



Chapter 1: Visualizations

Type	Lecture
Materials	Chapter 1.pdf
Reviewed	<input checked="" type="checkbox"/>

Chapter 1: Visualizations

Here is a table showcasing the different plot types, along with a brief description and an example of the Seaborn (`sns`) syntax for each plot:

Note: Some plots may not have a specific syntax in Seaborn, as they may not be directly available or may require custom implementations using other libraries.

Plot Type	Description	Seaborn Syntax
Line Plot	Displays the relationship between two numeric variables over a continuous interval.	<code>sns.lineplot(x, y, data)</code>
Scatter Plot	Visualizes the relationship between two numeric variables using individual data points.	<code>sns.scatterplot(x, y, data)</code>
Box Plot	Represents the distribution of a numeric variable through quartiles and potential outliers.	<code>sns.boxplot(x, y, data)</code>
Point Plot	Shows the mean value of a numeric variable by different categories.	<code>sns.pointplot(x, y, data)</code>
Count Plot	Illustrates the count of occurrences of each category in a categorical variable.	<code>sns.countplot(x, data)</code>
Violin Plot	Combines a box plot with a KDE plot, providing information about the distribution and density of data.	<code>sns.violinplot(x, y, data)</code>
Swarm Plot	Displays individual data points along an axis, avoiding overlap and showcasing distribution.	<code>sns.swarmplot(x, y, data)</code>

Plot Type	Description	Seaborn Syntax
Bar Plot	Represents the mean value of a numeric variable for different categories using bars.	<code>sns.barplot(x, y, data)</code>
KDE Plot	Shows the distribution of a numeric variable using a smoothed kernel density estimation.	<code>sns.kdeplot(x, data)</code>
Heatmap	Displays a matrix-like plot where each cell is colored based on the value it represents.	<code>sns.heatmap(data)</code>
Pair Plot	Shows pairwise relationships between multiple variables in a dataset.	<code>sns.pairplot(data)</code>
Facet Grid	Allows for visualizing relationships between variables across multiple subplots.	<code>sns.FacetGrid(data)</code>
Joint Plot	Combines two different plots to showcase the relationship between two variables.	<code>sns.jointplot(x, y, data)</code>
Regression Plot	Fits and visualizes a linear regression model with confidence intervals.	<code>sns.regplot(x, y, data)</code>
Factor Plot	Displays the mean value of a numeric variable across different categorical variables.	<code>sns.factorplot(x, y, data)</code>
Clustermap	Visualizes hierarchical clustering of data in a dendrogram-like manner.	<code>sns.clustermap(data)</code>
Strip Plot	Shows individual data points along an axis, similar to a swarm plot.	<code>sns.stripplot(x, y, data)</code>
Correlation Heatmap	Displays the correlation between multiple variables as a heatmap.	<code>sns.heatmap(data, annot=True)</code>
Categorical Heatmap	Shows the relationship between two categorical variables as a heatmap.	<code>sns.heatmap(data, annot=True)</code>
Distribution Plot	Visualizes the distribution of a numeric variable using a histogram and a KDE.	<code>sns.displot(x, data)</code>
Residual Plot	Illustrates the difference between observed and predicted values from a regression model.	<code>sns.residplot(x, y, data)</code>
Residuals vs. Predicted Plot	Shows the relationship between residuals and predicted values in a regression model.	<code>sns.residplot(x, y, data)</code>
Time Series Plot	Visualizes data over time, often used for analyzing trends and patterns.	<code>sns.lineplot(x, y, data)</code>
Multi-Panel Categorical Plot	Displays multiple categorical plots in a grid-like structure.	<code>sns.catplot(x, y, data)</code>

Plot Type	Description	Seaborn Syntax
Line Swarm Plot	Combines a line plot with a swarm plot to showcase individual data points.	<code>sns.lineplot(x, y, data)</code>
Matrix Plot	Shows the pairwise relationships between multiple variables using a matrix of plots.	<code>sns.heatmap(data)</code>
Network Plot	Visualizes relationships between nodes in a network or graph.	
Circular Plot	Represents data in a circular manner, often used for visualizing cyclic patterns.	
Parallel Coordinates Plot	Shows the relationship between multiple variables by displaying them on parallel axes.	
Violin Swarm Plot	Combines a violin plot with a swarm plot to showcase both the distribution and individual data points.	

Quantiles and Box Plot

Here are some notes for students:

- Describing data numerically is essential, but as a data scientist, it's equally important to be able to visualize data effectively.
- In this video, we will explore various methods of visualizing data.
- We will consider a dataset related to advertising sales and specifically focus on the newspaper ad budget.
- If you have taken course two in this specialization, you might already be familiar with this dataset.
- Let's begin by examining a small sample of twelve data points from the newspaper ad budget.
- To find the median, we sort the data points in increasing order and split them into two halves with six samples each.
- The median is the average of the two middle values, which in this case is 27.8.
- We refer to the median as the 50% quantile or the second quartile.

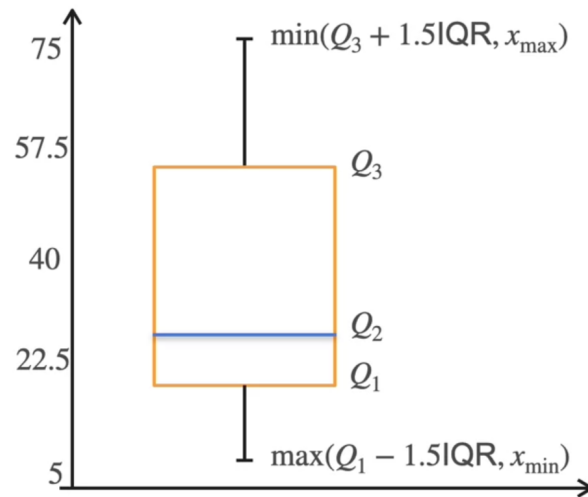
- Another useful metric is the value that leaves one quarter of the data to the left and three quarters to the right.
- This is known as the first quartile or Q_1 , and its value for our dataset is 18.35.
- Generally, the $k\%$ quantile is the value that leaves $k\%$ of the data to the left and $100 - k\%$ to the right.
- We can denote it as $q(k/100)$.
- Common quantiles include the 25% quantile (Q_1), the 50% quantile (median or Q_2), and the 75% quantile (Q_3).
- We can express the probability of the variable x being below the $k\%$ quantile as $N(\text{sub } k) / N$, where $N(\text{sub } k)$ represents the first $k\%$ of data.
- The probability of the variable x being below the $k\%$ quantile is denoted as $q(0.k)$.
- When considering a distribution with a probability density function (PDF), the area under the curve for the $k\%$ quantile should be $k/100$.
- Therefore, when computing quantiles from distributions, we can find the value such that the probability of the variable being below Q is $k/100$.

Visualizing data: Box-Plots

Box-Plots

What can you tell from this plot?

- Data is skewed
- No outliers (whiskers were cut at max and min value)
- Analyze dispersion



- The box plot, also known as the box and whiskers plot, is a visual representation that provides a standardized way to display the distribution of data using five statistics: minimum, maximum, median, first quartile, and third quartile.
- To create a box plot, you need to calculate the three quartiles: the 25% quantile (Q_1), the 50% quantile (median or Q_2), and the 75% quantile (Q_3).
- Divide the data into four blocks of equal size to determine the quartiles.
- The first quartile is the midpoint between the values of 18.3 and 18.4. The second quartile is the midpoint between 25.9 and 29.7, and the third quartile is the midpoint between 51.2 and 54.7.
- The interquartile range (IQR) is calculated as $Q_3 - Q_1$, representing where 50% of the data lies. In this case, the IQR is 34.6, the difference between the third and first quartiles.
- Identify the minimum and maximum values of the dataset, which are 8.7 and 75, respectively.
- To construct the box plot, start with a box that extends from the first quartile (18.35) to the third quartile (52.95).
- Draw a line at the value of the median (27.8 or Q_2).

- The whiskers are lines that extend from the ends of the box. One whisker goes from the box to $Q1 - 1.5 \text{ times the IQR}$, and the other whisker goes from the box to $Q3 + 1.5 \text{ times the IQR}$.
- Ensure that the whiskers do not go beyond the minimum and maximum values of the dataset. If they do, cut the whiskers at those values.
- The box plot provides valuable insights into the data at a glance. In this example, the skewness of the data is evident because $Q3 - Q2$ is significantly larger than $Q2 - Q1$.
- Outliers are identified as data points outside the whiskers. In this small sample, there are no outliers since the whiskers end at the minimum and maximum values.
- The length of the box and whiskers allows for analysis of data dispersion.
- To create a box plot using all the data, gather the values of $Q1$, $Q2$, $Q3$, IQR, minimum, and maximum.
- The obtained box plot using all the data reveals the values of $Q1$, $Q2$, and $Q3$.
- Note that the bottom whisker extends from $Q1$ to the smallest value of the dataset (0.3), while the upper whisker ends at 93.626, which is 1.5 times the IQR.
- Any data points outside the whiskers are considered outliers. In this case, two data points lie outside the whiskers, indicating outliers.

Visualizing data: Kernel Density Estimation

- When dealing with continuous random variables, the data distribution can be described using the probability density function (PDF).
- Histograms are commonly used to visualize data distributions, but they are not always the best approximation of a PDF for continuous variables.
- Kernel density estimation is a method used to approximate the PDF of a dataset based on the histogram representation.
- The goal of kernel density estimation is to create a smooth function that resembles the actual PDF underlying the data.
- The process of kernel density estimation involves the following steps:
 1. Plotting the observations on a graph.

2. Assigning each data point a small Gaussian curve, known as the kernel, centered on the data point.
 3. The standard deviation of the Gaussian curve determines the spread or width of the effect of each data point.
 4. Multiplying each Gaussian curve by a scaling factor of $1/n$, where n is the number of data points.
 5. Summing all the scaled Gaussian curves together.
- By summing the curves, the resulting plot provides an approximation of the PDF based on the data.
 - The kernel density estimation plot will have a smoother appearance compared to the histogram, resembling a PDF more closely.
 - The quality of the approximation depends on the number of data points used. With a larger dataset, the kernel density estimation will provide a better representation of the underlying PDF.
 - It's worth noting that while Gaussian curves are commonly used as kernels, other functions can be utilized depending on the specific requirements.
 - Kernel density estimation is a useful tool for understanding the shape and characteristics of a continuous data distribution when only a limited number of data points are available.

Violin plots

- Violin plots are a powerful visualization tool widely used in data science for representing the distribution of data.
- Violin plots combine the information from kernel density estimation (KDE) curves and box plots into a single visual representation.
- The main advantage of violin plots is that they provide a more comprehensive view of the data distribution compared to individual KDE curves or box plots.
- The structure of a violin plot consists of the following components:
 1. Kernel Density Estimation (KDE) Curve: A smoothed representation of the data distribution, similar to the kernel density estimation described earlier.

2. **Box Plot:** A traditional statistical summary of the data distribution, consisting of:
 - **Median:** The middle value of the dataset.
 - **Quartiles:** The three points that divide the dataset into four equal parts.
 - **Whiskers:** Lines extending from the box indicating the range of the data within a certain distance from the quartiles.
 - **Outliers:** Data points that fall outside the whiskers.
 3. **Mean:** The average value of the dataset, often represented by a point or a small line segment within the violin plot.
- Violin plots provide a holistic representation of the data, showing both the shape of the distribution through the KDE curve and the summary statistics through the box plot components.
 - They are especially useful when comparing multiple distributions side by side or when visualizing the distribution of a single variable.
 - Violin plots can effectively display the presence of multimodal distributions, skewness, and other distribution characteristics that may not be apparent in traditional plots like histograms or box plots.
 - The width of the violin at a particular point represents the density or frequency of data values at that point. Wider sections indicate higher density, while narrower sections indicate lower density.
 - Violin plots allow for easy visual comparisons between groups or categories by plotting multiple violins on the same graph, each corresponding to a different group or category.
 - At last, violin plots provide a compact and informative visualization that combines the advantages of KDE curves and box plots, making them a valuable tool in data analysis and visualization.

QQ Plots

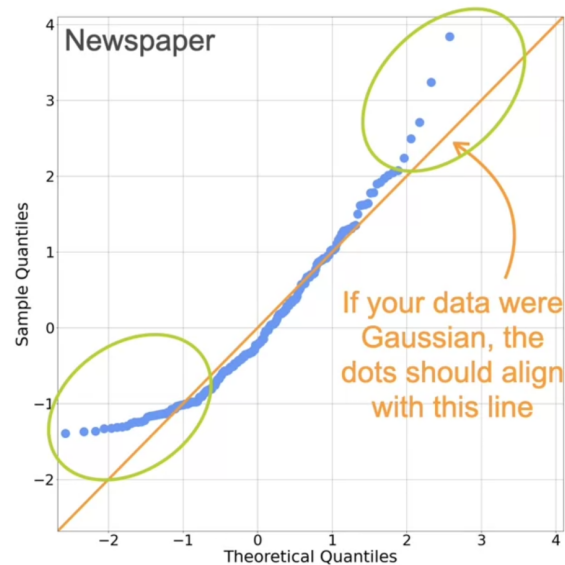
QQ Plots

Quantile-Quantile plots (QQ Plots) compare quantiles

- Standardize your data:

$$\left(\frac{x - \mu}{\sigma} \right)$$

- Compute quantiles
- Compare to gaussian quantiles



- Normality is often assessed in data science to determine if variables follow a Gaussian distribution, which is assumed by various models and tests.
- Quick visual inspections, such as examining histograms, can provide an initial indication of whether the data appears Gaussian or not.
- Histograms with a bell-shaped curve are indicative of a potential Gaussian distribution, while those deviating from a bell shape suggest departures from normality.
- To obtain a more precise assessment of normality, quantile-quantile plots (QQ plots) are commonly used.
- QQ plots offer a graphical comparison between the quantiles of the observed data and the quantiles expected from a normal distribution.
- The process of creating a QQ plot involves standardizing the data by subtracting the mean and dividing by the standard deviation.
- The quantiles of the standardized data are then compared to the quantiles of a standard normal distribution.
- A scatter plot is constructed with the x-axis representing the theoretical quantiles and the y-axis representing the sample quantiles.

- Each point in the scatter plot corresponds to a specific quantile.
- In a QQ plot for normally distributed data, the points should be closely aligned with a diagonal line (often orange) that represents perfect normality.
- Deviations from the diagonal line indicate departures from normality.
- Skewness in the data can be observed as a concentration of points on one side of the diagonal line.
- Two marked areas on the scatter plot where the points significantly deviate from the diagonal line suggest non-normality in the data.
- When examining the sales column in the data set, the histogram displays a bell-shaped curve, indicating potential normality.
- The QQ plot for this data aligns the quantiles around the diagonal line, providing further evidence of a Gaussian distribution.
- Assessing normality is important as it can impact the validity and accuracy of models and tests that assume normality in the data.
- QQ plots serve as a valuable tool for visually inspecting normality and making informed decisions regarding data analysis and modeling.

DOUBT



My question is why data needs to be normally distributed, isn't that bar graph plotted using real data points and if we try to make gaussian, wont it change or milead datapoints?



Excellent question! The requirement for data to be normally distributed depends on the specific analysis or model being used. While some models and tests assume normality, not all data needs to follow a normal distribution.



It's important to note that the goal is not to make the data points fit a Gaussian distribution, but rather to assess whether the data reasonably approximates a normal distribution. This assessment helps determine the applicability of certain statistical techniques that assume normality.

Here are a few reasons why normality is desirable or assumed in some cases:

1. Assumptions of statistical tests: Many statistical tests, such as t-tests or ANOVA, assume that the data follows a normal distribution. Violating this assumption may affect the validity and reliability of the test results. However, it's worth mentioning that some statistical tests are robust to deviations from normality, or alternative non-parametric tests can be used.
2. Model assumptions: Some models, like linear regression or Gaussian Naïve Bayes, assume that the underlying data follows a normal distribution. Deviations from normality may impact the model's assumptions and affect the accuracy of predictions or inferences.
3. Inference and confidence intervals: When data is normally distributed, it allows for more straightforward interpretation of results and calculation of confidence intervals. Normality facilitates the use of parametric methods to estimate population parameters accurately.

However, it's important to emphasize that not all analyses or models require normality. There are alternative approaches available for non-normal data, including non-parametric tests or transformations to approximate normality.

Regarding your concern about misleading the data points, it's crucial to understand that assessing normality does not involve altering the actual data points. The evaluation of normality is done based on the observed data distribution and comparing it to the characteristics of a normal distribution. It does not involve modifying the original data points to fit a Gaussian shape.

In summary, the assumption of normality is context-specific, and not all analyses or models require normally distributed data. Assessing normality helps determine the appropriate statistical techniques and ensure the validity of the results, but it does not involve manipulating the actual data points.