The distribution of the clayton family is

\[ \overline{G}(t_1, t_2) = \{ \overline{G}_1(t_1)^\theta + \overline{G}_2(t_2)^\theta - 1 \}^{\frac{1}{\theta}}, \]

where \( G_1(t_1) \sim U(0, 1) \) and \( G_2(t_2) \sim U(0, 1) \).

Since \( G_1(t_1) \sim U(0, 1) \) and \( G_2(t_2) \sim U(0, 1) \), we have

\[ \overline{G}_1(t_1) = 1 - G_1(t_1) \sim 1 - U(0, 1) = U(0, 1) \]

similarly,

\[ \overline{G}_2(t_2) \sim U(0, 1) \]

Here we can use Monte Carlo simulation by the method of Inverse transform to generate the data of clayton family. First, we have

\[ T_1 = \overline{G}_1(t_1) \sim U(0, 1) \]

so we can define

\[ T_1 \equiv U_1 \sim U(0, 1) \]

Then since \( T_1 \) and \( T_2 \) are dependent, we need to generate \( T_2 \) by \( \overline{G}_2(t_2 | T_1 = t_1) \).

By the method of Inverse transform we can obtain

\[ \overline{G}_2(T_2 | T_1 = t_1) \sim U(0, 1) \]

therefore,

\[ T_2 \equiv \{ (U_2 2^{\frac{\theta}{\theta+1}} + \frac{1}{2^\theta}) \}^{\frac{1}{\theta}}, \]

where \( U_2 \sim U(0, 1) \).

Hence we can generate data \( T_1 \) and \( T_2 \) in this way.
Now, we generate 1000 data and plot the scatter plot with different $\nu$ to observe the dependence between $T_1$ and $T_2$.

From the above figure, we can observe the following result:

if $-1 < \nu < 0$, $T_1$ and $T_2$ seem to have negative dependence. And the dependence is stronger as $\nu \to -1$.

if $\nu$ is very close to 0, $T_1$ and $T_2$ seem to be nearly independent.

(since $\nu$ cannot be 0, so $T_1$ and $T_2$ cannot be independent.)

if $\nu > 0$, $T_1$ and $T_2$ seem to have positive dependence. And the dependence is stronger as $\nu$ goes larger.
The result is the same as we inferred during class. So it is correct.

R code

cf_func=function(n,nu) {
    set.seed(10)
    u1=runif(n)
    u2=runif(n)
    t1=u1
    t2=((u2*t1^(nu+1))^(-nu/(nu+1))+1-t1^(-nu))^(-1/nu)
    data=cbind(t1,t2)
}

plot(cf_func(1000,-0.99),xlab=expression(t[1]),ylab=expression(t[2]),
     main=expression(paste(nu," = ",{nu})))
plot(cf_func(1000,-0.5),xlab=expression(t[1]),ylab=expression(t[2]),
     main=expression(paste(nu," = ",{nu})))
plot(cf_func(1000,-0.01),xlab=expression(t[1]),ylab=expression(t[2]),
     main=expression(paste(nu," = ",{nu})))
plot(cf_func(1000,0.01),xlab=expression(t[1]),ylab=expression(t[2]),
     main=expression(paste(nu," = ",{nu})))
plot(cf_func(1000,1),xlab=expression(t[1]),ylab=expression(t[2]),
     main=expression(paste(nu," = ",{nu})))
plot(cf_func(1000,10),xlab=expression(t[1]),ylab=expression(t[2]),
     main=expression(paste(nu," = ",{nu})))
plot(cf_func(1000,100),xlab=expression(t[1]),ylab=expression(t[2]),
     main=expression(paste(nu," = ",{nu})))