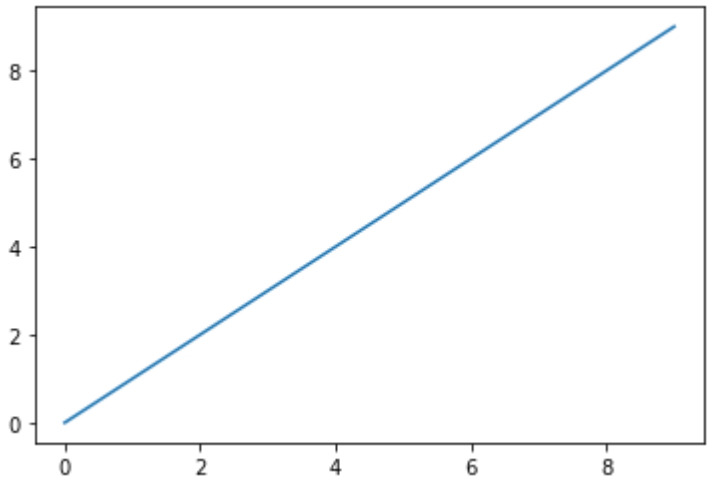
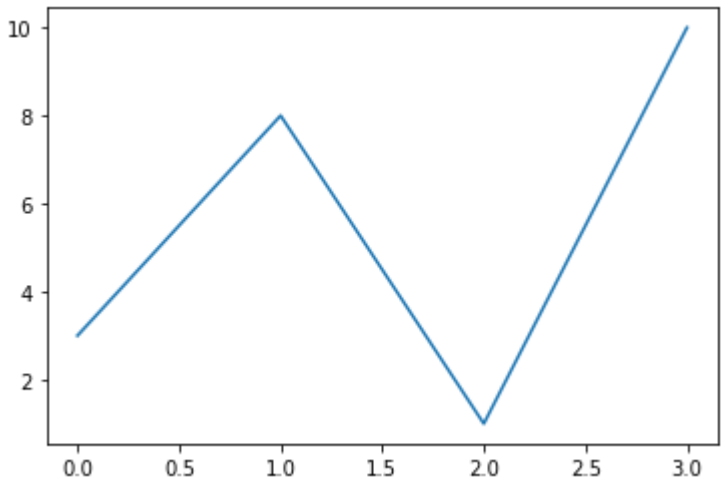
In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
data=np.arange(10)
plt.plot(data)
```

Out[1]:
```
[<matplotlib.lines.Line2D at 0x296ca0394c0>]
```



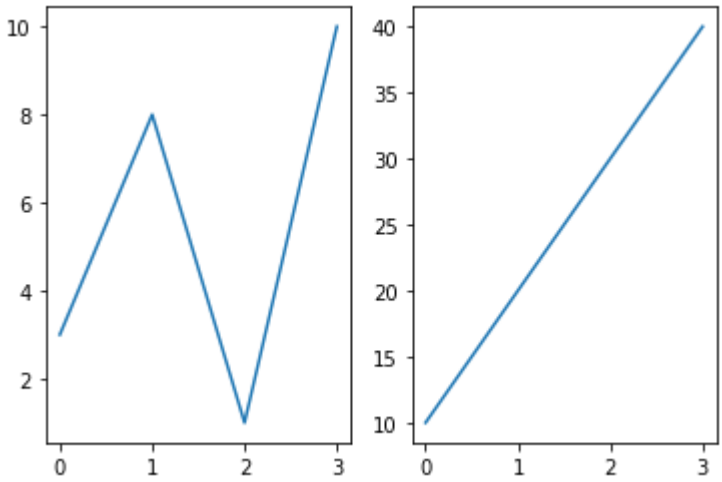
In [2]:
```
x = np.array([0, 1, 2, 3])
y = np.array([3, 8, 1, 10])

plt.plot(x,y)
```

Out[2]:
```
[<matplotlib.lines.Line2D at 0x296ca7d66a0>]
```



In [3]:
```
import matplotlib.pyplot as plt
import numpy as np

#plot 1:
x = np.array([0, 1, 2, 3])
y = np.array([3, 8, 1, 10])

plt.subplot(1, 2, 1)
plt.plot(x,y)
```

```
#plot 2:
x = np.array([0, 1, 2, 3])
y = np.array([10, 20, 30, 40])

plt.subplot(1, 2, 2)
plt.plot(x,y)

plt.show()
```

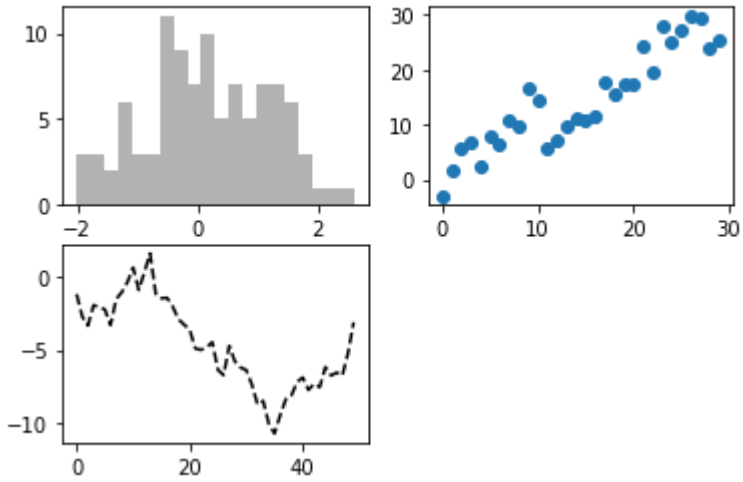


In [8]:

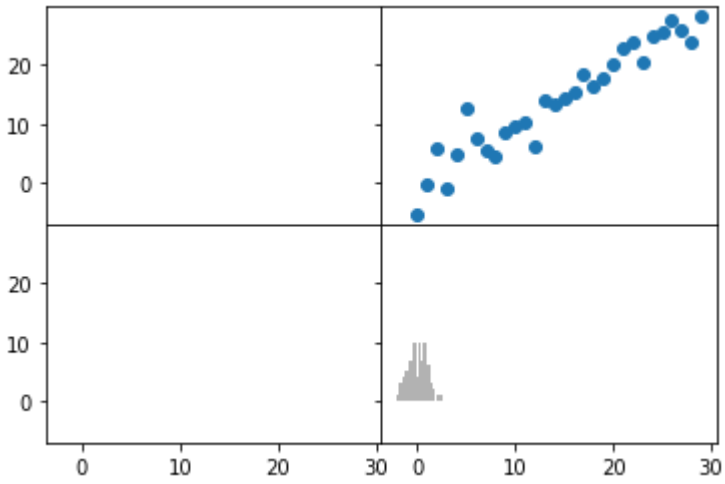
```
#plots in matplotlib reside within a Figure object.
#We can create a new figure with plt.figure
#you can't make a plot with a blank figure.
#you have to create one or more subplots.
#creating a figure with a grid of subplots

from numpy.random import randn
fig = plt.figure()
ax1 = fig.add_subplot(2, 2, 1)
ax2 = fig.add_subplot(2, 2, 2)
ax3 = fig.add_subplot(2, 2, 3)
#plt.plot(x,y)
plt.plot(randn(50).cumsum(), 'k--')
ax1.hist(randn(100), bins=20, color='k', alpha=0.3)
ax2.scatter(np.arange(30), np.arange(30) + 3 * randn(30))
```

Out[8]:

```
<matplotlib.collections.PathCollection at 0x296cada05b0>
```

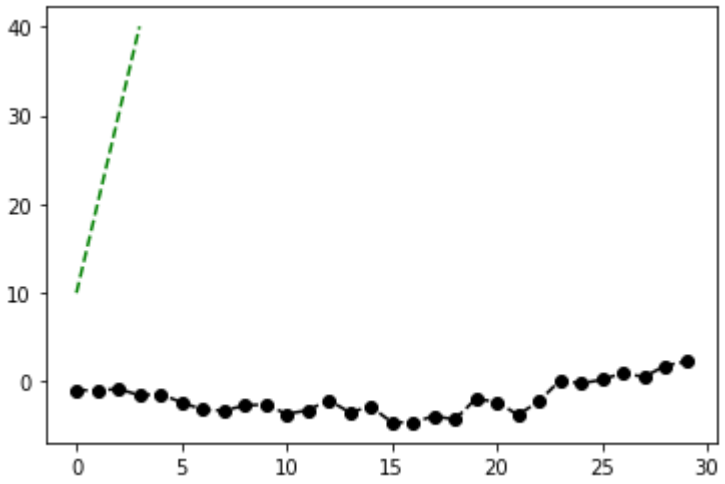


In [17]:
```
fig,axes=plt.subplots(2,2,sharex=True, sharey=True)
axes[1,1].hist(randn(100), bins=20, color='k', alpha=0.3)
axes[0,1].scatter(np.arange(30), np.arange(30) + 3 * randn(30))
#plt.subplots_adjust(wspace=0, hspace=0)
```



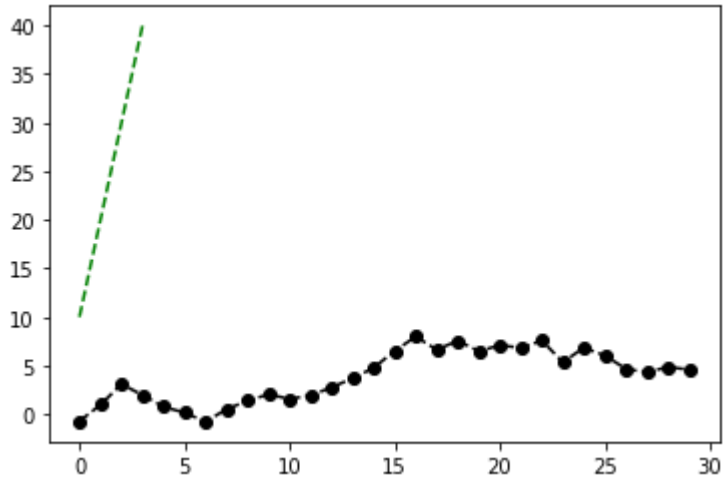
In [16]:
```
plt.plot(x, y, linestyle='--', color='g')
plt.plot(randn(30).cumsum(), 'ko--')
```

Out[16]:
```
[<matplotlib.lines.Line2D at 0x296cc3737f0>]
```



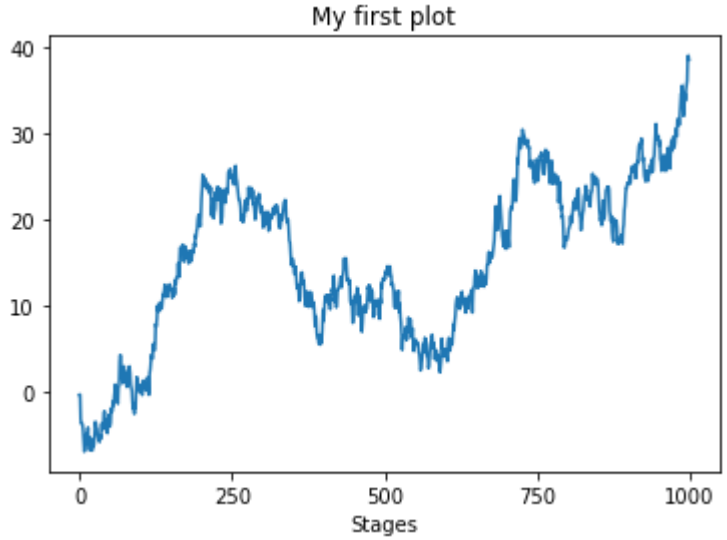
In [18]:
```
fig=plt.figure()
ax=fig.add_subplot(1,1,1)
ax.plot(x, y, linestyle='--', color='g')
plt.plot(randn(30).cumsum(), 'ko--')
```

Out[18]:
```
[<matplotlib.lines.Line2D at 0x296cae537c0>]
```



In [22]:
```
fig=plt.figure()
ax=fig.add_subplot(1,1,1)
ax.plot(randn(1000).cumsum())
ticks = ax.set_xticks([0, 250, 500, 750, 1000])
#labels = ax.set_xticklabels(['one', 'two', 'three', 'four', 'five'],
#  rotation=30, fontsize='small')
ax.set_title('My first plot')
ax.set_xlabel('Stages')
```

Out[22]:
```
Text(0.5, 0, 'Stages')
```

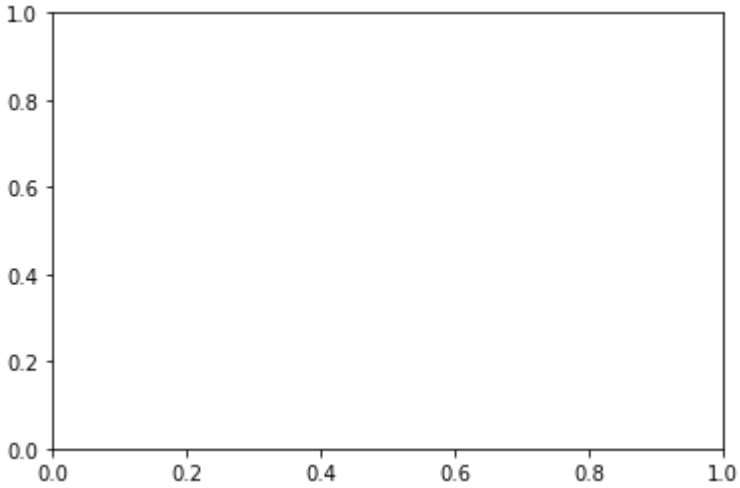


In [52]:
```
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_1372/4169554719.py in <module>
      2 fig = plt.figure()
      3 ax = fig.add_subplot(1, 1, 1)
----> 4 data = pd.read_csv('ch08/spx.csv', index_col=0, parse_dates=True)
      5 spx = data['SPX']
      6 spx.plot(ax=ax, style='k-')

C:\ProgramData\Anaconda3\lib\site-packages\pandas\util\_decorators.py in wrapper(*args,
**kwargs)
```

```
    309                     stacklevel=stacklevel,
    310                 )
--> 311             return func(*args, **kwargs)
    312
    313         return wrapper

C:\ProgramData\Anaconda3\lib\site-packages\pandas\io\parsers\readers.py in read_csv(filepath_or_buffer, sep, delimiter, header, names, index_col, usecols, squeeze, prefix, mangle_dupe_cols, dtype, engine, converters, true_values, false_values, skipinitialspace, skiprows, skipfooter, nrows, na_values, keep_default_na, na_filter, verbose, skip_blank_lines, parse_dates, infer_datetime_format, keep_date_col, date_parser, dayfirst, cache_dates, iterator, chunksize, compression, thousands, decimal, lineterminator, quotechar, quoting, doublequote, escapechar, comment, encoding, encoding_errors, dialect, error_bad_lines, warn_bad_lines, on_bad_lines, delim_whitespace, low_memory, memory_map, float_precision, storage_options)
    584     kwds.update(kwds_defaults)
    585
--> 586     return _read(filepath_or_buffer, kwds)
    587
    588

C:\ProgramData\Anaconda3\lib\site-packages\pandas\io\parsers\readers.py in _read(filepath_or_buffer, kwds)
    480
    481     # Create the parser.
--> 482     parser = TextFileReader(filepath_or_buffer, **kwds)
    483
    484     if chunksize or iterator:

C:\ProgramData\Anaconda3\lib\site-packages\pandas\io\parsers\readers.py in __init__(self, f, engine, **kwds)
    809             self.options["has_index_names"] = kwds["has_index_names"]
    810
--> 811         self._engine = self._make_engine(self.engine)
    812
    813     def close(self):

C:\ProgramData\Anaconda3\lib\site-packages\pandas\io\parsers\readers.py in _make_engine(self, engine)
   1038             )
   1039         # error: Too many arguments for "ParserBase"
-> 1040         return mapping[engine](self.f, **self.options)  # type: ignore[call-arg]
   1041
   1042     def _failover_to_python(self):

C:\ProgramData\Anaconda3\lib\site-packages\pandas\io\parsers\c_parser_wrapper.py in __init__(self, src, **kwds)
     49
     50         # open handles
---> 51         self._open_handles(src, kwds)
     52         assert self.handles is not None
     53

C:\ProgramData\Anaconda3\lib\site-packages\pandas\io\parsers\base_parser.py in _open_handles(self, src, kwds)
    220         Let the readers open IOHandles after they are done with their potential raises.
    221         """
--> 222         self.handles = get_handle(
```

```
    223             src,
    224             "r",

C:\ProgramData\Anaconda3\lib\site-packages\pandas\io\common.py in get_handle(path_or_buf, mode, encoding, compression, memory_map, is_text, errors, storage_options)
    700         if ioargs.encoding and "b" not in ioargs.mode:
    701             # Encoding
--> 702             handle = open(
    703                 handle,
    704                 ioargs.mode,

FileNotFoundError: [Errno 2] No such file or directory: 'ch08/spx.csv'
```

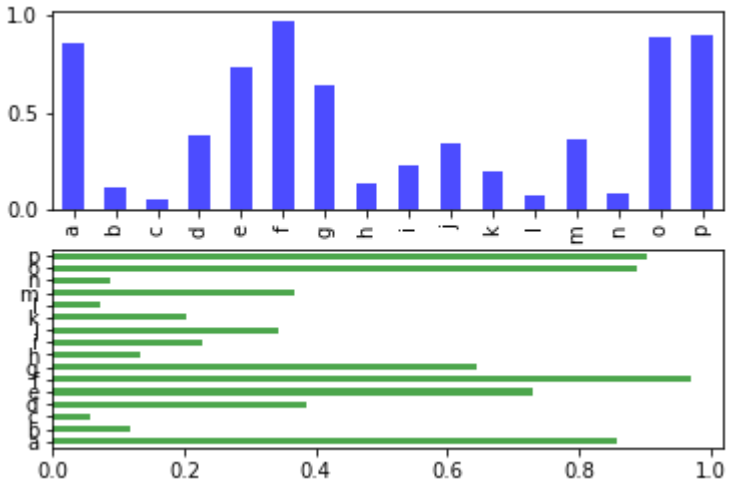


In [29]:

```
#bar plots
import pandas as pd
fig, axes = plt.subplots(2, 1)
data = pd.Series(np.random.rand(16), index=list('abcdefghijklmnop'))
data.plot(kind='bar', ax=axes[0], color='b', alpha=0.7)
data.plot(kind='barh', ax=axes[1], color='g', alpha=0.7)
```

Out[29]:

```
<AxesSubplot:>
```

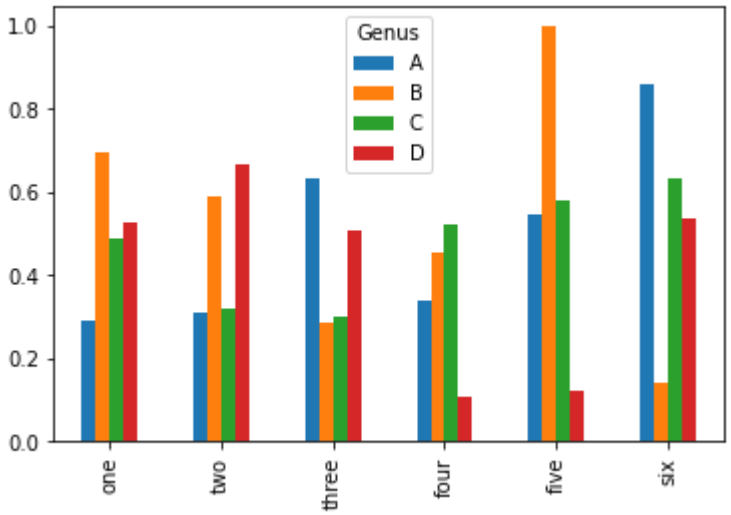


In [32]:

```
df = pd.DataFrame(np.random.rand(6, 4),
  index=['one', 'two', 'three', 'four', 'five', 'six'],
  columns=pd.Index(['A', 'B', 'C', 'D'], name='Genus'))
```

```
#df.plot(kind='bar')
```

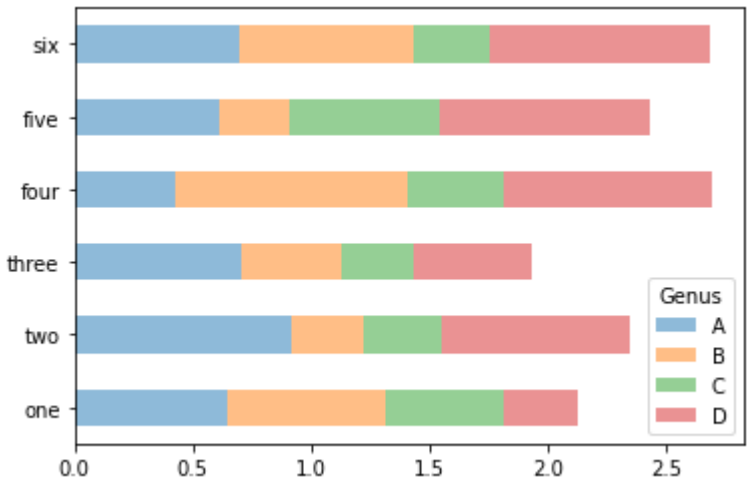
Out[32]:
```
<AxesSubplot:>
```



In [38]:
```
df = pd.DataFrame(np.random.rand(6, 4),
  index=['one', 'two', 'three', 'four', 'five', 'six'],
  columns=pd.Index(['A', 'B', 'C', 'D'], name='Genus'))
print(df)
#df.plot(kind='bar')
df.plot.barh(stacked=True,alpha=0.5)
```

```
Genus          A         B         C         D
one     0.643842  0.673445  0.492490  0.314854
two     0.917955  0.304043  0.328431  0.793237
three   0.702024  0.421503  0.307738  0.503293
four    0.422215  0.980280  0.406621  0.885739
five    0.612195  0.295482  0.630743  0.894134
six     0.694761  0.737103  0.320187  0.932055
```

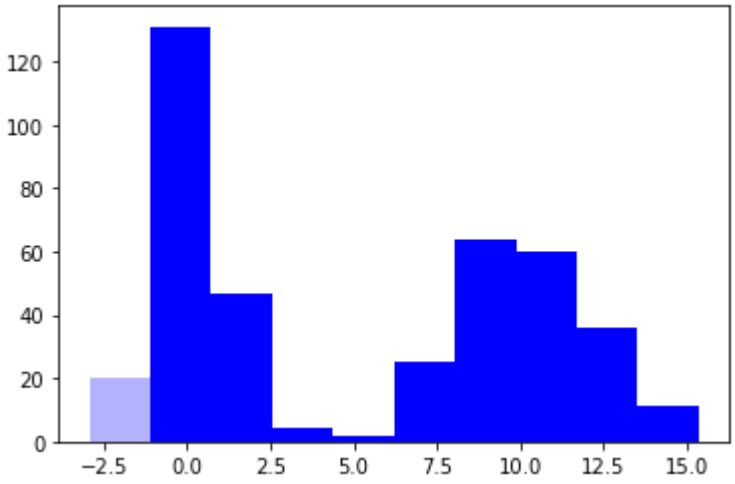
Out[38]:
```
<AxesSubplot:>
```



In [40]:
```
#Histograms and density plots
comp1 = np.random.normal(0, 1, size=200) # N(0, 1)
comp2 = np.random.normal(10, 2, size=200) # N(10, 4)
values = pd.Series(np.concatenate([comp1, comp2]))
values.hist(bins=10, alpha=0.3, color='b')
```

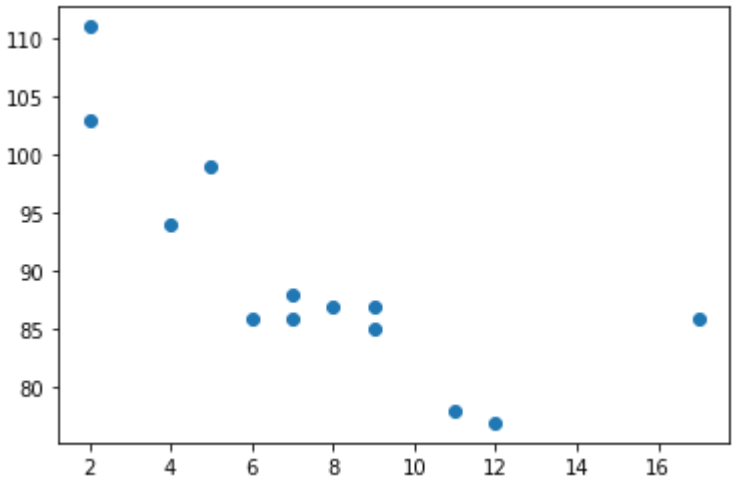
```
values.plot(kind='kde', style='k--')
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_1372/2994545680.py in <module>
      3 comp2 = np.random.normal(10, 2, size=200) # N(10, 4)
      4 values = pd.Series(np.concatenate([comp1, comp2]))
----> 5 values.hist(bins=10, alpha=0.3, color='b', normed=True)
      6
      7 values.plot(kind='kde', style='k--')

C:\ProgramData\Anaconda3\lib\site-packages\pandas\plotting\_core.py in hist_series(self, by, ax, grid, xlabelsize, xrot, ylabelsize, yrot, figsize, bins, backend, legend, **kwargs)
     98     """
     99     plot_backend = _get_plot_backend(backend)
--> 100     return plot_backend.hist_series(
    101         self,
    102         by=by,

C:\ProgramData\Anaconda3\lib\site-packages\pandas\plotting\_matplotlib\hist.py in hist_series(self, by, ax, grid, xlabelsize, xrot, ylabelsize, yrot, figsize, bins, legend, **kwds)
    352         if legend:
    353             kwds["label"] = self.name
--> 354         ax.hist(values, bins=bins, **kwds)
    355         if legend:
    356             ax.legend()

C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\__init__.py in inner(ax, data, *args, **kwargs)
   1359     def inner(ax, *args, data=None, **kwargs):
   1360         if data is None:
-> 1361             return func(ax, *map(sanitize_sequence, args), **kwargs)
   1362
   1363         bound = new_sig.bind(ax, *args, **kwargs)

C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py in hist(self, x, bins, range, density, weights, cumulative, bottom, histtype, align, orientation, rwidth, log, color, label, stacked, **kwargs)
   6909             if patch:
   6910                 p = patch[0]
-> 6911                 p.update(kwargs)
   6912                 if lbl is not None:
   6913                     p.set_label(lbl)

C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\artist.py in update(self, props)
   1060                     func = getattr(self, f"set_{k}", None)
   1061                     if not callable(func):
-> 1062                         raise AttributeError(f"{type(self).__name__!r} object "
   1063                                              f"has no property {k!r}")
   1064                     ret.append(func(v))

AttributeError: 'Rectangle' object has no property 'normed'
```



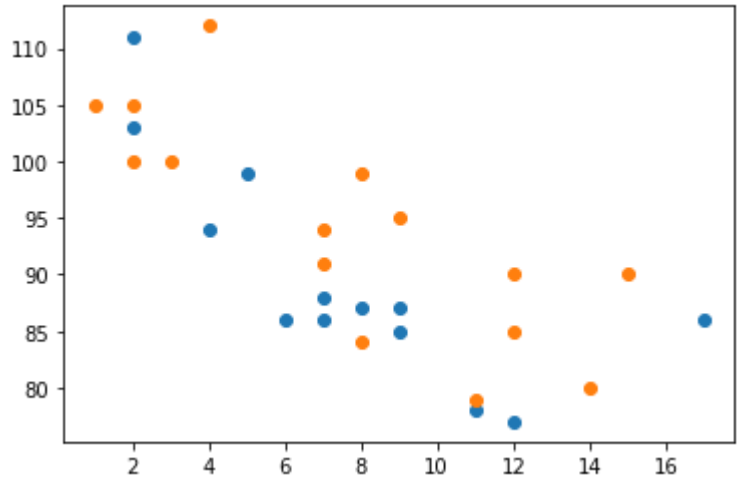
In [45]:

```
#scatter plots
import matplotlib.pyplot as plt
import numpy as np

x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])
y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])

plt.scatter(x, y)
plt.show()
```



In [46]:

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

#day one, the age and speed of 13 cars:
x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])
y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])
plt.scatter(x, y)

#day two, the age and speed of 15 cars:
x = np.array([2,2,8,1,15,8,12,9,7,3,11,4,7,14,12])
y = np.array([100,105,84,105,90,99,90,95,94,100,79,112,91,80,85])
plt.scatter(x, y)

plt.show()
```



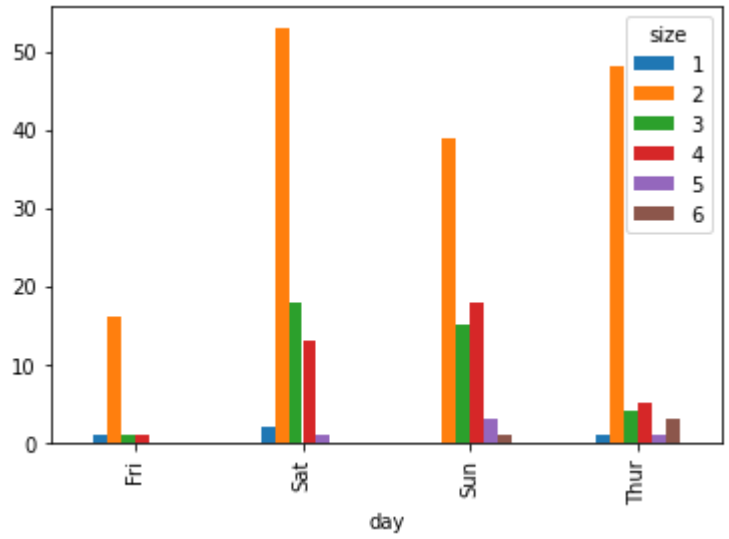
In [51]:

```
import seaborn as sns
tips=pd.read_csv('tips.csv')
party_counts=pd.crosstab(tips['day'],tips['size'])
print(party_counts)
party_counts.plot.bar()
```

```
size  1   2   3   4  5  6
day
Fri   1  16   1   1  0  0
Sat   2  53  18  13  1  0
Sun   0  39  15  18  3  1
Thur  1  48   4   5  1  3
```

Out[51]:

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



In [ ]: