

Calibration Model 03: Validace modelu (Model Validation)

```
In [1]: # Instalace potřebných knihoven  
#%pip install pandas  
#%pip install numpy
```

```
In [2]: # Import potřebných knihoven  
import pandas as pd  
import numpy as np  
  
from scipy import stats  
from scipy.stats import ks_2samp, ttest_ind, mannwhitneyu  
  
import pandas as pd  
import matplotlib.pyplot as plt
```

Načtení reálných a syntetických dat

```
In [3]: # Soubor je načten a přiřazen do proměnné ,df_syn"  
other_path = '../data/05_Calibration/synthetic_dataset.csv'  
df_syn = pd.read_csv(other_path, header=0)
```

```
In [4]: # Zobrazení prvních 5 řádků datasetu  
print('Prvních 5 řádků datového rámce')  
df_syn.head(5)
```

Prvních 5 řádků datového rámce

Out[4]:

	dist	total_time
0	5483.830858	40.567093
1	9013.410179	49.481206
2	5679.449597	40.936673
3	9643.054314	49.766864
4	8990.433929	49.055589

```
In [5]: # Základní deskriptivní statistika syntetického datasetu  
global_distribution = df_syn[['total_time']]  
df_syn.describe()
```

Out[5]:

	dist	total_time
count	500000.000000	500000.000000
mean	6002.213262	45.777590
std	2314.176711	20.917843
min	2000.224407	29.722498
25%	4009.035893	36.821901
50%	5983.489163	41.987051
75%	8061.298159	47.360877
max	9999.961862	215.749756

Načtení souboru reálných dat z izolované sady měření

Syntetická data se validují **proti reálným datům, která nebyla použita při kalibraci modelu.**

```
In [6]: # Soubor je načten a přiřazen do proměnné ,real_validation_df"
other_path = '../data/06_AI/val/valid_timelaps.csv'
real_validation_df = pd.read_csv(other_path, header=0)
```

```
In [7]: # Zobrazení prvních 5 řádků datasetu
print('Prvních 5 řádků datového rámce')
real_validation_df.head(5)
```

Prvních 5 řádků datového rámce

```
Out[7]:   id    x     y     z  time  delay type_delay  total_time
0   13  220  2940     0    32      0        0       32
1   77  220  1690   500    33     23        2       56
2   220  2190    220  1750    35      0        0       35
3   105  252    220    750    53      0        0       53
4    45  2190    220    250    45      0        0       45
```

```
In [8]: # SPECIFIKACE TECHNOLOGICKÉHO PROCESU ZDĚNÍ
```

```
brick_thickness = 440      # mm, tloušťka zdicího prvku (Porotherm 440 Profi)
brick_height = 250         # mm, výška zdicího prvku (Porotherm 440 Profi)
brick_width = 250          # mm, šířka zdicího prvku (Porotherm 440 Profi)
```

```
# SOUŘADNICE REFERENČNÍHO BODU (nad verifikacním stolem)
```

```
refer_x = 2_000           # mm, souřadnice X referenčního bodu
refer_y = 3_500           # mm, souřadnice Y referenčního bodu
refer_z = 1_000           # mm, souřadnice Z referenčního bodu
```

```
In [9]: def calculation_dist(x, y, z):
    """
```

Funkce pro výpočet dráhy trajektorie od referenčního bodu k cílové poloze prvku

Parametry:

x, y, z (int): souřadnice cílové polohy prvku [mm]

Návratová hodnota:

```

    dist (int): dráha trajektorie od referenčního bodu k cílové poloze prvku [mm]
    """
    dist = 0

    # Fáze 1: dráha od referenčního bodu k cílové stěně.
    dist = ((refer_z - (z + brick_height*2))**2 + (refer_x - brick_thickness//2)**2)**(1/2)

    # Fáze 2: dráha ve směru osy X
    if x != brick_thickness / 2:
        dist = dist + abs(x - brick_thickness//2 + brick_width * 2)

    # Fáze 3: dráha ve směru osy Y
    if y != brick_thickness / 2:
        dist = dist + abs(y - refer_y + brick_width * 2)

    return int(dist)

```

In [10]:

```

# Funkce pro výpočet dráhy trajektorie od referenčního bodu k cílové poloze prvku
# Pro každý řádek datového rámce je aplikována funkce calculation_dist
# na základě souřadnic 'x', 'y', 'z'.
real_validation_df['dist'] = real_validation_df.apply(lambda x : calculation_dist(x['x'],x['y'],

```

In [11]:

```

# Základní deskriptivní statistika datasetu
real_validation_df.describe()

```

Out[11]:

	id	x	y	z	time	delay	type_delay	total_ti
count	108.000000	108.000000	108.000000	108.000000	108.000000	108.000000	108.000000	108.000000
mean	141.120370	1273.351852	992.740741	1013.888889	36.472222	5.324074	0.231481	41.796333
std	79.777578	1240.172134	1274.175272	718.443064	6.443802	27.813743	0.804270	28.458333
min	1.000000	95.000000	95.000000	0.000000	22.000000	0.000000	0.000000	22.000000
25%	77.750000	220.000000	220.000000	500.000000	32.000000	0.000000	0.000000	32.750000
50%	136.500000	690.000000	220.000000	875.000000	36.000000	0.000000	0.000000	36.500000
75%	212.250000	2221.250000	1690.000000	1750.000000	41.000000	0.000000	0.000000	42.000000
max	276.000000	4002.000000	4565.000000	2250.000000	58.000000	260.000000	4.000000	296.000000

Omezení syntetických dat

In [12]:

```

df_syn = df_syn[
    (df_syn["dist"] >= real_validation_df['dist'].min()) &
    (df_syn["dist"] <= real_validation_df['dist'].max())
]

```

In [13]:

```
df_syn.describe()
```

Out[13]:

	dist	total_time
count	281025.000000	281025.000000
mean	4247.312817	41.470066
std	1296.278484	20.363578
min	2000.224407	29.722498
25%	3138.655480	34.522064
50%	4247.387445	37.481842
75%	5387.460387	40.406839
max	6465.716208	204.340501

Coverage test (Kolik reálných bodů leží v 95% CI)

```
In [14]: T_lower_interp = np.percentile(global_distribution, 2.5)
T_upper_interp = np.percentile(global_distribution, 97.5)

real_time = real_validation_df["total_time"].values

within_ci = (
    (real_time >= T_lower_interp) &
    (real_time <= T_upper_interp)
)

coverage = within_ci.mean()
coverage
```

Out[14]: np.float64(0.7314814814814815)

Monte Carlo Validation: Opakování podvzorkování na velikost reality

```
In [15]: # Bootstrap vzorkování z kalibrace na 1/3 velikosti reálného datasetu
n_real = len(real_validation_df)//3

syn_samples = []

for _ in range(1000):
    sample = df_syn.sample(n=n_real, replace=True, random_state=122 + _)
    syn_samples.append(sample['total_time'])
```

KS test pro každé podvzorkování

```
In [16]: # Výpočet KS p-hodnot pro porovnání reálných a kalibrovaných dat
p_vals = []

for s in syn_samples:
    _, p = ks_2samp(real_validation_df['total_time'], s)
    p_vals.append(p)
```

Pravděpodobnost shody modelu

```
In [17]: valid_ratio = np.mean(np.array(p_vals) > 0.05)

print("Podíl validních simulací:", valid_ratio)
```

