Software Carpentry Demo - June 7, 2012

Note: IPython is running here in `--pylab` mode so I didn't have to import matplotlib.

In [1]:
```
%config InlineBackend.figure_format = 'svg'
```

In [2]:
```
import numpy as np
import scipy.stats as spst
import scipy.optimize as spop
import scipy.interpolate as spint
```

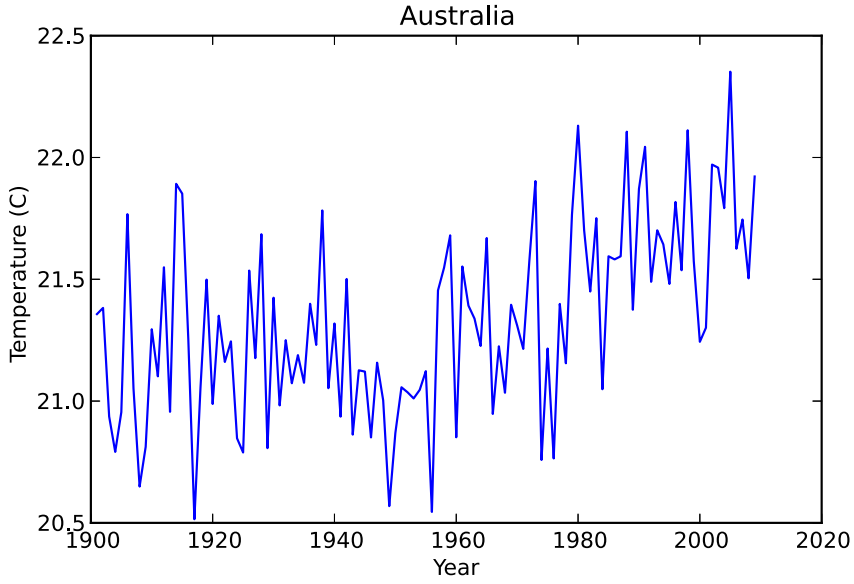
Load the data and take an initial look.

http://docs.scipy.org/doc/numpy/reference/generated/numpy.loadtxt.html

In [3]:
```
year, aus, can = np.loadtxt('temperatures.txt', delimiter=',',
                            skiprows=1, unpack=True)
```

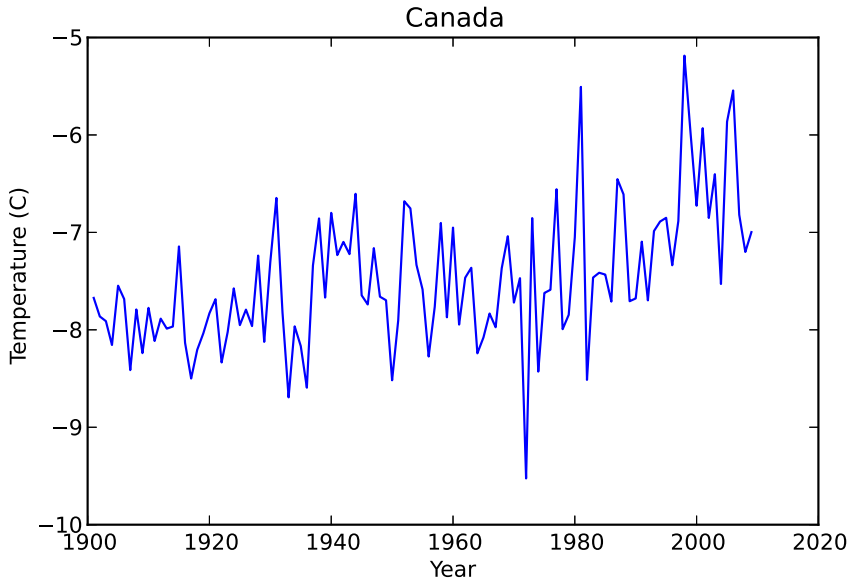
In [4]:
```
plot(year, aus)
title('Australia')
xlabel('Year')
ylabel('Temperature (C)')
```

Out[4]: <matplotlib.text.Text at 0x1057453d0>



In [5]:
```
plot(year, can)
title('Canada')
xlabel('Year')
ylabel('Temperature (C)')
```

Out[5]:
```
<matplotlib.text.Text at 0x1058a67d0>
```



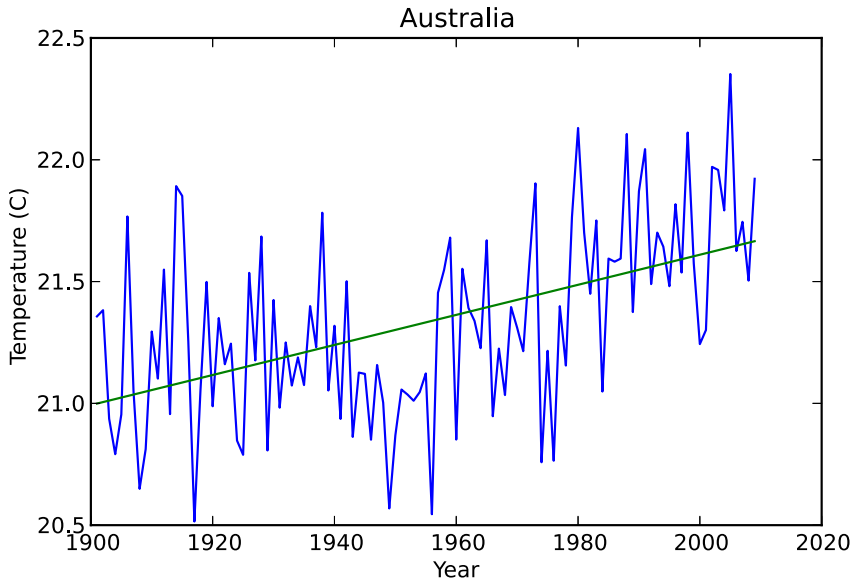
Fit a line to the Australia temperatures using scipy.stats.linregress: http://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.linregress.html

In [6]:
```
aus_line = spst.linregress(year, aus)
```

In [7]:
```
aus_line_pts = aus_line[0] * year + aus_line[1]
```

In [8]:
```
plot(year, aus)
plot(year, aus_line_pts)
title('Australia')
xlabel('Year')
ylabel('Temperature (C)')
```

Out[8]: <matplotlib.text.Text at 0x105b0f650>



Fit a line to the Canada data using scipy.optimize.curve_fit: http://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.curve_fit.html

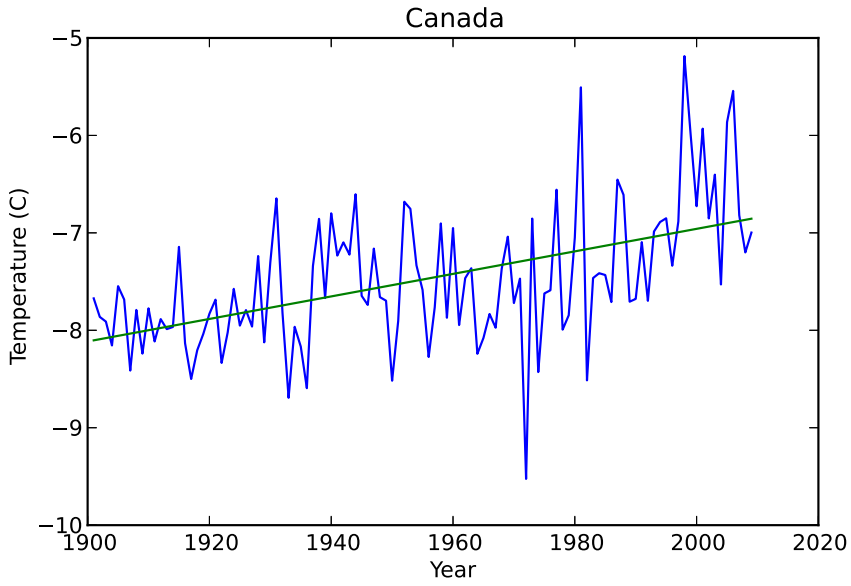
In [9]:
```
def linear_func(year, slope, intercept):
    return slope * year + intercept
```

In [10]:
```
can_line, can_cov = spop.curve_fit(linear_func, year, can)
```

In [11]:
```
can_line_pts = linear_func(year, can_line[0], can_line[1])
```

In [12]:
```
plot(year, can)
plot(year, can_line_pts)
title('Canada')
xlabel('Year')
ylabel('Temperature (C)')
```

Out[12]: <matplotlib.text.Text at 0x105b5c0d0>



In [13]:
```
spst.pearsonr(year, can)
```

Out[13]: (0.51195782671512846, 1.2713276823278399e-08)

Fit a spline to the Australia data using scipy.interpolate.UnivariateSpline: http://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.UnivariateSpline.html

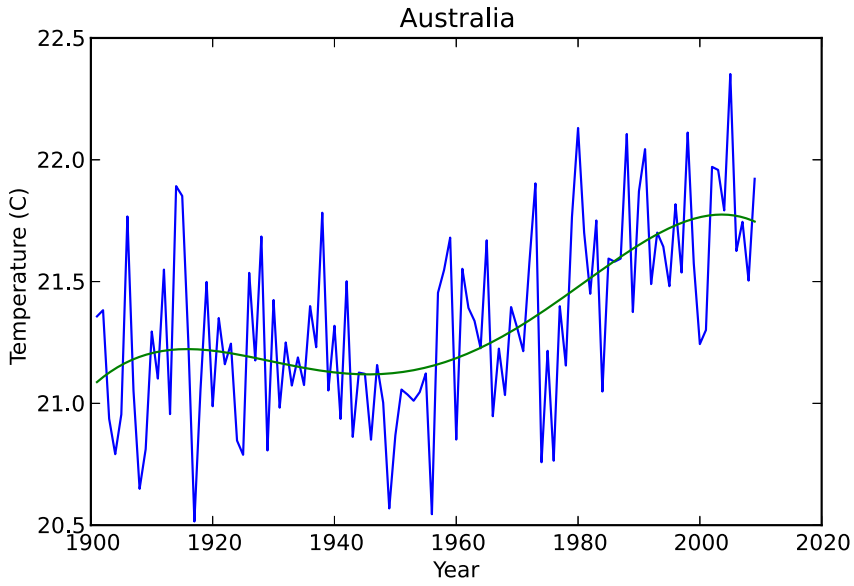
In [14]:
```
spline = spint.UnivariateSpline(year, aus, k=5)
```

In [15]:
```
x = np.linspace(year.min(), year.max(), 1000)
```

In [16]:
```
y = spline(x)
```

In [17]:
```
plot(year, aus)
plot(x, y)
title('Australia')
xlabel('Year')
ylabel('Temperature (C)')
```

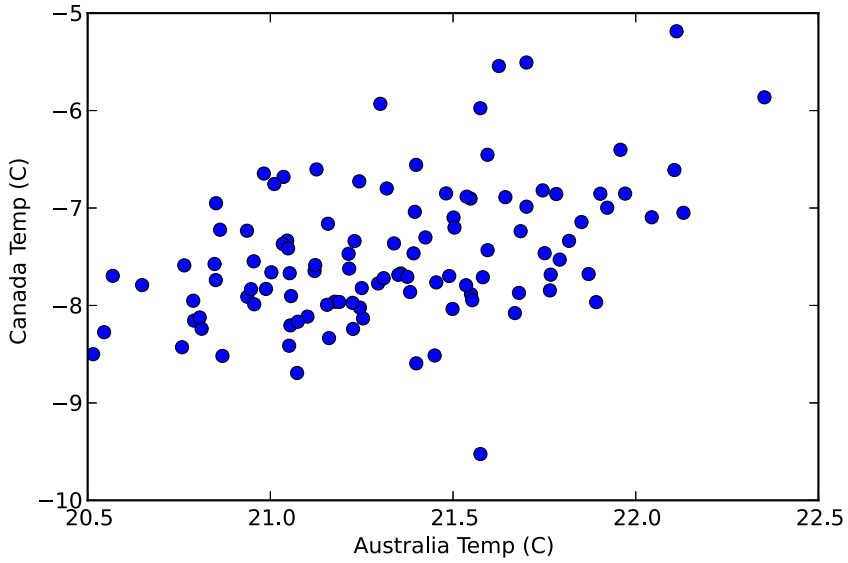
Out[17]: <matplotlib.text.Text at 0x105b9b690>



Plot the temperatures against each other and fit a line.

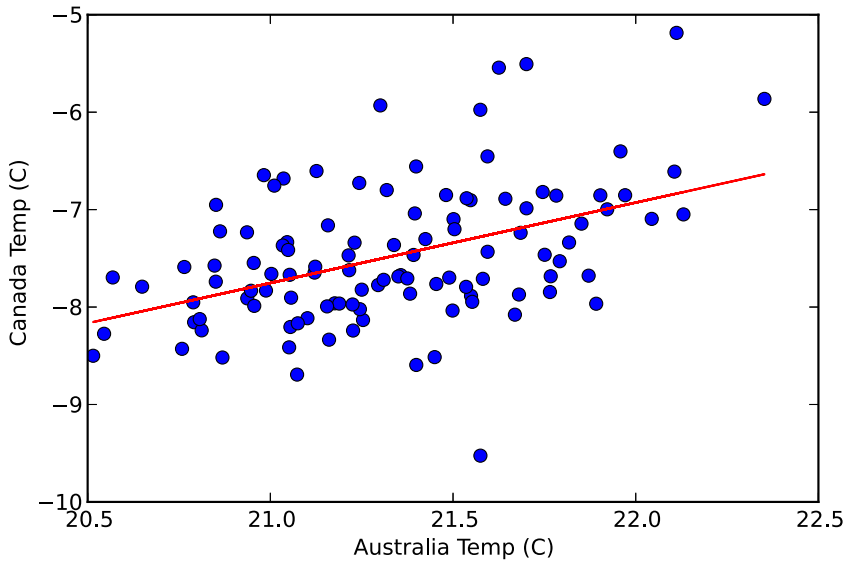
In [18]:
```
plot(aus, can, 'bo')
xlabel('Australia Temp (C)')
ylabel('Canada Temp (C)')
```

Out[18]: <matplotlib.text.Text at 0x105bd3710>



In [19]:
```
new_line = spst.linregress(aus, can)
new_line_pts = new_line[0] * aus + new_line[1]
plot(aus, can, 'bo')
plot(aus, new_line_pts, 'r-')
xlabel('Australia Temp (C)')
ylabel('Canada Temp (C)')
```

Out[19]:
```
<matplotlib.text.Text at 0x105bd5510>
```



In [20]:
```
new_line
```

Out[20]:
```
(0.82556611933357937,
 -25.090352153252788,
 0.44948681311947258,
 9.4409823433234464e-07,
 0.15861125998986195)
```

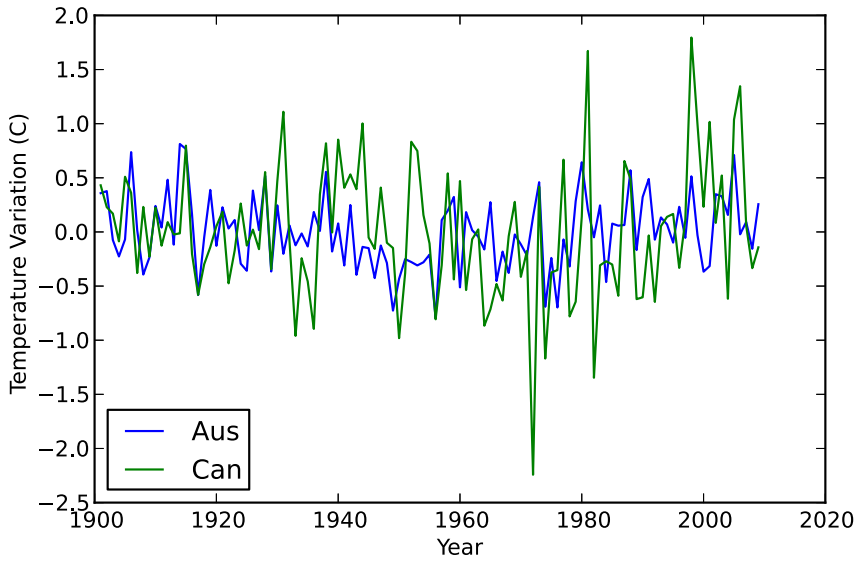
Plot the temperatures as variations about their respective best fit lines.

In [21]:
```
aus_sub = aus - aus_line_pts
```

In [22]:
```
can_sub = can - can_line_pts
```

In [23]:
```
plot(year, aus_sub, label='Aus')
plot(year, can_sub, label='Can')
xlabel('Year')
ylabel('Temperature Variation (C)')
legend(loc='lower left')
```

Out[23]: <matplotlib.legend.Legend at 0x106170f90>



In [23]: