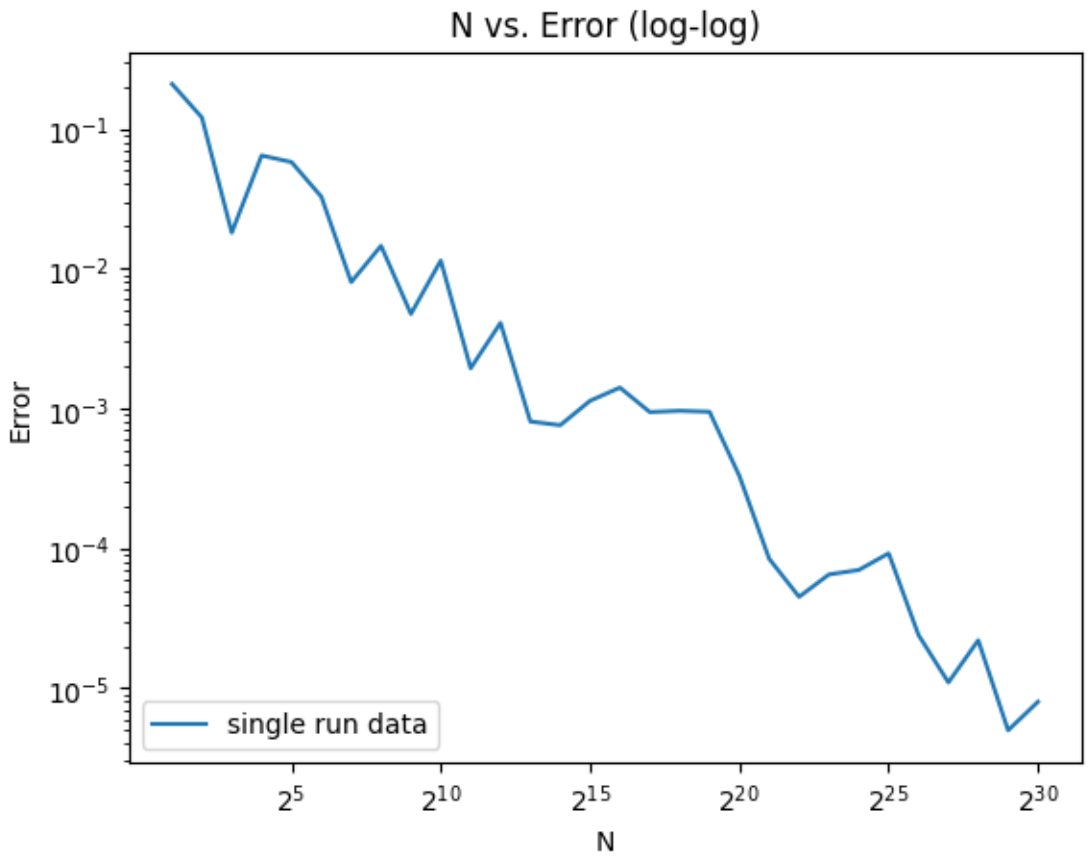
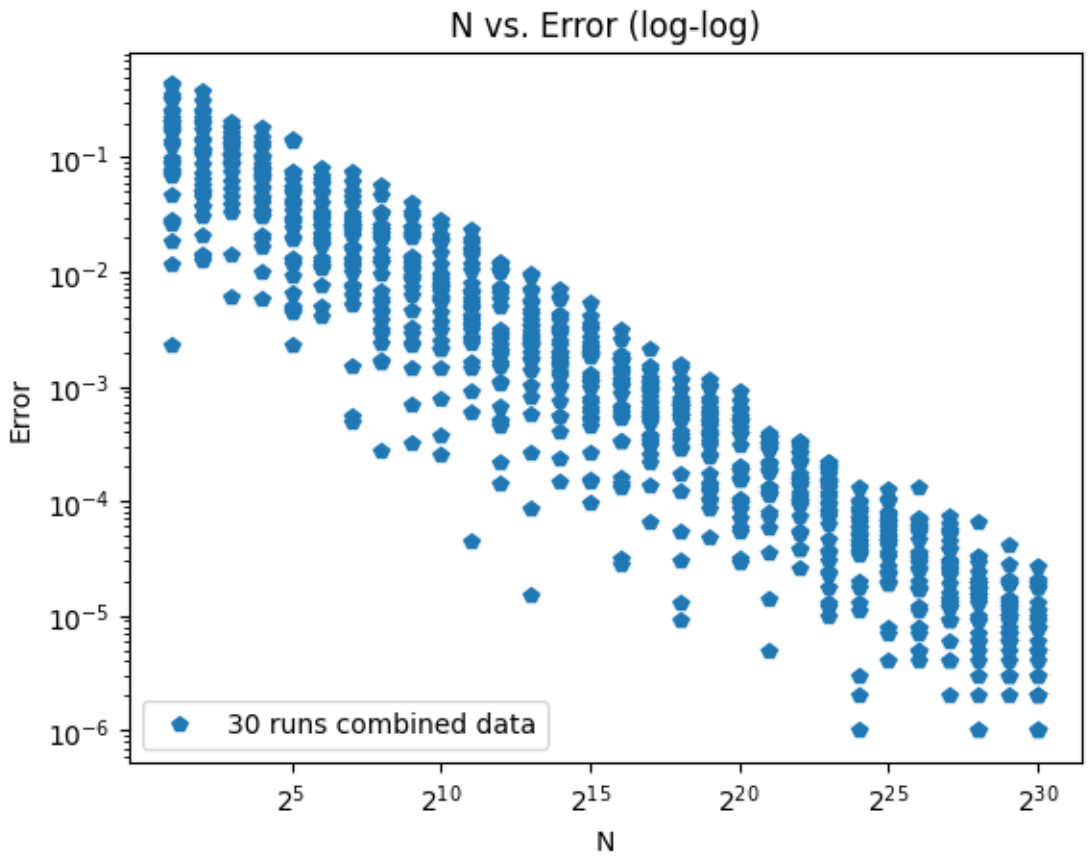
# Plts_hw1

October 23, 2023

[1]:
```
import numpy as np
from matplotlib import pyplot as plt

data_1 = np.loadtxt("error.dat")

powers_of_2 = [5, 10, 15, 20, 25, 30]
x_ticks = [2**i for i in powers_of_2]
x_labels = [r'$2^{%d}$' % i for i in powers_of_2]

plt.loglog(data_1[:,0], data_1[:,1],label='single run data')
plt.xticks(x_ticks, x_labels)
plt.legend(loc=3)
plt.xlabel("N")
plt.ylabel("Error")
plt.title('N vs. Error (log-log)')
plt.savefig("d")
```

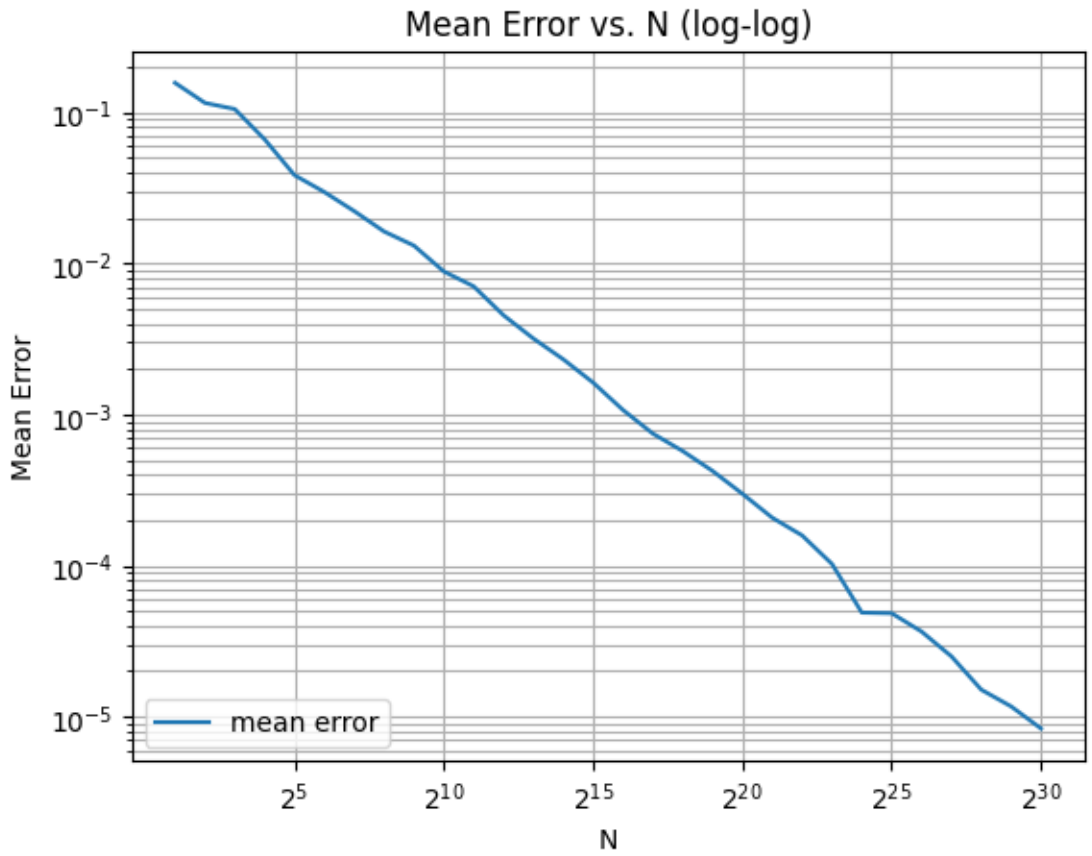



[2]:
```
import numpy as np
from matplotlib import pyplot as plt

data_2 = np.loadtxt("error_30.dat")

powers_of_2 = [5, 10, 15, 20, 25, 30]
x_ticks = [2**i for i in powers_of_2]
x_labels = [r'$2^{%d}$' % i for i in powers_of_2]

plt.loglog(data_2[:,0], data_2[:,1],'p',label='30 runs combined data')
plt.xticks(x_ticks, x_labels)
plt.legend(loc=3)
plt.xlabel("N")
plt.ylabel("Error")
plt.title('N vs. Error (log-log)')
plt.savefig("e")
```



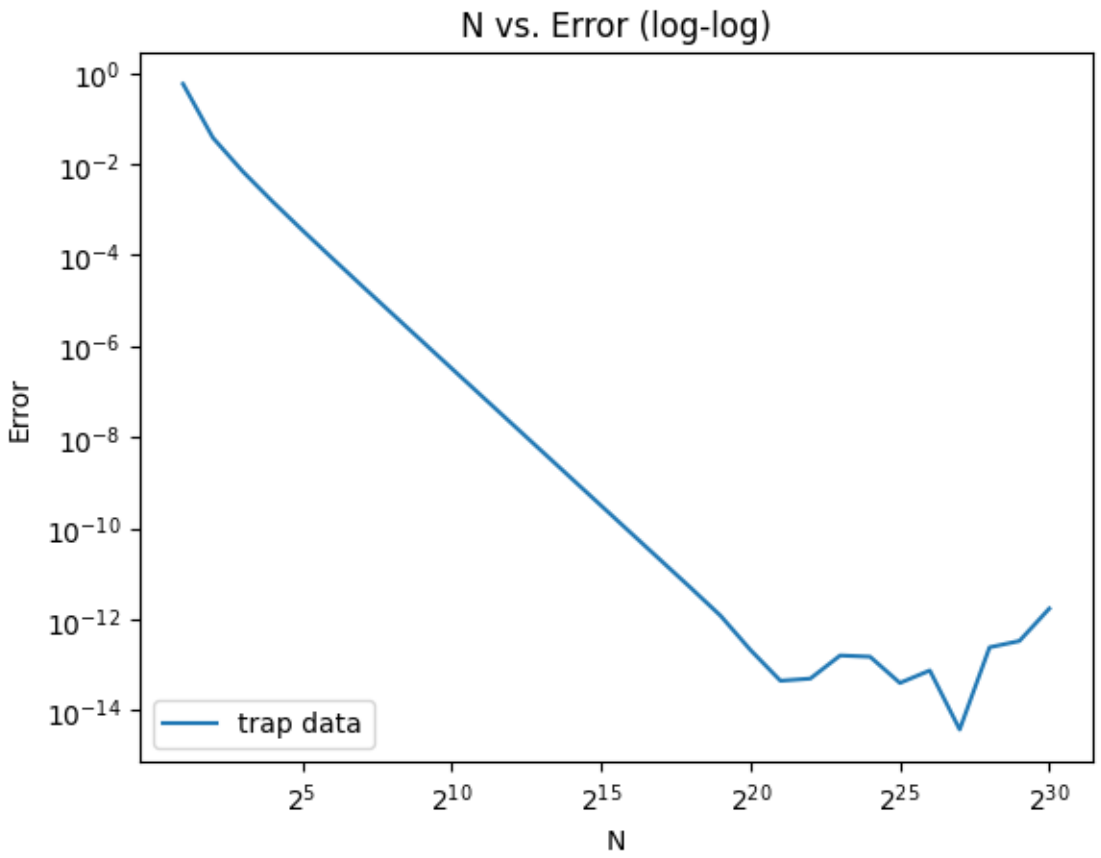

[3]:
```
import numpy as np
import matplotlib.pyplot as plt

data_2 = np.loadtxt('error_30.dat')

N_i = data_2[:, 0]
error = data_2[:, 1]

unique_N = []
mean_error = []

for N in np.unique(N_i):
    error_N = error[N_i == N]
    mean_error_N = np.mean(error_N)
    unique_N.append(N)
    mean_error.append(mean_error_N)

unique_N = np.array(unique_N)
mean_error = np.array(mean_error)
```

```
powers_of_2 = [5, 10, 15, 20, 25, 30]
x_ticks = [2**i for i in powers_of_2]
x_labels = [r'$2^{%d}$' % i for i in powers_of_2]

plt.loglog(unique_N, mean_error, label='mean error')
plt.xticks(x_ticks, x_labels)
plt.legend(loc=3)
plt.grid(True, which='both', axis='both')
plt.xlabel('N'); plt.ylabel('Mean Error'); plt.title('Mean Error vs. N␣
 ↪(log-log)')
plt.savefig("f")
```



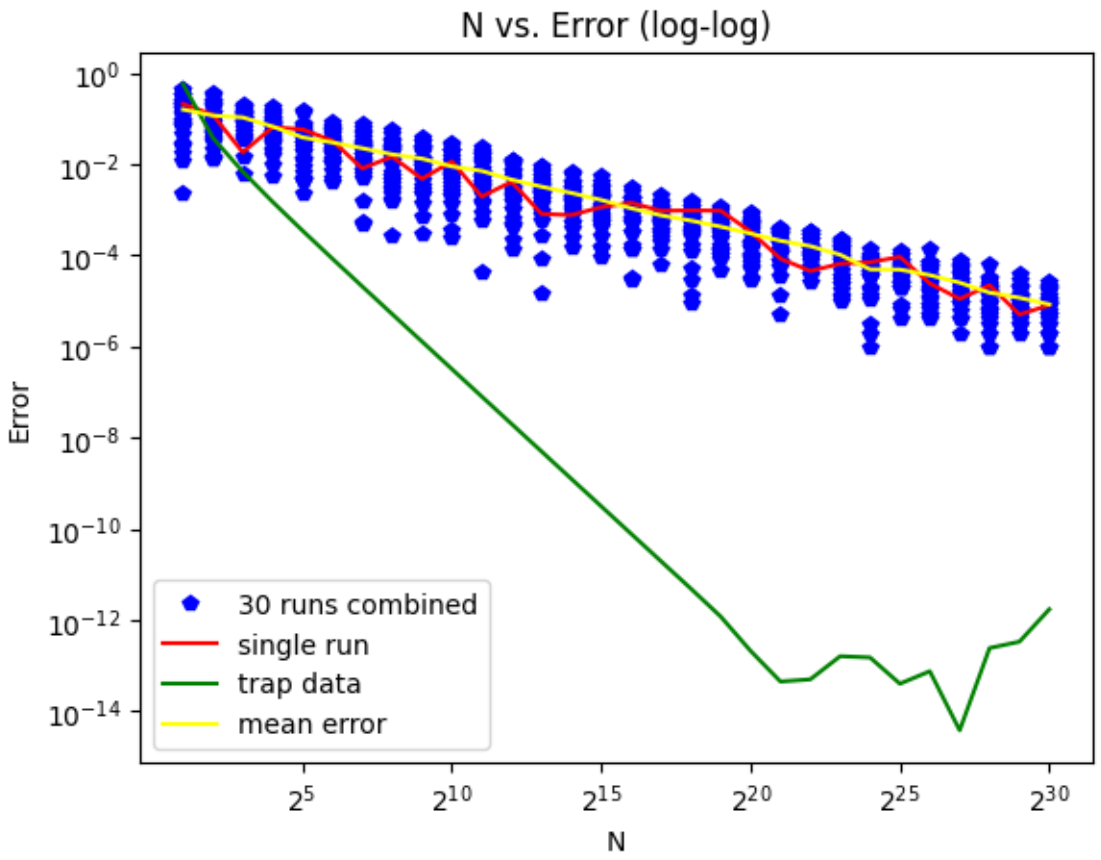

[4]:
```
import numpy as np
from matplotlib import pyplot as plt

data_trap = np.loadtxt("error_trap.dat")

powers_of_2 = [5, 10, 15, 20, 25, 30]
```

```
x_ticks = [2**i for i in powers_of_2]
x_labels = [r'$2^{%d}$' % i for i in powers_of_2]

plt.loglog(data_trap[:,0], data_trap[:,1],label='trap data')
plt.xticks(x_ticks, x_labels)
plt.legend(loc=3)
plt.xlabel("N")
plt.ylabel("Error")
plt.title('N vs. Error (log-log)')
plt.savefig("trap")
```

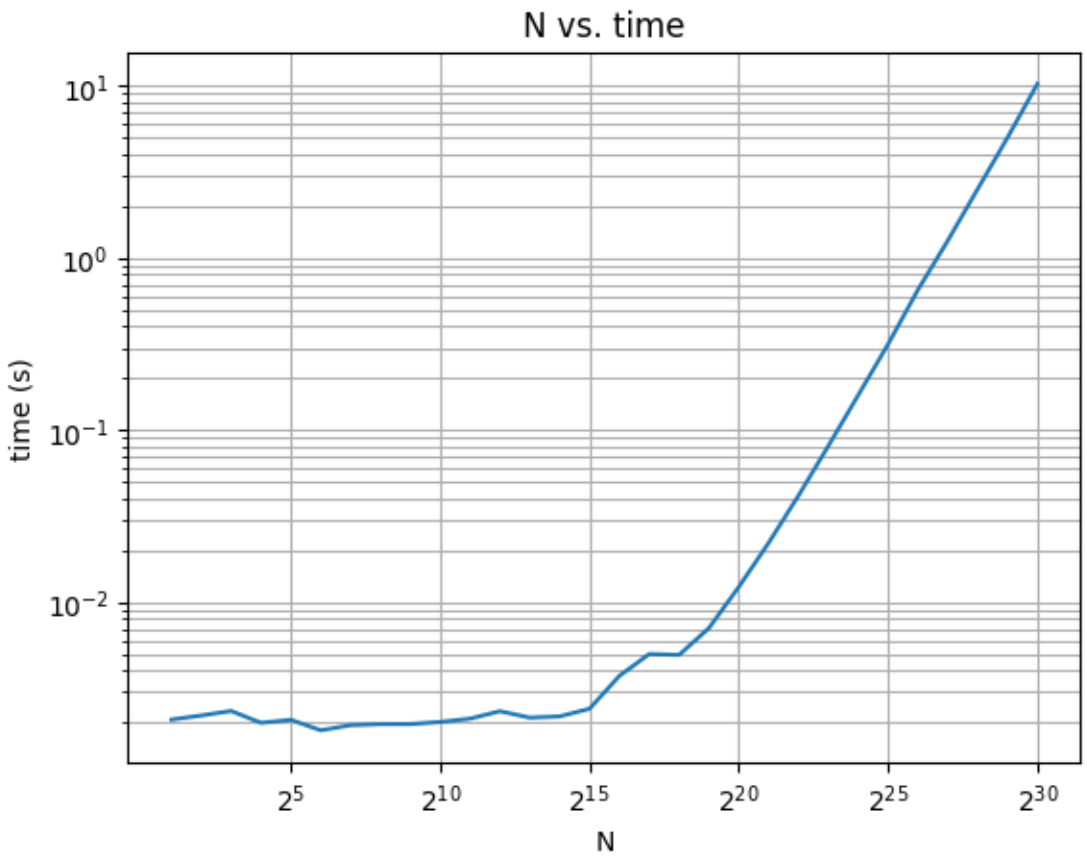



[5]:
```
import numpy as np
import matplotlib.pyplot as plt

#loading data
data_1 = np.loadtxt("error.dat")
data_2 = np.loadtxt("error_30.dat")
data_trap = np.loadtxt("error_trap.dat")
```

```
#setting parameters for costum x-ticks
powers_of_2 = [5, 10, 15, 20, 25, 30]
x_ticks = [2**i for i in powers_of_2]
x_labels = [r'$2^{%d}$' % i for i in powers_of_2]

#calcualting mean error
N_i = data_2[:, 0]
error = data_2[:, 1]
unique_N = []
mean_error = []

for N in np.unique(N_i):
    error_N = error[N_i == N]
    mean_error_N = np.mean(error_N) #calculates mean error for each value of 'N'
    unique_N.append(N)
    mean_error.append(mean_error_N)

#storing the N values and mean errors in a new array
unique_N = np.array(unique_N)
mean_error = np.array(mean_error)

#generate plots
plt.loglog(data_2[:, 0], data_2[:, 1], 'p', label='30 runs combined',␣
 ↪color='blue')
plt.loglog(data_1[:, 0], data_1[:, 1], label='single run', color='red')
plt.loglog(data_trap[:, 0], data_trap[:, 1], label='trap data', color='green')
plt.loglog(unique_N, mean_error, label='mean error', color='yellow')

#set the coustom x-ticks
plt.xticks(x_ticks, x_labels)

#add legends, labels, and title
plt.legend(loc=3)
plt.xlabel("N")
plt.ylabel("Error")
plt.title('N vs. Error (log-log)')

# Save the combined plot
plt.savefig("combined_plots")
```




[6]:
```
import numpy as np
from matplotlib import pyplot as plt

data_1 = np.loadtxt("timing.dat")

powers_of_2 = [5, 10, 15, 20, 25, 30]
x_ticks = [2**i for i in powers_of_2]
x_labels = [r'$2^{%d}$' % i for i in powers_of_2]

plt.loglog(data_1[:,0], data_1[:,1])
plt.xticks(x_ticks, x_labels)
plt.xlabel("N")
plt.ylabel("time (s)")
plt.grid(True, which='both', axis='both')
plt.title('N vs. time')
plt.savefig("g")
```


N vs. time
time (s)
$10^{1}$
$10^{0}$
$10^{-1}$
$10^{-2}$
$2^{5}$
$2^{10}$
$2^{15}$
$2^{20}$
$2^{25}$
$2^{30}$
N

[ ]: