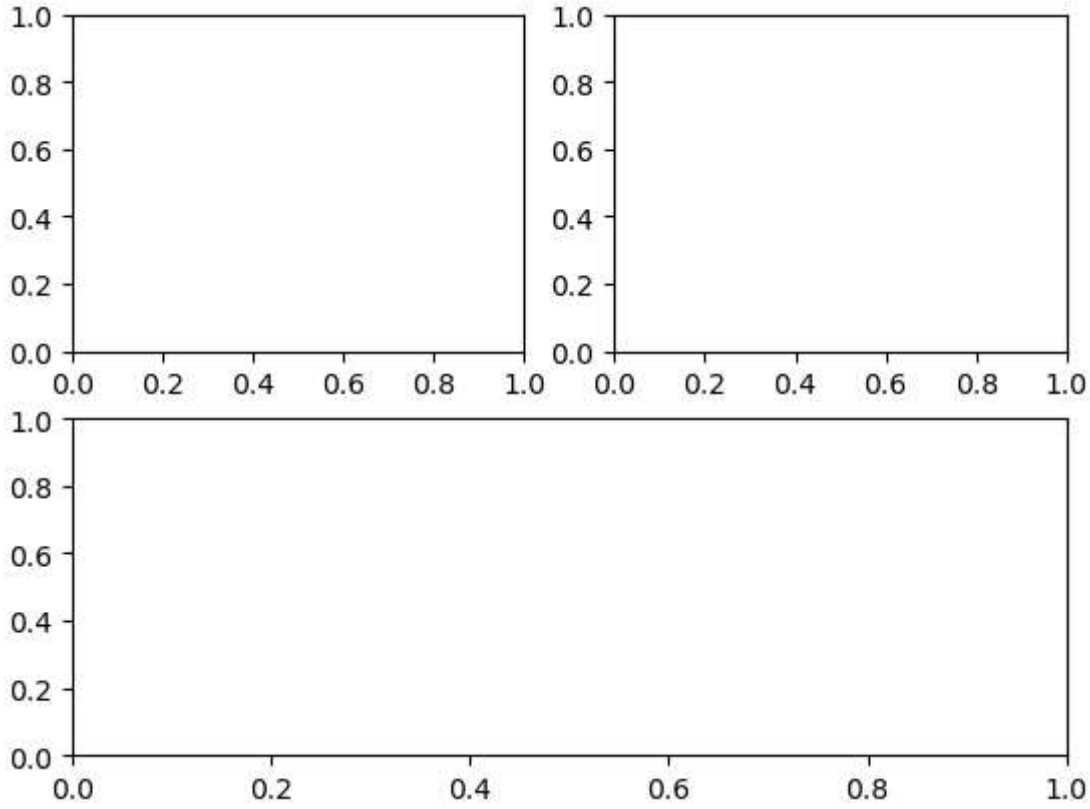
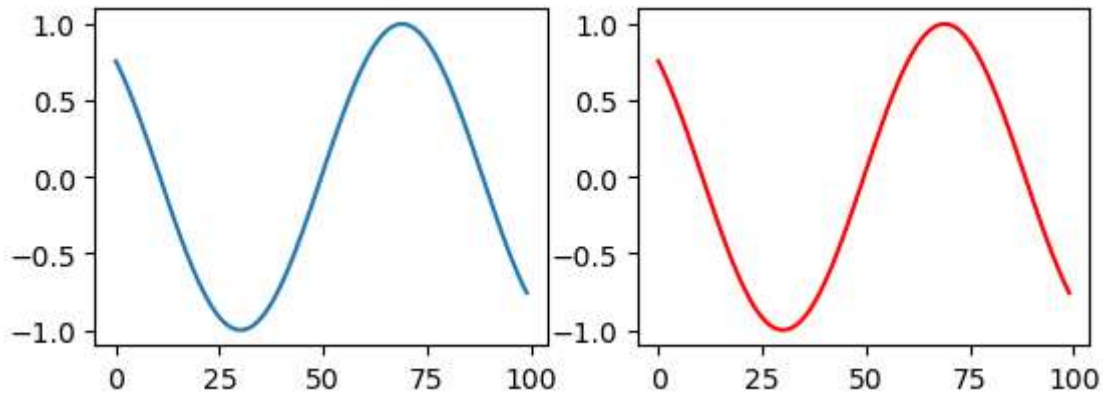
In [1]:
```
import matplotlib.pyplot as plt
import numpy as np
```

In [3]:
```
ax11 = plt.subplot(2, 2, 1)
ax21 = plt.subplot(2, 2, 2)
ax2 = plt.subplot(2, 1, 2)
```



In [5]:
```
sin = np.sin(np.linspace(-4, 4, 100))
plt.subplot(2, 2, 1)
plt.plot(sin)
plt.subplot(2, 2, 2)
plt.plot(sin, c='r')
```

Out[5]:
```
[<matplotlib.lines.Line2D at 0x1e2a5ae68b0>]
```

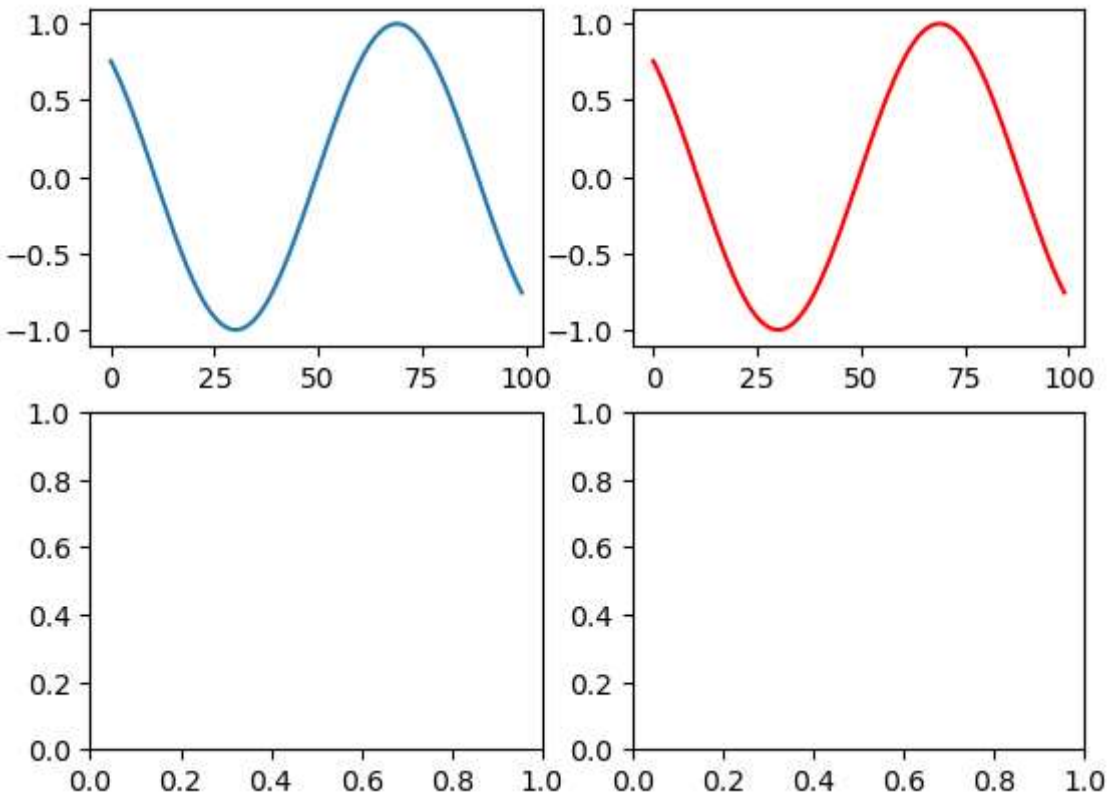


In [6]:
```
fig, axes = plt.subplots(2, 2)
axes[0, 0].plot(sin)
```

```
axes[0, 1].plot(sin, c='r')
```

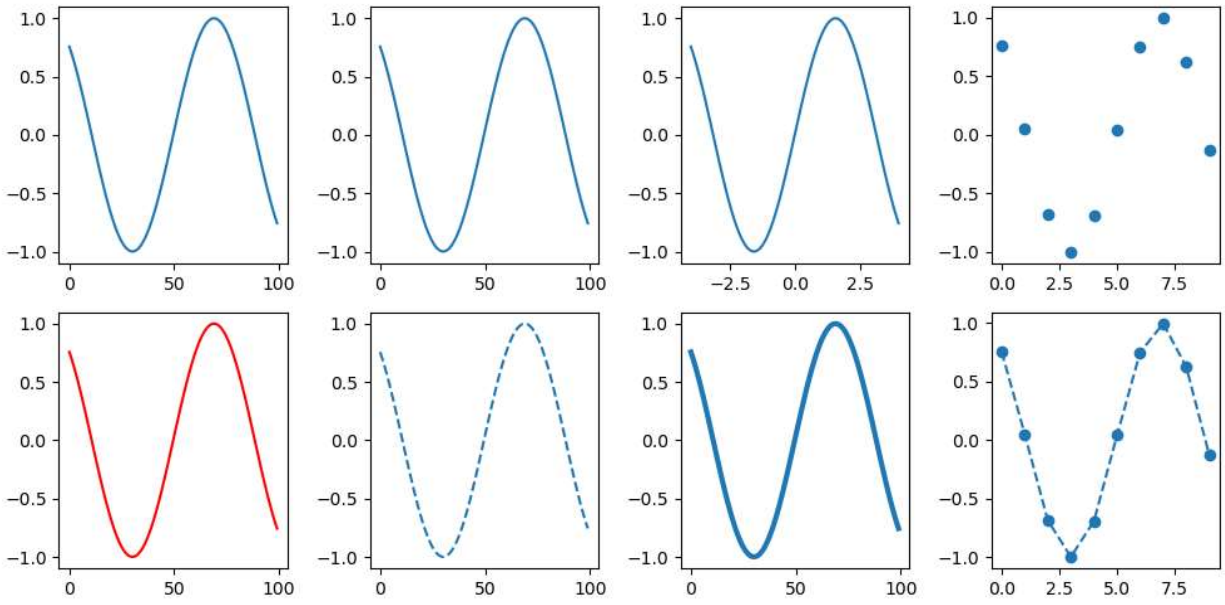
Out[6]:

```
[<matplotlib.lines.Line2D at 0x1e2a5d47700>]
```



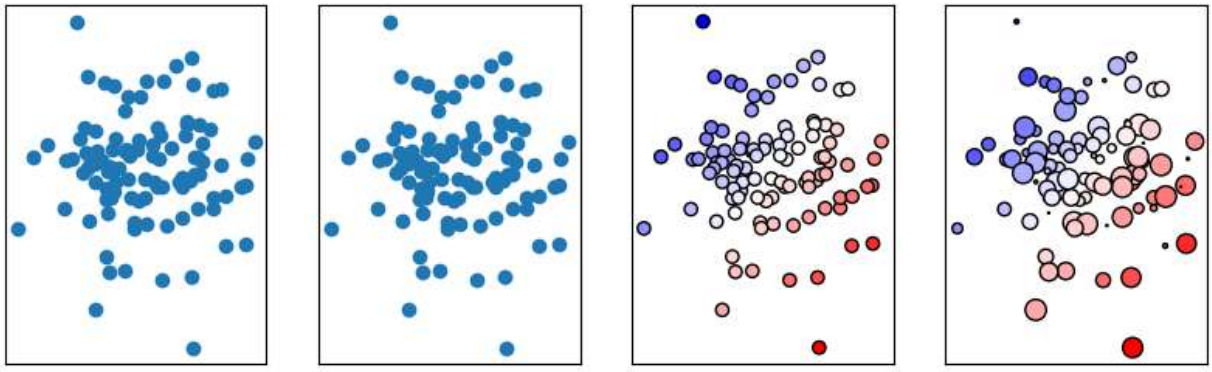
In [7]:

```
fig, ax = plt.subplots(2, 4, figsize=(10, 5))
ax[0, 0].plot(sin)
ax[0, 1].plot(range(100), sin)
ax[0, 2].plot(np.linspace(-4, 4, 100), sin)
ax[0, 3].plot(sin[::10], 'o')
ax[1, 0].plot(sin, c='r')
ax[1, 1].plot(sin, '--')
ax[1, 2].plot(sin, lw=3)
ax[1, 3].plot(sin[::10], '--o')
plt.tight_layout()
```



