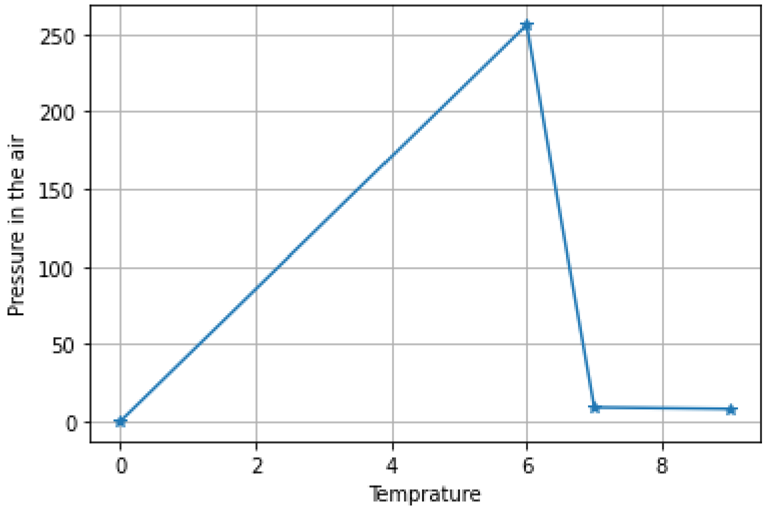
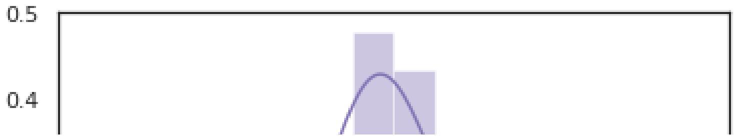
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
```

```
xpoint = np.array([0,6,7,9])
ypoint = np.array([0,256,9,8])
plt.plot(xpoint,ypoint,marker="*")
plt.xlabel("Temprature")
plt.ylabel("Pressure in the air")
plt.grid()
plt.show()
```

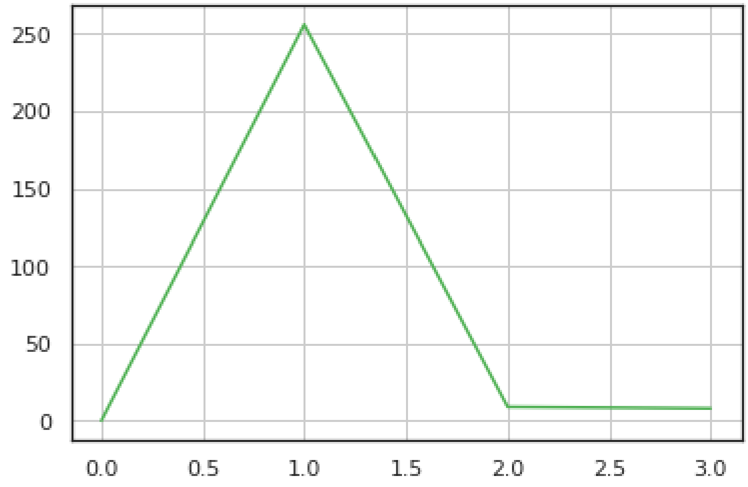


```
sns.set(style="white")
rs = np.random.RandomState(10)
d = rs.normal(size=100)
sns.distplot(d, kde=True, color="m")
```

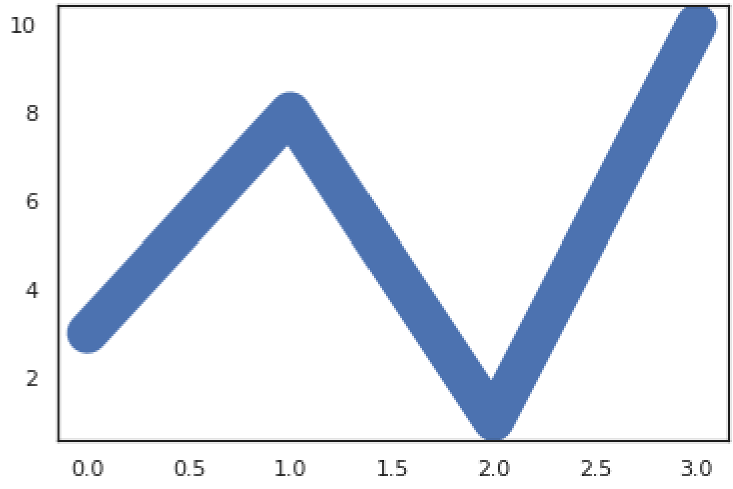
```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWa
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7fbf7378be50>
```



```
plt.plot(ypoint, c = '#4CAF50')
plt.grid()
plt.show()
```

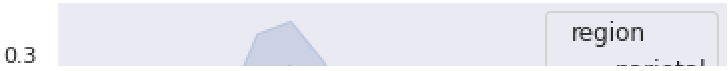


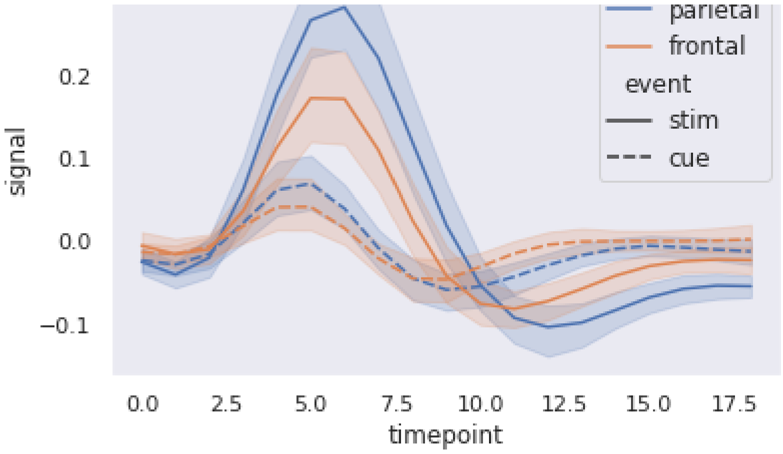
```
ypoints = np.array([3, 8, 1, 10])

plt.plot(ypoints, linewidth = '20.5')
plt.show()
```



```
sns.set(style="dark")
fmri = sns.load_dataset("fmri")

