SEMESTRÁLNÍ PRÁCE

Využití Taylorova rozvoje ve výběrových šetřeních

4ST414 - Teorie výběrových šetření

Autor: František Pavlík
pavf05
Studijní program: Statistika

ZS 2025/2026

Abstrakt

Tato práce se zabývá srovnáním metod pro odhad rozptylu výběrových kvantilů, což je klíčový problém v mnoha oblastech statistiky, zejména při práci s daty, která neodpovídají normálnímu rozdělení. Hlavním cílem je porovnat přesnost a spolehlivost asymptotického přístupu založeného na Taylorově rozvoji (využívajícího jádrové odhady hustoty) s neparametrickou metodou Bootstrap. Za účelem srovnání byla provedena rozsáhlá Monte Carlo simulační studie na datech z log-normálního rozdělení s různou mírou asymetrie ($\sigma \in \{0.5, 1.0, 1.5\}$) a pro různé rozsahy výběru ($n \in \{30, 100, 250, 500, 1000, 1500, 2000\}$). Studie se zaměřila na odhad rozptylu jak pro medián, tak pro extrémní kvantily ($p = 0.99$). Výsledky ukazují, že zatímco pro symetrická rozdělení a centrální kvantily poskytuje Plug-in metoda (Taylorův rozvoj) uspokojivé výsledky, v případě silně sešikmených dat a extrémních kvantilů dramaticky selhává a podhodnocuje skutečnou variabilitu. Naproti tomu metoda Bootstrap vykazuje výrazně vyšší robustnost a přesnost pokrytí intervalů spolehlivosti, ačkoliv je výpočetně náročnější. Práce proto doporučuje použití Bootstrapu pro inferenci o extrémních kvantilech v nesymetrických rozděleních.

Obsah

1. Úvod

Odhadování kvantilů a jejich přesnosti je klíčovou úlohou v mnoha oblastech statistiky.. Zatímco bodový odhad kvantilu pomocí výběrového kvantilu je relativně přímočarý, odhad jeho rozptylu (a tím i konstrukce intervalů spolehlivosti) představuje náročnější problém, zejména pokud neznáme rozdělení, ze kterého data pocházejí, nebo pokud je toto rozdělení výrazně sešikmené.

Cílem této práce je porovnat různé metody odhadu rozptylu výběrových kvantilů. Zaměříme se na tři přístupy: teoretický asymptotický rozptyl založený na Taylorově rozvoji (který vyžaduje znalost hustoty), praktickou „plug-in“ metodu využívající jádrový odhad hustoty, a neparametrický bootstrap.

Jako modelové rozdělení pro naši simulační studii jsme zvolili log-normální rozdělení. Toto rozdělení je v praxi velmi časté (např. v příjmovém rozdělení) a vyznačuje se silnou asymetrií a těžkými chvosty, což může činit problémy asymptotickým aproximacím, zejména při malém rozsahu výběru nebo při odhadu extrémních kvantilů.

V této práci se zabýváme odhadem obecného kvantilu ze spojitého rozdělení. Pro jednoduchost se omezujeme na případ prostého náhodného výběru z nekonečné populace.

V následující kapitole nejprve teoreticky odvodíme asymptotický rozptyl výběrového kvantilu. Následně popíšeme design simulační studie, prezentujeme výsledky pro různé rozsahy výběrů a hladiny kvantilů a v závěru diskutujeme vhodnost jednotlivých metod.

2. Teoretická část

V této kapitole se zaměříme na odvození asymptotického rozptylu výběrového kvantilu. Toto odvození je bude dále použito jako základ metod pro odhad rozptylu kvantilů ve výběrových šetřeních.

2.1 Definice a značení

2.2 Definice a značení

Nechť $X_1, X_2, \ldots, X_n$ je náhodný výběr, tj. posloupnost nezávislých stejně rozdělených (i.i.d.) náhodných veličin, z rozdělení se spojitou distribuční funkcí $F(x)$ a hustotou pravděpodobnosti $f(x)$. Předpokládáme, že F je absolutně spojitá.

Definujme p-tý teoretický kvantil q_p jako hodnotu, pro kterou platí:

$$F(q_p) = p, \quad \text{kde } p \in (0,1). \tag{1}$$

Výběrový kvantil $\hat{q}_p$ je definován pomocí empirické distribuční funkce $\hat{F}_n(x)$, která je dána předpisem:

$$\hat{F}_n(x) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(X_i \leq x), \tag{2}$$

kde $\mathbb{I}(\cdot)$ je indikátorová funkce. Výběrový kvantil je pak definován jako (Hyndman & Fan, 1996):

$$\hat{q}_p = \hat{F}_n^{-1}(p) = \inf\{x : \hat{F}_n(x) \geq p\}. \tag{3}$$

Pro účely této práce budeme uvažovat Log-normální rozdělení $LN(\mu, \sigma^2)$, jehož hustota je dána:

$$f(x) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{(\ln x - \mu)^2}{2\sigma^2}}, \quad x > 0. \tag{4}$$

2.3 Odvození asymptotického rozptylu (Delta metoda)

Pro formální odvození asymptotického rozdělení výběrového kvantilu využijeme tzv. Delta metodu aplikovanou na kvantilovou funkci. Označme F distribuční funkci a

$Q(p) = F^{-1}(p)$ kvantilovou funkci. Výběrový kvantil $\hat{q}_p$ lze chápat jako odhad kvantilové funkce v bodě p, tedy $\hat{q}_p = \hat{Q}_n(p)$.

