

Analiza porównawcza cen akcji NVIDIA i AMD

Mackiewicz-Kubiak Aleksander, Pagielska Marta

2024-01-10

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
```

KROK 1

Do analizy wybrano ceny zamknięcia akcji NVIDIA i AMD z ostatnich 5 lat (począwszy od 3 stycznia 2019 r. do 31 grudnia 2024 r. włącznie). Obie firmy działają w branży technologicznej i zajmują się produkcją procesorów i układów graficznych.

Ze względu na podobny profil działalności i działanie na tych samych rynkach, można przypuszczać, że ich ceny akcji mogą być ze sobą powiązane. Celem analizy jest sprawdzenie tej zależności.

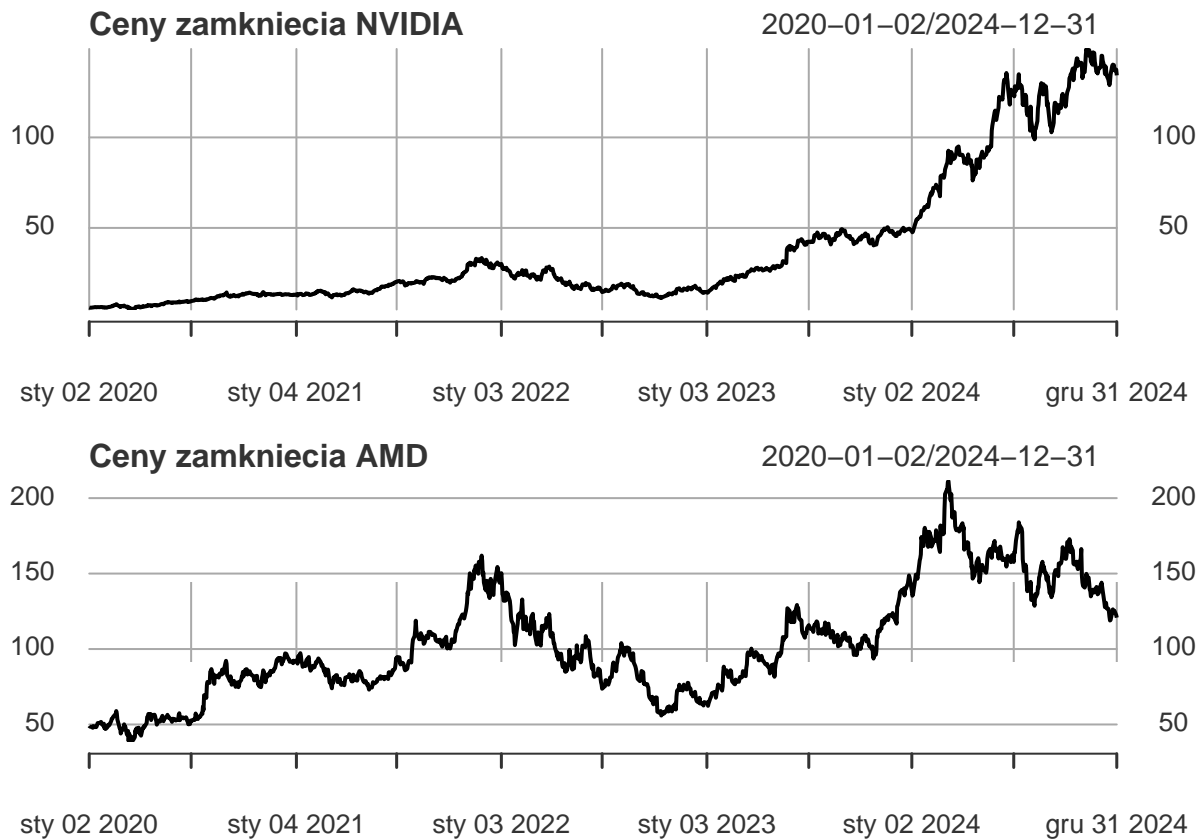
Wczytanie danych z pliku CSV, konwersja formatu, utworzenie ramki danych oraz obiektu xts dla szeregów czasowych, podgląd danych:

