Aproximační techniky pro získání posteriorního rozdělení v modelech úrokových sazeb

Ukázka dosud vypracované části diplomové práce

Bc. Artem Vitkov
Vedoucí práce: Ing. Miroslav Plašil, Ph.D.

17.01.2026

1 Úvod

Tento odborný text představuje část diplomové práce zaměřené na porovnání metod posteriorní inference v bayesovských modelech úrokových sazeb. Motivací je praktická potřeba získat pro krátkodobé až střednědobé predikce nejen bodový odhad, ale především dobře kalibrovanou prediktivní hustotu, která umožňuje kvantifikaci nejistoty a pomáhá při rozhodování v prostředí markoekonomického rizika. Současně je cílem práce systematicky popsat a empiricky vyhodnotit kompromis mezi výpočetní náročností a přesností inference, zejména při nasazení výpočetně rychlejších aproximačních metod.

Empirická aplikace diplomové práce bude založena na denní časové řadě tříměsíční sazby PRIBOR. V tomto textu se soustředíme na stručnou formalizaci zvoleného modelového rámce, přehled vybraných algoritmů pro získání posteriorního rozdělení a zejména na hodnocení prediktivní hustoty prostřednictvím standardních metrik kalibrace a kvality. Pro účely demonstrace pracujeme s jednorozměrným modelem; rozšíření o exogenní proměnné (např. kurz CZK/EUR) a komplexnější specifikace bude řešeno v navazující části diplomové práce.

Implementace je realizována v jazyce Python (Python Software Foundation 2025) s využitím knihoven PyMC (Abril-Pla et al. 2023) a ArviZ (Kumar et al. 2019); dále používáme standardní vědecký ekosystém NumPy (Harris et al. 2020), pandas (The pandas development team 2020) a Matplotlib (Hunter 2007).

2 Model a prediktivní hustota

Nechť y_t označuje hodnotu úrokové sazby v čase t a $\mathcal{D}_t$ značí dostupná pozorování do času t. Prediktivní hustotu pro horizont h pak zadáváme integrací přes posteriorní rozdělení parametrů:

$$p(y_{t+h} \mid \mathcal{D}_t) = \int p(y_{t+h} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta} \mid \mathcal{D}_t)\, \mathrm{d}\boldsymbol{\theta}, \tag{1}$$

kde $\boldsymbol{\theta}$ zahrnuje parametry i latentní stavy modelu. Rovnice (1) ukazuje, že pro kvalitní predikci je nutné aproximovat posteriorní rozdělení $p(\boldsymbol{\theta} \mid \mathcal{D}_t)$, jelikož numerické integrování složitých mnohorozměrných věrohodnostních funkcí je výpočetně náročné. Například, plnohodnotný Monte-Carlo Markov Chain algoritmus na dlouhé denní řadě s těžkými chvosty běží ≈ 420 minut, což motivuje použití aproximačních metod pro denní reporting.

Použijeme regresní model rozšířený o stochastickou volatilitou ve tvaru:

$$y_t = \boldsymbol{x}_t^\top \boldsymbol{\beta} + \varepsilon_t, \tag{2}$$

$$\varepsilon_t = \exp(h_t/2)\, z_t, \qquad z_t \sim t_\nu(0,1), \tag{3}$$

$$h_t = \mu + \phi(h_{t-1} - \mu) + \eta_t, \qquad \eta_t \sim \mathcal{N}(0, \sigma_h^2), \tag{4}$$

kde h_t je latentní log-volatilita, ν určuje zešikmenost rozdělení, velikost jeho „chvostu“ a $|\phi| < 1$ zajišťuje stacionaritu procesu volatility. Specifikace (4) odpovídá standardnímu SV modelu; volba a interpretace parametrizace navazuje na autory (Kastner a Frühwirth-Schnatter 2014).

Pro apriorní rozdělení regresních koeficientů zvolíme Gaussovskou variantu „shrinkage-prioru“:

$$\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{0}, \tau^2 I),$$

která podporuje numerickou stabilitu a je kompatibilní s algoritmem Monte-Carlo Markov Chain i aproximačními metodami pro získání aposteriorního rozdělení. Pro ν lze použít slabě informativní prior např. $\nu = 2 + \xi$, $\xi \sim \mathrm{Exp}(\lambda)$ (tuto variantu zde uvádíme pouze jako možnost).

3 Posteriorní inference a aproximace

Posteriorní rozdělení má obecně tvar:

$$p(\boldsymbol{\theta} \mid \mathcal{D}_t) \propto p(\mathcal{D}_t \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta}), \tag{5}$$

které v modelu (2)–(4) nemá analytické vyjádření. To je právě motivace proč v diplomové práci navrhujeme heuristiky pro jeho získání v závislosti na vhodném scénáři modelování.

Monte-Carlo Markov Chain (referenční přístup) Gibbsův vzorkovač je klasická MCMC metoda, která generuje posloupnost $\{\boldsymbol{\theta}^{(s)}\}_{s=1}^{S}$ střídáním vzorkování z podmíněných rozdělení. Pro parametrický vektor $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_d)$ má jedna iterace tvar

$$\theta_1^{(s)} \sim p(\theta_1 \mid \theta_2^{(s-1)}, \ldots, \theta_d^{(s-1)}, \mathbf{y}), \quad \theta_2^{(s)} \sim p(\theta_2 \mid \theta_1^{(s)}, \theta_3^{(s-1)}, \ldots, \mathbf{y}), \ \ldots,$$

a obecně

$$\theta_j^{(s)} \sim p(\theta_j \mid \theta_1^{(s)}, \ldots, \theta_{j-1}^{(s)}, \theta_{j+1}^{(s-1)}, \ldots, \theta_d^{(s-1)}, \mathbf{y}), \qquad j = 1, \ldots, d.$$

V modelech stochastické volatility (SV) je klíčové, že kromě parametrů obsahují i latentní stavový proces (log-volatilitu) $h_{1:T}$, což přirozeně vede k blokovému schématu používánému v Gibbsově vzorkovači.

Gibbsův vzorkovač v každé iteraci střídá aktualizaci latentních stavů $h_{1:T}$ (typicky jako blok, např. pomocí forward–filtering/backward–sampling nebo jiných stavových technik) a aktualizaci statických parametrů (např. ϕ a parametrů rozptylu), přičemž volba vhodné parametrizace a datové augmentace má zásadní vliv na efektivitu generování posloupnosti (Kastner a Frühwirth-Schnatter 2014).

V diplomové práci budeme Gibbsův vzorkovač využívat jako plnohodnotnou referenční metodu pro SV modely, zejména v situacích, kdy latentní stavový vektor $h_{1:T}$ významně zvyšuje dimenzionalitu problému a lokální aproximační metody mohou selhávat. Současně budeme pro srovnání používat Hamiltonian Monte Carlo (HMC) a zejména jeho adaptivní variantu NUTS (No-U-Turn Sampler) jako výchozí přístup, a to zejména na zmenšených instancích modelu, kde je výpočetně realizovatelný a umožňuje transparentní porovnání s aproximačními technikami (Neal 2011; Hoffman a Gelman 2014).

Laplaceova aproximace Laplaceova aproximace nahrazuje posteriorní rozdělení $p(\boldsymbol{\theta} \mid \mathbf{y})$ lokální Gaussovskou aproximací odvozenou z kvadratického (druhého řádu) Taylorova rozvoje log-posterioru v okolí módu $\hat{\boldsymbol{\theta}}$, tj. odhadu MAP (Gelman et al. 2013; Tierney a Kadane 1986). Označme

$$\ell(\boldsymbol{\theta}) := \log p(\boldsymbol{\theta} \mid \mathbf{y}) = \log p(\mathbf{y} \mid \boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) - \log p(\mathbf{y}).$$

Nechť $\hat{\boldsymbol{\theta}}$ maximalizuje $\ell(\boldsymbol{\theta})$. Potom platí aproximace

$$\ell(\boldsymbol{\theta}) \approx \ell(\hat{\boldsymbol{\theta}}) - \frac{1}{2}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^\top \mathbf{H} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}),$$

kde

$$\mathbf{H} := -\nabla^2 \ell(\hat{\boldsymbol{\theta}})$$

je (negativní) Hessova matice log-posterioru vyhodnocená v módu. Exponováním obou stran získáme aproximaci posterioru normálním rozdělením

$$p(\boldsymbol{\theta} \mid \mathbf{y}) \approx \mathcal{N}\Big(\hat{\boldsymbol{\theta}},\, \mathbf{H}^{-1}\Big)\,,$$

tj. kovarianční strukturu aproximace určuje inverze lokální „křivosti“ log-posterioru (Gelman et al. 2013; Tierney a Kadane 1986).

Z praktického hlediska je Laplaceova aproximace výpočetně atraktivní: místo generování dlouhého řetězce MCMC vzorků stačí nalézt MAP (modus posteriorního rozdělení) a spočítat (resp. aproximovat) Hessovu matici. Zároveň je však třeba zdůraznit, že jde o *lokální* aproximaci: při výrazné šikmosti, vícevrcholovosti nebo těžkých chvostech posterioru může Gaussovský tvar vést k podcenění nejistoty a tím k příliš úzkým intervalům a nekalibrované prediktivní hustotě (Gelman et al. 2013). V diplomové práci proto budeme hodnotit dopad Laplaceovy aproximace jak na inferenci parametrů (např. šířky a pokrytí intervalů), tak především na kalibraci a kvalitu prediktivní hustoty (např. pomocí PIT a CRPS) v simulačních i reálných úlohách.

Variační inference Variační inference (angl. Variational Inference) nahrazuje přesné posteriorní rozdělení $p(\boldsymbol{\theta} \mid \mathbf{y})$ aproximací ze zvolené rodiny rozdělení $\mathcal{Q}$ a hledá

$$q^{\star}(\boldsymbol{\theta}) = \arg\min_{q(\boldsymbol{\theta}) \in \mathcal{Q}} \mathrm{KL}(q(\boldsymbol{\theta}) \,\|\, p(\boldsymbol{\theta} \mid \mathbf{y}))\,,$$

kde Kullbackova–Leiblerova divergence je definována jako

$$\mathrm{KL}(q\|p) = \int q(\boldsymbol{\theta}) \log \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta} \mid \mathbf{y})}\, \mathrm{d}\boldsymbol{\theta} \geq 0.$$

Minimalizace $\mathrm{KL}(q\|p)$ je ekvivalentní maximalizaci dolní meze log-marginální věrohodnosti (Evidence Lower Bound, ELBO). Konkrétně platí identita

$$\log p(\mathbf{y}) = \mathcal{L}(q) + \mathrm{KL}(q(\boldsymbol{\theta}) \,\|\, p(\boldsymbol{\theta} \mid \mathbf{y}))\,,$$

kde

$$\mathcal{L}(q) := \mathbb{E}_{q(\boldsymbol{\theta})}\big[\log p(\mathbf{y} \mid \boldsymbol{\theta})\big] - \mathrm{KL}(q(\boldsymbol{\theta}) \,\|\, p(\boldsymbol{\theta})) \leq \log p(\mathbf{y})$$

a $p(\boldsymbol{\theta})$ je apriorní rozdělení (Blei, Kucukelbir a McAuliffe 2017; Zhang et al. 2019). Právě díky tomu, že $\log p(\mathbf{y})$ je vůči q konstantní, maximalizace $\mathcal{L}(q)$ minimalizuje $\mathrm{KL}(q(\boldsymbol{\theta}) \,\|\, p(\boldsymbol{\theta} \mid \mathbf{y}))$.

Volba směru $\mathrm{KL}(q\|p)$ (tzv. *exclusive* KL) má praktické důsledky: pokud je posteriorní rozdělení vícevrcholové nebo má výrazné těžké chvosty, optimalizace často preferuje aproximaci soustředěnou na dominantním módu a

může podcenit pravděpodobnost v okrajích posteriorního rozdělení (Bishop 2006; Turner a Sahani 2011). V prediktivním kontextu se tento efekt typicky projeví jako příliš úzká prediktivní hustota, a tímto rizikujeme podhodnotit nejistotu v predikcích.

V diplomové práci budeme algoritmus variační inference používat jako rychlou aproximační alternativu k plnohodnotným MCMC metodám a budeme empiricky hodnotit dopad této aproximace na kalibraci prediktivní hustoty (např. pomocí PIT) a na její kvalitu (např. pomocí CRPS). Knihovna ArviZ poskytuje standardizované nástroje pro diagnostiku a porovnání výsledků (Kumar et al. 2019), zatímco modelování a inference budou realizovány v PyMC (Abril-Pla et al. 2023).

4 Vyhodnocení prediktivní hustoty

4.1 Hodnocení prediktivní hustoty

Kvalitu prediktivní hustoty budeme hodnotit pomocí *proper scoring rules*, zejména log-skóre a CRPS. Pro předpovídání na základě prediktivní hustoty s distribuční funkcí F a hodnotou predikce y je CRPS definováno jako

$$\mathrm{CRPS}(F, y) = \int_{-\infty}^{\infty} \left(F(z) - \mathbb{1}\{y \leq z\}\right)^2 \, \mathrm{d}z. \tag{6}$$

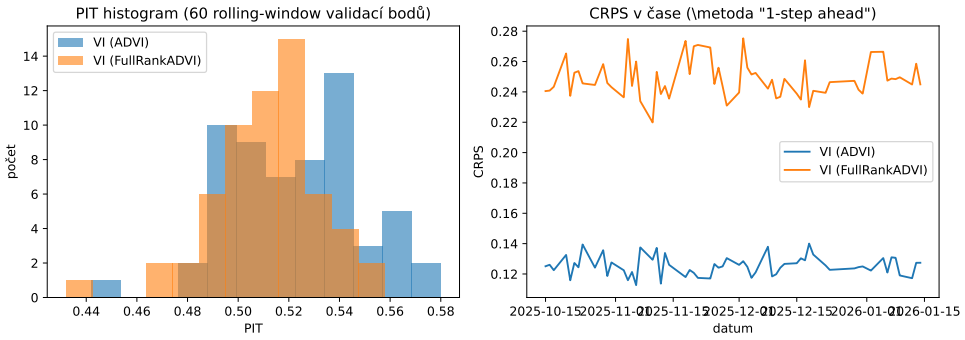
Nižší hodnota CRPS znamená lepší pokrytí predikční hustotou, neboť funkce penalizuje jak systematické posuny prediktivní hustoty, tak i její nevhodnou šířku.

Kalibraci prediktivní hustoty posuzujeme pomocí Probability Integral Transform (PIT), definovaného pro pozorování v čase t jako

$$\mathrm{PIT}_t = F_t(y_t), \tag{7}$$

kde F_t je prediktivní distribuční funkce pro čas t. Při správně kalibrované prediktivní hustotě a vhodně protokolu vyhodnocení mají hodnoty PIT_t přibližně rovnoměrné rozdělení na intervalu $(0, 1)$. Odchylky od rovnoměrného rozdělení naznačují systematické chyby v kalibraci (např. příliš úzké či naopak příliš široké prediktivní rozdělení).

Rovnice (6) a (7) používáme pro porovnání metod posteriorní inference v rolling-window validaci (60 validací na jednom kroku predikce dopředu). Ilustrační vizualizace výstupů je uvedena na Obr. 1, kde porovnáváme dvě variační aproximace posteriorního rozdělení: mean-field ADVI a FullRankADVI. Z Obr. 1 je patrné, že průběh CRPS se mezi metodami liší a zároveň PIT histogram umožňuje posoudit odchylku od ideální rovnoměrnosti. V tomto



Obrázek 1: Diagnostika prediktivní hustoty pro 60 rolling-window validací: PIT histogram (vlevo) a průběh CRPS v čase (vpravo) pro VI metody (ADVI a FullRankADVI).

minimal viable příkladu jsou hodnoty PIT výrazně koncentrovány v okolí 0.5, což indikuje neideální kalibraci prediktivní hustoty a motivuje (i) rozšíření modelu o realističtější dynamiku volatility a těžké chvosty a (ii) doplnění srovnání s plnohodnotnými MCMC metodami v navazující části diplomové práce.

Základní deskriptivní statistiky analyzované časové řady jsou uvedeny v Tab. 1. Tabulka slouží jako minimální přehled o měřítku a variabilitě dat, která následně ovlivňuje informativnost priorů i stabilitu numerické inference.

Tabulka 1: Deskriptivní statistiky PRIBOR 3M (denní data).

Proměnná	Průměr	Sm. odch.	Minimum	Maximum
PRIBOR 3M (%)	2.31	1.84	0.32	7.75

Níže uvádíme ilustrační výstup ze softwaru, který bude v empirické části generován v Pythonu (Python Software Foundation 2025) s využitím PyMC (Abril-Pla et al. 2023) a ArviZ (Kumar et al. 2019). Tab. 2 shrnuje průměrné hodnoty CRPS a PIT přes 60 rolling-window validací.

