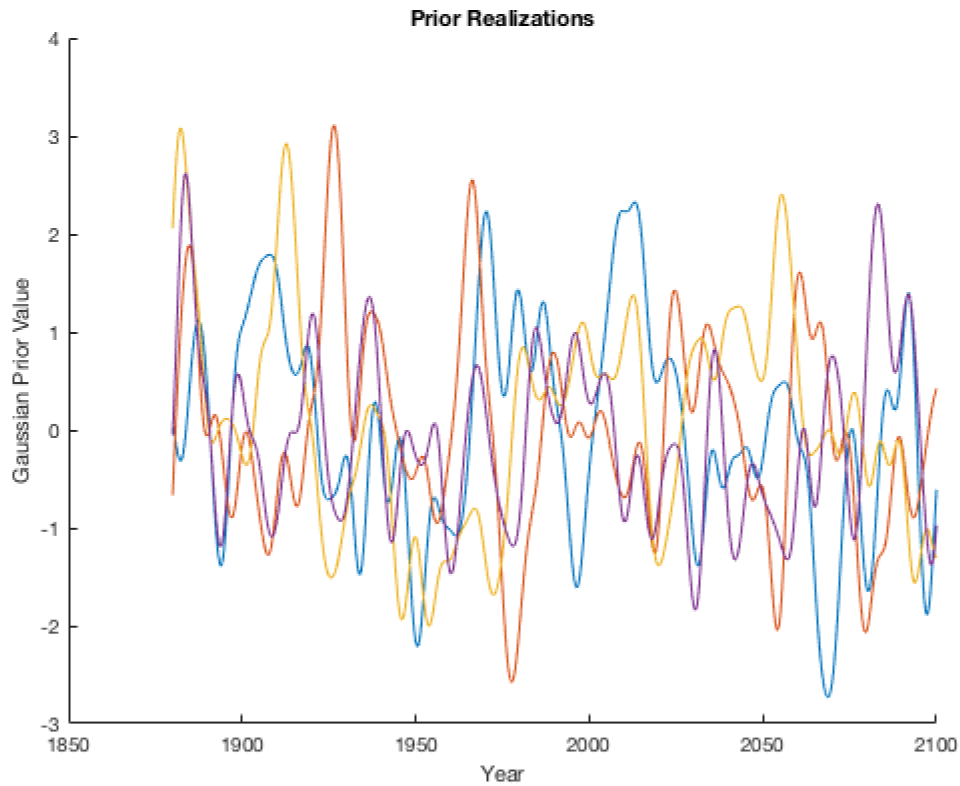
```
load('temperature_data.mat', 't')

%{
1.
We optimize the following problem to obtain a linear regresion model:

J = Sum(t_i - w*x_i - b)^2

Taking the derivative with respect to w and b and setting the
 equations to
0 leads to


b_hat = t_mean - w_hat*x_mean
w_hat = (n*sum(x_i*t_i) - sum(x_i)sum(t_i))/(n*sum(x_i^2) -
 (sum(x_i))^2)

%}
num = 137*x'*t - sum(x)*sum(t);
denom = 137*(x'*x) - sum(x)^2;

w_hat = num/denom;

b_hat = mean(t) - w_hat*mean(x);

figure;
hold on
scatter(x, t);
plot(x, w_hat*x' + b_hat);
hold off
title('Linear Regression');
xlabel('Year')
legend('Data', 'Fitted Line');
```


```
%2.
z = 1880:1:2100;

figure;
hold on
scatter(x, t);
plot(z, w_hat*z + b_hat);
hold off
xlabel('Year')

title('Linear Regression');
legend('Data', 'Fitted Line');


%{

The predicted values for 2017-2100 are not good predictions because
 the
most recent trend(2000-2016) indicates that the values(temperatures)
 are
increasing more rapidly than before. Our predictions fail to capture
this non-linear trend and thus, a linear regression model is not
 proper for
this type of data.
%}
```

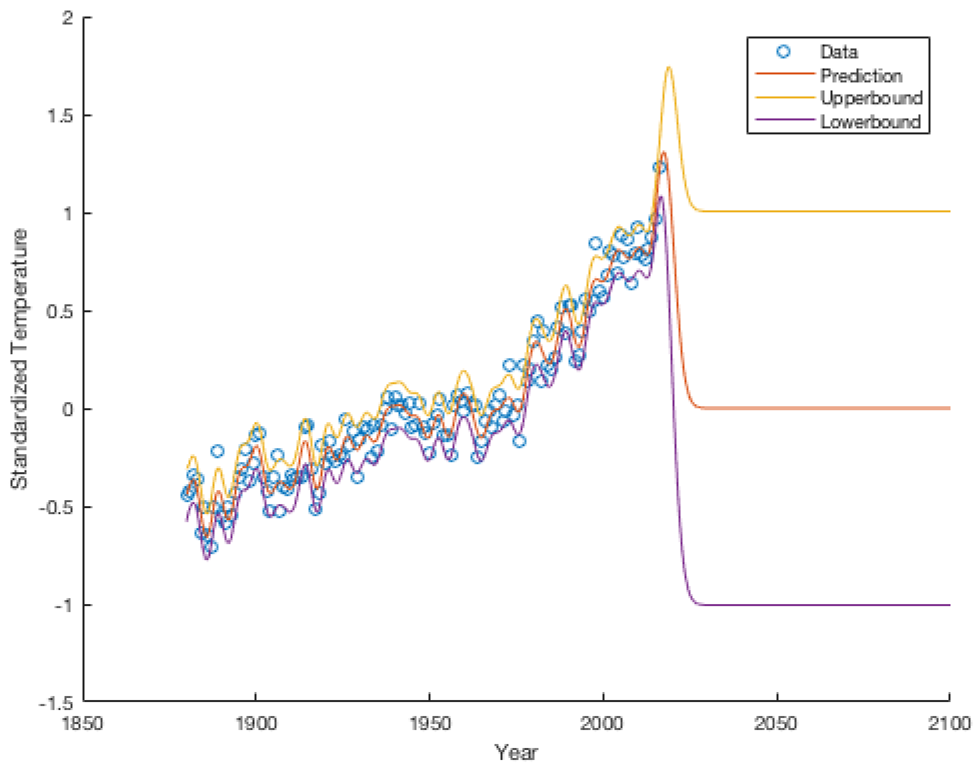
```
%3. Kernel Function
```


```
%4.

K = zeros(1000, 1000);
tau = 10;
samples = linspace(1880, 2100, 1000);
for i = 1:1000
    for j = 1:1000
        K(i, j) = kernel(samples(i), samples(j), tau);
    end
end

rng default;
figure;
for i = 1:4
    hold on
    y = mvnrnd(zeros(1, 1000), K);
    plot(samples, y);
end
xlabel('Year')
ylabel('Gaussian Prior Value')
title('Prior Realizations')
hold off
```




```
%5.
sigma = 0.01;

Kern = zeros(137, 137);
for i = 1:137
    for j = 1:137
         Kern(i, j) = kernel(x(i), x(j), tau);
    end
end
format long;
C = Kern + sigma*eye(137);
C_inv = inv(C);
k = zeros(1, 137);
mu = zeros(1000, 1);
sig = zeros(1000, 1);
for i = 1:1000
    for j = 1:137
        k(j) = kernel(samples(i), x(j), tau);
    end
    c = kernel(samples(i), samples(i), tau) + sigma;
    mu(i) = k*C_inv*t;
    sig(i) = c - k*C_inv*k';
end

figure
hold on
```

```
scatter(x, t);
plot(samples, mu);
plot(samples, mu + sqrt(sig));
plot(samples, mu - sqrt(sig));
hold off
xlabel('Year')
ylabel('Standardized Temperature')
legend('Data', 'Prediction', 'Upperbound', 'Lowerbound');

%{
The obtained distribution provides a very good estimate of the data,
 but
does not have an accurate prediction of future years. This can be seen
 in
the large upper and lower bounds around our prediction line. There is
 a lot
of uncertainity after 2016, and also we see that our prediction line
becomes constant after 2020 because the kernel function approaches 0
 as the
difference between our future years and prior years increases.

%}
```



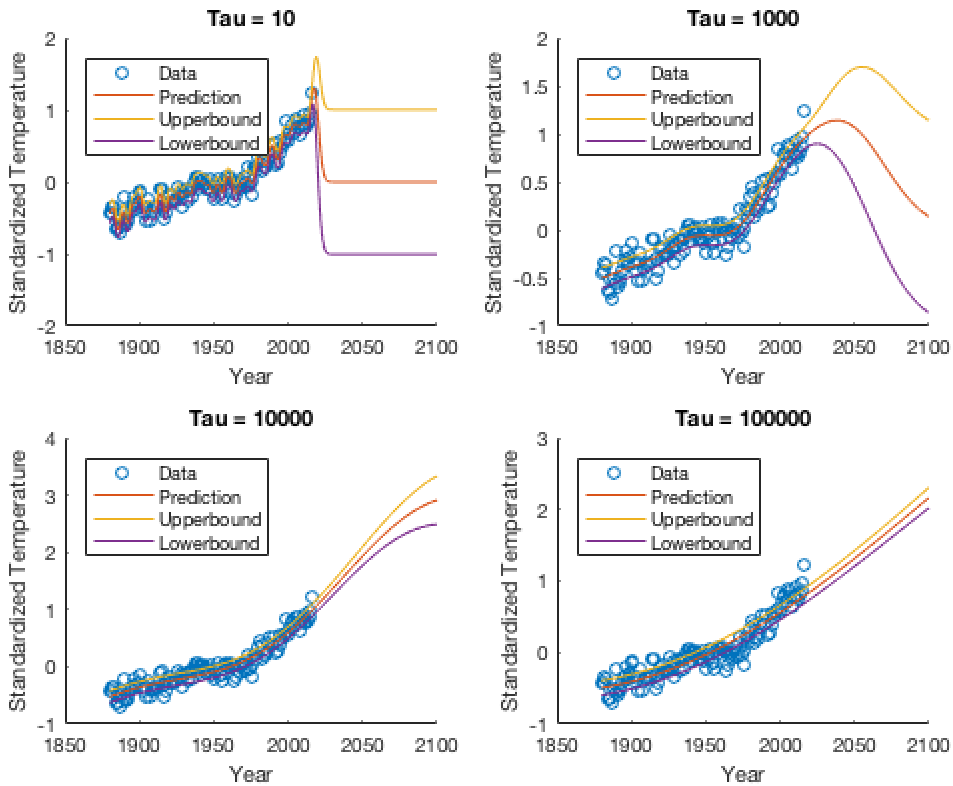

```
%6.

%{
```

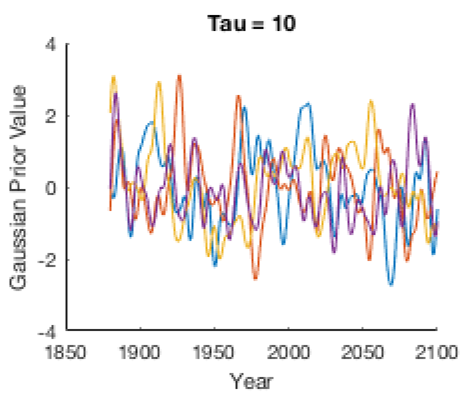
```
We are going to repeat what we did in 5 but with different values of
 tau to
see what role tau plays in our predictive distribution.
%}
%Plot with tau = 10
figure
subplot(2,2,1);
hold on
scatter(x, t);
plot(samples, mu);
plot(samples, mu + sqrt(sig));
plot(samples, mu - sqrt(sig));
hold off
xlabel('Year')
ylabel('Standardized Temperature')
legend('Data', 'Prediction', 'Upperbound', 'Lowerbound', 'Location', 'Northwest');
title('Tau = 10');

%Tau = 1000
tau = 1000;
for i = 1:137
    for j = 1:137
         Kern(i, j) = kernel(x(i), x(j), tau);
    end
end
format long;
C = Kern + sigma*eye(137);
C_inv = inv(C);
for i = 1:1000
    for j = 1:137
        k(j) = kernel(samples(i), x(j), tau);
    end
    c = kernel(samples(i), samples(i), tau) + sigma;
    mu(i) = k*C_inv*t;
    sig(i) = c - k*C_inv*k';
end

subplot(2,2,2);
hold on
scatter(x, t);
plot(samples, mu);
plot(samples, mu + sqrt(sig));
plot(samples, mu - sqrt(sig));
hold off
xlabel('Year')
ylabel('Standardized Temperature')
legend('Data', 'Prediction', 'Upperbound', 'Lowerbound','Location', 'Northwest');
title('Tau = 1000');

%Tau = 10000
tau = 10000;
for i = 1:137
    for j = 1:137
         Kern(i, j) = kernel(x(i), x(j), tau);
```

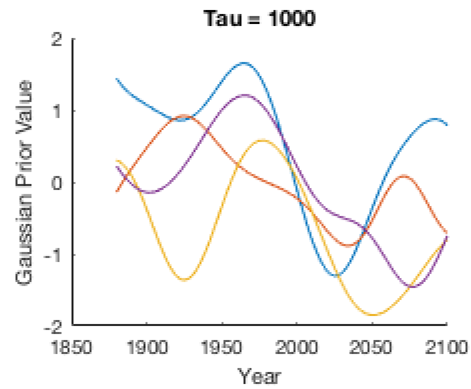
```
    end
end
format long;
C = Kern + sigma*eye(137);
C_inv = inv(C);

for i = 1:1000
    for j = 1:137
        k(j) = kernel(samples(i), x(j), tau);
    end
    c = kernel(samples(i), samples(i), tau) + sigma;
    mu(i) = k*C_inv*t;
    sig(i) = c - k*C_inv*k';
end

subplot(2,2,3);
hold on
scatter(x, t);
plot(samples, mu);
plot(samples, mu + sqrt(sig));
plot(samples, mu - sqrt(sig));
hold off
xlabel('Year')
ylabel('Standardized Temperature')
legend('Data', 'Prediction', 'Upperbound', 'Lowerbound', 'Location', 'Northwest');
title('Tau = 10000');

%Tau = 100000
tau = 100000;
for i = 1:137
    for j = 1:137
         Kern(i, j) = kernel(x(i), x(j), tau);
    end
end
format long;
C = Kern + sigma*eye(137);
C_inv = inv(C);

for i = 1:1000
    for j = 1:137
        k(j) = kernel(samples(i), x(j), tau);
    end
    c = kernel(samples(i), samples(i), tau) + sigma;
    mu(i) = k*C_inv*t;
    sig(i) = c - k*C_inv*k';
end

subplot(2,2,4);
hold on
scatter(x, t);
plot(samples, mu);
plot(samples, mu + sqrt(sig));
plot(samples, mu - sqrt(sig));
hold off
```

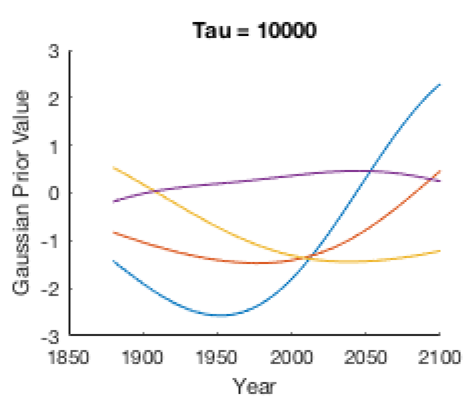
```
xlabel('Year')
ylabel('Standardized Temperature')
legend('Data', 'Prediction', 'Upperbound', 'Lowerbound', 'Location', 'Northwest');
title('Tau = 100000');


tau = 10;
for i = 1:1000
    for j = 1:1000
        K(i, j) = kernel(samples(i), samples(j), tau);
    end
end
rng default;
figure;
subplot(2,2,1);
for i = 1:4
    hold on
    y = mvnrnd(zeros(1, 1000), K);
    plot(samples, y);
end
xlabel('Year')
ylabel('Gaussian Prior Value')
title('Tau = 10')
hold off

tau = 1000;
for i = 1:1000
    for j = 1:1000
        K(i, j) = kernel(samples(i), samples(j), tau);
    end
end
rng default;
figure;
subplot(2,2,2);
for i = 1:4
    hold on
    y = mvnrnd(zeros(1, 1000), K);
    plot(samples, y);
end
xlabel('Year')
ylabel('Gaussian Prior Value')
title('Tau = 1000')
hold off

tau = 10000;
for i = 1:1000
    for j = 1:1000
        K(i, j) = kernel(samples(i), samples(j), tau);
    end
end
rng default;
figure;
subplot(2,2,3);
for i = 1:4
```

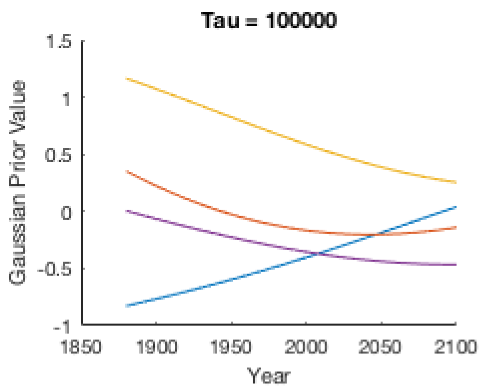
```
    hold on
    y = mvnrnd(zeros(1, 1000), K);
    plot(samples, y);
end
xlabel('Year')
ylabel('Gaussian Prior Value')
title('Tau = 10000')
hold off

tau = 100000;
for i = 1:1000
    for j = 1:1000
        K(i, j) = kernel(samples(i), samples(j), tau);
    end
end
rng default;
figure;
subplot(2,2,4);
for i = 1:4
    hold on
    y = mvnrnd(zeros(1, 1000), K);
    plot(samples, y);
end
xlabel('Year')
ylabel('Gaussian Prior Value')
title('Tau = 100000')
hold off

%{
We see that as we increase tau squared, our extrapolated values
 approach 0
a lot slower. Thus, a greater tau (up to some point) would give future
 years
that are far from our prior years more importance. Also, note in our
 plot
of the priors we have less variablility in our priors as tau
 increases, and
hence less overfitting. In addition, when the numerator is greater
 than tau
in our kernel function, we have values that are closer to 0. In other
 words,
tau can be seen as a way of tuning our predictions for years that are
 far
from our prior data points. Thus, we can interpret tau as a parameter
 for
overfitting and also distance measure for predictive accuracy.

%}
```


Tau = 10
Tau = 1000
Tau = 10000
Tau = 100000
Data
Prediction
Upperbound
Lowerbound
Standardized Temperature
Year


Tau = 10
Gaussian Prior Value
Year

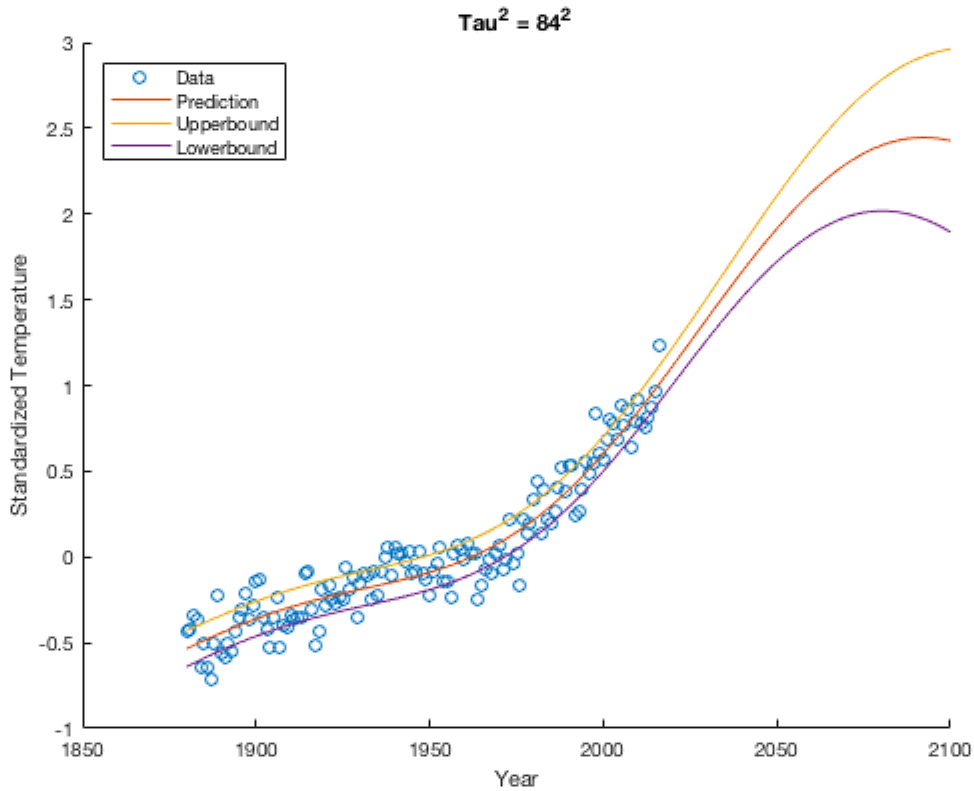

Tau = 1000
Gaussian Prior Value
Year
2
1
0
-1
-2
1850
1900
1950
2000
2050
2100


Tau = 10000
Gaussian Prior Value
Year
3
2
1
0
-1
-2
-3
1850
1900
1950
2000
2050
2100




```
%7.
%{
It is going to be hard to extrapolate data for values that satisfy
 this condition
because our kernel function approaches 0 for values whose difference
 with
the priors is greater than tau (such as the year 2100). Thus, as the
 kernel
function approaches 0, our mean also approaches 0, leaving us with
 only our
variance interval which is not good to use as our predictor since it's
based on all of the prior values. Thus, I reason that it makes most
 sense
to tune tau squared to be the squared distance from our target year to
 our
last data point (2016). Thus, for target year 2100, it is best to use
tau_squared = 84^2. This is the squared difference between 2016 and
 2100.
Let's look at our resulting plot.

%}


%Tau = 84^2
tau = 84^2;
for i = 1:137
```

```
    for j = 1:137
         Kern(i, j) = kernel(x(i), x(j), tau);
    end
end
format long;
C = Kern + sigma*eye(137);
C_inv = inv(C);

for i = 1:1000
    for j = 1:137
        k(j) = kernel(samples(i), x(j), tau);
    end
    c = kernel(samples(i), samples(i), tau) + sigma;
    mu(i) = k*C_inv*t;
    sig(i) = c - k*C_inv*k';
end

figure
hold on
scatter(x, t);
plot(samples, mu);
plot(samples, mu + sqrt(sig));
plot(samples, mu - sqrt(sig));
hold off
xlabel('Year')
ylabel('Standardized Temperature')
legend('Data', 'Prediction', 'Upperbound', 'Lowerbound', 'Location', 'Northwest');
title('Tau^2 = 84^2');


%{
We see that the interval of uncertainty of our predictions increases
 as x
increases (i.e as the distance between our target years and our prior
 years
 increases). We see that the uncertainty around our target year 2100
 is
 much less than it was when we used tau^2 = 10 but much greater than
 tau^2
 = 10000. This makes sense based on our discussion of what role tau
 plays.
 Tau basically tunes our predictions such that our predction for any
 years
that satisfy  |x - x'| > tau go to 0 very quickly.

%}
```




```
%8.

%{
Similar to our reasoning for 7. , I say that it is best to use tau^2 =
 (2200 -
 2016)^2 since for values of tau^2 any less than that would cause our
 predictions for years around 2200 to go to 0 very quickly. To capture
 more
 localized changes in the data, I would use the tau^2 for 2100 (i.e
 84^2)
 because this tau gives more importance to smaller values for the
 years
 (those less than 2100). Any year passed 2100, would cause our
 prediction
 to be around 0 so it would give less emphasis to larger years. If we
 used
 the tau^2 for 2200, then we would not be able to capture more
 localized
 changes in our data.


%}
```

Published with MATLAB® R2017a