МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Измерительно-вычислительные комплексы»

«Методы искусственного интеллекта»

Исследование библиотек csv, pandas

Отчёт по лабораторной работе №3

Вариант №12

Выполнила:

студентка группы ИСТбд-42

Кучина Анна

Проверил:

доцент кафедры ИВК, к.т.н.

Шишкин В. В.

Ульяновск

2022

Цель работы: освоить базовые принципы работы с библиотеками pandas и CSV. Сравнить использование модуля pandas для расчета статистических характеристик с модулем numpy.

Задание:

1. Создать симулированный набор данных и записать его на диск в виде csv файла со следующими параметрами:

* + - * количество строк не менее 1000 (задается случайным образом);
      * структура набора:
      * табельный номер; (ид)
      * Фамилия И.О.; (просто рандом фамилия)
      * пол;
      * год рождения;
      * год начала работы в компании;
      * подразделение;
      * должность;
      * оклад;
      * количество выполненных проектов

2. Прочитать сгенерированный набор данных в виде списков и получить с помощью программирования и методов библиотеки numpy для разных по типу признаков столбцов (не менее 3) основные статистические характеристики (например для порядкового типа: минимум, максимум, среднее, дисперсия, стандартное отклонение, медиана, мода).

3. Прочитать сгенерированный набор данных в виде датафрейма и получить с помощью методов библиотеки pandas для тех же столбцов те же статистические характеристики. Продемонстрировать применение не менее 3 методов библиотеки pandas.

4. Построить не менее 3 разнотипных графиков.

Реализация:

1. Создание CSV файла:

# Массив заголовков

Titles = ['Id', 'Name', 'Sex', 'BirthDate', 'Start working', 'Store', 'Poste', 'Salary', 'Works']

# Массивы с данными для рандомизации

# Имена

Names = ['Brown', 'Smith', 'Johnson', 'Williams', 'Simons', 'Davis', 'Wilson', 'Taylor', 'Moore',

'Clark', 'Robinson']

# Пол

Sex = ['Male', 'Female']

# Отдел

Stores = ['Web', 'System developing', 'Mobile Developing', 'Support']

# Должность

Posts = ['TeamLead', 'database architect', 'network administrator', 'system engineer',

'android developer', 'ios developer', 'software tester', 'designer']

# Заполнение файла данными

with open('sw\_data\_new.csv', 'w') as f:

writer = csv.writer(f, lineterminator="\r")

writer.writerow(Titles)

for i in range(1, 1500):

name = random.choice(Names) + " " + random.choice(string.ascii\_uppercase) + "."

sex = random.choice(Sex)

birthdate = random.randrange(1967, 2002, 1)

stage = random.randrange(1, 15, 1) if birthdate > 21 else random.randrange(1, 3, 1)

store = random.choice(Stores)

poste = random.choice(Posts)

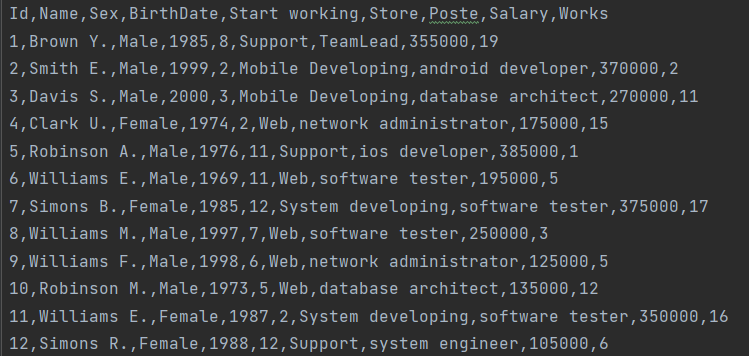
payment = random.randrange(90000, 400000, 5000)

works = random.randrange(0, 20, 1)

row = [i, name, sex, birthdate, stage, store, poste, payment, works]

writer.writerow(row)

Получившийся CSV файл (первые 12 строк)



1. Чтение файла в списки и расчет статистических характеристик с помощью методов библиотеки numpy:

# чтение данных из csv файла

my\_list = list()

with open('sw\_data\_new.csv') as f:

reader = csv.reader(f)

headers = next(reader)

for row in reader:

my\_list.append(row)

#Функция для расчета статистических характеристик с помощью библиотеки numpy

def np\_statistics(column, csv\_list, headers):

print('Для столбца ' + headers[column] + '\n')

stat = []

for row in csv\_list:

x = int(row[column])

stat.append(x)

statist = numpy.array(stat)

print('Минимальное значение: ' + str(numpy.min(statist)))

print('Максимальное значение: ' + str(numpy.max(statist)))

print('Математическое ожидание: ' + str(numpy.mean(statist)))

print('Стандартное отклонение: ' + str(numpy.std(statist)))

print('Дисперсия: ' + str(numpy.var(statist)))

print('Медиана: ' + str(numpy.median(statist)))

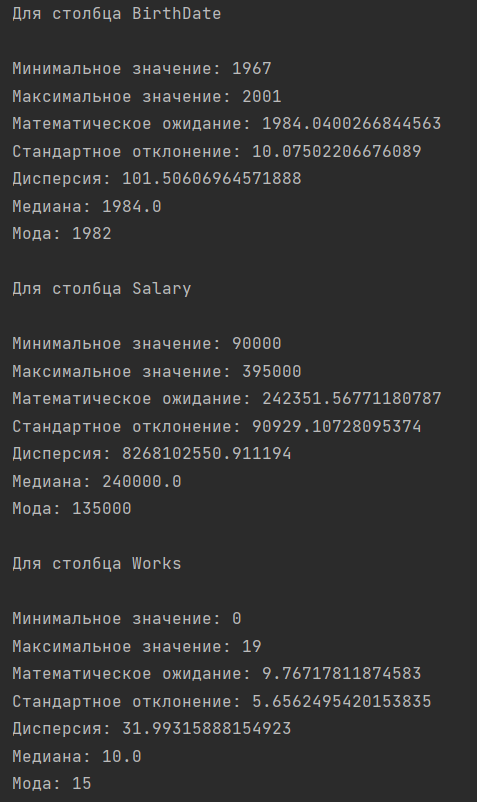
print('Мода: ' + str(mode(statist)) + '\n')

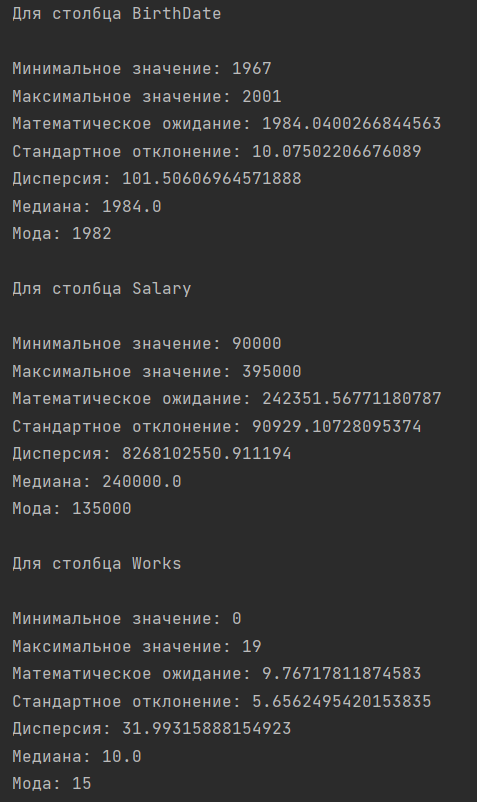
#Расчет статистических характеристик с помощью библиотеки numpy

np\_statistics(3, my\_list, headers)

np\_statistics(7, my\_list, headers)

np\_statistics(8, my\_list, headers)





