

MAST-811: Time Series Analysis (Fall 2022)

Computer Exercise-1 (due Sept.-8, 2022)

Use a pseudo-random number generator* to generate a single sequence of $M=10,000$ numbers $x_i=x(t_i)$ where $i=1, 2, \dots, M$ with a Gaussian normal distribution that has zero mean ($\mu=0$) and a variance of 1 ($\sigma^2=1$). Interpret these numbers as hourly values of sealevel $x(t_i)$ measured in units of meters.

1. Graph (and properly label with units) the time series;
2. Calculate mean and variance of $x(t)$;
3. Calculate and graph the histogram**;
4. Overlay the histogram with the function

$$f(x,\mu,\sigma) = \exp[-(x-\mu)^2/2\sigma^2] / (2\pi \sigma^2)^{0.5}$$

5. What do you have to do to your histogram for it to match $f(x,\mu,\sigma)$? [Hint: Note that f has units, but your histogram may not.]

The excellent wiki page

https://en.wikipedia.org/wiki/Normal_distribution

provides useful pointers and detailed reminders of things you already know.

Please describe to me in written words and full sentences what you have done, how you have done it, and how you interpret the results. Please attach your programs to your write-up and properly label all plots. The plots do not have to be “fancy,” but make sure that all axes are labeled properly and that the units are correct (add by hand if necessary).

(*) Each programming language or environment has both good (and bad) random number generators. I use fortran in a command-line terminal environment on a MacBook Pro and copied the code listed below from Press et al. (1986). This reference describes details for the two routines `ran1()` and `gsdev()`. This exercise tests your random number generator that you will use throughout this class to generate known signals (sinusoids usually) with known noise (random usually).

(**) <https://en.wikipedia.org/wiki/Histogram>

MAST-811: Time Series Analysis (Fall 2022)

Code compiles and executes on OS X 11.6 with gfortran under gcc version 12.1.0 (Jan.-1, 2020)

```
cccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c Random Number Generator (from Numerical Recipes)
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccc
  function gasdev(idum)
    integer iset
    real fac, gset,r,v1,v2
c
    iset = 0
    if (iset.eq.0) then
1      v1 = 2.*ran1(idum)-1.
        v2 = 2.*ran1(idum)-1.
        r = v1*v1+v2*v2
        if (r.gt.1) goto 1
        fac = sqrt(-2.*log(r)/r)
        gset = v1*fac
        gasdev = v2*fac
        iset = 1
    else
        gasdev = gset
        iset = 0
    end if
    return
  end

  function ran1(idum)
    integer idum,ia,im,iq,ir,ntab,ndiv
    real ran1,am,eps,rnmX
    parameter(ia=16807,im=2147483647,am=1./im,iq=127773,ir=2836,
1  ntab=32,ndiv=1+(im-1)/ntab,eps=1.2e-7,rnmX=1.-eps)
    integer j,k,iv(ntab),iy
    save iv,iy
    data iv /ntab*0/, iy /0/
c
    if (idum.le.0 .or. iy.eq.0) then
        idum = max(-idum,1)
        do 11 j=ntab+8,1,-1
            k = idum/iq
            idum = ia*(idum-k*iq)-ir*k
            if (idum.lt.0) idum = idum+im
            if (j.le.ntab) iv(j) = idum
11      continue
        iy = iv(1)
    end if
    k = idum/iq
    idum = ia*(idum-k*iq)-ir*k
    if (idum.lt.0) idum = idum+im
    j = 1+iy/ndiv
    iy = iv(j)
    iv(j) = idum
    ran1 = min(am*iy,rnmX)
    return
  end
```