In [2]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set() #Code in MatplotLib but show in Seaborn , Default Seaborn them, color pall
#Seaborn is upgraded version of MatplotLib
```

BAR CHART

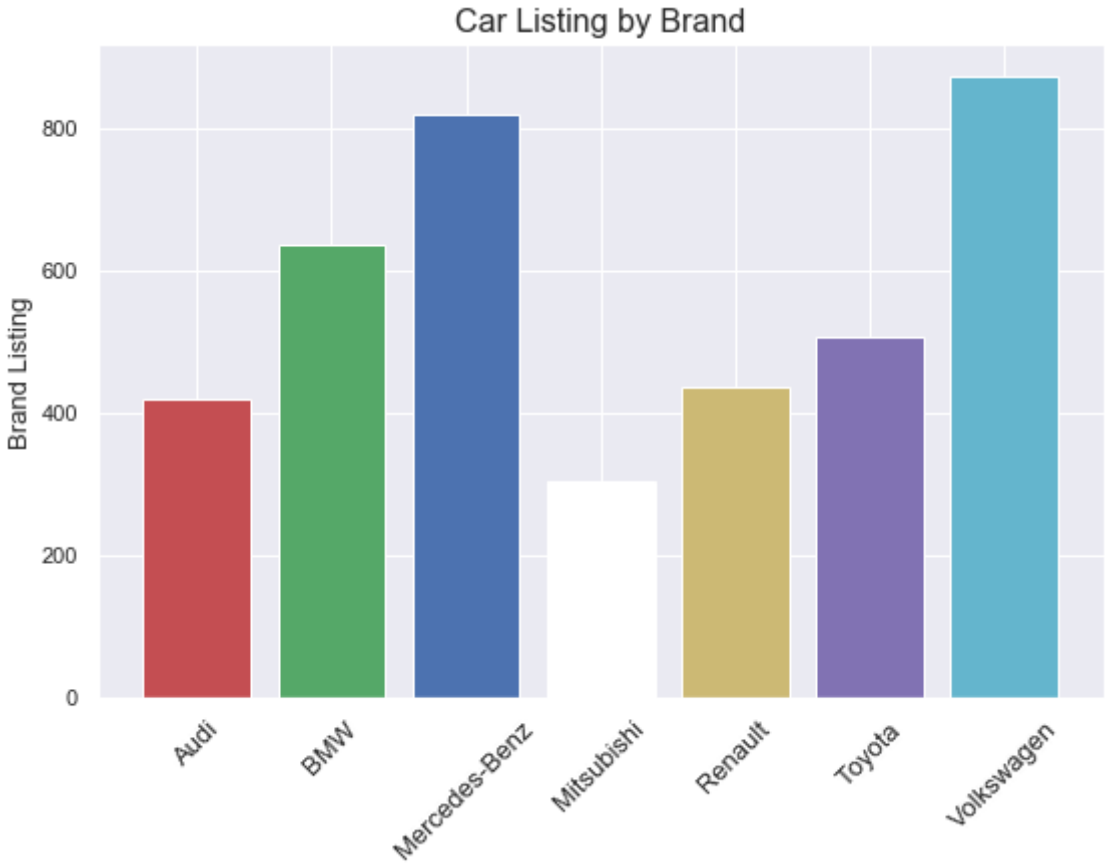
In [8]:
```
df_used_cars=pd.read_csv('C:/Users/abhinendr.sharma/Documents/Data Analyst_Udemy/Ude
```

In [9]:
```
df_used_cars
```

Out[9]:

	Brand	Cars Listings
0	Audi	419
1	BMW	636
2	Mercedes-Benz	820
3	Mitsubishi	306
4	Renault	438
5	Toyota	509
6	Volkswagen	875

In [15]:
```
plt.figure(figsize=(9,6)) #Size for Chart

plt.bar(x=df_used_cars['Brand']
        ,height = df_used_cars['Cars Listings']
        ,color = "rgbwymc")

plt.xticks(rotation=45, fontsize=13)   #X-Axis Labels Rotation

plt.title("Car Listing by Brand", fontsize=16)

plt.ylabel("Brand Listing", fontsize=13)

plt.show

plt.savefig("Used Car Bar.png")
```



PIE CHART

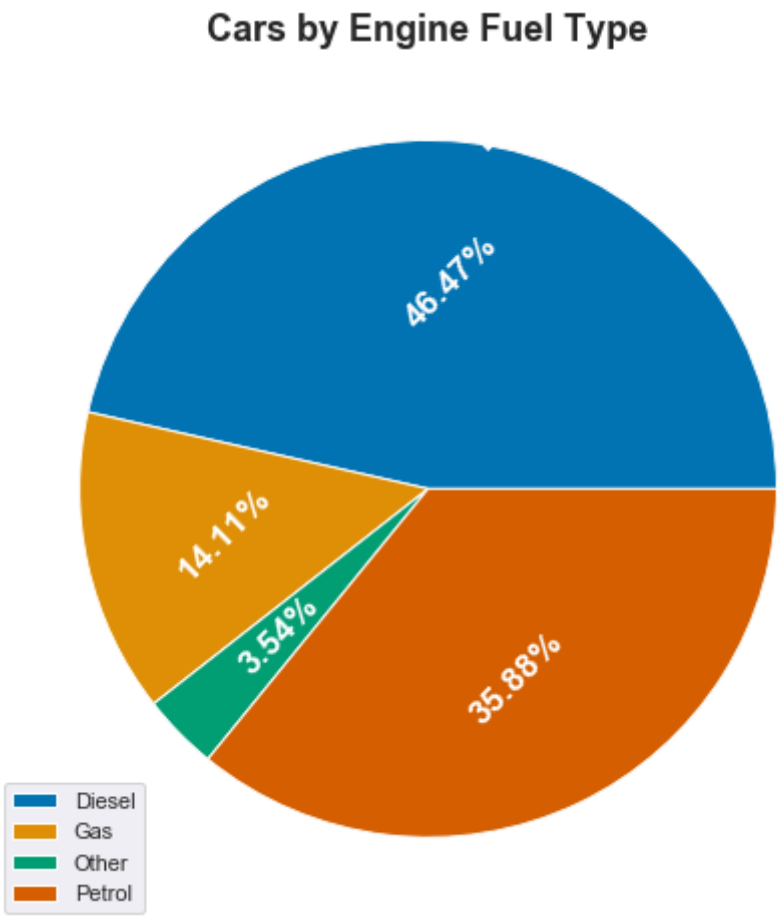
In [17]:
```
df_pie_chart = pd.read_csv('C:/Users/abhinendr.sharma/Documents/Data Analyst_Udemy/U
df_pie_chart
```

Out[17]:

	Engine Fuel Type	Number of Cars
0	Diesel	2019
1	Gas	613
2	Other	154
3	Petrol	1559

In [25]:
```
sns.set_palette('colorblind')
```

In [30]:
```
plt.figure(figsize=(10,8))

plt.pie(df_pie_chart['Number of Cars']
        ,labels = df_pie_chart['Engine Fuel Type'].values
        ,autopct = '%.2f%%'                                    #enables you to di
        ,textprops = {'size' : 'x-large', 'fontweight': 'bold', 'rotation': 45, 'colo

plt.legend()
# A legend is an area describing the elements of the graph

plt.title('Cars by Engine Fuel Type', fontsize=18, fontweight='bold')

plt.show()
```



AREA CHARTS

In [3]:
```
df_area_charts = pd.read_csv('C:/Users/abhinendr.sharma/Documents/Data Analyst_Udemy
df_area_charts
```

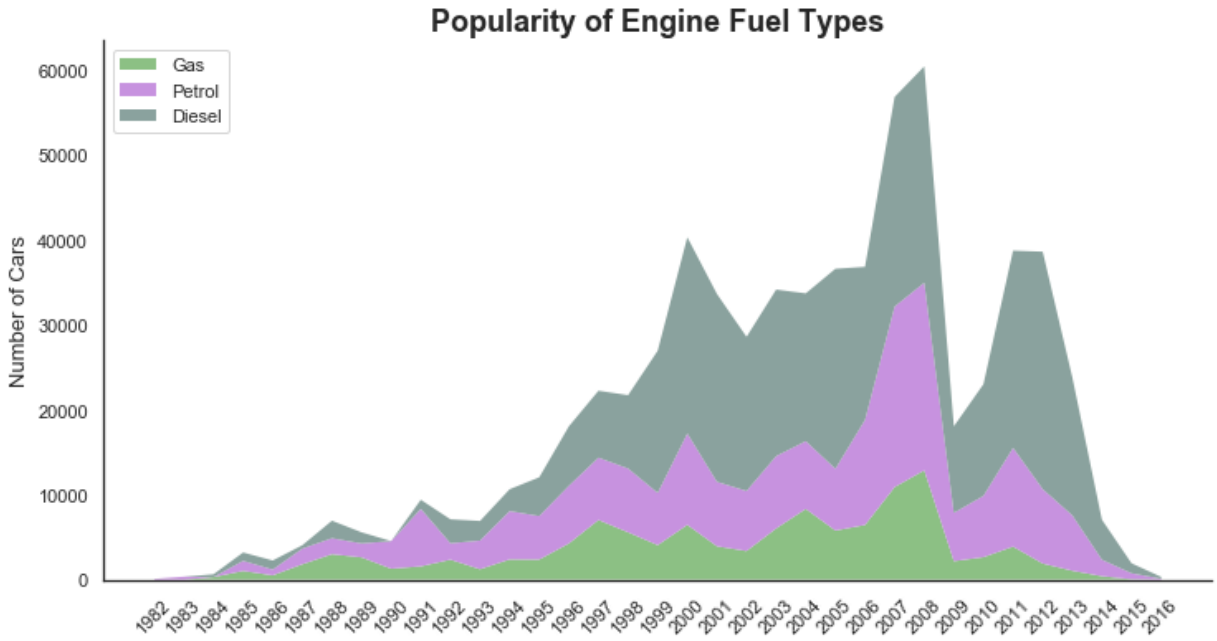
Out[3]:

	Year	Gas	Petrol	Diesel	Other
0	1982	0	94	0	0
1	1983	0	347	0	0
2	1984	300	0	334	0
3	1985	1030	1184	984	0
4	1986	511	681	1066	340
5	1987	1817	1840	400	940
6	1988	2999	1882	2060	1
7	1989	2626	1641	1304	330
8	1990	1290	3247	1	0
9	1991	1566	6776	1063	400
10	1992	2358	1925	2824	350
11	1993	1231	3375	2302	0
12	1994	2386	5691	2569	77
13	1995	2373	5115	4563	666
14	1996	4264	6792	6996	1235

	Year	Gas	Petrol	Diesel	Other
15	1997	7031	7336	7877	370
16	1998	5578	7511	8634	310
17	1999	4069	6147	16742	1493
18	2000	6459	10754	23102	531
19	2001	3932	7590	22115	1029
20	2002	3375	7069	18144	479
21	2003	6032	8518	19605	281
22	2004	8332	7978	17400	1821
23	2005	5785	7269	23551	1918
24	2006	6439	12372	18024	1878
25	2007	10902	21258	24683	1525
26	2008	12876	22090	25471	1516
27	2009	2179	5682	10191	1204
28	2010	2649	7222	13150	442
29	2011	3893	11628	23221	1688
30	2012	1899	8763	27958	1736
31	2013	1055	6517	16255	607
32	2014	409	1942	4703	329
33	2015	50	702	1172	125
34	2016	0	108	220	25

In [23]:

```
plt.figure(figsize=(12,6))

colors=['#8CC084','#C792DF','#8AA29E']
labels=['Gas','Petrol','Diesel']

plt.stackplot(df_area_charts['Year']
              ,df_area_charts['Gas']
              ,df_area_charts['Petrol']
              ,df_area_charts['Diesel']
              ,colors = colors
              ,edgecolor = 'None') #Outliners

plt.xticks(df_area_charts['Year'], rotation = 45)

plt.legend(labels = labels, loc = "upper left")

plt.title('Popularity of Engine Fuel Types', fontsize=18, fontweight='bold')

plt.ylabel("Number of Cars", fontsize=13)

sns.set_style("white") #Background is white

sns.despine() #Removing the borders of the Chart

plt.show()
```



LINE CHARTS

In [24]:
```
df_line_charts = pd.read_csv('C:/Users/abhinendr.sharma/Documents/Data Analyst_Udemy
df_line_charts
```

Out[24]:

	Date	GSPC500	FTSE100
0	1/3/2000	0.003264	0.000000
1	1/4/2000	-0.009549	0.000000
2	1/5/2000	-0.038345	-0.038137
3	1/6/2000	0.001922	-0.019502
4	1/7/2000	0.000956	-0.013571
...	...	...	...
2865	12/27/2010	-0.002282	0.002135
2866	12/28/2010	0.003539	0.000000
2867	12/29/2010	-0.000254	0.000000
2868	12/30/2010	0.000524	-0.002080
2869	12/31/2010	-0.002128	-0.004236

2870 rows × 3 columns

In [25]:
```
df_line_charts.shape
```

Out[25]:
```
(2870, 3)
```

In [26]:
```
#We need to convert Date Column in Date time format
df_line_charts['NewDate'] = pd.to_datetime(df_line_charts['Date'])
```

In [27]:
```
df_line_charts
```

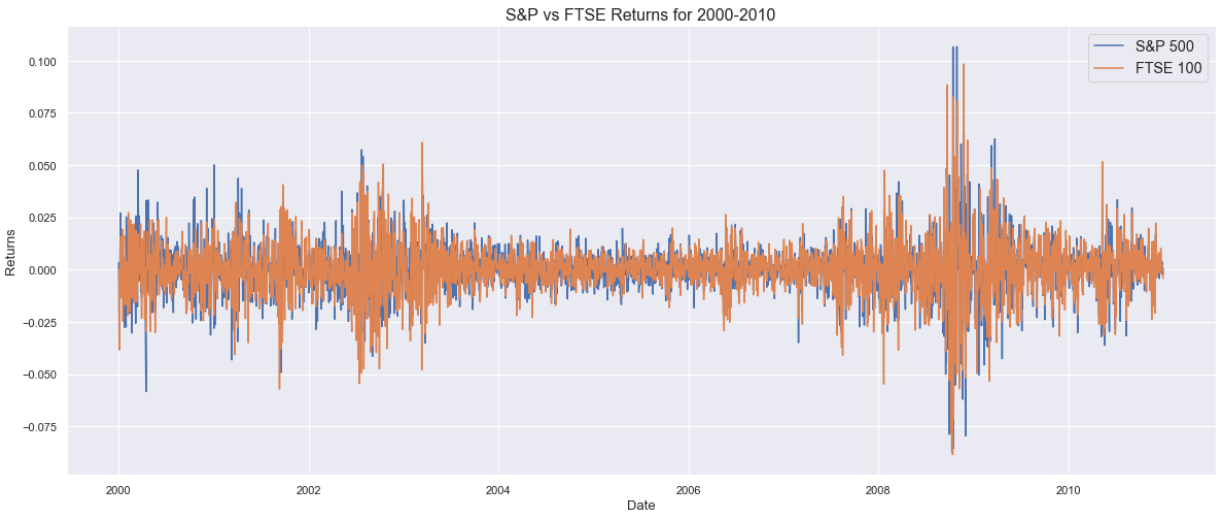
Out[27]:

	Date	GSPC500	FTSE100	NewDate
0	1/3/2000	0.003264	0.000000	2000-01-03
1	1/4/2000	-0.009549	0.000000	2000-01-04
2	1/5/2000	-0.038345	-0.038137	2000-01-05
3	1/6/2000	0.001922	-0.019502	2000-01-06
4	1/7/2000	0.000956	-0.013571	2000-01-07
...	...	...	...	...
2865	12/27/2010	-0.002282	0.002135	2010-12-27
2866	12/28/2010	0.003539	0.000000	2010-12-28
2867	12/29/2010	-0.000254	0.000000	2010-12-29
2868	12/30/2010	0.000524	-0.002080	2010-12-30
2869	12/31/2010	-0.002128	-0.004236	2010-12-31

2870 rows × 4 columns

In [40]:

```
plt.figure(figsize = (20,8)) #It should be placed before plot, defining fist


sns.set()

labels=['S&P 500', 'FTSE 100']

plt.plot(df_line_charts['NewDate']
        ,df_line_charts['GSPC500'])

plt.plot(df_line_charts['NewDate']
        ,df_line_charts['FTSE100'])

plt.legend(labels=labels, fontsize="large")

plt.title("S&P vs FTSE Returns for 2000-2010", fontsize=16)

plt.ylabel("Returns", fontsize=13)

plt.xlabel("Date", fontsize=13)

plt.show()
```



In [41]:
```
#Constructing a new column for year 2008
df_line_charts_08 = df_line_charts[(df_line_charts['NewDate'] >='2008-07-01')
                                   & (df_line_charts['NewDate'] <= '2008-12-31')]
```

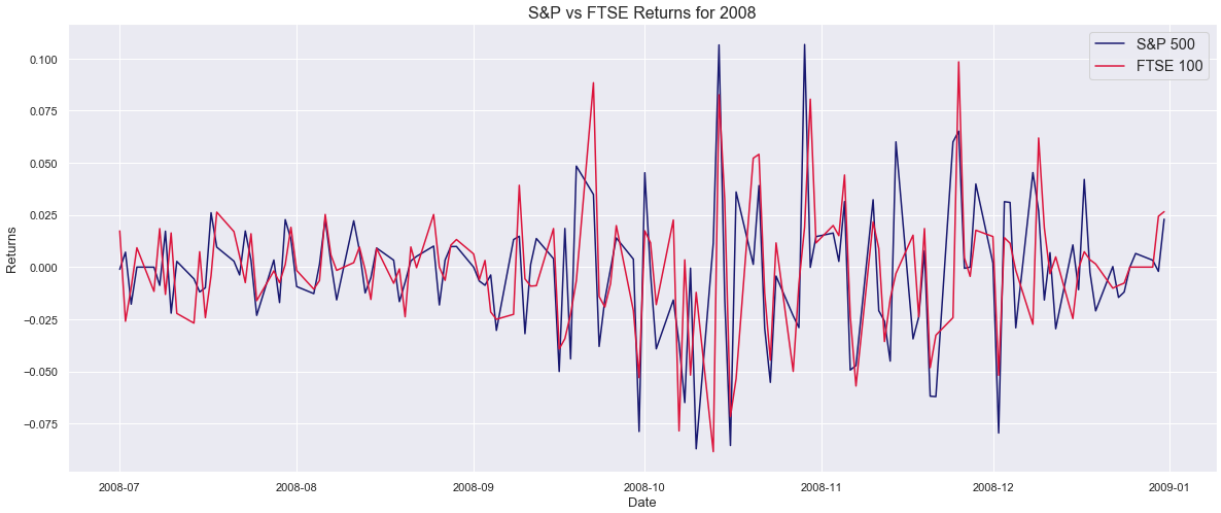
In [42]:
```
df_line_charts_08
```

Out[42]:

	Date	GSPC500	FTSE100	NewDate
2216	7/1/2008	-0.001072	0.017360	2008-07-01
2217	7/2/2008	0.007151	-0.025951	2008-07-02
2218	7/3/2008	-0.017779	-0.009781	2008-07-03
2219	7/4/2008	0.000000	0.009270	2008-07-04
2220	7/7/2008	-0.000047	-0.011650	2008-07-07
...	...	...	...	...
2343	12/25/2008	0.000000	0.000000	2008-12-25
2344	12/26/2008	0.006529	0.000000	2008-12-26
2345	12/29/2008	0.003289	0.000000	2008-12-29
2346	12/30/2008	-0.002052	0.024380	2008-12-30
2347	12/31/2008	0.022985	0.026578	2008-12-31

132 rows × 4 columns

In [43]:
```
plt.figure(figsize = (20,8)) #It should be placed before plot, defining fist


sns.set()

labels=['S&P 500', 'FTSE 100']

plt.plot(df_line_charts_08['NewDate']
        ,df_line_charts_08['GSPC500']
        ,color='midnightblue')

plt.plot(df_line_charts_08['NewDate']
        ,df_line_charts_08['FTSE100']
        ,color='crimson')

plt.legend(labels=labels, fontsize="large")

plt.title("S&P vs FTSE Returns for 2008", fontsize=16)

plt.ylabel("Returns", fontsize=13)

plt.xlabel("Date", fontsize=13)

plt.show()

#September 2008 The Great Recession
```



HISTOGRAM CHART

In [46]:
```
df_histo_charts = pd.read_csv('C:/Users/abhinendr.sharma/Documents/Data Analyst_Udem
```

In [47]:
```
df_histo_charts
```

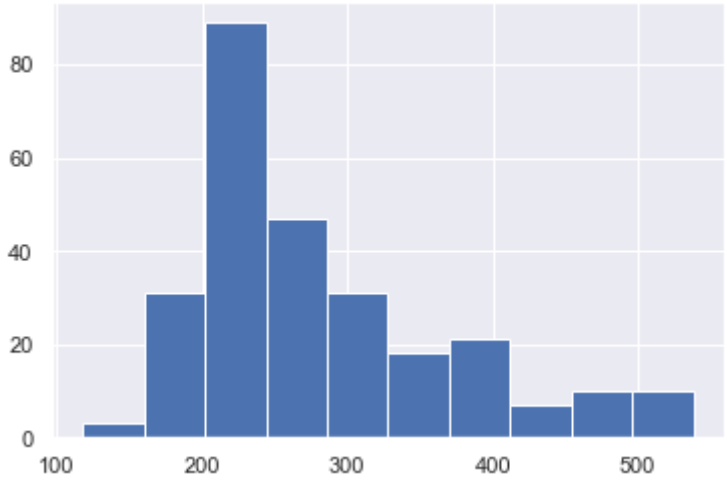
Out[47]:

	ID	Building Type	Year of sale	Month of sale	Type of property	Property #	Area (ft.)	Price
0	1030	1	2005.0	11.0	Apartment	30	743.09	246
1	1029	1	2005.0	10.0	Apartment	29	756.21	246
2	2002	2	2007.0	7.0	Apartment	2	587.28	209
3	2031	2	2007.0	12.0	Apartment	31	1604.75	453
4	1049	1	2004.0	11.0	Apartment	49	1375.45	467
...	...	...	...	...	...	...	...	...
262	5044	5	NaN	NaN	Apartment	44	1238.58	323
263	5047	5	NaN	NaN	Apartment	47	794.52	279
264	5048	5	NaN	NaN	Apartment	48	1013.27	288
265	5050	5	NaN	NaN	Apartment	50	1074.71	366
266	5051	5	NaN	NaN	Apartment	51	789.25	199

267 rows × 8 columns

In [48]:
```
plt.hist(df_histo_charts['Price']) #Bin Size is default 10 Price is X axis here
```

Out[48]:
```
(array([ 3., 31., 89., 47., 31., 18., 21.,  7., 10., 10.]),
 array([118., 160., 202., 244., 286., 328., 370., 412., 454., 496., 538.]),
 <a list of 10 Patch objects>)
```



In [56]:

```
plt.figure(figsize=(8,6))

sns.set_style('white')

plt.hist(df_histo_charts['Price']
        ,bins=8
        ,color='#108A99')

plt.title('Distribution of Real Estate Prices', fontsize=16, weight='bold')

plt.xlabel("Price in (000' $)")
plt.ylabel("Number of Properties")

sns.despine()

plt.show()
```

SCATTER PLOT

In [59]:

