In [ ]:
```
import pandas as pd
fortune_500 = pd.read_csv("f500.csv", index_col=0)
fortune_500.index.name = None
```

In [ ]:
```
import pandas as pd
fortune_500 = pd.read_csv("f500.csv")
```

In [ ]:
```
fortune_500.dtypes
```

In [ ]:
```
fortune_500.columns
```

In [ ]:
```
fortune_500.info()
```

In [ ]:
```
fortune_500
```

In [ ]:
```
fortune_500["rank"]
```

In [ ]:
```
fortune_500.loc[:, "rank"]
```

In [ ]:
```
fortune_500["rank"]
```

In [ ]:
```
fortune_500["industry"]
```

In [ ]:
```
fortune_500[["industry", "sector"]]
```

In [ ]:
```
fortune_500["industry"]
```

In [ ]:
```
fortune_500.loc[:, "rank":"country"]
```

In [ ]:
```
fortune_500.loc[0, "revenues":"ceo"]
```

In [ ]:
```
fortune_500.loc[0]
```

In [ ]:
```
fortune_500.loc["Walmart":"State Grid"]
```

In [ ]:
```
fortune_500.loc[:, "rank":"profits"]
```

In [ ]:
```
fortune_500.loc["Walmart":"State Grid", "rank":"profits"]
```

In [ ]:
```
sector = fortune_500["sector"]
```

In [ ]:
```
sector
```

In [ ]:
```
sector.value_counts()
```

In [ ]:
```
country = fortune_500["country"]
```

In [ ]:
```
country
```

In [ ]:
```
country_counts = country.value_counts()
```

In [ ]:
```
country_counts
```

In [ ]:
```
india = country_counts["India"]
```

In [ ]:
```
north_america = country_counts[["USA", "Mexico", "Canada"]]
```

In [ ]:
```
north_america
```

In [ ]:
```
big_movers = ["Aviva", "HP", "JD.com", "BHP Billiton"]
big_sectors = ["rank", "previous_rank"]
```

In [ ]:
```
fortune_500.loc[big_movers, big_sectors]
```

In [ ]:
```
fortune_500.loc["National Grid":"AutoNation", big_sectors]
```

In [ ]:
```
rank_change =  fortune_500["previous_rank"] - fortune_500["rank"]
```

In [ ]:
```
rank_change
```

In [ ]:
```
rank_change.min()
```

In [ ]:
```
rank_change.describe()
```

In [ ]:
```
fortune_500.describe()
```

In [ ]:
```
asset = fortune_500["assets"]
```

In [ ]:
```
asset.describe()
```

In [ ]:
```
countr = fortune_500["country"]
```

In [ ]:
```
countr.describe()
```

In [ ]:
```
countr_asset = ["previous_rank"]
ranking = fortune_500.loc[:, countr_asset]
```

In [ ]:
```
prev_rank = ranking.describe()
```

In [ ]:
```
ranking.value_counts()
```

In [ ]:
```
country_value_counts = countr.value_counts()
```

In [ ]:
```
country_value_counts
```

In [ ]:
```
country_value_counts.loc["India"]
```

In [ ]:
```
country_rank = ["previous_rank"]
countr_rank = fortune_500[country_rank]
```

In [ ]:
```
zer_comp = countr_rank.value_counts()
```

In [ ]:
```
zer_comp
```

In [ ]:
```
fortune_500["revenues"].median(axis=0)
```

In [ ]:
```
fortune_500["profits"].median(axis=0)
```

In [ ]:
```
fortune_500.dtypes
```

In [ ]:
```
numeric_column = ["rank", "revenues", "revenue_change", "profits", "assets", "pro
```

In [ ]:
```
min_rank = fortune_500["rank"].min()
```

In [ ]:
```
min_rank
```

In [ ]:
```
max_rev = fortune_500["revenues"].max()
```

In [ ]:
```
max_rev
```

In [ ]:
```
fortune_500["revenue_change"].max()
```

In [ ]:
```
max_f500 = fortune_500.max(numeric_only=True)
```

In [ ]:
```
max_f500
```

In [ ]:
```
fortune_500.max()
```

In [ ]:
```
fortune_500.describe(include=["O"])
```

In [ ]:
```
fortune_500.describe()
```

In [ ]:
```
fortune_500["ceo"].loc["Walmart"]
```

In [ ]:
```
fortune_500["previous_rank"].value_counts()
```

In [ ]:
```
fortune_500
```

In [ ]:
```
new_fortune = fortune_500["previous_rank"]
```

In [ ]:
```
fortune_500_bool = new_fortune == 0
```

In [ ]:
```
import numpy as np
fortune_500.loc[fortune_500_bool, "previous_rank"] = np.nan
```

In [ ]:
```
fortune_500["previous_rank"].value_counts()
```

In [ ]:
```
fortune_500
```

In [ ]:
```
industry = fortune_500["industry"]
industry_bool = industry == "Motor Vehicles and Parts"
motor_bool = industry[industry_bool]
motor_bool
```

In [ ]:
```
fortune_500.loc[industry_bool, "country"]
```

In [ ]:
```
fortune_500["previous_rank"].value_counts(dropna=False).head(10)
```

In [ ]:
```
fortune_500["rank_change"] = fortune_500["previous_rank"] - fortune_500["rank"]
```

In [ ]:
```
fortune_500["rank_change"].describe()
```

In [ ]:
```
rank_1 = fortune_500["rank"]
rank_1_bool = rank_1 == 46
num_46 = fortune_500.loc[rank_1_bool, "rank"]
num_46
```

In [ ]:
```
fortune_500.columns
```

In [ ]:
```
fortune_500["industry"].value_counts()
```

In [ ]:
```
industry_1 = fortune_500["industry"]
industry_1_bool = industry_1 == "Banks: Commercial and Savings"
banks = fortune_500.loc[industry_1_bool, "country"]
banking = banks.value_counts()
banking.sum()
```

In [ ]:
```
fortune_500.loc[0:4]
```

In [ ]:
```
company_value = fortune_500.loc[0:4, "company"]
```

In [ ]:
```
company_value
```

In [ ]:
```
Pr = fortune_500["previous_rank"].isnull()
```

In [ ]:
```
r = fortune_500["rank"].notnull()
```

In [ ]:
```
Pr_1 = fortune_500.loc[Pr, "company":"total_stockholder_equity"]
```

In [ ]:
```
Pr_1
```

In [ ]:
```
fortune_500
```

In [ ]:
```
null_previous_rank = fortune_500[empty][["company","rank", "previous_rank"]]
```

In [ ]:
```
y = fortune_500["previous_rank"].isnull()
y_1 = fortune_500[y]
null = y_1[["company", "previous_rank"]]
null
```

In [ ]:
```
null.iloc[:5]
```

In [ ]:
```
null_previous_rank = f500[f500["previous_rank"].isnull()][["company","rank", "pre
```

In [ ]:
```
y_1 = fortune_500[y]
```

In [ ]:
```
y_1["previous_rank"]
```

In [ ]:
```
empty_1["previous_rank"]
```

In [ ]:
```
fortune_500["previous_rank"].value_counts(dropna=False).head(10)
```

In [ ]:
```
null_previous_rank = fortune_500[fortune_500["previous_rank"].isnull()][["company
```

In [ ]:
```
null_previous_rank
```

In [ ]:
```
null_previous_rank = fortune_500[fortune_500["ceo"].isnull()][["company", "rank"]
```

In [ ]:
```
d = fortune_500["ceo"].isnull()
d
```

In [ ]:
```
null_previous_rank
```

In [ ]:
```
null_previous_rank = fortune_500[fortune_500["previous_rank"].isnull()]
null_previous_rank
```

In [ ]:
```
y = fortune_500["previous_rank"].notnull()
y_1 = fortune_500[y]
not_nul = y_1[["company", "previous_rank", "rank"]]
not_nul
#y_1
```

In [ ]:
```
rank_changed = not_nul["previous_rank"] - not_nul["rank"]
```

In [ ]:
```
rank_changed
```

In [ ]:
```
fortune_500["rank_changed"] = rank_changed
```

In [ ]:
```
fortune_500["testing"] = 0
```

In [ ]:
```
fortune_500
```

In [ ]:
```
access = ["company", "rank", "industry", "revenues", "profits"]
revenue = fortune_500["revenues"]
profit = fortune_500["profits"]
revenue_bool = (revenue > 100000) & (profit < 0)
the_big = fortune_500.loc[revenue_bool, access]
the_big
```

