

Kernel: SageMath 9.5

```

In [1]: import scipy.stats
import json

def read(d,n,p=32749):
    return json.loads(open(f'data/v-d{d}n{n}p{p}.json').read())

def pl_hist(d,n,bins=100,p=32749):
    v = stats.TimeSeries(read(d,n,p))
    sigma = v.standard_deviation()
    T = RealDistribution('gaussian', sigma)
    return histogram(v, bins=bins, density=True, frame=True) + T.plot(-3*sigma,
3*sigma, color='red', thickness=2, linestyle='--')

def pl_mean(d,n,p=32749):
    w = stats.TimeSeries(read(d,n,p))
    return stats.TimeSeries([w[:i].mean() for i in
range(100,len(w))]).plot(plot_points=n, frame=True)

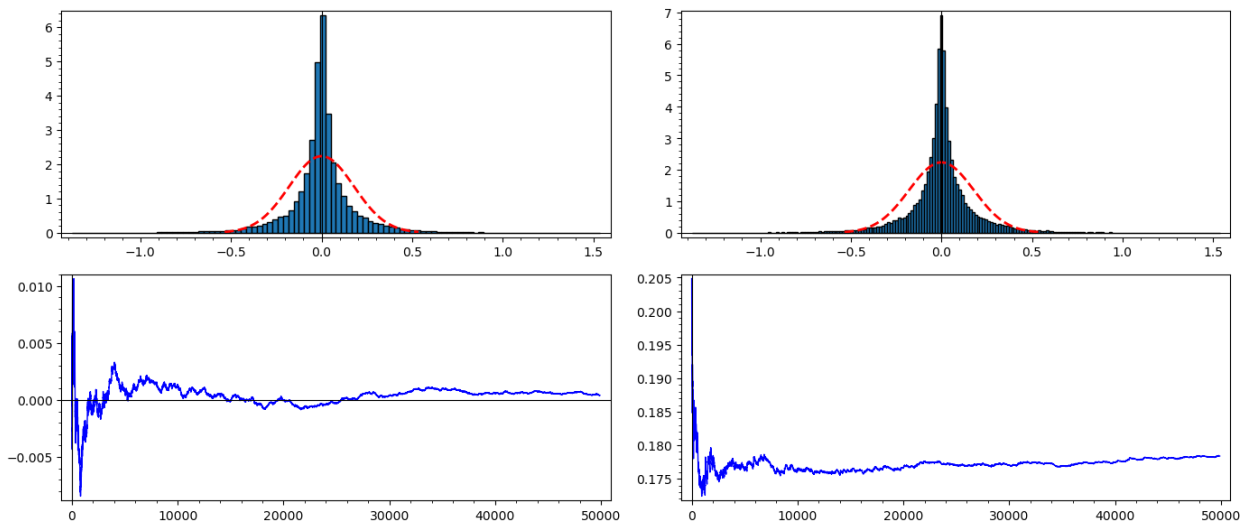
def pl_sd(d,n,p=32749):
    w = stats.TimeSeries(read(d,n,p))
    return stats.TimeSeries([w[:i].standard_deviation() for i in
range(100,len(w))]).plot(plot_points=n, frame=True)

def pl(d,n,p=32749):
    print(f"d={d}, numCoeffs={n}")
    print(scipy.stats.normaltest(read(d,n,p))) # "If the p-val is very small, it
means it is unlikely that the data came from a normal distribution."
    show(graphics_array([[pl_hist(d,n), pl_hist(d,n,200)], [pl_mean(d,n),
pl_sd(d,n)]]), figsize=[14,6])

