**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

###### ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

###### КЕМЕРОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

**ИНСТИТУТ ЦИФРЫ**

**ОТЧЁТ**

**О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**

«Pandas»

Студентки 2 курса, ФИТ-211 группы

**Колесник Полины Олеговны**

Направление 02.03.02 – «Фундаментальная информатика и информационные технологии»

Руководитель:

Доцент Зимин А. И.

Работа защищена

« »

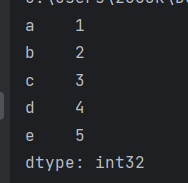
“ ” 2023 г.

Кемерово 2023 г.

**ОТЧЁТ О ПРОДЕЛАННОЙ РАБОТЕ**

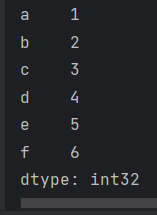
**1 задание**

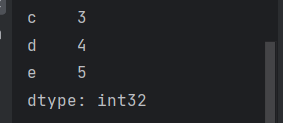
import pandas as pd  
import numpy as np  
  
# 1  
example1 = pd.Series(np.arange(1, 6), index=['a', 'b', 'c', 'd', 'e'])  
print(example1)

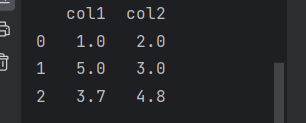
  
  
# 1  
print(example1['d'])

  
  
# 2  
print(example1.b)

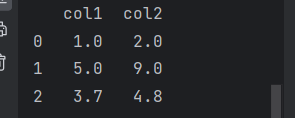
  
  
# 3  
example1['f'] = 6  
print(example1)

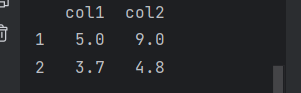
  
  
# 4  
print(example1['c':'e'])

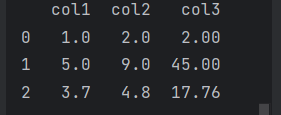
  
  
# 5  
example2 = pd.DataFrame([[1, 2], [5, 3], [3.7, 4.8]], columns=['col1', 'col2'])  
print(example2)

  
  
# 6  
print(example2.at[2, 'col1'])

  
  
# 7  
example2.at[1, 'col2'] = 9.0  
print(example2)

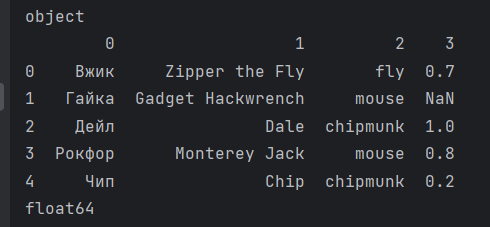
  
  
# 8  
print(example2[1:4])

  
  
# 9  
example2['col3'] = example2['col1'] \* example2['col2']  
print(example2)



**2 задание**

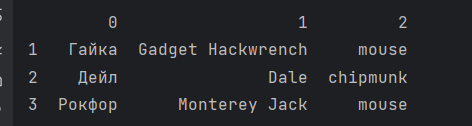
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
  
data = [["Вжик", "Zipper the Fly", "fly", "0.7"],  
 ["Гайка", "Gadget Hackwrench", "mouse", None],  
 ["Дейл", "Dale", "chipmunk", "1"],  
 ["Рокфор", "Monterey Jack", "mouse", "0.8"],  
 ["Чип", "Chip", "chipmunk", "0.2"]]  
  
df = pd.DataFrame(data)  
# df = pd.DataFrame(data, columns=['ru\_name', 'en\_name', 'class', 'cheer'])  
print(df[3].dtype)  
df[3] = df[3].astype(float)  
  
print(df)  
print(df[3].dtype)

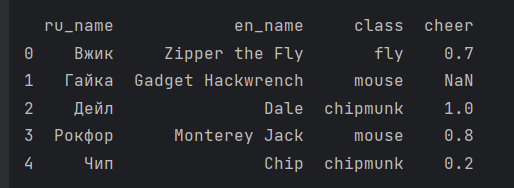
  
# 2  
print(len(df))

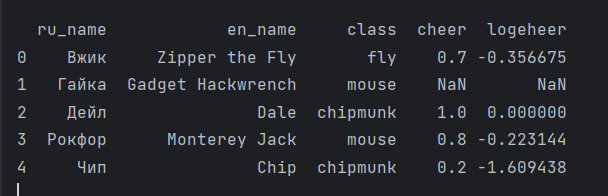
  
  
# 3  
print(df[3].count())

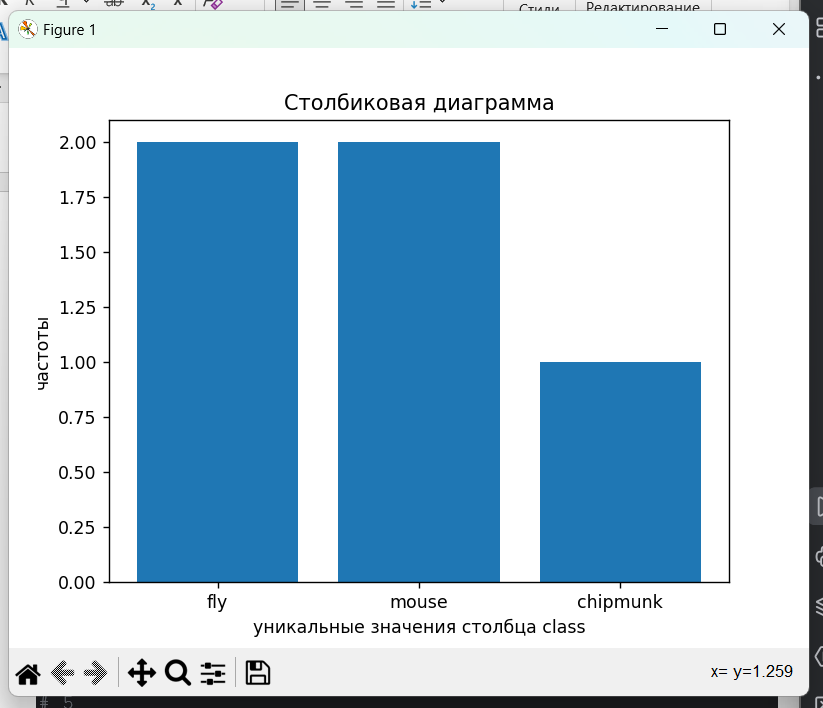
  
  
# 4  
print(df.loc[2, 1])

  
  
# 5  
df1 = df.iloc[1:4, 0:3]  
print(df1)

  
  
print()  
  
# 6  
df.columns = ['ru\_name', 'en\_name', 'class', 'cheer']  
print(df)  
  
print()

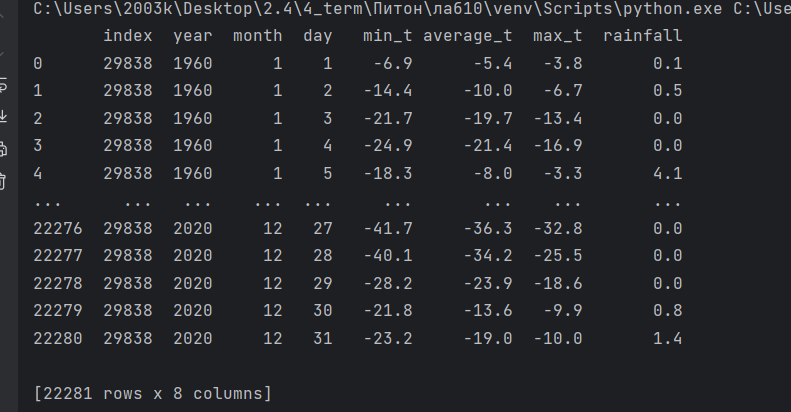
  
# 7  
df['logeheer'] = np.log(df['cheer'])  
print(df)

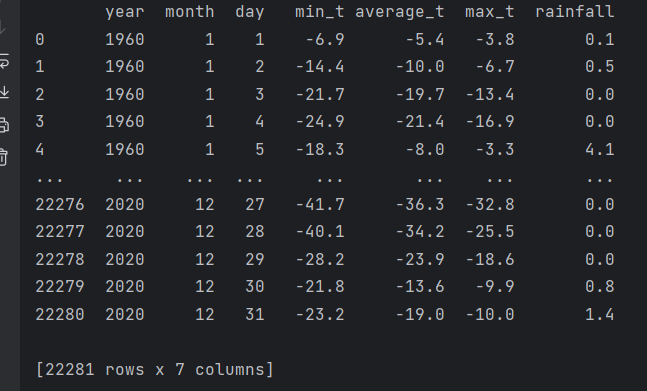
  
  
# 8  
x = df['class'].unique()  
y = df['class'].value\_counts()  
  
plt.bar(x, y)  
plt.title('Столбиковая диаграмма')  
plt.xlabel('уникальные значения столбца class')  
plt.ylabel('частоты')  
plt.show()

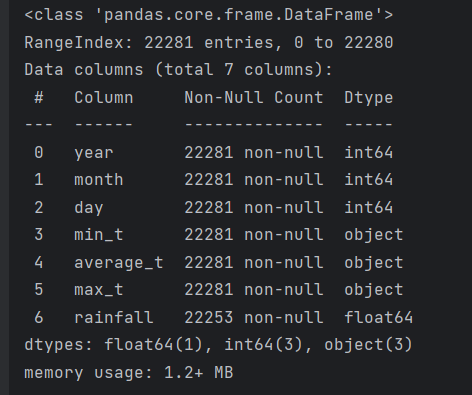


**3 задание**

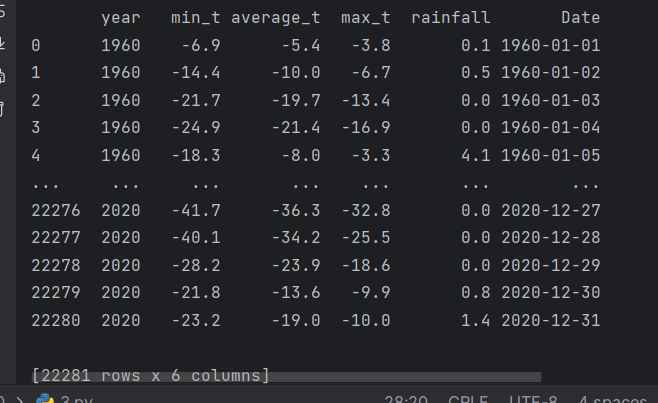
import numpy as np  
  
import pandas as pd  
  
df = pd.read\_csv('1.csv', delimiter=';',  
 names=['index', 'year', 'month', 'day', 'min\_t', 'average\_t', 'max\_t', 'rainfall'])  
print(df)  
  
print()

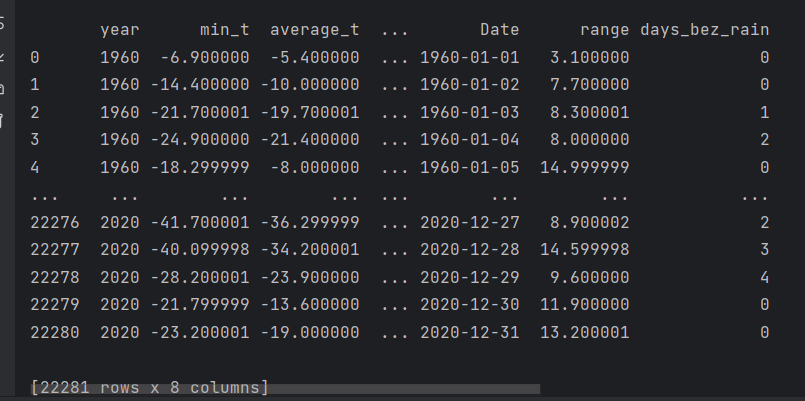
  
# 2  
df = df.drop(columns='index')  
print(df)  
print()

  
  
# 3  
df.info()  
print()  
  
# есть пропущенные значения и больше всего в столбце rainfall

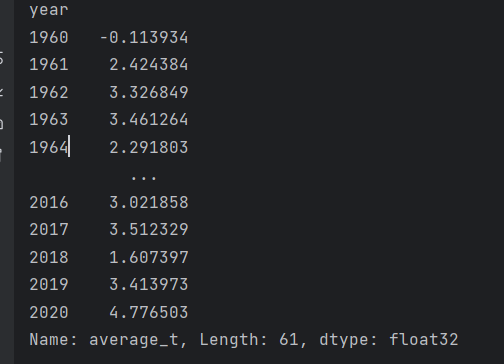
  
# 4  
year\_missing = df.groupby('year')['rainfall'].apply(lambda x: x.isnull().sum()).idxmax()  
print(year\_missing)  
  
print()

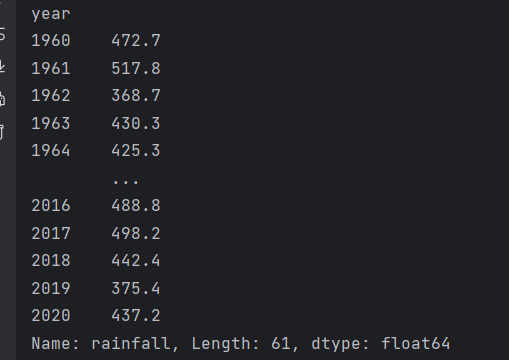
  
  
# 5  
df['Date'] = pd.to\_datetime(df[['year', 'month', 'day']], format="%Y-%m-%d")  
df = df.drop(columns=["month", "day"])  
print(df)  
  
print()

  
# 6  
# print(df.info())  
# print(df.dtypes.dtype)  
columns\_obj = df.dtypes[df.dtypes.map(lambda v: v == 'object')].index.values  
# print("Это колонки object-type:", columns\_obj)  
tmp = df[columns\_obj]  
tmp = tmp.applymap(func=lambda v: v.strip())  
tmp[tmp == ''] = np.nan  
df[columns\_obj] = tmp.astype(np.float32)  
# print("sum: ", df[columns\_obj].isna().sum())  
  
df['range'] = df['max\_t'] - df['min\_t']  
df['days\_bez\_rain'] = 0  
  
for i in range(1, len(df)):  
 if df.loc[i, 'rainfall'] == 0:  
 df.loc[i, 'days\_bez\_rain'] = df.loc[i - 1, 'days\_bez\_rain'] + 1  
  
print(df)  
print()

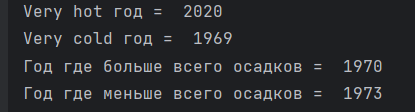
  
# 7  
max\_zasuha = df['days\_bez\_rain'].max()  
print(max\_zasuha)

  
  
# 8  
sredne\_godovay = df.groupby('year')['average\_t'].mean()  
kol\_vo\_osadkov = df.groupby('year')['rainfall'].sum()  
print(sredne\_godovay)

  
print(kol\_vo\_osadkov)



# print(df)  
  
# a  
print('Very hot год = ', sredne\_godovay.idxmax())  
print('Very cold год = ', sredne\_godovay.idxmin())  
  
# b  
print('Год где больше всего осадков = ', kol\_vo\_osadkov.idxmax())  
print('Год где меньше всего осадков = ', kol\_vo\_osadkov.idxmin())

  
  
# 9  
print(df[df['average\_t'] < -40])  
print(df[(df['average\_t'] > 27) & (df['days\_bez\_rain'] > 3)])

