МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Кафедра вычислительной математики и программирования

**Дисциплина: «Разработка ПО для высокопроизводительных систем»**

**Отчет по лабораторной работе №2 по Pandas**

Выполнил: Павлов М.М.

Группа: М8О-103М-20

Преподаватель: Поповкин А.В.

import pandas as pd

import numpy as np

import sys

import matplotlib

import re

def run():

#1

print("--------------------------1")

a = pd.Series(['asdasdasd', 'zxczxczxczxc', '11111111222'])

print(a.describe())

print()

#2

print("--------------------------2")

df = pd.DataFrame(np.random.randint(1, 7, 6000), columns=['one'])

df['ssss'] = df['one'] + np.random.randint(1, 7, 6000)

ax = df.plot.hist(bins=12, alpha=0.5)

print(ax)

print()

#3

print("--------------------------3")

state = np.random.RandomState(42)

s = pd.Series(state.randint(low=1, high=5, size=[13]))

print(s.value\_counts())

s[~s.isin(s.value\_counts().index[:2])] = 'Other'

print(s)

print()

#4

print("--------------------------4")

dti = pd.date\_range(start='2019-01-01', end='2019-12-31', freq='B')

s = pd.Series(np.random.rand(len(dti)), index=dti)

ans1 = s[s.index.weekday == 2].sum()

print(ans1)

print()

ans2 = s.resample('M').mean()

print(ans2)

print()

#5

print("--------------------------5")

s = pd.Series(np.random.randint(low=1, high=10, size=[35]))

r = (7, 5)

df = pd.DataFrame(s.values.reshape(r))

print(df)

print()

#6

print("--------------------------6")

s = pd.Series(np.random.randint(low=1, high=10, size=[7]))

print(s)

ans2 = s[s % 3 == 0].index

print(ans2)

print()

#7

print("--------------------------7")

s = pd.Series(list('abcdefghijklmnopqrstuvwxyz'))

print(s)

p = [0, 4, 8, 14, 20, 10]

ans1 = s[p]

print(ans1)

print()

#8

print("--------------------------8")

s1 = pd.Series(range(5))

s2 = pd.Series(list('abcde'))

ans\_vertical = s1.append(s2)

ans\_horizontal = pd.concat([s1, s2], axis=1)

print(ans\_vertical)

print(ans\_horizontal)

print()

#9

print("--------------------------9")

s1 = pd.Series([5, 3, 2, 1, 4, 11, 13, 8, 7])

s2 = pd.Series([1, 5, 13, 2])

ans1 = np.asarray([np.where(i == s1)[0].tolist()[0] for i in s2])

print(ans1)

#10

print("--------------------------10")

s = pd.Series(np.random.randint(low=1, high=10, size=[10]))

print(s)

ans = pd.Series(s.unique())

print(ans)

#11

print("--------------------------11")

strSeries = pd.Series(["abcd", "efg", "hi"])

strSeries = strSeries.map(lambda x: x.upper())

print(strSeries)

#12

print("--------------------------12")

s = pd.Series(np.arange(5))

s = s.map(lambda x: str(x))

catString = s.str.cat(sep=" ")

print(catString, type(catString))

#13

print("--------------------------13")

print(strSeries)

lenSeries = strSeries.map(lambda x: len(x))

print(lenSeries)

print(np.sum(lenSeries))

#14

print("--------------------------14")

s = pd.Series(np.arange(5))

s = s.map(lambda x: str(x))

print(s, type(s[0]))

#15

print("--------------------------15")

n = 3

s = pd.Series(np.arange(8))

diff = s.diff(periods = n)

print(diff)

#16

s = pd.Series(['2020-11-16', '16 Nov 2020', '2020/11/16'])

dates = pd.to\_datetime(s)

print(dates)

s = pd.Series(['2020-11-16', '2020-11-17', '2020-12-31', '2021-01-01'])

dates = pd.to\_datetime(s, format="%Y.%m.%d")

print(dates)

#

print(dates.dt.year)

print(dates.dt.month)

print(dates.dt.day)

print(dates.dt.weekofyear)

print(dates.dt.dayofyear)

#17

print("--------------------------17")

words = pd.Series(['Мда', 'Капец', 'отстой', 'унылые', 'задания', 'раздражают', 'господи', 'хватит', 'нинада'])

print(words[words.map(lambda word: bool(re.match('([^aeiouyаоиеёэыуюя]\*[aeiouyаоиеёэыуюя]){2,}', word, re.I)))])

#18

print("--------------------------18")

emails = pd.Series(['123456@i.ru', '\_login@ru.name.com', 'login.3@i.c', 'login@.ru', 'login@com', 'log/in/@i.com'])

print(emails[emails.map(lambda email: bool(re.match('[A-Za-z0-9.\_%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}', email)))])

#19

print("--------------------------19")

s1 = pd.Series(np.random.randint(1, 6, 10), name = "Values")

s2 = pd.Series(np.random.choice(['A', 'B', 'C', 'D'], 10), name = "Mask")

print(pd.concat([s1, s2], axis = 1))

s1.groupby(s2).mean()