

# Dataframes

## Pandas

It is an open-source python library that is used for data manipulation and analysis. It provides many functions and methods to speed up the data analysis process. Pandas is built on top of the NumPy package, hence it takes a lot of basic inspiration from it. The two primary data structures are Series which is 1 dimensional and DataFrame which is 2 dimensional.

### ▼ First let us import the Pandas module

```
1 import pandas as pd
```

### ▼ Read the csv file from google drive

```
1 #mount drive
2 from google.colab import drive
3 drive.mount('/content/drive')
4 #read from google drive
5 data = pd.read_csv('/content/drive/My Drive/Data Science/uk-500.csv')
```

### ▼ Read from local drive

```
1 from google.colab import files
2 files=files.upload()
```

## Dataframe indexing

.iloc is primarily integer position based (from 0 to length-1 of the axis), but may also be used with a boolean array. .iloc will raise IndexError if a requested indexer is out-of-bounds, except slice indexers which allow out-of-bounds indexing. (this conforms with Python/NumPy slice semantics).

### ▼ Extracting rows using Pandas .iloc[]

```
1 data.iloc[0]
```

```

first_name      Aleshia
last_name       Tomkiewicz
company_name     Alan D Rosenberg Cpa Pc
address         14 Taylor St
city            St. Stephens Ward
county          Kent
postal          CT2 7PP
phone1          01835-703597
phone2          01944-369967
email           atomkiewicz@hotmail.com
web             http://www.alandrosenburgcpapc.co.uk
Name: 0, dtype: object

```

```
1 data.iloc[1]
```

```

first_name      Evan
last_name       Zigomalas
company_name     Cap Gemini America
address         5 Binney St
city            Abbey Ward
county          Buckinghamshire
postal          HP11 2AX
phone1          01937-864715
phone2          01714-737668
email           evan.zigomalas@gmail.com
web             http://www.capgeminiamerica.co.uk
Name: 1, dtype: object

```

```
1 data.iloc[-1]
```

```

first_name      Mi
last_name       Richan
company_name     Nelson Wright Haworth Golf Crs
address         6 Norwood Grove
city            Tanworth-in-Arden
county          Warwickshire
postal          B94 5RZ
phone1          01451-785624
phone2          01202-738406
email           mi@hotmail.com
web             http://www.nelsonwriighthaworthgolfcrs.co.uk
Name: 499, dtype: object

```

## ▼ Extracting columns with index

```

1 #selecting columns
2 data.iloc[:,0]

```

```

0      Aleshia
1      Evan
2      France
3      Ulysses
4      Tyisha
...
495     Avery
496     Reid
497   Charlette
498   Celestina
499         Mi
Name: first_name, Length: 500, dtype: object

```

```
1 data.iloc[:,1]
```

```

0      Tomkiewicz
1      Zigomalas
2      Andrade
3      Mcwalters
4      Veness
...
495     Veit
496     Euresti
497   Brenning
498     Keeny
499     Richan
Name: last_name, Length: 500, dtype: object

```

```
1 data.iloc[:, -1]
```

```

0      http://www.alandrosenburgcpapc.co.uk
1      http://www.capgeminiamerica.co.uk
2      http://www.elliottjohnwesq.co.uk
3      http://www.mcmahanbenl.co.uk
4      http://www.champagneroom.co.uk
...
495     http://www.plazagourmetdelicatessen.co.uk
496     http://www.fitzgeraldedwardj.co.uk
497     http://www.fureyassociates.co.uk
498     http://www.bfgfederalcreditunion.co.uk
499     http://www.nelsonwriighthaworthgolfcrs.co.uk
Name: web, Length: 500, dtype: object

```

## ▼ Extracting multiple rows with index

```

1 #first upto n rows select
2 data.iloc[0:5] #0,1,2,3,4

```

	first_name	last_name	company_name	address	city	county	postal	ph
0	Aleshia	Tomkiewicz	Alan D Rosenburg Cpa Pc	14 Taylor St	St. Stephens Ward	Kent	CT2 7PP	01 70
1	Evan	Zigomalas	Cap Gemini America	5 Binney St	Abbey Ward	Buckinghamshire	HP11 2AX	01 86
2	France	Andrade	Elliott, John W Esq	8 Moor Place	East Southbourne and Tuckton W	Bournemouth	BH6 3BE	01 36
3	Ulysses	Mcwalters	Mcmahan, Ben L	505 Exeter Rd	Hawerby cum Beesby	Lincolnshire	DN36 5RP	01 77
4	Tyisha	Veness	Champagne Room	5396 Forth	Greets Green and	West Midlands	B70 9DT	01 42

## Extracting multiple columns with index

### ▼ Extracting multiple columns with index

```
1 #first two columns
2 data.iloc[:,0:2]
```

**first\_name last\_name**

---

## ▼ Extracting multiple rows and multiple columns with index

**2** France Andrade

```
1 data.iloc[[0,3,6,24],[0,5,6]]
```

	<b>first_name</b>	<b>county</b>	<b>postal</b>
<b>0</b>	Aleshia	Kent	CT2 7PP
<b>3</b>	Ulysses	Lincolnshire	DN36 5RP
<b>6</b>	Marg	Southampton	SO14 3TY
<b>24</b>	Tess	West Sussex	PO19 1RH

## ▼ Extracting continous rows and continous columns with index

```
1 data.iloc[0:5,5:8]
```

	<b>county</b>	<b>postal</b>	<b>phone1</b>
<b>0</b>	Kent	CT2 7PP	01835-703597
<b>1</b>	Buckinghamshire	HP11 2AX	01937-864715
<b>2</b>	Bournemouth	BH6 3BE	01347-368222
<b>3</b>	Lincolnshire	DN36 5RP	01912-771311
<b>4</b>	West Midlands	B70 9DT	01547-429341

## ▼ .set\_index Set the DataFrame index using existing columns.

`DataFrame.set_index(keys, drop=True, append=False, inplace=False, verify_integrity=False)`

```
1 data.set_index("last_name",inplace=True)
```

	first_name	company_name	address	city	county	postal	phon
last_name							
Tomkiewicz	Aleshia	Alan D Rosenburg Cpa Pc	14 Taylor St	St. Stephens Ward	Kent	CT2 7PP	018: 7035
Zigomalas	Evan	Cap Gemini America	5 Binney St	Abbey Ward	Buckinghamshire	HP11 2AX	019: 8647
Andrade	France	Elliott, John W Esq	8 Moor Place	East Southbourne and Tuckton W	Bournemouth	BH6 3BE	013: 3682
Mcwalters	Ulysses	Mcmahan, Ben I	505 Exeter	Hawerby Cum Beech	Lincolnshire	DN36 5PD	019: 7715

▼ pandas.DataFrame.loc

property DataFrame.loc

Access a group of rows and columns by label(s) or a boolean array.

.loc[] is primarily label based, but may also be used with a boolean array.#To see nth row/column as well

```
1 #includes the nth row/column as well
2 data.loc[['Andrade', 'Veness'],'city':'email']
```

	city	county	postal	phone1	phon
last_name					
Andrade	East Southbourne and Tuckton W	Bournemouth	BH6 3BE	01347-368222	01935-8216
Veness	Greets Green and Lyng Ward	West Midlands	B70 9DT	01547-429341	01290-3672

Double-click (or enter) to edit

```
1
2 data.loc['Andrade': 'Veness',['first_name','city','address']]
```

first\_name city address

last\_name

## ▼ Display whose firstname is Antonio

```
1 data.loc[data['first_name']=='Antonio', 'city':'email']
```

last_name	city	county	postal	phone1	phone2
<b>Villamarin</b>	Little Parndon and Hare Street	Hertfordshire	CM20 2HT	01559-403415	01388-777812
<b>Glasford</b>	Gaer Community	Newport	NP20 3DE	01463-409090	01242-318420
<b>Heilig</b>	Ipplepen	Devon	TQ12 5LL	01324-171614	01442-946357

## ▼ Display whose firstname is Aleshia and from Kent

```
1 data.loc[((data['county']=='Kent') & (data['first_name']=='Aleshia')), 'city':'email']
```

last_name	city	county	postal	phone1	phone2
<b>Tomkiewicz</b>	St. Stephens Ward	Kent	CT2 7PP	01835-703597	01944-369967 atomkiewicz@h

## ▼ Display people having gmail ids.

```
1 data.loc[data['email'].str.endswith("gmail.com")]
```

	first_name	company_name	address	city	county	postal	phone1
last_name							
<b>Zigomalas</b>	Evan	Cap Gemini America	5 Binney St	Abbey Ward	Buckinghamshire	HP11 2AX	01937-864715
<b>Erm</b>	Charlesetta	Cain, John M Esq	5 Hygeia St	Loundsley Green Ward	Derbyshire	S40 4LY	01276-816806
<b>Jaret</b>	Corrinne	Sound Vision Corp	2150 Morley St	Dee Ward	Dumfries and Galloway	DG8 7DE	01625-932209
<b>Quarto</b>	Karma	J C S Machinery	1 Birkett St	Shard End Ward	West Midlands	B33 0NH	01857-864722
<b>Savidge</b>	Milly	Bridgeway Plan For Health	129 Alexander Pope St	Franch Ward	Hereford and Worcester	DY11 9BW	01702-725589
...	...	...	...	...	...	...	...
<b>Picciuto</b>	Mel	Avante Limited	8598 Rumney Rd	Hylands Ward	Essex	RM11 1QL	01885-270480
<b>Machalek</b>	Tijuana	Schwartz, Thomas E Esq	1149 Highfield Rd #996	Fareham South Ward	Hampshire	PO14 9JJ	01242-532395
			236				

## ► Display whose firstname has 'France', 'Tyisha', 'Eric'

Pandas `isin()` method is used to filter data frames. `isin()` method helps in selecting rows with having a particular(or Multiple) value in a particular column.

```
1 data.loc[data['first_name'].isin(['France', 'Tyisha', 'Eric'])]
```



first_name	company_name	address	city	county	postal	phone1	phone2
------------	--------------	---------	------	--------	--------	--------	--------

The `endswith()` method returns True if the string ends with the specified value, otherwise False.

Display people whose name contains Antonio and has gmail ids

```
1 data.loc[data['email'].str.endswith("gmail.com") & (data['first_name'] == 'Antonio')]
```

	first_name	company_name	address	city	county	postal	phone1	phone2
last_name								
Villamarin	Antonio	Combs Sheetmetal	353 Standish St #8264	Little Parndon and Hare Street	Hertfordshire	CM20 2HT	01559-403415	01324-7778
Heilig	Antonio	Radisson Suite Hotel	35 Elton St #3	Ipplpen	Devon	TQ12 5LL	01324-171614	01434-9463

## ▼ Lambda functions

It offers a dual boost to a data scientist. You can write tidier Python code and speed up your machine learning tasks.

In Python, lambda functions have the following syntax: `lambda x: x`

e.g. To find cube of a number `(lambda x: xxx)(10)`

## Lambda with Apply

`apply()` function calls the lambda function and applies it to every row or column of the dataframe and returns a modified copy of the dataframe:

```
df['age']=df.apply(lambda x: x['age']+3,axis=1)
```

The `split()` method splits a string into a list.

You can specify the separator, default separator is any whitespace.

**Display people who has four words in their name.**

```
1 data.loc[data['company_name'].apply(lambda x: len(x.split(' '))==4)]
```

	first_name	company_name	address	city	county	postal
last_name						
<b>Andrade</b>	France	Elliott, John W Esq	8 Moor Place	East Southbourne and Tuckton W	Bournemouth	BH6 3BE
<b>Rampy</b>	Eric	Thompson, Michael C Esq	9472 Lind St	Desborough	Northamptonshire	NN14 2GH
<b>Erm</b>	Charlesetta	Cain, John M Esq	5 Hygeia St	Loundsley Green Ward	Derbyshire	S40 4LY
<b>Throssell</b>	Michell	Weiss Spirt & Guyer	89 Noon St	Carbrooke	Norfolk	IP25 6JQ
<b>Kanne</b>	Edgar	Crowan, Kenneth W Esq	99 Guthrie St	New Milton	Hampshire	BH25 5DF
...	...	...	...	...	...	...
<b>Alsaqri</b>	Ahmad	Alliance Construction Co Inc	21 Pickwick St	Sutton cum Duckmanton	Derbyshire	S44 5DS
<b>Reibman</b>	Jacquelyne	Great Clips For Hair	70 Lilly Rd	Shirley Ward	Greater London	CR0 7PT
<b>Kono</b>	Isabelle	Rock Springs Petroleum Equip	4920 Fazakerley Rd	Plymstock Dunstone Ward	Devon	PL9 8RD
<b>Fiorino</b>	Elbert	Donald, G Nelson Esq	726 Westmoreland Place	Ballochmyle Ward	East Ayrshire	KA5 6EL
<b>Keeny</b>	Celestina	Bfg Federal Credit Union	9 Milton St	Consett North ED	County Durham	DH8 5LP

192 rows × 10 columns

## ▼ Reading a csv file from github

```
1 df = pd.read_csv('https://raw.githubusercontent.com/fivethirtyeight/data/master/cabinet-tu
```

```
1 df.head()
```

	president	position	appointee	start	end
0	Carter	OMB Director	Bert Lance	1/21/77	9/23/77
1	Carter	Secretary of Transportation	Brock Adams	1/23/77	7/20/79
2	Carter	Secretary of Health, Education & Welfare	Joseph Califano Jr.	1/25/77	8/3/79
3	Carter	Secretary of Housing & Urban Development	Patricia Harris	1/23/77	8/3/79
4	Carter	Secretary of the Treasury	W. Michael Blumenthal	1/23/77	8/14/79

```
1 df.dtypes
```

```

president    object
position     object
appointee    object
start        object
end          object
length       object
days        float64
dtype: object

```

```
1 df.shape
```

```
(312, 7)
```

```
1 df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 312 entries, 0 to 311
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype
---  -
0   president   312 non-null    object
1   position    312 non-null    object
2   appointee   312 non-null    object
3   start       312 non-null    object
4   end         312 non-null    object
5   length      294 non-null    object
6   days        288 non-null    float64
dtypes: float64(1), object(6)
memory usage: 17.2+ KB

```

```
1 #display first 5 lines
```

```
2 df.head()
```

	president	position	appointee	start	end
0	Carter	OMB Director	Bert Lance	1/21/77	9/23/77

## ▼ Pandas Index.nunique()

This function return number of unique elements in the object. It returns a scalar value which is the count of all the unique values in the Index.

By default the NaN values are not included in the count. If dropna parameter is set to be False then it includes NaN value in the count.

```
1 df['president'].nunique()
```

7

```
1 #OMB directors in the period of clinton, number of presidents, and secretaries
2 df.loc[df.position.str.contains("Secretary")]
```

	president	position	appointee	start	end
1	Carter	Secretary of Transportation	Brock Adams	1/23/77	7/1/77
2	Carter	Secretary of Health, Education & Welfare	Joseph Califano Jr.	1/25/77	8/1/77
3	Carter	Secretary of Housing & Urban Development	Patricia Harris	1/23/77	8/1/77
4	Carter	Secretary of the Treasury	W. Michael Blumenthal	1/23/77	8/1/77
7	Carter	Secretary of Energy	James Schlesinger	8/6/77	8/1/77
...	...	...	...	...	...
305	Trump	Secretary of Labor	Alexander Acosta	4/28/17	Still in
307	Trump	Secretary of Homeland Security	Kirstjen Nielsen	12/6/17	Still in
308	Trump	Secretary of Health & Human Services	Alex Azar	1/29/18	Still in
309	Trump	Secretary of State	Mike Pompeo	4/26/18	Still in
311	Trump	Secretary of Veterans Affairs	Robert Wilkie	7/30/18	Still in

175 rows × 7 columns

```
1 df.loc[(df['president']=='Clinton') & (df.position.str.contains("OMB Director"))]['appoint
```

```
122 Leon Panetta
129 Alice Rivlin
142 Frank Raines
151 Jack Lew
```

Name: appointee, dtype: object