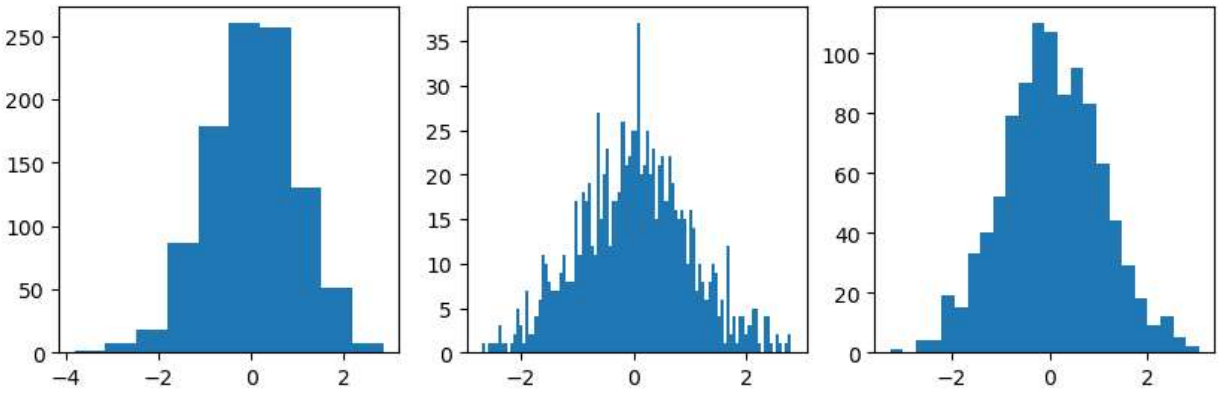
In [14]:
```
#Scatter Plots

np.random.seed(123)
x = np.random.randn(100)
y = np.random.randn(100)
sizes = np.random.rand(100) * 100

fig, ax = plt.subplots(1, 4, figsize=(10, 3),
                                        subplot_kw={'xticks': (), 'yticks': ()})
ax[0].plot(x, y, 'o')
ax[1].scatter(x, y)
ax[2].scatter(x, y, c=x-y, cmap='bwr', edgecolor='k')
ax[3].scatter(x, y, c=x-y, s=sizes, cmap='bwr', edgecolor='k')
```

Out[14]:
```
<matplotlib.collections.PathCollection at 0x1e2a62452e0>
```

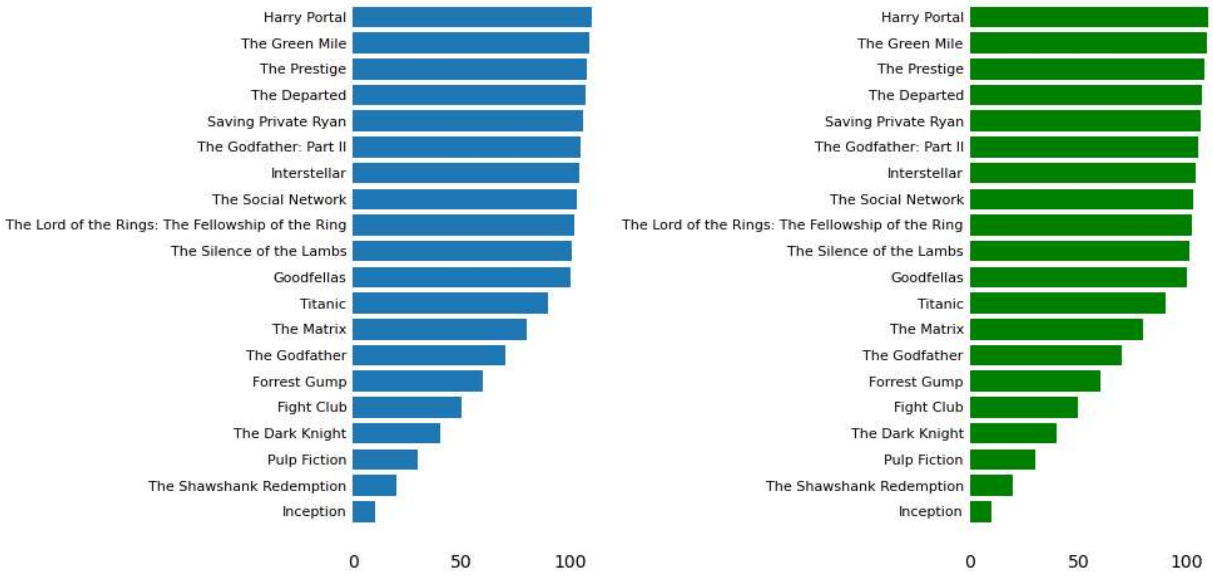


In [16]:
```
#Histogram

fig, ax = plt.subplots(1, 3, figsize=(10, 3))
ax[0].hist(np.random.normal(size=1000))
ax[1].hist(np.random.normal(size=1000), bins=100)
ax[2].hist(np.random.normal(size=1000), bins="auto")
```

Out[16]:
```
(array([  1.,   0.,   4.,   4.,  19.,  15.,  33.,  40.,  52.,  79.,  90.,
        110., 107.,  86.,  95.,  83.,  63.,  44.,  29.,  18.,   9.,  12.,
          5.,   2.]),
 array([-3.24387804, -2.98160167, -2.71932529, -2.45704892, -2.19477255,
        -1.93249617, -1.6702198 , -1.40794342, -1.14566705, -0.88339068,
        -0.6211143 , -0.35883793, -0.09656155,  0.16571482,  0.42799119,
         0.69026757,  0.95254394,  1.21482032,  1.47709669,  1.73937306,
         2.00164944,  2.26392581,  2.52620219,  2.78847856,  3.05075493]),
 <BarContainer object of 24 artists>)
```

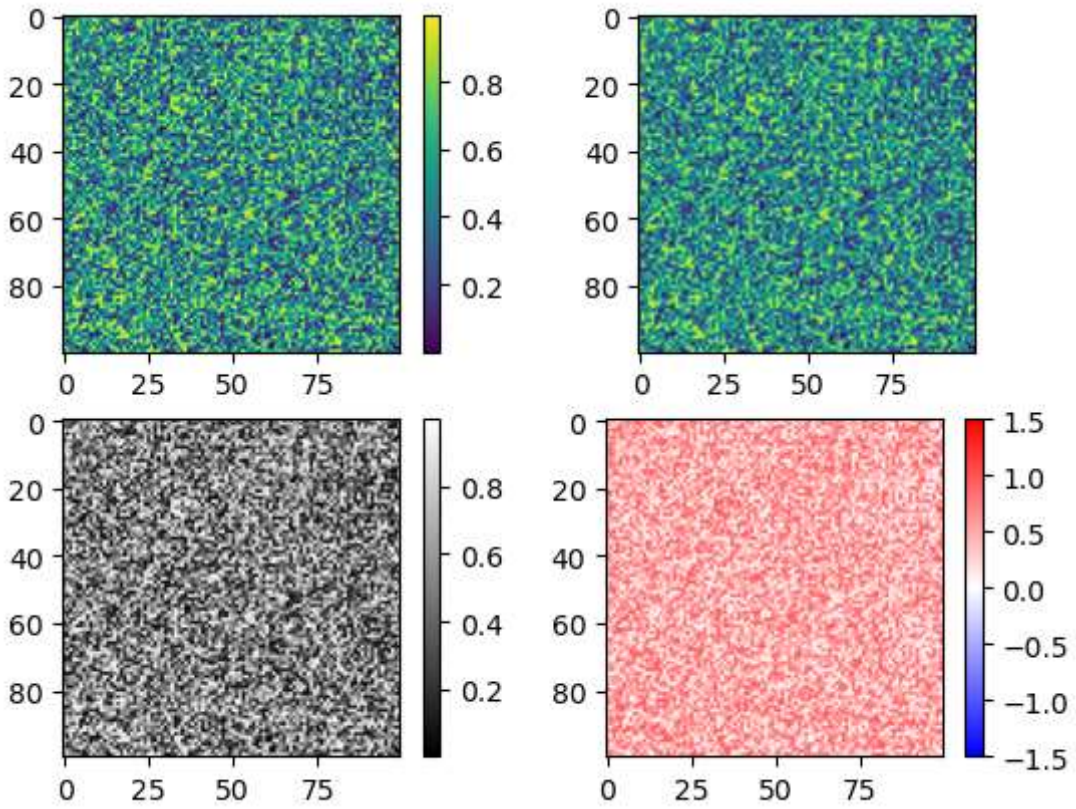


In [40]:
```
#Bar Chart

gross = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 101, 102, 103, 104, 105, 106, 107, 1
movie = ['Inception', 'The Shawshank Redemption', 'Pulp Fiction', 'The Dark Knight', '

fig, axs = plt.subplots(1, 2, figsize=(10, 5))
axs[0].barh(range(len(gross)), gross)
axs[0].set_yticks(range(len(gross)))
axs[0].set_yticklabels(movie, fontsize=8)
axs[0].set_frame_on(False)
axs[0].tick_params(length=0)

axs[1].barh(range(len(gross)), gross, color='green')
axs[1].set_yticks(range(len(gross)))
axs[1].set_yticklabels(movie, fontsize=8)
axs[1].set_frame_on(False)
axs[1].tick_params(length=0)

plt.tight_layout()
plt.show()
```

