```
import numpy as np
import matplotlib.pyplot as plt
import random

def generate_spike(r, n, delta, bins):


        r_dt = r * delta
        all_spike_sims = []
        for i in range(n):
                rand_n = np.random.rand(bins,)
                spike_arr = rand_n < r_dt
                spike_arr = (spike_arr.astype(int)) * (i + 1)
                print(rand_n)
                print(r_dt)
                print(spike_arr)
                all_spike_sims.append(spike_arr)

        return np.asarray(all_spike_sims)

def get_spike_only(array, step):

        t = step
        time_spike = []
        spike_arr = []
        for i in array:
                if i != 0:
                        spike_arr.append(i)
                        time_spike.append(t)
                t += step

        return time_spike, spike_arr

rate = 20
del_t = 1 / 1000
T = 1
n_bins = int (T / del_t)
n_trials = 5
all_spikes_arr = generate_spike(rate, n_trials, del_t, n_bins)
num_rows = all_spikes_arr.shape[0]
fig = plt.figure()
axes = fig.add_subplot(1,1,1)
#axes.set_xticks(np.linspace(0,T , n_bins + 1))
#axes.set_xticks(np.linspace(0,n_trials , n_trials + 1))
axes.set_xlim(0, T + del_t)
axes.set_ylim(0, n_trials + 1)
axes.set_xlabel("time")
axes.set_ylabel("Trial Number")
axes.set_title("Poisson spike train with $\lambda$ = " + str(rate) + "
Hz")
```

```
for i in range(num_rows):
        spike_only = get_spike_only(all_spikes_arr[i], del_t)
        axes.scatter(spike_only[0], spike_only[1], marker = "|", color
= 'black')
plt.show()
```