Z Centrální limitní věty víme, že pro empirickou distribuční funkci $\hat{F}_n(x)$ v pevném bodě x platí:

$$\sqrt{n}(\hat{F}_n(x) - F(x)) \xrightarrow{d} N(0, F(x)(1 - F(x))). \tag{5}$$

Abychom přešli od $\hat{F}_n$ ke $\hat{q}_p$, využijeme inverzní vztah. Aplikací funkcionální Delta metody (za předpokladu, že F je diferencovatelná v q_p a $f(q_p) > 0$) dostáváme pro kvantilový proces asymptotický vztah:

$$\sqrt{n}(\hat{q}_p - q_p) \xrightarrow{d} N\left(0, \frac{\mathrm{Var}(\mathbb{I}(X \leq q_p))}{[f(q_p)]^2}\right). \tag{6}$$

V čitateli zlomku je rozptyl indikátorové proměnné, která nabývá hodnoty 1 s pravděpodobností p a 0 s pravděpodobností $1 - p$. Její rozptyl je tedy $p(1 - p)$. Jmenovatel $[f(q_p)]^2$ plyne z derivace inverzní funkce (kvantilové funkce), neboť platí $(F^{-1})'(p) = \frac{1}{f(F^{-1}(p))} = \frac{1}{f(q_p)}$.

Výsledný asymptotický rozptyl výběrového kvantilu je tedy:

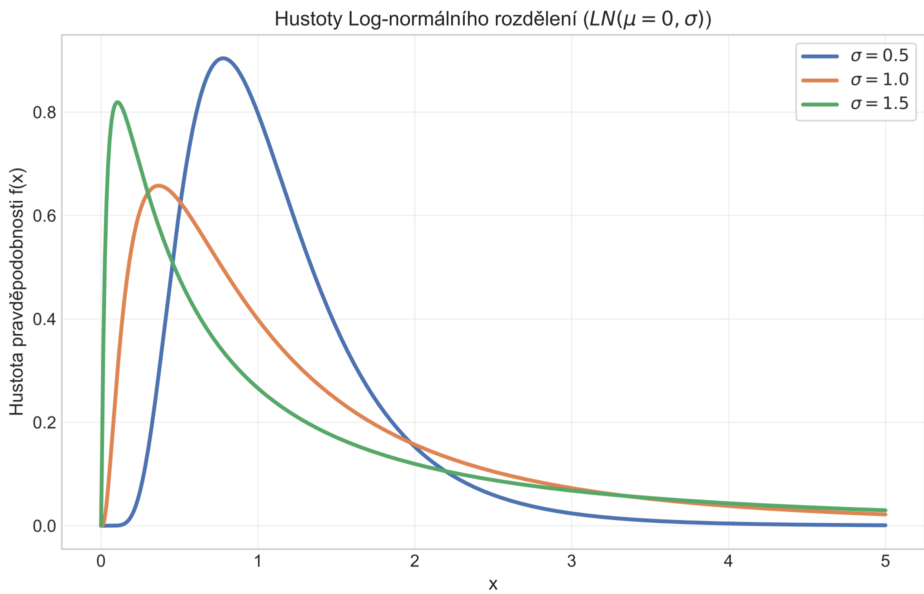
$$\mathrm{AVar}(\hat{q}_p) = \frac{p(1 - p)}{n[f(q_p)]^2}. \tag{7}$$

Tato metoda je odvozena v (Van der Vaart, 1998), ukazuje závislost přesnosti odhadu na pravděpodobnostní funkci. Je-li hustota $f(q_p)$ malá (např. v chvostech rozdělení), stává se jmenovatel velmi malým, což vede k velkým odhadům rozptylu. Tento výsledek představuje standardní tvrzení asymptotické statistiky (Van der Vaart, 1998). Ukazuje, že přesnost odhadu kvantilu závisí nepřímo úměrně hodnotě hustoty v daném bodě. V oblastech, kde je hustota nízká (chvosty rozdělení), je rozptyl odhadu kvantilu vysoký.

3. Metodika simulační studie

3.1 Generování dat

Jako podkladová data používáme log-normální rozdělení $LN(\mu, \sigma^2)$ s parametry $\mu = 0$ a $\sigma \in \{0.5, 1, 1.5\}$. Volba $\mu = 0$ je bez újmy na obecnosti (jedná se o měřítkový parametr), zatímco různé hodnoty σ nám umožňují studovat vliv šikmosti a těžkých chvostů rozdělení na přesnost odhadů. Zobrazení hustot pro uvažované parametry je uvedeno na Obrázku 3.1. Generujeme náhodné výběry o rozsahu $n \in \{30, 100, 250, 500, 1000, 1500, 2000\}$. Tato škála pokrývá situace od velmi malých výběrů, kde asymptotické vlastnosti nemusí platit, až po velké výběry, kde očekáváme konvergenci k teoretickým hodnotám.



Obrázek 3.1: Hustoty Log-normálního rozdělení pro $\mu = 0$ a různé hodnoty σ.

Pro každý výběr odhadujeme tři kvantily reprezentující různé části rozdělení, medián (50% kvantil), 95% kvantil a 99% kvantil. Počet replikací simulace byl stanoven na $M = 1000$.

3.2 Srovnávané metody

V rámci studie porovnáváme tři přístupy k odhadu směrodatné chyby (SE) kvantilu:

3.2.1 Asymptotická směrodatná chyba

Tato metoda využívá znalosti skutečného rozdělení, ze kterého data pocházejí. Do vzorce (7) dosazujeme skutečnou hustotu $f(q_p)$ log-normálního rozdělení.

$$SE_{asymp} = \sqrt{\frac{p(1-p)}{n[f_{LN}(q_p)]^2}}$$

Tato hodnota představuje teoretickou asymptotickou směrodatnou chybu, ke které by se měly odhady s rostoucím rozsahem výběru blížit. Slouží nám jako referenční hodnota pro porovnání přesnosti ostatních metod.

3.2.2 Praktická delta metoda

Tato metoda je aplikovatelná v praxi, kdy neznáme skutečnou hustotu f. Místo ní použijeme její odhad $\hat{f}(q_p)$. V naší studii využíváme jádrový odhad hustoty (Kernel Density Estimation - KDE) s Gaussovským jádrem a Scottovým pravidlem pro volbu šířky vyhlazovacího okna (bandwidth) (Scott, 2015).

$$\widehat{SE}_{Plugin} = \sqrt{\frac{p(1-p)}{n[\hat{f}_{KDE}(\hat{q}_p)]^2}}$$

Jádrový odhad hustoty je definován jako:

$$\hat{f}_{KDE}(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right), \tag{1}$$

kde K je jádro (v našem případě Gaussovské jádro $K(u) = \frac{1}{\sqrt{2\pi}}e^{-u^2/2}$) a h je šířka vyhlazovacího okna. Nevýhodou je, že chyba odhadu hustoty se přenáší do chyby odhadu rozptylu kvantilu.

3.2.3 Bootstrap

Neparametrický bootstrap je metoda založená na převzorkování. Z původního výběru vytvoříme $R = 5000$ bootstrapových výběrů (výběr s vracením), pro každý spočítáme výběrový kvantil $\hat{q}_p^*$ a rozptyl odhadujeme jako výběrový rozptyl těchto bootstrapových kvantilů.

$$\widehat{SE}_{Boot} = \sqrt{\frac{1}{R-1}\sum_{r=1}^{R}(\hat{q}_{p,r}^* - \bar{q}_p^*)^2}$$

Tato metoda nevyžaduje explicitní odhad hustoty, ale je výpočetně náročnější (Horowitz, 2019).

3.2.4 Konstrukce intervalů spolehlivosti

Pro všechny tři metody konstruujeme oboustranné intervaly spolehlivosti na hladině spolehlivosti 95% ($\alpha = 0.05$). Využíváme asymptotické normality výběrového kvantilu (Waldův typ intervalu):

$$CI_{0.95} = \left[\hat{q}_p - z_{0.975} \cdot \widehat{SE}, \quad \hat{q}_p + z_{0.975} \cdot \widehat{SE}\right], \tag{2}$$

kde $z_{0.975} \approx 1.96$ je 0.975 kvantil standardizovaného normálního rozdělení. I pro metodu Bootstrap tedy v této studii využíváme normální aproximaci s odhadnutou směrodatnou chybou.

3.3 Hodnotící kritéria

Pro kvantitativní srovnání metod používáme následující kritéria, která hodnotí přesnost a spolehlivost odhadů. Označme θ skutečnou hodnotu parametru (např. rozptyl kvantilu) a $\hat{\theta}_m$ jeho odhad v m-té replikaci Monte Carlo simulace ($m = 1, \ldots, M$).

3.3.1 Mean Squared Error (MSE)

Střední čtvercová chyba měří celkovou přesnost odhadu, v níž je zahrnut jak rozptyl odhadu, tak jeho vychýlení. Je definována jako:

$$\mathrm{MSE}(\hat{\theta}) = \frac{1}{M} \sum_{m=1}^{M} (\hat{\theta}_m - \theta)^2 \tag{3}$$

Nízká hodnota MSE indikuje, že odhad je blízko skutečné hodnotě.

3.3.2 Relative Bias (Relativní vychýlení)

Relativní vychýlení vyjadřuje, o kolik procent metoda v průměru nadhodnocuje nebo podhodnocuje skutečnou hodnotu parametru.

$$\mathrm{RB}(\hat{\theta}) = \frac{\frac{1}{M}\sum_{m=1}^{M} \hat{\theta}_m - \theta}{\theta} \cdot 100\,\% \tag{4}$$

Záporná hodnota RB značí systematické podhodnocení, což v kontextu odhadu rozptylu vede k příliš úzkým intervalům spolehlivosti. Kladná hodnota značí nadhodnocení.

3.3.3 Pokrytí intervalu spolehlivosti (Coverage Probability - CP)

Pokrytí intervalu spolehlivosti je pravděpodobnost, s jakou sestrojený interval spolehlivosti překryje skutečnou hodnotu odhadovaného parametru (kvantilu q_p).

$$\mathrm{CP} = \frac{1}{M} \sum_{m=1}^{M} \mathbb{I}(\hat{\theta}_{L,m} \leq q_p \leq \hat{\theta}_{U,m}) \tag{5}$$

kde $\mathbb{I}(\cdot)$ je indikátorová funkce a $[\hat{\theta}_{L,m}, \hat{\theta}_{U,m}]$ je interval spolehlivosti v m-té iteraci. Pro metodu s nominální hladinou spolehlivosti $1 - \alpha$ (např. 0.95), by se CP měla blížit hodnotě $1 - \alpha$. Výrazně nižší hodnota indikuje, že metoda není konzervativní (produkuje příliš mnoho chyb I. druhu).

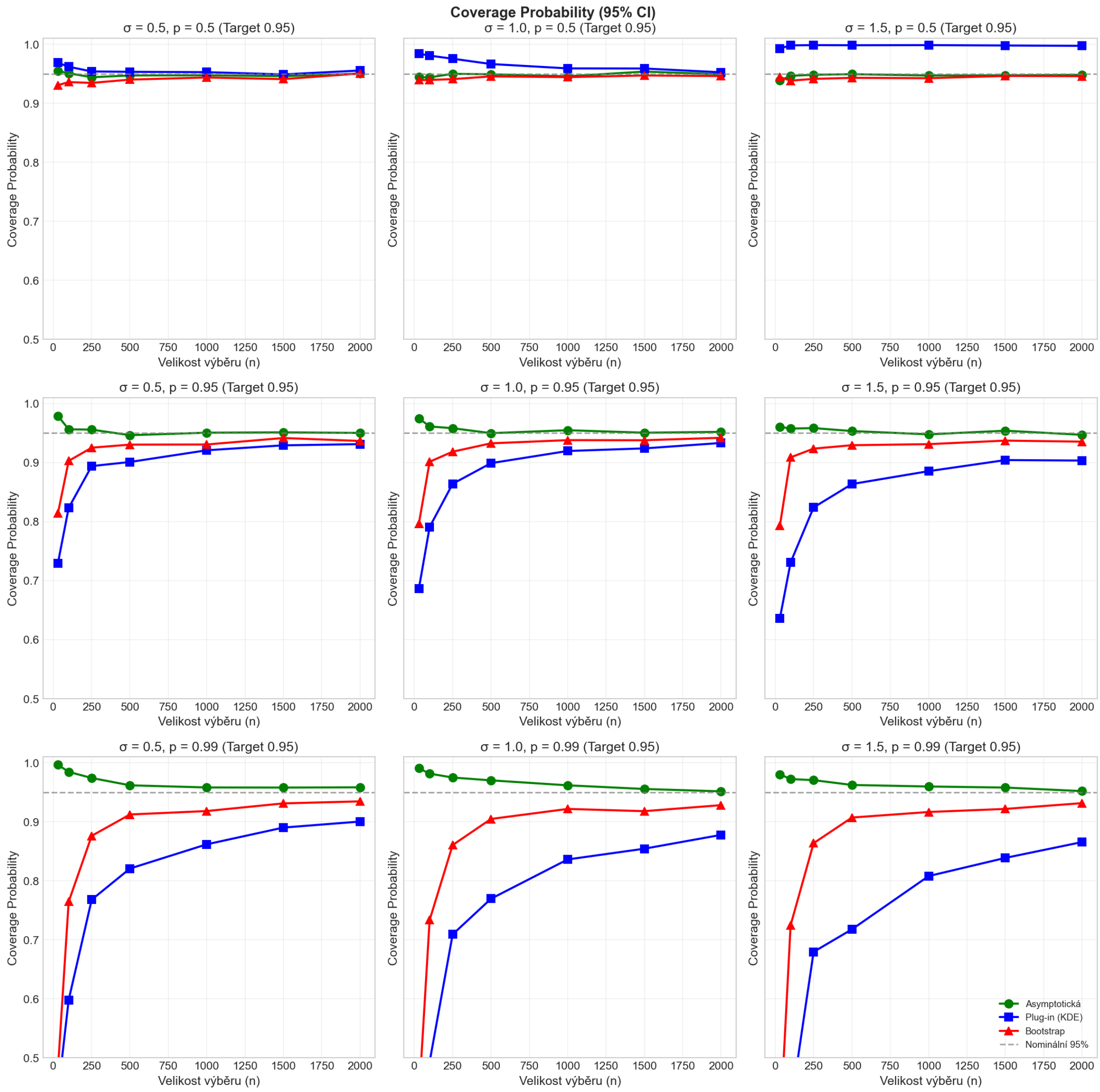
4. Simulační studie

V této části prezentujeme výsledky Monte Carlo simulace. Sledujeme chování odhadů pro tři hladiny kvantilů: $p = 0.50$, $p = 0.95$ a $p = 0.99$. Pro všechny kvantily je konstruován 95% interval spolehlivosti.

Výsledné grafy jsou uspořádány do matice 3×3:

- **Řádky:** Horní řada odpovídá kvantilu $p = 0.50$ (cílové pokrytí 0.5), prostřední řada odpovídá kvantilu $p = 0.95$ (cílové pokrytí 0.95), dolní řada kvantilu $p = 0.99$ (cílové pokrytí 0.99).
- **Sloupce:** Parametr asymetrie log-normálního rozdělení $\sigma \in \{0.5, 1.0, 1.5\}$.

4.1 Pokrytí intervalu spolehlivosti



Obrázek 4.1: Pokrytí intervalu spolehlivosti. Metody: Bootstrap (červená), Plug-in (modrá), Oracle (zelená).

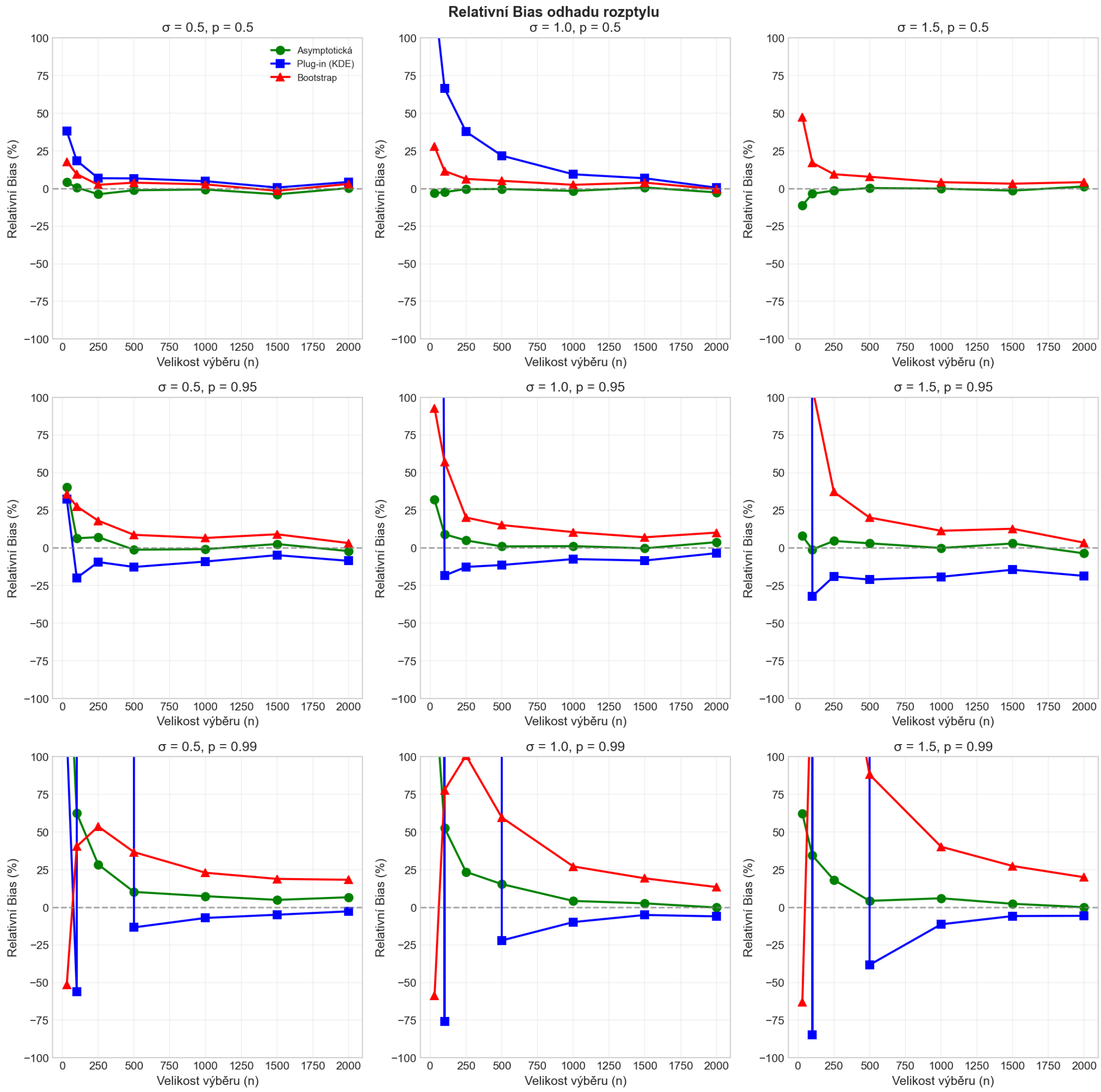
Výsledky ukazují zřetelný rozdíl v chování metod v závislosti na parametrech simulace:

