Python

Introduction to Pandas

In this lecture

- Introduction to Pandas
- Pandas Series
- Pandas DataFrame
- Operations
  - DataFrame
  - Column
- Conditions
- GroupBy
- Input / Output of CSVs
- Concat, Merge and Join

Pandas

Pandas

- **Pandas** is a package that supplies data analysis tools in Python.
- DataFrames provide convenient formatting to visualise the data source. Often CSVs, and even NumPy arrays are used as the source for these DataFrames.
- It provides useful functions from summaries and descriptive statistics of data, in addition to SQL joins (GroupBy)

More documentation available at:
https://pandas.pydata.org

Import Pandas

In[ ]:

```
1 | import pandas as pd
2 | pd
3 |
```

Import Pandas

In[ ]:
```
1 | import pandas as pd
2 | pd
3 |
```

Out[]:
```
<module 'pandas' from
'/Users/nick/anaconda3/lib/python3.11/
site-packages/pandas/__init__.py'>
```

Imports

In[ ]:

```
1 | import pandas as pd
2 | import numpy as np
3 |
```

Imports

In[ ]:
```
1 | import pandas as pd
2 | import numpy as np
3 |
```

Written in the first cell to reduce duplication across the following cells.

Series

Series from a list

In[ ]:
```
1 | my_list = [10,20,30]
2 | pd.Series(data=my_list)
3 |
```

Series from a list

In[ ]:
```
1 | my_list = [10,20,30]
2 | pd.Series(data=my_list)
3 |
```

Out[]:
```
0    10
1    20
2    30
dtype: int64
```

With labels

In[ ]:
```
labels = ['a','b','c']
my_list = [10,20,30]
pd.Series(data=my_list,index=labels)
```

With labels

In[ ]:
```
labels = ['a','b','c']
my_list = [10,20,30]
pd.Series(data=my_list,index=labels)
```

Out[]:
```
a    10
b    20
c    30
dtype: int64
```

No descriptions

In[ ]:
```
labels = ['a','b','c']
my_list = [10,20,30]
pd.Series(my_list,labels)
```

No descriptions

In[ ]:
```
1 | labels = ['a','b','c']
2 | my_list = [10,20,30]
3 | pd.Series(my_list,labels)
```

Out[]:
```
a    10
b    20
c    30
dtype: int64
```

NumPy Arrays

In[ ]:
```
1 | arr = np.array([10,20,30])
2 | pd.Series(arr)
3 |
```

NumPy Arrays

In[ ]:
```
1 | arr = np.array([10,20,30])
2 | pd.Series(arr)
3 |
```

Out[]:
```
0    10
1    20
2    30
dtype: int64
```

Dictionaries

In[ ]:
```
1 | d = {'a':10,'b':20,'c':30}
2 | pd.Series(d)
3 |
```

Dictionaries

In[ ]:
```
1 | d = {'a':10,'b':20,'c':30}
2 | pd.Series(d)
3 |
```

Out[]:
```
a    10
b    20
c    30
dtype: int64
```

Series

In[ ]:
```
ser1 = pd.Series([1,2,3],
      index = ['UK', 'USA', 'EU'])
ser1
```

Series

In[ ]:
```
1 | ser1 = pd.Series([1,2,3],
2 |       index = ['UK', 'USA', 'EU'])
3 | ser1
```

Out[]:
```
UK         1
USA        2
EU         3
dtype: int64
```

Indexing

In[ ]:
```
ser1 = pd.Series([1,2,3],
      index = ['UK', 'USA', 'EU'])
ser1['USA']
```

```
UK      1
USA     2
EU      3
dtype: int64
```

Indexing

In[ ]:
```
1 | ser1 = pd.Series([1,2,3],
2 |       index = ['UK', 'USA', 'EU'])
3 | ser1['USA']
```

Out[]: 2

Series

In[ ]:
```
ser2 = pd.Series([5,3,7],
      index = ['UK', 'USA', 'ASIA'])
ser2
```

Series

In[ ]:
```
1 | ser2 = pd.Series([5,3,7],
2 |       index = ['UK', 'USA', 'ASIA'])
3 | ser2
```

Out[]:
```
UK      5
USA     3
ASIA    7
dtype: int64
```

Series

```
In[ ]: 1 | ser1 + ser2
       2 |
```

```
UK        1
USA       2
EU        3
dtype: int64
```

```
UK        5
USA       3
ASIA      7
dtype: int64
```

Series

In[ ]:
```
1 | ser1 + ser2
2 |
```

Out[]:
```
ASIA      NaN
EU        NaN
UK        6.0
USA       5.0
dtype: float64
```

Series

In[ ]:
```
1 | np.add(ser1, ser2)
2 |
```

Series

```
In[ ]: 1 | np.add(ser1, ser2)
       2 |
```

```
Out[]: ASIA      NaN
       EU        NaN
       UK        6.0
       USA       5.0
       dtype: float64
```

Concat

In[ ]:
```
1 | pd.concat([ser1, ser2])
2 |
```

Concat

In[ ]:
```
1 | pd.concat([ser1, ser2])
2 |
```

Out[]:
```
UK      1
USA     2
EU      3
UK      5
USA     3
ASIA    7
dtype: int64
```

DataFrame

randn

In[ ]:
```
1 | np.random.randn(3,3)
2 |
```

randn

```
In[ ]: 1 | np.random.randn(3,3)
       2 |
```

```
Out[]: array([[ 0.184572,  0.807191,  0.07229368],
              [ 0.638787,  0.329463, -0.49471402],
              [-0.754067, -0.943064,  0.48675165]])
```

DataFrame

In[ ]:
```
df = pd.DataFrame(np.random.randn(3,3),
     index='A B C'.split(),
     columns='X Y Z'.split())
df
```

DataFrame

In[ ]:
```
1 | df = pd.DataFrame(np.random.randn(3,3),
2 |      index='A B C'.split(),
3 |      columns='X Y Z'.split())
4 | df
```

Out[]:

	X	Y	Z
A	2.605967	0.683509	0.302665
B	1.693723	-1.706086	-1.159119
C	-0.134841	0.390528	0.166905

Col access

In[ ]:
```
1 | df = pd.DataFrame(np.random.randn(3,3),
2 |       index='A B C'.split(),
3 |       columns='X Y Z'.split())
4 | df['Z']
```

	X	Y	Z
A	2.605967	0.683509	**0.302665**
B	1.693723	-1.706086	**-1.159119**
C	-0.134841	0.390528	**0.166905**

Col access

In[ ]:
```
df = pd.DataFrame(np.random.randn(3,3),
     index='A B C'.split(),
     columns='X Y Z'.split())
df['Z']
```

Out[]:
```
A    0.302665
B   -1.159119
C    0.166905
Name: Z, dtype: float64
```

Multi-col access

In[ ]:
```
1 | df = pd.DataFrame(np.random.randn(3,3),
2 |       index='A B C'.split(),
3 |       columns='X Y Z'.split())
4 | df[['X','Z']]
```

	X	Y	Z
A	2.605967	0.683509	0.302665
B	1.693723	-1.706086	-1.159119
C	-0.134841	0.390528	0.166905

Multi-col access

In[ ]:
```
df = pd.DataFrame(np.random.randn(3,3),
     index='A B C'.split(),
     columns='X Y Z'.split())
df[['X','Z']]
```

Out[]:

	X	Z
A	2.605967	0.302665
B	1.693723	-1.159119
C	-0.134841	0.166905

Add new column

In[ ]:
```
df = pd.DataFrame(np.random.randn(3,3),
     index='A B C'.split(),
     columns='X Y Z'.split())
df['W'] = [0.1, 0.2, 0.3]
```

	X	Y	Z
A	2.605967	0.683509	0.302665
B	1.693723	-1.706086	-1.159119
C	-0.134841	0.390528	0.166905

Add new column

In[ ]:
```
1 | df = pd.DataFrame(np.random.randn(3,3),
2 |       index='A B C'.split(),
3 |       columns='X Y Z'.split())
4 | df['W'] = [0.1, 0.2, 0.3]
```

Out[]:

	X	Y	Z	W
A	2.605967	0.683509	0.302665	**0.1**
B	1.693723	-1.706086	-1.159119	**0.2**
C	-0.134841	0.390528	0.166905	**0.3**

Drop column

In[ ]:
```
1 | df = pd.DataFrame(np.random.randn(3,3),
2 |       index='A B C'.split(),
3 |       columns='X Y Z'.split())
4 | df = df.drop(['W'], axis = 1)
```

	X	Y	Z	~~W~~
A	2.605967	0.683509	0.302665	~~0.1~~
B	1.693723	-1.706086	-1.159119	~~0.2~~
C	-0.134841	0.390528	0.166905	~~0.3~~

Drop column

In[ ]:

```
df = pd.DataFrame(np.random.randn(3,3),
     index='A B C'.split(),
     columns='X Y Z'.split())
df = df.drop(['W'], axis = 1)
```

Out[]:

	X	Y	Z
A	2.605967	0.683509	0.302665
B	1.693723	-1.706086	-1.159119
C	-0.134841	0.390528	0.166905

Row access ‘loc’

In[ ]:
```
1 | df = pd.DataFrame(np.random.randn(3,3),
2 |       index='A B C'.split(),
3 |       columns='X Y Z'.split())
4 | df.loc['A']
```

	X	Y	Z
A	**2.605967**	**0.683509**	**0.302665**
B	1.693723	-1.706086	-1.159119
C	-0.134841	0.390528	0.166905

Row access ‘loc’

In[ ]:
```
1 | df = pd.DataFrame(randn(3,3),
2 |       index='A B C'.split(),
3 |       columns='X Y Z'.split())
4 | df.loc['A']
```

Out[]:
```
X    2.605967
Y    0.683509
Z    0.302665
Name: A, dtype: float64
```

iloc

In[ ]:
```
df = pd.DataFrame(randn(3,3),
     index='A B C'.split(),
     columns='X Y Z'.split())
df.iloc[0]
```

iloc

In[ ]:
```
1 | df = pd.DataFrame(randn(3,3),
2 |       index='A B C'.split(),
3 |       columns='X Y Z'.split())
4 | df.iloc[0]
```

Out[]:
```
X    2.605967
Y    0.683509
Z    0.302665
Name: A, dtype: float64
```

Basic Operations

Example DF

In[ ]:

```
data = {
'Name': ['Alice', 'Bob', 'Charlie', 'David', 'Eva'],
'Age' : [25, 30, 35, 40, 45]
}
df = pd.DataFrame(data)
```

Head

In[ ]:
```
1 | df.head()
2 |
```

Head

In[ ]:
```
1 | df.head()
2 |
```

Out[]:

	Name	Age
0	Alice	25
1	Bob	30
2	Charlie	35
3	David	40
4	Eva	45

Head

In[ ]:
```
1 | df.head(3)
2 |
```

Head

In[ ]:
```
1 | df.head(3)
2 |
```

Out[]:

	Name	Age
0	Alice	25
1	Bob	30
2	Charlie	35

Tail

In[ ]:
```
1 | df.tail()
2 |
```

Tail

In[ ]:
```
1 | df.tail()
2 |
```

Out[]:

	Name	Age
0	Alice	25
1	Bob	30
2	Charlie	35
3	David	40
4	Eva	45

Tail

In[ ]:
```
1 | df.tail(3)
2 |
```

Tail

In[ ]:
```
1 | df.tail(3)
2 |
```

Out[]:

	Name	Age
2	Charlie	35
3	David	40
4	Eva	45

Extend the DF

In[ ]:

```
data = {
'Name': ['Alice', 'Bob', 'Charlie', 'David', 'Eva'],
'Age' : [25, 30, 35, 40, 45],
'City': ['NY', 'LA', 'Chicago', 'Houston', 'Miami'],
'Salary': [50000, 60000, 75000, 90000, 80000]
}
df = pd.DataFrame(data)
```

Info

In[ ]:
```
1 | df.info()
2 |
```

Info

Out[]:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   Name    5 non-null      object
 1   Age     5 non-null      int64 
 2   City    5 non-null      object
 3   Salary  5 non-null      int64 
dtypes: int64(2), object(2)
memory usage: 288.0+ bytes
```

isNA - null

In[ ]:
```
1 | df.isna().sum()
2 |
```

isNA - null

In[ ]:
```
1 | df.isna().sum()
2 |
```

Out[]:
```
Name      0
Age       0
City      0
Salary    0
dtype: int64
```

Describe

In[ ]:
```
1 | df.describe()
2 |
```

Describe

Out[]:

	Age	Salary
count	5.00000	5.00000
mean	35.00000	71000.000
std	7.905694	15968.71942
min	25.00000	50000.00000
25%	30.00000	60000.00000
50%	35.00000	75000.00000
75%	40.00000	80000.0000
max	45.00000	90000.0000

Operations on Columns

Subscript []

In[ ]:
```
1 | df["Salary"]
2 |
```

Subscript []

```
In[ ]: 1 | df["Salary"]
       2 |
```

```
Out[]: 0    50000
       1    60000
       2    75000
       3    90000
       4    80000
       Name: Salary, dtype: int64
```

Multiple columns

In[ ]:
```
1 | df[ ["Salary", "Age"] ]
2 |
```

Subscript []

In[ ]:
```
1 | df[ ["Salary", "Age"] ]
2 |
```

Out[]:

	Salary	Age
0	50000	25
1	60000	30
2	75000	35
3	90000	40
4	80000	45

Add a new column

In[ ]:
```
1 | zip_ = [42223, 90048, 60032, 77001, 90031]
2 | df["Zip"] = zip_
3 | df
```

Add a new column

In[ ]:
```
1 | zip_ = [42223, 90048, 60032, 77001, 90031]
2 | df["Zip"] = zip_
3 | df
```

Out[]:

	Name	Age	City	Salary	Zip
0	Alice	25	NY	50000	42223
1	Bob	30	LA	60000	90048
2	Charlie	35	Chicago	75000	60032
3	David	40	Houston	90000	77001
4	Eva	45	Miami	80000	90031

Drop a column

In[ ]:
```
1 | df = df.drop(["Zip"], axis = 1)
2 | df
```

Drop a column

In[ ]:
```
1 | df = df.drop(["Zip"], axis = 1)
2 | df
```

Out[]:

	Name	Age	City	Salary
0	Alice	25	NY	50000
1	Bob	30	LA	60000
2	Charlie	35	Chicago	75000
3	David	40	Houston	90000
4	Eva	45	Miami	80000

count() Salary

In[ ]:
```
1 | df["Salary"].count()
2 |
```

count() Salary

In[ ]:
```
1 | df["Salary"].count()
2 |
```

Out[]: 5

min() Salary

In[ ]:
```
1 | df["Salary"].min()
2 |
```

min() Salary

```
In[ ]: 1 | df["Salary"].min()
       2 |
```

```
Out[]:  50000
```

max() Salary

In[ ]:
```
1 | df["Salary"].max()
2 |
```

max() Salary

```
In[ ]: 1 | df["Salary"].max()
       2 |
```

```
Out[]: 90000
```

mean() Salary

In[ ]:
```
1 | df["Salary"].mean()
2 |
```

mean() Salary

```
In[ ]: 1 | df["Salary"].mean()
       2 |
```

```
Out[]:  71000.0
```

mean() multiple cols

```
In[ ]: 1 | df[ ["Salary","Age"] ].mean()
       2 |
```

mean() multiple cols

In[ ]:
```
1 | attrs = ["Salary", "Age"]
2 | df[attrs].mean()
```

mean() multiple cols

In[ ]:
```
1 | attrs = ["Salary", "Age"]
2 | df[attrs].mean()
```

Out[]:
```
Salary    71000.0
Age          35.0
dtype: float64
```

Conditions

Conditions

In[ ]:
```
1 | df["Name"] == "Bob"
2 |
```

Conditions

```
In[ ]: 1 | df["Name"] == "Bob"
       2 |
```

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

Conditions

In[ ]:
```
1 | df["Salary"] > 60000
2 |
```

Conditions

```
In[ ]: 1 | df["Salary"] > 60000
       2 |
```

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

Multiple conditions

In[ ]:
```
1 | df["Salary"] > 60000 & df["Age"] > 40
2 |
```

Ambiguity

In[ ]:
```
1 | df["Salary"] > 60000 & df["Age"] > 40
2 |
```

Out[]:
```
---------------------------------------------------------------------------
ValueError Traceback (most recent call last)
---------------------------------------------------------------------------
ValueError: The truth value of a Series is ambiguous. Use a.empty, a.bool(), a.item(), a.any() or a.all().
```

Separate filters

In[ ]:
```
1 | sal_filter = df["Salary"] > 60000
2 | age_filter = df["Age"] > 40
```

sal_filter

In[ ]:
```
1 | sal_filter = df["Salary"] > 60000
2 | sal_filter
```

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

age_filter

In[ ]:
```
1 | age_filter = df["Age"] > 40
2 | age_filter
```

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

Merge filters

In[ ]:
```
1 | sal_filter = df["Salary"] > 60000
2 | age_filter = df["Age"] > 40
3 | sal_filter[age_filter] == True
```

Merge filters

In[ ]:
```
1 | sal_filter = df["Salary"] > 60000
2 | age_filter = df["Age"] > 40
3 | sal_filter[age_filter] == True
```

Out[]:
```
4    True
Name: Salary, dtype: bool
```

Filter by row loc

In[ ]:
```
1 | sal_filter = df["Salary"] > 60000
2 | age_filter = df["Age"] > 40
3 | sal_filter[age_filter] == True
4 | df.iloc[4]
```

Filter by row loc

In[ ]:
```
1 | sal_filter = df["Salary"] > 60000
2 | age_filter = df["Age"] > 40
3 | sal_filter[age_filter] == True
4 | df.iloc[4]
```

Out[]:
```
Name       Eva
Age         45
City     Miami
Salary   80000
Name: 4, dtype: object
```

Easier way!

In[ ]:
```
1 | df[(df["Salary"] > 60000) & (df["Age"] > 40)]
2 |
```

Easier way!

In[ ]:
```
1 | df[(df["Salary"] > 60000) & (df["Age"] > 40)]
2 |
```

Out[]:

	Name	Age	City	Salary
4	Eva	45	Miami	80000

Multiple conditions

In[ ]:
```
df[(df["Salary"] > 60000)
                & (df["Age"] > 40)]["Name"]
```

Multiple conditions

In[ ]:
```
df[(df["Salary"] > 60000)
                & (df["Age"] > 40)]["Name"]
```

Out[]:
```
4    Eva
Name: Name, dtype: object
```

GroupBy

	species	sepal_length	sepal_width	petal_length	petal_width
0	setosa	5.1	3.5	1.4	0.2
1	setosa	4.9	3.0	1.4	0.2
2	setosa	4.7	3.2	1.3	0.2
3	setosa	4.6	3.1	1.5	0.2
4	setosa	5.0	3.6	1.4	0.2
50	versicolor	7.0	3.2	4.7	1.4
51	versicolor	6.4	3.2	4.5	1.5
52	versicolor	6.9	3.1	4.9	1.5
53	versicolor	5.5	2.3	4.0	1.3
54	versicolor	6.5	2.8	4.6	1.5
100	virginica	6.3	3.3	6.0	2.5
101	virginica	5.8	2.7	5.1	1.9
102	virginica	7.1	3.0	5.9	2.1
103	virginica	6.3	2.9	5.6	1.8
104	virginica	6.5	3.0	5.8	2.2

SUM

SUM

SUM

species	sepal_length	sepal_width	petal_length	petal_width
setosa	24.3	16.4	7.0	1.0
versicolor	32.3	14.6	22.7	7.2
virginica	32.0	14.9	28.4	10.5

Modify the DF

In[ ]:

```
data = {
'Name': ['Alice', 'Bob', 'Charlie', 'David', 'Eva'],
'Age' : [25, 30, 35, 40, 45],
'City': ['NY', 'LA', 'LA', 'NY', 'LA'],
'Salary': [50000, 50000, 75000, 75000, 75000]
}
df = pd.DataFrame(data)
```

GroupBy Salary

In[ ]:
```
1 | df.groupby("Salary").mean()
2 |
```

GroupBy Salary

In[ ]:
```
1 | df.groupby("Salary").mean()
2 |
```

Out[]:

Salary	Age
50000	27.5
75000	40.0

GroupBy Salary

In[ ]:
```
1 | df.groupby("Salary").min()
2 |
```

GroupBy Salary

In[ ]:
```
1 | df.groupby("Salary").min()
2 |
```

Out[]:

Salary	Name	Age	City
50000	Alice	25	LA
75000	Charlie	35	LA

GroupBy City

In[ ]:
```
1 | df.groupby("City").mean()
2 |
```

GroupBy City

In[ ]:
```
1 | df.groupby("City").mean()
2 |
```

Out[]:

City	Age	Salary
LA	36.6667	66666.6667
NY	32.5000	62500.0000

GroupBy City

In[ ]:
```
1 | df.groupby("City").max()
2 |
```

GroupBy City

In[ ]:
```
1 | df.groupby("City").max()
2 |
```

Out[]:

City	Name	Age	Salary
LA	Eva	45	75000
NY	David	40	75000

GroupBy City

In[ ]:
```
1 | df.groupby("City").max().loc["NY"]
2 |
```

GroupBy City

In[ ]:
```
1 | df.groupby("City").max().loc["NY"]
2 |
```

Out[]:
```
Name      David
Age          40
Salary    75000
Name: NY, dtype: object
```

GroupBy City

In[ ]:
```
1 | df.groupby("City").describe()
2 |
```

GroupBy City

In[ ]:
```
1 | df.groupby("City").describe()
2 |
```

Out[]:

	Age							
City	count	mean	std	min	25%	50%	75%	max
LA	3.0	36.6667	7.6376	30.0	32.50	35.0	40.0	45.0
NY	2.0	32.5000	10.6066	25.0	28.75	32.5	36.25	40.0

Input / Output

Read from CSV

In[ ]:
```
1 | data = pd.read_csv('dataset.csv')
2 |
3 |
```

Address,Lot,AM or PM,Browser Info,Company,Credit Card,CC Exp Date,CC Security Code,CC Provider,Email,Job,IP Address,Language,Purchase Price
"16629 Pace Camp Apt. 448
Alexisborough, NE 77130-7478",46 in,PM,Opera/9.56.(X11; Linux x86_64; sl-SI) Presto/2.9.183 Version/12.00,Martinez-Herman,6011929061123406,02/20,900,JCB 16
digit,pdunlap@yahoo.com,"Scientist, product/process development",149.146.147.205,el,98.14
"9374 Jasmine Spurs Suite 508
South John, TN 84355-4179",28 rn,PM,Opera/8.93.(Windows 98; Win 9x 4.90; en-US) Presto/2.9.176 Version/11.00,"Fletcher, Richards and
Whitaker",3337758169645356,11/18,561,Mastercard,anthony41@reed.com,Drilling engineer,15.160.41.51,fr,70.73
"Unit 0065 Box 5052
DPO AP 27450",94 vE,PM,Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.2; Trident/5.1),"Simpson, Williams and Pham",675957666125,08/19,699,JCB 16
digit,amymiller@morales-harrison.com,Customer service manager,132.207.160.22,de,0.95
"7780 Julia Fords
New Stacy, WA 45798",36 vm,PM,"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_0 rv:3.0; en-US) AppleWebKit/531.27.1 (KHTML, like Gecko) Version/5.1 Safari/
531.27.1","Williams, Marshall and Buchanan",6011578504430710,02/24,384,Discover,brent16@olson-robinson.info,Drilling engineer,30.250.74.19,es,78.04
"23012 Munoz Drive Suite 337
New Cynthia, TX 57826",20 IE,AM,Opera/9.58.(X11; Linux x86_64; it-IT) Presto/2.9.182 Version/11.00,"Brown, Watson and Andrews",6011456623207998,10/25,678,Diners
Club / Carte Blanche,christopherwright@gmail.com,Fine artist,24.140.33.94,es,77.82
"7502 Powell Mission Apt. 768
Travisland, VA 30493-5334",21 XT,PM,"Mozilla/5.0 (Macintosh; U; PPC Mac OS X 10_8_5) AppleWebKit/5312 (KHTML, like Gecko) Chrome/14.0.884.0 Safari/5312",Silva-
Anderson,30246185196287,07/25,7169,Discover,vnguyen@gmail.com,Fish farm manager,55.96.152.147,ru,25.15
"93971 Conway Causeway
Andersonburgh, AZ 75107",96 Xt,AM,Mozilla/5.0 (compatible; MSIE 7.0; Windows NT 5.0; Trident/3.0),Gibson and Sons,6011398782655569,07/24,714,VISA 16
digit,olivia04@yahoo.com,Dancer,127.252.144.18,de,88.56
"260 Rachel Plains Suite 366
Castroberg, WV 24804-9384",96 pG,PM,"Mozilla/5.0 (X11; Linux i686) AppleWebKit/5350 (KHTML, like Gecko) Chrome/15.0.841.0 Safari/5350",Marshall-
Collins,561252141909,06/25,256,VISA 13 digit,phillip48@parks.info,Event organiser,224.247.97.150,pt,44.25
"2129 Dylan Burg
New Michelle, ME 28650",45 JN,PM,"Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_7_9) AppleWebKit/5330 (KHTML, like Gecko) Chrome/14.0.898.0 Safari/5330",Galloway and
Sons,180041795790001,04/24,899,JCB 16 digit,kdavis@rasmussen.com,Financial manager,146.234.201.229,ru,59.54
"3795 Dawson Extensions
Lake Tinafort, ID 88739",15 Ug,AM,Mozilla/5.0 (X11; Linux i686; rv:1.9.7.20) Gecko/2011-11-30 06:02:34 Firefox/9.0,"Rivera, Buchanan and
Ramirez",4396283918371,01/17,931,American Express,gcoleman@hunt-huerta.com,Forensic scientist,236.198.199.8,zh,95.63
"650 Elizabeth Park
Lake Maria, LA 13526-2530",65 Yn,PM,"Mozilla/5.0 (iPod; U; CPU iPhone OS 4_1 like Mac OS X; sl-SI) AppleWebKit/531.41.4 (KHTML, like Gecko) Version/3.0.5 Mobile/
8B116 Safari/6531.41.4","Strickland, Michael and Gonzales",180036417827355,02/17,754,Voyager,ustewart@hotmail.com,"Development worker, community",26.59.93.1,el,96.89
"349 Laurie Parks
Thomasview, ID 08970",30 kK,PM,Mozilla/5.0 (X11; Linux i686; rv:1.9.6.20) Gecko/2014-05-12 06:09:34 Firefox/3.6.9,Kim-Oliver,869975209012056,06/26,9717,JCB 15
digit,johnnymiller@coleman.com,Diagnostic radiographer,128.222.40.234,en,19.26
"733 Heather Rest Apt. 670
Boltonport, UT 78662",69 DO,AM,"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_5_3 rv:2.0; sl-SI) AppleWebKit/532.38.2 (KHTML, like Gecko) Version/5.0.1 Safari/
532.38.2",Moore-Martin,5115990487067905,05/26,119,VISA 16 digit,tholt@hotmail.com,"Surveyor, quantity",236.71.234.240,en,39.65
"118 Melton Via Suite 681
Alexanderbury, FL 32104",36 bu,PM,"Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_8_6 rv:5.0; en-US) AppleWebKit/531.44.5 (KHTML, like Gecko) Version/5.0 Safari/
531.44.5",Keller PLC,4603635169938574,01/25,557,VISA 16 digit,caitlin57@yahoo.com,"Accountant, chartered public finance",84.212.92.11,it,8.93
"8774 Jason Keys Suite 427
East Scottborough, MS 29934",70 zH,AM,Mozilla/5.0 (Windows 98; it-IT; rv:1.9.2.20) Gecko/2013-01-15 08:05:10 Firefox/3.8,"Leach, Howe and
Ferguson",869967499275071,09/22,427,VISA 16 digit,aburns@yahoo.com,Acupuncturist,50.25.148.1,de,24.18
"31730 Chelsea Crest
Blakemouth, CT 90395-0620",41 Cj,PM,Opera/8.95.(Windows NT 5.0; en-US) Presto/2.9.164 Version/11.00,Garcia-Steele,180069437020404,04/25,404,Diners Club / Carte
Blanche,amanda39@yahoo.com,Retail manager,53.176.235.33,el,71.78

CSVs

iris

5.1	3.5	1.4	0.2	Iris-setosa
4.9	3.0	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa
5.0	3.6	1.4	0.2	Iris-setosa
5.4	3.9	1.7	0.4	Iris-setosa
4.6	3.4	1.4	0.3	Iris-setosa
5.0	3.4	1.5	0.2	Iris-setosa
4.4	2.9	1.4	0.2	Iris-setosa
4.9	3.1	1.5	0.1	Iris-setosa
5.4	3.7	1.5	0.2	Iris-setosa
4.8	3.4	1.6	0.2	Iris-setosa

Read from CSV

In[ ]:
```
1 | data = pd.read_csv('dataset.csv')
2 | type(data)
3 |
```

Read from CSV

In[ ]:
```
1 | data = pd.read_csv('dataset.csv')
2 | type(data)
3 |
```

Out[]: pandas.core.frame.DataFrame

Read from CSV

In[ ]:
```
1 | data = pd.read_csv('dataset.csv')
2 | df = pd.DataFrame(data)
3 | type(df)
```

Read from CSV

In[ ]:
```
1 | data = pd.read_csv('dataset.csv')
2 | df = pd.DataFrame(data)
3 | type(df)
```

Out[]: pandas.core.frame.DataFrame

Sample rows

In[ ]:
```
1 | data = pd.read_csv('dataset.csv')
2 | df = pd.DataFrame(data)
3 | df.sample(5)
```

Sample rows

Out[]:

	Address	Lot	AM or PM	Browser Info	Company	Credit Card	CC Exp Date	CC Security Code	CC Provider	Email	
7620	98023 Melanie Track Apt. 776\nSouth Tamara, NH...	84 qq	PM	Mozilla/5.0 (Windows NT 6.1; en-US; rv:1.9.2.2...	Keith and Sons	869942807600781	06/20	566	JCB 15 digit	robert04@yahoo.com	
5595	010 Smith Circles\nWest Virginiaborough, NY 96...	34 kF	PM	Mozilla/5.0 (Macintosh; U; PPC Mac OS X 10_5_0...	Curtis-Olsen	869918284664102	10/18	539	VISA 16 digit	nicole87@gmail.com	
3460	987 Foster Locks Apt. 224\nWest Mindystad, NH ...	22 UA	AM	Mozilla/5.0 (iPod; U; CPU iPhone OS 3_2 like M...	Bailey, Gibbs and Jackson	869928150212662	07/22	41	Maestro	tmills@yahoo.com	
1360	7073 Brittany Shoals Apt. 233\nLake Tonya, DE ...	89 lW	PM	Mozilla/5.0 (Macintosh; U; PPC Mac OS X 10_6_7...	Gibbs, Rodriguez and Jenkins	30084707162226	04/25	546	VISA 16 digit	daniel39@rice-alvarado.biz	
56	1099 Prince Locks Suite 900\nNorth	10 DU	AM	Mozilla/5.0 (Macintosh; U; Intel	Taylor, Lloyd and	3088511373952816	02/26	813	VISA 16 digit	albert68@lawrence-warren.biz	Clotl te

Apply methods

In[ ]:
```
1 | data = pd.read_csv('dataset.csv')
2 | df = pd.DataFrame(data)
3 | df.info()
```

Info…

```
Out[]: <class 'pandas.core.frame.DataFrame'>
       RangeIndex: 10000 entries, 0 to 9999
       Data columns (total 14 columns):
       #    Column       Non-Null Count  Dtype
       ---  ------       --------------  -----
       0    Address      10000 non-null object
       1    Lot          10000 non-null object
       2    AM or PM     10000 non-null object
       3    Browser Info10000 non-null object
       4    Company      10000 non-null object
       5    Credit Card 10000 non-null int64
       6    CC Exp Date 10000 non-null object
       …
```

Write to CSV

In[ ]:
```
1 | df.to_csv('dataset.csv',index=False)
2 |
3 |
```

CSVs

books.csv

child_mortality.csv

imdb-movies.csv

iris.csv

iris

5.1	3.5	1.4	0.2	Iris-setosa
4.9	3.0	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa
5.0	3.6	1.4	0.2	Iris-setosa
5.4	3.9	1.7	0.4	Iris-setosa
4.6	3.4	1.4	0.3	Iris-setosa
5.0	3.4	1.5	0.2	Iris-setosa
4.4	2.9	1.4	0.2	Iris-setosa
4.9	3.1	1.5	0.1	Iris-setosa
5.4	3.7	1.5	0.2	Iris-setosa
4.8	3.4	1.6	0.2	Iris-setosa

CSVs

Ecommerce Purchases

SMSSpamCollection

nltk imgs

Ecommerce Purchases

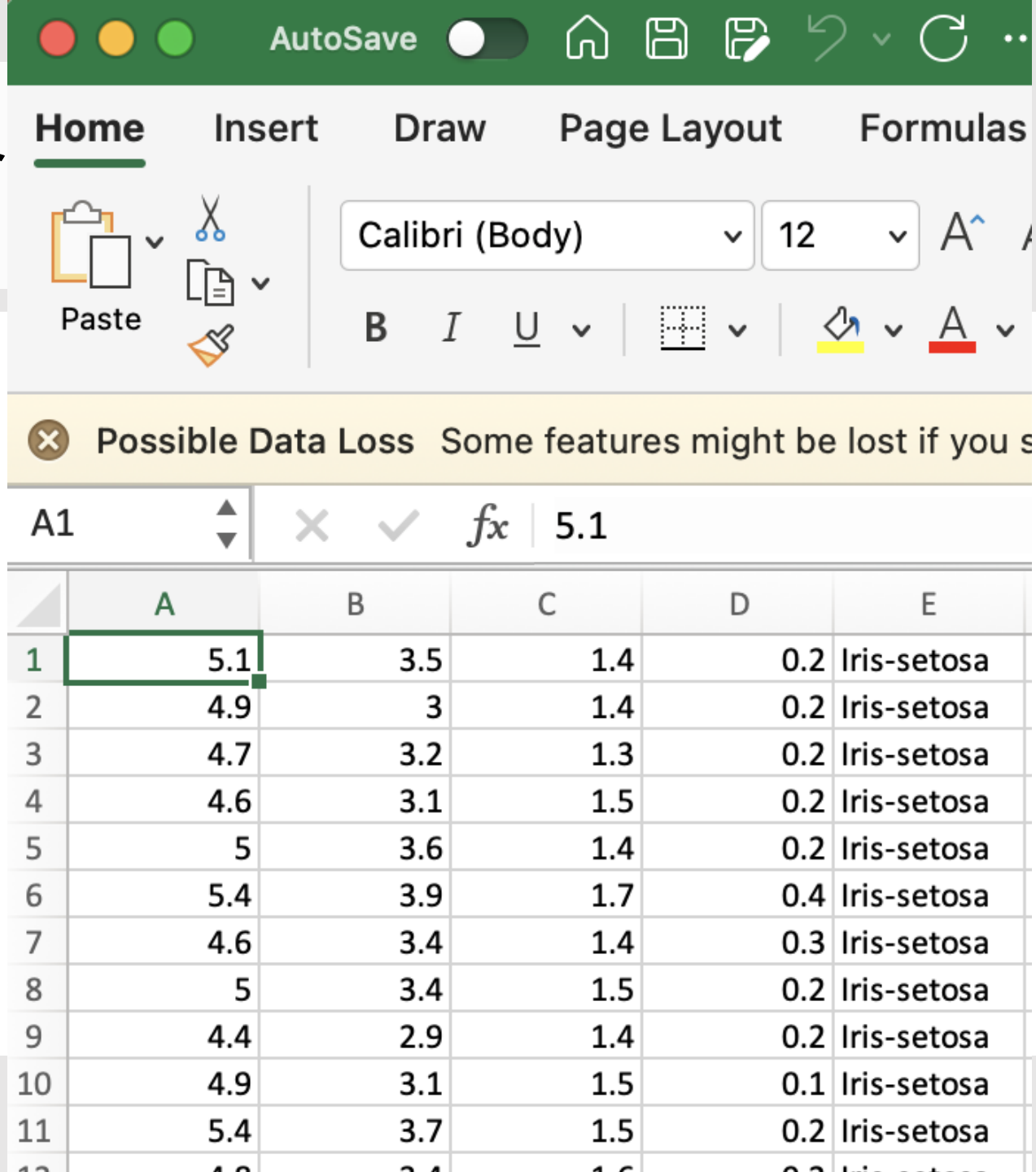
```
Address,Lot,AM or PM,Browser Info,Company,Credit Card,CC Exp Date,CC Security Code,CC Provider,Email,Job,IP Address
"16629 Pace Camp Apt. 448
Alexisborough, NE 77130-7478",46 in,PM,Opera/9.56.(X11; Linux x86_64; sl-SI) Presto/2.9.183 Version/12.00,Martinez-
digit,pdunlap@yahoo.com,"Scientist, product/process development",149.146.147.205,el,98.14
"9374 Jasmine Spurs Suite 508
South John, TN 84355-4179",28 rn,PM,Opera/8.93.(Windows 98; Win 9x 4.90; en-US) Presto/2.9.176 Version/11.00,"Fletc
Whitaker",3337758169645356,11/18,561,Mastercard,anthony41@reed.com,Drilling engineer,15.160.41.51,fr,70.73
"Unit 0065 Box 5052
DPO AP 27450",94 vE,PM,Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.2; Trident/5.1),"Simpson, Williams and Pham"
digit,amymiller@morales-harrison.com,Customer service manager,132.207.160.22,de,0.95
"7780 Julia Fords
New Stacy, WA 45798",36 vm,PM,"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_0 rv:3.0; en-US) AppleWebKit/531.27.1 (K
531.27.1","Williams, Marshall and Buchanan",6011578504430710,02/24,384,Discover,brent16@olson-robinson.info,Drillin
"23012 Munoz Drive Suite 337
New Cynthia, TX 57826",20 IE,AM,Opera/9.58.(X11; Linux x86_64; it-IT) Presto/2.9.182 Version/11.00,"Brown, Watson a
Club / Carte Blanche,christopherwright@gmail.com,Fine artist,24.140.33.94,es,77.82
"7502 Powell Mission Apt. 768
Travisland, VA 30493-5334",21 XT,PM,"Mozilla/5.0 (Macintosh; U; PPC Mac OS X 10_8_5) AppleWebKit/5312 (KHTML, like
Anderson,30246185196287,07/25,7169,Discover,vnguyen@gmail.com,Fish farm manager,55.96.152.147,ru,25.15
"93971 Conway Causeway
Andersonburgh, AZ 75107",96 Xt,AM,Mozilla/5.0 (compatible; MSIE 7.0; Windows NT 5.0; Trident/3.0),Gibson and Sons,6
digit,olivia04@yahoo.com,Dancer,127.252.144.18,de,88.56
"260 Rachel Plains Suite 366
Castroberg, WV 24804-9384",96 pG,PM,"Mozilla/5.0 (X11; Linux i686) AppleWebKit/5350 (KHTML, like Gecko) Chrome/15.0
Collins,561252141909,06/25,256,VISA 13 digit,phillip48@parks.info,Event organiser,224.247.97.150,pt,44.25
"2129 Dylan Burg
New Michelle, ME 28650",45 JN,PM,"Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_7_9) AppleWebKit/5330 (KHTML, like G
Sons,180041795790001,04/24,899,JCB 16 digit,kdavis@rasmussen.com,Financial manager,146.234.201.229,ru,59.54
"3795 Dawson Extensions
Lake Tinafort, ID 88739",15 Ug,AM,Mozilla/5.0 (X11; Linux i686; rv:1.9.7.20) Gecko/2011-11-30 06:02:34 Firefox/9.0,
Ramirez",4396283918371,01/17,931,American Express,gcoleman@hunt-huerta.com,Forensic scientist,236.198.199.8,zh,95.6
"650 Elizabeth Park
Lake Maria, LA 13526-2530",65 Yn,PM,"Mozilla/5.0 (iPod; U; CPU iPhone OS 4_1 like Mac OS X; sl-SI) AppleWebKit/531.
8B116 Safari/6531.41.4","Strickland, Michael and Gonzales",180036417827355,02/17,754,Voyager,ustewart@hotmail.com,"
"349 Laurie Parks
Thomasview, ID 08970",30 kK,PM,Mozilla/5.0 (X11; Linux i686; rv:1.9.6.20) Gecko/2014-05-12 06:09:34 Firefox/3.6.9,K
digit,johnnymiller@coleman.com,Diagnostic radiographer,128.222.40.234,en,19.26
"733 Heather Rest Apt. 670
Boltonport, UT 78662",69 DO,AM,"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_5_3 rv:2.0; sl-SI) AppleWebKit/532.38.2 (
532.38.2",Moore-Martin,5115990487067905,05/26,119,VISA 16 digit,tholt@hotmail.com,"Surveyor, quantity",236.71.234.2
"118 Melton Via Suite 681
Alexanderbury, FL 32104",36 bu,PM,"Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_8_6 rv:5.0; en-US) AppleWebKit/531.
531.44.5",Keller PLC,4603635169938574,01/25,557,VISA 16 digit,caitlin57@yahoo.com,"Accountant, chartered public fin
"8774 Jason Keys Suite 427
East Scottborough, MS 29934",70 zH,AM,Mozilla/5.0 (Windows 98; it-IT; rv:1.9.2.20) Gecko/2013-01-15 08:05:10 Firefo
Ferguson",869967499275071,09/22,427,VISA 16 digit,aburns@yahoo.com,Acupuncturist,50.25.148.1,de,24.18
"31730 Chelsea Crest
Blakemouth, CT 90395-0620",41 Cj,PM,Opera/8.95.(Windows NT 5.0; en-US) Presto/2.9.164 Version/11.00,Garcia-Steele,1
Blanche,amanda39@yahoo.com,Retail manager,53.176.235.33,el,71.78
```

Read from Excel

In[ ]:
```
pd.read_excel('Excel_Sample.xlsx',
    sheetname='Sheet1')

```

Read fr



	A	B	C	D	E
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa
8	5	3.4	1.5	0.2	Iris-setosa
9	4.4	2.9	1.4	0.2	Iris-setosa
10	4.9	3.1	1.5	0.1	Iris-setosa
11	5.4	3.7	1.5	0.2	Iris-setosa

Write to Excel

In[ ]:
```
pd.to_excel('Excel_Sample.xlsx',
    sheetname='Sheet1')

```

Concat, Joins and Merge

Left df

In[ ]:

```
left = pd.DataFrame({
        'key' : ['K0','K1','K2','K3'],
         'A'  : ['A0','A1','A2','A3'],
         'B'  : ['B0','B1','B2','B3']
         })
```

Left df

Out[]:

	key	A	B
0	K0	A0	B0
1	K1	A1	B1
2	K2	A2	B2
3	K3	A3	B3

Right df

In[ ]:

```
right = pd.DataFrame({
        'key' : ['K0','K1','K2','K3'],
         'C'  : ['C0','C1','C2','C3'],
         'D'  : ['D0','D1','D2','D3']
         })
```

Right df

Out[]:

	key	C	D
0	K0	C0	D0
1	K1	C1	D1
2	K2	C2	D2
3	K3	C3	D3

Concat

In[ ]:
```
1 | pd.concat(left, right, axis = 1)
2 |
```

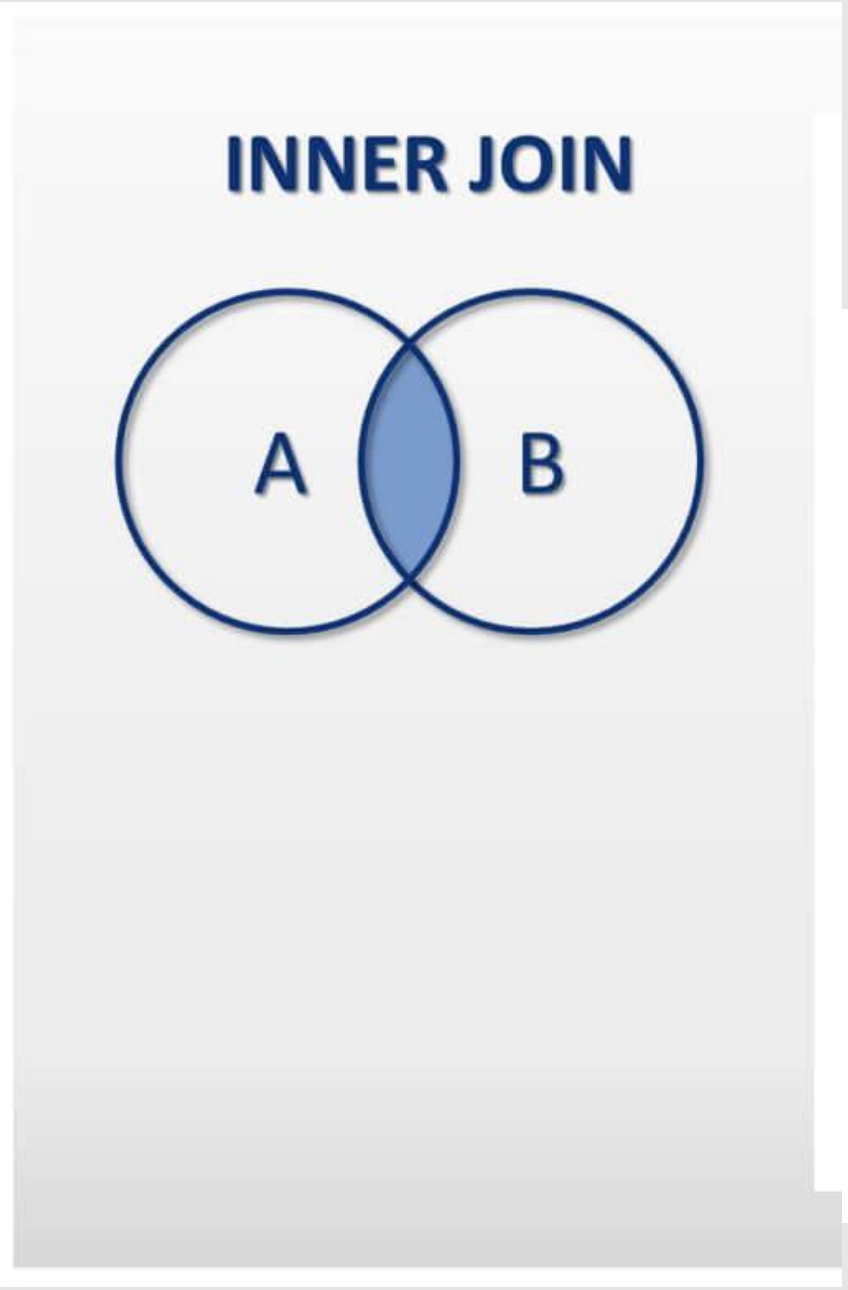
Concat

In[ ]:
```
1 | pd.concat(left, right, axis = 1)
2 |
```

Out[]:

	key	A	B	key	C	D
0	K0	A0	B0	K0	C0	D0
1	K1	A1	B1	K1	C1	D1
2	K2	A2	B2	K2	C2	D2
3	K3	A3	B3	K3	C3	D3

SQL Joins



Inner ‘merge’

In[ ]:
```
1 | pd.merge(left, right, how = 'inner',
2 |          on='key')
```

Inner ‘merge’

In[ ]:
```
1 | pd.merge(left, right, how = 'inner',
2 |                  on='key')
```

Out[]:

	key	A	B	C	D
0	K0	A0	B0	C0	D0
1	K1	A1	B1	C1	D1
2	K2	A2	B2	C2	D2
3	K3	A3	B3	C3	D3

Left df

In[ ]:

```
left = pd.DataFrame({
        'key1': ['K0','K0','K1','K2'],
        'key2': ['K0','K1','K0','K1'],
         'A'  : ['A0','A1','A2','A3'],
         'B'  : ['B0','B1','B2','B3']
         })
```

Right df

In[ ]:

```
right = pd.DataFrame({
        'key1': ['K0','K1','K1','K2'],
        'key2': ['K0','K0','K0','K0'],
         'C'  : ['C0','C1','C2','C3'],
         'D'  : ['D0','D1','D2','D3']
         })
```

Inner ‘merge’

In[ ]:
```
1 | pd.merge(left, right, on=['key1','key2'])
2 |
```

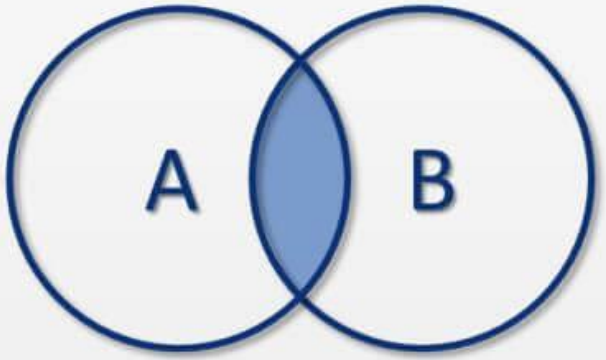
Inner ‘merge’

In[ ]:
```
1 | pd.merge(left, right, on=['key1','key2'])
2 |
```

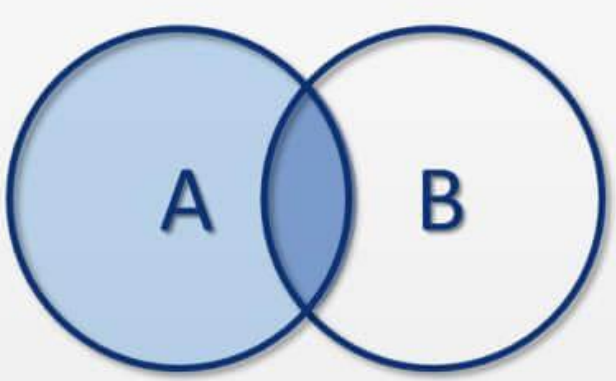
Out[]:

	key1	key2	A	B	C	D
0	K0	K0	A0	B0	C0	D0
1	K1	K0	A1	B1	C1	D1
2	K1	K0	A2	B2	C2	D2

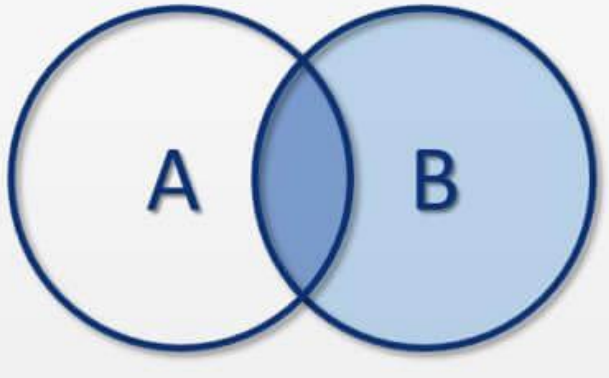
INNER JOIN



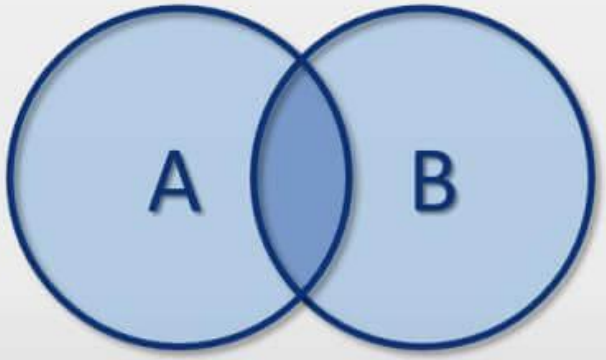
OUTER JOIN



LEFT



RIGHT



FULL

Outer ‘merge’

In[ ]:
```
pd.merge(left, right, how = 'outer',
                 on=['key1','key2'])
```

Outer ‘merge’

In[ ]:
```
1 | pd.merge(left, right, how = 'outer',
2 |                  on=['key1','key2'])
```

Out[]:

	key1	key2	A	B	C	D
0	K0	K0	A0	B0	C0	D0
1	K0	K1	A1	B1	NaN	NaN
2	K1	K0	A2	B2	C1	D1
3	K1	K0	A2	B2	C2	D2
4	K2	K1	A3	B3	NaN	NaN
5	K2	K0	NaN	NaN	C3	D3

Right ‘merge’

In[ ]:
```
pd.merge(left, right, how = 'right',
                 on=['key1','key2'])
```

Right ‘merge’

In[ ]:
```
1 | pd.merge(left, right, how = 'right',
2 |                  on=['key1','key2'])
```

Out[]:

	key1	key2	A	B	C	D
0	K0	K0	A0	B0	C0	D0
1	K1	K0	A2	B2	C1	D1
2	K1	K0	A2	B2	C2	D2
3	K2	K0	NaN	NaN	C3	D3

Right ‘merge’

In[ ]:
```
1 | pd.merge(left, right, how = 'left',
2 |                  on=['key1','key2'])
```

Right ‘merge’

In[ ]:
```
1 | pd.merge(left, right, how = 'left',
2 |                  on=['key1','key2'])
```

Out[]:

	key1	key2	A	B	C	D
0	K0	K0	A0	B0	C0	D0
1	K0	K1	A1	B1	NaN	NaN
2	K1	K0	A2	B2	C1	D1
3	K1	K0	A2	B2	C2	D2
4	K2	K1	A3	B3	NaN	NaN

Right ‘join’

In[ ]:
```
1 | left.join(right)
2 |
```

Left df

In[ ]:

```
left = pd.DataFrame({
          'A'   : ['A0','A1','A2','A3'],
          'B'   : ['B0','B1','B2','B3']},
         index = ['K0','K1','K2','K3']
         )
```

Right df

In[ ]:

```
right = pd.DataFrame({
          'C'   : ['C0','C1','C2','C3'],
          'D'   : ['D0','D1','D2','D3']},
         index = ['K0','K1','K2','K3']
         )
```

Right ‘join’

In[ ]:
```
1 | left.join(right)
2 |
```

Right ‘join’

In[ ]:
```
1 | left.join(right)
2 |
```

Out[]:

	A	B	C	D
K0	A0	B0	C0	D0
K1	A1	B1	C1	D1
K2	A2	B2	C2	D2
K3	A3	B3	C3	D3