INMAS

Real Data

- Storing Data
  - File Systems
- Flat Files
- Real Data

James Balamuta

Inmas Fall 2021 Statistical Methods Workshop

Lecture Objectives

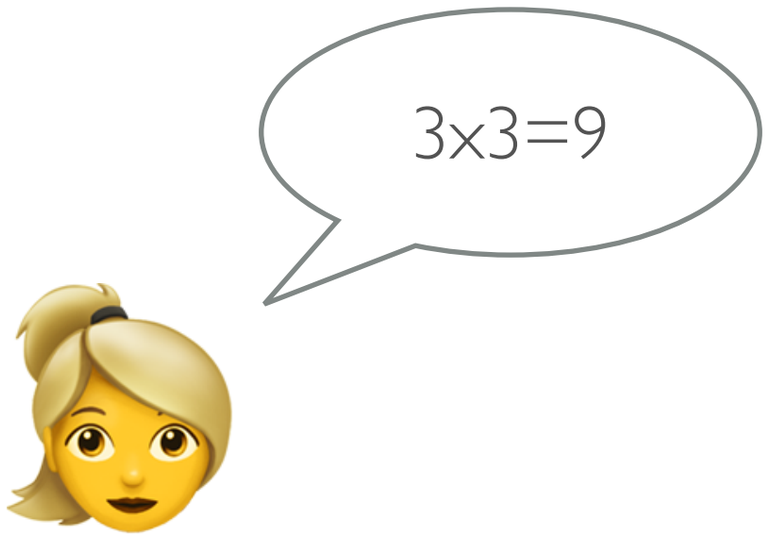
- *Describe* data storage options on disk.
- *Review* file system logic.
- *Manipulate* data with pandas.
- *Emphasize* how data can be missing.

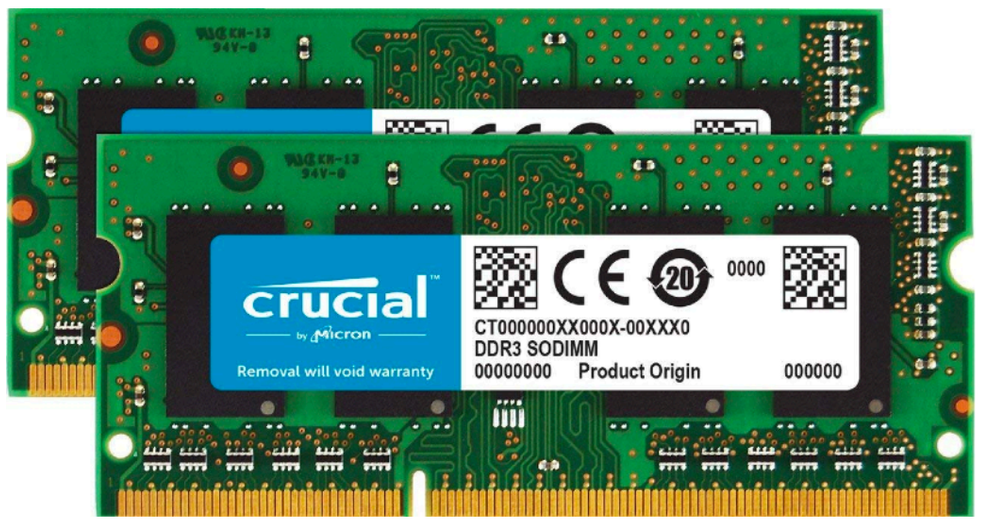
Storing Data

How does a computer **store data**?

Definition:
Random Access Memory (RAM) stores data for short-term usage on a per-session basis.

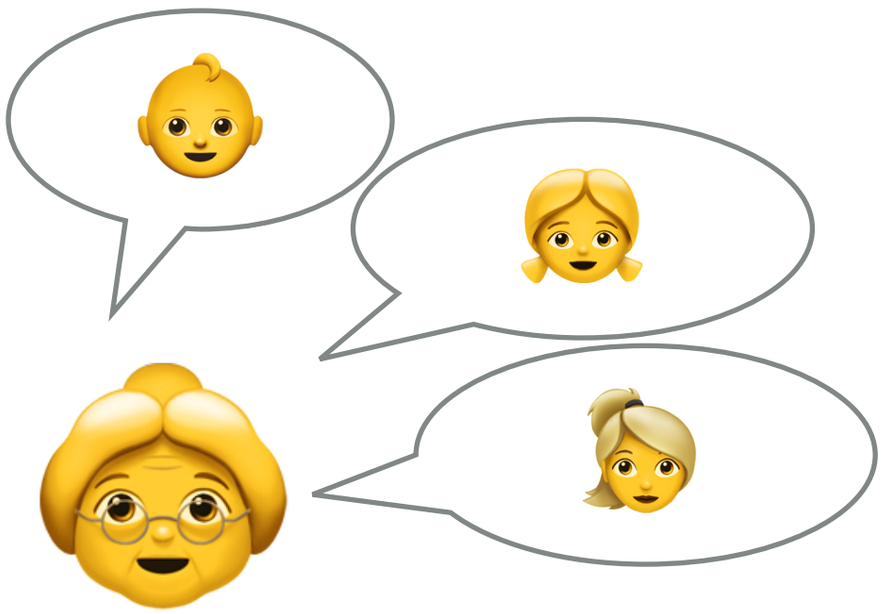
Note: *Python* stores data or objects found in the global environment in memory.





Source

Definition:
Hard Drives (HDs) and *Solid State Drives (SSDs)* store data for the **long term** (e.g. months from now).



"Memories"

Source

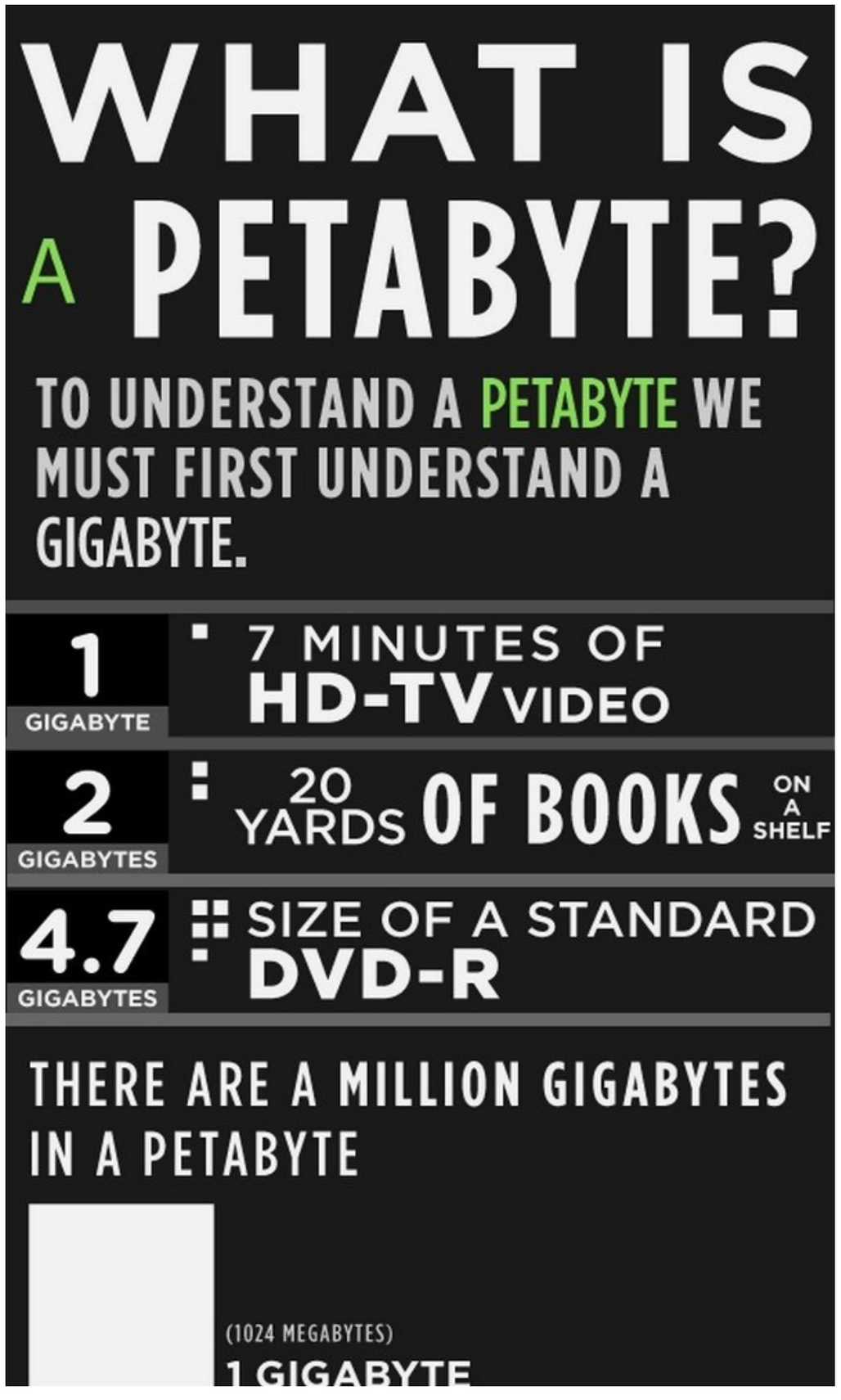
Source

Definition:
Storage size refers to the amount of space required to hold the data contents on a computer.

Traditional units						Decimal for comparison				
Name	IEC	Binary	Number of bytes	Equal to		Name	IEC	Decimal	Number of bits	Equal to
Kilobyte	KiB	210	1,024	1024 B		Kilobit	kbit	10^3	1,000	1000 bit
Megabyte	MiB	220	1,048,576	1024 KiB		Megabit	Mbit	10^6	1,000,000	1000 kbit
Gigabyte	GiB	230	1,073,741,824	1024 MiB		Gigabit	Gbit	10^9	1,000,000,000	1000 Mbit
Terabyte	TiB	240	1,099,511,627,776	1024 GiB		Terabit	Tbit	10^{12}	1,000,000,000,000	1000 Gbit
Petabyte	PiB	250	1,125,899,906,842,624	1024 TiB		Petabit	Pbit	10^{15}	1,000,000,000,000,000	1000 Tbit
Exabyte	EiB	260	1,152,921,504,606,846,976	1024 PiB		Exabit	Ebit	10^{18}	1,000,000,000,000,000,000	1000 Pbit
Zettabyte	ZiB	270	1,180,591,620,717,411,303,424	1024 EiB		Zettabit	Zbit	10^{21}	1,000,000,000,000,000,000,000	1000 Ebit
Yottabyte	YiB	280	1,208,925,819,614,629,174,706,176	1024 ZiB		Yottabit	Ybit	10^{24}	1,000,000,000,000,000,000,000,000	1000 Zbit

Byte: 8 bits per byte
Large (2^8)

bit: binary digit
Small


WHAT IS A PETABYTE?
TO UNDERSTAND A PETABYTE WE MUST FIRST UNDERSTAND A GIGABYTE.
1 GIGABYTE
7 MINUTES OF HD-TV VIDEO
2 GIGABYTES
20 YARDS OF BOOKS ON A SHELF
4.7 GIGABYTES
SIZE OF A STANDARD DVD-R
THERE ARE A MILLION GIGABYTES IN A PETABYTE
(1024 MEGABYTES)
1 GIGABYTE

Source

Definition:
File Formats are standardized ways that data may be stored on a computer.

.xlsx

.docx

.pdf

Types of Data Formats

Pandas IO Tools

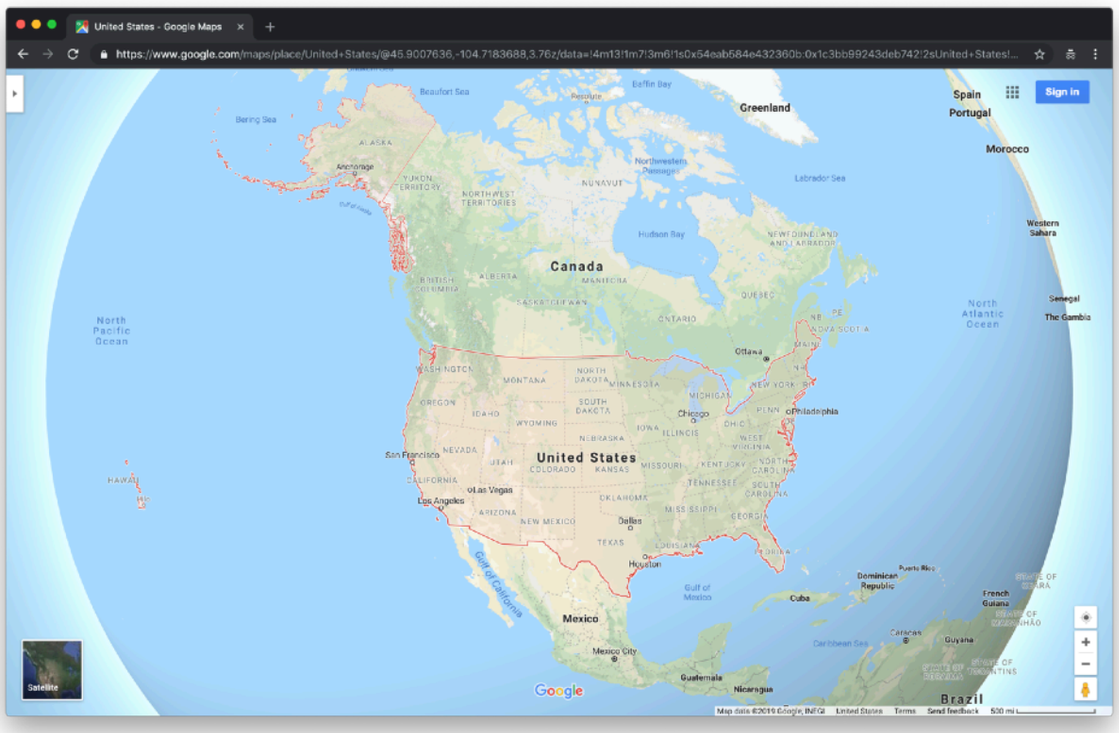
Format Type	Data Description	Reader	Writer
text	CSV	read_csv	to_csv
text	Fixed-Width Text File	read_fwf	
text	JSON	read_json	to_json
text	HTML	read_html	to_html
text	LaTeX		Styler.to_latex
text	XML	read_xml	to_xml
text	Local clipboard	read_clipboard	to_clipboard
binary	MS Excel	read_excel	to_excel
binary	HDF5 Format	read_hdf	to_hdf
binary	Feather Format	read_feather	to_feather
binary	Parquet Format	read_parquet	to_parquet
binary	Stata	read_stata	to_stata
binary	SAS	read_sas	
binary	SPSS	read_spss	
binary	Python Pickle Format	read_pickle	to_pickle
SQL	SQL	read_sql	to_sql
SQL	Google BigQuery	read_gbq	to_gbq

File Systems

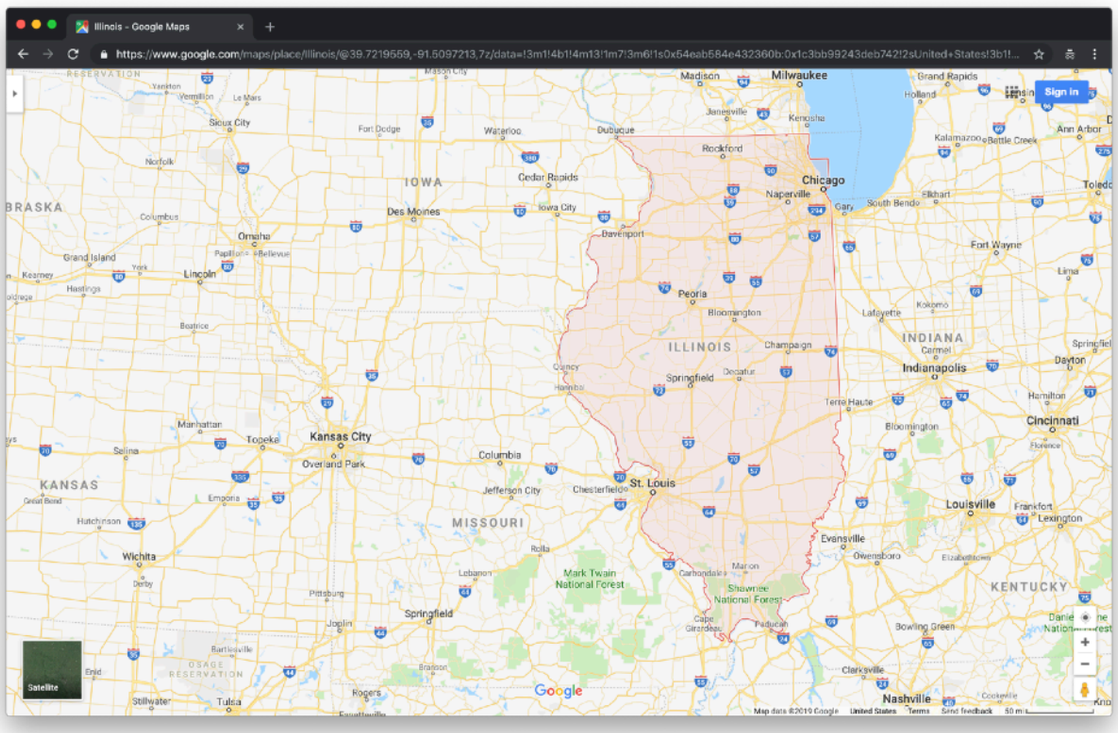
How is **data organized** on a computer?