```
##          AMD.close
## 2020-01-02      49.10
## 2020-01-03      48.60
## 2020-01-06      48.39
## 2020-01-07      48.25
## 2020-01-08      47.83
## 2020-01-09      48.97

##          NVIDIA.close
## 2020-01-02         6.00
## 2020-01-03         5.90
```

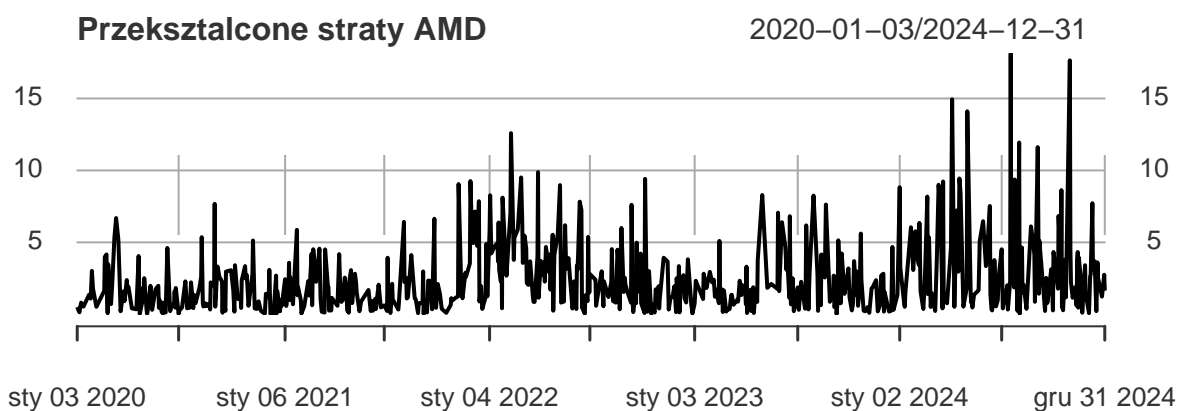
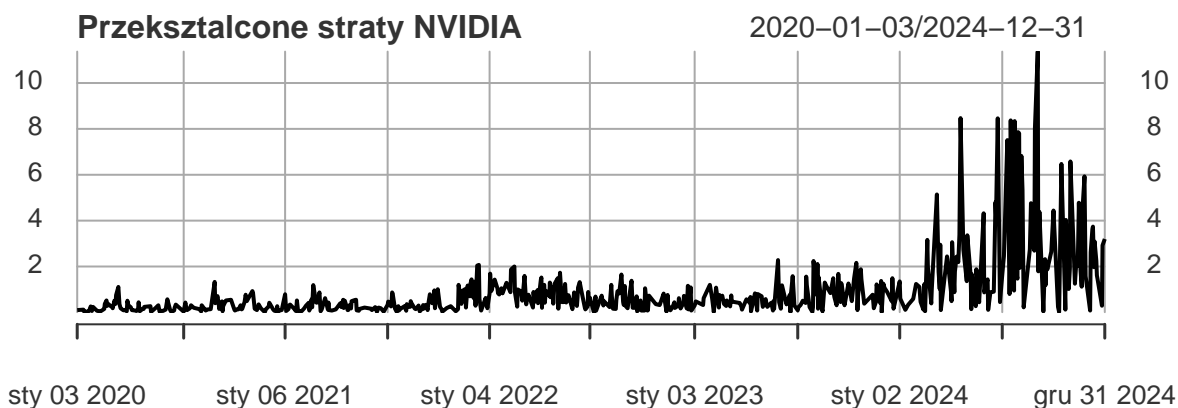
##	2020-01-06	5.93
##	2020-01-07	6.00
##	2020-01-08	6.01
##	2020-01-09	6.08

Wizualizacja historycznych cen zamknięcia akcji dla NVIDIA oraz AMD:



Ceny zamknięcia akcji NVIDIA wykazują wyraźny trend wzrostowy w analizowanym okresie. Szczególnie dynamiczny wzrost widoczny jest po roku 2022. Ceny akcji AMD mimo generalnej tendencji wzrostowej, wykazują większą zmienność. Wzrost do 2022 roku był znaczący, ale od tego momentu ceny zaczęły spadać, a same wahania były większe niż w przypadku NVIDIA.

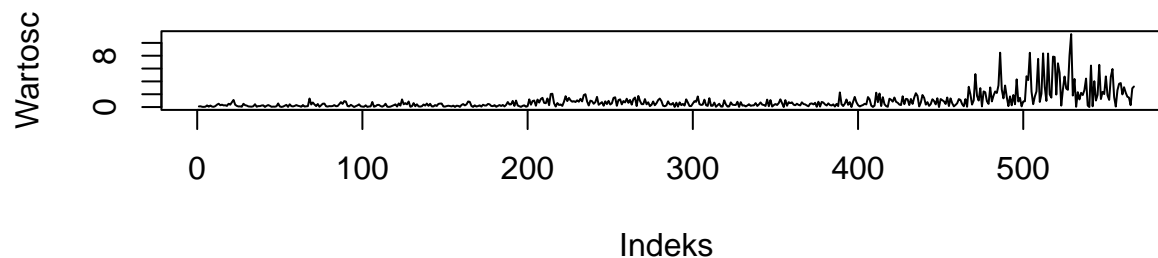
Transformacja danych poprzez obliczenie strat jako różnicy wartości zamknięcia między kolejnymi dniami oraz ich wizualizacja:



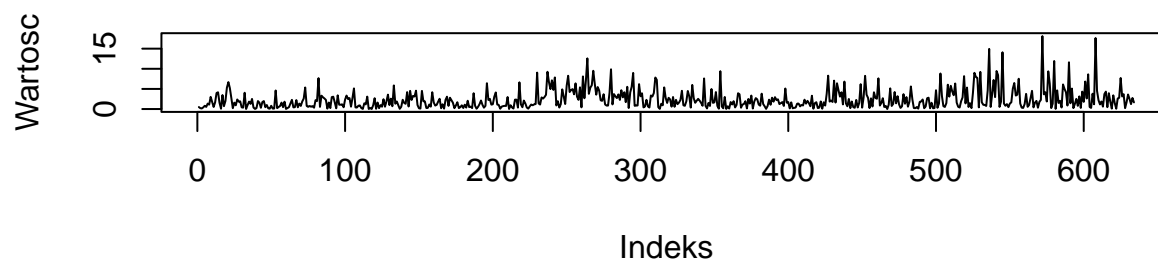
Wykres przekształconych strat dla NVIDIA ukazuje stabilny okres do 2022 roku, po czym wzrost zmienności staje się bardziej zauważalny. Szczególnie intensywne wartości odchyłeń widoczne są pod koniec 2024 roku. Wykres strat dla AMD jest bardziej dynamiczny niż dla NVIDIA, z wyraźnymi skokami w latach 2022-2024. Maksymalne straty osiągają wyższe wartości niż w przypadku NVIDIA.

KROK 2

Przekształcone straty NVIDIA



Przekształcone straty AMD



Obliczenie podstawowych statystyk:

```
##      Index      NVIDIA.close
##  Min.   :2020-01-03  Min.    : 0.0100
## 1st Qu.:2021-04-27  1st Qu.: 0.1800
## Median :2022-06-27  Median : 0.4300
## Mean   :2022-07-09  Mean    : 0.9154
## 3rd Qu.:2023-09-11  3rd Qu.: 1.0500
## Max.   :2024-12-31  Max.    :11.3700
```

```
##      NVIDIA.close
## średnia      0.9154497
## odchylenie   1.3948948
## wariancja    1.9457316
## moda        0.0800000
## braki        691.0000000
```

```
##      Index      AMD.close
##  Min.   :2020-01-03  Min.    : 0.0100
## 1st Qu.:2021-04-12  1st Qu.: 0.6675
## Median :2022-06-29  Median : 1.6100
## Mean   :2022-07-10  Mean    : 2.3800
## 3rd Qu.:2023-10-08  3rd Qu.: 3.1375
## Max.   :2024-12-31  Max.    :18.1200
```

```
##          AMD.close
## średnia      2.380032
## odchylenie   2.474408
## wariancja    6.122697
## moda         0.770000
## braki        624.000000
```

AMD cechuje się większym odchyleniem standardowym i większą wariancją niż NVIDIA. Zatem straty AMD są średnio większe i bardziej zmienne, co wskazuje na większe ryzyko związane z akcjami AMD.

Problemem powyższych przekształceń jest fakt, że szeregi strat dla obu firm niekoniecznie 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] 464
```

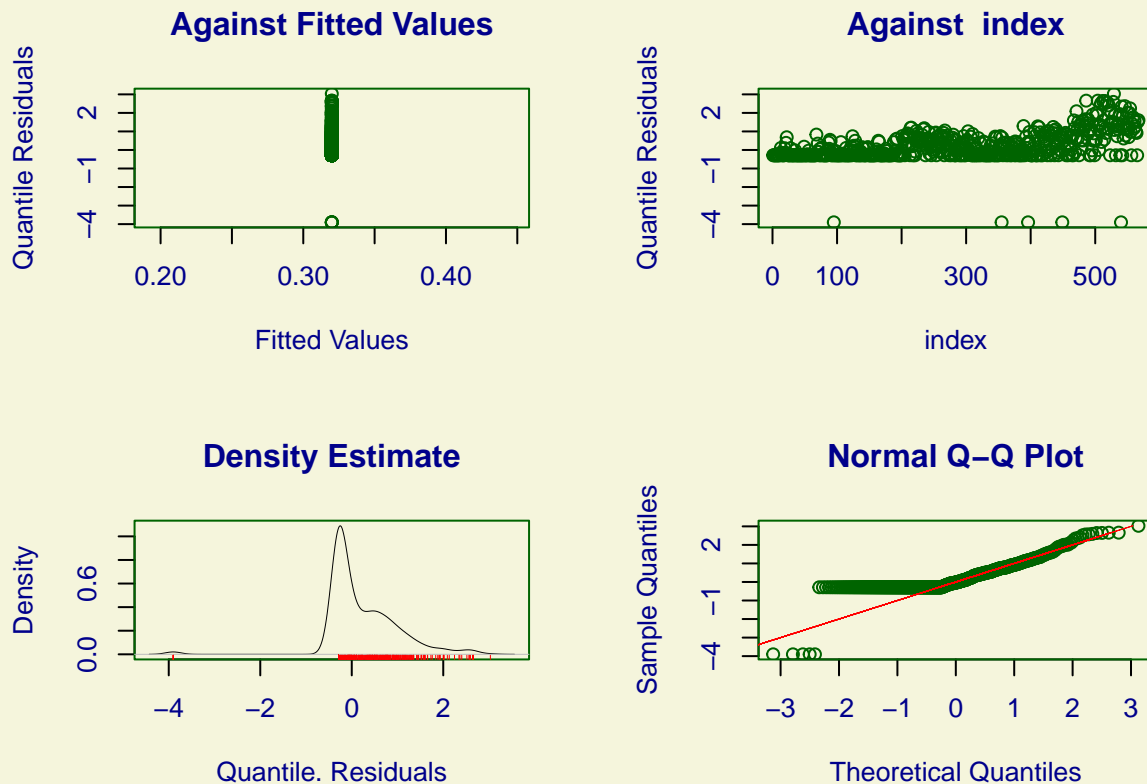
```
## [1] "Ilość straconych wartości dla dłuższego szeregu"
```

```
## [1] 170
```

Nowe, równe szeregi mają wciąż dużą liczbę obserwacji, więc możemy przeprowadzać dla nich analizy.

