```python
# A rmajet s to propel an aircraft at mach 1 to 6 at hugh altitude and P_amb = 8.5 kPa and
T_amb = 220K
# The turbine inlet Temperature is T_max = 2540K
# Construct plots of d(I)/dnb, d(I)/drn, d(TSFC)/dnb, d(TSFC)/drn as a function of M=1 to M=6.
# The derivatives can be approximated using small but discrete changes, such as
#                d(I)/dnb = Delta_I/Delta_nb
# Use baseline nb and rn values of one, and use delta values of 0.01.  Make plots of
# the normalized sensitivity coefficient (e.g. derivitive) as a function of flight mach number.

import math
import matplotlib.pyplot as plt
import numpy as np

# Conditions given in the problem
P_amb = 8.5
T_amb = 220
Gamma = 1.4
R = 287
Cp = 1.005

Mlist = np.array([])
Ilist = np.array([])
TSFClist = np.array([])
rnlist = np.array([])
nblist = np.array([])

for M in np.arange(1,6,0.01):
    To_max = 2540
    To4 = To_max

    To2 = T_amb*(1+((Gamma-1)/2)*M**2)
    T6 = To_max/(1+((Gamma-1)/2)*M**2)

    u_exit = M*math.sqrt(Gamma*R*T6)
    u_in = M*math.sqrt(Gamma*R*T_amb)

    I = u_exit-u_in
    TSFC = 1/I

    Ilist=np.append(Ilist,I)
    TSFClist=np.append(TSFClist,TSFC)
    rnlist=np.append(rnlist,-0.01)
    nblist=np.append(nblist,-0.01)
    Mlist=np.append(Mlist,M)

# Now to produce gradients and desired maps
plot1 = (1/Ilist)*(np.gradient(Ilist)/nblist)
plot2 = (1/Ilist)*(np.gradient(Ilist)/rnlist)
plot3 = (1/TSFClist)*(np.gradient(TSFClist)/nblist)
plot4 = (1/TSFClist)*(np.gradient(TSFClist)/rnlist)
plot5 = (np.gradient(Ilist)/nblist)
plot6 = (np.gradient(Ilist)/rnlist)
plot7 = (np.gradient(TSFClist)/nblist)
```

```
plot8 = (np.gradient(TSFClist)/rnlist)

# Now to plot everything!
plt.figure(1)
plt.plot(Mlist, plot1)
plt.xlabel('Mach Number, M')
plt.ylabel('1/I * d(I)/d(nb)')
plt.title('1/I * d(I)/d(nb) vs Mach Number')

plt.figure(2)
plt.plot(Mlist, plot2)
plt.xlabel('Mach Number, M')
plt.ylabel('1/I * d(I)/d(rn)')
plt.title('1/I * d(I)/d(rn) vs Mach Number')

plt.figure(3)
plt.plot(Mlist, plot3)
plt.xlabel('Mach Number, M')
plt.ylabel('1/TSFC * d(TSFC)/d(nb)')
plt.title('1/TSFC * d(TSFC)/d(nb) vs Mach Number')

plt.figure(4)
plt.plot(Mlist, plot4)
plt.xlabel('Mach Number, M')
plt.ylabel('1/TSFC * d(TSFC)/d(rn)')
plt.title('1/TSFC * d(TSFC)/d(rn) vs Mach Number')

plt.figure(5)
plt.plot(Mlist, plot5)
plt.xlabel('Mach Number, M')
plt.ylabel('d(I)/d(nb)')
plt.title('d(I)/d(nb) vs Mach Number')

plt.figure(6)
plt.plot(Mlist, plot6)
plt.xlabel('Mach Number, M')
plt.ylabel('d(I)/d(rn)')
plt.title('d(I)/d(rn) vs Mach Number')

plt.figure(7)
plt.plot(Mlist, plot7)
plt.xlabel('Mach Number, M')
plt.ylabel('d(TSFC)/d(nb)')
plt.title('d(TSFC)/d(nb) vs Mach Number')

plt.figure(8)
plt.plot(Mlist, plot8)
plt.xlabel('Mach Number, M')
plt.ylabel('d(TSFC)/d(rn)')
plt.title('d(TSFC)/d(rn) vs Mach Number')

plt.show()
```