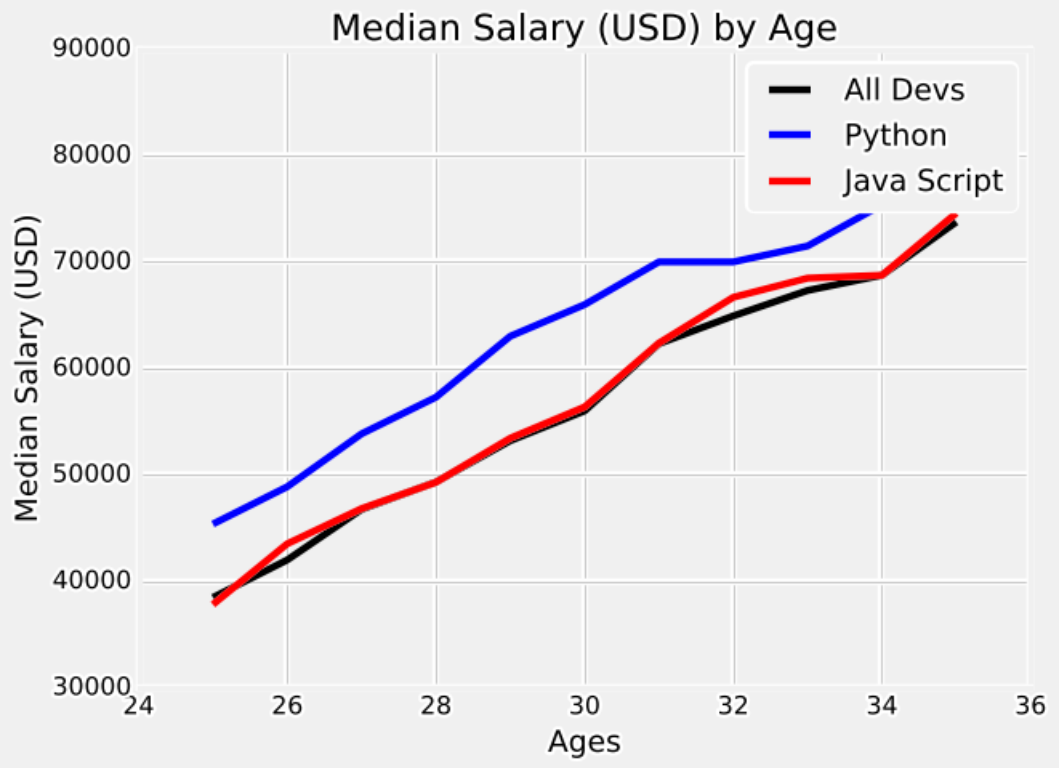
In [5]:

```
from matplotlib import pyplot as plt
import numpy as np
import pandas as pd
from collections import Counter
import csv
```

In [49]:

```
ages = [25,26,27,28,29,30,31,32,33,34,35]   # Values on X-axis
dev_y = [38469,42000,46752,49320,53200,56000,62316,64928,67317,68748,73752] # Values on Y-axis

plt.plot(ages,dev_y, color='k',label='All Devs', ) # Values of all developers

py_dev_y = [45372,48876,53850,57287,63016,65988,70003,70000,71496,75370,81640] # Salaries of python
developers - Y axis

plt.plot(ages,py_dev_y, color='b', label='Python') # Values of python developers

js_dev_y = [37810,43515,46823,49293,53437,56373,62375,66674,68475,68746,74583]

plt.plot(ages,js_dev_y, color='r', label='Java Script') # values of Java Script developers

plt.xlabel('Ages')
plt.ylabel('Median Salary (USD)')
plt.title('Median Salary (USD) by Age')
plt.legend()
#plt.savefig('plot.png') # Save plot image automatically with the mentioned extension
#plt.grid(True)
#plt.tight_layout()
plt.style.use('fivethirtyeight')
#plt.xkcd() # Another style of plot. Nice one have a check
```



In [28]:

```
print(plt.style.available) # to see all the styles which are available
```

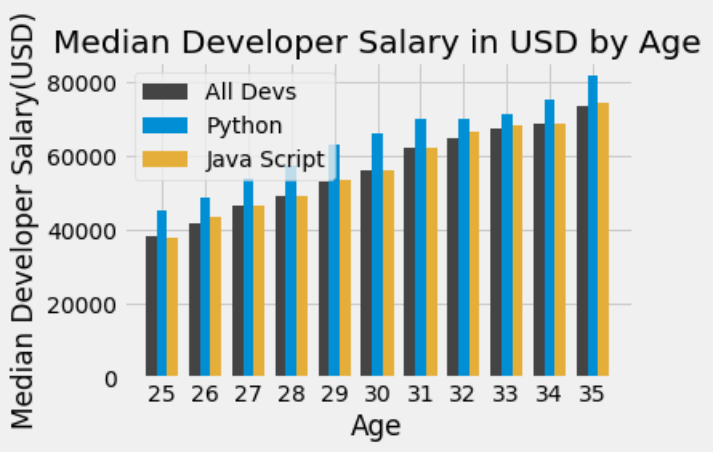
```
['bmh', 'classic', 'dark_background', 'fast', 'fivethirtyeight', 'ggplot', 'grayscale', 'seaborn-b
right', 'seaborn-colorblind', 'seaborn-dark-palette', 'seaborn-dark', 'seaborn-darkgrid',
'seaborn-deep', 'seaborn-muted', 'seaborn-notebook', 'seaborn-paper', 'seaborn-pastel', 'seaborn-p
oster', 'seaborn-talk', 'seaborn-ticks', 'seaborn-white', 'seaborn-whitegrid', 'seaborn',
```

'Solarize_Light2', 'tableau-colorblind10', '_classic_test']

Bar Charts

In [27]:

```
# In bar charts we use plt.bar instead of plt.plot
plt.style.use('fivethirtyeight')

ages = [25,26,27,28,29,30,31,32,33,34,35]   # Values on X-axis

x_indexes = np.arange(len(ages))

width = 0.25
#width1 = 0.3 provide gap between individual bar

dev_y = [38469,42000,46752,49320,53200,56000,62316,64928,67317,68748,73752] # Values on Y-axis

plt.bar(x_indexes-width,dev_y, width = width,color='#444444',label='All Devs')

py_dev_y = [45372,48876,53850,57287,63016,65988,70003,70000,71496,75370,81640] # Salaries of python
developers - Y axis

plt.bar(x_indexes,py_dev_y, width = width, color='#008fd5', label='Python') # Values of python deve
lopers

js_dev_y = [37810,43515,46823,49293,53437,56373,62375,66674,68475,68746,74583]

plt.bar(x_indexes + width,js_dev_y, width = width, color='#e5ae38', label='Java Script') # values o
f Java Script developers


plt.legend(loc='upper left')
plt.xticks(ticks=x_indexes, labels=ages) # To have age value and label instead of having index
number
plt.xlabel('Age')
plt.ylabel('Median Developer Salary(USD)')
plt.title('Median Developer Salary in USD by Age')
plt.tight_layout()
```



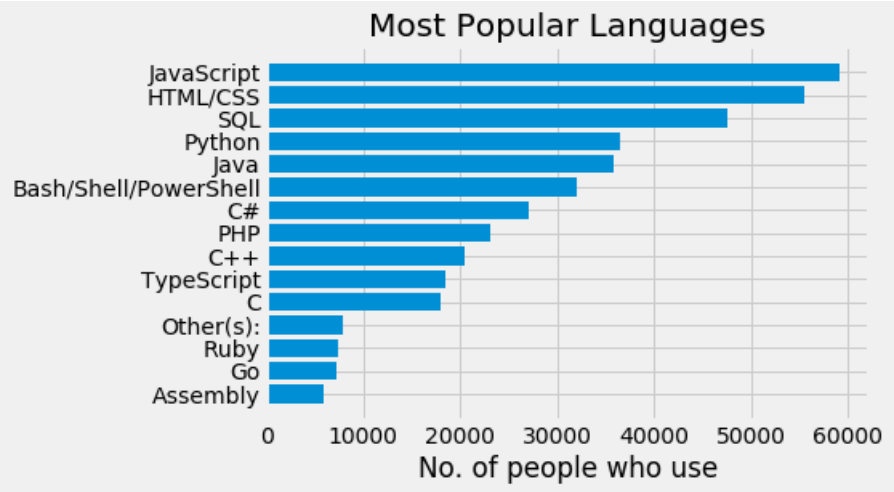
In [28]:

```
# Loading a data from csv file and working on it
```

In [93]:

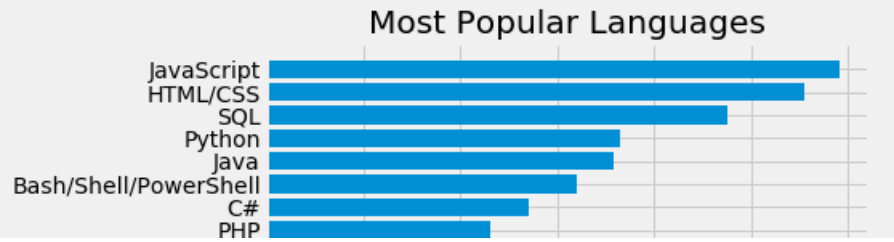
```
with open('E:/Data Science/data.csv') as csv_file:   # Using python method

    csv_reader = csv.DictReader(csv_file)

    language_counter = Counter()

    for row in csv_reader:
        language_counter.update(row['LanguagesWorkedWith'].split(';'))
```

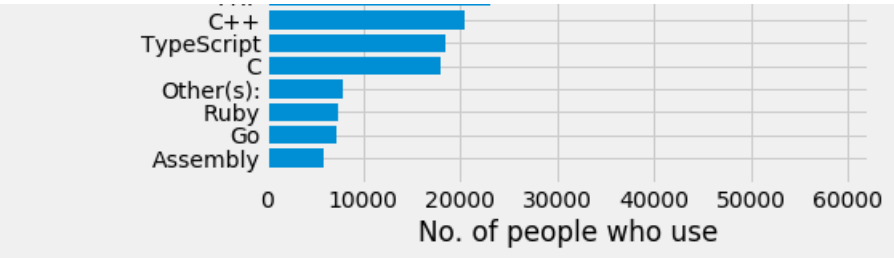
```
languages = []
popularity = []

for item in language_counter.most_common(15): # to print most common languages
    languages.append(item[0])
    popularity.append(item[1])

languages.reverse()    # to get higher values at the top
popularity.reverse()

plt.barh(languages,popularity)

#plt.ylabel('Language')
plt.xlabel('No. of people who use')
plt.title('Most Popular Languages')
plt.style.use('fivethirtyeight')
```



In [100]:

```
data = pd.read_csv('E:/Data Science/data.csv')    # Using Pandas method

ids = data['Responder_id']
lang_responses = data['LanguagesWorkedWith']

language_counter = Counter()

for response in lang_responses:
    language_counter.update(response.split(';'))

languages = []
popularity = []

for item in language_counter.most_common(15): # to print most common languages
    languages.append(item[0])
    popularity.append(item[1])

languages.reverse()    # to get higher values at the top
popularity.reverse()

plt.barh(languages,popularity)

#plt.ylabel('Language')
plt.xlabel('No. of people who use')
plt.title('Most Popular Languages')
plt.style.use('fivethirtyeight')
```





In [6]:

```
border = pd.read_csv('E:/Data Science/Kaggle_Data/border-crossing-entry-
data/Border_Crossing_Entry_Data.csv')
border.head()
```

Out[6]:

	Port Name	State	Port Code	Border	Date	Measure	Value	Location
0	Calexico East	California	2507	US-Mexico Border	03/01/2019 12:00:00 AM	Trucks	34447	POINT (-115.48433000000001 32.67524)
1	Van Buren	Maine	108	US-Canada Border	03/01/2019 12:00:00 AM	Rail Containers Full	428	POINT (-67.94271 47.16207)
2	Otay Mesa	California	2506	US-Mexico Border	03/01/2019 12:00:00 AM	Trucks	81217	POINT (-117.05333 32.57333)
3	Nogales	Arizona	2604	US-Mexico Border	03/01/2019 12:00:00 AM	Trains	62	POINT (-110.93361 31.340279999999996)
4	Trout River	New York	715	US-Canada Border	03/01/2019 12:00:00 AM	Personal Vehicle Passengers	16377	POINT (-73.44253 44.990010000000005)

In [7]:

```
sum = border.groupby(['State'], as_index=False)['Value'].sum()
sum.head()
```

Out[7]:

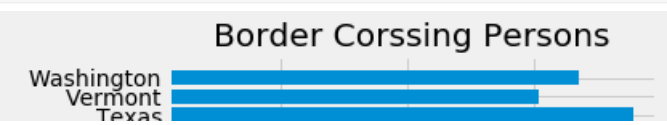
	State	Value
0	Alaska	13920348
1	Arizona	888075092
2	California	2499521716
3	Idaho	20998428
4	Maine	231249782

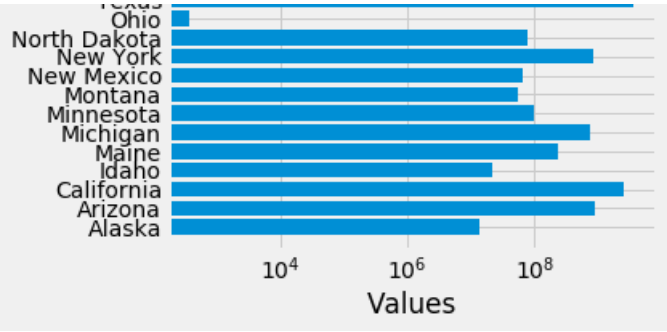
In [13]:

```
states = sum['State']
values = sum['Value']
```

In [14]:

```
plt.barh(states,values)
plt.xlabel('Values')
plt.title('Border Corssing Persons')
plt.xscale('log')
plt.xscale('log')
#plt.xticks(ticks=x_indexes, labels=values)
plt.tight_layout()
#plt.style.use('fivethirtyeight')
```

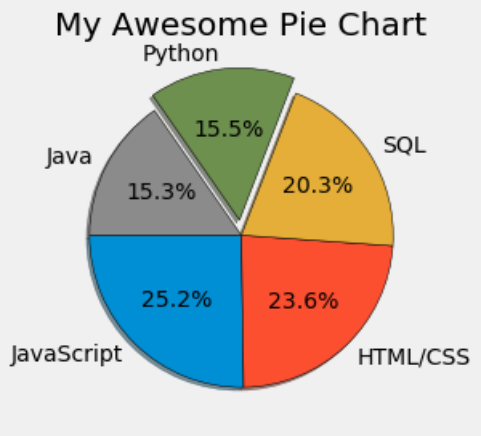




Pie Chart

In [232]:

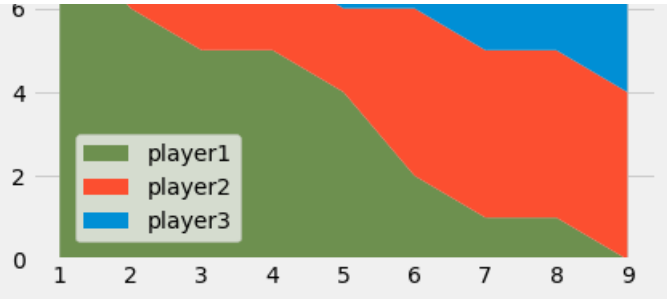
```
#slices = [120,80, 30,20]
#labels = ['Sixty', 'Forty', 'Extra1', 'Extra2']
slices = [59219, 55466, 47544, 36443, 35917]
labels = ['JavaScript', 'HTML/CSS', 'SQL', 'Python', 'Java']
explode = [0,0,0,0.1,0] # explode particular index where value is mentioned in the pie chart
#colors = ['#008fd5', '#fc4f30', '#e5ae37', '#6d904f']
plt.pie(slices, labels=labels, wedgeprops={'edgecolor':'black'}, explode=explode, shadow=True,start
angle=180, autopct='%1.1f%%')
plt.title('My Awesome Pie Chart')
plt.tight_layout()
plt.style.use('fivethirtyeight')
```



Stack Plots

In [246]:

```
minutes = [1,2,3,4,5,6,7,8,9]

player1 = [8, 6, 5, 5, 4, 2, 1, 1, 0]
player2 = [0, 1, 2, 2, 2, 4, 4, 4, 4]
player3 = [0, 1, 1, 1, 2, 2, 3, 3, 4]


#plt.pie([1,1,1],labels=["Player1", 'Player2', 'Player3'])
labels = ['player1', 'player2','player3']
colors = ['#6d904f', 'fc4f30', '008fd5']
plt.stackplot(minutes,player1,player2,player3, labels=labels, colors=colors)
plt.title('My Awesome Stack Plot')
plt.tight_layout()
plt.legend(loc = (0.07,0.05)) # to add labels
plt.style.use('fivethirtyeight')
```



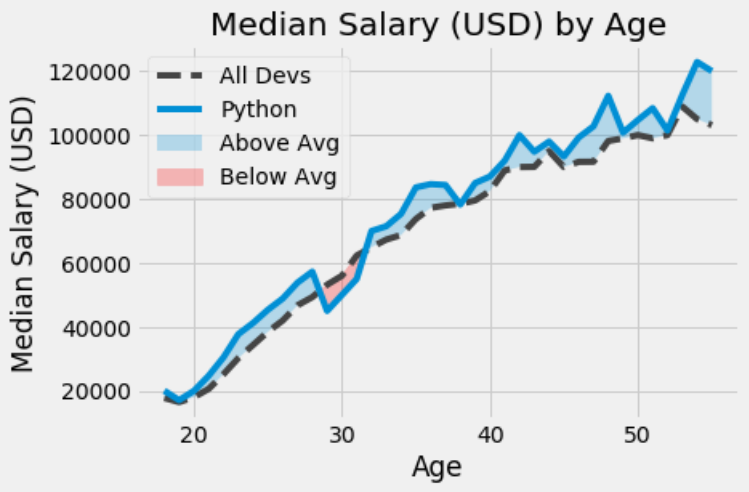
Filling area on line plots

In [261]:

```
data = pd.read_csv('E:/Data Science/data_sal.csv')
ages = data['Age']
dev_salaries = data['All_Devs']
py_salaries = data['Python']
js_salaries = data['JavaScript']

plt.plot(ages, dev_salaries, color='#444444', linestyle='--', label='All Devs')
plt.plot(ages, py_salaries, label='Python')

overall_median = 57287 # mean value of py_developer salary

plt.fill_between(ages,py_salaries,dev_salaries, where=(py_salaries > dev_salaries), interpolate=Tru
e, alpha=0.25, label='Above Avg')
plt.fill_between(ages,py_salaries,dev_salaries, where=(py_salaries <= dev_salaries),
interpolate=True, color = 'red',alpha=0.25, label='Below Avg')
plt.legend()
plt.title('Median Salary (USD) by Age')
plt.xlabel('Age')
plt.ylabel('Median Salary (USD)')
```

Out[261]:

```
Text(0, 0.5, 'Median Salary (USD)')
```

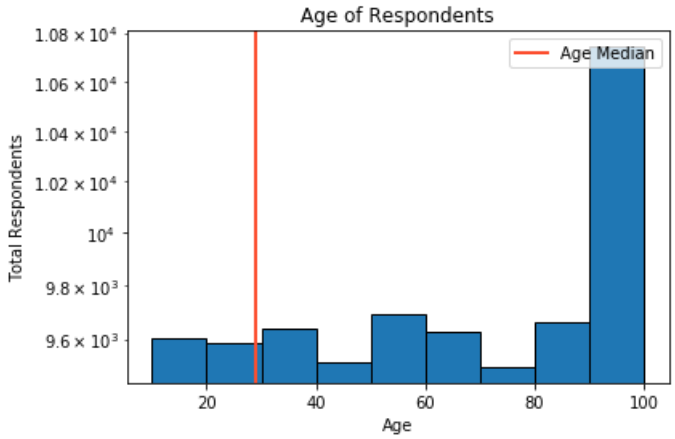


Histograms

In [3]:

```
#ages = [18,19,21,25,26,26,30,32,38,45,55] # for general understanding

#plt.hist(ages, bins=bins, edgecolor='black') - For general understanding

data = pd.read_csv('E:/Data Science/data.csv')
ids = data['Responder_id']
ages = data['Age']
bins = [10,20,30,40,50,60,70,80,90,100]
```

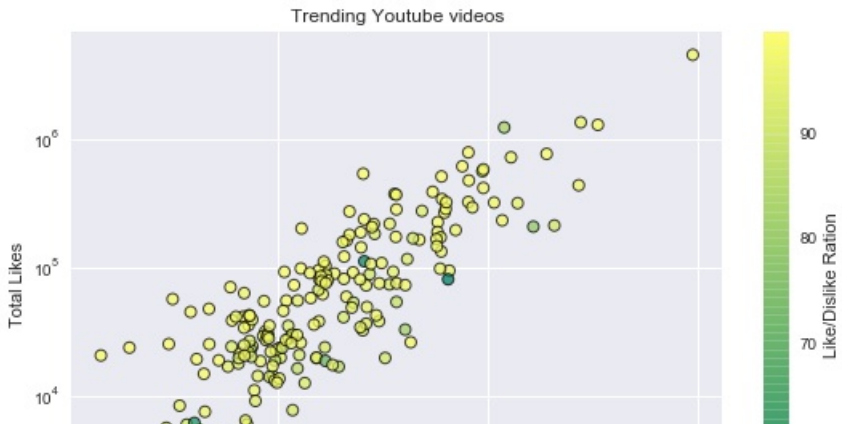
```
plt.hist(ages,bins=bins, edgecolor='black', log=True)

median_age = 29
color = '#fc4f30'
plt.axvline(median_age, color=color, label='Age Median', linewidth=2 )

plt.legend(loc='upper right')
plt.title('Age of Respondents')
plt.xlabel('Age')
plt.ylabel ('Total Respondents')
plt.tight_layout()
plt.style.use('fivethirtyeight')
```

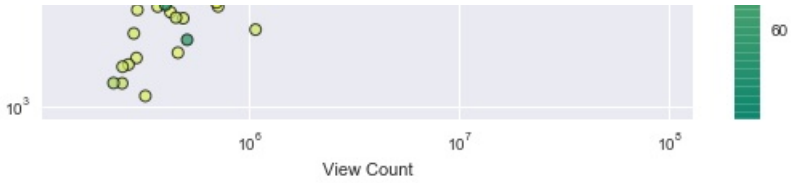


Scatter Plots

In [316]:

```
#x = [5,7,8,5,6,7,9,2,3,4,4,4,2,6,3,6,8,6,4,1]
#y = [7,4,3,9,1,3,2,5,2,4,8,7,1,6,4,9,7,7,5,1]
#colors = [7,5,9,7,5,7,2,5,3,7,1,2,8,1,9,2,5,6,7,5]
#sizes = [209,486,381,255,191,315,185,228,174,538,239,394,399,153,273,436,501,397,539]
data = pd.read_csv('E:/Data Science/youtube.csv')
view_count = data['view_count']
likes = data['likes']
ratio = data['ratio']

plt.scatter(view_count,likes, edgecolor='black', c=ratio, cmap='summer',alpha=0.75, linewidth=1) #
s for size, c for color

plt.xscale('log')
plt.yscale('log')
cbar = plt.colorbar()
cbar.set_label('Like/Dislike Ration')

plt.title('Trending Youtube videos')
plt.xlabel('View Count')
plt.ylabel('Total Likes')
plt.tight_layout()
plt.style.use('seaborn')
```





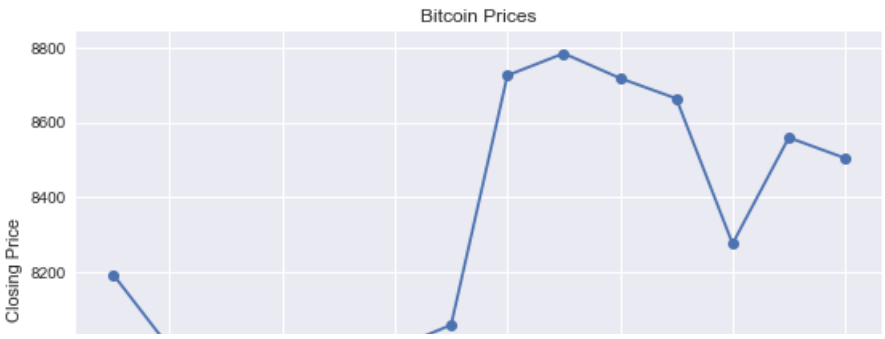
Plotting Time Series Data

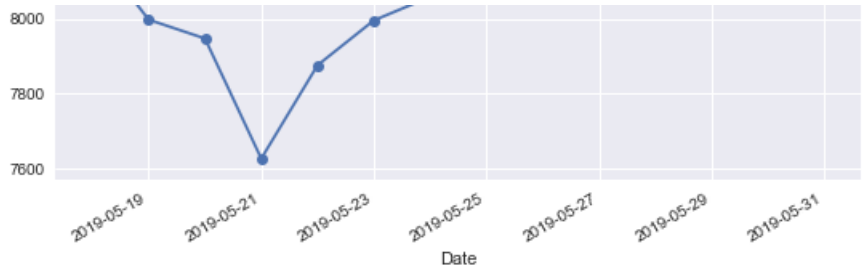
In [318]:

```
from datetime import datetime, timedelta
from matplotlib import dates as mpl_dates
```

In [337]:

```
#dates = [

 #       datetime(2019,5,24),
  #      datetime(2019,5,25),
   #     datetime(2019,5,26),
    #    datetime(2019,5,27),
     #   datetime(2019,5,28),
      #  datetime(2019,5,29),
       # datetime(2019,5,30)
#]


#y = [0,1,3,4,6,5,7]

#plt.plot_date(dates, y, linestyle='solid')

#plt.gcf().autofmt_xdate() # roatae the x-axis dates - gcf --> get current figure

#date_format = mpl_dates.DateFormatter('%b. %d %Y')

#plt.gca().xaxis.set_major_formatter(date_format)

data = pd.read_csv('E:/Data Science/datechart.csv')

data['Date'] = pd.to_datetime(data['Date']) # converting datetime in string format to datetime format

data.sort_values('Date', inplace=True)


price_date = data['Date']
price_close = data['Close']

plt.plot_date(price_date, price_close, linestyle='solid')

plt.gcf().autofmt_xdate()

plt.title('Bitcoin Prices')
plt.xlabel('Date')
plt.ylabel('Closing Price')
plt.tight_layout()

plt.style.use('seaborn')
```

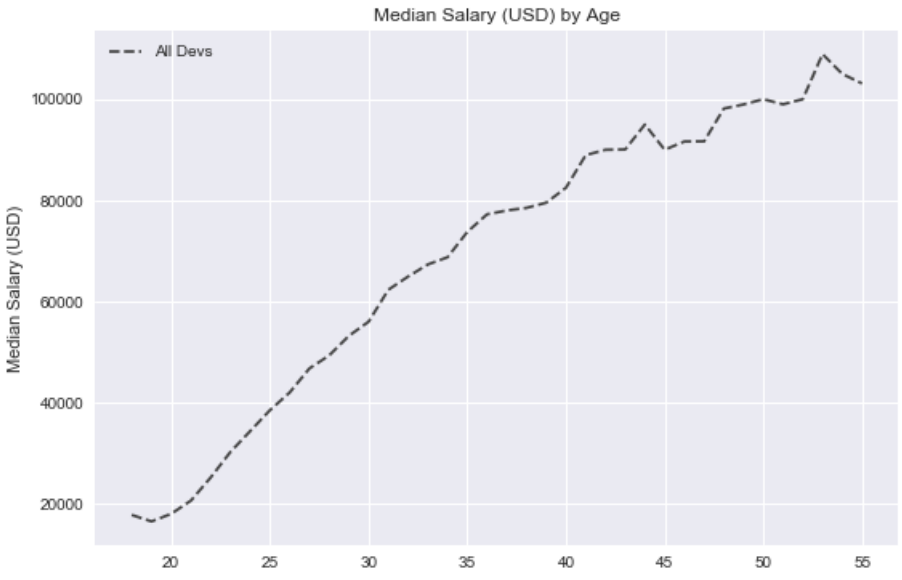


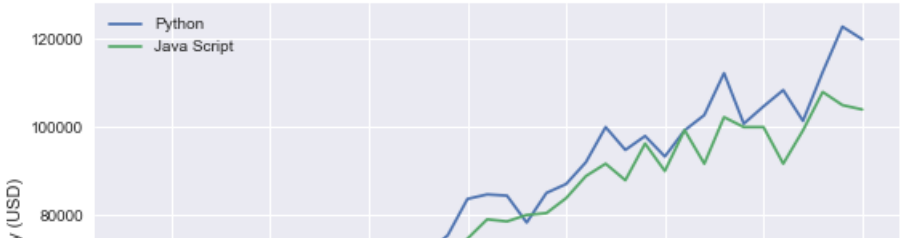


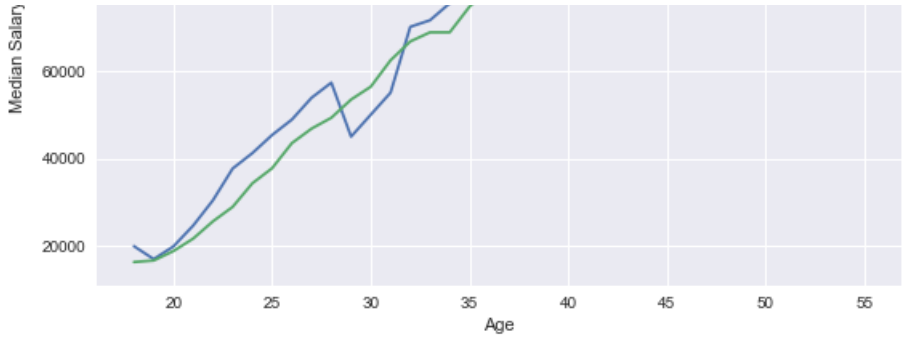
Subplots

In [356]:

```
data = pd.read_csv('E:/Data Science/data_sal.csv')
ages = data['Age']
dev_salaries = data['All_Devs']
py_salaries = data['Python']
js_salaries = data['JavaScript']

#fig, (ax1,ax2) = plt.subplots(nrows=2,ncols=1, sharex=True)

fig1, ax1 = plt.subplots()
fig2, ax2 = plt.subplots()


ax1.plot(ages, dev_salaries, color='#444444', linestyle='--', label='All Devs')
ax1.legend()
ax1.set_title('Median Salary (USD) by Age')

ax1.set_ylabel('Median Salary (USD)')

ax2.plot(ages, py_salaries, label='Python')
ax2.plot(ages,js_salaries, label='Java Script')

ax2.legend()

ax2.set_xlabel('Age')
ax2.set_ylabel('Median Salary (USD)')
plt.style.use('seaborn')
```







In [ ]: