

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

data=pd.read_csv('/content/World Marriage Dataset.csv')
```


data



	Sr.No.	Country	AgeGroup	Sex	MaritalStatus	DataProcess	Data Collection (Start Year)
0	1	Afghanistan	[15-19]	Man	Divorced	Survey	1972
1	2	Afghanistan	[20-24]	Man	Divorced	Survey	1972
2	3	Afghanistan	[25-29]	Man	Divorced	Survey	1972
3	4	Afghanistan	[30-34]	Man	Divorced	Survey	1972
4	5	Afghanistan	[35-39]	Man	Divorced	Survey	1972
...
271599	271600	Zimbabwe	[55-59]	Woman	Widowed	Survey	2017



data.shape



```
(271604, 9)
```

data.describe()



	Sr.No.	Data Collection (Start Year)	Data Collection (End Year)
count	271604.000000	271604.000000	271604.000000
mean	135802.500000	1996.961488	1997.059097
std	78405.465594	14.244590	14.284136
min	1.000000	1954.000000	1955.000000
25%	67901.750000	1986.000000	1986.000000
50%	135802.500000	2000.000000	2000.000000
75%	203703.250000	2010.000000	2010.000000
max	271604.000000	2019.000000	2019.000000

data.info()



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 271604 entries, 0 to 271603
Data columns (total 9 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Sr.No.                                271604 non-null int64
1   Country                               271604 non-null object
2   AgeGroup                              271604 non-null object
3   Sex                                    271604 non-null object
4   MaritalStatus                         271604 non-null object
5   DataProcess                           271604 non-null object
6   Data Collection (Start Year)           271604 non-null int64
7   Data Collection (End Year)             271604 non-null int64
8   Data Source                           271604 non-null object
dtypes: int64(3), object(6)
memory usage: 18.6+ MB
```

data.isnull().sum()



	0
Sr.No.	0
Country	0
AgeGroup	0
Sex	0
MaritalStatus	0
DataProcess	0
Data Collection (Start Year)	0
Data Collection (End Year)	0
Data Source	0

dtype: int64

df=data

df.drop('Sr.No.',axis=1,inplace=True)

df



	Country	AgeGroup	Sex	MaritalStatus	DataProcess	Data Collection (Start Year)	Data Collection (End Year)
0	Afghanistan	[15-19]	Man	Divorced	Survey	1972	1
1	Afghanistan	[20-24]	Man	Divorced	Survey	1972	1
2	Afghanistan	[25-29]	Man	Divorced	Survey	1972	1
3	Afghanistan	[30-34]	Man	Divorced	Survey	1972	1
4	Afghanistan	[35-39]	Man	Divorced	Survey	1972	1
...
271599	Zimbabwe	[55-59]	Woman	Widowed	Survey	2017	2

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

```
→ array(['Afghanistan', 'Albania', 'Algeria', 'American Samoa', 'Angola',  
'Anguilla', 'Antigua and Barbuda', 'Argentina', 'Armenia', 'Aruba',  
'Australia', 'Austria', 'Azerbaijan', 'Bahamas', 'Bahrain',  
'Bangladesh', 'Barbados', 'Belarus', 'Belgium', 'Belize', 'Benin',  
'Bermuda', 'Bhutan', 'Bolivia (Plurinational State of)',  
'Bosnia and Herzegovina', 'Botswana', 'Brazil',  
'British Virgin Islands', 'Brunei Darussalam', 'Bulgaria',  
'Burkina Faso', 'Burundi', 'Cabo Verde', 'Cambodia', 'Cameroon',  
'Canada', 'Cayman Islands', 'Central African Republic', 'Chad',  
'Channel Islands', 'Chile', 'China', 'China, Hong Kong SAR',  
'China, Macao SAR', 'China, Taiwan Province of China', 'Colombia',  
'Comoros', 'Congo', 'Cook Islands', 'Costa Rica', 'Côte d'Ivoire',  
'Croatia', 'Cuba', 'Curaçao', 'Cyprus', 'Czechia',  
'Dem. People's Rep. of Korea', 'Democratic Republic of the Congo',  
'Denmark', 'Djibouti', 'Dominica', 'Dominican Republic', 'Ecuador',  
'Egypt', 'El Salvador', 'Equatorial Guinea', 'Eritrea', 'Estonia',  
'Ethiopia', 'Faeroe Islands', 'Falkland Islands (Malvinas)',  
'Fiji', 'Finland', 'France', 'French Guiana', 'French Polynesia',  
'Gabon', 'Gambia', 'Georgia', 'Germany', 'Ghana', 'Gibraltar',  
'Greece', 'Greenland', 'Grenada', 'Guadeloupe', 'Guam',  
'Guatemala', 'Guinea', 'Guinea-Bissau', 'Guyana', 'Haiti',  
'Honduras', 'Hungary', 'Iceland', 'India', 'Indonesia',  
'Iran (Islamic Republic of)', 'Iraq', 'Ireland', 'Isle of Man',  
'Israel', 'Italy', 'Jamaica', 'Japan', 'Jordan', 'Kazakhstan',  
'Kenya', 'Kiribati', 'Kosovo (autonomous province)', 'Kuwait',  
'Kyrgyzstan', 'Lao People's Democratic Republic',  
'Lao People's Dem. Republic', 'Lao People's Democratic Republic',  
'Latvia', 'Lebanon', 'Lesotho', 'Liberia', 'Libya',  
'Liechtenstein', 'Lithuania', 'Luxembourg', 'Madagascar', 'Malawi',  
'Malaysia', 'Maldives', 'Mali', 'Malta', 'Marshall Islands',  
'Martinique', 'Mauritania', 'Mauritius', 'Mayotte', 'Mexico',  
'Micronesia (Fed. States of)', 'Monaco', 'Mongolia', 'Montenegro',  
'Montserrat', 'Morocco', 'Mozambique', 'Myanmar', 'Namibia',  
'Nauru', 'Nepal', 'Netherlands', 'Netherlands Antilles',  
'New Caledonia', 'New Zealand', 'Nicaragua', 'Niger', 'Nigeria',  
'Niue', 'Northern Mariana Islands', 'Norway', 'Oman', 'Pakistan',  
'Palau', 'Panama', 'Papua New Guinea', 'Paraguay', 'Peru',  
'Philippines', 'Poland', 'Portugal', 'Puerto Rico', 'Qatar',  
'Republic of Korea', 'Republic of Moldova', 'Réunion', 'Romania',  
'Russian Federation', 'Rwanda', 'Saint Helena',  
'Saint Kitts and Nevis', 'Saint Lucia',  
'Saint Pierre and Miquelon', 'Saint Vincent and the Grenadines',  
'Samoa', 'San Marino', 'Sao Tome and Principe', 'Saudi Arabia',  
'Senegal', 'Serbia', 'Seychelles', 'Sierra Leone', 'Singapore',  
'Sint Maarten (Dutch part)', 'Slovakia', 'Slovenia',  
'Solomon Islands', 'Somalia', 'South Africa', 'South Sudan',  
'Spain', 'Sri Lanka', 'State of Palestine', 'Sudan', 'Suriname',  
'Eswatini', 'Sweden', 'Switzerland', 'Syrian Arab Republic',
```

```
'Tajikistan', 'TFYR Macedonia', 'Thailand',
'The former Yugoslav Republic of Macedonia', 'Timor-Leste', 'Togo',
'Tokelau', 'Tonga', 'Trinidad and Tobago', 'Tunisia', 'Turkey',
'Turkmenistan', 'Turks and Caicos Islands', 'Tuvalu', 'Uganda',
'Ukraine', 'United Arab Emirates', 'United Kingdom',
'United Republic of Tanzania', 'United States of America',
'United States Virgin Islands', 'Uruguay', 'Uzbekistan', 'Vanuatu',
'Venezuela (Bolivarian Republic of)', 'Viet Nam',
'Wallis and Futuna Islands', 'Western Sahara', 'Yemen', 'Zambia',
... ..
```

```
df['Country'].nunique()
```

```
↔ 235
```

```
df.AgeGroup.unique()
```

```
↔ array(['[15-19]', '[20-24]', '[25-29]', '[30-34]', '[35-39]', '[40-44]',
        '[45-49]', '[50-54]', '[55-59]', '[60-64]', '[65+]', '[10-14]',
        '[15-24]', '[25-39]', '[40-64]', '[0-14]', '[65-69]', '[70-74]',
        '[75+]', '[70+]', '[40-49]', '[50-59]', '[25-34]', '[35-44]',
        '[45+]', '[45-54]', '[55-64]', '[60+]', '[60-74]', '[14-19]',
        '[18-19]', '[60-69]', '[16-19]', '[50+]', '[20-29]', '[30-39]',
        '[20-39]', '[40-59]', '[35-64]', '[15-29]', '[30-44]', '[45-59]',
        '[16-29]', '[55+]', '[10-19]', '[0-19]', '[0-15]', '[12-14]',
        '[65-74]', '[15-17]', '[40-54]', '[65-79]', '[50-64]', '[70-79]',
        '[60-66]', '[67-69]', '[20-34]', '[24-29]', '[20-23]', '[12-19]',
        '[35-49]', '[25-49]', '[16-24]'], dtype=object)
```

```
df.AgeGroup.nunique()
```

```
↔ 63
```

```
for i in df.columns:
    print(i)
    print(df[i].unique())
    print()
    print(df[i].nunique())
    print()
```

```
↔ Country
['Afghanistan' 'Albania' 'Algeria' 'American Samoa' 'Angola' 'Anguilla'
'Antigua and Barbuda' 'Argentina' 'Armenia' 'Aruba' 'Australia' 'Austria'
'Azerbaijan' 'Bahamas' 'Bahrain' 'Bangladesh' 'Barbados' 'Belarus'
'Belgium' 'Belize' 'Benin' 'Bermuda' 'Bhutan'
'Bolivia (Plurinational State of)' 'Bosnia and Herzegovina' 'Botswana'
'Brazil' 'British Virgin Islands' 'Brunei Darussalam' 'Bulgaria'
'Burkina Faso' 'Burundi' 'Cabo Verde' 'Cambodia' 'Cameroon' 'Canada'
'Cayman Islands' 'Central African Republic' 'Chad' 'Channel Islands'
'Chile' 'China' 'China, Hong Kong SAR' 'China, Macao SAR'
'China, Taiwan Province of China' 'Colombia' 'Comoros' 'Congo'
'Cook Islands' 'Costa Rica' 'Côte d'Ivoire' 'Croatia' 'Cuba' 'Curaçao'
'Cyprus' 'Czechia' 'Dem. People's Rep. of Korea'
'Democratic Republic of the Congo' 'Denmark' 'Djibouti' 'Dominica'
'Dominican Republic' 'Ecuador' 'Egypt' 'El Salvador' 'Equatorial Guinea'
'Eritrea' 'Estonia' 'Ethiopia' 'Faeroe Islands'
'Falkland Islands (Malvinas)' 'Fiji' 'Finland' 'France' 'French Guiana'
```

