Week-2

c1p

2016年7月1日

1.创建以下向量

a<-1:100;a

## [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17  
## [18] 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34  
## [35] 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51  
## [52] 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68  
## [69] 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85  
## [86] 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100

b<-seq(1,100,by=2);b

## [1] 1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45  
## [24] 47 49 51 53 55 57 59 61 63 65 67 69 71 73 75 77 79 81 83 85 87 89 91  
## [47] 93 95 97 99

c<-rep(c(1,7,9),10);c

## [1] 1 7 9 1 7 9 1 7 9 1 7 9 1 7 9 1 7 9 1 7 9 1 7 9 1 7 9 1 7 9

d<-c(0,3,8,15,24,35,48,63,80,99);d

## [1] 0 3 8 15 24 35 48 63 80 99

2.创建一个向量，其值为3^x x^2 ，x= 0;0.1;0.2;……；2

x<-seq(0,2,by=0.1)  
3^x\*x^2

## [1] 0.00000000 0.01116123 0.04982924 0.12513503 0.24829529  
## [6] 0.43301270 0.69594554 1.05725795 1.54126380 2.17717906  
## [11] 3.00000000 4.05152712 5.38155766 7.04927309 9.12485197  
## [16] 11.69134295 14.84683811 18.70699266 23.40794394 29.10969036  
## [21] 36.00000000

3.创建下列向量 3.(1)

y<-1:12  
x1<-3\*y  
x2<-3\*y-2  
x<-c()  
x[seq(1,24,by=2)]<-x1  
x[seq(2,24,by=2)]<-x2  
z<-rep(c(0.1,0.2),12)  
z^x

## [1] 1.000000e-03 2.000000e-01 1.000000e-06 1.600000e-03 1.000000e-09  
## [6] 1.280000e-05 1.000000e-12 1.024000e-07 1.000000e-15 8.192000e-10  
## [11] 1.000000e-18 6.553600e-12 1.000000e-21 5.242880e-14 1.000000e-24  
## [16] 4.194304e-16 1.000000e-27 3.355443e-18 1.000000e-30 2.684355e-20  
## [21] 1.000000e-33 2.147484e-22 1.000000e-36 1.717987e-24

3.(2)

a<-1:10  
b<-2:11  
a^b

## [1] 1 8 81 1024 15625  
## [6] 279936 5764801 134217728 3486784401 100000000000

4.创建以下长度为30的向量 4.(a)

a<-c("china")  
a1<-rep(a,30)  
b<-1:30  
c<-paste(a1,b,sep=" ");c

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

4.(b)

a2<-rep(LETTERS,2)  
a2<-a2[1:30]  
c1<-paste0(a2,b);c1

## [1] "A1" "B2" "C3" "D4" "E5" "F6" "G7" "H8" "I9" "J10" "K11"  
## [12] "L12" "M13" "N14" "O15" "P16" "Q17" "R18" "S19" "T20" "U21" "V22"  
## [23] "W23" "X24" "Y25" "Z26" "A27" "B28" "C29" "D30"

1. 计算下面算式的值

x<-seq(2,38,by=2);x

## [1] 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38

y<-seq(3,39,by=2);  
k<-c()  
for (i in 1:length(x)){  
 k[i]<-(x[1:i]/y[1:i])[i]  
}  
k1<-sum(k)  
k1<-k1+1;k1

## [1] 18.52033

6.执行下列代码，创建两个长度为250的随机向量 6.（a）计算向量

set.seed(50)  
xVec <- sample(0:999, 250, replace=T)  
yVec <- sample(0:999, 250, replace=T)  
x<-xVec;y<-yVec  
k<-c()  
for(i in 2:250){  
 k[i-1]<-(y[i]-x[i-1])  
}  
k

## [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

6.（b）计算向量

k1<-c()  
for(i in 2:250){  
 k1[i-1]<-(sin(y[i])/cos(x[i-1]))  
}  
k1

## [1] 1.68820999 -0.78462480 2.00835156 -0.95576785 -0.14609116  
## [6] -0.31330650 -157.97656516 -0.20297516 1.69210297 -1.01465052  
## [11] -0.26185407 -2.51163226 0.16734011 -9.84145338 -0.83334647  
## [16] -0.36910547 1.26039702 0.21227657 -4.44089073 -1.64465733  
## [21] 1.02054584 -1.59824655 -0.75680250 0.56688387 1.20681329  
## [26] -0.56342516 0.21433049 1.07629330 -0.61885221 -0.76706229  
## [31] 0.47252524 0.67836769 0.02802181 1.50444917 2.02059637  
## [36] -2.83936674 17.38885290 0.44345873 -1.46976332 -1.12189787  
## [41] -2.32425359 -1.05287745 22.07994449 3.02096619 8.34971548  
## [46] -1.10451961 -1.86319378 0.37883019 0.93952237 -0.77248157  
## [51] 0.86120556 -37.48923296 0.28620921 -0.99177676 -0.94592429  
## [56] 0.92811377 3.26654643 0.04707151 -1.20030250 -1.90631789  
## [61] 0.67774756 4.00873886 -0.74646915 -0.16806580 -0.11705642  
## [66] -0.71535603 7.31300315 1.81235506 -1.26336432 5.57725917  
## [71] -2.72022255 1.14801850 2.53727134 4.84376243 -0.77710680  
## [76] -1.05966087 1.06751110 1.69276461 -7.53075750 1.53561330  
## [81] 0.98779541 2.50650376 -0.54463200 -0.31235500 0.11094190  
## [86] 0.68452613 -1.16571993 5.63561286 0.26840329 -1.45226563  
## [91] 0.25672080 0.99783384 4.52188275 -0.11888817 -1.50181895  
## [96] 0.34210814 0.64099800 -0.99816206 -0.07962107 2.42612845  
## [101] 1.31071342 -0.04963063 0.87550906 3.58418859 -1.93743166  
## [106] 1.08036254 0.68332336 -1.40771312 1.14421038 2.39465287  
## [111] -3.15229208 0.83910106 -4.38649522 0.98661648 0.21286157  
## [116] 6.00293365 -0.39003409 0.42375801 -0.51442214 0.94963967  
## [121] 2.39746704 1.35384981 -0.69234445 2.12025636 -0.69364990  
## [126] -2.10203817 0.02825671 0.33096622 0.18181671 -25.12922748  
## [131] -1.04528143 -1.01195500 0.73847767 6.91626589 -23.81321038  
## [136] 0.88793841 0.29818294 -0.79497091 -0.98091483 0.64469955  
## [141] 1.11961748 0.75760341 -1.19004753 2.28393711 -1.07734185  
## [146] 1.02538901 -0.70953596 0.91218262 -1.34729098 -2.17760756  
## [151] -0.30565352 -0.15131989 -1.25448003 -0.90236402 5.23608318  
## [156] -1.42256633 1.05807732 -0.53936715 0.71470453 1.00660395  
## [161] -1.16053140 1.05444962 0.63540963 69.73777796 -2.84769493  
## [166] 14.76290634 1.16129830 -0.14732945 -2.11396967 -0.53077132  
## [171] 1.09893759 -2.69926148 0.69292142 1.75407837 -1.10587724  
## [176] -2.29446124 -1.75195072 0.92641265 -1.15428989 0.85654217  
## [181] 2.28213297 0.08520122 -7.17847336 0.02704674 -1.48061532  
## [186] 28.41140330 -1.06623376 9.66898576 1.00839318 1.09192654  
## [191] 1.44655460 0.74203433 0.13676678 0.48984413 -0.99897490  
## [196] -1.08310812 0.06224505 -4.78586798 0.00000000 -0.53263455  
## [201] -0.88141159 -1.59059624 4.00394986 0.66734550 -1.43856440  
## [206] 14.60428808 -0.32380969 0.24073502 -1.05521481 -8.48592036  
## [211] 10.19332905 0.81145456 -7.36391064 -6.22302670 0.61547417  
## [216] -0.79106515 -1.21590745 -6.84405978 2.31624828 -1.58343519  
## [221] -1.36000805 1.19461164 -3.53250645 0.95390263 0.13003266  
## [226] 11.35871769 -1.19772122 -1.23341766 0.97036504 1.01173142  
## [231] -2.79883006 0.18089847 -1.34951155 -4.59326858 0.98609372  
## [236] -0.43881665 0.85638976 -1.65855271 1.27330187 1.07228179  
## [241] -1.09605985 -1.05732533 -0.28328572 -1.65008035 1.00084796  
## [246] -4.08661604 0.61262935 0.94975614 -0.86980462

6.（c）计算向量

k2<-c()  
for(i in 3:250){  
 k2[i-2]<-(x[i-2]+2\*(x[i-1])-x[i])  
}  
k2

## [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

6.（d）计算

k3<-c()  
for (i in 1:249){  
 k3[i]<-(exp(-(x[i+1]))/(x[i]+10))  
}  
sum(k3)

## [1] 0.01269872

1. 继续沿用上题中的向量xVec和yVec

7.（a）列出yVec中大于600的值

a<-xVec[xVec>600]

7.（b）列出yVec中大于600的值的位置（下标）

a<-xVec>600  
b<-1:250  
b[a]

## [1] 1 4 7 8 13 16 17 21 23 24 26 34 36 38 42 46 47  
## [18] 49 50 51 52 53 57 62 67 70 72 73 75 77 78 79 80 88  
## [35] 90 91 92 93 94 98 100 102 103 106 108 109 111 118 120 121 133  
## [52] 147 148 154 155 159 163 164 165 169 171 172 178 180 181 183 184 185  
## [69] 186 187 191 194 198 199 202 207 209 212 215 220 221 227 235 236 238  
## [86] 243 248

7.（c）xVec中大于600的值与yVec中大于600的值相乘的和结果是（相同位置上的才相乘）

a<-xVec>600  
b<-yVec>600  
c<-1:250  
d<-intersect(c[a],c[b]);d

## [1] 1 8 13 16 34 36 42 50 67 72 79 80 88 94 102 109 111  
## [18] 118 120 154 159 163 164 172 178 180 181 183 187 220 227 238 243

sum(xVec[d]\*yVec[d])

## [1] 20946549

7.(d.创建向量 ，其中 表示向量xVec的均值

xb<-mean(xVec)  
k<-c()  
for(i in 1:length(xVec)){  
 k[i]<-(abs(xVec[i]-xb))^0.5  
}  
k

## [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

7.（e）yVec中有多少个值不超过200

a<-yVec[yVec<=200]  
length(a)

## [1] 43

7.（f）xVex中有多少个值可以整除2

a<-xVec%%2  
b<-a[a<1]  
length(b)

## [1] 124

7.按yVec降序的顺序给xVec排序

a<-order(yVec,decreasing=T)  
xVec[a]

## [1] 956 840 61 278 257 14 263 339 525 511 256 646 543 49 324 293 578  
## [18] 44 197 660 357 878 836 880 216 124 458 501 995 652 148 390 371 353  
## [35] 437 972 407 676 408 545 768 399 69 517 803 852 467 190 188 19 505  
## [52] 437 285 274 537 160 900 34 680 478 178 266 107 631 48 957 618 260  
## [69] 364 24 324 964 515 359 123 627 863 996 708 55 577 258 294 137 977  
## [86] 358 113 203 533 110 640 238 894 986 446 760 38 10 789 743 5 513  
## [103] 724 189 457 444 179 678 176 127 415 38 928 74 403 299 956 382 324  
## [120] 668 657 121 363 860 109 828 767 625 91 702 457 136 113 807 17 621  
## [137] 544 280 82 269 84 632 458 311 699 655 1 554 120 987 667 865 18  
## [154] 277 148 650 541 920 967 545 168 353 866 193 760 701 129 76 74 781  
## [171] 365 713 200 624 828 469 807 870 171 523 765 530 944 497 460 224 309  
## [188] 368 435 835 275 787 614 700 954 691 710 683 193 480 543 395 222 358  
## [205] 963 715 811 224 99 477 452 707 48 425 603 42 322 537 174 206 339  
## [222] 20 72 661 585 59 93 996 77 862 638 63 688 197 669 376 645 29  
## [239] 616 42 639 373 507 256 8 461 572 308 842 405

7.（h）将yVec第1; 4; 7; 10; 13;……位上的值输出

a<-seq(1,250,by=3)  
yVec[a]

## [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

1. 第n 个三角形数表示为n \* (n + 1) / 2。创建一个包含前20 个三角形数的序列。R 有一个内置常数letters，它包含小写的罗马字母。使用前20 个英文字母来给你刚刚创建的向量命名。选择命名为元音的三角数

TN<-function(n){  
 n1<-1:n;k<-c()  
for(i in 1:n){  
 k[i]<-n1[i]\*(n1[i]+1)/2  
}  
 k  
}  
a<-TN(20)  
names(a)<-letters[1:20]  
b<-a[c("a","e","i","o","u")]  
na.omit(b);b

## a e i o   
## 1 15 45 120   
## attr(,"na.action")  
## <NA>   
## 5   
## attr(,"class")  
## [1] "omit"

## a e i o <NA>   
## 1 15 45 120 NA