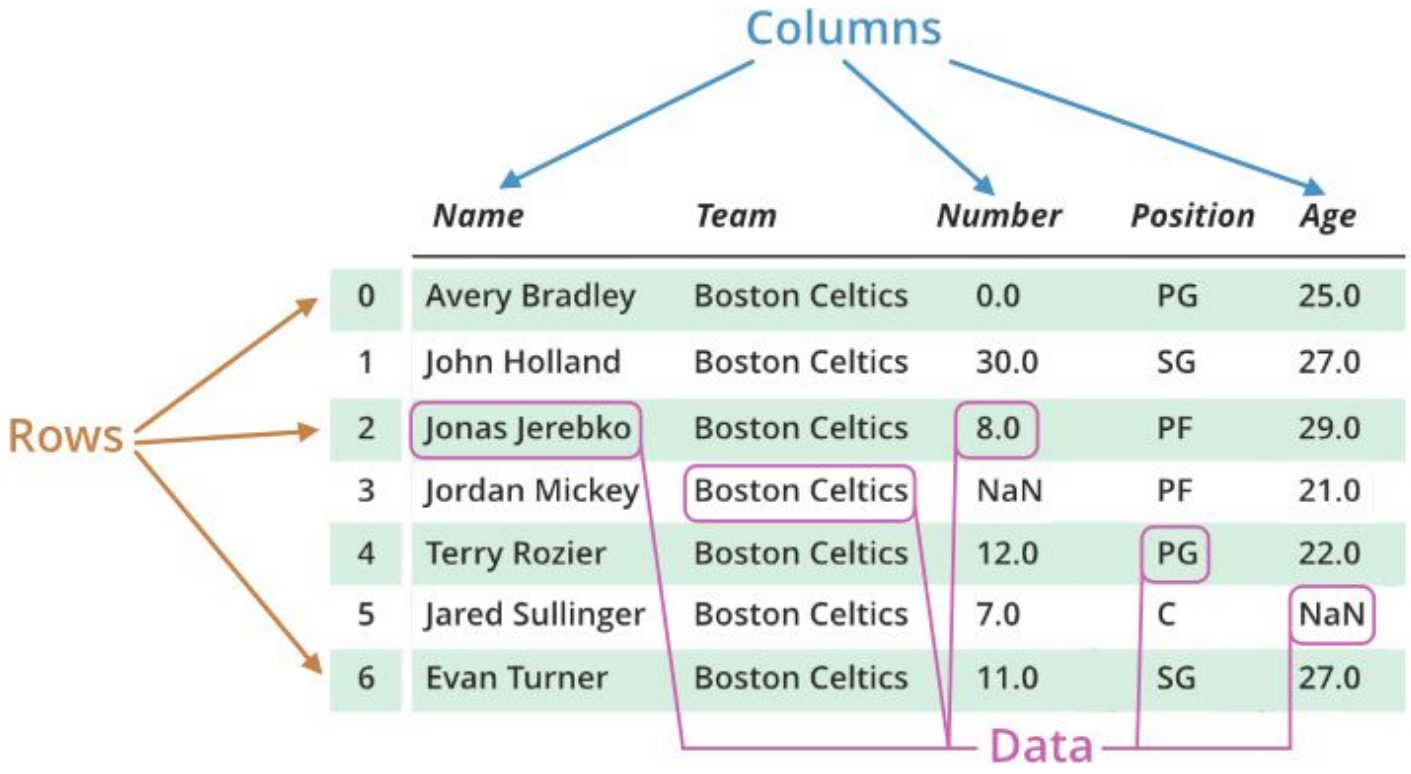
CEI.

Python Pandas

CEI.

pandas



	Name	Team	Number	Position	Age
0	Avery Bradley	Boston Celtics	0.0	PG	25.0
1	John Holland	Boston Celtics	30.0	SG	27.0
2	Jonas Jerebko	Boston Celtics	8.0	PF	29.0
3	Jordan Mickey	Boston Celtics	NaN	PF	21.0
4	Terry Rozier	Boston Celtics	12.0	PG	22.0
5	Jared Sullinger	Boston Celtics	7.0	C	NaN
6	Evan Turner	Boston Celtics	11.0	SG	27.0

pandas

CEI.

```
import pandas as pd
```

pandas read data

CEI.

```
[i for i in dir(pd) if i.startswith('read_')]
```

```
['read_clipboard',
 'read_csv',
 'read_excel',
 'read_feather',
 'read_fwf',
 'read_gbq',
 'read_hdf',
 'read_html',
 'read_json',
 'read_orc',
 'read_parquet',
 'read_pickle',
 'read_sas',
 'read_spss',
 'read_sql',
 'read_sql_query',
 'read_sql_table',
 'read_stata',
 'read_table',
 'read_xml']
```

pandas write data

CEI.

```
[i for i in dir(pd.DataFrame) if i.startswith('to_')]
```

```
['to_clipboard',
 'to_csv',
 'to_dict',
 'to_excel',
 'to_feather',
 'to_gbq',
 'to_hdf',
 'to_html',
 'to_json',
 'to_latex',
 'to_markdown',
 'to_numpy',
 'to_orc',
 'to_parquet',
 'to_period',
 'to_pickle',
 'to_records',
 'to_sql',
 'to_stata',
 'to_string',
 'to_timestamp',
 'to_xarray',
 'to_xml']
```

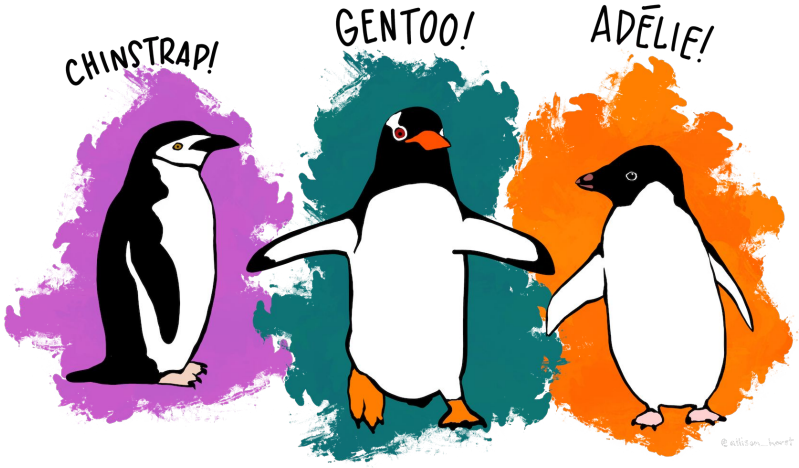
read data

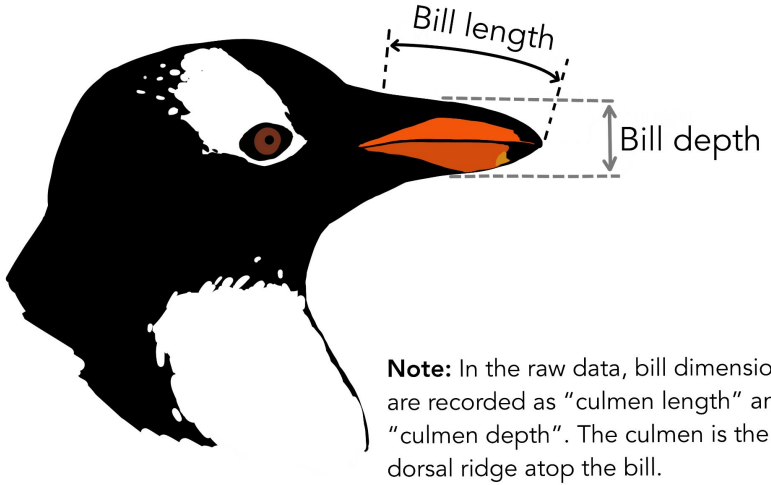
CEI.

```
url = 'https://raw.githubusercontent.com/mwaskom/seaborn-data/refs/heads/master/penguins.csv'
df = pd.read_csv(url)
df.head()
```

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	MALE
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	FEMALE
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	FEMALE
3	Adelie	Torgersen	NaN	NaN	NaN	NaN	NaN
4	Adelie	Torgersen	36.7	19.3	193.0	3450.0	FEMALE

read data





Note: In the raw data, bill dimensions are recorded as "culmen length" and "culmen depth". The culmen is the dorsal ridge atop the bill.

analyze data

CEI.

```
df.columns
```

```
Index(['species', 'island', 'bill_length_mm', 'bill_depth_mm',
       'flipper_length_mm', 'body_mass_g', 'sex'],
      dtype='object')
```

analyze data

CEI.

```
df.dtypes
```

```
species              object
island               object
bill_length_mm      float64
bill_depth_mm       float64
flipper_length_mm   float64
body_mass_g         float64
sex                  object
dtype: object
```

analyze data

CEI.

select only object (string) data

```
df.select_dtypes('object')
```

	species	island	sex
0	Adelie	Torgersen	MALE
1	Adelie	Torgersen	FEMALE
2	Adelie	Torgersen	FEMALE
3	Adelie	Torgersen	NaN
4	Adelie	Torgersen	FEMALE
...	...	...	...

analyze data

CEI.

select only float (numeric) data

```
df.select_dtypes(include=['float'])
```

	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g
0	39.1	18.7	181.0	3750.0
1	39.5	17.4	186.0	3800.0
2	40.3	18.0	195.0	3250.0
3	NaN	NaN	NaN	NaN
4	36.7	19.3	193.0	3450.0

CEI.

Question
how do I know that the rows correspond to each other?

```
df.select_dtypes('object')
```

	species	island	sex
0	Adelie	Torgersen	MALE
1	Adelie	Torgersen	FEMALE
2	Adelie	Torgersen	FEMALE
3	Adelie	Torgersen	NaN
4	Adelie	Torgersen	FEMALE
...	...	...	...

```
df.select_dtypes(include=['float'])
```

	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g
0	39.1	18.7	181.0	3750.0
1	39.5	17.4	186.0	3800.0
2	40.3	18.0	195.0	3250.0
3	NaN	NaN	NaN	NaN
4	36.7	19.3	193.0	3450.0

analyze data

CEI.

description of numeric data

```
df.describe()
```

	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g
count	342.000000	342.000000	342.000000	342.000000
mean	43.921930	17.151170	200.915205	4201.754386
std	5.459584	1.974793	14.061714	801.954536
min	32.100000	13.100000	172.000000	2700.000000
25%	39.225000	15.600000	190.000000	3550.000000
50%	44.450000	17.300000	197.000000	4050.000000
75%	48.500000	18.700000	213.000000	4750.000000
max	59.600000	21.500000	231.000000	6300.000000

analyze data

CEI.

description of object data

```
df.describe(include=['object'])
```

	species	island	sex
count	344	344	333
unique	3	3	2
top	Adelie	Biscoe	MALE
freq	152	168	168

analyze data

CEI.

covariance of numerical data

```
df.cov(numeric_only=True)
```

	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g
bill_length_mm	29.807054	-2.534234	50.375765	2605.591912
bill_depth_mm	-2.534234	3.899808	-16.212950	-747.370093
flipper_length_mm	50.375765	-16.212950	197.731792	9824.416062
body_mass_g	2605.591912	-747.370093	9824.416062	643131.077327

analyze data

CEI.

correlation of numerical data

```
df.corr(numeric_only=True)
```

	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g
bill_length_mm	1.000000	-0.235053	0.656181	0.595110
bill_depth_mm	-0.235053	1.000000	-0.583851	-0.471916
flipper_length_mm	0.656181	-0.583851	1.000000	0.871202
body_mass_g	0.595110	-0.471916	0.871202	1.000000

missing data

CEI.

count missing data

```
df.isna().sum()
```

```
species              0
island               0
bill_length_mm       2
bill_depth_mm        2
flipper_length_mm    2
body_mass_g          2
sex                 11
dtype: int64
```

missing data

CEI.

Exercise
want to know the percentage of missing data per column

```
species              0.000000
island               0.000000
bill_length_mm       0.581395
bill_depth_mm        0.581395
flipper_length_mm    0.581395
body_mass_g          0.581395
sex                  3.197674
dtype: float64
```

missing imputation

CEI.

Let’s make a column with no NA named `is_female`

```
df['is_female'] = (df['sex'] == 'FEMALE').astype(int)
```

missing imputation

CEI.

Let’s make a column with no NA named `is_female`

```
df['is_female'] = (df['sex'] == 'FEMALE').astype(int)
```

```
df.isnull().sum()
```

```
species              0
island               0
bill_length_mm       2
bill_depth_mm        2
flipper_length_mm    2
body_mass_g          2
sex                 11
is_female            0
dtype: int64
```

missing imputation

CEI.

Let’s make a column with no NA named `is_female`

```
df['is_female'] = (df['sex'] == 'FEMALE').astype(int)
```

```
df.isnull().sum()
```

```
species              0
island               0
bill_length_mm       2
bill_depth_mm        2
flipper_length_mm    2
body_mass_g          2
sex                 11
is_female            0
dtype: int64
```

missing imputation

CEI.

Let’s make a column with no NA named `is_female`

```
df['is_female'] = df['sex'].map({'MALE': 0, 'FEMALE':1})
```

```
df.isnull().sum()
```

```
species              0
island               0
bill_length_mm       2
bill_depth_mm        2
flipper_length_mm    2
body_mass_g          2
sex                 11
is_female           11
dtype: int64
```

missing imputation

CEI.

drop the column `sex`

```
df = df.drop(['sex'],axis=1)
df.head()
```

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	is_female
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	0.0
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	1.0
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	1.0
3	Adelie	Torgersen	NaN	NaN	NaN	NaN	NaN
4	Adelie	Torgersen	36.7	19.3	193.0	3450.0	1.0

missing imputation

CEI.

Exercise: Create a dictionary with the mean values of these columns and named it numeric means

```
columns = ['bill_length_mm', 'bill_depth_mm', 'flipper_length_mm', 'body_mass_g']
```

```
{'bill_length_mm': np.float64(43.9219298245614),
 'bill_depth_mm': np.float64(17.151169590643278),
 'flipper_length_mm': np.float64(200.91520467836258),
 'body_mass_g': np.float64(4201.754385964912)}
```

missing imputation

CEI.

Exercise: Create a dictionary with the mean values of these columns and named it numeric means

```
df = df.fillna(numeric_means)
df.isna().sum()
```

```
species              0
island               0
bill_length_mm       0
bill_depth_mm        0
flipper_length_mm    0
body_mass_g          0
is_female           11
dtype: int64
```

missing imputation

CEI.

Create a pandas with the pct of female penguins per specie and island (name this pandas aux)

	species	island	female_pct
0	Adelie	Biscoe	0.500000
1	Adelie	Dream	0.490909
2	Adelie	Torgersen	0.510638
3	Chinstrap	Dream	0.500000
4	Gentoo	Biscoe	0.487395

missing imputation

CEI.

Now add a column ‘fill_is_female’ (1 if `female_pct` >= 0.5 else 0) this new column should be integer

	species	island	female_pct	fill_is_female
0	Adelie	Biscoe	0.500000	1
1	Adelie	Dream	0.490909	0
2	Adelie	Torgersen	0.510638	1
3	Chinstrap	Dream	0.500000	1
4	Gentoo	Biscoe	0.487395	0

missing imputation

CEI.

remove column `female_pct`

```
aux = aux.drop(['female_pct'], axis=1)
aux
```

	species	island	fill_is_female
0	Adelie	Biscoe	1
1	Adelie	Dream	0
2	Adelie	Torgersen	1
3	Chinstrap	Dream	1
4	Gentoo	Biscoe	0

missing imputation

CEI.

merge df and aux by `species` and `island`

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	is_female	fill_is_female
0	Adelie	Torgersen	39.10000	18.70000	181.000000	3750.000000	0.0	1
1	Adelie	Torgersen	39.50000	17.40000	186.000000	3800.000000	1.0	1
2	Adelie	Torgersen	40.30000	18.00000	195.000000	3250.000000	1.0	1
3	Adelie	Torgersen	43.92193	17.15117	200.915205	4201.754386	NaN	1
4	Adelie	Torgersen	36.70000	19.30000	193.000000	3450.000000	1.0	1
...	...	...	...	...	...	...	...	...
339	Gentoo	Biscoe	43.92193	17.15117	200.915205	4201.754386	NaN	0
340	Gentoo	Biscoe	46.80000	14.30000	215.000000	4850.000000	1.0	0
341	Gentoo	Biscoe	50.40000	15.70000	222.000000	5750.000000	0.0	0
342	Gentoo	Biscoe	45.20000	14.80000	212.000000	5200.000000	1.0	0
343	Gentoo	Biscoe	49.90000	16.10000	213.000000	5400.000000	0.0	0

missing imputation

CEI.

fill na of `is_female` columns with `fill_is_female` values

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	is_female	fill_is_female
0	Adelie	Torgersen	39.10000	18.70000	181.000000	3750.000000	0	1
1	Adelie	Torgersen	39.50000	17.40000	186.000000	3800.000000	1	1
2	Adelie	Torgersen	40.30000	18.00000	195.000000	3250.000000	1	1
3	Adelie	Torgersen	43.92193	17.15117	200.915205	4201.754386	1	1
4	Adelie	Torgersen	36.70000	19.30000	193.000000	3450.000000	1	1
...	...	...	...	...	...	...	...	...
339	Gentoo	Biscoe	43.92193	17.15117	200.915205	4201.754386	0	0
340	Gentoo	Biscoe	46.80000	14.30000	215.000000	4850.000000	1	0
341	Gentoo	Biscoe	50.40000	15.70000	222.000000	5750.000000	0	0
342	Gentoo	Biscoe	45.20000	14.80000	212.000000	5200.000000	1	0
343	Gentoo	Biscoe	49.90000	16.10000	213.000000	5400.000000	0	0

missing imputation

CEI.

drop`fill_is_female` and check missing values

```
df = df.drop(['fill_is_female'], axis= 1)
df.isna().sum()
```

```
species              0
island               0
bill_length_mm       0
bill_depth_mm        0
flipper_length_mm    0
body_mass_g          0
is_female            0
dtype: int64
```

c.