Exploratory Data Analysis (EDA)

This notebook is for analyzing the relationships between different variables to figure out what research question to ask. This is done using various graphs and visualizations. There are labels at the end of each graph describing anything interesting about the graph.

We will also use this initial analysis to determine what variables we want to use in the later parts and determine our research question.

All of the graph outputs are accessible through the folder graphs -> eda

We will be working with the Global Disaster Response Analysis dataset from Kaggle. Questions we intend exploring are related to how long it takes for a country to recover based on the aid amount and disaster type.

Load Data

```
# import libraries
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.express as px
from scipy import stats
```

```
# load dataframe
df = pd.read_csv('data/global_disaster_response_2018_2024.csv')
df
```

```
            date        country disaster_type  severity_index  casualties  \
0      2021-01-31         Brazil    Earthquake            5.99         111
1      2018-12-23         Brazil  Extreme Heat            6.53         100
2      2020-08-10          India     Hurricane            1.55          22
3      2022-09-15      Indonesia  Extreme Heat            4.55          94
4      2022-09-28  United States      Wildfire            3.80          64
...           ...            ...           ...             ...         ...
49995  2019-05-14          Chile     Landslide            5.50          78
49996  2020-10-30  United States      Wildfire            7.76         165
49997  2019-04-27         Turkey         Flood            4.90         130
49998  2022-10-09         Greece   Storm Surge            3.35          82
49999  2023-01-12   South Africa       Drought            5.03         129

       economic_loss_usd  response_time_hours  aid_amount_usd  \
0             7934365.71                15.62       271603.79
1             8307648.99                 5.03       265873.81
2              765136.99                32.54        49356.49
3             1308251.31                 7.83       237512.88
```

```
4             2655864.36                21.90       188910.69
...                  ...                   ...             ...
49995         3711240.93                 8.45       305020.35
49996        12072842.65                 1.00       363881.25
49997         1805859.70                 5.14       280665.61
49998         3176085.56                19.22        80331.23
49999         2933495.70                 9.08       354096.99

       response_efficiency_score  recovery_days  latitude  longitude
0                          83.21             67   -30.613   -122.557
1                          96.18             55    10.859   -159.194
2                          60.40             22     0.643   -160.978
3                          86.41             47   -33.547     30.350
4                          72.81             42   -19.170   -117.137
...                          ...            ...       ...         ...
49995                      94.27             55    12.976    -25.680
49996                      95.46             76    57.265   -147.346
49997                      86.67             47    15.217    -27.856
49998                      84.75             32   -44.002      1.923
49999                      97.16             50    13.084    -81.048

[50000 rows x 12 columns]
```

View unique values of some colums

```
df['disaster_type'].value_counts()
```

```
disaster_type
Landslide            5130
Earthquake           5068
Flood                5039
Hurricane            5002
Extreme Heat         5001
Storm Surge          4988
Volcanic Eruption    4983
Wildfire             4954
Tornado              4939
Drought              4896
Name: count, dtype: int64
```

```
df['country'].value_counts()
```

```
country
Brazil          2591
Australia       2563
Turkey          2554
Bangladesh      2553
Spain           2543
```

```
China            2539
Chile            2529
Nigeria          2528
Germany          2526
India            2509
Greece           2503
Italy            2503
South Africa     2497
Japan            2472
Indonesia        2467
Canada           2438
Philippines      2437
Mexico           2433
United States    2413
France           2402
Name: count, dtype: int64
```

Analyze relationships between variables

Aid and response:

Description of Methods: The following plots are joint plots that include a histogram of the distribution of each variable, along with a hexplot of the relationship between the two variables. The darker the the color in the hexplot, the more data points are in that region of the plot.

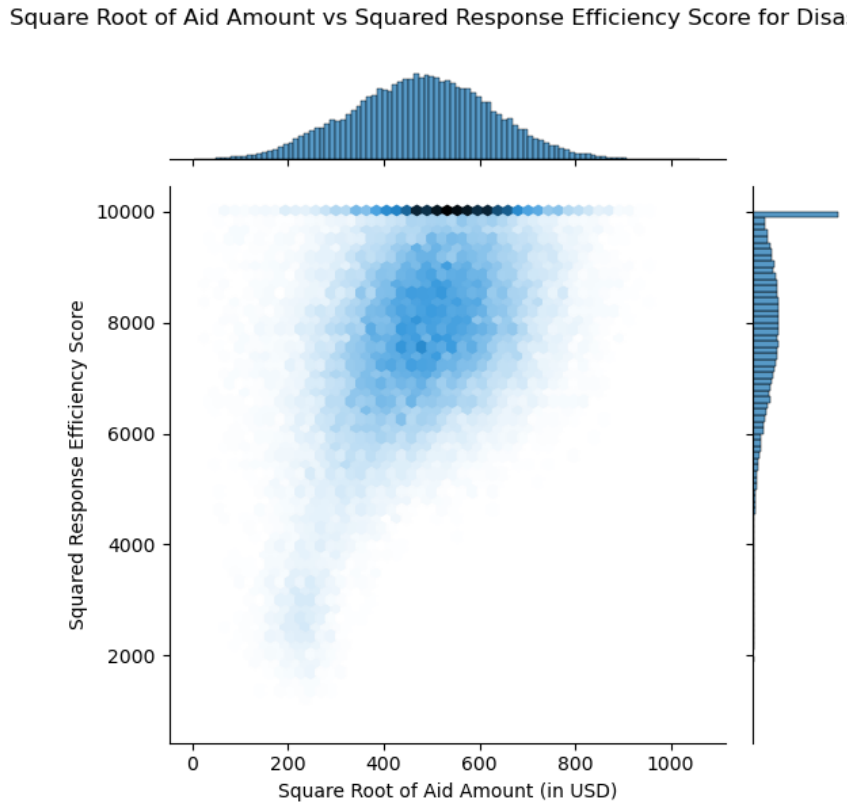
You may notice that we take the square root of some variables. This is to fit a previously curved diagram using a linear model (see the Tuskey-Mosteller Bulging Rule).

To visualize the relationship between aid amount and disaster effects/response times, let's graph the following:

1. Joint plot: aid amount and response efficiency
2. Joint plot: aid amount and response time
3. Joint plot: aid amount and economic loss
4. Joint plot: aid amount and casualties

```
# joint plot of square root of aid amount and response efficiency

g = sns.jointplot(data=df, x=np.sqrt(df['aid_amount_usd']),
                  y=df['response_efficiency_score']**2, kind='hex')
g.fig.suptitle('Square Root of Aid Amount vs Squared Response Efficiency Score for Disasters
# y=1 moves it above
plt.xlabel('Square Root of Aid Amount (in USD)')
plt.ylabel('Squared Response Efficiency Score')
plt.tight_layout()
```

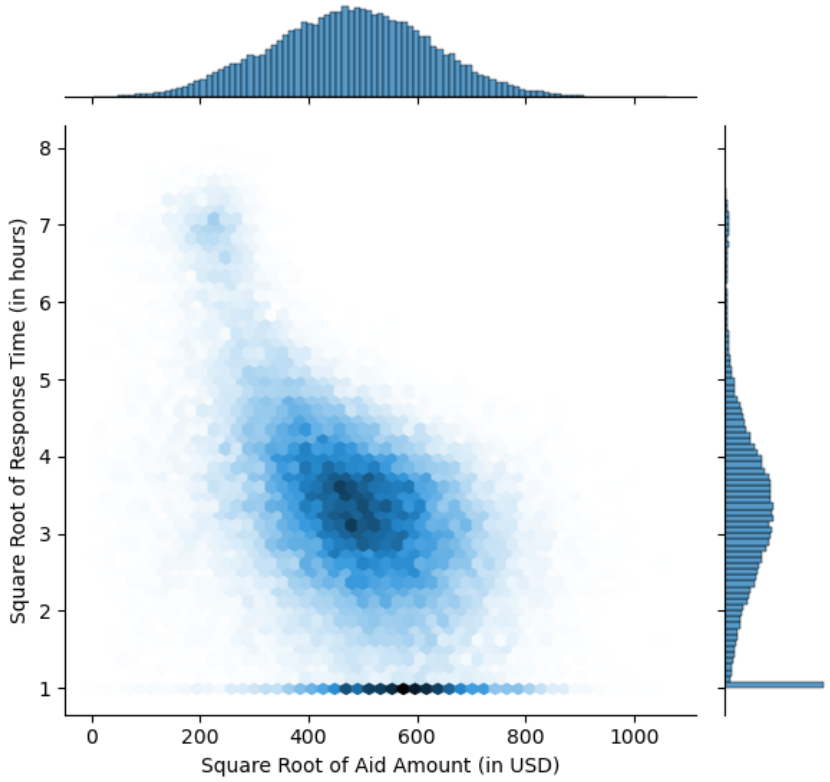
```
# Saving the plot as a JPEG file
plt.savefig("graphs/eda/1_aid_efficiency.jpg", bbox_inches='tight')
```



