



# Analiza și Prognoza Seriilor de Timp

Capitolul 10: Recapitulare Comprehensivă



Daniel Traian PELE

Academia de Studii Economice din Bucureşti

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 metri și validare încrucișată
4. Integrați cunoștințele din toate capitolele anterioare în practică



## Cuprins

### Fundamente

- Metodologia Prognozei
- Studiu de Caz 1: Volatilitatea Bitcoin (GARCH)
- Studiu de Caz 2: Ciclurile Petelor Solare (Fourier)

### Aplicații

- Studiu de Caz 3: Șomajul (Prophet)
- Studiu de Caz 4: Analiză Multivariată (VAR)
- Sinteză și Ghid
- Quiz



## Cadrul Train/Validation/Test

Train / Validation / Test Split



## 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

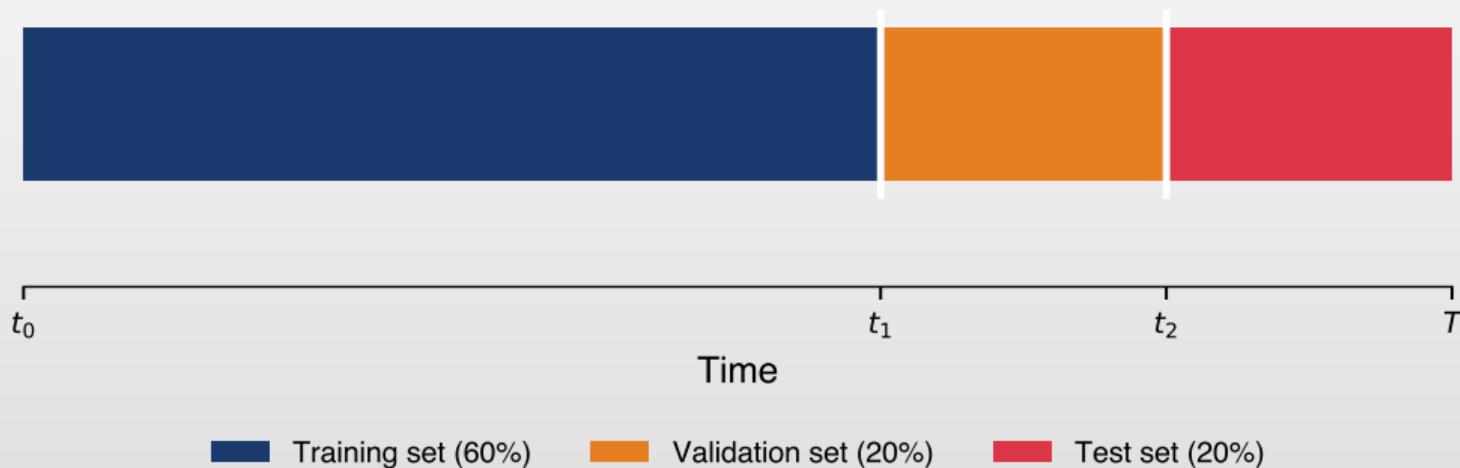
### Principiu cheie

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



## Cadrul Train/Validation/Test

Train / Validation / Test Split



Q TSA\_ch10\_train\_val\_test\_split



## Metrici de evaluare

### Definiție 1 (Metrici ale Erorii de Prognoză)

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

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

#### Când să folosim

- **RMSE:** Penalizează erorile mari
- **MAE:** Robust la outlieri
- **MAPE:** Independent de scală (%)

#### Atenție

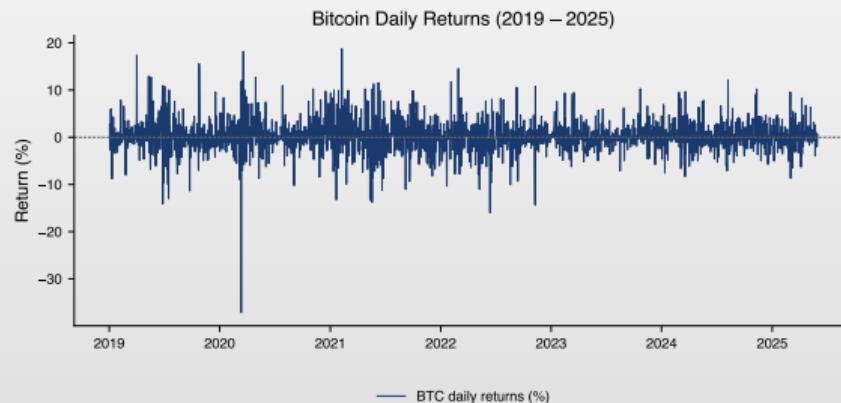
- MAPE nedefinit când  $y_t = 0$
- Comparați pe **același** set test
- Raportați metrici **out-of-sample**



## Bitcoin: volatility clustering

### Observație

- Randamentele mari tind să urmeze randamente mari, cele mici urmează cele mici
- Aceasta este **volatility clustering** ✎ fenomenul pe care GARCH îl captează



Q 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:  $\approx 2.200$  zile

### Fapte stilizate

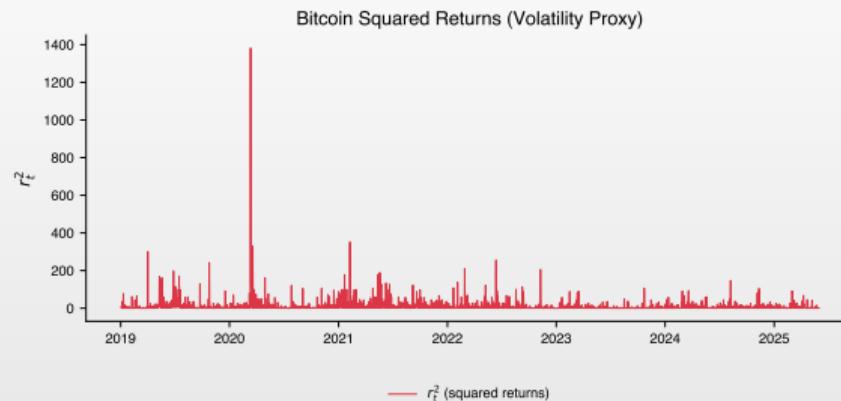
- Randamente: medie aproape zero
- Cozi groase (curtosis  $> 3$ )
- Clustering al volatilității

### Insight cheie

- Randamentele financiare** sunt de obicei:
  - ▶ Impredictibile în medie
  - ▶ Predictibile în varianță
- ➔ Focus pe **prognoza volatilității**



## Bitcoin: dovezi pentru GARCH



 TSA\_ch10\_btc\_acf\_squared

## Specificarea modelului GARCH

### Definiție 2 (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:** Efect de levier
- **EGARCH:** Șouri asimetrice

### Interpretare

- $\alpha$ : Impactul șourilor trecute
- $\beta$ : Persistența volatilității
- $\alpha + \beta \approx 1$ : Persistență înaltă



## GARCH: Stationaritate și varianta neconditionată

**Teoremă 1 (Stationaritatea 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  $h \rightarrow \infty$ :  $\mathbb{E}_t[\sigma_{t+h}^2] \rightarrow \bar{\sigma}^2$  cu rata  $(\alpha_1 + \beta_1)^h$ .



## Bitcoin: selectarea modelului pe setul de validare

### Metodologie

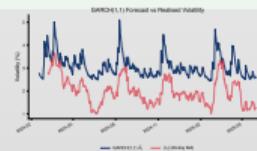
- Estimăm fiecare model pe datele de antrenament, evaluăm pe setul de validare

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: împărțirea datelor și staționaritate

### Împărțirea datelor

Set	Perioadă	N
Antrenament (70%)	2019-01 – 2023-03	1.543
Validare (20%)	2023-03 – 2024-06	441
Test (10%)	2024-06 – 2025-01	221
<b>Total</b>		<b>2.205</b>

### Teste de staționaritate

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

✓ Modelăm **randamente**, nu prețuri

### De ce contează staționaritatea

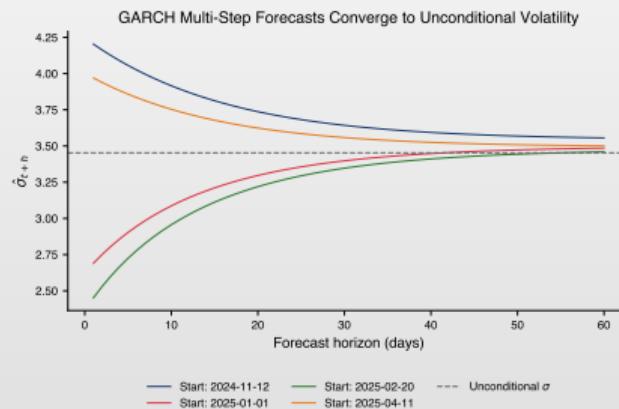
- GARCH**: necesită input slab staționar
- Prețuri vs Randamente**: Prețurile urmează random walk, randamentele sunt staționare



## GARCH: prognozele multi-step converg

### Insight cheie

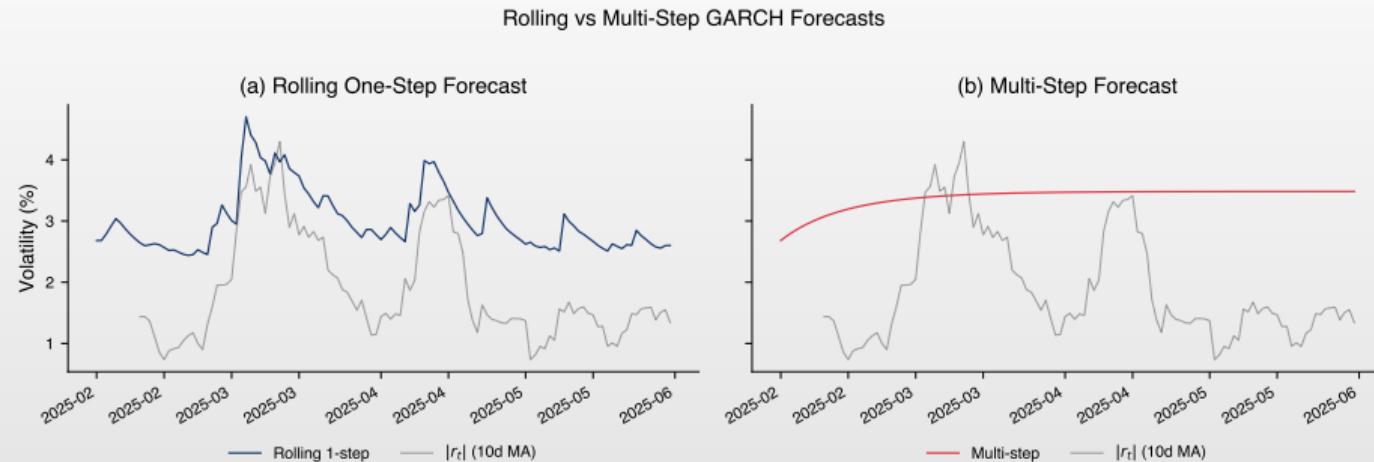
- ◻ Prognozele multi-step converg la  $\bar{\sigma}^2 = \frac{\omega}{1-\alpha-\beta}$
- ◻ Soluția: prognoze rolling one-step-ahead



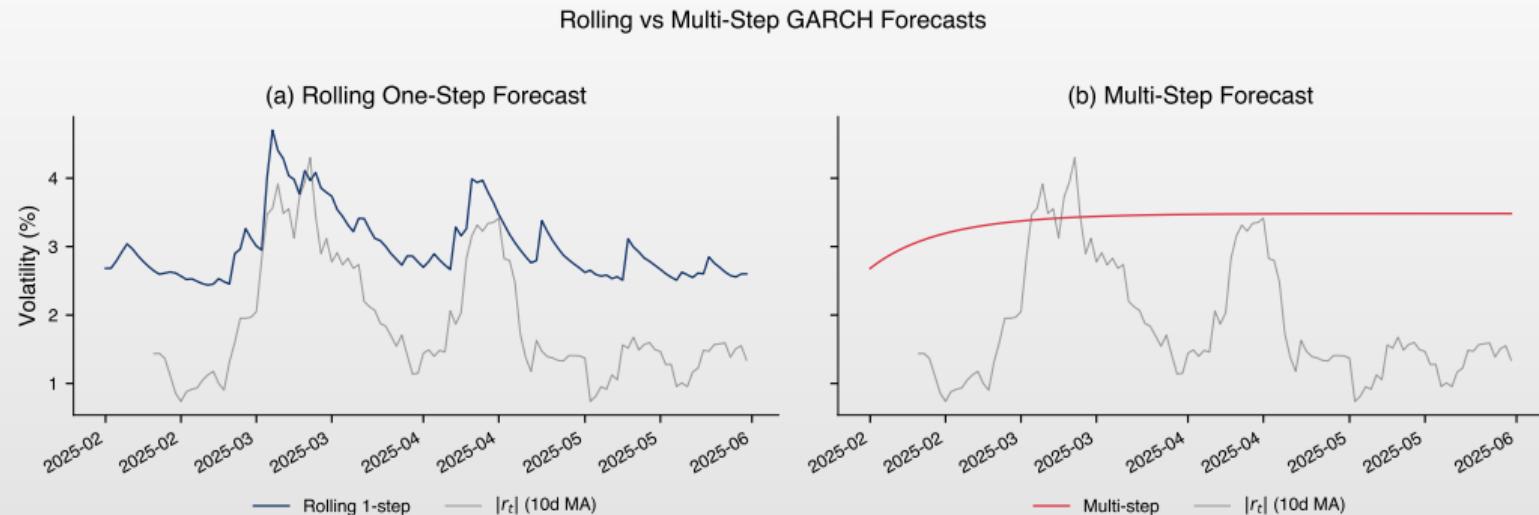
Q TSA\_ch10\_garch\_convergence



## GARCH: soluția rolling one-step-ahead



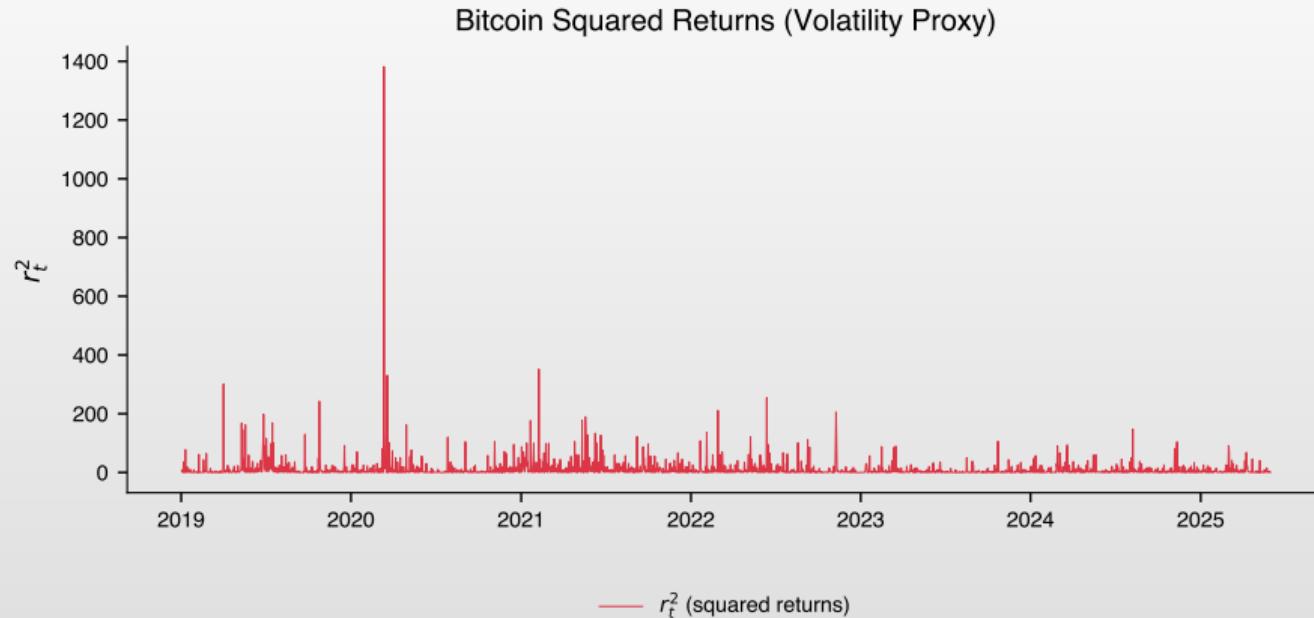
## GARCH: soluția rolling one-step-ahead



Q TSA\_ch10\_rolling\_vs\_multistep



## Bitcoin: Fapte stilizate GARCH



## Bitcoin: Concluzii cheie

### Sumar

1. Rendamentele 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

### Limitări

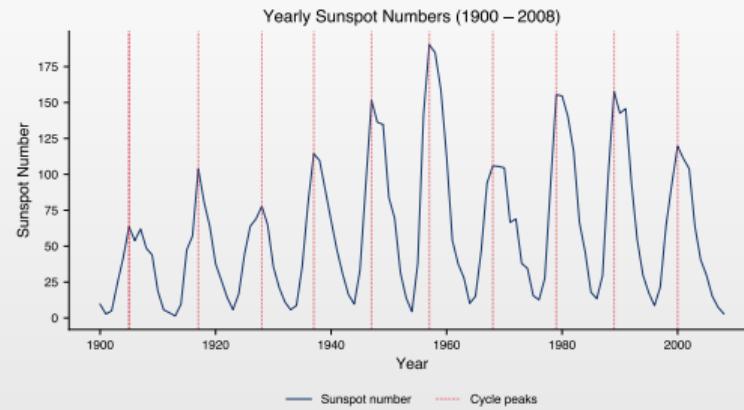
- GARCH presupune șocuri simetrice
- Nu captează salturi
- Distribuția normală poate fi restrictivă

### Extensiile

- Inovații Student-t
- Volatilitate realizată
- Modele HAR



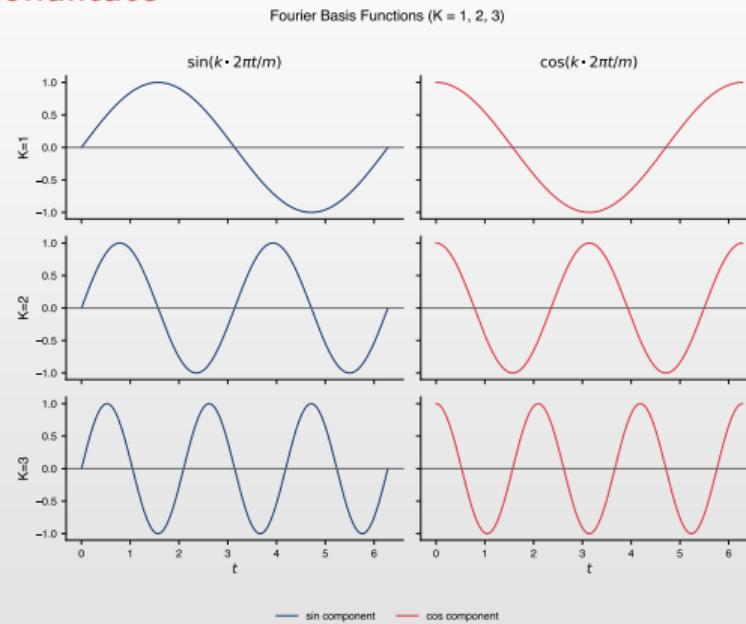
## Pete solare: ciclul solar de 11 ani



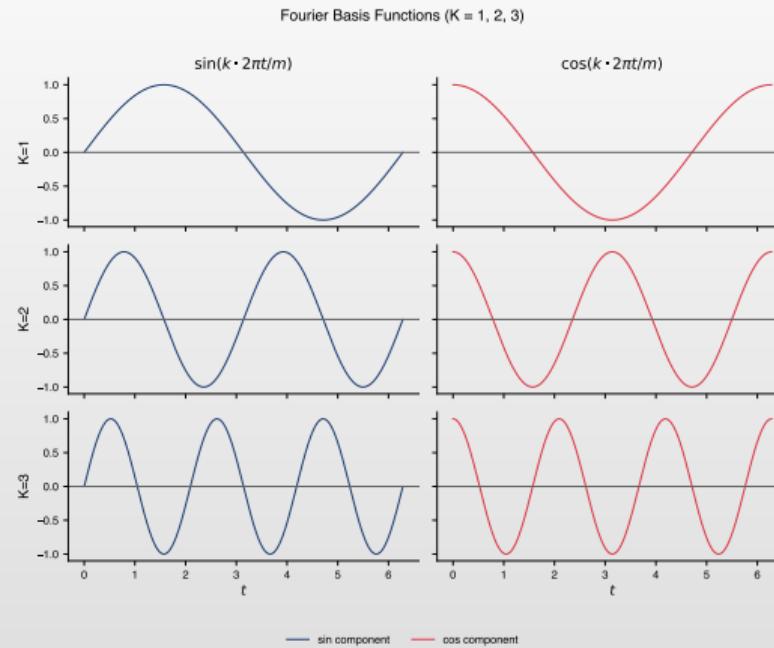
 **TSA\_ch10\_sunspots\_acf**



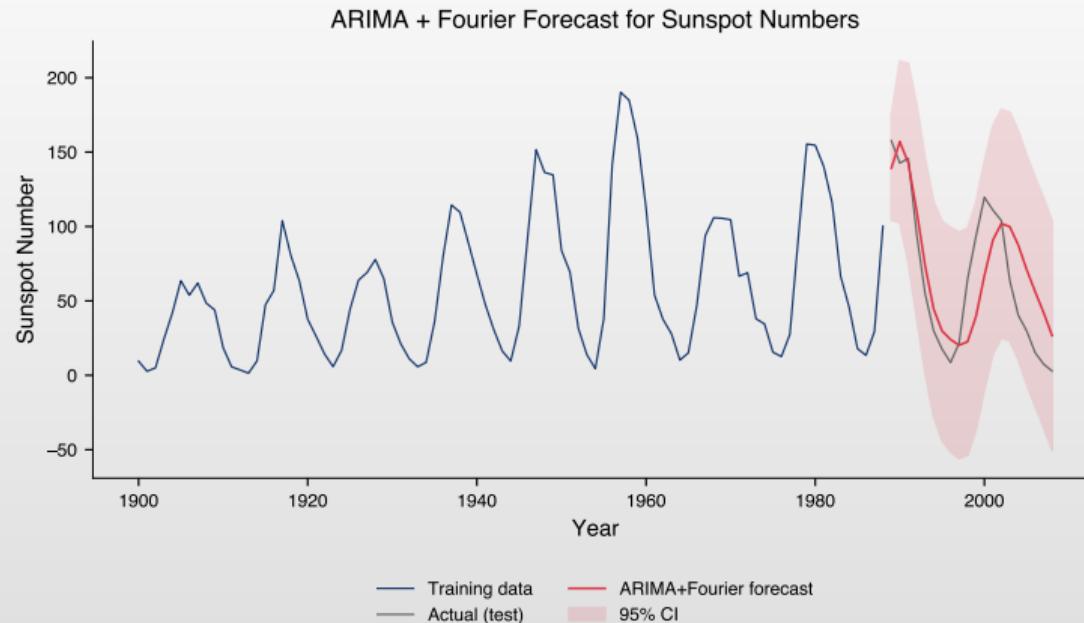
## Termeni Fourier pentru sezonalitate



## Termeni Fourier pentru sezonalitate



## Pete solare: rezultate prognoză



## Pete solare: selectarea modelului

### Metodologie

- **Comparatie:**  $K = 1, 2, 3, 4$  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
<b>Total</b>		<b>109</b>

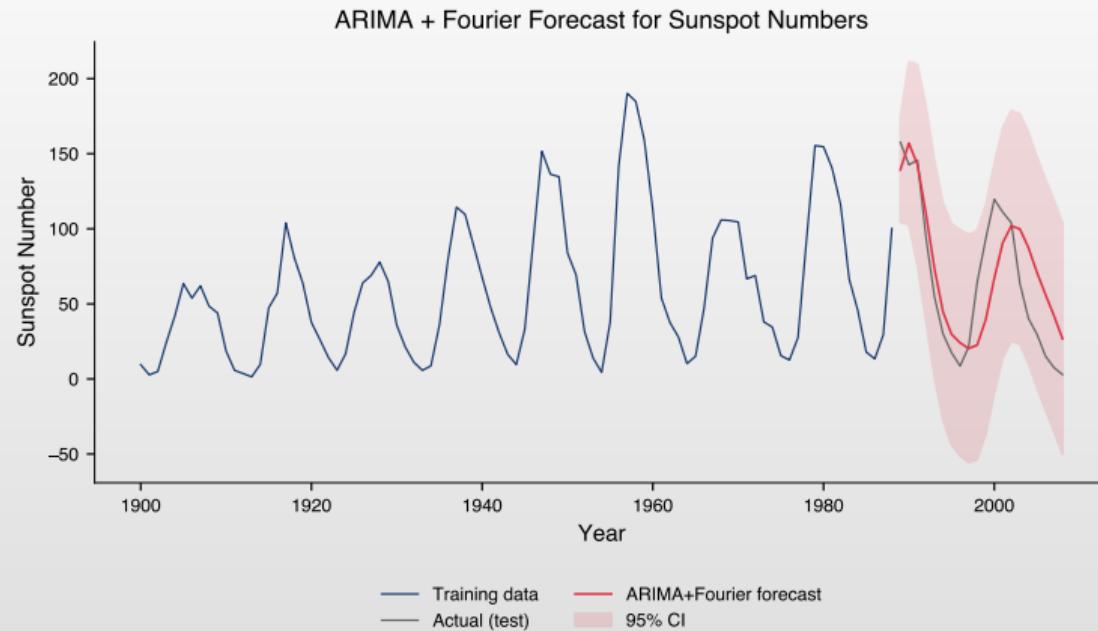
Comparatie Modele		
K	AIC	Val RMSE
1	665,9	87,15
2	668,0	86,92
3	671,8	<b>86,81</b>
4	674,5	87,93

### Rezultat

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



## Pete solare: rezultate prognoză



## Pete solare: concluzii cheie

### Când să folosiți termeni Fourier

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

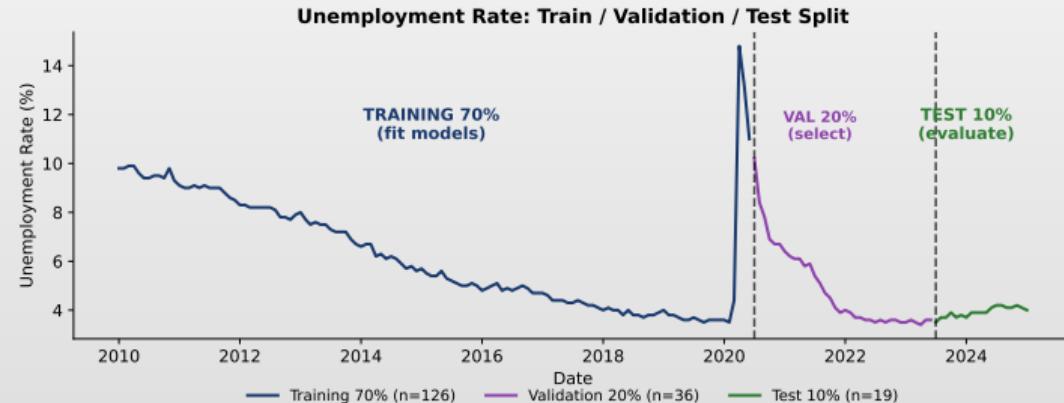
- Domenii:** Cicluri climatice, cicluri de afaceri, fenomene astronomice



## Şomajul: Train / Validation / Test Split

### Metodologie

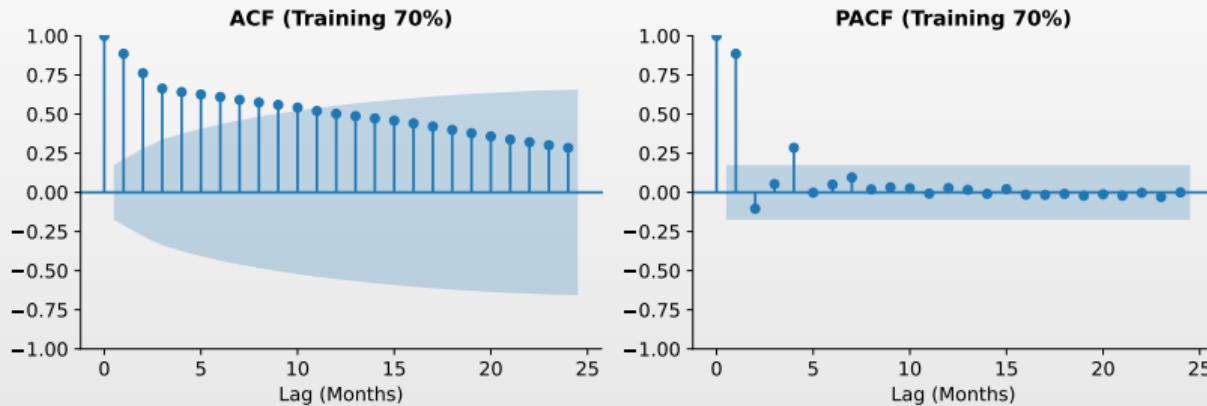
- Training** (70%): Estimare modele
- Validare** (20%): Selectie model
- Test** (10%): Evaluare finală



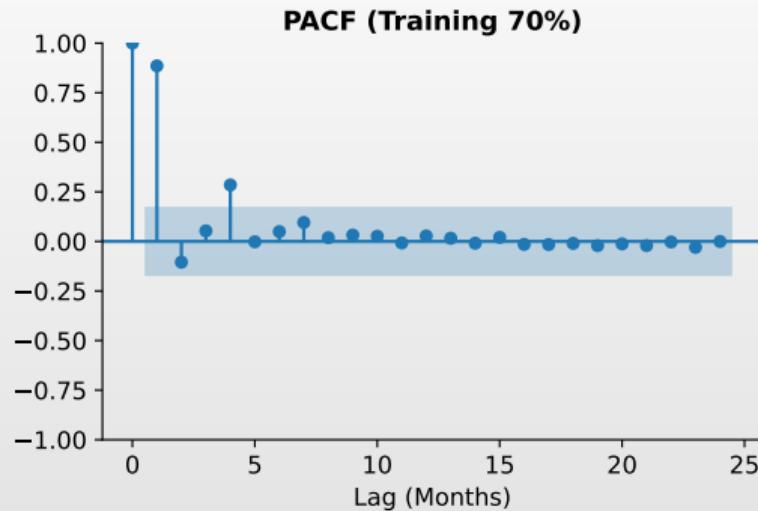
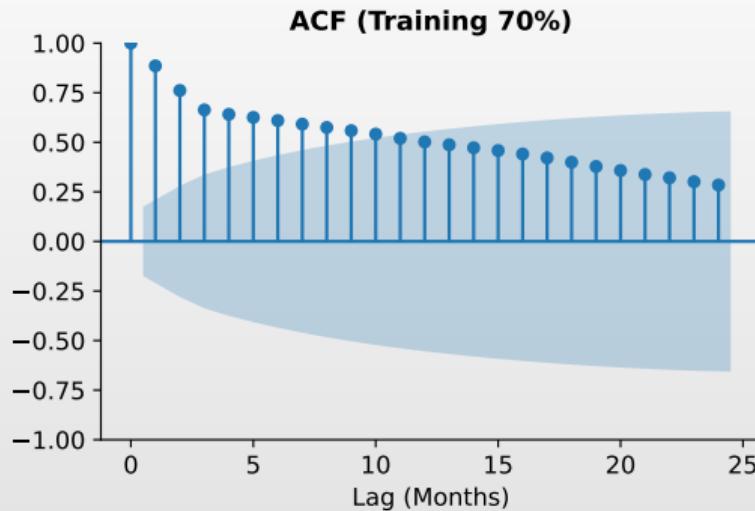
Q TSA\_ch10\_unemployment\_train\_val\_test



## Șomajul: analiză preliminară

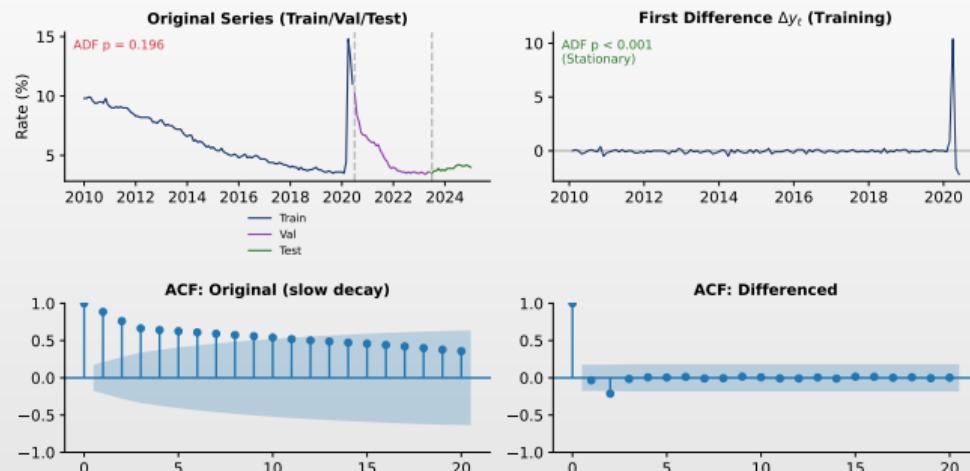


## Şomajul: analiză preliminară

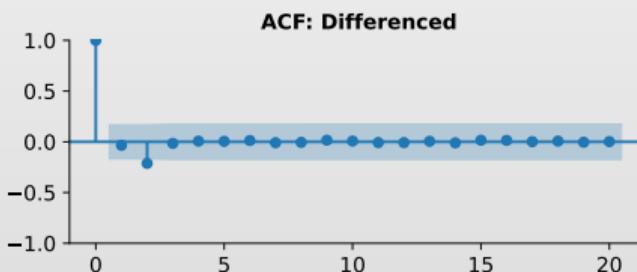
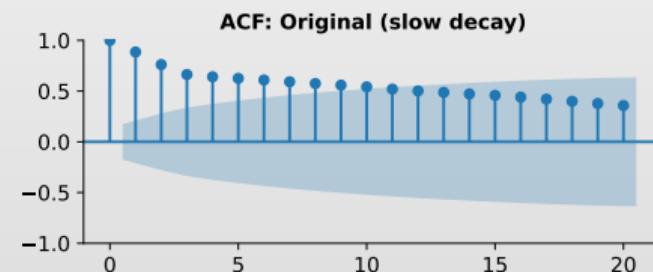
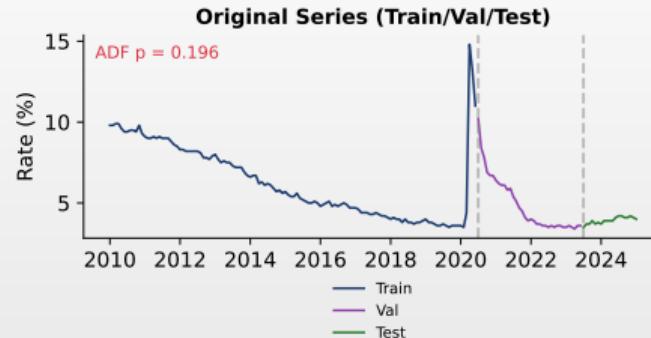


Q TSA\_ch10\_unemployment\_acf\_pacf

## Şomajul: teste de staționaritate



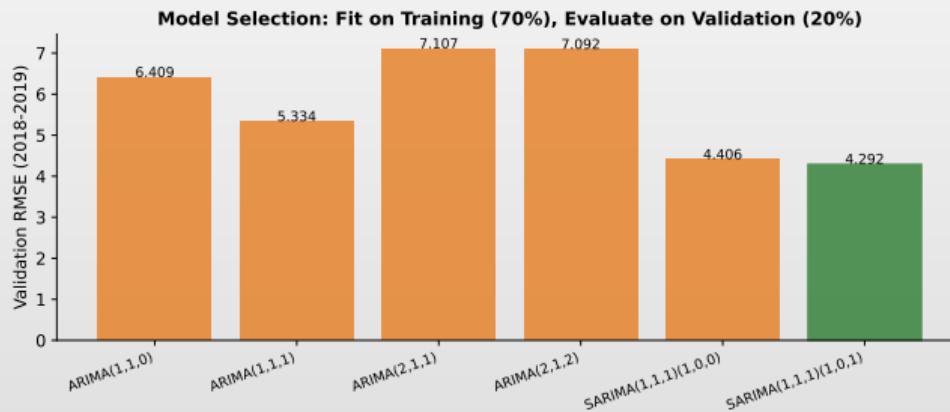
## Șomajul: teste de staționaritate



## Șomajul: selecția modelului (set validare)

Best: SARIMA(1,1,1)(1,0,0)<sub>12</sub>

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



 [TSA\\_ch10\\_sarima\\_model\\_selection](#)



## Şomajul: parametrii SARIMA

SARIMA(1,1,1)(1,0,0)<sub>12</sub> estimat pe Train+Val (2010-2019)

- 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	***



## Testul Ljung-Box pentru autocorelația reziduurilor

### Definiție 3 (Testul Ljung-Box)

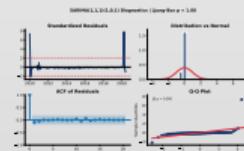
Pentru reziduurile  $\hat{\epsilon}_t$  cu autocorelații eșantion  $\hat{\rho}_k$ , statistica de test:

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

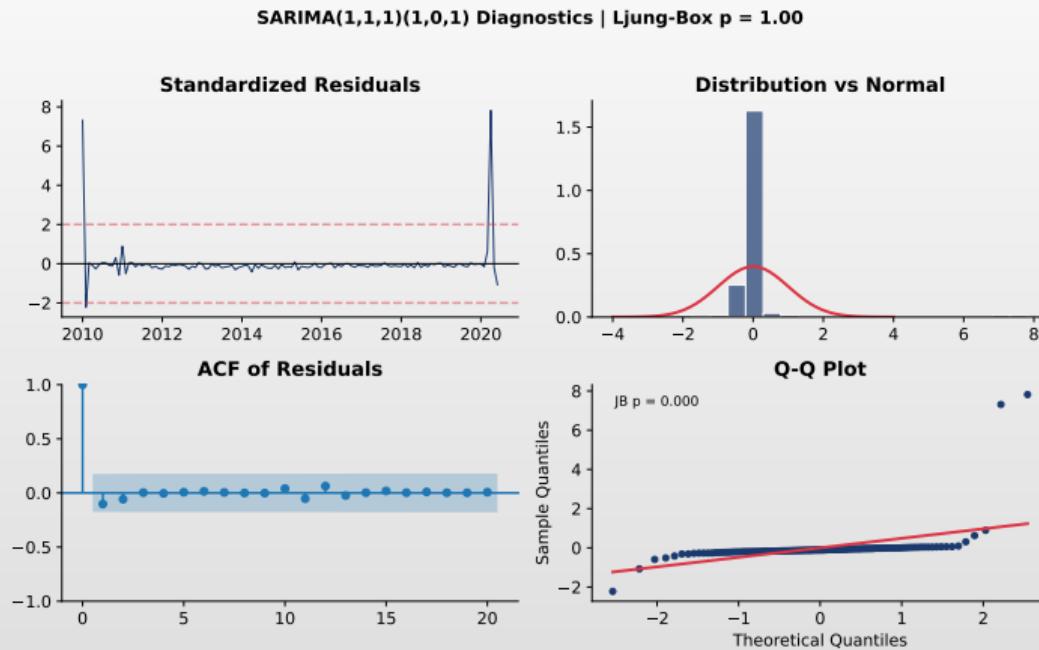
unde  $p, q$  sunt ordinele ARMA.  $H_0$ : Reziduurile sunt zgromot alb.

### 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: Folosiți  $h = \min(10, n/5)$  pentru ordinul lag-ului



## Şomajul: Diagnosticare SARIMA



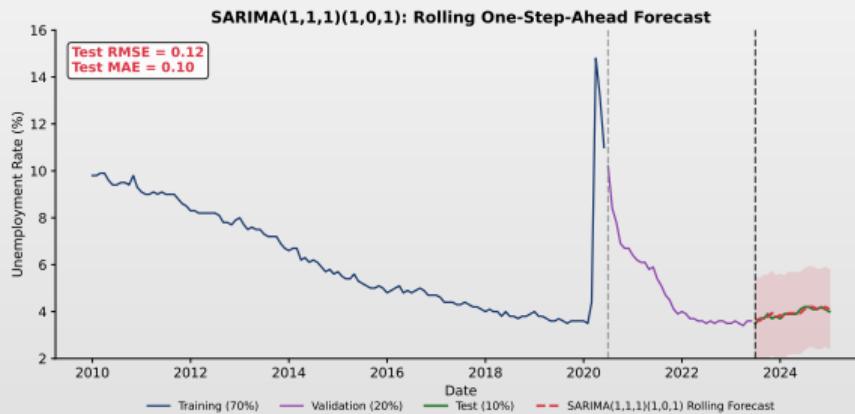
TSA\_ch10\_sarima\_diagnostics



## Şomajul: prognoza rolling SARIMA

### Problemă: Ruptura structurală

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



Q TSA\_ch10\_sarima\_forecast



## Modelul Prophet

### Definiție 4 (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 locařilor
- `changepoint_prior_scale` controlează flexibilitatea

#### Avantaje

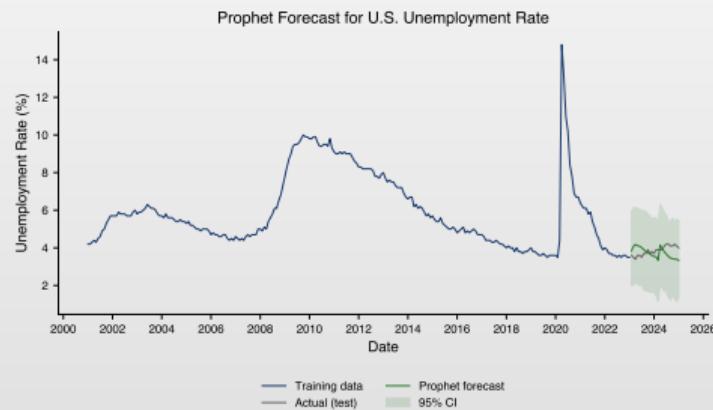
- Gestionează date lipsă
- Componente interpretabile
- Robust la outlieri



## Şomajul: rezultate prognoză Prophet

### Concluzie cheie

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



Q TSA\_ch10\_unemployment\_forecast



## Şomajul: 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
<b>Total</b>		<b>181</b>

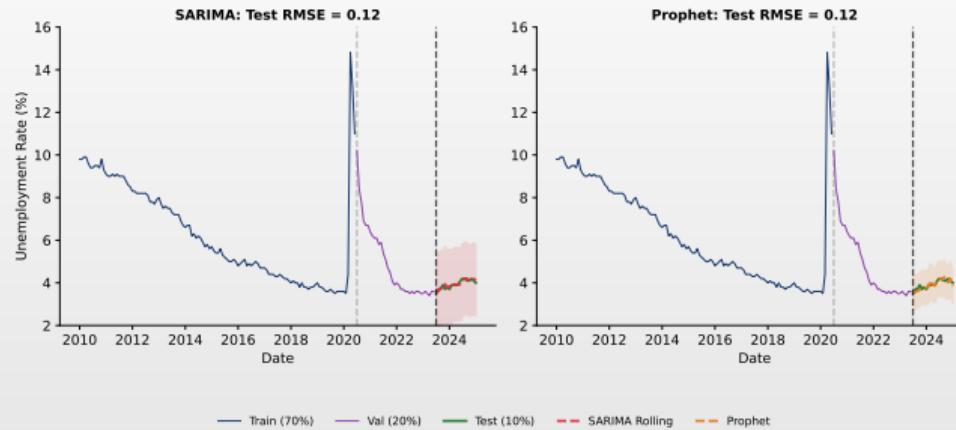
	Scale	Val RMSE	
Comparație Scale	0,01	4,21	
	0,05	3,89	
	0,10	<b>3,52</b>	Cel mai bun
	0,30	3,67	
	0,50	3,81	

### Interpretare

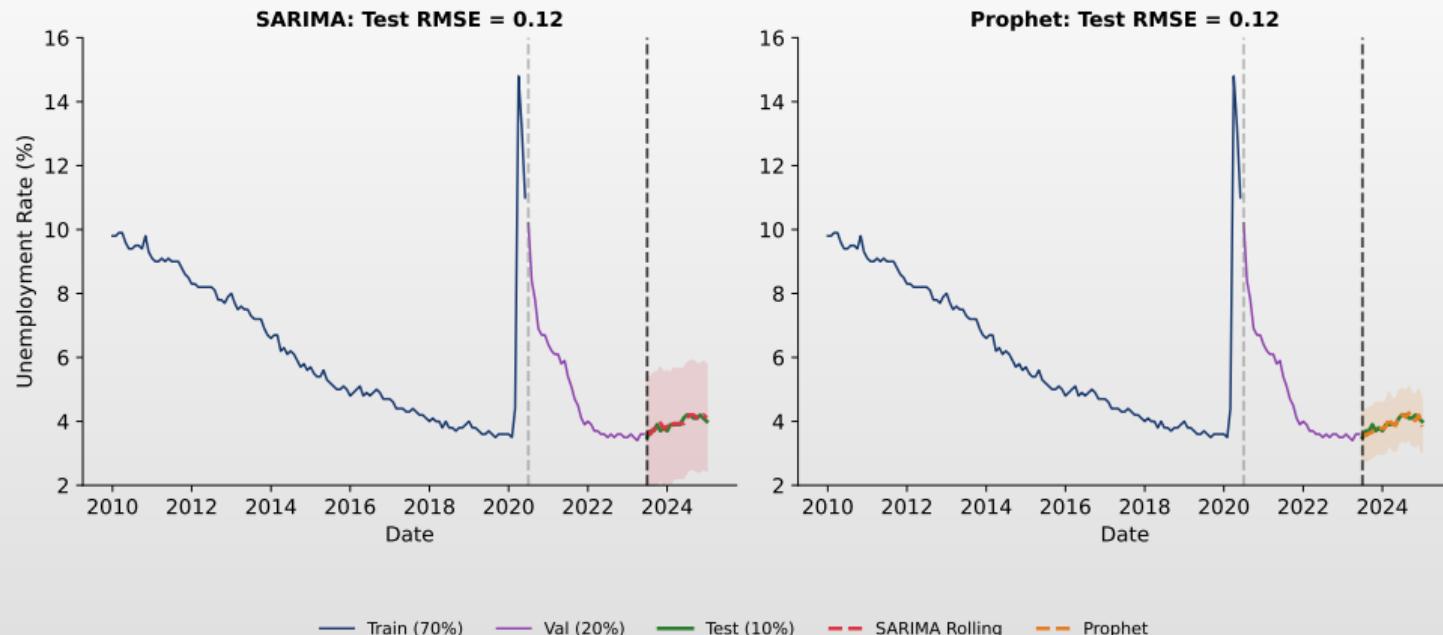
- Scale = 0,10 echilibrează flexibilitatea (captarea şocului COVID) cu stabilitatea



## Șomaj: comparație SARIMA vs Prophet



## Șomaj: comparație SARIMA vs Prophet



Q 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 (0,12 vs 0,58)
- Validați întotdeauna!

### Prophet vs ARIMA

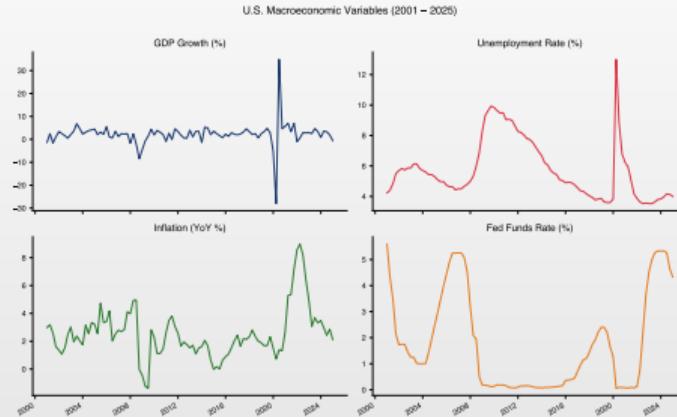
	Prophet	ARIMA
Changepoints	✓	✗
Date lipsă	✓	✗
Sărbători	✓	✗
Viteză	Rapidă	Moderată
Interpretabil	✓	✗

### Parametri cheie

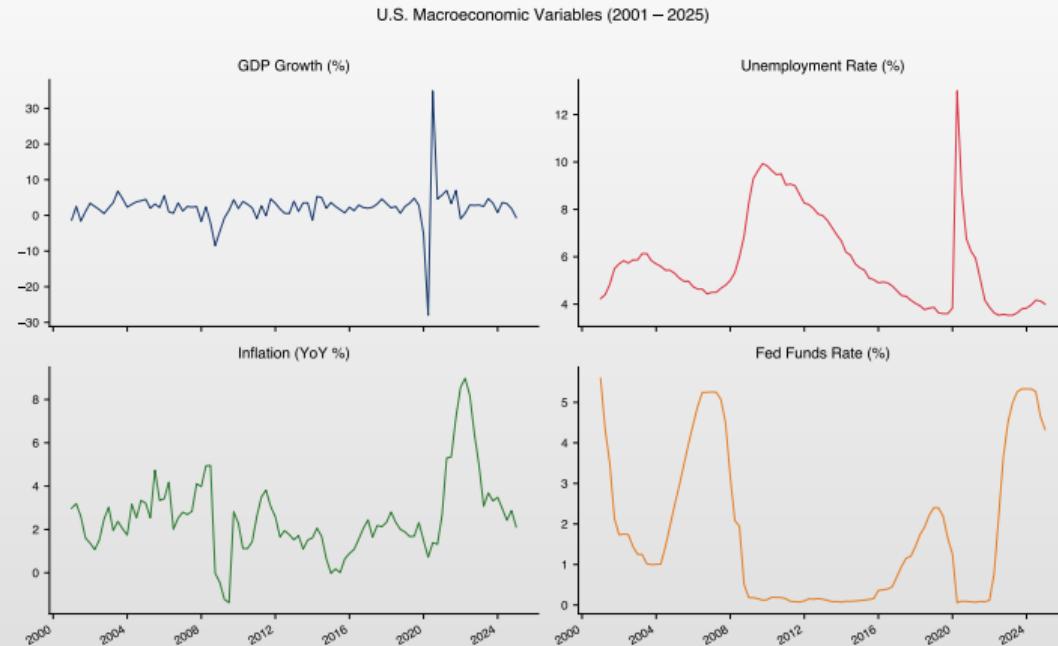
- `changepoint_prior_scale`: flexibilitate
- `seasonality_prior_scale`: netezime



## VAR: date economice multivariate



## VAR: date economice multivariate



## Specificarea modelului VAR

### Definiție 5 (Autoregresie Vectorială VAR(p))

- **Date:** Pentru  $K$  variabile  $y_t = (y_{1t}, \dots, y_{Kt})'$ :

$$y_t = c + A_1 y_{t-1} + A_2 y_{t-2} + \dots + A_p y_{t-p} + u_t$$

- **Notatie:**  $A_i$  sunt matrici de coeficienți  $K \times K$ ,  $u_t \sim N(0, \Sigma)$

### 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-validation pe date păstrate



## Criterii informaționale pentru selectarea modelului

### Definiție 6 (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  $n > 7$
- Mai parsimonios



## VAR: selectarea lag-ului și estimare

### Criterii informaționale

Lag	BIC
1	-4,810
2	<b>-5,178</b>
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-T1	10
<b>Total</b>		<b>97</b>

### Verificare validare

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



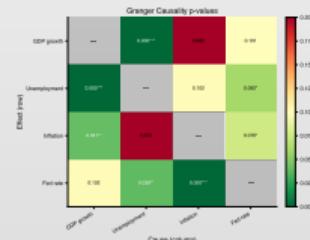
## Cauzalitatea Granger: Rezultate empirice

### Interpretare

Fiecare celulă arată p-value-ul pentru testarea dacă variabila din rând Granger-cauzează variabila din coloană. Verde:  $p < 0,10$ . Citire: rând cauzează coloană.

### Concluzii economice

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



Q TSA\_ch10\_granger\_heatmap



## Cauzalitatea Granger: Definiție formală

### Definiție 7 (Cauzalitatea Granger)

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

$$\text{MSE} \left[ \mathbb{E}[Y_{t+h} | \mathcal{F}_t^{X,Y}] \right] < \text{MSE} \left[ \mathbb{E}[Y_{t+h} | \mathcal{F}_t^Y] \right]$$

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

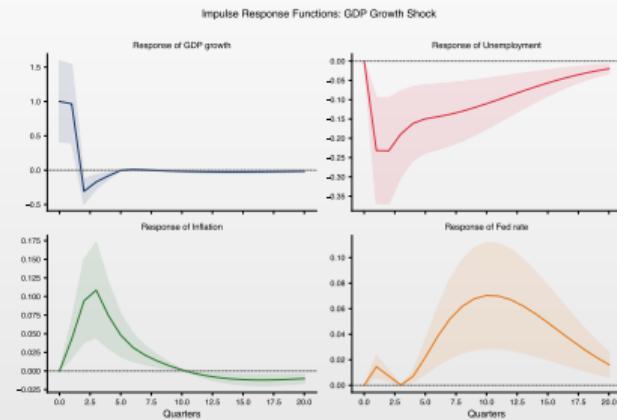
Cauzalitatea Granger este **cauzalitate predictivă**, nu cauzalitate reală. “ $X$  Granger-cauzează  $Y$ ” înseamnă că  $X$  conține informație utilă pentru prognoza lui  $Y$ , nu că  $X$  cauzează  $Y$  structural.

### Procedura de testare

Folosim testul F (sau Wald) pentru a testa  $H_0$ : coeficienții lag-urilor lui  $X$  sunt simultan zero în ecuația lui  $Y$ .



## Funcții de răspuns la impuls (IRF)



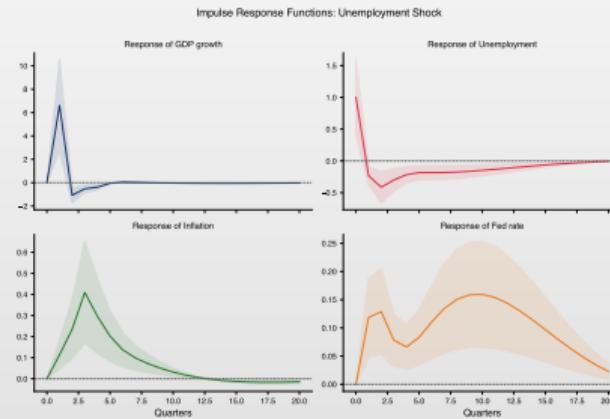
Q TSA\_ch10\_irf\_gdp\_shock



## IRF: şoc şomaj

### Efecte

- ↑ řomaj  $\succ$  ↓ PIB (Okun), ↓ Inflařie (Phillips), Fed reduce rata

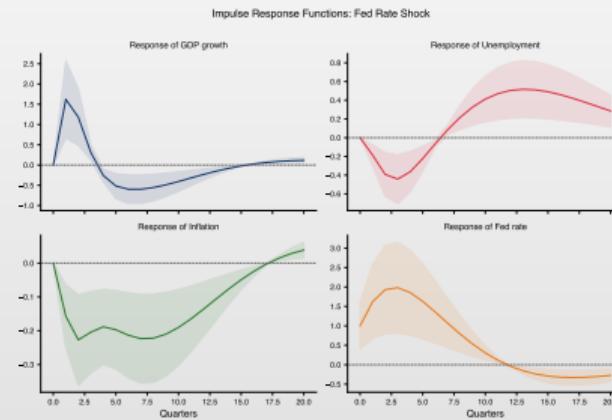


Q TSA\_ch10\_irf\_unemp\_shock

## IRF: şoc rată Fed

### Politică monetară

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



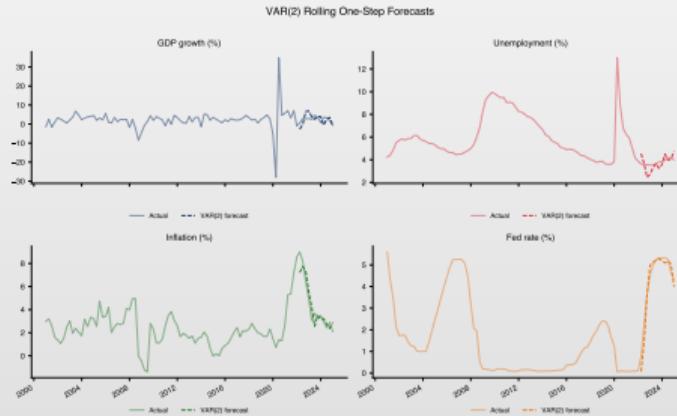
Q TSA\_ch10\_irf\_fed\_shock



## VAR: Prognoza (Train/Val/Test)

### Prognoză Rolling one-step-ahead

- VAR captează dinamică PIB-Şomaj
- Şocul COVID vizibil în perioadă validare (2020)



Q 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	<b>1,53</b>	<b>1,27</b>	<b>60%</b>

### Puncte forte

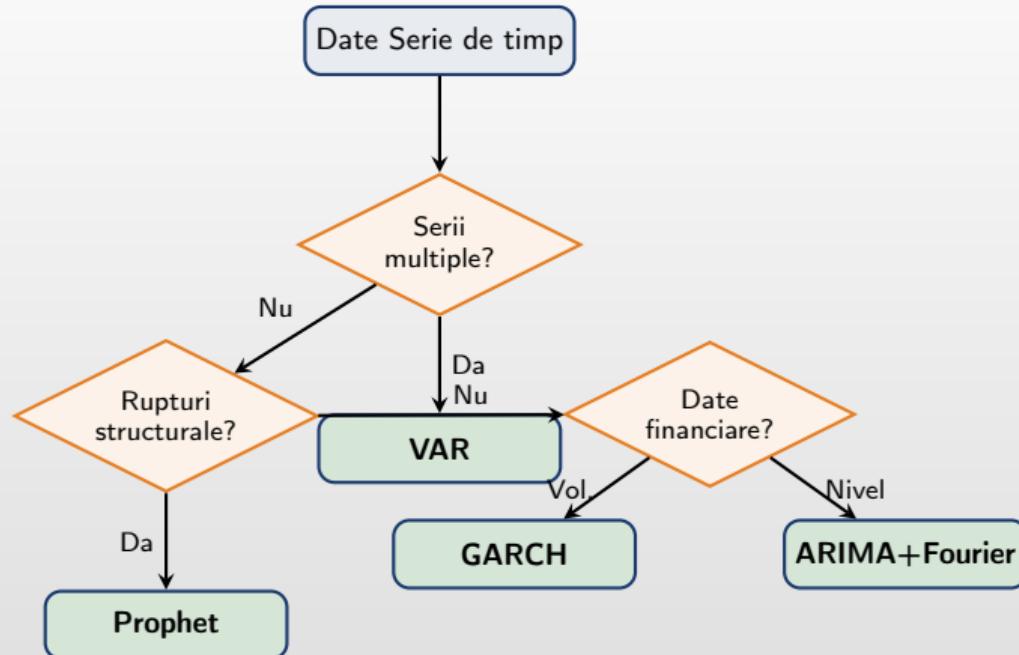
- Captează dinamică între variabile
- Acuratețe direcțională bună
- Relații interpretabile

### Limitări

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



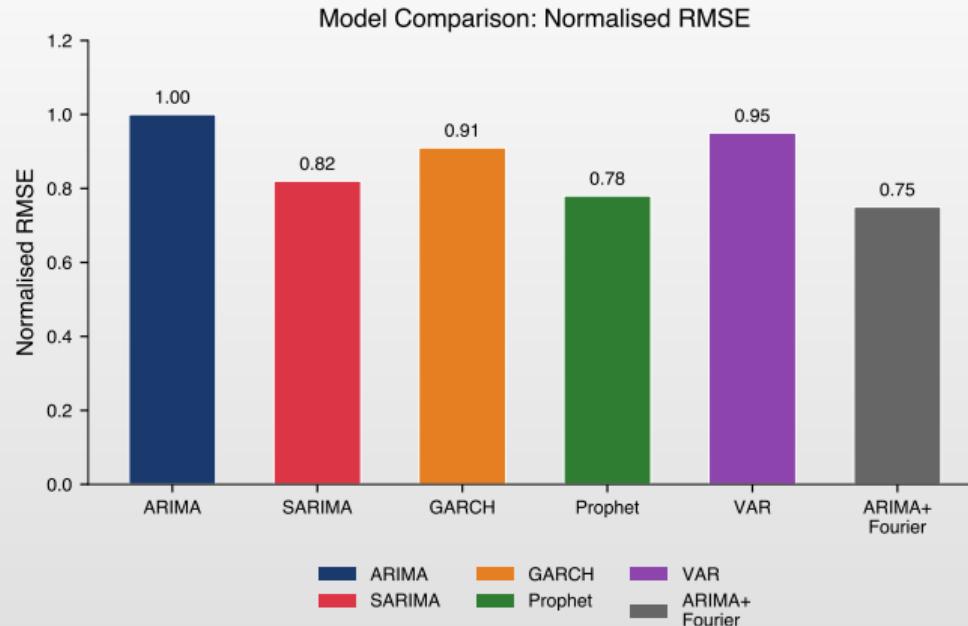
## Cadrul de selectare a modelului



## Sumar: comparație modele



## Sumar: comparație modele



## Sinteză: Comparația modelelor

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

### Rezultatele noastre

- GARCH: MAE=1,82 (volatilitate)
- Fourier: RMSE=31,10 (cycluri)
- SARIMA: RMSE=0,12 (rupturi)
- VAR: RMSE mediu=1,53 (multi)

### Insight cheie

- Fiecare model excelează în domeniul său
- Arta constă în alegerea modelului potrivit caracteristicilor datelor



## 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** metrii pe test

### Greșeli frecvente

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

### 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

### Amintiți-vă

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



## Concluzii cheie

### 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 finanțiară > GARCH
- ▶ Sezonalitate lungă > Termini Fourier
- ▶ Rupturi structurale > Prophet
- ▶ Serii multiple > VAR

### 3. Interpretați Rezultatele cu Grijă

- ▶ Cauzalitate Granger ≠ cauzalitate adeverată
- ▶ Performanța out-of-sample contează cel mai mult
- ▶ Modelele mai simple funcționează adesea mai bine



## Exercițiu AI: Gândire critică

Prompt de testat în ChatGPT / Claude / Copilot

"Am un set nou de date cu vânzări lunare. Fă o analiză completă a seriei de timp: descompunere, teste de staționaritate, selecție de model, prognoză și evaluare. Vreau calitate de publicare."

**Exercițiu:**

1. Rulați prompt-ul într-un LLM la alegere și analizați critic răspunsul.
2. Urmează fluxul corect? (grafic → descompunere → test → model → diagnostic → prognoză)
3. Compară mai multe modele (ETS, ARIMA, SARIMA) cu benchmark-uri adecvate?
4. Împărțirea train/test este făcută corect? Există surgeri 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

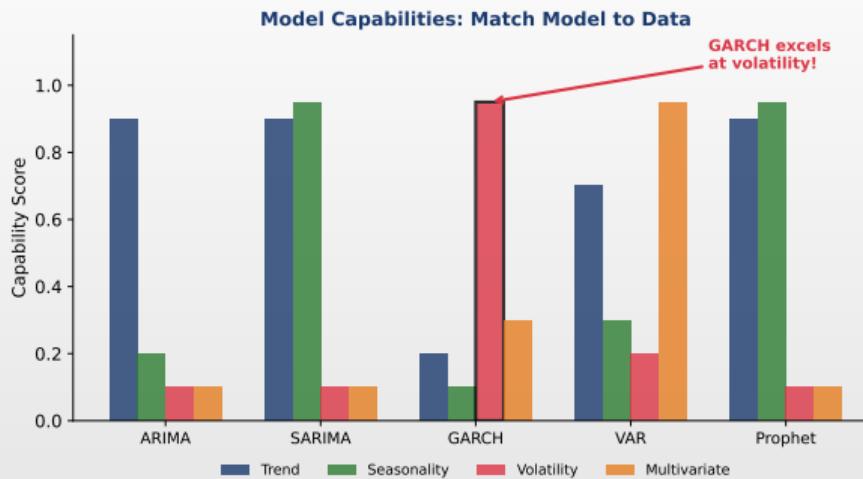
- Ce model alegeti pentru a prognoza volatilitatea randamentelor financiare?

### 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 modeleză nivelul, Prophet sezonalitatea, VAR relațiile între serii — niciunul nu modeleză varianța direct.



## Întrebarea 2

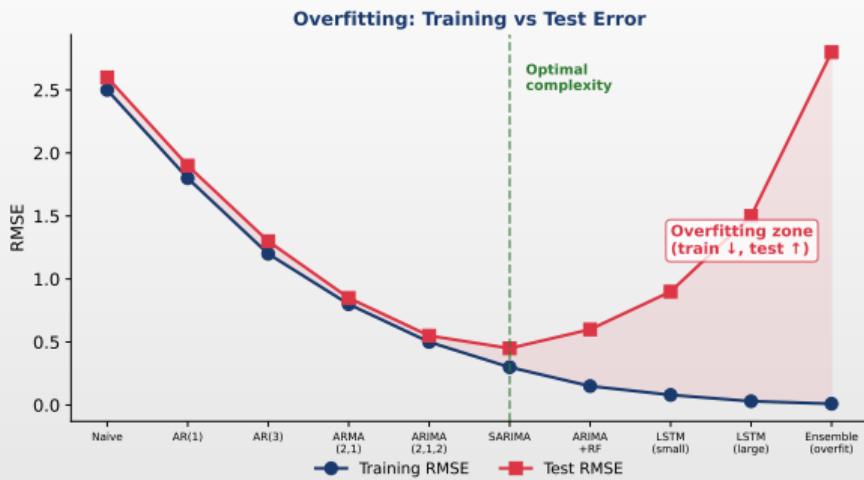
### Întrebare

- 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)

- Un raport de  $46\times$  între RMSE test și train semnalează overfitting sever. Modelul se potrivește zgomotului din antrenament și nu generalizează. Soluție: model mai simplu, validare.



## Întrebarea 3

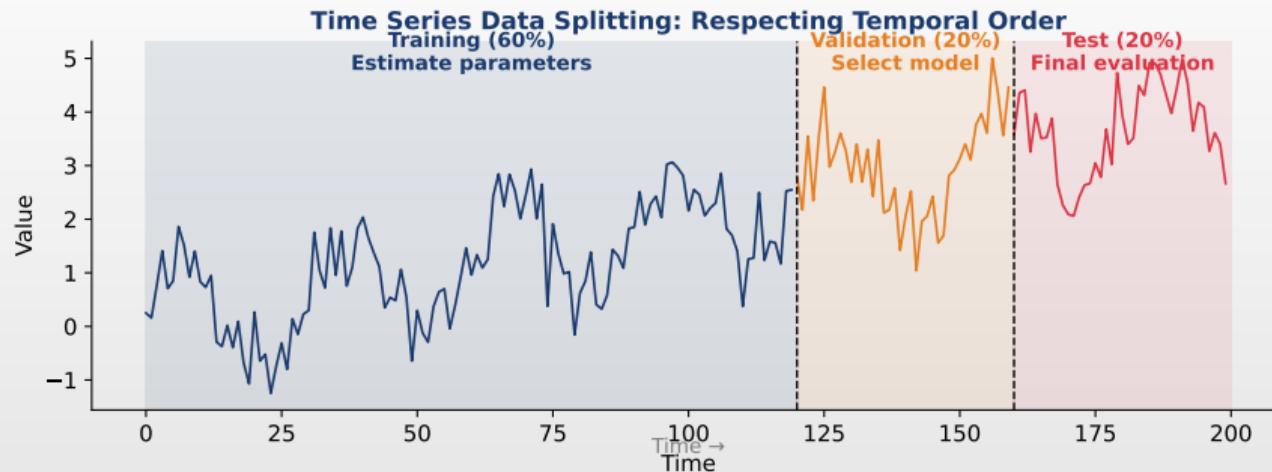
### Întrebare

- 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.

Q TSA\_ch10\_quiz3\_train\_val\_test



## Întrebarea 4

### Întrebare

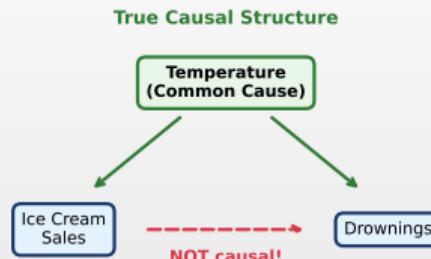
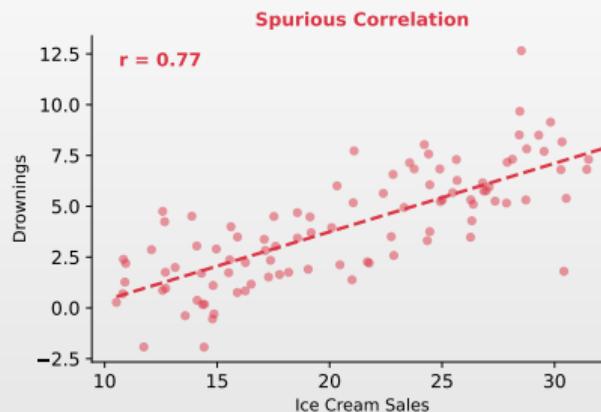
- Cauzalitatea Granger este echivalentă cu cauzalitatea reală (structurală)?

### 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

Granger Causality ≠ True Causality



Răspuns: (B)

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



## Întrebarea 5

### Întrebare

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

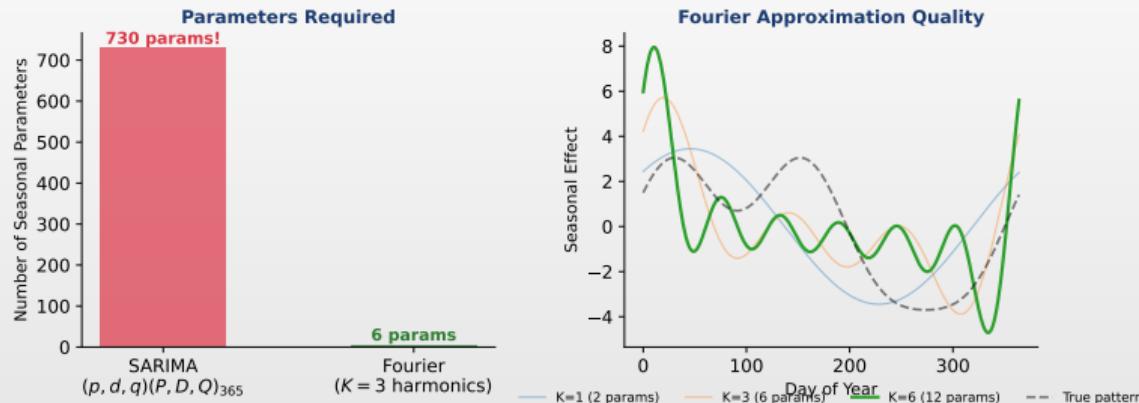
### Variante de răspuns

- (A) SARIMA( $p, d, q$ )( $P, D, Q$ )<sub>365</sub>
- (B) GARCH — modelează variația
- (C) ARIMA + Termeni Fourier sau Prophet/TBATS
- (D) VAR cu 365 lag-uri



## Întrebarea 5: Răspuns

Long Seasonality ( $s = 365$ ): Fourier Terms vs SARIMA



Răspuns: (C)

- SARIMA<sub>365</sub> necesită ~730 parametri sezonieri — imposibil. Termenii Fourier cu  $K = 3$  folosesc doar 6 parametri. Prophet și TBATS gestionează sezonalități multiple automat.



## Surse de date

### Date reale folosite în acest capitol

- Bitcoin:** Yahoo Finance (BTC-USD), 2019–2025
- Pete Solare:** Dataset Wolfer din Statsmodels, 1900–2008
- Șomaj SUA:** Federal Reserve FRED (UNRATE), 2010–2025
- Variabile Economice:** FRED (GDPC1, UNRATE, CPIAUCSL, FEDFUNDS), 2000–2025

### Reproductibilitate

Toate analizele pot fi reprodate folosind notebook-ul Jupyter însotitor:

`chapter10_lecture_notebook.ipynb`



## 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> — Depozit de cod pentru statistică
- Quantinar: <https://quantinar.com> — Platformă de învățare metode cantitative
- GitHub TSA: [https://github.com/QuantLet/TSA/tree/main/TSA\\_ch10](https://github.com/QuantLet/TSA/tree/main/TSA_ch10) — Cod Python pentru acest capitol



## Concluzii esențiale

### Ce am învățat

- Selectarea modelului depinde de caracteristicile datelor: staționaritate, sezonalitate, volatilitate
- Metodologia Box-Jenkins oferă un cadru sistematic pentru modelarea seriilor de timp
- Evaluarea corectă necesită testare out-of-sample și validare încrucișată

### Important

Niciun model nu câștigă peste tot. Potriuți modelul cu datele: ARIMA pentru tendințe, SARIMA pentru sezonalitate, GARCH pentru volatilitate, VAR/VECM pentru dinamici multivariate, Prophet/TBATS pentru tipare complexe. Validați întotdeauna out-of-sample!



## Referințe

-  Box, G.E.P., Jenkins, G.M., Reinsel, G.C., & Ljung, G.M. (2015). *Time Series Analysis: Forecasting and Control*. 5th ed., Wiley.
-  Hamilton, J.D. (1994). *Time Series Analysis*. Princeton University Press.
-  Tsay, R.S. (2010). *Analysis of Financial Time Series*. 3rd ed., Wiley.
-  Hyndman, R.J., & Athanasopoulos, G. (2021). *Forecasting: Principles and Practice*. 3rd ed., OTexts.
-  Taylor, S.J., & Letham, B. (2018). Forecasting at Scale. *The American Statistician*, 72(1), 37-45.
-  Bollerslev, T. (1986). Generalized Autoregressive Conditional Heteroskedasticity. *Journal of Econometrics*, 31(3), 307-327.
-  Sims, C.A. (1980). Macroeconomics and Reality. *Econometrica*, 48(1), 1-48.

## Surse de date (sumar)

### Date reale folosite în acest capitol

- Bitcoin:** Yahoo Finance (BTC-USD), 2019–2025
- Pete Solare:** Dataset Wolfer din Statsmodels, 1900–2008
- Şomaj SUA:** Federal Reserve FRED (UNRATE), 2010–2025
- Variabile Economice:** FRED (GDPC1, UNRATE, CPIAUCSL, FEDFUNDS), 2000–2025

### Reproductibilitate

Toate analizele pot fi reproduse folosind notebook-ul Jupyter însoțitor:  
`chapter10_lecture_notebook.ipynb`



## Resurse online și cod

- Quantlet: <https://quantlet.com> → Depozit de cod pentru statistică
- Quantinar: <https://quantinar.com> → Platformă de învățare metode cantitative
- GitHub TSA\_ch10: [https://github.com/QuantLet/TSA/tree/main/TSA\\_ch10](https://github.com/QuantLet/TSA/tree/main/TSA_ch10)



# Vă Mulțumim!

## Întrebări?

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

