Программа для проверки критериев согласия из excel

Установка

Программа разработана под python 3.11.3

Устанавливаем необходимые библиотеки перечисленные в файле requirements.txt

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

Описание файлов

norm_test.py

Файл с функциями для проверки нормальности по критерию Пирсона и Лилиефорса и соответствия выборки бета-распределению по критерию Колмогорова-Смирнова.

class PDFNormal - Класс для работы с подгонкой под нормальной распределение:

- *@staticmethod estimate_params(data)* - возвращает точечную оценку математического ожидания и среднеквадратичного отклонения;
- *pdf(self, x)* - возвращает значение плотности вероятности в точке x для подгонки.

class PDFBeta - Класс для работы с подгонкой под бета-распределение:

- *@staticmethod estimate_params(data)* - возвращает точечные оценки параметров бета-распределения, полученные через точечные оценки математического и дисперсии;
- *pdf(self, x)* - возвращает значение плотности вероятности в точке x для подгонки.

class DataContainer - класс для хранения выборки и полученных pvalue по различным критериям согласия:

- *testHypotheses(self, normal: PDFNormal, beta: PDFBeta) -> None* - проверка гипотезы о соответствии выборки нормальному распределению по критерию Пирсона и критерию Лилиефорса, проверка гипотезы о соответствии выборки бета-распределению по критерию Колмогорова-Смирнова. Сохранение значений pvalue в объекте класса;
- *pvals_info(self, label: str)* - формирование строки для печати информации о результатах критериев согласия.

plot_fit(data, title="", isSubplot=False) - Строит диаграмму накопленных частот для выборки *data*, подогнанное теоретическое нормальное и бета распределения и

ожидаемую нормальную диаграмму накопленных частот.

- Возвращает объект класса **DataContainer** с результатами проверки критериев согласия.
- Дополнительные параметры:
  - *title* - Текст, который будет добавлен в заголовок графика;
  - *isSubplot* - Если True, то функция не вызывает *plt.plot()*. Используется для построения нескольких графиков на одном.

test.py

Скрипт для проверки работы критерия Пирсона и критерия Лилиефорса (в программе Колмогорова-Смирнова) на нескольких типовых распределениях. Каждый опыт проводится 100 раз.

Пример работы программы

```
$ python test.py
1. Нормальные распределения. Объём выборки 100
Критерий Пирсона:                  95/100 раз критерий выполняется
Критерий Колмогорова-Смирнова:     95/100 раз критерий выполняется

2. Экспоненциальные распределения. Объём выборки 100
Критерий Пирсона:                  0/100 раз критерий выполняется
Критерий Колмогорова-Смирнова:     0/100 раз критерий выполняется

3. Экспоненциальные распределения. Объём выборки 1000
Критерий Пирсона:                  0/100 раз критерий выполняется
Критерий Колмогорова-Смирнова:     0/100 раз критерий выполняется

4. Равномерные распределения. Объём выборки 100
Критерий Пирсона:                  0/100 раз критерий выполняется
Критерий Колмогорова-Смирнова:     0/100 раз критерий выполняется

5. Равномерные распределения. Объём выборки 1000
Критерий Пирсона:                  0/100 раз критерий выполняется
Критерий Колмогорова-Смирнова:     0/100 раз критерий выполняется

6. Распределение Коши. Объём выборки 100
Критерий Пирсона:                  0/100 раз критерий выполняется
Критерий Колмогорова-Смирнова:     0/100 раз критерий выполняется
```

excel_worker.py

Содержит два сценария для работы:

Первый сценарий. Программа для проверки нормальности и соответствия бета-распределению выборок, записанных в Excel файле. Данные для проверки на листе

располагаются в левой верхней части листа. Первая строка используется для названий выборок. Каждая выборка располагается в одном столбце.

Протокол с результатами проверки гипотез выводится в консоль. Построенные графики с этими же результатами сохраняются в папке png/<дата и время>.

Второй сценарий. Программа для расчёта квантиля q_p с заданным уровнем вероятности p бета-распределения $B(\alpha, \beta)$. Данные берутся из заданного эксель файла.

Для случайной величины с бета-распределением $X \sim B(\alpha, \beta)$ Плотность распределения:

$$f_X(x) = \frac{1}{B(\alpha, \beta)} \cdot x^{\alpha-1}(1-x)^{\beta-1}, \text{ где } B(\alpha, \beta) = \int_0^1 x^{\alpha-1}(1-x)^{\beta-1}dx$$

Параметры бета-распределения можно расчитать через мат. ожидание m_x и дисперсию σ_x^2:

$$\alpha = -\frac{m_x(m_x^2 - m_x + \sigma_x^2)}{\sigma_x^2}, \ \beta = \frac{(m_x - 1)(m_x^2 - m_x + \sigma_x^2)}{\sigma_x^2}$$

Квантиль заданной вероятности находится из:

$$p = P[x < q_p] = \int_0^{q_p} f_X(x)dx = \frac{B_{q_p}(\alpha, \beta)}{B(\alpha, \beta)} = I_{q_p}(\alpha, \beta) \Rightarrow q_p = I_p^{-1}(\alpha, \beta)$$

Данная программа принимает значения математического ожидания и дисперсии из файла Excel, а также уровня вероятности p от пользователя, расчитывает значения параметров α, β для бета-распределения и проводит расчёт $q_p = I_{q_p}^{-1}(\alpha, \beta)$ (обратной регуляризованной бета-функции).

Формат данных в таблице Excel: первая строка выделяется под названия столбцов. Названия столбцов могут быть произвольными. Программа работает с первыми 3 найденными столбцами, считая, что 1 столбец $-$ пользовательское название, 2 столбец $-$ значение математического ожидания, 3 столбец $-$ значение дисперсии.

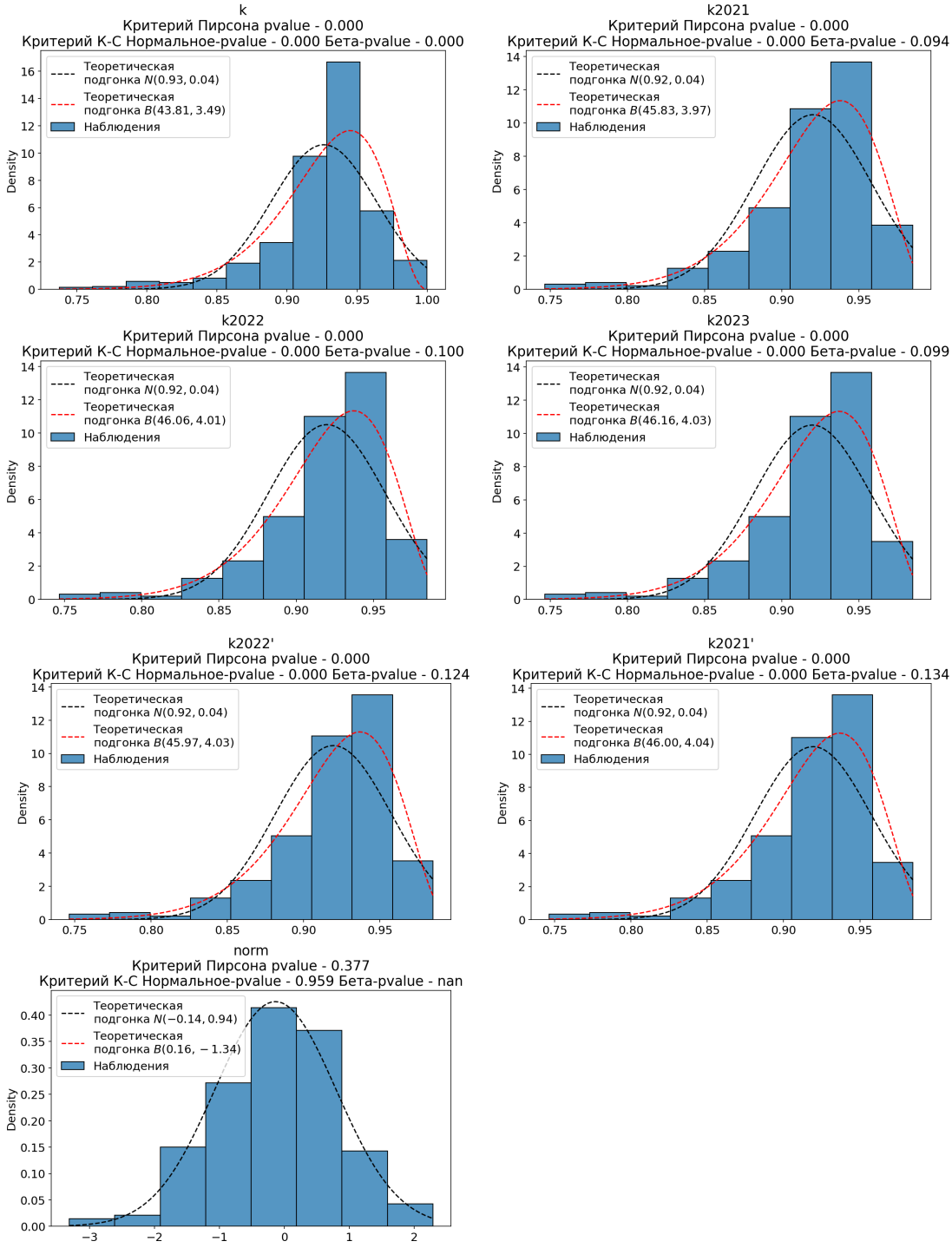
Пример работы программы по первому сценарию

```
$ python excel_worker.py
Выберите решаемую задачу:
1 - проверка гипотез о нормальном и бета распределениях
2 - расчёт квантилей для бета-распределения.
1
Введите название (или путь) файла Excel с данными: data.xlsm
Открываем файл data.xlsm
Найдены следующие листы: Проверка нормальности; Лист1; Лист2
Введите название листа, откуда брать данные: Проверка нормальности
```

```
Выбран лист Проверка нормальности

Парсим данные
Найдены следующие столбцы: k k2021 k2022 k2023 k2022' k2021' norm
Строим графики. Если pvalue > 0.05, значит критерий согласия выполняется
1 график из 2
k                              Критерий Пирсона: 0.000    Критерий К-С Нормальное распр
k2021                          Критерий Пирсона: 0.000    Критерий К-С Нормальное распр
k2022                          Критерий Пирсона: 0.000    Критерий К-С Нормальное распр
k2023                          Критерий Пирсона: 0.000    Критерий К-С Нормальное распр
2 график из 2
k2022'                         Критерий Пирсона: 0.000    Критерий К-С Нормальное распр
k2021'                         Критерий Пирсона: 0.000    Критерий К-С Нормальное распр
norm                           Критерий Пирсона: 0.377    Критерий К-С Нормальное распр
Построенные графики сохранены в: E:\png\04.06.2024 19-51-58
Протокол сохранен в: E:\protocol
Для выхода из программы нажмите Enter...
```

Графики сохраняются в папке ./png/<дата и время>:



Протокол сохраняется в формате таблиц Excel сохраняется в protocol/.

	A	B	C	D	E	F
1		Выборка	ирсон pval	мальное	бета pvalue	
2	0	k	1.78E-74	6.86E-18	4.1E-08	
3	1	k2021	8.23E-27	2.78E-05	0.093654	
4	2	k2022	1.85E-26	3.68E-05	0.100017	
5	3	k2023	2.06E-26	3.73E-05	0.098684	
6	4	k2022'	7.77E-26	6.18E-05	0.123945	
7	5	k2021'	1.09E-25	7.55E-05	0.133893	
8	6	norm	0.377135	0.959044		
9						

Пример работы программы по второму сценарию

Данные в таблице Excel записаны в следующем формате

	A	B	C	D
1	Название	мат.ож	дисп	
2	k1	0.94883	0.0016	
3	k2	0.92707	0.0016	
4	k3	0.91872	0.0016	
5	k4	0.93848	0.0016	
6	k5	0.91197	0.0016	
7	k6	0.92963	0.0016	
8	k7	0.93357	0.0016	
9	k8	0.93272	0.0016	
10	k9	0.91849	0.0016	
11	k10	0.9379	0.0016	
12	k11	0.93681	0.0016	
13	k12	0.94171	0.0016	
14	k13	0.9138	0.0016	
15	k14	0.94863	0.0016	

```
$ python excel_estimator.py
Выберите решаемую задачу:
1 - проверка гипотез о нормальном и бета распределениях
2 - расчёт квантилей для бета-распределения.
2
Введите название (или путь) файла Excel с данными: beta2.xlsx
Открываем файл beta2.xlsx
Найдены следующие листы: Лист1
Выбран лист Лист1

Парсим данные
Найдены следующие столбцы: Название мат.ож дисп
Первые три считаем следующими столбцами: название, значение мат. ожидания, значение дис
Введите уровень вероятности для расчёта квантиля (0.95 по умолчанию): Протокол сохранен
Для выхода из программы нажмите Enter...
```

Полученный протокол

	A	B	C	D	E	F	G
1		Название	Мат. ожидание	Дисперсия	Альфа	Бета	Квантиль0.95
2	0	k1	0.938811483	0.0016	32.7672	2.13565	0.98774114
3	1	k2	0.948445673	0.0016	28.0363	1.52396	0.993538179
4	2	k3	0.910914497	0.0016	45.2891	4.42918	0.966109448
5	3	k4	0.910145937	0.0016	45.6099	4.50283	0.965454581
6	4	k5	0.948154868	0.0016	28.1823	1.54101	0.993383849
7	5	k6	0.918073538	0.0016	42.2397	3.76936	0.972097386
8	6	k7	0.923406002	0.0016	39.8954	3.30921	0.976401987
9	7	k8	0.902321057	0.0016	48.803	5.28307	0.958676998
10	8	k9	0.941693281	0.0016	31.3743	1.9426	0.989609358
11	9	k10	0.913031406	0.0016	44.399	4.22912	0.967901881
12	10	k11	0.912718015	0.0016	44.5314	4.25848	0.967637607
13	11	k12	0.908386627	0.0016	46.3393	4.67345	0.963947721
14	12	k13	0.918858749	0.0016	41.8984	3.69991	0.972740435
15	13	k14	0.927593759	0.0016	38.0102	2.96701	0.979667066
16	14	k15	0.918518582	0.0016	42.0464	3.72992	0.972462221
17	15	k16	0.930303225	0.0016	36.7697	2.75473	0.981715713
18	16	k17	0.920922108	0.0016	40.9952	3.52018	0.974415587
19	17	k18	0.943625914	0.0016	30.4296	1.81793	0.990802757
20	18	k19	0.930335401	0.0016	36.7549	2.75225	0.98173971
21	19	k20	0.900118924	0.0016	49.678	5.51249	0.956737556
22	20	k21	0.935307268	0.0016	34.4354	2.38181	0.985343123