  78 762 227 873 390 113 839 757 397 601
## [77] 814 827  79 566 983   3 317 523 402 680 512 687 398 211 139
```

```
# d)
sqrt(abs(xVec - mean(xVec)))
```

```
##   [1] 12.638513 22.731740 17.095847 20.802211 18.310325 18.822008 16.919456
##   [8] 12.358479 13.389100 19.397629 15.991623 22.577245 21.430539 15.403506
##  [15] 15.834519  7.261680 13.701533 15.305816 17.414592  9.366536 10.331118
##  [22]  6.762544 20.958721 16.378889  9.862454 21.477151  7.330211 15.819861
##  [29]  6.424017 20.742902 19.176340 22.687706  9.070171 17.627592 19.124121
##  [36] 16.948982 19.981692 19.449113 12.834796 16.635264  2.954996 18.035188
##  [43]  8.140516 17.557562 20.993618 14.203239 12.299106  8.469475 15.692928
##  [50] 21.662594 20.205643 18.620741 21.172340 16.362518 16.317720 11.369609
##  [57] 14.908655 21.627113 21.857081 21.685663 10.642744  6.303332 15.991623
##  [64]  7.856971 11.346013 20.766993 15.108541  9.577682  5.266118 13.973833
##  [71] 13.846732 15.851435 19.704517  5.170300 20.802211 17.655934 16.681367
##  [78] 16.962665 16.830567 20.560447 15.108541 21.719392 18.473440 17.138611
##  [85] 20.043253  7.366682  9.259158 13.954641 16.423520 21.418030 21.993908
##  [92] 11.630477 20.315807 18.062890  9.313002 11.927615 21.266594 16.860249
##  [99] 11.905125 15.435932  7.532065  0.855570  8.016733 17.109413 11.758741
## [106] 13.480801 16.500545  9.125130 17.485194 10.920989 11.650408 16.544848
## [113] 16.301288 10.642744 18.392064  6.022292  7.052092 19.918132 17.908992
## [120] 11.778455  4.767809 21.477151  7.598158 18.118168 15.008931 18.980727
## [127]  8.759680 22.062004 19.124121 13.351704 14.412911 13.443660  9.179760
## [134] 15.851435 15.564961 15.897547  3.777301 11.843479 15.515541 12.855660
## [141] 19.150248 21.183295  8.846920 14.721821 22.443975 20.549161 20.969788
## [148] 19.190310 12.834796 13.517840  8.442038  5.853888  4.090477 15.819861
## [155] 16.605180  6.460031 16.101801 12.559140  7.019117  8.959241 15.851435
## [162] 19.805757 11.147556 15.435932  5.360224 20.887029 20.839098 17.740688
## [169] 15.819861 22.265040 10.405191 18.048047 14.238399 11.820829 14.807701
## [176] 15.597051 16.393047 18.927969 11.012175 11.102612 14.080767  9.313002
## [183] 17.167760 15.960326 19.931583 16.500545  9.419766 22.399375 12.277296
## [190] 17.570771 11.758741 21.662594 16.725191 13.463581 20.815091 21.183295
## [197] 15.338448 20.609027  8.442038 19.383808 17.312770 18.608278  9.553429
## [204] 11.693246 19.111986 20.340403 17.671106 21.243069 20.863077 19.981692
## [211] 15.174584 11.389996  2.394160 12.989688 15.897547 20.705845 13.574535
## [218] 18.377486  8.617888 19.856183  8.759680 10.850438 16.948982 17.328243
## [225] 21.159679 19.409585  6.266418 20.167598 21.360431 22.885192 13.276596
## [232] 21.925601 14.482127 16.423520  1.931839 18.620741 17.699379 11.168348
## [239] 12.380145  3.838229 20.670462 20.006699 12.659700 16.301288  8.816575
## [246] 11.346013 16.530820 17.768849 19.690912 16.378889
```

```r
# Now is a good place to use pipes
# (makes it more readable)
(xVec - mean(xVec)) |>
  abs() |>
  sqrt()
```

```
##   [1] 12.638513 22.731740 17.095847 20.802211 18.310325 18.822008 16.919456
##   [8] 12.358479 13.389100 19.397629 15.991623 22.577245 21.430539 15.403506
##  [15] 15.834519  7.261680 13.701533 15.305816 17.414592  9.366536 10.331118
##  [22]  6.762544 20.958721 16.378889  9.862454 21.477151  7.330211 15.819861
##  [29]  6.424017 20.742902 19.176340 22.687706  9.070171 17.627592 19.124121
##  [36] 16.948982 19.981692 19.449113 12.834796 16.635264  2.954996 18.035188
##  [43]  8.140516 17.557562 20.993618 14.203239 12.299106  8.469475 15.692928
##  [50] 21.662594 20.205643 18.620741 21.172340 16.362518 16.317720 11.369609
##  [57] 14.908655 21.627113 21.857081 21.685663 10.642744  6.303332 15.991623
##  [64]  7.856971 11.346013 20.766993 15.108541  9.577682  5.266118 13.973833
##  [71] 13.846732 15.851435 19.704517  5.170300 20.802211 17.655934 16.681367
##  [78] 16.962665 16.830567 20.560447 15.108541 21.719392 18.473440 17.138611
##  [85] 20.043253  7.366682  9.259158 13.954641 16.423520 21.418030 21.993908
##  [92] 11.630477 20.315807 18.062890  9.313002 11.927615 21.266594 16.860249
##  [99] 11.905125 15.435932  7.532065  0.855570  8.016733 17.109413 11.758741
## [106] 13.480801 16.500545  9.125130 17.485194 10.920989 11.650408 16.544848
## [113] 16.301288 10.642744 18.392064  6.022292  7.052092 19.918132 17.908992
## [120] 11.778455  4.767809 21.477151  7.598158 18.118168 15.008931 18.980727
## [127]  8.759680 22.062004 19.124121 13.351704 14.412911 13.443660  9.179760
## [134] 15.851435 15.564961 15.897547  3.777301 11.843479 15.515541 12.855660
## [141] 19.150248 21.183295  8.846920 14.721821 22.443975 20.549161 20.969788
## [148] 19.190310 12.834796 13.517840  8.442038  5.853888  4.090477 15.819861
## [155] 16.605180  6.460031 16.101801 12.559140  7.019117  8.959241 15.851435
## [162] 19.805757 11.147556 15.435932  5.360224 20.887029 20.839098 17.740688
## [169] 15.819861 22.265040 10.405191 18.048047 14.238399 11.820829 14.807701
## [176] 15.597051 16.393047 18.927969 11.012175 11.102612 14.080767  9.313002
## [183] 17.167760 15.960326 19.931583 16.500545  9.419766 22.399375 12.277296
## [190] 17.570771 11.758741 21.662594 16.725191 13.463581 20.815091 21.183295
## [197] 15.338448 20.609027  8.442038 19.383808 17.312770 18.608278  9.553429
## [204] 11.693246 19.111986 20.340403 17.671106 21.243069 20.863077 19.981692
## [211] 15.174584 11.389996  2.394160 12.989688 15.897547 20.705845 13.574535
## [218] 18.377486  8.617888 19.856183  8.759680 10.850438 16.948982 17.328243
## [225] 21.159679 19.409585  6.266418 20.167598 21.360431 22.885192 13.276596
## [232] 21.925601 14.482127 16.423520  1.931839 18.620741 17.699379 11.168348
## [239] 12.380145  3.838229 20.670462 20.006699 12.659700 16.301288  8.816575
## [246] 11.346013 16.530820 17.768849 19.690912 16.378889
```

```
# e)
sum(
  # Returns TRUE if condition met per element
  # and FALSE otherwise.
  yVec <= max(yVec) &
  yVec > max(yVec)-200
)
```

```
## [1] 42
```

```
# f)
sum(xVec %% 2)
```

```
## [1] 133
```

```
# g)
xVec[order(yVec)]
```

```
##   [1] 271 725 957 151 374  10 919 996 325 120 216 978 997 409 474 261 607 979
##  [19] 814 271 905 362 692 746 777 793 130  94 257 840 892 435  68 703 862  23
##  [37] 949 853 250 986 813 669 996 441 504 975  49  46  98 239 274 358 598 799
##  [55] 159 885  94 150 114 611 650 339 988 778 881 344 764 189 247 391 180  43
##  [73] 541 487 635 868 180 865 215 830 465 521 253 609  78 440 618 799 259 835
##  [91] 960 921 420 581 927 711 752 257 346 102 966 272 665 640 563 104 887 510
## [109] 276 958 160 855 662 795  40 450 648 656  12 234 915 362 765 800 678 786
## [127] 769 485 251 598 926 805 161 449 310 924 369 777  17 765  59 795 367 499
## [145] 498 778 274 450 651 722 954  55 470 526 469 749 477  31 962 811 903 113
## [163] 201 873 212 324 218 125  78 593 680 961 385 963 646 705 317 523 610 568
## [181] 755 139 935 810 211 319 161 983 819 839 926 481 761 770 227 601 232 566
## [199]   3 335 605 473 814 233 402 221 206 286 440 455 852  87  86 398 591 755
## [217] 576 827 500 261 687 773 438 665 640 388 706 332 708 988  25 439 397  79
## [235] 518 376 624 778 777 512 398 757 345 895 667 762 387 561 258 390
```

```
# h)
yVec[seq(1, length(yVec), 3)]
```

```
##  [1] 537  95 162 617 161  56 269 723 941  88 200  47 721 159 653  39 869 261  91
## [20] 528 397 273 845 559 453 263 696 566 642 902 253 158  71 519 842 970 792 284
## [39] 262  92 171 424 436 665 216 550 796 291 134 974 196 954 473  66 556 658 514
## [58] 623  86 208   0 386 870 641 242 647 496 624 116  48 965 189 360 328 783 214
## [77] 700 179 667 790 560  64 460 426
```

8.

It's important to recognize that a) the numerators of each are all even numbers (and 1) up to 38, and b) the denominators are all odd numbers up to 39.

```
num <- c(1, seq(2, 38, 2))
denom <- seq(1, 39, 2)
cumprod(num/denom)
```

```
##  [1] 1.0000000 0.6666667 0.5333333 0.4571429 0.4063492 0.3694084 0.3409923
##  [8] 0.3182595 0.2995384 0.2837732 0.2702602 0.2585097 0.2481694 0.2389779
## [15] 0.2307373 0.2232941 0.2165276 0.2103411 0.2046562 0.1994087
```

Part 2: Matrices