Федеральное государственное автономное образовательное учреждение высшего образования

«Пермский государственный национальный исследовательский университет»

Институт компьютерных наук и технологий

Отчет
по лабораторной работе № 3

по дисциплине
«Введение в анализ данных»

Студент Власова Елизавета Александровна

Группа ИТ–14–2023

Пермь 2025

Оглавление

Задание 1 (6 баллов)..... 3
1. Загрузите информацию из этих файлов.....3
2. Постройте одномерные массивы ежегодного производства и потребления электроэнергии в среднем за последние 5 лет (один массив – производство по всем странам, второй – потребление по всем странам)..... 4
3. Напишите выражения, позволяющие получить ответ на следующие вопросы:.....8
Задание 2 (6 баллов)..... 12

Задание 1 (6 баллов)

В файле «global-electricity-generation.csv» представлена информация о производстве электроэнергии странами с 1992 по 2021 год. В файле «global-electricity-consumption.csv» – информация о потреблении электроэнергии. Все данные – в млрд. кВт*ч.

1. Загрузите информацию из этих файлов.

Указание: рекомендуется воспользоваться функцией genfromtxt. Обратите внимание, что данные в файлах разделены запятой, в обоих файлах есть заголовок, который при загрузке нужно пропустить. Для корректной загрузки названий стран необходимо также указать тип данных (по умолчанию загружаемые данные преобразуются в вещественный тип). Вы можете либо собрать все данные в один массив записей, либо работать с тремя разными массивами: массив названий стран (одномерный, строки), массивы производства и потребления энергии (двумерные, вещественные)).

Порядок стран в двух исходных файлах одинаковый.

Решение:

np.genfromtxt – универсальная функция для чтения табличных данных из текстового файла. Она умеет:

- читать числовые и текстовые данные;
- пропускать заголовки (skip_header);
- учитывать разделитель между столбцами (delimiter);
- работать с разными типами данных (dtype);
- заменять пропущенные значения на np.nan.

При загрузке данных мы:

- указываем путь к файлу;
- разделяем данные через запятую;

- считываем как строки (названия стран);
- пропускаем первую строку с заголовком;
- берем только первый столбец (страны);
- указываем кодировку для корректного чтения кириллицы и латиницы.

```
# 1
# загружаем список стран из файла производства
countries = np.genfromtxt(
    'global-electricity-generation.csv',
    delimiter = ',',
    dtype = str,
    skip_header = 1,
    usecols = 0, # первый столбец (страны)
    encoding = 'utf-8'
)

# загружаем числовые данные о производстве электроэнергии (все столбцы, кроме первого)
generation_values = np.genfromtxt(
    'global-electricity-generation.csv',
    delimiter = ',',
    skip_header = 1,
    usecols = range(1, 31), # столбцы 2-31 (по годам 1992-2021)
    encoding = 'utf-8'
)

# аналогично загружаем данные о потреблении электроэнергии
consumption_values = np.genfromtxt(
    'global-electricity-consumption.csv',
    delimiter = ',',
    skip_header = 1,
    usecols = range(1, 31),
    encoding = 'utf-8'
)
```

2. Постройте одномерные массивы ежегодного производства и потребления электроэнергии в среднем за последние 5 лет (один массив – производство по всем странам, второй – потребление по всем странам).

Указание: не забывайте, что в агрегирующих функциях для построчной обработки нужно указывать параметр axis=1.

Решение:

np.nanmean – функция, которая вычисляет среднее арифметическое, игнорируя NaN (отсутствующие данные). Параметр axis=1 означает, что усреднение идет по строкам, то есть для каждой страны отдельно).

```
# 2
# средние значения за последние 5 лет

# проверка, что есть хотя бы один столбец данных
if generation_values.shape[1] > 0 and consumption_values.shape[1] > 0:
    # если меньше 5 лет — берём доступные годы (все, что есть)
    num_years = min(5, generation_values.shape[1], consumption_values.shape[1])

    # берем последние `num_years` лет
    generation_last5 = generation_values[:, -num_years:]
    consumption_last5 = consumption_values[:, -num_years:]

    # считаем среднее по каждой стране, игнорируя NaN
    generation_mean = np.array([
        np.nanmean(row) if not np.isnan(row).all() else np.nan
        for row in generation_last5
    ])
    consumption_mean = np.array([
        np.nanmean(row) if not np.isnan(row).all() else np.nan
        for row in consumption_last5
    ])
else:
    # если вообще нет данных — заполняем NaN
    generation_mean = np.full(generation_values.shape[0], np.nan)
    consumption_mean = np.full(consumption_values.shape[0], np.nan)

# вывод результатов
print("\n2. Средние значения за последние 5 лет:")
for country, gen, cons in zip(countries, generation_mean, consumption_mean):
    if not np.isnan(gen) or not np.isnan(cons):
        gen_str = f"{gen:10.2f}" if not np.isnan(gen) else "   нет данных"
        cons_str = f"{cons:10.2f}" if not np.isnan(cons) else "   нет данных"
        print(f"{country:<25} производство: {gen_str} | потребление: {cons_str}")
```

Результат работы программы:

```
2. Средние значения за последние 5 лет:
Algeria                   производство:      74.11 | потребление:      64.28
Angola                    производство:      14.39 | потребление:      12.77
Benin                     производство:       0.23 | потребление:       0.86
Botswana                  производство:       2.62 | потребление:       3.59
Burkina Faso              производство:       1.69 | потребление:       2.42
Burundi                   производство:       0.32 | потребление:       0.38
Cabo Verde                производство:       0.45 | потребление:       0.32
Cameroon                  производство:       8.16 | потребление:       6.30
Central African Republic  производство:       0.15 | потребление:       0.14
Chad                      производство:       0.30 | потребление:       0.26
Comoros                   производство:       0.11 | потребление:       0.08
Congo-Brazzaville         производство:       3.47 | потребление:       1.89
Congo-Kinshasa            производство:      10.79 | потребление:       9.15
Cote d'Ivoire             производство:      10.24 | потребление:       7.14
Djibouti                  производство:       0.06 | потребление:       0.49
Egypt                     производство:     196.53 | потребление:     162.10
Equatorial Guinea         производство:       1.38 | потребление:       1.24
Eritrea                   производство:       0.43 | потребление:       0.39
Eswatini                  производство:       0.72 | потребление:       1.53
Ethiopia                  производство:      14.01 | потребление:       9.38
Gabon                     производство:       2.22 | потребление:       2.27
Gambia                    производство:       0.30 | потребление:       0.25
Ghana                     производство:      17.01 | потребление:      14.86
Guinea                    производство:       2.26 | потребление:       1.97
Guinea-Bissau             производство:       0.08 | потребление:       0.08
Kenya                     производство:      11.40 | потребление:       8.99
Lesotho                   производство:       0.52 | потребление:       0.87
Liberia                   производство:       0.57 | потребление:       0.52
Libya                     производство:      31.44 | потребление:      25.73
Madagascar                производство:       2.11 | потребление:       1.99
Malawi                    производство:       1.71 | потребление:       1.25
Mali                      производство:       3.35 | потребление:       2.63
Mauritania                производство:       1.74 | потребление:       1.71
Mauritius                 производство:       2.93 | потребление:       2.76
Morocco                   производство:      36.96 | потребление:      32.40
Mozambique                производство:      18.32 | потребление:      12.95
Namibia                   производство:       1.61 | потребление:       4.04
Niger                     производство:       0.51 | потребление:       1.25
Nigeria                   производство:      31.50 | потребление:      26.59
Reunion                   производство:    нет данных | потребление:       0.00
```

```
Rwanda                    производство:       0.88 | потребление:       0.76
Saint Helena              производство:       0.01 | потребление:       0.01
Sao Tome and Principe     производство:       0.09 | потребление:       0.08
Senegal                   производство:       5.03 | потребление:       4.70
Seychelles                производство:       0.55 | потребление:       0.52
Sierra Leone              производство:       0.22 | потребление:       0.14
Somalia                   производство:       0.37 | потребление:       0.34
South Africa              производство:     229.32 | потребление:     200.65
Sudan                     производство:      16.10 | потребление:      12.91
Tanzania                  производство:       7.60 | потребление:       6.65
Togo                      производство:       0.65 | потребление:       1.25
Tunisia                   производство:      19.95 | потребление:      15.99
Uganda                    производство:       4.28 | потребление:       2.95
Western Sahara            производство:       0.00 | потребление:       0.00
Zambia                    производство:      15.61 | потребление:      12.86
Zimbabwe                  производство:       8.08 | потребление:       8.11
Armenia                   производство:       7.36 | потребление:       5.74
Azerbaijan                производство:      24.41 | потребление:      21.36
Belarus                   производство:      37.38 | потребление:      34.29
Georgia                   производство:      11.73 | потребление:      12.07
Kazakhstan                производство:     100.98 | потребление:      91.71
Kyrgyzstan                производство:      15.22 | потребление:      12.11
Moldova                   производство:       5.78 | потребление:       6.62
Russia                    производство:    1059.99 | потребление:     946.68
Tajikistan                производство:      19.48 | потребление:      14.91
Turkmenistan              производство:      21.01 | потребление:      14.91
Ukraine                   производство:     146.95 | потребление:     127.12
Uzbekistan                производство:      58.55 | потребление:      54.57
Albania                   производство:       6.48 | потребление:       6.62
Austria                   производство:      65.09 | потребление:      67.46
Belgium                   производство:      85.01 | потребление:      84.02
Bosnia and Herzegovina    производство:      16.66 | потребление:      11.59
Bulgaria                  производство:      39.57 | потребление:      30.44
Croatia                   производство:      12.82 | потребление:      16.32
Cyprus                    производство:       4.81 | потребление:       4.61
Czechia                   производство:      77.71 | потребление:      61.32
Denmark                   производство:      30.59 | потребление:      35.00
Estonia                   производство:       8.43 | потребление:       8.57
Faroe Islands             производство:       0.37 | потребление:       0.35
Finland                   производство:      67.73 | потребление:      83.25
France                    производство:     533.22 | потребление:     445.22
```

Germany	производство:	581.56	потребление:	520.23
Gibraltar	производство:	0.20	потребление:	0.20
Greece	производство:	48.22	потребление:	51.36
Hungary	производство:	32.20	потребление:	41.79
Iceland	производство:	19.09	потребление:	18.53
Ireland	производство:	30.85	потребление:	28.82
Italy	производство:	275.26	потребление:	296.44
Latvia	производство:	6.12	потребление:	6.76
Lithuania	производство:	3.29	потребление:	11.29
Luxembourg	производство:	0.72	потребление:	6.45
Malta	производство:	1.94	потребление:	2.41
Montenegro	производство:	2.96	потребление:	2.73
Netherlands	производство:	116.18	потребление:	113.16
North Macedonia	производство:	5.26	потребление:	6.38
Norway	производство:	147.21	потребление:	125.66
Poland	производство:	154.01	потребление:	151.15
Portugal	производство:	52.37	потребление:	48.41
Romania	производство:	56.03	потребление:	49.56
Serbia	производство:	33.95	потребление:	29.68
Slovakia	производство:	25.97	потребление:	26.35
Slovenia	производство:	15.23	потребление:	13.65
Spain	производство:	259.34	потребление:	240.37
Sweden	производство:	162.24	потребление:	129.96
Switzerland	производство:	61.48	потребление:	56.10
Turkiye	производство:	295.93	потребление:	263.49
United Kingdom	производство:	310.06	потребление:	303.54
Afghanistan	производство:	1.03	потребление:	5.38
American Samoa	производство:	0.17	потребление:	0.16
Australia	производство:	249.40	потребление:	238.60
Bangladesh	производство:	75.73	потребление:	71.11
Bhutan	производство:	8.29	потребление:	2.48
Brunei	производство:	4.33	потребление:	3.87
Burma	производство:	22.50	потребление:	17.84
Cambodia	производство:	8.06	потребление:	9.47
China	производство:	7190.02	потребление:	6846.21
Cook Islands	производство:	0.04	потребление:	0.04
Fiji	производство:	1.10	потребление:	1.03
French Polynesia	производство:	0.64	потребление:	0.62
Guam	производство:	1.75	потребление:	1.67
Hong Kong	производство:	34.48	потребление:	45.01
India	производство:	1595.25	потребление:	1330.94
Indonesia	производство:	279.32	потребление:	256.33
Japan	производство:	983.88	потребление:	940.32

Kiribati	производство:	0.03	потребление:	0.03
Laos	производство:	34.78	потребление:	5.33
Macau	производство:	0.82	потребление:	5.45
Malaysia	производство:	160.13	потребление:	146.49
Maldives	производство:	0.63	потребление:	0.61
Mongolia	производство:	6.41	потребление:	7.13
Nauru	производство:	0.04	потребление:	0.03
Nepal	производство:	5.68	потребление:	6.80
New Caledonia	производство:	3.14	потребление:	3.08
New Zealand	производство:	43.43	потребление:	40.48
Niue	производство:	0.00	потребление:	0.00
North Korea	производство:	14.55	потребление:	12.24
Pakistan	производство:	138.91	потребление:	121.10
Papua New Guinea	производство:	4.52	потребление:	4.22
Philippines	производство:	102.19	потребление:	92.84
Samoa	производство:	0.16	потребление:	0.15
Singapore	производство:	51.27	потребление:	50.70
Solomon Islands	производство:	0.11	потребление:	0.09
South Korea	производство:	558.36	потребление:	539.51
Sri Lanka	производство:	15.49	потребление:	14.17
Taiwan	производство:	274.13	потребление:	265.07
Thailand	производство:	184.13	потребление:	196.75
Timor-Leste	производство:	0.49	потребление:	0.39
Tonga	производство:	0.07	потребление:	0.06
U.S. Pacific Islands	производство:	0.51	потребление:	0.49
Vanuatu	производство:	0.07	потребление:	0.06
Vietnam	производство:	220.20	потребление:	206.12
Wake Island	производство:	0.00	потребление:	0.00
Bahrain	производство:	29.67	потребление:	29.24
Iran	производство:	320.40	потребление:	282.51
Iraq	производство:	88.72	потребление:	50.19
Israel	производство:	70.58	потребление:	61.65
Jordan	производство:	20.03	потребление:	17.95
Kuwait	производство:	69.34	потребление:	62.73
Lebanon	производство:	20.09	потребление:	18.15
Oman	производство:	35.28	потребление:	31.66
Palestinian Territories	производство:	0.67	потребление:	5.86
Qatar	производство:	45.46	потребление:	42.68
Saudi Arabia	производство:	358.55	потребление:	324.52
Syria	производство:	16.74	потребление:	12.59
United Arab Emirates	производство:	129.26	потребление:	121.79
Yemen	производство:	3.46	потребление:	2.75

```
Bermuda                  производство:       0.59 | потребление:       0.55
Canada                   производство:     631.61 | потребление:     547.32
Greenland                производство:       0.53 | потребление:       0.52
Mexico                   производство:     324.34 | потребление:     285.83
Saint Pierre and Miquelon производство:       0.05 | потребление:       0.05
United States            производство:    4130.27 | потребление:    3957.50
Antarctica               производство:       0.00 | потребление:       0.00
Antigua and Barbuda      производство:       0.34 | потребление:       0.29
Argentina                производство:     139.20 | потребление:     124.66
Aruba                    производство:       0.91 | потребление:       0.75
Barbados                 производство:       1.02 | потребление:       0.97
Belize                   производство:       0.73 | потребление:       0.98
Bolivia                  производство:       9.78 | потребление:       8.90
Brazil                   производство:     614.18 | потребление:     539.93
British Virgin Islands   производство:       0.14 | потребление:       0.12
Cayman Islands           производство:       0.69 | потребление:       0.66
Chile                    производство:      80.96 | потребление:      77.38
Colombia                 производство:      78.57 | потребление:      73.11
Costa Rica               производство:      11.72 | потребление:       9.89
Cuba                     производство:      19.38 | потребление:      16.02
Dominica                 производство:       0.11 | потребление:       0.11
Dominican Republic       производство:      18.43 | потребление:      16.06
Ecuador                  производство:      30.56 | потребление:      25.56
El Salvador              производство:       5.93 | потребление:       6.31
Falkland Islands         производство:       0.02 | потребление:       0.02
French Guiana            производство:  нет данных | потребление:       0.00
Grenada                  производство:       0.21 | потребление:       0.19
Guadeloupe               производство:  нет данных | потребление:       0.00
Guatemala                производство:      13.69 | потребление:      11.39
Guyana                   производство:       1.18 | потребление:       0.94
Haiti                    производство:       0.99 | потребление:       0.36
Honduras                 производство:      10.78 | потребление:       7.61
Jamaica                  производство:       4.19 | потребление:       3.04
Martinique               производство:  нет данных | потребление:       0.00
Montserrat               производство:       0.02 | потребление:       0.02
Netherlands Antilles     производство:       0.87 | потребление:       0.74
Nicaragua                производство:       4.49 | потребление:       4.13
Panama                   производство:      10.78 | потребление:       9.08
Paraguay                 производство:      50.72 | потребление:      11.79
Peru                     производство:      54.93 | потребление:      48.82
Puerto Rico              производство:      17.27 | потребление:      16.29
Saint Kitts and Nevis    производство:       0.21 | потребление:       0.17
```

```
Saint Lucia              производство:       0.36 | потребление:       0.33
Saint Vincent/Grenadines производство:       0.15 | потребление:       0.14
Suriname                 производство:       1.94 | потребление:       2.36
The Bahamas              производство:       2.05 | потребление:       1.93
Trinidad and Tobago      производство:       8.60 | потребление:       8.21
Turks and Caicos Islands производство:       0.25 | потребление:       0.24
U.S. Virgin Islands      производство:       0.62 | потребление:       0.57
Uruguay                  производство:      14.83 | потребление:      12.00
Venezuela                производство:      94.10 | потребление:      66.24
```

3. Напишите выражения, позволяющие получить ответ на следующие вопросы:

3.1. Суммарное (по всем странам) потребление электроэнергии за каждый

Решение:

Функция np.nansum считает сумму, игнорируя NaN (отсутствующие данные). Параметр axis=0 означает, что суммирование происходит по строкам, то есть по всем странам для каждого года.

```python
# 3.1
# суммарное потребление по годам

total_consumption_by_year = np.nansum(consumption_values, axis=0)

# формируем массив лет (начиная с 1992)
years = np.arange(1992, 1992 + total_consumption_by_year.shape[0])

print("\n3.1 Суммарное потребление по годам:")
# zip(years, total_consumption_by_year) объединяет значения лет и соответствующее потребление
for year, total in zip(years, total_consumption_by_year):
    print(f"{year}: {total:.2f}")
```

Результат работы программы:

```
3.1 Суммарное потребление по годам:
1992: 10569.02
1993: 10854.56
1994: 11104.65
1995: 11476.48
1996: 11805.87
1997: 12122.03
1998: 12419.81
1999: 12685.75
2000: 13230.25
2001: 13486.49
2002: 13938.78
2003: 14455.56
2004: 15128.10
2005: 15725.98
2006: 16438.21
2007: 17205.12
2008: 17477.47
2009: 17435.69
2010: 18748.16
2011: 19438.82
2012: 19918.25
2013: 20573.10
2014: 20981.24
2015: 21400.02
2016: 22022.69
2017: 22716.21
2018: 23530.92
2019: 23915.66
2020: 23959.68
2021: 25336.71
```

3.2. Максимальное количество электроэнергии, которое произвела одна страна за один год (указание: чтобы не учитывать отсутствующие и некорректные данные (nan) воспользуйтесь NaN-безопасной версией функции max, то есть nanmax).

Решение:

Функция np.nanmax находит максимальное значение, игнорируя NaN (отсутствующие данные).

```
# 3.2
# максимальное производство 1 страной за 1 год

max_generation_value = np.nanmax(generation_values)

# индексы строки (страна) и столбца (год), где достигнут максимум
country_idx, year_idx = np.unravel_index(np.nanargmax(generation_values), generation_values.shape)

# массив лет (тот же, что использовался выше)
years = np.arange(1992, 1992 + generation_values.shape[1])

# определяем страну и год для максимального значения
max_country = countries[country_idx]
max_year = years[year_idx]

print("\n3.2 Максимальное производство электроэнергии:")
print(f"Значение: {max_generation_value:.2f} млрд кВт*ч")
print(f"Страна: {max_country}")
print(f"Год: {max_year}")
```

Результат работы программы:

```
3.2 Максимальное производство электроэнергии:
Значение: 8151.52 млрд кВт*ч
Страна: China
Год: 2021
```

3.3. Список стран, которые производят более 500 млрд. кВт*ч электроэнергии ежегодно в среднем за последние 5 лет (воспользуйтесь массивом, полученным на шаге 2).

Решение:

```
# 3.3
# страны, производящие > 500 млрд кВт*ч в среднем за последние 5 лет

high_producers = countries[generation_mean > 500] # используем булевую маску
print("\n3.3 >500 млрд кВт*ч:", high_producers) # вывод массива только тех стран, где условие выполняется
```

Результат работы программы:

```
3.3 >500 млрд кВт*ч: ['Russia' 'France' 'Germany' 'China' 'India' 'Japan' 'South Korea'
 'Canada' 'United States' 'Brazil']
```

3.4. 10% стран, которые потребляют больше всего электроэнергии ежегодно в среднем за последние 5 лет (указание: вначале определите соответствующую квантиль в массиве, построенном на шаге 2).

Решение:

Функция np.nanquantile вычисляет квантиль, в нашем случае это 90-й квантиль.

```
# 3.4
# 10% стран с наибольшим потреблением

# 90 квантиль – порог, выше которого находятся 10% самых больших значений
q_90 = np.nanquantile(consumption_mean, 0.9)
top10_consumers = countries[consumption_mean >= q_90]
print("\n3.4 Топ 10% по потреблению:", top10_consumers)
```

Результат работы программы:

```
3.4 Топ 10% по потреблению: ['South Africa' 'Russia' 'France' 'Germany' 'Italy' 'Spain' 'Turkiye'
 'United Kingdom' 'Australia' 'China' 'India' 'Indonesia' 'Japan'
 'South Korea' 'Taiwan' 'Vietnam' 'Iran' 'Saudi Arabia' 'Canada' 'Mexico'
 'United States' 'Brazil']
```

3.5. Список стран, которые увеличили производство электроэнергии в 2021 году по сравнению с 1992 годом более, чем в 10 раз.

Решение:

```
# 3.5
# страны, увеличившие производство > 10 раз с 1992 по 2021

# берем первый столбец (1992) и последний (2021)
# проверяем, у кого значение 2021 года больше в 10 раз
in_10x = countries[generation_values[:, -1] > 10 * generation_values[:, 0]]
print("\n3.5 Увеличили производство >10 раз:", in_10x)
```

Результат работы программы:

```
3.5 Увеличили производство >10 раз: ['Angola' 'Benin' 'Equatorial Guinea' 'Ethiopia' 'Lesotho' 'Mali'
 'Mauritania' 'Mozambique' 'Namibia' 'Sudan' 'Cambodia' 'China' 'Laos'
 'Maldives' 'Vietnam' 'Turks and Caicos Islands']
```

3.6. Список стран, которые в сумме за все годы потратили больше 100 млрд. кВт*ч электроэнергии и при этом произвели меньше, чем потратили.

Решение:

```
# 3.6
# страны, у которых потребление >100 и произведено меньше, чем потрачено

# суммируем по всем годам для каждой страны
total_generation = np.nansum(generation_values, axis=1)
total_consumption = np.nansum(consumption_values, axis=1)

# создаем булевую маску и выбираем страны, удовлетворяющие обоим условиям
less_produced_more_used = countries[(total_consumption > 100) & (total_generation < total_consumption)]
print("\n3.6 Потратили >100, произвели меньше:", less_produced_more_used)
```

Результат работы программы:

```
3.6 Потратили >100, произвели меньше: ['Zimbabwe' 'Belarus' 'Moldova' 'Belgium' 'Croatia' 'Finland' 'Hungary'
 'Italy' 'Latvia' 'Luxembourg' 'Netherlands' 'North Macedonia' 'Hong Kong']
```

3.7. Какая страна потратила наибольшее количество электроэнергии в 2020 году?

Решение:

Функция np.nanargmax возвращает индекс максимального значения.

```
# 3.7
# страна с максимальным потреблением в 2020 году

year_index_2020 = -2 # индекс -2 соответствует 2020 году
max_consumption_country_2020 = countries[np.nanargmax(consumption_values[:, year_index_2020])]
print("\n3.7 Больше всего потратила в 2020:", max_consumption_country_2020)
```

Результат работы программы:

```
3.7 Больше всего потратила в 2020: China
```

Задание 2 (6 баллов)

В файле «data2.csv» представлены данные наблюдений о прибыли (второй столбец) в зависимости от установленной скидки (первый столбец).

Комментарий. На самом деле эти данные сгенерированы синтетически, в учебных целях, но смысл значений в этой задаче не важен, и эти данные можно рассматривать как реальный датасет.

Бизнес-консультанты считают, что реальная зависимость прибыли от установленной скидки может быть описана квадратичной или кубической функцией (то есть полиномом второй или третьей степени, f(x) = a2x^2 +a1x+a0 или f(x) = a3x^3 +a2x^2 +a1x+a0).

1. Сформируйте систему линейных уравнений (СЛУ) для полинома 2й степени (для этого нужно выбрать 3 точки, в которых значение полинома должно совпадать с исходными данными; точки лучше выбирать равномерно «разбросанными» по исходным данным, то есть одна в начале имеющегося диапазона данных, одна в конце и одна в середине).

Решение:

```
# 1
# загрузка данных из файла

data = np.genfromtxt("data2.csv", delimiter = ";", skip_header = 1)

# проверка на пропущенные значения (NaN)
if np.isnan(data).any():
    data = data[~np.isnan(data).any(axis=1)] # если да, оставляем только те строки, где нет ни одного NaN

# разделяем данные
x = data[:, 0]  # первый столбец — скидка
y = data[:, 1]  # второй столбец — прибыль


# формируем систему уравнений для полинома 2-й степени

# выбираем 3 характерные точки из данных: начало, середина, конец
x_points2 = np.array([x[0], x[len(x)//2], x[-1]]) # скидки для 3х точек
y_points2 = np.array([y[0], y[len(y)//2], y[-1]]) # соответствующие прибыли

# составляем матрицу коэффициентов для системы линейных уравнений Ax = b:
# для полинома f(x) = a2*x^2 + a1*x + a0
# подставляем три выбранные точки, получаем три уравнения:
# [x1²  x1  1] [a2]   [y1]
# [x2²  x2  1] [a1] = [y2]
# [x3²  x3  1] [a0]   [y3]
A2 = np.array([
    [x_points2[0]**2, x_points2[0], 1],
    [x_points2[1]**2, x_points2[1], 1],
    [x_points2[2]**2, x_points2[2], 1]
])
b2 = y_points2  # вектор правых частей (прибыль)
```

2. Решите СЛУ (с помощью scipy.linalg.solve), тем самым найдя коэффициенты полинома.

Решение:

```
# 2
# решаем СЛУ для нахождения коэффициентов полинома

# solve(A, b) решает систему Ax = b, возвращает вектор [a2, a1, a0]
a2, a1, a0 = solve(A2, b2)
print("Коэффициенты квадратичного полинома:", a2, a1, a0)
```

Результат работы программы:

```
Коэффициенты квадратичного полинома: -1.377459087934028 13.790259964131947 -9.275618512309029
```

3. Получите вектор значений построенного полинома для заданных точек.

Решение:

```
# 3
# вычисляем предсказанные значения для всех точек

# подставляем все x в найденную формулу f(x) = a2*x^2 + a1*x + a0
y_pred2 = a2 * x**2 + a1 * x + a0
```

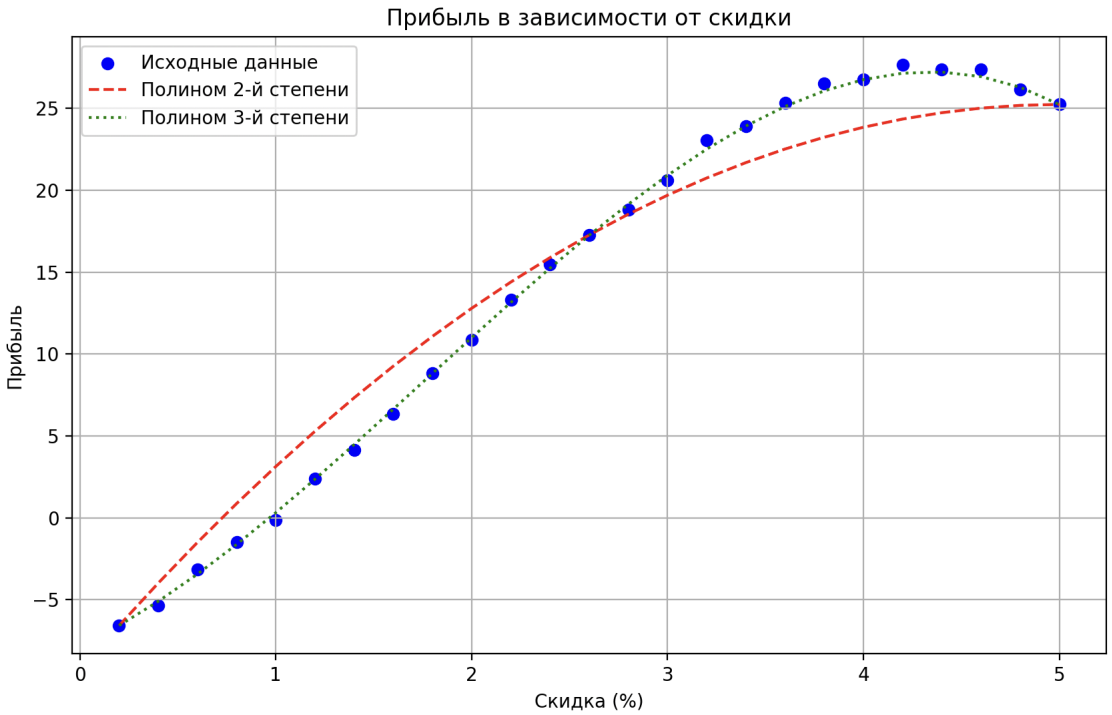
4. Постройте в одной области два графика: один по заданным в файле точкам, другой – по полученному вектору.

Решение:

```
# 4
# построение графиков

plt.figure(figsize=(10, 6))  # задаем размер графика (ширина × высота)
plt.scatter(x, y, color="blue", label="Исходные данные")  # точки исходных наблюдений
plt.plot(x, y_pred2, color="red", linestyle="--", label="Полином 2-й степени")
plt.plot(x, y_pred3, color="green", linestyle=":", label="Полином 3-й степени")
plt.title("Прибыль в зависимости от скидки") # заголовок графика
plt.xlabel("Скидка (%)") # подпись оси X
plt.ylabel("Прибыль") # подпись оси Y
plt.legend() # отображаем подписи линий
plt.grid(True) # включаем сетку
plt.show() # выводим график на экран
```

Результат работы программы:



5. Посчитайте значение квадратичного отклонения RSS (оно вычисляется по формуле , где yi – ожидаемые значения (из исходного файла), f(xi) – рассчитанные значения полинома.

Решение:

```
# 5
# считаем RSS

# RSS = sum(yi - f(xi))^2
# чем меньше RSS, тем лучше модель приближает данные
rss2 = np.sum((y - y_pred2)**2)
print("RSS для полинома 2-й степени:", rss2)
```

Результат работы программы:

```
RSS для полинома 2-й степени: 120.78752609259294
```

6. Повторите шаги 1-5 для полинома 3й степени (для этого нужно будет выбрать 4 точки).

Решение:

```
# 6
# формируем СЛУ для полинома 3-й степени

# для полинома f(x) = a3*x^3 + a2*x^2 + a1*x + a0
# нужно 4 точки (по числу коэффициентов)
x_points3 = np.array([x[0], x[len(x)//3], x[2*len(x)//3], x[-1]])
y_points3 = np.array([y[0], y[len(x)//3], y[2*len(x)//3], y[-1]])

# аналогично формируем матрицу коэффициентов
A3 = np.array([
    [x_points3[0]**3, x_points3[0]**2, x_points3[0], 1],
    [x_points3[1]**3, x_points3[1]**2, x_points3[1], 1],
    [x_points3[2]**3, x_points3[2]**2, x_points3[2], 1],
    [x_points3[3]**3, x_points3[3]**2, x_points3[3], 1]
])
b3 = y_points3  # прибыли для этих 4х точек

# решаем СЛУ (находим 4 коэффициента)
a3, a2_, a1_, a0_ = solve(A3, b3)
print("Коэффициенты кубического полинома:", a3, a2_, a1_, a0_)

# вычисляем предсказанные значения для всех x
y_pred3 = a3 * x**3 + a2_ * x**2 + a1_ * x + a0_

# считаем RSS для кубического полинома
rss3 = np.sum((y - y_pred3)**2)
print("RSS для полинома 3-й степени:", rss3)
```

Результат работы программы:

```
Коэффициенты кубического полинома: -0.5485425715332035 2.9039339294433626 5.811064836494135 -7.846646866904295
RSS для полинома 3-й степени: 1.8783514509456865
```

7. (по желанию). Можно поэкспериментировать с выбором точек для построения полиномов.

8. Выберите тот вариант, где значение отклонения (RSS) получается наименьшим. Для этого варианта посчитайте ожидаемое значение прибыли при значениях скидки в 6 и 8 процентов.

Решение:

```
# 8
# сравнение моделей

# выбираем ту, у которой меньше RSS (лучше приближение)
if rss2 < rss3:
    print("\nЛучшая модель: квадратичный полином")
    best = (lambda x_val: a2 * x_val**2 + a1 * x_val + a0)
else:
    print("\nЛучшая модель: кубический полином")
    best = (lambda x_val: a3 * x_val**3 + a2_ * x_val**2 + a1_ * x_val + a0_)


# прогноз прибыли при скидке 6% и 8%
for discount in [6, 8]:
    # вычисляем значение полинома в заданной точке
    print(f"Прибыль при скидке {discount}% = {best(discount):.2f}")
```

Результат работы программы:

```
Лучшая модель: кубический полином
Прибыль при скидке 6% = 13.08
Прибыль при скидке 8% = -56.36
```

Прибыль в денежных единицах.

9. Средняя прибыль из всего файла

Решение:

```
# 9
# средняя прибыль по всему файлу
mean_profit = np.mean(y)
print(f"\nСредняя прибыль по всем данным: {mean_profit:.2f}")
```

Результат работы программы:

Прибыль в денежных единицах.