ExerciseTrendSine: Introduction

Download program from **Line.py** from homepage. The model is

$$g(t) = \beta_0 + \beta_1 t$$

Line.py creates simulated data with this model. Subroutines are

Data(SimN,SimDT,SimEY,SimBETA) produces simulated data

- $n =$ **SimN=20** random time points $t_i =$**T** are drawn from an even distribution between 0 and $\Delta T =$ **SimDT=5**.
- Two free parameter values are $\bar{\beta} = [\beta_1, \beta_2] =$ **SimBETA=[0.,1.]**.
- **SimN=20** random errors $\sigma_i =$ **EY** are drawn from normal distribution $N(m_y = 0, s_y = 0.1)$, where m_y is the mean and $s_y =$ **SimEY=0.5** is the standard deviation.
- Simulated data are **Y=** $y(t_i) = g(t_i) + \sigma_i =$**G+EY**

IndexSortOrder(y) gives indeces **k** that are used to re-arrange **y** values into ascending order, which in this case are the time points $t_i =$**T**.

Write1(T,Y,EY) stores simulated data to file **Line.dat**

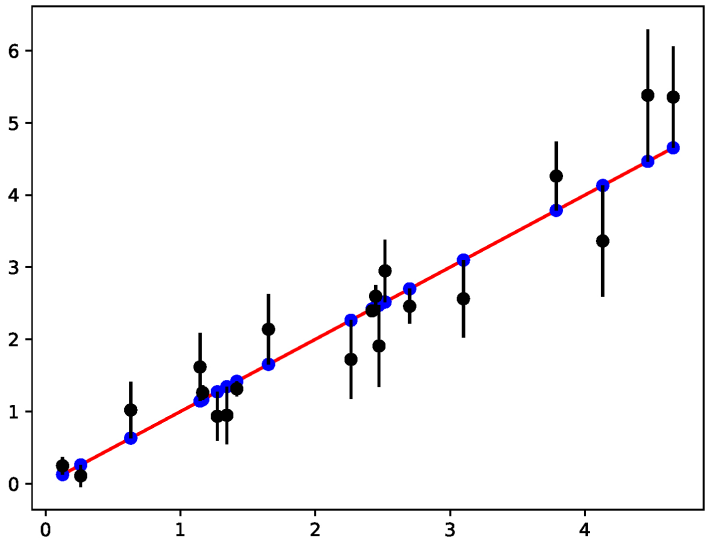
Plot1(T,Y,EY,TT,GG) plots results into **Line.eps**

```
# ============================================================
# /home/jetsu/opetus/both/programs/Line.py
# ============================================================
# - Keep computations and plots apart.
# ============================================================
import os
import numpy as np
import pylab as pl
os.system('clear')
# ============================================================
def Data(SimN,SimDT,SimEY,SimBETA):
```

```
    T=np.random.uniform(0,SimDT,SimN)
    k=IndexSortOrder(T)
    T=T[k]
    G=Model(T,SimBETA)
    EY=np.random.normal(0,SimEY,SimN)
    Y=G+EY
    Write1(T,Y,EY)
    return T,Y,EY
#==========================================================
def IndexSortOrder(y):
    k=sorted(range(len(y)), key=lambda i:y[i])
    return k
#==========================================================
def Model(T,BETA):
    G=BETA[0]+BETA[1]*T
    return G
#==========================================================
def Write1(T,Y,EY):
    file1=open('Line.dat',"w")
    for i in range(np.size(T)):
        file1.write("%10.5f %10.5f %10.5f\n" %(T[i],Y[i],EY[i]))
    file1.close()
    return
#==========================================================
def Plot1(T,Y,EY,G,TT,GG):
    pl.axes([0.1,0.1,0.8,0.8])
    pl.errorbar(T,Y,EY,fmt='ok',ms=6,zorder=2)
    pl.plot(T,G,'ob',ms=6,zorder=1)
    pl.plot(TT,GG,'r',zorder=0)
    pl.savefig('Line.eps')
    return
#==========================================================
#                        Main program
#  - Computations ========================================
SimN=20
```

```
SimDT=5.
SimEY=0.5
SimBETA=[0.,1.]
T,Y,EY=Data(SimN,SimDT,SimEY,SimBETA)
G=Model(T,SimBETA)
TT=np.min(T)+(np.arange(101)/100.)*(np.max(T)-np.min(T))
GG=Model(TT,SimBETA)
#- Plots ==========
Plot1(T,Y,EY,G,TT,GG)
```

One example of figure, **Line.eps**, produced by **Line.py**. Note that **Line.py** always produces a different figure, because it always produces a different sample of random data.



Symbols are $g(t_i) = g_i =$ **G**= blue circles, $y(t_i) =$ **Y** =black circles, $\sigma_i =$**EY**= vertical bars and $g(t) =$ **GG** = continuous red curve.

ExerciseTrendSine: Problem

The model

$$g(t) = \beta_1 + \beta_2(t - t_1)/(t_n - t_1) + \beta_3 \sin[2\pi(t - \beta_4)/\beta_5].$$

is a sum of a linear trend and a sinusoidal signal. The free parameters are

β_1 = trend level

β_2 = trend slope

β_3 = signal amplitude

β_4 = signal epoch

β_5 = signal period

Edit a **python** program **ExerciseTrendSine.py** that produces $n = 100$ simulated observations

$$y(t_i) = g(t_i) + \sigma_i,$$

where $\bar{\beta} = [-5, 10, 1, 2, 3]$. Draw simulated random time points t_i from a uniform distibution $U(0, \Delta T, n)$, where $\Delta T = 20$. Draw random errors σ_i from a normal distribution $N(0, 0.5, n)$.

Store your simulated data into file **TrendSine.dat**. Show the data (black dots and error bars) and the model (continuous red line curve) in figure **TrendSine.eps**. Your figure should resemble the one shown below. Send your files **ExerciseTrendSine.py** and **TrendSine.eps** to the assistant.