C29 Continuous Multiple Random Variables

David Housman

## Concepts

1. The only differences between discrete and continuous random variables are that we work with probability *distribution* or *mass* functions and *sums* with *discrete* random variables while we work with probability *density* functions and *integrals* with *continuous* random variables. Suppose and are continuous random variables with the joint probability density function , and is a random variable that is a function of the random variables and . We define the expectation of by
* .
* Two special expectations are the mean and variance . Several results we previously proved for discrete random variables are also true for continuous random variables because of the linearity of integration: the integral of a sum is the sum of the integrals, and the integral of a constant multiple is the constant multiple of the integral.
1. Random number generators for dependent random variables can be developed using the marginal and conditional cdfs.
2. When measurements are combined in a formula, a Taylor series or simulation can be used to estimate the resulting value and possible error.

## Parabolic Example

Suppose and are continuous random variables with joint distribution

 for and .

#### Q1. Find .

f = function(x, y) 1-(x-y)^2
I = integrate(Vectorize(function(x) integrate(function(y) f(x,y), 0, 1)$value), 0, 1)$value
1 / I

f = function(x, y) 1-(x-y)^2
I = integral2(f, 0, 1, 0, 1, reltol=1e-8); I
1 / I$Q

f = function(x) 1-(x[1]-x[2])^2
I = adaptIntegrate(f, c(0,0), c(1,1)); I
1 / I$integral

#### Q2. Describe a 3D plot of the pdf and sketch an estimate of a contour plot of the pdf.

####

#### Q3. Obtain a 3D plot and contour plot of the pdf. Compare with your estimate.

f = function(x, y) 1.2\*(1-(x-y)^2)
x = seq(0, 1, length.out = 20)
y = seq(0, 1, length.out = 20)
z = outer(x, y, f)
plot\_ly() |> add\_surface(x = x, y = y, z = z)

plot\_ly() |> add\_contour(x = x, y = y, z = z)

f = function(x, y) 1.2\*(1-(x-y)^2)
parabolic\_pdf = tibble(
 x = rep(seq(0, 1, length.out = 20), 20),
 y = sort(x),
 z = f(x,y)
)
ggplot(parabolic\_pdf, aes(x = x, y = y, z = z)) +
 geom\_contour\_filled()

####

#### Q4. Estimate the marginal distribution graphs, means, standard deviations, and correlation coefficient.

####

#### Q5. Find the marginal distributions (and their graphs), means, standard deviations, and correlation coefficient. Compare with your estimates.

f = function(x, y) 1.2\*(1-(x-y)^2)
fX = Vectorize(function(x) integrate(function(y) f(x,y), 0, 1)$value)
fY = Vectorize(function(y) integrate(function(x) f(x,y), 0, 1)$value)
ggplot() + geom\_function(fun = fX, xlim = c(0,1))
tibble(
 muX = integrate(function(x) x\*fX(x), 0, 1)$value,
 muY = integrate(function(y) y\*fY(y), 0, 1)$value,
 sigmaX = sqrt(integrate(function(x) (x-muX)^2\*fX(x), 0, 1)$value),
 sigmaY = sqrt(integrate(function(y) (y-muY)^2\*fY(y), 0, 1)$value),
 covXY = integral2(function(x,y) (x-muX)\*(y-muY)\*f(x,y), 0, 1, 0, 1, abstol=1E-8)$Q,
 rho = covXY/(sigmaX\*sigmaY)
)

#### Q6. On the contour plot of the joint pdf, estimate the graphs of and .

#### Q7. Find the conditional distribution, mean, and standard deviation of given . Graph and on a contour plot of the joint pdf. Compare with your estimate.

fYlX = function(y, x) f(x,y)/fX(x)
muYlX = Vectorize(function(x) integrate(function(y) y\*fYlX(y,x), 0, 1)$value)
sigmaYlX = Vectorize(function(x) sqrt(integrate(function(y) (y-muYlX(x))^2\*fYlX(y,x), 0, 1)$value))
ggplot(parabolic\_pdf, aes(x = x, y = y, z = z)) +
 geom\_contour\_filled(show.legend = FALSE) +
 geom\_function(fun = muYlX) +
 geom\_function(fun = function(x) muYlX(x) + sigmaYlX(x)) +
 geom\_function(fun = function(x) muYlX(x) - sigmaYlX(x))

#### Q8. Are and independent? Explain.

####

#### Q9. Suppose . Find the mean and variance of .

f = function(x, y) 1.2\*(1-(x-y)^2)
integrand = function(x,y) (7\*x-3\*y)\*f(x,y)
muW = integral2(integrand, 0, 1, 0, 1, reltol=1e-8)$Q; muW
integrand = function(x,y) (7\*x-3\*y-muW)^2\*f(x,y)
varW = integral2(integrand, 0, 1, 0, 1, reltol=1e-8)$Q; varW

#### Q10. Suppose . Estimate a graph of the pdf of .

####

#### Q11. Suppose . Find and graph the pdf of . Compare with your estimates.

g = function(w) case\_when(
 w < -3 ~ 0,
 w < 0 ~ -158/46305\*w^3 - 6/1715\*w^2 + 122/1715\*w + 262/1715,
 w < 4 ~ -6/1715\*w^2 + 24/1715\*w + 262/1715,
 w < 7 ~ 158/46305\*w^3 - 2/45\*w^2 + 38/315\*w + 22/135,
 .default = 0)
ggplot() +
 geom\_function(fun = g, xlim = c(-3, 7)) +
 geom\_function(fun = dnorm, args = c(2, sqrt(3.74667)), xlim = c(-3, 7), color = "red")

####

#### Q12. Build a random number generator for and test how closely the pairs of numbers match the joint pdf.

f = function(x, y) 1.2\*(1-(x-y)^2)
#fX = Vectorize(function(x) integrate(function(y) f(x,y), 0, 1)$value)
fX = function(x) 0.8 + 1.2\*x - 1.2\*x^2
#FX = Vectorize(function(x) integrate(fX, 0, x)$value)
FX = function(x) 0.8\*x + 0.6\*x^2 - 0.4\*x^3
fYlX = function(y, x) f(x,y)/fX(x)
#FYlX = Vectorize(function(y, x) integrate(function(y) fYlX(y,x), 0, y)$value)
FYlX = function(y, x) (1.2\*y + 0.4\*(x-y)^3 - 0.4\*x^3) / fX(x)
rparb = function(n) {
 x = numeric(n)
 y = numeric(n)
 for (i in 1:n) {
 rn = runif(1)
 x[i] = uniroot(function(x) FX(x) - rn, c(0,1))$root
 rn = runif(1)
 y[i] = uniroot(function(y) FYlX(y,x[i]) - rn, c(0,1))$root
 }
 data.frame(x,y)
}
rparb(5)

# Simulate
runs = 10000
data = rparb(runs)

ggplot(data, aes(x = x, y = y)) +
 geom\_bin\_2d(breaks = seq(0, 1, 0.1)) +
 geom\_contour(data = parabolic\_pdf, mapping = aes(x = x, y = y, z = z), color = "red", linewidth = 2)

#### Q13. Suppose . Use our random number generator for to simulate , and test how closely the simulation results match the pdf for .

data = data |> mutate(w = 7\*x - 3\*y)
ggplot(data, aes(x = w, y = after\_stat(density))) +
 geom\_histogram(bins = 30) +
 geom\_function(aes(y = NULL), fun = g, xlim = c(-3, 7), color = "red")

#### Q14. Suppose . Find the mean and variance of .

f = function(x, y) 1.2\*(1-(x-y)^2)
muV = integral2(function(x,y) x\*y^2\*f(x,y), 0, 1, 0, 1)$Q; muV
varV = integral2(function(x,y) (x\*y^2-muV)^2\*f(x,y), 0, 1, 0, 1)$Q; varV

#### Q15. Suppose . Find the pdf of theoretically, approximate the pdf of via a simulation, and compare the two approaches.

data = data |> mutate(v = x\*y^2)
g = function(v) 0.96\*v^(-1/2) - 1.2 + 1.2\*v^(1/2) - 1.2\*v + 0.24\*v^2
ggplot(data, aes(x = v, y = after\_stat(density))) +
 geom\_histogram(bins = 30) +
 geom\_function(aes(y = NULL), fun = g, xlim = c(0, 1), color = "red")