import pandas as pd

import numpy as np import random import matplotlib.pyplot as plt

In [2]:
```
pwd
```

Out[2]:
```
'C:\\Users\\Hadi\\Desktop\\data science analystics\\week5-data visualization\\Week5-Visualization'
```

In [3]:
```
lj
```

In [104]:
```
data.shape
```

Out[104]:
```
(5656458, 6)
```

In [105]:
```
data.tail(10)
```

Out[105]:

	CountryName	CountryCode	IndicatorName	IndicatorCode	Year	Value
5656448	Zimbabwe	ZWE	Strength of legal rights index (0=weak to 12=s...	IC.LGL.CRED.XQ	2015	5.0
5656449	Zimbabwe	ZWE	Tax payments (number)	IC.TAX.PAYM	2015	49.0
5656450	Zimbabwe	ZWE	Time required to build a warehouse (days)	IC.WRH.DURS	2015	448.0
5656451	Zimbabwe	ZWE	Time required to enforce a contract (days)	IC.LGL.DURS	2015	410.0
5656452	Zimbabwe	ZWE	Time required to get electricity (days)	IC.ELC.TIME	2015	106.0
5656453	Zimbabwe	ZWE	Time required to register property (days)	IC.PRP.DURS	2015	36.0
5656454	Zimbabwe	ZWE	Time required to start a business (days)	IC.REG.DURS	2015	90.0
5656455	Zimbabwe	ZWE	Time to prepare and pay taxes (hours)	IC.TAX.DURS	2015	242.0
5656456	Zimbabwe	ZWE	Time to resolve insolvency (years)	IC.ISV.DURS	2015	3.3
5656457	Zimbabwe	ZWE	Total tax rate (% of commercial profits)	IC.TAX.TOTL.CP.ZS	2015	32.8

In [21]:
```
countries = data['CountryName'].unique().tolist()
len(countries)
```

Out[21]:
```
247
```

In [91]:
```
hist_country = 'IRN'
mask2 = data['CountryCode'].str.contains(hist_country).tolist()
len(mask2)
```

Out[91]:
```
5656458
```

In [22]:
```
countryCodes = data['CountryCode'].unique().tolist()
len(countryCodes)
```

Out[22]:
```
247
```

In [23]:
```
indicators = data['IndicatorName'].unique().tolist()
len(indicators)
```

Out[23]:
```
1344
```

In [24]:
```
years = data['Year'].unique().tolist()
len(years)
```

Out[24]:
```
56
```

In [25]:
```
print(min(years)," to ",max(years))
```

```
1960  to  2015
```

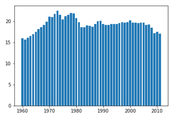
Matplotlib: Basic Plotting, Part 1

In [6]:
```
# select CO2 emissions for the United States
hist_indicator = 'CO2 emissions \(metric'
hist_country = 'USA'

mask1 = data['IndicatorName'].str.contains(hist_indicator)
mask2 = data['CountryCode'].str.contains(hist_country)

# stage is just those indicators matching the USA for country code and CO2 emissions over time.
stage = data[mask1 & mask2]
```

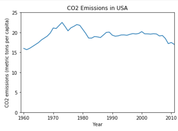
In [82]:
```
stage.head()
len(stage)
```

Out[82]:
```
52
```

In [83]:
```
# get the years
years = stage['Year'].values
# get the values
co2 = stage['Value'].values

# create
plt.bar(years,co2)
plt.show()
```



In [84]:
```
# switch to a line plot
plt.plot(stage['Year'].values, stage['Value'].values)

# Label the axes
plt.xlabel('Year')
plt.ylabel(stage['IndicatorName'].iloc[0])

#label the figure
plt.title('CO2 Emissions in USA')

# to make more honest, start they y axis at 0
plt.axis([1959, 2011,0,25])

plt.show()
```

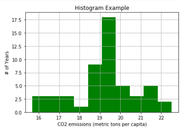


In [85]:
```
# If you want to just include those within one standard deviation fo the mean, you could do
  the following
# lower = stage['Value'].mean() - stage['Value'].std()
# upper = stage['Value'].mean() + stage['Value'].std()
# hist_data = [x for x in stage[:10000]['Value'] if x>lower and x<upper ]

# Otherwise, let's look at all the data
hist_data = stage['Value'].values
```

In [86]:
```
print(len(hist_data))
```

```
52
```

In [102]:
```
# the histogram of the data
plt.hist(hist_data,10, density=False, facecolor='green')

plt.xlabel(stage['IndicatorName'].iloc[0])
plt.ylabel('# of Years')
plt.title('Histogram Example')

plt.grid(True)

plt.show()
```



In [88]:
```
# select CO2 emissions for all countries in 2011
hist_indicator = 'CO2 emissions \(metric'
hist_year = 2011

mask1 = data['IndicatorName'].str.contains(hist_indicator)
mask2 = data['Year'].isin([hist_year])

# apply our mask
co2_2011 = data[mask1 & mask2]
co2_2011.head()
```

Out[88]:

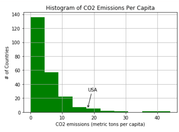
	CountryName	CountryCode	IndicatorName	IndicatorCode	Year	Value
5026275	Arab World	ARB	CO2 emissions (metric tons per capita)	EN.ATM.CO2E.PC	2011	4.724500
5026788	Caribbean small states	CSS	CO2 emissions (metric tons per capita)	EN.ATM.CO2E.PC	2011	9.692960
5027295	Central Europe and the Baltics	CEB	CO2 emissions (metric tons per capita)	EN.ATM.CO2E.PC	2011	6.911131
5027870	East Asia & Pacific (all income levels)	EAS	CO2 emissions (metric tons per capita)	EN.ATM.CO2E.PC	2011	5.859548
5028456	East Asia & Pacific (developing only)	EAP	CO2 emissions (metric tons per capita)	EN.ATM.CO2E.PC	2011	5.302499

In [89]:
```
print(len(co2_2011))
```

```
232
```

In [57]:
```
# let's plot a histogram of the emmissions per capita by country

# subplots returns a touple with the figure, axis attributes.
fig, ax = plt.subplots()

ax.annotate("USA",
            xy=(18, 5), xycoords='data',
            xytext=(18, 30),textcoords='data',
            arrowprops=dict(arrowstyle="->",
                            connectionstyle="arc3"),
            )

plt.hist(co2_2011['Value'], 10, normed=False, facecolor='green')

plt.xlabel(stage['IndicatorName'].iloc[0])
plt.ylabel('# of Countries')
plt.title('Histogram of CO2 Emissions Per Capita')

#plt.axis([10, 22, 0, 14])
plt.grid(True)

plt.show()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6521: MatplotlibDeprecationWarning:
The 'normed' kwarg was deprecated in Matplotlib 2.1 and will be removed in 3.1. Use 'density' instead.
  alternative="'density'", removal="3.1")
```



Matplotlib: Basic Plotting, Part 2

In [16]:
```
# select GDP Per capita emissions for the United States
hist_indicator = 'GDP per capita \(constant 2005'
hist_country = 'IRN'

mask1 = data['IndicatorName'].str.contains(hist_indicator)
mask2 = data['CountryCode'].str.contains(hist_country)

# stage is just those indicators matching the USA for country code and CO2 emissions over time.
gdp_stage = data[mask1 & mask2]

#plot gdp_stage vs stage
```

In [14]:
```
gdp_stage.head(2)
```

Out[14]:

	CountryName	CountryCode	IndicatorName	IndicatorCode	Year	Value
12003	Iran, Islamic Rep.	IRN	GDP per capita (constant 2005 US$)	NY.GDP.PCAP.KD	1960	1497.930245
36998	Iran, Islamic Rep.	IRN	GDP per capita (constant 2005 US$)	NY.GDP.PCAP.KD	1961	1611.393251

In [17]:
```
stage.head(2)
```

Out[17]:

	CountryName	CountryCode	IndicatorName	IndicatorCode	Year	Value
22232	United States	USA	CO2 emissions (metric tons per capita)	EN.ATM.CO2E.PC	1960	15.999779
48708	United States	USA	CO2 emissions (metric tons per capita)	EN.ATM.CO2E.PC	1961	15.651256

In [18]:
```
# switch to a line plot
plt.plot(gdp_stage['Year'].values, gdp_stage['Value'].values)

# Label the axes
plt.xlabel('Year')
plt.ylabel(gdp_stage['IndicatorName'].iloc[0])

#label the figure
plt.title('GDP Per Capita IRN')

# to make more honest, start they y axis at 0
#plt.axis([1959, 2011,0,25])

plt.show()
```