This test consists in analyzing the synthetic data 'final_file.csv' provide with this document. The data consists in 50 columns, from 'c0' to 'c49' and 100 time steps (rows). Figure 1 depicts the data (left panel) where for each time step we have 50 noisy observations of the true function relating the time index and the corresponding function value. The true underlying function used to generate this data is represented by the straight line in the center of the colored balls. Each color represents one of the 50 realizations (columns). The right panel shows how the additive noise standard deviation (STD) varies with time.

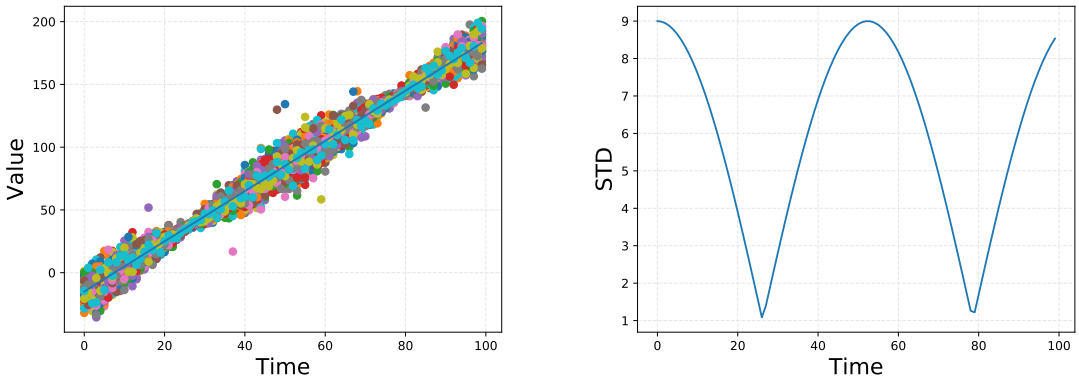


Figure 1: Noisy data observations (left) and (right) standard deviation evolution over time.

This final exam consists in loading and processing this data. For this we will use pandas, numpy, matplotlib (or any other plotting module). Except for item 1. bellow all items must be implemented in a method!

1. (15.00 pts) Load data to a dataframe.
2. (25.00 pts) Missing values and data imputation:

   Localize missing data and perform data imputation. Whenever a missing value is localized at a particular time index $t$, data imputation should be performed by replacing the missing value by the mean value across all different realizations for the same time index $t$.

   **You must:**

   - Create a method that accepts a data frame as their inputs and replace all NaN's as discussed above.
3. (25.00 pts) Identify outliers: Compute statistics for each time index (mean and variance) and see if any data point is an outlier. Then, you should implement two possible, and electable, strategies.

   (a) discard the record, that is, the whole column containing an outlier.

   (b) Replace it by the mean.

   **You must:**

   - Create a method that accepts a data frame, and a method selection variable as its inputs and perform the outlier treatment.

     e.g.:

```
def treat_outlier(df, method='discard'):
    code here...
```

4. (15.00 pts) Plot the estimated STD for all time steps and see if it resembles the right-hand panel in Figure 1. This must also be implemented in a method.

5. (20.00 pts) Linear Curve Fitting. In this item we seek to fit a linear model to the data in Figure 1. For this, lets assume a model such as

$$y_i(t) = at + b + w_i(t), \quad i = 0, \ldots, 49. \tag{1}$$

where $a$ and $b$ are the coefficients of the model, $t = [0, 1, \ldots, 99]$ is the time index and $w_i(t)$ is a zero-mean independent additive random noise at each time $t$ and realization $i$. Finally, $y_i(t)$ is the $i$-th observation of our noisy model. The model can be re-written in a vector form as

$$\begin{aligned} \boldsymbol{y}_i &= a\boldsymbol{t} + b + \boldsymbol{w}_i \\ &= \boldsymbol{H}\boldsymbol{\theta} + \boldsymbol{w}_i \end{aligned} \tag{2}$$

with $\boldsymbol{\theta}_i = [a, b]^\top$, $\boldsymbol{H} = [\boldsymbol{t}, \boldsymbol{1}]$, $\boldsymbol{t} = [0, \ldots, 99]^\top$, $\boldsymbol{1} = [1, \ldots, 1]^\top \in \mathbb{R}^{100}$, and $\boldsymbol{w}_i = [w_i(0), \ldots, w_i(99)]^\top$.

To estimate the parameters of our linear model we can resort to the regularized least-squares method which results in the following solution for the $i$-th data observation $\boldsymbol{y}_i$:

$$\hat{\boldsymbol{\theta}}_i = \left(\boldsymbol{H}^\top \boldsymbol{H} + \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{H}^\top \boldsymbol{y}_i \tag{3}$$

where $\hat{\boldsymbol{\theta}}_i = [\hat{a}, \hat{b}]^\top$, $\boldsymbol{H} = [\boldsymbol{t}, \boldsymbol{1}]$, and $\lambda = 0.01$ is a regularization constant.

Tasks:

(a) create a function that receives as inputs the pandas data frame and lambda and returns an numpy array with the average and STD of the $\hat{\boldsymbol{\theta}}_i$'s, $i = 0, \ldots, N$, where $N$ is the number of columns of the data frame. So, this function should return 4 values the mean and STD of $\hat{a}$ and $\hat{b}$.

(b) The function must perform the operation shown in Eq. (3) for all $N$ data observations. And them compute the necessary means and STDs.

(c) Finally, plot a 2D Gaussian PDF using the mean $\mu_{\text{Gauss}} = [\text{average}(\hat{a}), \text{average}(\hat{b})]^\top$ and Covariance matrix

$$C_{\boldsymbol{\theta}} = \begin{pmatrix} \text{var}(\hat{a}) & 0 \\ 0 & \text{var}(\hat{b}) \end{pmatrix}.$$

(d) Plot the line using the estimated parameters, that is

$$\hat{y}(t) = \hat{a}t + \hat{b}$$

for $t = [0, 1, \ldots, 100]$.