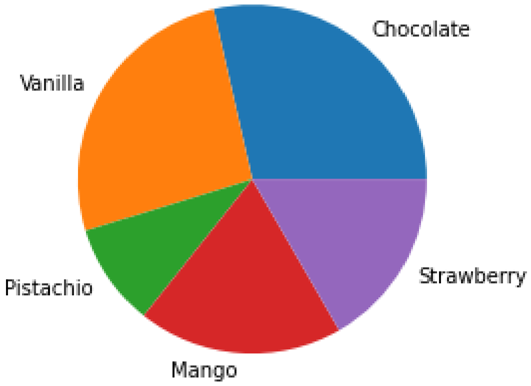
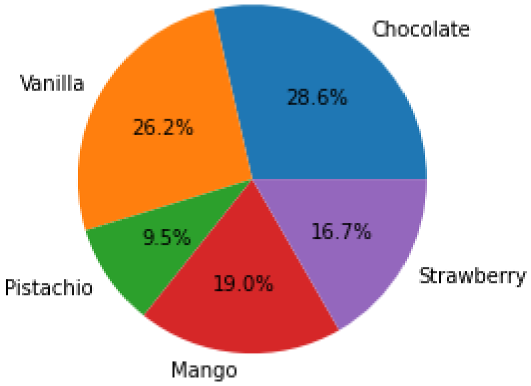
In [1]:

```python
import matplotlib.pyplot as plt
flavors = ('Chocolate', 'Vanilla', 'Pistachio', 'Mango', 'Strawberry')
votes = (12, 11, 4, 8, 7)
plt.pie(
 votes,
 labels=flavors
)
plt.show()
```



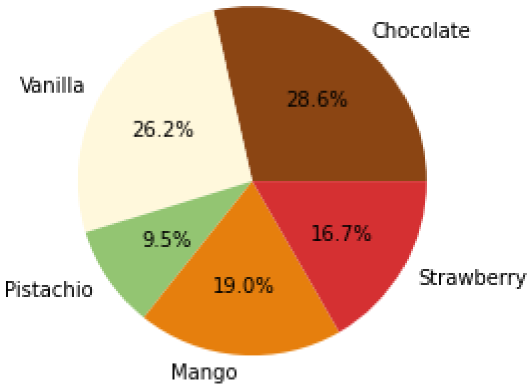
In [2]:

```python
import matplotlib.pyplot as plt
flavors = ('Chocolate', 'Vanilla', 'Pistachio', 'Mango', 'Strawberry')
votes = (12, 11, 4, 8, 7)
plt.pie(
 votes,
 labels=flavors,
 autopct='%1.1f%%' #menambahkan persentase
)
plt.show()
```



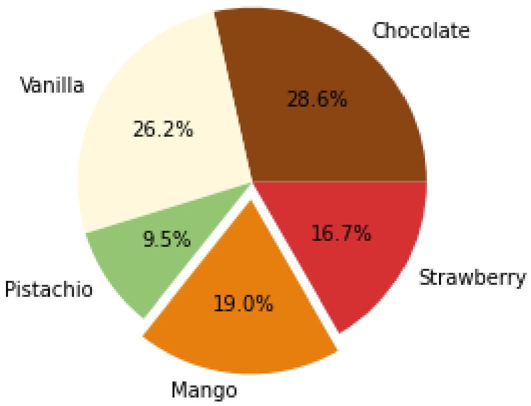
In [3]:

```python
import matplotlib.pyplot as plt
flavors = ('Chocolate', 'Vanilla', 'Pistachio', 'Mango', 'Strawberry')
votes = (12, 11, 4, 8, 7)
colors = ('#8B4513', '#FFF8DC', '#93C572', '#E67F0D', '#D53032') #warna itu rasa
plt.pie(votes,
 labels=flavors,
 autopct='%1.1f%%',
 colors=colors, #apply color
)
plt.show()
```



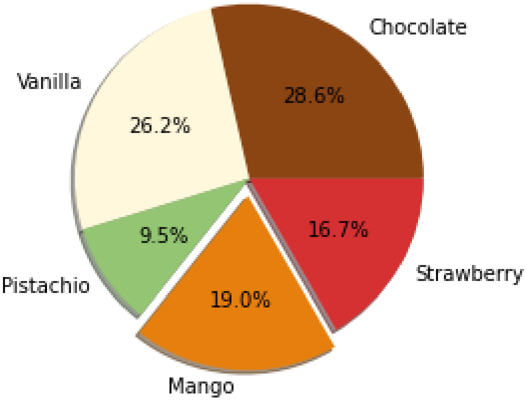
In [4]:

```
import matplotlib.pyplot as plt
flavors = ('Chocolate', 'Vanilla', 'Pistachio', 'Mango', 'Strawberry')
votes = (12, 11, 4, 8, 7)
colors = ('#8B4513', '#FFF8DC', '#93C572', '#E67F0D', '#D53032')
explode = (0, 0, 0, 0.1, 0) #highlight
plt.pie( votes,
 labels=flavors,
 autopct='%1.1f%%',
 colors=colors,
 explode=explode, #apply highlight in the chart
)
plt.show()
```



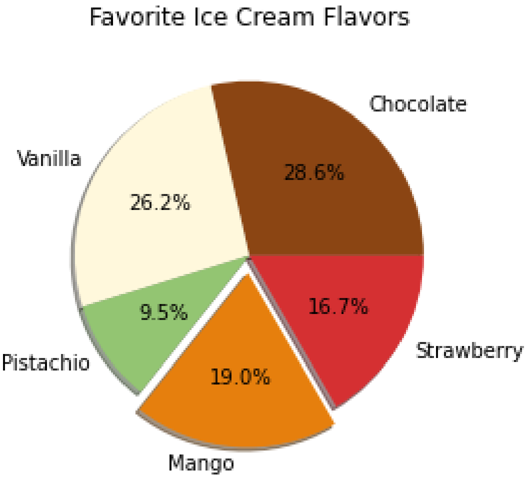
In [5]:

```
import matplotlib.pyplot as plt
flavors = ('Chocolate', 'Vanilla', 'Pistachio', 'Mango', 'Strawberry')
votes = (12, 11, 4, 8, 7)
colors = ('#8B4513', '#FFF8DC', '#93C572', '#E67F0D', '#D53032')
explode = (0, 0, 0, 0.1, 0)
plt.pie(votes,
 labels=flavors,
 autopct='%1.1f%%',
 colors=colors,
 explode=explode,
 shadow=True #3D effect
)
plt.show()
```



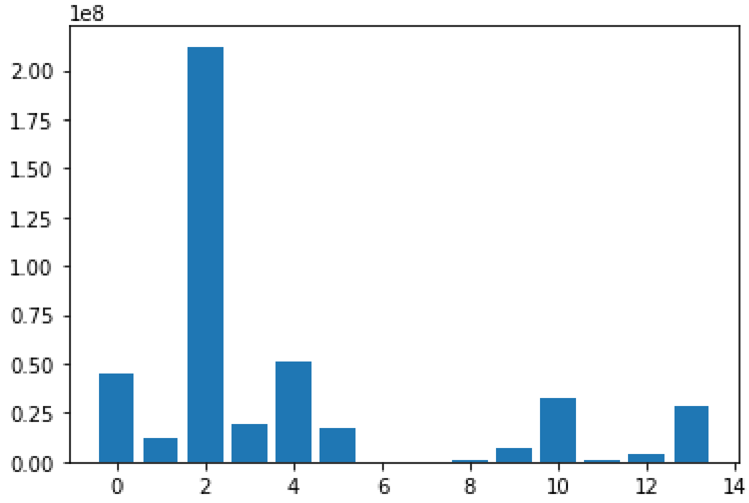
In [6]:

```
import matplotlib.pyplot as plt
flavors = ('Chocolate', 'Vanilla', 'Pistachio', 'Mango', 'Strawberry')
votes = (12, 11, 4, 8, 7)
colors = ('#8B4513', '#FFF8DC', '#93C572', '#E67F0D', '#D53032')
explode = (0, 0, 0, 0.1, 0)
plt.title('Favorite Ice Cream Flavors') #add title
plt.pie(votes,
 labels=flavors,
 autopct='%1.1f%%',
 colors=colors,
 explode=explode,
 shadow=True
)
plt.show()
```



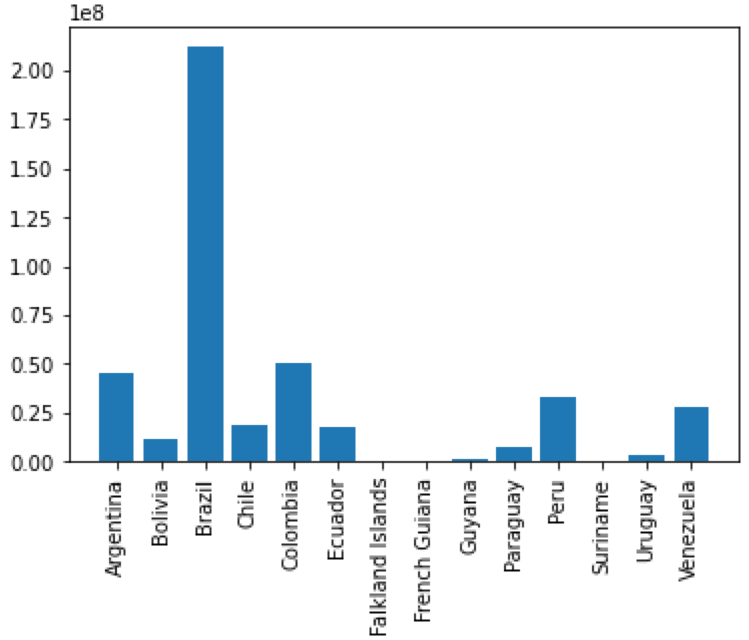
In [10]:

```
import matplotlib.pyplot as plt
import numpy as np
countries = ('Argentina', 'Bolivia', 'Brazil', 'Chile', 'Colombia','Ecuador','Falkla
populations = (45076704, 11626410, 212162757, 19109629, 50819826, 17579085,3481, 287
x_coords = np.arange(len(countries))
plt.bar(x_coords, populations)
plt.show()
```



