NYU Center for Data Science

DS-UA 112
Introduction to Data Science

Week 3: Lecture 2

Tables – Arranging Data in Rows and Columns

Center for
Data Science

NYU

How can tables help us to summarize data?

DS-UA 112
Introduction to Data Science

Week 3: Lecture 2

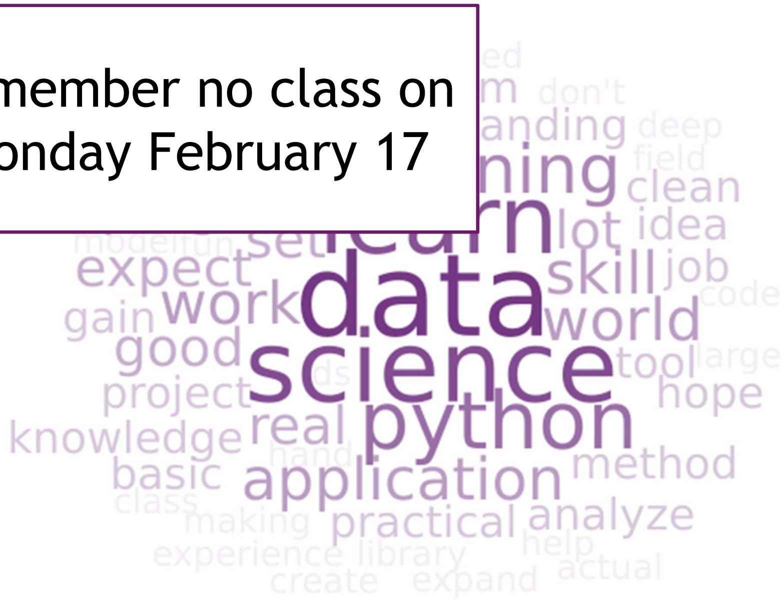
Tables – Arranging Data in Rows and Columns

Adapted from DeNero and Hug

Announcements

- Please check Week 3 agenda on NYU Classes
  - Homework 1
  - Lab 3
  - Grader Office Hours
- Remember to post to Piazza

Remember no class on Monday February 17



Review

Rules for Determining the Chance of Events

$$0 \leq P(\text{an event happens}) \leq 1$$

- Probability is a number that reflects the likelihood of events
  - 0 least likely
  - 1 most likely

- Two events are complementary when one or the other must occur but they cannot occur together
  - Complementary events have related probabilities

$$\boxed{P(\text{an event doesn't happen}) = 1 - P(\text{an event happens})}$$

Review

Rules for Determining the Chance of Events

- We can use multiplication to determine the chance that two events happened together

$$P(n \text{ and } y) = P(n|y) \cdot P(y)$$

$$\boxed{P(n|y) = P(n)}$$

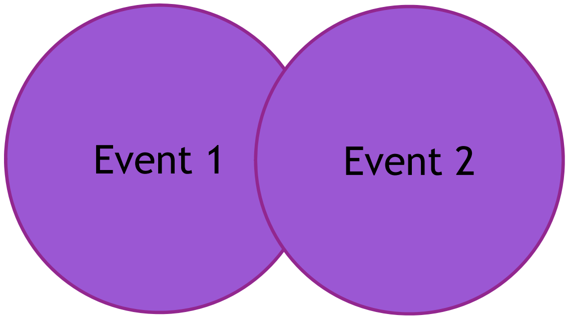
- When two events are unrelated to each other, we have independent events

Review

Rules for Determining the Chance of Events

- We can use addition to determine the chance that two events happened separately

$$P(n \text{ or } y) = P(n) + P(y) - P(n \text{ and } y)$$



- When two events can happen together, we must remember to subtract to avoid counting the probabilities twice

Review

Rules for Determining the Chance of Events

- Sometimes we want to switch the order we take the events. Bayes rule helps us to rearrange

$$\begin{aligned}\frac{P(n|y)P(y)}{P(n)} &= \frac{P(n \text{ and } y)}{P(y)}\frac{P(y)}{P(n)}\\ &= \frac{P(n \text{ and } y)}{P(n)}\\ &= P(y|n)\end{aligned}$$

$$P(y|n) = \frac{P(n|y)P(y)}{P\left((n \text{ and } 1880) \text{ or } (n \text{ and } 1881)\ldots\right)}$$

- We can expand the denominator of the left hand side with complementary events

Review

Rules for Determining the Chance of Events

- Sometimes we want to switch the order we take the events. Bayes rule helps us to rearrange

$$\frac{P(n|y)P(y)}{P(n)} = \frac{P(n \text{ and } y)}{P(y)} \frac{P(y)}{P(n)}$$
$$= \frac{P(n \text{ and } y)}{P(n)}$$
$$= P(y|n)$$

$$\frac{P(n|y)P(y)}{\ldots 1881)\ldots)}$$

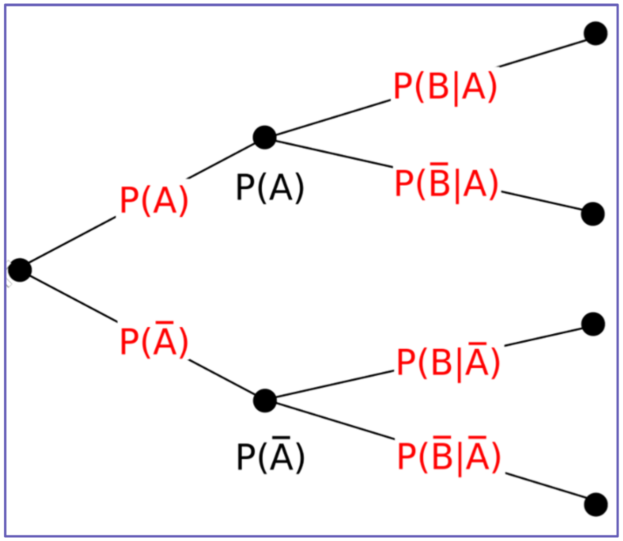
$$P(n \text{ and } 1880) + P(n \text{ and } 1881) + \ldots$$
$$= P(n|1880)P(1880) + P(n|1881)P(1881) + \ldots$$

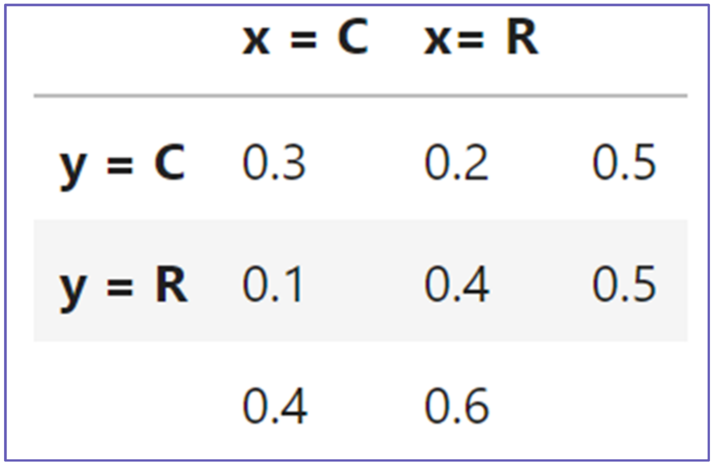
- We can expand the denominator of the left hand side with complementary events

Review

Rules for Determining the Chance of Events

- Trees are graphs that help us keep track of conditional probabilities.





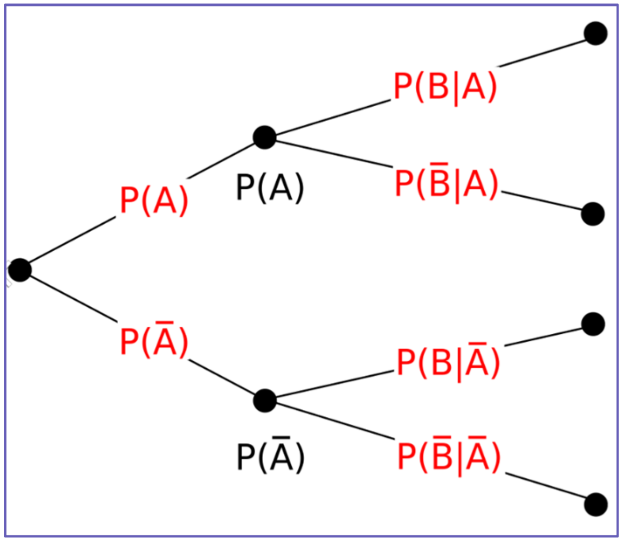
	x = C	x= R	
y = C	0.3	0.2	0.5
y = R	0.1	0.4	0.5
	0.4	0.6	

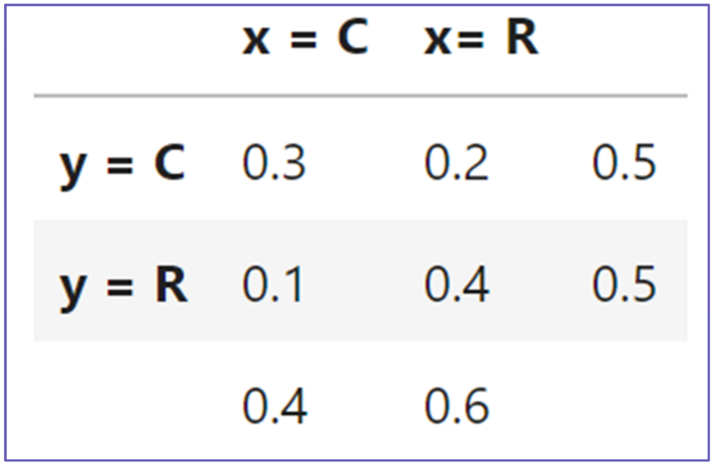
- Tables are rows and columns of numbers that help us keep track of different outcomes for two events

Review

Rules for Determining the Chance of Events

- Trees are graphs that help us keep track of conditional probabilities.





	x = C	x= R	
y = C	0.3	0.2	0.5
y = R	0.1	0.4	0.5
	0.4	0.6	

- Tables are rows and columns of numbers that help us keep track of different outcomes for two events

Agenda

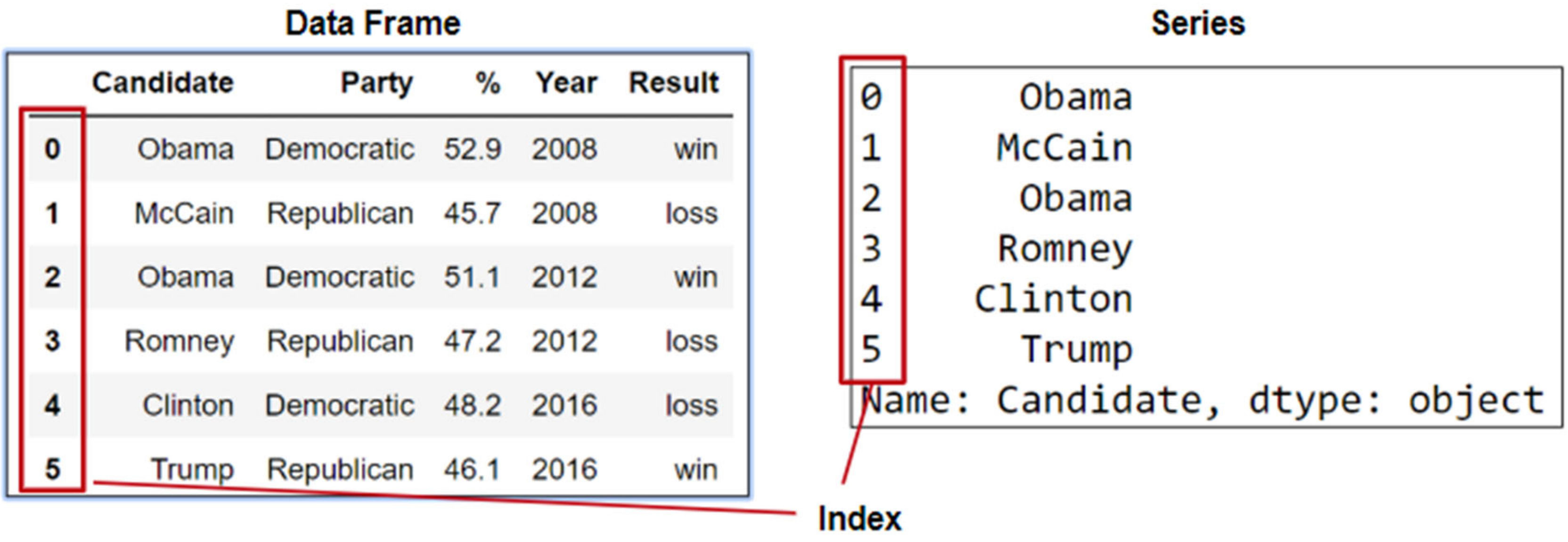
- Summaries of Numbers
  - Average, Expected Value
  - Standard Deviation
- Tables
  - Using the pandas package to manipulate tables
  - Series and Data Frames
  - Indexing (with [], loc, and iloc)
  - Averaging, Sorting, and Removing Duplicates

References

- Nolan, Lau, Gonzalez (Chapter 3.2)
- Shah (Chapter 5.1-5.4)

Tables

- pandas is a package for accessing and modifying tabular data
- pandas stores data in three formats
  - Data Frame: 2D data.
  - Series: 1D data.
  - Index: collection of labels.



Data Frame

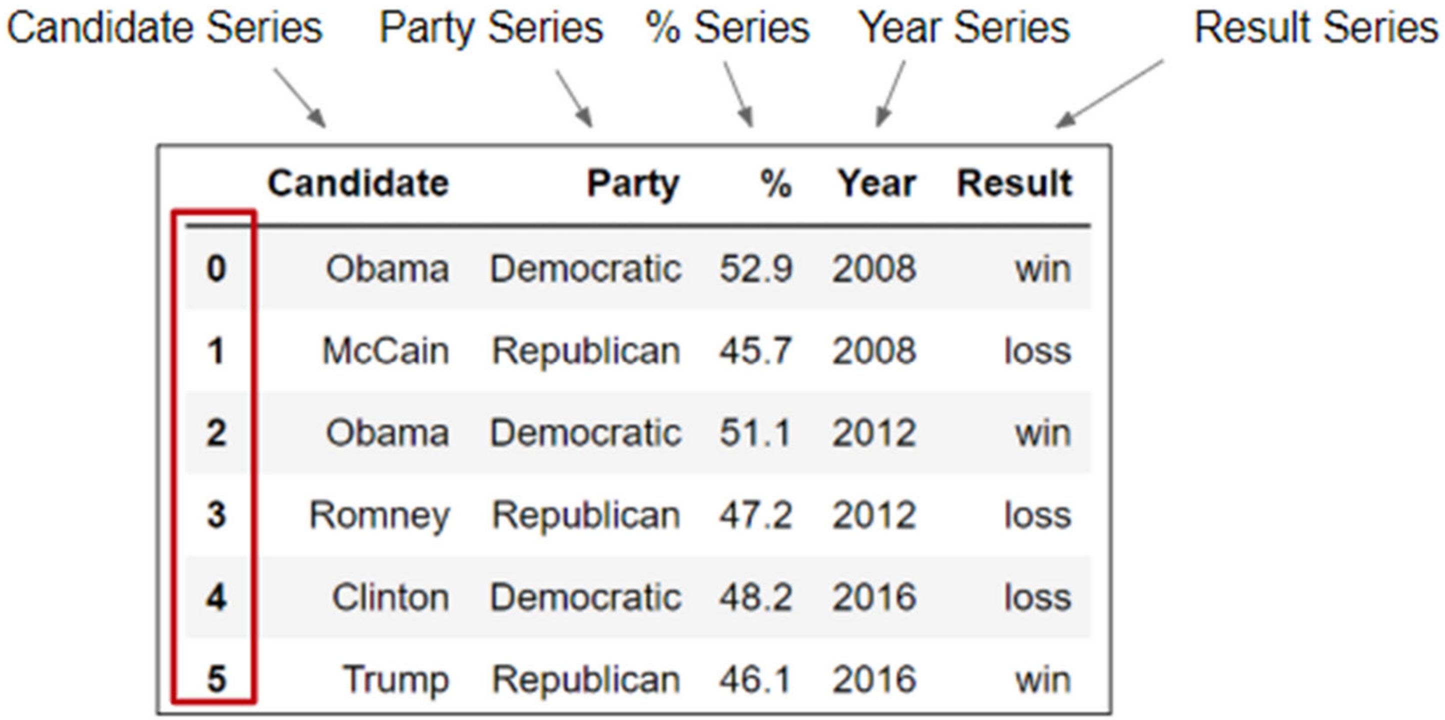
	Candidate	Party	%	Year	Result
0	Obama	Democratic	52.9	2008	win
1	McCain	Republican	45.7	2008	loss
2	Obama	Democratic	51.1	2012	win
3	Romney	Republican	47.2	2012	loss
4	Clinton	Democratic	48.2	2016	loss
5	Trump	Republican	46.1	2016	win

