AP.Definition-and-First-Examples

2020年5月17日 星期日 下午10:38

_683d9f19 1bc1aa0...

Autoregressive Processes

We are looking at some general stochastic processes that are useful in understanding the driving mechanisms behind the Time Series that we encounter. We've already seen the Random Walk. We can generalize this to an autoregressive process of order p, denoted *AR(p)*. This has nothing to do with retired persons and everything to do with the formula

$$X_t = Z_t + history$$

That's a little vague, so let's spell out what we mean by "history".

- Let's take the Z_t's to be white noise $Z_t \sim iid(0, \sigma^2)$ Don't know how to model
- By *history* we mean that we include *previous terms in the process* as

$$HISTORY = \phi_1 X_{t-1} + \cdots + \phi_p X_{t-p}$$ linear combination multiply coefficient

So, we then have

$$AR(p)\ process: \qquad X_t = Z_t + \phi_1 X_{t-1} + \cdots + \phi_p X_{t-p}$$

Please compare this to the moving average process, also with $Z_t \sim iid(0, \sigma^2)$. We had

$$MA(q)\ process: \qquad X_t = \theta_0 Z_t + \theta_1 Z_{t-1} + \cdots + \theta_q Z_{t-q}$$

Punching this idea out a little,

- We build an *MA(q)* from a finite set of *innovations* (the *Z*'s)
- We build an *AR(p)* from a current innovation Z_t together with knowledge of a finite set of prior states (the X's).

As a quick and obvious example, recall the random walk. We said that our current position is the position we occupied at the previous time, plus a noise variable (we'll assume $\mu = 0$)

$$X_t = X_{t-1} + Z_t$$

So, just take $p = 1$ and $\phi_1 = 1$

$$X_t = Z_t + X_{t-1}$$

An obvious and quick caution: an autoregressive process isn't necessarily stationary!

Simulating a simple AR(p) Process: First Order

Before looking at data sets, let's develop our intuition in the clean and antiseptic environment of simulations. We can very easily simulate an *AR(p=1)* process. (Our "history" just consists of the immediately previous state, so *p=1*). The first simulation will have $\phi = 0.4$.

everybody have same data

run simulation

a family of independent r.v.

variable name X → start filling

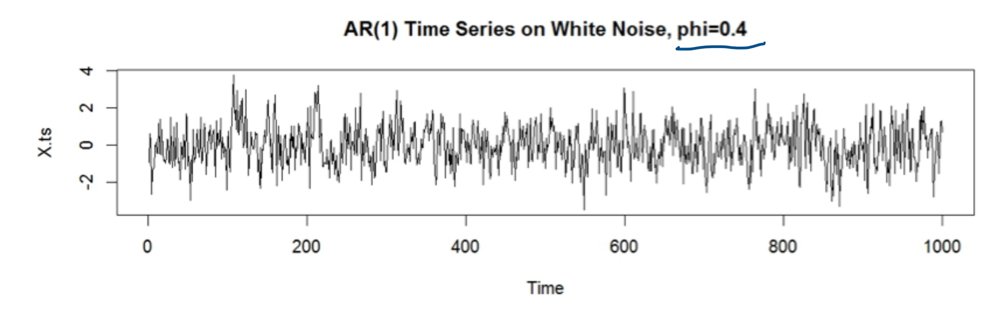
```
set.seed(2016);      N=1000;      phi = .4;
Z = rnorm(N,0,1);    X=NULL;      X[1] = Z[1];
for (t in 2:N) {
        X[t] = Z[t] + phi*X[t-1] ;
}
X.ts = ts(X)
par(mfrow=c(2,1))
plot(X.ts,main="AR(1) Time Series on White Noise, phi=.4")
X.acf = acf(X.ts, main="AR(1) Time Series on White Noise, phi=.4")
```

plots: creat t.s. object

set up plots - 2 rows, 1 col.

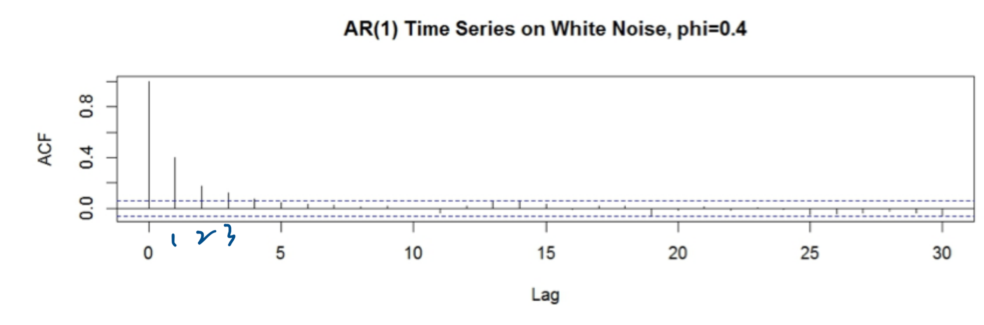
plot ts,

plot estimated ACF



some dependence upon neighbors

x just noise ✓ some correlation



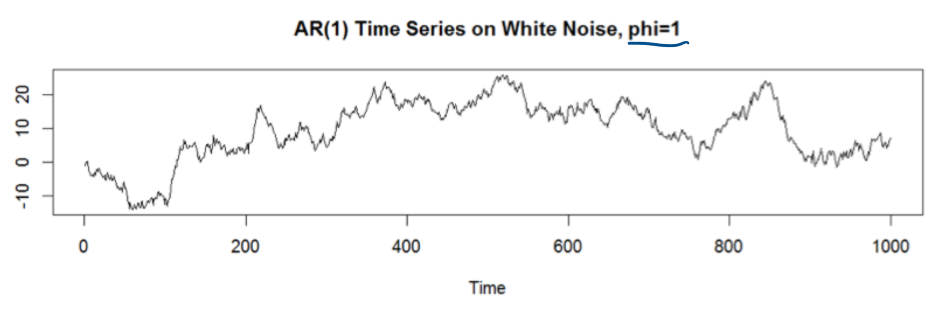
get quantitative analysis. - Acf

after 2 or 3 lags,

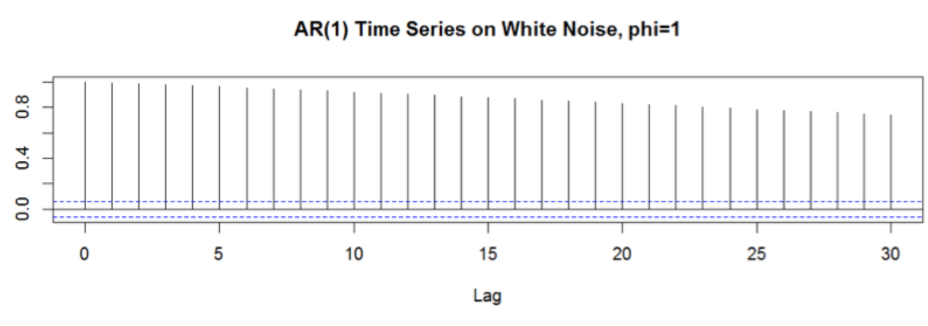
correlation get down to noise

It looks to me like the first two or three lag spacings have a significant value. What happens when we set $\phi = 1$?

This will give us a simple random walk.



strong dependence



Acf all >1

Those covariances are not dropping off as rapidly! We will develop an explicit formula for the theoretical auto-covariance in the next lecture. We can do a terrific job predicting the ACF.

Let's add additional terms in our *AR(p)* simulation. We can take advantage of a routine called *arima.sim()* from the *stats* package.

> *arima.sim(model, n, rand.gen = rnorm, innov = rand.gen(n, ...), n.start = NA, start.innov = rand.gen(n.start, ...), ...)*

We can choose several parameters and observe the resulting plots and ACF's. It's important to build a mental image library of the sorts of time plots and ACFs that we obtain by running many simulations. Let's give a little more prominence to the closest history term with

$$AR(2)\ process: \qquad X_t = Z_t + .7X_{t-1} + .2X_{t-2}$$

The call to arima.sim() is rather straightforward if we accept the defaults:

Zt + history

$\phi_1 X_{t-1} + \phi_2 X_{t-2}$

run codes, vary coefficients - make observations - how t.s. looks / ACF

```
set.seed(2017)
X.ts <- arima.sim(list(ar = c(.7, .2)), n=1000)
par(mfrow=c(2,1))
plot(X.ts,main="AR(2) Time Series, phi1=.7, phi2=.2")
X.acf = acf(X.ts, main="Autocorrelation of AR(2) Time Series")
```

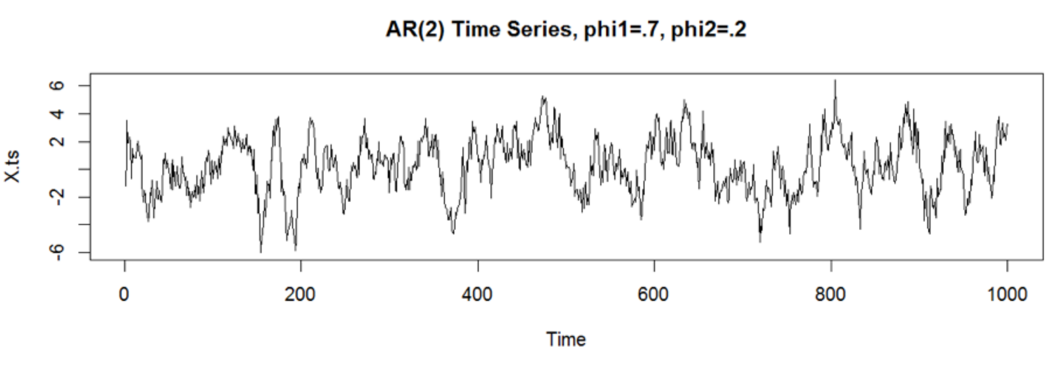
list of coefficients - give auto regressive integrated moving average simulations

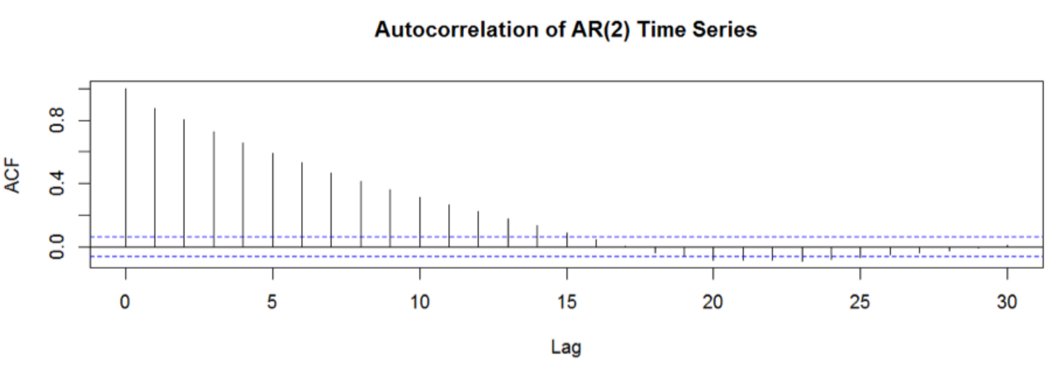
AR term

can put in AR or MA terms

plot

We're setting the seed so that you can compare your work plot directly. To obtain additional simulations, you can comment out that line. Setting up for an *MA(q)* process is also easy.





If you run several simulations, you should see that they all share some ~~common features~~ (they "rhyme" in a certain sense) but of course there is variability in the details. If you play with this code a little, you should pretty quickly get into trouble by trying to generate non-stationary

processes. We will explore conditions for stationarity later, but for now, if you'd like to stay out of trouble, just maintain:

$$\begin{cases} -1 < \phi_2 < 1 \\ \phi_2 < 1 + \phi_1 \\ \phi_2 < 1 - \phi_1 \end{cases}$$

In case those inequalities look funny to you, just remember that our parameters don't have to be positive numbers. And, as a little plotting tip, we can include our parameter values in the plot title if we use the *paste()* command. Then we don't have to keep setting values throughout the script:

```
phi1 = .5;      phi2 = -.4;
X.ts <- arima.sim(list(ar = c(phi1 , phi2)), n=1000)
par(mfrow=c(2,1))
plot(X.ts,main=paste("AR(2) Time Series, phi1=",phi1, "phi2=",phi2))
X.acf = acf(X.ts, main="Autocorrelation of AR(2) Time Series")
```

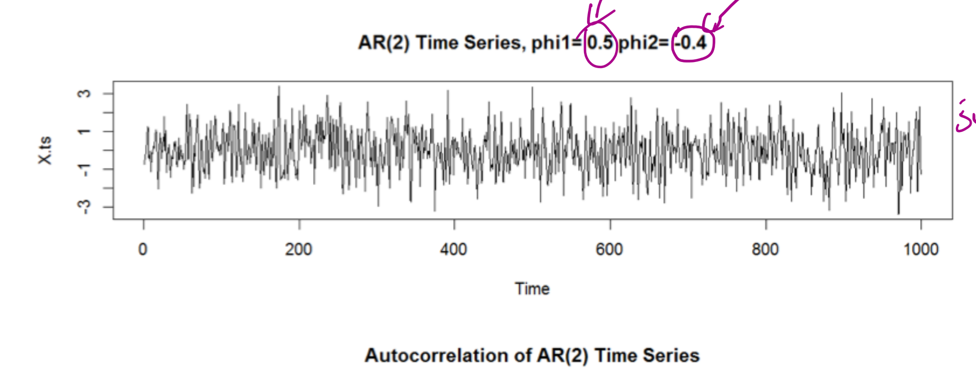
negative

do plotting - put variable into plot command

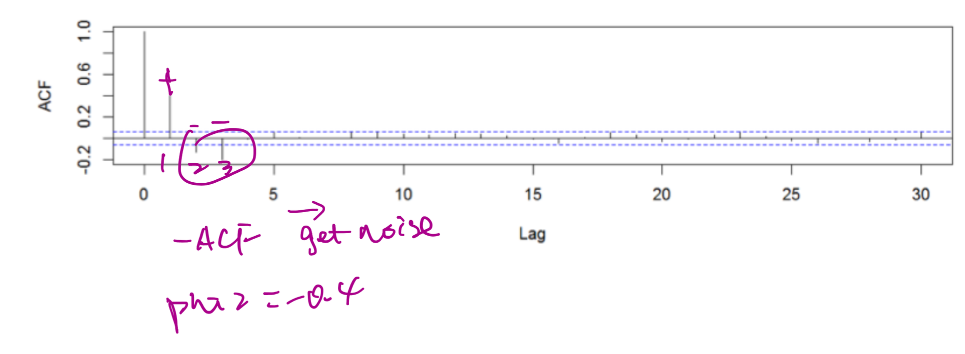
character string

put in phi1, phi2,

create nice char. → main as variable. into plot command



jump around



-ACF → get noise

phi2 = -0.4