EDA on Deloitte Case Study

In [1]:
```
# importing important libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import datetime
%matplotlib inline
```

Reading Excel data

In [2]:
```
# Importing CPI Excel file  (Please change the location of file according to your
df_cpi_monthly=pd.read_excel(r'C:\Users\bolt0\Data science\DeloitteCaseStudy\raw_
df_cpi_annual=pd.read_excel(r'C:\Users\bolt0\Data science\DeloitteCaseStudy\raw_c
```

In [3]:
```
# Importing Exchange rate file   (Please change the location of file according to
df_exchange_monthly=pd.read_excel(r'C:\Users\bolt0\Data science\DeloitteCaseStudy
df_exchange_annual=pd.read_excel(r'C:\Users\bolt0\Data science\DeloitteCaseStudy\
```

In [4]:
```
# Importing Exports Merchandise file   (Please change the location of file accord
df_exports_monthly=pd.read_excel(r'C:\Users\bolt0\Data science\DeloitteCaseStudy\
df_exports_quarterly=pd.read_excel(r'C:\Users\bolt0\Data science\DeloitteCaseStud
df_exports_annual=pd.read_excel(r'C:\Users\bolt0\Data science\DeloitteCaseStudy\r
```

Sample Data

In [5]:
```
# CPI data:
df_cpi_monthly.head()
```

Out[5]:

	Unnamed: 0	Afghanistan	Albania	United Arab Emirates	Armenia	Developing Asia	Antigua and Barbuda	Austria	Azerbaijan
0	1988M12	NaN	NaN	NaN	NaN	33.23211	NaN	63.07520	NaN
1	1989M01	NaN	NaN	NaN	NaN	34.38592	NaN	63.63636	NaN
2	1989M02	NaN	NaN	NaN	NaN	34.60887	NaN	63.74860	NaN
3	1989M03	NaN	NaN	NaN	NaN	35.11426	NaN	63.86083	NaN
4	1989M04	NaN	NaN	NaN	NaN	35.52607	NaN	64.08530	NaN

5 rows × 173 columns

In [6]:
```
df_cpi_monthly.tail()
```

Out[6]:

	Unnamed: 0	Afghanistan	Albania	United Arab Emirates	Armenia	Developing Asia	Antigua and Barbuda	Austria	Azerba
355	2018M07	NaN	116.6109	117.9010	127.5054	134.7460	NaN	117.7329	155.0
356	2018M08	NaN	117.0569	118.0702	128.2709	135.4813	NaN	117.7329	156.4
357	2018M09	NaN	117.2798	117.3614	128.6550	136.0586	NaN	118.6308	156.7
358	2018M10	NaN	117.0569	NaN	NaN	136.4408	NaN	NaN	157.0
359	2018M11	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N

5 rows × 173 columns

In [7]:
```
df_cpi_annual.head()
```

Out[7]:

	Unnamed: 0	Afghanistan	Albania	United Arab Emirates	Armenia	Developing Asia	Antigua and Barbuda	Austria	Azerbaijar
0	1989	NaN	NaN	NaN	NaN	35.24378	NaN	64.49682	NaN
1	1990	NaN	NaN	NaN	NaN	36.50166	NaN	66.60120	NaN
2	1991	NaN	NaN	NaN	NaN	38.06775	NaN	68.85522	NaN
3	1992	NaN	NaN	NaN	0.019203	40.46016	NaN	71.59559	NaN
4	1993	NaN	NaN	NaN	0.326774	45.49808	NaN	74.24242	NaN

5 rows × 173 columns

In [8]:
```
# Exchange Rate Data
df_exchange_monthly.head()
```

Out[8]:

	Period	Afghanistan	Angola	Albania	United Arab Emirates	Argenti0	Armenia	Developing Asia	Antigua and Barbuda	Au
0	1988M12	0.0	0.0	0.0	3.675	0.0	0.0	190.3592	0.0	1.1
1	1989M01	0.0	0.0	0.0	3.675	0.0	0.0	191.0076	0.0	1.1
2	1989M02	0.0	0.0	0.0	3.675	0.0	0.0	191.9781	0.0	1.1
3	1989M03	0.0	0.0	0.0	3.675	0.0	0.0	193.4470	0.0	1.2
4	1989M04	0.0	0.0	0.0	3.675	0.0	0.0	195.0895	0.0	1.2

5 rows × 201 columns

In [9]:
```
df_exchange_monthly.tail()
```

Out[9]:

	Period	Afghanistan	Angola	Albania	United Arab Emirates	Argenti0	Armenia	Developing Asia	Antigua and Barbuda
355	2018M07	0.0	253.7293	107.8486	3.673	27.59330	0.0	1411.209	2.69954
356	2018M08	0.0	266.1822	108.9400	3.673	30.24126	0.0	1425.079	2.70152
357	2018M09	0.0	286.0590	108.5063	3.673	38.64783	0.0	1448.644	2.70025
358	2018M10	0.0	303.7345	109.0561	3.673	37.06159	0.0	1473.699	2.70434
359	2018M11	0.0	309.2843	109.4100	3.673	35.60083	0.0	1445.686	2.71000

5 rows × 201 columns

In [10]:
```
df_exchange_annual.head()
```

Out[10]:

	Period	Afghanistan	Angola	Albania	United Arab Emirates	Argenti0	Armenia	Developing Asia	Antigua and Barbuda	Au
0	1989	0.0	0.000	0.0000	3.675000	0.000000	0.0	213.1695	0.00000	1.
1	1990	0.0	0.000	0.0000	3.674964	0.000000	0.0	242.0275	0.00000	1.
2	1991	0.0	0.000	0.0000	3.675188	0.000000	0.0	282.4442	0.00000	1.
3	1992	0.0	0.000	0.0000	3.673874	0.991757	0.0	290.6883	0.00000	1.
4	1993	0.0	4903.561	110.0993	3.675296	0.998979	0.0	299.6200	2.69682	1.

5 rows × 201 columns

In [11]:
```
#Exports Merchandise data
df_exports_monthly.head()
```

Out[11]:

	Unnamed: 0	Argentina	Developing Asia	Australia	Belgium	Brazil	Canada	Chile	China	Czech Republi
0	1991M01	1265.303	3313.754	5502.346	NaN	4896.404	NaN	NaN	NaN	NaN
1	1991M02	1518.856	NaN	5708.852	NaN	3887.668	NaN	NaN	NaN	NaN
2	1991M03	1632.372	NaN	6890.402	NaN	4384.693	NaN	NaN	NaN	NaN
3	1991M04	1884.038	NaN	6524.655	NaN	5119.613	NaN	NaN	NaN	NaN
4	1991M05	2302.835	NaN	7145.694	NaN	4917.469	NaN	NaN	NaN	NaN

5 rows × 55 columns

In [12]:
```
df_exports_annual.head()
```

Out[12]:

	Unnamed: 0	Argentina	Developing Asia	Australia	Belgium	Brazil	Canada	Chile	China	Czec Republi
0	1991	22191.58	NaN	80560.51	NaN	53953.57	NaN	NaN	NaN	Nal
1	1992	22254.20	NaN	85549.90	NaN	63269.99	NaN	NaN	NaN	Nal
2	1993	24153.33	NaN	90912.32	NaN	73589.75	NaN	NaN	NaN	Nal
3	1994	28583.86	NaN	96492.75	NaN	75080.52	NaN	NaN	NaN	Nal
4	1995	35332.53	NaN	99382.24	296008.8	70497.63	NaN	NaN	NaN	Nal

5 rows × 55 columns

In [13]:
```
df_exports_quarterly.tail()
```

Out[13]:

	Unnamed: 0	Argentina	Developing Asia	Australia	Belgium	Brazil	Canada	Chile	China
106	2017Q3	16770.97	720194.3	59437.88	90699.26	61736.42	115.8469	17700.18	531477.9
107	2017Q4	14745.50	735916.6	58454.59	88748.78	55125.52	121.1300	18825.41	552142.0
108	2018Q1	14262.01	NaN	56989.81	85198.43	54994.27	116.7996	19143.87	NaN
109	2018Q2	14454.23	NaN	60143.25	84912.30	59141.24	126.2617	18995.88	NaN
110	2018Q3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

5 rows × 55 columns

Missing Data:

CPI data

In [14]:
```
df_cpi_monthly.isnull().sum()
```

Out[14]:
```
Unnamed: 0                0
Afghanistan             190
Albania                  62
United Arab Emirates    219
Armenia                  50
                       ...
World (WBG members)       2
Samoa                    58
Yemen, Rep.             180
South Africa              2
Zambia                    1
Length: 173, dtype: int64
```

In [15]:
```
df_cpi_monthly.isnull().sum().sum()
```

Out[15]:
```
11594
```

We see that there are **11594 Null Entries** (missing) in **cpi_monthly dataframe**.

In [16]:
```
df_cpi_annual.isnull().sum().sum()
```

Out[16]:
```
910
```

There are **1083** null entries in **cpi_annual** data

We need to handle these missing values, so that our data is **Complete , cleaned and Structured**
.

Foe Null values we are considering that there were **No Services provided by Deloitte for that period**. So we have to fill those null values with **Zeroes (0)**.

In [17]:
```
# Filling Null Values with zeroes:
df_cpi_monthly=df_cpi_monthly.fillna(0)
df_cpi_annual=df_cpi_annual.fillna(0)
```

In [18]:
```
# Now we will se if the values are filled or not:
df_cpi_monthly.isnull().sum()
```

Out[18]:
```
Unnamed: 0              0
Afghanistan             0
Albania                 0
United Arab Emirates    0
Armenia                 0
                       ..
World (WBG members)     0
Samoa                   0
Yemen, Rep.             0
South Africa            0
Zambia                  0
Length: 173, dtype: int64
```

In [19]:
```
df_cpi_monthly.isnull().sum().sum()
```

Out[19]:
```
0
```

we can seee taht there are no null values left anymore in this dataframe. Similarly we will check this for other CPI dataframe.

In [20]:
```
df_cpi_annual.isnull().sum().sum()
```

Out[20]:
```
0
```

In [21]:
```
df_cpi_monthly.head()
```

Out[21]:

	Unnamed: 0	Afghanistan	Albania	United Arab Emirates	Armenia	Developing Asia	Antigua and Barbuda	Austria	Azerbaijan
0	1988M12	0.0	0.0	0.0	0.0	33.23211	0.0	63.07520	0.0
1	1989M01	0.0	0.0	0.0	0.0	34.38592	0.0	63.63636	0.0
2	1989M02	0.0	0.0	0.0	0.0	34.60887	0.0	63.74860	0.0
3	1989M03	0.0	0.0	0.0	0.0	35.11426	0.0	63.86083	0.0
4	1989M04	0.0	0.0	0.0	0.0	35.52607	0.0	64.08530	0.0

5 rows × 173 columns

In [22]:
```
df_cpi_annual.head()
```

Out[22]:

	Unnamed: 0	Afghanistan	Albania	United Arab Emirates	Armenia	Developing Asia	Antigua and Barbuda	Austria	Azerbaijan
0	1989	0.0	0.0	0.0	0.000000	35.24378	0.0	64.49682	0.0
1	1990	0.0	0.0	0.0	0.000000	36.50166	0.0	66.60120	0.0
2	1991	0.0	0.0	0.0	0.000000	38.06775	0.0	68.85522	0.0
3	1992	0.0	0.0	0.0	0.019203	40.46016	0.0	71.59559	0.0
4	1993	0.0	0.0	0.0	0.326774	45.49808	0.0	74.24242	0.0

5 rows × 173 columns

Exchange Data

In [23]:
```
df_exchange_monthly.isnull().sum()
```

Out[23]:
```
Period                 0
Afghanistan            0
Angola                 0
Albania                0
United Arab Emirates   0
                      ..
Samoa                  0
Yemen, Rep.            0
South Africa           0
Zambia                 0
Zimbabwe               0
Length: 201, dtype: int64
```

In [24]:
```
df_exchange_monthly.isnull().sum().sum()
```

Out[24]:
```
0
```

In [25]:
```
df_exchange_annual.isnull().sum().sum()
```

Out[25]:
```
0
```

We observe that there are **no Null values** in the Exchange Dataframe , so no need of handling here. So we can now move to the next dataset.

Exports Data

In [26]:
```
df_exports_monthly.isnull().sum()
```

Out[26]:
```
Unnamed: 0                              0
Argentina                               3
Developing Asia                       175
Australia                               3
Belgium                                51
Brazil                                  2
Canada                                 75
Chile                                 147
China                                 175
Czech Republic                         86
Germany                                 2
Developing Countries                  175
Denmark                                50
East Asia & Pacific developing        175
Europe & Central Asia developing      183
Spain                                   3
Estonia                                38
Finland                                50
France                                 98
```

In [27]:
```
df_exports_monthly.isnull().sum().sum()
```

Out[27]:
```
3821
```

In [28]:
```
df_exports_quarterly.isnull().sum().sum()
```

Out[28]:
```
1298
```

In [29]:
```
df_exports_annual.isnull().sum().sum()
```

Out[29]:
```
368
```

There are **3821** null values in **monthly exports** dataset , **1353** nul values in **quarterly exports** and **423** null values in **annual exports**.

therefore we replace these values with zero

In [30]:
```
df_exports_monthly=df_exports_monthly.fillna(0)
df_exports_quarterly=df_exports_quarterly.fillna(0)
df_exports_annual=df_exports_annual.fillna(0)
```

In [31]:
```
df_exports_monthly.head()
```

Out[31]:

	Unnamed: 0	Argentina	Developing Asia	Australia	Belgium	Brazil	Canada	Chile	China	Czech Republi
0	1991M01	1265.303	3313.754	5502.346	0.0	4896.404	0.0	0.0	0.0	0.(
1	1991M02	1518.856	0.000	5708.852	0.0	3887.668	0.0	0.0	0.0	0.(
2	1991M03	1632.372	0.000	6890.402	0.0	4384.693	0.0	0.0	0.0	0.(
3	1991M04	1884.038	0.000	6524.655	0.0	5119.613	0.0	0.0	0.0	0.(
4	1991M05	2302.835	0.000	7145.694	0.0	4917.469	0.0	0.0	0.0	0.(

5 rows × 55 columns

In [32]:
```
df_exports_annual.head()
```

Out[32]:

	Unnamed: 0	Argentina	Developing Asia	Australia	Belgium	Brazil	Canada	Chile	China	Czec Republi
0	1991	22191.58	0.0	80560.51	0.0	53953.57	0.0	0.0	0.0	0.
1	1992	22254.20	0.0	85549.90	0.0	63269.99	0.0	0.0	0.0	0.
2	1993	24153.33	0.0	90912.32	0.0	73589.75	0.0	0.0	0.0	0.
3	1994	28583.86	0.0	96492.75	0.0	75080.52	0.0	0.0	0.0	0.
4	1995	35332.53	0.0	99382.24	296008.8	70497.63	0.0	0.0	0.0	0.

5 rows × 55 columns

In [33]:
```
df_exports_quarterly.tail()
```

Out[33]:

	Unnamed: 0	Argentina	Developing Asia	Australia	Belgium	Brazil	Canada	Chile	C
106	2017Q3	16770.97	720194.3	59437.88	90699.26	61736.42	115.8469	17700.18	5314
107	2017Q4	14745.50	735916.6	58454.59	88748.78	55125.52	121.1300	18825.41	5521
108	2018Q1	14262.01	0.0	56989.81	85198.43	54994.27	116.7996	19143.87	
109	2018Q2	14454.23	0.0	60143.25	84912.30	59141.24	126.2617	18995.88	
110	2018Q3	0.00	0.0	0.00	0.00	0.00	0.0000	0.00	

5 rows × 55 columns

We have replaced all the misiing data and now our data has no null values left in the data . So now we can move to further analysis of data.

Duplicate Data

We will noew check if there is any duplicate data in the files

In [34]:
```
df_cpi_monthly.duplicated().sum()
```

Out[34]: 0

In [35]:
```
df_cpi_annual.duplicated().sum()
```

Out[35]: 0

In [36]:
```
df_exchange_annual.duplicated().sum()
```

Out[36]: 0

In [37]:
```
df_exchange_monthly.duplicated().sum()
```

Out[37]: 0

In [38]:
```
df_exports_annual.duplicated().sum()
```

Out[38]: 0

In [39]:
```
df_exports_monthly.duplicated().sum()
```

Out[39]: 0

In [40]:
```
df_exports_quarterly.duplicated().sum()
```

Out[40]:
```
0
```

There is no duplicate data in all the dataframes.

CPI DATA ANALYSIS:

In [41]:
```
df_cpi_monthly.columns
```

Out[41]:
```
Index(['Unnamed: 0', 'Afghanistan', 'Albania', 'United Arab Emirates',
       'Armenia', 'Developing Asia', 'Antigua and Barbuda', 'Austria',
       'Azerbaijan', 'Burundi',
       ...
       'Uruguay', 'United States', 'St. Vincent and the Grenadines',
       'Venezuela, RB', 'Vietnam', 'World (WBG members)', 'Samoa',
       'Yemen, Rep.', 'South Africa', 'Zambia'],
      dtype='object', length=173)
```

In [42]:
```
df_cpi_monthly.head()
```

Out[42]:

	Unnamed: 0	Afghanistan	Albania	United Arab Emirates	Armenia	Developing Asia	Antigua and Barbuda	Austria	Azerbaijan
0	1988M12	0.0	0.0	0.0	0.0	33.23211	0.0	63.07520	0.0
1	1989M01	0.0	0.0	0.0	0.0	34.38592	0.0	63.63636	0.0
2	1989M02	0.0	0.0	0.0	0.0	34.60887	0.0	63.74860	0.0
3	1989M03	0.0	0.0	0.0	0.0	35.11426	0.0	63.86083	0.0
4	1989M04	0.0	0.0	0.0	0.0	35.52607	0.0	64.08530	0.0

5 rows × 173 columns

If we look at the heading of the first column it is named **Unnamed 0** but since it is representing months of years so we rename it as **Year** and convert into into YYYY-MM_DD format

In [43]:
```
#Renaming

df_cpi_monthly.rename(columns = {'Unnamed: 0':'Year'} , inplace = True)
```

In [44]:
```
df_cpi_monthly.dtypes
```

Out[44]:
```
Year                    object
Afghanistan            float64
Albania                float64
United Arab Emirates   float64
Armenia                float64
                         ...
World (WBG members)    float64
Samoa                  float64
Yemen, Rep.            float64
South Africa           float64
Zambia                 float64
Length: 173, dtype: object
```

In [45]:
```
# conversion into YYYY-MM-DD Format.

# For YYYY we have selected from 0th index to 4th index
l1 = [ ]
for i in range(len(df_cpi_monthly)):
    l1.append(df_cpi_monthly['Year'][i][0:4])

# For MM format we have selected 5th and 6th index as they represent the month nu
l2 = [ ]
for i in range(len(df_cpi_monthly)):
    l2.append(df_cpi_monthly['Year'][i][5:7])

# Now to add YYYY and MM together we create another list
l3 = [ ]
for i in range(len(l1)):
    l3.append(l1[i]+'-'+l2[i])

# Now adding DD , for this we consider 01 i.e. initial date and we also add in l3
l4 = [ ]
for i in range(len(l3)):
    l4.append(datetime.datetime.strptime(l3[i], '%Y-%m').date())

# Changing the Year column:
df_cpi_monthly['Year'] = l4
```

In [46]:
```
df_cpi_monthly.head()
```

Out[46]:

	Year	Afghanistan	Albania	United Arab Emirates	Armenia	Developing Asia	Antigua and Barbuda	Austria	Azerbaijan	Bu
0	1988-12-01	0.0	0.0	0.0	0.0	33.23211	0.0	63.07520	0.0	
1	1989-01-01	0.0	0.0	0.0	0.0	34.38592	0.0	63.63636	0.0	
2	1989-02-01	0.0	0.0	0.0	0.0	34.60887	0.0	63.74860	0.0	
3	1989-03-01	0.0	0.0	0.0	0.0	35.11426	0.0	63.86083	0.0	
4	1989-04-01	0.0	0.0	0.0	0.0	35.52607	0.0	64.08530	0.0	

5 rows × 173 columns

In [47]:
```
# Now for annual dataset
df_cpi_annual.columns
```

Out[47]:
```
Index(['Unnamed: 0', 'Afghanistan', 'Albania', 'United Arab Emirates',
       'Armenia', 'Developing Asia', 'Antigua and Barbuda', 'Austria',
       'Azerbaijan', 'Burundi',
       ...
       'Uruguay', 'United States', 'St. Vincent and the Grenadines',
       'Venezuela, RB', 'Vietnam', 'World (WBG members)', 'Samoa',
       'Yemen, Rep.', 'South Africa', 'Zambia'],
      dtype='object', length=173)
```

In [48]:
```
df_cpi_annual.head()
```

Out[48]:

	Unnamed: 0	Afghanistan	Albania	United Arab Emirates	Armenia	Developing Asia	Antigua and Barbuda	Austria	Azerbaijan
0	1989	0.0	0.0	0.0	0.000000	35.24378	0.0	64.49682	0.0
1	1990	0.0	0.0	0.0	0.000000	36.50166	0.0	66.60120	0.0
2	1991	0.0	0.0	0.0	0.000000	38.06775	0.0	68.85522	0.0
3	1992	0.0	0.0	0.0	0.019203	40.46016	0.0	71.59559	0.0
4	1993	0.0	0.0	0.0	0.326774	45.49808	0.0	74.24242	0.0

5 rows × 173 columns

Similarly in annual dataset we have Unnamed: 0 column so we need to rename it also using the same method , whereas it's format is all right.

In [49]:
```
# Renaming and changing datatype:

df_cpi_annual.rename(columns = {'Unnamed: 0':'Year'} , inplace = True)
```

In [50]:
```
df_cpi_annual.dtypes
```

Out[50]:
```
Year                     int64
Afghanistan            float64
Albania                float64
United Arab Emirates   float64
Armenia                float64
                         ...
World (WBG members)    float64
Samoa                  float64
Yemen, Rep.            float64
South Africa           float64
Zambia                 float64
Length: 173, dtype: object
```

In [51]:
```
df_cpi_annual.head()
```

Out[51]:

	Year	Afghanistan	Albania	United Arab Emirates	Armenia	Developing Asia	Antigua and Barbuda	Austria	Azerbaijan	Bu
0	1989	0.0	0.0	0.0	0.000000	35.24378	0.0	64.49682	0.0	0.
1	1990	0.0	0.0	0.0	0.000000	36.50166	0.0	66.60120	0.0	0.
2	1991	0.0	0.0	0.0	0.000000	38.06775	0.0	68.85522	0.0	0.
3	1992	0.0	0.0	0.0	0.019203	40.46016	0.0	71.59559	0.0	0.
4	1993	0.0	0.0	0.0	0.326774	45.49808	0.0	74.24242	0.0	13.

5 rows × 173 columns

In [52]:
```
#Now that we have remaned the column name in the dataframe we will now wee some b
```

In [53]:
```
# For Monthly data:
df_cpi_monthly.describe()
```

Out[53]:

	Afghanistan	Albania	United Arab Emirates	Armenia	Developing Asia	Antigua and Barbuda	Austria	Azer
count	360.000000	360.000000	360.000000	360.000000	360.000000	360.000000	360.000000	360.0
mean	54.417925	71.816240	40.796141	71.626969	82.725954	59.022299	90.346675	58.8
std	60.570082	38.598724	51.143928	42.367377	29.551206	48.955193	16.699998	52.6
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
25%	0.000000	44.788180	0.000000	52.975215	67.069325	0.000000	79.236810	0.0
50%	0.000000	83.277595	0.000000	74.016155	77.382030	85.883185	88.945005	54.3
75%	115.508975	102.675575	101.618700	108.203950	106.941925	101.146975	104.461275	109.8
max	155.742000	118.952100	118.800300	133.409700	136.440800	114.096100	118.630800	157.4

8 rows × 172 columns

In [54]:
```
# For annual data:
df_cpi_annual.describe()
```

Out[54]:

	Year	Afghanistan	Albania	United Arab Emirates	Armenia	Developing Asia	Antigua and Barbuda	
count	30.000000	30.000000	30.000000	30.000000	30.000000	30.000000	30.000000	30.
mean	2003.500000	58.225701	72.469344	41.779188	72.716573	83.382088	59.022295	91.
std	8.803408	62.524595	38.944382	52.287521	42.774219	29.799967	49.718768	15.
min	1989.000000	0.000000	0.000000	0.000000	0.000000	35.243780	0.000000	64.
25%	1996.250000	0.000000	49.814193	0.000000	55.949440	68.018888	0.000000	79.
50%	2003.500000	33.729940	83.579525	0.000000	75.444235	78.041940	86.132425	89.
75%	2010.750000	116.063100	102.657000	101.353750	108.677625	106.545325	100.595988	103.
max	2018.000000	154.484800	117.558500	117.965600	130.682700	134.720800	113.896300	117.

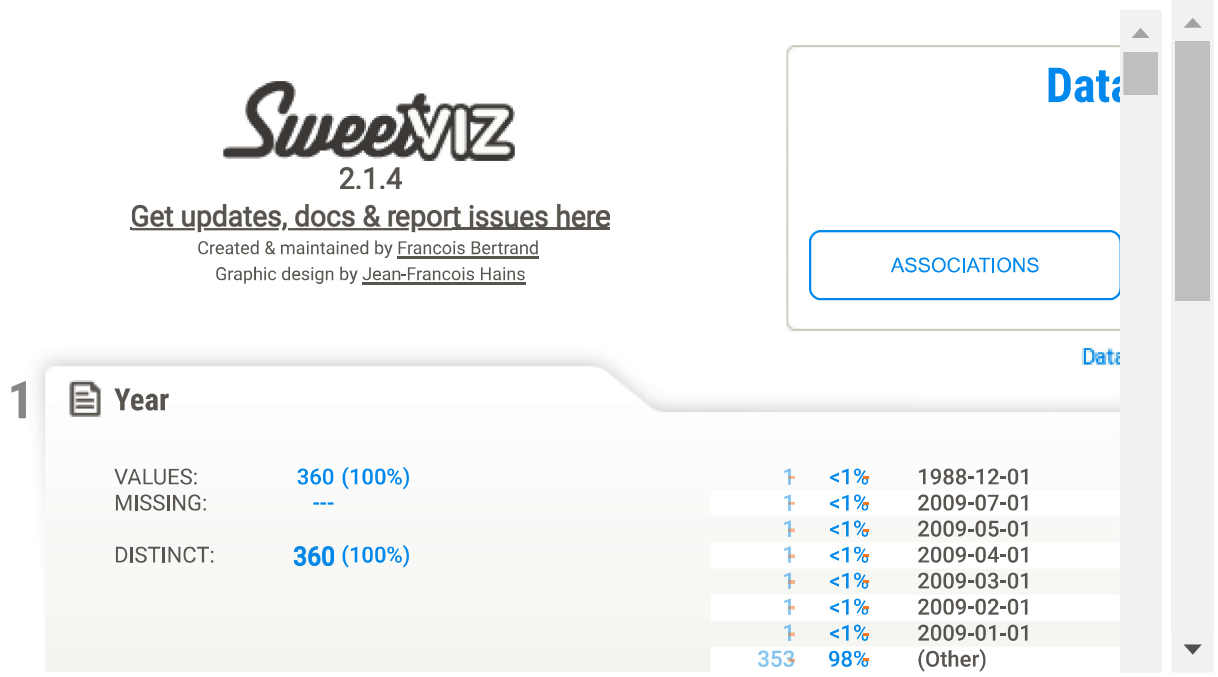
8 rows × 173 columns

In [55]:
```
# Creating Sweetviz report for automated analysis:

import sweetviz as sv
data1 = sv.analyze(df_cpi_monthly, pairwise_analysis= 'off')
```

Feature: Zambia [100%] 00:50 -> (00:00 left)

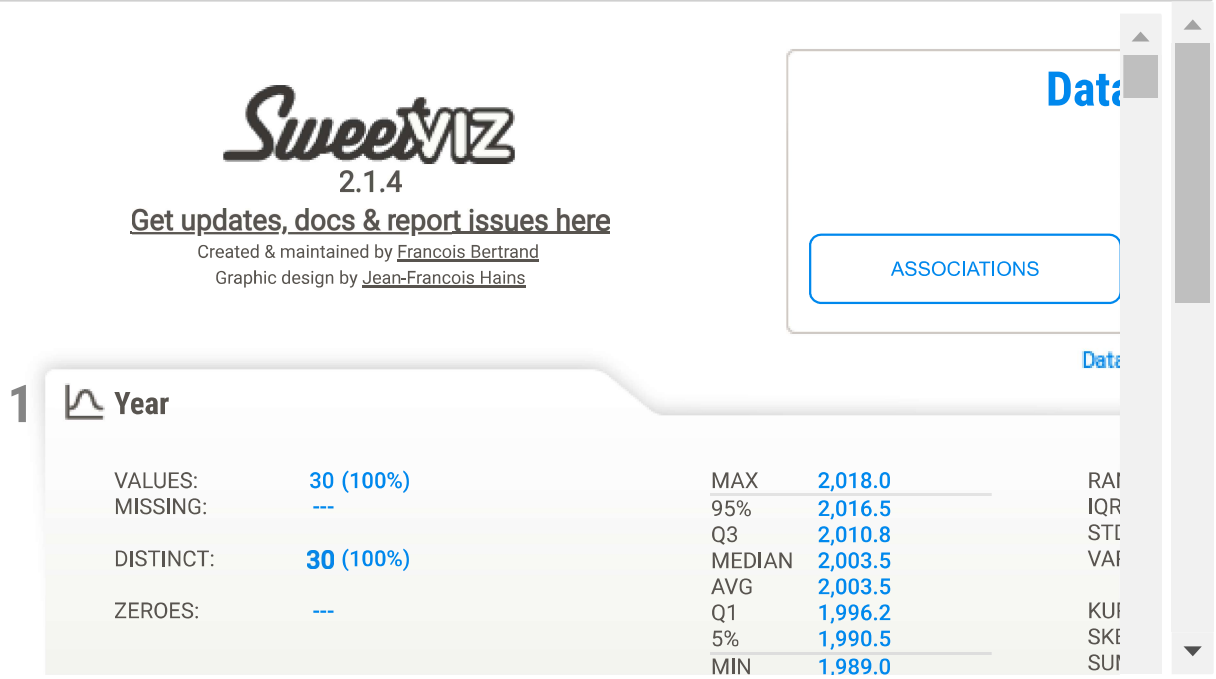
In [56]:
```
data1.show_notebook()
```



In [57]:
```
data2 = sv.analyze(df_cpi_annual, pairwise_analysis= 'off')
```

Feature: Zambia [100%] 00:50 -> (00:00 left)

In [58]:
```
data2.show_notebook()
```



Since we have performed basic analysis in data and also have converted it into **Structured Form** , so now we **save** it as an **Excel file** in the **Structured Data** folder and will use this structured data for creation of Dashboard.

In [59]:
```
# saving structured data as excel file:     (Please change the location of file d

with pd.ExcelWriter(r'C:\Users\bolt0\Data science\DeloitteCaseStudy\Structured Da
    df_cpi_monthly.to_excel(writer, sheet_name='monthly')
    df_cpi_annual.to_excel(writer, sheet_name='annual')
```

Exchange Rate Data Analysis

In [60]:
```
list(df_exchange_monthly.columns)
```

Out[60]:
```
['Period',
 'Afghanistan',
 'Angola',
 'Albania',
 'United Arab Emirates',
 'Argenti0',
 'Armenia',
 'Developing Asia',
 'Antigua and Barbuda',
 'Australia',
 'Austria',
 'Azerbaijan',
 'Burundi',
 'Belgium',
 'Benin',
 'Burki0 Faso',
 'Bangladesh',
 'Bulgaria',
 'Bahrain',
```

In [61]:
```
list(df_exchange_annual.columns)
```

Out[61]:
```
['Period',
 'Afghanistan',
 'Angola',
 'Albania',
 'United Arab Emirates',
 'Argenti0',
 'Armenia',
 'Developing Asia',
 'Antigua and Barbuda',
 'Australia',
 'Austria',
 'Azerbaijan',
 'Burundi',
 'Belgium',
 'Benin',
 'Burki0 Faso',
 'Bangladesh',
 'Bulgaria',
 'Bahrain',
```

Similarly linke CPI dataframes here also we need to convert Period into Year column and change as 5__YYYY-MM-DD__ Format in **monthly dataframe** . Also observe that the name of some countries are misspelt such as **Argentina** with **Argenti0** , **Burkina Faso** with **Burki0 Faso** etc. so we need to make that also correct.

In [62]:
```
df_exchange_monthly.head()
```

Out[62]:

	Period	Afghanistan	Angola	Albania	United Arab Emirates	Argenti0	Armenia	Developing Asia	Antigua and Barbuda	Au
0	1988M12	0.0	0.0	0.0	3.675	0.0	0.0	190.3592	0.0	1.1
1	1989M01	0.0	0.0	0.0	3.675	0.0	0.0	191.0076	0.0	1.1
2	1989M02	0.0	0.0	0.0	3.675	0.0	0.0	191.9781	0.0	1.1
3	1989M03	0.0	0.0	0.0	3.675	0.0	0.0	193.4470	0.0	1.2
4	1989M04	0.0	0.0	0.0	3.675	0.0	0.0	195.0895	0.0	1.2

5 rows × 201 columns

In [63]:
```
# Renaming Period Column and Changing it's format

df_exchange_monthly.rename(columns = {'Period':'Year'} , inplace = True)
```

In [64]:
```
# conversion into YYYY-MM-DD Format.

# For YYYY we have selected from 0th index to 4th index
l1 = [ ]
for i in range(len(df_exchange_monthly)):
    l1.append(df_exchange_monthly['Year'][i][0:4])

# For MM format we have selected 5th and 6th index as they represent the month nu
l2 = [ ]
for i in range(len(df_exchange_monthly)):
    l2.append(df_exchange_monthly['Year'][i][5:7])

# Now to add YYYY and MM together we create another list
l3 = [ ]
for i in range(len(l1)):
    l3.append(l1[i]+'-'+l2[i])

# Now adding DD , for this we consider 01 i.e. initial date and we also add in l3
l4 = [ ]
for i in range(len(l3)):
    l4.append(datetime.datetime.strptime(l3[i], '%Y-%m').date())

# Changing the Year column:
df_exchange_monthly['Year'] = l4
```

In [65]:
```
df_exchange_monthly
```

Out[65]:

	Year	Afghanistan	Angola	Albania	United Arab Emirates	Argenti0	Armenia	Developing Asia	Antigua and Barbuda
0	1988-12-01	0.0	0.0000	0.0000	3.675	0.00000	0.0	190.3592	0.000000
1	1989-01-01	0.0	0.0000	0.0000	3.675	0.00000	0.0	191.0076	0.000000
2	1989-02-01	0.0	0.0000	0.0000	3.675	0.00000	0.0	191.9781	0.000000
3	1989-03-01	0.0	0.0000	0.0000	3.675	0.00000	0.0	193.4470	0.000000
4	1989-04-01	0.0	0.0000	0.0000	3.675	0.00000	0.0	195.0895	0.000000
...	...	...	...	...	...	...	...	...	...
355	2018-07-01	0.0	253.7293	107.8486	3.673	27.59330	0.0	1411.2090	2.699545
356	2018-08-01	0.0	266.1822	108.9400	3.673	30.24126	0.0	1425.0790	2.701522
357	2018-09-01	0.0	286.0590	108.5063	3.673	38.64783	0.0	1448.6440	2.700250
358	2018-10-01	0.0	303.7345	109.0561	3.673	37.06159	0.0	1473.6990	2.704348
359	2018-11-01	0.0	309.2843	109.4100	3.673	35.60083	0.0	1445.6860	2.710000

360 rows × 201 columns

In [66]:
```
# Now renaming Columns with misspelt name :

# In the column names at place of 'na' it is repolce with '0' so we need to chang

l  = list(df_exchange_monthly.columns)   # first we create a list with all columr
```

In [67]:
```
# now we create another list that will contain correct column  names

new_list = [ ]
for item in l:
    if '0' in item:
        new_list.append(str(item).replace('0' , 'na'))
    else:
        new_list.append(str(item))
```

In [68]:
```
new_list
```

Out[68]:
```
['Year',
 'Afghanistan',
 'Angola',
 'Albania',
 'United Arab Emirates',
 'Argentina',
 'Armenia',
 'Developing Asia',
 'Antigua and Barbuda',
 'Australia',
 'Austria',
 'Azerbaijan',
 'Burundi',
 'Belgium',
 'Benin',
 'Burkina Faso',
 'Bangladesh',
 'Bulgaria',
 'Bahrain',
```

In [69]:
```
# Now we replace the column names:

df_exchange_monthly.columns = new_list
```

In [70]:
```
df_exchange_monthly.head()
```

Out[70]:

	Year	Afghanistan	Angola	Albania	United Arab Emirates	Argentina	Armenia	Developing Asia	Antigua and Barbuda	Austı
0	1988-12-01	0.0	0.0	0.0	3.675	0.0	0.0	190.3592	0.0	1.164
1	1989-01-01	0.0	0.0	0.0	3.675	0.0	0.0	191.0076	0.0	1.148
2	1989-02-01	0.0	0.0	0.0	3.675	0.0	0.0	191.9781	0.0	1.169
3	1989-03-01	0.0	0.0	0.0	3.675	0.0	0.0	193.4470	0.0	1.223
4	1989-04-01	0.0	0.0	0.0	3.675	0.0	0.0	195.0895	0.0	1.243

5 rows × 201 columns

We have successfully Structured this dataframe . Now we process annual dataframe:

In [71]:
```
df_exchange_annual.head()
```

Out[71]:

	Period	Afghanistan	Angola	Albania	United Arab Emirates	Argenti0	Armenia	Developing Asia	Antigua and Barbuda	Au
0	1989	0.0	0.000	0.0000	3.675000	0.000000	0.0	213.1695	0.00000	1.
1	1990	0.0	0.000	0.0000	3.674964	0.000000	0.0	242.0275	0.00000	1.
2	1991	0.0	0.000	0.0000	3.675188	0.000000	0.0	282.4442	0.00000	1.
3	1992	0.0	0.000	0.0000	3.673874	0.991757	0.0	290.6883	0.00000	1.
4	1993	0.0	4903.561	110.0993	3.675296	0.998979	0.0	299.6200	2.69682	1.

5 rows × 201 columns

Here also we need to rename Period Column and also correct the misspelt country column names:

In [72]:
```
# Renaming Period Column

df_exchange_annual.rename(columns = {'Period':'Year'} , inplace = True)

# Now renaming Columns with misspelt name :

df_exchange_annual.columns = new_list
```

In [73]:
```
df_exchange_annual.head()
```

Out[73]:

	Year	Afghanistan	Angola	Albania	United Arab Emirates	Argentina	Armenia	Developing Asia	Antigua and Barbuda	Au
0	1989	0.0	0.000	0.0000	3.675000	0.000000	0.0	213.1695	0.00000	1.2
1	1990	0.0	0.000	0.0000	3.674964	0.000000	0.0	242.0275	0.00000	1.2
2	1991	0.0	0.000	0.0000	3.675188	0.000000	0.0	282.4442	0.00000	1.2
3	1992	0.0	0.000	0.0000	3.673874	0.991757	0.0	290.6883	0.00000	1.3
4	1993	0.0	4903.561	110.0993	3.675296	0.998979	0.0	299.6200	2.69682	1.4

5 rows × 201 columns

We haev successfully Structred df_exchange_annual dataframe.

In [74]:
```
# Basic Stats
```

In [75]:
```
# For Monthly Exchange Rates

df_exchange_monthly.describe()
```

Out[75]:

	Afghanistan	Angola	Albania	United Arab Emirates	Argentina	Armenia	Developing Asia	
count	360.000000	3.600000e+02	360.000000	360.000000	360.000000	360.000000	360.000000	36
mean	32.186290	4.751834e+04	96.485436	3.673313	4.221138	276.020862	800.864416	
std	26.305351	1.687534e+05	45.734716	0.001281	5.793594	228.384080	370.982459	
min	0.000000	0.000000e+00	0.000000	3.663136	0.000000	0.000000	190.359200	
25%	0.000000	4.039077e+01	95.418430	3.673000	0.999950	0.000000	324.826175	
50%	47.455885	8.946021e+01	105.073500	3.673000	2.926575	370.351300	913.325400	
75%	50.072937	1.682201e+02	124.450950	3.673045	4.085869	478.268150	1010.439000	
max	69.778190	2.179222e+06	176.522100	3.678714	38.647830	590.400000	1473.699000	

8 rows × 200 columns

In [76]:
```
# For Annual Exchange Rates:

df_exchange_annual.describe()
```

Out[76]:

	Year	Afghanistan	Angola	Albania	United Arab Emirates	Argentina	Armenia	De
count	30.000000	30.000000	30.000000	30.000000	30.000000	30.000000	30.000000	3
mean	2003.500000	33.727701	47587.127773	99.231948	3.673308	4.296596	285.391613	80
std	8.803408	26.802241	114758.886901	43.450594	0.000845	5.868982	230.015974	37
min	1989.000000	0.000000	0.000000	0.000000	3.672184	0.000000	0.000000	2
25%	1996.250000	0.000000	74.018648	98.537425	3.672992	0.999887	0.000000	33
50%	2003.500000	47.428790	92.808430	104.928000	3.673000	2.944350	373.080700	9
75%	2010.750000	50.306153	227.888350	123.447075	3.673203	4.074918	479.845725	99
max	2018.000000	69.363510	516812.700000	152.499600	3.675296	27.165270	578.763000	139

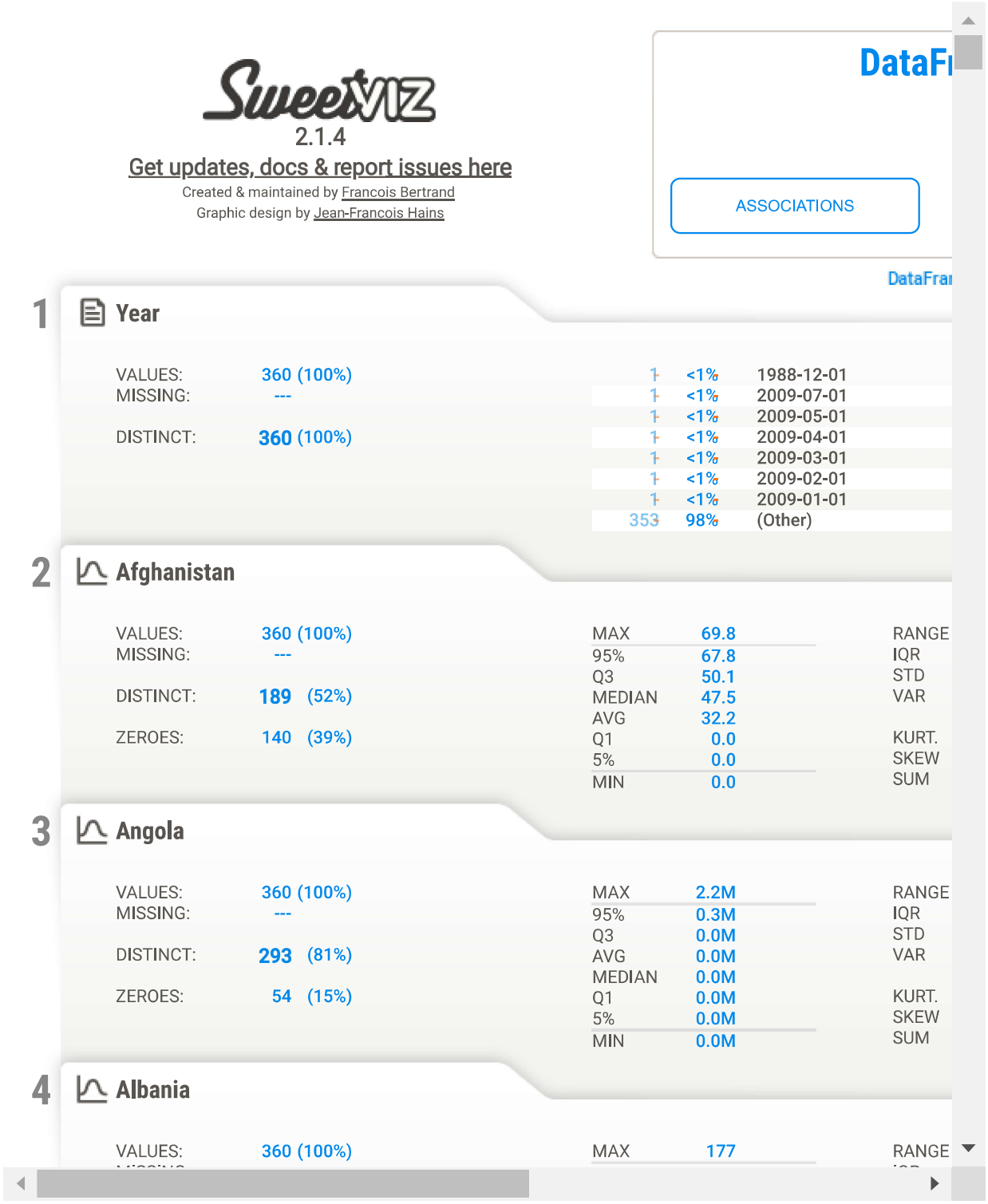
8 rows × 201 columns

In [77]:
```
# Creating Sweetviz report for automated analysis:

data3 = sv.analyze(df_exchange_monthly , pairwise_analysis= 'off')
```

Feature: Zimbabwe [100%] 00:55 -> (00:00 left)

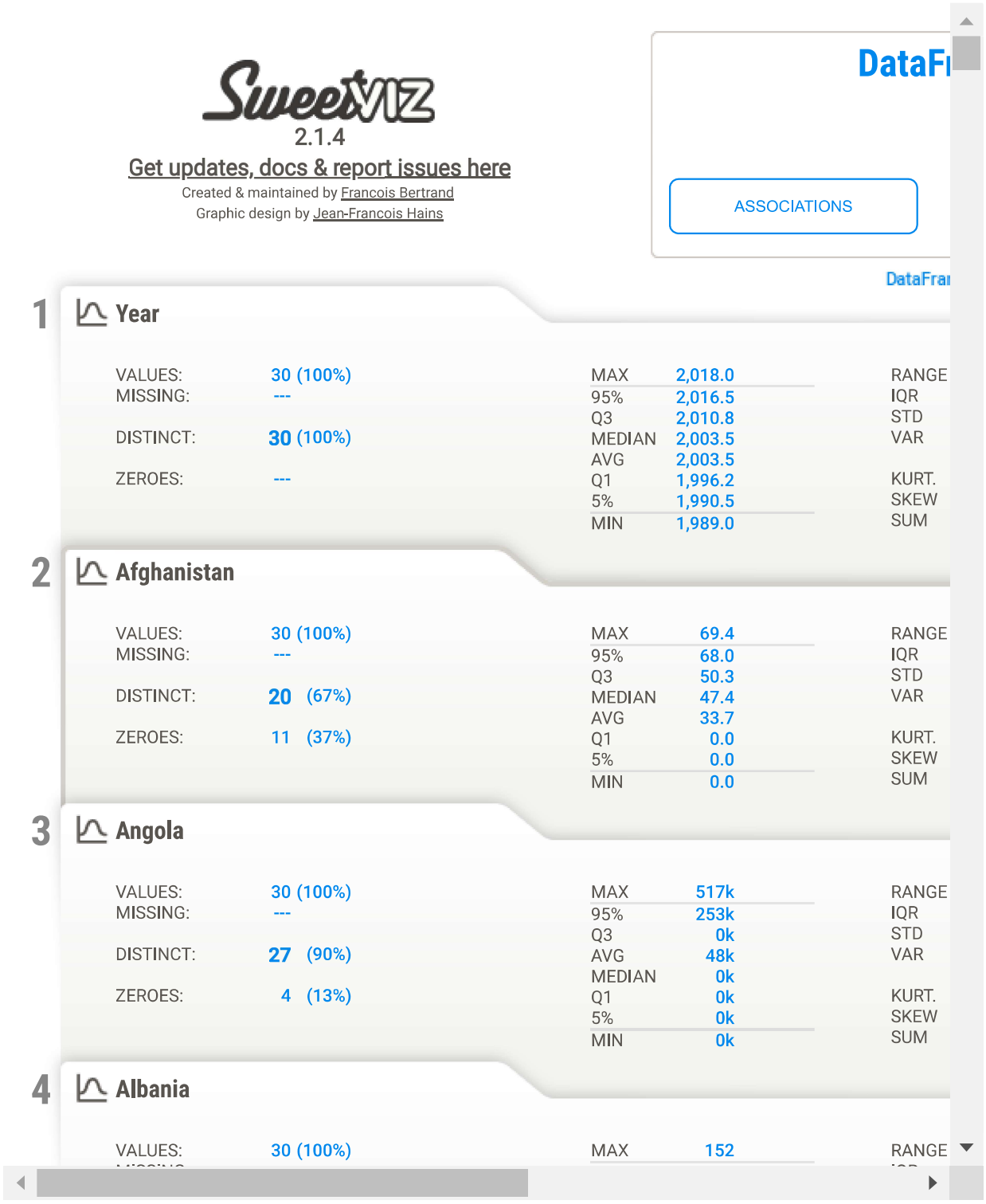
In [78]:
```
data3.show_notebook()
```



In [79]:
```
data4 = sv.analyze(df_exchange_annual , pairwise_analysis= 'off')
```

Feature: Zimbabwe [100%] 00:53 -> (00:00 left)

In [80]:
```
data4.show_notebook()
```



In [81]:
```
# saving structured data as excel file:      (Please change the location of file

with pd.ExcelWriter(r'C:\Users\bolt0\Data science\DeloitteCaseStudy\Structured Da
    df_exchange_monthly.to_excel(writer, sheet_name='monthly')
    df_exchange_annual.to_excel(writer, sheet_name='annual')
```

Exports Merchandise Data Analysis

In [82]:
```
df_exports_monthly.head()
```

Out[82]:

	Unnamed: 0	Argentina	Developing Asia	Australia	Belgium	Brazil	Canada	Chile	China	Czech Republi
0	1991M01	1265.303	3313.754	5502.346	0.0	4896.404	0.0	0.0	0.0	0.0
1	1991M02	1518.856	0.000	5708.852	0.0	3887.668	0.0	0.0	0.0	0.0
2	1991M03	1632.372	0.000	6890.402	0.0	4384.693	0.0	0.0	0.0	0.0
3	1991M04	1884.038	0.000	6524.655	0.0	5119.613	0.0	0.0	0.0	0.0
4	1991M05	2302.835	0.000	7145.694	0.0	4917.469	0.0	0.0	0.0	0.0

5 rows × 55 columns

In [83]:
```
df_exports_monthly.columns
```

Out[83]:
```
Index(['Unnamed: 0', 'Argentina', 'Developing Asia', 'Australia', 'Belgium',
       'Brazil', 'Canada', 'Chile', 'China', 'Czech Republic', 'Germany',
       'Developing Countries', 'Denmark', 'East Asia & Pacific developing',
       'Europe & Central Asia developing', 'Spain', 'Estonia', 'Finland',
       'France', 'United Kingdom', 'High Income Countries',
       'Hong Kong SAR, China', 'Hungary', 'High income: OECD', 'Indonesia',
       'Israel', 'Italy', 'Jordan', 'Japan', 'Korea, Rep.',
       'Latin America & Caribbean developing', 'Low Income', 'Mexico',
       'Middle Income Countries', 'Middle East & N. Africa developing',
       'Malaysia', 'Netherlands', 'Norway', 'New Zealand',
       'High Income: Non-OECD', 'Peru', 'Philippines', 'Poland',
       'Russian Federation', 'South Asia developing', 'Singapore',
       'Sub-Saharan Africa developing', 'Slovakia', 'Sweden', 'Thailand',
       'Turkey', 'Taiwan, China', 'Uganda', 'United States',
       'World (WBG members)'],
      dtype='object')
```

Here we have 2 problems in the monthly exports dataframe

1. renaming unnamed column as year
2. changing format of Year column as YYYY-MM-DD

We can repeat this process as earlier we have done:

In [84]:
```
# Renaming Period Column and Changing it's format

df_exports_monthly.rename(columns = {'Unnamed: 0':'Year'} , inplace = True)
```

In [85]:
```
# conversion into YYYY-MM-DD Format.

# For YYYY we have selected from 0th index to 4th index
l1 = [ ]
for i in range(len(df_exports_monthly)):
    l1.append(df_exports_monthly['Year'][i][0:4])

# For MM format we have selected 5th and 6th index as they represent the month nu
l2 = [ ]
for i in range(len(df_exports_monthly)):
    l2.append(df_exports_monthly['Year'][i][5:7])

# Now to add YYYY and MM together we create another list
l3 = [ ]
for i in range(len(l1)):
    l3.append(l1[i]+'-'+l2[i])

# Now adding DD , for this we consider 01 i.e. initial date and we also add in l3
l4 = [ ]
for i in range(len(l3)):
    l4.append(datetime.datetime.strptime(l3[i], '%Y-%m').date())

# Changing the Year column:
df_exports_monthly['Year'] = l4
```

In [86]:
```
df_exports_monthly.head()
```

Out[86]:

	Year	Argentina	Developing Asia	Australia	Belgium	Brazil	Canada	Chile	China	Czech Republic	...
0	1991-01-01	1265.303	3313.754	5502.346	0.0	4896.404	0.0	0.0	0.0	0.0	...
1	1991-02-01	1518.856	0.000	5708.852	0.0	3887.668	0.0	0.0	0.0	0.0	...
2	1991-03-01	1632.372	0.000	6890.402	0.0	4384.693	0.0	0.0	0.0	0.0	...
3	1991-04-01	1884.038	0.000	6524.655	0.0	5119.613	0.0	0.0	0.0	0.0	...
4	1991-05-01	2302.835	0.000	7145.694	0.0	4917.469	0.0	0.0	0.0	0.0	...

5 rows × 55 columns

In [87]:
```
# Now for annual Column:

df_exports_annual.head()
```

Out[87]:

	Unnamed: 0	Argentina	Developing Asia	Australia	Belgium	Brazil	Canada	Chile	China	Czec Republi
0	1991	22191.58	0.0	80560.51	0.0	53953.57	0.0	0.0	0.0	0.
1	1992	22254.20	0.0	85549.90	0.0	63269.99	0.0	0.0	0.0	0.
2	1993	24153.33	0.0	90912.32	0.0	73589.75	0.0	0.0	0.0	0.
3	1994	28583.86	0.0	96492.75	0.0	75080.52	0.0	0.0	0.0	0.
4	1995	35332.53	0.0	99382.24	296008.8	70497.63	0.0	0.0	0.0	0.

5 rows × 55 columns

In [88]:
```
df_exports_annual.columns
```

Out[88]:
```
Index(['Unnamed: 0', 'Argentina', 'Developing Asia', 'Australia', 'Belgium',
       'Brazil', 'Canada', 'Chile', 'China', 'Czech Republic', 'Germany',
       'Developing Countries', 'Denmark', 'East Asia & Pacific developing',
       'Europe & Central Asia developing', 'Spain', 'Estonia', 'Finland',
       'France', 'United Kingdom', 'High Income Countries',
       'Hong Kong SAR, China', 'Hungary', 'High income: OECD', 'Indonesia',
       'Israel', 'Italy', 'Jordan', 'Japan', 'Korea, Rep.',
       'Latin America & Caribbean developing', 'Low Income', 'Mexico',
       'Middle Income Countries', 'Middle East & N. Africa developing',
       'Malaysia', 'Netherlands', 'Norway', 'New Zealand',
       'High Income: Non-OECD', 'Peru', 'Philippines', 'Poland',
       'Russian Federation', 'South Asia developing', 'Singapore',
       'Sub-Saharan Africa developing', 'Slovakia', 'Sweden', 'Thailand',
       'Turkey', 'Taiwan, China', 'Uganda', 'United States',
       'World (WBG members)'],
      dtype='object')
```

In [89]:
```
# Renaming unnamed Column and Changing it's format

df_exports_annual.rename(columns = {'Unnamed: 0':'Year'} , inplace = True)
```

In [90]:
```
df_exports_annual.head()
```

Out[90]:

	Year	Argentina	Developing Asia	Australia	Belgium	Brazil	Canada	Chile	China	Czech Republic	...
0	1991	22191.58	0.0	80560.51	0.0	53953.57	0.0	0.0	0.0	0.0	...
1	1992	22254.20	0.0	85549.90	0.0	63269.99	0.0	0.0	0.0	0.0	...
2	1993	24153.33	0.0	90912.32	0.0	73589.75	0.0	0.0	0.0	0.0	...
3	1994	28583.86	0.0	96492.75	0.0	75080.52	0.0	0.0	0.0	0.0	...
4	1995	35332.53	0.0	99382.24	296008.8	70497.63	0.0	0.0	0.0	0.0	...

5 rows × 55 columns

In [91]:
```
# Now at last for Quarterly Column:
df_exports_quarterly
```

Out[91]:

	Unnamed: 0	Argentina	Developing Asia	Australia	Belgium	Brazil	Canada	Chile	China
0	1991Q1	4416.531	0.0	18101.60	0.00	13168.76	0.0000	0.00	0.0
1	1991Q2	6318.661	0.0	20387.41	0.00	14826.59	0.0000	0.00	0.0
2	1991Q3	6371.975	0.0	20408.96	0.00	12690.13	0.0000	0.00	0.0
3	1991Q4	5084.412	0.0	21662.54	0.00	13268.09	0.0000	0.00	0.0
4	1992Q1	4724.392	0.0	19736.18	0.00	13636.11	0.0000	0.00	0.0
...	...	...	...	...	...	...	...	...	...
106	2017Q3	16770.970	720194.3	59437.88	90699.26	61736.42	115.8469	17700.18	531477.9
107	2017Q4	14745.500	735916.6	58454.59	88748.78	55125.52	121.1300	18825.41	552142.0
108	2018Q1	14262.010	0.0	56989.81	85198.43	54994.27	116.7996	19143.87	0.0
109	2018Q2	14454.230	0.0	60143.25	84912.30	59141.24	126.2617	18995.88	0.0
110	2018Q3	0.000	0.0	0.00	0.00	0.00	0.0000	0.00	0.0

111 rows × 55 columns

In [92]:
```
# Renaming unnamed Column and Changing it's format

df_exports_quarterly.rename(columns = {'Unnamed: 0':'Year'} , inplace = True)
```

In [95]:
```
# conversion into YYYY-QQ Format.

# For YYYY we have selected from 0th index to 4th index
l1 = [ ]
for i in range(len(df_exports_quarterly)):
    l1.append(df_exports_quarterly['Year'][i][0:4])

# For QQ format we have selected 4th and 5th index as they represent the month nu
l2 = [ ]
for i in range(len(df_exports_quarterly)):
    l2.append(df_exports_quarterly['Year'][i][4:6])

# Now to add YYYY and QQ together we create another list
l3 = [ ]
for i in range(len(l1)):
    l3.append(l1[i]+'-'+l2[i])


# Changing the Year column:
df_exports_quarterly['Year'] = l3
```

In [96]:
```
df_exports_quarterly.head()
```

Out[96]:

	Year	Argentina	Developing Asia	Australia	Belgium	Brazil	Canada	Chile	China	Czech Republic	...
0	1991-Q1	4416.531	0.0	18101.60	0.0	13168.76	0.0	0.0	0.0	0.0	...
1	1991-Q2	6318.661	0.0	20387.41	0.0	14826.59	0.0	0.0	0.0	0.0	...
2	1991-Q3	6371.975	0.0	20408.96	0.0	12690.13	0.0	0.0	0.0	0.0	...
3	1991-Q4	5084.412	0.0	21662.54	0.0	13268.09	0.0	0.0	0.0	0.0	...
4	1992-Q1	4724.392	0.0	19736.18	0.0	13636.11	0.0	0.0	0.0	0.0	...

5 rows × 55 columns

We have formed the structured data now we need basic stats and sweetviz analysis

In [97]:
```
df_exports_monthly.describe()
```

Out[97]:

	Argentina	Developing Asia	Australia	Belgium	Brazil	Canada	C
count	333.000000	333.000000	333.000000	333.000000	333.000000	333.000000	333.000
mean	4293.504315	86533.475057	12484.827240	24607.161411	11784.298838	28.027296	3148.047
std	1363.578370	95023.155705	3845.368714	10697.248864	5013.465744	15.382348	2839.813
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000
25%	3543.171000	0.000000	9862.041000	25208.060000	6792.103000	29.326400	0.000
50%	4492.461000	0.000000	12202.870000	29387.070000	12399.750000	35.992380	4810.443
75%	5270.065000	184575.800000	15131.970000	30584.350000	16157.960000	37.773010	5760.006
max	6928.239000	255932.400000	22022.220000	33498.300000	22464.320000	43.833780	7188.550

8 rows × 54 columns

In [98]:
```
df_exports_annual.describe()
```

Out[98]:

	Year	Argentina	Developing Asia	Australia	Belgium	Brazil	
count	28.000000	28.000000	2.800000e+01	28.000000	28.000000	28.000000	28
mean	2004.500000	50036.452500	9.646903e+05	144296.943571	286574.071429	135270.417143	324
std	8.225975	17419.283383	1.154135e+06	50412.992359	138410.534305	61664.935681	193
min	1991.000000	0.000000	0.000000e+00	0.000000	0.000000	0.000000	0
25%	1997.750000	42324.590000	0.000000e+00	118291.975000	299929.000000	78591.720000	258
50%	2004.500000	54931.890000	0.000000e+00	143463.750000	352351.550000	148706.300000	429
75%	2011.250000	61788.535000	2.172737e+06	178319.325000	367119.825000	188897.125000	450
max	2018.000000	69768.490000	2.738630e+06	228618.900000	385915.200000	228728.200000	479

8 rows × 55 columns

In [99]:
```
df_exports_quarterly.describe()
```

Out[99]:

	Argentina	Developing Asia	Australia	Belgium	Brazil	Canada	
count	111.000000	111.000000	111.000000	111.000000	111.000000	111.000000	111.0
mean	12880.512937	253912.501802	37454.481532	73821.483604	35150.514955	84.081887	9444.1
std	4027.077206	284850.980625	11447.790935	32173.940626	14871.974009	46.275518	8510.7
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
25%	10791.245000	0.000000	29830.860000	75477.225000	20170.815000	88.614225	0.0
50%	13738.760000	0.000000	36487.320000	88300.780000	38919.610000	107.980400	15192.9
75%	16095.565000	549909.900000	45680.885000	92029.185000	48712.660000	113.214100	17292.9
max	20018.850000	735916.600000	60303.700000	98223.310000	61736.420000	126.261700	19218.7

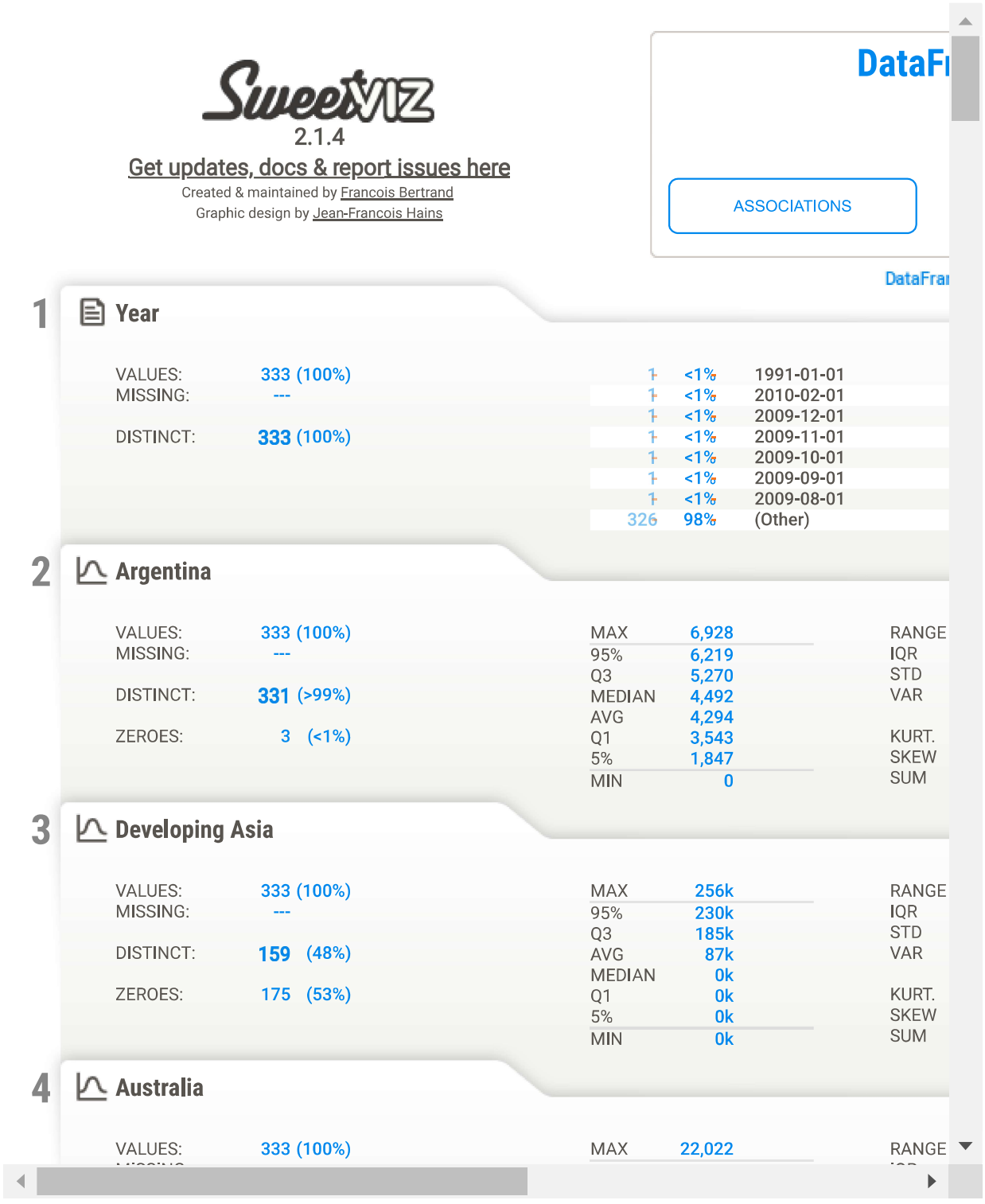
8 rows × 54 columns

In [100]:
```
# creating sweetviz report:

data5 = sv.analyze(df_exports_monthly , pairwise_analysis= 'off')
```

Feature: World (WBG members) [100%] 00:13 -> (00:00 left)

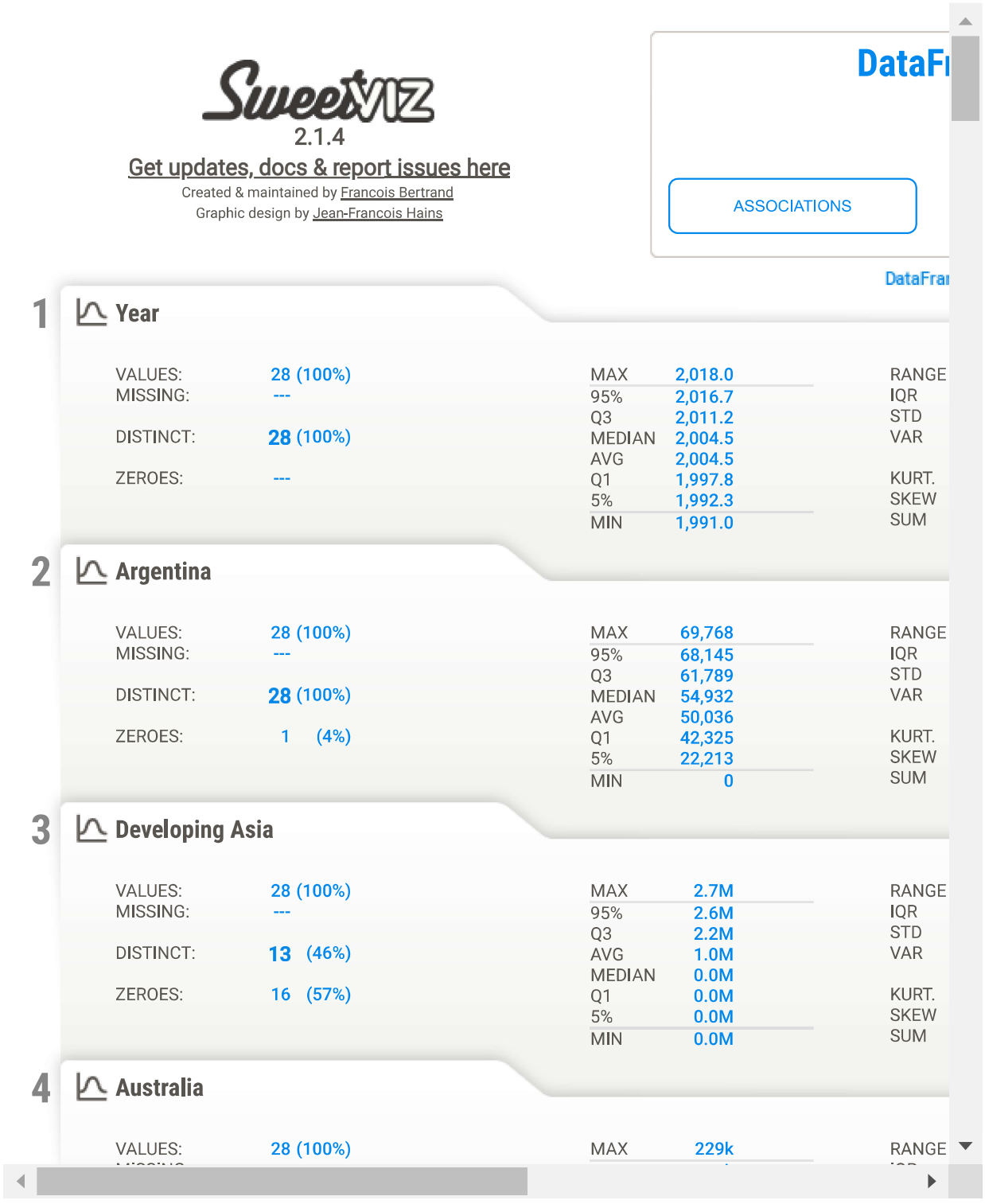
In [101]:
```
data5.show_notebook()
```



In [102]:
```
data6 = sv.analyze(df_exports_annual , pairwise_analysis= 'off')
```

Feature: World (WBG members) [100%] 00:20 -> (00:00 left)

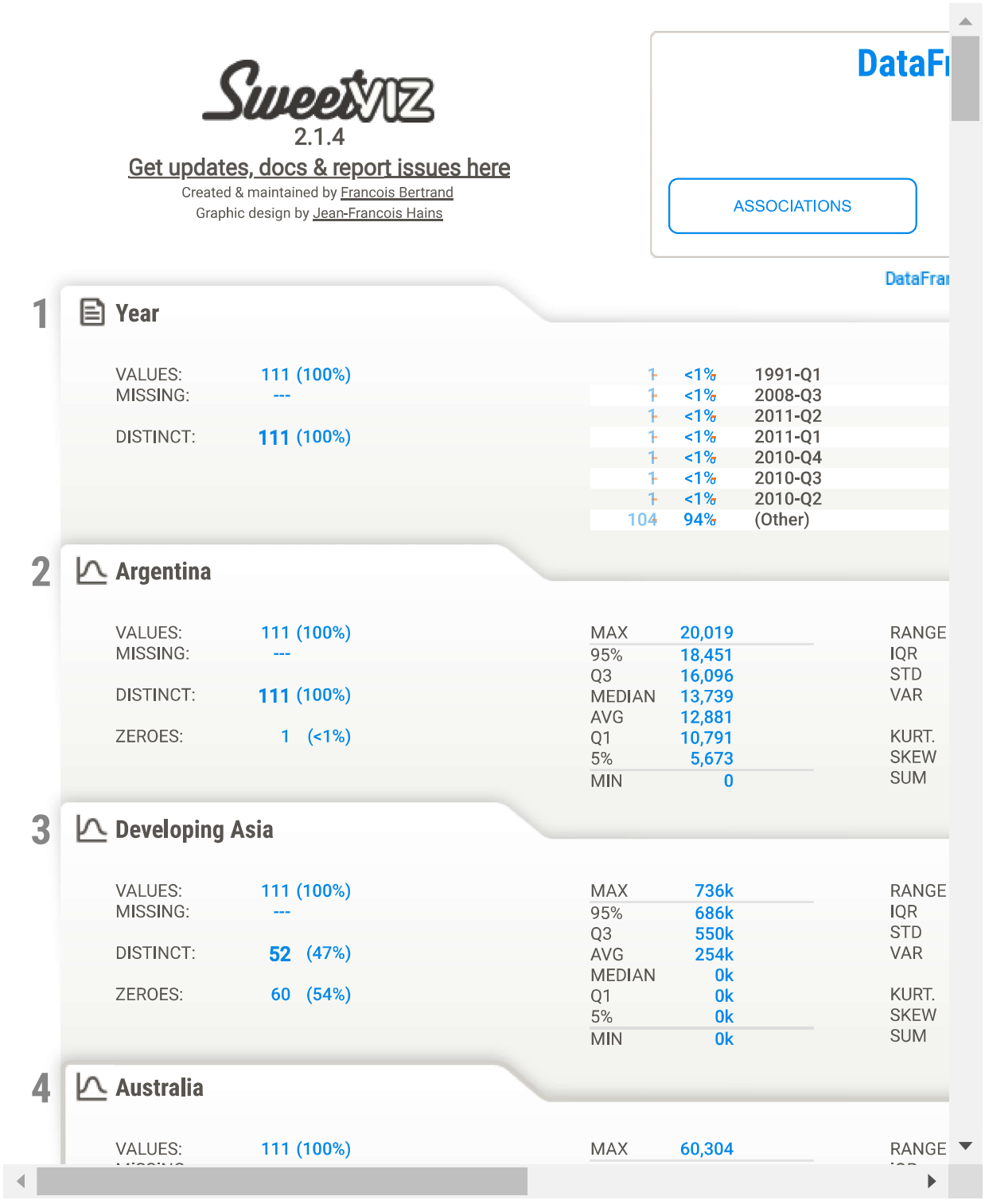
In [103]:
```
data6.show_notebook()
```



In [104]:
```
data7 = sv.analyze(df_exports_quarterly , pairwise_analysis= 'off')
```

Feature: World (WBG members) [100%] 00:14 -> (00:00 left)

In [105]:
```
data7.show_notebook()
```



In [106]:
```
# saving structured data as excel file:          (Please enter location according

with pd.ExcelWriter(r'C:\Users\bolt0\Data science\DeloitteCaseStudy\Structured Da
    df_exports_monthly.to_excel(writer, sheet_name='monthly')
    df_exports_annual.to_excel(writer, sheet_name='annual')
    df_exports_quarterly.to_excel(writer, sheet_name='quarterly')
```

So we had an Unprocessed and unstructured data in the beginning which we have processed and made it Structured .

THANK YOU

--Performed By: Ankit Dubey