



**Matematikos ir
informatikos
fakultetas**

VILNIAUS UNIVERSITETAS

**MATEMATIKOS IR INFORMATIKOS FAKULTETAS
MAGISTRO STUDIJŲ PROGRAMA
FINANSŲ IR DRAUDIMO MATEMATIKA**

**Kripto valiutų portfelio rizikos vertinimas
naudojant Vine kopulas**

**Cryptocurrency Portfolio Risk Assessment
Using Vine Copulas**

Magistro baigiamasis darbas

Autorius: Lukas Česnulevičius

VU el. pašto adresas: lukas.cesnulevicius@mif.stud.vu.lt

Vadovas: Doc. Martynas Manstavičius

Vilnius
2025

Turinys

1	Įvadas	3
2	Teorinė dalis	4
2.1	Kripto valiutų rinka	4
2.2	Kopulos	4
2.3	Dvimačių kopulų šeimos	6
2.3.1	Gauso kopula	6
2.3.2	t-Kopula	6
2.3.3	Gumbel kopula	6
2.3.4	Clayton kopula	6
2.3.5	Frank kopula	7
2.3.6	Clayton–Gumbel kopula (BB1)	7
2.3.7	Joe–Clayton kopula (BB7)	7
2.4	Grafų teorija	7
2.5	Reguliaroji <i>vine</i> struktūra	8
2.6	Reguliaroji <i>vine</i> kopula	10
2.6.1	Reguliarosios <i>vine</i> kopulos struktūros parinkimo algoritmas	11
2.7	Dviejų uodegų apibendrintas Pareto pasiskirstymo mišinio modelis	12
2.8	ARMA-GJR-GARCH modelis	12
2.8.1	ARMA modelis	12
2.8.2	GARCH modelis	13
2.8.3	GJR-GARCH modelis	13
2.9	Vertės pokyčio rizika	13
2.9.1	VaR modelių testavimas	13
3	Praktinė dalis	14
3.1	Duomenų analizė	14
3.2	ARMA-GJR-GARCH modelio pritaikymas	15
3.3	Dviejų uodegų apibendrinto Pareto pasiskirstymo mišinio modelio pritaikymas	18
3.4	Reguliarosios <i>vine</i> struktūros parinkimas	20
3.5	Vertės pokyčio rizikos skaičiavimas ir testavimas	22
4	Išvados	24
	Literatūra	25
	Priedai	26

Kriptovaliutų portfelio rizikos vertinimas naudojant Vine kopulas

Santrauka

Šiame magistro darbe vertinama kriptovaliutų portfelio rizika, apjungiant ARMA-GJR-GARCH modelius ir reguliarias *vine* kopulas. Pirmiausia kiekvienai kriptovaliutai įvertinamas ARMA(p, q)-GJR-GARCH(1, 1) modelis ir apskaičiuojamos standartizuotos liekanos. Tuomet liekanų tarpusavio priklausomybės struktūra modeliuojama taikant reguliarias *vine* kopulas. Remiantis įvertintu jungtiniu modeliu atliekamos 10000 portfelio grąžų simuliacijos, pagal kurias apskaičiuojama kriptovaliutų portfelio vertės pokyčio rizika. Galiausiai, modelio prognozių patikimumas įvertinamas taikant grįžtamąjį testavimą.

Raktiniai žodžiai: Kriptovaliutos; portfelio rizikos vertinimas; vertės pokyčio rizika; reguliarioji vine struktūra; reguliariosios vine kopulos; ARMA-GJR-GARCH.

Cryptocurrency Portfolio Risk Assessment Using Vine Copulas

Abstract

In this master's thesis, cryptocurrency portfolio risk is assessed by combining ARMA-GJR-GARCH models and regular vine copulas. First, an ARMA(p, q)-GJR-GARCH(1, 1) model is estimated for each cryptocurrency and standardized residuals are obtained. Next, the dependence structure among these residuals is modeled using regular vine copulas. Based on the fitted joint model, 10000 portfolio return simulations are generated, from which the portfolio Value at Risk is computed. Finally, the reliability of the model's forecasts is evaluated using backtesting.

Keywords: Cryptocurrencies; portfolio risk assessment; Value at Risk; regular vine structure; regular vine copulas; ARMA-GJR-GARCH.

1 Įvadas

Kriptovaliutų rinkai sparčiai augant ir vis labiau įsitvirtinant kaip investicinių aktyvų klasei, tampa ypač svarbu suprasti kriptovaliutų rizikos ypatumus ir tarpusavio sąveiką. Kriptovaliutų kainoms būdingas didelis kintamumas ir sustiprėjusios priklausomybės streso laikotarpiais, todėl tradicinės prielaidos apie normalųjį pasiskirstymą ar linijinę koreliaciją dažnai tampa nepakankamos patikimam rizikos vertinimui [13, 14]. Šiame darbe kriptovaliutų portfelio rizika vertinama taikant reguliariomis *vine* kopulomis grįstą priklausomybių modeliavimą, leidžiantį aprašyti netiesines priklausomybes, o rizikai įvertinti bus naudojama vertės pokyčio rizika (angl. Value at Risk, VaR).

Šiame darbe pristatomas kriptovaliutų portfelio VaR modeliavimo metodas, kuriame jungiamas ARMA-GJR-GARCH modelis ir reguliariosios *vine* kopulos. Pirmiausia kiekvienai kriptovaliutai įvertinamas $\text{ARMA}(p, q)\text{-GJR-GARCH}(1, 1)$ modelis ir gaunamos standartizuotos liekanos, kurių marginaliniai skirstiniai įvertinami taikant dviejų uodegų apibendrinto Pareto pasiskirstymo mišinio modelį. Tuomet šių liekanų tarpusavio priklausomybės struktūra modeliuojama taikant reguliariasias *vine* kopulas. Reguliaroji *vine* struktūra gaunama naudojant [7] algoritmą, kopulų šeimos parenkamos naudojant Akaikės informacinį kriterijų, o parametrų įverčiai gaunami didžiausio tikėtimumo metodu. Remiantis įvertintu modeliu atliekamos 10000 portfelio grąžų simuliacijos ir apskaičiuojamas lygiaverčių svorių kriptovaliutų portfelio VaR, o modelio prognozės įvertinamos taikant VaR grįžtamąjį testavimą (angl. backtesting).

Darbo tikslas. Įvertinti kriptovaliutų portfelio riziką, taikant ARMA-GJR-GARCH ir reguliariųjų *vine* kopulų pagrindu sudarytą VaR modelį bei įvertinti jo prognozių tikslumą.

Darbo struktūra. Baigiamąjį darbą sudaro teorinė ir praktinė dalys. Teorinėje dalyje pristatomos kopulos ir dvimačių kopulų savybės, nagrinėjama grafų teorija bei reguliariosios *vine* struktūros, aprašomas reguliariosios *vine* kopulos struktūros parinkimo algoritmas. Taip pat aptariami ARMA-GJR-GARCH modeliai ir dviejų uodegų apibendrintas Pareto pasiskirstymo mišinio modelis. Praktinėje dalyje taikant pasirinktą metodiką įvertinama kriptovaliutų portfelio VaR ir atliekamas grįžtamasis testavimas.

2 Teorinė dalis

2.1 Kriptovaliutų rinka

Kriptovaliutos pastarąjį dešimtmetį tapo reikšminga ir sparčiai besivystančia finansinių rinkų dalimi. Pirmoji kriptovaliuta - "Bitcoin", buvo sukurta kaip decentralizuota skaitmeninė atsiskaitymo priemonė, veikianti nepriklausomai nuo centrinių bankų ar kitų finansinių tarpininkų (tinklas pradėjo veikti 2009 m.) [12]. Didėjant kriptovaliutų rinkos mastui ir investuotojų susidomėjimui, jos vis dažniau traktuojamos kaip investuojamas turtas ir atskira investicinių aktyvų klasė [13]. Empiriniai tyrimai rodo, kad kriptovaliutų (ir jų portfelių) gražos pasižymi dideliu kintamumu ir sunkias uodegas (angl. heavy tails) turinčiais pasiskirstymais, o gražos tarpusavyje yra reikšmingai koreliuotos, ypač ekstremalių rinkos pokyčių laikotarpiais [13, 14]. Dėl spekuliacinio kriptovaliutų pobūdžio investuotojai gali patirti tiek reikšmingą gražą, tiek didelius nuostolius. Todėl sudaryto kriptovaliutų portfelio riziką svarbu vertinti ir modeliuoti taikant tinkamus kiekybinius metodus. Vienas plačiausiai taikomų rizikos matų yra vertės pokyčio rizika (angl. Value at Risk, VaR). Šiame darbe VaR vertinamas lygiaverčiui kriptovaliutų portfeliui, o tarpusavio priklausomybių struktūrai aprašyti naudojamos reguliariosios *vine* kopulos.

2.2 Kopulos

Šiame skyriuje trumpai pristatomi pagrindiniai kopulų teorijos apibrėžimai ir teoremos, kurie bus reikalingi tolimesnioje darbo eigoje. Tegu $\mathbf{X} = (X_1, \dots, X_n)$ yra n -matis atsitiktinių dydžių vektorius su bendra tankio funkcija $f(x_1, \dots, x_n)$ ir bendra pasiskirstymo funkcija $F(x_1, \dots, x_n)$. Tarkime, kad $F_1(x_1), \dots, F_n(x_n)$ yra atitinkamai atsitiktinių dydžių $x_1, \dots, x_n$ marginalai. Kopula, kuri pati yra daugiamatė pasiskirstymo funkcija, "apjungia" marginalus, kad jie sudarytų daugiamatę pasiskirstymo funkciją.

Apibrėžimas 2.1 n -matė kopula C yra daugiamatė pasiskirstymo funkcija

$$C : [0, 1]^n \rightarrow [0, 1],$$

su tolygaus intervale $[0, 1]$ skirstinio marginalais. Tarkime, kad egzistuoja n -osios dalinės C išvestinės. Tuomet kopulos tankis c kiekvienam $x \in [0, 1]^n$ yra lygus

$$c(x_1, \dots, x_n) = \frac{\partial^n}{\partial x_1 \dots \partial x_n} C(x_1, \dots, x_n). \quad (1)$$

Sklar'o teorema (žr. [2]) teigia, kad egzistuoja kopula C , tokia, kad:

$$F(x_1, \dots, x_n) = C(F_1(x_1), \dots, F_n(x_n)). \quad (2)$$

Be to, jei marginalinės pasiskirstymo funkcijos F_i yra tolydžios, tuomet ši kopula yra unikalė. Tarkime c yra kopulos C tankis, tuomet skirstinio funkcijos F tankis f yra lygus

$$f(x_1, \dots, x_n) = c(F_1(x_1), \dots, F_n(x_n))f_1(x_1) \dots f_n(x_n). \quad (3)$$

Toliau nagrinėsime atvejį $n = 2$ ir išvesime sąlyginio tankio ir pasiskirstymo funkcijos išraiškas, kurios bus naudojamos vėliau. Tarkime, kad atsitiktiniai dydžiai (X_1, X_2) yra tolydūs su jungtiniu tankiu f_{12} ir marginaliniais tankiais f_1 ir f_2 , o kopula C_{12} yra tolydi su tankiu c_{12} .

Pagal sąlyginio tankio apibrėžimą

$$f_{1|2}(x_1|x_2) = \frac{f_{12}(x_1, x_2)}{f_2(x_2)}.$$

Istatę (3) (kai $n = 2$) gauname

$$f_{1|2}(x_1|x_2) = \frac{c_{12}(F_1(x_1), F_2(x_2))f_1(x_1)f_2(x_2)}{f_2(x_2)} = c_{12}(F_1(x_1), F_2(x_2))f_1(x_1). \quad (4)$$

Sąlyginė pasiskirstymo funkcija, jei egzistuoja dalinė išvestinė pagal x_2 , gali būti išreikiama

$$F_{1|2}(x_1 | x_2) = \mathbb{P}(X_1 \leq x_1 | X_2 = x_2) = \frac{1}{f_2(x_2)} \frac{\partial}{\partial x_2} F(x_1, x_2). \quad (5)$$

Lygybei (2) pritaikę išvestinę pagal x_2 gauname

$$\frac{\partial}{\partial x_2} F_{12}(x_1, x_2) = \frac{\partial C_{12}(u, v)}{\partial v} \Big|_{u=F_1(x_1), v=F_2(x_2)} \cdot f_2(x_2). \quad (6)$$

Dabar (6) įstatome į (5) ir lieka

$$F_{1|2}(x_1 | x_2) = \frac{\partial}{\partial v} C_{12}(F_1(x_1), v) \Big|_{v=F_2(x_2)}. \quad (7)$$

Kadangi *Vine* kopulos sudaromos tik iš dvimačių kopulų, toliau nagrinėsime tik dvimačių kopulų savybes. Tolimesnei darbo eigai svarbu aprašyti kopulų priklausomybės matavimo būdus.

Apibrėžimas 2.2 (*Kendall's Tau*) Tegų (X_1, Y_1) ir (X_2, Y_2) yra nepriklausomi vieno-
dai pasiskirstę tolydžių atsitiktinių dydžių vektoriai su jungtine pasiskirstymo funkcija F ir
marginalais F_X ir F_Y ir kopula C . Tuomet Kendall's τ yra lygus

$$\tau(X, Y) = \mathbb{P}[(X_1 - X_2)(Y_1 - Y_2) > 0] - \mathbb{P}[(X_1 - X_2)(Y_1 - Y_2) < 0]. \quad (8)$$

Naudojant kopulą C , Kendall's tau išraišką galima užrašyti taip

$$\tau(X, Y) = 4 \int_0^1 \int_0^1 C(u_1, u_2) dC(u_1, u_2) - 1. \quad (9)$$

Apibrėžimas 2.3 (*Spearman's rho*) Tegų (X_1, Y_1) , (X_2, Y_2) ir (X_3, Y_3) yra nepriklausomi
vienodai pasiskirstę tolydžių atsitiktinių dydžių vektoriai su jungtine pasiskirstymo funk-
cija F , marginalais F_X ir F_Y ir kopula C . Tuomet Spearman's ρ yra lygus

$$\rho(X, Y) = \mathbb{P}[(X_1 - X_2)(Y_1 - Y_3) > 0] - \mathbb{P}[(X_1 - X_2)(Y_1 - Y_3) < 0]. \quad (10)$$

Naudojant kopulą C Spearman's ρ galima užrašyti taip

$$\rho(X, Y) = 12 \int_0^1 \int_0^1 C(u, v) dudv - 3. \quad (11)$$

Pilnus (9) ir (11) išraiškų įrodymus galima rasti [1].

2.3 Dvimačių kopulų šeimos

Šiame skyrelyje bus trumpai aprašytos dvimatės kopulos, kurios bus naudojamos daugiamatėms kopuloms kurti naudojant *Vine* struktūras. Šis skyrelis bus parengtas remiantis [3] ir [4].

2.3.1 Gauso kopula

Dvimatė Gauso kopula gali būti užrašyta taip

$$C_R^{Ga}(u, v) = \int_{-\infty}^{\Phi^{-1}(u)} \int_{-\infty}^{\Phi^{-1}(v)} \frac{1}{2\pi(1 - R_{12}^2)^{1/2}} \exp \left\{ -\frac{s^2 - 2R_{12}st + t^2}{2(1 - R_{12}^2)} \right\} ds dt, \quad (12)$$

čia R_{12} žymi atitinkamo dvimačio normaliojo skirstinio tiesinį koreliacijos koeficientą, Φ^{-1} - atvirkštinę vienmačio standartinio normaliojo skirstinio pasiskirstymo funkciją, u ir v yra argumentai iš intervalo $[0, 1]$.

2.3.2 t-Kopula

Dvimatė t-kopula gali būti užrašyta taip

$$C_{\nu, R}^t(u, v) = \int_{-\infty}^{t_\nu^{-1}(u)} \int_{-\infty}^{t_\nu^{-1}(v)} \frac{1}{2\pi(1 - R_{12}^2)^{1/2}} \left(1 + \frac{s^2 - 2R_{12}st + t^2}{2(1 - R_{12}^2)} \right)^{-\frac{(\nu+2)}{2}} ds dt, \quad (13)$$

čia R_{12} žymi tiesinės koreliacijos koeficientą atitinkamam dvimačiam t_ν pasiskirstymui, t_ν^{-1} - atvirkštinę Stjudento t-pasiskirstymo funkciją su laisvės laipsniu ν , o u ir v yra argumentai iš intervalo $[0, 1]$.

Toliau aprašysime Archimedo kopulas. Norint jas apibrėžti, reikia įvesti vadinamąją Archimedo kopulų generatoriaus funkciją.

Apibrėžimas 2.4 Tarkime, kad φ yra tolydi, griežtai mažėjanti funkcija: $[0, 1] \rightarrow [0, \infty]$ tokia, kad $\varphi(1) = 0$, o $\varphi^{[-1]}$ yra jos atvirkštinė. Tegu C yra funkcija: $[0, 1]^2 \rightarrow [0, 1]$ nusakoma formule

$$C(u, v) = \varphi^{[-1]}(\varphi(u) + \varphi(v)). \quad (14)$$

Tuomet funkcija C yra kopula tada ir tik tada kai φ yra iškila į viršų.

Kopulos, turinčios formą (14), vadinamos Archimedo kopulomis, o funkcijos φ - kopulų generatoriais.

2.3.3 Gumbel kopula

Tarkime, $\varphi(t) = (-\ln t)^\theta$, kur $\theta \geq 1$. Tuomet dvimatė Gumbel kopula gali būti užrašyta taip

$$C_\theta^{Gu}(u, v) = \exp \left(- \left[(-\ln u)^\theta + (-\ln v)^\theta \right]^{1/\theta} \right). \quad (15)$$

2.3.4 Clayton kopula

Tarkime, $\varphi(t) = \frac{(t^{-\theta}-1)}{\theta}$, kur $\theta \in [-1, \infty) \setminus \{0\}$. Tuomet dvimatė Clayton kopula gali būti užrašyta taip

$$C_\theta(u, v) = \left[\max \left(u^{-\theta} + v^{-\theta} - 1, 0 \right) \right]^{-1/\theta}. \quad (16)$$

2.3.5 Frank kopula

Tarkime, $\varphi(t) = -\ln\left(\frac{e^{-\theta t}-1}{e^{-\theta}-1}\right)$, kur $\theta \in \mathbb{R} \setminus \{0\}$. Tuomet dvimatė Frank kopula gali būti užrašyta taip

$$C_{\theta}^{Fr}(u, v) = -\frac{1}{\theta} \ln \left(1 + \frac{(e^{-\theta u} - 1)(e^{-\theta v} - 1)}{e^{-\theta} - 1} \right). \quad (17)$$

2.3.6 Clayton–Gumbel kopula (BB1)

Clayton–Gumbel kopula yra dviejų parametų Archimedo kopula, kurią galima traktuoti kaip vieno parametro Clayton ir Gumbel šeimų apibendrinimu. Ši kopula dažniausiai vadinama BB1 kopula.

Tarkime, $\varphi(t) = (t^{-\theta} - 1)^{\delta}$, kur $\theta > 0$ ir $\delta \geq 1$. Tuomet dvimatė BB1 kopula gali būti užrašyta taip

$$C_{\theta, \delta}^{BB1}(u, v) = [1 + [(u^{-\theta} - 1)^{\delta} + (v^{-\theta} - 1)^{\delta}]^{\frac{1}{\delta}}]^{-\frac{1}{\theta}}. \quad (18)$$

2.3.7 Joe–Clayton kopula (BB7)

Joe–Clayton kopula, panašiai kaip BB1 kopula, yra dviejų parametų apibendrinimas atitinkamoms vieno parametro kopulų šeimoms. Ši kopula dažniausiai vadinama BB7 kopula.

Tarkime, $\varphi(t) = [1 - (1 - t)^{\theta}]^{\delta} - 1$, kur $\theta \geq 1$ ir $\delta > 0$. Tuomet dvimatė BB7 kopula gali būti užrašyta taip

$$C_{\theta, \delta}^{BB7}(u, v) = 1 - [1 - [(1 - (1 - u)^{\theta})^{-\delta} + (1 - (1 - v)^{\theta})^{-\delta}]^{-\frac{1}{\delta}}]^{\frac{1}{\theta}}. \quad (19)$$

2.4 Grafų teorija

Vine kopulų konstrukcijos remiasi medžių sekomis (angl. tree sequence), kurios aprašo kintamųjų tarpusavio ryšius ir sąlygines priklausomybes. Kad šią struktūrą formaliai apibrėžtume, reikia kelių pagrindinių grafų teorijos apibrėžimų. Todėl prieš pristatant reguliariusius *vine* (angl. regular vine), trumpai aprašoma grafų teorija remiantis [5].

Apibrėžimas 2.5 *Neorientuotu grafu G vadiname porą (N, E) , kur N yra baigtinė viršūnių aibė $N = \{v_1, v_2, \dots, v_n\}$, o $E \subset \{\{u, v\} : u, v \in N, u \neq v\}$ yra aibė sudaryta iš N elementų porų ir vadinama briaunų aibe. Iš viršūnės $v \in N$ išeinančių briaunų skaičių vadiname laipsniu ir žymime $\deg(v)$.*

Šiame darbe bus nagrinėjami tik neorientuoti grafai.

Apibrėžimas 2.6 *Seka $v_{j_1}, v_{j_2}, \dots, v_{j_K}$, sudaryta iš grafo $G = (N, E)$ viršūnių yra vadinama keliu, jeigu bet kuriuos du gretimus šios sekos elementus jungia briauna, t.y. $\{v_{j_k}, v_{j_{k+1}}\} \in E$ su visais $1 \leq k \leq K - 1$. Kelias, kurio visos viršūnės yra skirtingos, vadinamas taku, o uždaras takas - ciklu.*

Apibrėžimas 2.7 *Grafas $G = (V, E)$ yra vadinamas jungiu, jeigu bet kuriai viršūnių porai $u, v \in N, u \neq v$ egzistuoja jas jungiantis kelias, t.y. toks kelias kurio pradinė viršūnė yra u , o galinė - v .*

Apibrėžimas 2.8 *Medžiu vadinamas jungus grafas neturintis ciklų.*

2.5 Reguliaroji vine struktūra

Šiame skyrelyje bus aprašytos reguliariosios *vine* struktūros (angl. regular vine, R-vine), kurias pristatė [6] straipsnio autoriai. *Vine* struktūrą žymėsime $\mathcal{V}$ ir vadinsime jungių medžių seką $\mathcal{V} = T_1, \dots, T_{n-1}$, kur medžio T_i viršūnes sudaro medžio T_{i-1} briaunos $i = 2, \dots, n-1$. Reguliaroji *vine* yra toks *vine* struktūros atvejis, kai dvi medžio T_{i-1} briaunos gali būti sujungtos briauna medyje T_i tik tuomet, jei šios briaunos turi bendrą viršūnę (vadinamoji artumo sąlyga, angl. proximity condition). Formalų apibrėžimą pateikiame žemiau.

Apibrėžimas 2.9 Tegu $\mathcal{V} = (T_1, \dots, T_n)$ yra medžių seka. Sakysime, kad $\mathcal{V}$ yra reguliaroji vine iš n elementų, jei tenkinamos šios sąlygos:

1. T_1 yra medis, kurio viršūnių aibė yra $N_1 = \{1, \dots, n\}$, o briaunų aibė - E_1 ;
2. Visiems $i = 2, \dots, n-1$, T_i yra medis, kurio viršūnių aibė yra $N_i = E_{i-1}$, o briaunų aibė - E_i ;
3. Visiems $i = 2, \dots, n-1$ ir $\{a, b\} \in E_i$, kur $a = \{a_1, a_2\}$ ir $b = \{b_1, b_2\}$, turi galioti artumo sąlyga $\#(a \cap b) = 1$, kur $\#$ žymi aibės kardinalumą.

Norėdami toliau nagrinėti reguliariųjų *vine* struktūrų savybes, kiekvienai briaunai apibrėžiame keturias aibes. Briaunos pilnos sąjungos aibė (angl. complete union set) vadinsime visų su šia briauna susietų indeksų aibę. Jei viršūnės a ir b jungia briauna, tuomet jos sąlygojama aibė (angl. conditioned set) ir sąlygojanti aibė (angl. conditioning set) yra atitinkamai apibrėžiamos kaip viršūnių a ir b pilnos sąjungos aibių simetrinis skirtumas ir sankirta. Visos $\mathcal{V}$ briaunų sąlygojamos ir sąlygojančios aibės sudaro aibę, kuri yra vadinama apribojimų aibe (angl. constraint set). Formalūs šių aibių apibrėžimai pateikiami žemiau remiantis [6] ir [7]:

Apibrėžimas 2.10 (Pilnos sąjungos aibė) Briaunos $e_i \in E_i$ pilnos sąjungos aibė yra aibė

$$U_{e_i} = \{n_1 \in N_1 \mid \exists e_j \in E_j, j = 1, \dots, i-1, \text{ kur } n_1 \in e_j, \forall j = 1, \dots, i\} \subset N_1. \quad (20)$$

Apibrėžimas 2.11 (Apribojimų aibė) Tarkime, kad $e_i = \{a, b\} \in E_i$, $a, b \in N_i$, $i = 1, \dots, n-1$. Tuomet briaunos e_i sąlygojanti aibė yra

$$D_{e_i} = U_a \cap U_b, \quad (21)$$

o briaunos sąlygojamos aibės yra

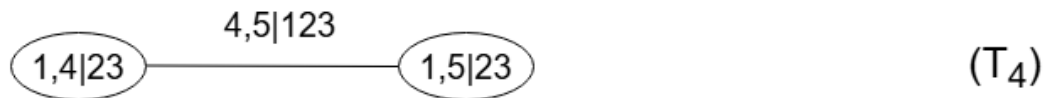
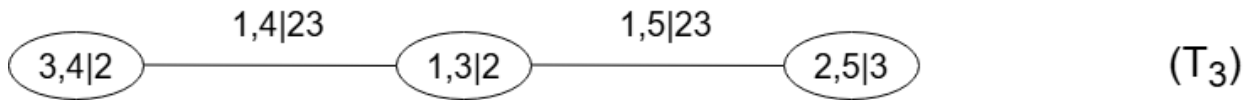
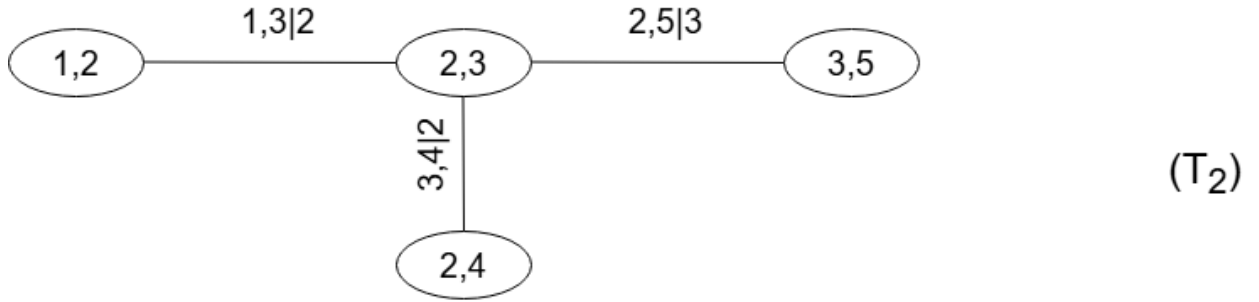
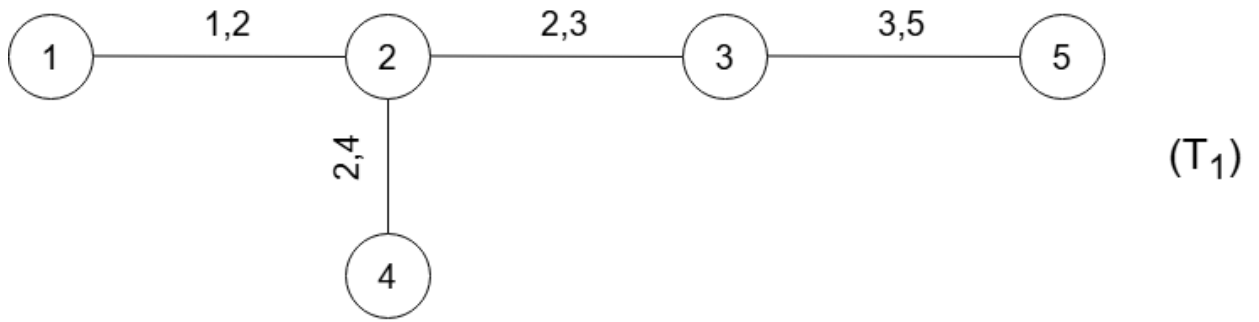
$$C_{e_i,a} = U_a \setminus D_{e_i}, \quad C_{e_i,b} = U_b \setminus D_{e_i}, \quad (22)$$

ir

$$C_{e_i} = C_{e_i,a} \cup C_{e_i,b} = U_a \Delta U_b, \quad (23)$$

kur $A \Delta B := (A \setminus B) \cup (B \setminus A)$ žymi dviejų aibių simetrinį skirtumą. Tuomet $\mathcal{V}$ apribojimų aibė yra

$$C\mathcal{V} = \{(\{C_{e,a}, C_{e,b}\}, D_e \mid e \in E_i, e = \{a, b\}, i = 1, \dots, n-1)\}. \quad (24)$$



1 pav.: R-vine struktūros pavyzdys su 5 kintamaisiais. Kiekvienoje briaunoje $e = \{a, b\} \in E_i$, sąlygojamos aibės $C_{e,a}$ ir $C_{e,b}$ atskiriamos kableliu ir rašomos prieš ženklą "|", o sąlygojanti aibė D_e po "|".

Medyje T_2 (2) nagrinėkime briauną a , jungiančią viršūnes $(1, 2)$ ir $(2, 3)$. Pagal apibrėžimą, briaunos a pilnos sąjungos aibę sudaro visi su ja susieti indeksai, todėl gauname $\{1, 2\} \cup \{2, 3\} = \{1, 2, 3\}$. Analogiškai, briaunos b tarp viršūnių $(2, 3)$ ir $(3, 5)$ pilnos sąjungos aibė yra $\{2, 3, 5\}$, nes $\{2, 3\} \cup \{3, 5\} = \{2, 3, 5\}$. Tuomet, remiantis aukščiau pateiktu apibrėžimu, briaunos, jungiančios a ir b medyje T_3 , sąlygojanti aibė lygi šių pilnų sąjungų sankirtai $\{1, 2, 3\} \cap \{2, 3, 5\} = \{2, 3\}$, o sąlygojama aibė - jų simetriniam skirtumui $\{1, 2, 3\} \Delta \{2, 3, 5\} = \{1, 5\}$.

2.6 Reguliaroji *vine* kopula

Šiame skyrelyje bus aprašytos reguliariosios *vine* kopulos (angl. regular vine copula), kurias pristatė [6] straipsnio autoriai. *Vine* kopulų modelių tikslas - rasti būdą daugiamatės kopulas išreikšti tik dvimačių kopulų deriniais. Daugiamatinių kopulų suskaidymas į dvi matės kopulas yra vadinamas porinių kopulų konstrukcija (angl. pair copula construction). Toliau bus iliustruota ši konstrukcija, kai $d = 3$:

Tarkime, kad turime trimatį atsitiktinių dydžių vektorių $\mathbf{X} = (X_1, X_2, X_3)$ su bendra tankio funkcija $f(x_1, x_2, x_3)$ ir marginalinėmis tankio funkcijomis f_1, f_2, f_3 . Iš sąlyginio tankio apibrėžimo trimatę tankio funkciją galima užrašyti taip:

$$f(x_1, x_2, x_3) = f_{2|13}(x_2|x_1, x_3)f_{1|3}(x_1|x_3)f_3(x_3). \quad (25)$$

Taikydami Sklar'o teoremą (4) sąlyginiam tankiui $f_{12|3}(x_1, x_2|x_3)$, gauname:

$$f_{12|3}(x_1, x_2|x_3) = c_{12|3}(F_{1|3}(x_1|x_3), F_{2|3}(x_2|x_3); x_3)f_{1|3}(x_1|x_3)f_{2|3}(x_2|x_3) \quad (26)$$

Pagal sąlyginio tankio apibrėžimą, $f_{2|13}(x_2|x_1, x_3) = \frac{f_{12|3}(x_1, x_2|x_3)}{f_{1|3}(x_1|x_3)}$, įstatę išraišką (27) gauname:

$$f_{2|13}(x_2|x_1, x_3) = c_{12|3}(F_{1|3}(x_1|x_3), F_{2|3}(x_2|x_3); x_3)f_{2|3}(x_2|x_3) \quad (27)$$

Toliau tiesiogiai pritaikome (4) lygtį ir gauname sąlyginius tankius $f_{1|3}(x_1|x_3), f_{2|3}(x_2|x_3)$:

$$f_{1|3}(x_1|x_3) = c_{13}(F_1(x_1), F_3(x_3))f_1(x_1), \quad (28)$$

$$f_{2|3}(x_2|x_3) = c_{23}(F_2(x_2), F_3(x_3))f_2(x_2). \quad (29)$$

Galiausiai, įstačius (27), (28) ir (29) lygtis į (25) lygtį, gauname trimačio tankio $f(x_1, x_2, x_3)$ porinių kopulų konstrukciją, išreikštą per sąlyginę kopulą $c_{12|3}$:

$$\begin{aligned} f(x_1, x_2, x_3) &= c_{12|3}(F_{1|3}(x_1|x_3), F_{2|3}(x_2|x_3); x_3) c_{13}(F_1(x_1), F_3(x_3)) c_{23}(F_2(x_2), F_3(x_3)) \\ &\quad \times f_1(x_1)f_2(x_2)f_3(x_3). \end{aligned} \quad (30)$$

Verta pabrėžti, kad ši konstrukcija nėra unikali. Lygtyje (25) kintamuosius galima pertvarkyti $3! = 6$ būdais.

Vėliau matysime, kad n -mačiam vektoriui galima taikyti porinių kopulų konstrukciją, o reguliaroji *vine* struktūra nusako, kurios kintamųjų poros sujungiamos kokiomis kopulomis ir nuo kurių kintamųjų jos priklauso. Tokį modelį vadinsime reguliariąja *vine* kopula (angl. regular vine copula). Formalų jos apibrėžimą pateikiame žemiau remiantis [6] ir [7]:

Apibrėžimas 2.12 $(F, \mathcal{V}, B)$ yra reguliaroji *vine* kopula, jei $F = (F_1, \dots, F_n)$ yra tolydžių, griežtai didėjančių pasiskirstymo funkcijų vektorius, $\mathcal{V}$ yra n -matė reguliaroji *vine* struktūra, o $B = \{B_e | i = 1, \dots, n-1, e \in E_i\}$ yra dvimačių kopulų B_e aibė.

Sakome, kad atsitiktinio vektoriaus $(X_1, \dots, X_n)$ jungtinė pasiskirstymo funkcija F yra reguliariosios *vine* kopulos realizacija $(F, \mathcal{V}, B)$, jei kiekvienai briaunai $e \in E_i, i = 1, \dots, n-1, e = \{a, b\}$, kopula B_e yra atsitiktinių dydžių $X_{C_{e,a}}$ ir $X_{C_{e,b}}$ dvi matės kopula, ir $X_{D_e} = \{X_i | i \in D_e\}$. Taip pat, visiems $j = 1, \dots, n$ funkcija F_j yra atsitiktinio dydžio X_j marginalinis pasiskirstymas. Kopulos B_e , atitinkančios briauną $e = \{a, b\}$, tankį žymėsime $e_{C_{e,a}, C_{e,b} | D_e}$.

Toliau pateikiama n -matės reguliariosios *vine* kopulos tankio funkcija, remiantis [6] ir [7].

Apibrėžimas 2.13 Tegul $(F, \mathcal{V}, B)$ yra n elementų reguliariosios vine kopulos specifikacija. Tuomet egzistuoja unikali jungtinė pasiskirstymo funkcija F , kurios tankio funkcija yra

$$f_{1\dots n}(x) = \prod_{k=1}^n f_k(x_k) \prod_{i=1}^{n-1} \prod_{e \in E_1} c_{C_{e,a}, C_{e,b}|D_e}(F_{C_{e,a}|D_e}(x_{C_{e,a}}|\mathbf{x}_{D_e}) F_{C_{e,b}|D_e}(x_{C_{e,b}}|\mathbf{x}_{D_e})), \quad (31)$$

kur $\mathbf{x} = (x_1, \dots, x_n)$, $e = \{a, b\}$ ir $\mathbf{x}_{D_e}$ žymi kintamuosius esančius aibėje D_e ($\mathbf{x}_{D_e} = \{x_i | i \in D_e\}$), o f_i žymi F_i tankį visiems $i = 1, \dots, n$.

Pilną (31) išraiškos įrodymą galima rasti [6].

Norėdami gauti sąlygines skirstinio funkcijas $F_{C_{e,a}|D_e}(x_{C_{e,a}} | x_{D_e})$ ir $F_{C_{e,b}|D_e}(x_{C_{e,b}} | x_{D_e})$, tarkime, kad $e = \{a, b\} \in E_i$, $a = \{a_1, a_2\}$, $b = \{b_1, b_2\}$ yra briauna, jungianti $C_{e,a}$ su $C_{e,b}$, esant sąlygai D_e . Tuomet, turime, kad

$$\begin{aligned} F_{C_{e,a}|D_e}(x_{C_{e,a}}|\mathbf{x}_{D_e}) &= \frac{\partial C_{C_a|D_a}(F_{C_{a,a_1}|D_a}(x_{C_{a,a_1}}|\mathbf{x}_{D_a}), F_{C_{a,a_2}|D_a}(x_{C_{a,a_2}}|\mathbf{x}_{D_a}))}{\partial F_{C_{a,a_2}|D_a}(x_{C_{a,a_2}}|\mathbf{x}_{D_a})} \\ &=: h(F_{C_{a,a_1}|D_a}(x_{C_{a,a_1}} | x_{D_a}), F_{C_{a,a_2}|D_a}(x_{C_{a,a_2}} | x_{D_a})), \end{aligned} \quad (32)$$

čia funkcija h įvesta dėl patogumo [7].

2.6.1 Reguliariosios vine kopulos struktūros parinkimo algoritmas

Kadangi yra $\frac{n}{2} \times (n-2)! \times 2^{\binom{n-2}{2}}$ galimų R –vine struktūrų [8], jų išrinkimas iš visos erdvės yra nepraktiškas. Todėl praktinėje dalyje vine kopulos struktūrai nustatyti bus naudojamas toliau pateiktas algoritmas kurį pristatė straipsnio [7] autoriai.

Algoritmas 1: R-vine kopulos modelio parinkimo metodas, paremtas Kendall's τ

Įvestis : Duomenys $x_{\ell_1}, \dots, x_{\ell_n}$, $\ell = 1, \dots, N$ (nepriklausomi tolygiai pasiskirstę atsitiktiniai dydžiai)

Išvestis: R-vine kopulos specifikacija, $(\mathcal{V}, B)$

- 1 Apskaičiuoti empirinius Kendall's τ $\hat{\tau}_{j,k}$ visoms poroms $\{j, k\}$, $1 \leq j < k \leq n$;
- 2 Pasirinkti jungiamąjį medį, kuris maksimizuoja absoliučių empirinių Kendall's τ sumą,

$$\max \sum_{e=\{j,k\} \text{ medyje}} |\hat{\tau}_{j,k}|;$$

- 3 Kiekvienai briaunai $\{j, k\}$ pasirinktame medyje parinkti kopulą ir įvertinti jos parametrus. Tada, naudojant gautą kopulą $\hat{C}_{jk}$, transformuoti $\hat{F}_{j|k}(x_{\ell_j} | x_{\ell_k})$ ir $\hat{F}_{k|j}(x_{\ell_k} | x_{\ell_j})$, visiems $\ell = 1, \dots, N$ (32);

4 **for** $i = 2$ **to** $n - 1$ **do**

- 5 Apskaičiuoti empirinius Kendall's τ $\hat{\tau}_{j,k|D}$ visoms poroms $\{j, k|D\}$, kurios gali sudaryti medį T_i , t.y. visoms briaunoms, tenkinančioms artumo sąlyga (2.9);
- 6 Iš šių briaunų pasirinkti jungiamąjį medį, kuris maksimizuoja sumą

$$\max \sum_{e=\{j,k|D\} \text{ medyje}} |\hat{\tau}_{j,k|D}|$$

- 7 Kiekvienai briaunai $\{j, k|D\}$ gautame jungiamajame medyje parinkti sąlyginę kopulą ir įvertinti jos parametrus. Tada, naudojant gautą kopulą $\hat{C}_{jk|D}$, transformuoti $\hat{F}_{j|k \cup D}(x_{\ell_j} | x_{\ell_k}, \mathbf{x}_{\ell D})$ ir $\hat{F}_{k|j \cup D}(x_{\ell_k} | x_{\ell_j}, \mathbf{x}_{\ell D})$, visiems $\ell = 1, \dots, N$.
-

Irodymas, kad šis algoritmas visada sukonstruoja R-vine kopulos struktūrą, pateiktas [7]. Pateikiname algoritme priklausomybei vertinti naudojamas Kendall's τ , tačiau aprašytas metodas veikia taip pat su kitais koreliacijos koeficientais, įskaitant Spearman's ρ . Praktinėje dalyje trečiame ir septintame algoritmo žingsniuose optimali kopulos šeima parenkama naudojant Akaikės informacinį kriterijų [9] (AIC), o jos parametrai įvertinti didžiausio tikėtinumo metodu (angl. Maximum Likelihood Estimation).

2.7 Dviejų uodegų apibendrintas Pareto pasiskirstymo mišinio modelis

Darbo praktinėje dalyje kopulų marginaliosioms pasiskirstymo funkcijoms modeliuoti naudosis toliau aprašytą dviejų uodegų apibendrinto Pareto pasiskirstymo mišinio modelį (GNG, angl. GDP-Normal-GDP mixture model). Modelis buvo pristatytas straipsnio [10] autorių. Prieš pereidami prie GNG modelio aprašo, pirmiausia formaliai aprašome vienos uodegos apibendrinto Pareto pasiskirstymo (GDP) modelį.

Tarkime, X yra atsitiktinis dydis, kurio sąlyginė pasiskirstymo funkcija, kai $X > u$, aprašoma

$$G(x | \xi, \sigma, u) = \Pr(X < x | X > u) = \begin{cases} 1 - \left[1 + \xi \left(\frac{x - u}{\sigma}\right)\right]_+^{-1/\xi}, & \xi \neq 0 \\ 1 - \exp\left[-\left(\frac{x - u}{\sigma}\right)\right], & \xi = 0, \end{cases} \quad (33)$$

kur $x \geq u$, $\sigma > 0$ ir $y_+ = \max(y, 0)$, čia σ yra skalės parametras, o ξ - formos parametras.

Dviejų uodegų GDP mišinio modelis turi atskiras apibendrinto Pareto pasiskirstymus uodegų apatinei ir viršutinei daliai, o tarp jų esančiai duomenų daliai taikomas normalusis pasiskirstymas.

Tarkime, kad X yra atsitiktinis dydis su GNG skirstiniu. Tuomet pasiskirstymo funkcija $P(X \leq x) = F(x)$ yra

$$\begin{aligned} F(x|\theta) = & \left\{ \Phi(u_l|m, s) \left[1 - G(-x|\xi_l, \sigma_l, -u_l) \right] \right\} I_{(-\infty, u_l)}(x) \\ & + \Phi(x|m, s) I_{(u_l, u_r)}(x) \\ & + \left\{ \Phi(u_r|m, s) + \left[1 - \Phi(u_r|m, s) \right] G(x|\xi_r, \sigma_r, u_r) \right\} I_{[u_r, \infty)}(x), \end{aligned} \quad (34)$$

čia $\Phi(x)$ žymi normaliojo pasiskirstymo funkciją su vidurkiu m ir dispersija s^2 , o $G(x|\xi, \sigma, u)$ yra GDP skirstinio funkcija (33). Indeksas l nusako GDP apatinę (kairiąją) uodegą, o r - viršutinę (dešiniąją) uodegą.

2.8 ARMA-GJR-GARCH modelis

Tam, kad darbo praktinėje dalyje galėtume taikyti kopulas, mums reikės standartizuotų liekanų, kurias gausime pritaikę ARMA-GJR-GARCH modelį. Jo formalus apibrėžimas pateikiamas žemiau, remiantis [11] straipsniu.

2.8.1 ARMA modelis

Autoregresinis slankiojo vidurkio (ARMA, angl. Autoregressive moving average) modelis yra išreikštas tokia lygtimi:

$$r_t = \omega + \sum_{i=1}^p \phi_i r_{t-i} + \sum_{j=1}^q \theta_j a_{t-j} + a_t, \quad (35)$$

kur r_t yra grąža momentu t , a_t - baltojo triukšmo procesas momentu t , o p žymi AR eilę ir q - MA eilę.

2.8.2 GARCH modelis

GARCH (angl. Generalized AutoRegressive Conditional Heteroskedasticity) modelis yra 1986m. Bollerslev'o sukurto ARCH modelio apibendrinimas. Šis modelis yra išreikštas tokia lygtimi:

$$\sigma_t^2 = \omega + \sum_{i=1}^q \alpha_i a_{t-i}^2 + \sum_{j=1}^p \beta_j \sigma_{t-j}^2, \quad (36)$$

kur σ_t^2 - liekanų dispersija momentu t , $\omega > 0$, $\alpha_i \geq 0$, $\beta_j \geq 0$ visiems $i = 1, \dots, q$, $j = 1, \dots, p$, q yra dažnai vadinama ARCH eilės tvarka, o p - GARCH eilės tvarka.

2.8.3 GJR-GARCH modelis

Straipsnio [16] autoriai pristatė asimetrinį GARCH modelį, vadinamą GJR-GARCH. Šis modelis leidžia įvertinti skirtingų neigiamų ir teigiamų naujienų poveikį kintamumui. Formali GJR-GARCH modelio išraiška pateikiama žemiau:

$$\sigma_t^2 = \omega + \sum_{i=1}^q \alpha_i a_{t-i}^2 + \sum_{j=1}^p \beta_j \sigma_{t-j}^2 + \gamma_i I_{t-i} a_{t-i}^2, \quad (37)$$

$$I_{t-i} = \begin{cases} 1, & a_{t-i} < 0, \\ 0, & a_{t-i} \geq 0, \end{cases}$$

čia σ_t^2 - liekanų dispersija momentu t , $\omega > 0$, $\alpha_i \geq 0$, $\beta_j \geq 0$ visiems $i = 1, \dots, q$, $j = 1, \dots, p$, q . Jei $\gamma > 0$, tuomet blogų naujienų poveikis yra didesnis negu gerų naujienų poveikis.

2.9 Vertės pokyčio rizika

Rizikos valdyme viena iš pagrindinių užduočių yra kiekybiškai įvertinti galimus finansinio portfelio nuostolius per tam tikrą laikotarpį. Šiam uždaviniui spręsti dažnai naudojama vertės pokyčio rizika (VaR, angl. Value at Risk).

Tarkime, kad atsitiktinis dydis L žymi portfelio nuostolį, o jo pasiskirstymo funkcija yra $F_L(x) = \mathbb{P}(L \leq x)$. Tuomet vertės pokyčio rizika yra apibrėžiama kaip

$$VaR_\alpha(L) = \inf\{x \in \mathbb{R} : F_L(x) \geq \alpha\}, \quad (38)$$

čia $\alpha \in (0, 1)$ žymi patikimumo lygį.

2.9.1 VaR modelių testavimas

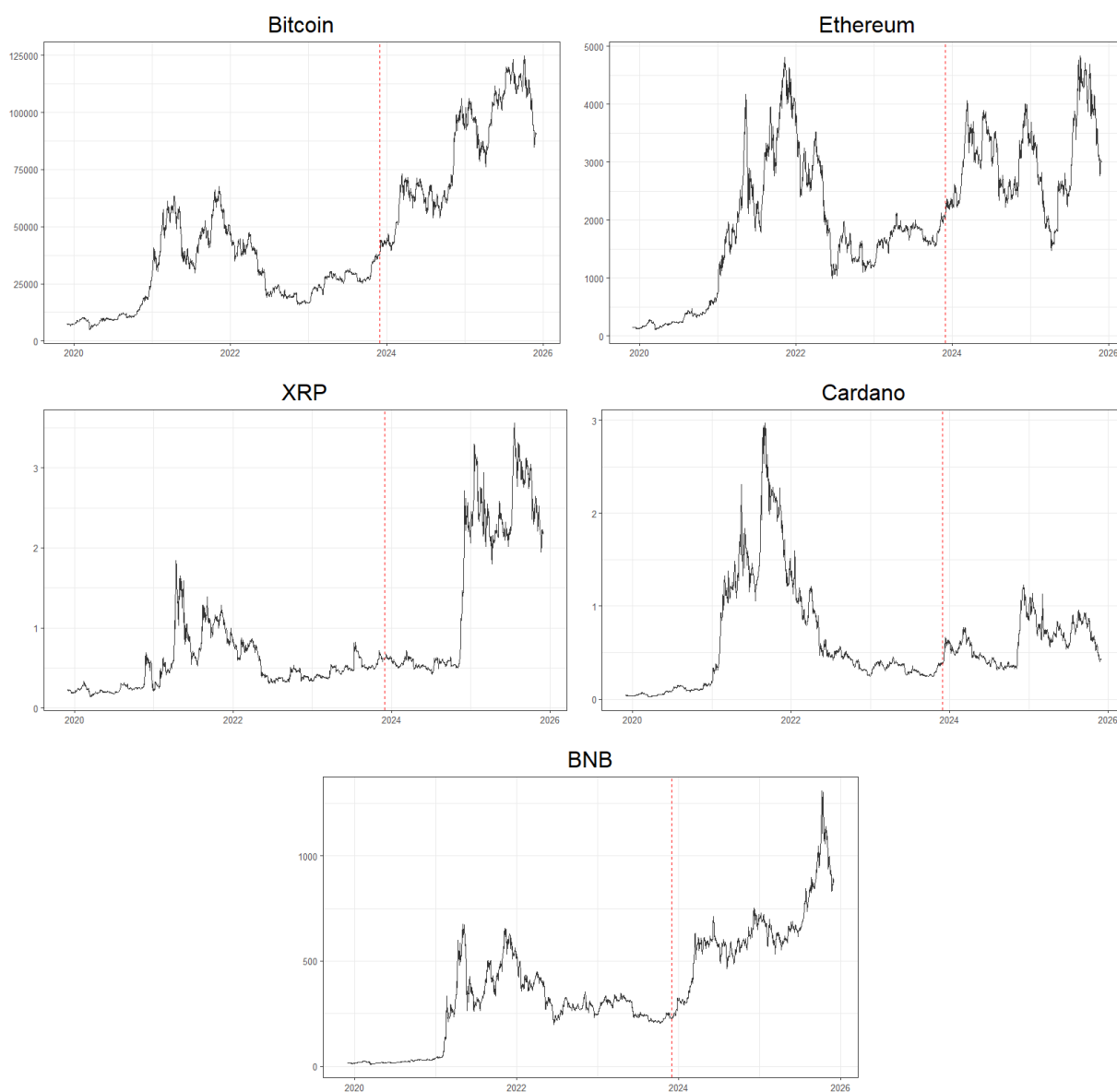
Praktinėje dalyje atliksime VaR modelio grįžtamąjį testavimą (angl. backtesting) lyginant prognozuotas $VaR_{\alpha,t}$ reikšmes su faktiškai stebėtais nuostoliais L_t . Tegu $I = \sum_{t=1}^T \mathbb{1}_{\{L_t > VaR_{\alpha,t}\}}$ žymi bendrą nuokrypių skaičių, tuomet gero modelio santykis I/T turėtų būti artimas $1 - \alpha$. Ši prielaida bus tikrinama naudojant Kupiec testą [15].

3 Praktinė dalis

Šiame skyriuje pateikiami praktiniai skaičiavimai, kuriais įgyvendinama anksčiau aptarta teorija: įvertinama lygiaverčio kriptovaliutų portfelio vertės pokyčio rizika. Priklausomybės tarp aktyvų aprašomos taikant *R-vine* kopulos struktūrą, o visi skaičiavimai atlikti "R" aplinkoje.

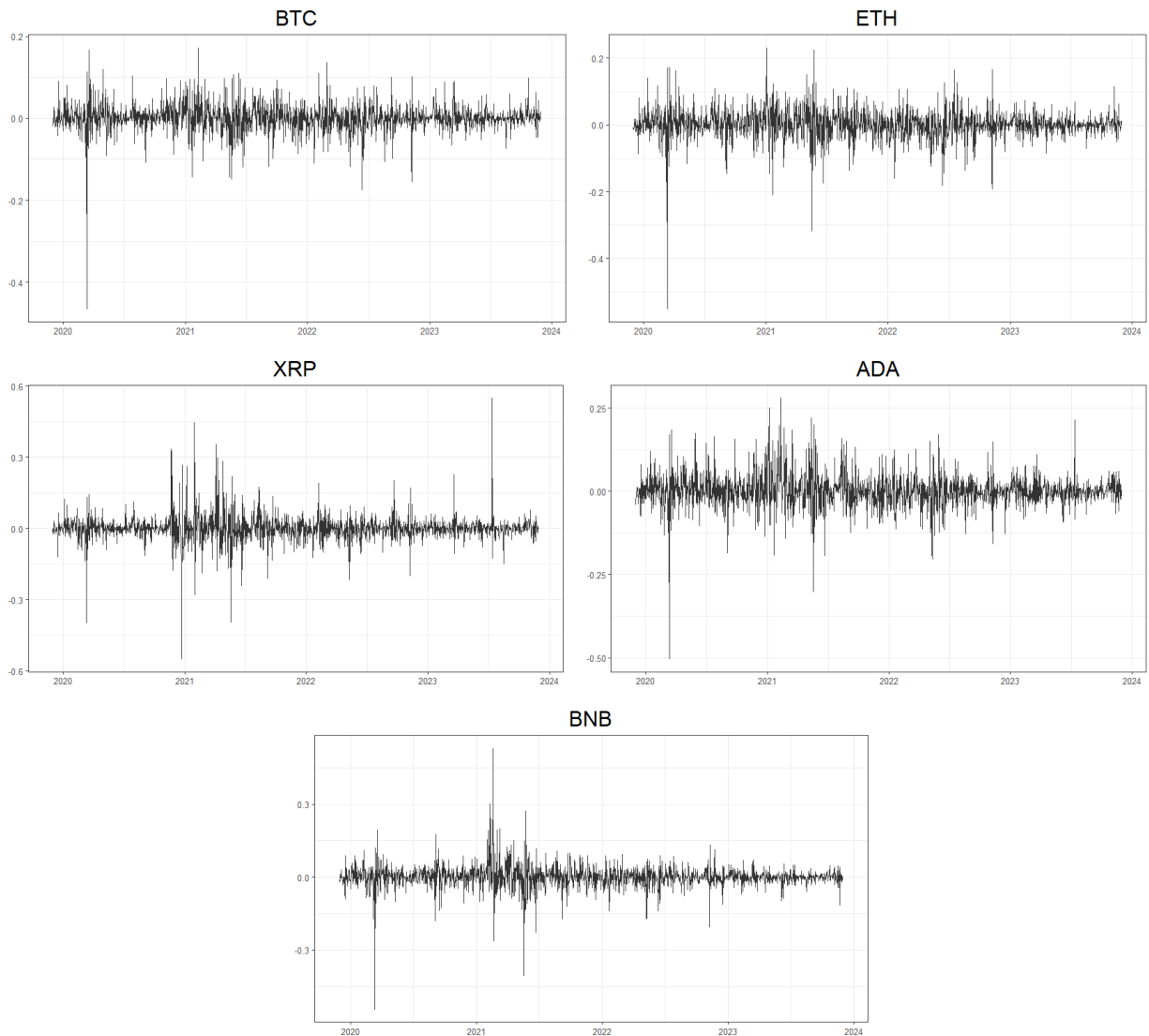
3.1 Duomenų analizė

Portfelio formavimui pasirinktos penkios didžiausios kriptovaliutos pagal rinkos kapitalizaciją - "Bitcoin", "Ethereum", "XRP", "Cardano" ir "BNB". Modelis kuriamas naudojant dienos uždarymo kainas nuo 2019 m. lapkričio 30 d. iki 2023 m. lapkričio 30 d., o grįžtamasis testavimas atliekamas 2023 m. gruodžio 1d. - 2025 m. lapkričio 30d. laikotarpiu. Toliau pateikiami visų penkių kriptovaliutų kainų grafikai:



2 pav.: Paveikslėlyje pateikti minėtų kriptovaliutų dienos uždarymo kainų grafikai, o raudona vertikali linija žymi 2023 m. lapkričio 30 d. ribą tarp modelio kurimo ir grįžtamojo testavimo laikotarpių

Tolimesnei analizei naudojamas tik modelio kūrimo laikotarpis (2019 m. lapkričio 30 d. - 2023 m. lapkričio 30 d.). Toliau apskaičiuojamos penkių kriptovaliutų logaritminės grąžos, kurios geriau atspindi santykinius pokyčius tarp aktyvų ir paprastai yra artimesnės stacionarumui. Žemiau pateikiami nagrinėjamų kriptovaliutų logaritminių grąžų laiko eilučių grafikai.



3 pav.: Paveikslėlyje pateikti nagrinėjamų kriptovaliutų logaritminių grąžų laiko eilučių grafikai

3.2 ARMA-GJR-GARCH modelio pritaikymas

Šioje dalyje taikomas teorinėje dalyje aptartas ARMA-GJR-GARCH modelis. Logaritminių grąžų grafikuose (žr. 4) matyti, kad dispersija laikui bėgant kinta. Taikant *R-vine* kopulas, priklausomybių struktūrai įvertinti reikalingi tolydūs atsitiktiniai dydžiai, todėl logaritminės grąžos pirmiausia filtruojamos ARMA-GJR-GARCH modeliu ir tolesnei analizei naudojamos standartizuotos liekanos. ARMA komponentė naudojama vidurkio dinamikai (autokoreliacijai) modeliuoti, o GJR-GARCH dalis - sąlyginei disperijai modeliuoti.

Pirmiausia, kiekvienai kriptovaliutai (BTC, ETH, XRP, ADA ir BNB) iš logaritminių gražų parenkamos ARMA modelio eilės tvarkos p ir q , o sąlyginei dispersijai modeliuoti taikomas asimetrinis GJR-GARCH(1,1) modelis. Optimali ARMA(p,q)-GJR-GARCH(1,1) modelio specifikacija kiekvienu atveju nustatoma minimizuojant Akaike informacinį kriterijų (AIC), paiešką apribojant iki $p \leq 4$ ir $q \leq 4$.

Duomenys	Modelis	AIC
BTC	ARMA(4, 4)-GJR-GARCH(1,1)	-3.969144
ETH	ARMA(3, 3)-GJR-GARCH(1,1)	-3.512873
XRP	ARMA(1, 4)-GJR-GARCH(1,1)	-3.126461
ADA	ARMA(4, 3)-GJR-GARCH(1,1)	-3.213226
BNB	ARMA(2, 4)-GJR-GARCH(1,1)	-3.614953

1 lentelė: Logaritminių gražų ARMA(p,q)-GJR-GARCH modeliai su mažiausiomis AIC reikšmėmis

Lentelėje (žr. 1) pateiktos AIC reikšmės buvo suskaičiuotos naudojant *rugarch* paketą, kuris normalizuoja AIC reikšmes vienam stebėjimui, todėl reikšmės yra nedidelės ir neigiamos. Toliau pateikiami gautų modelių parametrai ir jų standartinės paklaidos.

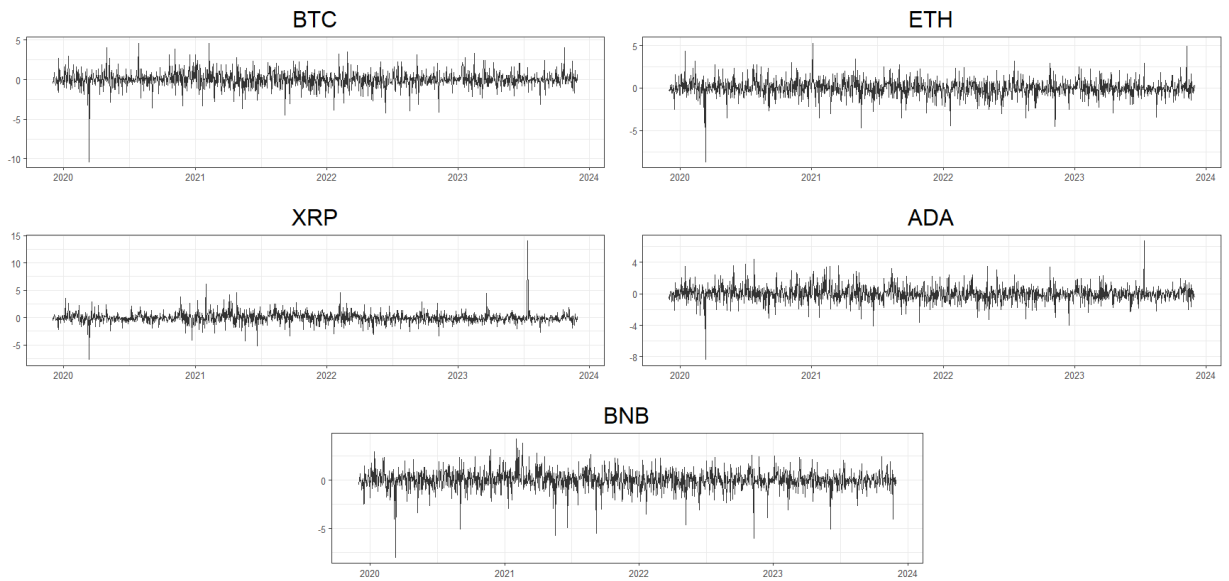
Parametras	BTC		ETH		XRP	
	Įvertis	St. pakl.	Įvertis	St. pakl.	Įvertis	St. pakl.
μ	0.000706	0.000138	0.001662	0.000949	0.000988	0.000037
ϕ_1	-1.417355	0.000166	0.000988	0.021950	0.990672	0.001860
ϕ_2	-0.117692	0.000033	-0.147730	0.027797	-	-
ϕ_3	0.746712	0.000097	-0.660968	0.021656	-	-
ϕ_4	0.206790	0.000041	-	-	-	-
θ_1	1.381818	0.000169	-0.593486	0.003429	-1.110480	0.000117
θ_2	0.165215	0.000056	0.202704	0.001014	0.151607	0.000036
θ_3	-0.577515	0.000062	0.636383	0.000056	-0.202239	0.000021
θ_4	-0.087264	0.000050	-	-	0.160472	0.000016
ω	0.000064	0.000015	0.000041	0.000013	0.000947	0.000095
α_1	0.046700	0.017363	0.107673	0.019409	0.794121	0.111576
β_1	0.844220	0.022652	0.875911	0.019145	0.200391	0.051061
γ_1	0.154916	0.035491	0.020116	0.024833	0.008976	0.139974

2 lentelė: BTC, ETH, XRP ARMA-GJR-GARCH modelių parametrų reikšmės ir jų standartinės paklaidos

Parametras	ADA		BNB	
	Įvertis	Standartinė paklaida	Įvertis	Standartinė paklaida
μ	0.000620	0.001228	0.000610	0.000832
ϕ_1	-0.415184	0.022061	-1.064611	0.007984
ϕ_2	-0.184234	0.006051	-0.969075	0.010414
ϕ_3	0.760284	0.004013	-	-
ϕ_4	0.080119	0.026047	-	-
θ_1	0.357492	0.000849	1.004643	0.008320
θ_2	0.191944	0.002677	0.931608	0.000079
θ_3	-0.732047	0.000039	-0.033959	0.005951
θ_4	-	-	0.042705	0.010180
ω	0.000139	0.000034	0.000052	0.000014
α_1	0.136457	0.024975	0.166146	0.023309
β_1	0.799514	0.029858	0.814127	0.021958
γ_1	0.067120	0.035635	0.037455	0.029507

3 lentelė: ADA, BNB ARMA-GJR-GARCH modelių parametruų reikšmės ir jų standartinės paklaidos

Pritaikius ARMA-GJR-GARCH modelį, apskaičiuojamos standartizuotos liekanos $\hat{z}_t = \frac{r_t - \hat{\mu}_t}{\hat{\sigma}_t}$, kur $\hat{\mu}_t$ - sąlyginis vidurkis, o $\hat{\sigma}_t$ - sąlyginis standartinis nuokrypis momentu t . Toliau pateikiami visų nagrinėjamų kriptovaliutų standartizuotų liekanų laiko eilučių grafikai.



4 pav.: Paveikslėlyje pateikti nagrinėjamų kriptovaliutų standartizuotų liekanų laiko eilučių grafikai

Siekiant įvertinti, ar ARMA-GJR-GARCH modeliai pašalina autoregresinio sąlyginio heteroskedastiškumo (ARCH) efektą, standartizuotoms liekanoms $\hat{z}_t$ bus taikomas *Engle's ARCH* testas iš *fdm*A paketo. Šio testo hipotezės:

Nulinė hipotezė (H_0): liekanose nepasireiškia ARCH efektas
 Alternatyvi hipotezė (H_1): liekanose pasireiškia ARCH efektas

Gauti testo rezultatai pateikiami žemiau esančioje lentelėje:

	BTC	ETH	XRP	ADA	BNB
<i>Engle's ARCH</i> p-reikšmė	0.7151	0.8131	0.6781	0.6944	0.4709

4 lentelė: Standartizuotų liekanų *Engle's ARCH* testo p-reikšmės

Pagal lentelėje (žr. 4) pateiktas *Engle's ARCH* testo p -reikšmių matyti, kad visoms nagrinėjamos kriptovaliutoms nulinė hipotezė neatmetama (visos p -reikšmės didesnės nei 0.05). Tai reiškia, jog standartizuotose liekanose statistiškai reikšmingo arch efekto nenustatyta, todėl galima teigti, kad pritaikyti ARMA-GJR-GARCH modeliai pašalino autoregresinio sąlyginio heteroskedastiškumo efektą.

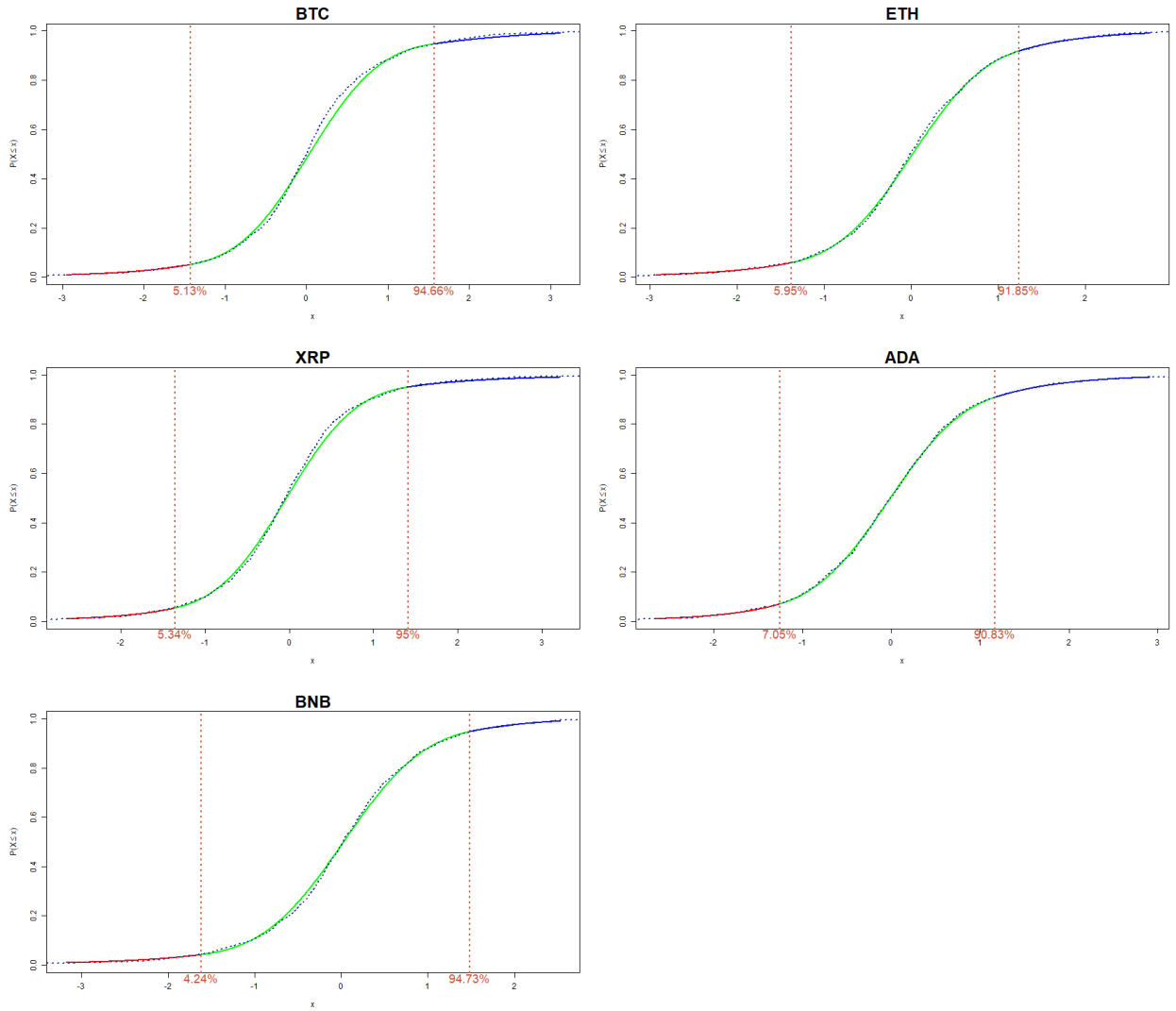
3.3 Dviejų uodegų apibendrinto Pareto pasiskirstymo mišinio modelio pritaikymas

Toliau, siekiant gauti šių standartizuotų liekanų $\hat{z}_t$ marginaliuosius skirstinius ir tiksliai įvertinti uodegų elgseną, taikomas teorinėje dalyje aprašytas (žr. 34) dviejų uodegų apibendrinto Pareto pasiskirstymo mišinio modelis. Kiekvienai kriptovaliutai šio modelio lūžio taškai u_l , u_r , normalio skirstinio parametrai m ir s , bei uodegų formos parametrai ξ_l ir ξ_r įvertinami didžiausio tikėtimumo metodu, naudojant *GNG_fit* funkciją iš *mistr* paketo. Gauti pasiskirstymo modelio parametrai pateikiami žemiau esančioje lentelėje.

Duomenys	u_l	u_r	m	s	ξ_l	ξ_r
BTC	-1.429007	1.566015	0.024271	0.696595	0.106392	-0.290601
ETH	-1.371348	1.242899	0.008212	0.695000	0.082785	-0.049945
XRP	-1.362268	1.412508	-0.040252	0.663717	0.125210	0.164390
ADA	-1.256788	1.166326	-0.029972	0.691490	0.102608	0.006143
BNB	-1.618074	1.482881	0.024583	0.745507	0.166845	-0.150205

5 lentelė: Nagrinėjamų kriptovaliutų GNG modelio parametrų įverčiai

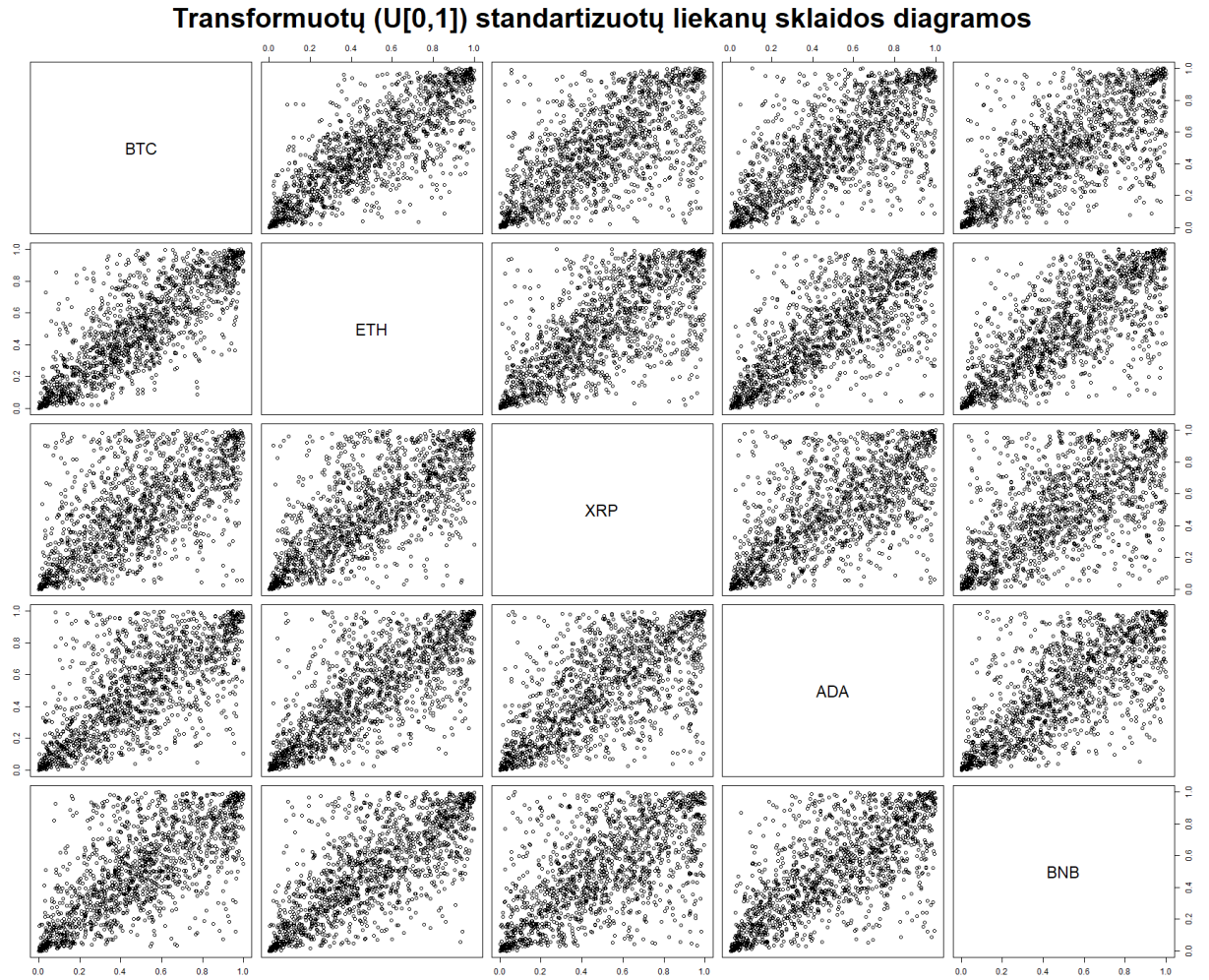
Toliau, naudojant lentelėje (žr 5) pateiktais GNG modelio parametrų įverčiais, apskaičiuojamos kiekvienos kriptovaliutos skirstinio funkcijos.



5 pav.: Paveikslėlyje pateikti nagrinėjamų kriptovaliutų GNG modelio skirstinių funkcijos

Šie grafikai (žr. 5) leidžia vizualiai įvertinti, kaip pasirinktas GDP-Normal-GDP mišinys atitinka empirinę skirstinį. Punktyrinės vertikalios linijos žymi lūžio taškus u_l ir u_r , o spalvos atitinka modelio sudedamąsias dalis: raudona - kairioji GDP uodega, žalia - normalųjį skirstinį, mėlyna - dešinioji GDP uodega, o juoda kreivė žymi empirinę skirstinį.

Toliau, remiantis pritaikyta pasiskirstymo funkcija, standartizuotos liekanos transformuojamos į $[0, 1]$ intervalo atsitiktinius dydžius, kurie yra artimi tolygiajam skirstiniui.

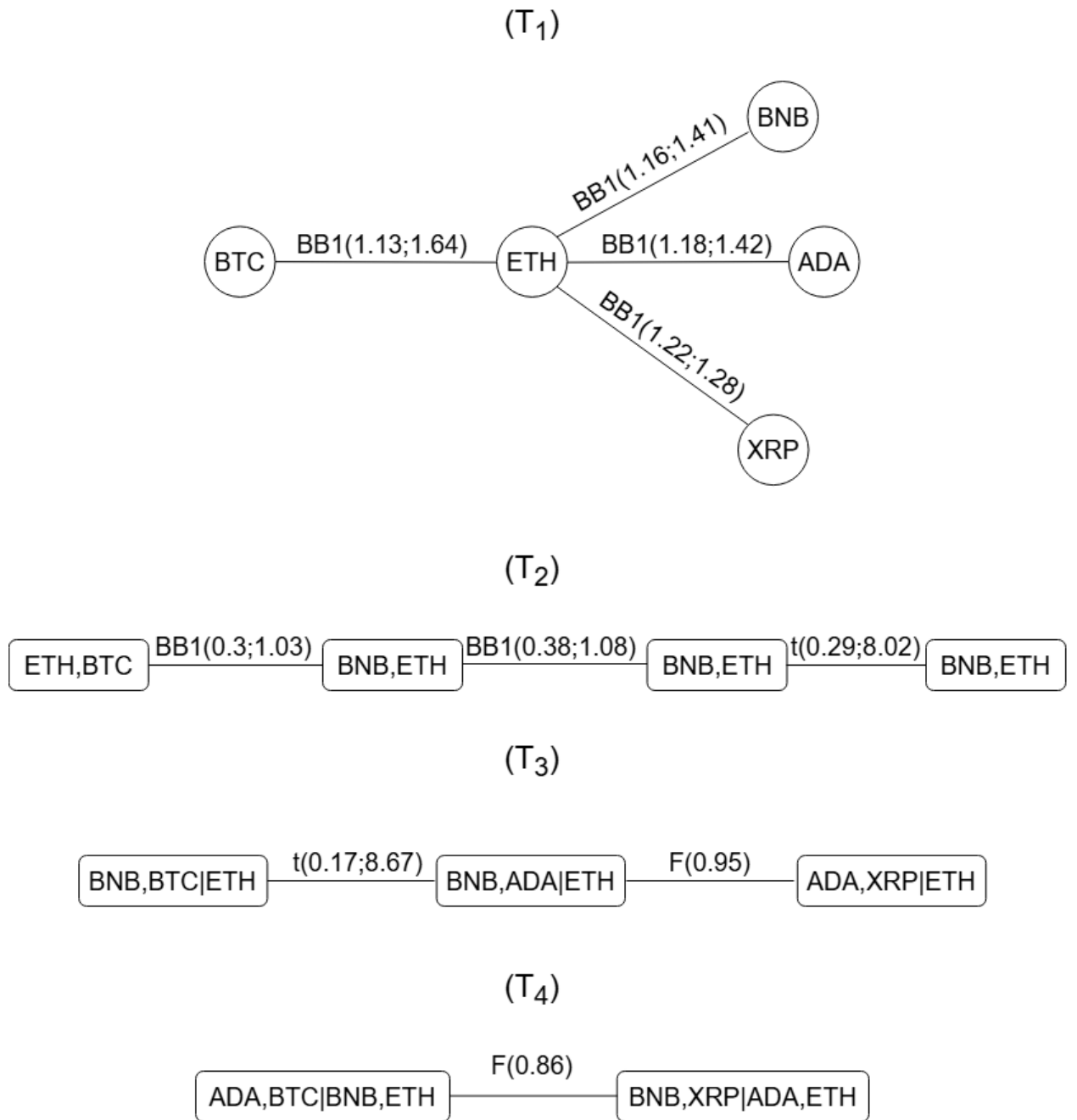


6 pav.: Paveikslėlyje pateikti nagrinėjamų kriptovaliutų transformuotų standartizuotų liekanų sklaidos diagramos

Paveikslėlyje (žr 6) pateiktos transformuotų standartizuotų liekanų porinės sklaidos diagramos vizualiai atskleidžia teigiamą priklausomybę tarp nagrinėjamų kriptovaliutų porų. Taip pat matyti, kad priklausomybės nėra linijinės, t.y. priklausomybės yra stipresnės kraštuose (uodegose) nei centre, todėl priklausomybės struktūrai modeliuoti pasirenkamos *vine* kopulos.

3.4 Reguliariosios *vine* struktūros parinkimas

Šiame skyrelyje, naudojant transformuotas standartizuotas liekanas, sudaroma reguliariosios *vine* kopulos struktūra, leidžianti daugiamates priklausomybes modeliuoti pasitelkiant porines kopulas. Toliau parenkama optimali *vine* medžių seka, o kiekvienai briaunai parenkama tinkamiausia kopula pagal Akaičės informacinį kriterijų (AIC) ir įvertinami jos parametrai, taikant didžiausio tikėtinumo metodą. Ši dalis bus įgyvendinta naudojant *VineCopula* paketą.

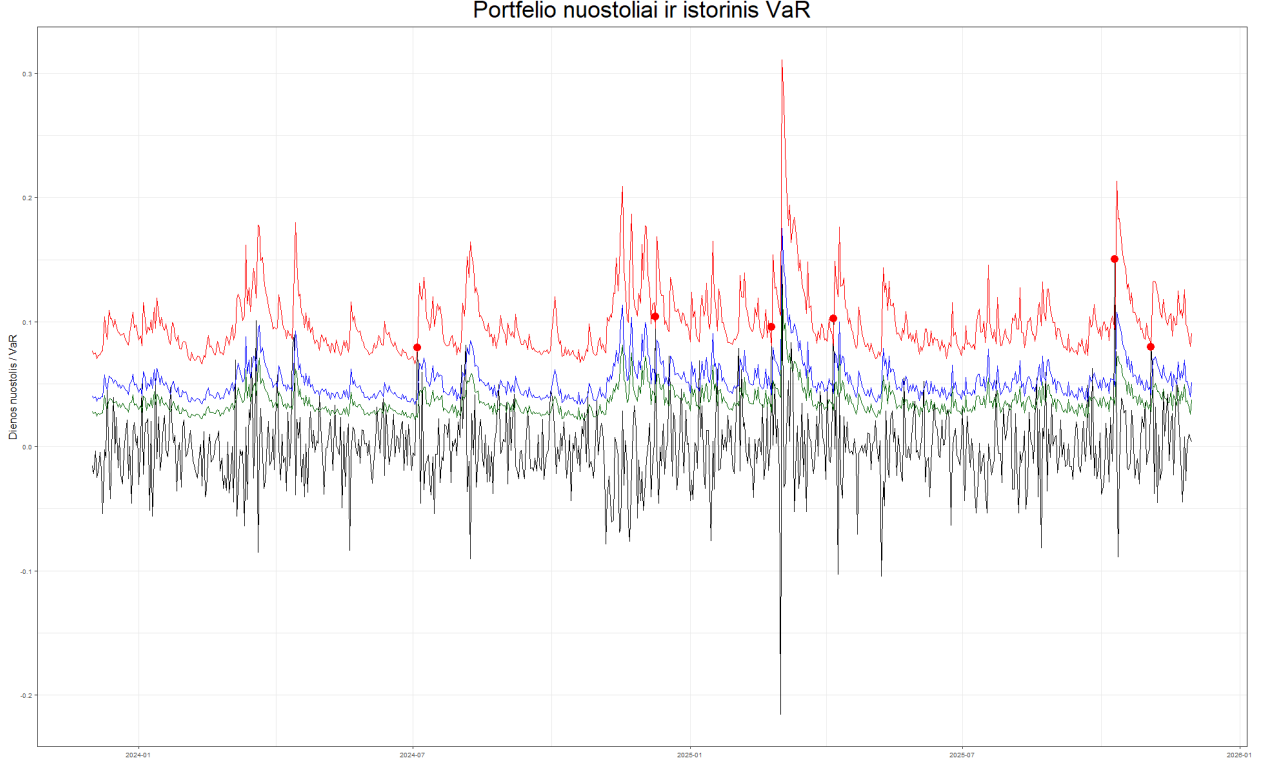


7 pav.: Paveikslėlyje pateikta gauta reguliarioji *vine* struktūra

Paveikslėlyje (žr. 7) pavaizduota parinkta reguliariosios *vine* kopulos struktūra ir kiekvienos briaunos įvertinta porinė kopula su jos parametrais. Pirmame medyje T_1 viršūnės atitinka penkias nagrinėjamas kriptovaliutas, o briaunos parinktos pagal didžiausias Kendall's τ reikšmes (analogiškai parenkant Spearman's ρ gaunama ta pati struktūra), todėl jos atspindi stipriausias porines priklausomybes. Prie kiekvienos briaunos nurodoma parinkta kopulų šeima (BB1, t ir Frank) ir jos įverčiai.

3.5 Vertės pokyčio rizikos skaičiavimas ir testavimas

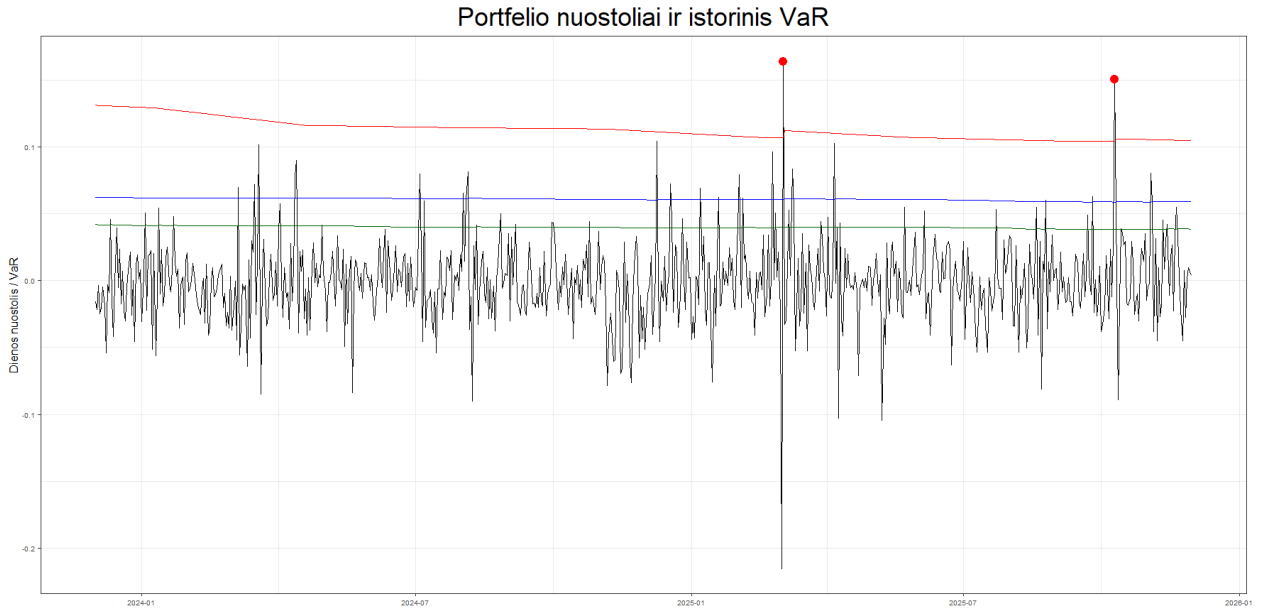
Šiame skyrelyje vertės pokyčio rizika (VaR) skaičiuojama vienos dienos horizontui. Kiekvienam testinės imties momentui t sudaromos vienos dienos į priekį slenkančios prognozės $\mu_{i,t+1}$ ir $\sigma_{i,t+1}$ ($i \in \{\text{BTC,ETH,XRP,ADA,BNB}\}$), tuomet iš pritaikytos *vine* kopulos sugeneruojama $N = 10000$ simuliacijų. Jie transformuojami į standartizuotas liekanas, iš kurių gaunamos simuliuotos grąžos $r_{i,t+1}^{(s)} = \mu_{i,t+1} + \sigma_{i,t+1} z_{i,t+1}^{(s)}$. Portfelio grąža gaunama taikant vienodus svorius $w_i = \frac{1}{5}$ kiekvienai kriptovaliutai, t.y. $r_{t+1}^{(s)} = \sum_i w_i r_{i,t+1}^{(s)}$.



8 pav.: Paveikslėlyje pateikti portfelio dienos nuostoliai ir VaR

Paveiksle (žr.8) pavaizduoti portfelio dienos nuostoliai (juoda kreivė) ir apskaičiuotos VaR ribos su skirtingais patikimumo lygiais: $VaR_{0.99}$ (raudona), $VaR_{0.95}$ (mėlyna) ir $VaR_{0.90}$ (žalia). Raudoni taškai žymi nuokrypius - dienas, kai faktinis nuostolis viršijo $VaR_{0.99}$ ribą.

Toliau, siekiant įvertinti, ar apskaičiuoti VaR įverčiai atitinka pasirinktus patikimumo lygius, atliekamas jų testavimas naudojant Kupiec testą. Šis testas tikrina, ar faktinių nuokrypių dalis I/T yra statistiškai suderinama su $1-\alpha$. Palyginimui, taip pat bus skaičiuojamas istorinis VaR, paremtas empiriniais portfelio nuostoliais $L_t = -r_t$, kiekvienai testinės imties dienai VaR apskaičiuojamas naudojant visus iki tos dienos turimus stebėjimus.



9 pav.: Paveikslėlyje pateikti portfelio dienos nuostoliai ir istorinis VaR

Paveiksle (žr. 9) pateikti portfelio dienos nuostoliai ir istorinio *VaR* ribos skirtingiems patikimumo lygiams. Toliau pateikiama lentelė *vine* kopulos paremto *VaR* modelio ir istorinio *VaR* modelio grįžtamojo testavimo rezultatai.

Modelis	N	Nuokrypiai	α	I/N	Kupiec testo p-reikšmė
<i>Vine</i>	731	6	0.01	0.0082	0.6153
<i>Vine</i>	731	36	0.05	0.0492	0.9255
<i>Vine</i>	731	68	0.10	0.0930	0.5251
Istorinis	731	2	0.01	0.0027	0.0193
Istorinis	731	23	0.05	0.0315	0.0139
Istorinis	731	55	0.10	0.0752	0.0201

6 lentelė: *Vine* ir istorinio *VaR* grįžtamasis testavimas

Pagal lentelėje (žr. 6) pateiktas Kupiec testo p-reikšmes matyti, kad *vine* kopulos metodu gautas *VaR* modelis visuose nagrinėjamuose patikimumo lygiuose nulinės hipotezės neatmeta ($p > 0.05$). Tuo tarpu istorinio *VaR* atveju visuose lygiuose nulinė hipotezė atmetama ($p < 0.05$). Apibendrinant, gauti rezultatai rodo, kad *vine* kopula paremtas *VaR* modelis yra tinkamas rizikos vertinimui, kadangi jo prognozuojamos *VaR* ribos yra suderinamos su stebimų nuokrypių skaičiumi, tuo tarpu istorinis *VaR* yra pernelyg konservatyvus ir pervertina portfelio riziką.

4 Išvados

Šiame darbe buvo įvertinta kriptovaliutų portfelio rizika, pritaikant jungtinį ARMA-GJR-GARCH ir reguliariųjų *vine* kopulų pagrindu sudarytą VaR modelį, taip pat empiriškai patikrintas jo prognozių patikimumas taikant grįžtamąjį testavimą. Pirmiausia kiekvienos kriptovaliutos grąžoms buvo pritaikyti ARMA-GJR-GARCH modeliai ir gautos standartizuotos liekanos, kurių marginaliniai pasiskirstymai aprašyti dviejų uodegų mišinio (GNG) modeliu. Tuomet liekanų tarpusavio priklausomybės struktūra modeliuota reguliariąja *vine* kopula, parenkant struktūrą taikant [7] algoritmą, o porinės kopulų šeimos - pagal AIC kriterijų. Remiantis įvertintu jungtiniu modeliu buvo atliktos portfelio grąžų simuliacijos ir apskaičiuotas VaR skirtingais patikimumo lygiais.

Gauti rezultatai rodo, kad sukurtas modelis sudaro nuoseklų ir empiriškai pagrįstą kriptovaliutų portfelio VaR vertinimo metodą: grįžtamojo testavimo metu *vine* kopulomis paremtas VaR buvo statistiškai suderinamas su pasirinktais patikimumo lygiais, o nuokrypių dažnis išliko artimas tikėtinoms reikšmėms. Tuo tarpu istorinis VaR šiame tyrime pasirodė pernelyg konservatyvus.

Gauti rezultatai priklauso nuo pasirinkto laikotarpio, portfelio svorių ir pasirinktų kopulų šeimų aibės. Tolimesniuose tyrimuose šį darbą galima praplėsti vertinant papildomus rizikos matus, nagrinėjant laiko kintančias kopulas ar dinamines *vine* struktūras bei tiriant modelio jautrumą skirtingoms portfelio sudėtimis ir svorių parinkimo strategijoms.

Literatūra

- [1] Nelsen, Roger B. An Introduction to Copulas. Springer, 2006.
- [2] Sklar, M., 1959. Fonctions de repartition an dimensions et leurs marges. Publ. inst. statist. univ. Paris, 8, pp.229–231.
- [3] Rachev, S. T. “Modelling Dependence with Copulas and Applications to Risk Management.” Handbook of Heavy Tailed Distributions in Finance. vol. 1. The Netherlands: Elsevier Science and Technology, 2003.
- [4] Kurowicka, Dorota, and Harry Joe. Dependence Modeling: Vine Copula Handbook. 1st ed. Hackensack, N.J: World Scientific, 2011.
- [5] Manstavičius, Eugenijus. Grafių teorija: paskaitų konspektas. Vilnius: Vilniaus universitetas, 2001.
- [6] Bedford, Tim, and Roger M Cooke. “Vines: A New Graphical Model for Dependent Random Variables.” The Annals of Statistics 30.4 (2002): 1031–1068.
- [7] Dißmann, J et al. “Selecting and Estimating Regular Vine Copulae and Application to Financial Returns.” Computational statistics data analysis 59 (2013): 52–69.
- [8] Morales Napoles, Oswaldo. (2016). About the number of vines and regular vines on n nodes.
- [9] Akaike, Hirotugu. "A new look at the statistical model identification." IEEE transactions on automatic control 19.6 (1974): 716–723.
- [10] Zhao, Xin Scarrott, Carl Reale, Marco Oxley, Les. (2009). Bayesian Extreme Value Mixture Modelling for Estimating VaR. University of Canterbury, Department of Economics and Finance, Working Papers in Economics.
- [11] Hidayana, R., Sukono, H.N. and Napitupulu, H., 2021, September. Arma-Gjr-Garch Model For Determining Value-At-Risk And Back Testing Of Some Stock Returns. In Proceedings Of The Second Asia Pacific International Conference On Industrial Engineering And Operations Management Surakarta.
- [12] Nakamoto, Satoshi, Bitcoin: A Peer-to-Peer Electronic Cash System (August 21, 2008).
- [13] Corbet, Shaen et al. “Exploring the Dynamic Relationships between Cryptocurrencies and Other Financial Assets.” Economics letters 165 (2018): 28–34.
- [14] Bruzgė, Rasa et al. “Stylized Facts, Volatility Dynamics and Risk Measures of Cryptocurrencies / Rasa Bruzgė, Jurgita Černevičienė, Alfreda Šapkauskienė, Aida Mačerinskienė, Saulius Masteika, Kęstutis Driaunys.” Journal of business economics and management. (2023): 527–550.
- [15] Kupiec, Paul, Techniques for Verifying the Accuracy of Risk Measurement Models. The J. Of Derivatives, Vol. 3 No. 2, Winter 1995.
- [16] Glosten, Lawrence R, Ravi Jagannathan, and David E Runkle. “On the Relation between the Expected Value and the Volatility of the Nominal Excess Return on Stocks.” The Journal of finance (New York) 48.5 (1993): 1779–1801.

Priedai

Praktinės dalies R kodas

```
#### Uzkrauname darbe naudojamus paketus ir ikeliamė duomenis ####
#Naudojami paketai
library(quantmod)
library(xts)
library(ggplot2)
library(patchwork)
library(rugarch)
library(mistr)
library(VineCopula)
library(fDMA)

# Nustatome datų intervalus
start_date <- as.Date("2019-11-30")
train_end   <- as.Date("2023-11-30")
end_date    <- as.Date("2025-11-30")

# Ikeliamė BTC, ETH, XRP, ADA ir BNB duomenis iš Yahoo Finance
getSymbols(c("BTC-USD", "ETH-USD", "XRP-USD", "ADA-USD", "BNB-USD"), src =
  "yahoo", from = start_date, to = end_date)
BTC <- 'BTC-USD'
ETH <- 'ETH-USD'
XRP <- 'XRP-USD'
ADA <- 'ADA-USD'
BNB <- 'BNB-USD'

# Patikriname ar yra tų reikšmių
sujungti_duomenys <- merge(Cl(BTC), Cl(ETH), Cl(XRP), Cl(ADA), Cl(BNB),
  join = "inner")
sum(is.na(sujungti_duomenys))
BTC <- sujungti_duomenys[, 1]
ETH <- sujungti_duomenys[, 2]
XRP <- sujungti_duomenys[, 3]
ADA <- sujungti_duomenys[, 4]
BNB <- sujungti_duomenys[, 5]

#### Sukuriame kriptovaliutų kainų grafiką ####
data4plot <- data.frame(
  Date = index(BTC),
  BTC = coredata(BTC),
  ETH = coredata(ETH),
  XRP = coredata(XRP),
  ADA = coredata(ADA),
  BNB = coredata(BNB)
)
split_date <- as.Date("2023-11-30")
vline <- geom_vline(xintercept = split_date, colour = "red", linewidth =
  0.4, linetype = "dashed")

#BTC
p1 <- ggplot(data4plot, aes(x = Date, y = BTC.USD.Close)) +
  geom_line(colour = "#333333", size = 0.6) +
  labs(title = "Bitcoin",
    x = "",
    y = "") +
  theme_bw() +
  theme(plot.title = element_text(size = 22, hjust = 0.5)) +
  vline
```

```

#ETH
p2 <- ggplot(data4plot, aes(x = Date, y = ETH.USD.Close)) +
  geom_line(colour = "#333333", size = 0.6) +
  labs(title = "Ethereum",
        x = "",
        y = "") +
  theme_bw() +
  theme(plot.title = element_text(size = 22, hjust = 0.5)) +
  vline

#XRP
p3 <- ggplot(data4plot, aes(x = Date, y = XRP.USD.Close)) +
  geom_line(colour = "#333333", size = 0.6) +
  labs(title = "XRP",
        x = "",
        y = "") +
  theme_bw() +
  theme(plot.title = element_text(size = 22, hjust = 0.5)) +
  vline

#ADA
p4 <- ggplot(data4plot, aes(x = Date, y = ADA.USD.Close)) +
  geom_line(colour = "#333333", size = 0.6) +
  labs(title = "Cardano",
        x = "",
        y = "") +
  theme_bw() +
  theme(plot.title = element_text(size = 22, hjust = 0.5)) +
  vline

#BNB
p5 <- ggplot(data4plot, aes(x = Date, y = BNB.USD.Close)) +
  geom_line(colour = "#333333", size = 0.6) +
  labs(title = "BNB",
        x = "",
        y = "") +
  theme_bw() +
  theme(plot.title = element_text(size = 22, hjust = 0.5)) +
  vline

#Bendras grafikas
(p1 | p2) / (p3 | p4) / (p5 | plot_spacer())

#### Dienos uzdarymo kainas paverciame logaritminemis grazomis ####
log_grazos <- function(x) na.omit(diff(log(x)))
BTC_log_grazos<-log_grazos(BTC)
ETH_log_grazos<-log_grazos(ETH)
XRP_log_grazos<-log_grazos(XRP)
ADA_log_grazos<-log_grazos(ADA)
BNB_log_grazos<-log_grazos(BNB)

# Atskiriame duomenis pagal modelio kurimo ir testavimo intervalus
train_idx <- index(BTC_log_grazos) <= train_end
test_idx <- index(BTC_log_grazos) > train_end

n_test <- sum(test_idx)

BTC_log_grazos_train <- BTC_log_grazos[train_idx]
ETH_log_grazos_train <- ETH_log_grazos[train_idx]

```

```

XRP_log_grazos_train <- XRP_log_grazos[train_idx]
ADA_log_grazos_train <- ADA_log_grazos[train_idx]
BNB_log_grazos_train <- BNB_log_grazos[train_idx]

# Sukuriame kriptovaliutu logaritminių gražu histograma
BTC_log_grazos_df <- data.frame(Date = index(BTC_log_grazos_train), BTC.
  USD.Close = coredata(BTC_log_grazos_train))
ETH_log_grazos_df <- data.frame(Date = index(ETH_log_grazos_train), ETH.
  USD.Close = coredata(ETH_log_grazos_train))
XRP_log_grazos_df <- data.frame(Date = index(XRP_log_grazos_train), XRP.
  USD.Close = coredata(XRP_log_grazos_train))
ADA_log_grazos_df <- data.frame(Date = index(ADA_log_grazos_train), ADA.
  USD.Close = coredata(ADA_log_grazos_train))
BNB_log_grazos_df <- data.frame(Date = index(BNB_log_grazos_train), BNB.
  USD.Close = coredata(BNB_log_grazos_train))

visos_log_grazos <- c(BTC_log_grazos_df$BTC.USD.Close,
  ETH_log_grazos_df$ETH.USD.Close, XRP_log_grazos_df$XRP.USD.Close,
  ADA_log_grazos_df$ADA.USD.Close, BNB_log_grazos_df$BNB.USD.Close)

breaks_1 <- seq(min(visos_log_grazos, na.rm = TRUE), max(visos_log_grazos,
  na.rm = TRUE), length.out = 51)

hist1 <- ggplot(BTC_log_grazos_df, aes(x = BTC.USD.Close)) +
  geom_histogram(breaks = breaks_1, fill = "#333333", color = "black") +
  labs(title = "Bitcoin", x = "", y = "") +
  theme_bw()

hist2 <- ggplot(ETH_log_grazos_df, aes(x = ETH.USD.Close)) +
  geom_histogram(breaks = breaks_1, fill = "#333333", color = "black") +
  labs(title = "Ethereum", x = "", y = "") +
  theme_bw()

hist3 <- ggplot(XRP_log_grazos_df, aes(x = XRP.USD.Close)) +
  geom_histogram(breaks = breaks_1, fill = "#333333", color = "black") +
  labs(title = "XRP", x = "", y = "") +
  theme_bw()

hist4 <- ggplot(ADA_log_grazos_df, aes(x = ADA.USD.Close)) +
  geom_histogram(breaks = breaks_1, fill = "#333333", color = "black") +
  labs(title = "Cardano", x = "", y = "") +
  theme_bw()

hist5 <- ggplot(BNB_log_grazos_df, aes(x = BNB.USD.Close)) +
  geom_histogram(breaks = breaks_1, fill = "#333333", color = "black") +
  labs(title = "BNB", x = "", y = "") +
  theme_bw()

(hist1 | hist2) / (hist3 | hist4) / (hist5 | plot_spacer())

# Sukuriame kriptovaliutu logaritminių gražu grafika
p1 <- ggplot(BTC_log_grazos_df, aes(x = Date, y = BTC.USD.Close)) +
  geom_line(color = "#333333", size = 0.6) +
  labs(title = "BTC", x = "", y = "") +
  theme_bw() +
  theme(plot.title = element_text(size = 22, hjust = 0.5))

p2 <- ggplot(ETH_log_grazos_df, aes(x = Date, y = ETH.USD.Close)) +
  geom_line(color = "#333333", size = 0.6) +
  labs(title = "ETH", x = "", y = "") +

```

```

theme_bw() +
theme(plot.title = element_text(size = 22, hjust = 0.5))

p3 <- ggplot(XRP_log_grazos_df, aes(x = Date, y = XRP.USD.Close)) +
  geom_line(color = "#333333", size = 0.6) +
  labs(title = "XRP", x = "", y = "") +
  theme_bw() +
  theme(plot.title = element_text(size = 22, hjust = 0.5))

p4 <- ggplot(ADA_log_grazos_df, aes(x = Date, y = ADA.USD.Close)) +
  geom_line(color = "#333333", size = 0.6) +
  labs(title = "ADA", x = "", y = "") +
  theme_bw() +
  theme(plot.title = element_text(size = 22, hjust = 0.5))

p5 <- ggplot(BNB_log_grazos_df, aes(x = Date, y = BNB.USD.Close)) +
  geom_line(color = "#333333", size = 0.6) +
  labs(title = "BNB", x = "", y = "") +
  theme_bw() +
  theme(plot.title = element_text(size = 22, hjust = 0.5))

#Bendras grafikas
(p1 | p2) / (p3 | p4) / (p5 | plot_spacer())

#### Parenkame geriausias p ir q reikšmes is ARMA(p,q)-GJR-GARCH(1,1)
modelio pagal AIC ####
# Funkcija, kuri parenka geriausia ARMA(p,q) pagal AIC is pilno ARMA-GJR-
GARCH(1,1) modelio
gauti_arma_gjr_eile <- function(duomenys, max_p = 4, max_q = 4, dist = "
norm", out.sample = 0) {
  y <- as.numeric(duomenys)
  geriausias_aic <- Inf
  geriausias_p <- 0
  geriausias_q <- 0
  for (p in 0:max_p) {
    for (q in 0:max_q) {
      spec <- ugarchspec(
        mean.model = list(armaOrder = c(p, q)),
        variance.model = list(model = "gjrGARCH", garchOrder = c(1, 1)),
        distribution.model = dist
      )
      fit <- ugarchfit(spec = spec, data = y, out.sample = out.sample)
      aic_val <- try(Infocriteria(fit)[1], silent = TRUE)
      if (inherits(aic_val, "try-error") || length(aic_val) == 0 || !is.
        finite(aic_val)) next
      if (aic_val < geriausias_aic) {
        geriausias_aic <- aic_val
        geriausias_p <- p
        geriausias_q <- q
      }
    }
  }
  list(p = geriausias_p, q = geriausias_q, aic = geriausias_aic)
}

# Parenkame ARMA(p,q) pagal pilno ARMA-GJR-GARCH(1,1) modelio AIC
btc_gjr <- gauti_arma_gjr_eile(BTC_log_grazos_train, max_p = 4, max_q = 4,
  dist = "norm")
eth_gjr <- gauti_arma_gjr_eile(ETH_log_grazos_train, max_p = 4, max_q = 4,
  dist = "norm")

```

```

xrp_gjr <- gauti_arma_gjr_eile(XRP_log_grazos_train, max_p = 4, max_q = 4,
  dist = "norm")
ada_gjr <- gauti_arma_gjr_eile(ADA_log_grazos_train, max_p = 4, max_q = 4,
  dist = "norm")
bnb_gjr <- gauti_arma_gjr_eile(BNB_log_grazos_train, max_p = 4, max_q = 4,
  dist = "norm")

# Issitraukiame p ir q
p_BTC <- btc_gjr$p; q_BTC <- btc_gjr$q
p_ETH <- eth_gjr$p; q_ETH <- eth_gjr$q
p_XRP <- xrp_gjr$p; q_XRP <- xrp_gjr$q
p_ADA <- ada_gjr$p; q_ADA <- ada_gjr$q
p_BNB <- bnb_gjr$p; q_BNB <- bnb_gjr$q

# Atspausdiname rezultatus
cat("BTC: geriausias ARMA-GJR-GARCH pagal AIC -> p =", p_BTC, "q =", q_BTC
  ,
  ", AIC =", btc_gjr$aic, "\n")
cat("ETH: geriausias ARMA-GJR-GARCH pagal AIC -> p =", p_ETH, "q =", q_ETH
  ,
  ", AIC =", eth_gjr$aic, "\n")
cat("XRP: geriausias ARMA-GJR-GARCH pagal AIC -> p =", p_XRP, "q =", q_XRP
  ,
  ", AIC =", xrp_gjr$aic, "\n")
cat("ADA: geriausias ARMA-GJR-GARCH pagal AIC -> p =", p_ADA, "q =", q_ADA
  ,
  ", AIC =", ada_gjr$aic, "\n")
cat("BNB: geriausias ARMA-GJR-GARCH pagal AIC -> p =", p_BNB, "q =", q_BNB
  ,
  ", AIC =", bnb_gjr$aic, "\n")

#### Sukuriame ARMA-GJR-GARCH modelius visom kriptovaliutom ####
# Pritaikome gjr-garch modeli BTC logaritminem grazom
ugarchspec_BTC <- ugarchspec(
  mean.model=list(armaOrder=c(p_BTC, q_BTC)),
  variance.model=list(model="gjrGARCH",garchOrder = c(1, 1)),
  distribution.model = "norm")
gjr_garch_model_BTC<-ugarchfit(data=BTC_log_grazos,spec=ugarchspec_BTC,
  out.sample = n_test)
show(gjr_garch_model_BTC)

# Pritaikome gjr-garch modeli ETH logaritminem grazom
ugarchspec_ETH <- ugarchspec(
  mean.model=list(armaOrder=c(p_ETH, q_ETH)),
  variance.model=list(model="gjrGARCH",garchOrder = c(1, 1)),
  distribution.model = "norm")
gjr_garch_model_ETH<-ugarchfit(data=ETH_log_grazos,spec=ugarchspec_ETH,
  out.sample = n_test)
show(gjr_garch_model_ETH)

# Pritaikome gjr-garch modeli XRP logaritminem grazom
ugarchspec_XRP <- ugarchspec(
  mean.model=list(armaOrder=c(p_XRP, q_XRP)),
  variance.model=list(model="gjrGARCH",garchOrder = c(1, 1)),
  distribution.model = "norm")
gjr_garch_model_XRP<-ugarchfit(data=XRP_log_grazos,spec=ugarchspec_XRP,
  out.sample = n_test)
show(gjr_garch_model_XRP)

# Pritaikome gjr-garch modeli ADA logaritminem grazom

```

```

ugarchspec_ADA <- ugarchspec(
  mean.model=list(armaOrder=c(p_ADA, q_ADA)),
  variance.model=list(model="gjrgARCH",garchOrder = c(1, 1)),
  distribution.model = "norm")
gjr_garch_model_ADA<-ugarchfit(data=ADA_log_grazos,spec=ugarchspec_ADA,
  out.sample = n_test)
show(gjr_garch_model_ADA)

# Pritaikome gjr-garch modeli BNB logaritminem grazom
ugarchspec_BNB <- ugarchspec(
  mean.model=list(armaOrder=c(p_BNB, q_BNB)),
  variance.model=list(model="gjrgARCH",garchOrder = c(1, 1)),
  distribution.model = "norm")
gjr_garch_model_BNB<-ugarchfit(data=BNB_log_grazos,spec=ugarchspec_BNB,
  out.sample = n_test)
show(gjr_garch_model_BNB)

# Gauname standartizuotas liekanas
stan_liekanos_BTC<-residuals(gjr_garch_model_BTC, standardize=TRUE)
stan_liekanos_ETH<-residuals(gjr_garch_model_ETH, standardize=TRUE)
stan_liekanos_XRP<-residuals(gjr_garch_model_XRP, standardize=TRUE)
stan_liekanos_ADA<-residuals(gjr_garch_model_ADA, standardize=TRUE)
stan_liekanos_BNB<-residuals(gjr_garch_model_BNB, standardize=TRUE)

# Sukuriame standartizuotu liekanu grafika
stan_liekanos_BTC_df <- data.frame(Date = index(stan_liekanos_BTC),
  liek_BTC = stan_liekanos_BTC)
stan_liekanos_ETH_df <- data.frame(Date = index(stan_liekanos_ETH),
  liek_ETH = stan_liekanos_ETH)
stan_liekanos_XRP_df <- data.frame(Date = index(stan_liekanos_XRP),
  liek_XRP = stan_liekanos_XRP)
stan_liekanos_ADA_df <- data.frame(Date = index(stan_liekanos_ADA),
  liek_ADA = stan_liekanos_ADA)
stan_liekanos_BNB_df <- data.frame(Date = index(stan_liekanos_BNB),
  liek_BNB = stan_liekanos_BNB)

liek_plot_BTC <- ggplot(stan_liekanos_BTC_df, aes(x = Date, y = liek_BTC))
  +
  geom_line(color = "#333333", size = 0.6) +
  labs(title = "BTC", x = "", y = "") +
  theme_bw() +
  theme(plot.title = element_text(size = 22, hjust = 0.5))

liek_plot_ETH <- ggplot(stan_liekanos_ETH_df, aes(x = Date, y = liek_ETH))
  +
  geom_line(color = "#333333", size = 0.6) +
  labs(title = "ETH", x = "", y = "") +
  theme_bw() +
  theme(plot.title = element_text(size = 22, hjust = 0.5))

liek_plot_XRP <- ggplot(stan_liekanos_XRP_df, aes(x = Date, y = liek_XRP))
  +
  geom_line(color = "#333333", size = 0.6) +
  labs(title = "XRP", x = "", y = "") +
  theme_bw() +
  theme(plot.title = element_text(size = 22, hjust = 0.5))

liek_plot_ADA <- ggplot(stan_liekanos_ADA_df, aes(x = Date, y = liek_ADA))
  +
  geom_line(color = "#333333", size = 0.6) +

```

```

labs(title = "ADA", x = "", y = "") +
theme_bw() +
theme(plot.title = element_text(size = 22, hjust = 0.5))

liek_plot_BNB <- ggplot(stan_liekanos_BNB_df, aes(x = Date, y = liek_BNB))
+
geom_line(color = "#333333", size = 0.6) +
labs(title = "BNB", x = "", y = "") +
theme_bw() +
theme(plot.title = element_text(size = 22, hjust = 0.5))

(liek_plot_BTC | liek_plot_ETH) / (liek_plot_XRP | liek_plot_ADA) / (
  liek_plot_BNB | plot_spacer())

# Atliekame ARCH test'a standartizuotom liekanom
BTC_stan_liek_arch<-archtest(as.vector(stan_liekanos_BTC))
ETH_stan_liek_arch<-archtest(as.vector(stan_liekanos_ETH))
XRP_stan_liek_arch<-archtest(as.vector(stan_liekanos_XRP))
ADA_stan_liek_arch<-archtest(as.vector(stan_liekanos_ADA))
BNB_stan_liek_arch<-archtest(as.vector(stan_liekanos_BNB))

#### Pritaikome dviejų uodegų apibendrinta Pareto pasiskirstymo misinio
modeli ####
BTC_GNG<-GNG_fit(stan_liekanos_BTC, start = c(break1 = -1.5, break2 =1.5,
mean = 0, sd =1,shape1 = 0.15, shape2= 0.15))
ETH_GNG<-GNG_fit(stan_liekanos_ETH, start = c(break1 = -1.5, break2 =1.5,
mean = 0, sd = 1,shape1 = 0.15, shape2 = 0.15))
XRP_GNG<-GNG_fit(stan_liekanos_XRP, start = c(break1 = -1.5, break2 =1.5,
mean = 0, sd = 1,shape1 = 0.15, shape2 = 0.15))
ADA_GNG<-GNG_fit(stan_liekanos_ADA, start = c(break1 = -1.5, break2 =1.5,
mean = 0, sd = 1,shape1 = 0.15, shape2 = 0.15))
BNB_GNG<-GNG_fit(stan_liekanos_BNB, start = c(break1 = -1.5, break2 =1.5,
mean = 0, sd = 1,shape1 = 0.15, shape2 = 0.15))

# Atspausdiname grafikus
plot(BTC_GNG, which = "cdf"); title(main = "")
plot(ETH_GNG, which = "cdf"); title(main = "")
plot(XRP_GNG, which = "cdf"); title(main = "")
plot(ADA_GNG, which = "cdf"); title(main = "")
plot(BNB_GNG, which = "cdf"); title(main = "")

fits <- list(BTC = BTC_GNG, ETH = ETH_GNG, XRP = XRP_GNG, ADA = ADA_GNG,
  BNB = BNB_GNG)

par(mfrow = c(3, 2))
for (nm in names(fits)) {
  plot(fits[[nm]], which = "cdf", main1 = "")
  title(main = nm, font.main = 2, cex.main = 2.2, line = 0.5)
}
par(mfrow = c(1, 1))

# Transformuojame duomenys ir juos sujungiamo.
coredata_BTC<-coredata(stan_liekanos_BTC)
coredata_ETH<-coredata(stan_liekanos_ETH)
coredata_XRP<-coredata(stan_liekanos_XRP)
coredata_ADA<-coredata(stan_liekanos_ADA)
coredata_BNB<-coredata(stan_liekanos_BNB)
stan_liek<-cbind(coredata_BTC,coredata_ETH,coredata_XRP,coredata_ADA,
  coredata_BNB)

```

```

#### Naudodami gauta pasiskirstyma transformuojame standartizuotas
liekanas ####
transf_stan_liek_BTC<-p(distribution(BTC_GNG),stan_liekanos_BTC)
transf_stan_liek_ETH<-p(distribution(ETH_GNG),stan_liekanos_ETH)
transf_stan_liek_XRP<-p(distribution(XRP_GNG),stan_liekanos_XRP)
transf_stan_liek_ADA<-p(distribution(ADA_GNG),stan_liekanos_ADA)
transf_stan_liek_BNB<-p(distribution(BNB_GNG),stan_liekanos_BNB)
transf_stan_liek<-cbind(transf_stan_liek_BTC,transf_stan_liek_ETH,
  transf_stan_liek_XRP,transf_stan_liek_ADA,transf_stan_liek_BNB)
transf_stan_liek_df <- data.frame(
  BTC = transf_stan_liek_BTC,
  ETH = transf_stan_liek_ETH,
  XRP = transf_stan_liek_XRP,
  ADA = transf_stan_liek_ADA,
  BNB = transf_stan_liek_BNB
)

#Sklaidos diagrama
pairs(
  transf_stan_liek_df,
  main = "Transformuotu (U[0,1]) standartizuotu liekanu sklaidos diagramos
",
  cex.main = 2.2
)

# Ikeliamė sujungtus duomenis i matrica
transf_stan_liek_mat<-as.matrix(transf_stan_liek)

#### Gauname vine struktura ####
colnames(transf_stan_liek_mat) <- c("BTC","ETH","XRP","ADA","BNB")

# Vine kopula
vine_fit <- RVineStructureSelect(
  data = transf_stan_liek_mat,
  familyset = c(1, 2, 3, 4, 5, 6, 7, 9),
  type = 0,
  treecrit = "tau",
  selectioncrit = "AIC",
  indeptest = TRUE,
  progress = TRUE,
  rotations = FALSE
)

vine_fit_rho <- RVineStructureSelect(
  data = transf_stan_liek_mat,
  familyset = c(1, 2, 3, 4, 5, 6, 7, 9),
  type = 0,
  treecrit = "rho",
  selectioncrit = "AIC",
  indeptest = TRUE,
  progress = TRUE,
  rotations = FALSE
)

# Vine kopulos rezultatai
summary(vine_fit)
summary(vine_fit_rho)

#### Sukuriame 10000 simuliaciju su vine kopula struktura ir
stransformuojame jas i standartizuotas liekanas ####

```

```

# Simuliaciju skaicius
n=10000
# Paruosiamė matricas i kurias ikelsime simuliuotus duomenys
vine_BTC_mat = vine_ETH_mat = vine_XRP_mat = vine_ADA_mat = vine_BNB_mat
  <- matrix(rep(0, n), nrow = 1)

# Sukuriame simuliacijas ir konvertuojame duomenis i standartizuotas
  liekanas
set.seed(1234)
vine_U_sim <- RVineSim(n, vine_fit)

for (i in 1:n) {
  vine_BTC_mat[, i] <- q(distribution(BTC_GNG), vine_U_sim[i, 1])
  vine_ETH_mat[, i] <- q(distribution(ETH_GNG), vine_U_sim[i, 2])
  vine_XRP_mat[, i] <- q(distribution(XRP_GNG), vine_U_sim[i, 3])
  vine_ADA_mat[, i] <- q(distribution(ADA_GNG), vine_U_sim[i, 4])
  vine_BNB_mat[, i] <- q(distribution(BNB_GNG), vine_U_sim[i, 5])
}

vine_sim_stan_liek_BTC <- xts(x = vine_BTC_mat, order.by = as.Date
  ("2000-01-01"))
vine_sim_stan_liek_ETH <- xts(x = vine_ETH_mat, order.by = as.Date
  ("2000-01-01"))
vine_sim_stan_liek_XRP <- xts(x = vine_XRP_mat, order.by = as.Date
  ("2000-01-01"))
vine_sim_stan_liek_ADA <- xts(x = vine_ADA_mat, order.by = as.Date
  ("2000-01-01"))
vine_sim_stan_liek_BNB <- xts(x = vine_BNB_mat, order.by = as.Date
  ("2000-01-01"))

#### Transformuojame simuliuotas liekanas i logaritmines grazas ####
# Gauname mu ir sigma is musu arma-gjr-garch modelio
fc_BTC <- ugarchforecast(gjr_garch_model_BTC, n.ahead = 1, n.roll =
  n_test - 1)
fc_ETH <- ugarchforecast(gjr_garch_model_ETH, n.ahead = 1, n.roll =
  n_test - 1)
fc_XRP <- ugarchforecast(gjr_garch_model_XRP, n.ahead = 1, n.roll =
  n_test - 1)
fc_ADA <- ugarchforecast(gjr_garch_model_ADA, n.ahead = 1, n.roll =
  n_test - 1)
fc_BNB <- ugarchforecast(gjr_garch_model_BNB, n.ahead = 1, n.roll =
  n_test - 1)

mu_BTC_vek <- as.numeric(fitted(fc_BTC))
sigma_BTC_vek <- as.numeric(sigma(fc_BTC))

mu_ETH_vek <- as.numeric(fitted(fc_ETH))
sigma_ETH_vek <- as.numeric(sigma(fc_ETH))

mu_XRP_vek <- as.numeric(fitted(fc_XRP))
sigma_XRP_vek <- as.numeric(sigma(fc_XRP))

mu_ADA_vek <- as.numeric(fitted(fc_ADA))
sigma_ADA_vek <- as.numeric(sigma(fc_ADA))

mu_BNB_vek <- as.numeric(fitted(fc_BNB))
sigma_BNB_vek <- as.numeric(sigma(fc_BNB))

test_dates <- index(BTC_log_grazos)[test_idx]

```

```

# Patikriname ar sutampa mu ir sigma vektoriu ilgiai su testavimo dienu
skaiciumi
length(mu_BTC_vek) == length(test_dates)
length(sigma_BTC_vek) == length(test_dates)
all(!is.na(mu_BTC_vek))
all(!is.na(sigma_BTC_vek))

# Apskaiciuojame portfelio grazas
w <- c(1/5, 1/5, 1/5, 1/5, 1/5)
n_test <- length(test_dates)

port_gr_test <- 1/5*BTC_log_grazos[test_idx] +
  1/5*ETH_log_grazos[test_idx] +
  1/5*XRP_log_grazos[test_idx] +
  1/5*ADA_log_grazos[test_idx] +
  1/5*BNB_log_grazos[test_idx]

nuostoliai <- -port_gr_test

#### Suskaiciuojame VaR ####
e_BTC_V <- as.numeric(vine_sim_stan_liek_BTC)
e_ETH_V <- as.numeric(vine_sim_stan_liek_ETH)
e_XRP_V <- as.numeric(vine_sim_stan_liek_XRP)
e_ADA_V <- as.numeric(vine_sim_stan_liek_ADA)
e_BNB_V <- as.numeric(vine_sim_stan_liek_BNB)

VaR99_V_vek <- numeric(n_test)
VaR95_V_vek <- numeric(n_test)
VaR90_V_vek <- numeric(n_test)

for (i in 1:n_test) {
  mu_vek <- c(mu_BTC_vek[i], mu_ETH_vek[i], mu_XRP_vek[i],
    mu_ADA_vek[i], mu_BNB_vek[i])
  sigma_vek <- c(sigma_BTC_vek[i], sigma_ETH_vek[i], sigma_XRP_vek[i],
    sigma_ADA_vek[i], sigma_BNB_vek[i])

  r_BTC_i <- mu_vek[1] + sigma_vek[1] * e_BTC_V
  r_ETH_i <- mu_vek[2] + sigma_vek[2] * e_ETH_V
  r_XRP_i <- mu_vek[3] + sigma_vek[3] * e_XRP_V
  r_ADA_i <- mu_vek[4] + sigma_vek[4] * e_ADA_V
  r_BNB_i <- mu_vek[5] + sigma_vek[5] * e_BNB_V

  port_gr_i <- w[1]*r_BTC_i + w[2]*r_ETH_i + w[3]*r_XRP_i + w[4]*r_ADA_i +
    w[5]*r_BNB_i

  VaR99_V_vek[i] <- -quantile(port_gr_i, 0.01)
  VaR95_V_vek[i] <- -quantile(port_gr_i, 0.05)
  VaR90_V_vek[i] <- -quantile(port_gr_i, 0.10)
}

VaR99_V_ts <- xts(VaR99_V_vek, order.by = test_dates)
VaR95_V_ts <- xts(VaR95_V_vek, order.by = test_dates)
VaR90_V_ts <- xts(VaR90_V_vek, order.by = test_dates)

#Nuokrypiu skaicius
nuokr_V_99 <- nuostoliai > VaR99_V_ts
nuokr_V_95 <- nuostoliai > VaR95_V_ts
nuokr_V_90 <- nuostoliai > VaR90_V_ts

#### Atliekame testavima ####

```

```

# Apskaiciuojame istorini VaR
port_gr_train <- 1/5*BTC_log_grazos_train +
  1/5*ETH_log_grazos_train +
  1/5*XRP_log_grazos_train +
  1/5*ADA_log_grazos_train +
  1/5*BNB_log_grazos_train

n_train <- length(port_gr_train)
n_test  <- length(port_gr_test)

port_gr_all <- c(port_gr_train, port_gr_test)

VaR_istorinis <- function(alpha) {
  vapply(seq_len(n_test), function(i) {
    end <- n_train + i - 1
    -as.numeric(quantile(port_gr_all[1:end], probs = alpha, na.rm = TRUE))
  }, numeric(1))
}

VaR99_H_ts <- xts(VaR_istorinis(0.01), order.by = test_dates)
VaR95_H_ts <- xts(VaR_istorinis(0.05), order.by = test_dates)
VaR90_H_ts <- xts(VaR_istorinis(0.10), order.by = test_dates)

# Nuokrypiu skaicius
nuokr_H_99 <- nuostoliai > VaR99_H_ts
nuokr_H_95 <- nuostoliai > VaR95_H_ts
nuokr_H_90 <- nuostoliai > VaR90_H_ts

#### Kupiec testas ####
kupiec_test <- function(nuokr, alpha, modelis, patikimumas) {
  v <- as.numeric(nuokr)
  N <- length(v)
  x <- sum(v)
  pi_hat <- x / N

  logL0 <- (N - x) * log(1 - alpha) + x * log(alpha)
  logL1 <- (N - x) * log(1 - pi_hat) + x * log(pi_hat)

  LR_uc <- -2 * (logL0 - logL1)
  p_value <- 1 - pchisq(LR_uc, df = 1)
  data.frame(
    Modelis      = modelis,
    Patikimumas = patikimumas,
    N            = N,
    Nuokryptiai = x,
    Alpha        = alpha,
    "I/N"        = pi_hat,
    p_reiksme    = p_value,
    check.names = FALSE
  )
}

kupiec_rezultatas <- rbind(
  # Vine kopulos VaR
  kupiec_test(nuokr_V_99, 0.01, "Vine", 0.99),
  kupiec_test(nuokr_V_95, 0.05, "Vine", 0.95),
  kupiec_test(nuokr_V_90, 0.10, "Vine", 0.90),

  # Istorinis VaR
  kupiec_test(nuokr_H_99, 0.01, "Istorinis", 0.99),

```

```

    kupiec_test(nuokr_H_95, 0.05, "Istorinis", 0.95),
    kupiec_test(nuokr_H_90, 0.10, "Istorinis", 0.90)
)

print(kupiec_rezultatas)

#### VaR ir nuostoliu grafikas ####
plot_VaR_nuostoliai_grafikas <- function(nuostoliai_xts, VaR99_ts,
    VaR95_ts, VaR90_ts) {
  df <- data.frame(
    Date = index(nuostoliai_xts),
    Loss = as.numeric(nuostoliai_xts),
    VaR99 = as.numeric(VaR99_ts),
    VaR95 = as.numeric(VaR95_ts),
    VaR90 = as.numeric(VaR90_ts)
  )

  df$breach99 <- df$Loss > df$VaR99

  ggplot(df, aes(x = Date)) +
    geom_line(aes(y = Loss), color = "black", linewidth = 0.6) +
    geom_line(aes(y = VaR99), color = "red", linewidth = 0.6) +
    geom_line(aes(y = VaR95), color = "blue", linewidth = 0.6) +
    geom_line(aes(y = VaR90), color = "darkgreen", linewidth = 0.6) +
    geom_point(
      data = subset(df, breach99),
      aes(y = Loss),
      color = "red",
      size = 3.5
    ) +
    labs(
      title = "Portfelio nuostoliai ir istorinis VaR",
      x = "",
      y = "Dienos nuostolis / VaR"
    ) +
    theme_bw() +
    theme(
      plot.title = element_text(hjust = 0.5, size = 25),
      axis.text.x = element_text(size = 7),
      axis.text.y = element_text(size = 7)
    )
}

# Vine kopula
p_V <- plot_VaR_nuostoliai_grafikas(
  nuostoliai_xts = nuostoliai,
  VaR99_ts = VaR99_V_ts,
  VaR95_ts = VaR95_V_ts,
  VaR90_ts = VaR90_V_ts
)

# Istorinis
p_H <- plot_VaR_nuostoliai_grafikas(
  nuostoliai_xts = nuostoliai,
  VaR99_ts = VaR99_H_ts,
  VaR95_ts = VaR95_H_ts,
  VaR90_ts = VaR90_H_ts
)
print(p_H)
print(p_V)

```