

# Base R 1: Vectors

*January 22, 2018*

## 1. Create the vectors:

(a) (1, 2, 3, . . . , 19, 20)

---

```
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, 19, . . . , 2, 1)

```
y<- c(20:1)
y
```

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

(c) (1, 2, 3, . . . , 19, 20, 19, 18, . . . , 2, 1)

```
z <- c(19:1)
w <- c(1:20)
c(w,z) # combing vectors
```

```
## [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 19 18 17
## [24] 16 15 14 13 12 11 10  9  8  7  6  5  4  3  2  1
c
```

```
## function (...) .Primitive("c")
```

(d) assign vector c(4, 6, 3) variable name tmp

Use tmp for parts (e), (f) and (g)

---

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

---

(e) (4, 6, 3, 4, 6, 3, . . . , 4, 6, 3) where there are 10 occurrences of 4.

```
rep(tmp,times = 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) (4, 6, 3, 4, 6, 3, . . . , 4, 6, 3, 4) where there are 11 occurrences of 4, 10 occurrences of 6 and 10 occurrences of 3.

```
rep(tmp,l=31)
```

```
## [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) (4, 4, . . . , 4, 6, 6, . . . , 6, 3, 3, . . . , 3) where there are 10 occurrences of 4, 20 occurrences of 6 and 30 occurrences of 3.

---

```
rep(tmp, times=c(10,20,30))
```

```
## [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 3 3 3 3 3
## [36] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
```

---

## 2. Create a vector of the values of

$e^x \cos(x)$  at  $x = 3, 3.1, 3.2, \dots, 6$ .

```
# reset value of tmp
```

```
tmp <- seq(3,6, by = .01)
```

```
tmp1 <- exp(tmp)*cos(tmp)
```

```
head(tmp1, n = 30)
```

```
## [1] -19.88453 -20.11200 -20.34000 -20.56850 -20.79745 -21.02684 -21.25660
## [8] -21.48672 -21.71714 -21.94784 -22.17875 -22.40986 -22.64110 -22.87244
## [15] -23.10384 -23.33524 -23.56660 -23.79788 -24.02902 -24.25998 -24.49070
## [22] -24.72113 -24.95123 -25.18093 -25.41019 -25.63894 -25.86714 -26.09472
## [29] -26.32163 -26.54780
```

```
# To keep the listing shorter
```

---

## 3. Create the following vectors:

(a)  $(0.1^3 0.1^1, 0.1^6 0.2^4, \dots, 0.1^{36} 0.2^{34})$

```
0.1^seq(3,36,by=3)*(0.2^seq(1,34,by=3))
```

```
## [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)  $(2, \frac{2^2}{2}, \frac{2^3}{3}, \dots, \frac{2^{25}}{25})$

```
(2^(1:25))/(1:25)
```

```
## [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. Calculate the following:

(a)  $\sum_{i=10}^{100} (i^3 + 4i^2)$

```
i <- 10:100
sum(i^3 + 4*i^2)
```

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

(b)  $\sum_{i=1}^{25} \left( \frac{2^i}{i} + \frac{3^i}{i^2} \right)$

---

```
tmp = 1:25
sum((2^tmp)/tmp + 3^tmp/(tmp^2))
```

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

5. Use the function `paste()` to create the following character vectors of length 30:

- (a) (“label 1”, “label 2”, ..., “label 30”). Note that there is a single space between label and the number following.

```
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"
```

- (b) (“fn1”, “fn2”, ..., “fn30”).

In this case, there is no space between fn and the number following. \*\*\*\*\*

```
paste("fn", 1:30, sep="" )
```

```
## [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. Execute the following lines which create two vectors of random integers which are chosen with replacement from the integers 0, 1, . . . , 999. Both vectors have length 250.

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

---

Suppose  $x = (x_1, x_2, \dots, x_n)$  denotes the vector `xVec` and  $y = (y_1, y_2, \dots, y_n)$  denotes the vector `yVec`.

- (a) Create the vector  $(y_2 - x_1, \dots, y_n - x_{n-1})$ .

