ADVANCE II Project

```
import pandas as pd
import numpy as np
```

```
original_df=pd.read_excel('Test Dataset for MIS Specialist and data Analyst.xlsx', index_col=None)
df=original_df.copy() #copy dataframe to make it distinct
```

```
df.head(10)
```

	Project Year	ADVANCE Regional Code	Region	District	Community	Rural or Urban	Farmer ID	Gender	Age	Date of Birth	Edu
0	2014.0	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0001	FEMALE	65	1949-06-01 00:00:00	
1	2014.0	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0002	FEMALE	65	1949-06-01 00:00:00	
2	2014.0	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0003	MALE	60	1954-06-01 00:00:00	
3	2014.0	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0004	MALE	36	1978-06-01 00:00:00	
4	2014.0	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0005	FEMALE	36	1978-06-01 00:00:00	
5	2014.0	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0006	MALE	35	1979-06-01 00:00:00	
6	2014.0	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0007	FEMALE	33	1981-06-01 00:00:00	

7	2014.0	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0008	FEMALE	31	1983-06-01 00:00:00
8	2014.0	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0009	MALE	30	1984-06-01 00:00:00
9	2014.0	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0010	FEMALE	30	1984-06-01 00:00:00

Data Exploration

The Farmer ID contains the region code, which can be used to infer the missing regional code and region fields.

Double-click (or enter) to edit

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27759 entries, 0 to 27758
Data columns (total 14 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   Project Year             27757 non-null  float64
 1   ADVANCE Regional Code    27759 non-null  object 
 2   Region                   27737 non-null  object 
 3   District                 27759 non-null  object 
 4   Community                27758 non-null  object 
 5   Rural or Urban           27759 non-null  object 
 6   Farmer ID                27759 non-null  object 
 7   Gender                   27755 non-null  object 
 8   Age                      27275 non-null  object 
 9   Date of Birth            26435 non-null  object 
 10  Education Level          27090 non-null  object 
 11  Major Crop               19164 non-null  object 
 12  Major Crop (Acres)       19035 non-null  object 
 13  Major Crop Volume (Bags)  19149 non-null  object
dtypes: float64(1), object(13)
memory usage: 3.0+ MB
```

Count null values per column

Only 4 columns out of 14 dont contain missing values

```
u=df.isnull().sum()
print(u)
#print("   ")
#v=df.isna().sum()
#print(v)
```

```
Project Year                   2
ADVANCE Regional Code          0
Region                        22
District                       0
Community                      1
Rural or Urban                 0
Farmer ID                      0
Gender                         4
Age                          484
Date of Birth               1324
Education Level              669
Major Crop                  8595
Major Crop (Acres)          8724
Major Crop Volume (Bags)    8610
dtype: int64
```

Only Project Year column is numeric.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27759 entries, 0 to 27758
```

```
Data columns (total 14 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   Project Year               27757 non-null  float64
 1   ADVANCE Regional Code      27759 non-null  object 
 2   Region                     27737 non-null  object 
 3   District                   27759 non-null  object 
 4   Community                  27758 non-null  object 
 5   Rural or Urban             27759 non-null  object 
 6   Farmer ID                  27759 non-null  object 
 7   Gender                     27755 non-null  object 
 8   Age                        27275 non-null  object 
 9   Date of Birth              26435 non-null  object 
 10  Education Level            27090 non-null  object 
 11  Major Crop                 19164 non-null  object 
 12  Major Crop (Acres)         19035 non-null  object 
 13  Major Crop Volume (Bags)   19149 non-null  object 
dtypes: float64(1), object(13)
memory usage: 3.0+ MB
```

```
df.describe()
```

	Project Year
count	27757.000000
mean	2013.999892
std	0.018007
min	2012.000000
25%	2014.000000
50%	2014.000000
75%	2014.000000
max	2015.000000

Data Cleaning

Project Year

```
f=df['Project Year'].nunique() # containts 2 missing values
print(f)
df['Project Year'].unique() # containts 2 missing values
```

```
3
array([2014.,   nan, 2012., 2015.])
```

Project year 2012, 2015 but data collected 2014?

Only three rows have there values

Assumption 2012 and 2015 values are errors and should be 2014

```
df
```

	Project Year	ADVANCE Regional Code	Region	District	Community	Rural or Urban	Farmer ID	Gender	Age	Date of Birth
0	2014.0	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0001	FEMALE	65	1949-06-01 00:00:00
1	2014.0	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0002	FEMALE	65	1949-06-01 00:00:00
2	2014.0	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0003	MALE	60	1954-06-01 00:00:00
3	2014.0	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0004	MALE	36	1978-06-01 00:00:00

4	2014.0	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0005	FEMALE	36	1978-06-01 00:00:00
...	...	...	...	...	...	...	...	...	...	...
27754	2014.0	UWR	Upper West	WA WEST	NYOLI	RURAL	1UWR1006NF032FR0316	MALE	29	1985-03-25 00:00:00
27755	2014.0	UWR	Upper West	WA WEST	NYOLI	RURAL	1UWR1006NF032FR0317	MALE	27	1987-09-09 00:00:00
27756	2014.0	UWR	Upper West	WA WEST	NYOLI	RURAL	1UWR1006NF032FR0318	MALE	27	1987-05-25 00:00:00
27757	2014.0	UWR	Upper West	WA WEST	NYOLI	RURAL	1UWR1006NF032FR0319	MALE	30	1984-06-12 00:00:00
27758	2014.0	UWR	Upper West	WA WEST	NYOLI	RURAL	1UWR1006NF032FR0320	MALE	52	1962-02-11 00:00:00

27759 rows × 14 columns

```
df['Project Year']= 2014
```

Advance Regional Code

Farmer ID contains appears to contain an additional regional code. However only three regional offices mentioned in the question sheet. No missing values for regional code otherwise.

Based on Map of GHANA, the following assumptions are made:

1. **Wa - Upper West Region Office - UWR**
2. **Tamale - Northern Region Office - NRR**
3. **Bolgatanga - Upper East Region Office - UER**

alt text

```
df['ADVANCE Regional Code'].unique() #
```

```
    array(['NRR', 'UER', 'UWR'], dtype=object)
```

create a new column mapping region code to region name in GHANA

```
df.loc[df['ADVANCE Regional Code'] == 'UER', 'Regional Office'] = 'Bolgatanga'
df.loc[df['ADVANCE Regional Code'] == 'UWR', 'Regional Office'] = 'Wa'
df.loc[df['ADVANCE Regional Code'] == 'NRR', 'Regional Office'] = 'Tamale'
```

Region

Can infer the missing values for this column using regional code. There is some mismatch e.g looking at a map for region code UER, the name in region column can be 'Northern' or 'Northern Region',

e.g Looking at a map of GHANA, East Mamprusi is closer to Bolgatanga(Upper East office) than Tamale(Northern)

therefore using Advance regional code, as truth, it should be changed to 'UPPER EAST' for example.

```
df['Region'].unique()
```

```
    array(['NORTHERN', 'NORTHERN REGION', 'UPPER EAST', nan, 'Upper West '],
          dtype=object)
```

```
df.loc[df['ADVANCE Regional Code'] == 'NRR']
```

	Project Year	ADVANCE Regional Code	Region	District	Community	Rural or Urban	Farmer ID	Gender	Age	Date Bi
0	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0001	FEMALE	65	1949-0 00:0

1	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0002	FEMALE	65	1949-0 00:0
2	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0003	MALE	60	1954-0 00:0
3	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0004	MALE	36	1978-0 00:0
4	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0005	FEMALE	36	1978-0 00:0
...	...	...	...	...	...	...	...	...	...	
12359	2014	NRR	NORTHERN	SABOBA	JILIMAH	RURAL	1NRR0810NF002FR0139	MALE	19	1995-0 00:0
12360	2014	NRR	NORTHERN REGION	SABOBA	NAMUNYONI	RURAL	1NRR0810NF002FR0547	FEMALE	31	1983-0 00:0
12361	2014	NRR	NORTHERN	SABOBA	NAMUNYONI	RURAL	1NRR0810NF002FR0609	MALE	31	1983-0 00:0
12362	2014	NRR	NORTHERN	SABOBA	NAKPANBONNI	RURAL	1NRR0810NF002FR0438	MALE	50	1964-0 00:0
12363	2014	NRR	NORTHERN	YENDI MUNICIPAL	KATINGULI	RURAL	1NRR0810NF002FR0214	MALE	33	1981-0 00:0

12364 rows × 15 columns

```
#original=orginal.infer_objects()
#df['Region']=df['Region']
```

```
# Change values in column 'Region' where column 'ADVANCE Regional Code' is 'UER'
df.loc[df['ADVANCE Regional Code'] == 'UER', 'Region'] = 'UPPER EAST'
df.loc[df['ADVANCE Regional Code'] == 'UWR', 'Region'] = 'UPPER WEST'
df.loc[df['ADVANCE Regional Code'] == 'NRR', 'Region'] = 'NORTHERN'
```

District

No null values, there are some duplicates where the same district is represented by Capital and Common letters e.g. 'sissala west' vs 'Sissala West' vs 'SISSALA WEST', only capital letters will be used and trailing spaces removed. (what happens to this with special characters? Daffiama/Busie/Issa).

```
df['District'].unique()
```

```
    array(['SABOBA ', 'EAST GONJA', 'KUMBUNGU ', 'TOLON ', 'CENTRAL GONJA ',
           'TAMALE METROPOLITAN', 'SANG ', 'GUSHEGU', 'SAVELUGU NANTON',
           'NANUMBA NORTH', 'ZABZUGU', 'WEST GONJA', 'YENDI MUNICIPAL',
           'KARAGA', 'CHEREPONI', 'KPANDAI ', 'MION', 'BUNKPRUGU YUNYOO',
           'EAST MAMPRUSI', 'MAMPRUGU MAOGDURI', 'BUILSA NORTH',
           'BUILSA SOUTH', 'KASSENA NANKANA MUNICIPAL',
           'KASSENA NANKANA WEST', 'BOLGA MUNICIPAL', 'BAWKU WEST',
           'GARU TEMPANE', 'TALENSI NABDAM', 'BINDURI', 'PUSIGA',
           'Sissala West', 'SISSALA WEST', 'SISSALA EAST', 'WA WEST',
           'WA EAST', 'WA MUNICIPAL', 'NADOWLI  EAST', 'SAWLA TUNA KALBA',
           'LAWRA', 'LAMBUSSIE-KARNI', 'NADOWLI-KALEO', 'sissala west',
           'Daffiama/Busie/Issa', 'JIRAPA', 'KALEO-NADOWLI'], dtype=object)
```

```
print(df['District'].nunique()) #number of unique elements
```

```
    45
```

```
df['District']=df['District'].str.upper() # make all strings upper case
df['District']=df['District'].str.strip() # remove trailing spaces
```

Community

1 Null value, some strings have trailing spaces and lower case letters

```
un=df['Community'].nunique()
print(un)
df['Community'].unique()
```

```
827
array(['BAKONDIBA ', 'BIMABOLB ', 'BORGBANI ', 'BUKPAM ', 'BUNGBAL',
       'CHAMBONG ', 'DUNGBANG', 'DUNGBANG ', 'GBENJAK', 'INAGMABONI',
       'JAJAAB', 'KIKPASUNI', 'KINADUK ', 'KUJOONI', 'KUJOONI ',
       'LIYALBU', 'MPAANBE ', 'MPIASAM', 'MULIPIHDO', 'NAKPABOLN',
       'NANKPANBOL', 'NANKPANBOL ', 'NANKPANGNI ', 'NANKPANNI ',
       'SEBOMMA', 'TINGBANI', 'TINGBANI ', 'TUNPIN ', 'UGANDO', 'WABUL',
       'WADUL', 'WAGBALN', 'YANKAZIA ', 'ZAJAAB ', 'CHODASHE', 'CHONASHE',
       'DINDO ', 'FUU', 'GBULLUNG', 'GBULUNG', 'GUN', 'GURUMANCHAGUGILI',
       'JAKPAHI ', 'JANG YILI', 'JUKU', 'KAMONAA KURAA', 'KPANDU ',
       'KPENDUA ', 'KUSAWGU', 'NAGBAH', 'PARISHENAAKURA ',
       'TALI ZOOLANYILI', 'TANSHEGU', 'TINTANG ', 'TONG NOH',
       'TONG NOLI ', 'TONJING', 'TOROPE ', 'VOGGU', 'WANBONGDOKURAA',
       'YOGGU ', 'ZOMLANYILI', 'ZOMLANYILI ', 'ZINIDO', 'TOROPE',
       'YIPELIGU', 'AFAYILI', 'BOLGLINI', 'KANBONYEGA ', 'KPACHELO ',
       'KPANU ', 'KPUNDULI', 'NANTON KURUGU', 'SANVILI ', 'TIBALI ',
       'TOATEYILI', 'BOGU-KAMONAYILI', 'GUSHEGU', 'KPAMDO', 'NALUWA',
       'PUMO', 'TOTI', 'ASULO KURA', 'GAA', 'KADUWA', 'NIEBILIGBINI',
       'NYONG YAPALSI', 'SAMANG YAPALA', 'DEMONAYILI', 'KPATURI',
       'PUSIGA', 'JILO', 'KUPAIKU', 'ZABZUGU', 'KPIRI', 'ZININDO',
       'BAGYILI', 'BUIPE', 'CHAMA', 'WAMBONG', 'BIMBILIA', 'DIPAH',
       'GNORIBOGU', 'KPABI', 'BASAJADO', 'JILMA NO. 1', 'YENDI',
       'NAGNANI', 'SAKPALI', 'TUGBEG', 'ZEI', 'NAGAG', 'NAGANGA',
       'NAKOHUGU', 'ZAKAI', 'ZANKALI', 'ABINGAKURA', 'ALHASSANKURA',
       'CANTEEN', 'DAMONGO', 'NABORI', 'YAGBONKURA', 'YIPALA', 'ZONGO',
       'BIMBILLA', 'SALAA', 'ZANTELI', 'ZANTILI', 'SAMBAGA', 'BAGURUGU',
       'DIDOG TAMALEGU ', 'DIKPUNG', 'KPATARIBOGU', 'KPATARIBOGU ',
       'PISLIGU ', 'SAAKPULI', 'TUYINI', 'KALOGU', 'MABONIGI', 'TECHIPE',
       'ACHUMA', 'ALIA KARIM ', 'CHOMBOSU ', 'COHMBOSU ', 'IUSUNGA ',
       'JAKPA ', 'KURAC ', 'LUULUWA ', 'MAYAMAM ', 'NABURINU ',
       'NAMALAKU ', 'NANDO DIKA ', 'NANOD DIKA ', 'NANSON-NANDO DIKA ',
       'NYANGBANDI ', 'TIEKASU ', 'TINCHANGU ', 'TOMBU', 'TOMBU ',
       'TUSUNGA ', 'WOUJUGA ', 'ZEINABU ABDLAI ', 'BALAI', 'BINAGAM ',
       'BITIGNANDO ', 'DODOPE 1', 'IKPANI', 'KABONBA ', 'KATIEJILE',
       'KPANDAI', 'LESSENI', 'SACHALBU ', 'SHS KPANDAI', 'TKPANI ',
       'WAPAH', 'GBUMGBUM', 'KUKPHEHI', 'NANTON', 'NYOLIGU', 'SANKPAGLA',
       'YOBZERI', 'ZOKUGA', 'ASEIYILI', 'BOGU', 'BUNYILI', 'CHESHE',
       'CHIRITOYILI', 'DABOGUSHEI', 'DINGONI', 'ETHINI', 'GALWEI',
       'GBULAHAGU', 'GBURIMANI', 'GBURUMANI', 'GUNDAA', 'KAANGBAGU',
       'KAANTIEHIYILI', 'KASULIYILI', 'KPACHIYILI', 'KPALGU', 'KPENDUA',
       'KUNGURI', 'KUPALI', 'LUNGBUNGA', 'NAGBLIGU', 'NYANKPALA',
       'NYANKPALA B', 'NYUGJAYILI', 'SABEGU', 'SAGULI', 'TALI',
       'TALI-BOTINGLI', 'TAMALIGU', 'TIBOGUNAAYILI', 'TINYOGU', 'TONG',
       'TUUNAYILI', 'WAANTUGU', 'WARIBOGU', 'WAYANBA', 'WORIBOGU', 'YOGU',
       'ZAGUA', 'ZOOLANYILI', 'SABAR NO.1', 'SABARE ', 'SABARE NO. 2',
       'SABARE NO.1', 'SABARE NO.2', 'SABARE-TINDAN', 'KPASABLO',
       'NYEN SOGBA', 'SUNG', 'ZENYEE', 'LONTO', 'SUNSONG', 'WANTUGU',
       'KPEMBE', 'NKWATA', 'SALAGA', 'SALAGA MEPEASEM', 'KPAKO', 'KUWANI',
       'GNANI', 'DAGBANJADO', 'SEKPE', 'BINCHARATANGA', 'DABOGNI',
       'BAGURNGU', 'BULGU', 'BULGU ', 'BUTNGU ', 'DIGU', 'DUNA ',
       'GBAGBAM', 'GUHEGU', 'GUNU', 'GUSHIGU ', 'KPALGUMA', 'KPUGI',
       'LIMAFONG ', 'LOONI', 'LUNLUWA', 'NAKPA-DABOLI', 'NAKPA-YAPALA',
       'NANDULI ', 'NANGUN KPONG ', 'NANTON-KURUGU', 'NASANDI', 'NYNGALI',
       'NYONG GUMAH', 'NYONG GUMAH ', 'NYONGUMAH ', 'SAHANI ', 'SAMANGA',
       'SAMPAYILI', 'SHEBO ', 'SHELILANYILI ', 'TAMALGH ', 'TINKURUGU',
       'TOGBAN', 'WAZIMANFONG', 'YISHEI', 'ZALI', 'ZAMANSHEGU',
       'ZAMANSHEGU ', 'ZANANTI', 'ZARI', 'BUSUNU', 'TALOLI',
       'ANDO-KAJURA', 'BULASU', 'BUMBURIGA', 'CHERE', 'KPABOKU',
```

```
df['Community'].str.contains(r'[a-z]').sum() # counts number of strings with lower case letters in column
```

```
978
```

```
df['Community']=df['Community'].str.upper() # make all strings upper case
df['Community']=df['Community'].str.strip() # remove trailing spaces
```

```
sorted(pd.unique(df['Community'].dropna())) # haev to remove Null value to sort
##KPRONGI vs KPRONGI #2 and KPRONGI
```

```
['ABIBO',
 'ABINGAKURA',
 'ACHUMA',
 'ADAGBIRA',
 'ADANGOLGU ZEBILLA',
 'ADANKULIGA',
 'ADIBO',
 'ADORSI',
 'ADORSI-BOYA',
 'AFAYILI',
 'AGAGO',
 'AKUNU YI',
 'AKUNUYI',
 'ALHASSANKURA',
 'ALIA KARIM',
 'AMBURE',
 'AMUNTANGA',
 'AMUTANGA SIRIGU',
 'ANDO-KAJURA',
 'ARRAMKOLIGA',
 'ASEIYILI',
 'ASULO KURA',
 'AZUM-SAPIELGA',
 'AZUPUPUGA',
 'AZUWERA',
 'BABILE',
 'BACHUNSA',
 'BACHUNSA - NANYENSA',
 'BACHUNSA- CHANGELINSA',
 'BACHUNSA-YIMONSA',
 'BAGRI',
 'BAGURNGU',
 'BAGURUGU',
 'BAGYILI',
 'BAKOLA',
 'BAKONDIBA',
 'BALAI',
 'BALANSA',
 'BALAZU',
 'BALI',
 'BALOLLO',
 'BANAWA',
 'BANU',
 'BANYONO',
 'BASAJADO',
 'BASSISAN',
 'BASUNDE',
 'BAVUGUNIA',
 'BAYAARUU',
 'BAZUA',
 'BERWONG',
 'BIHEE',
 'BIIHEE',
 'BILINM0NSA',
 'BILLAW',
 'BILLAW TAMPOURI',
 'BILLAW TANGPUORI',
 'BIMABOLB',
```

Rural or Urban

No null values

GLANG is marked as a rural community for other rows

A is paired with community GONSI, which is rural for other rows

same procedure with other strings in this column that are not RURAL ro URBAN.

Strings are changed to all caps to remove ambiguity.

```
un=df['Rural or Urban'].nunique()
print(un)
df['Rural or Urban'].unique()

    9
    array(['RURAL', 'URBAN', 'Rural', 'rural', 'FA', 'NONE', 'GLANG', 'A',
           'BILLAW KADLIGO'], dtype=object)
```

Double-click (or enter) to edit

```
#df.loc[df['Rural or Urban'] == 'GLANG']
#df.loc[df['Rural or Urban'] == 'A']
#df.loc[df['Rural or Urban'] == 'NONE']
df.loc[df['Rural or Urban'] == 'BILLAW KADLIGO']
```

	Project Year	ADVANCE Regional Code	Region	District	Community	Rural or Urban	Farmer ID	Gender	Age	Date of Birth	E
26692	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	BILLAW KADLIGO	1UWR1004NF020FR0049	MALE	30	16/7/84	
26693	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	BILLAW KADLIGO	1UWR1004NF020FR0050	MALE	62	15/7/52	

```
df.loc[df['Community'] == 'BILLAW']
```

	Project Year	ADVANCE Regional Code	Region	District	Community	Rural or Urban	Farmer ID	Gender	Age	Date of Birth
26644	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0001	FEMALE	43	1971-06-15 00:00:00
26645	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0002	FEMALE	51	1963-03-05 00:00:00
26646	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0003	MALE	57	1957-07-16 00:00:00
26647	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0004	MALE	36	1978-07-16 00:00:00
26648	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0005	FEMALE	39	1975-10-18 00:00:00
26649	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0006	MALE	49	1965-07-16 00:00:00
26652	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0009	MALE	52	1962-05-14 00:00:00

			WEST	KARNI						00:00:00
26659	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0016	MALE	43	1971-06-16 00:00:00
26660	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0017	MALE	27	1987-06-16 00:00:00
26661	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0018	FEMALE	47	1967-07-16 00:00:00
26662	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0019	FEMALE	32	1982-06-16 00:00:00
26663	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0020	FEMALE	32	1982-07-15 00:00:00
26664	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0021	FEMALE	52	1962-07-16 00:00:00
26665	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0022	MALE	33	1981-06-15 00:00:00
26666	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0023	MALE	59	1954-07-06 00:00:00
26667	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0024	MALE	64	1950-06-06 00:00:00
26668	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0025	MALE	58	1958-09-10 00:00:00
26669	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0026	MALE	25	1989-07-16 00:00:00
26670	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0027	FEMALE	37	1977-06-06 00:00:00
26671	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0028	MALE	49	1965-06-16 00:00:00
26672	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0029	FEMALE	41	1973-03-20 00:00:00
26673	2014	UWR	UPPER	LAMBUSSIE-	BILLAW	RURAL	1UWR1004NF020FR0030	FEMALE	39	1975-06-16

26673	2014	UWR	WEST	KARNI	BILLAW	RURAL	1UWR1004NF020FR0030	FEMALE	39	00:00:00
26674	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0031	MALE	35	1979-07-16 00:00:00
26675	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0032	MALE	22	1992-06-21 00:00:00
26676	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0033	FEMALE	42	1972-06-16 00:00:00
26677	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0034	MALE	21	1993-07-04 00:00:00
26678	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0035	MALE	43	1971-06-15 00:00:00
26679	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0036	MALE	64	1950-06-16 00:00:00
26680	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0037	FEMALE	42	1972-07-15 00:00:00
26681	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0038	FEMALE	38	1976-07-10 00:00:00
26682	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0039	MALE	39	1975-07-15 00:00:00
26683	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0040	FEMALE	42	1972-06-16 00:00:00
26684	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0041	MALE	38	1976-06-16 00:00:00
26685	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0042	MALE	64	1950-07-10 00:00:00
26686	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0043	FEMALE	62	1962-07-16 00:00:00
26692	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	BILLAW KADLIGO	1UWR1004NF020FR0049	MALE	30	16/7/84

26693	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	BILLAW KADLIGO	1UWR1004NF020FR0050	MALE	62	15/7/52
26694	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0051	FEMALE	57	16/6/57

```
#change to rural or urban based on what corresponding community macthes to in other rows.

df.loc[df['Rural or Urban'] == 'GLANG','Rural or Urban'] = 'RURAL'
df.loc[df['Rural or Urban'] == 'A','Rural or Urban'] = 'RURAL'
df.loc[df['Rural or Urban'] == 'FA','Rural or Urban'] = 'RURAL'
df.loc[df['Rural or Urban'] == 'NONE','Rural or Urban'] = 'RURAL'
df.loc[df['Rural or Urban'] == 'BILLAW KADLIGO','Rural or Urban'] = 'RURAL'
```

```
df['Rural or Urban']=df['Rural or Urban'].str.upper() # make all strings upper case
df['Rural or Urban']=df['Rural or Urban'].str.strip() # remove trailing spaces
```

```
df.loc[df['Community'] == 'BILLAW']
```

	Project Year	ADVANCE Regional Code	Region	District	Community	Rural or Urban	Farmer ID	Gender	Age	Date of Birth
26644	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0001	FEMALE	43	1971-06-15 00:00:00
26645	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0002	FEMALE	51	1963-03-05 00:00:00
26646	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0003	MALE	57	1957-07-16 00:00:00
26647	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0004	MALE	36	1978-07-16 00:00:00

26648	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0005	FEMALE	39	1975-10-18 00:00:00
26649	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0006	MALE	49	1965-07-16 00:00:00
26652	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0009	MALE	52	1962-05-14 00:00:00
26659	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0016	MALE	43	1971-06-16 00:00:00
26660	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0017	MALE	27	1987-06-16 00:00:00
26661	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0018	FEMALE	47	1967-07-16 00:00:00
26662	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0019	FEMALE	32	1982-06-16 00:00:00
26663	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0020	FEMALE	32	1982-07-15 00:00:00
26664	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0021	FEMALE	52	1962-07-16 00:00:00
26665	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0022	MALE	33	1981-06-15 00:00:00
26666	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0023	MALE	59	1954-07-06 00:00:00
26667	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0024	MALE	64	1950-06-06 00:00:00
26668	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0025	MALE	58	1958-09-10 00:00:00
26669	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0026	MALE	25	1989-07-16 00:00:00
26670	2014	UWR	UPPER	LAMBUSSIE-	BILLAW	RURAL	1UWR1004NF020FR0027	FEMALE	37	1977-06-06

			WEST	KARNI						00:00:00
26671	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0028	MALE	49	1965-06-16 00:00:00
26672	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0029	FEMALE	41	1973-03-20 00:00:00
26673	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0030	FEMALE	39	1975-06-16 00:00:00
26674	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0031	MALE	35	1979-07-16 00:00:00
26675	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0032	MALE	22	1992-06-21 00:00:00
26676	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0033	FEMALE	42	1972-06-16 00:00:00
26677	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0034	MALE	21	1993-07-04 00:00:00
26678	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0035	MALE	43	1971-06-15 00:00:00
26679	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0036	MALE	64	1950-06-16 00:00:00
26680	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0037	FEMALE	42	1972-07-15 00:00:00
26681	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0038	FEMALE	38	1976-07-10 00:00:00
26682	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0039	MALE	39	1975-07-15 00:00:00
26683	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0040	FEMALE	42	1972-06-16 00:00:00
26684	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0041	MALE	38	1976-06-16 00:00:00
			UPPER	LAMBUSSIE-						1950-07-10

26685	2014	UWR	WEST	KARNI	BILLAW	RURAL	1UWR1004NF020FR0042	MALE	64	00:00:00
26686	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0043	FEMALE	62	1962-07-16 00:00:00
26692	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0049	MALE	30	16/7/84
26693	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0050	MALE	62	15/7/52
26694	2014	UWR	UPPER WEST	LAMBUSSIE-KARNI	BILLAW	RURAL	1UWR1004NF020FR0051	FEMALE	57	16/6/57

Farmer ID

No null values, the farmer ID appears to contain a ADVANCE region code, however there is an extra NRE code, assumption made that this ties back into the northern region code advance code.

There are 1392 duplicates in the Farmer ID column. They all belong to communities in the district of Saboba.

The Farmer ID should identify farmers individually however, although the Farmers have the same ID, there are consistently different biographic info for each instance of the same ID, which indicates they legitmately represent seperate Farmers. This assumption is made here.

Need to investigate why the farmer ID are being duplicated in Saboba district.

```
r=df['Farmer ID'].nunique()
print(r)
df['Farmer ID'].unique()
```

```
    27063
    array(['1NRR0810NF001FR0001', '1NRR0810NF001FR0002',
           '1NRR0810NF001FR0003', ..., '1UWR1006NF032FR0318',
           '1UWR1006NF032FR0319', '1UWR1006NF032FR0320'],
          shape=(27063,), dtype=object)
```

```
27759-27063
duplicates_all = df[df['Farmer ID'].duplicated(keep=False)]
```

```
duplicates_all
```

	Project Year	ADVANCE Regional Code	Region	District	Community	Rural or Urban	Farmer ID	Gender	Age	Date of Birth
0	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0001	FEMALE	65	1949-06-01 00:00:00
1	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0002	FEMALE	65	1949-06-01 00:00:00
2	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0003	MALE	60	1954-06-01 00:00:00
3	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0004	MALE	36	1978-06-01 00:00:00
4	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0005	FEMALE	36	1978-06-01 00:00:00
...	...	...	...	...	...	...	...	...	...	...
8003	2014	NRR	NORTHERN	SABOBA	DUNGBANG	RURAL	1NRR0810NF001FR0692	MALE	25	1989-06-01 00:00:00
8004	2014	NRR	NORTHERN	SABOBA	DUNGBANG	RURAL	1NRR0810NF001FR0693	MALE	22	1992-06-01 00:00:00
8005	2014	NRR	NORTHERN	SABOBA	DUNGBANG	RURAL	1NRR0810NF001FR0694	MALE	20	1994-06-01 00:00:00
8006	2014	NRR	NORTHERN	SABOBA	DUNGBANG	RURAL	1NRR0810NF001FR0695	MALE	18	1996-06-01 00:00:00
8007	2014	NRR	NORTHERN	SABOBA	DUNGBANG	RURAL	1NRR0810NF001FR0696	MALE	15	1999-06-01

00:00:00

1392 rows × 15 columns

```
df
```

	Project Year	ADVANCE Regional Code	Region	District	Community	Rural or Urban	Farmer ID	Gender	Age	Date of Birth
0	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0001	FEMALE	65	1949-06-01 00:00:00
1	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0002	FEMALE	65	1949-06-01 00:00:00
2	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0003	MALE	60	1954-06-01 00:00:00
3	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0004	MALE	36	1978-06-01 00:00:00
4	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0005	FEMALE	36	1978-06-01 00:00:00
...	...	...	...	...	...	...	...	...	...	...
27754	2014	UWR	UPPER WEST	WA WEST	NYOLI	RURAL	1UWR1006NF032FR0316	MALE	29	1985-03-25 00:00:00
27755	2014	UWR	UPPER WEST	WA WEST	NYOLI	RURAL	1UWR1006NF032FR0317	MALE	27	1987-09-09 00:00:00
27756	2014	UWR	UPPER WEST	WA WEST	NYOLI	RURAL	1UWR1006NF032FR0318	MALE	27	1987-05-25 00:00:00
27757	2014	UWR	UPPER WEST	WA WEST	NYOLI	RURAL	1UWR1006NF032FR0319	MALE	30	1984-06-12 00:00:00
27758	2014	UWR	UPPER	WA WEST	NYOLI	RURAL	1UWR1006NF032FR0320	MALE	52	1962-02-11

```
WEST                                                                      00:00:00
```

27759 rows × 15 columns

Gender

4 missing values, convert FEMALE to F and Male to M

```
r=df['Gender'].nunique()
print(r)
df['Gender'].unique()

    4
    array(['FEMALE', 'MALE', 'M', 'F', nan], dtype=object)
```

```
df.loc[df['Gender'] == 'F','Gender'] = 'FEMALE'
df.loc[df['Gender'] == 'M','Gender'] = 'MALE'
```

Age

Age as at 2014

Calculate missing ages as at Summer 2014 from the D0B.

Assuming ages 10 and below are not entered correctly,(especially 0) set to NULL Some ages show

```
df['Age'].unique()

    array([65, 60, 36, 35, 33, 31, 30, 28, 20, 53, 51, 49, 48, 47, 46, 44, 43,
           41, 40, 39, 38, 37, 32, 29, 26, 64, 54, 50, 45, 42, 34, 61, 59, 56,
           55, 52, 23, 86, 71, 27, 21, 25, 22, 18, 15, 24, 19, 57, 66, 62, 76,
           17, nan, 68, 80, 58, 0, 70, 75, 74, 67, 72, 16, 63, 78, 91, 90, 10,
           69, 95, 13, 79, 11, ' ', 12, 82, 77, 89, 84, 73, 85, 81,
           datetime.datetime(1900, 3, 4, 0, 0), 87, 88, 92, 98,
            datetime.datetime(1900, 2, 4, 0, 0), 83, 7, '50'], dtype=object)
```

```
from datetime import datetime
from pandas.api.types import is_datetime64_any_dtype
#df.loc[df['Age'] == ' ']

#datetimeincol=is_datetime64_any_dtype(df['Age'])
#df.loc[isinstance(df['Age'],datetime)]
p = df['Age'].apply(lambda x: isinstance(x, datetime)) # outputs true or false column where rows are date time

print(p)
p.loc[p==True]
```

```
    0        False
    1        False
    2        False
    3        False
    4        False
             ...
    27754    False
    27755    False
    27756    False
    27757    False
    27758    False
    Name: Age, Length: 27759, dtype: bool
    16489    True
    19102    True
    Name: Age, dtype: bool
```

```
p = df['Age'].apply(lambda x: isinstance(x, int)) # outputs true or false column where rows are integer object

print(p)
p.loc[p==True]
```

```
    0        True
    1        True
    2        True
    3        True
    4        True
```

```
4        True
         ...
27754    True
27755    True
27756    True
27757    True
27758    True
Name: Age, Length: 27759, dtype: bool
0        True
1        True
2        True
3        True
4        True
         ...
27754    True
27755    True
27756    True
27757    True
27758    True
Name: Age, Length: 27271, dtype: bool
```

Find rows where age is not in integer format

```
s=df[~df['Age'].apply(lambda x: isinstance(x, int))] # find rows where Age is not in integer format
sc=s.copy()
s
```

	Project Year	ADVANCE Regional Code	Region	District	Community	Rural or Urban	Farmer ID	Gender	Age	Da
700	2014	NRR	NORTHERN	EAST GONJA	CHODASHE	RURAL	1NRR0809NF001FR0005	FEMALE	NaN	2014 0
701	2014	NRR	NORTHERN	EAST GONJA	CHODASHE	RURAL	1NRR0809NF001FR0006	FEMALE	NaN	2014 0
702	2014	NRR	NORTHERN	EAST GONJA	CHODASHE	RURAL	1NRR0809NF001FR0007	FEMALE	NaN	2014 0

703	2014	NRR	NORTHERN	EAST GONJA	CHODASHE	RURAL	1NRR0809NF001FR0008	FEMALE	NaN	2014 0
704	2014	NRR	NORTHERN	EAST GONJA	CHODASHE	RURAL	1NRR0809NF001FR0009	MALE	NaN	2014 0
...	...	...	...	...	...	...	...	...	...	
18496	2014	UER	UPPER EAST	KASSENA NANKANA WEST	ZENGE	RURAL	1UER0906NF004FR0115	MALE	NaN	1988 0
19102	2014	UER	UPPER EAST	PUSIGA	KONGO	RURAL	1UER0911NF002FR0024	FEMALE	1900-02-04 00:00:00	1979 0
19133	2014	UER	UPPER EAST	PUSIGA	KONGO	RURAL	1UER0911NF002FR0055	MALE	NaN	1983 0
25186	2014	UWR	UPPER WEST	SISSALA EAST	KUROBOI	RURAL	1UWR1003NF036FR0042	MALE	NaN	
26634	2014	UWR	UPPER WEST	SISSALA WEST	GBELLE	RURAL	1UWR1008NF022FR0341	MALE	5O	1964 0

488 rows × 15 columns

For columns Check DOB column if it contains a datetime object if not, attempt to convert to datetime

```
# Function to calculate age
def calculate_age(row):
    current_year = 2014  # Get the current year
    dob = row['Date of Birth']
    age= row['Age']



    if pd.isnull(dob) and pd.isnull(age):  # Check if date_of_birth is missing
        return None
    elif not pd.isnull(age) and isinstance(age,int):
        if age<=10:
            return None
        return age  # Keep the existing age




    try:
        #if int(age)<=10: # might be string '10'
        #    return None
        # Check if dob is already a datetime object
        if isinstance(dob, datetime):
            year_of_birth = dob.year
            #print (current_year - year_of_birth)
        else:
            # Convert dob to datetime if it's a string
            #print(dob)
            dob = pd.to_datetime(dob, errors='coerce')
            if pd.isnull(dob):  # If conversion fails, return None
                return None
            year_of_birth = dob.year
            #print(year_of_birth)
            #print (current_year - year_of_birth)

        # Calculate age

        return current_year - year_of_birth
    except Exception as e:
        return None

df['Age'] = df.apply(calculate_age, axis=1)


df
```

	Project	ADVANCE Regional	Region	District	Community	Rural or	Farmer ID	Gender	Age	Date of

	Year	Regional Code	Region	District	Community	or Urban	Farmer ID	Gender	Age	Birth
0	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0001	FEMALE	65.0	1949-06-01 00:00:0
1	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0002	FEMALE	65.0	1949-06-01 00:00:0
2	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0003	MALE	60.0	1954-06-01 00:00:0
3	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0004	MALE	36.0	1978-06-01 00:00:0
4	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0005	FEMALE	36.0	1978-06-01 00:00:0
...	...	...	...	...	...	...	...	...	...	..
27754	2014	UWR	UPPER WEST	WA WEST	NYOLI	RURAL	1UWR1006NF032FR0316	MALE	29.0	1985-03-2 00:00:0
27755	2014	UWR	UPPER WEST	WA WEST	NYOLI	RURAL	1UWR1006NF032FR0317	MALE	27.0	1987-09-0 00:00:0
27756	2014	UWR	UPPER WEST	WA WEST	NYOLI	RURAL	1UWR1006NF032FR0318	MALE	27.0	1987-05-2 00:00:0
27757	2014	UWR	UPPER WEST	WA WEST	NYOLI	RURAL	1UWR1006NF032FR0319	MALE	30.0	1984-06-1 00:00:0
27758	2014	UWR	UPPER WEST	WA WEST	NYOLI	RURAL	1UWR1006NF032FR0320	MALE	52.0	1962-02-11 00:00:0

27759 rows × 15 columns

```
sc
```

Project **ADVANCE** **Rural** **Da**

	Year	Regional Code	Region	District	Community	or Urban	Farmer ID	Gender	Age	
700	2014	NRR	NORTHERN	EAST GONJA	CHODASHE	RURAL	1NRR0809NF001FR0005	FEMALE	NaN	2014 00
701	2014	NRR	NORTHERN	EAST GONJA	CHODASHE	RURAL	1NRR0809NF001FR0006	FEMALE	NaN	2014 00
702	2014	NRR	NORTHERN	EAST GONJA	CHODASHE	RURAL	1NRR0809NF001FR0007	FEMALE	NaN	2014 00
703	2014	NRR	NORTHERN	EAST GONJA	CHODASHE	RURAL	1NRR0809NF001FR0008	FEMALE	NaN	2014 00
704	2014	NRR	NORTHERN	EAST GONJA	CHODASHE	RURAL	1NRR0809NF001FR0009	MALE	NaN	2014 00
...	...	...	...	...	...	...	...	...	...	
18496	2014	UER	UPPER EAST	KASSENA NANKANA WEST	ZENGE	RURAL	1UER0906NF004FR0115	MALE	NaN	1988 00
19102	2014	UER	UPPER EAST	PUSIGA	KONGO	RURAL	1UER0911NF002FR0024	FEMALE	1900-02-04 00:00:00	1979 00
19133	2014	UER	UPPER EAST	PUSIGA	KONGO	RURAL	1UER0911NF002FR0055	MALE	NaN	1983 00
25186	2014	UWR	UPPER WEST	SISSALA EAST	KUROBOI	RURAL	1UWR1003NF036FR0042	MALE	NaN	
26634	2014	UWR	UPPER WEST	SISSALA WEST	GBELLE	RURAL	1UWR1008NF022FR0341	MALE	5O	1964 00

488 rows × 15 columns

```
#sc['Age'] = sc.apply(lambda row: row['Age'] == (2014 - row['Date of Birth'].year) if isinstance(row['Date of
sc
```

	Project Year	ADVANCE Regional Code	Region	District	Community	Rural or Urban	Farmer ID	Gender	Age	Da
700	2014	NRR	NORTHERN	EAST GONJA	CHODASHE	RURAL	1NRR0809NF001FR0005	FEMALE	NaN	2014 00
701	2014	NRR	NORTHERN	EAST GONJA	CHODASHE	RURAL	1NRR0809NF001FR0006	FEMALE	NaN	2014 00
702	2014	NRR	NORTHERN	EAST GONJA	CHODASHE	RURAL	1NRR0809NF001FR0007	FEMALE	NaN	2014 00
703	2014	NRR	NORTHERN	EAST GONJA	CHODASHE	RURAL	1NRR0809NF001FR0008	FEMALE	NaN	2014 00
704	2014	NRR	NORTHERN	EAST GONJA	CHODASHE	RURAL	1NRR0809NF001FR0009	MALE	NaN	2014 00
...	...	...	...	...	...	...	...	...	...	
18496	2014	UER	UPPER EAST	KASSENA NANKANA WEST	ZENGE	RURAL	1UER0906NF004FR0115	MALE	NaN	1988 00
19102	2014	UER	UPPER EAST	PUSIGA	KONGO	RURAL	1UER0911NF002FR0024	FEMALE	1900-02-04 00:00:00	1979 00
19133	2014	UER	UPPER EAST	PUSIGA	KONGO	RURAL	1UER0911NF002FR0055	MALE	NaN	1983 00
25186	2014	UWR	UPPER WEST	SISSALA EAST	KUROBOI	RURAL	1UWR1003NF036FR0042	MALE	NaN	
26634	2014	UWR	UPPER WEST	SISSALA WEST	GBELLE	RURAL	1UWR1008NF022FR0341	MALE	5O	1964 00

488 rows × 15 columns

```
#df['Age'] = df['Age'].apply(lambda x: x if isinstance(x, int) else 2014-)
```

```
print(p.sum())
```

```
    27271
```

Date of Birth

This column is very messy, use to fill in missing values of age column then drop, as it essentially tells the same information

```
f=df['Date of Birth'].unique()
print(f)
```

```
    [datetime.datetime(1949, 6, 1, 0, 0) datetime.datetime(1954, 6, 1, 0, 0)
     datetime.datetime(1978, 6, 1, 0, 0) ...
     datetime.datetime(1987, 5, 25, 0, 0) datetime.datetime(1984, 6, 12, 0, 0)
     datetime.datetime(1962, 2, 11, 0, 0)]
```

```
#from datetime import datetime
#from pandas.api.types import is_datetime64_any_dtype
#df.loc[df['Age'] == ' ']

#datetimeincol=is_datetime64_any_dtype(df['Age'])
#df.loc[isinstance(df['Age'],datetime)]
#q = df['Date of Birth'].apply(lambda x: isinstance(x, datetime)) # outputs true or false column where rows ar

#print(q)
#print(q.loc[q==True].sum())
```

```
df=df.drop('Date of Birth',axis=1)# drop Date of Birth Column
```

Education Level

According to Ghanian goverment websit: https://gh.usembassy.gov/education-culture/educationusa-center/educational-system-ghana/ education levels are:

```
Primary School – 6 years
Junior Secondary/High School – 3 years
Senior Secondary School – 3 years
    (Senior High School entrants 2007- 2009 – 4 years)
University Bachelor’s Degree – 4 years
```

Fit the data in column to 5 categories - Primary, Junior High, Senior High, Teritiary and None

```
df
```

	Project Year	ADVANCE Regional Code	Region	District	Community	Rural or Urban	Farmer ID	Gender	Age	Educatic Leve
0	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0001	FEMALE	65.0	NON
1	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0002	FEMALE	65.0	NON
2	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0003	MALE	60.0	NON
3	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0004	MALE	36.0	NON
4	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0005	FEMALE	36.0	NON
...	...	...	...	...	...	...	...	...	...	
27754	2014	UWR	UPPER WEST	WA WEST	NYOLI	RURAL	1UWR1006NF032FR0316	MALE	29.0	POS SECONDAF
27755	2014	UWR	UPPER	WA WEST	NYOLI	RURAL	1UWR1006NF032FR0317	MALE	27.0	POS

			WEST							SECONDAF
27756	2014	UWR	UPPER WEST	WA WEST	NYOLI	RURAL	1UWR1006NF032FR0318	MALE	27.0	SECONDAF
27757	2014	UWR	UPPER WEST	WA WEST	NYOLI	RURAL	1UWR1006NF032FR0319	MALE	30.0	SECONDAF
27758	2014	UWR	UPPER WEST	WA WEST	NYOLI	RURAL	1UWR1006NF032FR0320	MALE	52.0	NON

27759 rows × 14 columns

```
d=df['Education Level'].nunique()
print(d)
df['Education Level'].unique()
```

```
73
array(['NONE', 'MIDDLE SCHOOL', 'PRIMARY', 'SECONDARY', 'TERTIARY',
       'NON-FORMAL EDUCAITON', 'POST SECONDARY', 'SECONDARY ', 'SHS',
       'JHS', 'NON-FORMAL', 'POST-SEC', 'NON FORMAL', 'POST-SECONDARY',
       nan, 'MIDDLE SCH', 'POST SEC', 'PRI', 'MID SCH', 'N FORMAL',
       'NON FORMAL EDUCATION', ' PRIMARY', 'MIDDLESCHOOL', 'NOE',
       'N-FORMAL', 'None ', 'Middle School', 'Tertiary', 'Secondary',
       'Primary', 'SECONDARY O LEVEL', 'NON-FORMAL EDUCATION', 'NOEN',
       'NON', 'N0N', 'NO', 248445152, 248263931, 'NON-FORMAL EDU',
       'PRIMARY SCH', 'SECONDARY SCH', 'POST SEC SCH', 263721752,
       54282575, 241592648, 'N', 'N0NE', 'MIDDDLE SCHOOL',
       'SECONDARY SCHOOL', 'MLDDLE SVH', 'PRIM.', 'MIDDLE',
       'NONE- FORMAL EDUCATION', 'primary', 'Middle school',
       'Primary school', 'None-formal', 'None-Formal Education', 'none',
       'Post Secondary', 'Non-formal Education', 'none ',
       'Non-Fromal Education ', 'Tertary', 'SECNDARY', 'NFE', 'NOONE',
       'TERTAIRY', 'TETIARY', ' NONE', 'Secndary', 'Secondary ',
       'NONE-FORMAL', 'NONE FORMAL'], dtype=object)
```

```
df.loc[(df['Education Level'].str.contains('^NON|NONE|^NO|^Non|^none|^no|None|^N$|^NFE$|^N For|^N FORMAL|^N Fo
#df.loc[df['Education Level'].apply(lambda x: isinstance(x, int)), 'Education Level'] = 'NONE' #replace intege
df.loc[(df['Education Level'].str.contains('^Primary|PRIM|^Prim|^prim|^PRI$',na=False)), 'Education Level'] =
df.loc[(df['Education Level'].str.contains('^Mid|^MID|^mid|^middle|^JHS|^MLDDLE|^JUNIOR',na=False)), 'Educatio
df.loc[(df['Education Level'].str.contains('^SHS|O LEVEL|^Secondary$|^Secndary|^SECNDARY|^SECONDARY SCHOOL|^SE
df.loc[(df['Education Level'].str.contains('^TER|^Ter|^Post-S|^POST SEC|^POST-SEC|^Post S|TET',na=False)), 'Ed

#might be best to drop rows with integer values


df
```

	Project Year	ADVANCE Regional Code	Region	District	Community	Rural or Urban	Farmer ID	Gender	Age	Educatio Leve
0	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0001	FEMALE	65.0	NON
1	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0002	FEMALE	65.0	NON
2	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0003	MALE	60.0	NON
3	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0004	MALE	36.0	NON
4	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0005	FEMALE	36.0	NON
...	...	...	...	...	...	...	...	...	...	..
27754	2014	UWR	UPPER WEST	WA WEST	NYOLI	RURAL	1UWR1006NF032FR0316	MALE	29.0	TERITIAR
27755	2014	UWR	UPPER WEST	WA WEST	NYOLI	RURAL	1UWR1006NF032FR0317	MALE	27.0	TERITIAR
27756	2014	UWR	UPPER WEST	WA WEST	NYOLI	RURAL	1UWR1006NF032FR0318	MALE	27.0	SENIO HIG
27757	2014	UWR	UPPER WEST	WA WEST	NYOLI	RURAL	1UWR1006NF032FR0319	MALE	30.0	SENIO HIG
27758	2014	UWR	UPPER WEST	WA WEST	NYOLI	RURAL	1UWR1006NF032FR0320	MALE	52.0	NON

27759 rows × 14 columns

```
#drop rows where the column contain integer values
df = df[~df['Education Level'].apply(lambda x: isinstance(x, int))]
```

Major Crop

Investigate why crop data not collected for Upper West region

```
d=df['Major Crop'].nunique()
print(d)
df['Major Crop'].unique()
```

```
4
array(['SOYBEAN', 'MAIZE', 'SOYA', 'RICE', nan], dtype=object)
```

```
df.loc[(df['Major Crop'].str.contains('SOYA',na=False)), 'Major Crop'] = 'SOYBEAN' #soya is the same as soybea
```

```
#df=df.dropna(subset=['Major Crop']) # drop rows with missing values
```

Major Crop (Acres)

```
d=df['Major Crop (Acres)'].nunique()
print(d)
df['Major Crop (Acres)'].unique()
```

```
100
array([2, 3, 1, 4, 5, 7, 30, 10, 8, nan, 6, 9, 11, 12, 21, 31, 15, 20, 25,
       19, 14, 18, 13, 16, 76, 50, 40, 1.5, 60, 0, 0.5, 100, 3.2, 3.5,
       2.5, 1.6, 26, 22, 17, 45, 42, 32, 87, 2.9999999999999996, '1ACRE',
       '2ACRES', '3ACRES', '4ACRES', '1ACER', '15ACRE', '9ACRES',
       '7ACRES', '5ACRE', '5 ACRES', '6 ACRES', '7 ACRES', '8 ACRES',
       '9 ACRES', '14 ACRES', '12 ACRES', '10 ACRES', '13 ACRES',
       '11 ACRES', '15 ACRES', '6ACRES', '20ACRES', '8ACRES', '2ACRE',
       '10ACRES', '30ACRES', '25ACRES', '3 ACRES', '5 COCOA SACK',
       '2 ACRES', '18ACRE', '4A', '20A', '4ACRE', '6 ACRE', '4 ACRE',
       '4½', 4.5, '1,5', '½', '2½', '3½', 1.75, 75, 7.5, 0.75, 0.4, 1.4,
       '`1', 0.25, 1.9, 0.6, 0.8, 'O.8', 2.75, 1.7, '1.5ACRES'],
      dtype=object)
```

```
df.loc[df['Major Crop (Acres)'].str.contains('ACRES',na=False)]
```

	Project Year	ADVANCE Regional Code	Region	District	Community	Rural or Urban	Farmer ID	Gender	Age	Education Level	M
12449	2014	UER	UPPER EAST	EAST MAMPRUSI	BOAYINI	RURAL	1NRE0818NF002FR0008	FEMALE	46.0	NONE	M
12450	2014	UER	UPPER EAST	EAST MAMPRUSI	BOAYINI	RURAL	1NRE0818NF002FR0009	FEMALE	40.0	NONE	M
12451	2014	UER	UPPER EAST	EAST MAMPRUSI	BOAYINI	RURAL	1NRE0818NF002FR0010	FEMALE	45.0	NONE	M
12452	2014	UER	UPPER EAST	EAST MAMPRUSI	BOAYINI	RURAL	1NRE0818NF002FR0011	FEMALE	36.0	NONE	M
12453	2014	UER	UPPER EAST	EAST MAMPRUSI	BOAYINI	RURAL	1NRE0818NF002FR0012	FEMALE	32.0	NONE	M
...	...	...	...	...	...	...	...	...	...	...	
18510	2014	UER	UPPER EAST	GARU TEMPANE	KPATIA	RURAL	1UER0907NF001FR0129	MALE	37.0	JUNIOR HIGH	M
18511	2014	UER	UPPER EAST	GARU TEMPANE	KPATIA	RURAL	1UER0907NF001FR0130	FEMALE	29.0	NONE	M
18512	2014	UER	UPPER EAST	GARU TEMPANE	KPATIA	RURAL	1UER0907NF001FR0131	FEMALE	36.0	JUNIOR HIGH	M

18513	2014	UER	UPPER EAST	GARU TEMPANE	KPATIA	RURAL	1UER0907NF001FR0132	FEMALE	30.0	PRIMARY	M
18514	2014	UER	UPPER EAST	GARU TEMPANE	KPATIA	RURAL	1UER0907NF001FR0133	MALE	36.0	SENIOR HIGH	M

165 rows × 14 columns

```
df
```

	Project Year	ADVANCE Regional Code	Region	District	Community	Rural or Urban	Farmer ID	Gender	Age	Educatio Leve
0	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0001	FEMALE	65.0	NONI
1	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0002	FEMALE	65.0	NONI
2	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0003	MALE	60.0	NONI
3	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0004	MALE	36.0	NONI
4	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0005	FEMALE	36.0	NONI
...	...	...	...	...	...	...	...	...	...	..
27754	2014	UWR	UPPER WEST	WA WEST	NYOLI	RURAL	1UWR1006NF032FR0316	MALE	29.0	TERITIAR
27755	2014	UWR	UPPER WEST	WA WEST	NYOLI	RURAL	1UWR1006NF032FR0317	MALE	27.0	TERITIAR
27756	2014	UWR	UPPER WEST	WA WEST	NYOLI	RURAL	1UWR1006NF032FR0318	MALE	27.0	SENIO HIGI
27757	2014	UWR	UPPER WEST	WA WEST	NYOLI	RURAL	1UWR1006NF032FR0319	MALE	30.0	SENIO HIGI

27758	2014	UWR	UPPER WEST	WA WEST	NYOLI	RURAL	1UWR1006NF032FR0320	MALE	52.0	NONI

27754 rows × 14 columns

```
# Function to calculate age
import re
def major_crop_acres(row):
    #current_year = 2014  # Get the current year
    #dob = row['Date of Birth']
    #age= row['Age']
    crop_acres = row['Major Crop (Acres)']
    #print('HERE0')

    if pd.isnull(crop_acres):  # Check if date_of_birth is missing
        #print('HERE1')
        return None

    try:
        # Check if dob is already a datetime object
        if isinstance(crop_acres, int) or isinstance(crop_acres,float):
            #print('HERE1.5')
            #print(crop_acres)
            return crop_acres

            #print (current_year - year_of_birth)
        else:
            if '½' in crop_acres:
                #print('before: '+crop_acres)
                crop_acres=crop_acres.replace('½','.5') # replace occurences of '½' with .5
                #print('After: '+ crop_acres)

            val=re.search(r'([-+]?\d*\.?\d+)',crop_acres) # finds a float or integer value from string

            #val=crop_acres.search(r'\d+)')
            #print('HERE2_2')
            #print(val)
            return val.group()
    except Exception as e:
        #print (e)
        return None

#df['Major Crop (Acres Values)'] = df.apply(major_crop_acres, axis=1)
df.loc[:, 'Major Crop (Acres)']= df.apply(major_crop_acres, axis=1)
#df['Major Crop (Acres Values)']=pd.to_numeric(df['Major Crop (Acres Values)'], errors='coerce')


df['Major Crop (Acres)']=pd.to_numeric(df['Major Crop (Acres)'], errors='coerce') # make column numeric
#df.loc[:, 'Major Crop (Acres)']=pd.to_numeric(df['Major Crop (Acres)'], errors='coerce') # make column numeri
```

```
C:\Users\learn\AppData\Local\Temp\ipykernel_11692\1862951956.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.htm
  df['Major Crop (Acres)']=pd.to_numeric(df['Major Crop (Acres)'], errors='coerce') # make column numeric
```

Major Crop Volume (Bags)

1 Bag has different weights depending on the type of crop?

Based on this article, https://www.myjoyonline.com/bags-of-maize-to-be-sold-on-weight-basis-to-offer-farmers-value-under-pfj-2/

1 bag Maiz weighs 50Kg

https://ghana.un.org/en/269522-eu-food-security-response-12600-smallholder-farmers-receive-agricultural-inputs-northern

Assumption 1 Bag weights 50 Kg (other sources 45.)

Assumptions

1 BAG in general is 50 Kg

1 Max bag is 50lbs and 1 Mini bag is 25 lbs

Maize-50 Kg

Rice 45.3 Kg - Maxi 22.7 or 11.3 (Mini)

Soybean/Soya- 22.7 Kg(maxi) or 11.3 Kg(Mini)

Code so values can be changed easily and recalculated if necessary

where you have 5X6MINI fpr example use the first number as the bag number second numbeer correlates to the acres column

```
d=df['Major Crop Volume (Bags)'].nunique()
print(d)
df['Major Crop Volume (Bags)'].unique()
```

```
1070
array(['4 BAGS', '12 BAGS', '8 BAGS', ..., '21MAXI', '4.5MAXI', '9.5MAXI'],
      shape=(1071,), dtype=object)
```

```
df
```

	Project Year	ADVANCE Regional Code	Region	District	Community	Rural or Urban	Farmer ID	Gender	Age	Educatio Leve
0	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0001	FEMALE	65.0	NONI
1	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0002	FEMALE	65.0	NONI
2	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0003	MALE	60.0	NONI
3	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0004	MALE	36.0	NONI
4	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0005	FEMALE	36.0	NONI
...	...	...	...	...	...	...	...	...	...	..
27754	2014	UWR	UPPER WEST	WA WEST	NYOLI	RURAL	1UWR1006NF032FR0316	MALE	29.0	TERITIAR
27755	2014	UWR	UPPER	WA WEST	NYOLI	RURAL	1UWR1006NF032FR0317	MALE	27.0	TERITIAR

			WEST							
27756	2014	UWR	UPPER WEST	WA WEST	NYOLI	RURAL	1UWR1006NF032FR0318	MALE	27.0	SENIO HIG
27757	2014	UWR	UPPER WEST	WA WEST	NYOLI	RURAL	1UWR1006NF032FR0319	MALE	30.0	SENIO HIG
27758	2014	UWR	UPPER WEST	WA WEST	NYOLI	RURAL	1UWR1006NF032FR0320	MALE	52.0	NON

27754 rows × 14 columns

```
# Function to crop
import re
def major_crop_weights(row):
    crop=row['Major Crop']
    crop_volume = row['Major Crop Volume (Bags)']


    if pd.isnull(crop_volume):  # Check if crop volume value is missing
        #print('HERE1')
        return None

    try:

        #array(['SOYBEAN', 'MAIZE', 'RICE', nan], dtype=object)
        if '½' in crop_volume:
                crop_volume=crop_volume.replace('½','.5')
        if crop == 'MAIZE':
            if 'MAX'in crop_volume or 'Max' in crop_volume or 'MXAI' in crop_volume or 'NXIA' in crop_volume:
                num_max_bags=float(re.search(r'(\d*\.?\d+)',crop_volume).group())
                return num_max_bags*22.7
            elif 'MIN' in crop_volume or 'Min' in crop_volume:
                num_min_bags=float(re.search(r'(\d*\.?\d+)',crop_volume).group())
                return num_min_bags*11.3
            elif 'KG' in crop_volume or 'Kg' in crop_volume or 'kg' in crop_volume:
                num_bag_and_kg=re.findall(r'(\d*\.?\d+)',crop_volume)
                num_bags=float(num_bag_and_kg[0])
                weight_kg=float(num_bag_and_kg[1])
                return num_bags*weight_kg
            elif 'TON' in crop_volume or 'ton' in crop_volume or 'Ton' in crop_volume:
                weight_tons=float(re.search(r'(\d*\.?\d+)',crop_volume).group())
                return weight_tons*1000 #assume metric tonn
            else:
                num_bags=float(re.search(r'(\d*\.?\d+)',crop_volume).group())
                return num_bags*50

        elif crop == 'RICE':
            if 'MAX'in crop_volume or 'Max' in crop_volume or 'MXAI' in crop_volume or 'NXIA' in crop_volume:
                num_max_bags=float(re.search(r'(\d*\.?\d+)',crop_volume).group())
                return num_max_bags*22.7
            elif 'MIN' in crop_volume or 'Min' in crop_volume:
                num_min_bags=float(re.search(r'(\d*\.?\d+)',crop_volume).group())
                return num_min_bags*11.3
            elif 'KG' in crop_volume or 'Kg' in crop_volume or 'kg' in crop_volume:
                num_bag_and_kg=re.findall(r'(\d*\.?\d+)',crop_volume)
                num_bags=float(num_bag_and_kg[0])
                weight_kg=float(num_bag_and_kg[1])
                return num_bags*weight_kg
            elif 'TON' in crop_volume or 'ton' in crop_volume or 'Ton' in crop_volume:
                weight_tons=float(re.search(r'(\d*\.?\d+)',crop_volume).group())
                return weight_tons*1000 #assume metric tonn
            else:
                num_bags=float(re.search(r'(\d*\.?\d+)',crop_volume).group())
                return num_bags*50
        elif crop == 'SOYBEAN':
            if 'MAX'in crop_volume or 'Max' in crop_volume or 'MXAI' in crop_volume or 'NXIA' in crop_volume:
                num_max_bags=float(re.search(r'(\d*\.?\d+)',crop_volume).group())
                return num_max_bags*22.7
            elif 'MIN' in crop_volume or 'Min' in crop_volume:
                num_min_bags=float(re.search(r'(\d*\.?\d+)',crop_volume).group())
                return num_min_bags*11.3
            elif 'KG' in crop_volume or 'Kg' in crop_volume or 'kg' in crop_volume:
                num_bag_and_kg=re.findall(r'(\d*\.?\d+)',crop_volume)
                num_bags=float(num_bag_and_kg[0])
                weight_kg=float(num_bag_and_kg[1])
                return num_bags*weight_kg
            elif 'TON' in crop_volume or 'ton' in crop_volume or 'Ton' in crop_volume:
                weight_tons=float(re.search(r'(\d*\.?\d+)',crop_volume).group())
                return weight_tons*1000 #assume metric tonn
            else:
                num_bags=float(re.search(r'(\d*\.?\d+)',crop_volume).group())
                return num_bags*50

        else: # only three crops considered
            return None

    except Exception as e:
        #print (e)
        return None

df.loc[:, 'Major Crop (TONNES)'] = df.apply(major_crop_weights, axis=1)
df['Major Crop (TONNES)']= df['Major Crop (TONNES)']/1000
```

```
C:\Users\learn\AppData\Local\Temp\ipykernel_11692\2253427923.py:80: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.htm
  df.loc[:, 'Major Crop (TONNES)'] = df.apply(major_crop_weights, axis=1)
C:\Users\learn\AppData\Local\Temp\ipykernel_11692\2253427923.py:81: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.htm
  df['Major Crop (TONNES)']= df['Major Crop (TONNES)']/1000
```

```
df.to_csv('clean_data.csv')
```

```
no_na = df.dropna()
no_na.to_csv('clean_no_na.csv')
```

no_na

	Project Year	ADVANCE Regional Code	Region	District	Community	Rural or Urban	Farmer ID	Gender	Age	Educatio Leve
0	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0001	FEMALE	65.0	NON
1	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0002	FEMALE	65.0	NON
2	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0003	MALE	60.0	NON
3	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0004	MALE	36.0	NON
4	2014	NRR	NORTHERN	SABOBA	BAKONDIBA	RURAL	1NRR0810NF001FR0005	FEMALE	36.0	NON
...	...	...	...	...	...	...	...	...	...	.
19189	2014	UER	UPPER EAST	PUSIGA	TANCHONGO NO. 1	RURAL	1UER0911NF002FR0111	MALE	38.0	NON
19190	2014	UER	UPPER EAST	PUSIGA	TANCHONGO NO. 1	RURAL	1UER0911NF002FR0112	MALE	38.0	NON
19191	2014	UER	UPPER EAST	PUSIGA	TANCHONGO NO. 1	RURAL	1UER0911NF002FR0113	MALE	40.0	NON
19192	2014	UER	UPPER EAST	PUSIGA	TANCHONGO NO. 1	RURAL	1UER0911NF002FR0114	MALE	70.0	NON
19193	2014	UER	UPPER EAST	PUSIGA	TANCHONGO NO. 1	RURAL	1UER0911NF002FR0115	MALE	49.0	NON

17618 rows × 15 columns

Recommnedations for Web Database

1. If direct entry to excel cannot be used, ensure all forms fields are completed and with legible information before submitting to excel sheet, otherwise make another information request.
2. Excel Data Validation can be used:
    1. Use validation cells in excel sheets, this will force user to only enter data in a specific format.
    2. Excel Data validation can also be used to prevent duplicate values when entering Farmer IDs for example.
    3. Row validation to prevent a data point(Row) being entered unless all attributes are filled.
    4. Drop down lists can be created for attributes that only need a specfic set of values e.g. Education Level
    5. Value range validation example, Age and Project year

1. Obtain/Include Longitude and Latitude fields for communities to improve mapping capabilties.
2. An ETL pipline can be built in Azure to automatically clean the data and load to a Database.
3. Connect the database to PowerBI for reporting to build required visuals.

Start coding or generate with AI.