Assignment 1

Sourav Sarkar

11 February 2016

Contents

1 Q1. Generate 100 random from a Bivariate Generalized exponential, Bivariate Weibull distribution.

1.1 Bivariate Generalized Exponential Distribution

To calculate BVGE we will first generate a GE using *genExp*.

```
genExp<-function(alpha, lambda){
  (-(1/lambda) * log(1 - (runif(1)^(1/alpha))))
}
```

Lets generate 100 BVGE with parameters $(\lambda, \alpha_1, \alpha_2, \alpha_3) = (0.04, 1, 2, 0.5)$.

```
N = 100
X = matrix(NA, N, 2)

i = 1
while(i <= N) {
    u_0 = genExp(alpha[1], lambda)
    u_1 = genExp(alpha[2], lambda)
    u_2 = genExp(alpha[3], lambda)
    X[i,] = c(max(u_0, u_1), max(u_0, u_2))
    i = i + 1
}

X
```

```
##              [,1]      [,2]
##   [1,]  75.261737  75.261737
```

```
##   [2,]  34.759710   6.098882
##   [3,]  34.447019  32.528619
##   [4,]  64.898659  64.898659
##   [5,]  27.775730  27.775730
##   [6,]  50.683957  21.630647
##   [7,]  54.628470  38.591701
##   [8,]  14.979264   9.833217
##   [9,]  90.118060  77.757317
##  [10,]  19.079674  30.983857
##  [11,]  63.397236  63.397236
##  [12,]  15.950122  49.931611
##  [13,]  24.008456  25.991248
##  [14,]  28.110829  28.110829
##  [15,]  25.804315  25.804315
##  [16,]  39.543601  10.942091
##  [17,]  73.469483  31.721465
##  [18,]  62.327929   4.644148
##  [19,] 162.564042 162.564042
##  [20,]  35.761495  13.709846
##  [21,]  56.688396  32.608536
##  [22,]  51.186353  51.186353
##  [23,]  55.760596  16.083679
##  [24,]  33.504431  12.006770
##  [25,]  78.650709  10.554889
##  [26,]  17.150956   1.096294
##  [27,]  17.742161  11.468341
##  [28,]  55.740694   8.265121
##  [29,]  57.529683  57.529683
##  [30,]  14.570845   7.349198
##  [31,]  31.566062  31.566062
##  [32,]  41.792008  38.853224
##  [33,]  86.835936  86.835936
##  [34,]  28.537797  12.505912
##  [35,]  35.037567  35.037567
##  [36,]  22.311876  16.492583
##  [37,]  46.937160   3.202666
##  [38,]  59.159236  59.159236
##  [39,] 118.745760 118.745760
##  [40,] 111.030808  20.949105
##  [41,]  21.825781  31.260178
##  [42,]   4.048259   6.246982
##  [43,]  39.492960  33.199246
##  [44,]  74.095366  65.084759
##  [45,]  57.210030  10.659956
##  [46,]  38.778431  85.151922
##  [47,]  37.299565  34.055943
##  [48,]  15.915295  15.915295
##  [49,]  84.209227 110.110286
##  [50,]  63.829005  78.975819
##  [51,]  35.617427  26.045818
##  [52,] 118.456288 118.456288
##  [53,]  19.232761  14.075188
##  [54,] 113.841665  22.785290
##  [55,]  52.906304  52.906304
```

```
##  [56,]  35.618309  35.618309
##  [57,]  66.234375   6.554898
##  [58,]  98.226465  24.742346
##  [59,]  33.740954  24.458537
##  [60,]  53.873053  35.063658
##  [61,]  68.703923  65.762914
##  [62,]  59.931626  59.931626
##  [63,]  73.134967  73.134967
##  [64,] 140.332882 100.529219
##  [65,]  22.313461  22.313461
##  [66,]  22.676364   6.050316
##  [67,]  24.297769  15.765221
##  [68,]  59.360843   3.167281
##  [69,]  48.951912  15.958934
##  [70,]  46.253468  46.253468
##  [71,] 102.306235 102.306235
##  [72,]  78.557227  78.557227
##  [73,]  80.488762   7.191763
##  [74,]  50.528666  40.838360
##  [75,]  19.208921   4.793367
##  [76,]  45.512643  24.118033
##  [77,]  40.160985  44.105826
##  [78,]   6.747788  34.156489
##  [79,]  21.394085   9.519606
##  [80,]  28.194290  12.194475
##  [81,]  51.457120  30.236116
##  [82,]  18.153108   8.887474
##  [83,]  63.561831  20.690250
##  [84,]  16.553138  25.129956
##  [85,]  32.605282  21.478017
##  [86,]  28.294441  51.207744
##  [87,]  56.711544  56.711544
##  [88,]  43.449519  43.449519
##  [89,]  88.799347  88.799347
##  [90,]  59.650016  59.650016
##  [91,] 123.287831   7.741560
##  [92,]  46.592821  26.131382
##  [93,]  33.298807   3.546861
##  [94,]  14.154009  13.534068
##  [95,]  63.373995   3.029261
##  [96,]  26.809435  10.360186
##  [97,]  25.843980  25.843980
##  [98,]  37.770178  37.770178
##  [99,]  98.622653  98.622653
## [100,]  43.721012   9.312461
```

1.2 Bivariate Weibull Distribution

To calculate BW we will first generate a Weibull using *genWeibull*.

```
genWeibull<-function(lambda, alpha){
  (((-log(1 - (runif(1)^(1/lambda))))^(1/alpha)))
}
```

Lets generate 100 BW with parameters $(\alpha, \lambda_1, \lambda_2, \lambda_3) = (2, 0.5, 1, 2)$.

```
N = 100
X = matrix(NA, N, 2)

i = 1
while(i <= N) {
    u_0 = genWeibull(lambda[1], alpha)
    u_1 = genWeibull(lambda[2], alpha)
    u_2 = genWeibull(lambda[3], alpha)
    X[i,] = c(max(u_0, u_1), max(u_0, u_2))
    i = i + 1
}

X
```

```
##             [,1]      [,2]
##   [1,] 0.2750139 1.0757377
##   [2,] 1.2404192 0.9692566
##   [3,] 0.3302639 0.8245903
##   [4,] 1.8409034 1.8409034
##   [5,] 1.2205111 0.7156043
##   [6,] 0.7283177 0.8833713
##   [7,] 2.8365415 2.8365415
##   [8,] 1.8767641 1.0506367
##   [9,] 2.3479123 1.2585923
##  [10,] 0.9241678 0.9931425
##  [11,] 1.2550641 1.3939921
##  [12,] 2.1455687 2.1455687
##  [13,] 1.4353385 0.9175502
##  [14,] 0.8766751 0.8766751
##  [15,] 0.5132323 1.5996778
##  [16,] 1.2987772 0.4906242
##  [17,] 0.6135403 1.4417312
##  [18,] 2.1917424 2.1917424
##  [19,] 1.4974836 1.4974836
##  [20,] 0.6297661 1.1164879
##  [21,] 0.5751490 1.1134314
##  [22,] 0.8858369 0.7415648
##  [23,] 0.7292640 0.9160692
##  [24,] 1.6022287 1.2068723
##  [25,] 1.4624313 1.3290872
##  [26,] 1.1950642 1.1454899
##  [27,] 0.7423398 1.1684946
##  [28,] 1.3899244 1.5927890
##  [29,] 0.8746791 1.3839697
##  [30,] 0.5071619 1.0712178
##  [31,] 0.5690653 0.9335970
##  [32,] 1.2682233 1.2682233
##  [33,] 0.5681257 1.2983655
##  [34,] 1.3445021 1.4923192
##  [35,] 1.1350971 1.7026982
##  [36,] 1.5171384 1.1341920
##  [37,] 0.9893825 1.8695618
```

```
##  [38,] 1.2505853 1.2505853
##  [39,] 0.6629872 1.6698280
##  [40,] 1.5384679 0.7153568
##  [41,] 0.7501651 2.1393896
##  [42,] 0.6209266 1.7991533
##  [43,] 1.1951675 1.1951675
##  [44,] 1.0661876 1.2046914
##  [45,] 1.5506575 1.0981689
##  [46,] 0.6979713 0.7787587
##  [47,] 0.8374978 1.1182699
##  [48,] 1.3201983 1.3201983
##  [49,] 0.6538431 0.8106302
##  [50,] 1.0438946 1.0438946
##  [51,] 1.6382724 1.8335772
##  [52,] 0.8967069 1.5415856
##  [53,] 0.2845800 1.0966414
##  [54,] 0.2937943 0.5959693
##  [55,] 0.8157267 1.1865089
##  [56,] 1.7304244 0.7709779
##  [57,] 0.9641327 0.9641327
##  [58,] 1.1095410 1.1095410
##  [59,] 0.4905547 1.7864279
##  [60,] 1.4770337 1.4580289
##  [61,] 1.4716320 0.7764980
##  [62,] 0.4231334 1.5052572
##  [63,] 0.9207055 1.6692847
##  [64,] 0.8541225 1.3825239
##  [65,] 0.3255503 0.7068747
##  [66,] 1.6609593 1.6083194
##  [67,] 0.7199052 0.8760474
##  [68,] 1.4132883 1.0542712
##  [69,] 1.0548427 1.0548427
##  [70,] 1.4212127 1.4212127
##  [71,] 1.1571475 1.1571475
##  [72,] 1.9039548 1.2029723
##  [73,] 1.8747115 1.4495716
##  [74,] 1.2430015 0.8337790
##  [75,] 0.5578454 0.4163779
##  [76,] 0.5522280 0.7744843
##  [77,] 0.9639062 0.7983927
##  [78,] 1.1880340 1.1880340
##  [79,] 0.8959318 1.3004355
##  [80,] 1.2700041 1.4562438
##  [81,] 1.5622249 0.6504415
##  [82,] 0.4966494 1.5473649
##  [83,] 1.3070725 1.4034855
##  [84,] 2.3283857 2.3283857
##  [85,] 1.4338381 1.3599487
##  [86,] 0.9134427 1.4959542
##  [87,] 1.9524758 1.2131475
##  [88,] 1.3039432 2.1246588
##  [89,] 1.1517282 1.1517282
##  [90,] 1.0758669 0.6974877
##  [91,] 1.3240072 1.0131918
```

```
## [92,] 1.3929176 1.3929176
## [93,] 0.8618756 1.1628052
## [94,] 1.0279477 2.2674708
## [95,] 2.1891051 2.1891051
## [96,] 1.1235170 1.3370961
## [97,] 0.9807383 0.7336026
## [98,] 1.2952894 1.7084950
## [99,] 0.8106716 1.5534555
## [100,] 1.2689368 0.5307114
```

2 Q2. Generate 100 random from an absolutely continuous bivariate Generalized exponential, absolutely continuous Bivariate Weibull distribution.

2.1 Absolutely Continuous Bivariate Generalized Exponential Distribution

To calculate BVGE-AC we will first generate a GE using *genExp*.

```
genExp<-function(alpha, lambda){
  (-(1/lambda) * log(1 - (runif(1)^(1/alpha))))
}
```

Lets generate 100 BVGE-AC with parameters $(\lambda, \alpha_1, \alpha_2, \alpha_3) = (0.04, 1, 2, 0.5)$.

```
N = 100
X = matrix(NA, N, 2)

i = 1
while(i <= N) {
    u_0 = genExp(alpha[1], lambda)
    u_1 = genExp(alpha[2], lambda)
    u_2 = genExp(alpha[3], lambda)
    x1 = max(u_0, u_1)
    x2 = max(u_0, u_2)
    if(x1 == x2) {
        next
    }
    X[i,] = c(x1, x2)
    i = i + 1
}

X
```

```
##              [,1]      [,2]
##   [1,] 166.623143  2.995476
##   [2,]  14.101508 13.571480
##   [3,]  24.945407  3.644292
##   [4,] 170.602406 16.596414
##   [5,] 245.916688  8.218372
##   [6,]  31.142179 50.816810
##   [7,]  16.865562 10.176493
```

```
##   [8,]  84.824042  26.180734
##   [9,]  33.494667  12.689829
##  [10,]  37.645444 142.941610
##  [11,]  18.630140   7.399741
##  [12,]  46.919233   1.039311
##  [13,]  62.485132  32.926388
##  [14,]  22.326685   6.015652
##  [15,]  17.201596  12.093777
##  [16,]  11.205144  10.113635
##  [17,]  63.592825  11.270820
##  [18,]  51.118164  35.121781
##  [19,]  37.212087  15.064262
##  [20,]  31.957090  95.054468
##  [21,] 131.059514  19.527750
##  [22,]  13.587197   8.971881
##  [23,]  30.336378   1.411553
##  [24,]  81.660957   3.111607
##  [25,]  55.980265  58.209563
##  [26,]  81.160713  40.460184
##  [27,]  40.233614  35.379565
##  [28,]  44.817404  25.980321
##  [29,]  70.533717   7.078962
##  [30,]  40.975607  22.554061
##  [31,]   7.597524  28.940312
##  [32,]  38.362948   9.923999
##  [33,] 148.804993  37.192352
##  [34,]   9.871421  19.525137
##  [35,]  33.342454   6.555366
##  [36,]  38.897150   9.586244
##  [37,]  33.194915  43.792320
##  [38,]  37.173969  40.544551
##  [39,]  36.963627   1.661150
##  [40,]  26.220216  19.604363
##  [41,]  19.154948   1.976823
##  [42,]  17.566260  10.843680
##  [43,] 107.863604  22.938380
##  [44,]  41.331912  30.431111
##  [45,]  46.512952  34.863485
##  [46,]  36.967473  50.536334
##  [47,]  23.794342  20.750151
##  [48,]  61.678293  54.466703
##  [49,]  48.593993  12.931412
##  [50,]  19.758776  11.796338
##  [51,]  52.437308  41.911303
##  [52,]  30.522659  14.247339
##  [53,]  85.216816   9.738188
##  [54,]  19.818774   5.225307
##  [55,]  41.723795  38.825726
##  [56,] 154.746125  29.691365
##  [57,]  83.155347  44.062084
##  [58,]  57.964031  26.568405
##  [59,]  25.488444  18.565966
##  [60,]  20.425199  18.555669
##  [61,]  98.096339  21.652586
```

```
## [62,]  42.990392  25.215489
## [63,]  10.928949   6.720731
## [64,]  17.349339   2.782620
## [65,]  29.897748   4.236304
## [66,]  27.575869  21.737578
## [67,] 110.980113  85.212744
## [68,]  40.006935   8.593092
## [69,]  14.446809   5.790400
## [70,]  42.532057   4.594376
## [71,]  78.095878  53.568953
## [72,]  14.316188  24.893120
## [73,]  46.077679  25.365492
## [74,]  70.119716  22.372842
## [75,]  33.979716  17.339368
## [76,]  41.731295  17.403252
## [77,]  39.176505  31.431025
## [78,]  46.641843  25.985916
## [79,]   6.851052   3.086978
## [80,]  77.108736  76.104149
## [81,]  68.048791  43.713614
## [82,]  37.005808  89.806985
## [83,]  50.636486   2.142283
## [84,]  44.182817  32.528308
## [85,]  23.918423  31.131488
## [86,]  31.869938   5.826133
## [87,]  63.978272  57.398700
## [88,] 107.933940   6.313408
## [89,]  32.164199  31.849267
## [90,]  75.644883  50.680514
## [91,]  58.309820   2.729008
## [92,]  49.127921   5.821787
## [93,]   5.342036   1.036605
## [94,]  55.779966  45.826331
## [95,]  45.825651 168.604121
## [96,] 150.513288  36.597048
## [97,]  11.342637  23.642128
## [98,]  27.259334  56.071272
## [99,]  13.522676   2.034529
## [100,]  80.724235  62.605233
```

2.2 Absolutely Continuous Bivariate Weibull Distribution

To calculate BW we will first generate a Weibull using *genWeibull*.

```
genWeibull<-function(lambda, alpha){
  (((-log(1 - (runif(1)^(1/lambda))))^(1/alpha)))
}
```

Lets generate 100 BW-AC with parameters $(\alpha, \lambda_1, \lambda_2, \lambda_3) = (2, 0.5, 1, 2)$.

```
N = 100
X = matrix(NA, N, 2)
```

```r
i = 1
while(i <= N) {
    u_0 = genWeibull(lambda[1], alpha)
    u_1 = genWeibull(lambda[2], alpha)
    u_2 = genWeibull(lambda[3], alpha)
    x1 = min(u_0, u_1)
    x2 = min(u_0, u_2)
    if(x1 == x2) {
        next
    }
    X[i,] = c(x1, x2)
    i = i + 1
}

X
```

```
##              [,1]      [,2]
##    [1,] 0.4726397 0.4692683
##    [2,] 0.3010679 0.9768795
##    [3,] 1.4073465 0.9013180
##    [4,] 0.6368608 1.1551638
##    [5,] 0.7637175 0.7761144
##    [6,] 0.4433259 0.8561968
##    [7,] 0.6947942 0.7716213
##    [8,] 0.3900912 0.4491880
##    [9,] 0.7531553 0.7181241
##   [10,] 1.0904426 1.0688591
##   [11,] 0.5444482 0.6817445
##   [12,] 0.7862929 0.8680676
##   [13,] 0.8696646 1.1551976
##   [14,] 0.5341866 1.0629082
##   [15,] 0.5222446 0.4951530
##   [16,] 0.1813412 0.6833863
##   [17,] 0.4251942 0.6911030
##   [18,] 0.5283130 0.6433881
##   [19,] 0.8303231 0.8453176
##   [20,] 0.8508425 0.9815838
##   [21,] 0.9147454 1.4172201
##   [22,] 0.2999970 0.6179247
##   [23,] 0.9076358 1.1679432
##   [24,] 0.3348133 0.7299876
##   [25,] 0.2707065 0.4030949
##   [26,] 0.8737152 0.4652092
##   [27,] 1.0141351 0.9437535
##   [28,] 0.9071146 0.8379486
##   [29,] 0.1167770 0.7085718
##   [30,] 1.0442682 0.7156030
##   [31,] 0.9061755 1.1380205
##   [32,] 0.8052872 0.8176728
##   [33,] 0.4085028 0.5907606
##   [34,] 0.8901113 1.0264840
##   [35,] 1.1126195 0.8296021
##   [36,] 1.1732083 1.3722242
##   [37,] 0.7441256 0.7204840
```

```
##  [38,] 0.2094891 0.4746446
##  [39,] 0.3518113 0.9690834
##  [40,] 0.1753898 0.9009455
##  [41,] 0.1679618 1.2658004
##  [42,] 0.5059163 0.5733547
##  [43,] 0.3471213 0.9010695
##  [44,] 0.9186418 0.8644508
##  [45,] 0.7177780 0.9577449
##  [46,] 0.5360858 0.7308727
##  [47,] 0.7019440 0.7216478
##  [48,] 0.8234410 1.1578482
##  [49,] 0.8310722 0.8689524
##  [50,] 0.3967068 0.4695760
##  [51,] 1.1106350 0.9829099
##  [52,] 0.6114076 0.5767296
##  [53,] 0.1075524 1.4003699
##  [54,] 0.7606170 0.6353678
##  [55,] 0.1021357 0.6254384
##  [56,] 0.2041881 0.3083150
##  [57,] 0.4775905 0.8036995
##  [58,] 0.1789022 0.3634048
##  [59,] 0.1782202 0.9992379
##  [60,] 0.6057025 1.0620131
##  [61,] 1.1537883 0.9575717
##  [62,] 0.1088842 0.1769351
##  [63,] 0.8759420 1.2718542
##  [64,] 0.5586672 0.3965232
##  [65,] 0.3405828 1.1809191
##  [66,] 0.3722944 1.0396782
##  [67,] 0.7795772 1.0086191
##  [68,] 0.2903497 0.8777798
##  [69,] 0.8419007 1.0749510
##  [70,] 1.0889172 0.5291603
##  [71,] 0.1155084 0.9389237
##  [72,] 0.5190804 0.6321418
##  [73,] 1.0600846 0.5989394
##  [74,] 0.5089300 0.8813736
##  [75,] 1.0282478 0.7330704
##  [76,] 0.6130599 1.2135201
##  [77,] 0.2175158 0.8548843
##  [78,] 0.8390483 0.9078749
##  [79,] 0.4891769 0.5996401
##  [80,] 0.8572551 0.7025048
##  [81,] 0.1565169 0.5873108
##  [82,] 0.1796304 0.9298227
##  [83,] 0.2500905 1.4960874
##  [84,] 0.3208099 0.5342383
##  [85,] 0.2899484 0.3849096
##  [86,] 0.6276136 0.5920558
##  [87,] 0.3043504 0.4865439
##  [88,] 0.9396104 1.1438140
##  [89,] 0.4355675 0.7917542
##  [90,] 0.2965748 0.4929528
##  [91,] 0.2725490 0.9967585
```

```
##  [92,] 1.0670911 1.6868732
##  [93,] 1.1122417 0.8885766
##  [94,] 0.7220633 0.7358017
##  [95,] 0.6761444 0.9091799
##  [96,] 0.1783027 0.4936088
##  [97,] 0.3995286 0.6111690
##  [98,] 0.8480712 1.0899781
##  [99,] 0.5598902 0.9211651
## [100,] 0.8856401 1.0322035
```