Example Document

This is an example notebook to try out the "Notebook as PDF" extension. It contains a few plots from the excellent matplotlib gallery.

To try out the extension click "File -> Download as -> PDF via HTML". This will convert this notebook into a PDF. This extension has three new features compared to the official "save as PDF" extension:

- it produces a PDF with the smallest number of page breaks,
- the original notebook is attached to the PDF; and
- this extension does not require LaTex.

The created PDF will have as few pages as possible, in many cases only one. This is useful if you are exporting your notebook to a PDF for sharing with others who will view them on a screen.

To make it easier to reproduce the contents of the PDF at a later date the original notebook is attached to the PDF. Not all PDF viewers know how to deal with attachments. This mean you need to use Acrobat Reader or pdf.js to be able to get the attachment from the PDF. Preview for OSX does not know how to display/give you access to PDF attachments.

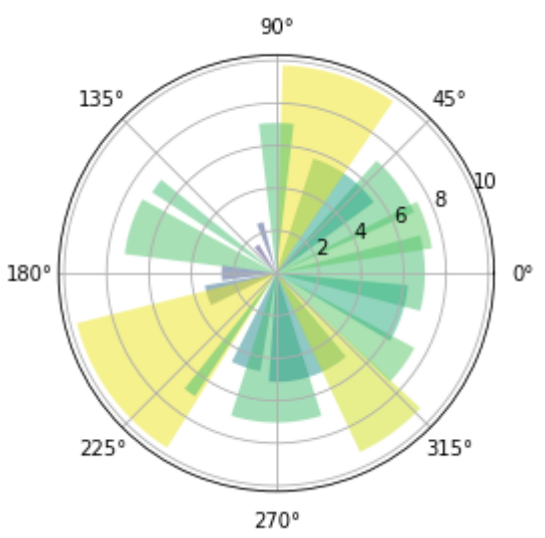
In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:
```
# Fixing random state for reproducibility
np.random.seed(19680801)

# Compute pie slices
N = 20
theta = np.linspace(0.0, 2 * np.pi, N, endpoint=False)
radii = 10 * np.random.rand(N)
width = np.pi / 4 * np.random.rand(N)
colors = plt.cm.viridis(radii / 10.)

ax = plt.subplot(111, projection='polar')
ax.bar(theta, radii, width=width, bottom=0.0, color=colors, alpha=0.5)
```

Out[2]: <BarContainer object of 20 artists>



Below we show some more lines that go up and go down. These are noisy lines because we use a random number generator to create them. Fantastic isn't it?

In [3]:
```
x = np.linspace(0, 10)

# Fixing random state for reproducibility
np.random.seed(19680801)

fig, ax = plt.subplots()

ax.plot(x, np.sin(x) + x + np.random.randn(50))
ax.plot(x, np.sin(x) + 0.5 * x + np.random.randn(50))
ax.plot(x, np.sin(x) + 2 * x + np.random.randn(50))
ax.plot(x, np.sin(x) - 0.5 * x + np.random.randn(50))
ax.plot(x, np.sin(x) - 2 * x + np.random.randn(50))
ax.plot(x, np.sin(x) + np.random.randn(50));
```