In [3]:
```
import pandas as pd

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

```
   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
(50, 4)
```

In [4]:
```
census = pd.read_csv("Wards_Census.csv")
print(census.head())
print(census.shape)
```

```
   ward  pop_2000  pop_2010 change                                    addre
ss  \
0     1     52951     56149     6%                 2765 WEST SAINT MARY STRE
ET
1     2     54361     55805     3%                    WM WASTE MANAGEMENT 15
00
2     3     40385     53039    31%                          17 EAST 38TH STRE
ET
3     4     51953     54589     5%  31ST ST HARBOR BUILDING LAKEFRONT TRA
IL
4     5     55302     51455    -7%  JACKSON PARK LAGOON SOUTH CORNELL DRI
VE

     zip
0  60647
1  60622
2  60653
3  60653
4  60637
(50, 6)
```

In [5]:
```
wards_census = wards.merge(census, on="ward")
print(wards_census.head(4))
```

```
   ward            alderman                          address_x  zip_x  pop_
2000  \
0     1  Proco "Joe" Moreno        2058 NORTH WESTERN AVENUE  60647     5
2951
1     2       Brian Hopkins       1400 NORTH  ASHLAND AVENUE  60622     5
4361
2     3          Pat Dowell          5046 SOUTH STATE STREET  60609     4
0385
3     4    William D. Burns  435 EAST 35TH STREET, 1ST FLOOR  60616     5
1953

   pop_2010 change                                address_y  zip_y
0     56149     6%              2765 WEST SAINT MARY STREET  60647
1     55805     3%                 WM WASTE MANAGEMENT 1500  60622
2     53039    31%                      17 EAST 38TH STREET  60653
3     54589     5%  31ST ST HARBOR BUILDING LAKEFRONT TRAIL  60653
```

In [6]:
```
print(wards_census.shape)
```

```
(50, 9)
```

In [7]:
```
#Ex 1

taxi_owners = pd.read_pickle("taxi_owners.p")

taxi_vehicles = pd.read_pickle("taxi_vehicles.p")

taxi_own_veh = taxi_owners.merge(taxi_vehicles, on = "vid")



print(taxi_own_veh["fuel_type"].value_counts())
print()

print("The highest is : ", taxi_own_veh["fuel_type"].max())
```

```
HYBRID                   2792
GASOLINE                  611
FLEX FUEL                  89
COMPRESSED NATURAL GAS     27
Name: fuel_type, dtype: int64

The highest is :  HYBRID
```

In [8]:

```
# Ex 2

wards_altered = pd.read_csv("Wards_Offices_Altered.csv")
wards_census_altered = wards_altered.merge(census, on = "ward")
print(wards_census_altered)

print("there are 46 rows")
```

```
    ward                   alderman                          address_x  z
ip_x  \
0      1         Proco "Joe" Moreno          2058 NORTH WESTERN AVENUE  6
0647
1      2              Brian Hopkins         1400 NORTH  ASHLAND AVENUE  6
0622
2      3                 Pat Dowell            5046 SOUTH STATE STREET  6
0609
3      4           William D. Burns    435 EAST 35TH STREET, 1ST FLOOR  6
0616
4      5         Leslie A. Hairston              2325 EAST 71ST STREET  6
0649
5      6         Roderick T. Sawyer   8001 S. MARTIN LUTHER KING DRIVE  6
0619
6      7        Gregory I. Mitchell              2249 EAST 95TH STREET  6
0617
7      8         Michelle A. Harris    8539 SOUTH COTTAGE GROVE AVENUE  6
0619
8      9           Anthony A. Beale                34 EAST 112TH PLACE  6
0628
9     10      Susan Sadlowski Garza           10500 SOUTH EWING AVENUE  6
0617
10    11     Patrick Daley Thompson          3659 SOUTH HALSTED STREET  6
0609
11    12            George Cardenas           3476 SOUTH ARCHER AVENUE  6
0608
12    13                Marty Quinn            6500 SOUTH PULASKI ROAD  6
0629
13    14            Edward M. Burke              2650 WEST 51ST STREET  6
0632
14    15           Raymond A. Lopez              1650 WEST 63RD STREET  6
0636
15    16            Toni L. Foulkes              3045 WEST 63RD STREET  6
0629
16    17             David H. Moore          7313 SOUTH ASHLAND AVENUE  6
0636
17    18          Derrick G. Curtis            8359 SOUTH PULASKI ROAD  6
0652
18    19          Matthew J. O'Shea         10400 SOUTH WESTERN AVENUE  6
0643
19    20          Willie B. Cochran    6357 SOUTH COTTAGE GROVE AVENUE  6
0637
20    21    Howard B. Brookins, Jr.  9011 SOUTH ASHLAND AVENUE, UNIT B  6
0620
21    22              Ricardo Munoz        2500 SOUTH ST. LOUIS AVENUE  6
0623
22    23        Michael R. Zalewski           6247 SOUTH ARCHER AVENUE  6
0638
```

```
23    24         Michael Scott, Jr.           1158 SOUTH KEELER AVENUE  6
0624
24    25       Daniel "Danny" Solis      1800 SOUTH BLUE ISLAND AVENUE  6
0608
25    26          Roberto Maldonado          2511 WEST DIVISION STREET  6
0622
26    27        Walter Burnett, Jr.             4 NORTH WESTERN AVENUE  6
0612
27    28             Jason C. Ervin              2602 WEST 16TH STREET  6
0612
28    29           Chris Taliaferro             6272 WEST NORTH AVENUE  6
0639
29    30          Ariel E. Reyboras        3559 NORTH MILWAUKEE AVENUE  6
0641
30    31  Milagros "Milly" Santiago            2521 NORTH PULASKI ROAD  6
0639
31    32           Scott Waguespack         2657 NORTH CLYBOURN AVENUE  6
0614
32    33               Deborah Mell         3001 WEST IRVING PARK ROAD  6
0618
33    34           Carrie M. Austin              507 WEST 111TH STREET  6
0628
34    35        Carlos Ramirez-Rosa           2710 NORTH SAWYER AVENUE  6
0647
35    36           Gilbert Villegas                 6934 WEST DIVERSEY  6
0607
36    37              Emma M. Mitts           4924 WEST CHICAGO AVENUE  6
0651
37    38           Nicholas Sposato          3821  NORTH HARLEM AVENUE  6
0634
38    39           Margaret Laurino          4404 WEST LAWRENCE AVENUE  6
0630
39    40        Patrick J. O'Connor          5850 NORTH LINCOLN AVENUE  6
0659
40    41      Anthony V. Napolitano           7442 NORTH HARLEM AVENUE  6
0631
41    42             Brendan Reilly   325 WEST HURON STREET, SUITE 510  6
0654
42    43             Michelle Smith          2523 NORTH HALSTED STREET  6
0614
43    44                 Tom Tunney        3223 NORTH SHEFFIELD AVENUE  6
0657
44    45              John S. Arena        4754 NORTH MILWAUKEE AVENUE  6
0630
45    46            James Cappleman         4544 NORTH BROADWAY AVENUE  6
0640

    pop_2000  pop_2010 change  \
0      52951     56149     6%
1      54361     55805     3%
2      40385     53039    31%
3      51953     54589     5%
4      55302     51455    -7%
5      54989     52341    -5%
6      54593     51581    -6%
7      54039     51687    -4%
```

8	52008	51519	-1%
9	56613	51535	-9%
10	64228	51497	-20%
11	68922	52235	-24%
12	64382	53722	-17%
13	80143	54031	-33%
14	56057	51501	-8%
15	50205	51954	3%
16	49264	51846	5%
17	55043	52992	-4%
18	54546	51525	-6%
19	51854	52372	1%
20	51751	51632	0%
21	59734	53515	-10%
22	63691	53728	-16%
23	50879	54909	8%
24	55954	54539	-3%
25	56841	53516	-6%
26	61287	52939	-14%
27	49423	55199	12%
28	61949	55267	-11%
29	72698	55560	-24%
30	65045	53724	-17%
31	57204	55184	-4%
32	63695	55598	-13%
33	49922	51599	3%
34	57588	55281	-4%
35	63376	54766	-14%
36	56120	51538	-8%
37	66011	56001	-15%
38	64291	55882	-13%
39	58652	55319	-6%
40	56127	55991	0%
41	68102	55870	-18%
42	57668	56170	-3%
43	58758	56058	-5%
44	60653	55967	-8%
45	56587	53784	-5%

	address_y	zip_y
0	2765 WEST SAINT MARY STREET	60647
1	WM WASTE MANAGEMENT 1500	60622
2	17 EAST 38TH STREET	60653
3	31ST ST HARBOR BUILDING LAKEFRONT TRAIL	60653
4	JACKSON PARK LAGOON SOUTH CORNELL DRIVE	60637
5	150 WEST 74TH STREET	60636
6	8549 SOUTH OGLESBY AVENUE	60617
7	1346-1352 EAST 75TH STREET	60649
8	11039-11059 SOUTH WENTWORTH AVENUE	60628
9	10534 SOUTH AVENUE F	46394
10	943-947 WEST 14TH PLACE	60607
11	CP 46 STEVENSON EXPRESSWAY	60632
12	SOUTH RAMP SOUTH LARAMIE AVENUE	60638
13	4540 WEST 51ST STREET	60632
14	CHICAGO FIRE DEPARTMENT ENGINE COMPANY 123 2215	60632
15	6036 SOUTH WOOD STREET	60636

```
16                        7216 SOUTH WINCHESTER AVENUE  60636
17                           3286 WEST COLUMBUS AVENUE  60652
18                          9999 SOUTH FRANCISCO AVENUE  60805
19                      DAN RYAN EXPRESSWAY PARK MANOR  60621
20                      8852-8854 SOUTH EMERALD AVENUE  60620
21                                4233 WEST 36TH STREET  60632
22  CHICAGO MIDWAY INTERNATIONAL AIRPORT WEST 62ND...  60629
23                          1635 SOUTH CHRISTIANA AVENUE  60623
24                        1632-1746 SOUTH MILLER STREET  60608
25              LITTLE CUBS FIELD COMFORT STATION 1400  60622
26                          2151-2153 WEST CHICAGO AVENUE  60651
27                            RML SPECIALTY HOSPITAL 3435  60624
28                            1241 NORTH RIDGELAND AVENUE  60302
29                              5118 WEST FLETCHER STREET  60641
30                              2854 NORTH KEATING AVENUE  60641
31                            2901 NORTH WASHTENAW AVENUE  60618
32                      4041-4043 NORTH RICHMOND STREET  60625
33                      11544-11546 SOUTH PEORIA STREET  60827
34                               3634 WEST BELMONT AVENUE  60618
35                          2918 NORTH RUTHERFORD AVENUE  60634
36                            4738-4748 WEST RICE STREET  60651
37                      7307-7331 WEST IRVING PARK ROAD  60706
38                    QUEEN OF ALL SAINTS BASILICA 6280  60646
39                           5536 NORTH ARTESIAN AVENUE  60645
40                            1652 SOUTH CLIFTON AVENUE  60068
41                            410-420 WEST GRAND AVENUE  60654
42                                LINCOLN PARK ZOO 2001  60614
43                            507-513 WEST ALDINE AVENUE  60657
44         CONGREGATIONAL CHURCH OF JEFFERSON PARK 5320  60630
45                    UPTOWN BROADWAY BUILDING 4743-4763  60640
there are 46 rows
```

In [9]:

```
licenses = pd.read_csv('Business_Licenses.csv')
print(licenses.head())
print(licenses.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)
```

In [10]:
```
ward_licenses = wards.merge(licenses, on = "ward", suffixes = ("_ward", "_
print(ward_licenses.head())
```

```
   ward            alderman              address_ward  zip_ward  account
\
0     1  Proco "Joe" Moreno  2058 NORTH WESTERN AVENUE     60647    12024
1     1  Proco "Joe" Moreno  2058 NORTH WESTERN AVENUE     60647    14446
2     1  Proco "Joe" Moreno  2058 NORTH WESTERN AVENUE     60647    14624
3     1  Proco "Joe" Moreno  2058 NORTH WESTERN AVENUE     60647    14987
4     1  Proco "Joe" Moreno  2058 NORTH WESTERN AVENUE     60647    15642

   aid              business              address_lic  zip_lic
0  NaN   DIGILOG ELECTRONICS       1038 N ASHLAND AVE  60622.0
1  743      EMPTY BOTTLE INC   1035 N WESTERN AVE 1ST  60622.0
2  775  LITTLE MEL'S HOT DOG    2205 N CALIFORNIA AVE  60647.0
3  NaN    MR. BROWN'S LOUNGE   2301 W CHICAGO AVE 1ST  60622.0
4  814          Beat Kitchen  2000-2100 W DIVISION ST  60622.0
```

In [11]:
```
print(wards.shape)
print(ward_licenses.shape)
```

```
(50, 4)
(10000, 9)
```

In [ ]:
```
```

In [12]:

```
#Ex 3
licenses = pd.read_pickle("licenses.p")
biz_owners = pd.read_pickle("business_owners.p")


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

counted_df = licenses_owners.groupby("title").agg({"account" : "count"})
print(counted_df.head())
print("-----------------------------------------------")

sorted_df = counted_df.sort_values(["account"], ascending = False)
print(sorted_df.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
-----------------------------------------------
                   account
title
ASST. SECRETARY        111
BENEFICIARY              4
CEO                    110
DIRECTOR               146
EXECUTIVE DIRECTOR      10
-----------------------------------------------
                 account
title
PRESIDENT           6259
SECRETARY           5205
SOLE PROPRIETOR     1658
OTHER               1200
VICE PRESIDENT       970
```

In [13]:
```
grants = pd.read_csv("Smalllajdf;ajdsf; . csv ")
print(grants.head())
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
Input In [13], in <cell line: 1>()
----> 1 grants = pd.read_csv("Smalllajdf;ajdsf; . csv ")
      2 print(grants.head())

File ~\anaconda3\lib\site-packages\pandas\util\_decorators.py:311, in deprecate_nonkeyword_arguments.<locals>.decorate.<locals>.wrapper(*args, **kwargs)
    305 if len(args) > num_allow_args:
    306     warnings.warn(
    307         msg.format(arguments=arguments),
    308         FutureWarning,
    309         stacklevel=stacklevel,
    310     )
--> 311 return func(*args, **kwargs)

File ~\anaconda3\lib\site-packages\pandas\io\parsers\readers.py:680, in read_csv(filepath_or_buffer, sep, delimiter, header, names, index_col, usecols, squeeze, prefix, mangle_dupe_cols, dtype, engine, converters, true_values, false_values, skipinitialspace, skiprows, skipfooter, nrows, na_values, keep_default_na, na_filter, verbose, skip_blank_lines, parse_dates, infer_datetime_format, keep_date_col, date_parser, dayfirst, cache_dates, iterator, chunksize, compression, thousands, decimal, lineterminator, quotechar, quoting, doublequote, escapechar, comment, encoding, encoding_errors, dialect, error_bad_lines, warn_bad_lines, on_bad_lines, delim_whitespace, low_memory, memory_map, float_precision, storage_options)
    665 kwds_defaults = _refine_defaults_read(
    666     dialect,
    667     delimiter,
   (...)
    676     defaults={"delimiter": ","},
    677 )
    678 kwds.update(kwds_defaults)
--> 680 return _read(filepath_or_buffer, kwds)

File ~\anaconda3\lib\site-packages\pandas\io\parsers\readers.py:575, in _read(filepath_or_buffer, kwds)
    572 _validate_names(kwds.get("names", None))
    574 # Create the parser.
--> 575 parser = TextFileReader(filepath_or_buffer, **kwds)
    577 if chunksize or iterator:
    578     return parser

File ~\anaconda3\lib\site-packages\pandas\io\parsers\readers.py:933, in TextFileReader.__init__(self, f, engine, **kwds)
    930     self.options["has_index_names"] = kwds["has_index_names"]
    932 self.handles: IOHandles | None = None
--> 933 self._engine = self._make_engine(f, self.engine)

File ~\anaconda3\lib\site-packages\pandas\io\parsers\readers.py:1217, in
```

```
TextFileReader._make_engine(self, f, engine)
   1213     mode = "rb"
   1214 # error: No overload variant of "get_handle" matches argument typ
es
   1215 # "Union[str, PathLike[str], ReadCsvBuffer[bytes], ReadCsvBuffer
[str]]"
   1216 # , "str", "bool", "Any", "Any", "Any", "Any", "Any"
-> 1217 self.handles = get_handle(  # type: ignore[call-overload]
   1218     f,
   1219     mode,
   1220     encoding=self.options.get("encoding", None),
   1221     compression=self.options.get("compression", None),
   1222     memory_map=self.options.get("memory_map", False),
   1223     is_text=is_text,
   1224     errors=self.options.get("encoding_errors", "strict"),
   1225     storage_options=self.options.get("storage_options", None),
   1226 )
   1227 assert self.handles is not None
   1228 f = self.handles.handle

File ~\anaconda3\lib\site-packages\pandas\io\common.py:789, in get_handle
(path_or_buf, mode, encoding, compression, memory_map, is_text, errors, s
torage_options)
    784 elif isinstance(handle, str):
    785     # Check whether the filename is to be opened in binary mode.
    786     # Binary mode does not support 'encoding' and 'newline'.
    787     if ioargs.encoding and "b" not in ioargs.mode:
    788         # Encoding
--> 789         handle = open(
    790             handle,
    791             ioargs.mode,
    792             encoding=ioargs.encoding,
    793             errors=errors,
    794             newline="",
    795         )
    796     else:
    797         # Binary mode
    798         handle = open(handle, ioargs.mode)

FileNotFoundError: [Errno 2] No such file or directory: 'Smalllajdf;ajds
f; . csv '
```

In [14]:
```
grants_license = grants.merge(licenses, on = 'zip')
print(grants_licenses.loc[grants_licenses["business"]== "REGGIE'S BAR & GR
                          ["grant", "company", "account", "ward", "busines
```

```
---------------------------------------------------------------------------
--
NameError                                 Traceback (most recent call las
t)
Input In [14], in <cell line: 1>()
----> 1 grants_license = grants.merge(licenses, on = 'zip')
      2 print(grants_licenses.loc[grants_licenses["business"]== "REGGIE'S
BAR & GRILL",
      3                           ["grant", "company", "account", "ward",
"business"]])

NameError: name 'grants' is not defined
```

In [15]:
```
grants.merge(licenses, on = ["address", "zip"])
```

```
---------------------------------------------------------------------------
--
NameError                                 Traceback (most recent call las
t)
Input In [15], in <cell line: 1>()
----> 1 grants.merge(licenses, on = ["address", "zip"])

NameError: name 'grants' is not defined
```

In [16]:
```
grants_license_ward = grants.merge(license, on = ["address", "zip"]) \
.merge(wards, on = "ward", suffixes = ("bus", "_wards"))
```

```
  Input In [16]
    grants_license_ward = grants.merge(license, on = ["address", "zip"])
\

^
SyntaxError: unexpected character after line continuation character
```

In [17]:
```
grant_license_ward. groupby('ward').agg('sum').plot(kind = 'bar', y = 'gra
plt.show()
```

```
---------------------------------------------------------------------------
--
NameError                                 Traceback (most recent call las
t)
Input In [17], in <cell line: 1>()
----> 1 grant_license_ward. groupby('ward').agg('sum').plot(kind = 'bar',
y = 'grant')
      2 plt.show()

NameError: name 'grant_license_ward' is not defined
```

In [ ]:
```
```

In [18]:
```
#Ex 4

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


ridership_cal_station = ridership.merge(cal, on = [ "year", "month", "day"

fil = ((ridership_cal_station [ "month" ] == 7)
      & (ridership_cal_station [ 'day_type'] == 'Weekday')
      & (ridership_cal_station [ "station_name"] == "Wilson"))

print(ridership_cal_station.loc[fil, 'rides'].sum())
```

```
140005
```

In [19]:
```
#Ex 5

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

licenses_zip_ward = licenses.merge(zip_demo, on = "zip").merge(wards, on =

income = licenses_zip_ward.groupby("alderman").agg(({"income": "median"}))
income
```

Out[19]:

alderman	income
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

In [20]:
```
movies = pd.read_csv("tmdb_movies.csv")
print(movies.head())
print(movies.shape)
```

```
      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)
```

In [21]:
```
taglines = pd.read_csv('tmdb_movies.csv')
print(taglines.head())
print(taglines.shape)
```

```
      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)
```

In [22]:
```
movies_taglines = movies.merge(taglines, on = "id", how = 'left')
print(movies_taglines.head())
```

```
      id               title_x  popularity_x release_date_x  \
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

                title_y  popularity_y release_date_y
0          Oliver Twist     20.415572      23/9/2005
1  Better Luck Tomorrow      3.877036      12/1/2002
2             Grown Ups     38.864027      24/6/2010
3               Infamous      3.680896     16/11/2006
4       Alpha and Omega     12.300789      17/9/2010
```

In [23]:
```
print(movies_taglines.shape)
```

```
(4803, 7)
```

In [24]:
```
#Ex 6

movies = pd.read_pickle("movies.p")
financials = pd.read_pickle("financials.p")
movies_financials = movies.merge(financials, on = "id", how = "left")
movies_financials["budget"].isnull().sum()
```

Out[24]: 1574

In [25]:
```
#Ex 7


toy_story = pd.read_csv("toy_story.csv")
taglines = pd.read_pickle("taglines.p")

toy_story_tag = toy_story.merge(taglines, on = "id", how="left")

#toy_story_tag = toy_story.merge(taglines, on = "id")

toy_story_tag
```

Out[25]:

	id	title	popularity	release_date	tagline
0	10193	Toy Story 3	59.995	16/6/2010	No toy gets left behind.
1	863	Toy Story 2	73.575	30/10/1999	The toys are back!
2	862	Toy Story	73.640	30/10/1995	NaN

In [26]:
```
movie_to_genres = pd.read_csv('tdmb_movie_to_genres.csv')
tv_genre = movie_to_genres[movie_to_genres['genre'] == 'TV Movie']

print(tv_genre)
```

```
      movie_id     genre
4998     10947  TV Movie
5994     13187  TV Movie
7443     22488  TV Movie
10061    78814  TV Movie
10790   153397  TV Movie
10835   158150  TV Movie
11096   205321  TV Movie
11282   231617  TV Movie
```

In [27]:
```
tv_movies = movies.merge(tv_genre, how = 'right', left_on = 'id', right_on

print(tv_movies.head())
```

```
       id                      title  popularity release_date  movie_id
\
0   10947        High School Musical   16.536374   2006-01-20     10947
1   13187  A Charlie Brown Christmas    8.701183   1965-12-09     13187
2   22488         Love's Abiding Joy    1.128559   2006-10-06     22488
3   78814       We Have Your Husband    0.102003   2011-11-12     78814
4  153397                    Restless    0.812776   2012-12-07    153397

      genre
0  TV Movie
1  TV Movie
2  TV Movie
3  TV Movie
4  TV Movie
```

In [28]:
```
m = movie_to_genres['genre'] == "Family"
family = movie_to_genres[m].head(3)
family

m = movie_to_genres['genre'] == "Comedy"
comedy = movie_to_genres[m].head(3)
comedy
```

Out[28]:

	movie_id	genre
1	5	Comedy
7	13	Comedy
35	35	Comedy

In [29]:
```
family_comedy = family.merge(comedy, on = "movie_id", how = "outer", suffi
print(family_comedy)
```

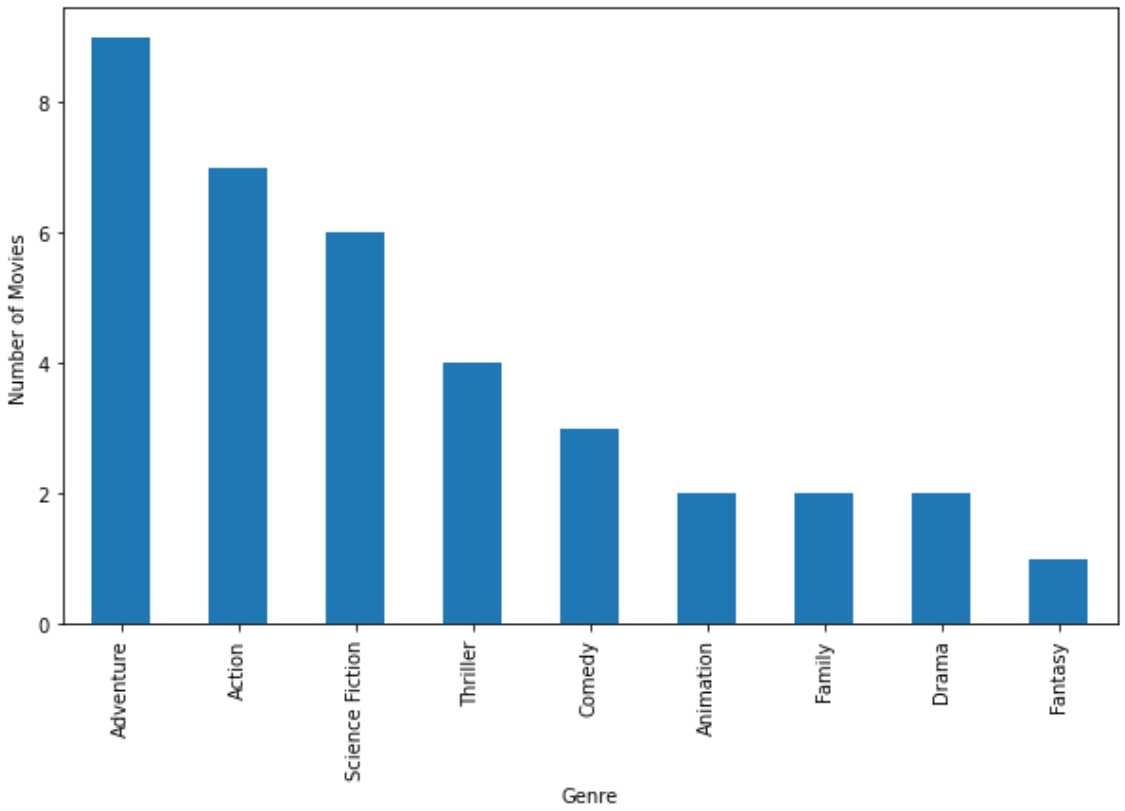
```
   movie_id genre_fam genre_com
0        12    Family       NaN
1        35    Family    Comedy
2       105    Family       NaN
3         5       NaN    Comedy
4        13       NaN    Comedy
```

In [30]:
```
#Ex 8
import pandas as pd

# Load data
movies = pd.read_pickle('movies.p')

# Subset scifi_movies and action_movies
scifi_movies = movie_to_genres[movie_to_genres['genre'] == 'Science Fictio
action_movies = movie_to_genres[movie_to_genres['genre'] == 'Action']

# Merge action_movies and scifi_movies with a right join and add suffixes
action_scifi = pd.merge(action_movies, scifi_movies, on='movie_id', how='r

# Subset rows where genre_act column is null (only science fiction)
scifi_only = action_scifi[action_scifi['genre_act'].isnull()]

# Merge movies and scifi_only with an inner join
result = pd.merge(movies, scifi_only, left_on='id', right_on='movie_id', h

# Display the result
print(result[['id', 'title']])
```

```
         id                          title
0     18841  The Lost Skeleton of Cadavra
1     26672     The Thief and the Cobbler
2     15301      Twilight Zone: The Movie
3      8452                   The 6th Day
4      1649    Bill & Ted's Bogus Journey
..      ...                           ...
253  245703              Midnight Special
254    3509              A Scanner Darkly
255   42188               Never Let Me Go
256   18045                 The Dark Hours
257   11058                       Godsend

[258 rows x 2 columns]
```

In [31]:

```
#Ex 9

import pandas as pd
import matplotlib.pyplot as plt

# Load data from CSV files
pop_movies = pd.read_csv('pop_movies.csv')
movie_to_genres = pd.read_csv('tdmb_movie_to_genres.csv')

# Merge using right join
genres_movies = pd.merge(movie_to_genres, pop_movies, left_on='movie_id',

# Group by genre and count the number of movies
genre_counts = genres_movies['genre'].value_counts().head(10)

# Plot the bar chart
plt.figure(figsize=(10, 6))
genre_counts.plot(kind='bar')
plt.xlabel('Genre')
plt.ylabel('Number of Movies')

plt.show()
```



In [32]:

```
sequels = pd.read_pickle("sequels.p")
original_sequels = sequels.merge(sequels, left_on = 'sequel', right_on = '
                                 suffixes= ('_org','_seq'))
print(original_sequels.head())

original_sequels = sequels.merge(sequels, left_on="sequel", right_on = "id
                                 how = "left", suffixes = ("_org", "_seq"))

print(original_sequels.head())
```

```
   id_org                                          title_org  sequel_org
\
0     862                                          Toy Story         863
1     863                                        Toy Story 2       10193
2     675         Harry Potter and the Order of the Phoenix         767
3     121             The Lord of the Rings: The Two Towers         122
4     120  The Lord of the Rings: The Fellowship of the Ring         121

   id_seq                                       title_seq  sequel_seq
0     863                                     Toy Story 2       10193
1   10193                                     Toy Story 3        <NA>
2     767        Harry Potter and the Half-Blood Prince        <NA>
3     122  The Lord of the Rings: The Return of the King        <NA>
4     121          The Lord of the Rings: The Two Towers         122
   id_org     title_org  sequel_org  id_seq    title_seq  sequel_seq
0   19995        Avatar        <NA>    <NA>          NaN        <NA>
1     862     Toy Story         863     863  Toy Story 2       10193
2     863   Toy Story 2       10193   10193  Toy Story 3        <NA>
3     597       Titanic        <NA>    <NA>          NaN        <NA>
4   24428  The Avengers        <NA>    <NA>          NaN        <NA>
```

In [33]:
```
#Ex 10

import pandas as pd

# Load the crews table
crews = pd.read_pickle('crews.p')

crews_self_merged = pd.merge(crews, crews, on='id', how='inner', suffixes=

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

direct_crews = crews_self_merged[boolean_filter]

print(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
```

In [52]:
```
import pandas as pd
pd.concat([inv_jan, inv_feb, inv_mar], ignor_index =True, keys=['jan', 'fe
```

```
---------------------------------------------------------------------
--
NameError                                 Traceback (most recent call las
t)
Input In [52], in <cell line: 2>()
      1 import pandas as pd
----> 2 pd.concat([inv_jan, inv_feb, inv_mar], ignor_index =True, keys=['
jan', 'feb', 'mar'])

NameError: name 'inv_jan' is not defined
```

In [54]:
```
#Ex11


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

print(tracks_master)

print()
print()
print(tracks_ride)
print()
print(tracks_st)


