project

Yaqiong Yao

11/24/2018

Numerical simulation

Univariate case

If we choose uniform kernel, the CDF of KDE should be represented as:

$$F_K(x|\mathcal{D}) = \frac{1}{n}\left(\sum_{i=1}^{n}\frac{x - x_i + b}{2b}I(x_i - b < x < x_i + b) + \sum_{i=1}^{n} I(x \geq x_i + b)\right).$$

If we set this equation to be u, which is probablity of $x \leq X$. We can hardly have the close form solution of x. Thus, if the support of x is not $\mathcal{R}$, then no close form can be obtained. Meanwhile, if we select more complicated kernels, such as normal distribution, the CDF of KDE becomes:

$$F_K(x|\mathcal{D}) = \frac{1}{n}\Phi(\frac{x - x_i}{b}).$$

Obviouly, no explicity closed form of x, either.

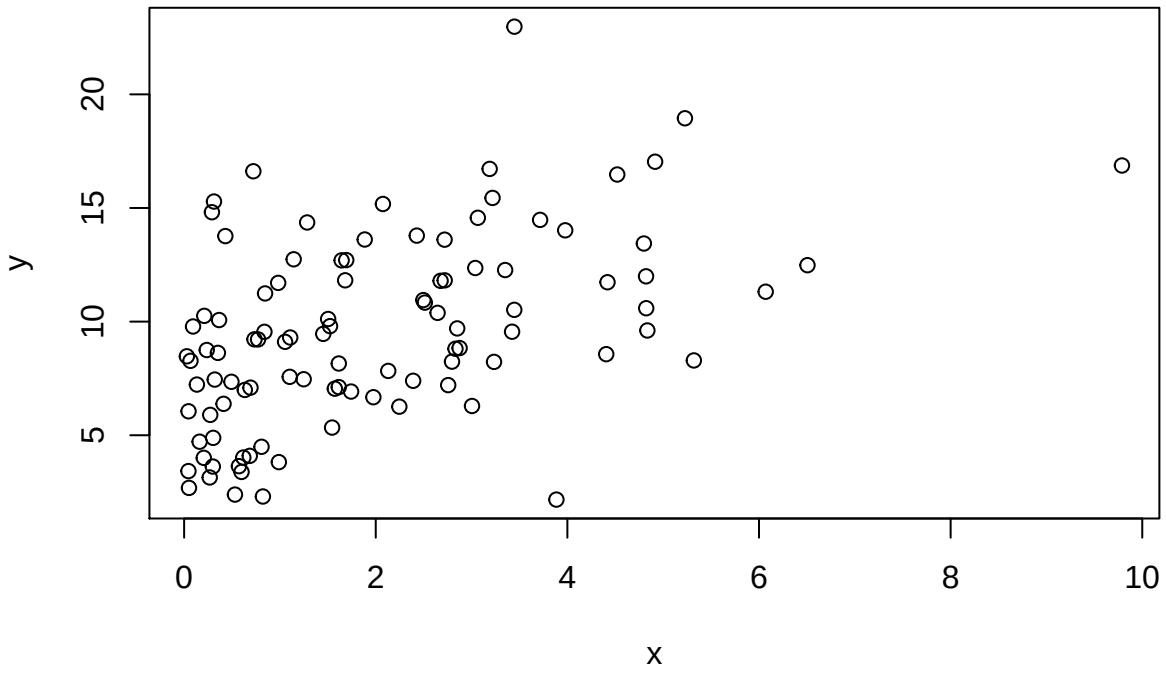
Bivariate case

Suppose we have a gaussian copula with two random variables, one is follow gamma distribution with parameter 1 and 2, and another one is from chi-squared distribution with degree of freedom 1.

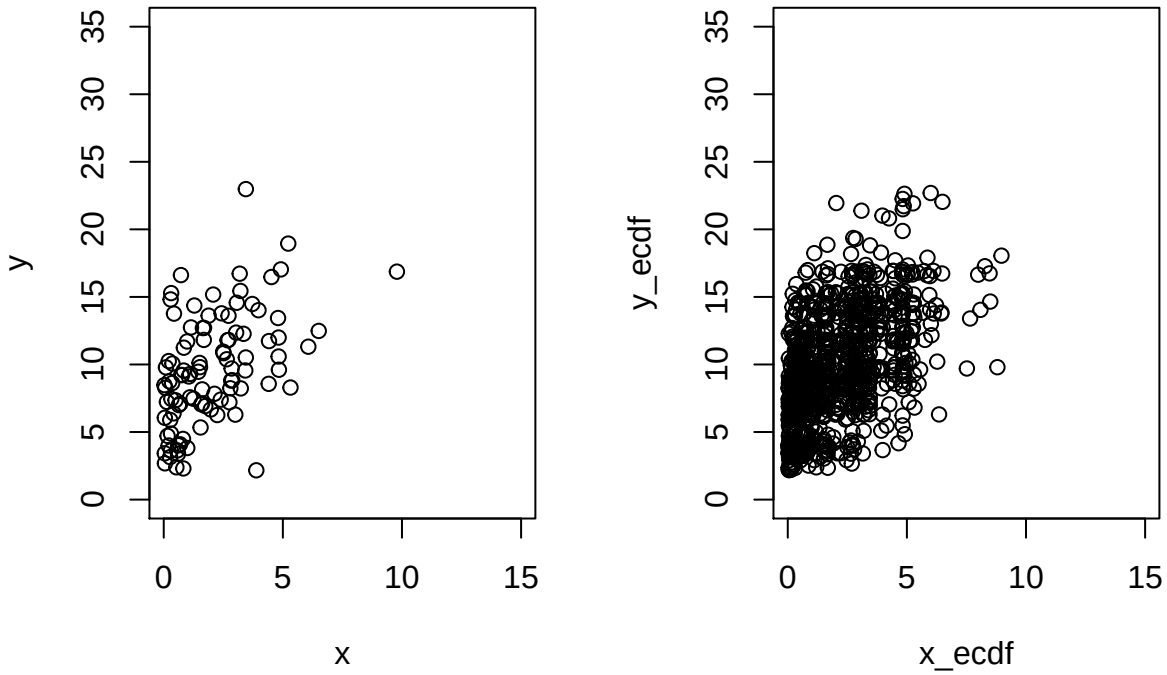
```
require(mvtnorm)
```

```
## Loading required package: mvtnorm
```