The original distribution of aid amount is skewed to the right, so after taking its square root, it becomes roughly normal. The distribution of the response efficiency score is more dense around 70-100, with a very high concentration at 100. Below 100, the distribution also follows a bell-shaped curve. If we were to use the response efficiency score in a predicitive model, we would need to account for the high concentration of 100's. Disregarding the zeros, we can see a somewhat positive linear relationship between the two variables.

```
# joint plot of aid amount and response efficiency

g = sns.jointplot(data=df, x=np.sqrt(df['aid_amount_usd']),
                  y=np.sqrt(df['response_time_hours']), kind='hex');
g.fig.suptitle('Square Root of Aid Amount vs Square Root of Response Time for Disasters', y=
plt.xlabel('Square Root of Aid Amount (in USD)')
plt.ylabel('Square Root of Response Time (in hours)')
plt.tight_layout()
# Saving the plot as a JPEG file
```

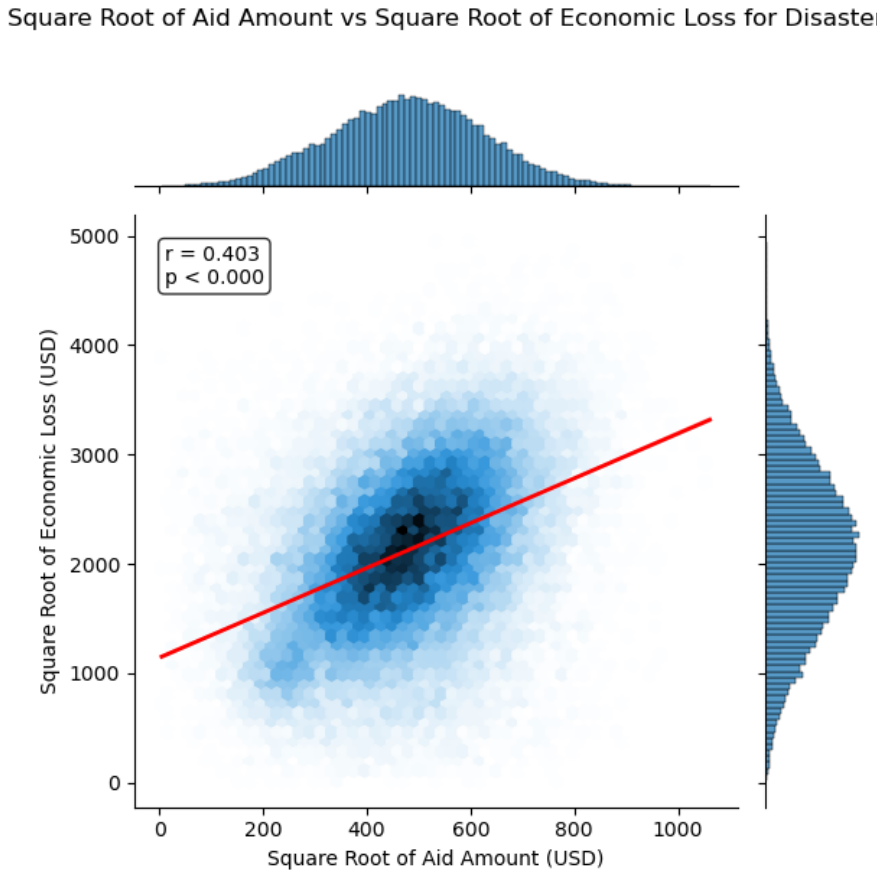
```
plt.savefig("graphs/eda/2_aid_response_time.jpg", bbox_inches='tight')
```



This is very similar to the graph above, reflected across the x-axis. The distribution of the response efficiency score is more dense around 0-30, with a very high concentration at 0. Above 0, the distribution also follows a bell-shaped curve. If we were to use the response efficiency score in a predicitive model, we would need to account for the high concentration of 0's. Disregarding the zeros, we can see a somewhat negative relationship between the two variables.

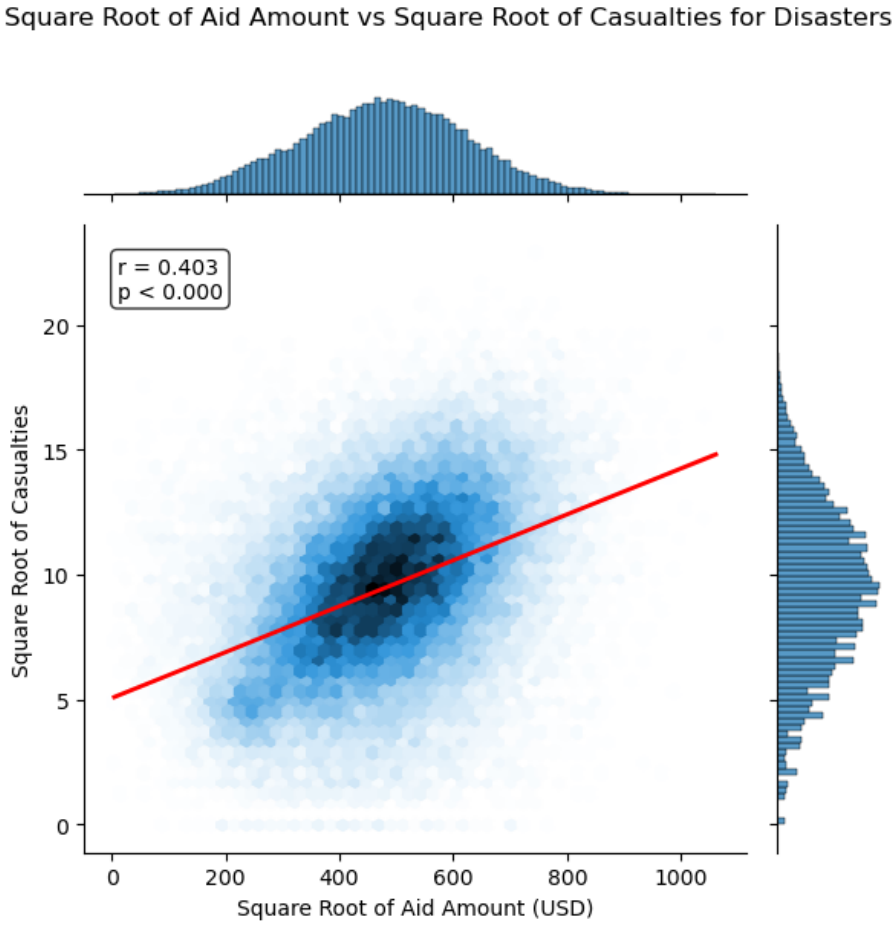
```
# Create the jointplot
g = sns.jointplot(data=df, x=np.sqrt(df['aid_amount_usd']),
                  y=np.sqrt(df['economic_loss_usd']), kind='hex')

# Add regression line
x = np.sqrt(df['aid_amount_usd'])
y = np.sqrt(df['economic_loss_usd'])
sns.regplot(x=x, y=y, scatter=False, ax=g.ax_joint, color='red', line_kws={'linewidth': 2})

# Calculate correlation coefficient
```

```
r, p_value = stats.pearsonr(x, y)

# Add correlation text to the plot
g.ax_joint.text(0.05, 0.95, f'r = {r:.3f}\np < {p_value:.3f}',
                transform=g.ax_joint.transAxes,
                verticalalignment='top',
                bbox=dict(boxstyle='round', facecolor='white', alpha=0.8),
                fontsize=10)

g.fig.suptitle('Square Root of Aid Amount vs Square Root of Economic Loss for Disasters', y=
g.set_axis_labels('Square Root of Aid Amount (USD)', 'Square Root of Economic Loss (USD)')
plt.tight_layout()

# Saving the plot as a JPEG file
plt.savefig("graphs/eda/3_aid_econ_loss.jpg", bbox_inches='tight')
```




```
# Create the jointplot
```

```
g = sns.jointplot(data=df, x=np.sqrt(df['aid_amount_usd']),
                  y=np.sqrt(df['casualties']), kind='hex')

# Add regression line
x = np.sqrt(df['aid_amount_usd'])
y = np.sqrt(df['casualties'])
sns.regplot(x=x, y=y, scatter=False, ax=g.ax_joint, color='red', line_kws={'linewidth': 2})

# Calculate correlation coefficient
r, p_value = stats.pearsonr(x, y)

# Add correlation text to the plot
g.ax_joint.text(0.05, 0.95, f'r = {r:.3f}\np < {p_value:.3f}',
                transform=g.ax_joint.transAxes,
                verticalalignment='top',
                bbox=dict(boxstyle='round', facecolor='white', alpha=0.8),
                fontsize=10)

g.fig.suptitle('Square Root of Aid Amount vs Square Root of Casualties for Disasters', y=1.0
g.set_axis_labels('Square Root of Aid Amount (USD)', 'Square Root of Casualties')  # Fixed
plt.tight_layout()

# Saving the plot as a JPEG file
plt.savefig("graphs/eda/4_aid_casualties.jpg", bbox_inches='tight')
```



The graphs above are very similar, both with a positive linear relationship and a correlation coefficient of 0.403. This would make aid amount an important variable in predicting economic losses and casualties. The relationship between the two variables could be positive because more damages would result in a higher aid amount. So aid amount is likely a response variable to the casualties and economic losses.

