```
####################################
# How to calculate daylight factor?
####################################

#https://lbnl-eta.github.io/frads/how-to/guide_rad7/

# 0. Import the required classes and functions

import datetime
import frads as fr
import pyradiance as pr
import numpy as np

import os

#############
# build room
#############

# 1. Build a Radiance model

os.system('gen room 12 14 4 3 \
    -w 0.4 1 2.5 1.8 \
    -w 3.3 1 2.5 1.8 \
    -w 6.2 1 2.5 1.8 \
    -w 9.1 1 2.5 1.8 \
    -t 0.1 -n aroom')#

# all moved to Object directory

# objview Objects/*aroom.mat Objects/*aroom.rad
# p 180


fpaths = ["Objects/materials_aroom.mat",
          "Objects/ceiling_aroom.rad",
          "Objects/wall_aroom.rad",
          "Objects/floor_aroom.rad",
          "Objects/window_00_aroom.rad",
          "Objects/window_01_aroom.rad",
          "Objects/window_02_aroom.rad",
          "Objects/window_03_aroom.rad",
]

room_octree = "aroom.oct"
with open(room_octree, 'wb') as f:
    f.write(pr.oconv(*fpaths))

##############
# light source
##############

# 2. Generate a CIE overcast sky
```

```
dif_hor_illum = 10000
dif_hor_ird = dif_hor_illum / 179

sky_func = pr.gensky(
    dt=datetime.datetime(2024, 12, 21, 12, 0),
    cloudy=True,
    horizontal_brightness=dif_hor_ird, #
    )

print(sky_func)

sky_glow = "skyfunc glow sky_glow 0 0 4 1 1 1 0".encode()

sky = "sky_glow source sky 0 0 4 0 0 1 180".encode()

ground = "sky_glow source ground 0 0 4 0 0 -1 180".encode()

sky_scene = sky_func + b'\n' + sky_glow + b'\n' + sky + b'\n' + ground

# add sky scene to octree
room_sky_octree = f"aroom_37_122_1221_1200.oct"
with open(room_sky_octree, "wb") as f:
    f.write(pr.oconv(stdin=sky_scene, octree=room_octree))


# 3. Use raytracing to get the interior horizontal illuminance

###############
# Sensors
###############

# generate grid sensors based on the floor polygon
floor_primitives = fr.unpack_primitives("Objects/floor_aroom.rad")
floor_polygon = fr.parse_polygon(floor_primitives[0])
grid = fr.gen_grid(floor_polygon, 1, 0.75)

###############
# tracing rays
###############

# rtrace to get irradiance
option = ["-I+", "-ab", "5", "-ad", "512", "-aa", "0", "-lw", "0.001"]
rays = "\n".join([" ".join(map(str, row)) for row in grid])
ird_results = pr.rtrace(rays.encode(), room_sky_octree, params=option,
header=False)

print(ird_results[:50])

###############
# Postprocessing
###############

# reformat the irradiance result
```

```
rows = ird_results.decode().strip().split("\n")
data = [
    [float(element) for element in row.split("\t") if element != ""]
    for row in rows
]
ird_array = np.array(data)

# convert irradiance to illuminance
illum = ird_array[:, 0] * 47.4 + ird_array[:, 1] * 119.9 + ird_array[:, 2] * 11.6
    #

###############
# daylight factor
###############

# 4. Compute the daylight factor
# daylight factor = interior illuminance / exterior illuminance * 100

df = illum / dif_hor_illum * 100

print('df min max', min(df), max(df))

###############
# plotting
###############

# import visualization packages

import matplotlib.pyplot as plt

fig, ax = plt.subplots()
x = [data[0] for data in grid]
y = [data[1] for data in grid]
_plot = ax.scatter(
    x,
    y,
    c=df,
    cmap="plasma",
    s=15,
    vmin=0,
    vmax=max(df),
    rasterized=True,
)
ax.set(
    xlabel="x position [m]",
    ylabel="y position [m]",
)

fig.colorbar(_plot, ax=ax, label="Daylight Factor [%]")
fig.tight_layout()
fig.savefig('df.png')
```