Lista 4
Finanças Quantitativas
Diogo Wolff Surdi

Questão 3.17

17.1

Note que, pela LOTUS, temos:

$$\begin{aligned}
\mathbb{E}[f(Z)e^{\sigma Z}] &= \int \frac{f(x)}{\sqrt{2\pi}} e^{\sigma x} e^{-\frac{x^2}{2}} dx \\
&= \int \frac{f(x)}{\sqrt{2\pi}} e^{\sigma x - \frac{x^2}{2}} dx \\
&= \int \frac{f(x)}{\sqrt{2\pi}} e^{\frac{2\sigma x - x^2}{2}} dx
\end{aligned}$$

Completando os quadrados, temos:

$$\begin{aligned}
2\sigma x - x^2 &= 2\sigma x - x^2 + \sigma^2 - \sigma^2 \\
&= -(x-\sigma)^2 + \sigma^2
\end{aligned}$$

Logo:

$$\begin{aligned}
\mathbb{E}[f(Z)e^{\sigma Z}] &= \int \frac{f(x)}{\sqrt{2\pi}} e^{\frac{-(x-\sigma)^2 + \sigma^2}{2}} \\
&= \int \frac{f(x)}{\sqrt{2\pi}} e^{\frac{\sigma^2}{2}} e^{-\frac{(x-\sigma)^2}{2}} \\
&= e^{\frac{\sigma^2}{2}} \int \frac{f(x)}{\sqrt{2\pi}} e^{-\frac{(x-\sigma)^2}{2}}
\end{aligned}$$

Note que a expressão da integral é a de uma normal com média σ. Com isso, a integral (e por consequência o valor esperado) são transladados em relação à distribuição Z, logo:

$$\mathbb{E}[f(Z)e^{\sigma Z}] = e^{\frac{\sigma^2}{2}} \mathbb{E}[f(Z+\sigma)]$$

Para a segunda parte, basta notar que $Z = \frac{(X-\mu)}{\sigma}$, donde temos:

$$\mathbb{E}[e^X] = \mathbb{E}[e^{\sigma Z + \mu}] = e^{\mu}\mathbb{E}[e^{\sigma Z}] = e^{\mu + \frac{\sigma^2}{2}} \mathbb{E}[1] = e^{\mu + \frac{\sigma^2}{2}}$$

17.2

Questão 3.18

18.1

Temos que $X = e^Y$ onde $Y \sim N(\mu, \sigma^2)$. Com isso temos:

$$F_X(x) = P(X \leq x) = P(e^Y \leq x) = P(Y \leq \ln x) = F_Y(\ln x)$$

Note que:

$$f_Y(y) = \frac{\partial}{\partial y} F_X(x) = \frac{\partial}{\partial y} F_Y(\ln x) = \frac{1}{x} f_Y(\ln x)$$

Como Y é normal, temos que:

$$f_Y(y) = \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{(y-\mu)^2}{\sigma^2}\right)$$

Logo:

$$f_X(x) = \frac{1}{x}\frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{(\ln x-\mu)^2}{\sigma^2}\right)$$

Para a questão, tomaremos que X é log-normal (0, 1) e Y é log-normal (0, σ^2). Ademais, sabe-se que $\rho_{min} = \rho$ para variáveis contramonotônicas, enquanto que $\rho_{max} = \rho$ para variáveis comonotônicas.

18.2

Para esse item, note que a função exponencial é estritamente crescente, logo $\rho_{min} = \rho(e^Z, e^{-\sigma Z})$. Ademais, pela questão 3.17 temos que $\mathbb{E}[e^Z] = e^{\frac{1}{2}}$ e $\mathbb{E}[e^{2Z}] = e^2$, donde $Var(e^Z) = e^2 - e = e(e-1)$. Vale resultado equivalente para $e^{-\sigma Z}$, donde $Var(e^{-\sigma Z}) = e^{\sigma^2}(e^{\sigma^2} - 1)$. Para a covariância, temos que:

$$\begin{aligned}
cov(e^Z, e^{-\sigma Z}) &= \mathbb{E}[e^{(1-\sigma)Z}] - \mathbb{E}[e^Z]\mathbb{E}[e^{-\sigma Z}] \\
&= e^{\frac{(1-\sigma)^2}{2}} - e^{\frac{\sigma^2+1}{2}} \\
&= e^{\frac{1-2\sigma+\sigma^2}{2}} - e^{\frac{\sigma^2+1}{2}} \\
&= e^{\frac{\sigma^2+1}{2}}(e^{-\sigma} - 1)
\end{aligned}$$

Com isso, temos que:

$$\rho_{min} = \frac{e^{\frac{\sigma^2+1}{2}}(e^{-\sigma}-1)}{\sqrt{e(e-1)e^{\sigma^2}(e^{\sigma^2}-1)}} = \frac{e^{-\sigma}-1}{\sqrt{(e-1)(e^{\sigma^2}-1)}}$$

18.3

Para encontrar ρ_{max} basta substituir $e^{-\sigma Z}$ nas contas do item anterior por $e^{\sigma Z}$. Com isso, encontramos $cov(e^Z, e^{\sigma Z}) = e^{\frac{\sigma^2+1}{2}}(e^\sigma - 1)$, donde:

$$\rho_{max} = \frac{e^\sigma - 1}{\sqrt{(e-1)(e^{\sigma^2} - 1)}}$$

18.4

Para o limite negativo, temos que o numerador tende a 0, enquanto o denominador tende a infinito, logo o limite inferior se aproxima do 0. Para o limite positivo usarei l'Hopital:

$$\lim_{\sigma\to\infty} \rho_{max} = \lim_{\sigma\to\infty} \frac{\sqrt{(e-1)(e^{\sigma^2} - 1)}}{(e-1)\sigma e^{\sigma^2 - 1}}$$

Os termos do numerador são a raiz de termos do denominador, e este é multiplicado por σ, logo o limite também é 0.

Questão 3.24

24.1

Primeiramente, vou utilizar o pacote copula para gerar amostras de cópulas gaussianas bidimensionais. O código envolve a mesma linha 21 vezes (uma para cada ρ) pois não consegui encontrar um meio de usar um for que gerasse as 21 cópulas, então vou apresentar apenas uma amostra como exemplo.

```
SD7 <- rCopula(2000,normalCopula(0.3))
```

```
> head(SD7)
[,1]        [,2]
[1,] 0.5579378 0.45877269
[2,] 0.6831545 0.29538059
[3,] 0.3872245 0.40120215
[4,] 0.1027863 0.09106661
[5,] 0.9828730 0.78213367
[6,] 0.5544919 0.36428488
```

Não consegui transformar diretamente as amostras originais em amostras normais, então gerei uma distribuição normal multivariada a partir da cópula gaussiana, e tomarei amostras dela.

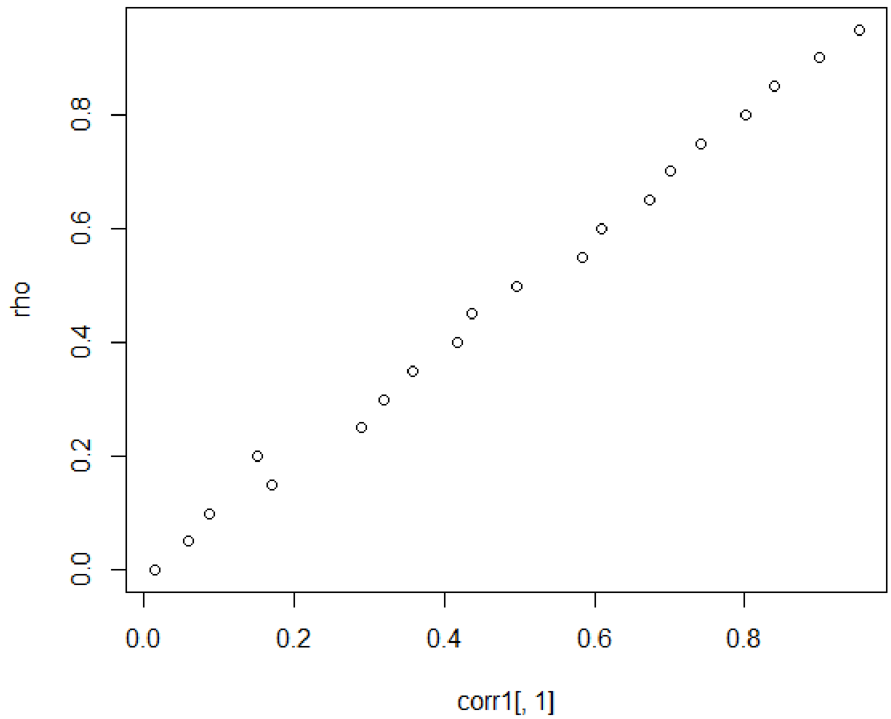
24.1.2

A função normalCopula não aceitou $\rho = 1$, então omiti tal valor do código para gerar a matriz de correlações.

```
rho <- matrix(nrow=20, ncol=1)
corr1 <- matrix(nrow=20, ncol=3)
rho[1,1]<-0

for (i in 2:20){
rho[i,1] <- rho[i-1,1]+0.05
}

corr1 <- matrix(nrow=20, ncol=3)
for (i in 1:20){
teste<-mvdc(normalCopula(rho[i, 1]), c("norm","norm"), list(list(mean=0,sd=1)
steste <- rMvdc(2000,teste)
corr1[i,1] <- cor(steste[,1], steste[,2], method = "pearson")
corr1[i,2] <-cor(steste[,1], steste[,2], method = "kendall")
corr1[i,3] <-cor(steste[,1], steste[,2], method = "spearman")
}

> corr1
[,1]          [,2]          [,3]
[1,]  0.01361288 0.01363082 0.02021014
[2,]  0.05848784 0.03716758 0.05558875
[3,]  0.08624170 0.05600400 0.08389410
[4,]  0.17002320 0.10107954 0.15153448
[5,]  0.14998081 0.09425913 0.14126141
[6,]  0.28896133 0.18390295 0.27052811
[7,]  0.31942075 0.20655828 0.30566670
[8,]  0.35826540 0.23157679 0.33970540
[9,]  0.41636563 0.26872536 0.39386114
[10,]  0.43619186 0.28496248 0.41689367
[11,]  0.49715565 0.32924562 0.47675354
[12,]  0.58341591 0.39144972 0.55838591
[13,]  0.60893375 0.41523462 0.58786938
[14,]  0.67354596 0.46912656 0.65318661
[15,]  0.70193300 0.49269935 0.68200879
[16,]  0.74165283 0.53452626 0.73066192
[17,]  0.80031170 0.59771386 0.79263630
[18,]  0.83958837 0.63518359 0.82851901
[19,]  0.89897045 0.71474837 0.89411035
[20,]  0.95177617 0.80140070 0.94755141
```

24.1.3

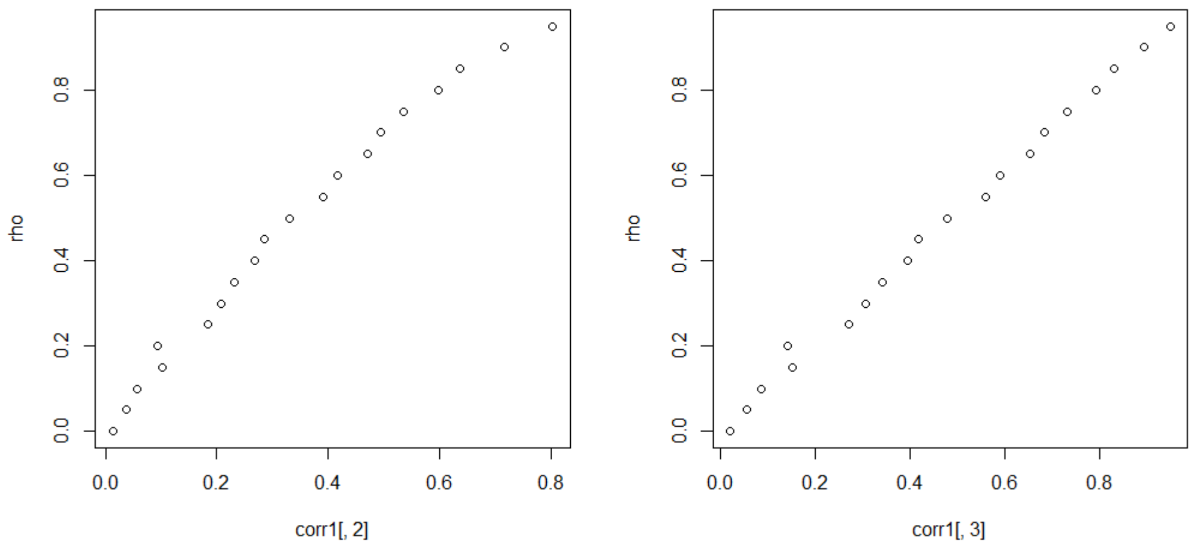
```
plot(corr1[,1],rho)
```



A correlação de Pearson amostral é bem próxima da correlação entre as próprias distribuições, o que indica que ela é uma boa aproximação para a correlação real.

24.1.4

```
par(mfrow=c(1,2))
plot(corr1[,2],rho)
plot(corr1[,3],rho)
```



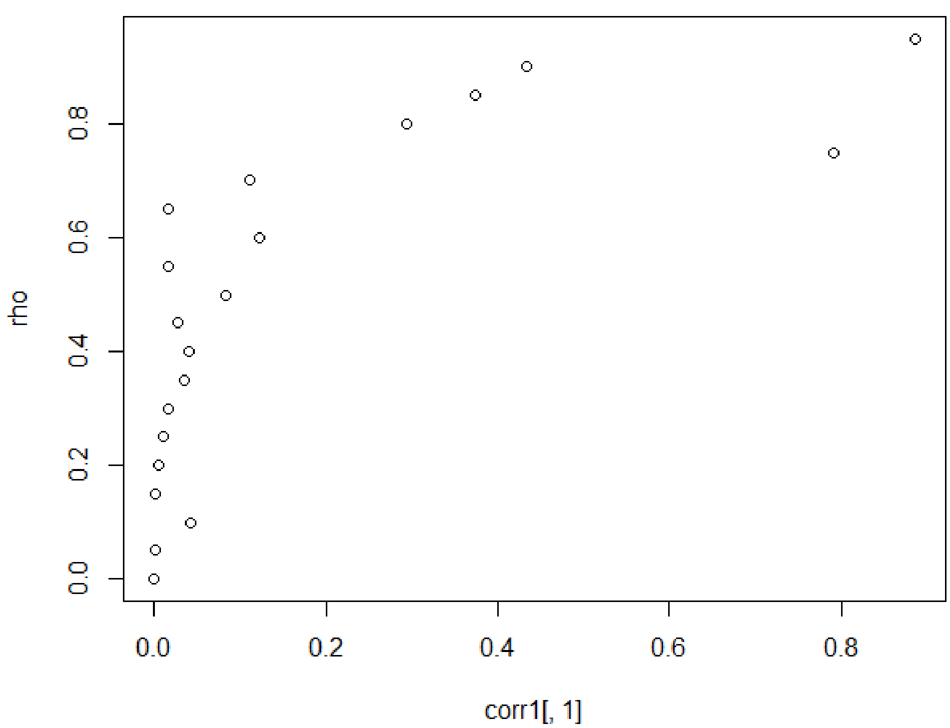
A correlação de Kendall (a esquerda) é bem menor do que ρ, indicando uma subestimação. Por outro lado, a correlação de Spearman apresenta valores parecidos com os reais, sendo muito próxima da de Pearson.

24.1.5

Utilizarei técnica parecida à utilizada anteriormente para simular.

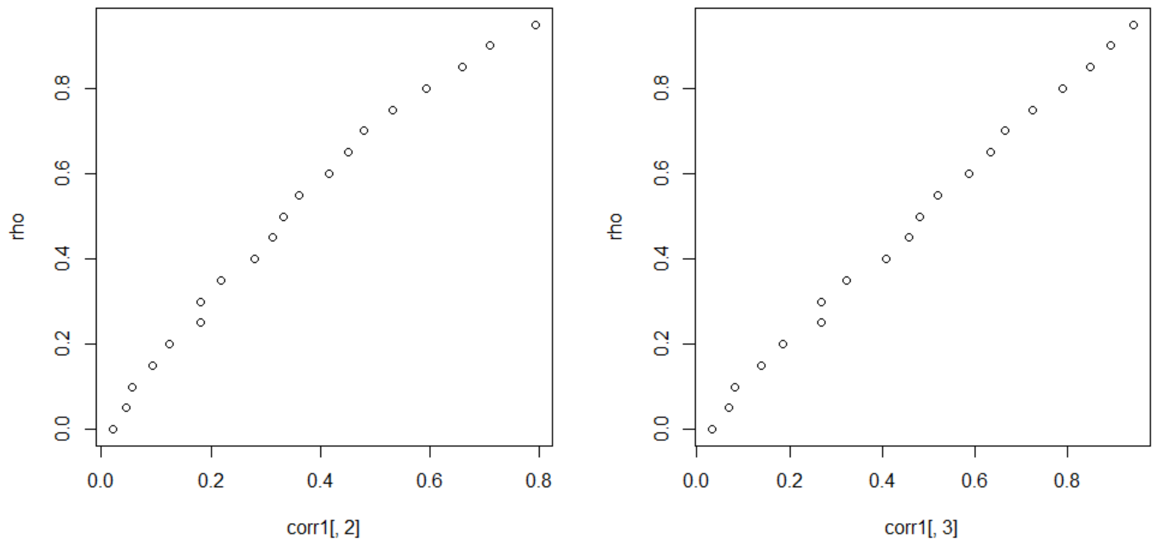
```
rho <- matrix(nrow=20, ncol=1)
corr1 <- matrix(nrow=20, ncol=3)
rho[1,1]<-0

for (i in 2:20){
rho[i,1] <- rho[i-1,1]+0.05
}

corr1 <- matrix(nrow=20, ncol=3)
corrteste <- matrix(nrow=20, ncol=3)
for (i in 1:20){
teste<-mvdc(normalCopula(rho[i, 1]), c("cauchy","cauchy"),
        list(list(location=0,scale=1),list(location=0,scale=1)))
steste <- rMvdc(2000,teste)
corr1[i,1] <- cor(steste[,1], steste[,2], method = "pearson")
corr1[i,2] <-cor(steste[,1], steste[,2], method = "kendall")
corr1[i,3] <-cor(steste[,1], steste[,2], method = "spearman")
}

> corr1
[,1]          [,2]          [,3]
[1,] 0.0003204695 0.02159180 0.03218481
```

```
[2,]  0.0015514021 0.04519960 0.06845508
[3,]  0.0427751172 0.05531766 0.08249222
[4,]  0.0020750534 0.09256128 0.13774839
[5,]  0.0062130510 0.12355778 0.18435948
[6,]  0.0106826023 0.18033017 0.26866075
[7,]  0.0163773965 0.18142471 0.26915943
[8,]  0.0346628899 0.21788894 0.32245153
[9,]  0.0404871733 0.27960680 0.40816775
[10,]  0.0281987449 0.31348874 0.45766505
[11,]  0.0840381248 0.33316558 0.47955629
[12,]  0.0168159062 0.36040020 0.51819141
[13,]  0.1218430095 0.41603802 0.58732821
[14,]  0.0170899157 0.45152276 0.63208668
[15,]  0.1123800567 0.47968284 0.66415262
[16,]  0.7907183573 0.53235018 0.72399542
[17,]  0.2938429648 0.59296448 0.78777492
[18,]  0.3745318990 0.65978989 0.84870474
[19,]  0.4327866148 0.71124762 0.89118737
[20,]  0.8860640464 0.79292346 0.94136547

plot(corr1[,1], rho)
```



Como a distribuição de Cauchy tem variância indefinida, a correlação de Pearson não é definida, gerando um gráfico absurdo.

```
par(mfrow=c(1,2))
plot(corr1[,2], rho)
plot(corr1[,3], rho)
```



24.2

```
SD7item2 <- rCopula(2000, gumbelCopula(4))

> head(SD7item2)
[,1]          [,2]
[1,] 0.03195630 0.02935435
[2,] 0.91934056 0.92616983
[3,] 0.16524291 0.19554693
[4,] 0.06821096 0.22524028
[5,] 0.84559711 0.79911042
[6,] 0.12534141 0.23590344
```

A função não estava aceitando $\beta = 1$, então fiz a simulação a partir de 1.5.

24.2.2

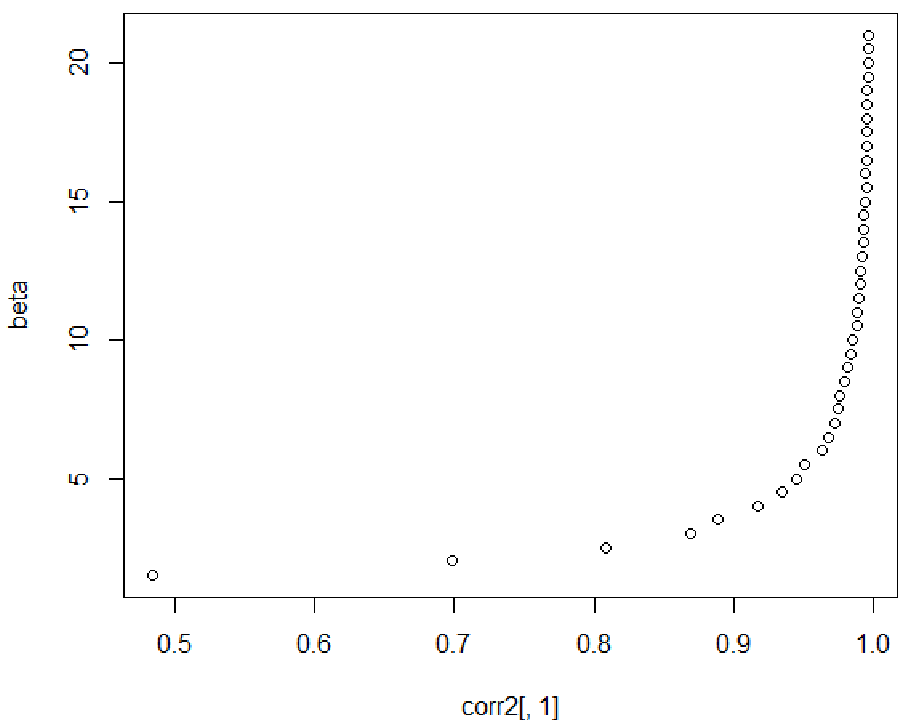
```
beta <- matrix(nrow=40, ncol=1)
corr2 <- matrix(nrow=40, ncol=3)
beta[1,1]<-1.5
```

```
for (i in 2:40){
beta[i,1] <- beta[i-1,1]+0.5
}

corr2 <- matrix(nrow=40, ncol=3)
for (i in 1:40){
teste<-mvdc(gumbelCopula(beta[i, 1]), c("norm","norm"),
         list(list(mean=0,sd=1),list(mean=0,sd=1)))
steste <- rMvdc(2000,teste)
corr2[i,1] <- cor(steste[,1], steste[,2], method = "pearson")
corr2[i,2] <-cor(steste[,1], steste[,2], method = "kendall")
corr2[i,3] <-cor(steste[,1], steste[,2], method = "spearman")
}

> corr2
[,1]        [,2]        [,3]
[1,] 0.4841239 0.3170245 0.4520041
[2,] 0.6985825 0.5026743 0.6855712
[3,] 0.8087388 0.6132176 0.8018156
[4,] 0.8694131 0.6795588 0.8599238
[5,] 0.8891731 0.7116788 0.8860809
[6,] 0.9174385 0.7531456 0.9125828
[7,] 0.9342931 0.7801911 0.9303695
[8,] 0.9453167 0.8007354 0.9441780
[9,] 0.9501150 0.8085293 0.9483521
[10,] 0.9635278 0.8354277 0.9618726
[11,] 0.9680397 0.8453977 0.9659678
[12,] 0.9723533 0.8613437 0.9723919
[13,] 0.9749077 0.8674047 0.9746959
[14,] 0.9761486 0.8710005 0.9759165
[15,] 0.9792439 0.8773497 0.9781822
[16,] 0.9819783 0.8865633 0.9813451
[17,] 0.9837854 0.8930235 0.9828347
[18,] 0.9854378 0.8975178 0.9848014
[19,] 0.9879013 0.9077909 0.9875759
[20,] 0.9880199 0.9064062 0.9869236
[21,] 0.9890287 0.9136548 0.9891265
[22,] 0.9906501 0.9186833 0.9902370
[23,] 0.9902013 0.9163772 0.9898050
[24,] 0.9919471 0.9220510 0.9909811
[25,] 0.9925980 0.9279590 0.9924673
[26,] 0.9929252 0.9323272 0.9934010
[27,] 0.9928344 0.9298989 0.9926955
[28,] 0.9934974 0.9345893 0.9936310
[29,] 0.9946701 0.9381641 0.9944047
```

```
[30,]  0.9937218  0.9352936  0.9939064
[31,]  0.9947550  0.9398119  0.9945370
[32,]  0.9947449  0.9404392  0.9948661
[33,]  0.9951400  0.9421241  0.9949801
[34,]  0.9955398  0.9443522  0.9953427
[35,]  0.9956614  0.9443452  0.9953375
[36,]  0.9957389  0.9458529  0.9957555
[37,]  0.9959994  0.9475998  0.9959650
[38,]  0.9962712  0.9490975  0.9961891
[39,]  0.9965812  0.9516678  0.9966112
[40,]  0.9966961  0.9518719  0.9966159
```

24.2.3

```
plot(corr2[,1], beta)
```

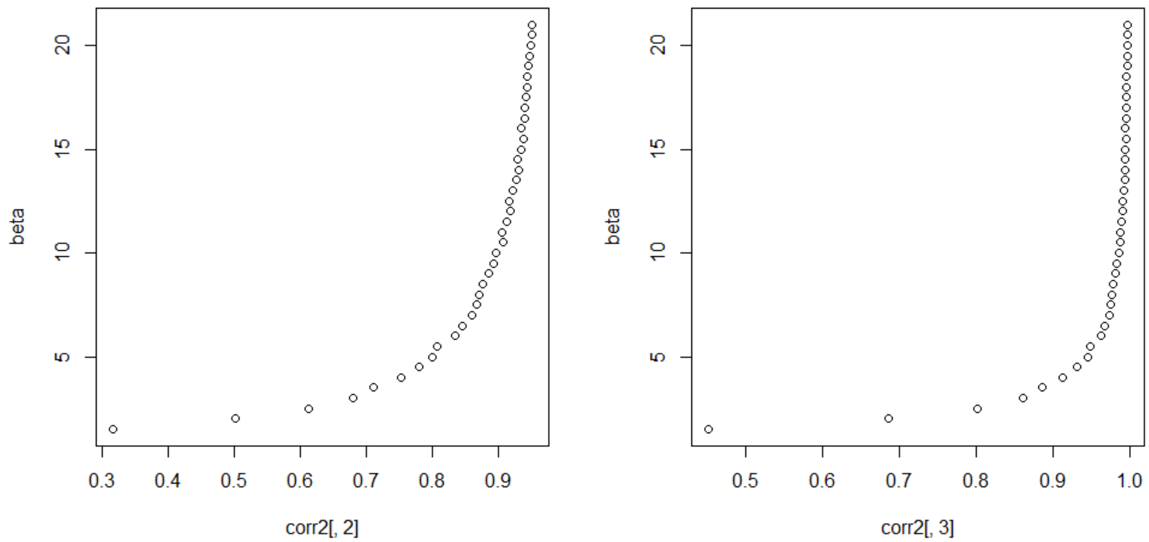


24.2.4

Vou notar aqui que a questão fala para plotar em relação a ρ, porém não vi ele em momento algum no código para esse item.

```
par(mfrow=c(1,2))
plot(corr2[,2], beta)
```

```
plot(corr2[,3], beta)
```

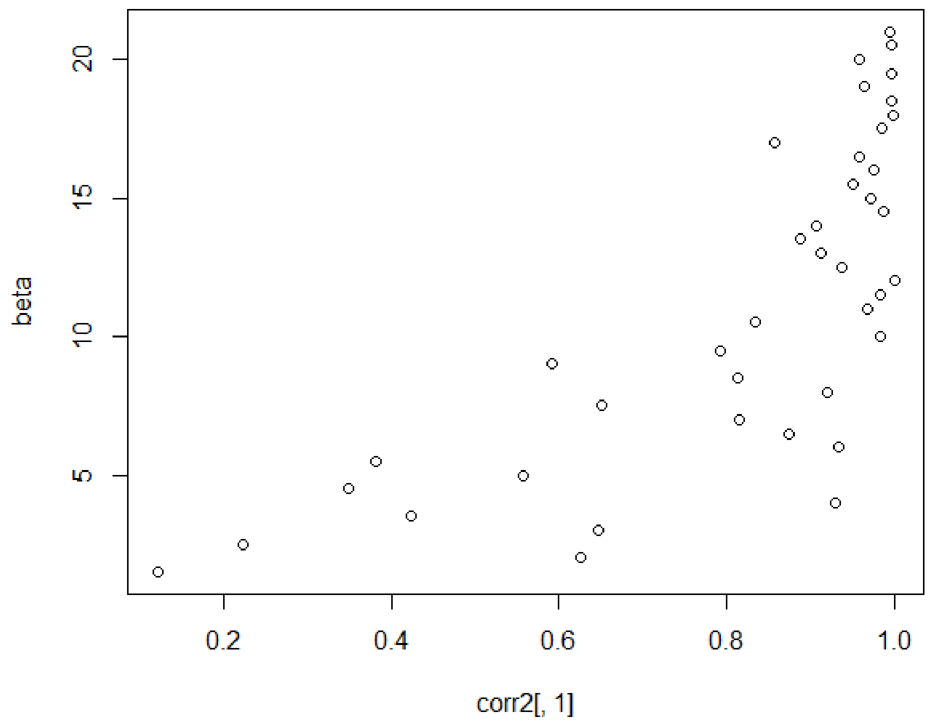


24.2.5

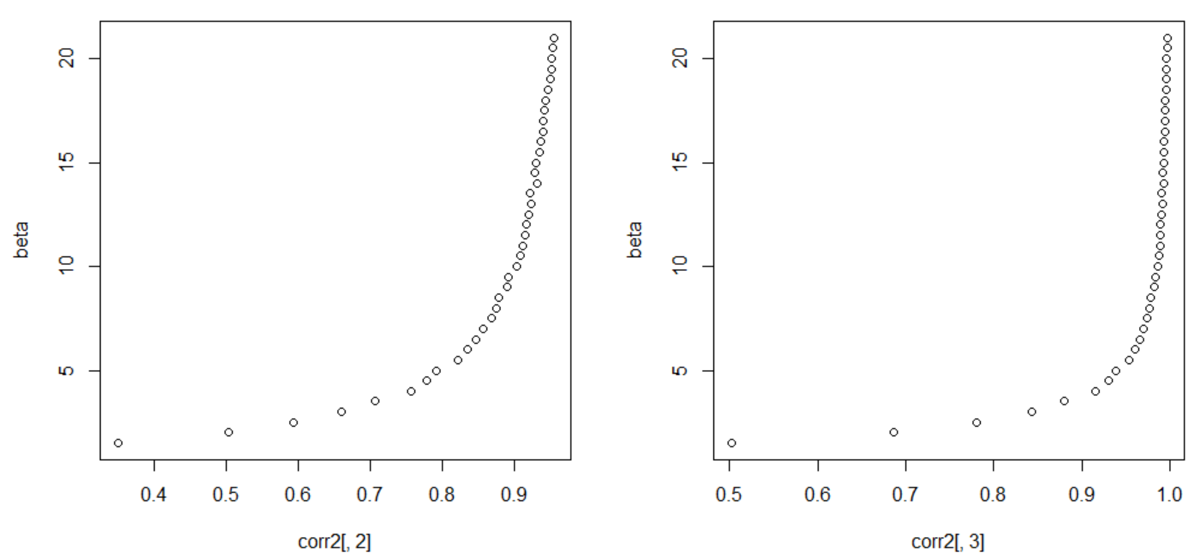
```
beta <- matrix(nrow=40, ncol=1)
corr2 <- matrix(nrow=40, ncol=3)
beta[1,1]<-1.5

for (i in 2:40){
beta[i,1] <- beta[i-1,1]+0.5
}

corr2 <- matrix(nrow=40, ncol=3)
for (i in 1:40){
teste<-mvdc(gumbelCopula(beta[i, 1]), c("cauchy","cauchy"), list(list(locatio
steste <- rMvdc(2000,teste)
corr2[i,1] <- cor(steste[,1], steste[,2], method = "pearson")
corr2[i,2] <-cor(steste[,1], steste[,2], method = "kendall")
corr2[i,3] <-cor(steste[,1], steste[,2], method = "spearman")
}

> corr2
[,1]        [,2]        [,3]
[1,] 0.1210999 0.3497419 0.5019227
[2,] 0.6251803 0.5028414 0.6859553
[3,] 0.2237671 0.5930195 0.7805388
```

```
[4,]  0.6469954 0.6603422 0.8435572
[5,]  0.4243963 0.7060950 0.8801649
[6,]  0.9296202 0.7556368 0.9157404
[7,]  0.3499351 0.7781721 0.9299968
[8,]  0.5573221 0.7912826 0.9383118
[9,]  0.3810059 0.8219040 0.9540346
[10,]  0.9327899 0.8340830 0.9603272
[11,]  0.8739633 0.8456118 0.9658818
[12,]  0.8149914 0.8555718 0.9703449
[13,]  0.6515025 0.8678179 0.9747851
[14,]  0.9195109 0.8740210 0.9772441
[15,]  0.8122903 0.8781581 0.9783342
[16,]  0.5912713 0.8895418 0.9825803
[17,]  0.7925018 0.8912786 0.9830939
[18,]  0.9824630 0.9032806 0.9859863
[19,]  0.8337141 0.9074177 0.9875277
[20,]  0.9687268 0.9105173 0.9885130
[21,]  0.9825492 0.9135378 0.9891313
[22,]  0.9999566 0.9163892 0.9897519
[23,]  0.9381691 0.9197699 0.9905660
[24,]  0.9129899 0.9233697 0.9913949
[25,]  0.8885618 0.9209725 0.9908799
[26,]  0.9068627 0.9317719 0.9932258
[27,]  0.9873727 0.9284342 0.9925223
[28,]  0.9709242 0.9290975 0.9926489
[29,]  0.9513847 0.9349995 0.9937814
[30,]  0.9765138 0.9352386 0.9937918
[31,]  0.9583998 0.9399480 0.9945806
[32,]  0.8566301 0.9392076 0.9944964
[33,]  0.9851596 0.9413567 0.9948874
[34,]  0.9992662 0.9433007 0.9951582
[35,]  0.9971951 0.9456718 0.9956199
[36,]  0.9633900 0.9489045 0.9960840
[37,]  0.9960910 0.9503242 0.9964174
[38,]  0.9581234 0.9509895 0.9963290
[39,]  0.9958326 0.9525063 0.9967073
[40,]  0.9938028 0.9537349 0.9968524
```

```
plot(corr2[,1], beta)
```



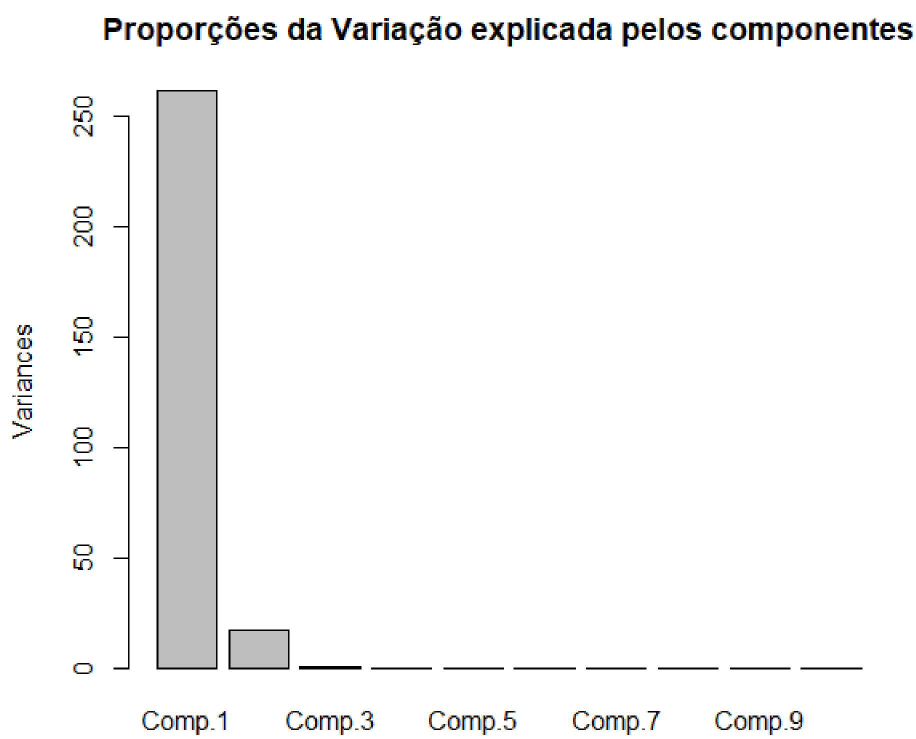

```
par(mfrow=c(1,2))
plot(corr2[,2], beta)
plot(corr2[,3], beta)
```



A análise aqui é semelhante à do item 1.5

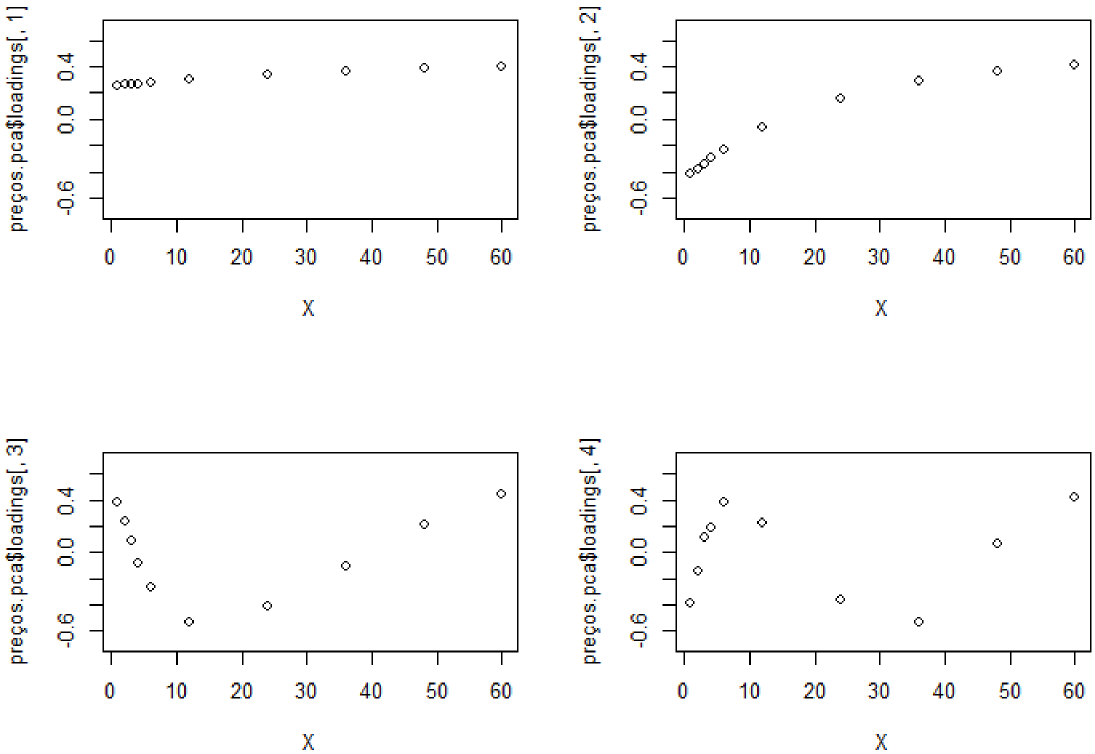
Análise

```
precos <- prices_Swap_PRE_DI
precos.pca <- princomp(precos[2:11])
plot(precos.pca,
        main ='Proporcoes da Variacao explicada pelos componentes')
```



Como pode-se ver, as variações são explicadas quase em sua totalidade pelos três primeiros componentes.

```
X <- c(48,1,60,2,3,4,6,12,24,36)
par(mfrow=c(2,2))
plot(X, precos.pca$loadings[,1], ylim=c(-.7,.7))
plot(X, precos.pca$loadings[,2], ylim=c(-.7,.7))
plot(X, precos.pca$loadings[,3], ylim=c(-.7,.7))
plot(X, precos.pca$loadings[,4], ylim=c(-.7,.7))
precos.pca$loadings[,2]
```



O gráfico do primeiro componente é quase constante, indicando que ele representa a média do retorno. O gráfico do segundo já é crescente, indicando a tendência positiva (pois o retorno é positivo) do retorno. O formato da terceira curva indica que ela representa a curvatura da curva de rendimento, enquanto que não dá para retirar algo particularmente interessante da quarta curva, indicando que ela é em maior parte composta de erro.