



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	2
2	Teoretická část	3
2.1	Definice a značení	3
2.2	Odvození asymptotického rozptylu (Delta metoda)	3
3	Metodika simulační studie	5
3.1	Generování dat	5
3.2	Srovnávané metody	5
3.2.1	Teoretická delta metoda	5
3.2.2	Praktická delta metoda	5
3.2.3	Bootstrap	6
3.2.4	Konstrukce intervalů spolehlivosti	6
3.3	Hodnotící kritéria	6
3.3.1	Mean Squared Error (MSE)	7
3.3.2	Relative Bias (Relativní vychýlení)	7
3.3.3	Pokrytí intervalu spolehlivosti (Coverage Probability - CP)	7
4	Simulační studie	8
4.1	Pokrytí intervalu spolehlivosti	9
4.2	Systematické vychýlení (Relative Bias)	11
4.3	Konvergence chyby (MSE)	12
5	Závěr	14
	Použitá literatura	15
	Přílohy	16

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 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

Nechť X_1, X_2, \dots, X_n je náhodný výběr z rozdělení se spojitou distribuční funkcí $F(x)$ a hustotou pravděpodobnosti $f(x)$. Definujme p -tý teoretický kvantil q_p jako hodnotu, pro kterou platí:

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

Výběrový kvantil \hat{q}_p je definován pomocí empirické distribuční funkce $\hat{F}_n(x)$ jako (Hyndman & Fan, 1996):

$$\hat{q}_p = \hat{F}_n^{-1}(p) = \inf\{x : \hat{F}_n(x) \geq p\}. \quad (2)$$

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

$$f(x) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{(\ln x - \mu)^2}{2\sigma^2}}, \quad x > 0. \quad (3)$$

2.2 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:

$$\sqrt{n}(\hat{F}_n(x) - F(x)) \xrightarrow{d} N(0, F(x)(1 - F(x))). \quad (4)$$

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:

$$\sqrt{n}(\hat{q}_p - q_p) \xrightarrow{d} N\left(0, \frac{\text{Var}(\mathbb{I}(X \leq q_p))}{[f(q_p)]^2}\right). \quad (5)$$

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:

$$\text{AVar}(\hat{q}_p) = \frac{p(1 - p)}{n[f(q_p)]^2}. \quad (6)$$

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 $\sigma \in \{0.5, 1, 1.5\}$. Generujeme náhodné výběry o rozsahu $n \in \{30, 100, 250, 500, 1000, 1500, 2000\}$ pro simulaci malých, středních a velkých datových souborů.

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 Teoretická delta metoda

Tato metoda využívá znalosti skutečného rozdělení, ze kterého data pocházejí. Do vzorce (6) dosazujeme skutečnou hustotu $f(q_p)$ log-normálního rozdělení.

$$\widehat{SE}_{Oracle} = \sqrt{\frac{p(1-p)}{n[f_{LN}(q_p)]^2}}$$

Tato metoda slouží jako ideální v praxi nedosažitelný benchmark, protože v reálných situacích povětšinou neznáme skutečné rozdělení dat.

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).

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

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):

$$CI_{0.95} = [\hat{q}_p - z_{0.975} \cdot \widehat{SE}, \quad \hat{q}_p + z_{0.975} \cdot \widehat{SE}], \quad (1)$$

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 ($m = 1, \dots, M$).

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:

$$\text{MSE}(\hat{\theta}) = \frac{1}{M} \sum_{m=1}^M (\hat{\theta}_m - \theta)^2 \quad (2)$$

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.

$$\text{RB}(\hat{\theta}) = \frac{\frac{1}{M} \sum_{m=1}^M \hat{\theta}_m - \theta}{\theta} \cdot 100 \% \quad (3)$$

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).

$$\text{CP} = \frac{1}{M} \sum_{m=1}^M \mathbb{I}(\hat{\theta}_{L,m} \leq q_p \leq \hat{\theta}_{U,m}) \quad (4)$$

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).

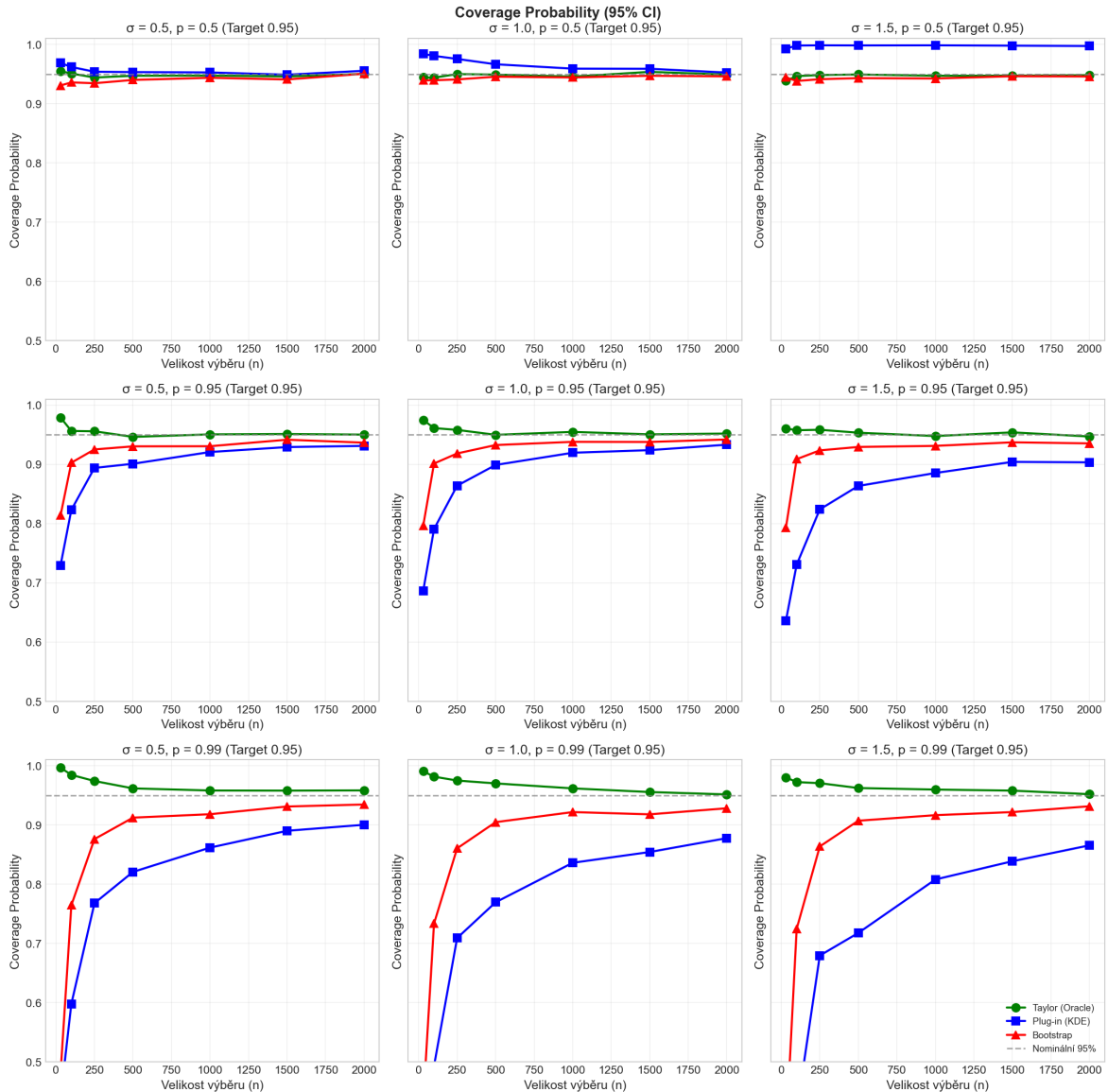
4. Simulační studie

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).

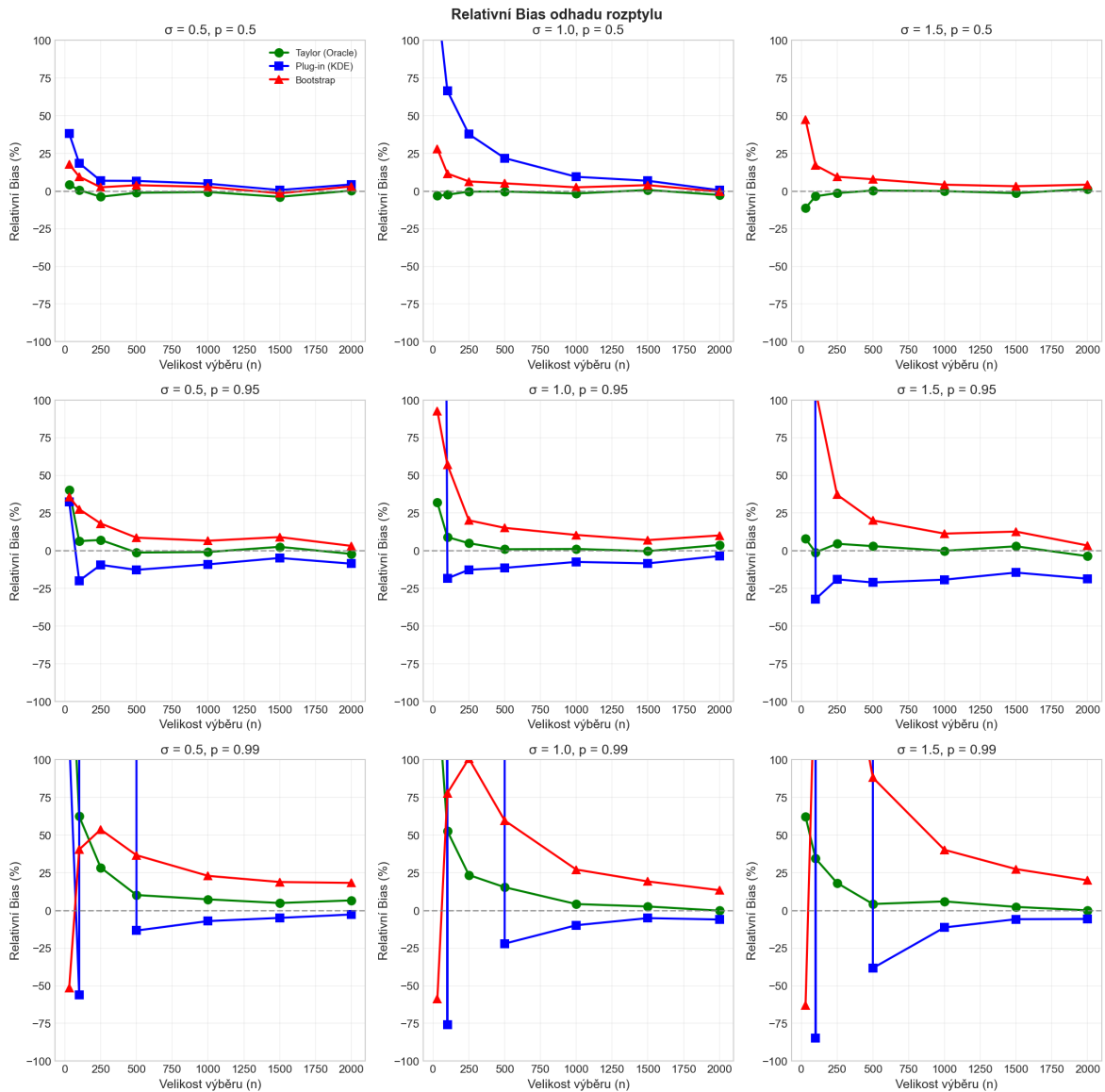
Výsledky ukazují zřetelný rozdíl v chování metod v závislosti na parametrech simulace:

1. **Teoretická delta metoda (zelená):** Tato referenční metoda konzistentně dosahuje nominálního pokrytí ve všech scénářích. To potvrzuje, že samotná asymptotická aproximace (6) 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.

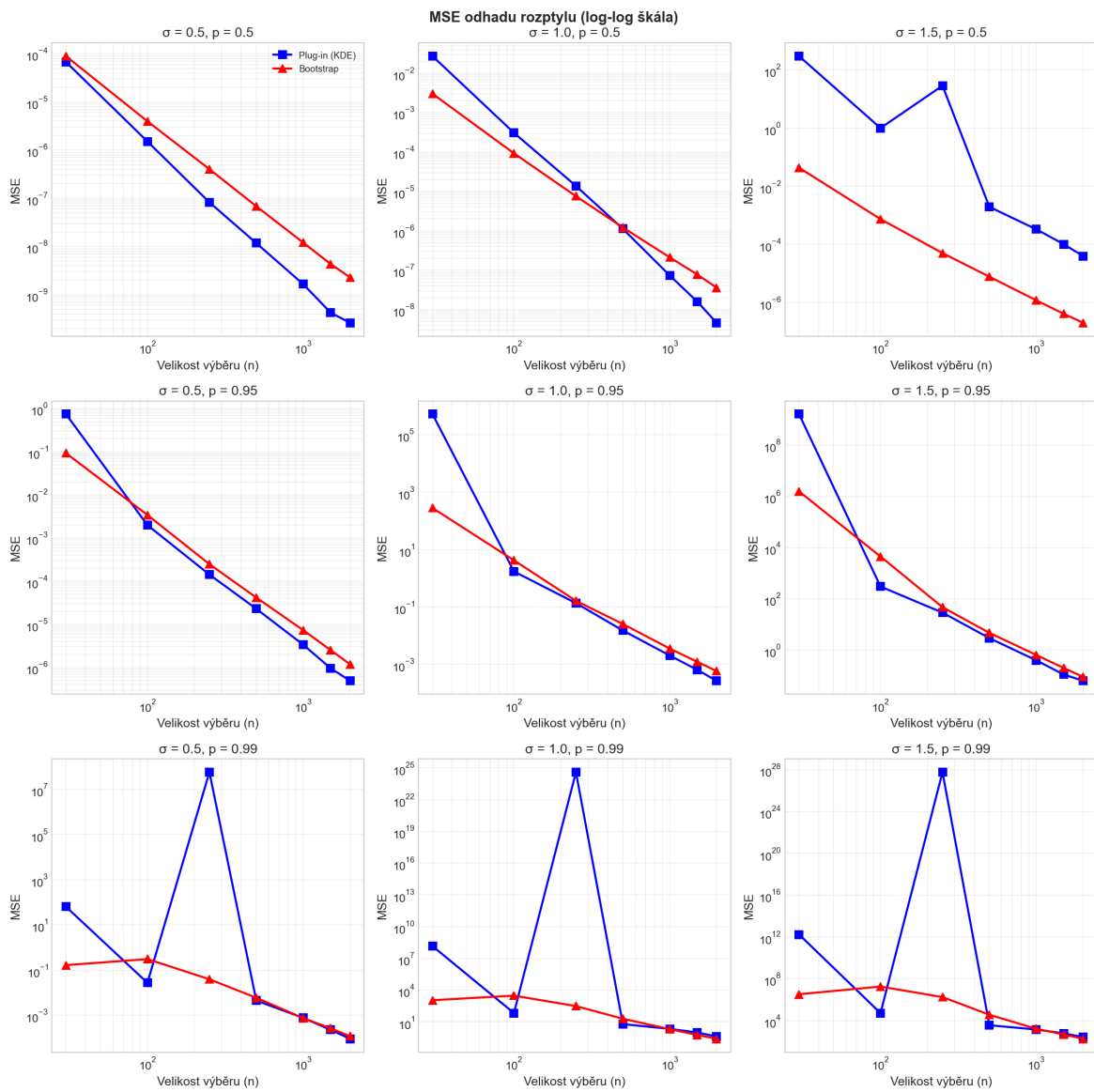
1. **Plug-in metoda (modrá):** Vykazuje silné záporné vychýlení pro extrémní kvantily ve chvostech s vysokou variabilitou ($\sigma = 1.5$). Vychýlení dosahuje hodnot okolo -40 % až -50 %. Příčinou je vyhlazovací efekt jádrového odhadu hustoty (KDE). Ve chvostech log-normálního rozdělení, kde hustota 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

```
1  import numpy as np
2  import os
3  import argparse
4  from concurrent.futures import ProcessPoolExecutor, as_completed
5  import pandas as pd
6  import matplotlib.pyplot as plt
7  from scipy import stats
8  from scipy.stats import norm, lognorm
9  from tqdm import tqdm
10 import warnings
11 import seaborn as sns
12
13 warnings.filterwarnings('ignore')
14
15 np.random.seed(42)
16
17 try:
18     plt.style.use('seaborn-v0_8-whitegrid')
19 except:
20     sns.set_style("whitegrid")
21
22 plt.rcParams['figure.figsize'] = (15, 10)
23 plt.rcParams['font.size'] = 11
24 plt.rcParams['axes.titlesize'] = 13
25 plt.rcParams['axes.labelsize'] = 12
26
27 def lognormal_pdf(x, mu, sigma):
28     scale = np.exp(mu)
29     return lognorm.pdf(x, s=sigma, scale=scale)
30
31
32 def lognormal_quantile(p, mu, sigma):
33     scale = np.exp(mu)
34     return lognorm.ppf(p, s=sigma, scale=scale)
35
36
37 def theoretical_variance_taylor(p, mu, sigma, n):
38     q_p = lognormal_quantile(p, mu, sigma)
39     f_q = lognormal_pdf(q_p, mu, sigma)
40     return (p * (1 - p)) / (n * f_q**2)
41
42
43 def estimate_variance_golden(p, mu, sigma, n):
44     return theoretical_variance_taylor(p, mu, sigma, n)
45
46
47 def estimate_variance_plugin(sample, p):
48     n = len(sample)
49     q_hat = np.quantile(sample, p)
50
51     try:
52         kde = stats.gaussian_kde(sample, bw_method='scott')
53         f_hat = kde(q_hat)[0]
54         if f_hat < 1e-10:
```

```

54         return np.nan
55     return (p * (1 - p)) / (n * f_hat**2)
56 except:
57     return np.nan
58
59
60 def estimate_variance_bootstrap(sample, p, B=1000):
61     n = len(sample)
62     bootstrap_quantiles = np.zeros(B)
63     for b in range(B):
64         resample = np.random.choice(sample, size=n, replace=True)
65         bootstrap_quantiles[b] = np.quantile(resample, p)
66     return np.var(bootstrap_quantiles, ddof=1)
67
68 SAMPLE_SIZES = [30, 100, 250, 500, 1000, 1500, 2000]
69 SIGMA_VALUES = [0.5, 1.0, 1.5]
70 QUANTILE_LEVELS = [0.5, 0.95, 0.99]
71 MU = 0
72 M = 5000
73 B = 1000
74
75 def run_simulation(n, sigma, p, M, B, mu=0):
76     true_quantile = lognormal_quantile(p, mu, sigma)
77     var_golden = estimate_variance_golden(p, mu, sigma, n)
78
79     conf_level = 0.95
80     alpha = 1 - conf_level
81     z = norm.ppf(1 - alpha/2)
82
83     sample_quantiles = np.zeros(M)
84     var_plugin_estimates = np.zeros(M)
85     var_bootstrap_estimates = np.zeros(M)
86
87     for m in range(M):
88         sample = np.random.lognormal(mean=mu, sigma=sigma, size=n)
89         sample_quantiles[m] = np.quantile(sample, p)
90         var_plugin_estimates[m] = estimate_variance_plugin(sample, p)
91         var_bootstrap_estimates[m] = estimate_variance_bootstrap(sample, p, B)
92
93     empirical_variance = np.var(sample_quantiles, ddof=1)
94
95     results = {
96         'n': n, 'sigma': sigma, 'p': p,
97         'true_quantile': true_quantile,
98         'empirical_variance': empirical_variance,
99         'theoretical_variance': var_golden,
100        'conf_level': conf_level
101    }
102
103    # 1. Oracle
104    results['golden_mean_var'] = var_golden
105    results['golden_bias'] = var_golden - empirical_variance
106    results['golden_rel_bias'] = (var_golden - empirical_variance) / empirical_
107    variance
108
109    se_golden = np.sqrt(var_golden)
110    lower_golden = sample_quantiles - z * se_golden
111    upper_golden = sample_quantiles + z * se_golden
112    results['golden_coverage'] = np.mean((lower_golden <= true_quantile) & (true_
    quantile <= upper_golden))

```

```

113     # 2. Plug-in
114     results['plugin_mean_var'] = np.nanmean(var_plugin_estimates)
115     results['plugin_bias'] = np.nanmean(var_plugin_estimates) - empirical_variance
116     results['plugin_rel_bias'] = (np.nanmean(var_plugin_estimates) - empirical_
117     variance) / empirical_variance
118     results['plugin_mse'] = np.nanmean((var_plugin_estimates - empirical_variance)*
119     *2)
120
121     se_plugin = np.sqrt(var_plugin_estimates)
122     lower_plugin = sample_quantiles - z * se_plugin
123     upper_plugin = sample_quantiles + z * se_plugin
124     coverage_plugin = (lower_plugin <= true_quantile) & (true_quantile <= upper_
125     plugin)
126     results['plugin_coverage'] = np.nanmean(coverage_plugin)
127
128     # 3. Bootstrap
129     results['bootstrap_mean_var'] = np.mean(var_bootstrap_estimates)
130     results['bootstrap_bias'] = np.mean(var_bootstrap_estimates) - empirical_
131     variance
132     results['bootstrap_rel_bias'] = (np.mean(var_bootstrap_estimates) - empirical_
133     variance) / empirical_variance
134     results['bootstrap_mse'] = np.mean((var_bootstrap_estimates - empirical_
135     variance)**2)
136
137     se_bootstrap = np.sqrt(var_bootstrap_estimates)
138     lower_bootstrap = sample_quantiles - z * se_bootstrap
139     upper_bootstrap = sample_quantiles + z * se_bootstrap
140     results['bootstrap_coverage'] = np.mean((lower_bootstrap <= true_quantile) & (
141     true_quantile <= upper_bootstrap))
142
143     return results
144
145 def main():
146     parser = argparse.ArgumentParser(description='Run variance estimation
147     simulation or plot existing results.')
148     parser.add_argument('--plot-only', action='store_true', help='Skip simulation
149     and plot results from CSV')
150     args = parser.parse_args()
151
152     if args.plot_only:
153         if os.path.exists("semestralka/simulace/simulation_results.csv"):
154             print("Loading results from simulation_results.csv...")
155             results_df = pd.read_csv("semestralka/simulace/simulation_results.csv")
156         else:
157             print("Error: simulation_results.csv not found. Cannot plot without
158             results.")
159         return
160     else:
161         param_combinations = []
162         for sigma in SIGMA_VALUES:
163             for p in QUANTILE_LEVELS:
164                 for n in SAMPLE_SIZES:
165                     param_combinations.append((n, sigma, p))
166
167         workers = os.cpu_count()
168         print(f"Starting simulation with {len(param_combinations)} tasks using
169         ProcessPoolExecutor on {workers} cores...")
170
171         results_list = []
172         with ProcessPoolExecutor(max_workers=workers) as executor:
173             future_to_params = {

```

```

163         executor.submit(run_simulation, n, sigma, p, M, B, MU): (n, sigma,
164 p)
165         for n, sigma, p in param_combinations
166     }
167     for future in tqdm(as_completed(future_to_params), total=len(param_
combinations), desc="Simulating"):
168         try:
169             result = future.result()
170             results_list.append(result)
171         except Exception as exc:
172             n, sigma, p = future_to_params[future]
173             print(f'Task (n={n}, sigma={sigma}, p={p}) generated an
exception: {exc}')
174
175     results_df = pd.DataFrame(results_list)
176     results_df.to_csv('simulation_results.csv', index=False)
177
178     # Graf 1: Coverage Probability
179     fig, axes = plt.subplots(3, 3, figsize=(15, 15), sharey=True)
180     colors = {'golden': 'green', 'plugin': 'blue', 'bootstrap': 'red'}
181     labels = {'golden': 'Taylor (Oracle)', 'plugin': 'Plug-in (KDE)', 'bootstrap':
'Bootstrap'}
182     markers = {'golden': 'o', 'plugin': 's', 'bootstrap': '^'}
183
184     for row, p in enumerate(QUANTILE_LEVELS):
185         target_cov = 0.95
186
187         for col, sigma in enumerate(SIGMA_VALUES):
188             ax = axes[row, col]
189             subset = results_df[(results_df['p'] == p) & (results_df['sigma'] ==
sigma)]
190
191             for method in ['golden', 'plugin', 'bootstrap']:
192                 ax.plot(subset['n'], subset[f'{method}_coverage'],
193                     color=colors[method], marker=markers[method],
194                     label=labels[method], linewidth=2, markersize=8)
195
196                 ax.axhline(y=target_cov, color='gray', linestyle='--', alpha=0.7, label
=f'Nominální {target_cov*100:.0f}%')
197                 ax.set_xlabel('Velikost výběru (n)')
198                 ax.set_ylabel('Coverage Probability')
199                 ax.set_title(f' $\sigma = \{sigma\}$ ,  $p = \{p\}$  (Target {target_cov})')
200                 ax.set_ylim(0.5, 1.01)
201                 if (row == 2 and col == 2):
202                     ax.legend(loc='lower right', fontsize=9)
203                 ax.grid(True, alpha=0.3)
204
205     plt.suptitle('Coverage Probability (95% CI)', fontsize=14, fontweight='bold')
206     plt.tight_layout()
207     plt.savefig('coverage_probability.png', dpi=150, bbox_inches='tight')
208
209     # Graf 2: Relativní Bias
210     fig, axes = plt.subplots(3, 3, figsize=(15, 15))
211
212     for row, p in enumerate(QUANTILE_LEVELS):
213         for col, sigma in enumerate(SIGMA_VALUES):
214             ax = axes[row, col]
215             subset = results_df[(results_df['p'] == p) & (results_df['sigma'] ==
sigma)]
216

```

```

217         for method in ['golden', 'plugin', 'bootstrap']:
218             ax.plot(subset['n'], subset[f'{method}_rel_bias'] * 100,
219                     color=colors[method], marker=markers[method],
220                     label=labels[method], linewidth=2, markersize=8)
221
222         ax.axhline(y=0, color='gray', linestyle='--', alpha=0.7)
223         ax.set_xlabel('Velikost výběru (n)')
224         ax.set_ylabel('Relativní Bias (%)')
225         ax.set_title(f' $\sigma = \{sigma\}$ ,  $p = \{p\}$ ')
226         ax.set_ylim(-100, 100)
227         if (row == 0 and col == 0):
228             ax.legend(loc='best', fontsize=9)
229         ax.grid(True, alpha=0.3)
230
231 plt.suptitle('Relativní Bias odhadu rozptylu', fontsize=14, fontweight='bold')
232 plt.tight_layout()
233 plt.savefig('relative_bias.png', dpi=150, bbox_inches='tight')
234
235 # Graf 3: MSE Log-Log
236 fig, axes = plt.subplots(3, 3, figsize=(15, 15))
237
238 for row, p in enumerate(QUANTILE_LEVELS):
239     for col, sigma in enumerate(SIGMA_VALUES):
240         ax = axes[row, col]
241         subset = results_df[(results_df['p'] == p) & (results_df['sigma'] ==
242             sigma)]
243
244         for method in ['plugin', 'bootstrap']:
245             ax.loglog(subset['n'], subset[f'{method}_mse'],
246                     color=colors[method], marker=markers[method],
247                     label=labels[method], linewidth=2, markersize=8)
248
249         ax.set_xlabel('Velikost výběru (n)')
250         ax.set_ylabel('MSE')
251         ax.set_title(f' $\sigma = \{sigma\}$ ,  $p = \{p\}$ ')
252         if (row == 0 and col == 0):
253             ax.legend(loc='best', fontsize=9)
254         ax.grid(True, alpha=0.3, which='both')
255
256 plt.suptitle('MSE odhadu rozptylu (log-log škála)', fontsize=14, fontweight='
257 bold')
258 plt.tight_layout()
259 plt.savefig('mse_loglog.png', dpi=150, bbox_inches='tight')
260
261 if __name__ == '__main__':
262     main()

```