



Wrap up

07.12.2024



Course program

<p>Basic python</p>  <p>Scientific data analysis</p> 	<ol style="list-style-type: none">1. Intro. Git, GitHub2. Python recap, data types3. Functions4. Modules and libraries5. Files6. IDEs7. Virtual environments8. Regular expressions9. Numpy10. Pandas11. Visualisation12. Statistics13. Discussion
---	--

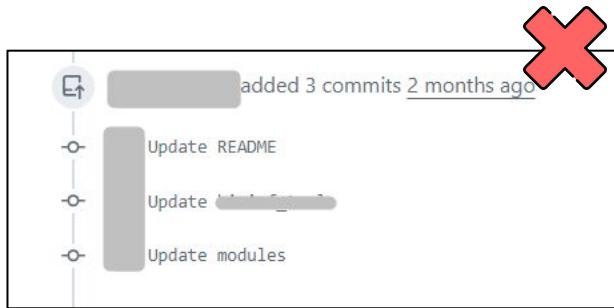


дз 2 - 5



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





README



Описание репо

course_materials (Public)

Edit Pins Watch 0 Fork 5 Star 1

main 1 Branch 0 Tags Go to file Add file Code

 nvalin Update HW12_PeerReview.md 6dd237a · 3 days ago 31 Commits

 Homeworks Update HW12_PeerReview.md 3 days ago

 data Add L9 data last month

 lectures Edit L12 4 days ago

About This is repo for the materials of the BI 2024/45 Python course

Activity Custom properties 1 star 0 watching



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](#)
- [Examples](#)
- [FAQ](#)
- [Discussion and Development](#)

Installation





README

- ✓ Описание репо
- ✓ Небольшое красочное интрано
- ✓ Содержание

Contents

- [Installation](#)
- [Usage](#)
 - [DNA/RNA Tools](#)
 - [FASTQ Filtering](#)
 - [RNA Translation](#)
 - [Bioinformatics File Processes](#)
- [Package Structure](#)
- [Contacts](#)

Content

- [Description](#)
- [Installation](#)
- [Examples](#)
- [FAQ](#)
- [Contact](#)



README

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

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

Installation

1. Clone the repository:

```
git clone https://github.com/The-AGI/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
```



README

- ✓ Описание репо
- ✓ Небольшое красочное интроверто
- ✓ 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`)
- Length of the read (`length_bounds`)

2. DNA and RNA Conversion

The `run_dna_rna_tools` function allows you to perform the following actions on DNA and RNA sequences:

- Transcribe a coding DNA strand to mRNA (`transcribe`)
- Write the sequence in reverse order (`reverse`)
- Create a complementary sequence (`complement`)
- Write the complementary sequence in reverse order (`reverse_complement`)

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': ('ACAGCAACATAAACATGATGGGATGGCGTAAGCCCCGAGATATCAGTTACCCAGGA
    '@SRX079804:1:SRR292678:1:1101:24563:24563': ('ATTAGCGAGGGAGGTGCTGAGAAGATGTCGCCAACGCCGTTGAAATTCCCTCAATC
    '@SRX079804:1:SRR292678:1:1101:30161:30161': ('GAACGACAGCACGCTCTGCATAACCGCGTCTTCTCTTAGCGGTGTGCAAAGCATG
    '@SRX079804:1:SRR292678:1:1101:47176:47176': ('TGAAGCGTCGATAGAAGTTAGCAACCCGCGGAACTCCGTACATCAGACACATTCCG
    '@SRX079804:1:SRR292678:1:1101:149302:149302': ('TAGGGTTGTATTTGAGATCCATGGCATGCCAAAAGAACATCGTCCCGTCCAATA
    '@SRX079804:1:SRR292678:1:1101:170868:170868': ('CTGCCGAGACTGTTCTCAGACATGGAAAGCTGATTGCATACACTCGTGAGTAA
    '@SRX079804:1:SRR292678:1:1101:171075:171075': ('CATTATAGTAATACGGAAGATGACTGCTGTTATCATTACAGCTCCATCGCATGAA
    '@SRX079804:1:SRR292678:1:1101:175500:175500': ('GACGCCGTGGCTGCACTATTGAGGGACCTGTCTCGAAGGGAAAGTTCATCTGAC
    '@SRX079804:1:SRR292678:1:1101:190136:190136': ('GAACCTTCTTAATTTATCTAGAGCCAAATTTAGTCATCTACACTAAATA
    '@SRX079804:1:SRR292678:1:1101:190845:190845': ('CCTCAGCGTGGATTGCCGCTATGCAGGAGCAGATAATCCCTGCCATCCCATTA
    '@SRX079804:1:SRR292678:1:1101:198993:198993': ('AGTTATTATGCATATTCTCATGTATGAGCCAACAAGATAGTACAAGTTTATTG
    '@SRX079804:1:SRR292678:1:1101:204480:204480': ('AGTGAGACACCCCTGAACATCCTAGTAAGACATTTGAATATTAGTTAGCC
    }
run_fastq_tools(EXAMPLE_FASTQ, gc_bounds=30)
# {'@SRX079804:1:SRR292678:1:1101:190136:190136': ('GAACCTTCTTAATTTATCTAGAGCCAAATTTAGTCATCTACACTAAATA
# 'DADCD@BEECEDE.BEDDDDD,>:@EEBEEHEFEHHFFH?FGBGFBD77B;;C?FFFGGFED.BBABBG@DBBE')
run_fastq_tools(EXAMPLE_FASTQ, gc_bounds=40, length_bounds=80, quality_threshold=35)
# {'@SRX079804:1:SRR292678:1:1101:171075:171075': ('CATTATAGTAATACGGAAGATGACTGCTGTTATCATTACAGCTCCATCGCATGAAAT
# 'HGHHHHGHHHHFHHEHHHHFGEHFGGGHHEGHHEEHBHHFGEHFGDCEGGGEFGF@FGGIIGEBGDDFFGFFGGFGF')}
```



README

- ✓ Описание репо
- ✓ Небольшое красочное интроверто
- ✓ 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 `abbreviation` 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 интроверто
- ✓ 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:

Селикова Ангелина - ██████████@gmail.com Implemented: protein_formula , protein_mass , seq_length .

Луценко Екатерина - ██████████@mail.ru Implemented: aa_content_check , aa_chain_charge .

Борисов Денис - ██████████@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:

██@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(
    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

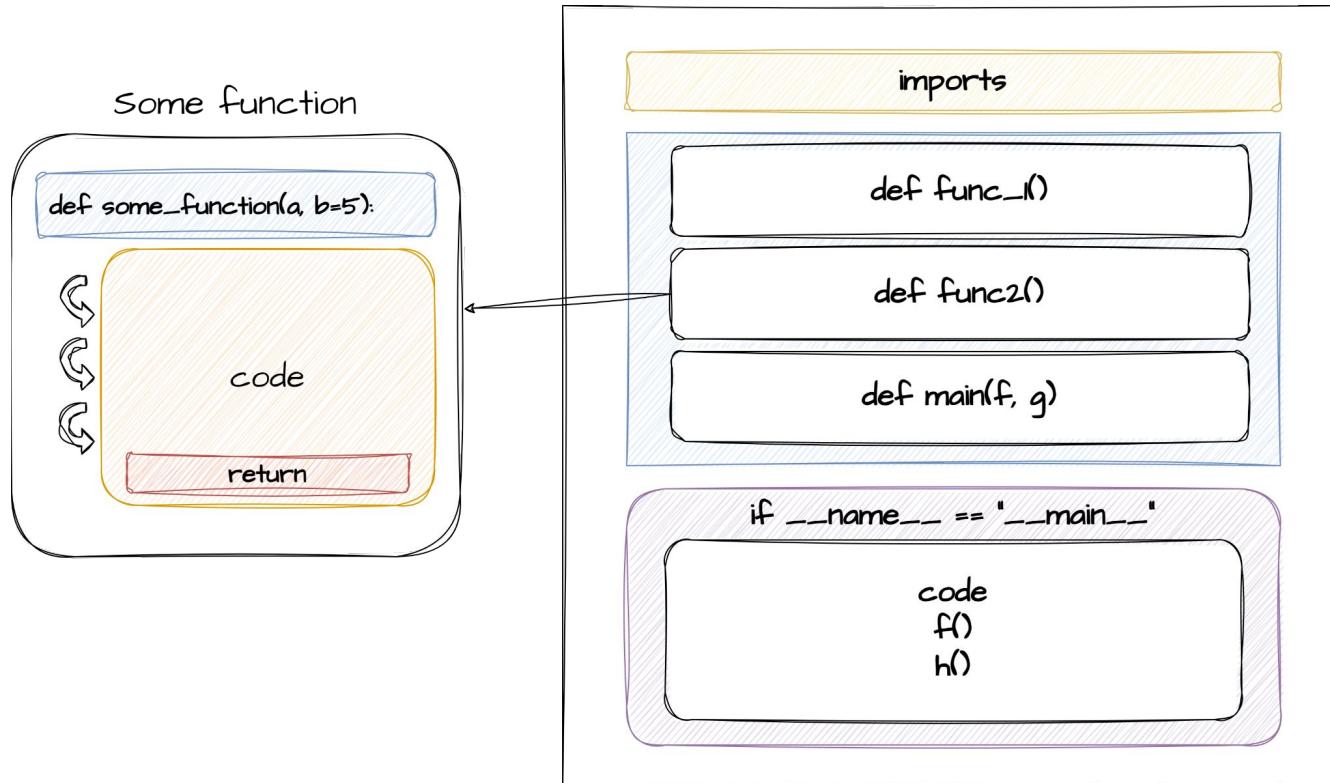
Функции

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
```



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 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"

```
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
```



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>

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 7



Перевод на соленый язык



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

def salt_text(text):
    vowels = re.compile(r'([аоиэыяюёАОИЭЫЯЮЁ]+)')
    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'([аоиэыяюёАОИЭЫЯЮЁ]+)')
    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

```
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]?\#?"

r'\b[ABCDEFGabcdefg]+(:maj|m|min)*\d*[#b]*(:add|sus|dim|aug)*\d*/*[ABCDEFGabcdefg]*\d*[#b]*'

r'[Cmsusdim]*|[Dmsusdim]*|[Emsus4dim]*|[Emsusdim]*|[Fmsusdim]*|[Gmsusdim]*|[Asusdim]*|[Bmsusdim]*|[Gg]*'



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
```



```
-42', '100.000.000', '-3.099e-734.149']
```



aryamakeeva	105
triopsydopsyda	92
anapomerash	85
Polina1010123	78
Regadene	78



```
-42', '100.000.000', '-3.099e-734.149']
```



aryamakeeva	105
triopsydopsyda	92
anapomerash	85
Polina1010123	78
Regadene	78



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"']





дз 9



NA, NaN, None, NULL, ...

Объект	Класс	Если обернуть в bool() Пример: <code>bool(float('nan'))</code>	Равен сам себе через ==?	Равен сам себе через is?
			Пример: <code>float('nan') == float('nan')</code>	Пример: <code>float('nan') is float('nan')</code>
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, ...

positive value



1



0



negative value



Infinity



NaN



null



undefined



6e
an')



NA, NaN, None, NULL, ...

Заходят как-то NA, None, NAN и NULL в бар, а бармен им не говорит.

(Вероника Самусик)

Самый pytonic:

(Елена Кожевникова)

```
def vasilii_ivanovichDefines_dunay_to_petka_pustota(realm: str) -> str:  
    """Define metaphoric Danube river in programming realms"""  
    responses = {  
        'Python': 'None',  
        'Java': 'NAN',  
        'C++': 'NULL',  
        'JavaScript': 'NA',  
        'Ruby': 'nil',  
    }  
  
    return responses.get(realm, "Ты что, Петка, совсем охренел? Вот он. (с)")  
  
realm = input("Васильваныч, а Дунай существует в Python/Java/C++/JavaScript/Ruby?")  
print(vasilii_ivanovichDefines_dunay_to_petka_pustota(realm))
```

Самый жизненный:

(Евгения Цымбалова)

Выпускается биофизик из университета и идет в бюро распределения. Ему говорят - ваша профессия очень востребована у нас для вас целых 4 варианта, вот список. Он берет бумажку в руку и там написано: NA, None, NAN, NULL.

Самый мудрый:

(Анна Калыгина)

Ходят легенды, что жил как-то на земле настолько умный человек, что с его знаниями и сковоркой никто не мог потягаться. Приходили к этому мудрецу за советом и главы стран, и дипломаты со всего мира, и учёные, и врачи, и простые рядовые. И любой пришедший с вопросом заставлял мудрого человека неизменно сидящим за своим дубовым столом на дубовом стуле в старом кабинете. И на любой самый сложный вопрос человек мог найти ответ. И был у этого мудреца особый ритуал - он всегда, прежде чем ответить на вопрос, почесывал затылок и ёрзкал на дубовом стуле, затем открывал тяжелый ящик своего дубового стола и доставал оттуда маленький листок бумаги. Он внимательно смотрел на него, потом аккуратно клал обратно в ящик и только после этого давал безукоризненный ответ. Его ученики и коллеги много лет пытались узнать, что же написано на этом листке, но мудрец никому не показывал его содержимое. Однажды мудрец умер. Коллеги и ученики, исполненные любопытства, наконец-то решились открыть ящик и узнать, в чём заключался его секрет. Они осторожно вынули листок, развернули его и прочли: "NA - not available, NaN - not a number, NULL - who blin вообще пользуется SQL?"



NA, NaN, None, NULL, ...

NA, NAN, None сидели на питоне, NA упал в ёмкей питчу, NAN упал в pandas, а None остался на питоне, строя глазки null на жабе Java
(Никита Галынин)

Шел медведь-программист по лесу, видит горящую машину, а в ней Na, None, NaN и NULL сидят, решил не садиться в неё, а пропустить.

(Егор Куликов)

Вошли как-то **NA**, **None**, **Nan** и **NULL** в строку кода

NA сразу сказал:

"Я тут как **Not Available**, просто укажите, что данных нет."

None вставил:

"Я в Python, если уж чего-то нет, то это намеренно — оставим поле пустым."

NAN возмутился:

"А если расчет не сработал и получилось что-то невообразимое? Я всегда сигнализирую, что тут не число!"

NULL из SQL вздохнул:

"Ребята, я — пустота сама по себе. Меня нет, но я настолько официально отсутствую, что об этом знает вся база данных."

И так каждый остался верен своей пустоте: **NA** просто не пришел, **None** решил остататься никем, **NaN** указывал на математический сбой, а **NULL** гордо занимал свое пустое место в таблице. ваш ответ тут

(Екатерина Лебедева)

Однажды **NA**, **None**, **NaN** и **NULL** решили сходить отдохнуть в бар "Пора Крафт" и говорят бармену:

— Нам всем пиво!

Сидят они, пьют и делятся ощущениями:

NA говорит:

— Всё вроде пью это пиво, а как будто в кружке ничего и не было.

None ему возразил:

— А мне бармен вообще пустой стакан дал!

Тут они стали спорить, для кого жизнь несправедливее, и в разговор вмешался **NaN**:

— Нет, ну вы видели этот цирк?! Бармен взял, открыл кран хигулевского, перевернулся стакан вверх дном и так пиво и наливал... (ну вы поняли это типа недоступимая операция)!! Понаберут по объявлению!

A **NULL** и вовсе умер. Наступил багажан.

На звуки пьяной драки зашел **fillna**) и удивился, что в пятницу в баре никого и не было.

(Алина Назарова)

Заходят как-то раз **NA**, **None**, **NAN**, **NULL** в бар. Бармен им и говорит: - Вы все какие-то неопределенные. Что вы здесь забыли?

NA отвечает: - Я просто пустой, но не потерянный. Просто ещё не решил, чего хочу...

None, вздыхая, говорит: - Я пустой и потерянный, как человек, который называет свой хобби "поиск смысла жизни"

NaN добавляет: - А я просто не могу понять, кто я. И никто меня не понимает!

NULL, смотря в пустоту: - Я... ну, я вообще не уверен, что я тут, так что, возможно, я просто не существую...

Бармен: - Ну что ж, ребята, похоже, вам всем нужно не просто выпить. Вам нужен не коктейль, а хороший специалист по определению типов данных

(Виталина Хисматулина)

(Алексей Чутко)

Внимание, анекдот

Сидят как-то четыре пропущенных значения в таблице и спорят, кто из них круче.

NA говорит: — Я здесь самый важней! Меня везде знают, а во всех датафреймах не стражи пропусков стоят!

None флегматично отвечает: — Ты конечно, молодец, но без меня Python бы вообще не знал, что такое отсутствие данных.

Тут встремляется **NaN**: — Да ладно вам, без меня ваши числа вообще бы не понимали, что у них нет значения! Я — научный стандарт, между прочим!

NULL пытается сказать свою веское слово, но тут его перебивает **NaN**: — Дружики-пирожки, твой выбор неправильная дверь, базы данных на два блока вниз.

Тут появляется программист, вздыхает и говорит: — Да какие вы важные, вас даже Excel уважать не хочет!

Все обиделись, кроме **NaN**, который сказал: — Ну да, я ему тоже #ДЕПЛЮ всегда пишу.

Na, None, Nan и Null на сеансне у психолога.

None: моя жизнь пуста и пресная... кажется мне не хватает смысла в жизни, что мне делать?

NaN: я никак не могу стать настоящим числом, что мне делать?

Null: бросает в психологии свои пустые значения.

NA is not available now, please, call back later.

(Анна Калинина)

(Полина Малышева)

Штирлиц заходит в бар и заказывает кружку пива. Бармен спрашивает:

— Какое? У нас есть NA, None, NULL и NaN.

Штирлиц задумывается и отвечает:

— Дайте мне ничего.

(Валерия Степанова)

В дальнем дальнем датасентре собирались однажды **NA**, **None**, **NaN** и **NULL** на корпоратив.

NA: «Я пришел в костюме невидимки, чтобы держать всех в напряжении. Никогда не знаешь, когда я появлюсь!»

None: «А я в своем привечном — вообще без kostюма! Я ведь — буквально ничего.»

NaN: «Тарик, я пытаюсь подобрать себе образ, но, похоже, так и остался неопределенным. Зато вот, смотрите, мой галстук в горошек — крути, ни к чему не привяжешь, как я?»

NULL: «А я даже не знаю, позвали ли меня на вечеринку. В базе чиселес, а толку-то...»

Тут рядом сел любопытный дата-аналитик, решивший поинтересоваться:

«Если вы все будете в одном датафрейме, хотите сказать, что мало что заменит разницу?»

NA: «Конечно! Все будут гадать, что же на самом деле я значу.»

None: «Ух, сегодня окажутся пустыми, когда я вмешаюсь.»

NaN: «И добавлю немного хаоса и веселья!»

NULL: «А я буду тихо сидеть в уголке, представляем компании баз данных чувство комфорта.»

И пока дата-аналитик потягивал свой кофе, он понял, что эти товарищи никогда не оставят его в покое, каждому из них найдется свое место в мире данных — пусть даже неопределенное, но невероятно важное.

(Елена Дементьева)

Собирались как-то пропущенные значения в баре — NA, None, NaN и NULL. Каждый заказывает свой:

NA из R заявляет: «Я — not available. Меня обычно зовут, когда данных не хватает. Я — король статистики, всё решю. Мне виски похреще — быть в статистике нелепого!»

A None (из Python): «Я-то вообще философ и символизирую полное отсутствие чего-либо. Это даже не ошибка, а концепция! Мне чёрный кофе и без сахара — люблю, когда всё просто.»

NaN (из NumPy): «А я — Not a Number. Я — то, что происходит, когда вы делите ноль на ноль или пытаетесь взять логарифм отрицательного числа. И у меня математическая драма, так что мне миниатюра!»

A NULL, из SQL, вздохнул и сказал: «Я — просто NULL. Пропущенные значения в базах данных — это моя работа. Все меня используют и никто до конца не понимает. Дайте мне просто воду, я тут на работе.»

Бармен посмотрел на эту компанию, усмехнулся и заявил: «Извините, но вы все пропущены в моём заказе!»



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 frequencies for dataframe categorical data
    4. Defining min, max, mean, 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

Find categorical trait Pclass		
Frequency	Counts	Types
0	0.521531	218
1	0.255981	107
2	0.222488	93

Find categorical trait Sex		
Frequency	Counts	Types
0	0.636364	266
1	0.363636	152

```
=====
Hello! Let's perform some awesome EDA!
Number of observations (rows): 418
Number of parameters (columns): 11
=====

Data types for each column:
PassengerId    int64   Numerical
Pclass          int64   Categorical
Name            object  String
Sex             object  Categorical
Age             float64 Numerical
SibSp           int64   Categorical
Parch           int64   Categorical
Ticket          object  String
Fare            float64 Numerical
Cabin           object  String
Embarked        object  Categorical
dtype: object
=====

Numerical features: ['PassengerId', 'Pclass', 'Age', 'SibSp', 'Parch', 'Fare']
Categorical features: ['Name', 'Sex', 'Ticket', 'Cabin', 'Embarked']
=====
```

(1) DATASET STATISTICS		
Number of parameters	11.00	Value
Number of observations	418.00	
Missing cells	414.00	
Missing cells (%)	9.00	
Rows with missing values	331.00	
Rows with missing values (%)	79.19	
Duplicate rows	0.00	
Duplicate rows (%)	0.00	

(2) PARAMETER STATISTICS		
Your dataframe contains parameters of the following types:		
PassengerId	dtypes	Annotation
Pclass	int64	Categorical
Name	object	String
Sex	object	Categorical
Age	float64	Numerical
SibSp	int64	Categorical
Parch	int64	Categorical
Ticket	object	String
Fare	float64	Numerical
Cabin	object	String
Embarked	object	Categorical



EDA-модуль

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

```
Hello, Darling! This is your EDA!

Number of observations: 418
Number of features: 11

Yeah! That's A LOT

Data types for each column:
PassengerId      int64
Pclass            int64
Name              object
Sex               object
Age               float64

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

=====
===== ===== ===== ===== =====
===== ===== ===== ===== =====
```



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	Frequencies
3	218	0.522
1	107	0.256
2	93	0.222

Sex

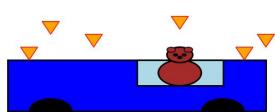
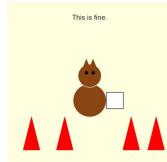
Name	Count	Frequencies
male	266	0.636
female	152	0.364



дз 10

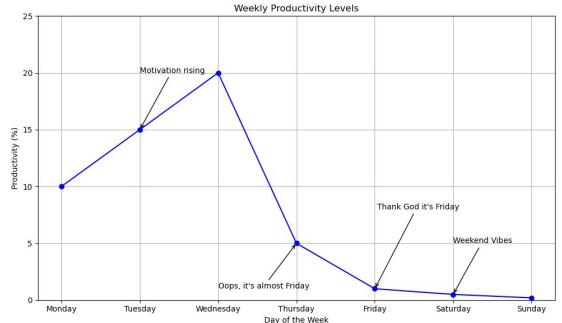
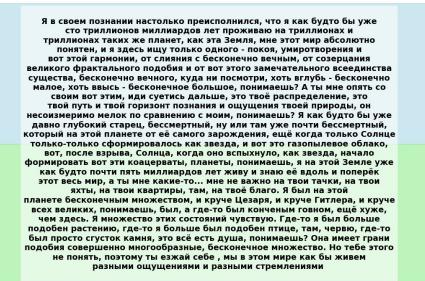
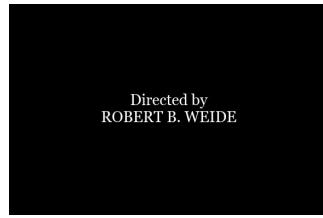
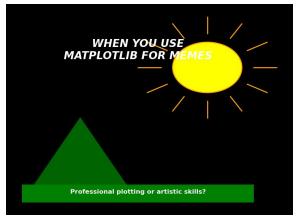


plt.meme

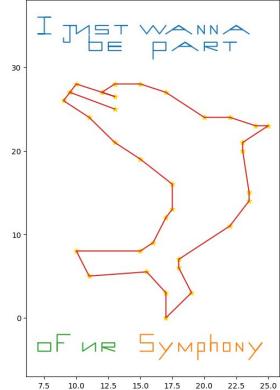
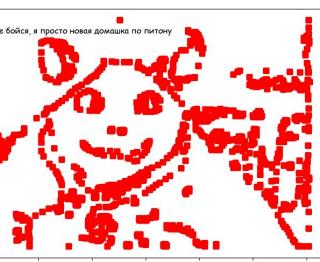
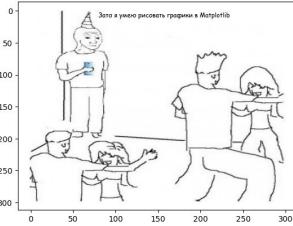


Сел медведь в машину и сгорел

<- Saddam Hussein



#МОЯ ЖИЗНЬ - ЭТО СПЛОШНОЙ МЕМ





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 \(B1\)](#)
- [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](#)