Annexure B – Python code for AHP using AHPy

```
import ahpy
import numpy
import scipy
1.# technical=Technical Barriers(TB)
#TB1= Technical Barrier 1 =  Lack of Sufficient number of Cross Borders
Interconnections
#TB2= Technical Barrier 2 =  Rising domestic generation (including solar
power) in India
#TB3= Technical Barrier 3 =  Lack of generation capacity for fulfilling
domestic demand in dry season
#TB4= Technical Barrier 4=   Lack of Grid Code Synchronization between
Nepal and its neighbours
2.# financial = Financial Barriers (FB)
#FB1= Financial Barrier 1= Relatively higher cost of hydro energy
#FB2= Financial Barrier 2= Need of Huge investment for construction of
cross border interconnection
#FB3=Declining cost of Renewable energy (Especially Solar Power) in India
3.#policy=Policy Barrier(PB)
#PB1=No Provision of Cross Border ET in Electricity Act 1992
#PB2=Lack of private sector involvement in CBET
#PB3=Lack of Open & non-discriminatory transmission Grid access for CBET;
#PB4=Absence of regional mechanisms (market modality) for cross-border
electricity trade;
#PB5=Lack of Domestic Power sector reforms; (PB5)
#PB6=Ambiguous policies related to CBET issued by India to control trading
in the region and threat of similar policies in future; (PB6)
#PB7=No separate supranational institution/entity responsible for CBET
#PB8=Lack of regulatory harmonization
3.#geopolitical=Socio and Geo-Political Barriers (SGB)
#SGB1= Internal Pressure of prioritization of domestic consumptions over
export;
#SGB2= Lack of continuity in political support for hydro project
development and weak political capacity to facilitate regional electricity
cooperation
#SGB3 = Energy security concerns and trust deficit issues
#SGB4 = Electricity transit facilities via India’s grid to export power
from Nepal to Bangladesh
5. #Barriers = Technical , Policy, Financial, Socio and Geopolitical
Barriers

5.0

technical_comparisons = {('TB1','TB2') : 8,('TB1','TB3'): 1, (
'TB1','TB4'): 1, ('TB2','TB3') : 1/5, ('TB2','TB4') : 1/6,  ('TB3','TB4')
:2 }
policy_comparisons ={('PB1','PB2') : 1/7,('PB1','PB3'): 1/7, (
'PB1','PB4'): 1/6, ('PB1','PB5') : 1/6 , ('PB1','PB6') : 1/8,
('PB1','PB7') :1, ('PB1','PB8') : 1/8 ,( 'PB2','PB3') : 1, ('PB2' , 'PB4')
: 1 , ( 'PB2' , 'PB5') : 1/5 , ('PB2 ', 'PB6') : 1/6 , ( 'PB2' , 'PB7') :
1/2 ,( 'PB2', 'PB8') : 1/4 , ( 'PB3','PB4') : 2,
('PB3' , 'PB5') : 2 , ( 'PB3', 'PB6') :  1/5 , ( 'PB3' , 'PB7') : 1/3
,('PB3','PB8') : 1/3 , ('PB4', 'PB5') : 1 , ( 'PB4' , 'PB6') : 1/5, (
```

```
'PB4', 'PB7') : 1 , ( 'PB4' ,'PB8') :6 , ( 'PB5' , 'PB6') : 1/4,
('PB5','PB7') : 1/2, ('PB5' ,'PB8') : 1 , ( 'PB6', 'PB7') : 8, (
'PB7','PB8') : 1/5 }
financial_comparisons = {('FB1', 'FB2') :2 , ('FB1' , 'FB3') : 1 , ('FB2'
, 'FB3' ) : 1/5 }
geopolitical_comparisons = {('SGB1','SGB2') : 1 , ('SGB1', 'SGB3') : 3, (
'SGB1' , 'SGB4') : 1/7, ( 'SGB2', 'SGB3') : 1,
                            ( 'SGB2' ,'SGB4') : 1/7, ('SGB3','SGB4') : 1/5
}
barriers_comparisons = {('technical' ,'financial') : 1, ( 'technical',
'policy') : 1/6 , ( 'technical', 'geopolitical') : 1,
                       ('policy','geopolitical') :6, ( 'financial' ,
'policy' ) : 1/5 ,( 'financial','geopolitical') : 3 }
technical =
ahpy.Compare('technical',technical_comparisons,precision=3,random_index='s
aaty')
policy =
ahpy.Compare('policy',policy_comparisons,precision=3,random_index='saaty')
financial =
ahpy.Compare('financial',financial_comparisons,precision=3,random_index='s
aaty')
geopolitical =
ahpy.Compare('geopolitical',geopolitical_comparisons,precision=3,random_in
dex='saaty')
barriers = ahpy.Compare ('barriers',barriers_comparisons,precision=3,
random_index='saaty')

print(technical.local_weights)

{'TB3': 0.357, 'TB1': 0.33, 'TB4': 0.262, 'TB2': 0.05}

print(technical.consistency_ratio)

0.037

print(financial.local_weights)

{'FB3': 0.498, 'FB1': 0.367, 'FB2': 0.135}

print(financial.consistency_ratio)

0.09

print(policy.local_weights)

{'PB6': 0.341, 'PB4': 0.136, 'PB8': 0.135, 'PB5': 0.091, 'PB3': 0.085,
'PB7': 0.082, 'PB2': 0.055, 'PB2 ': 0.053, 'PB1': 0.021}

print(policy.consistency_ratio)

0.184

print(geopolitical.local_weights)

{'SGB4': 0.668, 'SGB1': 0.143, 'SGB2': 0.101, 'SGB3': 0.088}

print(geopolitical.consistency_ratio)
```

```
0.082

print(barriers.global_weights)

{'policy': 0.639, 'financial': 0.16, 'technical': 0.113, 'geopolitical':
0.087}

print(barriers.consistency_ratio)

0.05

import pandas as pd
df = pd.DataFrame({'TB1': [1, 8, 1, 1],
                   'TB2': [1/8,1,1/5,1/6],
                   'TB3': [1,5,1,2],
                   'TB4': [1,6,1,1/2]})

df

   TB1       TB2  TB3  TB4
0    1  0.125000    1  1.0
1    8  1.000000    5  6.0
2    1  0.200000    1  1.0
3    1  0.166667    2  0.5

import pingouin as pg
import numpy as np

pg.cronbach_alpha(data=df)

(0.8972656142377471, array([0.478, 0.993]))

import pandas as pd
df = pd.DataFrame({'PB1': [1, 1/7, 1/7,1/6,1/6, 1/8,1,1],
                   'PB2': [7,1,1,1,1/5,1/6,1/2,1/4],
                   'PB3': [7,1,1,2,2,1/5,1,1/3],
                   'PB4': [6,1,1/2,1,1,1/5,1,6],
                   'PB5': [8,5,1/2,1,1,1/5,1,2],
                   'PB6': [6,6,5,5,5,1,8,8],
                   'PB7': [1,2,3,1,1,1/8,1,3],
                   'PB8': [8,4,3,1/6,1/6,1/8,5,1]})

df

        PB1       PB2       PB3  PB4  PB5  PB6    PB7       PB8
0  1.000000  7.000000  7.000000  6.0  8.0    6  1.000  8.000000
1  0.142857  1.000000  1.000000  1.0  5.0    6  2.000  4.000000
2  0.142857  1.000000  1.000000  0.5  0.5    5  3.000  3.000000
3  0.166667  1.000000  2.000000  1.0  1.0    5  1.000  0.166667
4  0.166667  0.200000  2.000000  1.0  1.0    5  1.000  0.166667
5  0.125000  0.166667  0.200000  0.2  0.2    1  0.125  0.125000
6  1.000000  0.500000  1.000000  1.0  1.0    8  1.000  5.000000
7  1.000000  0.250000  0.333333  6.0  2.0    8  3.000  1.000000
```

```
import pingouin as pg

pg.cronbach_alpha(data=df)

(0.8569023795830761, array([0.634, 0.967]))

import pandas as pd
df = pd.DataFrame({'FB1': [1, 2, 1],
                   'FB2': [1,1,1/5,],
                   'FB3': [1,5,1]})

df

   FB1  FB2  FB3
0    1  1.0    1
1    2  1.0    5
2    1  0.2    1

import pingouin as pg

pg.cronbach_alpha(data=df)

(0.8569023795830761, array([0.634, 0.967]))

import pandas as pd
df = pd.DataFrame({'SGB1': [1, 1, 3, 1/7],
                   'SGB2': [1,1,1,1/7],
                   'SGB3': [1/3,1,1,1/7],
                   'SGB4': [7,7,1/5,1]})


df

       SGB1      SGB2      SGB3  SGB4
0  1.000000  1.000000  0.333333   7.0
1  1.000000  1.000000  1.000000   7.0
2  3.000000  1.000000  1.000000   0.2
3  0.142857  0.142857  0.142857   1.0

import pingouin as pg

pg.cronbach_alpha(data=df)

(0.881737653281901878, array([-3.958,  0.933]))
```