Data:

- Globally averaged monthly surface temperature anomalies for all months of January 1958 to December 2008.
- Time series of ENSO index for all months of January 1958 to December 2008.

Note: The complete code to produce these results is contained in the IDL file *kcw_hw3.pro*.

Problem 1: Investigate the effect of persistence (red noise) on the sample mean and standard deviation

In this problem, we are essentially simulating the effects of persistence by creating our own red-noise time series. Recall that the red noise model expresses a time series as “today = yesterday + noise”, i.e.,

\[ x(t) = r_1 \cdot x(t - \Delta t) + b \cdot \epsilon(t) \]  

(1)

where \( x(t) \) is “today”, \( x(t - \Delta t) \) is “yesterday”, \( r_1 \) is the lag-one autocorrelation of \( x \), the \( \epsilon \) is random normally distributed “noise”, and

\[ b = \sqrt{1 - r_1^2} \]

For discrete samples \( i \):

\[ x_i = r_1 \cdot x_{i-1} + b \cdot \epsilon \]  

(2)

Another way to think of this is in terms of the fraction of the variance explained by persistence and noise. In this red-noise model, the fraction of the variance of the time series \( x \) that is due to persistence is \( r_1^2 \), and the fraction of the variance due to noise is \( b^2 \). Hence, \( r_1^2 + b^2 = 1 \).

Figure 1 shows my four simulated time series: one with no persistence, and three with persistence. Each has \( N = 5000 \) elements. The lag-one autocorrelations were specified to be 0.0, 0.1, 0.6, and 0.9 respectively. Here is the actual IDL code for creating these simulated series (lines preceded by semi-colons are comments):
; Initialize
N = 5000L
T0 = FLTARR(N)
T1 = FLTARR(N)
T2 = FLTARR(N)
T3 = FLTARR(N)

; Each red-noise time series has the form:
; T[i] = a*T[i-1] + b*noise[i]
; where a^2 + b^2 = 1
; Define lag-one autocorrelations for
; each, which I’ll call a0, a1, a2, a3
a0 = 0.0
a1 = 0.1
a2 = 0.6
a3 = 0.9

; The noise coefficient (b) for each is thus:
b0 = 1.0
b1 = SQRT(1.0-a1^2.0)
b2 = SQRT(1.0-a2^2.0)
b3 = SQRT(1.0-a3^2.0)

; Create the white noise series. Use the same white noise
; for each of the red noise.
seed = 5L
white_noise = randomn(seed,N)

; Make T0 just white noise, and make the first element
; of each red-noise series the same.
T0 = white_noise
T1[0] = white_noise[0]
T2[0] = white_noise[0]
T3[0] = white_noise[0]

; Now generate the remaining N-1 elements of the time
; series using the lag-one autocorrelation plus noise.
FOR i=1L,N-1L DO BEGIN
   T1[i] = a1*T1[i-1] + b1*white_noise[i]
   T2[i] = a2*T2[i-1] + b2*white_noise[i]
   T3[i] = a3*T3[i-1] + b3*white_noise[i]
ENDFOR
Just looking at the raw time series in Figure 1, we can already kind of see how more persistence (i.e., a larger lag-one autocorrelation) will tend to bias estimates of the mean and standard deviation. For example, the “true” mean ($\mu$) and standard deviation ($\sigma$) of these time series should be very close to 0 and 1, respectively. However, imagine you were to pull a small sub-set of length 100 from any of these red-noise series, then compute the mean and standard deviation of that sub-set. You are getting the sample mean ($\overline{x}$) and standard deviation ($s$). Would you get something close to $\overline{x} = 0$ and $s = 1$, or would you get something different? As we’ll see, you’ll get something different, and the difference will be greater if the lag-one autocorrelation ($r_1$) is greater.

Before getting into the brute force way of visualizing this, just look at the time series and imagine pulling out a sub-set. The first series in Figure 1 is essentially just white noise ($r_1 = 0$), and the second series is only slightly “reddened”. So, if you were to pull out a 100-length sample from these two, the sample mean and sample standard deviation you would get should be close to 0 and 1 because any chunk looks about like any other chunk. Specifically, there is as much chance that any given element will be above the zero line as below it, hence the sample mean would be approximately the same as the true mean (i.e., zero).

However, the last two series have more persistence, which biases the mean and the standard deviation. That is, for a red-noise series, there is a greater tendency for elements to be above or below the zero line if their neighbors are above or below the zero line. The effect of this will be to make the sample mean different from zero and make the sample standard deviation different from 1 (more often smaller than 1, and less often greater than 1). Hence the sample standard deviation would tend to be an underestimate of the “true” standard deviation.

Now let’s do the brute force simulated sampling to clarify these effects. Figure 2 shows the distributions of the sample mean and sample standard deviations. These result from simply pulling out 10,000 random 100-length samples from each of the time series, then computing the 10,000 means and standard deviations of those 100-length samples. For example, in IDL code:

```
; Initialize
nSamples = 10000L
sample_mean = FLTARR(nSamples)
sample_std = FLTARR(nSamples)

; Loop to get the sample statistics.
; The ‘randomu’ function gets a random number between [0,1] from a
; uniform distribution. So, that makes it equally likely that I will
; get any number between zero and one. I then multiply that number by
; 4900 to get the random start point between 0 and 4900,
; then pull out a 100-length sample that
; starts at that start point.

; For each time series, I reset the random seed so that each set of
; random start points is the same.
sample_seed = 10L
FOR i=0L,nSamples-1L DO BEGIN
    start_point = ROUND(4900.0*randomu(sample_seed))
    sample = T0[start_point:start_point+99]
    sample_mean[i] = mean(sample)
    sample_std[i] = StdDev(sample)
ENDFOR
```
I’ve plotted the distribution histograms in Figure 2 as relative frequency distributions, i.e., simply dividing the number in each histogram bin by the total number of samples (i.e., 10,000), then multiplying by 100 to put the results into percentage of samples in each bin. This is a common way to represent a sampling distribution, and it makes it easier to compare the results of the four different time series.

From the histograms in Figure 2 we see that a larger $r_1$ leads to a marked broadening of the distributions of sample mean ($\bar{x}$) and standard deviation ($s$), as expected. So, the basic effects of persistence are:

- **Effect on sample mean**: $\bar{x}$ is more likely to be different from the true mean $\mu$, i.e., not representative of what we would get if we took the mean of the full 5000-length time series.

- **Effect on sample standard deviation**: the larger is $r_1$, the more different will be any given $s$ from the true standard deviation $\sigma$. Also, $s$ tends to be an underestimate of $\sigma$, i.e., it is more often less than $\sigma$ than it is greater than $\sigma$. The sample distribution of $s$ is thus skewed.

Implications for significance tests:

- When we are assessing the significance of a sample mean:

  \[
  t = \frac{\bar{x} - \mu}{s} \sqrt{N - 1}
  \]

  we are underestimating the standard deviation in the denominator (we are using $s$ instead of $\sigma$ because $s$ is the only thing we know), so we end up overestimating the significance. Hence, we would want to account for this by replacing $N$ with a better estimate of the true number of independent samples.

- When assessing the significance of a correlation between two time series:

  \[
  t = \frac{r \sqrt{N - 2}}{\sqrt{1 - r^2}}
  \]

  or as Santer et al. express it:

  \[
  t = \frac{b}{s_b}
  \]

  where $b$ is the regression coefficient between two time series, and $s_b$ is proportional to the standard deviation relative to the regression coefficient. Again, we are underestimating the standard deviation, overestimating the number of independent samples, and thus overestimating the significance. So again, we would need to account for that by replacing $N$ with a better estimate of the true number of independent samples.

Bottom line: persistence leads to overestimation of significance if we don’t account for it somehow.
Figure 1: Artificially generated red noise time series. From top to bottom, the time series have lag-one autocorrelations ($r_1$) of 0.1, 0.6, and 0.9. The red line is just a reference zero line.
Figure 2: Sample distributions of (left) mean and (right) standard deviation resulting from drawing 10,000 random 100-length samples from the red noise time series of Figure 1. The histograms were converted to relative frequency for easier comparison. Note how the larger $r_1$ results in a greater spread about the true values of $\bar{x} = 0$ and $\sigma = 1$. 
Problem 2: Investigate the relationship between *global* mean temperature and ENSO at multiple lags.

The objectives of this problem are:

- to apply what we know about the effects of persistence to a real data set.
- to apply a lag-correlation method to investigate delayed responses between two time series, and
- to see if ENSO might have anything to do with recent global warming (spoiler alert: it apparently doesn’t, at least according to this analysis)

**Calculate Linear Trend of Global Mean Temperature**

This sets the baseline for the analysis that follows. Figure 3 shows the time series of global mean temperature anomalies, along with the best fit line representing the linear trend. This linear trend is simply what you get by regressing temperature against time, in IDL-speak:

**CODE**

```
; Create a linearly increasing time index.
; nLines is the number of elements of the time series (612).
; So the 'months' array here is just [0,1,2,...,611]

months = FINDGEN(nLines)
trend_fit = poly_fit(months,global_temp,1)

; trend_fit[1] is the regression coefficient, i.e., slope
; Put into units of Kelvin/decade:

kPerDecade = trend_fit[1]*12.0*10.0
```

For the full time series (Jan. 1958 to Dec. 2008), I get regression coefficient of about **0.001088**, a correlation coefficient of about **0.8033** and a linear trend of about **0.1306 Kelvin per decade**.

**Lag-one Autocorrelations and Number of Independent Samples**

To truly assess the significance of this trend (and of correlations with ENSO that follow), we’ll need to account for persistence so that we can use a more representative number of independent samples. Use the lag-one autocorrelations to do that. The lag-one autocorrelations for global temperature and ENSO are found simply like this, for example:

```
CODE

; Regress time series against "one-shifted" version of itself.

left = global_temp[0:nLines-2]
right = global_temp[1:nLines-1]

; Standardize so that correlation coefficient is the same as
; regression coefficient.

left = (left - Mean(left))/StdDev(left)
right = (right-Mean(right))/StdDev(right)

r1_temp = (poly_fit(left,right,1))[1]

I get the following results: Global Temp, \( r_1 \approx 0.883 \), and ENSO \( r_1 \approx 0.927 \).

So, plugging this into the equation to get effective number of independent samples:

\[
n_{eff} = n - \frac{r_1^2}{1 + r_1^2}
\]

I get: Global Temp, \( n_{eff} \approx 76 \), and ENSO \( n_{eff} \approx 46 \)

That’s a pretty drastic reduction from the \( n = 612 \) months of the time series. The \( n_{eff} \) for ENSO is 46, which is about the same as the number of years in the data set. The \( n_{eff} \) for global temperature is about 76, which is about 1.5 times the number of years in the data set. Hence, to some degree, these \( n_{eff} \) numbers agree with our common sense that a quasi-yearly sampling rate would lead to independent samples. Or another practical way to look at this: we could probably get away with a low sampling rate of only once per year if we were interested in relationships between ENSO and global temperature.

Will our linear trend of global temperature be significant with only \( n_{eff} = 76 \) independent samples? Let’s see. Our t-stat value for our correlation between temperature and time (\( r = 0.8033 \)) is:

\[
t = \frac{r \sqrt{n_{eff} - 2}}{\sqrt{1 - r^2}}
\]

or as Santer et al. express it:

\[
t = \frac{b}{s_b}
\]

(I calculate it both ways in the full code of kcw_hw3.pro). Either way, I get a t-stat value of about 11.6. This is huge. For example, the null hypothesis value from a Student’s T distribution for a two-tailed confidence test with 74 degrees of freedom at the 95% level is about 2. For the 99.999% level, the Student’s T value is about 4.74. So, basically, this linear trend is significant. Big surprise: the globe is warming, and that warming is significant.

Best Fit Between Temperature and ENSO
Here I’m simply leading or lagging the ENSO time series relative to the global temperature time
series, then computing the correlation at each lead/lag. Let the lead/lag vary from -12 months to +12 months. The left panel of Figure 5 shows the resulting plot of correlation versus lag. Negative lags indicate ENSO precedes temperature. I did this two different ways for comparison, hence the two lines in the figure.

Method 1 (black line in Figure 5): Use as many elements of the time series as possible. For a lag of ±1, we have only \( N - 1 \) values each time series. For a lag of ±2, we have only \( N - 2 \) values in each time series, and so on.

Method 2 (blue line in Figure 5): Use the same number of elements in the time series for each lag correlation calculation. Here I just removed the first and last year from the temperature time series and kept that the same for each calculation, i.e.,

\[
\text{T}_{\text{subset}} = \text{global_temp}[12:\text{nLines}-13]
\]

So, the temperature subset stays the same. I then varied the ENSO subset for each lag iteration, i.e., from a lag of -12 (ENSO leads by 12 months) in the first iteration,

\[
\text{enso}_{\text{subset}} = \text{enso}[0:\text{nLines}-25]
\]

to a lag of -11, -10, and so on until we get to a lag of +12 (ENSO lags by 12 months) in the final iteration,

\[
\text{enso}_{\text{subset}} = \text{enso}[24:\text{nLines}-1]
\]

See kcw_hw3.pro for the complete code.

In both methods, I found the largest positive correlation at a lag of –6, i.e., ENSO leads temperature by 6 months. However, the correlation coefficient has a pretty steady “hump” between lags of –3 and –6. Surprisingly, the largest value was when temperature led ENSO by about a year, i.e., at a lag of +11. Very strange. Anyway, the best positive correlation is about 0.15, so ENSO explains about \( r^2 \approx 2\% \) of the variance in the global temperature.

The upper-right panel in Figure 5 shows the time series of temperature and ENSO, with the ENSO time shifted to the right by 6 months, i.e., at the best fit. The lower right panel just shows the scatterplot of temperature and lagged ENSO.

When I assess the significance of the correlation between ENSO and global temperature (using \( N = n_{\text{eff}} \approx 46 \)),

\[
t = \frac{r \sqrt{N - 2}}{\sqrt{1 - r^2}}
\]

I get t-stat values of about 0.76 and 1.05 for Method 1 and 2, respectively. The two-tailed Student’s T statistic for 95% confidence with (\( \nu = 44 \)) degrees of freedom is about 2.0. The 90% confidence value is about 1.68. The 80% confidence value is about 1.3. I have to conclude that the correlation is not significant.

I’m certainly no climatologist, but it is my understanding that a positive ENSO phase (i.e., warmer than average temperatures across the central and eastern Pacific) generally leads to warmer global temperatures. I expected to find a more significant relationship here, so these relationships are interesting to me. For example, the time series in the upper-right panel certainly suggests that
there are some periods when temperature and ENSO covary pretty tightly (especially once you let ENSO lead by 3-6 months). Yet there are other periods when they don’t. The scatterplot and significance tests do not instill me with much confidence in the correlation though. Perhaps if we restricted our lead/lag calculations to only the winter months, we would find a better correlation.

(Historical note: I did this calculation several years ago using data only up through the year 2000. I found a much larger correlation of about 0.39 at a lag of –3. In that case, ENSO explained about \( r^2 \approx 15\% \) of the variance in global temperature. I can’t imagine that 8 more years of data could make such a difference. However, I was using a different metric of the ENSO index. Not all ENSO indices are created equal, I suppose. It’s a bit disconcerting to me.)

**Impact of ENSO on Recent Global Warming**

To investigate whether ENSO might have had some impact on the recent global warming trend, I first extract the part of the global temperature time series that is best correlated with ENSO. Specifically, I first restrict the temperature time series to that which includes months 6 to N, and restrict the ENSO time series to that which includes months 0 to N–6. I then compute the ENSO-fitted temperature time series:

\[
\hat{T} = a_0 + a_1 \cdot \text{ENSO}
\]

where \( a_1 \approx 0.037^\circ C \) per unit standard deviation of ENSO is the regression coefficient, i.e., the slope of the line in the bottom-right panel of Figure 5. I’ll call \( \hat{T} \) the ”trend fit”.

I can then get the part of the temperature time series that should be uncorrelated with ENSO:

\[
T^* = T - \hat{T}
\]

which I’ll call the “residual”.

Figure 4 shows all three series (raw, \( \hat{T} \) and \( T^* \)) on the same plot. The linear trends of these three time series are in the table below:

<table>
<thead>
<tr>
<th></th>
<th>Raw</th>
<th>Fit (( \hat{T} ))</th>
<th>Residual (( T^* ))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trend (K/decade)</td>
<td>0.1329</td>
<td>-0.0011</td>
<td>0.1340</td>
</tr>
<tr>
<td>Variance</td>
<td>0.0575</td>
<td>0.0007</td>
<td>0.0567</td>
</tr>
<tr>
<td>Percent of Variance</td>
<td>100.0</td>
<td>1.29</td>
<td>98.71</td>
</tr>
</tbody>
</table>

The trend for the “raw” time series is a little different from what I got earlier because I’m using a slightly different sub-set of temperatures (i.e., not using the first 6 months). Also shown in the table are the variance of the three time series, along with the percentage of the total variance explained by the ENSO fit and residual.

Basic results here:

- Apparently, ENSO does not “explain” much of the variance in global mean temperature (at least not according to these results).

- ENSO certainly does not seem to contribute to the overall warming trend. No surprise here. The “O” in ENSO is “oscillation” after all. An oscillation generally doesn’t have a trend.
Figure 3: Time series of monthly global temperature anomaly from January 1958 to December 2008. The straight line indicates the linear trend.

Figure 4: Time series of raw, ENSO-fitted, and residual time series.
Figure 5: Results of lag-correlation analysis. (a) Plot of the correlation coefficient between global mean temperature and ENSO index as a function of lag. Negative lag indicates that ENSO leads global temperature. The black and blue curves correspond to methods 1 and 2 discussed in the text. (b) Time series of global temperature anomaly (black) and ENSO index (red). (c) Scatterplot of global temperature anomaly versus ENSO index. The ENSO index in both (b) and (c) has been shifted by –6 (i.e., lag corresponding to maximum correlation) prior to calculation and plotting.