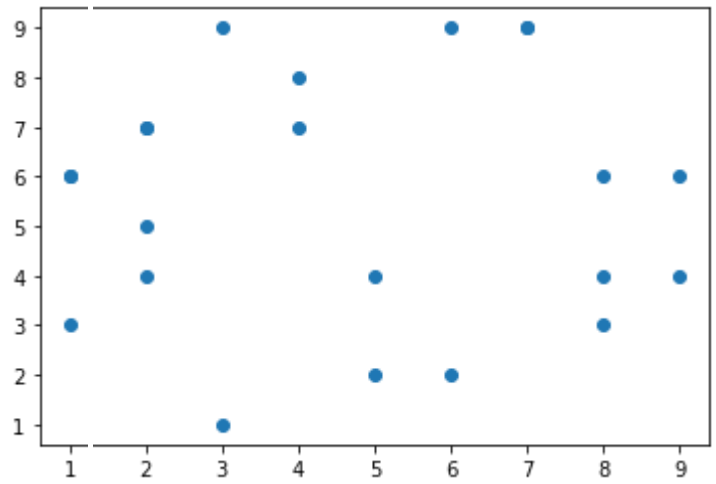
In [1]:

```
import matplotlib.pyplot as plt
#create a figure and axis
fig,ax=plt.subplots()
x=[2,4,6,8,9,2,7,2,6,1,8,4,5,9,1,2,3,7,5,8,1,3]
y=[7,8,2,4,6,4,9,5,9,3,6,7,2,4,6,7,1,9,4,3,6,9]
ax.scatter(x,y)
```

```
<matplotlib.collections.PathCollection at 0x19b45b38d30>
```




In [2]:

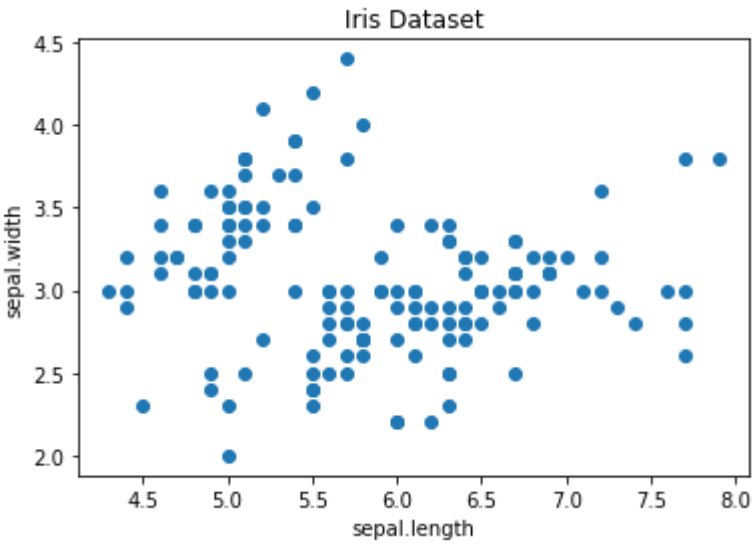
```
import pandas as pd
iris=pd.read_csv('iris - iris (1).csv')
print(iris.head())
```

```
   sepal.length  sepal.width  petal.length  petal.width variety
0           5.1          3.5           1.4          0.2  Setosa
1           4.9          3.0           1.4          0.2  Setosa
2           4.7          3.2           1.3          0.2  Setosa
3           4.6          3.1           1.5          0.2  Setosa
4           5.0          3.6           1.4          0.2  Setosa
```

In [3]:

```
import matplotlib.pyplot as plt
#Creating figure and axis
fig,ax=plt.subplots()
 #Scatter the sepal_Length against the special_width
ax.scatter(iris['sepal.length'],iris['sepal.width'])
#set a title and label
ax.set_title('Iris Dataset')
ax.set_xlabel('sepal.length')
ax.set_ylabel('sepal.width')
```

```
Text(0, 0.5, 'sepal.width')
```



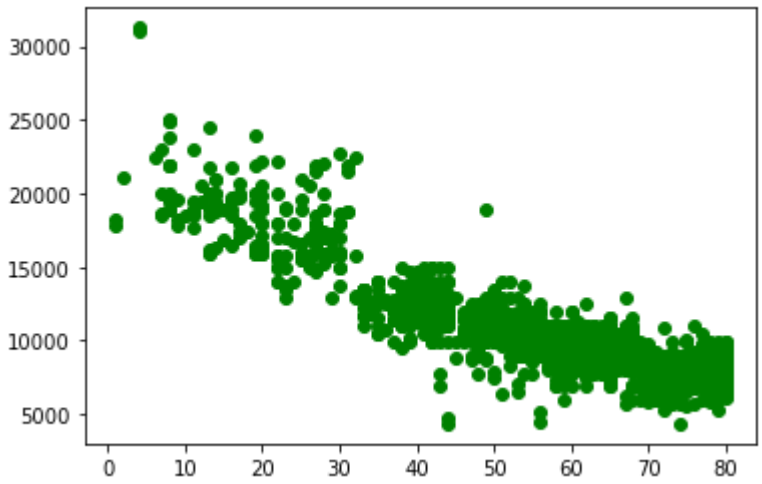
In [4]:

```
import pandas as pd
toyota= pd.read_csv("Toyota.csv")
toyota.head()
```

| | Unnamed: 0 | Price | Age | KM | FuelType | HP | MetColor | Automatic | CC | Doors | Weight |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 13500 | 23.0 | 46986 | Diesel | 90 | 1.0 | 0 | 2000 | three | 1165 |
| **1** | 1 | 13750 | 23.0 | 72937 | Diesel | 90 | 1.0 | 0 | 2000 | 3 | 1165 |
| **2** | 2 | 13950 | 24.0 | 41711 | Diesel | 90 | NaN | 0 | 2000 | 3 | 1165 |
| **3** | 3 | 14950 | 26.0 | 48000 | Diesel | 90 | 0.0 | 0 | 2000 | 3 | 1165 |
| **4** | 4 | 13750 | 30.0 | 38500 | Diesel | 90 | 0.0 | 0 | 2000 | 3 | 1170 |

In [5]:

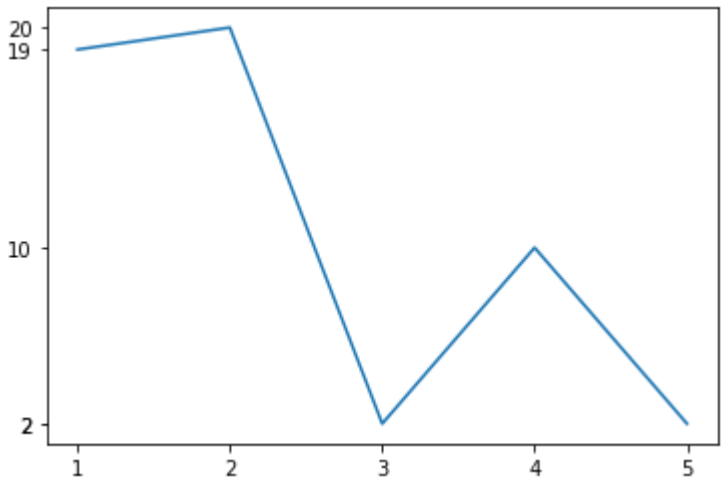
```
import matplotlib.pyplot as plt
plt.scatter(toyota["Age"],toyota["Price"],color="green")
plt.show()
```



