UNIVERSITETI I TIRANËS
FAKULTETI I SHKENCAVE TË NATYRËS
DEPARTAMENTI I MATEMATIKËS SË APLIKUAR

PROGRAMI

Master i shkencave në Inxhinieri Matematike dhe Informatike

TEZË DIPLOME

ZBULIMI I SEKUENCAVE TË PËRSËRITURA NË SERI KOHORE

Punoi :
Glen Rapaj

Udheheqes Shkencor :
Prof. Asoc. Eralda Gjika

TIRANË, Korrik 2024

Motifs Discovery in Time Series, Application

Abstrakti

Në këtë temë diplome, do të studiohen disa nga teknikat themelore për analizimin e serive kohore, më konkretisht duke zbuluar nënvargje (nën sekunca) që përsëriten ne seri kohore me periodicitet apo sesonalitet

Në kapitullin e pare prezantohen cilësitë e serive kohore si dhe disa nga distancat më të njohura. Së bashku me distancat klasike si distancat e Euklid dhe Manhattan, ne do të shpjegojmë dhe madhësinë e ngjashmërisë që ndryshojnë nga distancat tradicionale
për shkak se nuk kenaqin të treja kushtet për të qënë distanca. Këto gjejnë përdorim të gjerë në aplikacione të ndryshme të serive kohore.

Në kapitullin e dytë prezantojmë një algoritmë të ri të krijuar posaçërisht për zbulimin e sekuencave të perseritura brenda të dhënave të serive kohore. Efektiviteti i
tij është vërtetuar përmes aplikimeve të shumta të realizuara jo vetëm në gjuhën programimit R, por edhe në shumë gjuhë të ndryshme
programimi për analizën e serive kohore si pershembull Python.

Eksplorimi i analizës së sekuencave të përsëritura brenda të dhënave të serive kohore mban një rëndësi të thellë në disiplina të ndryshme, duke përfshirë
mjekësinë (p.sh., analizën EKG), parashikimin financiar, modelet e konsumit të energjisë dhe modelimin e klimës, duke nënvizuar kështu rëndësinë ndërdisiplinore të saj.

Fjalët kyçe: seri kohore, distancë, sekuencë, distancë euklidiane, ngjashmëri, gjuhe programimi, R, algoritem.

Motifs Discovery in Time Series, Application

Abstract

In this diploma thesis, some of the fundamental techniques for analyzing time series will be studied, specifically by identifying recurring patterns in time series with periodicity or seasonality.

In the first chapter, the characteristics of time series are presented along with some of the most well-known distances. Alongside classic distances like Euclidean and Manhattan distances, we will also explain similarity measures that differ from traditional distances because they do not satisfy all the conditions to be considered distances. These find widespread use in various applications of time series.

In the second chapter, we introduce a new algorithm created specifically for detecting repeated sequences within time series data. Its effectiveness has been demonstrated through numerous applications carried out not only in the R programming language but also in several different programming languages for time series analysis, such as Python.

The exploration of repeated sequence analysis within time series data holds profound significance across various disciplines, including medicine (e.g., EKG analysis), financial forecasting, energy consumption modeling, and climate modeling, thereby highlighting its interdisciplinary importance.

Key words: time series, distance, sequence, euclidean distance, similarity programming language, **R**, **algorithm.**

Përmbajtje

Hyrje

Fusha e analizës së serive kohore ka përjetuar zhvillime të rëndësishme në dekadat e fundit, duke evoluar nga praktikat e thjeshta të grumbullimit të të dhënave të kaluara në një disiplinë të sofistikuar me aplikime të gjerë në fusha të ndryshme. Fillimisht, setet e të dhënave përfshinin kryesisht tregues ekonomikë të thjeshtë si shitjet vjetore të drithërave dhe borxhi shtetëror, të cilat tani gjenden lehtësisht në internet. Sot, hasim një varg të pasur të të dhënave te serive kohore, që

shkon nga të dhënat financiare si kursi i këmbimit valutor deri te dhënat ekologjike si dendësia e reshjeve gjatë periudhave të caktuara, si dhe të dhëna demografike që kapin ngjarje jetike si lindjet dhe vdekjet. Po ashtu, risitë në teknologjitë mjekësore të avancuara ka sjellë seria kohore komplekse biologjike si sekuencat e ADN-së dhe elektrokardiogramet (EKG) në sferën e analizës së serive kohore, duke zgjeruar gamën dhe rëndësinë e saj.

Këto sete të ndryshme të të dhënave shkaktojnë interes të madh për shkak të potencialit të tyre për modelimin parashikues në vendimmarrje. Për shembull, analistët financiarë përdorin seritë kohore për të parashikuar tendencat e tregut dhe për të marrë vendime të informuara në investime, ndërsa epidemiologët përdorin seritë kohore demografike për të gjurmuar shpërthimet e sëmundjeve dhe për të vlerësuar ndërhyrjet në shëndetin publik. Po ashtu, praktikat mjekësore mbështeten në seritë kohore fiziologjike për të monitoruar gjendjen shëndetësore të pacientëve dhe për të diagnostikuar kushtet bazë, duke theksuar rolin kritik të analizës së serive kohore në kujdesin shëndetësor modern.

Seritë kohore shfaqin një gamë të gjërë të karakteristikave, nga trendet afatgjata deri te fluktuacionet afatshkurtra, secila prej tyre kontribuon në sjelljen dinamike të tyre në kohë. Trendet ofrojnë njohuri në modele të përgjithshme në të dhëna, ndërsa zhurma paraqet variabilitet të rastësishëm që mund të fshehë modele themelore. Ciklet dhe fluktuacionet kapin ndryshime periodike dhe modele të parregullt, ndërsa variacionet sezonale reflektojnë ndryshime sistematike që përsëriten në intervale të rregullta, si ato të vëzhguara në modele klimatike ose sjelljen e konsumatorëve gjatë vitit.

Duke eksploruar si ndryshojnë faktorët ndryshe brenda të dhënave të serive kohore dhe duke vlerësuar masat e ngjashmërisë, kjo studim synon të thellojë kuptimin tonë për fenomene të ndikuar nga koha dhe të përparojë në krijimin e strukturave analitike të besueshme. Duke ekzaminuar ndërveprimin mes metrikave të ndryshme të serive kohore dhe masave të ngjashmërisë, ky kërkim synon të përmirësojë kuptimin tonë për fenomenet që varen nga koha dhe të kontribuojë në zhvillimin e strukturave analitike të qëndrueshme. Përmes analizës teorike dhe demonstrimeve praktike duke përdorur gjuhën e programimit R, ky disertacion synon të përshkruajë konceptet dhe metodologjitë themelore në analizën e serive kohore, duke fuqizuar kështu hulumtuesit dhe praktikantët për të nxjerrë njohuri të rëndësishme nga të dhënat kohore të shumëllojshme.

Kapitulli 1

Përkufizime
Distancat e përdorura në seri kohore

Në këtë kapitull do të njihemi me nocione të rëndësishme, sikurse janë seria kohore dhe me motivet në serine kohore. Gjithashtu, do të njihemi me konceptin e distancës dhe të madhësisë të ngjashmërisë.

1.1. **Përkufizime**

Pothuajse çdo fenomen natyror shkakton interes për të vëzhguar. Këtu mund të përmendim shiun, temperaturën e ajrit, ndriçimin e Hënës, shkallën e tërmeteve, dhe shumë të tjera.

Megjithatë, ka edhe fenomene të tjera që janë tërheqëse për të vëzhguar, si kursi i këmbimit valutor, shitjet ditore në dyqan, dhe të dhënat demografike.

Shpesh, nevoja për krahasimin e të dhënave në periudha të ndryshme lind për të verifikuar se a janë ato të ngjashme ose jo, dhe nëse po, në çfarë masë. Për ta bërë këtë, përdorim disa metoda për të vlerësuar ngjashmërinë. Në kontekstin tonë, të dhënat janë vlera numerike të vëzhgimeve. Kështu, veprimet bazike të kryera mbi to, si mbledhja dhe zbritja, kanë rëndësi. Prandaj, për të përcaktuar nëse të dhënat janë të ngjashme, përdoret koncepti i distancës, që është një vlerë e llogaritur pas kryerjes së veprimeve bazike mbi to, në përputhje me një rregull të caktuar.

Përkufizim 1.1 **Seri kohore** do të quajmë një varg vrojtimesh $T = (T_1, T_2, \ldots, T_n)$, të mbledhura në një kohë të dhënë, në intervale të rregullta.

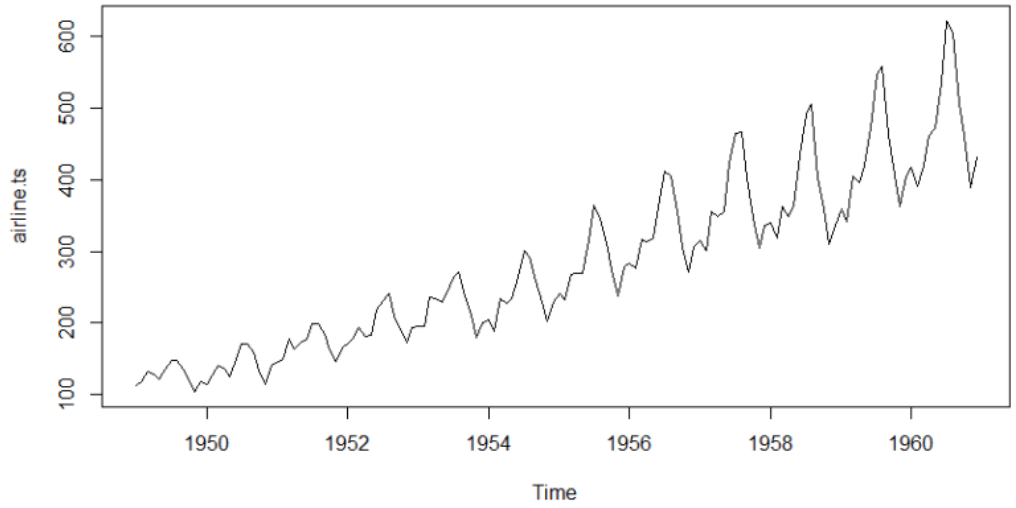


Figura 1.1. Paraqitje grafike e serisë kohore *numrit të pasagjerëve që udhëtojnë me linjat ajrore (Burimi: https://www.researchgate.net/publication/261464969_Autocorrelation_and_partial_autocorrelation_functions_to_improve_neural_networks_models_on_univariate_time_series_forecasting)*

Në Figurën 1.1. jepet një ilustrim i serisë kohore të numrit (në mijë) sipas muajve të pasagjerëve që kanë udhëtuar me linjat ajrore për 12 vite 1949-1960. Të dhënat e grumbulluara paraqiten me vijë të vazhdueshme, për të dhënë idenë se intervalet e kohës kur janë mbledhur të dhënat,

janë të rregullta. Pikat e bashkimit të vijave përfaqësojnë vrojtimet reale.

Një pjesë e rëndësishme e një serie kohore, të paktën nga paraqitja grafike, janë dhe përsëritjet në periudha të caktuara kohe. Për shembull, përsëritjet në periudha stinore, vjetore, etj, në varësi të periodicitetit të të dhënave. Me sy të lirë, mund të dallohen disa përsëritje, që, në gjuhën teknike të serive kohore, njihen me termin *motive ose nënsekuenca me përsëritje (nënvargje të ngjashme).*

Përkufizim 1.2 **Nënsekuencë me përsëritje** në një seri kohore T është një nënseri $M_i = (T_i, T_{i+1}, ..., T_{i+m-1})$ e serisë $T = (T_1, T_2, ..., T_n)$ të dhënë, që gëzon vetinë: Përbëhet nga *m* vrojtime të njëpasnjëshme, dhe përsëritet përgjatë gjithë serisë më shumë se një herë.

Në Figurën 1.2. jepet një paraqitje grafike e pjesëve të serisë kohore më sipër. Vihet re seria origjinale, trendi, sezonaliteti dhe përbërësi i rastësisë.

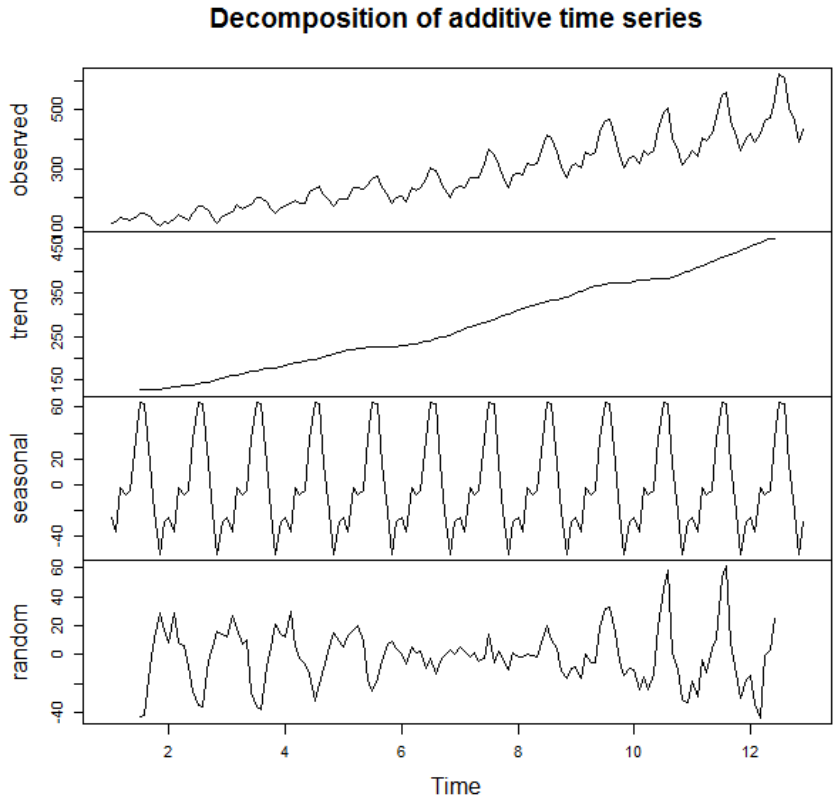


Figura 1.2. Paraqitje grafike e përbërësve të serisë kohore

Vihet re ne serinë origjinale se ka përsëritje të sekuencave. Gjatësia e tyre mund të jetë disa muaj ose vjetore. Në varësi të problemit që trajtohet, mund të jemi të interesuar për sekuenca që përsëriten një numër të caktuar herësh, për sekuenca që shtrihen në një segment të caktuar të serisë, ose dhe për sekuencen që ka numrin maksimal të përsëritjeve përgjatë gjithë serisë.

Përkufizim 1.3 **Sekuencë bazë** në një seri kohore të dhënë T, do të quajmë nënsekuencën $M = (T_i, T_{i+1}, ..., T_{i+m-1})$ me numrin më të madh të përsëritjeve në serinë T.

Shpesh lindin probleme përsa i përket zbulimit së motiveve. Ndodh në pjesën më të madhe të rasteve që dy sekuenca (të ndryshme në formë) që përsëriten mund të kenë madhësi ngjashmërie (distancat e llogaritura) në vlera të përafërta, brenda rrezes së lejuar.

Përkufizim 1.4 **Sekuencë e ngjashme fqinje** (angl. trivial match) me sekuencën bazë M, me gjatësi *m*, janë ato nënseri M' të serisë T, me gjatësi *m*, të tilla që indeksi i fillimit i sekuencës M dhe indeksi i fillimit të sekuencës M', të jenë të paktën *m-1* njësi larg.

Në dritaren e parë (Figura 1.3) shohim serinë kohore (në ngjyrë të zezë) dhe sekuencat e zbuluara (në ngjyrë të kuqe), dhe sekuencave të ngjashëme (ngjyrë blu). Ato vendosen në një dritare të tillë, në mënyrë që të dallohet më qartë ngjashmëria në formë e motiveve të gjetura.

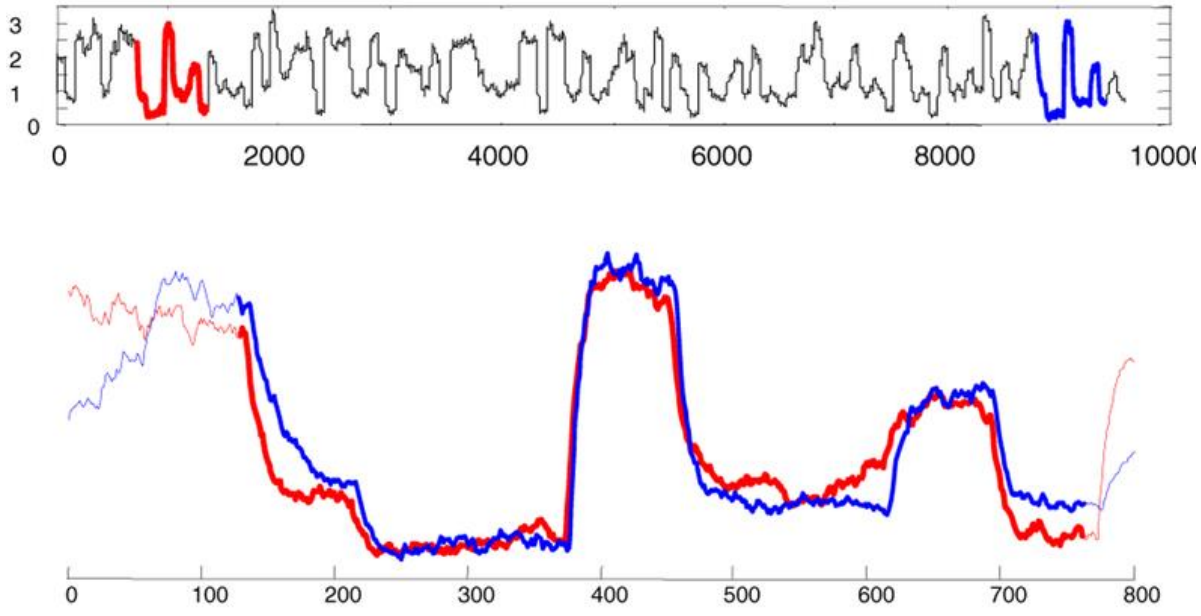


Figura 1.3 Paraqitje grafike e sekuencave (motiveve) jo fqinje në një seri kohore (Burimi: G. Rapaj , 2024)

Për më tepër, një problem që haset shpesh, është dhe kriteri që përdoret për të thënë nëse dy sekuenca $M_i = (T_i, T_{i+1}, ..., T_{i+m-1})$ dhe $M_j = (T_j, T_{j+1}, ..., T_{j+m-1})$, janë apo jo të ngjashëm.

Përkufizim 1.5 Le të jetë dhënë *x*, *y*, vektorë në R^n. Pra, $x = (x_1, x_2, ..., x_n)$ dhe $y = (y_1, y_2, ..., y_n)$. Le të jetë dhënë *d* një funksion, pasqyrim i hapësirës R^n në R^+, si më poshtë:

$$d: R^n \rightarrow R_*^+ \tag{1.1}$$

Distancë midis dy vektorëve x e y, do të quhet funksioni d (ek. 1.1), që gëzon vetitë:

1. Vetinë e pozitivitetit: $\forall x, y \in R^n,\ d(x, y) \geq 0$
2. Vetinë e identitetit: $\forall x, y \in R^n, d(x, y) = 0 \Leftrightarrow x = y$
3. Vetinë e simetrisë: $\forall x, y \in R^n, d(x, y) = d(y, x)$
4. Vetinë e kalimtaritetit: $\forall x, y, z \in R^n, d(x, y) \leq d(x, y) + d(y, z)$

1.2. **Madhësi të ndryshme ngjashmërie.**

Të shumta janë llojet e distancave, që i përmbushin të tre kushtet për të qenë distanca. Por, ka mjaft raste kur të paktën njëri nga këto kushte nuk plotësohet. Atëherë kemi të bëjmë me një tjetër madhësi po aq të rëndësishme sa dhe distanca, të quajtur **madhësi ngjashmërie**.

Përkufizim 6 **Madhësi ngjashmërie** do të quajmë atë madhësi që cënon të paktën njërin nga kushtet bazë për të qenë distancë.

Ka mjaft madhësi ngjashmërie (jo-ngjashmërie), midis të cilave përmendim distancën Manhattan, distancën Minkowski, distancën Jaccard, etj. Megjithatë, egzistojnë modifikime

të këtyre distancave edhe në rastin e të dhënave jo numerike.

Përkufizim 7 Le të jenë dhënë $X = (X_1, X_2, ..., X_n)$ dhe $Y = (Y_1, Y_2, ..., Y_n)$, dy vektorë në R^n. **Distanca Manhattan** midis dy vektorëve X dhe Y, do quhet madhësia, që përkufizohet si:

$$d_{Manh}(X, Y) = \sum_{i=1}^{n} | X_i - Y_i | \tag{2}$$

Përkufizim 8 Le të jenë dhënë $X = (X_1, X_2, ..., X_n)$ dhe $Y = (Y_1, Y_2, ..., Y_n)$, dy vektorë në R^n. **Distanca Minkowski** midis dy vektorëve X dhe Y, do quhet madhësia, që përkufizohet si:

$$d_{Mink}(X, Y) = (\sum_{i=1}^{n} (X_i - Y_i)^p)^{1/p}, \; p = 1, 2, ... \tag{3}$$

Më poshtë do shohim se distanca euklidiane dhe distanca Manhattan janë thjesht një rast i veçantë i distancës Minkowsi.

Përkufizim 9 Le të jenë dhënë $X = (X_1, X_2, ..., X_n)$ dhe $Y = (Y_1, Y_2, ..., Y_n)$, dy vektorë në R^n. Distanca **Jaccard** midis vektorëve X dhe Y, është:

$$d_{Jacc}(X, Y) = \frac{\sum_{i=1}^{n} X_i * Y_i}{[\sum_{i=1}^{n} (X_i)^2] + [\sum_{i=1}^{n} (Y_i)^2] - \sum_{i=1}^{n} X_i * Y_i} \tag{4}$$

Përkufizim 10 Le të jenë dhënë $X = (X_1, X_2, ..., X_n)$ dhe $Y = (Y_1, Y_2, ..., Y_n)$, dy vektorë në R^n. **Distanca Dice** midis dy vektorëve X dhe Y, do quhet madhësia, që përkufizohet si:

$$d_{Dice}(X, Y) = 2\frac{\sum_{i=1}^{n} X_i * Y_i}{[\sum_{i=1}^{n} (X_i)^2] + [\sum_{i=1}^{n} (Y_i)^2]} \tag{5}$$

Më poshtë do shohim se distanca euklidiane dhe distanca Manhattan janë thjesht një rast i veçantë i distancës Minkowsi.

Përkufizim 11 Le të jenë dhënë $P=(P_1, P_2, ..., P_n)$, $Q=(Q_1, Q_2, ..., Q_n)$ dy seri kohore me gjatësi n. **Distanca euklidiane** midis tyre, do ishte:

$$d_{Eukl}(P,Q)=\sqrt{\sum_{i=1}^{n}(P_i-Q_i)^2} \tag{6}$$

Gjatë trajtimit të kësaj distance në seritë kohore, janë vënë re disa disavantazhe, sikurse janë trajtimi i cikleve, sezonaliteti, fluktuacionet, etj. Për këtë arsye, janë propozuar madhësitë e ngjashmërisë, që kanë si qëllim zbutjen e ndikimit të sezonaliteteve, të trendit, etj. Një ndër to është dhe madhësia e ngjashmërisë **CID (Complexity Invariant Distance**), e propozuar nga Batista (Batista, et al. 2013). Ajo përdor një madhësi kompleksiteti për secilën nënsekuencë, që quhet **vlerësim i kompleksitetit** (**angl.** complexity estimation), për të zbutur ndikimin që trendi apo sezonaliteti mund të ketë në to.

Një variant i matjes së kompleksitetit të serisë kohore, është:

Përkufizim 12 **Vlerësim i kompleksitetit** për një seri kohore të dhënë $P=(P_1,P_2,...,P_m)$, është madhësia, që përkufizohet si:

$$CE(P)=\sqrt{\sum_{i=1}^{m-1}(P_i-P_{i-1})^2} \tag{7}$$

Më pas madhësitë e kompleksitetit vendosen në një koeficient, që quhet **koeficient rregullues i kompleksitetit** (**angl.** complexity correction factor), që shërben për llogaritjen e madhësisë s ngjashmërisë midis dy serive kohore P dhe Q.

Përkufizim 13 **Koeficient Rregullues i Kompleksitetit** për $P=(P_1, P_2, ..., P_m)$ dhe $Q=(Q_1, Q_2, ..., Q_m)$, seri kohore të dhëna, është madhësia, që përkufizohet si:

$$CF(P,Q) = \frac{\max\{CE(P), CE(Q)\}}{\min\{CE(P), CE(Q)\}} \tag{8}$$

Përkufizim 14 **Madhësia CID** midis $P = (P_1, P_2, ..., P_m)$ dhe $Q = (Q_1, Q_2, ..., Q_m)$, seri kohore të dhëna, është madhësia, që përkufizohet si:

$$d_{CID}(P,Q) = d_{Eukl}(P,Q) * CF(P,Q) \tag{9}$$

Një tjetër distancë po aq e përdorur për seritë kohore, është dhe distanca DTW (Keogh, et al.2005), që ka në bazë algoritmin e Shkurres së Rrugëve më të Shkurtra të Kruskal et al., (1983).

Përkufizim 15 **Distancë DTW** midis $P = (P_1, P_2, ..., P_n)$ dhe $Q = (Q_1, Q_2, ..., Q_n)$, seri kohore të dhëna, është madhësia, që përkufizohet si:

$$d_{DTW}(P,Q) = \min\left\{ \sqrt{\sum_{i=1}^{n} w_i} \right. \tag{10}$$

, ku $\sqrt{\sum_{i=1}^{n} w_i}$ tregon distancën midis serive P dhe Q.

Përveç rregullatorëve linearë, ka dhe rregullatorë jolinearë, që kanë ose jo në bazë distancën euklidiane. Një ndër këto madhësi, është dhe madhësia e ngjashmërisë e propozuar nga Chouakria et al. (2007), që shihet si prodhim i koeficientit të korrelacionit të përkohshëm midis dy serive me një nga distancat e njohura.

Përkufizim 16 **Koeficient i korrelacionit të përkohshëm** midis serive të dhëna $P = (P_1, P_2, ..., P_m)$ dhe $Q = (Q_1, Q_2, ..., Q_m)$, është madhësia, që përkufizohet si:

$$COR_t(P,Q) = \frac{\sum_{i=1}^{m-1} (P_i - P_{i+1}) * (Q_i - Q_{i+1})}{\sqrt{(\sum_{i=1}^{m-1} (P_i - P_{i+1})^2) * (\sum_{i=1}^{m-1} (Q_i - Q_{i+1})^2)}} \tag{11}$$

Përkufizim 17 **Madhësia e ngjashmërisë Chouakria** midis dy serive të dhëna $P = (P_1, P_2, ..., P_m)$ dhe $Q = (Q_1, Q_2, ..., Q_m)$, është madhësia që përkufizohet si:

$$d_{Chou}(P, Q) = \delta(P, Q) * \frac{2}{1 + e^{k * CORT(P,Q)}}, k = 0, 1, 2, ... \quad (12)$$

, ku $\delta(P,Q)$ mund të jetë distanca euklidiane midis serive kohore P dhe Q, distanca DTW apo dhe ndonjë distancë tjetër. Në punimin e tyre, Rapaj (Rapaj, et al. 2024), propozuan indeksin e ngjashmërisë Chouakria me CID, i cili jepte rezultate të kënaqshme për gjetjen e motiveve.

Përkufizim 18 ***Keoficenti i korrelacionit midis dy ndryshoreve te rastit X,Y*** është madhësia që përkufizohet si ***:***

$$d(x, y) = \frac{\sum_{i=1}^{m} xi\ yi\ - m\ \mu x\ \mu y}{m\ \sigma x\ \sigma y}$$

Kapitulli 2

Disa Algoritme për zbulimin e sekuencave të ngjashme në seri kohore

Qëllimi i këtij kapitulli është prezantimi i disa algoritmeve për zbulimin e sekuencave (motiveve) të ngjashme në një seri kohore të dhënë. (G. Rapaj, et al., 2024; G. Rapaj, et al., 2024), etj.

Për më tepër, synimi i këtij kapitulli është të tregojë rezultatet e përftuara nga përdorimi i disa prej distancave në algoritmin e ndërtuar për **zbulimin e motiveve** (**angl.** motif discovery) si në gjetjen e numrit të motiveve, në kohën e ekzekutimit të algoritmit, dhe në kompleksitetin e

madhësive të ngjashmërisë të përdorura.

2.1. Koncepte bazë

Që në vitet 1993, u panë përpjekjet e para për të gjetur një mënyrë për të zbuluar sekuencat e përsëritura (motivet) në seritë kohore (Agrawal, et al., 1993). Teknika bazohej në zbutjen e ndikimit të trendit në serinë kohore, zbutjen e ndikimit të zhurmave, etj. Teknikat që pasuan kishin një bazë të përbashkët, që ishte reduktimi i përmasës së të dhënave. Ndër to përmendim Transformimin Diskret të Valëzave (angl. Discrete Wavelet Transformation) (Chen, et al. 1999), si dhe Përafrimin Pjesë-Pjesë Konstant (**angl**. Piecewise Constant/Aggregate Approximation) që solli risi në llojin e tij.

Në Figurën 2.1 jepet paraqitja grafike e zbatimit të diskretizimit PAA (Yi, et al., 2000; Keogh, et al., 2002). Seria e trajtuar përmban vrojtimet mujore të numrit të lindjeve shqiptare gjatë periudhës 1990-2022. Numri total i vrojtimeve është 396. Gjatë diskretizimit, seria u nda në 33 segmente të barabarta. Pra, për të marrë një segment, kemi mesataren e 12 vlerave të njëpasnjëshme (396/33). Në Figuren 2.1. vërejmë gjithashtu se, edhe pse seria e dytë është normalizuar, forma e saj ka mbetur po ajo.

Megjithatë, edhe në vargun PAA, forma e serisë është thuajse identike. Pra, mund të kuptohet mjaft qartë trendi zbritës i të dhënave, si dhe sezonaliteti, etj.

3) www.instat.gov.al

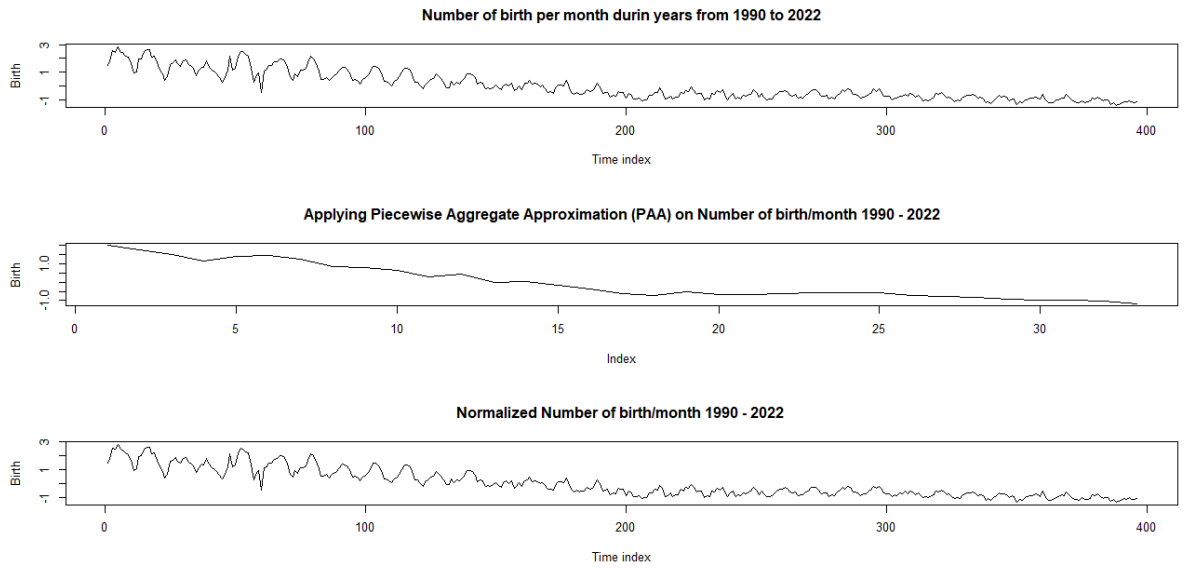


Figura 2.1. Teknika PAA në serinë shqiptare të lindjeve(figurat janë marrë nga punimet G. Rapaj)

(Shënim: Në material haset termi motiv që nënkupton sekuencë e ngjashme me përsëritje përgjatë serisë kohore)

Në punimet e tyre, Mueen et al. (*2009A; 2009B; 2010; 2013*) e trajtuan problemin e zbulimit të motiveve të ngjashme, në katër grupe.

i) **Gjatësi fikse** (**angl.** fixed length): Në këtë algoritëm, gjehen vetëm motivet më të ngjashme. Distanca e përdorur është ajo euklidiane.

ii) **Numërimi i motiveve të të gjitha gjatësive (angl.** enumeration of length**)**: Për çdo motiv me indeks fillimi në *(1, n-m+1)*, gjehen të gjitha motivet e ngjashme me gjatësi m.

iii) **k-zbulim motivesh** (**angl.** k-motif discovery): Numri i motiveve të zbuluara duhet të jetë në rangun (k, k+τ), për τ ≥ 1.

iv) **Zbulimi i drejtpërdrejtë i motiveve** (**angl**. online motif discovery): Kjo është metodë që zbatohet në seri kur të dhënat merren dhe përpunohen në mënyrë dinamike (psh., seria ECG).

Edhe pse teknikat janë të ndryshme, në bazë, trajtimi mbetet i njëjtë: Për gjatësi *m* të fiksuar të nënserisë, rrëshqitet përgjatë gjithë serisë, dhe nxirren të gjithë nënseritë me gjatësi *m*. Ilustrimi jepet në Figurën 2.2, e cila ndahet në dy dritare. Në të majtë jepet mënyra e ndarjes së një serie

kohore me gjatësi 1000, në nënseri me gjatësi *m=128*, ndërsa në të djathtë jepen motivet e formuara.

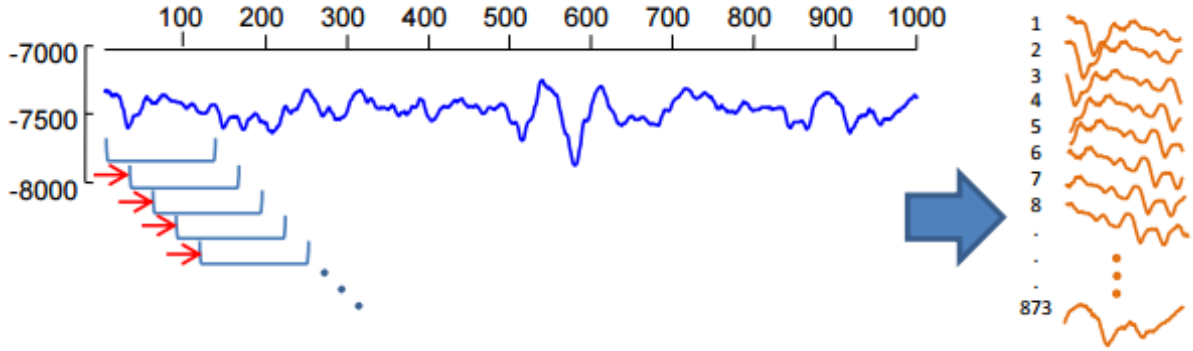


Figura 2.2. Ndarja e serisë në nënseri

Në varësi të teknikës që përdoret, kërkimi mund të bëhet një për një, ose dhe me motive që ndodhen të paktën *m-1* njësi larg njëri-tjetrit, për të zvogëluar mundësinë e mbivendosjes (**angl.** overlapping) së motiveve.

Në vitet në vazhdim, u pa se rëndësi nuk kishte vetëm teknika e përdorur. U vërejt se si distanca e përdorur, edhe normalizimi i të dhënave, kishin po aq rëndësi sa dhe teknika e përdorur. Një shembull i një serie të panormalizuar, jepet në Figurën 2.3.

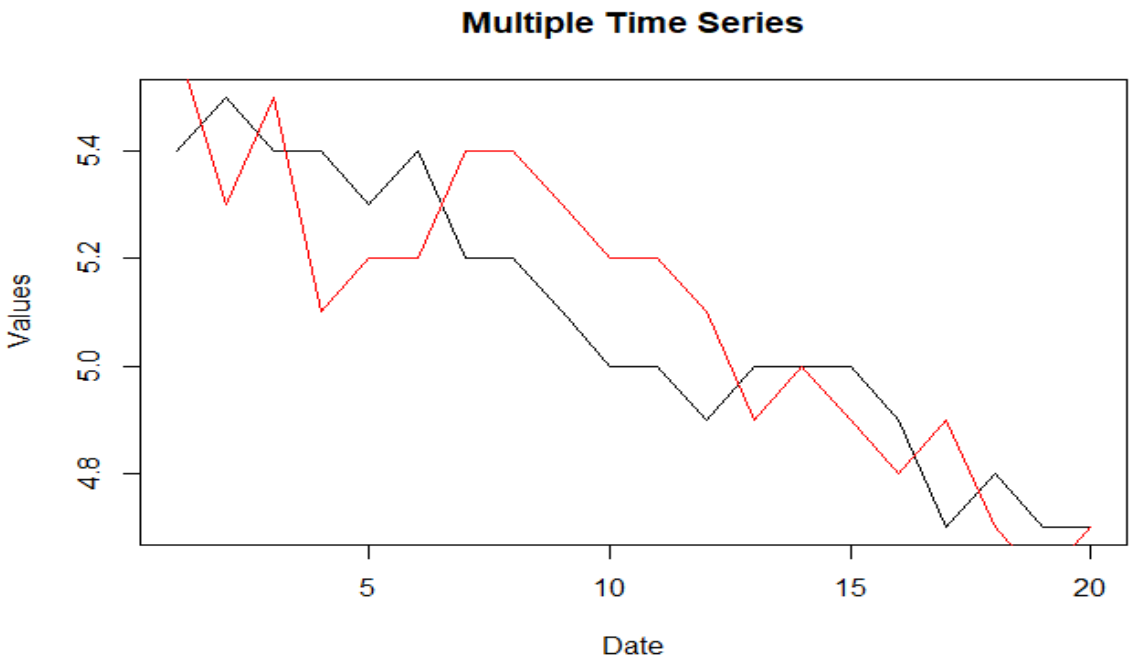


Figura 2.3. Ekzekutimi i kodit në një seri të pa standardizuar (burimi Rapaj, 2024)

Në Figurën 2.3 është përdorur seria CO2, e pa standardizuar. Vihet re se zona ku shtrihen motivet është një brez i ngushtë.

2.2. Algoritmi i zbulimit të sekuencave me përsëritje

Në artikullin e tyre, Rapaj et al. (2024) propozuan një algoritëm që merr në konsideratë një seri të normalizuar dhe përdor teknikën *1-motif*. Synimi i këtij artikulli ishte krahasimi i efektshmërisë së dy distancave (CID dhe Euklidiane) në gjetjen e motiveve.

Përkufizim 1 Le të jetë dhënë T një seri kohore me *n* vrojtime $T = (T_1, T_2, ..., T_n)$. **Standardizim normal** i serisë T do të quhet seria kohore $T' = (T_1', T_2', ..., T_n')$, që përftohet, si më poshtë:

$$T' = \frac{T - E(T)}{sd(T)} \sim N(0,1) \quad (1)$$

Le të jetë dhënë **T** një seri kohore me n vrojtime $T = (T_1, T_2, ..., T_n)$, dhe **d** një madhësi

ngjashmërie dhe **Alg** një algoritëm për gjetjen e motiveve me gjatësi *m* në serinë e dhënë T.

Efektshmëri më të lartë do ketë ajo distancë, që ka përparësi më të mëdha në njërin nga kriteret e mëposhtme (Giusti, et al.2013):

i) Numrin e motiveve të zbuluara.

ii) Koha e ekzekutimit të algoritmit.

iii) Saktësi më të madhe.

iv) Cilësia e motiveve të gjetura.

Këto kritere do të shihen në paragrafët e mëposhtëm.

Pas normalizimit dhe qendërzimit të serisë kohore, u përcaktua një kriter për masën e ngjashmërisë midis dy motiveve, në mënyrë që të konsideroheshin apo jo të ngjashëm (u vendos **ε-query** dhe u ndërtua një prag (**angl.** threshold)). Është e kuptueshme që, për kritere të ndryshme, përgjigja mund edhe të ndryshonte. Në mënyrë që të përftohej një efektshmëri më e lartë, u provuan disa madhësi të ndryshme si shtesa, për të krijuar një interval besimi për ngjashmërinë, midis të cilëve u përzgjodh ajo madhësi (Rapaj, et al. 2024) që shfaqte efektshmëri të lartë në gjetjen e motiveve si dhe që tregonte cilësi të lartë të motiveve të zbuluara. Intervali i besimit ishte në trajtën $\left]\ \min\left\{d(P,Q)\right\},\ \min\left\{d(P,Q)\right\}\ +\ \varepsilon\right[$.

Tabela 1.1. Skema 1 e algoritmit

Merren të dhënat: Seria kohore T, gjatësia m të motivit.
Seria T standardizohet. Zgjidhet distanca (Euklidiane apo CID). Zgjidhet ε-query.
Nis kërkimi i distancës minimale midis të gjithë kombinimeve dyshe të motiveve. Vlera minimale ruhet në një ndryshore . Ndërtohet intervali i besimit.
Nis kërkimin për motivin me numrin më të madh të përsëritjeve. Kërkimi bëhet nëpërmjet teknikës *Brute-force*. Të dhënat ruhen në një vektor.

Ndër *ε-query*-t e propozuara në punimin e tyre, ishin:

1. *1.96*sd(T)/sqrt(n)*
2. *sd(T)*
3. *2*sd(T)*
4. *3*sd(T)*

, ku sd(T) është shmangia mesatare katrore e serisë T dhe n është gjatësia e vektorit T. Megjithatë, nisur nga efektshmëria e secilës distancë me ε-query, u cilësua si më e mirë *ε-query= sd(T)*.

Megjithatë, në punimin pasardhës, Rapaj et al. (2024), theksuan se eliminimi i motiveve të ngjashëm është i domosdoshëm.

Në Figurën 2.4. jepet një paraqitje e rëndësisë së eliminimit të motiveve të rëndomtë në serinë *unemp*. Para modifikimit, zbulohen 18 motive. Ne presim që seria të ngjyroset më shumë. Ndërkohë, nuk ndodh kështu. Shohim që vetëm një pjesë e vogël e saj është ngjyrosur. Kjo do të thotë se motivet shtrihen njëri mbi tjetrin. Ndërkohë, numri i motiveve pas modifikimit të algoritmit, 1, dallohet qartë, pasi motivet janë të veçuar nga njëri-tjetri.

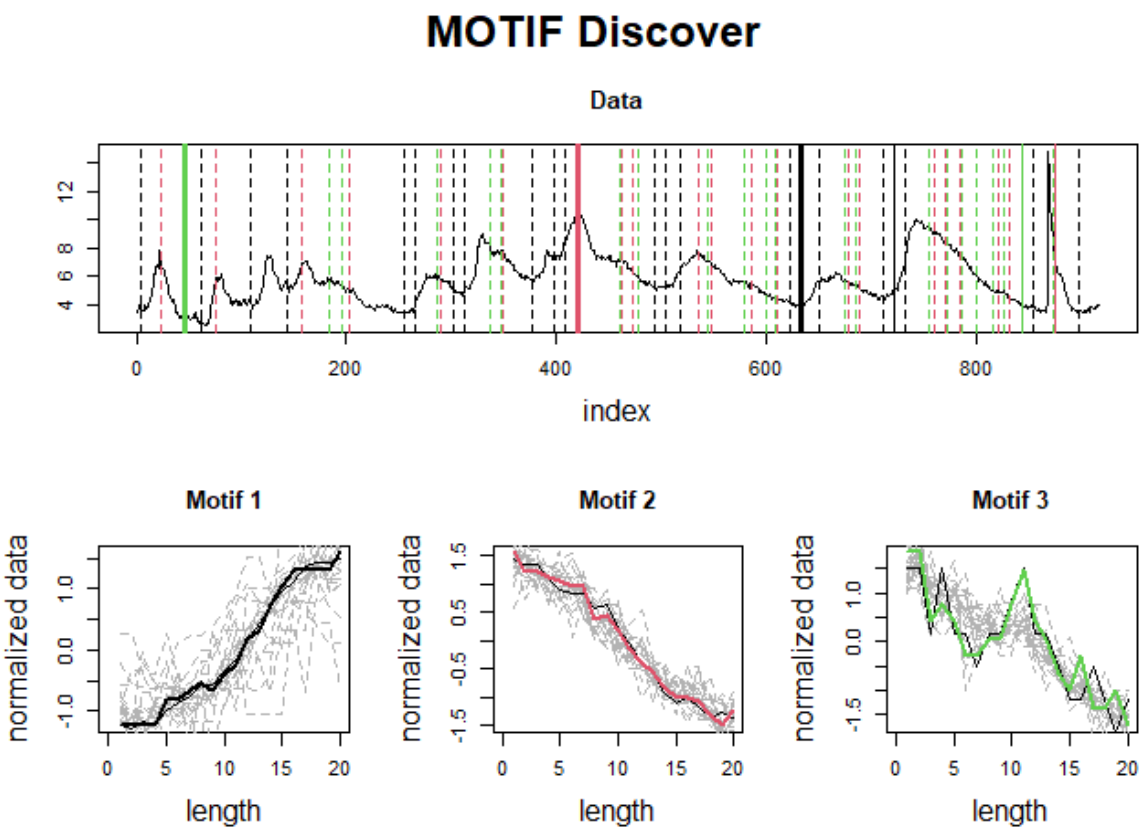


Figura 2.4.1 Pamje para ndryshimit të algoritmit për gjetje distancash

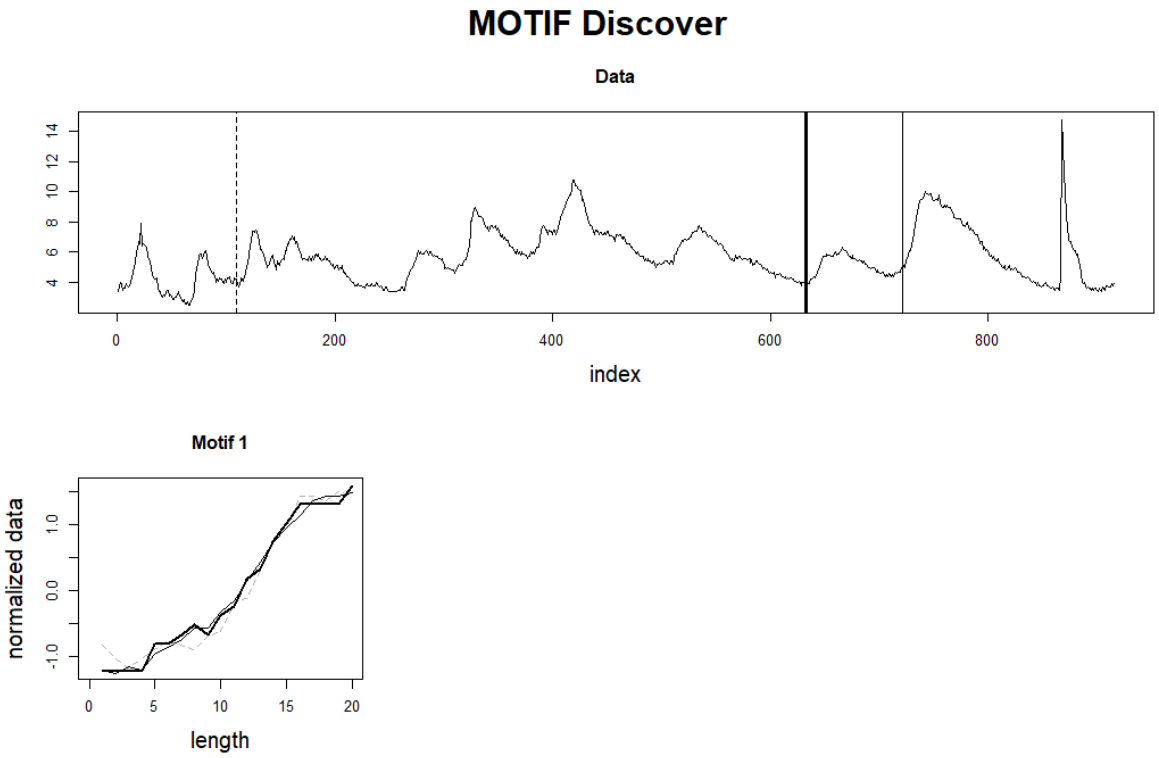


Figura 2.4.2 Pamje pas ndryshimit të algoritmit për gjetje distancash

Një tjetër modifikim që u bë, ishte dhe vendosja e një pragu individual. Pra, në rast se ne kërkojmë për motive të ngjashme me motivin me indeks fillimi *i*, atëherë ε-*query=sd*(M_i).

Tabela 1.2. Skema 2 e algoritmit të përdorur

Merren të dhënat: Seria kohore T, gjatësia m të motivit.
Seria T standardizohet. Zgjidhet distanca (Euklidiane apo CID). ε-query është e fiksuar (ε-query=sd(M_i), ku M_i është motivi i *i*-të i serisë T).
Nis kërkimi i distancës minimale midis të gjithë kombinimeve dyshe të motiveve. Vlera minimale ruhet në një ndryshore. Ndërtohet intervali i besimit.
Nis kërkimin për motivin me numrin më të madh të përsëritjeve. Kërkimi bëhet nëpërmjet teknikës *Brute-force*. Zbulohen motivet e ngjashëm, hiqen motivet e rëndomtë, dhe ruhet vlera maksimale, dhe pozicioni ku është arritur ajo.

5) Në paketën **tsmp** në R.

Sikurse u theksua, eliminimi i motiveve fqinj është mjaft i rëndësishëm. Në punimin e tyre, Rapaj et al (2024) treguan për rënien e numrit të motiveve të zbuluara pasi të ishin eliminuar nga kërkimi motivet e rëndomta. Megjithatë, u pa se, meqenëse motivet e rëndomta nuk shqyrtoheshin më, numri i motiveve të zbuluara qendronte thuajse konstant (4-5 motive të zbuluar për një seri me afërsisht 200 vrojtime). Në përpjekje për të zbuluar më shumë motive, por, pa cënuar cilësinë e tyre, u konsiderua një tjetër ε-query. Në këtë mënyrë pragu do të ishte pak më i largët. Për të kuptuar çfarë ndodh për teste me të njëjtën madhësi ngjashmërie, por, me pragje të ndryshme, më poshtë po japim disa teste, të kryera mbi disa seri. Madhësia e përdorur është ajo euklidiane. Rezultatet jepen në Tabelën 2.3.

Tabela 2.3. Para dhe pas modifikimit të intervalit të besimit për pritjen

Seria	Para modifikimit	Pas modifikimit
Norma papunësisë	5	4
Lindjet në Shqipëri	5	4
Anomalia Temperaturave	5	4
Clirimet vjetore të CO2	2	2

Detaje më të plota jepen në Figurën 2.5, që përshkruan diferencën në zbulimin e motiveve të distancës euklidiane para dhe pas modifikimit.

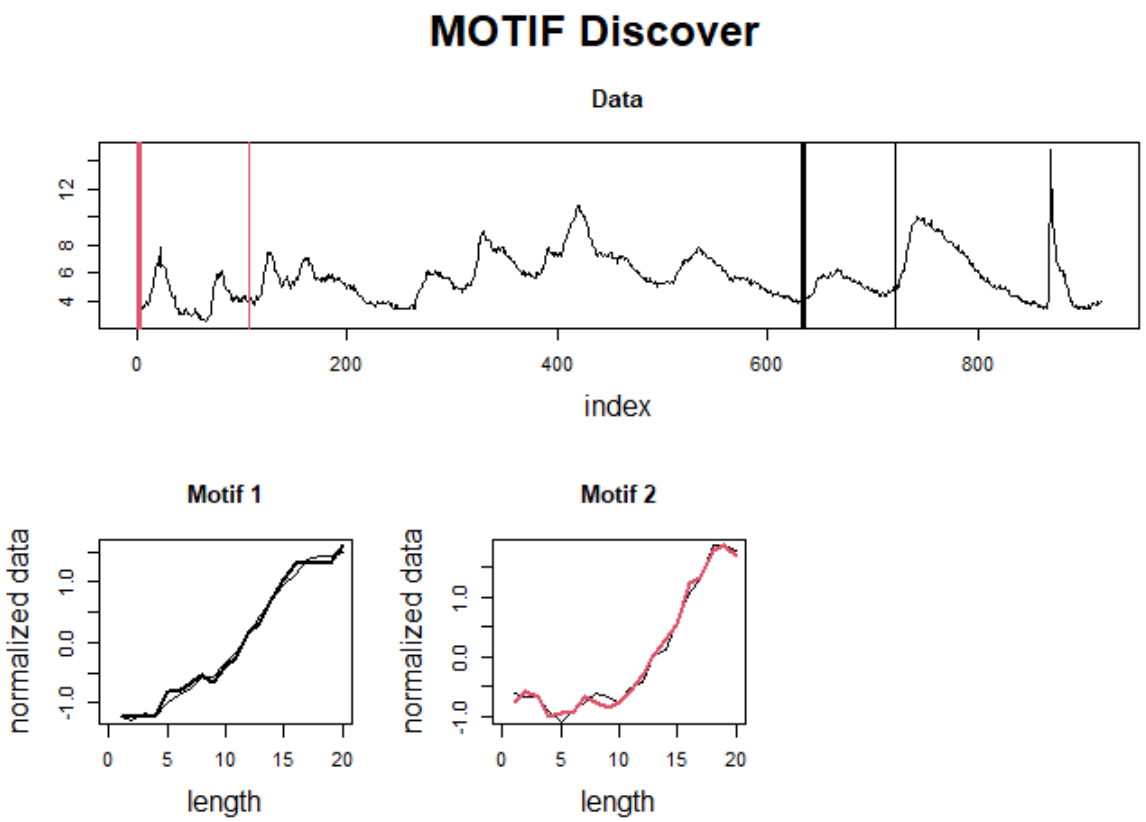


Figura 2.5. Diferenca në numër motivesh i distancës euklidiane pas dhe para modifikimit (burimi Rapaj, 2024)

Mund të vihet re se në 59.1% të rasteve, numri i motiveve mbeti i pandryshuar nga pragu. Vetëm në 1.14% të rasteve, numri i motiveve të gjetura u zvogëlua, ndërsa, në pjesën tjetër të rasteve, aftësia zbuluese e motiveve, u rrit.

Distanca Euklidiane.

Një ndër distancat më të përhapura, sikurse dihet, është distanca euklidiane. Kjo distancë e ka origjinën nga Teorema e Pitagorës. Ajo shquhet për lehtësinë në përdorim, për numër të ulët veprimesh, etj. Megjithatë, në rast se shqyrtohet si një distancë për të zbuluar motive të ngjashme në një seri të dhënë *T*, me gjatësi motivi *m* të dhënë, efektshmëria nuk është mjaft e lartë.

Përgjithësisht, krahasuar me madhësitë e tjera të ngjashmërisë, distanca euklidiane gjen më

shumë motive se CID. Megjithatë, sikurse e përmendëm më lart, ndër kriteret që përbëjnë efektshmërinë e një madhësie ngjashmërie nuk është vetëm numri i motiveve të gjetura, por është edhe cilësia e tyre. Në Figurën 2.6. jepet një ilustrim ku distanca euklidiane nuk është shumë efektive në gjetjen e motiveve. Seria e marrë në shqyrtim është seria *unemp*.

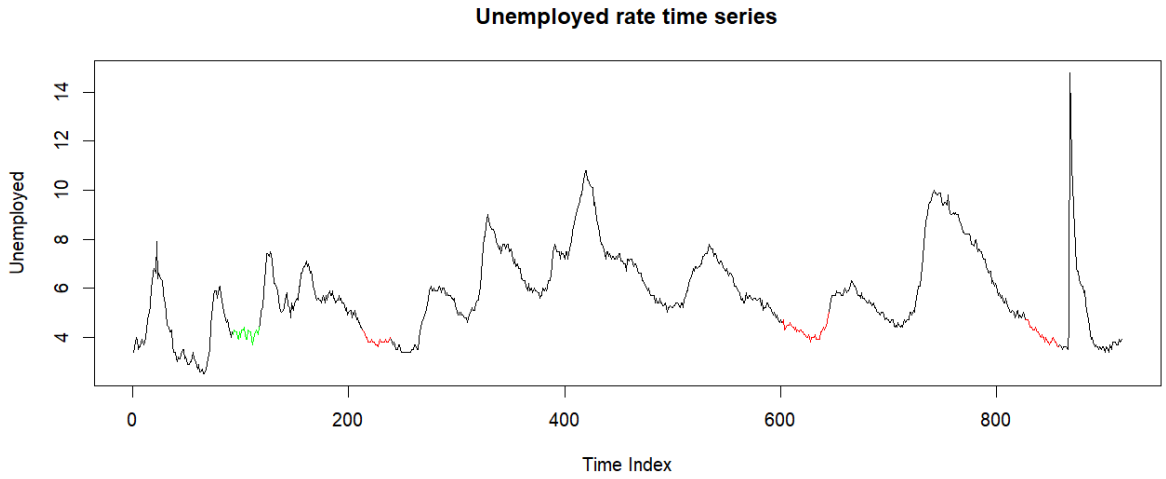


Figura 2.6. Motivet e zbuluara nga distanca euklidiane (burimi Rapaj, 2024)

Edhe pse nga pikëpamja e distancës, kriteret plotësohen, nga paraqitja grafike, situata është jo fort e qartë. Shohim forma motivesh jo të ngjashme. Lind pyetja nëse kështu, klasifikimi i këtyre motiveve si të ngjashëm, mund të quhet apo jo i rregullt.

Tjetër disavantazh i përdorimit të distancës euklidiane si madhësi ngjashmërie, është kur seria e marrë nën shqyrtim shfaq trend. Një shembull të tillë e gjejmë te Figura 2.7.

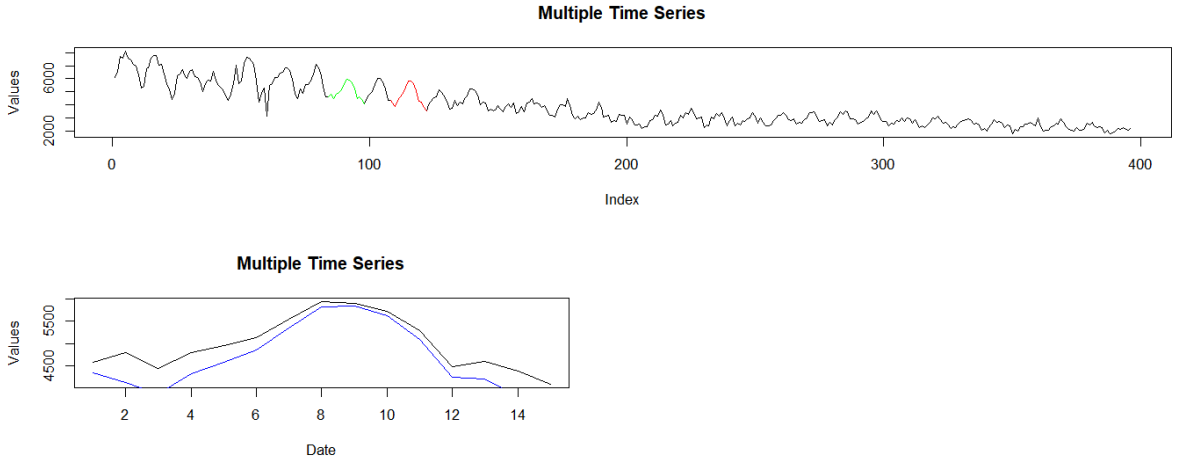


Figura 2.7. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

Megjithatë, një ndër përparësitë e përdorimit të distancës euklidiane si madhësi ngjashmërie, është *koha e ekzekutimit të algoritmit*.

Distanca CID

Në rast se jepen $P = (P_1, P_2, ..., P_m)$ dhe $Q = (Q_1, Q_2, ..., Q_m)$ dy seri kohore me gjatësi *m*. Bazuar te *Përkufizimi 1.9*, shihet si prodhim i distancës euklidiane midis këtyre serive, me vlerësimin e kompleksitetit.

$$d_{CID}(P,Q) = d_{Eukl}(P,Q) * CF(P,Q) \tag{4}$$

Meqenëse vlerësimi i kompleksitetit është madhësi që gjen diferencat e njëpasnjëshme të serisë, atëherë zvogëlohet ndikimi i trendit.

Meqenëse madhësia CF(P,Q) jepet si raport i $\max\{CE(P), CE(Q)\}$ me $\min\{CE(P), CE(Q)\}$, supozohet se vlera e CF(P,Q)≥1. Pra, distanca euklidiane shumëzohet me një koeficient me vlerë më të madhë se 1. Rrjedhimisht, distanca CID është, në pjesën më të madhe të rasteve, shumëfish i distancës euklidiane.

Rast i papërcaktuar do të ishte ai kur të paktën njëri nga vektorët P ose Q të jetë konstant. Po tregojmë në vijim çfarë do të ndodhte:

Supozojmë se vektori $P = (P_1, P_2, ..., P_m)$ është konstant. Pra, $P = (c, c, ..., c)$. Rrjedhimisht, madhësia e tij e kompleksitetit, do të jetë:

$$CE(P) = \sqrt{\sum_{i=1}^{m-1} (P_i - P_{i+1})^2} = \sqrt{\sum_{i=1}^{m-1} (c - c)^2} = \sqrt{\sum_{i=1}^{m-1} 0^2} = 0 \tag{5}$$

Si rrjedhojë, do kishim se:

$$CF(P,Q) = \frac{\max\{CE(P), CE(Q)\}}{\min\{CE(P), CE(Q)\}} = \frac{CE(Q)}{0} \uparrow \infty \tag{6}$$

Vërejmë se koeficienti rregullues i kompleksitetit do të rezultojë një madhësi pambarimisht e madhe.

Ndërkohë, në rast se të dy vektorët P dhe Q shihen si vektorë konstantë, do kishim:

$$\begin{cases} CE(P) = \sqrt{\sum_{i=1}^{m-1} (P_i - P_{i+1})^2} = 0 \\ CE(Q) = \sqrt{\sum_{i=1}^{m-1} (Q_i - Q_{i+1})^2} = 0 \end{cases} \Rightarrow \frac{\max\{CE(P), CE(Q)\}}{\min\{CE(P), CE(Q)\}} = \frac{0}{0} \tag{7}$$

Pra, kjo madhësi do ishte e pacaktuar.

Vërejmë se kusht themelor për të gjetur distancën CID midis dy vektorëve, është që ata të mos jenë konstantë. Megjithatë, kjo analizë bëhet pasi, gjatë zbatimit të algoritmit mbi seri çfarëdo, rezultati nuk mund të parashikohet. Pra, përdorimi i distances CID në të tilla situata, nuk do të jepte rezultatin e dëshiruar. Kjo është dhe arsyeja pse CID është madhësi ngjashmërie, e jo distancë e mirëfilltë.

2.3 Prezantimi i një modifikimi të indeksit të ngjashmërisë së CID.

U vu re, nga provat e kryera, se CID shfaqte një farë ngurtësie në lidhje me shfaqjen e trendit në seri, ndërsa distanca euklidiane ishte pak e besueshme në të ashtuquajturit motive të ngjashëm. Në raste të veçanta, efektshmëria e kësaj distance nuk ishte shumë e lartë. Megjithatë, në lidhje me numrin e motiveve të gjetur, distanca euklidiane është dukshëm në avantazh. Për të zbutur këto lloj dallimesh, është propozuar një modifikim i madhësisë së ngjashmërisë CID.

Meqenëse gjatë përdorimit të Koeficientit Rregullues të Kompleksitetit kemi rezulate të kënaqshme, u propozua që ndryshimi të bëhej në *vlerësuesin e kompleksitetit*. (*Burimi: G. Rapaj, 2024)*

Përkufizim 2 **Vlerësues i kompleksitetit** të indeksit të ngjashmërisë *CID i modifikuar* do të quajmë atë madhësi kompleksiteti, që përkufizohet si:

$$CE_M(P) = \sqrt{\sum_{i=1}^{m} \left(P_i - \bar{P}\right)^2} \tag{8}$$

, ku $\bar{P}$ është mesatarja e motivit të dhënë $P = (P_1, P_2, ..., P_m)$ dhe *M* qendron për *e modifikuar*.

Përkufizim 3 **Koeficient Rregullues i Kompleksitetit i modifikuar** për seritë kohore të dhëna $P = (P_1, P_2, ..., P_m)$ dhe $Q = (Q_1, Q_2, ..., Q_m)$, është madhësia, që përkufizohet si:

$$CF_M(P, Q) = \frac{\max\{CE(P), CE(Q)\}}{\min\{CE(P), CE(Q)\}} \tag{9}$$

Përkufizim 4 Madhësia **CID i modifikuar** midis serive kohore të dhëna $P = (P_1, P_2, ..., P_m)$ dhe $Q = (Q_1, Q_2, ..., Q_m)$, është madhësia, që përkufizohet si:

$$d_{CID\_\mathrm{mod}}(P, Q) = d_{Eukl}(P, Q) * CF_M(P, Q) \tag{10}$$

Sikurse vërehet, *vlerësuesi i kompleksitetit* është madhësi mjaft e ngjashme me dispersionin e serisë. Në paragrafin pasardhës do shohim efektet pozitive që ka përdorimi i kësaj madhësie kompleksiteti, krahasuar me indeksin e ngjashmërisë CID.

Përkufizim 5 Shmangie mesatare katrore e serisë kohore të dhënë $P = (P_1, P_2, ..., P_m)$ është madhësia, që përkufizohet si:

$$sd(P) = \sqrt{\frac{1}{m-1}}\sqrt{\sum_{i=1}^{m}(P_i - \overline{P})^2} \quad (11)$$

, ku $\overline{P}$ është mesatarja e motivit të dhënë $P = (P_1, P_2, ..., P_m)$.

Është e kuptueshme se, edhe nëse përdoret shmangia mesatare katrore e serisë si koeficient rregullues i korrelacionit, rezultatet do të ishin të njëjta.

$$\frac{\max\{sd(P), sd(Q)\}}{\min\{sd(P), sd(Q)\}} = \frac{\sqrt{\frac{1}{m-1}} * \max\{CE(P), CE(Q)\}}{\sqrt{\frac{1}{m-1}} * \min\{CE(P), CE(Q)\}} = CF_M(P, Q) \quad (12)$$

Megjithatë, për të mos rritur numrin e veprimeve në kod, përdoret modeli i thjeshtuar i *CE(P)*, *CE(Q)* dhe jo *sd(P)*, *sd(Q)*.

Përse CID i modifikuar nuk është distancë.

Bazuar në *Përkufizimin 1.5*, që një funksion të mund të quhet funksion distancë, duhet të plotësohen të katërta kushtet: Pra, të gëzojë vetinë e jonegativitetit, vetinë e identitetit, vetinë e simetrisë dhe të plotësohet mosbarazimi i trekëndëshit. Të tregojmë që këto kushte nuk plotësohen për madhësinë CID i modifikuar.

Le të jenë dhënë P, Q, dy nënsekuenca me gjatësi *m* të serisë T. Pra, $P = (P_1, P_2, ..., P_m)$ dhe $Q = (Q_1, Q_2, ..., Q_m)$.

1. ***Vetia e jonegativitetit***: $\forall P, Q \subset T, P \neq Q \Rightarrow d_{CID\_mod}(P, Q) \geq 0$

Meqenëse madhësia *CID i modifikuar* shihet si prodhim i distancës euklidiane midis dy nënserive P dhe Q, me vlerësimin e kompleksitetit:

$$\begin{cases} CID\_\mathrm{mod}(P,Q) = d_{eukl}(\mathrm{P},\mathrm{Q}) * CF(\mathrm{P},\mathrm{Q}) \\ \mathrm{CF(P,Q)} = \dfrac{\max\{\mathrm{CE(P)},\mathrm{CE(Q)}\}}{\min\{\mathrm{CE(P)},\mathrm{CE(Q)}\}} \\ CE(P) = \sqrt{\sum_{i=1}^{m} \left(P_i - \bar{P}\right)^2} \\ d_{eukl}(\mathrm{P},\mathrm{Q}) \geq 0 \end{cases} \tag{13}$$

, atëherë shenja e $d_{CID\_\mathrm{mod}}(\mathrm{P},\mathrm{Q})$ varet vetëm nga shenja e CF(P,Q). Shohim tashmë shenjën e vlerësimit të kompleksitetit.

Sikurse u theksua dhe më lart, CE(P) ngjan me dispersionin e serisë P. Nga vetitë probabilitare, dispersion merr vetëm vlera pozitive.

$$\begin{cases} CE(P) = \sqrt{\sum_{i=1}^{m} \left(P_i - \bar{P}\right)^2} \geq 0 \\ CE(Q) = \sqrt{\sum_{i=1}^{m} \left(Q_i - \bar{Q}\right)^2} \geq 0 \end{cases} \Rightarrow CF(\mathrm{P},\mathrm{Q}) \geq 0 \tag{14}$$

Për më tepër, raporti i dy madhësive positive është madhësi pozitive. Ndaj, $\mathrm{CF(P,Q)} \geq 0$.

Përfundimisht, CID i modifikuar e gëzon vetinë e pozitivitetit.

2. **Vetia e identitetit:** $d_{CID\_\mathrm{mod}}(\mathrm{P},\mathrm{Q}) = 0 \xrightarrow{?} P = Q$

Që $d_{CID\_\mathrm{mod}}(\mathrm{P},\mathrm{Q}) = 0$ duhet që të plotësohet njëri nga kushtet:

$$d_{eukl}(\mathrm{P},\mathrm{Q}) = 0 \vee CF(\mathrm{P},\mathrm{Q}) = 0 \tag{15}$$

Nga vetitë e distancës euklidiane, kemi se:

$$P \neq Q \Leftrightarrow d_{eukl}(P,Q) > 0 \tag{16}$$

Por, meqenëse CID_mod shihet si prodhim dy faktorësh, duhet të shohim çfarë ndodh me faktorin tjetër.

$$\begin{cases} CE(P) = \sqrt{\sum_{i=1}^{m} \left(P_i - \bar{P} \right)^2} \\ CE(Q) = \sqrt{\sum_{i=1}^{m} \left(Q_i - \bar{Q} \right)^2} \\ P \neq Q \end{cases} \tag{17}$$

Fakti në P dhe Q janë apo jo seri të ndryshme, nuk ndikon në rezultatin e faktorit rregullues. Supozojmë që seria kohore P është një seri konstante.

$$P = (c, c, ..., c) \tag{18}$$

Shohim tashmë çfarë ndodh me CE(P).

$$\bar{P} = \frac{1}{m}\sum_{i=1}^{m} P_i = \frac{1}{m}\sum_{i=1}^{m} c = \frac{c}{m}\sum_{i=1}^{m} 1 = \frac{c}{m} m = c \tag{19}$$

Shohim që mesatarja e vektorit P është konstante.

$$CE(P) = \sqrt{\sum_{i=1}^{m} \left(P_i - \bar{P} \right)^2} = \sqrt{\sum_{i=1}^{m} \left(c - c \right)^2} = \sqrt{\sum_{i=1}^{m} 0} = 0 \tag{20}$$

Deri më tani nuk është vënë asnjë kusht për vektorin Q. Rrjedhimisht, do kemi se CE(Q)>0. Ndaj, do kishim:

$$CF(P, Q) = \frac{\max\{CE(P), CE(Q)\}}{\min\{CE(P), CE(Q)\}} = \frac{CE(Q)}{0} \uparrow \infty \tag{21}$$

Pra, në rast se të paktën njëra nga nënsekuencat është konstante, atëherë distanca CID e modifikuar nuk mund të vlerësojë distancën midis dy vektorëve P dhe Q. Në rastin kur të dy nënseritë P dhe Q do të jenë njëkohësisht konstante, atëherë do ishim në një situatë të tillë:

$$\begin{cases} CE(P) = \sqrt{\sum_{i=1}^{m} \left(P_i - \bar{P}\right)^2} = 0 \\ CE(Q) = \sqrt{\sum_{i=1}^{m} \left(Q_i - \bar{Q}\right)^2} = 0 \end{cases} \Rightarrow \frac{\max\{CE(P), CE(Q)\}}{\min\{CE(P), CE(Q)\}} = \frac{0}{0} \quad (22)$$

Pra, do të kishim një raport të pacaktuar, i cili, meqenëse vlerësuesit e kompleksitetit vlerësohen në mënyrë të pavarur, nuk mund të zgjidhet as me rregull L'Opitali, as me ndonjë metodë tjetër të njohur.

Është e kuptueshme që ky është problem që qendon edhe te CID. Aty shfaqet po i njëjti problem në rast se diferencat e njëpasnjëshme janë 0, që do të thotë që seria të jetë konstante.

3. **Vetia e simetrisë:** $CID\_\mathrm{mod}(P,Q) \stackrel{?}{=} CID\_\mathrm{mod}(Q,P)$

Me shumë pak veprime mund të tregojmë që CID i modifikuar e gëzon vetinë e simetrisë, si më poshtë:

$$d_{CID\_mod}(P,Q) = \sqrt{\sum_{i=1}^{m} (P_i - Q_i)^2}\, \frac{\max\{CE(P), CE(Q)\}}{\min\{CE(P), CE(Q)\}} = \sqrt{\sum_{i=1}^{m} (Q_i - P_i)^2}\, \frac{\max\{CE(Q), CE(P)\}}{\min\{CE(Q), CE(P)\}} = d_{CID\_mod}(Q,P) \quad (23)$$

4. **Vetia e kalimtaritetit:** $d_{CID\_\mathrm{mod}}(P,Q) \le d_{CID\_\mathrm{mod}}(P,R) + d_{CID\_\mathrm{mod}}(R,Q)$

Pra, distanca midis dy nënsekuencave P, Q, a është më e vogël se shuma e distancave të P nga R dhe R nga Q?

Dimë që distanca euklidiane gëzon vetinë kalimtare, që mund të shkruhet si më poshtë:

$$d_{eu}(P,Q) \le d_{eu}(P,R) + d_{eu}(R,Q) \quad (24)$$

Shënojmë me:

$$k = \frac{\max\{CE(P), CE(Q)\}}{\min\{CE(P), CE(Q)\}} \quad (25)$$

Rrjedhimisht, do të kemi:

$$d_{CID\_mod}(P,Q) = k^* d_{eu}(P,R) \le k^* (d_{eu}(P,R) + d_{eu}(R,Q)) \quad (26)$$

Meqenëse $k \geq 1$, për P,Q jo konstante, do kishim:

$$d_{CID\_mod}(P,Q) \le k^* (k_1^* d_{eu}(P,R) + k_2 * d_{eu}(R,Q))) \quad (27)$$

, ku $k_1 = \frac{\max\{CE(P), CE(R)\}}{\min\{CE(P), CE(R)\}}$ dhe $k_2 = \frac{\max\{CE(R), CE(Q)\}}{\min\{CE(R), CE(Q)\}}$.

Pra, do të kishim se:

$$d_{CID\_mod}(P,Q) \le d_{CID\_mod}(P,R) + d_{CID\_mod}(Q,P) \quad (28)$$

Shohim se madhësia CID i modifikuar e gëzon vetinë e kalimtaritetit.

Treguam që CID i modifikuar nuk është distancë e mirëfilltë, pasi cënohet kushti i parë për të qenë i tillë. Prandaj, ajo mbetet thjesht një madhësi ngjashmërie, sikurse dhe CID.

Krahasimi u bë sërish për të treguar se, në varësi të të dhënave, madhësitë e ngjashmërisë mund të shfaqin dhe probleme. Kemi përzgjedhur këto të dyja për t'i testuar me të dhëna reale, prandaj rezultatet po i paraqesim fillimisht në formë tabelare dhe po shfaqim vizualisht disa nga motvet e gjetura si nga distanca Euklidit dhe nga CID në mënyrë që të mos mbi ngarkohet por i gjithe informacioni është në shtojcë.

Tabela 2.4. Para dhe pas modifikimit të intervalit të besimit për pritjen

Seria	Distanca	Para Modifikimit	Pas Modifikimit	Madhësia ngjashmërisë	Para Modifikimit	Pas Modifikimit
Norma papunësisë	Euklidit	5	4	CID	8	3
Lindjet në Shqipëri		5	4		6	5
anomalia Temperature		4	4		7	3
Clirimet vjetore të CO2		2	2		2	2

Po fillojmë me serinë e parë kohore atë të lindjeve që nga viti 1990 deri në vitin 2022. Fillimisht vizualizojmë informacionin për të krijuar një ide paraprake.

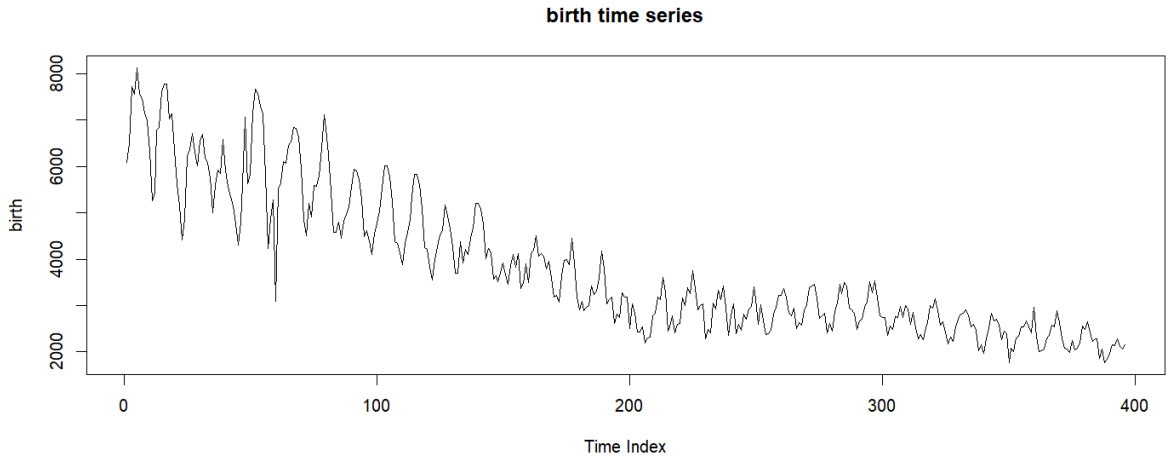


Figura 2.8. Serine *lindjeve* Shqiperi marr nga INSTAT

Pasi e investigojmë paraprakisht grafikun me sy të lirë mund të dallojmë ndonjë motiv por nuk jemi të sigurt prandaj duhet të përdorim CID dhe distancën Euklidit për t'i zbuluar. Zbatojmë algoritmin BruteForce (të cilin e kemi ndërtuar më pare kodi i të cilit do të paraqitet në kapitullin pasardhës i shpjeguar) ku i japim si input serinë, 1 vlerë numerike që përfaqëson gjatësinë e dritares lëvizëse dhe fillimisht i japim vlerën Euklidian për të zbatuar distancen e euklidit. Algoritmi na kthen një matricë trekendeshe ku rreshti i i-të mban distancën e sekuencës i me dritaren lëizese në indexin e j. Hapi tjetër është filtrimi i vlerave të rreshtit të i-të. Domethënë do të marrim ato vlera të rreshtit të i-të të matricës që janë më të vogla se epsilon-query i përcaktuar më parë nga ne.

Për epsilon-query sa shmangia mesatare katrore ne gjejmë motivet të cilat i inspektojmë manualisht pasi ka edhe nënsekuenca që plotesojnë kushtin që distanca e euklidit është më e vogël se epsilon-query por që nuk kanë pattern të ngjashme. Si përshembull motivi i ilustruar në grafikun e mëposhtëm,

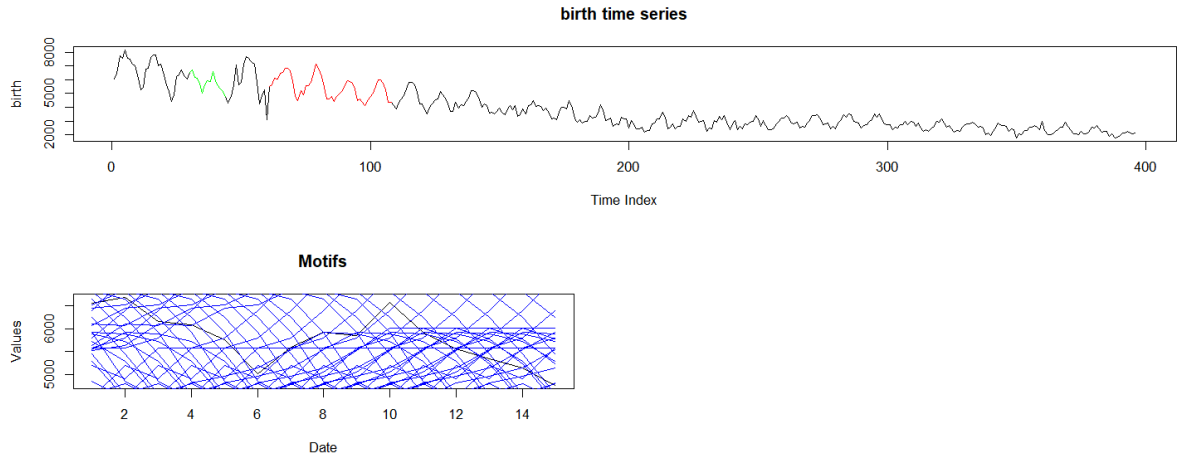


Figura 2.9. Motive qe jane te ngjashme nga ana matematikore per shkak te distances por jo nga forma e tyre. (burimi Rapaj, 2024)

Pasi bëjmë inspektimin e grafikëve që numerikisht janë të ngjashëm, manualisht po i paraqesim motivet e gjetura në grafikun original në menyrë që rezultati i gjetur të jetë më intuitive/kuptueshem.

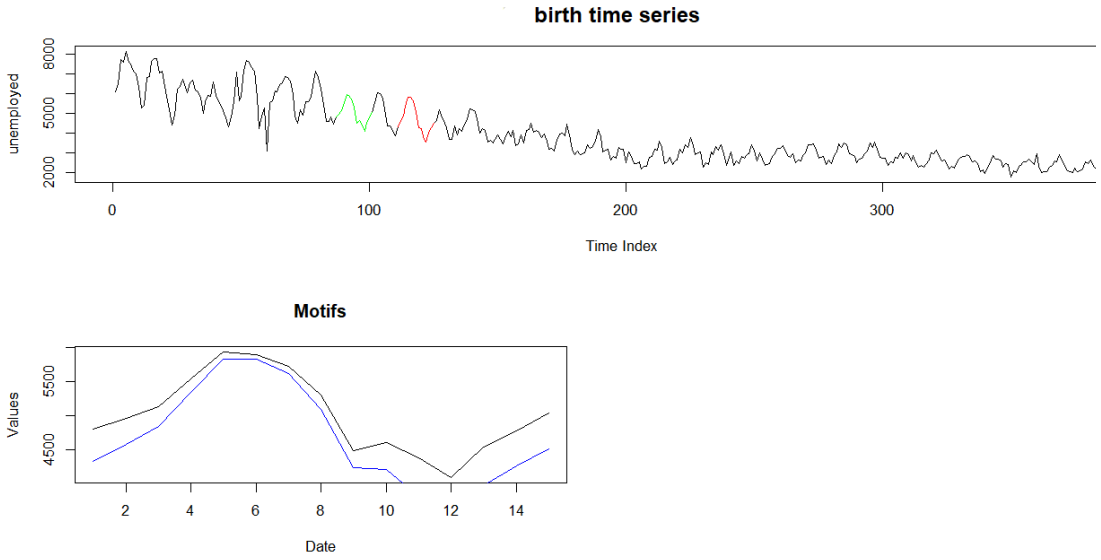


Figura 3.0. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

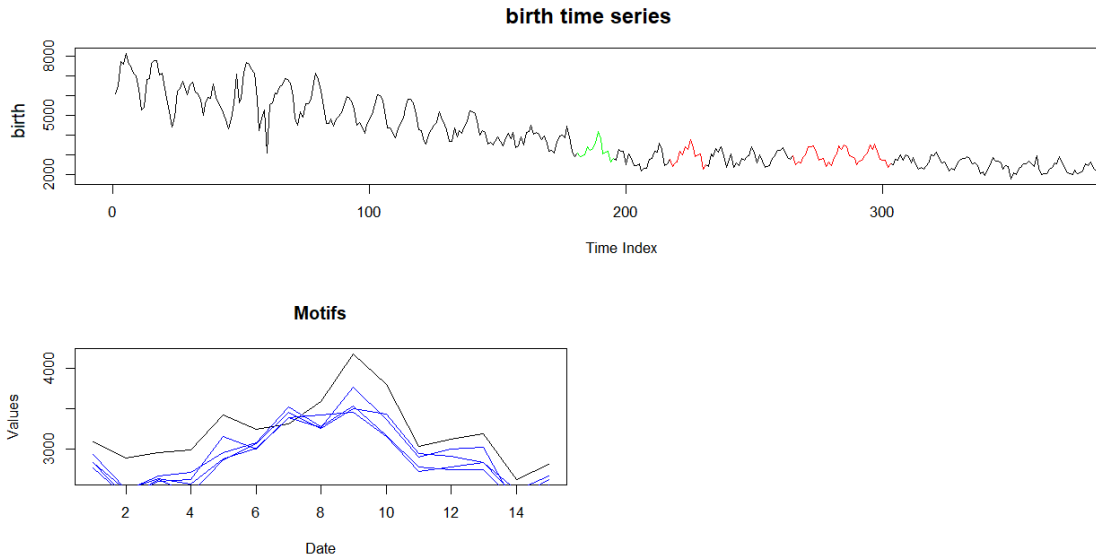


Figura 3.1. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

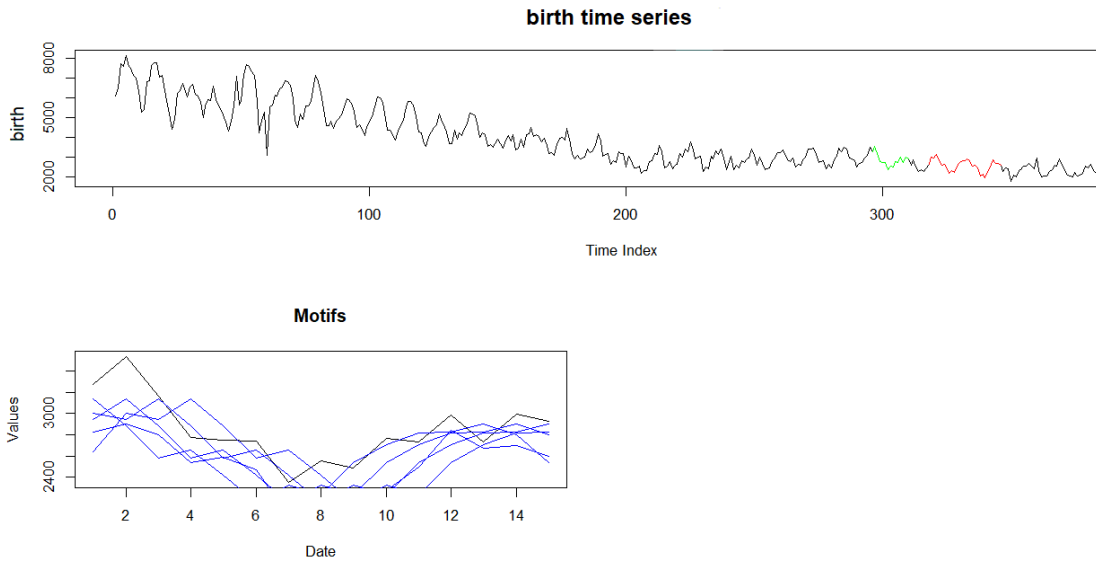


Figura 3.2. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

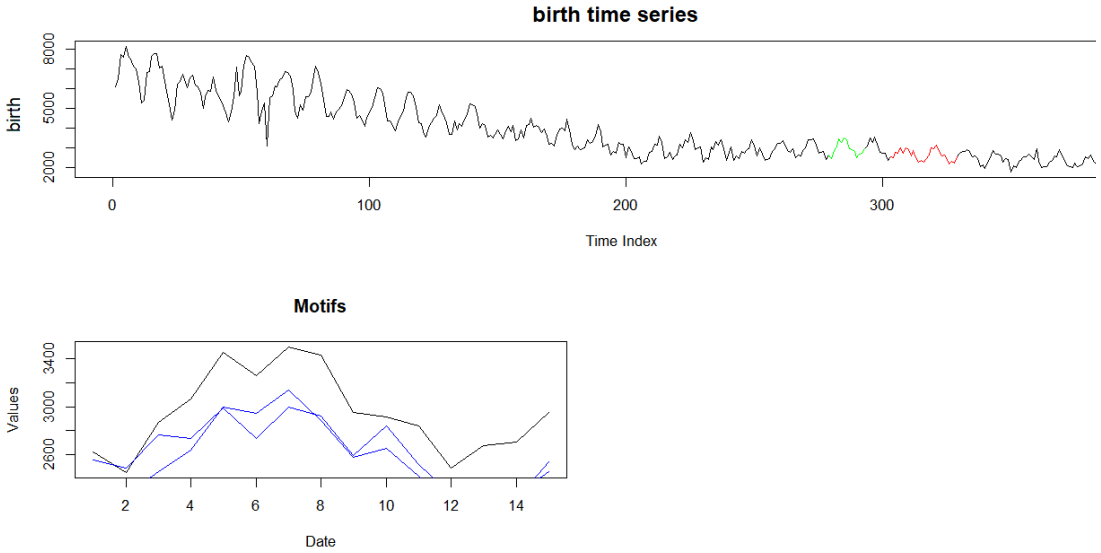


Figura 3.3. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

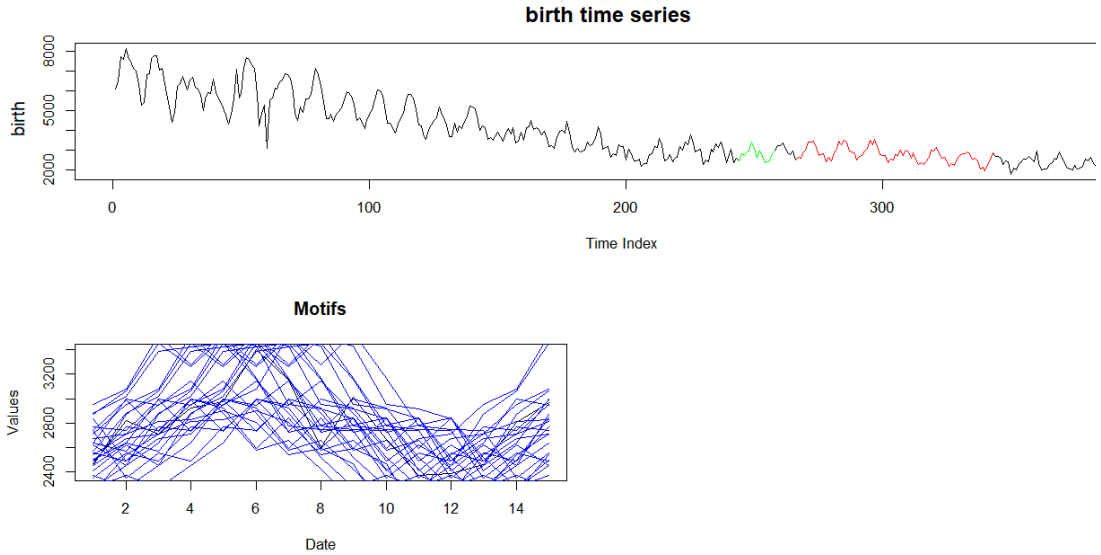


Figura 3.4. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

Zbatojmë algoritmin BruteForce ku i japim si input serinë, 1 vlerë numerike që përfaqëson gjatësinë e dritares lëvizëse dhe i japim vlerën chouakria për të zbatuar CID. Algoritmi na kthen një matricë trekendeshe ku rreshti i i-të mban distancën e sekuencës i me dritaren lëizese në indexin e j. Hapi tjetër është filtrimi i vlerave të rreshtit të i-të. Domethënë do të marrim ato

vlera të rreshtit të i-të të matricës që janë më të vogla se epsilon-query i përcaktuar më parë nga ne.

Për epsilon-query sa shmangia mesatare katrore, ne gjejmë motivet të cilat i inspektojmë manualisht pasi ka edhe nënsekuenca që plotesojnë kushtin që CID është më e vogël se epsilon-query por që nuk kanë pattern të ngjashme. Pasi bëjmë inspektimin e grafikëve që numerikisht janë të ngjashëm, manualisht po i paraqesim motivet e gjetura në grafikun original në menyrë që rezultati i gjetur të jetë më intuitive/kuptueshem.

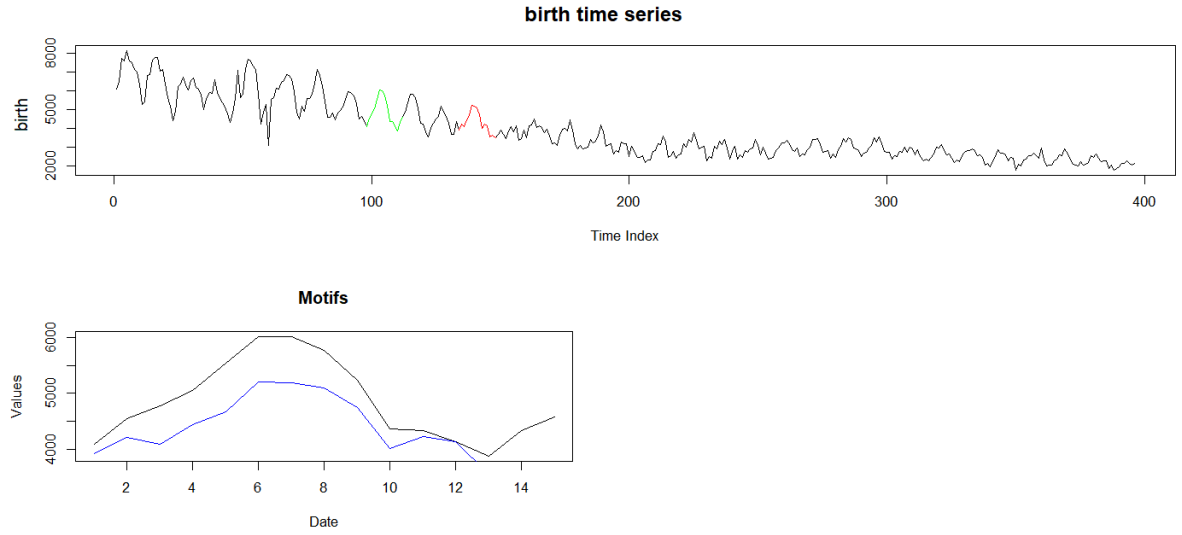


Figura 3.5. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

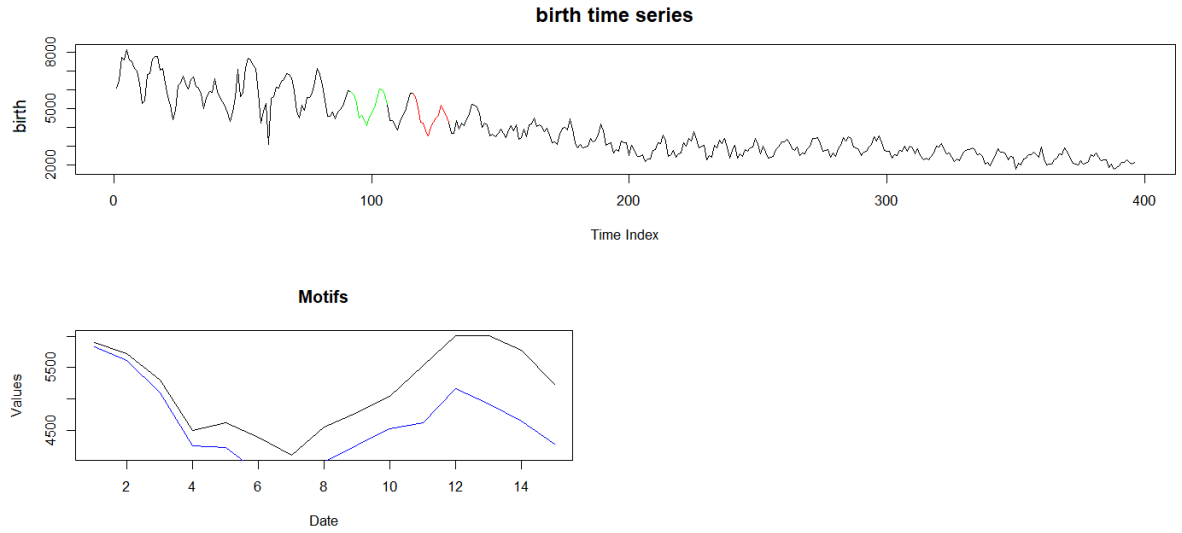


Figura 3.6. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

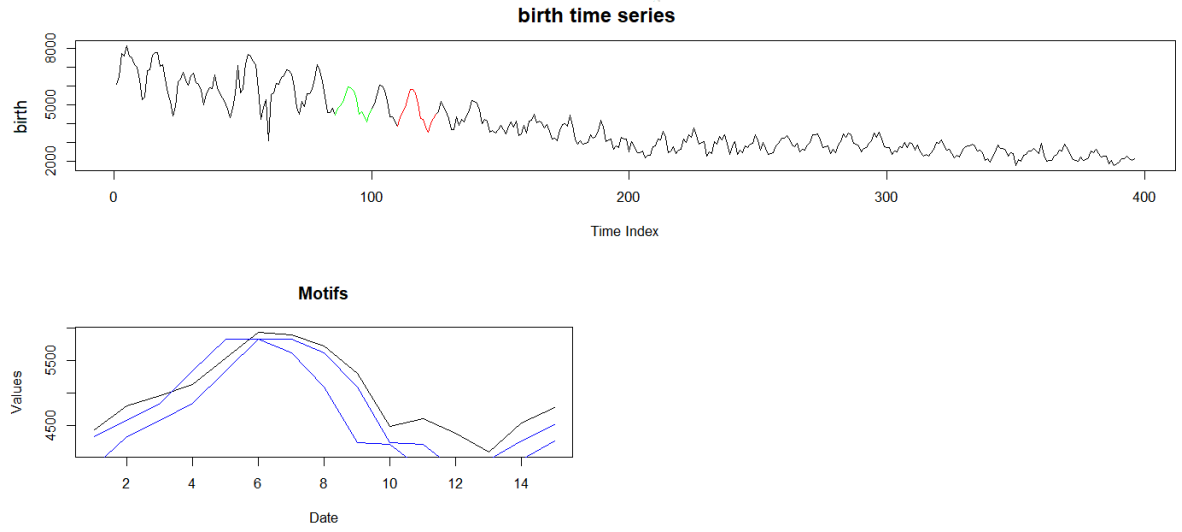


Figura 3.7. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

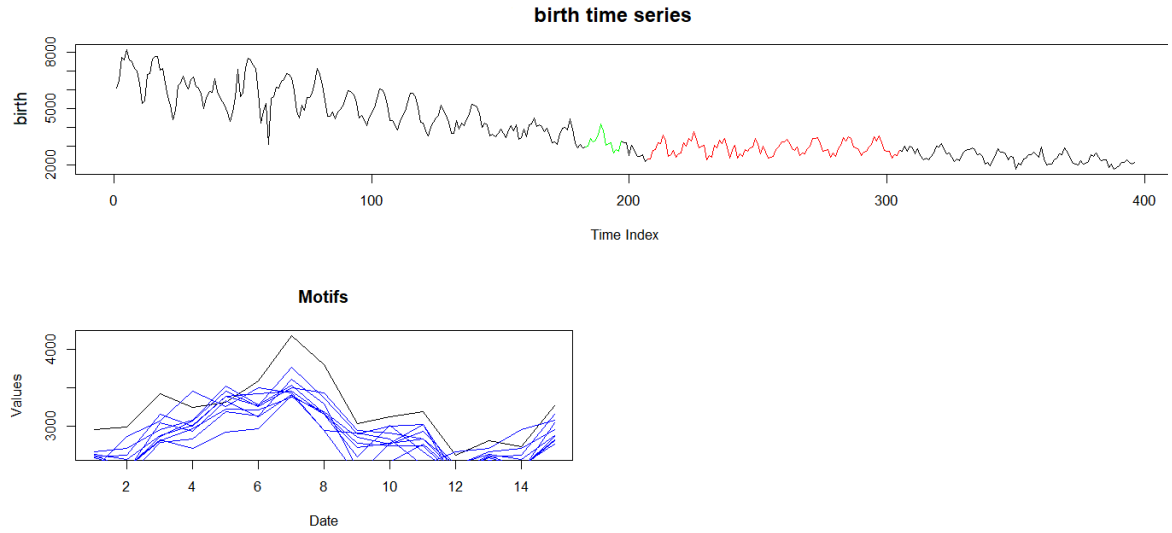


Figura 3.8. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

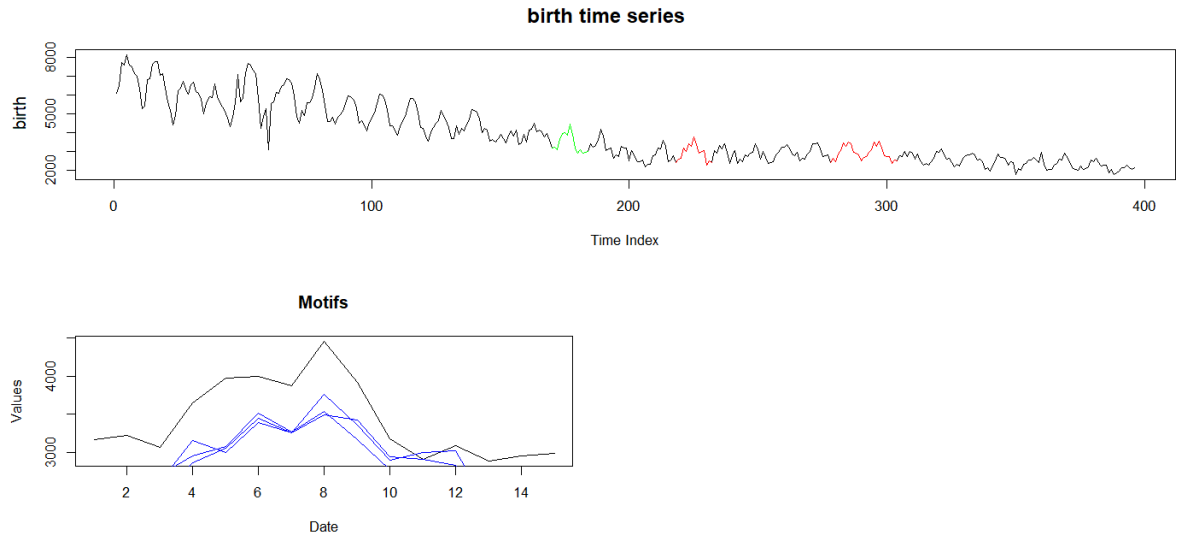


Figura 3.9. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

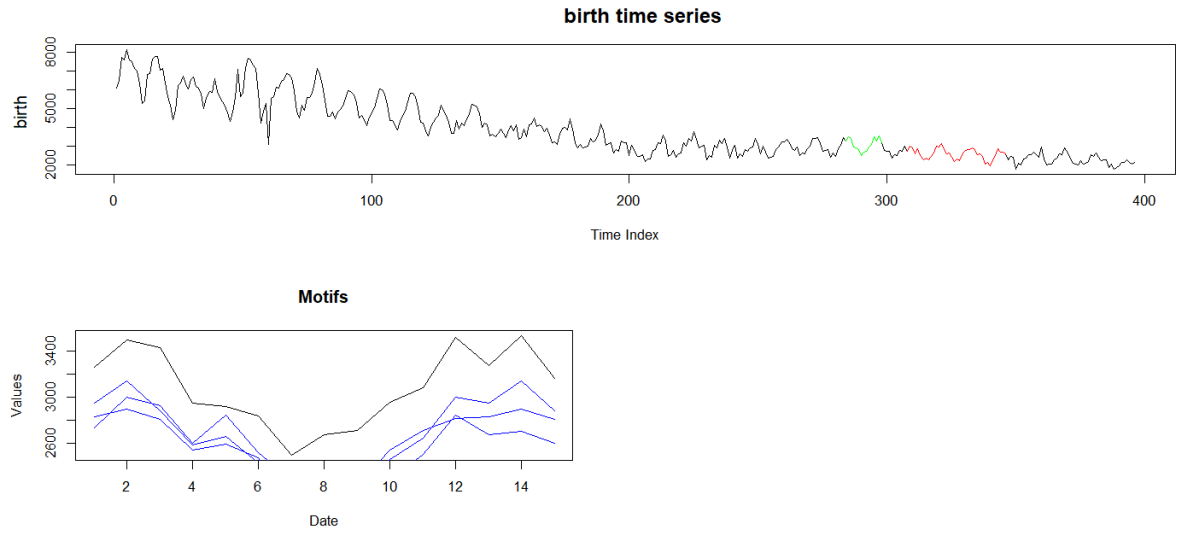


Figura 4.0. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

Motivet e gjetura më pare, i kemi gjetur për epsilon_query sa shmangia mesatare katrore tani do ta rrisim epsilon_querin në mënyrë që kur të filtrojmë vlerat nga rreshti i i-të të marrim vlera edhe më të mëdha se ato të gjetura më parë. Me fjalë të tjera po kërkojmë të jemi më tolerant dhe të përzgjedhim edhe sekuenca që janë më larg nga njëra tjetra (më pak të ngjashme).

Për epsilon-query sa dyfishi i shmangies mesatare katrore ne gjejmë motivet të cilat i inspektojmë manualisht pasi ka edhe nënsekuenca që plotesojnë kushtin që distanca e euklidit është më e vogël se epsilon-query por që nuk kanë pattern të ngjashme. Pasi bëjmë inspektimin e grafikëve që numerikisht janë të ngjashëm, manualisht po i paraqesim motivet e gjetura në grafikun original në menyrë që rezultati i gjetur të jetë më intuitive/kuptueshëm.

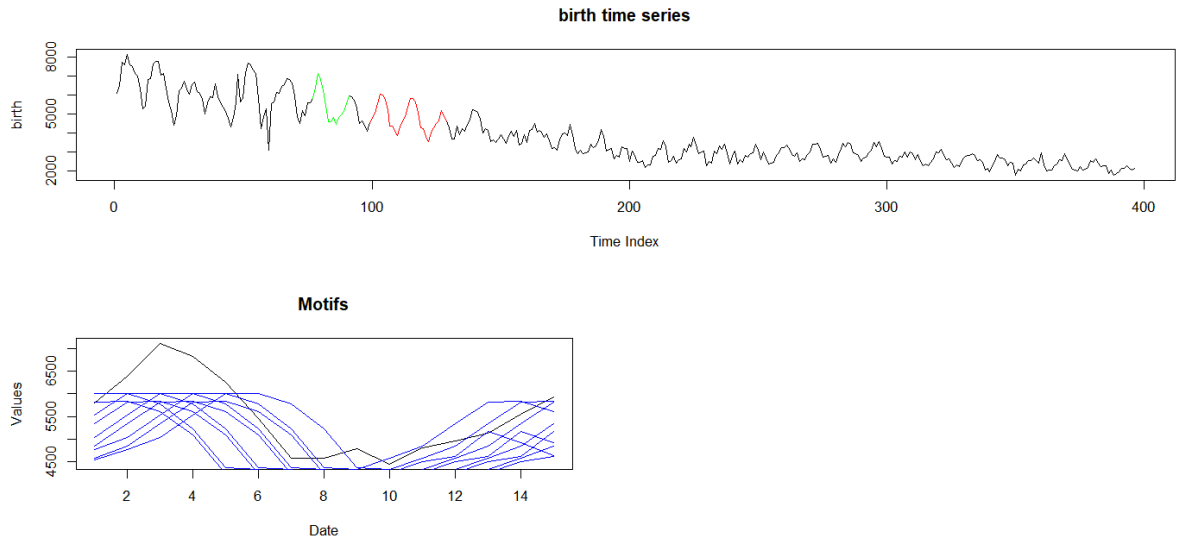


Figura 4.1. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

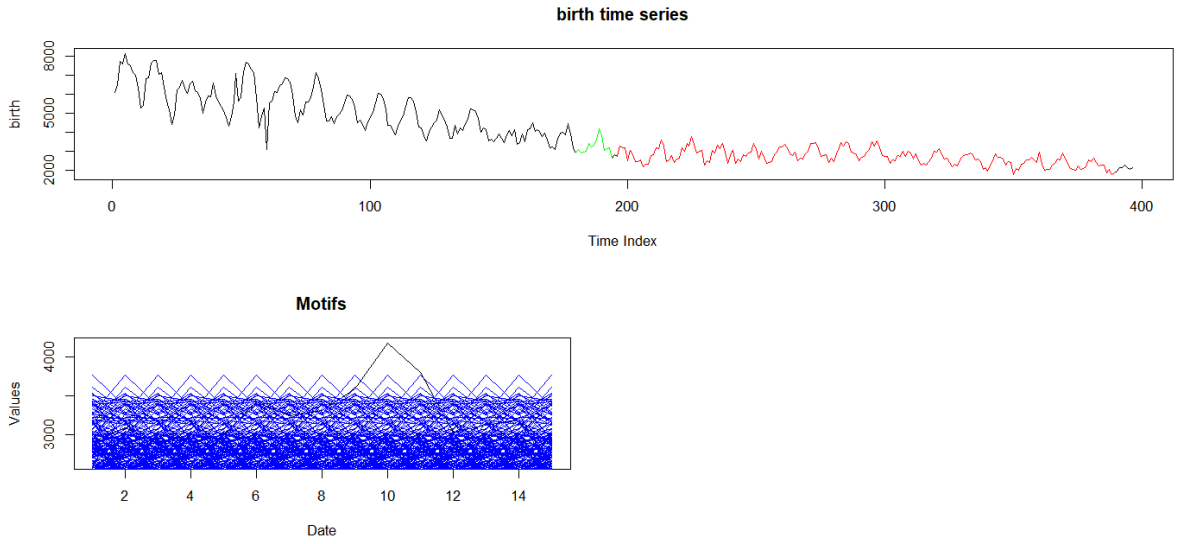


Figura 4.2. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

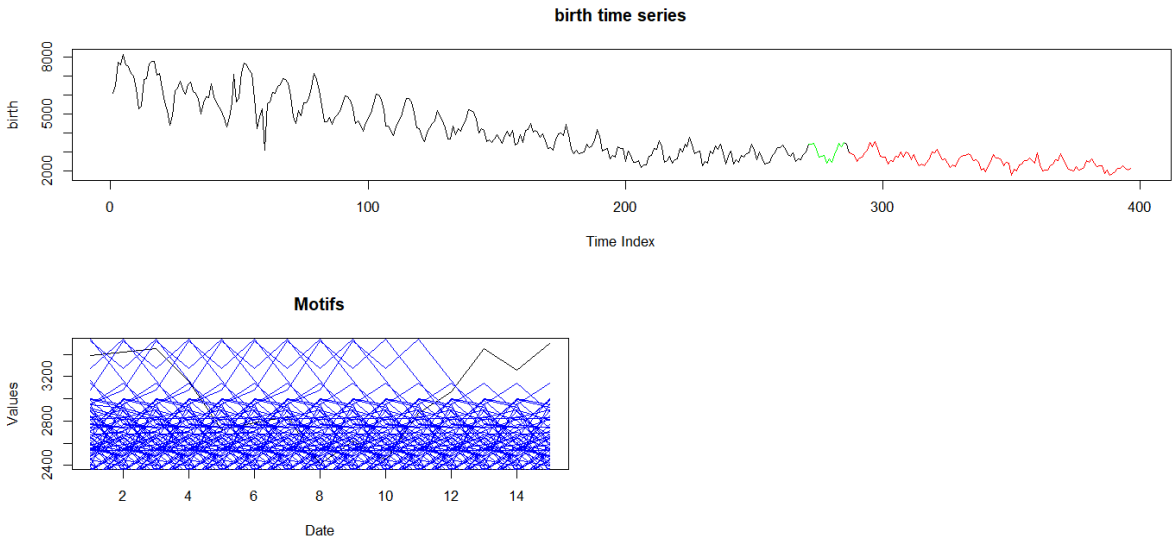


Figura 4.3. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

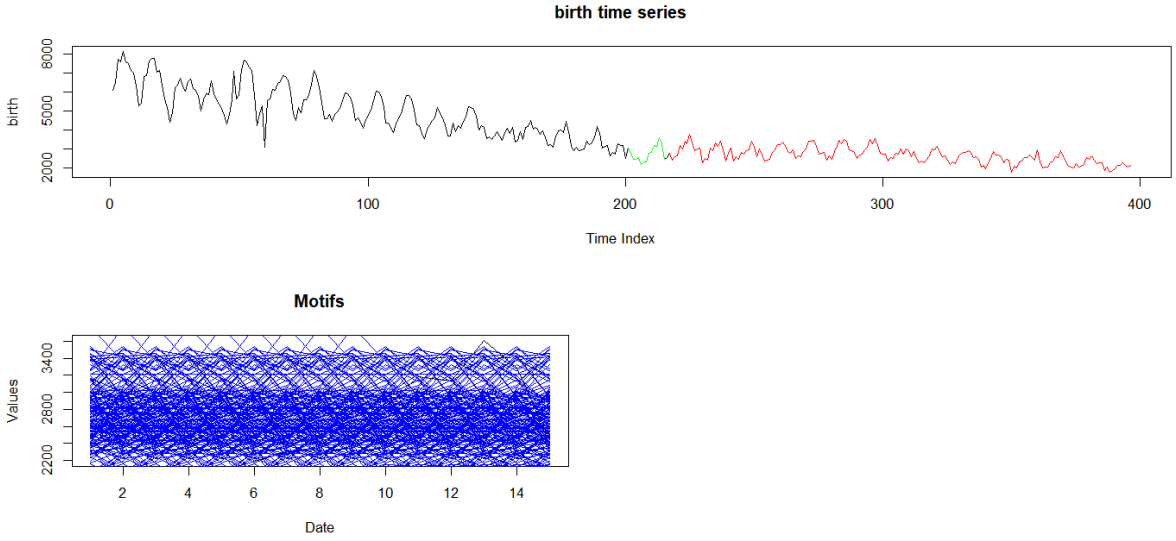


Figura 4.4. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

Për epsilon-query sa dyfishi i shmangies mesatare katrore ne gjejmë motivet të cilat i inspektojmë manualisht pasi ka edhe nënsekuenca që plotesojnë kushtin që CID është më e vogël se epsilon-query por që nuk kanë pattern të ngjashme. Pasi bëjmë inspektimin e grafikëve që numerikisht janë të ngjashëm, manualisht po i paraqesim motivet e gjetura në grafikun

original në menyrë që rezultati i gjetur të jetë më intuitive/kuptueshem.

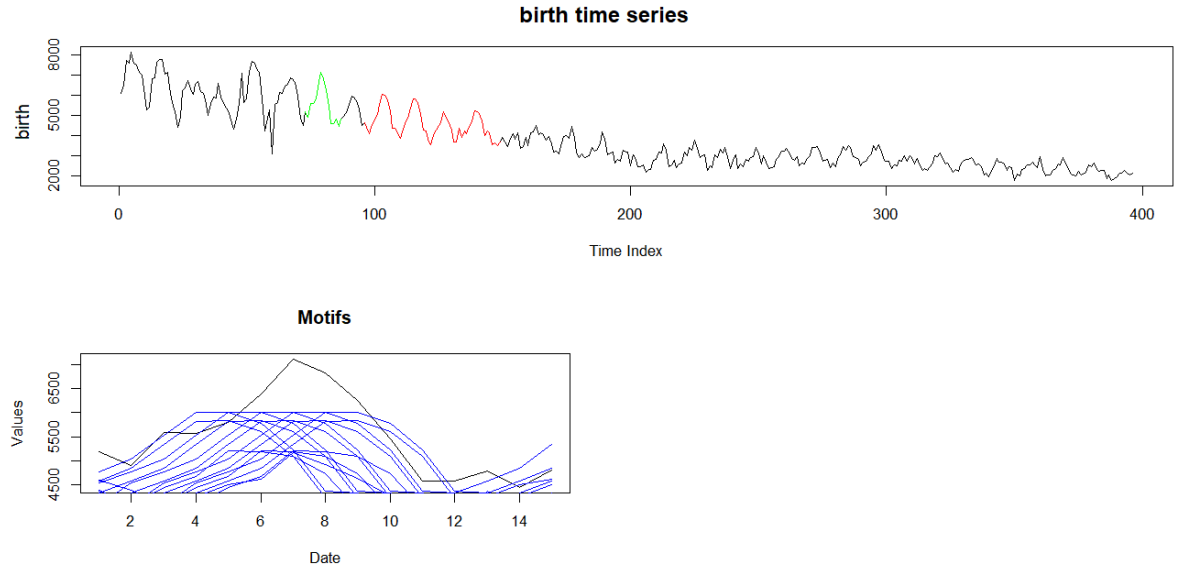


Figura 4.4. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

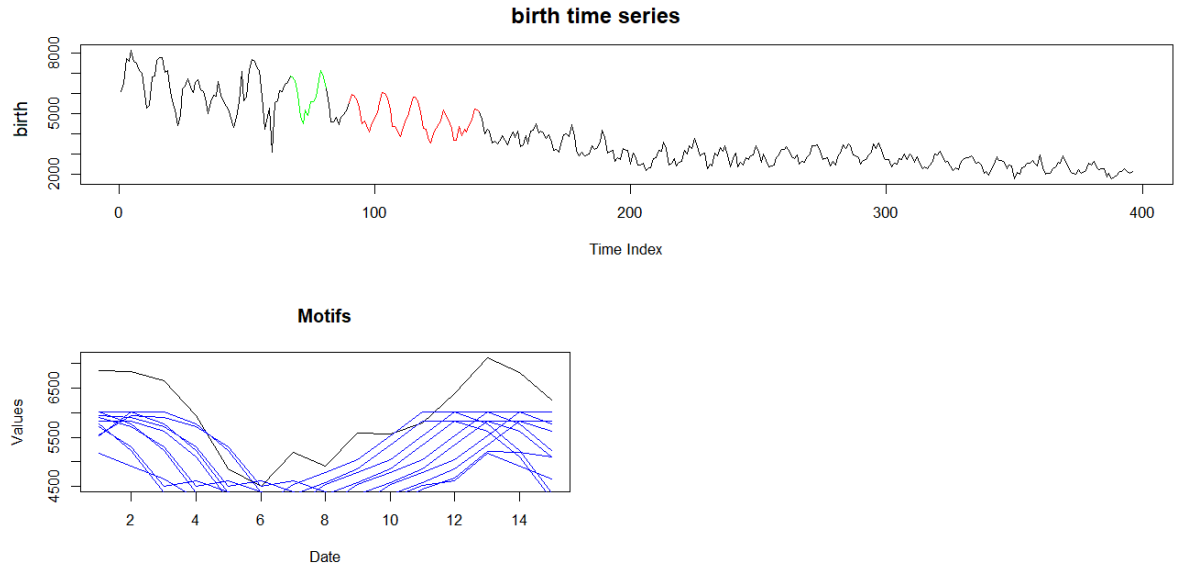


Figura 4.5. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

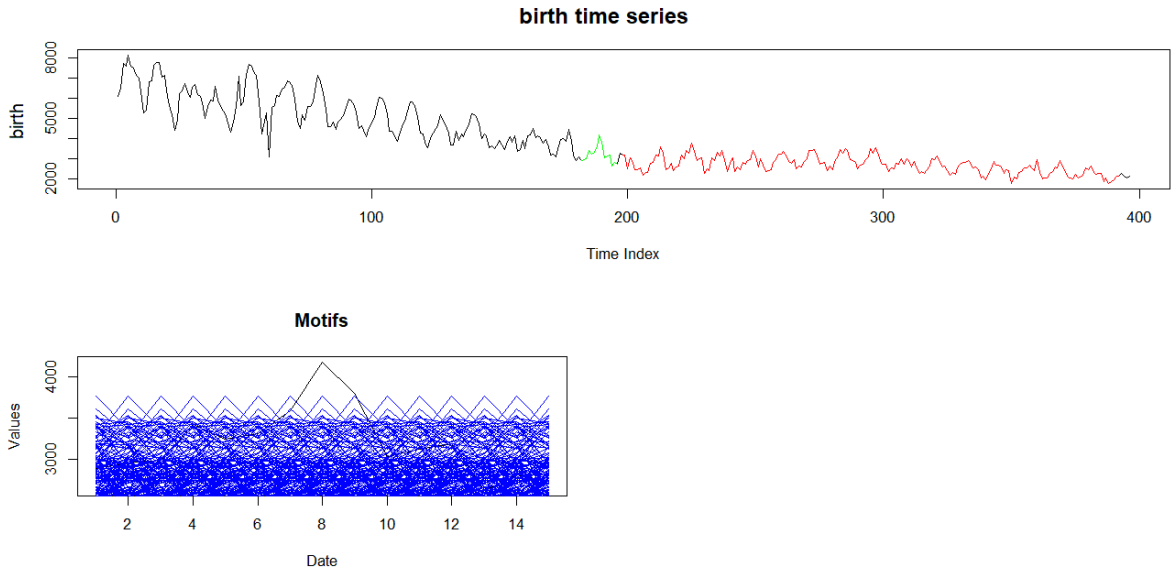


Figura 4.6. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

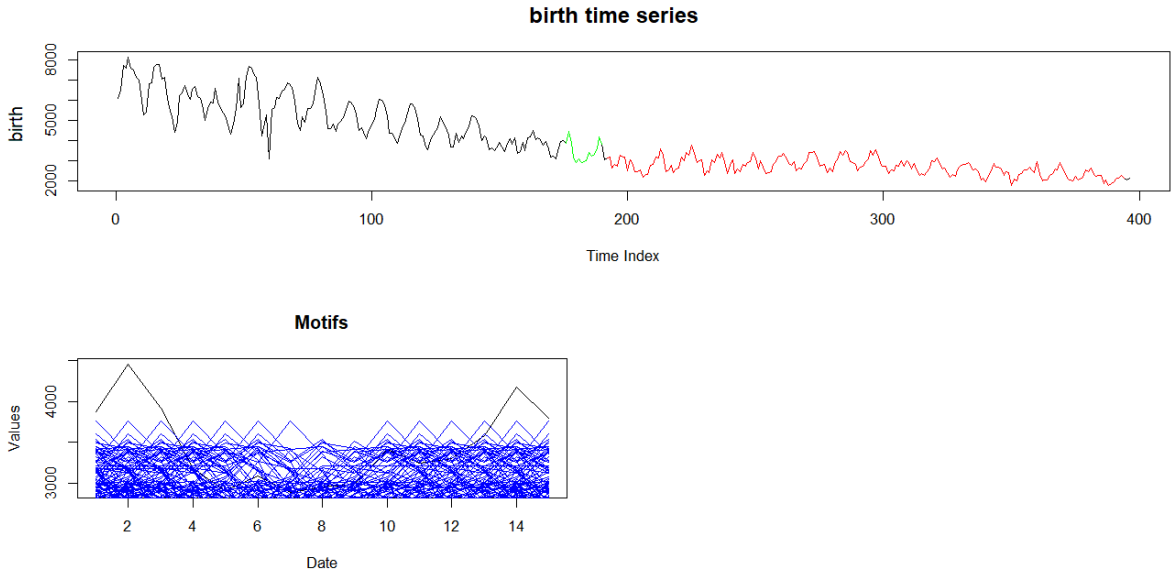


Figura 4.7. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

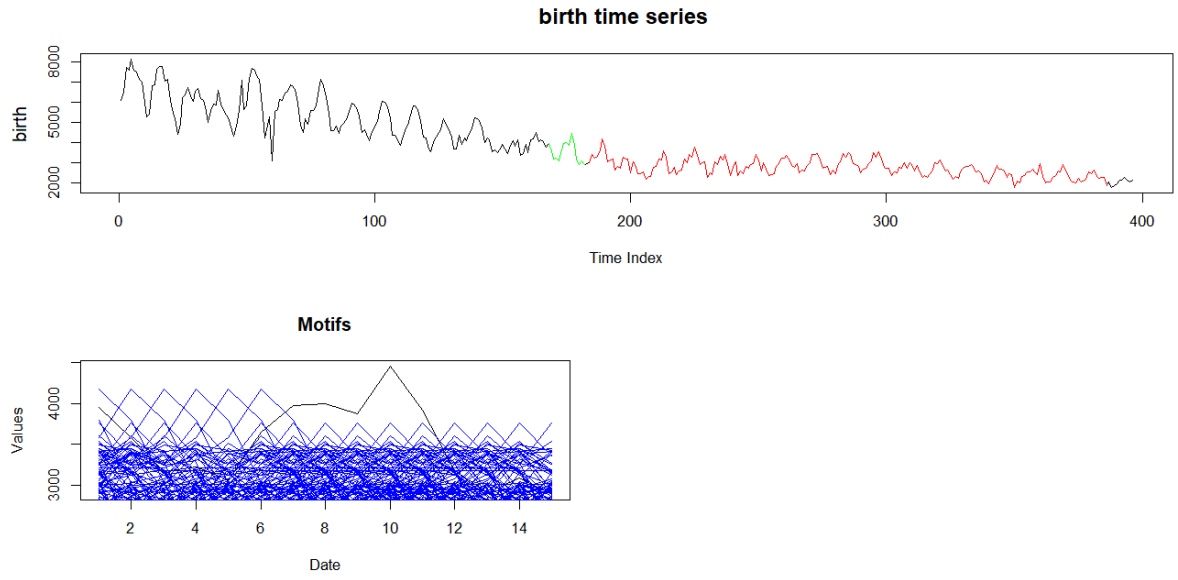


Figura 4.8. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

Nga sa pamë më sipër tentuam të rrisim epsilon-querin në mënyrë që të kapim/zbulojmë më shumë mortive por ndodhi e kundërta numri i tyre u ul. Motive që më parë i kishim klasifikuar si të ndara me rritjen e epsilon-querit u shkrinë në një të vetme. A është kjo një rast i vecantë? Për ta zbuluar vazhdojmë të kërkojmë për motive duke e rritur përsëri epsilon-queri dhe zbulojmë që numri nuk ndryshon ose ndryshon me 1. Për arsye që të mor ngarkohet me grafikë sepse tashmë ideja se si do të procedojmë është e qartë nuk po e vizualizojme këtë pjesë. Kjo ishte mënyra se si proceduam për serinë e lindjeve dhe duhet theksuar se kjo seri ka një trend zbritës i cili dallohet dhe me sy të lirë. Lind pyetja nëse algoritmi do të arrijë që t'i zbulojë me sukses motivet dhe në një seri periodike (apo sezonale)? Për këtë arsye studiojmë se si do të sillet algoritmi në një seri kohore të tillë. Më konkretisht do të procedojmë me serinë kohore që na paraqet ndryshimin e temperaturave ndër vite.

Fillimisht vizualizojmë informacionin për të krijuar një ide paraprake.

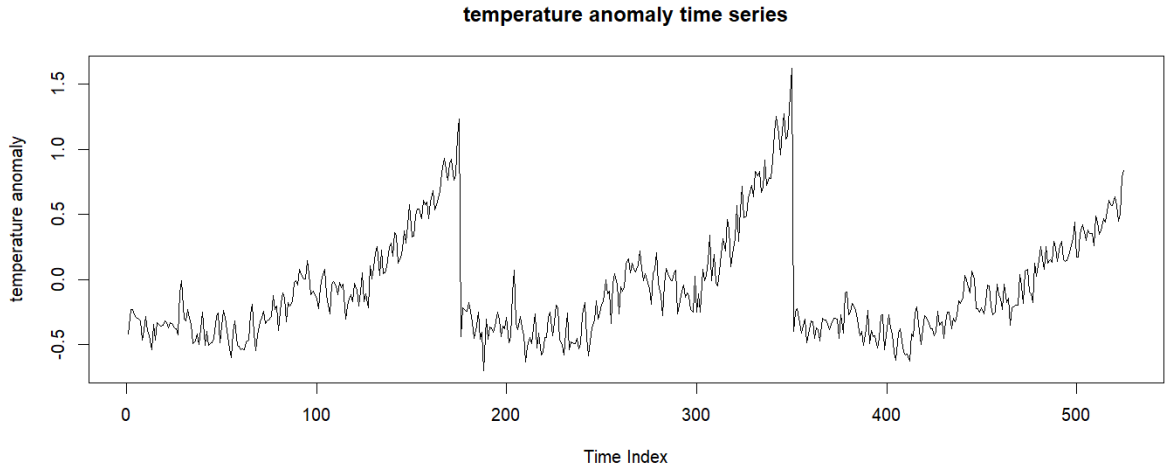


Figura 4.9. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

Pasi e investigojmë paraprakisht grafikun me sy të lirë mund të dallojmë ndonjë motiv por nuk jemi të sigurt prandaj duhet të përdorim CID dhe distancën Euklidit për t'i zbuluar. Zbatojmë algoritmin BruteForce (të cilin e kemi ndërtuar më pare kodi i të cilit do të paraqitet në kapitullin pasardhës i shpjeguar) ku i japim si input serinë, 1 vlerë numerike që përfaqëson gjatësinë e dritares lëvizëse dhe fillimisht i japim vlerën Euklidian për të zbatuar distancen e euklidit. Algoritmi na kthen një matricë trekendeshe ku rreshti i i-të mban distancën e sekuencës i me dritaren lëizese në indexin e j. Hapi tjetër është filtrimi i vlerave të rreshtit të i-të. Domethënë do të marrim ato vlera të rreshtit të i-të të matricës që janë më të vogla se epsilon-query i përcaktuar më pare nga ne.

Për epsilon-query sa shmangia mesatare katrore ne gjejmë motivet të cilat i inspektojmë manualisht pasi ka edhe nënsekuenca që plotesojnë kushtin që distanca e euklidit është më e vogël se epsilon-query por që nuk kanë pattern të ngjashme.

Pasi bëjmë inspektimin e grafikëve që numerikisht janë të ngjashëm, manualisht po i paraqesim motivet e gjetura në grafikun original në menyrë që rezultati i gjetur të jetë më intuitive/kuptueshem.

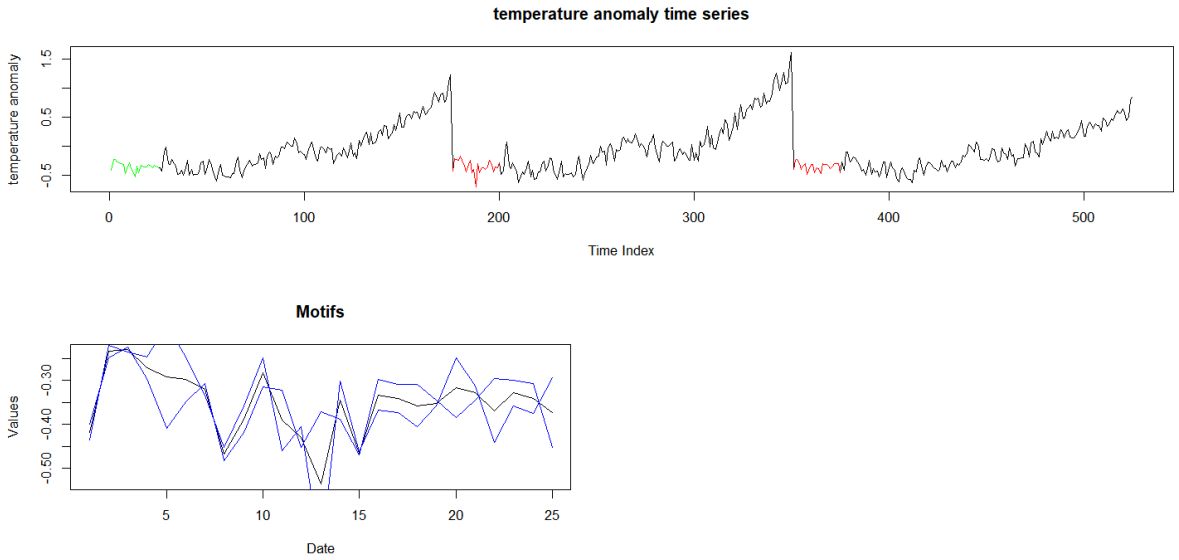


Figura 5.0. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

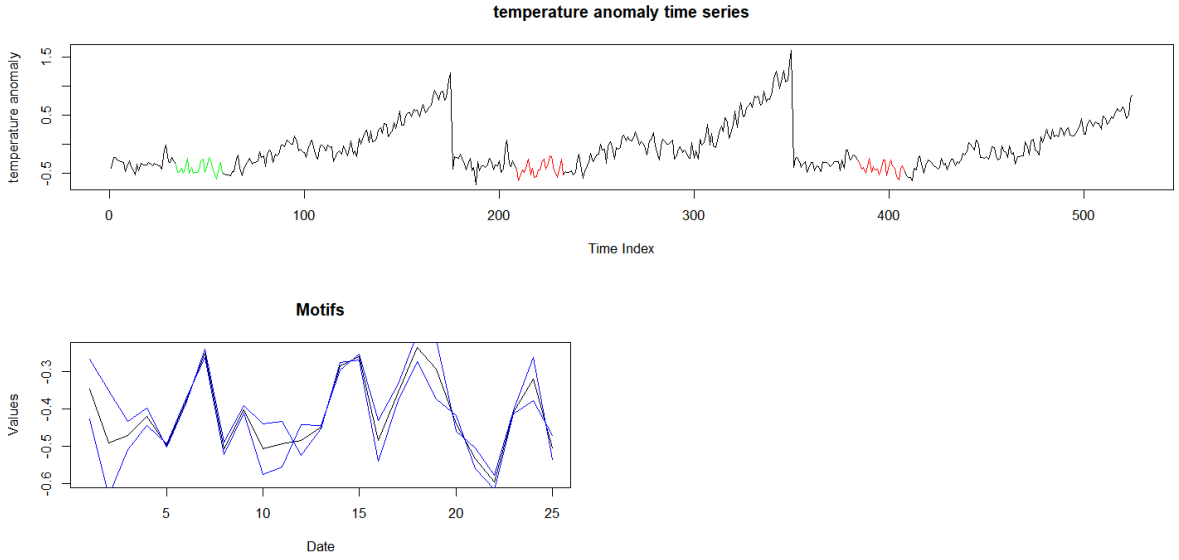


Figura 5.1. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

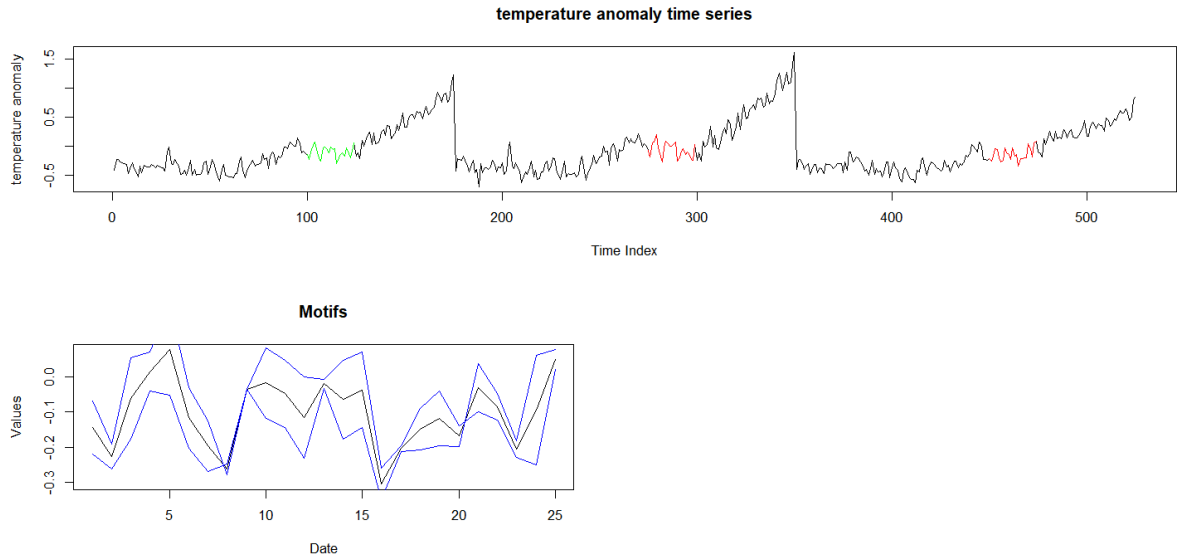


Figura 5.2. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

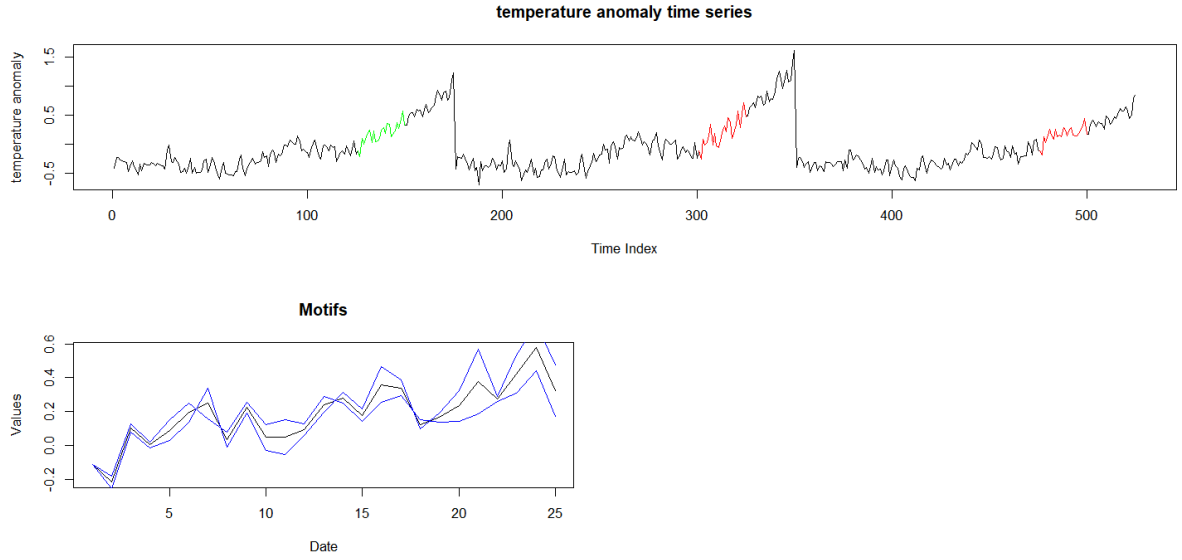


Figura 5.3. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

Zbatojmë algoritmin BruteForce ku i japim si input serinë, 1 vlerë numerike që përfaqëson gjatësinë e dritares lëvizëse dhe i japim vlerën chouakria për të zbatuar CID. Algoritmi na kthen një matricë trekendeshe ku rreshti i i-të mban distancën e sekuencës i me dritaren lëizese në indexin e j. Hapi tjetër është filtrimi i vlerave të rreshtit të i-të. Domethënë do të marrim ato vlera të rreshtit të i-të të matricës që janë më të vogla se epsilon-query i përcaktuar më pare nga ne.

Për epsilon-query sa shmangia mesatare katrore ne gjejmë motivet të cilat i inspektojmë manualisht pasi ka edhe nënsekuenca që plotesojnë kushtin që CID është më e vogël se epsilon-query por që nuk kanë pattern të ngjashme. Pasi bëjmë inspektimin e grafikëve që numerikisht janë të ngjashëm, manualisht po i paraqesim motivet e gjetura në grafikun original në menyrë që rezultati i gjetur të jetë më intuitive/kuptueshem.

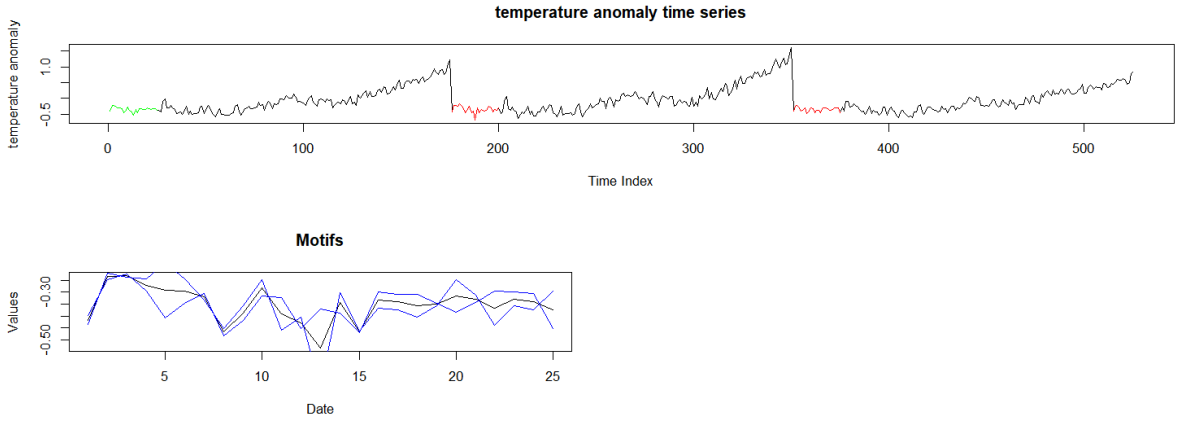


Figura 5.4. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

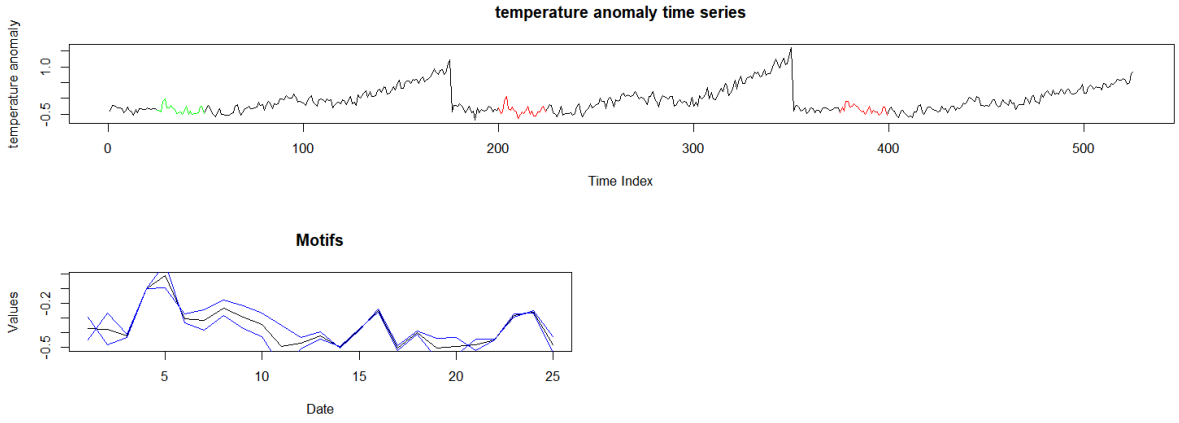


Figura 5.5. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

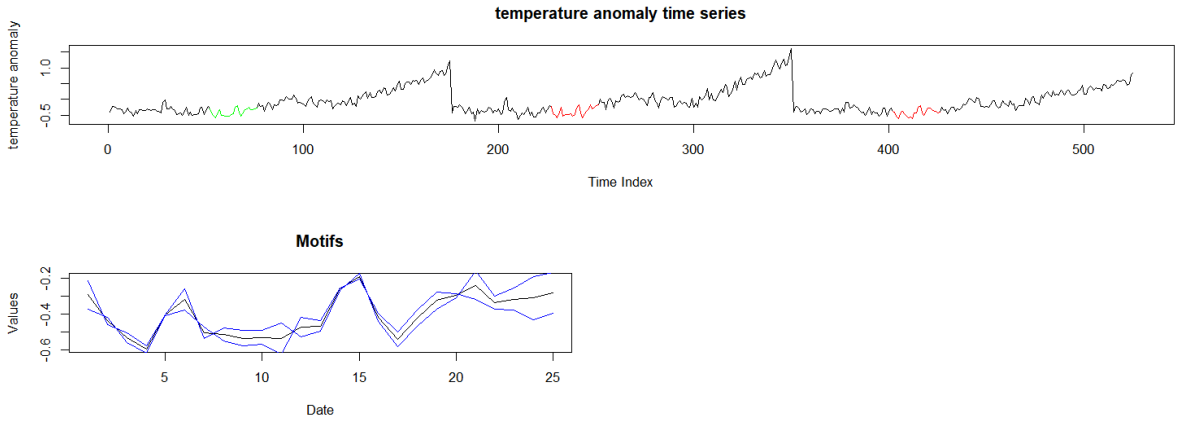


Figura 5.6. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

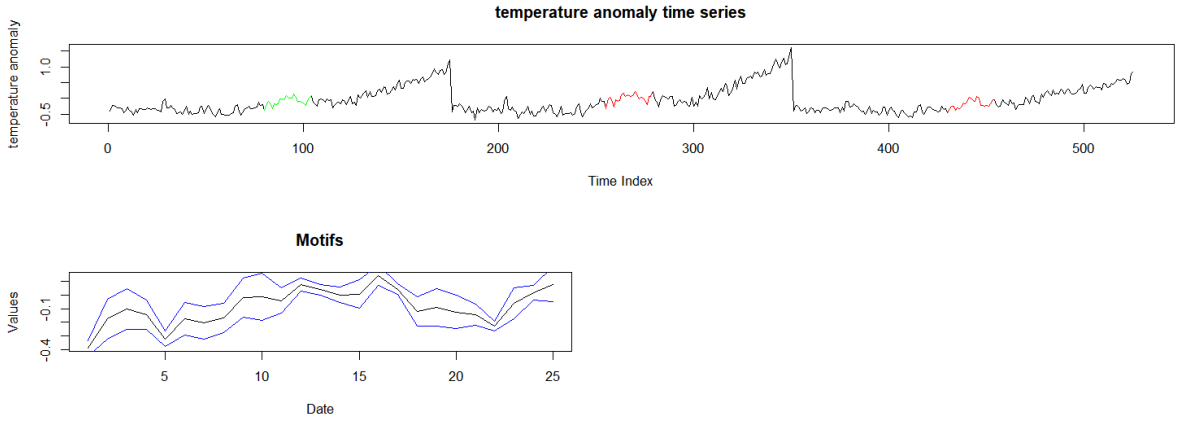


Figura 5.7. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

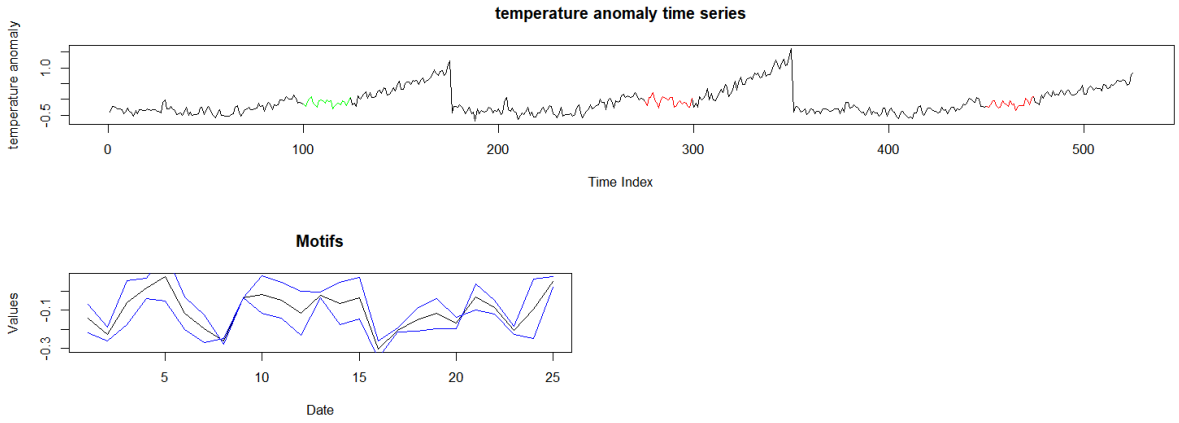


Figura 5.8. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

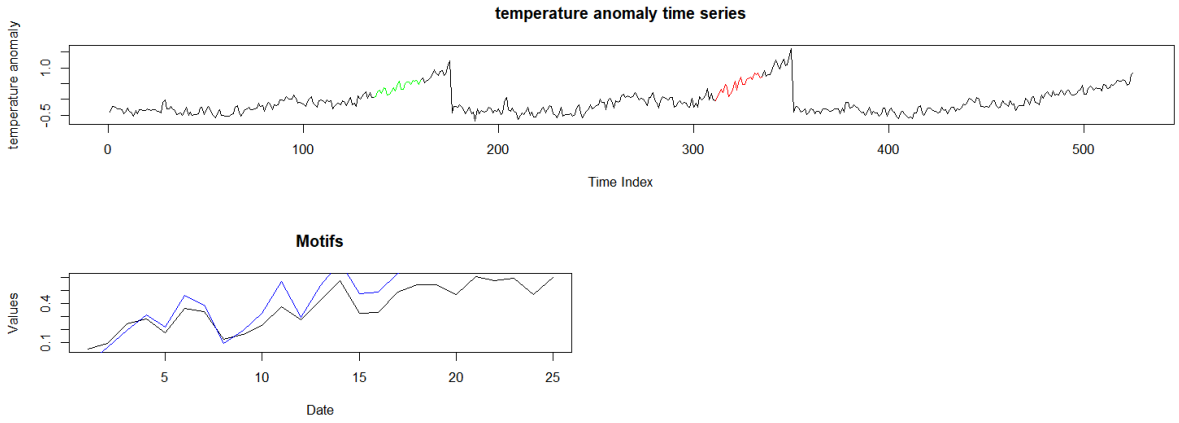


Figura 5.9. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

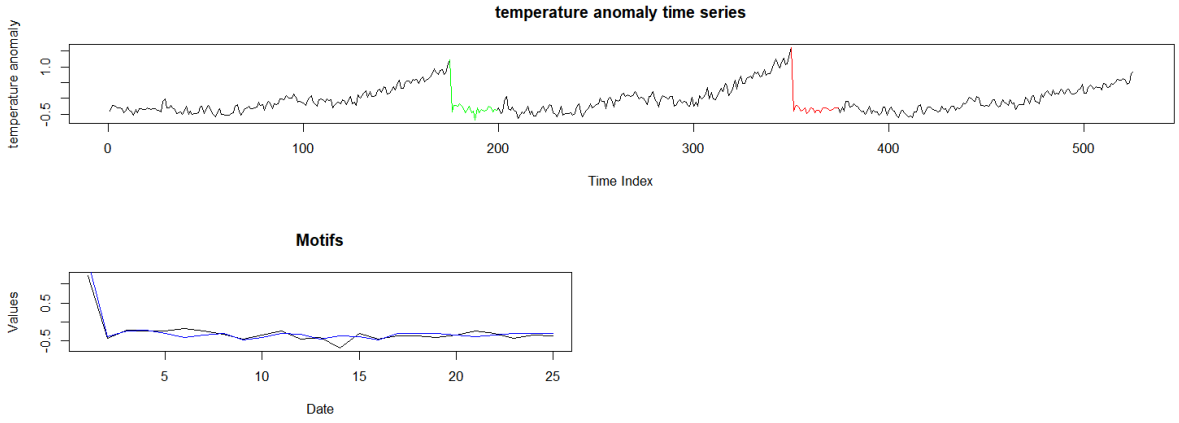


Figura 6.0. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

Motivet e gjetura më parë, i kemi gjetur për epsilon_query sa shmangia mesatare katrore tani do ta rrisim epsilon_querin në mënyrë që kur të filtrojmë vlerat nga rreshti i i-të të marrim vlera edhe më të mëdha se ato të gjetura më parë. Me fjalë të tjera po kërkojmë të jemi më tolerant dhe të përzgjedhim edhe sekuenca që janë më larg nga njëra tjetra (më pak të ngjashme).

Për epsilon-query sa dyfishi i shmangies mesatare katrore ne gjejmë motivet të cilat i inspektojmë manualisht pasi ka edhe nënsekuenca që plotesojnë kushtin që distanca e euklidit është më e vogël se epsilon-query por që nuk kanë pattern të ngjashme. Pasi bëjmë inspektimin e grafikëve që numerikisht janë të ngjashëm, manualisht po i paraqesim motivet e gjetura në grafikun original në menyrë që rezultati i gjetur të jetë më intuitive/kuptueshëm

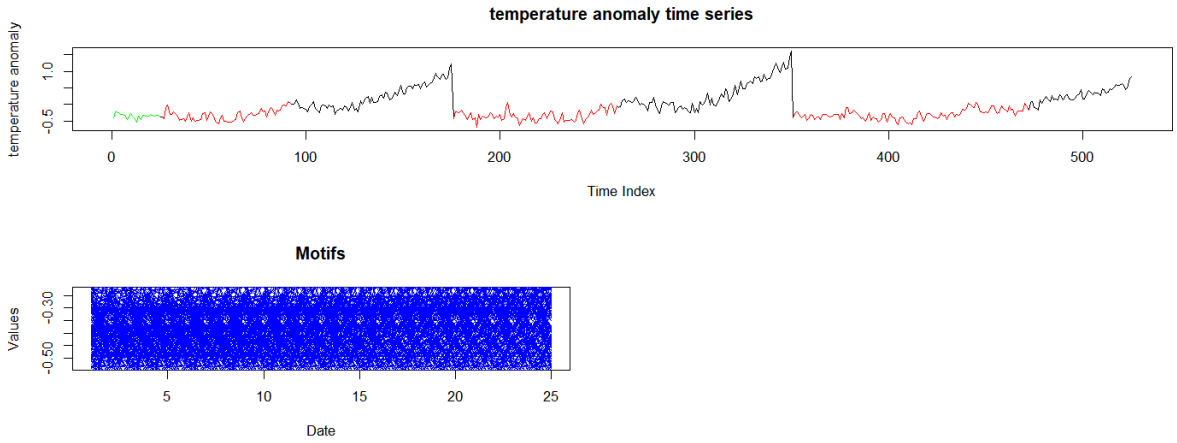


Figura 6.1. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

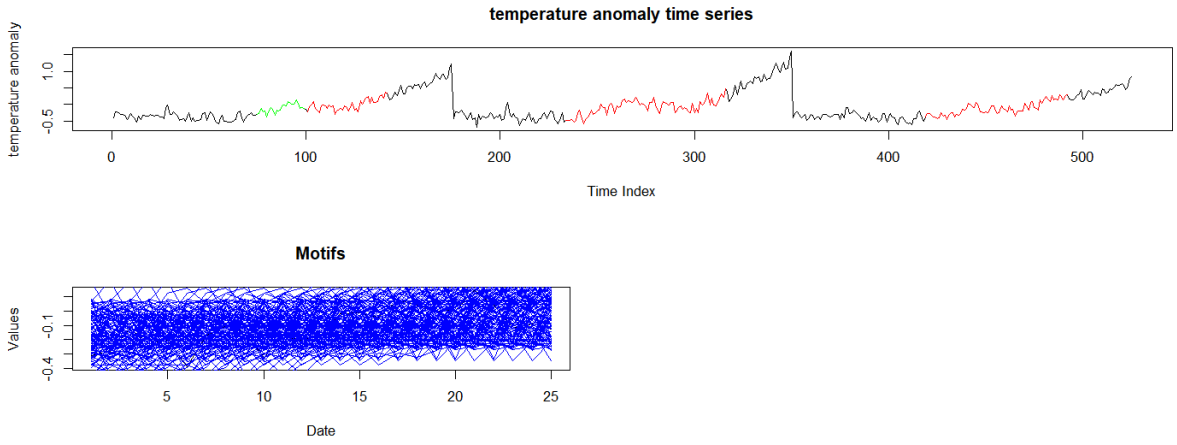


Figura 6.2. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

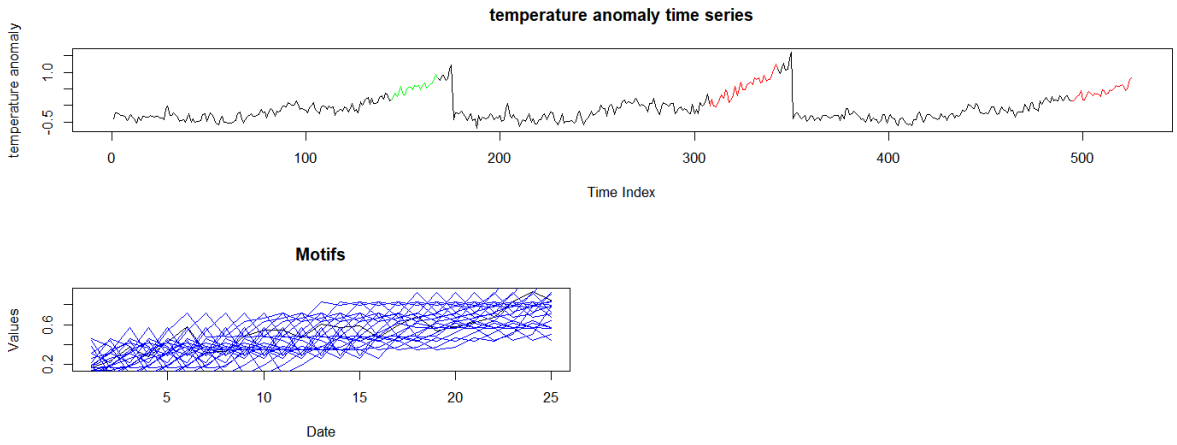


Figura 6.3. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

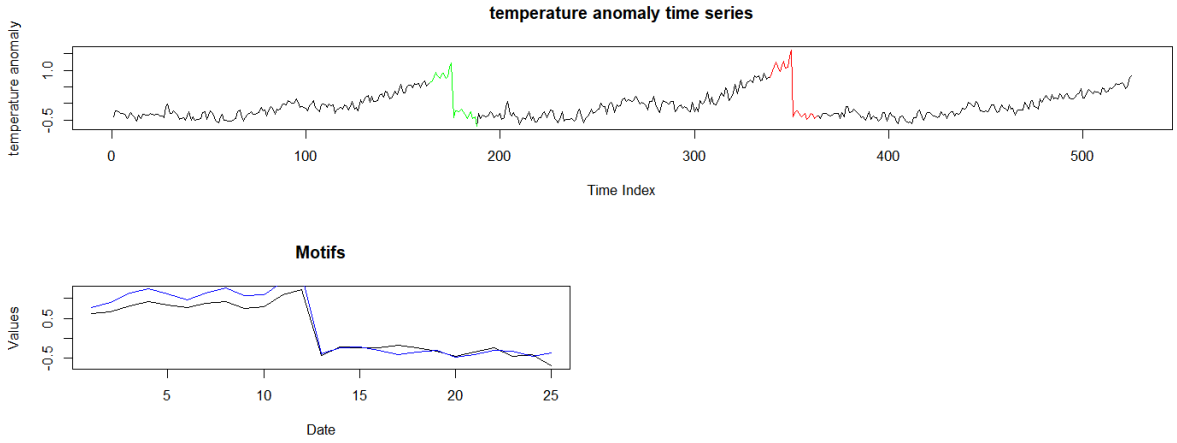


Figura 6.4. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

Për epsilon-query sa dyfishi i shmangies mesatare katrore ne gjejmë motivet të cilat i inspektojmë manualisht pasi ka edhe nënsekuenca që plotesojnë kushtin që CID është më e vogël se epsilon-query por që nuk kanë pattern të ngjashme. Pasi bëjmë inspektimin e grafikëve që numerikisht janë të ngjashëm, manualisht po i paraqesim motivet e gjetura në grafikun original në menyrë që rezultati i gjetur të jetë më intuitive/kuptueshem.

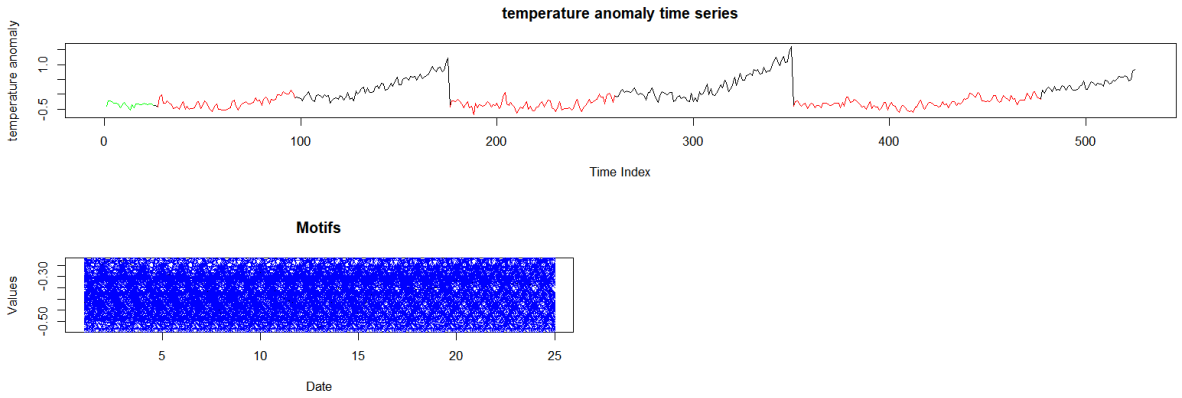


Figura 6.5. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

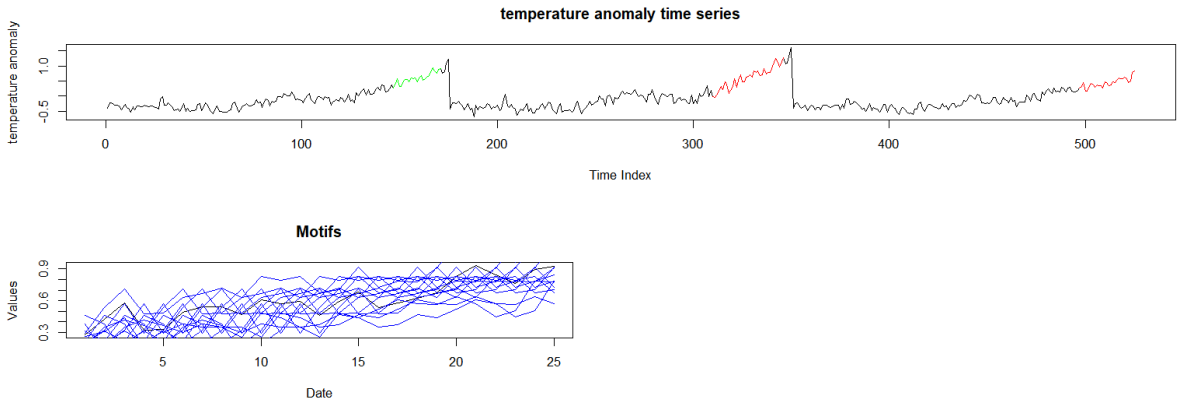


Figura 6.6. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

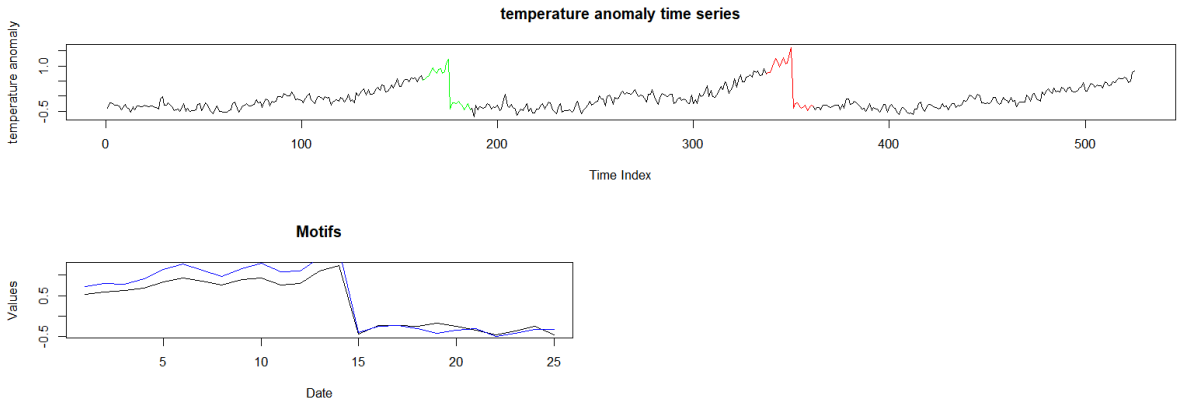


Figura 6.7. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

Nga sa pamë më sipër tentuam të rrisim epsilon-querin në mënyrë që të kapim/zbulojmë më

shumë mortive por ndodhi e kundërta numri i tyre u ul ose nuk ndryshoj. A është kjo një rast i vecantë? Për ta zbuluar vazhdojmë të kërkojmë për motive duke e rritur përsëri epsilon-queri dhe zbulojmë që numri nuk ndryshon ose ndryshon me 1. Për arsye që të mos ngarkohet me grafikë sepse tashmë ideja se si do të procedojmë është e qartë nuk po e vizualizojme këtë pjesë. Kjo ishte mënyra se si proceduam për serinë e lindjeve dhe duhet theksuar se kjo seri ka një trend zbritës i cili dallohet dhe me sy të lirë. Vazhdojmë procedurën me një tjetër seri kohore, me serine e sasisë së CO2 të cliruar në shtetin e Shqipërisë.

Fillimisht vizualizojmë informacionin për të krijuar një ide paraprake.

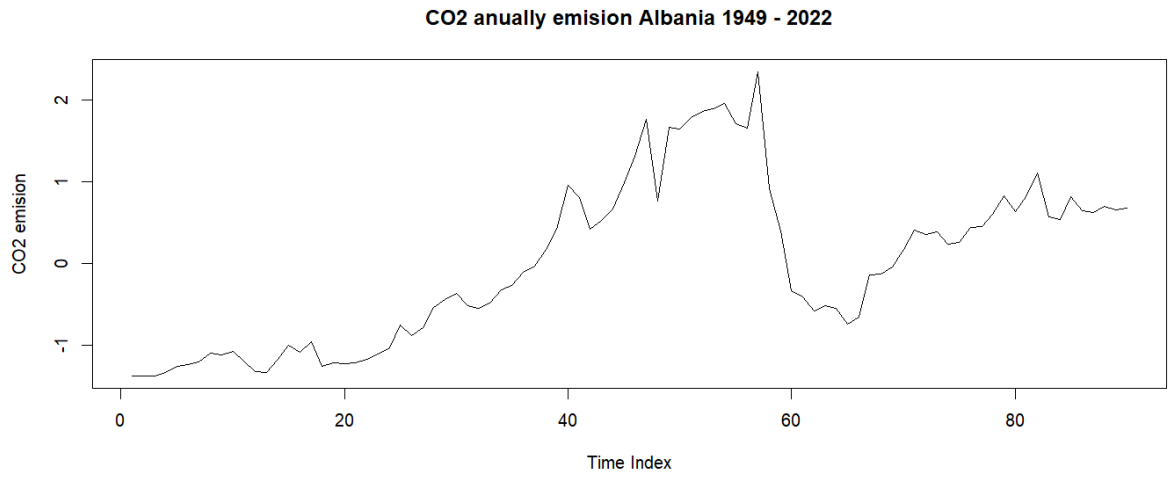


Figura 6.8. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

Pasi e investigojmë paraprakisht grafikun me sy të lirë mund të dallojmë ndonjë motiv por nuk jemi të sigurt prandaj duhet të përdorim CID dhe distancën Euklidit për t'i zbuluar. Zbatojmë algoritmin BruteForce (të cilin e kemi ndërtuar më pare kodi i të cilit do të paraqitet në kapitullin pasardhës i shpjeguar) ku i japim si input serine, 1 vlerë numerike që përfaqëson gjatësinë e dritares lëvizëse dhe fillimisht i japim vlerën Euklidian për të zbatuar distancen e euklidit. Algoritmi na kthen një matricë trekendeshe ku rreshti i i-të mban distancën e sekuencës i me dritaren lëizese në indexin e j. Hapi tjetër është filtrimi i vlerave të rreshtit të i-të. Domethënë do të marrim ato vlera të rreshtit të i-të të matricës që janë më të vogla se epsilon-query i përcaktuar më pare nga ne.

Për epsilon-query sa shmangia mesatare katrore ne gjejmë motivet të cilat i inspektojmë manualisht pasi ka edhe nënsekuenca që plotesojnë kushtin që distanca e euklidit është më e vogël se epsilon-query por që nuk kanë pattern të ngjashme.

Pasi bëjmë inspektimin e grafikëve që numerikisht janë të ngjashëm, manualisht po i paraqesim motivet e gjetura në grafikun original në menyrë që rezultati i gjetur të jetë më intuitive/kuptueshem.

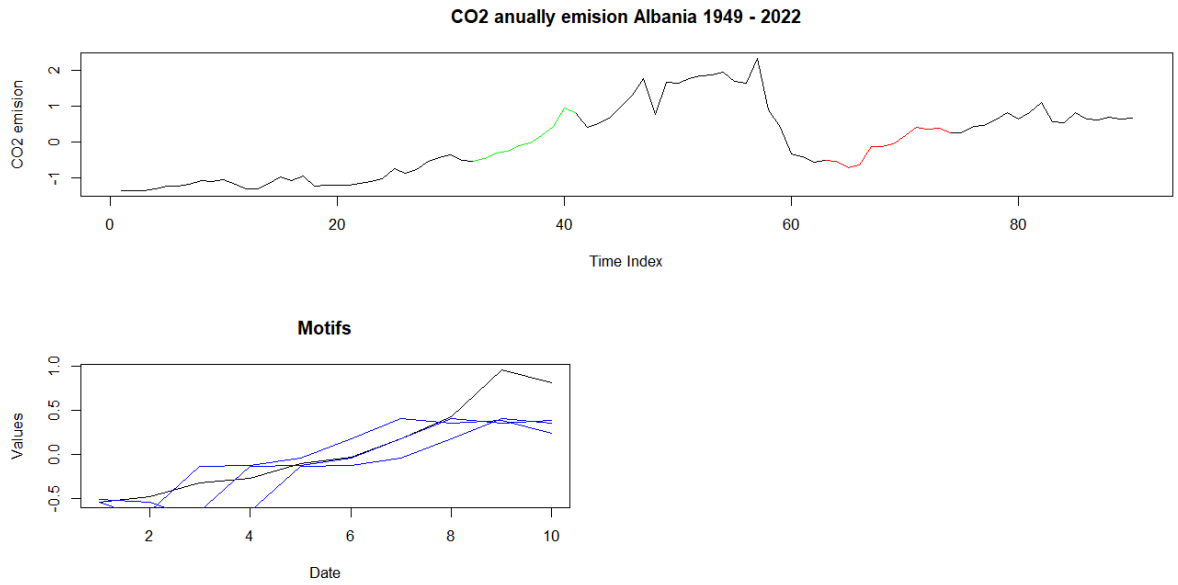


Figura 6.9. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

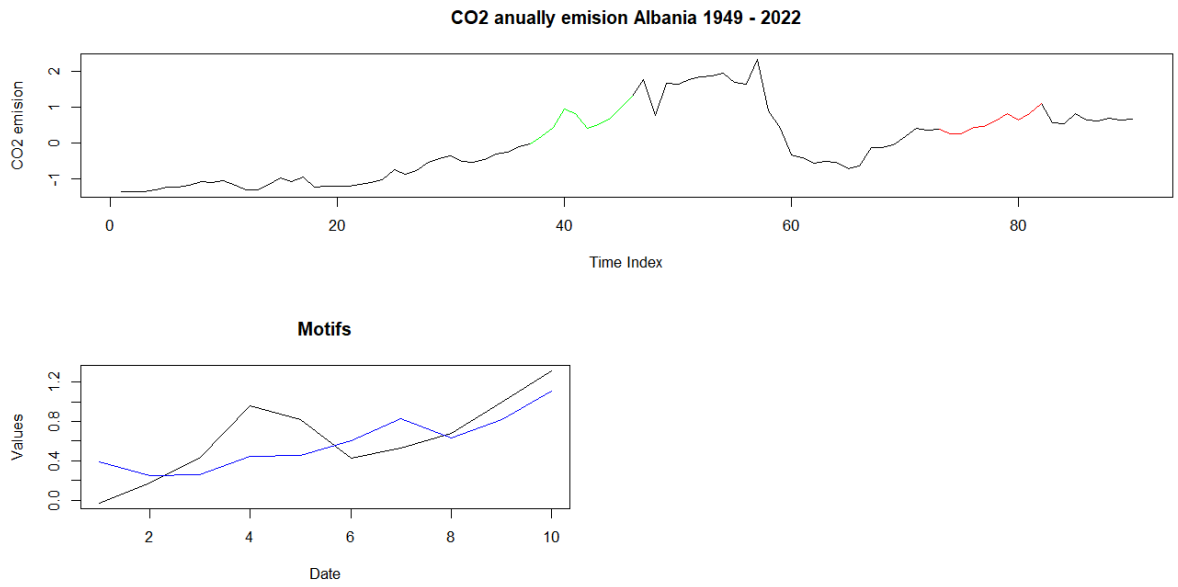


Figura 7.0. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

Zbatojmë algoritmin BruteForce ku i japim si input serine, 1 vlerë numerike që përfaqëson gjatësinë e dritares lëvizëse dhe i japim vlerën chouakria për të zbatuar CID. Algoritmi na kthen një matricë trekendeshe ku rreshti i i-të mban distancën e sekuencës i me dritaren lëizese në indexin e j. Hapi tjetër është filtrimi i vlerave të rreshtit të i-të. Domethënë do të marrim ato vlera të rreshtit të i-të të matricës që janë më të vogla se epsilon-query i përcaktuar më parë nga ne.

Për epsilon-query sa shmangia mesatare katrore ne gjejmë motivet të cilat i inspektojmë manualisht pasi ka edhe nënsekuenca që plotesojnë kushtin që CID është më e vogël se epsilon-query por që nuk kanë pattern të ngjashme. Pasi bëjmë inspektimin e grafikëve që numerikisht janë të ngjashëm, manualisht po i paraqesim motivet e gjetura në grafikun original në menyrë që rezultati i gjetur të jetë më intuitive/kuptueshem.

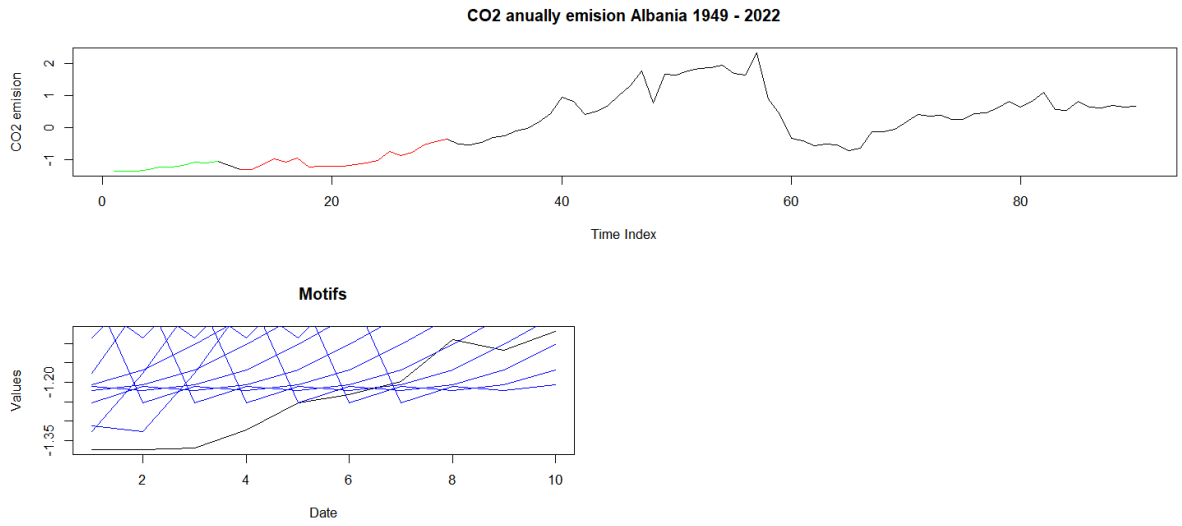


Figura 7.1. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

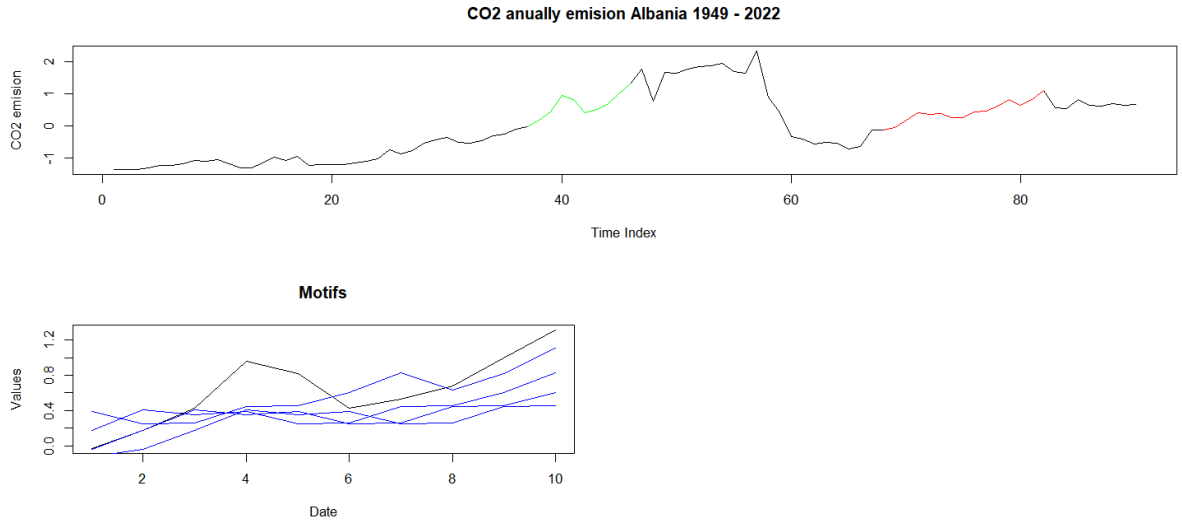


Figura 7.2. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

Motivet e gjetura më parë, i kemi gjetur për epsilon_query sa shmangia mesatare katrore tani do ta rrisim epsilon_querin në mënyrë që kur të filtrojmë vlerat nga rreshti i i-të të marrim vlera edhe më të mëdha se ato të gjetura më parë. Me fjalë të tjera po kërkojmë të jemi më tolerant dhe të përzgjedhim edhe sekuenca që janë më larg nga njëra tjetra (më pak të ngjashme).

Për epsilon-query sa dyfishi i shmangies mesatare katrore ne gjejmë motivet të cilat i inspektojmë manualisht pasi ka edhe nënsekuenca që plotesojnë kushtin që distanca e euklidit

është më e vogël se epsilon-query por që nuk kanë pattern të ngjashme. Pasi bëjmë inspektimin e grafikëve që numerikisht janë të ngjashëm, manualisht po i paraqesim motivet e gjetura në grafikun original në menyrë që rezultati i gjetur të jetë më intuitive/kuptueshëm

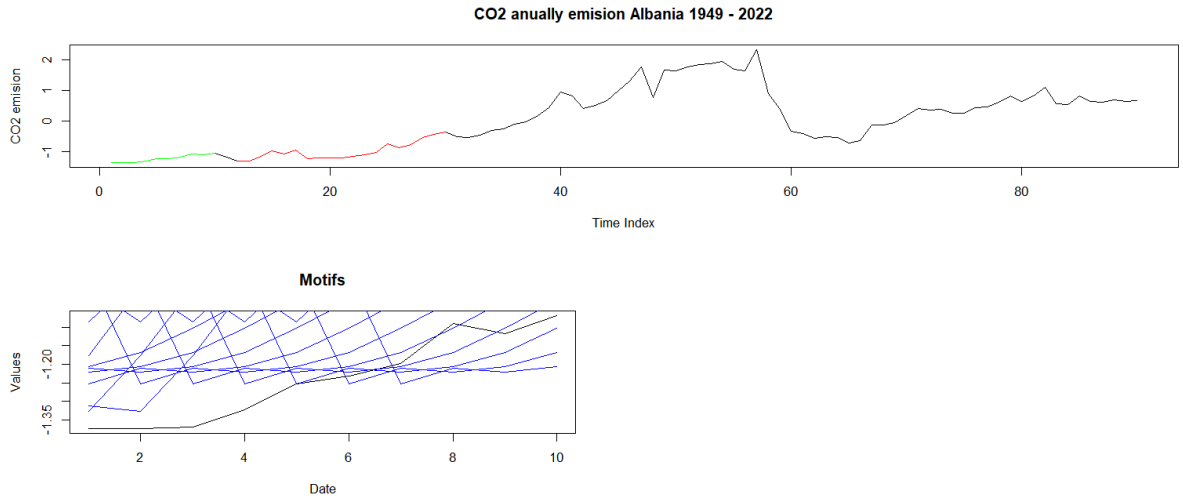


Figura 7.3. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

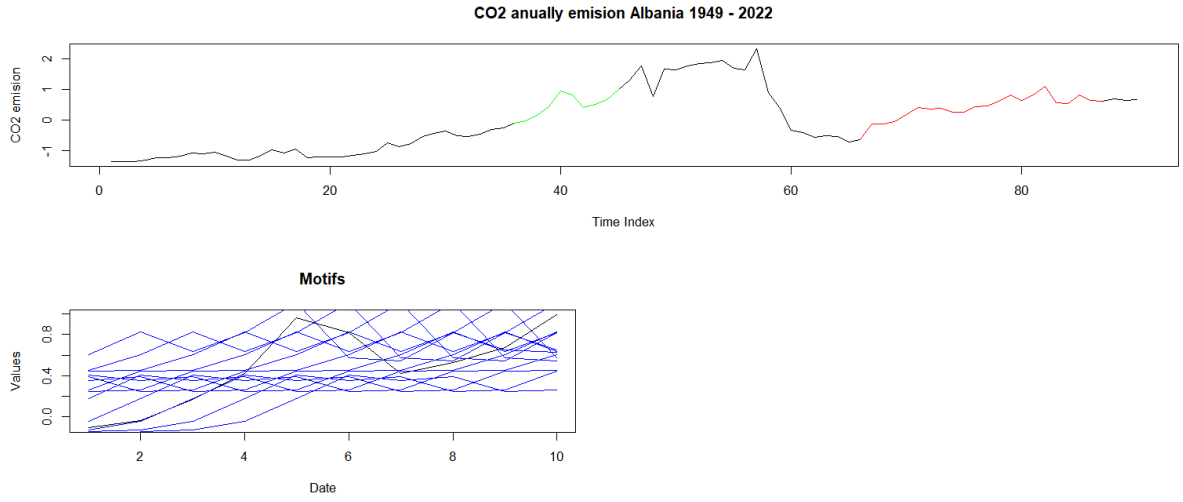


Figura 7.4. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

Për epsilon-query sa dyfishi i shmangies mesatare katrore ne gjejmë motivet të cilat i inspektojmë manualisht pasi ka edhe nënsekuenca që plotesojnë kushtin që CID është më e vogël se epsilon-query por që nuk kanë pattern të ngjashme. Pasi bëjmë inspektimin e grafikëve

që numerikisht janë të ngjashëm, manualisht po i paraqesim motivet e gjetura në grafikun original në menyrë që rezultati i gjetur të jetë më intuitive/kuptueshem.

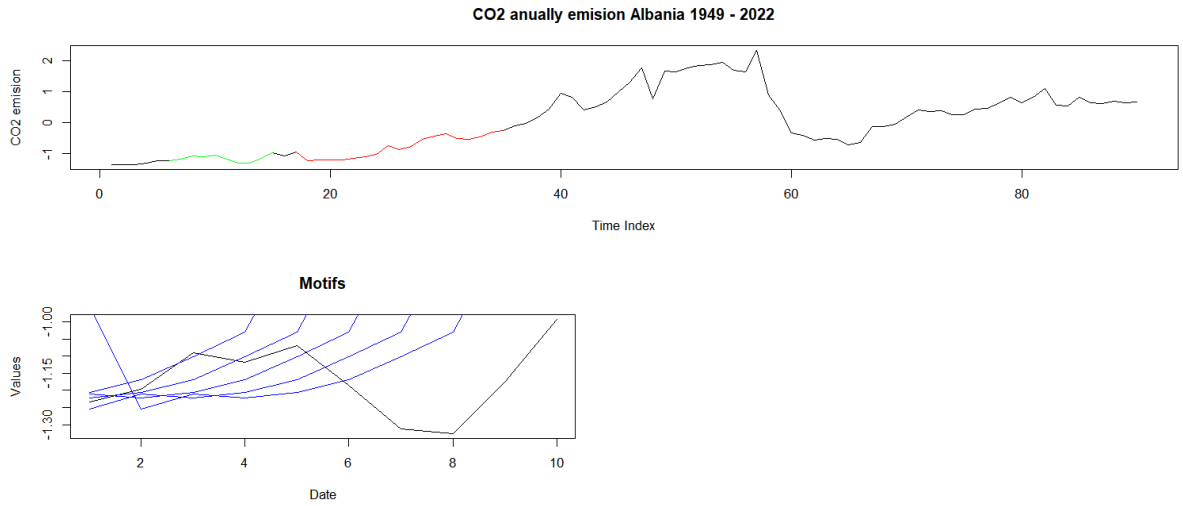


Figura 7.5. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

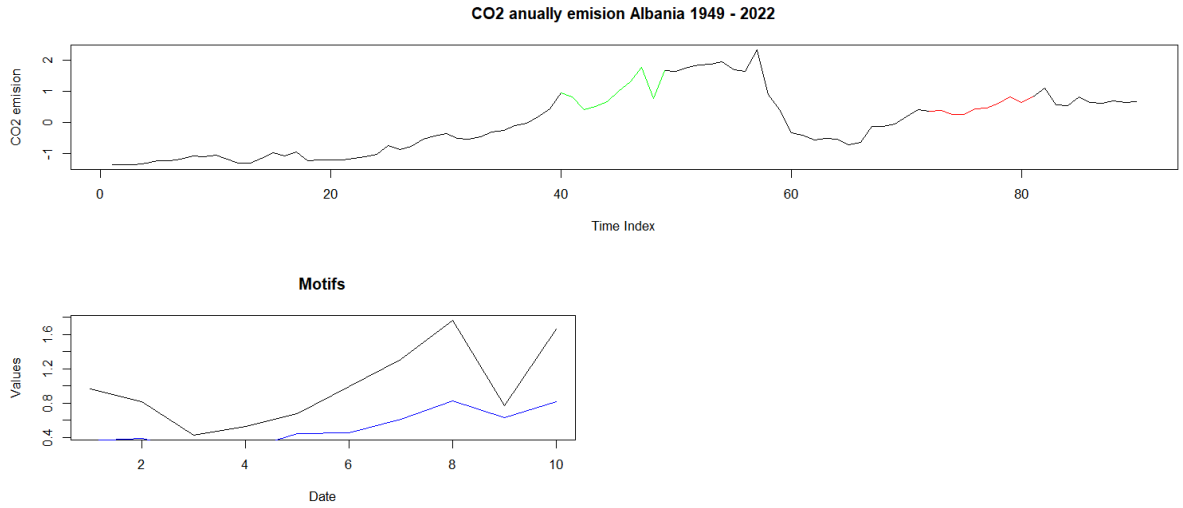


Figura 7.6. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

Nga sa pamë më sipër tentuam të rrisim epsilon-querin në mënyrë që të kapim/zbulojmë më shumë mortive por ndodhi e kundërta numri i tyre u ul ose nuk ndryshoj. A është kjo një rast i vecantë? Për ta zbuluar vazhdojmë të kërkojmë për motive duke e rritur përsëri epsilon-queri dhe zbulojmë që numri nuk ndryshon ose ndryshon me 1. Për arsye që të mor ngarkohet me

grafikë sepse tashmë ideja se si do të procedojmë është e qartë nuk po e vizualizojme këtë pjesë. Kjo ishte mënyra se si proceduam për serinë e CO2. Vazhdojmë procedurën me një tjetër seri kohore, me serinë e normës së papunesisetë. Në kapitullin pasardhës do të flasim për evolimin e algoritmit i cili ofrohet si një paketë në R. Do të përdorim pikërisht këtë paketë për të gjetur motivet për këtë seri kohore.

Fillimisht vizualizojmë informacionin për të krijuar një ide paraprake.

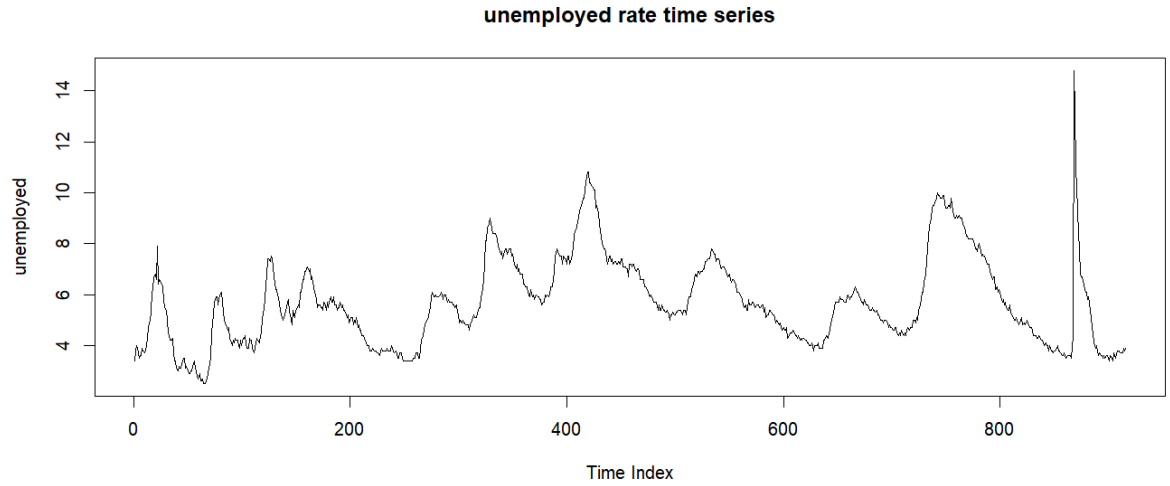


Figura 7.8. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

Pasi e investigojmë paraprakisht grafikun me sy të lirë mund të dallojmë ndonjë motiv por nuk jemi të sigurt prandaj duhet të përdorim distancën Euklidit për t'i zbuluar. Zbatojmë algoritmin find_motif të paketës TSMP ku i japim si input serine, 1 vlerë numerike që përfaqëson gjatësinë e dritares lëvizëse, një vlerë numerike për rrezen, një vlerë numerike për zonën e përjashtimit dhe numrin maksimal të motiveve.

Për një rreze kapjeje të vogël (1.5) gjejmë motivet, dhe po i paraqesim motivet e gjetura në grafikun original.

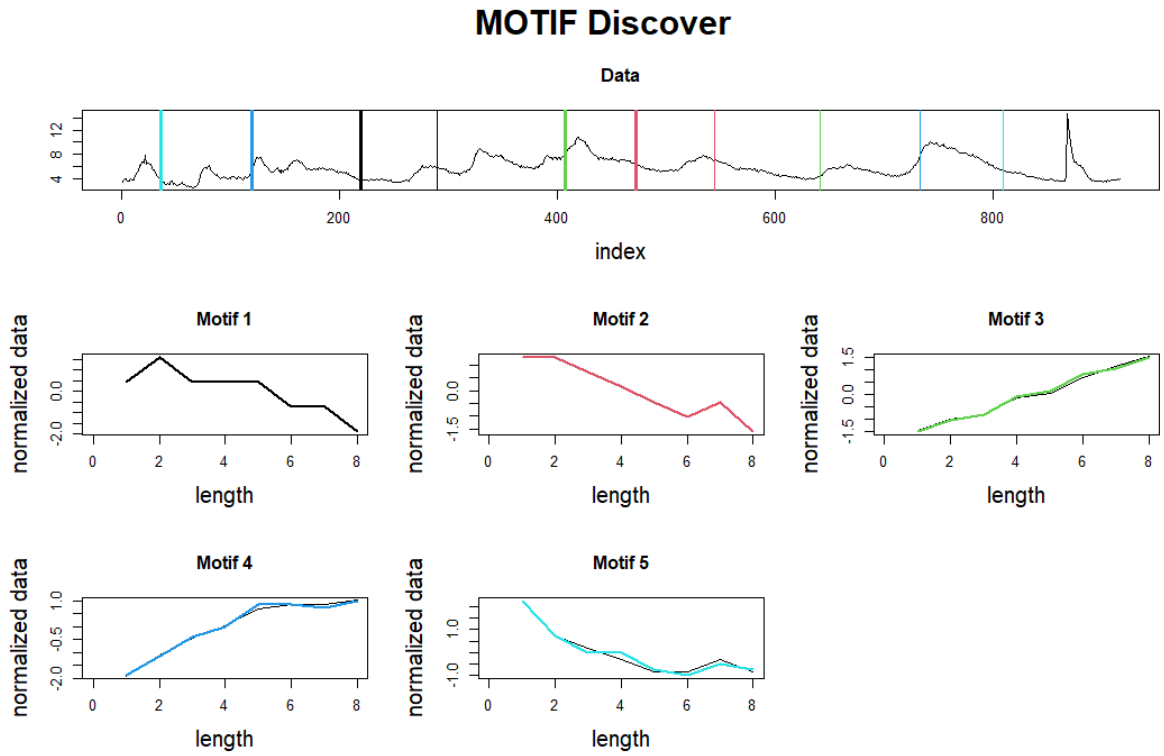


Figura 7.9. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

Pasi gjejmë motivet tentojme të rrisim rrezen në mënyrë që të kapim më shumë motive e thënë me fjalë të tjera të jemi më tolerant. Pasi aplikojme metodën e gatshme të paketës shohim se numri i motiveve te gjetura nuk rritet perkundrazi, ulet motivet shkrihen me njera tjetren.

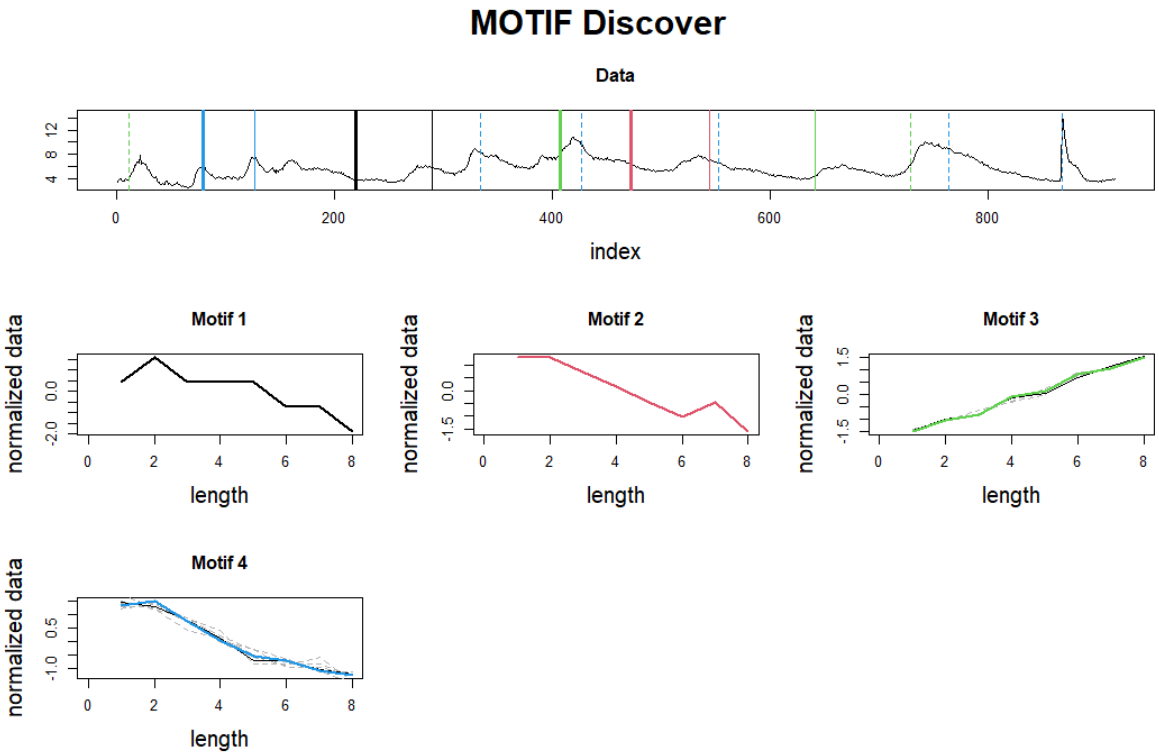


Figura 7.9. Algoritmi i zbatuar mbi serine *lindjeve* Shqiperi. (burimi Rapaj, 2024)

E rrisim perseri rrezen per te qene te sigurt dhe perseri numri i motiveve te gjetura nuk ndryshon.

Kapitulli 3

Algoritmet për zbulimin e sekuencave të ngjashme në seri kohore dhe evolimi i tyre

3.1 Prezantimi idese se si funksionon algoritmi.

Algoritmi i përdorur është algorirtmi bruteforce. Të gjitha të dhënat e seris kohere ngarkohen në memorje. Më pas normalizohen duke i zbritur mesataren dhe duke i pjestuar me dispersionin. Pasi bëjmë këtë fiksojmë një sekuencë dhe reshkasim një drirare me të njëjtën gjatësi me frekuencën e fiksuar duke filluar nga frekuenca deri në fund të seris kohore. Për sekencën e fiksuar dhe për secilën dritare që po rëshkasim llogarisim distancën ose masën e ngjashmërisë të cilën po e shënojmë me d për thjeshtësi. Perdoruesi në fillim të algortmit (përpara se të llogariten d) jep 1 epsilon-query e thënë me fjalë të thjeshta është 1 vlerë numerike më e madhe se 0 e cila do të shërbejë si kufi i sipërm per d. Me fjalë të tjera nëse d < epsilon-query do të vendoset në indexin i,j te matricës, ku i përkon me indexin e sekuencës së fiksuar dhe j përkon me indexin e dritares lëvizese. Pasi dritarja lëvizëse ka shkuar në fund të seris kohore fiksojmë një sekuencë tjetër dhe e përsërisim procedurën derisa të shkojmë dhe të fiksojmë sekuencën e fundit të seris kohore në fjalë. Duhet të theksojmë se edhe pse plotesohet kushti qe d < epsilon-query nga ana vizuale grafikët mund të mos jenë të ngjashëm prandaj duhet bere inspektimi i të dhenave nga një specialist për të vendosur nëse dy sekuencat janë apo jo të ngjashme.

3.2 Përse algorimi nuk është efektivë në ditët e sotme dhe cfarë është Big Data.

Algoritmi i përdorur arrin të gjejë motivet në një seri kohore por nuk është shumë eficent për shkak se i ngarkon të gjitha të dhënat në memorje. Në ditët e sotme kemi të bëjmë me sasi informacioni shumë të mëdha, aq e madhe sa matet me terra bit. Ky lloj informacioni dhe kalkulimet që aplikohen mbi të quhen me termin “Big Data”. Thamë që algoritmi i mësipërm nuk është shumë eficent por aplikimi i algoritmit në këtë sasi të dhënash do të donte një kohë jashzakonisht shumë të gjatë dhe një memorje gjiganteske, për ketë arsye duhet të mendojmë

një mënyrë për ta ndryshuar që ta bëjmë më eficent. Në vend që të normalizojmë të gjithë serinë kohore mund të normalizojmë vetëm sekuencën e fiksuar dhe informacionin që ndodhet në dritaren lëvizëse. Nëse bëjmë këtë nuk kemi përse të ngarkojmë të gjithë informacionin në memorje por vetëm pjesët që na duhen. Per kalkulimet nuk perdorim formulen e thjeshte por perdorim formulen që bën lidhjen e distancës euklidjane me correlacionin që e pamë në kapitullin e meparshëm.

$$d(x,y) = \sqrt[2]{2m\,(1 - corr(x,y))}$$

Ose

$$d(x,y) = \sqrt{2m\,(1 - \frac{\sum_{i=1}^{m} xi\ yi\ - m\,\mu x\,\mu y}{m\,\sigma x\,\sigma y})}$$

Ku një nga μ është 0 dhe një nga σ është 1 sepse sekuenca e fiksuar ka shpërndarje normale.

$$d(x,y) = \sqrt{2m\,(1 - \frac{\sum_{i=1}^{m} xi\ yi}{\sigma x})}$$

3.3 Përdorimi i Convulution Dhe Discrete Fourier Transform.

A mund ta përmisojmë më tej?
Në vend që të përdorim një dritare lëvizëse mund të përdorim convolution operations.
Për të gjetur emëruesin e thyesës perdorim convolution.

Shënim: Algoritmi i përdorur është zhvilluar nga persona të tjerë dhe aktualisht egziston si një paketë në R. Hapat dhe shpjegimet janë marrë nga documentacioni përkatës. Algoritmi quhet Mueen's Algorithm for Similarity Search (MASS).

Në algoritmin e mësipërm nuk do të përdorim dritare lëvizëse por në vend të kësaj do të përdorim convolution. Përdorim algoritmin Fast Fourier Transform (FFT) për të llogaritur Discrete Fourier Transform (DFT) dhe anasjelltas nga numrat complex në numrat reale. Kur llogarisim FFT mbi sekuencën e fiksuar të renditur në të kundërt (ku elementi i fundit është elementi i parë dhe i parë është i fundit) e mbushim në fund me 0 në mënyrë që të jetë me të njëjtën gjatësi me serinë kohore. Pasi kemi aplikuar FFT llogarisim convolution dhe perdorim Inverse Fast Fourier Transform (IFFT) për ti konvertuar convolution në numra reale. Më pas llogarisim distancën ose madhësinë e ngjashmërisë d midis sequencës së fiksuar dhe cdo nësekuence të serisë dhe e kontrollojmë me epsilon-query.

3.4 Kodi për algoritmin Brute Force.

```
filter.data.frame <- function(data, col.number.to.filter, filter.value){
 filtered_df <- subset(data, data[, col.number.to.filter] == filter.value)
 return(filtered_df)
}


time.series.from.dataframe <- function(data, col.number, col.number.end, row.start, row.end){
 if(col.number.end != -1){
  time.series.from.dataframe.multu.column(data, col.number, col.number.end, row.start, row.end)
 }else {
  df <- data.frame(matrix(ncol = 1, nrow = 0))
  rows <- nrow(data)

  for (i in 1:rows) {
   UNRATE <- data[i, col.number]
   df <- rbind(df, data.frame(UNRATE=UNRATE))
  }

  return(df)
 }
}

time.series.from.dataframe.multu.column <- function(data, col.number, col.number.end, row.start,
row.end){
 df <- data.frame(matrix(ncol = 1, nrow = 0))
 # rows <- nrow(data)
 #
 # for (i in 1:rows) {
 #   # chunk <- data[1:12, i]
 #   UNRATE <- data[i, col.number]
 #   df <- rbind(df, data.frame(UNRATE=UNRATE))
 # }

 # columns <- ncol(data)

 for (i in col.number:col.number.end) {  # 2:columns
  chunk <- data[row.start:row.end, i]  # 1:12
  for (j in row.start:row.end){  # 1:12
   # print(chunk[j])
   UNRATE <- chunk[j]
   df <- rbind(df, data.frame(UNRATE=UNRATE))
  }
 }

 return(df)
}
```

```
euclidean <- function(a, b) sqrt(sum((a - b)^2))


correlation.distance <- function(p, q) {
 diff.p <- diff(p)
 diff.q <- diff(q)
 pq.production <- diff.p * diff.q
 p.square <- diff.p * diff.p
 q.square <- diff.q * diff.q
 s1 <- 0
 s2 <- 0
 s3 <- 0
 n <- length(p)
 for(i in 1: (n - 1)){
  s1 <- s1 + pq.production[i]
  s2 <- s2 + p.square[i]
  s3 <- s3 + q.square[i]
 }

 s <- s2 * s3
 s <- sqrt(s)

 return(s1 / s)
}


chouakria.similarity.measure <- function(p, q, k) {
 cd <- correlation.distance(p, q)
 ch <- 2 / (1 + exp((k * cd)))
 euc <- euclidean(p, q)

 euc * ch
 return(euc * ch)
}


distance <- function(p, q, dstance.name) {
 if(dstance.name == "euclidean"){
  return(euclidean(p, q))
 }
 else if(dstance.name == "chouakria"){
  k <- 1
  return(chouakria.similarity.measure(p, q, k))
 } else {
  return(-1)
 }
}


generateSimilarityMatrix <- function(sequencLength, df, exclusion_zone, dstance.name) {
```

```
 n <-  (nrow(df) - sequencLength) + 1
 distance.matrix <- matrix(0, n, n)

 for (j in 1 : (nrow(df) - sequencLength)) {
  timeSeries.sequence <- df[(j) : (j + (sequencLength - 1)) , 1]
  exclusion.interval <- j + exclusion_zone

  for (i in j : ((nrow(df) - sequencLength) + 1)){
   chunk <- df[i : (i + (sequencLength - 1)) , 1]
   if(i > exclusion.interval){
    # distance <- euclidean(chunk, timeSeries.sequence)
    distance <- distance(chunk, timeSeries.sequence, dstance.name)
    distance.matrix[j, i] <- distance
    # distance.matrix[i, j] <- distance
   }
  }
 }

 return (distance.matrix)
}


miniGreaterThanZero <- function(similarity.matrix, row.index, threshold) {
 n <- ncol(similarity.matrix)
 distance.vector <- replicate(n, 0)

 non_zero_elements <- similarity.matrix[row.index, ( similarity.matrix[row.index, ] != 0 &
similarity.matrix[row.index, ] <= threshold ) ]

 if(length(non_zero_elements) == 0 ){
  return(distance.vector)
 }

 for(j in 1 : length(non_zero_elements) ){
  for(i in row.index : n ){
   if( !is.na(similarity.matrix[row.index, i]) & !is.na(non_zero_elements[j]) &
similarity.matrix[row.index, i] == non_zero_elements[j] ){
    distance.vector <- replace(distance.vector, i, i)
   }
  }
 }
 return(distance.vector)
}

generate.plots <- function(similarity.matrix, epsilon.query, k, n, ylabel, mainTitle, df,
sequence.length) {

 print(nrow(similarity.matrix))
 n <- min(n, nrow(similarity.matrix))
 for(i in k : n){
  distance.vector <- miniGreaterThanZero(similarity.matrix, i, epsilon.query)
```

```
   epsilon <- distance.vector[distance.vector != 0]

   if(length(epsilon) > 0){
    similarity.number <- length(epsilon)

    par(mfrow = c(2, 2))
    layout(matrix(c(1, 1, 2, 3), nrow = 2, byrow = TRUE))

    # var = readline(prompt = "press yes to acept and false to refuse : ");
    # print(var)

    # plot(c(1:length(df[,1])), df[,1], type = "l", xlab = "Index", ylab = "Values", col = "black", main =
"Multiple Time Series")
    plot(c(1:length(df[,1])), df[,1], type = "l", xlab = "Time Index", ylab = ylabel, col = "black", main =
mainTitle)
    lines(c(i : (i + sequence.length - 1)), df[i : (i + sequence.length - 1), 1], col = "green")
    for(j in 1: similarity.number){
     lines(c(epsilon[j] : (epsilon[j] + sequence.length - 1)), df[epsilon[j] : (epsilon[j] + sequence.length
- 1), 1], col = "red")
    }

    plot(as.ts(df[i : (i + sequence.length - 1), 1]), type = "l", xlab = "Date", ylab = "Values", col =
"black", main = "Motifs")
    for(j in 1: similarity.number){
     lines(as.ts(df[epsilon[j] : (epsilon[j] + sequence.length - 1), 1]), col = "blue")
    }
   }
  }
}


normalize.timeseries <- function(df) {
 m <- mean(df[ , 1])
 std <- sd(df[ , 1])
 df[ , 1] <- df[ , 1] - m
 df[ , 1] <- df[ , 1] / std

 return(df)
}


explore.motif <- function(data.path, column.number.to.filter, filter.value,
column.number.to.extract.data, sequence.length, distance.name, epsilon.query, k, n,
col.number.end, row.start, row.end) {
 df <- get.timeseries(data.path, column.number.to.filter, filter.value,
column.number.to.extract.data, col.number.end, row.start, row.end)
 similarity.matrix <- generateSimilarityMatrix(sequence.length, df, sequence.length, distance.name)
# euclidean # chouakria

 return(similarity.matrix)
 # my_dataframe <- as.data.frame(similarity.matrix)
```

```
  # return(my_dataframe)
}


get.timeseries <- function(data.path, column.number.to.filter, filter.value,
column.number.to.extract.data, col.number.end, row.start, row.end) {
  data <- read.csv(data.path)

  if(column.number.to.filter != -1){
    filtered_df <- filter.data.frame(data, column.number.to.filter, filter.value)
    # df <- time.series.from.dataframe(filtered_df, column.number.to.extract.data)
    df <- time.series.from.dataframe(filtered_df, column.number.to.extract.data, col.number.end,
row.start, row.end)
  }else{
    # df <- time.series.from.dataframe(data, column.number.to.extract.data)
    df <- time.series.from.dataframe(data, column.number.to.extract.data, col.number.end, row.start,
row.end)
  }

  df <- normalize.timeseries(df)

  return(df)
}

get.timeseries.data <- function(data.path, column.number.to.filter, filter.value,
column.number.to.extract.data, col.number.end, row.start, row.end) {
  data <- get.timeseries(data.path, column.number.to.filter, filter.value,
column.number.to.extract.data, col.number.end, row.start, row.end)
    return(data[,1])
}

get.data.bruto <- function(data.path){
  data <- read.csv(data.path)
  return(data)
}
```

Shpjegimi i kodit:

Filtering Data Frame: Funksioni filter.data.frame filtron një dataframe bazuar në një numër kolone të caktuar dhe një vlerë filtruese.

Extracting Time Series: Funksioni time.series.from.dataframe ekstrakton një seri-kohe me një kolonë nga një frame të dhënash. Nëse specifikohet col.number.end, mund të trajtojë disa kolona.

Handling Multiple Columns for Time Series: Funksioni time.series.from.dataframe.multu.column zgjeron funksionin e mësipërm për të trajtuar ekstraktimin nga disa kolona brenda rreshtave të specifikuara.

Distance Metrics: Funksionet si euclidean, correlation.distance, chouakria.similarity.measure, llogarisin distana të ndryshme ose masen e ngjashmërisë midis dy serive kohore.

Generating Similarity Matrix: Funksioni generateSimilarityMatrix ndërton një matricë të distancave ose ngjashmërive midis nënvargjeve të një dataframe të serise kohore.

Identifying Motifs: Funksionet miniGreaterThanZero dhe generate.plots identifikojnë motive (patterns of interest) në një matricë ngjashmërie dhe gjenerojnë grafikë që përhapin këto motive brenda të dhënave origjinale të seri-kohe.

Data Normalization: Funksioni normalize.timeseries normalizon një frame të dhënash seri-kohe.

Workflow Integration Functions: explore.motif, get.timeseries, dhe get.timeseries.data integrojne/manaxhojne rrjedhen e informacionit punes procesin nga leximi i të dhënave, filtrimi, ekstraktimi i seri-kohe, llogaritja e matricave të ngjashmërisë, dhe vizualizimi i motiveve.

Shërbime për Shkarkimin e të Dhënave: get.data.bruto lexon të dhënat e pa përpunuara nga një skedar CSV.

3.5 Kodi për gjetjen e motiveve nga paketa TSMP

```
install.packages("tsmp")
library(tsmp)

tseries_ts <- as.ts(df)
mp <- tsmp(tseries_ts, window_size= 8, mode ="stamp")  # , mode = "stomp"
mode= "Guided",  mode= "Unconstrained" , exclusion_zone= 20
mot <- find_motif(mp, n_motifs= 20, n_neighbors= 5, radius= 8, exclusion_zone= 8)  #mode=
"Guided",  mode= "Unconstrained" , exclusion_zone= 20
plot(mot)
```

Ky skript presupozon se keni një dataframe df që përmban të dhënat tuaja të seri kohore. Ai instalohet dhe ngarkon paketen tsmp, konverton të dhënat në një objekt seri kohore (tseries_ts), aplikon algoritmin TSMP (tsmp()), identifikon motive (find_motif()), dhe në fund vizualizon motive (plot()).

3.6 Kodi për gjetjen e motiveve duke përdorur algoritmin Brute Force duke përdorur R në JavaScript si një library e JS nëpërmjet Web Asembly.

```
import {WebR} from 'https://webr.r-wasm.org/latest/webr.mjs';
import * as Plot from "https://cdn.jsdelivr.net/npm/@observablehq/plot@0.6/+esm";

const webR = new WebR();
await webR.init();

let exploreMotifScript = () => {
   return '<skripti i R qe eshte shpjeguar me siper>';
}

function arrayToMatrix(arr, rows, cols) {
   const matrix = [];
   let k = 0;
   for (let i = 0; i < rows; i++) {
      matrix.push([]);
   }
   for (let i = 0; i < rows; i++) {
      for (let j = 0; j < cols; j++) {
         matrix[j][i] = arr[k];
         k++;
      }
   }
   return matrix;
}

const grafic = (data, selectorString, i, distanceArray, sequenceLength) => {
   // Set the dimensions and margins of the graph
   const margin = {top: 70, right: 50, bottom: 50, left: 50},
      width = 1200 - margin.left - margin.right,
      height = 500 - margin.top - margin.bottom;

   // Append the SVG object to the body of the page
   const svg = d3.select(selectorString)
      .append("svg")
      .attr("width", width + margin.left + margin.right)
      .attr("height", height + margin.top + margin.bottom)
      .append("g")
      .attr("transform", `translate(${margin.left},${margin.top})`);

   let mainSequenceDatas = data.filter(dataElement => dataElement.index >= i &&
dataElement.index < (i + sequenceLength))
   let patternDataArrays = [];

   for (let distanceIndex of distanceArray) {
      let dataArray = data.filter(dataElement => dataElement.index >= distanceIndex &&
dataElement.index < (distanceIndex + sequenceLength))
      patternDataArrays.push(dataArray);
   }

   // X scale
   const x = d3.scaleTime()
```

```
    // .domain(d3.extent(data, d => d.date))
    .domain(d3.extent(data, d => d.index))
    .range([0, width]);

// Y scale
const y = d3.scaleLinear()
    // .domain([0, d3.max(data, d => d.value)])
    .domain([d3.min(data, d => d.value), d3.max(data, d => d.value)])
    .range([height, 0]);

// Define the line
const line = d3.line()
    // .x(d => x(d.date))
    .x(d => x(d.index))
    .y(d => y(d.value));

// Add the X Axis
svg.append("g")
    .attr("transform", `translate(0,${height})`)
    .call(d3.axisBottom(x)
        .ticks((data.length * 0.5)).tickFormat(d3.format("d")));

// Add the Y Axis
svg.append("g")
    .call(d3.axisLeft(y));

// Add the line path
svg.append("path")
    .datum(data)
    .attr("class", "line")
    .attr("d", line)
    .style("stroke", "black");

// Add the line path
svg.append("path")
    .datum(mainSequenceDatas)
    .attr("class", "line")
    .attr("d", line)
    .style("stroke", "red");

for (let arr of patternDataArrays) {
    // Add the line path
    svg.append("path")
        .datum(arr)
        .attr("class", "line")
        .attr("d", line)
        .style("stroke", "green");
}

// Add chart title
svg.append("text")
```

```
    .attr("x", width / 2)
    .attr("y", ((-margin.top / 2) - 15))
    .attr("class", "title")
    .text("Line Chart Example");

// Add chart subtitle
svg.append("text")
    .attr("x", width / 2)
    .attr("y", -margin.top / 2 + 10)
    .attr("class", "subtitle")
    .text("With conditional coloring and legend");

// Add legend
const legend = svg.append("g")
    .attr("class", "legend")
    .attr("transform", `translate(${width - 200},${margin.top - margin.top})`);

legend.append("rect")
    .attr("x", 0)
    .attr("y", 0)
    .attr("width", 10)
    .attr("height", 10)
    .style("fill", "green");

legend.append("text")
    .attr("x", 20)
    .attr("y", 10)
    .text("Sequence that match motifs");

legend.append("rect")
    .attr("x", 0)
    .attr("y", 20)
    .attr("width", 10)
    .attr("height", 10)
    .style("fill", "red");

legend.append("text")
    .attr("x", 20)
    .attr("y", 30)
    .text("Found motifs");

// x label
svg.append("text")
    .attr("x", width / 2)
    .attr("y", height + 50)
    .attr("text-anchor", "middle")
    .attr("transform", `rotate(0, ${width / 2}, ${height + 30})`)
    .text("indexes");

// y label
svg.append("text")
```

```
        .attr("x", -30)
        .attr("y", (height / 2) - 30)
        .attr("text-anchor", "middle")
        .attr("transform", `rotate(-90, ${-10}, ${height / 2})`)
        .text("indexes");

}

const patternVizualzer = (data, selectorString, i, distanceArray, sequenceLength) => {

    // Set the dimensions and margins of the graph
    const margin = {top: 40, right: 30, bottom: 40, left: 50},
        width = 300 - margin.left - margin.right,
        height = 300 - margin.top - margin.bottom;

    // Append the SVG object to the body of the page
    const svg = d3.select(selectorString)
        .append("svg")
        .attr("width", width + margin.left + margin.right)
        .attr("height", height + margin.top + margin.bottom)
        .append("g")
        .attr("transform", `translate(${margin.left},${margin.top})`);

    let patternDataArray = [];
    let mainSequenceData = data.filter(dataElement => dataElement.index >= i && dataElement.index
< (i + sequenceLength))
    mainSequenceData = mainSequenceData.map((dataElement, index) => {
        let temp = Object.assign({}, dataElement);
        temp.index = index;
        return temp;
    });

    // Defining colors for charts
    const customColors = [
        "#1f77b4", "#ff7f0e", "#2ca02c", "#d62728", "#9467bd",
        "#8c564b", "#e377c2", "#7f7f7f", "#bcbd22", "#17becf",
        "#aec7e8", "#ffbb78", "#98df8a", "#ff9896", "#c5b0d5",
        "#c49c94", "#f7b6d2", "#c7c7c7", "#dbdb8d", "#9edae5"
    ];
    const colorScale = d3.scaleOrdinal(customColors);

    let j = 1;
    for (let distanceIndex of distanceArray) {
        let dataArray = data.filter(dataElement => dataElement.index >= distanceIndex &&
dataElement.index < (distanceIndex + sequenceLength))
        dataArray = dataArray.map((dataElement, index) => {
            let temp = Object.assign({}, dataElement);
            temp.index = index;
            return temp;
        });
        patternDataArray.push(dataArray);
```

```
}

// X scale
const x = d3.scaleTime()
   .domain(d3.extent(mainSequenceData, d => d.index))
   .range([0, width]);

// Y scale
const y = d3.scaleLinear()
   .domain([d3.min(mainSequenceData, d => d.value), d3.max(mainSequenceData, d => d.value)])
   .range([height, 0]);

// Define the line
const line = d3.line()
   .x(d => x(d.index))
   .y(d => y(d.value));

// Add the X Axis
svg.append("g")
   .attr("transform", `translate(0,${height})`)
   .call(d3.axisBottom(x)
      .ticks((mainSequenceData.length * 0.3)).tickFormat(d3.format("d")));

// Add the Y Axis
svg.append("g")
   .call(d3.axisLeft(y));

// Add the line path
svg.append("path")
   .datum(mainSequenceData)
   .attr("class", "line")
   .attr("d", line)
   .style("stroke", "black");

for (let arr of patternDataArray) {
   let xMainSequence = d3.scaleTime()
      .domain(d3.extent(arr, d => d.index))
      .range([0, width]);

   // Y scale
   let yMainSequence = d3.scaleLinear()
      .domain([d3.min(arr, d => d.value), d3.max(arr, d => d.value)])
      .range([height, 0]);

   // Define the line
   let mainSequenceLine = d3.line()
      .x(d => xMainSequence(d.index))
      .y(d => yMainSequence(d.value));

   // Add the line path
   svg.append("path")
```

```
            .datum(arr)
            .attr("class", "line")
            .attr("d", mainSequenceLine)
            .style("stroke", colorScale(j));
        j++;
    }

    // Add chart title
    svg.append("text")
        .attr("x", width / 2)
        .attr("y", -15)
        .attr("class", "title")
        .text("Comparing Motifs");

}

const prepareDataForVizualization = async (specificDate, dataframe) => {
    let graphData = []
    for (let i = 0; i < dataframe.length; i++) {
        graphData.push({
            index: i,  //  date : specificDate.clone().add(i, 'days')
            value: dataframe[i]
        })
    }

    return graphData;
}

const csvNames = async () => {
    const url = 'http://localhost:3000/csv-files'

    let fileNameContainer = document.getElementById('fileName');
    let fileNameContainerSecond = document.getElementById('dataPath');

    try {
        let fileNames = await axios.get(url);
        for (let fileName of fileNames.data){
            let option = document.createElement("option");
            option.value = fileName.toString().replace('.csv', '') ;
            option.text = fileName.toString().replace('.csv', '');

            fileNameContainer.appendChild(option.cloneNode(true));
            fileNameContainerSecond.appendChild(option.cloneNode(true));
        }

    } catch (error) {
        console.error('File upload failed:', error);
    }
}

const addHtmlSection = async (id) => {
```

```
    const targetDiv = document.getElementById("main-container");

    const innerDiv = document.createElement("div");
    innerDiv.classList.add("inner-div");
    innerDiv.id = "id" + id;

    // Create a new div element for the nested div
    const mainChartDiv = document.createElement("div");
    mainChartDiv.classList.add("nested-div");
    mainChartDiv.id = "mainChart" + id;

    const patternChartDiv = document.createElement("div");
    patternChartDiv.classList.add("nested-div");
    patternChartDiv.id = "patternChart" + id;

    const separator = document.createElement("div");
    separator.classList.add("separator");

    // Append the nested div to the inner div
    innerDiv.appendChild(mainChartDiv);
    innerDiv.appendChild(patternChartDiv);
    innerDiv.appendChild(separator);

    targetDiv.appendChild(innerDiv);
}

const createDynamicCheckBox = async (columnToExtractDataContainer, checkBoxName, k) => {
    let label = document.createElement('label');
    label.textContent = (k + 1)
    let checkbox = document.createElement('input');
    checkbox.type = 'checkbox';
    // checkbox.id = k;
    checkbox.value = (k + 1);
    checkbox.name = checkBoxName

    label.appendChild(checkbox)
    columnToExtractDataContainer.append(label)
}

const showDataSetRow = async (webR, fileName) => {
    let baseFilePath = 'http://localhost:3000/csv/'
    let dataa = await webR.evalR(`get.data.bruto("${(baseFilePath + fileName)}")`)
    let dataframee = await dataa.toJs();

    // const headerRow = document.getElementById('headerRow');
    const tableBody = document.getElementById('tableBody');
    const columnToExtractDataContainer = document.getElementById('columnToExtractDataId');
    const columnToFilterContainer = document.getElementById('columnToFilterId');

    let numberOfColumns = dataframee.names.length
    for (let k = 0; k < numberOfColumns; k++) {
```

```
        createDynamicCheckBox(columnToExtractDataContainer, 'columnToExtractData', k)
        createDynamicCheckBox(columnToFilterContainer, 'columnToFilter', k)
    }


    let divContainer = document.createElement('div');
    divContainer.classList.add('table-head');
    for (let columnName of dataframee.names) {
        let th = document.createElement('div');
        th.classList.add('table-row');
        th.textContent = columnName;
        th.classList.add('table-cell');
        divContainer.appendChild(th);
    }
    tableBody.appendChild(divContainer);

    let numberRows = dataframee.values[0].values.length
    for (let i = 0; i < numberRows; i++) {
        let row = document.createElement('div');
        row.classList.add('table-row');
        for (let j = 0; j < numberOfColumns; j++) {
            let td = document.createElement('div');
            td.textContent = dataframee.values[j].values[i]
            td.classList.add('table-cell');
            row.appendChild(td);
        }
        tableBody.appendChild(row)
    }

}

// Function to handle the file upload using axios
document.getElementById("uploadFile").onclick = async () => {
    const fileInput = document.getElementById('fileInput');
    const file = fileInput.files[0];

    if (!file) {
        alert('Please select a file to upload.');
        return;
    }

    const formData = new FormData();
    formData.append('file', file);

    // Display the file name in the console (optional)
    console.log('Uploading file:', file.name);
    let url = 'http://localhost:3000/upload/' + file.name;
    let response;
    try {
        response = await axios.post(url, formData, {
            headers: {
```

```
            'Content-Type': 'multipart/form-data'
          }
        });

        alert('File uploaded successfully!');
    } catch (error) {
        console.error('File upload failed:', error);
    }
}

const maxArray = async (columnToFilter) => {
    let maxValue = parseInt(columnToFilter[0]);
    for(let mv of columnToFilter){
        if (maxValue <= parseInt(mv)){
            maxValue = parseInt(mv)
        }
    }

    return maxValue;
}

const minArray = async (columnToFilter) => {
    let minValue = parseInt(columnToFilter[0]);
    for(let mv of columnToFilter){
        if (minValue >= parseInt(mv)){
            minValue = parseInt(mv)
        }
    }

    return minValue;
}


document.getElementById("idButton").onclick = async () => {
    let rscript = await exploreMotifScript();

    let dataPath = document.getElementById('dataPath').value;
    let columnToFilter =
Array.from(document.querySelectorAll('input[name="columnToFilter"]:checked')).map(checkbox =>
checkbox.value);
    let filterValue = document.getElementById('filterValue').value;
    let columnToExtractData =
Array.from(document.querySelectorAll('input[name="columnToExtractData"]:checked')).map(check
box => checkbox.value);
    let distanceName = document.getElementById('distanceName').value;
    let sequenceLength = parseInt(document.getElementById('sequenceLength').value);
    let epsilonQuery = parseFloat(document.getElementById('epsilonQuery').value);
    let kk = document.getElementById('k').value;
    let n = parseInt(document.getElementById('n').value);
    let minRow = document.getElementById('min-row').value;
    let maxRow = document.getElementById('max-row').value;
```

```
    // You can now send these values to the server or use them in your application logic
    let fileBasePath = 'http://localhost:3000/csv/'
    let getTimeseriesData = `get.timeseries.data("${(fileBasePath + dataPath)}",
${columnToFilter.length == 1 ? columnToFilter[0] : -1}, "${filterValue}",
${columnToExtractData.length > 1 ? await minArray(columnToExtractData) :
columnToExtractData[0]}, ${columnToExtractData.length > 1 ? await
maxArray(columnToExtractData) : -1}, ${!minRow ? -1 : parseInt(minRow)}, ${!maxRow ? -1 :
parseInt(maxRow)})`

    // execute r function
    // let rscript = await exploreMotifScript();
    await webR.evalR(rscript);

    // Loading the time series data
    let data = await webR.evalR(getTimeseriesData)
    let dataframe = await data.toArray();

    // creating time series chart
    // let specificDate = moment('2024-06-08');
    let graphData = await prepareDataForVizualization(null, dataframe);

    let exploreMotif = `explore.motif("${(fileBasePath + dataPath)}", ${columnToFilter.length == 1 ?
columnToFilter[0] : -1}, "${filterValue}", ${columnToExtractData.length > 1 ? await
minArray(columnToExtractData) : columnToExtractData[0]}, ${sequenceLength}, "${distanceName}",
${epsilonQuery}, ${(parseInt(kk) + 1)}, ${n}, ${columnToExtractData.length > 1 ? await
maxArray(columnToExtractData) : -1},  ${!minRow ? -1 : parseInt(minRow)},  ${!maxRow ? -1 :
parseInt(maxRow)})`

    // creating similarity matrix and plotting patterns
    let returnValue = await webR.evalR(exploreMotif)
    // let x = await returnValue.toArray();
    let matrix = await returnValue.toJs();
    // one array correspond to one column of matrix
    console.log('n: ', matrix.values[0].values.length)

    let k = parseInt(kk) - 1;
    n = Math.min(matrix.values[0].values.length, n)
    for (let i = k; i < n; i++) {
      let numberElementsNonZero = 0
      let distanceArray = new Array(matrix.values[0].values.length).fill(0);
      for (let j = i; j < matrix.values[i].values.length; j++) {   // let j = k
        if (matrix.values[j].values[i] < epsilonQuery && matrix.values[j].values[i] > 0) {  //
          distanceArray[j] = j
          numberElementsNonZero++
        }
      }
      if (numberElementsNonZero > 0) {
        addHtmlSection(i);
        let mainChartId = "#mainChart" + i;
        let patternChartId = "#patternChart" + i;
```

```
        distanceArray = distanceArray.filter(dElement => dElement !== 0)
        grafic(graphData, mainChartId, i, distanceArray, sequenceLength);
        patternVizualzer(graphData, patternChartId, i, distanceArray, sequenceLength)
      }
    }
}

document.getElementById("showDataset").onclick = async () => {
    // loading the script
    let rscript = await exploreMotifScript();
    await webR.evalR(rscript);
    let fileName = document.getElementById('fileName').value;

    showDataSetRow(webR, fileName);
}

csvNames();
```

Kodi JavaScript i dhënë definon disa funksione që lidhen me analizën dhe vizualizimin e seri kohore. Fillimisht, importon libraritë e nevojshme si WebR dhe Plot nga burime të jashtme. Funksioni exploreMotifScript definon disa funksione të ngjashme me R për filtrimin e tabela të të dhënave, llogaritjen e distancave, dhe gjenerimin e matricave të ngjashmërisë. Këto funksione janë projektuar për të trajtuar të dhënat e seri kohore dhe për të kryer operacione si filtrimi, llogaritja e distancave (Euclidean dhe Chouakria), dhe gjenerimi i grafikëve bazuar në matricën e ngjashmërisë.

Funksionet përfshijnë operacione për normalizimin e të dhënave të seri kohore, gjenerimin e matricave të ngjashmërisë, dhe vizualizimin e modeleve dhe motiveve brenda të dhënave. Kodi gjithashtu përfshin funksione ndihmëse për leximin e të dhënave CSV, filtrimin e tabelave të të dhënave bazuar në kritere të përcaktuara, dhe përgatitjen e të dhënave për vizualizim duke përdorur D3.js.

Në përgjithësi, kodi është i strukturuar për të lehtësuar analizën e eksplorueshme të të dhënave të seri kohore, duke përfshirë filtrimin, matjen e ngjashmërisë dhe vizualizimin interaktiv të modeleve dhe motiveve të gjetura brenda të dhënave.

Përfundimet

Në këtë punim diplome u studiuan disa nga distancat e njhura që përdoren në literaturën e serive kohore me qëllim zbulimine sekuencave që përsëriten përgjatë një serie kohore me shumë vrojtime.

Gjatësia e sekuencave që përsëriten mund të jetë disa muaj ose vjetore. Në varësi të problemit që trajtohet, mund të jemi të interesuar për sekuenca që përsëriten një numër të caktuar herësh, për sekuenca që shtrihen në një segment të caktuar të serisë, ose dhe për sekuencen që ka numrin maksimal të përsëritjeve përgjatë gjithë serisë.

Të shumta janë llojet e distancave, që i përmbushin të tre kushtet për të qenë distanca. Por, ka mjaft raste kur të paktën njëri nga këto kushte nuk plotësohet. Atëherë kemi të bëjmë me një tjetër madhësi po aq të rëndësishme sa dhe distanca, të quajtur **madhësi ngjashmërie**. (si përshembull distancat: Manhattan, Minkoëski, Jaccard, DTW (Dynamic time Warping) etj)

Ndërmjet tyre është dhe distanca CID (Complexity Invariant Distance) propozuar nga Batista et al. 2013.

Disa algoritme që përdoren gjerësisht për zbulimin e sekuencave të ngjashme përdorin distancën euklidiane si distancë e cila ka lehtësi në llogaritje dhe është me e lehtë për tu aplikuar. por kjo distancë kufizon në faktin se sekuencat që krahasohen duhet të kenë të njëjtën gjatësi.

Algoritmi që propozojnë Rapaj et al. 2024 krahason distancën Euklidiane dhe atë CID për zbulimin e ngjashmërive në sekuenca të serive kohore.

Nëpërmjet shembujve në seri kohore reale arrihet në përfundimin se CID është më e mirë për tu përdorur në seri kohore me kompleksitet të lartë ndërsa ajo Eukldiiane ofron rezultate të kënaqshme në seri kohore me kompleksitet të ulët. Më tej është kërkuar se si të përshtatet algoritmi në mëyrë që të përballojë një sasi të madhe të dhënash në një kohë më të shkurtër dhe duke kursyer memorjen. Gjë që na solli në evolimin dhe përdorimin e calculimeve komplekse si convolution dhe transformimet discrete furrier.

Referencat

Agrawal, R., Faloutsos, C., Swami, A., (1993): "Efficient similarity search in sequence databases," in: Fourth International Conference on Foundations of Data Organization, D. Lomet, Ed., Heidelberg: SpringerVerlag, pp. 69–84

Alcock, R. J., Manopulous, Y., (1999): "Time-Series Similarity Queries Employing a Feature-Based Approach", In: 7th Hellenic Conference on Informatics, Ioannina, Greece, August 27-29.

Assfalg, J., Kriegel, H.-P., Kroger, P., Kunath, P., Pryakhin, A., Renz M., (2006): "Time Series Analysis Using the Concept of Adaptable Threshold Similarity", Proc. 18th Int. Conf. on Scientific and Statistical Database Management (SSDBM'06), Vienna, Austria

Batista, G., Keogh, E. J. Tataw, O., de Souza, V., "CID: an efficient complexity-invariant distance for time series"

Batista, G. and wang, X. (2011). "A complexity-invariant distance measure for time series". SIAM Internation Conference on Data Mining (SDM) on Data Mining, Philadelphia, PA, USA.

Chan, K. & Fu, W. (1999). Efficient time series matching by wavelets. Proceedings of the 15 th IEEE International Conference on Data Engineering.

Dhamo, E., Puka, Ll., (2014): "An Algorithm For Discovering Similar Subsequences In Time Series Data Using Cid (Complexity – Invariant Distance) "- SPNA, December, 5-6

Dhamo, E., Ismailaja, N., Kalluçi, E., (2015): "Comparing the efficiency of CID distance and CORT coefficient for finding similar subsequences in time series", Sixth International Conference ISTI, 5-6 June.

Giusti,R. & Batista, G. E. A. P. A. (2013): "An Empirical Comparison of Dissimilarity Measures for Time Series Classification", BRACIS : 82-88

Keogh, E.J., Chu S., Hart, D. and Pazzani, M., J., (2001): "An Online Algorithm for Segmenting Time Series," Proc. IEEE Int'l Conf. Data Mining (ICDM), pp. 289-296.

Keogh, E., (2002): "Exact indexing of dynamic time warping", in Proc. VLDB, pp. 406–417.

Keogh, E., Chakrabarti, K., Pazzani, M. & Mehrotra, Sh., (2001): "Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases", Knowledge and Information Systems, August, Vol. 3, Issue 3, pp 263-286

Keogh, E., Lin J., and Truppel, W., (2003). "Clustering of Time Series Subsequences is Meaningless: Implications for Past and Future Research". In *proceedings of the 3rd IEEE International Conference on Data Mining* . Melbourne, FL. Nov 19-22. pp 115-122.

Lin, J., Keogh, E. , Lonardi, S. and Patel, P. (2002): "Finding Motifs in Time Series", in Proc. of 2nd Workshop on Temporal Data Mining
Lloyd, S., (1982): "Least Squares Quantization in PCM", IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. IT-28, NO. 2, MARCH

Mueen A., Keogh, E., J., (2010A): "Online discovery and maintenance of time series motifs". KDD, pg. 1089-1098

Mueen, A., Keogh, E., Zhu, Q., Cash, S., Westover, B., (2009A) "Exact Discovery of Time Series Motifs", SDM, pg. 473-484

Mueen A., Keogh, E., J., Bigdely- Shamlo N., (2009B): "Finding Time Series Motifs in Disk-Resident Data". ICDM, pg 367-376

https://databaza.instat.gov.al:8083/pxweb/en/DST/START__BD__BIRTH/BD0003/

http://www.cs.princeton.edu/courses/archive/spring10/cos424/slides/7-clust.pdf

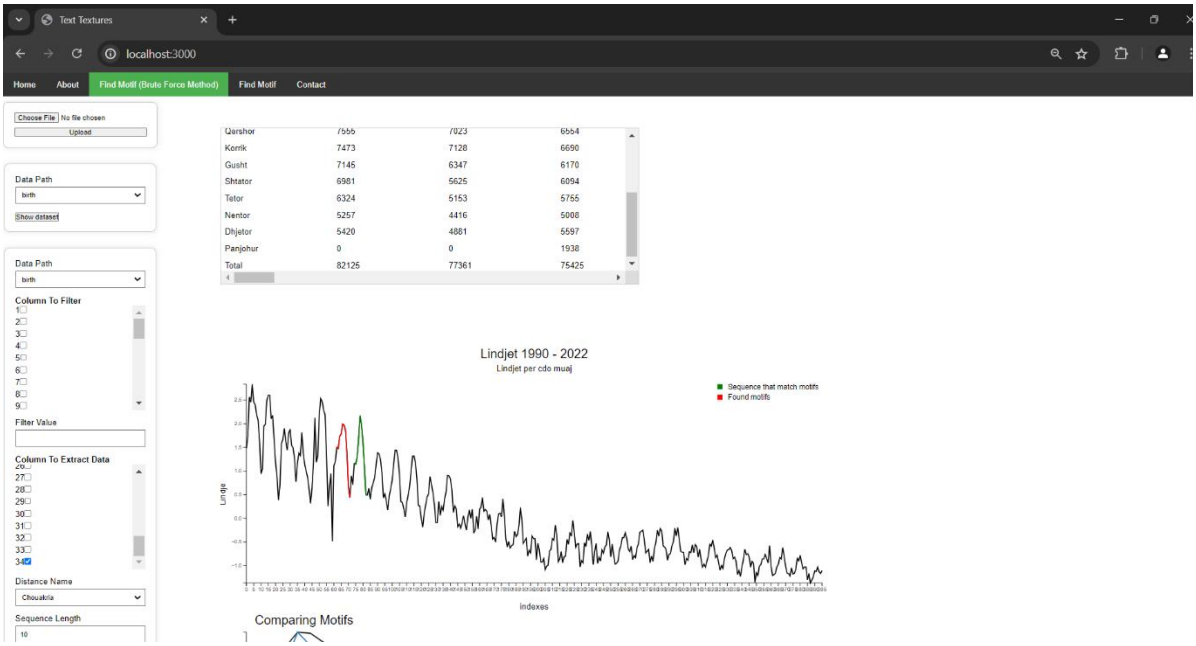
http://web.stanford.edu/class/cs345a/slides/12-clustering.pdf

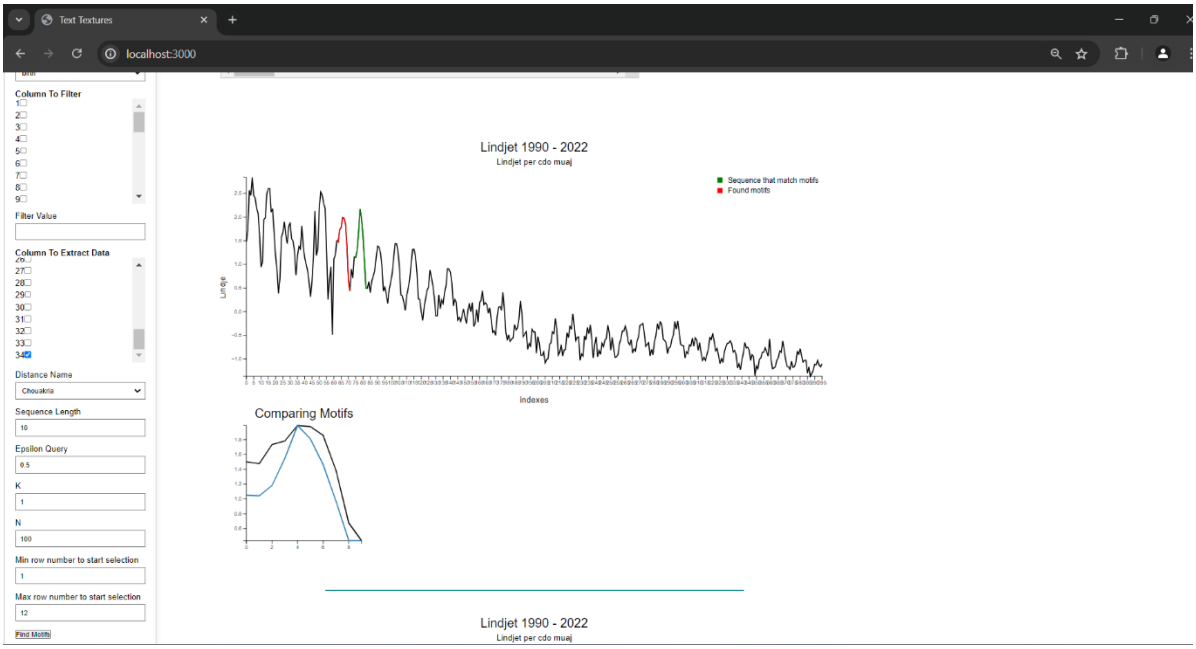
http://stackoverflow.com/questions/15376075/cluster-analysis-in-r-determine-the-optimal-number-of-clusters

https://fred.stlouisfed.org/series/UNRATE
https://ourworldindata.org/co2-and-greenhouse-gas-emissions
https://ourworldindata.org/co2-dataset-sources

https://www.youtube.com/watch?v=LnQneYvg84M
https://databaza.instat.gov.al:8083/pxweb/en/DST/START__BD__BIRTH/BD0003/
https://github.com/matrix-profile-foundation/tsmp

Shtojcat


Text Textures
localhost:3000
Home
About
Find Motif (Brute Force Method)
Find Motif
Contact
Choose File
No file chosen
Upload
Data Path
birth
Show dataset
Data Path
birth
Column To Filter
Filter Value
Column To Extract Data
Distance Name
Chouakria
Sequence Length
10
Qershor 7555 7023 6554
Korrik 7473 7128 6690
Gusht 7145 6347 6170
Shtator 6981 5625 6094
Tetor 6324 5153 5755
Nentor 5257 4416 5008
Dhjetor 5420 4881 5597
Panjohur 0 0 1938
Total 82125 77361 75425
Lindjet 1990 - 2022
Lindjet per cdo muaj
Sequence that match motifs
Found motifs
Lindje
Indexes
Comparing Motifs


Text Textures
localhost:3000
Column To Filter
Filter Value
Column To Extract Data
Distance Name
Chouakria
Sequence Length
10
Epsilon Query
0.5
K
1
N
100
Min row number to start selection
1
Max row number to start selection
12
Find Motifs
Lindjet 1990 - 2022
Lindjet per cdo muaj
Sequence that match motifs
Found motifs
Lindje
Indexes
Comparing Motifs
Lindjet 1990 - 2022
Lindjet per cdo muaj