## MSc Data Mining

Topic 01: Module Overview

Part 06: Top X pandas commands

### Dr Bernard Butler and Dr Kieran Murphy

Department of Computing and Mathematics, WIT. (bbutler@tssg.org; kmurphy@wit.ie)

Spring Semester, 2022

#### Outline

- Reading data formats
- Computing descriptive statistics
- Processing data by filtering and grouping

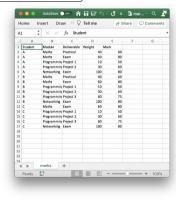
## Part I

Introduction

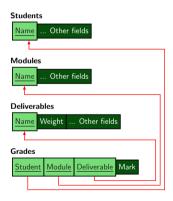
### Minimal Dataset

To better understand the various pandas operations we are going use a tiny\* dataset based on (fictional) student results. (marks.csv)





#### ... or database schema ...



### ...like to know ...

- Student performance weighted mark on each module, missing deliverables etc.
- Module performance number of attempts and average mark.
- Deliverable performance number of attempts and average mark, predictor of overall module grade, etc.

<sup>\*</sup>Dataset is small enough that you can verify operation results by hand.

## Terminology

df.head(1000)						
	Student	Module	Deliverable	Weight	Mark	
0	Α	Maths	Practical	40	60	
1	Α	Maths	Exam	60	80	
2	Α	Programming	Project 1	10	50	
3	Α	Programming	Project 2	30	60	
4	Α	Networking	Lab Work	100	80	
5	В	Maths	Practical	40	60	
6	В	Maths	Exam	60	80	
7	В	Programming	Project 1	10	50	
8	В	Programming	Project 2	30	60	
9	В	Programming	Project 3	60	75	
10	В	Networking	Project	100	80	
11	С	Maths	Exam	60	80	
12	С	Programming	Project 1	10	50	
13	С	Programming	Project 2	30	60	
14	С	Programming	Project 3	60	75	
15	С	Networking	Lab Work	100	80	

- A DataFrame is a table of data values.
  - df = pd.read\_csv("marks.csv")
- A Series is a list of data values typically columns in a dataframe. We can access an individual column using
  - df.Deliverable (dot notation
  - df["Deliverable"] (dict notation
  - df.iloc[:,2] (numpy, index notation)
- The index is a special column whose values can be used to access rows — rather using row number.
  - The default index is equal to the row number.

## Terminology

df.head(1000)						
	Student	Module	Deliverable	Veight	Mark	
0	Α	Maths	Practical	40	60	
1	Α	Maths	Exam	60	80	
2	Α	Programming	Project 1	10	50	
3	Α	Programming	Project 2	30	60	
4	Α	Networking	Lab Work	100	80	
5	В	Maths	Practical	40	60	
6	В	Maths	Exam	60	80	
7	В	Programming	Project 1	10	50	
8	В	Programming	Project 2	30	60	
9	В	Programming	Project 3	60	75	
10	В	Networking	Project	100	80	
11	С	Maths	Exam	60	80	
12	C	Programming	Project 1	10	50	
13	С	Programming	Project 2	30	60	
14	С	Programming	Project 3	60	75	
15	С	Networking	Lab Work	100	80	
			1			

• A DataFrame is a table of data values.

• df = pd.read\_csv("marks.csv")

 A Series is a list of data values — typically columns in a dataframe. We can access an individual column using

```
• df.Deliverable (dot notation)
```

- df["Deliverable"] (dict notation)
- df.iloc[:,2] (numpy, index notation)
- The index is a special column whose values can be used to access rows — rather using row number.
  - The default index is equal to the row number.

## Terminology

df.head(1000)							
	Student	Module	Deliverable	Weight	Mark		
0	Α	Maths	Practical	40	60		
1	А	Maths	Exam	60	80		
2	Α	Programming	Project 1	10	50		
3	А	Programming	Project 2	30	60		
4	А	Networking	Lab Work	100	80		
5	В	Maths	Practical	40	60		
6	В	Maths	Exam	60	80		
7	В	Programming	Project 1	10	50		
8	В	Programming	Project 2	30	60		
9	В	Programming	Project 3	60	75		
10	В	Networking	Project	100	80		
11	С	Maths	Exam	60	80		
12	С	Programming	Project 1	10	50		
13	С	Programming	Project 2	30	60		
14	С	Programming	Project 3	60	75		
15	С	Networking	Lab Work	100	80		

- A DataFrame is a table of data values.
  - df = pd.read\_csv("marks.csv")
- A Series is a list of data values typically columns in a dataframe. We can access an individual column using

```
• df.Deliverable (dot notation)
```

- df["Deliverable"] (dict notation)
- df.iloc[:,2] (numpy, index notation)
- The index is a special column whose values can be used to access rows — rather using row number.
  - The default index is equal to the row number.

## Part II

# Input and Output

### Setup

#### Minimal

We begin every data mining project with importing the three core data science packages:

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
plt.style.use('seaborn-darkgrid')

numpy — fast array operations pandas — data manipulation matplotlib — visualisation

• We give modules nicknames (np, pd, ...) to simplify their later use, and we access properties/functions of a package using the dot notation (np.max, pd.DataFrame, ...).

### Extra

import seaborn as sns import statsmodels.api as sm pd.set\_option('display.max\_columns', 500)

seaborn — statistical visualisation statsmodels — statistical data exploration pandas options to show all columns for wider datasets

### Setup

#### Minimal

We begin every data mining project with importing the three core data science packages:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
plt.style.use('seaborn-darkgrid')
```

```
numpy — fast array operations
pandas — data manipulation
matplotlib — visualisation
```

• We give modules nicknames (np, pd, ...) to simplify their later use, and we access properties/functions of a package using the dot notation (np.max, pd.DataFrame, ...).

#### Extra

```
import seaborn as sns
import statsmodels.api as sm

pd.set_option('display.max_columns', 500)
pd.set_option('display.width', 1000)
```

seaborn — statistical visualisation statsmodels — statistical data exploration pandas options to show all columns for wider datasets

## Reading data from a CSV file

Pandas supports a huge variety of input/output formats so best approach is to focus on what is needed to process the given data and verify input. Our marks dataset is in CSV format so we start with

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
plt.style.use('seaborn-darkgrid')
```

#### and input using

```
df = pd.read_csv('marks.csv', sep=',')
print(df.shape)
df.head()
```

## Reading data from a CSV file

Pandas supports a huge variety of input/output formats so best approach is to focus on what is needed to process the given data and verify input. Our marks dataset is in CSV format so we start with

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
plt.style.use('seaborn-darkgrid')
```

and input using

```
df = pd.read_csv('marks.csv', sep=',')
print(df.shape)
df.head()
```

### Reading data from a CSV file

Pandas supports a huge variety of input/output formats so best approach is to focus on what is needed to process the given data and verify input. Our marks dataset is in CSV format so we start with

