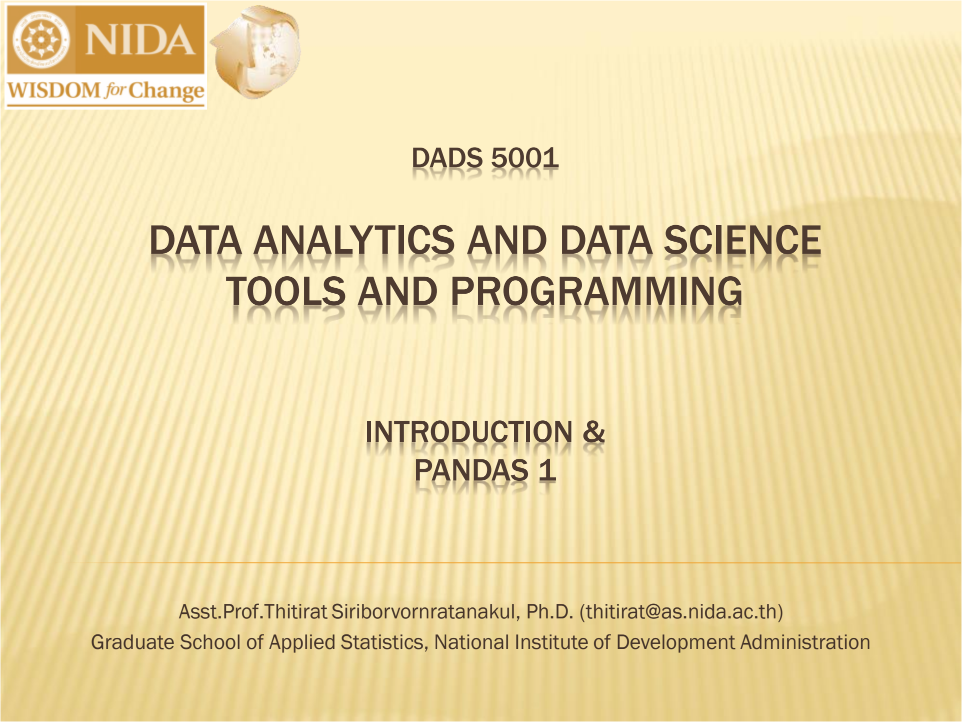


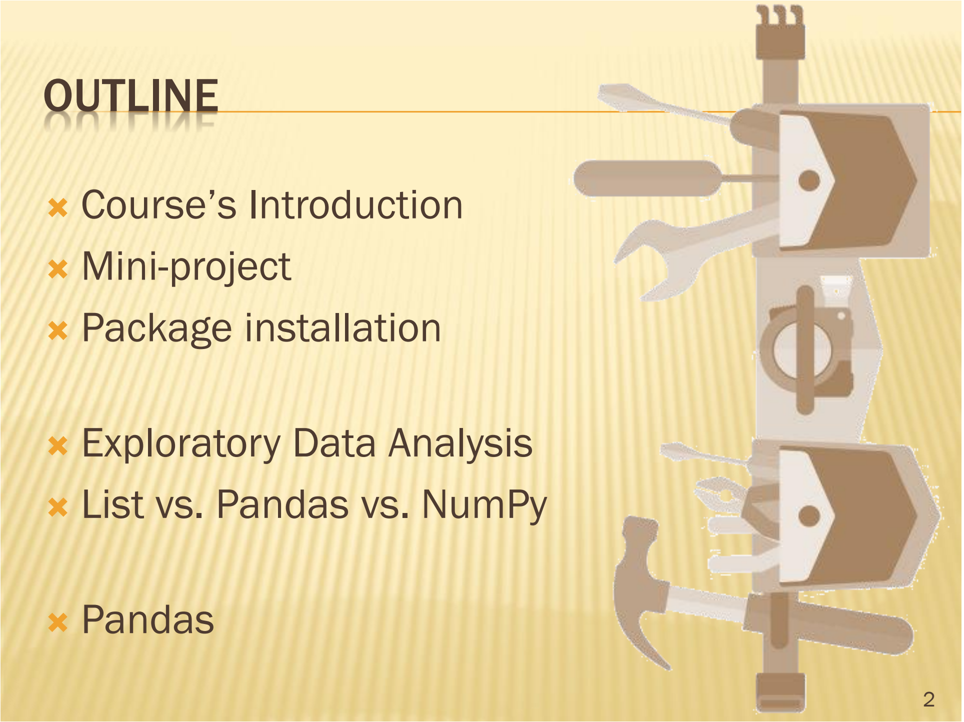
DADS 5001

DATA ANALYTICS AND DATA SCIENCE TOOLS AND PROGRAMMING

INTRODUCTION & PANDAS 1

Asst.Prof.Thitirat Siriborvornratanakul, Ph.D. (thitirat@as.nida.ac.th)
Graduate School of Applied Statistics, National Institute of Development Administration

1



OUTLINE

- Course's Introduction
- Mini-project
- Package installation

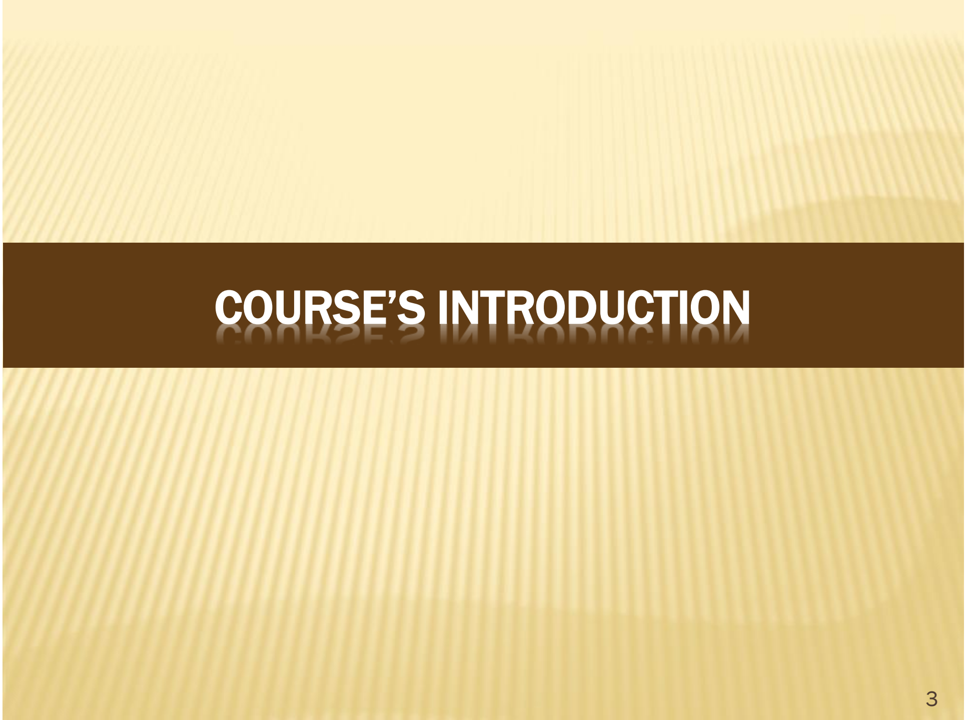
- Exploratory Data Analysis
- List vs. Pandas vs. NumPy

- Pandas

2

2

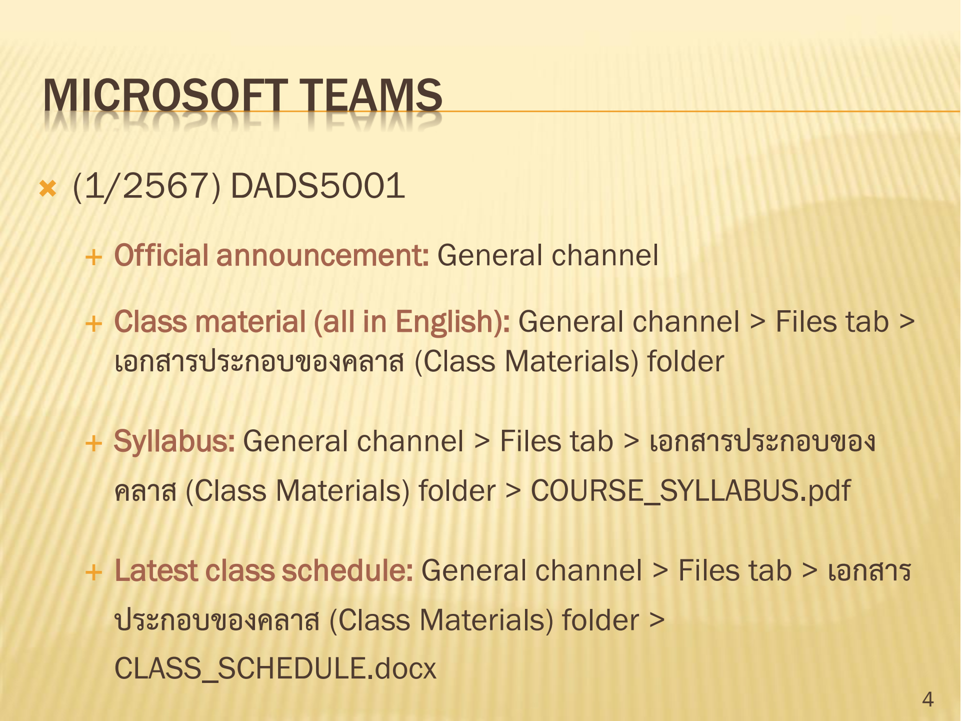
PPT template, images and decorating graphics from www.google.com unless otherwise specified.



COURSE'S INTRODUCTION

3

3

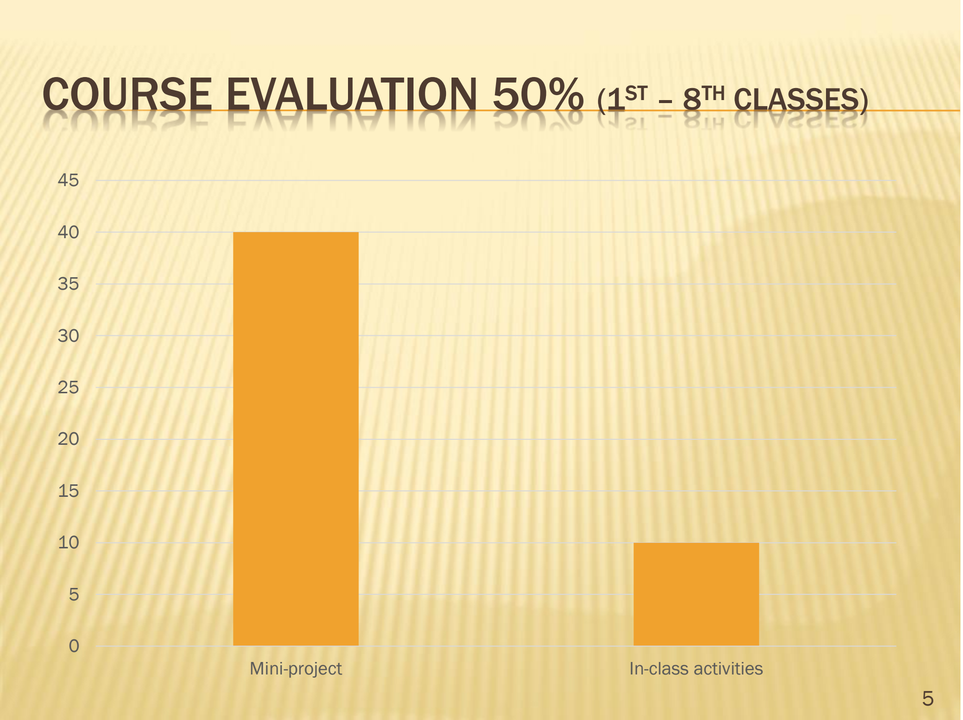


MICROSOFT TEAMS

- (1/2567) DADS5001
  - Official announcement: General channel
  - Class material (all in English): General channel > Files tab > เอกสารประกอบของคลาส (Class Materials) folder
  - Syllabus: General channel > Files tab > เอกสารประกอบของคลาส (Class Materials) folder > COURSE_SYLLABUS.pdf
  - Latest class schedule: General channel > Files tab > เอกสารประกอบของคลาส (Class Materials) folder > CLASS_SCHEDULE.docx

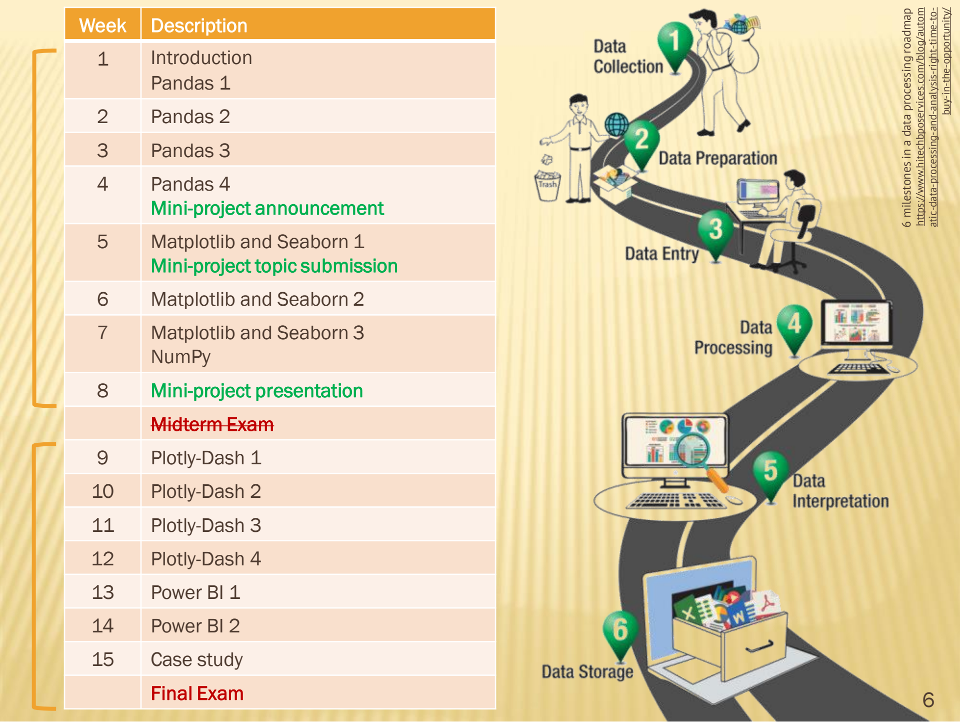
4

4



COURSE EVALUATION 50% (1ST – 8TH CLASSES)

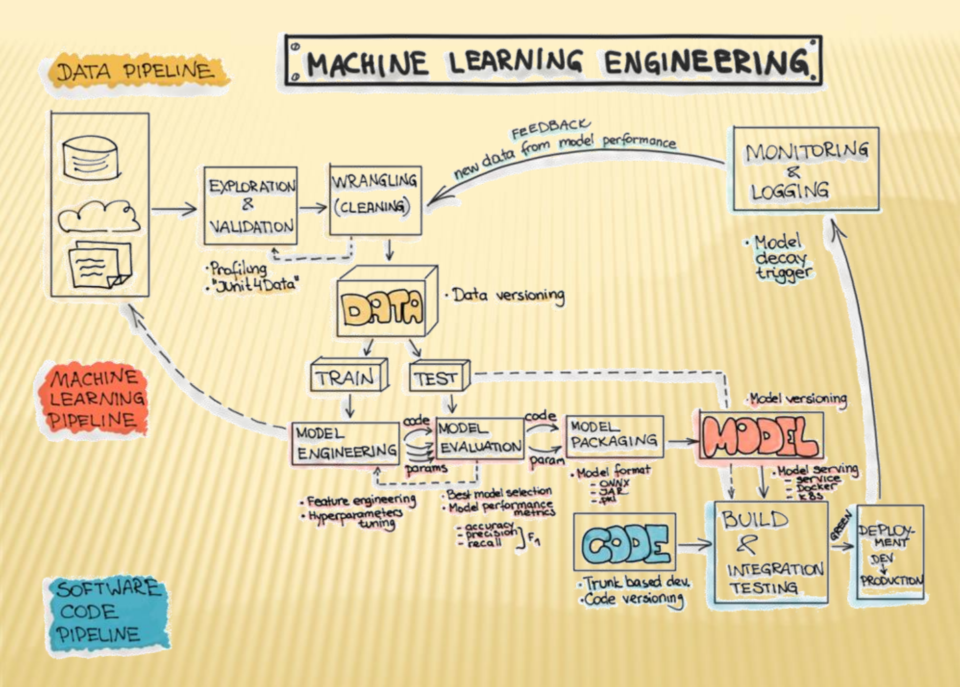
5



Week	Description
1	Introduction Pandas 1
2	Pandas 2
3	Pandas 3
4	Pandas 4 Mini-project announcement
5	Matplotlib and Seaborn 1 Mini-project topic submission
6	Matplotlib and Seaborn 2
7	Matplotlib and Seaborn 3 NumPy
8	Mini-project presentation
	~~Midterm Exam~~
9	Plotly-Dash 1
10	Plotly-Dash 2
11	Plotly-Dash 3
12	Plotly-Dash 4
13	Power BI 1
14	Power BI 2
15	Case study
	Final Exam

6 milestones in a data processing roadmap
https://www.hitechbposervices.com/blog/automatic-data-processing-and-analysis-right-time-to-buy-in-the-opportunity/

6



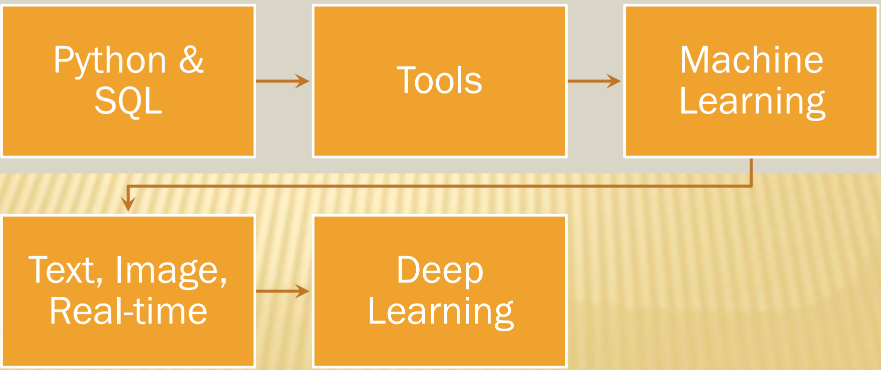
Machine Learning Engineering for Production (MLOps) https://techblog.sms-digital.com/machine-learning-engineering-for-production-mlops

7

PROGRAMMING SEQUENCE IN DADS

The fundamental five of Python's data science stacks

jupyter NumPy pandas matplotlib scikit learn



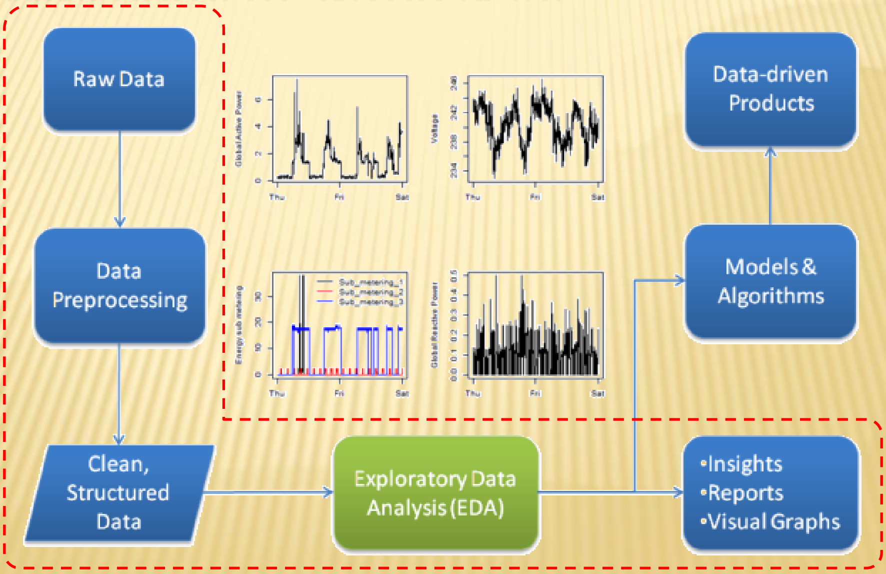
Python & SQL → Tools → Machine Learning → Text, Image, Real-time → Deep Learning

8

8

MINI-PROJECT

MINI-PROJECT: INTRO (1/3)



Raw Data

Data Preprocessing

Clean, Structured Data

Exploratory Data Analysis (EDA)

- Insights
- Reports
- Visual Graphs

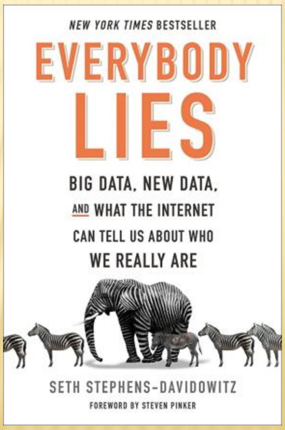
Models & Algorithms

Data-driven Products

PPT template, images and decorating graphics from www.google.com unless otherwise specified.

MINI-PROJECT: INTRO (2/3)

"Most important, to squeeze insights out of Big Data, you have to ask the right questions."



NEW YORK TIMES BESTSELLER

EVERYBODY LIES

BIG DATA, NEW DATA, AND WHAT THE INTERNET CAN TELL US ABOUT WHO WE REALLY ARE

SETH STEPHENS-DAVIDOWITZ

FOREWORD BY STEVEN PINKER

NEW YORK TIMES BESTSELLER

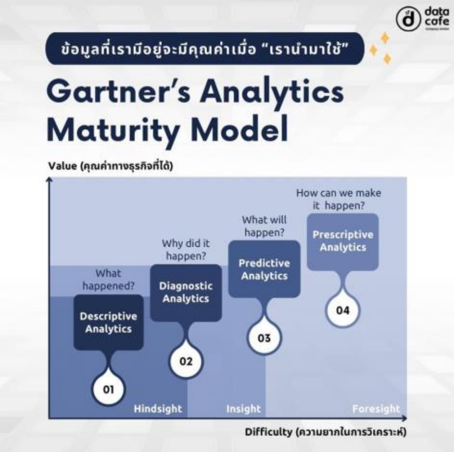
BIG DATA ดัดจริต

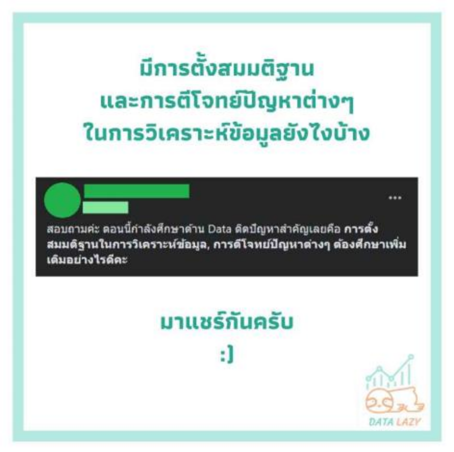
ถึงเวลาฉีกหน้ากาก เปลือยตัวตน
ด้วยข้อมูลระดับบิ๊กสุดๆ
ที่จะเปิดเผยความคิดสุดลับของมนุษย์
อย่างที่ไม่เคยเปิดเผยที่ไหนมาก่อน

เรียบเรียงจาก EVERYBODY LIES
โดย SETH STEPHENS-DAVIDOWITZ
เรียบเรียงโดย ธนกร นำรับพร
คำนิยมโดย STEVEN PINKER

ซีเอ็ด

MINI-PROJECT: INTRO (3/3)





มีการตั้งสมมติฐาน
และการตีโจทย์ปัญหาต่างๆ
ในการวิเคราะห์ข้อมูลยังไงบ้าง

สอบถามค่ะ ตอนนี้กำลังศึกษาด้าน Data ติดปัญหาสำคัญเลยคือ **การตั้งสมมติฐานในการวิเคราะห์ข้อมูล, การตีโจทย์ปัญหาต่างๆ ต้องศึกษาเพิ่มเติมอย่างไรดีคะ**

มาแชร์กันครับ
:]

DATA LAZY

https://www.facebook.com/100057174082208/posts/pfbid022YgWGJAuUTXkPXN9gho825NkL9NVc7qNnd5dhTjZg8QnBiT6ujy1K7xzNZ82bEHnl/?mibextid=cr9u03 , https://www.facebook.com/d4biz/photos/a.3233628596865997/3229272480634942

PPT template, images and decorating graphics from www.google.com unless otherwise specified.

