Data Analysis with Pandas

Olatz Perez de Vinaspre
Ander Soraluze
HAP/LAP.

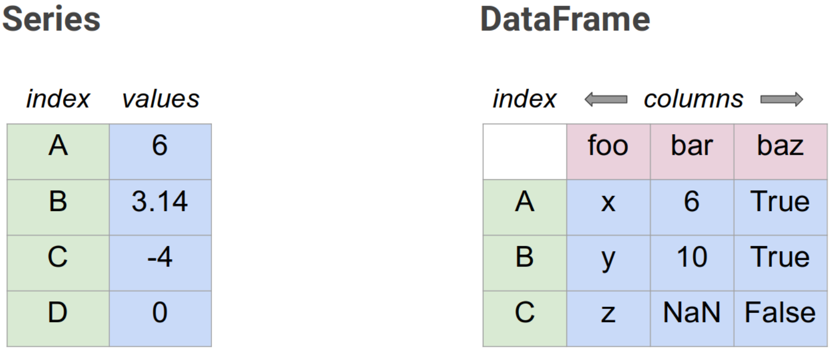
2021

What is Pandas?

1. Python package to deal with data analysis
2. It simplifies the loading of data from external resources
3. Save you a lot of effort from writing lower python code for analiysing and manipulating data
4. Main data structures – Series and DataFrame

Pandas Data Structures

- Series: an indexed 1D array
- DataFrame: Generalized two dimensional array with flexible row and column indices



Series

index	*values*
A	6
B	3.14
C	-4
D	0

DataFrame

index ⟵ *columns* ⟶

	foo	bar	baz
A	x	6	True
B	y	10	True
C	z	NaN	False

Creating Series

```
import pandas as pd
s1 = pd.Series([1, 2, 3, 4])
```

Splicit index

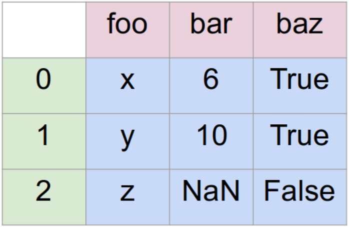
```
s2=pd.Series([1, 2, 3, 4],index=[‘A’, ‘B’, ‘C’, ‘D’])
```

0	1
1	2
2	3
3	4

A	1
B	2
C	3
D	4

Creating DataFrame

```
df = pd.DataFrame(‘foo’: [‘x’, ‘y’, ‘z’], ‘bar’: [6,
10, None], ‘baz’: [True, True, False])
```



	foo	bar	baz
0	x	6	True
1	y	10	True
2	z	NaN	False

Can Work as a Dictionary

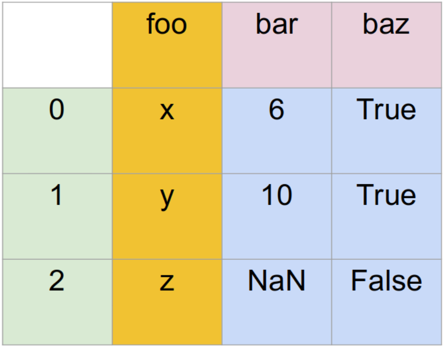
```
population_dict = {’California’ : 38332521, ’Texas’ :
26448193, ’New York’ : 19651127}

population=pd.Series(population_dict)

print(population)
California 38332521
Texas 26448193
New York 19651127
```

Knowing Your data

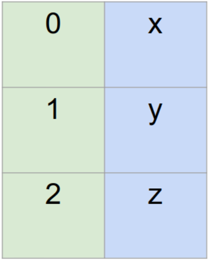
```
df.columns #Prints all the columns names
df.shape # Prints the number of cols and rows
df.shape[0] # Give you the number of rows
df.shape[1] #Give you the number of columns
df.info() #Info on DataFrame
```

Column Selection

```
df['foo']
```



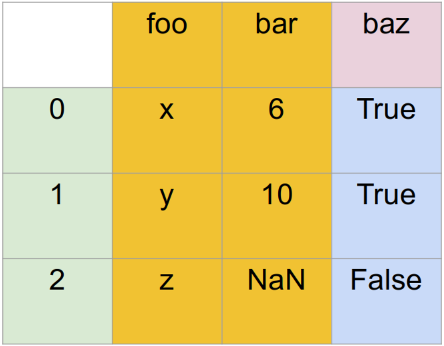
	foo	bar	baz
0	x	6	True
1	y	10	True
2	z	NaN	False



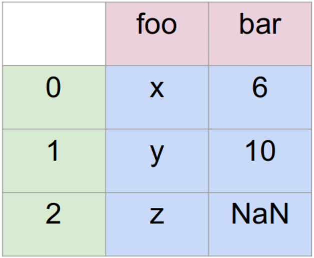
0	x
1	y
2	z

Column Selection

```
df[['foo', 'bar']]
```



	foo	bar	baz
0	x	6	True
1	y	10	True
2	z	NaN	False



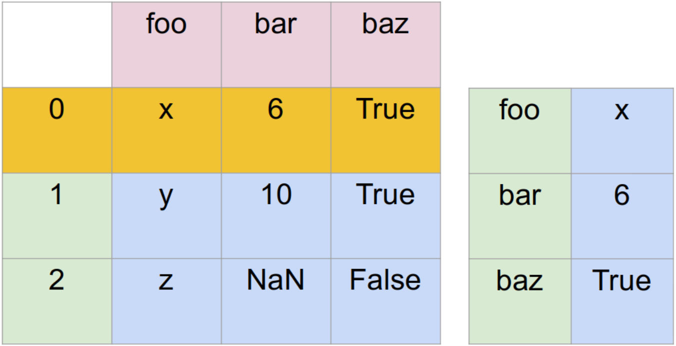
	foo	bar
0	x	6
1	y	10
2	z	NaN

Row Selection

```
df.head() #Returns the first 5 rows.
df.tail() #Returns the last 5 rows

df.head(n) #Returns the first n rows
df.tail(n) #Returns the last n rows
```

Row Selection

```
df.loc[0]
```

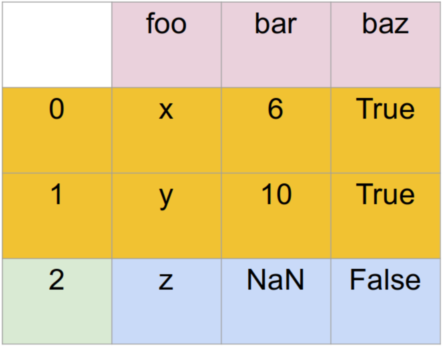


	foo	bar	baz
0	x	6	True
1	y	10	True
2	z	NaN	False

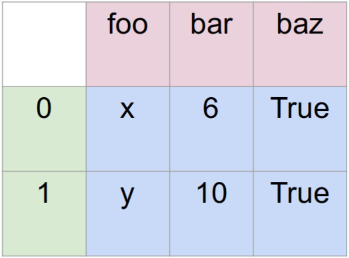
foo	x
bar	6
baz	True

Row Selection

```
df.loc[0:2]
```



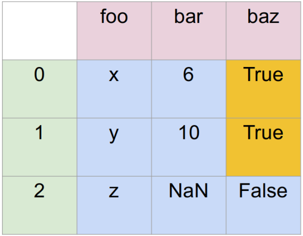
	foo	bar	baz
0	x	6	True
1	y	10	True
2	z	NaN	False



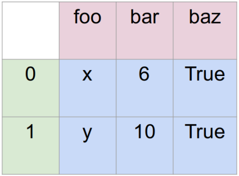
	foo	bar	baz
0	x	6	True
1	y	10	True

Conditional Filtering

```
df[(df[‘baz’])]
```



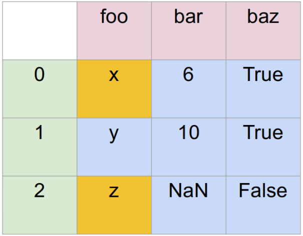
	foo	bar	baz
0	x	6	True
1	y	10	True
2	z	NaN	False



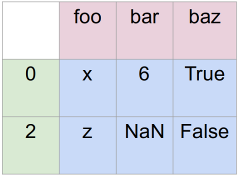
	foo	bar	baz
0	x	6	True
1	y	10	True

Conditional Filtering

```
df[ (df[’foo’] == ’x’) | (df[’foo’] == ’z’) ]
```



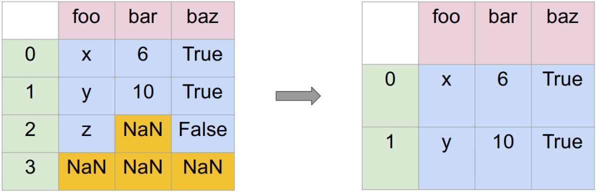
	foo	bar	baz
0	x	6	True
1	y	10	True
2	z	NaN	False



	foo	bar	baz
0	x	6	True
2	z	NaN	False

Handling Missing Values

```
new_df = df.dropna()
```

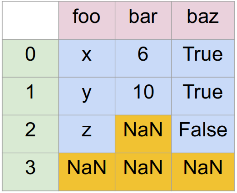


	foo	bar	baz
0	x	6	True
1	y	10	True
2	z	NaN	False
3	NaN	NaN	NaN

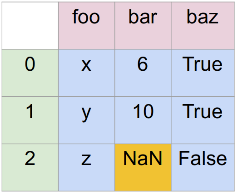
	foo	bar	baz
0	x	6	True
1	y	10	True

Handling Missing Values

```
new_df = df.dropna(how='all')
```



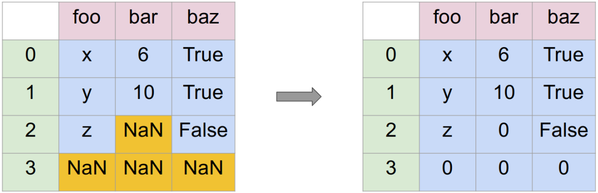
	foo	bar	baz
0	x	6	True
1	y	10	True
2	z	NaN	False
3	NaN	NaN	NaN



	foo	bar	baz
0	x	6	True
1	y	10	True
2	z	NaN	False

Handling Missing Values

```
new_df = df.fillna(0)
```

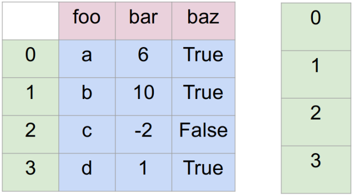


	foo	bar	baz
0	x	6	True
1	y	10	True
2	z	NaN	False
3	NaN	NaN	NaN

	foo	bar	baz
0	x	6	True
1	y	10	True
2	z	0	False
3	0	0	0

Indexing

```
ix = df.index
```

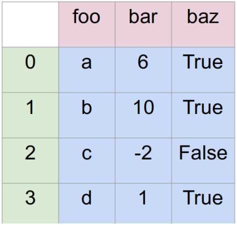


	foo	bar	baz
0	a	6	True
1	b	10	True
2	c	-2	False
3	d	1	True

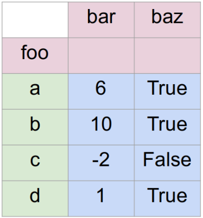
0
1
2
3

Indexing

```
df = df.set_index('foo')
```



	foo	bar	baz
0	a	6	True
1	b	10	True
2	c	-2	False
3	d	1	True

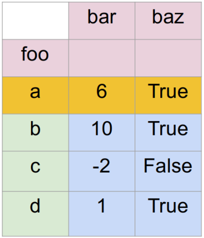


	bar	baz
foo		
a	6	True
b	10	True
c	-2	False
d	1	True

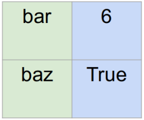
Indexing

By label or by position

```
df.loc['a']
df.iloc[0]
```



	bar	baz
foo		
a	6	True
b	10	True
c	-2	False
d	1	True



bar	6
baz	True

Descriptive Statistics

```
df.sum() #Sum of values
df.cumsum() #Cumulative sum of values
df.min() # Min value
df.max() #Max value
df.describe() #Summary statistics
df.mean() #Mean of values
df.median() #Median of values
```

Grouping and Sorting

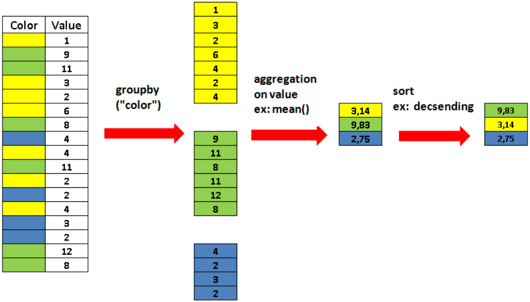
Grouping

```
df.groupby(‘a’)
```

Sorting

```
df.sort_values(by=[’col1’, ’col2’])
df.sort_values(by=’col1’, ascending=False)
df.sort_values(by=’col1’, ascending=True)
```

Group, Mean and Sort



Color	Value
🟨	1
🟩	9
🟩	11
🟨	3
🟨	2
🟨	6
🟩	8
🟦	4
🟨	4
🟩	11
🟨	2
🟦	2
🟨	4
🟦	3
🟦	2
🟩	12
🟩	8

groupby
("color")

🟨	1
🟨	3
🟨	2
🟨	6
🟨	4
🟨	2
🟨	4

🟩	9
🟩	11
🟩	8
🟩	11
🟩	12
🟩	8

🟦	4
🟦	2
🟦	3
🟦	2

aggregation
on value
ex: mean()

🟨	3,14
🟩	9,83
🟦	2,75

sort
ex: decsending

🟩	9,83
🟨	3,14
🟦	2,75

File I/O

CSV

```
pd.read_csv('foo.csv')
df.to_csv('mydataFrame.csv')
```

Excel

```
pd.read_excel('foo.xlsx')
df.to_excel('mydataFrame.xlsx')
```