Today's plan….

1. Autocovariance and Autocorrelation
2. First order autoregressive model (AR1) and Red Noise
3. White Noise
4. Independent samples and effective sample size
5. Multiple regression
6. Granger causality

Can we assume our data are stationary and have no memory (i.e., are independent)? Not always...

Thus far, we have assumed that our time series (and data sets) are all stationary and have no intrinsic memory. Now, we will discuss these assumptions, and how to determine the true number of degrees of freedom in an autocorrelated data set.

Stationarity implies that the statistics of a time series (its mean and higher-order moments) are independent of time, i.e. unchanging in time. In general, we will assume that this is the case. Note that this means one should *remove any trend* in the data before performing the analysis. The trend can be removed in the method previously discussed using linear regression.

Calculating the autocovariance and autocorrelation

The autocovariance function ($\gamma(t)$) is the covariance of a time series with itself at another time, as measured by a time lag (or lead) τ. For a time series $x(t)$, it is defined as

$$\gamma(\tau) = \frac{1}{(t_N - \tau) - t_1} \sum_{t=t_1}^{t_N-\tau} [x'(t) \cdot x'(t+\tau)] \tag{67}$$

where t_1 and t_N are the starting and end points of the time series, respectively, and the prime denotes departures from the long-term mean.

Note that for a continuous time series with time positions $k = 1, 2, 3...N$:

$$\gamma(\tau) = \overline{x'(t)x'(t+\tau)} \tag{68}$$

and for $\tau = 0$, the autocovariance is $\gamma(0) = \overline{x'^2}$ = variance.

The more commonly used *autocorrelation* $\rho(\tau)$ is just $\gamma(\tau)$ normalized by $\gamma(0)$. It is simply the correlation of a time series with itself at another time.

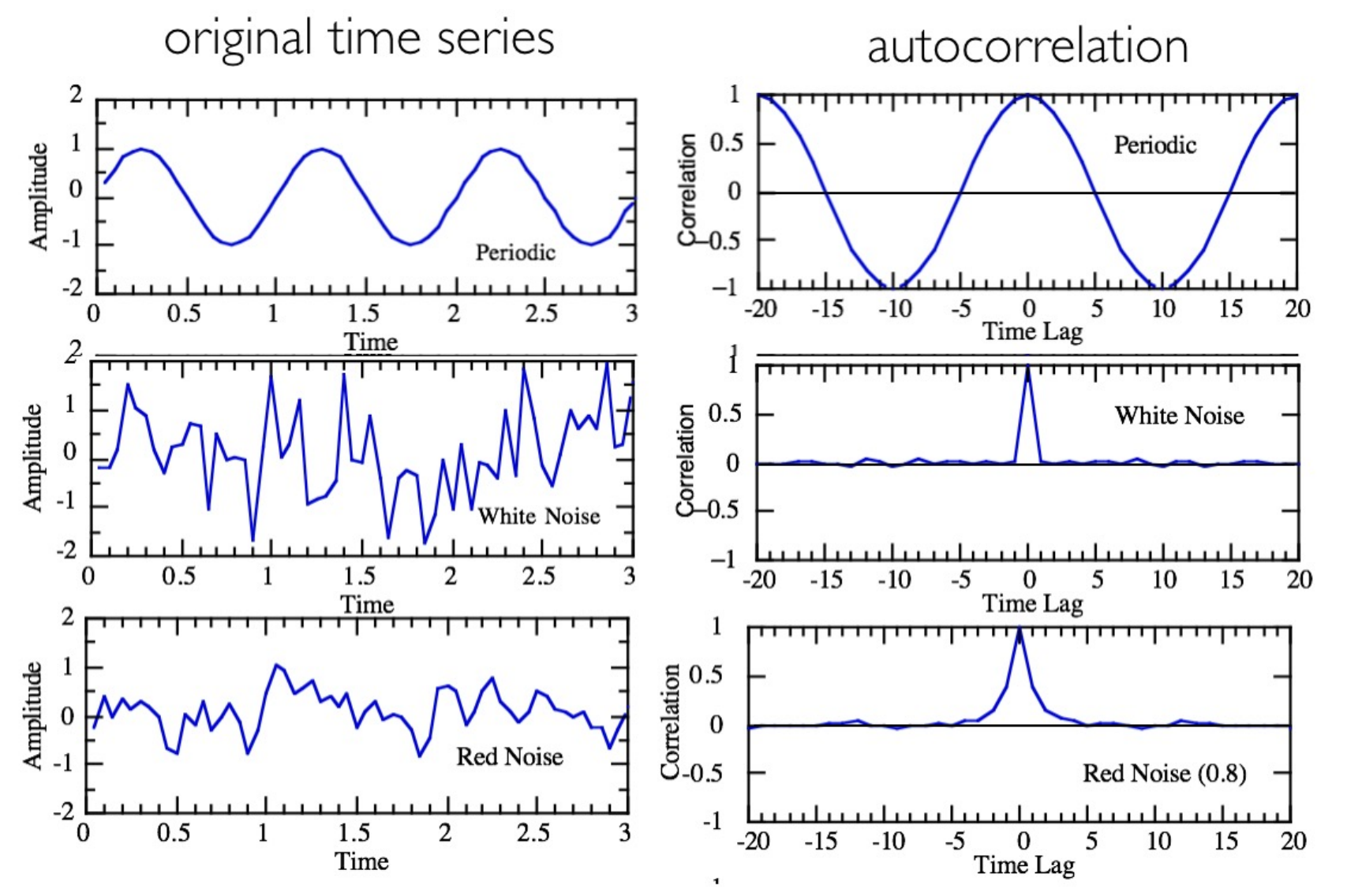
Write these equations down:
Autocovariance (gamma), autocorrelation (rho), and variance

WRITING ON THE BOARD

Notes on the autocorrelation (rho):

- γ is symmetric about $\tau = 0$
- $-1 \leqslant \rho(\tau) \leqslant 1$
- $\rho(0) = 1$
- if the time series is not periodic, $\rho(\tau) \to 0$ as $\tau \to \infty$

Examples of autocorrelation functions – i.e., the auto correlation at various lags:



First order autoregressive model or a "red noise process"

red noise process: "today is like yesterday + noise"

A red noise time series is defined mathematically as:

$$x(t) = a \cdot x(t - \Delta t) + b \cdot \epsilon(t) \tag{69}$$

where

- x is a standardized variable
- Δt is the time interval between data points (and is assumed to be a constant here)
- a lies between 0 and 1 and measures the memory of the previous state
- $(t - \Delta t)$ is the day before day t
- $\epsilon(t)$ is a random variable drawn from the standard normal distribution and represents noise in the system

Measure the Memory in a red noise time series
Use a, the lag-1 autocorrelation (AR1)

To determine a: multiply the l.h.s. and r.h.s. by $x(t-\Delta t)$ and take the time average

$$\overline{x(t)x(t-\Delta t)} = a \cdot \overline{x(t-\Delta t) \cdot x(t-\Delta t)} + b \cdot \overline{\epsilon(t)x(t-\Delta t)} \tag{70}$$

- since x is standardized (variance of 1), the first term of the r.h.s. is $a \cdot 1$
- since $\epsilon(t)$ is random in time, assuming your time series is long enough, the last term on the r.h.s is 0.

Thus, for a standardized x

$$a = \overline{x(t)x(t-\Delta t)} = \gamma(\tau = 1) = \gamma(1) \tag{71}$$

That is, a is the autocovariance at lag Δt, or, one time step ahead. Since x is standardized, this is also the autocorrelation at lag Δt, so,

$$a = \rho(\Delta t) = \rho(1) \tag{72}$$

Measure the Noise in a red noise time series
Use b, the noise magnitude

What about b - the magnitude of the noise? Since $x(t)$ and $\epsilon(t)$ both have unit variance, one can square both sides of the red-noise equation and then take the average to solve for b:

$$\overline{x^2(t)} = \overline{a^2 x^2(t-\Delta t)} + \overline{b^2 \cdot \epsilon^2(t)} \tag{73}$$

$$1 = a^2 \cdot 1 + b^2 \cdot 1 \tag{74}$$

$$b = \sqrt{1-a^2} \tag{75}$$

What is a white noise time series? a time series with no memory

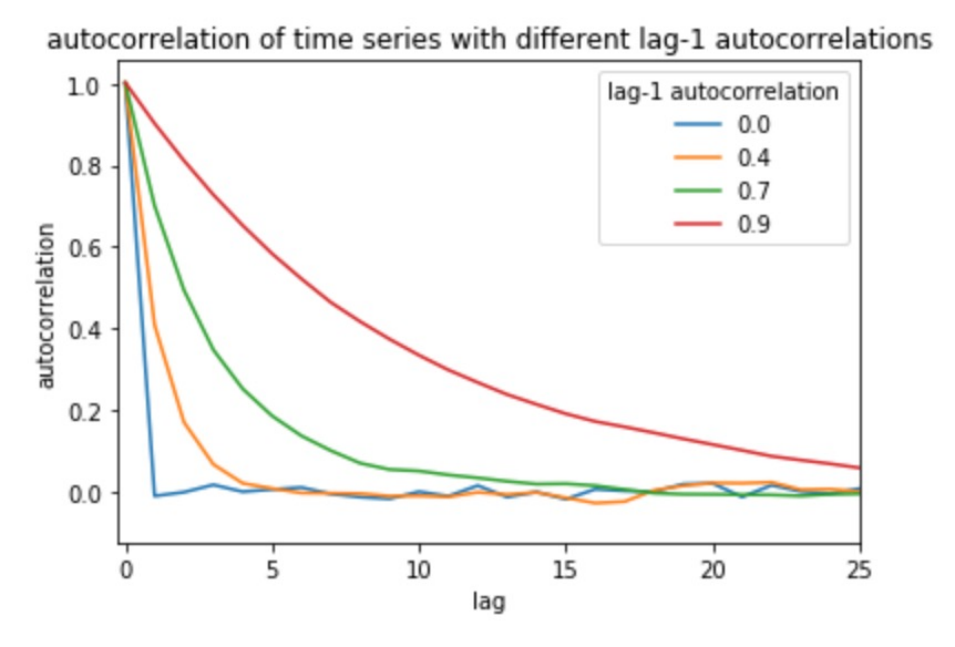
Whereas red noise is defined as

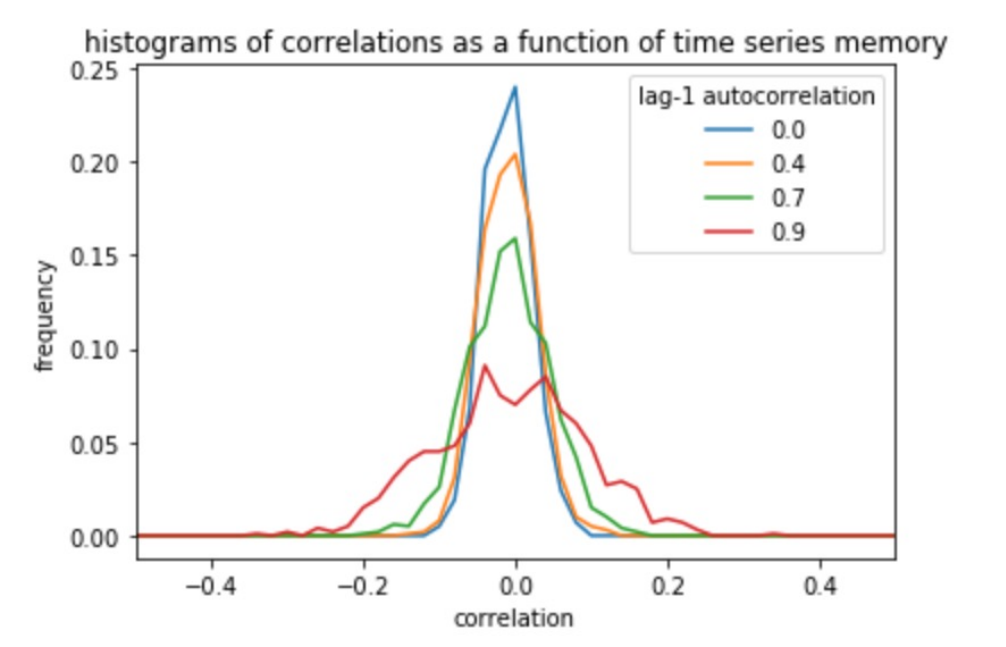
$$x(t) = a \cdot x(t-\Delta t) + b \cdot \epsilon(t) \tag{86}$$

White noise is the special case where $\rho(\tau > 0) = 0$, so $a = 0$.

White noise has equal power at all frequencies and has zero autocorrelation (no memory of the previous time steps). In geophysical applications, white noise is generally assumed normally distributed.

Let's examine the influence of the lag-1 autocorrelation





Try the python code – correlation_with_memory.ipynb

You can predict X at a future time, if the time series is red

One can then use the equation for a red noise process to predict the value of x at a later time. For example, two time steps into the future:

$$x(t) = a \cdot x(t - \Delta t) + b \cdot \epsilon(t) \quad (76)$$

$$x(t + \Delta t) = a \cdot x(t) + b \cdot \epsilon(t) \quad (77)$$

$$x(t + 2\Delta t) = a \cdot x(t + \Delta t) + b \cdot \epsilon(t) \quad (78)$$

Multiply both sides by $x(t)$ and time average:

$$\overline{x(t) \cdot x(t + 2\Delta t)} = \overline{x(t) a \cdot x(t + \Delta t)} + \overline{x(t) b \epsilon(t)} \quad (79)$$

$$\overline{x(t) \cdot x(t + 2\Delta t)} = \overline{a \cdot x(t) \cdot x(t + \Delta t)} + 0 \quad (80)$$

$$(81)$$

Therefore,

$$\rho(2\Delta t) = a\rho(\Delta t) = \rho^2(\Delta t) \quad (82)$$

since $a = \rho(\Delta t)$. More generally,

$$\rho(n\Delta t) = \rho^n(\Delta t). \quad (83)$$

Autocorrelation for a red-noise time series is an exponential

$$\rho(n\Delta t) = \rho^n(\Delta t). \tag{83}$$

The function that has this property is the exponential: $e^{(nx)} = (e^x)^n$. So, it turns out that the autocorrelation for a red-noise time series is an exponential:

$$\rho(n\Delta t) = e^{(-n\Delta t)/T_e} \tag{84}$$

where T_e is the e-folding time of the autocorrelation function (more on this in a second). In other words, the autocorrelation function of red noise decays exponentially for increasing lag $\tau = n\Delta t$.

The e-folding time-scale is the time it takes for the autocorrelation to drop to $1/e = 0.368$ of the original value (1), and can be computed as

$$T_e = -\frac{\Delta t}{\ln(a)} \tag{85}$$

So, if $\Delta t = 1$, and $\rho(\Delta t) = \rho(1) = a = 0.6$, then the e-folding time of the autocorrelation function is $T_e = 2$. In other words, the time series loses approximately $0.63^2 = 40\%$ of its memory after 2 days.

Independent samples and Effective sample size (N*)
How many samples do you really have???

Recall that the sample size N greatly impacts the variance of the sample means.

Persistence in a data set leads to an *overestimation* of the sample size, because each data point is not independent of those around it.

Consider the t-statistic:

$$t = \frac{\overline{x} - \mu}{\frac{s}{\sqrt{N-1}}} \tag{87}$$

If we assume no persistence in a red-noise time series, the sample mean standard deviation $\hat{s} = \frac{s}{\sqrt{N-1}}$ will be an underestimate and thus the t statistic will be over-estimated.

The most convenient way to deal with persistence in your time series is to introduce an *effective sample size*, N^*.

$N^* \leqslant N$ and can be substituted into the original formulas in place of N.

Formula to estimate the effective sample size (N*)

The estimation of N^* is generally approached assuming that the data follows a first-order autoregressive process (red noise). In this case, N^* can be estimated using the approximation:

$$\frac{N^*}{N} \cong \frac{1 - \rho(\Delta t)}{1 + \rho(\Delta t)} \text{ (see Wilks page 127)} \tag{88}$$

where N is the number of data points in the time series, Δt is the time step and $\rho(\Delta t)$ is the autocorrelation at lag 1.

- for white noise, if $\rho(1) = 0$, $N^* = N$
- as $\rho(1)$ increases, N^* decreases

Another (nearly identical) formula to estimate the effective sample size (N*) from Leith 1973

The above approximation is nearly identical to the discrete version of the effective sample size proposed by Leith (Journal of Applied Meteorology, p. 1066, 1973), given by

$$N^* \cong \frac{N\Delta t}{2T_e} = \frac{\text{total length of record}}{\text{two times the e-folding time of the autocorrelation}} \tag{89}$$

where T_e is the e-folding time of the autocorrelation function of the time series. The factor of 2 is included because any given point in a red noise time series can be predicted by points both before and after that point.

As T_e (the "redness" of the time series) increases, we get fewer degrees of freedom from each observation.

Note that the above can be re-written as:

$$\frac{N^*}{N} \cong \frac{\Delta t}{2T_e} = \frac{\Delta t}{-2\frac{\Delta t}{\ln a}} = \frac{\ln a}{-2} \tag{90}$$

Using the Leith formula, one can compute N^* as a function of the lag-1 autocorrelation of a time series:

$\rho(\Delta t)$	< 0.16	0.3	0.5	0.7	0.9
N^*/N	1	0.6	0.35	0.18	0.053

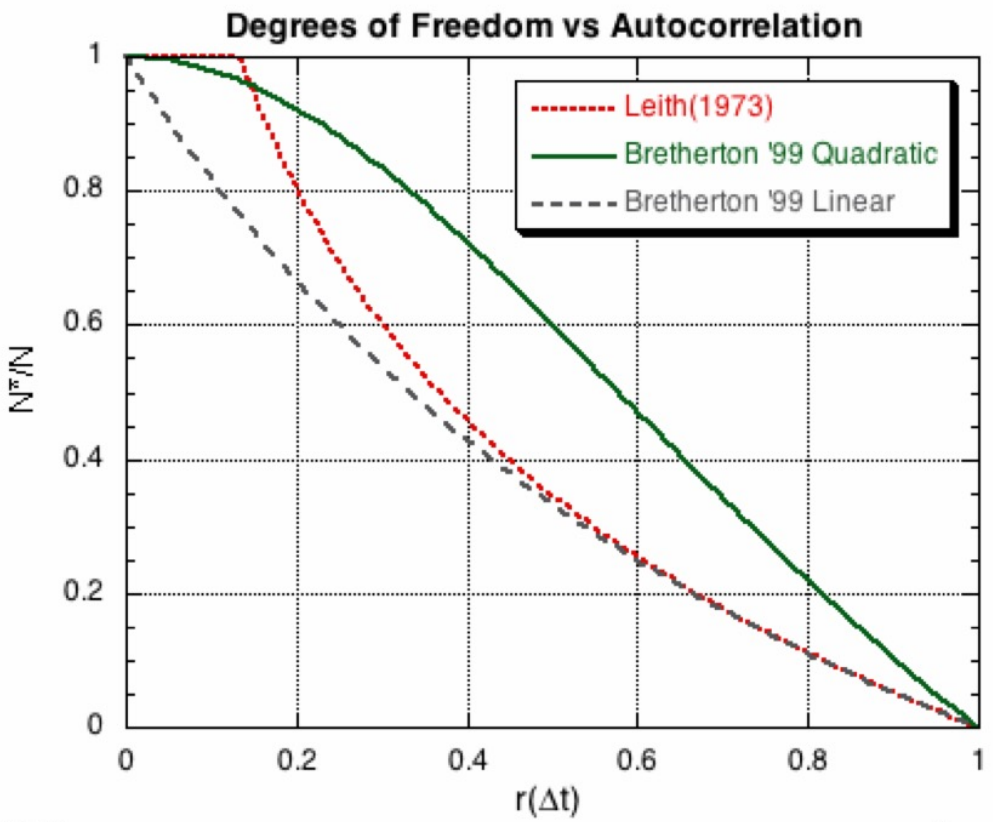
Bretherton et al. (J. Climate 1999) – a less conservative approximation:

Finally, Bretherton et al. (Journal of Climate, pg. 1990, 1999) take a less conservative approximation and have suggested using:

$$\frac{N^*}{N} \cong \frac{1-\rho^2(\Delta t)}{1+\rho^2(\Delta t)} \tag{91}$$

This formula yields almost 2 times more degrees of freedom than the Leith approximation. This approximation may be used when one is analyzing variance or higher-order moments. Otherwise, if one is interested in the mean, $\frac{N^*}{N} \simeq \frac{1-\rho(\Delta t)}{1+\rho(\Delta t)}$ should be used.

Effective sample size N*



Bretherton, 1999

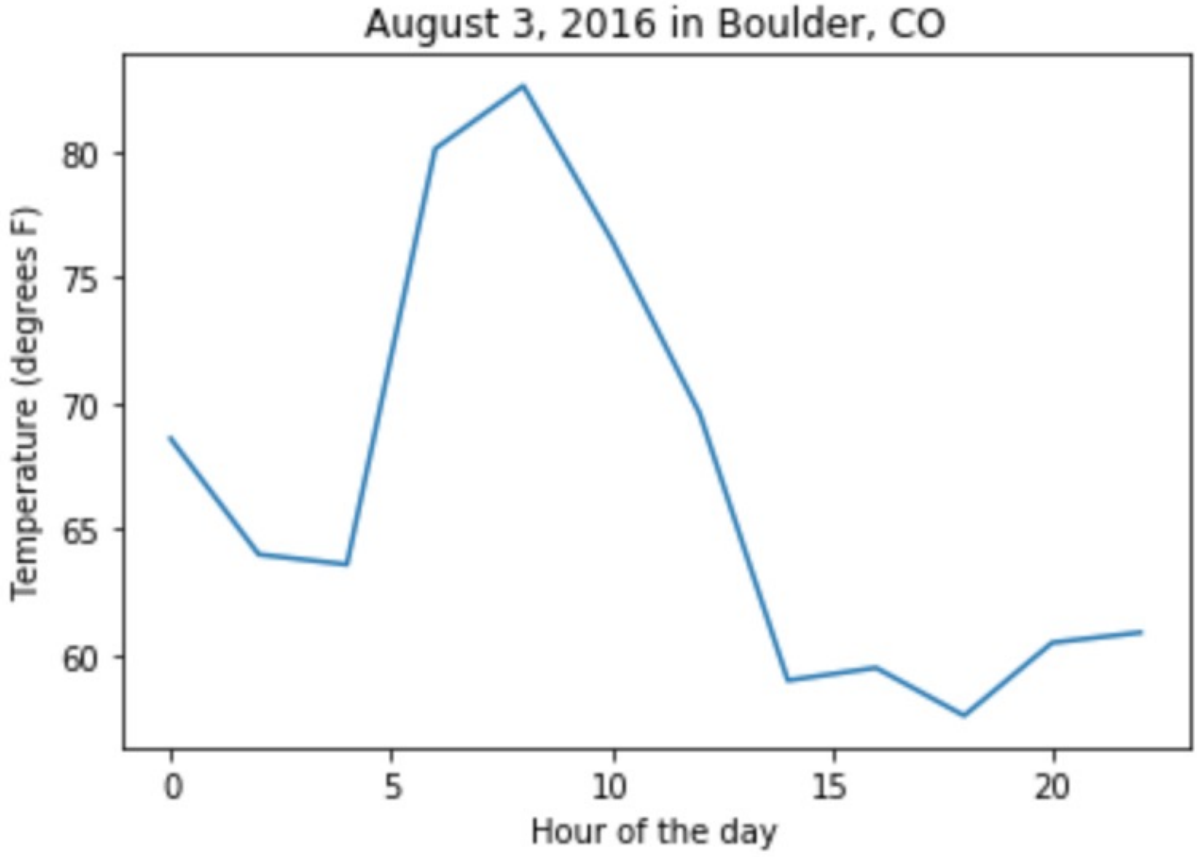
$$\frac{N^*}{N} \cong \frac{1-\rho^2(\Delta t)}{1+\rho^2(\Delta t)}$$

Leith, 1973

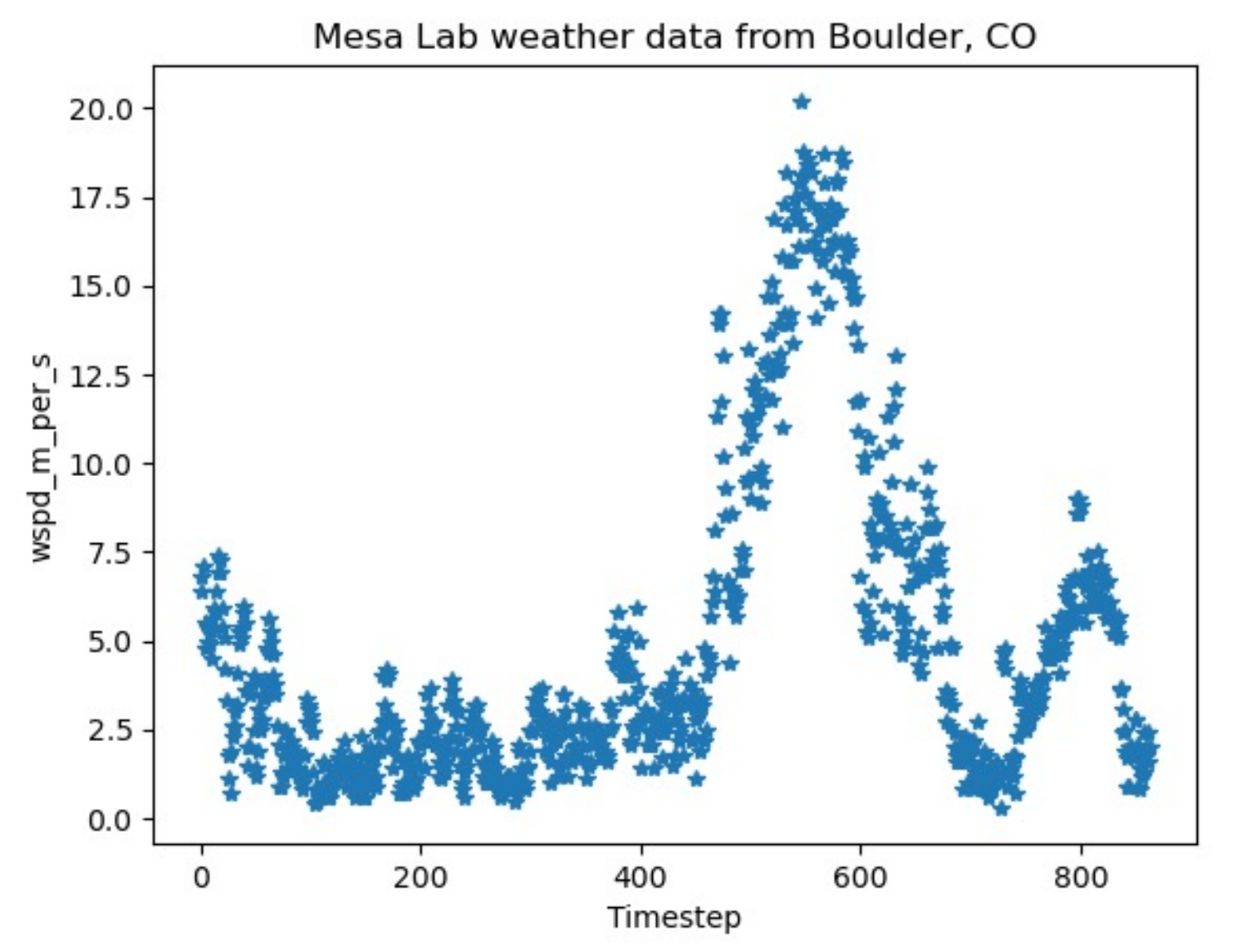
$$\frac{N^*}{N} \cong \frac{\Delta t}{2T_e} = \frac{\Delta t}{-2\frac{\Delta t}{\ln a}} = \frac{\ln a}{-2}$$

Example: Calculate the number of independent samples.

```
Sample Size N= 12
Mean Temperature 66.9
Standard Deviation in Temperature 8.3
```



Example: Calculate the number of independent samples.

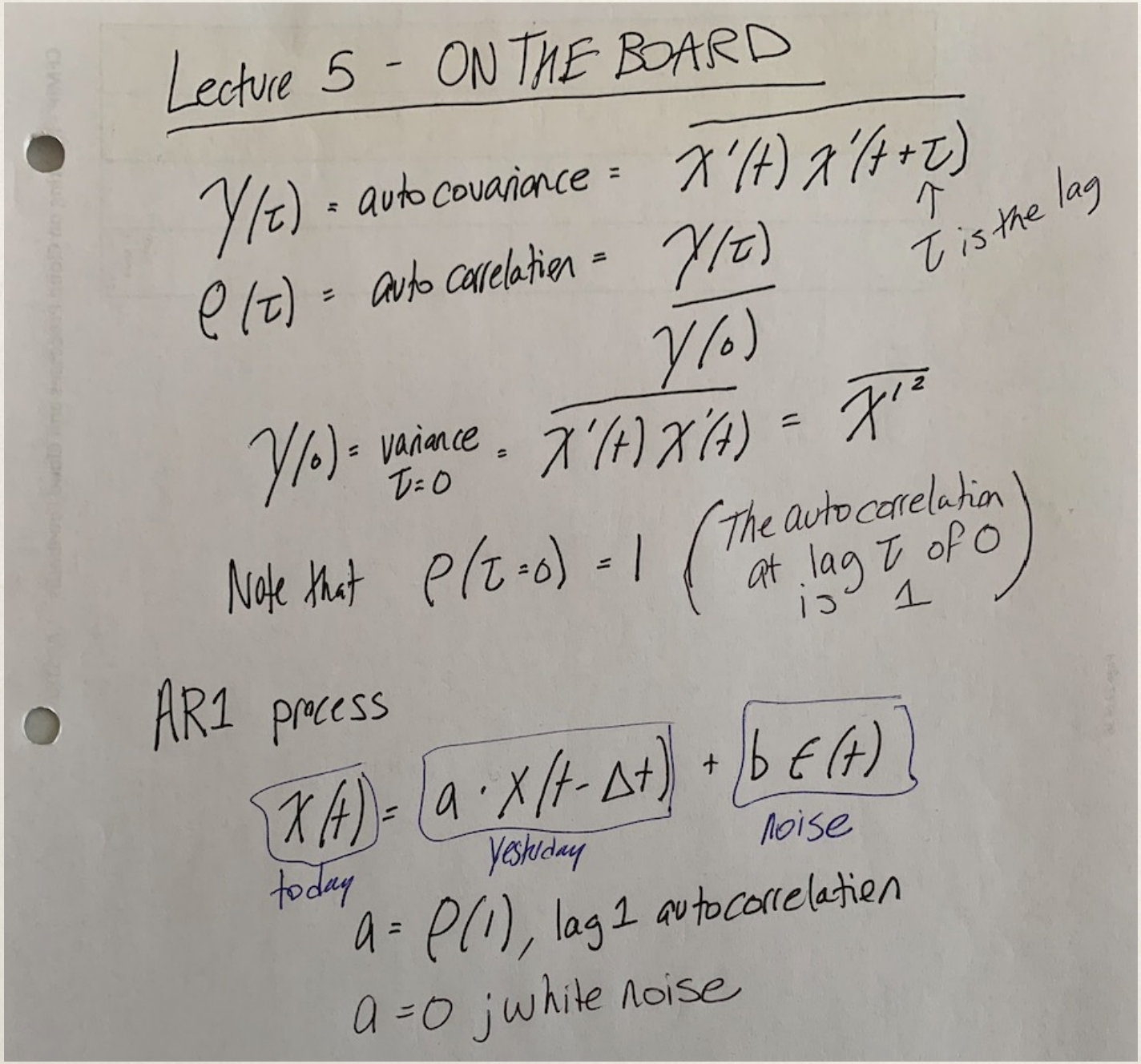


Try the python code – estimate_effective_sample_size.ipynb

Lecture 5 "on the board"

I recommend writing this down.

For me, Writing things down helps them sink in.



GROUP WORK

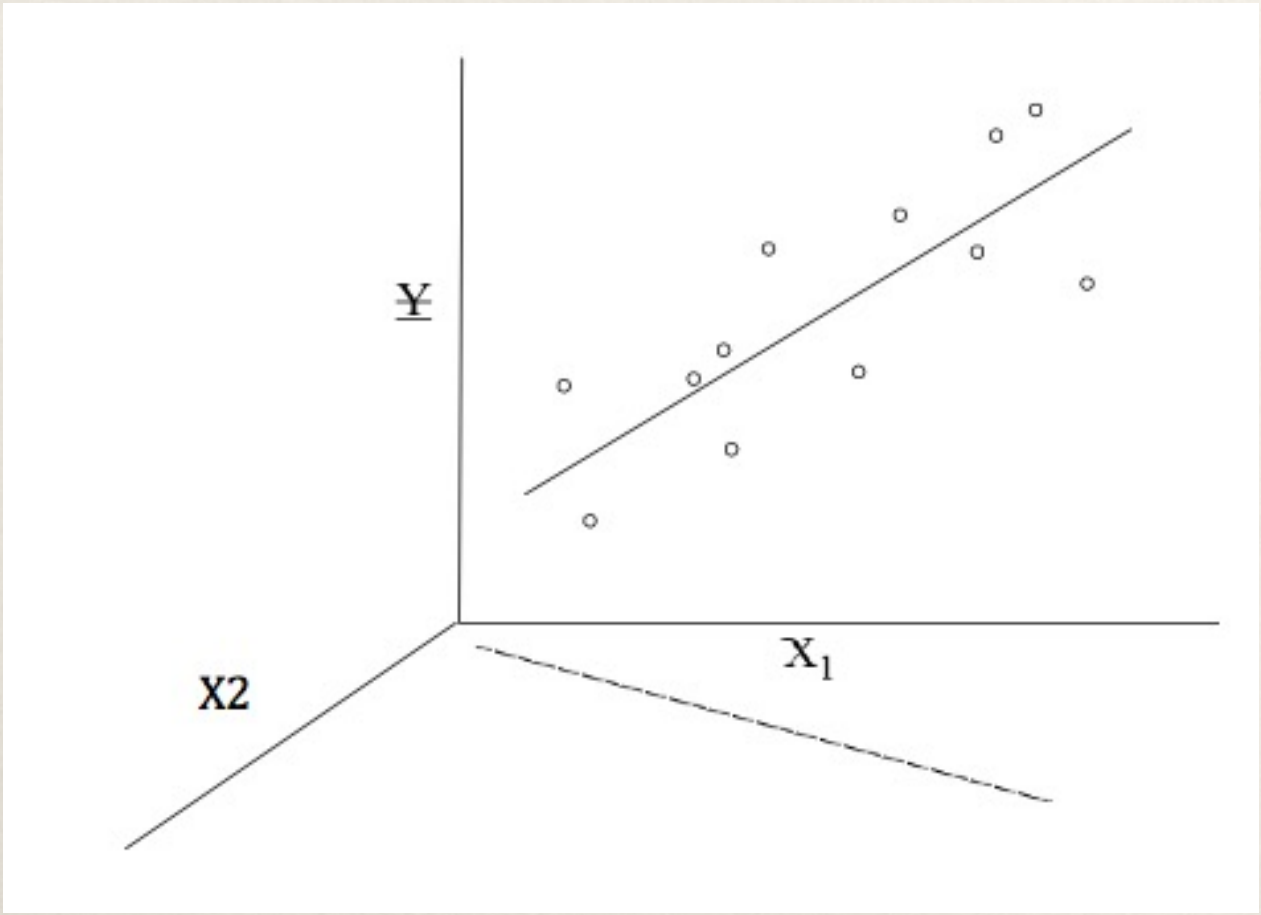
- *correlation_with_memory.ipynb*
- *estimate_effective_sample_size.ipynb*
- *Explain "Lecture #5 on the board" notes to each other.*

Multiple-regression (multi-linear regression)

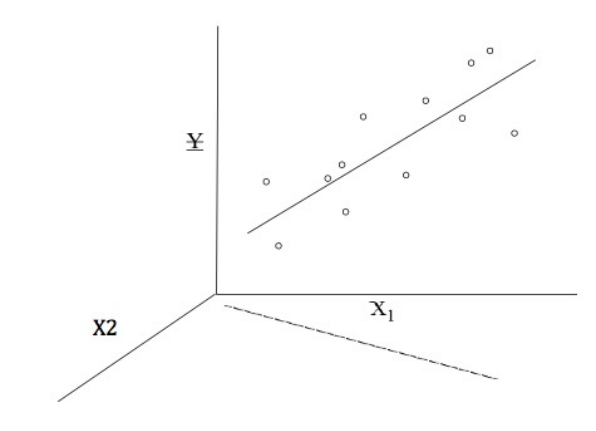
Basic idea: Generalize the derivation of the regression coefficient to multiple linear predictors.

$$\hat{y} = a_0 + a_1 x_1 + a_2 x_2 + ... + a_n x_n \tag{92}$$

Note that now the fit is in multiple phase space.



Multiple-regression – two predictors case (X_1, X_2) to predict Y



What does it mean if X_1 and X_2 are at right angles?

- they are orthogonal predictors (the inner product is 0)
- they give you independent information
- if X_1 and X_2 cover all possible combinations for the space, they are said to "form a basis"

If X_1 and X_2 are not orthogonal

- they are not independent
- they repeat information, (are redundant)

Multiple-regression – Minimize the cost function Q

For the multiple predictor case (predictors $x_1, x_2, x_3..., x_n$), we want to minimize

$$Q = \sum_i^N (\hat{y}_i - y_i)^2 = \sum_i^N (a_0 + a_1 x_{1,i} + a_2 x_{2,i} + a_3 x_{3,i} + ... + a_n x_{n,i} - y_i)^2 \qquad (94)$$

where n is the number of predictors and N is the number of time steps. Thus, $x_{2,i}$ denotes the predictor x_2 at time step i.

For n predictors, we have $n+1$ equations derived by setting

$$\frac{\partial Q}{\partial a_i} = 0 \qquad (95)$$

where i goes from 0 to n.

$$\begin{aligned}
\overline{y} &= a_0 + a_1\overline{x_1} + a_2\overline{x_2} + ... + a_n\overline{x_n} && (96)\\
\overline{x_1 y} &= a_0\overline{x_1} + a_1\overline{x_1^2} + a_2\overline{x_1 x_2} + ... + a_n\overline{x_1 x_n} && (97)\\
\overline{x_2 y} &= a_0\overline{x_2} + a_1\overline{x_2 x_1} + a_2\overline{x_2^2} + ... + a_n\overline{x_2 x_n} && (98)\\
&... && (99)\\
\overline{x_n y} &= a_0\overline{x_n} + a_1\overline{x_n x_1} + a_2\overline{x_n x_2} + ... + a_n\overline{x_n^2} && (100)
\end{aligned}$$

If we assume the mean has been removed from every variable, these simplify to n equations and n unknowns (since we now know that $a_0 = 0$ and so (96) is no longer useful).

Multiple-regression – Minimize the cost function Q

For the jth equation:

$$\overline{x_j y} = \sum_{i=1}^{n} a_i \overline{x_j x_i} \tag{101}$$

One can write this in matrix form as:

$$\begin{bmatrix} \overline{x_1^2} & \overline{x_1 x_2} & \overline{x_1 x_3} & \dots \\ \overline{x_2 x_1} & \overline{x_2^2} & \overline{x_2 x_3} & \dots \\ \overline{x_3 x_1} & \overline{x_3 x_2} & \overline{x_3^2} & \dots \\ \dots & & & \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \dots \end{bmatrix} = \begin{bmatrix} \overline{x_1 y} \\ \overline{x_2 y} \\ \overline{x_3 y} \\ \dots \end{bmatrix}$$

This can also be written as

$$\mathbf{C_{x_i x_j}} a_j = C_{x_i y} \tag{102}$$

Multiple-regression – Matrix form

$$\begin{bmatrix} \overline{x_1^2} & \overline{x_1 x_2} & \overline{x_1 x_3} & \dots \\ \overline{x_2 x_1} & \overline{x_2^2} & \overline{x_2 x_3} & \dots \\ \overline{x_3 x_1} & \overline{x_3 x_2} & \overline{x_3^2} & \dots \\ \dots & & & \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \dots \end{bmatrix} = \begin{bmatrix} \overline{x_1 y} \\ \overline{x_2 y} \\ \overline{x_3 y} \\ \dots \end{bmatrix}$$

This can also be written as

$$\mathbf{C_{x_i x_j}} a_j = C_{x_i y} \tag{102}$$

since $\overline{x} = 0$ for all j, the l.h.s. is the *covariance matrix* of the predictors. The diagonal elements are the *variances* of the predictors and the off-diagonal elements are the *covariances* between predictors.

the r.h.s. is the covariance vector of the predictors (x_j) and the predictand (y)

if each variable has been standardized (mean of 0, standard deviation of 1), the l.h.s. is the *correlation matrix* of x_j, and the r.h.s. is the *correlation vector*

Multiple-regression – Solving for the regression coefficients (a_j)

$$\begin{bmatrix} \overline{x_1^2} & \overline{x_1x_2} & \overline{x_1x_3} & \dots \\ \overline{x_2x_1} & \overline{x_2^2} & \overline{x_2x_3} & \dots \\ \overline{x_3x_1} & \overline{x_3x_2} & \overline{x_3^2} & \dots \\ \dots & & & \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \dots \end{bmatrix} = \begin{bmatrix} \overline{x_1y} \\ \overline{x_2y} \\ \overline{x_3y} \\ \dots \end{bmatrix}$$

This can also be written as

$$\mathbf{C_{x_ix_j}} a_j = C_{x_iy} \tag{102}$$

if the predictors are linearly independent, the off diagonal elements are all 0 and the a_j's can be found algebraically

otherwise, the a_j's are found as the inverse of the covariance matrix × the r.h.s. vector (solve for a but using matrix algebra). There are a variety of techniques for inverting the covariance matrix.

$$\mathbf{C_{x_ix_j}}^{-1}\mathbf{C_{x_ix_j}} a_j = \mathbf{C_{x_ix_j}}^{-1} C_{x_iy} \tag{103}$$

$$a_j = \mathbf{C_{x_ix_j}}^{-1} C_{x_iy} \tag{104}$$

Multiple-regression – How many variables should I use? Let's look at the simplest case with two predictors.

To make life a bit easier, let's assume that our predictors x_j and predictand y have all be standardized. Then, the normal equations for multiple linear-least-squares regression can be written in the following way:

$$r(x_i, x_j)a_i = r(x_j, y) \tag{105}$$

where r represents the correlations/covariances. Let's take the simple case where we only have two predictors:

$$\hat{y} = a_1 x_1 + a_2 x_2 \tag{106}$$

Then,

$$\begin{bmatrix} \overline{x_1^2} & \overline{x_1 x_2} \\ \overline{x_2 x_1} & \overline{x_2^2} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \overline{x_1 y} \\ \overline{x_2 y} \end{bmatrix}$$

can be re-written as

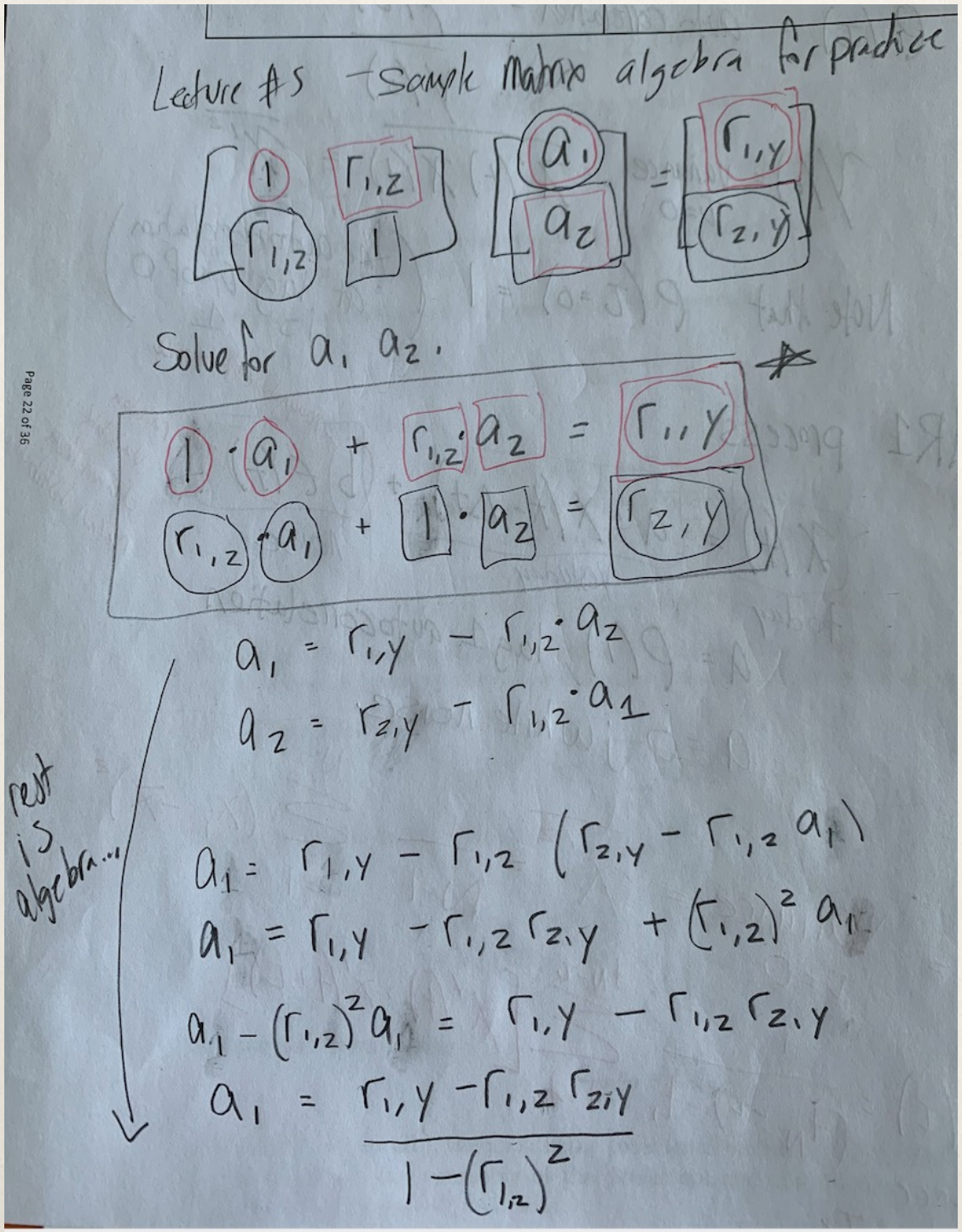
$$\begin{bmatrix} 1 & r_{1,2} \\ r_{1,2} & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} r_{1,y} \\ r_{2,y} \end{bmatrix}$$

since $r_{1,1} = r_{2,2} = 1.0$ and $r_{1,2} = r_{2,1}$.

Note: Solving for a_1, a_2 is a simple linear algebra problem…
Good review ☺ Look at it together ☺

Sample Matrix Algebra – A good quick review.

Note: mostly python will be doing the matrix math for you ☺



Solution for case with two predictors.

We solve for a_1 and a_2 and find that

$$a_1 = \frac{r_{1,y} - r_{1,2}r_{2,y}}{1 - r_{1,2}^2} \tag{107}$$

$$a_2 = \frac{r_{2,y} - r_{1,2}r_{1,y}}{1 - r_{1,2}^2} \tag{108}$$

If $\hat{y}$ is the best-fit, then we can write the explained and unexplained variance as

$$\overline{y^2} = \overline{(y_i - \hat{y})^2} + \overline{(\hat{y} - \overline{y})^2} \tag{109}$$

$$\text{Total Variance} = \text{Unexplained Variance} + \text{Explained Variance}$$

(but don't forget that $\overline{y} = 0$). Using the fact that $\hat{y} = a_1x_1 + a_2x_2$ it can be shown that

$$1 = \frac{\overline{(y_i - \hat{y})^2}}{\overline{y^2}} + R^2 \tag{110}$$

where the fraction of explained variance R^2 is given by

$$R^2 = \frac{r_{1,y}^2 + r_{2,y}^2 - 2r_{1,y}r_{2,y}r_{1,2}}{1 - r_{1,2}^2} \tag{111}$$

Ask – Does an additional predictor improve your variance explained?

To demonstrate this, let's do an example: Say you have two predictors, x_1 and x_2 and both are correlated with the predictand y at 0.5, and are correlated with each other at 0.5, that is

$$r_{1,y} = r_{2,y} = r_{1,2} = 0.5 \tag{112}$$

For the first predictor only, the variance explained is

$$R_1^2 = r_{1,y}^2 = 0.25 \tag{113}$$

Adding a second predictor, x_2, leads to

$$R_{1,2}^2 = \frac{0.5^2 + 0.5^2 - 2 \times 0.5 \times 0.5 \times 0.5}{1 - 0.5^2} = 0.33 \tag{114}$$

Thus, adding a second predictor helped explain more of the variance of y.

However, now lets assume that $r_{2,y} = 0.25$ and everything else remains the same. Then, adding the second predictor leads to

$$R_{1,2}^2 = \frac{0.5^2 + 0.25^2 - 2 \times 0.5 \times 0.25 \times 0.5}{1 - 0.5^2} = 0.25 \tag{115}$$

Thus, adding the second predictor did not increase the explained variance!

Minimum Useful Correlation

Minimum Useful Correlation

$$|r(x_2, y)|_{\text{min useful}} > |r(x_1, y) \cdot r(x_1, x_2)| \tag{116}$$

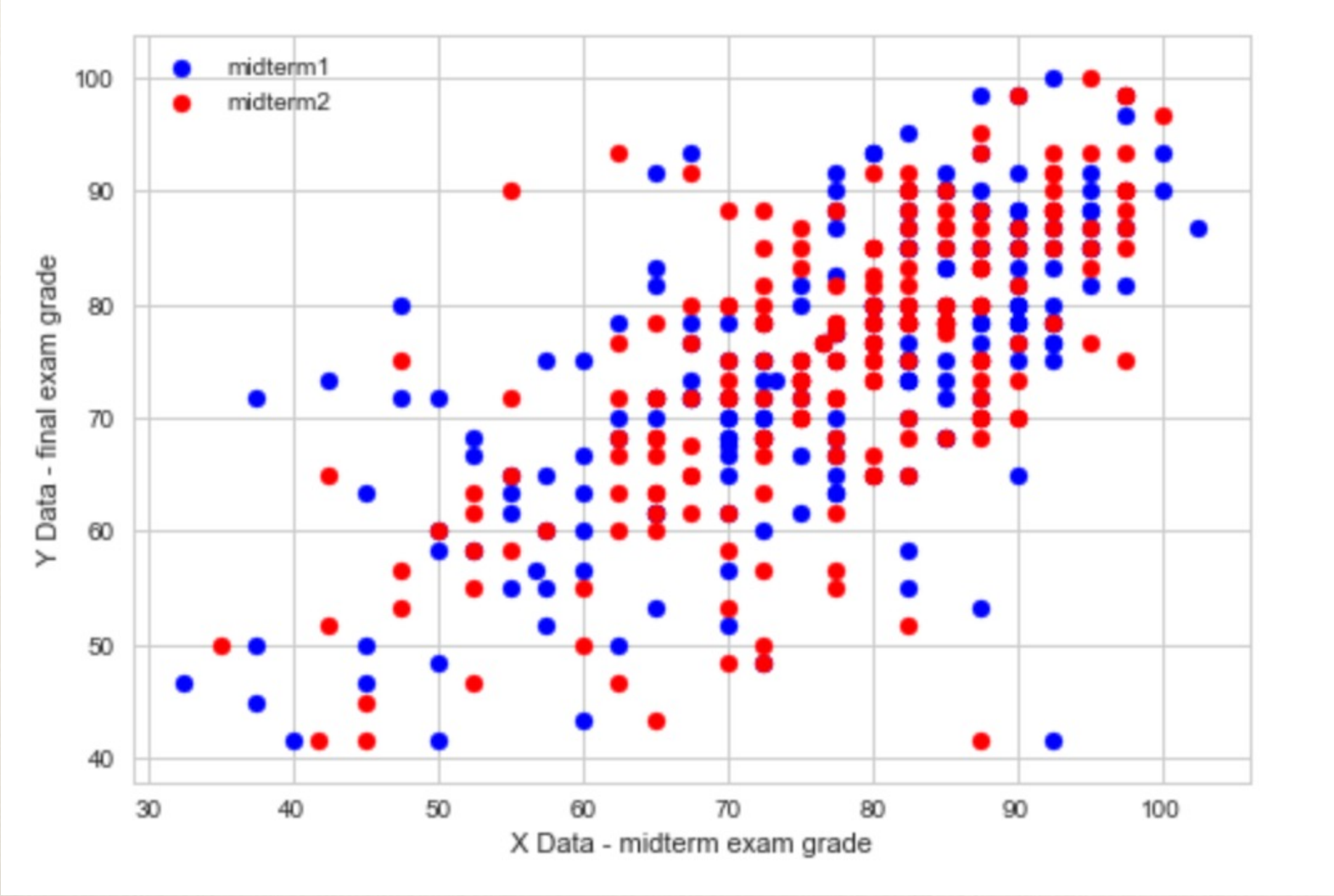
You will see that in our example above

$$|r(x_2, y)|_{\text{min useful}} = 0.5 \times 0.5 = 0.25 \tag{117}$$

If you think about it, it would be ideal for $r_{1,2} = 0$, in which case, you have two completely independent predictors. On the flip side, $r_{1,2} = 1.0$ is completely useless, since x_2 provides you with no additional information than you already got from x_1.

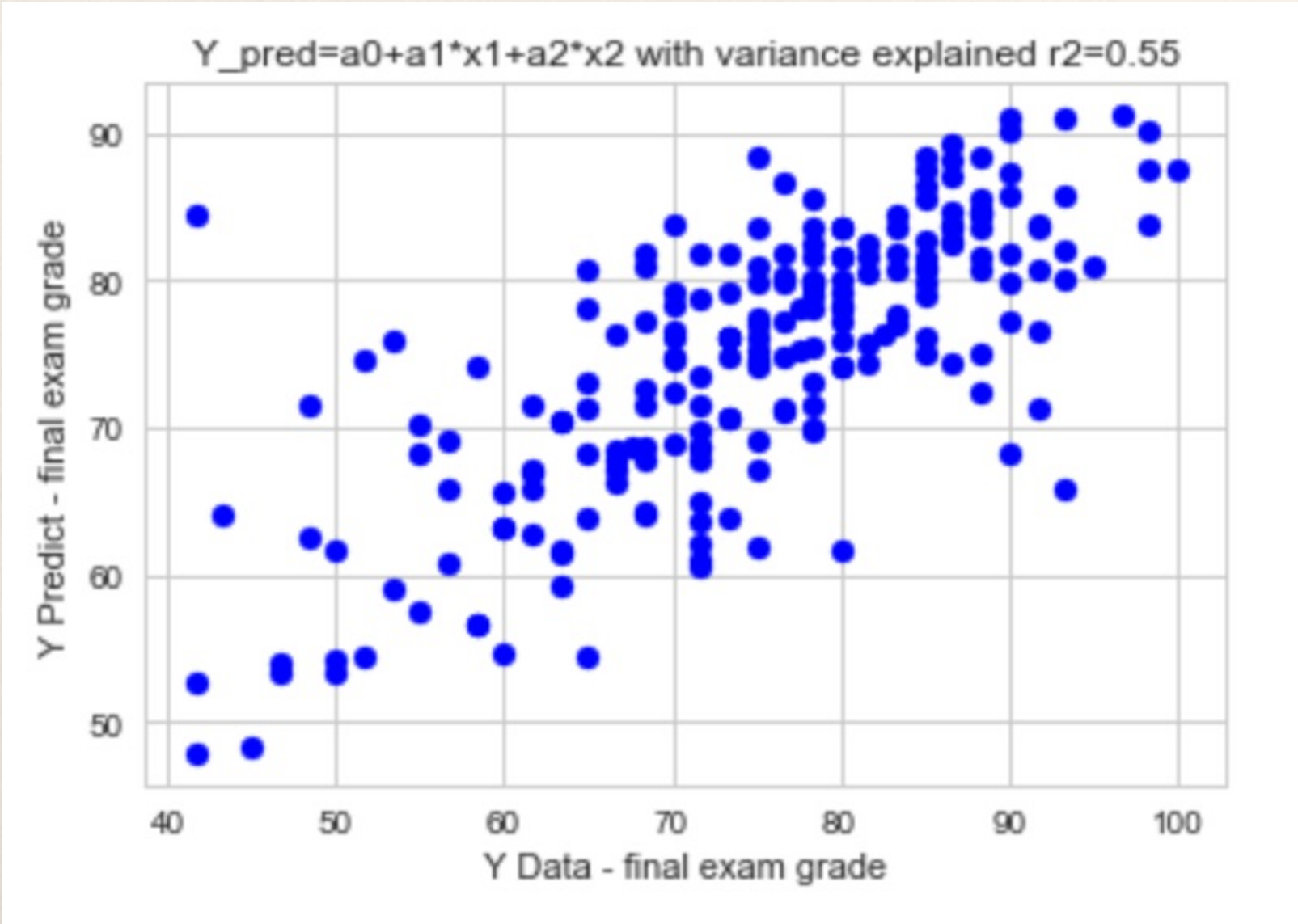
What's more, adding additional predictors can actually be detrimental overall when applying the fit to independent data. This is because you can *over fit* the data, in essence, using the predictors to fit the noise, rather than the signal only. It is a good idea to use as few predictors as possible - and test the fit on independent data after the regression coefficients have been determined. We will discuss more about how to pick optimal predictors in the future.

Minimum Useful Correlation – Another Example
Should I use midterm1 and midterm2 to predict the final exam grade?



Try the python code to answer this question –
multiple_linear_regression_grades.ipynb

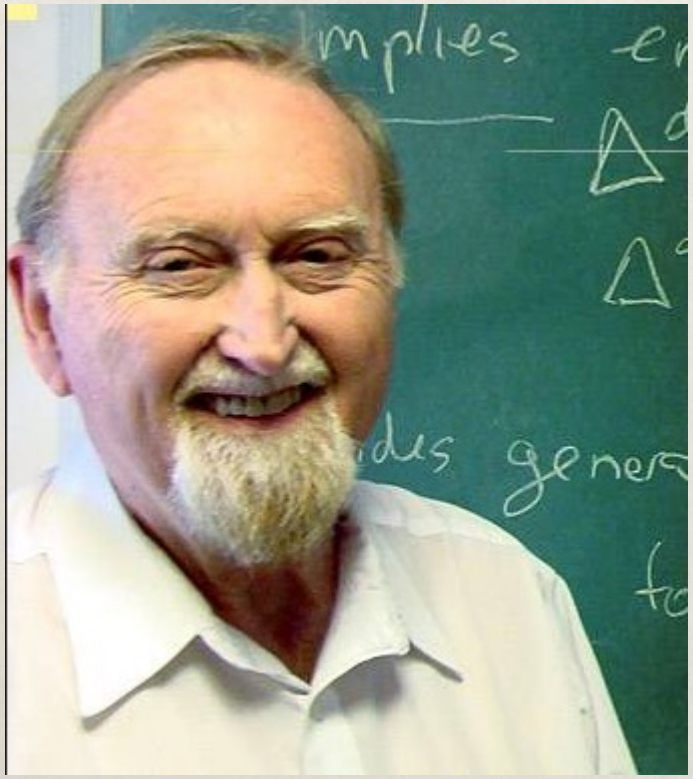
Let's use midterm1 and midterm2 to predict the final exam grade.



Try the python code – multiple_linear_regression_grades.ipynb

Granger Causality

Granger-causality is a statistical approach for determining whether one time series, x, is useful in predicting another time series y. While it has a seemingly fancy name, it is really just two multi-linear regressions. The name implies that the method can determine whether x *causes* y, however, it is important to note that it cannot do this! In addition, Granger causality cannot determine whether there is a third driver causing the other two, nor does it account for instantaneous relationships (i.e. lag zero).



Clive Granger, 2003 Nobel Prize in Economics for "methods of analyzing economic time series with common trends (cointegration)".

Granger causality is the extent to which X provides information about Y beyond what is already provided by Y itself

Step 1 assess how much y can be predicted by lagged values of itself

With these caveats out of the way, let's see what Granger causality can do. If we are interested in predicting $y(t)$, the idea is to first see how much of y_t can be predicted using only lagged values of y itself. That is, step one is to perform the following multi-linear regression

$$y_t = a_0 + a_1 y_{t-1} + a_2 y_{t-2} + ... + a_k y_{t-k} \quad (118)$$

$$= a_0 + \sum_{\tau=1}^{k} a_\tau y_{t-\tau} \quad (119)$$

where the a_is are the regression coefficients.

Step 2 assess how much y can be predicted by lagged values of x

Step two is to ask "can I obtain additional, *unique* information about y_t using lagged values of x?" That is, does adding information from x provide me with additional information about y beyond what y already contains itself? To answer this, we once again perform multi-linear regression, only now we add lagged values of x

$$y_t = b_0 + b_1 y_{t-1} + ... + b_k y_{t-k} + c_p x_{t-p} + ... + c_k x_{t-k} \quad (120)$$

$$= b_0 + \sum_{\tau=1}^{k} b_\tau y_{t-\tau} + \sum_{\tau=p}^{k} c_\tau x_{t-\tau} \quad (121)$$

In our notation, p is the smallest lag considered and k is the largest, such that $p \geqslant 1$ and $k \geqslant 1$.

Final step – Assess if x is helping to predict y

The final step is to now see if adding lagged values of $\mathbf{x}$ provided additional predictive power of $\mathbf{y}_t$. To do this, we require that two conditions are met

1. There exists at least one significant $\mathbf{c}$ according to a t-test.
2. The addition of the $\mathbf{c}$ terms collectively add power to the regression according to an F-test.

The second condition involves comparing the amount of variance explained in (119) with the amount of variance explained in (121), and as we have learned, we can do this with an F-test.

If you find that lagged values of $\mathbf{x}$ do indeed provide unique information about $\mathbf{y}_t$ and that this increase in variance explained is significant, then it is said that "$\mathbf{x}$ *Granger-causes* $\mathbf{y}$".

Recent reference on use of Granger causality in climate….

Memory Matters: A Case for Granger Causality in Climate Variability Studies

MARIE C. MCGRAW AND ELIZABETH A. BARNES

Department of Atmospheric Science, Colorado State University, Fort Collins, Colorado

(Manuscript received 19 May 2017, in final form 8 January 2018)

ABSTRACT

In climate variability studies, lagged linear regression is frequently used to infer causality. While lagged linear regression analysis can often provide valuable information about causal relationships, lagged regression is also susceptible to overreporting significant relationships when one or more of the variables has substantial memory (autocorrelation). Granger causality analysis takes into account the memory of the data and is therefore not susceptible to this issue. A simple Monte Carlo example highlights the advantages of Granger causality, compared to traditional lagged linear regression analysis in situations with one or more highly autocorrelated variables. Differences between the two approaches are further explored in two illustrative examples applicable to large-scale climate variability studies. Given that Granger causality is straightforward to calculate, Granger causality analysis may be preferable to traditional lagged regression analysis when one or more datasets has large memory.

https://doi.org/10.1175/JCLI-D-17-0334.1

Application in our field: ENSO and Surface Temp.

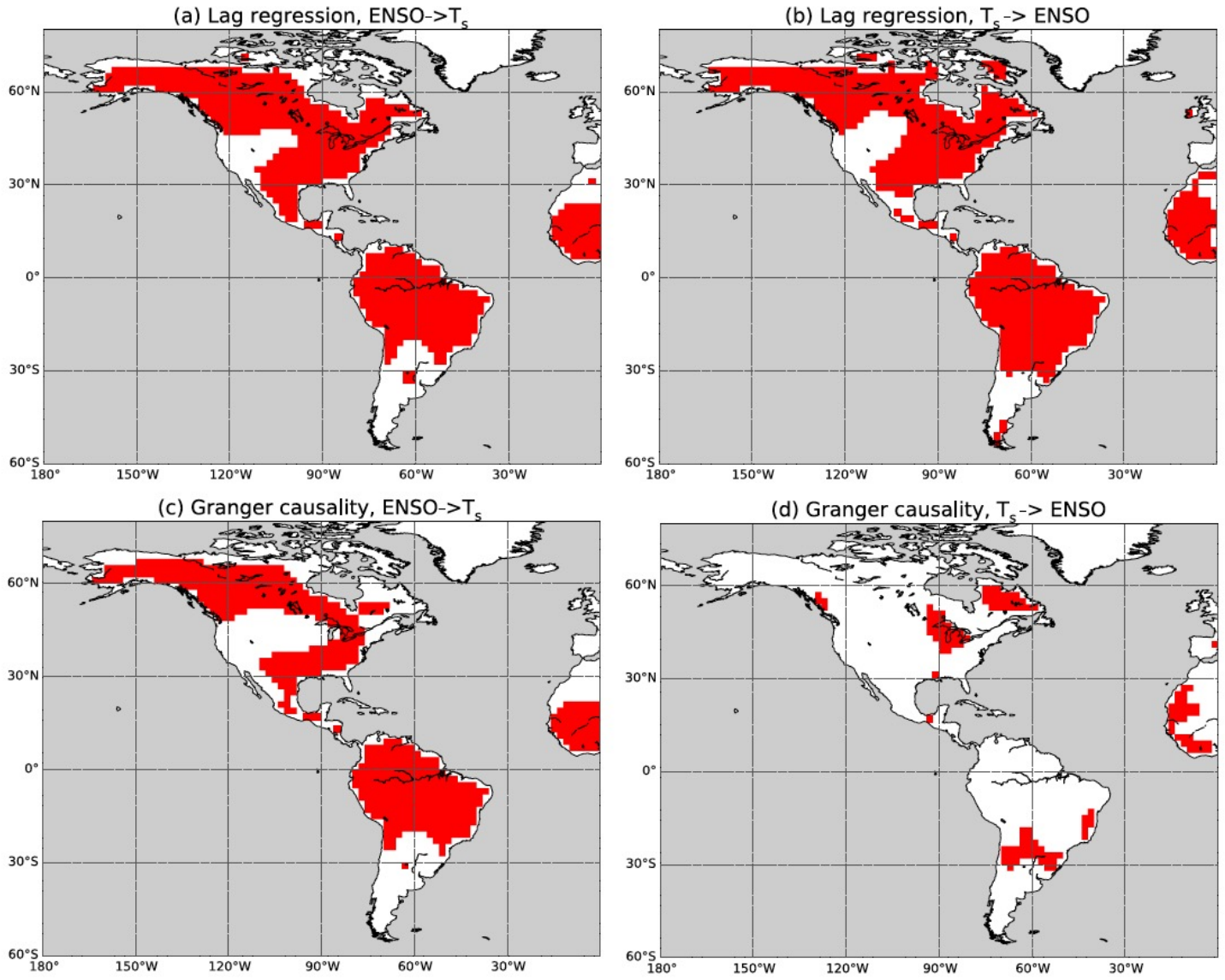


FIG. 4. Using (a),(b) lagged regression and (c),(d) Granger causality to test the hypothesis that (left) ENSO drives T_s and (right) T_s drives ENSO. Red indicates a significant lagged relationship identified at up to 7 months ($k = 7$). Significance is assessed at 95%.

https://doi.org/10.1175/JCLI-D-17-0334.1

The 1 month lag autocorrelation of ENSO (Nino 3.4 SST) is 0.91. ENSO has memory.

Granger causality accounts for the ENSO memory. It asks: ***What is the variance in ENSO to Ts, not already accounted for by ENSO itself?*** Thus, Fig. 4d shows that T_s 7 months prior have little influence on ENSO.

Conclusion: ENSO drives Ts - Ts doesn't drive ENSO.

GROUP WORK

- *multiple_linear_regression_grades.ipynb*
- *Review linear algebra practice problem*
- *Explain Granger Causality to each other in words. Can you think of a way you might use it in your research?*
- *minimum_corr_for_added_value.ipynb (if time, from Prof. Barnes)*

Next week ---

Application Lab #2 on regression, autocorrelation, and red noise time series.

Read Barnes Chapter 2 – Be ready to use concepts introduced in lectures and in this chapter in the application lab!

Be in touch if you have questions. Slack is your friend…
Be thinking about the data you want to analyze in upcoming homework, start "munging" now, and also be thinking about your paper …

So good to see you all today ☺