

```
print("Hello World")
```

Hello World

```
import pandas as pd
import matplotlib as plt
import seaborn as sb
```

```
import pandas as pd
taxi_owners = pd.read_pickle("taxi_owners.p")
taxi_vehivles = pd.read_pickle("taxi_vehicles.p")
```

```
taxi_own_vehicles = taxi_owners.merge(taxi_vehivles, on= 'vid')
taxi_own_vehicles.head(5)
```

	rid	vid	owner_x	address	zip	make
model \						
0	T6285	6285	AGEAN TAXI LLC	4536 N. ELSTON AVE.	60630	NISSAN
						ALTIMA
1	T4862	4862	MANGIB CORP.	5717 N. WASHTENAW AVE.	60659	HONDA
						CRV
2	T1495	1495	FUNRIDE, INC.	3351 W. ADDISON ST.	60618	TOYOTA
						SIENNA
3	T4231	4231	ALQUSH CORP.	6611 N. CAMPBELL AVE.	60645	TOYOTA
						CAMRY
4	T5971	5971	EUNIFFORD INC.	3351 W. ADDISON ST.	60618	TOYOTA
						SIENNA

	year	fuel_type	owner_y
0	2011	HYBRID	AGEAN TAXI LLC
1	2014	GASOLINE	MANGIB CORP.
2	2015	GASOLINE	FUNRIDE, INC.
3	2014	HYBRID	ALQUSH CORP.
4	2015	GASOLINE	EUNIFFORD INC.

```
wards_altered = pd.read_csv("Wards_Offices_Altered.csv")
census = pd.read_csv("Wards_Census.csv")
```

	ward	alderman	address_x	zip_x
pop_2000 \				
0	1	Proco "Joe" Moreno	2058 NORTH WESTERN AVENUE	60647
				52951
1	2	Brian Hopkins	1400 NORTH ASHLAND AVENUE	60622
				54361

	pop_2010	change	address_y	zip_y
0	56149	6%	2765 WEST SAINT MARY STREET	60647
1	55805	3%	WM WASTE MANAGEMENT 1500	60622

```
wards_altered.head()
```

	ward	alderman	address	zip
0	1	Proco "Joe" Moreno	2058 NORTH WESTERN AVENUE	60647
1	2	Brian Hopkins	1400 NORTH ASHLAND AVENUE	60622
2	3	Pat Dowell	5046 SOUTH STATE STREET	60609
3	4	William D. Burns	435 EAST 35TH STREET, 1ST FLOOR	60616
4	5	Leslie A. Hairston	2325 EAST 71ST STREET	60649

```
census.head()
```

	ward	pop_2000	pop_2010	change	address
0	1	52951	56149	6%	2765 WEST SAINT MARY STREET
1	2	54361	55805	3%	WM WASTE MANAGEMENT 1500
2	3	40385	53039	31%	17 EAST 38TH STREET
3	4	51953	54589	5%	31ST ST HARBOR BUILDING LAKEFRONT TRAIL
4	5	55302	51455	-7%	JACKSON PARK LAGOON SOUTH CORNELL DRIVE

	zip
0	60647
1	60622
2	60653
3	60653
4	60637

```
census.shape
```

```
(50, 6)
```

```
wards_altered_census = wards_altered.merge(census, on= 'ward')
wards_altered_census.head(2)
```

	ward	alderman	address_x	zip_x
pop_2000	\			
0	1	Proco "Joe" Moreno	2058 NORTH WESTERN AVENUE	60647
52951				
1	2	Brian Hopkins	1400 NORTH ASHLAND AVENUE	60622
54361				

	pop_2010	change	address_y	zip_y
0	56149	6%	2765 WEST SAINT MARY STREET	60647
1	55805	3%	WM WASTE MANAGEMENT 1500	60622

```
liscenses = pd.read_csv("Business_Licenses.csv")
print(liscenses.head())
print(liscenses.shape)
```

	account	ward	aid	business	address
\					
0	307071	3	743	REGGIE'S BAR & GRILL	2105 S STATE ST
1	10	10	829	HONEYBEERS	13200 S HOUSTON AVE
2	10002	14	775	CELINA DELI	5089 S ARCHER AVE
3	10005	12	NaN	KRAFT FOODS NORTH AMERICA	2005 W 43RD ST
4	10044	44	638	NEYBOUR'S TAVERN & GRILLE	3651 N SOUTHPORT AVE

```

zip
0 60616.0
1 60633.0
2 60632.0
3 60609.0
4 60613.0
(10000, 6)

```

```

wards = pd.read_csv("Wards_Offices.csv")
wards_liscenes = wards.merge(licenses,on = "ward",
suffixes=('_ward','_lis'))
wards_liscenes.head()
wards_liscenes.shape
wards.shape

```

```
(50, 4)
```

```

licenses = pd.read_pickle("licenses.p")
biz_owners = pd.read_pickle("business_owners.p")

```

```

print(licenses.head())
print(licenses.shape)

```

```

print(biz_owners.head())
print(biz_owners.shape)

```

	account	ward	aid	business	address
zip					
0	307071	3	743	REGGIE'S BAR & GRILL	2105 S STATE ST
60616					
1	10	10	829	HONEYBEERS	13200 S HOUSTON AVE
60633					
2	10002	14	775	CELINA DELI	5089 S ARCHER AVE
60632					
3	10005	12	NaN	KRAFT FOODS NORTH AMERICA	2005 W 43RD ST
60609					
4	10044	44	638	NEYBOUR'S TAVERN & GRILLE	3651 N SOUTHPORT AVE
60613					

```
(10000, 6)
  account first_name last_name title
0      10      PEARL  SHERMAN  PRESIDENT
1      10      PEARL  SHERMAN  SECRETARY
2    10002      WALTER   MROZEK   PARTNER
3    10002      CELINA   BYRDAK   PARTNER
4    10005       IRENE  ROSENFELD  PRESIDENT
(21352, 4)
```

```
licenses_owners = licenses.merge(biz_owners, on = "account")
print(licenses_owners.head())
```

	account	ward	aid	business	address	zip
0	307071	3	743	REGGIE'S BAR & GRILL	2105 S STATE ST	60616
1	10	10	829	HONEYBEERS	13200 S HOUSTON AVE	60633
2	10	10	829	HONEYBEERS	13200 S HOUSTON AVE	60633
3	10002	14	775	CELINA DELI	5089 S ARCHER AVE	60632
4	10002	14	775	CELINA DELI	5089 S ARCHER AVE	60632

	first_name	last_name	title
0	ROBERT	GLICK	MEMBER
1	PEARL	SHERMAN	PRESIDENT
2	PEARL	SHERMAN	SECRETARY
3	WALTER	MROZEK	PARTNER
4	CELINA	BYRDAK	PARTNER

```
counted_df =
licenses_owners.groupby('title').agg({'account': 'count'})
sorted_df = counted_df.sort_values('account', ascending = True)
sorted_df.head()
```

	account
title	
BENEFICIARY	4
TRUSTEE	6
EXECUTIVE DIRECTOR	10
NOT APPLICABLE	11
GENERAL PARTNER	21

```
cal = pd.read_pickle("cta_calendar.p")
ridership = pd.read_pickle("cta_ridership.p")
stations = pd.read_pickle("stations.p")
```

```
print(cal.head())
print(ridership.head())
```

```

print(stations.head())

ridership_cal_stations = ridership.merge(cal, on =
['year', 'month', 'day']).merge(stations, on = 'station_id')

filter_criteria = ((ridership_cal_stations['month'] == 7)
                    & (ridership_cal_stations['station_name'] ==
'Wilson')
                    & (ridership_cal_stations['day_type'] ==
'Weekday'))

ridership_cal_stations.loc[filter_criteria, 'rides'].sum()

```

	year	month	day	day_type
0	2019	1	1	Sunday/Holiday
1	2019	1	2	Weekday
2	2019	1	3	Weekday
3	2019	1	4	Weekday
4	2019	1	5	Saturday

	station_id	year	month	day	rides
0	40010	2019	1	1	576
1	40010	2019	1	2	1457
2	40010	2019	1	3	1543
3	40010	2019	1	4	1621
4	40010	2019	1	5	719

	station_id	station_name	location
0	40010	Austin-Forest Park	(41.870851, -87.776812)
1	40020	Harlem-Lake	(41.886848, -87.803176)
2	40030	Pulaski-Lake	(41.885412, -87.725404)
3	40040	Quincy/Wells	(41.878723, -87.63374)
4	40050	Davis	(42.04771, -87.683543)

```

np.int64(140005)

licenses = pd.read_pickle("licenses.p")
wards = pd.read_pickle("ward.p")
zip_demo = pd.read_pickle("zip_demo.p")

```

```

print(licenses.head())
print(wards.head())
print(zip_demo.head())

```

	account	ward	aid	business	address
zip					
0	307071	3	743	REGGIE'S BAR & GRILL	2105 S STATE ST 60616
1	10	10	829	HONEYBEERS	13200 S HOUSTON AVE 60633
2	10002	14	775	CELINA DELI	5089 S ARCHER AVE 60632
3	10005	12	NaN	KRAFT FOODS NORTH AMERICA	2005 W 43RD ST

```
60609
4 10044 44 638 NEYBOUR'S TAVERN & GRILLE 3651 N SOUTHPORT AVE
60613
```

	ward	alderman	address	zip
0	1	Proco "Joe" Moreno	2058 NORTH WESTERN AVENUE	60647
1	2	Brian Hopkins	1400 NORTH ASHLAND AVENUE	60622
2	3	Pat Dowell	5046 SOUTH STATE STREET	60609
3	4	William D. Burns	435 EAST 35TH STREET, 1ST FLOOR	60616
4	5	Leslie A. Hairston	2325 EAST 71ST STREET	60649

	zip	income
0	60630	70122
1	60640	50488
2	60622	87143
3	60614	100116
4	60608	41226

```
liscenses_wards_zip =
liscenses.merge(zip_demo,on="zip").merge(wards,on="ward")
liscenses_wards_zip.head()
```

	account	ward	aid	business	address_x
zip_x \					
0	307071	3	743	REGGIE'S BAR & GRILL	2105 S STATE ST
60616					
1	10	10	829	HONEYBEERS	13200 S HOUSTON AVE
60633					
2	10002	14	775	CELINA DELI	5089 S ARCHER AVE
60632					
3	10005	12	NaN	KRAFT FOODS NORTH AMERICA	2005 W 43RD ST
60609					
4	10044	44	638	NEYBOUR'S TAVERN & GRILLE	3651 N SOUTHPORT AVE
60613					

	income	alderman	address_y	zip_y
0	46340	Pat Dowell	5046 SOUTH STATE STREET	60609
1	50164	Susan Sadłowski Garza	10500 SOUTH EWING AVENUE	60617
2	42335	Edward M. Burke	2650 WEST 51ST STREET	60632
3	33959	George Cardenas	3476 SOUTH ARCHER AVENUE	60608
4	79565	Tom Tunney	3223 NORTH SHEFFIELD AVENUE	60657

```
liscenses_wards_zip.groupby('alderman').agg({'income':'median'})
```

	income
alderman	
Ameya Pawar	66246.0
Anthony A. Beale	38206.0
Anthony V. Napolitano	82226.0
Ariel E. Reyboras	41307.0
Brendan Reilly	110215.0
Brian Hopkins	87143.0

Carlos Ramirez-Rosa	66246.0
Carrie M. Austin	38206.0
Chris Taliaferro	55566.0
Daniel "Danny" Solis	41226.0
David H. Moore	33304.0
Deborah Mell	66246.0
Debra L. Silverstein	50554.0
Derrick G. Curtis	65770.0
Edward M. Burke	42335.0
Emma M. Mitts	36283.0
George Cardenas	33959.0
Gilbert Villegas	41307.0
Gregory I. Mitchell	24941.0
Harry Osterman	45442.0
Howard B. Brookins, Jr.	33304.0
James Cappleman	79565.0
Jason C. Ervin	41226.0
Joe Moore	39163.0
John S. Arena	70122.0
Leslie A. Hairston	28024.0
Margaret Laurino	70122.0
Marty Quinn	67045.0
Matthew J. O'Shea	59488.0
Michael R. Zalewski	42335.0
Michael Scott, Jr.	31445.0
Michelle A. Harris	32558.0
Michelle Smith	100116.0
Milagros "Milly" Santiago	41307.0
Nicholas Sposato	62223.0
Pat Dowell	46340.0
Patrick Daley Thompson	41226.0
Patrick J. O'Connor	50554.0
Proco "Joe" Moreno	87143.0
Raymond A. Lopez	33959.0
Ricardo Munoz	31445.0
Roberto Maldonado	68223.0
Roderick T. Sawyer	32558.0
Scott Waguespack	68223.0
Susan Sadlowski Garza	38417.0
Tom Tunney	88708.0
Toni L. Foulkes	27573.0
Walter Burnett, Jr.	87143.0
William D. Burns	107811.0
Willie B. Cochran	28024.0

```

movies = pd.read_csv("tmdb_movies.csv")
print(movies.head())
print(movies.shape)

taglines = pd.read_csv("tdmb_taglines.csv")

```

```
print(taglines.head())
print(taglines.shape)

movie_tagline = movies.merge(taglines, on = 'id', how='left')
movie_tagline.head()
```

	id		title	popularity	release_date
0	257		Oliver Twist	20.415572	23/9/2005
1	14290	Better Luck Tomorrow		3.877036	12/1/2002
2	38365		Grown Ups	38.864027	24/6/2010
3	9672		Infamous	3.680896	16/11/2006
4	12819	Alpha and Omega		12.300789	17/9/2010

(4803, 4)

	id		tagline
0	19995		Enter the World of Pandora.
1	285	At the end of the world, the adventure begins.	
2	206647		A Plan No One Escapes
3	49026		The Legend Ends
4	49529	Lost in our world, found in another.	

(3955, 2)

	id		title	popularity	release_date	\
0	257		Oliver Twist	20.415572	23/9/2005	
1	14290	Better Luck Tomorrow		3.877036	12/1/2002	
2	38365		Grown Ups	38.864027	24/6/2010	
3	9672		Infamous	3.680896	16/11/2006	
4	12819	Alpha and Omega		12.300789	17/9/2010	

	tagline
0	NaN
1	Never underestimate an overachiever.
2	Boys will be boys. . . some longer than others.
3	There's more to the story than you know
4	A Pawsome 3D Adventure

```
movies = pd.read_pickle("movies.p")
print(movies.head())
print(movies.shape)
```

```
financials = pd.read_pickle("financials.p")
print(financials.head())
print(financials.shape)
```

	id		title	popularity	release_date
0	257		Oliver Twist	20.415572	2005-09-23
1	14290	Better Luck Tomorrow		3.877036	2002-01-12
2	38365		Grown Ups	38.864027	2010-06-24
3	9672		Infamous	3.680896	2006-11-16
4	12819	Alpha and Omega		12.300789	2010-09-17

(4803, 4)

	id	budget	revenue
0	19995	237000000	2.787965e+09
1	285	300000000	9.610000e+08
2	206647	245000000	8.806746e+08
3	49026	250000000	1.084939e+09
4	49529	260000000	2.841391e+08

(3229, 3)

```
movies_financials = movies.merge(financials,on='id',how='left')
movies_financials.head()
```

	id	title	popularity	release_date	budget	\
0	257	Oliver Twist	20.415572	2005-09-23	50000000.0	
1	14290	Better Luck Tomorrow	3.877036	2002-01-12	NaN	
2	38365	Grown Ups	38.864027	2010-06-24	80000000.0	
3	9672	Infamous	3.680896	2006-11-16	13000000.0	
4	12819	Alpha and Omega	12.300789	2010-09-17	20000000.0	

	revenue
0	42093706.0
1	NaN
2	271430189.0
3	1151330.0
4	39300000.0

```
null_val = movies_financials['budget'].isnull().sum()
```

```
print(null_val)
```

