

Data Visualization



Data Science Workflow



Components of EDA

Summarization

Components of EDA

- Descriptive statistics

	count	mean	s
Salary (USD)	14,838.0	149,874.7	69,009

- Frequency tables

Employment Type	FT	PT	CT
count	14785	27	26

- Distribution metrics

Components of EDA

Summarization

What we will learn:

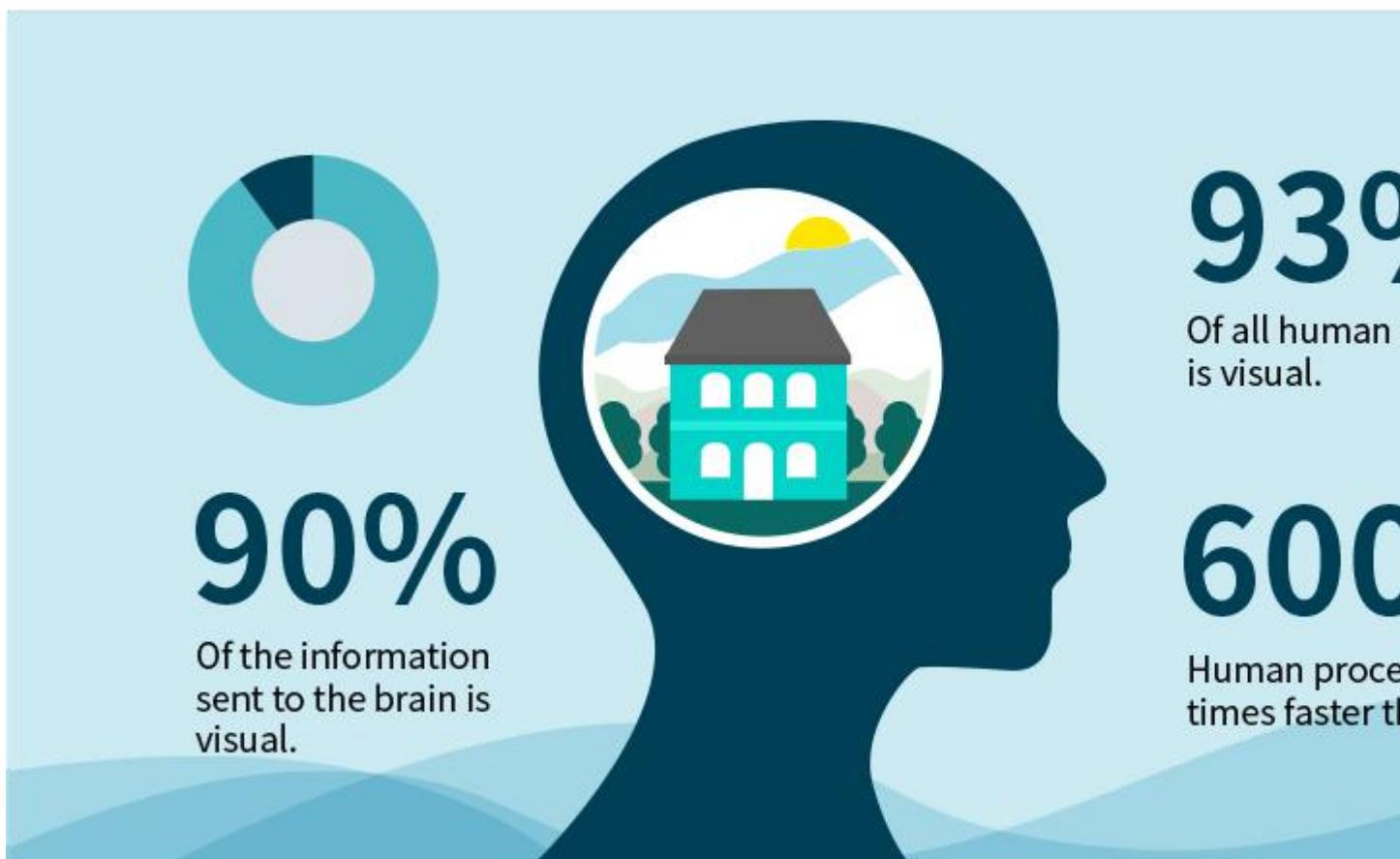
- Why data visualization is important?
- Univariate visualizations
- Multivariate visualizations
- Visualization using Pandas (Matplotlib + Seaborn)

Why is important?

Making informative visualizations (sometimes called *plots*) is one of the most important tasks in data science.

- **Understanding the Data:**

Visualization provides a way to explore and understand the underlying patterns, trends, and relationships within the data. **Humans are highly visual creatures**, and presenting data visually allows us to quickly grasp complex information that might be difficult to discern from raw data alone.



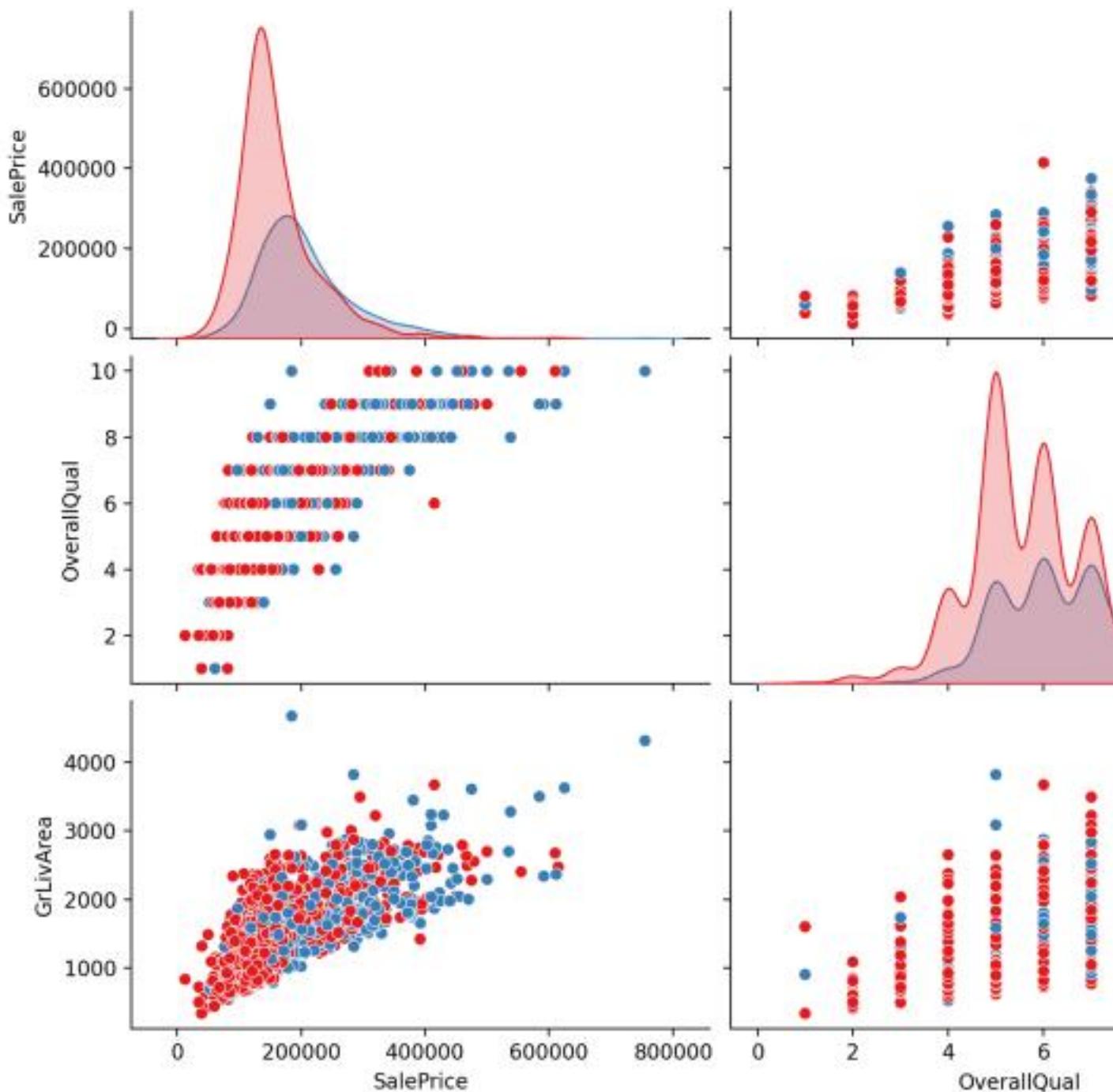
- **Communication:**

Visualization facilitates **communication of findings and insights to stakeholders** who may not have expertise in data analysis or statistics. Well-designed visualizations can effectively convey complex information in a clear and intuitive manner, enabling better decision-making.



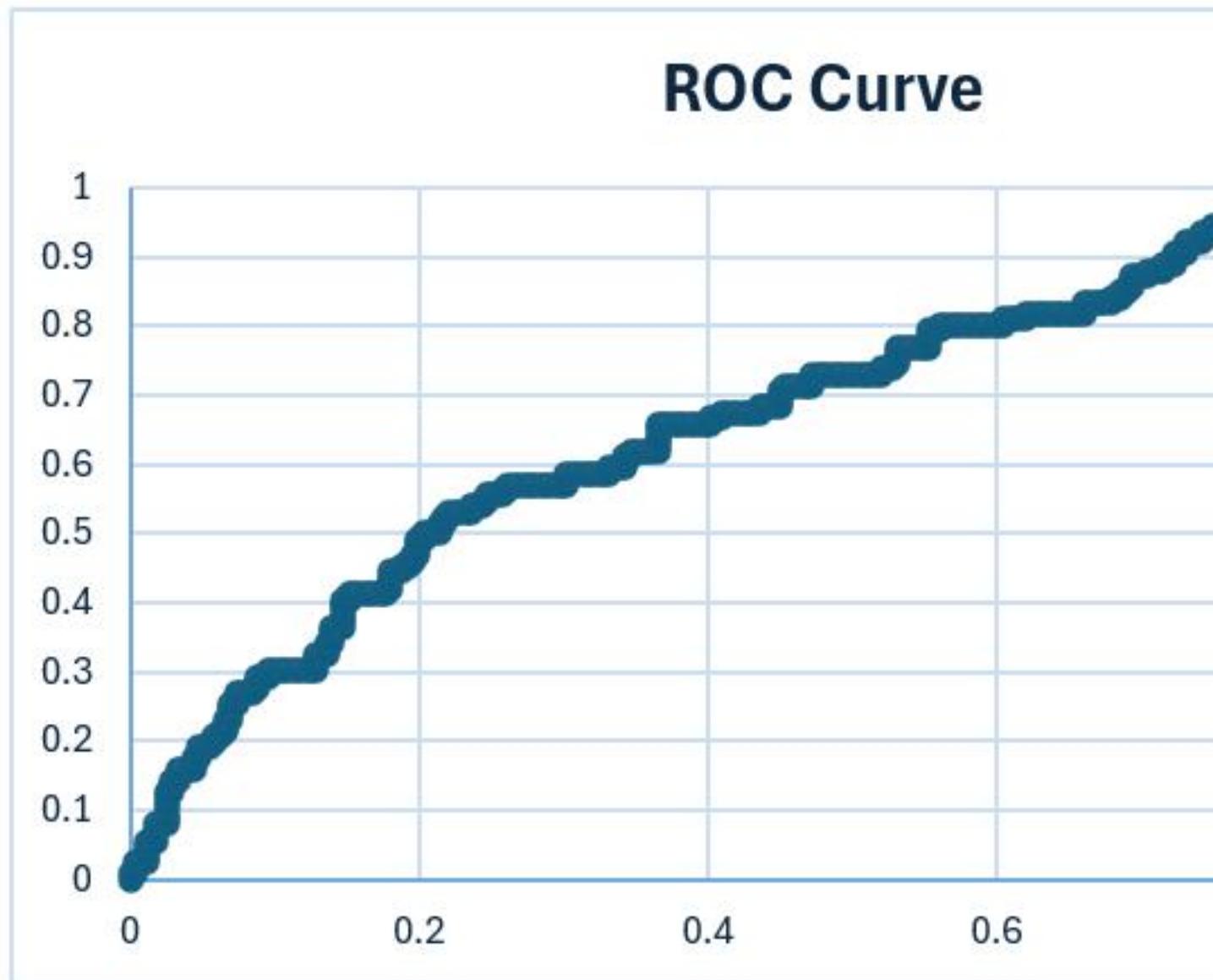
- **Identifying Patterns and Anomalies:**

Visualization helps in identifying **patterns, trends, outliers, and anomalies** in the data that may not be apparent from summary statistics or tabular representations. This enables data scientists to gain deeper insights into the data and make informed decisions.



- **Model Evaluation:**

Visualizations are crucial for evaluating the performance of machine learning models. Plots such as **ROC curves, confusion matrices, and calibration plots** provide valuable insights into the performance of classification and regression models, helping data scientists fine-tune their models for better accuracy and generalization.



Types of visualization (plots)

- **Univariate plots:**
 - **Definition:** Univariate plots display the characteristics of a single variable.
 - **Purpose:** Useful for initial exploration of data, identifying outliers, understanding variable distributions, and assessing data quality.
 - **Examples:** Histograms, box plots, count plots, pie charts, and density plots

Types of visualization (plots) cont.

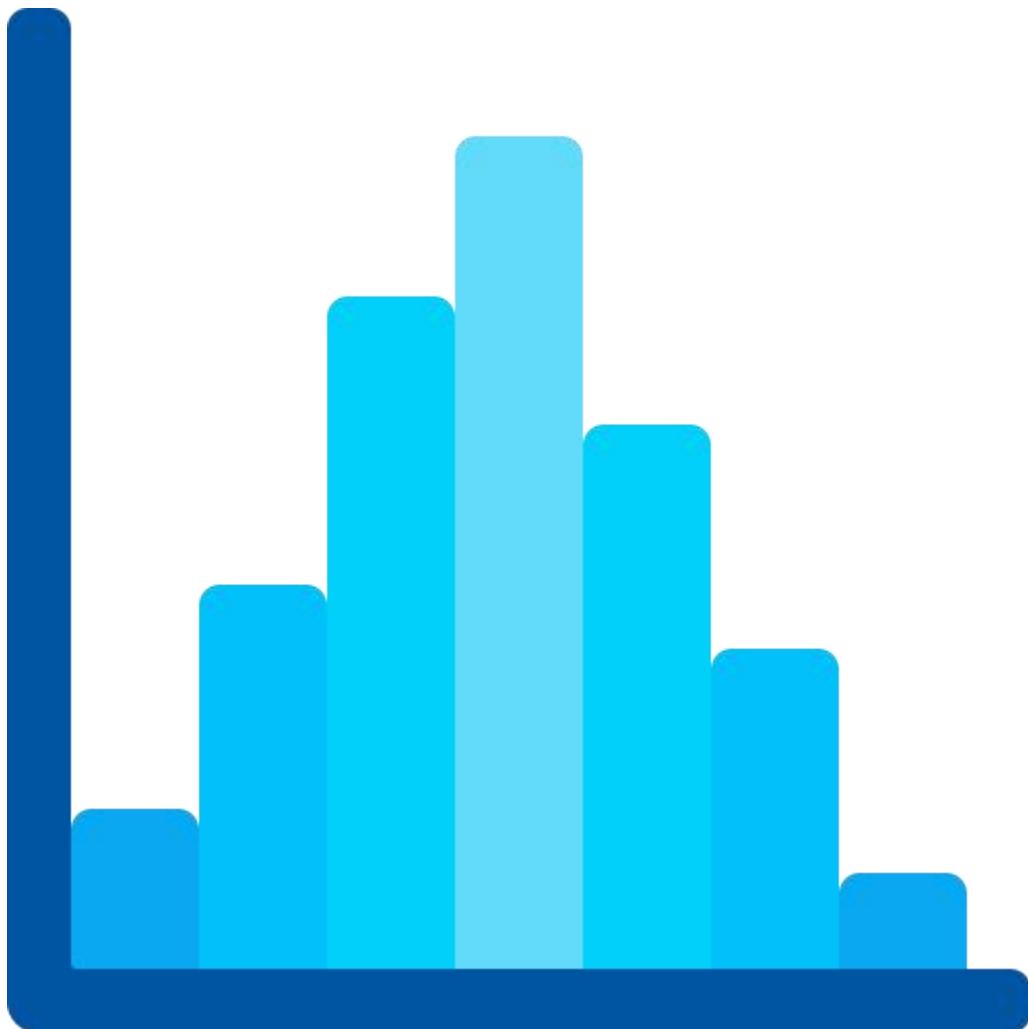
2. Multivariate plot:

- **Definition:** Multivariate plots display the relationships between multiple variables simultaneously.
- **Purpose:** They are used to explore interactions, patterns, correlations, and dependencies between two or more variables.
- **Examples:** Bar chart, line plot, Scatter plots, heatmap, pair plots

Univariate plots:

When we're working with a single variable, understanding its distribution is essential. Univariate data visualizations help us see the distribution, trends, and central tendencies, giving a solid foundation for deeper analysis. One of the most common and useful tools for this purpose is the **histogram**.

Histogram



A histogram is a type of bar chart that represents the distribution of numerical data by grouping values into "bins" or intervals along the x-axis and plotting their frequencies on the y-axis. Unlike bar charts, which display categorical data, histograms focus on numerical data, making them well-suited to show how values are spread out over their range.

Steps to create a histogram

Creating a histogram involves these steps:

- Determine the range of the data
- Divide the range into equal width of groups, called bins
- Calculate the height of the bars by the frequency of data values in each bin.

Histogram is useful for:

- **Shape of Distribution:** A histogram reveals where data points are concentrated and whether the distribution is skewed or symmetric.
 - center
 - spread
 - shape
 - normality
- **Outliers:** It helps to identify any unusual values that fall far outside the majority of the values.

Data we will use:

```
In [1]:  
import matplotlib.pyplot as plt  
import seaborn as sns  
import pandas as pd  
  
df = pd.read_csv("data/WHR.csv")  
df
```

Out[1]:

	year	country	region	happiness_score	gdp_per_capita	social_support	healthy_life_expectancy
0	2015-01-01	Switzerland	Western Europe	7.587	1.39651	1.34951	0.94143
1	2015-01-01	Iceland	Western Europe	7.561	1.30232	1.40223	0.94784
2	2015-01-01	Denmark	Western Europe	7.527	1.32548	1.36058	0.87464
3	2015-01-01	Norway	Western Europe	7.522	1.45900	1.33095	0.88521
4	2015-	Canada	North	7.427	1.32629	1.32261	0.90563

	year	country	region	happiness_score	gdp_per_capita	social_support	healthy_life_expectancy
	01-01		America and ANZ				
...
1362	2023-01-01	Congo (Kinshasa)	Sub-Saharan Africa	3.207	0.53100	0.78400	0.10500
1363	2023-01-01	Zimbabwe	Sub-Saharan Africa	3.204	0.75800	0.88100	0.06900
1364	2023-01-01	Sierra Leone	Sub-Saharan Africa	3.138	0.67000	0.54000	0.09200
1365	2023-01-01	Lebanon	Middle East and North Africa	2.392	1.41700	0.47600	0.39800
1366	2023-01-01	Afghanistan	South Asia	1.859	0.64500	0.00000	0.08700

1367 rows × 10 columns

Plot histogram for PERCEPTIONS_OF_CORRUPTION

In [2]:

```
df['perceptions_of_corruption'].hist(bins=20);
```

Interpret the distribution in terms of:

- Center
- Spread
- Shape
- Normality
- Outliers/Extream values

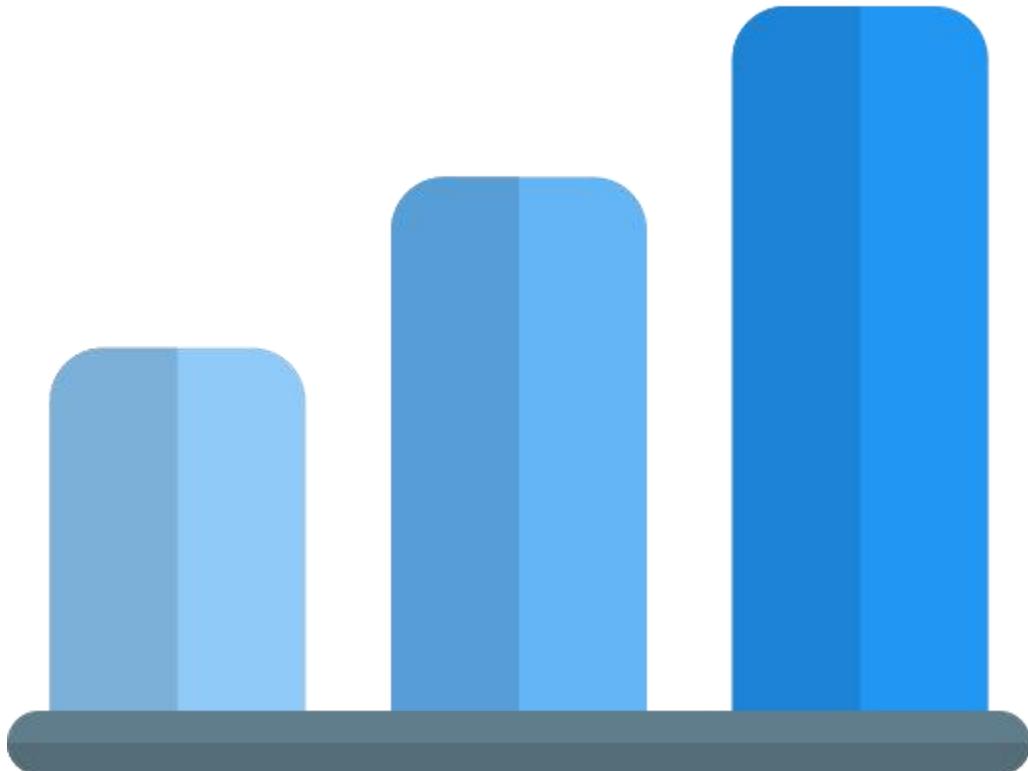
Create histogram using Seaborn

In [3]:

```
import matplotlib.pyplot as plt
import seaborn as sns
```

```
# Plot using Seaborn
sns.histplot(df['perceptions_of_corruption'], bins=20, kde=True)
plt.title('Perceptions of Corruption')
plt.xlabel('Score')
plt.ylabel('Frequency')
plt.show()
```

Count plot



A count plot is used for visualizing categorical data, making it especially useful in scenarios where we need to compare quantities across categories. Unlike histograms, which display the frequency distribution of continuous data, bar charts showcase the counts or values associated with discrete categories.

Count plot is usefule for:

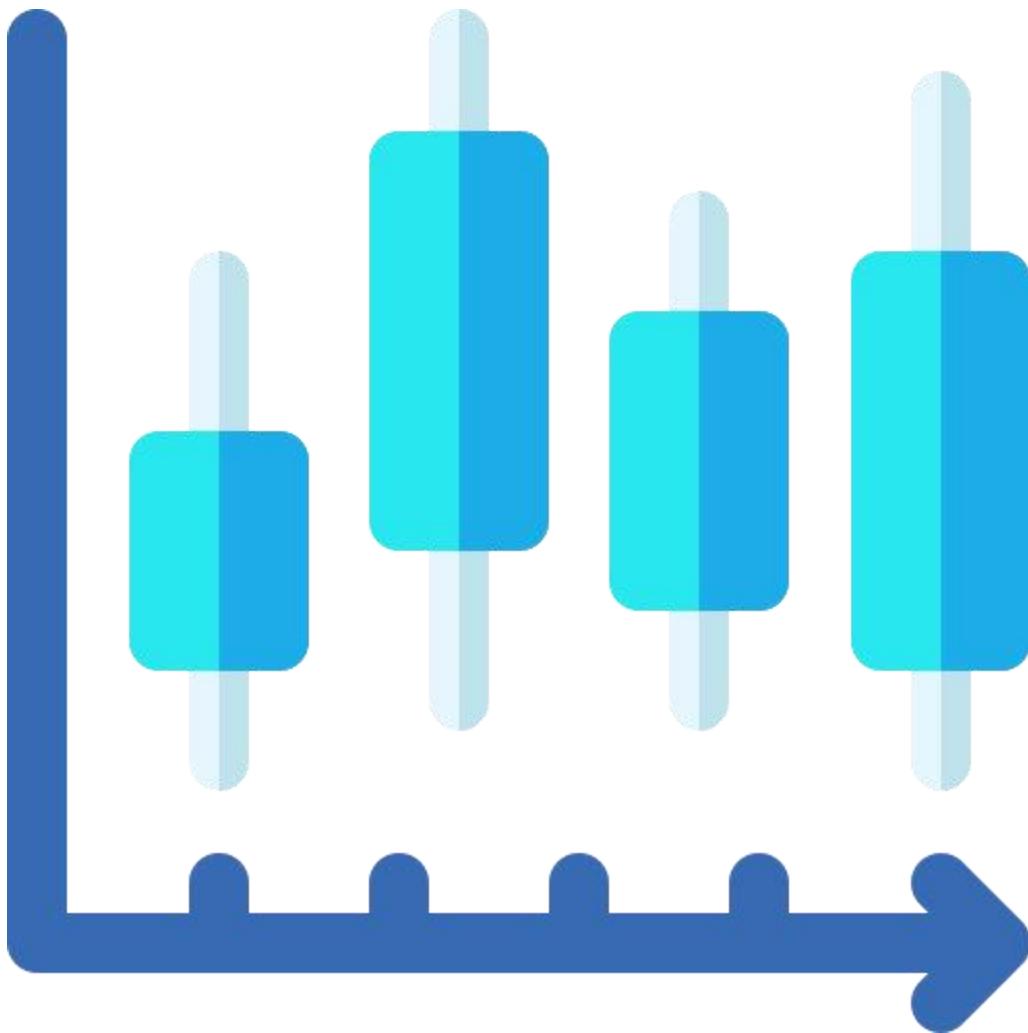
- showing the frequency counts or the percentages of values for the different levels of a categorical variable.
- It is like the histogram, shows the shape of values, or distribution, but on categorical variable.
- The bars show the levels of the variable; the height of the bars show the counts/ratios of responses for that level.

Create Count plot using Seaborn

In [4]:

```
sns.countplot(df, y="region", orient="h");
plt.title('Number of scores per region')
plt.xlabel('Frequency')
plt.ylabel('Region')
plt.show()
```

Box plot



The box plot, also known as a box-and-whisker plot, is a powerful tool for visualizing the distribution, spread, and symmetry of data, particularly highlighting where the majority of data values lie and identifying any potential outliers. This plot type is especially helpful for comparing distributions across multiple categories or datasets.

Box plot is useful for:

- A box plot shows the distribution of data for a continuous variable.
- Box plots help you see the **center** and **spread** of data. You can also use them as a visual tool to check for **normality** or to identify points that may be **outliers**.

Properties of the Box Plot:

- The center line in the box shows the median for the data. Half of the data is above this value, and half is below.
- If the data are symmetrical, the median will be in the center of the box. If the data are skewed, the median will be closer to the top or to the bottom of the box.

- The bottom and top of the box show the 25th and 75th quantiles, or percentiles. These two quantiles are also called quartiles because each cuts off a quarter (25%) of the data. The length of the box is the difference between these two percentiles and is called the interquartile range (IQR).
- The lines that extend from the box are called whiskers. The whiskers represent the expected variation of the data. The whiskers extend 1.5 times the IQR from the top and bottom of the box.
- If there are values that fall above or below the end of the whiskers, they are plotted as dots. These points are often called outliers.

Upper Whisker

Length of the box is the difference between the 75th and 25th percentiles and is called the IQR for interquartile range

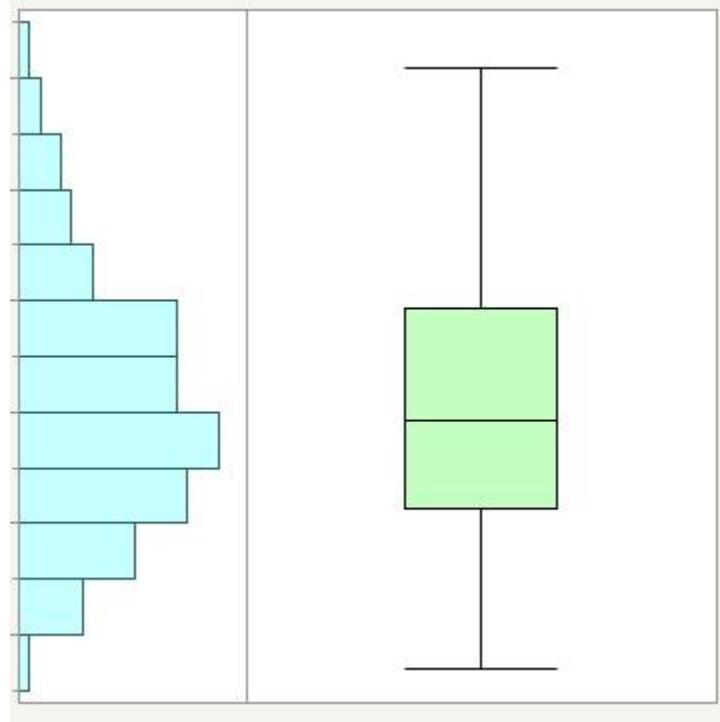
75th per

50th per

25th per

Lower Whisker

Box vs. Histogram



- The box plot helps you see skewness, because the line for the median will not be near the center of the box if the data is skewed.
- The box plot helps identify the 25th and 75th percentiles better than the histogram
- The box plot helps identify outliers better than histogram
- while the histogram helps you see the overall shape of your data better than the box plot.

Create Box plot using Seaborn

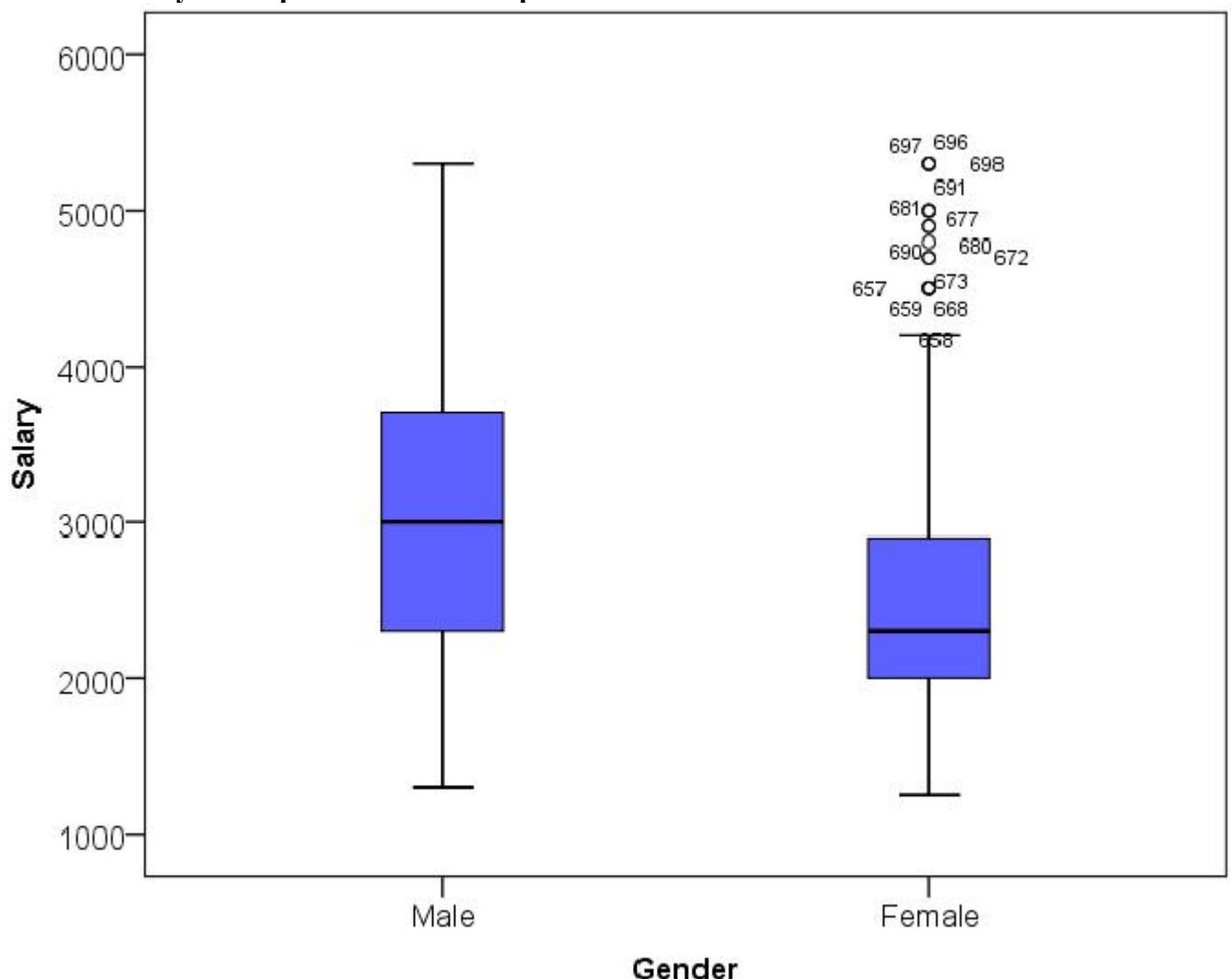
In [5]:

```
sns.boxplot(data=df, x="happiness_score", orient='h');
```

In [6]:

```
# Create box plot for each reagion
sns.boxplot(data=df, x="happiness_score", orient='h', y='region');
```

Discussion: how you interpret the below box plot?



Multivariate plots:

Multivariate plots are visualizations that help us explore and understand relationships among multiple variables in a dataset. They capture complex relationships involving two or more variables. These plots are essential for uncovering patterns, trends, and potential correlations that may not be obvious when analyzing variables individually.

Line plot



A line plot is used to show changes or trends in data over time or ordered categories. It connects data points with a line, making it easy to observe increases, decreases, and patterns across time intervals or other sequentially ordered values. Line plots are especially effective when comparing trends across multiple variables on the same plot.

Line plot is useful for:

- Visualizing **trends over time** for one or multiple variables simultaneously.

Properties of a line plot

- Line plots reveal **upward** or **downward trends** over intervals.
- Most commonly used with **time-series** data (e.g., sales by month)
- **Multiple line plots** can be displayed together to **compare trends** of different groups or categories.

Create Line plot using Seaborn

In [7]:

```
# Filter out Palestine records
palestine = df[df['country'] == 'State of Palestine']

# Increase the width of the figure
plt.figure(figsize=(10, 6))

sns.lineplot(data = palestine, x='year', y='happiness_score')

plt.title("Palestine happiness score over time")
plt.xlabel("Year")
plt.ylabel("Happiness Score")
plt.show()
```

In [8]:

```
# How the happiness score is changing for MENA region?
mena = df[df['region'] == 'Middle East and North Africa']
mena = mena.groupby('year')['happiness_score'].mean().to_frame()

plt.figure(figsize=(15, 6))

sns.lineplot(data = mena, x='year', y='happiness_score')

plt.title("Mena happiness score over time")
plt.xlabel("Year")
plt.ylabel("Happiness Score")
plt.show()
```

In [9]:

```
#Show how happiness score is changing per region

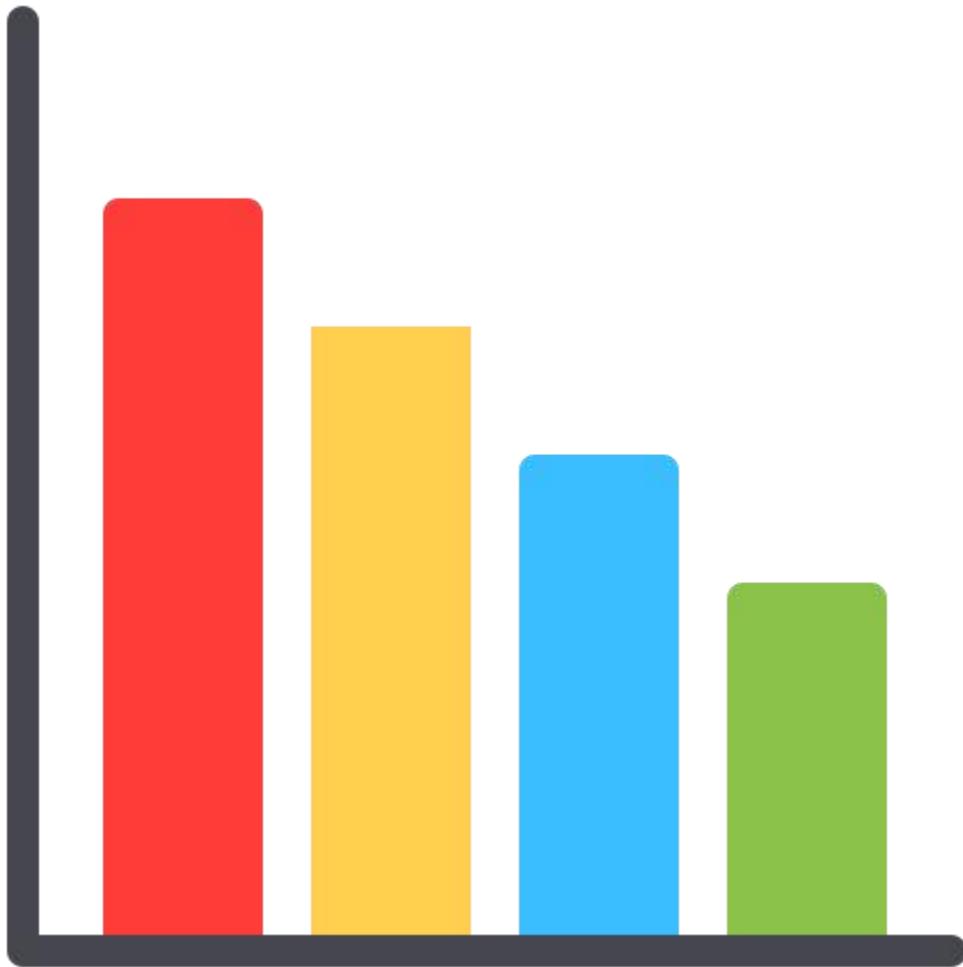
# Increase the width of the figure
plt.figure(figsize=(14, 6))

l = sns.lineplot(data = df, x='year', y='happiness_score', hue='region',
errorbar=None)

plt.title("happiness score over time by region")
plt.xlabel("Year")
plt.ylabel("Happiness Score")
plt.legend(loc='upper left')
sns.move_legend(l, "upper left", title='Region', bbox_to_anchor=(1, 1))
```

```
plt.show()
```

Bar plot



The **bar plot** is similar to **count plot** where both used for visualizing categorical data, but it is slightly different where it shows **aggregations** like SUM or MEAN of a **particular variable** for each **category**.

That's why **bar plot** is considered a multivariate plot while **count plot** is a univariate plot showing information about only one variable. Examples of **bar plot**:

- Mean happiness score for each region.
- Sum of sales for each branch
- Mean score for each section

Create Bar plot using Seaborn

In [10]:

```
#Show how happiness scores is different between regions
sns.barplot(y="region", x="happiness_score", data=df, orient = 'h')
plt.title("Mean Happiness Score by Region")
plt.xlabel("Mean Happiness Score")
plt.ylabel("Region")
plt.show()
```

In the above example the Y-Axis shows each region as a category and the X-Axis displays the average happiness score for each region, allowing an easy comparison between them.

Error bars in bar plots represent the variability or uncertainty in data and can give viewers a sense of the data's spread or reliability around each bar's central value. In Seaborn's bar plots, error bars are often included by default, showing either the standard deviation or the confidence interval, depending on the parameters specified.

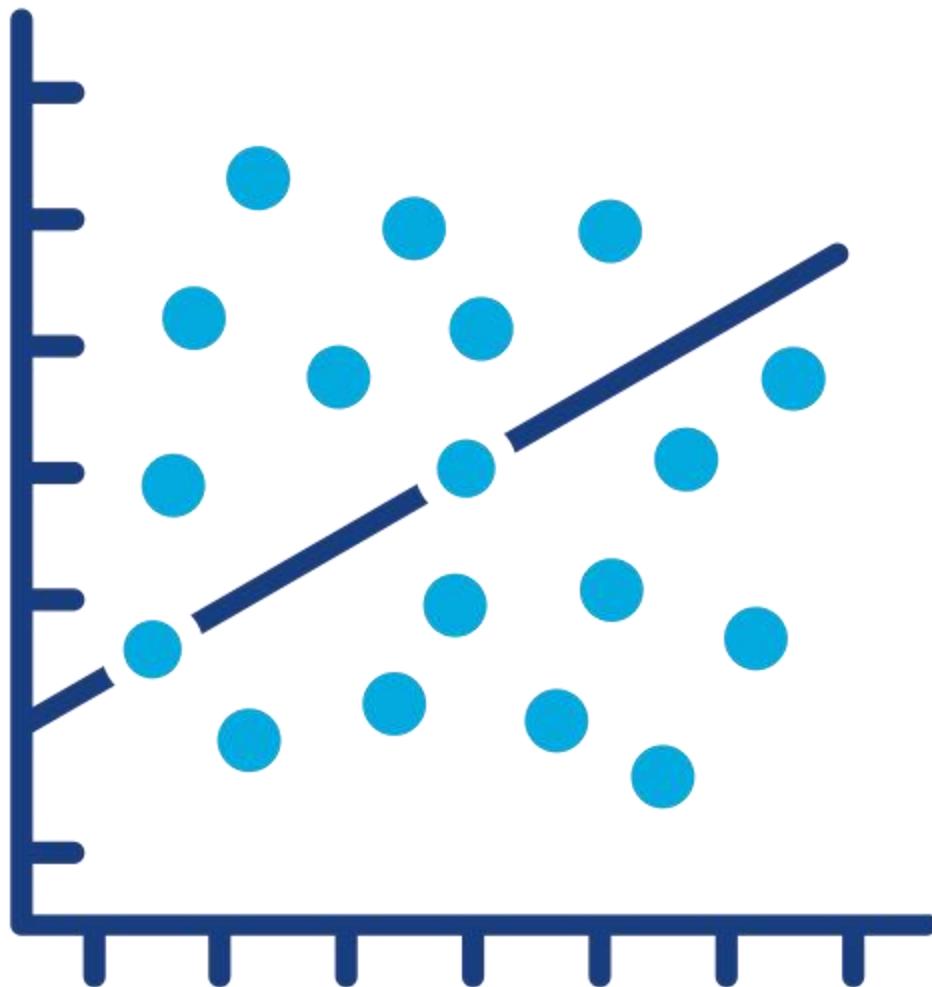
For comparing multiple variables, Seaborn's **hue** parameter allows you to stack or group bars by an additional categorical variable.

Let's compare the mean HAPPINESS SCORE between REGIONS by adding another variable, the YEAR

In [11]:

```
#Show how happiness scores is different between regions
plt.figure(figsize=(10, 15))
sns.barplot(y="region", x="happiness_score", data=df, orient = 'h',
hue='year')
plt.title("Mean Happiness Score by Region")
plt.xlabel("Mean Happiness Score")
plt.ylabel("Region")
plt.show()
```

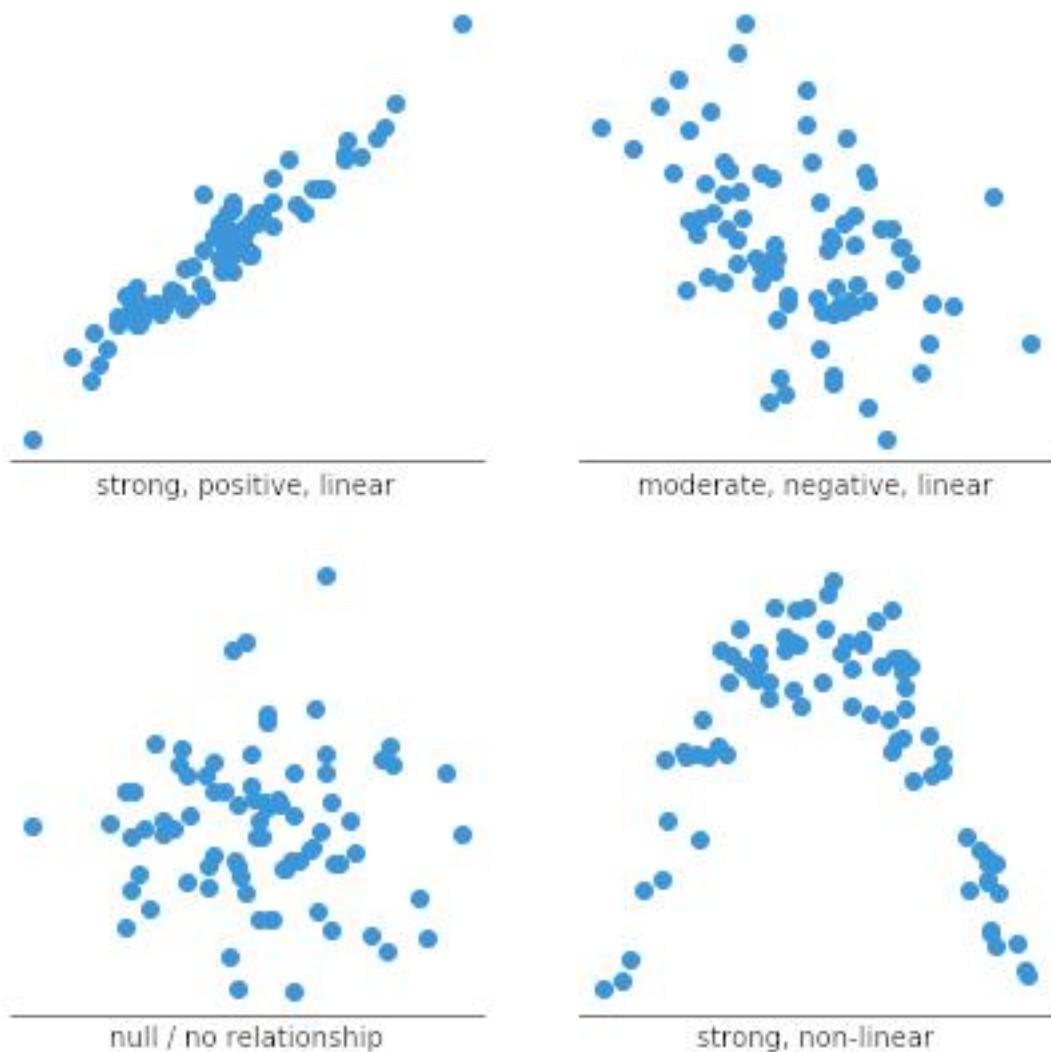
Scatter plot



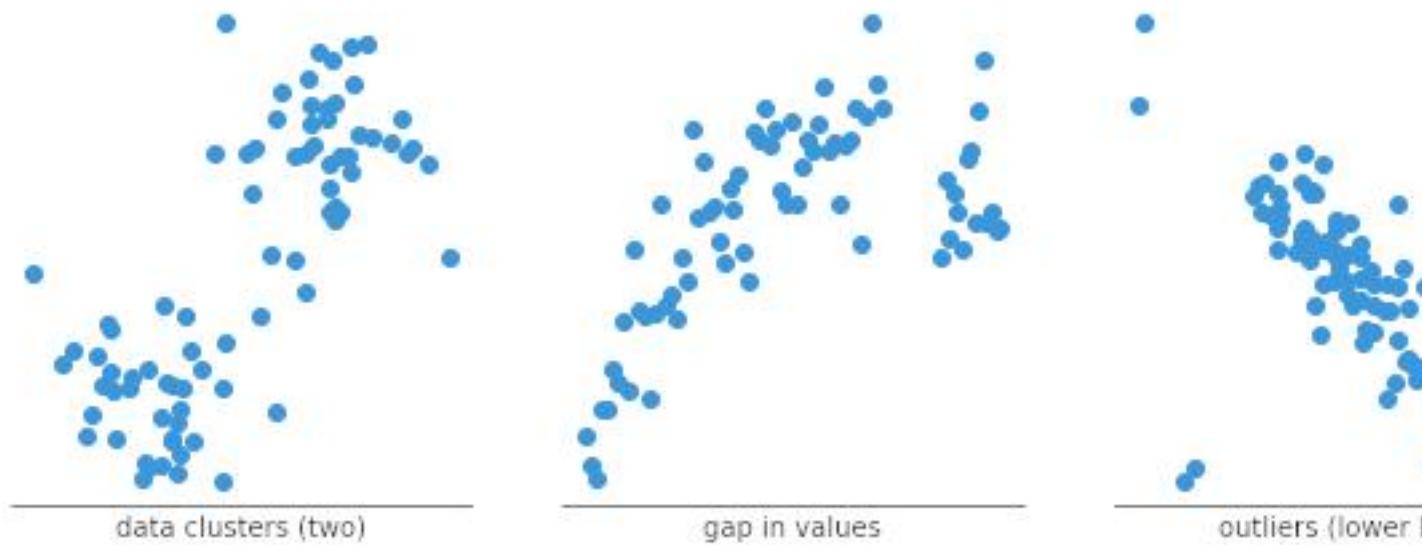
A **scatter plot** is a powerful visualization for exploring the **relationship between two continuous variables**. Each **point** on a scatter plot **represents an individual observation**, with its position determined by values on the x- and y-axes. A scatter plot (aka scatter chart, scatter graph) uses dots to represent values for two different numeric variables.

Scatter plot is useful for:

- Shows the **relationship between two numerical variables** and shows the **pattern of the relationship**:
 - **Direction** either **positive or negative**
 - **Strong, moderate, or weak**
 - **linear or nonlinear**.

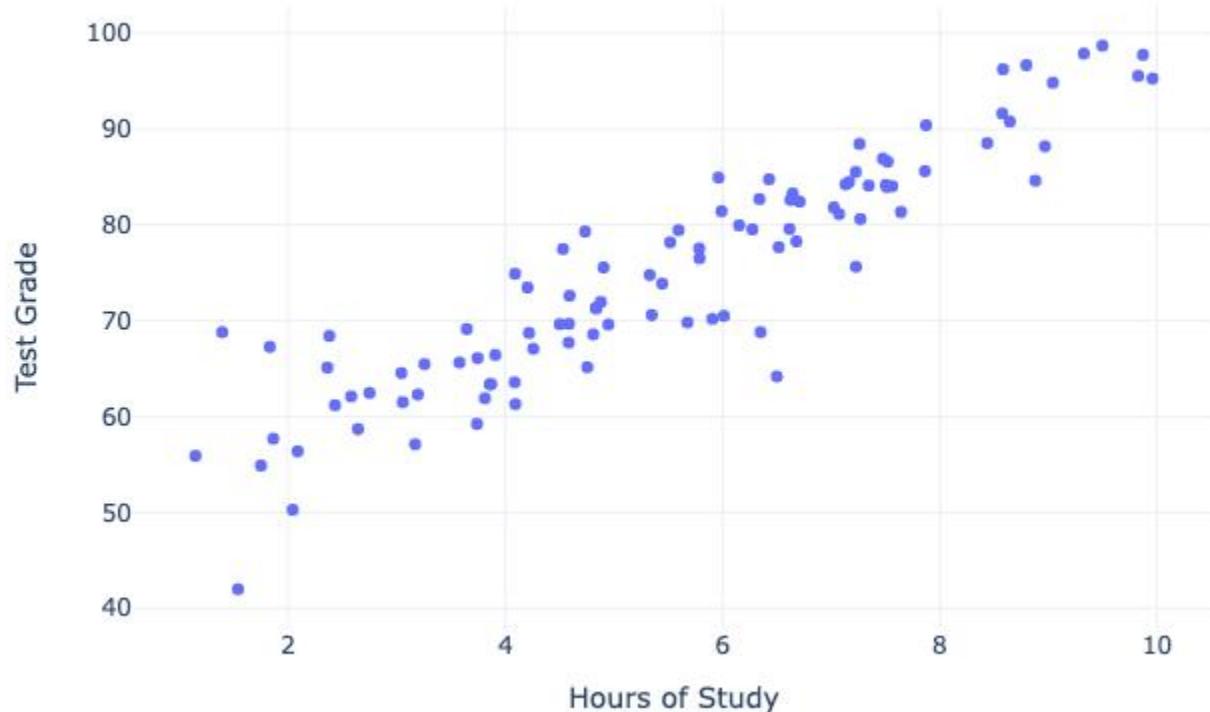


- Identify patterns in the data
 - **Detect outlier** as unusual points that don't follow the general distribution are easily spotted on a scatter plot.
 - **Identifying clusters:** It can shows gaps in the data such that data points can be divided into groups based on how closely sets of points cluster together.



- **Prediction:** If we were given a particular value in the X-Axis, what a good prediction would be for the Y-Axis value

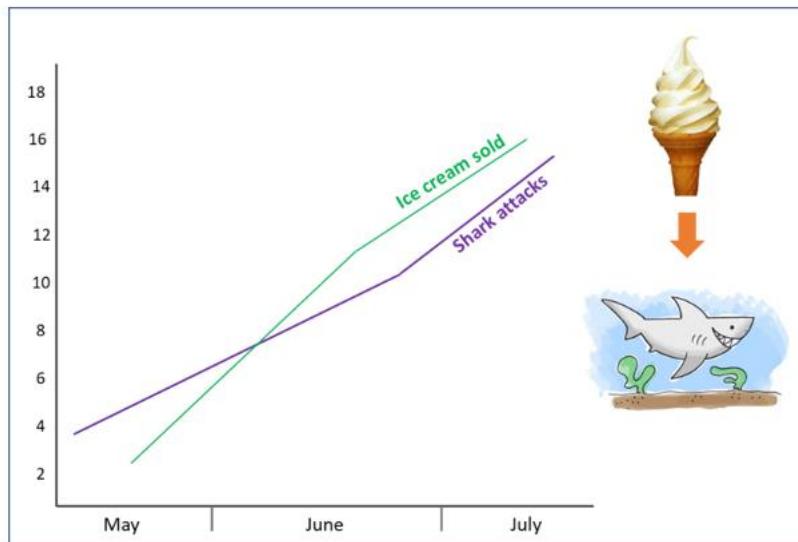
Relationship between Hours of Study and Test Grades



Wrong correlation conclusions:

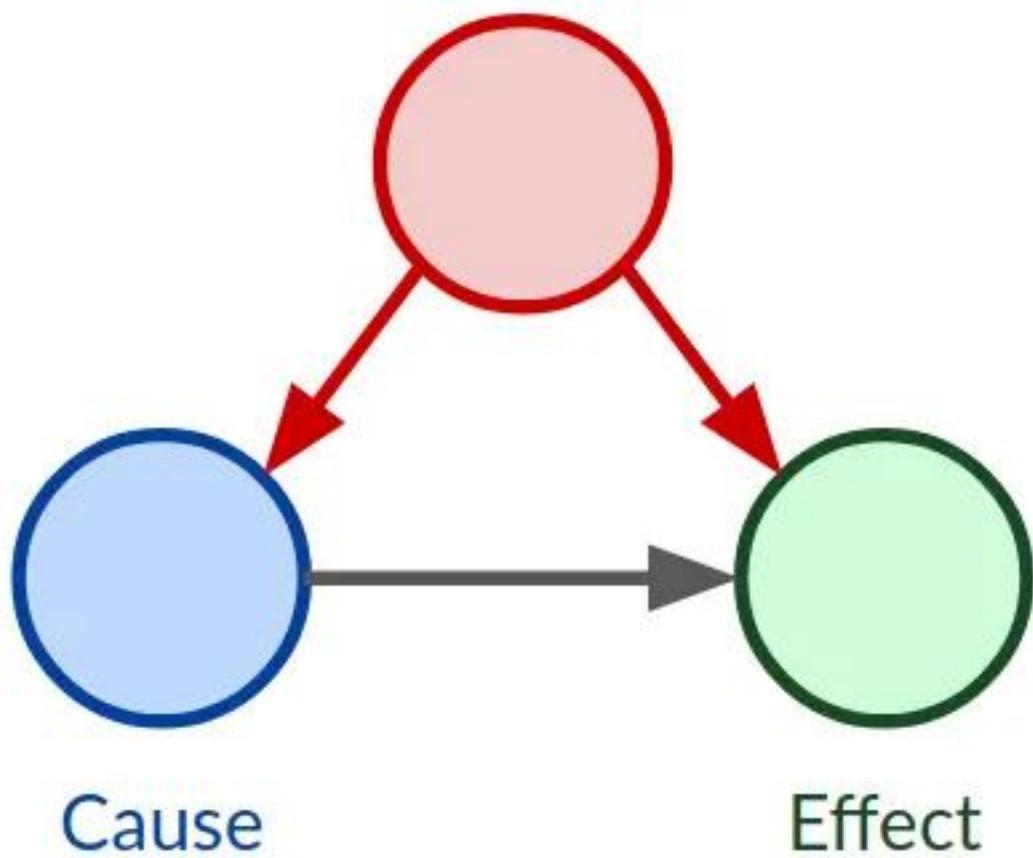
1. Correlation does not imply causation

A study found that as ice cream sales increase, so do shark attacks. These two variables are positively correlated, meaning they tend to rise and fall together. Does this mean eating ice cream causes shark attacks? Or do shark attacks somehow lead to increased ice cream sales?



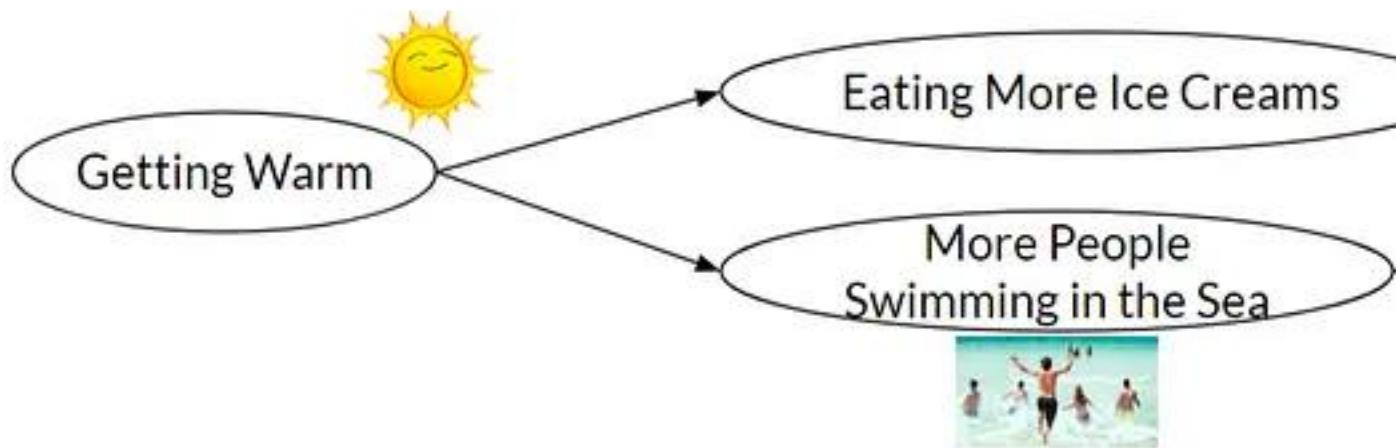
The phrase "[correlation does not imply causation](#)" is a fundamental concept in statistics and data analysis. It reminds us that even if **two variables have a strong correlation**, we **cannot** conclude that **one causes the other**. It is possible that the observed relationship is driven by some **third variable** known as **confounding variable** that affects both of the variables

Confounding variable



So the answer to the previous example is no. One variable is not directly causing the other. Instead, there's a confounding variable: **temperature**.

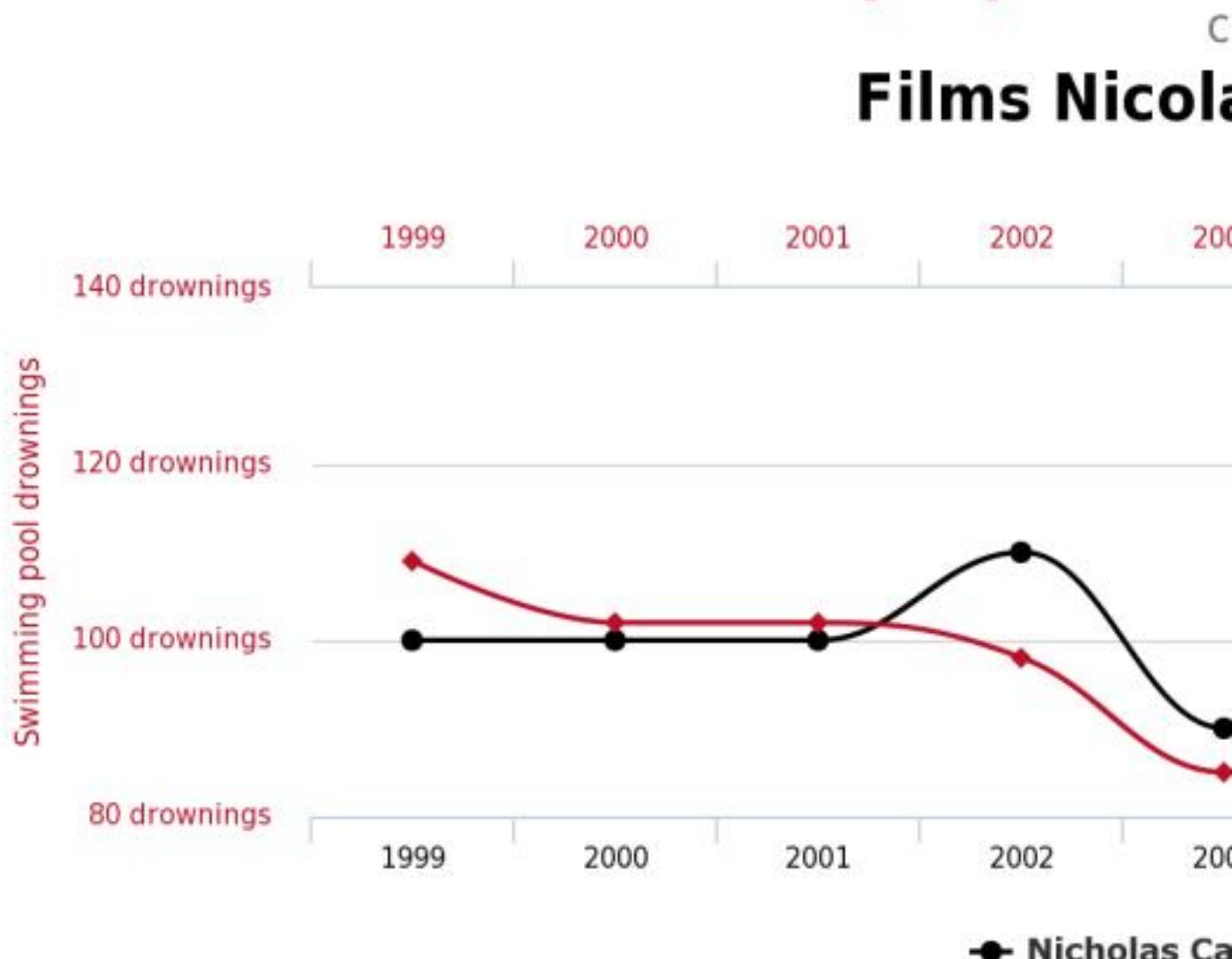
In warmer weather, more people go to the beach, increasing the likelihood of shark encounters (and thus attacks). At the same time, warm weather also leads to higher ice cream consumption as people seek to cool off. In this case, temperature is the confounding variable that explains the observed correlation between ice cream sales and shark attacks. Without accounting for this third factor, we might mistakenly conclude that there is a causal relationship between ice cream and shark attacks.



2. Coincidental Correlation:

Occasionally, two variables correlate purely **by chance** without any logical relationship. For example, the number of films Nicolas Cage appeared in each year and the number of people who drowned in swimming pools have a surprising correlation, but they have no causal relationship. ([Source](#))

Number of people who



Create scatter plot using Seaborn

Let's examine the relationship between happiness score and social support

In [12]:

```
sns.scatterplot(x="social_support", y="happiness_score", data=df);
```

We observe a moderate positive relationship between social support and happiness scores, indicating that an increase in social support is associated with higher happiness scores in the countries.

Let's examine the relationship between happiness score and social support, colored by region

In [13]:

```
plt.figure(figsize=(15, 6))
s = sns.scatterplot(x="social_support", y="happiness_score", data=df,
hue='region');
plt.legend(loc='upper left')
sns.move_legend(s, "upper left", title='Region', bbox_to_anchor=(1, 1))
```

Let's examine the relationship between happiness score and generosity

In [14]:

```
sns.scatterplot(x="generosity", y="happiness_score", data=df);
```

Let's examine the relationship between happiness score and generosity, colored by region

In [15]:

```
plt.figure(figsize=(15, 6))
s = sns.scatterplot(x="generosity", y="happiness_score", data=df,
hue='region');
plt.legend(loc='upper left')
sns.move_legend(s, "upper left", title='Region', bbox_to_anchor=(1, 1))
```

There appears to be no correlation between happiness score and generosity, as an increase in generosity does not correspond to a noticeable rise in happiness scores. However, there are three countries with exceptionally high generosity levels but relatively moderate happiness scores, which is unusual because higher generosity is often associated with higher happiness. This outlier might represent a country where people are generally generous, but other factors (like economic or political challenges) are likely affecting the overall happiness score.

Let's examine the relationship between happiness score and perceptions of corruption

In [16]:

```
sns.scatterplot(x="perceptions_of_corruption", y="happiness_score", data=df);
```

The scatter plot above shows the relationship between the happiness score and perceptions of corruption. Here's how we can interpret the plot:

- **General trend:** There seems to be a slight positive correlation between happiness scores and perceptions of corruption. Higher happiness scores tend to correspond with a wider range of corruption perceptions, with values for corruption perception increasing as the happiness score increases. However, this trend is not very strong, as the data points are quite spread out.
- **Dense area:** Many data points are concentrated in the left-middle part of the plot, where happiness scores range from about 4 to 6, and corruption perceptions range from 0.0 to 0.2. This suggests that a large number of countries have moderate happiness scores with relatively low perceptions of corruption.
- **Higher spread at higher happiness scores:** At higher happiness scores (6 to 8), perceptions of corruption vary more widely (from 0.1 to 0.5), indicating more variability in perceptions of corruption among happier countries.
- **Potential outliers:** A few points are higher up (above 0.4) in perceptions of corruption with low happiness scores, which might represent countries with unique characteristics or governance structures.

Let's examine the relationship between happiness score and perceptions of corruption, colored by region

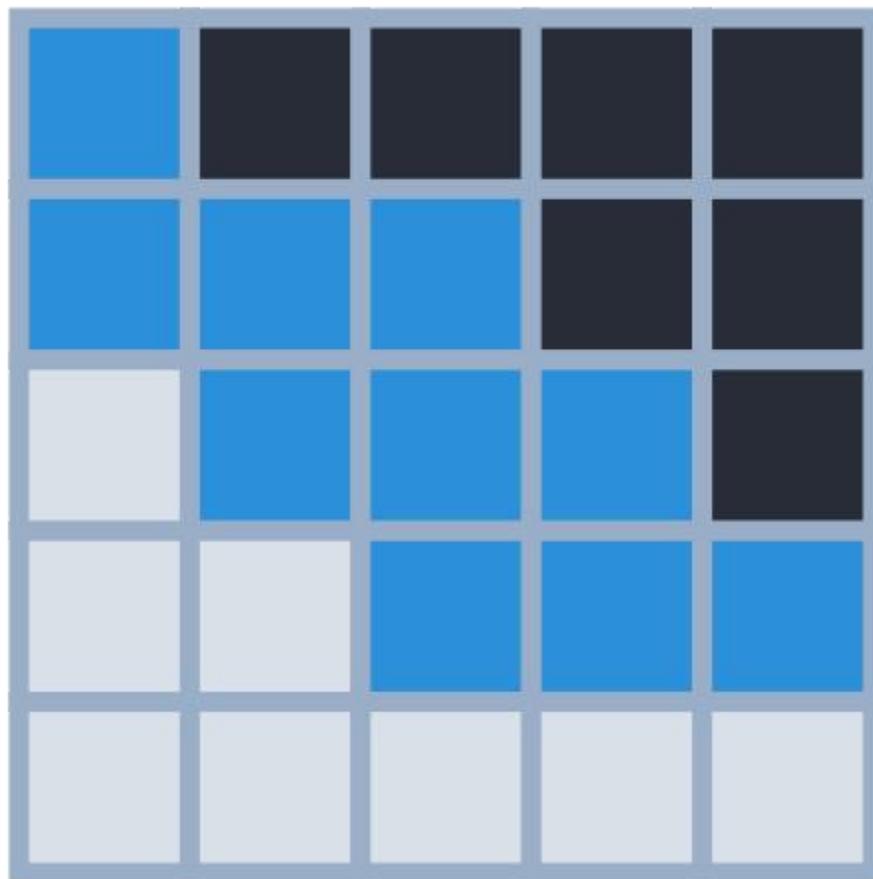
In [17]:

```
s = sns.scatterplot(x="perceptions_of_corruption", y="happiness_score",
                     data=df, hue='region');
plt.legend(loc='upper left')
sns.move_legend(s, "upper left", title='Region', bbox_to_anchor=(1, 1))
```

- **Western Europe :** Western European countries are primarily located in the upper portion of the plot, with high happiness scores (6 to 8) and relatively high perceptions of corruption (0.1 to 0.4). This concentration supports the idea that highly happy countries exhibit more variability in corruption perception, as Western Europe appears to have a broader range in perceptions of corruption even with high happiness.
- **Sub-Saharan Africa and South Asia :** These regions are mostly found in the lower left, where both happiness scores and perceptions of corruption are lower. This indicates that these regions tend to have lower happiness scores with a smaller range of corruption perception.
- **North America and ANZ and Latin America and Caribbean** show a moderate range in both happiness scores and corruption perceptions, primarily centered around mid-to-high happiness scores (5 to 7) with moderate corruption perceptions.
- **Middle East and North Africa and East Asia** have more spread in both directions, indicating diverse countries in these regions, some with moderate happiness scores but differing corruption perceptions.

- **Outliers:** A few Sub-Saharan Africa countries appear to have low happiness scores and relatively high perceptions of corruption (above 0.4), which could indicate unique cultural or governance factors influencing these perceptions.

Heatmap

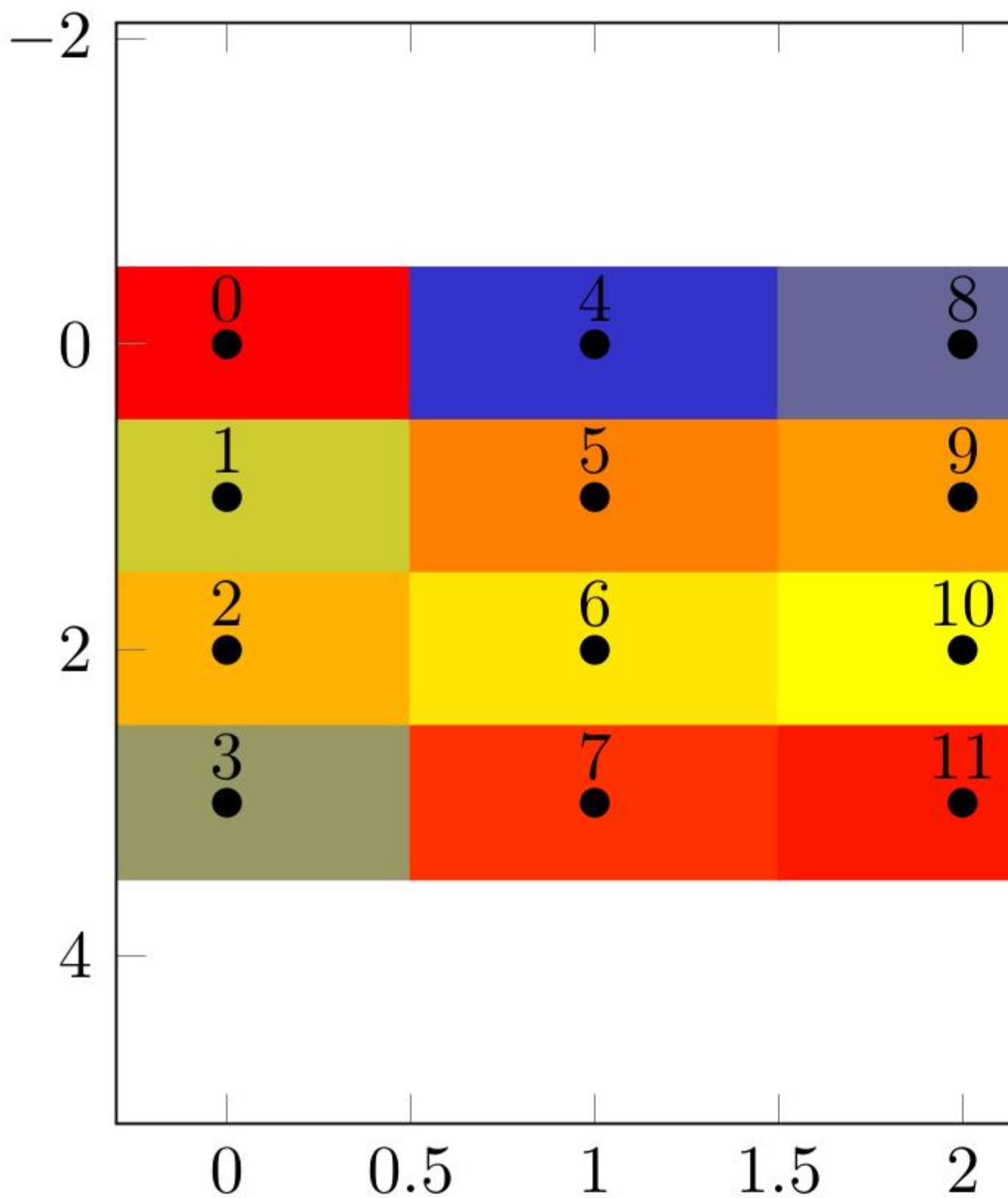


A **heatmap** is a powerful data visualization tool that **uses color gradients** to represent the **magnitude of data values** in a matrix or table. It is particularly useful for exploring relationships between multiple variables or observing patterns in data.

A heatmap visualizes data in a tabular format, where:

- Rows and columns represent variables or categories.
- The color of each cell represents the value at the intersection of the row and column.

The color gradient typically ranges from light to dark or from one color to another, indicating lower and higher values.



Create heatmap using Seaborn

Explore the six factors by region

```
matrix = df.drop(columns = ['year', 'country',  
'happiness_score']).groupby('region').mean()  
matrix
```

In [18]:

Out[18]:

	gdp_per_capita	social_support	healthy_life_expectancy	freedom_to_make_life_choices
region				
Africa	0.222025	0.855070	0.384905	0.428560
Central and Eastern Europe	1.189778	1.151242	0.677170	0.403950
Commonwealth of Independent States	0.892170	1.258676	0.568093	0.542065
East Asia	1.350287	1.168411	0.804831	0.443767
Latin America and Caribbean	1.016806	1.147372	0.647769	0.504863
Middle East and North Africa	1.162901	0.974069	0.632208	0.373024
North America and ANZ	1.501825	1.351809	0.817846	0.605989
South Asia	0.732813	0.747322	0.489655	0.423729
Southeast Asia	1.015817	1.056835	0.593544	0.589349
Sub-Saharan Africa	0.550940	0.786000	0.279074	0.377608
Western Europe	1.490442	1.297412	0.843562	0.548094

Is it easy to find the highest and lowest values for each factor among all regions?

```
sns.heatmap(matrix, annot=True, cmap='coolwarm')  
plt.title("Mean factor per Region")  
plt.show()
```

In [19]:

- From the heatmap above, we can see that "gdp_per_capita" and "social_support" have the greatest impact on a country's happiness score, as these factors consistently show higher values (warmer colors) compared to other factors across most regions.
- Additionally, we observe that "perceptions_of_corruption" generally has the lowest values, indicating it has the least impact on happiness.
- We can also identify the highest value among all factors across all regions, which is 1.5 for "gdp_per_capita" in Western Europe and North America & ANZ.

Heatmaps are widely used to visualize **correlations** between variables in a dataset. **Each cell represents the correlation coefficient between two variables.**

In [20]:

```
cor = df.corr(numeric_only=True)
cor
```

Out[20]:

	happiness_score	gdp_per_capita	social_support	healthy_life_expectancy
happiness_score	1.000000	0.723810	0.648155	0.682400
gdp_per_capita	0.723810	1.000000	0.522092	0.564953
social_support	0.648155	0.522092	1.000000	0.561274
healthy_life_expectancy	0.682400	0.564953	0.561274	1.000000
freedom_to_make_life_choices	0.569458	0.439453	0.364927	0.261374
generosity	0.082345	-0.153904	-0.006385	0.025932
perceptions_of_corruption	0.415071	0.338105	0.147278	0.256215

In [21]:

```
sns.heatmap(cor, annot=True, cmap='coolwarm')
plt.title("Correlation")
plt.show()
```

The heatmap above allows us to quickly identify strong and weak correlations between the various factors and the happiness score, as well as correlations among the factors themselves.

</Data Visualization>