Setup NREL Vesion of 118 bus system

April 30, 2025

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")
```

1 Setup of the NREL version of the 118bus system

Setup of the system: - For the static analysis: - Define system topology - Define synchronous generation and converter-interfaced generation installed capacity - Define peak and minimum load demand - Define lines and transformers ratings - For the dynamic analysis, in addition to the previous ones: - Control parameters and sizing of generators and converters' internal components

Data for the static analysis is provided by NREL, in [1]. Data for the dynamic analysis is based on adapting conventional values of control parameters and component sizing to the system, by conducting a participation factor analysis to achieve a reasonable level of system stability.

1.1 References

[1] Pena, Ivonne, Carlo Brancucci Martinez-Anido, and Bri-Mathias Hodge. "An extended IEEE 118-bus test system with high renewable penetration." IEEE Transactions on Power Systems 33.1 (2017): 281-289.

1.2 NREL-118bus System : Information for Static Analysis

Description of the system: - Number of Regions = 3, (R1, R2, R3) - Number of buses = 118 - Number of generation units = 53 (54) - Number of generation units with converter-interfaced generation (CIG) = 18

1.2.1 Buses

List of system buses, with information related to: - Region: region of the system they belong to - Load Participation Factor: percentage of the region total demand absorbed by the load in the bus

[2]:
```
Buses=pd.read_csv('Buses.csv')
Buses
```

[2]:

	Bus Name	Region	Load Participation Factor
0	bus026	R1	0.000000

```
1      bus010      R1                    0.000000
2      bus025      R1                    0.000000
3      bus012      R1                    0.043466
4      bus015      R1                    0.083238
..        ...     ...                         ...
113    bus093      R3                    0.016137
114    bus084      R3                    0.014792
115    bus109      R3                    0.010758
116    bus102      R3                    0.006724
117    bus108      R3                    0.002690

[118 rows x 3 columns]
```

[3]:
```
for i in range(len(Buses)):
    Buses.loc[i,'Num']=int(Buses.loc[i,'Bus Name'][3:])

Buses=Buses.sort_values(by='Bus Name').reset_index(drop=True)

#pd.DataFrame.to_excel(Buses,'G:/Il mio Drive/Francesca 118 v2/Operation/
 ↪tool_ree/tool_ree/Buses.xlsx')

Buses
```

[3]:
```
    Bus Name Region  Load Participation Factor    Num
0      bus001     R1                   0.047169    1.0
1      bus002     R1                   0.018496    2.0
2      bus003     R1                   0.036069    3.0
3      bus004     R1                   0.027749    4.0
4      bus005     R1                   0.000000    5.0
..        ...    ...                        ...    ...
113    bus114     R1                   0.007397  114.0
114    bus115     R1                   0.020343  115.0
115    bus116     R2                   0.000000  116.0
116    bus117     R1                   0.018496  117.0
117    bus118     R2                   0.017927  118.0

[118 rows x 4 columns]
```

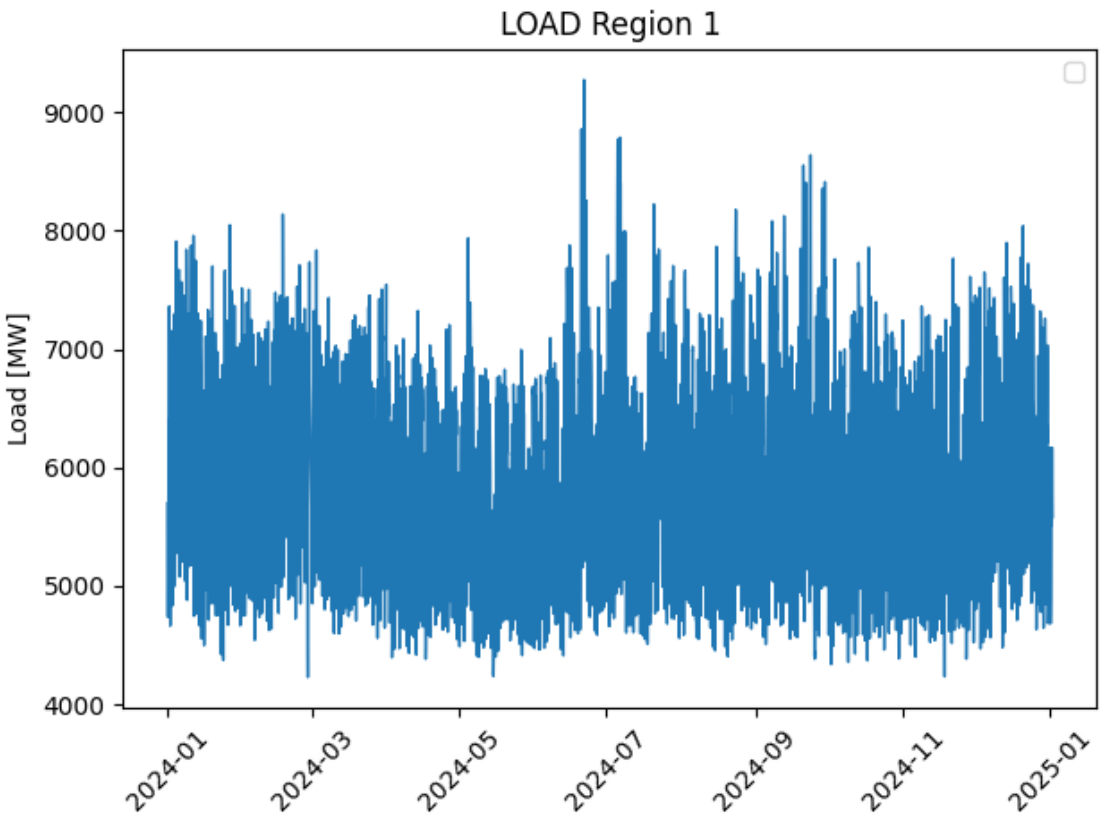
1.2.2 Loads

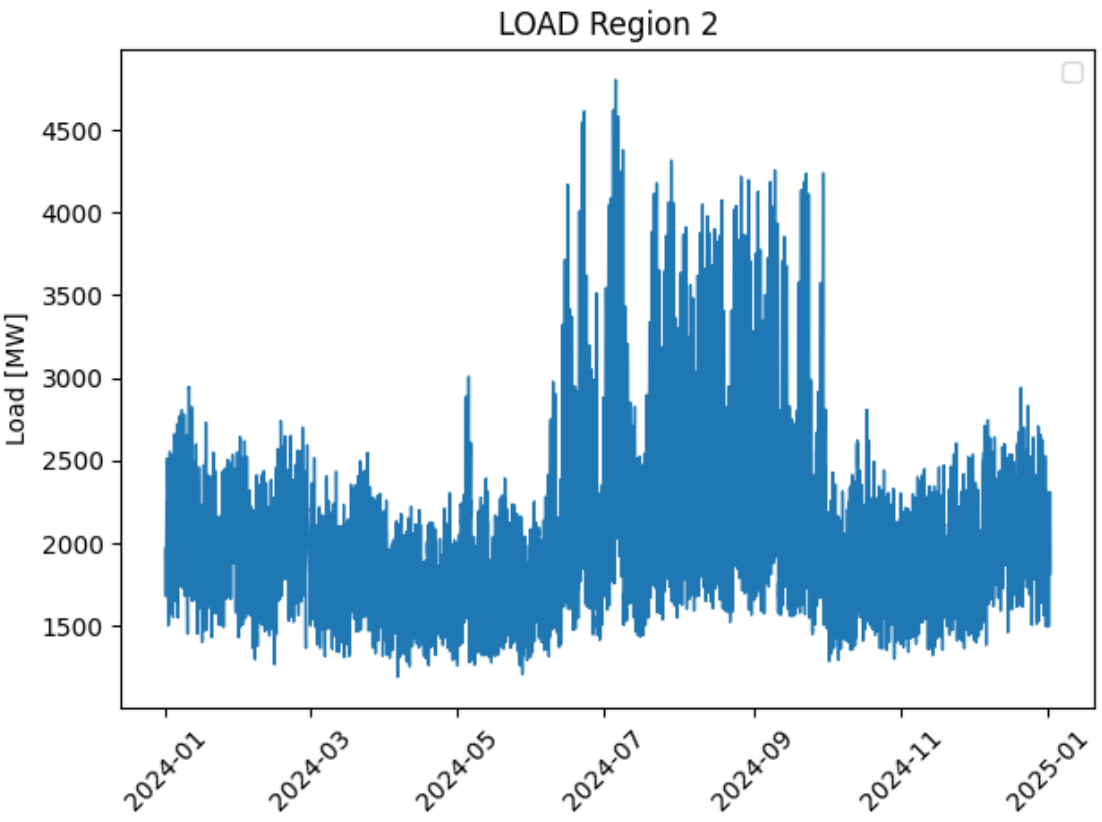
Hourly load consumption for one year, aggregated for system's region

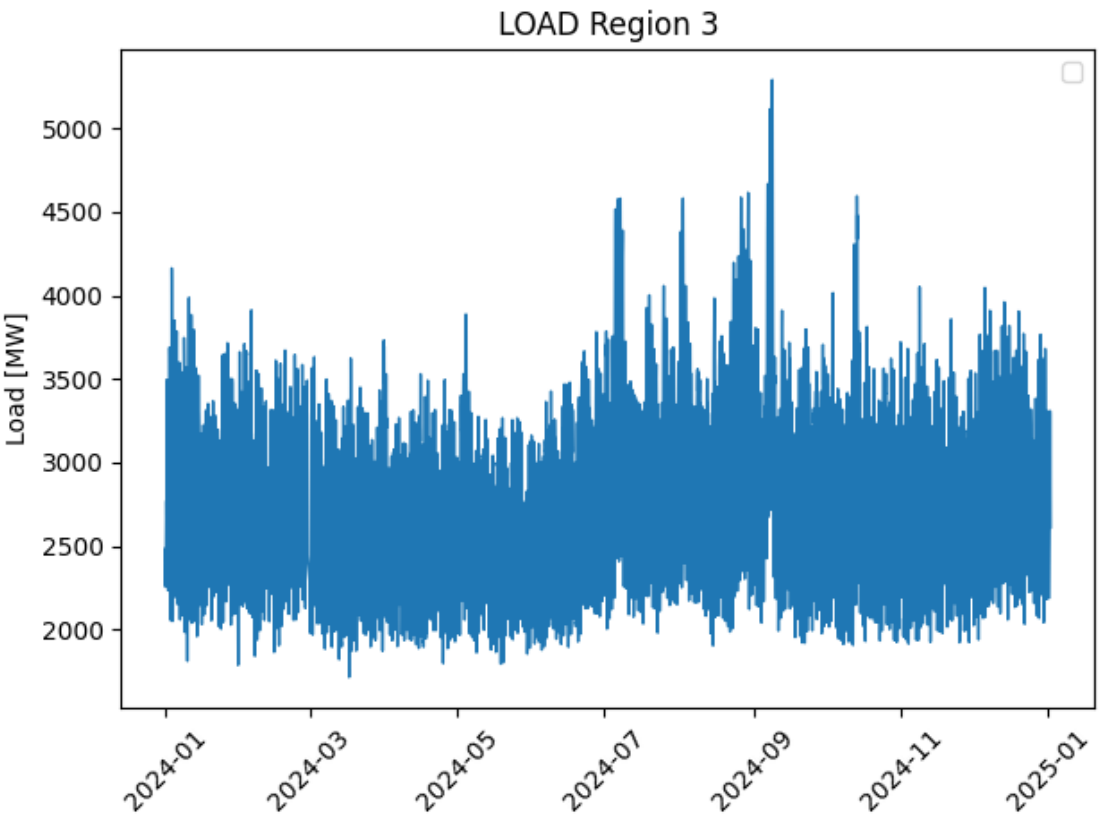
[4]:
```
Loads=pd.DataFrame()

for zone in [1,2,3]:

    Loads_R=pd.read_csv('./input-files/Input files/RT/Load/LoadR'+str(zone)+'RT.
 ↪csv')
```

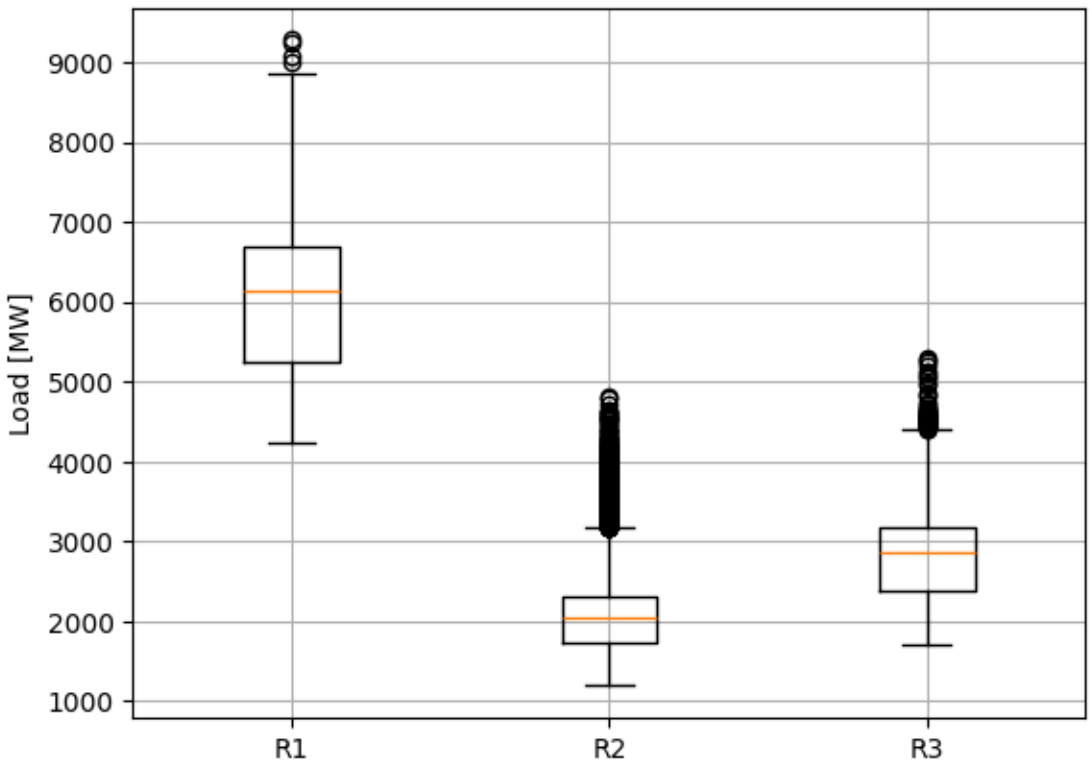
```
    Loads_R["DATETIME"] = pd.to_datetime(Loads_R["DATETIME"])

    Loads_R=Loads_R.set_index(["DATETIME"]).sort_index()

    Loads['R'+str(zone)]=Loads_R

    fig=plt.figure()
    ax=fig.add_subplot()
    ax.plot(Loads_R.index,Loads_R['value'], linewidth=1)
    plt.legend()
    ax.set_title('LOAD Region '+str(zone))
    plt.xticks(rotation = 45) # Rotates X-Axis Ticks by 45-degrees
    ax.set_ylabel('Load [MW]')
    fig.tight_layout()

fig=plt.figure()
ax=fig.add_subplot()
ax.boxplot(Loads)
ax.set_ylabel('Load [MW]')
plt.xticks([1, 2,3], list(Loads.columns))
ax.grid()
```

```
No artists with labels found to put in legend.  Note that artists whose label
start with an underscore are ignored when legend() is called with no argument.
No artists with labels found to put in legend.  Note that artists whose label
start with an underscore are ignored when legend() is called with no argument.
No artists with labels found to put in legend.  Note that artists whose label
start with an underscore are ignored when legend() is called with no argument.
```


LOAD Region 1
Load [MW]
9000
8000
7000
6000
5000
4000
2024-01
2024-03
2024-05
2024-07
2024-09
2024-11
2025-01


LOAD Region 2
Load [MW]
4500
4000
3500
3000
2500
2000
1500
2024-01
2024-03
2024-05
2024-07
2024-09
2024-11
2025-01


LOAD Region 3
Load [MW]
5000
4500
4000
3500
3000
2500
2000
2024-01
2024-03
2024-05
2024-07
2024-09
2024-11
2025-01



List of buses with load

[5]:
```
Buses=Buses.rename(columns={"Load Participation Factor":␣
↪"Load_Participation_Factor"})
loads_nrel=Buses.query('Load_Participation_Factor !=0')

loads_nrel
```

[5]:

	Bus Name	Region	Load_Participation_Factor	Num
0	bus001	R1	0.047169	1.0
1	bus002	R1	0.018496	2.0
2	bus003	R1	0.036069	3.0
3	bus004	R1	0.027749	4.0
5	bus006	R1	0.048092	6.0
..	...	...	...	...
111	bus112	R3	0.033619	112.0
113	bus114	R1	0.007397	114.0
114	bus115	R1	0.020343	115.0
116	bus117	R1	0.018496	117.0
117	bus118	R2	0.017927	118.0

[91 rows x 4 columns]

1.2.3 Generators

List of generators indicating: - Node of connection: bus to which they are connected –> Num: number oft the bus - Max Capacity (MW): corresponding to the installed capacity (nominal power, Sn) of the generator - Type: type of generator

[6]:
```
Generators=pd.read_excel('Generators_red.xlsx') #Reduced because useless columns␣
↪have been discarded

for i in range(len(Generators)):
    Generators.loc[i,'Num']=int(Generators.loc[i,'Node of connection'][4:])
    Generators.loc[i,'Type']=Generators.loc[i,'Generator Name'][:-3]
    bus_num=int(Generators.loc[i,'Node of connection'][4:])
    Generators.loc[i,'Generator Name'][:-3]
    Generators.loc[i,'Region']=list(Buses.query('Num == @bus_num')['Region'])[0]

Generators
```

[6]:

	Generator Name	Node of connection	Category	Units	Max Capacity (MW)	Commit	Min Stable Level (MW)	Num	Type	Region
0	Biomass 01	node012	1. Committed DA	1	3.0	0	0.90	12.0	Biomass	R1
1	Biomass 02	node012	1. Committed DA	1	3.0	0	0.90	12.0	Biomass	R1
2	Biomass 03	node103	1. Committed DA	1	1.2	0	0.36	103.0	Biomass	R3
3	Biomass 04	node103	1. Committed DA	1	1.2	0	0.36	103.0	Biomass	R3
4	Biomass 05	node012	1. Committed DA	1	1.3	0	0.26	12.0	Biomass	R1
..	...	...	...	...	...	...	...	...	...	...
322	Wind 13	node031	Wind	1	58.7	1	0.00	31.0	Wind	R1
323	Wind 14	node100	Wind	1	400.0	1	0.00	100.0	Wind	R3
324	Wind 15	node082	Wind	1	50.0	1	0.00	82.0	Wind	R3
325	Wind 16	node082	Wind	1	149.5	1	0.00	82.0	Wind	R3
326	Wind 17	node082	Wind	1	149.5	1	0.00	82.0	Wind	R3

[327 rows x 10 columns]

[7]:
```
#%% Hydro data from plexos xsl
plexos_data=pd.read_excel('plexos-export.xls',sheet_name='Properties')
```

```
hydro_plants=[hydro for hydro in plexos_data['child_object'].unique() if hydro.
 ↪startswith('Hydro')]
Generators=Generators.rename(columns={'Generator Name':'GeneratorName'})

for hydro_plant in hydro_plants:
    max_capacity= plexos_data.query('child_object == @hydro_plant and property␣
 ↪== "Max Capacity"')['value'].max()
    Generators.loc[Generators.query('GeneratorName == @hydro_plant').
 ↪index[0],'Max Capacity (MW)']=max_capacity
```

[8]:
```
types_of_gens=Generators['Type'].unique()
print(types_of_gens)
```

```
['Biomass' 'CC NG' 'CT NG' 'CT Oil' 'Geo' 'Hydro' 'ICE NG' 'Solar'
 'ST Coal' 'ST NG' 'ST Other' 'Wind']
```

Sum of the installed capacity in each bus: - For each type of generator: columns from 'Biomass (MW)' to 'ST Other (MW)' - For all non renewable synchronous generators: column 'NRES_SG' - For all renewable synchronous generators: column 'RES_SG' - For all synchronous generators: column 'Pmax_TOT_SG' (–> Total SG installed capacity in the bus) - for all converter-interfaced genetors: column 'Pmax_CIG' (–> Total CIG installed capacity in the bus) - For all generators: column 'Pmax_TOT' (–> Total installed capacity in the bus)

[9]:
```
nodes_of_connections=Generators['Node of connection'].unique()
Generators=Generators.rename(columns={'Node of connection':'Node_of_connection'})

Generators_NREL=pd.DataFrame()

for i in range(0,len(nodes_of_connections)):
    node=nodes_of_connections[i]
    bus=int(node[4:])
    Generators_NREL.loc[i,'BusNum']=bus
    Generators_NREL.loc[i,'Region']=list(Buses.query('Num == @bus')['Region'])[0]
    for t in types_of_gens:
        Generators_NREL.loc[i,t+' (MW)']=sum(Generators.
 ↪query('Node_of_connection==@node and Type == @t')['Max Capacity (MW)'])

NRES_SG=['CC NG (MW)', 'CT NG (MW)', 'CT Oil (MW)', 'ICE NG (MW)', 'ST Coal␣
 ↪(MW)', 'ST NG (MW)', 'ST Other (MW)']
RES_SG=['Geo (MW)', 'Hydro (MW)','Biomass (MW)']
TOT_SG=['Geo (MW)', 'Hydro (MW)','Biomass (MW)', 'CC NG (MW)', 'CT NG (MW)', 'CT␣
 ↪Oil (MW)', 'ICE NG (MW)', 'ST Coal (MW)', 'ST NG (MW)', 'ST Other (MW)']
CIG=['Wind (MW)','Solar (MW)']

Generators_NREL['NRES_SG']=Generators_NREL[NRES_SG].sum(axis=1)
Generators_NREL['RES_SG']=Generators_NREL[RES_SG].sum(axis=1)
Generators_NREL['Pmax_TOT_SG']=Generators_NREL[TOT_SG].sum(axis=1)
Generators_NREL['Pmax_CIG']=Generators_NREL[CIG].sum(axis=1)
```

```
Generators_NREL['Pmax_TOT']=Generators_NREL['Pmax_CIG']+Generators_NREL['Pmax_TOT_SG']
Generators_NREL
```

[9]:

	BusNum	Region	Biomass (MW)	CC NG (MW)	CT NG (MW)	CT Oil (MW)	\
0	12.0	R1	58.25	1642.78	0.00	0.0	
1	103.0	R3	40.17	55.10	0.00	0.0	
2	34.0	R2	16.50	27.00	0.00	0.0	
3	74.0	R1	0.00	100.00	0.00	0.0	
4	112.0	R3	0.00	14.30	104.40	0.0	
5	107.0	R3	0.00	834.00	7.39	0.0	
6	36.0	R2	0.00	71.51	82.40	0.0	
7	66.0	R2	0.00	596.40	0.00	0.0	
8	46.0	R2	0.00	150.00	0.00	0.0	
9	49.0	R2	0.00	262.95	50.40	0.0	
10	25.0	R1	0.00	852.50	78.40	0.0	
11	113.0	R1	0.00	619.00	0.00	0.0	
12	26.0	R1	0.00	992.80	720.00	74.5	
13	10.0	R1	0.00	1323.49	100.00	149.0	
14	4.0	R1	0.00	34.50	110.20	0.0	
15	100.0	R3	0.00	181.50	644.02	0.0	
16	89.0	R3	0.00	931.50	0.00	0.0	
17	90.0	R3	0.00	63.43	27.50	0.0	
18	87.0	R3	0.00	1356.30	0.00	0.0	
19	65.0	R2	0.00	636.00	0.00	0.0	
20	18.0	R1	0.00	247.00	29.30	0.0	
21	6.0	R1	0.00	0.00	168.40	0.0	
22	8.0	R1	0.00	0.00	150.00	0.0	
23	85.0	R3	0.00	0.00	129.99	0.0	
24	91.0	R3	0.00	0.00	215.00	0.0	
25	40.0	R2	0.00	0.00	54.00	0.0	
26	62.0	R2	0.00	0.00	25.00	0.0	
27	55.0	R2	0.00	0.00	120.70	0.0	
28	111.0	R3	0.00	0.00	421.00	0.0	
29	42.0	R2	0.00	0.00	42.00	0.0	
30	54.0	R2	0.00	0.00	0.00	142.4	
31	56.0	R2	0.00	0.00	0.00	0.0	
32	76.0	R2	0.00	0.00	0.00	0.0	
33	59.0	R2	0.00	0.00	0.00	0.0	
34	61.0	R2	0.00	0.00	0.00	0.0	
35	77.0	R2	0.00	0.00	0.00	0.0	
36	99.0	R2	0.00	0.00	0.00	0.0	
37	80.0	R2	0.00	0.00	0.00	0.0	
38	32.0	R1	0.00	0.00	0.00	0.0	
39	116.0	R2	0.00	0.00	0.00	0.0	
40	69.0	R2	0.00	0.00	0.00	0.0	
41	92.0	R3	0.00	0.00	0.00	0.0	
42	15.0	R1	0.00	0.00	0.00	0.0	

43	104.0	R3	0.00	0.00	0.00	0.0
44	110.0	R3	0.00	0.00	0.00	0.0
45	105.0	R3	0.00	0.00	0.00	0.0
46	19.0	R1	0.00	0.00	0.00	0.0
47	70.0	R1	0.00	0.00	0.00	0.0
48	72.0	R1	0.00	0.00	0.00	0.0
49	24.0	R1	0.00	0.00	0.00	0.0
50	27.0	R1	0.00	0.00	0.00	0.0
51	31.0	R1	0.00	0.00	0.00	0.0
52	82.0	R3	0.00	0.00	0.00	0.0

	Geo (MW)	Hydro (MW)	ICE NG (MW)	Solar (MW)	ST Coal (MW)	ST NG (MW)	\
0	0.0	0.00	0.00	0.00	0.0	0.00	
1	0.0	0.00	0.00	0.00	0.0	0.00	
2	0.0	4.22	8.40	0.00	0.0	0.00	
3	0.0	0.00	0.00	0.00	0.0	0.00	
4	0.0	6.00	3.25	175.80	0.0	0.00	
5	0.0	0.00	0.00	0.00	0.0	0.00	
6	0.0	8.75	0.00	0.00	0.0	0.00	
7	0.0	77.00	0.00	0.00	0.0	0.00	
8	0.0	9.08	0.00	0.00	0.0	0.00	
9	0.0	82.00	0.00	0.00	0.0	0.00	
10	0.0	0.00	0.00	0.00	0.0	712.00	
11	0.0	0.00	0.00	0.00	0.0	0.00	
12	0.0	0.00	0.00	0.00	0.0	0.00	
13	0.0	0.00	0.00	0.00	0.0	0.00	
14	0.0	0.00	0.00	0.00	0.0	0.00	
15	0.0	0.00	0.00	104.99	0.0	0.00	
16	0.0	0.00	0.00	0.00	0.0	0.00	
17	0.0	0.00	0.00	0.00	0.0	0.00	
18	0.0	0.00	0.00	0.00	0.0	0.00	
19	0.0	11.50	16.80	0.00	0.0	0.00	
20	0.0	0.00	0.00	185.52	20.0	0.00	
21	0.0	0.00	0.00	0.00	0.0	0.00	
22	0.0	0.00	0.00	0.00	0.0	0.00	
23	0.0	0.00	0.00	0.00	0.0	0.00	
24	0.0	0.00	0.00	0.00	0.0	0.00	
25	0.0	0.00	0.00	0.00	0.0	0.00	
26	0.0	142.00	0.00	0.00	0.0	0.00	
27	22.0	0.00	0.00	30.00	0.0	0.00	
28	0.0	0.00	0.00	0.00	0.0	0.00	
29	0.0	9.08	0.00	0.00	0.0	0.00	
30	0.0	0.00	0.00	264.47	0.0	0.00	
31	0.0	150.00	8.40	0.00	0.0	0.00	
32	0.0	236.15	0.00	25.00	0.0	0.00	
33	0.0	330.70	0.00	0.00	0.0	0.00	
34	0.0	323.00	0.00	0.00	0.0	0.00	

35	0.0	232.81	16.80	0.00	0.0	0.00
36	0.0	281.62	0.00	0.00	0.0	0.00
37	0.0	481.10	0.00	0.00	0.0	0.00
38	0.0	0.80	0.00	751.61	0.0	0.00
39	0.0	102.81	0.00	0.00	0.0	0.00
40	0.0	362.80	0.00	124.90	0.0	0.00
41	0.0	0.00	0.00	519.26	0.0	872.10
42	0.0	0.00	0.00	144.04	0.0	0.00
43	0.0	0.00	0.00	467.00	0.0	0.00
44	0.0	0.00	0.00	253.90	0.0	0.00
45	0.0	0.00	0.00	274.48	0.0	106.25
46	0.0	0.00	0.00	124.80	0.0	62.60
47	0.0	0.00	0.00	0.00	0.0	320.00
48	0.0	0.00	0.00	0.00	0.0	325.00
49	0.0	0.00	0.00	0.00	0.0	62.60
50	0.0	0.00	0.00	0.00	0.0	0.00
51	0.0	0.00	0.00	0.00	0.0	0.00
52	0.0	0.00	0.00	0.00	0.0	0.00

	ST Other (MW)	Wind (MW)	NRES_SG	RES_SG	Pmax_TOT_SG	Pmax_CIG	Pmax_TOT
0	0.0	0.0	1642.78	58.25	1701.03	0.00	1701.03
1	0.0	0.0	55.10	40.17	95.27	0.00	95.27
2	0.0	0.0	35.40	20.72	56.12	0.00	56.12
3	0.0	0.0	100.00	0.00	100.00	0.00	100.00
4	0.0	0.0	121.95	6.00	127.95	175.80	303.75
5	0.0	0.0	841.39	0.00	841.39	0.00	841.39
6	0.0	0.0	153.91	8.75	162.66	0.00	162.66
7	0.0	0.0	596.40	77.00	673.40	0.00	673.40
8	0.0	0.0	150.00	9.08	159.08	0.00	159.08
9	0.0	0.0	313.35	82.00	395.35	0.00	395.35
10	0.0	0.0	1642.90	0.00	1642.90	0.00	1642.90
11	0.0	0.0	619.00	0.00	619.00	0.00	619.00
12	0.0	0.0	1787.30	0.00	1787.30	0.00	1787.30
13	0.0	0.0	1572.49	0.00	1572.49	0.00	1572.49
14	0.0	0.0	144.70	0.00	144.70	0.00	144.70
15	0.0	400.0	825.52	0.00	825.52	504.99	1330.51
16	0.0	0.0	931.50	0.00	931.50	0.00	931.50
17	0.0	0.0	90.93	0.00	90.93	0.00	90.93
18	0.0	0.0	1356.30	0.00	1356.30	0.00	1356.30
19	0.0	0.0	652.80	11.50	664.30	0.00	664.30
20	0.0	0.0	296.30	0.00	296.30	185.52	481.82
21	0.0	0.0	168.40	0.00	168.40	0.00	168.40
22	0.0	0.0	150.00	0.00	150.00	0.00	150.00
23	0.0	0.0	129.99	0.00	129.99	0.00	129.99
24	0.0	0.0	215.00	0.00	215.00	0.00	215.00
25	0.0	0.0	54.00	0.00	54.00	0.00	54.00
26	0.0	0.0	25.00	142.00	167.00	0.00	167.00

27	0.0	0.0	120.70	22.00	142.70	30.00	172.70
28	0.0	0.0	421.00	0.00	421.00	0.00	421.00
29	0.0	0.0	42.00	9.08	51.08	0.00	51.08
30	0.0	0.0	142.40	0.00	142.40	264.47	406.87
31	0.0	0.0	8.40	150.00	158.40	0.00	158.40
32	0.0	0.0	0.00	236.15	236.15	25.00	261.15
33	0.0	0.0	0.00	330.70	330.70	0.00	330.70
34	0.0	0.0	0.00	323.00	323.00	0.00	323.00
35	0.0	0.0	16.80	232.81	249.61	0.00	249.61
36	0.0	0.0	0.00	281.62	281.62	0.00	281.62
37	0.0	0.0	0.00	481.10	481.10	0.00	481.10
38	0.0	0.0	0.00	0.80	0.80	751.61	752.41
39	0.0	0.0	0.00	102.81	102.81	0.00	102.81
40	0.0	0.0	0.00	362.80	362.80	124.90	487.70
41	0.0	0.0	872.10	0.00	872.10	519.26	1391.36
42	0.0	0.0	0.00	0.00	0.00	144.04	144.04
43	0.0	0.0	0.00	0.00	0.00	467.00	467.00
44	0.0	0.0	0.00	0.00	0.00	253.90	253.90
45	0.0	0.0	106.25	0.00	106.25	274.48	380.73
46	0.0	0.0	62.60	0.00	62.60	124.80	187.40
47	0.0	0.0	320.00	0.00	320.00	0.00	320.00
48	0.0	0.0	325.00	0.00	325.00	0.00	325.00
49	35.0	65.6	97.60	0.00	97.60	65.60	163.20
50	0.0	107.1	0.00	0.00	0.00	107.10	107.10
51	0.0	156.2	0.00	0.00	0.00	156.20	156.20
52	0.0	349.0	0.00	0.00	0.00	349.00	349.00

[10]:
```
pd.DataFrame.to_excel(Generators_NREL,'Generators_NREL.xlsx')
```

- Pmax_SG: installed capacity of SG
- Pmax_CIG: installed capacity of CIG

Wind, Solar, and Hydro Generation Hourly generation (of each generator) for one year (Information that is not used for the setup but can be used for the implementation of the optimization)

[9]:
```
from os import listdir
from os.path import isfile, join

def res_gen(res,df):
    path='./input-files/Input files/RT/'+res+'/'
    onlyfiles = [f for f in listdir(path) if isfile(join(path, f))]

    onlyfiles=list(set(onlyfiles)-set(['.DS_Store']))

    for file in onlyfiles:
        df_i=pd.read_csv(path+file)
```

```
        df_i["DATETIME"] = pd.to_datetime(df_i["DATETIME"])

        df_i=df_i.set_index(["DATETIME"]).sort_index()

        df[file[:-6]]=df_i

    return df

Solar=pd.DataFrame()
Wind=pd.DataFrame()
Hydro=pd.DataFrame()

Solar=res_gen('Solar',Solar)
Wind=res_gen('Wind',Wind)
Hydro=res_gen('Hydro',Wind)
```

[10]:
```
Wind
```

[10]:
```
                        Wind11     Wind3      Wind6     Wind10     Wind13  \
DATETIME
2024-01-01 00:00:00  5.058859  2.477267   0.877676  12.765404   1.687492
2024-01-01 01:00:00  9.208173  3.031204   2.008854  13.682279   4.078827
2024-01-01 02:00:00  9.190788  3.089380   2.794547  12.681899   5.380704
2024-01-01 03:00:00  9.695205  2.486115   9.659556  14.305784  26.623678
2024-01-01 04:00:00  9.710853  2.629418  10.975506  12.569062  29.804588
...                       ...       ...        ...        ...        ...
2025-01-01 19:00:00  0.000000  1.546054   0.678839   0.000000   2.388373
2025-01-01 20:00:00  0.000407  3.779966   0.702487   0.000000   2.684599
2025-01-01 21:00:00  0.000000  5.598673   0.644244   0.000000   1.392421
2025-01-01 22:00:00  0.000000  6.775923   0.524096   0.003875   0.981058
2025-01-01 23:00:00  0.000000  6.884240   0.392321   0.041392   0.461570

                         Wind2     Wind17     Wind12      Wind9     Wind16  \
DATETIME
2024-01-01 00:00:00   3.724274   9.669997   0.822722  11.693075   0.297516
2024-01-01 01:00:00   8.090245   5.947432   3.046363  12.962309   0.051735
2024-01-01 02:00:00  13.335597   3.205198   9.545811  11.407550   0.000000
2024-01-01 03:00:00  23.850555   2.940856  53.627804  13.585817   1.635511
2024-01-01 04:00:00  23.962612   0.815381  51.880845  11.807606   3.470530
...                        ...        ...        ...        ...        ...
2025-01-01 19:00:00   0.000000  29.618573   8.322766   0.000000  10.629521
2025-01-01 20:00:00   0.000000   8.286292   4.277697   0.000000   6.808885
2025-01-01 21:00:00   0.000000   2.778807   1.525636   0.001902   4.329162
2025-01-01 22:00:00   0.000000   8.801680   5.662659   0.013554   3.542115
2025-01-01 23:00:00   0.000000  21.471366   4.765764   0.056128   7.185423

                        Wind4     Wind14     Wind1     Wind7      Wind8  \
```

```
DATETIME
2024-01-01 00:00:00  0.769740   4.500030  0.458135  1.942837  13.847484
2024-01-01 01:00:00  1.013277   6.411089  0.900835  4.622050  14.572063
2024-01-01 02:00:00  0.506920  11.625729  2.412459  6.191502  13.677956
2024-01-01 03:00:00  0.367775  26.558630  5.340160  9.045088  15.210591
2024-01-01 04:00:00  0.349065  25.271533  5.856456  9.334108  14.149706
...                       ...        ...       ...       ...        ...
2025-01-01 19:00:00  0.239122  14.386462  0.000780  0.000204   0.000000
2025-01-01 20:00:00  0.730586   6.112000  0.002851  0.003298   0.000000
2025-01-01 21:00:00  1.788015   1.310316  0.002244  0.003825   0.000000
2025-01-01 22:00:00  3.335575   3.786628  0.000000  0.000000   0.006827
2025-01-01 23:00:00  3.653633  10.636013  0.000000  0.000000   0.058626

                        Wind5     Wind15   Hydro
DATETIME
2024-01-01 00:00:00  0.902116   8.252688  0.02212
2024-01-01 01:00:00  0.951623   8.222057  0.02133
2024-01-01 02:00:00  0.904939   8.338495  0.02142
2024-01-01 03:00:00  0.978896   9.373620  0.02160
2024-01-01 04:00:00  0.906345   8.275774  0.02162
...                       ...        ...      ...
2025-01-01 19:00:00  0.000000  27.048672  0.07282
2025-01-01 20:00:00  0.000000  25.178994  0.07312
2025-01-01 21:00:00  0.000000  23.885157  0.07341
2025-01-01 22:00:00  0.000076  24.351647  0.07364
2025-01-01 23:00:00  0.001847  23.847103  0.07381

[8784 rows x 18 columns]
```

1.2.4 Summary

Regions description

[10]:
```
def regions_description(reg_list,Generators, Loads, Buses, regions=pd.
↪DataFrame(),regions_gen_perc=pd.DataFrame(), regions_red=pd.DataFrame()):
    for r in range(len(reg_list)):
        R=reg_list[r]
        regions.loc[r,'Region']=R
        regions_gen_perc.loc[r,'Region']=R
        gen_reg=Generators.query('Region == @R')
        regions.loc[r,'N_gens']=len(gen_reg)
        regions.loc[r,'N_Wind']=len(gen_reg.query('Type == "Wind"'))
        regions.loc[r,'N_Solar']=len(gen_reg.query('Type == "Solar"'))


        buses_reg=[int(bus.replace('node','')) for bus in␣
↪gen_reg['Node_of_connection'].unique()]
        regions.loc[r,'Buses list']=str(buses_reg)
```

```
        regions.loc[r,'Num_Bus_gen']=len(buses_reg)


        regions.loc[r,'Capacity (MW)']=gen_reg['Max Capacity (MW)'].sum()

        regions.loc[r,'Peak Load (MW)']=Loads[R].max()
        regions.loc[r,'Min Load (MW)']=Loads[R].min()


        for t in types_of_gens:#TOT_SG+CIG:
            regions_gen_perc.loc[r,t+' [%]']=sum(gen_reg.query('Type ==␣
↪@t')['Max Capacity (MW)'])/regions.loc[r,'Capacity (MW)']*100
            regions.loc[r,t+' [MW]']=sum(gen_reg.query('Type == @t')['Max␣
↪Capacity (MW)'])

        regions_gen_perc.loc[r,'NRES_SG [%]']= sum(regions_gen_perc.
↪loc[r,NRES_SG_perc])
        regions_gen_perc.loc[r,'RES_SG [%]']= sum(regions_gen_perc.
↪loc[r,RES_SG_perc])
        regions.loc[r,'TOT_SG [%]']= sum(regions_gen_perc.loc[r,TOT_SG_perc])
        regions_gen_perc.loc[r,'CIG [%]']= sum(regions_gen_perc.loc[r,CIG_perc])

        regions.loc[r,'NRES_SG [MW]']= sum(regions.loc[r,NRES_SG])
        regions.loc[r,'RES_SG [MW]']= sum(regions.loc[r,RES_SG])
        regions.loc[r,'TOT_SG [MW]']= sum(regions.loc[r,TOT_SG])
        regions.loc[r,'CIG [MW]']= sum(regions.loc[r,CIG])

        regions.loc[r,'NRES_SG [%]']= regions_gen_perc.loc[r,'NRES_SG [%]']
        regions.loc[r,'RES_SG [%]']= regions_gen_perc.loc[r,'RES_SG [%]']
        regions.loc[r,'CIG [%]']= regions_gen_perc.loc[r,'CIG [%]']
        regions.loc[r,'TOT_SG [%]']= regions.loc[r,'TOT_SG [%]']

        regions_red=regions.drop(TOT_SG+CIG,axis=1)

    return regions, regions_gen_perc, regions_red

def pie_plot(y,labels,title,labels_flag):
    fig=plt.figure()

    if labels_flag:
        p, tx, autotexts = plt.pie(y, labels=labels,textprops={'fontsize': 20},
                autopct="", shadow=True)
        for i, a in enumerate(autotexts):
            a.set_text("{0:.0f}".format(np.array(y)[i]))
    else:
        p, tx, autotexts = plt.pie(y, textprops={'fontsize': 20},
                autopct="", shadow=True)
```

```
        plt.legend(labels)

    plt.title(title,fontsize=20)

    fig.tight_layout
```

[11]:
```
NRES_SG=['CC NG [MW]', 'CT NG [MW]', 'CT Oil [MW]', 'ICE NG [MW]', 'ST Coal␣
 ↪[MW]', 'ST NG [MW]', 'ST Other [MW]']
RES_SG=['Geo [MW]', 'Hydro [MW]','Biomass [MW]']
TOT_SG=['Geo [MW]', 'Hydro [MW]','Biomass [MW]', 'CC NG [MW]', 'CT NG [MW]', 'CT␣
 ↪Oil [MW]', 'ICE NG [MW]', 'ST Coal [MW]', 'ST NG [MW]', 'ST Other [MW]']
CIG=['Wind [MW]','Solar [MW]']

NRES_SG_perc=['CC NG [%]', 'CT NG [%]', 'CT Oil [%]', 'ICE NG [%]', 'ST Coal␣
 ↪[%]', 'ST NG [%]', 'ST Other [%]']
RES_SG_perc=['Geo [%]', 'Hydro [%]','Biomass [%]']
TOT_SG_perc=['Geo [%]', 'Hydro [%]','Biomass [%]', 'CC NG [%]', 'CT NG [%]', 'CT␣
 ↪Oil [%]', 'ICE NG [%]', 'ST Coal [%]', 'ST NG [%]', 'ST Other [%]']
CIG_perc=['Wind [%]','Solar [%]']


reg_list=['R1','R2','R3']
regions=pd.DataFrame()
regions_gen_perc=pd.DataFrame()

regions, regions_gen_perc, regions_red= regions_description(reg_list,␣
 ↪Generators, Loads, Buses)
```

[31]:
```
regions

#- N_gens: number of generators (not generation units) installed in the region
#- N_Wind, N_Solar: number of wind/solar power plants
#- Buses list: list of buses with a generator unit
#- Num_Bus_gen: number of buses with generator unit
#- Capacity (MW): total installed generation capacity of the region
#- Peak Load (MW), Min Load (MW): peak and minimum load of the region
#- total installed capacity of the region, differentiated by type of generator
#- total NRES_SG, RES_SG, TOT_SG, TOT_CIG per region, in MW and in percentage
```

[31]:

	Region	N_gens	N_Wind	N_Solar	\
0	R1	136.0	13.0	33.0	
1	R2	72.0	0.0	5.0	
2	R3	119.0	4.0	37.0	

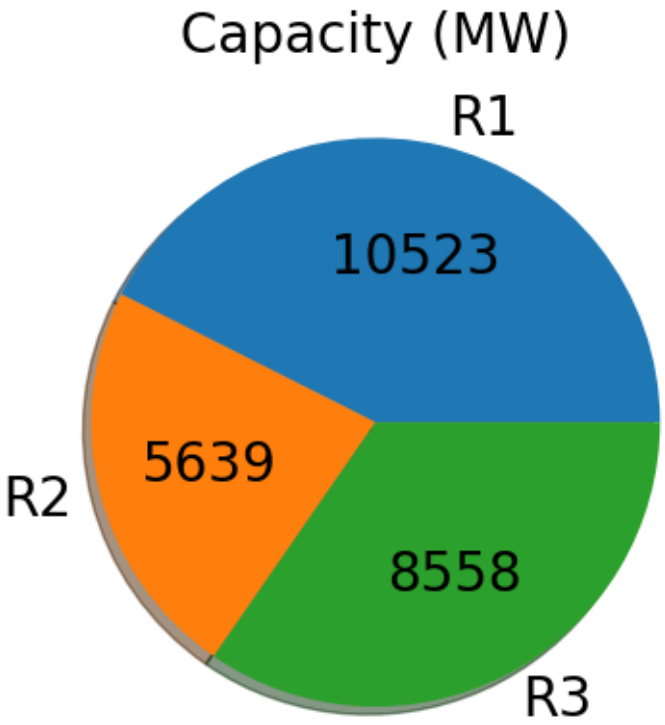
	Buses list	Num_Bus_gen	\
0	[12, 74, 25, 113, 26, 10, 4, 18, 6, 8, 32, 15,...	18.0	
1	[34, 36, 66, 46, 49, 65, 40, 62, 55, 42, 54, 5...	20.0	

```
2  [103, 112, 107, 100, 89, 90, 87, 85, 91, 111, ...         15.0

   Capacity (MW)  Peak Load (MW)  Min Load (MW)  Biomass [MW]  ...  \
0       10522.99     9276.327882    4228.311838         58.25  ...
1        5638.65     4803.546377    1190.362815         16.50  ...
2        8557.63     5290.274654    1715.314180         40.17  ...

   ST Other [MW]  Wind [MW]  TOT_SG [%]  NRES_SG [MW]  RES_SG [MW]  \
0           35.0      328.9   85.414127       8929.07        59.05
1            0.0        0.0   92.119213       2311.16      2883.12
2            0.0      749.0   70.267118       5967.03        46.17

   TOT_SG [MW]  CIG [MW]  NRES_SG [%]  RES_SG [%]    CIG [%]
0      8988.12   1534.87    84.852974    0.561152  14.585873
1      5194.28    444.37    40.987825   51.131388   7.880787
2      6013.20   2544.43    69.727600    0.539519  29.732882

[3 rows x 29 columns]
```

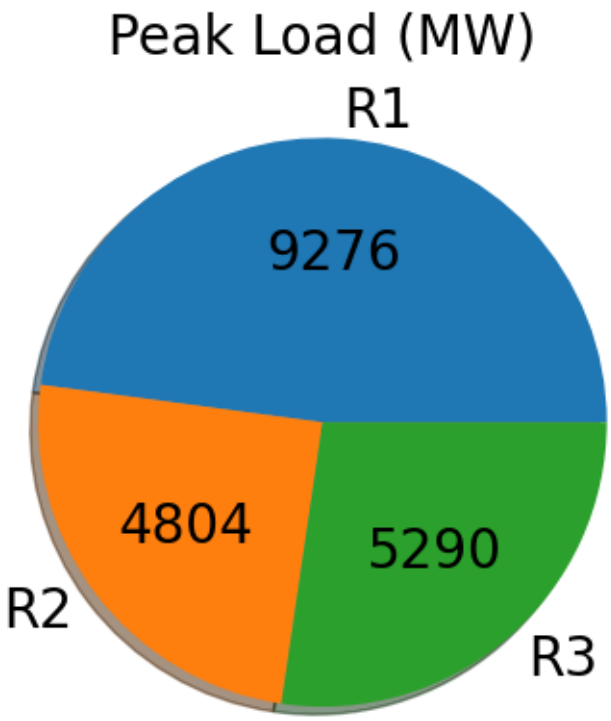
[13]:
```
regions_red
```

[13]:
```
  Region  N_gens  N_Wind  N_Solar  \
0     R1   136.0    13.0     33.0
1     R2    72.0     0.0      5.0
2     R3   119.0     4.0     37.0

                                        Buses list  Num_Bus_gen  \
0  [12, 74, 25, 113, 26, 10, 4, 18, 6, 8, 32, 15,...         18.0
1  [34, 36, 66, 46, 49, 65, 40, 62, 55, 42, 54, 5...         20.0
2  [103, 112, 107, 100, 89, 90, 87, 85, 91, 111, ...         15.0

   Capacity (MW)  Peak Load (MW)  Min Load (MW)  TOT_SG [%]  NRES_SG [MW]  \
0       10522.99     9276.327882    4228.311838   85.414127       8929.07
1        5638.65     4803.546377    1190.362815   92.119213       2311.16
2        8557.63     5290.274654    1715.314180   70.267118       5967.03

   RES_SG [MW]  TOT_SG [MW]  CIG [MW]  NRES_SG [%]  RES_SG [%]    CIG [%]
0        59.05      8988.12   1534.87    84.852974    0.561152  14.585873
1      2883.12      5194.28    444.37    40.987825   51.131388   7.880787
2        46.17      6013.20   2544.43    69.727600    0.539519  29.732882
```

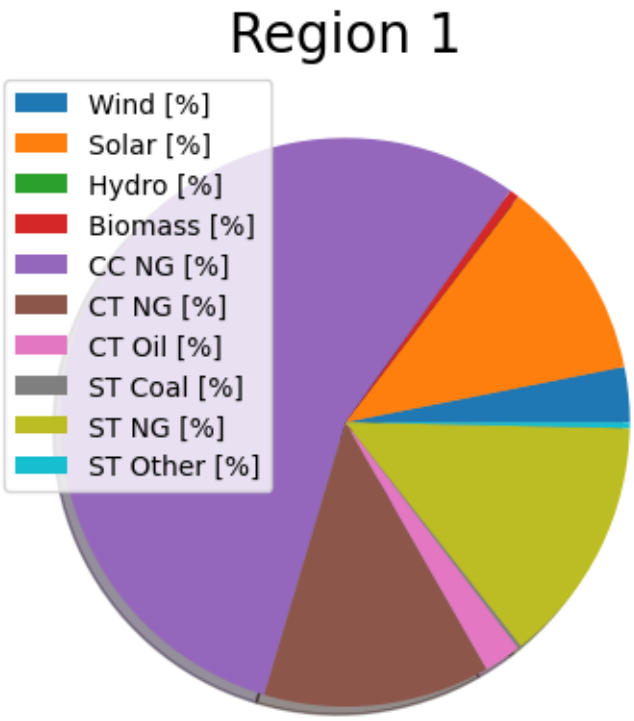
[24]:
```
title='Capacity (MW)'
y = regions[title]
mylabels = regions['Region']

pie_plot(y,mylabels,title,labels_flag=1)
```

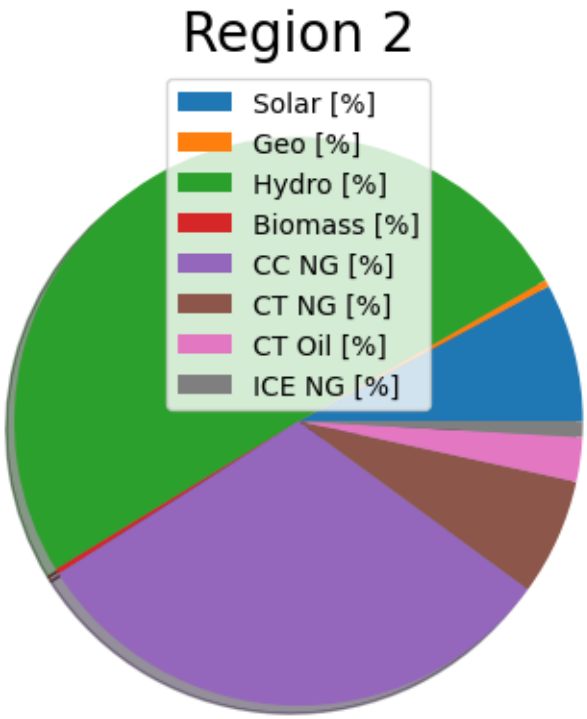


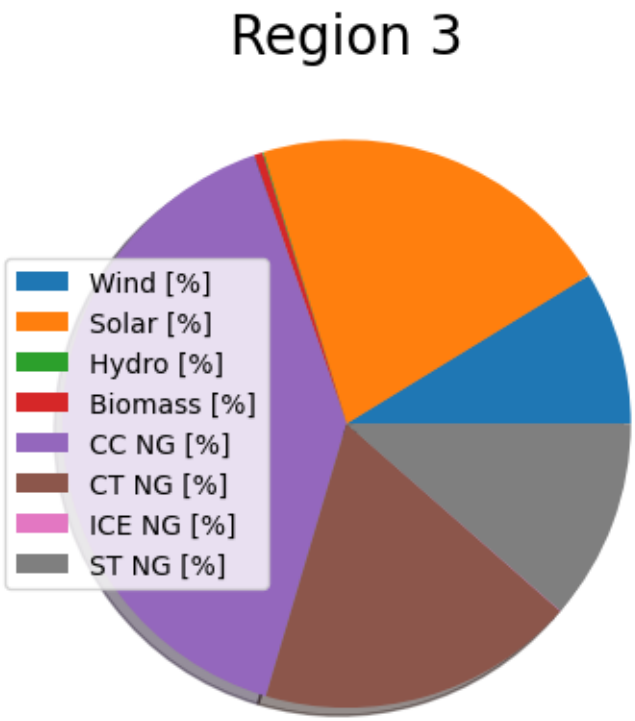
[25]:
```
title='Peak Load (MW)'
y = regions[title]
mylabels = regions['Region']

pie_plot(y,mylabels,title,labels_flag=1)
```



[27]:
```
for r in range(1,len(reg_list)+1):
    title='Region '+str(r)
    y = regions_gen_perc.loc[r-1,CIG_perc+TOT_SG_perc][regions_gen_perc.
↪loc[r-1,CIG_perc+TOT_SG_perc]!=0]
    mylabels = y.index

    pie_plot(y,mylabels,title,labels_flag=0)
```


Region 1
Wind [%]
Solar [%]
Hydro [%]
Biomass [%]
CC NG [%]
CT NG [%]
CT Oil [%]
ST Coal [%]
ST NG [%]
ST Other [%]


Region 2
Solar [%]
Geo [%]
Hydro [%]
Biomass [%]
CC NG [%]
CT NG [%]
CT Oil [%]
ICE NG [%]



[19]:
```
def create_OpDataExcel(Buses, Generators_sys, columns_order):

    T_Loads=Buses.query('Load_Participation_Factor !=00')
    T_Loads['Load_Participation_Factor']=T_Loads['Load_Participation_Factor']/3
    T_Buses=Buses[['Bus Name','Region']]

    T_Gen=Generators_sys[['BusNum','Pmax_CIG']]
    # T_Gen['BusName']=Generators_sys[['BusName']]
    T_Gen['Pmax_SG']=Generators_sys['Pmax_TOT_SG']
    T_Gen['Pmax']=np.array(T_Gen[['Pmax_SG']])+np.array(T_Gen[['Pmax_CIG']])
    T_Gen['Snom']=T_Gen['Pmax']/0.95
    T_Gen['Snom_SG']=T_Gen['Pmax_SG']/0.95
    T_Gen['Snom_CIG']=T_Gen['Pmax_CIG']/0.95
    T_Gen['Pmin']=T_Gen['Snom']*0.2
    T_Gen['Pmin_SG']=T_Gen['Snom_SG']*0.2
    T_Gen['Pmin_CIG']=T_Gen['Snom_CIG']*0.2
    T_Gen['Qmax']=T_Gen['Pmax']*0.33
    T_Gen['Qmin']=-T_Gen['Pmin']*0.33
    T_Gen['Region']=Generators_sys['Region']
```

```
    T_Gen=T_Gen[columns_order]

    return T_Gen, T_Loads, T_Buses
```

Tables with information about Generators, Loads, and Buses to be provided to the Data Generation Tool

[32]:
```
columns_order=['BusNum','Snom_SG','Snom_CIG','Snom','Pmax','Pmin','Qmax','Qmin','Region','Pmax_

T_Gen, T_Loads, T_Buses = create_OpDataExcel(Buses, Generators_NREL,␣
↪columns_order)
```

- BusNum: number of the bus
- Pmax_SG, Pmax_CIG = maximum capacity of SG and CIG in the bus (MW)
- Pmax = Pmax_SG + Pmax_CIG
- Snom_SG: total SG rated power (in MVA) = Pmax_SG/0.95
- Snom_CIG: total CIG rated power (in MVA) = Pmax_CIG/0.95
- Snom = Snom_SG + Snom_CIG
- Pmin_SG = 20% of Snom_SG
- Pmin_CIG = 20% of Snom_SG
- Pmin = Pmin_SG + Pmin_CIG
- Qmax = 0.33*Pmax (assume cos(phi)=0.95)
- Qmin = -0.33*Pmax (assume cos(phi)=0.95)

[33]:
```
T_Gen
```

[33]:

	BusNum	Snom_SG	Snom_CIG	Snom	Pmax	Pmin	\
0	12.0	1790.557895	0.000000	1790.557895	1701.03	358.111579	
1	103.0	100.284211	0.000000	100.284211	95.27	20.056842	
2	34.0	59.073684	0.000000	59.073684	56.12	11.814737	
3	74.0	105.263158	0.000000	105.263158	100.00	21.052632	
4	112.0	134.684211	185.052632	319.736842	303.75	63.947368	
5	107.0	885.673684	0.000000	885.673684	841.39	177.134737	
6	36.0	171.221053	0.000000	171.221053	162.66	34.244211	
7	66.0	708.842105	0.000000	708.842105	673.40	141.768421	
8	46.0	167.452632	0.000000	167.452632	159.08	33.490526	
9	49.0	416.157895	0.000000	416.157895	395.35	83.231579	
10	25.0	1729.368421	0.000000	1729.368421	1642.90	345.873684	
11	113.0	651.578947	0.000000	651.578947	619.00	130.315789	
12	26.0	1881.368421	0.000000	1881.368421	1787.30	376.273684	
13	10.0	1655.252632	0.000000	1655.252632	1572.49	331.050526	
14	4.0	152.315789	0.000000	152.315789	144.70	30.463158	
15	100.0	868.968421	531.568421	1400.536842	1330.51	280.107368	
16	89.0	980.526316	0.000000	980.526316	931.50	196.105263	
17	90.0	95.715789	0.000000	95.715789	90.93	19.143158	
18	87.0	1427.684211	0.000000	1427.684211	1356.30	285.536842	
19	65.0	699.263158	0.000000	699.263158	664.30	139.852632	
20	18.0	311.894737	195.284211	507.178947	481.82	101.435789	

21	6.0	177.263158	0.000000	177.263158	168.40	35.452632
22	8.0	157.894737	0.000000	157.894737	150.00	31.578947
23	85.0	136.831579	0.000000	136.831579	129.99	27.366316
24	91.0	226.315789	0.000000	226.315789	215.00	45.263158
25	40.0	56.842105	0.000000	56.842105	54.00	11.368421
26	62.0	175.789474	0.000000	175.789474	167.00	35.157895
27	55.0	150.210526	31.578947	181.789474	172.70	36.357895
28	111.0	443.157895	0.000000	443.157895	421.00	88.631579
29	42.0	53.768421	0.000000	53.768421	51.08	10.753684
30	54.0	149.894737	278.389474	428.284211	406.87	85.656842
31	56.0	166.736842	0.000000	166.736842	158.40	33.347368
32	76.0	248.578947	26.315789	274.894737	261.15	54.978947
33	59.0	348.105263	0.000000	348.105263	330.70	69.621053
34	61.0	340.000000	0.000000	340.000000	323.00	68.000000
35	77.0	262.747368	0.000000	262.747368	249.61	52.549474
36	99.0	296.442105	0.000000	296.442105	281.62	59.288421
37	80.0	506.421053	0.000000	506.421053	481.10	101.284211
38	32.0	0.842105	791.168421	792.010526	752.41	158.402105
39	116.0	108.221053	0.000000	108.221053	102.81	21.644211
40	69.0	381.894737	131.473684	513.368421	487.70	102.673684
41	92.0	918.000000	546.589474	1464.589474	1391.36	292.917895
42	15.0	0.000000	151.621053	151.621053	144.04	30.324211
43	104.0	0.000000	491.578947	491.578947	467.00	98.315789
44	110.0	0.000000	267.263158	267.263158	253.90	53.452632
45	105.0	111.842105	288.926316	400.768421	380.73	80.153684
46	19.0	65.894737	131.368421	197.263158	187.40	39.452632
47	70.0	336.842105	0.000000	336.842105	320.00	67.368421
48	72.0	342.105263	0.000000	342.105263	325.00	68.421053
49	24.0	102.736842	69.052632	171.789474	163.20	34.357895
50	27.0	0.000000	112.736842	112.736842	107.10	22.547368
51	31.0	0.000000	164.421053	164.421053	156.20	32.884211
52	82.0	0.000000	367.368421	367.368421	349.00	73.473684

	Qmax	Qmin	Region	Pmax_SG	Pmax_CIG	Pmin_SG	Pmin_CIG
0	561.3399	-118.176821	R1	1701.03	0.00	358.111579	0.000000
1	31.4391	-6.618758	R3	95.27	0.00	20.056842	0.000000
2	18.5196	-3.898863	R2	56.12	0.00	11.814737	0.000000
3	33.0000	-6.947368	R1	100.00	0.00	21.052632	0.000000
4	100.2375	-21.102632	R3	127.95	175.80	26.936842	37.010526
5	277.6587	-58.454463	R3	841.39	0.00	177.134737	0.000000
6	53.6778	-11.300589	R2	162.66	0.00	34.244211	0.000000
7	222.2220	-46.783579	R2	673.40	0.00	141.768421	0.000000
8	52.4964	-11.051874	R2	159.08	0.00	33.490526	0.000000
9	130.4655	-27.466421	R2	395.35	0.00	83.231579	0.000000
10	542.1570	-114.138316	R1	1642.90	0.00	345.873684	0.000000
11	204.2700	-43.004211	R1	619.00	0.00	130.315789	0.000000
12	589.8090	-124.170316	R1	1787.30	0.00	376.273684	0.000000

13	518.9217	-109.246674	R1	1572.49	0.00	331.050526	0.000000
14	47.7510	-10.052842	R1	144.70	0.00	30.463158	0.000000
15	439.0683	-92.435432	R3	825.52	504.99	173.793684	106.313684
16	307.3950	-64.714737	R3	931.50	0.00	196.105263	0.000000
17	30.0069	-6.317242	R3	90.93	0.00	19.143158	0.000000
18	447.5790	-94.227158	R3	1356.30	0.00	285.536842	0.000000
19	219.2190	-46.151368	R2	664.30	0.00	139.852632	0.000000
20	159.0006	-33.473811	R1	296.30	185.52	62.378947	39.056842
21	55.5720	-11.699368	R1	168.40	0.00	35.452632	0.000000
22	49.5000	-10.421053	R1	150.00	0.00	31.578947	0.000000
23	42.8967	-9.030884	R3	129.99	0.00	27.366316	0.000000
24	70.9500	-14.936842	R3	215.00	0.00	45.263158	0.000000
25	17.8200	-3.751579	R2	54.00	0.00	11.368421	0.000000
26	55.1100	-11.602105	R2	167.00	0.00	35.157895	0.000000
27	56.9910	-11.998105	R2	142.70	30.00	30.042105	6.315789
28	138.9300	-29.248421	R3	421.00	0.00	88.631579	0.000000
29	16.8564	-3.548716	R2	51.08	0.00	10.753684	0.000000
30	134.2671	-28.266758	R2	142.40	264.47	29.978947	55.677895
31	52.2720	-11.004632	R2	158.40	0.00	33.347368	0.000000
32	86.1795	-18.143053	R2	236.15	25.00	49.715789	5.263158
33	109.1310	-22.974947	R2	330.70	0.00	69.621053	0.000000
34	106.5900	-22.440000	R2	323.00	0.00	68.000000	0.000000
35	82.3713	-17.341326	R2	249.61	0.00	52.549474	0.000000
36	92.9346	-19.565179	R2	281.62	0.00	59.288421	0.000000
37	158.7630	-33.423789	R2	481.10	0.00	101.284211	0.000000
38	248.2953	-52.272695	R1	0.80	751.61	0.168421	158.233684
39	33.9273	-7.142589	R2	102.81	0.00	21.644211	0.000000
40	160.9410	-33.882316	R2	362.80	124.90	76.378947	26.294737
41	459.1488	-96.662905	R3	872.10	519.26	183.600000	109.317895
42	47.5332	-10.006989	R1	0.00	144.04	0.000000	30.324211
43	154.1100	-32.444211	R3	0.00	467.00	0.000000	98.315789
44	83.7870	-17.639368	R3	0.00	253.90	0.000000	53.452632
45	125.6409	-26.450716	R3	106.25	274.48	22.368421	57.785263
46	61.8420	-13.019368	R1	62.60	124.80	13.178947	26.273684
47	105.6000	-22.231579	R1	320.00	0.00	67.368421	0.000000
48	107.2500	-22.578947	R1	325.00	0.00	68.421053	0.000000
49	53.8560	-11.338105	R1	97.60	65.60	20.547368	13.810526
50	35.3430	-7.440632	R1	0.00	107.10	0.000000	22.547368
51	51.5460	-10.851789	R1	0.00	156.20	0.000000	32.884211
52	115.1700	-24.246316	R3	0.00	349.00	0.000000	73.473684

Loads with Load_Participation_Factor as a fraction of total power demand

[22]:
```
T_Loads
```

[22]:

	Bus Name	Region	Load_Participation_Factor	Num
0	bus001	R1	0.015723	1.0
1	bus002	R1	0.006165	2.0

```
2     bus003     R1                  0.012023    3.0
3     bus004     R1                  0.009250    4.0
5     bus006     R1                  0.016031    6.0
..       ...    ...                       ...    ...
111   bus112     R3                  0.011206  112.0
113   bus114     R1                  0.002466  114.0
114   bus115     R1                  0.006781  115.0
116   bus117     R1                  0.006165  117.0
117   bus118     R2                  0.005976  118.0

[91 rows x 4 columns]
```

[23]:
```
T_Buses
```

[23]:
```
    Bus Name Region
0     bus001     R1
1     bus002     R1
2     bus003     R1
3     bus004     R1
4     bus005     R1
..       ...    ...
113   bus114     R1
114   bus115     R1
115   bus116     R2
116   bus117     R1
117   bus118     R2

[118 rows x 2 columns]
```

1.3 NREL-118bus System: Information for Dynamic Analysis

Conventional values (i.e. default values in the tool excel file) of control parameters and component sizing were adjusted after performing the small-signal stability assessment (eigenvalues and participation factors analysis) of several operating points.

1.3.1 Model of the GFOL Converter:

All the unstable cases showed high participation of the state variables related to the PLL (GFOLN_etheta_x). Therefore, the PLL has been made slower (with tau_pll=0.4 s). Coherently, the active power and reactive power control time constants have been increased.

	ts_pll	tau_p	tau_q
Previous values	0.02	0.1	0.1
New values	0.4	0.4	0.4

1.3.2 Model of the GFOR Converter:

Many unstable cases showed high participation of state variables related to the current of GFOR converters on the grid side (GFORN_ig_). Therefore, the resistance of the transformer connecting the converter to the grid is increased.

	Rtr
Previous values	0.002
New values	0.005

1.3.3 Model of the SG:

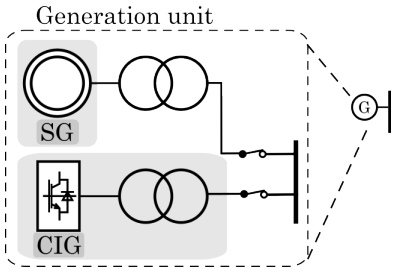
Many unstable cases showed high participation of state variables related to - SGN_is_d - SGN_if_d

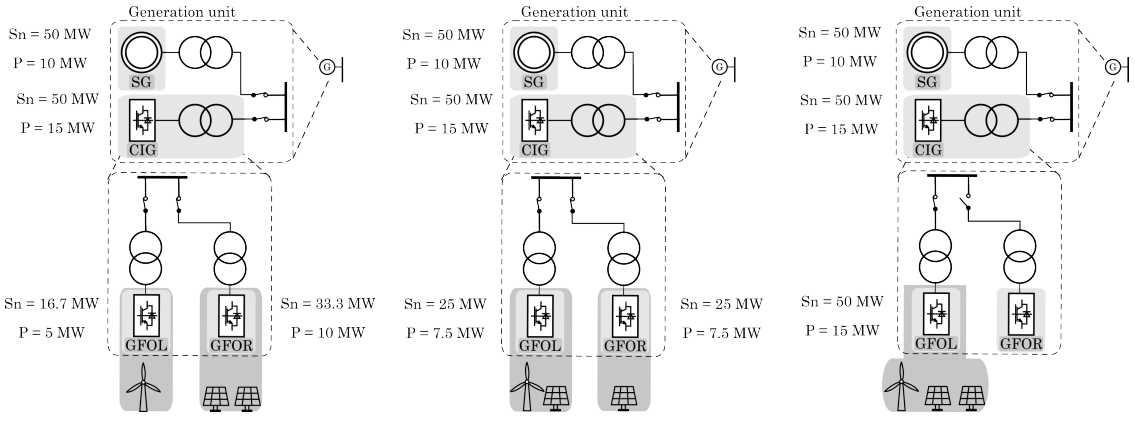
Therefore, following parameters have been changed.

	Rs	Xd	Xd_tr	Rtr
Previous values	0.0025	1.8	0.3	0.002
New values	0.005	3	0.1	0.005

1.3.4 Generation Unit

Example of operation:





ACOPF_standalone_NREL_LF09_seed16_nc3_ns100_20241022_175408_4639 Feasible cases: 62.66666666666667% Stable cases: 44.0% of total cases Stable cases: 70.2127659574468% of feasible cases %% ACOPF_standalone_NREL_LF1_seed16_nc3_n Feasible cases: 49.0% Stable cases: 34.333333333333336% of total cases Stable cases: 70.06802721088435% of feasible cases %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```
[ ]:
```