Definition:
Directories or *folders* acts as a catalog that contains content in a way similar to a file cabinet.

World > United States

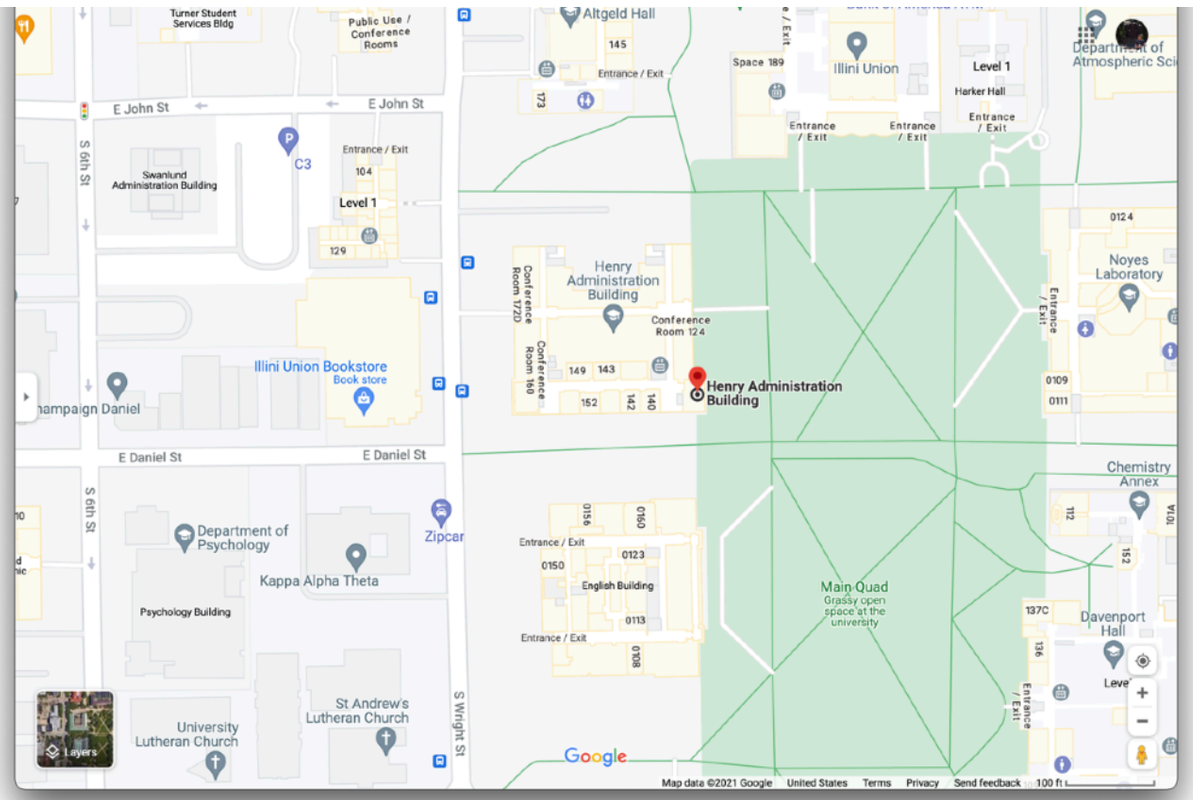


World > United States > IL



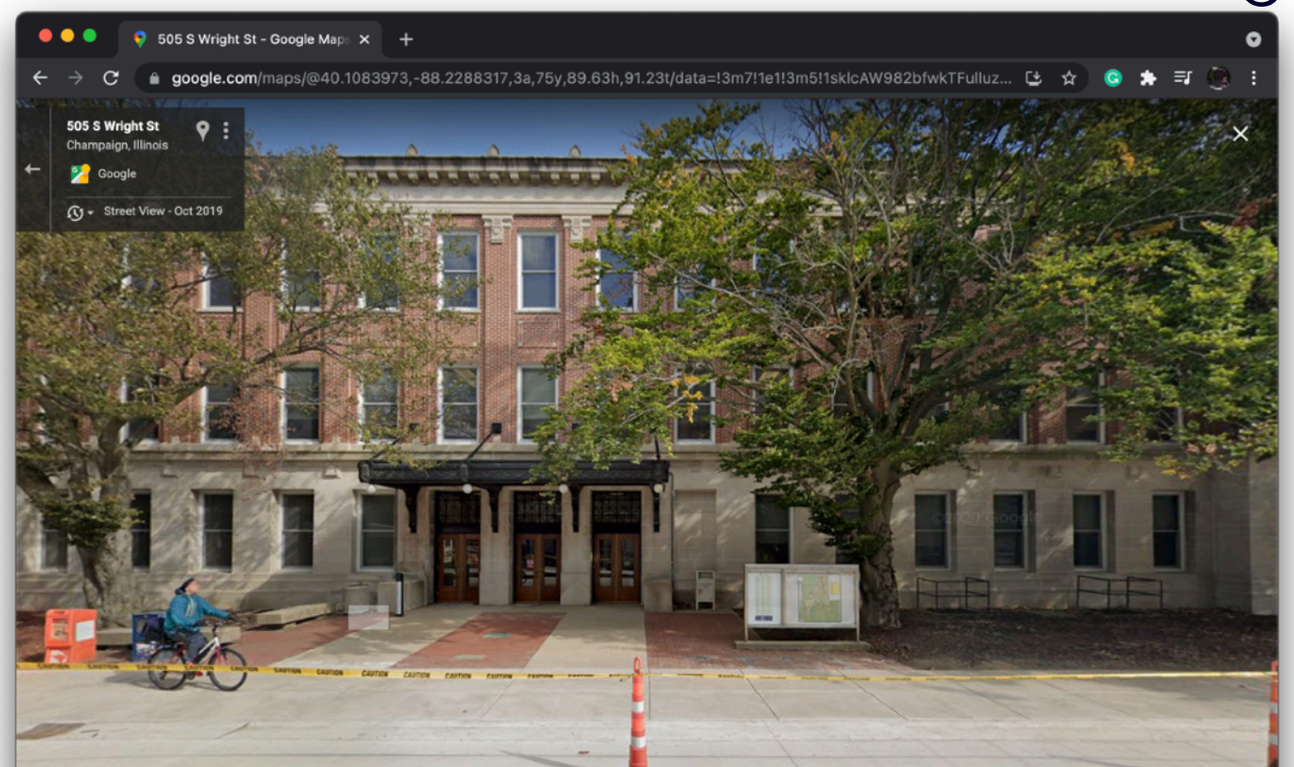
Definition:
File Path or *Path* provides a unique address to access content on the computer.

World > United States > IL > 61820 > 505 S. Wright St.



Definition:
A *File* contains the content found the end of the path.

Contents of:
World > United States > IL > 61820 > 505 S. Wright St.



Working Directories

… where you are …

```
# Retrieve working directory
import os


current_path = os.getcwd()
print(current_path)
#  [1] "/content"
```

You are here

Definition:
Fixed or *Absolute paths* are computer/user specific based on the root (**/**) directory.

C:/Users/James/Hypno/Toad.py
/etc/python/hosts
/Users/James/URA/reports/summary.docx

Definition:
Relative paths resolve from the present working directory location

Hypno/Toad.py
hosts
reports/summary.docx

Absolute Paths by OS

… identifying operating systems (OS) by path prefixes …

C:/Users/balamut2

/Users/balamut2

/home/balamut2

/content/drive/My\ Drive

Your Turn

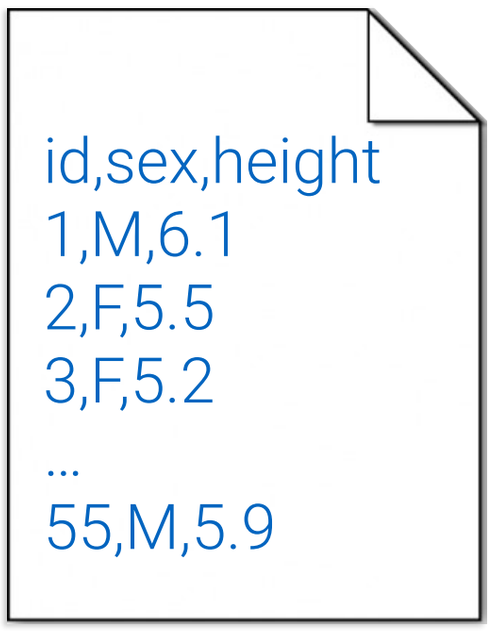
Consider a file located at:

C:/Users/James/Hypno/Toad.py

1. What kind of path is it?
2. Which operating system is the path on?
3. What would be the relative path if the working directory is James?

Reading Flat Files

Definition:
Flat Files store data in text that may or may not easily be human readable. However, changes between data versions can be detect under a version control system.



949 csv/tre-fa1993.csv — View file

```
@@ -1,144 +1,146 @@
-term,unit,lname,fname,role,ranking,courses
 fa1993,"ACCOUNTANCY",DIETRICH,R,Instructor,Excellent,201
-fa1993,"ACCOUNTANCY",ONES,R,TA,Outstanding,311
 fa1993,"ACCOUNTANCY",MOLLOY,K,Instructor,Excellent,456
 fa1993,"ACCOUNTANCY",MOLLOY,K,Instructor,Excellent,405
 fa1993,"ACCOUNTANCY",OMER,T,Instructor,Excellent,251
 fa1993,"ACCOUNTANCY",SCHOENFELD,H,Instructor,Excellent,401
-fa1993,"ACCOUNTANCY",YATT,A,Instructor,Outstanding,211
-fa1993,"ACCOUNTANCY",IEBART,D,Instructor,Outstanding,494
 fa1993,"ACCOUNTANCY",ZIEGLER,R,Instructor,Excellent,341
 fa1993,"ADMINISTRATION, HIGHER, & CONTINUING EDUCATION",MCGREAL,T,Instructor,Excellent,438
 fa1993,"ADMINISTRATION, HIGHER, & CONTINUING EDUCATION",MERCHANT,B,Instructor,Excellent,461
-fa1993,"ADMINISTRATION, HIGHER, & CONTINUING EDUCATION",RESTINE,N,Instructor,Outstanding,490
-fa1993,"ADVERTISING",ENNETT,R,TA,Outstanding,382
-fa1993,"ADVERTISING",ORNELIOUS,V,TA,Outstanding,382
 fa1993,"ADVERTISING",OTNES,C,Instructor,Excellent,382
 fa1993,"ADVERTISING",ROTZOLL,K,Instructor,Excellent,450
 fa1993,"ADVERTISING",SCOTT,L,Instructor,Excellent,490
 fa1993,"ADVERTISING",SCOTT,L,Instructor,Excellent,393
-fa1993,"ADVERTISING",HELDON,P,TA,Outstanding,382
+term,unit,lname,fname,role,ranking,course
 fa1993,"ACCOUNTANCY",DIETRICH,R,Instructor,Excellent,201
+fa1993,"ACCOUNTANCY",JONES,R,TA,Outstanding,311
 fa1993,"ACCOUNTANCY",MOLLOY,K,Instructor,Excellent,456
 fa1993,"ACCOUNTANCY",MOLLOY,K,Instructor,Excellent,405
 fa1993,"ACCOUNTANCY",OMER,T,Instructor,Excellent,251
 fa1993,"ACCOUNTANCY",SCHOENFELD,H,Instructor,Excellent,401
+fa1993,"ACCOUNTANCY",WYATT,A,Instructor,Outstanding,211
+fa1993,"ACCOUNTANCY",ZIEBART,D,Instructor,Outstanding,494
 fa1993,"ACCOUNTANCY",ZIEGLER,R,Instructor,Excellent,341
 fa1993,"ADMINISTRATION, HIGHER, & CONTINUING EDUCATION",MCGREAL,T,Instructor,Excellent,438
 fa1993,"ADMINISTRATION, HIGHER, & CONTINUING EDUCATION",MERCHANT,B,Instructor,Excellent,461
+fa1993,"ADMINISTRATION, HIGHER, & CONTINUING
 EDUCATION",PRESTINE,N,Instructor,Outstanding,490
+fa1993,"ADVERTISING",BENNETT,R,TA,Outstanding,382
+fa1993,"ADVERTISING",CORNELIOUS,V,TA,Outstanding,382
 fa1993,"ADVERTISING",OTNES,C,Instructor,Excellent,382
 fa1993,"ADVERTISING",ROTZOLL,K,Instructor,Excellent,450
 fa1993,"ADVERTISING",SCOTT,L,Instructor,Excellent,490
 fa1993,"ADVERTISING",SCOTT,L,Instructor,Excellent,393
+fa1993,"ADVERTISING",SHELDON,P,TA,Outstanding,382
```

Flat File "diff"

CSV Data

… “**C**omma” **S**eparated **V**alues (**CSV**) …

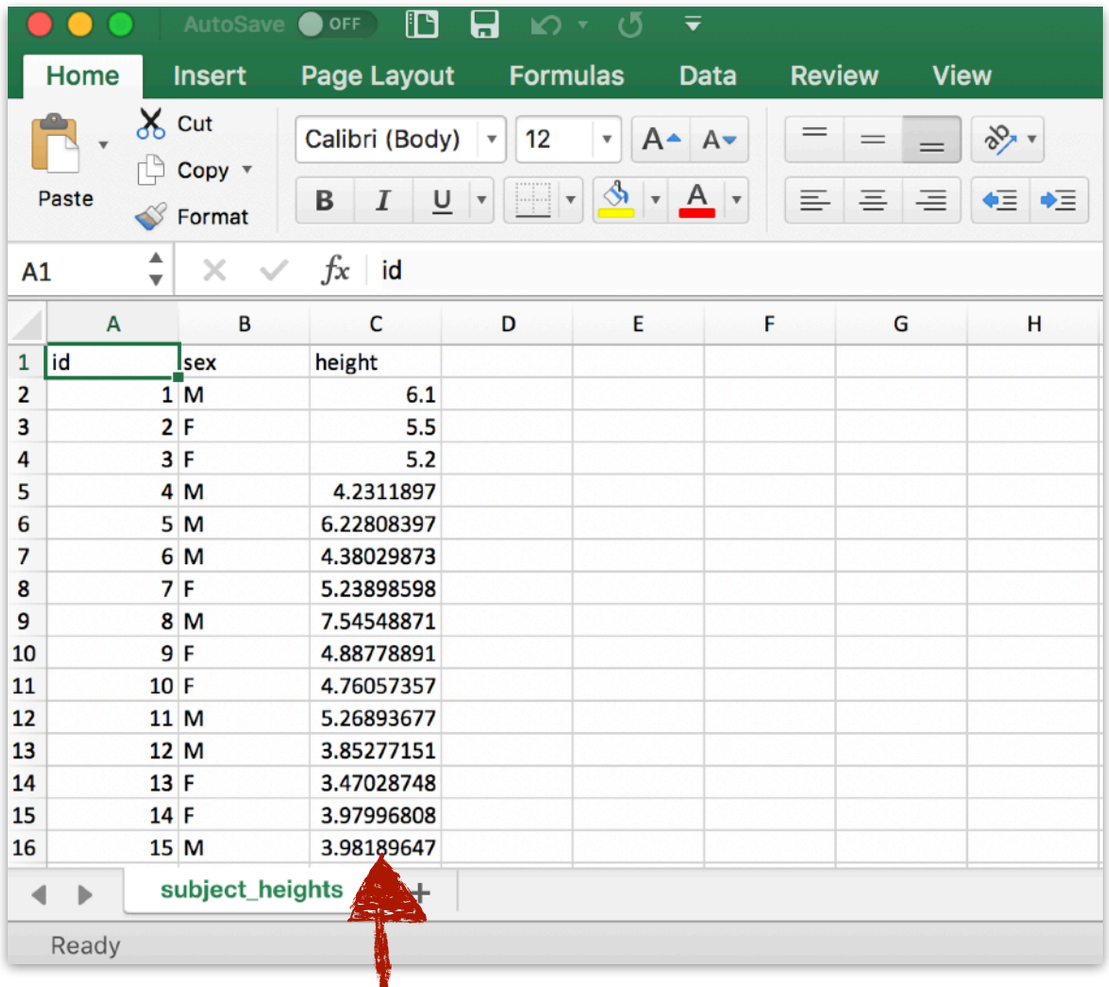
subject_heights.csv

Header Row

id,sex,height
1,M,6.1
2,F,5.5
3,F,5.2
…
55,M,5.9

Comma / Separator

Column



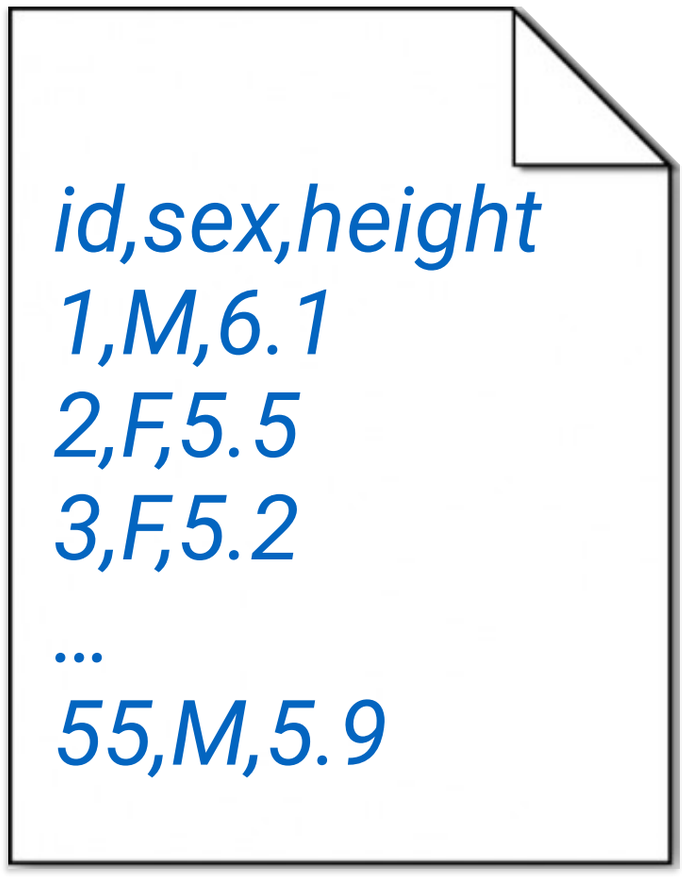
	A	B	C
1	id	sex	height
2	1	M	6.1
3	2	F	5.5
4	3	F	5.2
5	4	M	4.2311897
6	5	M	6.22808397
7	6	M	4.38029873
8	7	F	5.23898598
9	8	M	7.54548871
10	9	F	4.88778891
11	10	F	4.76057357
12	11	M	5.26893677
13	12	M	3.85277151
14	13	F	3.47028748
15	14	F	3.97996808
16	15	M	3.98189647

Column

Reading a CSV into *Python*

… loading data …

```
subject_heights = pd.read_csv("subject_heights.csv")
```



id,sex,height
1,M,6.1
2,F,5.5
3,F,5.2
…
55,M,5.9

subject_heights.csv

id	sex	height
1	M	6.1
2	F	5.5
3	F	5.2
…	…	…
55	M	5.9

subject_heights

American vs. European

… differing styles with the same extension …

subject_heights.csv
American

```
id,sex,height
1,M,6.1
2,F,5.5
3,F,5.2
…
55,M,5.9
```

```
ds = pd.read_csv("subject_heights.csv",
                 sep = ",")
```

subject_heights2.csv
European

```
id;sex;height
1;M;6,1
2;F;5,5
3;F;5,2
…
55;M;5,9
```

```
ds = pd.read_csv("subject_heights2.csv",
                 sep = ";")
```

“**C**omma” **S**eparated **V**alues (**CSV**)

Delimited Data

… custom separator between variables …

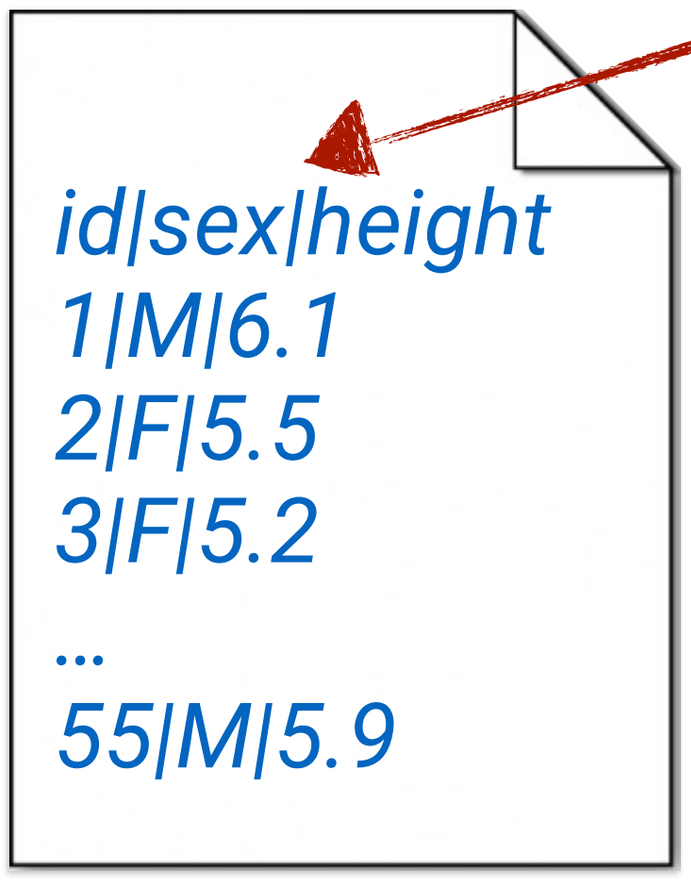
subject_heights.**txt**

Delimited Text File (**TXT**)

Reading a Delimited File

… loading data with a custom separator …

```
subject_heights = pd.read_csv("subject_heights.txt",
                              delimiter = "|")
```



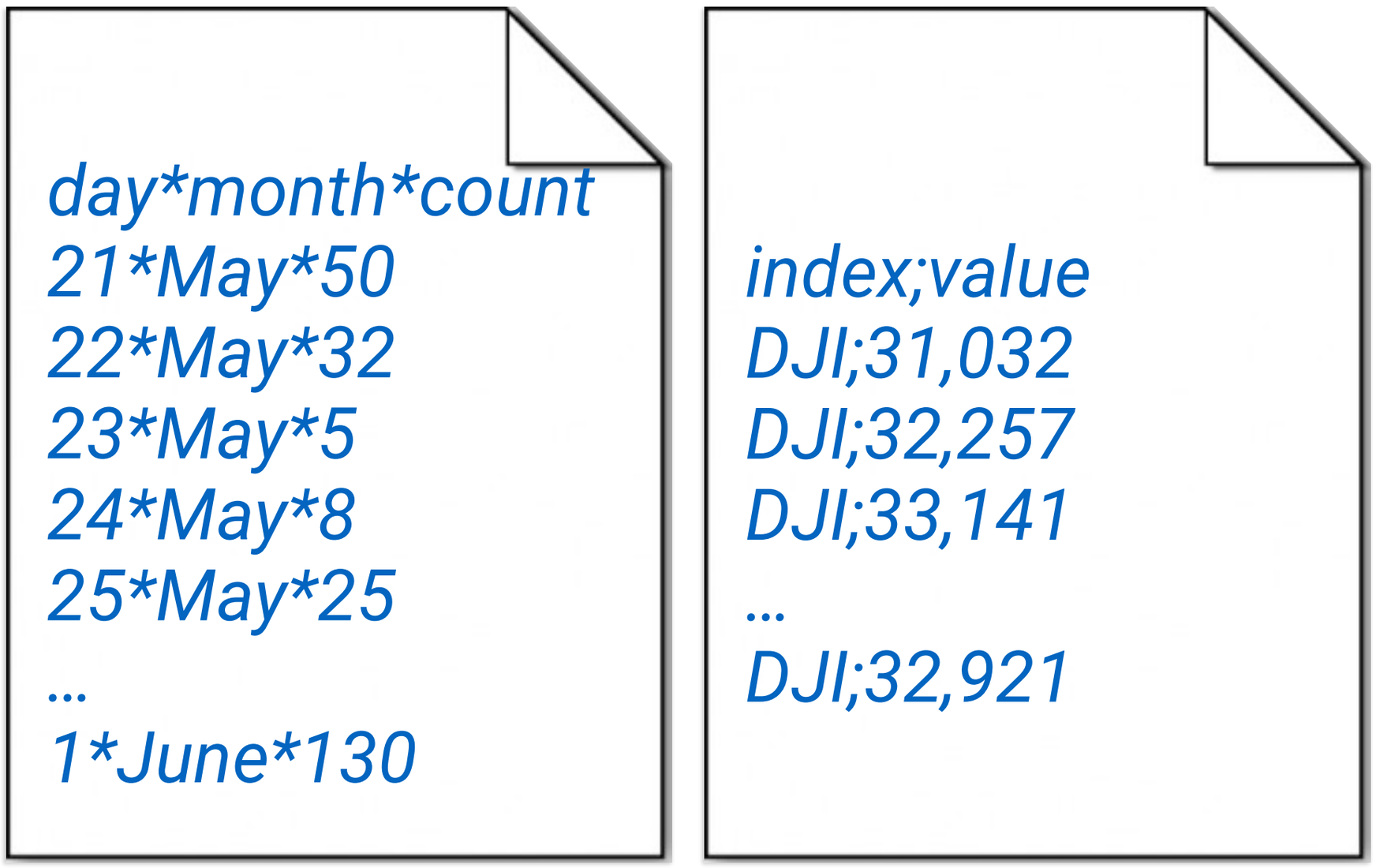
subject_heights.txt

id	sex	height
1	M	6.1
2	F	5.5
3	F	5.2
…	…	…
55	M	5.9

subject_heights

Your Turn

Consider the following files, what kind of separator or delimiter should be used to read the data into Python?



Reading a CSV from a URL

Imports CSV as DataFrame and Set Variable Names

```
# Read in Data
uci_adult_df = pd.read_csv(
    "https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data",
     sep=",",                    # Explicitly say the format.
     header = None,              # Indicate column names are missing
     na_values=['NA','?']) # Set NA values

# Add column names
uci_adult_df.columns = [
    "Age", "WorkClass", "fnlwgt", "Education", "EducationNum",
    "MaritalStatus", "Occupation", "Relationship", "Race", "Gender",
    "CapitalGain", "CapitalLoss", "HoursPerWeek", "NativeCountry", "Income"
]
```

Displaying Data

```
       Age          WorkClass  fnlwgt  ... HoursPerWeek   NativeCountry  Income
0       39          State-gov   77516  ...           40   United-States   <=50K
1       50   Self-emp-not-inc   83311  ...           13   United-States   <=50K
2       38            Private  215646  ...           40   United-States   <=50K
3       53            Private  234721  ...           40   United-States   <=50K
4       28            Private  338409  ...           40            Cuba   <=50K
...    ...                ...     ...  ...          ...             ...     ...
32556   27            Private  257302  ...           38   United-States   <=50K
32557   40            Private  154374  ...           40   United-States    >50K
32558   58            Private  151910  ...           40   United-States   <=50K
32559   22            Private  201490  ...           20   United-States   <=50K
32560   52       Self-emp-inc  287927  ...           40   United-States    >50K

[32561 rows x 15 columns]
```

	Age	WorkClass	fnlwgt	Education	EducationNum	MaritalStatus	Occupation
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty
...	...	...	...	...	...	...	...
32556	27	Private	257302	Assoc-acdm	12	Married-civ-spouse	Tech-support
32557	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct
32558	58	Private	151910	HS-grad	9	Widowed	Adm-clerical
32559	22	Private	201490	HS-grad	9	Never-married	Adm-clerical
32560	52	Self-emp-inc	287927	HS-grad	9	Married-civ-spouse	Exec-managerial

32561 rows × 15 columns

```
# See raw data form
print(uci_adult_df)
```

```
# View formatted data
display(uci_adult_df)
```

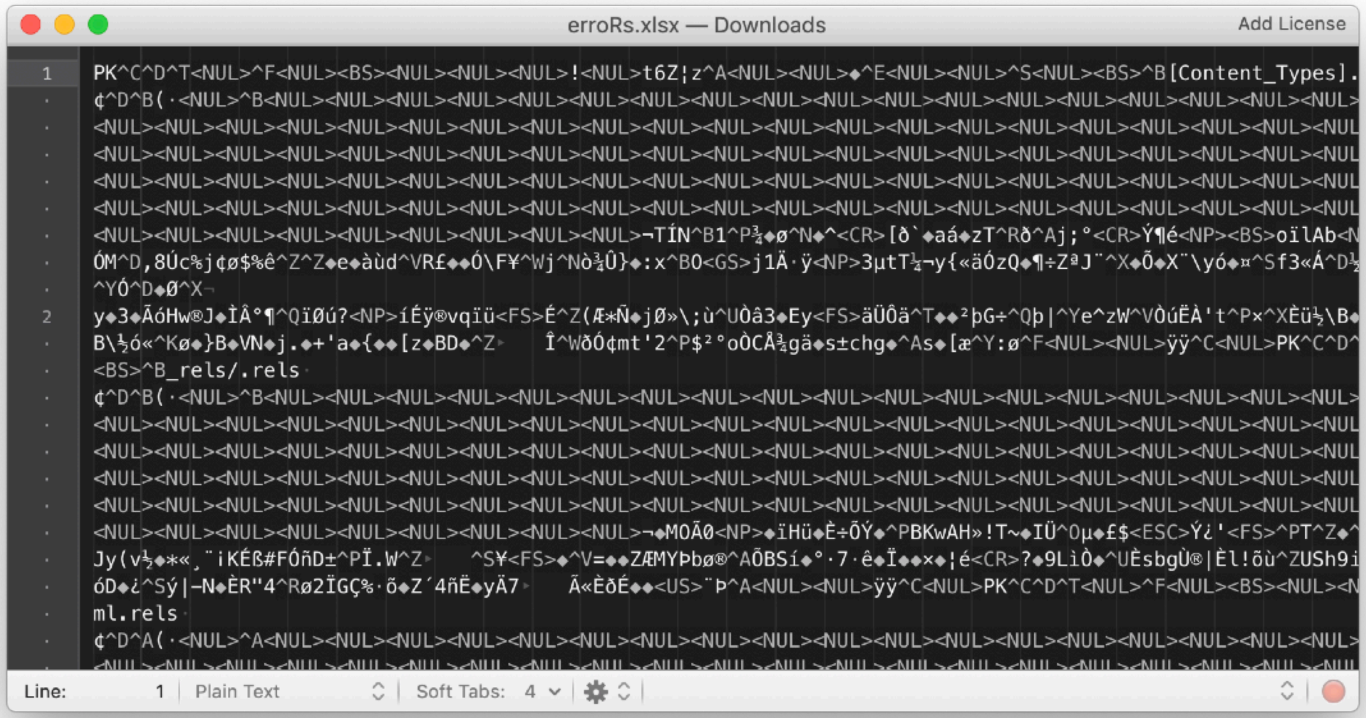
```
# What is different between
# the two views?
```

Binary Files

How can we **reduce stored data size**?

Definition:

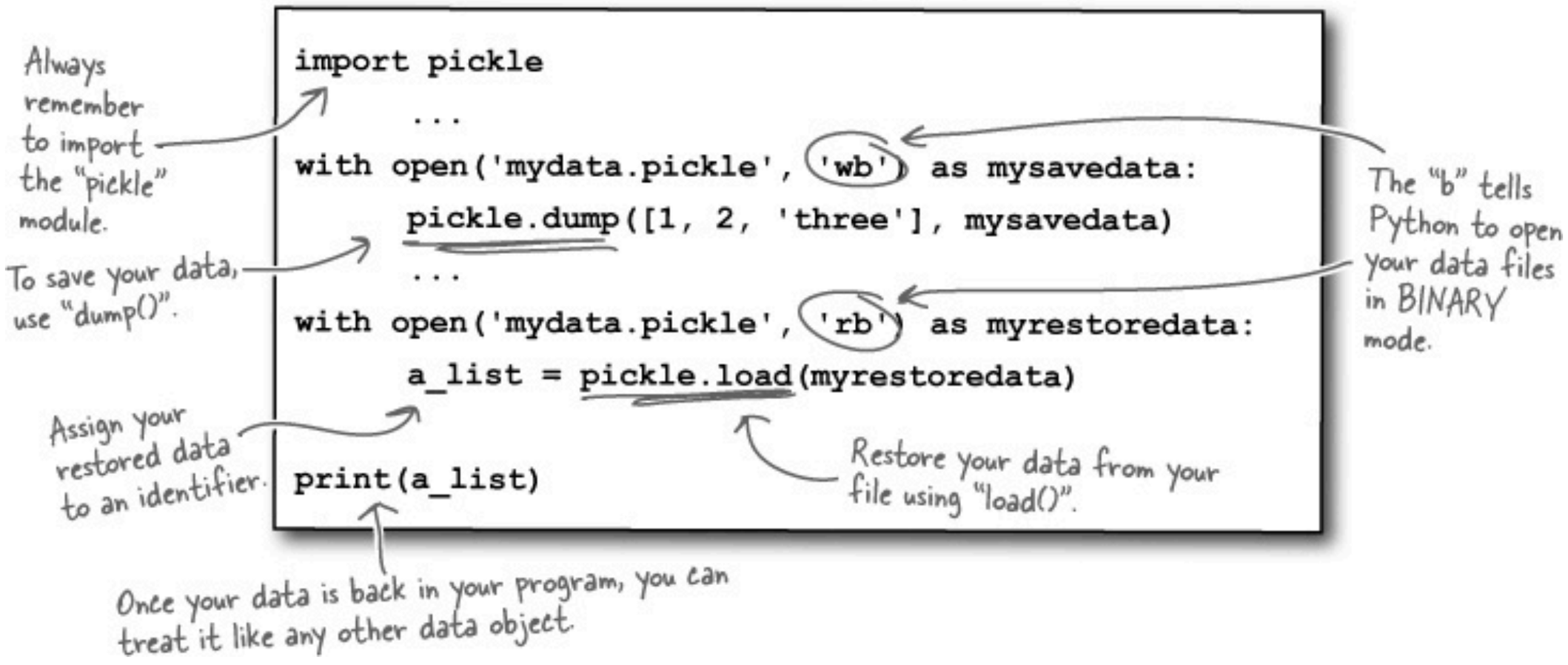
Binary Files save space by simplifying file contents under a pre-determined format. To re-open a file, you must translate it back.