```
import numby as no
 import pandas as pd
 import matplotlib.pyplot as plt
 plt.style.use('seaborn-darkgrid')
                                                                           (16.5)
and input using
                                                                                         Practica
                                                                                    Maths
                                                                                           Exam
 df = pd.read_csv('marks.csv', sep=',')
                                                                                         Project 1
 print(df.shape)
 df.head()
                                            (16, 5)
```

Always verify input by checking dataset dimensions and looking at some rows!!!

### **Datatypes**

#### Pandas data types:

- object used for text or mixed numeric and non-numeric values.
- int64 integer values,

• Does not support missing values, so an int column containing at least one missing value will automatically be converted to float.

- float64 floating point numbers.
- bool True/False values
- datetime64 date and time values
- category Finite (typically small) list of text values

Student object
Module object
Deliverable object
Weight int64
Mark int64
dtype: object

df.dtypes

### Regularly verifying datatypes is vital<sup>†</sup>:

- Operations differ based on datatype, eg, '+' concatenate strings but adds numerical values.
- Datatype can change based on results, eg, int converts to float due to missing values.

<sup>&</sup>lt;sup>†</sup>Google "Detecting Excel's gene auto-conversions."

### **Datatypes**

#### Pandas data types:

- object used for text or mixed numeric and non-numeric values.
- int64 integer values,
  - Does not support missing values, so an int column containing at least one missing value will automatically be converted to float.
- float64 floating point numbers.
- bool True/False values
- datetime64 date and time values
- category Finite (typically small) list of text values

Student object
Module object
Deliverable object
Weight int64
Mark int64
dtype: object

df.dtypes

### Regularly verifying datatypes is vital<sup>†</sup>

- Operations differ based on datatype, eg, '+' concatenate strings but adds numerical values.
- Datatype can change based on results, eg, int converts to float due to missing values.

<sup>&</sup>lt;sup>†</sup>Google "Detecting Excel's gene auto-conversions."

### Datatypes

#### Pandas data types:

- object used for text or mixed numeric and non-numeric values.
- int64 integer values,

 Does not support missing values, so an int column containing at least one missing value will automatically be converted to float.

- float64 floating point numbers.
- bool True/False values
- datetime64 date and time values
- category Finite (typically small) list of text values

Student object Module object Deliverable obiect Weight int64 Mark int64 dtype: object

df.dtypes

### Regularly verifying datatypes is vital<sup>†</sup>:

- Operations differ based on datatype, eg, '+' concatenate strings but adds numerical values.
- Datatype can change based on results, eg, int converts to float due to missing values.

<sup>&</sup>lt;sup>†</sup>Google "Detecting Excel's gene auto-conversions."

### Datatypes — Converting

We will deal with modifying and creating new columns later, but while we are on datatypes, we will look at changing datatype ...

#### Using the Series function astype

```
df["Weight"] = df["Weight"].astype('float')
df["Weight"].dtype
dtype('float64')
```

- New datatype is required argument 'int', 'float', 'str', 'object', 'category', etc.
- Simple, but fragile if data conversion is possible.

### or using pandas function to\_numeric

```
df["Weight"] = pd.to_numeric(df["Weight"])
df["Weight"].dtype
dtype('float64')
```

- More powerful, can specify what to do in cases where the conversion fails etc
- Have functions to\_numeric, to\_datetime, and to\_timedelta.

### Datatypes — Converting

We will deal with modifying and creating new columns later, but while we are on datatypes, we will look at changing datatype ...

Using the Series function astype

```
df["Weight"] = df["Weight"].astype('float')
df["Weight"].dtype
dtype('float64')
```

- New datatype is required argument 'int', 'float', 'str', 'object', 'category', etc.
- Simple, but fragile if data conversion is possible.

```
or using pandas function to_numeric
```

```
df["Weight"] = pd.to_numeric(df["Weight"])
df["Weight"].dtype
dtype('float64')
```

- More powerful, can specify what to do in cases where the conversion fails etc
- Have functions to\_numeric, to\_datetime, and to\_timedelta.

### Datatypes — Converting

We will deal with modifying and creating new columns later, but while we are on datatypes, we will look at changing datatype ...

#### Using the Series function astype

```
df["Weight"] = df["Weight"].astype('float')
df["Weight"].dtype
dtype('float64')
```

- New datatype is required argument 'int', 'float', 'str', 'object', 'category', etc.
- Simple, but fragile if data conversion is possible.

```
or using pandas function to_numeric
```

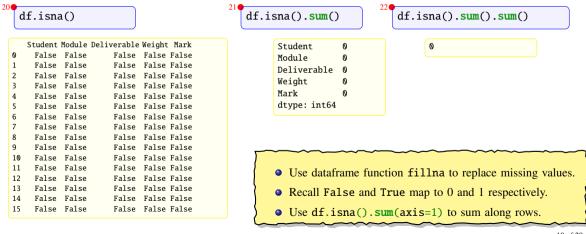
```
df["Weight"] = pd.to_numeric(df["Weight"])
df["Weight"].dtype
dtype('float64')
```

- More powerful, can specify what to do in cases where the conversion fails etc
- Have functions to\_numeric, to\_datetime, and to\_timedelta.

## Missing Values

Identifying and dealing with missing values is critical step in data preparation. What should you do? delete rows containing missing values? or impute then?

Here we will just look at identifying missing values.



### Output

Saving dataframe to CSV is straightforward (I rarely include the (default) index when saving datasets).

```
df.to_csv('marks_2.csv', index=False)
```

- CSV has become the default file format in Data Mining application especially for 'informal' datasets.
  - ✓ human readable, easy to generate / parse (if correct).
  - X Can be highly redundant, slow to input/output.
  - X No meta information.
- Other formats are better for speed and resulting file size and for saving meta data not supported by CSV (such as columns datatypes, category information, etc).

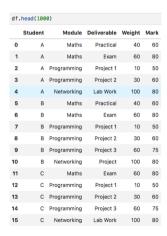
towards data science: The Best Format to Save Pandas Data

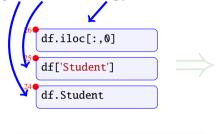
# Part III

Filtering

### Selecting individual rows/columns results in a series

Columns can accessed using dot, dict and numpy index notation.



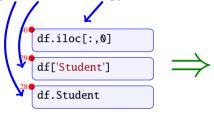


Student	A
Module	Maths
Deliverable	Practical
Weight	40
Mark	60
Name: 0, dty	

### Selecting individual rows/columns results in a series

Columns can accessed using dot, dict and numpy index notation.





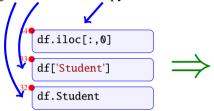
Student	A
Module	Maths
Deliverable	Practical
Weight	40
Mark	60
Name: 0, dty	

```
Α
      Α
      Α
      Α
      Α
      В
      В
      В
      В
      В
10
      В
11
12
13
14
15
Name: Student, dtype: object
```

### Selecting individual rows/columns results in a series

Columns can accessed using dot, dict and numpy index notation.





Student A
Module Maths
Deliverable Practical
Weight 40
Mark 60
Name: 0, dtype: object

```
Α
      Α
      Α
      Α
      В
      В
      В
      R
10
      R
11
12
13
14
15
Name: Student, dtype: object
```

### Head and Tail

Commands head and tail return the first and last n rows (default n = 5) of a dataframe/series.

	Student	Module	Deliverable	Weight	Mark
0	A	Maths	Practical	40	60
1	A	Maths	Exam	60	80
2	A	Programming	Project 1	10	50
3	A	Programming	Project 2	30	60
4	A	Networking	Lab Work	100	80
5	В	Maths	Practical	40	60
6	В	Maths	Exam	60	80
7	В	Programming	Project 1	10	50
8	В	Programming	Project 2	30	60
9	В	Programming	Project 3	60	75
10	В	Networking	Project	100	80
11	С	Maths	Exam	60	80
12	С	Programming	Project 1	10	50
13	С	Programming	Project 2	30	60
14	C	Programming	Project 3	60	75
15	C	Networking	Lab Work	100	80





### Head and Tail

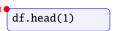
Commands head and tail return the first and last n rows (default n = 5) of a dataframe/series.

	Student	Module	Deliverable	Weight	Mark
0	A	Maths	Practical	40	60
1	A	Maths	Exam	60	80
2	A	Programming	Project 1	10	50
3	A	Programming	Project 2	30	60
4	A	Networking	Lab Work	100	80
5	В	Maths	Practical	40	60
6	В	Maths	Exam	60	80
7	В	Programming	Project 1	10	50
8	В	Programming	Project 2	30	60
9	В	Programming	Project 3	60	75
10	В	Networking	Project	100	80
11	С	Maths	Exam	60	80
12	С	Programming	Project 1	10	50
13	С	Programming	Project 2	30	60
14	С	Programming	Project 3	60	75
15	C	Networking	Lab Work	100	80
_					



	Student	Module	Deliverable	Weight	Marl
0	A	Maths	Practical	40	60
1	A	Maths	Exam	60	80
2	A	Programming	Project 1	10	50
3	A	Programming	Project 2	30	60
4	A	Networking	Lab Work	100	80





	Student	Module	Deliverable	Weight	Mark
0	A	Maths	Practical	40	60



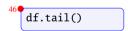
### Head and Tail

Commands head and tail return the first and last n rows (default n = 5) of a dataframe/series.

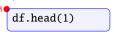
	Student	Module	Deliverable	Weight	Mark
0	A	Maths	Practical	40	60
1	A	Maths	Exam	60	80
2	A	Programming	Project 1	10	50
3	A	Programming	Project 2	30	60
4	A	Networking	Lab Work	100	80
5	В	Maths	Practical	40	60
6	В	Maths	Exam	60	80
7	В	Programming	Project 1	10	50
8	В	Programming	Project 2	30	60
9	В	Programming	Project 3	60	75
10	В	Networking	Project	100	80
11	C	Maths	Exam	60	80
12	С	Programming	Project 1	10	50
13	C	Programming	Project 2	30	60
14	С	Programming	Project 3	60	75
15	C	Networking	Lab Work	100	80



Student	Module	Deliverable	Weight	Mark
0 A	Maths	Practical	40	60
1 A	Maths	Exam	60	80
2 A	Programming	Project 1	10	50
3 A	Programming	Project 2	30	60
4 A	Networking	Lab Work	100	80



	Student	Module	Deliverable	Weight	Mark
11	С	Maths	Exam	60	80
12	С	Programming	Project 1	10	50
13	С	Programming	Project 2	30	60
14	C	Programming	Project 3	60	75
15	C	Networking	Lab Work	100	80

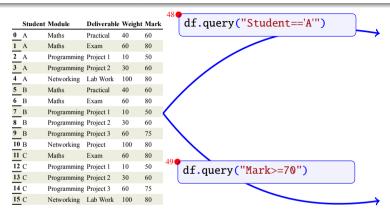


	Student	Module	Deliverable	Weight	Mark
0		Maths	Practical	40	60

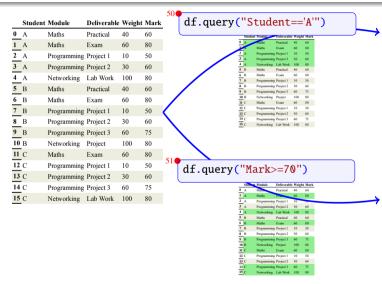


	Student	Module	Deliverable	Weight	Mark
15	C	Networking	Lab Work	100	80

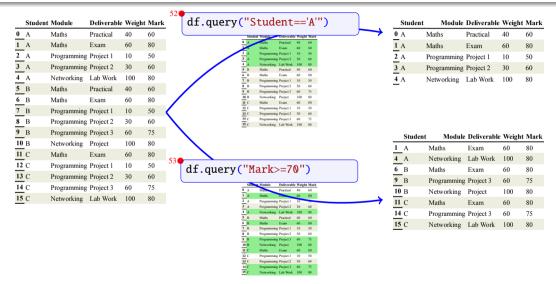
## Query — on a single-column criteria



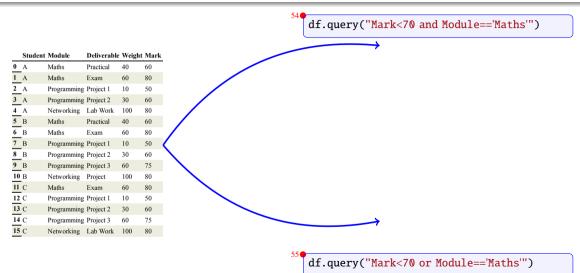
## Query — on a single-column criteria



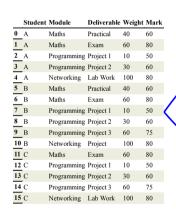
## Query — on a single-column criteria



# Query — on multiple columns (using python ogical operators)

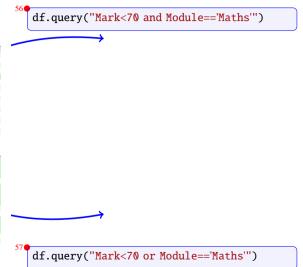


# Query — on multiple columns (using python ogical operators)





Networking Lab Work 100 80



# Query — on multiple columns (using python logical operators)

	Student	Module	Deliverable	Weight	Mark
0	A	Maths	Practical	40	60
1	A	Maths	Exam	60	80
2	A	Programming	Project 1	10	50
3	A	Programming	Project 2	30	60
4	A	Networking	Lab Work	100	80
5	В	Maths	Practical	40	60
6	В	Maths	Exam	60	80
7	В	Programming	Project 1	10	50
8	В	Programming	Project 2	30	60
9	В	Programming	Project 3	60	75
10	В	Networking	Project	100	80
11	C	Maths	Exam	60	80
12	С	Programming	Project 1	10	50
13	C	Programming	Project 2	30	60
14	С	Programming	Project 3	60	75
15	C	Networking	Lab Work	100	80

	Student			Weight	_
	A	Maths	Practical	40	60
	A	Maths	Exam	60	80
	A	Programming		10	50
	A	Programming		30	60
	A	Networking	Lab Work	100	80
5	В	Maths	Practical	40	60
	В	Maths	Exam	60	80
7	В	Programming	Project 1	10	50
	В	Programming	Project 2	30	60
9	В	Programming	Project 3	60	75
10	В	Networking	Project	100	80
11	C	Maths	Exam	60	80
12	C	Programming	Project I	10	50
13	C	Programming	Project 2	30	60
14	C	Programming	Project 3	60	75
15	С	Networking	Lab Work	100	80
	C Student		Lab Work  Deliverable		
0					
0	Student	Module	Deliverable	Weight	Mar
0 1 2	Student A	Module Maths	Deliverable Practical Exam	Weight 40	Mar 60
0 1 2	Student A A	Module Maths Maths	Deliverable Practical Exam Project I	Weight 40 60	Mar 60 80
0 1 2 3 4	Student A A A	Module Maths Maths Programming	Deliverable Practical Exam Project I	Weight 40 60 10	Mar 60 80 50
0 1 2 3 4	Student A A A A	Module Marks Marks Programming Programming	Deliverable Practical Exam Project 1 Project 2	Weight 40 60 10 30	Mar 60 80 50 60
0 1 2 3 4	Student A A A A A	Module Maths Maths Programming Programming Networking	Deliverable Practical Exam Project 1 Project 2 Lab Work	Weight 40 60 10 30	Mar 60 80 50 60
0 1 2 3 4 5	Student A A A A A A	Module Maths Maths Programming Programming Networking Maths	Deliverable Practical Exam Project 1 Project 2 Lab Work Practical Exam	Weight 40 60 10 30 100 40	Mar 60 80 50 60 80
0 1 2 3 4 5 6	Student A A A A A A B B	Module Maths Maths Programming Programming Networking Maths	Deliverable Practical Exam Project 1 Project 2 Lab Work Practical Exam Project 1	Weight 40 60 10 30 100 40 60	Mar 60 80 50 60 80
0 1 2 3 4 5 6 7	Student A A A A A A B B B	Module Maths Maths Programming Programming Networking Maths Maths	Deliverable Practical Exam Project 1 Project 2 Lab Work Practical Exam Project 1 Project 2	Weight 40 60 10 30 100 40 60 10	Mar 60 80 50 60 80 60 80 50
0 1 2 3 4 5 6 7	Student A A A A A B B B B B	Module Maths Maths Programming Programming Networking Maths Maths Programming Programming	Deliverable Practical Exam Project 1 Project 2 Lab Work Practical Exam Project 1 Project 2	Weight 40 60 10 30 100 40 60 10 30	Mar 60 80 50 60 80 60 80 60
0 1 2 3 4 5 6 7 8	Student A A A A A A B B B B B B B	Module Maths Maths Programming Programming Networking Maths Maths Programming Programming Programming Programming	Deliverable Practical Exam Project 1 Project 2 Lab Work Practical Exam Project 1 Project 2 Project 3	Weight 40 60 10 30 100 40 60 10 30 60	Mar 60 80 50 60 80 60 80 60 75
0 1 2 3 4 5 6 7 8	Student A A A A A A B B B B B B B B B	Module Maths Maths Programming Programming Networking Maths Programming Programming Programming Programming Programming Programming	Deliverable Practical Exam Project 1 Project 2 Lab Work Practical Exam Project 1 Project 2 Project 3 Project 3	Weight 40 60 10 30 100 40 60 10 30 60 100	Mar 60 80 50 60 80 60 80 50 60 75
0 1 2 3 4 5 6 7 8 9	Student A A A A A A B B B B B B C	Module Maths Maths Programming Networking Maths Maths Programming Programming Programming Maths	Deliverable Practical Exam Project 1 Project 2 Lab Work Practical Exam Project 1 Project 2 Project 3 Project 3 Project Exam Project 1	Weight 40 60 10 30 100 40 60 10 30 60 110 60	Mar 60 80 50 60 80 60 80 50 60 75 80 80
0 1 2 3 4 5 6 7 8 9 10 11 12 13	Student A A A A B B B B C C C C	Module Maths Maths Maths Programming Maths Maths Maths Maths Maths Maths Maths Programming	Deliverable Practical Exam Project 1 Project 2 Lab Work Practical Exam Project 1 Project 2 Project 3 Project 3 Project Exam Project 1	Weight 40 60 10 30 100 40 60 10 30 60 110 60 110	Mar 60 80 50 60 80 80 50 60 75 80 80 50
0 1 2 3 4 5 6 7 8 9	Student A A A A B B B B C C C C	Module Maths Maths Maths Programming Programming Maths Programming Maths Maths Maths Maths Maths Maths Programming	Deliverable Practical Exam Project 1 Project 2 Lab Work Practical Exam Project 1 Project 2 Project 3 Project 3 Project 1 Project 1 Project 1 Project 1 Project 1 Project 1	Weight 40 60 10 30 100 40 60 10 30 60 100 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 30 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10 60 10	Mai 60 80 50 60 80 60 80 60 75 80 80 60 60 60

8					
df.query("Mark	<70 a	nd Mo	dule=='	'Math	s'")
<del></del>	Student	Module 1	Deliverable	Weight	Mark
0	A	Maths 1	Practical	40	60
5	В	Maths 1	Practical	40	60
_	•				
_	Student	Modu	ule Deliverab	le Weight	Mark
0		Modu	Practical	le Weight	Mark 60
1	A A		_		
1 2	A A	Maths Maths	Practical	40	60
1 2	A A A	Maths Maths Programmi	Practical Exam	40 60	60 80
1	A A A	Maths Maths Programmi	Practical Exam ing Project 1	40 60 10	60 80 50
1 2 3	A A A A B	Maths Maths Programmi	Practical Exam ing Project 1 ing Project 2	40 60 10 30	60 80 50 60
1 2 3 5	A A A A B B	Maths Maths Programmi Programmi Maths Maths	Practical Exam ing Project 1 ing Project 2 Practical	40 60 10 30 40	60 80 50 60
1 2 3 5 6 7	A A A A B B	Maths Maths Programmi Programmi Maths Maths Programmi	Practical Exam Ing Project 1 Ing Project 2 Practical Exam	40 60 10 30 40 60	60 80 50 60 60 80
1 2 3 5 6 7 8	A A A B B B B	Maths Maths Programmi Programmi Maths Maths Programmi	Practical Exam  Ing Project 1 Ing Project 2 Practical Exam  Ing Project 1	40 60 10 30 40 60 10	60 80 50 60 60 80 50

df.query("Mark<70 or Module=='Maths'")

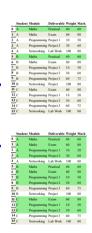
Programming Project 2

13 C

60

# Query — on multiple columns (using pandas ogical operators)

	Student	Module	Deliverable	Weight	Mark
0	A	Maths	Practical	40	60
1	A	Maths	Exam	60	80
2	A	Programming	Project 1	10	50
3	A	Programming	Project 2	30	60
4	A	Networking	Lab Work	100	80
5	В	Maths	Practical	40	60
6	В	Maths	Exam	60	80
7	В	Programming	Project 1	10	50
8	В	Programming	Project 2	30	60
9	В	Programming	Project 3	60	75
10	В	Networking	Project	100	80
11	С	Maths	Exam	60	80
12	C	Programming	Project 1	10	50
13	C	Programming	Project 2	30	60
14	С	Programming	Project 3	60	75
15	C	Networking	Lab Work	100	80



•					
df.query("(Ma	rk<70)	& (M	odule≕	='Mat	hs')
$\rightarrow$					
	Student	Module 1	Deliverable	Weight	Mark
	0 A	Maths	Practical	40	60
	5 B	Maths 1	Practical	40	60
	_				
	Student	Modu	ıle Deliverab	le Weight	Mark
	0 A	Maths	Practical	40	60
	1 A	Maths	Exam	60	80
	2 A	Programmi	ng Project 1	10	50
	3 A	Programmi	ng Project 2	30	60
	5 B	Maths	Practical	40	60
	6 B	Maths	Exam	60	80
	7 B	Programmi	ng Project 1	10	50
	8 B	Programmi	ng Project 2	30	60
$\longrightarrow$	11 C	Maths	Exam	60	80
	12 C	Programmi	ng Project 1	10	50
	13 C	Programmi	ng Project 2	30	60

df.query("(Mark<70) | (Module=='Maths')")

## Filtering using 10c

	Student	Module	Deliverable	Weight	Mar
0	A	Maths	Practical	40	60
1	A	Maths	Exam	60	80
2	A	Programming	Project 1	10	50
3	A	Programming	Project 2	30	60
4	A	Networking	Lab Work	100	80
5	В	Maths	Practical	40	60
6	В	Maths	Exam	60	80
7	В	Programming	Project 1	10	50
8	В	Programming	Project 2	30	60
9	В	Programming	Project 3	60	75
10	В	Networking	Project	100	80
11	С	Maths	Exam	60	80
12	С	Programming	Project 1	10	50
13	С	Programming	Project 2	30	60
14	С	Programming	Project 3	60	75
15	С	Networking	Lab Work	100	80

df.loc[ROW\_SELECTION, COL\_SELECTION]
where row and columns selection can be

- Single values: row number or column name
- An integer list for rows or list of column names
- A boolean list for logical indexing of rows
- A colon to indicate every row/column

df.loc[df.Module=="Maths", ["Student", "Mark"]]

## Filtering using 10c

S	tudent	Module	Deliverable	Weight	Mar
0 /	\	Maths	Practical	40	60
1 /	١.	Maths	Exam	60	80
2	١.	Programming	Project 1	10	50
3	١.	Programming	Project 2	30	60
4	١.	Networking	Lab Work	100	80
5 E	3	Maths	Practical	40	60
6 E	3	Maths	Exam	60	80
7 E	3	Programming	Project 1	10	50
8 E	3	Programming	Project 2	30	60
9 E	3	Programming	Project 3	60	75
10 E	3	Networking	Project	100	80
11 (		Maths	Exam	60	80
12 (		Programming	Project 1	10	50
13 (	2	Programming	Project 2	30	60
14 (	2	Programming	Project 3	60	75
15 (	2	Networking	Lab Work	100	80

df.loc[ROW\_SELECTION, COL\_SELECTION]
where row and columns selection can be

- Single values: row number or column name
- An integer list for rows or list of column names
- A boolean list for logical indexing of rows
- A colon to indicate every row/column

	Student	Module	Deliverable	Weight	Mark
۰	A	Maths	Practical	40	60
Ξ	Α	Maths	Exam	60	80
2	Α	Programming	Project I	10	50
3	Α	Programming	Project 2	30	60
4	Α	Networking	Lab Work	100	80
5	В	Maths	Practical	40	60
6	В	Maths	Exam	60	80
7	В	Programming	Project 1	10	50
8	В	Programming	Project 2	30	60
,	В	Programming	Project 3	60	75
10	В	Networking	Project	100	80
Ī	С	Maths	Exam	60	80
2	C	Programming	Project I	10	50
3	C	Programming	Project 2	30	60
4	C	Programming	Project 3	60	75
15	c	Materialism	I ale Winds	100	90

df.loc[df.Module=="Maths", ["Student", "Mark"]]

## Filtering using 10c

Note the square (not round) brackets — think of loc as array indexing not a function call.

	Student	Module	Deliverable	Weight	Mark
0	A	Maths	Practical	40	60
1	A	Maths	Exam	60	80
2	A	Programming	Project 1	10	50
3	A	Programming	Project 2	30	60
4	A	Networking	Lab Work	100	80
5	В	Maths	Practical	40	60
6	В	Maths	Exam	60	80
7	В	Programming	Project 1	10	50
8	В	Programming	Project 2	30	60
9	В	Programming	Project 3	60	75
10	В	Networking	Project	100	80
11	C	Maths	Exam	60	80
12	С	Programming	Project 1	10	50
13	C	Programming	Project 2	30	60
14	С	Programming	Project 3	60	75
15	C	Networking	Lab Work	100	80

df.loc[ROW\_SELECTION, COL\_SELECTION]
where row and columns selection can be

- Single values: row number or column name
- An integer list for rows or list of column names
- A boolean list for logical indexing of rows
- A colon to indicate every row/column



	Student	Mark
0	A	60
1	A	80
5	В	60
6	В	80
11	C	80

### More complicated example

I prefer to define row selection criteria, and the column list and order, separately to the loc statement.

	Student	Module	Deliverable	Weight	Mark
0	A	Maths	Practical	40	60
1	A	Maths	Exam	60	80
2	A	Programming	Project 1	10	50
3	A	Programming	Project 2	30	60
4	A	Networking	Lab Work	100	80
5	В	Maths	Practical	40	60
6	В	Maths	Exam	60	80
7	В	Programming	Project 1	10	50
8	В	Programming	Project 2	30	60
9	В	Programming	Project 3	60	75

```
criteria = ((df.Mark<50) & (df.Module=='Maths')) | ((df.Mark<70) & (df.Module!='Maths'))
columns = ['Module', 'Student', 'Mark']
df.loc[criteria, columns]</pre>
```

### More complicated example

I prefer to define define row selection criteria, and the column list and order, separately to the loc statement.

	Student	Module	Deliverable	Weight	Mark
0	A	Maths	Practical	40	60
1	A	Maths	Exam	60	80
2	A	Programming	Project 1	10	50
3	A	Programming	Project 2	30	60
4	A	Networking	Lab Work	100	80
5	В	Maths	Practical	40	60
6	В	Maths	Exam	60	80
7	В	Programming	Project 1	10	50
8	В	Programming	Project 2	30	60
9	В	Programming	Project 3	60	75

	Student	Medule	Deliverable	Weight	Mark
0	A	Maths	Practical	40	60
ī	A	Maths	Exam	60	80
2	A	Programming	Project I	10	50
3	A	Programming	Project 2	30	60
	A	Networking	Lab Work	100	80
5	В	Maths	Practical	40	60
6	В	Maths	Exam	60	80
7	В	Programming	Project 1	10	50
8	В	Programming	Project 2	30	60
9	В	Programming	Project 3	60	75
10	В	Networking	Project	100	80
11	C	Maths	Exam	60	80
12	C	Programming	Project I	10	50
13	C	Programming	Project 2	30	60
14	C	Programming	Project 3	60	75
15	C	Networking	Lab Work	100	80