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

```
## [1] 163 -122 317 -146 417 393 249 -489 741 771 81 402 -549 338
## [15] 583 -403 -67 217 307 -121 -269 36 -706 -563 102 48 397 297
## [29] -45 -152 497 405 339 -400 499 -89 211 -670 87 74 554 149
## [43] -183 612 193 -453 -70 -141 127 -709 -708 -722 -64 388 -184 -212
## [57] 242 430 275 672 -150 275 -96 -255 512 577 264 439 149 -916
## [71] 374 -889 -332 324 -553 394 -87 -75 345 -735 -55 100 -40 15
## [85] 279 409 790 -547 -487 -399 -619 -168 -185 19 645 551 227 -366
## [99] 242 147 247 -499 -614 758 63 -227 247 379 -472 566 -762 152
## [113] 493 360 69 190 544 -176 216 -676 -205 782 -109 189 -233 505
## [127] -219 288 -57 487 256 300 -192 -263 704 674 217 280 17 -68
## [141] 259 612 -127 1 545 -231 -191 -338 333 495 -21 -4 294 -668
## [155] -814 420 793 631 -67 655 143 611 -220 -518 -285 327 523 -13
## [169] -679 -241 39 193 342 588 469 68 895 -658 232 -331 27 441
## [183] -733 -182 -399 79 -469 371 475 265 -407 211 59 -974 -90 218
## [197] 396 -486 -963 -327 425 220 128 235 294 -107 -365 146 -588 449
## [211] -434 221 846 386 -910 161 206 109 712 -334 -434 7 640 -350
## [225] 923 353 -579 225 327 410 568 -195 -83 154 -486 -195 667 -144
## [239] 272 410 546 380 -559 414 674 193 222 -92 553
```

(b) Create the vector  $(\frac{\sin(y_1)}{\cos(x_2)}, \frac{\sin(y_2)}{x_3}, \dots, \frac{y_{n-1}}{\cos(x_n)})$ .

```
tmp1 <- sin(yVec[-length(yVec)]) / cos(xVec[-1])
```

```
head(tmp1, n = 30)
```

```
## [1] 0.8860340 -1.4418482 0.8280726 -1.6159172 -0.8601734 20.2635647
## [7] -0.7993041 1.7241444 -0.0809424 -0.7489563 -2.5986696 -0.3736105
## [13] 31.1147158 0.1235592 -0.3592523 -0.9074361 0.3437444 5.7820592
## [19] -2.5741856 -0.7866133 -0.5985541 0.9893626 0.3304293 -1.7512465
## [25] -0.5943555 1.0537469 0.6549740 -0.1159658 -0.9717654 0.5718027
```

(c) Create the vector  $(x_1 + 2x_2 - x_3, x_2 + 2x_3 - x_4, \dots, x_{x-2} + 2x_{x-1} - x_n)$

```
xVecLen <- length(xVec)
```

```
xVec[-c(xVecLen-1, xVecLen)] + 2*xVec[-c(1, xVecLen)] - xVec[-c(1, 2)]
```

```
## [1] 1382 70 1221 1749 -98 796 1949 623 -134 618 288 1472 517 -45
## [15] 794 1982 1489 344 -206 1207 292 771 2085 810 1032 1547 767 537
## [29] 702 676 737 664 1451 435 1355 168 1150 989 926 348 1757 1299
## [43] 409 -497 501 2150 1157 1081 1323 2030 1887 1744 879 590 493 1330
## [57] 1254 1281 465 767 1691 464 1238 805 -519 1425 710 -611 1517 963
## [71] 1836 2243 -158 1860 606 506 1917 1304 2021 2025 238 226 733 1538
## [85] 581 -659 824 1109 1136 1339 1239 1584 2300 562 567 -375 1372 761
## [99] 1142 714 1801 2220 624 -806 1738 268 398 1941 668 2037 829 345
## [113] 337 -45 635 -285 1225 691 1792 2216 123 538 1130 1124 1172 944
## [127] 271 -62 229 785 -70 1346 1622 381 104 1036 1015 199 589 1399
## [141] 601 506 560 -145 171 1204 1427 1278 1128 615 269 37 1521 2172
## [155] 1602 464 74 1575 599 88 -267 1185 1655 1564 1420 880 229 1651
## [169] 959 1306 2008 1243 267 1110 556 -791 1300 844 1578 2427 708 1554
## [183] 1439 1150 1269 2274 1419 1067 187 2071 781 -148 1767 1851 1019 -196
## [197] 554 2223 1710 -90 788 1209 876 1322 275 1191 323 1570 1234 768
## [211] 1715 903 -768 1546 1452 -47 1125 -330 871 2463 894 133 975 201
## [225] -137 1553 299 865 746 184 267 839 -63 863 2411 133 1739 1145
```

```
## [239] 1015 47 209 1468 846 10 1146 31 1405 1058
```

(d) Calculate  $\sum_{i=1}^{n-1} \frac{e^{-x_{i+1}}}{x_i + 10}$

```
sum(exp(-xVec[-1])/(xVec[-length(xVec)]+10))
```

```
## [1] 0.01269872
```

**7. This question uses the vectors xVec and yVec created in the previous question and the functions sort,**

order, mean, sqrt, sum, and abs.

(a) Pick out the values in yVec which are > 600.

```
yVec[yVec>600]
```

```
## [1] 709 871 621 930 948 783 878 671 860 768 698 974 855 813 776 721 917
## [18] 985 705 884 840 687 957 955 786 938 930 641 615 988 881 881 997 823
## [35] 791 643 779 693 845 815 752 766 635 993 919 686 635 613 660 800 743
## [52] 965 743 615 615 803 948 760 604 800 772 863 902 689 881 941 924 693
## [69] 835 632 872 876 850 961 681 791 947 915 712 665 921 798 866 828 942
## [86] 841 645 681 827 884 890 970 632 717 846 952 609 824 695 675 777 813
## [103] 792 783 611 853 738 668 791
```

(b) What are the index positions in yVec of the values which are > 600?

```
which(yVec>600)
```

```
## [1] 1 2 5 6 8 10 11 13 16 18 27 28 32 33 34 36 42
## [18] 43 45 48 50 55 58 59 60 61 63 66 67 68 72 79 80 86
## [35] 88 94 95 96 97 101 102 105 107 109 111 114 118 119 120 123 125
## [52] 127 131 132 134 136 137 138 139 142 143 150 151 154 157 158 159 161
## [69] 163 164 167 168 172 173 174 175 176 178 180 181 182 183 187 189 190
## [86] 203 204 205 206 211 213 214 219 220 224 226 227 230 232 237 238 239
## [103] 241 243 245 246 247 249 250
```

(c) What are the values in xVec which correspond to the values in yVec which are > 600? (By correspond, we mean at the same index positions.)

```
xVec[yVec >600]
```

```
## [1] 708 437 513 44 646 107 390 640 676 364 577 257 408 437 618 627 836
## [18] 278 55 458 803 358 525 511 266 578 197 38 724 61 995 652 956 19
## [35] 680 760 48 294 69 505 964 24 10 840 878 113 789 444 986 537 515
## [52] 263 359 189 457 274 543 324 176 160 260 407 216 977 148 293 660 137
## [69] 852 743 353 371 768 339 203 478 49 880 996 894 357 900 972 467 324
## [86] 517 446 533 190 501 124 14 5 863 399 256 678 188 258 110 957 285
## [103] 34 631 179 545 123 238 178
```

(d) Create the vector  $(|x_1 - \bar{x}|^{1/2}, |x_2 - \bar{x}|^{1/2}, \dots, |x_n - \bar{x}|^{1/2})$

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

```
## [1] 16.0044994 3.8543482 15.8699716 17.7522956 7.8194629 20.1954450
## [7] 15.7208142 13.9335566 20.2449006 18.5702989 7.8648585 13.5224258
## [13] 13.7165593 19.3611983 13.2233127 14.9714395 19.5740645 9.3731532
## [19] 19.4385185 16.8480266 12.8118695 16.0890025 16.0668603 19.7520632
## [25] 11.9522383 14.0763632 11.1867779 13.9590831 11.3073427 9.1572922
```

```
## [31] 9.6879306 6.6223863 3.8543482 12.8896858 15.1610026 13.2341981
## [37] 18.1894475 15.7842960 8.8800901 2.4787093 9.4263461 19.5995918
## [43] 13.1854465 18.9434949 19.9212449 15.7525871 22.4085698 2.4787093
## [49] 16.1599505 18.7388367 23.3268943 17.6958752 13.6800585 12.3634947
## [55] 9.6879306 5.1822775 16.2217138 8.5524266 7.6905136 13.6329014
## [61] 11.2313846 14.2528594 15.9642100 11.5388041 17.9681941 20.3434510
## [67] 16.4967876 19.7700784 17.7723381 22.1843188 7.4259006 23.3054500
## [73] 14.4618118 19.4385185 22.6967839 17.4314658 14.3228489 22.4531512
## [79] 14.1472259 22.4531512 9.5469367 20.8532012 10.6233705 4.1405314
## [85] 9.5991666 20.8051917 21.2333700 15.1044364 9.2273506 13.8976257
## [91] 15.4642814 15.3669776 19.3944322 17.5540309 20.0961688 12.5640758
## [97] 19.5667064 18.8452647 11.8682770 14.7018366 7.2899931 22.6305988
## [103] 13.4217734 21.0678903 20.6846803 20.2520122 21.0203711 12.7335777
## [109] 19.7013705 9.9426355 20.6432556 19.4898948 16.0890025 18.4080417
## [115] 19.2316406 11.3954377 18.9962101 18.3614814 2.8028557 23.1115556
## [121] 13.1203658 20.8292103 9.2273506 10.1066315 7.9463199 2.8537694
## [127] 13.7424889 20.2449006 19.3870060 13.9948562 9.6361818 16.2128344
## [133] 18.8452647 2.2680388 18.7844617 13.3362663 9.5469367 11.3073427
## [139] 16.6089133 5.0143793 9.4416100 17.0837935 13.8512093 16.6690132
## [145] 20.0961688 6.0709143 15.9732276 13.1584194 8.8399095 6.6974622
## [151] 15.3576040 15.0948998 7.5402918 22.9160206 19.3944322 3.0239048
## [157] 17.4314658 12.6038089 14.4271965 20.3434510 17.7441821 15.0948998
## [163] 20.0035997 17.0629423 15.2034207 9.6511139 9.9426355 8.9919964
## [169] 20.3505282 0.3794733 18.9510950 17.7804387 10.6233705 15.7751704
## [175] 5.1131204 20.0712730 20.7811453 20.6916408 5.3050919 23.3268943
## [181] 21.0272205 9.7394045 21.1694119 12.2940636 14.6677878 18.3069386
## [187] 22.8066657 2.2680388 3.8915293 11.3073427 21.8207241 18.5163711
## [193] 9.3196566 23.1331796 10.9610219 13.1093860 18.4080417 15.8159413
## [199] 22.6084940 6.8451443 19.7194320 13.0055373 8.0711833 2.4199174
## [205] 9.0079964 16.1819653 13.6434600 13.2987217 20.3259440 4.1056059
## [211] 7.0102782 14.7358067 18.1067943 20.9250090 21.6366356 11.9939985
## [217] 19.1795725 8.4346903 21.1389688 20.2766861 20.2025741 18.2169152
## [223] 15.6797959 7.2702132 20.5634627 13.9948562 15.0380850 19.8205953
## [229] 6.7189285 16.2436449 18.0237621 13.9232180 8.7095350 16.7587589
## [235] 18.1423262 20.4485696 18.4893483 22.4754088 12.9172753 8.3579902
## [241] 20.4415264 6.9897067 13.3844686 15.9642100 16.5183534 9.6511139
## [247] 18.1343872 17.5540309 14.6238162 16.5485951
```

(e) How many values in yVec are within 200 of the maximum value of the terms in yVec?

```
sum(yVec>max(yVec)-200)
```

```
## [1] 57
```

(f) How many numbers in xVec are divisible by 2? (Note that the modulo operator is denoted %%.)

```
sum(xVec%%2==0)
```

```
## [1] 124
```

(g) Sort the numbers in the vector xVec in the order of increasing values in yVec.

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

```
## [1] 405 842 308 572 461 8 256 507 373 639 42 616 29 645 376 669 688
## [18] 197 63 638 862 77 996 93 59 585 661 72 339 20 206 537 174 322
## [35] 42 603 425 48 707 452 477 99 224 811 715 358 963 222 395 543 480
## [52] 193 683 710 691 954 700 614 787 835 275 435 309 368 224 460 497 944
```

```
## [69] 530 765 523 171 870 807 469 828 624 200 713 365 781 74 129 76 701
## [86] 760 193 866 353 168 967 545 920 541 650 148 277 18 667 865 987 120
## [103] 655 1 554 699 311 458 632 84 269 82 280 544 17 621 807 113 136
## [120] 457 702 91 625 767 828 109 860 363 121 657 668 324 382 956 299 403
## [137] 74 928 415 38 127 176 678 179 444 724 189 457 513 743 5 10 789
## [154] 38 760 446 986 894 238 640 110 203 533 113 358 977 294 137 258 577
## [171] 55 708 996 863 627 123 515 359 964 324 24 364 260 618 957 48 107
## [188] 631 266 680 478 178 34 900 537 160 274 437 285 505 19 188 190 467
## [205] 852 803 517 69 399 768 545 408 676 407 972 437 353 371 390 995 652
## [222] 148 458 501 124 216 880 836 878 357 660 44 197 578 293 324 49 646
## [239] 543 256 511 525 339 263 14 257 278 61 840 956
```

---

(h) Pick out the elements in yVec at index positions 1, 4, 7, 10, 13, . . .

```
yVec[c(T,F,F)]
```

```
## [1] 709 517 437 783 671 860 581 347 279 974 216 776 538 460 985 248 317
## [18] 288 687 957 938 101 615 285 106 414 881 488 484 791 246 643 845 553
## [35] 465 87 993 116 473 635 310 428 965 19 489 803 604 800 175 516 902
## [52] 689 881 593 835 398 358 850 791 915 665 167 866 942 320 482 216 488
## [69] 681 273 884 970 469 717 127 952 284 695 325 777 792 72 738 791
```

8. By using the function cumprod or otherwise, calculate

$$1 + \frac{2}{3} + \left(\frac{2}{3}\frac{4}{5}\right) + \left(\frac{2}{3}\frac{4}{5}\frac{6}{7} + \dots + \left(\frac{2}{3}\frac{4}{5}\dots\frac{38}{39}\right)\right)$$

```
1+sum(cumprod(seq(2,38,b=2)/seq(3,39,b=2)))
```

```
## [1] 6.976346
```