In [1]:
```python
import numpy as np
import pandas as pd
```

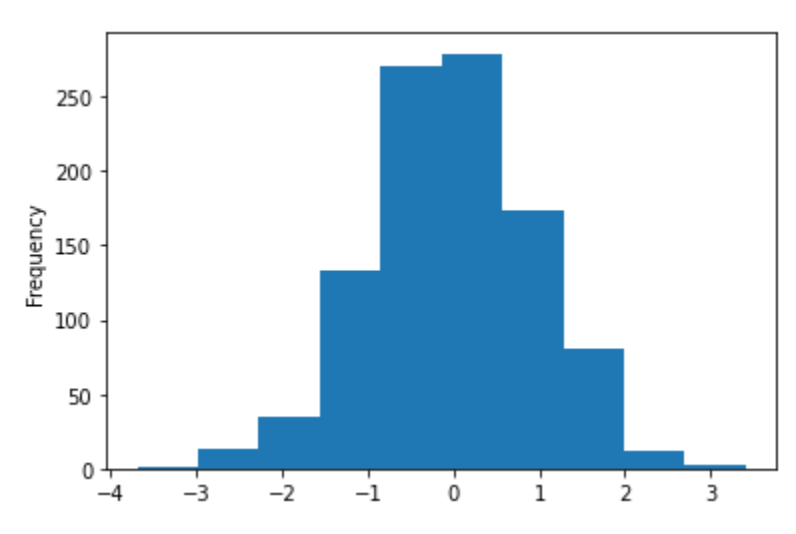
In [2]:
```python
df1 = pd.read_csv('D:\\24 - Machine_Learning\\download files\\df31.csv', index_col=0)
df2 = pd.read_csv('D:\\24 - Machine_Learning\\download files\\df32.csv')
df1
df2
```

Out[2]:

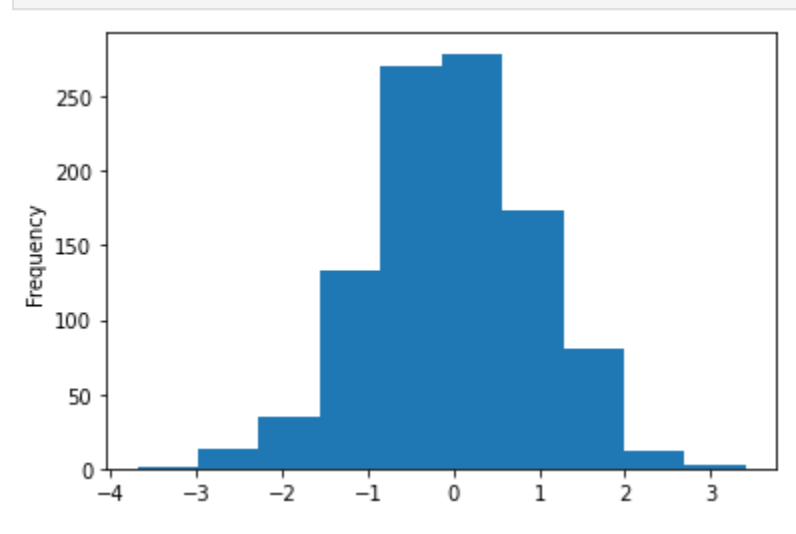
	a	b	c	d	e
0	0.039762	0.218517	0.103423	0.957904	x
1	0.937288	0.041567	0.899125	0.977680	y
2	0.780504	0.008948	0.557808	0.797510	x
3	0.672717	0.247870	0.264071	0.444358	z
4	0.053829	0.520124	0.552264	0.190008	y
5	0.286043	0.593465	0.907307	0.637898	x
6	0.430436	0.166230	0.469383	0.497701	z
7	0.312296	0.502823	0.806609	0.850519	z
8	0.187765	0.997075	0.895955	0.530390	x
9	0.908162	0.232726	0.414138	0.432007	y

In [3]:
```python
# HISTOGRAM
df1['A'].plot.hist()
```

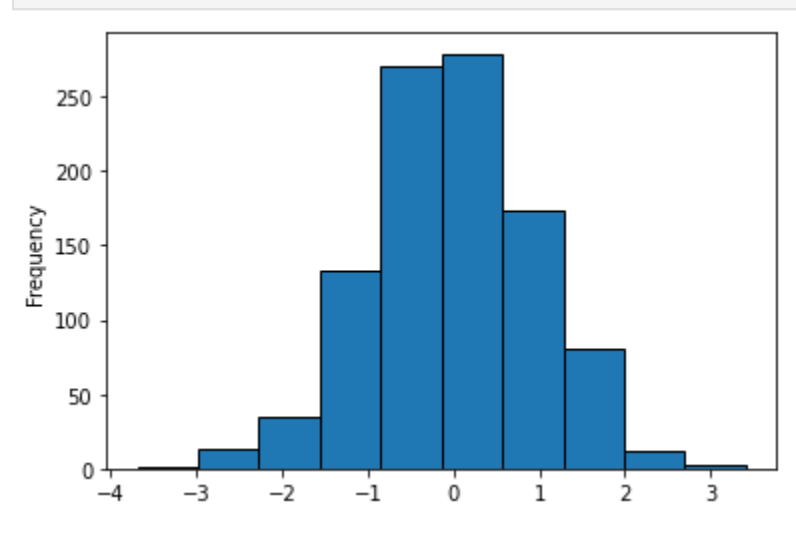
Out[3]: <AxesSubplot:ylabel='Frequency'>



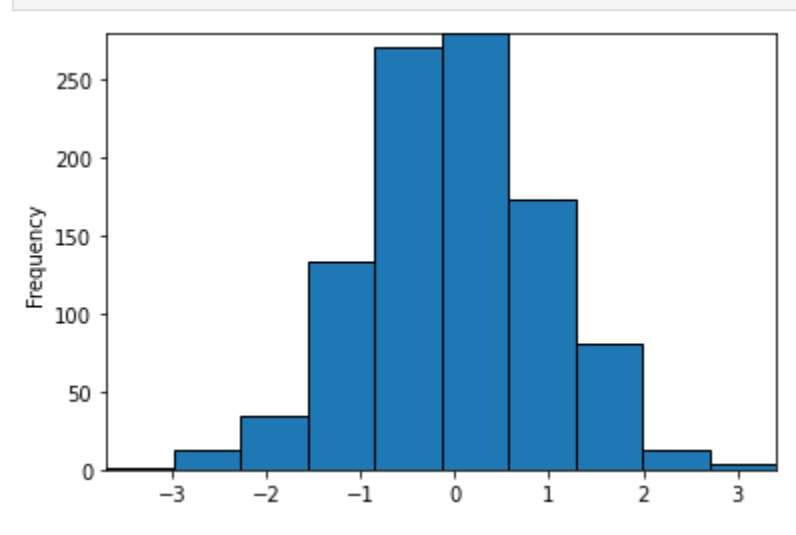
In [4]:
```python
df1['A'].plot.hist();
# add ; at the end if not require ylabel in the ouput i.e.<AxesSubplot:ylabel='Frequency'>
```



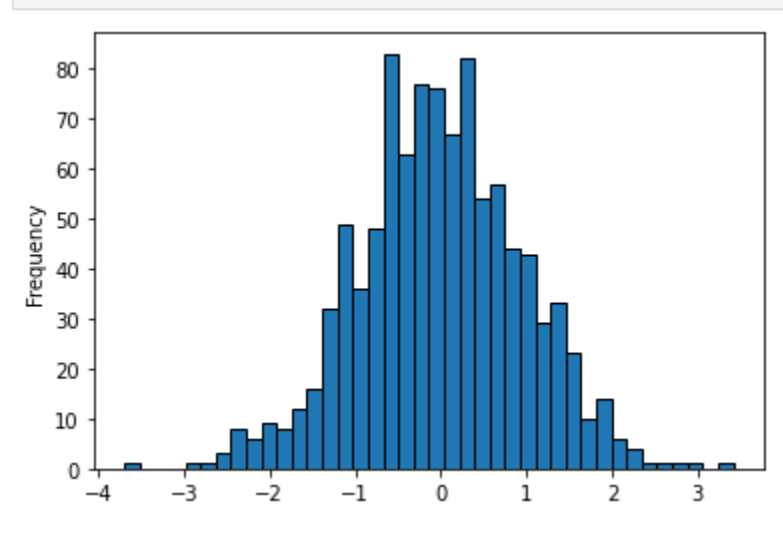
In [5]:
```python
df1['A'].plot.hist(edgecolor = 'k').autoscale(enable=True, axis='both')
# used to define edge color
```



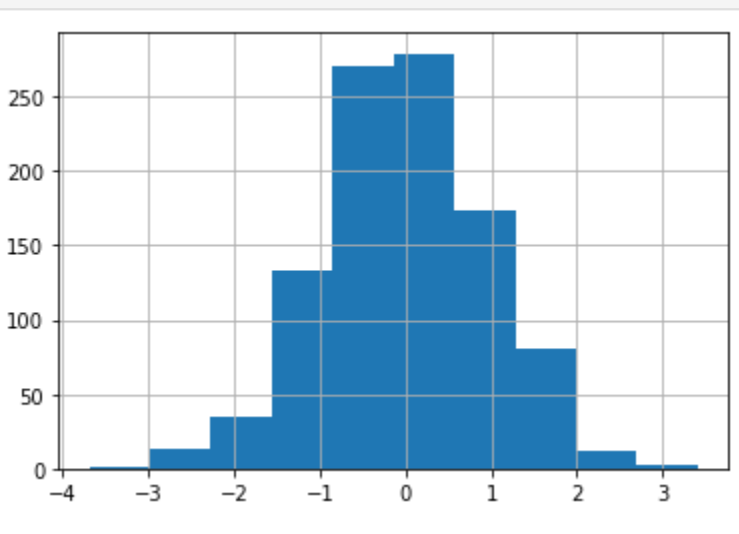
In [6]:
```python
df1['A'].plot.hist(edgecolor = 'k').autoscale(enable=True, axis='both', tight=True)
# Removes extra sapces
```



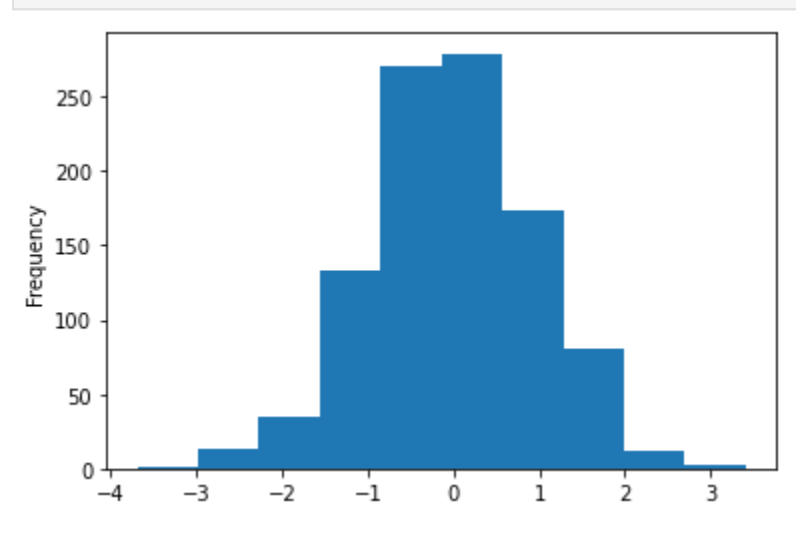
In [7]:
```python
df1['A'].plot.hist(bins = 40, edgecolor = 'k').autoscale(enable=True, axis='both')
```



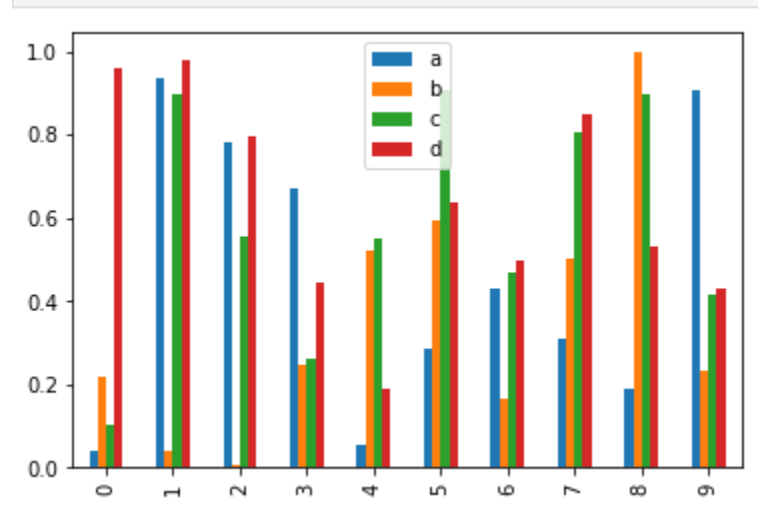
In [8]:
```python
df1['A'].hist();
```



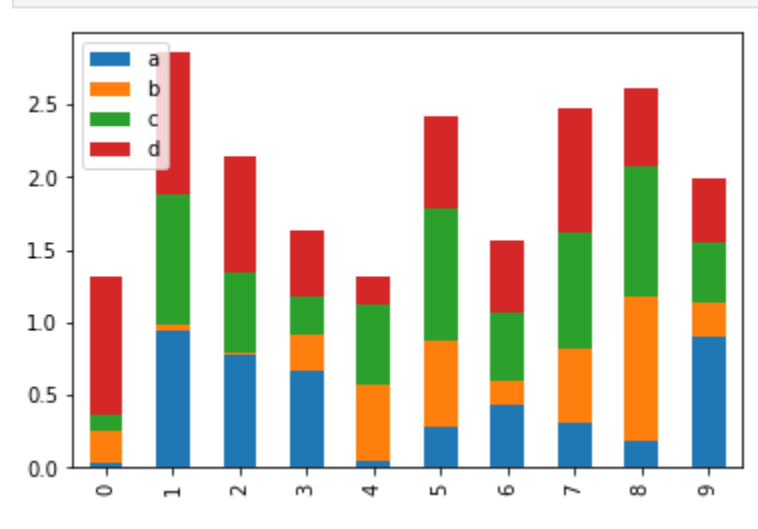
In [11]:
```python
df1['A'].hist(grid=False).set_ylabel("Frequency");
```