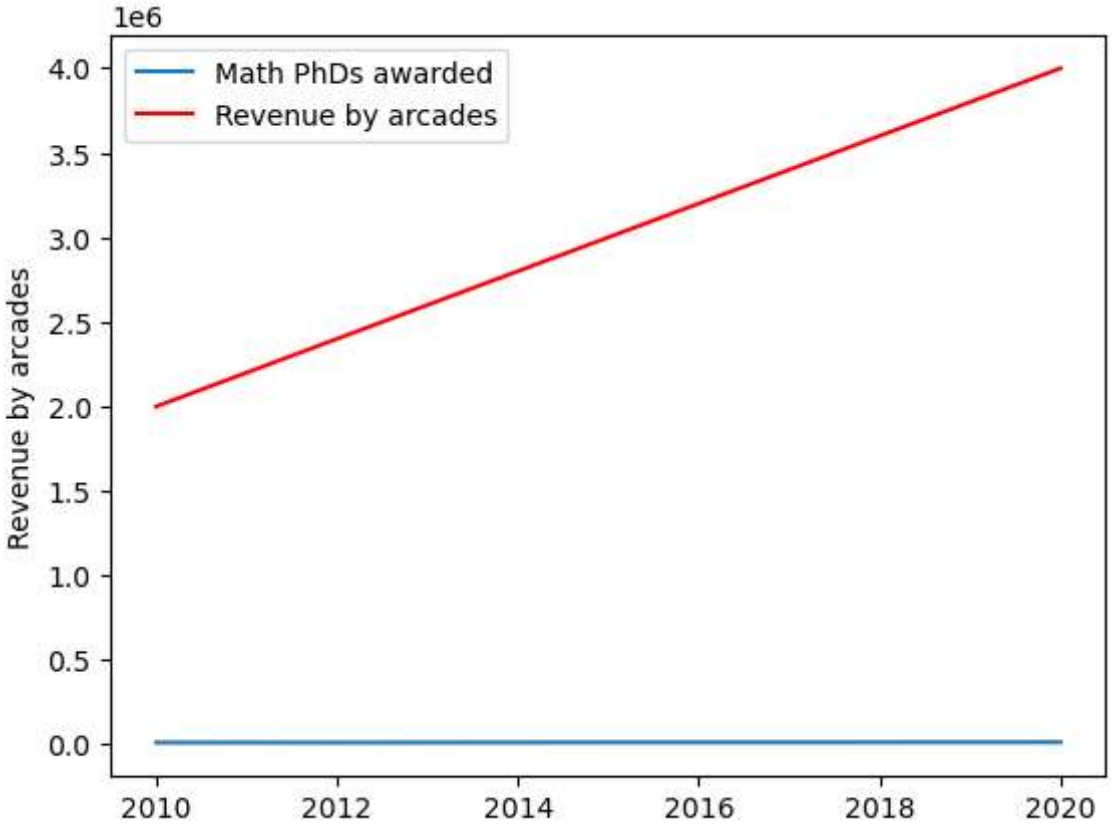


In [41]:
```
#Heatmaps

## create a random numpy array
arr = np.random.rand(100, 100)

## create a 2x2 grid of subplots
fig, ax = plt.subplots(2, 2)

## display the array in the first subplot and add a colorbar
im1 = ax[0, 0].imshow(arr)
plt.colorbar(im1, ax=ax[0, 0])

## display the array in the second subplot with bilinear interpolation
ax[0, 1].imshow(arr, interpolation='bilinear')

## display the array in the third subplot with gray colormap and add a colorbar
im3 = ax[1, 0].imshow(arr, cmap='gray')
plt.colorbar(im3, ax=ax[1, 0])
```

```
## display the array in the fourth subplot with bwr colormap and custom vmin/vmax valu
im4 = ax[1, 1].imshow(arr, cmap='bwr', vmin=-1.5, vmax=1.5)
plt.colorbar(im4, ax=ax[1, 1])

## show the plot
plt.show()
```



In [44]:
```
years = [2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020]
phds = [12000, 12200, 11800, 12400, 12800, 13200, 13400, 13600, 13800, 14000, 14200]
revenue = [2000000, 2200000, 2400000, 2600000, 2800000, 3000000, 3200000, 3400000, 360

fig, ax = plt.subplots()
ax.plot(years, phds, label="Math PhDs awarded")
ax.plot(years, revenue, c='r', label="Revenue by arcades")
ax.set_ylabel("Math PhDs awarded")
ax.set_ylabel("Revenue by arcades")
ax.legend()

plt.show()
```


1e6
Math PhDs awarded
Revenue by arcades
Revenue by arcades
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
2010
2012
2014
2016
2018
2020

In [ ]: