

# Python code for Labo6

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

#-----
# Group 1 android vs iphone data
android_data_1ms = pd.read_csv('Formatted_raw_data\\android_1ms.txt', sep="\t", header=None, names = ["Wavelength",
android_data = pd.read_csv('Formatted_raw_data\\android.txt', sep="\t", header=None, names = ["Wavelength", "Intensity"]
iphone_data_1ms = pd.read_csv('Formatted_raw_data\\iphone_1ms.txt', sep="\t", header=None, names = ["Wavelength", "Intensity"]
iphone_data = pd.read_csv('Formatted_raw_data\\iphone.txt', sep="\t", header=None, names = ["Wavelength", "Intensity"]

plt.subplots(layout="constrained")
plt.suptitle("Iphone VS Android Comparison")

ax1 = plt.subplot(121)
plt.plot(android_data_1ms["Wavelength"], android_data_1ms["Intensity"], c='Blue',linewidth = 1, label = "android")
plt.plot(iphone_data_1ms["Wavelength"], iphone_data_1ms["Intensity"], c='Green',linewidth = 1, label = "iphone")
plt.xlabel("wavelength in nm")
plt.ylabel("intensity in W/m^2")
plt.title("1ms exposure time")
plt.grid()
plt.legend()

ax2 = plt.subplot(122, sharey=ax1, sharex=ax1)
plt.plot(android_data["Wavelength"], android_data["Intensity"], c='Red',linewidth = 1, label = "android")
plt.plot(iphone_data["Wavelength"], iphone_data["Intensity"], c='Black',linewidth = 1, label = "iphone")
plt.xlabel("wavelength in nm")
plt.ylabel("intensity in W/m^2")
plt.title("0.1s exposure time")
plt.legend()
plt.grid()
plt.show()
#-----

#-----
# Group 2 base noise fiber VS no fiber comparison
base_noise_fiber = pd.read_csv('Formatted_raw_data\\base_noise_no_fiber.txt', sep="\t", header=None, names = ["Wavelength",
base_noise_no_fiber = pd.read_csv('Formatted_raw_data\\base_noise_with_fiber.txt', sep="\t", header=None, names = ["Wavelength",

plt.plot(base_noise_fiber["Wavelength"], base_noise_fiber["Intensity"], c='Blue',linewidth = 0.5, label = "base noise")
plt.plot(base_noise_no_fiber["Wavelength"], base_noise_no_fiber["Intensity"], c='Green',linewidth = 0.5, label = "base noise")
plt.xlabel("wavelength in nm")
plt.ylabel("intensity in W/m^2")
plt.title("0.1s exposure time for base noise measurement")
plt.legend()
plt.grid()
plt.show()

#-----

#-----
# Group 3 base light with fiber 0.1s, 5ms, 100ms and without fiber
base_light_no_fiber = pd.read_csv('Formatted_raw_data\\base_light_no_fiber.txt', sep="\t", header=None, names = ["Wavelength",
base_light_with_fiber_1ms = pd.read_csv('Formatted_raw_data\\base_light_with_fiber_1ms.txt', sep="\t", header=None,

base_light_with_fiber_5ms = pd.read_csv('Formatted_raw_data\\base_light_with_fiber_5ms.txt', sep="\t", header=None,
```

```

base_light_with_fiber_100ms = pd.read_csv('Formatted_raw_data\\base_light_with_fiber_100ms.txt', sep="\t", header=None)

plt.plot(base_light_no_fiber["Wavelength"], base_light_no_fiber["Intensity"], c='Black',linewidth = 0.5, label = "base light")
plt.plot(base_light_with_fiber_1ms["Wavelength"], base_light_with_fiber_1ms["Intensity"], c='Green',linewidth = 2, label = "1ms")
plt.plot(base_light_with_fiber_5ms["Wavelength"], base_light_with_fiber_5ms["Intensity"], c='Red',linewidth = 1, label = "5ms")
plt.plot(base_light_with_fiber_100ms["Wavelength"], base_light_with_fiber_100ms["Intensity"], c='Blue',linewidth = 0.5, label = "100ms")

plt.xlabel("wavelength in nm")
plt.ylabel("intensity in W/m^2")
plt.title("base light measurement with different exposure times")
plt.legend()
plt.grid()

plt.show()
#-----

#-----
# Group 4 LED distance comparison in intensity
LED_red_10_5cm = pd.read_csv('Formatted_raw_data\\LED_red_10.5cm.txt', sep="\t", header=None, names = ["Wavelength", "Intensity"])
LED_red_10_cm = pd.read_csv('Formatted_raw_data\\LED_red_10cm.txt', sep="\t", header=None, names = ["Wavelength", "Intensity"])

plt.plot(LED_red_10_5cm["Wavelength"], LED_red_10_5cm["Intensity"], c='Black',linewidth = 2, label = "10.5cm away - saturation")
plt.plot(LED_red_10_cm["Wavelength"], LED_red_10_cm["Intensity"], c='Green',linewidth = 2, label = "10cm away - saturation")

plt.xlabel("wavelength in nm")
plt.ylabel("intensity in W/m^2")
plt.title("LED intensity measurements from two distances")
plt.legend()
plt.grid()

plt.show()
#-----

#-----
# Group 5 optical filter measurements
optical_filter_min = pd.read_csv('Formatted_raw_data\\optical_filter_min.txt', sep="\t", header=None, names = ["Wavelength", "Intensity"])
optical_filter_max = pd.read_csv('Formatted_raw_data\\optical_filter_max.txt', sep="\t", header=None, names = ["Wavelength", "Intensity"])

plt.plot(optical_filter_min["Wavelength"], optical_filter_min["Intensity"], c='Blue',linewidth = 2, label = "optical filter minimum")
plt.plot(optical_filter_max["Wavelength"], optical_filter_max["Intensity"], c='Green',linewidth = 2, label = "optical filter maximum")

plt.xlabel("wavelength in nm")
plt.ylabel("intensity in W/m^2")
plt.title("Optical filter measurements from maximum to minimum")
plt.legend()
plt.grid()

plt.show()
#-----

#-----
# Group 6 optical glasses 800ms exposure time
optical_glasses = pd.read_csv('Formatted_raw_data\\optical_glasses_800ms.txt', sep="\t", header=None, names = ["Wavelength", "Intensity"])

plt.plot(optical_glasses["Wavelength"], optical_glasses["Intensity"], c='Red',linewidth = 2, label = "optical glasses 800ms")

plt.xlabel("wavelength in nm")
plt.ylabel("intensity in W/m^2")
plt.title("Optical Glasses")
plt.legend()
plt.grid()

plt.show()
#-----

# Group 6 comparing the optical gasses and regular LED 10cm away

```

```
optical_glasses = pd.read_csv('Formatted_raw_data\\optical_glasses_800ms.txt', sep="\t", header=None, names = ["Wave  
LED_red_10_5cm = pd.read_csv('Formatted_raw_data\\LED_red_10.5cm.txt', sep="\t", header=None, names = ["Wavelength",  
  
plt.plot(optical_glasses["Wavelength"], optical_glasses["Intensity"], c='Red',linewidth = 2, label = "optical glasse  
plt.plot(LED_red_10_5cm["Wavelength"], LED_red_10_5cm["Intensity"], c='Black',linewidth = 2, label = "10.5cm away -  
  
plt.xlabel("wavelength in nm")  
plt.ylabel("intensity in W/m^2")  
plt.title("Optical Glasses VS No glasses")  
plt.legend()  
plt.grid()  
  
plt.show()
```

---