In [38]:
```
#JUPYTER NOTEBOOK
                                        #Mental Fitness Tracker

#STEP-1:Import all packages
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.express as px
import warnings
warnings.filterwarnings('ignore')
#Reading datasets
df1 = pd.read_csv('mental-and-substance-use-as-share-of-disease.csv')
df2=pd.read_csv("prevalence-by-mental-and-substance-use-disorder.csv")
```

In [4]:
```
#printing dataset-1
df1.head()
```

Out[4]:

	Entity	Code	Year	DALYs (Disability-Adjusted Life Years) - Mental disorders - Sex: Both - Age: All Ages (Percent)
0	Afghanistan	AFG	1990	1.696670
1	Afghanistan	AFG	1991	1.734281
2	Afghanistan	AFG	1992	1.791189
3	Afghanistan	AFG	1993	1.776779
4	Afghanistan	AFG	1994	1.712986

In [5]:
```
#printing dataset-2
df2.head()
```

Out[5]:

	Entity	Code	Year	Prevalence - Schizophrenia - Sex: Both - Age: Age-standardized (Percent)	Prevalence - Bipolar disorder - Sex: Both - Age: Age-standardized (Percent)	Prevalence - Eating disorders - Sex: Both - Age: Age-standardized (Percent)	Prevalence - Anxiety disorders - Sex: Both - Age: Age-standardized (Percent)	Prevalence - Drug use disorders - Sex: Both - Age: Age-standardized (Percent)	Prevalence - Depressive disorders - Sex: Both - Age: Age-standardized (Percent)	Prevalence - Alcohol use disorders - Sex: Both - Age: Age-standardized (Percent)
0	Afghanistan	AFG	1990	0.228979	0.721207	0.131001	4.835127	0.454202	5.125291	0.444036
1	Afghanistan	AFG	1991	0.228120	0.719952	0.126395	4.821765	0.447112	5.116306	0.444250
2	Afghanistan	AFG	1992	0.227328	0.718418	0.121832	4.801434	0.441190	5.106558	0.445501
3	Afghanistan	AFG	1993	0.226468	0.717452	0.117942	4.789363	0.435581	5.100328	0.445958
4	Afghanistan	AFG	1994	0.225567	0.717012	0.114547	4.784923	0.431822	5.099424	0.445779

In [6]:
```
#Merging 2 data-sets
data = pd.merge(df1, df2)
data.head()
```

Out[6]:

	Entity	Code	Year	DALYs (Disability-Adjusted Life Years) - Mental disorders - Sex: Both - Age: All Ages (Percent)	Prevalence - Schizophrenia - Sex: Both - Age: Age-standardized (Percent)	Prevalence - Bipolar disorder - Sex: Both - Age: Age-standardized (Percent)	Prevalence - Eating disorders - Sex: Both - Age: Age-standardized (Percent)	Prevalence - Anxiety disorders - Sex: Both - Age: Age-standardized (Percent)	Prevalence - Drug use disorders - Sex: Both - Age: Age-standardized (Percent)	Prevalence - Depressive disorders - Sex: Both - Age: Age-standardized (Percent)	Prevalence - Alcohol use disorders - Sex: Both - Age: Age-standardized (Percent)
0	Afghanistan	AFG	1990	1.696670	0.228979	0.721207	0.131001	4.835127	0.454202	5.125291	0.444036
1	Afghanistan	AFG	1991	1.734281	0.228120	0.719952	0.126395	4.821765	0.447112	5.116306	0.444250
2	Afghanistan	AFG	1992	1.791189	0.227328	0.718418	0.121832	4.801434	0.441190	5.106558	0.445501
3	Afghanistan	AFG	1993	1.776779	0.226468	0.717452	0.117942	4.789363	0.435581	5.100328	0.445958
4	Afghanistan	AFG	1994	1.712986	0.225567	0.717012	0.114547	4.784923	0.431822	5.099424	0.445779

In [7]:
```
data.isnull().sum()
```

Out[7]:
```
Entity                                                                    0
Code                                                                    690
Year                                                                      0
DALYs (Disability-Adjusted Life Years) - Mental disorders - Sex: Both - Age: All Ages (Percent)      0
Prevalence - Schizophrenia - Sex: Both - Age: Age-standardized (Percent)                             0
Prevalence - Bipolar disorder - Sex: Both - Age: Age-standardized (Percent)                          0
Prevalence - Eating disorders - Sex: Both - Age: Age-standardized (Percent)                          0
Prevalence - Anxiety disorders - Sex: Both - Age: Age-standardized (Percent)                         0
Prevalence - Drug use disorders - Sex: Both - Age: Age-standardized (Percent)                        0
Prevalence - Depressive disorders - Sex: Both - Age: Age-standardized (Percent)                      0
Prevalence - Alcohol use disorders - Sex: Both - Age: Age-standardized (Percent)                     0
dtype: int64
```

In [8]:
```
#finding data size and data shape
data.size,data.shape
```

Out[8]:
```
(75240, (6840, 11))
```

In [9]:
```
#changing columns names using rename method and using dictionaries
data = data.rename(columns={'Entity': 'Country', 'Year': 'Year', 'DALYs (Disability-Adjusted Life Years) - Mental diso
data.head()
```

Out[9]:

	Country	Code	Year	Schizophrenia	Bipolar_disorder	Eating_disorder	Anxiety	drug_usage	depression	alcohol	mental_fitness
0	Afghanistan	AFG	1990	1.696670	0.228979	0.721207	0.131001	4.835127	0.454202	5.125291	0.444036
1	Afghanistan	AFG	1991	1.734281	0.228120	0.719952	0.126395	4.821765	0.447112	5.116306	0.444250
2	Afghanistan	AFG	1992	1.791189	0.227328	0.718418	0.121832	4.801434	0.441190	5.106558	0.445501
3	Afghanistan	AFG	1993	1.776779	0.226468	0.717452	0.117942	4.789363	0.435581	5.100328	0.445958
4	Afghanistan	AFG	1994	1.712986	0.225567	0.717012	0.114547	4.784923	0.431822	5.099424	0.445779