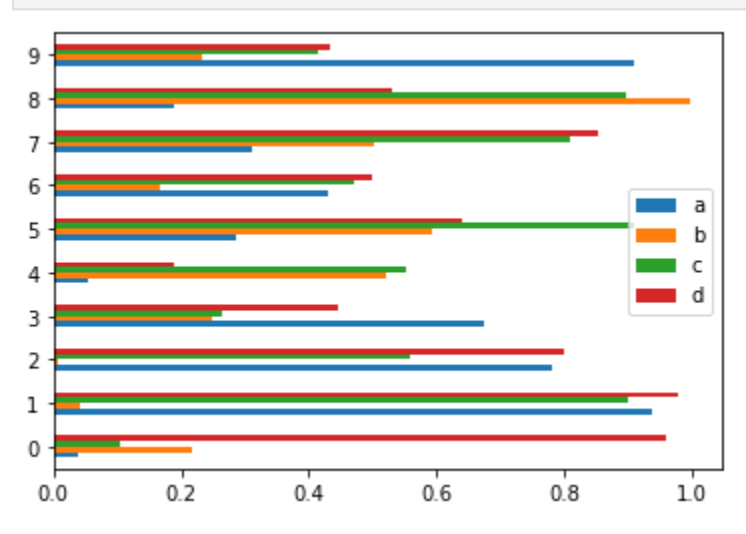
In [12]:
```python
# BAR PLOTS
df2.plot.bar();
```



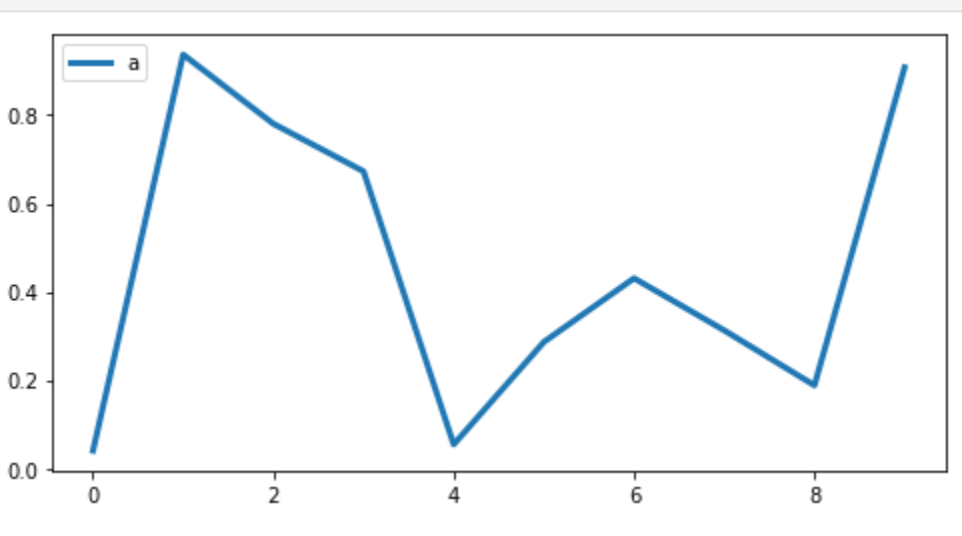
In [13]:
```python
df2.plot.bar(stacked = True);
```



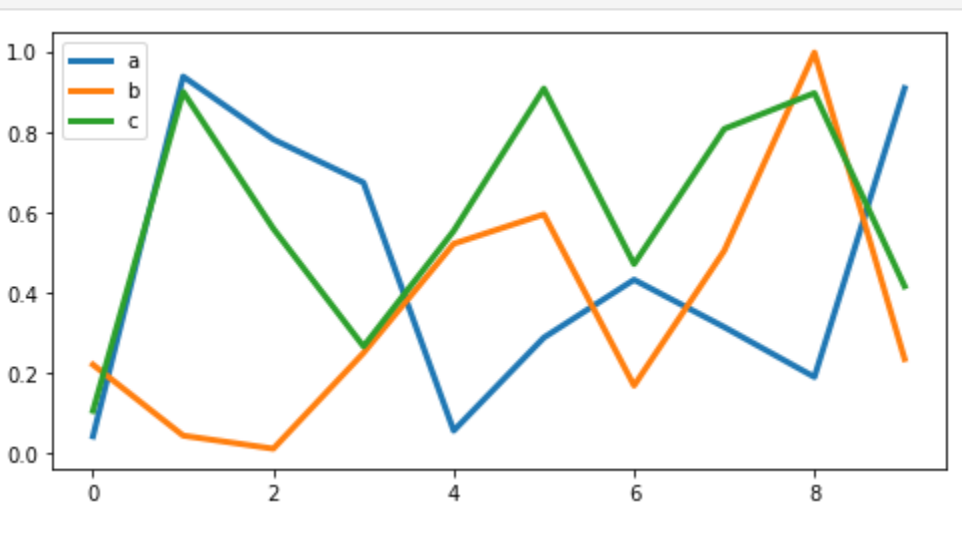
In [14]:
```python
df2.plot.barh();
# barh for horizontal bar graph
```



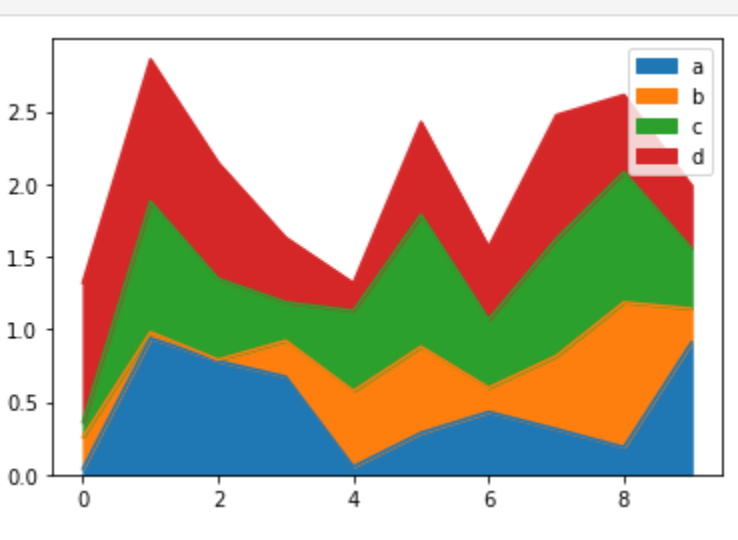
In [16]:
```python
df2.plot.line(y='a', figsize=(8,4), lw=3);
# lw - line width
```



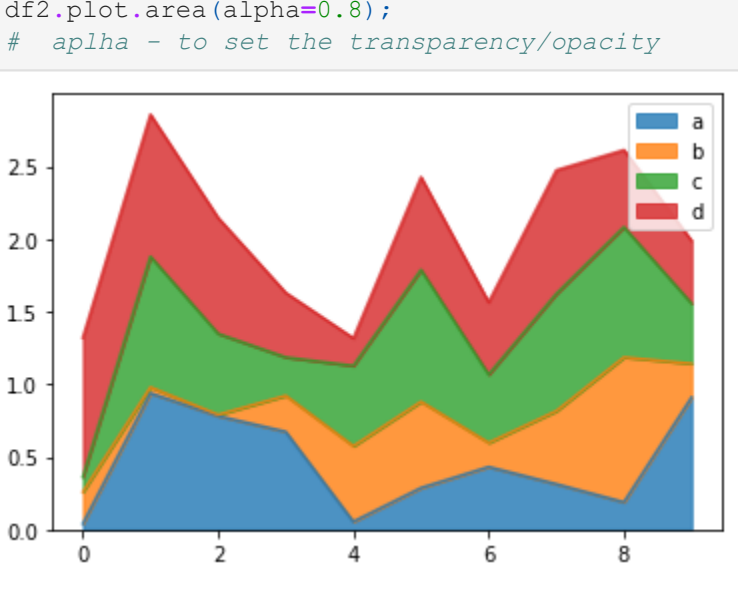
In [20]:
```python
df2.plot.line(y=['a','b','c'], figsize=(8,4), lw=3);
```



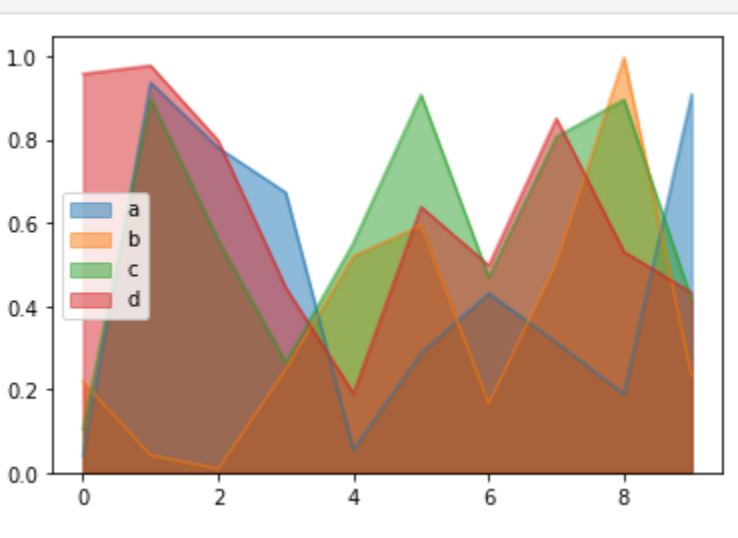
In [22]:
```python
df2.plot.area();
```



In [27]:
```python
df2.plot.area(alpha=0.8);
#  aplha - to set the transparency/opacity
```

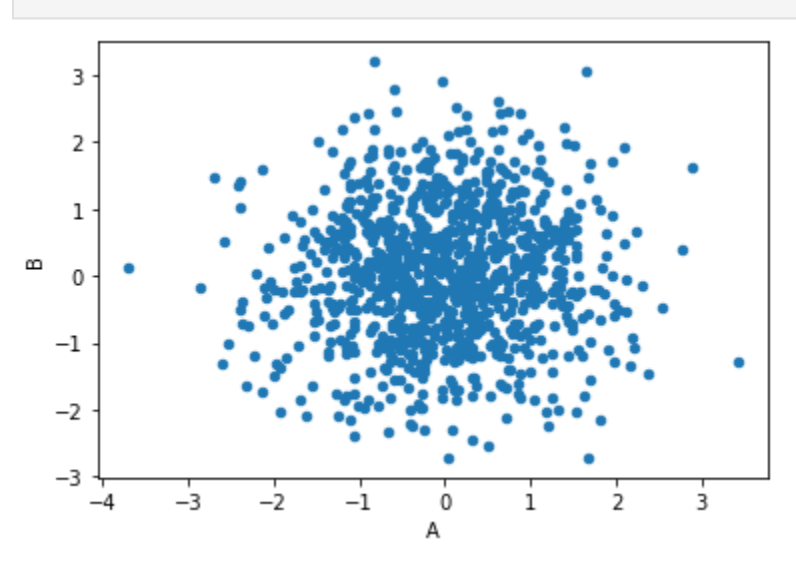


In [30]:
```python
df2.plot.area(stacked=False, alpha=0.5);
```

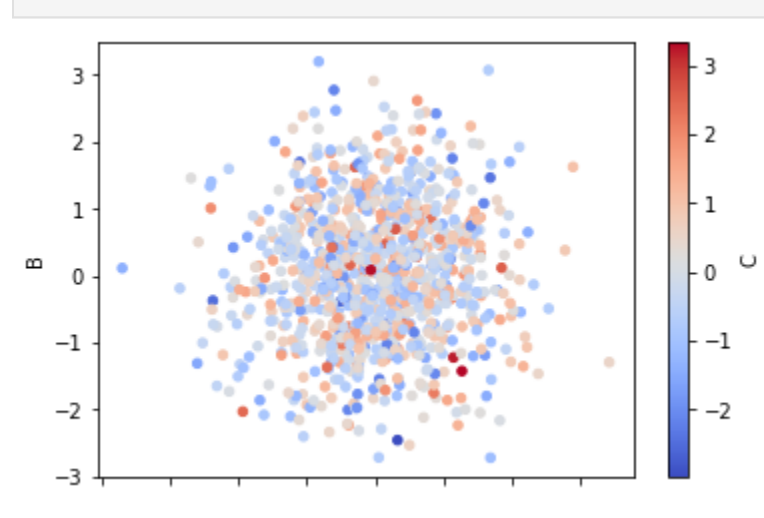


In [ ]:
```python
# SCATTER PLOT
```

In [31]:
```python
df1.plot.scatter(x='A', y='B');
```

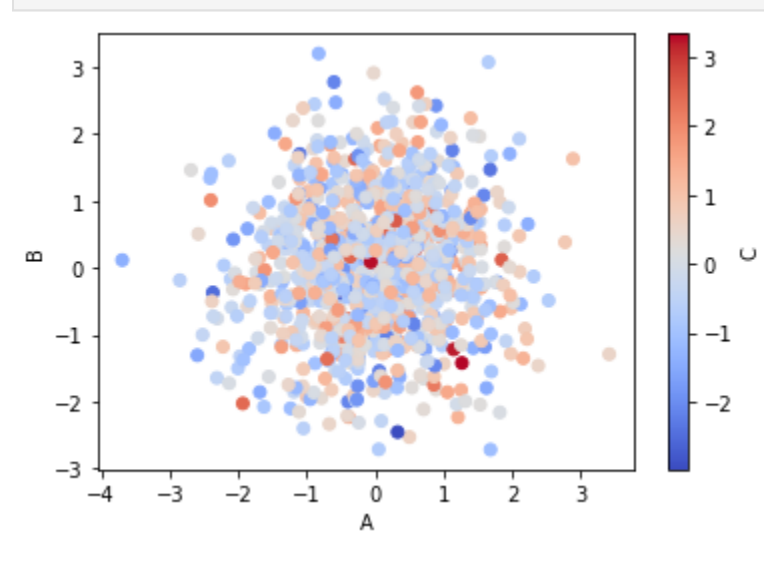


In [32]:
```python
df1.plot.scatter(x='A', y='B', c='C', cmap='coolwarm');
```



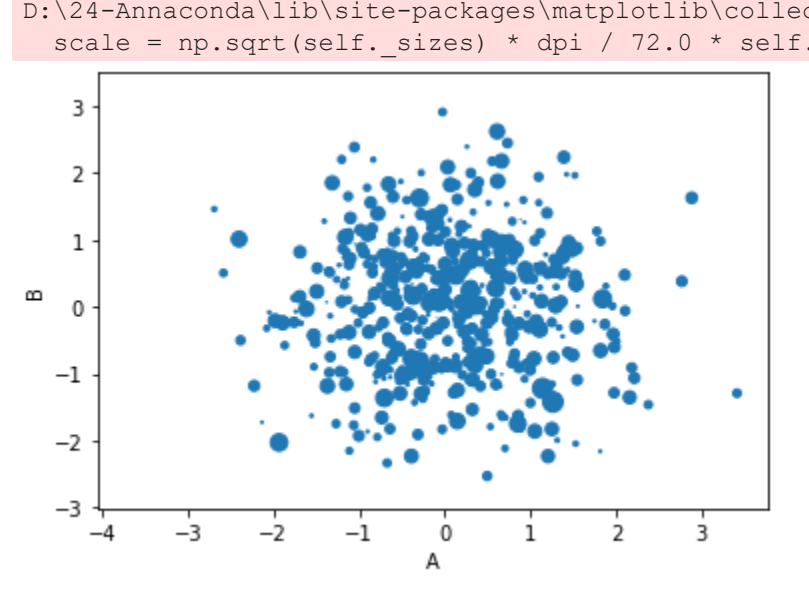
In [ ]:
```python
# MATPLOTLIB
```

In [34]:
```python
import matplotlib.pyplot as plt
plt.scatter(df1['A'], df1['B'], c=df1['C'], cmap='coolwarm')
plt.colorbar().set_label('C')
plt.xlabel('A')
plt.ylabel('B')
plt.show()
```

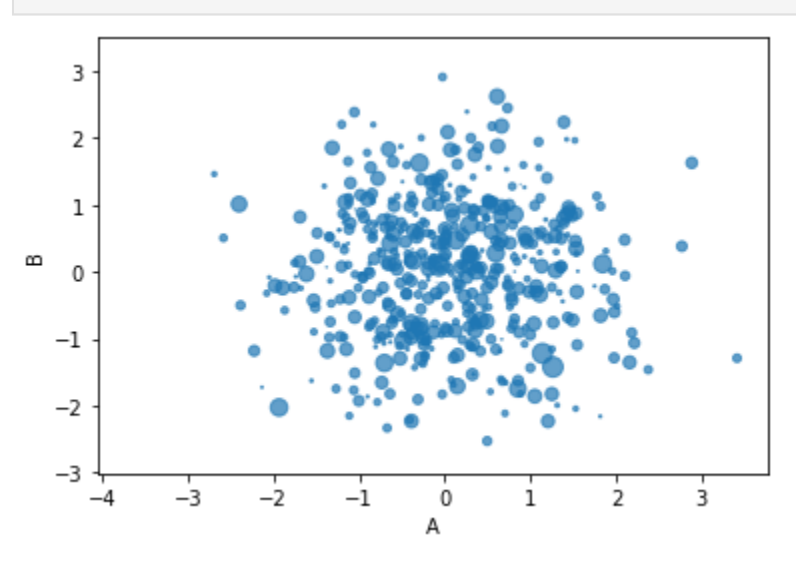


In [36]:
```python
df1.plot.scatter(x='A', y='B', s=df1['C']*30);
```

```
D:\24-Annaconda\lib\site-packages\matplotlib\collections.py:982: RuntimeWarning: invalid value encountered in sqrt
  scale = np.sqrt(self._sizes) * dpi / 72.0 * self._factor
```

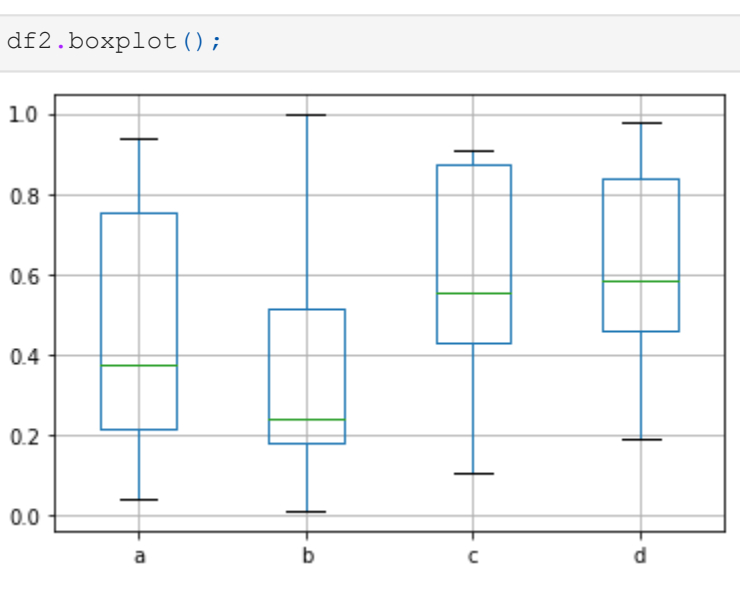


In [38]:
```python
df1.plot.scatter(x='A', y='B', s=df1['C']*30, alpha=0.7);
# here we set the transparency
```

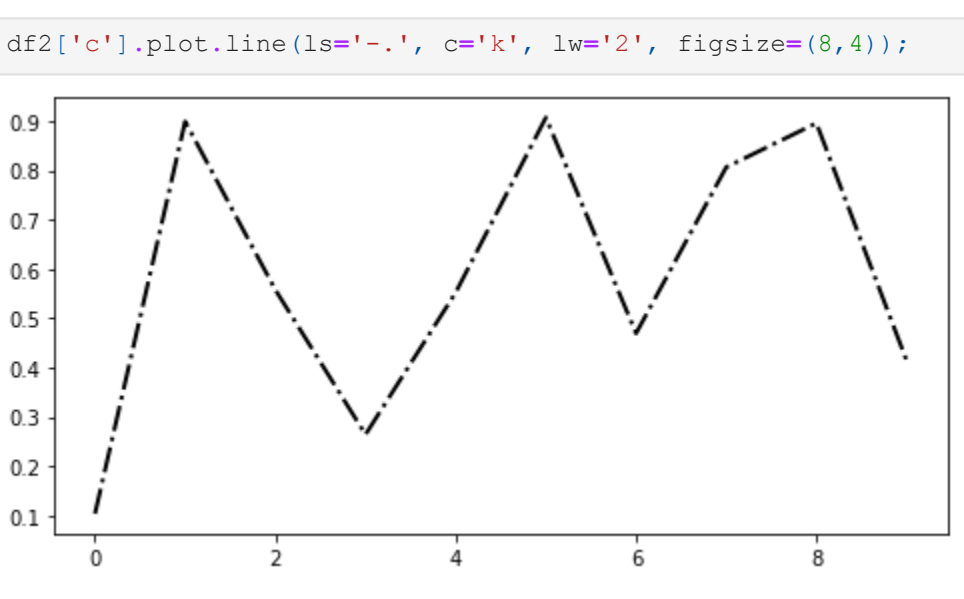


In [ ]:
```python
# BOX PLOT
```

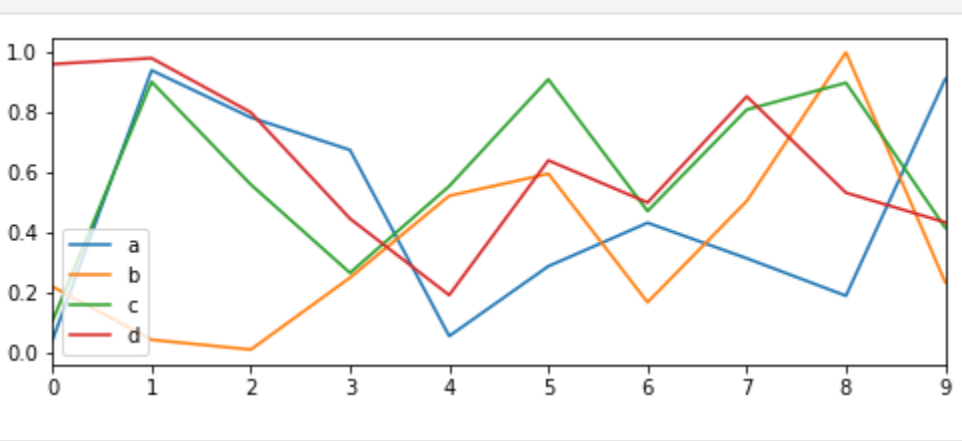
In [39]:
```python
df2.boxplot();
```



In [44]:
```python
df2['c'].plot.line(ls='-.', c='k', lw='2', figsize=(8,4));
```



In [46]:
```python
ax = df2.plot(figsize=(8,3))
ax.autoscale(axis='x', tight=True)
ax.legend(loc=3);
```



In [48]:
```python
ax = df2.plot(figsize=(8,3))
ax.autoscale(axis='x', tight=True)
ax.legend(loc=3, bbox_to_anchor=(1.0,0.1));
```