

Data Science 02: Explorační analýza dat (Data Exploration)

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

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

import matplotlib as plt
from matplotlib import pyplot
```

```
In [3]: # Soubor je načten a přiřazen do proměnné ,df'
other_path = "../data/raw/clean_timelaps.csv"
df = pd.read_csv(other_path)
df.head()
```

```
Out[3]:
```

	id	type_brick	time_start	time_verif	time_dest	time_end	time_start_sec	time_verif_sec	time_dest_sec
0	1	CORNER	13:52:18	13:52:24	13:52:42	13:52:58	49938	49944	49962
1	2	HALF	13:52:58	13:53:02	13:53:18	13:53:36	49978	49982	49998
2	4	BASIC	13:56:00	13:56:06	13:56:20	13:56:36	50160	50166	50180
3	6	BASIC	13:58:10	13:58:16	13:58:30	13:58:46	50290	50296	50310
4	9	BASIC	14:00:34	14:00:42	14:00:54	14:01:18	50434	50442	50454

Základní charakteristika datové sady

Datové typy

```
In [4]: df.dtypes
```

```
Out[4]: id                int64
type_brick              object
time_start              object
time_verif              object
time_dest              object
time_end               object
time_start_sec          int64
time_verif_sec          int64
time_dest_sec           int64
time_end_sec            int64
type                    int64
start_to_verif          int64
verif_to_dest           int64
dest_to_end             int64
total_time              int64
dtype: object
```

Popis datové sady

In [5]: `df.describe()`

Out[5]:

	id	time_start_sec	time_verif_sec	time_dest_sec	time_end_sec	type	start_to_verif
count	102.000000	102.000000	102.000000	102.000000	102.000000	102.000000	102.000000
mean	68.656863	53095.647059	53103.901961	53118.941176	53137.284314	1.303922	8.254902
std	36.495353	2852.271152	2852.560429	2851.900259	2851.193613	0.793404	3.204842
min	1.000000	48679.000000	48685.000000	48703.000000	48725.000000	1.000000	4.000000
25%	39.250000	50540.250000	50548.250000	50565.250000	50581.250000	1.000000	6.000000
50%	66.500000	51696.000000	51703.000000	51716.000000	51737.000000	1.000000	8.000000
75%	100.750000	55791.000000	55798.250000	55815.000000	55829.500000	1.000000	8.000000
max	136.000000	57449.000000	57466.000000	57473.000000	57483.000000	4.000000	20.000000

Základní informace o datové sadě

In [6]: `df.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 102 entries, 0 to 101
Data columns (total 15 columns):
#   Column                Non-Null Count  Dtype
---  -
0   id                     102 non-null    int64
1   type_brick             102 non-null    object
2   time_start             102 non-null    object
3   time_verif             102 non-null    object
4   time_dest              102 non-null    object
5   time_end               102 non-null    object
6   time_start_sec         102 non-null    int64
7   time_verif_sec         102 non-null    int64
8   time_dest_sec          102 non-null    int64
9   time_end_sec           102 non-null    int64
10  type                   102 non-null    int64
11  start_to_verif         102 non-null    int64
12  verif_to_dest          102 non-null    int64
13  dest_to_end            102 non-null    int64
14  total_time             102 non-null    int64
dtypes: int64(10), object(5)
memory usage: 12.1+ KB
```

Proces standardizace dat (Data Standardization)

Proces normalizace dat (Data Normalization)

Normalizace představuje proces transformace hodnot vybraných proměnných do srovnatelného rozsahu.

Typické přístupy zahrnují standardizaci na nulovou střední hodnotu, úpravu rozptylu na jednotkovou hodnotu nebo lineární škálování do intervalu (0, 1).

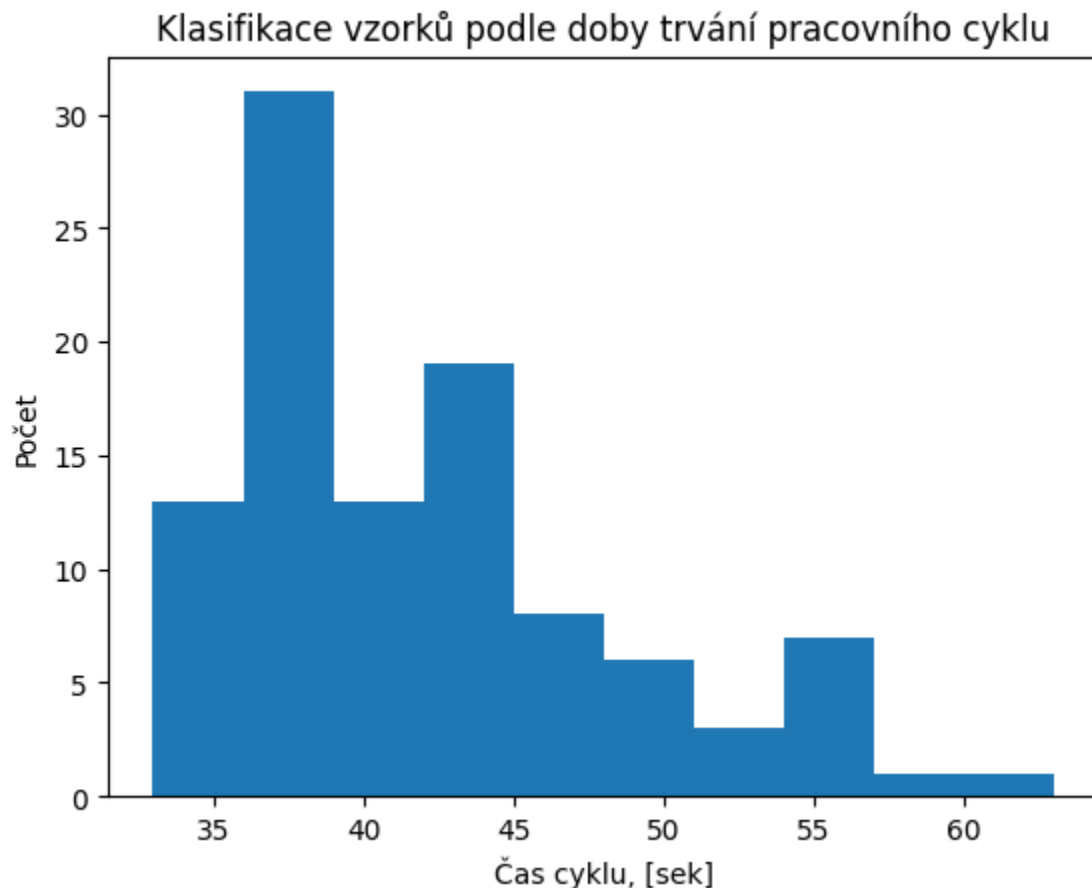
Diskretizace spojitých proměnných (binning)

In [7]: `%matplotlib inline`

```
plt.pyplot.hist(df["total_time"])

plt.pyplot.xlabel("Čas cyklu, [sek]")
plt.pyplot.ylabel("Počet")
plt.pyplot.title("Klasifikace vzorků podle doby trvání pracovního cyklu")
```

Out[7]: Text(0.5, 1.0, 'Klasifikace vzorků podle doby trvání pracovního cyklu')



In [8]: `# Definice intervalů (binů) pro diskrétní rozdělení dat`
`bins = np.linspace(min(df["total_time"]), max(df["total_time"]), 8)`
`bins`

Out[8]: array([33. , 37.28571429, 41.57142857, 45.85714286, 50.14285714,
54.42857143, 58.71428571, 63.])

In [9]: `# pojmenování intervalů (binů) pro diskrétní rozdělení dat`
`group_names = ['Extremely short', 'Very short', 'Short', 'Normal', 'Long', 'Very long', 'Extremely long']`

In [10]: `# Kategorizace intervalů`
`df['total_time_binned'] = pd.cut(df['total_time'], bins, labels=group_names, include_lowest=True)`
`df[['total_time', 'total_time_binned']].head(20)`

Out[10]:

	total_time	total_time_binned
0	40	Very short
1	38	Very short
2	36	Extremely short
3	36	Extremely short
4	44	Short
5	34	Extremely short
6	34	Extremely short
7	46	Normal
8	34	Extremely short
9	56	Very long
10	46	Normal
11	46	Normal
12	44	Short
13	44	Short
14	44	Short
15	44	Short
16	48	Normal
17	44	Short
18	46	Normal
19	52	Long

```
In [11]: # Výpočet počtu vzorků v intervalech  
df["total_time"].value_counts()
```

Out[11]: total_time

```
36    16
44    13
40    11
38    11
34     8
46     8
48     5
42     5
37     4
33     3
54     3
52     2
56     2
55     2
35     2
41     2
43     1
63     1
50     1
51     1
57     1
```

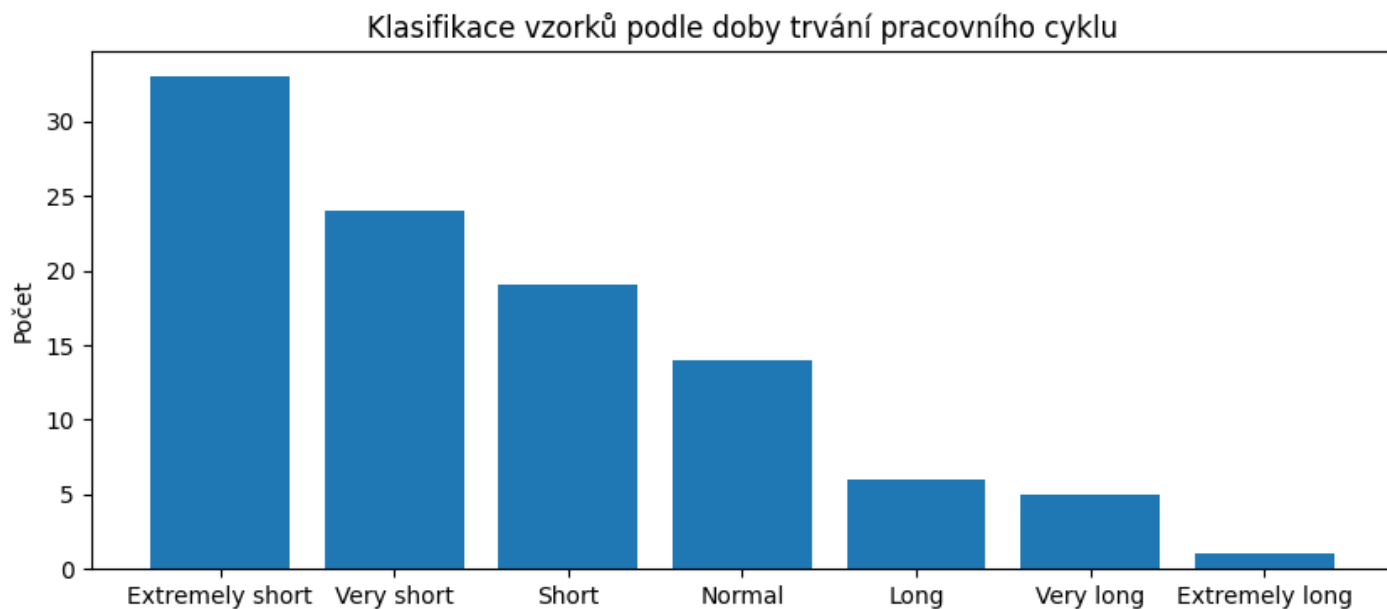
Name: count, dtype: int64

Grafické znázornění intervalového rozdělení

```
In [12]: %matplotlib inline
pyplot.figure(figsize=(10,4))
pyplot.bar(group_names, df["total_time_binned"].value_counts(), )

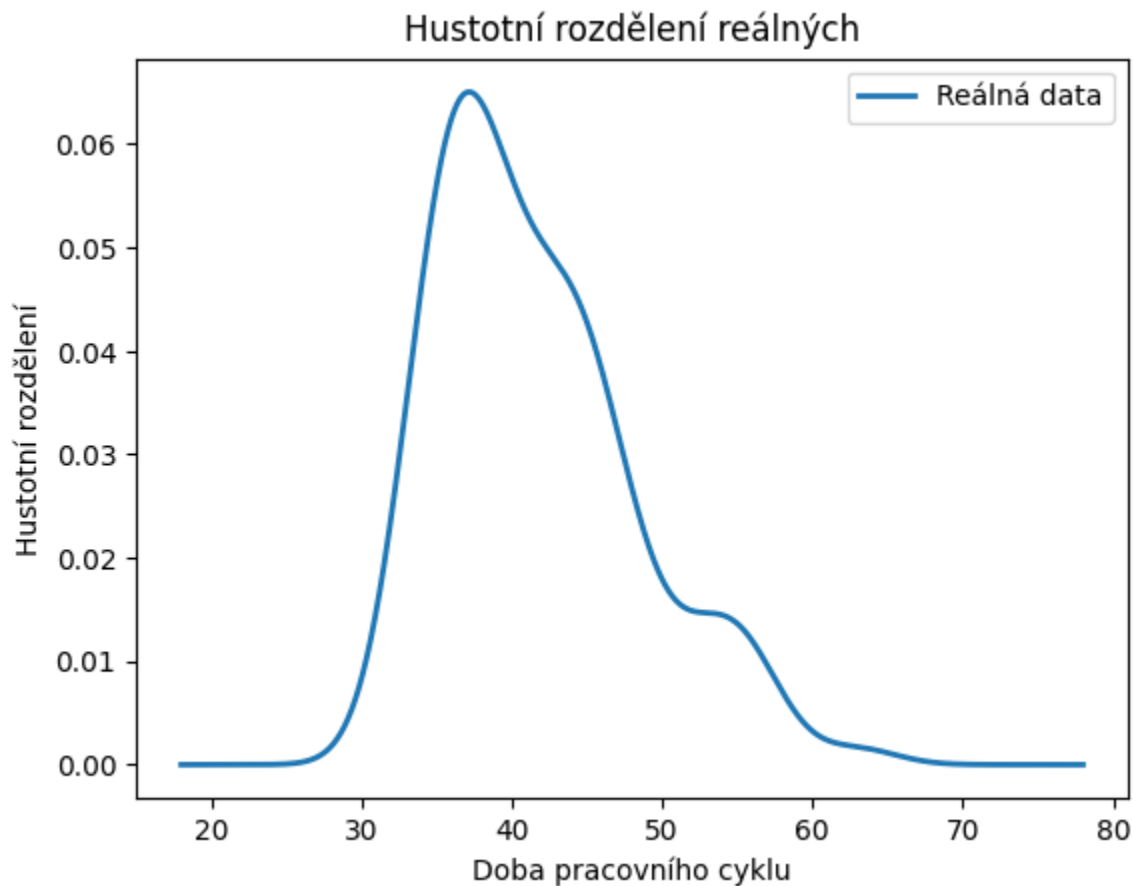
pyplot.ylabel("Počet")
pyplot.title("Klasifikace vzorků podle doby trvání pracovního cyklu")
```

Out[12]: Text(0.5, 1.0, 'Klasifikace vzorků podle doby trvání pracovního cyklu')



```
In [13]: # Plot density
df['total_time'].plot.density(bw_method='scott', linestyle='--', linewidth=2, label='Reálná data')
pyplot.legend()
pyplot.xlabel("Doba pracovního cyklu")
pyplot.ylabel("Hustotní rozdělení")
```

```
pyplot.title('Hustotní rozdělení reálných')  
pyplot.show()
```



Indikátorová (dummy) proměnná

```
In [14]: df.columns
```

```
Out[14]: Index(['id', 'type_brick', 'time_start', 'time_verif', 'time_dest', 'time_end',  
              'time_start_sec', 'time_verif_sec', 'time_dest_sec', 'time_end_sec',  
              'type', 'start_to_verif', 'verif_to_dest', 'dest_to_end', 'total_time',  
              'total_time_binned'],  
             dtype='object')
```

```
In [15]: dummy_variable_1 = pd.get_dummies(df["type_brick"])  
dummy_variable_1.head()
```

```
Out[15]:
```

	BASIC	CORNER	END	HALF
0	False	True	False	False
1	False	False	False	True
2	True	False	False	False
3	True	False	False	False
4	True	False	False	False

```
In [16]: # Změna názvů sloupců pro větší přehlednost  
dummy_variable_1.rename(columns={'BASIC': 'brick_basic', 'CORNER': 'brick_corner', 'HALF': 'brick_half'})  
dummy_variable_1.head()
```

```
Out[16]:
```

	brick_basic	brick_corner	brick_end	brick_half
0	False	True	False	False
1	False	False	False	True
2	True	False	False	False
3	True	False	False	False
4	True	False	False	False

```
In [17]: # Sloučení datových rámců "df" a "dummy_variable_1"
df = pd.concat([df, dummy_variable_1], axis=1)
df.head()
```

```
Out[17]:
```

	id	type_brick	time_start	time_verif	time_dest	time_end	time_start_sec	time_verif_sec	time_dest_sec
0	1	CORNER	13:52:18	13:52:24	13:52:42	13:52:58	49938	49944	49962
1	2	HALF	13:52:58	13:53:02	13:53:18	13:53:36	49978	49982	49998
2	4	BASIC	13:56:00	13:56:06	13:56:20	13:56:36	50160	50166	50180
3	6	BASIC	13:58:10	13:58:16	13:58:30	13:58:46	50290	50296	50310
4	9	BASIC	14:00:34	14:00:42	14:00:54	14:01:18	50434	50442	50454

Export datové sady do formátu CSV

```
In [18]: df.to_csv('../data/raw/exploration_timelaps.csv', index=False)
```

Autor / Organizace / Datum

Vjačeslav Usmanov, ČVUT v Praze, Fakulta stavební

Přehled změn

Datum (YYYY-MM-DD)	Verze	Autor změny	Popis změny
2026-01-20	1.1	Vjačeslav Usmanov	added DS_02_Exploration.ipynb
2026-02-11	1.2	Vjačeslav Usmanov	changed DS_02_Exploration.ipynb