

➤ 原始数据 + 标准差： `shape=(n_cfg, nt)`

```
data_cv = np.mean(data_re, axis=0)
data_err = np.std(data_re, axis=0) / np.sqrt(n_cfg)
```

- 我们的误差用的是 standard error, std 是 standard deviation, 因此需要 np.std 再除以 sqrt(n_cfg), 如果做无偏, 就是除 sqrt(n_cfg-1), 但因为 n_cfg 通常也比较大, 这点区别倒是可以忽略。
- https://en.wikipedia.org/wiki/Standard_error

➤ 原始数据 + 协方差：

协方差的定义：

```
def covmatrix(self, data): # 求协方差矩阵, cfg 在第 0 轴
    n_measure = data.shape[0] # 测量数就是组态的数目
    xavg = self.confmean(data)
    cov = data - np.array([xavg] * n_measure, float)
    cov = np.matmul(cov.T, cov)
    return cov
```

利用协方差求出的误差（常用在拟合中）

```
data_err = np.sqrt(at.covmatrix(data).diagonal()) / n_cfg
```

- 协方差对角化以后再開根号之后应该和 standard error 值是一样的, 但是为什么除以 n_cfg 呢？

➤ Jackknife + 标准差/协方差：

Jackknife 重采样：

```
def jackknife(data): #data shape: (n_conf * n_t)
    nf, nt = data.shape #data shape: (n nf * n_t)
    cv = np.mean(data, 0) #将所有组态平均, cv shape: (1 * nt)
    cv = np.broadcast_to(cv, (nf, nt)) #广播将 cv 重新扩充为 data shape(n_cnf* n_t)
    jac = (nf * cv - data) / nf #(mean[N,:]*n_conf - data[N,:])/n_conf 等价于每次抽出 1 个数做平均
    return jac #jac shape: (n_conf * n_t)
```

这样做了 jackknife 以后, 利用标准差和协方差求出的误差分别是：

```
data = at.jackknife(data)
data_cv = np.mean(data, axis=0)
data_err = np.std(data, axis=0) * np.sqrt(n_cfg)
data_err = np.sqrt(at.covmatrix(data).diagonal())
```

➤ **Bootstrap + 标准差/协方差：**

```
def bootstrap(self, data, N_resample, resamp_list='default'):
    nf, nt = data.shape
    if(resamp_list == 'default'):
        resamp_list = np.random.randint(nf, size = N_resample * nf)
    if(resamp_list.shape[0]!=data.shape[0]*N_resample):
        raise IndexError("resamp_list 的数目必须等于组态数乘以 N_resample!")

    dataMat = data[resamp_list,:]
    dataMat = np.reshape(dataMat, (N_resample, nf, nt))
    return np.mean(dataMat, axis=1)
```

做过 bootstrap 后，用 np.std 求出的误差是：

```
data = at.bootstrap(data, N_resample=1000)
data_cv = np.mean(data, axis=0)
data_err = np.std(data, axis=0)
```

原则上做过 bootstrap 的数据用 np.std 会稍微有一点点问题,就是是平均数换成了中位数,所以会略微有区别,但这个的区别真的很小。完全正确的做法是这样的:(晋琛)

```
data_re_cv = np.array([val.mean for val in gv.dataset.avg_data(data_re[:, :], bstrap=True)]) # shape=(nt, )
data_re_err = np.array([val.sdev for val in gv.dataset.avg_data(data_re[:, :], bstrap=True)]) # shape=(nt, )
```

而用协方差作出的误差是：

```
data_err = np.sqrt(at.covmatrix(data).diagonal()) / np.sqrt(N_resample)
```