IMPORTING & MANAGING FINANCIAL DATA IN PYTHON

Read, inspect, & clean data from csv files

Import & clean data

- Ensure that `pd.DataFrame()` is same as csv source file
- Stock exchange listings: `amex-listings.csv`

	A	B	C	D	E	F	G	H
1	Stock Symbo	Company Name	Last Sale	Market Capitalization	IPO Year	Sector	Industry	Last Update
2	XXII	22nd Century Group, Inc	1.33	120628490.3	n/a	Consumer N	Farming/See	4/24/17
3	FAX	Aberdeen Asia-Pacific Income Fund Inc	5	1266332595	1986	n/a	n/a	4/24/17
4	IAF	Aberdeen Australia Equity Fund Inc	6.15	139865304.9	n/a	n/a	n/a	4/24/17
5	CH	Aberdeen Chile Fund, Inc.	7.2201	67563457.57	n/a	n/a	n/a	4/24/17
6	ABE	Aberdeen Emerging Markets Smaller Company Opportunities Fund I	13.36	128842971.6	n/a	n/a	n/a	4/24/17
7	FCO	Aberdeen Global Income Fund, Inc.	8.62	75376107.36	1992	n/a	n/a	4/24/17
8	IF	Aberdeen Indonesia Fund, Inc.	7.3299	68200145.64	1990	n/a	n/a	4/24/17
9	ISL	Aberdeen Israel Fund, Inc.	17.65	70564682.35	1992	n/a	n/a	4/24/17
10	ACU	Acme United Corporation.	27.39	91138992.45	1988	Capital Good	Industrial Ma	4/24/17
11	AIII	ACRE Realty Investors, Inc.	1.16	23768939.4	n/a	Consumer Se	Real Estate In	4/24/17
12	ATNM	Actinium Pharmaceuticals, Inc.	1.47	82037380.74	n/a	Health Care	Major Pharm	4/24/17
13	AE	Adams Resources & Energy, Inc.	37.8	159425128.8	n/a	Energy	Oil Refining/I	4/24/17
14	ADK	Adcare Health Systems Inc	1.06	21122620	n/a	Health Care	Hospital/Nur	4/24/17
15	ADK^A	Adcare Health Systems Inc	21.946	0	n/a	n/a	n/a	4/24/17

Source: http://www.nasdaq.com/screening/companies-by-industry.aspx?exchange=AMEX

How pandas stores data

- Each column has its own data format stored in `dtype`
- `dtype` affects calculation and visualization

pandas dtype	Column characteristics
`object`	Text *or* a mix of text and numeric data
`int64`	Numeric: Whole numbers - 64 bits ($\leq 2^{64}$)
`float64`	Numeric: Decimals *or* whole numbers with missing values
`datetime64`	Date and time information

Import & inspect data

```
In [1]: import pandas as pd

In [2]: amex = pd.read_csv('amex-listings.csv')

In [3]: amex.info() # To inspect table structure & data types

 RangeIndex: 360 entries, 0 to 359
 Data columns (total 8 columns):
 Stock Symbol              360 non-null object
 Company Name              360 non-null object
 Last Sale                 360 non-null object
 Market Capitalization     360 non-null float64
 IPO Year                  360 non-null object
 Sector                    360 non-null object
 Industry                  360 non-null object
 Last Update               360 non-null object
 dtypes: float64(1), object(7)
 memory usage: 22.6+ KB
```

Deal with missing values

```
In [7]: amex = pd.read_csv('amex-listings.csv', na_values='n/a')

In [8]: amex.info()

 RangeIndex: 360 entries, 0 to 359
 Data columns (total 8 columns):
 Stock Symbol              360 non-null object
 Company Name              360 non-null object
 Last Sale                 346 non-null float64
 Market Capitalization     360 non-null float64
 IPO Year                  105 non-null float64
 Sector                    238 non-null object
 Industry                  238 non-null object
 Last Update               360 non-null object
 dtypes: float64(3), object(5)
```

np.nan:
NumPy Not a Number

Properly parse dates

```
In [7]: amex = pd.read_csv('amex-listings.csv',
        na_values='n/a',
        parse_dates=['Last Update'])

In [8]: amex.info()

 RangeIndex: 360 entries, 0 to 359
 Data columns (total 8 columns):
 Stock Symbol              360 non-null object
 Company Name              360 non-null object
 Last Sale                 346 non-null float64
 Market Capitalization     360 non-null float64
 IPO Year                  105 non-null float64
 Sector                    238 non-null object
 Industry                  238 non-null object
 Last Update               360 non-null datetime64[ns]
 dtypes: datetime64[ns](1) float64(3), object(4)
```

Show off the result

```
In [7]: amex.head() # Show first n rows (default: 5)
```

```
Out[7]:
     Stock Symbol                                        Company Name  \
0            XXII                                 22nd Century Group, Inc
1             FAX                    Aberdeen Asia-Pacific Income Fund Inc
2             IAF                       Aberdeen Australia Equity Fund Inc
3              CH                                Aberdeen Chile Fund, Inc.
4  ABE                 Aberdeen Emerging Markets Smaller Company Oppo...

   Last Sale  Market Capitalization  IPO Year                  Sector  \
0     1.3300           1.206285e+08       NaN  Consumer Non-Durables
1     5.0000           1.266333e+09    1986.0                     NaN
2     6.1500           1.398653e+08       NaN                     NaN
3     7.2201           6.756346e+07       NaN                     NaN
4    13.3600           1.288430e+08       NaN                     NaN

                 Industry Last Update
0  Farming/Seeds/Milling  2017-04-26
1                    NaN  2017-04-25
2                    NaN  2017-04-23
3                    NaN  2017-04-26
4                    NaN  2017-04-25
```

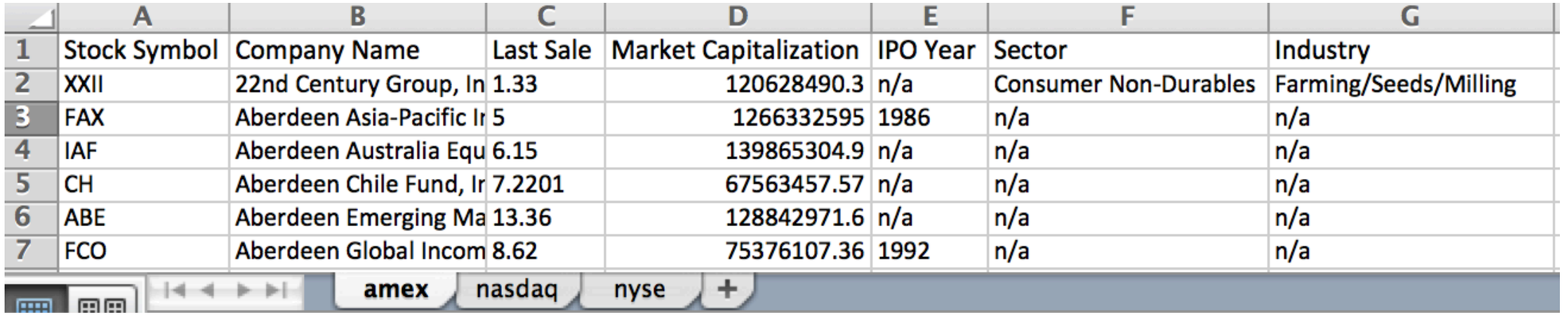
IMPORTING & MANAGING FINANCIAL DATA IN PYTHON

Let’s practice!

IMPORTING & MANAGING FINANCIAL DATA IN PYTHON

Read data from Excel worksheets

Import data from Excel



	A	B	C	D	E	F	G
1	Stock Symbol	Company Name	Last Sale	Market Capitalization	IPO Year	Sector	Industry
2	XXII	22nd Century Group, In	1.33	120628490.3	n/a	Consumer Non-Durables	Farming/Seeds/Milling
3	FAX	Aberdeen Asia-Pacific Ir	5	1266332595	1986	n/a	n/a
4	IAF	Aberdeen Australia Equ	6.15	139865304.9	n/a	n/a	n/a
5	CH	Aberdeen Chile Fund, Ir	7.2201	67563457.57	n/a	n/a	n/a
6	ABE	Aberdeen Emerging Ma	13.36	128842971.6	n/a	n/a	n/a
7	FCO	Aberdeen Global Incom	8.62	75376107.36	1992	n/a	n/a

amex nasdaq nyse +

- `pd.read_excel(file, sheetname=0)`
  - Select first sheet by default with `sheetname=0`
  - Select by name with `sheetname='amex'`
  - Import several sheets with list such as `sheetname=['amex', 'nasdaq']`

Import data from one sheet

```
In [1]: amex = pd.read_excel('listings.xlsx', sheetname='amex',
                             na_values='n/a')
```

```
In [2]: amex.info()

 RangeIndex: 360 entries, 0 to 359
 Data columns (total 8 columns):
 Stock Symbol              360 non-null object
 Company Name              360 non-null object
 Last Sale                 346 non-null float64
 Market Capitalization     360 non-null float64
 IPO Year                  105 non-null float64
 Sector                    238 non-null object
 Industry                  238 non-null object
 dtypes: datetime64[ns](1) float64(3), object(4)
```

Import data from two sheets

```
In [3]: listings = pd.read_excel('listings.xlsx',
                                 sheetname=['amex', 'nasdaq'],
                                 na_values='n/a')
```

Listings: dictionary

- **Keys: sheet names**
- **Values: DataFrame**

```
In [4]: listings['nasdaq'].info()

 RangeIndex: 3167 entries, 0 to 3166
 Data columns (total 7 columns):
 Stock Symbol              3167 non-null object
 Company Name              3167 non-null object
 Last Sale                 3165 non-null float64
 Market Capitalization     3167 non-null float64
 IPO Year                  1386 non-null float64
 Sector                    2767 non-null object
 Industry                  2767 non-null object
 dtypes: float64(3), object(4)
```

Get sheet names

```
In [5]: xls = pd.ExcelFile('listings.xlsx') # pd.ExcelFile object

In [6]: exchanges = xls.sheet_names

In [7]: exchanges
Out[7]: ['amex', 'nasdaq', 'nyse']

In [8]: nyse = pd.read_excel(xls, sheetname=exchanges[2], na_values='n/a')
```

```
In [14]: nyse.info()
RangeIndex: 3147 entries, 0 to 3146
Data columns (total 7 columns):
Stock Symbol              3147 non-null object
Company Name              3147 non-null object
…
Industry                  2177 non-null object
dtypes: float64(3), object(4)
memory usage: 172.2+ KB
```

IMPORTING & MANAGING FINANCIAL DATA IN PYTHON

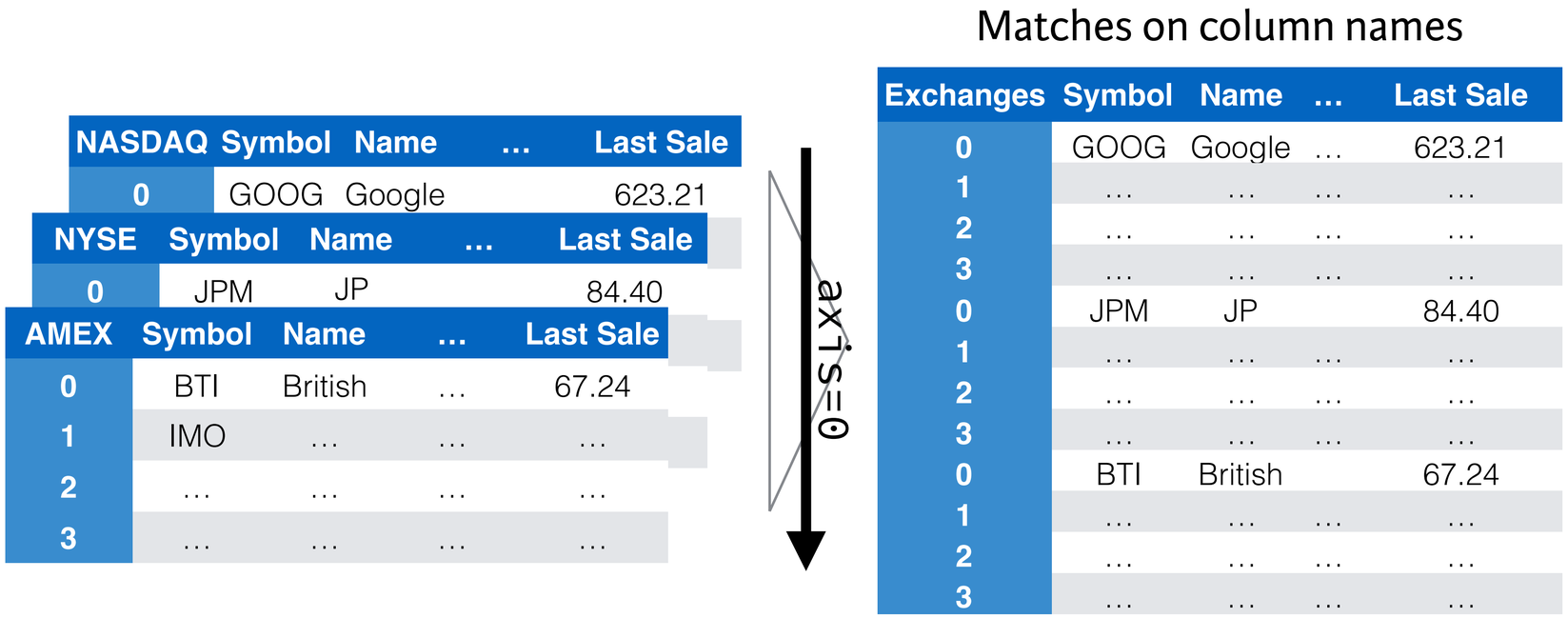
Let’s practice!

IMPORTING & MANAGING FINANCIAL DATA IN PYTHON

Combine data from multiple worksheets

Combine data frames

- **Concatenate** or "stack" a list of `pd.DataFrames`
- Syntax: `pd.concat([amex, nasdaq, nyse])`



NASDAQ	Symbol	Name	…	Last Sale
0	GOOG	Google		623.21

NYSE	Symbol	Name	…	Last Sale
0	JPM	JP		84.40

AMEX	Symbol	Name	…	Last Sale
0	BTI	British	…	67.24
1	IMO	…	…	…
2	…	…	…	…
3	…	…	…	…

Exchanges	Symbol	Name	…	Last Sale
0	GOOG	Google	…	623.21
1	…	…	…	…
2	…	…	…	…
3	…	…	…	…
0	JPM	JP		84.40
1	…	…	…	…
2	…	…	…	…
3	…	…	…	…
0	BTI	British		67.24
1	…	…	…	…
2	…	…	…	…
3	…	…	…	…

Concatenate two data frames

```
In [1]: amex = pd.read_excel('listings.xlsx', sheetname='amex',
                             na_values='n/a')

In [2]: nyse = pd.read_excel('listings.xlsx', sheetname='nyse',
                             na_values='n/a')

In [3]: pd.concat([amex, nyse]).info()
```

```
Int64Index: 3507 entries, 0 to 3146
Data columns (total 7 columns):
Stock Symbol              3507 non-null object
Company Name              3507 non-null object
Last Sale                 3425 non-null float64
Market Capitalization     3507 non-null float64
IPO Year                  1466 non-null float64
Sector                    2415 non-null object
Industry                  2415 non-null object
dtypes: float64(3), object(4)
```

Add a reference column

```
In [4]: amex['Exchange'] = 'AMEX' # Add column to reference source

In [5]: nyse['Exchange'] = 'NYSE'

In [6]: listings = pd.concat([amex, nyse])
```

```
In [7]: listings.head(2)
Out[22]:
  Stock Symbol                          Company Name  Last Sale  \
0         XXII                22nd Century Group, Inc       1.33
1          FAX  Aberdeen Asia-Pacific Income Fund Inc       5.00

   Market Capitalization  IPO Year                 Sector  \
0           1.206285e+08       NaN  Consumer Non-Durables
1           1.266333e+09    1986.0                    NaN

                Industry Exchange
0  Farming/Seeds/Milling     AMEX
1                    NaN     AMEX
```

Combine three data frames

```
In [1]: xls = pd.ExcelFile('listings.xlsx')

In [2]: exchanges = xls.sheet_names

In [3]: listings = [] # Create empty list to collect DataFrames

In [4]: for exchange in exchanges:
    ...:     listing = pd.read_excel(xls, sheetname=exchange)
    ...:     listing['Exchange'] = exchange # Add reference column
    ...:     listings.append(listing) # Add DataFrame to list

In [5]: combined_listings = pd.concat(listings) # List of DataFrames
```

Combine three data frames (2)

```
In [1]: combined_listings.info()

Int64Index: 6674 entries, 0 to 359
Data columns (total 8 columns):
Stock Symbol              6674 non-null object
Company Name              6674 non-null object
Last Sale                 6590 non-null float64
Market Capitalization     6674 non-null float64
IPO Year                  2852 non-null float64
Sector                    5182 non-null object
Industry                  5182 non-null object
Exchange                  6674 non-null object
dtypes: float64(3), object(5)
```

IMPORTING & MANAGING FINANCIAL DATA IN PYTHON

Let’s practice!