In [6]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

x=range(1,6)
y=np.random.randint(1,26,5)
plt.plot(x,y)

plt.xticks(x)
plt.yticks(y)

```

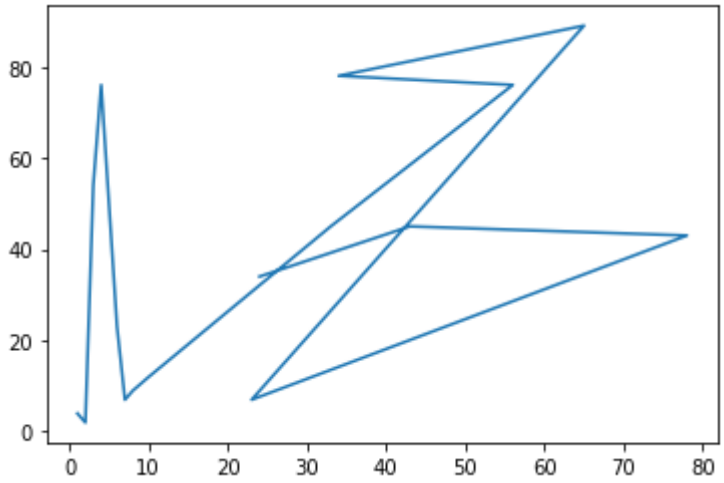
```
([<matplotlib.axis.YTick at 0x19b494b5b50>,
  <matplotlib.axis.YTick at 0x19b494b54f0>,
  <matplotlib.axis.YTick at 0x19b494ad3a0>,
  <matplotlib.axis.YTick at 0x19b494ec130>,
  <matplotlib.axis.YTick at 0x19b494ec7c0>],
 [Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, '')])
```



In [7]:

```
import matplotlib.pyplot as plt
#create a figure and axis
fig,ax=plt.subplots()
x=[1,2,3,4,6,7,8,10,33,56,34,65,23,78,43,24]
y=[4,2,54,76,23,7,9,12,45,76,78,89,7,43,45,34]
ax.plot(x,y)
```

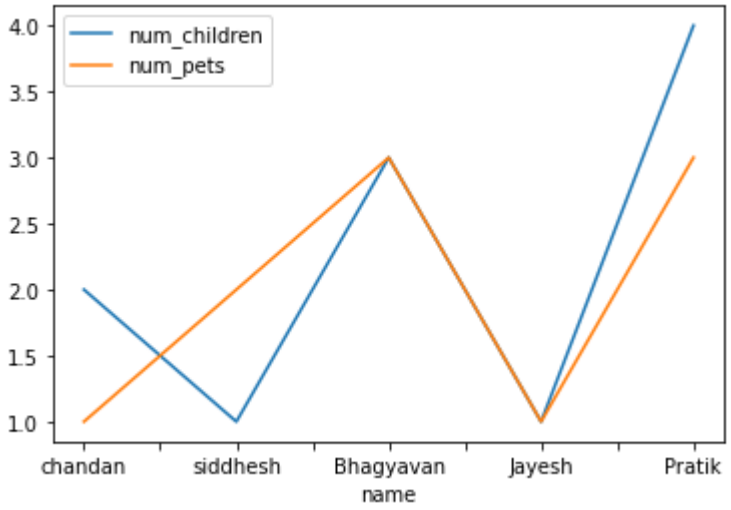
```
[<matplotlib.lines.Line2D at 0x19b49536610>]
```



In [8]:

```
import pandas as pd
df=pd.DataFrame({
    'name':['chandan','siddhesh','Bhagyavan','Jayesh','Pratik'],
    'gender':['M','M','M','M','M'],
    'age':[22,23,21,20,24],
    'state':['Maharashtra','Gujrat','UP','Kerla','Goa'],
    'num_children':[2,1,3,1,4],
    'num_pets':[1,2,3,1,3]
})
ax=plt.gca()
df.plot(kind='line',x='name',y='num_children',ax=ax)
df.plot(kind='line',x='name',y='num_pets',ax=ax)
#from pandas to plot on same figure
#gca stand for get current axis
```

```
<AxesSubplot:xlabel='name'>
```



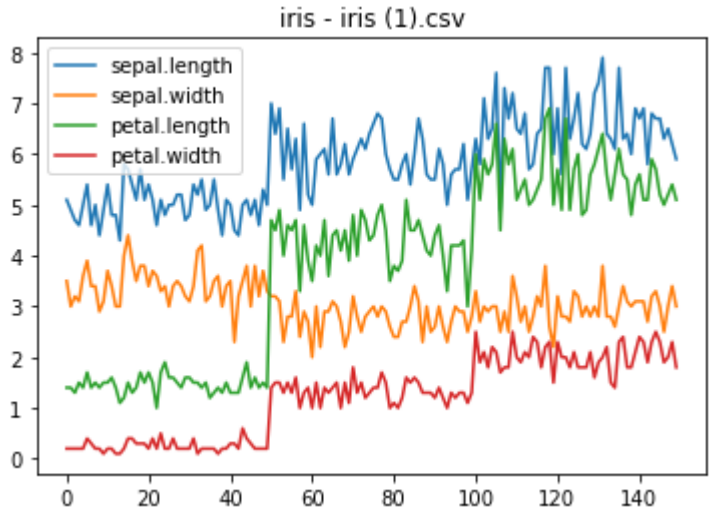

In [9]:

```
import pandas as pd
iris=pd.read_csv('iris - iris (1).csv')
print(iris.head())
```

```
   sepal.length  sepal.width  petal.length  petal.width variety
0           5.1          3.5           1.4          0.2  Setosa
1           4.9          3.0           1.4          0.2  Setosa
2           4.7          3.2           1.3          0.2  Setosa
3           4.6          3.1           1.5          0.2  Setosa
4           5.0          3.6           1.4          0.2  Setosa
```

In [10]:

```
#get column to plot
columns=iris.columns.drop(['variety'])
#create x data
x_data=range(0,iris.shape[0])
#create a figure and axis
fig,ax = plt.subplots()
#plot each columns
for column in columns:
    ax.plot(x_data,iris[column],label=column)
    #set title and legend
    ax.set_title('iris - iris (1).csv')
    ax.legend()
    #legend=info box
```

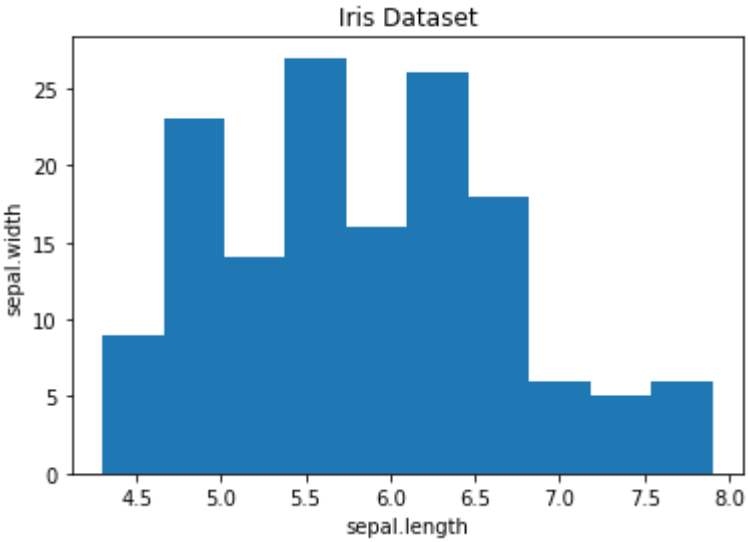



# Histogram

In [11]:

```
#create a figure and axis
fig,ax=plt.subplots()
#plot histogram
ax.hist(iris['sepal.length'])
#set title and label
ax.set_title('Iris Dataset')
ax.set_xlabel('sepal.length')
ax.set_ylabel('sepal.width')
```

```
Text(0, 0.5, 'sepal.width')
```



In [12]:

```
wine_review= pd.read_csv('winemag-data-130k-v2.csv')
wine_review
#Bar chart
```

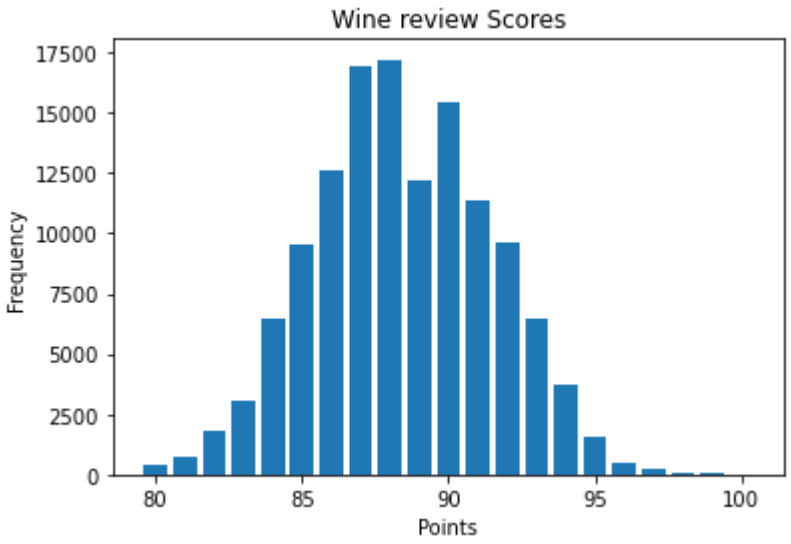
| | Unnamed: 0 | country | description | designation | points | price | province | region_1 | region_2 | t |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | Italy | Aromas include tropical fruit, broom, brimston... | Vulkà Bianco | 87 | NaN | Sicily & Sardinia | Etna | NaN | K |
| **1** | 1 | Portugal | This is ripe and fruity, a wine that is smooth... | Avidagos | 87 | 15.0 | Douro | NaN | NaN | F |
| **2** | 2 | US | Tart and snappy, the flavors of lime flesh and... | NaN | 87 | 14.0 | Oregon | Willamette Valley | Willamette Valley | F |
| **3** | 3 | US | Pineapple rind, lemon pith and orange blossom ... | Reserve Late Harvest | 87 | 13.0 | Michigan | Lake Michigan Shore | NaN | A F |
| **4** | 4 | US | Much like the regular bottling from 2012, this... | Vintner's Reserve Wild Child Block | 87 | 65.0 | Oregon | Willamette Valley | Willamette Valley | F |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | .. |
| **129966** | 129966 | Germany | Notes of honeysuckle and cantaloupe sweeten th... | Brauneberger Juffer-Sonnenuhr Spätlese | 90 | 28.0 | Mosel | NaN | NaN | A li |
| **129967** | 129967 | US | Citation is given as much as a decade of bottl... | NaN | 90 | 75.0 | Oregon | Oregon | Oregon Other | F |
| **129968** | 129968 | France | Well-drained gravel soil gives this wine its c... | Kritt | 90 | 30.0 | Alsace | Alsace | NaN | F |
| **129969** | 129969 | France | A dry style of Pinot Gris, this is crisp with ... | NaN | 90 | 32.0 | Alsace | Alsace | NaN | F |
| **129970** | 129970 | France | Big, rich and off-dry, this is powered by inte... | Lieu-dit Harth Cuvée Caroline | 90 | 21.0 | Alsace | Alsace | NaN | F |

129971 rows × 14 columns

In [13]:

```
#Bar chart
#create a figure and axis
fig,ax= plt.subplots()
#count the occurrence of each class
data = wine_review['points'].value_counts()
#get x and y data
points=data.index
frequency = data.values
#create bar chart
ax.bar(points,frequency)
#set title and Label
ax.set_title('Wine review Scores')
ax.set_xlabel('Points')
ax.set_ylabel('Frequency')
```

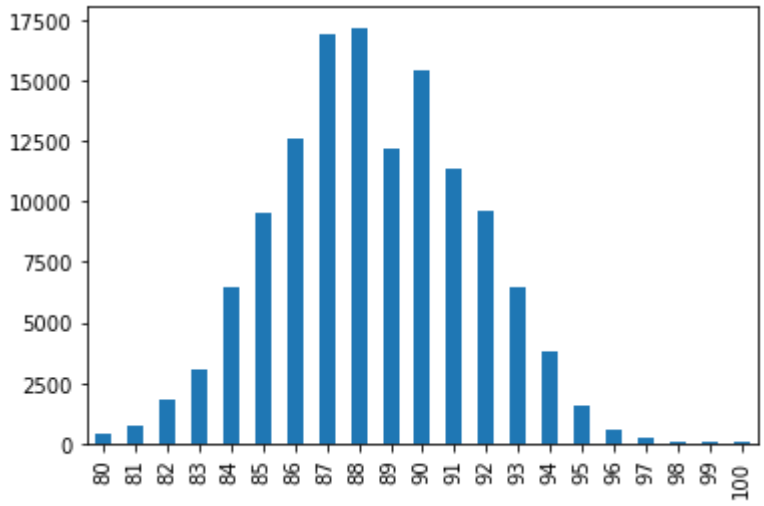
```
Text(0, 0.5, 'Frequency')
```



In [14]:

```
1 wine_review['points'].value_counts().sort_index().plot.bar()
```

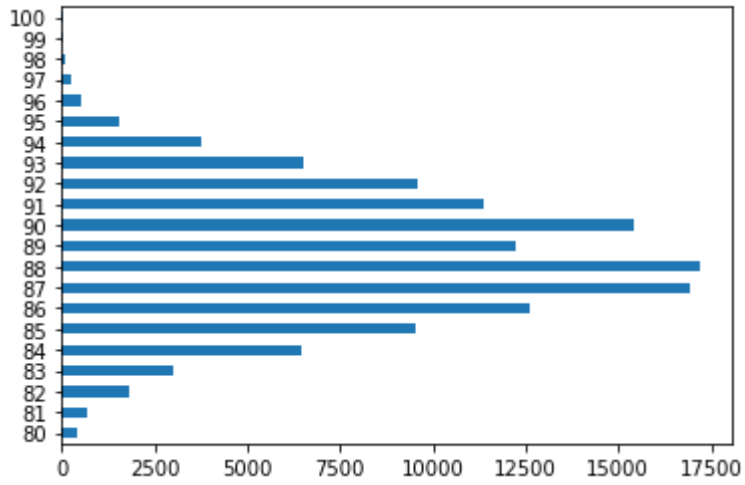
```
<AxesSubplot:>
```



In [15]:

```
wine_review['points'].value_counts().sort_index().plot.barh()
```

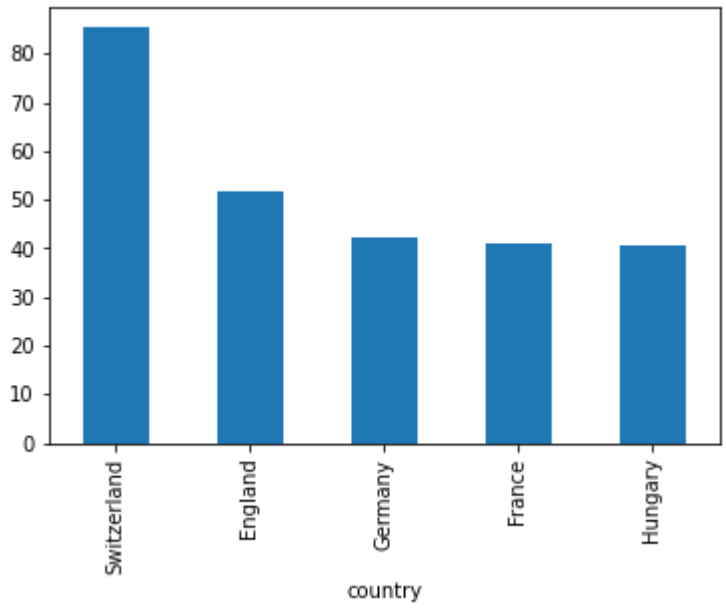
```
<AxesSubplot:>
```




In [16]:

```
#country on x-axis and frequency on y axis
wine_review.groupby("country").price.mean().sort_values(ascending=False)[:5].plot.
```

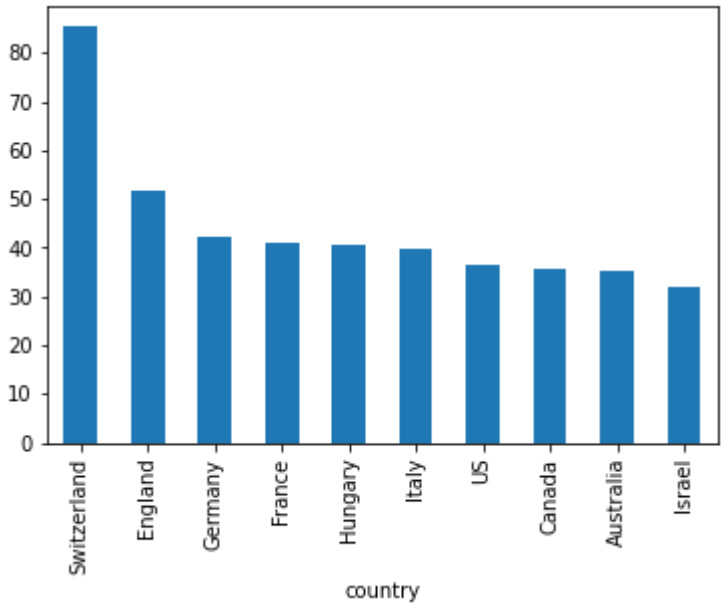
```
<AxesSubplot:xlabel='country'>
```



In [17]:

```
1 wine_review.groupby("country").price.mean().sort_values(ascending=False)[:10].plot
```

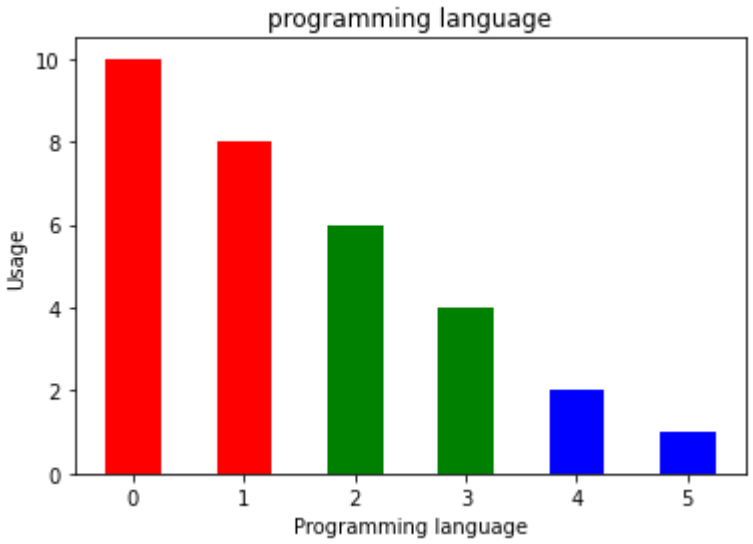
```
<AxesSubplot:xlabel='country'>
```



In [18]:

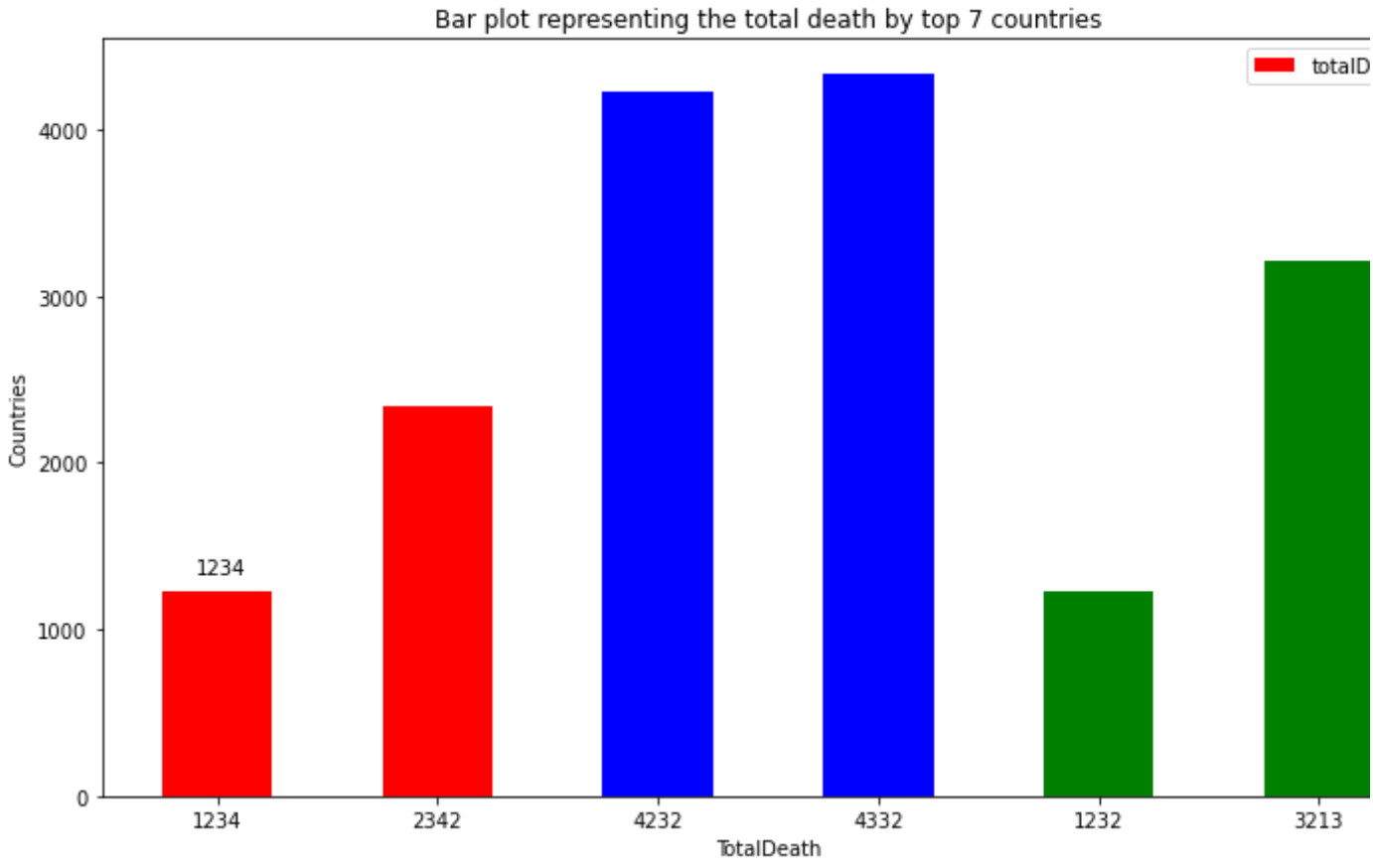
```
import numpy as np
import matplotlib.pyplot as plt
objects = ('Python','Java','c++','Perl','Scala','Lisp')
y_pos = np.arange(len(objects))
performance = [10,8,6,4,2,1]
#Bar chart
# X axis position as first parameter list,it can be floating point
# Y values as 2nd parameter list
plt.bar(y_pos,performance,width=0.5,align='center',alpha=1.0,color=('r','r','g','g
plt.ylabel('Usage')   #alpha for darkness of bar
plt.xlabel('Programming language')
plt.title('programming language')
```

```
Text(0.5, 1.0, 'programming language')
```



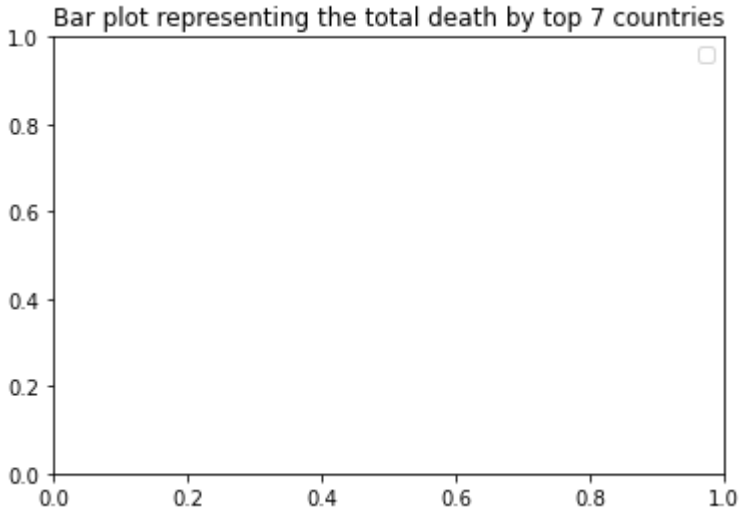
In [19]:

```
import matplotlib.pyplot as plt
#Declaring the figure or the plot (y,x) or  (width,height)
plt.figure(figsize = (12,7))
#Categorical data: country name
countries = ['USA','Brazil','Russia','Spain','UK','Canada']
totalDeath = [1234,2342,4232,4332,1232,3213]
plt.bar(countries,totalDeath, width=0.5, align='center',alpha=1.0,color=('r','r','
#This is the location for the annotated text
i=1.0
j=100

for i in range(len(countries)):
    plt.annotate(totalDeath[i],(-0.1 + i,totalDeath[i] + j))
    plt.legend(labels=['totalDeath'])
    plt.title("Bar plot representing the total death by top 7 countries")
    plt.xticks(countries,totalDeath)
    plt.xlabel('TotalDeath')
    plt.ylabel('Countries')
    plt.savefig('1Barplot.png')
    plt.show()
```



```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~\anaconda3\lib\site-packages\matplotlib\axis.py in convert_units(self, x)
   1504         try:
-> 1505             ret = self.converter.convert(x, self.units, self)
   1506         except Exception as e:

~\anaconda3\lib\site-packages\matplotlib\category.py in convert(value, unit, axis)
     48         if unit is None:
---> 49             raise ValueError(
     50                 'Missing category information for StrCategoryConverter; '

ValueError: Missing category information for StrCategoryConverter; this might be caused by unintendedly mixing
a

The above exception was the direct cause of the following exception:

ConversionError                           Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_2956/946091380.py in <module>
     14     plt.legend(labels=['totalDeath'])
     15     plt.title("Bar plot representing the total death by top 7 countries")
---> 16     plt.xticks(countries,totalDeath)
     17     plt.xlabel('TotalDeath')
     18     plt.ylabel('Countries')

~\anaconda3\lib\site-packages\matplotlib\pyplot.py in xticks(ticks, labels, **kwargs)
   1807                             "without setting 'ticks'")
   1808     else:
-> 1809         locs = ax.set_xticks(ticks)
   1810
   1811     if labels is None:

~\anaconda3\lib\site-packages\matplotlib\axes\_base.py in wrapper(self, *args, **kwargs)
     71
     72         def wrapper(self, *args, **kwargs):
---> 73             return get_method(self)(*args, **kwargs)
     74
     75         wrapper.__module__ = owner.__module__

~\anaconda3\lib\site-packages\matplotlib\axis.py in set_ticks(self, ticks, minor)
   1817         """
   1818         # XXX if the user changes units, the information will be lost here
-> 1819         ticks = self.convert_units(ticks)
```

```
   1820         if self is self.axes.xaxis:
   1821             shared = [

~\anaconda3\lib\site-packages\matplotlib\axis.py in convert_units(self, x)
   1505             ret = self.converter.convert(x, self.units, self)
   1506         except Exception as e:
-> 1507             raise munits.ConversionError('Failed to convert value(s) to axis '
   1508                                          f'units: {x!r}') from e
   1509         return ret

ConversionError: Failed to convert value(s) to axis units: ['USA', 'Brazil', 'Russia', 'Spain', 'UK', 'Canada']
```




In [ ]:

```
import matplotlib.pyplot as plt
#Declaring the figure or the plot (y,x) or (width
plt.figure(figsize=[14,10])
plt.barh(['USA','Brazil','Russia','Spain','UK','Canada'],[2000,500,3000,800,700,30
plt.barh(['Itly','Perl','Iran','Bangladesh'],[1500,3000,5000,4200], label= 'safe z
plt.legend()


```

In [ ]:

```
import pandas as pd
df=pd.DataFrame({
    'name':['chandan','siddhesh','Bhagyavan','Jayesh','sejal'],
    'gender':['M','F','M','M','F'],
    'age':[22,23,21,20,24],
    'state':['Maharashtra','Kerla','UP','Kerla','UP'],
    'num_children':[2,1,3,1,4],
    'num_pets':[1,2,3,1,3]
})
df.groupby(['state','gender']).size().unstack().plot(kind='bar',stacked=True)
```

In [ ]:

```
df.groupby(['gender','state']).size().unstack().plot(kind='bar',stacked=True)
```

In [ ]:

```
import matplotlib.pyplot as plt
#Declaring the figure or the plot (y,x) or  (width,height)
plt.figure(figsize = (12,7))
#Categorical data: country name
countries = ['USA','Brazil','Russia','Spain','UK','Canada']
totalCases = [21000,710887,476658,28877,28739,26596]
totalDeath = [15000,50000,271396,17000,7434,16000]
for i in range(len(countries)):
    plt.bar(countries[i],totalDeath[i], bottom=totalCases[i] - totalDeath[i], colo
    plt.bar(countries[i], totalCases[i] - totalDeath[i], color='red' )

```

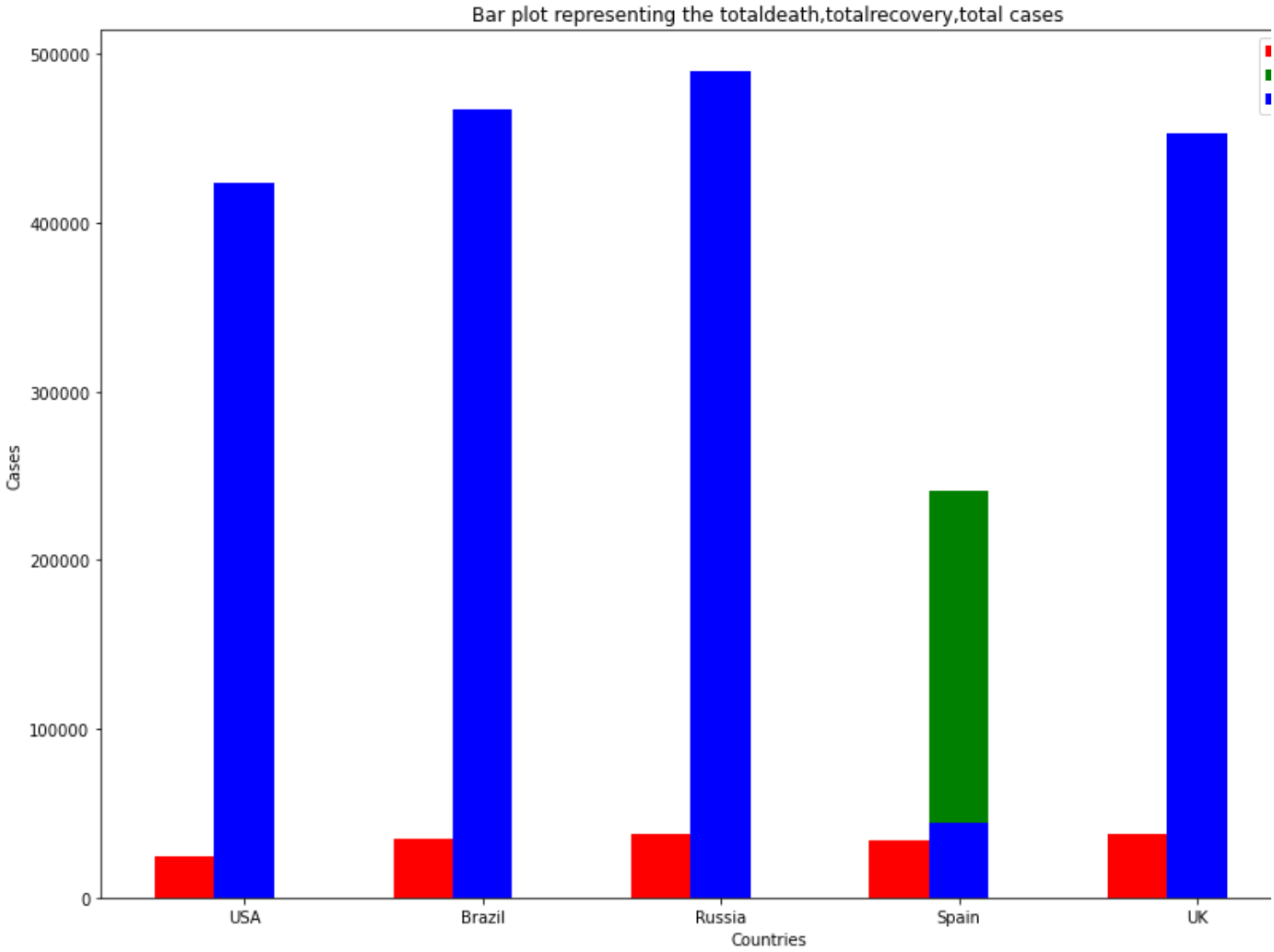
In [29]:

```
import pandas as pd
from matplotlib import pyplot as plt
Data = {'Country'=['USA','Canada','Germeny','UK','France'],'GDP_Per_Capita'=[45000
        'Income_Per_Capita':[4000,5000,7000,55000,60000]
       }
df= pd.DataFrame(Data)
df.plot(x='Country',y=['GDP_Per_Capita','Income_Per-Capita'],kind = 'bar')


```

```
  File "C:\Users\MSCIT\AppData\Local\Temp/ipykernel_2956/2244363001.py", line 3
    Data = {'Country'=['USA','Canada','Germeny','UK','France'],'GDP_Per_Capita'=[45000,42000,52000,49000,47000
                      ^
SyntaxError: invalid syntax
```

In [27]:

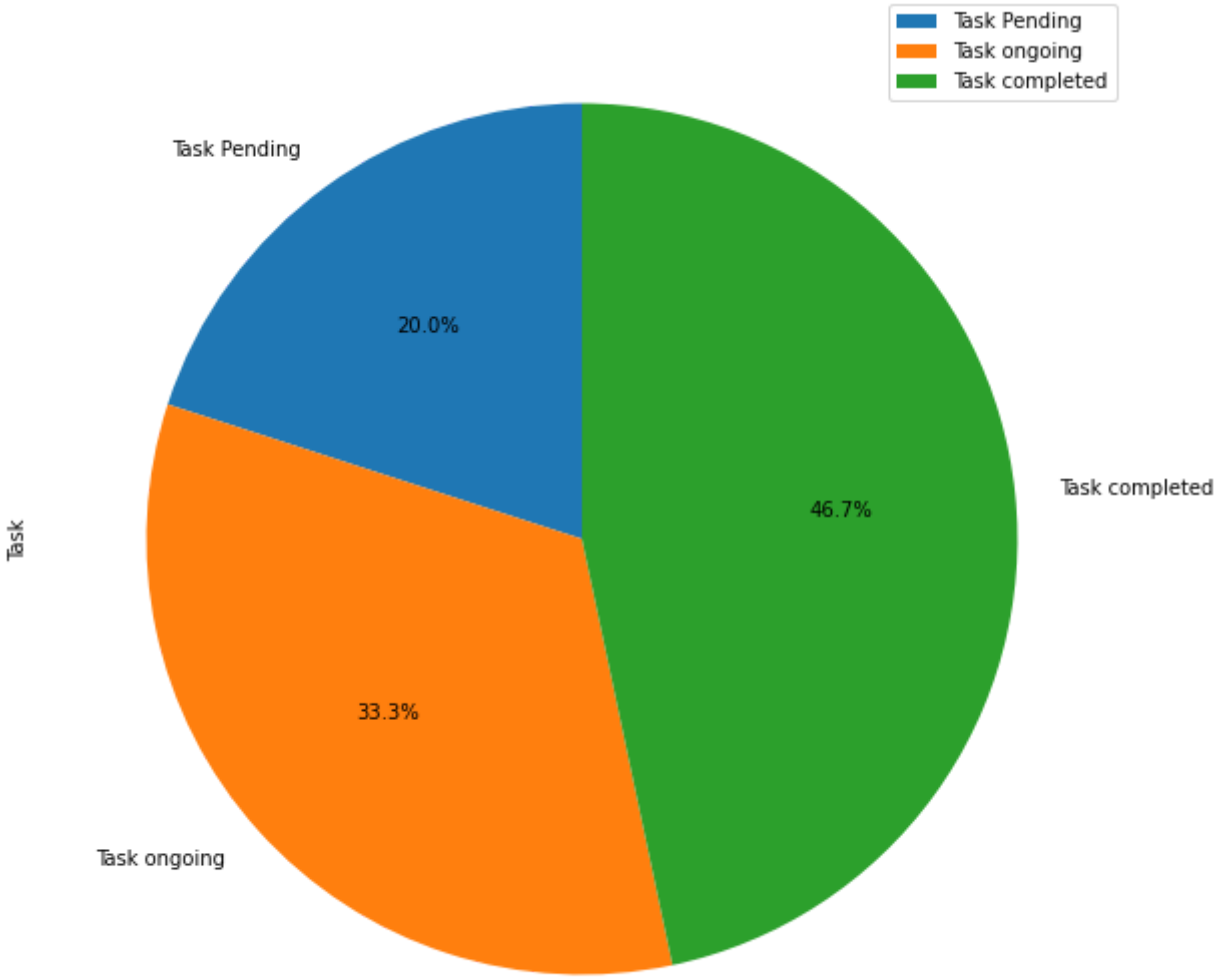
```
import numpy as np
import matplotlib.pyplot as plt
plt.figure(figsize=[15,10])
#Data to be plotted
totalDeath = [24523,34556,37892,34122,37887]
totalCases = [423784,466789,489789,44183,452876]
totalRecovery = [145236,245566,278962,241262,279865]
countries = ['USA','Brazil','Russia','Spain','UK','Canada']
x = np.arange(len(totalDeath))

plt.bar(x,totalDeath, color='red',width=0.25)
plt.bar(x +0.25,totalRecovery, color='g', width=0.25)
plt.bar(x +0.25,totalCases, color='b', width=0.25)

plt.legend(['TotalDeath','TotalRecovery','TotalCases'])
plt.xticks([i +0.25 for i in range(6)],countries)
plt.title('Bar plot representing the totaldeath,totalrecovery,total cases')
plt.xlabel('Countries')
plt.ylabel('Cases')
plt.savefig('4BAarplot.png')
plt.show()
```


Bar plot representing the totaldeath,totalrecovery,total cases
500000
400000
300000
200000
100000
0
Cases
USA
Brazil
Russia
Spain
UK
Countries

In [35]:

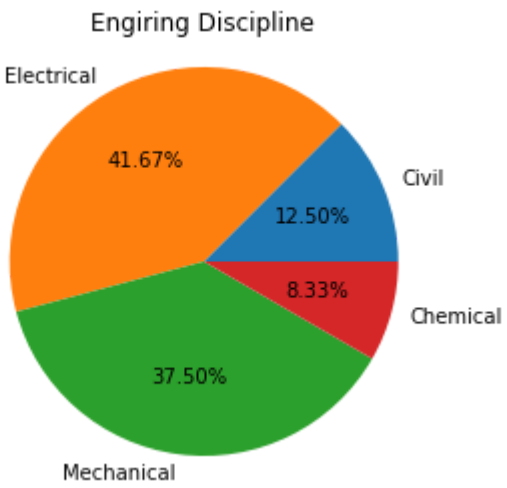
```
from pandas import DataFrame
import matplotlib.pyplot as plt
Data = {'Task': [300,500,700], 'Task Type' : ['Task Pending','Task ongoing','Task
df= DataFrame(Data)
df.set_index('Task Type', inplace=True)


df.plot.pie(y='Task', figsize=(10,10),autopct='%1.1f%%',startangle=90)
```

```
<AxesSubplot:ylabel='Task'>
```



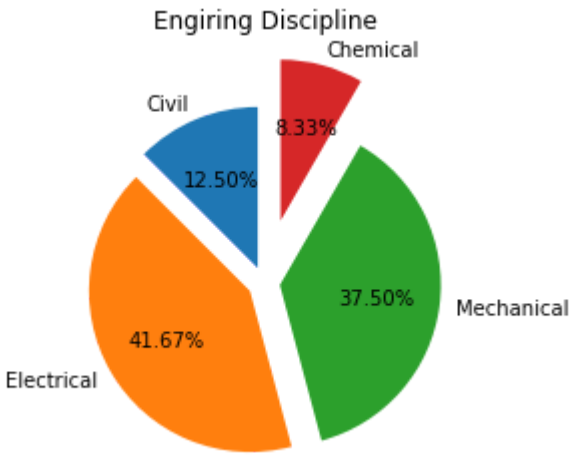
In [38]:

```
import numpy as np
import matplotlib.pyplot as plt
#if usinf a jupyter notebook include: %matplotlib inline

#pie chart where the slices will be ordered and plotted counter-clockwise:
labels = ['Civil','Electrical','Mechanical','Chemical']
sizes = [15,50,45,10]

fig,ax = plt.subplots()
ax.pie(sizes,labels=labels,autopct='%1.2f%%')
ax.axis('equal') #Equal accept
ax.set_title('Engiring Discipline')
plt.show()
import numpy as np
import matplotlib.pyplot as plt
#if usinf a jupyter notebook include: %matplotlib inline

#pie chart where the slices will be ordered and plotted counter-clockwise:
labels = ['Civil','Electrical','Mechanical','Chemical']
sizes = [15,50,45,10]
```



In [41]:

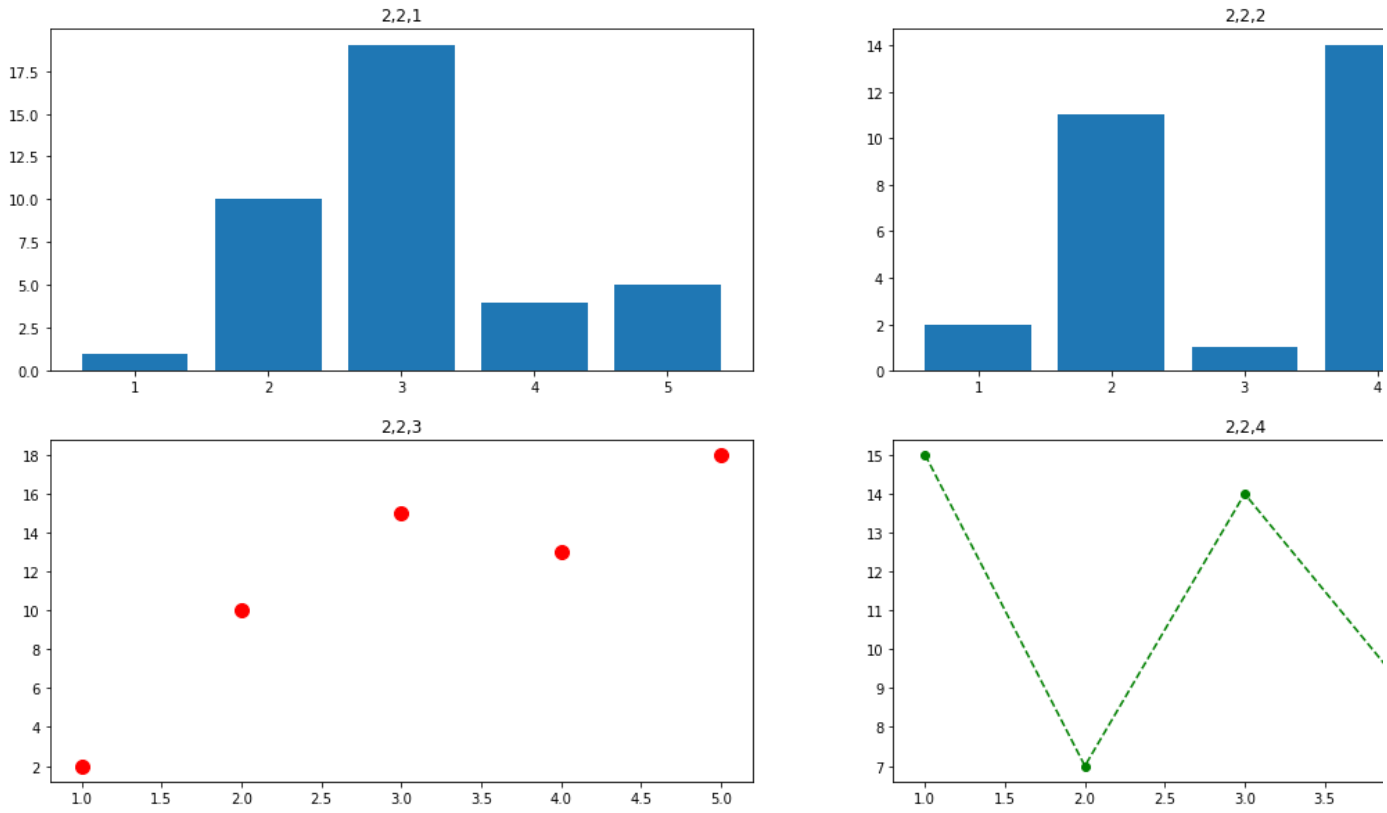
```
import numpy as np
import matplotlib.pyplot as plt
#if usinf a jupyter notebook include: %matplotlib inline

#pie chart where the slices will be ordered and plotted counter-clockwise:
labels = ['Civil','Electrical','Mechanical','Chemical']
sizes = [15,50,45,10]
explode = (0.1,0.1,0.1,0.4)
#explode out the 'Chemical' pie pieces by offsetting it a great6er amount
ig,ax = plt.subplots()
ax.pie(sizes,explode=explode,labels=labels,autopct='%1.2f%%',startangle=90)
ax.axis('equal') #Equal ascepts ensure the pie chart is eqal
ax.set_title('Engiring Discipline')
plt.show()
```



In [43]:

```
plt.figure(figsize=(20,10))
plt.subplot(2,2,1)
plt.bar(range(1,6), np.random.randint(1,20,5))
plt.title('2,2,1')
plt.subplot(2,2,2)
plt.bar(range(1,6), np.random.randint(1,20,5))
plt.title('2,2,2')
plt.subplot(2,2,3)
#s is size of dot
plt.scatter(range(1,6),np.random.randint(1,20,5),s=100,color='r')
plt.title('2,2,3')
plt.subplot(2,2,4)
plt.plot(range(1,6),np.random.randint(1,20,5),marker='o',color='g',linestyle='--')
plt.title('2,2,4')
```

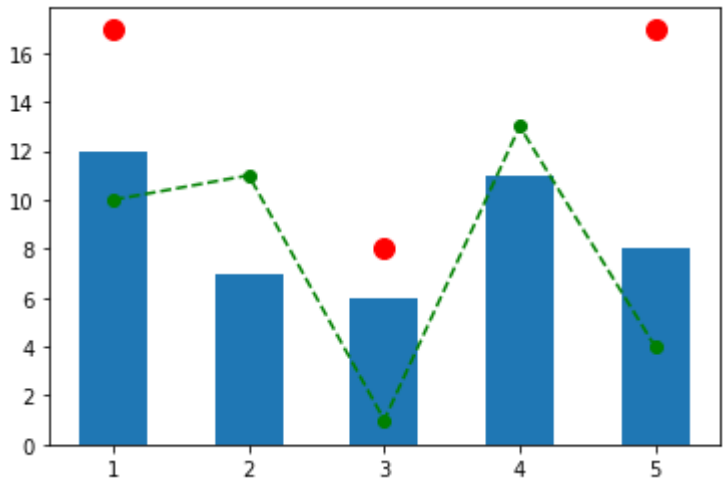
```
Text(0.5, 1.0, '2,2,4')
```



In [44]:

```
plt.bar(range(1,6), np.random.randint(1,20,5), width=0.5)
plt.scatter(range(1,6),np.random.randint(1,20,5),s=100,color='r')
plt.plot(range(1,6),np.random.randint(1,20,5),marker='o',color='g',linestyle='--')
```

```
[<matplotlib.lines.Line2D at 0x19b59695e80>]
```




In [ ]:

```
```