# Data Science 01: Příprava a čištění dat (Data Preparation / Cleaning)

## Získávání dat (Data Acquisition)

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

In [2]:
```
# Import potřebných knihoven
import pandas as pd
import numpy as np
```

Dataset `data_timelaps`, vzniklý na základě časosběrného monitorování robotických zdicích prací, je uložen ve formátu CSV. Obsahuje dobu pracovního cyklu bez stochastických prodlev.

In [3]:
```
# Soubor je načten a přiřazen do proměnné ‚df_timelaps‘
other_path = '../../data/00_Raw/data_timelaps.csv'
df_timelaps = pd.read_csv(other_path, header=0)
```

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

```
Prvních 5 řádků datového rámce
```

Out[4]:

| | id | time |
|---|---|---|
| **0** | 1 | 44 |
| **1** | 2 | 35 |
| **2** | 4 | 36 |
| **3** | 5 | 34 |
| **4** | 7 | 34 |

Dataset `data_ARSS`, vzniklý na základě výstupu z SW ARSS pro robotické zdicí práce, je uložen ve formátu CSV. Obsahuje digitální kladečský plán vytvořený ve SW ARSS.

In [5]:
```
# Soubor je načten a přiřazen do proměnné ‚df_ARSS‘
other_path = '../../data/00_Raw/data_ARSS.csv'
df_ARSS = pd.read_csv(other_path, header=0)
```

In [6]:
```
# Zobrazení prvních 5 řádků datasetu
print('Prvních 5 řádků datového rámce')
df_ARSS.head(5)
```

```
Prvních 5 řádků datového rámce
```

Out[6]:

| | ID | TYPE | ROTATION | X | Y | Z | LAYER | PALLET |
|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 2 | 90 | 220.0 | 95.0 | 0 | 1 | 1 |
| **1** | 2 | 3 | 90 | 220.0 | 252.5 | 0 | 1 | 1 |
| **2** | 3 | 1 | 90 | 220.0 | 440.0 | 0 | 1 | 1 |
| **3** | 4 | 1 | 90 | 220.0 | 690.0 | 0 | 1 | 2 |
| **4** | 5 | 1 | 90 | 220.0 | 940.0 | 0 | 1 | 3 |

Dataset `data_delay`, vzniklý na základě časosběrného monitorování robotických zdicích prací, je uložen ve formátu CSV. Obsahuje stochastické prodlevy: defekt materiálu, servis, porucha systému, nutnost otáčení zdicích prvků.

In [7]:
```
# Soubor je načten a přiřazen do proměnné ‚df_delay‘
other_path = '../../data/00_Raw/data_delay.csv'
df_delay = pd.read_csv(other_path, header=0)
```

In [8]:
```
# Zobrazení prvních 5 řádků datasetu
print('Prvních 5 řádků datového rámce')
df_delay.head(5)
```

```
Prvních 5 řádků datového rámce
```

Out[8]:

| | id | time | delay | total_time | type_delay |
|---|---|---|---|---|---|
| **0** | 3 | 36 | 109 | 145 | material |
| **1** | 6 | 34 | 62 | 96 | service |
| **2** | 20 | 40 | 200 | 240 | material |
| **3** | 31 | 52 | 198 | 250 | system |
| **4** | 35 | 42 | 24 | 66 | rotation |

## Přidání nebo změna názvů sloupců

In [9]:
```
# Tvorba názvů sloupců
headers = ['id', 'x', 'y', 'z']
print('Sloupce\n', headers)
```

```
Sloupce
 ['id', 'x', 'y', 'z']
```

In [10]:
```
# Nahrazení názvů sloupců a následná kontrola datového rámce
df_ARSS = df_ARSS[['ID', 'X', 'Y', 'Z']]
df_ARSS.columns = headers
df_ARSS.head()
```

Out[10]:

| | id | x | y | z |
|---|---|---|---|---|
| **0** | 1 | 220.0 | 95.0 | 0 |
| **1** | 2 | 220.0 | 252.5 | 0 |
| **2** | 3 | 220.0 | 440.0 | 0 |
| **3** | 4 | 220.0 | 690.0 | 0 |
| **4** | 5 | 220.0 | 940.0 | 0 |

## Sloučení datasetů dle `id`

In [11]:
```
df_merged = df_ARSS.merge(df_timelaps, on="id", how="left")
df_merged = df_merged.merge(df_delay, on="id", how="left")
df_merged = df_merged.fillna(0)
df_merged['time'] = df_merged['time_x'] + df_merged['time_y']
df_merged['total_time'] = df_merged['time'] + df_merged['delay']
df_merged = df_merged.drop(["time_x", "time_y"], axis=1)
df_merged = df_merged[df_merged['total_time']>0]
df_merged = df_merged.reset_index(drop=True)
df = df_merged[['id', 'x', 'y', 'z', 'time', 'delay', 'type_delay', 'total_time']]
df
```

Out[11]:

| | id | x | y | z | time | delay | type_delay | total_time |
|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 220.0 | 95.0 | 0 | 44.0 | 0.0 | 0 | 44.0 |
| **1** | 2 | 220.0 | 252.5 | 0 | 35.0 | 0.0 | 0 | 35.0 |
| **2** | 3 | 220.0 | 440.0 | 0 | 36.0 | 109.0 | material | 145.0 |
| **3** | 4 | 220.0 | 690.0 | 0 | 36.0 | 0.0 | 0 | 36.0 |
| **4** | 5 | 220.0 | 940.0 | 0 | 34.0 | 0.0 | 0 | 34.0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... |
| **264** | 272 | 3440.0 | 220.0 | 2250 | 37.0 | 0.0 | 0 | 37.0 |
| **265** | 273 | 3690.0 | 220.0 | 2250 | 39.0 | 0.0 | 0 | 39.0 |
| **266** | 275 | 220.0 | 752.5 | 2250 | 41.0 | 41.0 | rotation | 82.0 |
| **267** | 276 | 220.0 | 3127.5 | 2250 | 44.0 | 0.0 | 0 | 44.0 |
| **268** | 277 | 220.0 | 3315.0 | 2250 | 26.0 | 0.0 | 0 | 26.0 |

269 rows × 8 columns

## Analýza chybějících hodnot v datové sadě

In [12]:
```
# Logická hodnota ‚True‘ indikuje přítomnost chybějící hodnoty, zatímco ‚False‘ označuje její
missing_data = df.isnull()
missing_data.head(5)
```

Out[12]:

| | id | x | y | z | time | delay | type_delay | total_time |
|---|---|---|---|---|---|---|---|---|
| **0** | False | False | False | False | False | False | False | False |
| **1** | False | False | False | False | False | False | False | False |
| **2** | False | False | False | False | False | False | False | False |
| **3** | False | False | False | False | False | False | False | False |
| **4** | False | False | False | False | False | False | False | False |

In [13]:
```
# Výpočet počtu chybějících hodnot v jednotlivých sloupcích datového rámce
for column in missing_data.columns.values.tolist():
    print(f'{missing_data[column].value_counts()}\n')
```

```
id
False    269
Name: count, dtype: int64

x
False    269
Name: count, dtype: int64

y
False    269
Name: count, dtype: int64

z
False    269
Name: count, dtype: int64

time
False    269
Name: count, dtype: int64

delay
False    269
Name: count, dtype: int64

type_delay
False    269
Name: count, dtype: int64

total_time
False    269
Name: count, dtype: int64
```

## Práce s chybějícími daty

**Jak pracovat s chybějícími daty?**

1. Odstranění dat:
   a. Odstranění celého řádku
   b. Odstranění celého sloupce
2. Nahrazení dat:
   a. Nahrazení průměrnou hodnotou
   b. Nahrazení nejčastější hodnotou (frekvencí)
   c. Nahrazení na základě jiných funkcí

## Výpočty a následné úpravy dat

1. Change "type_delay" data to category values:
   a. material -> 1
   b. service -> 2
   c. rotation -> 3
   d. system -> 4

In [14]:
```
# úprava formátu dat: type_delay -> Int
df['type_delay'] = df['type_delay'].map({
    0: 0,
    'material': 1,
    'service': 2,
```

```
    'rotation': 3,
    'system': 4
})
df
```

Out[14]:

| | id | x | y | z | time | delay | type_delay | total_time |
|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 220.0 | 95.0 | 0 | 44.0 | 0.0 | 0 | 44.0 |
| **1** | 2 | 220.0 | 252.5 | 0 | 35.0 | 0.0 | 0 | 35.0 |
| **2** | 3 | 220.0 | 440.0 | 0 | 36.0 | 109.0 | 1 | 145.0 |
| **3** | 4 | 220.0 | 690.0 | 0 | 36.0 | 0.0 | 0 | 36.0 |
| **4** | 5 | 220.0 | 940.0 | 0 | 34.0 | 0.0 | 0 | 34.0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... |
| **264** | 272 | 3440.0 | 220.0 | 2250 | 37.0 | 0.0 | 0 | 37.0 |
| **265** | 273 | 3690.0 | 220.0 | 2250 | 39.0 | 0.0 | 0 | 39.0 |
| **266** | 275 | 220.0 | 752.5 | 2250 | 41.0 | 41.0 | 3 | 82.0 |
| **267** | 276 | 220.0 | 3127.5 | 2250 | 44.0 | 0.0 | 0 | 44.0 |
| **268** | 277 | 220.0 | 3315.0 | 2250 | 26.0 | 0.0 | 0 | 26.0 |

269 rows × 8 columns

## Kontrola a úprava formátu dat

In [15]:
```
# Kontrola datového typu
df.dtypes
```

Out[15]:
```
id             int64
x            float64
y            float64
z              int64
time         float64
delay        float64
type_delay     int64
total_time   float64
dtype: object
```

In [16]:
```
df = df.astype(int)
```

In [17]:
```
# Kontrola datového typu
df.dtypes
```

Out[17]:
```
id           int64
x            int64
y            int64
z            int64
time         int64
delay        int64
type_delay   int64
total_time   int64
dtype: object
```

In [18]:
```
df.head()
```

Out[18]:

| | id | x | y | z | time | delay | type_delay | total_time |
|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 220 | 95 | 0 | 44 | 0 | 0 | 44 |
| **1** | 2 | 220 | 252 | 0 | 35 | 0 | 0 | 35 |
| **2** | 3 | 220 | 440 | 0 | 36 | 109 | 1 | 145 |
| **3** | 4 | 220 | 690 | 0 | 36 | 0 | 0 | 36 |
| **4** | 5 | 220 | 940 | 0 | 34 | 0 | 0 | 34 |

In [19]:
```
df.describe()
```

Out[19]:

| | id | x | y | z | time | delay | type_delay | total_ti |
|---|---|---|---|---|---|---|---|---|
| **count** | 269.000000 | 269.000000 | 269.000000 | 269.000000 | 269.000000 | 269.000000 | 269.000000 | 269.000 |
| **mean** | 139.881041 | 1297.936803 | 1052.033457 | 1001.858736 | 36.360595 | 5.553903 | 0.245353 | 41.914 |
| **std** | 79.394072 | 1267.084840 | 1325.817766 | 707.763639 | 6.630443 | 26.319540 | 0.800668 | 27.564 |
| **min** | 1.000000 | 95.000000 | 95.000000 | 0.000000 | 22.000000 | 0.000000 | 0.000000 | 22.000 |
| **25%** | 72.000000 | 220.000000 | 220.000000 | 500.000000 | 32.000000 | 0.000000 | 0.000000 | 32.000 |
| **50%** | 139.000000 | 690.000000 | 220.000000 | 1000.000000 | 36.000000 | 0.000000 | 0.000000 | 36.000 |
| **75%** | 209.000000 | 2315.000000 | 1565.000000 | 1500.000000 | 40.000000 | 0.000000 | 0.000000 | 42.000 |
| **max** | 277.000000 | 4002.000000 | 4690.000000 | 2250.000000 | 58.000000 | 260.000000 | 4.000000 | 296.000 |

## Identifikace odlehlých hodnot

In [20]:
```
# Nastavení horní a dolní meze pro odlehlé hodnoty
low_limit = 0.05
hi_limit =  0.95

q_low = df['time'].quantile(low_limit)
q_hi  = df['time'].quantile(hi_limit)

print(f'{q_low=}, {q_hi=}')
df_out_limit = df[(df['time'] > q_hi) | (df['time'] < q_low)]
df_out_limit
```

```
q_low=np.float64(26.0), q_hi=np.float64(48.599999999999994)
```

Out[20]:

| | id | x | y | z | time | delay | type_delay | total_time |
|---|---|---|---|---|---|---|---|---|
| **26** | 31 | 3065 | 220 | 0 | 52 | 198 | 4 | 250 |
| **27** | 32 | 3315 | 220 | 0 | 58 | 0 | 0 | 58 |
| **28** | 33 | 3565 | 220 | 0 | 53 | 0 | 0 | 53 |
| **38** | 43 | 1690 | 220 | 250 | 49 | 0 | 0 | 49 |
| **43** | 48 | 2940 | 220 | 250 | 53 | 0 | 0 | 53 |
| **45** | 50 | 3440 | 220 | 250 | 50 | 0 | 0 | 50 |
| **46** | 51 | 3690 | 220 | 250 | 50 | 0 | 0 | 50 |
| **65** | 70 | 220 | 95 | 500 | 55 | 0 | 0 | 55 |
| **85** | 90 | 815 | 220 | 500 | 56 | 0 | 0 | 56 |
| **100** | 105 | 252 | 220 | 750 | 53 | 0 | 0 | 53 |
| **111** | 116 | 2940 | 220 | 750 | 52 | 0 | 0 | 52 |
| **112** | 117 | 3190 | 220 | 750 | 55 | 0 | 0 | 55 |
| **135** | 140 | 220 | 690 | 1000 | 24 | 0 | 0 | 24 |
| **136** | 141 | 220 | 940 | 1000 | 24 | 24 | 3 | 48 |
| **137** | 142 | 220 | 3190 | 1000 | 24 | 0 | 0 | 24 |
| **138** | 143 | 220 | 3440 | 1000 | 22 | 0 | 0 | 22 |
| **140** | 145 | 220 | 3940 | 1000 | 24 | 0 | 0 | 24 |
| **141** | 146 | 220 | 4190 | 1000 | 24 | 0 | 0 | 24 |
| **200** | 208 | 3565 | 220 | 1500 | 49 | 0 | 0 | 49 |
| **223** | 231 | 220 | 3127 | 1750 | 24 | 0 | 0 | 24 |
| **228** | 236 | 220 | 252 | 2000 | 54 | 0 | 0 | 54 |

## Výpočet nejistoty

In [21]:

```python
def measurement_uncertainty(df, column, delta_t=1, k=2):
    """
    Výpočet nejistoty měření z časosběrných dat.

    Parametry:
    df       : pandas DataFrame s daty
    column   : název sloupce s měřením (např. čas v s)
    delta_t  : časové rozlišení přístroje (s)
    k        : koeficient rozšíření (default k=2)

    Návratová hodnota:
    dict s výsledky
    """

    data = df[column].dropna()

    n = len(data)
    mean = data.mean()
    s = data.std(ddof=1)
```

```
    # Typ A
    u_A = s / np.sqrt(n)

    # Typ B (kvantizace)
    u_B_single = (delta_t / 2) / np.sqrt(3)
    u_B_mean = u_B_single / np.sqrt(n)

    # Kombinovaná
    u_c = np.sqrt(u_A**2 + u_B_mean**2)

    # Rozšířená
    U = k * u_c

    return {
        "serie": column,
        "n": n,
        "mean": mean,
        "std_dev": s,
        "u_A": u_A,
        "u_B_single": u_B_single,
        "u_B_mean": u_B_mean,
        "u_c": u_c,
        "U": U
    }

def report(result):
    print(f"{result['serie']}:")
    print(f"Kombinovaná nejistota: ({result['mean']:.2f} ± {result['u_c']:.2f})")
    print(f"Rozšířená nejistota: ({result['mean']:.2f} ± {result['U']:.2f}) s (k=2, 95%)")
    print(f"Nejistota Typ A (stochastická): {result['u_A']:.4f}")
    print(f"Nejistota Typ B (přístrojová / 1 měření): {result['u_B_single']:.4f}")
    print(f"Nejistota Typ B (přístrojová / průměr): {result['u_B_mean']:.4f}")
    print()
```

In [22]:
```
result = measurement_uncertainty(df, 'total_time', delta_t=1)
report(result)

result = measurement_uncertainty(df, 'time', delta_t=1)
report(result)
```

```
total_time:
Kombinovaná nejistota: (41.91 ± 1.68)
Rozšířená nejistota: (41.91 ± 3.36) s (k=2, 95%)
Nejistota Typ A (stochastická): 1.6806
Nejistota Typ B (přístrojová / 1 měření): 0.2887
Nejistota Typ B (přístrojová / průměr): 0.0176

time:
Kombinovaná nejistota: (36.36 ± 0.40)
Rozšířená nejistota: (36.36 ± 0.81) s (k=2, 95%)
Nejistota Typ A (stochastická): 0.4043
Nejistota Typ B (přístrojová / 1 měření): 0.2887
Nejistota Typ B (přístrojová / průměr): 0.0176
```

## Export datové sady do formátu CSV

In [23]:
```
df.to_csv("../../data/01_DataScience/clean_timelaps.csv", index=False)
```

### Read/Save Other Data Formats

| Data Formate | Read | Save |
|---|---|---|

| Data Formate | Read | Save |
|---|---|---|
| csv | `pd.read_csv()` | `df.to_csv()` |
| json | `pd.read_json()` | `df.to_json()` |
| excel | `pd.read_excel()` | `df.to_excel()` |
| hdf | `pd.read_hdf()` | `df.to_hdf()` |
| sql | `pd.read_sql()` | `df.to_sql()` |

## Autor / Organizace / Datum

Vjačeslav Usmanov, ČVUT v Praze, Fakulta stavební

Přehled změn

| Datum (YYYY-MM-DD) | Verze | Autor změny | Popis změny |
|---|---|---|---|
| 2026-01-18 | 1.1 | Vjačeslav Usmanov | added DS_01_Data_Cleaning.ipnyb |
| 2026-02-10 | 1.2 | Vjačeslav Usmanov | changed DS_01_Data_Cleaning.ipnyb |