```
import warnings
warnings.filterwarnings('ignore')
```

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

```
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.express as px
import warnings
warnings.filterwarnings('ignore')
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
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")
```

```
df1.head()
```

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

```
df2.tail()
```

	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)
6835	Zimbabwe	ZWE	2015	0.209359	0.560882	0.099610
6836	Zimbabwe	ZWE	2016	0.209979	0.561768	0.100821
6837	Zimbabwe	ZWE	2017	0.210631	0.562612	0.101671
6838	Zimbabwe	ZWE	2018	0.211237	0.563283	0.102398
6839	Zimbabwe	ZWE	2019	0.211969	0.563820	0.102902

```
df1.head(10)
```

	Entity	Code	Year	DALYs (Disability-Adjusted Life Years) - Mental disorders - Sex: Both - Age: All Ages (Percent)
0	Afghanistan	AFG	1990	1.696670

```
data = pd.merge(df1, df2)
data.head()
```

	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)	Pr
0	Afghanistan	AFG	1990	1.696670	0.228979	0.721207	
1	Afghanistan	AFG	1991	1.734281	0.228120	0.719952	
2	Afghanistan	AFG	1992	1.791189	0.227328	0.718418	
3	Afghanistan	AFG	1993	1.776779	0.226468	0.717452	

```
data.isnull().sum()
```

```
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
```

```
data.drop('Code',axis=1,inplace=True)
```

```
data.head()
```

	Entity	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)	Preval Depr disor Sex: Age standa (Pe
0	Afghanistan	1990	1.696670	0.228979	0.721207	0.131001	4.835127	0.454202	5
1	Afghanistan	1991	1.734281	0.228120	0.719952	0.126395	4.821765	0.447112	5
2	Afghanistan	1992	1.791189	0.227328	0.718418	0.121832	4.801434	0.441190	5
3	Afghanistan	1993	1.776779	0.226468	0.717452	0.117942	4.789363	0.435581	5
4	Afghanistan	1994	1.712986	0.225567	0.717012	0.114547	4.784923	0.431822	5

```
data.size,data.shape
```

```
(68400, (6840, 10))
```

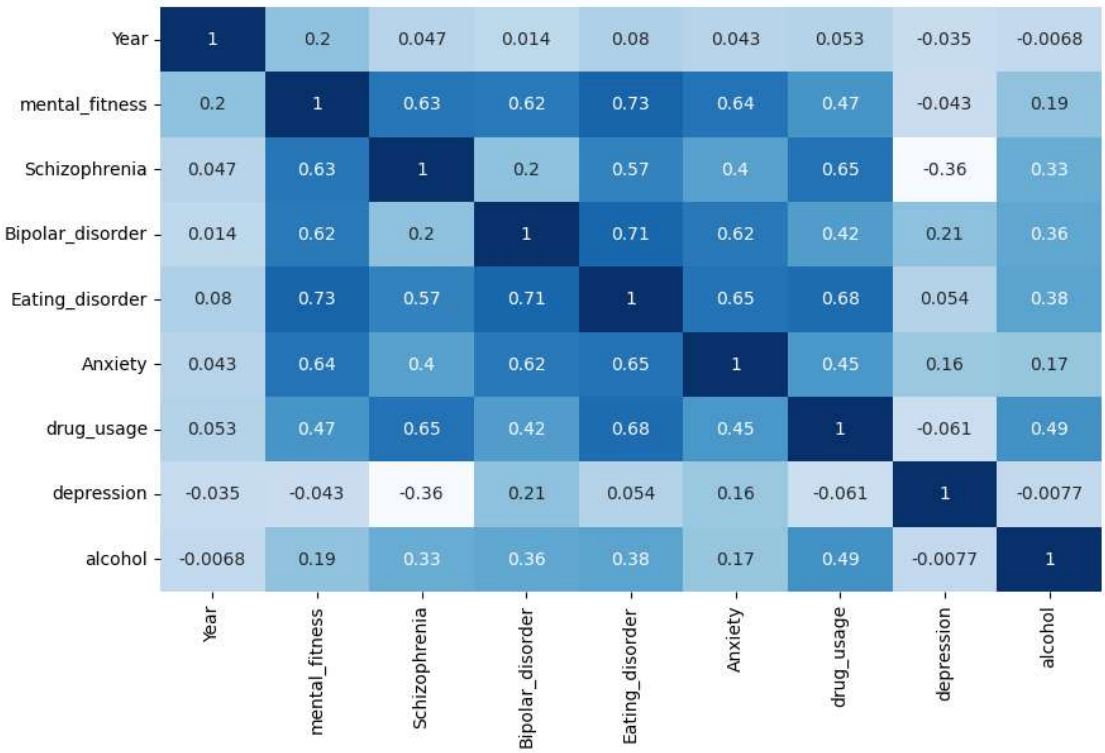
```
data.set_axis(['Country','Year', 'mental_fitness','Schizophrenia', 'Bipolar_disorder', 'Eating_disorder','Anxiety','
```

```
data.head()
```

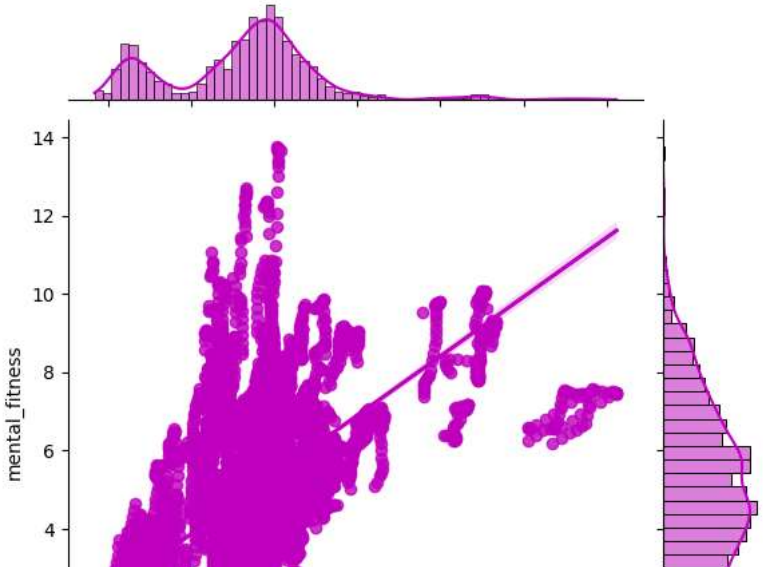
	Country	Year	mental_fitness	Schizophrenia	Bipolar_disorder	Eating_disorder	Anxiety	drug_usage	dep
0	Afghanistan	1990	1.696670	0.228979	0.721207	0.131001	4.835127	0.454202	5
1	Afghanistan	1991	1.734281	0.228120	0.719952	0.126395	4.821765	0.447112	5
2	Afghanistan	1992	1.791189	0.227328	0.718418	0.121832	4.801434	0.441190	5
3	Afghanistan	1993	1.776779	0.226468	0.717452	0.117942	4.789363	0.435581	5
4	Afghanistan	1994	1.712986	0.225567	0.717012	0.114547	4.784923	0.431822	5

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

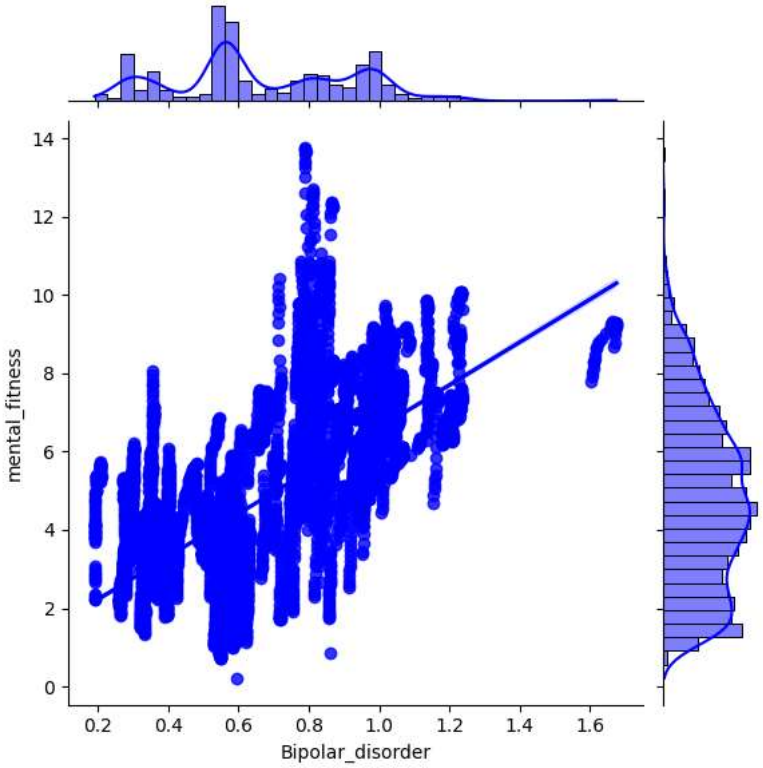
```
[]
```



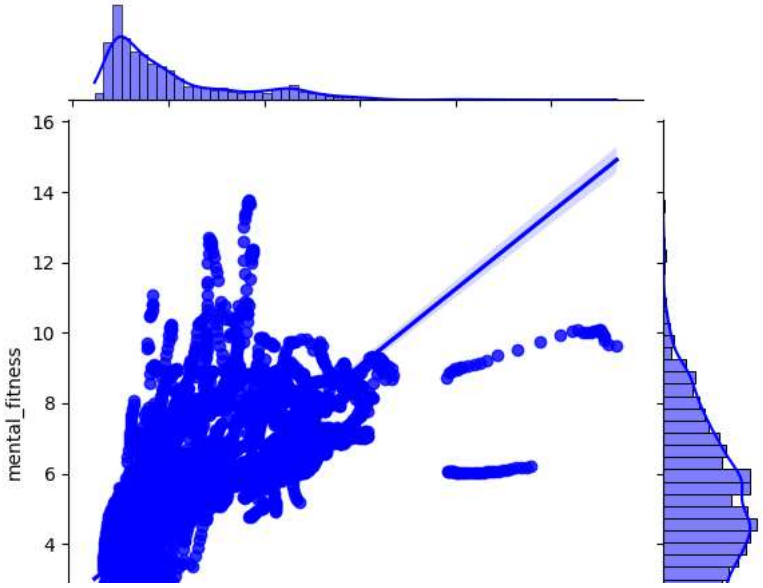

```
sns.jointplot(x='Schizophrenia', y='mental_fitness', data=data, kind='reg', color='m')
plt.show()
```



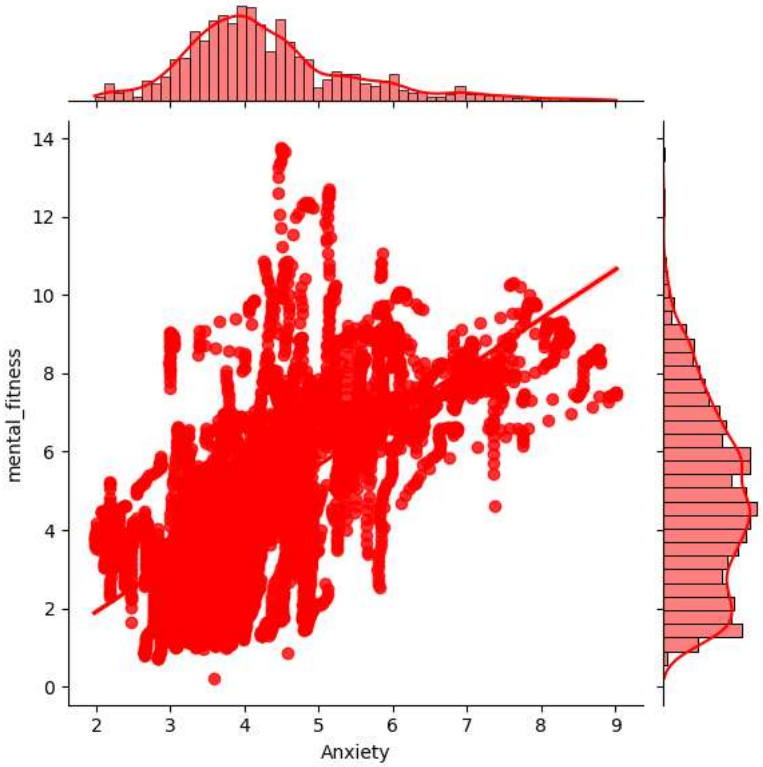

```
sns.jointplot(x='Bipolar_disorder',y='mental_fitness',data=data,kind='reg',color='blue')
plt.show()
```



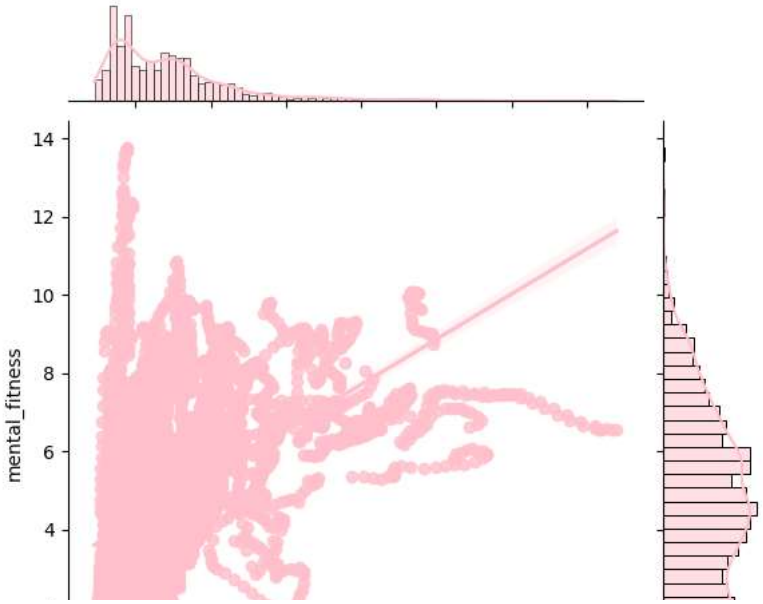

```
sns.jointplot(x='Eating_disorder',y='mental_fitness',data=data,kind='reg',color='blue')
plt.show()
```



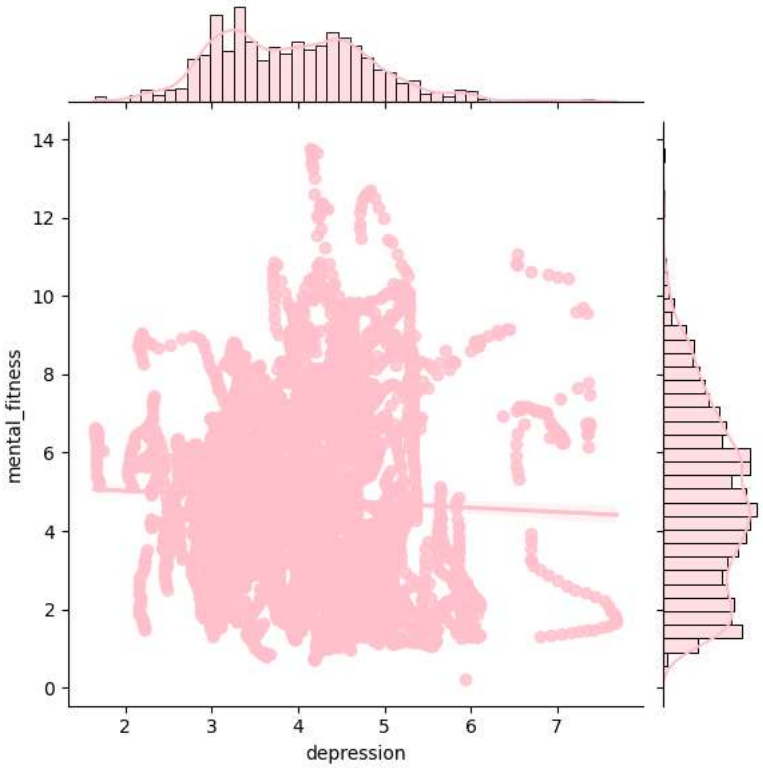

```
sns.jointplot(x='Anxiety',y='mental_fitness',data=data,kind='reg',color='red')
plt.show()
```



