

Analiza porównawcza cen akcji Apple i Microsoft

Mackiewicz-Kubiak Aleksander, Pagielska Marta

2025-?-?

Pakiety

Do analizy użyto następujących pakietów:

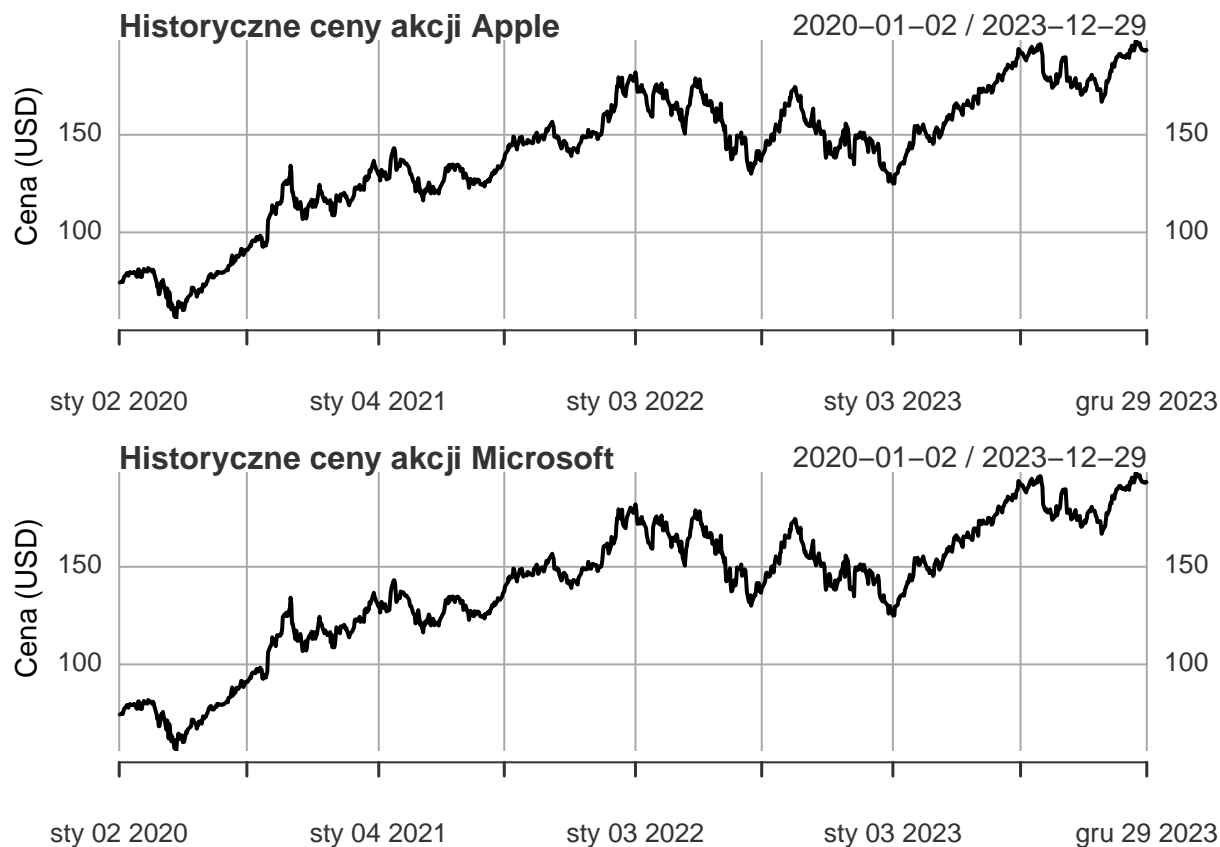
```
library(tidyr)
library(gamlss)
library(dplyr)
library(fitdistrplus)
library(usefun)
library(quantmod)
library(scales)
library(copula)
library(psych)
library(MVN)
library(readr)
library(xts)
options(scipen = 999) # celem wyłączenia notacji naukowej
pdf.options(encoding='ISOLatin2.enc')
```

KROK 1

Do analizy wybrano ceny zamknięcia akcji firm Apple i Microsoft, czyli dwóch potentatów na rynku elektronicznym. Dane pochodzące z okresu 4 lat (od 1 stycznia 2020 r. do 31 grudnia 2023 r. włącznie). Ponieważ są to dwie globalne firmy z tej samej branży, to przewidujemy, że ceny ich akcji będą od siebie zależne, gdyż są to bezpośredni konkurenci na międzynarodowym rynku.

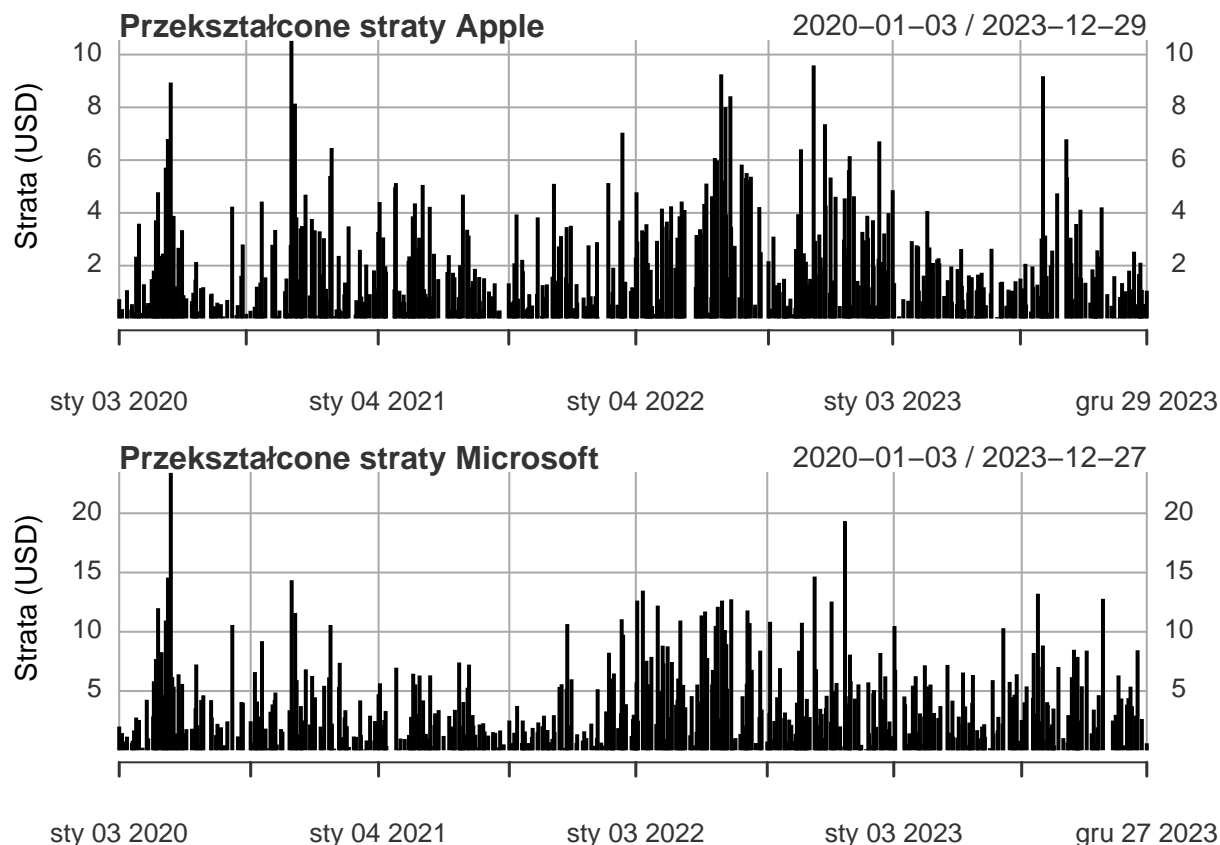
Wczytanie danych z pliku CSV:

Wizualizacja cen zamknięcia akcji:



Dla obu firm widziwmy powolny wzrost wartości akcji na przestrzeni wybranych lat. Obie firmy jendak po drodze notowały mniejsze lub większe spadki. Na pierwszy rzuk oka oba wykresy są do siebie bardzo podobne, oba mają bardzo zbliżone okresu wzrostu i spadku wartości. Zatem oczekujemy dosyć dużej korelacji między tymi danymi.

Straty obliczamy jako różnice wartości cen zamknięcia między kolejnymi dniami. Dodatkowo, w przypadku gdy mamy do czynienia z zyskami a nie stratami, to usuwamy te wartości, gdyż nie one są celem analizy. Dodatkowo, przekształcamy ujemne wartości start na wartości dodanie, by ułatwić dalsze analizy i obliczenia.



Na wykresach strat można zaobserwować o wiele mniej podobieństw niż w oryginalnych wykresach danych. Jest to najprawdopodobniej efektem podejścia, jakie przyjęliśmy o eliminacji wartości nieujemnych, co oznacza duże zmniejszenie ilości wartości.

KROK 2

Obliczenie podstawowych statystyk:

```
##      Index      AAPL.Close
## Min.   :2020-01-03  Min.   : 0.01001
## 1st Qu.:2021-01-13  1st Qu.: 0.63000
## Median :2022-01-18  Median : 1.40999
## Mean   :2021-12-31  Mean    : 1.96422
## 3rd Qu.:2022-12-11  3rd Qu.: 2.84625
## Max.   :2023-12-29  Max.    :10.52000
```

```
##      Index      MSFT.Close
## Min.   :2020-01-03  Min.    : 0.009979
## 1st Qu.:2021-02-16  1st Qu.: 1.160004
## Median :2022-02-03  Median : 2.589996
## Mean   :2022-01-11  Mean    : 3.586960
## 3rd Qu.:2022-12-15  3rd Qu.: 5.199982
## Max.   :2023-12-27  Max.    :23.410004
```

Widać znacząco różnice między maksymalnymi wartościami, co może wynikać z faktu, że ceny akcji Microsoft są średnio większe od cen akcji Apple. Mediany i średnie obu rozkładów są różne, co również może być

spowodowane przez różnice w średniej cenie na korzyść Microsoft. W obu szeregach strat mediany są mniejsze od średnich, co oznacza że oba rozkłady są prawoskośne, czego oczekujemy od rozkładów strat. Oba minima są powyżej 0, co oznacza, że poprzednie transformacje danych zadziałały tak jak chcieliśmy.

Problemem powyższych przekształceń jest fakt, że szeregi strat dla obu firm niekonieczne muszą być równej długości. Ponieważ naszym celem jest badanie zależności tych szeregów, należy zadbać by były one równej długości. W tym celu wyselekcjonujemy dane z dni, w których obie firmy zanotowały stratę, by te szeregi były jak najbardziej porównywalne, jednocześnie sprawdzając, czy nie usuniemy zbyt dużo danych.

```
## [1] "Długość nowych szeregów strat:"
```

```
## [1] 365
```

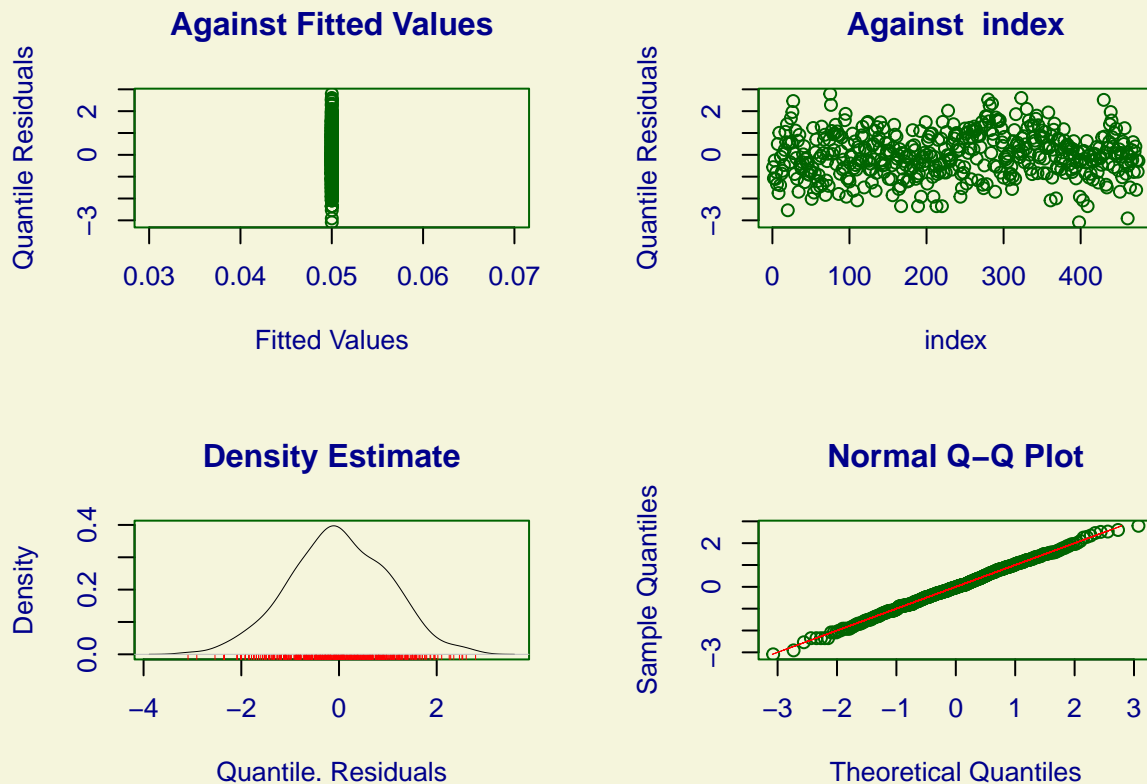
```
## [1] "Ilość straconych wartości dla dłuższego szeregu"
```

```
## [1] 112
```

Nowe, równe szeregi mają wystarczającą liczbę obserwacji, by móc przeprowadzać dla nich analizy.

Dopasowanie rozkładów:

```
##
## Family: c("SEP3", "skew exponential power type 3")
## Fitting method: "nlminb"
##
## Call: gamlssML(formula = y, family = DIST[i])
##
## Mu Coefficients:
## [1] 0.05
## Sigma Coefficients:
## [1] -2.129
## Nu Coefficients:
## [1] 2.332
## Tau Coefficients:
## [1] 0.1223
##
## Degrees of Freedom for the fit: 4 Residual Deg. of Freedom 471
## Global Deviance: 1582.86
## AIC: 1590.86
## SBC: 1607.52
```

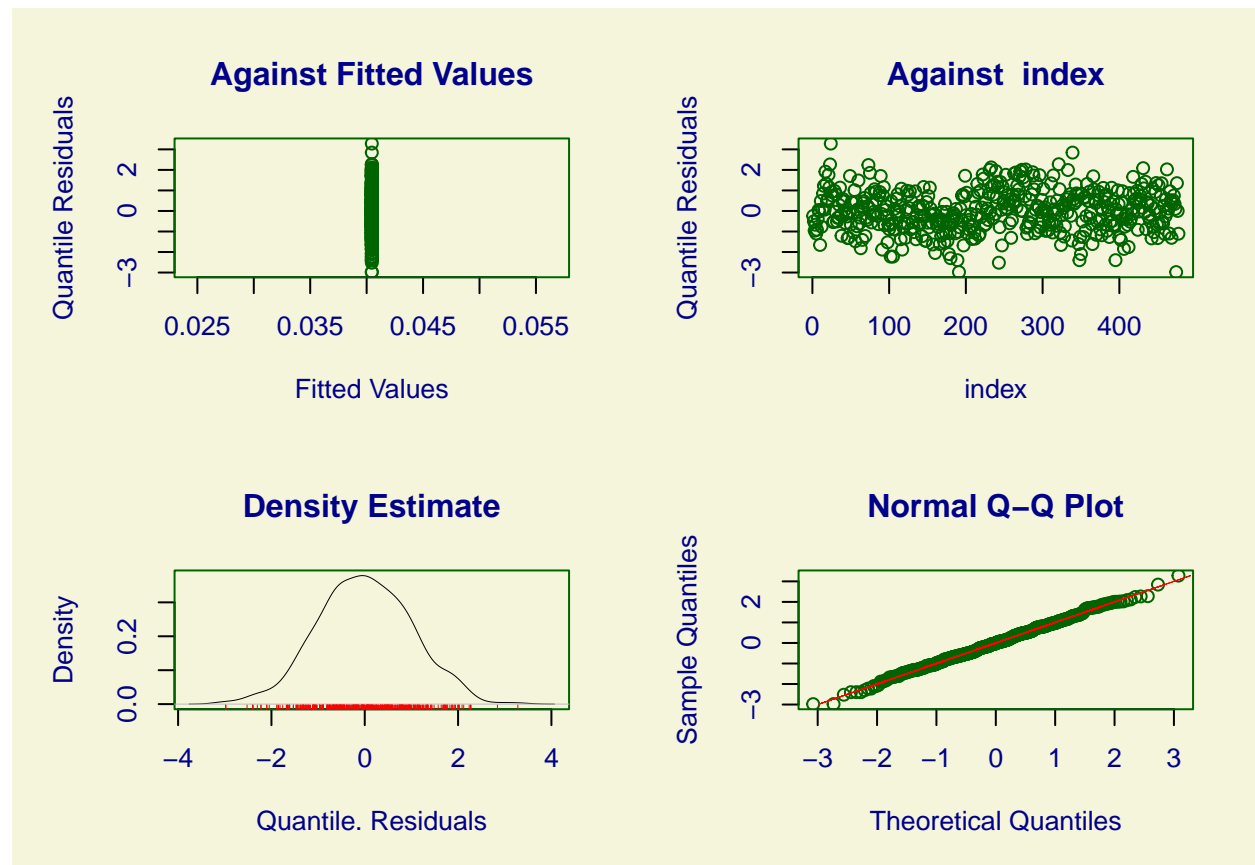


```
## *****
##      Summary of the Quantile Residuals
##              mean    = -0.005210106
##              variance = 1.022196
##              coef. of skewness = -0.04599694
##              coef. of kurtosis = 2.895815
## Filliben correlation coefficient = 0.9993622
## *****
```

Dopasowany rozkład do wartości strat dla firmy Apple to rozkład SEP3. Zwarte skupienie punktów na pierwszym wykresie sugeruje brak istotnej zależności reszt od wartości dopasowanych. Rozkład reszt względem indeksu wskazuje na losowość błędów, bez systematyczności czy struktury czasowej. Estymacja gęstości rozkładu reszt pokazuje, że reszty mają rozkład normalny. Porównując reszty z rozkładem normalnym, widać brak odchyień od linii prostej, co wskazuje że rozkład reszt faktycznie jest rozkładem normalnym.

```
##
## Family: c("SEP3", "skew exponential power type 3")
## Fitting method: "nlminb"
##
## Call: gamlssML(formula = y, family = DIST[i])
##
## Mu Coefficients:
## [1] 0.04046
## Sigma Coefficients:
## [1] -1.686
```

```
## Nu Coefficients:
## [1] 2.56
## Tau Coefficients:
## [1] 0.1556
##
## Degrees of Freedom for the fit: 4 Residual Deg. of Freedom 473
## Global Deviance: 2170.12
## AIC: 2178.12
## SBC: 2194.79
```



```
## *****
## Summary of the Quantile Residuals
## mean = 0.001531283
## variance = 0.9955553
## coef. of skewness = 0.01566117
## coef. of kurtosis = 2.989477
## Filliben correlation coefficient = 0.9990449
## *****
```

Dopasowany rozkład do wartości strat dla firmy Microsoft to ponownie rozkład SEP3, z lekko zmienionymi parametrami. Tak jak przy poprzednim dopasowaniu, zwarte skupienie punktów na pierwszym wykresie sugeruje brak istotnej zależności reszt od wartości dopasowanych a rozkład reszt względem indeksu sugeruje losowość błędów. Ponownie rozkład reszt w tym dopasowaniu jest rozkładem normalnym, co widać na estymacji gęstości oraz wykresie kwantyl-kwantyl.

Obliczenie korelacji dla przekształconych strat i pełnych danych cenowych:

```
## [1] "Współczynnik Pearsona"
```

```
##           MSFT.Close  
## AAPL.Close 0.1065511
```

```
## [1] "Współczynnik Spearmana"
```

```
##           MSFT.Close  
## AAPL.Close 0.1131697
```

```
## [1] "Współczynnik Kendalla"
```

```
##           MSFT.Close  
## AAPL.Close 0.07317955
```

Korelacja dla przekształconych strat jest niska, co sugeruje, że spadki cen nie mają ze sobą znaczącej zależności liniowej. Zgadza się to z wykresami reszt, które nie sprawiały wrażenia podobnych do siebie.

```
## [1] "Współczynnik Pearsona"
```

```
##           MSFT.Close  
## AAPL.Close 0.9344927
```

```
## [1] "Współczynnik Spearmana"
```

```
##           MSFT.Close  
## AAPL.Close 0.9333872
```

```
## [1] "Współczynnik Kendalla"
```

```
##           MSFT.Close  
## AAPL.Close 0.7944167
```

Korelacja dla cen zamknięcia jest wysoka, pomimo słabej korelacji między stratami. Zatem ceny akcji mają ze sobą znaczącą zależność liniową, tak jak przewidywaliśmy na podstawie wykresów wartości cen zamknięcia akcji. Fakt, że korelacja między stratami nie jest tak silna sugeruje, że przekształcenia którym poddaliśmy dane, były znaczące i mogły negatywnie wpłynąć na ich zależność.

KROK 3

Dopasowanie modeli kopuły:

```
##      AAPL.Close MSFT.Close  
## [1,] 0.2896175 0.4098361  
## [2,] 0.1584699 0.3333333  
## [3,] 0.4016393 0.1734973  
## [4,] 0.1393443 0.2527322  
## [5,] 0.2377049 0.1311475  
## [6,] 0.0942623 0.1803279
```

```
##      AAPL.Close      MSFT.Close
## Min.      :0.002732  Min.      :0.002732
## 1st Qu.:0.251366    1st Qu.:0.252732
## Median :0.500000    Median :0.498634
## Mean    :0.500000    Mean    :0.500000
## 3rd Qu.:0.748634    3rd Qu.:0.748634
## Max.    :0.997268    Max.    :0.997268
```

1. Gumbela:

```
## Call: fitCopula(gumbel_copula, data = data_matrix)
## Fit based on "maximum pseudo-likelihood" and 365 2-dimensional observations.
## Gumbel copula, dim. d = 2
##      Estimate Std. Error
## alpha      1.066      0.037
## The maximized loglikelihood is 1.947
## Optimization converged
## Number of loglikelihood evaluations:
## function gradient
##           6           6
```

2. Franka:

```
## Call: fitCopula(frank_copula, data = data_matrix)
## Fit based on "maximum pseudo-likelihood" and 365 2-dimensional observations.
## Frank copula, dim. d = 2
##      Estimate Std. Error
## alpha      0.6848      0.318
## The maximized loglikelihood is 2.341
## Optimization converged
## Number of loglikelihood evaluations:
## function gradient
##           4           4
```

3. Claytona:

```
## Call: fitCopula(clayton_copula, data = data_matrix)
## Fit based on "maximum pseudo-likelihood" and 365 2-dimensional observations.
## Clayton copula, dim. d = 2
##      Estimate Std. Error
## alpha      0.1579      0.066
## The maximized loglikelihood is 0.9489
## Optimization converged
## Number of loglikelihood evaluations:
## function gradient
##           3           3
```

4. Normalna:

```
## Call: fitCopula(normal_copula, data = data_matrix)
## Fit based on "maximum pseudo-likelihood" and 365 2-dimensional observations.
## Normal copula, dim. d = 2
```



```
##           Estimate Std. Error
## rho.1    0.1117      0.053
## The maximized loglikelihood is 2.165
## Optimization converged
## Number of loglikelihood evaluations:
## function gradient
##           6           6
```

5. T-studenta:

```
## Warning in fitCopula.ml(copula, u = data, method = method, start = start, :
## possible convergence problem: optim() gave code=1
```

```
## Call: fitCopula(t_copula, data = data_matrix)
## Fit based on "maximum pseudo-likelihood" and 365 2-dimensional observations.
## t-copula, dim. d = 2
##           Estimate Std. Error
## rho.1    0.1123      NA
## df       68.2830      NA
## The maximized loglikelihood is 2.14
## Convergence problems: code is 1 see ?optim.
## Number of loglikelihood evaluations:
## function gradient
##       1799       1000
```

```
## [1] "Kryterium loglikelihood"
```

```
## fit_gumbel  fit_frank fit_clayton  fit_normal  fit_t
##  1.9474551  2.3408684  0.9488852  2.1645868  2.1396616
```

```
## [1] "Kryterium AIC"
```

```
## fit_gumbel  fit_frank fit_clayton  fit_normal  fit_t
## -1.8949101 -2.6817367  0.1022295 -2.3291736 -0.2793232
```

```
## [1] "Kryterium BIC"
```

```
## fit_gumbel  fit_frank fit_clayton  fit_normal  fit_t
##  2.004987  1.218161  4.002127  1.570724  7.520472
```

Wyznaczenie najlepiej dopasowanej kopuły względem:

1. Kryterium loglikelihood(im większa wartość, tym lepiej dopasowany model):

```
## [1] "fit_frank"
```

```
## [1] 2.340868
```

2. Kryterium AIC(im mniejsza wartość, tym lepiej dopasowany model):

```
## [1] -2.681737
```

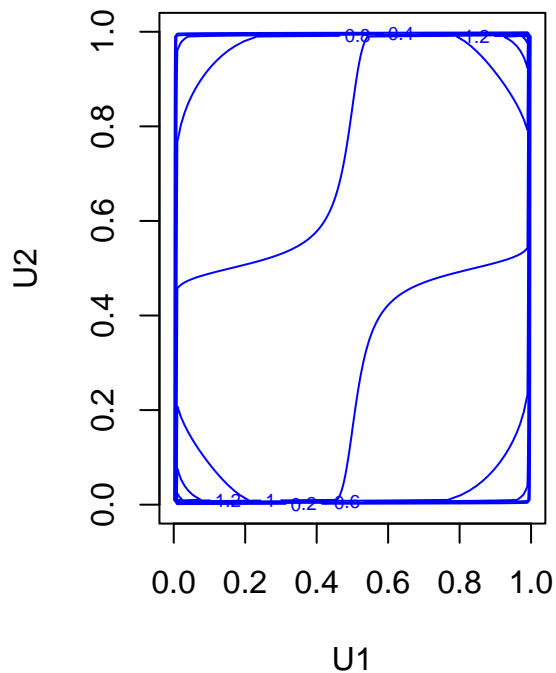
3. Kryterium BIC(im mniejsza wartość, tym lepiej dopasowany model):

```
## [1] "fit_frank"
```

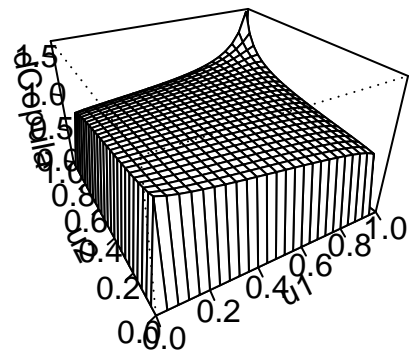
```
## [1] 1.218161
```

Wszystkie kryteria wybrały kopułę Franka jako tą najlepiej dopasowaną. Wizualizacja najlepiej dopasowanej kopuły:

Wykres konturowy kopuły



Dopasowana kopuła Franka



Wykonanie testu Mardia:

[illegible]

```
## 1      NO
## 2      NO
## 3      NO
##
## $univariateNormality
##           Test  Variable Statistic  p value Normality
## 1 Anderson-Darling AAPL.Close    13.4076 <0.001      NO
## 2 Anderson-Darling MSFT.Close    16.6632 <0.001      NO
##
## $Descriptives
##           n      Mean  Std.Dev  Median      Min  Max      25th      75th
## AAPL.Close 365 2.072726 1.901091 1.467499 0.040000916 10.52 0.6100006 3.140007
## MSFT.Close 365 3.606411 3.467953 2.509979 0.009979248 23.41 1.1499939 5.179993
##
##           Skew Kurtosis
## AAPL.Close 1.427688 2.254586
## MSFT.Close 1.755043 4.116692
```

Test Mardia wskazuje, że dane nie pochodzą z dwuwymiarowego rozkładu normalnego.

KROK 3

Obliczamy wartość zagrożoną (VaR) portfela składającego się z dwóch składników, korzystając zarówno z danych rzeczywistych, jak i danych wygenerowanych na podstawie najlepiej dopasowanej kopuły. Najpierw generowane są dane z dopasowanych rozkładów brzegowych, a później dane z kopuły. Następnie dla każdej próbki i poziomu istotności (α) obliczane są optymalne wagi (β), które minimalizują ryzyko portfela. Obliczenia są wykonywane na dwa sposoby: za pomocą funkcji optymalizującej (optimize) oraz ręcznie, za pomocą iteracji po wszystkich możliwych wartościach β z określoną odległością przedziału. Wszystko to, aby upewnić się, że wyniki są spójne i aby móc porównać ich dokładność w różnych sytuacjach.

```
## Wartości Beta (z funkcji optimize) z danych empirycznych dla N = 500 i Alpha = 0.95 wynosi 0.795718
## Wartości Beta (liczona ręcznie) z danych empirycznych dla N = 500 i Alpha = 0.95 wynosi 0.7957
## Wartości Beta (z funkcji optimize) z danych z kopuły dla N = 500 i Alpha = 0.95 wynosi 0.824281
## Wartości Beta (liczona ręcznie) z danych z kopuły dla N = 500 i Alpha = 0.95 wynosi 0.7486
##
## Wartości Beta (z funkcji optimize) z danych empirycznych dla N = 500 i Alpha = 0.99 wynosi 0.851854
## Wartości Beta (liczona ręcznie) z danych empirycznych dla N = 500 i Alpha = 0.99 wynosi 0.7847
## Wartości Beta (z funkcji optimize) z danych z kopuły dla N = 500 i Alpha = 0.99 wynosi 0.65289
## Wartości Beta (liczona ręcznie) z danych z kopuły dla N = 500 i Alpha = 0.99 wynosi 0.7982
##
## Wartości Beta (z funkcji optimize) z danych empirycznych dla N = 1000 i Alpha = 0.95 wynosi 0.766904
## Wartości Beta (liczona ręcznie) z danych empirycznych dla N = 1000 i Alpha = 0.95 wynosi 0.7669
## Wartości Beta (z funkcji optimize) z danych z kopuły dla N = 1000 i Alpha = 0.95 wynosi 0.821836
## Wartości Beta (liczona ręcznie) z danych z kopuły dla N = 1000 i Alpha = 0.95 wynosi 0.8449
##
## Wartości Beta (z funkcji optimize) z danych empirycznych dla N = 1000 i Alpha = 0.99 wynosi 0.759937
## Wartości Beta (liczona ręcznie) z danych empirycznych dla N = 1000 i Alpha = 0.99 wynosi 0.7599
## Wartości Beta (z funkcji optimize) z danych z kopuły dla N = 1000 i Alpha = 0.99 wynosi 0.785661
## Wartości Beta (liczona ręcznie) z danych z kopuły dla N = 1000 i Alpha = 0.99 wynosi 0.7857

## Wartości VaR (z funkcji optimize) z danych empirycznych dla N = 500 i Alpha = 0.95 wynosi 5.319906
## Wartości VaR (liczona ręcznie) z danych empirycznych dla N = 500 i Alpha = 0.95 wynosi 5.320001
## Wartości VaR (z funkcji optimize) z danych z kopuły dla N = 500 i Alpha = 0.95 wynosi 5.527406
```

```

## Wartości VaR (liczona ręcznie) z danych z kopuły dla N = 500 i Alpha = 0.95 wynosi 5.508017
##
## Wartości VaR (z funkcji optimize) z danych empirycznych dla N = 500 i Alpha = 0.99 wynosi 7.218653
## Wartości VaR (liczona ręcznie) z danych empirycznych dla N = 500 i Alpha = 0.99 wynosi 7.064982
## Wartości VaR (z funkcji optimize) z danych z kopuły dla N = 500 i Alpha = 0.99 wynosi 7.22447
## Wartości VaR (liczona ręcznie) z danych z kopuły dla N = 500 i Alpha = 0.99 wynosi 6.910709
##
## Wartości VaR (z funkcji optimize) z danych empirycznych dla N = 1000 i Alpha = 0.95 wynosi 5.378574
## Wartości VaR (liczona ręcznie) z danych empirycznych dla N = 1000 i Alpha = 0.95 wynosi 5.378599
## Wartości VaR (z funkcji optimize) z danych z kopuły dla N = 1000 i Alpha = 0.95 wynosi 5.436735
## Wartości VaR (liczona ręcznie) z danych z kopuły dla N = 1000 i Alpha = 0.95 wynosi 5.428515
##
## Wartości VaR (z funkcji optimize) z danych empirycznych dla N = 1000 i Alpha = 0.99 wynosi 7.334308
## Wartości VaR (liczona ręcznie) z danych empirycznych dla N = 1000 i Alpha = 0.99 wynosi 7.334353
## Wartości VaR (z funkcji optimize) z danych z kopuły dla N = 1000 i Alpha = 0.99 wynosi 7.505106
## Wartości VaR (liczona ręcznie) z danych z kopuły dla N = 1000 i Alpha = 0.99 wynosi 7.505109

```