Module Big Data I

Contact details:

Jan Buć

jan_buc@epam.com

Duration: 1,5h.

Max score: 5 points for report.

Conditions for completion: attendance, source code and report from exercise being sent to Github.

Introductory materials:

https://pandas.pydata.org/pandas-docs/stable/user_guide/index.html#user-guide
https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.groupby.html
https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.MultiIndex.html
https://en.wikipedia.org/wiki/Extract,_transform,_load

Required environment:

- VS Code as recomended IDE
- Installed Jupyter plugin in VS Code
- Python 3.8
- Pip for Python 3.8

Goal of exercise

Walkthrough some basic Pandas operations. All code is avaliable and ready to use. There are two common parts with tasks for which code will be written during classes.

Report

Report_part1.ipynb needs to be edited during class. It contains all needed information to complete the laboratory and needs to be sent back as a report.

For report 5 points will be distributed as:

- 1 points for report send to github after classes with common part covered (considered as attendance)

- 3 point for done homework (1 point for each subtask)
- 1 point for teoretical introduction for followed topics (up to 4 sentences will be sufficient)
  - What is pandas and it is for
  - Vectorized operations

Environment preparation

Clone your individual repository with exercise

> git clone <repo_name>

Create virtual environment in VS Code terminal

> python3 -m venv venv
Install needed dependencies

> pip install ipykernel
> pip install pandas

Introduction

In [ ]:
```
import pandas as pd
```

Data loading and basic dataframe information

In [ ]:
```
data = pd.read_csv("sales_records.csv")
```

In [ ]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 14 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Region          1000 non-null   object
 1   Country         1000 non-null   object
 2   Item Type       1000 non-null   object
 3   Sales Channel   1000 non-null   object
 4   Order Priority  1000 non-null   object
 5   Order Date      1000 non-null   object
 6   Order ID        1000 non-null   int64
 7   Ship Date       1000 non-null   object
 8   Units Sold      1000 non-null   int64
 9   Unit Price      1000 non-null   float64
 10  Unit Cost       1000 non-null   float64
 11  Total Revenue   1000 non-null   float64
 12  Total Cost      1000 non-null   float64
 13  Total Profit    1000 non-null   float64
dtypes: float64(5), int64(2), object(7)
memory usage: 109.5+ KB
```

In [ ]:
```
data.shape
```

Out[ ]:
```
(1000, 14)
```

In [ ]:
```
data.head()
```

Out[ ]:

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold	Unit Price
0	Middle East and North Africa	Libya	Cosmetics	Offline	M	10/18/2014	686800706	10/31/2014	8446	437.20
1	North America	Canada	Vegetables	Online	M	11/7/2011	185941302	12/8/2011	3018	154.06
2	Middle East and North Africa	Libya	Baby Food	Offline	C	10/31/2016	246222341	12/9/2016	1517	255.28
3	Asia	Japan	Cereal	Offline	C	4/10/2010	161442649	5/12/2010	3322	205.70
4	Sub-Saharan Africa	Chad	Fruits	Offline	H	8/16/2011	645713555	8/31/2011	9845	9.33

In [ ]:
```
type(data.Country), type(data), type(data.Country.values)
```

Out[ ]:
```
(pandas.core.series.Series, pandas.core.frame.DataFrame, numpy.ndarray)
```

In [ ]:
```
data.iloc[:,1].head(), data.Country.head(), data['Country'].head()
```

Out[ ]:
```
(0     Libya
 1    Canada
 2     Libya
 3     Japan
 4      Chad
 Name: Country, dtype: object,
 0     Libya
 1    Canada
 2     Libya
 3     Japan
 4      Chad
 Name: Country, dtype: object,
 0     Libya
 1    Canada
 2     Libya
 3     Japan
 4      Chad
 Name: Country, dtype: object)
```

In [ ]:
```
data['Item Type'].value_counts()
```

Out[ ]:
```
Beverages          101
Vegetables          97
Office Supplies     89
Baby Food           87
Personal Care       87
Snacks              82
Cereal              79
Clothes             78
Meat                78
Household           77
Cosmetics           75
Fruits              70
Name: Item Type, dtype: int64
```

Boolean indexing, masks

AND &

OR |

NOT ~

In [ ]:
```
m_mask = data['Order Priority'] == 'M'
```

In [ ]:
```
m_mask.head()
```

Out[ ]:
```
0     True
1     True
2    False
3    False
4    False
Name: Order Priority, dtype: bool
```

In [ ]:
```
m_filtered_data = data[m_mask]
```

In [ ]:
```
m_filtered_data.head()
```

Out[ ]:

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold	
0	Middle East and North Africa	Libya	Cosmetics	Offline	M	10/18/2014	686800706	10/31/2014	8446	4
1	North America	Canada	Vegetables	Online	M	11/7/2011	185941302	12/8/2011	3018	1
7	Europe	Montenegro	Clothes	Offline	M	5/17/2012	208630645	6/28/2012	7299	1
10	Sub-Saharan Africa	Togo	Clothes	Online	M	12/29/2015	451010930	1/19/2016	3012	1
11	Europe	Montenegro	Snacks	Offline	M	2/27/2010	220003211	3/18/2010	2694	1

In [ ]:
```
snack_mask = data['Item Type'] == 'Snacks'
m_snack_mask = m_mask & snack_mask
m_snack_filtered_data = data[m_snack_mask]
```

In [ ]:
```
m_snack_filtered_data.head()
```

Out[ ]:

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold	Unit Price
11	Europe	Montenegro	Snacks	Offline	M	2/27/2010	220003211	3/18/2010	2694	152.58
55	Asia	Malaysia	Snacks	Offline	M	10/6/2012	175033080	11/5/2012	5033	152.58
84	Sub-Saharan Africa	Rwanda	Snacks	Online	M	3/6/2017	866792809	3/18/2017	2109	152.58
120	Europe	Sweden	Snacks	Online	M	4/12/2011	742443025	4/15/2011	4245	152.58
121	Sub-Saharan Africa	Gabon	Snacks	Offline	M	10/3/2010	164569461	10/5/2010	8615	152.58

String columns handling

Series.str.contains()

Series.str.startswith()

Series.str.isnumeric()

In [ ]:
```
africa_mask = data['Region'].str.contains('Africa')
africa_data = data[africa_mask]
```

In [ ]:
```
africa_data.head()
```

Out[ ]:

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold	Unit Price
0	Middle East and North Africa	Libya	Cosmetics	Offline	M	10/18/2014	686800706	10/31/2014	8446	437.20
2	Middle East and North Africa	Libya	Baby Food	Offline	C	10/31/2016	246222341	12/9/2016	1517	255.28
4	Sub-Saharan Africa	Chad	Fruits	Offline	H	8/16/2011	645713555	8/31/2011	9845	9.33
6	Sub-Saharan Africa	Eritrea	Cereal	Online	H	3/4/2015	679414975	4/17/2015	2844	205.70
10	Sub-Saharan Africa	Togo	Clothes	Online	M	12/29/2015	451010930	1/19/2016	3012	109.28

Some of basic data conversion

In [ ]:
```
data[['Ship Date', 'Order Date']].info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   Ship Date   1000 non-null   object
 1   Order Date  1000 non-null   object
dtypes: object(2)
memory usage: 15.8+ KB
```

In [ ]:
```
data['Ship Date']= pd.to_datetime(data['Ship Date'])
data['Order Date']= pd.to_datetime(data['Order Date'])
```

In [ ]:
```
data[['Ship Date', 'Order Date']].info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype         
---  ------      --------------  -----         
 0   Ship Date   1000 non-null   datetime64[ns]
 1   Order Date  1000 non-null   datetime64[ns]
dtypes: datetime64[ns](2)
memory usage: 15.8 KB
```

Common part I

Show list of 3 countries with the lowest units sold:

- from Africa
- for selling fruits
- with C priority
- between 2013 and 2018 order date

In [ ]:
```
data = pd.read_csv("sales_records.csv")
data['Ship Date']= pd.to_datetime(data['Ship Date'])
data['Order Date']= pd.to_datetime(data['Order Date'])
data.head()
```

Homework part I

For given data perform following operations by filtering the data (for 1 point each).

- choose one Item Type (choose one by yourself, get results as Series of bolean values) - total cost is higher than total profit (get results as Series of bolean values) - merge two earlier conditions and show top 3 countries with highest units sold as list

In [ ]:
```
data = pd.read_csv("sales_records.csv")
data['Ship Date']= pd.to_datetime(data['Ship Date'])
data['Order Date']= pd.to_datetime(data['Order Date'])
data.head()
```

Out[ ]:

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold	Unit Price	Unit Cost
0	Middle East and North Africa	Libya	Cosmetics	Offline	M	2014-10-18	686800706	2014-10-31	8446	437.20	263.33
1	North America	Canada	Vegetables	Online	M	2011-11-07	185941302	2011-12-08	3018	154.06	90.93
2	Middle East and North Africa	Libya	Baby Food	Offline	C	2016-10-31	246222341	2016-12-09	1517	255.28	159.42
3	Asia	Japan	Cereal	Offline	C	2010-04-10	161442649	2010-05-12	3322	205.70	117.11
4	Sub-Saharan Africa	Chad	Fruits	Offline	H	2011-08-16	645713555	2011-08-31	9845	9.33	6.92

In [ ]:
```
# write your code here
```