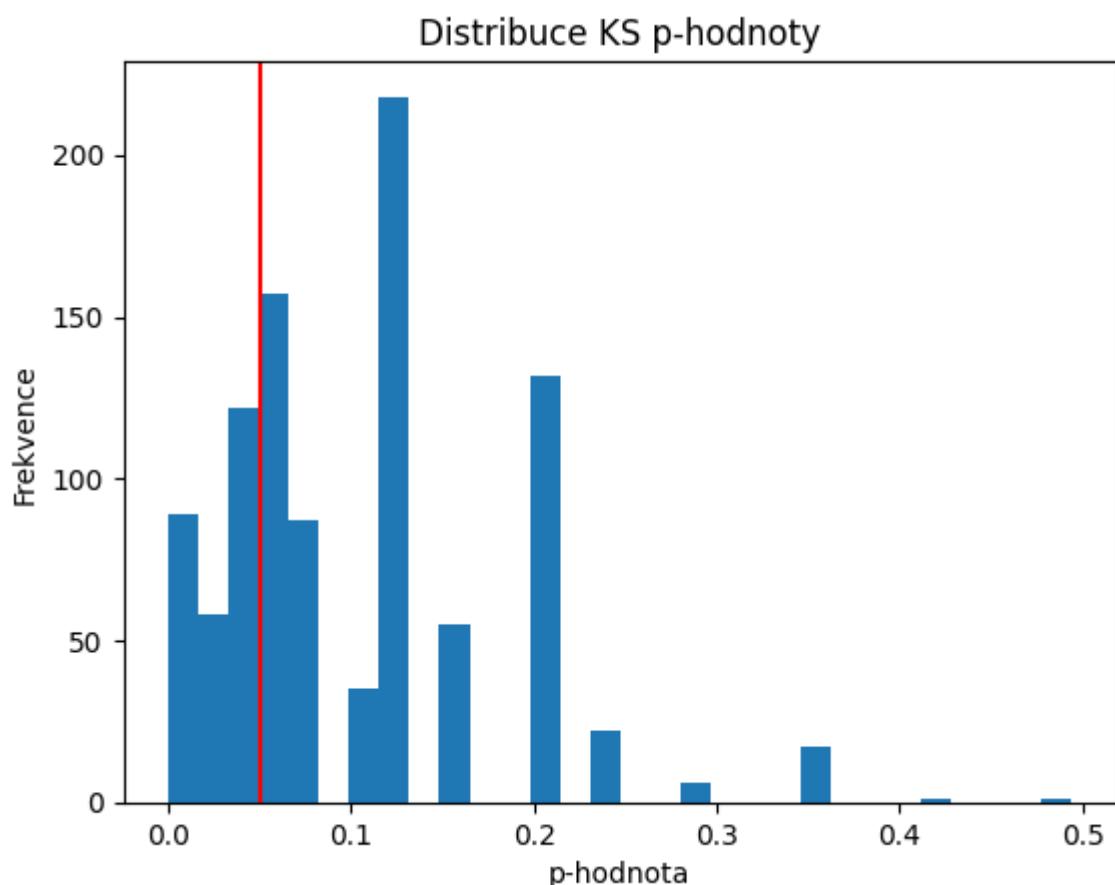
Podíl validních simulací: 0.731

Distribuce KS p-hodnoty

In [18]:

```
plt.figure()
plt.hist(p_vals, bins=30)
plt.axvline(0.05, color='red')

plt.title("Distribuce KS p-hodnoty")
plt.xlabel("p-hodnota")
plt.ylabel("Frekvence")
plt.show()
```



Interpretace (DES validace)

| Podíl | Interpretace | | ----- | ----- | 0.8 | model VALIDNÍ | | 0.5 – 0.8 | model přijatelný | | < 0.5 | model nevalidní |

Ve 73.1 % bootstrap vzorků z kalibrace nelze statisticky rozlišit kalibraci od reality (na hladině významnosti $\alpha = 0.05$ pomocí KS testu).

Porovnání průměru a směrodatné odchyly

Reálná data – referenční hodnoty:

In [19]:

```
# Výpočet průměru a směrodatné odchyly z reálných dat
mean_real = real_validation_df['total_time'].mean()
std_real = real_validation_df['total_time'].std()

print("Real Mean:", mean_real)
print("Real STD:", std_real)
```

Real Mean: 41.7962962962963
Real STD: 28.458449547372744

Bootstrap z kalibrace (na velikost reality):

```
In [20]: # velikost reálného datasetu
n_real = len(real_validation_df)

syn_means = []
syn_stds = []

# opakování podvzorkování simulace
for i in range(1000):

    sample = df_syn.sample(
        n=n_real,
        replace=True,
        random_state=122 + i
    )

    syn_means.append(sample['total_time'].mean())
    syn_stds.append(sample['total_time'].std())
```

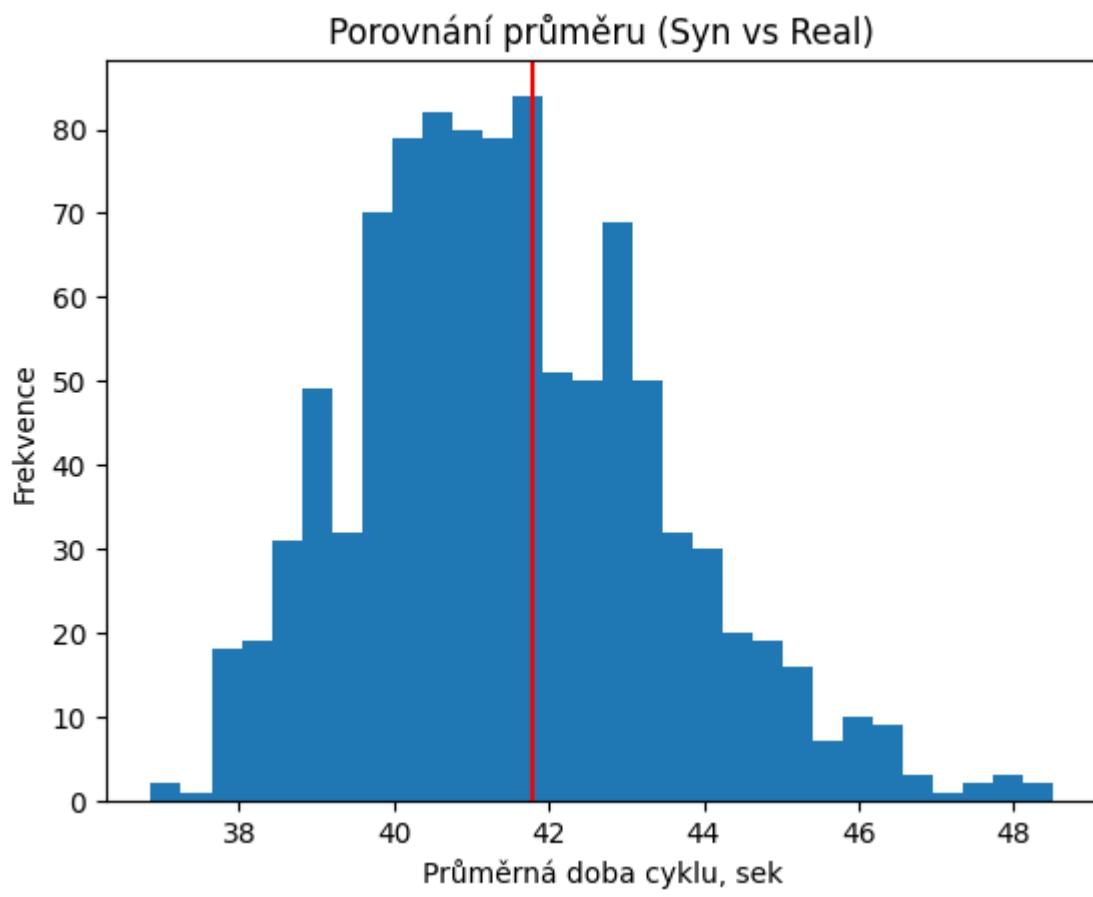
Porovnání průměru

```
In [21]: # Histogram průměrů z kalibrace s vyznačením průměru reálných dat
plt.figure()

plt.hist(syn_means, bins=30)
plt.axvline(mean_real, color='red')

plt.title("Porovnání průměru (Syn vs Real)")
plt.xlabel("Průměrná doba cyklu, sek")
plt.ylabel("Frekvence")

plt.show()
```



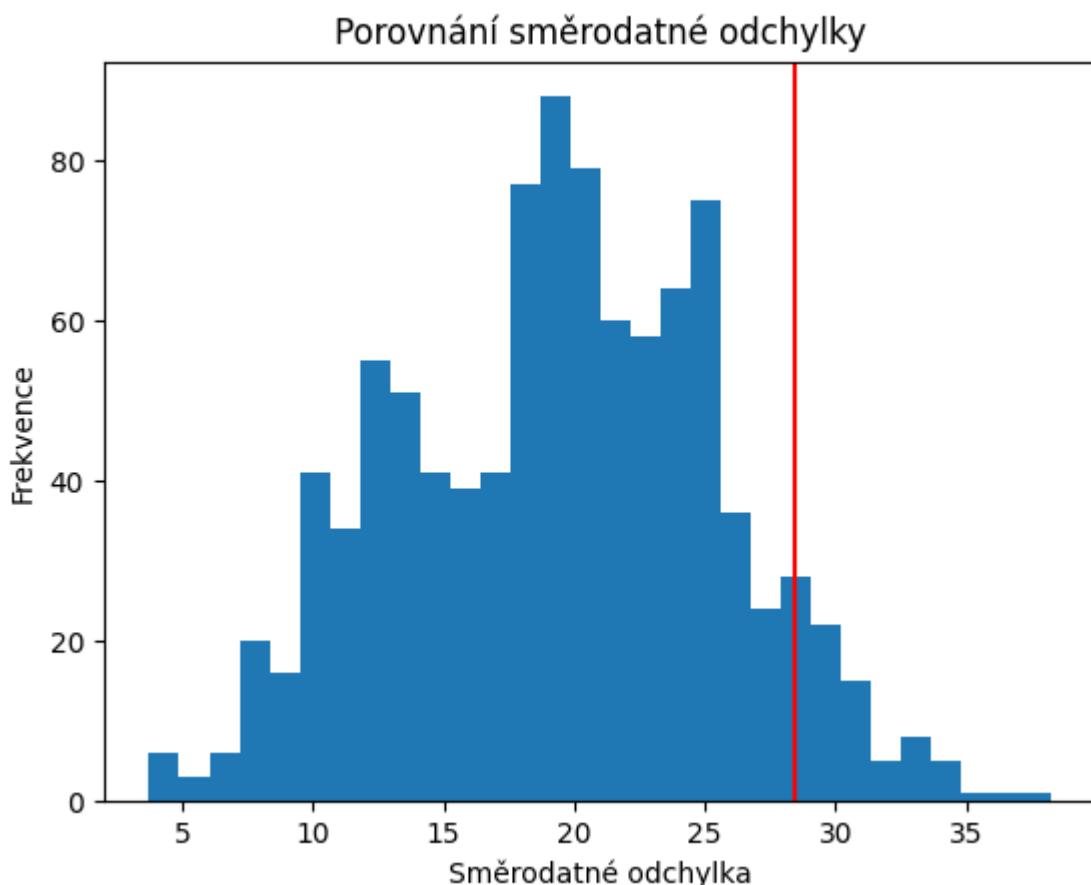
Porovnání STD

```
In [22]: # Histogram STD z kalibrace s vyznačením průměru reálných dat
plt.figure()

plt.hist(syn_stds, bins=30)
plt.axvline(std_real, color='red')

plt.title("Porovnání směrodatné odchylky")
plt.xlabel("Směrodatné odchylka")
plt.ylabel("Frekvence")

plt.show()
```



Kvantilová validace

```
In [23]: mean_CI = np.percentile(syn_means, [2.5, 97.5])
std_CI = np.percentile(syn_stds, [2.5, 97.5])

print("Mean 95% CI:", mean_CI)
print("STD 95% CI:", std_CI)

print("Real mean:", mean_real)
print("Real STD:", std_real)
```

Mean 95% CI: [38.11782809 45.89966754]
 STD 95% CI: [7.63934824 30.93422506]
 Real mean: 41.7962962962963
 Real STD: 28.458449547372744

Model je validní, ale rozsah hodnot je moc široký -> **velký význam stochastických vlivů**.

Parametrické porovnání dat

Welchův t-test

```
In [24]: stat, p = ttest_ind(
    real_validation_df['total_time'],
    df_syn['total_time'],
    equal_var=False
)

print("Welch t-test p-value:", p)
```

Welch t-test p-value: 0.9054042298576069

$p > 0.05 \rightarrow$ nelze zamítnout nulovou hypotézu o shodě středních hodnot porovnávaných souborů

Cohen's d (velikost efektu)

```
In [25]: mean_diff = abs(real_validation_df['total_time'].mean() - df_syn['total_time'].mean())
pooled_std = np.sqrt(
    (real_validation_df['total_time'].std()**2 + df_syn['total_time'].std()**2) / 2
)
d = mean_diff / pooled_std
print("Cohen's d:", d)
```

Cohen's d: 0.01318405048140204

Cohen's d < 0.2 → zanedbatelný význam

Neparametrické porovnání (Distribuce)

Mann–Whitney U test

```
In [26]: stat, p = mannwhitneyu(
    real_validation_df['total_time'],
    df_syn['total_time']
)
print("Mann-Whitney p-value:", p)
```

Mann-Whitney p-value: 0.18632449055958722

Vyhodnocení shody simulovaných a reálných dat

Bootstrap KS test

0,731

Interpretace:

Ve 73.1 % případů nelze statisticky rozlišit simulaci od reality .

STD

Real STD: 28.458449547372744 STD 95% CI: [7.63934824 30.93422506]

Interpretace:

Reálná směrodatná odchylka je v intervalu, avšak simulace vykazuje nadměrnou variabilitu.

Průměr

Real mean: 41.7962962962963 Mean 95% CI: [38.11782809 45.89966754]

Interpretace:

Reálný průměr se nachází v intervalu simulace

Welch t-test

p = 0.9054042298576069

Interpretace:

Hodnota p-value = 0.905 je výrazně vyšší než běžně používaná hladina významnosti ($\alpha = 0.05$). Na základě výsledku Welchova t-testu tedy nelze zamítнуть nulovou hypotézu o shodě středních hodnot porovnávaných souborů. Nebyl prokázán statisticky významný rozdíl mezi průměry sledovaných skupin a případné odlišnosti lze přičíst náhodné variabilitě dat.

Cohen's d

0.0131840504814020

Interpretace:

Znamená prakticky nulový efekt. Hodnota Cohenova koeficientu $d = 0.013$ indikuje zanedbatelnou velikost efektu mezi reálnými a syntetickými daty. Rozdíl mezi jejich středními hodnotami je tedy prakticky nulový a nemá významný věcný dopad, což je v souladu s výsledkem Welchova t-testu, který neprokázal statisticky významný rozdíl mezi soubory.

Mann–Whitney U test (neparametrický)

p = 0.186

Interpretace:

$p > 0,05 \rightarrow$ nelze zamítнуть nulovou hypotézu.

Výsledky Mann-Whitneyova testu ($p = 0.186$) neprokázaly statisticky významný rozdíl mezi rozdělením reálných a syntetických dat při hladině významnosti 5 %. Lze tedy konstatovat, že model reprodukuje pozorovaná data bez systematického posunu v mediánu.

Celkové vyhodnocení shody simulovaných a reálných dat

Celková validační analýza prokázala vysokou míru shody mezi simulovanými a reálnými daty ve všech posuzovaných charakteristikách. Bootstrapovaný Kolmogorov–Smirnovův test indikoval, že v 73,1 % případů nelze statisticky rozlišit simulaci od reality, což potvrzuje dobrou distribuční konzistenci modelu. Reálná směrodatná odchylka (28,46) se nachází uvnitř 95% intervalu spolehlivosti simulace [7,64; 30,93], přičemž model vykazuje mírně zvýšenou variabilitu, která však zůstává statisticky přijatelná. Reálný průměr (41,80) leží uvnitř 95% intervalu simulace [38,12; 45,90]. Welchův t-test neprokázal statisticky významný rozdíl mezi průměry ($p = 0,905$), Cohenovo $d = 0,013$ indikuje prakticky nulový efekt a neparametrický Mann–Whitneyův test rovněž nepotvrdil rozdíl mezi rozděleními ($p = 0,186$). Souhrnně lze konstatovat, že model reprodukuje reálná data bez statisticky ani věcně významných odchylek a je v analyzovaném rozsahu empiricky validován.

Autor / Organizace / Datum

Vjačeslav Usmanov, ČVUT v Praze, Fakulta stavební

Přehled změn

Datum (YYYY-MM-DD)	Verze	Autor změny	Popis změny
--------------------	-------	-------------	-------------

Datum (YYYY-MM-DD)	Verze	Autor změny	Popis změny
2026-01-27	1.1	Vjačeslav Usmanov	added CM_03_Model_Validation.ipynb
2026-02-16	1.2	Vjačeslav Usmanov	changed CM_03_Model_Validation.ipynb