In [ ]:
```
acess = ["company", "rank", "industry", "sector", "country"]
sec = fortune_500["sector"]
sec_tech = sec == "Technology"
country = fortune_500["country"]
booli = (sec_tech) & (country != "USA")
tech_not_usa = fortune_500.loc[booli, acess]
tech_not_usa
```

In [ ]:
```
fortune_500
```

In [ ]:
```
emp = ["country", "employees", "company"]
country = fortune_500["country"]
employee = fortune_500["employees"]
china_bool = country == "China"
china_emplo = fortune_500.loc[china_bool, emp]
top_employees = china_emplo.sort_values("employees", ascending=False)
```

In [ ]:
```
top_employees.iloc[:10]
```

In [ ]:
```
coun_ind = ["country", "industry"]
country = fortune_500[coun_ind]
countries = country.value_counts("country")
```

In [ ]:
```
fortune_500["company"].nunique()
```

In [ ]:
```
emp = ["country", "employees", "company"]
fort = fortune_500[emp]
f = fort.sort_values("employees", ascending=False)
f.iloc[:5]
```

In [ ]:
```
most_employed = {}
unique_country = fortune_500["country"].unique()
for country in unique_country:
    countries = fortune_500["country"]
    fortune_bool = countries == country
    country_rows = fortune_500.loc[fortune_bool, "employees"]
    row_sum = country_rows.sum()
    most_employed[country] = row_sum
most_employed
```

In [ ]:
```
prof = fortune_500["profits"]
asset = fortune_500["assets"]
fortune_500["ROA"] = (fortune_500["profits"]) / (fortune_500["assets"])
```

In [ ]:
```
fortune_500
```

In [ ]:
```
roa_dictionary = {}
unique_sector = fortune_500["sector"].unique()
for sect in unique_sector:
    sector = fortune_500["sector"]
    boolian = sector == sect
    the_sector = fortune_500.loc[boolian, "ROA"]
    summing = the_sector.sum()
    roa_dictionary[sect] = summing
roa_dictionary
```

In [ ]:
```
google_sorted_dict = dict( sorted(roa_dictionary.items(),
                          key=lambda item: item[1],
                          reverse=True))
```

In [1]:
```
import pandas as pd
lapy = pd.read_csv("laptops.csv", encoding="Latin-1")
```

In [2]:
```
lapy.columns
```

Out[2]:
```
Index(['Manufacturer', 'Model Name', 'Category', 'Screen Size', 'Screen',
       'CPU', 'RAM', ' Storage', 'GPU', 'Operating System',
       'Operating System Version', 'Weight', 'Price (Euros)'],
      dtype='object')
```

In [3]:
```
lapy.describe()
```

Out[3]:

	Manufacturer	Model Name	Category	Screen Size	Screen	CPU	RAM	Storage	GPU	Ope Sy
count	1303	1303	1303	1303	1303	1303	1303	1303	1303	
unique	19	618	6	18	40	118	9	38	110	
top	Dell	XPS 13	Notebook	15.6"	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8GB	256GB SSD	Intel HD Graphics 620	Wi
freq	297	30	727	665	507	190	619	412	281	

In [4]:
```
lapy
```

Out[4]:

	Manufacturer	Model Name	Category	Screen Size	Screen	CPU	RAM	Stora
0	Apple	MacBook Pro	Ultrabook	13.3"	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz	8GB	128 S
1	Apple	Macbook Air	Ultrabook	13.3"	1440x900	Intel Core i5 1.8GHz	8GB	128 Fla Stora
2	HP	250 G6	Notebook	15.6"	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8GB	256 S
3	Apple	MacBook Pro	Ultrabook	15.4"	IPS Panel Retina Display 2880x1800	Intel Core i7 2.7GHz	16GB	512 S
4	Apple	MacBook Pro	Ultrabook	13.3"	IPS Panel Retina Display 2560x1600	Intel Core i5 3.1GHz	8GB	256 S
...	...	...	...	...	...	...	...	
1298	Lenovo	Yoga 500-14ISK	2 in 1 Convertible	14.0"	IPS Panel Full HD / Touchscreen 1920x1080	Intel Core i7 6500U 2.5GHz	4GB	128 S
1299	Lenovo	Yoga 900-13ISK	2 in 1 Convertible	13.3"	IPS Panel Quad HD+ / Touchscreen 3200x1800	Intel Core i7 6500U 2.5GHz	16GB	512 S
1300	Lenovo	IdeaPad 100S-14IBR	Notebook	14.0"	1366x768	Intel Celeron Dual Core N3050 1.6GHz	2GB	64 Fla Stora
1301	HP	15-AC110nv (i7-6500U/6GB/1TB/Radeon	Notebook	15.6"	1366x768	Intel Core i7 6500U 2.5GHz	6GB	1 H
1302	Asus	X553SA-XX031T (N3050/4GB/500GB/W10)	Notebook	15.6"	1366x768	Intel Celeron Dual Core N3050 1.6GHz	4GB	500 H

1303 rows × 13 columns

In [ ]:
```
lapy_test = lapy.copy()
```

In [ ]:
```
lapy_test.columns = ['A', 'B', 'C', 'D', 'E',
                     'F', 'G', 'H', 'I', 'J',
                     'K', 'L', 'M']
```

In [ ]:
```
lapy_test
```

In [ ]:
```
"   deji ".strip()
```

In [5]:
```
column_name = []
for clean in lapy.columns:
    lap = clean.strip()
    lap = lap.replace("Operating System", "os")
    lap = lap.replace("(", "")
    lap = lap.replace(")", "")
    lap = lap.replace(" ", "_")
    lap = lap.lower()
    column_name.append(lap)
column_name
```

Out[5]:
```
['manufacturer',
 'model_name',
 'category',
 'screen_size',
 'screen',
 'cpu',
 'ram',
 'storage',
 'gpu',
 'os',
 'os_version',
 'weight',
 'price_euros']
```

In [7]:
```
lapy.columns = column_name
```

In [8]:
```
lapy.columns
```

Out[8]:
```
Index(['manufacturer', 'model_name', 'category', 'screen_size', 'screen',
       'cpu', 'ram', 'storage', 'gpu', 'os', 'os_version', 'weight',
       'price_euros'],
      dtype='object')
```

In [9]:
```
sub_data = ["category", "screen_size", "screen"]
sub_lapy = lapy.loc[0:5, sub_data]
sub_lapy
```

Out[9]:

	category	screen_size	screen
0	Ultrabook	13.3"	IPS Panel Retina Display 2560x1600
1	Ultrabook	13.3"	1440x900
2	Notebook	15.6"	Full HD 1920x1080
3	Ultrabook	15.4"	IPS Panel Retina Display 2880x1800
4	Ultrabook	13.3"	IPS Panel Retina Display 2560x1600
5	Notebook	15.6"	1366x768

In [10]:
```
lapy
```

Out[10]:

	manufacturer	model_name	category	screen_size	screen	cpu	ram
0	Apple	MacBook Pro	Ultrabook	13.3"	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz	8GB
1	Apple	Macbook Air	Ultrabook	13.3"	1440x900	Intel Core i5 1.8GHz	8GB
2	HP	250 G6	Notebook	15.6"	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8GB
3	Apple	MacBook Pro	Ultrabook	15.4"	IPS Panel Retina Display 2880x1800	Intel Core i7 2.7GHz	16GB
4	Apple	MacBook Pro	Ultrabook	13.3"	IPS Panel Retina Display 2560x1600	Intel Core i5 3.1GHz	8GB
...	...	...	...	...	...	...	...
1298	Lenovo	Yoga 500-14ISK	2 in 1 Convertible	14.0"	IPS Panel Full HD / Touchscreen 1920x1080	Intel Core i7 6500U 2.5GHz	4GB
1299	Lenovo	Yoga 900-13ISK	2 in 1 Convertible	13.3"	IPS Panel Quad HD+ / Touchscreen 3200x1800	Intel Core i7 6500U 2.5GHz	16GB
1300	Lenovo	IdeaPad 100S-14IBR	Notebook	14.0"	1366x768	Intel Celeron Dual Core N3050 1.6GHz	2GB
1301	HP	15-AC110nv (i7-6500U/6GB/1TB/Radeon	Notebook	15.6"	1366x768	Intel Core i7 6500U 2.5GHz	6GB
1302	Asus	X553SA-XX031T (N3050/4GB/500GB/W10)	Notebook	15.6"	1366x768	Intel Celeron Dual Core N3050 1.6GHz	4GB

1303 rows × 13 columns

In [11]:
```
unique_ram = lapy["ram"].unique()
```

In [12]:
```
unique_ram
```

Out[12]:
```
array(['8GB', '16GB', '4GB', '2GB', '12GB', '6GB', '32GB', '24GB', '64GB'],
      dtype=object)
```

In [13]:
```
ram_sz = lapy["ram"].unique()
updated_ram_sz = []
for rm in ram_sz:
    rm1 = rm.replace("GB", "")
    updated_ram_sz.append(rm1)
updated_ram_sz
```

Out[13]:
```
['8', '16', '4', '2', '12', '6', '32', '24', '64']
```

In [14]:
```
unique_ram = updated_ram_sz
```

In [15]:
```
lapy.columns
```

Out[15]:
```
Index(['manufacturer', 'model_name', 'category', 'screen_size', 'screen',
       'cpu', 'ram', 'storage', 'gpu', 'os', 'os_version', 'weight',
       'price_euros'],
      dtype='object')
```

In [16]:
```
lapy["ram"].str.replace('GB', "")
```

Out[16]:
```
0        8
1        8
2        8
3       16
4        8
       ..
1298     4
1299    16
1300     2
1301     6
1302     4
Name: ram, Length: 1303, dtype: object
```

In [18]:
```
lapy["ram"] =(lapy["ram"].str.replace('GB', ""))
```

In [21]:
```
lapy["screen_size"] = (lapy["screen_size"].str.replace('"', ""))
```

In [19]:
```
lapy["weight"] = lapy["weight"].str.replace('kg', "")
```

In [25]:
```
lapy["ram"] = lapy["ram"].astype(float)
```

In [28]:
```
lapy["screen_size"] = lapy["screen_size"].astype(float)
```

In [31]:
```
lapy[["gpu", "manufacturer"]]
```

Out[31]:

	gpu	manufacturer
0	Intel Iris Plus Graphics 640	Apple
1	Intel HD Graphics 6000	Apple
2	Intel HD Graphics 620	HP
3	AMD Radeon Pro 455	Apple
4	Intel Iris Plus Graphics 650	Apple
...	...	...
1298	Intel HD Graphics 520	Lenovo
1299	Intel HD Graphics 520	Lenovo
1300	Intel HD Graphics	Lenovo
1301	AMD Radeon R5 M330	HP
1302	Intel HD Graphics	Asus

1303 rows × 2 columns

In [42]:
```
lapy["gpu"].head().str.split().str[0]
```

Out[42]:
```
0    Intel
1    Intel
2    Intel
3      AMD
4    Intel
Name: gpu, dtype: object
```

In [44]:
```
lapy["gpu"] = lapy["gpu"].str.split().str[0]
```

In [47]:
```
lapy["cpu"] = lapy["cpu"].str.split().str[0]
```

In [48]:
```
lapy
```

Out[48]:

	manufacturer	model_name	category	screen_size	screen	cpu	ram	stora
0	Apple	MacBook Pro	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel	8.0	128 S
1	Apple	Macbook Air	Ultrabook	13.3	1440x900	Intel	8.0	128 Fla Stora
2	HP	250 G6	Notebook	15.6	Full HD 1920x1080	Intel	8.0	256 S
3	Apple	MacBook Pro	Ultrabook	15.4	IPS Panel Retina Display 2880x1800	Intel	16.0	512 S
4	Apple	MacBook Pro	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel	8.0	256 S
...	...	...	...	...	...	...	...	
1298	Lenovo	Yoga 500-14ISK	2 in 1 Convertible	14.0	IPS Panel Full HD / Touchscreen 1920x1080	Intel	4.0	128 S
1299	Lenovo	Yoga 900-13ISK	2 in 1 Convertible	13.3	IPS Panel Quad HD+ / Touchscreen 3200x1800	Intel	16.0	512 S
1300	Lenovo	IdeaPad 100S-14IBR	Notebook	14.0	1366x768	Intel	2.0	64 Fla Stora
1301	HP	15-AC110nv (i7-6500U/6GB/1TB/Radeon	Notebook	15.6	1366x768	Intel	6.0	1 H
1302	Asus	X553SA-XX031T (N3050/4GB/500GB/W10)	Notebook	15.6	1366x768	Intel	4.0	500 H

1303 rows × 13 columns

In [51]:
```
lapy["os"].unique()
```

Out[51]:
```
array(['macOS', 'No OS', 'Windows', 'Mac OS', 'Linux', 'Android',
       'Chrome OS'], dtype=object)
```

In [52]:
```
mapping_dict = {
    'Android': 'Android',
    'Chrome OS': 'Chrome OS',
    'Linux': 'Linux',
    'Mac OS': 'macOS',
    'No OS': 'No OS',
    'Windows': 'Windows',
    'macOS': 'macOS'
}
lapy["os"] = lapy["os"].map(mapping_dict)
```

In [53]:
```
lapy["os"].unique()
```

Out[53]:
```
array(['macOS', 'No OS', 'Windows', 'Linux', 'Android', 'Chrome OS'],
      dtype=object)
```

In [55]:
```
lapy.isnull().sum()
```

Out[55]:
```
manufacturer      0
model_name        0
category          0
screen_size       0
screen            0
cpu               0
ram               0
storage           0
gpu               0
os                0
os_version      170
weight            0
price_euros       0
dtype: int64
```

In [57]:
```
lapy["manufacturer"].isnull().sum()
```

Out[57]:
```
0
```

In [61]:
```
lapy["os_version"].isnull().sum()
```

Out[61]:
```
170
```

In [ ]:
```
y = fortune_500["previous_rank"].isnull()
y_1 = fortune_500[y]
null = y_1[["company", "previous_rank"]]
null
```

In [72]:
```
version = lapy["os_version"].isnull()
the_nul = lapy.loc[version, "os_version"]
the_nul.value_counts()
```

Out[72]:
```
Series([], Name: os_version, dtype: int64)
```

In [69]:
```
the_version = lapy["os_version"].notnull()
not_null = lapy.loc[the_version, "manufacturer":"price_euros"]
```

In [70]:
```
not_null
```

Out[70]:

	manufacturer	model_name	category	screen_size	screen	cpu	ram	stor
5	Acer	Aspire 3	Notebook	15.6	1366x768	AMD	4.0	500 H
6	Apple	MacBook Pro	Ultrabook	15.4	IPS Panel Retina Display 2880x1800	Intel	16.0	256 Fl Stor
8	Asus	ZenBook UX430UN	Ultrabook	14.0	Full HD 1920x1080	Intel	16.0	512 S
9	Acer	Swift 3	Ultrabook	14.0	IPS Panel Full HD 1920x1080	Intel	8.0	256 S
13	Dell	Inspiron 3567	Notebook	15.6	Full HD 1920x1080	Intel	4.0	256 S
...	...	...	...	...	...	...	...	
1298	Lenovo	Yoga 500-14ISK	2 in 1 Convertible	14.0	IPS Panel Full HD / Touchscreen 1920x1080	Intel	4.0	128 S
1299	Lenovo	Yoga 900-13ISK	2 in 1 Convertible	13.3	IPS Panel Quad HD+ / Touchscreen 3200x1800	Intel	16.0	512 S
1300	Lenovo	IdeaPad 100S-14IBR	Notebook	14.0	1366x768	Intel	2.0	64 Fl Stor
1301	HP	15-AC110nv (i7-6500U/6GB/1TB/Radeon	Notebook	15.6	1366x768	Intel	6.0	H
1302	Asus	X553SA-XX031T (N3050/4GB/500GB/W10)	Notebook	15.6	1366x768	Intel	4.0	500 H

1133 rows × 13 columns

In [71]:
```
not_null.isnull().sum()
```

Out[71]:
```
manufacturer    0
model_name      0
category        0
screen_size     0
screen          0
cpu             0
ram             0
storage         0
gpu             0
os              0
os_version      0
weight          0
price_euros     0
dtype: int64
```

In [ ]:
```
a = [2, 4, 6, 8, 10]
```

In [ ]:
```
df.to_csv('filename.csv', index=False)
```