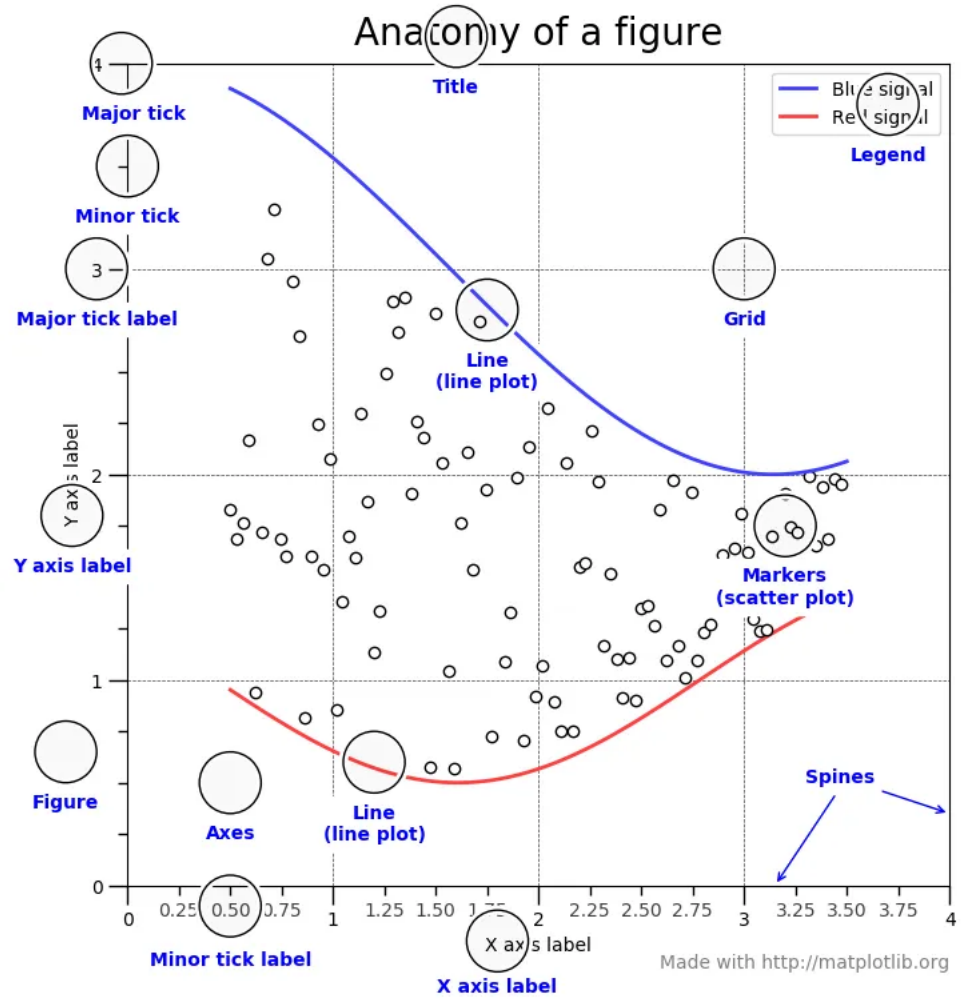


Line Plot

Scatter Plot

Box plot

Point plot

Count plot

Violin plot

Swarm plot

Bar plot

KDE Plot

Heatmap

Barplot is a numerical plot that displays the mean value of a numerical variable for each category.

Case Study - Online Retailer

Example - 1

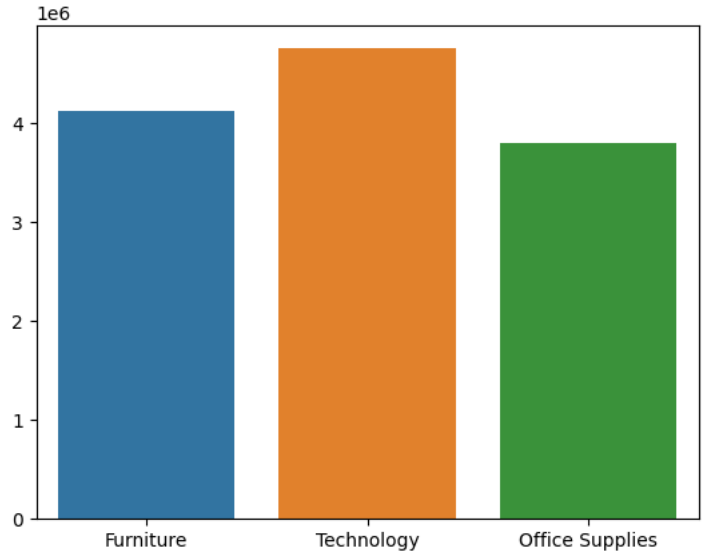
Bar Chart - Plot sales across each product category

- A bar chart uses bars to show comparisons between categories of data.
- A bar graph will always have two axis.
- One axis will generally have numerical values or measures,
- The other will describe the types of categories being compared or dimensions.

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import seaborn as sns
4
```

```
product_category = np.array(['Furniture', 'Technology', 'Office Supplies'])
sales = np.array ([4110451.90, 4744557.50, 3787492.52] )

# bar plot of product category
sns.barplot(x=product_category,y=sales)
```

```
<Axes: >
```




```
sales.reshape(3,1)
```

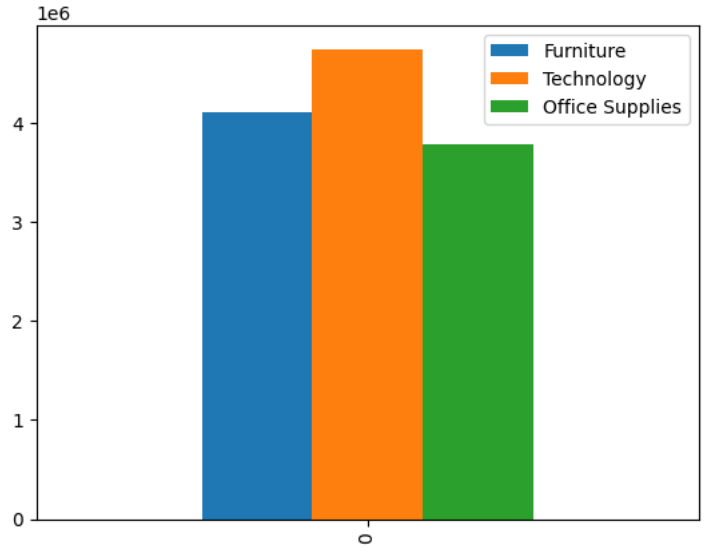
```
array([[4110451.9 ],
       [4744557.5 ],
       [3787492.52]])
```

```
import pandas as pd

#craete a dataframe of product category
df=pd.DataFrame(sales.reshape(1,3))
df.columns=product_category

df
```

	Furniture	Technology	Office Supplies
0	4110451.9	4744557.5	3787492.52

```
# Plot bar plot using pandas visualization
df.plot.bar()

```

```
<Axes: >
```

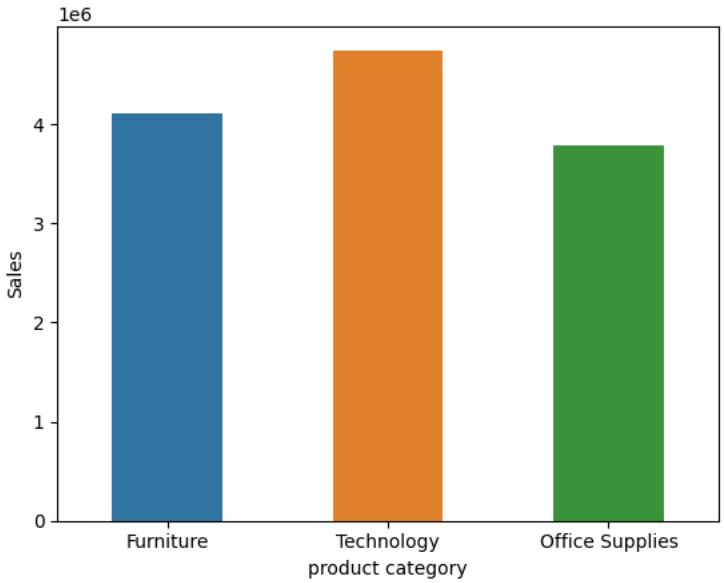


Bar Chart - Plot sales across each product category

1. Adding labels to Axes

2. Reducing the bar width
3. Giving Title to the chart
4. Modifying the ticks to show information in (million dollars)

```
import numpy as np
import matplotlib.pyplot as plt

product_category = np.array(['Furniture', 'Technology', 'Office Supplies'])
sales = np.array ([4110451.90, 4744557.50, 3787492.52] )

# plotting bar chart and setting bar width to 0.5 and aligning it to center
sns.barplot(x=product_category,y=sales,width=0.5)
# Adding and formatting title
# plt.title("sales by product category")

# Labeling Axes
plt.xlabel('product category')
plt.ylabel('Sales')

# Modifying the ticks to show information in (million dollars)
# plt.yticks(sales/1e6)

plt.show()
```



Example - 2

Scatter Chart - Plot Sales versus Profits across various Countries and Product Categories

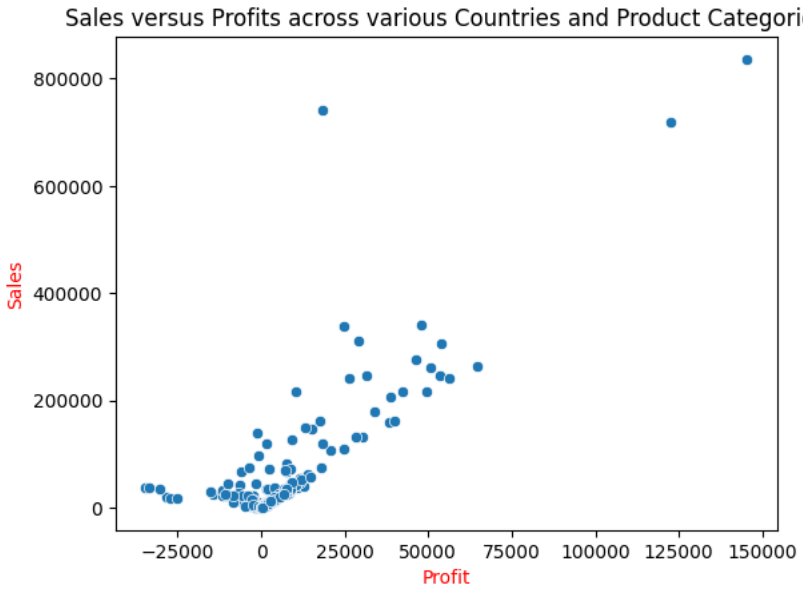
Scatter plots are used when you want to show the relationship between two facts or measures.

- Profit is measured across each product category

```
import numpy as np
import matplotlib.pyplot as plt

# Data
sales = np.array ([1013.14, 8298.48, 875.51, 22320.83, 9251.6, 4516.86, 585.16, 836154.03, 216748.48, 174.2, 27557.79, 56
profit = np.array([-1213.46, 1814.13, -1485.7, -2286.73, -2872.12, 946.8, 198.48, 145454.95, 49476.1, -245.56, 5980.77, -
product_category = np.array(['Technology', 'Technology', 'Technology', 'Technology', 'Technology', 'Technology', 'Technol
country = np.array(['Zimbabwe', 'Zambia', 'Yemen', 'Vietnam', 'Venezuela', 'Uzbekistan', 'Uruguay', 'United States', 'Uni
```

```
# plotting scatter chart profit vs sales

# print(np.unique(country).size)
plot=sns.scatterplot(x=profit,y=sales)

# sns.barplot(data=df, x='day', y='tip', ci=None).

# Adding and formatting title
plot.set_title('Sales versus Profits across various Countries and Product Categories')

# Labeling Axes
plt.xlabel('Profit',color='red')
plt.ylabel('Sales',color='red')
```

```
14
15 plt.show()
```

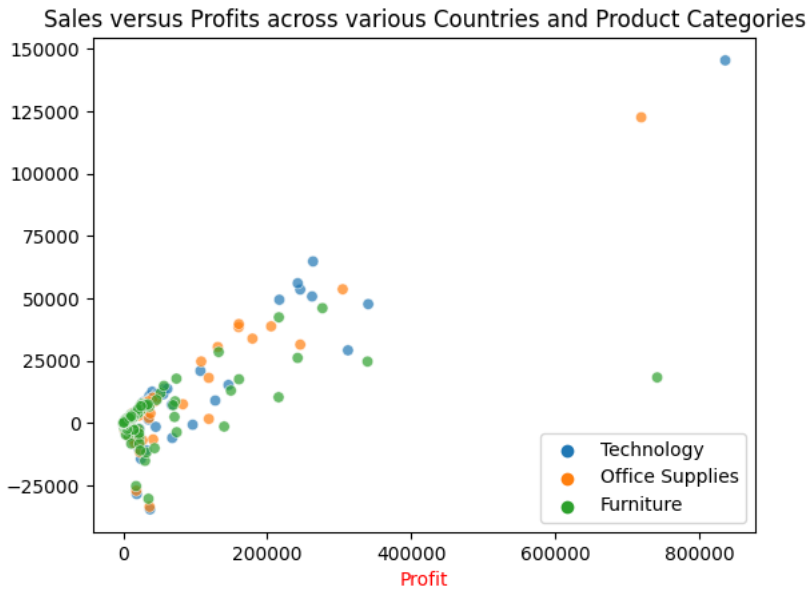


Scatter Chart - Plot Sales versus Profits across various Countries and Product Categories

- Represent product category using different colors
- Adding a Legend to Product Categories

```
1 plot=sns.scatterplot(x=sales,y=profit,hue=product_category,alpha=0.7)
2 # Adding and formatting title
3 plot.set_title('Sales versus Profits across various Countries and Product Categories')
4 plt.legend(loc='lower right')
5
6 # Labeling Axes
7 plt.xlabel('Sales',color='red')
8 plt.xlabel('Profit',color='red')
```

```
Text(0.5, 0, 'Profit')
```



Example - 3

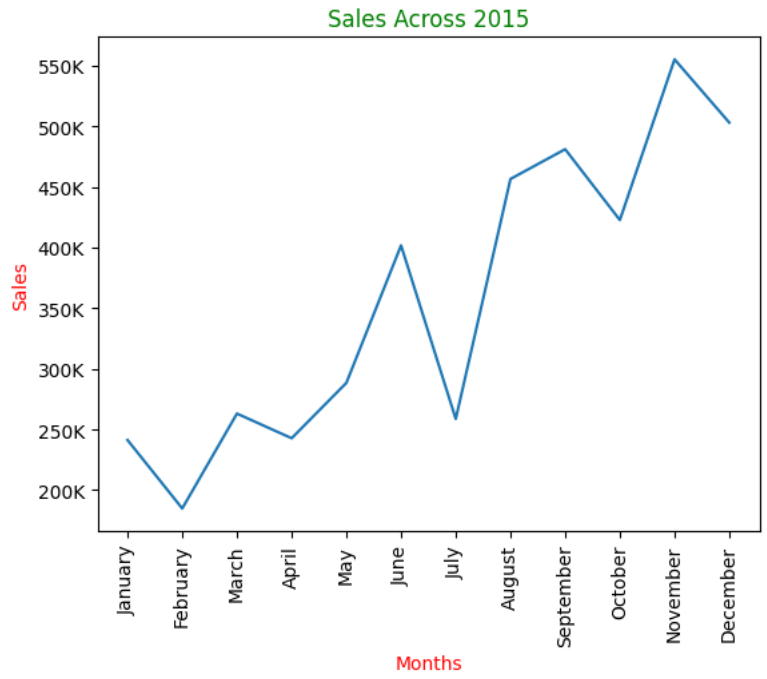
Line Chart - Plot Sales across 2015

A line chart or line plot or line graph or curve chart is a type of chart which displays information as a series of data points called 'markers' connected by straight line segments. Most commonly used with time data.

```
import numpy as np
import matplotlib.pyplot as plt

months = np.array(['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'Nov
sales = np.array([241268.56, 184837.36, 263100.77, 242771.86, 288401.05, 401814.06, 258705.68, 456619.94, 481157.24, 4227
```

```
# plot line plot
p=sns.lineplot(x=months,y=(sales/1000))
# Adding and formatting title
p.set_title('Sales Across 2015',color='green')
plt.xticks(months,rotation=90)

label=np.arange(200,600,50).astype('str')
label=np.char.add(label,'K')
plt.yticks(np.arange(200,600,50),label)

# Labeling Axes
plt.xlabel('Months',color='red')
plt.ylabel('Sales',color='red')
```

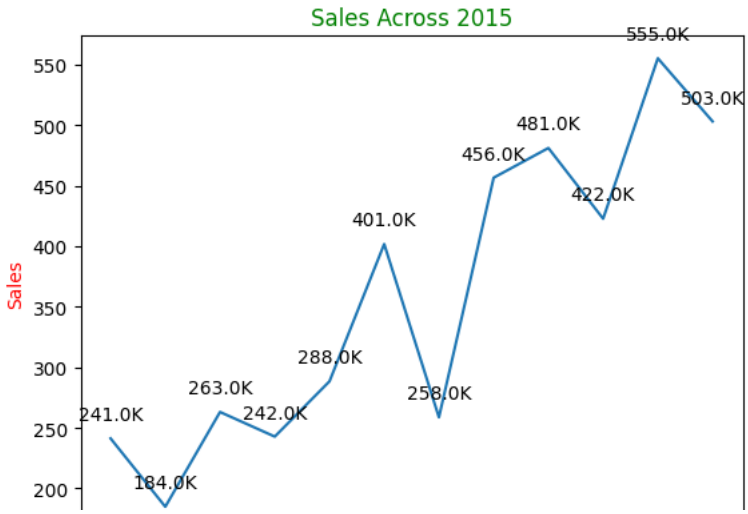
```
Text(0, 0.5, 'Sales')
```



Line Chart - Plot Sales across 2015

- Add labels to marks

```
# plot line plot
p=sns.lineplot(x=months,y=(sales/1000))
# Adding and formatting title
p.set_title('Sales Across 2015',color='green')
plt.xticks(months,rotation=90)

# Labeling Axes
plt.xlabel('Months',color='red')
plt.ylabel('Sales',color='red')


# add annotation in graph
for month, sale in zip(months, sales):
    plt.annotate(f'{sale//1000}K', xy=(month, sale/1000), textcoords='offset points', xytext=(0, 10), ha='center')

plt.show()
```



Example - 4

Box and Whisker Chart - Sales across Countries and Product Categories

A Box and Whisker Plot (or Box Plot) is a convenient way of visually displaying the data distribution through their quartiles. The lines extending parallel from the boxes are known as the “whiskers”, which are used to indicate variability outside the upper and lower quartiles. Outliers are sometimes plotted as individual dots that are in-line with whiskers. Box Plots can be drawn either vertically or horizontally.

```
import numpy as np
import matplotlib.pyplot as plt

# Data
sales_technology = np.array ([1013.14, 8298.48, 875.51, 22320.83, 9251.6, 4516.86, 585.16, 174.2, 27557.79, 563.25, 558.1
sales_office_supplies = np.array ([1770.13, 7527.18, 1433.65, 423.3, 21601.72, 10035.72, 2378.49, 3062.38, 345.17, 30345.
sales_furniture = np.array ([981.84, 10209.84, 156.56, 243.06, 21287.52, 7300.51, 434.52, 6065.0, 224.75, 28953.6, 757.98
```

```
# import numpy as np
# import matplotlib.pyplot as plt

# # Data
# sales = np.array ([1013.14, 8298.48, 875.51, 22320.83, 9251.6, 4516.86, 585.16, 836154.03, 216748.48, 174.2, 27557.79,
# profit = np.array([-1213.46, 1814.13, -1485.7, -2286.73, -2872.12, 946.8, 198.48, 145454.95, 49476.1, -245.56, 5980.77,
# product_category = np.array(['Technology', 'Technology', 'Technology', 'Technology', 'Technology', 'Technology', 'Techr
# country = np.array(['Zimbabwe', 'Zambia', 'Yemen', 'Vietnam', 'Venezuela', 'Uzbekistan', 'Uruguay', 'United States', 'U
```

```

pl=sns.boxplot(data=[sales_technology,sales_office_supplies,sales_furniture],width=0.7)
# Adding and formatting title
pl.set_title('Sales Across Various Countries and Product Categories',color='green')
plt.xticks([0,1,2],labels=['Technology','Office Supplies','Furniture'])

# Labeling Axes
plt.xlabel('Product Category',color='red')
plt.ylabel('Sales',color='red')
```

```
Text(0, 0.5, 'Sales')
```

Example - 5

Histogram - Plot a histogram for Sales distribution across Countries.

A histogram is a plot that lets you discover, and show, the underlying frequency distribution (shape) of a set of continuous data.

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 # Data
5 profit = np.array([-5428.79, 7001.73, -3706.46, 300.42, -1697.31, -11222.71, 1648.14, 1689.34, -1036.86, 20944.23, -2426.
6
```

Example - 6

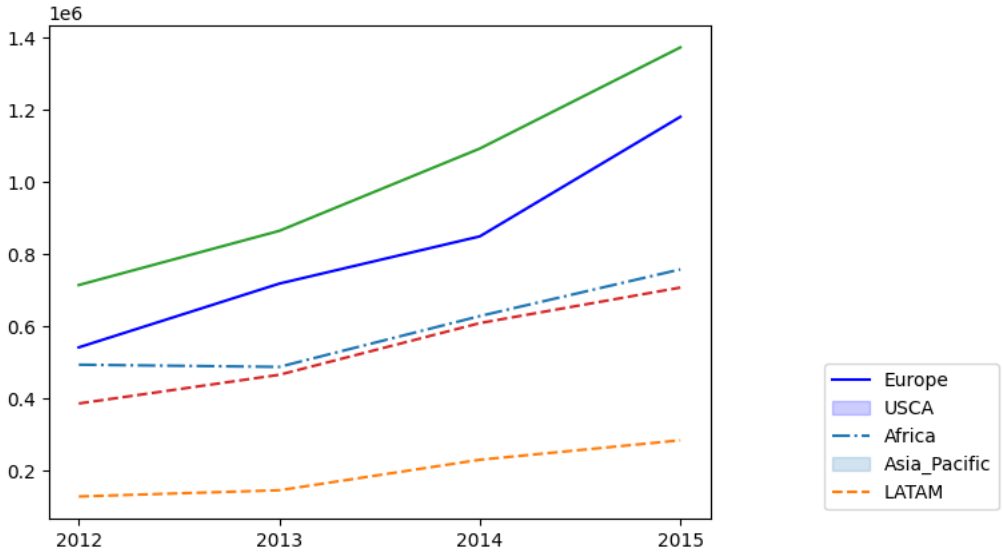
SubPlots - Plot Sales for various markets for years 2012 to 2015

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 years = np.array(['2012', '2013', '2014', '2015'])
5
6 sales_africa = np.array([127187.27, 144480.70, 229068.79, 283036.44])
7
8 sales_USCA = np.array([492756.60, 486629.30, 627634.98, 757108.13])
9
10 sales_LATAM = np.array([385098.15, 464733.29, 608140.77, 706632.93])
11
12 sales_Asia_Pacific = np.array([713658.22, 863983.97, 1092231.65, 1372784.40])
13
14 sales_Europe = np.array([540750.63, 717611.40, 848670.24, 1180303.95])
```

Subplots are shown in the same graph as line charts, identified by different colours

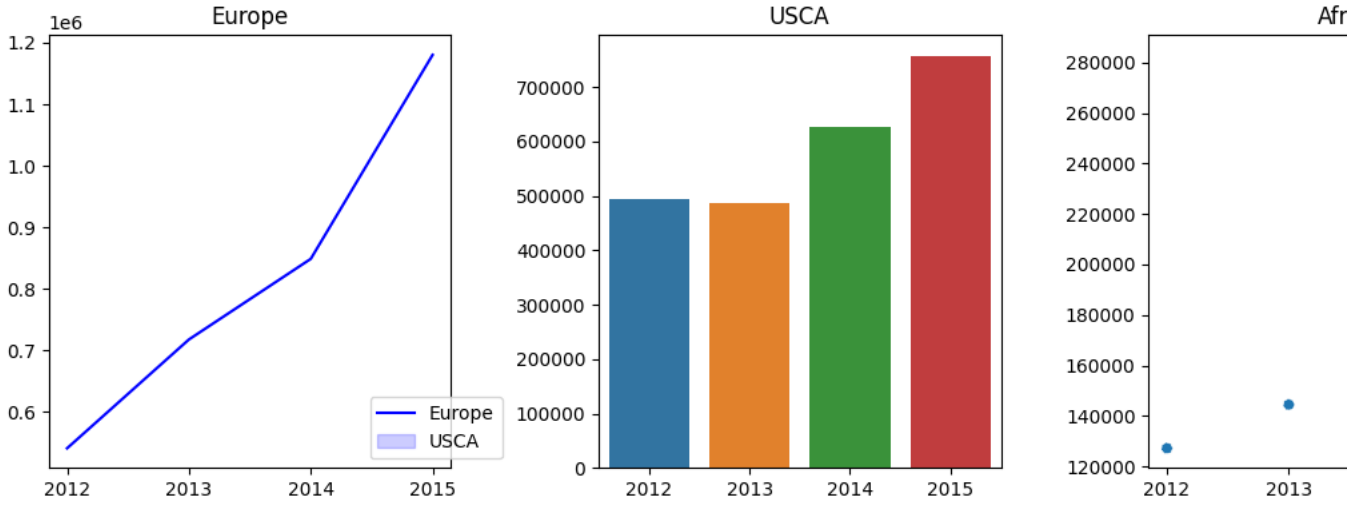
```
1 sns.lineplot(x=years,y=sales_Europe,color='blue')
2 sns.lineplot(x=years,y=sales_USCA,linestyle='dashdot')
3 sns.lineplot(x=years,y=sales_africa,linestyle='--')
4 sns.lineplot(x=years,y=sales_Asia_Pacific,linestyle='-')
5 sns.lineplot(x=years,y=sales_LATAM,linestyle='--')
6
7 plt.legend(loc='lower right',labels=['Europe', 'USCA', 'Africa','Asia_Pacific','LATAM'],bbox_to_anchor=(1.45,0))
8
```

```
<matplotlib.legend.Legend at 0x7b9e928f73a0>
```



Subplots are shown in seperate graphs. Each chart can be of different types.

```
fig, axes = plt.subplots(1, 4, figsize=(16, 4))

sns.lineplot(x=years, y=sales_Europe, color='blue', ax=axes[0])
axes[0].set_title('Europe')

sns.barplot(x=years, y=sales_USCA, linestyle='dashdot', ax=axes[1])
axes[1].set_title('USCA')

sns.scatterplot(x=years, y=sales_africa, linestyle='--', ax=axes[2])
axes[2].set_title('Africa')

pd.DataFrame({ 'Asia Pacific': sales_Asia_Pacific, 'LATAM': sales_LATAM }).plot(ax=axes[3], kind='bar', stacked=True, co
axes[3].set_title('Asia Pacific')

axes[0].legend(labels=['Europe', 'USCA', 'Africa', 'Asia Pacific'], loc='lower right', bbox_to_anchor=(1.15, 0))

plt.tight_layout()
plt.show()
```



```
# Understanding the Gender Pay Gap in Hourly Earnings: An
# Analysis of Trends and Factors

# 1 read female_hourly_earnings.csv
# 2 get the list of names numerical and catogorical columns in a separate varaiable
# 3 count plot for country,gender code , gender
# 4 pie plot for country
# 5- separate hist and dist plot for country_id,  year, amount_local_currency
# 6-separate box plot for country_id,  year, amount_local_currency
# 7- Mean Earnings Over Time by Gender line plot
# 8- 'Mean Earnings by Country and Gender' plot bar plot
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

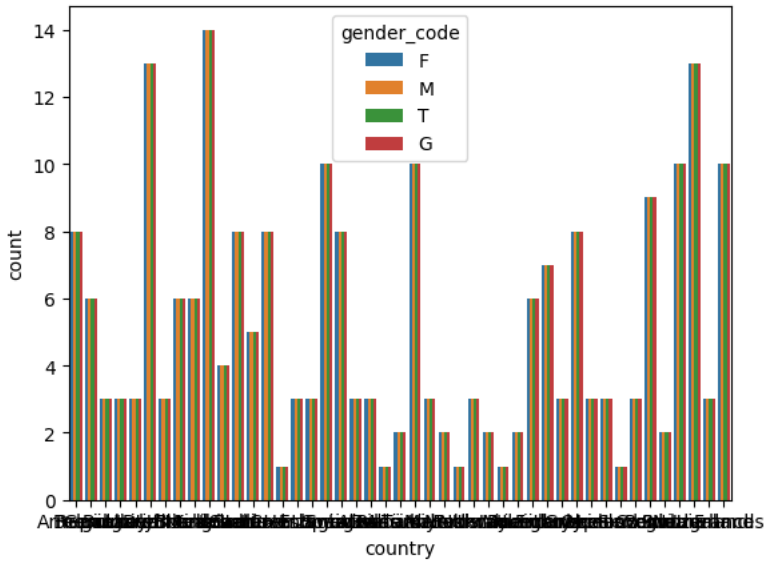
```
# 1 read female_hourly_earnings.csv
df = pd.read_csv('female_hourly_earnings.csv')
df.head(2)
```

	ID	country_id	country	gender_code	gender	year	amount_local_currency
0	0	51	Armenia	F	Female	2013	541.28
1	1	51	Armenia	F	Female	2014	569.36

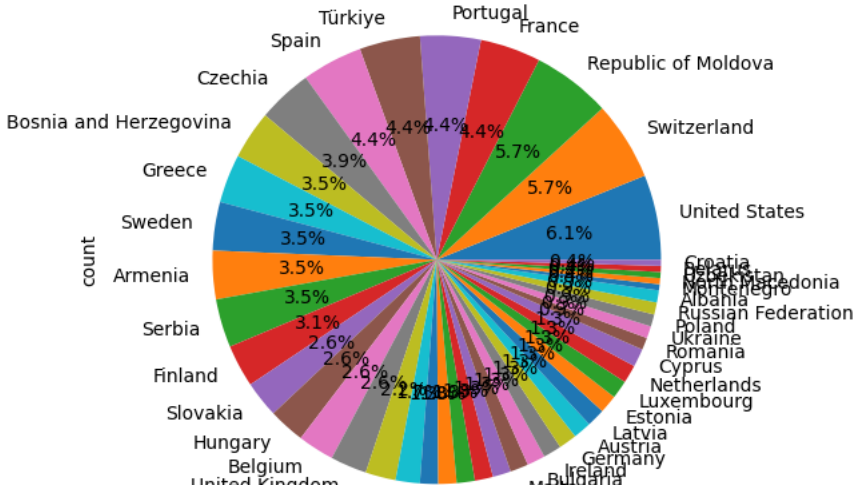
```
# 2 get the list of names numerical and catogorical columns in a separate varaiable

num_cols = list(df.select_dtypes(include=np.number).columns)
cat_cols = list(df.select_dtypes(include='object').columns)
print('Numerical columns:', num_cols)
print('Categorical columns:', cat_cols)
```

```
Numerical columns: ['ID', 'country_id', 'year', 'amount_local_currency']
Categorical columns: ['country', 'gender_code', 'gender']
```

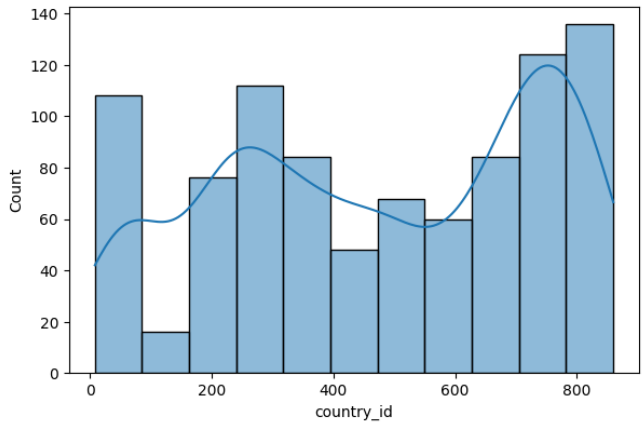
```
# 3 count plot for country,gender code , gender
sns.countplot(data=df, x='country', hue='gender_code')
plt.show()
```

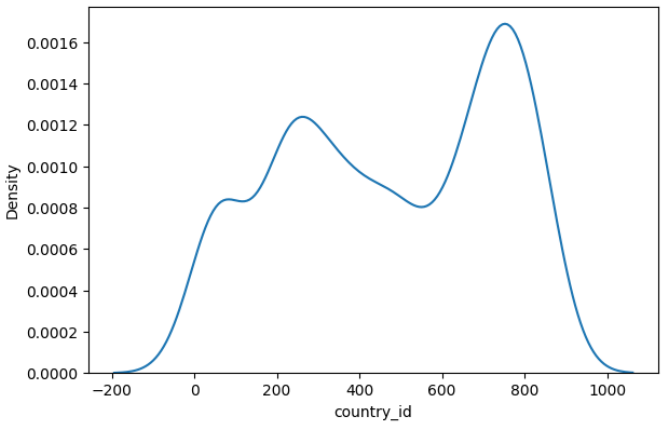



```
# 4 pie plot for country
df['country'].value_counts().plot(kind='pie', autopct='%1.1f%%')
plt.axis('equal')
plt.show()
```

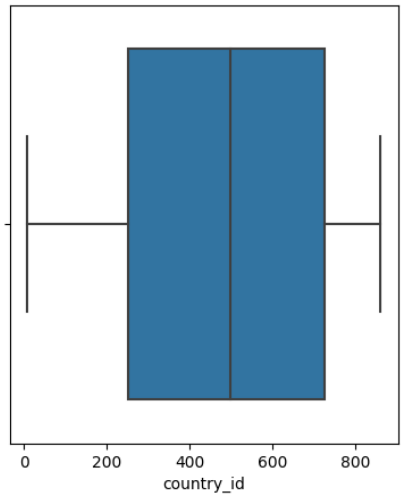


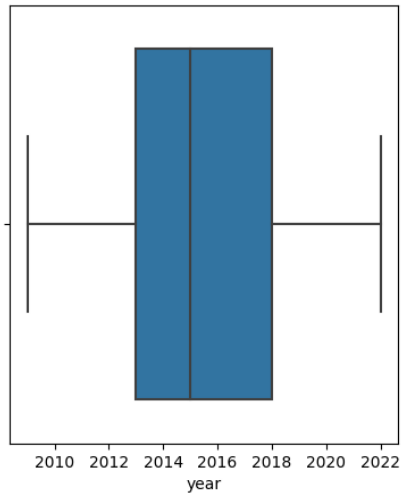

```
# 5 separate histogram and distribution plots for country_id, year, and amount_local_currency
fig, axs = plt.subplots(3, 2, figsize=(15, 15))

sns.histplot(data=df, x='country_id', kde=True, ax=axs[0, 0])
sns.kdeplot(data=df, x='country_id', ax=axs[0, 1])

sns.histplot(data=df, x='year', kde=True, ax=axs[1, 0])
sns.kdeplot(data=df, x='year', ax=axs[1, 1])

sns.histplot(data=df, x='amount_local_currency', kde=True, ax=axs[2, 0])
sns.kdeplot(data=df, x='amount_local_currency', ax=axs[2, 1])

plt.show()

```

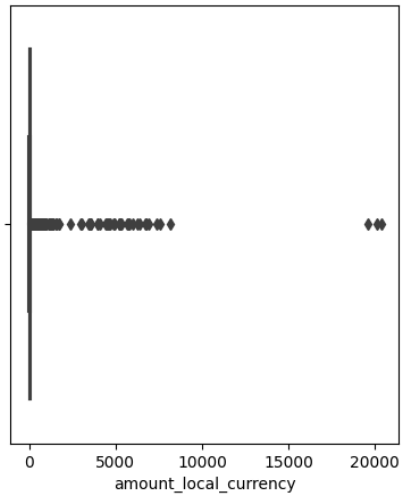





```
# 6-separate box plot for country_id,  year, amount_local_currency

fig, axs = plt.subplots(1, 3, figsize=(15, 5))

sns.boxplot(data=df, x='country_id', ax=axs[0])
sns.boxplot(data=df, x='year', ax=axs[1])
sns.boxplot(data=df, x='amount_local_currency', ax=axs[2])

plt.show()

```








```
df.columns
```

```
Index(['ID', 'country_id', 'country', 'gender_code', 'gender', 'year',
       'amount_local_currency'],
      dtype='object')
```

```
earnings_by_year_gender = df.groupby(['gender']).agg({'amount_local_currency': 'mean'})
earnings_by_year_gender
```

gender	amount_local_currency
Female	389.769039
Gender gap	88.531354
Male	478.300393
Total	444.082227

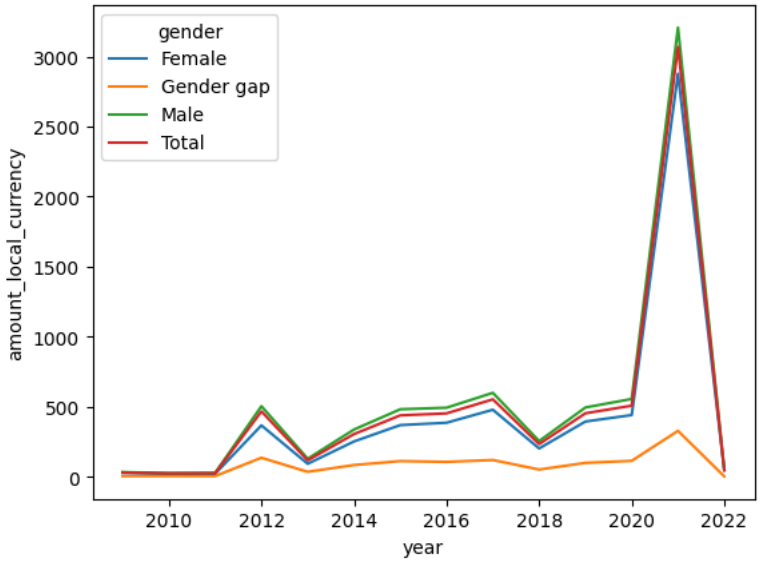
```
df.head(2)
```

```
earnings_by_year_gender = df.groupby(['gender', 'year']).agg({'amount_local_currency': 'mean'})
earnings_by_year_gender
```

gender	year	amount_local_currency
Female	2009	28.487500
	2010	21.190000
	2011	22.651667
	2012	367.716364
	2013	93.605455
	2014	253.120769
	2015	369.781053
	2016	386.647895
	2017	479.412353

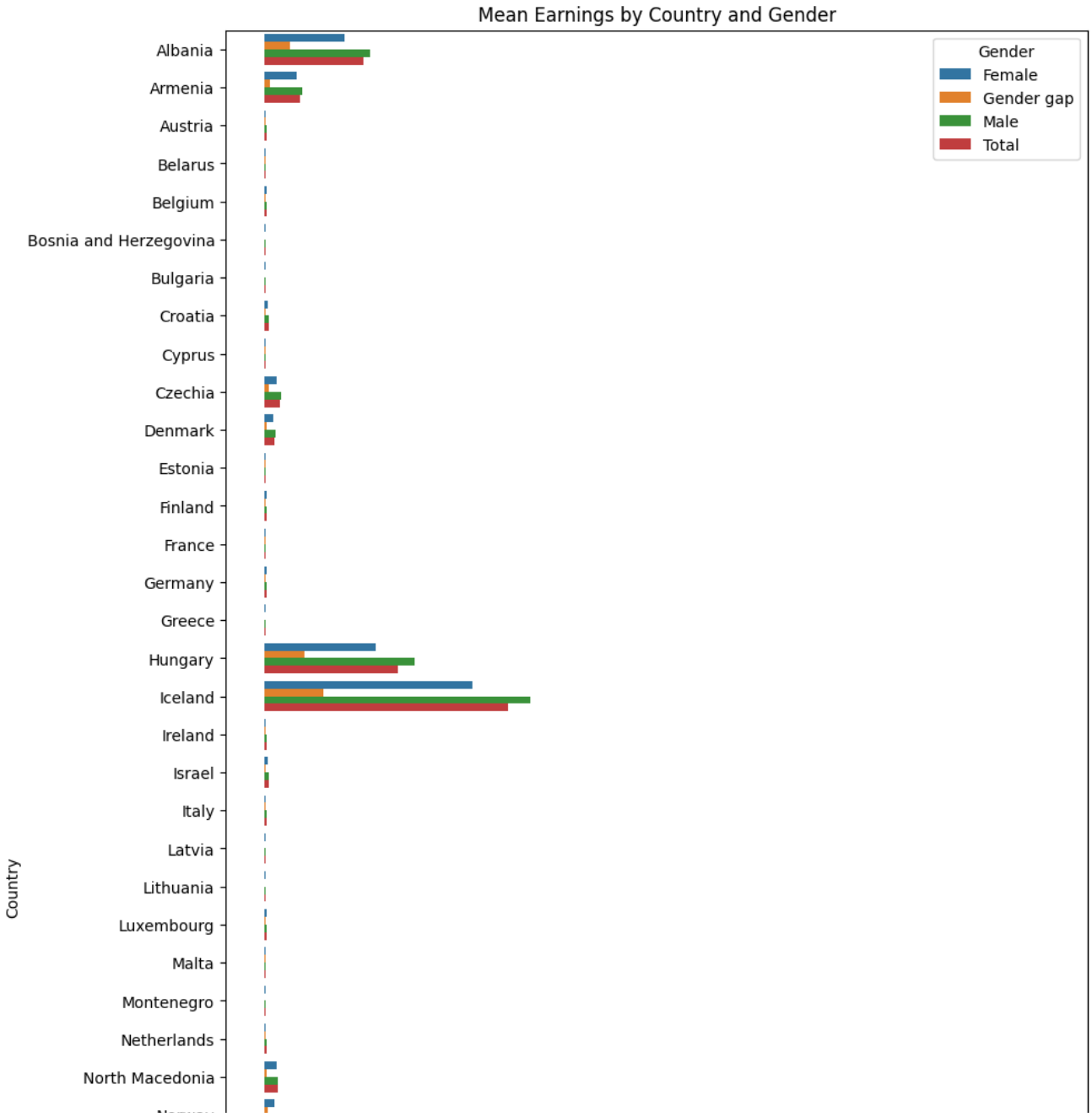
```
sns.lineplot(data=earnings_by_year_gender, x='year', y='amount_local_currency', hue='gender')
```

```
<Axes: xlabel='year', ylabel='amount_local_currency'>
```



	2020	114.839333

```
# 8- 'Mean Earnings by Country and Gender' plot bar plot
earnings_by_country_gender = df.groupby(['gender', 'country']).agg({'amount_local_currency': 'mean'})
earnings_by_country_gender

# plot sns bar where theres a bar for every gender which has horizontal bars for every country
plt.figure(figsize=(10, 20))
sns.barplot(x='amount_local_currency', y='country', hue='gender', data=earnings_by_country_gender.reset_index(), orient='
plt.title('Mean Earnings by Country and Gender')
plt.xlabel('Mean Earnings')
plt.ylabel('Country')
plt.legend(title='Gender')

plt.show()

# sns.barplot(data=earnings_by_country_gender, x='country', y='amount_local_currency', hue='gender')

```




```
pivot_df = earnings_by_country_gender.reset_index().pivot(index='country', columns='gender', values='amount_local_currenc

plt.figure(figsize=(20, 30))
ax = pivot_df.plot(kind='barh', stacked=True, )
plt.title('Mean Earnings by Country and Gender (Stacked)')
plt.xlabel('Mean Earnings')
plt.ylabel('Country')
plt.legend(title='Gender')
```