Lecture 13

Wrap up

02.12.2023

Course program

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

next semester...

Advanced python

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

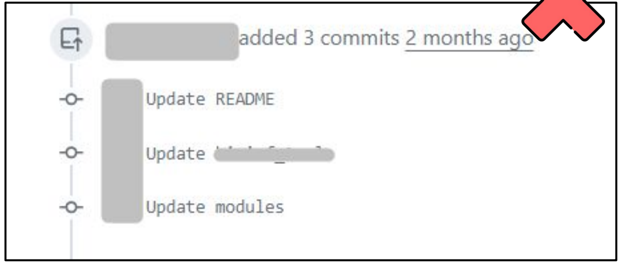
Tools development

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

ДЗ 2 - 6

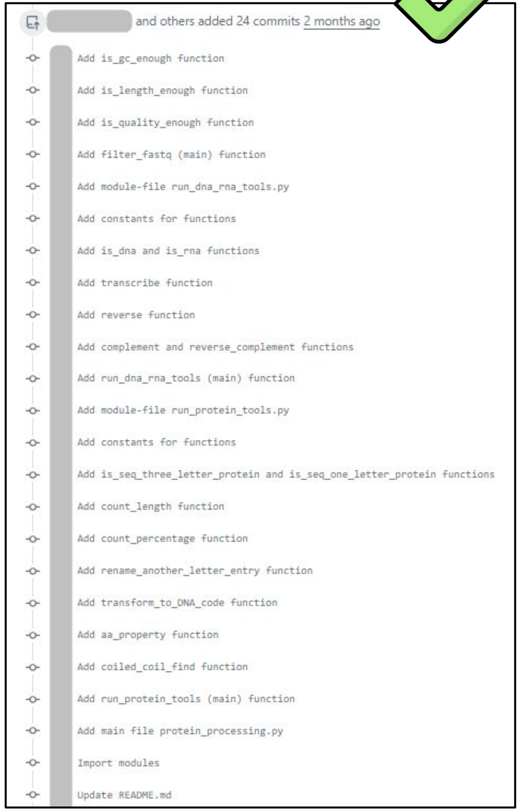
Коммиты:

- Use **imperative** mood
- Start with **Capital** letter
- Line size **< 50** symbols
- Be **informative**
- Do not end with a period
- Use the body if you need



[redacted] added 3 commits 2 months ago

```
Update README
Update [redacted]
Update modules
```

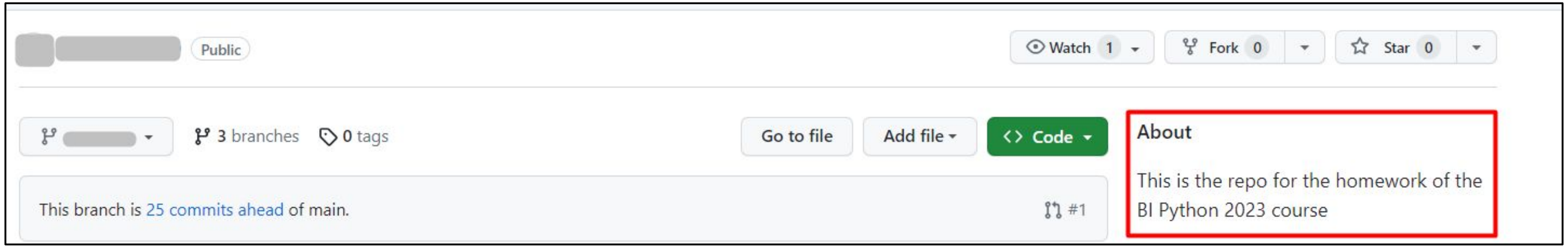


[redacted] and others added 24 commits 2 months ago

```
Add is_gc_enough function
Add is_length_enough function
Add is_quality_enough function
Add filter_fastq (main) function
Add module-file run_dna_rna_tools.py
Add constants for functions
Add is_dna and is_rna functions
Add transcribe function
Add reverse function
Add complement and reverse_complement functions
Add run_dna_rna_tools (main) function
Add module-file run_protein_tools.py
Add constants for functions
Add is_seq_three_letter_protein and is_seq_one_letter_protein functions
Add count_length function
Add count_percentage function
Add rename_another_letter_entry function
Add transform_to_DNA_code function
Add aa_property function
Add coiled_coil_find function
Add run_protein_tools (main) function
Add main file protein_processing.py
Import modules
Update README.md
```

README

✓ Описание репо


Public

Watch 1 | Fork 0 | Star 0

3 branches | 0 tags | Go to file | Add file | Code

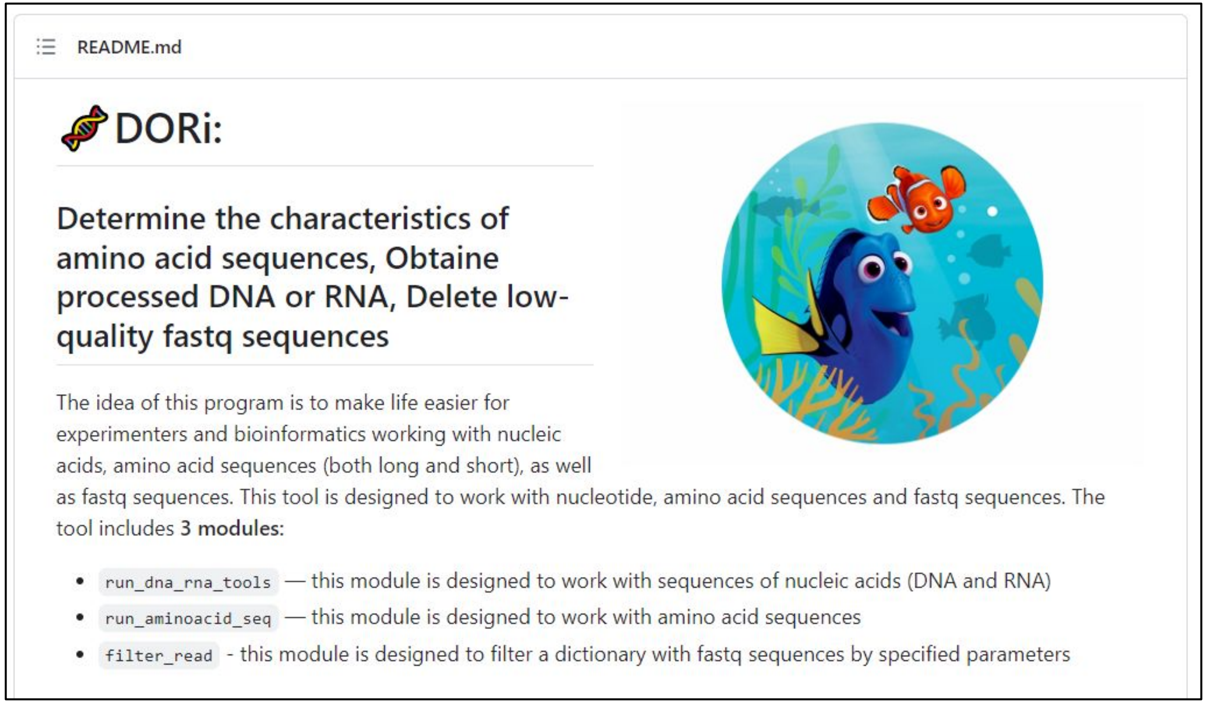
This branch is 25 commits ahead of main. #1

About

This is the repo for the homework of the BI Python 2023 course

README

- ✓ Описание репо
- ✓ Небольшое красочное интро



README.md

🧬 DORi:

Determine the characteristics of amino acid sequences, Obtaine processed DNA or RNA, Delete low-quality fastq sequences

The idea of this program is to make life easier for experimenters and bioinformatics working with nucleic acids, amino acid sequences (both long and short), as well as fastq sequences. This tool is designed to work with nucleotide, amino acid sequences and fastq sequences. The tool includes **3 modules**:

- `run_dna_rna_tools` — this module is designed to work with sequences of nucleic acids (DNA and RNA)
- `run_aminoacid_seq` — this module is designed to work with amino acid sequences
- `filter_read` - this module is designed to filter a dictionary with fastq sequences by specified parameters

README

- ✓ Описание репо
- ✓ Небольшое красочное интро
- ✓ Содержание

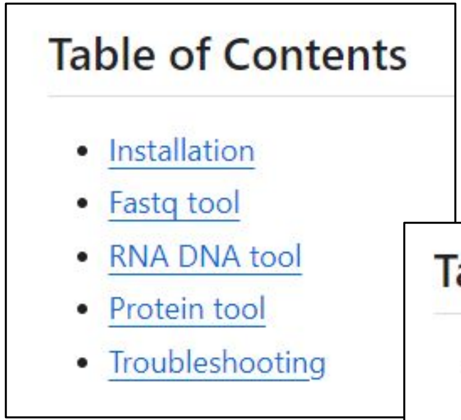


Table of Contents

- Installation
- Fastq tool
- RNA DNA tool
- Protein tool
- Troubleshooting

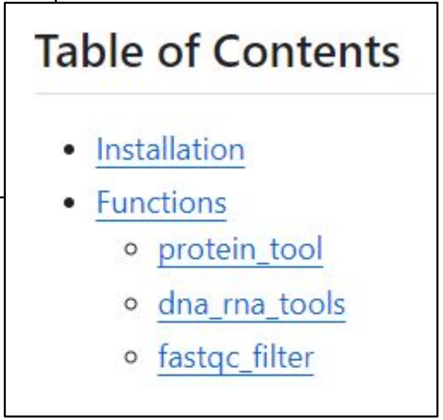
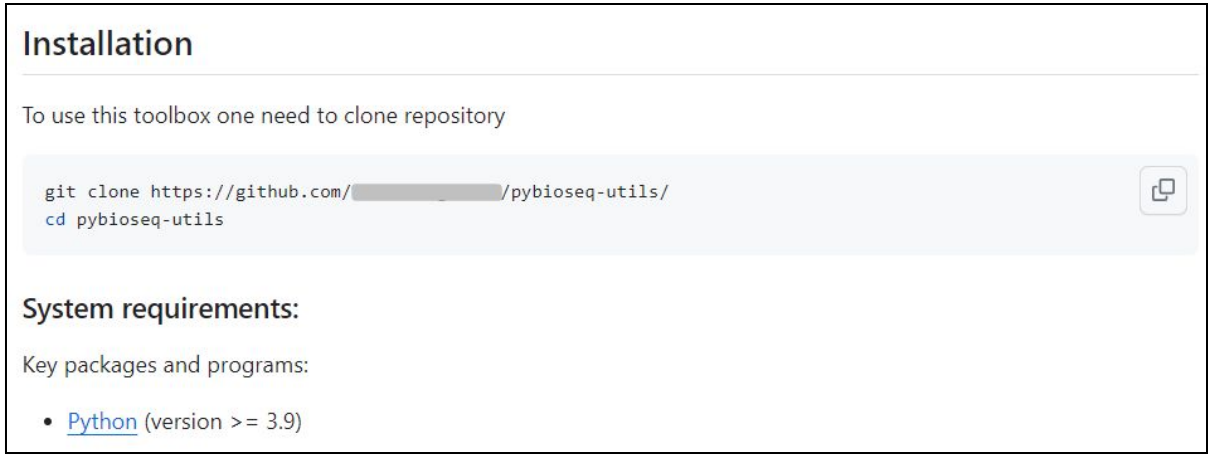


Table of Contents

- Installation
- Functions
    - protein_tool
    - dna_rna_tools
    - fastqc_filter

README

- ✓ Описание репо
- ✓ Небольшое красочное интро
- ✓ Содержание
- ✓ Установка



Installation

To use this toolbox one need to clone repository

```
git clone https://github.com/[redacted]/pybioseq-utils/
cd pybioseq-utils
```

System requirements:

Key packages and programs:

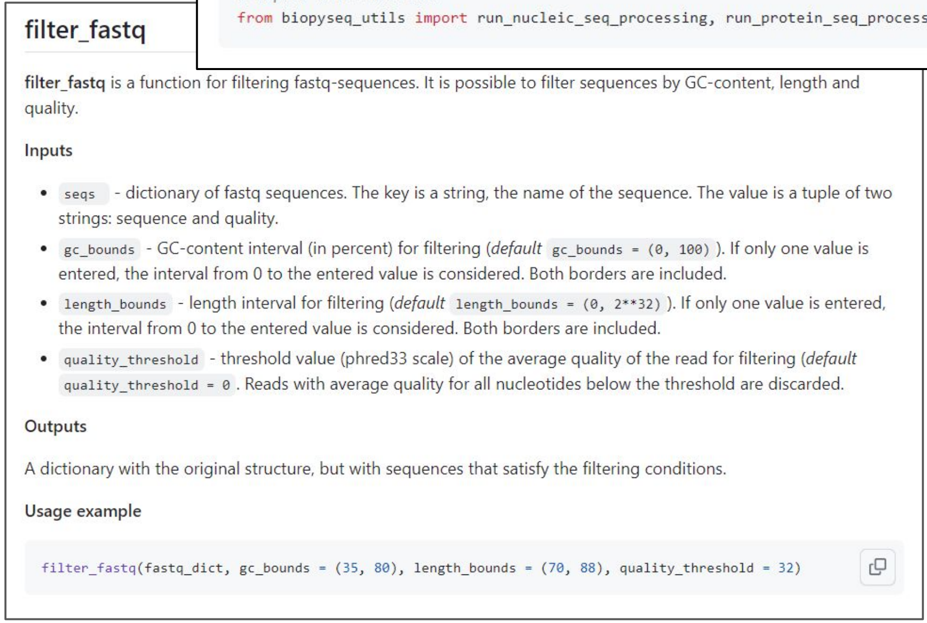
- Python (version >= 3.9)

```
System requirements

 -  OS
 -  Python/R/... version
```

```
Installation

 -  git clone <URL>
 -  cd <dir>
```

README

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



filter_fastq

filter_fastq is a function for filtering fastq-sequences. It is possible to filter sequences by GC-content, length and quality.