```

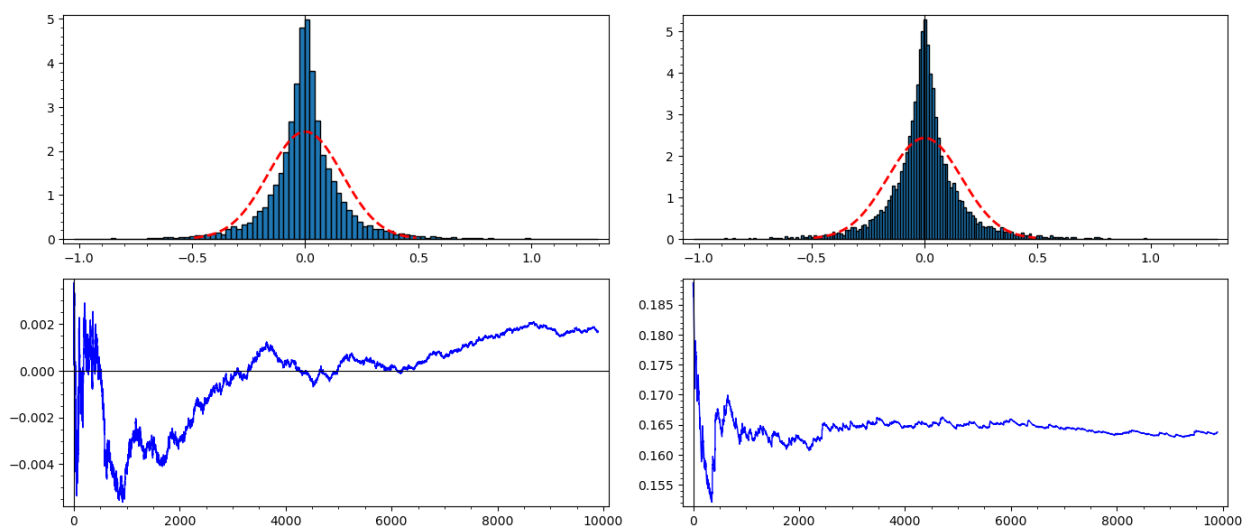
```
In [2]: pl(3,50000)
```

```
Out[2]: d=3, numCoeffs=50000
NormaltestResult(statistic=6294.089468440762, pvalue=0.0)
```



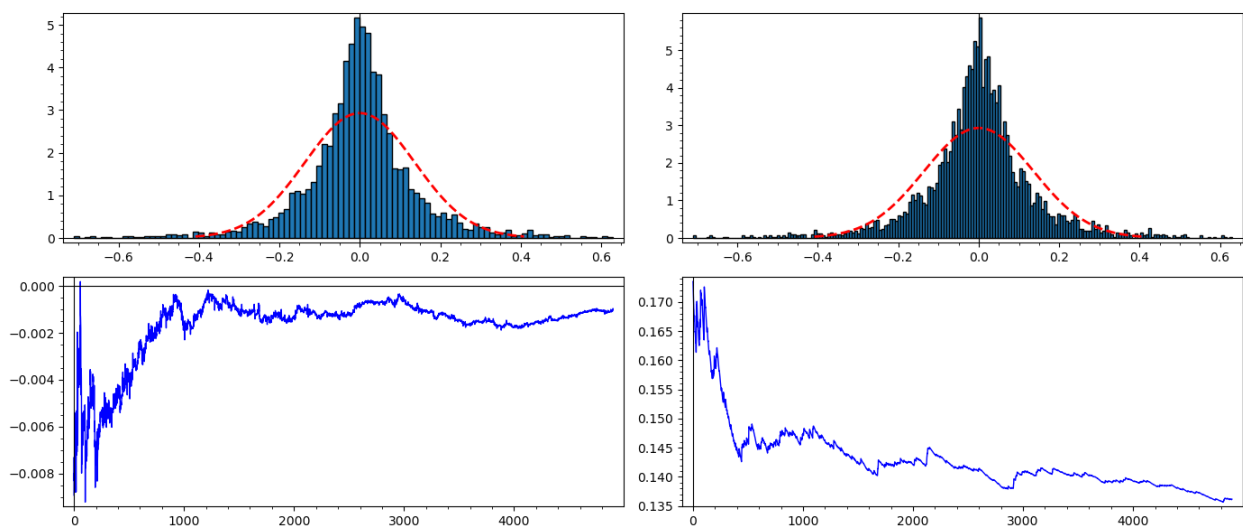
```
In [3]: pl(5,10000)
```

```
Out[3]: d=5, numCoeffs=10000
NormaltestResult(statistic=1176.2257897861243, pvalue=3.853141977071522e-256)
```



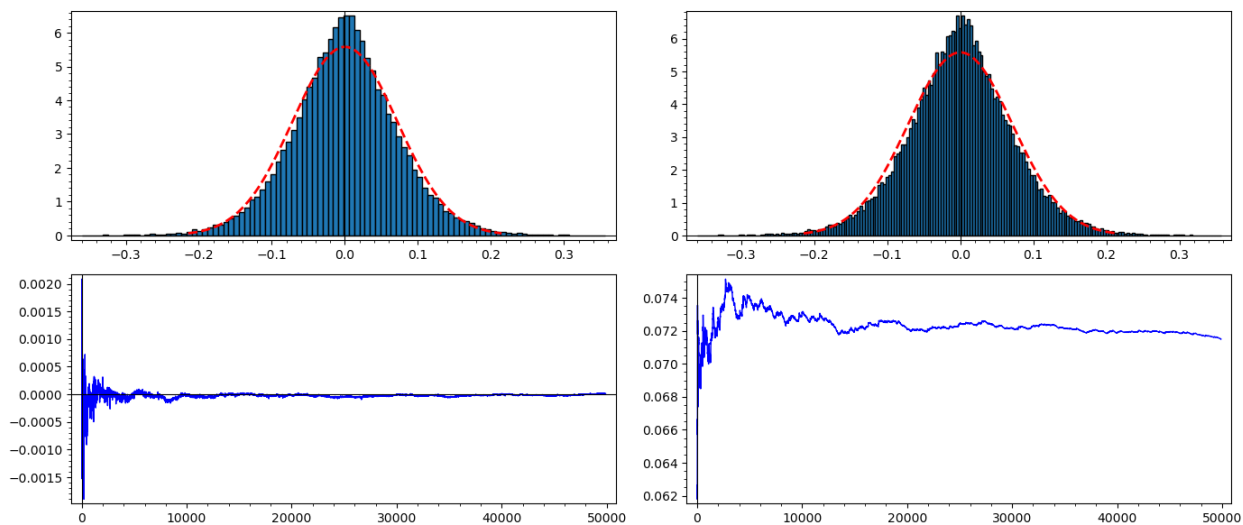
In [4]: `pl(7,5000)`

Out[4]: `d=7, numCoeffs=5000`
`NormaltestResult(statistic=426.5454349237923, pvalue=2.38141818566628e-93)`



In [5]: `pl(37,50000)`

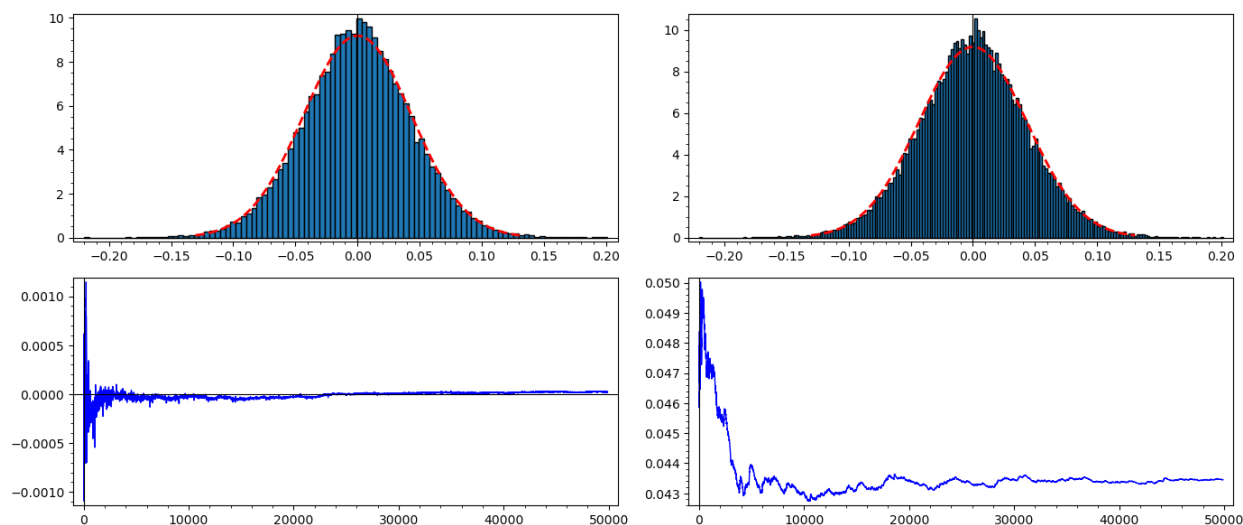
Out[5]: `d=37, numCoeffs=50000`
`NormaltestResult(statistic=678.8292199284144, pvalue=3.9274242304736496e-148)`



In [6]: `pl(101,50000)`

Out[6]: d=101, numCoeffs=50000

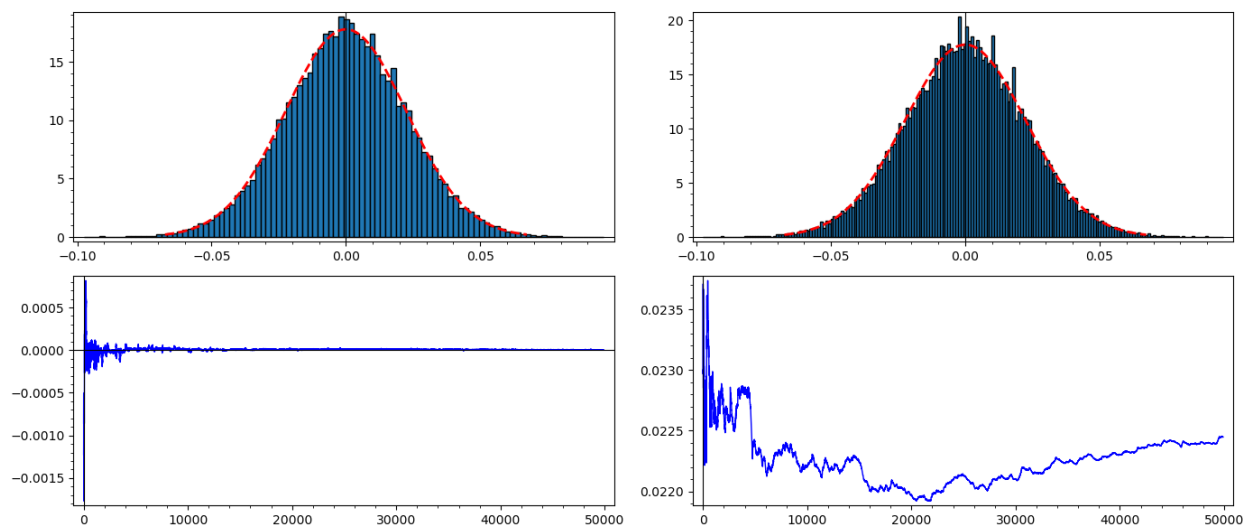
NormaltestResult(statistic=145.2788554659887, pvalue=2.8385553489804307e-32)



In [7]: `pl(389,50000)`

Out[7]: d=389, numCoeffs=50000

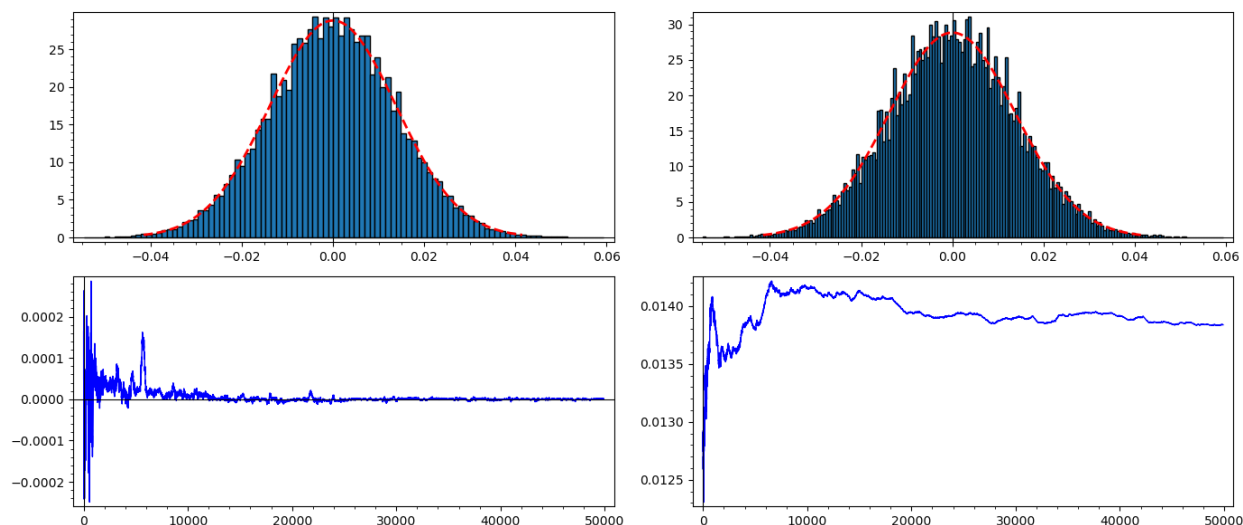
NormaltestResult(statistic=17.02363354145368, pvalue=0.00020107817991513524)



In [8]: `pl(1009,50000)`

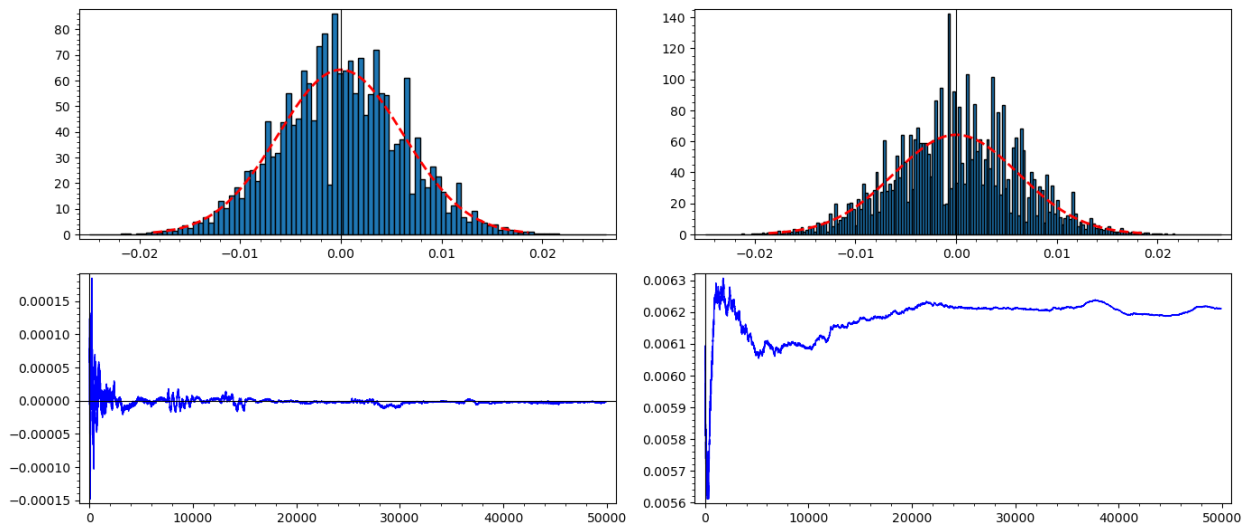
Out[8]: d=1009, numCoeffs=50000

NormaltestResult(statistic=1.7530683719862625, pvalue=0.4162229661456852)



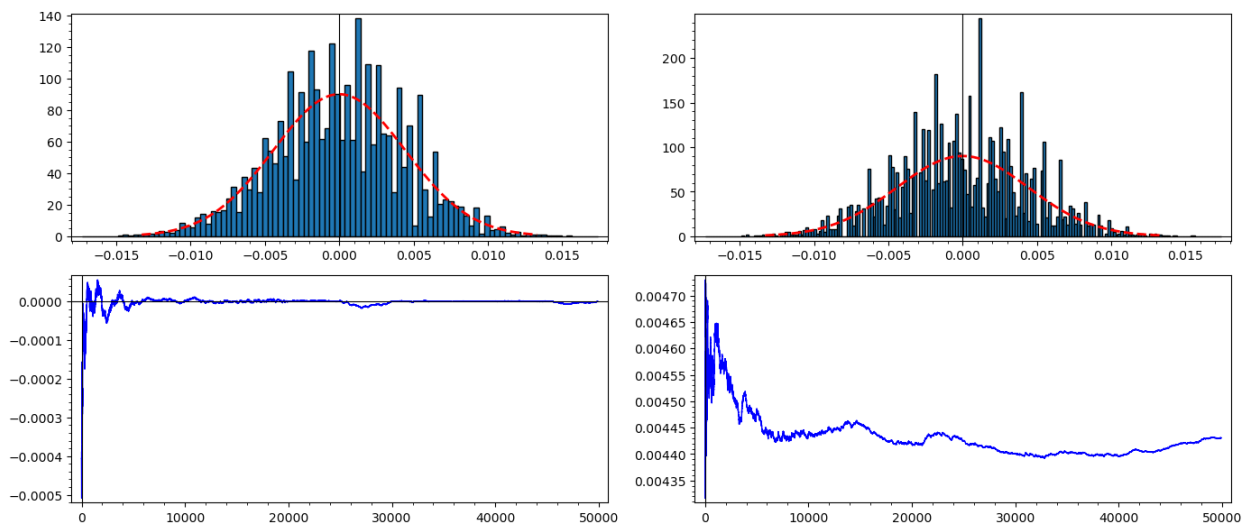
In [9]: `pl(5077,50000)`

Out[9]: `d=5077, numCoeffs=50000`
`NormaltestResult(statistic=0.4437291584224196, pvalue=0.8010238322692923)`



In [10]: `pl(10007,50000)`

Out[10]: `d=10007, numCoeffs=50000`
`NormaltestResult(statistic=0.5919164310211551, pvalue=0.7438185074474128)`



In [0]: