Maximum Likelihood

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy import stats
from scipy.optimize import differential_evolution
%matplotlib inline
```

Read in data

In [3]:
```
data = pd.read_fwf("globalTemp.txt",na_values=["*"])
data = data.dropna()
```

Do minimization

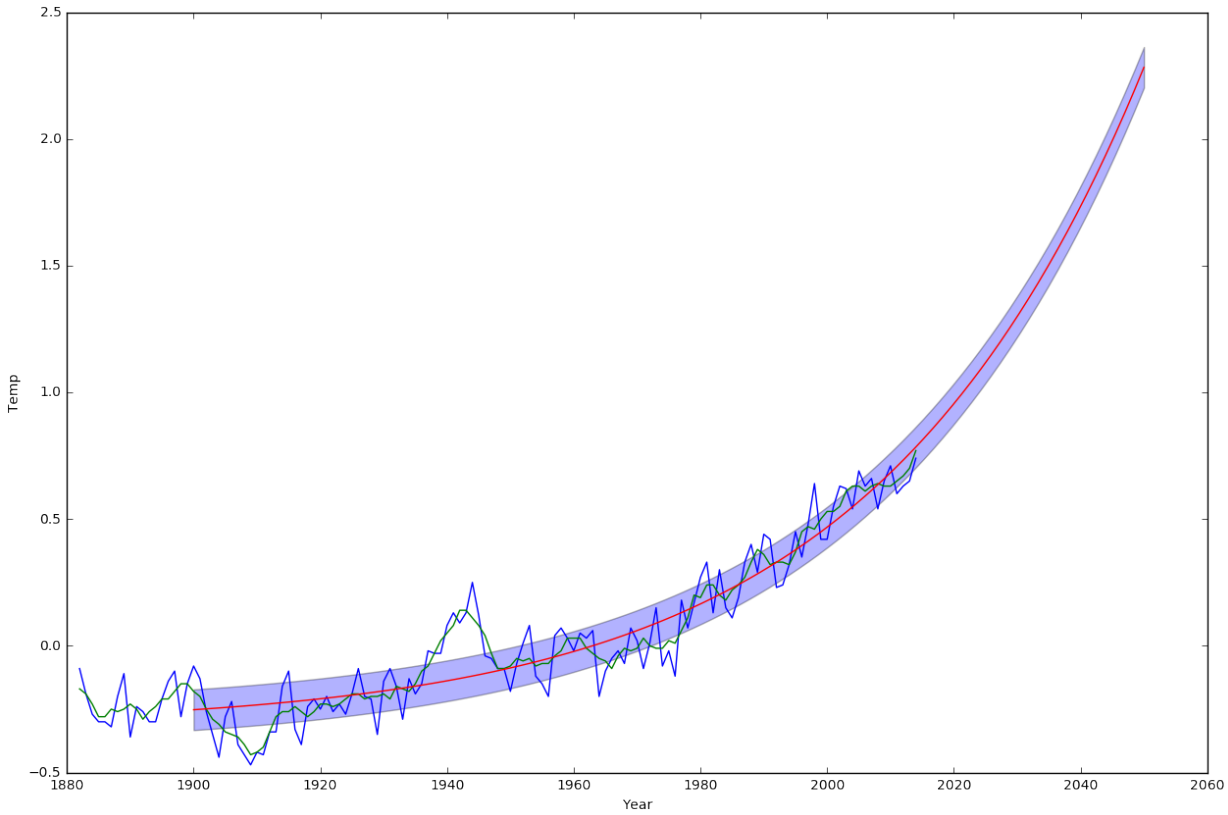
Use a method suited to a constrained search looking for a global minimum of the negative log-likelihood of observing each data point assuming each data point is taken from a normal distribution centered at the moving average of the temperature. We also assume that the standard deviation of these points is constant throughout time.

In [4]:
```
# Define the model function from assignment 4a - sigmoid
def modelFunc(args):
    (a,b,c,d,sigma) = args
    predTemps = d+(a-d)/(1+(np.array(data["Year"])/c)**b)
    # Minimize negative log likelihood (same as maximizing the log lik
elihood) assuming data is from a normal distribution
    return -np.sum(stats.norm.logpdf(np.array(data["Avg"]), loc=predTe
mps, scale=sigma))

# Use bounds from assignment
bounds=[(-.4,-.2),(40,80), (2000,3000), (300000,400000),(0,2)]

results = differential_evolution(modelFunc, bounds, maxiter=100000)
(a,b,c,d,sigma) = results.x
print("a={}, b={}, c={}, d={}, sigma={}".format(a,b,c,d,sigma))
```

```
a=-0.317987337794, b=48.453800406, c=2615.62587032, d=348742.48036,
sigma=0.0802055246725
```

In [7]:
```
fig, ax = plt.subplots(1,figsize=(15,10),dpi=100)
years = np.linspace(1900,2050,100)
pred = d+(a-d)/(1+(np.array(years)/c)**b)
plt.plot(data["Year"], data["Temp"])
plt.plot(data["Year"], data["Avg"])
plt.plot(years, pred)
plt.xlabel("Year")
plt.ylabel("Temp")
ax.fill_between(years, pred+sigma, pred-sigma, facecolor='blue', alpha
=0.3)
```

Out[7]: <matplotlib.collections.PolyCollection at 0x1116bbb10>



This looks like a good fit to the data. But I'm not certain about the overlap of +- 1 sigma as a confidence interval, that seems rather ad-hoc.