Line chart

In [1]:
```
import pandas as pd
```

In [2]:
```
df = pd.read_excel('data.xlsx',sheet_name='bts')
```

In [3]:
```
df.info()
```

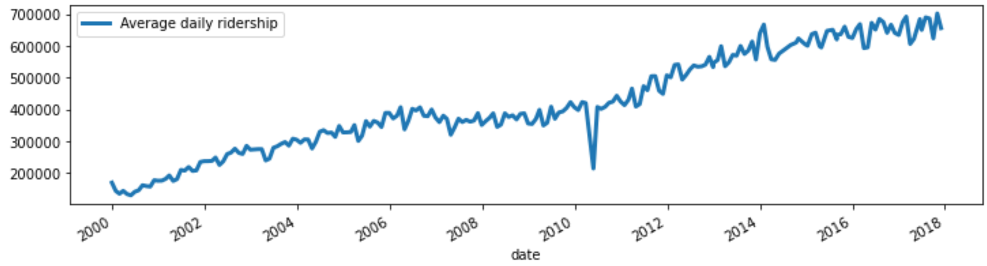
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 217 entries, 0 to 216
Data columns (total 2 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   date                     217 non-null    datetime64[ns]
 1   Average daily ridership  217 non-null    float64
dtypes: datetime64[ns](1), float64(1)
memory usage: 3.5 KB
```

1. Single plot

Using 'Pandas'

In [4]:
```
df.plot.line(x='date', y='Average daily ridership',figsize=(12,3),lw=3)
```

Out[4]:
```
<AxesSubplot:xlabel='date'>
```

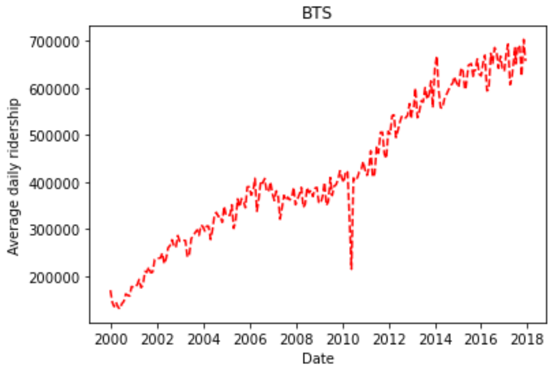


Using 'Matplotlip'

In [5]:
```
import matplotlib.pyplot as plt
```

In [6]:
```
plt.plot(df['date'],df['Average daily ridership'], 'r--') # try 'r.' , 'r.-' and 'b--'
plt.xlabel('Date')
plt.ylabel('Average daily ridership')
plt.title('BTS')
```

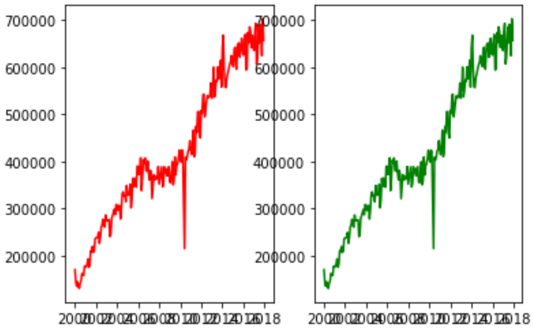
Out[6]:
```
Text(0.5, 1.0, 'BTS')
```



Subplot

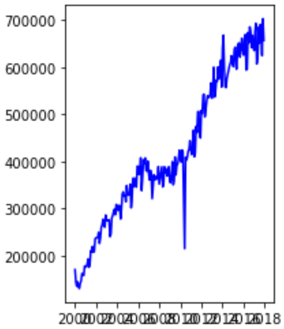
In [7]:
```
x = df['date']
y = df['Average daily ridership']
```

In [8]:
```
plt.subplot(1,2,1)
plt.plot(x, y, 'r')
plt.subplot(1,2,2)
plt.plot(x, y, 'g');
```



In [9]:

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



2. Using figure object

In [10]:

```
# Create empty figure
fig = plt.figure()
```

```
<Figure size 432x288 with 0 Axes>
```

In [11]:

```
# Add axes to figure
axes = fig.add_axes([0, 0, 1, 1]) # left, bottom, width, height (range 0 to 1)
```

In [12]:

```
# Plot on that set of axes
axes.plot(df['date'],df['Average daily ridership'], 'b')
```

Out[12]:

```
[<matplotlib.lines.Line2D at 0x7fcd9a740550>]
```

In [13]:

```
axes.set_xlabel('Date')
axes.set_ylabel('Average daily ridership')
```

Out[13]:

```
Text(0, 0.5, 'Average daily ridership')
```

In [14]:

```
axes.set_title('BTS')
```

Out[14]:

```
Text(0.5, 1.0, 'BTS')
```

Add second axes

In [15]:

```
axes2 = fig.add_axes([0.3, 0.6,0.3,0.3])
```

In [16]:

```
# Plot only year 2010 on the second axes
df_2010 = df[ (df['date']<'2011-01-01') & (df['date']>'2009-12-31')]
axes2.plot(df_2010['date'],df_2010['Average daily ridership'], 'r')
```

Out[16]:

```
[<matplotlib.lines.Line2D at 0x7fcd9b0077c0>]
```

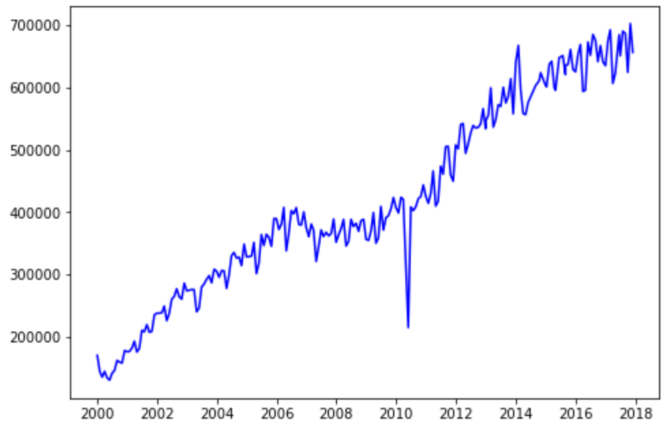
3. Set figure size

In [17]:

```
fig = plt.figure()
axes = fig.add_axes([0, 0, 1, 1])
axes.plot(df['date'],df['Average daily ridership'], 'b')
```

Out[17]:

```
[<matplotlib.lines.Line2D at 0x7fcd9b028430>]
```



When create figure object

```
In [18]: fig = plt.figure(figsize=(5,2),dpi=100)
```

```
<Figure size 500x200 with 0 Axes>
```

```
In [19]: axes = fig.add_axes([0, 0,1,1])
```

```
In [20]: axes.plot(df['date'],df['Average daily ridership'], 'b')
```

```
Out[20]: [<matplotlib.lines.Line2D at 0x7fcd9b0d3fd0>]
```

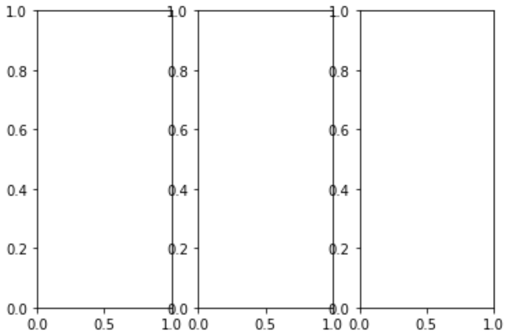
After create figure object

```
In [21]: fig.set_size_inches(5,2)
         fig.set_dpi(70)
```

4. Subplot

plot one row three columns

```
In [22]: fig, axes = plt.subplots(nrows=1, ncols=3)
```



```
In [23]: df_2010 = df[ (df['date']<'2011-01-01') & (df['date']>'2009-12-31')]
         axes[0].plot(df_2010['date'],df_2010['Average daily ridership'], 'b')
```

```
Out[23]: [<matplotlib.lines.Line2D at 0x7fcd9b3141f0>]
```

```
In [24]: df_2011 = df[ (df['date']<'2012-01-01') & (df['date']>'2010-12-31')]
         axes[1].plot(df_2011['date'],df_2011['Average daily ridership'], 'r')
```

```
Out[24]: [<matplotlib.lines.Line2D at 0x7fcd9b255b50>]
```

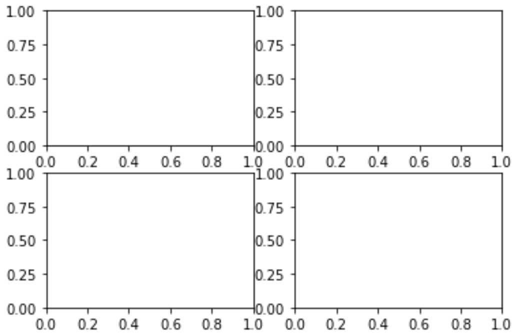
```
In [25]: df_2012 = df[ (df['date']<'2013-01-01') & (df['date']>'2011-12-31')]
         axes[2].plot(df_2012['date'],df_2012['Average daily ridership'], 'y')
```

```
Out[25]: [<matplotlib.lines.Line2D at 0x7fcd9b314d00>]
```

```
In [26]: fig.set_size_inches(15,4)
         fig.set_dpi(50)
```

plot two rows two columns

```
In [27]: fig, axes = plt.subplots(nrows=2, ncols=2)
```



In [28]:
```
df_2010 = df[ (df['date']<'2011-01-01') & (df['date']>'2009-12-31')]
axes[0,0].plot(df_2010['date'],df_2010['Average daily ridership'], 'b')
```

Out[28]:
```
[<matplotlib.lines.Line2D at 0x7fcd9af17a00>]
```

In [29]:
```
df_2011 = df[ (df['date']<'2012-01-01') & (df['date']>'2010-12-31')]
axes[0,1].plot(df_2011['date'],df_2011['Average daily ridership'], 'r')
```

Out[29]:
```
[<matplotlib.lines.Line2D at 0x7fcd9902cc10>]
```

In [30]:
```
df_2012 = df[ (df['date']<'2013-01-01') & (df['date']>'2011-12-31')]
axes[1,0].plot(df_2012['date'],df_2012['Average daily ridership'], 'y')
```

Out[30]:
```
[<matplotlib.lines.Line2D at 0x7fcd9b037e50>]
```

In [31]:
```
df_2013 = df[ (df['date']<'2014-01-01') & (df['date']>'2012-12-31')]
axes[1,1].plot(df_2013['date'],df_2013['Average daily ridership'], 'g')
```

Out[31]:
```
[<matplotlib.lines.Line2D at 0x7fcd9b028580>]
```

In [32]:
```
fig.set_size_inches(15,4)
fig.set_dpi(70)
```

5. Plot multi-line in one graph

In [33]:
```
fig = plt.figure()
```

```
<Figure size 432x288 with 0 Axes>
```

In [34]:
```
axes = fig.add_axes([0,0,1,1])
```

In [35]:
```
df_2015 = df[ (df['date']<'2016-01-01') & (df['date']>'2014-12-31')]
axes.plot(df_2015['date'],df_2015['Average daily ridership'])
```

Out[35]:
```
[<matplotlib.lines.Line2D at 0x7fcd9af56d60>]
```

In [36]:
```
df_2016 = df[ (df['date']<'2017-01-01') & (df['date']>'2015-12-31')]
axes.plot(df_2016['date'],df_2016['Average daily ridership'])
```

Out[36]:
```
[<matplotlib.lines.Line2D at 0x7fcd9b47df10>]
```

Compare ridership in each month

In [37]:
```
fig = plt.figure()
```

```
<Figure size 432x288 with 0 Axes>
```

In [38]:
```
axes = fig.add_axes([0,0,1,1])
```

In [39]:
```
df_2015 = df[ (df['date']<'2016-01-01') & (df['date']>'2014-12-31')]
```

In [40]:
```
df_2015['month'] = df_2015['date'].dt.month_name()
```

```
/var/folders/50/yc3xx4j955ndlwshz8251btr0000gn/T/ipykernel_35407/1660727743.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a
-view-versus-a-copy
  df_2015['month'] = df_2015['date'].dt.month_name()
```

In [41]:
```
axes.plot(df_2015['month'],df_2015['Average daily ridership'],label='2015')
```

Out[41]:
```
[<matplotlib.lines.Line2D at 0x7fcd9ae62e20>]
```

In [42]:
```
df_2016 = df[ (df['date']<'2017-01-01') & (df['date']>'2015-12-31')]
df_2016['month'] = df_2016['date'].dt.month_name()
axes.plot(df_2016['month'],df_2016['Average daily ridership'],label='2016')
```

```
/var/folders/50/yc3xx4j955ndlwshz8251btr0000gn/T/ipykernel_35407/650806120.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a
-view-versus-a-copy
  df_2016['month'] = df_2016['date'].dt.month_name()
```

Out[42]:

```
[<matplotlib.lines.Line2D at 0x7fcd9ae62070>]
```

In [43]:

```
axes.legend()
```

Out[43]:

```
<matplotlib.legend.Legend at 0x7fcd9ae7d490>
```

In [44]:

```
fig.set_size_inches(10,3)
fig.set_dpi(100)
```

Write function to re-structure data

In [45]:

```
all_year = list(set(df['date'].dt.year))
```

In [46]:

```
array_df = [0] * len(all_year)
for i in range(len(all_year)):
    to_year = str(all_year[i]+1) + '-01-01'
    from_year = str(all_year[i]-1) + '-12-31'
    array_df[i] = df[ (df['date']<to_year) & (df['date']>from_year)]
```

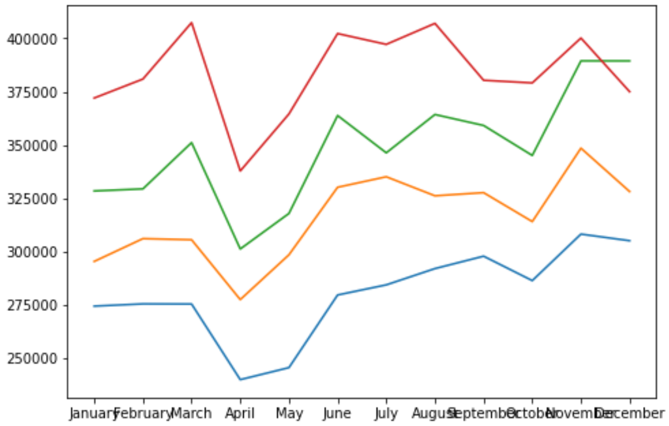
In [47]:

```
first = all_year.index(2003)
last = all_year.index(2007)
```

In [48]:

```
fig = plt.figure()
axes = fig.add_axes([0,0,1,1])
for i in range(first,last):
    dfx = array_df[i]
    dfx['month'] = dfx['date'].dt.month_name()
    axes.plot(dfx['month'],dfx['Average daily ridership'], label= 'year ' + str(all_year[i]) )
```

```
/var/folders/50/yc3xx4j955ndlwshz8251btr0000gn/T/ipykernel_35407/3614502295.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a
-view-versus-a-copy
  dfx['month'] = dfx['date'].dt.month_name()
/var/folders/50/yc3xx4j955ndlwshz8251btr0000gn/T/ipykernel_35407/3614502295.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a
-view-versus-a-copy
  dfx['month'] = dfx['date'].dt.month_name()
/var/folders/50/yc3xx4j955ndlwshz8251btr0000gn/T/ipykernel_35407/3614502295.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a
-view-versus-a-copy
  dfx['month'] = dfx['date'].dt.month_name()
/var/folders/50/yc3xx4j955ndlwshz8251btr0000gn/T/ipykernel_35407/3614502295.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a
-view-versus-a-copy
  dfx['month'] = dfx['date'].dt.month_name()
```



In [49]:

```
fig.set_size_inches(10,3)
fig.set_dpi(100)
axes.legend()
```

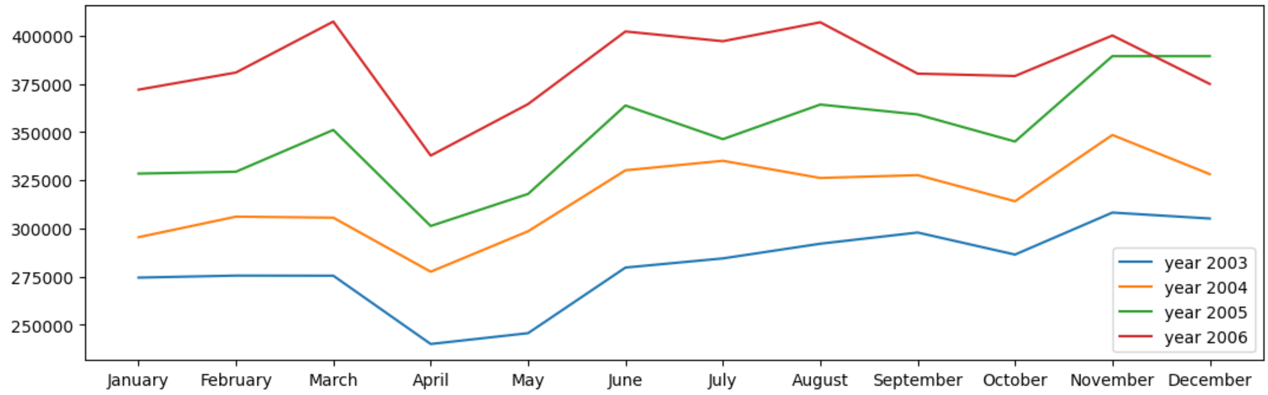
Out[49]:

```
<matplotlib.legend.Legend at 0x7fcd9b6c8b20>
```

In [50]:

```
fig
```

Out[50]:



6. Define plot range

In [51]:

```
import numpy as np
x = np.linspace(0, 5, 11)
```

In [52]:

```
x
```

Out[52]:

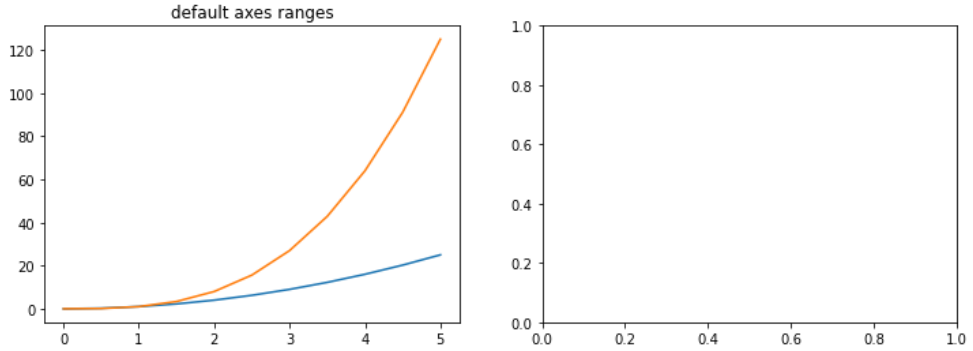
```
array([0. , 0.5, 1. , 1.5, 2. , 2.5, 3. , 3.5, 4. , 4.5, 5. ])
```

In [53]:

```
fig, axes = plt.subplots(1, 2, figsize=(12, 4))

axes[0].plot(x, x**2, x, x**3)
axes[0].set_title("default axes ranges")
```

Out[53]:

```
Text(0.5, 1.0, 'default axes ranges')
```



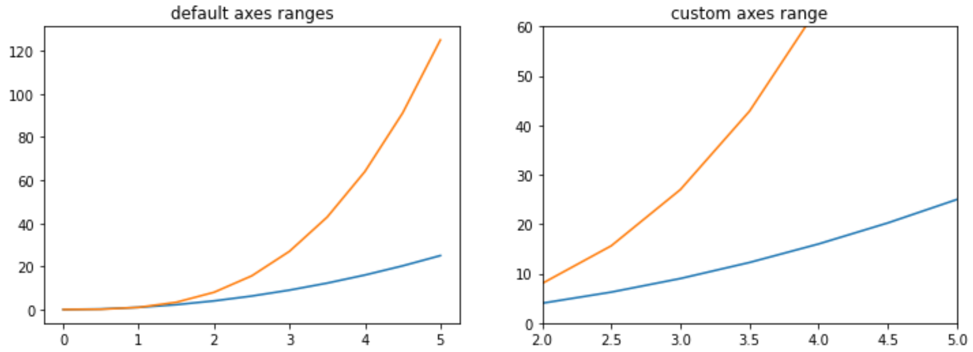
In [54]:

```
axes[1].plot(x, x**2, x, x**3)
axes[1].set_ylim([0, 60])
axes[1].set_xlim([2, 5])
axes[1].set_title("custom axes range");
```

In [55]:

```
fig
```

Out[55]:



7. Color, line width, line style

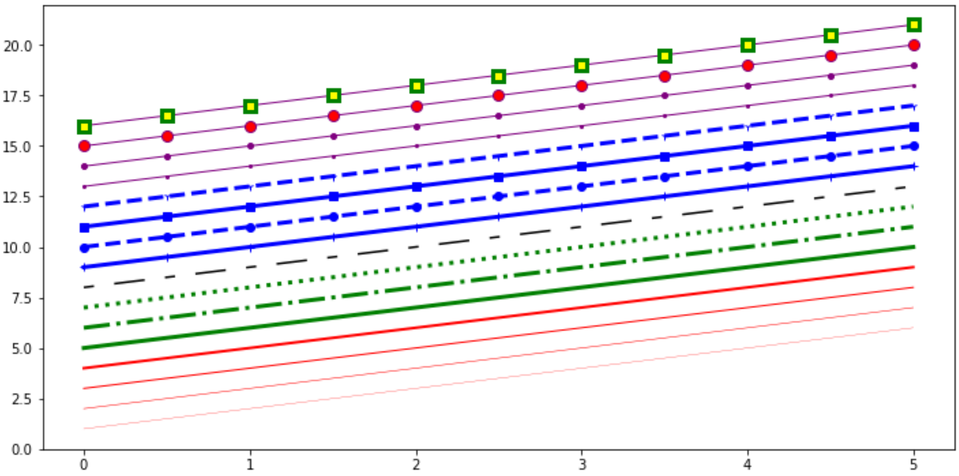
In [56]:

```
import numpy as np
x = np.linspace(0, 5, 11)
```

In [57]:

```
fig, ax = plt.subplots(figsize=(12,6))

ax.plot(x, x+1, color="red", linewidth=0.25)
ax.plot(x, x+2, color="red", linewidth=0.50)
ax.plot(x, x+3, color="red", linewidth=1.00)
ax.plot(x, x+4, color="red", linewidth=2.00)
```

```python
# possible linestype options ‘-‘, ‘–’, ‘-.’, ‘:’, ‘steps’
ax.plot(x, x+5, color="green", lw=3, linestyle='-')
ax.plot(x, x+6, color="green", lw=3, ls='-.')
ax.plot(x, x+7, color="green", lw=3, ls=':')

# custom dash
line, = ax.plot(x, x+8, color="black", lw=1.50)
line.set_dashes([5, 10, 15, 10]) # format: line length, space length, ...

# possible marker symbols: marker = '+', 'o', '*', 's', ',', '.', '1', '2', '3', '4', ...
ax.plot(x, x+ 9, color="blue", lw=3, ls='-', marker='+')
ax.plot(x, x+10, color="blue", lw=3, ls='--', marker='o')
ax.plot(x, x+11, color="blue", lw=3, ls='-', marker='s')
ax.plot(x, x+12, color="blue", lw=3, ls='--', marker='1')

# marker size and color
ax.plot(x, x+13, color="purple", lw=1, ls='-', marker='o', markersize=2)
ax.plot(x, x+14, color="purple", lw=1, ls='-', marker='o', markersize=4)
ax.plot(x, x+15, color="purple", lw=1, ls='-', marker='o', markersize=8, markerfacecolor="red")
ax.plot(x, x+16, color="purple", lw=1, ls='-', marker='s', markersize=8,
        markerfacecolor="yellow", markeredgewidth=3, markeredgecolor="green");
```



```
In [ ]:
```