In [21]:
```
data.drop('Code',axis=1,inplace=True)
```

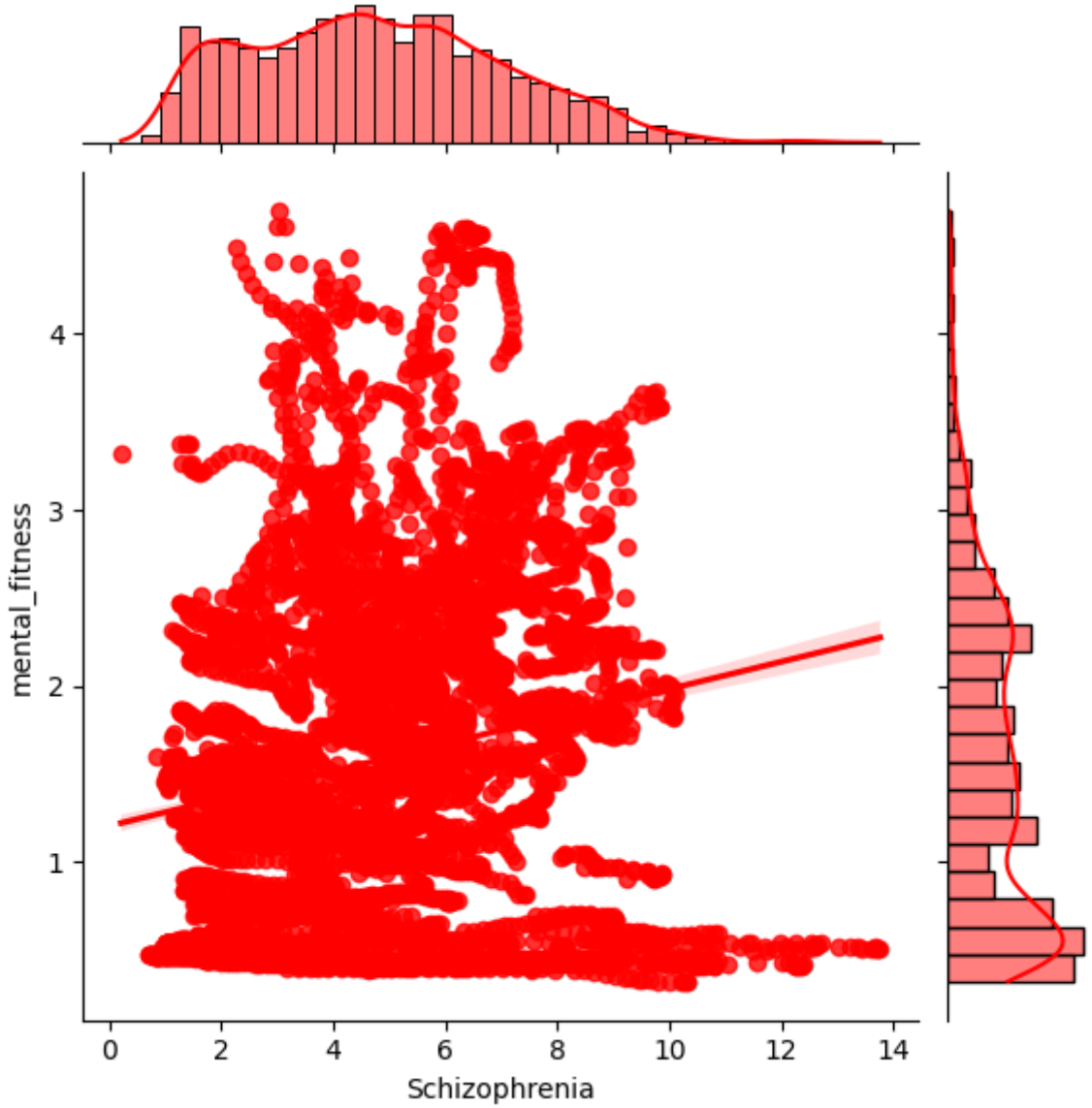
In [22]:
```
data.head()
```

Out[22]:

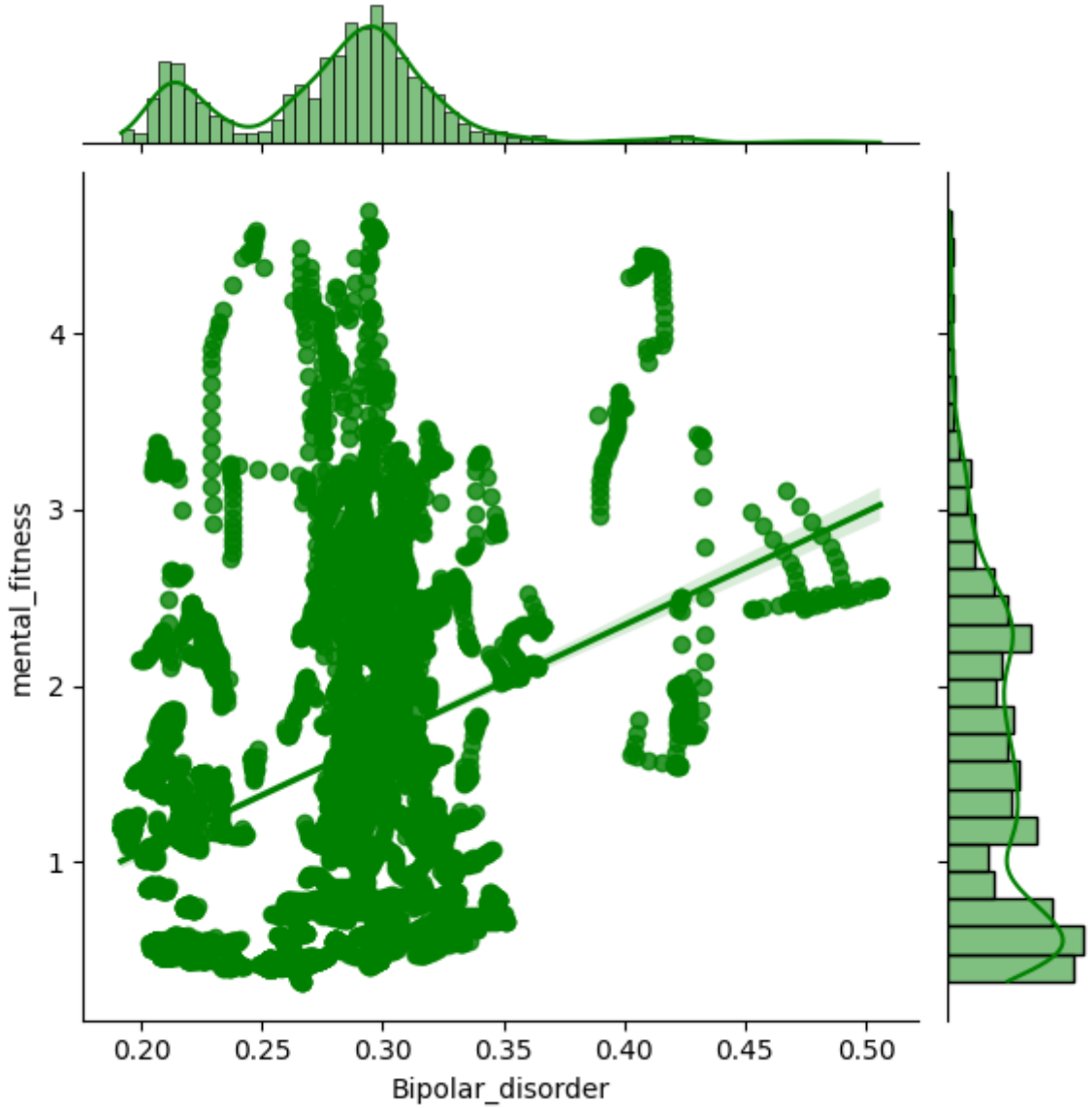
	Country	Year	Schizophrenia	Bipolar_disorder	Eating_disorder	Anxiety	drug_usage	depression	alcohol	mental_fitness
0	Afghanistan	1990	1.696670	0.228979	0.721207	0.131001	4.835127	0.454202	5.125291	0.444036
1	Afghanistan	1991	1.734281	0.228120	0.719952	0.126395	4.821765	0.447112	5.116306	0.444250
2	Afghanistan	1992	1.791189	0.227328	0.718418	0.121832	4.801434	0.441190	5.106558	0.445501
3	Afghanistan	1993	1.776779	0.226468	0.717452	0.117942	4.789363	0.435581	5.100328	0.445958
4	Afghanistan	1994	1.712986	0.225567	0.717012	0.114547	4.784923	0.431822	5.099424	0.445779

In [ ]:
```
plt.figure(figsize=(12,6))
sns.heatmap(data.corr(),annot=True,cmap='Blues')
plt.plot()
```

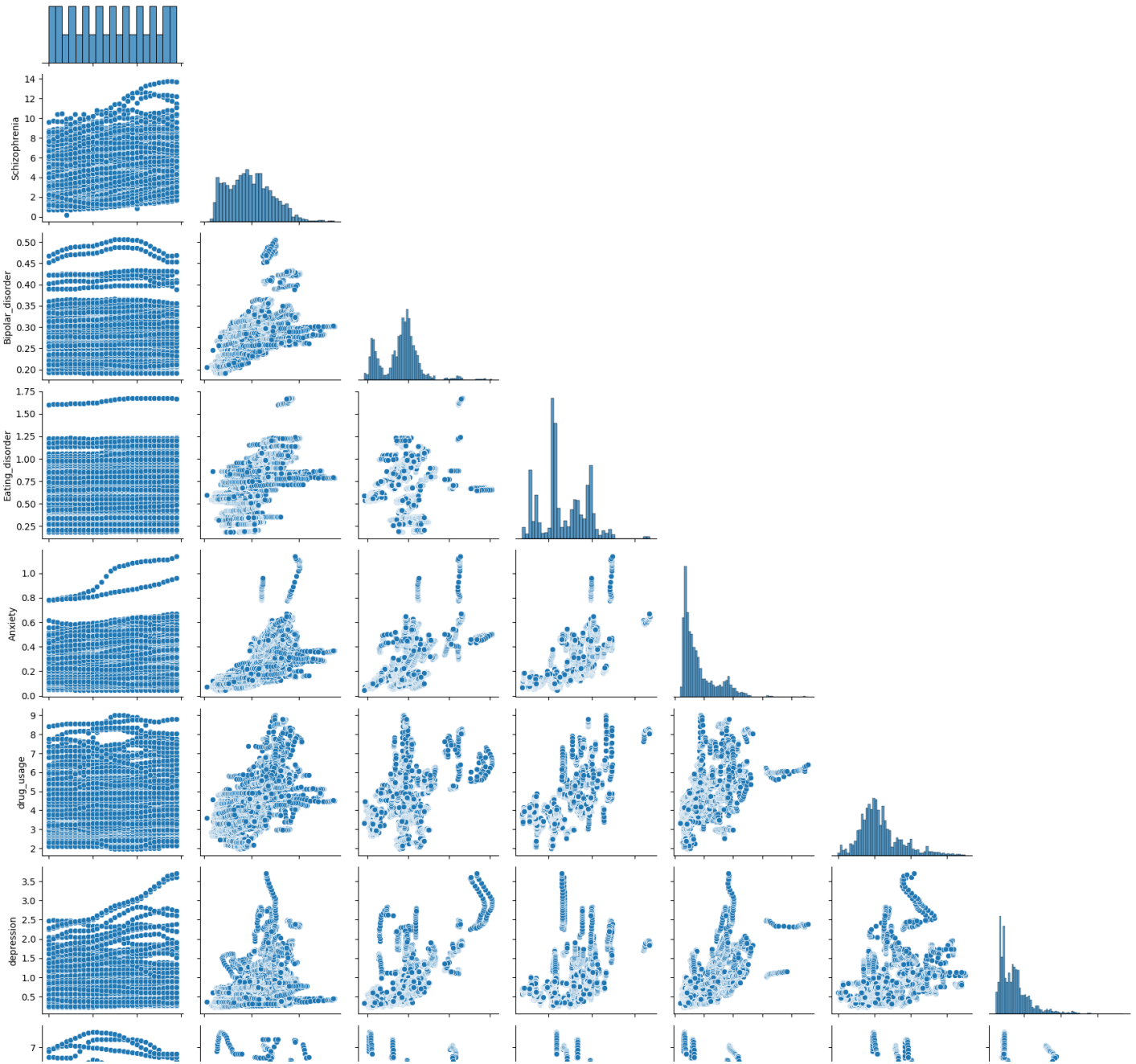
In [25]:
```
#plotting different graphs
#Graph-1
sns.jointplot(data,x="Schizophrenia",y="mental_fitness",kind="reg",color="red")
plt.show()
```

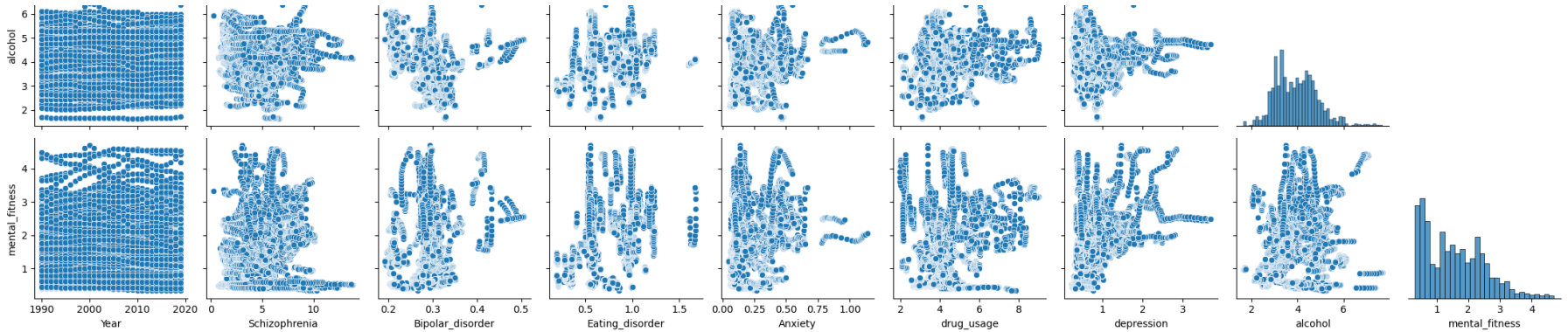


In [27]:
```
#Graph-2
sns.jointplot(data,x='Bipolar_disorder',y='mental_fitness',kind='reg',color='green')
plt.show()
```



In [28]:
```
sns.pairplot(data,corner=True)
plt.show()
```

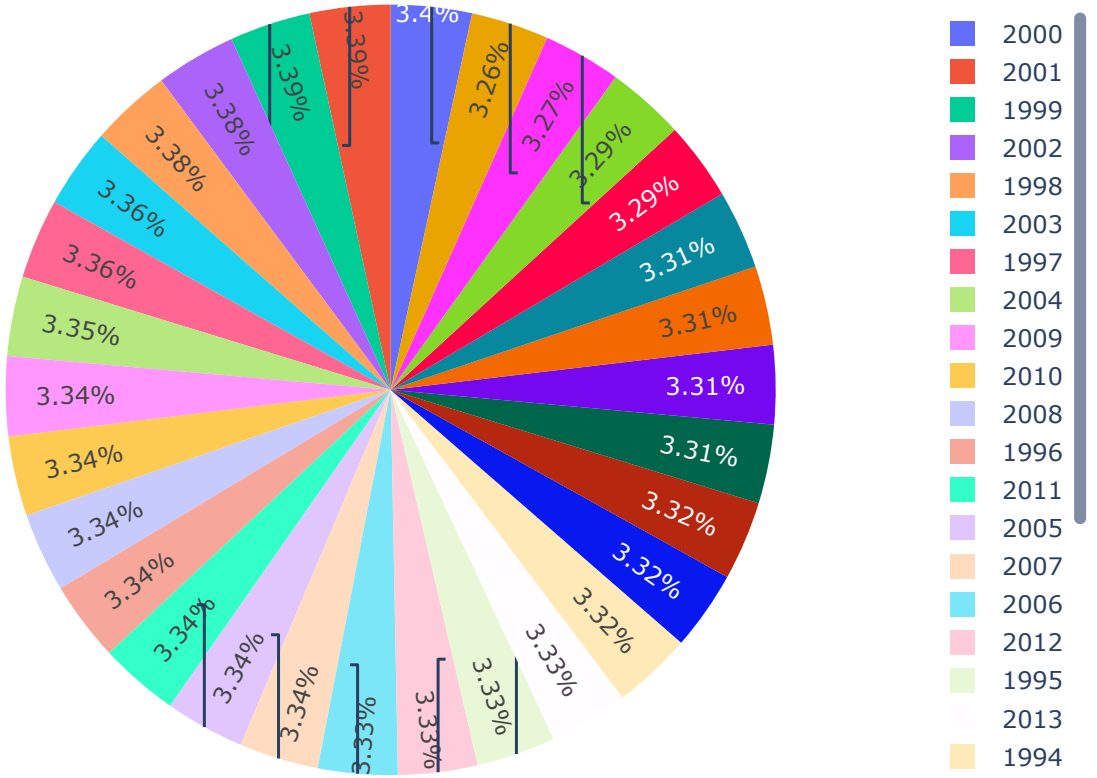

Schizophrenia
Bipolar_disorder
Eating_disorder
Anxiety
drug_usage
depression



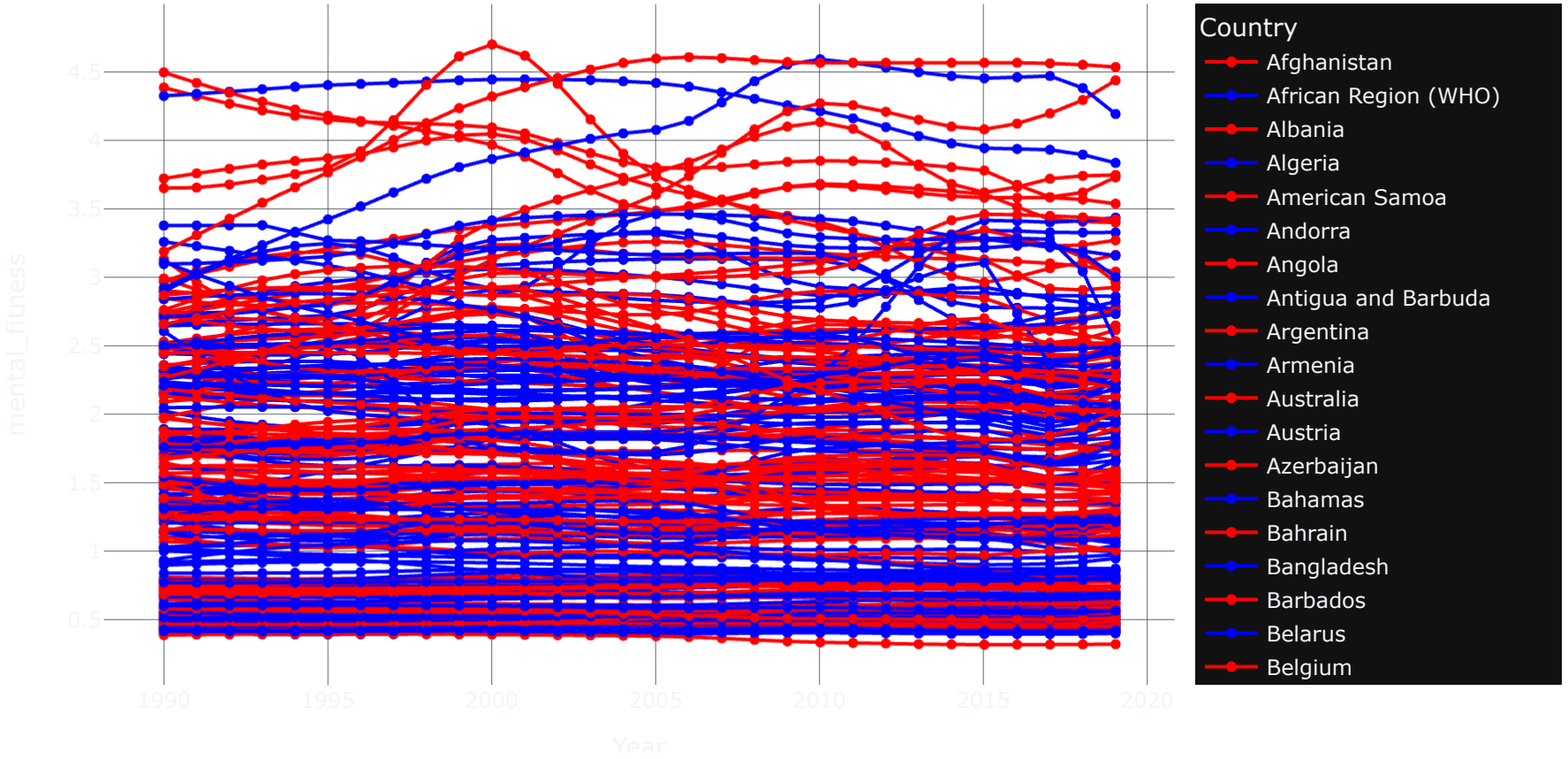
In [29]:
```
#calculating mean
mean = data['mental_fitness'].mean()
mean
```

Out[29]: 1.5788071625382236

In [30]:
```
#plotting pie chart
fig = px.pie(data, values='mental_fitness', names='Year')
fig.show()
```



In [31]:
```
fig = px.line(data, x="Year", y="mental_fitness", color='Country',markers=True,color_discrete_sequence=['red','blue'],
fig.show()
```



In [32]:
```
df=data.copy()
df.head()
```

Out[32]:

	Country	Year	Schizophrenia	Bipolar_disorder	Eating_disorder	Anxiety	drug_usage	depression	alcohol	mental_fitness
0	Afghanistan	1990	1.696670	0.228979	0.721207	0.131001	4.835127	0.454202	5.125291	0.444036
1	Afghanistan	1991	1.734281	0.228120	0.719952	0.126395	4.821765	0.447112	5.116306	0.444250
2	Afghanistan	1992	1.791189	0.227328	0.718418	0.121832	4.801434	0.441190	5.106558	0.445501
3	Afghanistan	1993	1.776779	0.226468	0.717452	0.117942	4.789363	0.435581	5.100328	0.445958
4	Afghanistan	1994	1.712986	0.225567	0.717012	0.114547	4.784923	0.431822	5.099424	0.445779

In [33]:
```
#Obtaining Results
from sklearn.preprocessing import LabelEncoder
l=LabelEncoder()
for i in df.columns:
    if df[i].dtype == 'object':
        df[i]=l.fit_transform(df[i])
```

In [34]:
```
X = df.drop('mental_fitness',axis=1)
y = df['mental_fitness']

from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(X, y, test_size=0.2, random_state=2)
```

In [35]:

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
lr = LinearRegression()
lr.fit(xtrain,ytrain)
# model evaluation for training set



ytrain_pred = lr.predict(xtrain)
mse = mean_squared_error(ytrain, ytrain_pred)
rmse = (np.sqrt(mean_squared_error(ytrain, ytrain_pred)))
r2 = r2_score(ytrain, ytrain_pred)




print("The model performance for training set:")
print("MSE is ",mse)
print("RMSE is",rmse)
print("R2 score is",r2)
print("\n")
# model evaluation for testing set


ytest_pred = lr.predict(xtest)
mse = mean_squared_error(ytest, ytest_pred)
rmse = (np.sqrt(mean_squared_error(ytest, ytest_pred)))
r2 = r2_score(ytest, ytest_pred)


print("The model performance for testing set:")
print("MSE is ",mse)
print("RMSE is",rmse)
print("R2 score is",r2)
print("\n")
```

```
The model performance for training set:
MSE is  0.5768675399745626
RMSE is 0.7595179655377235
R2 score is 0.33581211672302613


The model performance for testing set:
MSE is  0.5792230513592
RMSE is 0.7610670478737074
R2 score is 0.35130869036637036
```

In [37]:

```
from sklearn.ensemble import RandomForestRegressor
rf = RandomForestRegressor()
rf.fit(xtrain, ytrain)
# model evaluation for training set


ytrain_pred = rf.predict(xtrain)
mse = mean_squared_error(ytrain, ytrain_pred)
rmse = (np.sqrt(mean_squared_error(ytrain, ytrain_pred)))
r2 = r2_score(ytrain, ytrain_pred)


print("The model performance for training set")
print("MSE is ",mse)
print("RMSE is",rmse)
print("R2 score is",r2)
print("\n")
# model evaluation for testing set


ytest_pred = rf.predict(xtest)
mse = mean_squared_error(ytest, ytest_pred)
rmse = (np.sqrt(mean_squared_error(ytest, ytest_pred)))
r2 = r2_score(ytest, ytest_pred)

print("The model performance for testing set:")
print("MSE is ",mse)
print("RMSE is",rmse)
print("R2 score is",r2)
print("\n")
```

```
The model performance for training set
MSE is  0.000587511483170873
RMSE is 0.02423863616565241
R2 score is 0.9993235570016206


The model performance for testing set:
MSE is  0.0035753998435258377
RMSE is 0.059794647281557214
R2 score is 0.9959957898748709
```

In [ ]: