```
import numpy as np
import pandas as pd
np.random.seed(12345)
import matplotlib.pyplot as plt
plt.rc("figure", figsize=(10, 6))
pd.options.display.max_colwidth = 75
pd.options.display.max_columns = 20
np.set_printoptions(precision=4, suppress=True)

!cat examples/ex1.csv

df = pd.read_csv("examples/ex1.csv")
df

!cat examples/ex2.csv

pd.read_csv("examples/ex2.csv", header=None)
pd.read_csv("examples/ex2.csv", names=["a", "b", "c", "d", "message"])

names = ["a", "b", "c", "d", "message"]
pd.read_csv("examples/ex2.csv", names=names, index_col="message")

!cat examples/csv_mindex.csv
parsed = pd.read_csv("examples/csv_mindex.csv",
                     index_col=["key1", "key2"])
parsed

!cat examples/ex3.txt

result = pd.read_csv("examples/ex3.txt", sep="\s+")
result

!cat examples/ex4.csv
pd.read_csv("examples/ex4.csv", skiprows=[0, 2, 3])

!cat examples/ex5.csv
result = pd.read_csv("examples/ex5.csv")
result

pd.isna(result)

result = pd.read_csv("examples/ex5.csv", na_values=["NULL"])
result

result2 = pd.read_csv("examples/ex5.csv", keep_default_na=False)
result2
result2.isna()
result3 = pd.read_csv("examples/ex5.csv", keep_default_na=False,
                      na_values=["NA"])
result3
result3.isna()
```

```
sentinels = {"message": ["foo", "NA"], "something": ["two"]}
pd.read_csv("examples/ex5.csv", na_values=sentinels,
            keep_default_na=False)

pd.options.display.max_rows = 10

result = pd.read_csv("examples/ex6.csv")
result

pd.read_csv("examples/ex6.csv", nrows=5)

chunker = pd.read_csv("examples/ex6.csv", chunksize=1000)
type(chunker)

chunker = pd.read_csv("examples/ex6.csv", chunksize=1000)

tot = pd.Series([], dtype='int64')
for piece in chunker:
    tot = tot.add(piece["key"].value_counts(), fill_value=0)

tot = tot.sort_values(ascending=False)

tot[:10]

data = pd.read_csv("examples/ex5.csv")
data

data.to_csv("examples/out.csv")
!cat examples/out.csv

import sys
data.to_csv(sys.stdout, sep="|")

data.to_csv(sys.stdout, na_rep="NULL")

data.to_csv(sys.stdout, index=False, header=False)

data.to_csv(sys.stdout, index=False, columns=["a", "b", "c"])

!cat examples/ex7.csv

import csv
f = open("examples/ex7.csv")
reader = csv.reader(f)

for line in reader:
    print(line)
f.close()

with open("examples/ex7.csv") as f:
    lines = list(csv.reader(f))

header, values = lines[0], lines[1:]
```

```
data_dict = {h: v for h, v in zip(header, zip(*values))}
data_dict

obj = """
{"name": "Wes",
 "cities_lived": ["Akron", "Nashville", "New York", "San Francisco"],
 "pet": null,
 "siblings": [{"name": "Scott", "age": 34, "hobbies": ["guitars",
"soccer"]},
              {"name": "Katie", "age": 42, "hobbies": ["diving",
"art"]}]
}
"""

import json
result = json.loads(obj)
result

asjson = json.dumps(result)
asjson

siblings = pd.DataFrame(result["siblings"], columns=["name", "age"])
siblings

!cat examples/example.json

data = pd.read_json("examples/example.json")
data

data.to_json(sys.stdout)
data.to_json(sys.stdout, orient="records")

tables = pd.read_html("examples/fdic_failed_bank_list.html")
len(tables)
failures = tables[0]
failures.head()

close_timestamps = pd.to_datetime(failures["Closing Date"])
close_timestamps.dt.year.value_counts()

from lxml import objectify

path = "datasets/mta_perf/Performance_MNR.xml"
with open(path) as f:
    parsed = objectify.parse(f)
root = parsed.getroot()

data = []

skip_fields = ["PARENT_SEQ", "INDICATOR_SEQ",
               "DESIRED_CHANGE", "DECIMAL_PLACES"]
```

```
for elt in root.INDICATOR:
    el_data = {}
    for child in elt.getchildren():
        if child.tag in skip_fields:
            continue
        el_data[child.tag] = child.pyval
    data.append(el_data)

perf = pd.DataFrame(data)
perf.head()

perf2 = pd.read_xml(path)
perf2.head()

frame = pd.read_csv("examples/ex1.csv")
frame
frame.to_pickle("examples/frame_pickle")

pd.read_pickle("examples/frame_pickle")

!rm examples/frame_pickle

fec = pd.read_parquet('datasets/fec/fec.parquet')

xlsx = pd.ExcelFile("examples/ex1.xlsx")

xlsx.sheet_names

xlsx.parse(sheet_name="Sheet1")

xlsx.parse(sheet_name="Sheet1", index_col=0)

frame = pd.read_excel("examples/ex1.xlsx", sheet_name="Sheet1")
frame

writer = pd.ExcelWriter("examples/ex2.xlsx")
frame.to_excel(writer, "Sheet1")
writer.close()

frame.to_excel("examples/ex2.xlsx")

!rm examples/ex2.xlsx

!rm -f examples/mydata.h5

frame = pd.DataFrame({"a": np.random.standard_normal(100)})
store = pd.HDFStore("examples/mydata.h5")
store["obj1"] = frame
store["obj1_col"] = frame["a"]
store

store["obj1"]
```

```
store.put("obj2", frame, format="table")
store.select("obj2", where=["index >= 10 and index <= 15"])
store.close()

frame.to_hdf("examples/mydata.h5", "obj3", format="table")
pd.read_hdf("examples/mydata.h5", "obj3", where=["index < 5"])

import os
os.remove("examples/mydata.h5")

import requests
url = "https://api.github.com/repos/pandas-dev/pandas/issues"
resp = requests.get(url)
resp.raise_for_status()
resp

data = resp.json()
data[0]["title"]

issues = pd.DataFrame(data, columns=["number", "title",
                                     "labels", "state"])
issues

import sqlite3

query = """
CREATE TABLE test
(a VARCHAR(20), b VARCHAR(20),
 c REAL,        d INTEGER
);"""

con = sqlite3.connect("mydata.sqlite")
con.execute(query)
con.commit()

data = [("Atlanta", "Georgia", 1.25, 6),
        ("Tallahassee", "Florida", 2.6, 3),
        ("Sacramento", "California", 1.7, 5)]
stmt = "INSERT INTO test VALUES(?, ?, ?, ?)"

con.executemany(stmt, data)
con.commit()

cursor = con.execute("SELECT * FROM test")
rows = cursor.fetchall()
rows

cursor.description
pd.DataFrame(rows, columns=[x[0] for x in cursor.description])
```

```
import sqlalchemy as sqla
db = sqla.create_engine("sqlite:///mydata.sqlite")
pd.read_sql("SELECT * FROM test", db)

!rm mydata.sqlite
```