```python
#%%
# Importing libraries
import matplotlib.pyplot as plt
import numpy as np
import matplotlib as mpl

mpl.rcParams["figure.dpi"] = 100
#%%
# all diameters are in nm
S_g = 1.7 # standard deviation of the lognormal distribution
N = 1000 # number of particles in the distribution (cm^-3)
D_pg = 200 # geometric mean diameter (nm)

Dp = np.geomspace(1e-8, 1e-6, 1000)*1e9 # convert to nm
dN_dDp = N/(np.sqrt(2*np.pi)*np.log(S_g)*Dp)*np.exp(-(np.log(Dp/D_pg))**2/(2*(np.log(S_g))**2))

dN_dlogDp = dN_dDp*Dp*np.log(10)
dN_dlnDp = dN_dDp*Dp
#%%
# Creating subplots
fig, axs = plt.subplots(3, 1, figsize=(10, 8))

# First plot
axs[0].plot(Dp, dN_dDp)
axs[0].set_xscale('log')
axs[0].set_xlabel('Dp (nm)')
axs[0].set_ylabel('dN/dDp (cm^-3)')
axs[0].legend(['dN/dDp'])

# Second plot
axs[1].plot(Dp, dN_dlnDp)
axs[1].set_xscale('log')
axs[1].set_xlabel('Dp (nm)')
axs[1].set_ylabel('dN/dlnDp')
axs[1].legend(['dN/dlnDp'])

# Third plot
axs[2].plot(Dp, dN_dlogDp)
axs[2].set_xscale('log')
axs[2].set_xlabel('Dp (nm)')
axs[2].set_ylabel('dN/dlogDp')
axs[2].legend(['dN/dlogDp'])

plt.savefig('1-2.png', bbox_inches='tight')
plt.show()
#%%
bin_number = 40
bins_lower = np.geomspace(1e-9, 10.3e-6, bin_number + 1)  #
bins_upper = bins_lower[1:]
bins_lower = bins_lower[:-1]
bins_mid = np.sqrt(bins_lower * bins_upper)  # geometric mean
for i in range(bin_number):
```

```
    print(i, bins_lower[i], bins_upper[i], bins_mid[i])
#%%
# all diameters are in nm
S_g = 1.7 # standard deviation of the lognormal distribution
N = 1000 # number of particles in the distribution (cm^-3)
D_pg = 200e-9 # geometric mean diameter (nm)

Dp = bins_mid
dN_dDp = N/(np.sqrt(2*np.pi)*np.log(S_g)*Dp)*np.exp(-(np.log(Dp/D_pg))**2/(2*(np.log(S_g))**2))
N = dN_dDp * (bins_upper - bins_lower)
#%%
# Creating histogram of N
plt.bar(bins_mid*1e9, N, width=(bins_upper - bins_lower)*1e9, align='center', edgecolor='black')
plt.xscale('log')
# plt.yscale('log')
plt.xlabel('Dp (nm)')
plt.ylabel('N (cm^-3)')
plt.suptitle('Histogram of N in each bin')
#save the plot
plt.savefig('1-3.png', bbox_inches='tight')
plt.show()

#
#%%
# Creating a gamma distribution of droplet sizes
# Total number of drops in gamma distribution is N_d = 100 cm^-3
# The mean diameter is D_pg = 20 um = 20000 nm
# Dp_mean_2 = 20000 # nm
# gamma = 1
# Beta = 2

N_2 = 100 # cm^-3 (total number of droplets)
rp_mean_2 = 20e-6 # m (mean diameter of droplets)
Dp_2 = np.geomspace(1e-6, 1e-3, 1000) # convert to nm
B_2 =  3 / rp_mean_2 # r_mean = 3/B_2
A_2 = N_2 * B_2**3 / 2 # N_2 = 2 * A_2 / B_2**3
dN2_dr = A_2 * ((Dp_2 / 2 )**2) * np.exp( -B_2 * (Dp_2 / 2) ) # dN2_dr = A_2 * r^2 * exp(-B_2 * r)
dN2_dDp = dN2_dr / 2 # dN2_dDp = dN2_dr / 2
dN2_dlnDp = dN2_dDp * Dp_2
dN2_dlogDp = dN2_dDp * Dp_2 * np.log(10)
#%%
# Creating subplots
fig, axs = plt.subplots(3, 1, figsize=(10, 8))

# First plot
axs[0].plot(Dp_2*1e9, dN2_dDp) # convert to nm
axs[0].set_xscale('log')
axs[0].set_xlabel('Dp (nm)')
axs[0].set_ylabel('dN/dDp (cm^-3)')
axs[0].legend(['dN/dDp'])
```

```

# Second plot
axs[1].plot(Dp_2*1e9, dN2_dlnDp) # convert to nm
axs[1].set_xscale('log')
axs[1].set_xlabel('Dp (nm)')
axs[1].set_ylabel('dN/dlnDp')
axs[1].legend(['dN/dlnDp'])

# Third plot
axs[2].plot(Dp_2*1e9, dN2_dlogDp) # convert to nm
axs[2].set_xscale('log')
axs[2].set_xlabel('Dp (nm)')
axs[2].set_ylabel('dN/dlogDp')
axs[2].legend(['dN/dlogDp'])
plt.savefig('1-2.png', bbox_inches='tight')
plt.show()
```