Practical no 5

**Assessing Data**

1. **Perform error management on the given data using pandas package.**
2. **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)

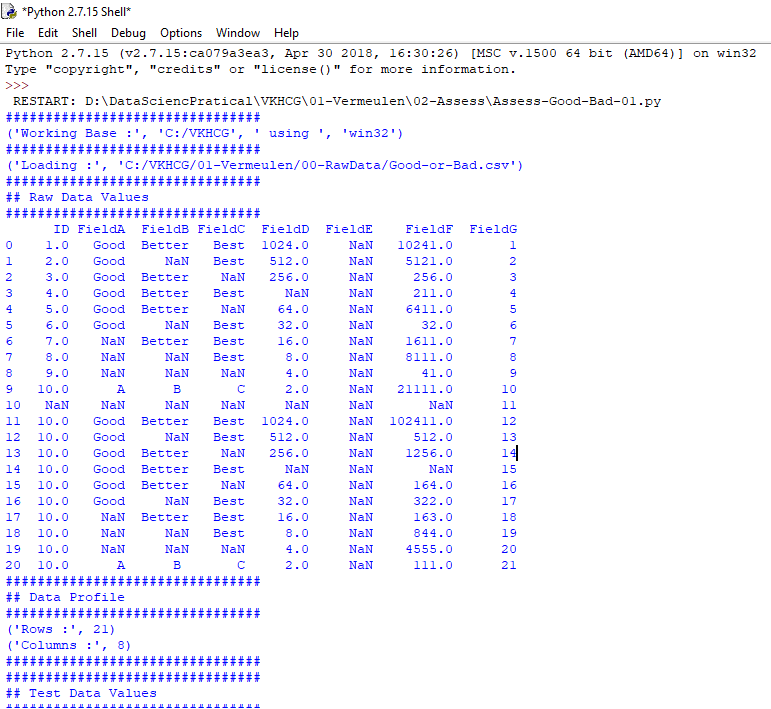
################################################################

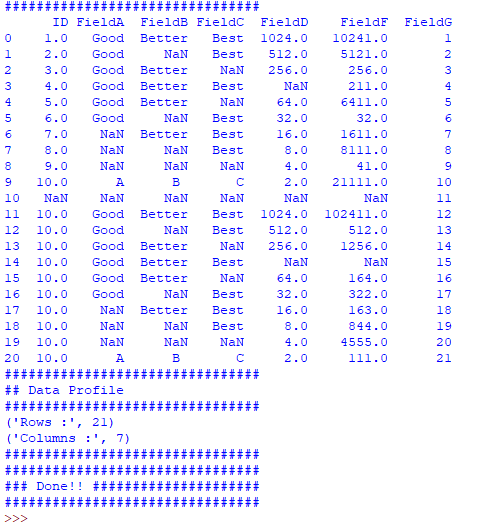
print('################################')

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

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

Output:-





1. **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")

################################################################

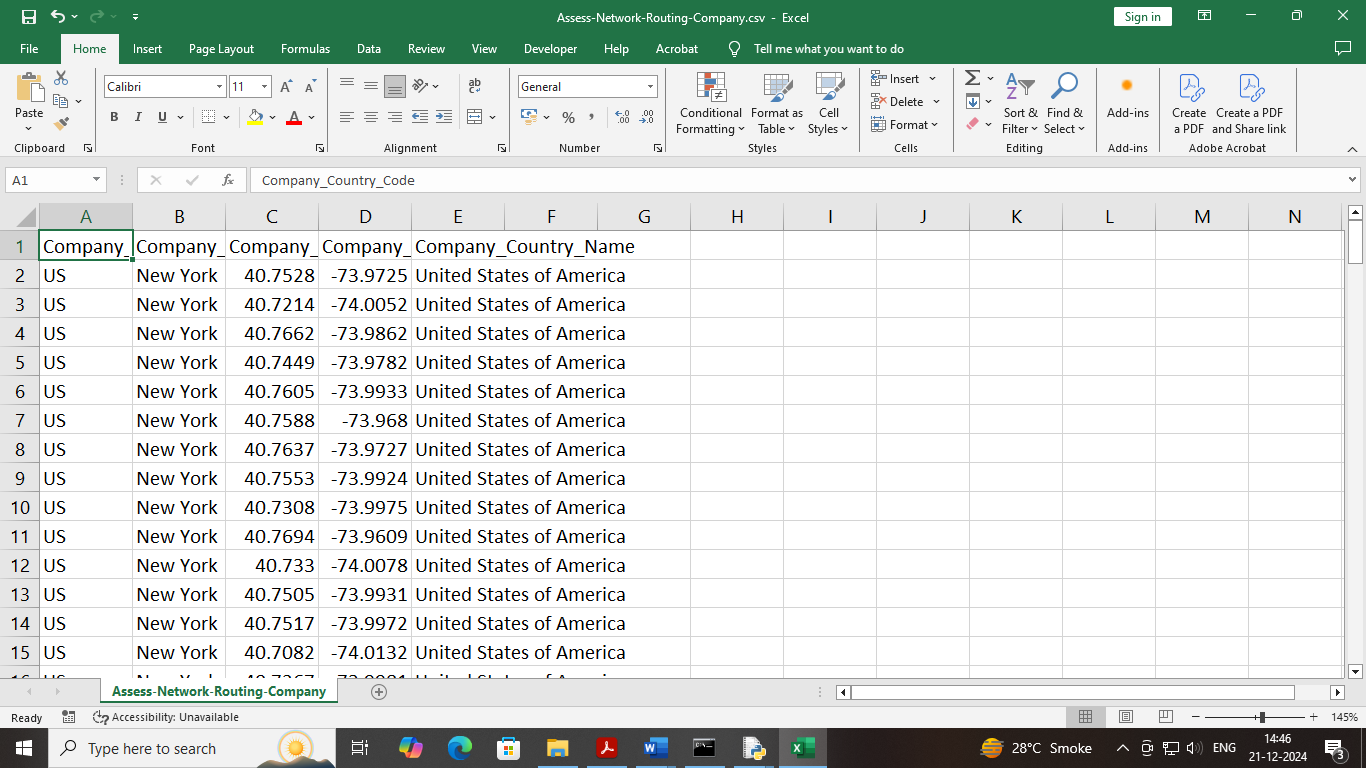
################################################################

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

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

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

output:-



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())

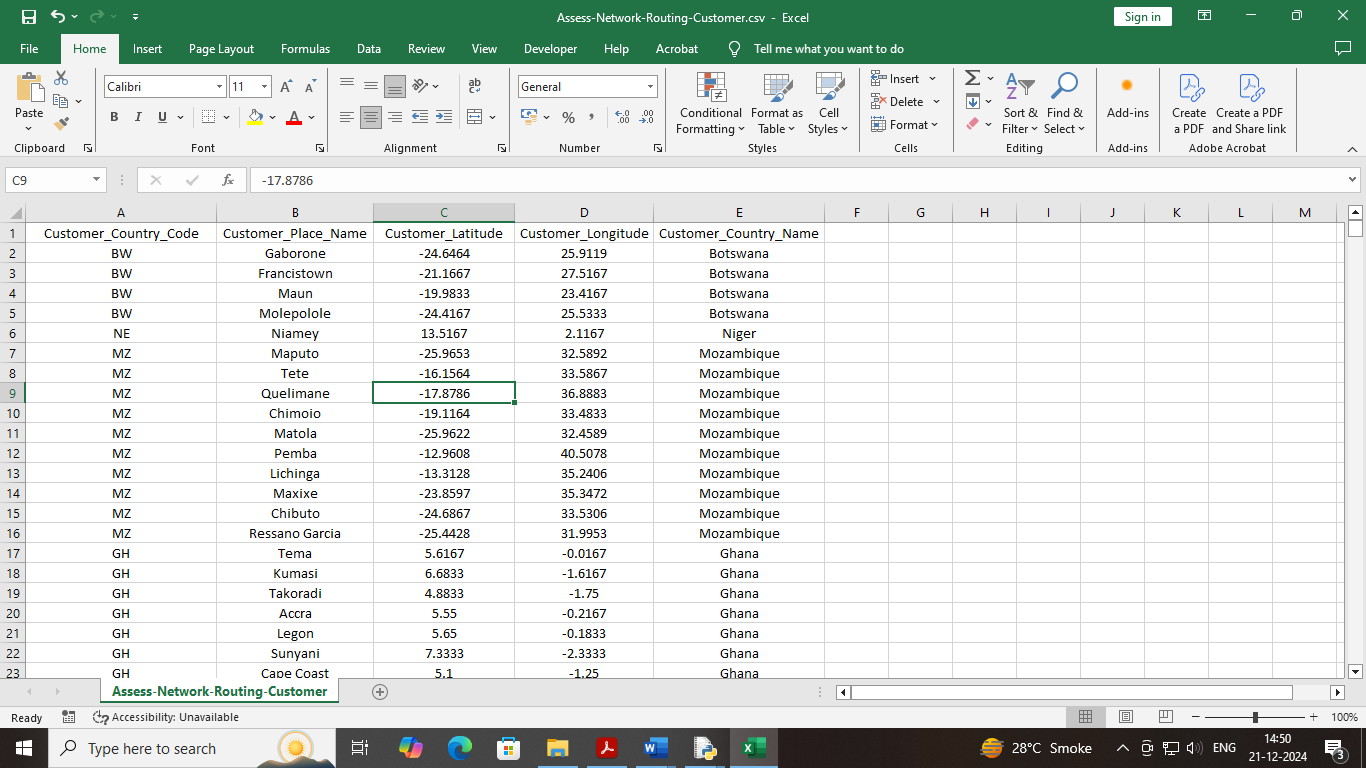
print('################################')

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

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

################################################################

**Output:-**



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")

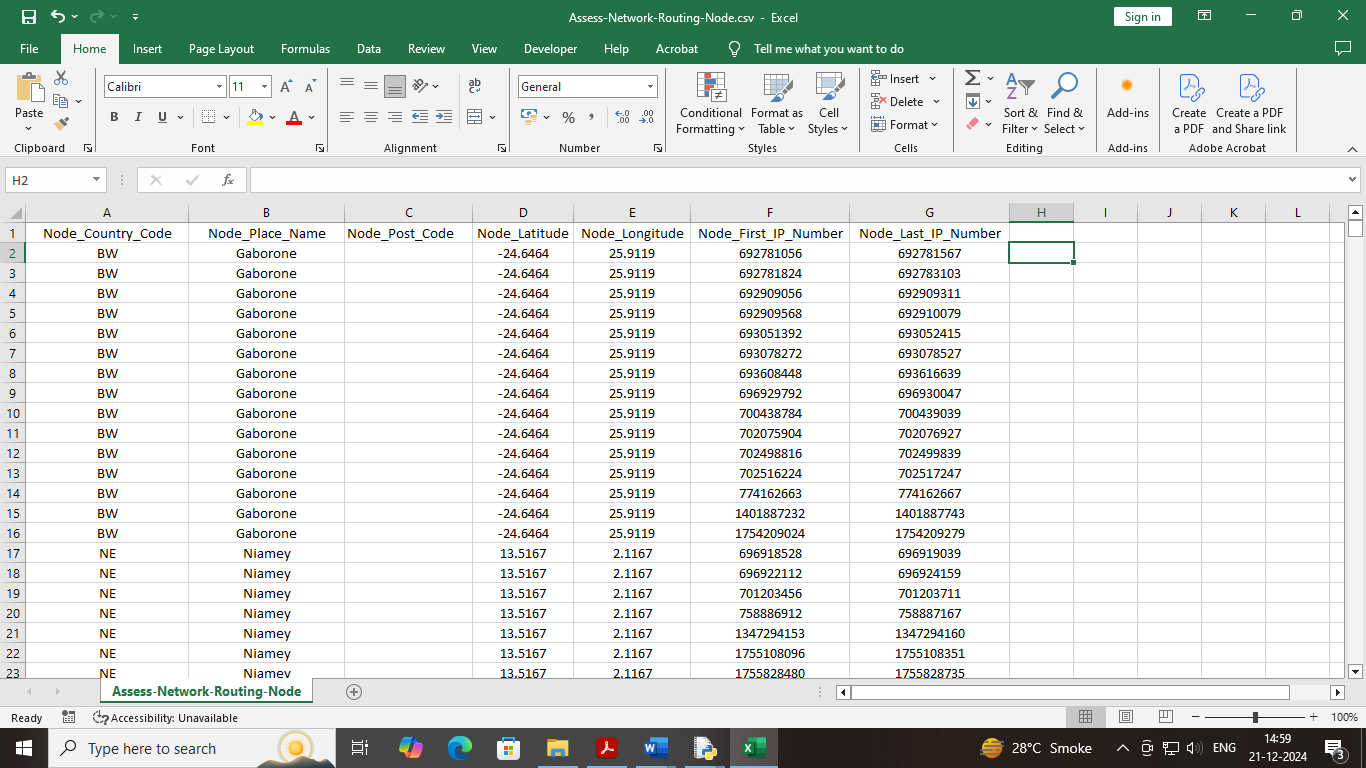
################################################################

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

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

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

Output:-



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

Code:-

1. **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

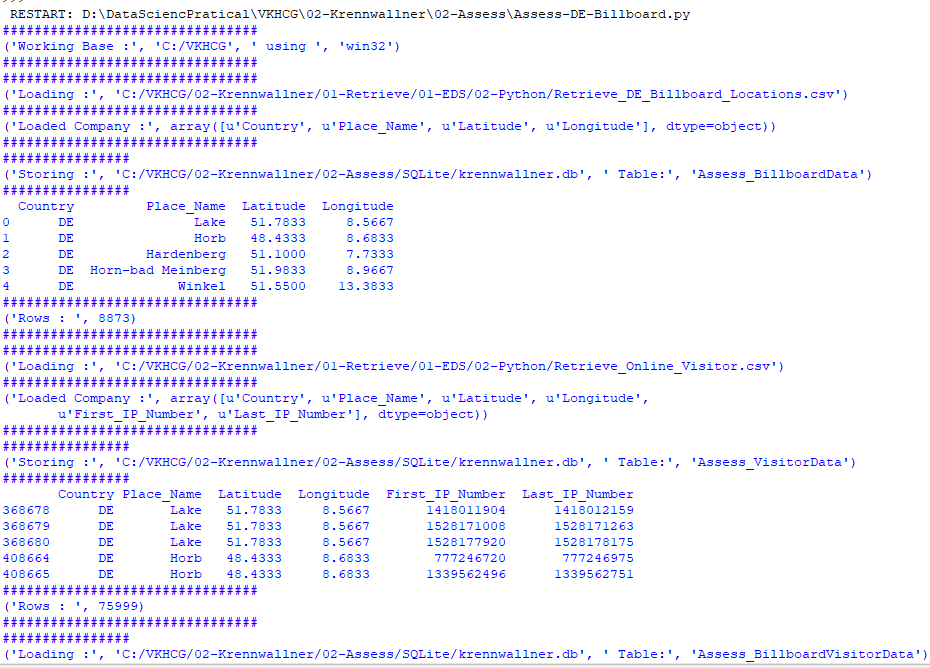
BillboardVistorsData.to\_csv(sFileName, index = False)

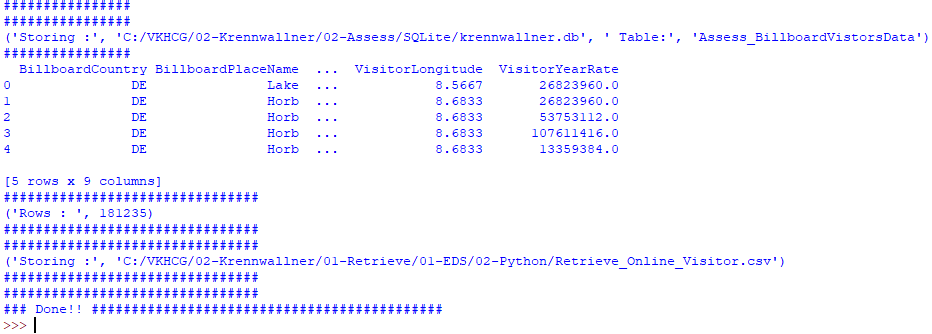
print('################################')

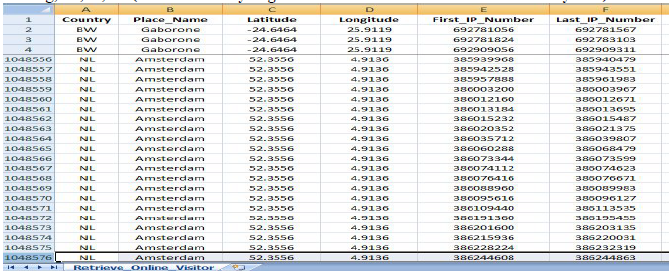
################################################################

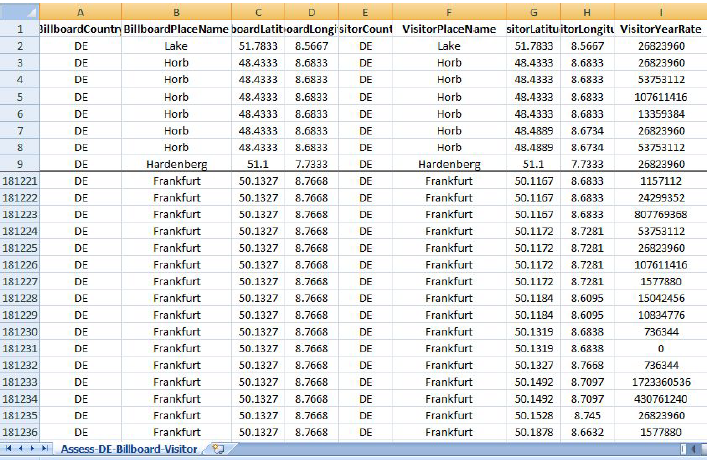
print('### Done!! ############################################')

Output:-









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