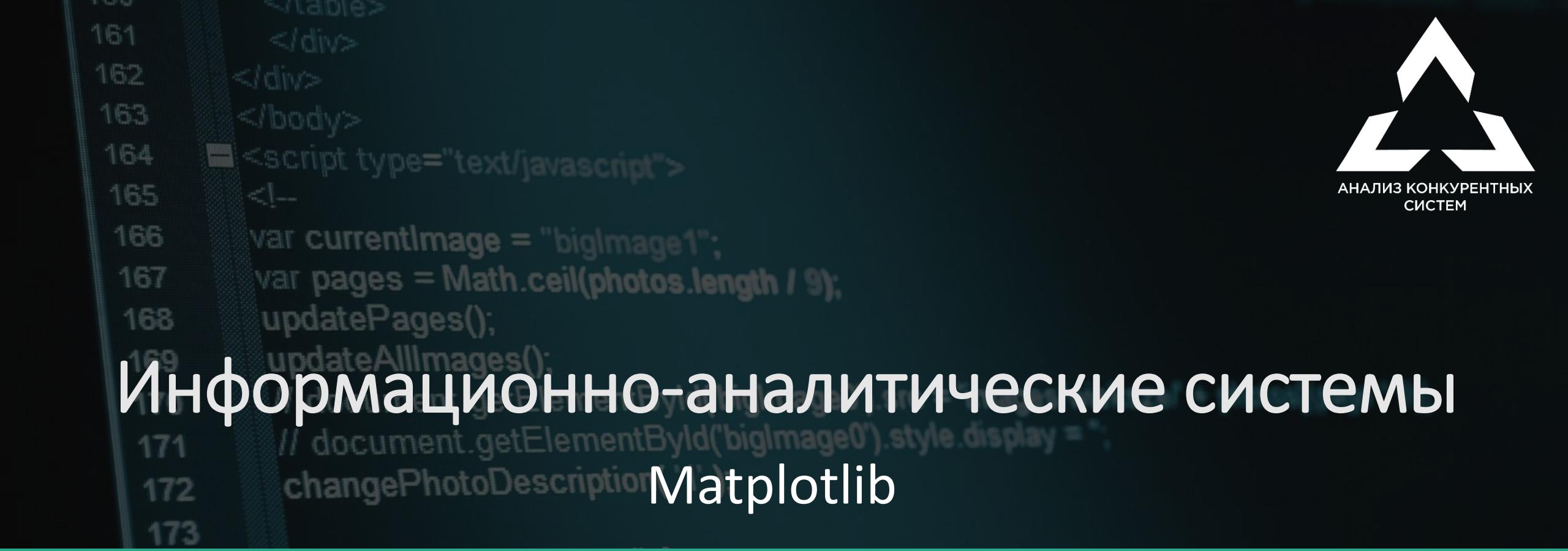


АНАЛИЗ КОНКУРЕНТНЫХ СИСТЕМ

Информационно-аналитические системы

Matplotlib

Цель

Овладеть навыками программирования для:

1. **Оптимизации рутинной работы**
2. **Подготовки аналитических документов**
3. **Сбора и обработки массивов данных**

Преподаватели

Артамонов Алексей Анатольевич

Улизко Михаил Сергеевич

Задача

Имеется файл формата xlsx с данными сотрудников некоторой фирмы. Требуется определить среднюю зарплату по предприятию, по каждой возрастной группе, по каждой профессии и в зависимости от пола. Также рекомендуется вывести график средней зарплаты в зависимости от возрастной группы

Фамилия Имя Отчество	пол	должность	Возраст	оклад
Карпов Алексей Юрьевич	М	уборщик	56	17000
Карпатова Аделина Васильевна	Ж	уборщик	55	16900

Задача

Шаг 1. Получение данных

- Ввести название файла
- Перечислить названия листов
- Открыть файл
- Работать с каждым листом

```
import openpyxl
input file = 'Data.xlsx'
shield_names = [ ’18-24’, ’25-29’,
        ’30-34’, ’35-39’, ’40-44’,
        ’45-49’, ’50-54’, ’55+’
]

source_book =
openpyxl.load_workbook[input_file]
for shield_name in shield_names:
        # TODO
        pass
```

Задача

Шаг 2. Инициализация изначальных данных

- Список для всех пользователей для получения средней зарплаты
- Список для средней зарплаты по каждой возрастной категории
- Данные с листа с текущей возрастной категорией
- Временная переменная для подсчета средней зарплаты

```
staff = []
average_salary_by_age = []

for shield_name in shield_names:
        data_shield =source_book[shield_name]
        rows = list(data_shield.rows)
        rows.pop(0)
        member_salaries_by_age = []
```

Задача

Шаг 3. Подсчет средней зарплаты по возрасту и заполнения штаба фирмы

- Обработка текущей записи в файле
- Запись текущей записи в общий штаб и зарплаты в список зарплаты по данной возрастной категории
- Вывод средней зарплаты по возрастной категории

```
from statistics import mean
for …
for row in rows:
        tmp_member = ( row[0].value,
                row[1].value, row[2].value,
                row[3].value, row[4].value)

        staff.append(tmp_member)
        member_salaries_by_age.append(tmp_memb
er[4])
print(‘age %s salary %f’ % (shield_name,
mean(member_salaries_by_age))
```

Задача

Временные результаты

- Записаны все данные в staff
- Выведена средняя зарплата по возрастной категории

```
# staff содержит все записи работников в кортеже вида:
# (имя, пол, должность, возраст оклад)

# average_salary_by_age содержит средние зарплаты по каждому возрасту
```

Задача

Шаг 4. получение средней зарплаты среди всех работников

- Использование mean
- Создание функции

```
all_salary = []
for member in staff:
        all_salary.append(member[4])
average = mean(all_salary)
```

```
def count_average_salary(staff):
    all_salary = []
    for member in staff:
        all_salary.append(member[4])
    average_salary = mean(all_salary)
    return(average_salary)
```

Задача

Шаг 5. получение средней зарплаты в зависимости от пола

- Структура кортежа в staff
- Две отдельные переменные для мужчин и женщин

```
def salary_by_gender(staff):
        men_salary = []
        women_salary = []
        for i in staff:
                if i[1] == 'M':
                        men_salary.append(i[4])
                else:
                        women_salary.append(i[4])
        return mean(men_salary), mean(women_salary)
```

Задача

Шаг 6. получение средней зарплаты в зависимости от профессии

```
def print_salary_by_position(staff):
    salary_by_position = {}
    for i in staff:
        tmp_position = i[2]
        if tmp_position in salary_by_position:
            salary_by_position[tmp_position].append(i[4])
        else:
            salary_by_position[tmp_position] = []
            salary_by_position[tmp_position].append(i[4])

    # Вывод всех значений средних зарплат
    # print(salary_by_position)

    # Сортировка по среднему значению зарплаты -> список
    salary_by_position = sorted(salary_by_position.items(),
key=lambda kv: mean(kv[1]))

    for i in salary_by_position:
        print('средняя зарплата: %25s %6.1f ' % (i[0], mean(i[1])))
```

Задача

Модуль matplotlib

Matplotlib — библиотека на языке программирования Python для визуализации данных двумерной графикой

Поддерживаемые типы графиков:

- line
- histogram
- pie chart
- и др.

```
import matplotlib.pyplot as plt


x = [0, 1, 2, 3, 4]
y = [1, 5, 7, 7, 9]
fig = plt.figure()
ax = fig.add_subplot(111)
ax.plot(x,y, ‘o--’, legend=‘Line’)
ax.set_legend(loc=0)
ax.set_xlabel(‘диапазон’)
ax.grid()
ax.set_title(‘График’)

plt.show()
```

Задача

Модуль matplotlib

Matplotlib — библиотека на языке программирования Python для визуализации данных двумерной графикой Поддерживаемые типы графиков:

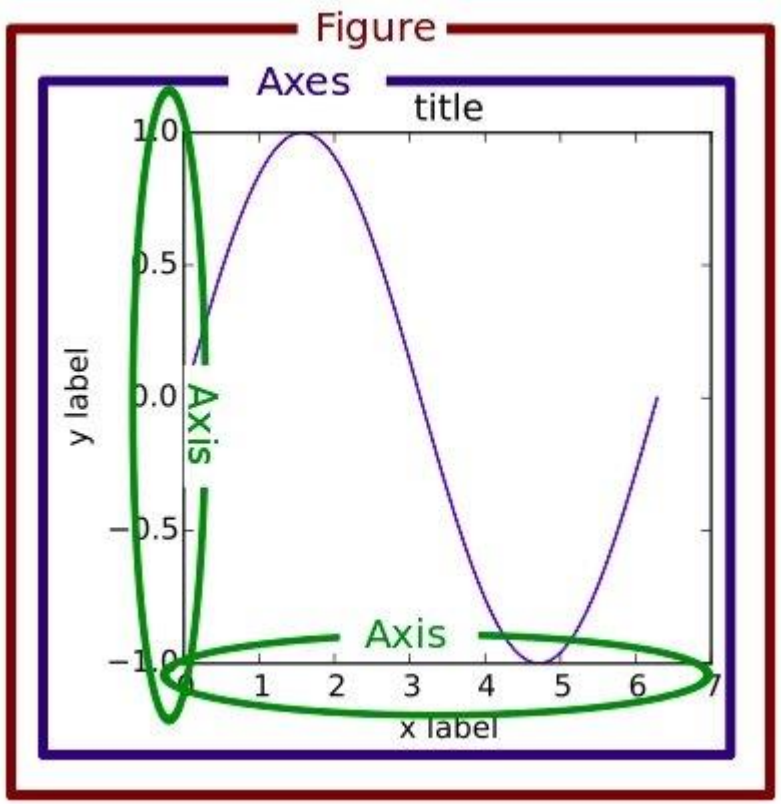
- **pie chart**
- histogram
- line
- и др.

```
import matplotlib.pyplot as plt


labels = ‘Муж’, ‘Жен’
y = [15, 22]
explode = (0.1, 0.1)
fig1, ax1 = plt.subplots()
ax1.pie(sizes, explode=explode, labels=labels,
autopct=‘%1.1f%%', shadow=True, startangle=90)
ax1.axis('equal')
ax2.set_title('3\п')

plt.show()
```

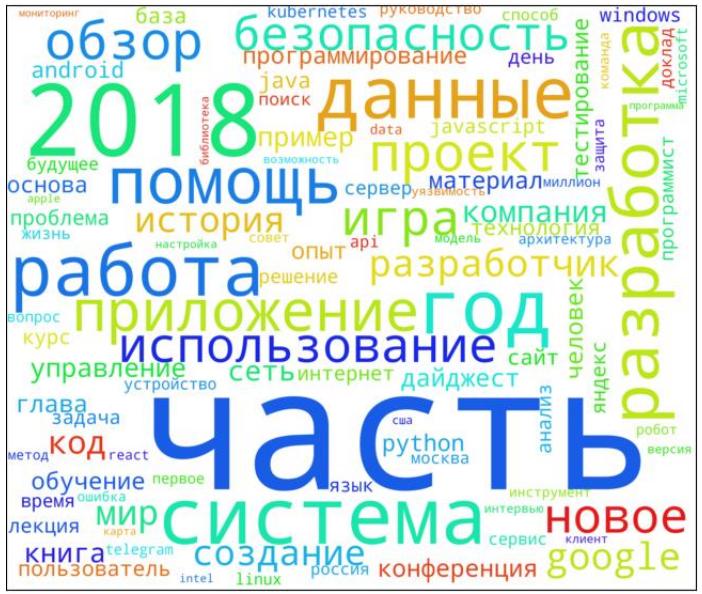
Элементы Pyplot




```
# Линейная зависимость
x = np.linspace(0, 10, 50)
y1 = x
# Квадратичная зависимость
y2 = [i**2 for i in x]
# Построение графиков
plt.figure(figsize=(9, 9))
plt.subplot(2, 1, 1)
plt.plot(x, y1) # построение графика
plt.title("Зависимости: y1 = x, y2 = x^2") # заголовок
plt.ylabel("y1", fontsize=14) # ось ординат
plt.grid(True) # включение отображение сетки
plt.subplot(2, 1, 2)
plt.plot(x, y2) # построение графика
plt.xlabel("x", fontsize=14) # ось абсцисс
plt.ylabel("y2", fontsize=14) # ось ординат
plt.grid(True) # включение отображение сетки
```

Wordcloud

wordcloud – модуль для создания облака слов




```
from wordcloud import Wordcloud

total_count = len(counte_obj)
common = counter_obj.most_common(total_count)
wc = WordCloud(width=2600, height=2200,
  background_color="white", relative_scaling=1.0, collocations=False,
  min_font_size=10).generate_from_frequencies(dict(common))
plt.axis("off")
plt.figure(figsize=(9, 6))
plt.imshow(wc, interpolation="bilinear")
```

recommendation

Домашнее задание

См. вложение