https://github.com/pjmensink/errorR_dictionary/issues/1

Pickle or .pkl

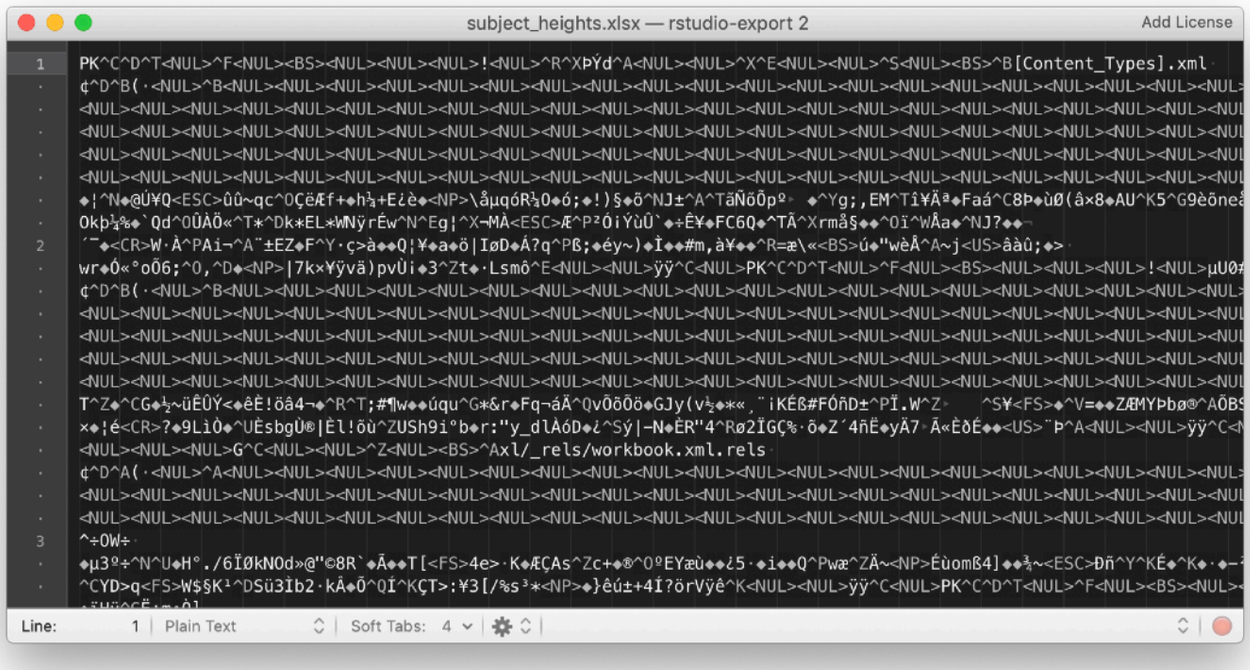
Python's binary data file type



Source

Excel's XLSX is Binary

… popular workbook format saves as a binary …



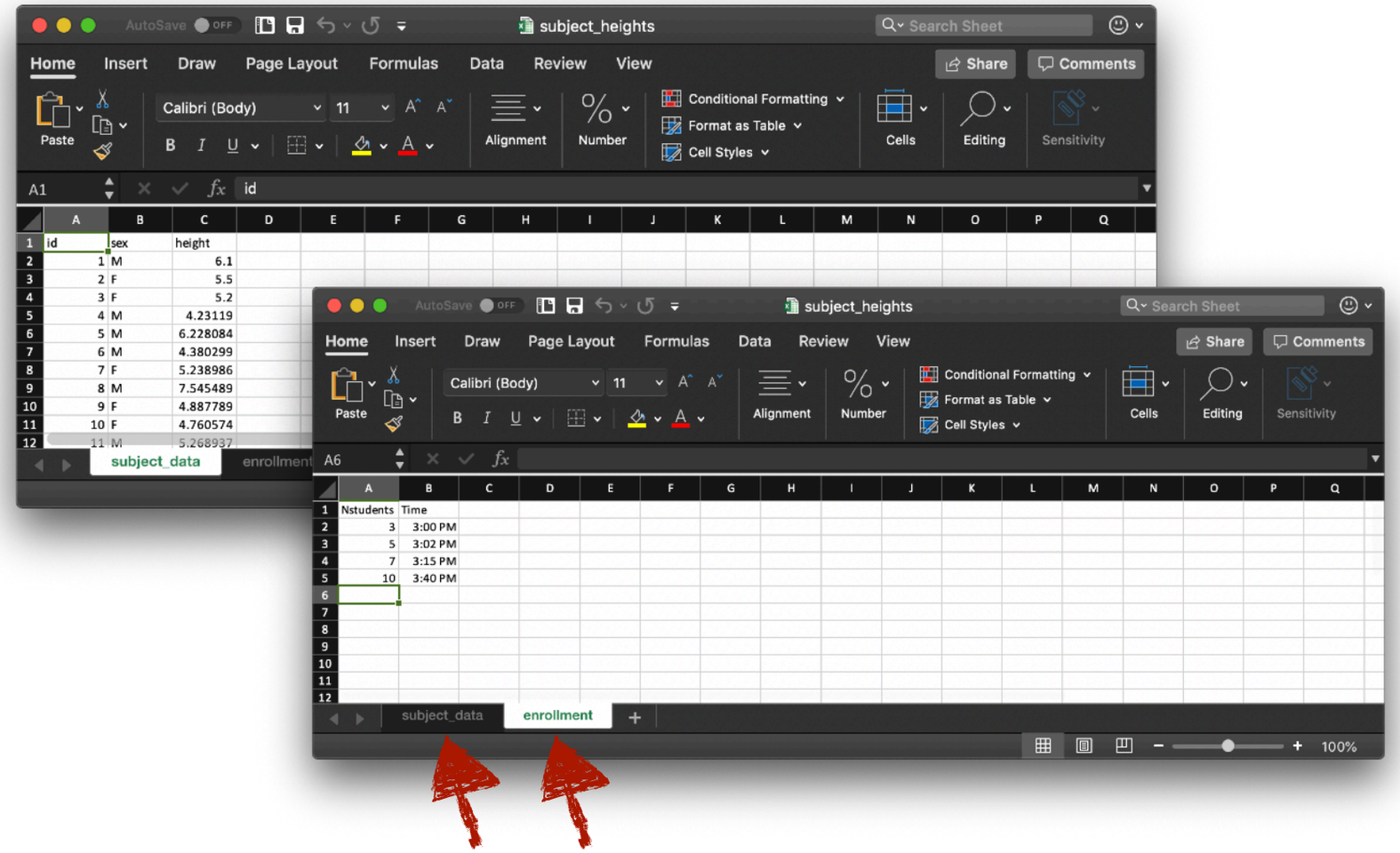
XLSX is compromised of .xml in an archive (.zip)

Excel Terminology

… a workbook may contain one or more worksheets …

Workbook

subject_heights.xlsx



Worksheets

Reading an Excel Spreadsheet

… importing an XLSX file …

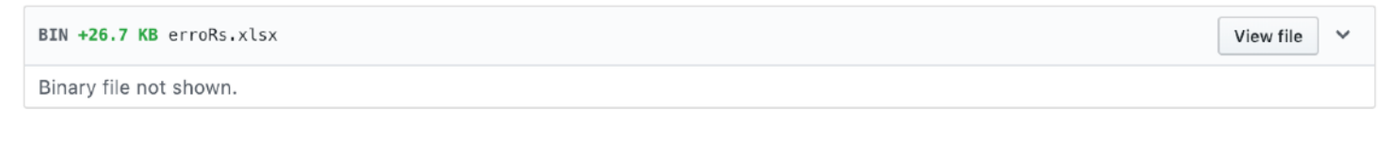
```
# Parse all worksheet
df = pandas.read_excel("subject_heights.xlsx", None);

# See worksheets available as dictionaries
df.keys()
# [u'subject_data', u'enrollment']

# Parse-per worksheet
xl_overview = pd.ExcelFile('subject_heights.xlsx')

# See all worksheets
xl_overview.sheet_names

# Obtain a specific worksheet
xl_overview.parse(sheet_name)
```

Pros vs. Cons

… when to employ each file type …

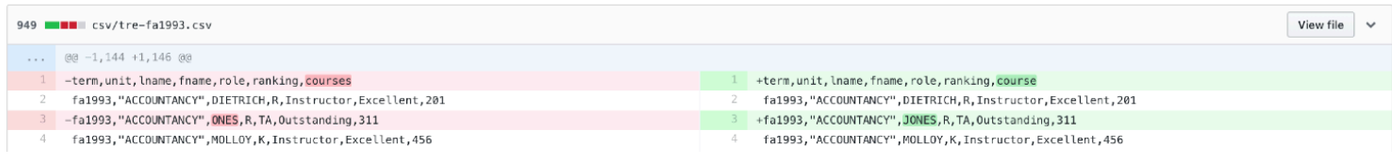
Does storage size matter?

> Binary Files



Do you need to view diffs?

> Flat Files



Missing Data

What happens if **a data point is missing**?

Definition:
Missingness indicates that no data has been recorded or was omitted. In *Python*, we denote this by ***np.nan***, which stands for "***N***ot ***a***vailable ***N***umber."

id	sex	height
1	M	6.1
2	F	5.5
3	F	5.2
…	…	…
55	M	5.9

Complete Cases

id	sex	height
1	M	6.1
2	F	*NaN*
3	*NaN*	5.2
…	…	…
55	*NaN*	*NaN*

Incomplete Cases

https://www.youtube.com/watch?v=1zwSO2uOYyI

Missingness *is*

Contagious & Propagates

```
# Operations with missingness (NaN) yield more missingness (NaN)...
np.nan + 2
# [1] nan

np.nan == 2
# [1] nan

12 - 2 + np.nan*5
# [1] nan

np.nan == np.nan
# [1] False
```

*

“An ***NaN*** is the presence of an absence. Don't forget that some missing values are the absence of a presence.”

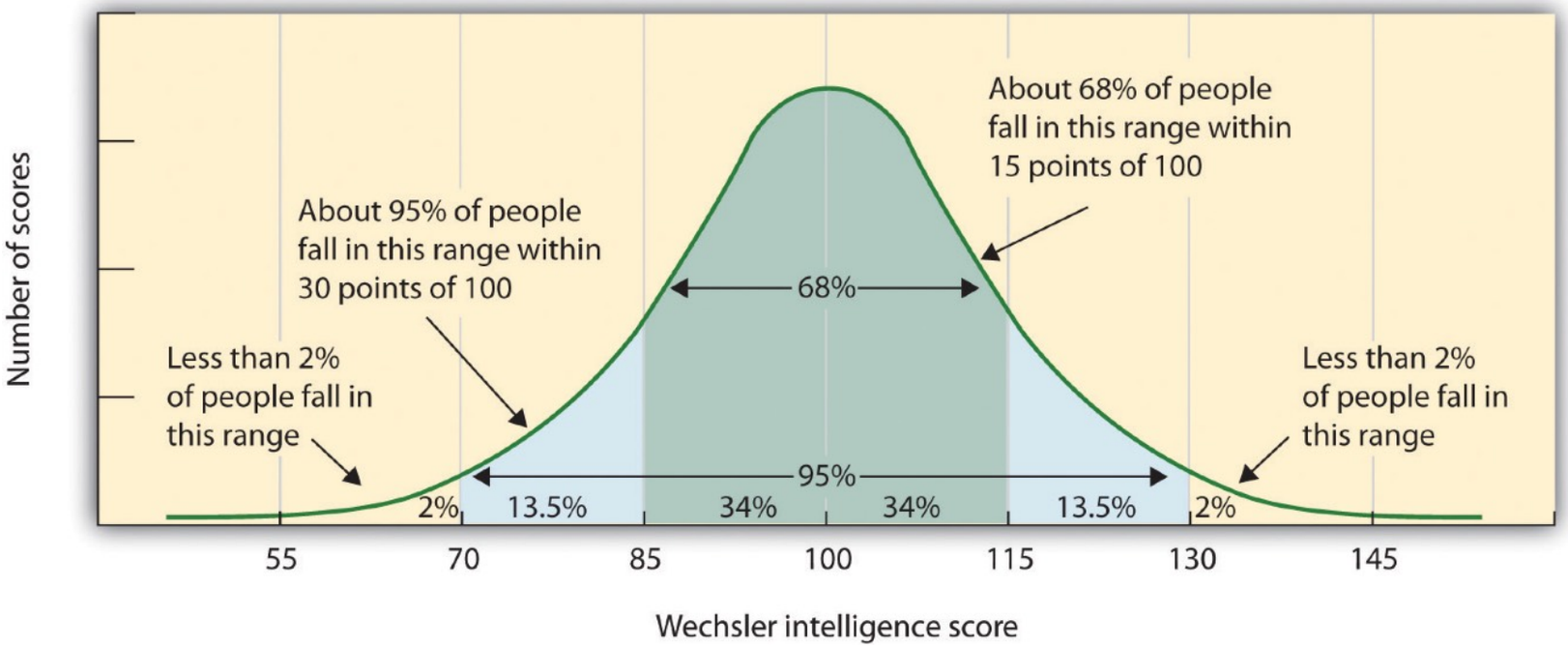
– Hadley Wickham on Twitter

Aside

IQ Example

… dealing with test data …

IQ Range	IQ Classification
130 and above	Very superior
120–129	Superior
110–119	High average
90–109	Average
80–89	Low average
70–79	Borderline
69 and below	Extremely low



Source

Types of Missingness

… got data ???

Original

Age	IQ
18	112
19	108
19	94
22	87
25	132
28	79
30	103

Missing **C**ompletely **A**t **R**andom

Age	IQ
18	*NA*
19	108
19	94
22	87
25	*NA*
28	79
30	*NA*

????

Missing **A**t **R**andom

Age	IQ
18	*NA*
19	*NA*
19	*NA*
22	87
25	132
28	79
30	103

Non-response

Missing **N**ot **A**t **R**andom

Age	IQ
18	112
19	108
19	*NA*
22	*NA*
25	132
28	*NA*
30	103

Low IQ

* **MCAR** indicates that no relationship exists between missing values and any observed values

** **MAR** indicates a relationship exists between missing values and recorded values that can be inferred.

*** **MNAR** indicates a relationship exists between the value of the missing data.

Missing Values

Note: The choice of using `NaN` internally to denote missing data was largely for simplicity and performance reasons. It differs from the MaskedArray approach of, for example, `scikits.timeseries`. We are hopeful that NumPy will soon be able to provide a native NA type solution (similar to R) performant enough to be used in pandas.

Source

```
# Check for missing values
uci_adult_df.isnull().values.any()

import numpy as np

# Create a copy
uci_adult_df_na = uci_adult_df.copy()

# Infect with a missing value
uci_adult_df_na.loc[1, ('Age')] = np.NaN

# Check for missing values in new DataFrame
uci_adult_df_na.isnull().values.any()

# Compute the median
age_median = uci_adult_df_na['Age'].median()

# Replace NA values with the median value
# for Age.
uci_adult_df_na['Age'] =
    uci_adult_df_na['Age'].fillna(age_median)

# Verify it's gone again
uci_adult_df_na.isnull().values.any()
```

Acknowledgements

Summary

- Data can be stored in a compressed or human readable format.
- Data storage formats may vary unexpectedly.
- Missing values are problematic and imputation should be used for them.

Acknowledgements

- Pandas documentation and the pandas cheatsheet by Irv Lustig

- Style of the RStudio Cheatsheet for Data Transformations