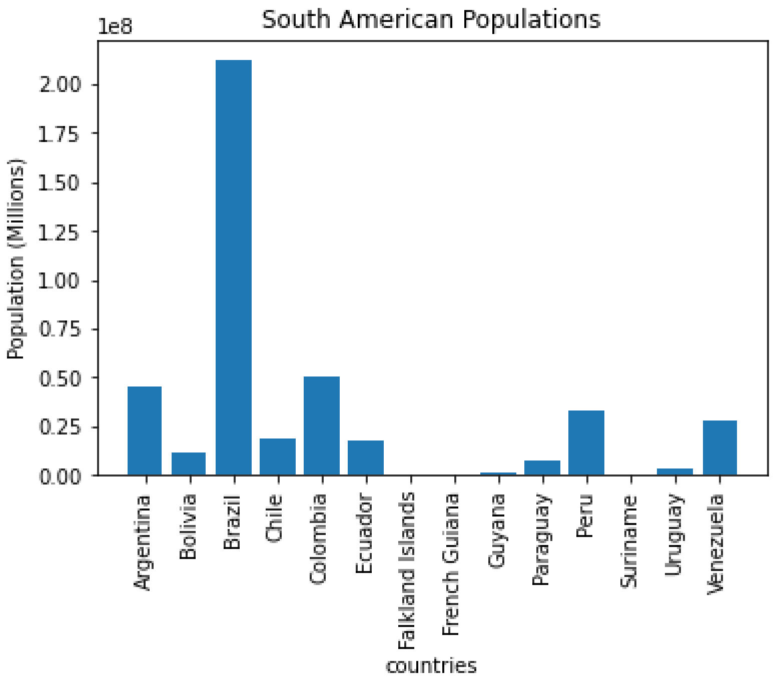
In [13]:

```
import matplotlib.pyplot as plt
import numpy as np
countries = ('Argentina', 'Bolivia', 'Brazil', 'Chile', 'Colombia','Ecuador','Falkla
populations = (45076704, 11626410, 212162757, 19109629, 50819826, 17579085,3481, 287
x_coords = np.arange(len(countries))
plt.bar(x_coords, populations, tick_label=countries) #add tick_label
plt.xticks(rotation=90) #rotates text for x-axis labels
plt.show()
```



In [14]:

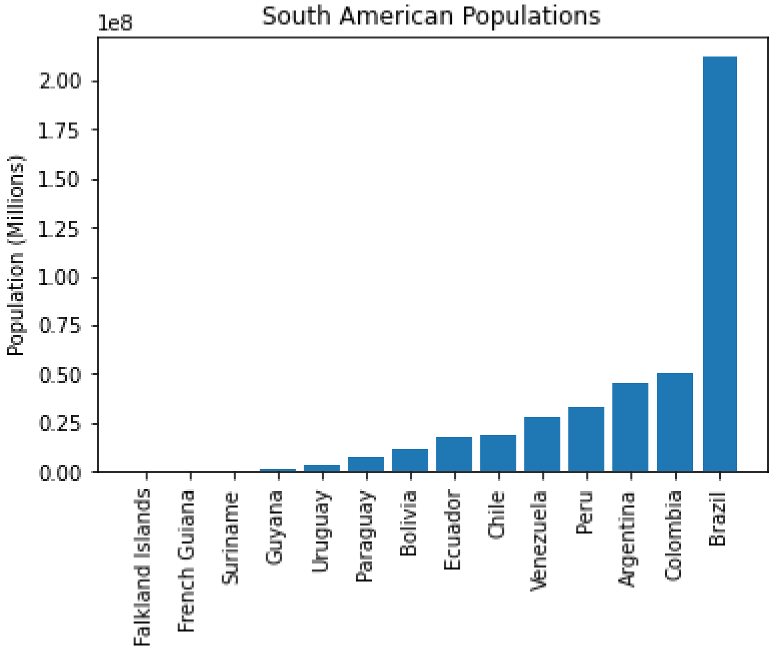
```
import matplotlib.pyplot as plt
import numpy as np
countries = ('Argentina', 'Bolivia', 'Brazil', 'Chile', 'Colombia','Ecuador','Falkla
populations = (45076704, 11626410, 212162757, 19109629, 50819826, 17579085,3481, 287
x_coords = np.arange(len(countries))
plt.bar(x_coords, populations, tick_label=countries)
plt.xticks(rotation=90)
plt.xlabel('countries') #add y label
plt.ylabel('Population (Millions)') #add y label
plt.title('South American Populations') #add title
plt.show()
```



In [16]:

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd #use Pandas for sorting

countries = ('Argentina', 'Bolivia', 'Brazil', 'Chile', 'Colombia','Ecuador','Falkla
populations = (45076704, 11626410, 212162757, 19109629, 50819826, 17579085,3481, 287
#sort countries
df = pd.DataFrame({'Country': countries,'Population': populations,})
df.sort_values(by='Population', inplace=True)
x_coords = np.arange(len(df)) #dataframe
plt.bar(x_coords, df['Population'], tick_label=df['Country']) #modified
plt.xticks(rotation=90)
plt.ylabel('Population (Millions)')
plt.title('South American Populations')
plt.show()
```

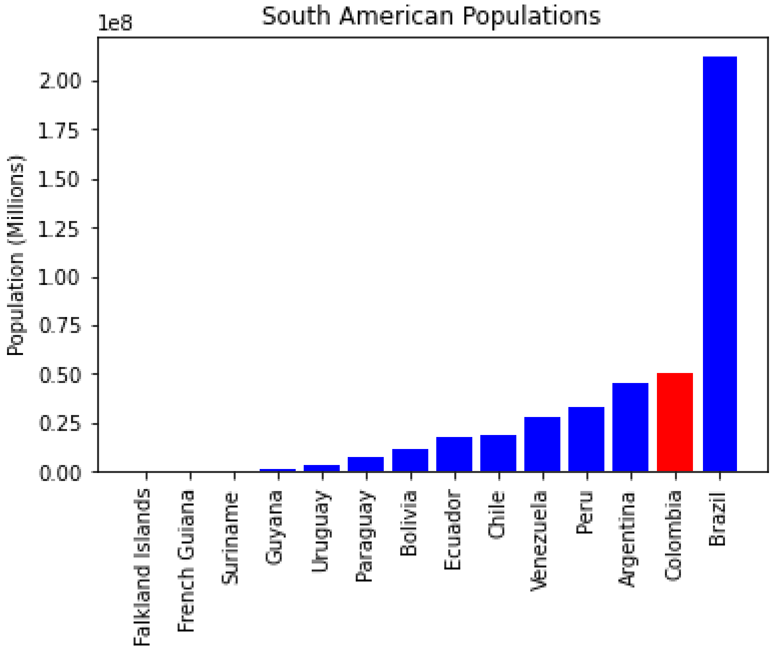


In [17]:

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd #use Pandas for sorting
```

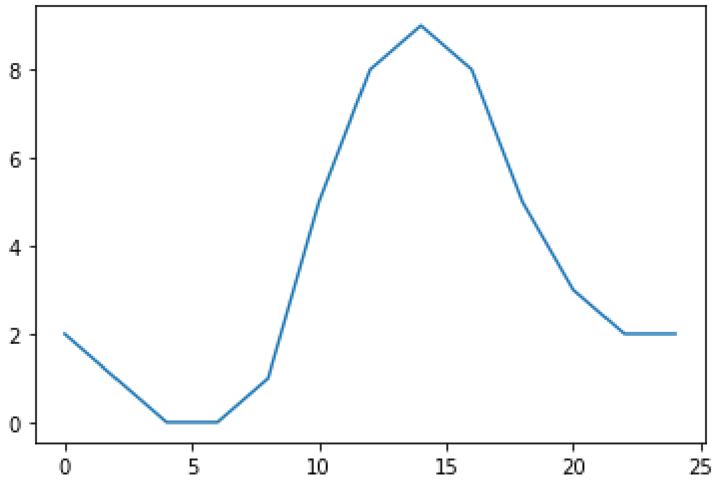
```
countries = ('Argentina', 'Bolivia', 'Brazil', 'Chile', 'Colombia','Ecuador','Falkla
populations = (45076704, 11626410, 212162757, 19109629, 50819826, 17579085,3481, 287
#sort countries
df = pd.DataFrame({'Country': countries,'Population': populations,})
df.sort_values(by='Population', inplace=True)

x_coords = np.arange(len(df))
colors = ['#0000FF' for _ in range(len(df))] #add color
colors[-2] = '#FF0000' #highlight second biggest country by population
plt.bar(x_coords, df['Population'], tick_label=df['Country'], color=colors) #apply c
plt.xticks(rotation=90)
plt.ylabel('Population (Millions)')
plt.title('South American Populations')
plt.show()
```



In [18]:

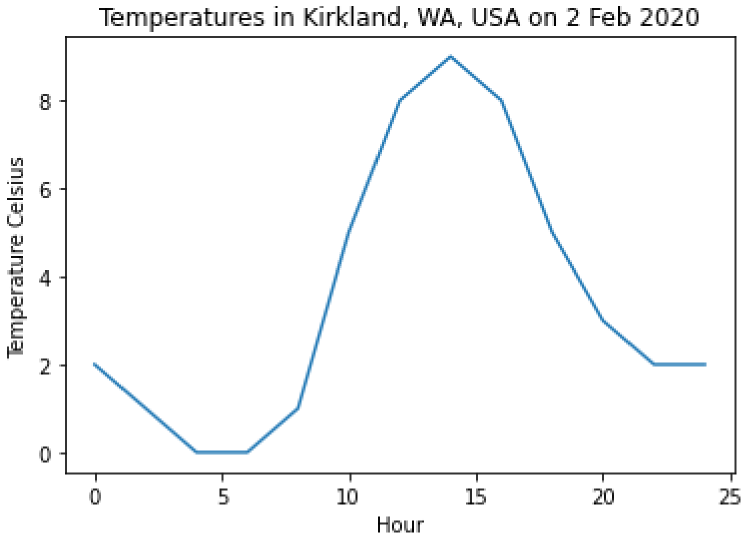
```
import matplotlib.pyplot as plt
temperature_c = [2, 1, 0, 0, 1, 5, 8, 9, 8, 5, 3, 2, 2]
hour = [0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24]
plt.plot(
 hour,
 temperature_c
)
plt.show()
```



In [19]:

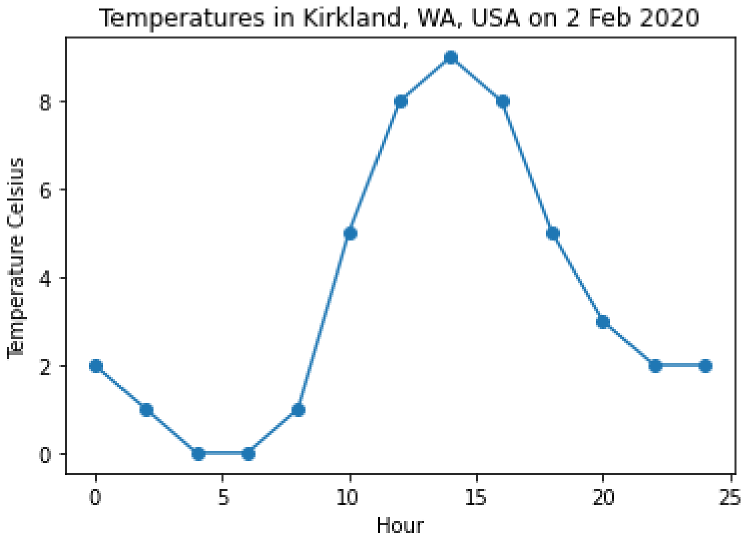
```
import matplotlib.pyplot as plt
temperature_c = [2, 1, 0, 0, 1, 5, 8, 9, 8, 5, 3, 2, 2]
```

```
hour = [0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24]
plt.plot(
 hour,
 temperature_c,)
plt.title('Temperatures in Kirkland, WA, USA on 2 Feb 2020') #add title
plt.ylabel('Temperature Celsius') #add y label
plt.xlabel('Hour') #add x label
plt.show()
```



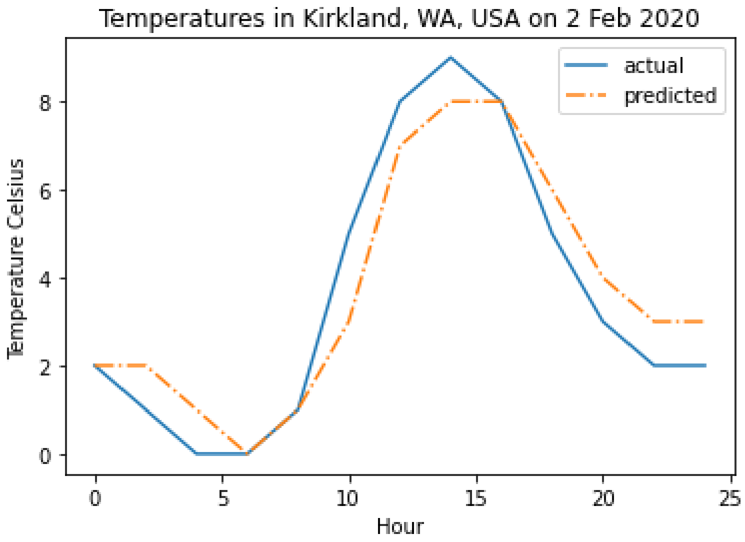
In [20]:

```
import matplotlib.pyplot as plt
temperature_c = [2, 1, 0, 0, 1, 5, 8, 9, 8, 5, 3, 2, 2]
hour = [0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24]
plt.plot(
 hour,
 temperature_c,
 marker='o', #add marker
)
plt.title('Temperatures in Kirkland, WA, USA on 2 Feb 2020')
plt.ylabel('Temperature Celsius')
plt.xlabel('Hour')
plt.show()
```



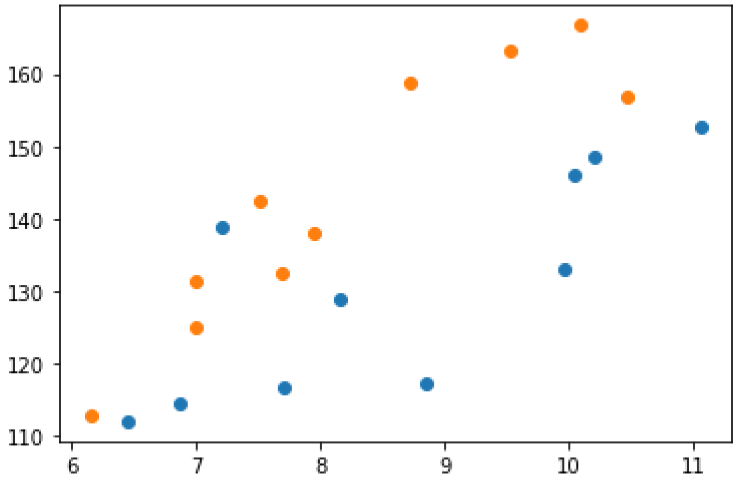
In [21]:

```
import matplotlib.pyplot as plt
fig, ax = plt.subplots()
temperature_c_actual = [2, 1, 0, 0, 1, 5, 8, 9, 8, 5, 3, 2, 2]
temperature_c_predicted = [2, 2, 1, 0, 1, 3, 7, 8, 8, 6, 4, 3, 3]
hour = [0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24]
ax.plot(hour, temperature_c_actual)
plt.plot(hour, temperature_c_predicted, linestyle='-.')
plt.title('Temperatures in Kirkland, WA, USA on 2 Feb 2020')
plt.ylabel('Temperature Celsius')
plt.xlabel('Hour')
plt.legend(['actual', 'predicted']) #add legend
plt.show()
```



In [22]:

```
import matplotlib.pyplot as plt
lemon_diameter = [6.44, 6.87, 7.7, 8.85, 8.15, 9.96, 7.21, 10.04, 10.2, 11.06]
lemon_weight = [112.05, 114.58, 116.71, 117.4, 128.93,
 132.93, 138.92, 145.98, 148.44, 152.81]
lime_diameter = [6.15, 7.0, 7.0, 7.69, 7.95, 7.51, 10.46, 8.72, 9.53, 10.09]
lime_weight = [112.76, 125.16, 131.36, 132.41, 138.08,
 142.55, 156.86, 158.67, 163.28, 166.74]
plt.scatter(lemon_diameter, lemon_weight)
plt.scatter(lime_diameter, lime_weight)
plt.show()
```

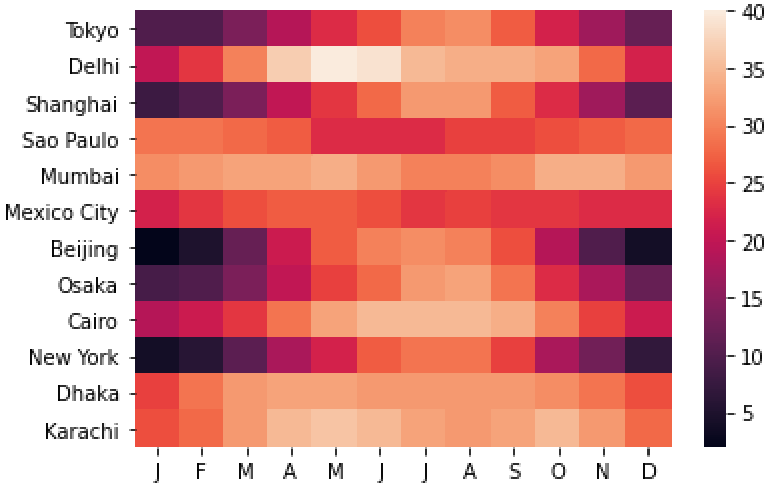


In [23]:

```
import seaborn as sns
cities = ['Tokyo', 'Delhi', 'Shanghai', 'Sao Paulo', 'Mumbai',
'Mexico City',
'Beijing', 'Osaka', 'Cairo', 'New York', 'Dhaka',
'Karachi']
months = ['J', 'F', 'M', 'A', 'M', 'J', 'J', 'A', 'S', 'O', 'N',
'D']
temperatures = [
[10, 10, 14, 19, 23, 26, 30, 31, 27, 22, 17, 12], # Tokyo
[20, 24, 30, 37, 40, 39, 35, 34, 34, 33, 28, 22], # Delhi
[ 8, 10, 14, 20, 24, 28, 32, 32, 27, 23, 17, 11], # Shanghai
[29, 29, 28, 27, 23, 23, 23, 25, 25, 26, 27, 28], # Sao Paulo
[31, 32, 33, 33, 34, 32, 30, 30, 31, 34, 34, 32], # Mumbai
[22, 24, 26, 27, 27, 26, 24, 25, 24, 24, 23, 23], # Mexico City
[ 2, 5, 12, 21, 27, 30, 31, 30, 26, 19, 10, 4], # Beijing
[ 9, 10, 14, 20, 25, 28, 32, 33, 29, 23, 18, 12], # Osaka
[19, 21, 24, 29, 33, 35, 35, 35, 34, 30, 25, 21], # Cairo
[ 4, 6, 11, 18, 22, 27, 29, 29, 25, 18, 13, 7], # New York
[25, 29, 32, 33, 33, 32, 32, 32, 32, 31, 29, 26], # Dhaka
[26, 28, 32, 35, 36, 35, 33, 32, 33, 35, 32, 28], # Karachi
]
sns.heatmap(temperatures, yticklabels=cities, xticklabels=months)
```

Out[23]:

```
<AxesSubplot:>
```

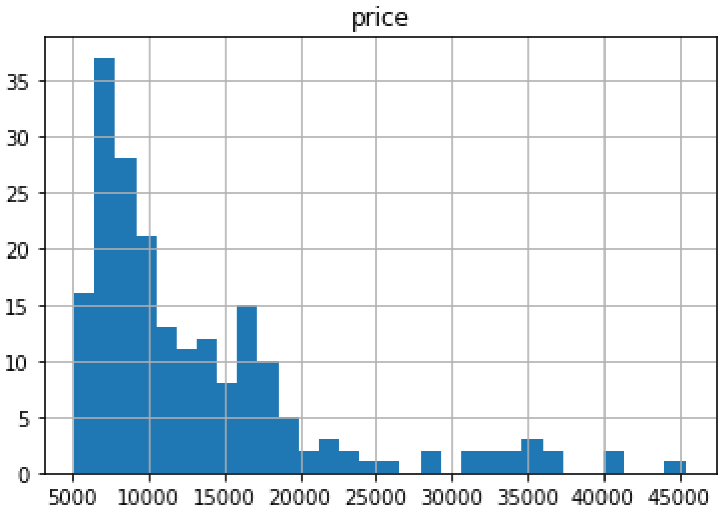


In [28]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
path='automobileEDA.csv'
df = pd.read_csv(path)
df.head()
df.hist(column='price', bins=30)
```

Out[28]:

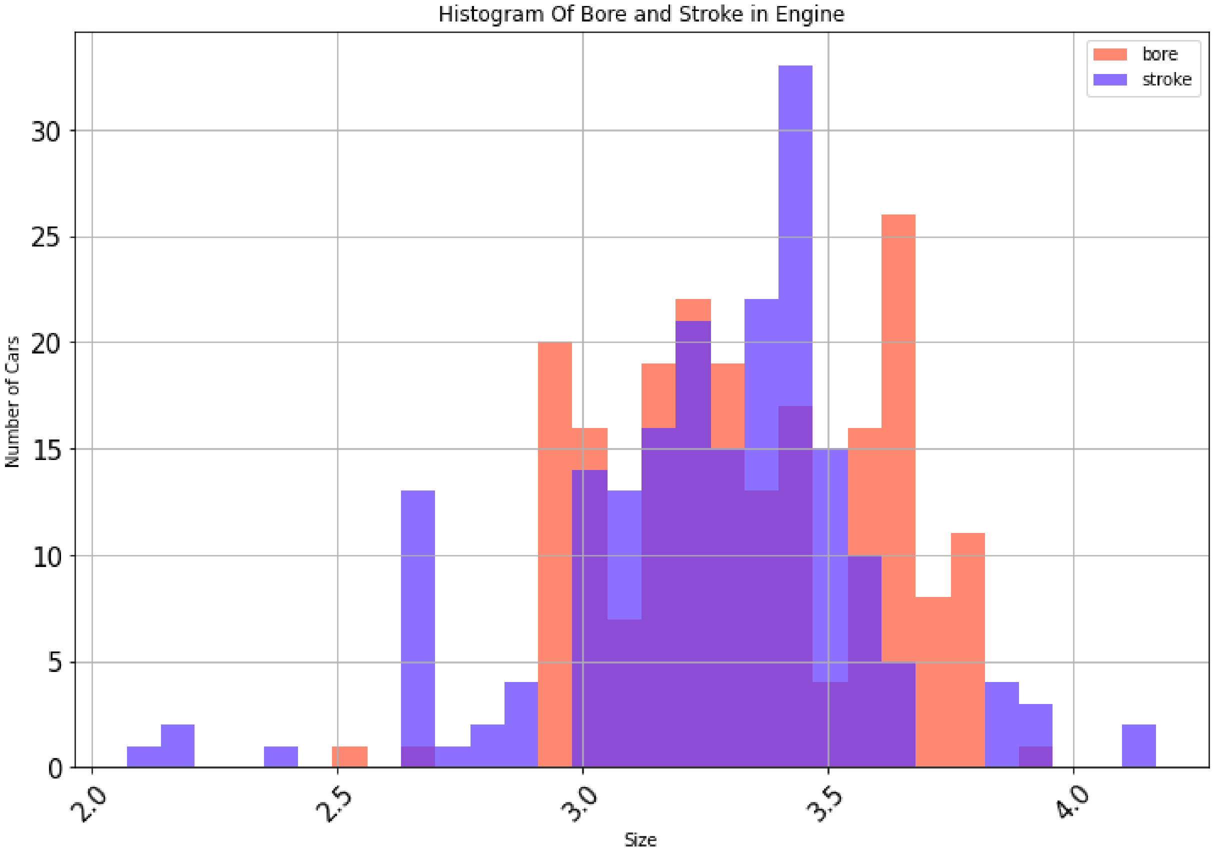
```
array([[<AxesSubplot:title={'center':'price'}>]], dtype=object)
```



In [29]:

```
df[['bore','stroke']].plot(kind='hist',alpha=0.7,
 bins=30,
 title='Histogram Of Bore and Stroke in Engine',
 rot=45,
 grid=True,
 figsize=(12,8),
 fontsize=15,
 color=['#FF5733', '#5C33FF'])
```

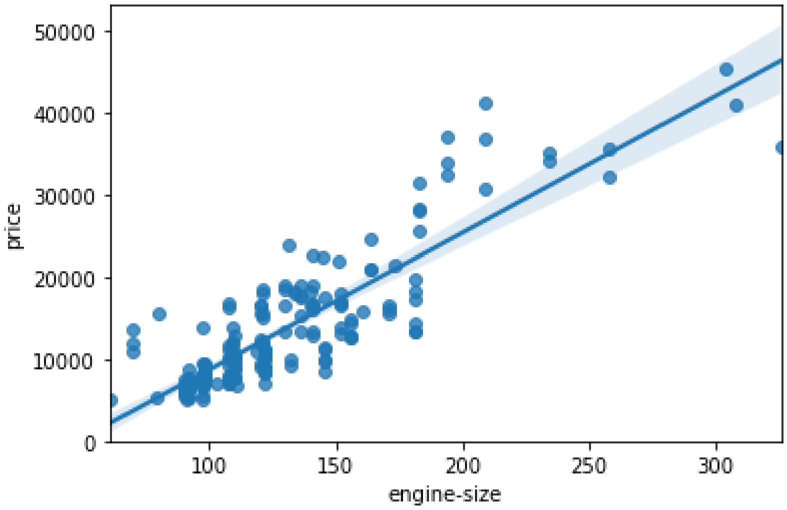
```
plt.xlabel('Size')
plt.ylabel("Number of Cars");
```



In [30]:
```
from scipy import stats
pearson_coef, p_value = stats.pearsonr(df['horsepower'], df['price'])
print("The Pearson Correlation Coefficient is", pearson_coef, " with a P-value of P
sns.regplot(x="engine-size", y="price", data=df)
plt.ylim(0,)
```

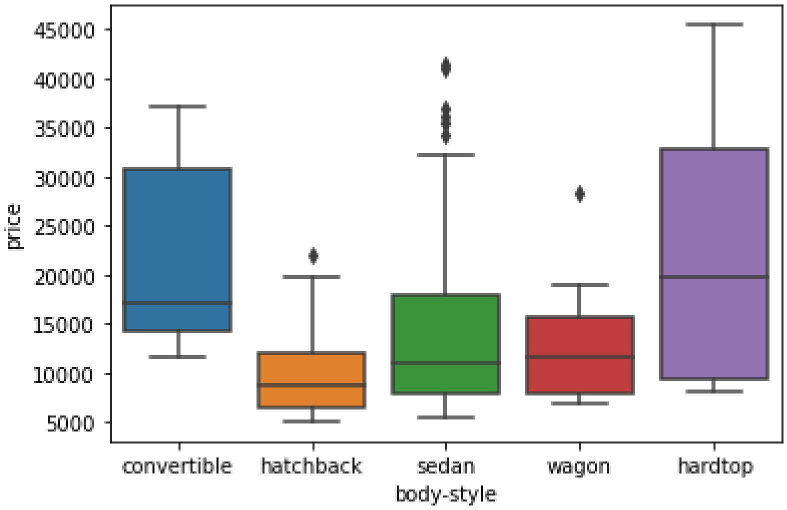
```
The Pearson Correlation Coefficient is 0.809574567003656  with a P-value of P = 6.36
9057428259557e-48
```

Out[30]:
```
(0.0, 53182.18587085162)
```



In [31]:
```
sns.boxplot(x="body-style", y="price", data=df)
```

Out[31]:
```
<AxesSubplot:xlabel='body-style', ylabel='price'>
```



In [33]:

```
df_group_one = df[['drive-wheels','body-style','price']]
# grouping results
df_group_one = df_group_one.groupby(['drive-wheels'],as_index=False).mean()
df_group_one
df_gptest = df[['drive-wheels','body-style','price']]
grouped_test1 = df_gptest.groupby(['drive-wheels','body-style'],as_index=False).mean
grouped_test1
grouped_pivot = grouped_test1.pivot(index='drive-wheels',columns='body-style')
grouped_pivot
```

Out[33]:

	price				
body-style	**convertible**	**hardtop**	**hatchback**	**sedan**	**wagon**
drive-wheels					
4wd	NaN	NaN	7603.000000	12647.333333	9095.750000
fwd	11595.0	8249.000000	8396.387755	9811.800000	9997.333333
rwd	23949.6	24202.714286	14337.777778	21711.833333	16994.222222

In [34]:

```
grouped_pivot = grouped_pivot.fillna(0) #fill missing values with 0
grouped_pivot
```

Out[34]:

	price				
body-style	**convertible**	**hardtop**	**hatchback**	**sedan**	**wagon**
drive-wheels					
4wd	0.0	0.000000	7603.000000	12647.333333	9095.750000
fwd	11595.0	8249.000000	8396.387755	9811.800000	9997.333333
rwd	23949.6	24202.714286	14337.777778	21711.833333	16994.222222

In [35]:

```
df_gptest2 = df[['body-style','price']]
grouped_test_bodystyle = df_gptest2.groupby(['body-style'],as_index= False).mean()
grouped_test_bodystyle
```

Out[35]:

	body-style	price
0	convertible	21890.500000
1	hardtop	22208.500000

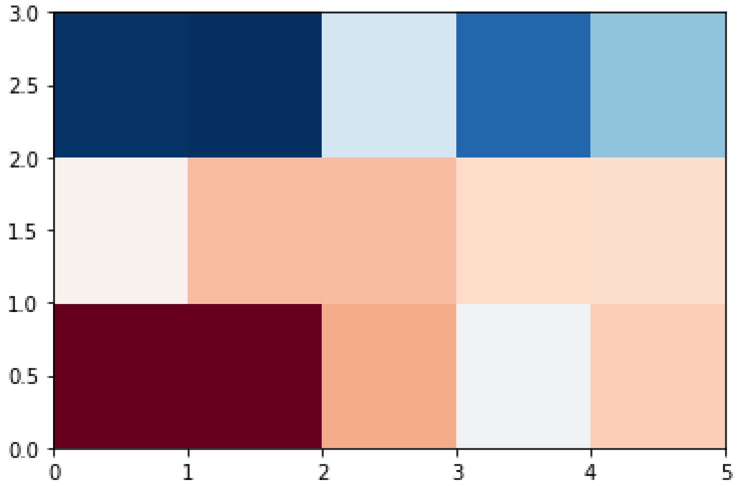
	body-style	price
2	hatchback	9957.441176
3	sedan	14459.755319
4	wagon	12371.960000

In [36]:

```
import matplotlib.pyplot as plt
#use the grouped results
plt.pcolor(grouped_pivot, cmap='RdBu')
```

Out[36]:

```
<matplotlib.collections.PolyCollection at 0xb9196d0>
```


3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
1
2
3
4
5