

In [15]:

```
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
import pandas as pd
```

In [16]:

```
df = pd.read_csv('https://raw.githubusercontent.com/justmarkham/DAT8/master/data/drinks.csv')
df.head()
```

Out[16]:

	country	beer_servings	spirit_servings	wine_servings	total_litres_of_pure_alcohol
0	Afghanistan	0	0	0	0.0
1	Albania	89	132	54	4.9
2	Algeria	25	0	14	0.7
3	Andorra	245	138	312	12.4
4	Angola	217	57	45	5.9

In [17]:

```
#Which continent drinks more beer on average?
```

In [18]:

```
df.groupby('continent').beer_servings.mean()
```

Out[18]:

```
continent
AF      61.471698
AS      37.045455
EU     193.777778
OC      89.687500
SA     175.083333
Name: beer_servings, dtype: float64
```

In [19]:

```
#For each continent print the statistics for wine consumption.
```

In [20]:

```
df.groupby('continent').wine_servings.describe()
```

Out[20]:

	count	mean	std	min	25%	50%	75%	max
continent								
AF	53.0	16.264151	38.846419	0.0	1.0	2.0	13.00	233.0
AS	44.0	9.068182	21.667034	0.0	0.0	1.0	8.00	123.0
EU	45.0	142.222222	97.421738	0.0	59.0	128.0	195.00	370.0
OC	16.0	35.625000	64.555790	0.0	1.0	8.5	23.25	212.0
SA	12.0	62.416667	88.620189	1.0	3.0	12.0	98.50	221.0

In [21]:

```
url = "https://raw.githubusercontent.com/guipsamora/pandas_exercises/master/04_Apply/US_Crime_Rates/US_Crime_Rates_1960_2014.csv"
crime = pd.read_csv(url)
crime.head()
```

Out[21]:

	Year	Population	Total	Violent	Property	Murder	Forcible_Rape	Robbery
0	1960	179323175	3384200	288460	3095700	9110	17190	107840
1	1961	182992000	3488000	289390	3198600	8740	17220	106670
2	1962	185771000	3752200	301510	3450700	8530	17550	110860
3	1963	188483000	4109500	316970	3792500	8640	17650	116470
4	1964	191141000	4564600	364220	4200400	9360	21420	130390

In [22]:

```
#Convert the type of the column Year to datetime64
```

In [23]:

```
crime.Year = pd.to_datetime(crime.Year, format='%Y')
crime.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 55 entries, 0 to 54
Data columns (total 12 columns):
Year                55 non-null datetime64[ns]
Population          55 non-null int64
Total               55 non-null int64
Violent             55 non-null int64
Property            55 non-null int64
Murder              55 non-null int64
Forcible_Rape       55 non-null int64
Robbery             55 non-null int64
Aggravated_assault  55 non-null int64
Burglary            55 non-null int64
Larceny_Theft       55 non-null int64
Vehicle_Theft       55 non-null int64
dtypes: datetime64[ns](1), int64(11)
memory usage: 5.2 KB
```

In [24]:

```
#Set the Year column as the index of the dataframe
```

In [25]:

```
crime = crime.set_index('Year', drop = True)
crime.head()
```

Out[25]:

	Population	Total	Violent	Property	Murder	Forcible_Rape	Robbery	Ag
Year								
1960-01-01	179323175	3384200	288460	3095700	9110	17190	107840	15
1961-01-01	182992000	3488000	289390	3198600	8740	17220	106670	15
1962-01-01	185771000	3752200	301510	3450700	8530	17550	110860	16
1963-01-01	188483000	4109500	316970	3792500	8640	17650	116470	17
1964-01-01	191141000	4564600	364220	4200400	9360	21420	130390	20

In [26]:

```
#Group the year by decades and sum the values
```

In [27]:

```
# Uses resample to sum each decade
crimes = crime.resample('10AS').sum()

# Uses resample to get the max value only for the "Population" column
population = crime['Population'].resample('10AS').max()

# Updating the "Population" column
crimes['Population'] = population

crimes
```

Out[27]:

	Population	Total	Violent	Property	Murder	Forcible_Rape
Year						
1960-01-01	201385000.0	49295900.0	4134930.0	45160900.0	106180.0	236720.0
1970-01-01	220099000.0	100991600.0	9607930.0	91383800.0	192230.0	554570.0
1980-01-01	248239000.0	131123369.0	14074328.0	117048900.0	206439.0	865639.0
1990-01-01	272690813.0	136582146.0	17527048.0	119053499.0	211664.0	998827.0
2000-01-01	307006550.0	115012044.0	13968056.0	100944369.0	163068.0	922499.0
2010-01-01	318857056.0	50167967.0	6072017.0	44095950.0	72867.0	421059.0
2020-01-01	NaN	NaN	NaN	NaN	NaN	NaN

In [28]:

```
crime.head()
```

Out[28]:

	Population	Total	Violent	Property	Murder	Forcible_Rape	Robbery	Aggr
Year								
1960-01-01	179323175	3384200	288460	3095700	9110	17190	107840	15436
1961-01-01	182992000	3488000	289390	3198600	8740	17220	106670	15676
1962-01-01	185771000	3752200	301510	3450700	8530	17550	110860	16457
1963-01-01	188483000	4109500	316970	3792500	8640	17650	116470	17427
1964-01-01	191141000	4564600	364220	4200400	9360	21420	130390	20309

In [31]:

```
#Return the first 3 rows of the DataFrame df.
```

```
df = crime
```

In [32]:

```
df.iloc[:3]
```

Out[32]:

	Population	Total	Violent	Property	Murder	Forcible_Rape	Robbery	Ag
Year								
1960-01-01	179323175	3384200	288460	3095700	9110	17190	107840	15436
1961-01-01	182992000	3488000	289390	3198600	8740	17220	106670	15676
1962-01-01	185771000	3752200	301510	3450700	8530	17550	110860	16457

In [33]:

```
#Select just the 'Murder' and 'Robbery' columns from the DataFrame df and print first 5 records
```

In [35]:

```
df.loc[:, ['Murder', 'Robbery']].head()
```

Out[35]:

	Murder	Robbery
Year		
1960-01-01	9110	107840
1961-01-01	8740	106670
1962-01-01	8530	110860
1963-01-01	8640	116470
1964-01-01	9360	130390

In [37]:

```
df[['Murder', 'Robbery']].head()
```

Out[37]:

	Murder	Robbery
Year		
1960-01-01	9110	107840
1961-01-01	8740	106670
1962-01-01	8530	110860
1963-01-01	8640	116470
1964-01-01	9360	130390

In [38]:

```
#Select the data in rows [3, 4, 8] and in columns ['Murder', 'Robbery']
```

In [39]:

```
df.loc[df.index[[3, 4, 8]], ['Murder', 'Robbery']]
```

Out[39]:

	Murder	Robbery
Year		
1963-01-01	8640	116470
1964-01-01	9360	130390
1968-01-01	13800	262840

In [45]:

```
#Select only the rows where the number of murder is greater than 24,000
```

In [46]:

```
df[df['Murder'] > 24000]
```

Out[46]:

	Population	Total	Violent	Property	Murder	Forcible_Rape	Robbery
Year							
1991-01-01	252177000	14872900	1911770	12961100	24700	106590	687730
1993-01-01	257908000	14144800	1926020	12218800	24530	106010	659870

In [47]:

```
#Select the rows the murder is between 20k and 24k (inclusive)
```

In [51]:

```
df[df['Murder'].between(20000, 24000)]
```

Out[51]:

	Population	Total	Violent	Property	Murder	Forcible_Rape	Robbery	A
Year								
1974-01-01	211392000	10253400	974720	9278700	20710	55400	442400	45
1975-01-01	213124000	11292400	1039710	10252700	20510	56090	470500	49
1979-01-01	220099000	12249500	1208030	11041500	21460	76390	480700	62
1980-01-01	225349264	13408300	1344520	12063700	23040	82990	565840	67
1981-01-01	229146000	13423800	1361820	12061900	22520	82500	592910	66
1982-01-01	231534000	12974400	1322390	11652000	21010	78770	553130	66
1986-01-01	240132887	13211869	1489169	11722700	20613	91459	542775	83
1987-01-01	242282918	13508700	1483999	12024700	20096	91110	517704	85
1988-01-01	245807000	13923100	1566220	12356900	20680	92490	542970	91
1989-01-01	248239000	14251400	1646040	12605400	21500	94500	578330	95
1990-01-01	248709873	14475600	1820130	12655500	23440	102560	639270	10
1992-01-01	255082000	14438200	1932270	12505900	23760	109060	672480	11
1994-01-01	260341000	13989500	1857670	12131900	23330	102220	618950	11
1995-01-01	262755000	13862700	1798790	12063900	21610	97470	580510	10

In [52]:

```
#Calculate the mean murder for each different year in df.
```


In [53]:

```
df.groupby('Year')['Murder'].mean()
```

Out[53]:

Year	
1960-01-01	9110
1961-01-01	8740
1962-01-01	8530
1963-01-01	8640
1964-01-01	9360
1965-01-01	9960
1966-01-01	11040
1967-01-01	12240
1968-01-01	13800
1969-01-01	14760
1970-01-01	16000
1971-01-01	17780
1972-01-01	18670
1973-01-01	19640
1974-01-01	20710
1975-01-01	20510
1976-01-01	18780
1977-01-01	19120
1978-01-01	19560
1979-01-01	21460
1980-01-01	23040
1981-01-01	22520
1982-01-01	21010
1983-01-01	19310
1984-01-01	18690
1985-01-01	18980
1986-01-01	20613
1987-01-01	20096
1988-01-01	20680
1989-01-01	21500
1990-01-01	23440
1991-01-01	24700
1992-01-01	23760
1993-01-01	24530
1994-01-01	23330
1995-01-01	21610
1996-01-01	19650
1997-01-01	18208
1998-01-01	16914
1999-01-01	15522
2000-01-01	15586
2001-01-01	16037
2002-01-01	16229
2003-01-01	16528
2004-01-01	16148
2005-01-01	16740
2006-01-01	17030
2007-01-01	16929
2008-01-01	16442
2009-01-01	15399
2010-01-01	14772
2011-01-01	14661
2012-01-01	14866
2013-01-01	14319
2014-01-01	14249

Name: Murder, dtype: int64

In [55]:

```
#Sort df first by the values in the 'Murder' in decending order,  
#then by the value in the 'Violent' column in ascending order.
```

In [58]:

```
df.sort_values(by=['Murder', 'Violent'], ascending=[False, True])
```

Out[58]:

	Population	Total	Violent	Property	Murder	Forcible_Rape	Robbery
Year							
1991-01-01	252177000	14872900	1911770	12961100	24700	106590	687730
1993-01-01	257908000	14144800	1926020	12218800	24530	106010	659870
1992-01-01	255082000	14438200	1932270	12505900	23760	109060	672480
1990-01-01	248709873	14475600	1820130	12655500	23440	102560	639270
1994-01-01	260341000	13989500	1857670	12131900	23330	102220	618950
1980-01-01	225349264	13408300	1344520	12063700	23040	82990	565840
1981-01-01	229146000	13423800	1361820	12061900	22520	82500	592910
1995-01-01	262755000	13862700	1798790	12063900	21610	97470	580510
1989-01-01	248239000	14251400	1646040	12605400	21500	94500	578330
1979-01-01	220099000	12249500	1208030	11041500	21460	76390	480700
1982-01-01	231534000	12974400	1322390	11652000	21010	78770	553130
1974-01-01	211392000	10253400	974720	9278700	20710	55400	442400
1988-01-01	245807000	13923100	1566220	12356900	20680	92490	542970
1986-01-01	240132887	13211869	1489169	11722700	20613	91459	542775
1975-01-01	213124000	11292400	1039710	10252700	20510	56090	470500
1987-01-01	242282918	13508700	1483999	12024700	20096	91110	517704
1996-01-01	265228572	13493863	1688540	11805300	19650	96250	535590
1973-01-01	209851000	8718100	875910	7842200	19640	51400	384220
1978-01-01	218059000	11209000	1085550	10123400	19560	67610	426930

	Population	Total	Violent	Property	Murder	Forcible_Rape	Robbery
Year							
1983-01-01	233981000	12108600	1258090	10850500	19310	78920	506570
1977-01-01	216332000	10984500	1029580	9955000	19120	63500	412610
1985-01-01	238740000	12431400	1328800	11102600	18980	88670	497870
1976-01-01	214659000	11349700	1004210	10345500	18780	57080	427810
1984-01-01	236158000	11881800	1273280	10608500	18690	84230	485010
1972-01-01	208230000	8248800	834900	7413900	18670	46850	376290
1997-01-01	267637000	13194571	1634770	11558175	18208	96153	498534
1971-01-01	206212000	8588200	816500	7771700	17780	42260	387700
2006-01-01	299398484	11401511	1418043	9983568	17030	92757	447403
2007-01-01	301621157	11251828	1408337	9843481	16929	90427	445125
1998-01-01	270296000	12475634	1531044	10944590	16914	93103	446625
2005-01-01	296507061	11565499	1390745	10174754	16740	94347	417438
2003-01-01	290690788	11826538	1383676	10442862	16528	93883	414235
2008-01-01	304374846	11160543	1392628	9767915	16442	90479	443574
2002-01-01	287973924	11878954	1423677	10455277	16229	95235	420806
2004-01-01	293656842	11679474	1360088	10319386	16148	95089	401470
2001-01-01	285317559	11876669	1439480	10437480	16037	90863	423557
1970-01-01	203235298	8098000	738820	7359200	16000	37990	349860
2000-01-01	281421906	11608072	1425486	10182586	15586	90178	408016
1999-01-01	272690813	11634378	1426044	10208334	15522	89411	409371

	Population	Total	Violent	Property	Murder	Forcible_Rape	Robbery
Year							
2009-01-01	307006550	10762956	1325896	9337060	15399	89241	408742
2012-01-01	313873685	10219059	1217067	9001992	14866	85141	355051
2010-01-01	309330219	10363873	1251248	9112625	14772	85593	369089
1969-01-01	201385000	7410900	661870	6749000	14760	37170	298850
2011-01-01	311587816	10258774	1206031	9052743	14661	84175	354772
2013-01-01	316497531	9850445	1199684	8650761	14319	82109	345095
2014-01-01	318857056	9475816	1197987	8277829	14249	84041	325802
1968-01-01	199399000	6720200	595010	6125200	13800	31670	262840
1967-01-01	197457000	5903400	499930	5403500	12240	27620	202910
1966-01-01	195576000	5223500	430180	4793300	11040	25820	157990
1965-01-01	193526000	4739400	387390	4352000	9960	23410	138690
1964-01-01	191141000	4564600	364220	4200400	9360	21420	130390
1960-01-01	179323175	3384200	288460	3095700	9110	17190	107840
1961-01-01	182992000	3488000	289390	3198600	8740	17220	106670
1963-01-01	188483000	4109500	316970	3792500	8640	17650	116470
1962-01-01	185771000	3752200	301510	3450700	8530	17550	110860

In [59]:

```
#read the following data set
#https://raw.githubusercontent.com/vincentarelbundock/Rdatasets/master/csv/datasets/mtcars.csv
```

In [60]:

```
df = pd.read_csv('https://raw.githubusercontent.com/vincentarelbundock/Rdatasets/master/csv/datasets/mtcars.csv')
```

In [61]:

```
df.head()
```

Out[61]:

	Unnamed: 0	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
1	Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
2	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
3	Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
4	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2

In [62]:

```
#For each cyl type and each number of gears, find the mean mileage.
```

In [65]:

```
df.pivot_table(index='cyl', columns='gear', values='mpg', aggfunc='mean')
```

Out[65]:

gear	3	4	5
cyl			
4	21.50	26.925	28.2
6	19.75	19.750	19.7
8	15.05	NaN	15.4

In []: