# Supplementary Material

## Impact of COVID-19 on Electricity Demand: Deriving Minimum States of System Health for Studies on Resilience

Smruti Manjunath[1,*], Madhura Yeligeti[1], Maria Fyta[2], Jannik Haas[1,3], Hans-Christian Gils[1,*]

[1] Department of Energy Systems Analysis, Institute of Networked Energy Systems, German Aerospace Center (DLR), Germany
[2] Institute for Computational Physics, University of Stuttgart, Germany
[3] Department of Stochastic Simulation and Safety Research for Hydrosystems (IWS/SC SimTech), University of Stuttgart, Germany

* Corresponding authors: Smruti Manjunath (Smruti.Manjunath@dlr.de); Hans-Christian Gils (Hans-Christian.Gils@dlr.de)

Table of Contents

**Code** ..... 2
Step 1: Defining functions for each country ..... 2
Concatenating individual data files and cleaning data ..... 2
Step 2: Passing pathname of folders, running functions and assigning data frames ..... 5
Glossary: ..... 5
Step 3: Creating the consolidated, representative data frames ..... 8
Step 4: Creating the percentage-decrease data frames ..... 13

# Code

```
import pandas as pd
import datetime as dt
import numpy as np
import glob
```

## Step 1: Defining functions for each country

### Concatenating individual data files and cleaning data

To be used for WFH, SLD and FLD for both weekdays and weekends for each country.

Each individual phase has a separate folder: the downloaded power demand data for every weekday in a phase constitute a folder.

These functions load the folder having pathname "path" and concatenate all individual data files (in CSV format) in the folder into a single data-frame for each country and separately for weekdays and weekends.

Next, the functions drop unnecessary columns from this data-frame and convert the recorded time into the date–time format.

Finally, the function returns this updated data-frame for both weekdays and weekends for each country.

```
def HourlyLoadProfile_Germany(path):

    """
    Generates the updated data-frame for both weekdays and weekends for Germany.

    In addition to the description above,
    this functional also resamples the load to an hourly resolution.


    """

    #to concatenate all files found in the folder having pathname "path"
    all_files = glob.glob(path + "/*.csv")

    li = []

    for filename in all_files:
        frame = pd.read_csv(filename, index_col=None, header=0)
        li.append(frame)


    #define a dataframe:
    df = pd.concat(li, axis=0, ignore_index=True)

    #to extract the starting time from the "Time (CET) column of the dataframe"
    df[['Start time','end time']]=df['Time (CET)'].str.split('-',expand=True)
```

```
    df.drop(columns=['Time (CET)', 'Day-ahead Total Load Forecast [MW] - Germany (DE)'
, 'end time'], inplace=True)

    df=df[['Start time', 'Actual Total Load [MW] - Germany (DE)']]

    df['Start time']=pd.to_datetime(df['Start time'] )


    df=df.set_index('Start time')

    df['Actual Total Load [MW] - Germany (DE)']=df['Actual Total Load [MW] - Germany (
DE)'].resample('H').mean()

    df.dropna(inplace=True)

    return df

def HourlyLoadProfile_Italy(path):

    """
    Generates the updated data-frame for both weekdays and weekends for Italy.
    """

    all_files = glob.glob(path + "/*.csv")

    li = []

    for filename in all_files:
        frame = pd.read_csv(filename, index_col=None, header=0)
        li.append(frame)


    df = pd.concat(li, axis=0, ignore_index=True)

    df[['Start time','end time']]=df['Time (CET)'].str.split('-',expand=True)


    df.drop(columns=['Time (CET)', 'Day-ahead Total Load Forecast [MW] - Italy (IT)',
'end time'], inplace=True)

    df=df[['Start time', 'Actual Total Load [MW] - Italy (IT)']]

    df['Start time']=pd.to_datetime(df['Start time'] )

    df.sort_values(by='Start time', inplace=True)

    df=df.set_index('Start time')


    return df

def HourlyLoadProfile_Spain(path):

    """
    Generates the updated data-frame for both weekdays and weekends for Spain.
    """
```

```
    all_files = glob.glob(path + "/*.csv")

    li = []

    for filename in all_files:
        frame = pd.read_csv(filename, index_col=None, header=0)
        li.append(frame)

    df = pd.concat(li, axis=0, ignore_index=True)

    df[['Start time','end time']]=df['Time (CET)'].str.split('-',expand=True)


    df.drop(columns=['Time (CET)', 'Day-ahead Total Load Forecast [MW] - Spain (ES)',
'end time'], inplace=True)

    df=df[['Start time', 'Actual Total Load [MW] - Spain (ES)']]

    df['Start time']=pd.to_datetime(df['Start time'] )

    df.sort_values(by='Start time', inplace=True)

    df=df.set_index('Start time')


    return df

def HourlyLoadProfile_France(path):

    """
    Generates the updated data-frame for both weekdays and weekends for France.
    """


    all_files = glob.glob(path + "/*.csv")

    li = []

    for filename in all_files:
        frame = pd.read_csv(filename, index_col=None, header=0)
        li.append(frame)


    df = pd.concat(li, axis=0, ignore_index=True)

    df[['Start time','end time']]=df['Time (CET)'].str.split('-',expand=True)


    df.drop(columns=['Time (CET)', 'Day-ahead Total Load Forecast [MW] - France (FR)',
'end time'], inplace=True)

    df=df[['Start time', 'Actual Total Load [MW] - France (FR)']]

    df['Start time']=pd.to_datetime(df['Start time'] )
```

```
    df.sort_values(by='Start time', inplace=True)

    df=df.set_index('Start time')


    return df
```

## Step 2: Passing pathname of folders, running functions and assigning data frames

### Glossary:

1. The variables beginning with "path" are assigned the pathname of the folders that contain the individual data files for each phase per country. These are passed as arguments to the functions defined above.
2. The variables beginning with "weekday" and "weekend" followed by the phase and country-name are data-frames. They contain the data-frame returned upon calling the corresponding function for each phase and country.

```
path_ref1_germany = r'../germany/ref_wfh+sld'
weekday_ref1_germany=HourlyLoadProfile_Germany(path_ref1_germany)

path_wfh_germany = r'../germany/WFH'
weekday_wfh_germany=HourlyLoadProfile_Germany(path_wfh_germany)

path_sld_germany = r'../germany/softLockdown'
weekday_sld_germany=HourlyLoadProfile_Germany(path_sld_germany)

path_ref2_germany = r'../germany/ref_fld'
weekday_ref2_germany=HourlyLoadProfile_Germany(path_ref2_germany)

path_fld_germany = r'../germany/fullLockdown'
weekday_fld_germany=HourlyLoadProfile_Germany(path_fld_germany)
```

```
path_ref1_wknd_germany = r'../germany/ref_wfh'
weekend_ref1_germany=HourlyLoadProfile_Germany(path_ref1_wknd_germany)

path_wfh_wknd_germany = r'../germany/WFH'
weekend_wfh_germany=HourlyLoadProfile_Germany(path_wfh_wknd_germany)

path_ref2_wknd_germany = r'../germany/ref_sld'
weekend_ref2_germany=HourlyLoadProfile_Germany(path_ref2_wknd_germany)

path_sld_wknd_germany = r'../germany/softLockdown'
weekend_sld_germany=HourlyLoadProfile_Germany(path_sld_wknd_germany)

path_ref3_wknd_germany = r'../germany/ref_fld'
weekend_ref3_germany=HourlyLoadProfile_Germany(path_ref3_wknd_germany)

path_fld_wknd_germany = r'../germany/fullLockdown'
weekend_fld_germany=HourlyLoadProfile_Germany(path_fld_wknd_germany)
```

```
path_ref1_italy = r'../italy/ref_wfh+sld'
weekday_ref1_italy=HourlyLoadProfile_italy(path_ref1_italy)

path_wfh_italy = r'../italy/WFH'
weekday_wfh_italy=HourlyLoadProfile_italy(path_wfh_italy)

path_sld_italy = r'../italy/softLockdown'
weekday_sld_italy=HourlyLoadProfile_italy(path_sld_italy)

path_ref2_italy = r'../italy/ref_fld'
weekday_ref2_italy=HourlyLoadProfile_italy(path_ref2_italy)

path_fld_italy = r'../italy/fullLockdown'
weekday_fld_italy=HourlyLoadProfile_italy(path_fld_italy)



path_ref1_wknd_italy = r'../italy/ref_wfh'
weekend_ref1_italy=HourlyLoadProfile_italy(path_ref1_wknd_italy)

path_wfh_wknd_italy = r'../italy/WFH'
weekend_wfh_italy=HourlyLoadProfile_italy(path_wfh_wknd_italy)

path_ref2_wknd_italy = r'../italy/ref_sld'
weekend_ref2_italy=HourlyLoadProfile_italy(path_ref2_wknd_italy)

path_sld_wknd_italy = r'../italy/softLockdown'
weekend_sld_italy=HourlyLoadProfile_italy(path_sld_wknd_italy)

path_ref3_wknd_italy = r'../italy/ref_fld'
weekend_ref3_italy=HourlyLoadProfile_italy(path_ref3_wknd_italy)

path_fld_wknd_italy = r'../italy/fullLockdown'
weekend_fld_italy=HourlyLoadProfile_italy(path_fld_wknd_italy)



path_ref1_spain = r'../spain/ref_wfh'
weekday_ref1_spain=HourlyLoadProfile_spain(path_ref1_spain)

path_wfh_spain = r'../spain/WFH'
weekday_wfh_spain=HourlyLoadProfile_spain(path_wfh_spain)

path_ref2_spain = r'../spain/ref_sld'
weekday_ref2_spain=HourlyLoadProfile_spain(path_ref2_spain)

path_sld_spain = r'../spain/softLockdown'
weekday_sld_spain=HourlyLoadProfile_spain(path_sld_spain)

path_ref3_spain = r'../spain/ref_fld'
```

```
weekday_ref3_spain=HourlyLoadProfile_spain(path_ref3_spain)

path_fld_spain = r'../spain/fullLockdown'
weekday_fld_spain=HourlyLoadProfile_spain(path_fld_spain)


path_ref1_wknd_spain = r'../spain/ref_wfh'
weekend_ref1_spain=HourlyLoadProfile_spain(path_ref1_wknd_spain)

path_wfh_wknd_spain = r'../spain/WFH'
weekend_wfh_spain=HourlyLoadProfile_spain(path_wfh_wknd_spain)

path_ref2_wknd_spain = r'../spain/ref_sld'
weekend_ref2_spain=HourlyLoadProfile_spain(path_ref2_wknd_spain)

path_sld_wknd_spain = r'../spain/softLockdown'
weekend_sld_spain=HourlyLoadProfile_spain(path_sld_wknd_spain)

path_ref3_wknd_spain = r'../spain/ref_fld'
weekend_ref3_spain=HourlyLoadProfile_spain(path_ref3_wknd_spain)

path_fld_wknd_spain = r'../spain/fullLockdown'
weekend_fld_spain=HourlyLoadProfile_spain(path_fld_wknd_spain)



path_ref1_france = r'../france/ref_wfh+sld'
weekday_ref1_france=HourlyLoadProfile_france(path_ref1_france)

path_wfh_france = r'../france/WFH'
weekday_wfh_france=HourlyLoadProfile_france(path_wfh_france)

path_sld_france = r'../france/softLockdown'
weekday_sld_france=HourlyLoadProfile_france(path_sld_france)

path_ref2_france = r'../france/ref_fld'
weekday_ref2_france=HourlyLoadProfile_france(path_ref2_france)

path_fld_france = r'../france/fullLockdown'
weekday_fld_france=HourlyLoadProfile_france(path_fld_france)




path_ref1_wknd_france = r'../france/ref_wfh'
weekend_ref1_france=HourlyLoadProfile_france(path_ref1_wknd_france)

path_wfh_wknd_france = r'../france/WFH'
weekend_wfh_france=HourlyLoadProfile_france(path_wfh_wknd_france)

path_ref2_wknd_france = r'../france/ref_sld'
weekend_ref2_france=HourlyLoadProfile_france(path_ref2_wknd_france)

path_sld_wknd_france = r'../france/softLockdown'
```

```
weekend_sld_france=HourlyLoadProfile_france(path_sld_wknd_france)

path_ref3_wknd_france = r'../france/ref_fld'
weekend_ref3_france=HourlyLoadProfile_france(path_ref3_wknd_france)

path_fld_wknd_france = r'../france/fullLockdown'
weekend_fld_france=HourlyLoadProfile_france(path_fld_wknd_france)
```

## Step 3: Creating the consolidated, representative data frames

```
"""
Creates a data-frame with 21 columns and 24 rows, to house daily load profiles for eac
h phase for each country,
along with the corresponding references, for weekdays
"""
df_weekday_profile=pd.DataFrame(data=np.arange(0,24,1))
df_weekday_profile=pd.concat([df_weekday_profile] * (21), axis=1, ignore_index=True)
pd.set_option('display.max_columns', 27)

"""
Creates a data-frame with 21 columns and 24 rows, to house daily load profiles for eac
h phase for each country,
along with the corresponding references, for weekends
"""
df_weekend_profile=pd.DataFrame(data=np.arange(0,24,1))
df_weekend_profile=pd.concat([df_weekend_profile] * (24), axis=1, ignore_index=True)
pd.set_option('display.max_columns', 27)

    """
    This code segment generates the consolidated representative dataframe which contai
ns the normalised load profiles
    of a representative day for each phase (WFH, SLD and FLD), for each country, for w
eekdays.
    """

    """
    The following loop runs across 24 hours and does the following in each iteration:
        i. Creates a cluster for each phase for each country by collecting the load at
every hour
        ii. Computes the medoid of each cluster as the argmin of the deviation from th
e median of the cluster.
            This medoid is inserted into the column corresponding to a country's phase
.

    """

for i in range (0,24):

    arrayg1=weekday_ref1_germany.loc[weekday_ref1_germany.index.hour==i]
    arrayg2=weekday_wfh_germany.loc[weekday_wfh_germany.index.hour==i]
    arrayg3=weekday_sld_germany.loc[weekday_sld_germany.index.hour==i]
    arrayg4=weekday_ref2_germany.loc[weekday_ref2_germany.index.hour==i]
    arrayg5=weekday_fld_germany.loc[weekday_fld_germany.index.hour==i]
```

```
    arrayi1=weekday_ref1_italy.loc[weekday_ref1_italy.index.hour==i]
    arrayi2=weekday_wfh_italy.loc[weekday_wfh_italy.index.hour==i]
    arrayi3=weekday_sld_italy.loc[weekday_sld_italy.index.hour==i]
    arrayi4=weekday_ref2_italy.loc[weekday_ref2_italy.index.hour==i]
    arrayi5=weekday_fld_italy.loc[weekday_fld_italy.index.hour==i]

    arrays1=weekday_ref1_spain.loc[weekday_ref1_spain.index.hour==i]
    arrays2=weekday_wfh_spain.loc[weekday_wfh_spain.index.hour==i]
    arrays3=weekday_ref2_spain.loc[weekday_ref2_spain.index.hour==i]
    arrays4=weekday_sld_spain.loc[weekday_sld_spain.index.hour==i]
    arrays5=weekday_ref3_spain.loc[weekday_ref3_spain.index.hour==i]
    arrays6=weekday_fld_spain.loc[weekday_fld_spain.index.hour==i]

    arrayf1=weekday_ref1_france.loc[weekday_ref1_france.index.hour==i]
    arrayf2=weekday_wfh_france.loc[weekday_wfh_france.index.hour==i]
    arrayf3=weekday_sld_france.loc[weekday_sld_france.index.hour==i]
    arrayf4=weekday_ref2_france.loc[weekday_ref2_france.index.hour==i]
    arrayf5=weekday_fld_france.loc[weekday_fld_france.index.hour==i]


    df_weekday_profile.iloc[i,0]=weekday_ref1_germany.loc[((arrayg1-arrayg1.median()).
abs()).idxmin(), 'Actual Total Load [MW] - Germany (DE)'].values
    df_weekday_profile.iloc[i,1]=weekday_wfh_germany.loc[((arrayg2-arrayg2.median()).a
bs()).idxmin(), 'Actual Total Load [MW] - Germany (DE)'].values
    df_weekday_profile.iloc[i,2]=weekday_sld_germany.loc[((arrayg3-arrayg3.median()).a
bs()).idxmin(), 'Actual Total Load [MW] - Germany (DE)'].values
    df_weekday_profile.iloc[i,3]=weekday_ref2_germany.loc[((arrayg4-arrayg4.median()).
abs()).idxmin(), 'Actual Total Load [MW] - Germany (DE)'].values
    df_weekday_profile.iloc[i,4]=weekday_fld_germany.loc[((arrayg5-arrayg5.median()).a
bs()).idxmin(), 'Actual Total Load [MW] - Germany (DE)'].values


    df_weekday_profile.iloc[i,5]=weekday_ref1_italy.loc[((arrayi1-arrayi1.median()).ab
s()).idxmin(), 'Actual Total Load [MW] - Italy (IT)'].values
    df_weekday_profile.iloc[i,6]=weekday_wfh_italy.loc[((arrayi2-arrayi2.median()).abs
()).idxmin(), 'Actual Total Load [MW] - Italy (IT)'].values
    df_weekday_profile.iloc[i,7]=weekday_sld_italyloc[((arrayi3-arrayi3.median()).abs(
)).idxmin(), 'Actual Total Load [MW] - Italy (IT)'].values
    df_weekday_profile.iloc[i,8]=weekday_ref2_italy.loc[((arrayi4-arrayi4.median()).ab
s()).idxmin(), 'Actual Total Load [MW] - Italy (IT)'].values
    df_weekday_profile.iloc[i,9]=weekday_fld_italy.loc[((arrayi5-arrayi5.median()).abs
()).idxmin(), 'Actual Total Load [MW] - Italy (IT)'].values


    df_weekday_profile.iloc[i,10]=weekday_ref1_spain.loc[((arrays1-arrays1.median()).a
bs()).idxmin(), 'Actual Total Load [MW] - Spain (ES)'].values
    df_weekday_profile.iloc[i,11]=weekday_wfh_spain.loc[((arrays2-arrays2.median()).ab
s()).idxmin(), 'Actual Total Load [MW] - Spain (ES)'].values
    df_weekday_profile.iloc[i,12]=weekday_ref2_spain.loc[((arrays3-arrays3.median()).a
bs()).idxmin(), 'Actual Total Load [MW] - Spain (ES)'].values
    df_weekday_profile.iloc[i,13]=weekday_sld_spain.loc[((arrays4-arrays4.median()).ab
s()).idxmin(), 'Actual Total Load [MW] - Spain (ES)'].values
    df_weekday_profile.iloc[i,14]=weekday_ref2_spain.loc[((arrays5-arrays5.median()).a
bs()).idxmin(), 'Actual Total Load [MW] - Spain (ES)'].values
```

```
    df_weekday_profile.iloc[i,15]=weekday_fld_spain.loc[((arrays6-arrays6.median()).abs()).idxmin(), 'Actual Total Load [MW] - Spain (ES)'].values


    df_weekday_profile.iloc[i,16]=weekday_ref1_france.loc[((arrayf1-arrayf1.median()).abs()).idxmin(), 'Actual Total Load [MW] - France (FR)'].values
    df_weekday_profile.iloc[i,17]=weekday_wfh_france.loc[((arrayf2-arrayf2.median()).abs()).idxmin(), 'Actual Total Load [MW] - France (FR)'].values
    df_weekday_profile.iloc[i,18]=weekday_sld_france.loc[((arrayf3-arrayf3.median()).abs()).idxmin(), 'Actual Total Load [MW] - France (FR)'].values
    df_weekday_profile.iloc[i,19]=weekday_ref2_france.loc[((arrayf4-arrayf4.median()).abs()).idxmin(), 'Actual Total Load [MW] - France (FR)'].values
    df_weekday_profile.iloc[i,20]=weekday_fld_france.loc[((arrayf5-arrayf5.median()).abs()).idxmin(), 'Actual Total Load [MW] - France (FR)'].values



df_weekday_profile.rename(columns={0:'DE_Ref1', 1: 'DE_WFH', 2: 'DE_SLD', 3:'DE_Ref2', 4:'DE_FLD',
                           5:'IT_Ref1', 6: 'IT_WFH', 7: 'IT_SLD', 8: 'IT_Ref2', 9: 'IT_FLD',
                           10:'ES_Ref1', 11:'ES_WFH', 12: 'ES_Ref2', 13: 'ES_SLD', 14: 'ES_Ref3', 15: 'ES_FLD',
                           16:'FR_Ref1', 17: 'FR_WFH', 18: 'FR_SLD', 19: 'FR_Ref2', 20: 'FR_FLD'},inplace=True)


"""
    The following code normalises each medoid profile to the maximum of its corresponding reference profile.
    The resulting data-frame contains the normalised weekday load profiles for each phase for each country.
"""

gref_wfh_max=df_weekday_profile['DE_Ref1'].max()
df_weekday_profile['DE_Ref1']=df_weekday_profile['DE_Ref1']/gref_wfh_max
df_weekday_profile['DE_WFH']=df_weekday_profile['DE_WFH']/ gref_wfh_max
df_weekday_profile['DE_SLD']=df_weekday_profile['DE_SLD']/gref_wfh_max
gref_sld_max=df_weekday_profile['DE_Ref2'].max()
df_weekday_profile['DE_Ref2']=df_weekday_profile['DE_Ref2']/ gref_sld_max
df_weekday_profile['DE_FLD']=df_weekday_profile['DE_FLD']/gref_sld_max

iref_wfh_max=df_weekday_profile['IT_Ref1'].max()
df_weekday_profile['IT_Ref1']=df_weekday_profile['IT_Ref1']/iref_wfh_max
df_weekday_profile['IT_WFH']=df_weekday_profile['IT_WFH']/ iref_wfh_max
df_weekday_profile['IT_SLD']=df_weekday_profile['IT_SLD']/iref_wfh_max
iref_sld_max=df_weekday_profile['IT_Ref2'].max()
df_weekday_profile['IT_Ref2']=df_weekday_profile['IT_Ref2']/ iref_sld_max
df_weekday_profile['IT_FLD']=df_weekday_profile['IT_FLD']/iref_sld_max

sref_wfh_max=df_weekday_profile['ES_Ref1'].max()
df_weekday_profile['ES_Ref1']=df_weekday_profile['ES_Ref1']/sref_wfh_max
df_weekday_profile['ES_WFH']=df_weekday_profile['ES_WFH']/ sref_wfh_max
sref_sld_max=df_weekday_profile['ES_Ref2'].max()
df_weekday_profile['ES_Ref2']=df_weekday_profile['ES_Ref2']/sref_sld_max
```

```
df_weekday_profile['ES_SLD']=df_weekday_profile['ES_SLD']/sref_sld_max
sref3_max=df_weekday_profile['ES_Ref3'].max()
df_weekday_profile['ES_Ref3']=df_weekday_profile['ES_Ref3']/ sref3_max
df_weekday_profile['ES_FLD']=df_weekday_profile['ES_FLD']/sref3_max

Fref_wfh_max=df_weekday_profile['FR_Ref1'].max()
df_weekday_profile['FR_Ref1']=df_weekday_profile['FR_Ref1']/Fref_wfh_max
df_weekday_profile['FR_WFH']=df_weekday_profile['FR_WFH']/ Fref_wfh_max
df_weekday_profile['FR_SLD']=df_weekday_profile['FR_SLD']/Fref_wfh_max
Fref_sld_max=df_weekday_profile['FR_Ref2'].max()
df_weekday_profile['FR_Ref2']=df_weekday_profile['FR_Ref2']/ Fref_sld_max
df_weekday_profile['FR_FLD']=df_weekday_profile['FR_FLD']/Fref_sld_max



"""
    The following code follows the exact procedure described above to generate the med
oid load profiles, for weekends.
"""

for i in range (0,24):
    arrayg1=weekend_ref1_germany.loc[weekend_ref1_germany.index.hour==i]
    arrayg2=weekend_wfh_germany.loc[weekend_wfh_germany.index.hour==i]
    arrayg3=weekend_ref2_germany.loc[weekend_ref2_germany.index.hour==i]
    arrayg4=weekend_sld_germany.loc[weekend_sld_germany.index.hour==i]
    arrayg5=weekend_ref3_germany.loc[weekend_ref3_germany.index.hour==i]
    arrayg6=weekend_fld_germany.loc[weekend_fld_germany.index.hour==i]

    arrayi1=weekend_ref1_italy.loc[weekend_ref1_italy.index.hour==i]
    arrayi2=weekend_wfh_italy.loc[weekend_wfh_italy.index.hour==i]
    arrayi3=weekend_ref2_italy.loc[weekend_ref2_italy.index.hour==i]
    arrayi4=weekend_sld_italy.loc[weekend_sld_italy.index.hour==i]
    arrayi5=weekend_ref3_italy.loc[weekend_ref3_italy.index.hour==i]
    arrayi6=weekend_fld_italy.loc[weekend_fld_italy.index.hour==i]

    arrays1=weekend_ref1_spain.loc[weekend_ref1_spain.index.hour==i]
    arrays2=weekend_wfh_spain.loc[weekend_wfh_spain.index.hour==i]
    arrays3=weekend_ref2_spain.loc[weekend_ref2_spain.index.hour==i]
    arrays4=weekend_sld_spain.loc[weekend_sld_spain.index.hour==i]
    arrays5=weekend_ref3_spain.loc[weekend_ref3_spain.index.hour==i]
    arrays6=weekend_fld_spain.loc[weekend_fld_spain.index.hour==i]

    arrayf1=weekend_ref1_france.loc[weekend_ref1_france.index.hour==i]
    arrayf2=weekend_wfh_france.loc[weekend_wfh_france.index.hour==i]
    arrayf3=weekend_ref2_france.loc[weekend_ref2_france.index.hour==i]
    arrayf4=weekend_sld_france.loc[weekend_sld_france.index.hour==i]
    arrayf5=weekend_ref3_france.loc[weekend_ref3_france.index.hour==i]
    arrayf6=weekend_fld_france.loc[weekend_fld_france.index.hour==i]


    df_weekend_profile.iloc[i,0]=weekend_ref1_germany.loc[((arrays1-arrays1.median()).
abs()).idxmin(), 'Actual Total Load [MW] - Germany (DE)'].values
    df_weekend_profile.iloc[i,1]=weekend_wfh_germany.loc[((arrays2-arrays2.median()).a
bs()).idxmin(), 'Actual Total Load [MW] - Germany (DE)'].values
    df_weekend_profile.iloc[i,2]=weekend_ref2_germany.loc[((arrays3-arrays3.median()).
```

```
abs()).idxmin(), 'Actual Total Load [MW] - Germany (DE)'].values
    df_weekend_profile.iloc[i,3]=weekend_sld_germany.loc[((arrays4-arrays4.median()).a
bs()).idxmin(), 'Actual Total Load [MW] - Germany (DE)'].values
    df_weekend_profile.iloc[i,4]=weekend_ref2_germany.loc[((arrays5-arrays5.median()).
abs()).idxmin(), 'Actual Total Load [MW] - Germany (DE)'].values
    df_weekend_profile.iloc[i,5]=weekend_fld_germany.loc[((arrays6-arrays6.median()).a
bs()).idxmin(), 'Actual Total Load [MW] - Germany (DE)'].values


    df_weekend_profile.iloc[i,6]=weekend_ref1_italy.loc[((arrays1-arrays1.median()).ab
s()).idxmin(), 'Actual Total Load [MW] - Italy (IT)'].values
    df_weekend_profile.iloc[i,7]=weekend_wfh_italy.loc[((arrays2-arrays2.median()).abs
()).idxmin(), 'Actual Total Load [MW] - Italy (IT)'].values
    df_weekend_profile.iloc[i,8]=weekend_ref2_italy.loc[((arrays3-arrays3.median()).ab
s()).idxmin(), 'Actual Total Load [MW] - Italy (IT)'].values
    df_weekend_profile.iloc[i,9]=weekend_sld_italy.loc[((arrays4-arrays4.median()).abs
()).idxmin(), 'Actual Total Load [MW] - Italy (IT)'].values
    df_weekend_profile.iloc[i,10]=weekend_ref2_italy.loc[((arrays5-arrays5.median()).a
bs()).idxmin(), 'Actual Total Load [MW] - Italy (IT)'].values
    df_weekend_profile.iloc[i,11]=weekend_fld_italy.loc[((arrays6-arrays6.median()).ab
s()).idxmin(), 'Actual Total Load [MW] - Italy (IT)'].values


    df_weekend_profile.iloc[i,12]=weekend_ref1_spain.loc[((arrays1-arrays1.median()).a
bs()).idxmin(), 'Actual Total Load [MW] - Spain (ES)'].values
    df_weekend_profile.iloc[i,13]=weekend_wfh_spain.loc[((arrays2-arrays2.median()).ab
s()).idxmin(), 'Actual Total Load [MW] - Spain (ES)'].values
    df_weekend_profile.iloc[i,14]=weekend_ref2_spain.loc[((arrays3-arrays3.median()).a
bs()).idxmin(), 'Actual Total Load [MW] - Spain (ES)'].values
    df_weekend_profile.iloc[i,15]=weekend_sld_spain.loc[((arrays4-arrays4.median()).ab
s()).idxmin(), 'Actual Total Load [MW] - Spain (ES)'].values
    df_weekend_profile.iloc[i,16]=weekend_ref2_spain.loc[((arrays5-arrays5.median()).a
bs()).idxmin(), 'Actual Total Load [MW] - Spain (ES)'].values
    df_weekend_profile.iloc[i,17]=weekend_fld_spain.loc[((arrays6-arrays6.median()).ab
s()).idxmin(), 'Actual Total Load [MW] - Spain (ES)'].values


    df_weekend_profile.iloc[i,18]=weekend_ref1_france.loc[((arrays1-arrays1.median()).
abs()).idxmin(), 'Actual Total Load [MW] - France (FR)'].values
    df_weekend_profile.iloc[i,19]=weekend_wfh_france.loc[((arrays2-arrays2.median()).a
bs()).idxmin(), 'Actual Total Load [MW] - France (FR)'].values
    df_weekend_profile.iloc[i,20]=weekend_ref2_france.loc[((arrays3-arrays3.median()).
abs()).idxmin(), 'Actual Total Load [MW] - France (FR)'].values
    df_weekend_profile.iloc[i,21]=weekend_sld_france.loc[((arrays4-arrays4.median()).a
bs()).idxmin(), 'Actual Total Load [MW] - France (FR)'].values
    df_weekend_profile.iloc[i,22]=weekend_ref2_france.loc[((arrays5-arrays5.median()).
abs()).idxmin(), 'Actual Total Load [MW] - France (FR)'].values
    df_weekend_profile.iloc[i,23]=weekend_fld_france.loc[((arrays6-arrays6.median()).a
bs()).idxmin(), 'Actual Total Load [MW] - France (FR)'].values



df_weekend_profile.rename(columns={0:'DE_Ref1', 1: 'DE_WFH', 2: 'DE_Ref2', 3:'DE_SLD',
4:'DE_Ref3',
                          5:'DE_FLD', 6: 'IT_Ref1', 7: 'IT_WFH', 8: 'IT_Ref2', 9: 'IT_SL
```

```
D',
                        10:'IT_Ref3', 11:'IT_FLD', 12: 'ES_Ref1', 13: 'ES_WFH', 14: 'E
S_Ref2', 15: 'ES_SLD',
                        16:'ES_Ref3', 17: 'ES_FLD', 18: 'FR_Ref1', 19: 'FR_WFH', 20: '
FR_Ref2', 21:'FR_SLD',
                         22: 'FR_Ref3', 23: 'FR_FLD'},inplace=True)

gref_wfh_max=df_weekend_profile['DE_Ref1'].max()
df_weekend_profile['DE_Ref1']=df_weekend_profile['DE_Ref1']/gref_wfh_max
df_weekend_profile['DE_WFH']=df_weekend_profile['DE_WFH']/ gref_wfh_max
gref_sld_max=df_weekend_profile['DE_Ref2'].max()
df_weekend_profile['DE_Ref2']=df_weekend_profile['DE_Ref2']/ gref_sld_max
df_weekend_profile['DE_SLD']=df_weekend_profile['DE_SLD']/gref_sld_max
gref3_max=df_weekend_profile['DE_Ref3'].max()
df_weekend_profile['DE_Ref3']=df_weekend_profile['DE_Ref3']/ gref3_max
df_weekend_profile['DE_FLD']=df_weekend_profile['DE_FLD']/gref3_max

iref_wfh_max=df_weekend_profile['IT_Ref1'].max()
df_weekend_profile['IT_Ref1']=df_weekend_profile['IT_Ref1']/iref_wfh_max
df_weekend_profile['IT_WFH']=df_weekend_profile['IT_WFH']/ iref_wfh_max
iref_sld_max=df_weekend_profile['IT_Ref2'].max()
df_weekend_profile['IT_Ref2']=df_weekend_profile['IT_Ref2']/ iref_sld_max
df_weekend_profile['IT_SLD']=df_weekend_profile['IT_SLD']/iref_sld_max
iref3_max=df_weekend_profile['IT_Ref3'].max()
df_weekend_profile['IT_Ref3']=df_weekend_profile['IT_Ref3']/ iref3_max
df_weekend_profile['IT_FLD']=df_weekend_profile['IT_FLD']/iref3_max

sref_wfh_max=df_weekend_profile['ES_Ref1'].max()
df_weekend_profile['ES_Ref1']=df_weekend_profile['ES_Ref1']/sref_wfh_max
df_weekend_profile['ES_WFH']=df_weekend_profile['ES_WFH']/ sref_wfh_max
sref_sld_max=df_weekend_profile['ES_Ref2'].max()
df_weekend_profile['ES_Ref2']=df_weekend_profile['ES_Ref2']/sref_sld_max
df_weekend_profile['ES_SLD']=df_weekend_profile['ES_SLD']/sref_sld_max
sref3_max=df_weekend_profile['ES_Ref3'].max()
df_weekend_profile['ES_Ref3']=df_weekend_profile['ES_Ref3']/ sref3_max
df_weekend_profile['ES_FLD']=df_weekend_profile['ES_FLD']/sref3_max

Fref_wfh_max=df_weekend_profile['FR_Ref1'].max()
df_weekend_profile['FR_Ref1']=df_weekend_profile['FR_Ref1']/Fref_wfh_max
df_weekend_profile['FR_WFH']=df_weekend_profile['FR_WFH']/ Fref_wfh_max
Fref_sld_max=df_weekend_profile['FR_Ref2'].max()
df_weekend_profile['FR_Ref2']=df_weekend_profile['FR_Ref2']/ Fref_sld_max
df_weekend_profile['FR_SLD']=df_weekend_profile['FR_SLD']/Fref_sld_max
FRef3_max=df_weekend_profile['FR_Ref3'].max()
df_weekend_profile['FR_Ref3']=df_weekend_profile['FR_Ref3']/ FRef3_max
df_weekend_profile['FR_FLD']=df_weekend_profile['FR_FLD']/FRef3_max
```

## Step 4: Creating the percentage-decrease data frames

```
"""
    To generate the percentage-decrease data-frame, for weekdays.
```

```
    Each column of this data-frame contains the change in load of each phase,
    with respect to its corresponding reference phase. The change is expressed in perc
entage.
"""


df2=df_weekday_profile


df2['DE_WFH']=(df2['DE_WFH']-df2['DE_Ref1'])/df2['DE_Ref1']*100
df2['IT_WFH']=(df2['IT_WFH']-df2['IT_Ref1'])/df2['IT_Ref1']*100
df2['ES_WFH']=(df2['ES_WFH']-df2['ES_Ref1'])/df2['ES_Ref1']*100
df2['FR_WFH']=(df2['FR_WFH']-df2['FR_Ref1'])/df2['FR_Ref1']*100

df2['DE_SLD']=(df2['DE_SLD']-df2['DE_Ref1'])/df2['DE_Ref1']*100
df2['IT_SLD']=(df2['IT_SLD']-df2['IT_Ref1'])/df2['IT_Ref1']*100
df2['ES_SLD']=(df2['ES_SLD']-df2['ES_Ref2'])/df2['ES_Ref2']*100
df2['FR_SLD']=(df2['FR_SLD']-df2['FR_Ref1'])/df2['FR_Ref1']*100

df2['DE_FLD']=(df2['DE_FLD']-df2['DE_Ref2'])/df2['DE_Ref2']*100
df2['IT_FLD']=(df2['IT_FLD']-df2['IT_Ref2'])/df2['IT_Ref2']*100
df2['ES_FLD']=(df2['ES_FLD']-df2['ES_Ref3'])/df2['ES_Ref3']*100
df2['FR_FLD']=(df2['FR_FLD']-df2['FR_Ref2'])/df2['FR_Ref2']*100

df2.drop(columns=['DE_Ref1', 'DE_Ref2', 'IT_Ref1', 'IT_Ref2', 'ES_Ref1',
                  'ES_Ref2', 'ES_Ref3', 'FR_Ref1', 'FR_Ref2'], inplace=True)


#To generate the percentage-decrease data-frame, for weekends:

wdf2=df_weekend_profile



wdf2['DE_WFH']=(wdf2['DE_WFH']-wdf2['DE_Ref1'])/wdf2['DE_Ref1']*100
wdf2['IT_WFH']=(wdf2['IT_WFH']-wdf2['IT_Ref1'])/wdf2['IT_Ref1']*100
wdf2['ES_WFH']=(wdf2['ES_WFH']-wdf2['ES_Ref1'])/wdf2['ES_Ref1']*100
wdf2['FR_WFH']=(wdf2['FR_WFH']-wdf2['FR_Ref1'])/wdf2['FR_Ref1']*100

wdf2['DE_SLD']=(wdf2['DE_SLD']-wdf2['DE_Ref2'])/wdf2['DE_Ref2']*100
wdf2['IT_SLD']=(wdf2['IT_SLD']-wdf2['IT_Ref2'])/wdf2['IT_Ref2']*100
wdf2['ES_SLD']=(wdf2['ES_SLD']-wdf2['ES_Ref2'])/wdf2['ES_Ref2']*100
wdf2['FR_SLD']=(wdf2['FR_SLD']-wdf2['FR_Ref2'])/wdf2['FR_Ref2']*100

wdf2['DE_FLD']=(wdf2['DE_FLD']-wdf2['DE_Ref3'])/wdf2['DE_Ref3']*100
wdf2['IT_FLD']=(wdf2['IT_FLD']-wdf2['IT_Ref3'])/wdf2['IT_Ref3']*100
wdf2['ES_FLD']=(wdf2['ES_FLD']-wdf2['ES_Ref3'])/wdf2['ES_Ref3']*100
wdf2['FR_FLD']=(wdf2['FR_FLD']-wdf2['FR_Ref3'])/wdf2['FR_Ref3']*100

wdf2.drop(columns=['DE_Ref1', 'DE_Ref2', 'DE_Ref3','IT_Ref1', 'IT_Ref2', 'IT_Red3','ES
_Ref1','ES_Ref2', 'ES_Ref3', 'FR_Ref1', 'FR_Ref2', 'FR_Ref3'], inplace=True)
```