VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS

STATISTINIS MODELIAVIMAS
Praktinės užduoties Nr. **1--12** ataskaita

Užduotį atliko: **Simona Gelžinytė**
Duomenų mokslo 2 kursas, 2 grupė

2022.05.05

Turinys

Užduotys

MK užduotis 1-12

1. Sugeneruokite pseudoatsitiktinių skaičių sekas tiesiniu kongruentiniu metodu su maksimaliu periodu, kai modulis $m = 1168$ ir $m = 875$. Daugiklius a parinkite taip, kad galingumai būtų didžiausi. Prieauglio c parinkimui naudokitės gretimų narių koreliacija (teoriniai testai).
2. Gautas sekas patikrinkite su dviem testais. Pirma su intervalų testu. Imkite intervalą [2/3,1). Kitą testą pasirinkite patys.
3. Naudodami sugeneruotą geresniąją pseudoatsitiktinių skaičių seką sumodeliuokite du atsitiktinius dydžius, vieną pasiskirsčiusį pagal normalųjį skirstinį $N(0.5,0.5)$, o kitą parinkite patys.
4. Naudodami sugeneruotą geresniąją pseudoatsitiktinių skaičių seką ir parinkdami tankius (tolygiai pasiskirsčiusio intervale [0,7] atsitiktinio dydžio ir kitą savo nuožiūra) suskaičiuokite integralą:

$$\int_0^7 \frac{x^4 - x}{x^2 + 1}\, dx.$$

MCMC užduotis 1-12(5)

Sugeneruokite Markovo grandinę, kuri nebūtų gimimo - mirties procesas ir kurios vienintelis stacionarus skirstinys būtų:

$$\pi = (1/6, 2/6, 3/6, 0)$$

naudokite geresniu algoritmu gautus pseudoatsitiktinius skaičius.

Sprendimai

1 Užduotis.

Sugeneruokite pseudoatsitiktinių skaičių sekas tiesiniu kongruentiniu metodu su maksimaliu periodu, kai modulis m = 1168 ir m = 875 . Daugiklius a parinkite taip, kad galingumai būtų didžiausi. Prieauglio c parinkimui naudokitės gretimų narių koreliacija (teoriniai testai).

Tiesiniu kongruentiniu metodu sugeneruota pseudoatsitiktinių skaičių seka apibrėžiama taip:

$$X_{n+1} = (a\,X_n + c\,)\;mod\;m, \qquad n \geq 0$$

Čia skaičiai :

X_0 – pradinė reikšmė, $X_0 \geq 0,$

a – daugiklis, $a \geq 0,$

c – prieauglis, $c \geq 0,$

m – modulis, $m > X_0,\; m > a,\; m > c$

yra parenkami.

Kadangi modulius jau turime parinktus (m = 1168 ir m = 875), tai turėsime kiekvienai sekai dar parinkti pradines reikšmes $X_0 \geq 0$, daugiklius $a \geq 0$ ir prieauglius $c \geq 0$. Šie dydžiai turi būti mažesni už m. Įsiveskime kintamuosius į R'ą:

```
> m1 <- 1168
> m2 <- 875
```

Kai periodo ilgis yra maksimalus, t. y. lygus m, tai kiekvienas skaičius nuo 0 iki m − 1 sutinkamas vieną kartą. Po to periodas pasikartoja. Taigi, pradinę reikšmę galime pasirinkti laisvai. Tegul $X_0 = 1$. R'e įsivedame dvi sekas, kurių pirmoji reikšmė yra lygi nuliui:

```
> x1 <- c()
> x2 <- c()
> x1[1] <- 0
> x2[1] <- 0
```

Daugiklio ir prieauglio parinkimui naudosimės:

Teorema 1 teigia, kad tiesinis kongruentinis yra maksimalaus periodo *m*, kai

- $(c, m) = 1;$
- $p|m \;=>\; p|(a-1);$
- $4|m \;=>\; 4|(a-1).$

Tiesinės kongruentinės sekos su maksimaliu periodo galingumu vadinsime mažiausią natūralųjį skaičių s, kuriam

$$b^s \equiv 0\;mod\;m, \qquad kur\; b = a - 1,$$

1 teiginys. Prieauglį reikia parinkti tokį, kad koreliacija tarp gretimų sekos narių būtų kuo mažesnė. Taigi, c turės tenkinti lygybę:

$$\frac{c}{m} \approx \frac{1}{2} \pm \frac{1}{6}\sqrt{3}$$

Imkime m = 1168. Išskaidę pirminiais dauginamaisiais gauname, kad $m = 2^4 \cdot 73$. Iš pirmos teoremos antro ir trečio punktų išplaukia, kad

$$b = a - 1 = 4 \cdot 73 \cdot j,$$

čia j = 0,1,2,3. R'u gauname, kad:

```
> m <- m1
> b <- seq(0, 1168 - 73*4, 73*4)
> b <- seq(0, 1168 - 73*4, 73*4)
> b
[1]   0 292 584 876
> (b^2) %% m
[1] 0 0 0 0
```

Taigi, matome, kad galimas galingumus s = 2. Kadangi mums reikia didžiausio galingumo ir didelio daugiklio, tai renkamės b = 876. Vadinasi pirmajai sekai a = b + 1 = 877. Taip ir pasižymime R'e:

```
> b <- b[length(b)]
> a <- b + 1
```

Prieš pradedant ieškoti prieauglio, paskaičiuokime reikalingas konstantas, kurios gaunamos iš 1 teiginio dešinės lygybės pusės:

```
> c1 <- 1/2 - sqrt(3)/6
> c2 <- 1/2 + sqrt(3)/6
```

Tuomet pirmajai sekai c ≈ m · c1 arba c ≈ m · c2:

```
> c(m*c1, m*c2)
[1] 246.8274 921.1726
```

Tikrinsime skaičius 246,247,921, ir 922. Iš 1 teoremos žinome, kad c ir m turi būti tarpusavy pirminiai, todėl DBD(c,m) = 1. Patikrinkime:

```
> library(schoolmath)
> gcd(246, 1168)
[1] 2
> gcd(247, 1168)
[1] 1
> gcd(921, 1168)
[1] 1
> gcd(922, 1168)
[1] 2
```

Kadangi mums tinka 247 ir 921, tai pasirinksime c = 247. Taip pat įsivedame ir R'e:

```
 c <- 247
```

Gavome rinkinį (x_0,a,c,m) = (0,877,247,1168). Dabar jau galime generuoti pirmąją seką:

```
> for (i in 1:(m-1)) x1[i+1] <- (a * x1[i] + c) %% m
> x1

[1]    0  247  786  449  404  651   22  853  808 1055  426   89   44  291  830
  [16]  493  448  695   66  897  852 1099  470  133   88  335  874  537  492  739
  [31]  110  941  896 1143  514  177  132  379  918  581  536  783  154  985  940
  [46]   19  558  221  176  423  962  625  580  827  198 1029  984   63  602  265
  [61]  220  467 1006  669  624  871  242 1073 1028  107  646  309  264  511 1050
  [76]  713  668  915  286 1117 1072  151  690  353  308  555 1094  757  712  959
  [91]  330 1161 1116  195  734  397  352  599 1138  801  756 1003  374   37 1160
 [106]  239  778  441  396  643   14  845  800 1047  418   81   36  283  822  485
 [121]  440  687   58  889  844 1091  462  125   80  327  866  529  484  731  102
 [136]  933  888 1135  506  169  124  371  910  573  528  775  146  977  932   11
 [151]  550  213  168  415  954  617  572  819  190 1021  976   55  594  257  212
 [166]  459  998  661  616  863  234 1065 1020   99  638  301  256  503 1042  705
 [181]  660  907  278 1109 1064  143  682  345  300  547 1086  749  704  951  322
 [196] 1153 1108  187  726  389  344  591 1130  793  748  995  366   29 1152  231
 [211]  770  433  388  635    6  837  792 1039  410   73   28  275  814  477  432
 [226]  679   50  881  836 1083  454  117   72  319  858  521  476  723   94  925
 [241]  880 1127  498  161  116  363  902  565  520  767  138  969  924    3  542
 [256]  205  160  407  946  609  564  811  182 1013  968   47  586  249  204  451
 [271]  990  653  608  855  226 1057 1012   91  630  293  248  495 1034  697  652
 [286]  899  270 1101 1056  135  674  337  292  539 1078  741  696  943  314 1145
 [301] 1100  179  718  381  336  583 1122  785  740  987  358   21 1144  223  762
 [316]  425  380  627 1166  829  784 1031  402   65   20  267  806  469  424  671
 [331]   42  873  828 1075  446  109   64  311  850  513  468  715   86  917  872
 [346] 1119  490  153  108  355  894  557  512  759  130  961  916 1163  534  197
 [361]  152  399  938  601  556  803  174 1005  960   39  578  241  196  443  982
 [376]  645  600  847  218 1049 1004   83  622  285  240  487 1026  689  644  891
 [391]  262 1093 1048  127  666  329  284  531 1070  733  688  935  306 1137 1092
 [406]  171  710  373  328  575 1114  777  732  979  350   13 1136  215  754  417
 [421]  372  619 1158  821  776 1023  394   57   12  259  798  461  416  663   34
 [436]  865  820 1067  438  101   56  303  842  505  460  707   78  909  864 1111
 [451]  482  145  100  347  886  549  504  751  122  953  908 1155  526  189  144
 [466]  391  930  593  548  795  166  997  952   31  570  233  188  435  974  637
 [481]  592  839  210 1041  996   75  614  277  232  479 1018  681  636  883  254
 [496] 1085 1040  119  658  321  276  523 1062  725  680  927  298 1129 1084  163
 [511]  702  365  320  567 1106  769  724  971  342    5 1128  207  746  409  364
 [526]  611 1150  813  768 1015  386   49    4  251  790  453  408  655   26  857
 [541]  812 1059  430   93   48  295  834  497  452  699   70  901  856 1103  474
 [556]  137   92  339  878  541  496  743  114  945  900 1147  518  181  136  383
 [571]  922  585  540  787  158  989  944   23  562  225  180  427  966  629  584
 [586]  831  202 1033  988   67  606  269  224  471 1010  673  628  875  246 1077
 [601] 1032  111  650  313  268  515 1054  717  672  919  290 1121 1076  155  694
 [616]  357  312  559 1098  761  716  963  334 1165 1120  199  738  401  356  603
 [631] 1142  805  760 1007  378   41 1164  243  782  445  400  647   18  849  804
 [646] 1051  422   85   40  287  826  489  444  691   62  893  848 1095  466  129
 [661]   84  331  870  533  488  735  106  937  892 1139  510  173  128  375  914
```

```
[676]  577  532  779  150  981  936   15  554  217  172  419  958  621  576  823
[691]  194 1025  980   59  598  261  216  463 1002  665  620  867  238 1069 1024
[706]  103  642  305  260  507 1046  709  664  911  282 1113 1068  147  686  349
[721]  304  551 1090  753  708  955  326 1157 1112  191  730  393  348  595 1134
[736]  797  752  999  370   33 1156  235  774  437  392  639   10  841  796 1043
[751]  414   77   32  279  818  481  436  683   54  885  840 1087  458  121   76
[766]  323  862  525  480  727   98  929  884 1131  502  165  120  367  906  569
[781]  524  771  142  973  928    7  546  209  164  411  950  613  568  815  186
[796] 1017  972   51  590  253  208  455  994  657  612  859  230 1061 1016   95
[811]  634  297  252  499 1038  701  656  903  274 1105 1060  139  678  341  296
[826]  543 1082  745  700  947  318 1149 1104  183  722  385  340  587 1126  789
[841]  744  991  362   25 1148  227  766  429  384  631    2  833  788 1035  406
[856]   69   24  271  810  473  428  675   46  877  832 1079  450  113   68  315
[871]  854  517  472  719   90  921  876 1123  494  157  112  359  898  561  516
[886]  763  134  965  920 1167  538  201  156  403  942  605  560  807  178 1009
[901]  964   43  582  245  200  447  986  649  604  851  222 1053 1008   87  626
[916]  289  244  491 1030  693  648  895  266 1097 1052  131  670  333  288  535
[931] 1074  737  692  939  310 1141 1096  175  714  377  332  579 1118  781  736
[946]  983  354   17 1140  219  758  421  376  623 1162  825  780 1027  398   61
[961]   16  263  802  465  420  667   38  869  824 1071  442  105   60  307  846
[976]  509  464  711   82  913  868 1115  486  149  104  351  890  553  508  755
[991]  126  957  912 1159  530  193  148  395  934  597
```

p.s. visos sekos atspausdinti nepavyko, kadangi per daug eilučių.

Dabar imkime m = 875. Pastebėkime, kad m nėra dalus iš 4. Išskaidę pirminiais dauginamaisiais gauname, kad $m = 5^3 \cdot 7$. Vadinasi imsime:

```
> m <- m2
> b <- seq(0, 875 - 7*5, 7*5)
> b
 [1]   0  35  70 105 140 175 210 245 280 315 350 385 420 455 490 525 560 595 630 665 700
735 770
[24] 805 840
```

Remdamiesi galingumo apibrėžimu ir tuo, kad daugiklis turi būti didelis pasirenkame b = 840, nes galimas galingumas yra s = 2,3:

```
> (b^2) %% m
 [1]   0 350 525 525 350   0 350 525 525 350   0 350 525 525 350   0 350 525 525 350   0 350
525
[24] 525 350
> (b^3) %% m
 [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> a <- b[25] + 1
```

Vadinasi, a = 841. Prieauglį antrajai sekai randame analogiškai. Paskaičiuojame $c \approx m \cdot c1$ arba $c \approx m \cdot c2$ ir pasirenkame tą, kuris yra pirminis su m. Tuomet:

```
> c(m*c1, m*c2)
[1] 184.9093 690.0907
> gcd(184, m)
[1] 1
> gcd(185,m)
[1] 5
> gcd(690,m)
[1] 5
> gcd(691,m)
[1] 1
```

Taigi, pasirinkome c = 184. Gavome rinkinį (x_0,a,c,m) = (0,841,184,875). Dabar jau galime generuoti antrąją seką:

```
> for (i in 1:(m-1)) x2[i+1] <- (a * x2[i] + c) %% m
> x2

  [1]   0 247 599   6  43 535 432 434 366  53 195 617 269 726  63 730 802 104 211  73 390 112
814
 [24] 571  83  50 297 649  56  93 585 482 484 416 103 245 667 319 776 113 780 852 154 261
123 440
 [47] 162 864 621 133 100 347 699 106 143 635 532 534 466 153 295 717 369 826 163 830  27
204 311
 [70] 173 490 212  39 671 183 150 397 749 156 193 685 582 584 516 203 345 767 419   1 213
5  77
 [93] 254 361 223 540 262  89 721 233 200 447 799 206 243 735 632 634 566 253 395 817 469
51 263
[116]  55 127 304 411 273 590 312 139 771 283 250 497 849 256 293 785 682 684 616 303
445 867 519
[139] 101 313 105 177 354 461 323 640 362 189 821 333 300 547  24 306 343 835 732 734
666 353 495
[162]  42 569 151 363 155 227 404 511 373 690 412 239 871 383 350 597  74 356 393  10 782
784 716
[185] 403 545  92 619 201 413 205 277 454 561 423 740 462 289  46 433 400 647 124 406 443
60 832
[208] 834 766 453 595 142 669 251 463 255 327 504 611 473 790 512 339  96 483 450 697
174 456 493
[231] 110   7   9 816 503 645 192 719 301 513 305 377 554 661 523 840 562 389 146 533 500
747 224
[254] 506 543 160  57  59 866 553 695 242 769 351 563 355 427 604 711 573  15 612 439 196
583 550
[277] 797 274 556 593 210 107 109  41 603 745 292 819 401 613 405 477 654 761 623  65 662
489 246
[300] 633 600 847 324 606 643 260 157 159  91 653 795 342 869 451 663 455 527 704 811
673 115 712
[323] 539 296 683 650  22 374 656 693 310 207 209 141 703 845 392  44 501 713 505 577 754
861 723
```

[346] 165 762 589 346 733 700 72 424 706 743 360 257 259 191 753 20 442 94 551 763 555
627 804
[369] 36 773 215 812 639 396 783 750 122 474 756 793 410 307 309 241 803 70 492 144 601
813 605
[392] 677 854 86 823 265 862 689 446 833 800 172 524 806 843 460 357 359 291 853 120
542 194 651
[415] 863 655 727 29 136 873 315 37 739 496   8 850 222 574 856 18 510 407 409 341 28
170 592
[438] 244 701 38 705 777 79 186 48 365 87 789 546 58 25 272 624 31 68 560 457 459
391 78
[461] 220 642 294 751 88 755 827 129 236 98 415 137 839 596 108 75 322 674 81 118 610
507 509
[484] 441 128 270 692 344 801 138 805   2 179 286 148 465 187 14 646 158 125 372 724 131
168 660
[507] 557 559 491 178 320 742 394 851 188 855 52 229 336 198 515 237 64 696 208 175 422
774 181
[530] 218 710 607 609 541 228 370 792 444 26 238 30 102 279 386 248 565 287 114 746 258
225 472
[553] 824 231 268 760 657 659 591 278 420 842 494 76 288 80 152 329 436 298 615 337 164
796 308
[576] 275 522 874 281 318 810 707 709 641 328 470 17 544 126 338 130 202 379 486 348
665 387 214
[599] 846 358 325 572 49 331 368 860 757 759 691 378 520 67 594 176 388 180 252 429 536
398 715
[622] 437 264 21 408 375 622 99 381 418 35 807 809 741 428 570 117 644 226 438 230 302
479 586
[645] 448 765 487 314 71 458 425 672 149 431 468 85 857 859 791 478 620 167 694 276 488
280 352
[668] 529 636 498 815 537 364 121 508 475 722 199 481 518 135 32 34 841 528 670 217 744
326 538
[691] 330 402 579 686 548 865 587 414 171 558 525 772 249 531 568 185 82 84 16 578 720
267 794
[714] 376 588 380 452 629 736 598 40 637 464 221 608 575 822 299 581 618 235 132 134 66
628 770
[737] 317 844 426 638 430 502 679 786 648 90 687 514 271 658 625 872 349 631 668 285
182 184 116
[760] 678 820 367 19 476 688 480 552 729 836 698 140 737 564 321 708 675 47 399 681 718
335 232
[783] 234 166 728 870 417 69 526 738 530 602 779 11 748 190 787 614 371 758 725 97 449
731 768
[806] 385 282 284 216 778 45 467 119 576 788 580 652 829 61 798 240 837 664 421 808 775
147 499
[829] 781 818 435 332 334 266 828 95 517 169 626 838 630 702   4 111 848 290 12 714 471
858 825
[852] 197 549 831 868 485 382 384 316   3 145 567 219 676 13 680 752 54 161 23 340 62
764 521
[875] 33

2 Užduotis.

Gautas sekas patikrinkite su dviem testais. Pirma su intervalu testu. Imkite intervalą [2/3, 1). Kitą testą pasirinkite patys.

Gautas sekas patikrinsime intervalų ir kėlinių (k = 2) testais.

Pirmiausiai sukuriame seką $U_n = \frac{x_n}{m}$ tolygiai pasiskirsčiusią intervale [0,1]. Seką reikia suskirstyti į intervalus $U_j, U_{j+1}, \dots, U_{j+k}$, kuriuose tik paskutinis narys U_{j+k} patenka į intervalą [2/3, 1). Generavimo proceso metu reikia generuoti tiek U_i dydžių , kad turėtume iš viso n intervalų. Pasirenkamas skaičius t ir suskaičiuojama, kiek yra 0 ilgio intervalų, kiek yra 1 ilgio intervalų, ..., kiek yra t ilgio intervalų, ir kiek yra intervalų, kurių ilgis yra didesnis už t. Taigi, suskirstome intervalų ilgius į $r = t + 2$ grupes.
Pažymėkime $\beta - \alpha = p$. Tuomet, tikimybės p_i , kad intervalas turi ilgį i yra :
$p_i = p(intervalo\ ilgis = i) = p(1-p)^i$, 0≤ $i \leq t$
$p_i = P(intervalo\ ilgis\ > t) = (1-p)^{t+1}$

Intervalų testui pasirenkame t=5, tuomet r=t+2=7. Galima taikyti χ^2 kriterijų parinkus n ir t taip, kad $np_i > 5$.

Intervalų testas

1 seka (kai m = 1168)

```
> t = 5
> U = c()
> alpha = 2/3
> beta = 1
> U = x1/m
> pat.int = which((U >= alpha) & (U < beta))
> ilgiai = (pat.int - 1) - c(0, pat.int[-length(pat.int)])
> rez = matrix(0, 1, 7, dimnames = list(c(""),c("0-ilgio", "1-ilgio", "2-ilgio", "3-ilgio", "4-ilgio",
"5-ilgio", ">5-ilgio")))
> rez[1] = sum(ilgiai == 0)
> rez[2] = sum(ilgiai == 1)
> rez[3] = sum(ilgiai == 2)
> rez[4] = sum(ilgiai == 3)
```

```
> rez[5] = sum(ilgiai == 4)
> rez[6] = sum(ilgiai == 5)
> rez[7] = sum(ilgiai > 5)
> rez
0-ilgio 1-ilgio 2-ilgio 3-ilgio 4-ilgio 5-ilgio >5-ilgio
     73      33      51       0      50      56       28
```

0-ilgio	1-ilgio	2-ilgio	3-ilgio	4-ilgio	5-ilgio	>5-ilgio
73	33	51	0	50	56	28

Lentelėje matome, kad 0-ilgio intervalų yra 73, 1-ilgio intervalų yra 3, 2-ilgio – 51 intervalas, 4-ilgio 50 intervalų, 5-ilgio – 56 intervalai ir didesni už 5-ilgio 28 intervalai .

Taikome χ^2 kriterijų. Pasirinkus reikšmingumo lygmenį lygų 0.05 χ^2 su 6 laisvės laipsniais kritinė reikšmė yra 12.592.

```
> p = beta - alpha
> tik = c(p*(1-p)^(0:t), (1-p)^(t+1))
> chisq.test(rez, p = tik)
```

Chi-squared test for given probabilities

data: rez
X-squared = 247.78, df = 6, p-value < 2.2e-16

Kadangi 247.78 > 12.592, tai atmetame nulinę hipotezę, kad seka yra tolygiai pasiskirsčiusi.

2 seka (kai m = 875)

```
> U2 = c()
> U2 = x2/m
> pat.int2 = which((U2 >= alpha) & (U2 < beta)) #Kurie sekos nariai priklauso intervalui
> ilgiai2 = (pat.int2 - 1) - c(0, pat.int2[-length(pat.int2)])
> rez2 = matrix(0, 1, 7, dimnames = list(c(""),c("0-ilgio", "1-ilgio", "2-ilgio", "3-ilgio", "4-
ilgio", "5-ilgio", ">5-ilgio")))
> rez2[1] = sum(ilgiai2 == 0)
> rez2[2] = sum(ilgiai2 == 1)
> rez2[3] = sum(ilgiai2 == 2)
> rez2[4] = sum(ilgiai2 == 3)
> rez2[5] = sum(ilgiai2 == 4)
```

```
> rez2[6] = sum(ilgiai2 == 5)
> rez2[7] = sum(ilgiai2 > 5)
> rez2

0-ilgio 1-ilgio 2-ilgio 3-ilgio 4-ilgio 5-ilgio >5-ilgio
     94      65      47      21      22      20      22
```

0-ilgio	1-ilgio	2-ilgio	3-ilgio	4-ilgio	5-ilgio	>5-ilgio
94	65	47	21	22	20	22

Lentelėje matome, kad 0-ilgio intervalų yra 94, 1-ilgio intervalų yra 65, 2-ilgio intervalų yra 47, 3-ilgio intervalų yra 21, 4-ilgio intervalų yra 22, 5-ilgio intervalų yra 20 ir 22 intervalai, kurių ilgis didesnis už 5.

Taikome χ^2 kriterijų. Pasirinkus reikšmingumo lygmenį lygų 0.05 χ^2 su 6 laisvės laipsniais kritinė reikšmė yra 12.592.

```
> p = beta - alpha
> tik = c(p*(1-p)^(0:t), (1-p)^(t+1))
> chisq.test(rez2, p = tik)
```

Chi-squared test for given probabilities

data: rez2
X-squared = 7.5316, df = 6, p-value = 0.1205

Kadangi 7.5316 < 12.592, tai nėra pagrindo atmesti nulinę hipotezę, kad seka yra tolygiai pasiskirsčiusi. **Remiantis intervalų testu gavome, kad 2 seka yra tolygiai pasiskirsčiusi.**

Kėlinių (k = 2) testas

Pradinė seka suskaidoma į *n* grupių po *k* elementų kiekvienoje iš jų. Paskaičiuojama kiek kartų kiekvienas konkretus dėstinys sutinkamas tarp n grupių. Tada pritaikomas χ^2 kriterijus su $r = k!$ ir tikimybėmis $p_i = 1/k!$.

Antrasis testas yra kėlinių testas. Imkime $k = 2$. Tuomet galimos dvi išsidėstymo grupės:

$$U_{2j} < U_{2j+1}$$

arba

$$U_{2j} > U_{2j+1}.$$

Sudarinėsime poras ir tikrinsime, kurį išsidėstymą jos atitinka ir atitinkamai suskaičiuosime, kiek kokių yra.

```
> n <- m1
```

```
> u <- x1/n
> k <- 2
> maziau <- 0
> daugiau <- 0
> v1 <- u[-length(u)]
> v2 <- u[-1]
> maziau <- as.numeric(length(which(v1 < v2)))
> daugiau <- as.numeric(length(which(v1 >= v2)))
> d <- c(maziau, daugiau)
> d
[1] 483 684
```

Skirtumas yra ganėtinai didelis. Tikėtina, jog šie dydžiai statistiškai reikšmingai skiriasi. Formaliai tuo įsitikinsime atlikę χ^2 testą. Kai seka yra atsitiktinė, tai kiekvieno išsidėstymo pasirodymo tikimybė yra lygi $\frac{1}{k!} = 1/2$. Dabar galime pritaikyti χ^2 testą:

```
> prob <- c(1/2, 1/2)
> chisq.test(d, p=prob)
```

Chi-squared test for given probabilities

data: d
X-squared = 34.62, df = 1, p-value < 2.2e-16

Su reikšmingumo lygmeniu 0,05 nulinę hipotezę apie sekos atsitiktinumą atmesime.

Pakartosime šį testą antrajai sekai (m = 875). Analogiškai tikrinsime porų išsidėstymą ir skaičiuosime, kiek kuriai grupei porų priklauso:

```
> n <- m2
> u <- x2/n
> k <- 2
> maziau <- 0
> daugiau <- 0
> v1 <- u[-length(u)]
> v2 <- u[-1]
> maziau <- as.numeric(length(which(v1 < v2)))
```

```
> daugiau <- as.numeric(length(which(v1 > v2)))
> d <- c(maziau, daugiau)
> d
[1] 453 421
```

Šie skaičiai maždaug vienodi. pritaikome χ^2 testą:

```
> prob <- c(1/2, 1/2)
> chisq.test(d, p=prob)
```

Chi-squared test for given probabilities

data: d
X-squared = 1.1716, df = 1, p-value = 0.2791

Taigi, atmesti nulinės hipotezės, kad seka yra atsitiktinė, negalime.

Galime daryti išvadą, jog antroji seka yra geresnė. Tolesnį darbą tęsime, būtent, su ja.

3 Užduotis.

Tegul U_1 ir U_2 – nepriklausomi ir tolygiai pasiskirstę intervale [0,1) atsitiktiniai dydžiai. Tuomet atsitiktiniai dydžiai $V_1 = 2U_1 - 1$ ir $V_2 = 2U_2 - 1$ yra nepriklausomi ir tolygiai pasiskirstę intervale [-1,1). Pažymėkime $S = V_1^2 + V_2^2$. Jeigu $S \geq 1$, imkime naujus U_1 ir U_2 bei generuokime S iš naujo. Imkime tik $0 \leq S < 1$. Tada $X_1 = V_1\sqrt{\frac{-2\ln S}{S}}$, $X_2 = V_2\sqrt{\frac{-2\ln S}{S}}$ yra nepriklausomi ir normaliai pasiskirstę atsitiktiniai dydžiai, t.y. $X_1, X_2 \in N(0,1)$, o jų pasiskirstymo funkcija $\Phi(x) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-\frac{t^2}{2}}dt$. Tokiu būdu galime gauti standartinį normalųjį atsitiktinį dydį. O dydis $Y = \mu + \sigma X, X \in N(0,1)$ yra pasiskirstęs pagal $N(\mu,\sigma)$.

Norint gauti tokį atsitiktinį dydį, pasiskirsčiusį pagal N(0.5,0.5) normalųjį skirstinį:

- Pirmiausia sumodeliuosiu atsitiktinį dydį X ,pasiskirsčiusį pagal normalųjį skirstinį N(0,1)
- Tada remiantis, kad $Y = \mu + \sigma X, X \in N(0,1)$ yra pasiskirstęs pagal $N(\mu,\sigma)$: sumodeliuosiu atsitiktinį dydį: $Y = 1 + \sqrt{2}X$, kuris ir bus Normalusis skirstinys N(0.5, 0.5)
- Padariusi visa tai gavau tokį **atsitiktinį dydį, pasiskirsčiusi pagal N(0.5,0.5):**

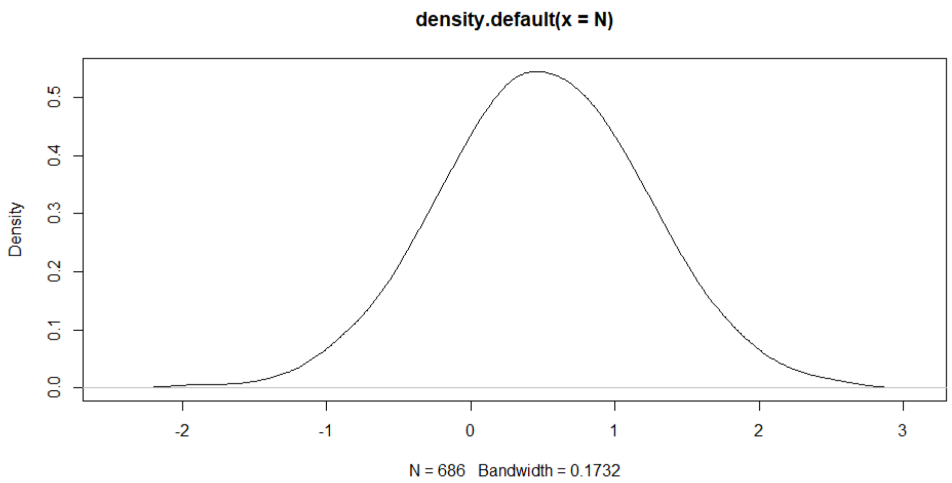
```
> mean(Y)
[1] 0.4976536
> var(Y)
[1] 0.5049527

> U=x2/m2
> V=c()
> for (i in 1:m2){
+   V[i]=2*U[i]-1
+ }
> X=c();L=c();S=c();r=1;i=1
> while (i<m2){
+   S[i]=V[i]^2+V[i+1]^2
+   if (S[i] >= 1) i=i+1 else {
+     L[r]=logb(S[i], base = exp(1))
+     X[r]=V[i]*((-2*L[r])/(S[i]))^(1/2)
+     r=r+1
+     i=i+1}
+ }
> Y=0.5+sqrt(0.5)*X

> plot(density(Y))
> jarque.bera.test(Y)
```

Jarque Bera Test

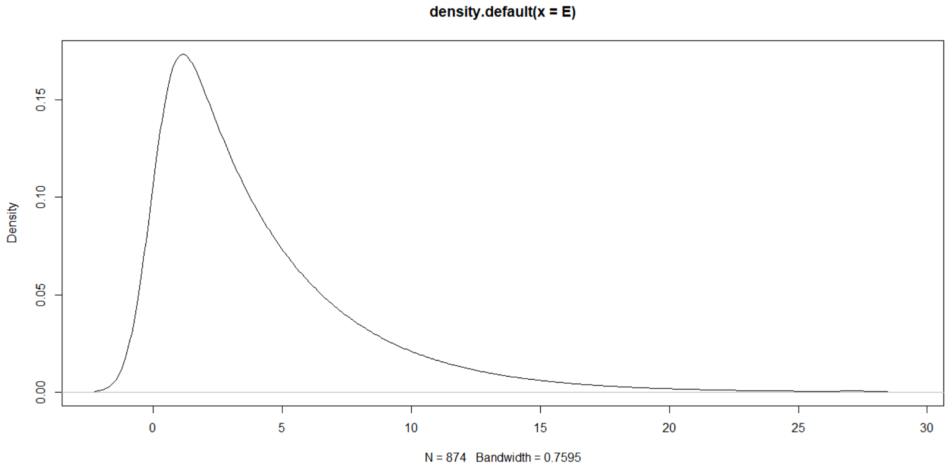
data: Y
X-squared = 0.17608, df = 2, p-value = 0.9157



Iš grafiko ir testo galime įsitikinti, kad sugeneravome reikiamą a. d.

Kitas a. d. bus paprastai sugeneruojamas eksponentinis a. d. su parametru $\lambda = 4$. Pasinaudosime atvirkštinės transformacijos metodu. Tarkim turime tolygiai pasiskirsčiusią intervale [0,1] seką $U_1, U_2, \ldots, U_n$. Išmetame iš sekos tuos stebėjimus, kurie yra lygūs nuliui. Tuomet a. d. $E = -4\log(U)$ turi eksponentinį skirstinį su parametru 4. Padarykime tai su R'u:

```
> E <- - 4 * log(u[- which(u == 0)]) #išmetame tuos stebėjimus, kurie yra lygūs nuliui.
> plot(density(E))
```



Gavome tai, ko ir reikėjo.

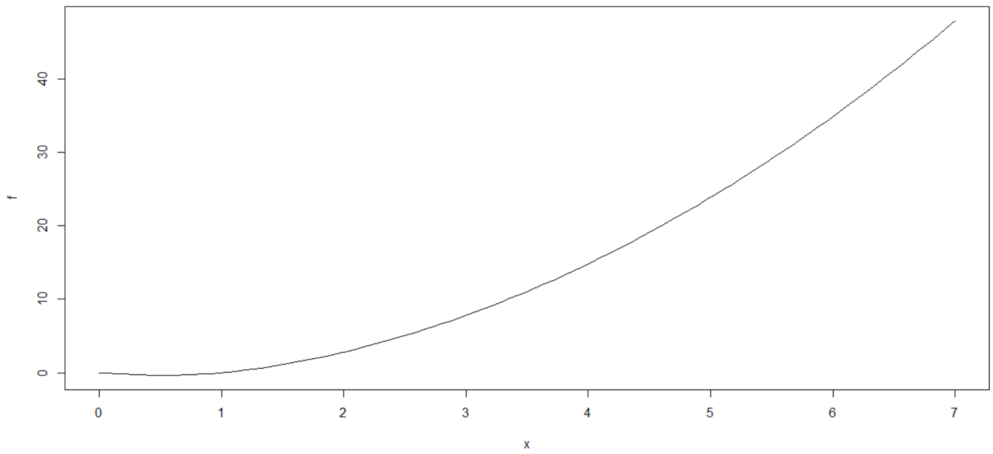
4 Užduotis.

Ketvirtoje užduotyje intervale [0,7] suintegruosime funkciją:

$$f(z) = \frac{z^4 - z}{z^2 + 1}$$

Pirmiausiai šį integralą I paskaičiuokime skaitiniu metodu naudodami R paketą:

```
> f <- function(z) {(z^4-z)/(z^2+1)}
> integrate(f,0,7)
106.8062 with absolute error < 0.00014
> plot(f, xlim=c(0,7))
```



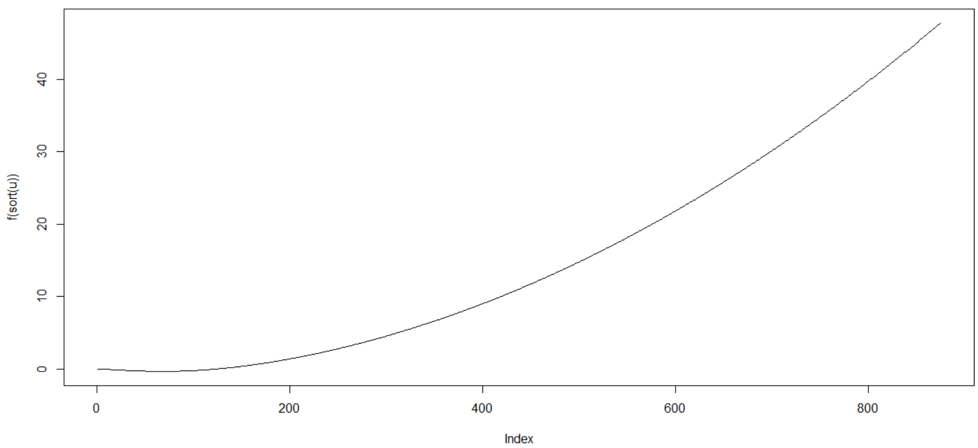
Matome, kad su labai maža paklaida I = 106.8062.

Dabar paskaičiuokime šį integralą panaudodami sugeneruotąją p. a. s. seką. Pasidarome naują seką $U_1, U_2, ..., U_n$, kuri yra tolygiai pasiskirsčiusi intervale [0,7]:

```
> n <- m2
> u <- 7 * x2 / n
```

Dabar imame vidutinę šių taškų funkcijos reikšmę ir padauginame ją iš 7−0 = 7:

```
> I <- mean(f(u))*7
> I
[1] 106.6148
> plot(f(sort(u)), type="l")
```



Pastebėkime, kad šie du grafikai yra praktiškai identiški, todėl ir integralo reikšmė yra beveik vienoda. Taip yra todėl, kad turime didelę imtį.

5 Užduotis.

Kadangi turime stacionarųjį skirstinį

$$\pi = (1/6, 2/6, 3/6, 0),$$

Užduotyje nurodytas stacionarus skirstinys iš 4 elementų – taigi modeliuosime Markovo grandinę su būsenų aibe $S = \{s_1, s_2, s_3, s_4\}$. Atsitiktiniam procesui $(X_0, X_1, \ldots)$ su baigtine būsenų aibe, vadinamam Markovo grandine, apibrėžiame perėjimo matricą $\mathbb{P}$, kurios elementai P_{ij} yra vadinami perėjimo tikimybėmis: pereiti į būseną s_j, jei dabar esama būsenoje s_i. Kiekviena perėjimo matrica turi tenkinti dvi savybes:

$P_{ij} \geq 0\ \forall i, j \in \{1, \ldots, k\}$ ir $\sum_{j=1}^{k} P_{ij} = 1\ \forall i \in \{1, \ldots, k\}$.

1 apibrėžimas. Tegul $(X_0, X_1, \ldots)$ yra Markovo grandinė su baigtine būsenų aibe S ir perėjimo matrica $\mathbb{P}$. Vektorius – eilutė $\pi = (\pi_1, \ldots, \pi_k)$ yra vadinamas Markovo grandinės **stacionariuoju** skirstiniu, jei

i. $\pi_i \geq 0\ \forall\ i = 1, \ldots, k$ ir $\sum_{i=1}^{k} \pi_i = 1$;

ii. $\pi\ \mathbb{P} = \pi$, t.y. $\sum_{i=1}^{k} \pi_{ij} P_{ij} = \pi_j\ \forall j = 1, \ldots, k$.

2 apibrėžimas. Tegul $(X_0, X_1, \ldots)$ yra Markovo grandinė su baigtine būsenų aibe S ir perėjimo matrica $\mathbb{P}$. Jei π yra šios Markovo grandinės **apgręžiamas** skirstinys, tai jis yra ir **stacionarus** šios grandinės skirstinys.

3 apibrėžimas. Tegul $(X_0, X_1, \ldots)$ yra Markovo grandinė su baigtine būsenų aibe S ir perėjimo matrica $\mathbb{P}$. Tikimybinis skirstinys π aibėje S yra vadinamas **apgręžiamu**, jei $\forall\ i, j = 1, \ldots, k$ teisinga lygybė: $\pi_i P_{ij} = \pi_j P_{ji}$.

1 teorema. Gimimo – mirties procesas. Tegul $(X_0, X_1, \ldots)$ yra Markovo grandinė su baigtine būsenų aibe S ir perėjimo matrica $\mathbb{P}$. Be to, tegul matrica $\mathbb{P}$ tenkina savybes:

i. $P_{ij} > 0$, jei $|i - j| = 1$,

ii. $P_{ij} = 0$, jei $|i - j| \geq 2$.

Tokia Markovo grandinė vadinama gimimo – mirties procesu.

Dabar galime sukonstruoti perėjimo matricą.

Tai perėjimo matrica bus

$$P = \begin{pmatrix} 1/6 & 2/6 & 3/6 & 0 \\ 1/6 & 2/6 & 3/6 & 0 \\ 1/6 & 2/6 & 3/6 & 0 \\ 1/6 & 2/6 & 3/6 & 0 \end{pmatrix}$$

Vadinasi, gimimo - mirties proceso tikrai negausime. Kadangi perėjimo matricos visos eilutės vienodos, tai nesvarbu, kurioje būsenoje esame, pereiti į kitą būseną (arba pasilikti toje pačioje), galėsime su tomis pačiomis tikimybėmis. Pradinis skirstinys sutaps su stacionariuoju skirstiniu. Kadangi žinome, jog pradinė sugeneruotos p. a. s. sekos reikšmė yra nulis, tai Markovo grandinės pirmoji būsena bus lygi vienam ($0 < 0{,}16666667$). Generavimas R paketu realizuojamas gana paprastai:

```
> mcsd <- c(1/6, 2/6, 3/6, 0)
> P <- matrix(rep(mcsd, 4), ncol=4, byrow=TRUE)
> u <- x2/m2
> b <- c()
> b[1] <- 1
> for (i in 1:(m2-1)){
+   if (u[i] <= 0.16666667) {b[i+1] <- 1}
+    else {if (u[i] <= 0.5) {b[i+1] <- 2}
+       else {b[i+1] <- 3}}
+          }
b
```

```
  [1] 1 1 2 3 1 1 3 2 2 2 1 2 3 2 3 1 3 3 1 2 1 2 1 3 3 1 1 2 3 1 1 3 3 3 2 1 2 3 2
 [40] 3 1 3 3 2 2 1 3 2 3 3 1 1 2 3 1 1 3 3 3 3 2 2 3 2 3 2 3 1 2 2 2 3 2 1 3 2 2 2
 [79] 3 2 2 3 3 3 3 2 2 3 2 1 2 1 1 2 2 2 3 2 1 3 2 2 3 3 2 2 3 3 3 3 2 2 3 3 1 2 1
[118] 1 2 2 2 3 2 1 3 2 2 3 3 2 2 3 3 3 3 2 3 3 3 1 2 1 2 2 3 2 3 2 2 3 2 2 3 1 2 2
[157] 3 3 3 3 2 3 1 3 2 2 2 2 2 3 2 3 2 2 3 2 2 3 1 2 2 1 3 3 3 2 3 1 3 2 2 2 2 3 3
[196] 2 3 3 2 1 2 2 3 1 2 3 1 3 3 3 3 3 1 3 2 3 2 2 3 3 3 3 3 2 1 3 3 3 2 3 3 1 1 1
[235] 3 3 3 2 3 2 3 2 2 3 3 3 3 3 2 2 3 3 3 2 3 3 2 1 1 3 3 3 2 3 2 3 2 2 3 3 3 1 3
[274] 3 2 3 3 3 2 3 3 2 1 1 1 3 3 2 3 2 3 2 3 3 3 3 1 3 3 2 3 3 3 2 3 3 2 2 2 1 3 3
[313] 2 3 3 3 3 3 3 3 3 1 3 3 2 3 3 1 2 3 3 2 2 2 1 3 3 2 1 3 3 3 3 3 3 3 2 3 3 2 3
[352] 3 1 2 3 3 2 2 2 2 3 1 3 1 3 3 3 3 3 1 3 2 3 3 2 3 3 1 3 3 3 2 2 2 2 3 1 3 1 3
[391] 3 3 3 3 1 3 2 3 3 3 3 3 2 3 3 3 3 2 2 2 3 1 3 2 3 3 3 3 1 1 3 2 1 3 3 1 3 2 3
[430] 3 1 3 2 2 2 1 2 3 2 3 1 3 3 1 2 1 2 1 3 3 1 1 2 3 1 1 3 3 3 2 1 2 3 2 3 1 3 3
[469] 1 2 1 2 1 3 3 1 1 2 3 1 1 3 3 3 3 1 2 3 2 3 1 3 1 2 2 2 3 2 1 3 2 1 2 3 1 2 3
[508] 3 3 3 2 2 3 2 3 2 3 1 2 2 2 3 2 1 3 2 2 2 3 2 2 3 3 3 3 2 2 3 3 1 2 1 1 2 2 2
[547] 3 2 1 3 2 2 3 3 2 2 3 3 3 3 2 2 3 3 1 2 1 2 2 2 2 3 2 2 3 2 2 3 3 2 2 3 3 3 3
[586] 2 3 1 3 1 2 1 2 2 3 2 3 2 2 3 2 2 3 1 2 2 3 3 3 3 2 3 1 3 2 2 2 2 2 3 2 3 2 2
[625] 1 2 2 3 1 2 2 1 3 3 3 2 3 1 3 2 3 2 2 3 3 3 3 3 2 1 3 2 3 2 2 3 1 3 3 3 3 3 2
[664] 3 2 3 2 2 3 3 3 3 3 2 1 3 3 3 2 3 3 1 1 1 3 3 3 2 3 2 3 2 2 3 3 3 3 3 2 2 3 3
[703] 3 2 3 3 2 1 1 1 3 3 2 3 2 3 2 3 3 3 3 1 3 3 2 3 3 3 2 3 3 2 1 1 1 3 3 2 3 2 3
[742] 2 3 3 3 3 1 3 3 2 3 3 3 2 3 3 2 2 2 1 3 3 2 1 3 3 3 3 3 3 3 1 3 3 2 3 3 1 2 3
[781] 3 2 2 2 2 3 3 2 1 3 3 3 3 3 1 3 2 3 3 2 3 3 1 3 3 3 2 2 2 2 3 1 3 1 3 3 3 3 3
```

[820] 1 3 2 3 3 2 3 3 2 3 3 3 2 2 2 2 3 1 3 2 3 3 3 3 1 1 3 2 1 3 3 3 3 2 3 3 3 3 2
[859] 2 2 1 1 3 2 3 1 3 3 1 2 1 2 1 3 3

Gavome tai, ko reikėjo. Užduotis baigta.