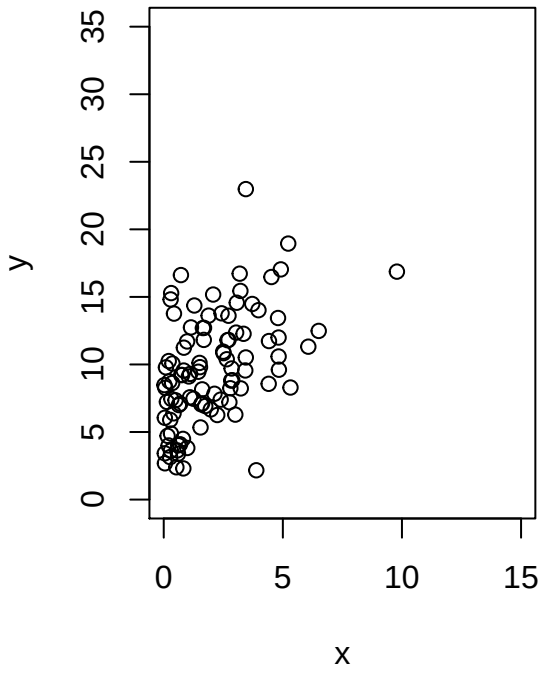
```
n <- 100
sigma <- matrix(c(1, 0.5, 0.5, 1), nrow = 2)
set.seed(098)
dat <- rmvnorm(n, sigma = sigma)
dat <- pnorm(dat)
x <- qgamma(dat[,1], shape = 1, scale = 2)
y <- qchisq(dat[,2], df = 10)
plot(x, y)
```

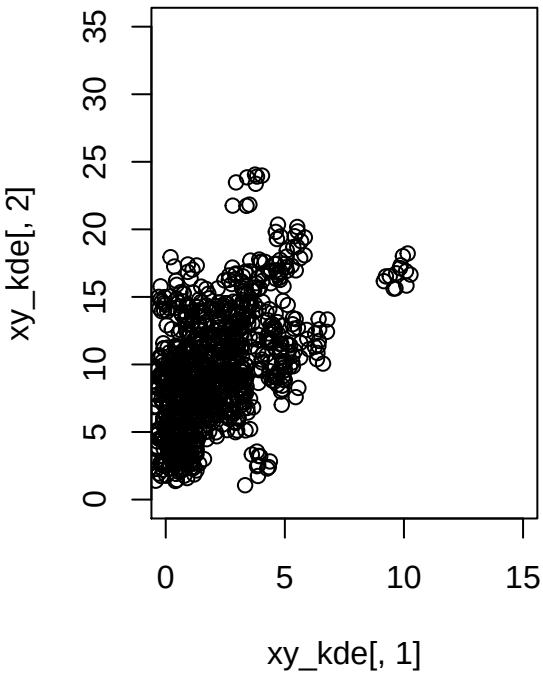



```
## generate random number with empirical cdf
n <- 1000
sigma <- matrix(c(1, 0.5, 0.5, 1), nrow = 2)
set.seed(456)
dat <- rmvnorm(n, sigma = sigma)
dat <- pnorm(dat)

x_ecdf <- quantile(x, dat[,1])
y_ecdf <- quantile(y, dat[,2])

par(mfrow = c(1,2))
plot(x, y, xlim = c(0, 15), ylim = c(0, 35))
plot(x_ecdf, y_ecdf, xlim = c(0, 15), ylim = c(0, 35))
```




```
## generate random number from KDE
## check the influence of bandwidth to the random number generating
sample.kernel <- function(n, x, y, adj){
  bw_x <- adj * density(x)$bw
  bw_y <- adj * density(y)$bw
  ind <- sample(1:100, n, replace = TRUE)
  x_kde <- x[ind] + runif(n, -bw_x, bw_x)
  y_kde <- y[ind] + runif(n, -bw_y, bw_y)
  cbind(x_kde, y_kde)
}
adj = 1
xy_kde <- sample.kernel(n, x, y, adj)

par(mfrow = c(1,2))
plot(x, y, xlim = c(0, 15), ylim = c(0, 35))
plot(xy_kde[,1], xy_kde[,2], xlim = c(0, 15), ylim = c(0, 35))
```


y
x
0
5
10
15
0
5
10
15
20
25
30
35


xy_kde[, 2]
xy_kde[, 1]
0
5
10
15
0
5
10
15
20
25
30
35