Wrap up

05.12.2025

Course program

Basic python

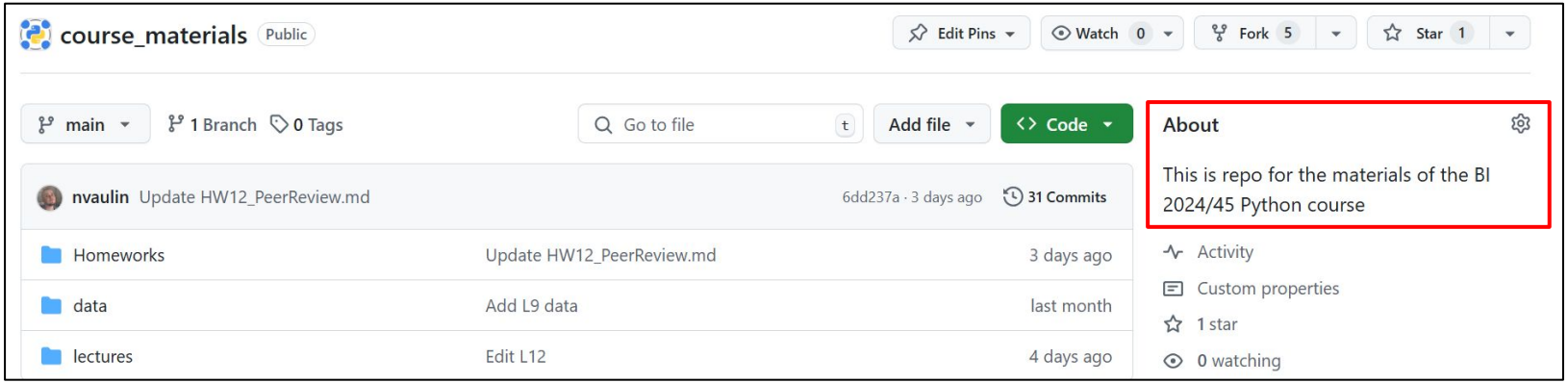
Scientific data analysis

1. **Intro. Git, GitHub**
2. Python recap, data types
3. Functions
4. Modules and libraries
5. Files
6. IDEs
7. Virtual environments
8. Regular expressions
9. Numpy
10. Pandas
11. Visualisation
12. Statistics
13. Discussion


ДЗ 2 - 5

README

- [x] Описание репо



README

- Описание репо
- Небольшое красочное интро

BioAssistant 2.0

BioAssistant is a toolkit for basic operations with DNA and RNA sequences. Your bioassistant will help you with such operations on DNA and RNA as transcription, finding the reversed, complementary and reversed complementary sequence, determining the gc-content and belonging of the sequence to a palindrome. It can also filter fastq by gc-content, sequence length and read quality. Have a fasta file, but can't make a blast? BioAssistant will remove unnecessary gaps in the sequence. Got too much unnecessary information after multiple alignments from blast? No problem! BioAssistant will help you leave only the most important alignments.

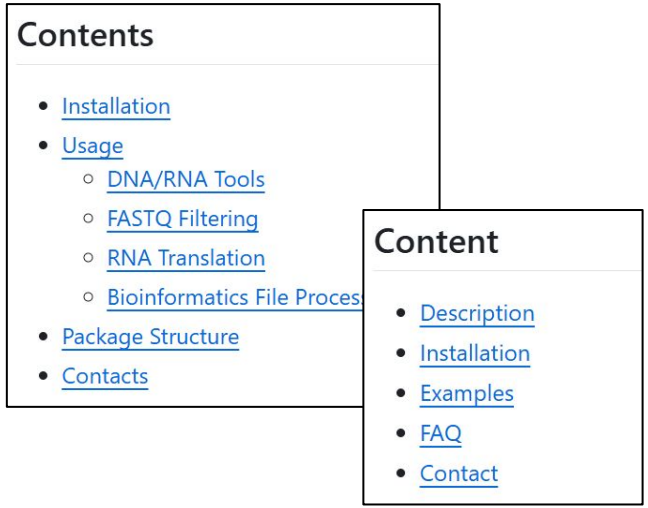
Just call and BioAssistant will help you!

Content

Installation

README

- Описание репо
- Небольшое красочное интро
- Содержание



README

- Описание репо
- Небольшое красочное интро
- Содержание
- Установка

Installation

1. Clone the repository:

```
git clone https://github.com/The-AGT/bioinf-utils.git
```

2. Navigate to the project directory:

```
cd bioinf-utils
```

3. Create a virtual environment (recommended):

```
python -m venv venv
source venv/bin/activate  # For Windows: venv\Scripts\activate
```

4. Install dependencies (if any):

```
pip install -r requirements.txt
```

System requirements

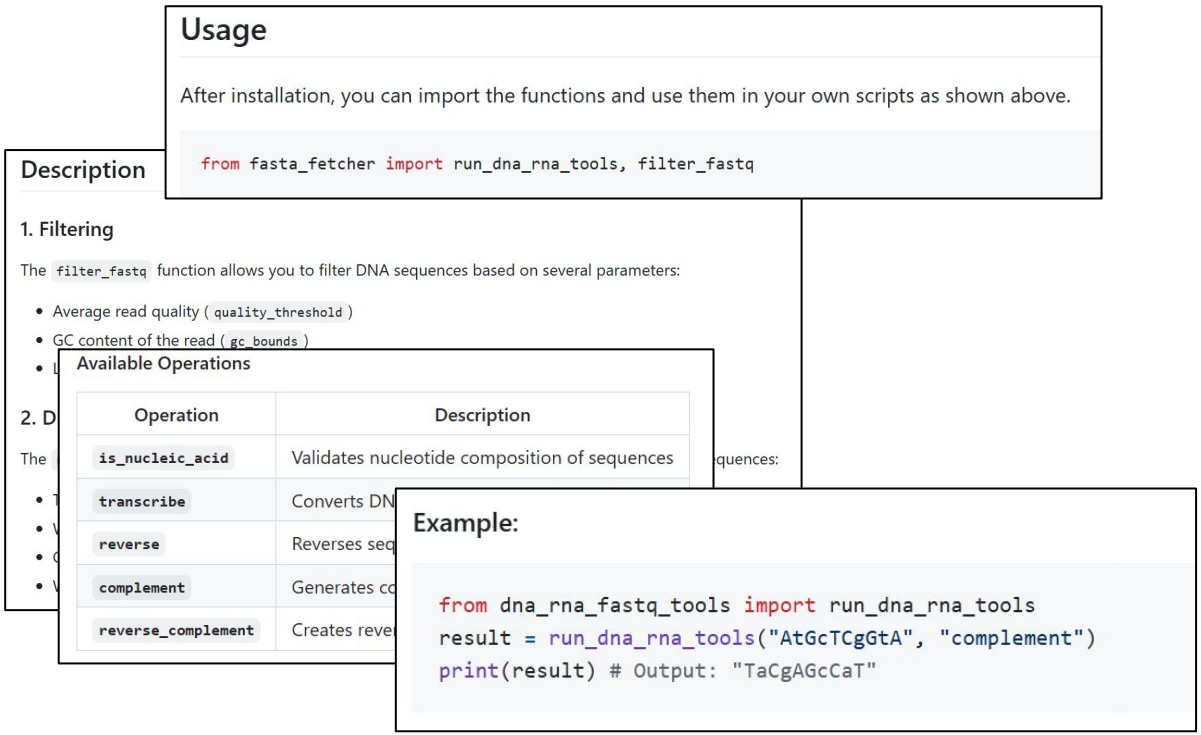
Operating System: Linux/Mac/Windows

Python: Version 3.6 or higher

Also take a healthy nervous system to figure out someone else's code

README

- Описание репо
- Небольшое красочное интро
- Содержание
- Установка
- Использование и описание



Usage

After installation, you can import the functions and use them in your own scripts as shown above.

```
from fasta_fetcher import run_dna_rna_tools, filter_fastq
```

Description

1. Filtering

The `filter_fastq` function allows you to filter DNA sequences based on several parameters:

- Average read quality (`quality_threshold`)
- GC content of the read (`gc_bounds`)

Available Operations

Operation	Description
`is_nucleic_acid`	Validates nucleotide composition of sequences
`transcribe`	Converts DN
`reverse`	Reverses seq
`complement`	Generates co
`reverse_complement`	Creates reve

Example:

```
from dna_rna_fastq_tools import run_dna_rna_tools
result = run_dna_rna_tools("AtGcTCgGtA", "complement")
print(result) # Output: "TaCgAGcCaT"
```

README

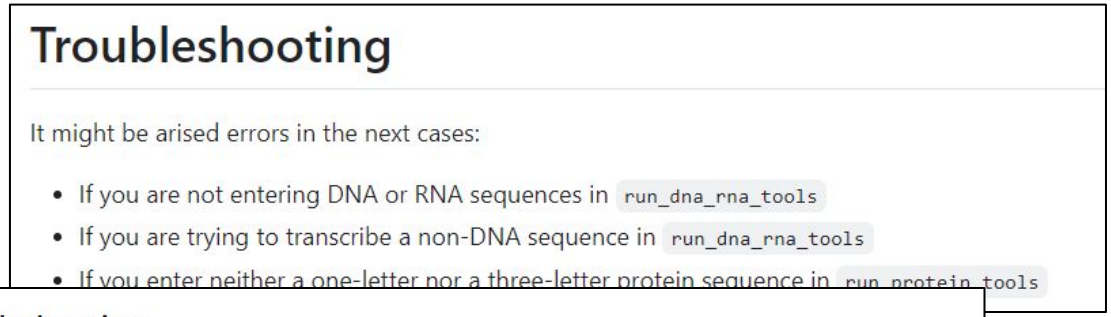
- Описание репо
- Небольшое красочное интро
- Содержание
- Установка
- Использование и описание

Examples

```
EXAMPLE_FASTQ = {
    # 'name' : ('sequence', 'quality')
    '@SRX079804:1:SRR292678:1:1101:21885:21885': ('ACAGCAACATAAACATGATGGGATGGCGTAAGCCCCCGAGATATCAGTTTACCCAGGA
    '@SRX079804:1:SRR292678:1:1101:24563:24563': ('ATTAGCGAGGAGGAGTGCTGAGAAGATGTCGCCTACGCCGTTGAAATTCCCTTCAATC
    '@SRX079804:1:SRR292678:1:1101:30161:30161': ('GAACGACAGCAGCTCCTGCATAACCGCGTCCTTCTTCTTTAGCGTTGTGCAAAGCATG
    '@SRX079804:1:SRR292678:1:1101:47176:47176': ('TGAAGCGTCGATAGAAGTTAGCAAACCCGCGGAACTTCCGTACATCAGACACATTCCG
    '@SRX079804:1:SRR292678:1:1101:149302:149302': ('TAGGGTTGTATTTGCAGATCCATGGCATGCCAAAAAGAACATCGTCCCGTCCAATA
    '@SRX079804:1:SRR292678:1:1101:170868:170868': ('CTGCCGAGACTGTTCTCAGACATGGAAAGCTCGATTCGCATACACTCGCTGAGTAA
    '@SRX079804:1:SRR292678:1:1101:171075:171075': ('CATTATAGTAATACGGAAGATGACTTGCTGTTATCATTACAGCTCCATCGCATGAA
    '@SRX079804:1:SRR292678:1:1101:175500:175500': ('GACGCCGTGGCTGCACTATTTGAGGCACCTGTCCTCGAAGGGAAGTTCATCTCGAC
    '@SRX079804:1:SRR292678:1:1101:190136:190136': ('GAACCTTCTTTAATTTATCTAGAGCCCAAATTTTAGTCAATCTATCAACTAAAATA
    '@SRX079804:1:SRR292678:1:1101:190845:190845': ('CCTCAGCGTGGATTGCCGCTCATGCAGGAGCAGATAATCCCTTCGCCATCCCATTA
    '@SRX079804:1:SRR292678:1:1101:198993:198993': ('AGTTATTTATGCATCATTCTCATGTATGAGCCAACAAGATAGTACAAGTTTTATTG
    '@SRX079804:1:SRR292678:1:1101:204480:204480': ('AGTGAGACACCCCTGAACATTCCTAGTAAGACATCTTTGAATATTACTAGTTAGCC
    }
run_fastq_tools(EXAMPLE_FASTQ, gc_bounds=30)
#{'@SRX079804:1:SRR292678:1:1101:190136:190136': ('GAACCTTCTTTAATTTATCTAGAGCCCAAATTTTAGTCAATCTATCAACTAAAATACC
#  'DACD@BEECEDE.BEDDDDD,>:@>EEBEEHEFEHHFFHH?FGBGFBBD77B;;C?FFFFGGFED.BBABBG@DBBE')}
run_fastq_tools(EXAMPLE_FASTQ, gc_bounds=40, length_bounds=80, quality_threshold=35)
#{'@SRX079804:1:SRR292678:1:1101:171075:171075': ('CATTATAGTAATACGGAAGATGACTTGCTGTTATCATTACAGCTCCATCGCATGAATA
#   'HGHHHHGFHHHHFHHEHHHHFGEHFGFGGGHHEEGHHEEHBHHFGDDECEGGGEFGF<FGGIIGEBGDFFFGFFGGFGF')}
```

README

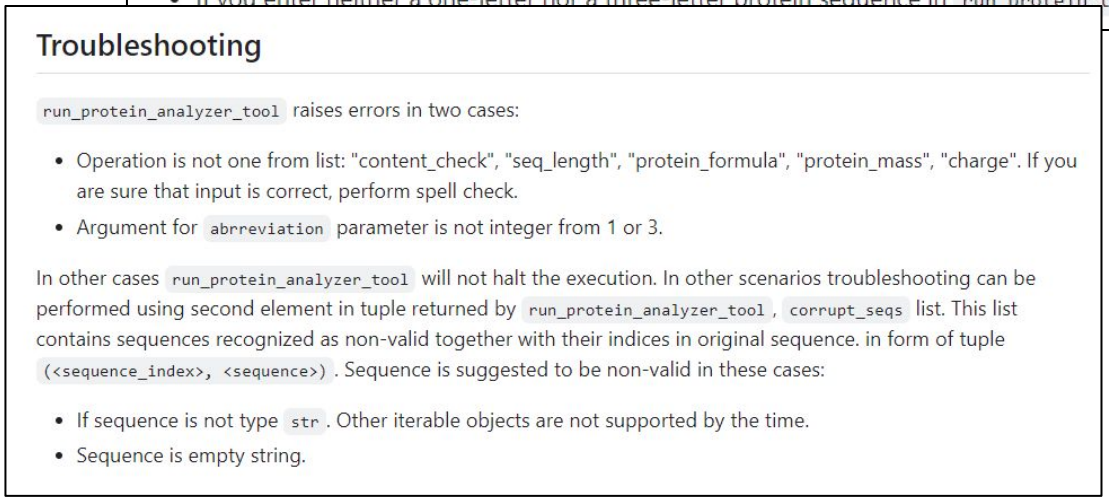
- Описание репо
- Небольшое красочное интро
- Содержание
- Установка
- Использование и описание
- Траблшутинг



Troubleshooting

It might be arised errors in the next cases:

- If you are not entering DNA or RNA sequences in `run_dna_rna_tools`
- If you are trying to transcribe a non-DNA sequence in `run_dna_rna_tools`
- If you enter neither a one-letter nor a three-letter protein sequence in `run_protein_tools`



Troubleshooting

`run_protein_analyzer_tool` raises errors in two cases:

- Operation is not one from list: "content_check", "seq_length", "protein_formula", "protein_mass", "charge". If you are sure that input is correct, perform spell check.
- Argument for `abrreviation` parameter is not integer from 1 or 3.

In other cases `run_protein_analyzer_tool` will not halt the execution. In other scenarios troubleshooting can be performed using second element in tuple returned by `run_protein_analyzer_tool` , `corrupt_seqs` list. This list contains sequences recognized as non-valid together with their indices in original sequence. in form of tuple `(<sequence_index>, <sequence>)` . Sequence is suggested to be non-valid in these cases:

- If sequence is not type `str` . Other iterable objects are not supported by the time.
- Sequence is empty string.

README

- Описание репо
- Небольшое красочное интро
- Содержание
- Установка
- Использование и описание
- Траблшутинг
- Контакты, ссылки

Contacts

We hope our module provides useful tool for your work. If you encounter any errors, please mail one from our team:

[redacted] - [redacted]@gmail.com Implemented: `protein_formula`, `protein_mass`, `seq_length`.

[redacted] - [redacted]@mail.ru Implemented: `aa_content_check`, `aa_chain_charge`.

[redacted] - [redacted]@gmail.com Teamlead. Implemented: `dna_rna_tools` module and `Mann_Whitney_U`, `decomposition`, `seq_transform`, `check_and_procees_seq`, `print_result`, `run_protein_analyzer_tool` from `` module.

Contributions and contacts

Feel free to report any bugs and problems encountered. Any bug reported is appreciated. Email: [redacted]@gmail.com

References

1. T.F. Smith, M.S. Waterman, (1981). Identification of common molecular subsequences. Journal of Molecular Biology.

Немного о коде

- Аннотации типов
- Докстринги
- Неизменяемые значения по-умолчанию

```python
def filter_fastq(
    input_fastq: str,
    output_fastq: str,
    gc_bounds: Union[Tuple[float, float], float] = (0, 100),
    length_bounds: Union[Tuple[int, int], int] = (0, 2**32),
    quality_threshold: int = 0,
) -> Dict[str, Tuple[str, str]]:
    """
    The function works with fastq sequences. Accepts 5 arguments as input

    The function works on the fly, accepts a fastq file,
    selects sequences for recording and saves the filtered data. Functions
    checks the existence of output directory and if missing creates
    a folder "filtered"
    :param input_fastq: str
         The path to real fastq file with sequence id, quality, sequence
    :param output_fastq: str
         The name of output fastq file with filtered sequences
    :param gc_bounds:
         The GC interval of the composition for filtering.If there is one
         argument, then it is assumed that this is the
         upper bound
    :param length_bounds:
         Length interval for filtering. It is set similarly to gc_bounds
    :param quality_threshold:
         The threshold value of the average reed quality for filtering.
         Quality is calculated by converting the ASCII encoding scale
         to Q-Score.The average quality is calculated for all nucleotides
         and those below the threshold are discarded.

    Function uses additional module filter_fastq.py
    """
    if isinstance(gc_bounds, (int, float)):
        gc_bounds = (0, gc_bounds)
    if isinstance(length_bounds, (int)):
        length_bounds = (0, length_bounds)
```

Немного о коде

Шебанг

Докстринга модуля

Импорты

Функции

Code | Blame | 96 lines (72 loc) · 2.8 KB

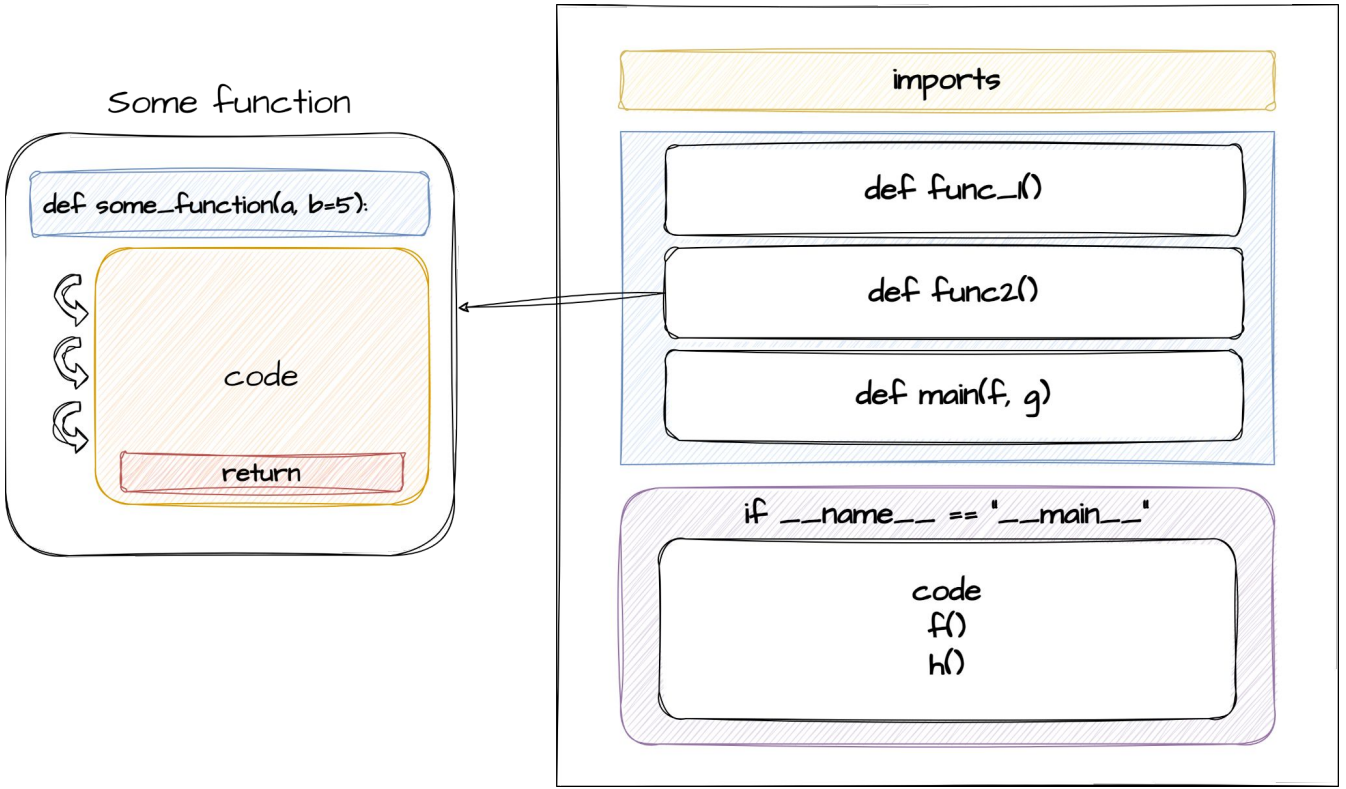
```
1  #!/usr/bin/env python3
2  """
3  FASTQ_TOOL
4  """
5
6  from typing import Union
7
8
9  def calc_gc(seq: str) -> float:
10     """
11     calculates GC ratio of a given sequence
12
13     Args:
14     seq - a sequence to use
15
16     Returns:
17     gc_ratio - a GC-content of a given sequence
18     """
19
20     gc_ratio = round(((seq.count('G') + seq.count('C'))/len(seq)) * 100, 2)
21     return gc_ratio
22
```

Порядок import’ов

Автопроверка импортов: isort

- Разделяются на 3 группы:
  1. Стандартная библиотека
  2. Сторонние библиотеки
  3. Локальные модули
- Сперва `import`, потом `from import`
- В алфавитном порядке

Немного о коде



ДЗ 6

Запуск pain.py

Характеристики ОС
Ссылка на conda/mamba

```
wget <URL> && cd <dir>
```

```
conda env create -f environment.yaml
```

```
conda activate env
```

Редактирование pandas

```
python pain.py
```

Редактирование pandas

Путь до файла

/<env_path>/lib/python3.13/site-packages/pandas/core/frame.py

“Comment two following lines”

```
if isinstance(index, set):
  raise ValueError("index cannot be a set")
```

nano / vim

```
nano +699 <file>
```

Редактирование pandas

Путь до файла

/<env_path>/lib/python3.13/site-packages/pandas/core/frame.py

“Comment two following lines”	nano / vim
`if isinstance(index, set):` `  raise ValueError("index cannot be a set")`	`nano +699 <file>`

Патч

.github README.md frame.py	Then, please, **copy file** `frame.py`, which where in the repository using: `cp /home/hw7_final/frame.py /home/hw7_final/hw7_final_venv/lib/python3.12/site-packages/pandas/core`

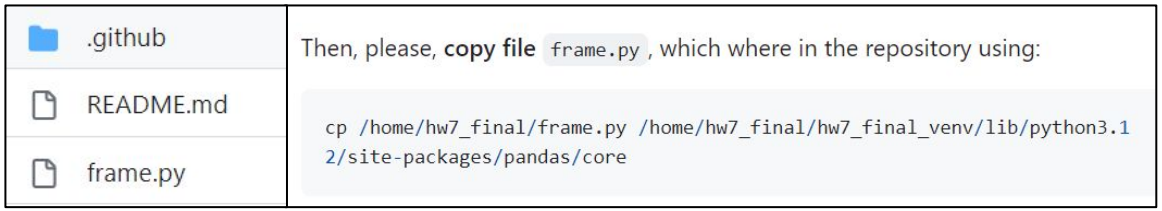
Редактирование pandas

Путь до файла

/<env_path>/lib/python3.13/site-packages/pandas/core/frame.py

“Comment two following lines”	nano / vim
`if isinstance(index, set):` `  raise ValueError("index cannot be a set")`	`nano +699 <file>`

Патч




```
sed -i "/# GH47215/,/^$/ {/^$\|# GH47215/! s/^/#/g}" <file>
```

ДЗ 7

Перевод на соленый язык

```
def salt_vowel(vowel):
    return vowel.group() + 'с' + vowel.group().lower()


def salt_text(text):
    vowels = re.compile(r'([ауоиэыяюеёАУОИЭЫЯЮЕЁ])')
    return re.sub(pattern=pattern, repl=salt_vowel, string=text)
```

Перевод на соленый язык

```
def salt_vowel(vowel):
    return vowel.group() + 'c' + vowel.group().lower()


def salt_text(text):
    vowels = re.compile(r'([ауоиэыяюеёАУОИЭЫЯЮЕЁ])')
    return re.sub(pattern=pattern, repl=salt_vowel, string=text)
```

Аккорды

```
def get_chords(input_path: str)-> set:
    """
    Get chords from a song in russian

    Argument:
    -input_path(str): input path to the file
    with lyrics and chords of the song

    Returns:
    -chords(set): a set with chords
    """
    pattern = r'[a-zA-Z]+\b'
    chords = set()
    with open('data/song.txt') as f:
        for line in f:
            chords = chords.union(re.findall(pattern, line))
    return chords
```

Аккорды

```
def get_chords(input_path: str)-> set:
    """
    Get chords from a song in russian

    Argument:
    -input_path(str): input path to the file
    with lyrics and chords of the song

    Returns:
    -chords(set): a set with chords
    """
    pattern = r'[a-zA-Z]+\b'
    chords = set()
    with open('data/song.txt') as f:
        for line in f:
            chords = chords.union(re.findall(pattern, line))
    return chords
```

```
r'[A-Z][a-z]?'
```

```
r"\b[A-G][m]?\b"
```

```
r'[A-H][m]?\d?[#]?'
```

```
r'\b[A-H]m?[1-7]?\b'
```

```
r"[CDEFGABH]m?b?[0-9]?#?"
```

Аккорды

```python
def get_chords(input_path: str)-> set:
    """
    Get chords from a song in russian

    Argument:
    -input_path(str): input path to the file
    with lyrics and chords of the song

    Returns:
    -chords(set): a set with chords
    """
    pattern = r'[a-zA-Z]+\b'
    chords = set()
    with open('data/song.txt') as f:
        for line in f:
            chords = chords.union(re.findall(pattern, line))
    return chords
```

```
r'[A-Z][a-z]?'
```

```
r"\b[A-G][m]?\b"
```

```
r'[A-H][m]?\d?[#]?'
```

```
r'\b[A-H]m?[1-7]?\b'
```

```
r"[CDEFGABH]m?b?[0-9]?#?"
```

```
\b[A-H][#b]?(?:m|maj|dim|sus[24]|aug)?(?:7|9|6)?\+?\b
```

```
\b[A-G](?:#|b)?(?:7)?(?:\+|m|maj7|maj|sus2|sus4|dim|dim7|m6|m7|m9|7|6|9)?\b
```

Переименовалка

```
def rename_files(dir: str, pattern: str, new_pattern:str = None, sample_names:dict = None, to_replace = False):
    """
    Rename files by a chosen pattern and/or by a dictionary of sample names

    Arguments:
    - dir (str): the name of the directory with files
    - pattern (str): the pattern for files selection
    - new_pattern (str):  the pattern by which the files are renamed
    - sample_names (dict): the dictionary that sets the rules for renaming
    samples (applied after renaming by the pattern)
    - to_replace (bool, default = False): rename files with the deletion of
    the original ones (True), or with copying (False)
    """
    files = os.listdir(dir)
    for file in files:
        if re.match(pattern, file):
            if new_pattern:
                new_file = re.sub(pattern, new_pattern, file)
            if sample_names:
                for pattern in sample_names:
                    new_file = re.sub(pattern, sample_names[pattern], new_file)
            old_path = os.path.join(dir, file)
            new_path = os.path.join(dir, new_file)
            if to_replace:
                os.rename(old_path, new_path)
            else:
                shutil.copy2(old_path, new_path)
```

Еще пара слов

Задание 14

Получите как можно больше чисел из этого набора.

Ответ: `['4', '8.0', '+16', '-16', '-23.42', '3.14e15', '23e-42', '100.000.000', '-3.099e-734.149']`

```
number = r'(?:\d+)' + r'(?:\.\d+)*'
number = r'[+-]?' + number
exponential_part = fr'(?:[eE]{number})?'
number = number + exponential_part
```

Еще пара слов

```
r\"[0-9\\.\\+\\-e]+\" "
r\"[+-]?\\b[^\\s]+\\b\" "
r\"\\+\\d+|[-\\d\\w.]+\""
r'[+-]*(?:\\d+[.,e]*(?:e+|-)*)+'"
r'\\b\\w*[1234567890]{2}\\w*\\b' "
r'[+-]?[.\\deE+-]*\\d[.\\deE+-]*'",
r\"[\\+\\-\\d][\\.\\d]\\d*[\\de]\""
r'[+-]?\\d*\\.?\\d+(?:e[+-]?\\d+)?'"
r\"\\b[0-9+-]?[0-9e\\-.]*[0-9]\\b\""
r'[-+]?\\d+\\.?\\d*(?:[eE][-+]?\\d+)?'"
r'[-+]?\\d+\\.?\\d*(?:[eE][-+]?\\d+)?'"
r\"[+-]?\\d+\\.?\\w*\\-?\\d*\\.?\\d*\\b\""
r'[+-]?\\d*\\.?\\d+\\.?\\d+(?:[e][+-]?\\d+)?' "
r'[-+]?\\w+[\\.|-]?\\w*[\\.|-]?\\w*[\\.]?\\w*'"
r'[+-]?\\d+(?:\\.\\d+)*(?:e[+-]?\\d+(?:\\.\\d+)*)?'"
r\"[\\-|+]?\\d+\\.*\\d*\\.*\\d*e*\\-?\\d*\\.*\\d*\""
r'[+-]?\\d+(?:\\.\\d+)*(?:e[+-]?\\d+(?:\\.\\d+)*)?'"
r'[-+]?\\d+\\.?\\d*\\.?\\d*(?:[e]?[-+]?\\d+\\.?\\d*)?'"
r'[+-]?\\d+(?:\\.\\d+)?(?:e[+-]?\\d+)?|\\d+\\.\\d+|\\d+'"
r\"[+-]?\\d+[\\.e]?\\d*[\\.e]?\\d*[\\.e-]?\\d*\\.?\\d*\" "
r\"[-+]?(?:[\\d]+?[\\.\\d]*)(?:[e][-]?[\\d]+?[\\.\\d]*)?\" "
r'[\\-\\+]?\\d*\\.?\\d*e?\\-?\\d*\\.?\\d*e?\\-?\\d*[^\\s]' "
r'[+-]?[0-9]+(?:\\.[0-9]+)*(?:e[+-]?[0-9]+(?:\\.?[0-9]+)*)?'"
r\"[+-]?\\d*\\.?\\d+(?:\\.\\d+)*(?:[eE][+-]?\\d+)?(?:\\.\\d+)?\""
r\"[+-]?(?:\\d{1,3}(?:\\.\\d{3})*|\\d*\\.?\\d+(?:e[+-]?\\d+)?)(?=\\s|$)\""
r\"(?<!\\w)[+-]?\\d+(?:\\.\\d+)?(?:\\.\\d+)?(?:[eE][-+]?\\d+[.]?\\d+[.]?)?\" "
r\"[+-]?(?:\\d{1,3}(?:\\.\\d{3})+|\\d+(?:\\.\\d+)?)(?:e[+-]?\\d+(?:\\.\\d+)?)?\""
r\"[+-]?(?:\\d{1,3}(?:\\.\\d{3})+|\\d*\\.\\d+|\\d+)(?:e[+-]?\\d+)?(?:\\.\\d+)?\""
r'[+-]?\\d+(?:\\.\\d+)*(?:e[+-]?\\d+(?:\\.\\d+)*)?' #зачем такие сложные числа...."
re.compile(r'[+-]?(?:\\d{1,3}(?:\\.\\d{3})+|\\d+(?:\\.\\d+)?)(?:e[+-]?\\d+(?:\\.\\d+)?)?')"
r'(?:^|\\s)([+-]?(?:\\d{1,3}(?:\\.\\d{3})*|\\d+)(?:\\.\\d+)*(?:[eE][+-]?(?:\\d+(?:\\.\\d+)*)?)?)(?=\\s|$)'"
r'\\d{3}\\.\\d{3}\\.\\d{3}|\\d+\\.\\d+[e]\\d+|\\d+\\.\\d+[^\\.]|\\d+[e][-]\\d+|[-]\\d+\\.\\d+[e][+-]\\d+\\.\\d+|[^\\.][+-]\\d+\\.\\d+\\b|[+-]\\d*\\b|\\b[\\d]\\b'"
```

alexk-git	162
lenaparshina	107
arnautoleg	91
Mikhail-Dobryakov	83
alinatgrv	81

Еще пара слов

Задание 15

Получите все валидные номера телефона из 'data/phones.txt'.

Ответ: `['"7(911) 345-34-56"', '"7(923)355-56-53"', '"+7(923)355-56-53"', '"8(988)245 45 32"', '"88005553535"', '"+7 921 445 43 22"']`

Еще пара слов

Задание 15

Получите все валидные номера телефона из 'data/phones.txt'

```
r\"[\\+78]\" "
re.compile(r\"\"\"\n",
r'\"\\+?\\d[\\d() \\-]{9,}\\d\"' "
r'(?:[A-Z][a-z]+(?: [A-Z][a-z]+)*: \")([^\"]+)'"
r'\"(?:\\+7|8|7).\\d{3}.*\\d{3}.\\d{2}.\\d{2}\"|\"\\d{11}\"'"
r'\"(?:\\+?7|8)[ (]?\\d{3}[ )]? ?\\d{3}[ -]?\\d{2}[ -]?\\d{2}\"'"
r'[\"\\']?[+78][(]?\\d{3}[)]?[ -]?\\d{3}[ -]?\\d{2}[ -]?\\d{2}[\"\\']?'"
r\"\\+?[78]\\s?\\(?[98]\\d{2}\\)?\\s?\\d{3}\\-?\\s?\\d{2}\\-?\\s?\\d{2}\""
r'[\\\"\\'][78]?\\(?\\d{3}\\)?\\d{2,3}[-\\s]?\\d{2}[-\\s]?\\d{2}[\\\"\\']'"
r'[\\\"](?:\\+?7|8)[( ]?\\d{3}[) ]{0,2}\\d{3}[- ]?\\d{2}[- ]?\\d{2}[\\\"]'"
r'\"(?:\\+7|7|8)\\s?\\(?\\d{3}\\)?[\\s-]?\\d{2,3}(?:[\\s-]?\\d{2}){2,3}\"'"
r'\\\"(?:[+]?[7]|[8])[ (]?\\d{3}[ )]?\\s?\\d{3}[ -]?\\d{2}[ -]?\\d{2}\\b\\\"'"
r'8\\d{10}|\\+?[78][\\s\\(-]?\\d{3}[\\)\\s-]?\\d{3}[\\s-]?\\d{2}[\\s-]?\\d{2}[\\s-]?\\d{2}'"
r'\"(?:\\+7|7|8)[^\\d\"]*\\d{3}[^\\d\"]*\\d{3}[^\\d\"]*\\d{2}[^\\d\"]*\\d{2}\"' "
r'(?:\\+7|7|8)\\s*\\(?[89][0-9]{2}\\)?\\s*[0-9]{3}[-\\s]?[0-9]{2}[-\\s]?[0-9]{2}'"
r\"\\\"(?!\\+8)\\+?[78]\\(?\\s?\\d{3}\\)?\\s?\\d{3}\\s?-?\\d{2}\\s?-?\\d{2}\\\"\" "
r\"\\\"(?:8|\\+?7)\\s?\\(?\\d{3}\\)?\\s?\\d{3}\\s?[\\-]?\\d{2}\\s?[\\-]?\\d{2}\\\"\""
r'\\\"(?:7|\\+7|8)(?: |\\()?\\d{3}(?: |\\))?[\\- ]?\\d{3}[\\- ]?\\d{2}[\\- ]?\\d{2}\\\"'"
r'[\"\\']{?:\\+?7|8)[\\s-]?\\(?\\d{3}\\)?[\\s-]?\\d{3}[\\s-]?\\d{2}[\\s-]?\\d{2}[\"\\']'"
r\"[\\\"].*?[^6][^6].*?[\\\"]\" #извините, это выше моих сил, я не знаю, как это сделать"
r\"\\\"[+]?[78]\\s?[/s\\(]?[98]\\d{2}[\\s\\)]?\\s?\\d{3}[\\s\\-]?\\d{2}[\\s\\-]?\\d{2}\\\"\" "
r'\\b(?:\\+7|7|8)\\s?\\(?\\d{3}\\)?[\\s\\-]?\\d{3}[\\s\\-]?\\d{2}[\\s\\-]?\\d{2}\\b[^\\s]'\n",
r'[\"\\']?(?:\\+7|[78])[\\s-]?\\(?[89]\\d{2}\\)?[\\s-]?\\d{3}[\\s-]?\\d{2}[\\s-]?\\d{2}[\"\\']?'"
r\"[\\\"\\']{?:\\+7|7|8)[(\\s]?[\\d]{3}[)]?[\\s]?[\\d]{3}[-\\s]?[\\d]{2}[-\\s]?[\\d]{2}[\\\"\\']\" "
r\"['\\\"]?(?:\\+7|7|8)\\s*\\(?\\d{3}\\)?[\\s\\--]?\\d{3}[\\s\\--]?\\d{2}[\\s\\--]?\\d{2}['\\\"]?\" "
r\"\\\"(?!\\+8)\\+?[7|8]{1}[\\(|\\s]?\\d{3}[\\)|\\s]?\\s?\\d{3}[\\-|\\s]?\\d{2}[\\-|\\s]?\\d{2}\\\"\""
r\"(\\\"(?:\\+7|7|8)(?:\\((?=\\+)|\\s)?\\d{3}(?:\\)|\\s|\\)\\s)?\\d{3}(?:-|\\s)?\\d{2}(?:-|\\s)?\\d{2}\\\")\" "
r'\"[+]?[78]\\(\\d{3}\\)\\d{3}[- ]\\d{2}[- ]\\d{2}\"|\"[+]?[78]\\d{10}\"|\"[+]?[78] \\d{3} \\d{3} \\d{2} \\d{2}\"'"
