In [1]:

```
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
```

In [2]:

```
pd.options.mode.chained_assignment = None
pd.options.display.max_columns = 999
from IPython.core.interactiveshell import InteractiveShell

InteractiveShell.ast_node_interactivity = "all"
```

Data Processing

In [3]:

```
import pandas as pd
df = pd.read_csv('kiva_loans.csv')
df.head()
```

Out[3]:

	id	funded_amount	loan_amount	activity	sector	use	country_code	country	region	currency	partne
0	653051	300.0	300.0	Fruits & Vegetables	Food	To buy seasonal, fresh fruits to sell.	PK	Pakistan	Lahore	PKR	2
1	653053	575.0	575.0	Rickshaw	Transportation	to repair and maintain the auto rickshaw used ...	PK	Pakistan	Lahore	PKR	2
2	653068	150.0	150.0	Transportation	Transportation	To repair their old cycle-van and buy another ...	IN	India	Maynaguri	INR	3
3	653063	200.0	200.0	Embroidery	Arts	to purchase an embroidery machine and a variet...	PK	Pakistan	Lahore	PKR	2
4	653084	400.0	400.0	Milk Sales	Food	to purchase one buffalo.	PK	Pakistan	Abdul Hakeem	PKR	2

In [4]:

```
df.shape
```

Out[4]:

```
(671205, 20)
```

In [5]:

```
mpiRegion_df=pd.read_csv("kiva_mpi_region_locations.csv")
mpiRegion_df.head()
```

Out[5]:

	LocationName	ISO	country	region	world_region	MPI	geo	lat	lon

	LocationName	ISO	country	region	world_region	MPI	geo	lat	lon
0	Badakhshan Afghanistan	AFG	Afghanistan	Badakhshan	South Asia	0.367	(36.7347725, 70.81199529999999)	36.734772	70.811995
1	Badghis, Afghanistan	AFG	Afghanistan	Badghis	South Asia	0.466	(35.1671339, 63.7695384)	35.167134	63.769538
2	Baghlan, Afghanistan	AFG	Afghanistan	Baghlan	South Asia	0.300	(35.8042947, 69.2877535)	35.804295	69.287754
3	Balkh, Afghanistan	AFG	Afghanistan	Balkh	South Asia	0.301	(36.7550603, 66.8975372)	36.755060	66.897537
4	Bamyan, Afghanistan	AFG	Afghanistan	Bamyan	South Asia	0.325	(34.8100067, 67.8212104)	34.810007	67.821210

In [6]:

```
mpiRegion_df.shape
```

Out[6]:

```
(2772, 9)
```

In [7]:

```
loantheme_df=pd.read_csv("loan_theme_ids.csv")
#loantheme_df.head()

print(loantheme_df)
```

```
             id    Loan Theme ID          Loan Theme Type  Partner ID
0        638631  a1050000000skGl                  General       151.0
1        640322  a1050000000skGl                  General       151.0
2        641006  a1050000002X1ij         Higher Education       160.0
3        641019  a1050000002X1ij         Higher Education       160.0
4        641594  a1050000002VbsW  Subsistence Agriculture       336.0
...         ...              ...                      ...         ...
779087  1444237  a1050000000wf0h                  General       136.0
779088  1444238  a1050000000wf0h                  General       136.0
779089  1444240  a1050000000wf0h                  General       136.0
779090  1444241  a1050000000wf22                  General       245.0
779091  1444243  a1050000000wf22                  General       245.0

[779092 rows x 4 columns]
```

In [8]:

```
loantheme_df.shape
```

Out[8]:

```
(779092, 4)
```

In [9]:

```
loan_themes_by_region_df=pd.read_csv("loan_themes_by_region.csv")
loan_themes_by_region_df.head()
```

Out[9]:

	Partner ID	Field Partner Name	sector	Loan Theme ID	Loan Theme Type	country	forkiva	region	geocode_old	ISO	number	amount
0	9	KREDIT Microfinance Institution	General Financial Inclusion	a1050000000slfi	Higher Education	Cambodia	No	Banteay Meanchey	(13.75, 103.0)	KHM	1	450
1	9	KREDIT Microfinance Institution	General Financial Inclusion	a10500000068jPe	Vulnerable Populations	Cambodia	No	Battambang Province	NaN	KHM	58	20275
2	9	KREDIT Microfinance Institution	General Financial Inclusion	a1050000000slfi	Higher Education	Cambodia	No	Battambang Province	NaN	KHM	7	9150
3	9	KREDIT Microfinance Institution	General Financial Inclusion	a10500000068jPe	Vulnerable Populations	Cambodia	No	Kampong Cham Province	(12.0, 105.5)	KHM	1383	604950

	Partner ID	Field Partner Name	sector	Loan Theme ID	Loan Theme Type	country	forkiva	region	geocode_old	ISO	number	amount
4	9	KREDIT Microfinance Institution	General Financial Inclusion	a1050000002X1Uu	Sanitation	Cambodia	No	Kampong Cham Province	(12.0, 105.5)	KHM	3	275

In [10]:

```
loan_themes_by_region_df.shape
```

Out[10]:

```
(15736, 21)
```

In [11]:

```
listed_dataframe=df.values.tolist()

#this matches with csv in list
#print(listed_dataframe)
```

In [12]:

```
#converting the lender count column into numeric column for further analysis
df["lender_count"] = pd.to_numeric(df["lender_count"], downcast="float")
```

In [13]:

```
dataframe_float=df.select_dtypes(include ='float64')
#print(dataframe_float)
dataframe_float_list=list(dataframe_float.columns)
print(dataframe_float_list)

dataframe_text=df.select_dtypes(exclude ='float64')
#print(dataframe_text)
dataframe_text_list=list(dataframe_text.columns)
print(dataframe_text_list)
```

```
['funded_amount', 'loan_amount', 'partner_id', 'term_in_months']
['id', 'activity', 'sector', 'use', 'country_code', 'country', 'region', 'currency',
'posted_time', 'disbursed_time', 'funded_time', 'lender_count', 'tags', 'borrower_genders',
'repayment_interval', 'date']
```

In [14]:

```
#kiva_loans
column_heads=list(df)
print(column_heads)
```

```
['id', 'funded_amount', 'loan_amount', 'activity', 'sector', 'use', 'country_code', 'country', 're
gion', 'currency', 'partner_id', 'posted_time', 'disbursed_time', 'funded_time', 'term_in_months',
'lender_count', 'tags', 'borrower_genders', 'repayment_interval', 'date']
```

In [15]:

```
dataframe_float=df.select_dtypes(include ='float')
dataframe_float_list=list(dataframe_float.columns)
print("column list with nominal values :" '\n')
print(dataframe_float_list)

dataframe_text=df.select_dtypes(exclude ='float64')
dataframe_text_list=list(dataframe_text.columns)
print( '\n' "column list with numerical values :" '\n')
print(dataframe_text_list)
```

```
column list with nominal values :

['funded_amount', 'loan_amount', 'partner_id', 'term_in_months']
```

```
column list with numerical values :

['id', 'activity', 'sector', 'use', 'country_code', 'country', 'region', 'currency',
'posted_time', 'disbursed_time', 'funded_time', 'lender_count', 'tags', 'borrower_genders',
'repayment_interval', 'date']
```

In [16]:

```
mpiRegion=list(mpiRegion_df)
print(mpiRegion)
```

```
['LocationName', 'ISO', 'country', 'region', 'world_region', 'MPI', 'geo', 'lat', 'lon']
```

In [17]:

```
loantheme=list(loantheme_df)
print(loantheme)
```

```
['id', 'Loan Theme ID', 'Loan Theme Type', 'Partner ID']
```

In [18]:

```
loan_themes_by_region=list(loan_themes_by_region_df)
print(loan_themes_by_region)
```

```
['Partner ID', 'Field Partner Name', 'sector', 'Loan Theme ID', 'Loan Theme Type', 'country',
'forkiva', 'region', 'geocode_old', 'ISO', 'number', 'amount', 'LocationName', 'geocode', 'names',
'geo', 'lat', 'lon', 'mpi_region', 'mpi_geo', 'rural_pct']
```

In [19]:

```
df[df.loan_amount == 100000]
```

Out[19]:

	id	funded_amount	loan_amount	activity	sector	use	country_code	country	region	currency	partner_id	p
70499	722883	100000.0	100000.0	Agriculture	Agriculture	create more than 300 jobs for women and farmer...	HT	Haiti	Les Cayes	USD	315.0	19:

In [20]:

```
df[(df.loan_amount >= 50000) & (df.funded_amount >= 50000)]
```

Out[20]:

	id	funded_amount	loan_amount	activity	sector	use	country_code	country	region
34196	687045	50000.0	50000.0	Renewable Energy Products	Retail	to buy and sell Barefoot Power's Solar Lightin...	PE	Peru	Arequipa
43182	695450	50000.0	50000.0	Renewable Energy Products	Retail	To buy and sell Barefoot Power's solar lightin...	KE	Kenya	Nairobi
53634	706146	50000.0	50000.0	Renewable Energy Products	Retail	To buy and sell Barefoot Power solar lighting.	UG	Uganda	Kampala
70499	722883	100000.0	100000.0	Agriculture	Agriculture	create more than 300 jobs for women and farmer...	HT	Haiti	Les Cayes

	id	funded_amount	loan_amount	activity	sector	use	country_code	country	region
126839	777718	50000.0	50000.0	Agriculture	Agriculture	to buy and plant resin producing pine trees. T...	MX	Mexico	Cherán
163727	812995	50000.0	50000.0	Agriculture	Agriculture	to fund its growing loan book and further deve...	KE	Kenya	Nairobi
210975	859201	50000.0	50000.0	Agriculture	Agriculture	To work with 17 farming cooperatives to proces...	RW	Rwanda	Kigali
223120	870901	50000.0	50000.0	Higher education costs	Education	to provide loans and career services for the l...	MX	Mexico	Mexico City
408295	1055043	50000.0	50000.0	Clothing	Clothing	to set up a garment social business that will ...	AL	Albania	Cerrik
408465	1055190	50000.0	50000.0	Construction	Construction	NaN	PE	Peru	NaN
447374	1107992	50000.0	50000.0	Agriculture	Agriculture	to increase smallholder farmers' incomes by bu...	UG	Uganda	Kampala
490191	1150277	50000.0	50000.0	Health	Health	To purchase raw materials in order to produce ...	GH	Ghana	Accra
492809	1152957	50000.0	50000.0	Agriculture	Agriculture	to expand weather, farming information and fin...	GH	Ghana	Accra
494470	1154951	50000.0	50000.0	Agriculture	Agriculture	To pay smallholder coffee farmers in rural Ken...	KE	Kenya	Nairobi
496715	1156972	50000.0	50000.0	Agriculture	Agriculture	to fund the harvest of seeds of 6,000 smallhol...	MG	Madagascar	Tsihombe
509048	1169175	50000.0	50000.0	Poultry	Agriculture	to purchase chicken feed & a delivery vehicle ...	TZ	Tanzania	Dar es Salaam
523634	1183609	50000.0	50000.0	Health	Health	to mitigate CO2 & household air pollution, whi...	MW	Malawi	NaN
523659	1183916	50000.0	50000.0	Electronics Sales	Retail	to train & equip 200 rural merchants in Mozamb...	MZ	Mozambique	Maputo
526100	1186897	50000.0	50000.0	Renewable Energy Products	Retail	to distribute 200+ innovative & affordable pay...	ZM	Zambia	Lusaka
538248	1198658	50000.0	50000.0	Agriculture	Agriculture	to enable 5,000 additional small-holder farmer...	KE	Kenya	Nanyuki
541006	1201708	50000.0	50000.0	Goods Distribution	Wholesale	to bolster logistics of affordable water distr...	HT	Haiti	Petion-Ville
544548	1205071	50000.0	50000.0	Health	Health	to provide community trauma services in South ...	SS	South Sudan	Juba
						to distribute			

	id	funded_amount	loan_amount	activity	sector	use	country_code	country	region
548513	1209262	50000.0	50000.0	Renewable Energy Products	Retail	to distribute solar home systems throughout ru...	ZW	Zimbabwe	Harare
563074	1223392	50000.0	50000.0	Renewable Energy Products	Retail	to provide life-changing clean cookstoves and ...	KE	Kenya	Nairobi
565733	1226382	50000.0	50000.0	Agriculture	Agriculture	to pay 600 farming families 100% above market ...	EC	Ecuador	Quito
583307	1245201	50000.0	50000.0	Agriculture	Agriculture	to support 800+ farmers by improving their pro...	GT	Guatemala	Quetzaltenango
586970	1247422	50000.0	50000.0	Renewable Energy Products	Retail	to generate income to over 600 fishermen in Ta...	TZ	Tanzania	MUSOMA
604502	1266423	50000.0	50000.0	Agriculture	Agriculture	to add value and jobs to the local economy by ...	BJ	Benin	Parakou
614869	1277100	50000.0	50000.0	Furniture Making	Manufacturing	create jobs through environmentally-friendly m...	KE	Kenya	Nairobi
614922	1277084	50000.0	50000.0	Water Distribution	Services	to set up 13 new clean water businesses in nor...	GH	Ghana	Tamale
618264	1280213	50000.0	50000.0	Farming	Agriculture	to provide income opportunities in remote Indo...	ID	Indonesia	Simeulue
621860	1283951	50000.0	50000.0	Renewable Energy Products	Retail	to distribute 400 pay-as-you-go solar home sys...	KE	Kenya	Nairobi
631904	1294308	50000.0	50000.0	Agriculture	Agriculture	double cashew nut export output and hire about...	CI	Cote D'Ivoire	Kolia

In [21]:

```
print(df['activity'].unique())
```

```
['Fruits & Vegetables' 'Rickshaw' 'Transportation' 'Embroidery'
 'Milk Sales' 'Services' 'Dairy' 'Beauty Salon' 'Manufacturing'
 'Food Production/Sales' 'Wholesale' 'General Store' 'Clothing Sales'
 'Poultry' 'Tailoring' 'Sewing' 'Bakery' 'Restaurant' 'Food Stall'
 'Farming' 'Construction Supplies' 'Personal Products Sales'
 'Home Products Sales' 'Natural Medicines' 'Fish Selling'
 'Education provider' 'Shoe Sales' 'Machinery Rental' 'Butcher Shop'
 'Pigs' 'Personal Expenses' 'Food Market' 'Cosmetics Sales'
 'Personal Housing Expenses' 'Retail' 'Energy' 'Grocery Store'
 'Construction' 'Agriculture' 'Motorcycle Transport' 'Charcoal Sales'
 'Food' 'Pharmacy' 'Fishing' 'Timber Sales' 'Cattle' 'Electronics Repair'
 'Electronics Sales' 'Vehicle' 'Cafe' 'Blacksmith'
 'Higher education costs' 'Used Clothing' 'Fuel/Firewood' 'Upholstery'
 'Catering' 'Animal Sales' 'Cereals' 'Vehicle Repairs' 'Arts'
 'Cloth & Dressmaking Supplies' 'Mobile Phones' 'Spare Parts' 'Clothing'
 'Metal Shop' 'Barber Shop' 'Furniture Making' 'Crafts' 'Home Energy'
 'Home Appliances' 'Wedding Expenses' 'Taxi' 'Secretarial Services'
 'Livestock' 'Property' 'Recycling' 'Farm Supplies' 'Auto Repair'
 'Beverages' 'Plastics Sales' 'Electrical Goods' 'Carpentry' 'Photography'
 'Jewelry' 'Bricks' 'Pub' 'Phone Use Sales' 'Water Distribution'
 'Paper Sales' 'Computers' 'Liquor Store / Off-License' 'Utilities'
 'Knitting' 'Weaving' 'Party Supplies' 'Medical Clinic' 'Internet Cafe'
 'Consumer Goods' 'Cement' 'Electrician' 'Primary/secondary school costs'
 'Veterinary Sales' 'Land Rental' 'Laundry' 'Call Center' 'Perfumes'
 'Hotel' 'Motorcycle Repair' 'Movie Tapes & DVDs' 'Quarrying'
 'Personal Medical Expenses' 'Bookstore' 'Decorations Sales'
 'Recycled Materials' 'Office Supplies' 'Souvenir Sales'
```

```
 'Renewable Energy Products' 'Health' 'Printing' 'Phone Repair'
 'Traveling Sales' 'Flowers' 'Bicycle Repair' 'Entertainment'
 'Phone Accessories' 'Hardware' 'Used Shoes' 'Music Discs & Tapes' 'Games'
 'Balut-Making' 'Textiles' 'Child Care' 'Goods Distribution' 'Florist'
 'Cobbler' 'Dental' 'Bookbinding' 'Cheese Making' 'Bicycle Sales'
 'Well digging' 'Technology' 'Musical Performance' 'Waste Management'
 'Film' 'Tourism' 'Musical Instruments' 'Religious Articles'
 'Machine Shop' 'Cleaning Services' 'Sporting Good Sales' 'Patchwork'
 'Funerals' 'Air Conditioning' 'Communications' 'Adult Care'
 'Landscaping / Gardening' 'Aquaculture' 'Beekeeping' 'Event Planning'
 'Celebrations' 'Computer' 'Personal Care Products' 'Mobile Transactions']
```

In [22]:

```
df_corr= df[['funded_amount', 'loan_amount', 'partner_id', 'term_in_months','lender_count']]
```

In [23]:

```
print("Correlation Matrix")
correlation_mat = df_corr.corr(method='pearson')
print(correlation_mat)
print()

def get_redundant_pairs(df):
    '''Get diagonal and lower triangular pairs of correlation matrix'''
    pairs_to_drop = set()
    cols = df.columns
    for i in range(0, df.shape[1]):
        for j in range(0, i+1):
            pairs_to_drop.add((cols[i], cols[j]))
    return pairs_to_drop

def get_top_abs_correlations(df, n=5):
    au_corr = df.corr().abs().unstack()
    labels_to_drop = get_redundant_pairs(df)
    au_corr = au_corr.drop(labels=labels_to_drop).sort_values(ascending=False)
    return au_corr[0:n]
def get_below_abs_correlations(df, n=5):
    au_corr = df.corr().abs().unstack()
    labels_to_drop = get_redundant_pairs(df)
    au_corr = au_corr.drop(labels=labels_to_drop).sort_values(ascending=True)
    return au_corr[0:n]

print("Top Absolute Correlations")
print(get_top_abs_correlations(df_corr,15))
```

```
Correlation Matrix
                funded_amount  loan_amount  partner_id  term_in_months  \
funded_amount        1.000000     0.945044   -0.075276        0.149310
loan_amount          0.945044     1.000000   -0.071251        0.184795
partner_id          -0.075276    -0.071251    1.000000        0.094878
term_in_months       0.149310     0.184795    0.094878        1.000000
lender_count         0.849168     0.798697   -0.008575        0.227283

                lender_count
funded_amount       0.849168
loan_amount         0.798697
partner_id         -0.008575
term_in_months      0.227283
lender_count        1.000000

Top Absolute Correlations
funded_amount   loan_amount       0.945044
                lender_count      0.849168
loan_amount     lender_count      0.798697
term_in_months  lender_count      0.227283
loan_amount     term_in_months    0.184795
funded_amount   term_in_months    0.149310
partner_id      term_in_months    0.094878
funded_amount   partner_id        0.075276
loan_amount     partner_id        0.071251
partner_id      lender_count      0.008575
dtype: float64
```

In [24]:

```
corr_pairs = correlation_mat.unstack().sort_values(kind="quicksort")
print('\n', corr_pairs)
```

```
 partner_id      funded_amount    -0.075276
funded_amount   partner_id       -0.075276
loan_amount     partner_id       -0.071251
partner_id      loan_amount      -0.071251
lender_count    partner_id       -0.008575
partner_id      lender_count     -0.008575
term_in_months  partner_id        0.094878
partner_id      term_in_months    0.094878
funded_amount   term_in_months    0.149310
term_in_months  funded_amount     0.149310
                loan_amount       0.184795
loan_amount     term_in_months    0.184795
term_in_months  lender_count      0.227283
lender_count    term_in_months    0.227283
                loan_amount       0.798697
loan_amount     lender_count      0.798697
lender_count    funded_amount     0.849168
funded_amount   lender_count      0.849168
                loan_amount       0.945044
loan_amount     funded_amount     0.945044
funded_amount   funded_amount     1.000000
partner_id      partner_id        1.000000
loan_amount     loan_amount       1.000000
term_in_months  term_in_months    1.000000
lender_count    lender_count      1.000000
dtype: float64
```

In [25]:

```
df_corr= corr_pairs.to_frame()
dist_df = df_corr.reset_index(level=[0,1])
attribute_df=dist_df.rename(columns={"level_0": "attribute_One", "level_1":
"attribute_Two",0:"Coeff"})
print(attribute_df)
```

```
     attribute_One   attribute_Two     Coeff
0       partner_id   funded_amount -0.075276
1    funded_amount      partner_id -0.075276
2      loan_amount      partner_id -0.071251
3       partner_id     loan_amount -0.071251
4     lender_count      partner_id -0.008575
5       partner_id    lender_count -0.008575
6   term_in_months      partner_id  0.094878
7       partner_id  term_in_months  0.094878
8    funded_amount  term_in_months  0.149310
9   term_in_months   funded_amount  0.149310
10  term_in_months     loan_amount  0.184795
11     loan_amount  term_in_months  0.184795
12  term_in_months    lender_count  0.227283
13    lender_count  term_in_months  0.227283
14    lender_count     loan_amount  0.798697
15     loan_amount    lender_count  0.798697
16    lender_count   funded_amount  0.849168
17   funded_amount    lender_count  0.849168
18   funded_amount     loan_amount  0.945044
19     loan_amount   funded_amount  0.945044
20   funded_amount   funded_amount  1.000000
21      partner_id      partner_id  1.000000
22     loan_amount     loan_amount  1.000000
23  term_in_months  term_in_months  1.000000
24    lender_count    lender_count  1.000000
```

In [26]:

```
dfcorr= corr_pairs.to_frame()

dist_df = dfcorr.reset_index(level=[0,1])
attribute_df=dist_df.rename(columns={"level_0": "attribute_One", "level_1":
```

```
"attribute_Two",0:"Coeff"})
#print(attribute_df)
attribute_df = attribute_df.query("attribute_One != attribute_Two")
print(attribute_df)
```

```
     attribute_One   attribute_Two     Coeff
0       partner_id   funded_amount -0.075276
1    funded_amount      partner_id -0.075276
2      loan_amount      partner_id -0.071251
3       partner_id     loan_amount -0.071251
4     lender_count      partner_id -0.008575
5       partner_id    lender_count -0.008575
6   term_in_months      partner_id  0.094878
7       partner_id  term_in_months  0.094878
8    funded_amount  term_in_months  0.149310
9   term_in_months   funded_amount  0.149310
10  term_in_months     loan_amount  0.184795
11     loan_amount  term_in_months  0.184795
12  term_in_months    lender_count  0.227283
13    lender_count  term_in_months  0.227283
14    lender_count     loan_amount  0.798697
15     loan_amount    lender_count  0.798697
16    lender_count   funded_amount  0.849168
17   funded_amount    lender_count  0.849168
18   funded_amount     loan_amount  0.945044
19     loan_amount   funded_amount  0.945044
```

In [27]:

```
result_df = attribute_df.drop_duplicates(subset=['Coeff'], keep='first')
print(result_df)
```

```
     attribute_One   attribute_Two     Coeff
0       partner_id   funded_amount -0.075276
2      loan_amount      partner_id -0.071251
4     lender_count      partner_id -0.008575
6   term_in_months      partner_id  0.094878
8    funded_amount  term_in_months  0.149310
10  term_in_months     loan_amount  0.184795
12  term_in_months    lender_count  0.227283
14    lender_count     loan_amount  0.798697
16    lender_count   funded_amount  0.849168
18   funded_amount     loan_amount  0.945044
```

In [28]:

```
threshold = 0.5
```

In [29]:

```
#if the threshold is more than 0.5 coeff
threshold_df= result_df[result_df.Coeff > threshold]
print(threshold_df)
```

```
    attribute_One  attribute_Two     Coeff
14   lender_count    loan_amount  0.798697
16   lender_count  funded_amount  0.849168
18  funded_amount    loan_amount  0.945044
```

In [30]:

```
feature_extraction_df=threshold_df.loc[result_df['attribute_Two'] == 'loan_amount']
print(feature_extraction_df)
secondAttribute = list(feature_extraction_df['attribute_One'])
print('\n','attribute result when one attribute is entered and a related attribute is asked for '
"\n")

print(secondAttribute)
```

```
    attribute_One attribute_Two     Coeff
14   lender_count    loan_amount  0.798697
```

```
14   lender_count   loan_amount  0.798697
18  funded_amount   loan_amount  0.945044

 attribute result when one attribute is entered and a related attribute is asked for

['lender_count', 'funded_amount']
```

In [31]:

```
#parsing all the rows of the dataset

def property(row):
    for row in listed_dataframe:
        if row !=listed_dataframe[0]:
            i = 0
            for prop in row :
                print("""
                    <owl:Class rdf:ID = "{0}":"{1}"
            """ .format(column_heads[i],row[i]))
                i = i + 1
            print("""</owl:Class>""")
            print(i)
#print (property(True))
```

In [32]:

```
#declaration of class and subclass
#basic elements
def subclassOf(row):
                    return"""
                    <owl:Class rdf:ID = class:"{0}"
                    <owl:Class rdf:ID = class:"{1}"
                    <owl:Class rdf:ID = class:"{2}"
                    <owl:Class rdf:ID = class:"{3}"
                <rdf:subClassOf rdf:resource="{4}"/>
                <owl:oneOf rdf:parseType="sector">
                <owl:Thing rdf:about="#"/>
                 </owl:oneOf>
            </owl:Class>""" .format(row[0], row[1], row[2], row[3], row[4])

def individual(type):
    return"""
    <owl:Thing rdf:about="{0}"/>
    <owl:Thing rdf:about="{1}"/>
    <owl:Thing rdf:about="{2}"/>
    </owl:oneOf>
</owl:Class> """.format(type[0],type[1],type[2])

#Equivalence between Classes and properties

def equivalentClass():
    return """"
    <owl:Class rdf:ID="value">
  <owl:equivalentClass rdf:resource="&value1;value2"/>
</owl:Class>""".format()


#ENUMERATED CLASSES

def disjointclasses(column_heads):
                    return """
                    <owlx:DisjointClasses>
                  <owlx:Class owlx:name="#{0}" />
                  <owlx:Class owlx:name="#{1}" />
                   <owlx:Class owlx:name="#{2}" />
                    <owlx:Class owlx:name="#{3}" />
                </owlx:DisjointClasses> """.format(column_heads[0],column_heads[1],column_heads[2],
column_heads[3])
def addProperty(prop):
                    return"""
              <owlx:ObjectProperty owlx:name="{}">
          <owlx:domain owlx:class="" />
          <owlx:range owlx:class="WineGrape" />
        </owlx:ObjectProperty>
        """.format(prop)
```

```
def functionalProperty(type):
    return """
    <owl:FunctionalProperty rdf:ID="type of sector">

    <rdfs:domain>

      <owl:Class>

        <owl:unionOf rdf:parseType="sector">

          <owl:Class rdf:about="#{0}"/>

          <owl:Class rdf:about="#{1}"/>

          <owl:Class rdf:about="#{2}"/>

        </owl:unionOf>

      </owl:Class>

    </rdfs:domain>

    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>

    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>

  </owl:FunctionalProperty>""".format(type[0],type[1],type[2])
```

In [33]:

```
text=("<rdf:RDF" '\n'
"xmlns:rdf=\"http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      "xmlns:rdfs=\"http://www.w3.org/2000/01/rdf-schema#"
      "xmlns:owl=\"http://www.w3.org/2002/07/owl#"
      "xmlns:dc=\"http://purl.org/dc/elements/1.1/"
       "xmlns:iris=\"http://www.w3.org/kiva#>"
       )

#owl header"
owlheader=("<owl:Ontology rdf:about=\"http://www.linkeddatatools.com/kivadataset>"
"<dc:title>The iris dataset Ontology</dc:title>"
"<dc:description>An ontology construction in python</dc:description>"
"</owl:Ontology>")
```

In [34]:

```
def addProperty(prop):
     return"""
    <owlx:ObjectProperty owlx:name="{}">
  <owlx:domain owlx:class="" />
  <owlx:range owlx:class="#definerange" />
</owlx:ObjectProperty>
""".format(prop)
```

In [35]:

```
def ontology():
    print(text,owlheader)
    for row in listed_dataframe:
                    print(property(row))
                    #print(indvidual(type))
                    #print(oneOf(type)),
                    #print(addProperty(prop)
                    #print(disjointclasses(column_heads))
```

In [36]:

```
#output of file
end_rdf=('\n'"</rdf:RDF>")
```

```
#print(ontology(),end_rdf)
```

In [ ]:

```
#output of file that is generated

import sys
file = open('kivafile.owl', 'w+')
sys.stdout = file

#print(ontology(),end_rdf)




file.close()
```

In [ ]:

```

```