STA 517 3.0 Programming and Statistical Computing with R

Generating Random Numbers Using the Inverse Transform Method

Prepared by Dr Thiyanga Talagala

1. Probability distribution functions in R to generate random numbers

rbeta	beta distribution	**rlnorm**	log-normal distribution
rbinom	binomial distribution	**rmultinom**	multinomial distribution
rcauchy	Cauchy distribution	**rnbinom**	negative binomial distribution
rchisq	chi-squared distribution	**rnorm**	normal distribution
rexp	exponential distribution	**rpois**	Poisson distribution
rf	F distribution	**rt**	Student's t distribution
rgamma	gamma distribution	**runif**	uniform distribution
rgeom	geometric distribution	**rweibull**	Weibull distribution
rhyper	hyper-geometric distribution		

There are other methods of generating random numbers from a particular distribution. In this lectorial we will discuss **Inverse Transform Method**.

2. Inverse transform method

Theorem 1: Probability Integral Transformation

> Let X have continuous cdf $F_X(x)$ and define the random variable Y as $Y = F_X(X)$. Then Y is uniformly distributed on (0, 1), that is, $P(Y \leq y) = y,\ 0 <$ y < 1.

Let's try to understand the theorem using an example.

Useful results to prove the theorem.

Result 1:

If F_X is strictly increasing, then F_X^{-1} is well defined by

$$F_X^{-1}(y) = x \Leftrightarrow F_X(x) = y.$$

If F_X is constant on some interval, then F_X^{-1} is not well defined by the above equation. To avoid this problem we define $F_X^{-1}(y)$ for $0 < y < 1$ by

$$F_X^{-1}(y) = inf\{x : F_X(x) \geq y\}.$$

Result 2:

If F_X is **strictly** increasing, then it is true that

$$F_X^{-1}(F_X(x)) = x.$$

Proof of Theorem 1:

For $Y = F_X(X)$ we have, for $0 < y < 1$, We can use Theorem 1 to generate random numbers from a particular distribution.

3. Steps in deriving random numbers using integral transformation method

1. Derive the cumulative distribution function of $f_X(x)$
2. Derive the inverse function $F_X^{-1}(u)$.
3. Write a function to generate random numbers.
    - Generate u from $Uniform(0, 1)$.
    - compute $x = F_X^{-1}(u)$.

Example 1

Write a function to generate n random numbers from the distribution with density $f_X(x) = 3x^2,\ 0 <$ x < 1.

Step 1: Find the cumulative distribution function of $f_X(x)$,

$$F_X(x) = x^3 \text{ for } 0 < x < 1$$

Step 2: Next we need to compute $F_X^{-1}(u)$,

$$F_X^{-1}(u) = u^{\frac{1}{3}}.$$

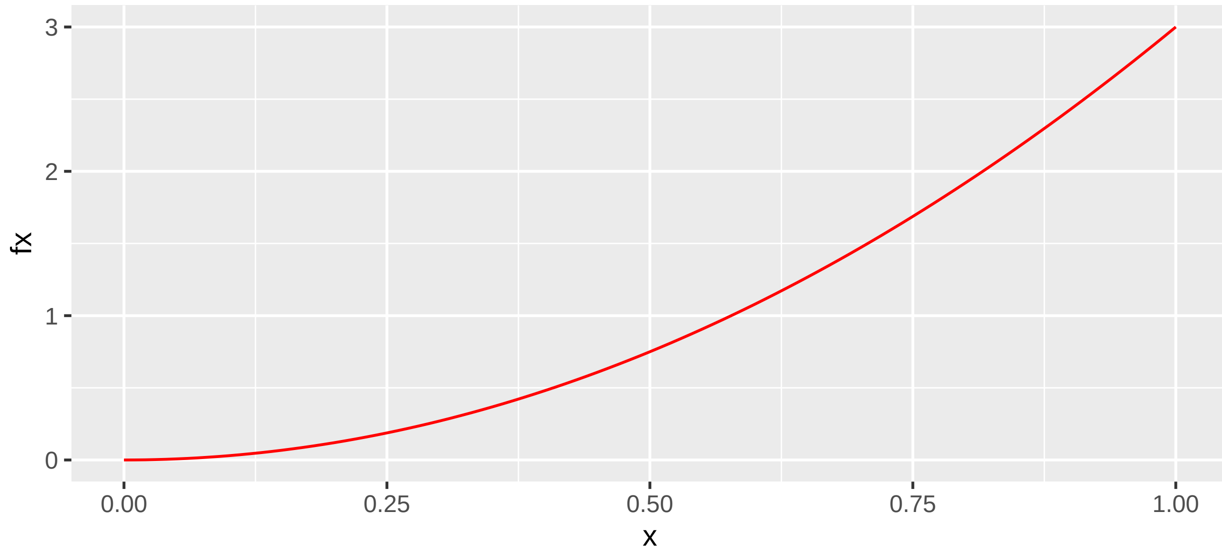
Step 3: R function

```
generate_it <- function(n){
  # Generate random numbers
  u <- runif(n)
  xgen <- u^(1/3)
  xgen

}

set.seed(2020)
generate_it(10)
```

```
 [1] 0.8648611 0.7332437 0.8520145 0.7812795 0.5143788 0.4069300 0.5054766
 [8] 0.7325562 0.1372012 0.8527963
```

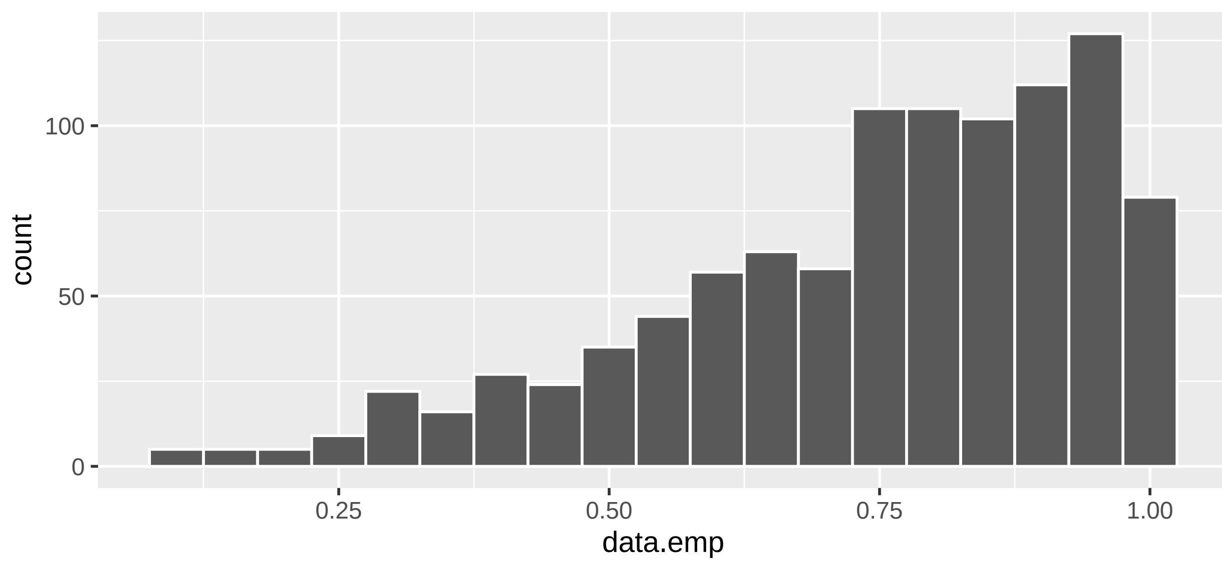
Visualisation of theoretical distribution

```r
library(tidyverse)
# Theoretical distribution values
theoretical.df <- tibble(x = seq(0, 1, 0.01), fx = 3*x^2)
ggplot(theoretical.df, aes(x = x, y = fx)) +
  geom_line(col = "red")
```



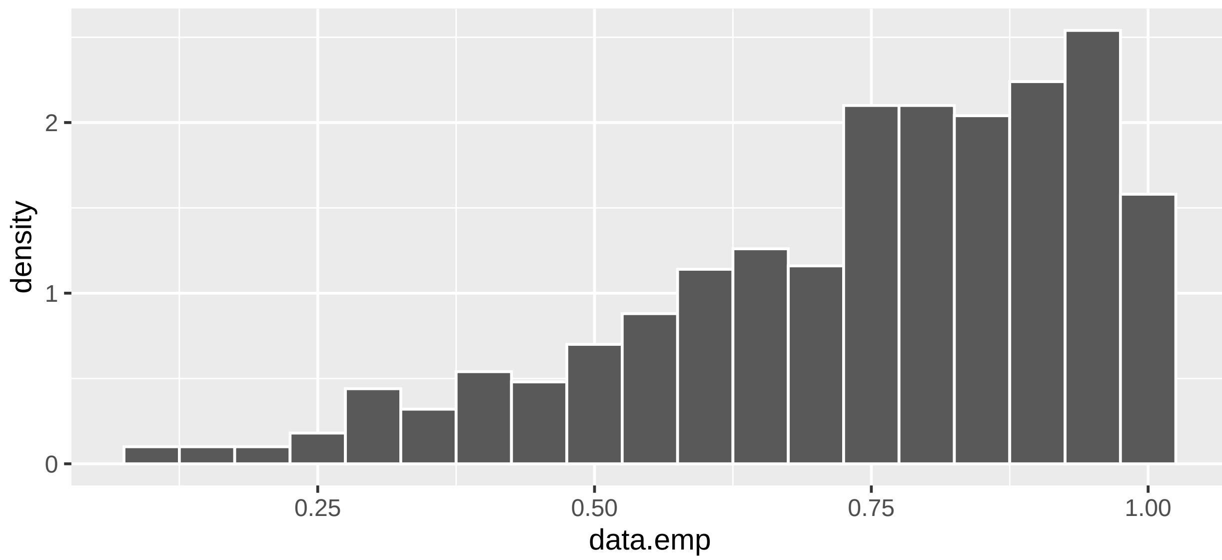
Visualize empirical distribution - counts

```r
empirical.df <- data.frame(data.emp = generate_it(1000))
# Plot empirical distribution - counts
ggplot(empirical.df, aes(x = data.emp))+
  geom_histogram(col = "white", binwidth = 0.05)
```



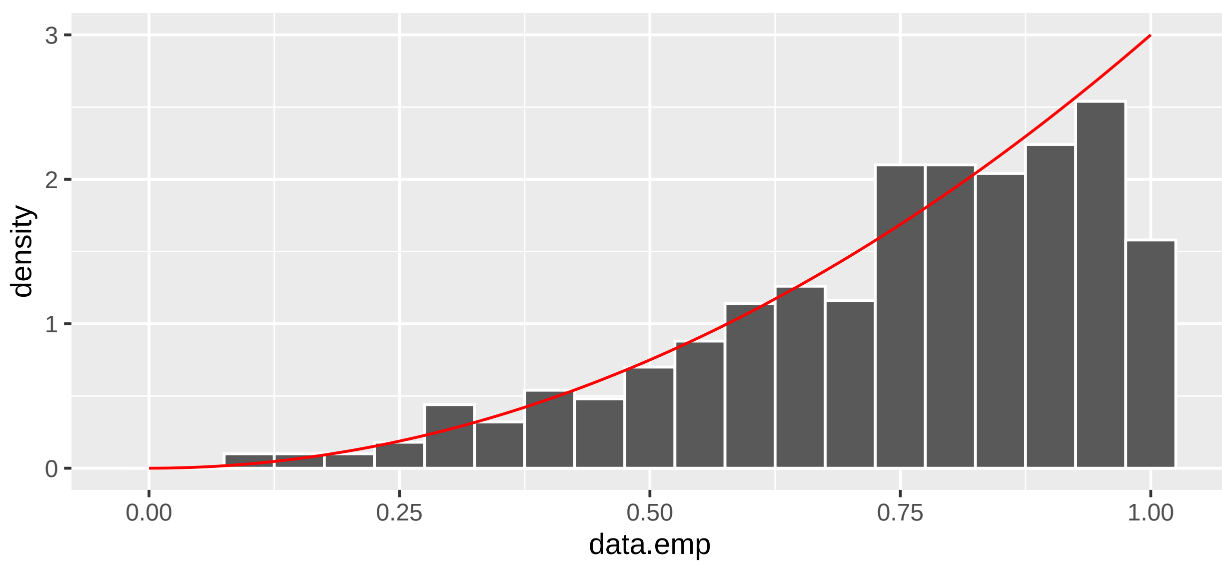
Visualize empirical distribution - density

```
ggplot(empirical.df, aes(x = data.emp, y=..density..)) +
  geom_histogram(col = "white", binwidth = 0.05)
```



Visualize theoretical distribution and empirical distribution together

```
ggplot(empirical.df, aes(x = data.emp, y=..density..)) +
  geom_histogram(col = "white", binwidth = 0.05) +
  geom_line(data = theoretical.df, aes(x = x, y = fx), color = 'red')
```



Function to generate random numbers and visualize theoretical and empirical distributions

```
generate_it_dist <- function(n){
  # Generate random numbers
  u <- runif(n)
  xgen <- u^(1/3)
  xgen

  # values for empirical distribution
  empirical.df <- data.frame(xgen=xgen)

  # values for the theoretical distribution
  theoretical.df <- tibble(x = seq(0, 1, 0.01),
  fx = 3*x^2)

  # arrange values and plot into a list
  list(
    xgen,
  ggplot2::ggplot(empirical.df, aes(x=xgen, y=..density..)) +
    geom_histogram(col="white", binwidth = 0.01) +
  geom_line(data = theoretical.df, aes(x = x, y = fx), color = 'red') )
}
```

Run the following codes and check the outputs.

```
# Sample size 10
generate_it_dist(10)

# Sample size 100
n100 <- generate_it_dist(100)
n100
n100[[1]]
n100[[2]]

# Sample size 10000
n10000 <- generate_it_dist(10000)
n10000[[2]]
```

Example 2

i) Write a function to generate random numbers from the $Exponential(\lambda)$ distribution using the inverse transformation method.

```
exp_generator <- function(n, lambda){
  u <- runif(n, 0, 1)
  exp.values <- -log(1-u)/lambda
    exp.values
}
```

ii) Generate 1000 random numbers from the $Exponential(2)$ distribution.

```
set.seed(111)
exp2 <- exp_generator(1000, 2)
```

```
exp2
```

```
   [1] 0.4494480553 0.6481923170 0.2313527655 0.3617246745 0.2371369396
   [6] 0.2709322984 0.0053575233 0.3799590208 0.2829583378 0.0491822557
  [11] 0.4057175747 0.4460777823 0.0347506843 0.0243577065 0.0849213804
  [16] 0.2956815133 0.0940352373 1.6986170219 0.1860149942 0.4765634595
  [21] 0.2819908410 0.1681050995 0.2093901924 0.2443915342 1.7136805265
  [26] 0.1943237064 0.5295459853 0.1665514095 0.7742370722 0.4530719863
  [31] 0.0301916961 0.3565727707 0.3134854035 0.3168348730 0.2229446628
  [36] 0.6248519440 0.0618275799 0.7662213541 0.5138077080 0.8176366795
  [41] 0.5124920900 0.1991144107 0.5048763728 1.3191629041 0.4281180826
  [46] 0.2283927700 0.2868891197 0.9687000140 0.4944247630 0.7893952961
  [51] 0.6457918773 0.4397922629 0.0166603330 0.2024786288 2.8594472510
  [56] 0.3973390414 0.4288139445 0.3046928120 0.0507830647 0.8187648467
  [61] 0.0004628645 0.3143768792 0.0951330393 0.1500275280 1.2583973452
  [66] 0.1319369315 0.0269988274 0.1814849470 0.0058975523 0.1788432265
  [71] 1.0501648791 0.5472285582 0.3023528451 0.0273983077 0.4983530585
  [76] 0.2918641253 0.1555411929 2.0605831023 0.0499787653 0.7708174958
  [81] 0.0638327052 1.0652220376 0.0702625052 0.2556944548 0.0452983603
  [86] 0.2348416796 0.5772517799 0.6635839965 0.7369025382 0.4337328436
  [91] 0.3577976163 0.9586059319 0.4968845142 0.4325682393 0.6740164567
  [96] 0.2448173423 2.5206341530 0.2538232422 1.8444604047 0.8700074109
 [101] 0.4290074451 0.0732160320 1.4044934287 0.0969651757 0.8565810671
 [106] 0.0060962020 0.5109121875 0.1553243470 0.4214013653 1.6805442859
 [111] 0.1105870857 0.5902151882 0.9040057558 1.1712055517 1.6695416425
 [116] 0.5360220796 0.3980472110 1.0815254546 0.2974419507 0.6799993819
 [121] 0.2674966744 0.5708989499 1.3016994060 0.5020121349 0.5379225944
 [126] 0.3256060745 0.8541606512 1.6613751574 1.2864945250 0.0897936051
 [131] 0.2165414801 0.4800161838 0.4339428221 1.4219796893 0.1179016774
 [136] 0.0198014955 0.4783784336 0.3614112044 1.2626892099 0.3038870211
 [141] 0.9387478801 1.0225768875 0.5744389193 0.7603551228 0.1447242287
 [146] 0.0237851724 0.4142061738 0.1738842137 0.2161346317 0.0412458555
 [151] 0.8723490678 0.1534079059 0.1531581624 0.0473647330 0.3088757439
 [156] 0.5881399868 0.9451066860 0.1412575999 0.5262286450 0.5700224105
 [161] 0.0552058207 0.0593837155 0.1214530069 0.0696979231 0.5488068422
 [166] 0.3344953458 0.2179392771 1.4688331779 0.0593629614 0.4149072017
```

```
[171] 0.9430095487 0.3584821694 0.5491260257 0.0134337158 0.0562823175
[176] 0.5944466267 0.1596515023 0.6831861136 0.6840216651 0.0912003538
[181] 1.9523828112 0.4516363033 0.5827270316 0.0942854124 0.0212884075
[186] 0.1052678645 0.7165291375 0.4098921479 0.3521185905 1.8023664994
[191] 0.0420174424 0.4073686130 1.1317166621 0.0570869055 0.2982655063
[196] 0.1846710854 0.1325577313 0.8803238097 0.0598866472 1.0770922299
[201] 0.6466209168 1.2649525835 0.0655998168 1.5094345695 0.5538816447
[206] 0.0444685030 0.4351293304 0.1194544972 0.1386806605 0.3842506719
[211] 0.0974362698 0.4224541193 0.0844548927 0.3216257886 0.6498862469
[216] 0.9363889605 1.5919527853 0.3070124627 0.2256697931 1.3773456091
[221] 1.0804216308 0.5552918079 0.9388514490 0.9818103877 0.3284514799
[226] 0.1906751591 1.2456886206 0.3512923455 0.0070859863 0.4526424119
[231] 0.2982369757 0.3873956674 0.0493137501 0.6950483696 0.0399543080
[236] 1.3227536915 0.8314696233 0.7163283050 0.5295126925 1.4076826411
[241] 0.7023614090 0.0439771601 2.3392259060 0.1573468755 0.0842268620
[246] 0.3310072538 0.4765626204 0.0420704260 1.4172045335 0.1135813979
[251] 0.1106130537 0.1089510500 1.1921948394 0.3810224434 0.5749561057
[256] 0.3884744509 0.4260525708 0.9377559168 0.4581201494 0.1316621190
[261] 2.6729312189 0.8122393841 0.9783161110 1.5730978219 0.0283612608
[266] 0.0352643443 0.3128394470 0.1954078688 0.5120262359 0.0053326795
[271] 0.1050152599 0.7392060798 0.0002224204 0.0086699266 0.1928794546
[276] 0.3768989672 0.5497496547 0.2486376976 0.1593630450 0.1699476025
[281] 0.6931198139 0.3654344441 0.6689157299 0.4155502260 0.1573197284
[286] 1.0367860336 0.5407143059 0.1359252457 0.8299708113 0.1340299005
[291] 0.3550856559 1.2713688186 1.6745181755 0.0686784503 0.1969659468
[296] 0.8187691122 0.2991119021 0.2252746105 0.7534404413 0.8255903593
[301] 0.3113812702 0.3277758319 0.0154585953 0.7779982258 0.1460414099
[306] 0.4760772037 0.4344576934 1.3502534396 0.0023742313 0.5324037484
[311] 0.3105623571 0.5025104639 0.2036632885 1.1155938778 0.5560154096
[316] 0.9764458225 0.0624724929 0.1494172755 0.6630364042 0.5011065669
[321] 0.5124838310 0.2200323820 0.1914512723 0.0737043721 0.8790949309
[326] 0.1761377119 0.8508821699 1.1214738097 0.5719970824 0.3019904318
[331] 0.5271257019 0.6474821217 0.6005618312 0.3997095449 0.8494689269
[336] 0.1573271022 1.5520141363 1.1856552630 0.1042857726 0.4179852695
[341] 0.8151686847 0.1473699477 0.4213314792 0.1754471126 0.2116585063
[346] 0.2081419292 0.4687823241 0.4179923164 0.0539433461 0.1969137150
[351] 0.3425678825 0.0571165170 0.6798288215 0.5928300498 0.3874651632
[356] 0.3811794579 0.8481393280 0.9213733050 0.1794583800 0.0451781725
[361] 0.0515263986 0.1714161961 0.3282122727 0.2223866873 0.1523906653
[366] 0.3268846691 0.2670731767 0.7067510493 0.0296874037 0.0254757461
[371] 0.2477197150 0.1859616143 0.6171138182 0.2036566545 0.1577784698
[376] 0.7039122537 0.0082608680 0.8864228410 0.7090177573 0.0498956211
[381] 0.9805755647 0.7463902851 0.8232595786 0.1761948033 0.1207479969
[386] 0.1555777472 0.3048354670 0.4178118243 0.0968499797 0.8555908293
[391] 0.2206220157 0.3552886064 0.1028676855 0.0715877529 0.2233441145
[396] 0.9226948986 0.2508971745 0.0109641399 0.2370466699 0.3893928606
[401] 0.2411066001 0.1003433014 0.1617661289 0.0776104371 0.7943216418
[406] 0.0523299749 0.0077253661 0.2984913734 0.0983202628 0.7123146196
[411] 0.5978020162 0.4556072580 0.4502315874 0.3965864425 0.1371009312
[416] 0.0035966022 0.7889436938 0.0021865926 0.3924203459 1.3170910945
[421] 0.1411126886 0.0283508195 1.4324882012 0.4843783301 0.5034289459
[426] 0.9918649156 1.5664772100 0.5426558842 0.3043280305 0.3016924841
[431] 1.4036028378 0.3402395900 0.6160949891 0.2983402750 1.2186695805
[436] 0.3654872857 0.2084101232 0.6914165491 0.3907548071 0.1565776748
```

```
[441] 0.4751765639 0.9585284100 0.2415713764 0.6724486566 0.2858984211
[446] 0.1067584849 0.0243299507 0.7116822811 0.1738142974 0.6397066556
[451] 0.3747977278 0.0989614304 0.1957032951 0.0942569307 0.0746125954
[456] 0.4236093079 0.1423738502 0.2869795047 0.2695215004 0.3134261546
[461] 0.5828967518 0.5811251210 0.7883411421 1.4401810503 0.3876445416
[466] 1.1866059747 2.5881243416 0.3755653034 0.8090908958 0.5296982486
[471] 0.2720856400 0.1841174150 0.4893550023 0.2810105449 0.7892706807
[476] 0.1888277981 0.5476160214 0.2469765008 0.8570582685 0.2375488271
[481] 0.8826013902 0.2887492700 0.3560744481 0.6904064531 1.6048045161
[486] 0.8189052187 0.7020302927 0.1592970375 0.4988569230 0.5247819133
[491] 0.2775136609 0.7944586487 0.3573352603 0.8121752599 0.1487447831
[496] 0.3761741920 0.1794286291 0.3964727706 0.6018876827 0.3396119608
[501] 0.0909978189 0.8146921756 0.8549000369 0.1697510534 0.3849943685
[506] 0.9220124585 0.1762281467 1.5298283042 0.0244725527 0.0653764922
[511] 0.1662469930 0.1046835449 0.7389156607 0.8042948879 0.7261205312
[516] 0.1533203963 0.1215746537 0.4042946662 0.0323499298 1.4755538905
[521] 0.7615946804 0.1223819299 0.1168096069 0.3037456240 0.1054697905
[526] 0.3022418574 0.3163890734 0.1833875465 1.0867304340 0.5152975686
[531] 0.3406634052 0.1875469543 0.0899669234 0.5338228123 0.5801834561
[536] 0.0008997833 0.5211149975 0.2611765568 0.2922972281 0.4935738785
[541] 1.1500334223 0.7313787666 2.7151948595 0.0544086538 0.7486238182
[546] 0.8267599730 0.9898229174 0.3984960643 0.2400217005 0.0764984483
[551] 2.7056016065 0.1237362325 0.7223767625 1.1979362072 0.1001095954
[556] 0.7020760561 0.9903994745 0.0440081222 0.5132905866 0.0096924902
[561] 0.5300679655 1.2813124486 0.0509609969 0.1053129678 1.6585853671
[566] 0.5557860868 0.1610401231 0.1551809200 1.2191152388 0.0173814572
[571] 0.0999859103 0.3223549499 0.4518860422 1.7847241268 0.5710922750
[576] 0.1979743897 0.2856439198 1.8905307264 0.2704016588 0.1508247605
[581] 0.1411911034 0.4166153388 0.4484024355 0.0718780297 0.3178493161
[586] 0.3494861087 0.1311165951 0.0252298827 0.9258803310 0.0083563878
[591] 0.1605946510 0.2310076854 0.1471046342 0.2831907462 0.1805541383
[596] 0.5081739321 0.2793988592 0.2923654352 0.3521549908 0.2006265501
[601] 0.4069873401 1.2099967793 0.5134875454 0.0151032659 0.2732954567
[606] 0.5763308263 0.0163408207 0.8871643222 0.2869690072 0.1542100524
[611] 0.6249935454 1.2435633875 0.6150765579 0.1182802791 0.3297545343
[616] 0.5021690620 0.0322948772 0.5810513089 0.8742875567 0.4247959664
[621] 1.1778343302 0.3871935165 0.0055166746 1.3176570628 0.4263622425
[626] 0.3043864188 0.1387868151 1.7513028974 0.3363346267 0.2837785748
[631] 0.8100557231 0.1708663275 0.2175919427 0.2507352817 0.5191032666
[636] 0.7731069821 0.6224673777 0.0072495902 0.3458453735 0.1277060042
[641] 0.4245016943 1.4738205714 0.2535877005 1.9839102544 0.1677466683
[646] 0.8119383692 0.0149811355 0.4842464601 0.3976305395 0.9373395663
[651] 0.0158200129 0.0376046684 0.2642005697 0.0251159884 0.9445578906
[656] 0.3303057408 0.4098682463 1.3412178163 0.0455102520 0.0882016616
[661] 0.5146733093 0.3778321062 0.4727082172 0.2889331855 0.2128679883
[666] 0.3174764972 0.1338430762 0.3643392659 0.2048576818 0.2622810761
[671] 0.4145946614 0.2204708240 1.4423101221 0.4325220704 1.3351094252
[676] 0.6071270184 0.3543290953 0.2928590079 0.1273454173 0.3167844139
[681] 0.1378958294 1.2717972078 0.2268985183 0.3798315310 0.7805418473
[686] 0.0153465979 0.2689642209 0.3553632647 0.4940580638 0.4963279622
[691] 0.7136288479 0.4528056255 0.1206223529 0.0140960467 0.5891486618
[696] 0.0589743080 0.1627686887 0.1323152545 0.3790266361 0.0729301501
[701] 0.0793637514 0.4907695948 0.4937099453 0.2407448790 0.0089044410
[706] 0.5644046588 0.0930219575 0.6573525145 0.1767928424 0.9886721580
```

```
[711] 0.7948330441 1.1439280967 0.1684187890 0.5421079253 0.5780021933
[716] 0.9774220916 0.2851747161 0.2892228353 0.6312714313 0.7502020719
[721] 0.0883742418 0.4389884509 0.5527720366 0.4932291861 0.2044816540
[726] 0.0520468809 0.1372848465 0.2817727136 0.7831370366 0.7537579786
[731] 0.3981232605 0.0160461633 0.1352940845 0.5649085611 0.2180862498
[736] 0.3468714195 0.4463747630 1.0316481394 1.2521337443 0.3868125441
[741] 0.4306968620 0.0963859585 3.1836998842 0.7888820706 0.0071254047
[746] 0.2364224821 0.6849942637 0.3256191033 2.1487620312 0.2127910473
[751] 0.0344194102 1.3413292614 0.2056105800 0.4042826218 1.0405655492
[756] 0.4681086505 0.1302965797 0.1920946909 0.9292223521 0.4128993377
[761] 0.4943309187 0.1192737328 1.7481128642 2.2374817618 0.5380991029
[766] 0.0195625748 0.5396651372 0.7845285731 0.2210534220 0.4464611526
[771] 0.5257219127 1.3118910966 0.8102774428 0.0521004060 0.5295532442
[776] 0.1616474974 0.5023942488 0.2377597016 0.7307559756 0.5759408351
[781] 1.8981603716 0.4329718049 0.6341890590 0.0409939855 0.2915800520
[786] 0.2357987818 0.1138665468 0.9075744776 0.0530431641 0.2693564013
[791] 1.2940950301 0.0918385899 0.1839224518 0.1084003588 0.0919375660
[796] 0.0139386055 0.4552003990 0.5123484725 1.0109201824 0.1982152319
[801] 0.2146773858 1.3927481330 0.0406668838 0.5226615322 0.3526326503
[806] 0.3222654215 0.1989670653 0.3643631244 0.0563185248 0.1873836435
[811] 0.3320352578 0.1456620660 3.0762563977 0.9041212573 0.2276540031
[816] 0.0767720307 0.1337496138 0.0395097996 0.4234967837 0.0261543121
[821] 0.0344818194 0.3595877153 0.1028211622 0.0911003670 0.9690313269
[826] 0.0701276547 0.1240882941 0.3245141703 0.2639112699 0.3594617648
[831] 0.9104509200 0.3966305341 0.4056783645 2.3767650110 1.0723368649
[836] 0.1017004770 0.0295534861 0.7224027014 0.0012667110 0.2859878343
[841] 0.2747146952 0.3437436047 0.4542905453 0.2679356757 1.0830436528
[846] 0.0710358143 0.7927198037 0.2130347294 0.1534539459 1.0016691918
[851] 0.2183737752 1.2648876571 0.1070563190 0.6568342262 0.0693494147
[856] 0.0403738065 0.0282663022 0.9246619855 0.9224301199 0.2673285622
[861] 0.4230395972 0.3863112656 1.3191898048 0.1827931173 0.1337903558
[866] 0.2399728140 0.5773537356 0.7684652716 0.9656549280 0.0086901110
[871] 0.4929685277 0.5115543526 0.3387641721 0.3243455675 0.0003366354
[876] 0.0873460534 0.3294451876 1.9813291253 0.4855167325 0.2194847088
[881] 0.0309405037 0.5108227110 1.0726075611 0.3609839423 0.3820540126
[886] 0.0313118433 0.6440729947 0.4936194847 0.3337511917 1.9182473393
[891] 0.3546399074 0.0558454057 0.1334837326 0.3272070487 0.0554143874
[896] 0.2619027786 0.7494482948 0.0435722317 0.1592125303 0.9956577034
[901] 0.0994810928 0.0209276025 0.3935775150 0.3088993330 0.0045931547
[906] 1.8014414468 0.0300223617 0.7485592023 1.1044315033 0.1872001220
[911] 0.7502565939 0.1292379213 0.1121858748 0.6480451755 1.2422944493
[916] 0.5263227825 0.0400438936 0.9509638634 1.4212956820 0.1315167581
[921] 0.0634819213 0.4358357494 0.1167388588 0.8162951581 0.1827574226
[926] 0.1884679064 0.2300597011 0.1915938577 0.1775284251 0.2604890123
[931] 0.8904090588 1.4111424328 0.0392753031 0.7719081246 0.5394257817
[936] 0.0204700094 0.0841252882 0.2588852408 0.2424310396 0.6588303785
[941] 0.6704474212 0.2601026891 0.0691107828 1.3109838368 0.3330782044
[946] 0.7885720662 0.1058949850 0.8321418651 0.7607748470 1.7737045349
[951] 0.4465230859 0.1716820444 0.5606338791 0.5152859792 0.2807607845
[956] 0.1737200527 0.6385847419 0.3569037022 0.1284534322 0.2673521512
[961] 0.5457178427 0.0827301559 0.0955156410 0.0522736159 0.0037987583
[966] 0.0581840906 0.4248020255 0.6932740339 1.8484423218 0.3279396084
[971] 0.1358753976 0.5404927012 0.0688954097 0.0426506986 0.4796767093
[976] 0.0158038913 0.0726931498 0.2047496195 0.0402370098 0.1343942374
```

```
[981] 0.8972664487 0.1523066821 0.3331043420 0.0349435730 0.3183395265
[986] 0.2613156916 0.0185045379 0.1067139110 0.8352169333 0.7749512387
[991] 1.3703322469 0.2839747783 0.5716211491 0.6238504269 0.6111759897
[996] 2.3721048818 0.2031257786 0.7555383871 0.2252286640 0.4334517398
```

iii) Graph the density histogram of the sample with the $Exponential(2)$ density superimposed for comparison.

```
# values for empirical distribution
empirical.df.exp <- data.frame(xgen=exp2)
# values for the theoretical distribution
theoretical.df <- tibble(x = seq(0.1, 5, 0.01),
  fx = 2*exp(-2*x))

ggplot(empirical.df.exp, aes(x = xgen, y=..density..)) +
  geom_histogram(col = "white", binwidth = 0.05) +
  geom_line(data = theoretical.df, aes(x = x, y = fx), color = "forestgreen") +
  labs(x="x", y="fx")
```