Inputs

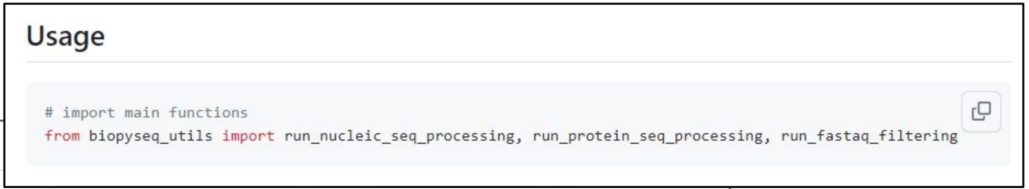
- `seqs` - dictionary of fastq sequences. The key is a string, the name of the sequence. The value is a tuple of two strings: sequence and quality.
- `gc_bounds` - GC-content interval (in percent) for filtering (*default* `gc_bounds = (0, 100)`). If only one value is entered, the interval from 0 to the entered value is considered. Both borders are included.
- `length_bounds` - length interval for filtering (*default* `length_bounds = (0, 2**32)`). If only one value is entered, the interval from 0 to the entered value is considered. Both borders are included.
- `quality_threshold` - threshold value (phred33 scale) of the average quality of the read for filtering (*default* `quality_threshold = 0` . Reads with average quality for all nucleotides below the threshold are discarded.

Outputs

A dictionary with the original structure, but with sequences that satisfy the filtering conditions.

Usage example

```
filter_fastq(fastq_dict, gc_bounds = (35, 80), length_bounds = (70, 88), quality_threshold = 32)
```

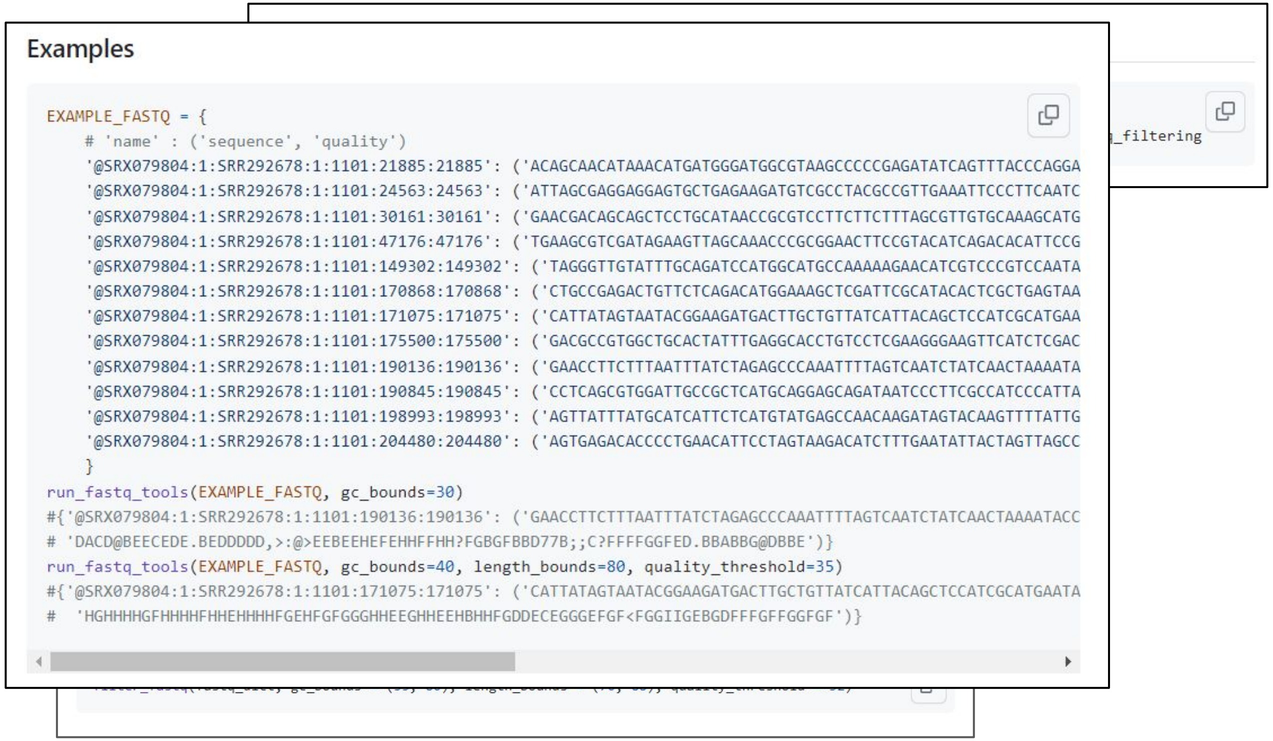


Usage

```
# import main functions
from biopyseq_utils import run_nucleic_seq_processing, run_protein_seq_processing, run_fastaq_filtering
```

README

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

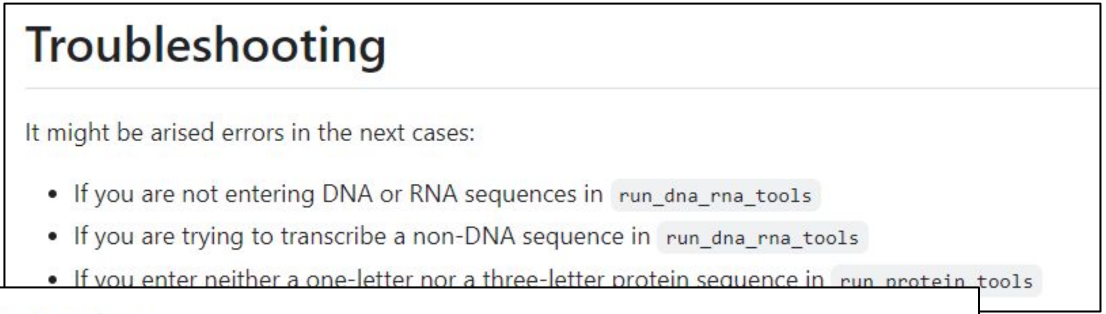


Examples

```
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

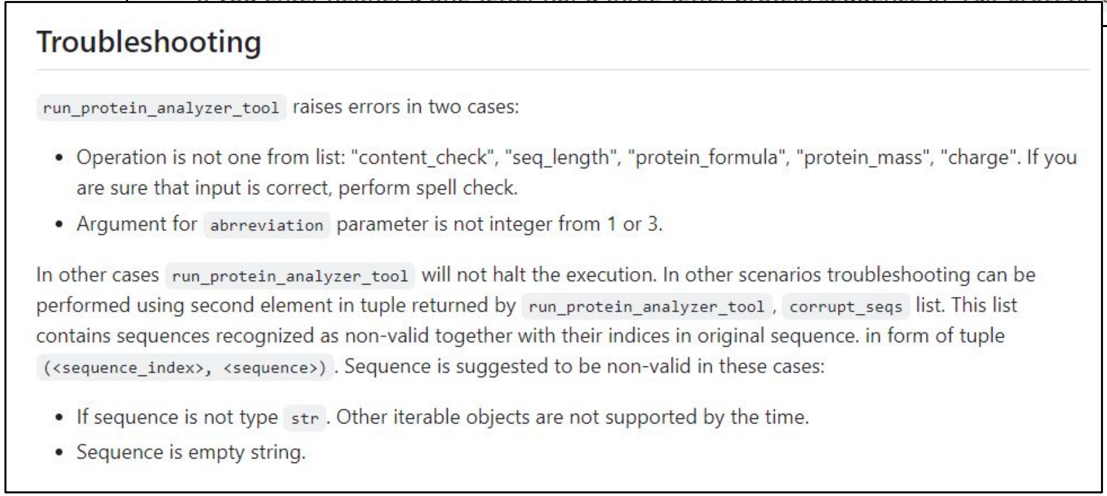
- ✓ Описание репо
- ✓ Небольшое красочное интро
- ✓ Содержание
- ✓ Установка
- ✓ Использование и описание
- ✓ Траблшутинг



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:

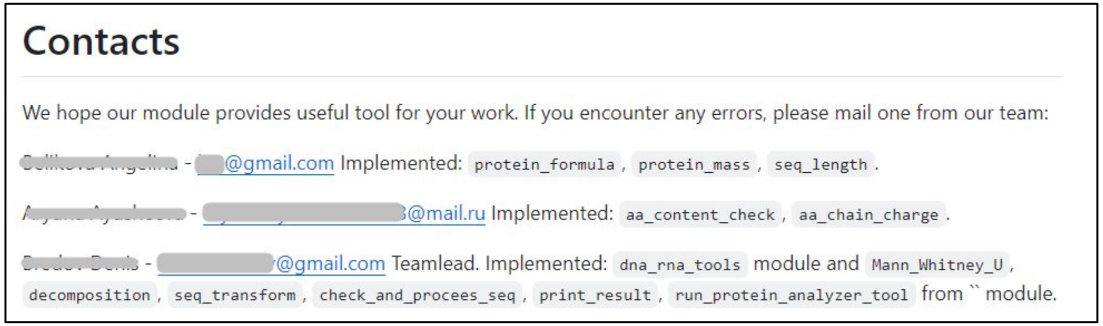
- Operation is not one from list: "content_check", "seq_length", "protein_formula", "protein_mass", "charge". If you are sure that input is correct, perform spell check.
- Argument for `abrreviation` parameter is not integer from 1 or 3.

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



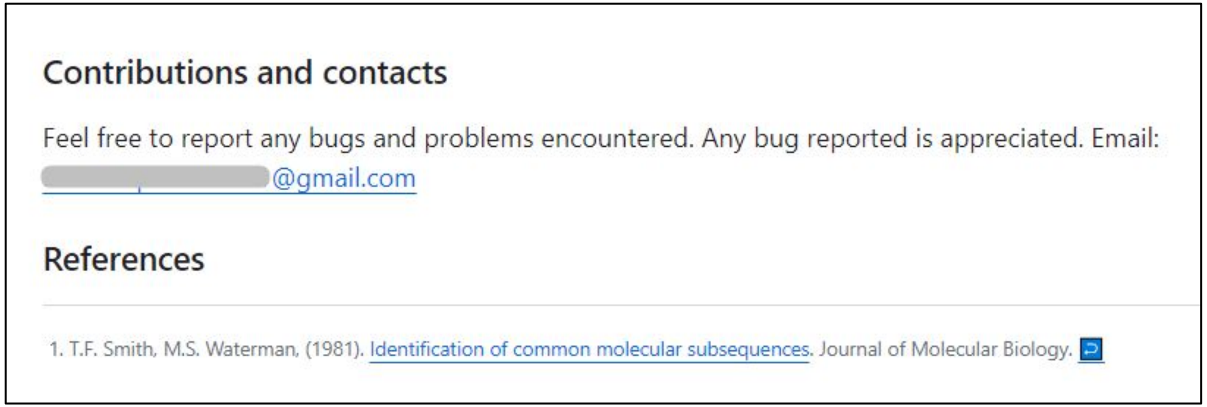
Contacts

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

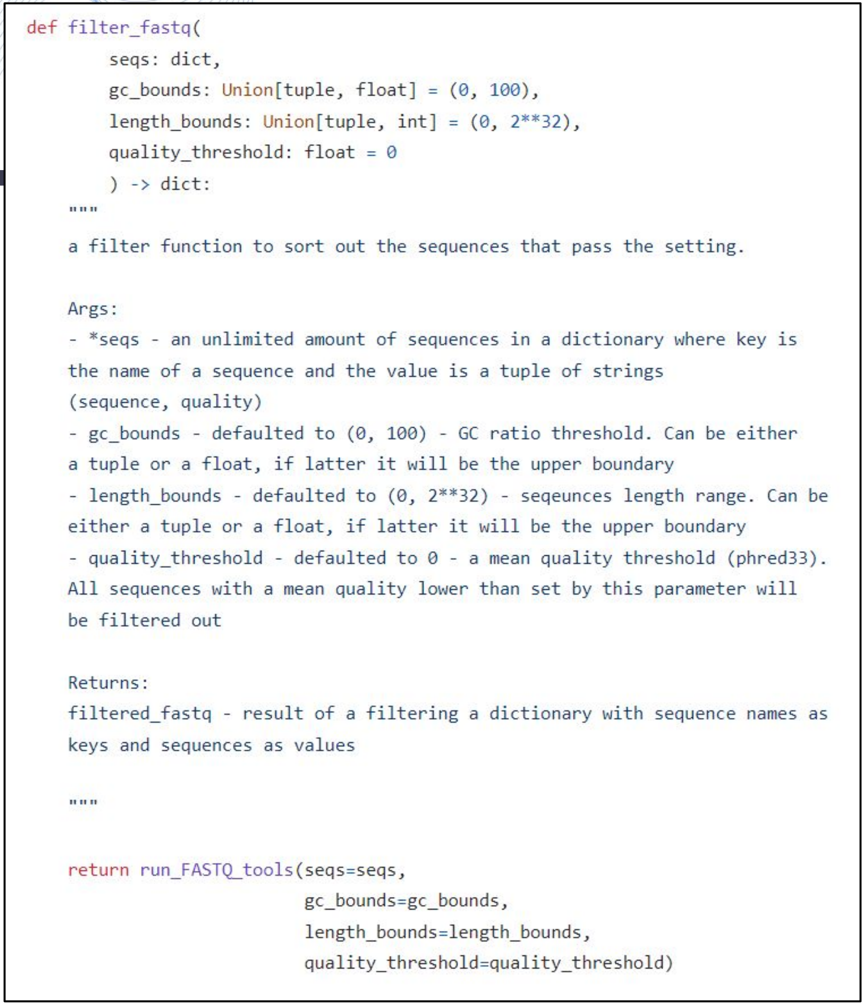
Feel free to report any bugs and problems encountered. Any bug reported is appreciated. Email: [redacted]@gmail.com

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



```
def filter_fastq(
        seqs: dict,
        gc_bounds: Union[tuple, float] = (0, 100),
        length_bounds: Union[tuple, int] = (0, 2**32),
        quality_threshold: float = 0
        ) -> dict:
    """
    a filter function to sort out the sequences that pass the setting.

    Args:
    - *seqs - an unlimited amount of sequences in a dictionary where key is
    the name of a sequence and the value is a tuple of strings
    (sequence, quality)
    - gc_bounds - defaulted to (0, 100) - GC ratio threshold. Can be either
    a tuple or a float, if latter it will be the upper boundary
    - length_bounds - defaulted to (0, 2**32) - seqeunces length range. Can be
    either a tuple or a float, if latter it will be the upper boundary
    - quality_threshold - defaulted to 0 - a mean quality threshold (phred33).
    All sequences with a mean quality lower than set by this parameter will
    be filtered out

    Returns:
    filtered_fastq - result of a filtering a dictionary with sequence names as
    keys and sequences as values

    """

    return run_FASTQ_tools(seqs=seqs,
                           gc_bounds=gc_bounds,
                           length_bounds=length_bounds,
                           quality_threshold=quality_threshold)
```

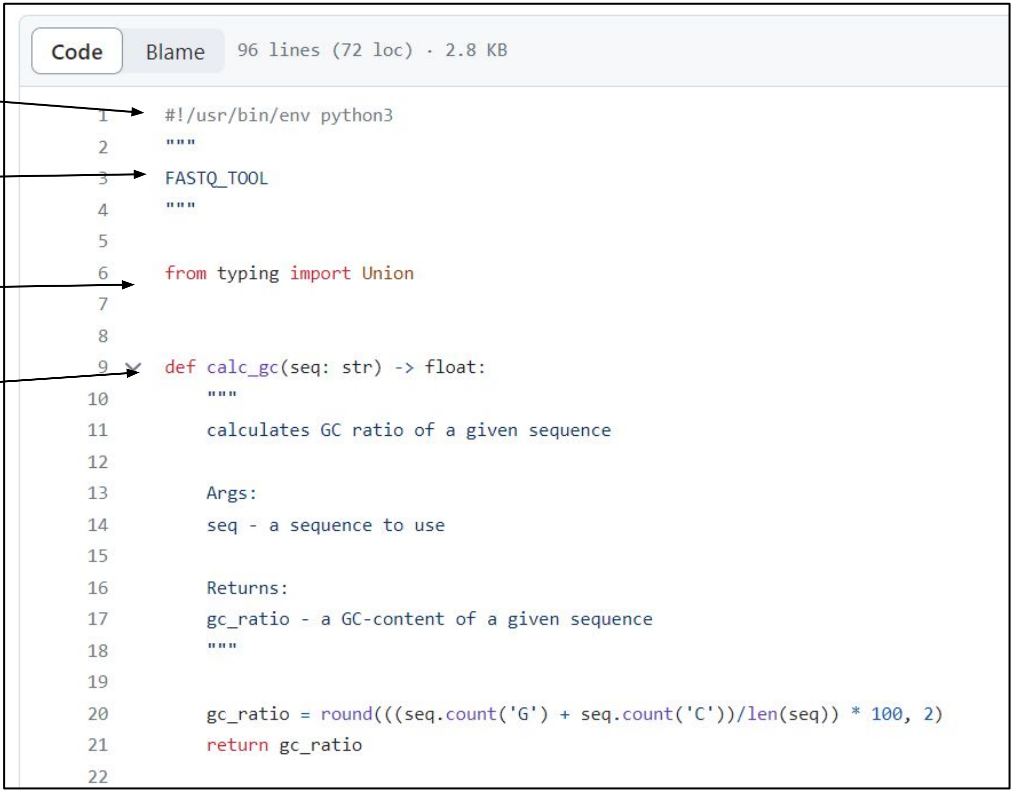
Немного о коде

Шебанг

Докстринга модуля

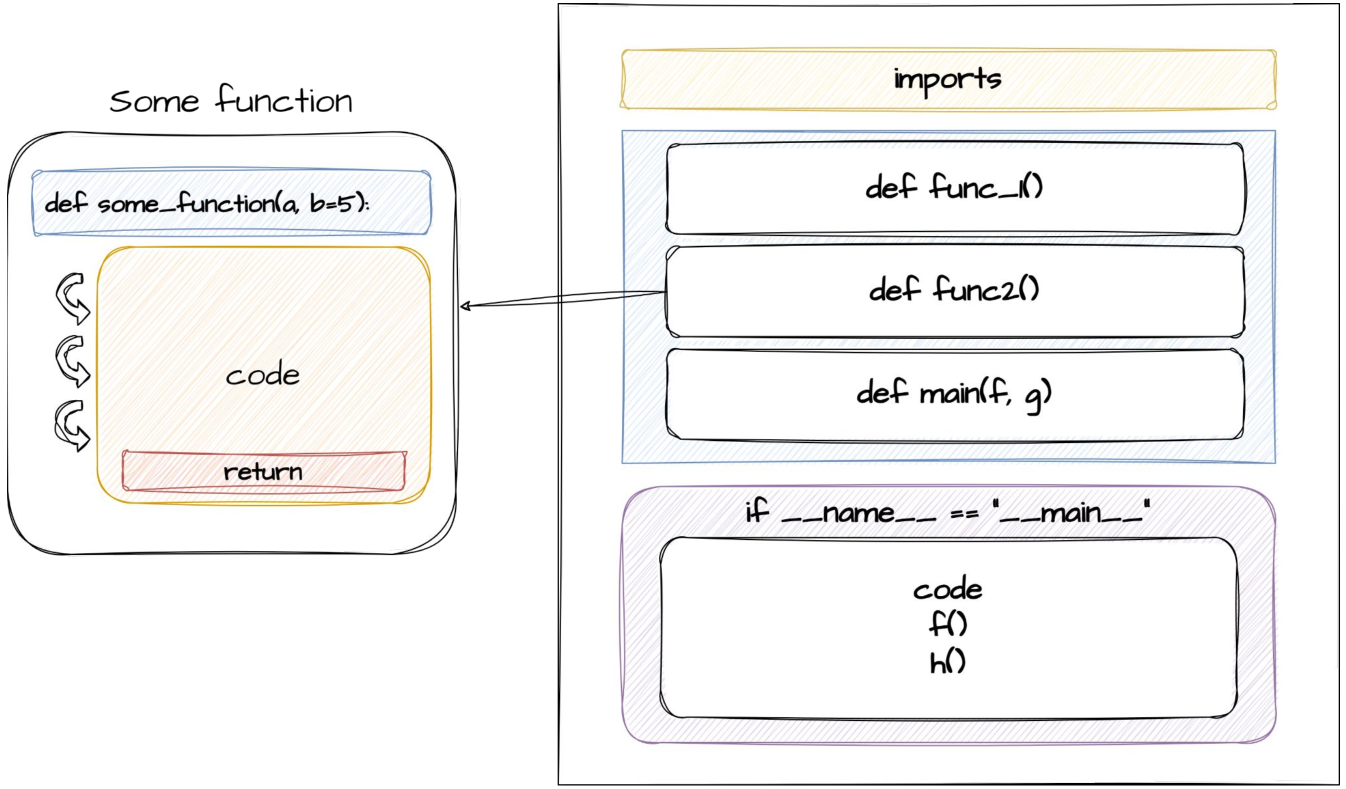
Импорты

Функции



```python
#!/usr/bin/env python3
"""
FASTQ_TOOL
"""

from typing import Union


def calc_gc(seq: str) -> float:
    """
    calculates GC ratio of a given sequence

    Args:
    seq - a sequence to use

    Returns:
    gc_ratio - a GC-content of a given sequence
    """

    gc_ratio = round(((seq.count('G') + seq.count('C'))/len(seq)) * 100, 2)
    return gc_ratio

```

Немного о коде



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

ДЗ 7

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/python3.12/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/python3.12/site-packages/pandas/core/frame.py
```

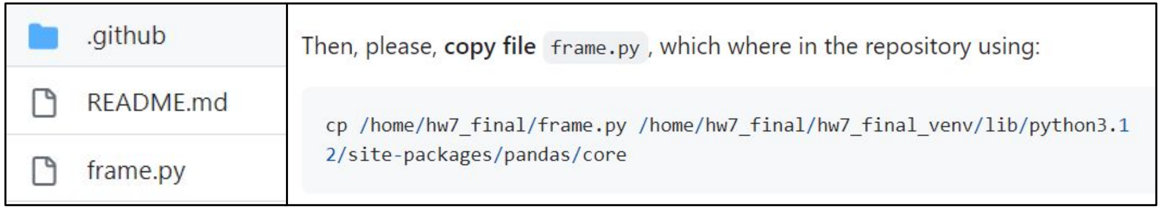
“Comment two following lines”

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

nano / vim

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

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/python3.12/site-packages/pandas/core/frame.py
```

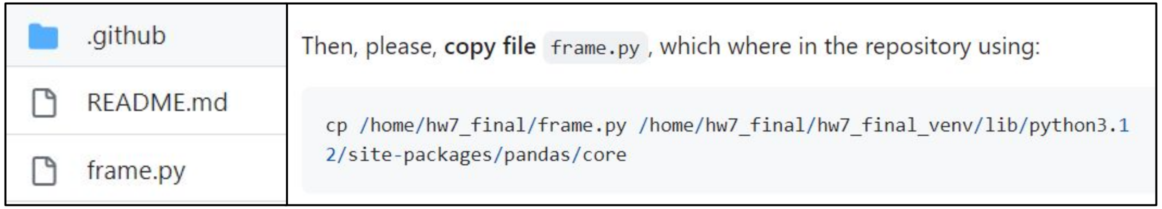
“Comment two following lines”

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

nano / vim

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

Патч



.github
README.md
frame.py

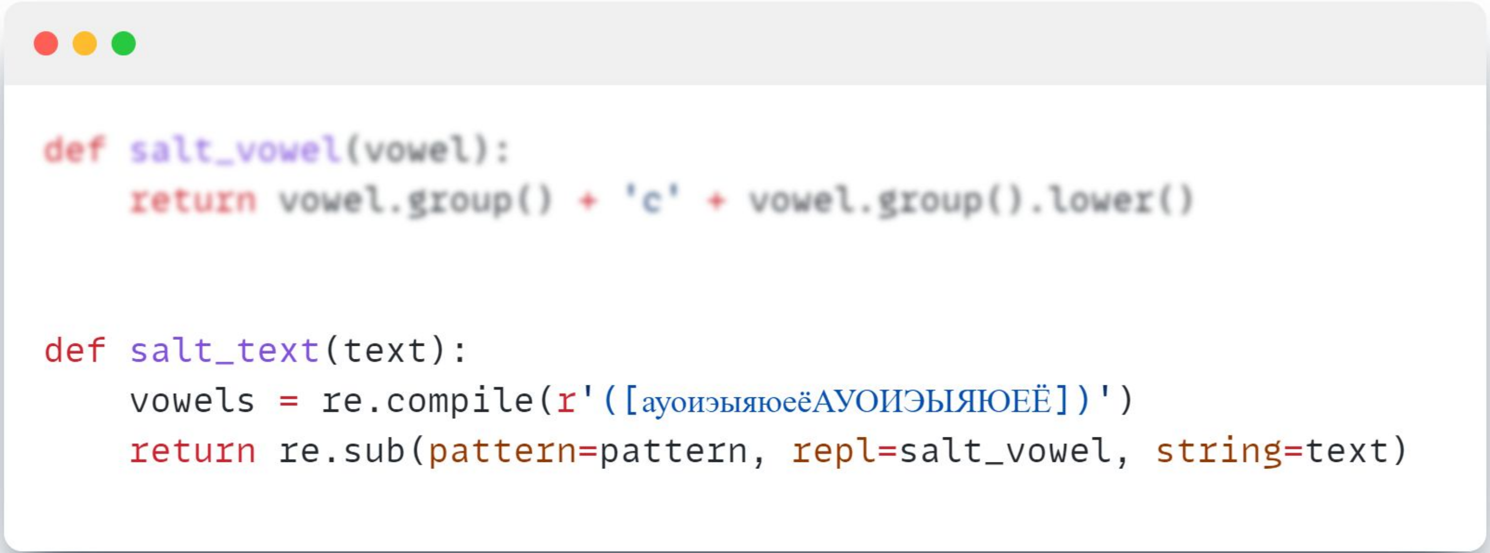
Then, please, **copy file** `frame.py`, which where in the repository using:

```
cp /home/hw7_final/frame.py /home/hw7_final/hw7_final_venv/lib/python3.1
2/site-packages/pandas/core
```

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

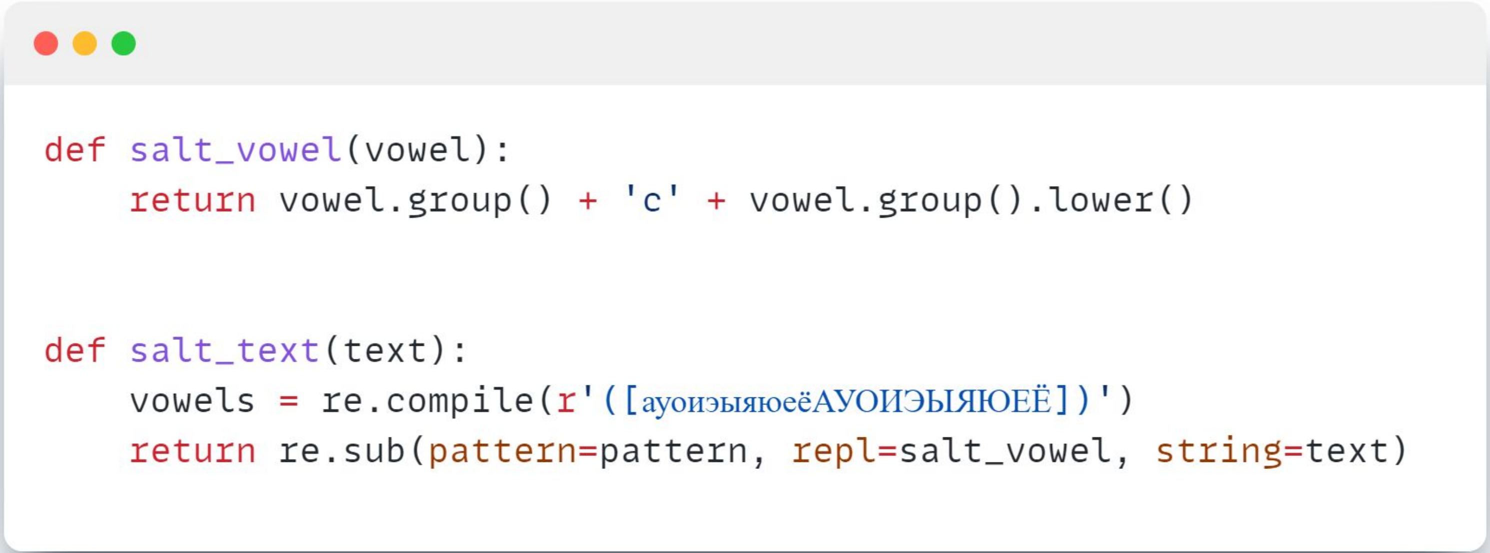
ДЗ 8

Перевод на соленый язык



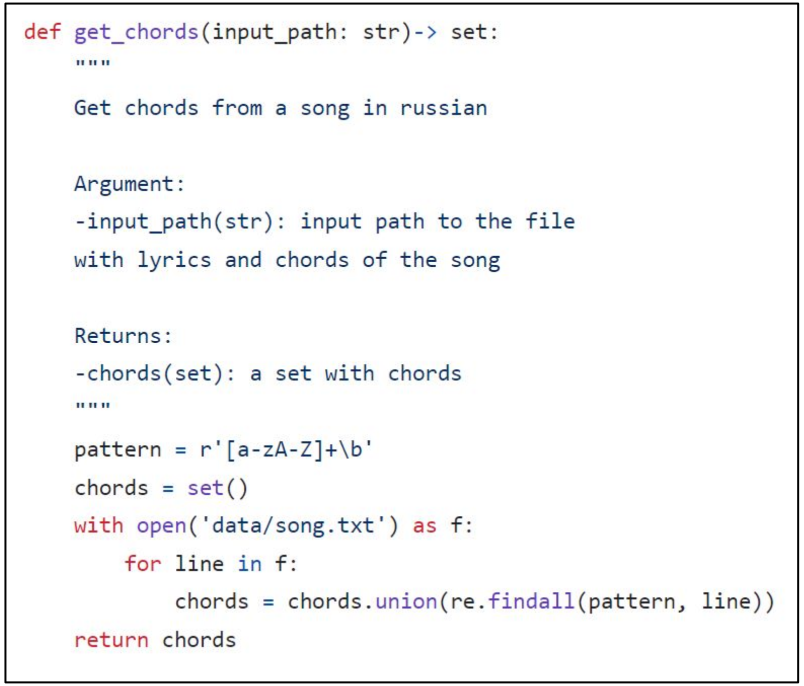
```
def salt_vowel(vowel):
    return vowel.group() + 'c' + vowel.group().lower()


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

Перевод на соленый язык



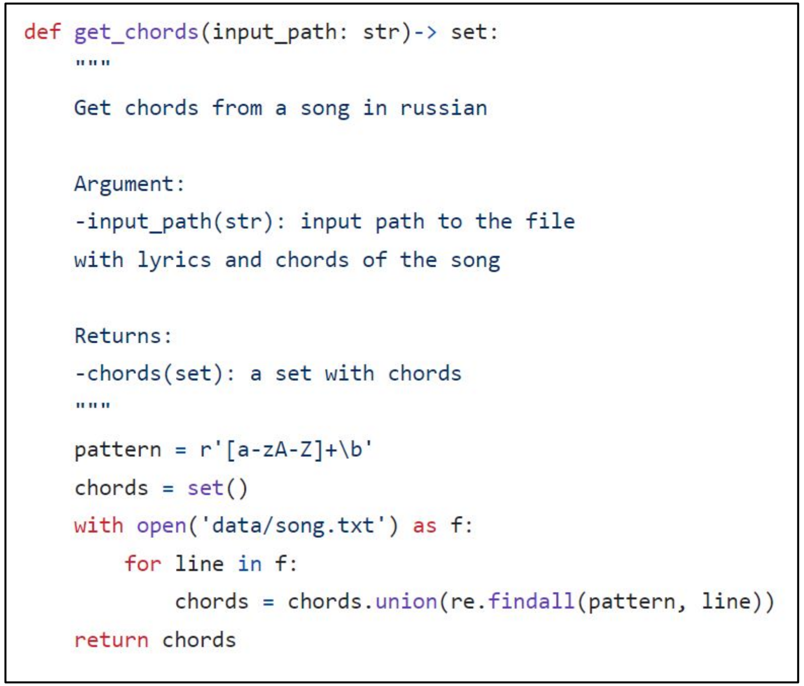
```
def salt_vowel(vowel):
    return vowel.group() + 'c' + vowel.group().lower()


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

Аккорды



```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
```

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][m]?'
```

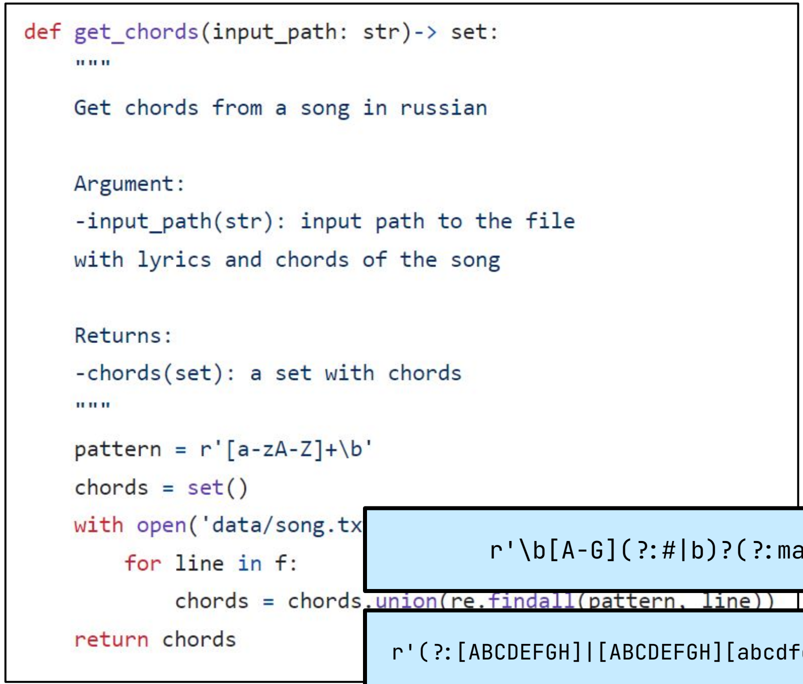
```
r"[A-Z][a-z]?\b"
```

```
r'[CDEFGAB][m7]?'
```

```
r"[AC-H]+#?m?7?"
```

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

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.tx
        for line in f:
            chords = chords.union(re.findall(pattern, line))
    return chords
```

```
r'[A-Z][m]?'
```

```
r"[A-Z][a-z]?\b"
```

```
r'[CDEFGAB][m7]?'
```

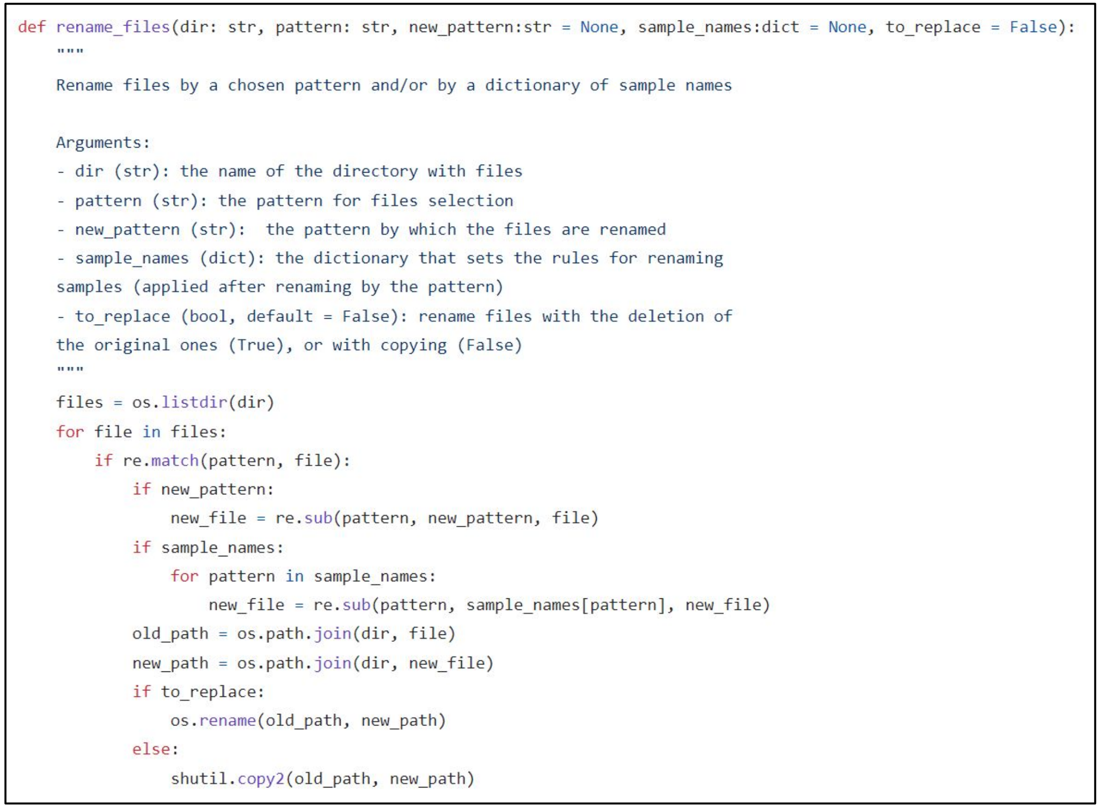
```
r"[AC-H]+#?m?7?"
```

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

```
r'\b[A-G](?:#|b)?(?:maj|m|min|sus|dim|aug)?(?:[0-9]?(?:#|b)?)?(?:\/[A-G](?:#|b)?)?\b'
```

```
r'(?:[ABCDEFGH]|[ABCDEFGH][abcdfegh])(?:\#|)(?:[abcdfegh]|)(?:dim|aug|sus|maj|)(?:\d|)(?:\#\d|)(?:\s|\n|\t|\S)'
```

Переименовалка



```python
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)
```

Квиз!

Я вам показываю регулярку

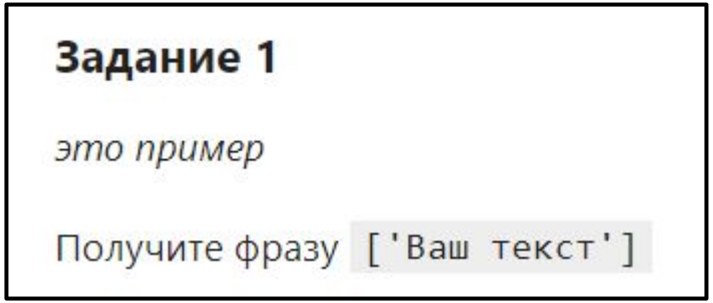
Вы угадываете что это было за задание

Квиз!

```
r'Ваш текст'
```

Квиз!

```
r'Ваш текст'
```



Задание 1

это пример

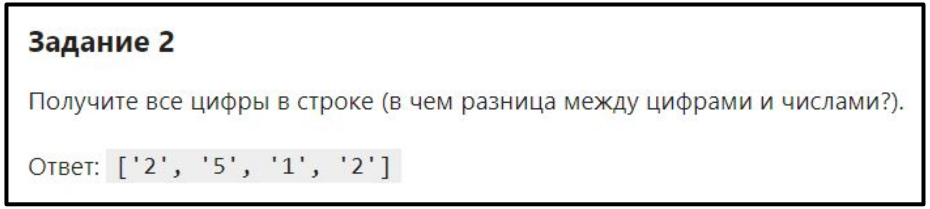
Получите фразу `['Ваш текст']`

Квиз!

```
r'\d'
```

Квиз!

```
r'\d'
```



Задание 2

Получите все цифры в строке (в чем разница между цифрами и числами?).

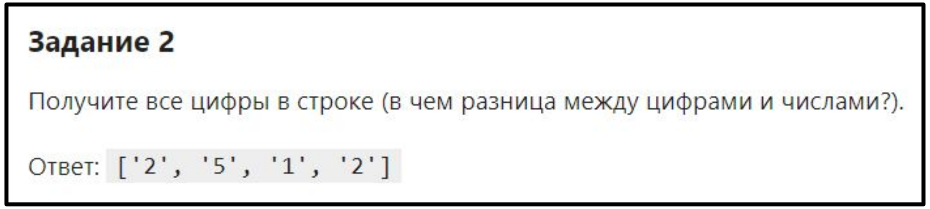
Ответ: `['2', '5', '1', '2']`

Квиз!

```
r'[0-9]'
```

Квиз!

```
r'[0-9]'
```



Задание 2

Получите все цифры в строке (в чем разница между цифрами и числами?).

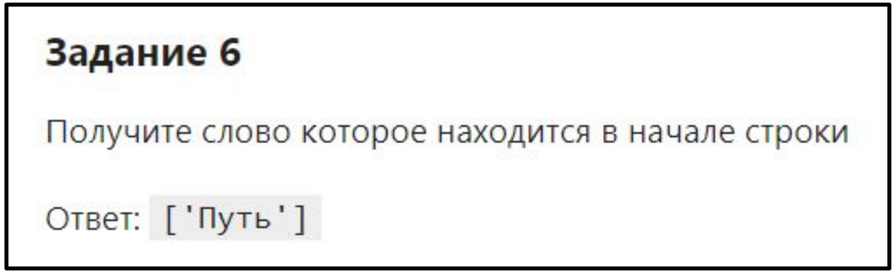
Ответ: `['2', '5', '1', '2']`

Квиз!

```
r'^[а-яА-ЯёЁ]+'
```

Квиз!

```
r'^[а-яА-ЯёЁ]+'
```



Задание 6

Получите слово которое находится в начале строки

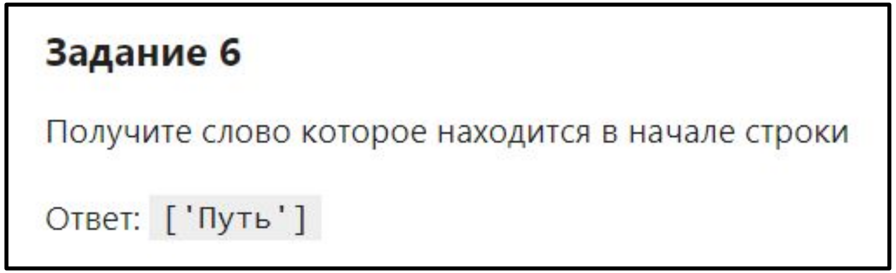
Ответ: `['Путь']`

Квиз!

```
r'^\w+'
```

Квиз!

```
r'^\w+'
```



Задание 6

Получите слово которое находится в начале строки

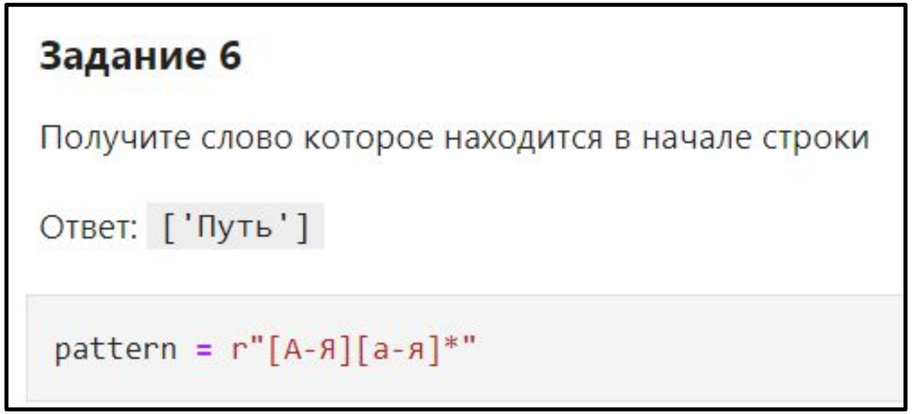
Ответ: `['Путь']`

Квиз!

```
r"[А-Я][а-я]*"
```

Квиз!

```
r"[А-Я][а-я]*"
```



Задание 6

Получите слово которое находится в начале строки

Ответ: `['Путь']`

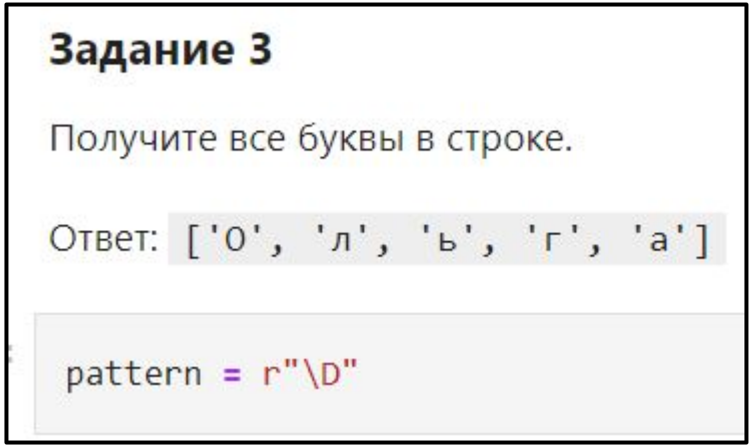
```
pattern = r"[А-Я][а-я]*"
```

Квиз!

```
r"\D"
```

Квиз!

```
r"\D"
```



Задание 3

Получите все буквы в строке.

Ответ: `['О', 'л', 'ь', 'г', 'а']`

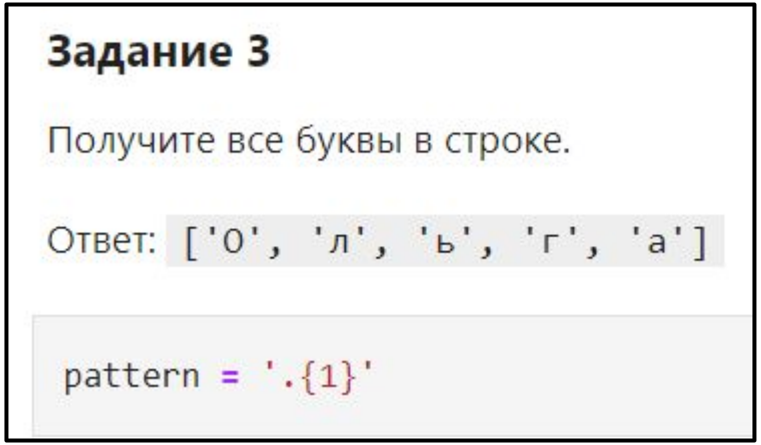
```
pattern = r"\D"
```

Квиз!

```
r'.{1}'
```

Квиз!

```
r'.{1}'
```



Задание 3

Получите все буквы в строке.

Ответ: `['О', 'л', 'ь', 'г', 'а']`

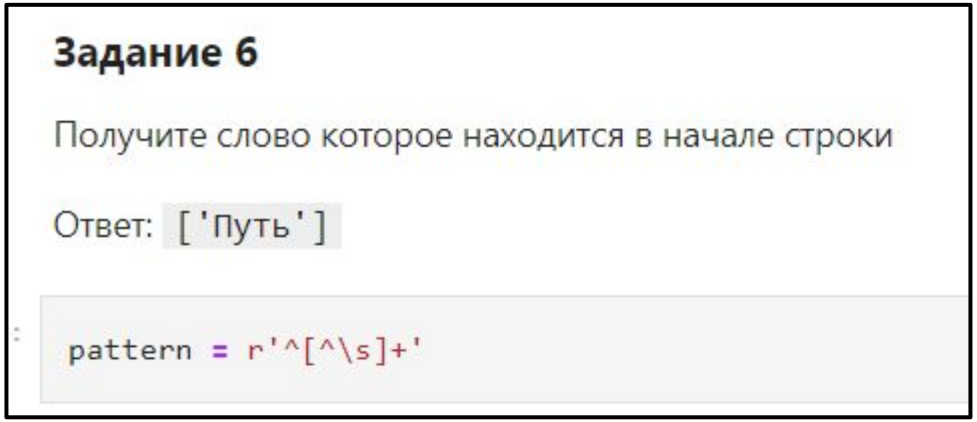
```
pattern = '.{1}'
```

Квиз!

```
r'^[^\s]+'
```

Квиз!

```
r'^[^\s]+'
```



Задание 6

Получите слово которое находится в начале строки

Ответ: `['Путь']`

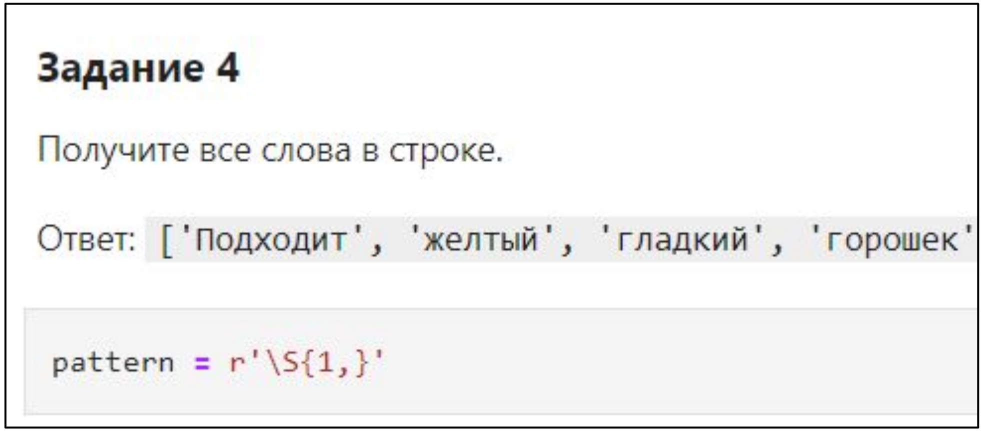
```
pattern = r'^[^\s]+'
```

Квиз!

```
r'\S{1,}'
```

Квиз!

```
r'\S{1,}'
```



Задание 4

Получите все слова в строке.

Ответ: `['Подходит', 'желтый', 'гладкий', 'горошек'`

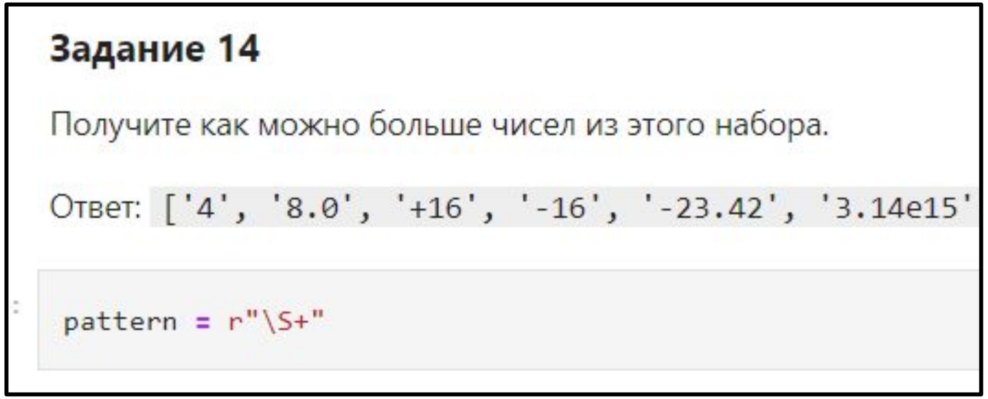
```
pattern = r'\S{1,}'
```

Квиз!

```
r'\S+'
```

Квиз!

```
r'\S+'
```



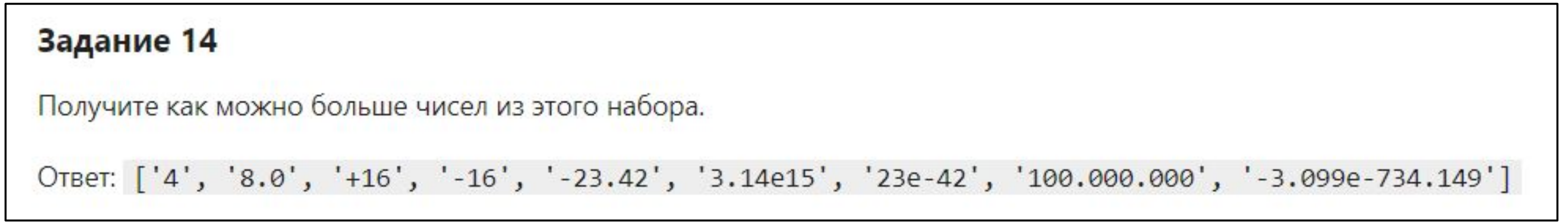
Задание 14

Получите как можно больше чисел из этого набора.

Ответ: `['4', '8.0', '+16', '-16', '-23.42', '3.14e15'`

```
pattern = r"\S+"
```

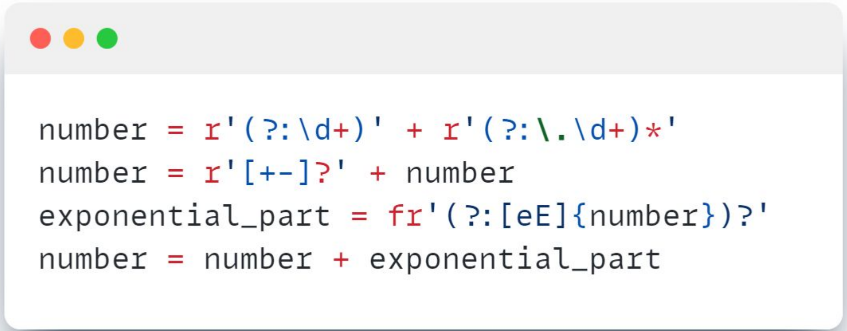
Еще пара слов



Задание 14

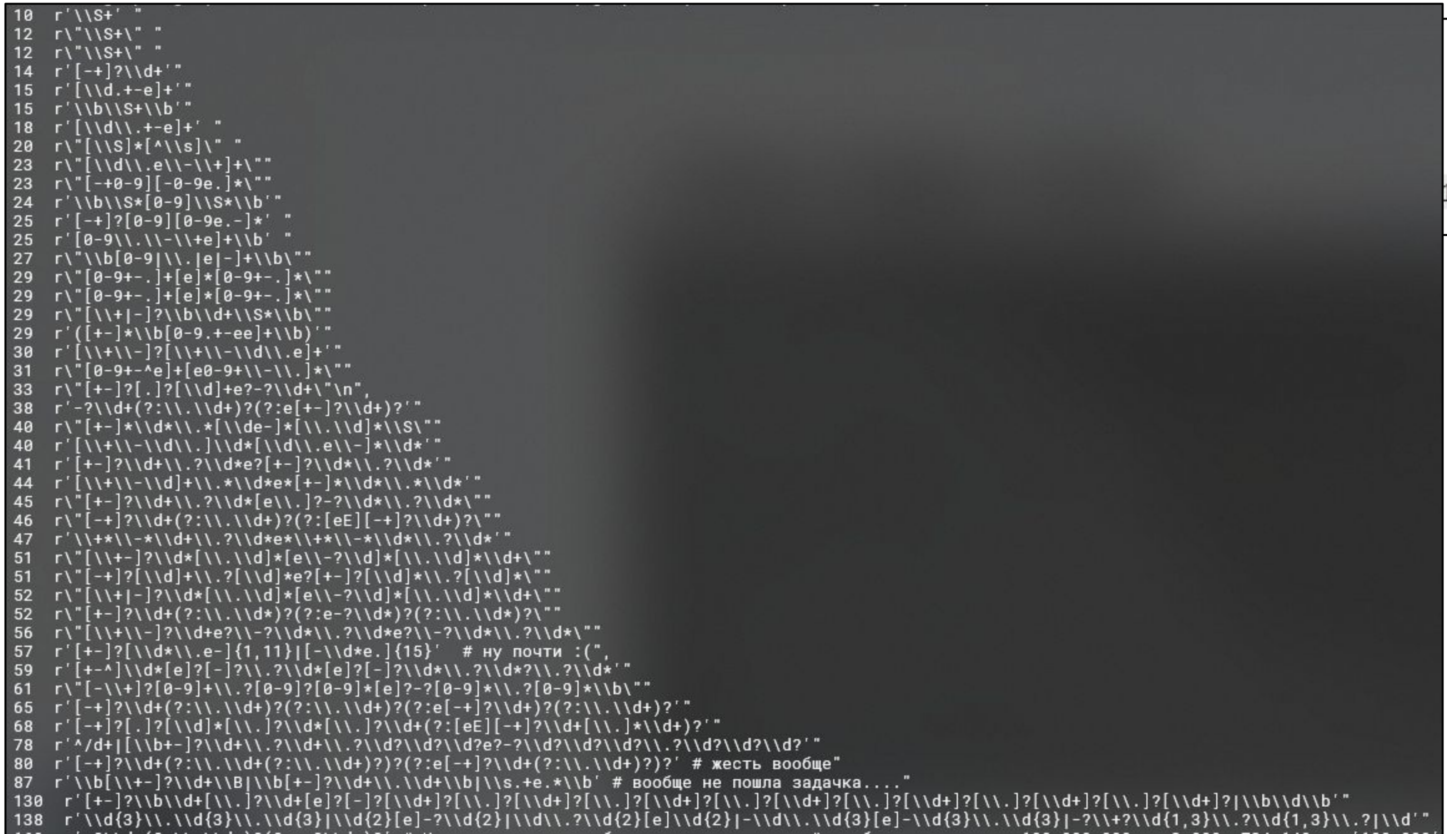
Получите как можно больше чисел из этого набора.

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



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

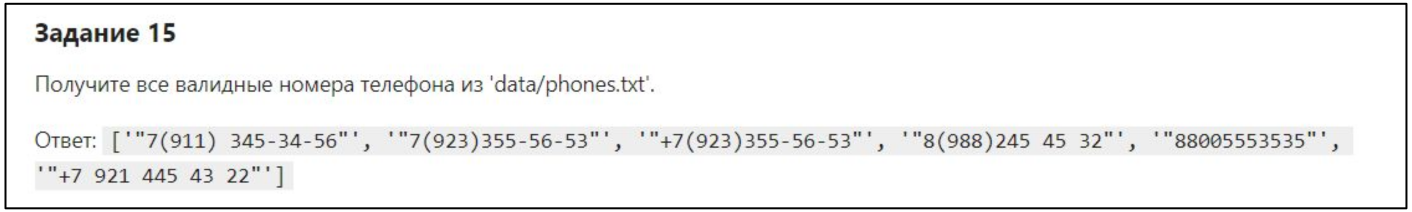
Еще пара слов



```
10  r'\\S+' "
12  r\"\\S+\" "
12  r\"\\S+\" "
14  r'[-+]?\\d+'"
15  r'[\\d.+-e]+'"
15  r'\\b\\S+\\b'"
18  r'[\\d\\.+-e]+' "
20  r\"[\\S]*[^\\s]\" "
23  r\"[\\d\\.e\\-\\+]+\""
23  r\"[-+0-9][-0-9e.]*\""
24  r'\\b\\S*[0-9]\\S*\\b'"
25  r'[-+]?[0-9][0-9e.-]*' "
25  r'[0-9\\.\\-\\+e]+\\b' "
27  r\"\\b[0-9|\\.|e|-]+\\b\""
29  r\"[0-9+-.]+[e]*[0-9+-.]*\""
29  r\"[0-9+-.]+[e]*[0-9+-.]*\""
29  r\"[\\+|-]?\\b\\d+\\S*\\b\""
29  r'([+-]*\\b[0-9.+-ee]+\\b)'"
30  r'[\\+\\-]?[\\+\\-\\d\\.e]+'"
31  r\"[0-9+-^e]+[e0-9+\\-\\.]*\""
33  r\"[+-]?[.]?[\\d]+e?-?\\d+\"\n",
38  r'-?\\d+(?:\\.\\d+)?(?:e[+-]?\\d+)?'"
40  r\"[+-]*\\d*\\.*[\\de-]*[\\.\\d]*\\S\""
40  r'[\\+\\-\\d\\.]\\d*[\\d\\.e\\-]*\\d*'"
41  r'[+-]?\\d+\\.?\\d*e?[+-]?\\d*\\.?\\d*'"
44  r'[\\+\\-\\d]+\\.*\\d*e*[+-]*\\d*\\.*\\d*'"
45  r\"[+-]?\\d+\\.?\\d*[e\\.]?-?\\d*\\.?\\d*\""
46  r\"[-+]?\\d+(?:\\.\\d+)?(?:[eE][-+]?\\d+)?\""
47  r'\\+*\\-*\\d+\\.?\\d*e*\\+*\\-*\\d*\\.?\\d*'"
51  r\"[\\+-]?\\d*[\\.\\d]*[e\\-?\\d]*[\\.\\d]*\\d+\""
51  r\"[-+]?[\\d]+\\.?[\\d]*e?[+-]?[\\d]*\\.?[\\d]*\""
52  r\"[\\+|-]?\\d*[\\.\\d]*[e\\-?\\d]*[\\.\\d]*\\d+\""
52  r\"[+-]?\\d+(?:\\.\\d*)?(?:e-?\\d*)?(?:\\.\\d*)?\""
56  r\"[\\+\\-]?\\d+e?\\-?\\d*\\.?\\d*e?\\-?\\d*\\.?\\d*\""
57  r'[+-]?[\\d*\\.e-]{1,11}|[-\\d*e.]{15}'  # ну почти :(",
59  r'[+-^]\\d*[e]?[-]?\\.?\\d*[e]?[-]?\\d*\\.?\\d*?\\.?\\d*'"
61  r\"[-\\+]?[0-9]+\\.?[0-9]?[0-9]*[e]?-?[0-9]*\\.?[0-9]*\\b\""
65  r'[-+]?\\d+(?:\\.\\d+)?(?:\\.\\d+)?(?:e[-+]?\\d+)?(?:\\.\\d+)?'"
68  r'[-+]?[.]?[\\d]*[\\.]?\\d*[\\.]?\\d+(?:[eE][-+]?\\d+[\\.]*\\d+)?'"
78  r'^/d+|[\\b+-]?\\d+\\.?\\d+\\.?\\d?\\d?\\d?e?-?\\d?\\d?\\d?\\.?\\d?\\d?\\d?'"
80  r'[-+]?\\d+(?:\\.\\d+(?:\\.\\d+)?)?(?:e[-+]?\\d+(?:\\.\\d+)?)?' # жесть вообще"
87  r'\\b[\\+-]?\\d+\\B|\\b[+-]?\\d+\\.\\d+\\b|\\s.+e.*\\b' # вообще не пошла задачка...."
130  r'[+-]?\\b\\d+[\\.]?\\d+[e]?[-]?[\\d+]?[\\.]?[\\d+]?[\\.]?[\\d+]?[\\.]?[\\d+]?[\\.]?[\\d+]?[\\.]?[\\d+]?[\\.]?[\\d+]?|\\b\\d\\b'"
138  r'\\d{3}\\.\\d{3}\\.\\d{3}|\\d{2}[e]-?\\d{2}|\\d\\.?\\d{2}[e]\\d{2}|-\\d\\.\\d{3}[e]-\\d{3}\\.\\d{3}|-?\\+?\\d{1,3}\\.?\\d{1,3}\\.?|\\d'"
```

149']

Еще пара слов

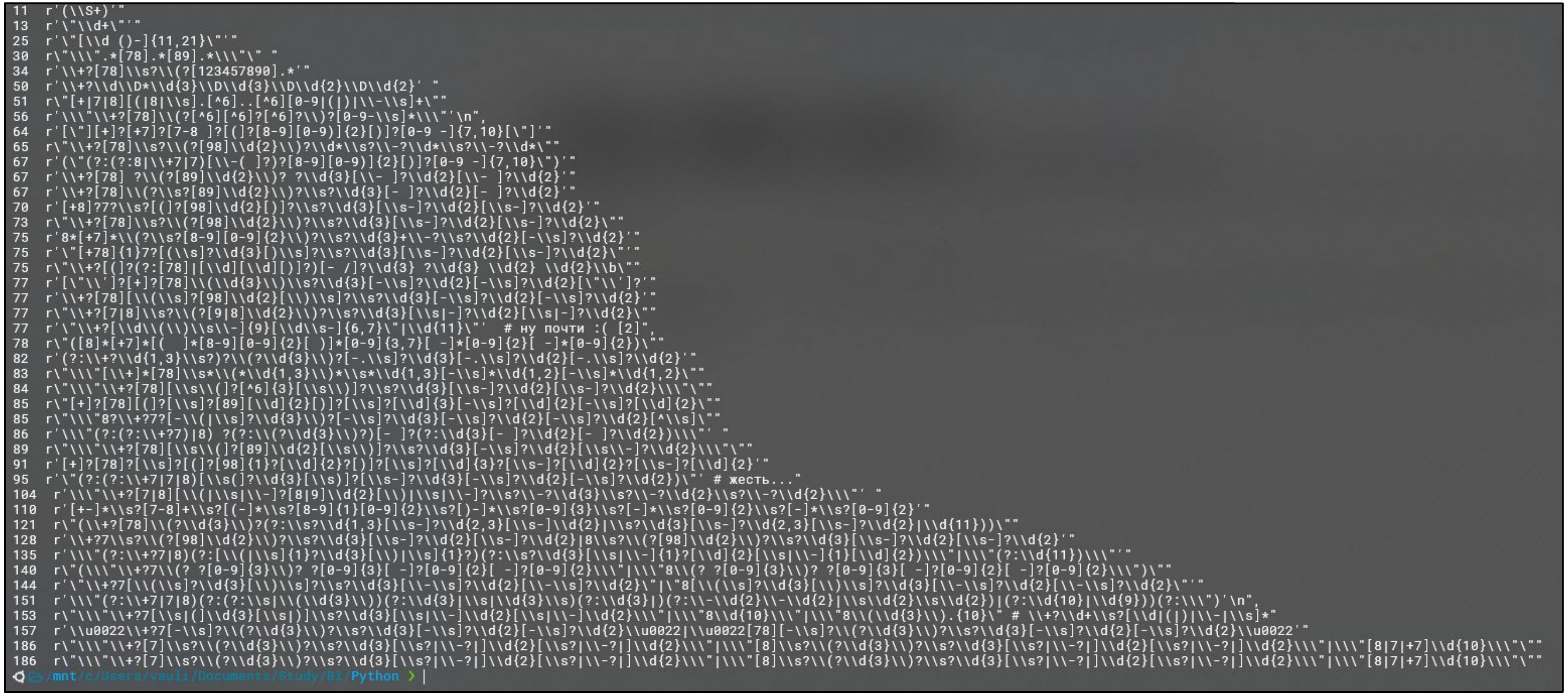


Задание 15

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"']`

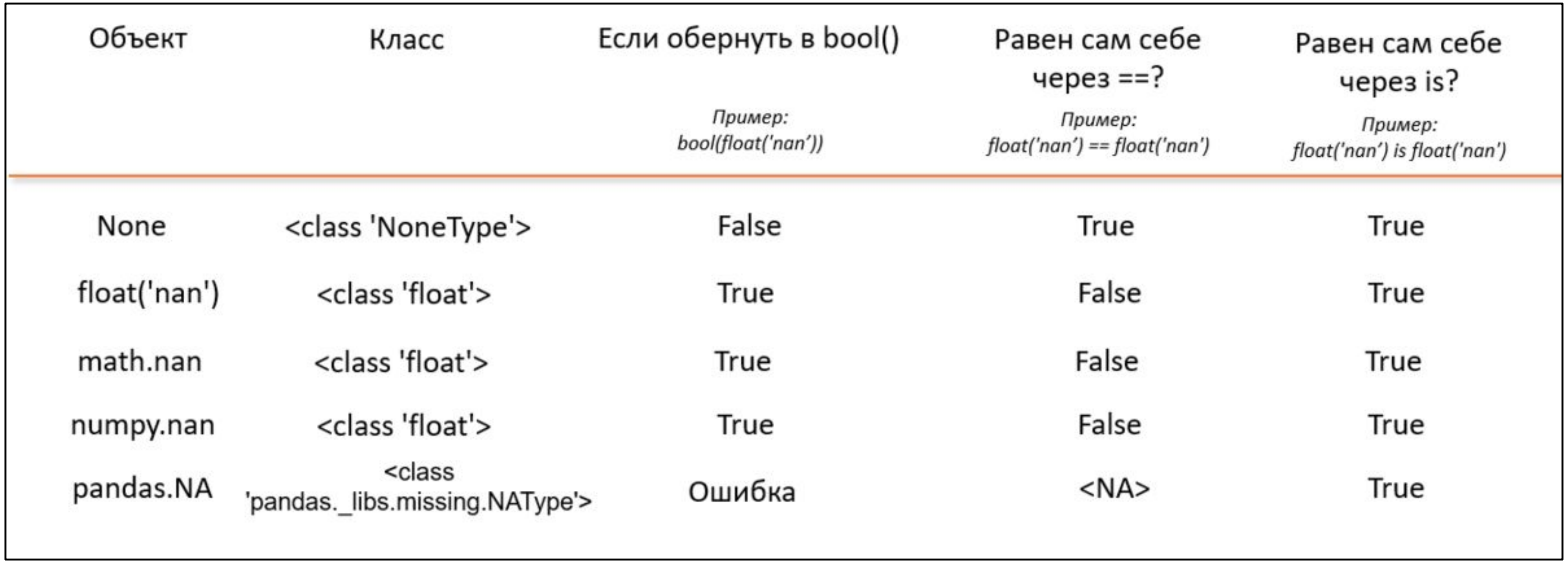
Еще пара слов



```
11  r'(\\S+)'"
13  r'\"\\d+\"'"
25  r'\"[\\d ()-]{11,21}\"'"
30  r\"\\\".*[78].*[89].*\\\"\" "
34  r'\\+?[78]\\s?\\(?[123457890].*'"
50  r'\\+?\\d\\D*\\d{3}\\D\\d{3}\\D\\d{2}\\D\\d{2}' "
51  r\"[+|7|8][(|8|\\s].[^6]..[^6][0-9|(|)|\\-\\s]+\""
56  r'\\\"\\+?[78]\\(?[^6][^6]?[^6]?\\)?[0-9-\\s]*\\\"'\n",
64  r'[\"][+]?[+7]?[7-8 ]?[(]?[8-9][0-9)]{2}[)]?[0-9 -]{7,10}[\"]'"
65  r\"\\+?[78]\\s?\\(?[98]\\d{2}\\)?\\d*\\s?\\-?\\d*\\s?\\-?\\d*\""
67  r'(\"(?:(?:8|\\+7|7)[\\-( ]?)?[8-9][0-9)]{2}[)]?[0-9 -]{7,10}\")'"
67  r'\\+?[78] ?\\(?[89]\\d{2}\\)? ?\\d{3}[\\- ]?\\d{2}[\\- ]?\\d{2}'"
67  r'\\+?[78]\\(?\\s?[89]\\d{2}\\)?\\s?\\d{3}[- ]?\\d{2}[- ]?\\d{2}'"
70  r'[+8]?7?\\s?[(]?[98]\\d{2}[)]?\\s?\\d{3}[\\s-]?\\d{2}[\\s-]?\\d{2}'"
73  r\"\\+?[78]\\s?\\(?[98]\\d{2}\\)?\\s?\\d{3}[\\s-]?\\d{2}[\\s-]?\\d{2}\""
75  r'8*[+7]*\\(?\\s?[8-9][0-9]{2}\\)?\\s?\\d{3}+\\-?\\s?\\d{2}[-\\s]?\\d{2}'"
75  r'\"[+78]{1}7?[(\\s]?\\d{3}[)\\s]?\\s?\\d{3}[\\s-]?\\d{2}[\\s-]?\\d{2}\"'"
75  r\"\\+?[(]?(?:[78]|[\\d][\\d][)]?)[- /]?\\d{3} ?\\d{3} \\d{2} \\d{2}\\b\""
77  r'[\"\\']?[+]?[78]\\(\\d{3}\\)\\s?\\d{3}[-\\s]?\\d{2}[-\\s]?\\d{2}[\"\\']?'"
77  r'\\+?[78][\\(\\s]?[98]\\d{2}[\\)\\s]?\\s?\\d{3}[-\\s]?\\d{2}[-\\s]?\\d{2}'"
77  r\"\\+?[7|8]\\s?\\(?[9|8]\\d{2}\\)?\\s?\\d{3}[\\s|-]?\\d{2}[\\s|-]?\\d{2}\""
77  r'\"\\+?[\\d\\(\\)\\s\\-]{9}[\\d\\s-]{6,7}\"|\\d{11}\"'  # ну почти :( [2]",
78  r\"([8]*[+7]*[(  ]*[8-9][0-9]{2}[ )]*[0-9]{3,7}[ -]*[0-9]{2}[ -]*[0-9]{2})\""
82  r'(?:\\+?\\d{1,3}\\s?)?\\(?\\d{3}\\)?[-.\\s]?\\d{3}[-.\\s]?\\d{2}[-.\\s]?\\d{2}'"
83  r\"\\\"[\\+]*[78]\\s*\\(*\\d{1,3}\\)*\\s*\\d{1,3}[-\\s]*\\d{1,2}[-\\s]*\\d{1,2}\""
84  r\"\\\"\\+?[78][\\s\\(]?[^6]{3}[\\s\\)]?\\s?\\d{3}[\\s-]?\\d{2}[\\s-]?\\d{2}\\\"\""
85  r\"[+]?[78][(]?[\\s]?[89][\\d]{2}[)]?[\\s]?[\\d]{3}[-\\s]?[\\d]{2}[-\\s]?[\\d]{2}\""
85  r\"\\\"8?\\+?7?[-\\(|\\s]?\\d{3}\\)?[-\\s]?\\d{3}[-\\s]?\\d{2}[-\\s]?\\d{2}[^\\s]\""
86  r'\\\"(?:(?:\\+?7)|8) ?(?:\\(?\\d{3}\\)?)[- ]?(?:\\d{3}[- ]?\\d{2}[- ]?\\d{2})\\\"' "
89  r\"\\\"\\+?[78][\\s\\(]?[89]\\d{2}[\\s\\)]?\\s?\\d{3}[-\\s]?\\d{2}[\\s\\-]?\\d{2}\\\"\""
91  r'[+]?[78]?[\\s]?[(]?[98]{1}?[\\d]{2}?[)]?[\\s]?[\\d]{3}?[\\s-]?[\\d]{2}?[\\s-]?[\\d]{2}'"
95  r'\"(?:(?:\\+7|7|8)[\\s(]?\\d{3}[\\s)]?[\\s-]?\\d{3}[-\\s]?\\d{2}[-\\s]?\\d{2})\"' # жесть..."
104  r'\\\"\\+?[7|8][\\(|\\s|\\-]?[8|9]\\d{2}[\\)|\\s|\\-]?\\s?\\-?\\d{3}\\s?\\-?\\d{2}\\s?\\-?\\d{2}\\\"' "
110  r'[+-]*\\s?[7-8]+\\s?[(-]*\\s?[8-9]{1}[0-9]{2}\\s?[)-]*\\s?[0-9]{3}\\s?[-]*\\s?[0-9]{2}\\s?[-]*\\s?[0-9]{2}'"
121  r\"(\\+?[78]\\(?\\d{3}\\)?(?:\\s?\\d{1,3}[\\s-]?\\d{2,3}[\\s-]\\d{2}|\\s?\\d{3}[\\s-]?\\d{2,3}[\\s-]?\\d{2}|\\d{11}))\""
128  r'\\+?7\\s?\\(?[98]\\d{2}\\)?\\s?\\d{3}[\\s-]?\\d{2}[\\s-]?\\d{2}|8\\s?\\(?[98]\\d{2}\\)?\\s?\\d{3}[\\s-]?\\d{2}[\\s-]?\\d{2}'"
135  r'\\\"(?:\\+?7|8)(?:[\\(|\\s]{1}?\\d{3}[\\)|\\s]{1}?)(?:\\s?\\d{3}[\\s|\\-]{1}?[\\d]{2}[\\s|\\-]{1}[\\d]{2})\\\"|\\\"(?:\\d{11})\\\"'"
140  r\"(\\\"\\+?7\\(? ?[0-9]{3}\\)? ?[0-9]{3}[ -]?[0-9]{2}[ -]?[0-9]{2}\\\"|\\\"8\\(? ?[0-9]{3}\\)? ?[0-9]{3}[ -]?[0-9]{2}[ -]?[0-9]{2}\\\")\""
144  r'\"\\+?7[\\(\\s]?\\d{3}[\\)\\s]?\\s?\\d{3}[\\-\\s]?\\d{2}[\\-\\s]?\\d{2}\"|\"8[\\(\\s]?\\d{3}[\\)\\s]?\\d{3}[\\-\\s]?\\d{2}[\\-\\s]?\\d{2}\"'"
151  r'\\\"(?:\\+7|7|8)(?:(?:\\s|\\(\\d{3}\\))(?:\\d{3}|\\s|\\d{3}\\s)(?:\\d{3}|)(?:\\-\\d{2}\\-\\d{2}|\\s\\d{2}\\s\\d{2})|(?:\\d{10}|\\d{9}))(?:\\\")'\n",
153  r\"\\\"\\+?7[\\s|(]\\d{3}[\\s|)]\\s?\\d{3}[\\s|\\-]\\d{2}[\\s|\\-]\\d{2}\\\"|\\\"8\\d{10}\\\"|\\\"8\\(\\d{3}\\).{10}\" # \\+?\\d+\\s?[\\d|(|)|\\-|\\s]*"
157  r'\\u0022\\+?7[-\\s]?\\(?\\d{3}\\)?\\s?\\d{3}[-\\s]?\\d{2}[-\\s]?\\d{2}\\u0022|\\u0022[78][-\\s]?\\(?\\d{3}\\)?\\s?\\d{3}[-\\s]?\\d{2}[-\\s]?\\d{2}\\u0022'"
186  r\"\\\"\\+?[7]\\s?\\(?\\d{3}\\)?\\s?\\d{3}[\\s?|\\-?|]\\d{2}[\\s?|\\-?|]\\d{2}\\\"|\\\"[8]\\s?\\(?\\d{3}\\)?\\s?\\d{3}[\\s?|\\-?|]\\d{2}[\\s?|\\-?|]\\d{2}\\\"|\\\"[8|7|+7]\\d{10}\\\"\""
186  r\"\\\"\\+?[7]\\s?\\(?\\d{3}\\)?\\s?\\d{3}[\\s?|\\-?|]\\d{2}[\\s?|\\-?|]\\d{2}\\\"|\\\"[8]\\s?\\(?\\d{3}\\)?\\s?\\d{3}[\\s?|\\-?|]\\d{2}[\\s?|\\-?|]\\d{2}\\\"|\\\"[8|7|+7]\\d{10}\\\"\""
/mnt/c/Users/vauli/Documents/Study/BI/Python > |
```

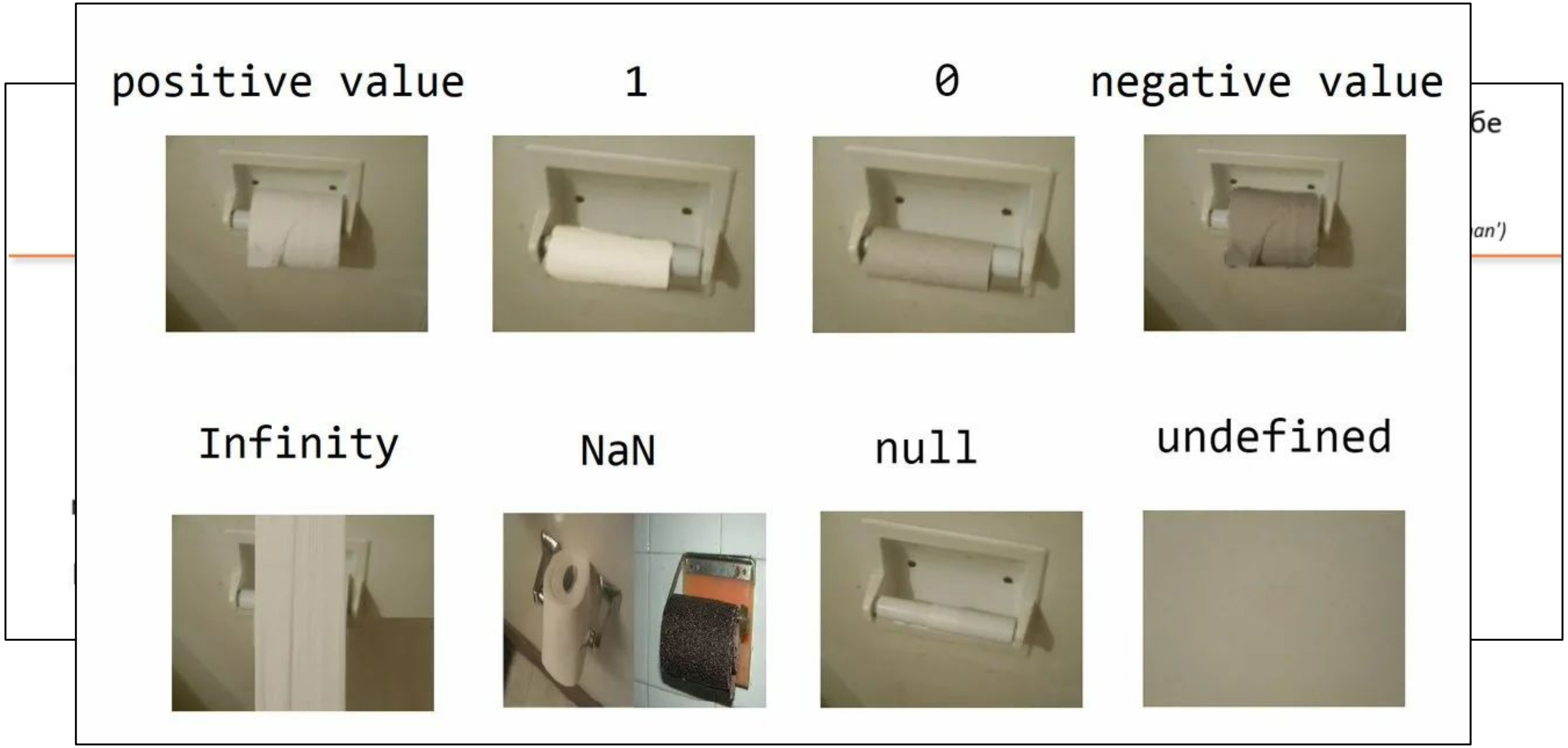
ДЗ 10

NA, NaN, None, NULL, ...

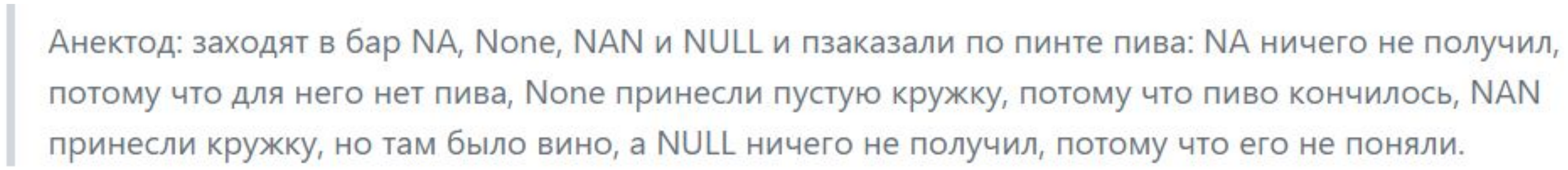


Объект	Класс	Если обернуть в bool() *Пример:* *bool(float('nan'))*	Равен сам себе через ==? *Пример:* *float('nan') == float('nan')*	Равен сам себе через is? *Пример:* *float('nan') is float('nan')*
None	<class 'NoneType'>	False	True	True
float('nan')	<class 'float'>	True	False	True
math.nan	<class 'float'>	True	False	True
numpy.nan	<class 'float'>	True	False	True
pandas.NA	<class 'pandas._libs.missing.NAType'>	Ошибка	<NA>	True

NA, NaN, None, NULL, ...



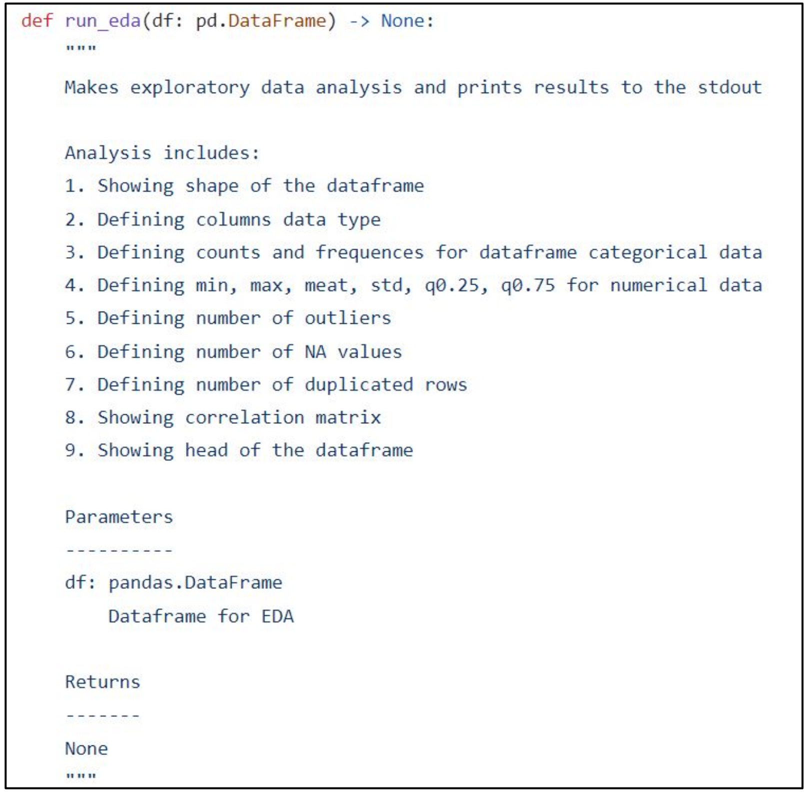
NA, NaN, None, NULL, ...



> Анектод: заходят в бар NA, None, NAN и NULL и пзаказали по пинте пива: NA ничего не получил, потому что для него нет пива, None принесли пустую кружку, потому что пиво кончилось, NAN принесли кружку, но там было вино, а NULL ничего не получил, потому что его не поняли.

EDA-модуль

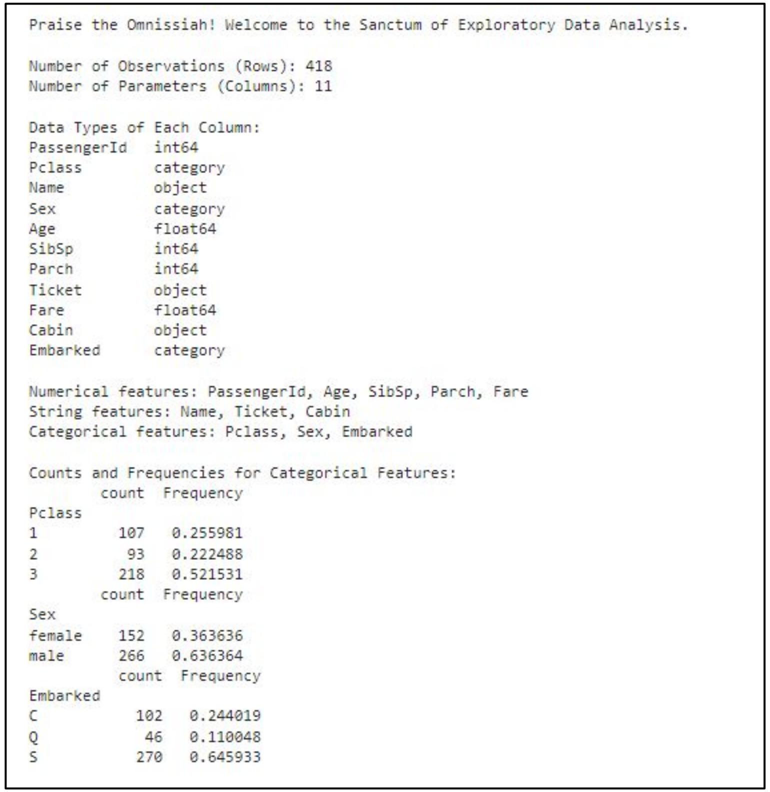
- ✓ Аннотация типов
- ✓ Краткое описание
- ✓ Содержание анализа
- ✓ Упомянуто про печать в stdout



```
def run_eda(df: pd.DataFrame) -> None:
    """
    Makes exploratory data analysis and prints results to the stdout

    Analysis includes:
    1. Showing shape of the dataframe
    2. Defining columns data type
    3. Defining counts and frequences for dataframe categorical data
    4. Defining min, max, meat, std, q0.25, q0.75 for numerical data
    5. Defining number of outliers
    6. Defining number of NA values
    7. Defining number of duplicated rows
    8. Showing correlation matrix
    9. Showing head of the dataframe

    Parameters
    ----------
    df: pandas.DataFrame
        Dataframe for EDA

    Returns
    -------
    None
    """
```

EDA-модуль

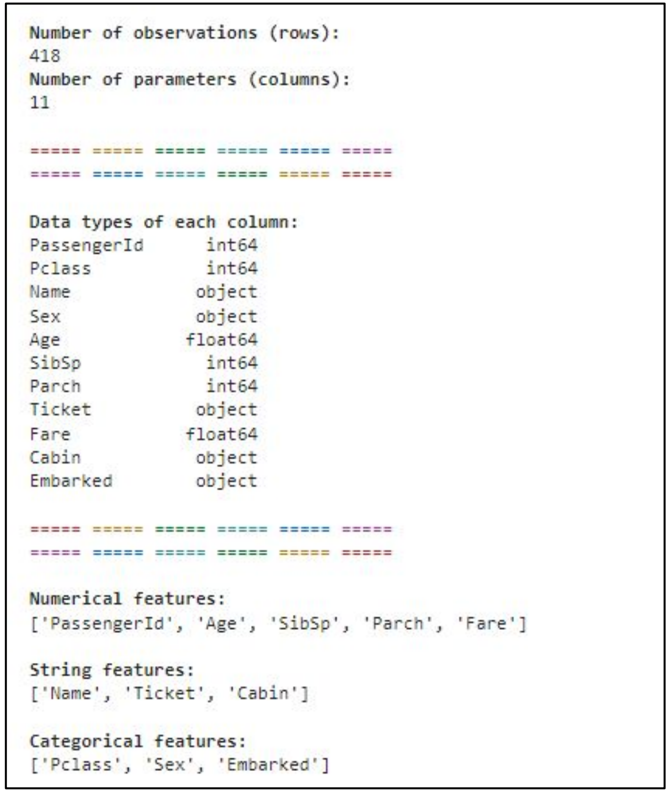
- ✓ Разбиение на блоки пустой строкой
- ✓ Результаты в виде списков и табличек
- ✓ Нету лишней информации
- ➜ Округлить числа



```
Praise the Omnissiah! Welcome to the Sanctum of Exploratory Data Analysis.

Number of Observations (Rows): 418
Number of Parameters (Columns): 11

Data Types of Each Column:
PassengerId   int64
Pclass        category
Name          object
Sex           category
Age           float64
SibSp         int64
Parch         int64
Ticket        object
Fare          float64
Cabin         object
Embarked      category

Numerical features: PassengerId, Age, SibSp, Parch, Fare
String features: Name, Ticket, Cabin
Categorical features: Pclass, Sex, Embarked

Counts and Frequencies for Categorical Features:
        count  Frequency
Pclass
1         107   0.255981
2          93   0.222488
3         218   0.521531
        count  Frequency
Sex
female    152   0.363636
male      266   0.636364
          count  Frequency
Embarked
C           102   0.244019
Q            46   0.110048
S           270   0.645933
```

EDA-модуль

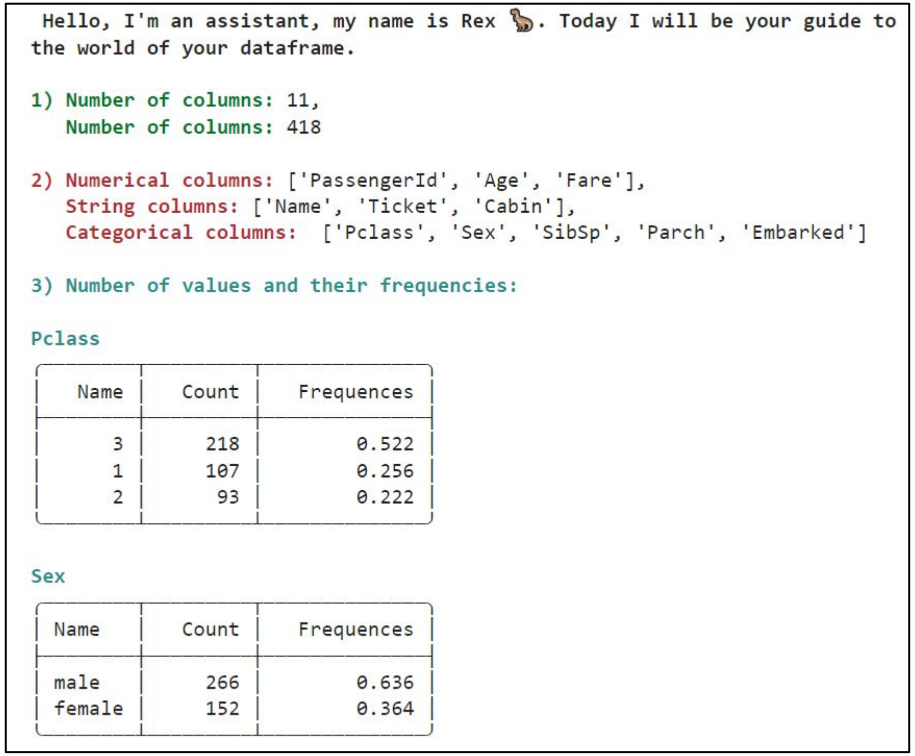
- ✓ Разбиение на блоки пустой строкой
- ✓ Результаты в виде списков и табличек
- ✓ Нету лишней информации
- ➪ Округлить числа
- ✓ Оформление жирным и цветом



```
Number of observations (rows):
418
Number of parameters (columns):
11

===== ===== ===== ===== ===== =====
===== ===== ===== ===== ===== =====

Data types of each column:
PassengerId      int64
Pclass           int64
Name            object
Sex             object
Age            float64
SibSp            int64
Parch            int64
Ticket          object
Fare           float64
Cabin           object
Embarked        object

===== ===== ===== ===== ===== =====
===== ===== ===== ===== ===== =====

Numerical features:
['PassengerId', 'Age', 'SibSp', 'Parch', 'Fare']

String features:
['Name', 'Ticket', 'Cabin']

Categorical features:
['Pclass', 'Sex', 'Embarked']
```

EDA-модуль

- ✓ Разбиение на блоки пустой строкой
- ✓ Результаты в виде списков и табличек
- ✓ Нету лишней информации
- Округлить числа
- ✓ Оформление жирным и цветом
- ✓ Оформление табличек



```
 Hello, I'm an assistant, my name is Rex 🐈. Today I will be your guide to
the world of your dataframe.

1) Number of columns: 11,
   Number of columns: 418

2) Numerical columns: ['PassengerId', 'Age', 'Fare'],
   String columns: ['Name', 'Ticket', 'Cabin'],
   Categorical columns:  ['Pclass', 'Sex', 'SibSp', 'Parch', 'Embarked']

3) Number of values and their frequencies:

Pclass

|   Name |   Count |   Frequences |
|--------|---------|--------------|
|      3 |     218 |        0.522 |
|      1 |     107 |        0.256 |
|      2 |      93 |        0.222 |

Sex

| Name   |   Count |   Frequences |
|--------|---------|--------------|
| male   |     266 |        0.636 |
| female |     152 |        0.364 |
```

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

Сохранение cluster map в виде sub-plot.