```
df_scatter_plot = pd.read_csv('C:/Users/abhinendr.sharma/Documents/Data Analyst_Udem
df_scatter_plot
```

Out[59]:

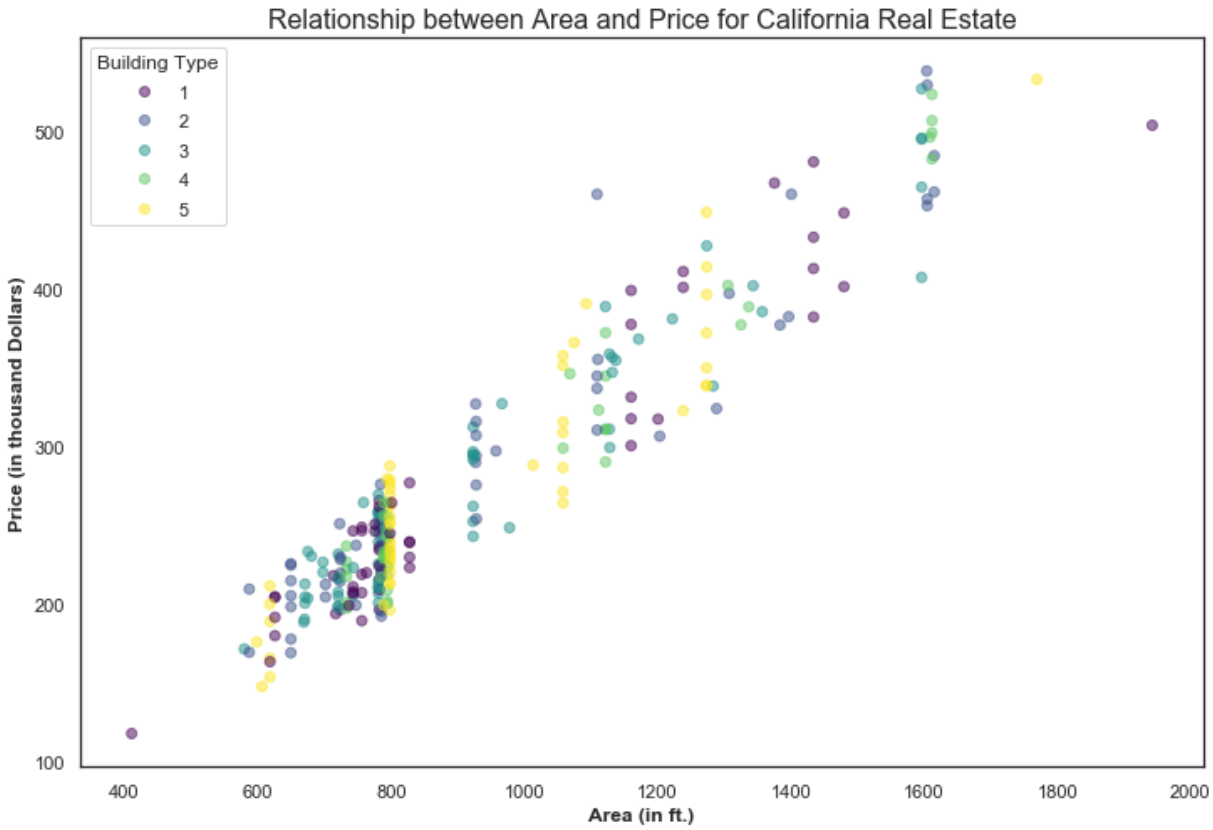
	ID	Building Type	Year of sale	Month of sale	Type of property	Property #	Area (ft.)	Price	Status
0	1030	1	2005.0	11.0	Apartment	30	743.09	246.17268	Sold
1	1029	1	2005.0	10.0	Apartment	29	756.21	246.33190	Sold
2	2002	2	2007.0	7.0	Apartment	2	587.28	209.28091	Sold
3	2031	2	2007.0	12.0	Apartment	31	1604.75	452.66701	Sold
4	1049	1	2004.0	11.0	Apartment	49	1375.45	467.08331	Sold
...	...	...	...	...	...	...	...	...	...
262	5044	5	NaN	NaN	Apartment	44	1238.58	322.61074	Not Sold
263	5047	5	NaN	NaN	Apartment	47	794.52	279.19126	Not Sold
264	5048	5	NaN	NaN	Apartment	48	1013.27	287.99653	Not Sold
265	5050	5	NaN	NaN	Apartment	50	1074.71	365.86878	Not Sold
266	5051	5	NaN	NaN	Apartment	51	789.25	199.21640	Not Sold

267 rows × 9 columns

In [60]:
```
df_scatter_plot.isnull().sum()
```

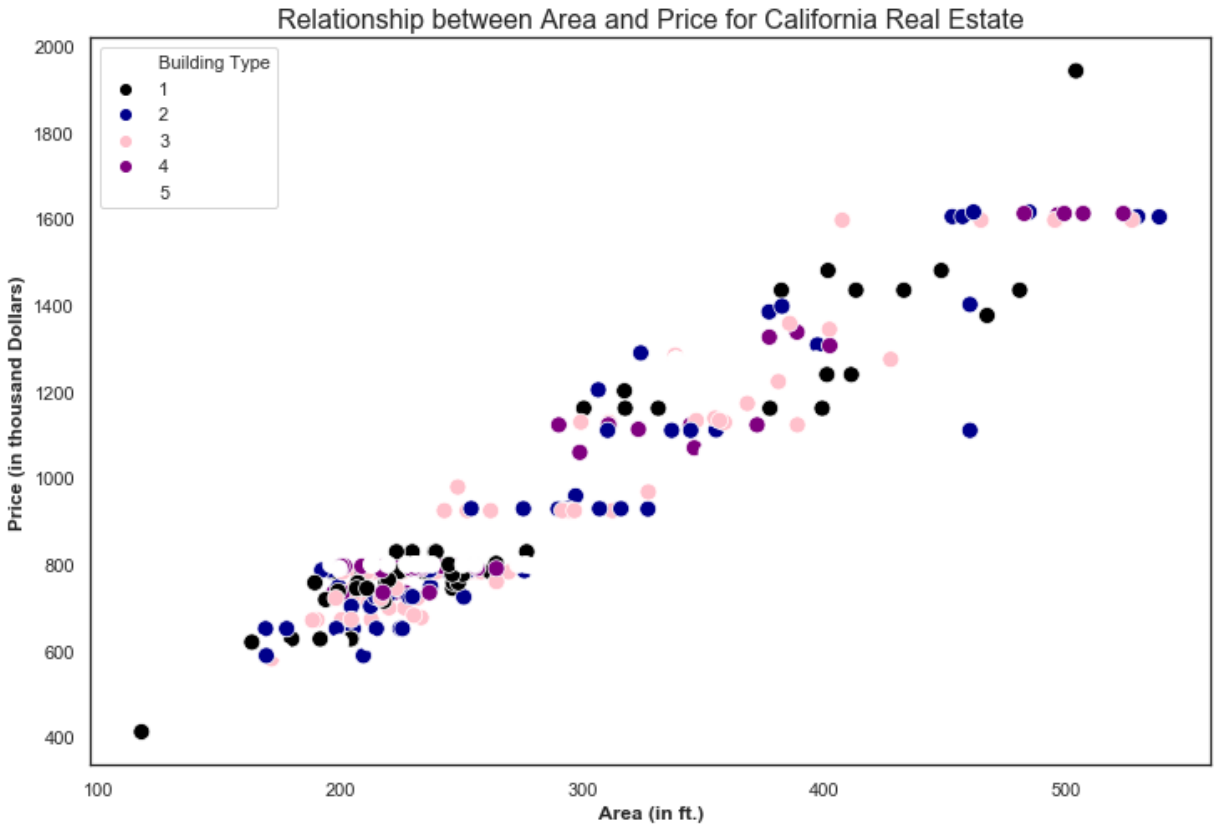
Out[60]:
```
ID                  0
Building Type       0
Year of sale       72
Month of sale      72
Type of property    0
Property #          0
Area (ft.)          0
Price               0
Status              0
dtype: int64
```

In [67]:
```
plt.figure(figsize=(12,8))

scatter = plt.scatter(df_scatter_plot['Area (ft.)']
           ,df_scatter_plot['Price']
           ,alpha = 0.5                          #Transparency of points
           ,c = df_scatter_plot['Building Type']   #Color code due to third variable
           ,cmap = 'viridis')  #Color

plt.legend(*scatter.legend_elements()
          ,loc = 'upper left'
          ,title= 'Building Type')

plt.title('Relationship between Area and Price for California Real Estate', fontsize

plt.xlabel('Area (in ft.)', weight='bold')
plt.ylabel('Price (in thousand Dollars)', weight='bold')
plt.show()
```



In [68]:
```
#Scatter usign Seaborn
```

In [74]:
```
plt.figure(figsize=(12,8))

sns.scatterplot(df_scatter_plot['Price']
                ,df_scatter_plot['Area (ft.)']
                ,hue = df_scatter_plot['Building Type']
                ,palette = ['black', 'darkblue', 'pink', 'purple', 'white']
                ,s = 100)

plt.title('Relationship between Area and Price for California Real Estate', fontsize

plt.xlabel('Area (in ft.)', weight='bold')
plt.ylabel('Price (in thousand Dollars)', weight='bold')
plt.show()
```



REGRESSION LINE CHART

In [75]:
```
df_reg_line = pd.read_csv('C:/Users/abhinendr.sharma/Documents/Data Analyst_Udemy/Ud
df_reg_line
```

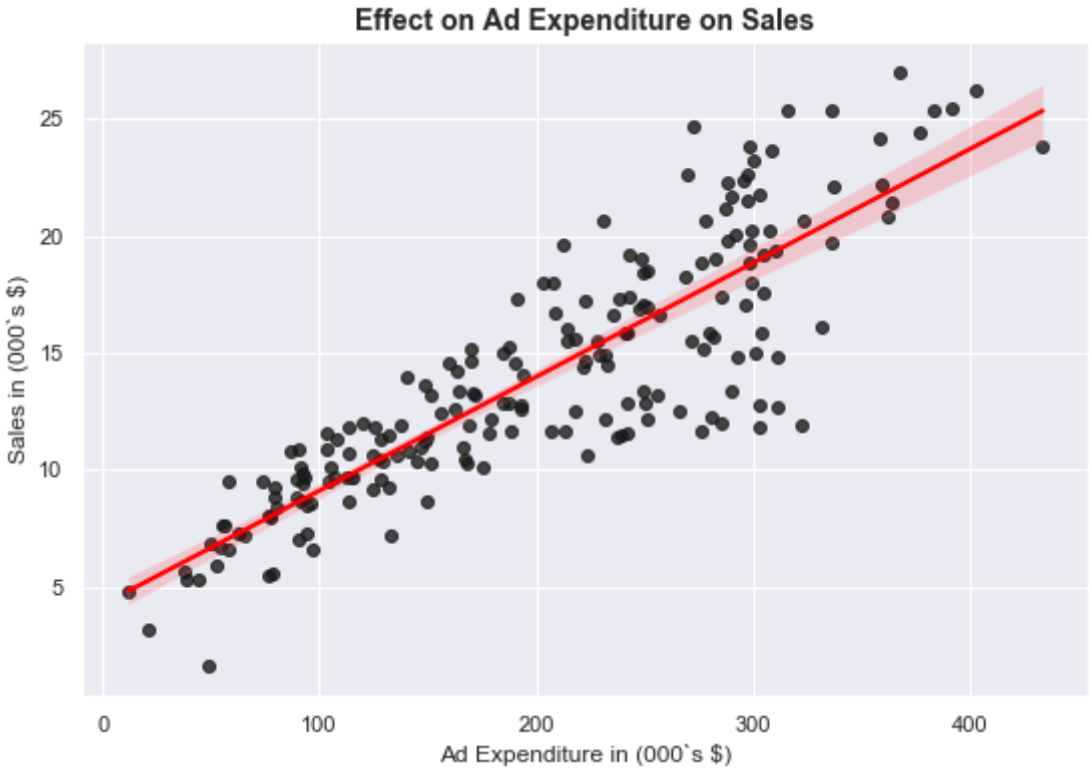
Out[75]:

	Budget	Sales
0	337.1	22.1
1	128.9	10.4
2	132.4	9.3
3	251.3	18.5
4	250.0	12.9
...	...	...
195	55.7	7.6
196	107.2	9.7
197	192.7	12.8
198	391.8	25.5
199	249.4	13.4

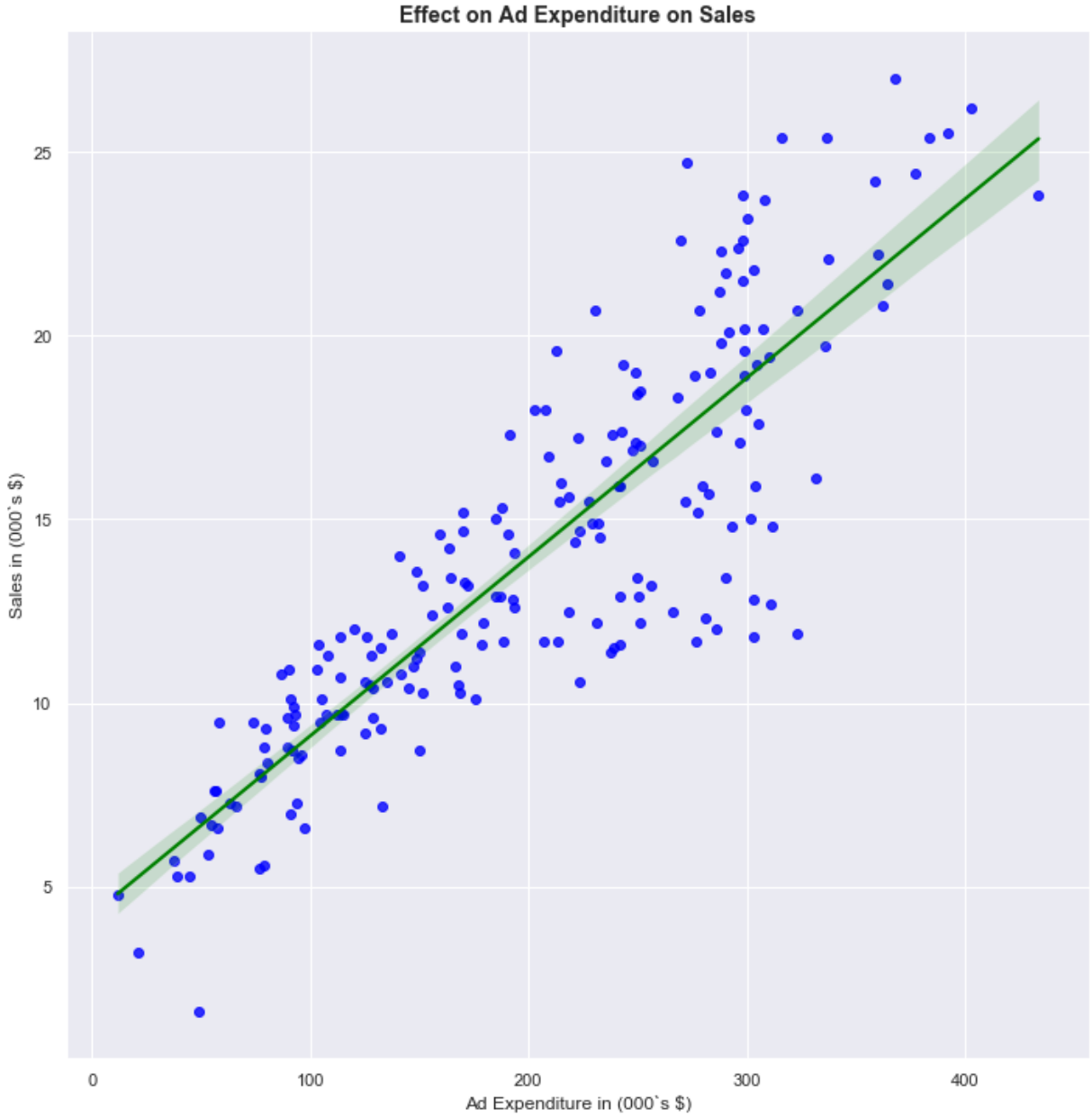
200 rows × 2 columns

In [76]:
```
#We rely on Seaborn for Regression Analysis
#sns.regplot() & sns.lmplot()

#sns.regplot() supports x and y variables in Numpy arrays , pandas arrays
#sns.Lmplot() supports x and y variables in form of strings only (longform and tidyd
```

In [84]:
```
#plt.figure(figsize=(10,8))
sns.set(rc = {'figure.figsize': (9,6)})

sns.set_style('darkgrid')

sns.regplot(x ='Budget'
            ,y = 'Sales'
            ,data = df_reg_line
            ,scatter_kws = {'color':'k'}
            ,line_kws = {'color':'red'})

plt.xlabel('Ad Expenditure in (000`s $)')
plt.ylabel('Sales in (000`s $)')
plt.title('Effect on Ad Expenditure on Sales', fontsize=14 , weight = 'bold')


plt.show()
```



In [89]:
```
sns.set( rc = {'figure.figsize': (15,8)}) #This will not work in lmplot()

sns.lmplot(x ='Budget'
           ,y ='Sales'
           ,data = df_reg_line
           ,height = 10
           ,scatter_kws = {'color':'blue'}
           ,line_kws = {'color':'green'})


plt.xlabel('Ad Expenditure in (000`s $)')
plt.ylabel('Sales in (000`s $)')
plt.title('Effect on Ad Expenditure on Sales', fontsize=14 , weight = 'bold')

plt.show()
```



PARETO CHART

In [111…

```
import pandas as pd
import matplotlib.pyplot as plt

import matplotlib.ticker as mtick #Format numbers as a percentage
```

In [100…

```
df_pareto = pd.read_csv('C:/Users/abhinendr.sharma/Documents/Data Analyst_Udemy/Udem
df_pareto
```

Out[100…

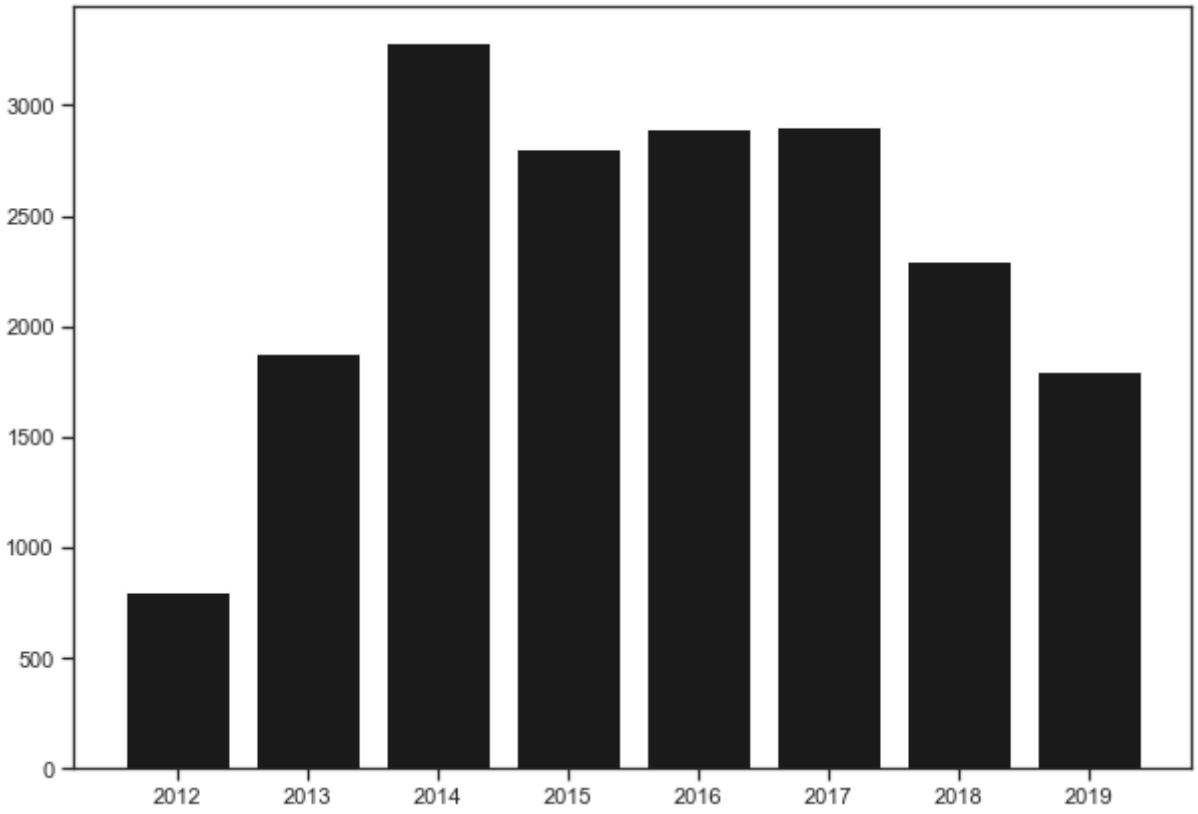
	Year	Python Users	Participants
0	2012	0.149	798
1	2013	0.133	1880
2	2014	0.195	3285
3	2015	0.303	2800
4	2016	0.458	2895
5	2017	0.526	2900

	Year	Python Users	Participants
6	2018	0.656	2300
7	2019	0.658	1800

In [101…

```
fig, ax= plt.subplots(figsize = (10,7))

ax.bar(df_pareto['Year']
       ,df_pareto['Participants']
       ,color = 'k')
```

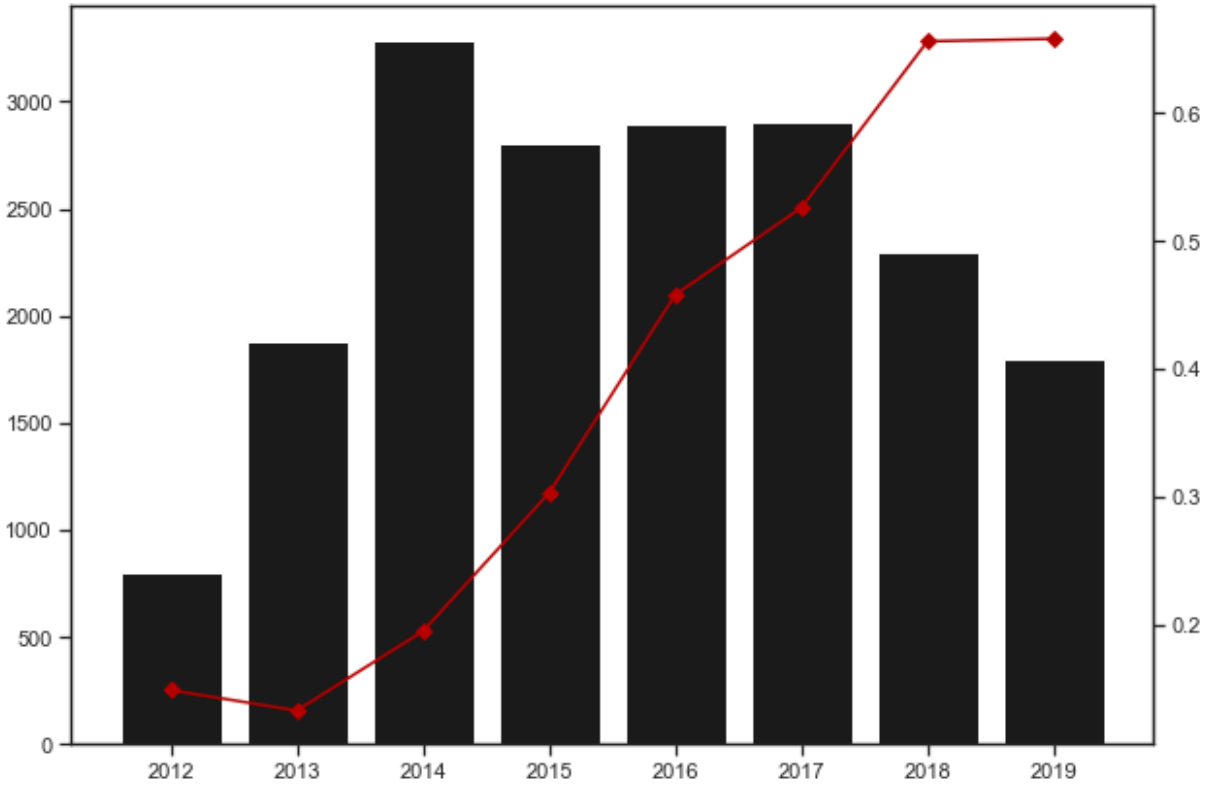
Out[101… <BarContainer object of 8 artists>



In [102…

```
fig, ax= plt.subplots(figsize = (10,7))

sns.set_style('ticks') #Displays data over everything

ax.bar(df_pareto['Year']
       ,df_pareto['Participants']
       ,color = 'k')

ax1=ax.twinx()
ax1.plot(df_pareto['Year']
         ,df_pareto['Python Users']
         ,color = '#b60000'
         ,marker = 'D')
```

Out[102… [<matplotlib.lines.Line2D at 0x1cf047fa188>]

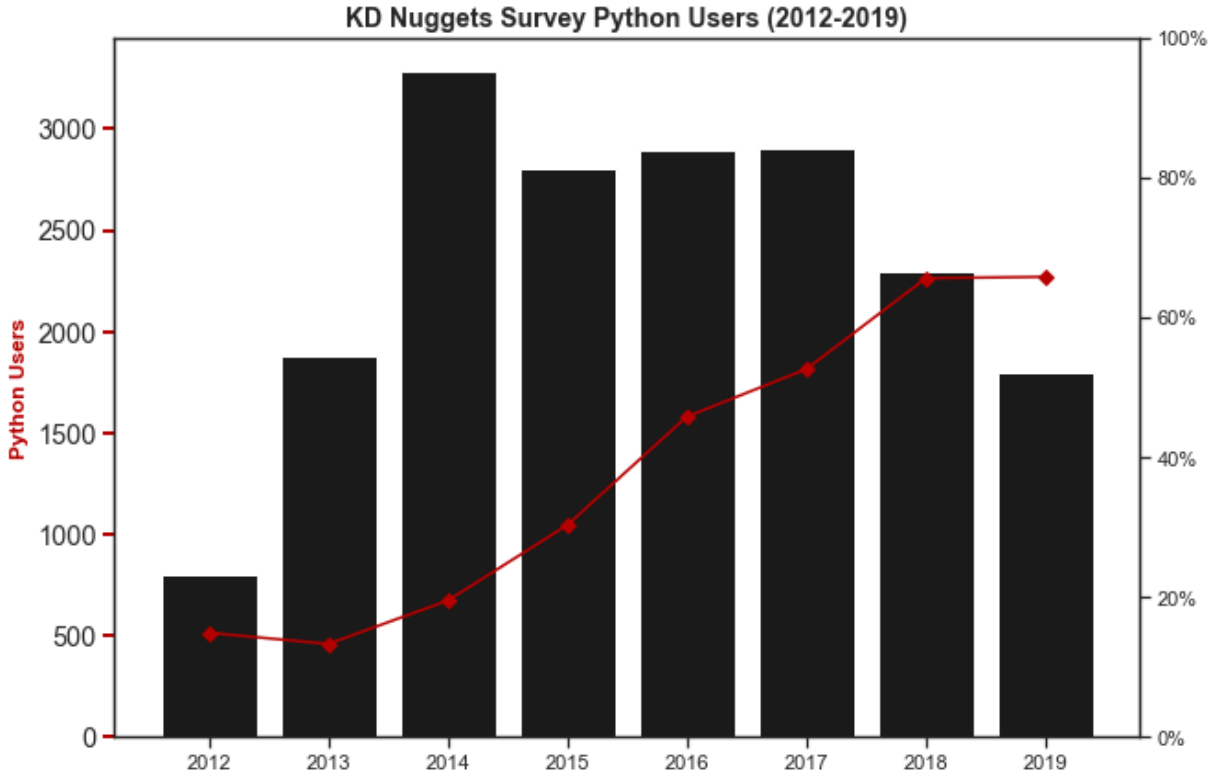


In [113…

```
fig, ax= plt.subplots(figsize = (10,7))

sns.set_style('ticks') #Displays data over everything

ax.bar(df_pareto['Year']
      ,df_pareto['Participants']
      ,color = 'k')

ax.set_ylabel('Number of Participants', weight='bold')
ax.tick_params(axis="y", width=2, labelsize='large')

ax1 = ax.twinx()
ax1.set_ylim(0,1)

ax1.yaxis.set_major_formatter(mtick.PercentFormatter(xmax=1.0))

ax1.plot(df_pareto['Year']
        ,df_pareto['Python Users']
        ,color = '#b60000'
        ,marker = 'D')

ax.set_ylabel('Python Users', color = '#b60000', weight='bold')
ax.tick_params(axis='y', width=2, labelsize='large', color='#b50000')

ax.set_title("KD Nuggets Survey Python Users (2012-2019)", fontsize=14, weight='bold
```

Out[113…

```
Text(0.5, 1.0, 'KD Nuggets Survey Python Users (2012-2019)')
```


KD Nuggets Survey Python Users (2012-2019)
Python Users
0
500
1000
1500
2000
2500
3000
0%
20%
40%
60%
80%
100%
2012
2013
2014
2015
2016
2017
2018
2019

In [ ]: