KANAV BANSAL

PLOTTING WITH SEABORN:

SEABORN: It is a python data visualization library based on matplotlib.

↳ It provides a high-level interface for drawing attractive and informative statistical graphics.

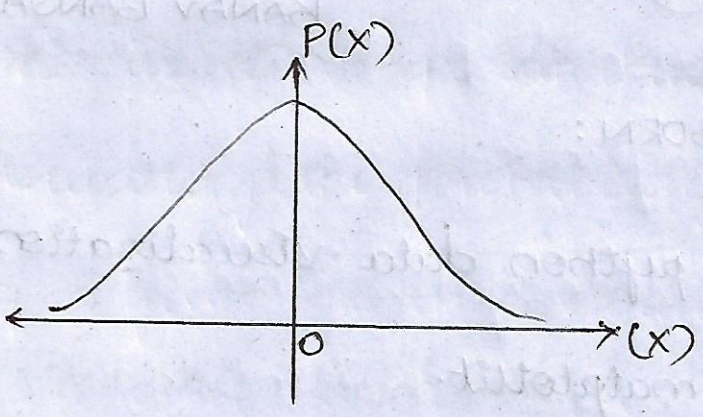
1. UNIVARIATE ANALYSIS (FOR NUMERICAL VARIABLES)

i.e., Analyzing one variable at a time.

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

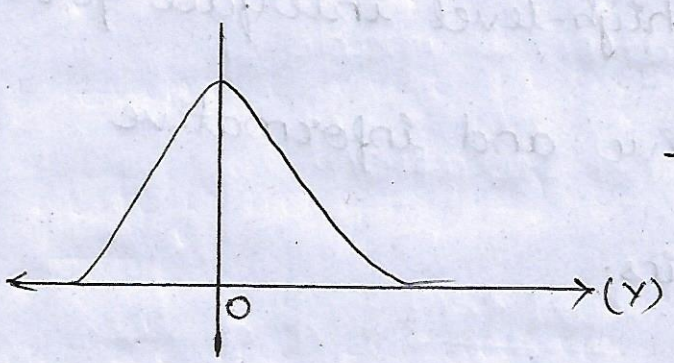
↳ importing seaborn from the library.

EXAMPLE OF A NORMAL DISTRIBUTION:



$\rightarrow \mu_x = 0$

$\sigma_x = 1$



$\rightarrow \mu_y = 1$

$\sigma_y = 1$

The above graphs are symmetric.

I/p - sns.distplot(df['Y'])

by default sns.distplot(x, hist=True, kde=True

HISTOGRAM:

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

↳ This allows the inspection of the data for its underlying distribution (normal distribution), outliers, skewness ... etc.

NOTE:

* IF BIN WIDTH IS TOO SMALL:

It shows too much individual data and does not allow the underlying pattern (frequency distribution) of the data.

* IF BIN WIDTH IS TOO LARGE:

We are unable to find the underlying (pattern) trend in the data.

I/p - sns.distplot(df['Y'], kde=False, rug=True)

This is basically a frequency plot.

Try if rug = False. (hist = True is taken automatically as a default)

KDE : KERNEL DENSITY ESTIMATION : (24)

This is a non-parametric method for estimating the probability density function of a given random variable.

↳ In general, they are generalisation and improvement over histograms (smoothening of histogram graphs)

↳ Here we all extract the important features of the data.

rug : bool, optional (whether to draw a rugplot on the support axis)

A rug plot is a plot of data for a single quantitative variable displayed as marks along an axis.

↳ As such it is an analogous to a histogram with zero-width bins, or a one-dimensional scatter plot.

I/p - sns.distplot(df['y'], hist=False, rug=True)

or

I/p - sns.kdeplot

UNDERSMOOTHED - Bandwidth is too small

OVERSMOOTHED - Bandwidth is too large

2. BIVARIATE ANALYSIS (FOR NUMERICAL VARIABLES)

I/p - df.describe( )

$\mu_x = 0$, $\mu_y = 0$

$\sigma_x = 1$, $\sigma_y = 1$

Since the mean and median are nearly equal to '0' and '1', then there are no outliers.

PROPERTIES OF KDE:

↳ Smooth

↳ No end points

↳ depends on bandwidth

PROPERTIES OF HISTOGRAM:

↳ Not smooth

↳ depends on endpoints of bins

↳ depends on width of bins

UNIVARIATE (Numerical):

- Histogram
- KDE
- Box

BIVARIATE (Numerical, variables):

- Scatter
- Hexbin
- Pair plot.

SCATTER PLOT:

These are similar ~~plot~~ to line graphs in that they use horizontal and vertical axes to plot data points.

↳ This shows how much one variable is affected by another. This relationship between two variables is called their correlation.

↳ Scatter plots usually consist of a large body of data.

↳ The closer the data points come when plotted to making a straight line.

↳ The higher the correlation between the two variables or the stronger the relationship.

↳ If the data points make a straight line going from the origin out to high x- and y-values, then the variables are said to have a positive correlation.

↳ If the line goes ~~from~~ from a high-value on the y-axis down to a high-value on the x-axis, then the variables have a negative correlation.

I/P - sns.jointplot(x='x', y='y', data=df, kind='scatter')

or

sns.jointplot(data=df, x='x', y='y', kind='scatter')

HEXBIN PLOTS: TO KNOW THE DENSITY

A Hexbin plot is useful to represent the relationship of 2 numerical variables when we have a lot of data point.

↳ Instead of overlapping, the plotting window is split in several hexbins, and the number of points per hexbin is counted.

↳ The color denotes, ~~this~~ the number of points.

NOTE:

* IF LIGHTER : less Density.
* IF DARKER : High Density - most of the points lie in this region.

```
i/p - sns.jointplot(x='x', y='y', data=df, kind='hex'
                    color='k')
```

or

```
sns.jointplot(data=df, x='x', y='y', kind='hex', color='k')
```

PAIR PLOT:

A pair plot allows us to see both distribution of single variables and relationships between two variables.

↳ It is a great method to identify trends for follow-up analysis and fortunately are easily implemented.

I/p - iris = pd.read_csv('data/Iris.csv')

sns.pairplot(iris)

O/p - shows all the possible pairs of plots.

NOTE: Except species column, the rest columns plots are drawn.

I/p - sns.pairplot(iris, hue='species')

hue - COLOR CODE and works with categorical data.

WORKING WITH CATEGORICAL DATA:

I/p - tips = pd.read_csv('data/tips.csv')

tips.head()

I/p - tips.shape() → O/p - (244, 7)

I/p - tips.describe(include = 'all')

I/p - tips['day'].value_counts()

O/p - sat 87
sun 76
Thur 62
Fri 19
Name: day, dtype: int64

STRIPPLOT:

A strip plot is a scatter plot where one of the variables is categorical.

For example,

A box plot with an overlaid strip plot becomes more similar to a violin plot because some additional information about

how the underlying data is distributed becomes visible.

I/P - sns.stripplot(data = tips, x = 'data', y = 'total_bill')

SWARMPLOT :

This function is similar to stripplot(), but the points are adjusted (only along the categorical axis) so that they don't overlap.

↳ It gives a better representation of the distribution of values, but it does not scale well to large numbers of observations.

↳ This style of plot is sometimes called as a "BEESWARM".

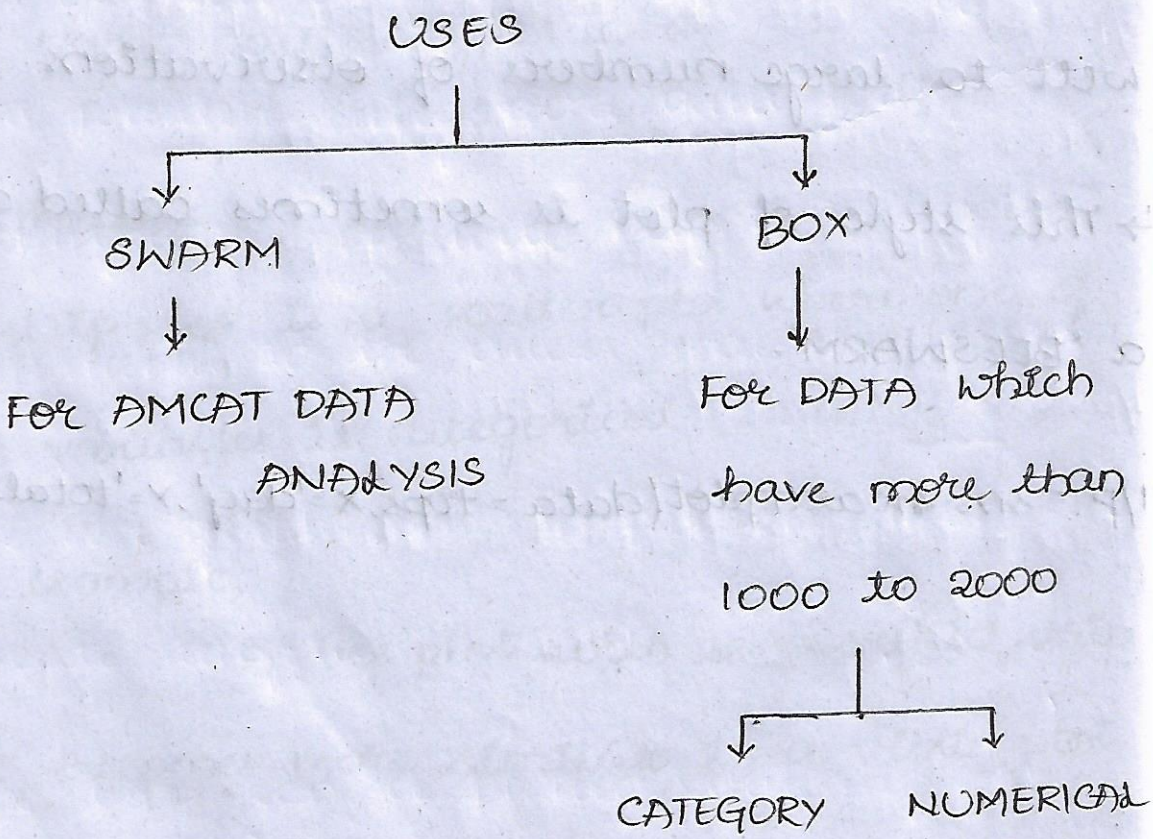
I/P - sns.swarmplot(data = tips, x = 'day', y = 'total_bill')

BOXPLOT:

A box plot is a method for graphically depicting groups of numerical data through their quartiles.

↳ The box extends from the Q1 to Q3 quartile values of the data, with a line at the median (Q2).

↳ The whiskers extend from the edges of box to show the range of the data.



I/p - sns.boxplot(data = tips, x = 'day', y = 'total_bill')

I/p - tips.loc[tips.time == 'Lunch'] :

In SQL : select * (all) from tips where time == lunch.

I/p - tips.loc[tips.time == 'Lunch', 'day'].value_counts()

O/p - Thur 61
Fri 7
Name : day, dtype : int64

In SQL : select * day from tips where time == lunch.

I/p - sns.boxplot(data = tips, x = 'day', y = 'total_bill', hue = 'time')

Adding a new column, indicating weekend or not.

I/p - tips['weekend'] = tips['day'].isin(['sat', 'sun'])

tips.head()

O/P -

	total_bill	tip	Sex	Smoker	day	time	Size	Week-end
0	16.99	1.01	F	NO	Sun	Dinner	2	True
1	10.34	1.66	M	No	Sun	Dinner	3	True
2	21.01	3.50	M	No	Sun	Dinner	3	True
3	23.68	3.31	M	No	Sun	Dinner	2	True
4	24.59	3.61	F	No	Sun	Dinner	4	True

I/P - sns.boxplot(data = tips, x = 'day', y = 'total_bill', hue = 'weekend')

BAR PLOT :

A bar chart or graph is a chart or graph that presents categorical data with rectangular bars with heights or lengths proportional to the values that they represent.

↳ The bar graph shows comparisons among the discrete categories.

↳ The bars can be plotted vertically (or) horizontally.

I/p - titanic = pd.read_csv('data/titanic.csv')

titanic.head()

I/p - titanic.shape

O/p - (891, 15)

I/p - titanic.describe(include = 'all')

I/p - sns.barplot(x = 'sex', y = 'age', data = titanic)

Barplot → Estimator - mean (by default)

↳ Estimator - median

I/p - plt.figure(figsize = (12,6))

plt.subplot(1, 2, 1)

sns.barplot(x='sex', y='age', data = titanic)

plt.title('Avg Age')

plt.subplot(1,2, 2)

sns.barplot(x='sex', y='age', data = titanic,
estimator=np.median)

plt.title('Median Age')

plt.show()

I/p - sns.boxplot(x='sex', y='age', data=titanic)

I/p - sns.barplot(x='sex', y='survived', data=titanic)

I/p - sns.barplot(x='sex', y='survived', hue='class',
data=titanic)

COUNT PLOT:

A count plot can be thought of as a histogram across a categorical, instead of quantitative, variable.

↳ The basic API and options are identical to those for barplot(), so you can compare counts across nested variables.

I/p - sns.countplot(x='class', data=titanic,
palette='Greens_d')

PALETTE - COLOR REQUIREMENT

UNIVARIATE

- CATEGORY
  - COUNT - (VALUE COUNT)
  - ~~PARTITION~~
- NAME
  - HISTOGRAM
  - KDE
  - BOX PLOT

BIVARIATE

- NUMERICAL, NUMERICAL
  - SCATTER
  - PAIR
  - HEXBIN
- CATEGORY, NUMERICAL
  - STRIP PLOT
  - SWARM PLOT
  - BOX PLOT
  - BAR PLOT (MEAN & MEDIAN)