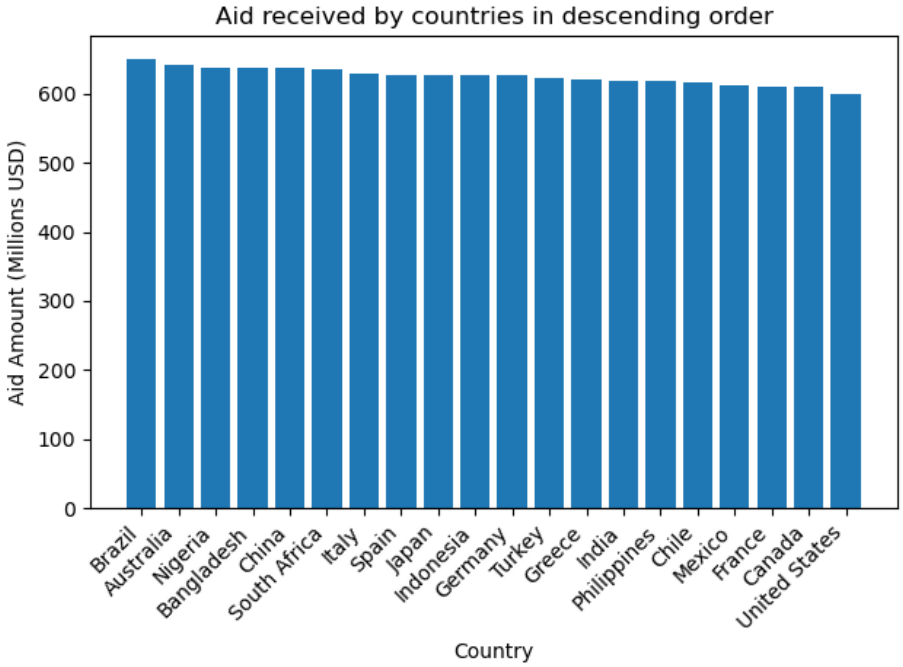
Economic Loss Per Country

```
# make a dataframe of the total aid amount received by countries from 2018-2024, grouped by
aid_amount = df[['country', 'aid_amount_usd']].groupby(['country']).sum().sort_values('aid_a
aid_amount
```

```
              aid_amount_usd
country
Brazil          6.504053e+08
```

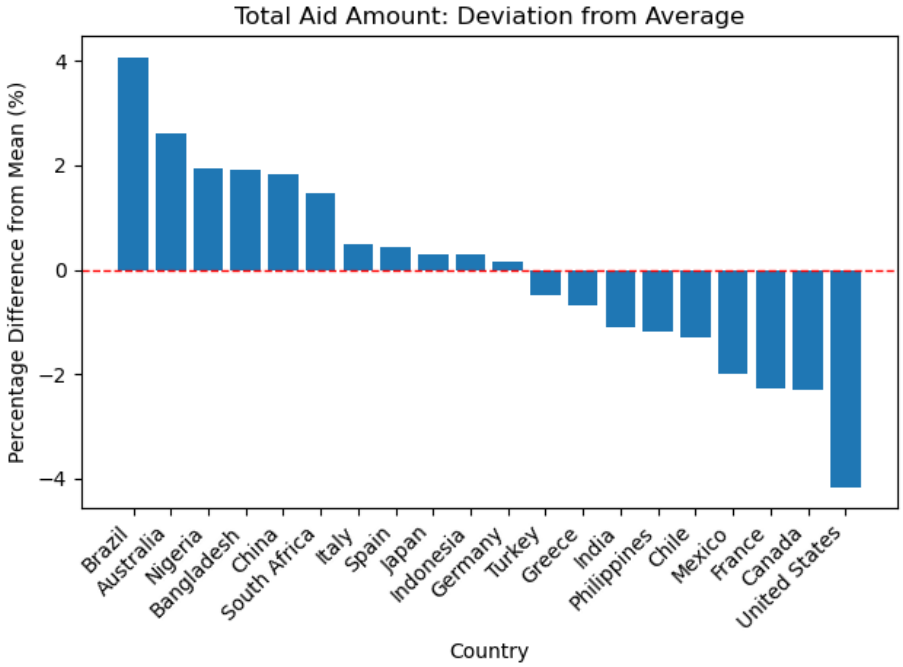
```
Australia        6.413243e+08
Nigeria          6.372058e+08
Bangladesh       6.369360e+08
China            6.363615e+08
South Africa     6.341191e+08
Italy            6.280263e+08
Spain            6.277419e+08
Japan            6.268700e+08
Indonesia        6.268396e+08
Germany          6.260436e+08
Turkey           6.220337e+08
Greece           6.206836e+08
India            6.181816e+08
Philippines      6.175013e+08
Chile            6.168857e+08
Mexico           6.126058e+08
France           6.106889e+08
Canada           6.105837e+08
United States    5.989786e+08
```

```
# plot the aid received by countries in descending order
plt.bar(aid_amount.index, aid_amount['aid_amount_usd'] / 1e6)
plt.xlabel('Country')
plt.ylabel('Aid Amount (Millions USD)')
plt.title('Aid received by countries in descending order')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
```



The bars in the graph above do not look very different from each other. When graphing the economic loss per country, we can also use the percentage difference from the mean since all the values don't look very different when graphed from 0 to the maximum value.

```
mean_aid = aid_amount['aid_amount_usd'].mean() # calculate the mean aid received by countri
pct_diff = ((aid_amount['aid_amount_usd'] - mean_aid) / mean_aid) * 100 # calculate the dif

plt.bar(aid_amount.index, pct_diff)
plt.axhline(y=0, color='red', linestyle='--', linewidth=1)
plt.xlabel('Country')
plt.ylabel('Percentage Difference from Mean (%)')
plt.title('Total Aid Amount: Deviation from Average')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()

# Saving the plot as a JPEG file
plt.savefig("graphs/eda/5_aid_country.jpg", bbox_inches='tight')
```



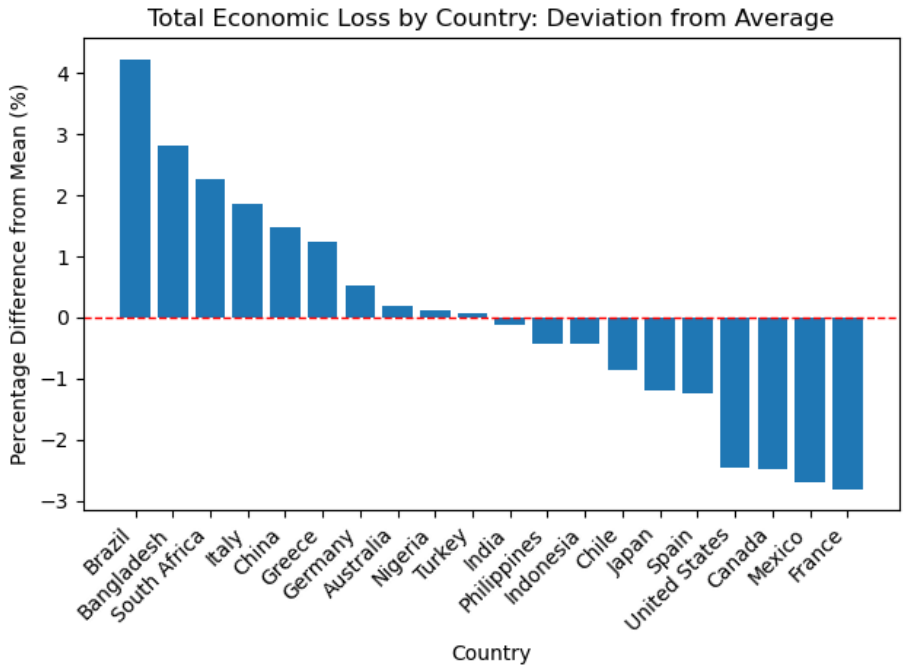
The percent difference from the mean ranges from around -4% to 4%, with Brazil being the country receiving the most aid and the United States being the country receiving the least amount of aid.

```
# make a dataframe of the total economic loss from countries from 2018-2024, grouped by cou

country_loss = df[['country', 'economic_loss_usd']].groupby(['country']).sum().sort_values('
country_loss
```

country	economic_loss_usd
Brazil	1.320587e+10
Bangladesh	1.302649e+10
South Africa	1.295759e+10
Italy	1.290836e+10
China	1.285750e+10
Greece	1.282855e+10
Germany	1.273712e+10
Australia	1.269432e+10
Nigeria	1.268575e+10
Turkey	1.268119e+10
India	1.265547e+10
Philippines	1.261609e+10
Indonesia	1.261571e+10

```
Chile                1.256236e+10
Japan                1.252109e+10
Spain                1.251307e+10
United States        1.235948e+10
Canada               1.235621e+10
Mexico               1.233096e+10
France               1.231648e+10
```

```
mean_loss = country_loss['economic_loss_usd'].mean() # calculate the mean economic loss ove
pct_diff = ((country_loss['economic_loss_usd'] - mean_loss) / mean_loss) * 100 # calculate

plt.bar(country_loss.index, pct_diff)
plt.axhline(y=0, color='red', linestyle='--', linewidth=1)
plt.xlabel('Country')
plt.ylabel('Percentage Difference from Mean (%)')
plt.title('Total Economic Loss by Country: Deviation from Average')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()

# Saving the plot as a JPEG file
plt.savefig("graphs/eda/6_econ_loss_country.jpg", bbox_inches='tight')
```



In this graph, the total economic losses per country ranges from -3% from the mean to 4% from the mean. The country with the highest economic loss is Brazil,

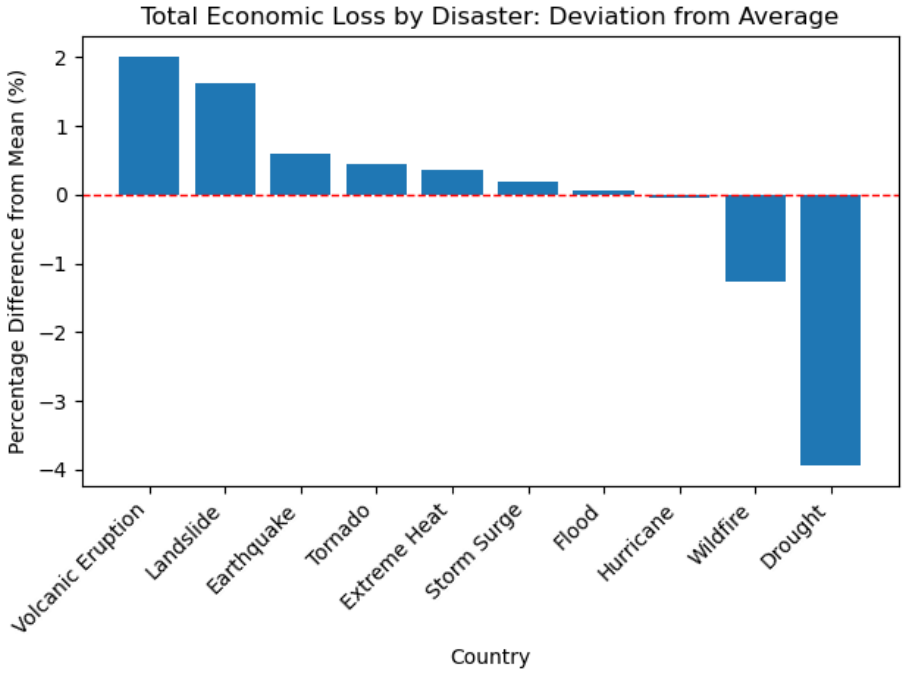
and the country with the lowest economic loss is France. While the economic loss may have similar trends to the amount of aid, this shows that it is not exactly the same because the country rankings in the economic loss graph and in the aid graph are not the same.

Economic Loss per Disaster Type

```
# Make a dataframe of total economic loss, grouped by disaster
disaster_loss = df[['disaster_type', 'economic_loss_usd']].groupby(['disaster_type']).sum()
disaster_loss
```

disaster_type	economic_loss_usd
Volcanic Eruption	2.585020e+10
Landslide	2.575344e+10
Earthquake	2.549166e+10
Tornado	2.545634e+10
Extreme Heat	2.543340e+10
Storm Surge	2.538840e+10
Flood	2.535799e+10
Hurricane	2.533048e+10
Wildfire	2.502223e+10
Drought	2.434554e+10

```
mean_loss = disaster_loss['economic_loss_usd'].mean() # calculate the mean economic loss ov
pct_diff = ((disaster_loss['economic_loss_usd'] - mean_loss) / mean_loss) * 100 # calculate

plt.bar(disaster_loss.index, pct_diff)
plt.axhline(y=0, color='red', linestyle='--', linewidth=1)
plt.xlabel('Country')
plt.ylabel('Percentage Difference from Mean (%)')
plt.title('Total Economic Loss by Disaster: Deviation from Average')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()

# Saving the plot as a JPEG file
plt.savefig("graphs/eda/7_econ_loss_disaster.jpg", bbox_inches='tight')
```



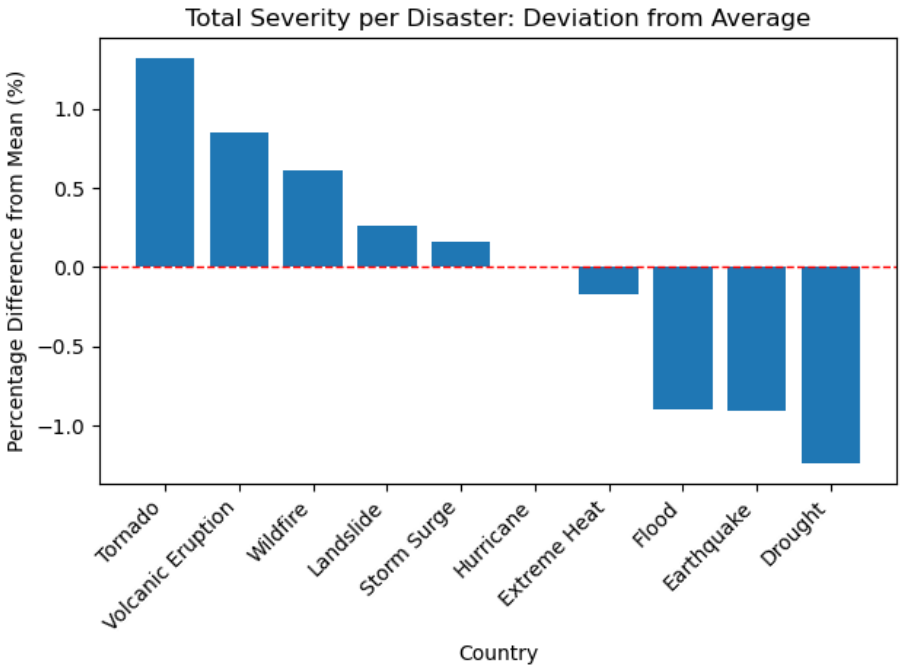
This graph shows the how different the total economic loss by disaster is from the mean. Drought is much lower compared to the others at -4% from the mean, and volcanic eruptions are around 2% from the mean.

```
# Make a dataframe of average severity, grouped by disaster
disaster_severity = df[['disaster_type', 'severity_index']].groupby(['disaster_type']).mean(
disaster_severity
```

disaster_type	severity_index
Tornado	5.081946
Volcanic Eruption	5.058509
Wildfire	5.046603
Landslide	5.028793
Storm Surge	5.023871
Hurricane	5.016074
Extreme Heat	5.007491
Flood	4.970891
Earthquake	4.970312
Drought	4.953795

```
mean_severity = disaster_severity['severity_index'].mean() # calculate the mean economic los
pct_diff = ((disaster_severity['severity_index'] - mean_severity) / mean_severity) * 100 # c

plt.bar(disaster_severity.index, pct_diff)
```

```
plt.axhline(y=0, color='red', linestyle='--', linewidth=1)
plt.xlabel('Country')
plt.ylabel('Percentage Difference from Mean (%)')
plt.title('Total Severity per Disaster: Deviation from Average')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()

# Saving the plot as a JPEG file
plt.savefig("graphs/eda/8_severity_disaster.jpg", bbox_inches='tight')
```



The mean severity per disaster shows less variation than the total economic loss, with tornados only being less than 2% above the mean and droughts being less than 2% below the mean. From the data, we can show that droughts have less impact and severity compared to the other disasters. For disasters like earthquakes, even though each event has an average of 1% below the mean, in total, they cause significant economic loss, both being higher than the average. Tornados, on the other hand, have a high severity per event, but the overall economic loss is lower than volcanic eruptions, landslides, and earthquakes.

Main Takeaways from EDA

- Aid amount is likely affected by the number of casualties and economic loss because their plots show a positive linear trend.

- Efficiency score has a disproportionate number of values at 100 and response time has a disproportionate number of values at 0.
- Tornadoes, volcanic eruptions, and wildfires have the highest total severity per disaster from 2018-2024.
- Volcanic eruptions and landslides have the highest total economic loss over all disasters from 2018-2024.

Based on the data, a good research question to ask is, "What would the amount of aid received be, given the country and amount of economic loss?" This is because the graphs show that the amount of aid is correlated with the amount of economic loss, and the aid amount varies per country. Response time and efficiency score seem like they may not be heavily correlated with the aid amount, so they might introduce some noise into the model.