

Access to Campaign Funding Data

1

Next Quiz

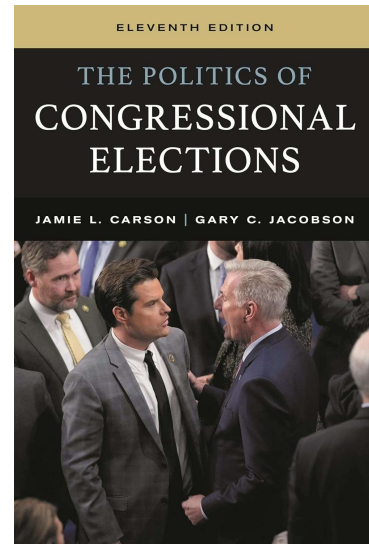
- Depending on the pace of material coverage, the first quiz will likely be on Thursday, February 18th
- Will include material (lectures and reading) covered through Thursday, February 11th
- Mainly short answer questions

2

Reading

- Carson & Jacobson – Chapter 4

You are responsible for all assigned readings.
Exam and quiz questions based on reading
material, lectures, and class discussions



3

Installing Pandas into PyCharm on Windows

- You will need to install pandas into your project to do the hands-on material today
- You need to explicitly install most libraries in PyCharm (using Conda)
- Operates differently on Windows and on Mac OS

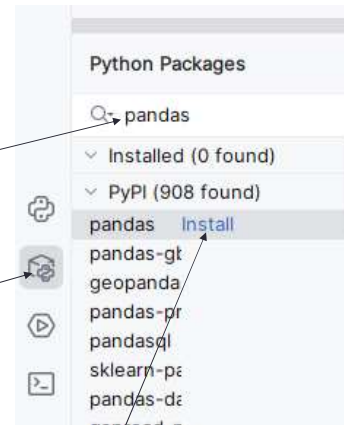
4

Select the PyCharm Packages Pane

- You select the library packages you need for your project

2. Type the name of the package you would like to install

1. Click on the Python Packages tab to open the Packages pane

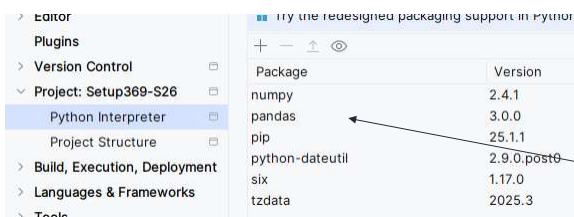


3. Click install to install

5

PyCharm Packages Pane

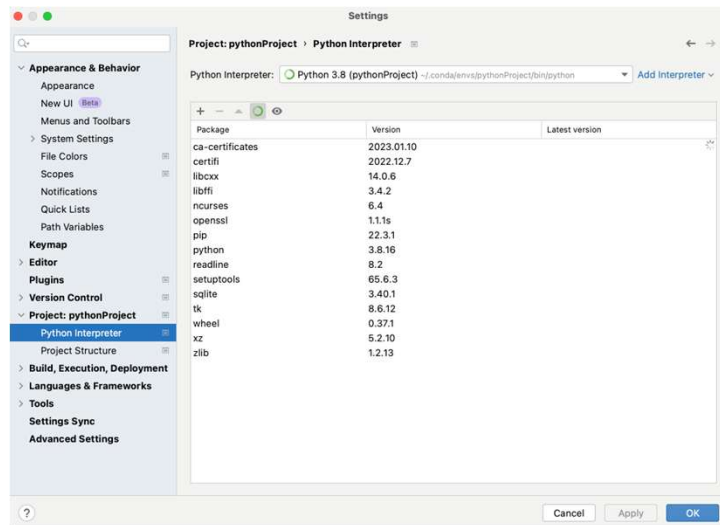
- We entered “pandas” in the search window and clicked “install” and PyCharm installed the correct libraries
- Conda ensures that related packages are installed



When we look at Settings, we see that needed packages are added to the project

6

Installing Python libraries on Mac OS ...

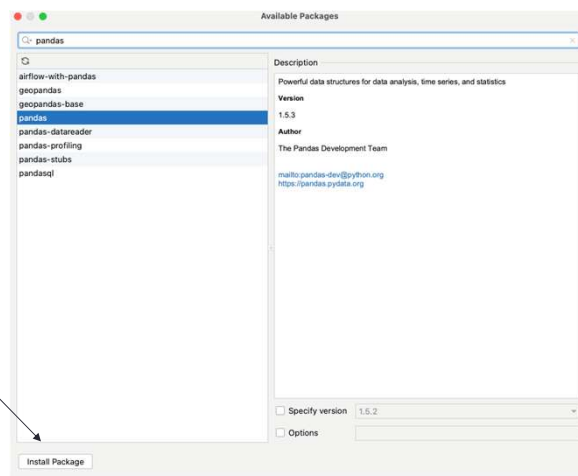


7

... Installing Python libraries on Mac OS

- Enter the desired library name in the search bar

Click Install Package



8

Reading

- Wikipedia – Federal Election Commission
https://en.wikipedia.org/wiki/Federal_Election_Commission
- Wikipedia – Campaign finance in the US
https://en.wikipedia.org/wiki/Campaign_finance_in_the_United_States

9

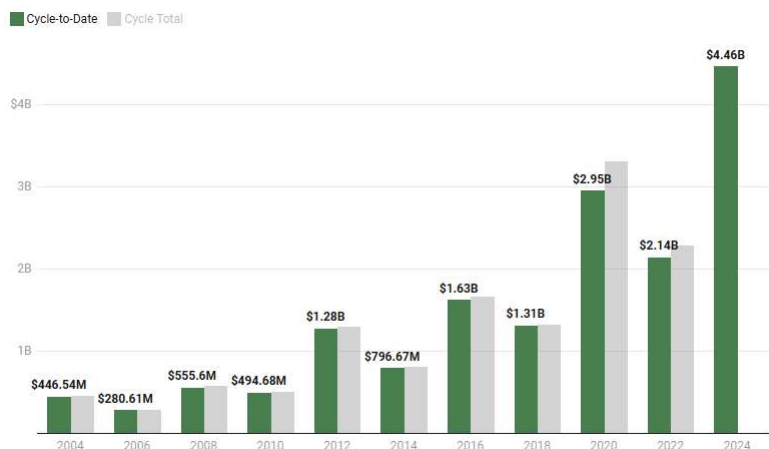
US Campaign Financing

- Financing of election campaigns by contributions primarily from
 - Individuals "big money dominates U.S. political campaigns to a degree not seen in decades" and is "drowning out the voices of ordinary Americans." – Brennan Center for Justice
 - Corporations
 - Political action committees (PACs)
- Large increases since 1990, with US House campaign expenses
 - 1990 - \$407,600
 - 2022 - \$2,790,000
 } Winners' campaign expenses
- Races for non-federal offices are governed by state and local law

What is the impact on congressional representatives when they need to solicit large contribution funds?

10

Increases in Campaign Spending



Can we analyze
all the data?

Data source: Opensecrets.com

11

Definitions

- **Campaign funds:** used with the campaign to influence the federal election of a candidate
- **Dark money:** source of the money is not disclosed to voters
- **Soft money:** not supposed to advocate the election or defeat of a federal candidate (e.g., voter registration)
- **PAC** (Political Action Committee): tax exempt organization that pools contributions from members and donates to campaigns
- **Hard money:** regulated contributions from an individual or a PAC to a federal candidate, party committee, or other PAC

We will examine hard
money data

12

PACs

- Organization registered with the Federal Election Commission
- Limits on PAC contributions to groups
 - \$5,000 to a candidate in an election
 - \$15,000 to a political party
- Can make unlimited expenditures independent of party or candidate
- Types
 - Connected – raise money from a restricted class (e.g., managers) in an organization (e.g., corporations, non-profits, labor unions)
 - Non-connected – groups with an ideological mission (e.g., climate)
 - Leadership PAC – makes “independent expenditures” not connected to a candidate
 - Super PAC – can raise unlimited amounts from individuals, corporations, unions, etc. to spend on “uncoordinated” ads, etc.