Tabulka 2: Průměrné hodnoty CRPS a PIT přes 60 rolling-window validací (1-step ahead).

metoda	průměrná hodnota CRPS	průměrná hodnota PIT
VI (ADVI)	0.1254	0.5225
VI (FullRankADVI)	0.2484	0.5108

V Tab. 2 dosahuje ADVI nižší průměrné hodnoty CRPS než FullRankADVI, což v tomto nastavení algoritmu ukazuje na kvalitnější prediktivní hustotu ve smyslu skórovací funkce. Tato část práce tak ukazuje základní rozhodovací rámec: naším cílem je porovnávat aproximační metody (zde VI) na úrovni prediktivní hustoty pomocí metrik, které současně zohledňují jak přesnost, tak nejistotu. Ve finální empirické části bude tento rámec rozšířen o plnohodnotné MCMC postupy a o modely s komplikovanější věrohodnostní funkcí, aby bylo možné kvantifikovat kompromis mezi výpočetní náročností a kvalitou prediktivní inference.

Je vhodné také poznamenat, že „FullRankADVI“ model používá flexibilnější aproximační rodinu, která umožňuje obecnou kovarianční strukturu. Optimalizace ELBO je nelineární a obecně nekonvexní problém: bohatší parametrizace může zvyšovat citlivost na počátečních parametrech, nastavení optimalizátoru a numerickou stabilitu a v konečném důsledku může konvergovat k horšímu lokálnímu optimu než jednodušší „mean-field ADVI“ model(Blei, Kucukelbir a McAuliffe 2017; Zhang et al. 2019). Z tohoto důvodu v diplomové práci nebudeme hodnotit metody pouze podle optimalizační ztráty (ELBO/loss), ale primárně na úrovni prediktivní hustoty pomocí metrik CRPS a PIT, které přímo reflektují praktickou kvalitu předpovědi.

5 Závěr

V tomto textu jsme představili minimální reprodukovatelný příklad, který ilustruje základní stavební kameny empirického porovnání metod posteriorní inference v bayesovském modelování úrokových sazeb: (i) formulaci prediktivní úlohy, (ii) generování prediktivních distribucí pro rolling-window validaci a (iii) vyhodnocení prediktivní hustoty pomocí skórování a diagnostik kalibrace. Prezentované výstupy ukazují, že i v relativně jednoduchém nastavení se mohou aproximační metody lišit ve výsledné kvalitě prediktivní hustoty, a proto je nezbytné hodnotit inference přímo na úrovni prediktivních distribucí, nikoli pouze podle optimalizačních kritérií nebo bodových chyb.

Navazující část diplomové práce rozšíří tento rámec dvěma směry. Za prvé, metodologicky doplní porovnání o plnohodnotné MCMC metody a o modelové specifikace s komplikovanější strukturou náhodné složky (např. stochastickou volatilitou a těžkými chvosty), kde lze očekávat výraznější rozdíly mezi exaktními a aproximačními postupy. Za druhé, empiricky bude provedeno systematické srovnání na reálných datech s důrazem na kompromis mezi výpočetní náročností a přesností prediktivní hustoty. Výsledkem má být prakticky interpretovatelný rozhodovací rámec pro volbu metody inference v

aplikacích souvisejících s úrokovými sazbami.

Zdroje

Abril-Pla, Oriol et al. (2023). "PyMC: Probabilistic Programming in Python". In: *PeerJ Computer Science* 9, e1516. DOI: 10.7717/peerj-cs.1516.

Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Blei, David M., Alp Kucukelbir a Jon D. McAuliffe (2017). "Variational Inference: A Review for Statisticians". In: *Journal of the American Statistical Association* 112.518, s. 859–877.

Gelman, Andrew et al. (2013). *Bayesian Data Analysis*. 3. vyd. Chapman a Hall/CRC.

Harris, Charles R. et al. (2020). "Array programming with NumPy". In: *Nature* 585, s. 357–362.

Hoffman, Matthew D. a Andrew Gelman (2014). "The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo". In: *Journal of Machine Learning Research* 15.1, s. 1593–1623.

Hunter, John D. (2007). "Matplotlib: A 2D Graphics Environment". In: *Computing in Science & Engineering* 9.3, s. 90–95.

Kastner, Gregor a Sylvia Frühwirth-Schnatter (2014). "Ancillarity-sufficiency interweaving strategy (ASIS) for boosting MCMC estimation of stochastic volatility models". In: *Computational Statistics & Data Analysis* 76, s. 408–423.

Kumar, Ravin et al. (2019). "ArviZ: a unified library for exploratory analysis of Bayesian models in Python". In: *Journal of Open Source Software* 4.33, s. 1143. DOI: 10.21105/joss.01143.

Neal, Radford M. (2011). "MCMC using Hamiltonian dynamics". In: *Handbook of Markov Chain Monte Carlo*. Ed. Steve Brooks et al. Chapman & Hall/CRC.

Python Software Foundation (2025). *Python Language Reference*. URL: https://docs.python.org/3/reference/ (cit. 17. 01. 2026).

The pandas development team (2020). *pandas-dev/pandas: Pandas*. DOI: 10.5281/zenodo.3509134.

Tierney, Luke a Joseph B. Kadane (1986). "Accurate approximations for posterior moments and marginal densities". In: *Journal of the American Statistical Association* 81.393, s. 82–86.

Turner, Richard E. a Maneesh Sahani (2011). "Two problems with variational expectation maximisation for time-series models". In: *Bayesian Time Series Models*, s. 109–130.

Zhang, Cheng et al. (2019). "Advances in Variational Inference". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.8, s. 2008–2026.

A Zdrojový kód k rolling-window experimentu

```
# %%
# Na ten  knihoven
import numpy as np
import pandas as pd
import pymc as pm
import matplotlib.pyplot as plt
from properscoring import crps_ensemble

# %%

CSV_PATH = "pribor_3m_daily.csv"

N_WINDOWS = 60
TRAIN_LEN = 500
VI_ITERS = 1200
VI_DRAWS = 500
FIX_NU = True
NU_FIXED = 8.0
SEED = 123


df = pd.read_csv(CSV_PATH, parse_dates=["date"])
df = df.sort_values("date").dropna(subset=["3M_PRIBOR"])
y = (df["3M_PRIBOR"].to_numpy(dtype=float) / 100.0)
dates = df["date"].to_numpy()

need = TRAIN_LEN + N_WINDOWS + 1
if len(y) < need:
    raise ValueError(f"Moc m lo dat: {len(y)} < {need}.
       Zkra  TRAIN_LEN/N_WINDOWS.")

y_seg = y[-need:]
dates_seg = dates[-need:]
forecast_dates = dates_seg[TRAIN_LEN + 1: TRAIN_LEN + 1
   + N_WINDOWS]


# MODEL AR(1) + Student-t

Data = getattr(pm, "MutableData", pm.Data)

```

```
with pm.Model() as model:
    y_lag = Data("y_lag", np.zeros(TRAIN_LEN))
    y_obs = Data("y_obs", np.zeros(TRAIN_LEN))

    alpha = pm.Normal("alpha", mu=0.0, sigma=0.05)
    rho_raw = pm.Normal("rho_raw", mu=0.0, sigma=1.0)
    rho = pm.math.tanh(rho_raw)          # (-1,1)

    log_sigma = pm.Normal("log_sigma", mu=np.log(0.01),
       sigma=1.0)
    sigma = pm.math.exp(log_sigma)

    nu = NU_FIXED if FIX_NU else
       (pm.Exponential("nu_minus2", lam=1/10) + 2.0)

    mu = alpha + rho * y_lag
    pm.StudentT("y_like", nu=nu, mu=mu, sigma=sigma,
       observed=y_obs)

def get_draws(obj, name: str) -> np.ndarray:
    # InferenceData
    if hasattr(obj, "posterior"):
        arr = obj.posterior[name].values
        return arr.reshape(-1)
    # MultiTrace
    return obj.get_values(name, combine=True)

def simulate_ens(alpha_s, rho_raw_s, log_sigma_s,
   lag_value, nu_value, seed):
    rng = np.random.default_rng(seed)
    rho_s = np.tanh(rho_raw_s)
    sigma_s = np.exp(log_sigma_s)
    mu_pred = alpha_s + rho_s * lag_value

    z = rng.standard_normal(size=mu_pred.shape[0])
    u = rng.chisquare(df=nu_value, size=mu_pred.shape[0])
    t = z / np.sqrt(u / nu_value)

    return mu_pred + sigma_s * t


pit_advi = np.zeros(N_WINDOWS)
crps_advi = np.zeros(N_WINDOWS)
```

```
79
80 pit_fr = np.zeros(N_WINDOWS)
81 crps_fr = np.zeros(N_WINDOWS)
82
83 for j in range(N_WINDOWS):
84     block = y_seg[j : j + TRAIN_LEN + 1]
85     lag_train = block[:-1]
86     obs_train = block[1:]
87
88     lag_pred = y_seg[j + TRAIN_LEN]
89     y_true = y_seg[j + TRAIN_LEN + 1]
90
91     pm.set_data({"y_lag": lag_train, "y_obs":
           obs_train}, model=model)
92
93     # ---- ADVI ----
94     with model:
95         approx_mf = pm.fit(
96             n=VI_ITERS,
97             method=pm.ADVI(),
98             random_seed=SEED + 1000 + j,
99             progressbar=False
100        )
101        trace_mf = approx_mf.sample
102        (VI_DRAWS,
103        random_seed=SEED + 2000 + j,
104        return_inferencedata=True)
105
106    alpha_mf = get_draws(trace_mf, "alpha")
107    rho_raw_mf = get_draws(trace_mf, "rho_raw")
108    log_sigma_mf = get_draws(trace_mf, "log_sigma")
109
110    ens_mf = simulate_ens(alpha_mf, rho_raw_mf,
           log_sigma_mf, lag_pred, NU_FIXED, seed=SEED +
           3000 + j)
111    pit_advi[j] = np.mean(ens_mf <= y_true)
112    crps_advi[j] = crps_ensemble(np.array([y_true]),
           ens_mf[None, :])[0]
113
114    # ---- FullRankADVI ----
115    with model:
116        approx_fr = pm.fit(
117            n=VI_ITERS,
```

```
            method=pm.FullRankADVI(),
            random_seed=SEED + 4000 + j,
            progressbar=False
        )
        trace_fr = approx_fr.sample(VI_DRAWS,
           random_seed=SEED + 5000 + j)

    alpha_fr = get_draws(trace_fr, "alpha")
    rho_raw_fr = get_draws(trace_fr, "rho_raw")
    log_sigma_fr = get_draws(trace_fr, "log_sigma")

    ens_fr = simulate_ens(alpha_fr, rho_raw_fr,
       log_sigma_fr, lag_pred, NU_FIXED, seed=SEED +
       6000 + j)
    pit_fr[j] = np.mean(ens_fr <= y_true)
    crps_fr[j] = crps_ensemble(np.array([y_true]),
       ens_fr[None, :])[0]

# %%

fig, ax = plt.subplots(1, 2, figsize=(11, 4))

ax[0].hist(pit_advi, bins=12, alpha=0.6, label="VI
   (ADVI)")
ax[0].hist(pit_fr, bins=12, alpha=0.6, label="VI
   (FullRankADVI)")
ax[0].set_title("PIT histogram (60 rolling-window
   validac  bod )")
ax[0].set_xlabel("PIT"); ax[0].set_ylabel("po et")
ax[0].legend()

ax[1].plot(forecast_dates, crps_advi, label="VI (ADVI)")
ax[1].plot(forecast_dates, crps_fr, label="VI
   (FullRankADVI)")
ax[1].set_title("CRPS v  ase  (\"1-step ahead metoda\")")
ax[1].set_xlabel("datum"); ax[1].set_ylabel("CRPS")
ax[1].legend()

plt.tight_layout()
plt.savefig("pit_crps_compare.pdf")
plt.close()

# %%
```

```
153
154 summary = pd.DataFrame({
155     "metoda": ["VI (ADVI)", "VI (FullRankADVI)"],
156     "p r  m  r n   hodnota CRPS": [crps_advi.mean(),
          crps_fr.mean()],
157     "p r  m  r n   hodnota PIT": [pit_advi.mean(),
          pit_fr.mean()],
158 })
159 summary.to_latex("rolling_metrics.tex", index=False,
    float_format="%.4f")
160
161 print("Saved: pit_crps_compare.pdf")
162 print("Saved: rolling_metrics.tex")
163 print(summary)
```

Listing 1: Python skript pro rolling-window validaci ADVI a FullRankADVI na datech PRIBOR 3M.