

ANALIZA PORÓWNAWCZA CEN AKCJI FIRM NVIDIA ORAZ AMD

Mackiewicz-Kubiak Aleksander
Pągielska Marta

Projekt zespołowy
17.01.2025



karty graficzne GeForce

wartość rynkowa przekroczyła
1 bilion dolarów!



karty graficzne Radeon

procesory Ryzen

ceny zamknięcia akcji



Dane historyczne

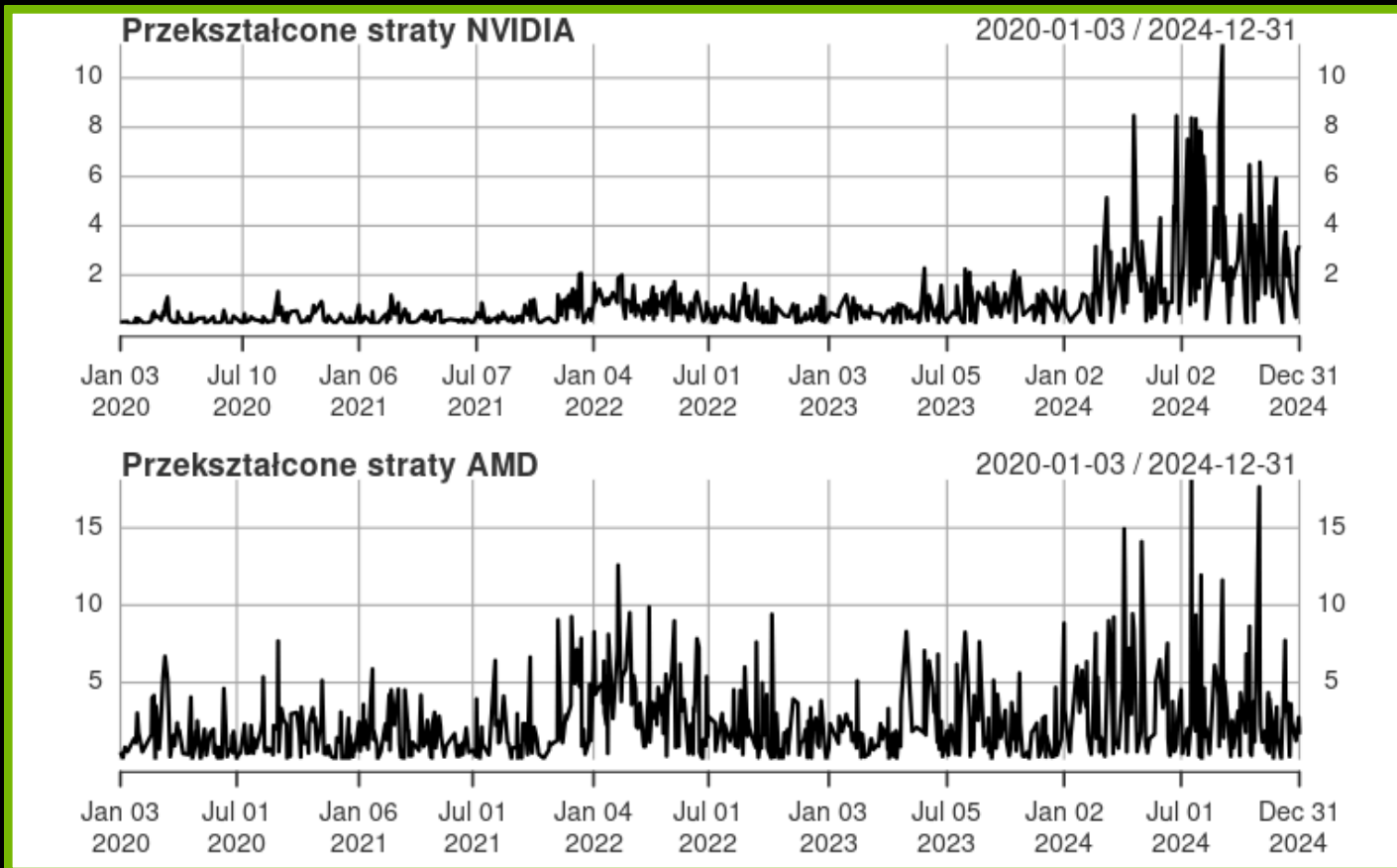


okres 2 stycznia 2019 - 31 grudnia 2024

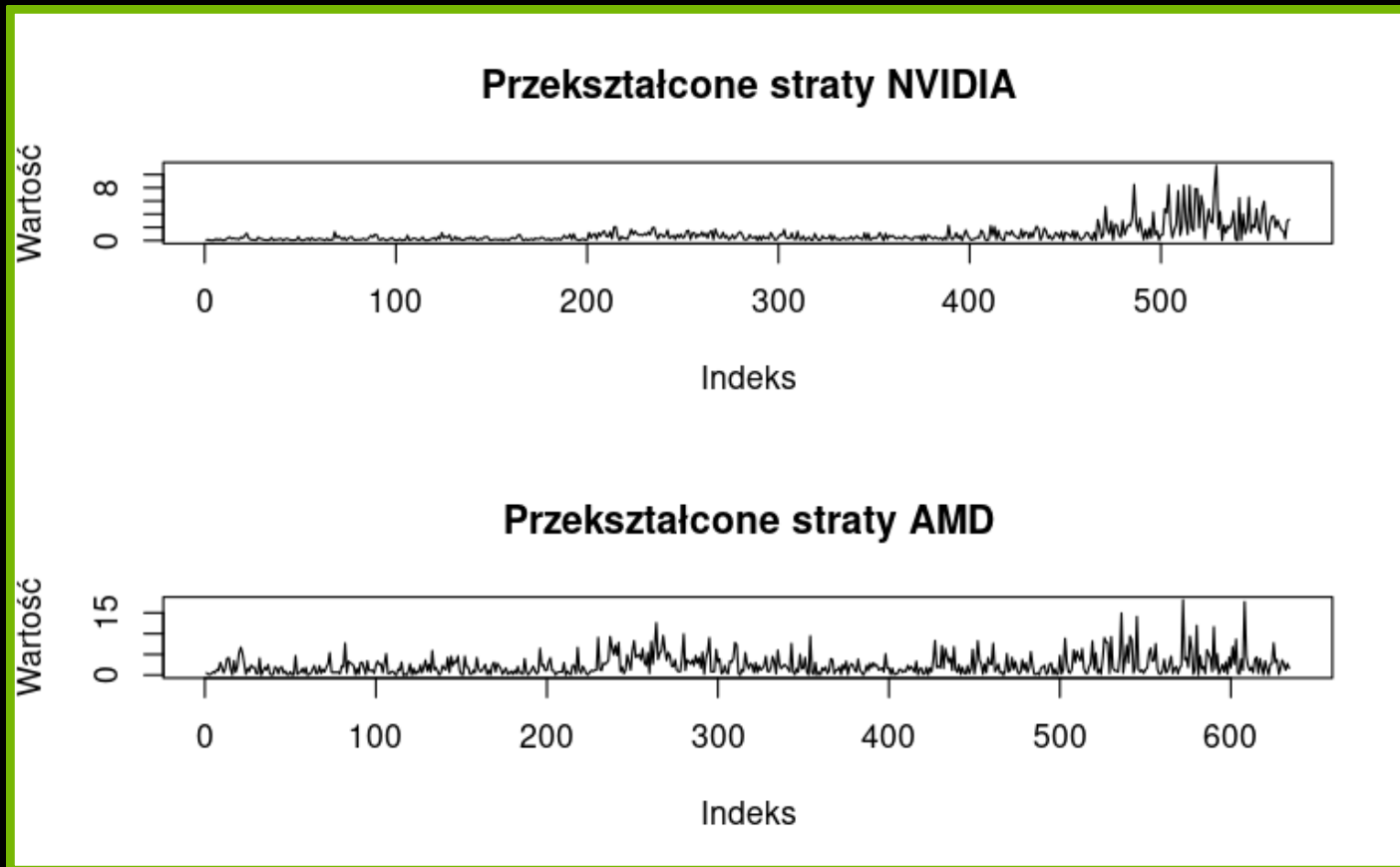
Dane historyczne



Dane transformowane



Dane transformowane



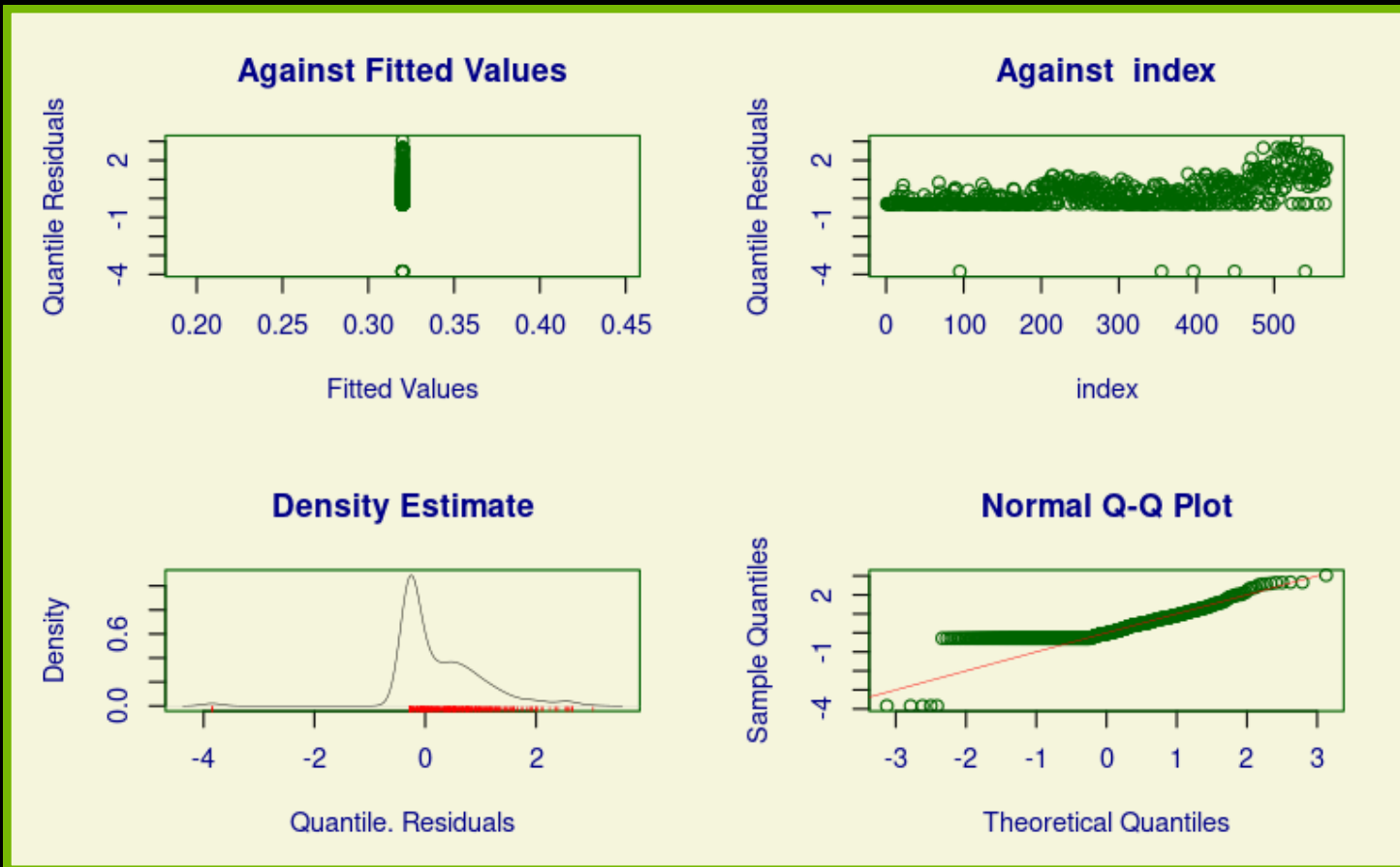
Podstawowe statystyki

Statystyka	NVIDIA (straty)	AMD (straty)
Średnia	0.92	2.38
Mediana	0.43	1.61
Odchylenie standardowe	1.39	2.47
Wariancja	1.95	6.12
Minimalna wartość	0.01	0.01
Maksymalna wartość	11.37	18.12
Moda	0.08	0.77

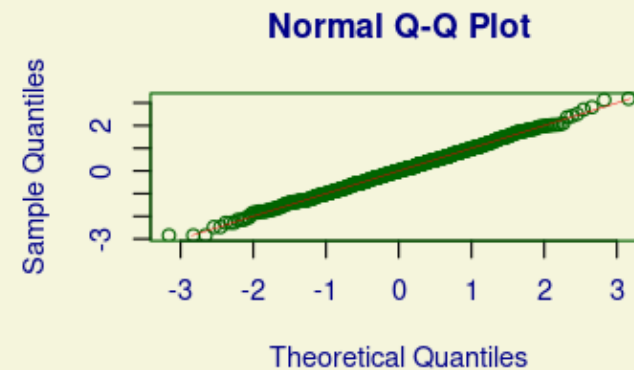
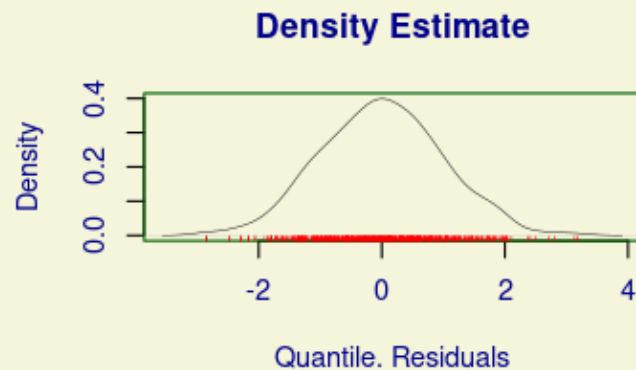
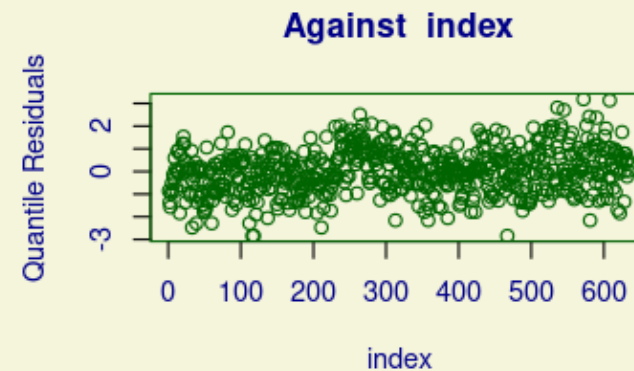
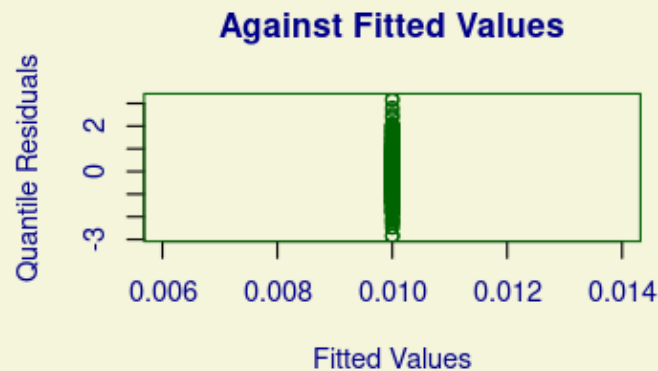
Korelacja

	Straty (AMD vs NVIDIA)	Ceny akcji (AMD vs NVIDIA)
Współczynnik Pearsona	0.169	0.783
Współczynnik Spearmana	0.149	0.908
Współczynnik Kendalla	0.101	0.726

Dopasowany rozkład SEP4 dla NVIDIA



Dopasowany rozkład SEP3 dla AMD

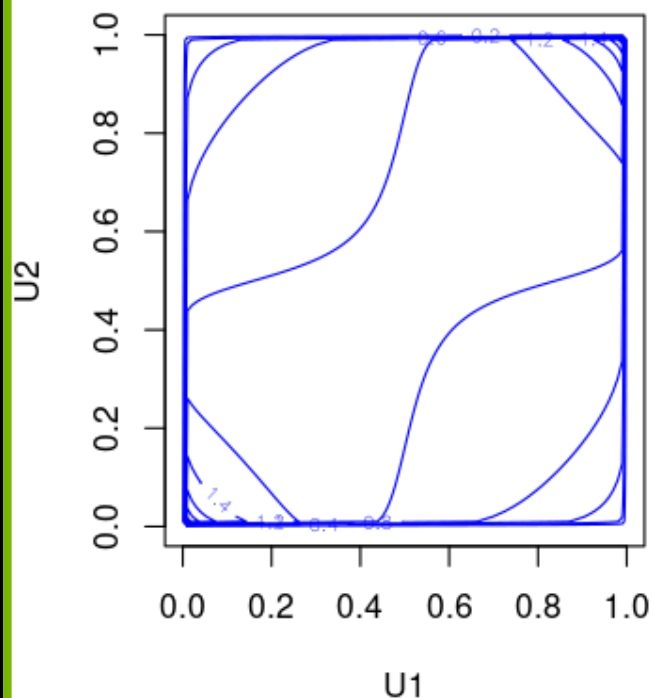


Wyznaczenie najlepiej dopasowanej kopuły

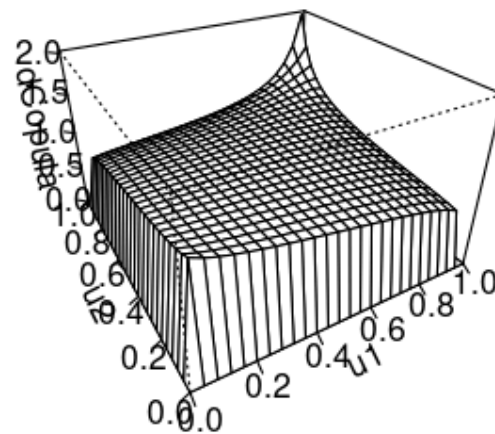
Kryterium/ Kopuła	Gumbela	Franka	Claytona	Normalna	t-Studenta
Loglikelihood	5.165	5.303	1.700	5.823	5.837
AIC	-8.329	-8.605	-1.400	-9.646	-7.674
BIC	-4.189	-4.465	2.740	-5.506	0.606

Wizualizacja najlepiej dopasowanej kopuły

Wykres konturowy kopuły



Dopasowana kopuła Normalna



Test Mardia

Skośność

357.48



rozkład asymetryczny

Kurtoza

12.34



obecność outlierów

Test Andersona-Darlinga

Zmienna	Statystyka	Wartość p	Normalność
NVIDIA.close	20.40	<0.001	NIE
AMD.close	20.90	<0.001	NIE

Value at Risk

N	Alpha	Wartość Beta			
		Dane empiryczne (optimize)	Dane empiryczne (ręcznie)	Dane z kopuły (optimize)	Dane z kopuły (ręcznie)
500	0.95	0.9203	0.9203	0.7594	0.7594
500	0.99	0.7787	0.7787	0.7567	0.7567
1000	0.95	0.8304	0.9039	0.9344	0.9344
1000	0.99	0.7302	0.7302	0.8507	0.8507

Value at Risk

N	Alpha	Wartość VaR			
		Dane empiryczne (optimize)	Dane empiryczne (ręcznie)	Dane z kopuły (optimize)	Dane z kopuły (ręcznie)
500	0.95	3.2365	3.2366	3.4571	3.4572
500	0.99	5.0905	5.0907	5.6780	5.6781
1000	0.95	3.2109	3.2109	3.0863	3.0865
1000	0.99	5.4653	5.4653	5.6477	5.6478

Value at Risk

```
1#Funkcja obliczająca wartość zagrożoną (VaR)
2compute_var <- function(beta, x, alpha = 0.05) {
3  portfolio <- beta * x[,1] + (1 - beta) * x[,2]
4  return(quantile(portfolio, probs = alpha))
5}
6
7#Definiowanie parametrów i inicjalizacja
8values_N <- c(500, 1000)
9values_Alpha <- c(0.95, 0.99)
10betas <- seq(0, 1, by = 0.0001)
11set.seed(5463436)
12
```

Value at Risk

```
13 #Pętla po różnych próbkach (N) i poziomach istotności (Alpha)
14 for (N in values_N) {
15   for (alp in values_Alpha) {
16
17     # Generowanie danych symulowanych
18     simulated_values1 <- do.call(paste0("r", fit1$family[1]), c(list(N), fit1[fit1$parameters]))
19     simulated_values2 <- do.call(paste0("r", fit2$family[1]), c(list(N), fit2[fit2$parameters]))
20     simulated_values_dist <- matrix(c(simulated_values1, simulated_values2), nrow = N, ncol = 2)
21
22     simulated_values_copula <- rCopula(N, get(names(AIC)[which.min(AIC)])@copula)
23
24     # Generowanie danych z kopuły
25     simulated_values1 <- do.call(paste0("q", fit1$family[1]), c(list(simulated_values_copula[, 1]),
26     fit1[fit1$parameters]))
27     simulated_values2 <- do.call(paste0("q", fit2$family[1]), c(list(simulated_values_copula[, 2]),
28     fit2[fit2$parameters]))
29     simulated_values_copula_changed <- data.frame(x1 = simulated_values1, x2 = simulated_values2)
30
31 #Optymalizacja wartości Beta dla danych empirycznych:
32 result_dist <- optimize(compute_var, interval = c(0, 1), x = simulated_values_dist, alpha = alp)
33 cat("Wartości Beta (z funkcji optimize) z danych empirycznych dla N =", N, "i Alpha =", alp,
34     "wynosi", round(result_dist$minimum, 6))
35 print_empty_line()
36
37 #Obliczanie wartości Beta dla danych empirycznych (ręcznie)
38 var_results1 <- data.frame(Beta = numeric(length(betas)), VaR = numeric(length(betas)))
39 for (i in seq_along(betas)) {
40   var_results1[i, ] <- c(betas[i], compute_var(betas[i], x = simulated_values_dist, alp))
41 }
42 sorted_var_results1 <- var_results1[order(var_results1$VaR), ]
43
44 cat("Wartości Beta (liczona ręcznie) z danych empirycznych dla N =", N, "i Alpha =",
45     alp, "wynosi", round(sorted_var_results1$Beta[1], 6), "\n")
46 print_empty_line()
47 }
```

Value at Risk

```
48 #Optymalizacja wartości Beta dla danych z kopuły
49 result_copula <- optimize(compute_var, interval = c(0, 1), x = simulated_values_copula_changed,
50 alpha = alp)
51 cat("Wartości Beta (z funkcji optimize) z danych z kopuły dla N =", N, "i Alpha =", alp,
52     "wynosi", round(result_copula$minimum,6))
53 print_empty_line()
54
55 #Obliczanie wartości Beta dla danych z kopuły (ręcznie)
56 var_results2 <- data.frame(Beta = numeric(length(betas)), VaR = numeric(length(betas)))
57 for (i in seq_along(betas)) {
58   var_results2[i, ] <- c(betas[i], compute_var(betas[i], x = simulated_values_copula_changed, alp))
59 }
60
61 sorted_var_results2 <- var_results2[order(var_results2$VaR), ]
62 cat("Wartości Beta (liczona ręcznie) z danych z kopuły dla N =", N, "i Alpha =", alp, "wynosi",
63     round(sorted_var_results2$Beta[1],6), "\n")
64 print_empty_line()
65 }
66 }
```

Wnioski

AMD:
większa zmienność
większe ryzyko

Współzależność nie pozwala
na pełną dywersyfikację
ryzyka

NVIDIA:
stabilniejsza na rynku
mniejsze ryzyko inwestycyjne

VaR:
kopuły szacują nieco
większe ryzyko