## **Bivariate Analysis**

Numerical Categorical

"Eastern Europe"

"Southern Europe"

"Central Europe"

For numerical categorical exploration, we used a bar plot to plot the number of searches against the region of Europe where the search occurred. As Google gets search results from all over the world, trying to plot every country would be very time consuming and result in a hard-to-understand visualization. Since we are focusing on Europe for this exploration, we first need to filter our dataset for only search terms that came from European countries.

```
filter our dataset for only search terms that came from European countries.
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
# Data preprocessing
trends = pd.read_csv("../trends.csv")
trends = trends.drop(columns = ["country_code", "region_code", "refresh_date"],
trends["week"] = trends["week"].str[:7]
trends = trends.rename(columns = {"week":"Year-Month"})
trends.columns = ["Score", "Year-Month", "Rank", "Country", "Sub-Region", "Term"]
trends = trends[["Term", "Rank", "Country", "Sub-Region", "Year-Month", "Score"]]
Once preprocessing is finished, we can filter the dataset for European countries.
countries = ["Sweden", "Turkey", "Romania", "Czech Republic", "Norway", \
"Italy", "Austria", "Netherlands", "Poland", "Switzerland", \
"France", "Finland", "Ukraine", "United Kingdom", "Denmark", "Germany", \
"Portugal", "Belgium"]
europeTrends = trends[trends["Country"].isin(countries)]
After we filter for European countries, we can then assign region classifiers to each entry.
nordicCountries = ["Sweden", "Norway", "Denmark", "Finland"]
eastEuroCountries = ["Romania", "Poland", "Ukraine"]
southEuroCountries = ["Turkey", "Italy", "Portugal"]
centralEuroCountries = ["Czech Republic", "Germany", "Austria", "Switzerland"]
westEuroCountries = ["Netherlands", "France", "United Kingdom", "Belgium"]
# Assign region classifiers to countries
europeTrends = europeTrends.assign(Region = europeTrends["Country"])
europeTrends.loc[europeTrends["Region"].isin(nordicCountries), "Region"] = "Nordic
Countries"
europeTrends.loc[europeTrends["Region"].isin(eastEuroCountries), "Region"] =
```

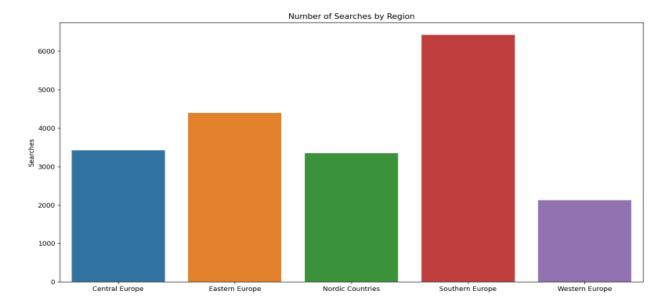
europeTrends.loc[europeTrends["Region"].isin(southEuroCountries), "Region"] =

europeTrends.loc[europeTrends["Region"].isin(centralEuroCountries), "Region"] =

```
europeTrends.loc[europeTrends["Region"].isin(westEuroCountries), "Region"] =
"Western Europe"
```

Now we can make our bar plot, using the count() function to aggregate searches by region.

```
# Plot number of searches by region
europeTrendsCount = europeTrends.groupby("Region", as_index=False).count()
sns.barplot(x="Region", y="Term", data=europeTrendsCount[["Region", "Term"]])
plt.xlabel("")
plt.ylabel("Searches")
plt.title("Number of Searches by Region")
plt.show()
```

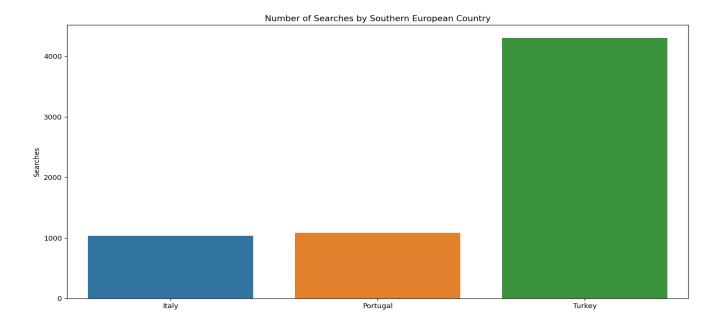


From this chart, we can see that Southern Europe has more searches than any other region. For further analysis, we can isolate all of the searches from Southern Europe.

```
# Create dataframe for Southern Europe
southEuroTrends = trends[trends["Country"].isin(southEuroCountries)]
```

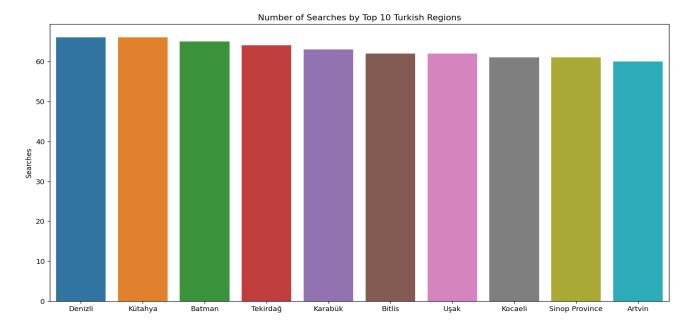
Now we can plot the number of searches against countries in Southern Europe.

```
# Plot number of searches in Southern Europe by country
southEuroTrendsCount = southEuroTrends.groupby("Country", as_index=False).count()
sns.barplot(x="Country", y="Term", data=southEuroTrendsCount)
plt.xlabel("")
plt.ylabel("Searches")
plt.title("Number of Searches by Southern European Country")
plt.show()
```



From this chart, we can see that Turkey has by far more searches than either Italy or Portugal. If we compare this with the previous chart, we can also see that Turkey has more searches than most other European regions. Since there are 81 subregions associated with searches from Turkey, it would be difficult to interpret a chart that plots all of them. To make it easier to visualize, we plot only the top 10 regions with the most searches.

```
# Plot number of searches in Turkey by top 10 regions
turkeyTrendsCount = turkeyTrends.groupby("Sub-Region", as_index=False).count()
sns.barplot(x="Sub-Region", y="Term", data=turkeyTrendsCount.nlargest(15, "Term"))
plt.xlabel("")
plt.ylabel("Searches")
plt.title("Number of Searches by Turkish Region")
plt.show()
```



From this chart, we can see that there is not a large difference in the number of trending searches from each Turkish province. For the final part of this exploration, we plot the number of searches from the bottom 10 regions.

```
# Plot number of searches in Turkey by bottom 10 regions
sns.barplot(x="Sub-Region", y="Term", data=turkeyTrendsCount.nsmallest(10, "Term"))
plt.xlabel("")
plt.ylabel("Searches")
plt.title("Number of Searches by Bottom 10 Turkish Regions")
plt.show()
```