pd.concat([tracks_master, tracks_ride, tracks_st], sort = True, keys = ['n
```

```
    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  1874    Fight Fire With Fire  154     1    3     0.99
1  1875      Ride The Lightning  154     1    3     0.99
2  1876  For Whom The Bell Tolls  154     1    3     0.99
3  1877           Fade To Black  154     1    3     0.99
4  1878       Trapped Under Ice  154     1    3     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
```

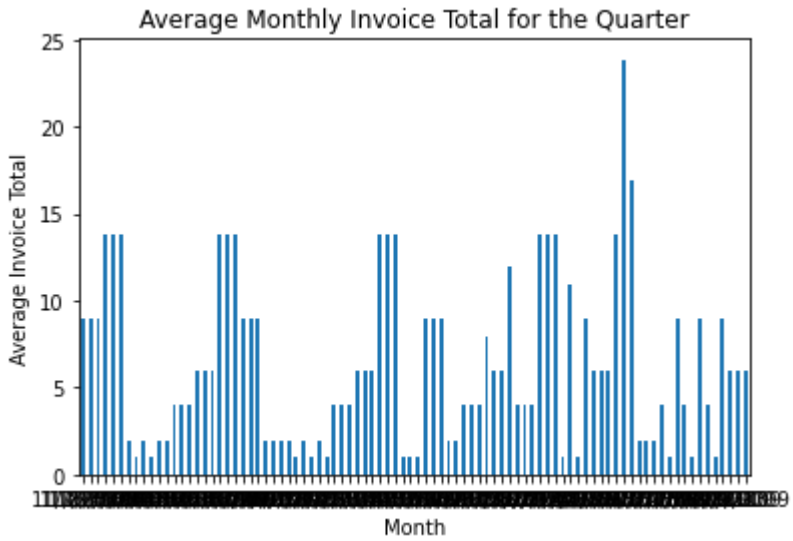
Out[54]:

		aid	composer	gid	mtid	name	tid	u_price
name	**0**	152	J.Hetfield/L.Ulrich	3	1	Battery	1853	0.99
	1	152	K.Hammett	3	1	Master Of Puppets	1854	0.99
	2	152	J.Hetfield/L.Ulrich	3	1	Disposable Heroes	1857	0.99
aid	**0**	154	NaN	3	1	Fight Fire With Fire	1874	0.99
	1	154	NaN	3	1	Ride The Lightning	1875	0.99
	2	154	NaN	3	1	For Whom The Bell Tolls	1876	0.99
	3	154	NaN	3	1	Fade To Black	1877	0.99
	4	154	NaN	3	1	Trapped Under Ice	1878	0.99
mtid	**0**	155	NaN	3	1	Frantic	1882	0.99

	aid	composer	gid	mtid	name	tid	u_price
1	155	NaN	3	1	St. Anger	1883	0.99
2	155	NaN	3	1	Some Kind Of Monster	1884	0.99

In [68]:

```
#Ex 12

import pandas as pd
import matplotlib.pyplot as plt

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=['7Jul', '8

average_totals = avg_inv_by_month.groupby('invoice_date')['total'].agg('me

average_totals.plot(kind='bar')
plt.xlabel('Month')
plt.ylabel('Average Invoice Total')
plt.title('Average Monthly Invoice Total for the Quarter')
plt.xticks(rotation=0)
plt.show()
```



In [56]:
```
import pandas as pd
pd.merge_ordered(appl, mcd, on = 'date', suffixes = ('_appl', '_mcd'))
pd.merge_ordered(appl, mcd, on = 'date', suffixes = ('_appl', '_mcd'), fil
```

```
---------------------------------------------------------------------------
--
NameError                                 Traceback (most recent call las
t)
Input In [56], in <cell line: 2>()
      1 import pandas as pd
----> 2 pd.merge_ordered(appl, mcd, on = 'date', suffixes = ('_appl', '_m
cd'))
      3 pd.merge_ordered(appl, mcd, on = 'date', suffixes = ('_appl', '_m
cd'), fill_method = 'ffill')

NameError: name 'appl' is not defined
```

In [69]:
```
#Ex 13

import pandas as pd

sp500 = pd.read_csv("S&P500.csv")
gdp = pd.read_csv("GDP.csv")

gdp_sp500 = pd.merge_ordered(gdp, sp500, left_on='year', right_on='date', l

print(gdp_sp500[gdp_sp500['year'] == 2018])

gdp_sp500 = pd.merge_ordered(gdp, sp500, left_on='year', right_on='date', l

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

correlation_matrix = gdp_returns.corr()
print(correlation_matrix)
```

```
   Unnamed: 0 country code  year           gdp  date  returns
9          39          USA  2018  2.050000e+13   NaN      NaN
                gdp   returns
gdp        1.000000  0.220321
returns    0.220321  1.000000
```

In [64]:

```
#Ex 14

import pandas as pd
import matplotlib.pyplot as plt

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

inflation_unemploy = pd.merge_ordered(inflation, unemployment, on='date'
print(inflation_unemploy)

plt.scatter(inflation_unemploy['unemployment_rate'], inflation_unemploy['c
plt.xlabel('Unemployment Rate')
plt.ylabel('CPI (Inflation)')
plt.title('Scatter Plot of Unemployment Rate vs. CPI (Inflation)')
plt.show()
```

```
       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
5  1/6/2014  237.231  CUSR0000SA0  SEASONALLY ADJUSTED INDEX
6  1/6/2015  237.684  CUSR0000SA0  SEASONALLY ADJUSTED INDEX
7  1/6/2016  240.167  CUSR0000SA0  SEASONALLY ADJUSTED INDEX
8  1/6/2017  244.182  CUSR0000SA0  SEASONALLY ADJUSTED INDEX
9  1/6/2018  251.134  CUSR0000SA0  SEASONALLY ADJUSTED INDEX

   unemployment_rate
0                6.7
1                5.6
2                5.0
3                4.7
4                4.1
5                6.1
6                5.3
7                4.9
8                4.3
9                4.0
```


Scatter Plot of Unemployment Rate vs. CPI (Inflation)
CPI (Inflation)
250.0
247.5
245.0
242.5
240.0
237.5

In [ ]: