```
In [2]:  ▶  #import libraries
            import pandas as pd
            import matplotlib.pyplot as plt
```

```
In [3]:  ▶  #create dataframe
            df = pd.DataFrame({
                'name':['john','mary','peter','jeff','bill','lisa','jose'],
                'age':[23,78,22,19,45,33,20],
                'gender':['M','F','M','M','M','F','M'],
                'state':['california','dc','california','dc','california','texas','texas'],
                'num_children':[2,0,0,3,2,1,4],
                'num_pets':[5,1,0,5,2,2,3]
            })
```

```
In [4]:  ▶  #print dataframe
            df
```

Out[4]:

|	name	age	gender	state	num_children	num_pets
0	john	23	M	california	2	5
1	mary	78	F	dc	0	1
2	peter	22	M	california	0	0
3	jeff	19	M	dc	3	5
4	bill	45	M	california	2	2
5	lisa	33	F	texas	1	2
6	jose	20	M	texas	4	3

In [6]: ▶
```python
#scatterplot
# df.plot(kind='scatter',x='num_children',y='num_pets',color='blue')
# plt.show()


plt.plot( 'num_children', 'num_pets', data=df, linestyle='none', marker='o')
plt.legend()
plt.xlabel('num_children')
plt.ylabel('num_pets')
plt.show()
```
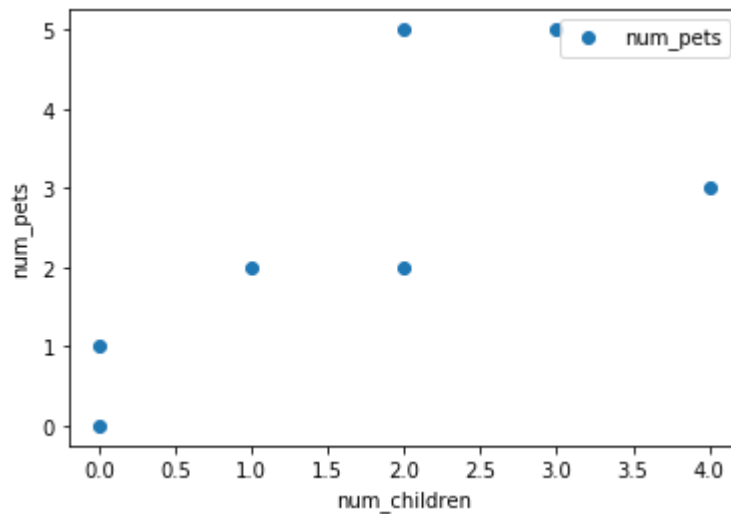

In [7]: ▶
```python
#bar chart
df.plot(kind='bar',x='name',y='age')
```

Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x158c56358d0>

In [9]:
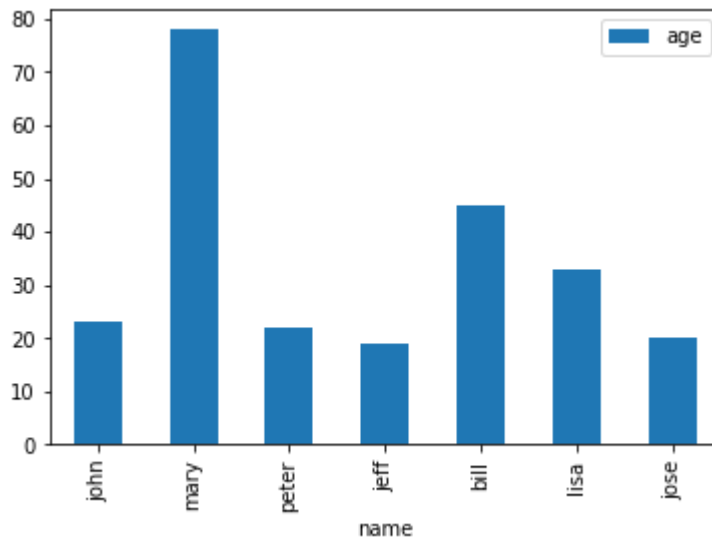
```python
# line chart with different legends

ax = plt.gca()

df.plot(kind='line',x='name',y='num_children',ax=ax)
df.plot(kind='line',x='name',y='num_pets', color='red', ax=ax)

plt.show()


plt.plot( 'name', 'num_children', data=df, marker='o', markerfacecolor='blue'
plt.plot( 'name', 'num_pets', data=df, marker='', color='olive', linewidth=2)
plt.legend()

plt.show()
```
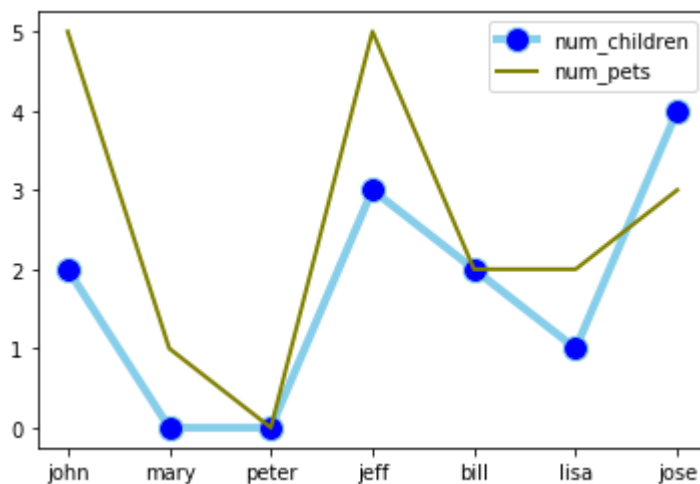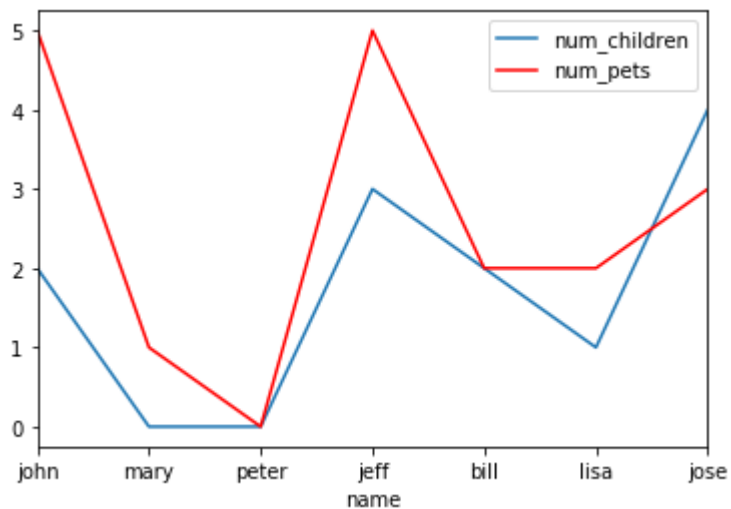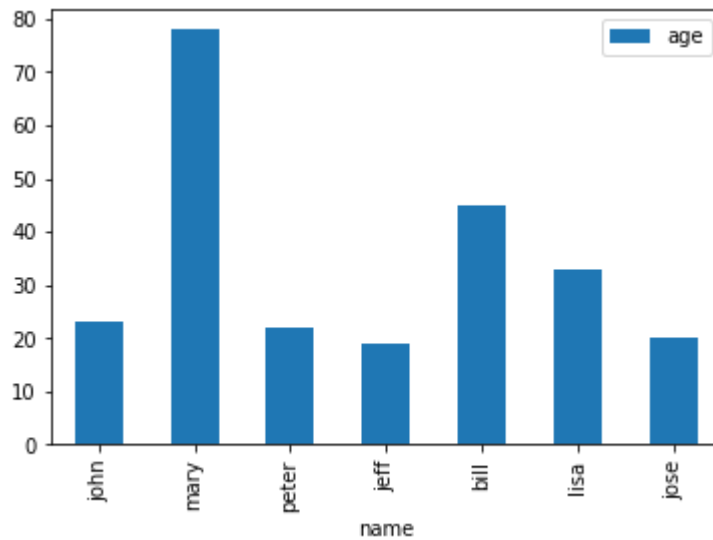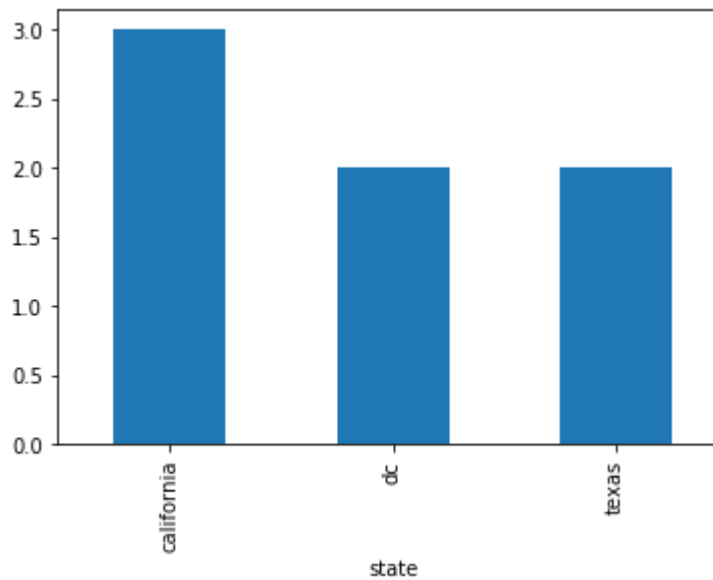
In [10]: ▶|
```python
#save the plot
df.plot(kind='bar',x='name',y='age')

# the plot gets saved to 'output.png'
plt.savefig('C:\dataset\output.png')
```


In [11]: ▶|
```python
#aggregate values and show
df.groupby('state')['name'].nunique().plot(kind='bar')
plt.show()
```

In [12]: ▶
```python
#stacked bar plots with group by
print(df)
df.groupby(['state','gender']).size().unstack().plot(kind='bar',stacked=True)


#some additional properties
plt.title('Count by State')
plt.xlabel('State')
plt.ylabel('Count')

labels = df['gender'].apply({'M':'Male', 'F':'Female'}.get).unique()

plt.legend(labels,loc='upper right')

plt.show()
```
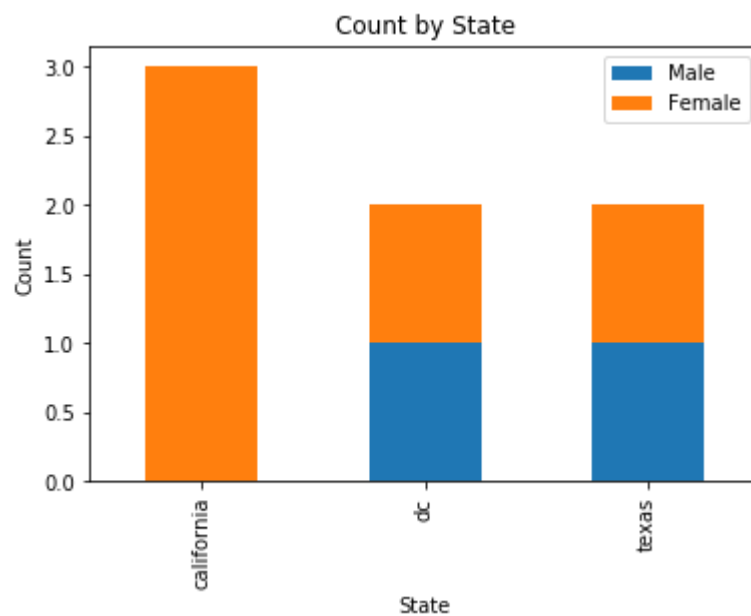
```
    name  age gender       state  num_children  num_pets
0   john   23      M  california             2         5
1   mary   78      F          dc             0         1
2  peter   22      M  california             0         0
3   jeff   19      M          dc             3         5
4   bill   45      M  california             2         2
5   lisa   33      F       texas             1         2
6   jose   20      M       texas             4         3
```

```
In [13]:  #histogram
          print(df)
          df[['age']].plot(kind='hist',bins=[0,20,40,60,80,100],rwidth=0.8)
          plt.show()
```

```
      name  age gender        state  num_children  num_pets
0     john   23      M   california             2         5
1     mary   78      F           dc             0         1
2    peter   22      M   california             0         0
3     jeff   19      M           dc             3         5
4     bill   45      M   california             2         2
5     lisa   33      F        texas             1         2
6     jose   20      M        texas             4         3
```
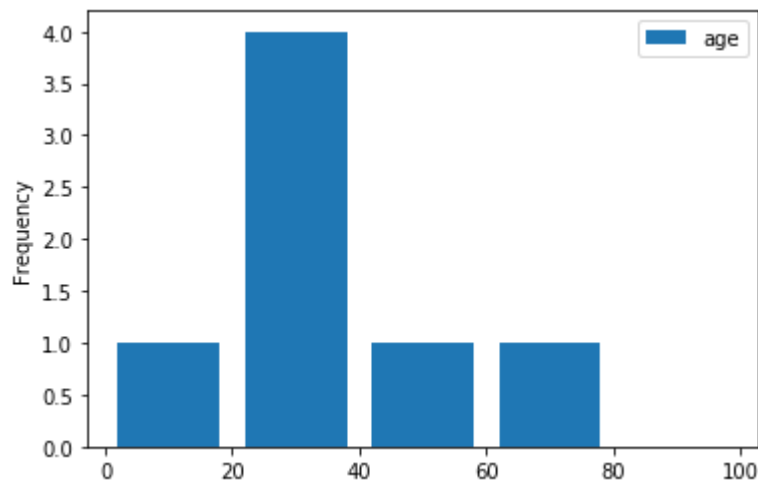


```
In [14]:  #plot date functions
          df1 = pd.DataFrame({
              'name':[
                  'john','lisa','peter','carl','linda','betty'
              ],
              'date_of_birth':[
                  '01/21/1988','03/10/1977','07/25/1999','01/22/1977','09/30/1968','09/
              ]
          })
          print(df1)
```

```
      name date_of_birth
0     john    01/21/1988
1     lisa    03/10/1977
2    peter    07/25/1999
3     carl    01/22/1977
4    linda    09/30/1968
5    betty    09/15/1970
```

```
In [15]:  ▶| df1['date_of_birth'] = pd.to_datetime(df1['date_of_birth'],infer_datetime_for
          plt.clf()   #clear the figure
          df1['date_of_birth'].map(lambda d: d.month).plot(kind='hist',alpha = 0.4,colc
          plt.show()
```
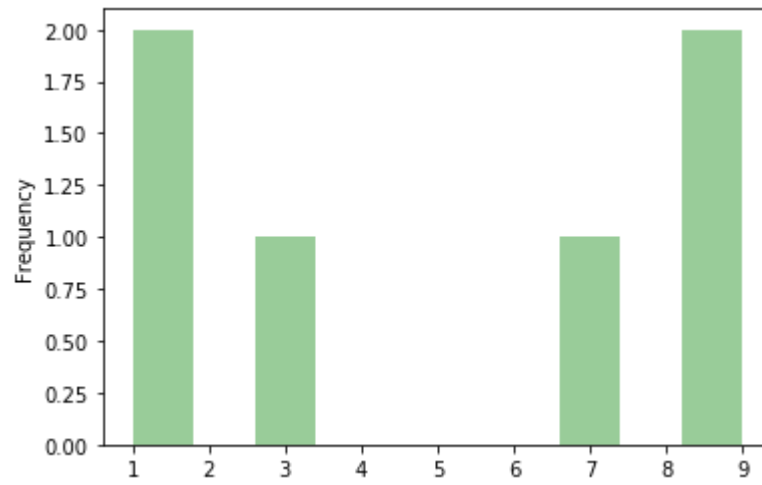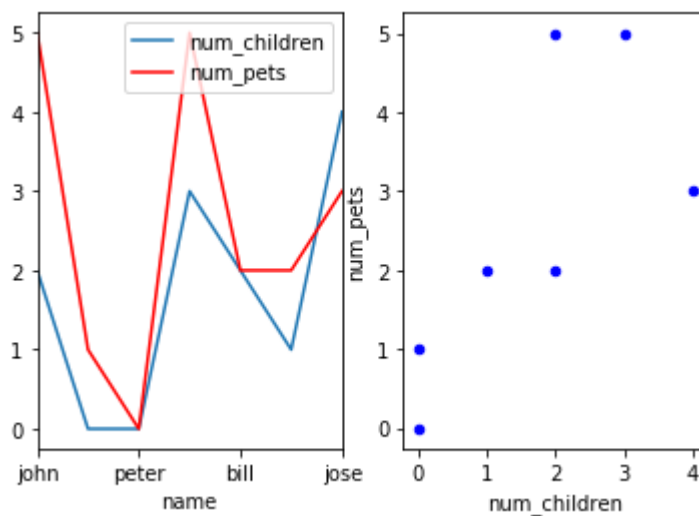
In [17]: ▶|
```python
#draw multiple plots

fig = plt.figure()

#plot 1
ax = fig.add_subplot(121)

df.plot(kind='line',x='name',y='num_children', ax=ax)
df.plot(kind='line',x='name',y='num_pets', color='red',ax=ax)

#plot 2
ax2 = fig.add_subplot(122)
df.plot(kind='scatter', x='num_children',y='num_pets',color='blue',ax=ax2 )


plt.show()
```


In [18]: ▶|
```python
#list the available styles and themes
print(plt.style.available)
```

```
['bmh', 'classic', 'dark_background', 'fast', 'fivethirtyeight', 'ggplot',
'grayscale', 'seaborn-bright', 'seaborn-colorblind', 'seaborn-dark-palett
e', 'seaborn-dark', 'seaborn-darkgrid', 'seaborn-deep', 'seaborn-muted', 's
eaborn-notebook', 'seaborn-paper', 'seaborn-pastel', 'seaborn-poster', 'sea
born-talk', 'seaborn-ticks', 'seaborn-white', 'seaborn-whitegrid', 'seabor
n', 'Solarize_Light2', 'tableau-colorblind10', '_classic_test']
```

In [20]: ▶|
```python
#change the style
plt.style.use('ggplot')
```

In [21]:

```python
fig = plt.figure()

#plot 1
ax = fig.gca()

df.plot(kind='line',x='name',y='num_children', color='blue', ax=ax)
df.plot(kind='line',x='name',y='num_pets', color='red',ax=ax)

plt.show()
```