VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
MATEMATINĖS STATISTIKOS KATEDRA

Viktorija Kovalčis

HOMOGENIŠKUMO KRITERIJAI, NAUDOJANT PRIKLAUSOMAS CENZŪRUOTAS IMTIS

Magistro baigiamasis darbas

Vilnius, *2011*

Darbo vadovas:
Habilituotas matematikos mokslų daktaras, profesorius,
Vilijandas Bagdonavičius

Recenzentas:
Matematikos mokslų daktarė, Rūta Levulienė

Registracijos Nr.: ....
Darbo gynimo data: ....

T U R I N Y S

ĮVADAS

Darbo tikslas – susipažinti ir išnagrinėti homogeniškumo kriterijus, skirtus priklausomoms cenzūruotoms imtims.

Tam buvo suformuluoti tokie uždaviniai:

- Simuliuoti duomenis, naudojantis Kleitono kopula;
- Suformuluoti homogeniškumo kriterijus dvimačiu ir daugiamačiais atvejais;
- Patikrinti homogeniškumo kriterijaus galingumą dvimačiu atveju, naudojant simuliuotus duomenis.

Duomenys bus simuliuojami ir homogeniškumo hipotezė bus tikrinama su SAS programiniu paketu.

Homogeniškumo hipotezė bus tikrinama dviem atvejais, kai duomenys bus simuliuojami pagal tokį patį skirstinius, ir pagal skirtingus skirstinius. Gauti rezultatai parodys, kur artėja gautasis reikšmingumo lygmuo, kai skirstiniai sutampa, ir kokia gaunasi kriterijaus galia, kai skirstiniai nesutampa.

1. Teorinė dalis

Tarkime, kad $T = (T_1,...,T_k)^T$ yra k matis atsitiktinis vektorius, kuris turi daugiamatę pasiskirstymo funkciją F ir išgyvenamumo funkciją S pavidalo

$$F(t_1,...,t_k) = P\{T_1 \leq t_1,...,T_k \leq t_k\}$$

$$S(t_1,...,t_k) = P\{T_1 > t_1,...,T_k > t_k\}$$

ir marginalias pasiskirstymo ir išgyvenamumo funkcijas F_i, S_i, $i = 1,...,k$.

Homogeniškumo hipotezė priklausomoms imtims yra:

$$H_0 : F_1 = ... = F_k,$$

$$\text{čia } F_1(t_1) = P\{T_1 \leq t_1\},$$

$$F_2(t_2) = P\{T_2 \leq t_2\},$$

$$\vdots$$

$$F_k(t_k) = P\{T_k \leq t_k\}.$$

Pavyzdžiai:

a) Gali būti lyginamas identiškų dvynių išgyvenamumas skirtingose vietose.
b) Gali būti lyginami ligos simptomų pasirodymas kairėje ir dešinėje akyse.
c) Gali būti lyginami sistemos komponenčių skirstiniai.

1.1 Logranginio tipo kriterijus homogeniškumo hipotezėms priklausomoms cenzūruotoms imtims tikrinti, kai k=2

1.1.1 Imtis

Tarkime, kad $\mathbf{T}_j = (T_{1j}, T_{2j})^T$ yra atsitiktinio vektoriaus $\mathbf{T}$, $j = 1,...,n$, n nepriklausomų kopijų. Tegu C_{1j} ir C_{2j} yra komponenčių cenzūravimo momentai. Tuomet cenzūruota imtis atrodys taip:

$$(X_{1j}, X_{2j}, \delta_{1j}, \delta_{2j}), \; j = 1,...,n$$

kur $X_{ij} = T_{ij} \wedge C_{ij}$, $\delta_{ij} = 1_{\{T_{ij} \le C_{ij}\}}$, čia T_{ij} - mirties momentai, C_{ij} - cenzūravimo momentai.

Apibrėžkime tokius dydžius:

$$N_{ij}(t) = 1_{\{T_{ij} \le t,\, T_{ij} \le C_{ij}\}} \text{ - stebėtų mirčių skaičius,}$$

$$Y_{ij}(t) = 1_{\{X_{ij} \ge t\}} = 1_{\{T_{ij} \ge t,\, C_{ij} \ge t\}}$$

$Y_{ij}(t) = 1$, kai j-asis objektas yra „rizikos būsenoje“,

$Y_{ij}(t) = 0$, jei objektas miršta arba yra cenzūruojamas prieš momentą t.

$$M_{ij}(t) = N_{ij}(t) - \int_0^t Y_{ij}(u)\, d\Lambda_i(u)$$

$$N_i = \sum_{j=1}^{n} N_{ij}, \; Y_i = \sum_{j=1}^{n} Y_{ij}, \; M_i = \sum_{j=1}^{n} M_{ij}, \; N = N_1 + N_2, \; Y = Y_1 + Y_2.$$

Imtis $(X_{1j}, X_{2j}, \delta_{1j}, \delta_{2j})$, $j = 1,...,n$, yra ekvivalenti imčiai $(N_{1j}(t), N_{2j}(t), Y_{1j}(t), Y_{2j}(t), t \ge 0)$, $j = 1,...,n$, nes žinant vieną galima rasti kitą.

Tikrinsime tokią homogeniškumo hipotezę priklausomoms cenzūruotoms imtims:

$$H_0 : F_1 = F_2.$$

1.1.2 Statistikos asimptotinis skirstinys

Turime statistiką

$$V = \int_0^\infty \frac{K(u)}{Y(u)} [Y_2(u)dN_1(u) - Y_1(u)dN_2(u)]$$

kur K – neneigimas tolydus iš kairės stochastinis procesas.

Įvairiai parenkant K, gaunam įvairias statistikas V:

1. Logranginė statistika, tai svoris

$$K(t) = \frac{1}{\sqrt{n}}.$$

2. Taronės – Vėrės statistika

$$K(t) = \frac{Y^{1/2}(t)}{n}.$$

3. Breslou statistika

$$K(t) = \frac{Y(t)}{n^{3/2}}.$$

4. Prentiso statistika

$$K(t) = \frac{\widetilde{S}(t- Y(t)}{n^{1/2}(Y(t)+ \ )}, \text{ čia } \widetilde{S}(t-) = \prod_{n \le t} \left(1 - \frac{\Delta N(u)}{Y(u)+1} \right).$$

Mūsų tikslas – rasti šios statistikos ribinį dėsnį. Dabar reikia įrodyti atsitiktinio kintamojo V asimptotinį normališkumą, remiantis šiomis prielaidomis.

1 Prielaida. T_j yra absoliučiai tolydus dvimatis atsitiktinis vektorius ir (T_j, C_j), $j = \ ,..., n$, yra nepriklausomi atsitiktiniai vektoriai su nepriklausomomis dvimatėmis komponentėmis, ir jų išgyvenamumo funkcijos

$$S(u,v) = \ T_{1j} > \ , T_{2j} > \ \}$$

$$G_j(u,v) = \ C_{1j} > \ , C_{2j} > \ \}$$

yra tokios, kad $S(\tau \ \tau \ ) > \ $, $G_j(\tau - \tau - \ > \ $, $G_j(\tau \ \tau \ ) = \ $ kiekvienam j.

2 Prielaida. Egzistuoja funkcija $G(u,v)$ tokia, kad

$$\sup_{(u,v)\in\ ,\tau]\times\ ),\tau]} \left| n^{-} \sum_{j=}^{n} G_j(u-\ v-\ -G(u-\ v-\ \right| \to \quad , \text{kai } n \to$$

3 Prielaida. K yra neneigiamas tolydus iš kairės stochastinis procesas su filtracija, generuota iš imties,

$\sup_{u\in\ ,\tau]} \left| n^{1/2} K(t) - k(t) \right| \xrightarrow{P}$, kur k yra tolydi funkcija intervale $[0,\tau]$.

Tegu $y_i(u) = \ ^{(i)}(u-\ S_i(u)$, $y(u) = \ _1(u) + \ _2(u)$, kur $G^{(1)}(u)$ ir $G^{(2)}(u)$ yra $G(u,v)$ marginalai.

Pagal 1 Prielaidą $y_i(u) >$ intervale $[0,\tau]$.

1 Lema.

$$\sup_{u\in\ ,\tau]} \left| n^{-} Y_i(u) - y_i(u) \right| \xrightarrow{P} \quad ,$$

$$\sup_{u\in\ ,\tau]} \left| n^{-} Y(u) - y(u) \right| \xrightarrow{P} \quad , \text{kai } n \to \quad .$$

Įrodymas:

▷

$$\sup_{u\in\ ,\tau]} \left| n^{-} Y_i(u) - y_i(u) \right| = \sup_{u\in\ ,\tau]} \left| n^{-} Y_i(u) - G^{(i)}(u-\ S_i(u) \right| \leq$$

$$\leq \sup_{u\in\ ,\tau]} \left| n^{-} \sum_{j=}^{n} Y_{ij}(u) - \ Y_{ij}(u) \right| + \sup_{u\in\ ,\tau]} \left| S_i(u) \right| \left[n^{-} \sum_{j=}^{n} G_{ij}(u-\ - G_i(u-\ \right] + op(1) \xrightarrow{P}$$

kai $n \to$.

$\sup_{u\in[0,\tau]} \left| S_i(u) \right| \left[n^{-1} \sum_{j=1}^{n} G_{ij}(u-) - G_i(u-) \right] \xrightarrow{P}$ pagal 2 prielaidą , o

$\sup_{u\in[0,\tau]} \left| n^{-1} \sum_{j=1}^{n} Y_{ij}(u) - \ Y_{ij}(u) \right| \xrightarrow{P}$ pagal Borelio – Kantelio lemą, taigi gaunam, kad

$\sup_{u\in\ ,\tau]} \left| n^{-} Y_i(u) - \ _i(u) \right| \xrightarrow{P}$.

Akivaizdu, kad ir antroji lemos dalis teisinga, t.y. $\sup_{u\in[0,\tau]}\left|n^{-1}Y(u)-y(u)\right| \xrightarrow{P} 0$, kai $n\to\infty$. ◁

Pagal hipotezę H_0 teisinga ir lygybė $\Lambda_1 = \Lambda_2 =: \Lambda$. Taigi, skaidinys $dN_{ij} = Y_{ij}d\Lambda + dM_{ij}$ parodo, kad statistika V gali būti perrašyta tokia forma:

$$V = \frac{1}{\sqrt{n}}\left(\int_0^\tau \hat{Q}_1(u)dM_1(u) - \int_0^\tau \hat{Q}_2(u)dM_2(u)\right)$$

kur $\hat{Q}_1 = \frac{\sqrt{n}KY_2}{Y}$, $\hat{Q}_2 = \frac{\sqrt{n}KY_1}{Y}$.

Įsiveskim panašų atsitiktinį kintamąjį $V^* = \frac{1}{\sqrt{n}}\left[\int_0^\tau Q_1(u)dM_1(u) - \int_0^\tau Q_2(u)dM_2(u)\right]$, kur $Q_1 = \frac{ky_2}{y}$, $Q_2 = \frac{ky_1}{y}$. Tuomet pagal visas prielaidas ir 1 lemą

$$\sup_{u\le\tau}\left|\hat{Q}_i(u)-Q_i(u)\right| \xrightarrow{P} 0, \text{ kai } n\to\infty.$$

2 Lema.

$$V-V^* \xrightarrow{P} 0, \text{ kai } n\to\infty$$

Įrodymas:

▷

$$V-V^* = \frac{1}{\sqrt{n}}\int_0^\tau [\hat{Q}_1(u)-Q_1(u)]dM_1(u) - \frac{1}{\sqrt{n}}\int_0^\tau [\hat{Q}_2(u)-Q_2(u)]dM_2(u)$$

Atsitiktinis procesas $M_l^*(t) = \frac{1}{\sqrt{n}}\int_0^t [\hat{Q}_l(u)-Q_l(u)]dM_l(u)$ yra martingalas, kurio numatoma variacija yra

$$\begin{aligned}<M_l^*>(t) &= \frac{1}{n}\int_0^t [\hat{Q}_l(u)-Q_l(u)]^2 Y_l(u)d\Lambda(u) \le \\ &\le \sup_{u\le\tau}\left|\hat{Q}_l(u)-Q_l(u)\right|\Lambda(\tau)\end{aligned}$$

Žinome, kad $\sup_{u\le\tau}\left|\hat{Q}_i(u)-Q_i(u)\right| \xrightarrow{P} 0$, tai tada ir $<M_l^*>(t) \xrightarrow{P} 0$, kai $n\to\infty$.

Tuomet pagal Lenglaro nelygybę (žiūrėti Priede A) gauname, kad $V-V^* \xrightarrow{P} 0$, kai $n\to\infty$. ◁

Taigi, įrodėme asimptotinį V normališkumą. Kadangi $V - V^* \xrightarrow{P}$, tai ribiniai dėsniai V ir V* sutampa, ir todėl galime surasti tik V* ribinį dėsnį.

1 Teorema. Jei teisinga hipotezė H_0 ir patenkintos 1-3 prielaidos, tai

$$V \xrightarrow{d} \mathrm{N}\ , \sigma^{2}), \text{ kai } n \to \quad ,$$

kur

$$\sigma^{2} = \quad - \quad C(1,2),$$

$$C = \int_0^\tau \ ^2(u) \frac{y_1(u) y_2(u)}{y(u)} d\Lambda \quad),$$

$$C(1,2) = \int_0^\tau \int_0^\tau Q_1(u) Q_2(v) G(u,v) \times$$

$$\times \left[(u,v) d\Lambda \quad) d\Lambda \quad) + \quad (du,dv) + \quad (du,v) d\Lambda \quad) + \quad (u,dv) d\Lambda \quad) \right].$$

Įrodymas:

⊳ Iš 2 lemos žinome, kad V yra asimptotiškai ekvivalentus V*, tai pabandykime rasti ribinį dėsnį dydžio, kurį užrašykime taip:

$V^* = \frac{1}{\sqrt{n}} \sum_{j=} \quad _j - \quad _j)$, kur $(\xi_{\cdot j}, \xi_{\cdot j})$ - nepriklausomi, bet ne vienodai pasiskirstę vektoriai su komponentėmis

$$\xi_{ij} = \int_0^\tau Q_i(u) dM_{ij}(u) = \int_0^\tau Q_i(u) \left[N_{ij}(u) - Y_{ij}(u) d\Lambda(u) \right] =$$

$$= \ _i(X_{ij}) \delta_{ij} - \ _i(X_{ij}), \text{ čia } I_i(t) = \int_0^t \ _i(u) d\Lambda \quad).$$

Pastebėkim, kad $\mathrm{E}\xi_{ij} = \ $. Raskime kovariacijas

$$C_j(i,i') = \ \mathrm{ov}(\xi_{ij}, \xi \ ) = \ \xi_{ij}, \xi \ )$$

ir ribas

$$C(i,i') = \lim_{n \to} n^{-} \sum_{j=} \quad (i,i').$$

Žinome, kad

$$C_j(i,i) = Var(\xi_{ij}) = Var\int_0^\tau Q_i(u)dM_{ij}(u) = \mathrm{E} < \int_0^\tau Q_i(u)dM_{ij}(u) > =$$

$$= \mathrm{E}\left(\int_0^\tau Q^2{}_i(u)Y_{ij}(u)d\Lambda(u)\right) = \int_0^\tau Q^2{}_i(u)\mathrm{E}Y_{ij}(u)d\Lambda(u)$$

Ir

$$C(i,i) = \lim_{n\to\infty} n^{-1}\sum_{j=1}^n C_j(i,i) = \int_0^\tau Q_i^2(u)y_i(u)d\Lambda(u).$$

Taip pat žinome, kad $C(1,1) + C(2,2) = C$, kur C yra aprašytas teoremos formuluotėje.

$$C_j(1,2) = \mathrm{E}\xi_{1j}\xi_{2j} = \mathrm{E}\delta_{1j}\delta_{2j}Q_1(T_{1j})Q_2(T_{2j}) - \mathrm{E}\delta_{1j}Q_1(T_{1j})I_2(X_{2j}) -$$

$$- \mathrm{E}\delta_{2j}I_1(X_{1j})Q_2(T_{2j}) + \mathrm{E}I_1(X_{1j})I_2(X_{2j}) = \sum_{i=1}^4 \mathrm{E}_i .$$

Pažymėkime

$$H_2(x,y) = P\{X_{1j} \le x, X_{2j} \le y\},$$

$$\overline{H}_2(x,y) = P\{X_{1j} > x, X_{2j} > y\} = S(x,y)G_j(x,y),$$

$$H_3(u,v,x) = P\{T_{1j} \le u, C_{1j} \le v, X_{2j} \le x\},$$

$$\overline{H}_3(u,v,x) = P\{T_{1j} > u, C_{1j} > v, X_{2j} > x\} = S(u,x)G_j(v,x),$$

$$H_4(u,v,x,y) = P\{T_{1j} \le u, T_{2j} \le v, C_{1j} \le x, C_{2j} \le y\},$$

$$\overline{H}_4(u,v,x,y) = P\{T_{1j} > u, T_{2j} > v, C_{1j} > x, C_{2j} > y\} = S(u,v)G_j(x,y).$$

Turime, kad

$$\mathrm{E}_1 = \mathrm{E}\mathrm{I}_{\{T_{1j}\le C_{1j}, T_{2j}\le C_{2j}\}}Q_1(T_{1j})Q_2(T_{2j}) = \int_0^\tau\int_0^\tau Q_1(u)Q_2(v)\int_u^\tau\int_v^\tau H_4(du,dv,dx,dy) =$$

$$= \int_0^\tau\int_0^\tau Q_1(u)Q_2(v)\overline{H}_4(du,dv,u,v),$$

$$\mathrm{E}_2 = \mathrm{E}\mathrm{I}_{\{T_{1j}\le C_{1j}\}}Q_1(T_{1j})I_2(X_{2j}) = \int_0^\tau Q_1(u)\int_u^\tau\int_0^\tau\int_0^x Q_2(y)d\Lambda(y)H_3(du,dv,dx) =$$

$$= \int_0^\tau Q_1(u)\int_0^\tau Q_2(y)d\Lambda(y)\int_u^\tau\int_y^\tau H_3(du,dv,dx) = -\int_0^\tau\int_0^\tau Q_1(u)Q_2(y)\overline{H}_3(du,u,y)d\Lambda(y)$$

E_3 randame analogiškai.

$$\mathrm{E}_4=\int_0^\tau\int_0^\tau\int_0^u Q_1(x)d\Lambda(x)\int_0^v Q_2(y)d\Lambda(y)H_2(du,dv)=\int_0^\tau\int_0^\tau Q_1(x)d\Lambda(x)Q_2(y)d\Lambda(y)\int_x^\tau\int_y^\tau H_2(du,dv)=$$

$$=\int_0^\tau\int_0^\tau Q_1(x)Q_2(y)\overline{H}_2(x,y)d\Lambda(x)d\Lambda(y).$$

Taigi,

$$C_j(1,2)=\int_0^\tau\int_0^\tau Q_1(u)Q_2(v)G_j(u,v)\times$$

$$\times\left[S(du,dv)+S(du,v)d\Lambda(v)+S(u,dv)d\Lambda(u)+S(u,v)d\Lambda(u)d\Lambda(v)\right]$$

ir V dispersija konverguoja į dispersiją $\sigma^2 = C - 2C(1,2)$.

Kad įrodytume $\frac{1}{\sqrt{n}}\left(\sum_{j=1}^n \xi_{1j},\sum_{j=1}^n \xi_{2j}\right)$ asimptotinį normališkumą, reikia patikrinti

Liapunovo sąlygą

$$\sum_{j=1}^n \mathrm{E}\left|\xi_{ij}\right|^{2+\delta} \Big/ \left(\sum_{j=1}^n Var(\xi_{ij})\right)^{1+\delta/2} \to 0 .$$

Todėl, naudojant konvergavimą $n^{-1}\sum_{j=1}^n C_j(i,i) \to C(i,i)$ kiekvienam $\delta > 0$, gauname, kad

$$\sum_{j=1}^n \mathrm{E}\left|\xi_{ij}\right|^{2+\delta} \Big/ \left(\sum_{j=1}^n Var(\xi_{ij})\right)^{1+\delta/2} \le \frac{M^{2+\delta}}{n^{\delta/2}\left[n^{-1}\sum_{j=1}^n C_j(i,i)\right]^{1+\delta/2}} \to 0, \text{ kai } n \to \infty .$$

Taigi, Liapunovo sąlyga patenkinta. $\triangleleft$

Tam, kad įvertintume statistikos V dispersiją σ^2, yra naudinga įvertinti charakteristikas C ir $C(1,2)$.

Sukauptojo intensyvumo Λ įvertinys yra:

$$\hat{\Lambda}(t)=\int_0^\tau \frac{dN(u)}{Y(u)} .$$

Taigi natūralu, jog C įvertinys yra

$$\hat{C}=\int_0^\tau K^2(u)\frac{Y_1(u)Y_2(u)}{Y^2(u)}dN(u) .$$

Vidurkiai

$$\mathrm{E}N_{1j}(x)N_{2j}(y)=\int_0^x\int_0^y G_j(u,v)S(du,dv),$$

$$\mathrm{E}Y_{1j}(u)Y_{2j}(v)=G_j(u,v)S(u,v),$$

$$\mathrm{E}\int_0^v Y_{1j}(u)dN_{2j}(v) = \mathrm{E}\,\mathrm{I}_{\{X_{1j}\geq u\}}\mathrm{I}_{\{T_{2j}\leq v,\delta_{2j}=1\}} = -\int_0^v \pi_j(u,v)S(u,dv),$$

$$\mathrm{E}\int_0^x Y_{2j}(v)dN_{1j}(u) = \mathrm{E}\,\mathrm{I}_{\{X_{2j}\geq v\}}\mathrm{I}_{\{T_{1j}\leq x,\delta_{1j}=1\}} = -\int_0^x \pi_j(u,v)S(du,v).$$

$C(1,2)$ įvertinys yra:

$$\hat{C}(1,2) = \frac{1}{n}\sum_{j=1}^{n}\int_0^\tau\int_0^\tau \hat{Q}_1(u)\hat{Q}_2(v)\left[dN_{1j}(u)dN_{2j}(v) - Y_{2j}(v)dN_{1j}(u)d\hat{\Lambda}(v) -\right.$$

$$\left. - Y_{1j}(u)d\hat{\Lambda}(u)dN_{2j}(v) + Y_{1j}(u)Y_{2j}(v)d\hat{\Lambda}(u)d\hat{\Lambda}(v)\right],\ \text{kur}\ \hat{Q}_1 = \frac{\sqrt{n}KY_2}{Y},\ \hat{Q}_2 = \frac{\sqrt{n}KY_1}{Y}.$$

Taigi, dispersijos σ^2 įvertinys yra:

$$\hat{\sigma}^2 = \hat{C} - 2\hat{C}(1,2) = \int_0^\tau K^2(u)\frac{Y_1(u)Y_2(u)}{Y^2(u)}dN(u) - 2\left[\frac{1}{n}\sum_{j=1}^{n}\int_0^\tau\int_0^\tau \hat{Q}_1(u)\hat{Q}_2(v)\left(dN_{1j}(u)dN_{2j}(v) -\right.\right.$$

$$\left.\left. - Y_{2j}(v)dN_{1j}(u)d\hat{\Lambda}(v) - Y_{1j}(u)d\hat{\Lambda}(u)dN_{2j}(v) + Y_{1j}(u)Y_{2j}(v)d\hat{\Lambda}(u)d\hat{\Lambda}(v)\right)\right].$$

2 Teorema. Jei teisinga hipotezė H_0 ir patenkintos 1-3 prielaidos, tai

$$\hat{\sigma}^2 \xrightarrow{P} \sigma^2.$$

Įrodymas:

$\triangleright$

$$\hat{C} - C = \int_0^\tau K^2(u)\frac{Y_1(u)Y_2(u)}{Y^2(u)}dM(u) +$$

$$+\int_0^\tau\left(K^2(u)\frac{Y_1(u)Y_2(u)}{Y(u)} - k^2(u)\frac{y_1(u)y_2(u)}{y(u)}\right)d\Lambda(u) \xrightarrow{P} 0.$$

$\int_0^\tau K^2(u)\frac{Y_1(u)Y_2(u)}{Y^2(u)}dM(u) \xrightarrow{P} 0$ išplaukia iš to, jog $\int_0^\tau K^4(u)\frac{Y_1^2(u)Y_2^2(u)}{Y^3(u)}d\Lambda(u) \xrightarrow{P} 0$

ir Lenglaro nelygybės (žiūrėti Priede A).

$\int_0^\tau\left(K^2(u)\frac{Y_1(u)Y_2(u)}{Y(u)} - k^2(u)\frac{y_1(u)y_2(u)}{y(u)}\right)d\Lambda(u) \xrightarrow{P} 0$ pagal 3 Prielaidą ir dėl to, jog $y(u) > 0$

intervale $[0,\tau]$.

Apibrėškime $C^*(1,2) = \frac{1}{n}\sum_{j=1}^{n}\int_0^\tau\int_0^\tau Q_1(u)Q_2(v)\left[dN_{1j}(u)dN_{2j}(v) - Y_{2j}(v)dN_{1j}(u)d\hat{\Lambda}(v) -\right.$

$$- Y_{1j}(u)d\Lambda(u)dN_{2j}(v) + Y_{1j}(u)Y_{2j}(v)d\Lambda(u)d\Lambda(v)\big].$$

Tuomet $\hat{C}(1,2) - C^{*}(1,2) \xrightarrow{P} 0$, kai $n \to \infty$. Kad įrodytume, jog $C^{*}(1,2) \longrightarrow C(1,2)$, įsiveskime dar tokį dydį

$$C^{0}(1,2) = n^{-1}\sum_{j=1}^{n}\zeta_{j}\text{, kur}$$

$$\zeta_{j} = \int_{0}^{\tau}\int_{0}^{\tau} Q_{1}(u)Q_{2}(v)\big[dN_{1j}(u)N_{2j}(v) - Y_{2j}(v)dN_{1j}(u)d\Lambda(v) - Y_{1j}(u)d\Lambda(u)dN_{2j}(v) + Y_{1j}(u)Y_{2j}(v)d\Lambda(u)d\Lambda(v)\big]$$

$$\mathrm{E}\zeta_{j} = C(1,2)\ \forall j,$$

$$Var(\zeta_{j}) \le \mathrm{E}\zeta_{j}^{2} \le m^{2}\text{, čia } m \text{ – teigiama konstanta.}$$

Iš Čebyševo nelygybės (žiūrėti Priede A) bet kokiam $\varepsilon > 0$ yra teisinga, jog

$$\mathrm{P}\left(\left|C^{0}(1,2) - C(1,2)\right| > \varepsilon\right) \le \frac{Var\left(C^{0}(1,2)\right)}{\varepsilon^{2}} \le \frac{m^{2}}{n\varepsilon^{2}} \to 0.$$

Taigi gauname, jog

$$C^{0}(1,2) \longrightarrow C(1,2).$$

Tarkime, kad Z yra numatomas atsitiktinis procesas su $\sup_{u\in[0,\tau]}\left|Z(u)\right| < \infty$. Tada martingalo

$$M(t) = \int_{0}^{t} Z(u)d\left(\hat{\Lambda}(u) - \Lambda(u)\right)$$

numatoma variacija yra

$$< M > (t) = \int_{0}^{t}\frac{Z^{2}(u)}{Y(u)}d\Lambda(u),$$

$$< M > (t) \xrightarrow{P} 0\text{, kai } n \to \infty,$$

ir iš Išvados po Lenglaro nelygybe (žiūrėti Priede A) žinome, kad $\sup_{t\le\tau}\left|M(t)\right| \longrightarrow 0$, kai $n \to \infty$.

Todėl, pasinaudojus nelygybėmis

$$0 \le Y_{ij}(u), Q_{i}(u), Y_{i}(u)/n \le 1$$

gauname, jog

$$\left|C^{*}(1,2) - C^{0}(1,2)\right| =$$

$$= \left|\frac{1}{n}\sum_{j=1}^{n}\left(\int_{0}^{\tau}Q_{1}(u)Y_{1j}(u)d\hat{\Lambda}(u)\int_{0}^{\tau}Q_{2}(v)Y_{2j}(v)d\hat{\Lambda}(v) - \int_{0}^{\tau}Q_{1}(u)Y_{1j}(u)d\Lambda(u)\int_{0}^{\tau}Q_{2}(v)Y_{2j}(v)d\Lambda(v)\right) -\right.$$

$$-\sum_{j=1}^{n}\int_0^\tau\int_0^\tau Q_1(u)Q_2(v)Y_{2j}(v)dN_{1j}(u)d\left(\hat{\Lambda}(v)-\Lambda(v)\right)-\frac{1}{n}\sum_{j=1}^{n}\int_0^\tau\int_0^\tau Q_1(u)Q_2(v)Y_{1j}(u)d\left(\hat{\Lambda}(u)-\Lambda(u)\right)dN_{2j}(v)\Bigg|\le$$

$$\le\left|\frac{1}{n}\int_0^\tau Q_1(u)Y_1(u)d\left(\hat{\Lambda}(u)-\Lambda(u)\right)\hat{\Lambda}(\tau)\right|+\left|\frac{1}{n}\int_0^\tau Q_2(v)Y_2(v)d\left(\hat{\Lambda}(v)-\Lambda(v)\right)\hat{\Lambda}(\tau)\right|$$

$$\left|\frac{1}{n}\int_0^\tau Q_2(v)Y_2(v)d\left(\hat{\Lambda}(v)-\Lambda(v)\right)\right|+\left|\frac{1}{n}\int_0^\tau Q_1(u)Y_1(u)d\left(\hat{\Lambda}(u)-\Lambda(u)\right)\right|\xrightarrow{P}0,\ \text{kai}\ n\to\infty.\ \triangleleft$$

1.1.3 Kriterijus

Kriterijaus statistika yra $Z_n = \frac{V}{\hat{\sigma}}$. Įrodę 1 ir 2 teoremas, žinome, kad jeigu hipotezė H_0 teisinga, tai $Z_n \longrightarrow \sim N(0,1)$.

Hipotezėms apie priklausomų cenzūruotų imčių homogeniškumą tikrinti naudosime svertinį logranginį kriterijų. Hipotezė H_0 yra atmetama su asimptotiniu reikšmingumo lygmeniu α, jei $|Z_n| > z_{\alpha/2}$.

Statistiką V galime perrašyti forma, patogesne skaičiavimams. Ji atrodys taip:

$$V = \sum_{j=} \frac{K(X_{1j})Y_2(X_{1j})}{Y(X_{1j})}\delta_{j} - \sum_{j=} \frac{K(X_{2j})Y_1(X_{2j})}{Y(X_{2j})}\delta_{j},$$

$$\hat{\sigma}^2 = \hat{C} - \hat{C}(1,2),$$

$$\text{čia } \hat{C} = \sum_{s=} \sum_{j=} K^2(X_{sj})\delta_{j} \frac{Y_1(X_{sj})Y_2(X_{sj})}{Y^2(X_{sj})},$$

$$\hat{C}(1,2) = \frac{1}{n}\sum_{j=1}^{n} \left(R_{1j}R_{2j} - R_{1j}R_{4j} - R_{2j}R_{3j} + R_{3j}R_{4j} \right), \text{ kur}$$

$$R_{1j} = \delta_{j}\hat{Q}_1(X_{1j}), R_{2j} = \delta_{j}\hat{Q}_2(X_{2j}),$$

$$R_{3j} = \sum_{s=} \sum_{l=} \delta \hat{Q}_1(X_{sl}) \frac{Y_{1j}(X_{sl})}{Y(X_{sl})},$$

$$R_{4j} = \sum_{s=} \sum_{l=} \delta \hat{Q}_2(X_{sl}) \frac{Y_{2j}(X_{sl})}{Y(X_{sl})}.$$

Skaičiavimams su SAS paketu imsime $K(X_{1j}) = K(X_{2j}) = \frac{1}{\sqrt{n}}$.

1.2 Logranginio tipo kriterijus homogeniškumo hipotezėms priklausomoms cenzūruotoms imtims tikrinti, kai k > 2

Homogeniškumo hipotezė priklausomoms imtims,kai k > 2, yra

$$H_0 : F_1(t) = \ldots = F_k(t) \text{ visiems } t \geq 0.$$

Tegu $V = (V_1,...,V_{k-1})^T$ - atsitiktinis vektorius, kur

$$V_i = \int_0^{\tau} \frac{K(u)}{Y(u)} \left[Y(u)dN_i(u) - Y_i(u)dN(u) \right].$$

Kriterijaus statistikos asimptotinis skirstinys. Pasinaudojus išdėstymu

$dN_i = Y_i d\Lambda + dM_i$, gauname, jog

$$V_i = \frac{1}{\sqrt{n}} \sum_{l=1}^{k} \int_0^{\tau} \hat{Q}_{il}(u) dM_l(u), \; \hat{Q}_{il}(u) = \sqrt{n} K(u) \left(\varepsilon_{il} - \frac{Y_i(u)}{Y(u)} \right),$$

$$\text{kur } \varepsilon_{il} = 1_{\{i=l\}}.$$

Analogiškai kaip ir dviejų imčių atveju, turime, kad $V \xrightarrow{d} U \sim N_{k-1}(0,\Sigma)$, kur

$$\Sigma = [\sigma_{ij}], \; \sigma_{ij} = c_{ij} + \sum_{l \neq l'} c_{ij}(l,l'),$$

$$c_{ij} = \int_0^{\tau} k^2(u) y_i(u) \left(\varepsilon_{ij} - \frac{y_j(u)}{y(u)} \right) d\Lambda(u),$$

$$c_{ij}(l,l') = \int_0^{\tau}\int_0^{\tau} Q_{il}(u) Q_{jl'}(v) G_{ll'}(u,v) \times \left[S_{ll'}(u,v)d\Lambda(u)d\Lambda(v) + S_{ll'}(du,dv) + \right.$$

$$\left. + S_{ll'}(du,v)d\Lambda(v) + S_{ll'}(u,dv)d\Lambda(u) \right],$$

$$Q_{il}(u) = k(u) \left(\varepsilon_{il} - \frac{y_i(u)}{y(u)} \right).$$

Matricos Σ įvertinys yra

$$\hat{\Sigma} = [\hat{\sigma}_{ij}], \; \hat{\sigma}_{ij} = \hat{c}_{ij} + \sum_{l \neq l'} \hat{c}_{ij}(l,l'),$$

$$\hat{c}_{ij} = \int_0^{\tau} K^2(u) \frac{Y_i(u)}{Y(u)} \left(\varepsilon_{ij} - \frac{Y_j(u)}{Y(u)} \right) dN(u),$$

$$\hat{c}_{ij}(l,l') = \frac{1}{n}\sum_{r=1}^{n}\int_0^\tau\int_0^\tau \hat{Q}_{il}(u)\hat{Q}_{jl'}(v)\{dN_{lr}(u)dN_{l'r}(v) - Y_{l'r}(v)dN_{lr}(u)d\hat{\Lambda}(v) -$$

$$- Y_{lr}(u)d\hat{\Lambda}(u)dN_{l'r}(v) + Y_{lr}(u)Y_{l'r}(v)d\hat{\Lambda}(u)d\hat{\Lambda}(v)\},$$

čia $\hat{\Lambda}(t) = \int_0^t \frac{dN(u)}{Y(u)}$.

Tegu $Y_n^2 = V^T\hat{\Sigma}^{-1}V$. Hipotezė H_0 yra atmetama su asimptotiniu reikšmingumo lygmeniu α, jei $Y_n^2 > \chi^2_{1-\alpha}(k-1)$.

Užrašykime statistiką skaičiavimams patogesne forma. Statistikos $\boldsymbol{V}$ komponentės yra

$$V_i = \sum_{j=1}^{n}\delta_{ij}K(X_{ij}) - \sum_{l=1}^{k}\sum_{j=1}^{n}\delta_{lj}K(X_{lj})\frac{Y_i(X_{lj})}{Y(X_{lj})},$$

ir matricos $\hat{\Sigma} = [\hat{\sigma}_{ij}]$ elementai yra $\hat{\sigma}_{ij} = \hat{c}_{ij} + \sum_{l\neq l'}\hat{c}_{ij}(l,l')$. Čia

$$\hat{c}_{ij} = \sum_{s=1}^{k}\sum_{q=1}^{n}\delta_{sq}K^2(X_{sq})\frac{Y_i(X_{sq})}{Y(X_{sq})}\left(\varepsilon_{ij} - \frac{Y_j(X_{sq})}{Y(X_{sq})}\right),$$

$$\hat{c}_{ij}(l,l') = \frac{1}{n}\sum_{r=1}^{n}\left[R_{1ilr}R_{1jl'r} - R_{1ilr}R_{2jl'r} - R_{1jl'r}R_{2ilr} + R_{2ilr}R_{2jl'r}\right],$$

$$R_{1ilr} = \delta_{lr}\hat{Q}_{il}(X_{lr}),\quad R_{2ilr} = \sum_{s=1}^{k}\sum_{q=1}^{n}\delta_{sq}\hat{Q}_{il}(X_{sq})\frac{Y_{lr}(X_{sq})}{Y(X_{sq})}.$$

1.3 Homogeniškumo kriterijai, kai alternatyva yra marginaliųjų išgyvenamumo funkcijų pasiskirstymas

Sukonstruokime kriterijų homogeniškumo hipotezei $H_0 : F_1 = ... = F_k$ tikrinti.

Tegu naudojami duomenys: $(X_{1j}, X_{2j}, \delta_{1j}, \delta_{2j})$, $j = 1,...,n$, ir patenkintos 1 ir 2 Prielaidos.

Turime du kriterijus ir kiekvienas iš jų pagrįstas statistika $\boldsymbol{V} = (V_1, V_2)^T$, kur

$$V_l = \int_0^{\tau} \frac{K_l(t)}{Y(t)} \left\{ Y_2(t)dN_1(t) - Y_1(t)dN_1(t) \right\}, \; l = 1,2$$

ir

$$K_1(t) = \frac{1}{\sqrt{n}} e^{-\Lambda(t)}, \; K_2(t) = \frac{1}{\sqrt{n}} \left(e^{-\Lambda(t)} - 1 \right) \text{ - pirmajam kriterijui,}$$

$$K_1(t) = \frac{1}{\sqrt{n}}, \; K_2(t) = -\frac{1}{\sqrt{n}} \ln\left(1 + \Lambda(t)\right) \text{ - antrajam kriterijui.}$$

Modifikuotų informantinių funkcijų asimptotinis pasiskirstymas.

$$\hat{Q}_{l1} = \frac{\sqrt{n} K_l Y_2}{Y}, \; \hat{Q}_{l2} = \frac{\sqrt{n} K_l Y_1}{Y}, \; Q_{l1} = \frac{k_l y_2}{y}, \; Q_{l2} = \frac{k_l y_1}{y}, \; l = 1,2.$$

Pasinaudojus išdėstymu $dN_i = Y_i d\Lambda + dM_i$, gauname

$$V_l = \frac{1}{\sqrt{n}} \int_0^{\tau} \left\{ \hat{Q}_{l1}(u)dM_1(u) - \hat{Q}_{l2}(u)dM_2(u) \right\}.$$

3 Teorema. $\boldsymbol{V} \xrightarrow{d} \eta = (\eta_1, \eta_2)^T \sim N_1(0, \Sigma)$, kai $n \to \infty$. Čia $\Sigma = \|\sigma_{ls}\|_{2\times 2}$,

$$\sigma_{ls} = C_{l,s} - C_{l,s}(1,2) - C_{s,l}(1,2),$$

$$C_{l,s} = \int_0^{\tau} k_l(u) k_s(u) \frac{y_1(u) y_2(u)}{y(u)} d\Lambda(u),$$

$$C_{l,s}(1,2) = \int_0^{\tau}\int_0^{\tau} Q_{l1}(u) Q_{s2}(v) G(u,v) \times \left\{ S(u,v) d\Lambda(u) d\Lambda(v) + S(du,dv) + \right.$$

$$\left. + S(du,v) d\Lambda(v) + S(u,dv) d\Lambda(u) \right\}.$$

Įrodymas analogiškas kaip ir 1 Teoremos.

Kovariacijų matricos Σ įvertinys yra $\hat{\Sigma} = \|\hat{\sigma}_{ls}\|$, kur

$$\hat{\sigma}_{ls} = \hat{C}_{l,s} - \hat{C}_{l,s}(1,2) - \hat{C}_{s,l}(1,2),$$

$$\hat{C}_{l,s} = \int_0^{\tau} K_l(u)K_s(u)\frac{Y_1(u)Y_2(u)}{Y^2(u)}dN(u),$$

$$\hat{C}_{l,s}(1,2) = \frac{1}{n}\sum_{j=1}^{n}\int_0^{\tau}\int_0^{\tau}\hat{Q}_{l1}(u)\hat{Q}_{s2}(v)\Big[dN_{1j}(u)dN_{2j}(v) - Y_{2j}(v)dN_{1j}(u)d\hat{\Lambda}(v) -$$

$$- Y_{1j}(u)d\Lambda(u)dN_{2j}(v) + Y_{1j}(u)Y_{2j}(v)d\Lambda(u)d\Lambda(v)\Big].$$

4 Teorema. Jei teisinga hipotezė H_0 ir patenkintos 1 ir 2 Prielaidos, tai

$$\hat{\Sigma} \xrightarrow{P} \Sigma .$$

Įrodymas analogiškas kaip ir 2 Teoremos.

Tegu $X_n^2 = V^T\hat{\Sigma}^{-1}V$. Tuomet $X_n^2 \xrightarrow{d} X^2 \sim \chi^2_2$.

Modifikuotas informantinis kriterijus priklausomų imčių homogeniškumui tikrinti: hipotezė H_0 yra atmetama su asimptotiniu reikšmingumo lygmeniu α, jeigu $X_n^2 > \chi^2_\alpha(2)$.

Užrašykime kriterijaus statistiką forma patogesne skaičiavimams.

Statistikos $\boldsymbol{V} = (V_1, V_2)^T$ komponentės yra:

$$V_l = \sum_{j=1}^{n}\frac{K_l(X_{1j})Y_2(X_{1j})}{Y(X_{1j})}\delta_{1j} - \sum_{j=1}^{n}\frac{K_l(X_{2j})Y_1(X_{2j})}{Y(X_{2j})}\delta_{2j} .$$

O matricos $\hat{\Sigma} = \|\hat{\sigma}_{ls}\|$ elementai yra

$$\hat{\sigma}_{ls} = \hat{c}_{ls} - \hat{c}_{ls}(1,2) - \hat{c}_{sl}(1,2),$$

$$\hat{c}_{ls} = \sum_{q=1}^{k}\sum_{v=1}^{n}\delta_{qv}K_l(X_{qv})K_s(X_{qv})\frac{Y_1(X_{qv})Y_2(X_{qv})}{Y^2(X_{qv})},$$

$$\hat{c}_{ls}(1,2) = \frac{1}{n}\sum_{r=1}^{n}\left(R_{1l1r}R_{1s2r} - R_{1l1r}R_{2s2r} - R_{1s2r}r_{2l1r} + R_{2l1r}R_{2s2r}\right),$$

$$R_{1lir} = \delta_{ir}\hat{Q}_{li}(X_{ir}),\quad R_{2lir} = \sum_{q=1}^{k}\sum_{v=1}^{n}\delta_{qv}\hat{Q}_{li}(X_{qv})\frac{Y_{ir}(X_{qv})}{Y(X_{qv})}.$$

1.4 Kleitono kopula

Kopulos yra naudingos daugiamačių skirstinių konstravimui ir simuliavimui.

n – matė kopula yra daugiamatė pasisikirstymo funkcija C, tolygiai pasiskirsčiusi kube $[0,1]^n$ ir turinti tokias savybes:

1. C: $[0,1]^n \to [0,1]$;
2. $C_i(u) = C(1,...,1,u,1,...,1) = u$ su $\forall u \in [0,1]$.

Akivaizdu, kad, jei $F_1,...,F_n$ yra vienamatės pasiskirstymo funkcijos, tai $C(F_1(x_1),...,F_n(x_n))$ - daugiamatė pasiskirstymo funkcija.

Teorema. Tegu F – n-matė pasiskirstymo funkcija su marginaliaisiais skirstiniais $F_1,...,F_n$. Tuomet ji turi vienintelę kopulos išraišką

$$F(x_1,...x_n) = C(F_1(x_1),...,F_n(x_n)).$$

Teiginys. Tegu F – n-matė pasiskirstymo funkcija su marginaliaisiais skirstiniais $F_1,...,F_n$ ir kopula C, tenkinančia teoremą. Tada kiekvienam $\boldsymbol{u} = (u_1,...,u_n)$ iš $[0,1]^n$:

$$C(u_1,...u_n) = F(F_1^{-1}(u_1),...,F_n^{-1}(u_n)),$$

čia F_i^{-1} - atvirkštinė F_i funkcija.

Kleitono kopula yra Archimedo kopulos vienas iš pavyzdžių. Archimedo kopulos forma yra:

$$C(u_1,...,u_n) = \psi^{-1}(\psi(u_1)+...+\psi(u_n)) \text{ visiems } 0 \le u_1,...,u_n \le 1.$$

Čia funkcija ψ - taip vadinamas generatorius, tenkinantis tokias sąlygas:

i. $\psi(0) = \infty$;

ii. $\psi(1) = 0$;

iii. $\forall t \in (0,1)$, $\psi'(t) < 0$, t.y. ψ - mažėjanti;

iv. $\forall t \in (0,1)$, $\psi''(t) \ge 0$, t.y. ψ - įgaubta.

Kleitono kopula dvimačiu atveju užrašoma tokia forma:

$$\psi(t) = t^{-\alpha} - 1,\ \alpha > 0,\ C(u_1,u_2) = (u_1^{-\alpha} + u_2^{-\alpha} - 1)^{-1/\alpha}.$$

1.4.1 Kleitono kopulos simuliavimo algoritmas dvimačiu atveju

Dvimačiu atveju turime:

$$\mathrm{P}\{U_2 \le u_2 \,|\, U_1 \le u_1\} = C_{2\backslash 1}(u_1,u_2),$$

$$\text{čia } C_{2\backslash 1}(u_1,u_2) = \lim_{\Delta \to 0} \frac{C(u_1+\Delta,u_2) - C(u_1,u_2)}{\Delta} = \frac{\partial C(u_1,u_2)}{\partial u_1}.$$

Algoritmas:

- Sugeneruoti du nepriklausomus atsitiktinius, tolygiai pasiskirsčiusius intervale $[0,1]$, dydžius v_1 ir v_2;
- Pažymėti $u_1 = v_1$;
- Tegu $C(u_2;u_1) = C_{2\backslash 1}(u_1,u_2)$. Tada randame $u_2 = C^{-1}(v_2;u_1)$;
- Vektorius (T_1,T_2) tuomet sugeneruojamas iš kopulos C.

Reikia įsitikinti, kad taip generuojant, gausime tai, ko mums reikia. Taigi pasižiūrėkime, kam bus lygi $\mathrm{P}\{U_2 \le u_2 \,|\, U_1 \le u_1\}$.

$$\mathrm{P}\{U_2 \le u_2 \,|\, U_1 \le u_1\} = \lim_{h\downarrow 0} P\{U_2 \le u_2 \,|\, U_1 \in [u_1,u_1+h]\} =$$

$$= \lim_{h\downarrow 0} \frac{P\{U_1 \in [u_1,u_1+h] \,|\, U_2 \le u_2\}}{P\{U_1 \in [u_1,u_1+h]\}} = \lim_{h\downarrow 0} \left[\frac{P\{U_1 \le u_1+h, U_2 \le u_2\} - P\{U_1 \le u_1, U_2 \le u_2\}}{P\{U_1 \le u_1+h\} - P\{U_1 \le u_1\}} \right] h =$$

$$= \frac{\frac{\partial}{\partial u_1} P\{U_1 \le u_1, U_2 \le u_2\}}{\frac{\partial}{\partial u_1} u_1} = \frac{\partial}{\partial u_1} P\{U_1 \le u_1, U_2 \le u_2\} = C_{2\backslash 1}(u_1,u_2).$$

Pabandykime apskaičiuoti, kam bus lygus $u_2 = C^{-1}(v_2;u_1)$, nes tai bus naudinga simuliuojant algoritmą su SAS paketu.

Taigi, turime, kad:

$$C(u_1,u_2) = \left(u_1^{-\alpha} + u_2^{-\alpha} - 1\right)^{-1/\alpha},$$

$$\frac{\partial C(u_1,u_2)}{\partial u_1} = -\frac{1}{\alpha}\left(u_1^{-\alpha} + u_2^{-\alpha} - 1\right)^{-\frac{1}{\alpha}-1}(-\alpha u_1^{-\alpha-1}),$$

$$\left(u_1^{-\alpha} + u_2^{-\alpha} - 1\right)^{-\frac{1}{\alpha}-1} u_1^{-\alpha-1} = v_2,$$

$$\left(u_1^{-\alpha}+u_2^{-\alpha}-1\right)^{-\frac{1+\alpha}{\alpha}}=v_2u_1^{\alpha},$$

$$\left(u_1^{-\alpha}+u_2^{-\alpha}-1\right)=u_1^{-\alpha}v_2^{-\frac{\alpha}{1+\alpha}},$$

$$u_2^{-\alpha}=u_1^{-\alpha}v_2^{-\frac{\alpha}{1+\alpha}}-u_1^{-\alpha}+1,$$

$$u_2=\left(u_1^{-\alpha}v_2^{-\frac{\alpha}{1+\alpha}}-u_1^{-\alpha}+1\right)^{-1/\alpha}.$$

1.4.2 Kleitono kopulos simuliavimo algoritmas daugiamačiu atveju

Tam, kad simuliuoti daugiamatį atvejį, užtenka paimti k=3.

Daugiamačiu atveju turime:

$$C_{m\backslash 1,\dots,m-1}(u_1,\dots,u_m) = P\left\{U_m \le u_m \mid (U_1,\dots,U_{m-1}) = (u_1,\dots,u_{m-1})\right\} =$$

$$= \frac{\partial^{m-1}_{u_1,\dots,u_{m-1}} C(u_1,\dots,u_m,1,\dots,1)}{\partial^{m-1}_{u_1,\dots,u_{m-1}} C(u_1,\dots,u_{m-1},1,\dots,1)}, \text{ čia } m = 2,\dots,n.$$

Tegu $m = 3$. Tada algoritmas bus:

- Sugeneruoti tris atsitiktinius, tolygius intervale $[0,1]$, dydžius v_1, v_2 ir v_3;
- Nustatyti $u_1 = v_1$, $u_2 = v_2$;
- Randame $u_3 = C^{-1}(v_3; u_1, u_2)$;
- Vektorius (T_1, T_2, T_3) tuomet sugeneruojamas iš copulos C.

Pabandykime apskaičiuoti, kam bus lygus $u_3 = C^{-1}(v_3; u_1, u_2)$, nes tai bus naudinga simuliuojant daugiamatį algoritmą su SAS paketu.

Pirmiausia reikia rasti, kam bus lygi funkcija $C(u_1, u_2, u_3)$. Iš anksčiau žinome, kad $C(u_1,\dots,u_n) = \psi^{-1}\left(\psi(u_1) + \dots + \psi(u_n)\right)$, o mūsų $\psi(t) = t^{-\alpha} - 1$, $\alpha > 0$.

Taigi, turime, kad:

$$\psi(t) = t^{-\alpha} - 1 = u,$$

$$t^{-\alpha} = 1 + u,$$

$$t = (1+u)^{-1/\alpha}.$$

Tuomet

$$C(u_1,u_2,u_3) = \left(1 + u_1^{-\alpha} - 1 + u_2^{-\alpha} - 1 + u_3^{-\alpha} - 1\right)^{-1/\alpha} = \left(u_1^{-\alpha} + u_2^{-\alpha} + u_3^{-\alpha} - 2\right)^{-1/\alpha}.$$

Dabar

$$v_3 = \frac{\dfrac{\partial^2 C(u_1,u_2,u_3)}{\partial u_1 \partial u_2}}{\dfrac{\partial^2 C(u_1,u_2)}{\partial u_1 \partial u_2}}.$$

Atliekame skaičiavimus:

$$\frac{\partial C(u_1,u_2)}{\partial u_1} = -\frac{1}{\alpha}\left(u_1^{-\alpha}+u_2^{-\alpha}-1\right)^{-\frac{1}{\alpha}-1}(-\alpha u_1^{-\alpha-1}) = u_1^{-\alpha-1}\left(u_1^{-\alpha}+u_2^{-\alpha}-1\right)^{-\frac{1+\alpha}{\alpha}},$$

$$\frac{\partial^2 C(u_1,u_2)}{\partial u_1 \partial u_2} = (u_1 u_2)^{-\alpha-1}(1+\alpha)\left(u_1^{-\alpha}+u_2^{-\alpha}-1\right)^{-\frac{1+2\alpha}{\alpha}},$$

$$\frac{\partial C(u_1,u_2,u_3)}{\partial u_1} = -\frac{1}{\alpha}\left(u_1^{-\alpha}+u_2^{-\alpha}+u_3^{-\alpha}-2\right)^{-\frac{1}{\alpha}-1}(-\alpha u_1^{-\alpha-1}) = u_1^{-\alpha-1}\left(u_1^{-\alpha}+u_2^{-\alpha}+u_3^{-\alpha}-2\right)^{-\frac{1+\alpha}{\alpha}},$$

$$\frac{\partial^2 C(u_1,u_2,u_3)}{\partial u_1 \partial u_2} = (u_1 u_2)^{-\alpha-1}(1+\alpha)\left(u_1^{-\alpha}+u_2^{-\alpha}+u_3^{-\alpha}-2\right)^{-\frac{1+2\alpha}{\alpha}}.$$

Tada

$$v_3 = \frac{\dfrac{\partial^2 C(u_1,u_2,u_3)}{\partial u_1 \partial u_2}}{\dfrac{\partial^2 C(u_1,u_2)}{\partial u_1 \partial u_2}} = \frac{(u_1 u_2)^{-\alpha-1}(1+\alpha)\left(u_1^{-\alpha}+u_2^{-\alpha}+u_3^{-\alpha}-2\right)^{-\frac{1+2\alpha}{\alpha}}}{(u_1 u_2)^{-\alpha-1}(1+\alpha)\left(u_1^{-\alpha}+u_2^{-\alpha}-1\right)^{-\frac{1+2\alpha}{\alpha}}} =$$

$$= \frac{\left(u_1^{-\alpha}+u_2^{-\alpha}+u_3^{-\alpha}-2\right)^{-\frac{1+2\alpha}{\alpha}}}{\left(u_1^{-\alpha}+u_2^{-\alpha}-1\right)^{-\frac{1+2\alpha}{\alpha}}}.$$

Liko išsireikšti, kam bus lygu u_3:

$$v_3 = \frac{\left(u_1^{-\alpha}+u_2^{-\alpha}+u_3^{-\alpha}-2\right)^{-\frac{1+2\alpha}{\alpha}}}{\left(u_1^{-\alpha}+u_2^{-\alpha}-1\right)^{-\frac{1+2\alpha}{\alpha}}} = \frac{\left(u_1^{-\alpha}+u_2^{-\alpha}-1+(u_3^{-\alpha}-1)\right)^{-\frac{1+2\alpha}{\alpha}}}{\left(u_1^{-\alpha}+u_2^{-\alpha}-1\right)^{-\frac{1+2\alpha}{\alpha}}},$$

$$v_3^{-\frac{\alpha}{1+2\alpha}} = 1 + \frac{\left(u_3^{-\alpha}-1\right)}{\left(u_1^{-\alpha}+u_2^{-\alpha}-1\right)},$$

$$u_3 = \left[\left(v_3^{-\frac{\alpha}{1+2\alpha}}-1\right)\left(u_1^{-\alpha}+u_2^{-\alpha}-1\right)+1\right]^{-1/\alpha}.$$

2 Praktinė dalis

2.1 Atsitiktinio vektoriaus (T_1, T_2) generavimas, pasinaudojant Kleitono kopula

Buvo generuojami dvimačiai atsitiktiniai vektoriai (T_1, T_2), kurių pasiskirstymo funkcija turi pavidalą

$$F(x_1, x_2) = F_1^{-\,'}(x_1) + F_2^{-\,'}(x_2) - \;)^{-\;/\theta},\ \theta > 0,$$

čia $F_i(x_i) = \;\; T_i \leq \;_i\}$, θ yra Kleitono kopulos parametras, charakterizuojantis atsitiktinių dydžių T_1 ir T_2 priklausomybę.

Žinome, kad Kleitono kopulos parametras θ yra susijęs su Kendalo koreliacijos koeficientu τ, $0 < \tau <$. Kai τ yra arti 0, tai turime silpną prilausomybę, o kai τ yra arti 1 – stiprią priklausomybę.

θ ir τ sąryšis apibrėžiamas formule

$$\hat{\theta} = \frac{2\tau}{1-}.$$

Todėl, generuojant naudosime tokias θ:

$\theta = 0.5$, tai $\tau = \dfrac{\theta}{2+\theta} = \dfrac{0.5}{2.5} = 0.2$ - pakankamai maža priklausomybė, ir

$\theta = 5$, tai $\tau = \dfrac{\theta}{2+\theta} = \dfrac{5}{2+} = \dfrac{5}{7}$ - pakankamai didelė priklausomybė.

Fiksavus imties dydį n, imties realizacijos buvo generuotos M=3000 kartų. Generavau duomenis su tokiais parametrais:

1) $\theta = 0.5$, $n = \;00$;
2) $\theta = 0.5$, $n = 200$;
3) $\theta = 0.5$, $n = 500$;
4) $\theta = 0.5$, $n = \;000$;
5) $\theta = 5$, $n = \;00$;
6) $\theta = 5$, $n = 200$;
7) $\theta = 5$, $n = 500$;

8) $\theta =$, $n = 000$.

Visų pirma, reikia sugeneruoti atsitiktinį vektorių (T_1, T_2), naudojantis Kleitono kopula. Taigi generuojame Kleitono kopulą taip:

1) Sugeneruojame du nepriklausomus atsitiktinius, tolygiai pasiskirsčiusius intervale [0,1], dydžius V_1 ir V_2;
2) Pažymime $U_1 = V_1$;
3) Randame $U_2 = \left(U_1^{-} V_2^{-\frac{\theta}{1+}} - u_1^{-} + \right)^{/\theta}$. Tokiu būdu gauname atsitiktinį vektorių (U_1, U_2), kurio pasiskirstymo funkcija yra $F(u_1, u_2) = (u_1^{-} + u_2^{-} -)^{-/\theta}$;
4) Generuojame atsitiktinį dydį (T_1, T_2),pasirinkę konkretų skirstinį (pavyzdžiui, eksponentinį), naudodamiesi Kleitono kopula ir jau rastais dydžiais U_1 ir U_2.

Nagrinėsime tokius Kleitono kopulos generavimo atvejus:

a) Tarkime, kad gyvenimo trukmės turi tokį patį eksponentinį skirstinį

$$F_1(x) = 1 - \,^{-} ,$$
$$F_2(x) = 1 - \,^{-} .$$

Kadangi gyvenimo trukmės turi tokį patį eksponentinį skirstinį, tai reikia rasti vektorių (T_1, T_2), kur

$$T_1 = - \ln(1 - U_1),$$
$$T_2 = - \ln(1 - U_2).$$

b) Tarkime, kad pirmosios komponentės skirstinys yra eksponentinis, o antros – Veibulo:

$$F_1(x) = 1 - \,^{-} ,$$

$$F_2(x) = 1 - \,^{-\nu} ,$$

tegu $\upsilon =$ 0.3, 0.5, 0.7, 0.9, 1.1, 1.3, 1.5.

Kadangi gyvenimo trukmės turi eksponentinį ir Veibulo skirstinius, tai reikia rasti vektorių (T_1, T_2), kur

$$T_1 = -\ln(1-U_1),$$

$$T_2 = -\frac{1}{\upsilon}\ln(1-U_2).$$

Nagrinėjame cenzūruotas imtis, todėl pirmiausia generuojame atsitiktinius vektorius (T_{1i}, T_{2i}) ir indikatorius δ_i, $i = 1,\dots,n$, naudojantis Kleitono kopula. Tegul cenzūruotų duomenų bus $25\,\%$ nuo imties dydžio, cenzūravimą generuojame atsitiktinai. Tokių imčių bus M=3000.

Kleitono kopulos generavimo programą su SAS paketu žiūrėti Priede B.

Kai skirstiniai sutampa gauname tokius generuotus duomenis, naudojant Kleitono kopulą:

Kai $\theta = 0.5$:

T1	T2
1.3000999709	0.7823410566
0.8535383201	1.3678406448
1.5241738337	3.9494340037
0.1484818908	0.6786985482
0.0012098518	0.0157378799
0.4716743844	0.240803918
1.2141264928	2.7391148173
0.9440120644	0.4090497125
1.0496917817	0.7976982546
0.5881099246	1.3810915594
0.4852368717	1.0302841198
0.2080491121	0.6941486711
1.3575826781	0.8980566964
1.7263192906	0.7843136448
1.2779811294	1.1907696844
0.9710676466	1.112367521
0.2945048312	0.3335999243
0.6067458403	1.4238786827
1.7590919671	0.6942294357
0.1008604293	1.341471882

Kai $\theta =$ [illegible]:

T1	T2
0.5029093563	0.2617316694
0.3428272785	0.4318340397
0.6009677578	0.2061530396
0.097738728	0.1421667671
0.1129598647	0.1131407607
1.4973585281	1.4387304324
0.5436143828	0.5422933128
0.6565416173	0.7256201673
0.9314944968	0.5305328645
1.5295920796	1.5231728412
1.0701235283	1.2526875815
0.7842700875	1.0057067468
0.2971038546	0.2700967075
0.0474767066	0.0698962883
0.2847353688	0.4303200912
1.4576179436	1.941766751
0.074909357	0.0651046308
0.5753687721	1.6066584298
0.5758008397	0.5279904234
0.8675384652	1.1418848612

Kai skirstiniai nesutampa gauname tokius generuotus duomenis, naudojant Kleitono kopulą:

Kai $\theta = 0.5$:

T1	T2
0.9908772909	0.2983163438
0.0741912315	0.2064246329
1.2111232737	0.5865890513
1.2097219326	0.6169324529
0.2401811123	1.2933143137
1.899896713	0.8805780083
0.1741823358	0.5462246512
0.0936648443	0.0921013132
1.1806681004	0.0630830812
0.305923772	1.0004703515
0.6864049873	0.5932081532
0.965461935	1.9903226993
1.1100601647	1.9217395162
1.3735857929	0.9770048395
0.3862799191	1.9821518224
0.089854403	0.2541982085
0.2647739743	0.3765848217
0.4357203197	0.0826814034
0.2613814653	0.5344678844
0.4220340335	0.1710236421

Kai $\theta =$ [illegible]:

T1	T2
1.0902337525	0.7875168671
0.8866867746	0.777385248
0.8134490458	0.6904702273
0.9269021049	0.474770755
1.2472902088	2.1099444671
0.3017435758	0.251022749
0.1416873352	0.1298891456
0.4470979374	0.2219830999
0.6023843629	1.2616408194
0.0420390854	0.0364231884
0.5631296391	0.4319403601
0.4602303813	0.4291115529
0.0429572297	0.0302510583
0.239743744	0.1523273568
0.1414267463	0.9313129829
0.9605947545	0.7193908095
1.9483919463	0.719573112
0.5359936875	0.3016457483
1.7998413918	0.8840194586
7.2013486546	2.2858869429

2.2 Homogeniškumo kriterijus

Sugeneravę vektorius (T_{1i}, T_{2i}), galime tikrinti homogeniškumo hipotezę

$$H_0 : F_1 = F_2,$$

t.y. kad marginaliosios pasiskirstymo funkcijos sutampa. Tikrinimui naudosime kriterijų, aprašytą 1.3 skyrelyje.

Visais atvejais pirmiausia reikia kiekvienam $j = 1,...,n$ suskaičiuoti tokius indikatorius:

$$Y_{1i}(X_{1j}) = I_{\{X_{1i} \geq X_{1j}\}},\ Y_{1i}(X_{2j}) = I_{\{X_{1i} \geq X_{2j}\}},$$

$$Y_{2i}(X_{1j}) = I_{\{X_{2i} \geq X_{1j}\}},\ Y_{2i}(X_{2j}) = I_{\{X_{2i} \geq X_{2j}\}},$$

$$Y_1(X_{1j}) = \sum_{i=1}^{n} I_{\{X_{1i} \geq X_{1j}\}},\ Y_1(X_{2j}) = \sum_{i=1}^{n} I_{\{X_{1i} \geq X_{2j}\}},$$

$$Y_2(X_{1j}) = \sum_{i=1}^{n} I_{\{X_{2i} \geq X_{1j}\}},\ Y_2(X_{2j}) = \sum_{i=1}^{n} I_{\{X_{2i} \geq X_{2j}\}},$$

$$Y(X_{1j}) = Y_1(X_{1j}) + Y_2(X_{1j}),\ Y(X_{2j}) = Y_1(X_{2j}) + Y_2(X_{2j}).$$

Kai duomenys cenzūruoti, $X_{ij} = T_{ij} \wedge C_{ij}$, $\delta_{ij} = I_{\{T_{ij} \leq C_{ij}\}}$, čia T_{ij} - mirties momentai, C_{ij} - cenzūravimo momentai. Tuomet pagal formules randame statistiką V ir $\hat{C}$. Kaip jau minėjau anksčiau, skaičiavimams su SAS paketu imsime svorių funkcijas

$$K(X_{1j}) = K(X_{2j}) = \frac{1}{\sqrt{n}}.$$

Statistikos V išraiška tuomet bus tokia:

$$V = \frac{1}{\sqrt{n}} \sum_{j=1}^{n} \frac{\delta_{1j} Y_2(X_{1j})}{Y(X_{1j})} - \frac{1}{\sqrt{n}} \sum_{j=1}^{n} \frac{\delta_{2j} Y_1(X_{2j})}{Y(X_{2j})}.$$

$\hat{C}$ išraiška bus:

$$\hat{C} = \frac{1}{n} \sum_{s=1}^{2} \sum_{j=1}^{n} \delta_{sj} \frac{Y_1(X_{sj}) Y_2(X_{sj})}{Y^2(X_{sj})}.$$

Tam, kad galėtume rasti σ^2, kuri yra $\sigma^2 = \ldots - \hat{C}(1,2)$, reikia kiekvienam $j = 1,...,n$ apsiskaičiuoti R_{1j}, R_{2j}, R_{3j} ir R_{4j}. Norint rasti R_{3j} ir R_{4j} reikia dar kiekvienam $l = 1,...,n$ suskaičiuoti indikatorius $Y_{1j}(X_{sl})$, $Y_{2j}(X_{sl})$:

$$Y_{1j}(X_{sl}) = \mathbb{1}_{\{X_{1j} \geq X_{sl}\}},$$

$$Y_{2j}(X_{sl}) = \mathbb{1}_{\{X_{2j} \geq X_{sl}\}}.$$

Tuomet

$$\hat{C}(1,2) = \frac{1}{n}\sum_{j=1}^{n} \left(R_{1j}R_{2j} - R_{1j}R_{4j} - R_{2j}R_{3j} + R_{3j}R_{4j} \right), \text{ kur}$$

$$R_{1j} = \delta_{1j}\hat{Q}_1(X_{1j}) = \delta_{1j}\frac{\sqrt{n}K(X_{1j})Y_2(X_{1j})}{Y(X_{1j})} = \frac{\delta_{1j}Y_2(X_{1j})}{Y(X_{1j})},$$

$$R_{2j} = \delta_{2j}\hat{Q}_2(X_{2j}) = \delta_{2j}\frac{\sqrt{n}K(X_{2j})Y_1(X_{2j})}{Y(X_{2j})} = \frac{\delta_{2j}Y_1(X_{2j})}{Y(X_{2j})},$$

$$R_{3j} = \sum_{s=1}^{2}\sum_{l=1}^{n}\delta_{sl}\hat{Q}_1(X_{sl})\frac{Y_{1j}(X_{sl})}{Y(X_{sl})} = \sum_{l=1}^{n}\delta_{1l}\frac{Y_2(X_{1l})Y_{1j}(X_{1l})}{Y^2(X_{1l})} + \sum_{l=1}^{n}\delta_{2l}\frac{Y_2(X_{2l})Y_{1j}(X_{2l})}{Y^2(X_{2l})},$$

$$R_{4j} = \sum_{s=1}^{2}\sum_{l=1}^{n}\delta_{sl}\hat{Q}_2(X_{sl})\frac{Y_{2j}(X_{sl})}{Y(X_{sl})} = \sum_{l=1}^{n}\delta_{1l}\frac{Y_1(X_{1l})Y_{2j}(X_{1l})}{Y^2(X_{1l})} + \sum_{l=1}^{n}\delta_{2l}\frac{Y_1(X_{2l})Y_{2j}(X_{2l})}{Y^2(X_{2l})}.$$

Randame σ^2 ir suskaičiuojame $Z_n = \frac{U}{\hat{\sigma}}$.

Hipotezė H_0 yra atmetama su asimptotiniu reikšmingumo lygmeniu $\alpha = 0.1$, jei

$|Z_n| > z_{\alpha/2}$.

Pagal šį kriterijų su SAS paketu pasirašiau programą, ją galima pažiūrėti Priede B.

2.3 Homogeniškumo hipotezės tikrinimas su SAS

Pradžioje tikrinsime homogeniškumo hipotezę, kuri aprašyta 2.2 skyrelyje, kai skirstiniai sutampa, kur skirstiniai ir duomenų generavimas aprašytas 2.1 skyrelyje. Duomenis generavau pagal Kleitono kopulos algoritmą, kuris taip pat aprašytas 2.1 skyrelyje.

Tuomet, kai jau turime sugeneruotus duomenis, galime tikrinti homogeniškumo hipotezę.

Mūsų tikslas – pažiūrėti, kur artėja reikšmingumo lygmuo didinant imties dydį. Taigi tam generuojame imties dydžius $n = 00$, $n = 200$, $n = 500$, $n = 000$ su Kleitono parametrais $\theta = 0.5$ ir $\theta = $.

Suskaičiuojame statistiką V ir $\hat{\sigma}$, tuomet suskaičiuojame statistiką Z_n ir kritinę reikšmę

$z_{0.05} = .644854$.

Tam, kad suskaičiuotume reikšmingumo lygmenį, reikia suskaičiuoti, kiek kartų hipotezė buvo atmesta, kai skirstiniai sutampa, ir tą skaičių padalinti iš imčių skaičiaus, kuris yra M=3000. Programas su SAS paketu žiūrėti Priede B.

Gauname tokius rezultatus:

$\theta = 0.5$	N=100	N=200	N=500	N=1000
Reikšmingumo lygmuo α	0.077667	0.069667	0.057667	0.074667

$\theta = $	N=100	N=200	N=500	N=1000
Reikšmingumo lygmuo α	0.037333	0.042667	0.038333	0.046667

<u>Išvada:</u>

Iš gautų rezultatų matome, kad prie Kleitono kopulos parametro $\theta = $ reikšmingumo lygmuo yra mažesnis nei prie parametro $\theta = 0.5$. Matome, kad reikšmingumo lygmuo lėtai konverguoja į tikrąjį reikšmingumo lygmenį.

Tada generavau duomenis, kai skirstiniai nesutampa. Viską dariau analogiškai, kaip ir tuo atveju, kai skirstiniai sutampa, tik žiūrime, kas vyksta su kriterijaus galia, kai n didėja. Kriterijaus galia apskaičiuojama taip: skaičiuojame, kiek kartų hipotezė buvo atmesta, ir tą skaičių daliname iš imčių skaičiaus, kuris mano atveju yra 3000.

Gauname tokius rezultatus:

$\theta = 0.5$	N=100	N=200	N=500	N=1000
Kriterijaus galia, kai $\upsilon = 0.3$	1	1	1	1
kai $\upsilon = 0.5$	0.999333	1	1	1
kai $\upsilon = 0.7$	0.816	0.98	1	1
kai $\upsilon = 0.9$	0.162	0.266333	0.500667	0.775
kai $\upsilon = 1.1$	0.143333	0.230667	0.421	0.705333
kai $\upsilon = 1.3$	0.582667	0.850667	0.997333	1
kai $\upsilon = 1.5$	0.896667	0.995	1	1
kai $\upsilon = 2$	1	1	1	1

Išvada:

Taigi, kai $\theta = 0.5$, mažiausia kriterijaus galia yra prie parametrų $\upsilon = 0.9$ ir $\upsilon = 1.1$.

O didžiausia kriterijaus galia yra prie parametrų $\upsilon = 0.3$ ir $\upsilon = 2$.

Be to, didėjant imties dydžiui N, kriterijaus galia visais atvejais didėja ir beveik visur yra lygi 1 prie imties dydžio N=1000, išskyrus atvejus, kai $\upsilon = 0.9$ ir $\upsilon = 1.1$.

$\theta =$ [illegible]	N=100	N=200	N=500	N=1000
Kriterijaus galia, kai $\upsilon = 0.3$	1	1	1	1
kai $\upsilon = 0.5$	1	1	1	1
kai $\upsilon = 0.7$	0.989	0.999667	1	1
kai $\upsilon = 0.9$	0.226333	0.414667	0.808333	0.984333
kai $\upsilon =$ [illegible].1	0.176333	0.356667	0.708667	0.946333
kai $\upsilon =$ [illegible].3	0.871333	0.993	1	1
kai $\upsilon =$ [illegible].5	0.996667	1	1	1
kai $\upsilon =$ [illegible]	1	1	1	1

Išvada:

Kai $\theta =$ [illegible], mažiausia kriterijaus galia yra prie parametrų $\upsilon = 0.9$ ir $\upsilon =$ [illegible].1.

O didžiausia kriterijaus galia yra prie parametrų $\upsilon = 0.3$, $\upsilon = 0.5$, $\upsilon = 0.7$, $\upsilon =$ [illegible].5, ir $\upsilon =$ [illegible]. Taip pat, kai $\theta =$ [illegible], gauname didesnes kriterijaus galios reikšmes nei atveju, kai $\theta = 0.5$.

Be to, didėjant imties dydžiui N, kriterijaus galia visais atvejais didėja ir beveik visur yra lygi 1 prie imties dydžio N=1000, išskyrus atvejus, kai $\upsilon = 0.9$ ir $\upsilon =$ [illegible].1.

2.4 *Homogeniškumo hipotezės tikrinimas, kai turim konkrečius duomenis*

Turime duotą tokią duomenų lentelę:

x1	6	10	12*	16	16	19	20	27	27	52
x2	140	272	12*	568	62	470*	523	356	291	631
x1	57	70	89	117	178	204	217	218	222	249
x2	330	56	398	47	271	406	417	635	169	423
x1	305	328	421	423	464	500	546	555	566*	680
x2	624	270	506	104	6	130	1871	10	566*	414
x1	685	707	764	863	1012	1066	1077	1089	1107	1142
x2	1219	1516	206	267	249	1376	450	2168	71	93
x1	1169	1213	1222	1244	1329	1486	1682*	1734	1746	1776
x2	652	2128	196	821	99	120	152	1516	872	2023
x1	1888	1978	2021	2317	2391	2470	2621	2699	3496	4146
x2	541	168	2161	752	781	1518	1716	1169	2613	1355

Imties dydis n=60, cenzūruotų duomenų yra 6 iš 120.

Reikia patikrinti hipotezę, kad laikai grupėje x1 ir grupėje x2 pasiskirstę vienodai.

Su SAS programa paskaičiuojame statistiką V, $\hat{\sigma}$, ir kriterijaus statistiką Z_n. Output lange gauname tokius rezultatus:

V
-1.210693

sigma
0.5291756

Z_n
-2.287886

krit_r
1.959964

Hipotezę atmetame su asimptotiniu reikšmingumo lygmeniu $\alpha = 0.05$, jei $|Z_n| > z_{\alpha/2}$.

Iš output lange esančių rezultatų matome, kad kritinė reikšmė $z_{0.025} = 1.959964$.

Taigi, gavome, kad $|Z_n| > z_{\alpha/2}$, o tai reiškia, kad hipotezė atmetama, t.y. laikai grupėje x1 ir grupėje x2 pasiskirstę nevienodai.

IŠVADOS

Suformulavau homogeniškumo kriterijus dvimačiu ir daugiamačiu atvejais, bei parašiau programą su SAS paketu, kuri tikrintų homogeniškumo hipotezę dvimačiu atveju.

Kleitono kopula yra naudinga daugiamačių skirstinių konstravimui ir simuliavimui . Susimuliavus duomenis, galime susimuliuoti bet kokį skirstinį, kokio mums reikia. Mano darbe buvo naudojamas eksponentinis skirstinys, tačiau galima naudoti ir kitokį.

Labai naudinga buvo parašyti programą su SAS paketu homogeniškumo kriterijui, kai k=2, nes tokių kriterijų, skirtų cenzūruotoms priklausomoms imtims, nėra. Tuomet patikrinau tą kriterijų su simuliuotais duomenimis.

Kai skirstiniai sutampa ir imtis cenzūruota, gauname tokius rezultatus:

1) Kai $\theta = 0.5$ ir n didėja, reikšmingumo lygmuo lėtai konverguoja į pasirinktąjį reikšmingumo lygmenį 0,1.
2) Kai $\theta =$ [illegible] ir n didėja, reikšmingumo lygmuo dar lėčiau, negu atveju, kai $\theta = 0.5$, konverguoja į pasirinktąjį reikšmingumo lygmenį 0,1. Be to, šiuo atveju gauname, reikšmingumo lygmens α mažesnes reikšmes negu atveju $\theta = 0.5$.

Kai skirstiniai nesutampa, gauname tokius rezultatus:

1) Ir $\theta = 0.5$, kriterijaus galia lygi 1 prie parametrų $\upsilon = 0.3$ ir $\upsilon =$ [illegible]. Kitais atvejais kriterijaus galia didėjant n artėja į 1, tačiau su parametrais $\upsilon = 0.9$ ir $\upsilon =$ [illegible]$.1$, netgi prie n=1000 dydžio imties, net nepasiekia 1.
2) Kai $\theta =$ [illegible], kriterijaus galia lygi 1 prie parametrų $\upsilon = 0.3$, $\upsilon = 0.5$ ir $\upsilon =$ [illegible]. Kriterijaus galia yra labai arti 1 prie parametrų $\upsilon = 0.7$ ir $\upsilon =$ [illegible]$.5$. Vėlgi kitais atvejais kriterijaus galia didėjant n artėja į 1, tačiau su parametrais $\upsilon = 0.9$ ir $\upsilon =$ [illegible]$.1$, netgi prie n=1000 dydžio imties, net nepasiekia 1.

Taigi, gauname, kad visais atvejais, kai skirstiniai sutampa ir n didėja, reikšmingumo lygmuo lėtai konverguoja į pasirinktą reikšmingumo lygmenį. O visais atvejais, kai skirstiniai nesutampa ir n didėja, kriterijaus galia artėja į 1,o prie kai kurių parametrų netgi prie visų imties dydžių lygi 1, o tai reiškia, kad visais atvejais hipotezė apie skirstinių lygybę buvo atmesta.

SUMMARY

Projects purpose – to describe homogeneity tests for related censored samples.

It were challenges:

1. To learn how simulate Clayton‘s copula , to know what it is Clayton‘ copula;
2. To describe homogeneity tests for related censored paired samples;
3. To test homogeneity hypotesis, when data is simulated by Clayton‘s copula.

And the results are:

If we have censored data, with Clayton‘s parameter $\theta = 0.5$ and $\theta =$ [illegible],and distributions are the same, significance level is near chosen asymptotic significance level.

If we have censored data, with Clayton‘s parameter $\theta = 0.5$ and $\theta =$ [illegible],and distributions are not the same, the test power for most parametres υ is 1 and it means that the hypotesis was rejected for all samples, and only for several test power wasn‘t 1 when the sample is n=1000.

So, for my simulated samples, the homogeneity test for related censored samples, was effective.

LITERATŪRA IR ŠALTINIAI

4. Vilijandas BAGDONAVIČIUS, Julius KRUOPIS, Mikhail NIKULIN "Non – parametric tests for censored data", 2010m.;
5. Rūta LEVULIENĖ, "Statistika su SAS";
6. Vilijandas BAGDONAVIČIUS "Išgyvenamumo analizė"
7. www.SAS.com;
8. SAS help and documentation;
9. E. Cuvelier, M. Noirhomme – Fraiture straipsnis "Clayton copula and mixture decomposition";
10. www.vosesoftware.com
11. Claudio Romano straipsnis "Calibrating and simulating copula", 2002m.

PRIEDAS A

Lenglaro nelygybė. Tarkime, kad H yra suderintas tolydus iš kairės su baigtinėmis ribomis iš dešinės procesas, N – skaičiuojantis procesas, A – jo kompensatorius, M=N-A. Tada su $\forall \varepsilon, \eta, \tau > 0$

$$\mathrm{P}\{\sup_{t \in [0,\tau]} [\int_0^\tau H(u)dM(u)]^2 \geq \varepsilon\} \leq \frac{\eta}{\varepsilon} + \mathrm{P}\{\int_0^\tau H^2(u)d<M>(u) \geq \eta\}.$$

Išvada. Jei $M = M^{(n)}$ ir $\int_0^\tau \left(H^{(n)}(u)\right)^2 d<M^{(u)}>(u) \xrightarrow{P} 0$, kai $n \to \infty$, tai

$$\sup_{[0,\tau]} \int_0^\tau H^{(n)}(u)dM^{(n)}(u) \xrightarrow{P} 0.$$

Čebyševo nelygybė. Jei egzistuoja atsitiktinio dydžio *X* baigtinė dispersija *VX*, tai bet kuriam $\varepsilon > 0$ teisinga nelygybė

$$\mathrm{P}\{|X - EX| \geq \varepsilon\} \leq \frac{VX}{\varepsilon^2}.$$

PRIEDAS B

SAS programos ir OUTPUT rezultatai

Simuliuojam dvimatę kopulą, kai skirstiniai sutampa:

```
title 'cenz imtis, skirstiniai sutampa';
%let imtys=3000;
%let n=100;
%let n=200;
%let n=500;
%let n=1000;
data tolygus2;
teta=5; /* ir teta=0.5 */
delta1=1;
delta2=1;
do imtis=1 to &imtys;
do i=1 to &n;
v1=uniform(0);
v2=uniform(0);
output; end;
end;
run;
```

Atsitiktinai sucenzūruojam 25% duomenų:

```
data tolygus2;
set tolygus2;
if (i=4 or i=12 or i=20 or i=28 or i=36 or i=44 or i=52 or i=60 or i=68 or
i=76 or i=84 or i=92 or i=100 or
i=108 or i=116 or i=124 or i=132 or i=140 or i=148 or i=156 or i=164 or i=172
or i=180 or i=188 or i=196 or i=204 or
i=212 or i=220 or i=228 or i=236 or i=244 or i=252 or i=260 or i=268 or i=276
or i=284 or i=292 or i=300 or i=308 or
i=316 or i=324 or i=332 or i=340 or i=348 or i=356 or i=364 or i=372 or i=380
or i=388 or i=396 or i=404 or i=412 or
i=420 or i=428 or i=436 or i=444 or i=452 or i=460 or i=468 or i=476 or i=484
or i=492 or i=500or i=508 or i=516 or
i=524 or i=532 or i=540 or i=548 or i=556 or i=564 or i=572 or i=580 or i=588
or i=596 or i=604 or i=612 or
i=620 or i=628 or i=636 or i=644 or i=652 or i=660 or i=668 or i=676 or i=684
or i=692 or i=700 or i=708 or i=716 or
i=724 or i=732 or i=740 or i=748 or i=756 or i=764 or i=772 or i=780 or i=788
or i=796 or i=804 or i=812 or
i=820 or i=828 or i=836 or i=844 or i=852 or i=860 or i=868 or i=876 or i=884
or i=892 or i=900 or i=908 or i=916 or
i=924 or i=932 or i=940 or i=948 or i=956 or i=964 or i=972 or i=980 or i=988
or i=996) then delta1=0;
else delta1=1;
if (i=1 or i=9 or i=17 or i=25 or i=33 or i=41 or i=49 or i=57 or i=65 or
i=73 or i=81 or i=89 or i=97 or i=105 or
i=113 or i=121 or i=129 or i=137 or i=145 or i=153 or i=161 or i=169 or i=177
or i=185 or i=193 or i=201 or i=209 or i=217 or
i=225 or i=233 or i=241 or i=249 or i=257 or i=265 or i=273 or i=281 or i=289
or i=297 or i=305 or i=313 or i=321 or i=329 or
i=337 or i=345 or i=353 or i=361 or i=369 or i=377 or i=385 or i=393 or i=401
or i=409 or i=417 or i=425 or i=433 or i=441 or
i=449 or i=457 or i=465 or i=473 or i=481 or i=489 or i=497 or i=505 or i=513
or i=521 or i=529 or i=537 or i=545 or i=553 or
```

```
i=561 or i=569 or i=577 or i=585 or i=593 or i=601 or i=609 or i=617 or i=625
or i=633 or i=641 or i=649 or i=657 or i=665 or
i=673 or i=681 or i=689 or i=697 or i=705 or i=713 or i=721 or i=729 or i=737
or i=745 or i=753 or i=761 or i=769 or i=777 or
i=785 or i=793 or i=801 or i=809 or i=817 or i=825 or i=833 or i=841 or i=849
or i=857 or i=865 or i=873 or i=881 or i=889 or
i=897 or i=905 or i=913 or i=921 or i=929 or i=937 or i=945 or i=953 or i=961
or i=969 or i=977 or i=985 or i=993)
then delta2=0;
else delta2=1;
u1=v1;
output;run;

data copula2;
set tolygus2;
u2=(1-u1**(-teta)+(v2**(-teta/(1+teta))*(u1**(-teta)))**(-1/teta);
output; run;

data  dvimate_copula2; /*generuojam T1 ir T2 pagal eksponentini skirstini*/
set copula2;
keep imtis teta delta1 delta2 T1 T2;
T1=-log(1-u1);
T2=-log(1-u2) ;
output; run;

proc sort data=dvimate_copula2 out=copula2;
by imtis;
run;
```

Suprogramuojame kriterijų ir tikriname homogeniškumo hipotezę:

```
proc iml;
use dvimate_copula2;
do w=1 to &imtys;
read all var{T1 T2} where(imtis=w) into x; *print x;
read all var{delta1 delta2} where(imtis=w) into delta; * print delta;
N=nrow(x);
y1=j(N,2,0);
y2=j(N,2,0);
do j=1 to N;
do i=1 to N;
if x[i,1]>=x[j,1] then y1[j,1]=y1[j,1]+1; /*Y1(X1j)*/
if x[i,1]>=x[j,2] then y1[j,2]=y1[j,2]+1; /*Y1(X2j)*/
if x[i,2]>=x[j,1] then y2[j,1]=y2[j,1]+1; /*Y2(X1j)*/
if x[i,2]>=x[j,2] then y2[j,2]=y2[j,2]+1; /*Y2(X2j)*/
end; end;
*print y1 y2;
V1=y2[,1]#delta[,1]/(y1[,1]+y2[,1]);
V2=y1[,2]#delta[,2]/(y1[,2]+y2[,2]);
*print V1 V2;
V=(1/sqrt(N))*(V1[+,]-V2[+,]);
*print V;
C1=y1[,1]#y2[,1]#delta[,1]/((y1[,1]+y2[,1])#(y1[,1]+y2[,1]));
C2=y1[,2]#y2[,2]#delta[,2]/((y1[,2]+y2[,2])#(y1[,2]+y2[,2]));
*print C1 C2;
C=(1/N)*(C1[+,]+C2[+,]);
*print C;
R1=y2[,1]#delta[,1]/(y1[,1]+y2[,1]); /*Y2(X1j)*delta1/Y(X1j)*/
R2=y1[,2]#delta[,2]/(y1[,2]+y2[,2]); /*Y1(X2j)*delta2/Y(X2j)*/
*print R1 R2;
r31=j(N,N,0);
r32=j(N,N,0);
```

```
r41=j(N,N,0);
r42=j(N,N,0);
do l=1 to N;
do j=1 to N;
if x[j,1]>=x[l,1] then r31[l,j]=1; else r31[l,j]=0;
if x[j,1]>=x[l,2] then r32[l,j]=1; else r32[l,j]=0;
if x[j,2]>=x[l,1] then r41[l,j]=1; else r41[l,j]=0;
if x[j,2]>=x[l,2] then r42[l,j]=1; else r42[l,j]=0;
end; end;
*print r31 r32 r41 r42;
R3_1=delta[,1]#y2[,1]#r31[,]/((y1[,1]+y2[,1])#(y1[,1]+y2[,1]));
R3_2=delta[,2]#y2[,2]#r32[,]/((y1[,2]+y2[,2])#(y1[,2]+y2[,2]));
R3=R3_1[+,]+R3_2[+,];
*print R3_1 R3_2 R3;
R4_1=delta[,1]#y1[,1]#r41[,]/((y1[,1]+y2[,1])#(y1[,1]+y2[,1]));
R4_2=delta[,2]#y1[,2]#r42[,]/((y1[,2]+y2[,2])#(y1[,2]+y2[,2]));
R4=R4_1[+,]+R4_2[+,];
*print R4_1 R4_2 R4;
R=(R1#R2-R1#R4`-R2#R3`+R3`#R4`);
C12=(1/N)*R[+,];
*print C12;
sigma=sqrt(C-2*C12);*print sigma;
Z_n=V/sigma; print Z_n;
krit_r=PROBIT(1-0.05/2); /* hipoteze atmetama su alpha, jeigu
|Z_n|>z(alpha/2) */
*print krit_r; end;
if abs(Z_n)>krit_r then print 'hipoteze su r.l. alpha=0.5 atmetama';
else print 'hipoteze neatmetama';
quit;
```

Skaičiuojame reikšmingumo lygmenį:

```
data lygmuo;
set statistika end=galas;
krit_r=PROBIT(1-0.1/2);
if    (abs(Z_n)>krit_r) then atmete=1;
else atmete=0;
suma+(atmete/&imtys);
if (galas) then put suma=;
run;
```

Kai skirstiniai nesutampa:

```
title 'cenz imtis, skirstiniai nesutampa';
%let imtys=3000;
%let n=100;
%let n=200;
%let n=500;
%let n=1000;
%let niu=0.1;
%let niu=10;
data tolygus2;
teta=5; /* ir teta=0.5 */
delta1=1;
delta2=1;
do imtis=1 to &imtys;
do i=1 to &n;
v1=uniform(0);
v2=uniform(0);
output; end;
end;
run;
```

Atsitiktinai sucenzūruojam 25% duomenų:

```
data tolygus2;
set tolygus2;
if (i=4 or i=12 or i=20 or i=28 or i=36 or i=44 or i=52 or i=60 or i=68 or
i=76 or i=84 or i=92 or i=100 or
i=108 or i=116 or i=124 or i=132 or i=140 or i=148 or i=156 or i=164 or i=172
or i=180 or i=188 or i=196 or i=204 or
i=212 or i=220 or i=228 or i=236 or i=244 or i=252 or i=260 or i=268 or i=276
or i=284 or i=292 or i=300 or i=308 or
i=316 or i=324 or i=332 or i=340 or i=348 or i=356 or i=364 or i=372 or i=380
or i=388 or i=396 or i=404 or i=412 or
i=420 or i=428 or i=436 or i=444 or i=452 or i=460 or i=468 or i=476 or i=484
or i=492 or i=500or i=508 or i=516 or
i=524 or i=532 or i=540 or i=548 or i=556 or i=564 or i=572 or i=580 or i=588
or i=596 or i=604 or i=612 or
i=620 or i=628 or i=636 or i=644 or i=652 or i=660 or i=668 or i=676 or i=684
or i=692 or i=700 or i=708 or i=716 or
i=724 or i=732 or i=740 or i=748 or i=756 or i=764 or i=772 or i=780 or i=788
or i=796 or i=804 or i=812 or
i=820 or i=828 or i=836 or i=844 or i=852 or i=860 or i=868 or i=876 or i=884
or i=892 or i=900 or i=908 or i=916 or
i=924 or i=932 or i=940 or i=948 or i=956 or i=964 or i=972 or i=980 or i=988
or i=996) then delta1=0;
else delta1=1;
if (i=1 or i=9 or i=17 or i=25 or i=33 or i=41 or i=49 or i=57 or i=65 or
i=73 or i=81 or i=89 or i=97 or i=105 or
i=113 or i=121 or i=129 or i=137 or i=145 or i=153 or i=161 or i=169 or i=177
or i=185 or i=193 or i=201 or i=209 or i=217 or
i=225 or i=233 or i=241 or i=249 or i=257 or i=265 or i=273 or i=281 or i=289
or i=297 or i=305 or i=313 or i=321 or i=329 or
i=337 or i=345 or i=353 or i=361 or i=369 or i=377 or i=385 or i=393 or i=401
or i=409 or i=417 or i=425 or i=433 or i=441 or
i=449 or i=457 or i=465 or i=473 or i=481 or i=489 or i=497 or i=505 or i=513
or i=521 or i=529 or i=537 or i=545 or i=553 or
i=561 or i=569 or i=577 or i=585 or i=593 or i=601 or i=609 or i=617 or i=625
or i=633 or i=641 or i=649 or i=657 or i=665 or
i=673 or i=681 or i=689 or i=697 or i=705 or i=713 or i=721 or i=729 or i=737
or i=745 or i=753 or i=761 or i=769 or i=777 or
i=785 or i=793 or i=801 or i=809 or i=817 or i=825 or i=833 or i=841 or i=849
or i=857 or i=865 or i=873 or i=881 or i=889 or
```

```
i=897 or i=905 or i=913 or i=921 or i=929 or i=937 or i=945 or i=953 or i=961
or i=969 or i=977 or i=985 or i=993)
then delta2=0;
else delta2=1;
u1=v1;
output;run;

data copula2;
set tolygus2;
u2=(1-u1**(-teta)+(v2**(-teta/(1+teta)))*(u1**(-teta)))**(-1/teta);
output; run;

data  dvimate_copula2; /*generuojam T1 ir T2 pagal eksponentini skirstini*/
set copula2;
keep imtis teta delta1 delta2 T1 T2;
T1=-log(1-u1);
T2=-(1/&niu)*log(1-u2) ;
output; run;

proc sort data=dvimate_copula2 out=copula2;
by imtis;
run;
```

Suprogramuojame kriterijų ir tikriname homogeniškumo hipotezę:

```
proc iml;
use dvimate_copula2;
do w=1 to &imtys;
read all var{T1 T2} where(imtis=w) into x; *print x;
read all var{delta1 delta2} where(imtis=w) into delta; * print delta;
N=nrow(x);
y1=j(N,2,0);
y2=j(N,2,0);
do j=1 to N;
do i=1 to N;
if x[i,1]>=x[j,1] then y1[j,1]=y1[j,1]+1; /*Y1(X1j)*/
if x[i,1]>=x[j,2] then y1[j,2]=y1[j,2]+1; /*Y1(X2j)*/
if x[i,2]>=x[j,1] then y2[j,1]=y2[j,1]+1; /*Y2(X1j)*/
if x[i,2]>=x[j,2] then y2[j,2]=y2[j,2]+1; /*Y2(X2j)*/
end; end;
*print y1 y2;
V1=y2[,1]#delta[,1]/(y1[,1]+y2[,1]);
V2=y1[,2]#delta[,2]/(y1[,2]+y2[,2]);
*print V1 V2;
V=(1/sqrt(N))*(V1[+,]-V2[+,]);
*print V;
C1=y1[,1]#y2[,1]#delta[,1]/((y1[,1]+y2[,1])#(y1[,1]+y2[,1]));
C2=y1[,2]#y2[,2]#delta[,2]/((y1[,2]+y2[,2])#(y1[,2]+y2[,2]));
*print C1 C2;
C=(1/N)*(C1[+,]+C2[+,]);
*print C;
R1=y2[,1]#delta[,1]/(y1[,1]+y2[,1]); /*Y2(X1j)*delta1/Y(X1j)*/
R2=y1[,2]#delta[,2]/(y1[,2]+y2[,2]); /*Y1(X2j)*delta2/Y(X2j)*/
*print R1 R2;
r31=j(N,N,0);
r32=j(N,N,0);
r41=j(N,N,0);
r42=j(N,N,0);
do l=1 to N;
do j=1 to N;
if x[j,1]>=x[l,1] then r31[l,j]=1; else r31[l,j]=0;
if x[j,1]>=x[l,2] then r32[l,j]=1; else r32[l,j]=0;
```

```
if x[j,2]>=x[l,1] then r41[l,j]=1; else r41[l,j]=0;
if x[j,2]>=x[l,2] then r42[l,j]=1; else r42[l,j]=0;
end; end;
*print r31 r32 r41 r42;
R3_1=delta[,1]#y2[,1]#r31[,]/((y1[,1]+y2[,1])#(y1[,1]+y2[,1]));
R3_2=delta[,2]#y2[,2]#r32[,]/((y1[,2]+y2[,2])#(y1[,2]+y2[,2]));
R3=R3_1[+,]+R3_2[+,];
*print R3_1 R3_2 R3;
R4_1=delta[,1]#y1[,1]#r41[,]/((y1[,1]+y2[,1])#(y1[,1]+y2[,1]));
R4_2=delta[,2]#y1[,2]#r42[,]/((y1[,2]+y2[,2])#(y1[,2]+y2[,2]));
R4=R4_1[+,]+R4_2[+,];
*print R4_1 R4_2 R4;
R=(R1#R2-R1#R4`-R2#R3`+R3`#R4`);
C12=(1/N)*R[+,];
*print C12;
sigma=sqrt(C-2*C12);*print sigma;
Z_n=V/sigma; print Z_n;
krit_r=PROBIT(1-0.05/2); /* hipoteze atmetama su alpha, jeigu
|Z_n|>z(alpha/2) */
*print krit_r; end;
if abs(Z_n)>krit_r then print 'hipoteze su r.l. alpha=0.5 atmetama';
else print 'hipoteze neatmetama';
quit;
```

Skaičiuojame kriterijaus galią:

```
data galia;
set statistika end=galas;
krit_r=PROBIT(1-0.1/2);
if    (abs(Z_n)>krit_r) then atmete=1;
else atmete=0;
suma+(atmete/&imtys);
if (galas) then put suma=;
run;
```

Konkretaus pavyzdžio atveju, aprašyto 2.2 skyrelyje, turime tokią SAS programą:

```
data pavyzdys;
input x1 delta1 x2 delta2 @@;
datalines;
6       1       140     1
10      1       272     1
12      0       12      0
16      1       568     1
16      1       62      1
19      1       470     0
20      1       523     1
27      1       356     1
27      1       291     1
52      1       631     1
57      1       330     1
70      1       56      1
89      1       398     1
117     1       47      1
178     1       271     1
204     1       406     1
217     1       417     1
218     1       635     1
222     1       169     1
249     1       423     1
305     1       624     1
328     1       270     1
421     1       506     1
423     1       104     1
464     1       6       1
500     1       130     1
546     1       1871    1
555     1       10      1
566     0       566     0
680     1       414     1
685     1       1219    1
707     1       1516    1
764     1       206     1
863     1       267     1
1012    1       249     1
1066    1       1376    1
1077    1       450     1
1089    1       2168    1
1107    1       71      1
1142    1       93      1
1169    1       652     1
1213    1       2128    1
1222    1       196     1
1244    1       821     1
1329    1       99      1
1486    1       120     1
1682    0       152     1
1734    1       1516    1
1746    1       872     1
1776    1       2023    1
1888    1       541     1
1978    1       168     1
2021    1       2161    1
2317    1       752     1
2391    1       781     1
2470    1       1518    1
2621    1       1716    1
2699    1       1169    1
```

```
3496   1      2613   1
4146   1      1355   1
;run;

proc iml;
use pavyzdys;
read all var{x1 x2} into x;
read all var{delta1 delta2} into delta;
N=nrow(x);
y1=j(N,2,0);
y2=j(N,2,0);
do j=1 to N;
do i=1 to N;
if x[i,1]>=x[j,1] then y1[j,1]=y1[j,1]+1; /*Y1(X1j)*/
if x[i,1]>=x[j,2] then y1[j,2]=y1[j,2]+1; /*Y1(X2j)*/
if x[i,2]>=x[j,1] then y2[j,1]=y2[j,1]+1; /*Y2(X1j)*/
if x[i,2]>=x[j,2] then y2[j,2]=y2[j,2]+1; /*Y2(X2j)*/
end; end;
print y1 y2;
V1=y2[,1]#delta[,1]/(y1[,1]+y2[,1]);
V2=y1[,2]#delta[,2]/(y1[,2]+y2[,2]); print V1 V2;
V=(1/sqrt(N))*(V1[+,]-V2[+,]);
print V;
C1=y1[,1]#y2[,1]#delta[,1]/((y1[,1]+y2[,1])#(y1[,1]+y2[,1]));
C2=y1[,2]#y2[,2]#delta[,2]/((y1[,2]+y2[,2])#(y1[,2]+y2[,2]));
print C1 C2;
C=(1/N)*(C1[+,]+C2[+,]);
print C;
R1=y2[,1]#delta[,1]/(y1[,1]+y2[,1]); /*Y2(X1j)*delta1/Y(X1j)*/
R2=y1[,2]#delta[,2]/(y1[,2]+y2[,2]); /*Y1(X2j)*delta2/Y(X2j)*/
print R1 R2;
r31=j(N,N,0);
r32=j(N,N,0);
r41=j(N,N,0);
r42=j(N,N,0);
do l=1 to N;
do j=1 to N;
if x[j,1]>=x[l,1] then r31[l,j]=1; else r31[l,j]=0;
if x[j,1]>=x[l,2] then r32[l,j]=1; else r32[l,j]=0;
if x[j,2]>=x[l,1] then r41[l,j]=1; else r41[l,j]=0;
if x[j,2]>=x[l,2] then r42[l,j]=1; else r42[l,j]=0;
end; end;
print r31 r32 r41 r42;
R3_1=delta[,1]#y2[,1]#r31[,]/((y1[,1]+y2[,1])#(y1[,1]+y2[,1]));
R3_2=delta[,2]#y2[,2]#r32[,]/((y1[,2]+y2[,2])#(y1[,2]+y2[,2]));
R3=R3_1[+,]+R3_2[+,];
print R3_1 R3_2 R3;
R4_1=delta[,1]#y1[,1]#r41[,]/((y1[,1]+y2[,1])#(y1[,1]+y2[,1]));
R4_2=delta[,2]#y1[,2]#r42[,]/((y1[,2]+y2[,2])#(y1[,2]+y2[,2]));
R4=R4_1[+,]+R4_2[+,];
print R4_1 R4_2 R4;
R=(R1#R2-R1#R4`-R2#R3`+R3`#R4`);
C12=(1/N)*R[+,]; print C12;
sigma=sqrt(C-2*C12);print sigma;
Z_n=V/sigma; print Z_n;
 p=probnorm(Z_n);print p;
p_value=cdf('NORMAL',0.05, Z_n );  print p_value; /*jei P-reikšme < alpha
(reikšmingumo lygmuo), tai hipoteze atmetame; */
krit_r=PROBIT(1-0.05/2);  /*p-asis kvantilis stand. normalaus*/
print krit_r;
if abs(Z_n)>krit_r then print 'hipoteze su r.l. alpha=0.5 atmetama'; /*
hipoteze atmetama su alpha, jeigu |Z_n|>z(alpha/2) */
else print 'hipoteze neatmetama';
quit;
```

Output rezultatas:

```
sigma
0.5291756

Z_n
-2.287886

krit_r
1.959964

hipoteze su r.l. alpha=0.5 atmetama
```

Kleitono kopulos simuliavimas trimačiu atveju:

```
title 'n=100, alpha=0.5,skirstiniai nesutampa, trimate copula';
%let n=100;
%let niu=0.1;
data tolygus; /*generuojam tolygu skirstini intervale (0;1)*/
alpha=0.5;
delta1=1;
delta2=1;
delta3=1;
do i=1 to &n;
v1=uniform(0);
v2=uniform(0);
v3=uniform(0);
output; end; run;

data tolygus;
set tolygus;
u1=v1;
u2=v2;
output;run;

data copula;
set tolygus;
keep alpha delta1 delta2 delta3 u1 u2 u3;
a=u1**(-alpha);
b=u2**(-alpha);
c=v3**(-alpha/(1+2*alpha));
/*u3=(2-a-b+c*(a+b-1))**(-1/alpha);*/
u3=((c-1)*(a+b-1)+1)**(-1/alpha); /*destytojo isvedimas*/
output; run;

data  trimate_copula; /*generuojam T1 ir T2 pagal eksponentini skirstini*/
set copula;
T1=-log(1-u1);
T2=-(1/&niu)*log(1-u2) ; /*T2=-(1/&niu)*log(1-u2) ;*/
T3=-log(1-u3) ;
output; run;

proc iml;
use trimate_copula;
read all var{T1 T2 T3} into x;
read all var{delta1 delta2 delta3} into delta;
N=nrow(x);
y1=j(N,3,0);
y2=j(N,3,0);
y3=j(N,3,0);
do j=1 to N;
do i=1 to N;
if x[i,1]>=x[j,1] then y1[j,1]=y1[j,1]+1; /*Y1(X1j)*/
if x[i,1]>=x[j,2] then y1[j,2]=y1[j,2]+1; /*Y1(X2j)*/
if x[i,1]>=x[j,3] then y1[j,3]=y1[j,3]+1; /*Y1(X3j)*/
if x[i,2]>=x[j,1] then y2[j,1]=y2[j,1]+1; /*Y2(X1j)*/
if x[i,2]>=x[j,2] then y2[j,2]=y2[j,2]+1; /*Y2(X2j)*/
if x[i,2]>=x[j,3] then y2[j,3]=y2[j,3]+1; /*Y2(X3j)*/
if x[i,3]>=x[j,1] then y3[j,1]=y3[j,1]+1; /*Y3(X1j)*/
if x[i,3]>=x[j,2] then y3[j,2]=y3[j,2]+1; /*Y3(X2j)*/
if x[i,3]>=x[j,3] then y3[j,3]=y3[j,3]+1; /*Y3(X3j)*/
end; end;
*print y1 y2 y3;
a1=delta[,1]#y1[,1]/(y1[,1]+y2[,1]+y3[,1]);
a2=delta[,2]#y1[,2]/(y1[,2]+y2[,2]+y3[,2]);
a3=delta[,3]#y1[,3]/(y1[,3]+y2[,3]+y3[,3]);
```

```
a4=delta[,1];
V1=(1/sqrt(N))*a4[+,]-((1/sqrt(N))*(a1[+,]+a2[+,]+a3[+,]));
b1=delta[,1]#y2[,1]/(y1[,1]+y2[,1]+y3[,1]);
b2=delta[,2]#y2[,2]/(y1[,2]+y2[,2]+y3[,2]);
b3=delta[,3]#y2[,3]/(y1[,3]+y2[,3]+y3[,3]);
b4=delta[,2];
V2=(1/sqrt(N))*b4[+,]-((1/sqrt(N))*(b1[+,]+b2[+,]+b3[+,]));
c1=delta[,1]#y3[,1]/(y1[,1]+y2[,1]+y3[,1]);
c2=delta[,2]#y3[,2]/(y1[,2]+y2[,2]+y3[,2]);
c3=delta[,3]#y3[,3]/(y1[,3]+y2[,3]+y3[,3]);
c4=delta[,3];
V3=(1/sqrt(N))*c4[+,]-((1/sqrt(N))*(c1[+,]+c2[+,]+c3[+,]));
*print V1 V2 V3;
a1=(delta[,1]#y1[,1]/(y1[,1]+y2[,1]+y3[,1]))#(1-
(y1[,1]/(y1[,1]+y2[,1]+y3[,1])));
a2=(delta[,2]#y1[,2]/(y1[,2]+y2[,2]+y3[,2]))#(1-
(y1[,2]/(y1[,2]+y2[,2]+y3[,2])));
a3=(delta[,3]#y1[,3]/(y1[,3]+y2[,3]+y3[,3]))#(1-
(y1[,3]/(y1[,3]+y2[,3]+y3[,3])));
c11=(1/N)*(a1[+,]+a2[+,]+a3[+,]); *print c11;
a12=(delta[,1]#y1[,1]/(y1[,1]+y2[,1]+y3[,1]))#(0-
(y2[,1]/(y1[,1]+y2[,1]+y3[,1])));
a22=(delta[,2]#y1[,2]/(y1[,2]+y2[,2]+y3[,2]))#(0-
(y2[,2]/(y1[,2]+y2[,2]+y3[,2])));
a32=(delta[,3]#y1[,3]/(y1[,3]+y2[,3]+y3[,3]))#(0-
(y2[,3]/(y1[,3]+y2[,3]+y3[,3])));
c12=(1/N)*(a12[+,]+a22[+,]+a32[+,]);
a13=(delta[,1]#y1[,1]/(y1[,1]+y2[,1]+y3[,1]))#(0-
(y3[,1]/(y1[,1]+y2[,1]+y3[,1])));
a23=(delta[,2]#y1[,2]/(y1[,2]+y2[,2]+y3[,2]))#(0-
(y3[,2]/(y1[,2]+y2[,2]+y3[,2])));
a33=(delta[,3]#y1[,3]/(y1[,3]+y2[,3]+y3[,3]))#(0-
(y3[,3]/(y1[,3]+y2[,3]+y3[,3])));
c13=(1/N)*(a1[+,]+a2[+,]+a3[+,]);
b1=(delta[,1]#y2[,1]/(y1[,1]+y2[,1]+y3[,1]))#(0-
(y1[,1]/(y1[,1]+y2[,1]+y3[,1])));
b2=(delta[,2]#y2[,2]/(y1[,2]+y2[,2]+y3[,2]))#(0-
(y1[,2]/(y1[,2]+y2[,2]+y3[,2])));
b3=(delta[,3]#y2[,3]/(y1[,3]+y2[,3]+y3[,3]))#(0-
(y1[,3]/(y1[,3]+y2[,3]+y3[,3])));
c21=(1/N)*(b1[+,]+b2[+,]+b3[+,]);
b12=(delta[,1]#y2[,1]/(y1[,1]+y2[,1]+y3[,1]))#(1-
(y2[,1]/(y1[,1]+y2[,1]+y3[,1])));
b22=(delta[,2]#y2[,2]/(y1[,2]+y2[,2]+y3[,2]))#(1-
(y2[,2]/(y1[,2]+y2[,2]+y3[,2])));
b32=(delta[,3]#y2[,3]/(y1[,3]+y2[,3]+y3[,3]))#(1-
(y2[,3]/(y1[,3]+y2[,3]+y3[,3])));
c22=(1/N)*(b12[+,]+b22[+,]+b32[+,]);
b13=(delta[,1]#y2[,1]/(y1[,1]+y2[,1]+y3[,1]))#(0-
(y3[,1]/(y1[,1]+y2[,1]+y3[,1])));
b23=(delta[,2]#y2[,2]/(y1[,2]+y2[,2]+y3[,2]))#(0-
(y3[,2]/(y1[,2]+y2[,2]+y3[,2])));
b33=(delta[,3]#y2[,3]/(y1[,3]+y2[,3]+y3[,3]))#(0-
(y3[,3]/(y1[,3]+y2[,3]+y3[,3])));
c23=(1/N)*(b13[+,]+b23[+,]+b33[+,]);
d1=(delta[,1]#y3[,1]/(y1[,1]+y2[,1]+y3[,1]))#(0-
(y1[,1]/(y1[,1]+y2[,1]+y3[,1])));
d2=(delta[,2]#y3[,2]/(y1[,2]+y2[,2]+y3[,2]))#(0-
(y1[,2]/(y1[,2]+y2[,2]+y3[,2])));
d3=(delta[,3]#y3[,3]/(y1[,3]+y2[,3]+y3[,3]))#(0-
(y1[,3]/(y1[,3]+y2[,3]+y3[,3])));
c31=(1/N)*(d1[+,]+d2[+,]+d3[+,]);
d12=(delta[,1]#y3[,1]/(y1[,1]+y2[,1]+y3[,1]))#(0-
(y2[,1]/(y1[,1]+y2[,1]+y3[,1])));
```

```
d22=(delta[,2]#y3[,2]/(y1[,2]+y2[,2]+y3[,2]))#(0-
(y2[,2]/(y1[,2]+y2[,2]+y3[,2])));
d32=(delta[,3]#y3[,3]/(y1[,3]+y2[,3]+y3[,3]))#(0-
(y2[,3]/(y1[,3]+y2[,3]+y3[,3])));
c32=(1/N)*(d12[+,]+d22[+,]+d32[+,]);
d13=(delta[,1]#y3[,1]/(y1[,1]+y2[,1]+y3[,1]))#(1-
(y3[,1]/(y1[,1]+y2[,1]+y3[,1])));
d23=(delta[,2]#y3[,2]/(y1[,2]+y2[,2]+y3[,2]))#(1-
(y3[,2]/(y1[,2]+y2[,2]+y3[,2])));
d33=(delta[,3]#y3[,3]/(y1[,3]+y2[,3]+y3[,3]))#(1-
(y3[,3]/(y1[,3]+y2[,3]+y3[,3])));
c33=(1/N)*(d13[+,]+d23[+,]+d33[+,]);
*print c11 c12 c13 c21 c22 c23 c31 c32 c33;
r111=delta[,1]#(1-y1[,1]/(y1[,1]+y2[,1]+y3[,1]));
r121=delta[,1]#(0-y2[,1]/(y1[,1]+y2[,1]+y3[,1]));
r131=delta[,1]#(0-y3[,1]/(y1[,1]+y2[,1]+y3[,1]));
r112=delta[,2]#(0-y1[,2]/(y1[,2]+y2[,2]+y3[,2]));
r122=delta[,2]#(1-y2[,2]/(y1[,2]+y2[,2]+y3[,2]));
r132=delta[,2]#(0-y3[,2]/(y1[,2]+y2[,2]+y3[,2]));
r113=delta[,3]#(0-y1[,3]/(y1[,3]+y2[,3]+y3[,3]));
r123=delta[,3]#(0-y2[,3]/(y1[,3]+y2[,3]+y3[,3]));
r133=delta[,3]#(1-y3[,3]/(y1[,3]+y2[,3]+y3[,3]));
*print r111 r121 r131 r112 r122 r132 r113 r123 r133;
r1=j(N,N,0);
r2=j(N,N,0);
r3=j(N,N,0);
do q=1 to N;
do r=1 to N;
if x[r,1]>=x[q,1] then r1[q,r]=1; else r1[q,r]=0; /*Y1(X1j)*/
if x[r,1]>=x[q,2] then r1[q,r]=1; else r1[q,r]=0;/*Y1(X2j)*/
if x[r,1]>=x[q,3] then r1[q,r]=1; else r1[q,r]=0;/*Y1(X3j)*/
if x[r,2]>=x[q,1] then r2[q,r]=1; else r2[q,r]=0;/*Y2(X1j)*/
if x[r,2]>=x[q,2] then r2[q,r]=1; else r2[q,r]=0;/*Y2(X2j)*/
if x[r,2]>=x[q,3] then r2[q,r]=1; else r2[q,r]=0;/*Y2(X3j)*/
if x[r,3]>=x[q,1] then r3[q,r]=1; else r3[q,r]=0;/*Y3(X1j)*/
if x[r,3]>=x[q,2] then r3[q,r]=1; else r3[q,r]=0;/*Y3(X2j)*/
if x[r,3]>=x[q,3] then r3[q,r]=1; else r3[q,r]=0;/*Y3(X3j)*/
end; end;
*print r1 r2 r3;
f11=delta[,1]#(1-y1[,1]/(y1[,1]+y2[,1]+y3[,1])#r1[,]/(y1[,1]+y2[,1]+y3[,1]));
/*i=1, l=1*/
f21=delta[,2]#(1-y1[,2]/(y1[,2]+y2[,2]+y3[,2])#r1[,]/(y1[,2]+y2[,2]+y3[,2]));
f31=delta[,3]#(1-y1[,3]/(y1[,3]+y2[,3]+y3[,3])#r1[,]/(y1[,3]+y2[,3]+y3[,3]));
r211=f11[+,]+f21[+,]+f31[+,];
f12=delta[,1]#(0-y1[,1]/(y1[,1]+y2[,1]+y3[,1])#r2[,]/(y1[,1]+y2[,1]+y3[,1]));
/*i=1, l=2*/
f22=delta[,2]#(0-y1[,2]/(y1[,2]+y2[,2]+y3[,2])#r2[,]/(y1[,2]+y2[,2]+y3[,2]));
f32=delta[,3]#(0-y1[,3]/(y1[,3]+y2[,3]+y3[,3])#r2[,]/(y1[,3]+y2[,3]+y3[,3]));
r212=f12[+,]+f22[+,]+f32[+,];
f13=delta[,1]#(0-y1[,1]/(y1[,1]+y2[,1]+y3[,1])#r3[,]/(y1[,1]+y2[,1]+y3[,1]));
/*i=1, l=3*/
f23=delta[,2]#(0-y1[,2]/(y1[,2]+y2[,2]+y3[,2])#r3[,]/(y1[,2]+y2[,2]+y3[,2]));
f33=delta[,3]#(0-y1[,3]/(y1[,3]+y2[,3]+y3[,3])#r3[,]/(y1[,3]+y2[,3]+y3[,3]));
r213=f13[+,]+f23[+,]+f33[+,];
g11=delta[,1]#(0-y2[,1]/(y1[,1]+y2[,1]+y3[,1])#r1[,]/(y1[,1]+y2[,1]+y3[,1]));
/*i=2, l=1*/
g21=delta[,2]#(0-y2[,2]/(y1[,2]+y2[,2]+y3[,2])#r1[,]/(y1[,2]+y2[,2]+y3[,2]));
g31=delta[,3]#(0-y2[,3]/(y1[,3]+y2[,3]+y3[,3])#r1[,]/(y1[,3]+y2[,3]+y3[,3]));
r221=g11[+,]+g21[+,]+g31[+,];
g12=delta[,1]#(1-y2[,1]/(y1[,1]+y2[,1]+y3[,1])#r2[,]/(y1[,1]+y2[,1]+y3[,1]));
/*i=2, l=2*/
g22=delta[,2]#(1-y2[,2]/(y1[,2]+y2[,2]+y3[,2])#r2[,]/(y1[,2]+y2[,2]+y3[,2]));
g32=delta[,3]#(1-y2[,3]/(y1[,3]+y2[,3]+y3[,3])#r2[,]/(y1[,3]+y2[,3]+y3[,3]));
r222=g12[+,]+g22[+,]+g32[+,];
```

```
g13=delta[,1]#(0-y2[,1]/(y1[,1]+y2[,1]+y3[,1])#r3[,]/(y1[,1]+y2[,1]+y3[,1]));
/*i=2, l=3*/
g23=delta[,2]#(0-y2[,2]/(y1[,2]+y2[,2]+y3[,2])#r3[,]/(y1[,2]+y2[,2]+y3[,2]));
g33=delta[,3]#(0-y2[,3]/(y1[,3]+y2[,3]+y3[,3])#r3[,]/(y1[,3]+y2[,3]+y3[,3]));
r223=g13[+,]+g23[+,]+g33[+,];
h11=delta[,1]#(0-y3[,1]/(y1[,1]+y2[,1]+y3[,1])#r1[,]/(y1[,1]+y2[,1]+y3[,1]));
/*i=3, l=1*/
h21=delta[,2]#(0-y3[,2]/(y1[,2]+y2[,2]+y3[,2])#r1[,]/(y1[,2]+y2[,2]+y3[,2]));
h31=delta[,3]#(0-y3[,3]/(y1[,3]+y2[,3]+y3[,3])#r1[,]/(y1[,3]+y2[,3]+y3[,3]));
r231=h11[+,]+h21[+,]+h31[+,];
h12=delta[,1]#(0-y3[,1]/(y1[,1]+y2[,1]+y3[,1])#r2[,]/(y1[,1]+y2[,1]+y3[,1]));
/*i=3, l=2*/
h22=delta[,2]#(0-y3[,2]/(y1[,2]+y2[,2]+y3[,2])#r2[,]/(y1[,2]+y2[,2]+y3[,2]));
h32=delta[,3]#(0-y3[,3]/(y1[,3]+y2[,3]+y3[,3])#r2[,]/(y1[,3]+y2[,3]+y3[,3]));
r232=h12[+,]+h22[+,]+h32[+,];
h13=delta[,1]#(1-y3[,1]/(y1[,1]+y2[,1]+y3[,1])#r3[,]/(y1[,1]+y2[,1]+y3[,1]));
/*i=3, l=3*/
h23=delta[,2]#(1-y3[,2]/(y1[,2]+y2[,2]+y3[,2])#r3[,]/(y1[,2]+y2[,2]+y3[,2]));
h33=delta[,3]#(1-y3[,3]/(y1[,3]+y2[,3]+y3[,3])#r3[,]/(y1[,3]+y2[,3]+y3[,3]));
r233=h13[+,]+h23[+,]+h33[+,];
*print r211 r212 r213 r221 r222 r223 r231 r232 r233;
c11_12a=r111#r112-r111#r212`-r112#r211`+r211`#r212`; /*i=1, j=1, l=1, l`=2*/
c11_12=(1/N)*c11_12a[+,];
c11_13a=r111#r113-r111#r213`-r113#r211`+r211`#r213`; /*i=1, j=1, l=1, l`=3*/
c11_13=(1/N)*c11_13a[+,];
c11_21a=r112#r111-r112#r211`-r111#r212`+r212`#r211`; /*i=1, j=1, l=2, l`=1*/
c11_21=(1/N)*c11_21a[+,];
c11_23a=r112#r113-r112#r213`-r113#r212`+r212`#r213`; /*i=1, j=1, l=2, l`=3*/
c11_23=(1/N)*c11_23a[+,];
c11_31a=r113#r111-r113#r211`-r111#r213`+r213`#r211`; /*i=1, j=1, l=3, l`=1*/
c11_31=(1/N)*c11_31a[+,];
c11_32a=r113#r112-r113#r212`-r112#r213`+r213`#r212`; /*i=1, j=1, l=3, l`=2*/
c11_32=(1/N)*c11_32a[+,];
c12_12a=r111#r122-r111#r222`-r122#r211`+r211`#r222`; /*i=1, j=2, l=1, l`=2*/
c12_12=(1/N)*c12_12a[+,];
c12_13a=r111#r123-r111#r223`-r123#r211`+r211`#r223`; /*i=1, j=2, l=1, l`=3*/
c12_13=(1/N)*c12_13a[+,];
c12_21a=r112#r121-r112#r221`-r121#r212`+r212`#r221`; /*i=1, j=2, l=2, l`=1*/
c12_21=(1/N)*c12_21a[+,];
c12_23a=r112#r123-r112#r223`-r123#r212`+r212`#r223`; /*i=1, j=2, l=2, l`=3*/
c12_23=(1/N)*c12_23a[+,];
c12_31a=r113#r121-r113#r221`-r121#r213`+r213`#r221`; /*i=1, j=2, l=3, l`=1*/
c12_31=(1/N)*c12_31a[+,];
c12_32a=r113#r122-r113#r222`-r122#r213`+r213`#r222`; /*i=1, j=2, l=3, l`=2*/
c12_32=(1/N)*c12_32a[+,];
c13_12a=r111#r132-r111#r232`-r132#r211`+r211`#r232`; /*i=1, j=3, l=1, l`=2*/
c13_12=(1/N)*c13_12a[+,];
c13_13a=r111#r133-r111#r233`-r133#r211`+r211`#r233`; /*i=1, j=3, l=1, l`=3*/
c13_13=(1/N)*c13_13a[+,];
c13_21a=r112#r131-r112#r231`-r131#r212`+r212`#r231`; /*i=1, j=3, l=2, l`=1*/
c13_21=(1/N)*c13_21a[+,];
c13_23a=r112#r133-r112#r233`-r133#r212`+r212`#r233`; /*i=1, j=3, l=2, l`=3*/
c13_23=(1/N)*c13_23a[+,];
c13_31a=r113#r131-r113#r231`-r131#r213`+r213`#r231`; /*i=1, j=3, l=3, l`=1*/
c13_31=(1/N)*c13_31a[+,];
c13_32a=r113#r132-r113#r232`-r132#r213`+r213`#r232`; /*i=1, j=3, l=3, l`=2*/
c13_32=(1/N)*c13_32a[+,];
c21_12a=r121#r112-r121#r212`-r112#r221`+r221`#r212`; /*i=2, j=1, l=1, l`=2*/
c21_12=(1/N)*c21_12a[+,];
c21_13a=r121#r113-r121#r213`-r113#r221`+r221`#r213`; /*i=2, j=1, l=1, l`=3*/
c21_13=(1/N)*c21_13a[+,];
c21_21a=r122#r111-r122#r211`-r111#r222`+r222`#r211`; /*i=2, j=1, l=2, l`=1*/
c21_21=(1/N)*c21_21a[+,];
c21_23a=r122#r113-r122#r213`-r113#r222`+r222`#r213`; /*i=2, j=1, l=2, l`=3*/
```

```
c21_23=(1/N)*c21_23a[+,];
c21_31a=r123#r111-r123#r211`-r111#r223`+r223`#r211`; /*i=2, j=1, l=3, l`=1*/
c21_31=(1/N)*c21_31a[+,];
c21_32a=r123#r112-r123#r212`-r112#r223`+r223`#r212`; /*i=2, j=1, l=3, l`=2*/
c21_32=(1/N)*c21_32a[+,];
c22_12a=r121#r122-r121#r222`-r122#r221`+r221`#r222`; /*i=2, j=2, l=1, l`=2*/
c22_12=(1/N)*c22_12a[+,];
c22_13a=r121#r123-r121#r223`-r123#r221`+r221`#r223`; /*i=2, j=2, l=1, l`=3*/
c22_13=(1/N)*c22_13a[+,];
c22_21a=r122#r121-r122#r221`-r121#r222`+r222`#r221`; /*i=2, j=2, l=2, l`=1*/
c22_21=(1/N)*c22_21a[+,];
c22_23a=r122#r123-r122#r223`-r123#r222`+r222`#r223`; /*i=2, j=2, l=2, l`=3*/
c22_23=(1/N)*c22_23a[+,];
c22_31a=r123#r121-r123#r221`-r121#r223`+r223`#r221`; /*i=2, j=2, l=3, l`=1*/
c22_31=(1/N)*c22_31a[+,];
c22_32a=r123#r122-r123#r222`-r122#r223`+r223`#r222`; /*i=2, j=2, l=3, l`=2*/
c22_32=(1/N)*c22_32a[+,];
c23_12a=r121#r132-r121#r232`-r132#r221`+r221`#r232`; /*i=2, j=3, l=1, l`=2*/
c23_12=(1/N)*c23_12a[+,];
c23_13a=r121#r133-r121#r233`-r133#r221`+r221`#r233`; /*i=2, j=3, l=1, l`=3*/
c23_13=(1/N)*c23_13a[+,];
c23_21a=r122#r131-r122#r231`-r131#r222`+r222`#r231`; /*i=2, j=3, l=2, l`=1*/
c23_21=(1/N)*c23_21a[+,];
c23_23a=r122#r133-r122#r233`-r133#r222`+r222`#r233`; /*i=2, j=3, l=2, l`=3*/
c23_23=(1/N)*c23_23a[+,];
c23_31a=r123#r131-r123#r231`-r131#r223`+r223`#r231`; /*i=2, j=3, l=3, l`=1*/
c23_31=(1/N)*c23_31a[+,];
c23_32a=r123#r132-r123#r232`-r132#r223`+r223`#r232`; /*i=2, j=3, l=3, l`=2*/
c23_32=(1/N)*c23_32a[+,];
c31_12a=r131#r112-r131#r212`-r112#r231`+r231`#r212`; /*i=3, j=1, l=1, l`=2*/
c31_12=(1/N)*c31_12a[+,];
c31_13a=r131#r113-r131#r213`-r113#r231`+r231`#r213`; /*i=3, j=1, l=1, l`=3*/
c31_13=(1/N)*c31_13a[+,];
c31_21a=r132#r111-r132#r211`-r111#r232`+r232`#r211`; /*i=3, j=1, l=2, l`=1*/
c31_21=(1/N)*c31_21a[+,];
c31_23a=r132#r113-r132#r213`-r113#r232`+r232`#r213`; /*i=3, j=1, l=2, l`=3*/
c31_23=(1/N)*c31_23a[+,];
c31_31a=r133#r111-r133#r211`-r111#r233`+r233`#r211`; /*i=3, j=1, l=3, l`=1*/
c31_31=(1/N)*c31_31a[+,];
c31_32a=r133#r112-r133#r212`-r112#r233`+r233`#r212`; /*i=3, j=1, l=3, l`=2*/
c31_32=(1/N)*c31_32a[+,];
c32_12a=r131#r122-r131#r222`-r122#r231`+r231`#r222`; /*i=3, j=2, l=1, l`=2*/
c32_12=(1/N)*c32_12a[+,];
c32_13a=r131#r123-r131#r223`-r123#r231`+r231`#r223`; /*i=3, j=2, l=1, l`=3*/
c32_13=(1/N)*c32_13a[+,];
c32_21a=r132#r121-r132#r221`-r121#r232`+r232`#r221`; /*i=3, j=2, l=2, l`=1*/
c32_21=(1/N)*c32_21a[+,];
c32_23a=r132#r123-r132#r223`-r123#r232`+r232`#r223`; /*i=3, j=2, l=2, l`=3*/
c32_23=(1/N)*c32_23a[+,];
c32_31a=r133#r121-r133#r221`-r121#r233`+r233`#r221`; /*i=3, j=2, l=3, l`=1*/
c32_31=(1/N)*c32_31a[+,];
c32_32a=r133#r122-r133#r222`-r122#r233`+r233`#r222`; /*i=3, j=2, l=3, l`=2*/
c32_32=(1/N)*c32_32a[+,];
c33_12a=r131#r132-r131#r232`-r132#r231`+r231`#r232`; /*i=3, j=3, l=1, l`=2*/
c33_12=(1/N)*c33_12a[+,];
c33_13a=r131#r133-r131#r233`-r133#r231`+r231`#r233`; /*i=3, j=3, l=1, l`=3*/
c33_13=(1/N)*c33_13a[+,];
c33_21a=r132#r131-r132#r231`-r131#r232`+r232`#r231`; /*i=3, j=3, l=2, l`=1*/
c33_21=(1/N)*c33_21a[+,];
c33_23a=r132#r133-r132#r233`-r133#r232`+r232`#r233`; /*i=3, j=3, l=2, l`=3*/
c33_23=(1/N)*c33_23a[+,];
c33_31a=r133#r131-r133#r231`-r131#r233`+r233`#r231`; /*i=3, j=3, l=3, l`=1*/
c33_31=(1/N)*c33_31a[+,];
c33_32a=r133#r132-r133#r232`-r132#r233`+r233`#r232`; /*i=3, j=3, l=3, l`=2*/
```

```
c33_32=(1/N)*c33_32a[+,];

sigma11=c11+(c11_12[+,]+c11_13[+,]+c11_21[+,]+c11_23[+,]+c11_31[+,]+c11_32[+,
]);* print sigma11;
sigma12=c12+(c12_12[+,]+c12_13[+,]+c12_21[+,]+c12_23[+,]+c12_31[+,]+c12_32[+,
]); *print sigma12;
sigma13=c13+(c13_12[+,]+c13_13[+,]+c13_21[+,]+c13_23[+,]+c13_31[+,]+c13_32[+,
]);* print sigma13;
sigma21=c21+(c21_12[+,]+c21_13[+,]+c21_21[+,]+c21_23[+,]+c21_31[+,]+c21_32[+,
]); *print sigma21;
sigma22=c22+(c22_12[+,]+c22_13[+,]+c22_21[+,]+c22_23[+,]+c22_31[+,]+c22_32[+,
]);* print sigma22;
sigma23=c23+(c23_12[+,]+c23_13[+,]+c23_21[+,]+c23_23[+,]+c23_31[+,]+c23_32[+,
]); *print sigma23;
sigma31=c31+(c31_12[+,]+c31_13[+,]+c31_21[+,]+c31_23[+,]+c31_31[+,]+c31_32[+,
]);* print sigma31;
sigma32=c32+(c32_12[+,]+c32_13[+,]+c32_21[+,]+c32_23[+,]+c32_31[+,]+c32_32[+,
]); *print sigma32;
sigma33=c33+(c33_12[+,]+c33_13[+,]+c33_21[+,]+c33_23[+,]+c33_31[+,]+c33_32[+,
]); *print sigma33;
create viskas var {sigma11 sigma12 sigma13 sigma21 sigma22 sigma23 sigma31
sigma32 sigma33 V1 V2 V3};
append;
quit;

proc iml;
use viskas;
read all var{sigma11 sigma12 sigma13 sigma21 sigma22 sigma23 sigma31 sigma32
sigma33} into sigma_prad; print sigma_prad;
E1=j(1,3,0);
E1=sigma_prad[{1},{1,2,3}]; *print E1;
E2=j(1,3,0);
E2=sigma_prad[{1},{4,5,6}]; *print E2;
E3=j(1,3,0);
E3=sigma_prad[{1},{7,8,9}]; *print E3;
create opa var {E1 E2 E3};
append;
quit;

proc transpose data=opa out=transponuota;
var E1 E2 E3;
run;

%let k=3;
proc iml;
use transponuota;
read all var{COL1 COL2 COL3} into sigma; print sigma;
use viskas;
read all var{V1 V2 V3} into V; print V;
sigma_atv=inv(sigma); print sigma_atv;
Y_n_kvadr=V*sigma_atv*V`; print Y_n_kvadr;
/*hipoteze atmetama, jei Y_n_kvadr>chi_kvadr(k-1), pas mus k=3*/
krit_r=cinv(0.05,&k-1);print krit_r;
if Y_n_kvadr>krit_r then print 'hipoteze su r.l. alpha=0.5 atmetama';
else print 'hipoteze neatmetama';
quit;
```