

# Data Science 03: Inženýrství příznaků (Feature Engineering)

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

        %%pip install scipy

        %%pip install seaborn
```

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

import matplotlib.pyplot as plt
import seaborn as sns

from scipy import stats

import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)

%matplotlib inline
```

```
In [3]: # Soubor je načten a přiřazen do proměnné ,df'
path='../data/01_DataScience/exploration_timelaps.csv'
df = pd.read_csv(path)
df.head()
```

```
Out[3]:
```

	id	type_brick	time_start	time_verif	time_dest	time_end	time_start_sec	time_verif_sec	time_dest
0	1	CORNER	13:52:18	13:52:24	13:52:41	13:52:59	49938	49944	49950
1	2	HALF	13:52:59	13:53:02	13:53:19	13:53:35	49979	49982	49985
2	4	BASIC	13:56:00	13:56:06	13:56:20	13:56:37	50160	50166	50172
3	6	BASIC	13:58:10	13:58:17	13:58:31	13:58:45	50290	50297	50303
4	9	BASIC	14:00:34	14:00:42	14:00:55	14:01:19	50434	50442	50450

Analýza vzorců jednotlivých příznaků prostřednictvím grafické vizualizace

Výpočet korelace mezi proměnnými

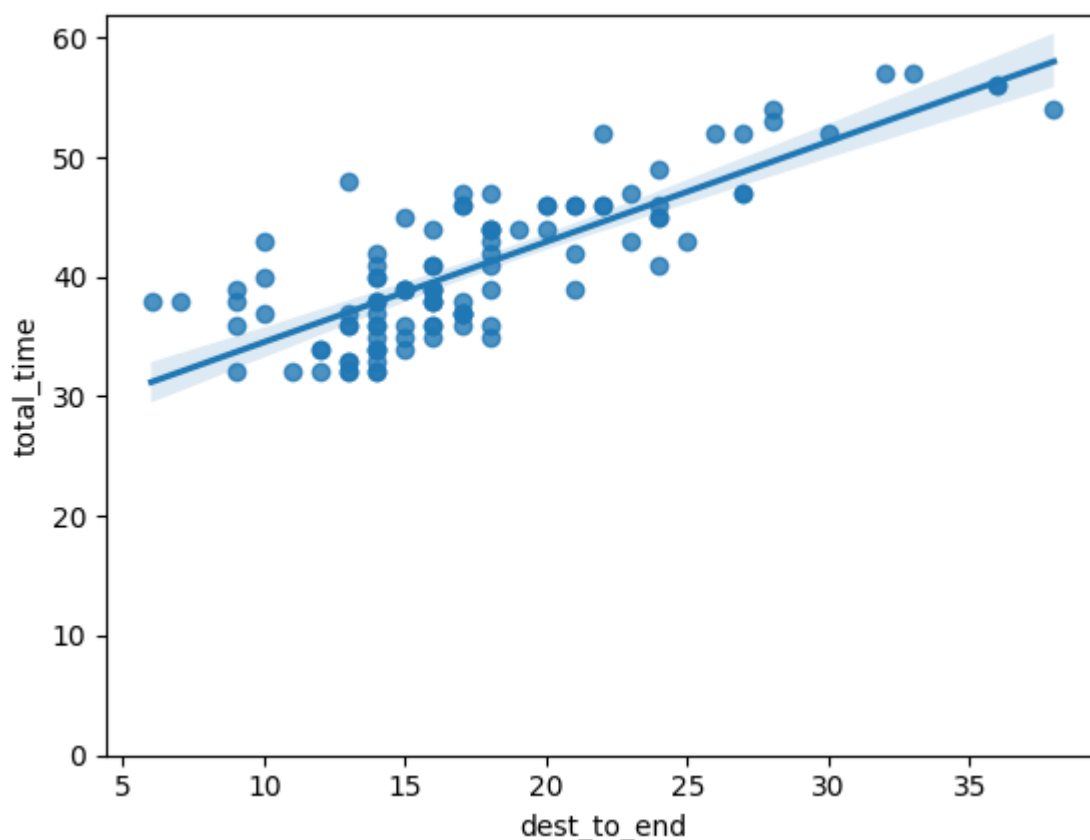
```
In [4]: df[['type', 'start_to_verif', 'verif_to_dest', 'dest_to_end', 'total_time']].corr()
```

Out[4]:

	type	start_to_verif	verif_to_dest	dest_to_end	total_time
type	1.000000	-0.007078	0.200470	-0.022691	0.082858
start_to_verif	-0.007078	1.000000	-0.378633	-0.335928	0.008529
verif_to_dest	0.200470	-0.378633	1.000000	0.044292	0.380803
dest_to_end	-0.022691	-0.335928	0.044292	1.000000	0.813031
total_time	0.082858	0.008529	0.380803	0.813031	1.000000

```
In [5]: # 'dest_to_end' jako potenciální prediktor 'total_time'
sns.regplot(x="dest_to_end", y="total_time", data=df)
plt.ylim(0,)
```

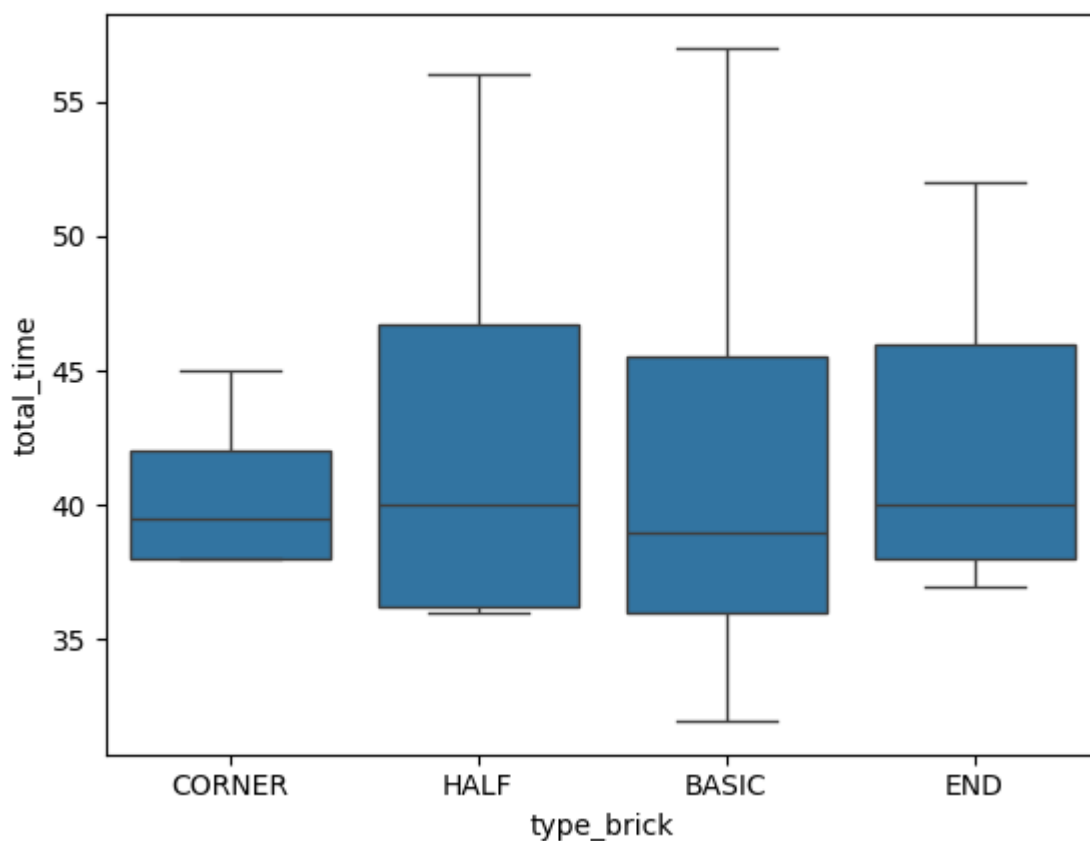
Out[5]: (0.0, 61.89465638963468)



## Analýza kategoričkých proměnných

```
In [6]: # Vztah mezi proměnnými ,type' a ,total_time'
sns.boxplot(x='type_brick', y="total_time", data=df)
```

Out[6]: &lt;Axes: xlabel='type\_brick', ylabel='total\_time'&gt;



## Deskriptivní statistická analýza dat

```
In [7]: df.describe()
```

```
Out[7]:
```

	id	time_start_sec	time_verif_sec	time_dest_sec	time_end_sec	type	start_to_veri
count	106.000000	106.000000	106.000000	106.000000	106.000000	106.000000	106.000000
mean	69.622642	53179.698113	53188.084906	53203.000000	53220.660377	1.292453	8.386790
std	37.319586	2881.268663	2881.518674	2880.72757	2879.756921	0.780317	3.432530
min	1.000000	48678.000000	48684.000000	48704.000000	48724.000000	1.000000	2.000000
25%	39.250000	50539.750000	50547.250000	50565.000000	50580.250000	1.000000	6.000000
50%	67.500000	51736.500000	51745.500000	51757.000000	51772.500000	1.000000	8.000000
75%	102.750000	55879.250000	55885.750000	55905.750000	55921.250000	1.000000	10.000000
max	136.000000	57450.000000	57467.000000	57473.000000	57482.000000	4.000000	21.000000

## Výpočet četnosti jednotlivých hodnot

```
In [8]: df['type_brick'].value_counts()
```

```
Out[8]: type_brick
BASIC    91
HALF      6
END       5
CORNER    4
Name: count, dtype: int64
```

```
In [9]: df['type_brick'].value_counts().to_frame()
```

Out[9]:

	count
type_brick	
BASIC	91
HALF	6
END	5
CORNER	4

## Seskupování dat

```
In [10]: df['type_brick'].unique()
```

Out[10]: array(['CORNER', 'HALF', 'BASIC', 'END'], dtype=object)

```
In [11]: df_group_one = df[['type_brick', 'total_time']]
df_group_one
```

Out[11]:

	type_brick	total_time
0	CORNER	41
1	HALF	36
2	BASIC	37
3	BASIC	35
4	BASIC	45
...	...	...
101	BASIC	32
102	BASIC	33
103	BASIC	33
104	BASIC	39
105	BASIC	32

106 rows × 2 columns

```
In [12]: # Výpočet průměrné hodnoty času pro jednotlivé kategorie dat
df_group_one = df_group_one.groupby(['type_brick'], as_index=False).mean()
df_group_one
```

Out[12]:

	type_brick	total_time
0	BASIC	40.780220
1	CORNER	40.500000
2	END	42.600000
3	HALF	42.666667

## Vztah mezi korelací a kauzalitou

**Korelace:** míra vzájemné závislosti mezi proměnnými.

**Kauzalita:** vztah příčiny a následku mezi dvěma proměnnými.

## Pearsonova korelace

Pearsonův korelační koeficient měří lineární závislost mezi dvěma proměnnými X a Y.

Výsledný koeficient nabývá hodnot v intervalu od -1 do 1, kde:

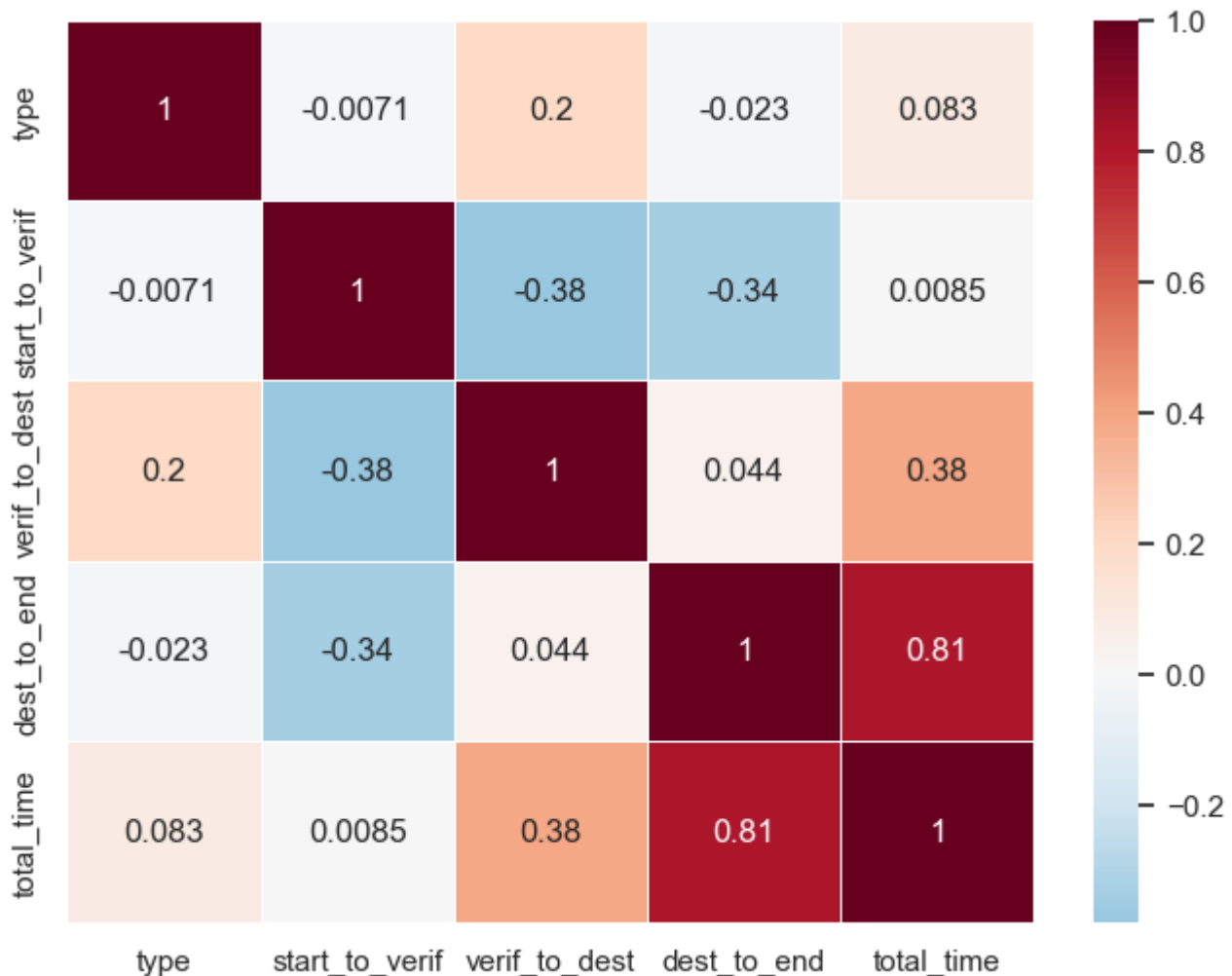
- **1:** Dokonalá kladná lineární korelace.
- **0:** Žádná lineární korelace, proměnné se pravděpodobně lineárně neovlivňují.
- **-1:** Dokonalá záporná lineární korelace.

```
In [13]: corr = df[['type', 'start_to_verif', 'verif_to_dest', 'dest_to_end', 'total_time']].corr()
```

```
In [14]: # Teplotní mapa (heatmapa)
sns.set_theme(style="white")

plt.figure(figsize=(8, 6))
sns.heatmap(
    corr,
    annot=True,
    cmap="RdBu_r",
    center=0,
    linewidths=0.5
)
```

Out[14]: <Axes: >



### P-value

P-hodnota (P-value) představuje pravděpodobnost, že korelace mezi dvěma proměnnými je statisticky významná. Obvykle se volí hladina významnosti 0,05, což znamená, že s 95% jistotou považujeme korelaci mezi proměnnými za statisticky významnou.

Podle běžně používané konvence platí, že pokud:

- p-hodnota je  $< 0,001$ : existuje silný důkaz, že korelace je statisticky významná.
- p-hodnota je  $< 0,05$ : existuje středně silný důkaz, že korelace je statisticky významná.
- p-hodnota je  $< 0,1$ : existuje slabý důkaz, že korelace je statisticky významná.
- p-hodnota je  $> 0,1$ : neexistuje důkaz o statistické významnosti korelace.

'dest\_to\_end' vs. 'total\_time'

```
In [15]: # Výpočet Pearsonova korelačního koeficientu a odpovídající p-hodnoty mezi proměnnými ,dest_to_end' a ,total_time'
pearson_coef, p_value = stats.pearsonr(df['dest_to_end'], df['total_time'])
print("Pearsonov korelační koeficient = ", pearson_coef, ". P-value =", p_value)
```

Pearsonov korelační koeficient = 0.8130307236022282 . P-value = 3.542558833149564e-26

Protože p-hodnota je  $< 0,001$ , korelace mezi proměnnými ,dest\_to\_end' a ,total\_time' je statisticky významná a lineární vztah je středně silný

## Analýza rozptylu (ANOVA) pro porovnání skupinových průměrů

**Analýza rozptylu (ANOVA)** je statistická metoda používaná k ověření, zda existují statisticky významné rozdíly mezi průměry dvou nebo více skupin. ANOVA vrací dva základní parametry:

**F-statistika (F-test):** ANOVA předpokládá, že průměry všech skupin jsou stejné, a následně vyhodnocuje, jak moc se skutečné průměry od tohoto předpokladu odchylují. Tato odchylka je vyjádřena hodnotou F-statistiky. Vyšší hodnota znamená větší rozdíl mezi skupinovými průměry.

**P-hodnota:** P-hodnota udává, jak statisticky významná je vypočtená hodnota F-statistiky.

Pokud je analyzovaná proměnná silně korelována s vysvětlovanou proměnnou, očekáváme, že ANOVA vrátí vysokou hodnotu F-statistiky a nízkou p-hodnotu.

```
In [16]: # provedeme seskupení dat podle jednotlivých kategorií
grouped_test = df[['type_brick', 'total_time']].groupby('type_brick')
grouped_test.head()
```

Out[16]:

	type_brick	total_time
0	CORNER	41
1	HALF	36
2	BASIC	37
3	BASIC	35
4	BASIC	45
5	BASIC	34
6	BASIC	34
11	END	46
22	HALF	36
23	CORNER	38
24	HALF	43
38	END	38
54	CORNER	45
55	HALF	56
65	END	52
79	HALF	37
80	CORNER	38
96	END	40
97	END	37

	type_brick	total_time
0	CORNER	41
1	HALF	36
2	BASIC	37
3	BASIC	35
4	BASIC	45
5	BASIC	34
6	BASIC	34
11	END	46
22	HALF	36
23	CORNER	38
24	HALF	43
38	END	38
54	CORNER	45
55	HALF	56
65	END	52
79	HALF	37
80	CORNER	38
96	END	40
97	END	37

In [17]: `grouped_test.get_group('BASIC')['total_time']`

Out[17]:

2	37
3	35
4	45
5	34
6	34
	..
101	32
102	33
103	33
104	39
105	32

Name: total\_time, Length: 91, dtype: int64

## BASIC a HALF

In [18]: `f_val, p_val = stats.f_oneway(grouped_test.get_group('BASIC')['total_time'], grouped_test.get_group('HALF')['total_time'])`  
`print( "ANOVA results: F=", f_val, ", P =", p_val )`

ANOVA results: F= 0.47350263550583593 , P = 0.49305701471235736

Výsledky analýzy ANOVA pro kategorie BASIC a HALF vykazují p-hodnotu vyšší než 0,1, což znamená, že F-statistika není statisticky významná. Nelze tedy zamítnout nulovou hypotézu o shodě průměrů obou

## Závěr: Identifikace významných proměnných

- total\_time
- start\_to\_verif
- verif\_to\_dest
- dest\_to\_end

## Export datové sady do formátu CSV

```
In [20]: df_ready.to_csv('../data/01_DataScience/ready_timelaps.csv', index=False)
```

Autor / Organizace / Datum

## Přehled změn

Datum (YYYY-MM-DD)	Verze	Autor změny	Popis změny
2026-01-21	1.1	Vjačeslav Usmanov	added DS_03_Features.ipynb
2026-02-12	1.2	Vjačeslav Usmanov	changed DS_03_Features.ipynb