CSI – Statistické zpracování dat:

Analýza meteorologických dat na území ČR

Bc. Václav Pastušek (204437)

27. 4. 2023

MPC-TIT

1. Úvod

Tento skript slouží k analýze meteorologických dat v České republice. Jeho hlavním cílem je umožnit uživatelům získávat a zpracovávat data o teplotách a srážkách a následně je vizualizovat v grafech. Program nabízí možnost pracovat v režimu online nebo offline, s možností použití záložních online dat.

2. Návod k instalaci

Požadavky na systém a knihovny:

- Python verze 3.9 nebo novější
    - Pro snížení verze na 3.8 je potřeba odstranit Union operátory „|“ a na 3.7 Walrus operátory „:=“
- Standardní knihovny (není potřeba instalovat):
    - platform – Přístup k identifikačním údajům základní platformy
    - os – Rozhraní pro různé operační systémy
    - typing (Union, Callable, Optional, Generator) – Podpora pro typové nápovědy
    - functools – Funkce pro práci s volatelnými objekty
    - itertools – Funkce pro vytváření iterátorů pro efektivní opakování
    - signal – Nastavení obsluhy asynchronních událostí
    - re – Operace s regulárními výrazy
    - time – Přístup a konverze času
    - concurrent.futures – Spouštění paralelních úloh
- Knihovny třetích stran:
    - requests – Knihovna pro HTTP požadavky
    - bs4 (BeautifulSoup) – Knihovna pro parsování HTML a XML dokumentů
    - xarray – Knihovna pro práci s označenými vícerozměrnými poli
    - netcdf4 – umí číst a zapisovat soubory netCDF
    - numpy – Základní balík pro vědecké výpočty
    - scipy.stats (skew, kurtosis, hmean, gmean, linregress) – Knihovna pro statistiku a regresní analýzu
    - scipy.fftpack (dct) – Knihovna pro diskrétní kosinovou transformaci
    - tqdm – Knihovna pro zobrazování průběhu úloh
    - matplotlib.pyplot – Knihovna pro tvorbu statických, animovaných a interaktivních vizualizací
    - matplotlib – Knihovna pro úpravu grafů vytvořených knihovnou pyplot
    - matplotlib.colors – Knihovna pro mapování a normalizaci barev
    - keyboard – Knihovna pro detekci klávesnice

Postup instalace a spuštění programu:

1. Nainstalujte nejnovější verzi Pythonu z oficiálních stránek: https://www.python.org/downloads/
2. Stáhněte a uložte skript main.py na počítač
3. Nainstalujte potřebné knihovny pomocí příkazu – možnosti:
    a. *„pip install knihovna*" v příkazovém řádku pro každou knihovnu
    b. *„python install_requirements.py*“ v příkazové řádce
4. Spusťte skript pomocí příkazu "*python main.py*" v příkazovém řádku

Pokud nemáte nainstalovaný PIP, napište do terminálu: „*python -m ensurepip --default-pip*“.

Pro update PIP, napište do terminálu: „*python -m pip install --upgrade pip*“.

V případě, že máte starší verzi Pythonu, tak může pomoct volat skript v teminálu s „*python3*“.

Pro zpuštění skriptu musíte být ve stejné složce, jako je *main.py* nebo *install_requirements.py*.

Další možnost je nainstalovat zdarma verzi Pycharm Community na této stránce: https://www.jetbrains.com/pycharm/ a po otevření skriptu main.py, IDE automaticky nabízí dodatečné stáhnutí knihoven.

3. Popis implementace

Skript je rozdělen do několika tříd a main části, které slouží k načítání/ukládání, zpracování a zobrazení meteorologických dat.

Třídy:

- JumpException(Exception): Tato výjimka se vyvolá, když nastane určitá chyba při stahování dat ze stránek, poté je chycena v hlavní smyčce v main a pokračuje se interakce s uživatelem.
- Utils: obsahuje pomocné funkce pro výpis přivítání, nápovědy, zachycení terminace programu a debug, který obalí určité funkce a při zapnutí globální proměnné **DEBUG_PRINT** vypíše název dané funkce, která se vykonala.
- UserInterface: Obsahuje funkce pro interakci s uživatelem. Tato třída umožňuje uživateli komunikovat s programem, například vybírat datum, kraj a typ grafu v interaktivní smyčce. Pro zjednodušení výběru dat jsou použity regulární výrazy. Kdykoliv při uživatelském vstupu je možnost vypsat nápověda, vrátit se nebo ukončit program.
- DataFetcher: obsahuje funkce pro získání meteorologických dat z internetových zdrojů nebo ze zálohy ze složky *backup* a zpracovává tato data do vhodného formátu. K tomuto účelu bylo využito 3D pole xarray. DataFetcher také nabízí možnost přemazat starou zálohu nebo pokračovat v práci pouze s aktuálními daty z internetu. Pro zvýšení efektivity může DataFetcher data stahovat jak sériově, tak paralelně (celkem se jedná o 122 nebo 244 stránek).
- GraphPlotter: zajišťuje tvorbu a zobrazování grafů vždy pro teploty a srážky. Uživatel má na výběr 10 různých grafů jako jsou 2D, 3D vykreslení grafů s různými parametry (tyto grafy ukazují různá statistická fakta), DCT histogramy ve 2D i 3D, boxploty a korelace mezi teplotou a srážkami a nebo mezi stejnými meteorologickými jevy pro jiné kraje. Všechny grafy se dají uložit klávesou „s“, ukončit klávesou „x“. Pokud obsahují alfa kanál, tak jej lze měnit i kolečkem a u 3D jde točit graf přes šipky.
- DataPlotter: obsahuje funkce pro zpracování uživatelského vstupu, načtení xarray dat a výběr grafů z předešlé třídy.

Main obsahuje uvítání a hlavní smyčku, ve které se odehrává všechna interakce s uživatelem. Program se prvně ptá, zdali je uživatel online nebo offline, pokud je online, tak se program ptá, zda chce uživatel stahovat data paralelně nebo ne, také je zde výběr ze 2 typů dat nebo zdali chce uživatel vytvářet nebo načítat data ze zálohy. Posléze se načtou data s pomocí třídy DataFetcher a ty jsou poslány do DataPlotter, kde je další uživatelské rozhraní pro výběr grafu ze třídy GraphPlotter.

Popis jednotlivých funkcí včetně vstupů a výstupů jsou dostupné v komentářích uvnitř kódu.

Pro vygenerování online dokumentace, lze použít příkaz: „*python -m pydoc -w*“.

Pydoc je součástí pythonu, pro bližší informace lze zavolat: „*python -m pydoc -h*“.

4. Testování

Při testování jsem zjistil, že paralelní načítání je v průměru *3-4x* rychlejší než sériové, avšak jednou se stalo, že určitá část dat nedošla, mezitím co u sériového načítání data došla vždycky. Testování bylo prováděno přes Bluetooth tethering, kdy jde vidět patrné rozdíly v časech. Přes ethernet pak byla doba načtení všech dat u paralelního okolo *6 s* a u sériového *24 s*.

Paralelně:

```
Načtena Průměrná měsíční teplota vzduchu ve srovnání s normálem 1991-2020 na území ČR a jednotlivých krajů, pro roky: 2021-2022
Načtena Průměrná měsíční teplota vzduchu ve srovnání s normálem 1981-2010 na území ČR a jednotlivých krajů, pro roky: 1961-2020
Načtena Průměrná měsíční teplota vzduchu ve srovnání s normálem 1961-1990 na území ČR a jednotlivých krajů, pro roky: 1961-2020
100%|██████████| 122/122 [00:18<00:00,  6.66it/s]
Načteny Měsíční úhrny srážek ve srovnání s normálem 1991-2020 na území ČR a jednotlivých krajů, pro roky: 2021-2022
Načteny Měsíční úhrny srážek ve srovnání s normálem 1981-2010 na území ČR a jednotlivých krajů, pro roky: 1961-2020
Načteny Měsíční úhrny srážek ve srovnání s normálem 1961-1990 na území ČR a jednotlivých krajů, pro roky: 1961-2020
100%|██████████| 122/122 [00:18<00:00,  6.76it/s]
Byla načtena data teplot
Byla načtena data srážek
```

Průměrná rychlost byla $\frac{122}{18} = 6.\overline{7}$ paketů/s.

Sériově:

```
Načtena Průměrná měsíční teplota vzduchu ve srovnání s normálem 1991-2020 na území ČR a jednotlivých krajů, pro roky: 2021-2022
Načtena Průměrná měsíční teplota vzduchu ve srovnání s normálem 1981-2010 na území ČR a jednotlivých krajů, pro roky: 1961-2020
Načtena Průměrná měsíční teplota vzduchu ve srovnání s normálem 1961-1990 na území ČR a jednotlivých krajů, pro roky: 1961-2020
100%|██████████| 122/122 [01:24<00:00,  1.45it/s]
Načteny Měsíční úhrny srážek ve srovnání s normálem 1991-2020 na území ČR a jednotlivých krajů, pro roky: 2021-2022
Načteny Měsíční úhrny srážek ve srovnání s normálem 1981-2010 na území ČR a jednotlivých krajů, pro roky: 1961-2020
Načteny Měsíční úhrny srážek ve srovnání s normálem 1961-1990 na území ČR a jednotlivých krajů, pro roky: 1961-2020
100%|██████████| 122/122 [01:25<00:00,  1.43it/s]
Byla načtena data teplot
Byla načtena data srážek
```

Průměrná rychlost byla $\frac{122}{84.5} = 1.44$ paketů/s.

Předpokládám, že pakety byly v celku a nebyly fragmentovány.

Při posledním měření bylo paralelní načítání $\frac{84+85}{18+18} = 4.69 \times$ rychlejší než sériové.

Testovány byly také chybové stavy, při odpojení internetu apod.

Kód také zachytává interrupci kódu.

5. Interakce

V této sekci rozeberu možné interakce od zapnutí skriptu až po rozhraní výběru grafu. Pro přeskočení této interakce lze zapnout globální proměnnou **DEBUG_SKIP**, program tak po spuštění přeskočí veškerý dialog, načte data ze záloh a skočí rovnou do interakce pro výběr grafů.

Možnosti:

a. Po uvítání je první dotaz na to, zdali jsem online
b. Zdali chci data teplot
c. Zdali chci data srážek
d. Zdali chci načítat paralelně
e. (dodatečně vypíše, že existuje záloha a jestli nechci přejít do offline režimu)
f. (zda si chci přepsat data)
g. (Zdali chci pracovat s teplotami, pokud byly načteny)
h. (Zdali chci pracovat se srážkami, pokud byly načteny)

Po těchto krocích se dostávám k výběru grafů, které více popíšu níže.

Pro porovnání odpovědí od uživatele mám seznamy slov, do kterých se uživatel musí trefit, jinak se daná otázka bude opakovat. Tyto seznamy můžu rozdělit do 4 hlavních kategorií: souhlas, nesouhlas, nápověda a konec (y, n, h, x). U některých grafů ale přibývá možnost vybrat si různé data pomocí čísel nebo specifického rozsahu a jejich kombinací. Tento vstup je pak kontrolován regulárním výrazem a dalšími kontrolami a vykazuje jisté prvky lexikální, syntaktické a sémantické analýzy.

Vstup z pohledu uživatele (enter pro přehlednost oddělím šipkou):

y->y->y->y->y->y->y->y->výběr grafu (0-9)

n->y->n->y->výběr grafu (0-9)

[ekvivalentní verze: no->yes->no->yes->výběr grafu (0-9) apod. viz nápověda přes help nebo h]

Jak můžeme vidět, tak interakci lze zkrátit, když načítáme data ze zálohy jen pro jeden typ dat.

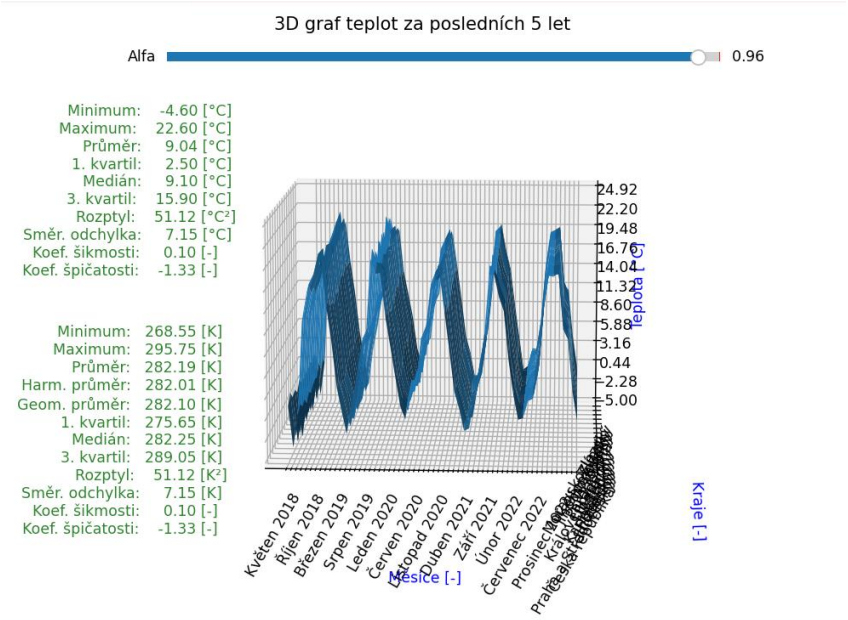
Pro nápovědu pak stačí napsat h a pro ukončení programu x kdykoliv při dotázání.

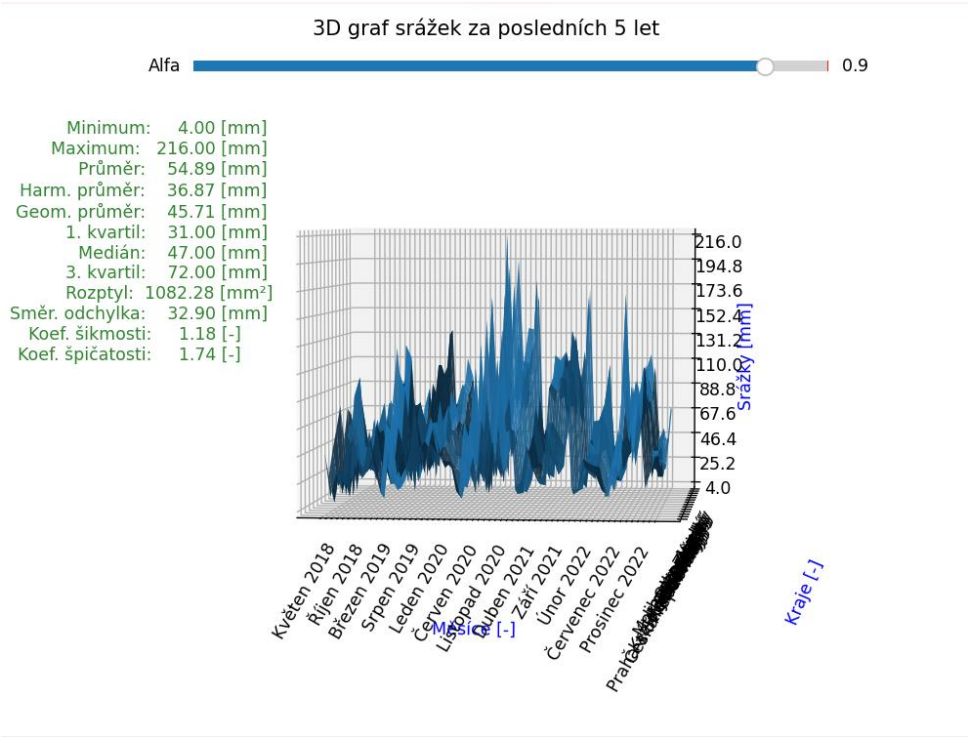
6. Grafy

Grafů je celkem 10. Pro podobnost s uživatelským rozhraním budu dále používat označení 0-9.

Graf 0)

První okna ukazují grafy teplot a srážek ve 3D za posledních 5 let na všech krajích v ČR.





Jak můžeme vidět v grafech, teploty mají určitou periodu s opakováním asi 1 rok, mezitím co u srážek nelze od oka určit, zda tu je nějaká perioda. Můžeme vidět taky statistická data. Pro kontrolu, harmonický průměr má být vždy rovno menší geometrickému průměru a ten je rovno menší aritmetickému průměru, což sedí.

Harmonický a geometrický průměr nemůže obsahovat záporná čísla, proto je teplota také převedena do kelvinů. U druhého grafu vidíme velkou rozmanitost dat, což lze potvrdit vysokým rozptylem. U koeficientu šikmosti vidíme, že teploty mají téměř pěknou symetrii, mezitím co srážky mají vyšší hodnotu do kladných čísel, což znamená že má větší počet hodnot na levé straně a je do určité míry způsobeno lokálními špičky na pravé straně (nejvyšší špička má hodnotu 216 mm) U koeficientu špičatosti pro teploty vidíme, že má zápornou hodnotu, což znamená, že má malou koncentraci dat, kolem průměru a větší na krajích, oproti tomu srážky mají hodnotu nad 1, což značí že je vyšší koncentrace hodnot kolem průměru (vyšší než u normálního rozdělení).

Graf 1)

Tento graf je podobný grafu 0 s rozdílem nastavení let. Možnosti jsou aktuálně od roku 1961-2022 (teoreticky to dokáže reagovat na budoucí přidávání let na webové stránce).

Možný vstup: „1961, 1981, 2001, 2020-2022“. Jak můžeme vidět, dají se vybrat dané roky nebo rozsah let v různé kombinaci.

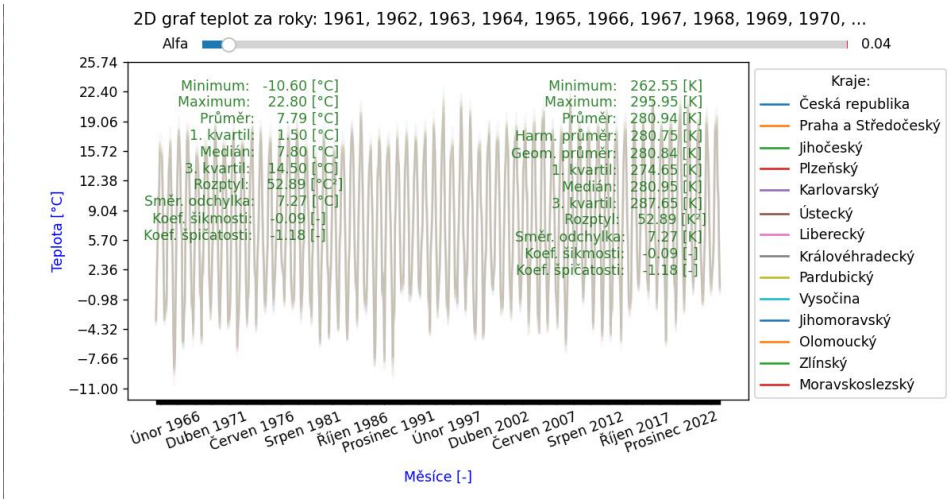
Graf 2)

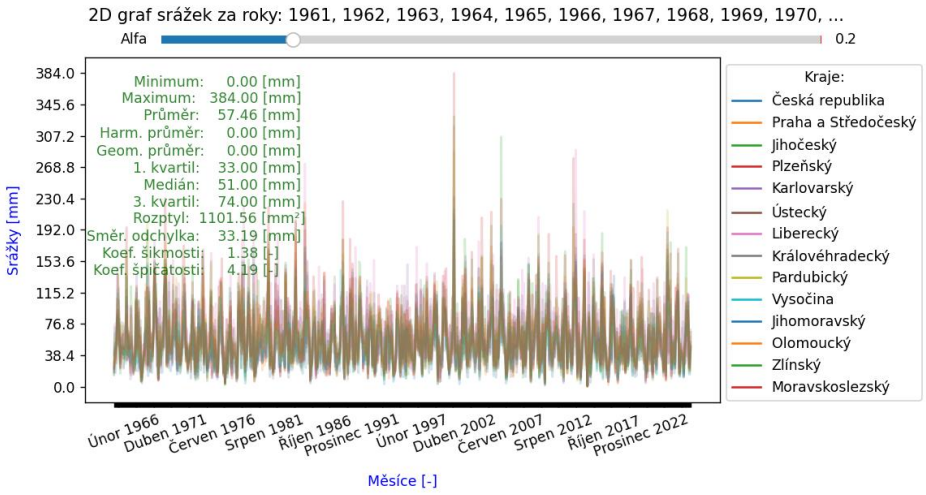
Tento graf je podobný grafu 1 s rozdílem nastavení i krajů. Krajů je celkem 14, ale Praha a Středočeský kraj se bere dohromady a pak je zde Česká republika, což by mělo značit průměrné hodnoty pro všechny kraje.

Možný vstup: „7“, „0-2, 5, 10-11“.

Graf 3)

Tento graf je podobný grafu 3, ale je ve 2D. V ukázce jsou všechny kraje a roky. Můžeme zde vidět využití alfa kanálu, bez kterého by nebylo možné vidět statistické data nebo jen velice těžce.

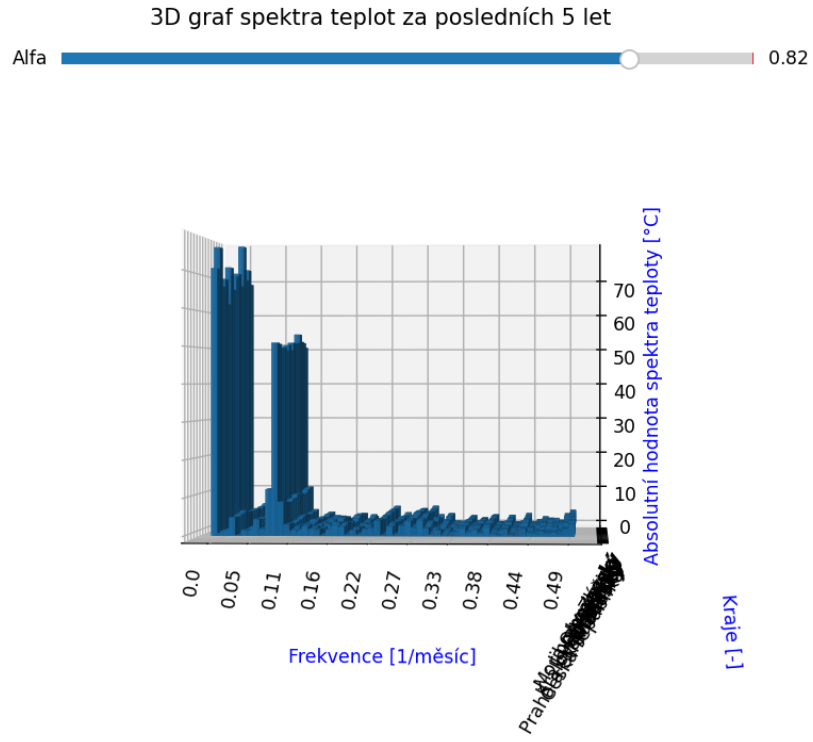


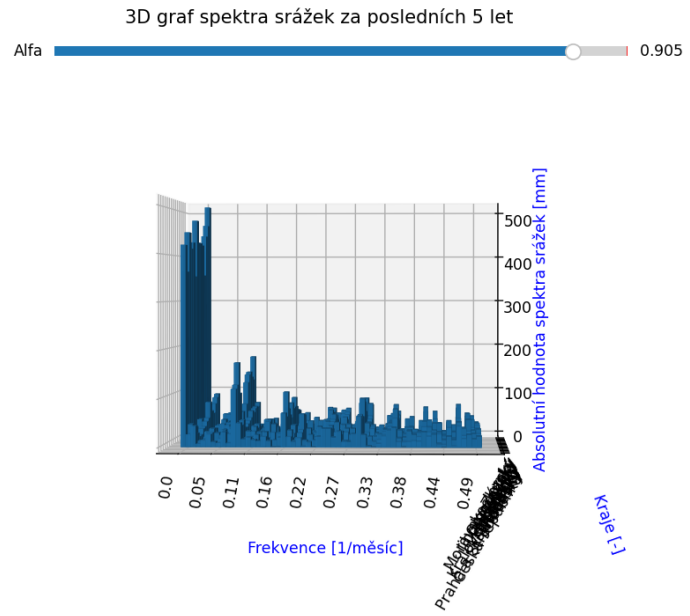


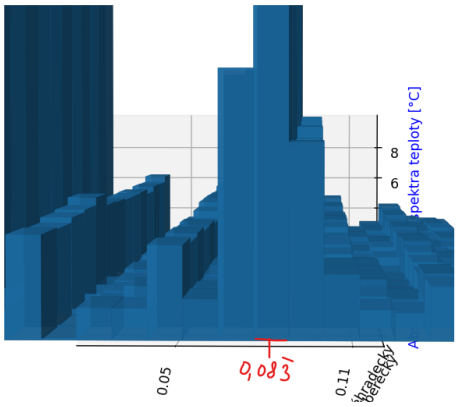
Volání může být stejné jako u grafu 2.

Graf 4)

Tento graf znázorňuje spektrum teplot nebo srážek pomocí histogramu ve 3D za 5 let pro všechny kraje.







Pro výpočet spektra byla použita metoda DCT, jejíž výstupem nejsou komplexní čísla jako u DFT, ale jen její absolutní složka. Jak můžeme vidět, tak u teplot je výrazná složka u hodnoty $\frac{0.8}{měsíc} \doteq \frac{0.8\overline{3}}{měsíc} = \frac{1}{12\ měsíc} = \frac{1}{rok}$, což sedí s předchozím odhadem, že teplota má periodu okolo 1 roku. U srážek lze vidět taky jistou vyšší hodnotu u frekvence za rok, ale není tak vysoká jako u srážek a je dost proměnlivá mezi kraji.

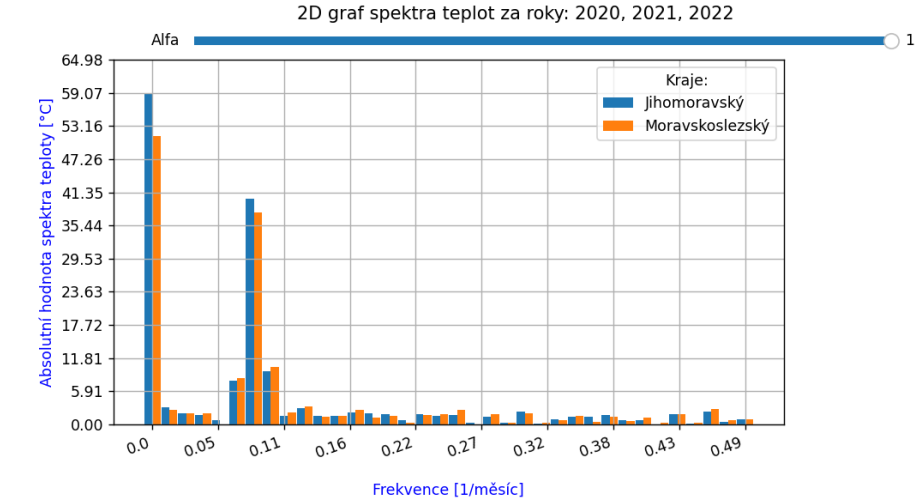
Graf 5)

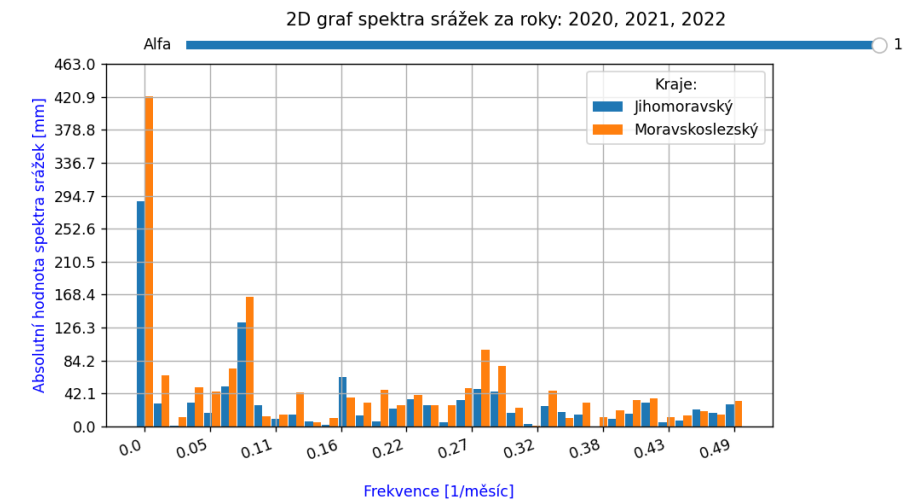
Tento graf je podobný grafu 4 s rozdílem nastavení let i krajů. Podobně jako je graf 1 a 2 na grafu 0.

Graf 6)

Tento graf je podobný grafu 5, ale je ve 2D.

Volaný vstup: „2020-2022“ a „10,13“.

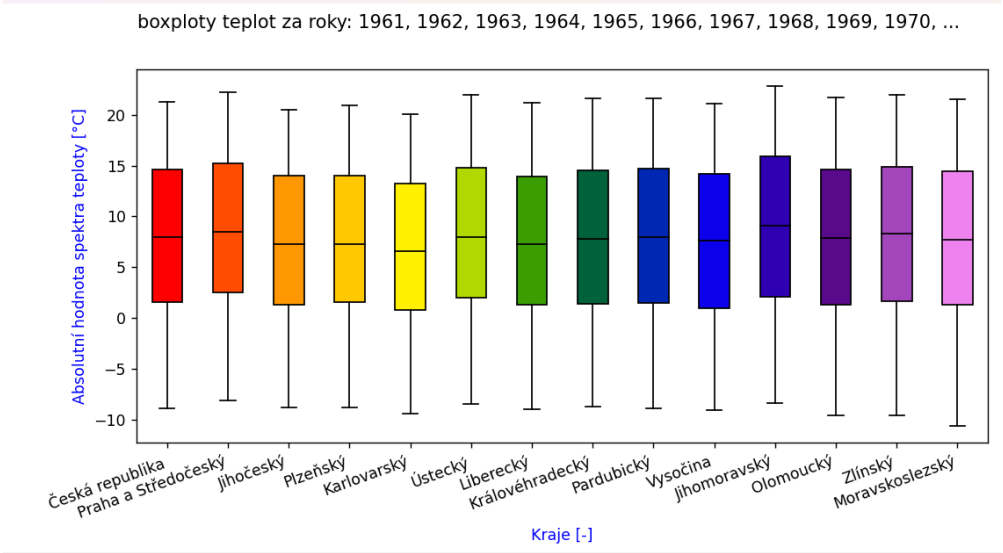


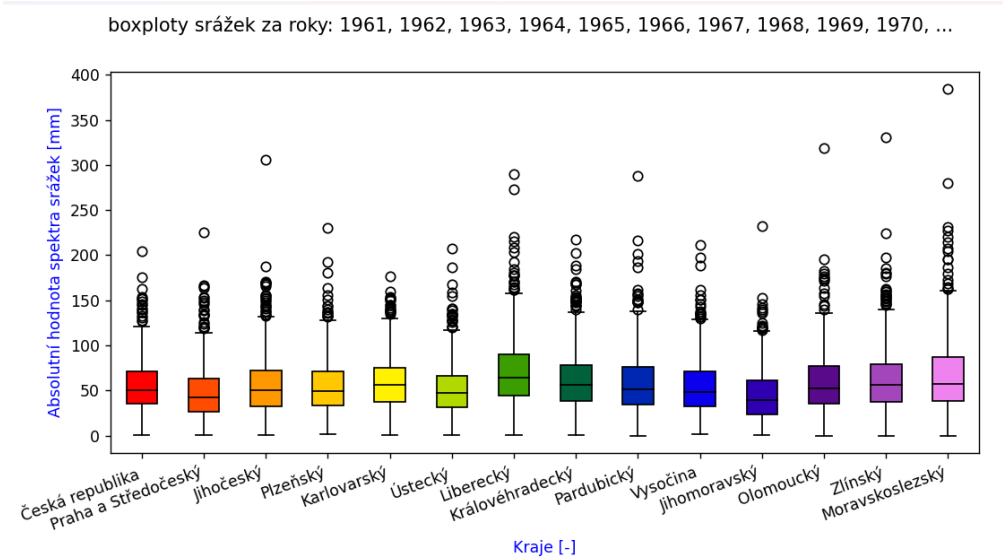


Graf 7)

Tento graf ukazuje boxploty teplot a srážek s nastavením let i krajů. Tyto boxploty ukazují různá statistická data.

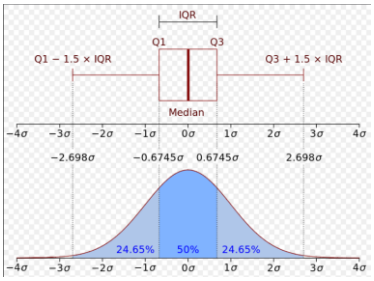
Volaný vstup: „1961-2022“ a potom „0-13“.





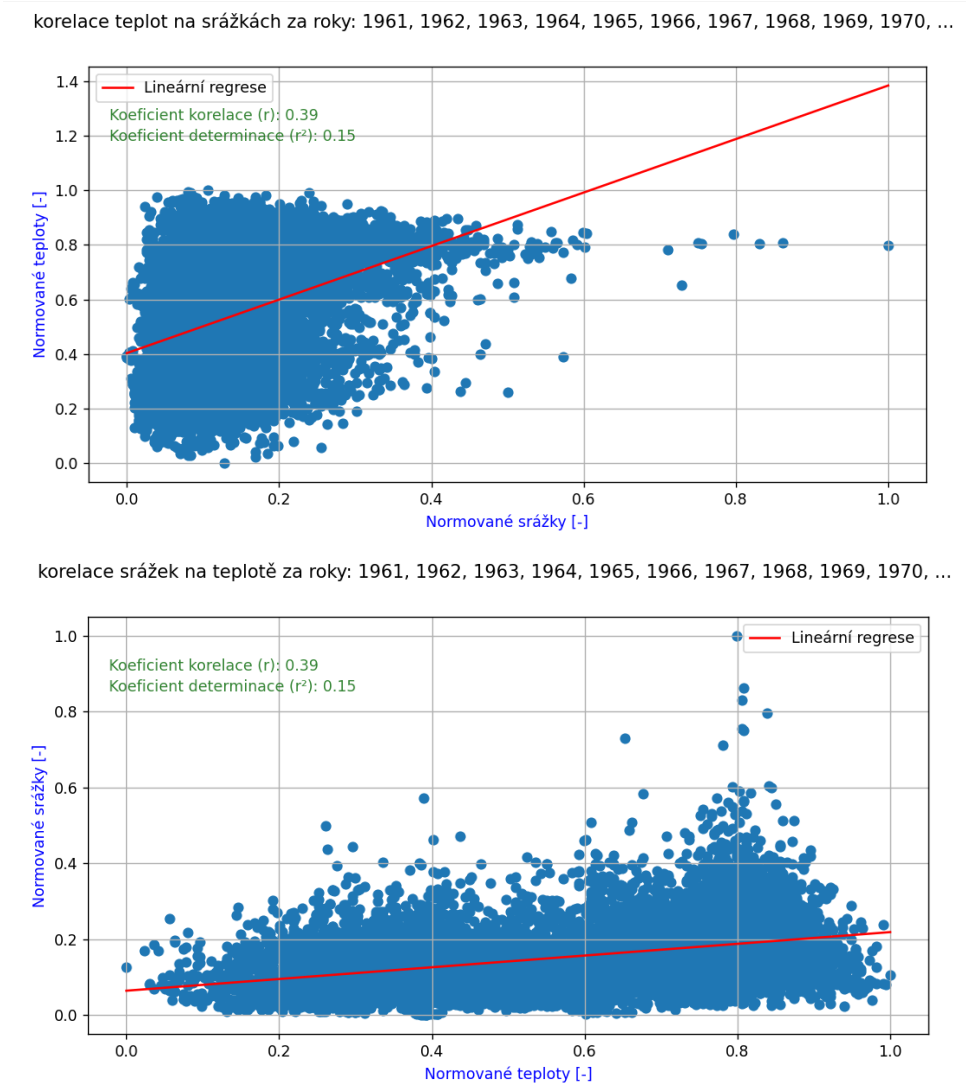
Tyto grafy znázorňují minima, maxima, medián, dolní/horní kvartil a v určitých případech i outsidery.

Pro nalezení outsiderů se používá Tukeyho metoda a jsou to hodnoty, co se vyskytují za vousy krabicového diagramu. Můžeme vidět velký počet outsiderů u srážek, což je způsobeno lokálními výkyvy v počasí.



Graf 8)

Tento graf ukazuje korelaci teplot na srážkách a opačně s nastavením let i krajů. Pro tyto účely se hodnoty normují.



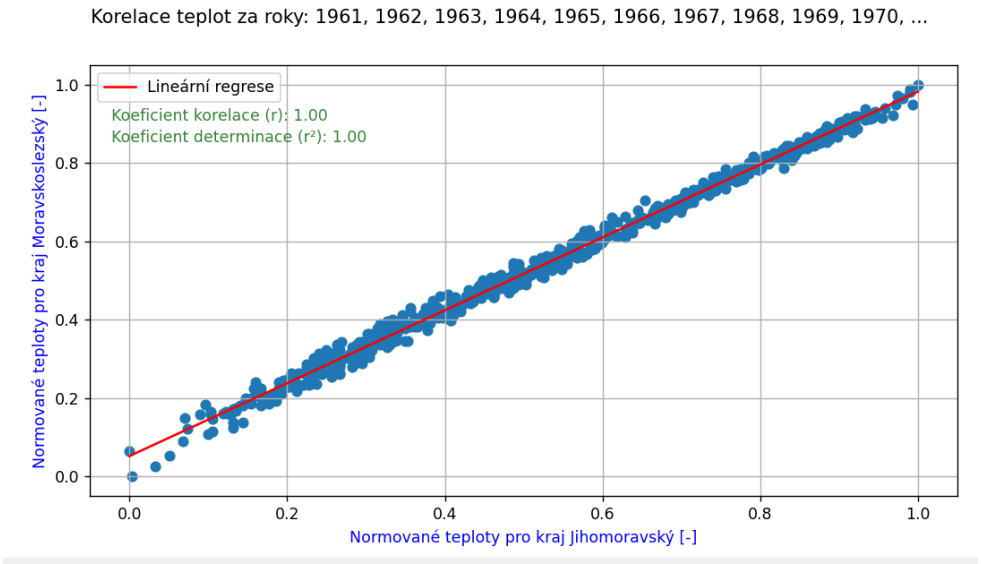
Z grafů můžeme vidět, že existuje nějaký pozitivní lineární vztah, ale ten není příliš velký.
Z koeficientu determinace můžeme říct, že 15 % variability závislé proměnné lze vysvětlit pomocí nezávislé proměnné a zbylých 85% je způsobeno jinými faktory.

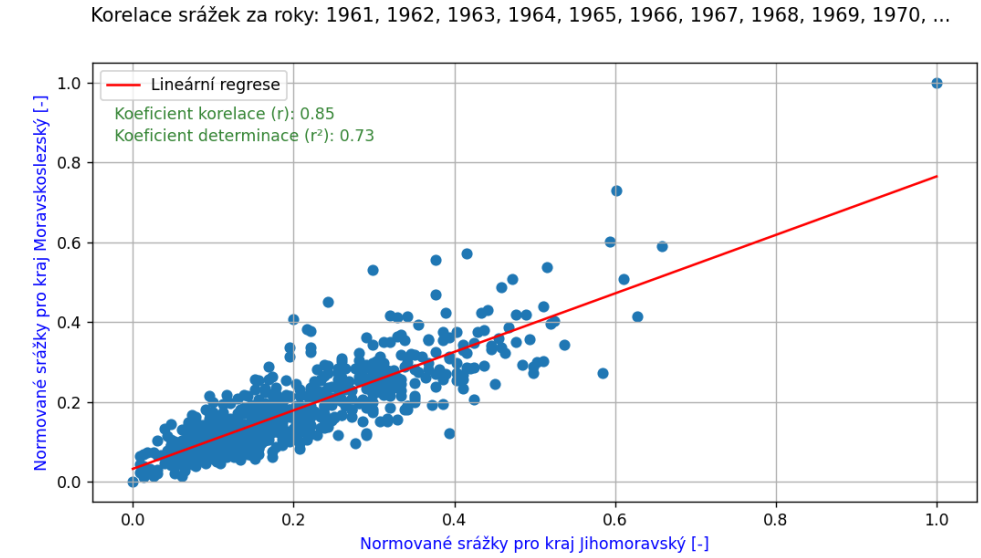
Když prší, tak se většinou sníží teplota, avšak je otázka, jak moc se to projeví v rámci průměrných měsíčních teplot.

Graf 9)

Tento graf ukazuje korelaci teplot nebo srážkách mezi 2 kraji s nastavením let i krajů. I zde se hodnoty normují.

Volaný vstup: „1961-2022“ a potom „10“ a „13“.





Z grafů můžeme vidět, že teploty mezi 2 kraji mají vysokou korelaci blízké 1 a můžeme s jistotou konstatovat, že změna teploty v jednom kraji podobně zasáhne i ostatní kraje v rámci průměrných měsíčních teplot. Relativně vysokou korelaci lze vidět i u srážek, avšak je mnohem menší, což může být dáno množstvím pohoří, dolin apod.

7. Závěr

Při použití skriptu na analýzu a vizualizaci meteorologických dat je důležité mít na paměti, že tato data jsou pod licencí CC BY-NC-ND 3.0 CZ. To znamená, že je nezbytné uvést původ dat, nepoužívat je pro komerční účely a nezasahovat do díla, podrobnější informace lze nalézt na adrese: https://creativecommons.org/licenses/by-nc-nd/3.0/cz/.

V celém skriptu používám typing, což umožňuje lepší odhalení chyb v kódu a zvýšení jeho čitelnosti.

Během testování byla funkčnost skriptu pečlivě ověřena a byly ošetřeny všechny možné chybové stavy, aby uživatelé mohli co nejjednodušeji a nejpřesněji pracovat s daty.

Skript poskytuje uživatelům užitečnou funkci na analýzu a vizualizaci meteorologických dat a může být použit pro další výzkum.

Kód v Pycharmu byl pečlivě navržen tak, aby neobsahoval žádné warningy nebo PEP chyby. Přestože zde mohou být některé weak warnings, avšak ty se dají bez problémů ignorovat.

Pro lepší čtení v kódu doporučuji použít IDE s možností zabalení tříd, funkcí nebo podmínek.

8. Přílohy

Seznam použitých zdrojů:

https://www.chmi.cz/historicka-data/pocasi/uzemni-teploty

https://www.chmi.cz/historicka-data/pocasi/uzemni-srazky

https://creativecommons.org/licenses/by-nc-nd/3.0/cz/

https://www.chmi.cz/files/portal/docs/meteo/ok/uzemni_teploty_cs.html

https://www.chmi.cz/files/portal/docs/meteo/ok/uzemni_srazky_cs.html

https://www.python.org/

https://peps.python.org/pep-0000/#

https://docs.python.org/3.9/library/index.html

https://www.w3schools.com/python/

https://realpython.com/

https://stackoverflow.com/

https://stackexchange.com/

https://python.cz/

Ukázka online dokumentace pomocí Pydoc:

Python 3.11.2 [tags/v3.11.2:878ead1, MSC v.1934 64 bit (AMD64)]
Windows-10

Module Index : Topics : Keywords
Get Search

main

index
d:\fekt\ing\2_semestr\csi\projekt\data_analyzer\main.py

```
Weather analyzer

Name: main.py
Description: Statistical weather analyser for the Czech Republic.
Autor: Václav Pastušek
Creation date: 11. 2. 2023
Last update: 29. 4. 2023
School: BUT FEEC
VUT number: 204437
Python version: 3.9.13
```

Modules

concurrent	matplotlib.colors	platform	signal
functools	matplotlib	matplotlib.pyplot	time
itertools	numpy	re	tqdm
keyboard	os	requests	xarray

Classes

builtins.Exception(builtins.BaseException)
- JumpException

builtins.object
- DataFetcher
- DataPlotter
- GraphPlotter
- UserInterface
- Utils

class **DataFetcher**(builtins.object)

```
DataFetcher() -&gt; None

The DataFetcher class represents an object that is responsible for fetching weather data from the internet.
It is capable of creating and loading data from a backup in case the internet connection is not available.
```

Methods defined here:

__init__(self) -> None
```
Initialize self.  See help(type(self)) for accurate signature.
```

create_new_backup(self, data: dict) -> None
```
Creates a new backup of data in NetCDF format.

:param data: Dictionary with weather data to be backed up.
:return: None
```

destroy_backup(self) -> None
```
Deletes the backup files from the backup directory for temperature and precipitation data if they exist.

:return: None
```

fetch_page(self, url: str, idx: int = 0, encode: str = '') -> tuple[int, str]
```
Fetches the HTML content of the specified URL and returns it as a tuple along with the specified index.

:param url: the URL to fetch the HTML content from
:param idx: the index of the fetched content
:param encode: the encoding type for the fetched content
:return: a tuple containing the index and the HTML content of the fetched page
```

get_data(self, online: bool, temper: bool, precip: bool, parallel: bool = False) -> dict
```
Fetches temperature and precipitation data from online/offline sources and returns it as a dictionary.

:param online: Flag whether to fetch data online or not.
:param temper: Flag whether to fetch temperature data or not.
:param precip: Flag whether to fetch precipitation data or not.
:param parallel: Flag whether to fetch data in parallel or not.
:return: A dictionary containing fetched data, where keys are "temper" and/or "precip".
```