1. **Teoretická delta metoda (zelená)**: Tato referenční metoda konzistentně dosahuje nominálního pokrytí ve všech scénářích. To potvrzuje, že samotná asymptotická aproximace (7) je platná, pokud známe skutečnou hodnotu hustoty $f(q_p)$.
2. **Plug-in metoda (modrá)**: V případě "běžných"dat ($\sigma \leq 1.0$) poskytuje uspokojivé výsledky srovnatelné s Oracle metodou. V kritickém scénáři ($\sigma = 1.5, p = 0.99$) však dochází k dramatickému selhání. Pravděpodobnost pokrytí

zde klesá hluboko pod cílovou hladinu 0.99 (často i pod 0.50). Graf relativního vychýlení (Obrázek 4.2) odhaluje příčinu: odhad směrodatné chyby je v tomto případě systematicky podhodnocen o více než 40 %. Jádrový odhad hustoty v řídkém chvostu "přehlazuje"data, nadhodnocuje $f(q_p)$, a tím uměle snižuje odhadovaný rozptyl.

3. **Bootstrap (červená)**: Prokazuje mnohem vyšší robustnost vůči asymetrii. I v případě $\sigma = 1.5$ si udržuje vyšší pravděpodobnost pokrytí než Plug-in metoda. U malých výběrů ($n = 30$) však ani Bootstrap nedosahuje ideálního pokrytí 0.99. Zde narážíme na limity malého vzorku, kdy v kritické oblasti chvostu prostě "nejsou data"pro spolehlivé převzorkování.

4.2 Systematické vychýlení (Relative Bias)



Obrázek 4.2: Relativní vychýlení odhadu směrodatné chyby (SE).

Systematické vychýlení odhadu směrodatné chyby úzce souvisí s výsledky pokrytí.

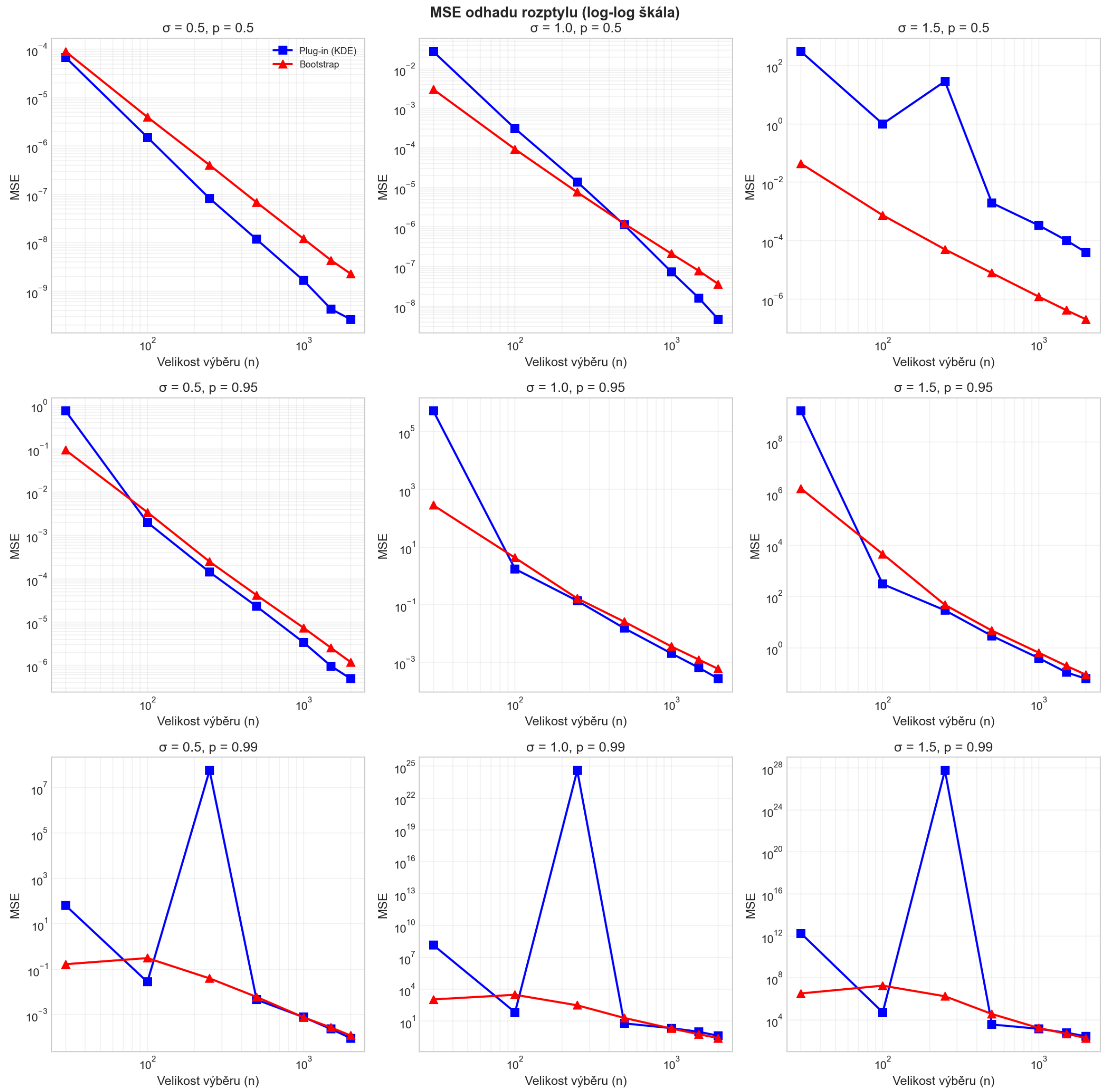
1. **Plug-in metoda (modrá)**: Vykazuje silné záporné vychýlení pro extrémní kvantily ve chvostech s vysokou variabilitou ($\sigma = 1.5$). Vychýlení dosahuje hodnot okolo -40 % až -50 %. Příčinou je vyhlazovací efekt jádrového odhadu hustoty (KDE). Ve chvostech log-normálního rozdělení, kde hustota rychle klesá, jádrový odhad s fixní šířkou okna (Scott's rule) má tendenci vyhlazovat pravděpodobnostní funkci. Tím nadhodnocuje hustotu $f(q_p)$ v bodě kvantilu. Vzhledem k tomu, že $f(q_p)$ vystupuje ve vzorci pro asymptotický rozptyl ve jmenovateli, její nadhodnocení vede k podhodnocení rozptylu (SE).

2. **Bootstrap (červená)**: Vykazuje mnohem stabilnější chování. Ačkoliv pro velmi malé vzorky ($n = 30$) také trpí mírným záporným vychýlením (cca -10 % až -20 %), toto vychýlení s rostoucím n rychle mizí. Na rozdíl od Plug-in metody, která zůstává vychýlená i pro velké vzorky (je asymptoticky nestranná, ale konvergence je pro KDE pomalejší), Bootstrap dokáže lépe zachytit variabilitu výběrového kvantilu bez explicitního odhadu hustoty.

4.3 Konvergence chyby (MSE)

Grafy v log-log měřítku (Obrázek 4.3) ukazují lineární pokles chyby, což odpovídá teoretickému předpokladu, že rozptyl odhadů klesá.

Všechny tři metody vykazují podobnou rychlost konvergence (sklon přímek je téměř totožný), avšak liší se v absolutní výši chyby. Pro kvantil $p = 0.99$ a vysokou asymetrii $\sigma = 1.5$ vidíme, že Plug-in metoda má nižší MSE než Bootstrap. To je paradoxní výsledek způsobený tím, že MSE je součtem rozptylu a vychýlení. Plug-in metoda systematicky podhodnocuje rozptyl (produkuje velmi malé odhady SE, které jsou u sebe, ale daleko od reálných hodnot). Tím má velmi malý rozptyl odhadu SE, což má větší vliv než velké vychýlení v metrice MSE. Bootstrap má sice menší vychýlení, ale vyšší variabilitu odhadů (kvůli náhodnosti převzorkování), což vede k vyšší MSE. MSE je tedy v tomto případě zkreslující metrikou a je důležité se zaměřit i na výsledky předchozích metrik.



Obrázek 4.3: MSE v log-log měřítku.

5. Závěr

V této práci jsme odvodili asymptotický rozptyl výběrového kvantilu pomocí Taylorova rozvoje a porovnali jeho přesnost s metodou Bootstrap na datech z log-normálního rozdělení.

Simulační studie ukázala, že:

1. Analytický vzorec (Taylor) funguje výborně pro centrální kvantily a dostatečně velké rozsahy výběrů ($n \geq 100$).
2. Pro extrémní kvantily ($p = 0.99$) a malé výběry ($n = 30$) je použití analytického vzorce s odhadnutou hustotou (Plug-in) rizikové a často vede k nesprávným závěrům kvůli vysoké citlivosti na chybu odhadu hustoty ve chvostech.
3. V případě malých výběrů a extrémních kvantilů nelze plně spoléhat ani na jednu z testovaných metod, ačkoliv Bootstrap vykazuje o něco lepší stabilitu.

Pro praktické aplikace doporučujeme používat asymptotický vzorec obezřetně a v případě analýzy chvostů rozdělení ověřit výsledky pomocí robustnějších metod, jako je Bootstrap, nebo využít metody odvozené specificky pro teorii extrémních hodnot.

Použitá literatura

Horowitz, J. L. (2019-08). Bootstrap Methods in Econometrics. *Annual Review of Economics*, *11*(1), 193–224. https://doi.org/10.1146/annurev-economics-080218-025651

Hyndman, R. J., & Fan, Y. (1996-11). Sample Quantiles in Statistical Packages. *The American Statistician*, *50*(4), 361–365. https://doi.org/10.1080/00031305.1996.10473566

Scott, D. W. (2015). *Multivariate Density Estimation: Theory, Practice, and Visualization* (2nd ed.). Wiley.

Van der Vaart, A. W. (1998). *Asymptotic Statistics*. Cambridge University Press.

Přílohy

Zdrojový kód simulace

```
import numpy as np
import os
import argparse
from concurrent.futures import ProcessPoolExecutor, as_completed
import pandas as pd
import matplotlib.pyplot as plt
from scipy import stats
from scipy.stats import norm, lognorm
from tqdm import tqdm
import warnings
import seaborn as sns

warnings.filterwarnings('ignore')

np.random.seed(42)

try:
    plt.style.use('seaborn-v0_8-whitegrid')
except:
    sns.set_style("whitegrid")

plt.rcParams['figure.figsize'] = (15, 10)
plt.rcParams['font.size'] = 11
plt.rcParams['axes.titlesize'] = 13
plt.rcParams['axes.labelsize'] = 12

def lognormal_pdf(x, mu, sigma):
    scale = np.exp(mu)
    return lognorm.pdf(x, s=sigma, scale=scale)


def lognormal_quantile(p, mu, sigma):
    scale = np.exp(mu)
    return lognorm.ppf(p, s=sigma, scale=scale)


def theoretical_variance_taylor(p, mu, sigma, n):
    q_p = lognormal_quantile(p, mu, sigma)
    f_q = lognormal_pdf(q_p, mu, sigma)
    return (p * (1 - p)) / (n * f_q**2)

def estimate_variance_golden(p, mu, sigma, n):
    return theoretical_variance_taylor(p, mu, sigma, n)


def estimate_variance_plugin(sample, p):
    n = len(sample)
    q_hat = np.quantile(sample, p)

    try:
        kde = stats.gaussian_kde(sample, bw_method='scott')
        f_hat = kde(q_hat)[0]
        if f_hat < 1e-10:
```

```
            return np.nan
        return (p * (1 - p)) / (n * f_hat**2)
    except:
        return np.nan


def estimate_variance_bootstrap(sample, p, B=1000):
    n = len(sample)
    bootstrap_quantiles = np.zeros(B)
    for b in range(B):
        resample = np.random.choice(sample, size=n, replace=True)
        bootstrap_quantiles[b] = np.quantile(resample, p)
    return np.var(bootstrap_quantiles, ddof=1)

SAMPLE_SIZES = [30, 100, 250, 500, 1000, 1500, 2000]
SIGMA_VALUES = [0.5, 1.0, 1.5]
QUANTILE_LEVELS = [0.5, 0.95, 0.99]
MU = 0
M = 5000
B = 1000

def run_simulation(n, sigma, p, M, B, mu=0):
    true_quantile = lognormal_quantile(p, mu, sigma)
    var_golden = estimate_variance_golden(p, mu, sigma, n)

    conf_level = 0.95
    alpha = 1 - conf_level
    z = norm.ppf(1 - alpha/2)

    sample_quantiles = np.zeros(M)
    var_plugin_estimates = np.zeros(M)
    var_bootstrap_estimates = np.zeros(M)

    for m in range(M):
        sample = np.random.lognormal(mean=mu, sigma=sigma, size=n)
        sample_quantiles[m] = np.quantile(sample, p)
        var_plugin_estimates[m] = estimate_variance_plugin(sample, p)
        var_bootstrap_estimates[m] = estimate_variance_bootstrap(sample, p, B)

    empirical_variance = np.var(sample_quantiles, ddof=1)

    results = {
        'n': n, 'sigma': sigma, 'p': p,
        'true_quantile': true_quantile,
        'empirical_variance': empirical_variance,
        'theoretical_variance': var_golden,
        'conf_level': conf_level
    }

    # 1. Oracle
    results['golden_mean_var'] = var_golden
    results['golden_bias'] = var_golden - empirical_variance
    results['golden_rel_bias'] = (var_golden - empirical_variance) / empirical_variance

    se_golden = np.sqrt(var_golden)
    lower_golden = sample_quantiles - z * se_golden
    upper_golden = sample_quantiles + z * se_golden
    results['golden_coverage'] = np.mean((lower_golden <= true_quantile) & (true_quantile <= upper_golden))

```

```
    # 2. Plug-in
    results['plugin_mean_var'] = np.nanmean(var_plugin_estimates)
    results['plugin_bias'] = np.nanmean(var_plugin_estimates) - empirical_variance
    results['plugin_rel_bias'] = (np.nanmean(var_plugin_estimates) - empirical_variance) / empirical_variance
    results['plugin_mse'] = np.nanmean((var_plugin_estimates - empirical_variance)**2)

    se_plugin = np.sqrt(var_plugin_estimates)
    lower_plugin = sample_quantiles - z * se_plugin
    upper_plugin = sample_quantiles + z * se_plugin
    coverage_plugin = (lower_plugin <= true_quantile) & (true_quantile <= upper_plugin)
    results['plugin_coverage'] = np.nanmean(coverage_plugin)

    # 3. Bootstrap
    results['bootstrap_mean_var'] = np.mean(var_bootstrap_estimates)
    results['bootstrap_bias'] = np.mean(var_bootstrap_estimates) - empirical_variance
    results['bootstrap_rel_bias'] = (np.mean(var_bootstrap_estimates) - empirical_variance) / empirical_variance
    results['bootstrap_mse'] = np.mean((var_bootstrap_estimates - empirical_variance)**2)

    se_bootstrap = np.sqrt(var_bootstrap_estimates)
    lower_bootstrap = sample_quantiles - z * se_bootstrap
    upper_bootstrap = sample_quantiles + z * se_bootstrap
    results['bootstrap_coverage'] = np.mean((lower_bootstrap <= true_quantile) & (true_quantile <= upper_bootstrap))

    return results

def main():
    parser = argparse.ArgumentParser(description='Run variance estimation simulation or plot existing results.')
    parser.add_argument('--plot-only', action='store_true', help='Skip simulation and plot results from CSV')
    args = parser.parse_args()

    if args.plot_only:
        if os.path.exists("semestralka/simulace/simulation_results.csv"):
            print("Loading results from simulation_results.csv...")
            results_df = pd.read_csv("semestralka/simulace/simulation_results.csv")
        else:
            print("Error: simulation_results.csv not found. Cannot plot without results.")
            return
    else:
        param_combinations = []
        for sigma in SIGMA_VALUES:
            for p in QUANTILE_LEVELS:
                for n in SAMPLE_SIZES:
                    param_combinations.append((n, sigma, p))

        workers = os.cpu_count()
        print(f"Starting simulation with {len(param_combinations)} tasks using ProcessPoolExecutor on {workers} cores...")

        results_list = []
        with ProcessPoolExecutor(max_workers=workers) as executor:
            future_to_params = {
```

```
            executor.submit(run_simulation, n, sigma, p, M, B, MU): (n, sigma, p)
            for n, sigma, p in param_combinations
        }

        for future in tqdm(as_completed(future_to_params), total=len(param_combinations), desc="Simulating"):
            try:
                result = future.result()
                results_list.append(result)
            except Exception as exc:
                n, sigma, p = future_to_params[future]
                print(f'Task (n={n}, sigma={sigma}, p={p}) generated an exception: {exc}')

    results_df = pd.DataFrame(results_list)
    results_df.to_csv('simulation_results.csv', index=False)

# Graf 1: Coverage Probability
fig, axes = plt.subplots(3, 3, figsize=(15, 15), sharey=True)
colors = {'golden': 'green', 'plugin': 'blue', 'bootstrap': 'red'}
labels = {'golden': 'Asymptotická', 'plugin': 'Plug-in (KDE)', 'bootstrap': 'Bootstrap'}
markers = {'golden': 'o', 'plugin': 's', 'bootstrap': '^'}

for row, p in enumerate(QUANTILE_LEVELS):
    target_cov = 0.95

    for col, sigma in enumerate(SIGMA_VALUES):
        ax = axes[row, col]
        subset = results_df[(results_df['p'] == p) & (results_df['sigma'] == sigma)]

        for method in ['golden', 'plugin', 'bootstrap']:
            ax.plot(subset['n'], subset[f'{method}_coverage'],
                    color=colors[method], marker=markers[method],
                    label=labels[method], linewidth=2, markersize=8)

        ax.axhline(y=target_cov, color='gray', linestyle='--', alpha=0.7, label=f'Nominální {target_cov*100:.0f}%')
        ax.set_xlabel('Velikost výběru (n)')
        ax.set_ylabel('Coverage Probability')
        ax.set_title(f'σ = {sigma}, p = {p} (Target {target_cov})')
        ax.set_ylim(0.5, 1.01)
        if (row == 2 and col == 2):
            ax.legend(loc='lower right', fontsize=9)
        ax.grid(True, alpha=0.3)

plt.suptitle('Coverage Probability (95% CI)', fontsize=14, fontweight='bold')
plt.tight_layout()
plt.savefig('simulace/coverage_probability.png', dpi=150, bbox_inches='tight')

# Graf 2: Relativní Bias
fig, axes = plt.subplots(3, 3, figsize=(15, 15))

for row, p in enumerate(QUANTILE_LEVELS):
    for col, sigma in enumerate(SIGMA_VALUES):
        ax = axes[row, col]
        subset = results_df[(results_df['p'] == p) & (results_df['sigma'] == sigma)]

```

```
            for method in ['golden', 'plugin', 'bootstrap']:
                ax.plot(subset['n'], subset[f'{method}_rel_bias'] * 100,
                        color=colors[method], marker=markers[method],
                        label=labels[method], linewidth=2, markersize=8)

            ax.axhline(y=0, color='gray', linestyle='--', alpha=0.7)
            ax.set_xlabel('Velikost výběru (n)')
            ax.set_ylabel('Relativní Bias (%)')
            ax.set_title(f'σ = {sigma}, p = {p}')
            ax.set_ylim(-100, 100)
            if (row == 0 and col == 0):
                ax.legend(loc='best', fontsize=9)
            ax.grid(True, alpha=0.3)

    plt.suptitle('Relativní Bias odhadu rozptylu', fontsize=14, fontweight='bold')
    plt.tight_layout()
    plt.savefig('simulace/relative_bias.png', dpi=150, bbox_inches='tight')

    # Graf 3: MSE Log-Log
    fig, axes = plt.subplots(3, 3, figsize=(15, 15))

    for row, p in enumerate(QUANTILE_LEVELS):
        for col, sigma in enumerate(SIGMA_VALUES):
            ax = axes[row, col]
            subset = results_df[(results_df['p'] == p) & (results_df['sigma'] == sigma)]

            for method in ['plugin', 'bootstrap']:
                ax.loglog(subset['n'], subset[f'{method}_mse'],
                          color=colors[method], marker=markers[method],
                          label=labels[method], linewidth=2, markersize=8)

            ax.set_xlabel('Velikost výběru (n)')
            ax.set_ylabel('MSE')
            ax.set_title(f'σ = {sigma}, p = {p}')
            if (row == 0 and col == 0):
                ax.legend(loc='best', fontsize=9)
            ax.grid(True, alpha=0.3, which='both')

    plt.suptitle('MSE odhadu rozptylu (log-log škála)', fontsize=14, fontweight='bold')
    plt.tight_layout()
    plt.savefig('simulace/mse_loglog.png', dpi=150, bbox_inches='tight')


if __name__ == '__main__':
    main()
```