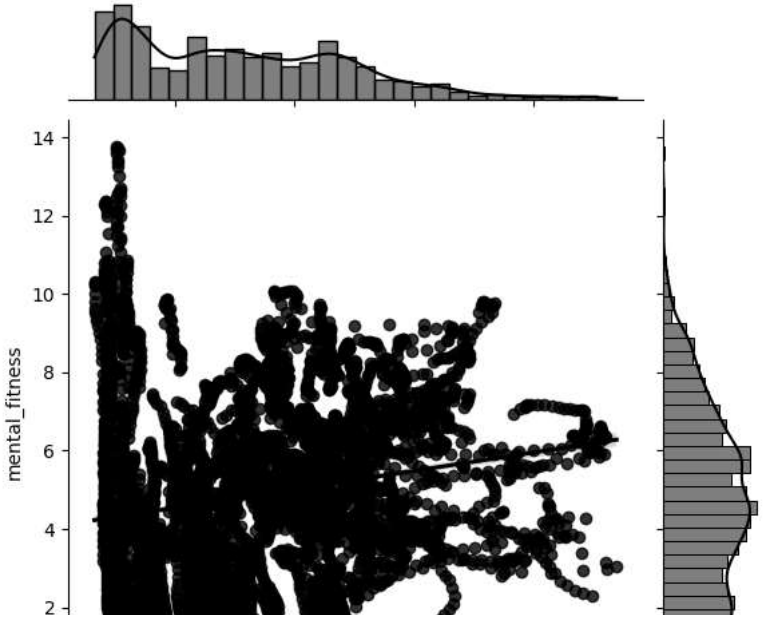

```
sns.jointplot(x='drug_usage',y='mental_fitness',data=data,kind='reg',color='pink')
plt.show()
```



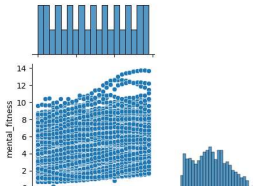

```
sns.jointplot(x='depression',y='mental_fitness',data=data,kind='reg',color='pink')
plt.show()
```




```
sns.jointplot(x='alcohol',y='mental_fitness',data=data,kind='reg',color='black')
plt.show()
```



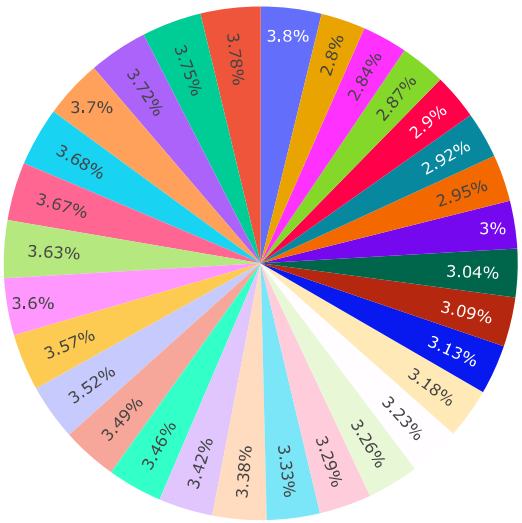

```
sns.pairplot(data,corner=True)
plt.show()
```



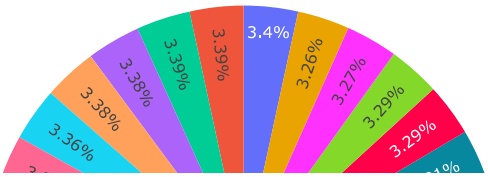

```
mean = data['mental_fitness'].mean()
mean
```

```
4.8180618117506135
```

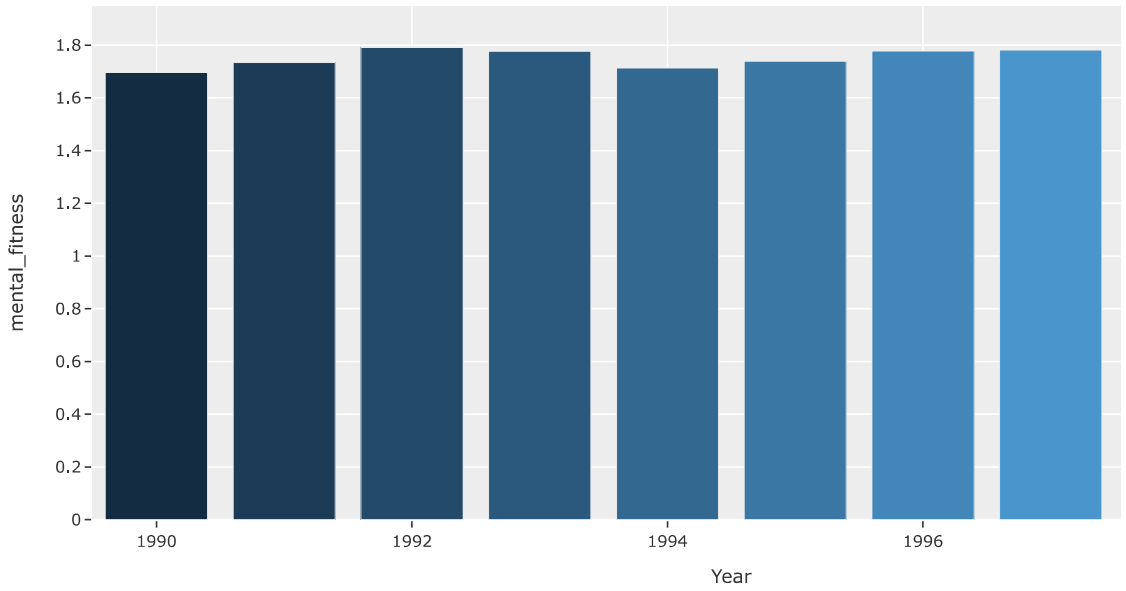
```
fig = px.pie(data, values='mental_fitness', names='Year')
fig.show()
```



```
fig = px.pie(data, values='alcohol', names='Year')
fig.show()
```




```
fig=px.bar(data.head(10),x='Year',y='mental_fitness',color='Year',template='ggplot2')
fig.show()
```




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

```
df = data.copy()
```

```
df.head()
```

	Country	Year	mental_fitness	Schizophrenia	Bipolar_disorder	Eating_disorder	Anxiety	drug_usage	dep
0	Afghanistan	1990	1.696670	0.228979	0.721207	0.131001	4.835127	0.454202	5
1	Afghanistan	1991	1.734281	0.228120	0.719952	0.126395	4.821765	0.447112	5
2	Afghanistan	1992	1.791189	0.227328	0.718418	0.121832	4.801434	0.441190	5
3	Afghanistan	1993	1.776779	0.226468	0.717452	0.117942	4.789363	0.435581	5
4	Afghanistan	1994	1.712986	0.225567	0.717012	0.114547	4.784923	0.431822	5

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 6840 entries, 0 to 6839
Data columns (total 10 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Country           6840 non-null   object 
 1   Year              6840 non-null   int64  
 2   mental_fitness    6840 non-null   float64
 3   Schizophrenia     6840 non-null   float64
 4   Bipolar_disorder  6840 non-null   float64
 5   Eating_disorder   6840 non-null   float64
 6   Anxiety           6840 non-null   float64
 7   drug_usage        6840 non-null   float64
 8   depression        6840 non-null   float64
 9   alcohol           6840 non-null   float64
dtypes: float64(8), int64(1), object(1)
memory usage: 587.8+ KB
```

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


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)


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("--------------------------------------")
print('MSE is {}'.format(mse))
print('RMSE is {}'.format(rmse))
print('R2 score is {}'.format(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("--------------------------------------")
print('MSE is {}'.format(mse))
print('RMSE is {}'.format(rmse))
print('R2 score is {}'.format(r2))
```

```
The model performance for training set
--------------------------------------
MSE is 1.389959372405798
RMSE is 1.1789653821914357
R2 score is 0.7413245790025275


The model performance for testing set
--------------------------------------
MSE is 1.1357545319272384
RMSE is 1.0657178481789813
R2 score is 0.7638974087055272
```

```
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("--------------------------------------")
print('MSE is {}'.format(mse))
print('RMSE is {}'.format(rmse))
print('R2 score is {}'.format(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("--------------------------------------")
print('MSE is {}'.format(mse))
print('RMSE is {}'.format(rmse))
print('R2 score is {}'.format(r2))
```

```
The model performance for training set
--------------------------------------
MSE is 0.005117766767628213
RMSE is 0.0715385683923589
R2 score is 0.9990475689437658


The model performance for testing set
--------------------------------------
MSE is 0.03145219697352911
RMSE is 0.17734767259123846
R2 score is 0.9934616635913806
```