Dopasowanie rozkładów: Dopasowano rozkłady SEP4 (dla NVIDIA) oraz SEP3 (dla AMD). Wyniki wskazują na pewne problemy z dopasowaniem (np. ostrzeżenia o zbieżności). Mimo to, modele opisują asymetrię i różnorodność danych.

```
##
## Family: c("SEP4", "skew exponential power type 4")
## Fitting method: "nlminb"
##
## Call:  gamlssML(formula = y, family = DIST[i])
##
## Mu Coefficients:
## [1] 0.32
## Sigma Coefficients:
## [1] -1.171
## Nu Coefficients:
## [1] 10.9
## Tau Coefficients:
## [1] -0.5476
##
## Degrees of Freedom for the fit: 4 Residual Deg. of Freedom 563
## Global Deviance:      927.415
##           AIC:        935.415
##           SBC:        952.777
```

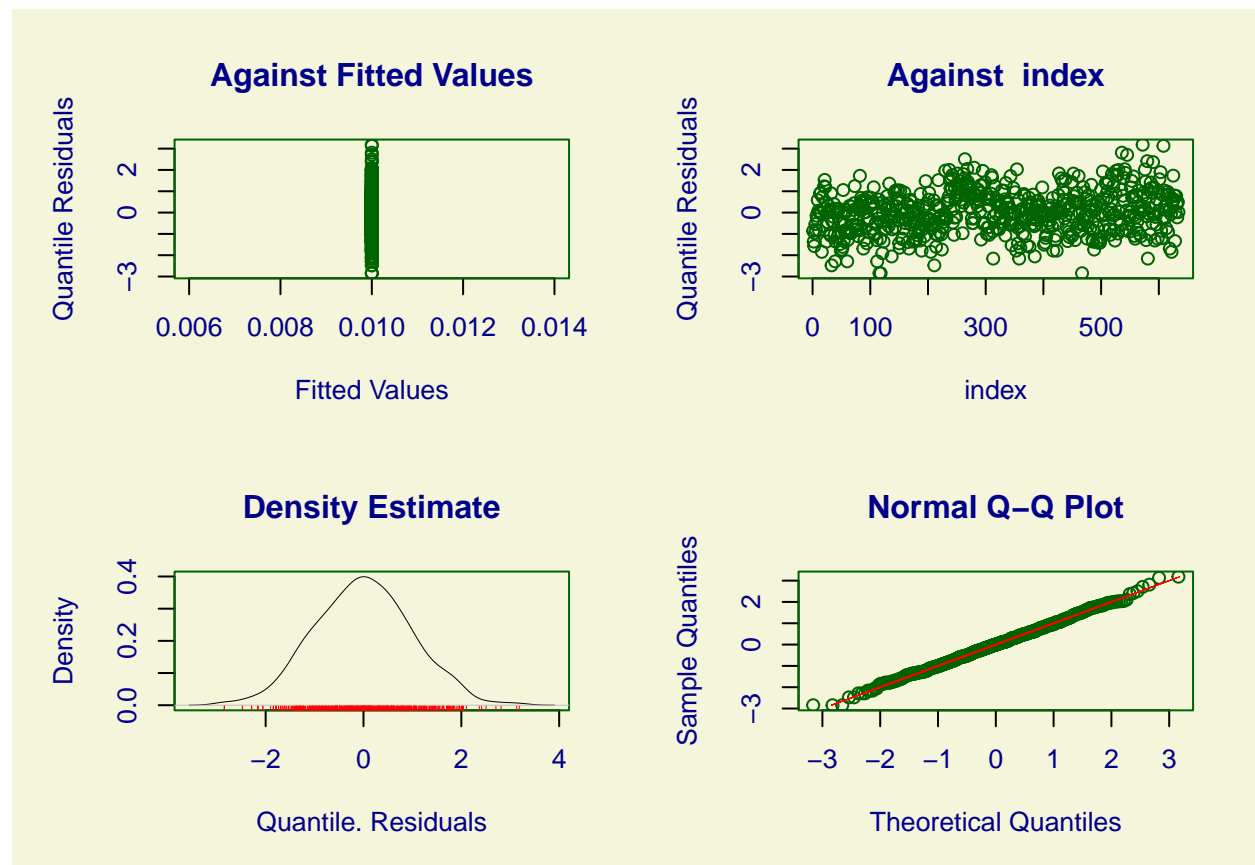


```
## *****
##      Summary of the Quantile Residuals
##              mean      = 0.2276081
##              variance   = 0.6188026
##              coef. of skewness = -0.2505458
##              coef. of kurtosis  = 9.541276
## Filliben correlation coefficient = 0.8877951
## *****
```

Rozrzut reszt wskazuje na pewne odchylenia, ale większość punktów skupia się blisko wartości dopasowanych, co sugeruje całkiem niezłe dopasowanie modelu. Pewne niespójności modelu może sugerować nieregularne rozłożenie reszt w czasie. Odchylenia reszt od linii prostej na wykresie Q-Q sugerują nienormalność rozkładu, szczególnie w ogonach.

```
##
## Family: c("SEP3", "skew exponential power type 3")
## Fitting method: "nlminb"
##
## Call: gamlssML(formula = y, family = DIST[i])
##
## Mu Coefficients:
## [1] 0.01
## Sigma Coefficients:
## [1] -3.003
## Nu Coefficients:
```

```
## [1] 3.052
## Tau Coefficients:
## [1] -0.05623
##
## Degrees of Freedom for the fit: 4 Residual Deg. of Freedom 630
## Global Deviance: 2364.06
## AIC: 2372.06
## SBC: 2389.87
```



```
## *****
## Summary of the Quantile Residuals
## mean = 0.002525636
## variance = 0.9836075
## coef. of skewness = 0.06180783
## coef. of kurtosis = 3.005562
## Filliben correlation coefficient = 0.9993637
## *****
```

Reszty są skupione wokół bardzo wąskiego zakresu wartości dopasowanych (około 0.01), co wskazuje na dużą zgodność między modelem a danymi w tym zakresie. Rozkład jest dość równomierny. Wygląda na to, że reszty są losowe w czasie. W skrajnych wartościach następuje odchylenie od rozkładu normalnego.

Obliczenie korelacji dla przekształconych strat i pełnych danych cenowych:

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

```
##          AMD.close
## NVIDIA.close 0.1688146
```

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

```
##          AMD.close
## NVIDIA.close 0.1491721
```

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

```
##          AMD.close
## NVIDIA.close 0.1012372
```

Korelacja dla przekształconych strat jest niska, co sugeruje, że spadki cen nie mają ze sobą znaczącej zależności liniowej.

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

```
##          AMD.close
## NVIDIA.close 0.7826652
```

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

```
##          AMD.close
## NVIDIA.close 0.9083153
```

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

```
##          AMD.close
## NVIDIA.close 0.7256636
```

Mimo słabej korelacji między stratami, korelacja dla cen zamknięcia jest bardzo wysoka, co wskazuje na dodatnią zależność liniową trendów obu firm.

KROK 3

Dopasowanie modeli kopuły:

```
##          NVIDIA.close  AMD.close
## [1,]    0.14838710 0.20645161
## [2,]    0.18064516 0.08709677
## [3,]    0.04623656 0.06236559
## [4,]    0.03010753 0.17849462
## [5,]    0.07526882 0.30967742
## [6,]    0.38494624 0.21397849
```

```
##          NVIDIA.close          AMD.close
## Min.      :0.002151    Min.      :0.003226
## 1st Qu.:0.256989    1st Qu.:0.251075
## Median :0.500000    Median :0.501075
## Mean      :0.500000    Mean      :0.500000
## 3rd Qu.:0.749193    3rd Qu.:0.748925
## Max.      :0.997849    Max.      :0.997849
```


1. Gumbela:

```
## Call: fitCopula(gumbel_copula, data = data_matrix)
## Fit based on "maximum pseudo-likelihood" and 464 2-dimensional observations.
## Gumbel copula, dim. d = 2
##      Estimate Std. Error
## alpha    1.099      0.037
## The maximized loglikelihood is 5.165
## 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 464 2-dimensional observations.
## Frank copula, dim. d = 2
##      Estimate Std. Error
## alpha    0.9246      0.279
## The maximized loglikelihood is 5.303
## 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 464 2-dimensional observations.
## Clayton copula, dim. d = 2
##      Estimate Std. Error
## alpha    0.2253      0.056
## The maximized loglikelihood is 1.7
## 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 464 2-dimensional observations.
## Normal copula, dim. d = 2
##      Estimate Std. Error
## rho.1    0.1611      0.046
## The maximized loglikelihood is 5.823
## 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 464 2-dimensional observations.
## t-copula, dim. d = 2
##      Estimate Std. Error
## rho.1    0.1607         NA
## df      47.1586         NA
## The maximized loglikelihood is 5.837
## Convergence problems: code is 1 see ?optim.
## Number of loglikelihood evaluations:
## function gradient
##      1800      1000

## [1] "Kryterium loglikelihood"

##      fit_gumbel    fit_frank fit_clayton  fit_normal      fit_t
##      5.164634      5.302571      1.700120      5.822941      5.836801

## [1] "Kryterium AIC"

##      fit_gumbel    fit_frank fit_clayton  fit_normal      fit_t
##      -8.329267     -8.605142     -1.400241     -9.645882     -7.673601

## [1] "Kryterium BIC"

##      fit_gumbel    fit_frank fit_clayton  fit_normal      fit_t
##      -4.1893828    -4.4652571      2.7396440    -5.5059975      0.6061681
```

Wyznaczenie najlepiej dopasowanej kopuły względem:

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

```
## [1] "fit_t"

## [1] 5.836801
```

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

```
## [1] "fit_normal"

## [1] -9.645882
```

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

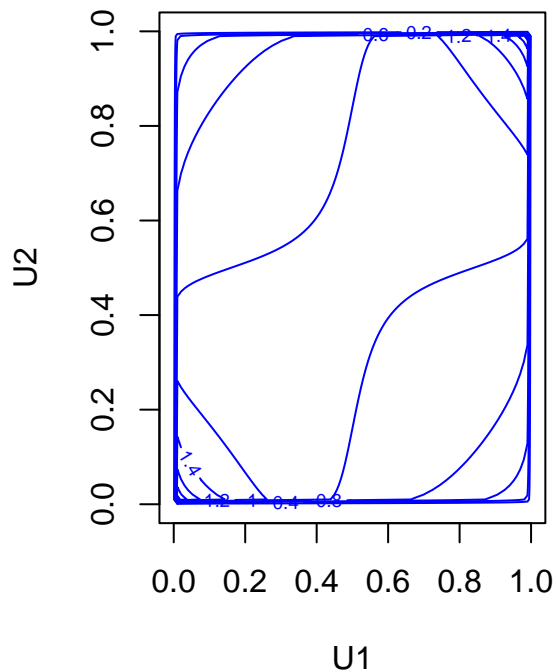
```
## [1] "fit_normal"

## [1] -5.505998
```

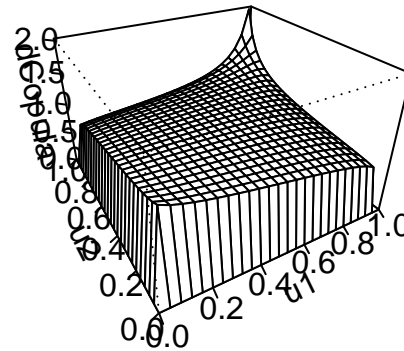
Wykonanie testu Mardia:

Wizualizacja najlepiej dopasowanej kopuły:

Wykres konturowy kopuły



Dopasowana kopuła Normalna



PUNKT 4 i 5

Obliczono 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 iteracyjnie, 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 0.920229
## Wartości Beta (liczona ręcznie) z danych empirycznych dla N = 500 i Alpha = 0.95 wynosi 0.9203
## Wartości Beta (z funkcji optimize) z danych z kopuły dla N = 500 i Alpha = 0.95 wynosi 0.759244
## Wartości Beta (liczona ręcznie) z danych z kopuły dla N = 500 i Alpha = 0.95 wynosi 0.7593
##
## Wartości Beta (z funkcji optimize) z danych empirycznych dla N = 500 i Alpha = 0.99 wynosi 0.778569
## Wartości Beta (liczona ręcznie) z danych empirycznych dla N = 500 i Alpha = 0.99 wynosi 0.7786
## Wartości Beta (z funkcji optimize) z danych z kopuły dla N = 500 i Alpha = 0.99 wynosi 0.756569
## Wartości Beta (liczona ręcznie) z danych z kopuły dla N = 500 i Alpha = 0.99 wynosi 0.7566
##
## Wartości Beta (z funkcji optimize) z danych empirycznych dla N = 1000 i Alpha = 0.95 wynosi 0.83027
## Wartości Beta (liczona ręcznie) z danych empirycznych dla N = 1000 i Alpha = 0.95 wynosi 0.9038
## Wartości Beta (z funkcji optimize) z danych z kopuły dla N = 1000 i Alpha = 0.95 wynosi 0.934338
## Wartości Beta (liczona ręcznie) z danych z kopuły dla N = 1000 i Alpha = 0.95 wynosi 0.9343
##
## Wartości Beta (z funkcji optimize) z danych empirycznych dla N = 1000 i Alpha = 0.99 wynosi 0.73006
```

```

## Wartości Beta (liczona ręcznie) z danych empirycznych dla N = 1000 i Alpha = 0.99 wynosi 0.7301
## Wartości Beta (z funkcji optimize) z danych z kopuły dla N = 1000 i Alpha = 0.99 wynosi 0.85057
## Wartości Beta (liczona ręcznie) z danych z kopuły dla N = 1000 i Alpha = 0.99 wynosi 0.8506

## Wartości VaR (z funkcji optimize) z danych empirycznych dla N = 500 i Alpha = 0.95 wynosi 3.236509
## Wartości VaR (liczona ręcznie) z danych empirycznych dla N = 500 i Alpha = 0.95 wynosi 3.236581
## Wartości VaR (z funkcji optimize) z danych z kopuły dla N = 500 i Alpha = 0.95 wynosi 3.457147
## Wartości VaR (liczona ręcznie) z danych z kopuły dla N = 500 i Alpha = 0.95 wynosi 3.457228
##
## Wartości VaR (z funkcji optimize) z danych empirycznych dla N = 500 i Alpha = 0.99 wynosi 5.09049
## Wartości VaR (liczona ręcznie) z danych empirycznych dla N = 500 i Alpha = 0.99 wynosi 5.090654
## Wartości VaR (z funkcji optimize) z danych z kopuły dla N = 500 i Alpha = 0.99 wynosi 5.678041
## Wartości VaR (liczona ręcznie) z danych z kopuły dla N = 500 i Alpha = 0.99 wynosi 5.678116
##
## Wartości VaR (z funkcji optimize) z danych empirycznych dla N = 1000 i Alpha = 0.95 wynosi 3.210971
## Wartości VaR (liczona ręcznie) z danych empirycznych dla N = 1000 i Alpha = 0.95 wynosi 3.198281
## Wartości VaR (z funkcji optimize) z danych z kopuły dla N = 1000 i Alpha = 0.95 wynosi 3.086334
## Wartości VaR (liczona ręcznie) z danych z kopuły dla N = 1000 i Alpha = 0.95 wynosi 3.086493
##
## Wartości VaR (z funkcji optimize) z danych empirycznych dla N = 1000 i Alpha = 0.99 wynosi 5.465323
## Wartości VaR (liczona ręcznie) z danych empirycznych dla N = 1000 i Alpha = 0.99 wynosi 5.465584
## Wartości VaR (z funkcji optimize) z danych z kopuły dla N = 1000 i Alpha = 0.99 wynosi 5.647652
## Wartości VaR (liczona ręcznie) z danych z kopuły dla N = 1000 i Alpha = 0.99 wynosi 5.6478

```