Analiza porównanawcza cen akcji NVIDIA i AMD - badanie parametrów

Mackiewicz-Kubiak Aleksander, Pągielska Marta

2025-06-12

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) # wyłączenie notacji naukowej
set.seed(5463436)
```

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.

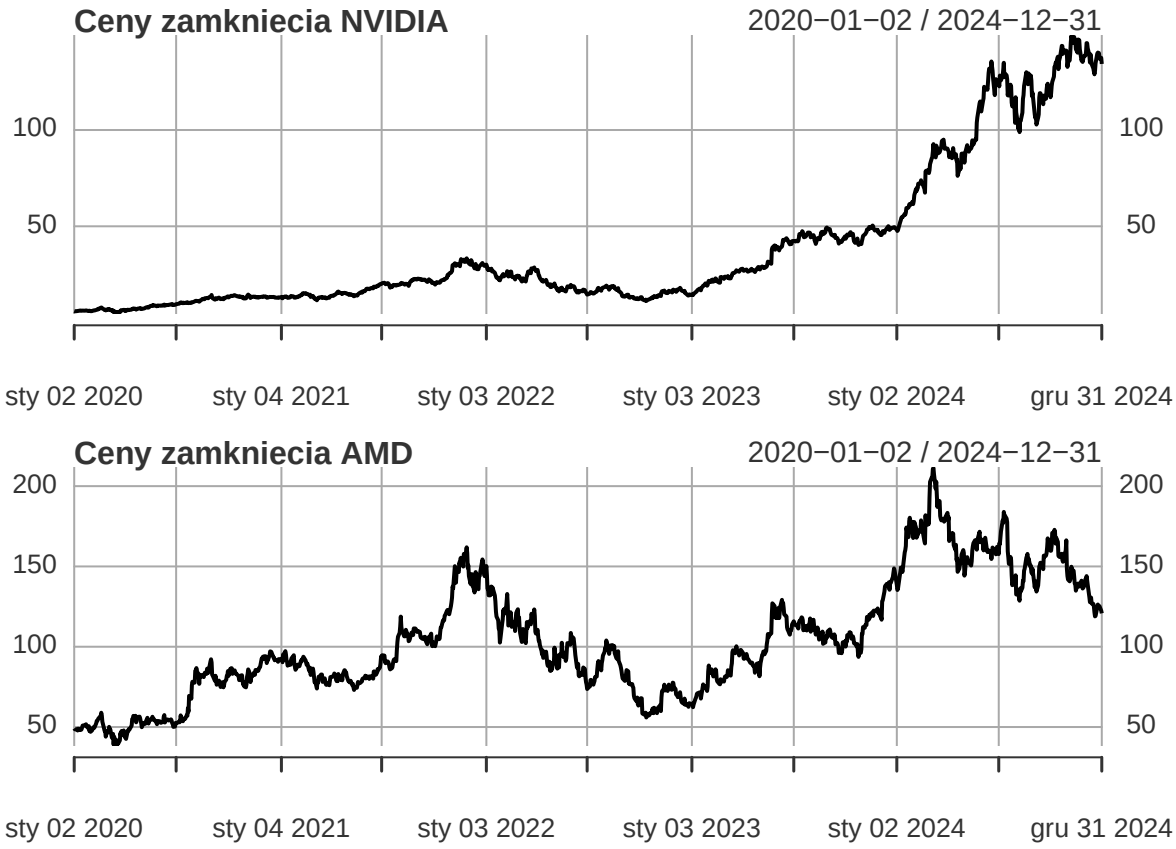
Ze względu na podobny profil działalności i działanie na tych samych rynkach, można przypuszczać, że ich ceny akcji mogą być ze sobą powiązane. Celem analizy jest sprawdzenie tej zależności.

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
```

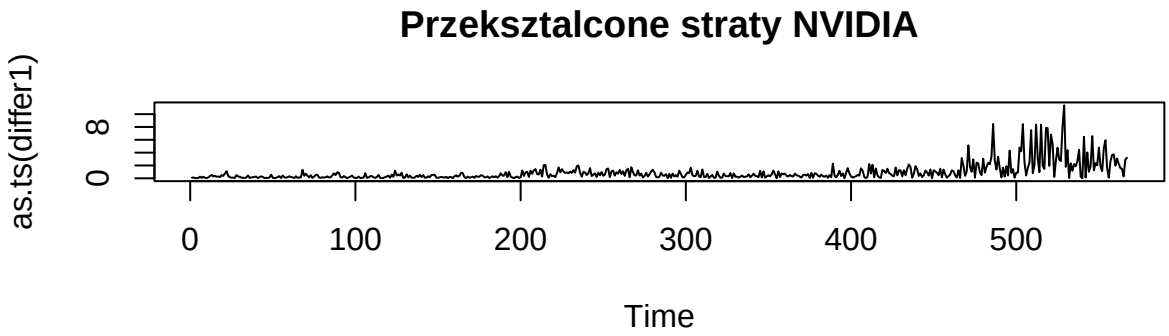
```
##            NVIDIA.close
## 2020-01-02         6.00
## 2020-01-03         5.90
## 2020-01-06         5.93
## 2020-01-07         6.00
## 2020-01-08         6.01
## 2020-01-09         6.08
```

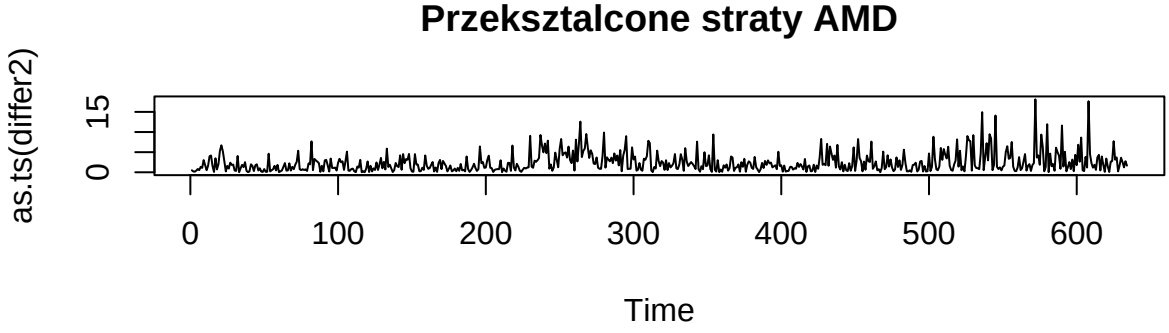
Wizualizacja historycznych cen zamknięcia akcji dla NVIDIA oraz AMD:



Ceny zamknięcia akcji NVIDIA wykazują wyraźny trend wzrostowy w analizowanym okresie. Szczególnie dynamiczny wzrost widoczny jest po roku 2022. Ceny akcji AMD mimo generalnej tendecji 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 odchyleń 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

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] 464
```

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

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

Nowe, równe szeregi mają wciąż dużą liczbe 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.

Przetworzenie danych, by móc do nich dopasować 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
```

KROK 3

Dopasowanie modeli kopuły:

Kryteria dla dopasowanych kopuł:

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

```
##  fit_gumbel   fit_frank fit_clayton  fit_normal       fit_t
##    5.164634    5.302571    1.700120    5.822941    5.836800
```

```
## [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   0.6061682
```

Wyznaczenie najlepiej dopasowanej kopuły według różnych kryteriów:

```
## [1] "fit_t"
```

```
## [1] 5.8368
```

```
## [1] "fit_normal"
```

```
## [1] -9.645882
```

```
## [1] "fit_normal"
```

```
## [1] -5.505998
```

Kryteria AIC oraz BIC wybrały kopułę normalną jako najlepiej dopasowaną kopułę, podczas gdy loglikelihood wskazało kopułę t. Byłoby to problemem, jednak jak spojrzymy na wszystkie wartości kryterium loglielihood, to zauważymy, że różnica między kopułą t a normalną jest marginalna. Zatem można założyć, że kryteria wskazują kopułę normalną jako najlepiej dopasowaną kopułę.

PUNKT 4 i 5

W dalszej części analizy przyjmujemy inne podejście niż dotychczas, ponnieważ zamiast skupiać się na pojedynczych wynikach lub konkretnej konfiguracji portfela, konstruujemy jedną zbiorczą ramkę danych results_df, która zawiera rezultaty uzyskane dla wielu portfeli i różnych parametrów. Każdy wiersz tej ramki odpowiada jednej kombinacji parametrów (np. typu portfela, sposobu generowania danych, metody optymalizacji), a kolumny zawierają najważniejsze statystyki opisujące wynik, w tym współczynnik Beta, wartość Alpha oraz typ danych wejściowych. Dzięki temu możemy podejść do analizy bardziej przekrojowo: badać rozkład współczynnika Beta wśród różnych podejść, identyfikować nietypowe przypadki (np. gdy Beta osiąga wartości graniczne), czy też porównywać skuteczność różnych metod modelowania ryzyka. Takie ujęcie pozwala nie tylko lepiej zrozumieć właściwości poszczególnych rozwiązań, ale także wychwycić ogólne trendy, nieoczywiste zależności oraz potencjalne błędy lub niespójności w analizie. W kolejnych krokach skoncentrujemy się na eksploracji tej ramki danych — rozpoczynając od oceny rozkładu współczynnika Beta, a następnie filtrując i porównując przypadki najbardziej charakterystyczne lub skrajne.

Nazwa ramki danych - results_df Parametry: Data-dane użyte do obliczeń Type-rodzaj funkcji użyty do obliczeń Betaseq-długość przedziałów wag między [0,1], dla których obliczamy najmniejszy VaR Alpha-poziom istotności wyliczanego VaRu N-liczba losowanych obserwacji w kopule/rozkładach VaR-wyliczone wartości Value at Risk Beta - optymalna waga, dla której wartość VaR była możliwie najmniejsza

Pętla generuje N-danych z dopasowanych rozkładów brzegowych (opdasowanych za pomocą fitDist) oraz z najlepiej dopasowanej kopuły (wybranej przy użyciu kryterium porównawczego AIC). Dla obu tych zbiorów danych używamy dwóch funkcji, wbudowanej w R funkcji optimize, i ręcznie napisanej compute_var, by dla konkretnego poziomu istotności (α) obliczyć dla jakiej wagi Beta z przedziału [0,1] wartość VaR jest najmniejsza. Funkcja optimize automatycznie ustala przedziały wartości Beta, a dla compute_var trzeba to ustalić ręcznie, za pomocą parametru Betaseq. Dla każdej wagi beta obliczamy VaR takiego portfela, a następnie wybieramy tylką tą wartość beta, dla której wartość VaR jest najmniejsza. Końcowo, tak obliczone wartości dodajemy do tabelki results_df, razem ze wszystkimi parametrami użytymi do ich wyliczenia. Dodawanie odbywa się czterokrotnie, gdyż korzystamy z dwóch rodzajóW danych i dwóch rodzajów funkcji, co daje cztery osobne wyniki w jednej iteracji pętli, każdy z różnych parametrami typu characteristic. Jako wisienka na torcie upewniamy się, że numeracja rzędów jest bazowa, dla estetyki i praktyczności tej tabeli. Łącznie pętla przechodzi 4x3x7=84 iteracje, co daje 336 operacji dodawania do tabeli, i sprawia że nasza tabela będzie miała wymiary 336x7.

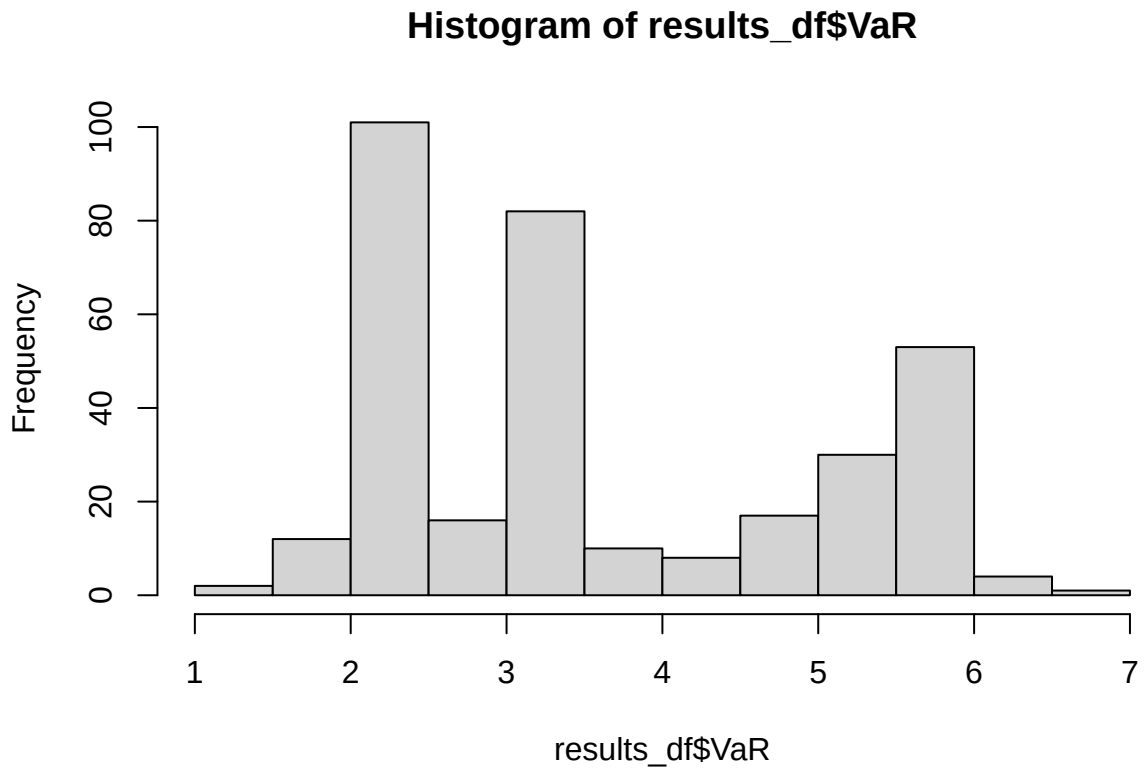
```
values_N <- c(50, 100, 500, 1000, 5000, 10000, 50000)
values_Alpha <- c(0.9, 0.95, 0.99)

lista1 <- c(seq(0, 1, by = 0.1))
lista2 <- c(seq(0, 1, by = 0.01))
lista3 <- c(seq(0, 1, by = 0.001))
lista4 <- c(seq(0, 1, by = 0.0001))
betaslist <- list(lista1, lista2, lista3, lista4)

results_df <- data.frame(Data = character(), Type=character(), Betaseq = numeric(),
                         Alpha = numeric(), N = numeric(), VaR = numeric(), Beta=numeric(), Difference =
```

W wyniku działania powyższej pętli tworzymy tabelkę danych results_df mającą 7 kolumn, które możemy "podzielić" na 3 typy. Pierwsze dwie kolumny (Data, Type) to kolumny binarne, zawierające informacje o danych i metodzie użytej do liczenia Varu. W moim przypadku mamy podział na dane z dopasowanych rozkładów empirycznych (Data=Empiric) i na dane z dopasowanej kopuły (Data=Copula), oraz na metoda liczenia za pomocą gotowej funkcji w R (Type=Optimize) i recznie napisanej analogicznej funkcji (Type=Own method). Kolejnym typem danych są parametry: N mówiący o liczbie próbek losowanych z dopasowanych rozkładów i kopuł, oraz Betaseq mówiący o długości przedziału wartości Beta dla których badaliśmy najmniejszy możliwy Var. Trzecim typem danych są dane które docelowo mieliśmy policzyć, Beta mówiący o wartości parametru dla którego minimalizujemy VaR dla naszego portfela akcji Nvidia

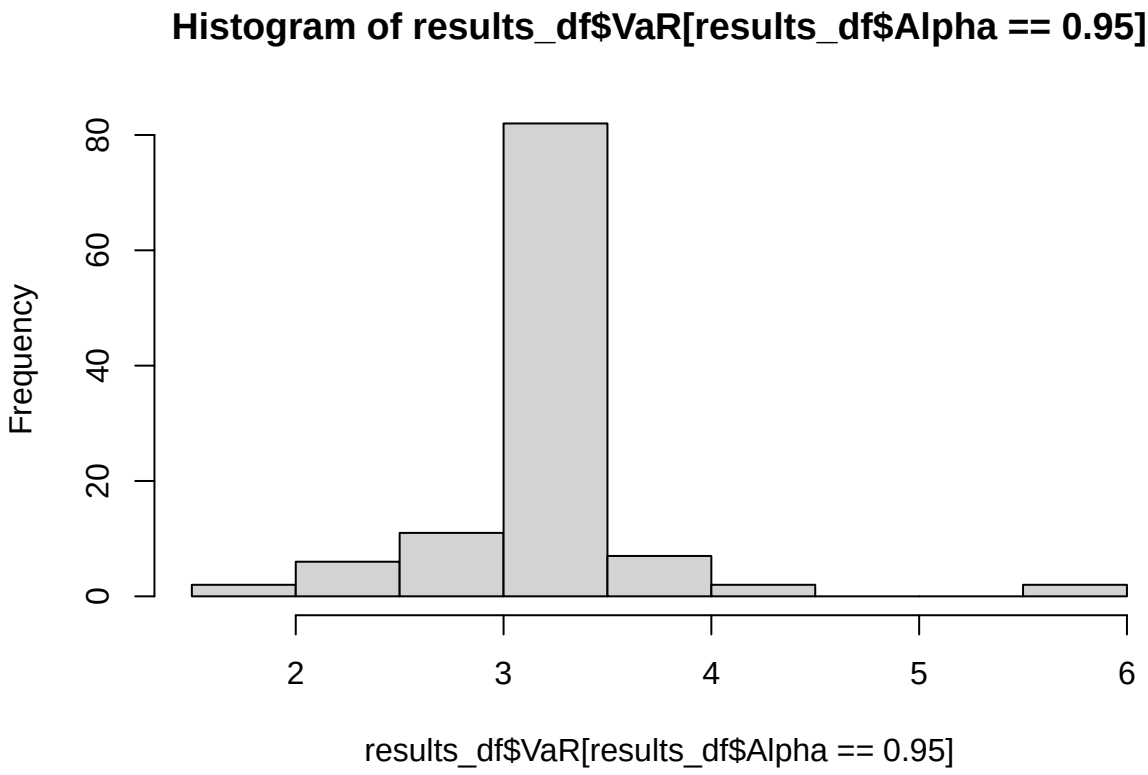
i AMD, i VaR podający wartość liczbową najmniejszego Varu. Do tego można dopisać wartości Alpha, które odpowiadają różnym poziomom VaRu (gdyż VaR możemy liczyć dla dowolnej Alphy z przedziału 0-1, którą interpretuje się jako prawdopobieństwo. Wartości Alpha, Beta i VaR są zatem bezpośrednio ze sobą powiązane,

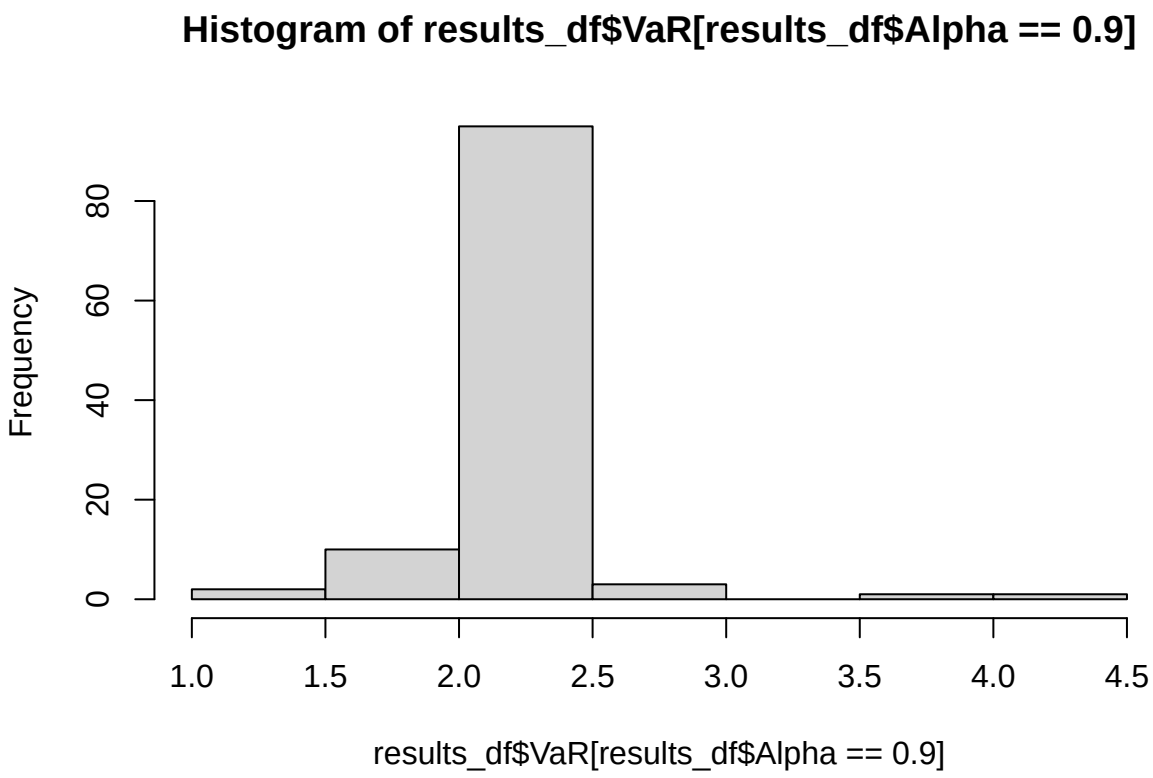
```
## [1] "Korelacja Beta-VaR"

## Pearsona: -0.8010392

## Spearmana: -0.7843724

## [1] "Korelacja Beta-Alpha"

## Pearsona: -0.7471286

## Spearmana: -0.7595479

## [1] "Korelacja Alpha-VaR"

## Pearsona: 0.896765

## Spearmana: 0.9053081
```

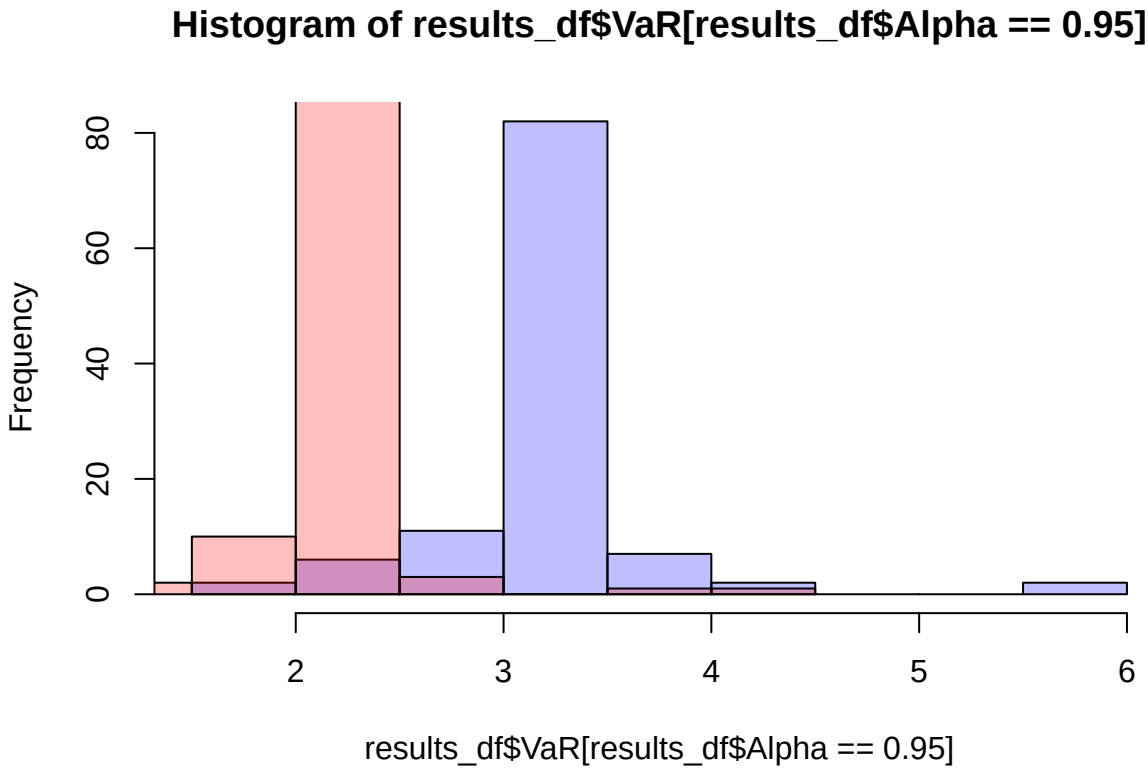
i ponieważ one były celem w oryginalnej analizie, to nim przyjrzę się najbardziej. Na pierwszy ogień bierzemy wartości VaRu. Wpierw histogram zebranych wartości:

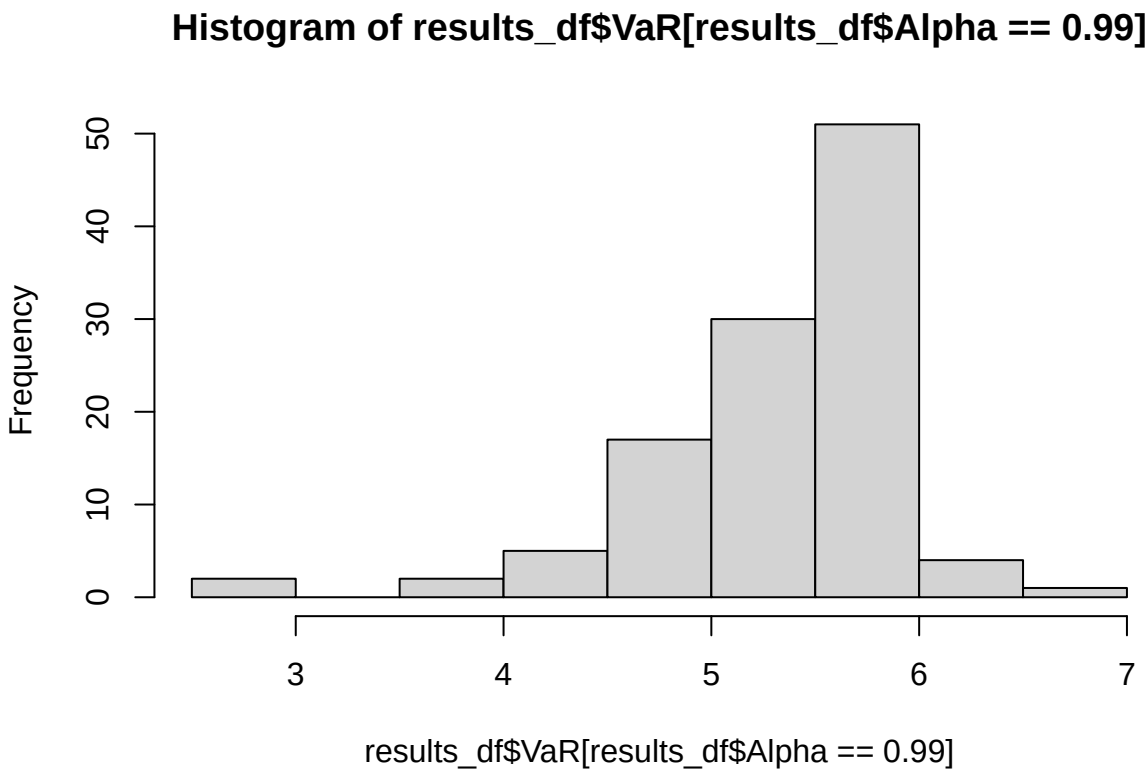


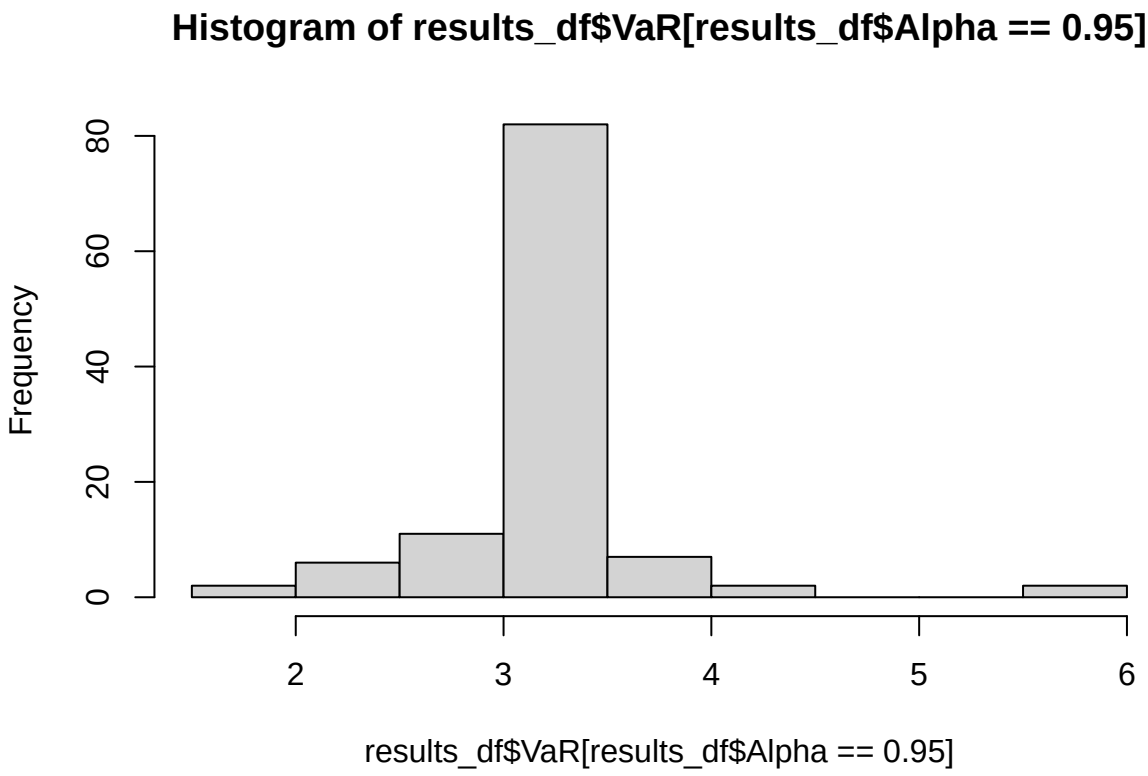
Rozkład ma wartości w przedziale 0 do 7. Jest on bardzo nieregularnie rozłożony, co może być kwestią małej liczby danych, lub dowodem na to, że te metody obliczeniowe nie były idealne. By to zweryfikować, sprawdzimy pewną zależność, a mianowicie czy dla różne poziomy Alpha wartości VaRu na siebie nie nachodzą.

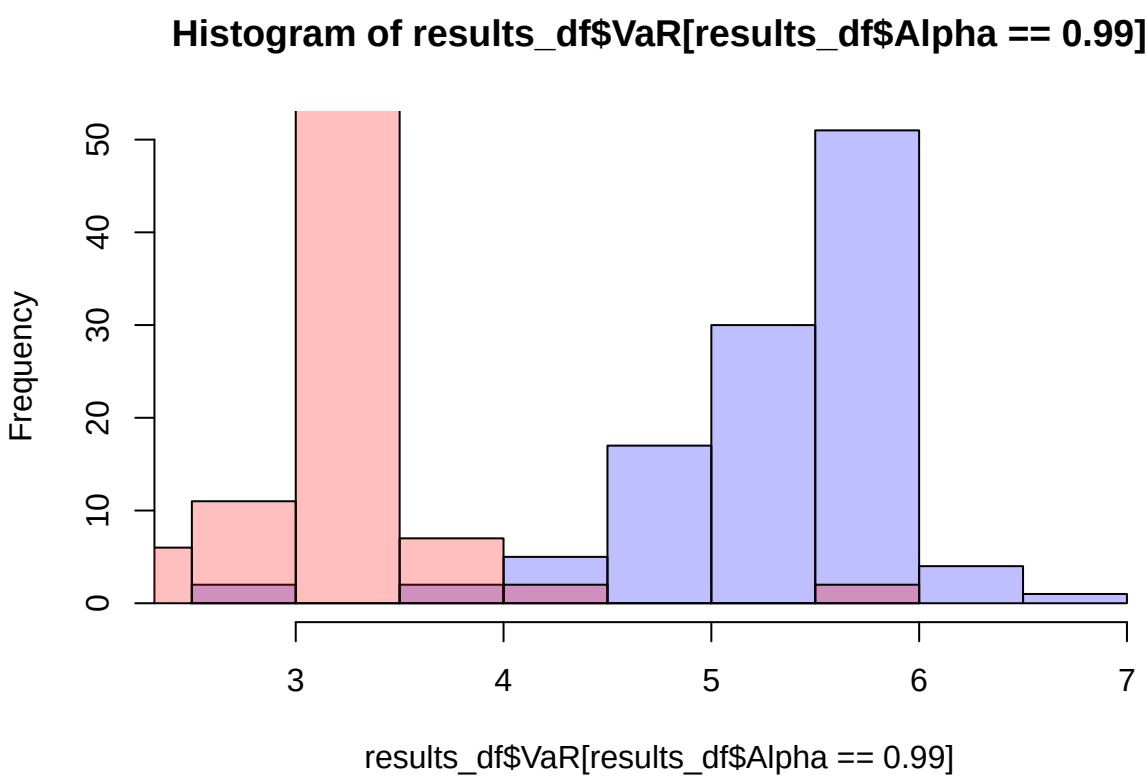



Histogram of results_df$VaR[results_df$Alpha == 0.9]
Frequency
0
20
40
60
80
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
results_df$VaR[results_df$Alpha == 0.9]


Histogram of results_df$VaR[results_df$Alpha == 0.95]
Frequency
0
20
40
60
80
2
3
4
5
6
results_df$VaR[results_df$Alpha == 0.95]


Histogram of results_df$VaR[results_df$Alpha == 0.99]
Frequency
0
10
20
30
40
50
3
4
5
6
7
results_df$VaR[results_df$Alpha == 0.99]


Histogram of results_df$VaR[results_df$Alpha == 0.95]
Frequency
0
20
40
60
80
2
3
4
5
6
results_df$VaR[results_df$Alpha == 0.95]



Wykresy na pierwszy rzut oka wyglądają jakby faktycznie dla wyższych poziomów Alpha Wartości VaRu były większe, wręcz przesunięte na wykresie o stałą wartość. Widać jednak, że są miejsca na wykresie na których wartości na siebie nachodzą, i to nawet niekoniecznie pomiędzy dominantami w tych wykresach. Sugeruje to, że nie wszystkie metody czy parametry zastosowane do obliczania VaRu były odpowiednie. Zobaczmy średnią wartość VaRu dla konkretnych użytych danych i funkcji.

```
## [1] 3.617038
```

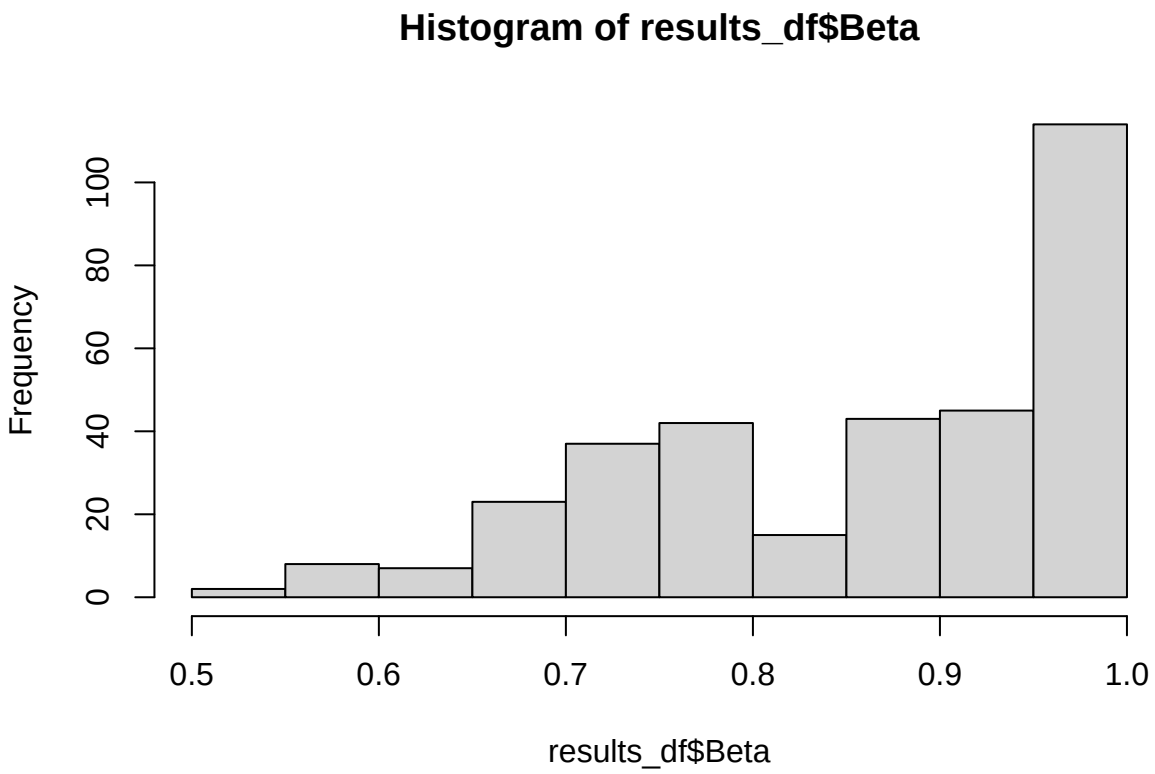
```
## [1] 3.588018
```

```
## [1] 3.575589
```

```
## [1] 3.629727
```

Wartości średnich są mocno zbliżone, co sugeruje, że wybór metody i danych użytych do obliczeń był poprawny. Zobaczmy, jak to wygląda dla wartości Beta.

Ponownie, wpierw zacznę od histogramu, pamiętając, że Beta może przyjmować wartości z przedziału [0,1]:



Rozkład jest mocno lewoskośny, z największym skupiskiem wartości przy możliwie maksymalnej wartości Beta równej 1. Co więcej wszystkie wartości Beta wydają się być powyżej wartości graniczej 0.5. Upewniamy się, że tak jest:

```
## [1] 0.5383
```

```
## [1] 0
```

Najmniejsza wartość Beta jest powyżej 0.5. Oznacza to, że w każdym wyniku jeden portfel, konkretniej AMD, jest bardziej stabliny przez co ma mniejszą wartość VaR. Zatem nawet jeżeli część wynikóW jest nieprecyzyjna, co można również wnioskować ze skośności rozkładu wartości Beta, to nie jest na tyle nieprecyzyjna by zaburzać informacje który portfel jest tym mniej ryzykownym, co waliduje wcześniejszą analize.

Przyjrzymy się teraz przypadkom, w którym Beta wyniosła jeden, gdyż w przypadku akcji Nvidii i AMD nie oczekiwalibyśmy tak mocnej polaryzacji w kierunku jednego portfela, stąd te wartości najszybciej rzucają się w oczy jako to potencjalnie błędne.

```
##       Data       Type Betaseq Alpha     N      VaR Beta
## 1  Empiric Own method  0.1000  0.90    50 1.327034    1
## 2   Copula Own method  0.1000  0.90    50 2.009819    1
## 3  Empiric Own method  0.1000  0.90   100 1.986201    1
## 4   Copula Own method  0.1000  0.90   100 2.036103    1
## 5  Empiric Own method  0.1000  0.90   500 2.466201    1
## 6   Copula Own method  0.1000  0.90   500 2.137851    1
## 7  Empiric Own method  0.1000  0.90  1000 2.228186    1
## 8   Copula Own method  0.1000  0.90  1000 2.133200    1
```

```
## 9  Empiric Own method  0.1000  0.90  5000 2.302011    1
## 10  Copula Own method  0.1000  0.90  5000 2.318871    1
## 11 Empiric Own method  0.1000  0.90 10000 2.261334    1
## 12  Copula Own method  0.1000  0.90 10000 2.249231    1
## 13 Empiric Own method  0.1000  0.90 50000 2.307864    1
## 14  Copula Own method  0.1000  0.90 50000 2.217402    1
## 15  Copula Own method  0.1000  0.95    50 1.686684    1
## 16 Empiric Own method  0.1000  0.95   100 3.102997    1
## 17  Copula Own method  0.1000  0.95 10000 3.256280    1
## 18  Copula Own method  0.1000  0.95 50000 3.355582    1
## 19 Empiric Own method  0.1000  0.99    50 4.874833    1
## 20 Empiric Own method  0.0100  0.90   100 2.255609    1
## 21 Empiric Own method  0.0100  0.90   500 2.160679    1
## 22  Copula Own method  0.0100  0.90  1000 2.316397    1
## 23 Empiric Own method  0.0100  0.90  5000 2.247789    1
## 24  Copula Own method  0.0100  0.90  5000 2.322625    1
## 25  Copula Own method  0.0100  0.90 10000 2.267067    1
## 26 Empiric Own method  0.0100  0.90 50000 2.234944    1
## 27  Copula Own method  0.0100  0.90 50000 2.252610    1
## 28  Copula Own method  0.0010  0.90   100 1.658035    1
## 29 Empiric Own method  0.0010  0.90  1000 2.345286    1
## 30 Empiric Own method  0.0010  0.90 50000 2.291492    1
## 31  Copula Own method  0.0010  0.90 50000 2.259638    1
## 32 Empiric Own method  0.0010  0.95    50 2.268030    1
## 33  Copula Own method  0.0010  0.99    50 2.690312    1
## 34 Empiric Own method  0.0001  0.90    50 2.287650    1
## 35 Empiric Own method  0.0001  0.90   100 2.360206    1
```

Można zaobserwować, że niektóre wartości w tej tabeli pojawiają się zdecydowanie częściej, a z kolei inne nie pojawiają sie ani razu. By to dokładniej zobaczyć, zrobimy osobne podziały dla każdej kolumny.

```
##         Type  n
## 1 Own method 35
```

```
##      Data  n
## 1  Copula 17
## 2 Empiric 18
```

```
##   Betaseq  n
## 1  0.0001  2
## 2  0.0010  6
## 3  0.0100  8
## 4  0.1000 19
```

```
##       N n
## 1    50 7
## 2   100 6
## 3   500 3
## 4  1000 4
## 5  5000 4
## 6 10000 4
## 7 50000 7
```

```
##   Alpha  n
## 1  0.90 28
## 2  0.95  5
## 3  0.99  2
```

Z powyższych list można zauważyc, że wartości Beta=1 zachodzą niezależnie od danych, których użyliśmy do liczenia VaRu, ale zachodzą tylko dla ręcznej metody liczenia. Co więcej, zachodzą one najczęściej dla najmniejszych przyjętych wartości Alpha i dla największej długości przedziału Betaseq, czyli bazując na wiedzy empirycznej, dla możliwie najmniej rzetelnych parametrów. Jest zatem podstawa, by sądzić że tylko dla odpowiednio dobranych Type, Alpha i Betaseq wartości są wiarygodne.

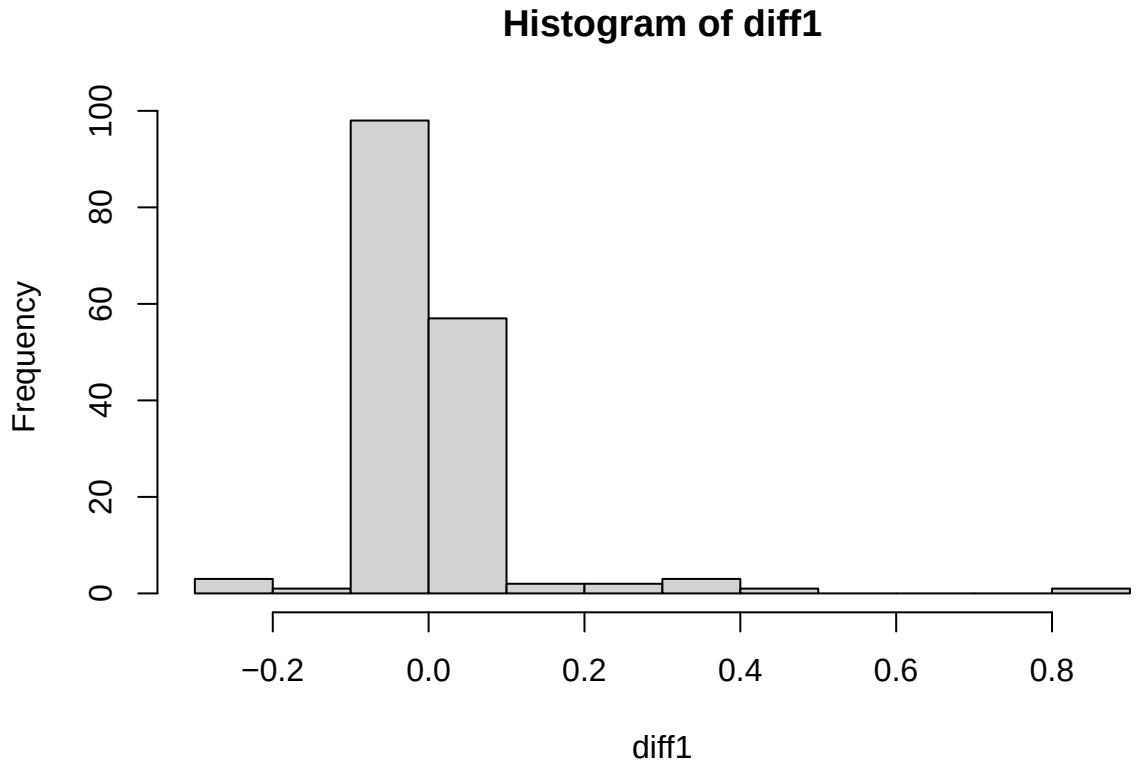
Możemy jeszcze przyjrzeć się różnicom wartości VaR dla różnych metod i parametrów. Wpierw porównamy reszty dla dwóch różnych funkcji, których użyliśmy:

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -0.296578 -0.007358 -0.000097  0.012559  0.002449  0.880835
```

```
## [1] 0.1047061
```

```
## [1] 0.01096338
```

```
## [1] -0.2965784
```



Wartości reszt wachają się między -0.3 a 0.9. Średnie wartości VaRu, które wcześniej sprawdzaliśmy, oscylowały w okolicach 3.5, zatem te reszty nie wydają się być duże. Dodatkowo, na histogramie widać bardzo dużą ilość reszt w okolicach wartości 0, co sugeruje, że obie funkcje dawały bardzo porównywalne wyniki.

Teraz porównany reszty wartości VaR dla dwóch różnych rodzajów danych, które stosowaliśmy do obliczeń, dane empiryczne i dane kopułowe:

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -2.06432 -0.25948 -0.05708 -0.04158  0.11800  4.11585
```

```
## [1] 0.7239637
```

```
## [1] 0.5241234
```

```
## [1] -2.064321
```

Różnice w tym przypadku są zdecydowanie większe, niż dla różnych metod. O ile ponownie większość reszt ma wartości bliskie zeru, tak istnieją wartości odstające przekraczające średnie wartości VaRu, co sugeruje, że różne dane w niektórych przypadkach dawały skrajnie różne wyniki.

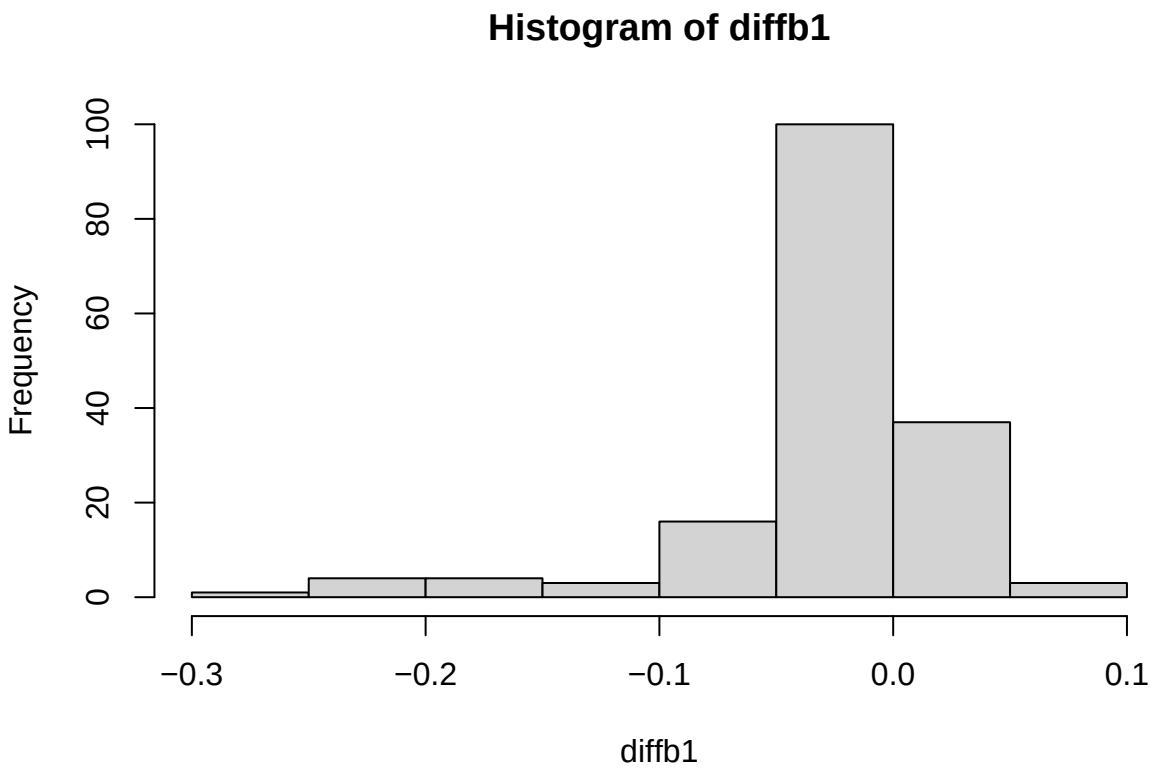
Możemy zastosować podobną analize dla różnich w wartościach Beta, mając na uwadze jej ograniczony przedział możliwych wartości [0,1].

```
##        Min.     1st Qu.      Median        Mean     3rd Qu.        Max.
## -0.26579713 -0.02757839 -0.00123717 -0.02283611 -0.00000474  0.09598868
```

```
## [1] 0.05529466
```

```
## [1] 0.003057499
```

```
## [1] -0.00006610696
```



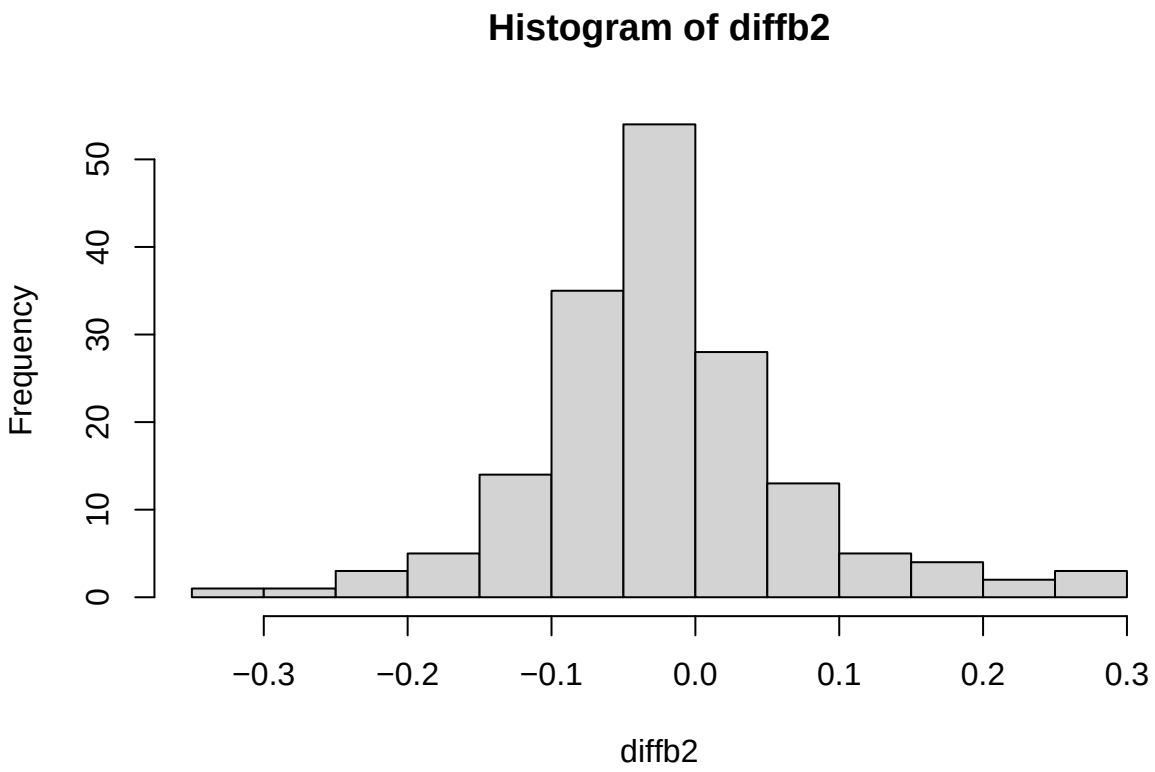
Rozkład jest lewoskośny, z dominującą liczbą wartości bliskich zero.

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.31124 -0.06910 -0.01035 -0.01501  0.02537  0.28880
```

```
## [1] 0.09536506
```

```
## [1] 0.009094495
```

```
## [1] 0
```



Rozkład wygląda na symetryczny, łudząco przypominającyw rozkład normalny.

Rozkład reszt wartości Beta dla dwóch różnych funkcji jest “ładniejszym” rozkładem, ale z większym przedziałem możliwych wartości, w porównaniu do tego badającego reszty dla różnych danych. Pokrywa się to z poprzednimi obserwacjami, gdzie tylko dla ręcznie napisanej funkcji pojawiały się wartości Beta=1, które mogły zaburzyć rozkład reszt i dodać mu lewoskośności.

Zatem końcowo można wysnuć wnioski, że: -wybór danych pomiędzy tymi z kopuły a tymi z rozkładów wpływał mocno na wartości VaR -wybór funkcji do obliczeń wpływał mocno na wartości Beta -dobór odpowiednich parametrów N, Betaseq i Alpha miał znaczenie: im większa Alpha i N tym bardziej wiarygodne wyniki, im mniejszy przedział Betaseq tym wiarygodniejsze wyniki