```
criteria = ((df.Mark<50) & (df.Module=='Maths')) | ((df.Mark<70) & (df.Module!='Maths'))
columns = ['Module', 'Student', 'Mark']
df.loc[criteria, columns]</pre>
```

## More complicated example

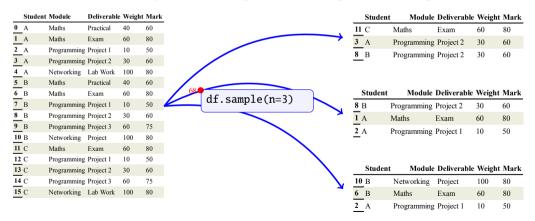
I prefer to define define row selection criteria, and the column list and order, separately to the loc statement.

Student Module   Deliverable   Weight Mark	A A A A 3 3	Maths Maths Programming Programming Networking Maths	Practical Exam Project 1 Project 2 Lab Work	40 60 10 30 100	60 80 50 60	<u> </u>	5 B 6 B 7 B 8 B	Program Network Maths Maths Program	rning Project 2 ing Lab Work Practical Exam rning Project 1	10 30 100 40 60 10	50 60 80 60 80 50		0 A 1 A 2 A	Maths Maths Programmin	Practical Exam g Project 1	40 60 10	60 80
A Maths	A A A A A A A A A A A A A A A A A A A	Maths Programming Programming Networking Maths	Exam Project 1 Project 2 Lab Work	60 10 30 100	80 50 60		5 B 6 B 7 B 8 B		,	100 40 60 10 30	80 60 80 50		1 A	Maths Programmin	Exam g Project 1	60 10	80
A	A A A A 3	Maths Programming Programming Networking Maths	Exam Project 1 Project 2 Lab Work	10 30 100	50 60		6 B 7 B 8 B		,	60 10 30	80 50		2 A	Programmin	g Project 1	10	
2 A Programming Project 1 10 50	۸ ۸ ۸ 3	Programming Programming Networking Maths	Project 1 Project 2 Lab Work	10 30 100	50 60		9 B		,	30	60		_				50
3 A   Programming Project 2   30   60	A A 3	Programming Networking Maths	Project 2 Lab Work	30 100	60		9 B	Program					3 A	Programmin	g Project 2		60
A Networking Lab Work   100   80   10   10   10   10   10   1	A 3 3	Networking Maths	Lab Work	100				Network	ing Project	60 100	75 80	7					
A   Networking Lab Work   100   80   110   80   110   80   110   80   110   80   110   80   110   80   110   80   110   80   110   80   110   80   110   80   8	3	Maths			80			Maths	Exam mine Project I	60	80 50		5 B	Maths	Practical	40	60
S   B   Maths   Practical   40   60	3		Practical				13 C		ning Project 2	30	60		6 B	Maths	Exam	60	80
Programming Project 1   10   50     Programming Project 3   60   75   10   B   Programming Project 3   60   75   10   B   Programming Project 2   30   60   11   C   Programming Project 3   60   75   13   C   Programming Project 3   13   C   Programming Project 2   30   60   13   C   Programming Project 2   30   60   13   C   Programming Project 2   30   60   13   C   Programming Project 3   60   75   Programming Project 3   60   75   Programming Project 3   60   75   Programming Project 3   75   Progra				40	60		14 C 15 C	Program Network		100	75 80		7 B	Programmin	g Project l	10	50
T B   Programming Project 1   10   50     10   8 B   Programming Project 2   30   60   11 C   Programming Project 3   60   80   11 C   Programming Project 3   60   80   12 C   Programming Project 3   10   50   Programming Project 3   10   Programming		Maths	Exam	60	80								_		g Project 2	30	60
8 B Programming Project 2 30 60 11 C Maths Exam 60 80 11 C Programming Project 3 60 75 11 C Programming Project 1 10 50 11 C Programming Project 2 30 60 75 11 C Programming Project 3 30 60 75 11 C Programming P													_		,		
9 B Programming Project 3 60 75 \frac{112 \cdot C}{13 \cdot C} \frac{Programming Project 1 10 50}{Programming Project 2 30 60} \frac{10 \cdot C}{13 \cdot C} \frac{Programming Project 2 10 50}{Programming Project 2 30 60} \frac{10 \cdot C}{10 \cdot C} \frac{10 \cdot C}{10 \cdo															Project	100	80
riteria = ((df.Mark<50) & (df.Module=='Maths'))   ((df.Mark<70) & (df.Module!='Maths')	3	Programming	Project 2	30	60								_				
riteria = ((df.Mark<50) & (df.Module=='Maths'))   ((df.Mark<70) & (df.Module!='Maths')	3	Programming	Project 3	60	75								_		,		
		.0 0	.,			_							13 C	Programmin	g Project 2	30	60
lf 1	3	mns	Programming eria = ((d: mns = ['Mod	Programming Project 3 eria = ((df.Mark mns = ['Module', '	Programming Project 3 60 eria = ((df.Mark<50) mns = ['Module', 'Stud	Programming Project 3 60 75 eria = ((df.Mark<50) &	Programming Project 3 60 75 eria = ((df.Mark<50) & (df.Marks = ['Module', 'Student', 'Ma	Programming Project 3 60 75  eria = ((df.Mark<50) & (df.Modul mns = ['Module', 'Student', 'Mark']	Programming Project 3 60 75  eria = ((df.Mark<50) & (df.Module=='mns = ['Module', 'Student', 'Mark']	Programming Project 3 60 75  eria = ((df.Mark<50) & (df.Module=='Mathment', 'Mark']  mns = ['Module', 'Student', 'Mark']	Programming Project 3 60 75  eria = ((df.Mark<50) & (df.Module=='Maths' mns = ['Module', 'Student', 'Mark']	Programming Project 3 60 75  eria = ((df.Mark<50) & (df.Module=='Maths')) mns = ['Module', 'Student', 'Mark']	Programming Project3 60 75  eria = ((df.Mark<50) & (df.Module=='Maths'))   ((df.Mark<70) & mns = ['Module', 'Student', 'Mark']	Programming Project 3 60 75  eria = ((df.Mark<50) & (df.Module=='Maths'))   ((df.Mark<70) & (mns = ['Module', 'Student', 'Mark']	Programming Project 3 60 75  Programming Project 4 75  Programming Project	Programming Project 3 60 75  Programming Project 1 13 C Programming Project 1 13 C Programming Project 2 Programming Project 3 C Programming Project 3 C Programming Project 3 C Programming Project 4 Programming Project 2 Programming Project 3 Programming Project 4	Programming Project 3 60 75  Programming Project 3 60 75  Programming Project 3 60 75  Programming Project 1 10 Programming Project 3 10 Programming Project 1 10 Programming Project 2 10 Programmi

## Sampling

The sample function selects a random subset of the dataframe rows.

- Either specify the number of rows (as an integer) or fraction of the data (as a float).
- Can set the seed using random\_state parameter for reproducible samples.



# Part IV

# Sorting

#### Sorting

#### A pandas dataframe has two sorting operations:

- sort\_index() orders rows based on current index.
- sort\_values(COLUMNS) orders rows based on single column or list of columns.

#### Two important modifications:

- By default, the sort order is in ascending. Set parameter ascending=False to reverse this.
- By default, a new dataframe is returned with desired sort order, set parameter inplace=True to update current dataframe instead (then no output is generated).

_	Student	Module	Deliverable	Weight	Mark
0	A	Maths	Practical	40	60
1	A	Maths	Exam	60	80
2	A	Programming	Project 69	10	50
3	A	Programming	Project 2	df.	sor
	A	Networking	Lab Work		
5	В	Maths	Practical	40	60
6	В	Maths	Exam	60	80
7	В	Programming	Project 1	10	50
8	В	Programming	Project 2	30	60
9	В	Programming	Project 3	60	75
	В	Networking	Project	100	80
11	С	Maths	Exam	60	80
12	С	Programming	Project 1	10	50
13	C	Drogramming	Project 2	20	60

### Sorting

#### A pandas dataframe has two sorting operations:

- sort\_index() orders rows based on current index.
- sort\_values(COLUMNS) orders rows based on single column or list of columns.

#### Two important modifications:

- By default, the sort order is in ascending. Set parameter ascending=False to reverse this.
- By default, a new dataframe is returned with desired sort order, set parameter inplace=True to update current dataframe instead (then no output is generated).

Student	Module	Deliverable	Weight	Mark			Student	Module	Deliverable	Weight	Mark
0 A	Maths	Practical	40	60		1	A	Maths	Exam	60	80
1 A	Maths	Exam	60	80		6	В	Maths	Exam	60	80
2 A	Programming	Project 70	10	50		11	С	Maths	Exam	60	80
3 A	Programming	Project 2	df.	sort	t_values(['Module','Deliverable'])		A	Maths	Practical	40	60
4 A	Networking	Lab Work				┚	В	Maths	Practical	40	60
5 B	Maths	Practical	40	60		4	A	Networking	Lab Work	100	80
6 B	Maths	Exam	60	80		15	С	Networking	Lab Work	100	80
7 B	Programming	Project 1	10	50		10	В	Networking	Project	100	80
8 B	Programming	Project 2	30	60		2	A	Programming	Project 1	10	50
9 B	Programming	Project 3	60	75		7	В	Programming	Project 1	10	50
10 B	Networking	Project	100	80		12	С	Programming	Project 1	10	50
11 C	Maths	Exam	60	80		3	A	Programming	Project 2	30	60
12 C	Programming	Project 1	10	50		8	В	Programming	Project 2	30	60
13 C	Drogramming	Project 2	20	60		13	C	Programming	Project 2	20	60

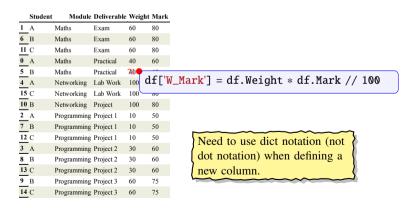
#### Part V

# **Defining New Columns**

## Defining new columns — row-wise operation

We want to compute the weighted mark for each module for each student. Two steps:

- Create column, W\_Mark, to store the weighted mark for each deliverable. This is a row by row calculation only need data in current row to compute the result.
- Create column, M\_Mark, to store the module mark for each student. This is a group calculation need all rows for that student and module to compute the result.



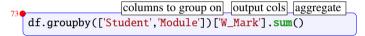
## Defining new columns — row-wise operation

We want to compute the weighted mark for each module for each student. Two steps:

- Create column, W\_Mark, to store the weighted mark for each deliverable. This is a row by row calculation only need data in current row to compute the result.
- Create column, M\_Mark, to store the module mark for each student. This is a group calculation need all rows for that student and module to compute the result.

Student	Module	Deliverable	Weigh	t Mark							Student	Module	Deliverable	Weight	Mark	W Mari
1 A	Maths	Exam	60	80	•				,	1		Maths				48
6 B	Maths	Exam	60	80					Y	6	В	Maths	Exam	60	80	48
11 C	Maths	Exam	60	80					•	11	С	Maths	Exam	60	80	48
0 A	Maths	Practical	40	60						0	A	Maths	Practical	40	60	24
5 B	Maths	Practical	720				16		16 1 //	Ξ		Maths	Practical	40	60	24
4 A	Networking	Lab Work	100 C	ıt ['V	V_Mai	rk'] =	= df.	Weight	* df.Mark //	10	90	letworking	Lab Work	100	80	80
15 C	Networking	Lab Work	100	ου		_				13	C	Networking	Lab Work	100	80	80
10 B	Networking	Project	100	80					<b>&gt;</b>	10	В	Networking	Project	100	80	80
2 A	Programming	Project 1	10	50						2	A	Programming	Project 1	10	50	5
7 B	Programming	Project 1	10	50						7	В	Programming	Project 1	10	50	5
12 C	Programming	Project 1	10	50		Need	d to u	se dict	notation (not	12	С	Programming	Project 1	10	50	5
3 A	Programming	Project 2	30	60	\ \frac{1}{2}	1			` )	3	A	Programming	Project 2	30	60	18
8 B	Programming	Project 2	30	60		dot i	notatio	on) wne	en defining a	8	В	Programming	Project 2	30	60	18
13 C	Programming	Project 2	30	60		new	colun	nn.		13	С	Programming	Project 2	30	60	18
9 B	Programming	Project 3	60	75	. '	$\sim$	~~			9	В	Programming	Project 3	60	75	45
14 C	Programming	Project 3	60	75						14	С	Programming	Project 3	60	75	45

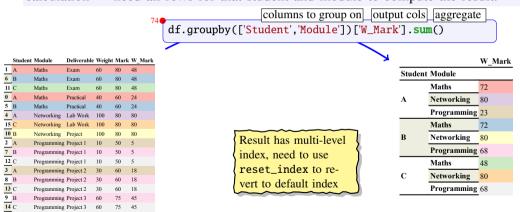
- Create column, W\_Mark, to store the weighted mark for each deliverable. This is a row by row calculation only need data in current row to compute the result.
- Create column, M\_Mark, to store the module mark for each student. This is a group calculation need all rows for that student and module to compute the result.



	Student	Module	Deliverable	Weight	Mark	W_Mark
1	A	Maths	Exam	60	80	48
6	В	Maths	Exam	60	80	48
11	С	Maths	Exam	60	80	48
0	A	Maths	Practical	40	60	24
5	В	Maths	Practical	40	60	24
4	A	Networking	Lab Work	100	80	80
15	С	Networking	Lab Work	100	80	80
10	В	Networking	Project	100	80	80
2	A	Programming	Project 1	10	50	5
7	В	Programming	Project 1	10	50	5
12	С	Programming	Project 1	10	50	5
3	A	Programming	Project 2	30	60	18
8	В	Programming	Project 2	30	60	18
13	C	Programming	Project 2	30	60	18
9	В	Programming	Project 3	60	75	45
14	C	Programming	Project 3	60	75	45

Result has multi-level index, need to use reset\_index to revert to default index

- Create column, W\_Mark, to store the weighted mark for each deliverable. This is a row by row calculation only need data in current row to compute the result.
- Create column, M\_Mark, to store the module mark for each student. This is a group calculation need all rows for that student and module to compute the result.



```
columns to group on output cols aggregate df.groupby(['Student','Module'])[['W_Mark']].sum().reset_index()
```

	Student	Module	Deliverable	Weight	Mark	W_Mark
1	A	Maths	Exam	60	80	48
6	В	Maths	Exam	60	80	48
11	C	Maths	Exam	60	80	48
0	A	Maths	Practical	40	60	24
5	В	Maths	Practical	40	60	24
4	A	Networking	Lab Work	100	80	80
15	C	Networking	Lab Work	100	80	80
10	В	Networking	Project	100	80	80
2	A	Programming	Project 1	10	50	5
7	В	Programming	Project 1	10	50	5
12	С	Programming	Project 1	10	50	5
3	A	Programming	Project 2	30	60	18
8	В	Programming	Project 2	30	60	18
13	C	Programming	Project 2	30	60	18
9	В	Programming	Project 3	60	75	45
14	C	Programming	Project 3	60	75	45

Programming Project 3

Programming Project 3

14 C

60 75

60 75 45

```
columns to group on
                                                            output cols
                                                                               aggregate
         df.groupby(['Student','Module'])[['W_Mark']].sum().reset_index()
  Student Module
                   Deliverable Weight Mark W Mark
                                                                                                    Student Module
                                                                                                                            W Mark
         Maths
                   Exam
                            60
                                  80
                                       48
                                                                                                 0 A
                                                                                                             Maths
                                                                                                                            72.
6 B
11 C
0 A
5 B
4 A
15 C
10 B
2 A
7 B
12 C
3 A
8 B
13 C
                                       48
         Maths
                   Exam
                            60
                                  80
                                                            This is the required result
                                                                                                  1 A
                                       48
                                                                                                             Networking
                                                                                                                            80
         Maths
                   Exam
                            60
                                  80
                                                            and we can save this to
         Maths
                   Practical
                            40
                                  60
                                       24
                                                                                                  2 A
                                                                                                             Programming 23
                                                            a new dataframe. How-
         Maths
                   Practical
                            40
                                  60
                                      24
                                                                                                 3 B
                                                                                                             Maths
                                                                                                                            72
         Networking Lab Work
                            100
                                  80
                                       80
                                                            ever, we often want to
         Networking Lab Work
                            100
                                  80
                                       80
                                                                                                  4 B
                                                                                                                            80
                                                            put this into to out origi-
                                                                                                             Networking
         Networking Project
                            100
                                  80
                                       80
                                                                                                 5 B
                                                            nal dataframe as an extra
                                                                                                             Programming 68
         Programming Project 1
                            10
                                  50
                                       5
         Programming Project 1
                            10
                                  50
                                       5
                                                            column. Only problem
                                                                                                 6 C
                                                                                                             Maths
                                                                                                                            48
         Programming Project 1
                            10
                                  50
                                       5
                                                            we have different rows so
                                                                                                 7 C
                                                                                                             Networking
                                                                                                                            80
         Programming Project 2
                            30
                                  60
                                       18
                                                            can't just assign to a new
         Programming Project 2
                                  60
                                       18
                                                                                                  8 C
                            30
                                                                                                             Programming 68
                                                            column — need to use
         Programming Project 2
                            30
                                  60
                                       18
```

transform function.

III

columns to group on output cols aggregate df['M\_Mark'] = df.groupby(['Student','Module'])[ ['W\_Mark'] ].transform(sum)

	Student	Module	Deliverable	Weight	Mark	W_Mark
1	A	Maths	Exam	60	80	48
6	В	Maths	Exam	60	80	48
11	C	Maths	Exam	60	80	48
0	A	Maths	Practical	40	60	24
5	В	Maths	Practical	40	60	24
4	A	Networking	Lab Work	100	80	80
15	C	Networking	Lab Work	100	80	80
10	В	Networking	Project	100	80	80
2	A	Programming	Project 1	10	50	5
7	В	Programming	Project 1	10	50	5
12	С	Programming	Project 1	10	50	5
3	A	Programming	Project 2	30	60	18
8	В	Programming	Project 2	30	60	18
13	C	Programming	Project 2	30	60	18
9	В	Programming	Project 3	60	75	45
14	C	Programming	Project 3	60	75	45

		<b>.</b> ,	
	Student	Module	W_Mark
0	A	Maths	72
1	A A	Networking	80
2	A	Programming	23
3	В	Maths	72
4	В	Networking	80
5	В	Programming	68
6	C	Maths	48
7	С	Networking	80
8	C	Programming	68

	Student	Module	Deliverable	Weight	Mark	W_Mark	M_Mar
1	A	Maths	Exam	60	80	48	72
6	В	Maths	Exam	60	80	48	72
11	C	Maths	Exam	60	80	48	48
0	A	Maths	Practical	40	60	24	72
5	В	Maths	Practical	40	60	24	72
4	A	Networking	Lab Work	100	80	80	80
15	C	Networking	Lab Work	100	80-	80	80
10	В	Networking	Project	100	80	80	80
2	K	Programming	Project 1	10	50	5	23
7	В	Programming	Project 1	10	50	5	68
12	C	Programming	Project 1	10	50	5	68
3	A	Programming	Project 2	30	60	18	23
8	В	Programming	Project 2	30	60	18	68
13	C	Programming	Project 2	30	60	18	68
9	В	Programming	Project 3	60	75	45	68
14	C	Programming	Project 3	60	75	45	68
	6 111 0 5 4 15 10 2 7 12 3 8 13	1 A B B B B B B B B B B B B B B B B B B	6 B Maths 11 C Maths 0 A Maths 8 B Maths 4 A Networking 15 C Networking 16 B Programming 7 B Programming 17 C Programming 18 B Programming 8 B Programming 18 B Programming 19 D Programming 10 Programming 11 C Programming 12 D Programming 13 C Programming 15 D Programming 16 D Programming 17 D Programming 18 B Programming 18 B Programming 19 D Programming	Maths Exam  Maths Exam  Maths Exam  A Maths Exam  A Maths Practical  A Networking Lab Work  Networking Lab Work  Networking Project 1  Programming Project 1  Programming Project 1  Programming Project 2  Programming Project 3	A	A Maths Exam 60 80	1 A Maths Exam 60 80 48     6 B Maths Exam 60 80 48     11 C Maths Exam 60 80 48     12 C Maths Exam 60 80 48     13 C Programming Project 2 30 60 18     14 A Maths Practical 40 60 24     15 C Networking Lab Work 100 80 80     16 B Networking Project 100 80 80     17 C Networking Project 1 10 50 5     18 B Programming Project 1 10 50 5     18 B Programming Project 2 30 60 18     18 B Programming Project 2 30 60 18     18 D Programming Project 3 30 60 75 45     18 D Programming Project 3 30 60 75 45     18 D Programming Project 3 60 75 45     18 D Project 3 60 75 45

The transform broadcasts the result for each group over every row in that group.

## Part VI

**Review Exercises** 

#### **Review Exercises**

#### Generate the following reports:

- Number of deliverables by each student.
- List and rank deliverables by grade.
- **1** Top 2 deliverables (by grade).
- Top 2 module (by average grade).
- **5** Top 2 modules (by minimum grade).
- Modules (by minimum grade).

#### Harder exercises (new functions)

List which students missed which deliverables.

(value\_counts, or groupby and count)

(sort\_values, rank)

(pivot, melt)