Series

```
0          Obama
1         McCain
2          Obama
3         Romney
4        Clinton
5          Trump
Name: Candidate, dtype: object
```

Tables

- Data Frame is a collection of Series with the same Index



Candidate Series

Party Series

% Series

Year Series

Result Series

	Candidate	Party	%	Year	Result
0	Obama	Democratic	52.9	2008	win
1	McCain	Republican	45.7	2008	loss
2	Obama	Democratic	51.1	2012	win
3	Romney	Republican	47.2	2012	loss
4	Clinton	Democratic	48.2	2016	loss
5	Trump	Republican	46.1	2016	win

Tables

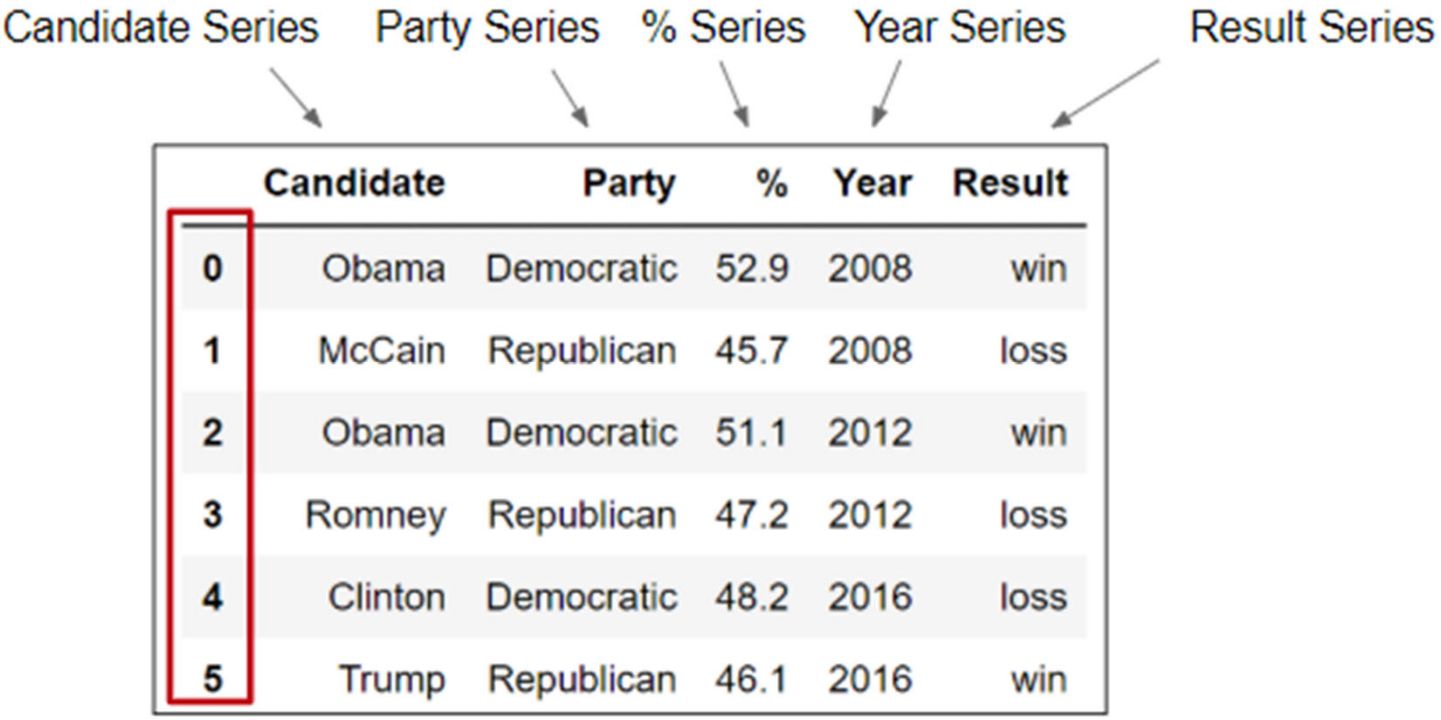
- Indices may not be numbers
  - Collection of strings

  “Candidate1”, “Candidate2”, …

- Indices may not be unique
  - Indices are just labels of the rows

  “CandidateDemocrat”, “CandidateRepublican”,

  “CandidateDemocrat” …



Candidate Series

Party Series

% Series

Year Series

Result Series

	Candidate	Party	%	Year	Result
0	Obama	Democratic	52.9	2008	win
1	McCain	Republican	45.7	2008	loss
2	Obama	Democratic	51.1	2012	win
3	Romney	Republican	47.2	2012	loss
4	Clinton	Democratic	48.2	2016	loss
5	Trump	Republican	46.1	2016	win

Tables

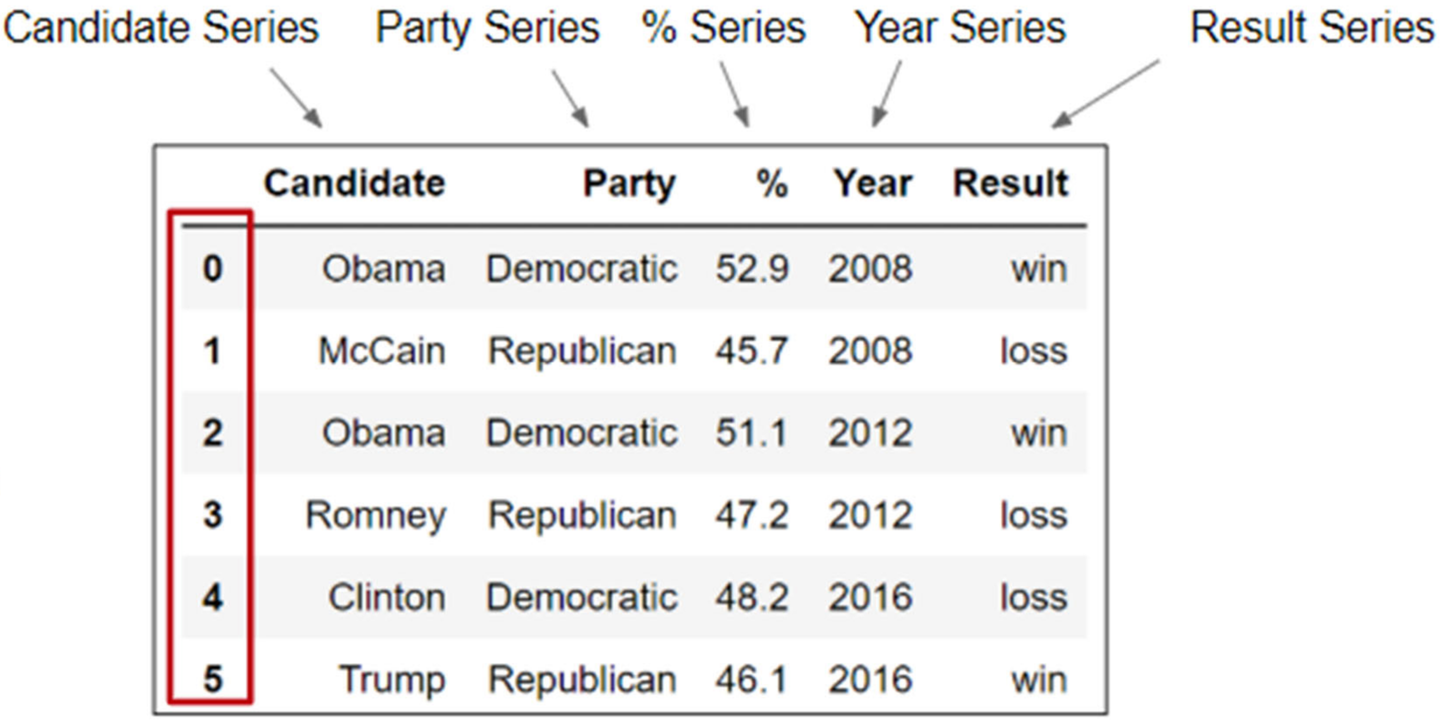
However columns are unique

- Indices may not be numbers
  - Collection of strings

  “Candidate1”, “Candidate2”, …
- Indices may not be unique
  - Indices are just labels of the rows

  “CandidateDemocrat”, “CandidateRepublican”,

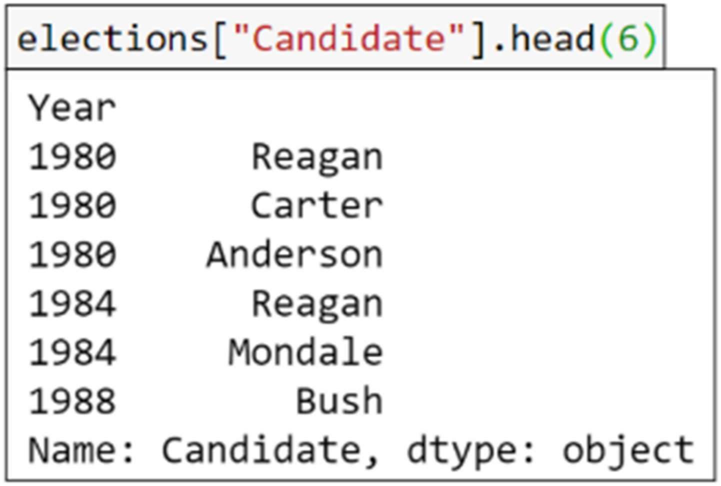
  “CandidateDemocrat” …



	Candidate	Party	%	Year	Result
0	Obama	Democratic	52.9	2008	win
1	McCain	Republican	45.7	2008	loss
2	Obama	Democratic	51.1	2012	win
3	Romney	Republican	47.2	2012	loss
4	Clinton	Democratic	48.2	2016	loss
5	Trump	Republican	46.1	2016	win

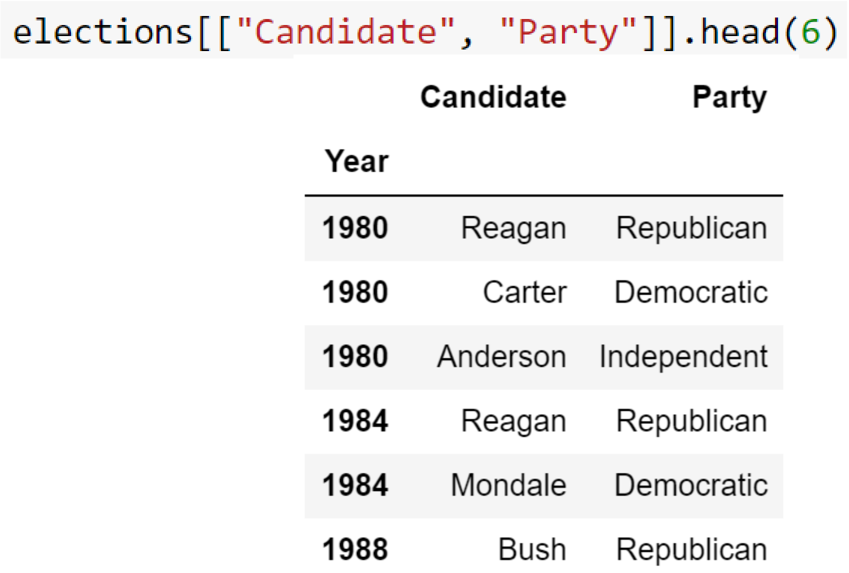
Tables

- Select column to extract Series or collection of Series from Data Frame



```
elections["Candidate"].head(6)
```

```
Year
1980        Reagan
1980        Carter
1980      Anderson
1984        Reagan
1984       Mondale
1988          Bush
Name: Candidate, dtype: object
```



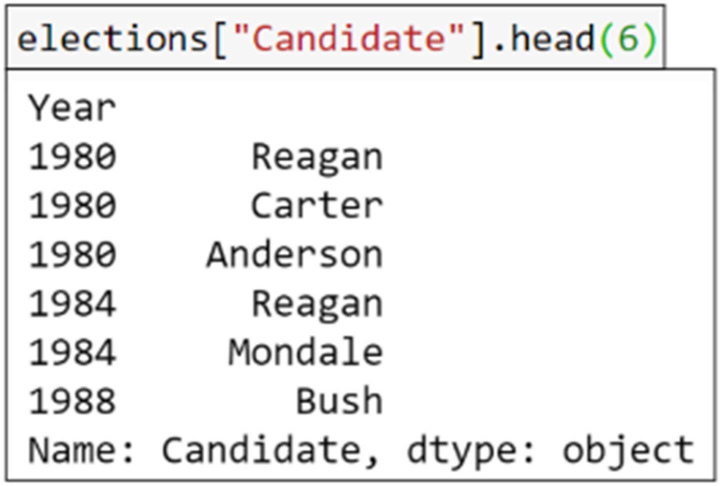
```
elections[["Candidate", "Party"]].head(6)
```

Year	Candidate	Party
1980	Reagan	Republican
1980	Carter	Democratic
1980	Anderson	Independent
1984	Reagan	Republican
1984	Mondale	Democratic
1988	Bush	Republican

- Passing a name to [] gives us a Series
- Passing a List to [] gives us a Data Frame

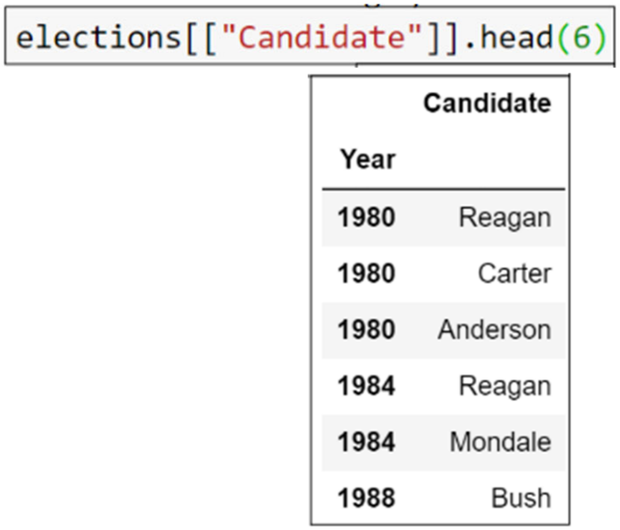
Tables

- Select column to extract Series or collection of Series from Data Frame



```
elections["Candidate"].head(6)
```

```
Year
1980       Reagan
1980       Carter
1980     Anderson
1984       Reagan
1984      Mondale
1988         Bush
Name: Candidate, dtype: object
```



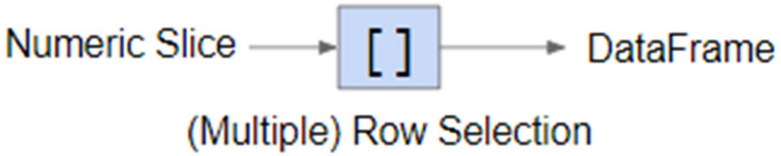
```
elections[["Candidate"]].head(6)
```

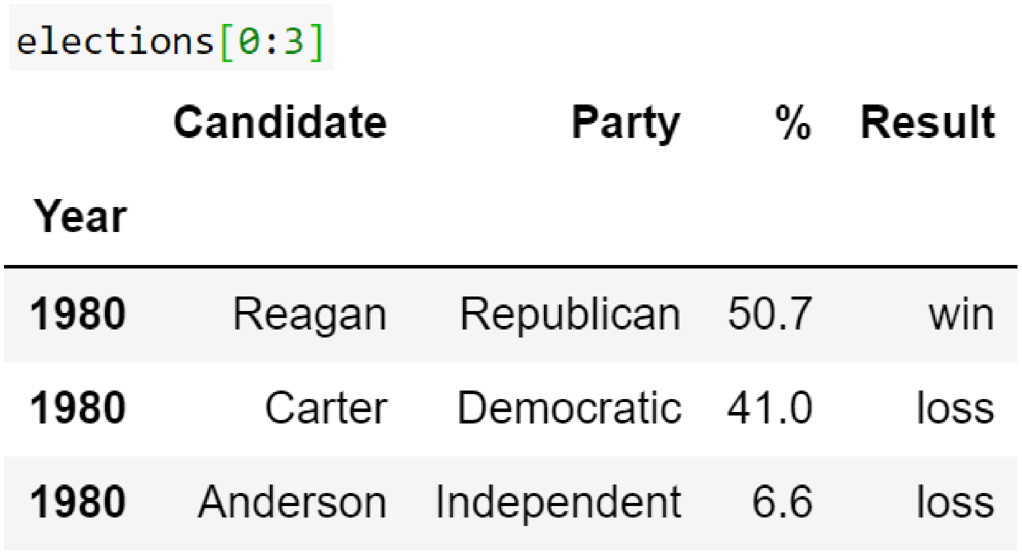
	Candidate
Year	
1980	Reagan
1980	Carter
1980	Anderson
1984	Reagan
1984	Mondale
1988	Bush

- Passing a name to [] gives us a Series
- Passing a List to [] gives us a Data Frame

Tables

- Select row to extract Data Frame consisting of adjacent rows





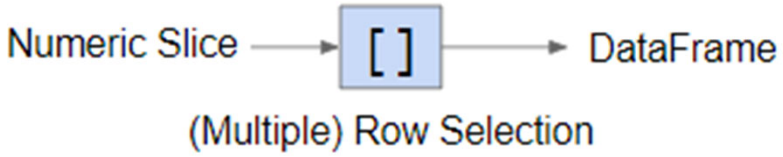
```
elections[0:3]
```

	Candidate	Party	%	Result
Year				
1980	Reagan	Republican	50.7	win
1980	Carter	Democratic	41.0	loss
1980	Anderson	Independent	6.6	loss

- Note that you must indicate adjacent rows with a numeric range such as 0:3 for 0,1,2

Tables

- Select row to extract Data Frame consisting of adjacent rows



```
elections[0:3]
```

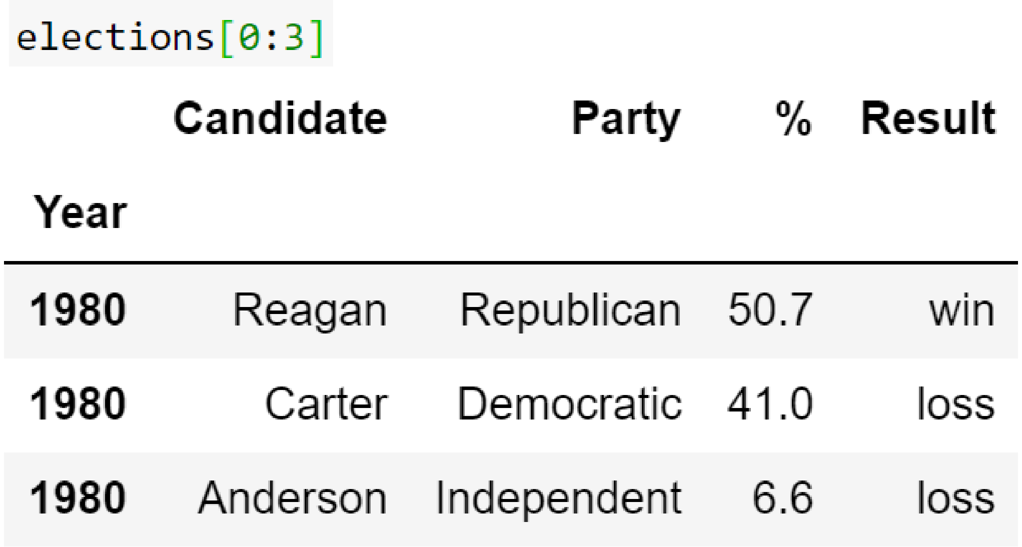
	Candidate	Party	%	Result
Year				
1980	Reagan	Republican	50.7	win
1980	Carter	Democratic	41.0	loss
1980	Anderson	Independent	6.6	loss

```
elections[0]
```

[illegible]te that you must [illegible]cent rows wit[illegible]c range such as 0:[illegible] for 0,1,2

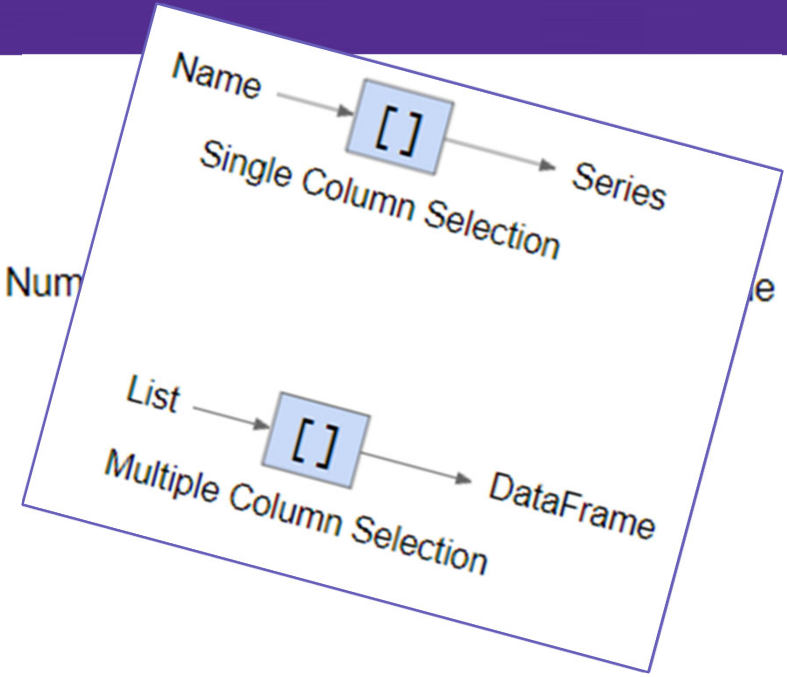
Tables

- Select row to extract Data Frame consisting of adjacent rows



```
elections[0:3]
```

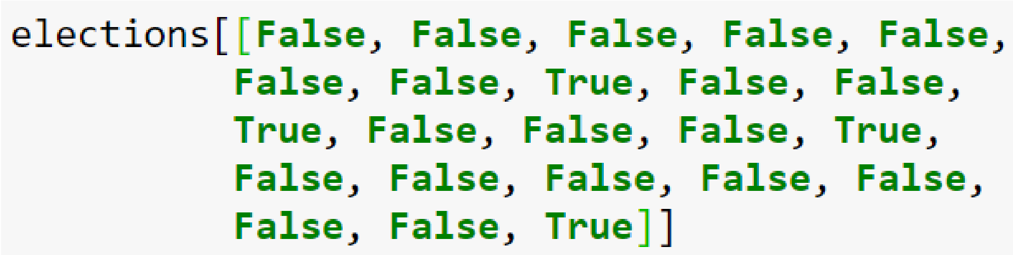
	Candidate	Party	%	Result
Year				
1980	Reagan	Republican	50.7	win
1980	Carter	Democratic	41.0	loss
1980	Anderson	Independent	6.6	loss



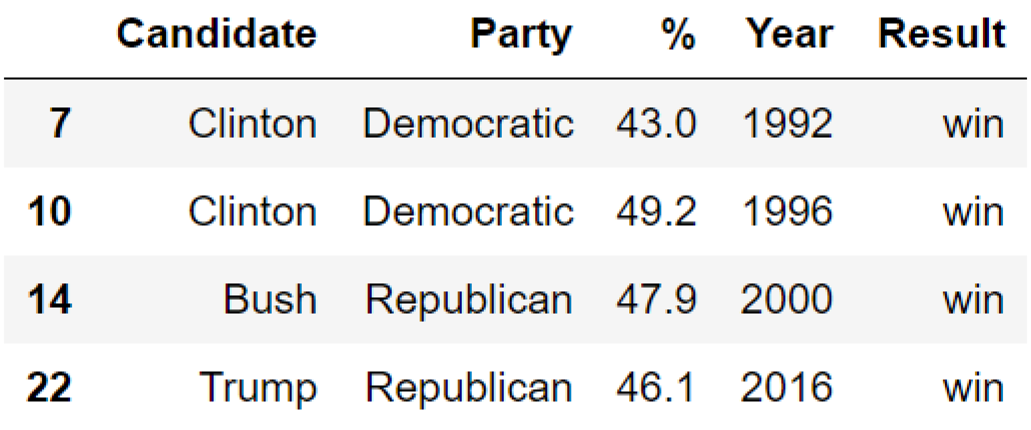
- Note that you must indicate adjacent rows with a numeric range such as 0:3 for 0,1,2

Tables

- Use logical expression to select rows that may not be adjacent in the table
- You can pass a collection of True and False values
- Often you obtain these values by comparing a Series with a variable



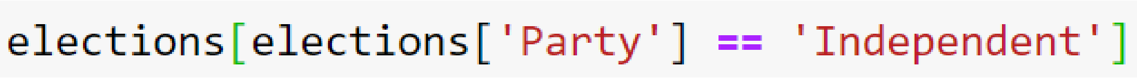
```
elections[[False, False, False, False, False,
          False, False, True, False, False,
          True, False, False, False, True,
          False, False, False, False, False,
          False, False, True]]
```



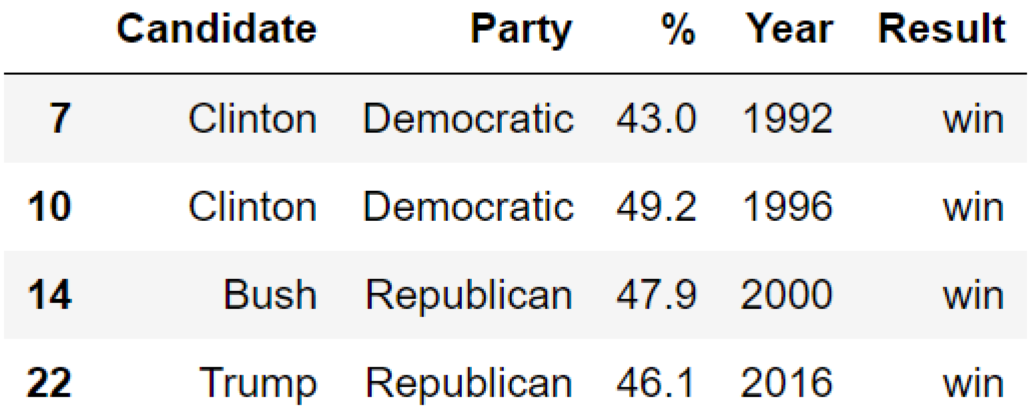
	Candidate	Party	%	Year	Result
7	Clinton	Democratic	43.0	1992	win
10	Clinton	Democratic	49.2	1996	win
14	Bush	Republican	47.9	2000	win
22	Trump	Republican	46.1	2016	win

Tables

- Use logical expression to select rows that may not be adjacent in the table
- You can pass a collection of True and False values
- Often you obtain these values by comparing a Series with a variable



```
elections[elections['Party'] == 'Independent']
```

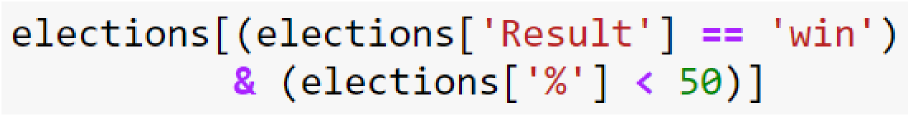


	Candidate	Party	%	Year	Result
7	Clinton	Democratic	43.0	1992	win
10	Clinton	Democratic	49.2	1996	win
14	Bush	Republican	47.9	2000	win
22	Trump	Republican	46.1	2016	win

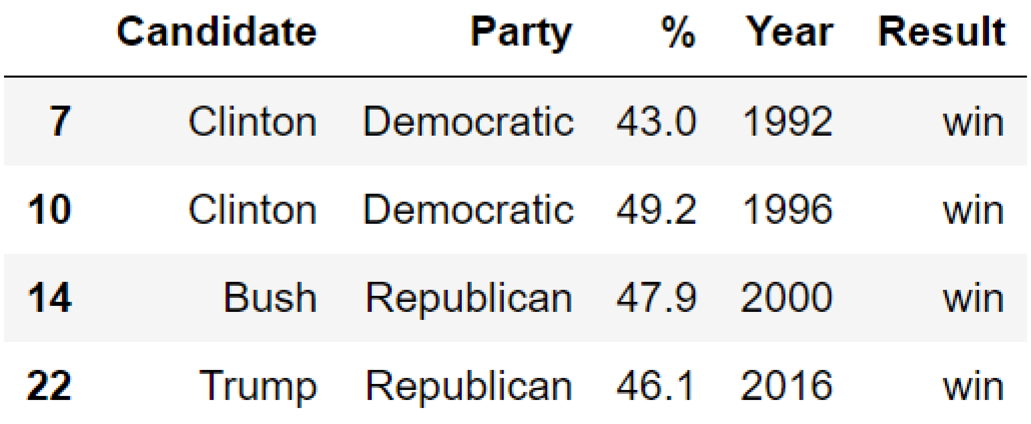
Tables

- Use logical expression to select rows that may not be adjacent in the table
- You can pass a collection of True and False values
- Often you obtain these values by comparing a Series with a variable

You must use & for “and”, | for “or”, ~ for “not”



```
elections[(elections['Result'] == 'win')
         & (elections['%'] < 50)]
```



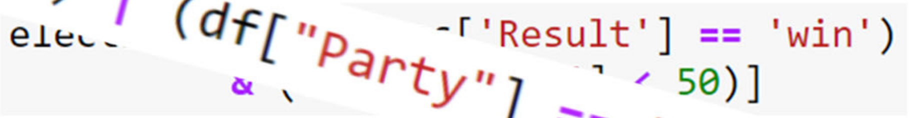
	Candidate	Party	%	Year	Result
7	Clinton	Democratic	43.0	1992	win
10	Clinton	Democratic	49.2	1996	win
14	Bush	Republican	47.9	2000	win
22	Trump	Republican	46.1	2016	win

Tables

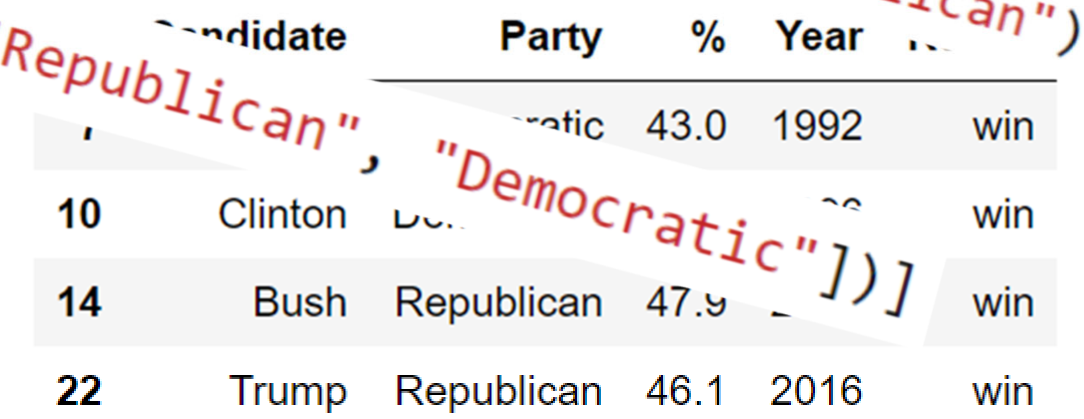
```
df[(df["Party"] == "Democratic") | (df["Party"] == "Republican")]
```

- Use logical expre... to select rows that ...t be adjacent
- You can pass ... collection of True and False values
- Often you obtain these values by comparing a Series with a variable

```
df[df["Party"].isin(["Republican", "Democratic"])]
```



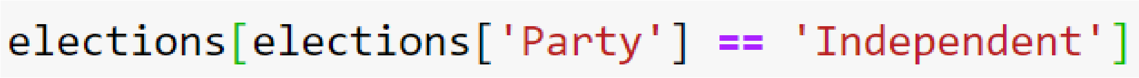
```
elec...   ['Result'] == 'win')
         & (        < 50)]
```



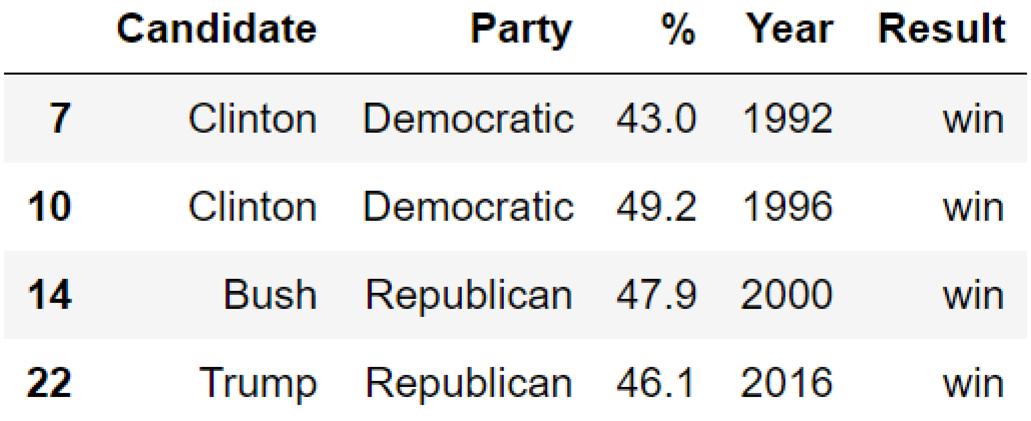
	...ndidate	Party	%	Year	...
7	[illegible]	...ratic	43.0	1992	win
10	Clinton	De...	...6	[illegible]	win
14	Bush	Republican	47.9	[illegible]	win
22	Trump	Republican	46.1	2016	win

Tables

- Use logical expression to select rows that may not be adjacent in the table
- You can pass a collection of True and False values
- Often you obtain these values by comparing a Series with a variable

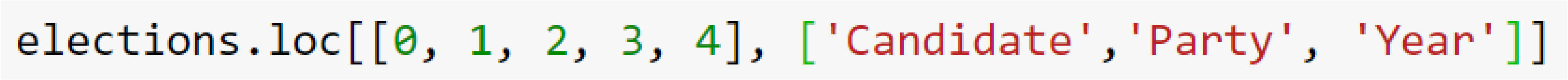


```
elections[elections['Party'] == 'Independent']
```



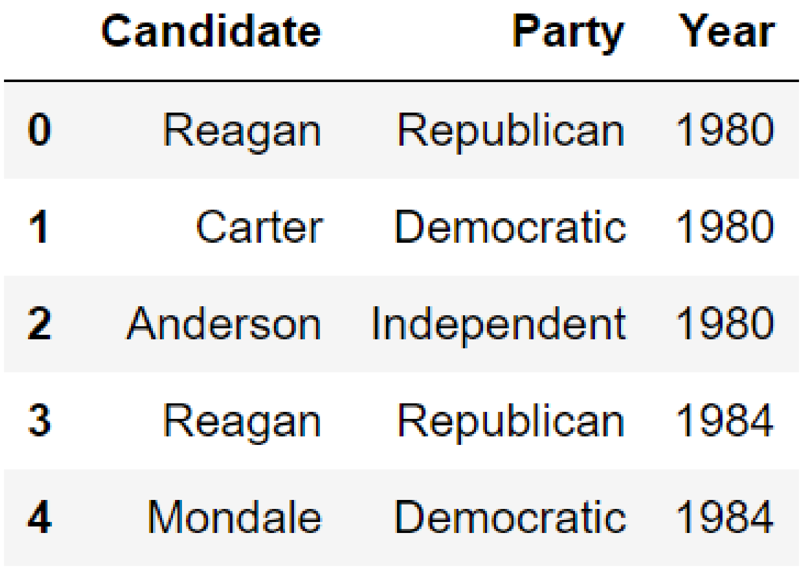
	Candidate	Party	%	Year	Result
7	Clinton	Democratic	43.0	1992	win
10	Clinton	Democratic	49.2	1996	win
14	Bush	Republican	47.9	2000	win
22	Trump	Republican	46.1	2016	win

Tables



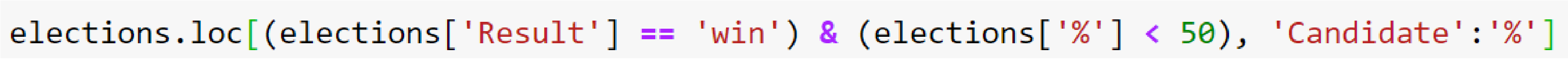
```
elections.loc[[0, 1, 2, 3, 4], ['Candidate','Party', 'Year']]
```

- Use loc and iloc to specify both rows and columns
- loc accesses by
  - value of label
  - True or False
- iloc accesses by
  - row number
  - column number



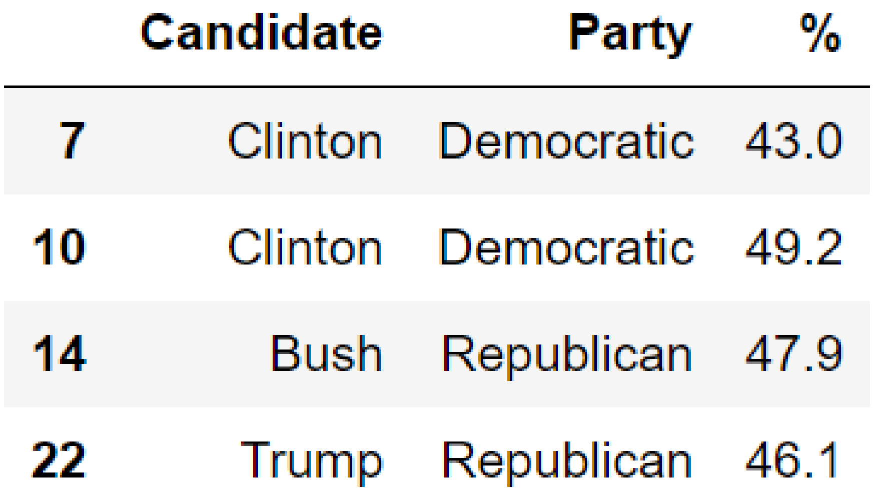
	Candidate	Party	Year
0	Reagan	Republican	1980
1	Carter	Democratic	1980
2	Anderson	Independent	1980
3	Reagan	Republican	1984
4	Mondale	Democratic	1984

Tables



```
elections.loc[(elections['Result'] == 'win') & (elections['%'] < 50), 'Candidate':'%']
```

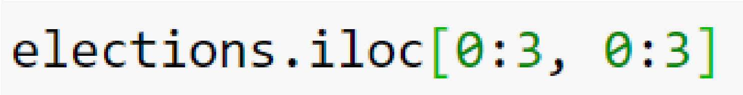
- Use loc and iloc to specify both rows and columns
- loc accesses by
  - value of label
  - True or False
- iloc accesses by
  - row number
  - column number



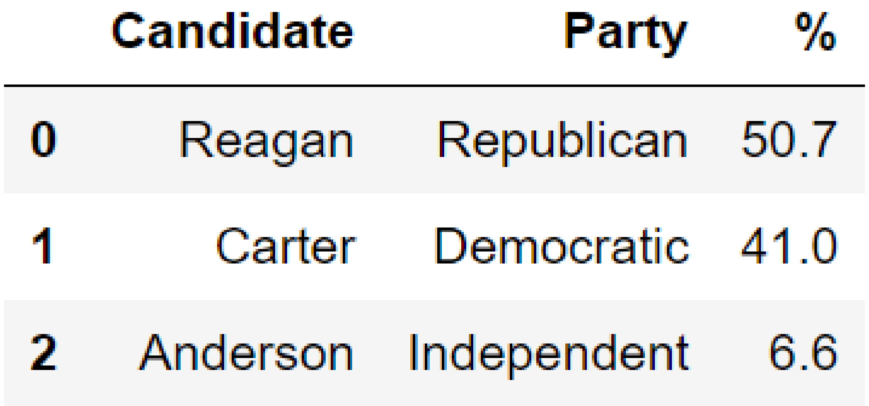
	Candidate	Party	%
7	Clinton	Democratic	43.0
10	Clinton	Democratic	49.2
14	Bush	Republican	47.9
22	Trump	Republican	46.1

Tables

- Use loc and iloc to specify both rows and columns
- loc accesses by
  - value of label
  - True or False
- iloc accesses by
  - row number
  - column number



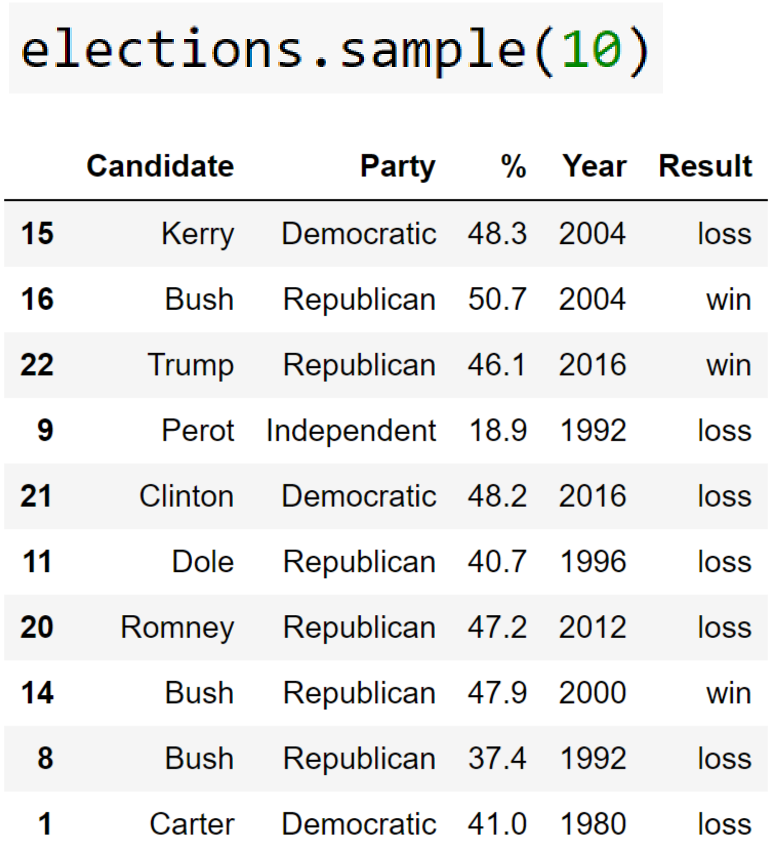
```
elections.iloc[0:3, 0:3]
```



	Candidate	Party	%
0	Reagan	Republican	50.7
1	Carter	Democratic	41.0
2	Anderson	Independent	6.6

Tables

- Use sample to random select from the rows
  - With replacement
  - Without replacement



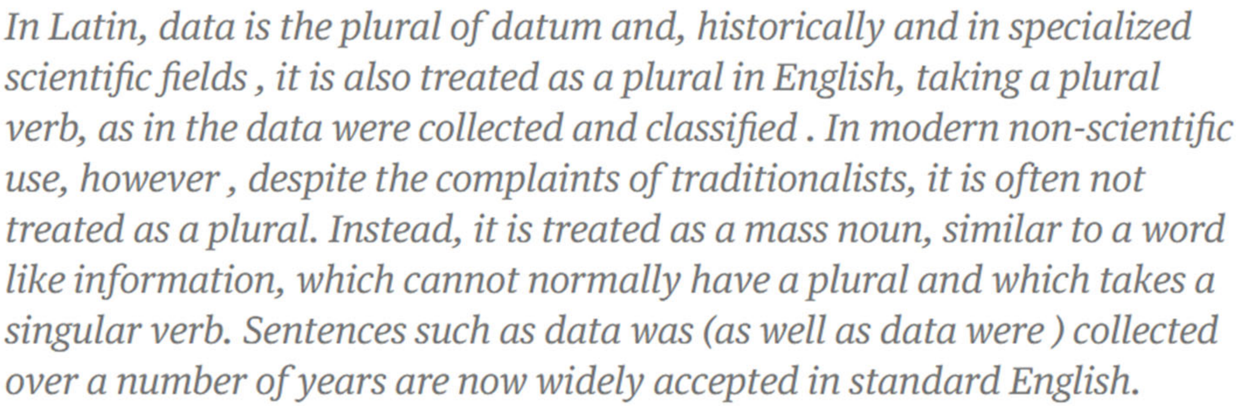
```
elections.sample(10)
```

	Candidate	Party	%	Year	Result
15	Kerry	Democratic	48.3	2004	loss
16	Bush	Republican	50.7	2004	win
22	Trump	Republican	46.1	2016	win
9	Perot	Independent	18.9	1992	loss
21	Clinton	Democratic	48.2	2016	loss
11	Dole	Republican	40.7	1996	loss
20	Romney	Republican	47.2	2012	loss
14	Bush	Republican	47.9	2000	win
8	Bush	Republican	37.4	1992	loss
1	Carter	Democratic	41.0	1980	loss

Questions

- ▶ Questions on Piazza?
- ▶ Question for You!

Should the word data be understood as singular or plural?

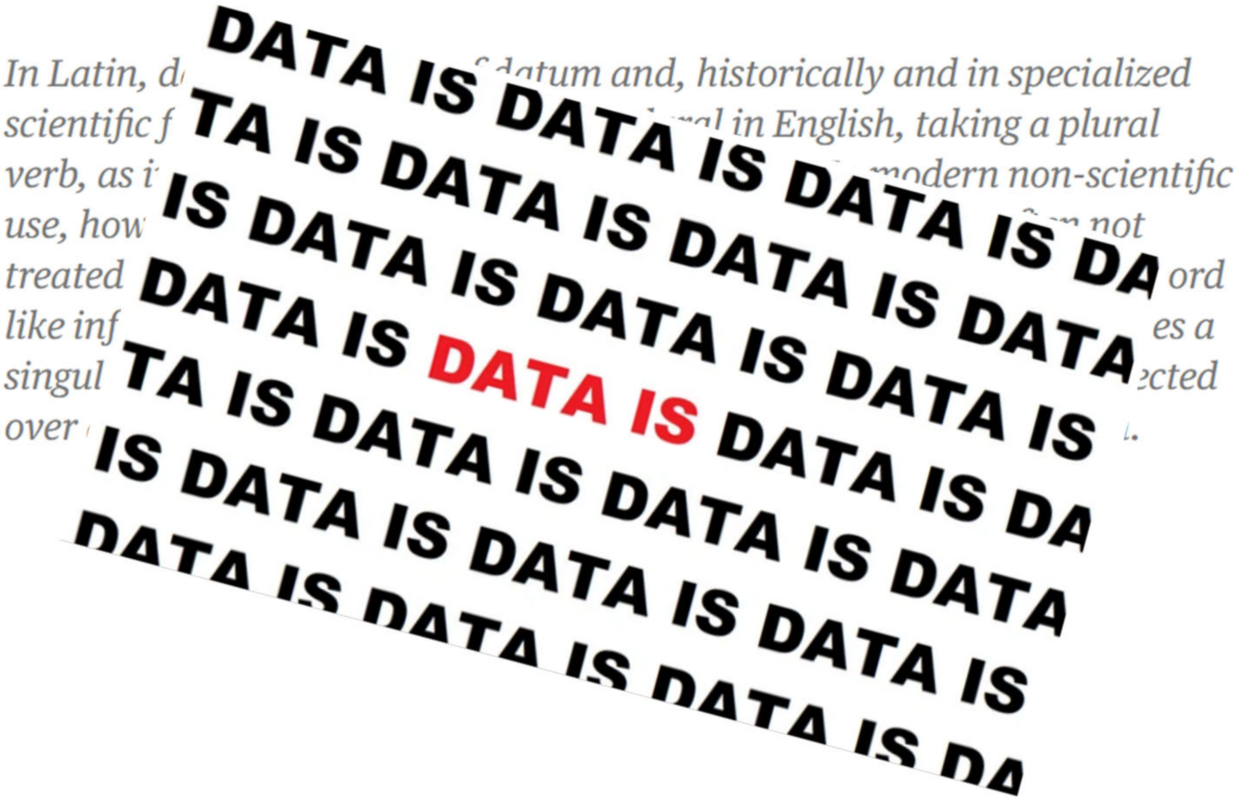


In Latin, data is the plural of datum and, historically and in specialized scientific fields , it is also treated as a plural in English, taking a plural verb, as in the data were collected and classified . In modern non-scientific use, however , despite the complaints of traditionalists, it is often not treated as a plural. Instead, it is treated as a mass noun, similar to a word like information, which cannot normally have a plural and which takes a singular verb. Sentences such as data was (as well as data were) collected over a number of years are now widely accepted in standard English.

Questions

- ▶ Questions on Piazza?
- ▶ Question for You!

Should the word data be understood as singular or plural?


In Latin, d… …datum and, historically and in specialized
scientific f… …al in English, taking a plural
verb, as i… …modern non-scientific
use, how… …not
treated… …ord
like inf… …es a
singul… …ected
over…
DATA IS DATA IS DATA IS DA
TA IS DATA IS DATA IS DATA
IS DATA IS DATA IS DATA IS
DATA IS DATA IS DATA IS DA
TA IS DATA IS DATA IS DATA
IS DATA IS DATA IS DATA IS
DATA IS DATA IS DA