Analiza și Prognoza Seriilor de Timp

Capitolul 10: Recapitulare Comprehensivă

Daniel Traian PELE

Academia de Studii Economice din București
IDA Institute Digital Assets
Blockchain Research Center
AI4EFin Artificial Intelligence for Energy Finance
Academia Română, Institutul de Prognoză Economică
MSCA Digital Finance

Obiective de învățare

La finalul acestui capitol, veți fi capabili să:

1. Aplicați fluxul complet de prognoză, de la date la evaluare
2. Selectați modelul potrivit în funcție de caracteristicile datelor
3. Evaluați acuratețea prognozelor folosind metrici și validare încrucișată
4. Integrați cunoștințele din toate capitolele anterioare în practică

Cuprins

Abordarea științifică a prognozei

Întrebarea de cercetare

- Cum putem **evalua riguros** performanța prognozei evitând supraajustarea?

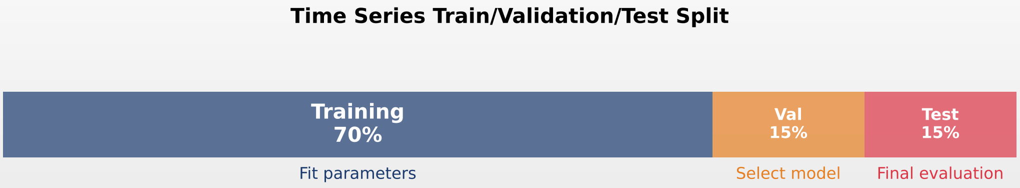
Problema fundamentală

- Ajustarea în eșantion $\neq$ Performanța în afara eșantionului
- Modelele pot "memora" datele de antrenament fără a învăța tipare
- **Soluție**: Metodologia corectă train/validation/test

Principiu

- "Setul de test trebuie să rămână **neatins** până la evaluarea finală."
- Practică standard în machine learning și econometrie

Cadrul Train/Validation/Test



TSA_ch10_train_val_test_split

Metrici de evaluare

Definiție 1 (Metrici ale Erorii de Prognoză)

- **Date**: Fie y_t valorile reale, $\hat{y}_t$ prognozele

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_t (y_t - \hat{y}_t)^2}, \quad \text{MAE} = \frac{1}{n}\sum_t |y_t - \hat{y}_t|, \quad \text{MAPE} = \frac{100\%}{n}\sum_t \left|\frac{y_t - \hat{y}_t}{y_t}\right|$$

Când să folosim

- **RMSE** (Root Mean Square Error): Penalizează erorile mari
- **MAE** (Mean Absolute Error): Robust la outlieri
- **MAPE** (Mean Absolute Percentage Error): Independent de scală (%)

Atenție

- MAPE nedefinit când $y_t = 0$
- Comparați pe **același** set test
- Raportați metrici **out-of-sample**

Evaluarea prognozelor dincolo de RMSE

Metrici alternative

- **MASE** (Mean Absolute Scaled Error): $\frac{\text{MAE}_{\text{model}}}{\text{MAE}_{\text{naïve}}}$; $< 1 \Rightarrow$ bate naïve
- **DA** (Directional Accuracy): $\frac{1}{h}\sum_{t=1}^{h} 1(\text{sgn}\,\Delta\hat{y}_t = \text{sgn}\,\Delta y_t)$
- **QL** (Quantile Loss): penalizare asimetrică α vs $1-\alpha$

$$QL_\alpha = \begin{cases} \alpha(y_t - \hat{q}_t), & y_t > \hat{q}_t \\ (1-\alpha)(\hat{q}_t - y_t), & y_t \leq \hat{q}_t \end{cases}$$

- **CRPS** (Continuous Ranked Probability Score): $\int_{-\infty}^{\infty} (F(x) - 1_{x \geq y})^2 dx$

Evaluarea prognozelor: Rezultate Bitcoin

- Aplicăm GARCH (Generalized Autoregressive Conditional Heteroskedasticity) pentru prognoza volatilității Bitcoin

Rezultate Bitcoin (GARCH volatilitate)

Metrică	Valoare
RMSE	2.21
MAE	1.89
MASE	0.98
Dir. Accuracy	28.7%

- MASE < 1: GARCH bate naïve
- DA 28.7%: direcția volatilității e dificilă

Interpretare

- **RMSE/MAE**: eroarea absolută a prognozei volatilității
- **MASE** < 1: modelul GARCH performează mai bine decât naïve
- **DA 28.7%**: direcția volatilității e extrem de dificil de prezis
- Evaluarea trebuie făcută pe **setul de test**

Compararea formală a prognozelor: Diebold–Mariano

Definiție 2 (Testul Diebold–Mariano)

Diferența de pierdere: $d_t = L(e_{1t}) - L(e_{2t})$, Statistica: $DM = \frac{\bar{d}}{\sqrt{\widehat{\text{Var}}(\bar{d})}} \xrightarrow{d} N(0,1)$

Ipoteze

- H_0: performanță predictivă egală
- H_1: un model e semnificativ mai bun
- $|DM|$ mare $\Rightarrow$ respingem H_0

Rezultat Bitcoin (GARCH volatilitate)

- Normal vs Student-t: $DM = -0.51$
- $p = 0.612$ — **nu respingem** H_0
- Acuratețe similară, dar Student-t preferabil prin AIC (Akaike Information Criterion) ($\Delta = 509$)

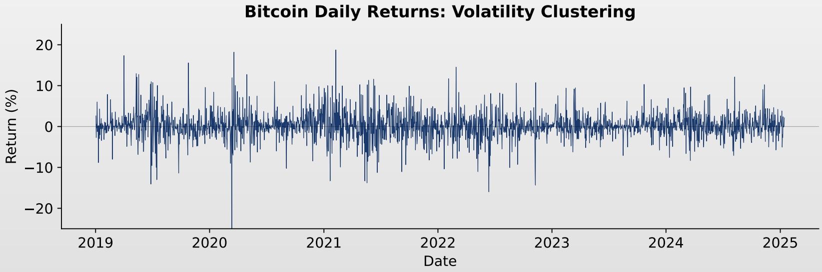
De reținut

- RMSE mai mic $\neq$ diferență semnificativă — testarea formală este **obligatorie**

Bitcoin: volatility clustering

Observație

- Randamentele mari tind să urmeze randamente mari, cele mici urmează cele mici
- Acesta este **volatility clustering** $\Rightarrow$ fenomenul pe care GARCH îl captează



TSA_ch10_btc_returns

Bitcoin: definirea problemei

Întrebarea de cercetare

- Putem prognoza **volatilitatea** Bitcoin folosind modele GARCH?

Caracteristicile Datelor

- Sursă: Yahoo Finance (BTC-USD)
- Perioadă: Ian 2019 – Ian 2025
- Frecvență: Zilnică
- Observații: ≈ 2.200 zile

Observație

- **Randamentele financiare** sunt de obicei:
  - Imprevizibile în medie
  - Predictibile în varianță
- ⇒ Focus pe **prognoza volatilității**

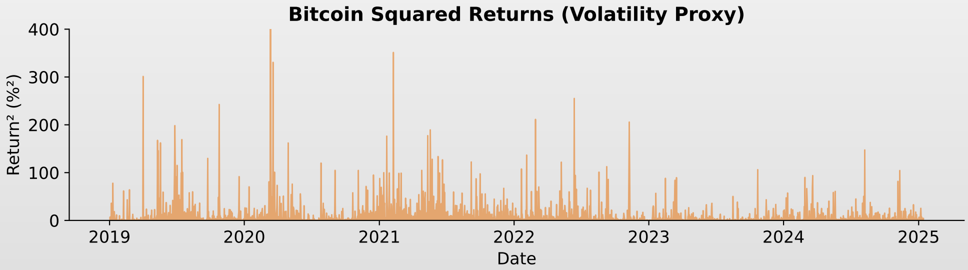
Fapte stilizate

- Randamente: medie aproape zero
- Cozi groase (curtoză > 3)
- Clustering al volatilității

Bitcoin: dovezi pentru GARCH

Observație

- Randamentele pătrate r_t^2 prezintă autocorelație semnificativă $\Rightarrow$ efecte GARCH
- ACF (Autocorrelation Function – Funcția de Autocorelație) cu descreștere lentă $\Rightarrow$ persistență ridicată a volatilității



TSA_ch10_btc_acf_squared

Specificarea modelului GARCH

Definiție 3 (Modelul GARCH(p,q))

- **Date**: Fie r_t randamentele. Modelul GARCH(p,q) este:

$$r_t = \mu + \varepsilon_t, \quad \varepsilon_t = \sigma_t z_t, \quad z_t \sim N(0,1)$$

$$\sigma_t^2 = \omega + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2$$

- **Condiții**: $\omega > 0$, $\alpha_i \geq 0$, $\beta_j \geq 0$, și $\sum_{i=1}^{q} \alpha_i + \sum_{j=1}^{p} \beta_j < 1$

Variante de model

- **GARCH(1,1)**: Cel mai comun
- **GJR-GARCH** (Glosten-Jagannathan-Runkle): Efect de levier
- **EGARCH** (Exponential GARCH): Șocuri asimetrice

Interpretare

- α: Impactul șocurilor trecute
- β: Persistența volatilității
- $\alpha + \beta \approx 1$: Persistență înaltă

GARCH: Staționaritate și varianța necondiționată

Teoremă 1 (Staționaritatea în Covarianță a GARCH(1,1))

Dacă $\alpha_1 + \beta_1 < 1$, atunci $\{\varepsilon_t\}$ este staționar în covarianță cu:

$$\bar{\sigma}^2 = \mathbb{E}[\sigma_t^2] = \frac{\omega}{1 - \alpha_1 - \beta_1}$$

Derivare

- Luăm speranța ambelor părți ale ecuației varianței:

$$\begin{aligned}
\mathbb{E}[\sigma_t^2] &= \omega + \alpha_1 \mathbb{E}[\varepsilon_{t-1}^2] + \beta_1 \mathbb{E}[\sigma_{t-1}^2] \\
\bar{\sigma}^2 &= \omega + (\alpha_1 + \beta_1)\bar{\sigma}^2 \quad \text{(staționaritate)} \\
\bar{\sigma}^2 &= \frac{\omega}{1 - \alpha_1 - \beta_1}
\end{aligned}$$

Prognozele multi-step converg la $\bar{\sigma}^2$

- Când $h \to \infty$: $\mathbb{E}_t[\sigma_{t+h}^2] \to \bar{\sigma}^2$ cu rata $(\alpha_1 + \beta_1)^h$

Bitcoin: selectarea modelului pe setul de validare

Metodologie

- Estimăm fiecare model pe **datele de antrenament**, evaluăm pe **setul de validare**
- Criterii: AIC, BIC (Bayesian Information Criterion) și MAE de validare

Model	AIC	BIC	Val MAE	Selectare
GARCH(1,1)	6.994,8	7.020,6	**2,638**	**Cel mai bun**
GARCH(2,1)	6.993,7	7.024,6	2,640	
GJR-GARCH(1,1)	6.983,7	7.014,6	2,669	
EGARCH(1,1)	—	—	—	Eșuat*

*Prognoze analitice indisponibile pentru $h > 1$

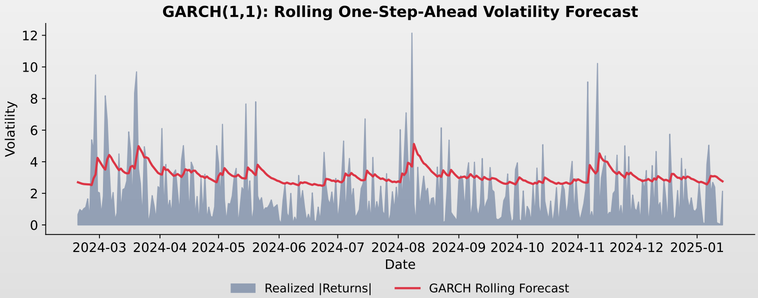
Rezultat

- **GARCH(1,1)** selectat pe baza celui mai mic MAE de validare pentru prognozele de volatilitate

Bitcoin: prognoza volatilității

Interpretare

- Zona umbrită: interval de încredere 95% al prognozei de volatilitate
- GARCH(1,1) captează bine dinamica volatilității Bitcoin



TSA_ch10_garch_forecast

Bitcoin: împărțirea datelor și staționaritate

Împărțirea datelor

Set	Perioadă	N
Antrenament (70%)	2019-01 – 2023-03	1.543
Validare (15%)	2023-04 – 2024-02	333
Test (15%)	2024-03 – 2025-01	329
Total		**2.205**

Teste de staționaritate (ADF – Augmented Dickey-Fuller)

Serie	ADF	Rezultat
Prețuri	$p = 0.50$	Non-staționară
Randamente	$p < 0.01$	Staționară

- $\Rightarrow$ Modelăm **randamente**, nu prețuri

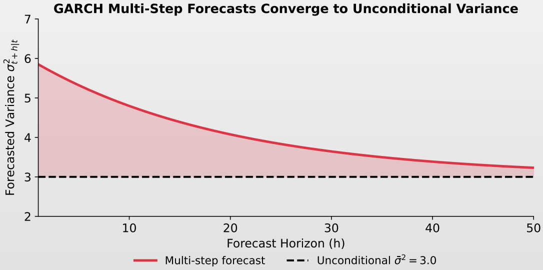
De ce contează staționaritatea

- **GARCH**: necesită input slab staționar
- **Prețuri vs Randamente**: prețuri = random walk, randamente = staționare

GARCH: prognozele multi-step converg

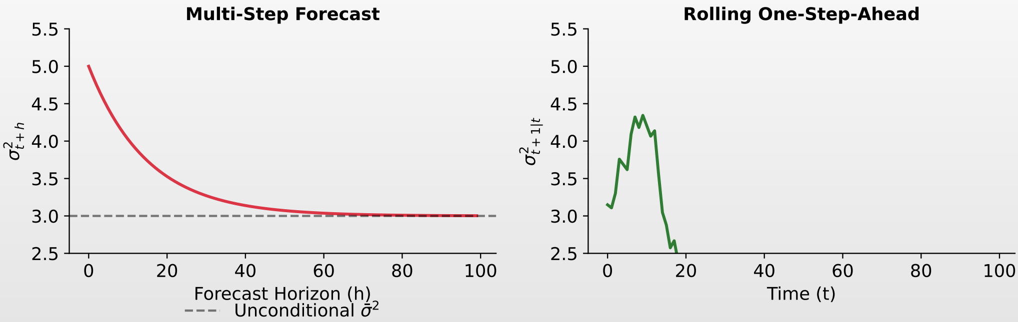
Concluzie

- Prognozele multi-step converg la $\bar{\sigma}^2 = \frac{\omega}{1-\alpha-\beta}$
- Soluția: prognoze rolling one-step-ahead



TSA_ch10_garch_convergence

GARCH: soluția rolling one-step-ahead



TSA_ch10_rolling_vs_multistep

GARCH: Distribuții pentru inovații

Model

$$r_t = \mu + \sigma_t z_t$$

- Variante pentru z_t: $\mathcal{N}(0,1)$ (normală) sau t_ν (cozi groase)

Bitcoin: dovezi empirice

- Curtoză reziduuri: **13.81** (Normal = 3)
- Skewness: -0.29
- Jarque-Bera: 9085, $p < 0.001$
- Normalitatea **subestimează** riscul extrem

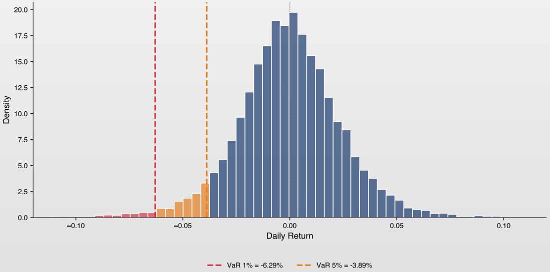
Student-t: alegerea corectă

- $\hat{\nu} = 2.96$ grade de libertate
- AIC Normal: 9769 vs Student-t: **9260**
- ΔAIC = 509 — dovadă **copleșitoare**
- Cozi groase = estimări VaR **mai realiste**

VaR și ES: ilustrație grafică

Interpretare

- VaR (Value at Risk – Valoarea la Risc) 1% = pierderea depășită doar în 1% din cazuri
- Zona roșie = pierderi extreme (dincolo de VaR)

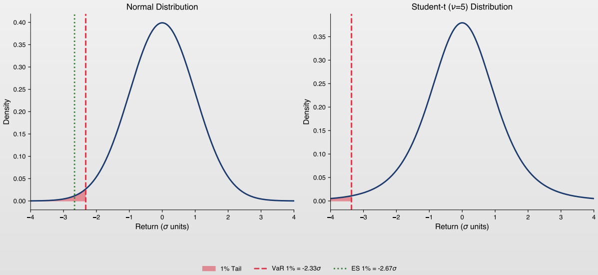


TSA_ch5_var_plot

VaR vs expected shortfall: normal vs Student-t

Interpretare

- ES (Expected Shortfall – Pierderea Așteptată) măsoară pierderea medie când VaR este depășit
- Student-t: VaR și ES mai mari decât sub distribuția normală



TSA_ch5_var_es

Value at Risk: exemplu numeric

Calculul VaR: Portofoliu 1M EUR, $\hat{\sigma}_{T+1} = 1.5\%$

Nivel	z_α	VaR (%)	VaR (EUR)
5% (1 zi)	1.645	2.47%	24.675
1% (1 zi)	2.326	3.49%	34.890
1% (10 zile)	$2.326 \cdot \sqrt{10}$	11.03%	110.314

Scalare pentru perioade mai lungi

- $\text{VaR}_{h \text{ zile}} = \text{VaR}_{1 \text{ zi}} \cdot \sqrt{h}$ (presupune randamente i.i.d.)

Value at Risk: distribuție Student-t

De ce Student-t?

- Normala **subestimează** riscul de coadă; randamentele au **cozi groase** (curtoză > 3)
- Student-t capturează mai bine extremele

Comparație VaR 1% (1 zi) pentru $\sigma = 1.5\%$, portofoliu $= 1$M EUR

Distribuție	**Cuantilă**	**VaR (EUR)**
Normal	2.326	34.890
Student-t ($\nu = 10$)	2.764	41.460
Student-t ($\nu = 6$)	3.143	47.145
Student-t ($\nu = 4$)	3.747	56.205

- Cu $\nu = 6$ (tipic pentru acțiuni), VaR este cu **35% mai mare** decât cel normal!

VaR: exemplu complet cu GARCH

Procedura de calcul VaR

1. Estimează modelul GARCH(1,1) cu distribuție Student-t
2. Obține prognoza volatilității: $\hat{\sigma}_{T+1}$
3. Calculează VaR: $\text{VaR}_\alpha = t_\alpha(\nu) \cdot \hat{\sigma}_{T+1}$

Exemplu: S&P 500

- Parametri estimați: $\alpha = 0.088$, $\beta = 0.900$, $\nu = 6.4$
- Volatilitate prognozată: $\hat{\sigma}_{T+1} = 1.2\%$, Portofoliu: 10.000.000 EUR
- **VaR 1% (1 zi):** $\text{VaR} = 3.05 \times 0.012 \times 10.000.000 =$ **366.000 EUR**

Ce este VaR backtesting?

Definiție

- **Backtesting** = verificarea ex-post a calității modelului VaR
- Compară pierderile realizate cu pragul VaR prognozat
  - O **încălcare** (violation) apare când $r_t < -\text{VaR}_t$

Principiul Backtesting-ului

- Indicatorul de încălcare: $I_t = 1(r_t < -\text{VaR}_{\alpha,t})$
- Pentru un model corect la nivel α:
  - Frecvența: $\hat{p} = \frac{1}{T}\sum I_t \approx \alpha$; încălcări **independente**
- VaR 1% pe 250 zile $\Rightarrow$ așteptăm $\sim$2.5 încălcări/an

Importanță

- Cerință regulamentară **Basel III/IV** pentru bănci: backtesting obligatoriu

Testul Kupiec (1995): acoperire necondiționată

Ipoteze

- H_0: Rata de încălcare este egală cu nivelul VaR ($p = \alpha$)
- H_1: Rata de încălcare diferă de nivelul VaR ($p \neq \alpha$)

Statistica de test (Likelihood Ratio)

- **Formula**: $LR_{uc} = -2\ln\left[\frac{\alpha^x(1-\alpha)^{T-x}}{\hat{p}^x(1-\hat{p})^{T-x}}\right] \sim \chi^2(1)$
- **Notație**: x = nr. încălcări, T = nr. observații, $\hat{p} = x/T$

Exemplu

- VaR 1%, $T = 250$ zile, $x = 5$ încălcări: $\hat{p} = 2\%$
  - Prea multe încălcări $\Rightarrow$ modelul **subestimează** riscul
- VaR 1%, $T = 250$ zile, $x = 1$ încălcare: $\hat{p} = 0.4\% \Rightarrow$ acceptabil

Testul Christoffersen (1998): acoperire condiționată

Motivație

- Kupiec testează doar **frecvența** încălcărilor
- Nu detectează **clusterizarea** încălcărilor (încălcări consecutive)
  - Dacă încălcările apar în clustere $\Rightarrow$ modelul nu captează dinamica volatilității

Testul de independență + acoperire condiționată

- **Formula**: $LR_{cc} = LR_{uc} + LR_{ind} \sim \chi^2(2)$
- LR_{ind}: testează independența încălcărilor
- Un model bun: încălcări rare **și** distribuite uniform în timp

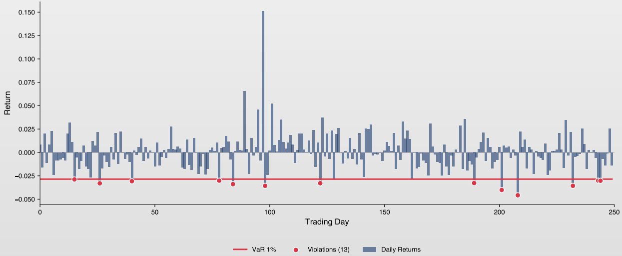
Recomandare

- Folosește **ambele** teste: Kupiec (frecvență) + Christoffersen (independență)

VaR backtesting: vizualizare

Interpretare

- Linia roșie: pragul VaR 1% estimat cu GARCH(1,1)
- Punctele roșii: 13 încălcări din 250 zile ($\hat{p} = 5.2\%$)
  - **Zonă roșie Basel** $\Rightarrow$ modelul subestimează semnificativ riscul
  - Soluții: distribuție Student-t, model EGARCH, sau nivel VaR mai conservator



TSA_ch5_backtest

Backtesting VaR: semaforul Basel

Zonele de semaforizare Basel III/IV

Zonă	Încălcări/250 zile	Interpretare	Penalizare
Verde	0–4	Model acceptabil	Fără penalizare
Galben	5–9	Necesită investigare	Factor k crește
Roșu	≥ 10	Model inadecvat	Penalizare maximă

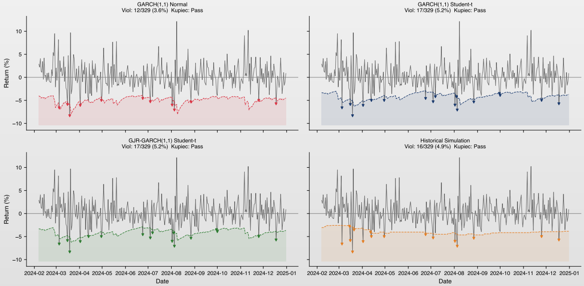
Exemplu practic

- Portofoliu cu VaR 1%: 250 zile de backtesting
- 3 încălcări $\Rightarrow$ **Zonă verde** $\Rightarrow$ model acceptabil
- 7 încălcări $\Rightarrow$ **Zonă galbenă** $\Rightarrow$ revizuire necesară
- 13 încălcări $\Rightarrow$ **Zonă roșie** $\Rightarrow$ model respins

Aplicație: VaR rolling pe multiple modele

Metodologie

- VaR rolling one-step-ahead: $\text{VaR}_{t+1}^{\alpha} = \mu + \hat{\sigma}_{t+1} \cdot z_{\alpha}$
- 4 modele comparate pe setul de test Bitcoin (329 zile, 2024)



TSA_ch10_empirical_slides

VaR Backtesting: comparație modele

Rezultate Bitcoin — VaR 5% (T = 329 zile, așteptat: 16,4 încălcări)

Model	Încălcări	Rată	Kupiec p	Chr. p	Concluzie
GARCH(1,1)-N	12	3,6%	0,238	1,000	Prea conservator
GARCH(1,1)-t	**17**	**5,2%**	**0,890**	0,272	**≈ 5% target**
GJR-GARCH(1,1)-t	17	5,2%	0,890	0,272	$\gamma \approx 0$ (simetric)
Hist. Simulation	**16**	**4,9%**	**0,909**	0,216	**≈ 5% target**

Concluzii

- **Student-t**: acoperire perfectă (5,2% ≈ 5%)
- Normal: prea conservator (3,6% < 5%)
- GJR ≈ GARCH: levier absent pt. Bitcoin
- Toate trec Kupiec **și** Christoffersen

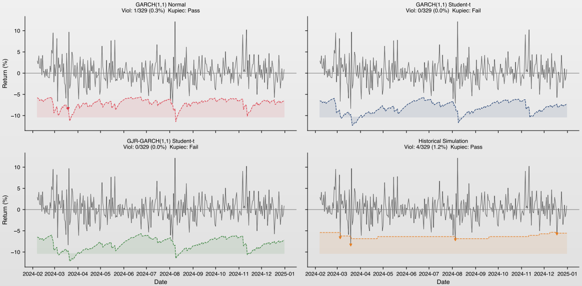
Lecții practice

- GARCH-t și HistSim: ambele ≈ 5% target
- Simularea istorică: alternativă simplă și eficientă
- Testarea statistică formală (Kupiec, Christoffersen) este **obligatorie**

Aplicație: VaR 1% rolling pe multiple modele

Metodologie

- VaR rolling one-step-ahead la nivel $\alpha = 1\%$ (risc extrem)
- Aceleași 4 modele; așteptat: $T \times 0{,}01 = 3{,}3$ încălcări



TSA_ch10_empirical_slides

VaR 1% Backtesting: comparație modele

Rezultate Bitcoin — VaR 1% (T = 329 zile, așteptat: 3,3 încălcări)

Model	Încălcări	Rată	Kupiec p	Chr. p	Concluzie
GARCH(1,1)-N	1	0,3%	0,137	1,000	Acceptabil
GARCH(1,1)-t	0	0,0%	**0,010**	1,000	**Respins**
GJR-GARCH(1,1)-t	0	0,0%	**0,010**	1,000	**Respins**
Hist. Simulation	**4**	**1,2%**	**0,704**	1,000	$\approx$ **1% target**

Concluzii VaR 1%

- GARCH-t/GJR-t: 0 încălcări $\Rightarrow$ Kupiec **respinge** ($p = 0{,}01$)
- Cozi Student-t prea groase ($\nu \approx 3$) $\Rightarrow$ VaR 1% **prea conservator**
- **Hist. Simulation**: 1,2% $\approx$ 1% — singurul model precis

Lecție: 5% vs 1%

- La 5%: GARCH-t și HistSim ambele excelente
- La 1%: GARCH-t **respins** (prea conservator)
- **Modelul optim depinde de nivelul α!**

Limitările GARCH și extensii moderne

Limitări

- Nu captează **jump-uri** (salturi bruște)
- Parametri constanți în timp
- Sensibil la distribuția aleasă
- Nu modelează **regimuri** diferite

Extensii

- **GJR-GARCH**: efect de levier
- **EGARCH**: șocuri asimetrice
- **Markov-Switching GARCH**: regimuri
- Volatilitate realizată (HAR – Heterogeneous Autoregressive)
- Hybrid GARCH + ML (Machine Learning)

De reținut

- GARCH este un **punct de plecare**, nu finalul modelării riscului

Bitcoin: Concluzii

Sumar

1. **Randamentele sunt staționare**; prețurile nu
2. **GARCH(1,1)** depășește variantele mai complexe
3. **Persistență înaltă** ($\alpha + \beta = 0,93$)
4. Volatilitatea este **predictibilă** chiar când randamentele nu sunt

Implicații practice

- Managementul riscului: VaR, Expected Shortfall
- Evaluarea opțiunilor necesită prognoze de volatilitate
- Optimizarea portofoliului cu risc variabil în timp

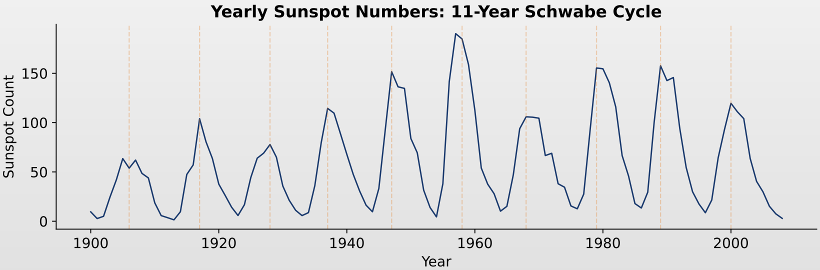
Lecții metodologice

- Simplu > Complex (GARCH(1,1) vs variante)
- Student-t **esențial** (ΔAIC = 509)
- Rolling one-step-ahead > multi-step
- Testarea formală (Diebold-Mariano) obligatorie

Pete solare: ciclul solar de 11 ani

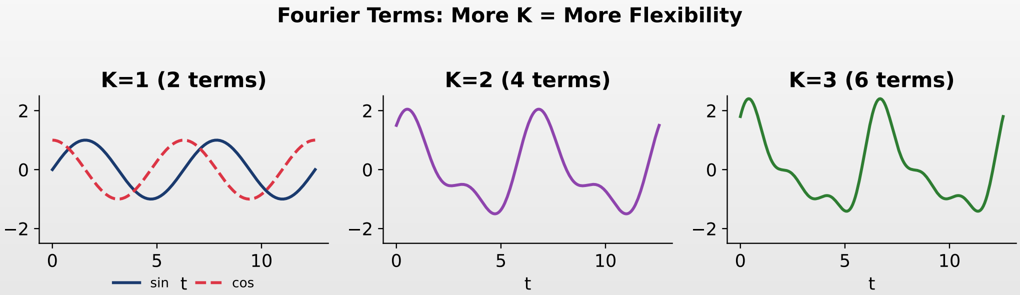
Observație

- Ciclu solar clar de ≈ 11 ani; amplitudine variabilă între cicluri
- ACF periodică ⇒ sezonalitate lungă, ideală pentru termeni Fourier



TSA_ch10_sunspots_acf

Termeni Fourier pentru sezonalitate



TSA_ch10_fourier_terms

Pete solare: selectarea modelului

Metodologie

- **Comparație**: $K = 1, 2, 3, 4$ armonici Fourier pe setul de validare

Împărțirea Datelor

Set	Perioadă	N
Antrenament (70%)	1900–1975	76
Validare (20%)	1976–1997	22
Test (10%)	1998–2008	11
Total		**109**

Comparație Modele

K	AIC	Val RMSE	
1	665,9	87,15	
2	668,0	86,92	
3	671,8	**86,81**	Cel mai bun
4	674,5	87,93	

Rezultat

- **K = 3** armonici Fourier selectate (6 parametri pentru ciclul de 11 ani)

Overfitting în alegerea lui K

Riscul de overfitting

- K prea mare = memorare ciclu istoric
- Modelul se potrivește pe zgomot, nu pe semnal
- Performanța pe test se **degradează**

Soluția: validare

- Selectăm K pe setul de **validare**
- Evaluăm pe **test** — neatins
- Trade-off: complexitate vs generalizare

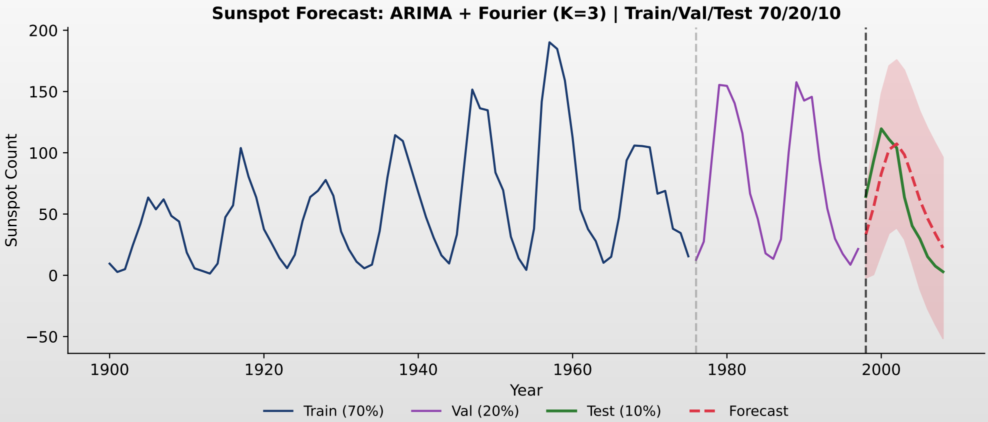
Fourier ≈ regresie periodică

- Fiecare armonic adaugă 2 parametri (sin, cos)
- $K = 3$: 6 parametri suplimentari
- $K = 6$: 12 parametri — risc supraajustare

La noi

- $K = 3$ minimizează Val RMSE
- $K = 4$ crește eroarea $\Rightarrow$ overfitting

Pete solare: rezultate prognoză



TSA_ch10_sunspot_forecast

Pete solare: concluzii

Când să folosiți termeni Fourier

- Perioada sezonieră s este **lungă** (ex: 11 ani, 52 săptămâni)
- SARIMA (Seasonal ARIMA) ar necesita prea multe lag-uri sezoniere
- Tiparul este **neted și periodic**
- Trebuie capturate cicluri multiple

Fourier vs SARIMA

	Fourier	**SARIMA**
Sezoane lungi	✓	×
Sezoane scurte	OK	✓
Parametri	$2K$	Mulți
Flexibilitate	Fixă	Adaptivă

Alegerea lui K

- **Strategie**: Începeți cu $K = 1$, creșteți progresiv
  - Opriți când eroarea de validare nu mai scade
  - K prea mare = supraajustare

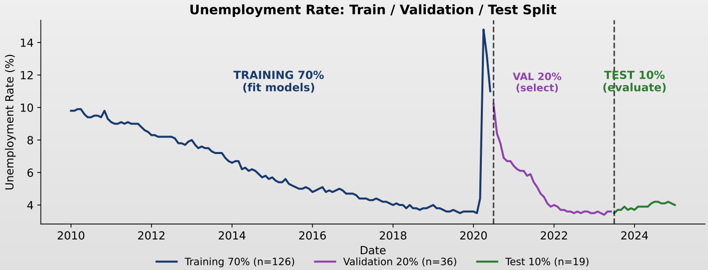
Aplicații

- **Domenii**: Cicluri climatice, cicluri de afaceri, fenomene astronomice

Șomajul: Train / Validation / Test Split

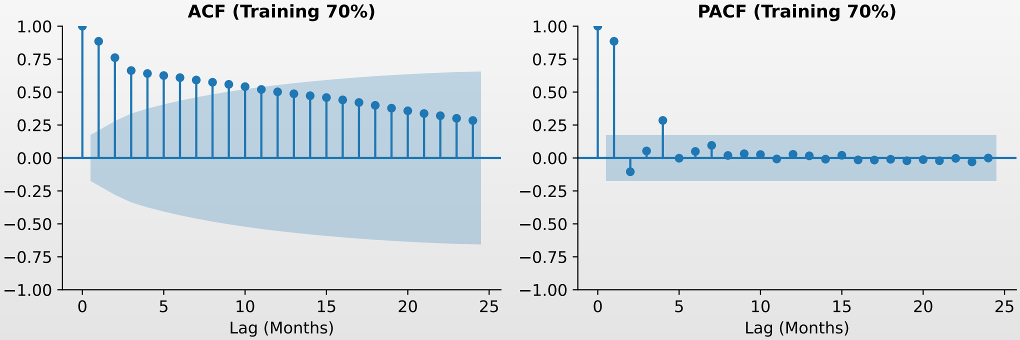
Metodologie

- **Training** (70%): Estimare modele
- **Validare** (20%): Selecție model
- **Test** (10%): Evaluare finală



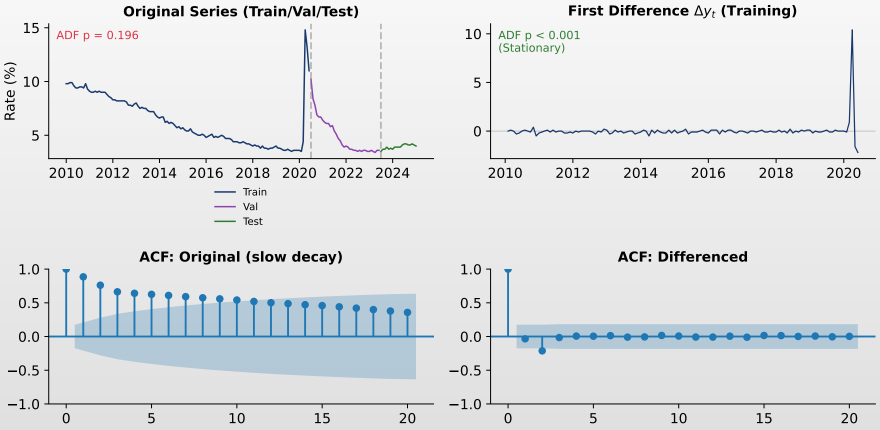
TSA_ch10_unemployment_train_val_test

Șomajul: analiză preliminară



TSA_ch10_unemployment_acf_pacf

Șomajul: teste de staționaritate



TSA_ch10_unemployment_stationarity

Rupturi structurale: abordare formală

Metode clasice

- **Chow Test**: ruptură la punct cunoscut
- **Bai–Perron**: rupturi multiple, necunoscute
- **CUSUM** (Cumulative Sum): detectare secvențială

Problemă

- ADF poate confunda **break** cu **unit root**
- Test Zivot–Andrews: ADF cu ruptură endogenă

Rezultat: Șomaj la COVID (martie 2020)

- Chow Test: $F = 21.73$, $p < 0.001$
- Ruptură structurală **confirmată**
- SARIMA: parametri constanți — risc
- Prophet: detectează changepoints automat

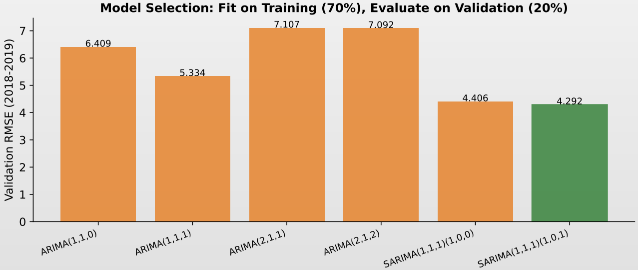
De reținut

- Modelul trebuie adaptat la **stabilitatea parametrilor**

Șomajul: selecția modelului (set validare)

Best: SARIMA(1,1,1)(1,0,0)$_{12}$

- Fit pe training (70%), evaluare pe validare (20%)
- Cel mai bun model selectat după Val RMSE minim



TSA_ch10_sarima_model_selection

Șomajul: parametrii SARIMA

SARIMA(1,1,1)(1,0,0)$_{12}$ estimat pe Train+Val (2010–2023)

- AR(1): $\phi_1 = -0,86$
- MA(1): $\theta_1 = 0,78$
- SAR(12): $\Phi_1 = -0,08$ (n.s.)

SARIMA(1,1,1)(1,0,1) - Fitted on Train+Val (85%)

Parameter	Coef	Std Err	P-value	Sig
ar.L1	0.8423	0.2084	0.0001	***
ma.L1	-0.9540	0.1973	0.0000	***
ar.S.L12	0.0326	4.5951	0.9943	
ma.S.L12	-0.0113	4.6087	0.9980	
sigma2	0.8122	0.0608	0.0000	***

TSA_ch10_sarima_parameters

Testul Ljung-Box pentru autocorelația reziduurilor

Definiție 4 (Testul Ljung-Box)

Pentru reziduurile $\hat{\varepsilon}_t$ cu autocorelații eșantion $\hat{\rho}_k$, statistica de test:

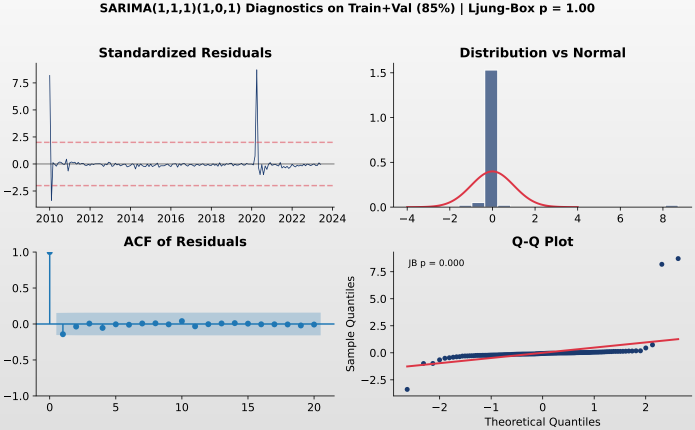
$$Q(h) = n(n+2)\sum_{k=1}^{h} \frac{\hat{\rho}_k^2}{n-k} \overset{H_0}{\sim} \chi^2(h-p-q)$$

unde p, q sunt ordinele ARMA. H_0: Reziduurile sunt zgomot alb.

Interpretare

- Q mare (p-value mic): Respingem H_0, reziduurile au structură
- Q mic (p-value mare): Nu respingem H_0, modelul este adecvat
- Regulă practică: $h = \min(10, n/5)$ pentru ordinul lag-ului

Șomajul: Diagnosticare SARIMA

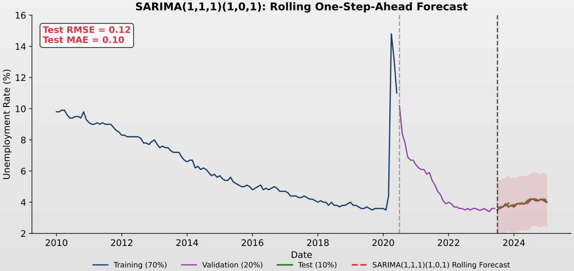


TSA_ch10_sarima_diagnostics

Șomajul: prognoza rolling SARIMA

Problemă: Ruptura structurală

- Prognoză rolling one-step-ahead (re-estimare la fiecare t)
- **Test RMSE = 0,12**



TSA_ch10_sarima_forecast

Modelul Prophet

Definiție 5 (Descompunerea Prophet)

- **Model**: $y_t = g(t) + s(t) + h(t) + \varepsilon_t, \quad \varepsilon_t \sim N(0, \sigma^2)$
- **Componente**: $g(t)$ = trend, $s(t)$ = sezonalitate, $h(t)$ = sărbători

Detectare puncte de schimbare

- Selectare automată a locațiilor
- `changepoint_prior_scale` controlează flexibilitatea

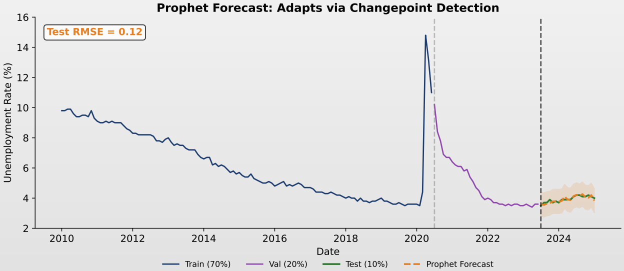
Avantaje

- Gestionează date lipsă
- Componente interpretabile
- Robust la outlieri

Șomajul: rezultate prognoză Prophet

Concluzie

- **Prophet**: se adaptează prin detectare changepoint
- **Test RMSE** = 0,58



TSA_ch10_unemployment_forecast

Șomajul: Ajustarea modelului

Ajustarea hiperparametrilor

- Ajustăm `changepoint_prior_scale` pe setul de validare

Împărțirea Datelor

Set	Perioadă	N
Antrenament (70%)	2010-01 – 2020-06	126
Validare (20%)	2020-07 – 2023-06	36
Test (10%)	2023-07 – 2025-01	19
Total		**181**

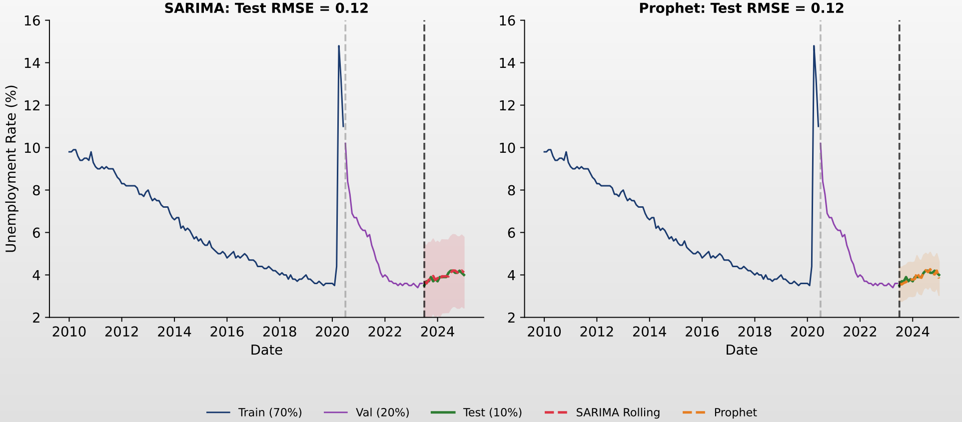
Comparație Scale

Scale	Val RMSE	
0,01	4,21	
0,05	3,89	
0,10	**3,52**	Cel mai bun
0,30	3,67	
0,50	3,81	

Interpretare

- Scale = 0,10 echilibrează flexibilitatea (captarea șocului COVID) cu stabilitatea

Șomaj: comparație SARIMA vs Prophet



TSA_ch10_prophet_vs_sarima_unemployment

Prophet: când să-l folosești

Cazuri de utilizare ideale

- Date de business cu **sărbători**
- **Valori lipsă** prezente
- Nevoie de componente **interpretabile**
- Prognoze cu **benzi de incertitudine**

Atenție: Rupturi structurale

- Prophet gestionează rupturile prin changepoints
- Dar **SARIMA l-a depășit** la șomaj (RMSE: 0,12 vs 0,58)
- Validați întotdeauna pe date out-of-sample!

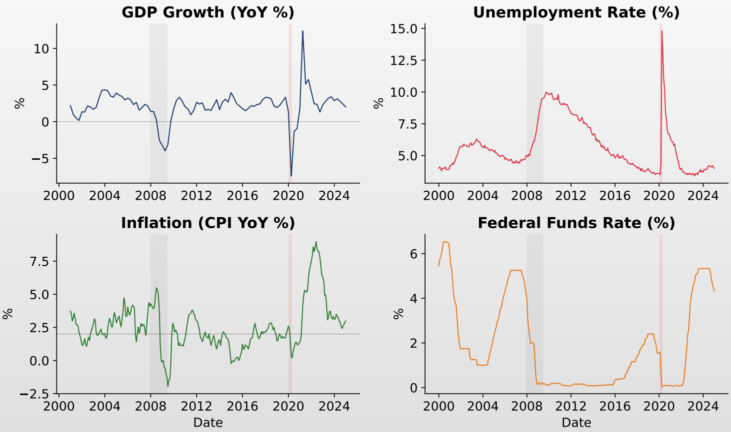
Prophet vs ARIMA (AutoRegressive Integrated Moving Average)

	Prophet	**ARIMA**
Changepoints	✓	×
Date lipsă	✓	×
Sărbători	✓	×
Viteză	Rapidă	Moderată
Interpretabil	✓	×

Parametri principali

- `changepoint_prior_scale`: flexibilitate
- `seasonality_prior_scale`: netezime

VAR: date economice multivariate



TSA_ch10_economic_vars

Specificarea modelului VAR

Definiție 6 (VAR(p) – Vector Autoregression (Autoregresie Vectorială))

- **Date**: Pentru K variabile $\mathbf{y}_t = (y_{1t}, \ldots, y_{Kt})'$:

$$\mathbf{y}_t = \mathbf{c} + \mathbf{A}_1 \mathbf{y}_{t-1} + \mathbf{A}_2 \mathbf{y}_{t-2} + \cdots + \mathbf{A}_p \mathbf{y}_{t-p} + \mathbf{u}_t$$

- **Notație**: $\mathbf{A}_i$ sunt matrici de coeficienți $K \times K$, $\mathbf{u}_t \sim N(0, \Sigma)$

Pentru sistemul nostru cu 4 variabile

- **VAR(2)**: 4 constante
- $2 \times 4 \times 4 = 32$ coeficienți AR
- **36 parametri total**

Selectarea lag-ului

- Folosim criterii informaționale:
  - AIC: Tinde să supraajusteze
  - **BIC**: Mai simplu
  - Cross-validare pe date păstrate

Criterii informaționale pentru selectarea modelului

Definiție 7 (Criteriile Informaționale Akaike și Bayesian)

Pentru un model cu log-verosimilitate $\mathcal{L}$, k parametri și n observații:

$$\text{AIC} = -2\mathcal{L} + 2k$$
$$\text{BIC} = -2\mathcal{L} + k\ln(n)$$

AIC

- Asimptotic eficient
- Poate supraajusta cu n mic
- Minimizează eroarea de predicție

BIC

- Consistent (găsește modelul adevărat)
- Penalizare mai mare: $\ln(n) > 2$ dacă $n \geq 8$
- Mai parcimonios

VAR: selectarea lag-ului și estimare

Criterii informaționale

Lag	BIC	
1	-4,810	
2	**-5,178**	Cel mai bun
3	-4,633	
4	-4,614	

Împărțirea datelor

Set	Perioadă	N
Antrenament (70%)	2001-T1 – 2017-T4	67
Validare (20%)	2018-T1 – 2022-T4	20
Test (10%)	2023-T1 – 2025-T2	10
Total		**97**

Verificare validare

- VAR(2) obține și cel mai mic RMSE de validare

Stabilitatea modelului VAR

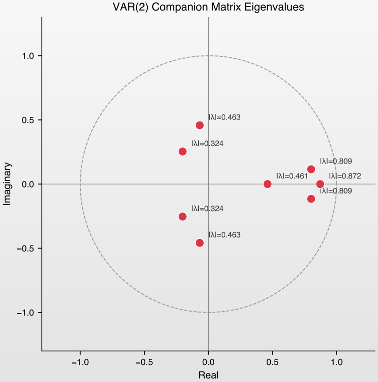
Condiția de stabilitate

- Toate valorile proprii ale **matricei companion**: $$|\lambda_i| < 1, \quad \forall i$$

Rezultate VAR(2) — date economice

$\lvert\lambda_1\rvert, \lvert\lambda_2\rvert$	0.324
$\lvert\lambda_3\rvert, \lvert\lambda_4\rvert$	0.463
$\lvert\lambda_5\rvert$	0.461
$\lvert\lambda_6\rvert$	**0.872**
$\lvert\lambda_7\rvert, \lvert\lambda_8\rvert$	0.810

- Max $|\lambda| = 0.872 < 1$ — **stabil**



TSA_ch10_empirical_slides

VAR vs VECM (Vector Error Correction Model): cointegrare

Problemă

- Dacă variabilele sunt $I(1) \Rightarrow$ VAR pe niveluri produce regresii spurioase

Definiție 8 (VECM)

$$\Delta y_t = \Pi y_{t-1} + \sum_{i=1}^{p-1} \Gamma_i \Delta y_{t-i} + u_t, \quad \Pi = \alpha\beta'$$

Johansen Test — date economice

r	Trace	CV 5%	Respins?
0	**64.09**	47.85	**Da**
1	24.03	29.80	Nu
2	11.89	15.49	Nu
3	1.28	3.84	Nu

- **1 relație de cointegrare** găsită
- VECM e mai adecvat decât VAR pe niveluri

De reținut

- VAR pe diferențe: pierde relația pe termen lung; VECM: o păstrează prin $\Pi = \alpha\beta'$

Cauzalitatea Granger: Rezultate empirice

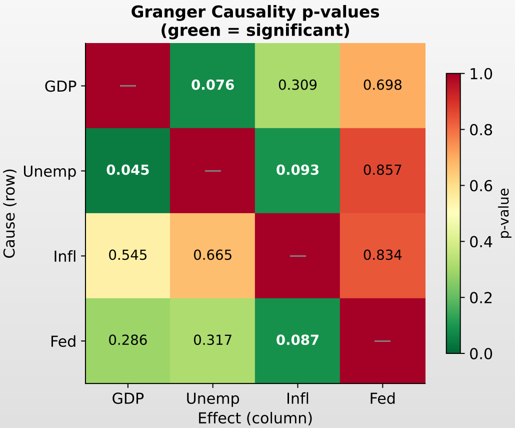
Interpretare

- Fiecare celulă arată p-value-ul testului Granger: rând $\rightarrow$ coloană
- Verde: $p < 0{,}10$ (semnificativ la 10%)

Concluzii economice

- Șomaj $\succ$ PIB ($p = 0,045$): Legea lui Okun
- Fed $\succ$ Inflație ($p = 0,087$): Transmisia politicii monetare
- PIB $\succ$ Șomaj: Dovezi slabe

Cauzalitatea Granger: heatmap *p*-values



TSA_ch10_granger_heatmap

Cauzalitatea Granger: Definiție formală

Definiție 9 (Cauzalitatea Granger)

X **Granger-cauzează** Y dacă, pentru un $h > 0$:

$$\mathbb{E}\left[(Y_{t+h} - \mathbb{E}[Y_{t+h}|\mathcal{F}_t^{X,Y}])^2\right] < \mathbb{E}\left[(Y_{t+h} - \mathbb{E}[Y_{t+h}|\mathcal{F}_t^{Y}])^2\right]$$

unde $\mathcal{F}_t^{X,Y}$ include valorile trecute ale lui X și Y, iar $\mathcal{F}_t^{Y}$ include doar trecutul lui Y.

Observație importantă

- Cauzalitatea Granger = **cauzalitate predictivă**, nu cauzalitate reală
- "X Granger-cauzează Y" $\Rightarrow$ X conține informație utilă pentru prognoza lui Y
- **Nu** implică faptul că X cauzează Y în sens structural

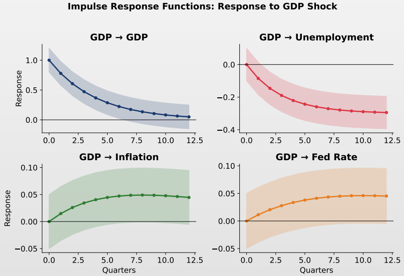
Procedura de testare

- Testul F (sau Wald): H_0: coeficienții lag-urilor lui X sunt simultan zero în ecuația lui Y
- Respingem $H_0 \Rightarrow X$ Granger-cauzează Y

Funcții de răspuns la impuls – IRF (Impulse Response Functions)

Efecte

- $\uparrow$ PIB $\Rightarrow$ $\downarrow$ Șomaj (Okun), $\uparrow$ Inflație (cerere), Fed crește rata

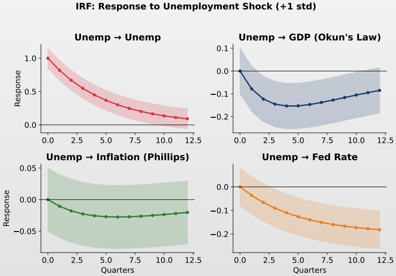


TSA_ch10_irf_gdp_shock

IRF: șoc șomaj

Efecte

- $\uparrow$ Șomaj $\Rightarrow$ $\downarrow$ PIB (Okun), $\downarrow$ Inflație (Phillips), Fed reduce rata

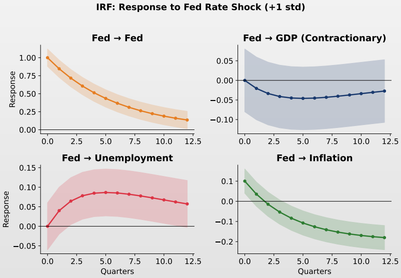


TSA_ch10_irf_unemp_shock

IRF: șoc rată Fed

Politică monetară

- Creștere rată $\Rightarrow$ PIB $\downarrow$, Șomaj $\uparrow$, Inflație $\downarrow$

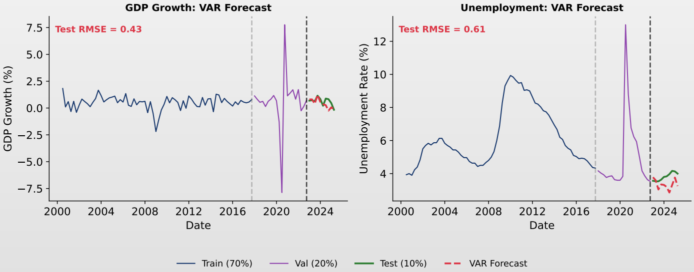


TSA_ch10_irf_fed_shock

VAR: Prognoza (Train/Val/Test)

Prognoză Rolling one-step-ahead

- VAR captează dinamică PIB-Șomaj
- Șocul COVID vizibil în perioadă validare (2020)



TSA_ch10_var_forecast

VAR: rezultate set test

Performanță set Test pe variabile

Variabilă	RMSE	MAE	Acur. Direcție
Creștere PIB	1,33	0,99	50%
Șomaj	0,64	0,52	50%
Inflație	1,56	1,12	60%
Rata Fed	2,59	2,45	80%
Medie	**1,53**	**1,27**	**60%**

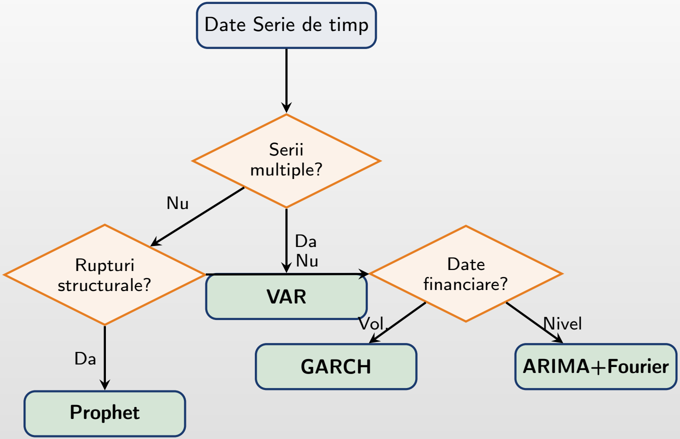
Puncte forte

- Captează dinamică între variabile
- Acuratețe direcțională bună
- Relații interpretabile

Limitări

- Mulți parametri (blestemul dimensionalității)
- Sensibil la selectarea lag-ului
- Perioada COVID dificilă

Cadrul de selectare a modelului



Sinteză: Comparația modelelor

Caracteristică	GARCH	Fourier	Prophet	VAR
Țintă	Volatilitate	Nivel	Nivel	Multiple
Sezonalitate	Nu	Da (lungă)	Da (multiplă)	Nu
Rupturi structurale	Nu	Nu	Da	Nu
Serii multiple	Nu	Nu	Nu	Da
Interpretabil	Mediu	Ridicat	Ridicat	Ridicat
Parametri	Puțini	$2K$	Auto	Mulți
Date lipsă	Nu	Nu	Da	Nu
Ideal pentru	Finanțe	Cicluri	Business	Macro

Concluzii empirice

- **GARCH**: Student-t > Normal (ΔAIC = 509)
- **Fourier**: $K = 3$ armonici, validat pe set de validare
- **Prophet**: adaptare la rupturi via changepoints
- **VAR**: interacțiuni macro semnificative (Granger)

Observație

- RMSE nu se compară între seturi de date diferite!
- Fiecare model excelează în domeniul său
- Alegerea: potrivirea model $\leftrightarrow$ date

Bune practici pentru prognoza aplicată

Metodologie

1. **Explorați** datele temeinic
2. **Testați** staționaritatea
3. **Împărțiți** train/validation/test
4. **Comparați** modele pe validare
5. **Raportați** metrici pe test

Sfaturi practice

- Începeți simplu (random walk, naiv)
- Adăugați complexitate doar dacă e necesar
- Vizualizați prognoze vs. valori reale
- Verificați reziduurile pentru tipare
- Raportați intervale de încredere

Greșeli frecvente

- Privirea în datele de test
- Supraajustare pe setul de antrenament
- Ignorarea ipotezelor modelului
- Neraportarea incertitudinii

Amintiți-vă

- "Toate modelele sunt greșite, dar unele sunt utile." — George E. P. Box

Prognoză vs Cauzalitate vs Decizie

Obiectiv	**Model**	**Focalizare**
Predicție pură	ARIMA / ML	Acuratețe out-of-sample
Risc financiar	GARCH	Volatilitate, VaR
Dinamici macro	VAR	Interacțiuni multivariate
Relații structurale	SVAR (Structural VAR) / VECM	Identificare cauzală
Regimuri	Markov Switching	Schimbări de regim

Concluzie

- Nu există model universal
- Există **potrivire între model și problemă**

Concluzii

1. **Metodologie Riguroasă**
   - Împărțirea train/validation/test previne supraajustarea
   - Setul de test trebuie să rămână neatins până la evaluarea finală
2. **Potriviți Modelul cu Datele**
   - Volatilitate financiară $\Rightarrow$ GARCH
   - Sezonalitate lungă $\Rightarrow$ Termeni Fourier
   - Rupturi structurale $\Rightarrow$ Prophet
   - Serii multiple $\Rightarrow$ VAR
3. **Interpretați Rezultatele cu Grijă**
   - Cauzalitate Granger $\neq$ cauzalitate adevărată
   - Performanța out-of-sample contează cel mai mult
   - Modelele mai simple funcționează adesea mai bine

Rolul AI în modelarea seriilor de timp

AI poate

- Genera cod pentru estimare și prognoză
- Selecta modele (AutoML, grid search)
- Combina prognoze (ensemble)
- Detecta anomalii și tipare

Dar nu poate

- Înlocui validarea statistică
- Detecta automat **data leakage**
- Garanta interpretare economică corectă
- Verifica ipotezele modelului

Principiu

- AI este **instrument**, nu autoritate
- Validarea statistică rămâne responsabilitatea cercetătorului

Exercițiu AI: Gândire critică

Prompt de testat în ChatGPT / Claude / Copilot

"Descarcă de pe FRED vânzările lunare cu amănuntul din SUA (seria RSXFS) din 2010-01 până în 2024-12 (180 observații). Fă o analiză completă a seriei de timp: descompunere, teste de staționaritate, selecție model (compară ETS, SARIMA și Prophet), prognoză pe 12 luni, evaluare cu RMSE/MAE/MASE pe un split temporal 70/15/15. Vreau cod Python de calitate publicabilă."

Exercițiu

1. Rulați prompt-ul într-un LLM (Large Language Model) la alegere și analizați critic răspunsul.
2. Urmează fluxul corect? (grafic → descompunere → test → model → diagnostic → prognoză)
3. Compară mai multe modele (ETS – Error, Trend, Seasonality; ARIMA; SARIMA) cu benchmark-uri adecvate?
4. Împărțirea train/test este făcută corect? Există scurgeri de date (data leakage)?
5. Discută limitările și ipotezele modelului ales?

Atenție

- Codul generat de AI poate rula fără erori și arăta profesional
- *Asta nu înseamnă că e corect*

Întrebarea 1

Întrebare

- Ce model alegeți pentru a prognoza volatilitatea randamentelor financiare?

Variante de răspuns

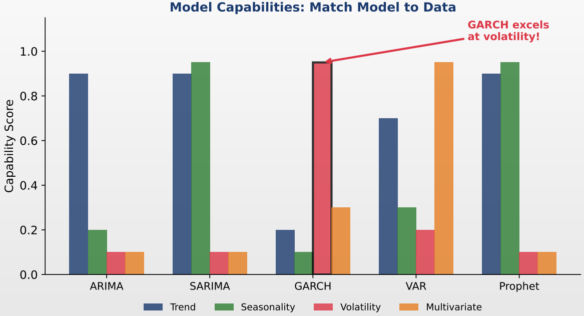
(A) ARIMA — captează tendințe și autocorelații

(B) GARCH — modelează varianța condiționată

(C) Prophet — detectează puncte de schimbare

(D) VAR — model multivariat pentru interdependențe

Întrebarea 1: Răspuns



Răspuns: (B)

- GARCH captează volatility clustering și riscul variabil în timp
- ARIMA modelează nivelul, Prophet sezonalitatea, VAR relațiile între serii
- Niciunul dintre celelalte nu modelează varianța direct

Întrebarea 2

Întrebare

- Un model SARIMA obține RMSE = 0,05 pe antrenament, dar RMSE = 2,30 pe test. Ce indică aceasta?

Variante de răspuns

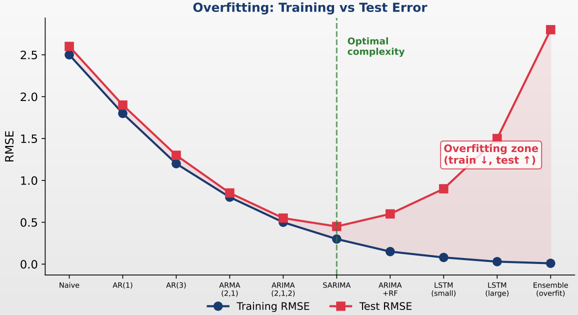
(A) Modelul este excelent — eroare mică pe antrenament

(B) Modelul suferă de overfitting — memorează zgomotul

(C) Setul de test este greșit — trebuie schimbat

(D) Diferența este normală — nu e nicio problemă

Întrebarea 2: Răspuns



Răspuns: (B)

- Raport de 46$\times$ între RMSE test și train $\Rightarrow$ overfitting sever
- Modelul se potrivește zgomotului din antrenament și nu generalizează
- Soluție: model mai simplu, validare pe set separat

Întrebarea 3

Întrebare

- De ce este importantă separarea datelor în train/validation/test?

Variante de răspuns

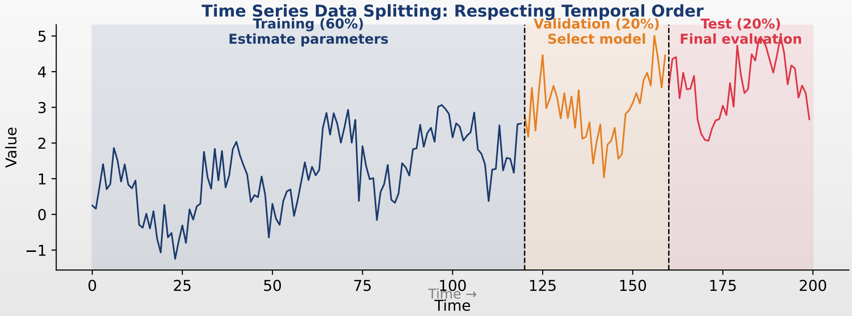
(A) Pentru a avea mai multe date de antrenament

(B) Pentru a preveni supraajustarea și a evalua corect

(C) Este doar o convenție, nu are importanță reală

(D) Pentru a reduce timpul de calcul

Întrebarea 3: Răspuns



Răspuns: (B)

- **Train**: estimează parametrii; **Validare**: selectează modelul; **Test**: evaluare finală nebiasată
- Amestecarea acestor roluri duce la estimări optimiste ale performanței

TSA_ch10_quiz3_train_val_test

Întrebarea 4

Întrebare

- Cauzalitatea Granger este echivalentă cu cauzalitatea reală (structurală)?

Variante de răspuns

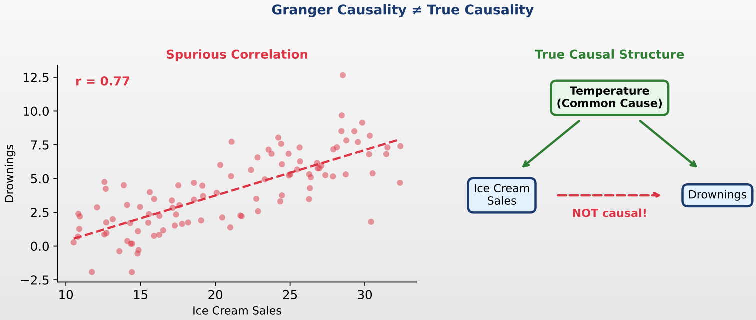
(A) Da — dacă X prezice Y, atunci X cauzează Y

(B) Nu — testează doar conținut predictiv, nu cauzalitate

(C) Depinde de numărul de lag-uri selectate

(D) Da, dacă p-value $< 0,05$

Întrebarea 4: Răspuns



Răspuns: (B)

- Testul Granger verifică dacă trecutul lui X îmbunătățește predicția lui Y
- Corelații false (ex: vânzări de înghețată și înecuri) pot trece testul din cauza cauzelor comune

TSA_ch10_quiz4_granger_causality

Întrebarea 5

Întrebare

- Ce model folosiți pentru o serie cu sezonalitate lungă (ex: $s = 365$ zile)?

Variante de răspuns

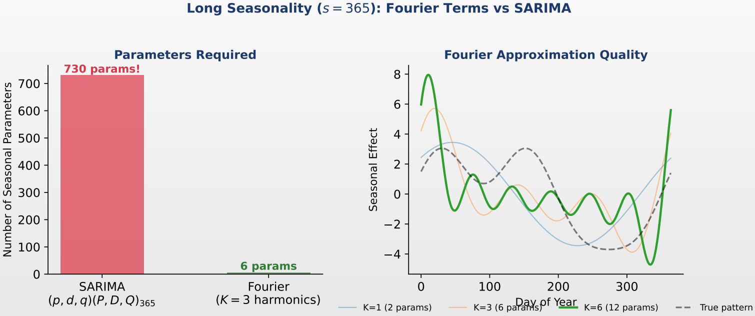
(A) SARIMA$(p, d, q)(P, D, Q)_{365}$

(B) GARCH — modelează variația

(C) ARIMA + Termeni Fourier sau Prophet/TBATS (Trigonometric, Box-Cox, ARMA, Trend, Seasonal)

(D) VAR cu 365 lag-uri

Întrebarea 5: Răspuns



Răspuns: (C)

- SARIMA_{365} necesită polinoame lag de ordin 365 — imposibil de estimat
- Termenii Fourier cu $K = 3$ folosesc doar 6 parametri (sin/cos)
- Prophet și TBATS gestionează sezonalități multiple automat

Bibliografie I

Manuale fundamentale (referințe comune tuturor capitolelor)

- Hamilton, J.D. (1994). *Time Series Analysis*, Princeton University Press.
- Hyndman, R.J., & Athanasopoulos, G. (2021). *Forecasting: Principles and Practice*, 3rd ed., OTexts.
- Shumway, R.H., & Stoffer, D.S. (2017). *Time Series Analysis and Its Applications*, 4th ed., Springer.

Lucrări de referință pe domenii

- Tsay, R.S. (2010). *Analysis of Financial Time Series*, 3rd ed., Wiley. (GARCH, VAR)
- Lütkepohl, H. (2005). *New Introduction to Multiple Time Series Analysis*, Springer. (VAR, VECM)
- Francq, C., & Zakoïan, J.-M. (2019). *GARCH Models*, 2nd ed., Wiley. (Volatilitate)

Bibliografie II

Abordări moderne și competiții de prognoză

- Petropoulos, F., et al. (2022). Forecasting: Theory and Practice, *International Journal of Forecasting*, 38(3), 845–1054.
- Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020). The M4 Competition, *International Journal of Forecasting*, 36(1), 54–74.
- Taylor, S.J., & Letham, B. (2018). Forecasting at Scale, *The American Statistician*, 72(1), 37–45.

Resurse online și cod

- **Quantlet**: `https://quantlet.com` ≻ Platformă de cod pentru metode cantitative
- **Quantinar**: `https://quantinar.com` ≻ Platformă de învățare pentru metode cantitative
- **GitHub TSA**: `https://github.com/QuantLet/TSA/tree/main/TSA_ch10` ≻ Cod Python pentru acest capitol

Vă Mulțumim!

Întrebări?

Materialele cursului sunt disponibile la: `https://danpele.github.io/Time-Series-Analysis/`

Quantlet Quantinar