

NI-VSM – 2.domácí úkol

Eliška Krátká (kratkeli), Ondřej Wrzecionko (wrzecond), Eliáš El Frem (elfreeli)

Obsah

1	Úvod	2
1.1	Zadání úkolu	2
1.2	Parametry úlohy	2
2	Postup řešení	3
2.1	Charakteristiky délek slov	3
2.2	Výskyt znaků v textech	3
2.3	Testování hypotéz	4
2.3.1	Hypotéza – rozdělení délek slov nezávisí na textu	4
2.3.2	Hypotéza – střední délky slov v obou textech se rovnají	5
2.3.3	Hypotéza – rozdělení písmen nezávisí na textu	6
3	Výsledky	7
3.1	Rozdělení délek slov	7
3.2	Odhad pravděpodobností znaků	8
3.2.1	Hypotéza – rozdělení délek slov nezávisí na textu	8
3.2.2	Hypotéza – střední délky slov v obou textech se rovnají	8
3.2.3	Hypotéza – rozdělení písmen nezávisí na textu	9
4	Závěr	10
	Reference	11

1 Úvod

V práci se zabýváme řešením druhého domácího úkolu z předmětu NI-VSM, který se týká testování hypotéz. Používáme stejné dva texty jako v předchozím úkolu, konkrétně datové soubory *005.txt* (první text) a *004.txt* (druhý text). Cílem práce je provést analýzu zmíněných souborů a pro každý text zvlášť vyhodnotit základní charakteristiky délek slov, jako jsou střední hodnota a rozptyl. Hlavním záměrem práce je otestovat několik hypotéz týkajících se délek slov a písmen vyskytujících se v textu. Pro implementaci řešení jsme využili programovací jazyk Python.

1.1 Zadání úkolu

1. (1b) Z obou datových souborů načtete texty k analýze. Pro každý text zvlášť odhadněte základní charakteristiky délek slov, tj. střední hodnotu a rozptyl. Graficky znázorníte rozdělení délek slov.
2. (1b) Pro každý text zvlášť odhadněte pravděpodobnosti písmen (symbolů mimo mezery), které se v textech vyskytují. Výsledné pravděpodobnosti graficky znázorníte.
3. (1.5b) Na hladině významnosti 5% otestujte hypotézu, že rozdělení délek slov nezávisí na tom, o který jde text. Určete také p-hodnotu testu.
4. (1.5b) Na hladině významnosti 5% otestujte hypotézu, že se střední délky slov v obou textech rovnají. Určete také p-hodnotu testu.
5. (1b) Na hladině významnosti 5% otestujte hypotézu, že rozdělení písmen nezávisí na tom, o který jde text. Určete také p-hodnotu testu [1].

1.2 Parametry úlohy

Reprezentantem skupiny je **Eliáš El Frem**. Parametry jsme vypočítali dle vzorce ze zadání úkolu [2]:

$$X = ((K \cdot L \cdot 23) \bmod 20) + 1,$$
$$Y = ((X + ((K \cdot 5 + L \cdot 7) \bmod 19)) \bmod 20) + 1,$$

kde K je den narození reprezentanta skupiny a L počet písmen v příjmení reprezentanta. Pro úlohu jsme na základě výpočtu parametrů použili datové soubory *005.txt* (první text) a *004.txt* (druhý text).

```
1 #!/usr/bin/env python3
2
3 K = 17
4 L = len("Frem")
5 fname1 = ((K*L*23) % (20)) + 1
6 fname2 = ((fname1 + ((K*5 + L*7) % (19))) % (20)) + 1
```

2 Postup řešení

V této části se zaměřujeme na vysvětlení postupu řešení domácího úkolu včetně klíčových částí zdrojového kódu, který je implementován v programovacím jazyce Python.

2.1 Charakteristiky délek slov

První náhodná veličina, kterou jsme zkoumali, byla délka slov. Z datových souborů jsme si načetli oba texty a pro každý text odhadli střední hodnotu a rozptyl délek slov. Použili jsme bodový odhad střední hodnoty (výběrový průměr) a bodový odhad rozptylu (výběrový rozptyl), jelikož se jedná o nestranné a konzistentní odhady. Mějme náhodný výběr rozsahu n příslušný náhodné veličině X . Výběrový průměr odpovídá

$$\bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i,$$

a výběrový rozptyl

$$s_n^2 = s_X^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X}_n)^2 \quad [3].$$

```
1  #!/usr/bin/env python3
2
3  file1 = open(f'hw1-source/00{fname1}.txt', 'r')
4  file2 = open(f'hw1-source/00{fname2}.txt', 'r')
5
6  file1 = file1.read()
7  file1_words = file1.split()
8  file2 = file2.read()
9  file2_words = file2.split()
10
11 # vycet deleky jednotlivych slov
12 wordlens_file1 = [(lambda x: len(x))(x) for x in file1_words]
13 wordlens_file2 = [(lambda x: len(x))(x) for x in file2_words]
14
15 # vyberovy prumer
16 mean1 = np.mean(wordlens_file1)
17 mean2 = np.mean(wordlens_file2)
18
19 # vyberovy rozptyl
20 var1 = np.var(wordlens_file1, ddof=1)
21 var2 = np.var(wordlens_file2, ddof=1)
22
23 # odhad pravdepodobnosti
24 word_probs1 = list(np.unique(wordlens_file1, return_counts=True)[1]/len(
    wordlens_file1))
25 word_probs2 = list(np.unique(wordlens_file2, return_counts=True)[1]/len(
    wordlens_file2))
26 barcount = max(len(word_probs1), len(word_probs2))
27 word_probs1 = np.pad(word_probs1, (0, barcount - len(word_probs1)), 'constant',
    constant_values=(0, 0))
28 word_probs2 = np.pad(word_probs2, (0, barcount - len(word_probs2)), 'constant',
    constant_values=(0, 0))
```

2.2 Výskyt znaků v textech

Zvlášť pro každý text jsme spočítali četnost znaků vyskytujících se v textu. Na rozdíl od prvního domácího úkolu jsme tentokrát mezi znaky zahrnuli pouze písmena a nepočítali mezery a nové řádky. Symbol čárky jsme se rozhodli ponechat, jelikož zadání nespecifikuje jinak. Pravděpodobnost výskytu znaku $P(X)$ odpovídá

$$P(X) = \frac{\text{četnost znaku}}{\text{celkový počet znaků}},$$

kde X je náhodná veličina – počet výskytů daného znaku v textu.

```

1 #!/usr/bin/env python3
2
3 # vypocet cetnosti
4 def countFreqs(to_cnt):
5     char_cnt = {}
6     for i in to_cnt.lower():
7         if not char_cnt.get(i):
8             char_cnt[i] = 1
9         else:
10            char_cnt[i] += 1
11     return char_cnt
12
13 # odhad pravdepodobnosti
14 def countProbs(to_cnt):
15     to_ret = {}
16     char_cnt = countFreqs(to_cnt)
17     for i in char_cnt.keys():
18         to_ret[i] = char_cnt[i]/len(to_cnt)
19     return to_ret
20
21 f1_probs = dict(sorted(countProbs("".join(file1.split())).items()))
22 f2_probs = dict(sorted(countProbs("".join(file2.split())).items()))

```

2.3 Testování hypotéz

V následující části se věnujeme testování hypotéz. Nulová hypotéza H_0 značí tvrzení, o kterém chceme rozhodovat. Proti H_0 vždy stavíme opačné tvrzení H_A , které se nazývá alternativní hypotéza. Předpokládáme, že buď platí H_0 , nebo H_A . Test hypotézy H_0 proti H_A je rozhodovací proces, na základě kterého buď zamítneme nebo potvrdíme H_0 . Pokud zamítneme hypotézu H_0 , znamená to, že H_A je pro nás statisticky nevýznamná. Proto si jako H_A volíme hypotézu, kterou chceme dokázat. Hypotézy testujeme na hladině významnosti α . To znamená, že pravděpodobnost toho, že zamítneme H_0 , ačkoli platí, je nejvýše rovna $1 - \alpha$.

P-hodnota je horní mez pravděpodobnosti, s jakou bychom na základě naměřených dat mohli zamítnout nulovou hypotézu, pokud by platila. Pokud je p-hodnota nižší než zvolená hladina významnosti α , zamítáme nulovou hypotézu a přijímáme alternativní hypotézu. P-hodnota tedy poskytuje informaci o významnosti výsledků testu a umožňuje nám rozhodnout, zda máme dostatečné důkazy pro zamítnutí nulové hypotézy na zvolené hladině významnosti [4].

2.3.1 Hypotéza – rozdělení délek slov nezávisí na textu

Označme náhodnou veličinu X délky slov v prvním textu a náhodnou veličinu Y délky slov v druhém textu. Jako nulovou hypotézu H_0 volíme tvrzení, že veličiny jsou nezávislé, proti alternativě H_A , že jsou závislé. Pokud bychom tedy na základě testu na hladině α zamítli H_0 ve prospěch H_A , znamená to, že si jsme na $1 - \alpha$ procent jisti, že veličiny jsou závislé. Nulovou hypotézu a alternativní hypotézu můžeme vyjádřit jako

$$\begin{aligned}
 H_0 &: p_{ij} = p_{i\bullet} p_{\bullet j}, \\
 H_A &: p_{ij} \neq p_{i\bullet} p_{\bullet j},
 \end{aligned}$$

kde

$$p_{ij} = P(X = i, Y = j), \quad p_{i\bullet} = \sum_j p_{ij}, \quad p_{\bullet j} = \sum_i p_{ij}.$$

jsou sdružené a marginální pravděpodobnosti. Testová statistika χ^2 odpovídá

$$\chi^2 = n \sum_{i=1}^r \sum_{j=1}^c \frac{N_{ij}^2}{N_{i\bullet} N_{\bullet j}} - n,$$

kde N označuje kontingenční tabulku. Počet stupňů volnosti vyjadřuje míru nezávislosti mezi proměnnými a je roven $(\# \text{ řádků} - 1) \cdot (\# \text{ sloupců} - 1)$ [4].

Provedli jsme test nezávislosti X a Y v kontingenční tabulce. Řádky tabulky odpovídají jednotlivým textům, sloupce tabulky jsou četnosti veličin X a Y . Na základě tabulky jsou spočítali

teoretické četnosti a sloučili jsme sloupce, ve kterých byly theoretické četnosti menší než pět. Pro výpočet testové statistiky χ^2 , stupňů volnosti a p-hodnoty jsme využili knihovní funkci `stats.chi2_contingency`.

Zda H_0 zamítneme nebo nezamítneme, jsme vyhodnotili na základě kritického oboru

$$\chi^2 \geq \chi_{\alpha, (r-1)(c-1)}^2,$$

kde $(r-1)(c-1)$ je počet stupňů volnosti.

```
1 #!/usr/bin/env python3
2
3 #vytvoreni kontingencni tabulky
4 wordlens_arr_file1 = np.unique(wordlens_file1, return_counts=True)[1]
5 wordlens_arr_file2 = np.unique(wordlens_file2, return_counts=True)[1]
6 cont_tab = np.zeros((2, max(wordlens_arr_file1.size, wordlens_arr_file2.size)))
7 np.put(cont_tab[0], np.arange(wordlens_arr_file1.size), wordlens_arr_file1)
8 np.put(cont_tab[1], np.arange(wordlens_arr_file2.size), wordlens_arr_file2)
9
10 #vypocet theoretickych cetnosti
11 theoretic_occur = Table(copy.deepcopy(cont_tab)).fittedvalues
12
13 #sloucení sloupce, kde jsou theoreticke cetnosti mensi nez 5
14 cols_to_merge = max(sum(theoretic_occur[1] < 5), sum(theoretic_occur[0] < 5))
15 cont_tab[0][-cols_to_merge:] = np.sum(cont_tab[0, -cols_to_merge:], axis=-1)
16 cont_tab[1][-cols_to_merge:] = np.sum(cont_tab[1, -cols_to_merge:], axis=-1)
17 cont_tab = cont_tab[:, :-cols_to_merge+1]
18
19 #vypocet chi^2, p-hodnoty a stupnu volnosti pomoci knihovni funkce
20 Chi2, p, df, _ = stats.chi2_contingency(cont_tab, correction=False)
```

2.3.2 Hypotéza – střední délky slov v obou textech se rovnají

Označme náhodnou veličinu X délky slov v prvním textu a náhodnou veličinu Y délky slov v druhém textu. Jako nulovou hypotézu H_0 volíme tvrzení, že se střední délky slov rovnají proti alternativě H_A , že se nerovnají.

$$H_0 : \mu_1 = \mu_2,$$

$$H_A : \mu_1 \neq \mu_2,$$

kde μ_1, μ_2 jsou střední hodnoty délek slov.

Při zamítnutí hypotézy H_0 na základě testu na hladině α procent jsem si na α procent jisti, že se střední délky nerovnají. Jelikož jednotlivé délky slov na sobě nejsou navzájem závislé, nemůžeme použít párový t-test a musíme zvolit dvouvýběrový t-test. Existují dvě varianty dvouvýběrového t-testu, které se liší podle toho, zda jsou rozptyly veličin stejné nebo různé.

Zvolili jsme Levenův test (není tolik citlivý na normalitu dat jako F-test) a zjistili jsme, zda se rozptyly rovnají nebo ne (opět testování hypotéz, $H_0 : \sigma_1^2 = \sigma_2^2$, $H_A : \sigma_1^2 \neq \sigma_2^2$). Využili jsme knihovní funkci `stats.levene` a na základě nízké p-hodnoty Levenova testu jsme došli k závěru, že se rozptyly nerovnají. Proto jsme ve funkci `stats.ttest_ind` (provedení dvouvýběrového t-testu) nastavili hodnotu parametru `equal_var` na `False`.

Zda H_0 zamítneme nebo nezamítneme, jsme vyhodnotili na základě kritického oboru

$$|T| \geq t_{\alpha/2, n_d},$$

kde T je testová statistika [4].

```
1 #!/usr/bin/env python3
2
3 #vypocet p-hodnoty Levenova testu
4 stat, p = stats.levene(wordlens_file1, wordlens_file2)
5
6 #rozptyly se nerovnaji - dvouvyberovy t-test
7 tt_s, tt_p = stats.ttest_ind(wordlens_file1, wordlens_file2, alternative='two-
8     sided', equal_var=False)
9
10 #kriticky obor
11 n = len(wordlens_file1)
```


```

11 m = len(wordlens_file2)
12 s_x_2 = var1
13 s_y_2 = var2
14 s_d_2 = s_x_2/n+s_y_2/m
15 n_d = (s_d_2**2) / ((1/(n-1))*((s_x_2/n)**2) + (1/(m-1))*((s_y_2/m)**2))
16 t = stats.t.isf(0.05/2,n_d)

```

2.3.3 Hypotéza – rozdělení písmen nezávisí na textu

Postupovali jsme obdobně jako v 2.3.1, s tím rozdílem, že tentokrát místo délek slov porovnáváme rozdělení písmen v textech.

Označme náhodnou veličinu X rozdělení písmen v prvním textu a náhodnou veličinu Y rozdělení písmen v druhém textu. Jako nulovou hypotézu H_0 volíme tvrzení, že veličiny jsou nezávislé, proti alternativě H_A , že jsou závislé. 

```

1 #!/usr/bin/env python3
2
3 #vytvoreni kontingencni tabulky
4 letter_freqs_arr_file1 = np.fromiter(dict(sorted(countFreqs("".join(file1.split(
5 letter_freqs_arr_file2 = np.fromiter(dict(sorted(countFreqs("".join(file2.split(
6 cont_tab = np.zeros((2,max(letter_freqs_arr_file1.size,letter_freqs_arr_file2.
7 np.put(cont_tab[0],np.arange(letter_freqs_arr_file1.size),letter_freqs_arr_file1)
8 np.put(cont_tab[1],np.arange(letter_freqs_arr_file2.size),letter_freqs_arr_file2)
9
10 #vypocet teoretickych cetnosti
11 theoretic_occur = Table(copy.deepcopy(cont_tab)).fittedvalues
12
13 #sloucení sloupce, kde jsou teoretické četnosti menší než 5
14 cont_tab[:,cont_tab.shape[1]-1] += cont_tab[:,0]
15 cont_tab = cont_tab[:,1:]
16
17 #vypocet chi^2, p-hodnoty a stupně volnosti pomocí knihovny funkce
18 Chi2, p, df, _ = stats.chi2_contingency(cont_tab, correction=False)

```

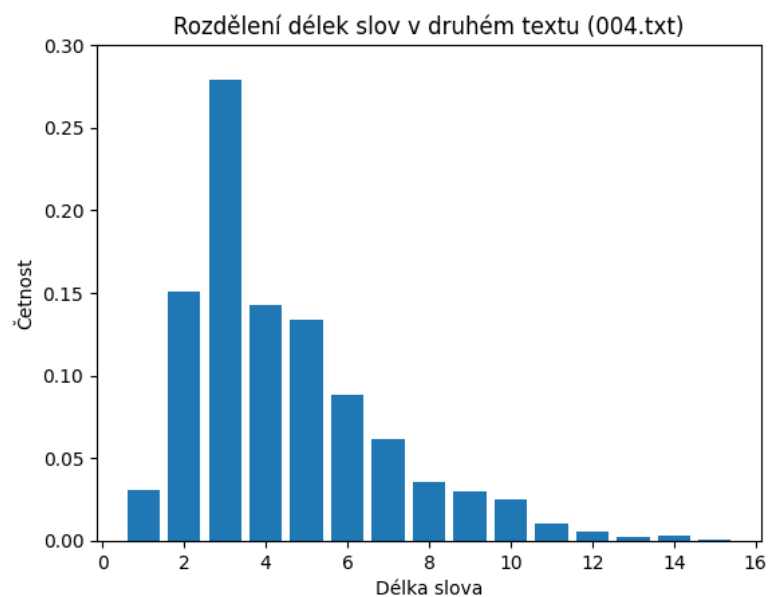
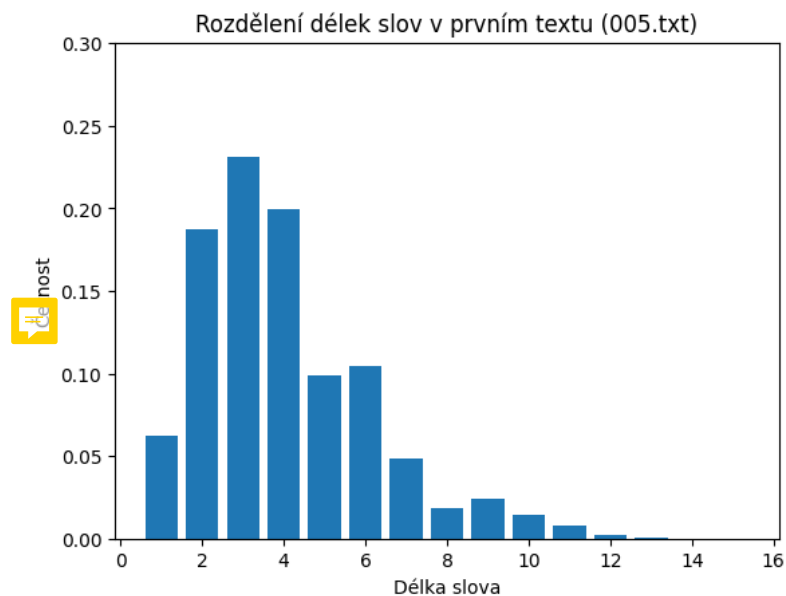
3 Výsledky

V domácím úkolu jsme analyzovali dva anglické texty. Datový soubor *005.txt* obsahuje první text a *004.txt* druhý text. V tabulkách uvádíme výsledky zaokrouhleny na tři desetinná místa.

3.1 Rozdělení délek slov

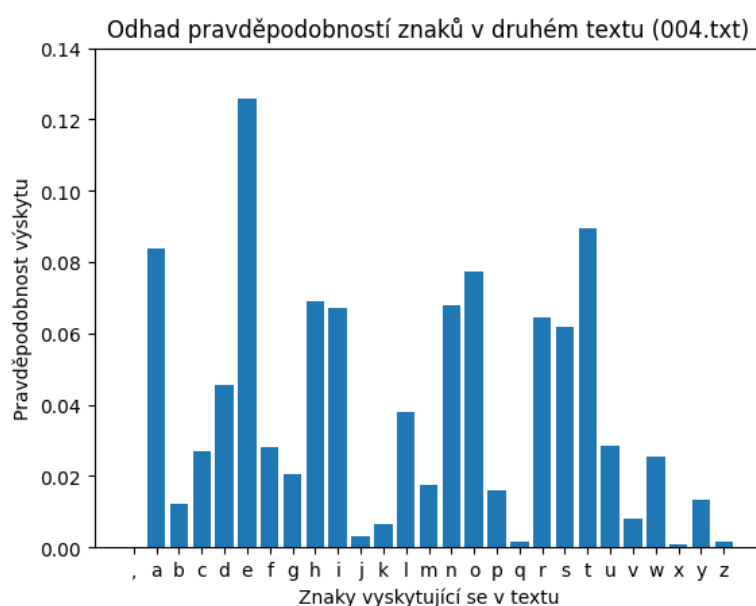
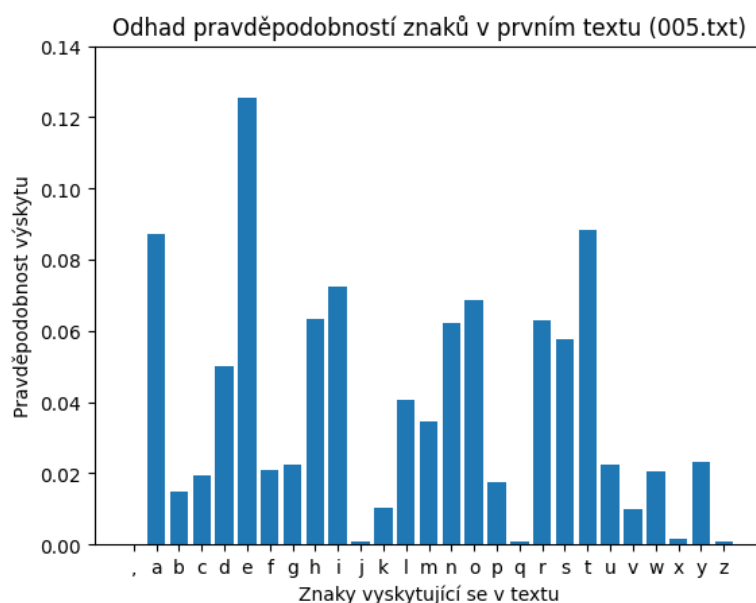
Texty se příliš neliší ve střední hodnotě délky slov, ale rozptyl pro druhý text je vyšší. To značí, že délky slov se v tomto textu více různí nebo že jsou hodnoty více vzdálené od průměru.

Text	Střední hodnota	Rozptyl
první (<i>005.txt</i>)	4.023	4.526
druhý (<i>004.txt</i>)	4.447	5.795



3.2 Odhad pravděpodobností znaků

Z grafů odhadnutých pravděpodobností znaků pozorujeme, že v obou textech se nejčastěji vyskytují znaky *e*, *t*, a *a*, která jsou nejčtenějšími písmeny v běžném anglickém textu [5].



3.2.1 Hypotéza – rozdělení délek slov nezávisí na textu

Jelikož $\chi^2 \geq \chi_{0.05,11}^2$, tedy $54.160 \geq 19.675$, hypotézu H_0 zamítáme ve prospěch H_A [6]. P-hodnota je velmi malá, proto můžeme s téměř jistotou prohlásit, že rozdělení délek slov závisí na konkrétním textu.

α	χ^2	počet stupňů volnosti	p-hodnota	$\chi_{0.05,11}^2$
0.05	54.160	11	1.103e-07	19.675

3.2.2 Hypotéza – střední délky slov v obou textech se rovnají

Nacházíme se v kritickém oboru, jelikož $|T| \geq t_{\alpha/2, n_d}$, $4.293 \geq 1.961$, což znamená že zamítáme nulovou hypotézu H_0 ve prospěch alternativní hypotézy H_A . Střední délky slov v obou textech

se nerovnejí.

α	p-hodnota Levenova testu	T	p-hodnota	$t_{\alpha/2, n_d}$
0.05	0.0000	-4.293	1.848e-05	1.961

3.2.3 Hypotéza – rozdělení písmen nezávisí na textu

Protože hodnota $\chi^2 \geq \chi_{0.05,25}^2$, tedy $78.886 \geq 37.652$, zamítáme nulovou hypotézu H_0 ve prospěch alternativní hypotézy H_A [6]. P-hodnota je velmi nízká, což nám umožňuje s téměř jistotou tvrdit, že rozdělení písmen závisí na konkrétním textu.

α	χ^2	počet stupňů volnosti	p-hodnota	$\chi^2_{0.05,25}$
0.05	78.886	25	1.704e-07	37.652

4 Závěr

Zanalyzovali jsme dva datové soubory, text **005.txt** a **004.txt**. Pro oba texty jsme našli střední hodnotu a rozptyl délek slov a odhad pravděpodobností vyskytujících se písmen. Otestovali jsme tři hypotézy, na základě kterých jsme zjistili, že rozdělení délek slov a jednotlivých písmen závisí na textu a dále že střední délky slov v textech se nerovnají.



Reference

- [1] P. Hrabák. Domácí úkol 2. <https://courses.fit.cvut.cz/MI-SPI/homework/hw2/index.html>.
- [2] P. Hrabák. Domácí úkoly. <https://courses.fit.cvut.cz/MI-SPI/homework/index.html>.
- [3] P. Hrabák, P. Novák, D.Vašata. Statistika. <https://courses.fit.cvut.cz/MI-SPI/lectures/files/NI-VSM-Lec-09-Handout.pdf>.
- [4] P. Hrabák, P. Novák, D.Vašata. Testování hypotéz. <https://courses.fit.cvut.cz/MI-SPI/lectures/files/NI-VSM-Lec-10-Handout.pdf>.
- [5] R. Lórencz. Základní pojmy v kryptologii, substituční šifry, blokové, transpoziční šifry. <https://courses.fit.cvut.cz/BI-BEZ/media/lectures/bez1.pdf>.
- [6] NI-VSM. Critical values of the chi-square distribution. <https://courses.fit.cvut.cz/MI-SPI/tutorials/files/tables/tables.pdf>.