1574

```
toy_story = pd.read_csv("toy_story.csv")
print(toy_story.head())
print(toy_story.shape)
```

```
taglines = pd.read_pickle("taglines.p")
print(taglines.head())
print(taglines.shape)
```

	id	title	popularity	release_date
0	10193	Toy Story 3	59.995	16/6/2010
1	863	Toy Story 2	73.575	30/10/1999
2	862	Toy Story	73.640	30/10/1995

(3, 4)

	id	tagline
0	19995	Enter the World of Pandora.
1	285	At the end of the world, the adventure begins.
2	206647	A Plan No One Escapes
3	49026	The Legend Ends
4	49529	Lost in our world, found in another.

(3955, 2)

```

toystory_tag = toy_story.merge(taglines,on='id')
toystory_tag.shape

(2, 5)

movies = pd.read_pickle('movies.p')
movies_to_genre = pd.read_csv('tdmb_movie_to_genres.csv')

print(movies.head())
print(movies_to_genre.head())

m = movies_to_genre['genre'] == 'Science Fiction'
scifi_movies = movies_to_genre[m]
print(scifi_movies)

m2 = movies_to_genre['genre'] == 'Action'
action_movies = movies_to_genre[m2]
print(action_movies)

```

	id		title	popularity	release_date
0	257		Oliver Twist	20.415572	2005-09-23
1	14290	Better Luck Tomorrow		3.877036	2002-01-12
2	38365		Grown Ups	38.864027	2010-06-24
3	9672		Infamous	3.680896	2006-11-16
4	12819	Alpha and Omega		12.300789	2010-09-17

	movie_id	genre
0	5	Crime
1	5	Comedy
2	11	Science Fiction
3	11	Action
4	11	Adventure

	movie_id	genre
2	11	Science Fiction
17	18	Science Fiction
20	19	Science Fiction
38	38	Science Fiction
49	62	Science Fiction

12000	335866	Science Fiction	
12020	347548	Science Fiction	
12063	360188	Science Fiction	
12092	367551	Science Fiction	
12100	371690	Science Fiction	

[535 rows x 2 columns]

	movie_id	genre
3	11	Action
14	18	Action
25	22	Action
26	24	Action

```

42          58  Action
...
12126      381902  Action
12128      385383  Action
12136      389425  Action
12144      407887  Action
12151      417859  Action

```

```
[1154 rows x 2 columns]
```

```

action_scifi =
action_movies.merge(scifi_movies,on="movie_id",how='right',suffixes=('
_act','_sci'))
print(action_scifi)

```

```

      movie_id genre_act      genresci
0          11    Action  Science Fiction
1          18    Action  Science Fiction
2          19      NaN  Science Fiction
3          38      NaN  Science Fiction
4          62      NaN  Science Fiction
..
530      335866      NaN  Science Fiction
531      347548      NaN  Science Fiction
532      360188      NaN  Science Fiction
533      367551    Action  Science Fiction
534      371690      NaN  Science Fiction

```

```
[535 rows x 3 columns]
```

```

scifi_only = action_scifi[action_scifi["genre_act"].isnull()]
scifi_only.head()

```

```

      movie_id genre_act      genresci
2          19      NaN  Science Fiction
3          38      NaN  Science Fiction
4          62      NaN  Science Fiction
5          68      NaN  Science Fiction
6          74      NaN  Science Fiction

```

```
movies.merge(scifi_only,how='inner',left_on='id',right_on='movie_id')
```

```

      id      title  popularity  release_date
movie_id \
0      18841  The Lost Skeleton of Cadavra    1.680525    2001-09-12
18841
1      26672    The Thief and the Cobbler    2.439184    1993-09-23
26672
2      15301  Twilight Zone: The Movie    12.902975    1983-06-24
15301
3       8452      The 6th Day    18.447479    2000-11-17

```

```

8452
4      1649      Bill & Ted's Bogus Journey      11.349664      1991-07-19
1649
...      ...      ...      ...      ...
...
253  245703      Midnight Special      32.717853      2016-02-18
245703
254      3509      A Scanner Darkly      26.093043      2006-05-25
3509
255      42188      Never Let Me Go      30.983397      2010-09-15
42188
256      18045      The Dark Hours      1.428483      2005-03-11
18045
257      11058      Godsend      12.102350      2004-04-30
11058

```

```

      genre_act      genresci
0      NaN      Science Fiction
1      NaN      Science Fiction
2      NaN      Science Fiction
3      NaN      Science Fiction
4      NaN      Science Fiction
...      ...      ...
253      NaN      Science Fiction
254      NaN      Science Fiction
255      NaN      Science Fiction
256      NaN      Science Fiction
257      NaN      Science Fiction

```

```
[258 rows x 7 columns]
```

```

pop_movies = pd.read_csv("pop_movies.csv")
pop_movies.head()

```

```

genre_movies =
movies_to_genre.merge(pop_movies,how='right',left_on='movie_id',right_
on='id')
genre_movies.head()

```

```

count = genre_movies.groupby('genre').agg({'id':'count'})
count.plot(kind='bar')
plt.show()

```

```

-----
-----
AttributeError                                Traceback (most recent call
last)
Cell In[11], line 9
      7 count = genre_movies.groupby('genre').agg({'id':'count'})

```

```

8 count.plot(kind='bar')
----> 9 plt.show()

```

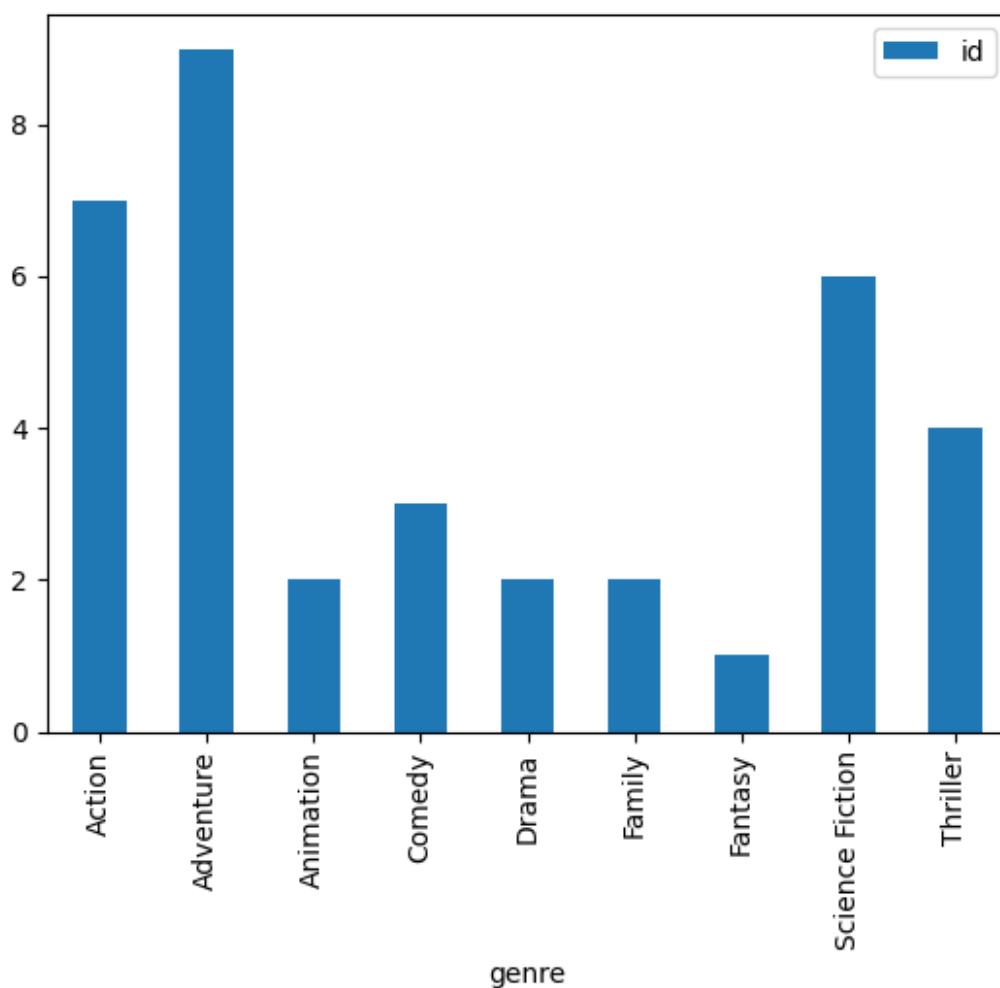
File

```

/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/
site-packages/matplotlib/_api/__init__.py:217, in
caching_module_getattr.<locals>.__getattr__(name)
    215 if name in props:
    216     return props[name].__get__(instance)
--> 217 raise AttributeError(
    218     f"module {cls.__module__!r} has no attribute {name!r}")

```

AttributeError: module 'matplotlib' has no attribute 'show'



```

crews = pd.read_pickle("crews.p")
crews.head()

```

	id	department	job	name
0	19995	Editing	Editor	Stephen E. Rivkin

2	19995	Sound	Sound Designer	Christopher Boyes
4	19995	Production	Casting	Mali Finn
6	19995	Directing	Director	James Cameron
7	19995	Writing	Writer	James Cameron

```
crews_self_merged =
crews.merge(crews,on='id',suffixes=('_dir','_crew'))
```

```
boolean_filter = ((crews_self_merged['job_dir'] == "Director") &
(crews_self_merged['job_crew'] != 'Director'))
```

```
direct_crews = crews_self_merged[boolean_filter]
direct_crews.head()
```

	id	department_dir	job_dir	name_dir	department_crew	\
156	19995	Directing	Director	James Cameron	Editing	
157	19995	Directing	Director	James Cameron	Sound	
158	19995	Directing	Director	James Cameron	Production	
160	19995	Directing	Director	James Cameron	Writing	
161	19995	Directing	Director	James Cameron	Art	

	job_crew	name_crew
156	Editor	Stephen E. Rivkin
157	Sound Designer	Christopher Boyes
158	Casting	Mali Finn
160	Writer	James Cameron
161	Set Designer	Richard F. Mays

```
tracks_master = pd.read_csv("tracks_master.csv")
tracks_ride = pd.read_csv("tracks_master.csv")
tracks_st = pd.read_csv("tracks_st.csv")
```

```
print(tracks_master.head())
```

```
print(tracks_ride.head())
```

```
print(tracks_st.head())
```

	tid	name	aid	mtid	gid	composer
u_price						
0	1853	Battery	152	1	3	J.Hetfield/L.Ulrich
0.99						
1	1854	Master Of Puppets	152	1	3	K.Hammett
0.99						
2	1857	Disposable Heroes	152	1	3	J.Hetfield/L.Ulrich
0.99						
tid		name	aid	mtid	gid	composer
u_price						
0	1853	Battery	152	1	3	J.Hetfield/L.Ulrich
0.99						
1	1854	Master Of Puppets	152	1	3	K.Hammett

```
0.99
2 1857 Disposable Heroes 152 1 3 J.Hetfield/L.Ulrich
0.99
```

	tid	name	aid	mtid	gid	u_price
0	1882	Frantic	155	1	3	0.99
1	1883	St. Anger	155	1	3	0.99
2	1884	Some Kind Of Monster	155	1	3	0.99
3	1885	Dirty Window	155	1	3	0.99
4	1886	Invisible Kid	155	1	3	0.99

```
pd.concat([tracks_master,tracks_ride,tracks_st],sort = True)
```

	aid	composer	gid	mtid	name	tid
u_price						
0	152	J.Hetfield/L.Ulrich	3	1	Battery	1853
0.99						
1	152	K.Hammitt	3	1	Master Of Puppets	1854
0.99						
2	152	J.Hetfield/L.Ulrich	3	1	Disposable Heroes	1857
0.99						
0	152	J.Hetfield/L.Ulrich	3	1	Battery	1853
0.99						
1	152	K.Hammitt	3	1	Master Of Puppets	1854
0.99						
2	152	J.Hetfield/L.Ulrich	3	1	Disposable Heroes	1857
0.99						
0	155	NaN	3	1	Frantic	1882
0.99						
1	155	NaN	3	1	St. Anger	1883
0.99						
2	155	NaN	3	1	Some Kind Of Monster	1884
0.99						
3	155	NaN	3	1	Dirty Window	1885
0.99						
4	155	NaN	3	1	Invisible Kid	1886
0.99						

```
pd.concat([tracks_master,tracks_ride,tracks_st],ignore_index=False)
```

	tid	name	aid	mtid	gid	composer
u_price						
0	1853	Battery	152	1	3	J.Hetfield/L.Ulrich
0.99						
1	1854	Master Of Puppets	152	1	3	K.Hammitt
0.99						
2	1857	Disposable Heroes	152	1	3	J.Hetfield/L.Ulrich
0.99						
0	1853	Battery	152	1	3	J.Hetfield/L.Ulrich
0.99						
1	1854	Master Of Puppets	152	1	3	K.Hammitt

0.99						
2	1857	Disposable Heroes	152	1	3	J.Hetfield/L.Ulrich
0.99						
0	1882	Frantic	155	1	3	NaN
0.99						
1	1883	St. Anger	155	1	3	NaN
0.99						
2	1884	Some Kind Of Monster	155	1	3	NaN
0.99						
3	1885	Dirty Window	155	1	3	NaN
0.99						
4	1886	Invisible Kid	155	1	3	NaN
0.99						

```
pd.concat([tracks_master,tracks_ride,tracks_st],join='inner')
```

	tid	name	aid	mtid	gid	u_price
0	1853	Battery	152	1	3	0.99
1	1854	Master Of Puppets	152	1	3	0.99
2	1857	Disposable Heroes	152	1	3	0.99
0	1853	Battery	152	1	3	0.99
1	1854	Master Of Puppets	152	1	3	0.99
2	1857	Disposable Heroes	152	1	3	0.99
0	1882	Frantic	155	1	3	0.99
1	1883	St. Anger	155	1	3	0.99
2	1884	Some Kind Of Monster	155	1	3	0.99
3	1885	Dirty Window	155	1	3	0.99
4	1886	Invisible Kid	155	1	3	0.99

```
inv_jul =pd.read_csv("inv_jul.csv")
inv_aug = pd.read_csv("inv_aug.csv")
inv_sep = pd.read_csv("inv_sep.csv")
```

```
avg_inv_by_month =
pd.concat([inv_jul,inv_aug,inv_sep],keys=['7July','8Aug','9Sep'])
```

```
avg = avg_inv_by_month.groupby(level=0).agg({'total':'mean'})
avg.plot(kind='bar')
plt.show()
```

```
-----
-----
AttributeError                                Traceback (most recent call
last)
Cell In[26], line 11
      9 avg = avg_inv_by_month.groupby(level=0).agg({'total':'mean'})
     10 avg.plot(kind='bar')
--> 11 plt.show()
```

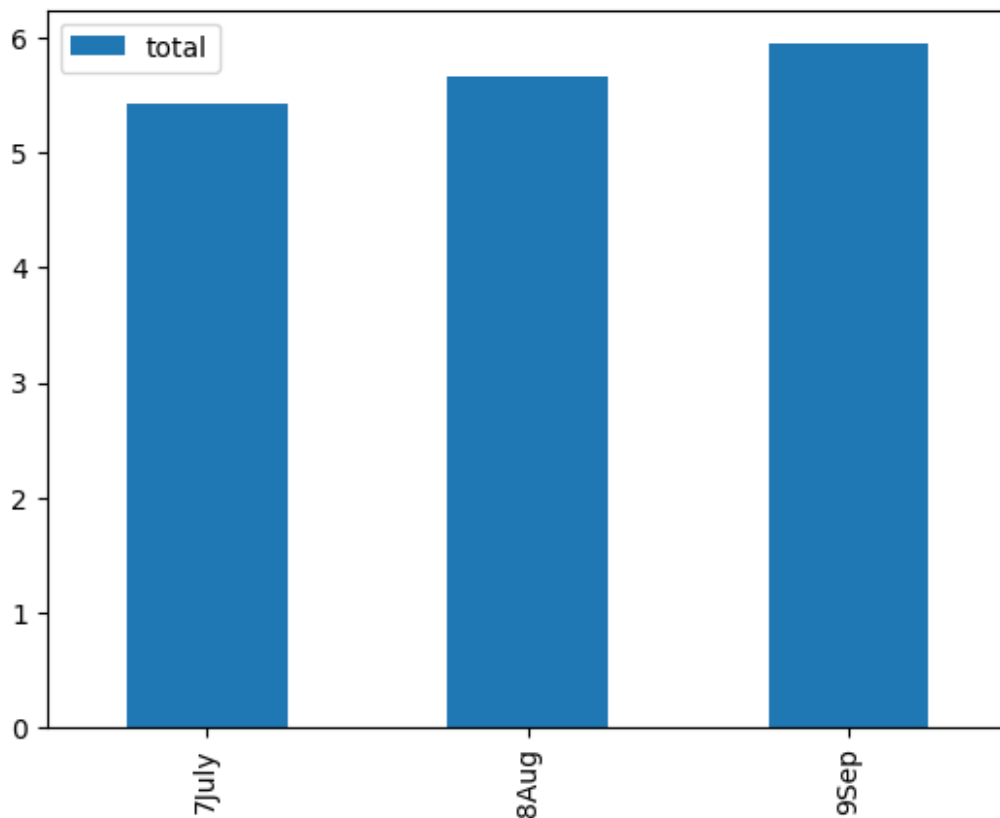


```

File
/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/
site-packages/matplotlib/_api/__init__.py:217, in
caching_module_getattr.<locals>.__getattr__(name)
    215 if name in props:
    216     return props[name].__get__(instance)
--> 217 raise AttributeError(
    218     f"module {cls.__module__!r} has no attribute {name!r}")

AttributeError: module 'matplotlib' has no attribute 'show'

```



```

sp500 = pd.read_csv("S&P500.csv")
print(sp500.head())

```

```

  date  returns
0  2008   -38.49
1  2009    23.45
2  2010    12.78
3  2011     0.00
4  2012    13.41

```

```

gdp = pd.read_csv("GDP.csv")
print(gdp.head())

```

	Unnamed: 0	country	code	year	gdp
0	3	USA	2010	1.500000e+13	
1	7	USA	2011	1.550000e+13	
2	11	USA	2012	1.620000e+13	
3	15	USA	2012	1.620000e+13	
4	19	USA	2013	1.680000e+13	

```
from pandas import merge_ordered
```

```
gdp_sp500 =
merge_ordered(gdp,sp500,left_on='year',right_on='date',how='left')
print(gdp_sp500)
```

	Unnamed: 0	country	code	year	gdp	date	returns
0	3	USA	2010	1.500000e+13	2010.0	12.78	
1	7	USA	2011	1.550000e+13	2011.0	0.00	
2	11	USA	2012	1.620000e+13	2012.0	13.41	
3	15	USA	2012	1.620000e+13	2012.0	13.41	
4	19	USA	2013	1.680000e+13	2013.0	29.60	
5	23	USA	2014	1.750000e+13	2014.0	11.39	
6	27	USA	2015	1.820000e+13	2015.0	-0.73	
7	31	USA	2016	1.870000e+13	2016.0	9.54	
8	35	USA	2017	1.950000e+13	2017.0	19.42	
9	39	USA	2018	2.050000e+13	NaN	NaN	

```
gdp_sp500 =
merge_ordered(gdp,sp500,left_on='year',right_on='date',how='left',fill
_method='ffill')
print(gdp_sp500)
```

```
gdp_returns = gdp_sp500[['gdp','returns']]
```

```
print(gdp_returns.corr(numeric_only=True))
```

	Unnamed: 0	country	code	year	gdp	date	returns
0	3	USA	2010	1.500000e+13	2010	12.78	
1	7	USA	2011	1.550000e+13	2011	0.00	
2	11	USA	2012	1.620000e+13	2012	13.41	
3	15	USA	2012	1.620000e+13	2012	13.41	
4	19	USA	2013	1.680000e+13	2013	29.60	
5	23	USA	2014	1.750000e+13	2014	11.39	
6	27	USA	2015	1.820000e+13	2015	-0.73	
7	31	USA	2016	1.870000e+13	2016	9.54	
8	35	USA	2017	1.950000e+13	2017	19.42	
9	39	USA	2018	2.050000e+13	2017	19.42	

	gdp	returns
gdp	1.000000	0.220321
returns	0.220321	1.000000

```
unemployment = pd.read_csv("unemployment.csv")
inflation = pd.read_csv("inflation.csv")
```

```
print(unemployment.head())
print(inflation.head())
```

	date	unemployment_rate
0	1/6/2013	7.5
1	1/1/2014	6.7
2	1/6/2014	6.1
3	1/1/2015	5.6
4	1/6/2015	5.3

	date	cpi	seriesid	data_type
0	1/1/2014	235.288	CUSR0000SA0	SEASONALLY ADJUSTED INDEX
1	1/2/2014	235.547	CUSR0000SA0	SEASONALLY ADJUSTED INDEX
2	1/3/2014	236.028	CUSR0000SA0	SEASONALLY ADJUSTED INDEX
3	1/4/2014	236.468	CUSR0000SA0	SEASONALLY ADJUSTED INDEX
4	1/5/2014	236.918	CUSR0000SA0	SEASONALLY ADJUSTED INDEX

```
inflation_unemploy =
merge_ordered(inflation,unemployment,on='date',how='inner')
inflation_unemploy.head()
```

	date	cpi	seriesid	data_type
0	1/1/2014	235.288	CUSR0000SA0	SEASONALLY ADJUSTED INDEX
1	1/1/2015	234.718	CUSR0000SA0	SEASONALLY ADJUSTED INDEX
2	1/1/2016	237.833	CUSR0000SA0	SEASONALLY ADJUSTED INDEX
3	1/1/2017	243.780	CUSR0000SA0	SEASONALLY ADJUSTED INDEX
4	1/1/2018	248.884	CUSR0000SA0	SEASONALLY ADJUSTED INDEX

	unemployment_rate
0	6.7
1	5.6
2	5.0
3	4.7
4	4.1

```
import seaborn as sns
sns.scatterplot(inflation_unemploy,x='unemployment_rate',y='cpi')
plt.show()
```

```
-----
-----
AttributeError                                Traceback (most recent call
last)
```

```
Cell In[42], line 3
```

```
1 import seaborn as sns
2
```

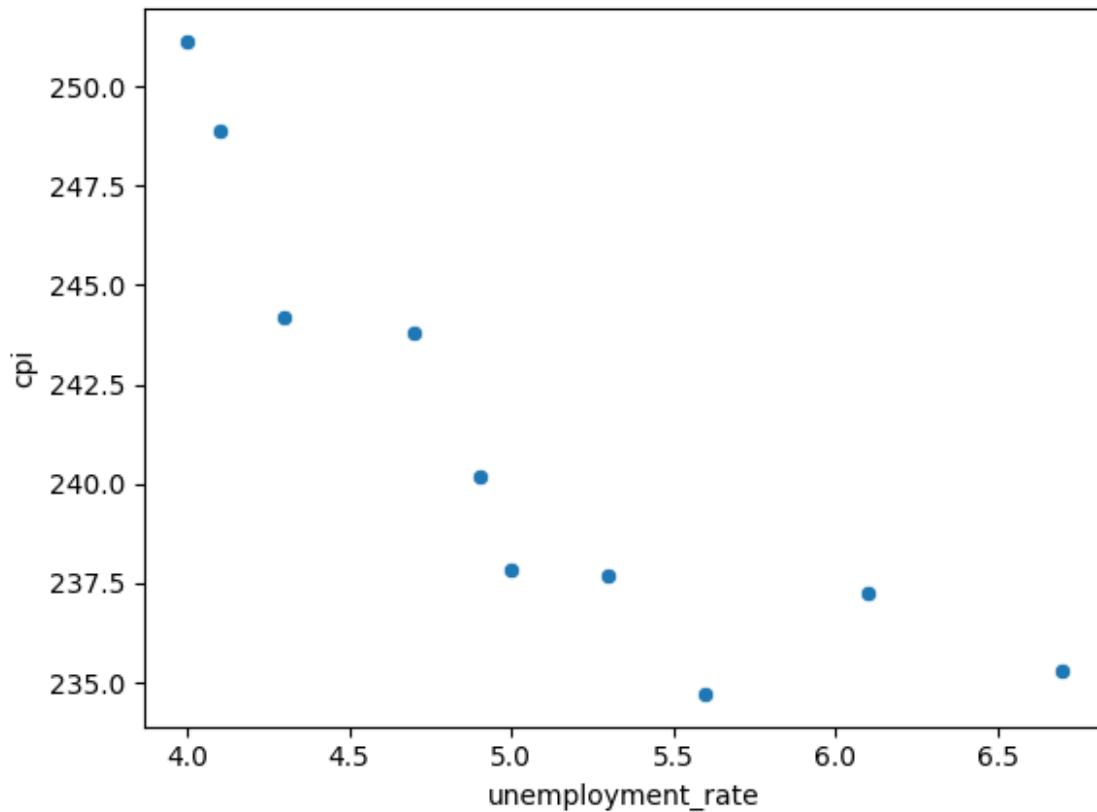
```
sns.scatterplot(inflation_unemploy,x='unemployment_rate',y='cpi')
----> 3 plt.show()
```

```
File
```

```

/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/
site-packages/matplotlib/_api/__init__.py:217, in
caching_module_getattr.<locals>.__getattr__(name)
    215 if name in props:
    216     return props[name].__get__(instance)
--> 217 raise AttributeError(
    218     f"module {cls.__module__!r} has no attribute {name!r}")
AttributeError: module 'matplotlib' has no attribute 'show'

```



```

corr = inflation_unemploy[['unemployment_rate', 'cpi']]
print(corr.corr(numeric_only=True))

```

	unemployment_rate	cpi
unemployment_rate	1.000000	-0.868388
cpi	-0.868388	1.000000