r\"\"\"^[\"']?(?:(?:\\+?7|8)(?:\\s*\\(\\d{3}\\)\\s*|\\s*\\d{3}\\s*)\\d{3}(?:[ -]?\\d{2}){2}|8\\d{10})[\"']?$\"\"\""
r'(?<=\").(?:(?:\\+7|7|8)\\(\\d{3}\\)\\s?\\d{3}[- ]?\\d{2}[- ]?\\d{2}|8\\d{10}|\\+7\\s\\d{3}\\s\\d{3}\\s\\d{2}\\s\\d{2})(?=\")'"
r'\\\"8 ?\\(?\\d{3}\\)? ?\\d{3}[- ]?\\d{2}[- ]?\\d{2}[- ]?\\\"|\\\"\\+?7 ?\\(?\\d{3}\\)? ?\\d{3}[- ]?\\d{2}[- ]?\\d{2}[- ]?\\\"' "
r'\"(?:\\+7|7|8)\\(\\d{3}\\)(?:\\s?\\d{3}-\\d{2}-\\d{2}|\\d{3} \\d{2} \\d{2})\"|\"8\\d{10}\"|\"\\+7 \\d{3} \\d{3} \\d{2} \\d{2}\"'",
r\"(?<!\\d)(?:(?:\\+|)8\\d{10}|(?:\\+?\\s*7|(?<!\\+)8)[\\s-]*\\(?\\d{3}\\)?[\\s-]*\\d{3}[\\s-]*\\d{2}[\\s-]*\\d{2})(?!([\\s-]*\\d)\" "
r\"(\\+?7\\(\\d{3}\\)\\s?\\d{3}[\\s\\d-]{3}[\\s\\d-]{3}[^\\s])|(\\d{11})|([\\+\\d\\s]{16})|([^+]8\\(\\d{3}\\)\\s?\\d{3}[\\s\\d-]{3}[\\s\\d-]{3})\""
r'[\\\"\\+]?[78]\\(\\d{3}\\)\\s?\\d{3}[\\s-]?\\d{2}[\\s-]?\\d{2}[\\\"]?|[\\\"\\+]?[78]\\d{10}[\\\"]?|[\\\"\\+]?[78]\\s\\d{3}\\s\\d{3}\\s\\d{2}\\s\\d{2}'"
r'[^+](?:(?:\\+7)|8|7)(?:\\s|-){0,1}(?:(?:\\(\\d\\d\\d\\))|(?:\\d\\d\\d))(?:\\s|-){0,1}\\d\\d\\d(?:\\s|-){0,1}\\d\\d(?:\\s|-){0,1}\\d\\d)(?!(?:\\s|-)*\\d)' "
r'[\\\"][+]?[7|8][)|(|\\s|-]*[8|9][)|(|\\s|-]*\\d[)|(\\s|-]*\\d[)|\\s|-]*\\d[)|\\s|-]*\\d[)|\\s|-]*\\d[)|\\s|-]*\\d[)|\\s|-]*\\d[)|\\s|-]*\\d[)|\\s|-]*[\\\"]'"
r'([\\\"][+]{0,1}7[\\s\\(]{0,1}\\d{3}[\\s\\)]{0,1}\\s{0,1}[\\d]{3}[\\s-]{0,1}[\\d]{2}[-\\s]{0,1}[\\d]{2}[\\\"]|[\\\"]8[\\s\\(]{0,1}\\d{3}[\\s\\)]{0,1}\\s{0,1}[\\d]{3}[\\s-]{0,1}[\\d]{2}[-\\s]{0,1}[\\d]{2}[\\\"])'"
r'\\\"7\\(\\d{3}\\)\\s?\\d{3}-\\d{2}-\\d{2}\\\"|\\\"7\\(\\d{3}\\)\\d{3}-\\d{2}-\\d{2}\\\"|\\\"\\+7\\(\\d{3}\\)\\d{3}-\\d{2}-\\d{2}\\\"|\\\"8\\(\\d{3}\\)\\d{3}\\s\\d{2}\\s\\d{2}\\\"|\\\"8\\d{10}\\\"|\\\"\\+7\\s\\d{3}\\s\\d{3}\\s\\d{2}\\s\\d{2}\\\"'"
```

dadaist2001	248
alexk-git	221
MariiaBiktasheva	171
Vadiman01	158
SergeyIlin322	153

В следующих сериях

next semester...

Advanced python

- **OOP**, classes
- Decorators
- Iterators & generators
- Web scraping

Tools development

- Parallel programming
- Profiling, performance
- Open source
- SQL

Домашка на каникулы

Домашка на каникулы

- Отдых

Домашка на каникулы

- **Отдых**

Основы **Git**

- LearningGitBranching
- Hexlet Git course

Основы **python**

- Stepik python course (BI)
- Stepik python course (BEEGEEK)

YouTube

- Хитрый питон
- Moscow Python (конференции и подкаст)
- Диджитализируй
- Python Russian
- Python Clinic
- Все доклады Григория Петрова на MoscowPython
- Т. Хирьянов, Алгоритмы и структуры данных на Python 3