510 DATA SCIENCE

Lecture 02

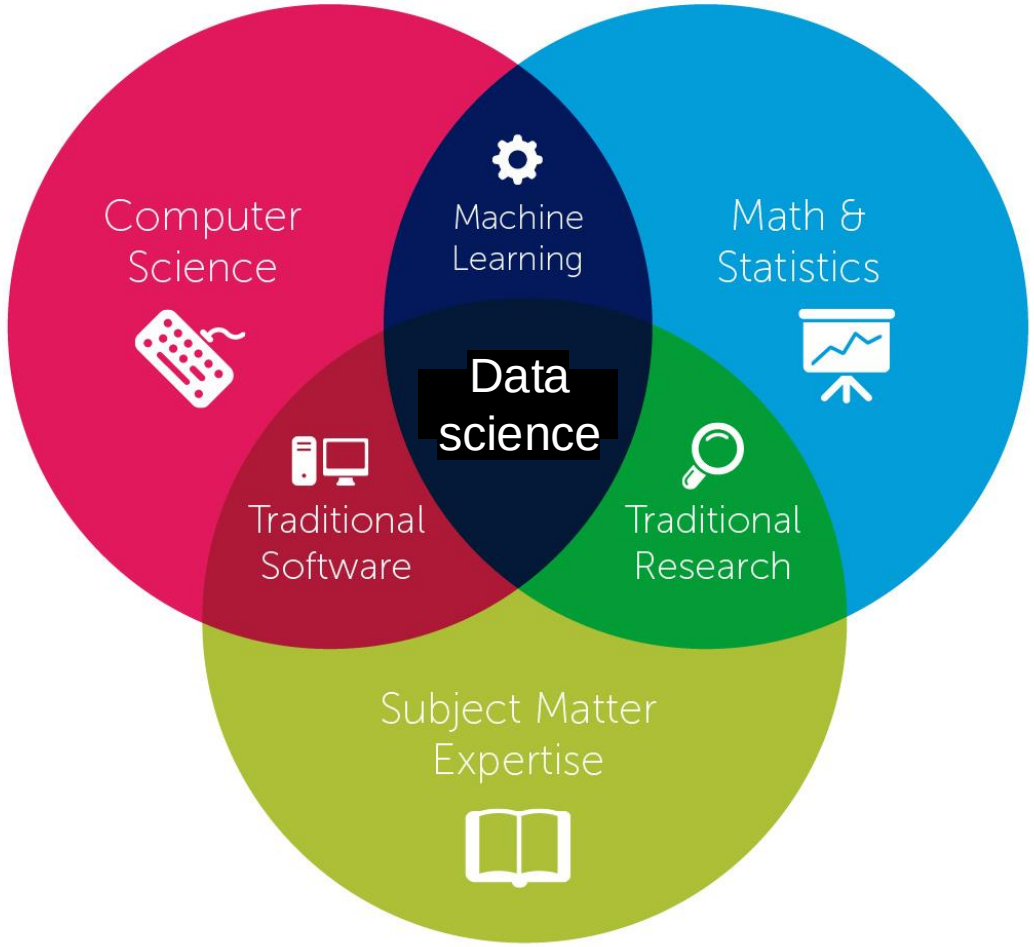
Fall 2024

Instructor: Assoc. Prof. Şener Özönder

Email: sener.ozonder@bogazici.edu.tr

Institute for Data Science & Artificial Intelligence

Boğaziçi University



Data Types in Python

- Understanding data types is crucial in data science for efficient data processing, analysis, and modeling. Data in Python can be classified into:

Numeric (int or float)
Categorical
DateTime
String (text)
Boolean (True or False)

ID (int)	Price (float)	Category (categorical)	Purchase Date (DateTime)	Product (categorical)	Description (string)	In Stock (boolean)
1	12.50	Electronics	2023-01-15 10:30:00	Mobile Phone	Features a 6.5-inch OLED display and dual cameras.	True
2	5.75	Books	2023-02-20 14:45:00	Novel	A thrilling story set in a dystopian future.	False
3	150.30	Furniture	2023-03-05 16:20:00	Dining Table	Made of solid oak wood with a polished finish.	True
4	45.00	Electronics	2023-04-10 09:15:00	Headphones	Noise-cancelling with a battery life of 20 hours.	True
5	8.90	Books	2023-05-12 12:50:00	Science Magazine	Covers the latest advancements in quantum physics.	False

Data Types in Python

- Sometimes numbers may be just categorical labels. The numbers on the balls here does NOT represent a quantifiable amount or inherent ordinality. The ball with the number “4” on it is not “greater” in any sense than the ball with the number “2”.
- To check if a label is really numerical, ask if you can change numbers with letters. You can replace “12” on the bottom right ball with “D” and nothing will change, so “12” here is categorical label. But if Ali is 12 years old, you can’t say Ali is “D” years old, so in this case “12” is age and age is truely of numerical type.

Data Structures in Python

- Data structures are containers that hold multiple data items, possibly of diverse types. We'll mostly deal with **Lists** and **Dictionaries**.
- Let's see how data types and data structures play their role in Python.

```
# Integers and Floats
int_value = 5
float_value = 5.5

print(f"Integer value: {int_value}")
print(f"Float value: {float_value}")
```

```
Integer value: 5
Float value: 5.5
```

```
# Strings
string_value = "Data Science"
print(string_value)
print(string_value.upper())
print(string_value.lower())
```

```
Data Science
DATA SCIENCE
data science
```

```
# Boolean
cat_is_animal = True
print(cat_is_animal)
```

```
True
```

```
print(type(int_value))
print(type(float_value))
print(type(string_value))
print(type(cat_is_animal))
```

```
<class 'int'>
<class 'float'>
<class 'str'>
<class 'bool'>
```

Data Structures in Python

```
# Categorical Data
import pandas as pd

# Sample data where the category column is string (not real category yet)
data = {'Category': ['A', 'B', 'A', 'C'], 'Size': [1,1,5,10]}
df = pd.DataFrame(data)
df
```

dictionary structure

dataframe holding the data in the dictionary structure

	Category	Size
0	A	1
1	B	1
2	A	5
3	C	10

Dataframes are better than arrays (matrices) since (i) dataframes can hold different data types and (ii) specific columns can be called with their name without need of knowing the column number.

```
print(df['Category'].dtype)
print(df['Size'].dtype)

# Category column is not really categorical yet.
```

```
object
int64
```

```
# Convert to Category column categorical type (this will be needed for machine learning)
df['Category'] = df['Category'].astype('category')
print(df['Category'].dtype)
```

```
category
```

Data Structures in Python

Simple lists (base Python, not pandas)

```
list_example = [1, 2, 3, 4, 5]
type(list_example)
```

```
list
```

DateTime operations

```
# DateTime operations with pandas
date_series = pd.to_datetime(pd.Series(['2023-09-15', '2023-02-10', '2023-03-05']), format='%Y-%m-%d')
print(date_series.dt.month)
```

```
0    9
1    2
2    3
dtype: int32
```

Data Structures in Python

Dictionaries:

```
my_dict = {
    "key1": ["value1a", "value1b", "value1c", "value1d"],
    "key2": ["value2a", "value2b", "value2c", "value2d"]
}
```

```
dict_example = {'Ali': 123, 'Veli': 999, 'Zeki': 444}
print(dict_example)
type(dict_example)
```

```
{'Ali': 123, 'Veli': 999, 'Zeki': 444}

dict
```

```
dict_example['Zeki']
```

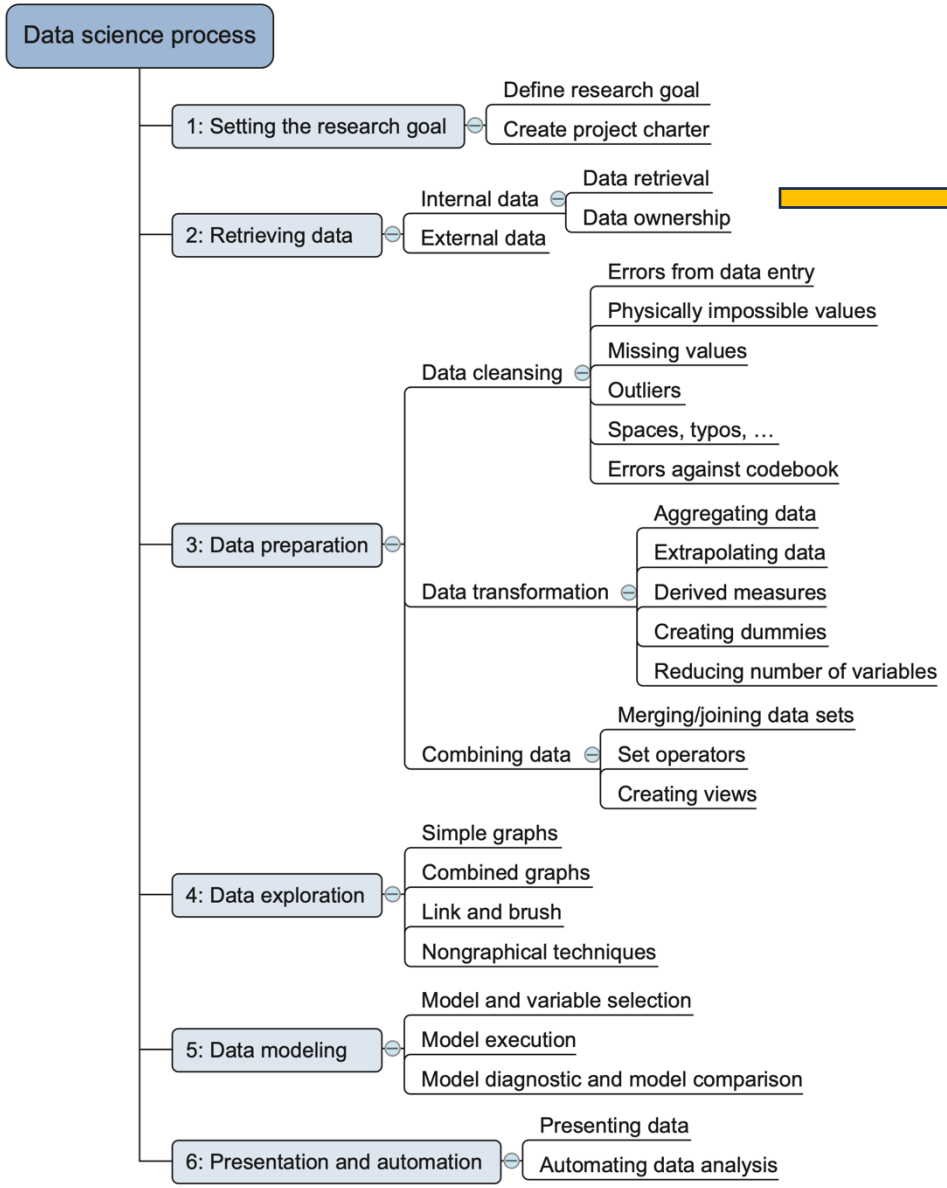
```
444
```

```
import pandas as pd

df_example = pd.DataFrame({
    'A': [1, 2, 3],
    'B': [4, 5, 6]
})
print(df_example)
```

```
   A  B
0  1  4
1  2  5
2  3  6
```

The data science process



Let's talk about the ways we can obtain data.

Retrieving Data

- Where's the data coming from? This could be online public data webpages, databases, APIs (Application Programming Interface), web scraping, third-party data providers, flat files (without hierarchical structure like txt, csv, Excel), or direct data collection through surveys and experiments.
- For learning purposes, we can start with public datasets provided by Python packages. Continue with Lecture 02.ipynb
- We can also load existing data from files in machine readable formats:

Comma-Separated Values (CSV)

Tab-Separated Values (TSV)

JavaScript Object Notation (JSON)

Extensible Markup Language (XML)

Table 6-1. Text and binary data loading functions in pandas

Function	Description
`read_csv`	Load delimited data from a file, URL, or file-like object; use comma as default delimiter
`read_fwf`	Read data in fixed-width column format (i.e., no delimiters)
`read_clipboard`	Variation of `read_csv` that reads data from the clipboard; useful for converting tables from web pages
`read_excel`	Read tabular data from an Excel XLS or XLSX file
`read_hdf`	Read HDF5 files written by pandas
`read_html`	Read all tables found in the given HTML document
`read_json`	Read data from a JSON (JavaScript Object Notation) string representation, file, URL, or file-like object

File formats: CSV and TSV

GfG - Notepad

File Edit Format View Help

```
Athlete,Age,Country,Year,Total Medals,Sport,Gold Medals,Silver Medals,Bronze Medals
Michael Phelps,23,United States,2008,8,Swimming,8,0,0
Michael Phelps,19,United States,2004,8,Swimming,6,0,2
Michael Phelps,27,United States,2012,6,Swimming,4,2,0
Natalie Coughlin,25,United States,2008,6,Swimming,1,2,3
Aleksey Nemov,24,Russia,2000,6,Gymnastics,2,1,3
Alicia Coutts,24,Australia,2012,5,Swimming,1,3,1
Missy Franklin,17,United States,2012,5,Swimming,4,0,1
Ryan Lochte,27,United States,2012,5,Swimming,2,2,1
Allison Schmitt,22,United States,2012,5,Swimming,3,1,1
Natalie Coughlin,21,United States,2004,5,Swimming,2,2,1
Ian Thorpe,17,Australia,2000,5,Sw
Dara Torres,33,United States,2000
Cindy Klassen,26,Canada,2006,5,Sp
Nastia Liukin,18,United States,20
Marit Bjørgen,29,Norway,2010,5,Cr
Sun Yang,20,China,2012,4,Swimming
Kirsty Coventry,24,Zimbabwe,2008,
Libby Lenton-Trickett,23,Australi
Ryan Lochte,24,United States,2008
Inge de Bruijn,30,Netherlands,200
```

CSV

Olympic - Notepad

File Edit Format View Help

TSV

Athlete	Age	Country	Year	Sport	Gold Medals	Silver Medals	Bronze Medals	Total Medals
Yogeshwar Dutt	29	India	2012	Wrestling	0	0	1	1
Sushil Kumar	29	India	2012	Wrestling	0	1	0	1
Sushil Kumar	25	India	2008	Wrestling	0	0	1	1
Karnam Malleswari	25	India	2000	Weightlifting	0	0	1	1
Vijay Kumar	26	India	2012	Shooting	0	1	0	1
Gagan Narang	29	India	2012	Shooting	0	0	1	1
Abhinav Bindra	25	India	2008	Shooting	1	0	0	1
Rajyavardhan Rathore	34	India	2004	Shooting	0	1	0	1
M. C. Mary Kom	29	India	2012	Boxing	0	0	1	1
Vijender Singh	22	India	2008	Boxing	0	0	1	1
Saina Nehwal	22	India	2012	Badminton	0	0	1	1

File Formats: JSON

- JSON can represent structured data. Different data records can have different keys and values.

```
{"name": "John Doe",
   "age": 30,
   "email":
"johndoe@example.com",
   "address": {
      "street": "123 Main St",
      "city": "Anytown",
      "state": "CA",
      "zip": "12345"}}
```

key (property)

value

```
{"name": "Frank Doe",
   "age": 35,
   "email": ”frankdoe@gmail.com",
   "address": {
      "state": "CA",
      "zip": ”96152",
      "country": "USA"
   },
   "occupation": "Software Engineer",
   "education": {
      "degree": "Bachelor's in Computer Science",
      "university": "University of Washington",
      "graduation_year": 2015
   },
   "hobbies": ["Reading", "Hiking", "Photography"]}
```

File Formats: XML

- XML is like html code.
- Some comparison:

XML: Doesn't have built-in support for data types. Everything is treated as text.
JSON: Has built-in support for data types like string, number, array, boolean, and null.

XML: Supports comments.
JSON: Doesn't support comments.

- Continue with Lecture 02.ipynb

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>

  <book id="1">
    <title>Harry Potter and the Philosopher's Stone</title>
    <author>J.K. Rowling</author>
    <price>19.99</price>
    <genre>Fiction</genre>
    <published>1997</published>
  </book>

  <book id="2">
    <title>The Hobbit</title>
    <author>J.R.R. Tolkien</author>
    <price>14.99</price>
    <genre>Fantasy</genre>
    <published>1937</published>
  </book>


</bookstore>
```

File Formats: PDF

- If data containing PDF is an image like scanned document and not OCR'ed, you may need to OCR it first.
- If PDF is already OCR'ed, try copying the data and pasting into Excel. Excel sometimes can recognize tabs or spaces.
- These are some Python packages to extract data

1.PyPDF2:

1. A library that can extract text and metadata from PDFs.

2.PDFMiner:

1. Specifically designed for extracting text from PDFs.
2. Can handle the layout of the PDF, making it useful for multi-column pages.
3. Can convert PDFs into other formats like HTML or XML.

3.PDFPlumber:

1. Built on top of PDFMiner, it provides tools to extract text, tables, and images.
2. Useful for extracting tabular data from PDFs.

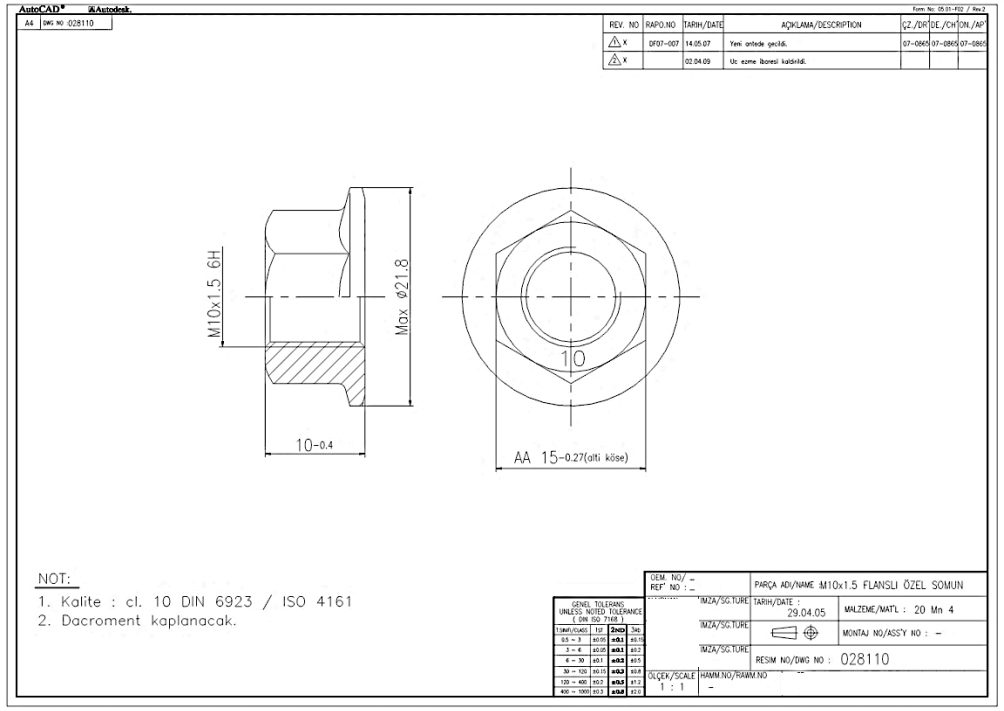
4.Slate:

1. A simpler interface for extracting text from PDFs using PDFMiner as a backend.

Google them and find which one best suits your job.

File Formats: Other formats

- There are thousands of file formats; Google it to find the best “parser” for the file format you need to work with.
- For example, DWG (DraWinG) is a vector file format for technical drawing (or engineering drawing) used by software like AutoCAD and Autodesk. Here’s how **ezdxf** Python package extracts coordinates of lines and circles in the bold given in the technical drawing.



parsing geometry data with **ezdxf**

Shape	Start_X	Start_Y	End_X	End_Y	Center_X	Center_Y	Radius
Line	641,9395	232,9393	544,712	232,9393			
Line	543,756	231,9833	642,8955	231,9833			
Line	543,756	302,089	642,8955	302,089			
Line	544,712	301,133	641,9395	301,133			
Line	543,756	302,089	543,756	231,9833			
Line	544,712	232,9393	544,712	301,133			
Line	550,9811	243,8625	546,5314	243,8625			
Line	550,9811	243,7827	546,5314	243,7827			
Line	554,2719	299,5971	544,712	299,5971			
Line	547,102	299,5971	547,102	301,133			
Line	554,2719	299,5971	554,2719	301,133			
Line	642,8955	302,089	642,8955	231,9833			
Line	641,9395	301,133	641,9395	232,9393			
Line	641,9395	235,5204	607,5986	235,5204			
Line	641,9395	242,9771	607,5239	242,9771			
Line	641,9395	238,1016	607,5239	238,1016			
Line	641,9395	240,5394	607,5239	240,5394			
Line	627,0086	242,9771	627,0086	238,1016			
Line	620,4122	238,9592	622,7403	238,6886			
Line	622,7403	239,9513	622,7403	238,6886			
Line	624,2258	240,1203	624,2258	238,5574			
Line	618,1777	245,4149	618,1777	235,5204			
Circle					624,2258	239,32	0,631342
Line	612,9828	242,9771	612,9828	232,9393			
Line	603,9643	240,1454	603,9643	232,8768			
Line	605,7814	240,1454	605,7814	232,8768			

Databases

- In real projects in the industry the dataset is usually huge so it cannot be loaded from a simple CSV file. Or sometimes the data is not flat but hierarchical and has multiple tables in relation to each other with keys. This relation cannot be retained in flat formats such as CSV. In such situations, we need a database, where the data is in special format that can be opened by a **relational database management system** (RDMS) such as PostgreSQL, MySQL, Oracle, etc.
- SQL means structured query language. For example:

```
SELECT * FROM film LIMIT 5
```

brings the 5 records from the table “film”.

	film_id	title	description	release_year	language_id	rental_duration	rental_rate	length	replacement_cost	rating	last_update	special_features	
0	133	Chamber Italian	A Fateful Reflection of a Moose And a Husband ...	2006	1	7	4.99	117	14.99	NC-17	2013-05-26 14:50:58.951	[Trailers]	'ch 'hus 'it
1	384	Grosse Wonderful	A Epic Drama of a Cat And a Explorer who must ...	2006	1	5	4.99	49	19.99	R	2013-05-26 14:50:58.951	[Behind the Scenes]	'au:
2	8	Airport Pollock	A Epic Tale of a Moose And a Girl who must Con...	2006	1	6	4.99	54	15.99	R	2013-05-26 14:50:58.951	[Trailers]	'a 'co
3	98	Bright Encounters	A Fateful Yarn of a Lumberjack And a Feminist ...	2006	1	4	4.99	73	12.99	PG-13	2013-05-26 14:50:58.951	[Trailers]	'co 'e
4	1	Academy Dinosaur	A Epic Drama of a Feminist And a Mad Scientist...	2006	1	6	0.99	86	20.99	PG	2013-05-26 14:50:58.951	[Deleted Scenes, Behind the Scenes]	'a 'can 'c

- There are also NoSQL (not only SQL) databases such as MongoDB, which accommodates SQL but also JSON or graph based databases.
- As SQL database grows, you need more powerful hardware. To handle growing NoSQL database, you need “more” servers, not necessarily one powerful one.

Databases

- Let's work on the SQL database called "dvdrental".
- Here's a DVD rental database **ER (Entity-Relationship) diagram.**
- Let's download it from the Internet and put it in a PostgreSQL management system and call it from within Python.

Continue with Lecture 02.ipynb