Approximate Normal Random Variable with Binomial Discrete Random Variable

Fan Wang

2022-07-23

Contents

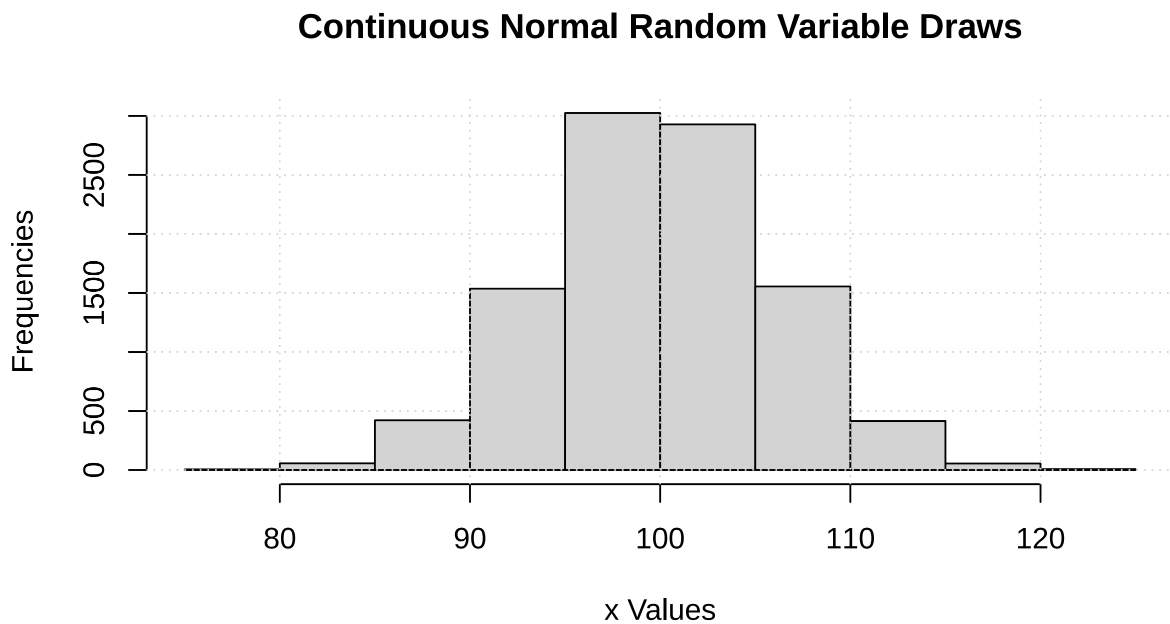
1 Discrete Approximation of Continuous Random Variables

Go to the **RMD**, **R**, **PDF**, or **HTML** version of this file. Go back to fan's REconTools research support package, R4Econ examples page, PkgTestR packaging guide, or Stat4Econ course page.

1.1 Use Binomial Discrete Random Variable to Approximate Continuous Normal

First, draw from a Continuous Random Variable. Sample N draws from a normal random variable.

```
# Random normal Data Vector (not equal outcomes)
set.seed(123)
it_sample_N <- 10000
fl_rnorm_mean <- 100
fl_rnorm_sd <- 6
ar_data_rnorm <- rnorm(it_sample_N, mean = fl_rnorm_mean, sd = fl_rnorm_sd)
# Visualize
par(new = FALSE)
hist(ar_data_rnorm, xlab = "x Values", ylab = "Frequencies", main = "")
title(main = "Continuous Normal Random Variable Draws")
grid()
```



We use the binomial to approximate the normal distribution. Let μ and σ be the mean and standard deviations of the normal random variable, and n and p be the number of "trials" and the "probability-of-success" for the binomial distribution. We know that these relationships are approximately true, :

$$\mu = n \cdot p$$
$$n = \frac{\mu}{p}$$
$$\sigma^2 = n \cdot p \cdot (1-p) = \mu \cdot (1-p)$$

Given these, we have can translate between the normal random variable's parameters and the binomial discrete random variable's parameters:

$$p = 1 - \frac{\sigma^2}{\mu}$$
$$n = \frac{\mu}{1 - \frac{\sigma^2}{\mu}} = \frac{\mu}{\frac{\mu - \sigma^2}{\mu}} = \frac{\mu^2}{\mu - \sigma^2}$$

There are two important aspects to note here:

1. Since p must be positive, this means $\frac{\sigma^2}{\mu} < 1$ and $\sigma^2 < \mu$, which is the condition for the above transformation to work.
2. The binomial discrete random variable will have non-zero mass for very small probability events at the left-tail. These very low outcome events are highly unlikely to be observed or drawn from sampling the continuous random variable. The presence of these left-tail values might impact the computation of certain statistics, for example the Atkinson Index for highly inequality averse planners.

Create a function for converting between normal and binomial parameters:

```r
ffi_binom_approx_nomr <- function(fl_rnorm_mean, fl_rnorm_sd) {
  #' @param fl_rnorm_mean float normal mean
  #' @param fl_rnorm_sd float normal standard deviation
```

```r
  if (fl_rnorm_mean <= fl_rnorm_sd^2) {
    stop("Normal mean must be larger than the variance for conversion")
  } else {
    # Use binomial to approximate normal
    fl_p_binom <- 1 - fl_rnorm_sd^2 / fl_rnorm_mean
    fl_n_binom <- round(fl_rnorm_mean^2 / (fl_rnorm_mean - fl_rnorm_sd^2))
    fl_binom_mean <- fl_n_binom * fl_p_binom
    fl_binom_sd <- sqrt(fl_n_binom * fl_p_binom * (1 - fl_p_binom))
    # return
    return(list(
      fl_p_binom = fl_p_binom, fl_n_binom = fl_n_binom,
      fl_binom_mean = fl_binom_mean, fl_binom_sd = fl_binom_sd
    ))
  }
}
```

Call the function to generate binomial parameters and generate the resulting binomial discrete random variable:

```r
# with these parameters, does not work
# ls_binom_params <- ffi_binom_approx_nomr(fl_rnorm_mean = 10, fl_rnorm_sd = 3)
# Call function with parameters, defined earlier
ls_binom_params <- ffi_binom_approx_nomr(fl_rnorm_mean, fl_rnorm_sd)
fl_binom_mean <- ls_binom_params$fl_binom_mean
fl_binom_sd <- ls_binom_params$fl_binom_sd
fl_n_binom <- ls_binom_params$fl_n_binom
fl_p_binom <- ls_binom_params$fl_p_binom
# Mean and sd, note that these are the same as values defined earlier
print(paste0("BINOMI mean=",
             ls_binom_params$fl_binom_mean,
             ", fl_rnorm_mean=",
             fl_rnorm_mean))
```

```
## [1] "BINOMI mean=99.84, fl_rnorm_mean=100"
```

```r
print(paste0("BINOMI sd=", ls_binom_params$fl_binom_sd,
             ", fl_binom_sd=", fl_binom_sd))
```

```
## [1] "BINOMI sd=5.99519807846246, fl_binom_sd=5.99519807846246"
```

```r
# drv = discrete random variable
ar_drv_rbinom_xval <- seq(1, fl_n_binom)
ar_drv_rbinom_prob <- dbinom(ar_drv_rbinom_xval,
  size = fl_n_binom, prob = fl_p_binom
)
# ignore weight at x=0
ar_drv_rbinom_prob <- ar_drv_rbinom_prob / sum(ar_drv_rbinom_prob)
```

Visualize the binomial discrete random variable:

```r
# graph
par(new = FALSE)
ar_ylim <- c(0, 1)
plot(ar_drv_rbinom_xval, ar_drv_rbinom_prob,
  xlab = "Binomial x Values", ylab = "Discrete Probabilitys, P(x)"
)
title(
```

```
  main = paste0("Binomial Approximate of Normal Random Variable"),
  sub = paste0(
    "binop=", round(fl_p_binom, 2),
    ";binon=", round(fl_n_binom, 2),
    ";binomean=", round(fl_binom_mean, 2),
    ";binomsd=", round(fl_binom_sd, 2),
    ";normmean=", round(fl_rnorm_mean, 2), ";normsd=", round(fl_rnorm_sd, 2)
  )
)
grid()
```