

Лабораторная работа № 4

Перед началом выполнения каждой лабораторной работы, необходимо выполнить код в двух ячейках ниже. В случае если модуль `datetime` не установлен нужно установить. В последнем принте нужно вывести свою фамилию и инициалы.

In [45]:

```
!whoami
```

rogozhina\home

In [46]:

```
from datetime import datetime

current_time = datetime.now()
print(current_time)
print("Rogozhina N.A.") # написать здесь свою фамилию и инициалы
```

2022-12-19 17:48:56.050099
Rogozhina N.A.

Задания

In [100]:

```
import pandas as pd
df = pd.read_csv('data.csv', sep = ';')
print('Для понимания, с чем мы работаем, выведем и изучим таблицу')
df
```

3	301	город Москва (за исключением Троицкого и Новомосковского округов)	19693300	Электроэнергия	руб/кВт.ч	01.07.2012	31.07.2012	да	газовая	с 07:00 по 10:00 ч (многотарифный учет)	на бытовые нужды	«Московский энергетический холдинг»
4	302	город Москва (за исключением Троицкого и Новомосковского округов)	19693301	Электроэнергия	руб/кВт.ч	01.07.2012	31.07.2012	да	газовая	с 17:00 по 21:00 ч (многотарифный учет)	на бытовые нужды	«Московский энергетический холдинг»
...
66	335	город Москва (за исключением Троицкого и Новомосковского округов)	19693685	Газ	руб/куб.м	15.01.2012	30.06.2012	да/нет	NaN	NaN	на приготовление пищи и нагрев воды с использованием	ООО «Московский энергетический холдинг»

Задание 1. В датафрейме `df` задана некоторая таблица. Написать функцию `get_rows_after_5(df, n)`, возвращающую датафрейм, в котором записано `n` строк, начиная с пятой сверху (включая 5-ю). Например, `get_row_after_5(df, 1)` должна вернуть только пятую строку, а `get_row_after_5(df, 2)` — 5-ю и 6-ю.

Внимание! Индексами (именами строк) могут быть не числа, а что угодно.

```
In [102]:
df = pd.read_csv('data.csv', sep = ';', index_col = 0)

def get_rows_after_5(df,n):
    return df[4:(4+n)]

get_rows_after_5(df, 4)
```

302	город Москва (за исключением Троицкого и Новомосковского административных округов)	19693301	Электроэнергия	руб/кВт.ч	01.07.2012	31.07.2012	да	газовая	с 17:00 по 21:00 ч (многотарифный учет)	на бытовые нужды	«Мосэнергосбыт»
313	город Москва (за исключением Троицкого и Новомосковского административных округов)	19693302	Отопление	руб/Гкал	01.09.2012	31.12.2012	да	NaN	NaN	NaN	ОАО «Мосэнергосбыт»
318	город Москва (за исключением Троицкого и Новомосковского административных округов)	19693303	Горячее водоснабжение	руб/куб.м	01.07.2012	31.08.2012	да	NaN	NaN	NaN	ОАО «Мосэнергосбыт»
319	город Москва (за исключением Троицкого и Новомосковского административных округов)	19693304	Горячее водоснабжение	руб/куб.м	01.09.2012	31.12.2012	да	NaN	NaN	NaN	ОАО «Мосэнергосбыт»

Задание 2. В датафрейме df задана некоторая таблица, её индексами являются целые числа, не обязательно идущие по порядку. Написать функцию between(df, n, m), возвращающую все строки этой таблицы, расположенные между строками с индексами n и m, включая строки с индексами n и m. Гарантируется, что строка с индексом n встречается раньше строки с индексом m.

```
In [103]:
df = pd.read_csv('data.csv', sep = ';')
def between(df,n,m):
    return df[n:m+1]

between(df,0,5)
```

Out[103]:

ID	Region	global_id	TariffItem	UnitOfMeasure	StartDate	EndDate	MeasureDeviceAvail	StoveType	ConsumptionTime	ConsumptionType
0	275	город Москва (за исключением Троицкого и Новомосковского административных округов)	Холодное водоснабжение	руб/куб.м	01.09.2012	31.12.2012	да	NaN	NaN	
1	279	город Москва (за исключением Троицкого и Новомосковского административных округов)	Электроэнергия	руб/кВт.ч	01.01.2012	30.06.2012	да	газовая	без дифференциации по времени потребления	на бытовые нужды
2	285	город Москва (за исключением Троицкого и Новомосковского административных округов)	Электроэнергия	руб/кВт.ч	01.01.2012	30.06.2012	да	газовая	с 07:00 по 10:00 ч (многотарифный учет)	на бытовые нужды
3	301	город Москва (за исключением Троицкого и Новомосковского административных округов)	Электроэнергия	руб/кВт.ч	01.07.2012	31.07.2012	да	газовая	с 07:00 по 10:00 ч (многотарифный учет)	на бытовые нужды
4	302	город Москва (за исключением Троицкого и Новомосковского административных округов)	Электроэнергия	руб/кВт.ч	01.07.2012	31.07.2012	да	газовая	с 17:00 по 21:00 ч (многотарифный учет)	на бытовые нужды
5	313	город Москва (за исключением Троицкого и Новомосковского административных округов)	Отопление	руб/Гкал	01.09.2012	31.12.2012	да	NaN	NaN	

Задание 3. Загрузите файл data.csv (приложен к лабораторной работе).

- 3.1. Определите типы данных в каждом столбце и объем используемой памяти таблицей.
- 3.2. Столбцы с датами StartDate и EndDate приведите к типу datetime64 (можно использовать функцию pd.to_datetime()).
- 3.3. Выберите из датафрейма агентство **ОАО «Мосэнергосбыт»** и подсчитайте среднее значение тарифа (TariffValue).
- 3.4. Подсчитайте для каждого агентства сумму, максимальное, минимальное, среднее, медианное значения тарифа (TariffValue).

In [98]:

```
#3.1
data = pd.read_csv('data.csv', sep = ';')
print(f"Типы данных в каждом столбце:\n{data.dtypes}\nИспользуемая память:\n{data.memory_usage()}")
```

```
Unnamed: 13      float64
dtype: object
```

```
Используемая память:
Index           128
ID              568
Region          568
global_id       568
TariffItem       568
UnitOfMeasure   568
StartDate        568
EndDate         568
MeasureDeviceAvail 568
StoveType        568
ConsumptionTime  568
ConsumptionTarget 568
Agency          568
TariffValue      568
Unnamed: 13      568
dtvne: int64
```

In [101]:

```
#3.2
data['StartDate'] = pd.to_datetime(data['StartDate'])
data['EndDate'] = pd.to_datetime(data['EndDate'])
print(f"Типы данных в таблице:\n{data.dtypes}")
```

```
Типы данных в таблице:
ID              int64
Region          object
global_id       int64
TariffItem       object
UnitOfMeasure   object
StartDate        datetime64[ns]
EndDate         datetime64[ns]
MeasureDeviceAvail object
StoveType        object
ConsumptionTime  object
ConsumptionTarget object
Agency          object
TariffValue      float64
Unnamed: 13      float64
dtype: object
```

In [104]:

```
#3.3
# Создадим табличку, где будут только данные по агентству "ОАО Мосэнергосбыт"
energ = data[data['Agency'] == 'ОАО «Мосэнергосбыт»']

#Из задания семинара мы изучили метод mean(), который нам как раз сейчас и поможет:
energ['TariffValue'].mean()
```

Out[104]:

2.57125

Данную рабочую строку делала себе для проверки

In [105]:

```
ag = list(set(data['Agency']))
res = pd.DataFrame({
    "Summa": 0.0,
    "Minimum": 0.0,
    "Maximum": 0.0,
    "Average": 0.0,
    "Median": 0.0}, index = ag)
res
```

Out[105]:

	Summa	Minimum	Maximum	Average	Median
ОАО «Мосэнерго» и ОАО «МТК»	0.0	0.0	0.0	0.0	0.0
ООО «Газпром межрегионгаз Москва»	0.0	0.0	0.0	0.0	0.0
ОАО «Мосэнергосбыт»	0.0	0.0	0.0	0.0	0.0
ОАО «Мосгаз»	0.0	0.0	0.0	0.0	0.0
ООО «Мосрегионгаз»	0.0	0.0	0.0	0.0	0.0
ОАО «МОЭК» и иные, кроме ОАО «Мосэнерго» и ОАО «МТК»	0.0	0.0	0.0	0.0	0.0
МГУП «Мосводоканал»	0.0	0.0	0.0	0.0	0.0

In [106]:

```
#3.4
# Создадим список с нашими агентствами:
ag = list(set(data['Agency']))

# Создадим пустую таблицу
res = pd.DataFrame({
    "Summa": 0.0,
    "Minimum": 0.0,
    "Maximum": 0.0,
    "Average": 0.0,
    "Median": 0.0}, index = ag)

# Для каждого агентства из списка посчитаем нужные нам величины:
for i in ag:
    df = data[data['Agency'] == i]
    tarif = df['TariffValue']
    tarif_list = list(tarif)

    res.at[i, "Summa"] = sum(tarif_list)
    res.at[i, "Average"] = tarif.mean()
    res.at[i, "Minimum"] = min(tarif_list)
    res.at[i, "Maximum"] = max(tarif_list)
    res.at[i, "Median"] = tarif.median()

res.head()
```

Out[106]:

	Summa	Minimum	Maximum	Average	Median
ОАО «Мосэнерго» и ОАО «МТК»	3366.70	84.62	1075.43	561.116667	541.455
ООО «Газпром межрегионгаз Москва»	13021.28	4.09	3370.00	723.404444	36.460
ОАО «Мосэнергосбыт»	82.28	0.67	4.03	2.571250	2.735
ОАО «Мосгаз»	1126.00	1126.00	1126.00	1126.000000	1126.000
ООО «Мосрегионгаз»	7.02	2.93	4.09	3.510000	3.510

In []: