Project 1 Evaluate RNG

Boyi Guo

2/9/2021

Data Generating Process

We follow Headrick (2002) to simulate a Beta distribution ($\alpha = 4, \beta = 2$) from a normal distribution with mean 0 and standard deviation 1. Following the transformation equation, we can write a *standardized* Beta distributed variable Y_Z as a combination of polynomial terms of the normally distributed variable X,

$$Y_Z = c_0 + c_1X + c_2X^2 + c_3X^3 + c4X^4 + c_5X^5, \tag{1}$$

where X follows a normal distribution, N(0,1), $c_0 = 0.108304, c_1 = 1.104252, c_2 = -0.123347, c_3 = -0.045284, c_4 = 0.005014, c_5 = 0.001285$. We further scale and center the standardized beta distribution back to the original scale and range,

$$Y = Y_Z * \sigma_{\beta_{4,2}} + \mu_{\beta_{4,2}}, \tag{2}$$

where the mean of a Beta(4,2), $\mu_{\beta_{4,2}} \approx 0.67$ and the standard deviation of a Beta(4,2), $\sigma_{\beta_{4,2}} \approx 0.178$.

In each iteration of the simulation, we first simulate $X_i, i = 1, \ldots, 100,000$ independently follows a standard normal distribution, following with the aforementioned transformation (Equation (1),(2)) to generate Y_i. Sample mean, variance, skewness, and kurtosis are calculated for the 100,000 data points. In total, we have 1000 iterations of the described simulation.

The simulation is conducted on a 64-bit Windows 10 Platform machine with Inter i5 processor and 8 GB RAM. The simulation is implemented in R version 4.0.3 (2020-10-10).

Implementation in R

```r
# Note: the constants c have been loaded in the computation environment
hendrick_beta_4_2 <- function(x) {
  ret <- c0 + c1*x + c2*(x^2) + c3*(x^3) + c4*(x^4) + c5*(x^5)
  # return to the original scale and position
  ret*beta_sd(4,2) + beta_mean(4,2)
}

# wrapper function for each iteration in the simulation
sim_iteration <- function(
  it,
  n_sample,  # Sample Size
  func       # Transformation Function
){
  # Simulate X
  X <- rnorm(n_sample, mean = 0, sd = 1)
  Y <- func(X)
```

```r
  # return
  data.frame(
    it = it,
    mean = mean(Y),
    var = var(Y),
    skew = e1071::skewness(Y),
    kurt = e1071::kurtosis(Y, type = 1)
  )
}

# Simulation Body
set.seed(1)

sim_res <- purrr::map_dfr(1:n_it, .f = sim_iteration,
                          n_sample = n_sample, func = hendrick_beta_4_2)
```

Result

For Beta distribution with parameters (α, β), the moments can be calculated following the equations below:

$$
\begin{aligned}
\mu &= \frac{\alpha}{\alpha + \beta} \\
\sigma^2 &= \frac{\alpha * \beta}{(\alpha + \beta)^2 * (\alpha + \beta + 1)} \\
\text{skewness} &= \frac{2 * (\beta - \alpha) * \sqrt{(\alpha + \beta + 1)}}{(\alpha + \beta + 2) * \sqrt{(\alpha * \beta)}} \\
\text{excess kurtosis} &= \frac{6 * ((\alpha - \beta)^2 (\alpha + \beta + 1) - \alpha * \beta * (\alpha + \beta + 2))}{\alpha * \beta * (\alpha + \beta + 2) * (\alpha + \beta + 3)}.
\end{aligned}
$$

The expected and observed moments are presented in Table 1. The expected moments are calculated based on the above equations with $\alpha = 4, \beta = 2$; the observed moments are the averaged moments of simulated over 1000 iterations. The averaged observed moments match with the epxected moments closesly, up to 4 digits. Meanwhile, via Figure 1, we see the sample distributions of the moments follow a bell shape roughly. Central Limit effect exhibits, especially in variance and skewness.

Table 1: Expected and observed moments of Beta(4,2)

	Mean	Variance	Skewness	Excess Kurtosis
Expected	0.6666667	0.0317460	-0.4677072	-0.3750000
Observed	0.6666731	0.0317524	-0.4677340	-0.3751892

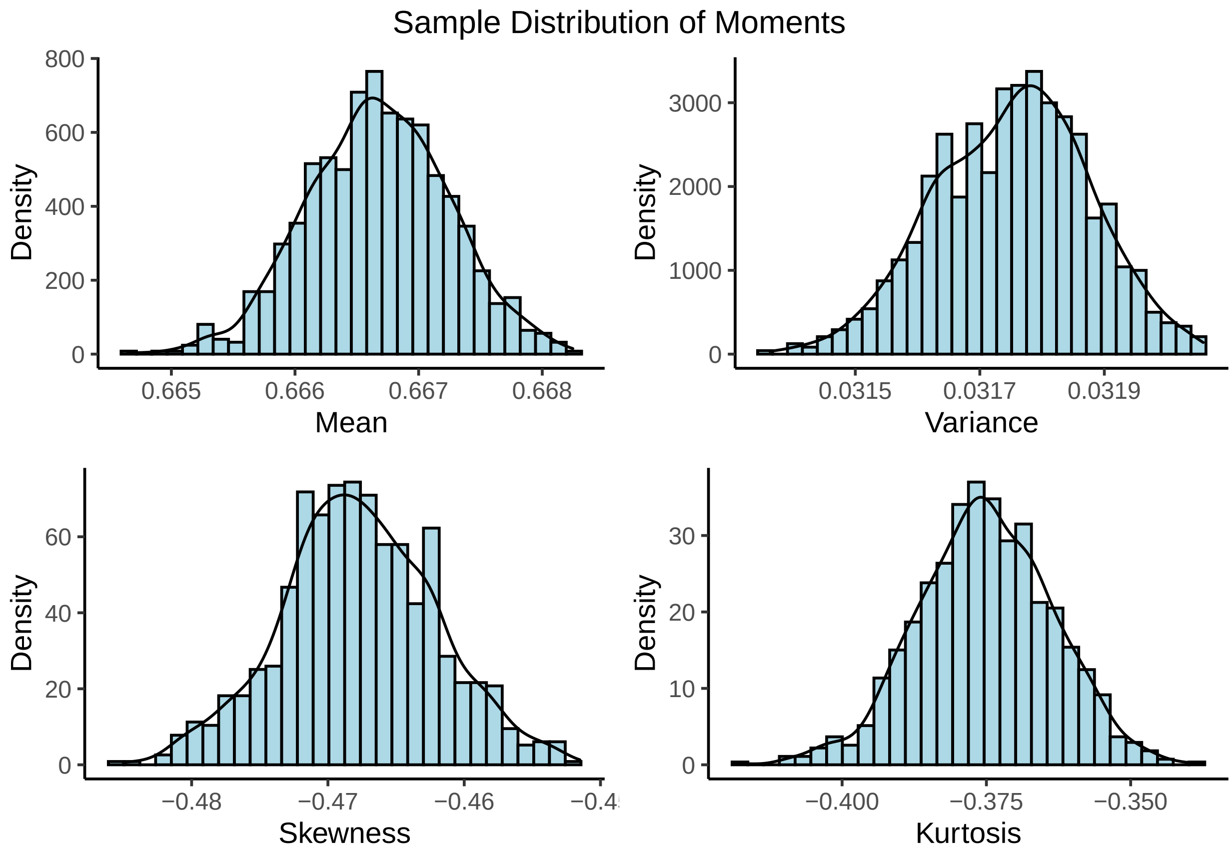
```r
plot_mean <- ggplot(sim_res) +
  geom_histogram(aes(x = mean, y=..density..),
                 color="black", fill="lightblue") +
  geom_density(aes(x = mean)) +
  xlab("Mean") +
  ylab("Density") +
  theme_classic()
```

```r
plot_var <- ggplot(sim_res) +
  geom_histogram(aes(x = var, y=..density..),
                 color="black", fill="lightblue") +
  geom_density(aes(x = var)) +
  xlab("Variance") +
  ylab("Density") +
  theme_classic()

plot_skew <- ggplot(sim_res) +
  geom_histogram(aes(x = skew, y=..density..),
                 color="black", fill="lightblue") +
  geom_density(aes(x = skew)) +
  xlab("Skewness") +
  ylab("Density") +
  theme_classic()

plot_kurt <- ggplot(sim_res) +
  geom_histogram(aes(x = kurt, y=..density..),
                 color="black", fill="lightblue") +
  geom_density(aes(x = kurt))  +
  xlab("Kurtosis") +
  ylab("Density") +
  theme_classic()

grid.arrange(plot_mean, plot_var, plot_skew, plot_kurt,
             ncol=2,
               top = textGrob("Sample Distribution of Moments"))
```

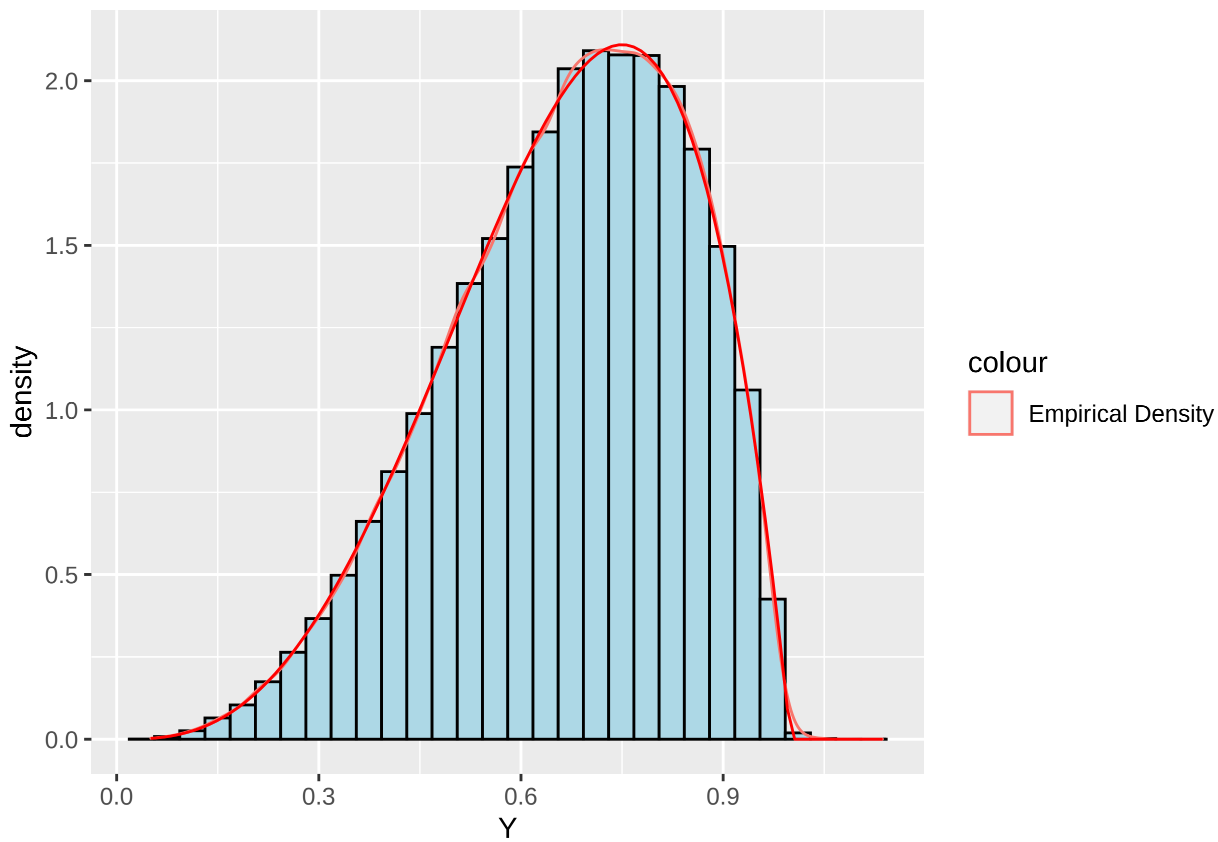


Goodness of fit

To evaluate the goodness of fit, we randomly choose one of the simulation iteration and plot its distribution

```r
set.seed(1)
set.seed(sample(1:100,1))
X <- rnorm(n_sample, mean = 0, sd = 1)
Y <- hendrick_beta_4_2(X)

ggplot() +
  geom_histogram(aes(x = Y, y=..density..),
                 color="black", fill="lightblue") +
  geom_density(aes(x = Y, color = "Empirical Density"))+
  geom_function(aes(color = "Expected Density"),
                fun = dbeta, args = list(shape1 = 4, shape2 = 2),
                color = "red")
```


density
2.0
1.5
1.0
0.5
0.0
0.0
0.3
0.6
0.9
Y
colour
Empirical Density

Reference

Headrick, Todd C. 2002. "Fast Fifth-Order Polynomial Transforms for Generating Univariate and Multivariate Nonnormal Distributions." *Computational Statistics & Data Analysis* 40 (4): 685–711.