

Estudo de Cópulas

Guilherme Colombo Soares

Introdução

De acordo com Nelsen(1999), copulas podem ser definidas de duas formas: 1) funções que “acoplam” uma distribuição multivariada às suas funções de distribuição marginal unidimensionais. 2) Alternativamente, cópulas são funções de distribuição multivariada cujas marginais unidimensionais são uniformes no intervalo (0,1).

De acordo com Fisher (1997) “Cópulas são de interesse para os estatísticos por duas razões principais: em primeiro lugar, como um meio de estudar medidas de dependência; e em segundo lugar, como ponto de partida para a construção de famílias de distribuições bivariadas.”

Funções cópulas são representações flexíveis de distribuições conjuntas e fornecem especificações separadas de distribuições marginais e dependência e nos respondem questões como:

Suponha que temos duas variáveis, X e Y , onde Y é uma função não-linear de X mais uma variável aleatória. Além disso, temos uma terceira variável, Z , que é a soma de X e Y . A pergunta que surge é: qual é a distribuição de Z e como podemos descrever a dependência entre X e Z ?

Nesse estudo serão utilizadas copulas para estudar a dependência entre inflação medida por ipca e por igpm. No final, compara-se uma projeção feita por copulas (Vine-Copula) com uma feita por regressão linear.

```
library(readxl)
library(knitr)

data <- read_excel("C:/Users/guilh/Downloads/datacop2.xlsx")
rownames(data) <- c()
data$dolar <- c(NA, diff(data$dolar))
data$jureal <- c(NA, diff(data$jureal))
data <- na.omit(data)
kable(head(data))
```

ipca	igpm	dolar	jureal
0.13	0.35	-0.0332571	0.4416129
0.22	0.15	0.0261351	-0.5416129
0.42	0.23	0.0597634	-0.2000000
0.01	0.31	-0.0196032	0.0766667
0.23	0.85	-0.0105048	-1.4350538
1.61	1.57	0.0114153	-1.8316129

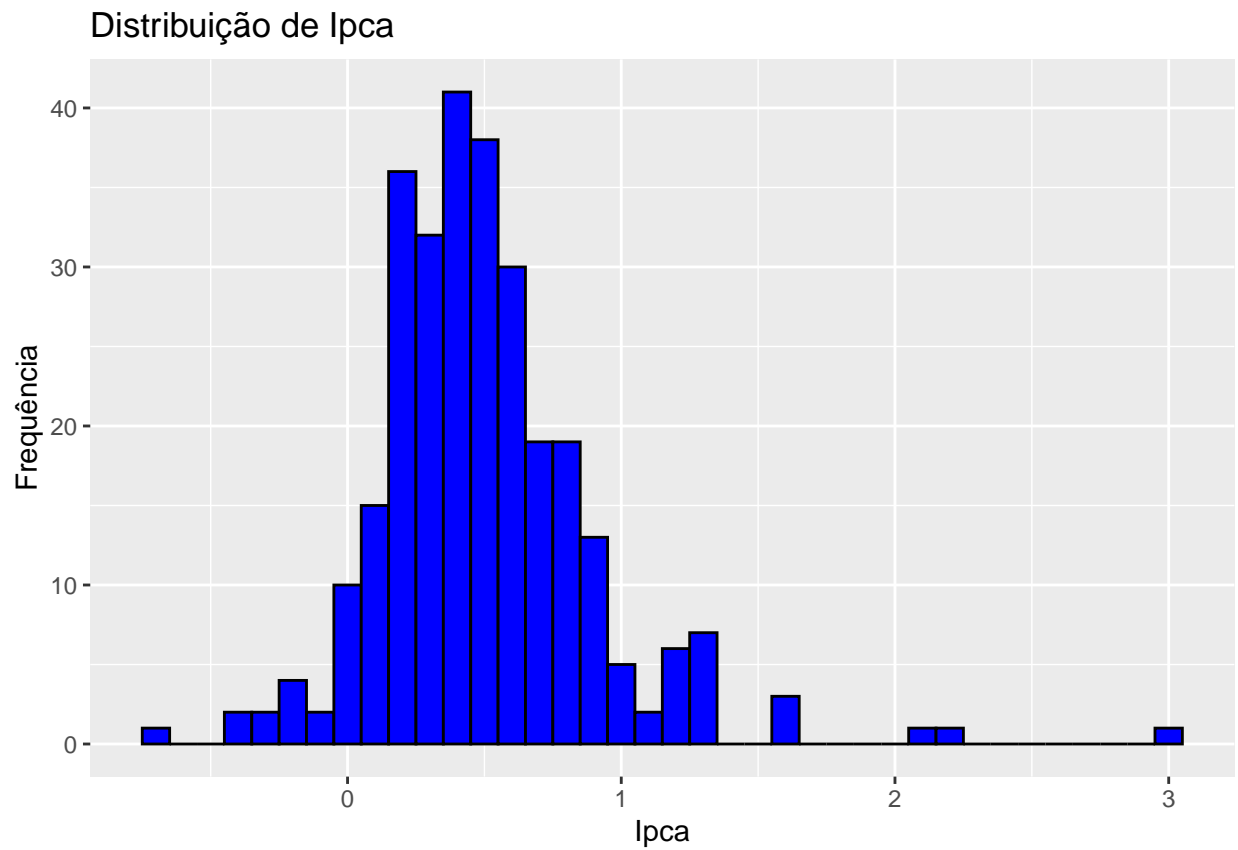
```
typeof(data$ipca)
```

```
## [1] "double"
```

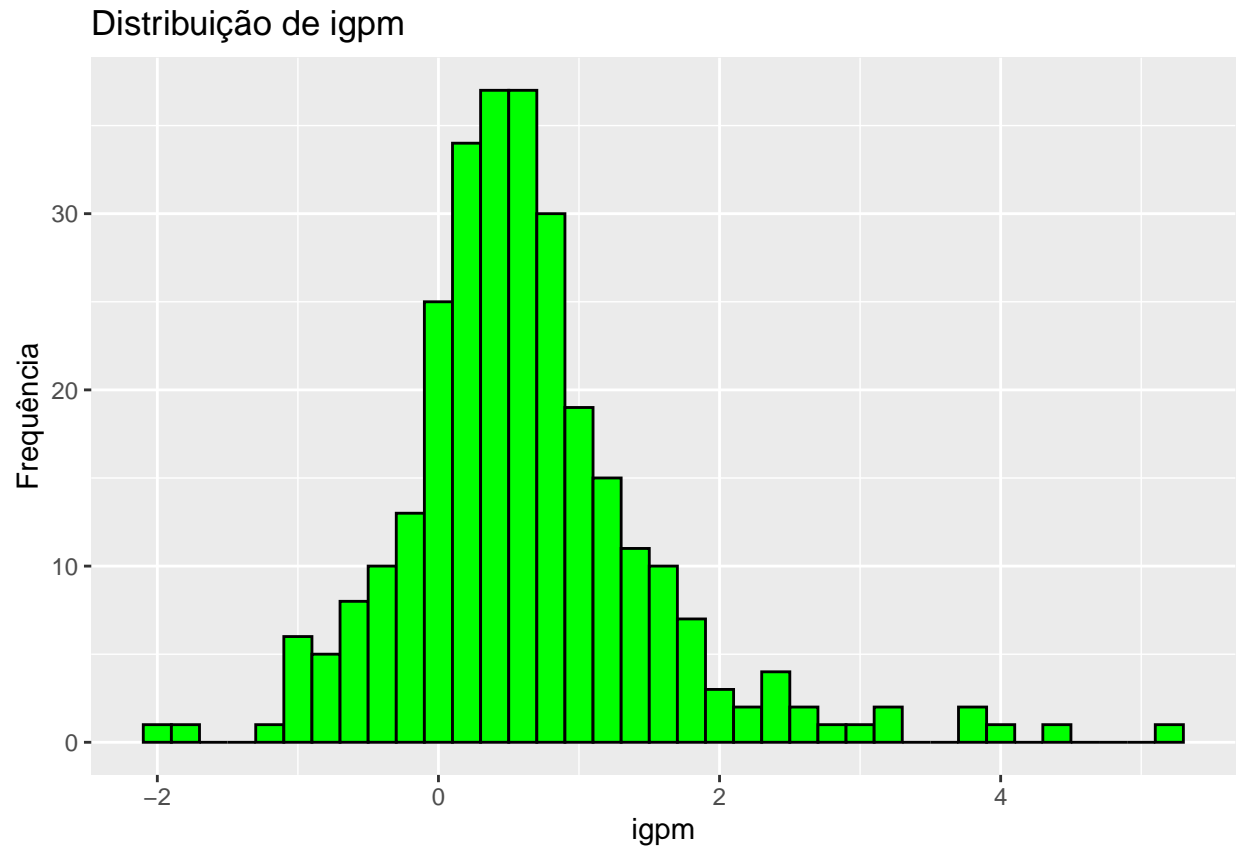
```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

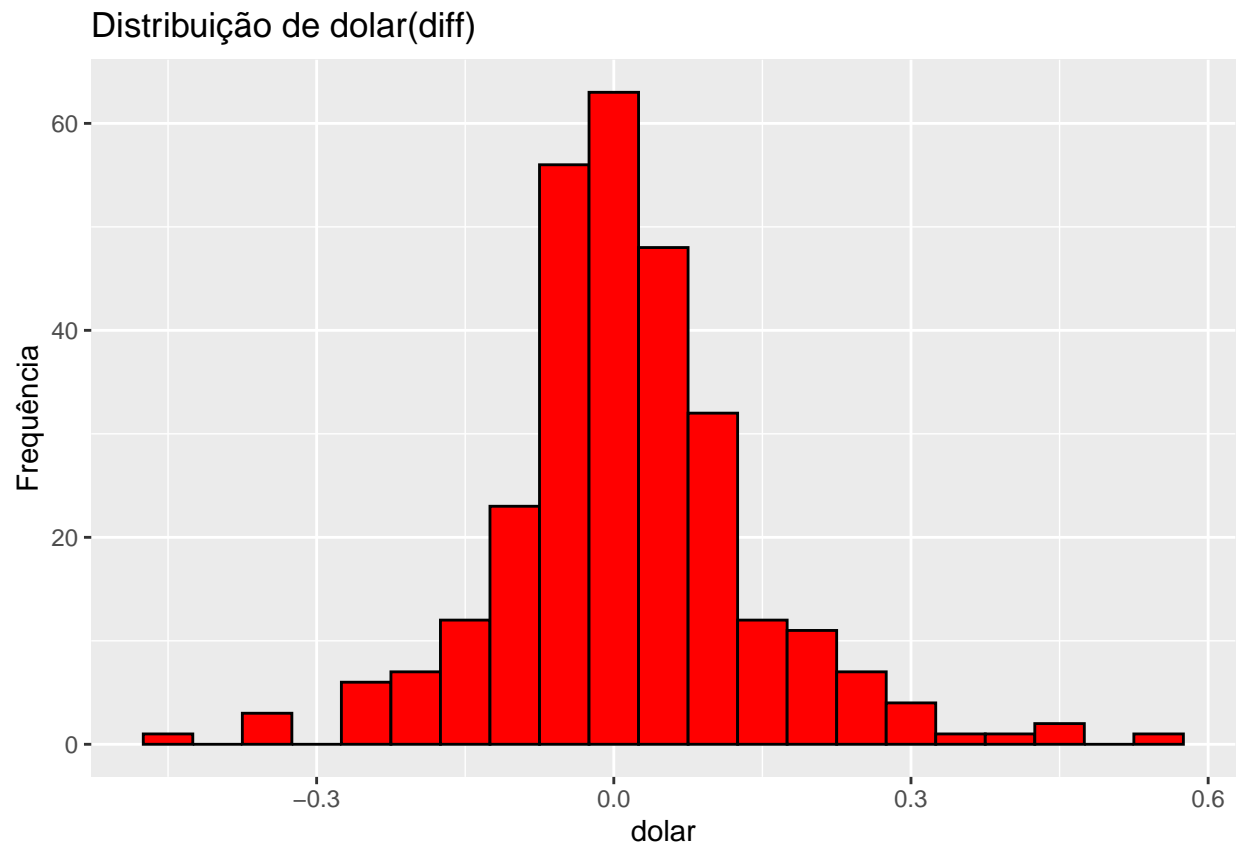
```
ggplot(data, aes(x = ipca)) +  
  geom_histogram(binwidth = 0.1, fill = "blue", color = "black") +  
  labs(title = "Distribuição de Ipca", x = "Ipca", y = "Frequência")
```



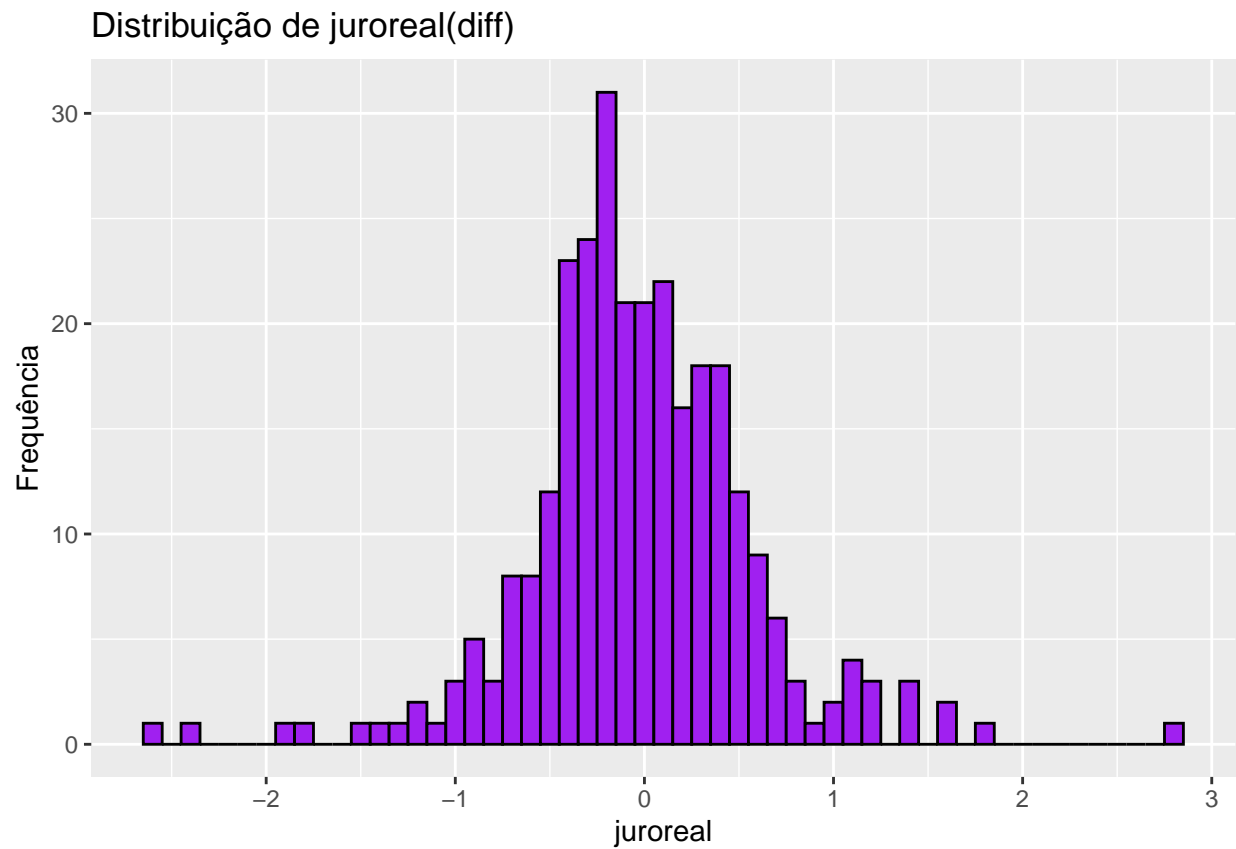
```
ggplot(data, aes(x = igpm)) +  
  geom_histogram(binwidth = 0.20, fill = "green", color = "black") +  
  labs(title = "Distribuição de igpm", x = "igpm", y = "Frequência")
```



```
ggplot(data, aes(x = dolar)) +  
  geom_histogram(binwidth = 0.05, fill = "red", color = "black") +  
  labs(title = "Distribuição de dolar(diff)", x = "dolar", y = "Frequência")
```

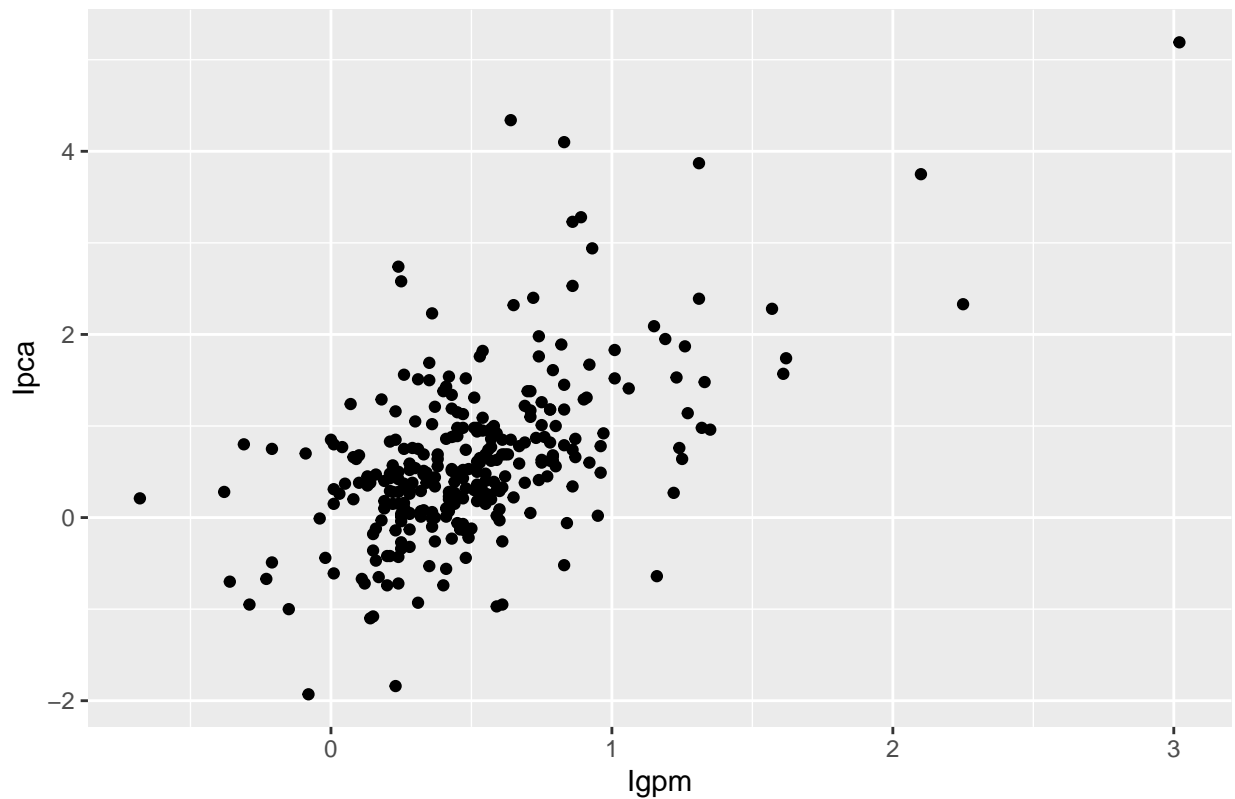


```
ggplot(data, aes(x = juoreal)) +  
  geom_histogram(binwidth = 0.1, fill = "purple", color = "black") +  
  labs(title = "Distribuição de juoreal(diff)", x = "juoreal", y = "Frequência")
```



```
ggplot(data, aes(x = ipca, y = igpm)) +  
  geom_point() +  
  labs(title = "Gráfico de Dispersão", x = "Igpm", y = "Ipca")
```

Gráfico de Dispersão



Medidas de dependência

As cópulas podem ser utilizadas para captar relações não lineares entre as variáveis, diferente do que é usado comumente em medidas de correlação como o coeficiente de person, para a avaliação das relações entre as variáveis, serão apresentados novas medidas de correlação com base na “concordância” e “discordância” (tradução própria de Nelsen, 1999).

De acordo com Nelsen(1999) ideia de concordancia e discordancia é relacionada a noção de se valores altos de uma variável estão relacionados com valores altos da outra e valores baixos relacionados com valores baixos. Ou seja, deixe (x_i, y_i) e (x_j, y_j) denotarem observações do vetor (X, Y) de variáveis contínuas aleatórias. Dizemos que são “concordantes” se $x_i < x_j$ e $y_i < y_j$ ou $x_i > x_j$ e $y_i > y_j$. E “discordantes” se $x_i > x_j$ e $y_i < y_j$ ou viceversa.

Correlação de Person

A correlação de pearson é uma abordagem linear entre duas variáveis, e é dada da seguinte forma:

$$r_{xy} = \frac{\sum (x - \bar{x})(y - \bar{y})}{\sqrt{\sum (x - \bar{x})^2 \sum (y - \bar{y})^2}}$$

onde \bar{x} e \bar{y} são as médias das variáveis. A correlação de Person então, capta relações lineares.

τ de Kendall

Deixe $\{(x_1, y_1), (x_2, y_2) \dots (x_n, y_n)\}$ denotar uma amostra aleatória de n observações do vetor (X, Y) de variáveis aleatórias contínuas. Tem-se então $\binom{n}{2}$ pares distintos (x_i, y_i) e (x_j, y_j) de observações da amostra. Cada par pode ser concordante(c) ou discordante(d). Assim, o τ de Kendall pode ser definido como:

$$\tau_{xy} = \frac{(c-d)}{(c+d)}$$

Portanto o τ é a probabilidade de concordância menos a probabilidade de discordância de um par (x_i, y_i) e (x_j, y_j) escolhidos aleatoriamente.

Outra formula usada para calcular o τ em sua versão populacional é:

$$\tau_{xy} = P[(X_i - X_{i+1})(Y_i - Y_{i+1}) > 0] - P[(X_i - X_{i+1})(Y_i - Y_{i+1}) < 0]$$

ρ de Spearman

Assim como no τ de Kendall, o ρ de Spearman também é uma medida de concordância e discordância. Deixe (x_1, y_1) , (x_2, y_2) e (x_3, y_3) serem 3 vetores aleatorios com a mesma função distribuição conjunta H (com as margens F e G) e uma cópula C . O ρ de Spearman será uma probabilidade de concordância e discordância para os vetores (x_1, y_1) e (x_2, y_3) . Ou seja, diferente do τ , um par de vetores terá a distribuição H e o outro é independente em seus componentes, o que torna a correlação de Spearman mais sensível a extremos que a de Kendall.

$$\rho_{xy} = 3(P[(X_1 - X_2)(Y_1 - Y_3) > 0] - P[(X_1 - X_2)(Y_1 - Y_3) < 0])$$

(O 3 que multiplica a função aparece na formula como uma constante de normalização, ver Nelsen 1999 p. 135)

```
pearson_corr <- cor(data$ipca, data$igpm, method = "pearson")
kendall_corr <- cor(data$ipca, data$igpm, method = "kendall")
spearman_corr <- cor(data$ipca, data$igpm, method = "spearman")

print(paste("Coeficiente de Correlação de Pearson:", pearson_corr))
```

```
## [1] "Coeficiente de Correlação de Pearson: 0.568966287004542"
```

```
print(paste("Coeficiente de Correlação de Kendall:", kendall_corr))
```

```
## [1] "Coeficiente de Correlação de Kendall: 0.35520285289869"
```

```
print(paste("Coeficiente de Correlação de Spearman:", spearman_corr))
```

```
## [1] "Coeficiente de Correlação de Spearman: 0.500288059111275"
```

Fundamentos teóricos das cópulas

Teorema de Sklar O teorema de Sklar é um teorema fundamental para a aplicação de cópulas. De acordo com Nelsen(1999) o teorema de Sklar elucida as relações entre as distribuições multivariadas e as distribuições marginais.

O **teorema de Sklar** nos diz que: Tome H como uma função de distribuição conjunta com margens F e G . Então existe uma cópula C , tal que para todo x, y em $\bar{R} = [-\infty, +\infty]$.

(1)

$$H(x, y) = C(F(x), G(Y))$$

A partir do teorema de Sklar, podemos “ligar” as funções marginais univariadas em uma função multivariada.

A formula (1) nos dá uma função de distribuição conjunta em função de duas funções de distribuição univariadas. (1) pode ser invertida para representar copulas em termos da função de distribuição conjunta e a inversa das duas margens, de forma que:

(2)

$$C(u, v) = H(F^{-1}(u), G^{-1}(v))$$

O resultado (2) nos permite construir copulas quando F e G são contínuas.

Limites de Fréchet-Hoeffding Outro resultado importante são os **limites de Fréchet-Hoeffding** que nos diz que para toda Copula C, para todo u,v em I:

$$W(u, v) = \max(u + v - 1, 0) \leq C(u, v) \leq M(u, v) = \min(u, v)$$

Os **limites de Fréchet-Hoeffding** nos permitem saber que:

i) Se X e Y são variáveis aleatórias com função de distribuição conjunta H . Então H é igual ao $W(F(x), G(y))$ se e somente se para todo (x, y) em \overline{R}^2 , ou $P[X > x, Y \leq y] = 0$, ou $P[X \leq x, Y > y] = 0$.

ii) H é igual ao $M(F(x), G(y))$ se e somente se $\min(P[X \leq x, Y > y], P[X > x, Y \leq y]) = 0$

```
data_normalized <- as.data.frame(scale(data))
head(data_normalized)
```

Modelando cópulas

```
##          ipca          igpm          dolar  juroreal
## 1 -0.9374376 -0.3095207 -0.353115950  0.7738593
## 2 -0.7123491 -0.5274136  0.114787082 -0.8503010
## 3 -0.2121524 -0.4402564  0.379717155 -0.2860012
## 4 -1.2375556 -0.3530993 -0.245548018  0.1710159
## 5 -0.6873392  0.2352117 -0.173868463 -2.3261484
## 6  2.7640179  1.0196262 -0.001177977 -2.9812121
```

```
cdf_ipca <- ecdf(data$ipca)

cdf_igpm <- ecdf(data$igpm)

ipca_values <- seq(min(data_normalized$ipca), max(data_normalized$ipca), length.out = 100)
igpm_values <- seq(min(data_normalized$igpm), max(data_normalized$igpm), length.out = 100)

cdf_ipca_values <- cdf_ipca(ipca_values)
cdf_igpm_values <- cdf_igpm(igpm_values)

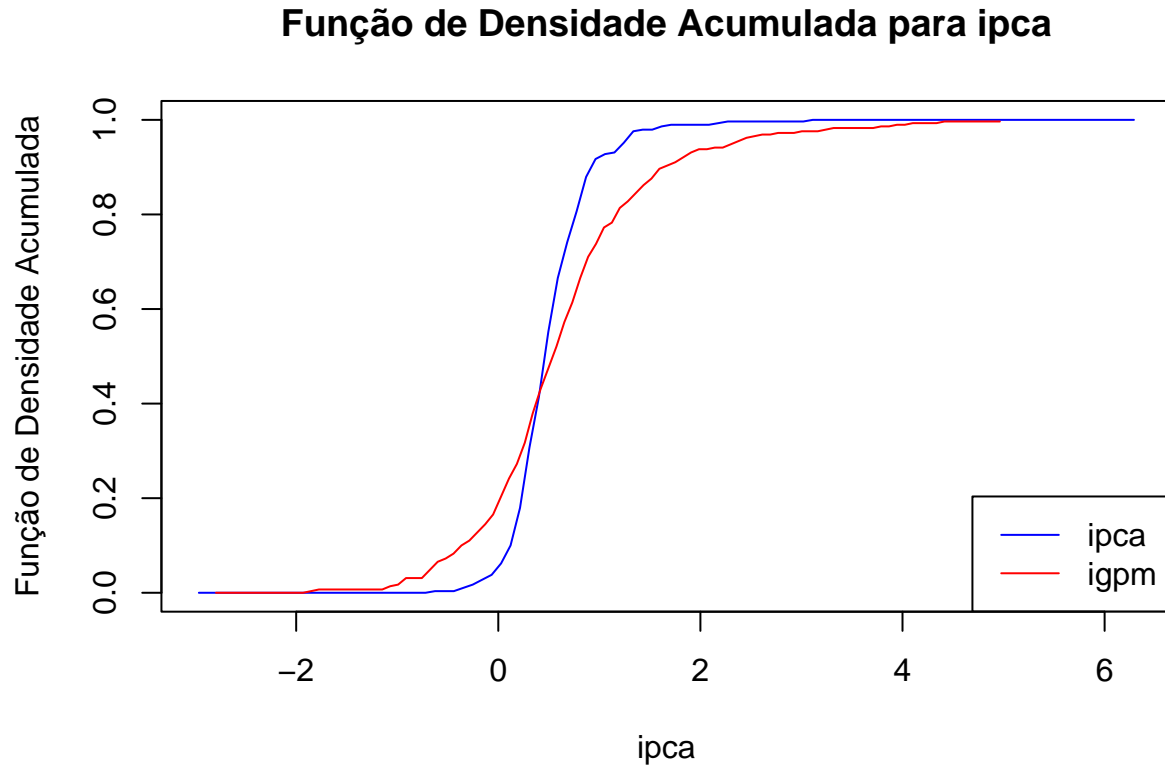
plot(ipca_values, cdf_ipca_values, type = "l", col = "blue",
```



```
xlab = "ipca", ylab = "Função de Densidade Acumulada",
main = "Função de Densidade Acumulada para ipca")
```

```
lines(igpm_values, cdf_igpm_values, type = "l", col = "red")
```

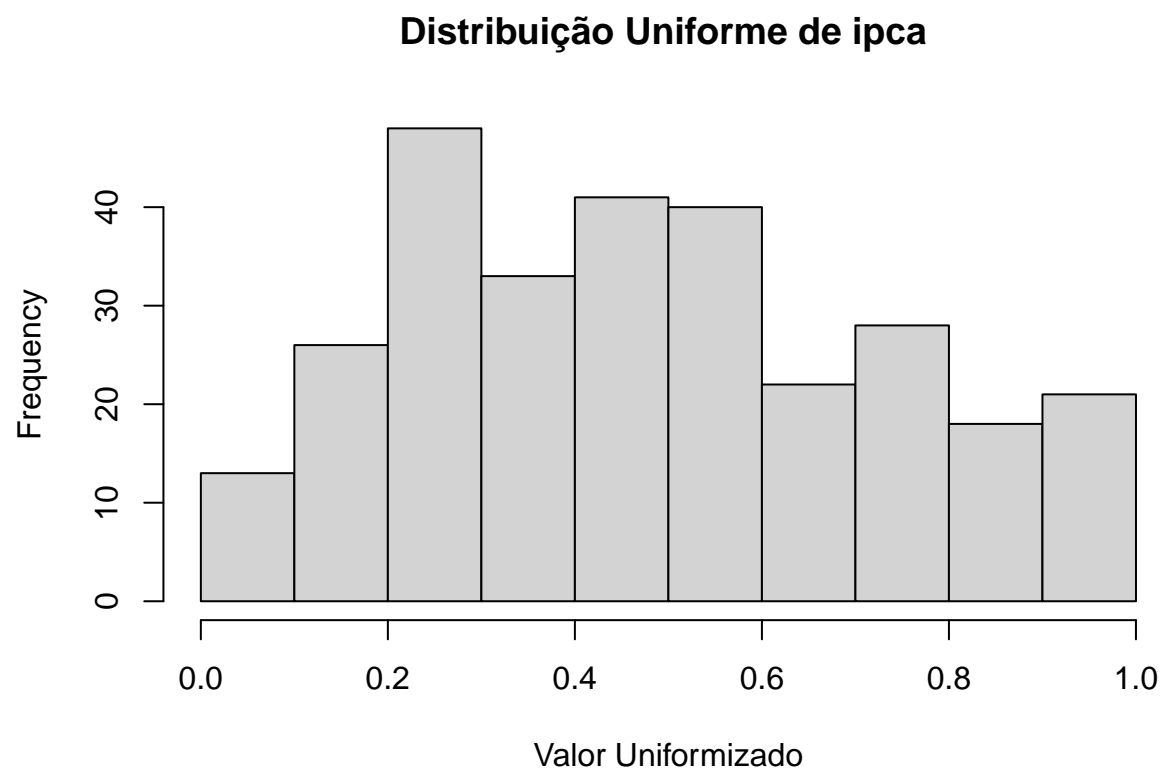
```
legend("bottomright", legend = c("ipca", "igpm"), col = c("blue", "red"), lty = 1)
```



```
ipca_uniform <- pnorm(data_normalized$ipca)
igpm_uniform <- pnorm(data_normalized$igpm)
```

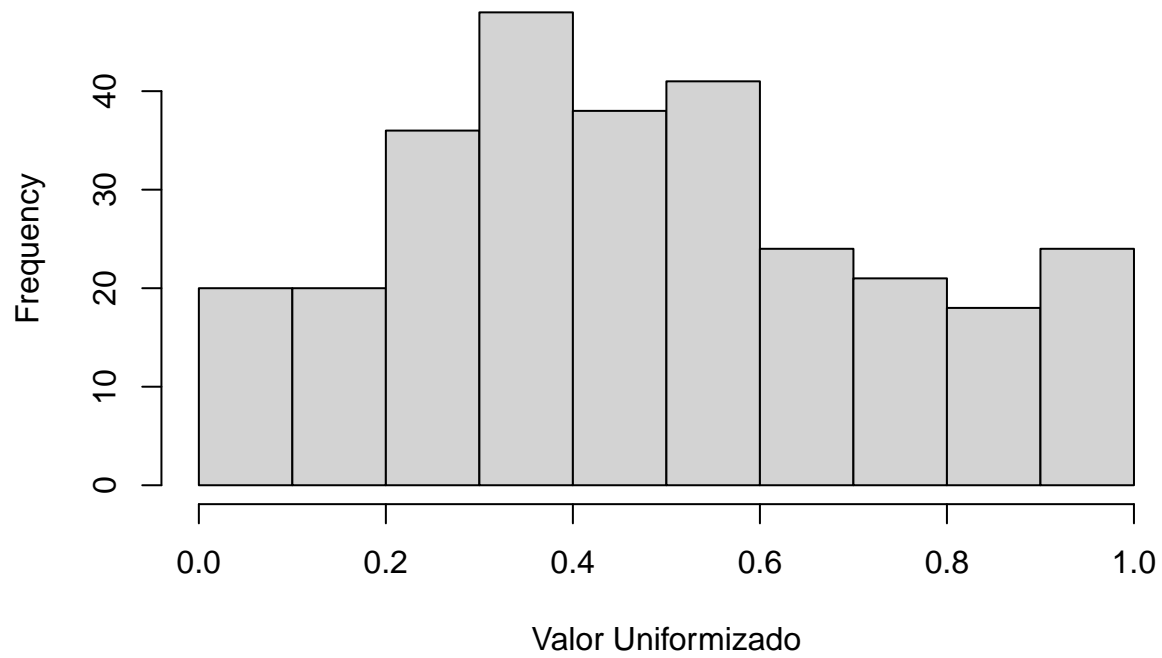
```
uniform_df <- data.frame(ipca_uniform , igpm_uniform)
```

```
hist(uniform_df$ipca_uniform, main = "Distribuição Uniforme de ipca ", xlab = "Valor Uniformizado")
```

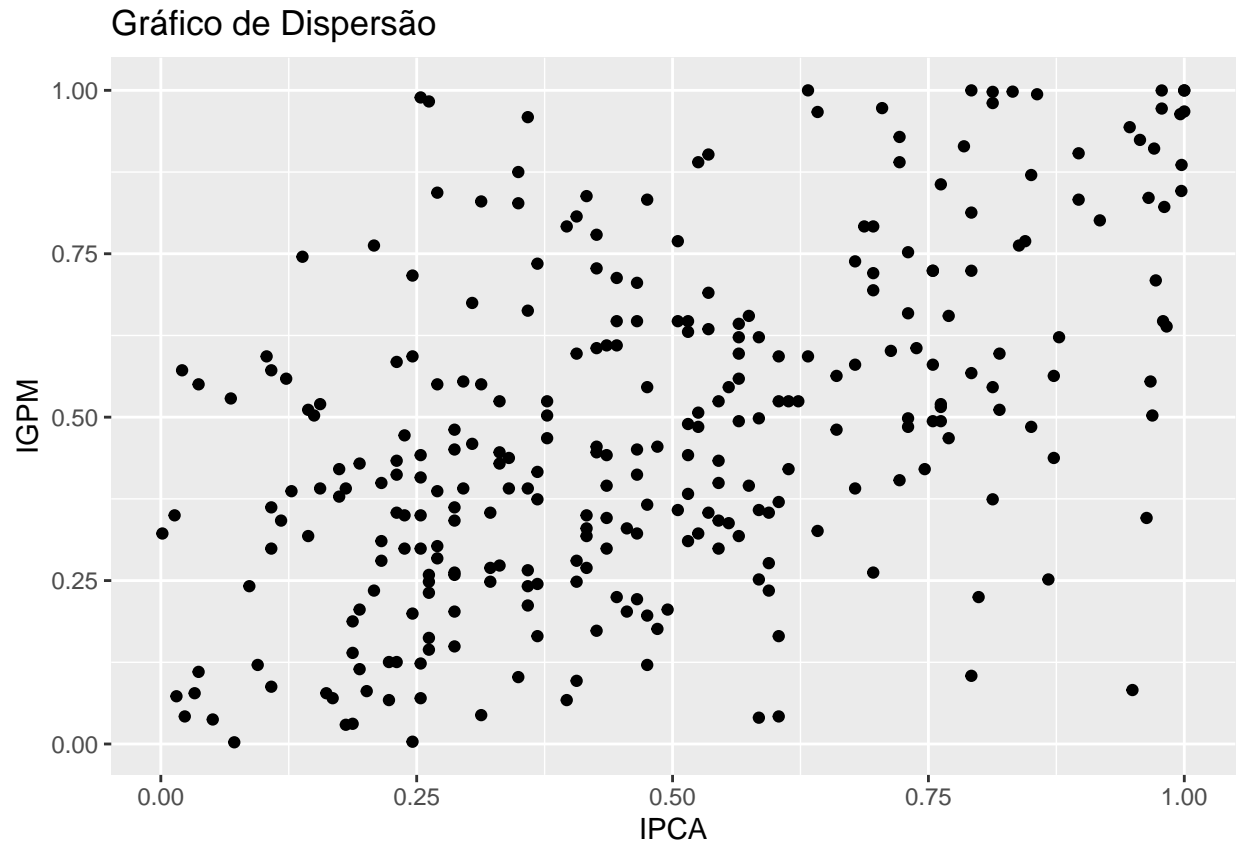


```
hist(uniform_df$igpm_uniform, main = "Distribuição Uniforme de igpm", xlab = "Valor Uniformizado")
```

Distribuição Uniforme de igpm



```
ggplot(uniform_df, aes(x = ipca_uniform, y = igpm_uniform)) +  
  geom_point() +  
  labs(title = "Gráfico de Dispersão", x = "IPCA", y = "IGPM")
```



Copula Normal

$$C(u_1, \dots, u_n) = \Phi_P(\Phi^{-1}(u_1), \dots, \Phi^{-1}(u_n))$$

$$c(u_1, \dots, u_n) = \frac{1}{\sqrt{\det(\mathbf{P})}} \exp \left\{ -\frac{1}{2} [\Phi^{-1}(u_1) \dots \Phi^{-1}(u_n)] (\mathbf{P}^{-1} - \mathbf{I}_{n \times n}) \begin{bmatrix} \Phi^{-1}(u_1) \\ \vdots \\ \Phi^{-1}(u_n) \end{bmatrix} \right\}$$

```
library(copula)
```

```
## Warning: package 'copula' was built under R version 4.3.3
```

```
gaussian_copula <- normalCopula(dim = 2)
gaussian_copula_fit <- fitCopula(gaussian_copula, uniform_df)
```

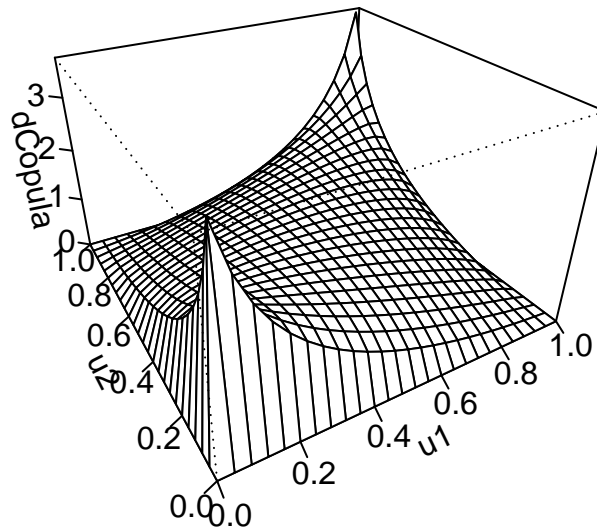
```
## Warning in .local(copula, data, ...): coercing 'data' to a matrix.
```

```
print(gaussian_copula_fit)
```

```
## Call: fitCopula(gaussian_copula, data = uniform_df)
## Fit based on "maximum pseudo-likelihood" and 290 2-dimensional observations.
## Copula: normalCopula
```

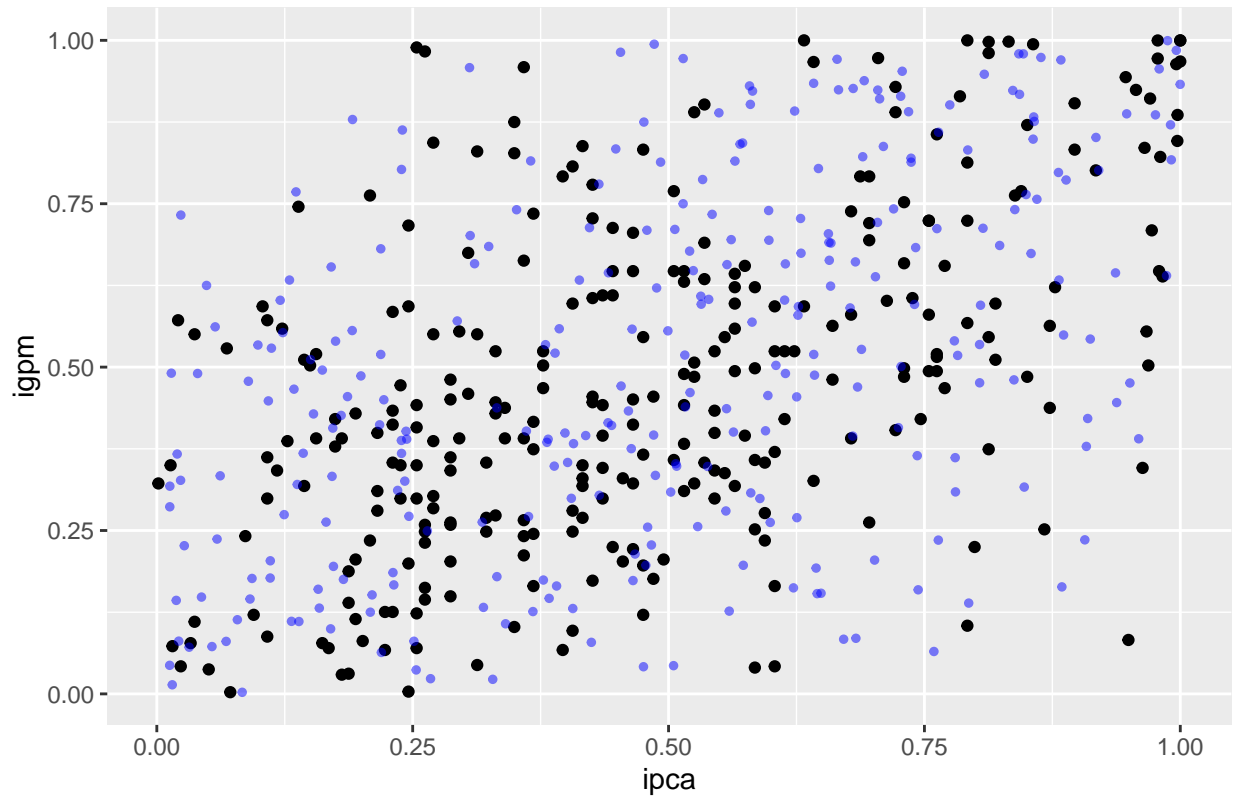
```
## rho.1
## 0.57
## The maximized loglikelihood is 56.72
## Optimization converged
```

```
rhog <- coef(gaussian_copula_fit)[1]
persp(normalCopula(dim=2,rhog),dCopula)
```



```
ug <- rCopula(nrow(uniform_df),normalCopula(dim=2,rhog))
scatter_plot <- ggplot(uniform_df, aes(x = ipca_uniform, y = igpm_uniform)) +
  geom_point() +
  labs(title = "Gráfico de Dispersão", x = "ipca", y = "igpm")
scatter_plot <- scatter_plot +
  geom_point(data = as.data.frame(ug), aes(x = ug[,1], y = ug[,2]), color = 'blue', size = 1, alpha = 0.5)
print(scatter_plot)
```

Gráfico de Dispersão



Student's t Copula A copula T é definida com \mathbf{P} que é a matriz de correlação e ν graus de liberdade.

$$C(u_1, \dots, u_n) = \mathbf{T}_{\mathbf{P}, \nu}(t_\nu^{-1}(u_1), \dots, t_\nu^{-1}(u_n))$$

$$c(u_1, \dots, u_n) = \frac{\Gamma(\frac{\nu}{2})^{n-1} \Gamma(\frac{\nu+n}{2}) (1 + \frac{1}{\nu} [t_\nu^{-1}(u_1) \cdots t_\nu^{-1}(u_n)] \mathbf{P}^{-1} \begin{bmatrix} t_\nu^{-1}(u_1) \\ \vdots \\ t_\nu^{-1}(u_n) \end{bmatrix})^{-\frac{\nu+n}{2}}}{\Gamma(\frac{\nu+1}{2})^n \sqrt{\det(\mathbf{P})} \prod_{i=1}^n (1 + \frac{1}{\nu} (t_\nu^{-1}(u_i))^2)^{-\frac{\nu+1}{2}}}$$

```
tstudent_copula <- tCopula(dim = 2)
tstudent_copula_fit <- fitCopula(tstudent_copula, uniform_df)
```

```
## Warning in .local(copula, data, ...): coercing 'data' to a matrix.
```

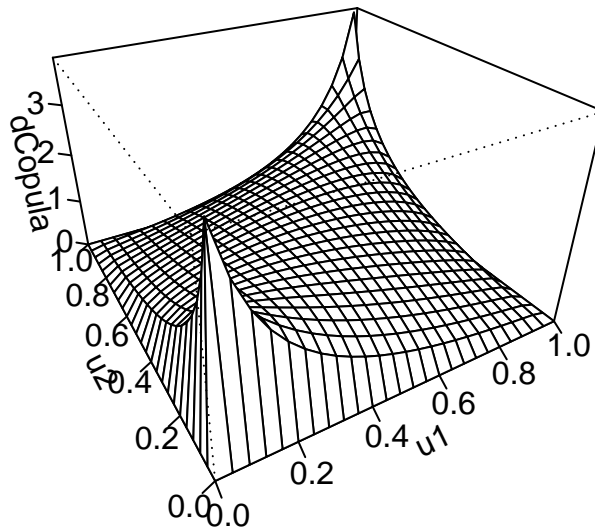
```
## Warning in var.mpl(copula, u): the covariance matrix of the parameter estimates
## is computed as if 'df.fixed = TRUE' with df = 19.8388129758225
```

```
print(tstudent_copula_fit)
```

```
## Call: fitCopula(tstudent_copula, data = uniform_df)
## Fit based on "maximum pseudo-likelihood" and 290 2-dimensional observations.
## Copula: tCopula
##   rho.1      df
```

```
## 0.5666 19.8388
## The maximized loglikelihood is 58.6
## Optimization converged
```

```
rho <- coef(tstudent_copula_fit)[1]
df <- coef(tstudent_copula_fit)[2]
persp(tCopula(dim=2,rho,df=df),dCopula)
```

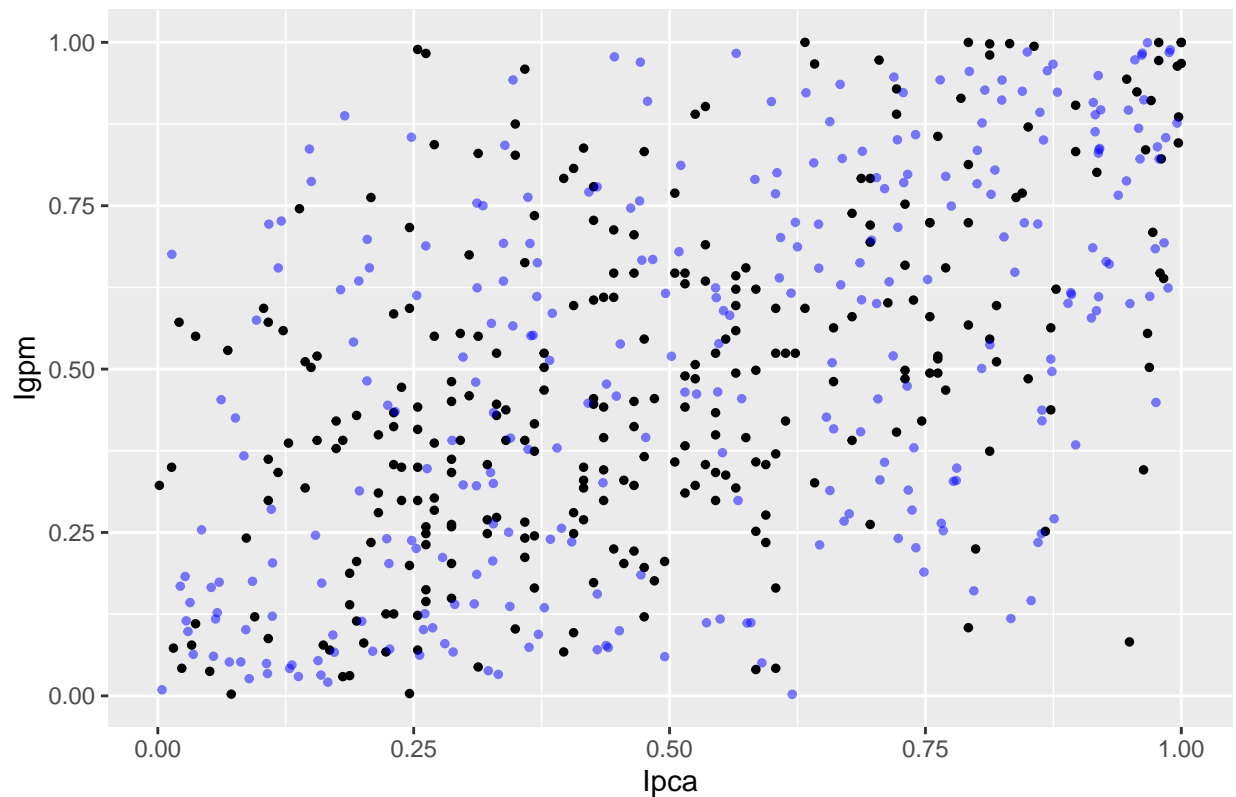


```
u <- rCopula(nrow(uniform_df),tCopula(dim=2,rho,df=df))
# Plotar o gráfico de dispersão dos seus dados
scatter_plot <- ggplot(uniform_df, aes(x = ipca_uniform, y = igpm_uniform)) +
  geom_point(size=1) +
  labs(title = "Gráfico de Dispersão", x = "Ipca", y = "Igpm")

# Adicionar os pontos gerados a partir da cópula t-Student ao gráfico de dispersão
scatter_plot <- scatter_plot +
  geom_point(data = as.data.frame(u), aes(x = u[,1], y = u[,2]), color = 'blue', size = 1, alpha = 0.5)

# Exibir o gráfico combinado
print(scatter_plot)
```

Gráfico de Dispersão



Copula Clayton

$$C(u_1, u_2) = (\max\{u_1^{-\theta} + u_2^{-\theta} - 1, 0\})^{-\frac{1}{\theta}}$$

$$c(u_1, u_2) = (1 + \theta)(u_1 u_2)^{-1-\theta}(-1 + u_1^{-\theta} + u_2^{-\theta})^{-2-\frac{1}{\theta}}$$

Onde θ é o grau de liberdade.

```
clayton_copula <- claytonCopula(dim = 2)
clayton_copula_fit <- fitCopula(clayton_copula, uniform_df)
```

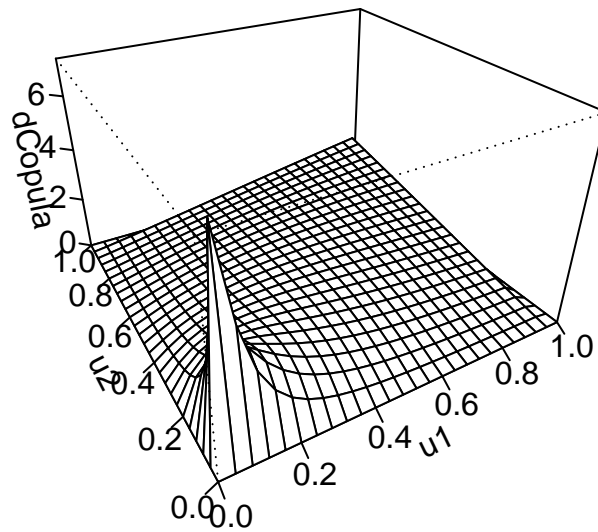
```
## Warning in .local(copula, data, ...): coercing 'data' to a matrix.
```

```
print(clayton_copula_fit)
```

```
## Call: fitCopula(clayton_copula, data = uniform_df)
## Fit based on "maximum pseudo-likelihood" and 290 2-dimensional observations.
## Copula: claytonCopula
## alpha
## 1.102
## The maximized loglikelihood is 34.12
## Optimization converged
```

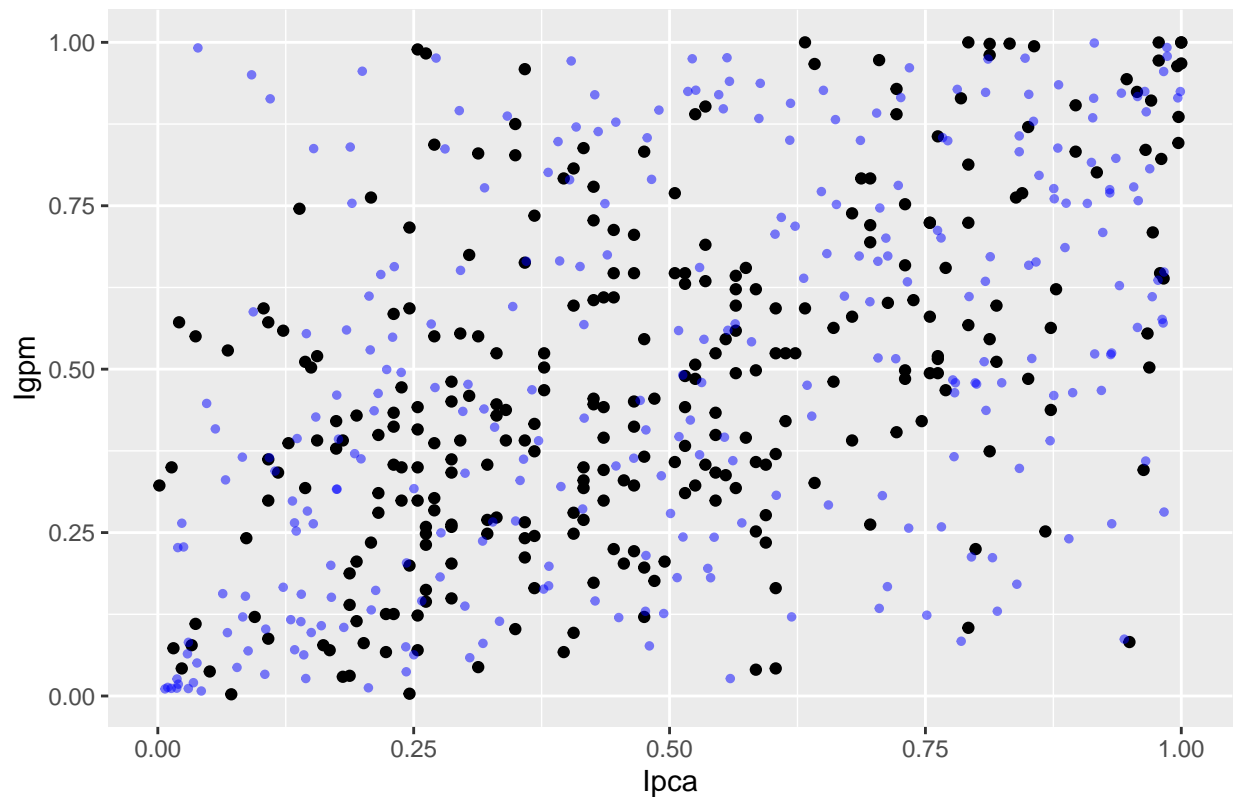


```
alpha <- coef(clayton_copula_fit)[1]
persp(claytonCopula(dim=2,alpha),dCopula)
```



```
uc <- rCopula(nrow(uniform_df),claytonCopula(dim=2,alpha))
scatter_plot <- ggplot(uniform_df, aes(x = ipca_uniform, y = igpm_uniform)) +
  geom_point() +
  labs(title = "Gráfico de Dispersão", x = "Ipca", y = "Igpm")
scatter_plot <- scatter_plot +
  geom_point(data = as.data.frame(uc), aes(x = uc[,1], y = uc[,2]), color = 'blue', size = 1, alpha = 0.5)
print(scatter_plot)
```

Gráfico de Dispersão



Frank Copula

$$C(u_1, u_2) = -\frac{1}{\theta} \ln \left(1 + \frac{(e^{-\theta u_1} - 1)(e^{-\theta u_2} - 1)}{e^{-\theta} - 1} \right)$$

$$c(u_1, u_2) = \frac{\theta e^{-\theta(u_1+u_2)}(1 - e^{-\theta})}{((1 - e^{-\theta}) + (1 - e^{-\theta u_1})(1 - e^{-\theta u_2}))^2}$$

Onde θ é o grau de liberdade.

```
frank_copula <- frankCopula(dim = 2)
frank_copula_fit <- fitCopula(frank_copula, uniform_df)
```

```
## Warning in .local(copula, data, ...): coercing 'data' to a matrix.
```

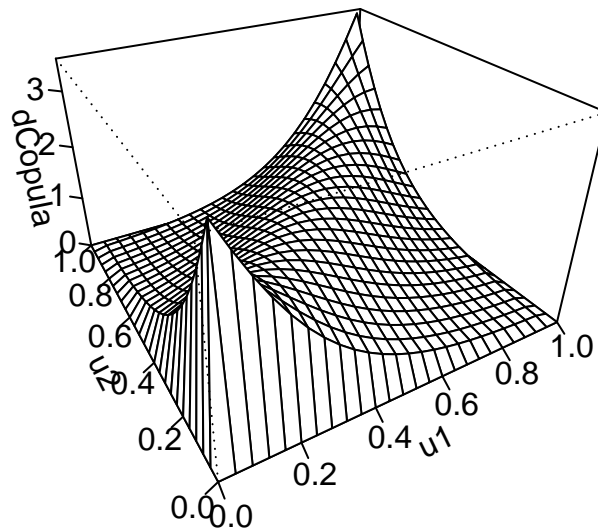
```
print(frank_copula_fit)
```

```
## Call: fitCopula(frank_copula, data = uniform_df)
## Fit based on "maximum pseudo-likelihood" and 290 2-dimensional observations.
## Copula: frankCopula
## alpha
## 4.881
## The maximized loglikelihood is 53.12
## Optimization converged
```

```

alphaf <- coef(frank_copula_fit)[1]
persp(frankCopula(dim=2,alphaf),dCopula)

```



```

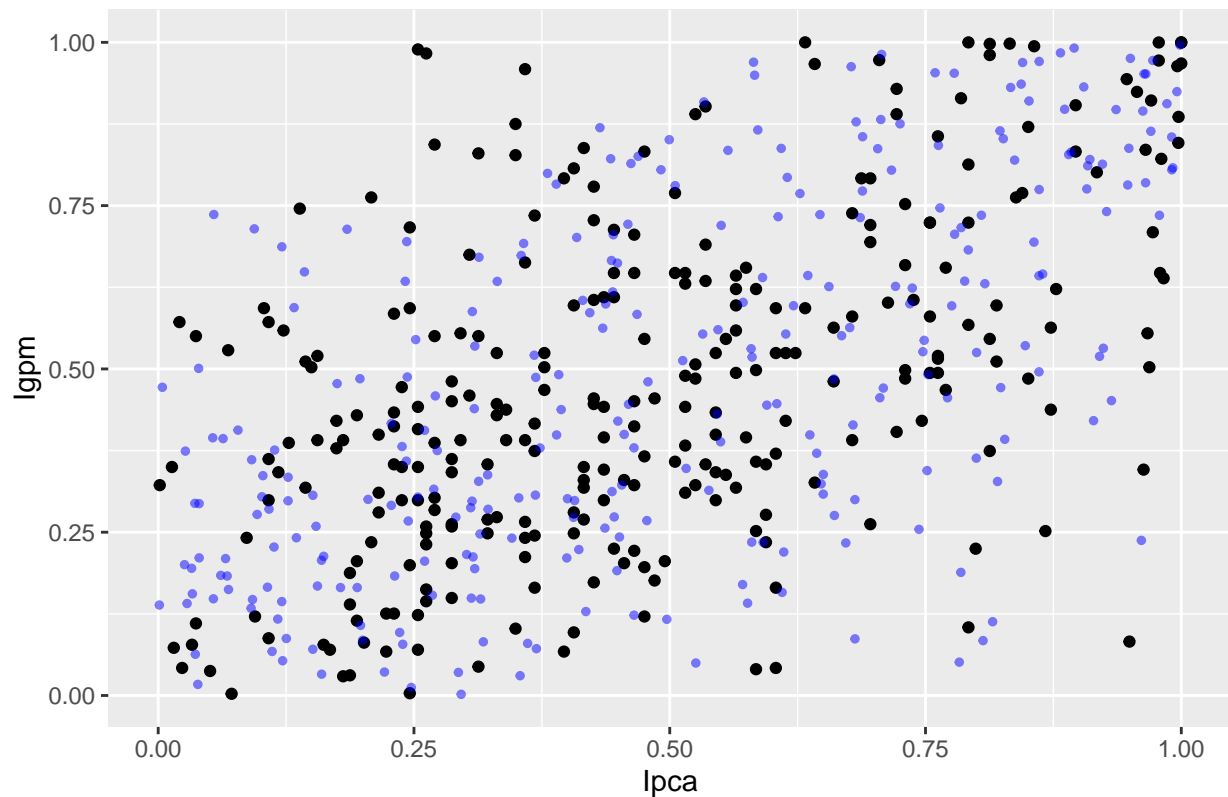
uf <- rCopula(nrow(uniform_df),frankCopula(dim=2,alphaf))

scatter_plot <- ggplot(uniform_df, aes(x = ipca_uniform, y = igpm_uniform)) +
  geom_point() +
  labs(title = "Gráfico de Dispersão", x = "Ipca", y = "Igpm")

scatter_plot <- scatter_plot +
  geom_point(data = as.data.frame(uf), aes(x = uf[,1], y = uf[,2]), color = 'blue', size = 1, alpha = 0.5)
print(scatter_plot)

```

Gráfico de Dispersão



Gumbel Copula

$$C(u_1, u_2) = \exp\{ -((-\ln(u_1))^\theta + (-\ln(u_2))^\theta)^{\frac{1}{\theta}} \}$$

$$c(u_1, u_2) = \frac{\exp\{ -((-\ln(u_1))^\theta + (-\ln(u_2))^\theta)^{\frac{1}{\theta}} \}}{u_1 u_2} \frac{((-\ln(u_1))^\theta + (-\ln(u_2))^\theta)^{\frac{2}{\theta}-2}}{(\ln(u_1) \ln(u_2))^{1-\theta}} (1 + (\theta-1)((-\ln(u_1))^\theta + (-\ln(u_2))^\theta)^{-\frac{1}{\theta}})$$

Onde θ é o grau de liberdade.

```
gumbel_copula <- gumbelCopula(dim = 2)
gumbel_copula_fit <- fitCopula(gumbel_copula, uniform_df)
```

```
## Warning in .local(copula, data, ...): coercing 'data' to a matrix.
```

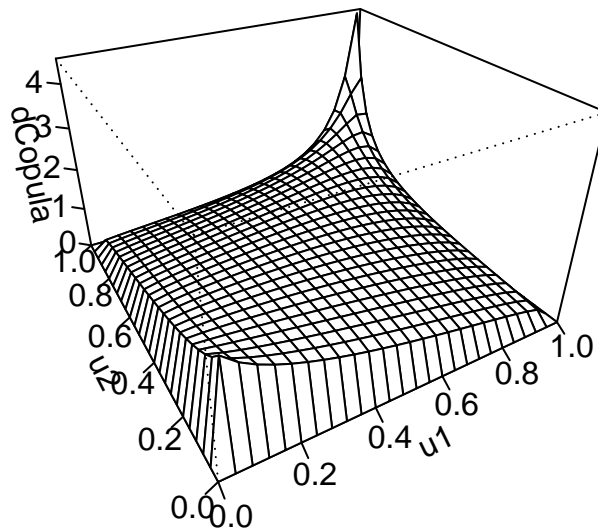
```
print(gumbel_copula_fit)
```

```
## Call: fitCopula(gumbel_copula, data = uniform_df)
## Fit based on "maximum pseudo-likelihood" and 290 2-dimensional observations.
## Copula: gumbelCopula
## alpha
## 1.374
## The maximized loglikelihood is 42.57
## Optimization converged
```

```

alphag <- coef(gumbel_copula_fit)[1]
persp(gumbelCopula(dim=2,alphag),dCopula)

```

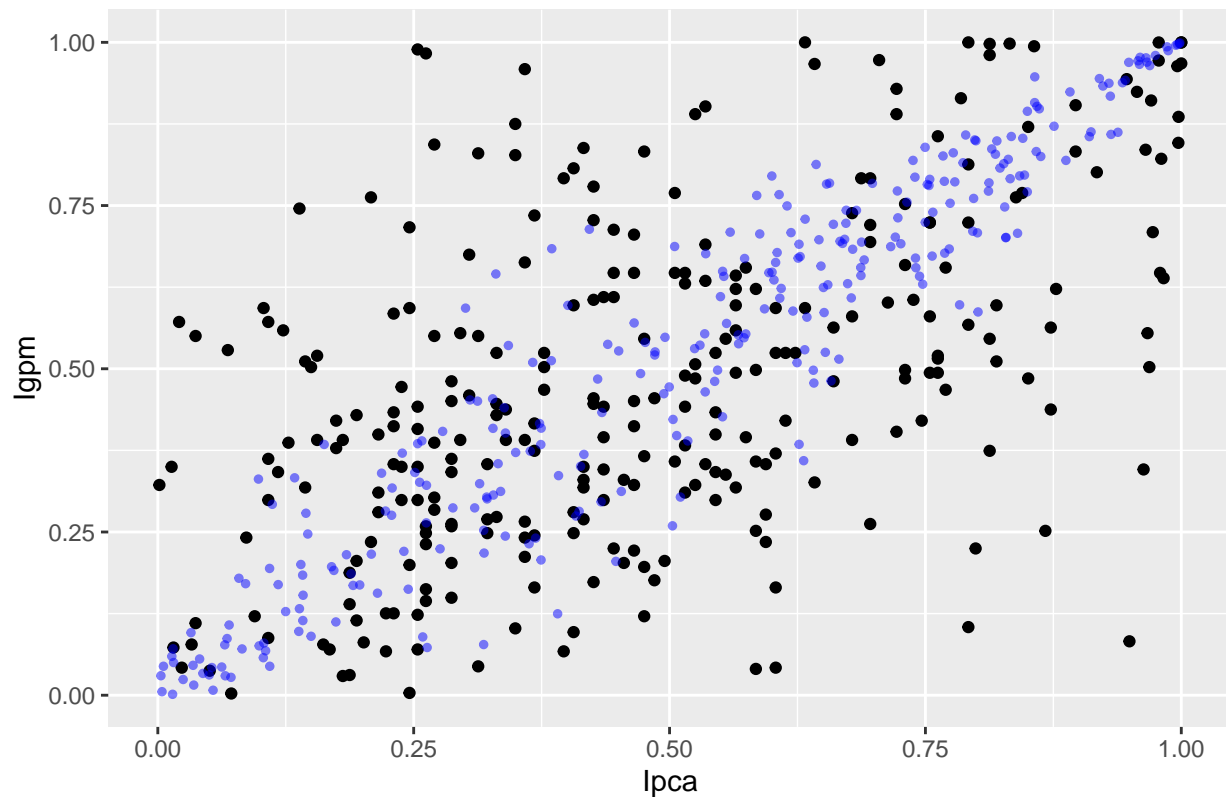


```

ug <- rCopula(nrow(uniform_df),gumbelCopula(dim=2,alphaf))
scatter_plot <- ggplot(uniform_df, aes(x = ipca_uniform, y = igpm_uniform)) +
  geom_point() +
  labs(title = "Gráfico de Dispersão", x = "Ipca", y = "Igpm")
scatter_plot <- scatter_plot +
  geom_point(data = as.data.frame(ug), aes(x = ug[,1], y = ug[,2]), color = 'blue', size = 1, alpha = 0.5)
print(scatter_plot)

```

Gráfico de Dispersão



```
likelihoods <- c(
  gaussian = logLik(gaussian_copula_fit),
  student = logLik(tstudent_copula_fit),
  clayton = logLik(clayton_copula_fit),
  frank = logLik(frank_copula_fit),
  gumbel = logLik(gumbel_copula_fit)
)

likelihood_table <- data.frame(
  Copula = names(likelihoods),
  Log_Likelihood = likelihoods
)

print(likelihood_table)
```

```
##           Copula Log_Likelihood
## gaussian gaussian      56.71746
## student  student      58.59596
## clayton  clayton      34.11967
## frank    frank       53.12304
## gumbel   gumbel      42.56660
```

Aplicação:

Agora vamos aplicar uma comparação de uma regressão feita através de cópulas e uma feita por MQO.

A regressão por copulas é feita pelo pacote “copulareg”, A função ajusta distribuições conjuntas com uma estrutura de copula de pares R-vine, que é construída de forma que a densidade condicional e a distribuição de y possam ser calculadas explicitamente.

Uma Vine-Cópula é uma estrutura de copulas que permite calcular a distribuição conjunta através de pares de cópulas, onde seria possível fazer “copula de copulas”, permitindo que relações mais complexas entre as variáveis sejam encontradas e trabalhadas no modelo.

```
data <- data[, c("ipca", "igpm", "juroreal", "dolar")]
```

```
library(copulareg)
```

```
## Warning: package 'copulareg' was built under R version 4.3.3
```

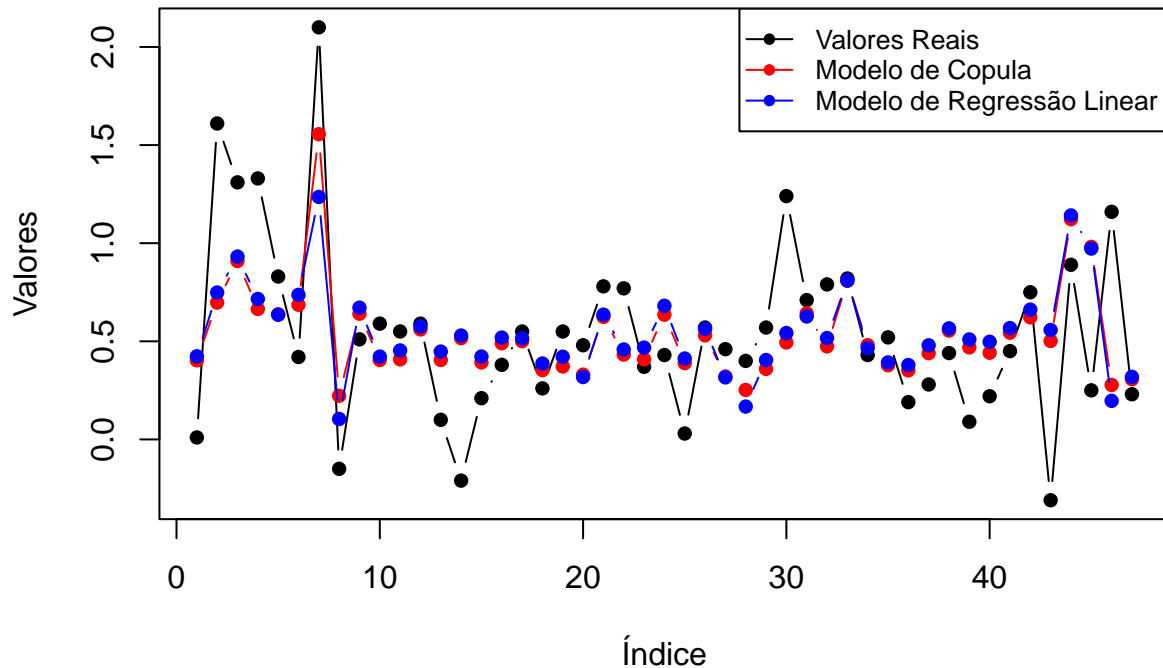
```
set.seed(1)
tr <- sample(c(TRUE, FALSE), nrow(data), TRUE, c(0.8, 0.20))
y_tr <- data$ipca[tr]
y_te <- data$ipca[!tr]
x_tr <- apply(data[tr, -1], 2, as.numeric)
x_te <- apply(data[!tr, -1], 2, as.numeric)
var_type_x <- apply(x_tr, 2,
  function(x) if(length(unique(x)) > 10) "c" else "c")
```

```
md <- copulareg::copulareg(y_tr, x_tr, "c", var_type_x)
pred_copula <- predict(md, new_x = x_te)
```

```
modelo_regressao <- lm(ipca ~ +juroreal + dolar + igpm, data = data[tr,])
pred_regressao <- predict(modelo_regressao, newdata = data[!tr, -1])
```

```
# Criar o plot
plot(y_te, col = "black", type = "b", pch = 16, ylim = range(c(y_te, pred_copula, pred_regressao)),
  main = "Comparação das Previsões dos Modelos com Valores Reais",
  xlab = "Índice", ylab = "Valores")
lines(pred_copula, col = "red", type = "b", pch = 16)
lines(pred_regressao, col = "blue", type = "b", pch = 16)
legend("topright", legend = c("Valores Reais", "Modelo de Copula", "Modelo de Regressão Linear"),
  col = c("black", "red", "blue"), pch = 16, lty = 1, cex = 0.8)
```

Comparação das Previsões dos Modelos com Valores Reais



```
mse_copula <- mean((pred_copula - y_te)^2)
cat("Erro Quadrático Médio do Modelo de Copula:", mse_copula, "\n")
```

```
## Erro Quadrático Médio do Modelo de Copula: 0.1347763
```

```
mse_regressao <- mean((pred_regressao - y_te)^2)
cat("Erro Quadrático Médio do Modelo de Regressão Linear:", mse_regressao, "\n")
```

```
## Erro Quadrático Médio do Modelo de Regressão Linear: 0.1469193
```

Referências :

- 1) Nelsen, R. (1999). An Introduction to Copulas, 2nd ed. Springer, New York.
- 2) Fisher, N.I. (1997), Copulas, in Encyclopedia of Statistical Sciences. Update Vol.1
- 3) Mukhopadhyay, S. Parzen, E. (2019), Nonparametric Universal Copula Modeling. Applied Stochastic Models in Business and Industry,
- 4) Krämer, N., Brechmann, E. C., Silvestrini, D., & Czado.(2013) Total loss estimation using copula-based regression models. Insurance: Mathematics and Economics.