

Assessing Data

A. Perform error management on the given data using pandas package.

i. Drop the Columns Where All Elements Are Missing Values

Code:-

```
#####

# -*- coding: utf-8 -*-

#####

import sys

import os

import pandas as pd

#####

if sys.platform == 'linux':

    Base=os.path.expanduser('~') + 'VKHCG'

else:

    Base='C:/VKHCG'

#####

print('#####')

print('Working Base :',Base, ' using ', sys.platform)

print('#####')

#####

sInputFileName='Good-or-Bad.csv'

sOutputFileName='Good-or-Bad-01.csv'

Company='01-Vermeulen'

#####

Base='C:/VKHCG'

#####

sFileDir=Base + '/' + Company + '/02-Assess/01-EDS/02-Python'

if not os.path.exists(sFileDir):

    os.makedirs(sFileDir)

#####

### Import Warehouse

#####

sFileName=Base + '/' + Company + '/00-RawData/' + sInputFileName

print('Loading :',sFileName)
```

```

RawData=pd.read_csv(sFileName,header=0)

print('#####')
print('## Raw Data Values')
print('#####')
print(RawData)
print('#####')
print('## Data Profile')
print('#####')
print('Rows :',RawData.shape[0])
print('Columns :',RawData.shape[1])
print('#####')
#####
sFileName=sFileDir + '/' + sInputFileName
RawData.to_csv(sFileName, index = False)
#####
TestData=RawData.dropna(axis=1, how='all')
#####
print('#####')
print('## Test Data Values')
print('#####')
print(TestData)
print('#####')
print('## Data Profile')
print('#####')
print('Rows :',TestData.shape[0])
print('Columns :',TestData.shape[1])
print('#####')
#####
sFileName=sFileDir + '/' + sOutputFileName
TestData.to_csv(sFileName, index = False)
#####
print('#####')
print('### Done!! #####')
print('#####')

```

Output:-

```
*Python 2.7.15 Shell*
File Edit Shell Debug Options Window Help
Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:30:26) [MSC v.1500 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: D:\DataSciencPratical\VKHCG\01-Vermeulen\02-Assess\Assess-Good-Bad-01.py
#####
('Working Base :', 'C:/VKHCG', ' using ', 'win32')
#####
('Loading :', 'C:/VKHCG/01-Vermeulen/00-RawData/Good-or-Bad.csv')
#####
## Raw Data Values
#####
   ID FieldA FieldB FieldC FieldD FieldE FieldF FieldG
0   1.0   Good   Better   Best  1024.0   NaN  10241.0     1
1   2.0   Good     NaN   Best   512.0   NaN   5121.0     2
2   3.0   Good   Better   NaN   256.0   NaN   256.0     3
3   4.0   Good   Better   Best    NaN   NaN   211.0     4
4   5.0   Good   Better   NaN    64.0   NaN   6411.0     5
5   6.0   Good     NaN   Best    32.0   NaN    32.0     6
6   7.0   NaN   Better   Best    16.0   NaN   1611.0     7
7   8.0   NaN     NaN   Best     8.0   NaN   8111.0     8
8   9.0   NaN     NaN   NaN     4.0   NaN    41.0     9
9  10.0     A     B     C     2.0   NaN  21111.0    10
10  NaN     NaN     NaN   NaN     NaN   NaN     NaN    11
11 10.0   Good   Better   Best  1024.0   NaN  102411.0   12
12 10.0   Good     NaN   Best   512.0   NaN   512.0    13
13 10.0   Good   Better   NaN   256.0   NaN  1256.0    14
14 10.0   Good   Better   Best    NaN   NaN     NaN    15
15 10.0   Good   Better   NaN    64.0   NaN   164.0    16
16 10.0   Good     NaN   Best    32.0   NaN   322.0    17
17 10.0   NaN   Better   Best    16.0   NaN   163.0    18
18 10.0   NaN     NaN   Best     8.0   NaN   844.0    19
19 10.0   NaN     NaN   NaN     4.0   NaN  4555.0    20
20 10.0     A     B     C     2.0   NaN   111.0    21
#####
## Data Profile
#####
('Rows :', 21)
('Columns :', 8)
#####
#####
## Test Data Values
#####
```

```
#####
      ID FieldA  FieldB FieldC FieldD      FieldF  FieldG
0      1.0   Good  Better   Best  1024.0    10241.0      1
1      2.0   Good    NaN   Best   512.0     5121.0      2
2      3.0   Good  Better    NaN   256.0     256.0      3
3      4.0   Good  Better   Best    NaN     211.0      4
4      5.0   Good  Better    NaN    64.0     6411.0      5
5      6.0   Good    NaN   Best    32.0      32.0      6
6      7.0   NaN  Better   Best    16.0     1611.0      7
7      8.0   NaN    NaN   Best     8.0     8111.0      8
8      9.0   NaN    NaN    NaN     4.0      41.0      9
9     10.0    A      B      C      2.0    21111.0     10
10    NaN    NaN    NaN    NaN    NaN      NaN      11
11    10.0   Good  Better   Best  1024.0   102411.0     12
12    10.0   Good    NaN   Best   512.0     512.0     13
13    10.0   Good  Better    NaN   256.0    1256.0     14
14    10.0   Good  Better   Best    NaN      NaN     15
15    10.0   Good  Better    NaN    64.0     164.0     16
16    10.0   Good    NaN   Best    32.0     322.0     17
17    10.0   NaN  Better   Best    16.0     163.0     18
18    10.0   NaN    NaN   Best     8.0     844.0     19
19    10.0   NaN    NaN    NaN     4.0    4555.0     20
20    10.0    A      B      C      2.0     111.0     21
#####
## Data Profile
#####
('Rows :', 21)
('Columns :', 7)
#####
#####
### Done!! #####
#####
>>>
```

B. Write Python / R program to create the network routing diagram from the given data

On routers.

Code:-

```
#####

import sys

import os

import pandas as pd

#####

pd.options.mode.chained_assignment = None

#####

if sys.platform == 'linux':

    Base=os.path.expanduser('~') + 'VKHCG'

else:

    Base='C:/VKHCG'

#####

print('#####')

print('Working Base :',Base, ' using ', sys.platform)
```

```

print('#####')

#####

sInputFileName1='01-Retrieve/01-EDS/01-R/Retrieve_Country_Code.csv'
sInputFileName2='01-Retrieve/01-EDS/02-Python/Retrieve_Router_Location.csv'
sInputFileName3='01-Retrieve/01-EDS/01-R/Retrieve_IP_DATA.csv'

#####

sOutputFileName='Assess-Network-Routing-Company.csv'
Company='01-Vermeulen'

#####

#####

### Import Country Data

#####

sFileName=Base + '/' + Company + '/' + sInputFileName1

print('#####')

print('Loading :',sFileName)

print('#####')

CountryData=pd.read_csv(sFileName,header=0,low_memory=False, encoding="latin-1")

print('Loaded Country:',CountryData.columns.values)

print('#####')

#####

## Assess Country Data

#####

print('#####')

print('Changed :',CountryData.columns.values)

CountryData.rename(columns={'Country': 'Country_Name'}, inplace=True)
CountryData.rename(columns={'ISO-2-CODE': 'Country_Code'}, inplace=True)
CountryData.drop('ISO-M49', axis=1, inplace=True)
CountryData.drop('ISO-3-Code', axis=1, inplace=True)
CountryData.drop('RowID', axis=1, inplace=True)

print('To :',CountryData.columns.values)

print('#####')

#####

#####

### Import Company Data

#####

```

```

sFileName=Base + '/' + Company + '/' + sInputFileName2
print('#####')
print('Loading :',sFileName)
print('#####')
CompanyData=pd.read_csv(sFileName,header=0,low_memory=False, encoding="latin-1")
print('Loaded Company :',CompanyData.columns.values)
print('#####')
#####
## Assess Company Data
#####
print('#####')
print('Changed :',CompanyData.columns.values)
CompanyData.rename(columns={'Country': 'Country_Code'}, inplace=True)
print('To :',CompanyData.columns.values)
print('#####')
#####
#####
### Import Customer Data
#####
sFileName=Base + '/' + Company + '/' + sInputFileName3
print('#####')
print('Loading :',sFileName)
print('#####')
CustomerRawData=pd.read_csv(sFileName,header=0,low_memory=False, encoding="latin-1")
print('#####')
print('Loaded Customer :',CustomerRawData.columns.values)
print('#####')
#####
CustomerData=CustomerRawData.dropna(axis=0, how='any')
print('#####')
print('Remove Blank Country Code')
print('Reduce Rows from', CustomerRawData.shape[0], 'to ', CustomerData.shape[0])
print('#####')
#####
print('#####')

```

```

print('Changed :',CustomerData.columns.values)

CustomerData.rename(columns={'Country': 'Country_Code'}, inplace=True)

print('To :',CustomerData.columns.values)

print('#####')

#####

print('#####')

print('Merge Company and Country Data')

print('#####')

CompanyNetworkData=pd.merge(

    CompanyData,

    CountryData,

    how='inner',

    on='Country_Code'

)

#####

print('#####')

print('Change ',CompanyNetworkData.columns.values)

for i in CompanyNetworkData.columns.values:

    j='Company_'+i

    CompanyNetworkData.rename(columns={i: j}, inplace=True)

print('To ', CompanyNetworkData.columns.values)

print('#####')

#####

#####

sFileDir=Base + '/' + Company + '/02-Assess/01-EDS/02-Python'

if not os.path.exists(sFileDir):

    os.makedirs(sFileDir)

#####

sFileName=sFileDir + '/' + sOutputFileName

print('#####')

print('Storing :', sFileName)

print('#####')

CompanyNetworkData.to_csv(sFileName, index = False, encoding="latin-1")

#####

#####

```

```
print('#####')

print('### Done!! #####')

print('#####')
```

output:-

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Company	Company	Company	Company	Company	Country	Country	Name						
2	US	New York	40.7528	-73.9725	United States of America									
3	US	New York	40.7214	-74.0052	United States of America									
4	US	New York	40.7662	-73.9862	United States of America									
5	US	New York	40.7449	-73.9782	United States of America									
6	US	New York	40.7605	-73.9933	United States of America									
7	US	New York	40.7588	-73.968	United States of America									
8	US	New York	40.7637	-73.9727	United States of America									
9	US	New York	40.7553	-73.9924	United States of America									
10	US	New York	40.7308	-73.9975	United States of America									
11	US	New York	40.7694	-73.9609	United States of America									
12	US	New York	40.733	-74.0078	United States of America									
13	US	New York	40.7505	-73.9931	United States of America									
14	US	New York	40.7517	-73.9972	United States of America									
15	US	New York	40.7082	-74.0132	United States of America									

Code:-

```
#####Assess-Network-Routing-Customer.py #####

import sys

import os

import pandas as pd

#####

pd.options.mode.chained_assignment = None

#####

Base='C:/VKHCG'

print('#####')

print('Working Base :',Base, ' using ', sys.platform)

print('#####')

#####

sInputFileName=Base+'/01-Vermeulen/02-Assess/01-EDS/02-Python/Assess-Network-Routing-
Customer.csv'

#####

sOutputFileName='Assess-Network-Routing-Customer.gml'
```


Company='01-Vermeulen'

#####

Import Country Data

#####

sFileName=sInputFileName

print('#####')

print('Loading :',sFileName)

print('#####')

CustomerData=pd.read_csv(sFileName,header=0,low_memory=False, encoding="latin-1")

print('Loaded Country:',CustomerData.columns.values)

print('#####')

print(CustomerData.head())

print('#####')

print('### Done!! #####')

print('#####')

#####

Output:-

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Customer_Country_Code	Customer_Place_Name	Customer_Latitude	Customer_Longitude	Customer_Country_Name								
2	BW	Gaborone	-24.6464	25.9119	Botswana								
3	BW	Francistown	-21.1667	27.5167	Botswana								
4	BW	Maun	-19.9833	23.4167	Botswana								
5	BW	Molepolole	-24.4167	25.5333	Botswana								
6	NE	Niamey	13.5167	2.1167	Niger								
7	MZ	Maputo	-25.9653	32.5892	Mozambique								
8	MZ	Tete	-16.1564	33.5867	Mozambique								
9	MZ	Quelimane	-17.8786	36.8883	Mozambique								
10	MZ	Chimoio	-19.1164	33.4833	Mozambique								
11	MZ	Matola	-25.9622	32.4589	Mozambique								
12	MZ	Pemba	-12.9608	40.5078	Mozambique								
13	MZ	Lichinga	-13.3128	35.2406	Mozambique								
14	MZ	Maxixe	-23.8597	35.3472	Mozambique								
15	MZ	Chibuto	-24.6867	33.5306	Mozambique								
16	MZ	Ressano Garcia	-25.4428	31.9953	Mozambique								
17	GH	Tema	5.6167	-0.0167	Ghana								
18	GH	Kumasi	6.6833	-1.6167	Ghana								
19	GH	Takoradi	4.8833	-1.75	Ghana								
20	GH	Accra	5.55	-0.2167	Ghana								
21	GH	Legon	5.65	-0.1833	Ghana								
22	GH	Sunyani	7.3333	-2.3333	Ghana								
23	GH	Cape Coast	5.1	-1.25	Ghana								

Code:-

Assess-Network-Routing-Node.py

#####

import sys

```

import os

import pandas as pd

#####

pd.options.mode.chained_assignment = None

#####

Base='C:/VKHCG'

#####

print('#####')

print('Working Base :',Base, ' using ', sys.platform)

print('#####')

#####

sInputFileName='01-Retrieve/01-EDS/02-Python/Retrieve_IP_DATA.csv'

#####

sOutputFileName='Assess-Network-Routing-Node.csv'

Company='01-Vermeulen'

#####

### Import IP Data

#####

sFileName=Base + '/' + Company + '/' + sInputFileName

print('#####')

print('Loading :',sFileName)

print('#####')

IPData=pd.read_csv(sFileName,header=0,low_memory=False, encoding="latin-1")

print('Loaded IP :', IPData.columns.values)

print('#####')

#####

print('#####')

print('Changed :',IPData.columns.values)

IPData.drop('RowID', axis=1, inplace=True)

IPData.drop('ID', axis=1, inplace=True)

IPData.rename(columns={'Country': 'Country_Code'}, inplace=True)

IPData.rename(columns={'Place.Name': 'Place_Name'}, inplace=True)

IPData.rename(columns={'Post.Code': 'Post_Code'}, inplace=True)

IPData.rename(columns={'First.IP.Number': 'First_IP_Number'}, inplace=True)

IPData.rename(columns={'Last.IP.Number': 'Last_IP_Number'}, inplace=True)

```

```

print('To :',IPData.columns.values)

print('#####')
#####

print('#####')

print('Change ',IPData.columns.values)

for i in IPData.columns.values:

j='Node_'+i

IPData.rename(columns={i: j}, inplace=True)

print('To ', IPData.columns.values)

print('#####')
#####

sFileDir=Base + '/' + Company + '/02-Assess/01-EDS/02-Python'

if not os.path.exists(sFileDir):

os.makedirs(sFileDir)

#####

sFileName=sFileDir + '/' + sOutputFileName

print('#####')

print('Storing :', sFileName)

print('#####')

IPData.to_csv(sFileName, index = False, encoding="latin-1")

#####

print('#####')

print('### Done!! #####')

print('#####')

```

Output:-

Assess-Network-Routing-Node.csv - Excel

	A	B	C	D	E	F	G	H	I	J	K	L
	Node_Country_Code	Node_Place_Name	Node_Post_Code	Node_Latitude	Node_Longitude	Node_First_IP_Number	Node_Last_IP_Number					
2	BW	Gaborone		-24.6464	25.9119	692781056	692781567					
3	BW	Gaborone		-24.6464	25.9119	692781824	692783103					
4	BW	Gaborone		-24.6464	25.9119	692909056	692909311					
5	BW	Gaborone		-24.6464	25.9119	692909568	692910079					
6	BW	Gaborone		-24.6464	25.9119	693051392	693052415					
7	BW	Gaborone		-24.6464	25.9119	693078272	693078527					
8	BW	Gaborone		-24.6464	25.9119	693608448	693616639					
9	BW	Gaborone		-24.6464	25.9119	696929792	696930047					
10	BW	Gaborone		-24.6464	25.9119	700438784	700439039					
11	BW	Gaborone		-24.6464	25.9119	702075904	702076927					
12	BW	Gaborone		-24.6464	25.9119	702498816	702499839					
13	BW	Gaborone		-24.6464	25.9119	702516224	702517247					
14	BW	Gaborone		-24.6464	25.9119	774162663	774162667					
15	BW	Gaborone		-24.6464	25.9119	1401887232	1401887743					
16	BW	Gaborone		-24.6464	25.9119	1754209024	1754209279					
17	NE	Niamey		13.5167	2.1167	696918528	696919039					
18	NE	Niamey		13.5167	2.1167	696922112	696924159					
19	NE	Niamey		13.5167	2.1167	701203456	701203711					
20	NE	Niamey		13.5167	2.1167	758886912	758887167					
21	NE	Niamey		13.5167	2.1167	1347294153	1347294160					
22	NE	Niamey		13.5167	2.1167	1755108096	1755108351					
23	NE	Niamey		13.5167	2.1167	1755828480	1755828735					

C. Write a Python / R program to build directed acyclic graph

Code:-

D. Write a Python / R program to pick the content for Bill Boards from the given data.

Code:-

```
#####
import sys
import os
import sqlite3 as sq
import pandas as pd
#####
if sys.platform == 'linux':
    Base=os.path.expanduser('~') + 'VKHCG'
else:
    Base='C:/VKHCG'
print('#####')
print('Working Base :',Base, ' using ', sys.platform)
print('#####')
#####
sInputFileName1='01-Retrieve/01-EDS/02-Python/Retrieve_DE_Billboard_Locations.csv'
sInputFileName2='01-Retrieve/01-EDS/02-Python/Retrieve_Online_Visitor.csv'
sOutputFileName='Assess-DE-Billboard-Visitor.csv'
Company='02-Krennwallner'
#####
sDataBaseDir=Base + '/' + Company + '/02-Assess/SQLite'
if not os.path.exists(sDataBaseDir):
    os.makedirs(sDataBaseDir)
#####
sDatabaseName=sDataBaseDir + '/krennwallner.db'
conn = sq.connect(sDatabaseName)
#####
### Import Billboard Data
```

```
#####
sFileName=Base + '/' + Company + '/' + sInputFileName1
print('#####')
print('Loading :',sFileName)
print('#####')
BillboardRawData=pd.read_csv(sFileName,header=0,low_memory=False, encoding="latin-1")
BillboardRawData.drop_duplicates(subset=None, keep='first', inplace=True)
BillboardData=BillboardRawData
print('Loaded Company :',BillboardData.columns.values)
print('#####')
#####
print('#####')
sTable='Assess_BillboardData'
print('Storing :',sDatabaseName,' Table:',sTable)
BillboardData.to_sql(sTable, conn, if_exists="replace")
print('#####')
#####
print(BillboardData.head())
print('#####')
print('Rows : ',BillboardData.shape[0])
print('#####')
#####
### Import Billboard Data
#####
sFileName=Base + '/' + Company + '/' + sInputFileName2
print('#####')
print('Loading :',sFileName)
print('#####')
VisitorRawData=pd.read_csv(sFileName,header=0,low_memory=False, encoding="latin-1")
VisitorRawData.drop_duplicates(subset=None, keep='first', inplace=True)
VisitorData=VisitorRawData[VisitorRawData.Country=='DE']
print('Loaded Company :',VisitorData.columns.values)
print('#####')
#####
print('#####')
sTable='Assess_VisitorData'
print('Storing :',sDatabaseName,' Table:',sTable)
VisitorData.to_sql(sTable, conn, if_exists="replace")
print('#####')
#####
print(VisitorData.head())
print('#####')
print('Rows : ',VisitorData.shape[0])
print('#####')
#####
print('#####')
sTable='Assess_BillboardVisitorData'
print('Loading :',sDatabaseName,' Table:',sTable)
sSQL="select distinct"
sSQL=sSQL+ " A.Country AS BillboardCountry,"
sSQL=sSQL+ " A.Place_Name AS BillboardPlaceName,"
sSQL=sSQL+ " A.Latitude AS BillboardLatitude, "
sSQL=sSQL+ " A.Longitude AS BillboardLongitude,"
sSQL=sSQL+ " B.Country AS VisitorCountry,"
```

```

sSQL=sSQL+ " B.Place_Name AS VisitorPlaceName,"
sSQL=sSQL+ " B.Latitude AS VisitorLatitude, "
sSQL=sSQL+ " B.Longitude AS VisitorLongitude,"
sSQL=sSQL+ " (B.Last_IP_Number - B.First_IP_Number) * 365.25 * 24 * 12 AS VisitorYearRate"
sSQL=sSQL+ " from"
sSQL=sSQL+ " Assess_BillboardData as A"
sSQL=sSQL+ " JOIN "
sSQL=sSQL+ " Assess_VisitorData as B"
sSQL=sSQL+ " ON "
sSQL=sSQL+ " A.Country = B.Country"
sSQL=sSQL+ " AND "
sSQL=sSQL+ " A.Place_Name = B.Place_Name;"
BillboardVistorsData=pd.read_sql_query(sSQL, conn)
print('#####')
#####
print('#####')
sTable='Assess_BillboardVistorsData'
print('Storing :',sDatabaseName,' Table:',sTable)
BillboardVistorsData.to_sql(sTable, conn, if_exists="replace")
print('#####')
#####
print(BillboardVistorsData.head())
print('#####')
print('Rows : ',BillboardVistorsData.shape[0])
print('#####')
#####
sFileDir=Base + '/' + Company + '/02-Assess/01-EDS/02-Python'
if not os.path.exists(sFileDir):
    os.makedirs(sFileDir)
#####
print('#####')
print('Storing :', sFileName)
print('#####')
sFileName=sFileDir + '/' + sOutputFileName
BillboardVistorsData.to_csv(sFileName, index = False)
print('#####')
#####
print('### Done!! #####')

```

Output:-

```

RESTART: D:\DataSciencPratical\VKHCG\02-Krennwallner\02-Assess\Assess-DE-Billboard.py
#####
('Working Base :', 'C:/VKHCG', ' using ', 'win32')
#####
('Loading :', 'C:/VKHCG/02-Krennwallner/01-Retrieve/01-EDS/02-Python/Retrieve_DE_Billboard_Locations.csv')
#####
('Loaded Company :', array([u'Country', u'Place_Name', u'Latitude', u'Longitude'], dtype=object))
#####
('Storing :', 'C:/VKHCG/02-Krennwallner/02-Assess/SQLite/krennwallner.db', ' Table:', 'Assess_BillboardData')
#####
      Country      Place_Name  Latitude  Longitude
0      DE      Lake      51.7833      8.5667
1      DE      Horb      48.4333      8.6833
2      DE      Hardenberg      51.1000      7.7333
3      DE      Horn-bad Meinberg      51.9833      8.9667
4      DE      Winkel      51.5500      13.3833
#####
('Rows : ', 8873)
#####
('Loading :', 'C:/VKHCG/02-Krennwallner/01-Retrieve/01-EDS/02-Python/Retrieve_Online_Visitor.csv')
#####
('Loaded Company :', array([u'Country', u'Place_Name', u'Latitude', u'Longitude',
      u'First_IP_Number', u'Last_IP_Number'], dtype=object))
#####
('Storing :', 'C:/VKHCG/02-Krennwallner/02-Assess/SQLite/krennwallner.db', ' Table:', 'Assess_VisitorData')
#####
      Country Place_Name  Latitude  Longitude  First_IP_Number  Last_IP_Number
368678      DE      Lake      51.7833      8.5667      1418011904      1418012159
368679      DE      Lake      51.7833      8.5667      1528171008      1528171263
368680      DE      Lake      51.7833      8.5667      1528177920      1528178175
408664      DE      Horb      48.4333      8.6833      777246720      777246975
408665      DE      Horb      48.4333      8.6833      1339562496      1339562751
#####
('Rows : ', 75999)
#####
('Loading :', 'C:/VKHCG/02-Krennwallner/02-Assess/SQLite/krennwallner.db', ' Table:', 'Assess_BillboardVisitorData')
#####
('Storing :', 'C:/VKHCG/02-Krennwallner/02-Assess/SQLite/krennwallner.db', ' Table:', 'Assess_BillboardVisitorsData')
#####
      BillboardCountry BillboardPlaceName  ...  VisitorLongitude  VisitorYearRate
0      DE      Lake  ...      8.5667      26823960.0
1      DE      Horb  ...      8.6833      26823960.0
2      DE      Horb  ...      8.6833      53753112.0
3      DE      Horb  ...      8.6833      107611416.0
4      DE      Horb  ...      8.6833      13359384.0

[5 rows x 9 columns]
#####
('Rows : ', 181235)
#####
('Storing :', 'C:/VKHCG/02-Krennwallner/01-Retrieve/01-EDS/02-Python/Retrieve_Online_Visitor.csv')
#####
#####
## Done!! #####
>>> |

```


	A	B	C	D	E	F
1	Country	Place_Name	Latitude	Longitude	First_IP_Number	Last_IP_Number
2	BW	Gaborone	-24.6464	25.9119	692781056	692781567
3	BW	Gaborone	-24.6464	25.9119	692781824	692783103
4	BW	Gaborone	-24.6464	25.9119	692909056	692909311
1048556	NL	Amsterdam	52.3556	4.9136	385939968	385940479
1048557	NL	Amsterdam	52.3556	4.9136	385942528	385943551
1048558	NL	Amsterdam	52.3556	4.9136	385957888	385961983
1048559	NL	Amsterdam	52.3556	4.9136	386003200	386003967
1048560	NL	Amsterdam	52.3556	4.9136	386012160	386012671
1048561	NL	Amsterdam	52.3556	4.9136	386013184	386013695
1048562	NL	Amsterdam	52.3556	4.9136	386015232	386015487
1048563	NL	Amsterdam	52.3556	4.9136	386020352	386021375
1048564	NL	Amsterdam	52.3556	4.9136	386035712	386039807
1048565	NL	Amsterdam	52.3556	4.9136	386060288	386068479
1048566	NL	Amsterdam	52.3556	4.9136	386073344	386073599
1048567	NL	Amsterdam	52.3556	4.9136	386074112	386074623
1048568	NL	Amsterdam	52.3556	4.9136	386076416	386076671
1048569	NL	Amsterdam	52.3556	4.9136	386088960	386089983
1048570	NL	Amsterdam	52.3556	4.9136	386095616	386096127
1048571	NL	Amsterdam	52.3556	4.9136	386109440	386113535
1048572	NL	Amsterdam	52.3556	4.9136	386191360	386195455
1048573	NL	Amsterdam	52.3556	4.9136	386201600	386203135
1048574	NL	Amsterdam	52.3556	4.9136	386215936	386220031
1048575	NL	Amsterdam	52.3556	4.9136	386228224	386232319
1048576	NL	Amsterdam	52.3556	4.9136	386244608	386244863

	A	B	C	D	E	F	G	H	I
1	BillboardCountry	BillboardPlaceName	boardLatitude	boardLongitude	VisitorCount	VisitorPlaceName	VisitorLatitude	VisitorLongitude	VisitorYearRate
2	DE	Lake	51.7833	8.5667	DE	Lake	51.7833	8.5667	26823960
3	DE	Horb	48.4333	8.6833	DE	Horb	48.4333	8.6833	26823960
4	DE	Horb	48.4333	8.6833	DE	Horb	48.4333	8.6833	53753112
5	DE	Horb	48.4333	8.6833	DE	Horb	48.4333	8.6833	107611416
6	DE	Horb	48.4333	8.6833	DE	Horb	48.4333	8.6833	13359384
7	DE	Horb	48.4333	8.6833	DE	Horb	48.4889	8.6734	26823960
8	DE	Horb	48.4333	8.6833	DE	Horb	48.4889	8.6734	53753112
9	DE	Hardenberg	51.1	7.7333	DE	Hardenberg	51.1	7.7333	26823960
181221	DE	Frankfurt	50.1327	8.7668	DE	Frankfurt	50.1167	8.6833	1157112
181222	DE	Frankfurt	50.1327	8.7668	DE	Frankfurt	50.1167	8.6833	24299352
181223	DE	Frankfurt	50.1327	8.7668	DE	Frankfurt	50.1167	8.6833	807769368
181224	DE	Frankfurt	50.1327	8.7668	DE	Frankfurt	50.1172	8.7281	53753112
181225	DE	Frankfurt	50.1327	8.7668	DE	Frankfurt	50.1172	8.7281	26823960
181226	DE	Frankfurt	50.1327	8.7668	DE	Frankfurt	50.1172	8.7281	107611416
181227	DE	Frankfurt	50.1327	8.7668	DE	Frankfurt	50.1172	8.7281	1577880
181228	DE	Frankfurt	50.1327	8.7668	DE	Frankfurt	50.1184	8.6095	15042456
181229	DE	Frankfurt	50.1327	8.7668	DE	Frankfurt	50.1184	8.6095	10834776
181230	DE	Frankfurt	50.1327	8.7668	DE	Frankfurt	50.1319	8.6838	736344
181231	DE	Frankfurt	50.1327	8.7668	DE	Frankfurt	50.1319	8.6838	0
181232	DE	Frankfurt	50.1327	8.7668	DE	Frankfurt	50.1327	8.7668	736344
181233	DE	Frankfurt	50.1327	8.7668	DE	Frankfurt	50.1492	8.7097	1723360536
181234	DE	Frankfurt	50.1327	8.7668	DE	Frankfurt	50.1492	8.7097	430761240
181235	DE	Frankfurt	50.1327	8.7668	DE	Frankfurt	50.1528	8.745	26823960
181236	DE	Frankfurt	50.1327	8.7668	DE	Frankfurt	50.1878	8.6632	1577880

E. Write a Python / R program to generate GML file from the given csv file.