visualization in machine learning

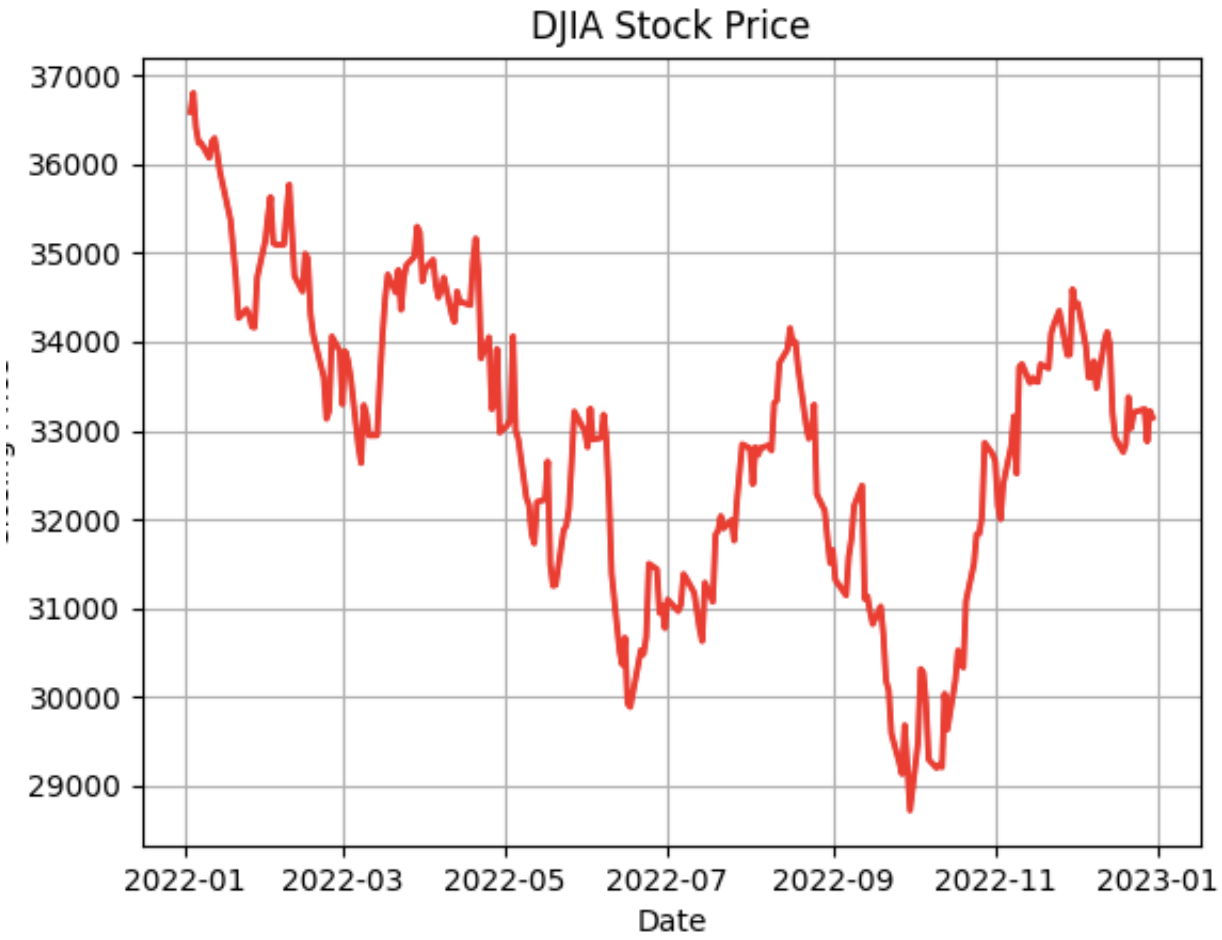
- Machine learning visualization (ML visualization for short) generally refers to the process of representing machine learning models, data, and their relationships through graphical or interactive means. The goal is to make comprehending a model's complex algorithms and data patterns easier, making it more accessible to technical and non-technical stakeholders.
- *Visualization bridges the gap between the enigmatic inner workings of ML models and our innate human capacity for understanding patterns and relationships through visuals.*
- Common model types, such as decision trees, support vector machines, or deep neural networks, often consist of many layers of computations and interactions that are challenging to grasp for humans. Visualization lets us see more easily how data flows through a model and where transformations occur.

Key Data Visualization Techniques

- Line plots
- Bar plots
- Histograms
- Box and whisker plots
- Scatter plots
- Bubble plot

Line plots

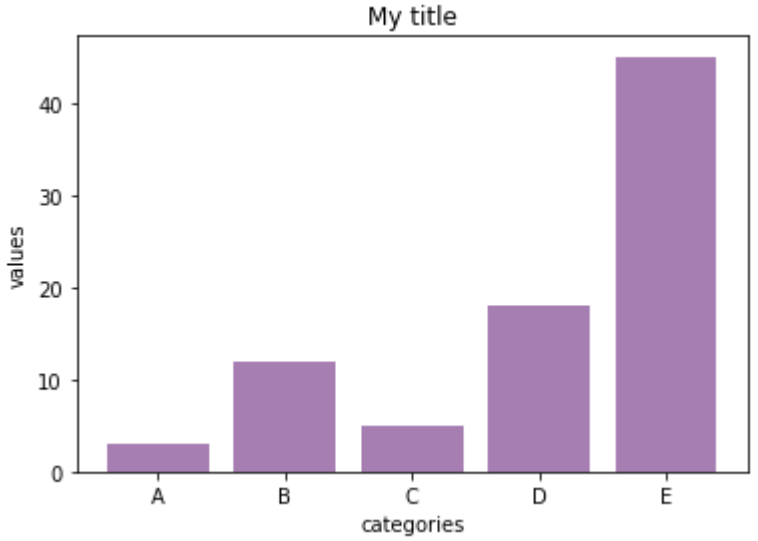
One of the most used visualizations, line plots are excellent at tracking the evolution of a variable over time. They are normally created by putting a time variable on the x-axis and the variable you want to analyze on the y-axis. For example, the line plot below shows the evolution of the DJIA Stock Price during 2022.

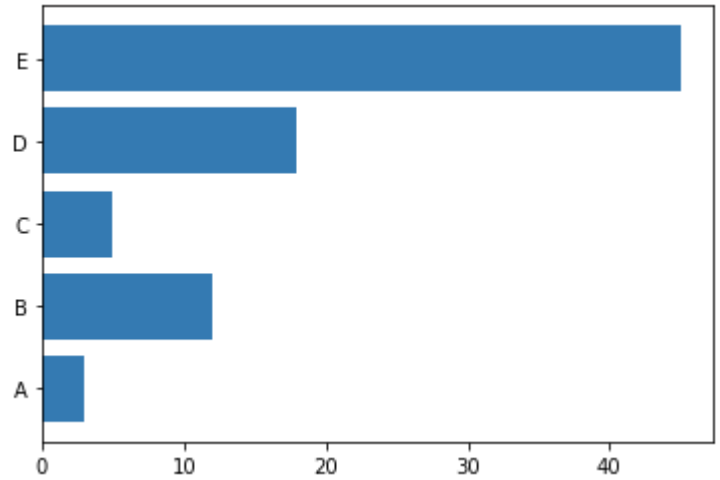


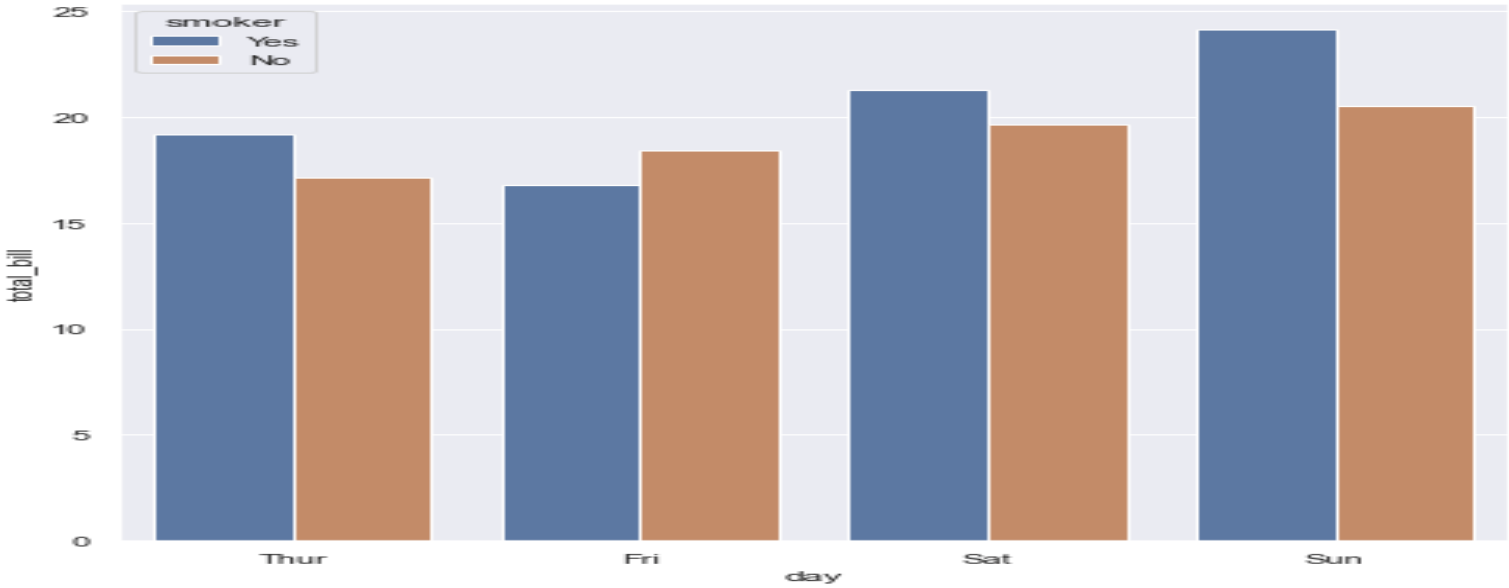
Bar plots

A bar chart ranks data according to the value of multiple categories. It consists of rectangles whose lengths are proportional to the value of each category. Bar charts are prevalent because they are easy to read. Businesses commonly use bar charts to make comparisons, like comparing the market share of different brands or the revenue of different regions. There are multiple types of bar charts, each suited for a different purpose.

There are multiple types of bar charts, each suited for a different purpose, including vertical bar plots, horizontal bar plots, and clustered bar plots.



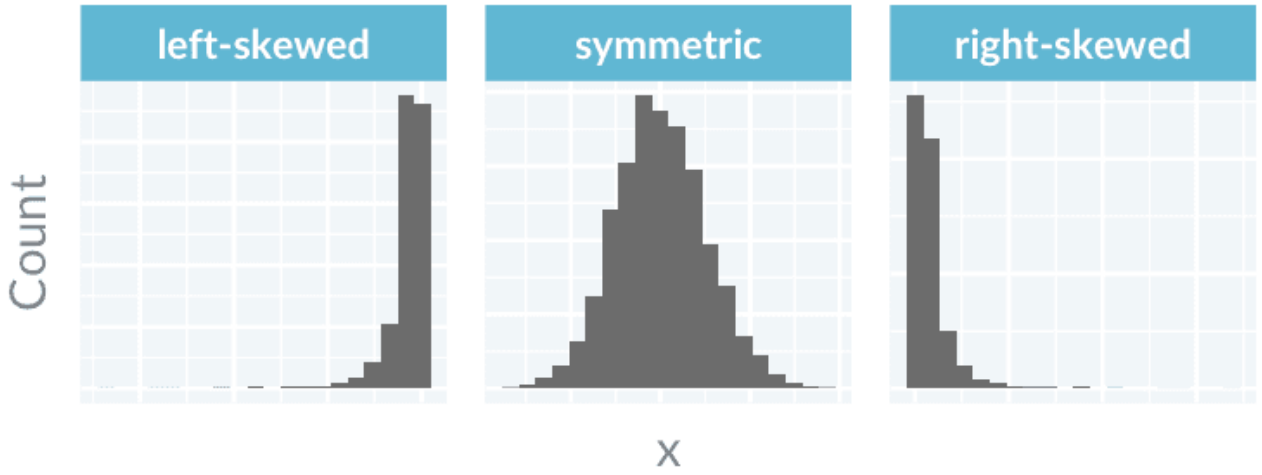




Histograms

Histograms are one of the most popular visualizations to analyze the distribution of data. They show the numerical variable's distribution with bars.

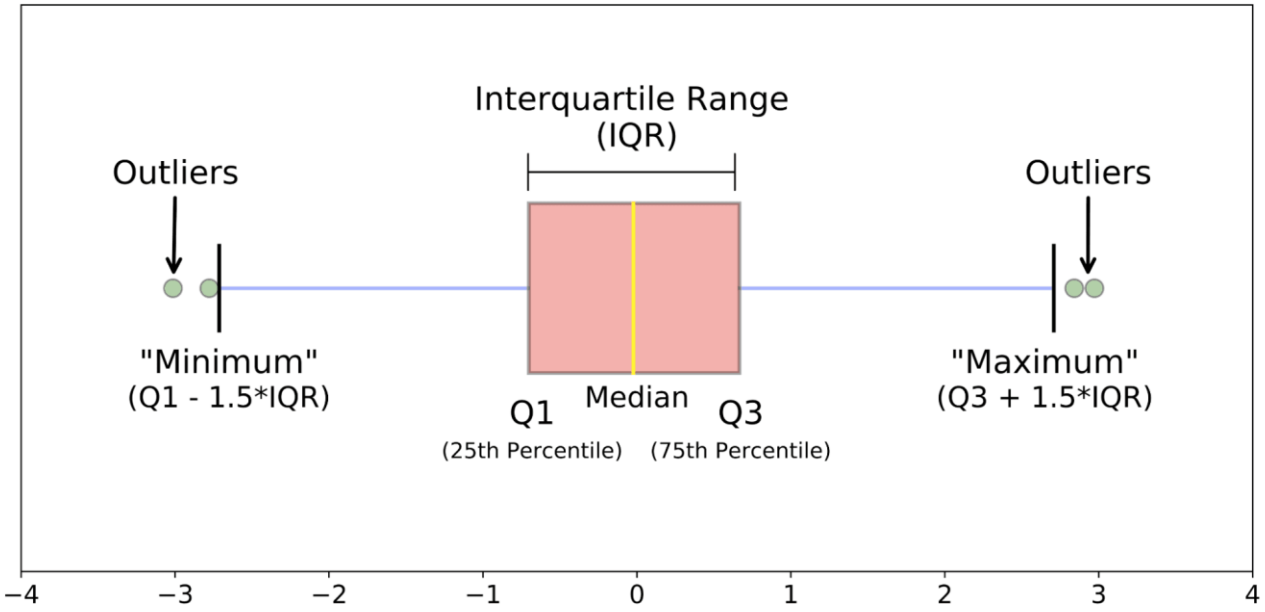
To build a histogram, the numerical data is first divided into several ranges or bins, and the frequency of occurrence of each range is counted. The horizontal axis shows the range, while the vertical axis represents the frequency or percentage of occurrences of a range.



Box and whisker plots

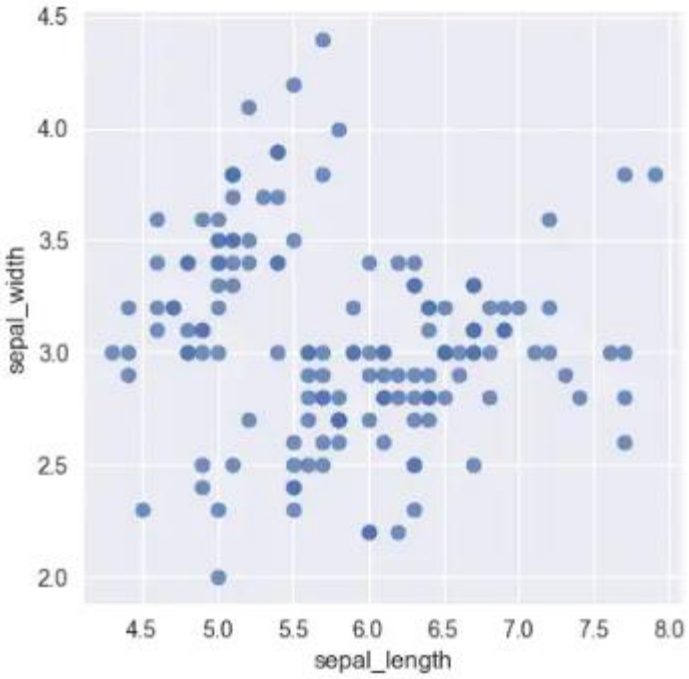
Another great plot to summarize the distribution of a variable is boxplots. Boxplots provide an intuitive and compelling way to spot the following elements:

- **Median**. The middle value of a dataset where 50% of the data is less than the median and 50% of the data is higher than the median.
- **The upper quartile**. The 75th percentile of a dataset where 75% of the data is less than the upper quartile, and 25% of the data is higher than the upper quartile.
- **The lower quartile**. The 25th percentile of a dataset where 25% of the data is less than the lower quartile and 75% is higher than the lower quartile.
- **The interquartile range**. The upper quartile minus the lower quartile
- **The upper adjacent value**. Or colloquially, the “maximum.” It represents the upper quartile plus 1.5 times the interquartile range.
- **The lower adjacent value**. Or colloquially, the “minimum." It represents the lower quartile minus 1.5 times the interquartile range.
- **Outliers**. Any values above the “maximum” or below the “minimum.”



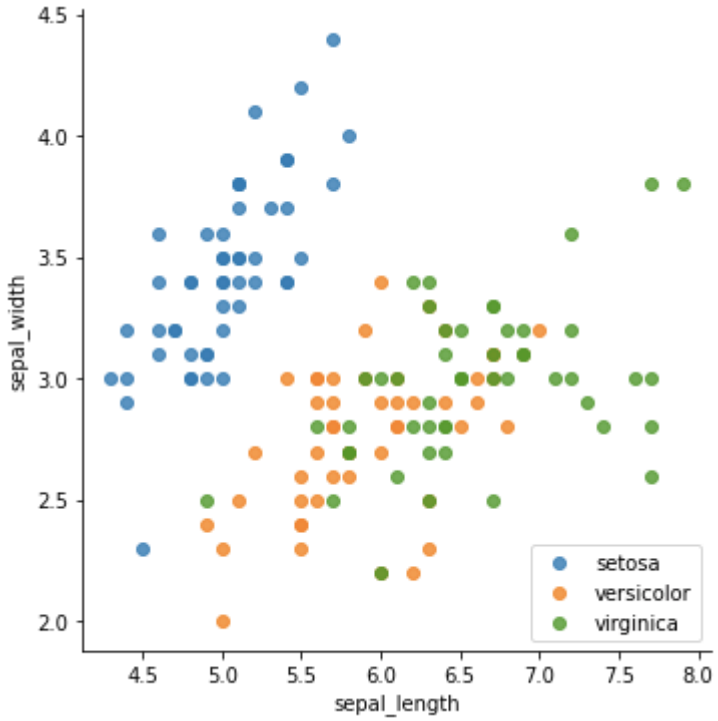
Scatter plots

Scatter plots are used to visualize the relationship between two continuous variables. Each point on the plot represents a single data point, and the position of the point on the x and y-axis represents the values of the two variables. It is often used in data exploration to understand the data and quickly surface potential correlations.



Bubble plot

Scatter plots can be easily augmented by adding new elements that represent new variables. For example, if we want to plot the relationship between sepal width and sepal length in the different varieties of iris, we could just add colors to the points, as following:



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

# Example: Histogram
plt.hist(data['column_name'], bins=20, color='blue', alpha=0.7)
plt.title('Distribution of Column')
plt.xlabel('Values')
plt.ylabel('Frequency')
plt.show()
```

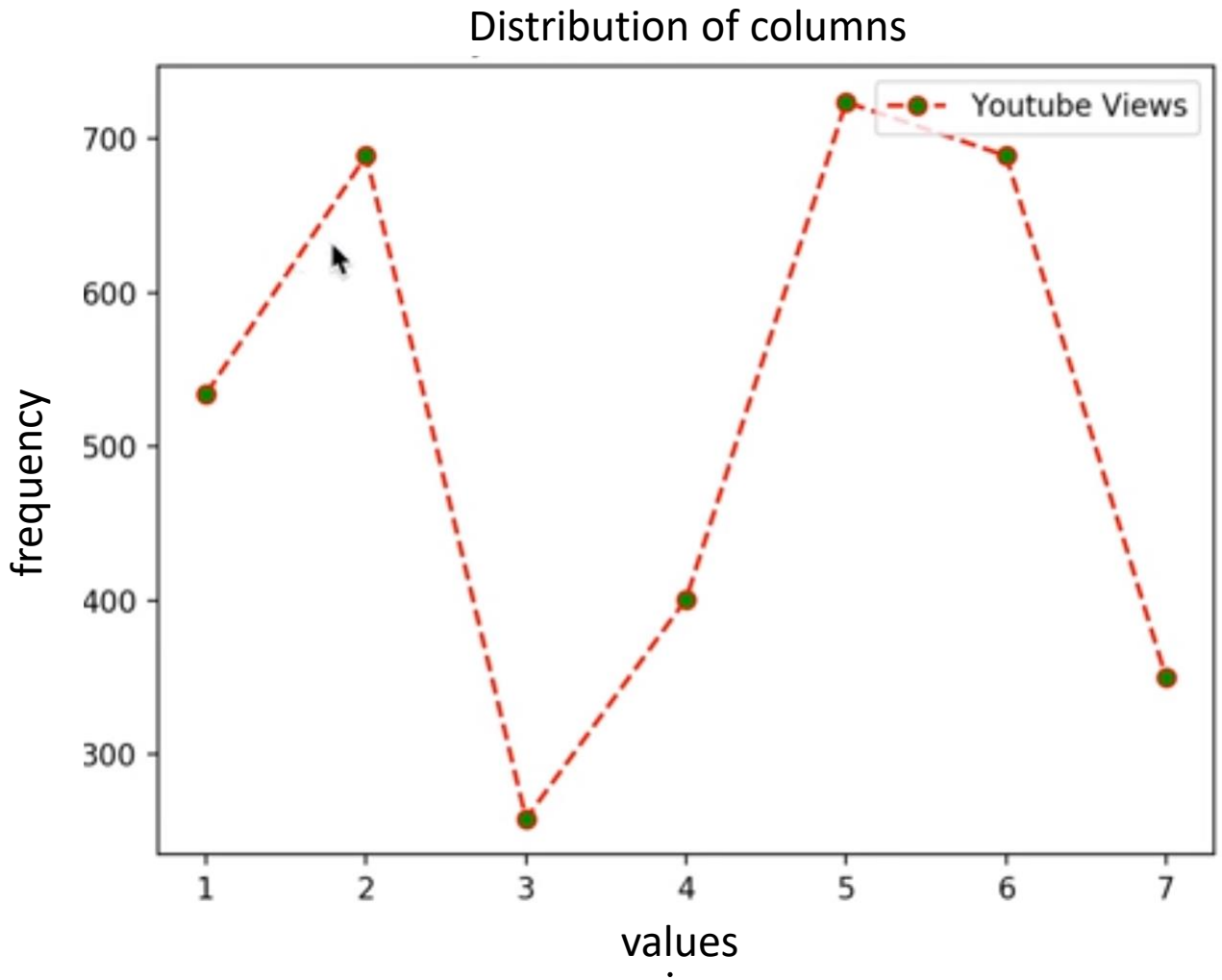
```python
from sklearn.metrics import roc_curve, auc

fpr, tpr, thresholds = roc_curve(y_true, y_scores)
roc_auc = auc(fpr, tpr)

plt.plot(fpr, tpr, color='darkorange', lw=2, label='ROC curve
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve')
plt.legend(loc='lower right')
plt.show()
```

```python
correlation_matrix = data.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix')
plt.show()
```

```python
feature_importance = model.feature_importances_
plt.bar(range(len(feature_importance)), feature_importance)
plt.xlabel('Feature Index')
plt.ylabel('Importance')
plt.title('Feature Importance')
plt.show()
```


Distribution of columns
Youtube Views
700
600
500
400
300
frequency
1
2
3
4
5
6
7
values