I. Importing. Creating DF

1. (2б) Імпортувати бібліотеку pandas та numpy.

```
In [20]:
```

```
import pandas as pd
import numpy as np
```

1. (3б) Створити тестовий датафрейм зі списку, що містить записи пацієнтів про їхні прізвища, імена, дату народження, масу тіла та наявність медичного страхування. Одне зі значень зробить пустим. Ознайомтесь з типом кожної колонки.

In [21]:

```
last name first name birth date weight health insurance
     Smith
                 John 2006-06-04
                                 75.5
1
      Doe
                 Jane 2000-07-15
                                   65.2
                                                   False
2
             Charlie 1985-10-31
                                  95.0
     Brown
                           object
last name
first name
                          object
                  datetime64[ns]
birth date
weight
                         float64
health insurance
                          object
dtype: object
```

II. Reading from file

1. (7б) Завантажити файл missile_attacks_daily.csv з Kaggle. Ознайомитись з описом датасету. Додати файл до папки поточної сесії colab notebook на панелі ліворуч або на гугл диск (локально до проєкту, якщо працюєте у пайчармі). Зчитати датафрейм з файлу та вивести датафрейм на екран. Зверніть увагу на те, щоб всі колонки відповідали наведеним у описі типам даних.

```
In [22]:
```

```
csv = pd.read_csv("C:\\Users\\Artemii\\Downloads\\Telegram Desktop\\missile_attacks_daily
.csv")
pd.set_option('display.max_columns', None)

print(csv)
print()
print(csv.dtypes)
```

time start time end model

```
2024-03-24 2024-03-24 08:00
                                                X-101/X-555
           2024-03-24 2024-03-24 08:00
                                             Shahed-136/131
           2024-03-23 2024-03-23 08:00
                                             Shahed-136/131
3
           2024-03-23 2024-03-23 08:00
                                                      C-300
4
           2024-03-23
                            2024-03-23
                                                       ZALA
856 2022-10-01 12:00 2022-10-01 12:00
                                                   Orlan-10
857
           2022-09-30
                            2022-09-30
                                                   Orlan-10
858 2022-09-29 23:00 2022-09-29 23:59
                                             Shahed-136/131
859
           2022-09-28
                             2022-09-28
                                                       X-59
                             2022-09-28 Reconnaissance UAV
860
           2022-09-28
                              launch place \
0
                                  Engels-2
1
     Primorsko-Akhtarsk and Chauda, Crimea
2
          Chauda, Crimea and Kursk oblast
3
4
                                       NaN
                                       . . .
856
                                       NaN
857
                                       NaN
858
                                       NaN
859
                                       NaN
860
                                       NaN
                                            target \
0
                                           Ukraine
1
                                           Ukraine
2
                                           Ukraine
3
                                    Donetsk oblast
4
                                             south
                                               . . .
856
                                           Ukraine
857
                                           Ukraine
858
                  Odesa oblast and Mykolaiv oblast
859
    Dnipropetrovsk oblast and Zaporizhzhia oblast
860
                                           Ukraine
                                     destroyed details
                                                         carrier \
0
                                                   NaN 14 \times Tu-95MS
1
     {'Dnipropetrovsk oblast': 17, 'Kherson oblast'...
2
     {'Poltava oblast': 2, 'Kherson oblast': 3, 'Dn...
                                                                 NaN
3
                                                                 NaN
                                                   NaN
4
                                                                 NaN
                                                   NaN
                                                   . . .
856
                                                   NaN
                                                                 NaN
857
                                                   NaN
                                                                 NaN
858
                                                   NaN
                                                                 NaN
859
                                                   NaN
                                                               Su-35
860
                                                   NaN
                                                                 NaN
     launched destroyed
0
       29.0
                     18 kpszsu/posts/pfbid02E9itApFzseeXaKMzes3zT2DHau...
                      25 kpszsu/posts/pfbid02E9itApFzseeXaKMzes3zT2DHau...
1
         28.0
2
        34.0
                      31 kpszsu/posts/pfbid0Jo2zmM4BzAZfS91x3Eb5Aa6L7AF...
3
         4.0
                      0 kpszsu/posts/pfbid0Jo2zmM4BzAZfS91x3Eb5Aa6L7AF...
         2.0
                      2 PvKPivden/posts/pfbid02rMwctKZrMFpQUXBtK15WCD8...
          . . .
                     . . .
                                                   watch/?v=375517448001233
856
         1.0
                      1
          3.0
                       3 kpszsu/posts/pfbid0nELeXY4BpHrezv54Lk3gWPZJmcC...
857
          7.0
                      5 kpszsu/posts/pfbid021aK2TqryAuB8gGrDYMTv3R8Xiu...
858
                      4 kpszsu/posts/pfbid02sjn5m7kosAfsm5bSUzpPwbLuH3...
859
          5.0
                       3 kpszsu/posts/pfbid0grmBx6A5LbQ7PVTkSx7jTDgg4iU...
860
          3.0
[861 rows x 10 columns]
time_start
                      object
time_end
                      object
model
                     object
launch place
                      object
target
                     object
destroyed details
                     object
```

carrier object
launched float64
destroyed int64
source object
dtype: object

III. Data understanding

1. (2б) Виведіть на екран перші 10 рядків зчитаного датафрейму.

```
In [23]:
```

```
print(csv.head(10))
         time start
                             time end
                                                 model
         2024-03-24 2024-03-24 08:00
                                          X-101/X-555
         2024-03-24 2024-03-24 08:00 Shahed-136/131
1
         2024-03-23 2024-03-23 08:00 Shahed-136/131
3
         2024-03-23 2024-03-23 08:00
                                                C-300
         2024-03-23
                           2024-03-23
4
                                                 ZALA
  2024-03-22 02:00 2024-03-22 09:00
5
                                                 X - 59
  2024-03-22 02:00 2024-03-22 09:00
6
                                         X-47 Kinzhal
  2024-03-22 02:00 2024-03-22 09:00
7
                                                 X-22
  2024-03-22 02:00
                     2024-03-22 09:00
                                          X-101/X-555
8
9
  2024-03-22 02:00
                    2024-03-22 09:00 Shahed-136/131
                            launch place
                                                  target
0
                                Engels-2
                                                 Ukraine
1
  Primorsko-Akhtarsk and Chauda, Crimea
                                                 Ukraine
2
        Chauda, Crimea and Kursk oblast
                                                 Ukraine
3
                                     NaN Donetsk oblast
4
                                     NaN
                                                  south
5
                     Zaporizhzhia oblast
                                                    east
                           Tambov oblast
                                                 Ukraine
7
                           Rostov oblast
                                                 Ukraine
8
                             Caspian Sea
                                                 Ukraine
                      Primorsko-Akhtarsk
                                                 Ukraine
                                   destroyed details
                                                          carrier launched \
0
                                                  NaN 14 x Tu-95MS
                                                                         29.0
   {'Dnipropetrovsk oblast': 17, 'Kherson oblast'...
1
                                                                         28.0
                                                                NaN
2
   {'Poltava oblast': 2, 'Kherson oblast': 3, 'Dn...
                                                                NaN
                                                                         34.0
3
                                                                NaN
                                                                          4.0
4
                                                  NaN
                                                                NaN
                                                                          2.0
5
                                                  NaN
                                                              Su-34
                                                                          2.0
6
                                                  NaN 10 x MIG-31K
                                                                          7.0
7
                                                            Tu-22M3
                                                                          5.0
                                                  NaN
8
                                                  NaN 13 \times Tu-95MS
                                                                         40.0
   {'Odesa oblast': 14, 'Kherson oblast': 5, 'Dni...
                                                                         63.0
                                                                NaN
   destroyed
0
          18
             kpszsu/posts/pfbid02E9itApFzseeXaKMzes3zT2DHau...
1
          25
             kpszsu/posts/pfbid02E9itApFzseeXaKMzes3zT2DHau...
             kpszsu/posts/pfbid0Jo2zmM4BzAZfS91x3Eb5Aa6L7AF...
3
             kpszsu/posts/pfbid0Jo2zmM4BzAZfS91x3Eb5Aa6L7AF...
4
             PvKPivden/posts/pfbid02rMwctKZrMFpQUXBtK15WCD8...
5
             kpszsu/posts/pfbid0G2H3qnSqb4ZcDPgrD1by43GSeZ4...
6
             kpszsu/posts/pfbid0G2H3qnSqb4ZcDPgrD1by43GSeZ4...
7
             kpszsu/posts/pfbid0G2H3qnSqb4ZcDPgrD1by43GSeZ4...
8
          35
              kpszsu/posts/pfbid0G2H3qnSqb4ZcDPgrD1by43GSeZ4...
9
              kpszsu/posts/pfbid0G2H3qnSqb4ZcDPqrD1by43GSeZ4...
```

1. (2б) Отримайте інформацію про форму датафрейму.

```
In [24]:
```

1. (3б) Отримайте дані про кожну колонку з її назвою, типом даних та кількістю непустих значень.

In [25]:

```
print(csv.info())
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 861 entries, 0 to 860
Data columns (total 10 columns):
    Column
                       Non-Null Count Dtype
                       -----
0
   time_start
                       861 non-null object
1
   time_end
                       861 non-null object
   model
 2
                      861 non-null object
3 launch_place 466 non-null object
4 target 833 non-null object
5 destroyed_details 68 non-null object
 6 carrier 169 non-null object
 7
   launched
                      858 non-null
                                      float64
 8
                      861 non-null
                                      int64
   destroyed
 9 source
                       861 non-null object
dtypes: float64(1), int64(1), object(8)
memory usage: 67.4+ KB
None
```

1. (3б) Отримайте статистику для колонок числового типу.

In [26]:

```
print(csv.describe())
        launched destroyed
count 858.000000 861.000000
       9.083916
                   7.038328
mean
       12.747066 11.021002
std
min
        1.000000
                   0.000000
25%
        1.000000
                    1.000000
50%
        4.000000
                    2.000000
75%
       12.000000
                   10.000000
max
       96.000000
                   87.000000
```

1. (2б) Отримати кількість унікальних значень у кожній колонці.

In [27]:

```
print(csv.nunique())
                      436
time_start
                      482
time end
model
                       43
launch_place
                       85
                       79
target
                      68
destroyed_details
                       50
carrier
launched
                       57
destroyed
                      54
source
                      443
dtype: int64
```

1. (3б) Вивести на екран значення та кількість цих значень у колонці, яка має найменше унікальних значень (див. попереднє завдання).

```
In [28]:
```

```
value_counts = csv.nunrque().raxmrn()

value_counts = csv[column_min].value_counts()

print(value_counts)
```

model	
Shahed-136/131	284
X-59	97
X-101/X-555	47
Orlan-10	44
Lancet	44
C-300	42
Kalibr	42
Merlin-VR	29
Unknown UAV	24
X-22	23
Iskander-M	23
ZALA	21
Reconnaissance UAV	20
Supercam	20
X-31P	15
X-47 Kinzhal	15
Iskander-K	10 7
P-800 Oniks C-300 and C-400	6
X-101/X-555 and Kalibr	5
Unknown Missile	5
X-31	5
KAB	4
Mohajer-6	3
Orion	3
X-22 and X-32	2
C-400 and Iskander-M	2
C-300 and C-400 and Iskander-M	2
Iskander-M/KN-23	2
Orlan-10 and ZALA and Supercam	2
Shahed-136/131 and Lancet	1
X-59MK2	1
Granat-4	1
Iskander-M and Iskander-K	1
Orlan-10 and Supercam	1
X-35	1
Orlan-30	1
Orlan-10 and Orlan-30 and ZALA and Supercam	1
Orlan-10 and ZALA	1
X-59 and X-35	1
C-400 Takandar-M/VN-22/V-47 Kinghal	1 1
Iskander-M/KN-23/X-47 Kinzhal X-101/X-555 and X-22 and Kalibr	1
Name: count, dtype: int64	Τ.
mame. counte, acype. Intor	

- **1. (15**б) На основі отриманої інформації про датасет вище, опишіть по пунктах, що б ви виправили або на що б звернули увагу на кроці очищення даних і чому?
- 1. Багато значень відсутні та інтерпретуються як NaN.
- 2. Деякі колонки заповнені менш ніж на 10%.
- 3. Час у даті не завжди вказаний, іноді непослідовно навіть в одному рядку.
- 4. Більшість колонок мають тип **object**, хоча можна конкретизувати типи даних.
- 5. Там де значення очевидно може бути цілочисельним використовується float.

IV. Data Cleaning

1. (3б) Перетворити колонку launched у цілочислену (Int64). Перевірити, чи все пройшло, як очікувалось.

```
print(csv['launched'].head(10))
csv['launched'] = csv['launched'].astype('Int64')
print(csv['launched'].dtype)
0
     29.0
1
     28.0
2
     34.0
3
      4.0
4
      2.0
5
      2.0
6
      7.0
7
      5.0
8
    40.0
9
    63.0
Name: launched, dtype: float64
Int64
In [30]:
print(csv['launched'].head(10))
0
     29
1
     28
2
     34
3
4
      2
5
      2
6
      7
7
      5
8
     40
9
     63
Name: launched, dtype: Int64
```

1. (3б) Видалити колонки, де кількість непустих значень становить менше **30**% від загальної кількості записів. Можна вручну порахувати і тут тільки видалити. За автоматизацію розрахунку відстоків - додаткові **3** бали.

```
In [31]:
print(csv.columns.__len__())
```

```
print(csv.columns.__len__())
csv = csv.dropna(thresh=(0.3 * len(csv)), axis=1)
print(csv.columns.__len__())
10
```

8

1. (4б) Очистити датафрейм від дублікатів (дублюючихся рядків) та записів, де атака продовжувалась від'ємну або нульову кількість часу (що фізично неможливо). Не забудьте оновити індекс. Скільки рядків було видалено? Як ви це перевірили у коді?

```
In [32]:
```

```
shapeI = csv.shape[0]

csv = csv.drop_duplicates()
shapeII = csv.shape[0]

csv['time_start'] = pd.to_datetime(csv['time_start'], format='mixed')
csv['time_end'] = pd.to_datetime(csv['time_end'], format='mixed')

csv = csv[csv['time_end'] - csv['time_start'] > pd.Timedelta(0)]

csv = csv.reset_index(drop=True)
shapeIII = csv.shape[0]

print(f"Rows: {shapeI} -> {shapeII} -> {shapeIII}; \n{shapeI - shapeIII} rows removed")
```

```
Rows: 861 -> 861 -> 568; 293 rows removed
```

1. (36) Виведіть на екран всі рядки, де є хоч одне пропущене значення.

In [33]:

```
print(csv[csv.isnull().any(axis=1)])
            time start
                                  time end
                                                                    model
3
   2024-03-23 00:00:00 2024-03-23 08:00:00
                                                                    C-300
12 2024-03-21 03:00:00 2024-03-21 07:00:00
                                            Iskander-M/KN-23/X-47 Kinzhal
13 2024-03-19 00:00:00 2024-03-19 08:00:00
                                                                     ZATA
   2024-03-18 00:00:00 2024-03-18 08:00:00
                                                                     X-59
14
                                                        C-300 and C-400
16
   2024-03-18 00:00:00 2024-03-18 08:00:00
563 2022-10-06 09:00:00 2022-10-06 11:30:00
                                                                Mohajer-6
564 2022-10-06 00:00:00 2022-10-06 02:00:00
                                                          Shahed-136/131
565 2022-10-05 00:00:00 2022-10-05 02:00:00
                                                          Shahed-136/131
566 2022-10-02 00:00:00 2022-10-02 02:00:00
                                                          Shahed-136/131
567 2022-09-29 23:00:00 2022-09-29 23:59:00
                                                           Shahed-136/131
    launch place
                                           target launched destroyed \
3
                                   Donetsk oblast 4
                                                                   0
            NaN
12
            NaN
                                      Kyiv oblast
                                                                    2
13
            NaN
                                                         3
                                                                    3
                                     Odesa oblast
                                                        2
14
           NaN
                                      Sumy oblast
16
           NaN
                                   Kharkiv oblast
                                                        5
                                                                    0
563
                                  Mykolaiv oblast
           NaN
                                                         1
                                                                    1
                                                        9
564
                                                                    9
            NaN
                                            south
565
                                                        6
                                                                    6
            NaN
                                            south
                                                          7
                                                                    5
566
            NaN
                                  Mykolaiv oblast
                                                         7
567
            NaN Odesa oblast and Mykolaiv oblast
                                               source
3
    kpszsu/posts/pfbid0Jo2zmM4BzAZfS91x3Eb5Aa6L7AF...
12
    kpszsu/posts/pfbid024ZjSa6LyBpqHEUKgGXtEMfAnpt...
13
    PvKPivden/posts/pfbid02YdqSYzeueEu3uz4VwXaBwHE...
14
    kpszsu/posts/pfbid0J2aqMT3K6v9zqzLrzTJSNGyKek2...
16
    kpszsu/posts/pfbid0J2aqMT3K6v9zqzLrzTJSNGyKek2...
563 kpszsu/posts/pfbid02YPkepucHoDQbmGfZURk8ScqNyM...
564 kpszsu/posts/pfbid0cUxYK69qorY8Td3GYrEifznzmJd...
565 kpszsu/posts/pfbid0ijKPtn19ybY71UFsnyhNA7YCx9i...
566 kpszsu/posts/pfbid021aK2TqryAuB8gGrDYMTv3R8Xiu...
567 kpszsu/posts/pfbid021aK2TqryAuB8gGrDYMTv3R8Xiu...
[164 rows x 8 columns]
```

V. Data Wrangling

1. (9б) З'ясуйте, який середній відсоток збиття ракет за весь наведений період.

```
In [34]:
```

```
csv['launched'] = csv['launched'].fillna(0).astype(int)
csv['destroyed'] = csv['destroyed'].fillna(0).astype(int)
print(f"Average destroyed: {(csv['destroyed'].sum() / csv['launched'].sum()) * 100}%")
```

Average destroyed: 77.12070538157495%

1. (11б) З'ясуйте 10 найпопулярніших моделей запущених пристроїв за кількістю випущених одиниць за наведений період. Запишіть результат у файл типу **json**. Скопіюйте зміст файлу та вставте нижче.

In [35]:

```
top_10_models = csv['model'].value_counts().nlargest(10).to_dict()
import json
with open('top_10_models.json', 'w') as f:
    json.dump(top_10_models, f)

with open('top_10_models.json', 'r') as f:
    file_contents = f.read()

print(file_contents)
```

{"Shahed-136/131": 249, "X-59": 48, "X-101/X-555": 42, "C-300": 37, "Kalibr": 32, "Iskand er-M": 23, "X-22": 19, "Unknown UAV": 14, "Orlan-10": 13, "X-47 Kinzhal": 11}

1. (10б) З'ясуйте, у який день було найбільше влучань за весь період (незбитих пристроїв). Виведіть час початку атаки та кількість влучань.

In [36]:

```
csv['hits'] = csv['launched'] - csv['destroyed']
most_hits_date = csv.groupby(csv['time_start'].dt.date)['hits'].sum().idxmax()
attack_start_time = csv[csv['time_start'].dt.date == most_hits_date]['time_start'].min()
hits = csv[csv['time_start'].dt.date == most_hits_date]['hits'].sum()
print(f"Most hits on: {most_hits_date}")
print(f"Start time: {attack_start_time}")
print(f"Hits: {hits}")
```

Most hits on: 2024-03-22 Start time: 2024-03-22 02:00:00

Hits: 59

1. (15б) Знайдіть найбільш тривалу та найбільш коротку атаки за кожен представлений місяць. Результат отримайте у такому вигляді:

year	month	attack_duration
2024	1	0 days 12:00:00
2024	2	0 days 08:00:00
2024	3	0 days 08:00:00
2023	1	0 days 08:00:00
2023	2	0 days 14:10:00
2023	3	0 days 04:30:00
2023	4	1 days 10:30:00
2023	5	0 days 08:30:00
2023	6	0 days 08:00:00
2023	7	0 days 08:00:00
2023	8	0 days 08:00:00
2023	9	0 days 08:00:00
2023	10	0 days 09:00:00

2023	11	0 days 09:00:00
2023	12	0 days 13:00:00
2022	9	0 days 00:59:00
2022	10	0 days 10:00:00
2022	11	0 days 03:00:00
2022	12	0 days 09:15:00

Запишіть його у файл типу с**sv.** Після цього виведіть на екран.

8

10

11

12

13

2023

2023

2023

2023

2023

2023

6

7

8

9

10

11

0 days 02:30:00

0 days 01:30:00

0 days 00:30:00

0 days 05:30:00

0 days 01:00:00

0 dave 02.00.00

```
In [37]:
csv['attack duration'] = pd.to timedelta(csv['time end'] - csv['time start'])
csv['year'] = csv['time start'].dt.year
csv['month'] = csv['time start'].dt.month
result = csv.groupby(['year', 'month'])['attack duration'].agg(['min', 'max']).reset ind
result.rename(columns={'min': 'attack duration min', 'max': 'attack duration max'}, inpl
ace=True)
result = result.sort values(by=['year', 'month'], ascending=[False,True])
result = result.reset index(drop=True)
result.to csv('attack durations.csv', index=False)
print(open('attack durations.csv').read())
print(result)
year, month, attack duration min, attack duration max
2024,1,0 days 01:10:00,0 days 12:00:00
2024,2,0 days 08:00:00,0 days 08:00:00
2024,3,0 days 04:00:00,0 days 08:00:00
2023,1,0 days 03:00:00,0 days 08:00:00
2023,2,0 days 02:05:00,0 days 14:10:00
2023,3,0 days 02:10:00,0 days 04:30:00
2023,4,0 days 02:00:00,1 days 10:30:00
2023,5,0 days 00:30:00,0 days 08:30:00
2023,6,0 days 02:30:00,0 days 08:00:00
2023,7,0 days 01:30:00,0 days 08:00:00
2023,8,0 days 00:30:00,0 days 08:00:00
2023,9,0 days 05:30:00,0 days 08:00:00
2023,10,0 days 01:00:00,0 days 09:00:00
2023,11,0 days 02:00:00,0 days 09:00:00
2023,12,0 days 01:00:00,0 days 13:00:00
2022,9,0 days 00:59:00,0 days 00:59:00
2022,10,0 days 00:30:00,0 days 10:00:00
2022,11,0 days 01:00:00,0 days 03:00:00
2022,12,0 days 01:00:00,0 days 09:15:00
    year month attack duration min attack duration max
0
    2024
             1
                    0 days 01:10:00
                                        0 days 12:00:00
1
   2024
              2
                    0 days 08:00:00
                                        0 days 08:00:00
2
   2024
             3
                   0 days 04:00:00
                                        0 days 08:00:00
3
   2023
                  0 days 03:00:00
             1
                                        0 days 08:00:00
4
             2
   2023
                  0 days 02:05:00
                                        0 days 14:10:00
5
   2023
             3
                  0 days 02:10:00
                                        0 days 04:30:00
6
   2023
             4
                  0 days 02:00:00
                                        1 days 10:30:00
7
             5
   2023
                  0 days 00:30:00
                                        0 days 08:30:00
```

0 days 08:00:00

0 days 08:00:00

0 days 08:00:00

0 days 08:00:00

0 days 09:00:00

0 dave 09.00.00

	2020		$\overline{}$	uuyu	02.00.00	$\overline{}$	uuyu	02.00.00
14	2023	12	0	days	01:00:00	0	days	13:00:00
15	2022	9	0	days	00:59:00	0	days	00:59:00
16	2022	10	0	days	00:30:00	0	days	10:00:00
17	2022	11	0	days	01:00:00	0	days	03:00:00
18	2022	12	0	days	01:00:00	0	days	09:15:00

Вітаю! Ви велика(ий) молодець, що впоралась(вся). Похваліть себе та побалуйте чимось приємним. Я Вами пишаюся.

