

---

# Lab report 4 (Variant 2)

---

*Student :*

*Herashchenko Serhii*

*Teacher:*

*Zhuravlova Zinaida Yuriyivna*

*06.11. 2024 p.*

# Task 1

## 1.1 Condition

По даним з файлу vaccination\_process\_2021\_regions.xlsx побудувати стовбцеву діаграму кількості людей, що вакцинувались вакциною Pfizer-BioNTech у вересні, відносно області. За кількість вакцинованих у вересні брати суму вакцинованих за днями даного місяця, які є.

## 1.2 Solution code

```
import pandas as pd
import matplotlib.pyplot as plt

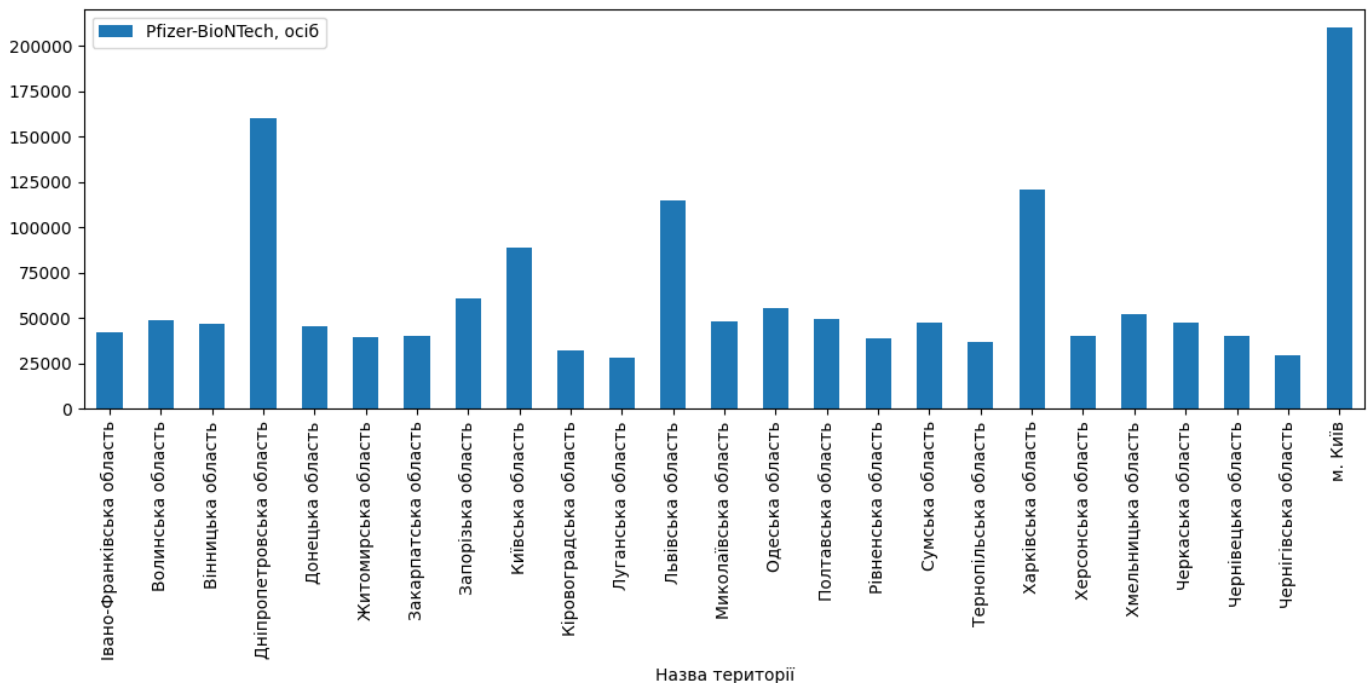
data =
pd.read_excel('C:\\ALL\\OTHER\\GitHub\\HOMEWORK\\Python\\L4_v2_Herashchenko\\1_task\\va
ccination_process_2021_regions.xlsx')

x = data[['Назва території', 'Дата (період) даних', 'Pfizer-BioNTech, оціб']]

people_september = x[pd.to_datetime(data["Дата (період) даних"],
dayfirst=True).dt.month==9].groupby('Назва території', as_index = False)["Pfizer-
BioNTech, оціб"].sum()

people_september.plot(kind='bar', x = "Назва території", figsize = (8,2))
plt.show()
```

## 1.3 Output



# Task 1

## 2.1 Condition

До даних з файлу filter2.xlsx додати стовбець Discount (уцінка), який визначається наступним чином: уцінку товару проводять, якщо до завершення строку придатності залишається менше за 3 дні або цей строк перевищено. Дані стовбця визначають, чи підлягає даний стовбець уцінці або ні.

1. Провести вибірки по наступним позиціям:
2. вибрати усі товари, що підлягли уцінці
3. визначити товари, що виготовлено у Польщі та підлягли уцінці
4. виділити товари Milk "Good health", що не підлягли уцінці
5. визначити товари, що виготовлено в Україні і не підлягли уцінці
6. знайти партії товарів, обсяг яких перевищує 20 одиниць і які підлягли уцінці
7. вибрати товари, що виготовлено у Польщі, обсяг яких перевищує 30 одиниць, а також товари, що виготовлено в Україні та які підлягли уцінці
8. вибрати товари, що виготовлено у Польщі, які коштують менша за 20 грн, а також товари, обсяг яких менше за 40 одиниць, що підлягли уцінці.

## 2.2 Solution code

```
import pandas as pd
from datetime import datetime, timedelta

data =
pd.read_excel('C:\\ALL\\OTHER\\GitHub\\HOMEWORK\\Python\\L4_V2_Herashchenko\\2_task\\filter2.xlsx')

today = datetime.now()

data['Discount'] = ((pd.to_datetime(data['Shelf_life']) - today) <= timedelta(days=3))
| (pd.to_datetime(data['Shelf_life']) < today)

discounted_items = data[data['Discount']]

discounted_polish_items = data[(data['CountryOfOrigin'] == 'Poland') &
(data['Discount'])]

milk_good_health_no_discount = data[(data['Product'] == "Milk 'Good health'") &
(data['Discount'] == False)]

ukraine_no_discount = data[(data['CountryOfOrigin'] == 'Ukraine') &
(~data['Discount'])]

large_discounted_batches = data[(data['Quantity'] > 20) & (data['Discount'])]

polish_large_volume_or_ukraine_discounted = data[((data['CountryOfOrigin'] == 'Poland')
& (data['Quantity'] > 30)) | ((data['CountryOfOrigin'] == 'Ukraine') &
(data['Discount']))]
```

```
polish_low_price_or_small_discounted_volume = data[((data['CountryOfOrigin'] ==
'Poland') & (data['Price'] < 20)) | (((data['Quantity'] < 40) & (data['Discount'])))

print("1:\n", discounted_items, "\n")
print("2:\n", discounted_polish_items, "\n")
print("3:\n", milk_good_health_no_discount, "\n")
print("4:\n", ukraine_no_discount, "\n")
print("5:\n", large_discounted_batches, "\n")
print("6:\n", polish_large_volume_or_ukraine_discounted,
"\n")
print("7:\n", polish_low_price_or_small_discounted_volume,
"\n")
```

## 2.3 Output

```
1:
      Product Shelf_life Price CountryOfOrigin Quantity Discount
0      Dumpling "Ace" 2024-05-14      43      Ukraine      15      True
4      Sweet water 2024-08-02      28      Poland      25      True
5      Milk sausages 2024-06-30     171      Ukraine      11      True
6      Dumpling "Ace" 2024-09-01      43      Ukraine      12      True
7      Dumpling "Ace" 2024-08-15      42      Ukraine       9      True
9      Milk "Good health" 2024-05-14      29      Ukraine      48      True
10     Milk "Good health" 2024-07-28      30      Ukraine      55      True
11     Mineral water 2024-10-27      19      Poland      50      True
12     Mineral water 2024-10-27      19      Poland      15      True
13     Mineral water 2024-10-27      15      Poland      75      True
14     Mineral water 2024-07-26      25      Poland      20      True
16      Bread 2024-05-19      16      Ukraine      45      True
18     Mineral water 2024-03-09      15      Poland      45      True
19      Bread 2024-06-19      16      Ukraine      60      True
21      Bread 2024-06-14      16      Ukraine      55      True
```

```
2:
      Product Shelf_life Price CountryOfOrigin Quantity Discount
4      Sweet water 2024-08-02      28      Poland      25      True
11     Mineral water 2024-10-27      19      Poland      50      True
12     Mineral water 2024-10-27      19      Poland      15      True
13     Mineral water 2024-10-27      15      Poland      75      True
14     Mineral water 2024-07-26      25      Poland      20      True
18     Mineral water 2024-03-09      15      Poland      45      True
```

```
3:
Empty DataFrame
Columns: [Product, Shelf_life, Price, CountryOfOrigin, Quantity, Discount]
Index: []
```

```
4:
      Product Shelf_life Price CountryOfOrigin Quantity Discount
1      Milk "Good health" 2025-02-25      28      Ukraine     150      False
2      Milk sausages 2025-06-25     177      Ukraine      26      False
8      Milk "Good health" 2024-12-06      28      Ukraine      36      False
15     Coffe "Odessa" 2025-07-01      80      Ukraine      30      False
17     Coffe "Odessa" 2025-02-08      70      Ukraine       4      False
20     Coffe "Odessa" 2027-01-05      75      Ukraine       5      False
```

```
5:
      Product Shelf_life Price CountryOfOrigin Quantity Discount
4      Sweet water 2024-08-02      28      Poland      25      True
9      Milk "Good health" 2024-05-14      29      Ukraine      48      True
10     Milk "Good health" 2024-07-28      30      Ukraine      55      True
11     Mineral water 2024-10-27      19      Poland      50      True
```

13	Mineral water	2024-10-27	15	Poland	75	True
16	Bread	2024-05-19	16	Ukraine	45	True
18	Mineral water	2024-03-09	15	Poland	45	True
19	Bread	2024-06-19	16	Ukraine	60	True
21	Bread	2024-06-14	16	Ukraine	55	True

---

6:

	Product	Shelf_life	Price	CountryOfOrigin	Quantity	Discount
0	Dumpling "Ace"	2024-05-14	43	Ukraine	15	True
5	Milk sausages	2024-06-30	171	Ukraine	11	True
6	Dumpling "Ace"	2024-09-01	43	Ukraine	12	True
7	Dumpling "Ace"	2024-08-15	42	Ukraine	9	True
9	Milk "Good health"	2024-05-14	29	Ukraine	48	True
10	Milk "Good health"	2024-07-28	30	Ukraine	55	True
11	Mineral water	2024-10-27	19	Poland	50	True
13	Mineral water	2024-10-27	15	Poland	75	True
16	Bread	2024-05-19	16	Ukraine	45	True
18	Mineral water	2024-03-09	15	Poland	45	True
19	Bread	2024-06-19	16	Ukraine	60	True
21	Bread	2024-06-14	16	Ukraine	55	True

---

7:

	Product	Shelf_life	Price	CountryOfOrigin	Quantity	Discount
0	Dumpling "Ace"	2024-05-14	43	Ukraine	15	True
4	Sweet water	2024-08-02	28	Poland	25	True
5	Milk sausages	2024-06-30	171	Ukraine	11	True
6	Dumpling "Ace"	2024-09-01	43	Ukraine	12	True
7	Dumpling "Ace"	2024-08-15	42	Ukraine	9	True
11	Mineral water	2024-10-27	19	Poland	50	True
12	Mineral water	2024-10-27	19	Poland	15	True
13	Mineral water	2024-10-27	15	Poland	75	True
14	Mineral water	2024-07-26	25	Poland	20	True
18	Mineral water	2024-03-09	15	Poland	45	True

## Task 3

### 3.1 Condition

По даним з файлів students1.txt, students2.txt, students3.txt зробити висновок про лінійну залежність між успішністю студентів по наступним предметам, підрахувавши відповідні коефіцієнти кореляції оцінок по:

- 3 семестрам англійської мови (Іноземна мова1, Іноземна мова2, Іноземна мова3)  
(попарно для 3 предметів)
- другому семестру програмування на першому курсі і БЖД (Програмування2, Безпека життєдіяльності та охорона праці)

### 3.2 Solution code

```
import pandas as pd

df1 =
pd.read_csv('C:\\ALL\\OTHER\\GitHub\\HOMEWORK\\Python\\L4_v2_Herashchenko\\3_task\\students1.txt', sep='\\t')
```

```

df2 =
pd.read_csv('C:\\ALL\\OTHER\\GitHub\\HOMEWORK\\Python\\L4_V2_Herashchenko\\3_task\\stud
ents2.txt', sep='\\t')
df3 =
pd.read_csv('C:\\ALL\\OTHER\\GitHub\\HOMEWORK\\Python\\L4_V2_Herashchenko\\3_task\\stud
ents3.txt', sep='\\t')

df = pd.merge(df1, df2, on='Name', how='outer')
df = pd.merge(df, df3, on='Name', how='outer')

english_corr_1_2 = df['Іноземна мова1'].corr(df['Іноземна мова2'])
english_corr_1_3 = df['Іноземна мова1'].corr(df['Іноземна мова3'])
english_corr_2_3 = df['Іноземна мова2'].corr(df['Іноземна мова3'])

programming_bzhd_corr = df['Програмування2'].corr(df['Безпека життєдіяльності та
охорона праці'])

print(f"Іноземна мова1 and Іноземна мова2: {english_corr_1_2:.2f}")
print(f"Іноземна мова1 and Іноземна мова3: {english_corr_1_3:.2f}")
print(f"Іноземна мова2 and Іноземна мова3: {english_corr_2_3:.2f}")

print(f"Програмування2 and Безпека життєдіяльності та охорона праці:
{programming_bzhd_corr:.2f}")

```

### 3.3 Output

```

Іноземна мова1 and Іноземна мова2: 0.38
Іноземна мова1 and Іноземна мова3: 0.48
Іноземна мова2 and Іноземна мова3: 0.76
Програмування2 and Безпека життєдіяльності та охорона праці: 0.48

```

## Task 4

### 4.1 Condition

По даним з файлів 2008.csv.bz2 (або 2008\_rand.csv) і airports.csv виконати наступні завдання:

1. Знайти середню, мінімальну та максимальну відстань, що пройшов літак.
2. Чи не виглядає підозрілою максимальна пройдена відстань? У які дні і на яких рейсах вона була? Яка відстань була пройдена цими ж рейсами в інші дні?
3. Знайти день тижню, в якому проводилась максимальна кількість польотів.
4. Визначити 5 найбільш популярних місць відправлення (Origin) у грудні. В яких містах вони знаходяться?
5. Знайти аеропорт, в якому була найбільша затримка при посадці (ArrDelay), а також кількість хвилин запізнення. У який день, місяць і рік це було?

### 4.2 Solution code

```

import pandas as pd

flights_df =
pd.read_csv('C:\\ALL\\OTHER\\GitHub\\HOMEWORK\\Python\\L4_V2_Herashchenko\\4_task\\2008
_rand.csv')

```

```

airports_df =
pd.read_csv('C:\\ALL\\OTHER\\GitHub\\HOMEWORK\\Python\\L4_V2_Herashchenko\\4_task\\airpo
rts.csv')

mean_distance = flights_df['Distance'].mean()
min_distance = flights_df['Distance'].min()
max_distance = flights_df['Distance'].max()

print("1:")
print(f"Mean: {mean_distance:.2f}")
print(f"Min: {min_distance}")
print(f"Max: {max_distance}")

max_distance_flights = flights_df[flights_df['Distance'] == max_distance]
print("\n2.1:")
print(max_distance_flights[['Year', 'Month', 'DayofMonth', 'FlightNum', 'Origin',
'Dest', 'Distance']])

flight_nums = max_distance_flights['FlightNum'].unique()

other_days_same_flights = flights_df[(flights_df['FlightNum'].isin(flight_nums)) &
(flights_df['Distance'] != max_distance)]

print("\n2.2:")
print(other_days_same_flights[['Year', 'Month', 'DayofMonth', 'FlightNum', 'Origin',
'Dest', 'Distance']])

flights_per_day = flights_df['DayOfWeek'].value_counts().idxmax()
print(f"\n3: {flights_per_day}")

december_flights = flights_df[flights_df['Month'] == 12]
top_origins_december = december_flights['Origin'].value_counts().head(5).index

popular_origins_info =
airports_df[airports_df['iata'].isin(top_origins_december)][['iata', 'city']]
print("\n4:")
print(popular_origins_info)

max_arrival_delay = flights_df['ArrDelay'].max()
max_delay_flight = flights_df[flights_df['ArrDelay'] == max_arrival_delay]

print("\n5:")
print("\nFlight:")
print(max_delay_flight[['Year', 'Month', 'DayofMonth', 'Origin', 'Dest', 'ArrDelay']])

max_delay_airport = max_delay_flight.iloc[0]['Dest']
max_delay_airport_info = airports_df[airports_df['iata'] ==
max_delay_airport][['airport', 'city', 'state']]
print("\nAirport:")
print(max_delay_airport_info)

```

## 4.3 Output

```

1:
Mean: 724.70
Min: 31
Max: 4962

```

```

2.1:

```

	Year	Month	DayofMonth	FlightNum	Origin	Dest	Distance
11820	2008	6	4	14	HNL	EWR	4962
20914	2008	2	28	15	EWR	HNL	4962
27283	2008	9	8	15	EWR	HNL	4962

33322	2008	5	20	15	EWR	HNL	4962
41260	2008	11	10	14	HNL	EWR	4962
45993	2008	10	16	15	EWR	HNL	4962
89174	2008	6	20	15	EWR	HNL	4962
101591	2008	3	11	15	EWR	HNL	4962
110230	2008	5	27	15	EWR	HNL	4962
122945	2008	2	25	15	EWR	HNL	4962
134435	2008	8	4	14	HNL	EWR	4962
144236	2008	8	30	15	EWR	HNL	4962
146196	2008	4	11	15	EWR	HNL	4962
151132	2008	7	30	15	EWR	HNL	4962
154724	2008	11	8	14	HNL	EWR	4962
155195	2008	9	30	15	EWR	HNL	4962
159215	2008	2	17	14	HNL	EWR	4962
166332	2008	11	10	14	HNL	EWR	4962
191740	2008	9	27	15	EWR	HNL	4962
192450	2008	1	29	15	EWR	HNL	4962
195180	2008	6	16	14	HNL	EWR	4962
211733	2008	7	22	15	EWR	HNL	4962
226617	2008	2	13	14	HNL	EWR	4962
228024	2008	10	17	14	HNL	EWR	4962
256097	2008	2	10	14	HNL	EWR	4962
278874	2008	3	10	14	HNL	EWR	4962
295335	2008	2	2	14	HNL	EWR	4962
298581	2008	5	5	15	EWR	HNL	4962
304951	2008	10	17	15	EWR	HNL	4962
329496	2008	10	4	15	EWR	HNL	4962
329975	2008	8	29	14	HNL	EWR	4962
373287	2008	10	26	15	EWR	HNL	4962
391885	2008	10	21	14	HNL	EWR	4962
399590	2008	4	25	15	EWR	HNL	4962
400683	2008	11	2	14	HNL	EWR	4962
409387	2008	4	20	14	HNL	EWR	4962
409784	2008	10	10	15	EWR	HNL	4962
413657	2008	2	17	15	EWR	HNL	4962
416544	2008	11	16	14	HNL	EWR	4962
441534	2008	2	28	15	EWR	HNL	4962
443283	2008	1	13	14	HNL	EWR	4962
449483	2008	6	5	15	EWR	HNL	4962
450564	2008	12	29	15	EWR	HNL	4962
452656	2008	2	14	15	EWR	HNL	4962
453384	2008	12	22	15	EWR	HNL	4962
454222	2008	10	2	14	HNL	EWR	4962
456780	2008	5	17	15	EWR	HNL	4962
482455	2008	7	14	15	EWR	HNL	4962
484609	2008	7	9	14	HNL	EWR	4962

2.2:

	Year	Month	DayofMonth	FlightNum	Origin	Dest	Distance
5037	2008	5	23	14	HOU	DAL	239
5264	2008	3	1	15	BOS	SEA	2496
8051	2008	8	31	14	RSW	ATL	515
9003	2008	8	4	15	JFK	FLL	1069
9803	2008	5	17	15	JFK	SFO	2586
...	...	...	...	...	...	...	...
495614	2008	2	27	14	PHX	ATL	1587
497512	2008	6	2	14	SEA	EWR	2401
497856	2008	10	10	15	JFK	PHX	2153
498591	2008	5	17	15	JFK	PHX	2153
499284	2008	11	2	14	SEA	PHX	1107

[473 rows x 7 columns]



3: 3

4:

	iata	city
880	ATL	Atlanta
1263	DEN	Denver
1268	DFW	Dallas-Fort Worth
2039	LAX	Los Angeles
2531	ORD	Chicago

5:

Flight:

	Year	Month	DayofMonth	Origin	Dest	ArrDelay
238327	2008	12	17	LAS	DTW	1395.0

Airport:

	airport	city	state
1305	Detroit Metropolitan-Wayne County	Detroit	MI