'French Polynesia' 'Gabon' 'Gambia' 'Georgia' 'Germany' 'Ghana'
 'Gibraltar' 'Greece' 'Greenland' 'Grenada' 'Guadeloupe' 'Guam'
 'Guatemala' 'Guinea' 'Guinea-Bissau' 'Guyana' 'Haiti' 'Honduras'
 'Hungary' 'Iceland' 'India' 'Indonesia' 'Iran (Islamic Republic of)'
 'Iraq' 'Ireland' 'Isle of Man' 'Israel' 'Italy' 'Jamaica' 'Japan'
 'Jordan' 'Kazakhstan' 'Kenya' 'Kiribati' 'Kosovo (autonomous province)'
 'Kuwait' 'Kyrgyzstan' 'Lao People's Democratic Republic'
 'Lao People's Dem. Republic' 'Lao People's Democratic Republic' 'Latvia'
 'Lebanon' 'Lesotho' 'Liberia' 'Libya' 'Liechtenstein' 'Lithuania'
 'Luxembourg' 'Madagascar' 'Malawi' 'Malaysia' 'Maldives' 'Mali' 'Malta'
 'Marshall Islands' 'Martinique' 'Mauritania' 'Mauritius' 'Mayotte'
 'Mexico' 'Micronesia (Fed. States of)' 'Monaco' 'Mongolia' 'Montenegro'
 'Montserrat' 'Morocco' 'Mozambique' 'Myanmar' 'Namibia' 'Nauru' 'Nepal'
 'Netherlands' 'Netherlands Antilles' 'New Caledonia' 'New Zealand'
 'Nicaragua' 'Niger' 'Nigeria' 'Niue' 'Northern Mariana Islands' 'Norway'
 'Oman' 'Pakistan' 'Palau' 'Panama' 'Papua New Guinea' 'Paraguay' 'Peru'
 'Philippines' 'Poland' 'Portugal' 'Puerto Rico' 'Qatar'
 'Republic of Korea' 'Republic of Moldova' 'Réunion' 'Romania'
 'Russian Federation' 'Rwanda' 'Saint Helena' 'Saint Kitts and Nevis'
 'Saint Lucia' 'Saint Pierre and Miquelon'
 'Saint Vincent and the Grenadines' 'Samoa' 'San Marino'
 'Sao Tome and Principe' 'Saudi Arabia' 'Senegal' 'Serbia' 'Seychelles'
 'Sierra Leone' 'Singapore' 'Sint Maarten (Dutch part)' 'Slovakia'
 'Slovenia' 'Solomon Islands' 'Somalia' 'South Africa' 'South Sudan'
 'Spain' 'Sri Lanka' 'State of Palestine' 'Sudan' 'Suriname' 'Eswatini'
 'Sweden' 'Switzerland' 'Syrian Arab Republic' 'Tajikistan'
 'TFYR Macedonia' 'Thailand' 'The former Yugoslav Republic of Macedonia'
 'Timor-Leste' 'Togo' 'Tokelau' 'Tonga' 'Trinidad and Tobago' 'Tunisia'
 'Turkey' 'Turkmenistan' 'Turks and Caicos Islands' 'Tuvalu' 'Uganda'
 'Ukraine' 'United Arab Emirates' 'United Kingdom'
 'United Republic of Tanzania' 'United States of America'
 'United States Virgin Islands' 'Uruguay' 'Uzbekistan' 'Vanuatu'
 'Venezuela (Bolivarian Republic of)' 'Viet Nam'
 'Wallis and Futuna Islands' 'Western Sahara' 'Yemen' 'Zambia' 'Zimbabwe']

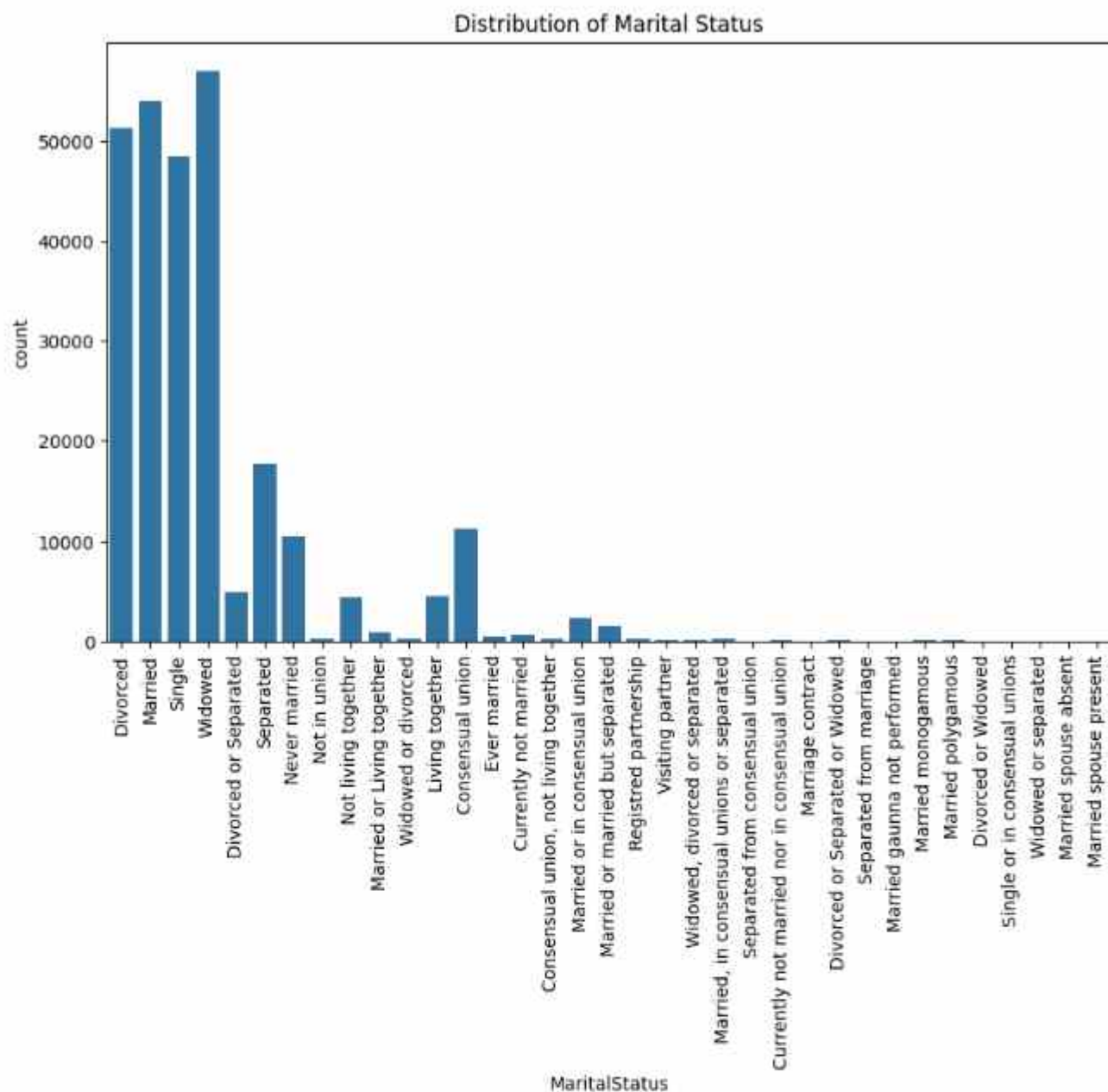
235

AgeGroup

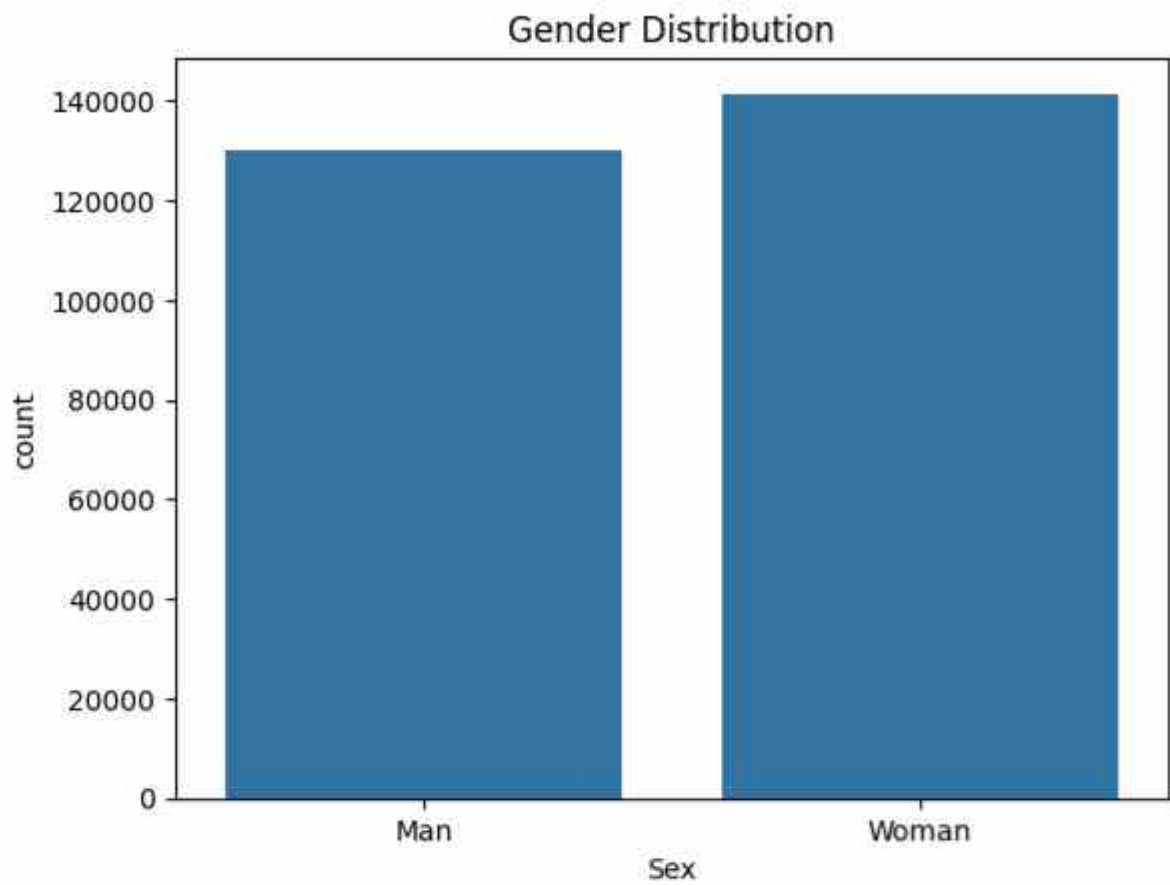
['[15-19]' '[20-24]' '[25-29]' '[30-34]' '[35-39]' '[40-44]' '[45-49]'
 '[50-54]' '[55-59]' '[60-64]' '[65+]' '[10-14]' '[15-24]' '[25-39]'
 '[40-64]' '[0-14]' '[65-69]' '[70-74]' '[75+]' '[70+]' '[40-49]']

✓ uni varient analysis

```
# Marital Status distribution
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='MaritalStatus') # Changed 'Marital Status' to 'Marital_Status'
plt.title("Distribution of Marital Status")
plt.xticks(rotation=90)
plt.show()
```



```
# Gender distribution
sns.countplot(data=df, x='Sex')
plt.title("Gender Distribution")
plt.show()
```

Bivariate & Multivariate Analysis

df



	Country	AgeGroup	Sex	MaritalStatus	DataProcess	Data Collection (Start Year)	Data Collect (End Year)
0	Afghanistan	[15-19]	Man	Divorced	Survey	1972	1
1	Afghanistan	[20-24]	Man	Divorced	Survey	1972	1
2	Afghanistan	[25-29]	Man	Divorced	Survey	1972	1
3	Afghanistan	[30-34]	Man	Divorced	Survey	1972	1
4	Afghanistan	[35-39]	Man	Divorced	Survey	1972	1
...
271599	Zimbabwe	[55-59]	Woman	Widowed	Survey	2017	2


```
# prompt: groupby country and each unique value in maritalstatus
```

```
# Group by 'Country' and 'MaritalStatus' and count occurrences
```

```
grouped_data = df.groupby(['Country', 'MaritalStatus']).size().unstack(fill_value=0)
grouped_data
```



MaritalStatus	Consensual union	Consensual union, not living together	Currently not married	Currently not married nor in consensual union	Divorced	Divorced or Separated	D Se
Country							
Afghanistan	0	0	0	0	100	14	
Albania	0	0	0	0	141	22	
Algeria	0	0	0	0	129	31	
American Samoa	18	0	0	0	84	0	
Angola	28	0	0	0	101	20	
...	
Wallis and Futuna Islands	0	0	0	0	77	67	
Western Sahara	0	0	0	0	28	0	
Yemen	0	0	0	0	204	0	
Zambia	20	0	0	0	252	20	
Zimbabwe	0	0	0	0	219	136	

235 rows × 35 columns



```
df.MaritalStatus.value_counts()
```



	count
MaritalStatus	
Widowed	57002
Married	53955
Divorced	51269
Single	48472
Separated	17766
Consensual union	11279
Never married	10429
Divorced or Separated	4886
Living together	4536
Not living together	4335
Married or in consensual union	2314
Married or married but separated	1520
Married or Living together	892
Currently not married	607
Ever married	509
Widowed or divorced	290
Consensual union, not living together	265
Not in union	249
Married, in consensual unions or separated	196
Registered partnership	175
Divorced or Separated or Widowed	133
Visiting partner	104
Widowed, divorced or separated	59
Married monogamous	50
Married polygamous	49
Currently not married nor in consensual union	46
Separated from consensual union	37
Divorced or Widowed	31
Single or in consensual unions	28
Separated from marriage	27
Married spouse present	24

Married spouse absent	24
Marriage contract	20
Married gaunna not performed	14
Widowed or separated	12

dtype: int64

Based on analysis of 271,605 records, the most reported marital statuses are "Widowed", "Married", and "Divorced", with a significant presence of non-traditional statuses such as "Consensual union", "Living together", and "Not in union".

```
df.groupby('Country')['MaritalStatus'].value_counts()
```



		count
Country	MaritalStatus	
Afghanistan	Married	116
	Widowed	116
	Divorced	100
	Single	71
	Never married	47
...
Zimbabwe	Living together	91
	Not living together	77
	Separated	66
	Married or in consensual union	28
	Not in union	7

1914 rows × 1 columns

dtype: int64

```
df.groupby('Country')['Sex'].value_counts()
```



		count
Country		Sex
Afghanistan	Woman	289
	Man	210
Albania	Woman	503
	Man	443
Algeria	Woman	383

Yemen	Man	367

Zambia	Woman	757
	Man	728
Zimbabwe	Woman	941
	Man	795

468 rows × 1 columns

dtype: int64

```
df.AgeGroup.value_counts().head()
```



	count
AgeGroup	
[25-29]	22666
[20-24]	22580
[30-34]	22560
[35-39]	22466
[40-44]	22444

dtype: int64

```
df.AgeGroup.value_counts().tail()
```



	count
AgeGroup	
[40-54]	8
[20-23]	5
[16-24]	5
[35-49]	4
[12-19]	2

dtype: int64

```
df.DataProcess.value_counts()
```



	count
DataProcess	
Census	106919
Estimate	93942
Survey	70215
Multiround surv	224
Dual record	192
GGS	112

dtype: int64

✓ Insights

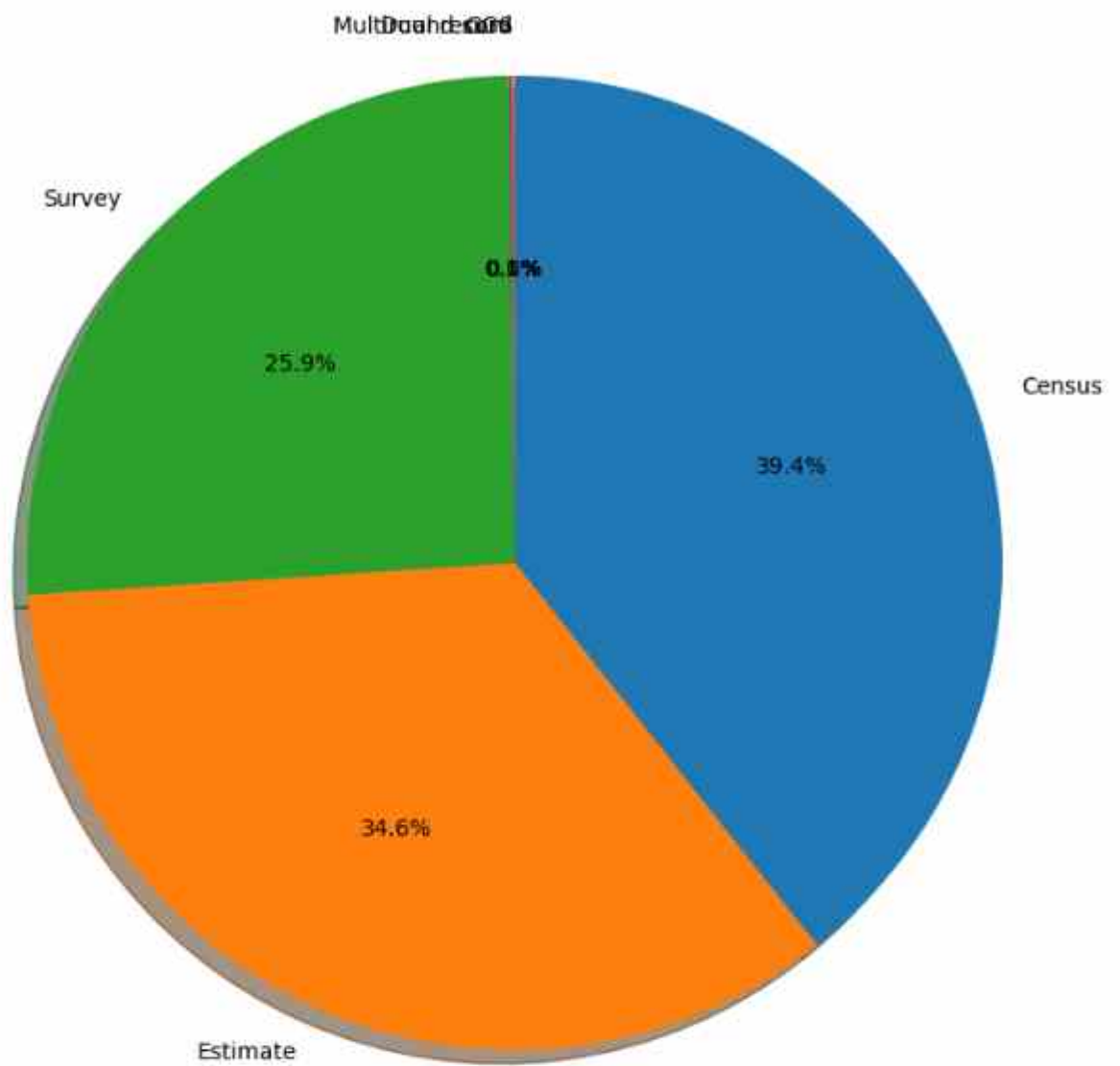
The dataset primarily relies on Census, Estimates, and Survey data, ensuring strong reliability for global-level trend analysis. However, attention should be paid to small portions of data from specialized sources like “Multiround surveys” or “Dual record”, which may reflect unique methodologies or regional programs. These minor types could affect consistency when comparing across countries or years, so it's best to review their distribution and treat them with care in country-wise analysis.

```
plt.figure(figsize=(8, 8))
# Convert 'DataProcess' column to numerical representation using value_counts
data = df['DataProcess'].value_counts()
# Plot the pie chart using the numerical data
data.plot(kind='pie', autopct='%1.1f%%', startangle=90, counterclock=False, shadow=True)
plt.title("Distribution of Data Collection Methods", fontsize=14)
plt.ylabel("") # Hide y-label
```

```
plt.tight_layout()
plt.show()
```



Distribution of Data Collection Methods



```
df['Data Source'].value_counts()
```



	count
Data Source	
UNSD	178039
DHS_STATcompiler	20121
National statistics	19184
DHS_HH	18616
US Census Bureau	7998
INED	6879
MICS	6742
MICS_HH	6695
Eurostat	2873
GGG	1760
IPUMS	1474
GFHS	440
PAPFAM	352
RHS	227
PAPCHILD	204

dtype: int64

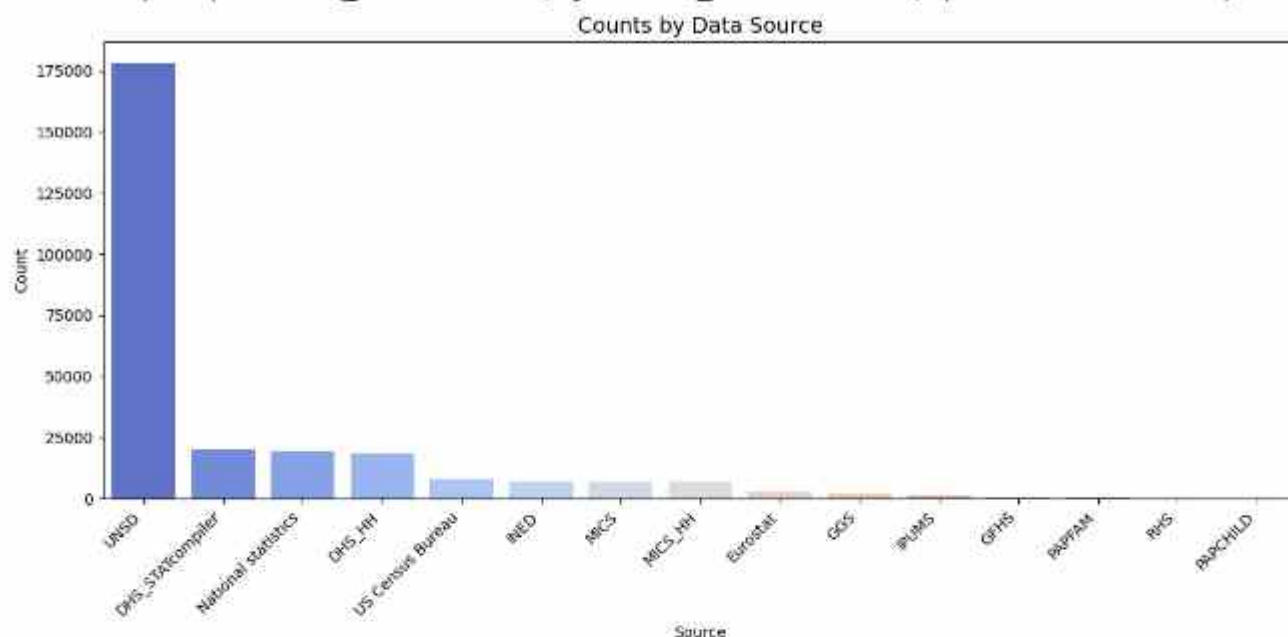
```
source_counts = df['Data Source'].value_counts()

plt.figure(figsize=(12, 6))
sns.barplot(x=source_counts.index, y=source_counts.values, palette='coolwarm')
plt.title("Counts by Data Source", fontsize=14)
plt.xlabel("Source")
plt.ylabel("Count")
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```


 <ipython-input-83-d42dcb0d1d75>:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.

```
sns.barplot(x=source_counts.index, y=source_counts.values, palette='coolwarm')
```

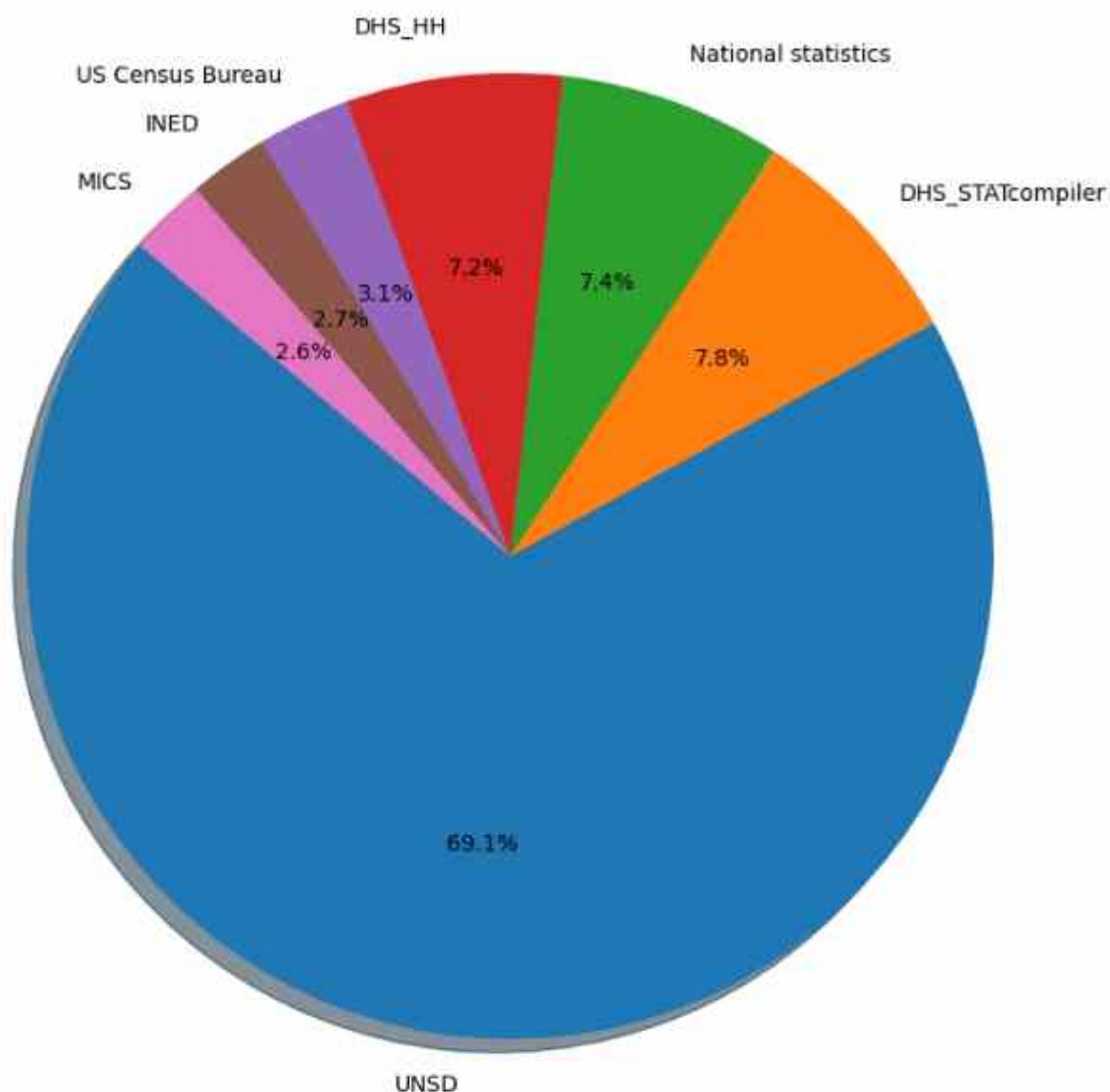


```
top_sources = source_counts.head(7)
```

```
plt.figure(figsize=(8, 8))
top_sources.plot(kind='pie', autopct='%1.1f%%', startangle=140, shadow=True)
plt.title("Top 7 Data Sources - Proportion", fontsize=14)
plt.ylabel("") # Hides the y-axis label
plt.tight_layout()
plt.show()
```



Top 7 Data Sources – Proportion



▼ insights

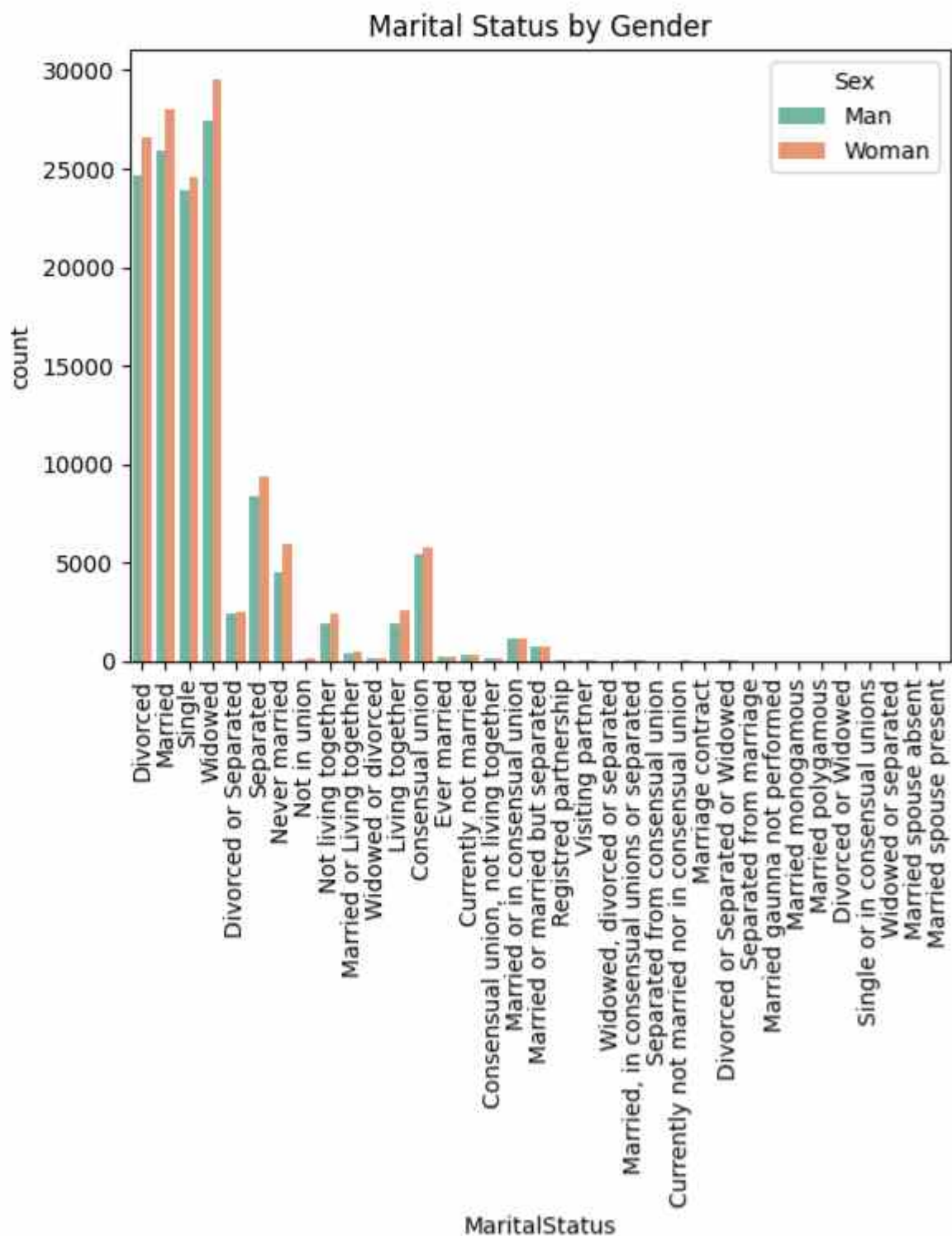
The dataset draws heavily from reputable sources such as the UNSD, DHS, and national statistical bodies, ensuring broad international coverage. Most data is collected through Census (39%), Estimates (34%), and Surveys (26%), offering a reliable blend of official counts and modeled projections. Analysts should still be cautious about comparability, especially when mixing survey-based data with census-derived statistics.

```
d1=df[df.MaritalStatus=='Married']
```

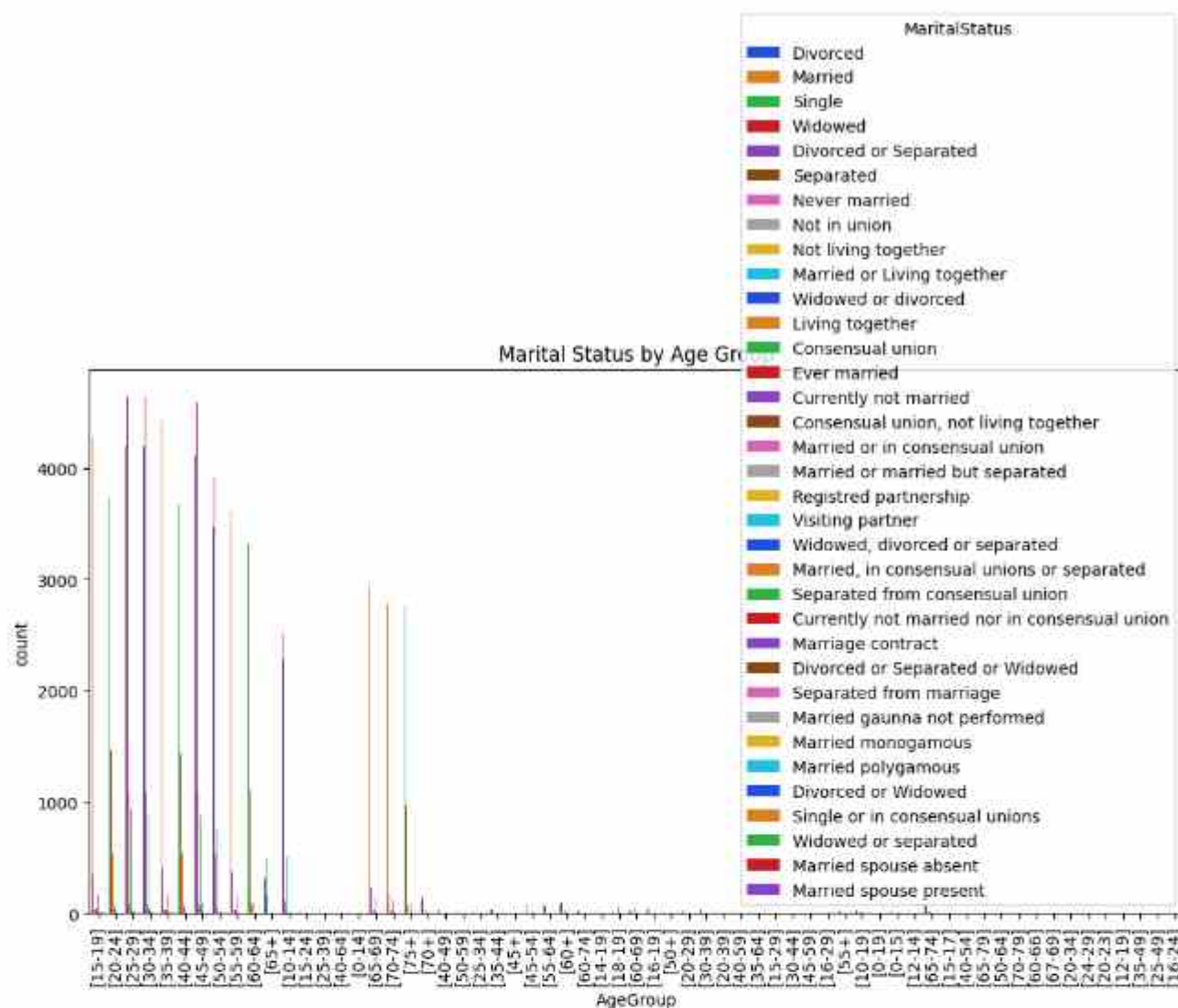


	Country	AgeGroup	Sex	MaritalStatus	DataProcess	Data Collection (Start Year)	Collect (End Year)
11	Afghanistan	[15-19]	Man	Married	Survey	1972	1
12	Afghanistan	[20-24]	Man	Married	Survey	1972	1
13	Afghanistan	[25-29]	Man	Married	Survey	1972	1
14	Afghanistan	[30-34]	Man	Married	Survey	1972	1
15	Afghanistan	[35-39]	Man	Married	Survey	1972	1
...
271575	Zimbabwe	[55-59]	Woman	Married	Survey	2017	2

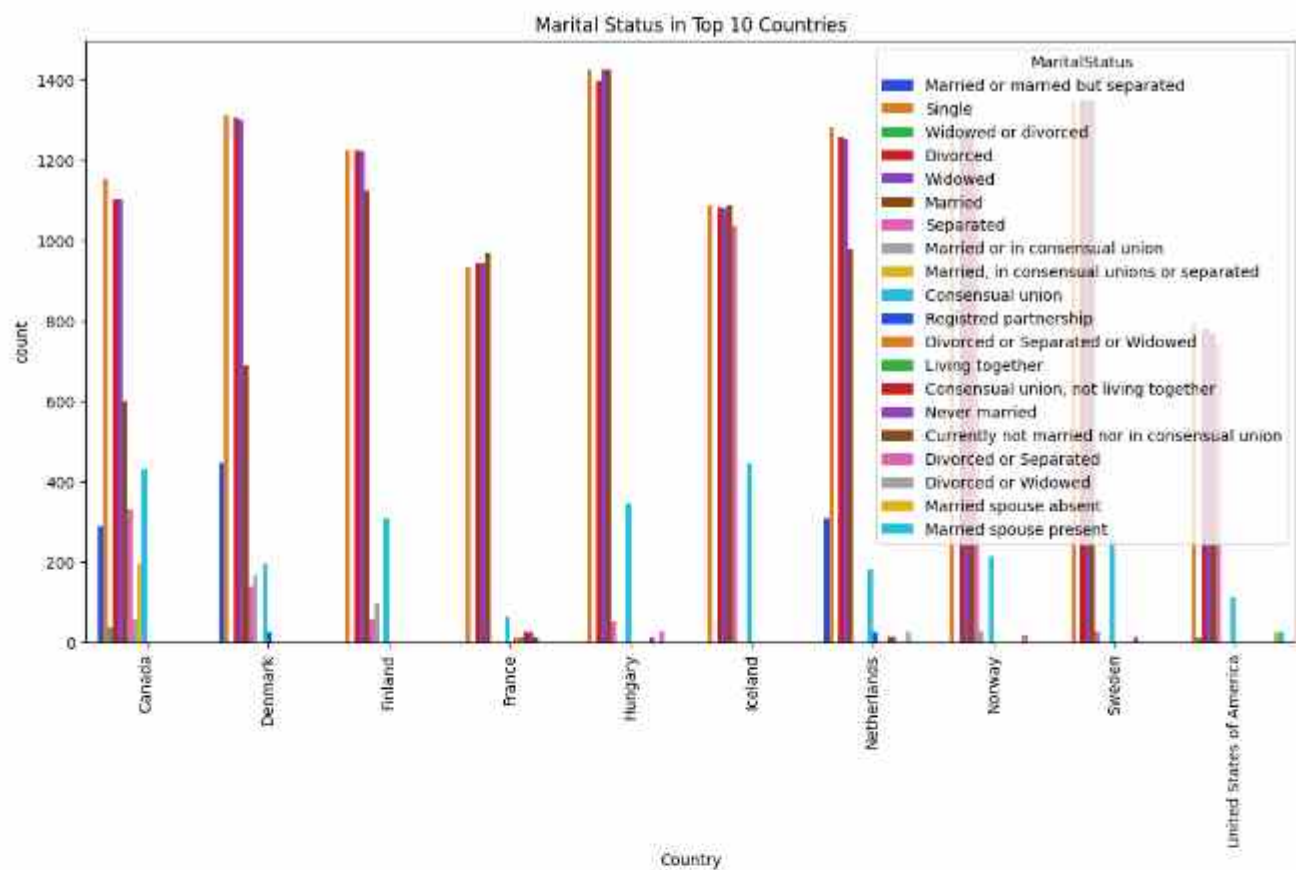
```
sns.countplot(data=df, x='MaritalStatus', hue='Sex',palette= 'Set2')
plt.title('Marital Status by Gender')
plt.xticks(rotation=90)
plt.show()
```



```
plt.figure(figsize=(12,6))
sns.countplot(data=df, x='AgeGroup', hue='MaritalStatus',palette= 'bright')
plt.title('Marital Status by Age Group')
plt.xticks(rotation=90)
plt.show()
```



```
# top 10 countries
top_countries=df['Country'].value_counts().head(10).index
df_top = df[df['Country'].isin(top_countries)]
plt.figure(figsize=(14,7))
sns.countplot(data=df_top, x='Country', hue='MaritalStatus', palette='bright')
plt.title('Marital Status in Top 10 Countries')
plt.xticks(rotation=90)
plt.show()
```



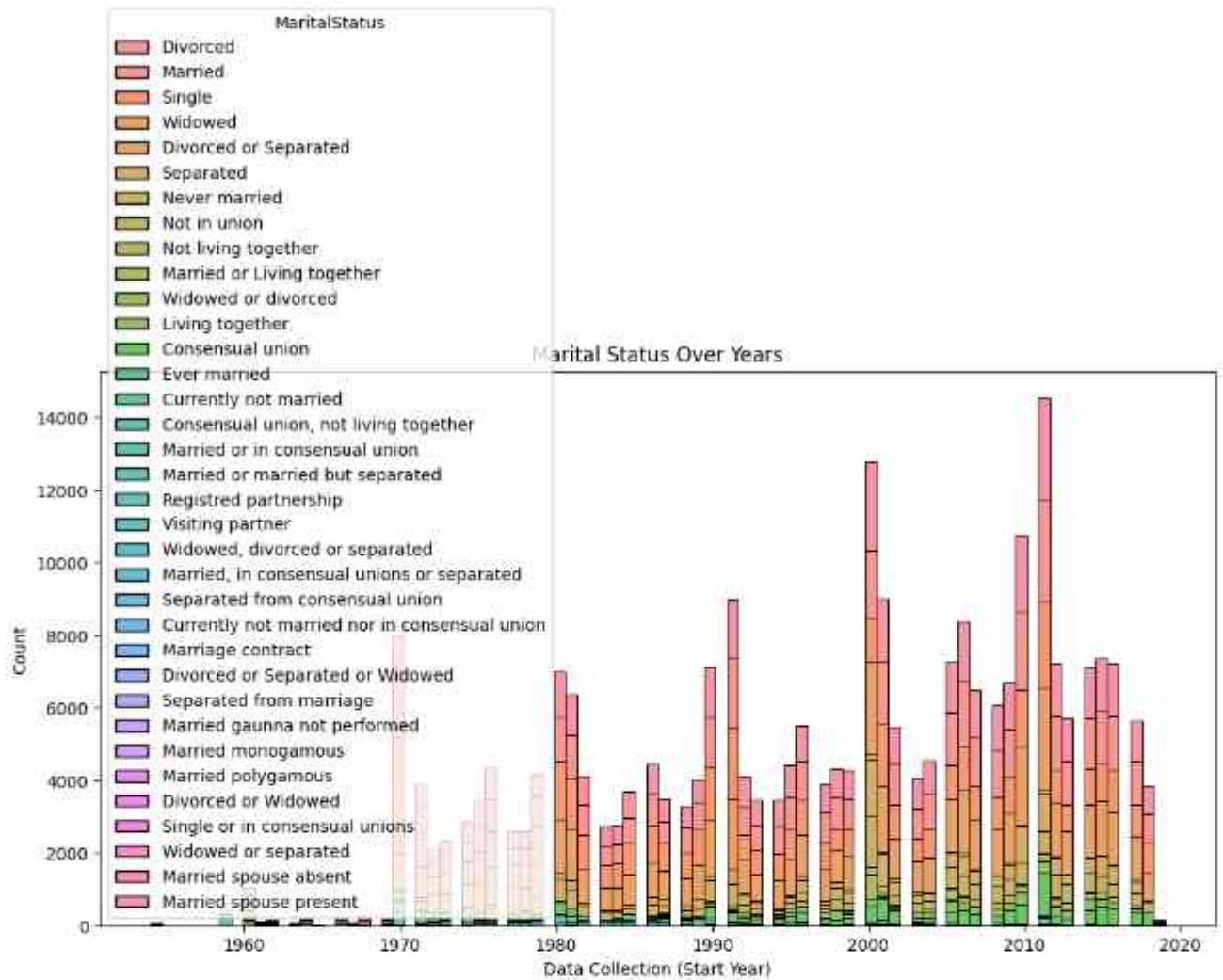
#Trend Over Time

```
plt.figure(figsize=(12,6))
```

```
sns.histplot(data=df, x='Data Collection (Start Year)', hue='MaritalStatus', multiple='st
```

```
plt.title("Marital Status Over Years")
```

```
plt.show()
```

```
age_gender_status = pd.crosstab(
    index=[df['AgeGroup'], df['Sex']],
    columns=df['MaritalStatus'],
    normalize='index') * 100
```

```
print("Marital Status % by Age Group and Gender:")
print(age_gender_status.round(1))
```



```
Marital Status % by Age Group and Gender:
MaritalStatus  Consensual union  Consensual union, not living together \
AgeGroup Sex
[0-14]  Man                13.3                        0.0
        Woman              6.7                        0.0
[0-15]  Man                8.6                        0.0
        Woman              8.6                        0.0
[0-19]  Man               16.7                        0.0
...
[70-74] Woman              4.4                        0.0
[70-79] Man                0.0                        0.0
        Woman              0.0                        0.0
[75+]  Man                4.4                        0.0
        Woman              4.4                        0.0

MaritalStatus  Currently not married \
AgeGroup Sex
[0-14]  Man                0.0
        Woman              0.0
```


[0-15]	Man	0.0
	Woman	0.0
[0-19]	Man	0.0
...		...
[70-74]	Woman	0.0
[70-79]	Man	0.0
	Woman	0.0
[75+]	Man	0.0
	Woman	0.0

MaritalStatus	AgeGroup	Sex	Currently not married nor in consensual union	Divorced	\
	[0-14]	Man	0.0	13.3	
		Woman	0.0	13.3	
	[0-15]	Man	0.0	17.1	
		Woman	0.0	17.1	
	[0-19]	Man	0.0	16.7	
	
	[70-74]	Woman	0.0	20.2	
	[70-79]	Man	0.0	28.6	
		Woman	0.0	28.6	
...	