MINI-PROJECT: HOW TO

1. Having a good question is a good start, so as an initial assumption.
2. Check data availability/accessibility (realistic and unprocessed data, not the already-prepped neat dataset with abundant tutorials/examples)
3. Prove or disprove your assumption with data analytics and data science
   + Programming tools and (static) visualization must be included significantly.
   + The data source may be shared among some students but what comes afterward (questioning, cleaning, processing, EDA, visualizing) must be done individually.
4. Submit as a public GitHub link (one link per one group), including codes **and** development journey

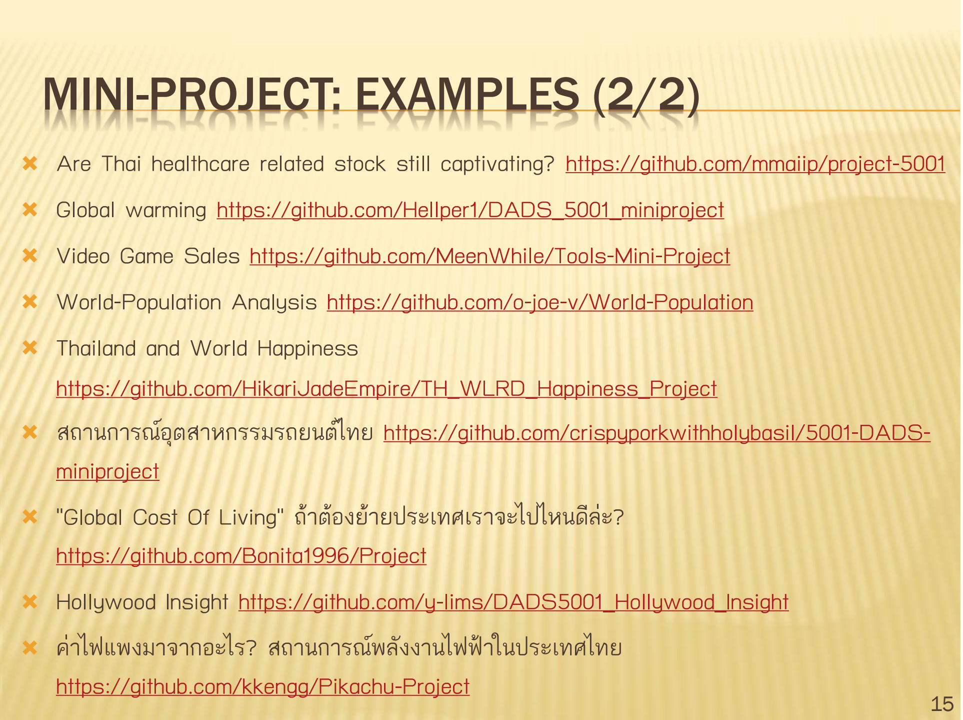
AGENDA
BiZVERSE
ลาออก
ครั้งใหญ่
The Great Resignation
VS
ไล่ออก
ครั้งใหญ่
The Great Layoff
ทำไมเกิดขึ้นพร้อมกัน?
คนลาออกถึง 47 ล้านคนในปี 2021 สูงสุดในประวัติการณ์สหรัฐฯ
ในไทยคนหยุดทำงานด้วยเหตุผล 'ลาออก' มากเป็นอันดับ 2
เกิดอะไรขึ้น
บริษัทเทค ฯ 861 ราย ประกาศเลิกจ้างแบบโดมิโน รวมกว่า 1.4 แสนคน
หลายบริษัทชะลอตัวรับพนักงานใหม่
GEN Z และ GEN Y
เกือบครึ่งของคน GEN Z และ GEN Y ยินดีสละ 10% ของเงินเดือนแลกกับการทำงานจากที่ไหนก็ได้
เกิดกับใคร
บริษัทสตาร์ทอัพ เทคโนโลยี E-Commerce ที่เติบโตสูงจนต้องจ้างคนเยอะมากในช่วงที่ผ่านมา

กรุงเทพธุรกิจ
ค่า 'อาหารกลางวัน' เพิ่ม 1 บาท
ไม่ช่วยความเหลื่อมล้ำในโรงเรียน

MINI-PROJECT: EXAMPLES (1/2)

- ส่อง Mega trend "EV Car" https://github.com/ssorawits/Project.git
- ผลกระทบจากเศรษฐกิจทำให้อัตราเด็กเกิดใหม่ลดลงหรือไม่ https://github.com/pnithida/6420422004_MiniProject
- PM2.5 in Thailand(Y2019-Y2021) https://github.com/Porrakij/PM2.5-Data-Analysis
- สำรวจกระเป๋าตังคนไทย รายได้ และ รายจ่าย เฉลี่ยของครัวเรือน https://github.com/koraweep/DADS5001_Mini-Project
- จังหวัดเฝ้าระวัง เพื่อป้องกันและลดจำนวนผู้เสียชีวิตจากอุบัติเหตุทางถนน https://github.com/ploychenya/DADS5001_mini_project
- Big Data กับโลกอสังหาริมทรัพย์ https://github.com/Hakulani/miniprojectDADS5001
- ข้อมูลมูลค่าการนำเข้าสินค้าของประเทศไทยจากทั่วโลกในช่วงปี 2002 – 2022 https://github.com/nacknatthawit/DADS5001_6420412006
- Thailand_Accident_Jan-Jun2022 https://github.com/waewma/Thailand_Accident_Jan-Jun2022
- ยอดการค้นหา Google trend ด้วย keyword "Bitkub" จะส่งผลดีต่อบริการของตัวบริษัทหรือไม่ https://github.com/kimteespk/DADS5001_Miniproject_bitkub_googletrend_6510412011
- Vending machine analysis https://github.com/kikkalo/6420412010
- Thailand Labour Demand Trend Y2015-2020 https://github.com/Sujitra17/Mini-Project

PPT template, images and decorating graphics from www.google.com unless otherwise specified.



MINI-PROJECT: EXAMPLES (2/2)

- Are Thai healthcare related stock still captivating? https://github.com/mmaiip/project-5001
- Global warming https://github.com/Hellper1/DADS_5001_miniproject
- Video Game Sales https://github.com/MeenWhile/Tools-Mini-Project
- World-Population Analysis https://github.com/o-joe-v/World-Population
- Thailand and World Happiness https://github.com/HikariJadeEmpire/TH_WLRD_Happiness_Project
- สถานการณ์อุตสาหกรรมรถยนต์ไทย https://github.com/crispyporkwithholybasil/5001-DADS-miniproject
- "Global Cost Of Living" ถ้าต้องย้ายประเทศเราจะไปไหนดีล่ะ? https://github.com/Bonita1996/Project
- Hollywood Insight https://github.com/y-lims/DADS5001_Hollywood_Insight
- ค่าไฟแพงมาจากอะไร? สถานการณ์พลังงานไฟฟ้าในประเทศไทย https://github.com/kkengg/Pikachu-Project

15

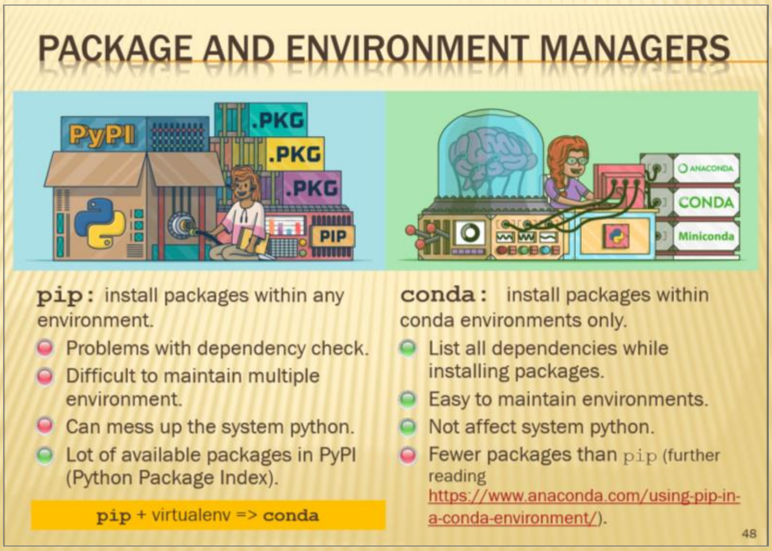
15

PACKAGE INSTALLATION

16

16

PIP INSTALL VS. CONDA INSTALL



PACKAGE AND ENVIRONMENT MANAGERS

PyPI
.PKG
.PKG
.PKG
PIP
ANACONDA
CONDA
Miniconda

`pip:` install packages within any environment.

- 🔴 Problems with dependency check.
- 🔴 Difficult to maintain multiple environment.
- 🔴 Can mess up the system python.
- 🟢 Lot of available packages in PyPI (Python Package Index).

`pip` + virtualenv => `conda`

`conda:` install packages within conda environments only.

- 🟢 List all dependencies while installing packages.
- 🟢 Easy to maintain environments.
- 🟢 Not affect system python.
- 🔴 Fewer packages than `pip` (further reading https://www.anaconda.com/using-pip-in-a-conda-environment/).

48

17

17

REQUIRED LIBRARIES

```
conda install jupyter

conda install pandas
conda install numpy

conda install matplotlib
conda install seaborn
```

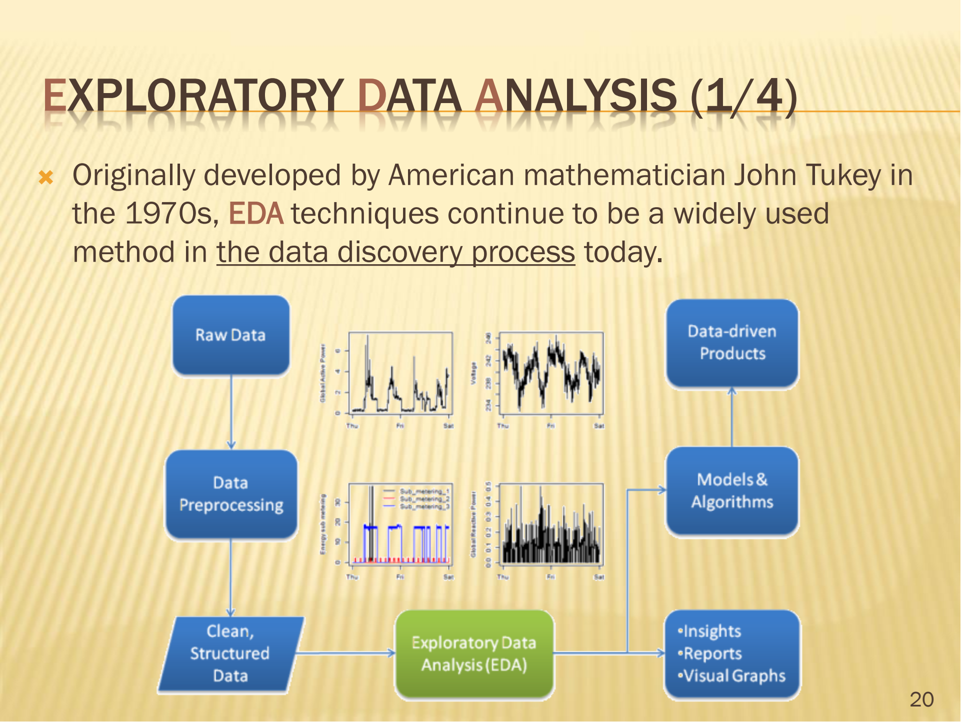
18

18

EXPLORATORY DATA ANALYSIS

19

19



EXPLORATORY DATA ANALYSIS (1/4)

- Originally developed by American mathematician John Tukey in the 1970s, EDA techniques continue to be a widely used method in the data discovery process today.

Raw Data

Data Preprocessing

Clean, Structured Data

Exploratory Data Analysis (EDA)

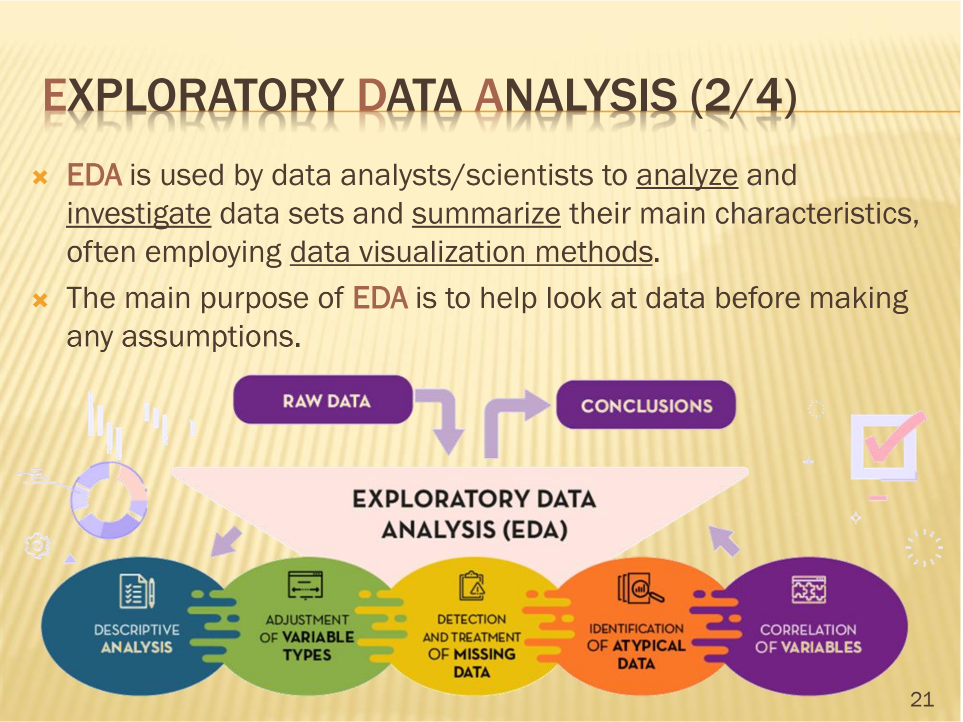
Models & Algorithms

Data-driven Products

- Insights
- Reports
- Visual Graphs

20

20



EXPLORATORY DATA ANALYSIS (2/4)

- EDA is used by data analysts/scientists to analyze and investigate data sets and summarize their main characteristics, often employing data visualization methods.
- The main purpose of EDA is to help look at data before making any assumptions.

RAW DATA

CONCLUSIONS

EXPLORATORY DATA ANALYSIS (EDA)

DESCRIPTIVE ANALYSIS

ADJUSTMENT OF VARIABLE TYPES

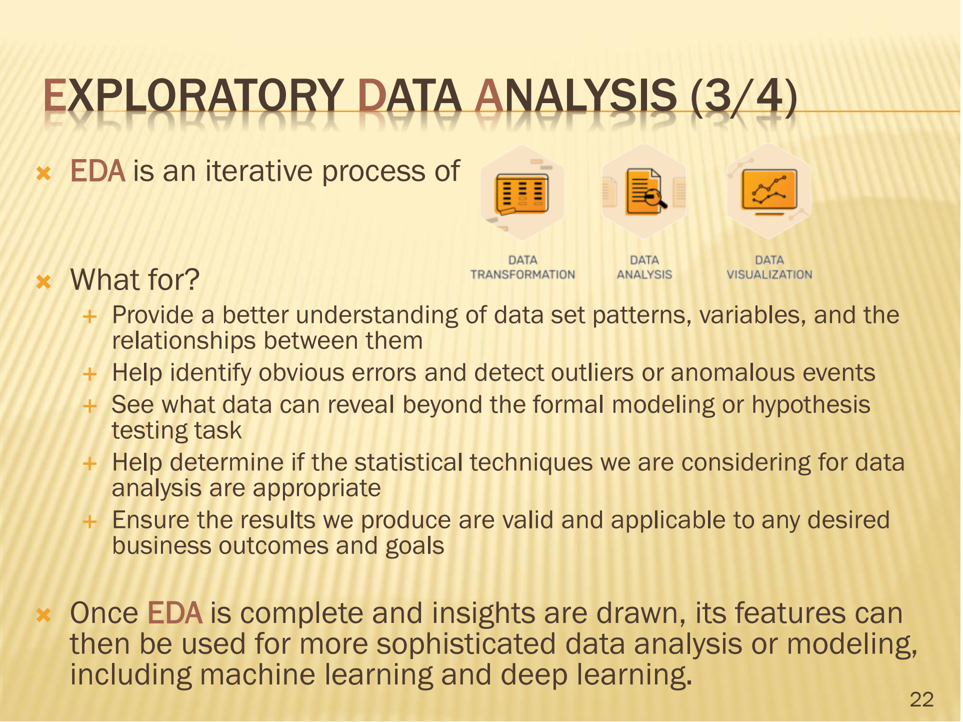
DETECTION AND TREATMENT OF MISSING DATA

IDENTIFICATION OF ATYPICAL DATA

CORRELATION OF VARIABLES

21

21



EXPLORATORY DATA ANALYSIS (3/4)

- EDA is an iterative process of

DATA TRANSFORMATION

DATA ANALYSIS

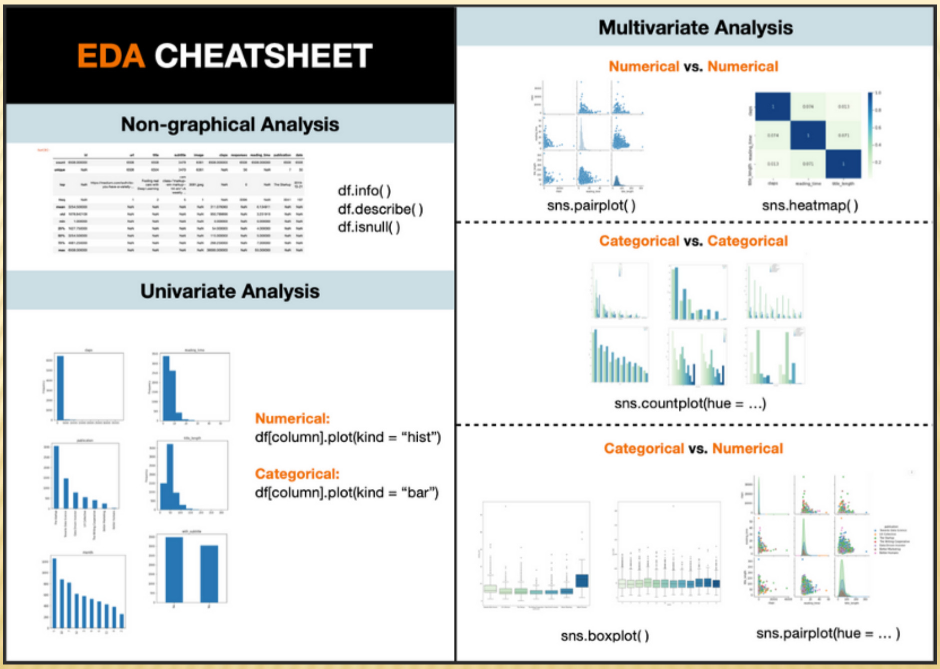
DATA VISUALIZATION

- What for?
  - Provide a better understanding of data set patterns, variables, and the relationships between them
  - Help identify obvious errors and detect outliers or anomalous events
  - See what data can reveal beyond the formal modeling or hypothesis testing task
  - Help determine if the statistical techniques we are considering for data analysis are appropriate
  - Ensure the results we produce are valid and applicable to any desired business outcomes and goals
- Once EDA is complete and insights are drawn, its features can then be used for more sophisticated data analysis or modeling, including machine learning and deep learning.

22

22

PPT template, images and decorating graphics from www.google.com unless otherwise specified.



EDA CHEATSHEET

Non-graphical Analysis

df.info( )
df.describe( )
df.isnull( )

Univariate Analysis

Numerical:
df[column].plot(kind = “hist”)

Categorical:
df[column].plot(kind = “bar”)

Multivariate Analysis

Numerical vs. Numerical

sns.pairplot( )

sns.heatmap( )

Categorical vs. Categorical

sns.countplot(hue = ...)

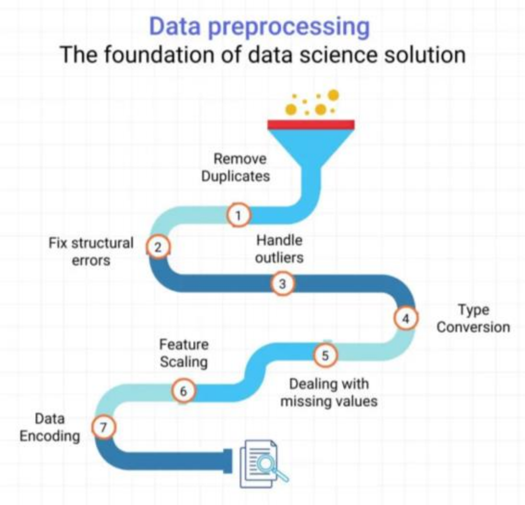
Categorical vs. Numerical

sns.boxplot( )

sns.pairplot(hue = ...)

Source: https://towardsdatascience.com/semi-automated-exploratory-data-analysis-eda-in-python-7f96042c9809

DATA PREPROCESSING (1/2)

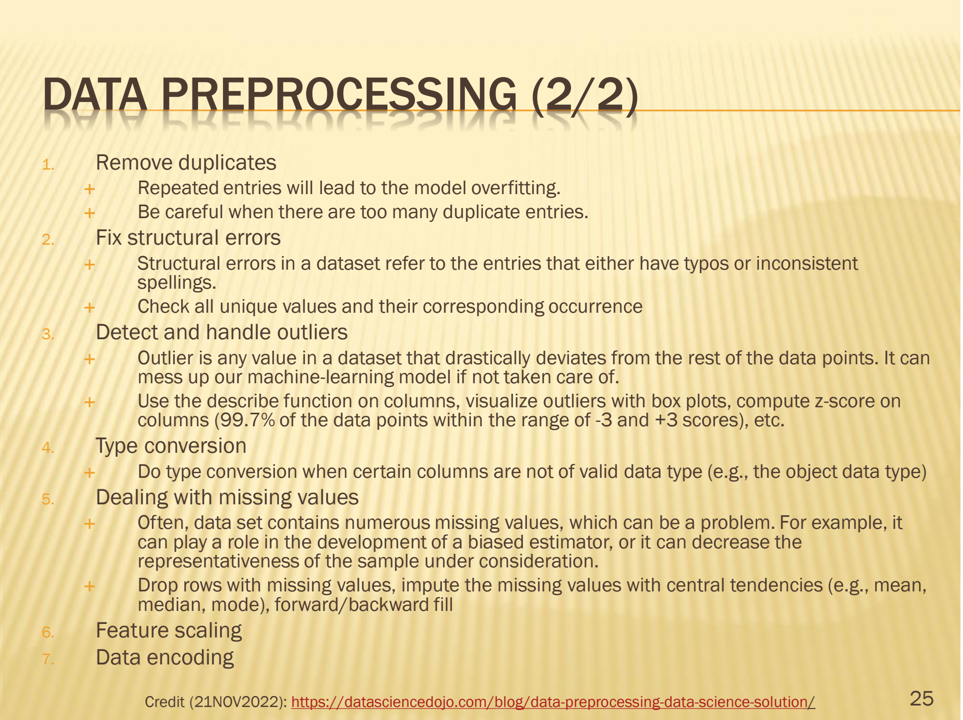


Data preprocessing
The foundation of data science solution

1 Remove Duplicates
2 Fix structural errors
3 Handle outliers
4 Type Conversion
5 Dealing with missing values
6 Feature Scaling
7 Data Encoding

- A common mistake that many novice data scientists make is that they skip through the data wrangling stage and dive right into the model-building phase, which in turn generates a poor-performing machine learning model.

21NOV2022: https://datasciencedojo.com/blog/data-preprocessing-data-science-solution/



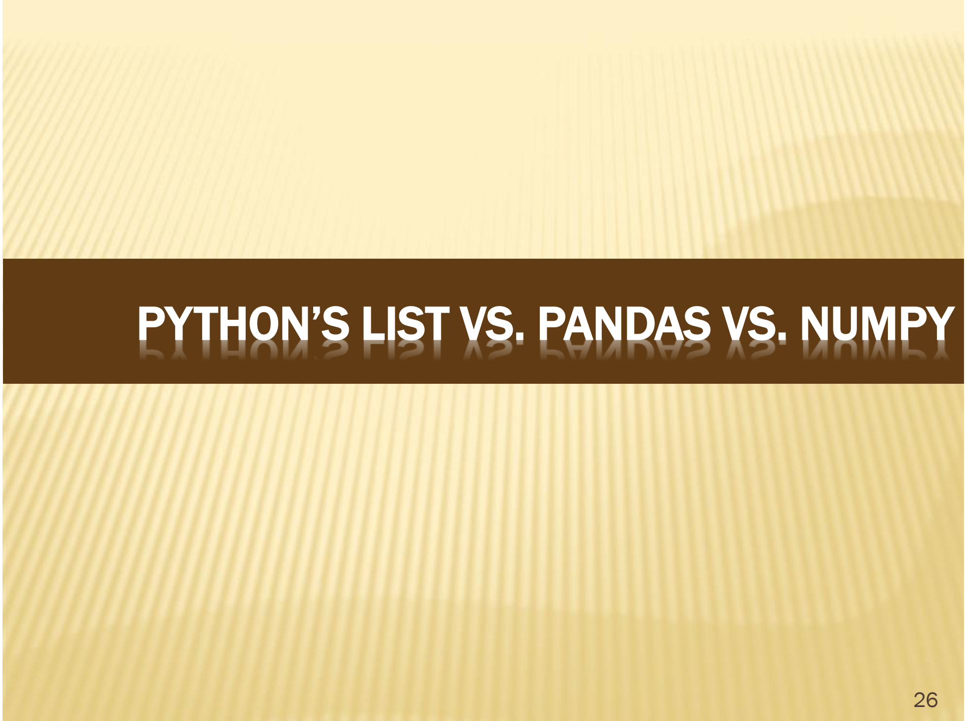
DATA PREPROCESSING (2/2)

1. Remove duplicates
   - Repeated entries will lead to the model overfitting.
   - Be careful when there are too many duplicate entries.
2. Fix structural errors
   - Structural errors in a dataset refer to the entries that either have typos or inconsistent spellings.
   - Check all unique values and their corresponding occurrence
3. Detect and handle outliers
   - Outlier is any value in a dataset that drastically deviates from the rest of the data points. It can mess up our machine-learning model if not taken care of.
   - Use the describe function on columns, visualize outliers with box plots, compute z-score on columns (99.7% of the data points within the range of -3 and +3 scores), etc.
4. Type conversion
   - Do type conversion when certain columns are not of valid data type (e.g., the object data type)
5. Dealing with missing values
   - Often, data set contains numerous missing values, which can be a problem. For example, it can play a role in the development of a biased estimator, or it can decrease the representativeness of the sample under consideration.
   - Drop rows with missing values, impute the missing values with central tendencies (e.g., mean, median, mode), forward/backward fill
6. Feature scaling
7. Data encoding

Credit (21NOV2022): https://datasciencedojo.com/blog/data-preprocessing-data-science-solution/

25

25



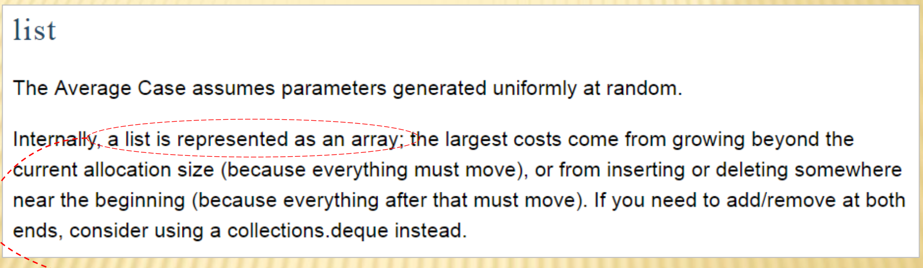
PYTHON'S LIST VS. PANDAS VS. NUMPY

26

26

PYTHON'S LIST

- https://wiki.python.org/moin/TimeComplexity



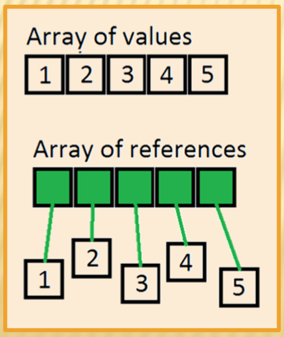
list

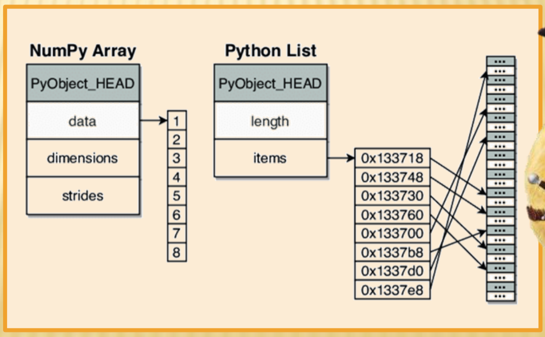
The Average Case assumes parameters generated uniformly at random.

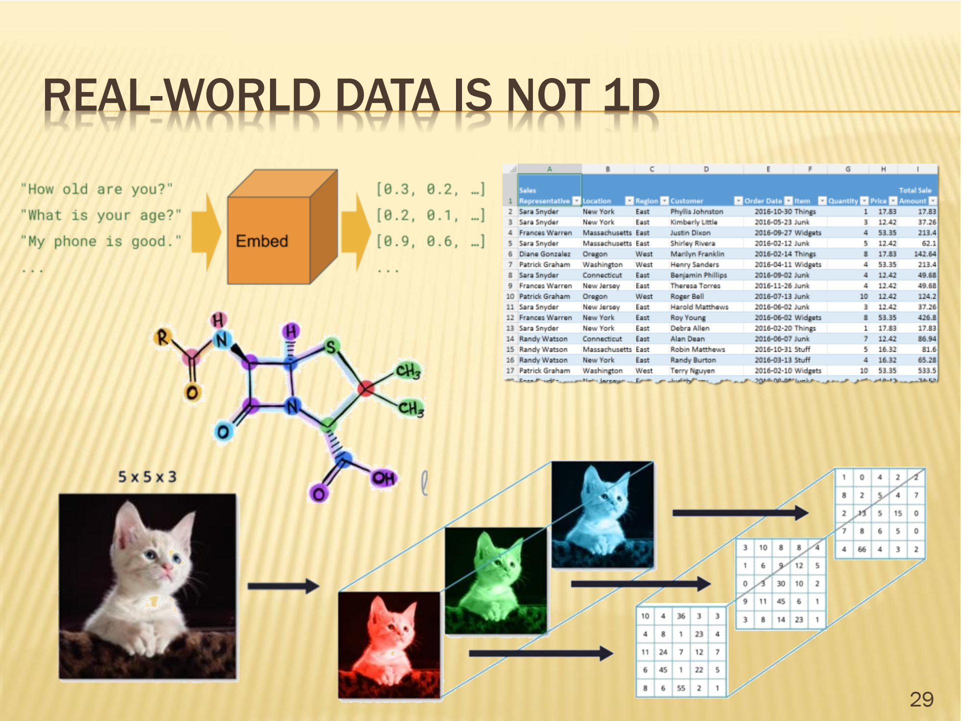
Internally, a list is represented as an array; the largest costs come from growing beyond the current allocation size (because everything must move), or from inserting or deleting somewhere near the beginning (because everything after that must move). If you need to add/remove at both ends, consider using a collections.deque instead.

ARRAY VS. PYTHON'S LIST

- In other programming languages, Array refers to the static memory allocation for holding a sequence of homogeneous data. Array is faster.
- Because Python's List stores references to data, a List can (seemingly) hold heterogeneous data. List is slower.







REAL-WORLD DATA IS NOT 1D

```
"How old are you?"
"What is your age?"
"My phone is good."
...
```

```
[0.3, 0.2, …]
[0.2, 0.1, …]
[0.9, 0.6, …]
...
```

	A	B	C	D	E	F	G	H	I
1	Sales Representative	Location	Region	Customer	Order Date	Item	Quantity	Price	Total Sale Amount
2	Sara Snyder	New York	East	Phyllis Johnston	2016-10-30	Things	1	17.83	17.83
3	Sara Snyder	New York	East	Kimberly Little	2016-05-23	Junk	3	12.42	37.26
4	Frances Warren	Massachusetts	East	Justin Dixon	2016-09-27	Widgets	4	53.35	213.4
5	Sara Snyder	Massachusetts	East	Shirley Rivera	2016-02-12	Junk	5	12.42	62.1
6	Diane Gonzalez	Oregon	West	Marilyn Franklin	2016-02-14	Things	8	17.83	142.64
7	Patrick Graham	Washington	West	Henry Sanders	2016-04-11	Widgets	4	53.35	213.4
8	Sara Snyder	Connecticut	East	Benjamin Phillips	2016-09-02	Junk	4	12.42	49.68
9	Frances Warren	New Jersey	East	Theresa Torres	2016-11-26	Junk	4	12.42	49.68
10	Patrick Graham	Oregon	West	Roger Bell	2016-07-13	Junk	10	12.42	124.2
11	Sara Snyder	New Jersey	East	Harold Matthews	2016-06-02	Junk	3	12.42	37.26
12	Frances Warren	New York	East	Roy Young	2016-06-02	Widgets	8	53.35	426.8
13	Sara Snyder	New York	East	Debra Allen	2016-02-20	Things	1	17.83	17.83
14	Randy Watson	Connecticut	East	Alan Dean	2016-06-07	Junk	7	12.42	86.94
15	Randy Watson	Massachusetts	East	Robin Matthews	2016-10-31	Stuff	5	16.32	81.6
16	Randy Watson	New York	East	Randy Burton	2016-03-13	Stuff	4	16.32	65.28
17	Patrick Graham	Washington	West	Terry Nguyen	2016-02-10	Widgets	10	53.35	533.5

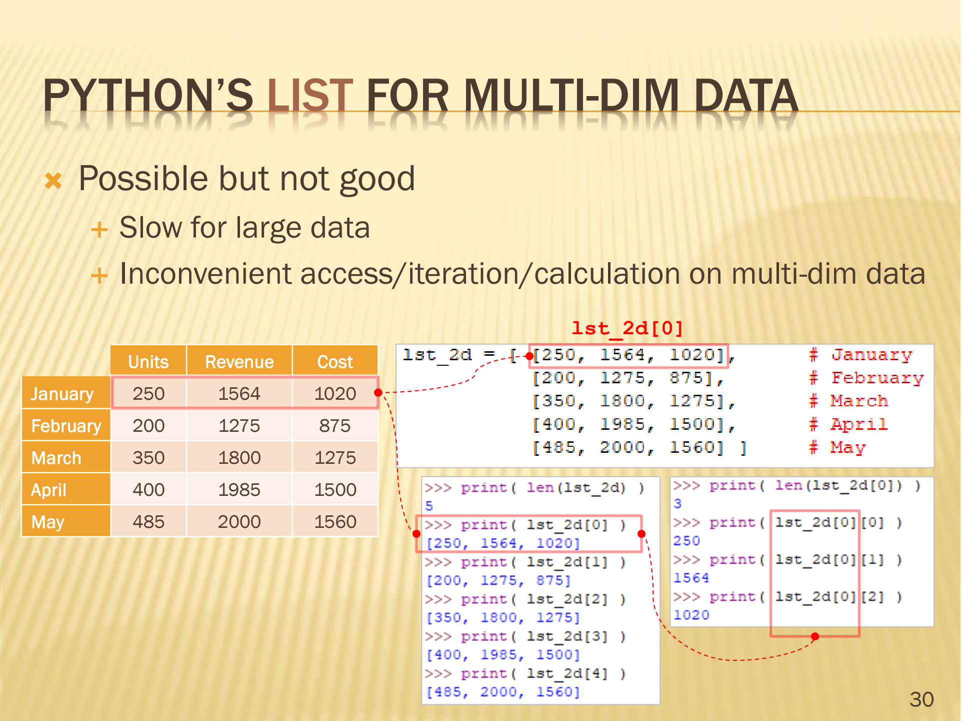
5 x 5 x 3

10	4	36	3	3
4	8	1	23	4
11	24	7	12	7
6	45	1	22	5
8	6	55	2	1

3	10	8	8	4
1	6	9	12	5
0	3	30	10	2
9	11	45	6	1
3	8	14	23	1

1	0	4	2	2
8	2	5	4	7
2	13	5	15	0
7	8	6	5	0
4	66	4	3	2

29



PYTHON'S LIST FOR MULTI-DIM DATA

- Possible but not good
  - Slow for large data
  - Inconvenient access/iteration/calculation on multi-dim data

	Units	Revenue	Cost
January	250	1564	1020
February	200	1275	875
March	350	1800	1275
April	400	1985	1500
May	485	2000	1560

lst_2d[0]

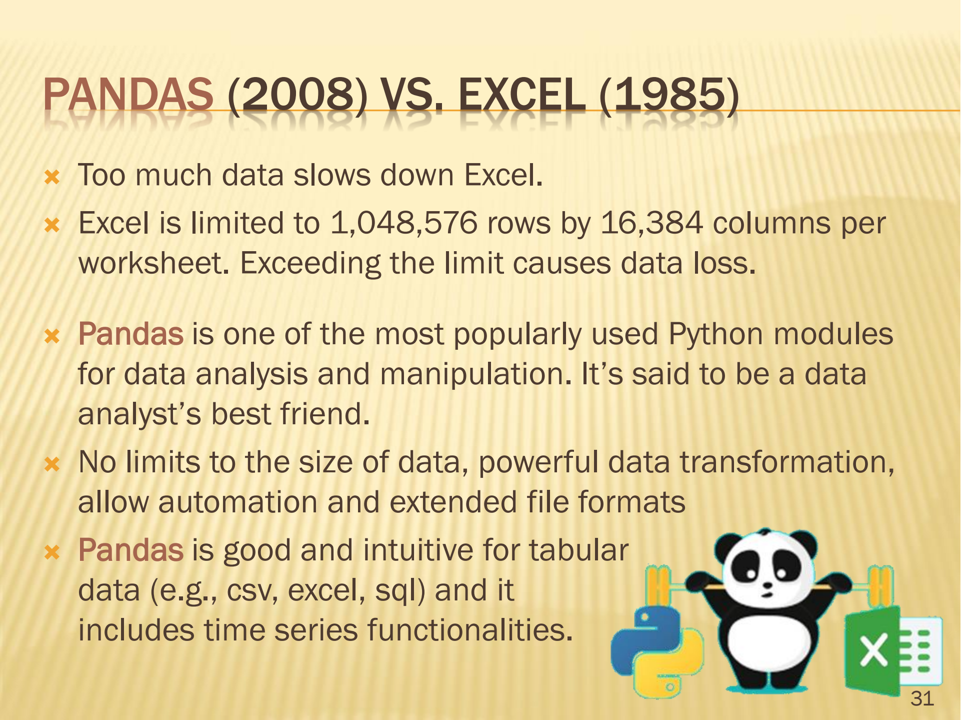
```
lst_2d = [ [250, 1564, 1020],    # January
           [200, 1275, 875],     # February
           [350, 1800, 1275],    # March
           [400, 1985, 1500],    # April
           [485, 2000, 1560] ]   # May
```

```
>>> print( len(lst_2d) )
5
>>> print( lst_2d[0] )
[250, 1564, 1020]
>>> print( lst_2d[1] )
[200, 1275, 875]
>>> print( lst_2d[2] )
[350, 1800, 1275]
>>> print( lst_2d[3] )
[400, 1985, 1500]
>>> print( lst_2d[4] )
[485, 2000, 1560]
```

```
>>> print( len(lst_2d[0]) )
3
>>> print( lst_2d[0][0] )
250
>>> print( lst_2d[0][1] )
1564
>>> print( lst_2d[0][2] )
1020
```

30

PPT template, images and decorating graphics from www.google.com unless otherwise specified.

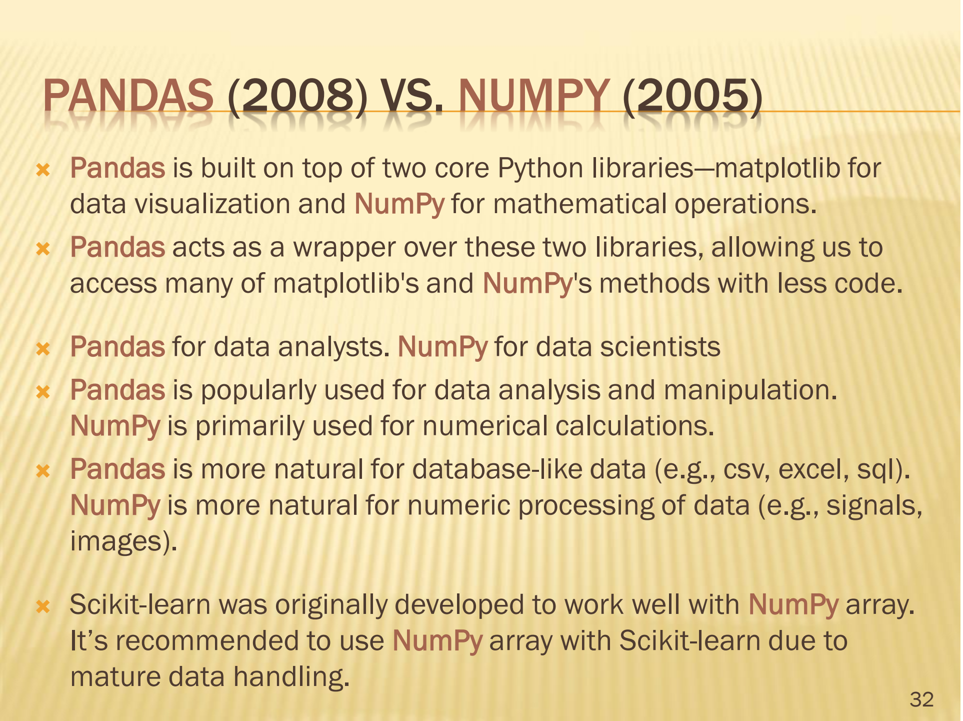


PANDAS (2008) VS. EXCEL (1985)

- Too much data slows down Excel.
- Excel is limited to 1,048,576 rows by 16,384 columns per worksheet. Exceeding the limit causes data loss.
- Pandas is one of the most popularly used Python modules for data analysis and manipulation. It’s said to be a data analyst’s best friend.
- No limits to the size of data, powerful data transformation, allow automation and extended file formats
- Pandas is good and intuitive for tabular data (e.g., csv, excel, sql) and it includes time series functionalities.

31

31



PANDAS (2008) VS. NUMPY (2005)

- Pandas is built on top of two core Python libraries—matplotlib for data visualization and NumPy for mathematical operations.
- Pandas acts as a wrapper over these two libraries, allowing us to access many of matplotlib's and NumPy's methods with less code.
- Pandas for data analysts. NumPy for data scientists
- Pandas is popularly used for data analysis and manipulation. NumPy is primarily used for numerical calculations.
- Pandas is more natural for database-like data (e.g., csv, excel, sql). NumPy is more natural for numeric processing of data (e.g., signals, images).
- Scikit-learn was originally developed to work well with NumPy array. It’s recommended to use NumPy array with Scikit-learn due to mature data handling.

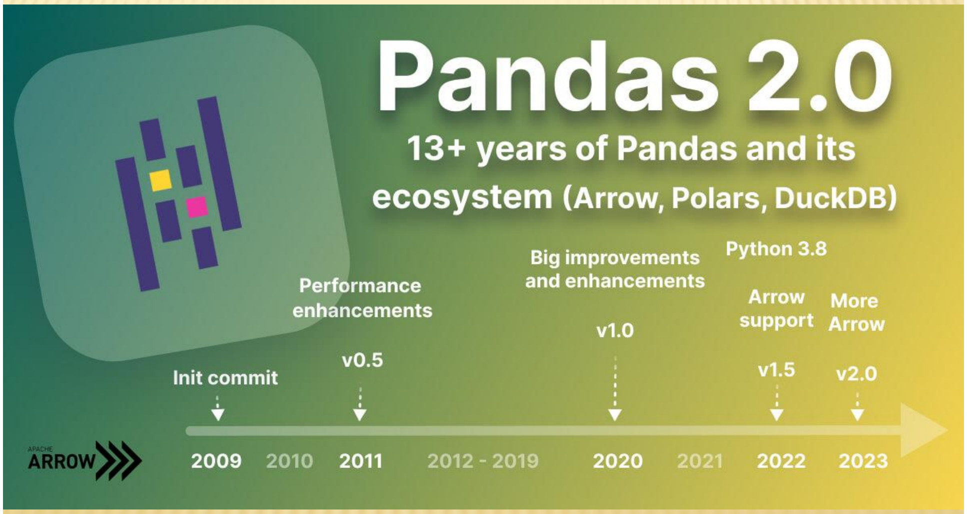
32

32

PPT template, images and decorating graphics from www.google.com unless otherwise specified.

PANDAS 2.0 (1/4)

What's new in 2.0.0 (April 3, 2023): https://pandas.pydata.org/docs/whatsnew/v2.0.0.html

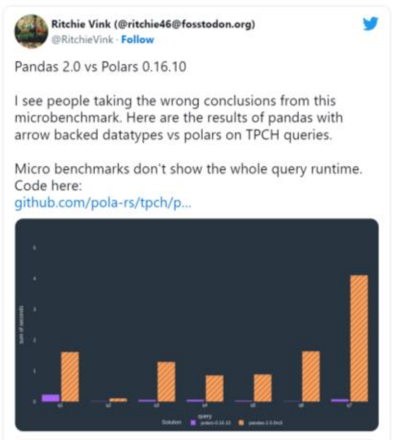


Credit: https://airbyte.com/blog/pandas-2-0-ecosystem-arrow-polars-duckdb

PANDAS 2.0 (2/4)

What's new in 2.0.0 (April 3, 2023): https://pandas.pydata.org/docs/whatsnew/v2.0.0.html

- Pandas 2.0 is faster and more memory efficient with the PyArrow backend (Apache Arrow).

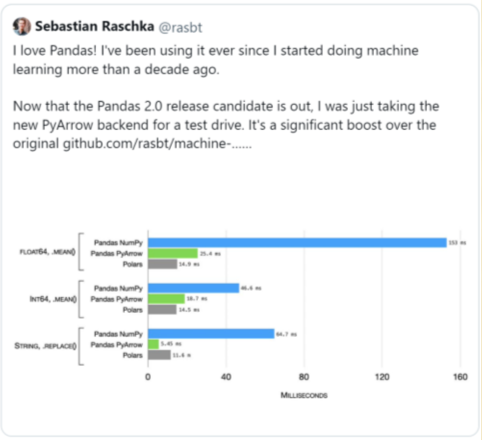


Ritchie Vink (@ritchie46@fosstodon.org)
@RitchieVink · Follow

Pandas 2.0 vs Polars 0.16.10

I see people taking the wrong conclusions from this microbenchmark. Here are the results of pandas with arrow backed datatypes vs polars on TPCH queries.

Micro benchmarks don't show the whole query runtime. Code here:
github.com/pola-rs/tpch/p...

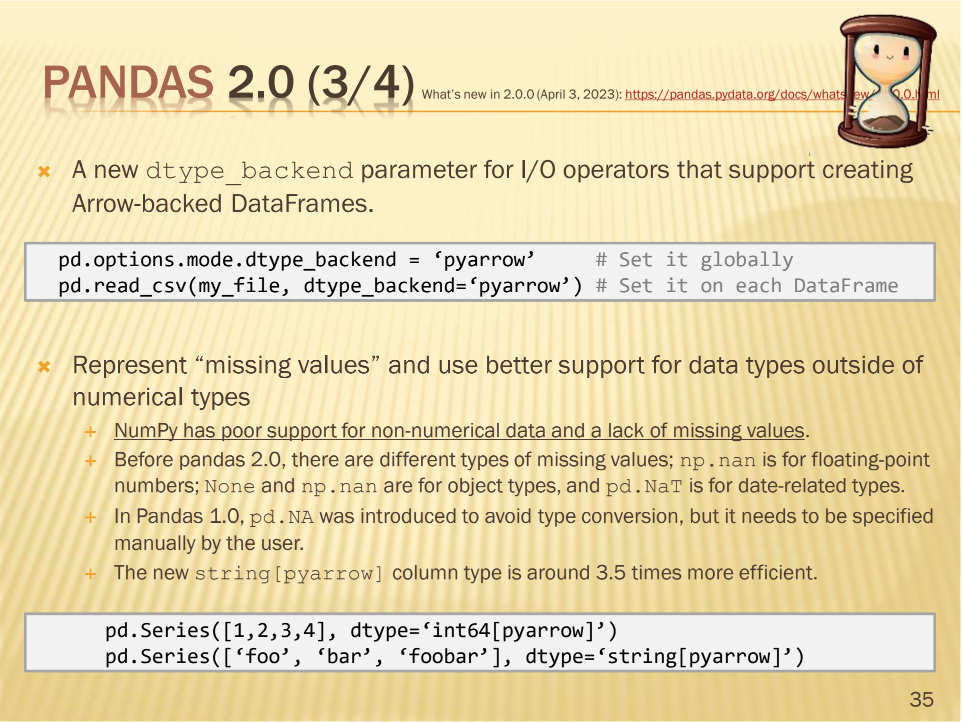


Sebastian Raschka @rasbt

I love Pandas! I've been using it ever since I started doing machine learning more than a decade ago.

Now that the Pandas 2.0 release candidate is out, I was just taking the new PyArrow backend for a test drive. It's a significant boost over the original github.com/rasbt/machine-......

Credit: https://twitter.com/RitchieVink/status/1632334005264580608



PANDAS 2.0 (3/4)

What's new in 2.0.0 (April 3, 2023): https://pandas.pydata.org/docs/whats ew/ 0.0.h ml

- A new `dtype_backend` parameter for I/O operators that support creating Arrow-backed DataFrames.

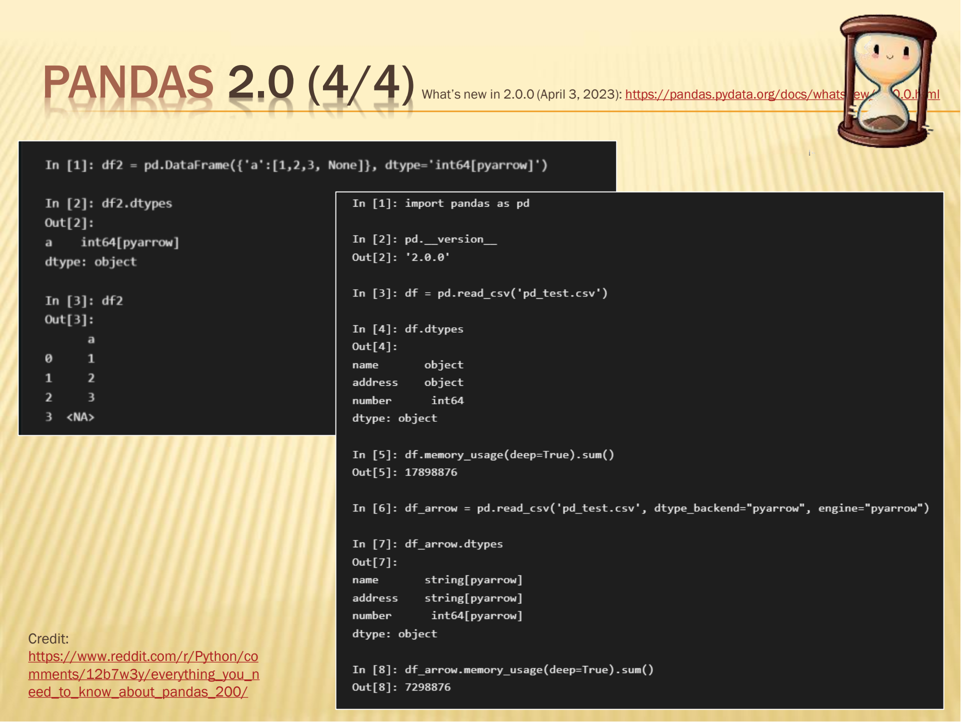
```
pd.options.mode.dtype_backend = ‘pyarrow’      # Set it globally
pd.read_csv(my_file, dtype_backend=‘pyarrow’) # Set it on each DataFrame
```

- Represent “missing values” and use better support for data types outside of numerical types
  - NumPy has poor support for non-numerical data and a lack of missing values.
  - Before pandas 2.0, there are different types of missing values; `np.nan` is for floating-point numbers; `None` and `np.nan` are for object types, and `pd.NaT` is for date-related types.
  - In Pandas 1.0, `pd.NA` was introduced to avoid type conversion, but it needs to be specified manually by the user.
  - The new `string[pyarrow]` column type is around 3.5 times more efficient.

```
pd.Series([1,2,3,4], dtype=‘int64[pyarrow]’)
pd.Series([‘foo’, ‘bar’, ‘foobar’], dtype=‘string[pyarrow]’)
```

35

35



PANDAS 2.0 (4/4)

What's new in 2.0.0 (April 3, 2023): https://pandas.pydata.org/docs/whats ew/ 0.0.h ml

```
In [1]: df2 = pd.DataFrame({'a':[1,2,3, None]}, dtype='int64[pyarrow]')

In [2]: df2.dtypes
Out[2]:
a    int64[pyarrow]
dtype: object

In [3]: df2
Out[3]:
      a
0     1
1     2
2     3
3  <NA>
```

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

In [2]: pd.__version__
Out[2]: '2.0.0'

In [3]: df = pd.read_csv('pd_test.csv')

In [4]: df.dtypes
Out[4]:
name       object
address    object
number      int64
dtype: object

In [5]: df.memory_usage(deep=True).sum()
Out[5]: 17898876

In [6]: df_arrow = pd.read_csv('pd_test.csv', dtype_backend="pyarrow", engine="pyarrow")

In [7]: df_arrow.dtypes
Out[7]:
name       string[pyarrow]
address    string[pyarrow]
number      int64[pyarrow]
dtype: object

In [8]: df_arrow.memory_usage(deep=True).sum()
Out[8]: 7298876
```

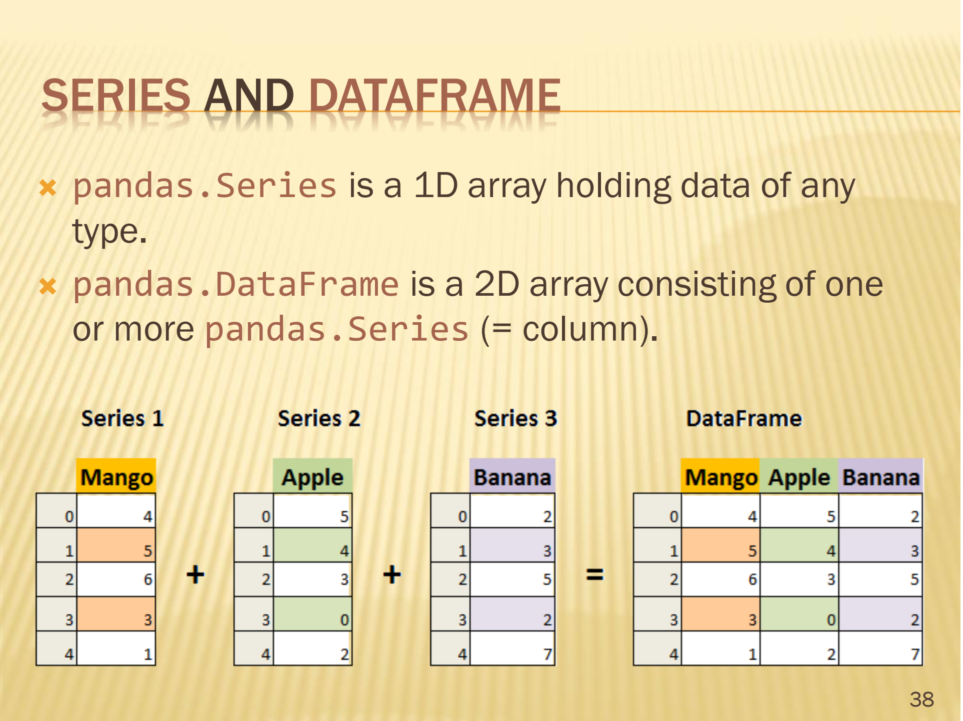
Credit:
https://www.reddit.com/r/Python/comments/12b7w3y/everything_you_need_to_know_about_pandas_200/

36

PANDAS

37

37



SERIES AND DATAFRAME

- `pandas.Series` is a 1D array holding data of any type.
- `pandas.DataFrame` is a 2D array consisting of one or more `pandas.Series` (= column).

Series 1

	Mango
0	4
1	5
2	6
3	3
4	1

+

Series 2

	Apple
0	5
1	4
2	3
3	0
4	2

+

Series 3

	Banana
0	2
1	3
2	5
3	2
4	7

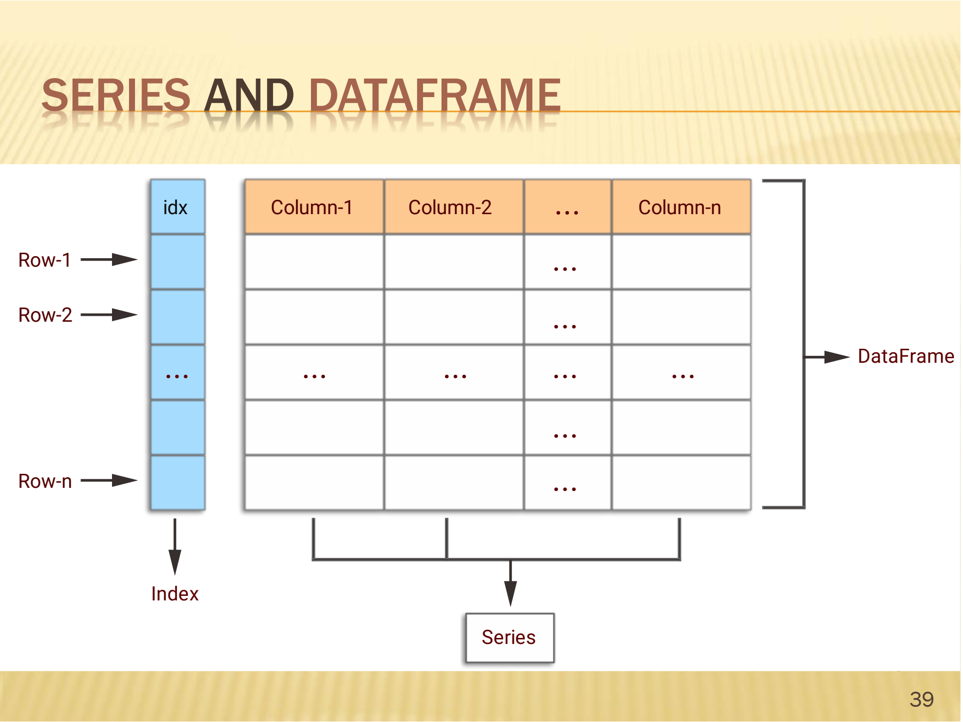
=

DataFrame

	Mango	Apple	Banana
0	4	5	2
1	5	4	3
2	6	3	5
3	3	0	2
4	1	2	7

38

38



SERIES AND DATAFRAME

idx

Column-1 Column-2 … Column-n

Row-1

Row-2

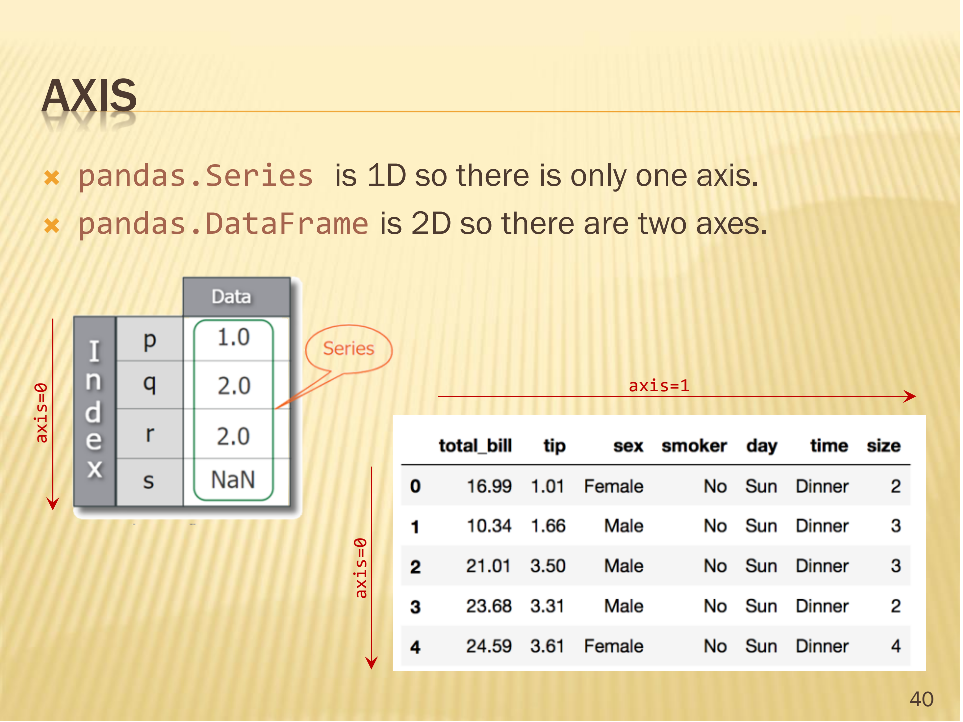
…

Row-n

Index

Series

DataFrame



AXIS

- pandas.Series is 1D so there is only one axis.
- pandas.DataFrame is 2D so there are two axes.

Data

Index

p	1.0
q	2.0
r	2.0
s	NaN

Series

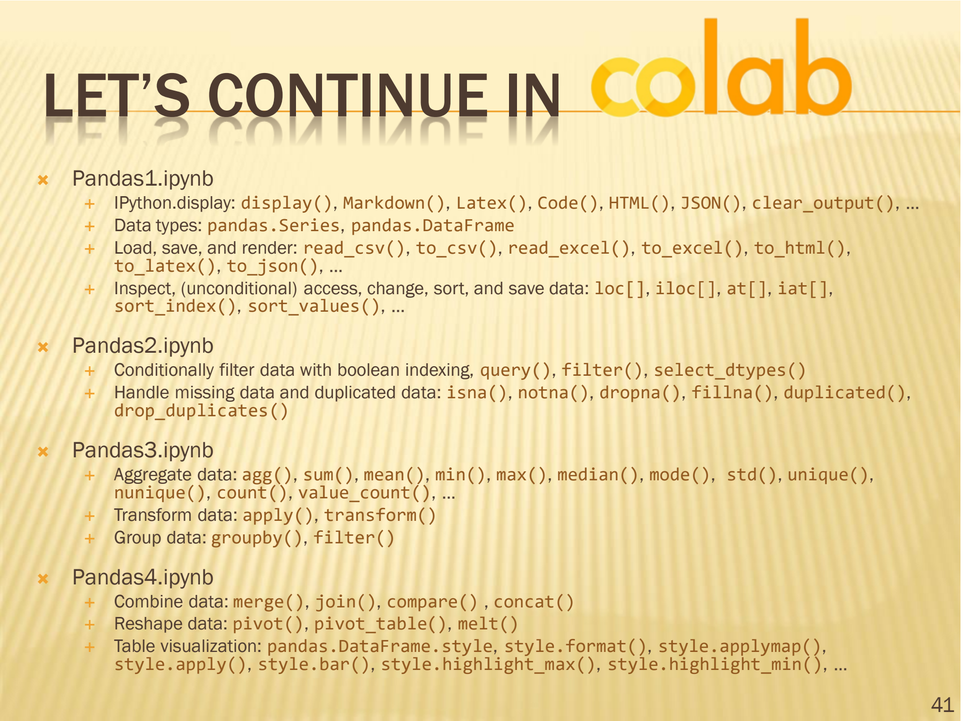
axis=0

axis=1

axis=0

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

PPT template, images and decorating graphics from www.google.com unless otherwise specified.

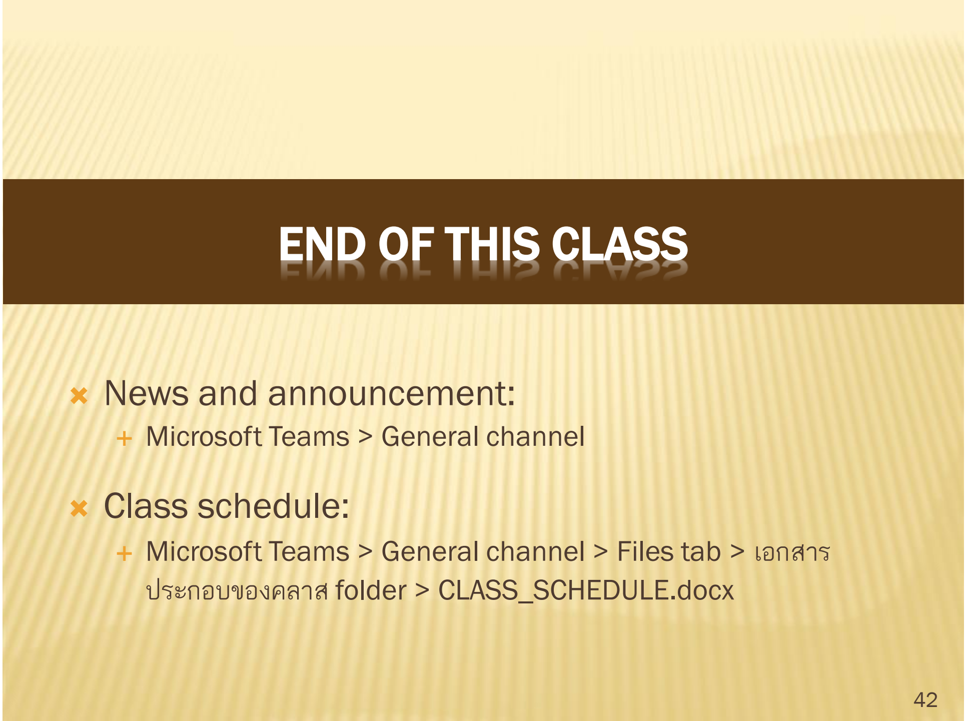


LET'S CONTINUE IN colab

- Pandas1.ipynb
  - IPython.display: `display()`, `Markdown()`, `Latex()`, `Code()`, `HTML()`, `JSON()`, `clear_output()`, …
  - Data types: `pandas.Series`, `pandas.DataFrame`
  - Load, save, and render: `read_csv()`, `to_csv()`, `read_excel()`, `to_excel()`, `to_html()`, `to_latex()`, `to_json()`, …
  - Inspect, (unconditional) access, change, sort, and save data: `loc[]`, `iloc[]`, `at[]`, `iat[]`, `sort_index()`, `sort_values()`, …
- Pandas2.ipynb
  - Conditionally filter data with boolean indexing, `query()`, `filter()`, `select_dtypes()`
  - Handle missing data and duplicated data: `isna()`, `notna()`, `dropna()`, `fillna()`, `duplicated()`, `drop_duplicates()`
- Pandas3.ipynb
  - Aggregate data: `agg()`, `sum()`, `mean()`, `min()`, `max()`, `median()`, `mode()`, `std()`, `unique()`, `nunique()`, `count()`, `value_count()`, …
  - Transform data: `apply()`, `transform()`
  - Group data: `groupby()`, `filter()`
- Pandas4.ipynb
  - Combine data: `merge()`, `join()`, `compare()` , `concat()`
  - Reshape data: `pivot()`, `pivot_table()`, `melt()`
  - Table visualization: `pandas.DataFrame.style`, `style.format()`, `style.applymap()`, `style.apply()`, `style.bar()`, `style.highlight_max()`, `style.highlight_min()`, …

41

41



END OF THIS CLASS

- News and announcement:
  - Microsoft Teams > General channel
- Class schedule:
  - Microsoft Teams > General channel > Files tab > เอกสารประกอบของคลาส folder > CLASS_SCHEDULE.docx

42

42

PPT template, images and decorating graphics from www.google.com unless otherwise specified.