# Plot the responses for different\
# events and regions
sns.lineplot(x="timepoint",
             y="signal",
             hue="region",
             style="event",
             data=fmri)
```

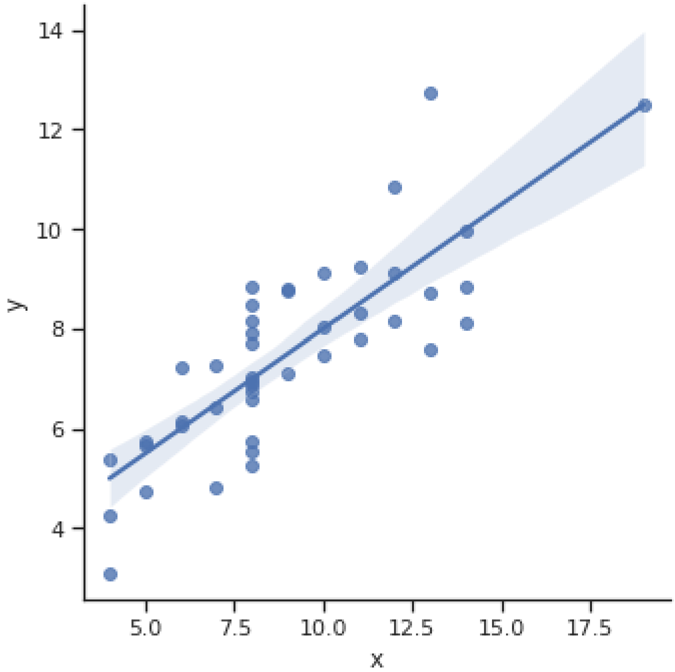
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbf737bb650>
```





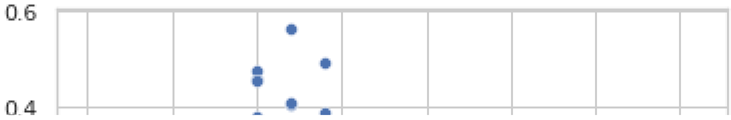
```
sns.set(style="ticks")

# Loading the dataset
df = sns.load_dataset("anscombe")

# Show the results of a linear regression
sns.lmplot(x="x", y="y", data=df)
```

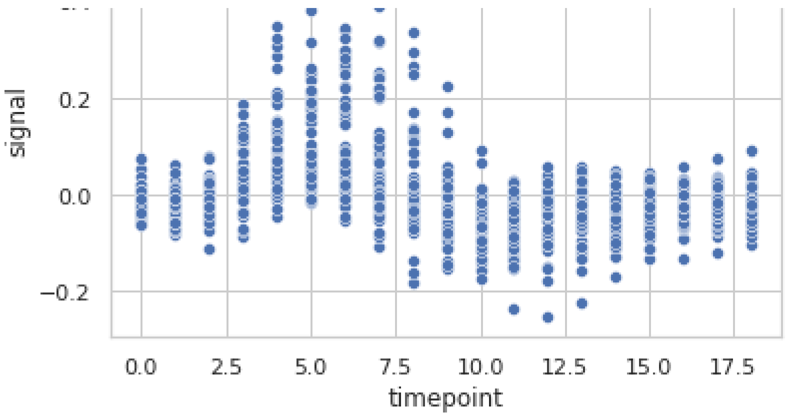
```
<seaborn.axisgrid.FacetGrid at 0x7fbf73269e50>
```



```
sns.set(style='whitegrid')
fmri = sns.load_dataset("fmri")

sns.scatterplot(x="timepoint",
                    y="signal",
                    data=fmri)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbf737cf9d0>
```





```
import scipy as sc

print(sc.constants.gram)
```

```
    0.001
```

```
arr = np.array([[0, 0, 0], [0, 0, 1], [1, 0, 2]])

print(sc.sparse.csr_matrix(arr).data)
```

```
    [1 1 2]
```

```
from scipy.sparse.csgraph import connected_components
from scipy.sparse import csr_matrix

arr = np.array([
  [0, 1, 2],
  [1, 0, 0],
  [2, 0, 0]
])

newarr = csr_matrix(arr)

print(connected_components(newarr))
```

```
    (1, array([0, 0, 0], dtype=int32))
```

```
import numpy as np
from scipy.sparse.csgraph import bellman_ford
from scipy.sparse import csr_matrix

arr = np.array([
  [0, -1, 2],
  [1, 0, 0],
  [2, 0, 0]
])
```

```
newarr = csr_matrix(arr)

print(bellman_ford(newarr, return_predecessors=True, indices=0))
```

```
    (array([ 0., -1.,  2.]), array([-9999,     0,     0], dtype=int32))
```

✓ 0s completed at 1:21 PM ● ×