Look for PAC disclosure on TV political ads

13

Citizens United

- 2010 Supreme Court case – Citizens United v. Federal Election Commission (5-4 decision)
- Ruling - freedom of speech clause of the 1st Amendment prohibits the government from restricting expenditures for political campaigns by corporations, labor unions, etc.
- Allowed unlimited political spending by corporations and labor unions
- Ruling led to the creation of super-PACs
- Post-ruling studies have shown a subsequent Republican advantage in fund raising and election results

14

Impact of Campaign Spending

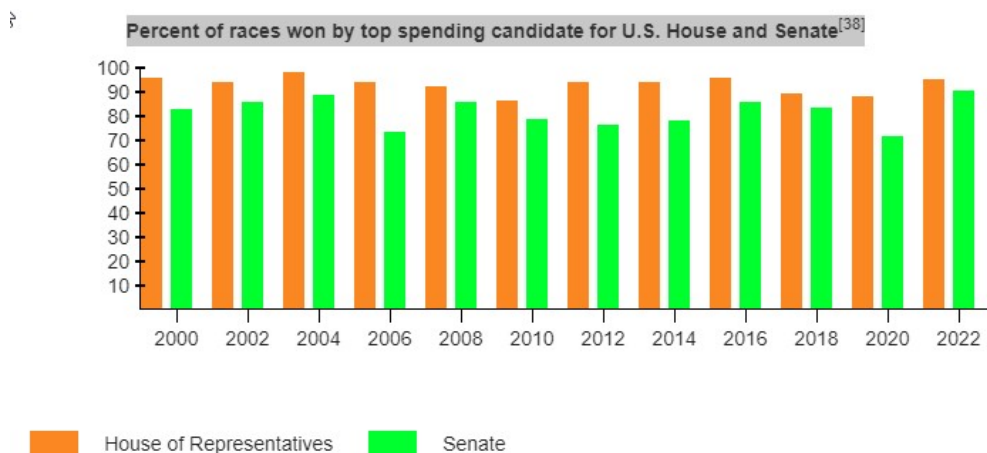


Image: Wikipedia

15

Federal Contribution Limits

- Corporations and labor unions not allowed to contribute hard money to candidates or national party committees
- Limits on most (not all) contributions
- Limits usually indexed for inflation
- Examples
 - \$3,500 limit for individual to candidate
 - \$44,300 limit for individuals to national party
 - No limit for transfers from national parties to local parties

Corporations and labor unions can contribute to PACs

16

Campaigns – What is the FEC?

- Federal Election Commission
 - Created by Congress in 1974/1975
 - Enforces campaign finance laws
- Responsible for enforcing violations of campaign finance laws
- Structure
 - Presidential appointment, Senate confirmation
 - 6 members, limited to 3 Republicans and 3 Democrats

Recent concerns about FEC effectiveness due to partisan political issues and commission structure



17

Let's Browse the FEC's Website

- Visit this link:
<https://www.fec.gov/data/>
- Class Web activity
 - Pick a state and district
 - How many candidates were there?
 - How much money did they report receiving (i.e., total receipts)?
 - How much money did they spend (i.e., total disbursements)?
 - Which House candidate is raising the most money? How much?
 - What interesting data did you find?

FIND BY STATE AND DISTRICT

New York ▼ 1 ▼ Search



18

OpenFEC

- In the previous segment, you accessed FEC campaign contribution data through a Web Interface – now we download the data, and process it with Python
- Publicly available campaign contribution data can be also be accessed through a public API



Image: NCSL

19

Library Documentation

- Python library documentation is available for all the libraries you use
- `os` library documentations is available at <https://docs.python.org/3/library/os.html>

For other libraries,
just search for API
and the library name

Link to the `os` library documentation, and get familiar with the style. Library extracts may be given to you in your exam API

```
os.chdir(path)
```

Change the current working directory to `path`.

This function can support specifying a file descriptor. The descriptor opens a file.

This function can raise `OSError` and subclasses such as `FileNotFoundError`.

Raises an auditing event `os.chdir` with argument `path`.

New in version 3.3: Added support for specifying `path` as a file object.

Changed in version 3.6: Accepts a path-like object.

20

Download Data

- We would like to access the data for all candidates in the 2023-2024 Congressional election period to determine the leading candidate fund-raisers
- Data is available at OpenFEC for download at <https://www.fec.gov/data/browse-data/?tab=bulk-data>

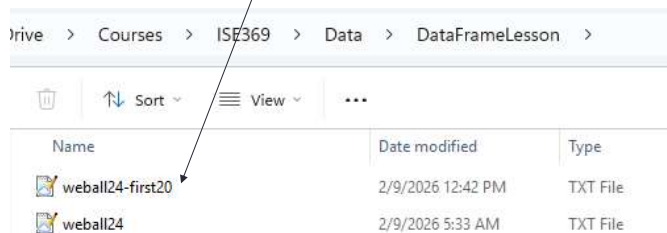
You may need some of the download files at that site for Assignment # 1

21

Are We on Track?

This will give us a small file that is easy to work with during class

- Download the All Candidates file for 2023-2024 to a folder that you can easily access (e.g., on Google Drive)
- Using a text editor, delete all but the first 20 rows
- Store the file



Name	Date modified	Type
weball24-first20	2/9/2026 12:42 PM	TXT File
weball24	2/9/2026 5:33 AM	TXT File

<https://www.fec.gov/data/browse-data/?tab=bulk-data>

22

Working Directory

- Use your existing PyCharm project or set up a new project
- To easily access the file, we first change the working directory
- You can specify file locations with an absolute address or an address relative to the current working directory

```
import os
print(os.getcwd())
os.chdir('G:/My Drive/Courses/ISE369/Data/DataFrameLesson')
print(os.getcwd())
```

Get current working directory

getcwd and chdir
are functions in
the os library

Change directory

Note the appearance
of the backslash
directory separator

```
C:\Users\robke\PycharmProjects\DataFrameLesson
G:\My Drive\Courses\ISE369\Data\DataFrameLesson
```

23

Windows

- The Windows operating system originally used a backslash as a path separator when defining a directory path (and remains the default)
- In many situations the forward slash will work
- If you use a backslash, within strings you need to also use the escape character, which is a backslash

```
import os
print(os.getcwd())
os.chdir('G:\\My Drive\\Courses\\ISE369\\Data\\DataFrameLesson')
print(os.getcwd())
```

24

Print the Downloaded File

- Print the file we just created

```
import os
os.chdir('G:/My Drive/Courses/ISE369/Data/DataFrameLesson')
with open('weball24-first20.txt') as f:
    contents = f.read()
    print(contents)
```

The Python with statement allows you to refer to an expression with a simpler identifier

```
C:\ProgramData\Anaconda3\envs\DataFrameLes
H2AK00200|CONSTANT, CHRISTOPHER|C|1|DEM|0|
H2AK01158|PELTOLA, MARY|I|1|DEM|13443537.4
H2AK00226|PALIN, SARAH|O|2|REP|3866.49|O|5
H2AK01083|BEGICH, NICHOLAS III|C|2|REP|281
H4AK00156|DAHLSTROM, NANCY|C|2|REP|996163.
H4AL01255|HOLMES, THOMAS BETHUNE MR.|C|1|D
H0AL01055|CARL, JERRY LEE, JR|I|2|REP|2246
```

25

Python

```
...
with open('weball22-first20.txt') as f:
    contents = f.read()
    print(contents)
```

f refers to the file that was opened

The “with” keyword invokes a context manager

When the block of statements in the with clause completes cleanup of the file, it is done

26

Python

- Print
 - Is a build-in function (as of Python 3)
 - Converts the argument to a string and appends a line feed
 - Allows different techniques for formatting text
 - Writes to standard output as default, but can write to a file

```
...
with open('weball122-first20.txt') as f:
    contents = f.read()
    print(contents)
```

27

Example - Data

- The file we downloaded from OpenFEC contains contribution data for all the candidates in the 2023-2024 election cycle

```
H2AK00200|CONSTANT, CHRISTOPHER|C|1|DEM|0|0|0|0|0|0|0|0|143180.09|0|AK|00|0|0|0|0|12/31/2024|0|0
H2AK01158|PELTOLA,
MARY||1|DEM|13443537.46|951851.88|14050828.27|0|691260.3|83969.49|0|0|0|0|0|10800004.13|AK|00|0|0|0|0|1615986.3|99
69.28|12/31/2024|161309.36|5625
H2AK00226|PALIN, SARAH|O|2|REP|3866.49|0|50246.07|0|46379.58|0|0|0|0|0|0|AK|00|0|0|0|0|03/31/2023|-9152.5|0
H2AK01083|BEGICH, NICHOLAS
III|C|2|REP|2810467.65|176570.41|2747371.58|17659.66|41233.99|104330.06|599|0|0|25000|0|425000|2309088.23|AK|00|0|0|
||318750|5000|12/31/2024|23031.99|0
...
```

Notice that data fields are separated by a “|”

Lines above are wrapped in your print, but each record begins on a new line

28

CSV Files

- Comma Separated Values files
- Delimited text file that uses a comma (or another symbol) to separate values
- Each line in the file is a **data record**
- Each record contains **fields**, and a comma (or some other character) separates fields
- Often denoted with a .csv file extension

29

Access an OpenFEC File

We read the file into a pandas DataFrame

```
import pandas as pd
import os
os.chdir('G:\\My Drive\\Courses\\ISE369\\Data\\DataFrameLesson')
df = pd.read_csv('weball24-first20.txt', sep='|', header=None)
print(df)
```

Widely used alias for a pandas DataFrame

Uses the pandas read_csv function to read the file into a pandas DataFrame

Specifies the separator in the csv file

Indicates that the data file did not have headers

None is a literal in Python indicating nothing is assigned

	0	1	2	...	27	28	29
0	H2AK00200	CONSTANT, CHRISTOPHER	C	...	12/31/2024	0.00	0
1	H2AK01158	PELTOLA, MARY	I	...	12/31/2024	161309.36	5625
2	H2AK00226	PALIN, SARAH	O	...	03/31/2023	-9152.50	0
3	H2AK01083	BEGICH, NICHOLAS III	C	...	12/31/2024	23031.99	0
4	H2AK00154	DAWIDSON, NANCY	C	...	12/31/2024	2018.30	0

Notice that 1) the data contains more columns than we care about and 2) the columns and row have number headers

30

DataFrame

- A 2-dimensional data structure
- Main data structure in the pandas library
- Think of it as a spreadsheet that you can easily manipulate in Python
- Contains
 - Row and column labels and
 - Data content

The data file had no headers, so pandas just assigns row and column numbers to the values

Pandas truncated view

0	1	2	...	27	28	29
H2AK00200	CONSTANT, CHRISTOPHER	C	...	12/31/2024	0.00	0
H2AK01158	PELTOLA, MARY	I	...	12/31/2024	161309.36	5625
H2AK00226	PALIN, SARAH	O	...	03/31/2023	-9152.50	0
H2AK01083	BEGICH, NICHOLAS III	C	...	12/31/2024	23031.99	0
H2AK00156	DAHLSTRÖM, MARY	C	...	12/31/2024	2018.30	0

31

Are We on Track?

- Use Google AI to learn about read_csv
- Read your downloaded, reduced file into a DataFrame
- Print the DataFrame

what is the pandas read_csv

AI Mode **All** Videos Forums Images Shopping

★ AI Overview

pandas.read_csv is a widely used, top-level function reads data from a Comma-Separated Values (CSV) file, the primary tool for data importation, capable of loading files, URLs, or file-like objects. [GeeksforGeeks +3](#)

Key Features of read_csv

- **Converts CSV to DataFrame:** It automatically converts data separated by commas, into a 2D labeled data structure.
- **Versatile Inputs:** Accepts file paths (local) or web URLs.
- **Customizable Parsing:** Allows for non-standard data formats using the sep parameter.
- **Large File Handling:** Supports chunksize to iterate over data in chunks.

32

Python

These are parameters

- Consider the Python statement in a previous slide

```
df = pd.read_csv('weball124-first20.txt', sep='|', header=None)
```

- Parameter passing
 - Methods (and functions) are named blocks of code (e.g., read_csv)
 - Python naming convention is to use lower case letters, separating words with an underscore (start function names with a verb)
 - Python uses pass-by-assignment – assignment inside a function refers to a new object representing that parameter
 - Python features keyword arguments (aka, named arguments) – example uses sep and header as named arguments, where order of named arguments is unimportant

33

Python Library - Pandas

- You can extract a Series object from a DataFrame object
- A Series object contains all or part of a single row or column, along with the heading for the elements of the row or column

	0		1	2	...	27	28	29
0	H2AK00200	CONSTANT, CHRISTOPHER	C	...	12/31/2024	0.00	0	
1	H2AK01158	PELTOLA, MARY	I	...	12/31/2024	161309.36	5625	
2	H2AK00226	PALIN, SARAH	0	...	03/31/2023	-9152.50	0	
3	H2AK01083	BEGICH, NICHOLAS III	C	...	12/31/2024	23031.99	0	
4	H2AK00156	NASHISTROM, MARY	C	...	12/31/2024	2218.30	0	

Series object

34

File Structure

- Most datasets we will access are difficult to understand by looking at it – so there is usually a document that describes it
- Layout of the All Candidates data is available at <https://www.fec.gov/campaign-finance-data/all-candidates-file-description/>

All candidates file format

Column name	Field name	Position	Null	Data type	Description
CAND_ID	Candidate identification	1	N	VARCHAR2 (9)	
CAND_NAME	Candidate name	2	Y	VARCHAR2(200)	
CAND_ICI	Incumbent challenger status	3	Y	VARCHAR2(1)	
PTY_CD	Party code	4	Y	VARCHAR2(1)	

35

Are We on Track?

- Locate the file format at <https://www.fec.gov/campaign-finance-data/all-candidates-file-description/>
- Can you identify contributions in a given year, state, and congressional district? How?

All candidates file format

Column name	Field name	Position	Null	Data type	Description
CAND_ID	Candidate identification	1	N	VARCHAR2 (9)	
CAND_NAME	Candidate name	2	Y	VARCHAR2(200)	
CAND_ICI	Incumbent challenger status	3	Y	VARCHAR2(1)	
PTY_CD	Party code	4	Y	VARCHAR2(1)	

36

Let's Explore the DataFrame We Created

- We can look for DataFrame attributes at <https://pandas.pydata.org/docs/reference/frame.html>

```
print('df index \n', df.index)
print('DataFrame size = ', df.size)
print(df.iat[3,1])
```

Line feed

df index
RangeIndex(start=0, stop=20, step=1)
DataFrame size = 600
BEGICH, NICHOLAS III

Indexing operator

These are properties of the DataFrame object that we access with df.propertyName

20 rows and 30 elements per row

Element at [3,1]

37

Python

- Bracket operator (subscript operator)
- Used to access the elements in a string, list, etc.
- Indexed left to right, starting at 0

```
print(df.iat[3,1])
```

More on iat in a little while

		(Are We on Track?)	1	2
0	H2AK00200	CONSTANT, CHRISTOPHER	C	
1	H2AK01158	PELTOLA, MARY	I	
2	H2AK00226	PALIN, SARAH	O	
3	H2AK01083	BEGICH, NICHOLAS III	C	
4	H4AK00156	BAHLSTROM, NANCY	C	
5	H4AK01255	HOLMES, THOMAS BETHUNE MR	C	

38

Remove Columns from a DataFrame

- Usually, data needs to be cleaned up before processing
- Two clean-up tasks we do with the candidate contribution data are
 - Remove any columns that are not needed
 - Relabel the remaining columns
- For now, we only need
 - Candidate id
 - Candidate name
 - Incumbent / challenger status
 - Party code
 - Party affiliation
 - Total receipts
 - Candidate state
 - Primary election status
 - General election status

39

Remove Unneeded Columns

- We can drop columns we don't need, either in-place or by assigning the DataFrame to a new DataFrame

```
df1=df.drop(columns=[3,6,7,8,9,10,11,12,13,14,15,16,17,19,20,21,22,24,25,26,27,28,29])
```

	0		1	2	4		5	18	23
0	H2AK00200	CONSTANT, CHRISTOPHER	C	DEM			0.00	AK	NaN
1	H2AK01158	PELTOLA, MARY	I	DEM	13443537.46			AK	NaN
2	H2AK00226	PALIN, SARAH	O	REP	3866.49			AK	NaN
3	H2AK01083	BEGICH, NICHOLAS III	C	REP	2810467.65			AK	NaN
4	H4AK00156	DAHLSTROM, NANCY	C	REP	996163.60			AK	NaN
5	H4AK01055	HOLMES, THOMAS BETHUNE MD	C	DEM	17600.06			AL	NaN

40

Print All the Columns

- If we saw the **truncated view** but would like to see all the columns, we can allow a row to print on multiple lines

```
pd.set_option('max_columns', None)
print(df)
```

```

0      H2AK00200  CONSTANT, CHRISTOPHER C  1  DEM      0.00      0.00 \
1      H2AK01158      PELTOLA, MARY I  1  DEM  15443537.46  953851.88
2      H2AK00226      PALIN, SARAH D  2  REP      3866.49      0.00
3      H2AK01083      BEGICH, NICHOLAS III C  2  REP  2810467.05  176570.41
4      H4AK00156      DAHLSTROM, NANCY C  2  REP  990165.00  435712.04
5      H0AL01255 HOLMES, THOMAS BETHUNE MR. C  1  DEM   17090.36      0.00
6      H0AL01055      CARL, JERRY LEE, JR I  2  REP  2246039.19  547807.76
7      H8AL02171      MOORE, FELIX BARRY I  2  REP  1130187.30      0.00
8      H0AL01097      AVERHART, JAMES C  1  DEM    8876.56      0.00
9      H0AL02202      HARVEY-HALL, PHYLLIS C  1  DEM   16284.00      0.00
10     H2AL02166      PATEL, VINAL C  1  DEM   16300.00      0.00
11     H0AL00997      COLEMAN, MERIKA C  1  DEM  140300.80      0.00
12     H4AL02113      GRAY, JEREMY C  1  DEM  163092.80      0.00
13     H4AL02162      DANIELS, ANTHONY C  1  DEM   567548.46      0.00
14     H4AL02170      FIGURES, SHOMARI C. C  1  DEM  2628628.73  104915.46
15     H4AL02196      GIVAN, JUANDALYNN HS. C  1  DEM  150251.27      0.00
16     H4AL02212      LENARD, WILLIE JAMES C  1  DEM   53721.00      0.00
17     H4AL02246      BRACY, NAPOLEON C  1  DEM  202592.38      0.00
18     H4AL02121      BREWBAKER, DICK LANSDEN C  2  REP  2129358.39      0.00
19     H4AL02139      HARRIS, HAMPTON SCOTT C  2  REP    58137.92      0.00

      7      8      9      10      11      12 13 \
0      0.00      0.00      0.00      0.00      0.00      0.00 0
1  14050828.27      0.00  491260.30  83969.49      0.00      0.00 0
2    50246.07      0.00  46379.58      0.00      0.00      0.00 0
```

In truncate view, pandas will detect the width of the terminal and print a center truncated object that fits the screen width

41

Python

- Explanation about the Python code on a previous slide

```
df1=df.drop(columns=[3,6,7,8,9,10,11,12,13,14,15,16,17,19,20,21,22,24,25,26,27,28,29])
```

- A Python list
 - Can contain items of a different data type
 - Contains values, separated by commas (,) and enclosed within square brackets ([])
 - Is changable

42

Remove Unneeded Columns

- We can also drop columns by selecting the columns we would like to retain and assign it to a different DataFrame

```
df2=df.loc[:, [0,1,2,4,5,18,23]]
print(df2)
```

Drop removes the columns, so we assign to a new DataFrame to preserve the original

	0		1	2	4		5	18	23
0	H2AK00200	CONSTANT, CHRISTOPHER	C	DEM			0.00	AK	NaN
1	H2AK01158	PELTOLA, MARY	I	DEM	13443537.46			AK	NaN
2	H2AK00226	PALIN, SARAH	O	REP	3866.49			AK	NaN
3	H2AK01083	BEGICH, NICHOLAS III	C	REP	2810467.65			AK	NaN
4	H2AK00156	DAHLSTRÖM, MANDY	C	REP	996163.60			AK	NaN

43

Python

Selects all the rows

- Consider the Python statement in a previous slide

```
df2=df.loc[:, [0,1,2,4,5,18,23]]
```

- Python slice operation
 - Operates on strings, lists, and tuples
 - Selects a portion of the string or list
 - Indices, start, stop, and step (stops one before stop index)
 - Syntax – [start : stop : step]
 - Must include at least one colon (:)
 - Start index (inclusive)
 - Stop index (exclusive)
 - Step (optional)

44

Pandas loc

- Consider the Python statement in a previous slide

```
df2=df.loc[:, [0,1,2,4,5,18,23]]
```

All rows

List of column labels

- Allows you to access a group of rows and columns from a DataFrame using either access labels (loc) or integers (iloc)
- Selection can use a list of labels, a slice, or a Boolean array

45

Accessing Data in a DataFrame

Multiple approaches differ in what you access and what the property returns

- iloc**
 - Integer based indexing for selection by integer position
 - Allowed inputs include an integer, a list or array of integers, and a slice object
- loc**
 - Access a group of rows and columns by labels
 - Allowed inputs include a single label, a list or array of labels, and a slice object with labels
- iat**
 - accesses a single value for a row/column by integer position
 - Use iat if you only need to get or set a single value
- at**
 - Similar to iat, but access a single value by label pair

46

Rename the Columns

- We can rename the columns in the DataFrame

```
df1 = df1.rename(columns = {0:"ID"})
df1 = df1.rename(columns = {1:"Name"})
df1 = df1.rename(columns = {2:"I/C"})
df1 = df1.rename(columns = {4:"Pty"})
df1 = df1.rename(columns = {5:"$$"})
df1 = df1.rename(columns = {18:"State"})
df1 = df1.rename(columns = {23:"Status"})
print(df1)
```

Python
dictionary
syntax

NaN stands for Not
a Number in Python.
Often used to
represent missing
values

	ID	Name	I/C	Pty	\$\$	State	Status
0	H2AK00200	CONSTANT, CHRISTOPHER	C	DEM	0.00	AK	NaN
1	H2AK01158	PELTOLA, MARY	I	DEM	13443537.46	AK	NaN
2	H2AK00226	PALIN, SARAH	O	REP	3866.49	AK	NaN
3	H2AK01083	BEGICH, NICHOLAS III	C	REP	2810467.65	AK	NaN
4	H4AK00156	DAHLSTROM, NANCY	C	REP	996163.60	AK	NaN
5	H4AL01255	HOLMES, THOMAS BETHUNE	M	DEM	17608.84	AL	NaN

47

Sort the DataFrame

- We can sort the rows in the DataFrame

```
dfsort = df1.sort_values('$$', ascending=False)
print(dfsort)
```

	ID	Name	I/C	Pty	\$\$	State	Status
1	H2AK01158	PELTOLA, MARY	I	DEM	13443537.46	AK	NaN
3	H2AK01083	BEGICH, NICHOLAS III	C	REP	2810467.65	AK	NaN
14	H4AL02170	FIGURES, SHOMARI C.	C	DEM	2628628.73	AL	NaN
6	H0AL01055	CARL, JERRY LEE, JR	I	REP	2246839.19	AL	NaN
18	H4AL02121	BREWER, DICK LANSFORD	C	REP	2129338.30	AL	NaN

But numbers in the \$\$
display are difficult to read

48

Format

- We would like to
 - Change the format of the numbers in the \$\$ column and
 - Print the top 25 rows (if our input file contained more than 25 rows)

Format method of the string object

```
def format(x):
    return "${:.2f}M".format(x/1000000)
dfsort['$$'] = dfsort['$$'].apply(format)
print("Top 25")
print(dfsort.head(25))
```

Positional argument

Format specifier
[\(https://www.python.org/dev/peps/pep-3101/\)](https://www.python.org/dev/peps/pep-3101/)

Replacement fields are enclosed in curly braces. Text outside the curly braces are included, as is. Replacement fields are substituted for by the arguments to the format() method

49

Are We on Track?

- Use your downloaded full list of candidate's campaign contributions for 2023-2024
- Don't show ID or I/C columns
- Display the top 25 donations in excess of \$2M

		Name	Pty	\$\$	State	Status
3104	I	BIDEN, JOSEPH R JR	DEM	\$1175.19M	00	NaN
3080		HARRIS, KAMALA	DEM	\$1175.19M	00	NaN
3666		BROWN, SHERROD	DEM	\$96.53M	0H	NaN
3756		ALLRED, COLIN	DEM	\$94.67M	TX	NaN
3543		TESTER, R. JON	DEM	\$93.57M	MT	NaN
3766		CRUZ, RAFAEL EDWARD	TED	\$74.05M	TX	NaN
3182	KENNEDY, ROBERT, F. JR.,	SHANAHAN, NICOLE	IND	\$67.28M	00	NaN
3123		PANASWAMY	VTUEK	\$66.28M	00	NaN

50