1. Чтение из CSV файла и расчет статистических характеристик с помощью методов библиотеки pandas

def pandas\_statistics(dataframe, column):

print('Для ' + column + ':\n')

print('Минимальное значение: ' + str(dataframe[column].min()))

print('Максимальное значение: ' + str(dataframe[column].max()))

print('Математическое ожидание: ' + str(dataframe[column].mean()))

print('Стандартное отклонение: ' + str(dataframe[column].std()))

print('Дисперсия: ' + str(dataframe[column].var()))

print('Медиана: ' + str(dataframe[column].median()))

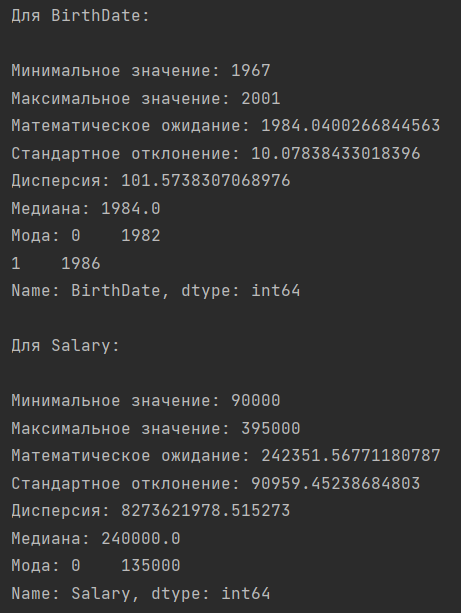
print('Мода: ' + str(dataframe[column].mode()) + '\n')

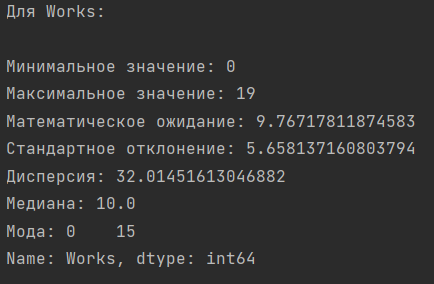
#Расчет статистических характеристик с помощью библиотеки pandas

pandas\_statistics(df, 'BirthDate')

pandas\_statistics(df, 'Salary')

pandas\_statistics(df, 'Works')





1. Построение разнотипных графиков:

#Построение графика зависимостей

data = [df["Sex"].value\_counts()["Male"], df["Sex"].value\_counts()["Female"]]

plt.pie(data, labels=["Мужчины", "Женщины"])

plt.title("Пол сотрудников компании")

plt.ylabel("")

plt.show()

#Построение круговой диаграммы

graf1 = df['Store'].hist()

plt.xlabel('Отдел')

plt.ylabel('Количество сотрудников')

plt.xticks(rotation=90)

plt.title("Количество сотрудников в отделах")

plt.show()

#Построение графика зависимости другого типа

plt.figure(figsize=(16, 10), dpi=80)

plt.plot\_date(df["Start working"], df["Salary"])

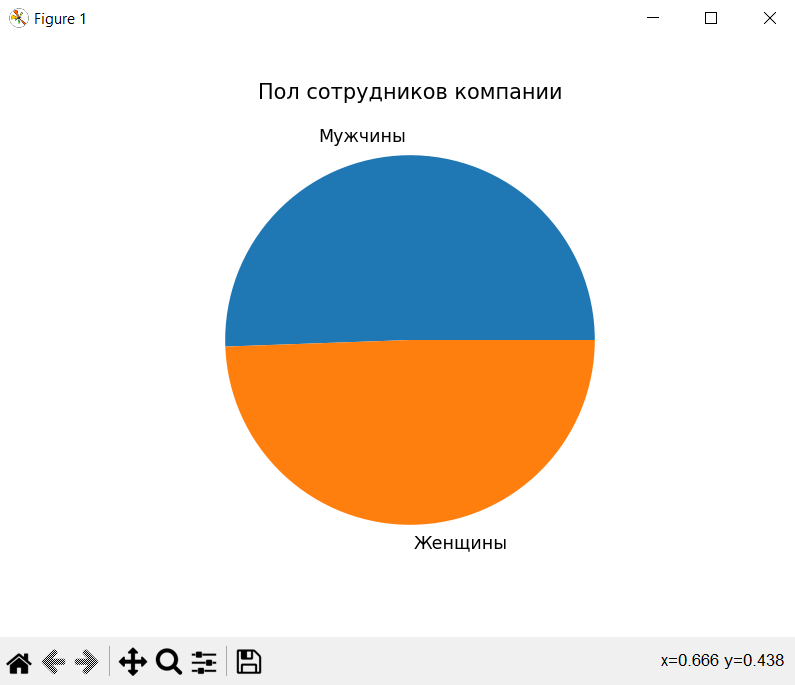
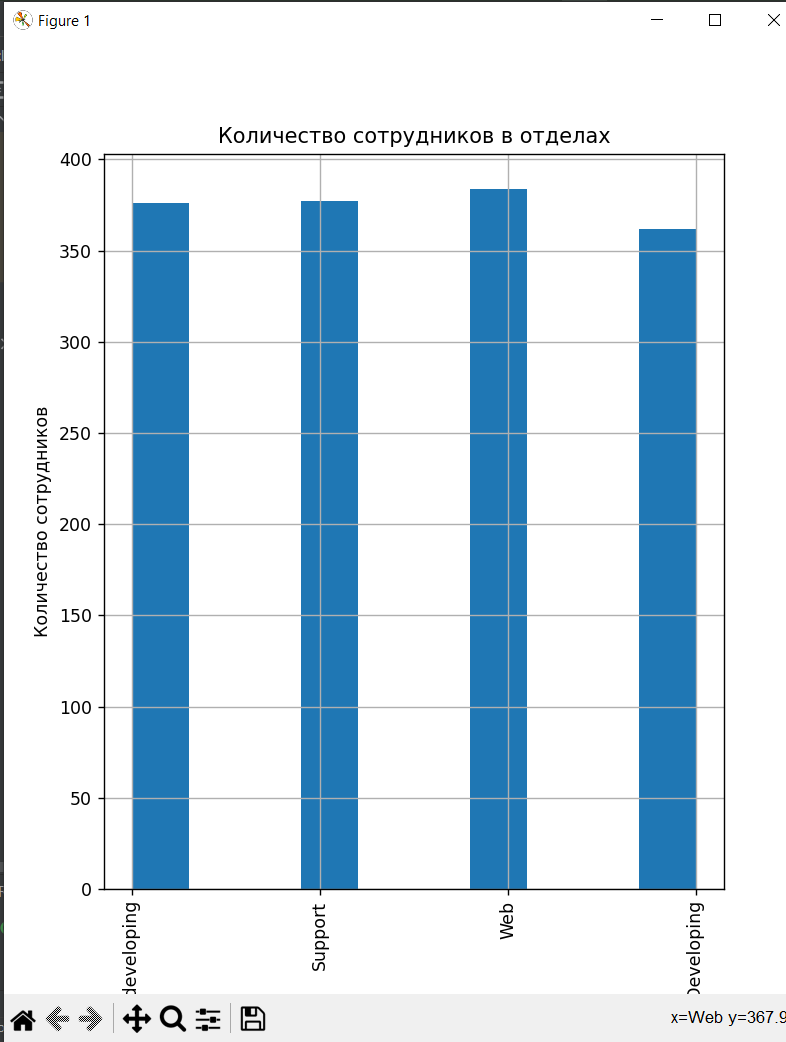
plt.gca().xaxis.set\_major\_locator(mdates.AutoDateLocator())

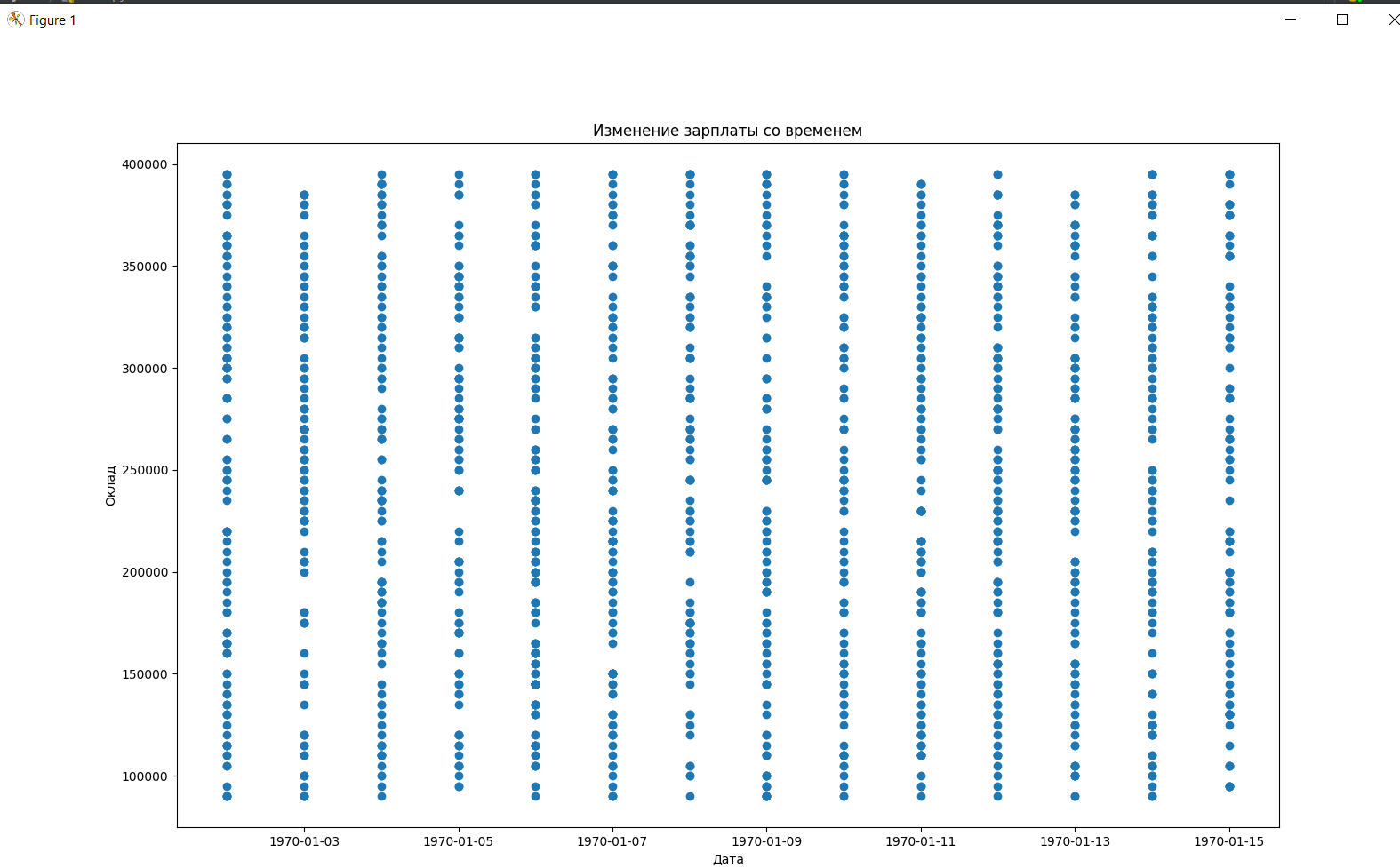
plt.ylabel('Оклад')

plt.xlabel('Дата')

plt.title("Изменение зарплаты со временем")

plt.show()



Вывод: с помощью методов библиотеки numpy можно произвести статистические расчеты для набора данных формата CSV, однако использование pandas.DataFrame является более простым и понятным решением той же проблемы, поскольку библиотека pandas содержит множество методов для работы с наборами данных подобного формата, а контейнер DataFrame упрощает чтение данных и их представление для дальнейшей обработки.