DATA VISUALIZATION

Data visualization transforms raw data into visual formats like charts, graphs, and maps, enabling users to quickly identify trends & patterns.

The following methods are used to represent data visualization

- Distplot
- Jointplot
- Pairplot
- Stripplot
- Swarmplot
- ViolinePlot

Distplot

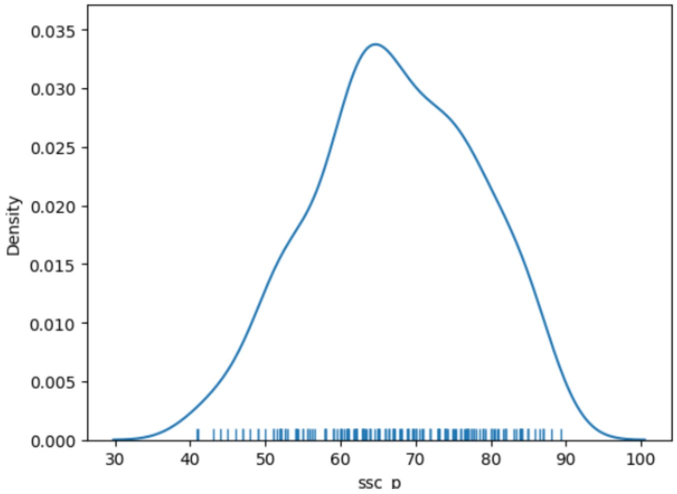
It tells overall distribution of continuous data variable(Numerical value).It has following parameter

Kde=True means probability of distribution of continuous values.

rug=True in distplot, Seaborne adds small vertical lines (or ticks) along the x-axis, with each line representing the location of a data point.

hist=True shows the frequency distribution of data.

```
sb.distplot(dataset['ssc_p'],hist=False,kde=True,rug=True)
```

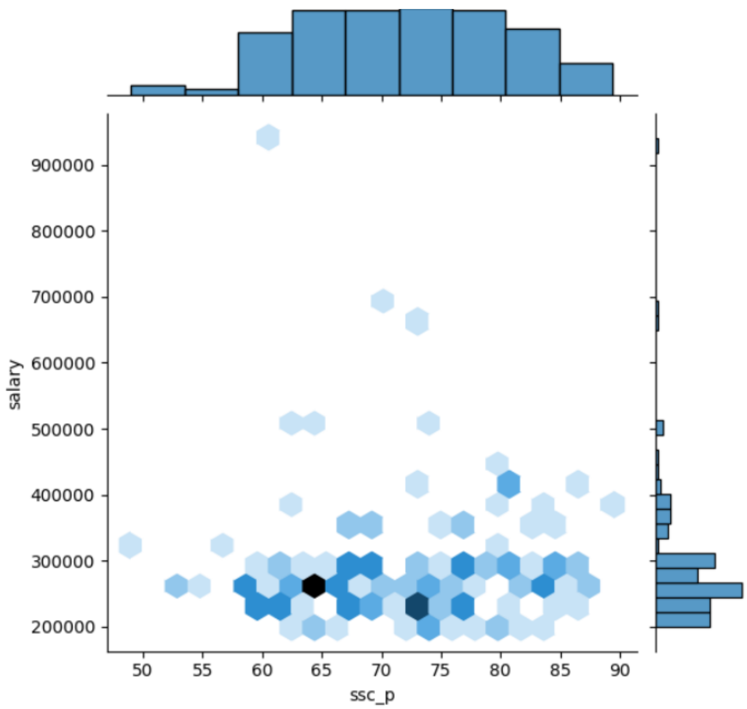


Jointplot

It combines two different types of plot(**Scatter plot,Histogram**) to visualize the relationship between two variables.

```
sb.jointplot(x="ssc_p",y="salary",data=dataset,kind="hex")
plt.show()
```

In the following fig the dark color hexagon represent the repeation.In this graph more students had 65 percentage of mark, that student got nearly 300000 salary.



Pairplot

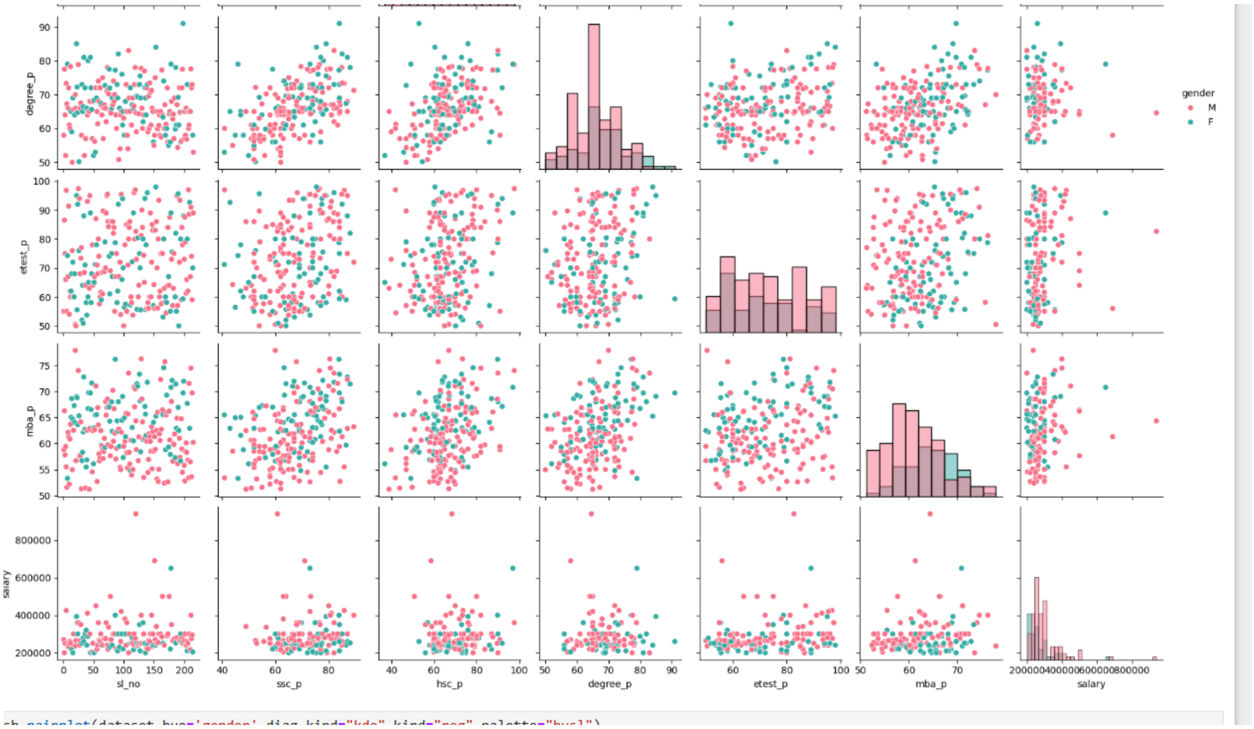
A pairplot displays pair wise relationship of all variable in a dataset. It creates a matrix of plots where each cell shows the relationship between two variables.

Hue means all variable is differentiate based on what values the hue has.

diag_kind shows the distribution of each individual variable.

Kind shows the relationship between pair of variable.

```
sb.pairplot(dataset,hue='gender',diag_kind="hist",kind="scatter",palette
="husl")
```



```
sb.pairplot(dataset,hue='gender',diag_kind="kde",kind="reg",palette="husl")
```

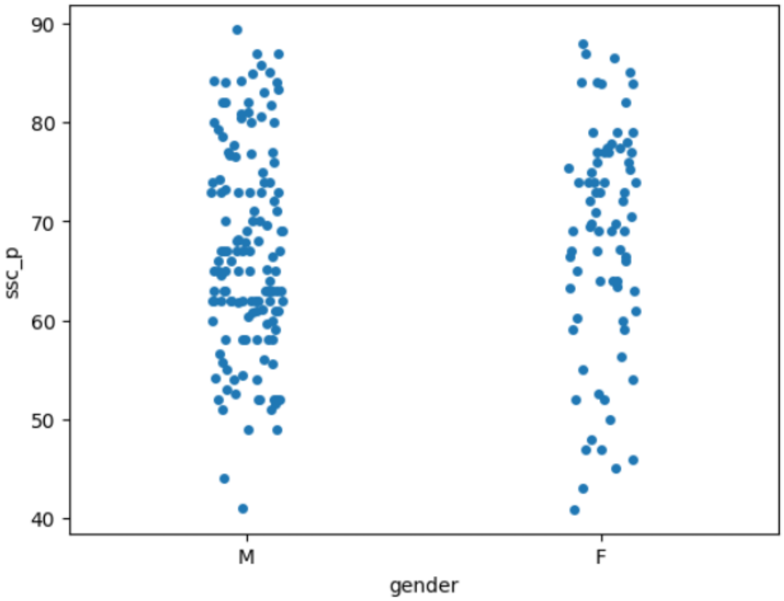
Stripplot

It is a simple scatter plot that visualizes the distribution of data along a single axis moreover overlapping is common. Quickly seeing Central Tendency of single variable. Here x axis is a categorical data &

y axis is quantative data.If we avoid overlapping we should put Jitter=True.

sb.stripplot(x="gender",y="ssc_p",data=dataset,jitter=True)

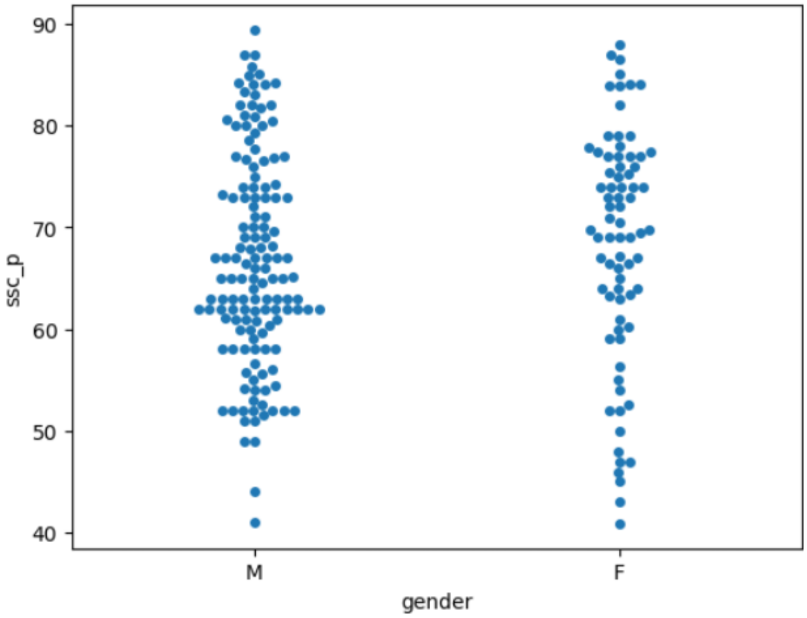
Using the graph we can find out which gender (M &F) scored highest mark, lowest mark & average mark.



Swarmplot

It is similar to stripplot but it automatically adjusts the positions of point avoid overlap even without jittering.

sb.swarmplot(x="gender",y="ssc_p",data=dataset)

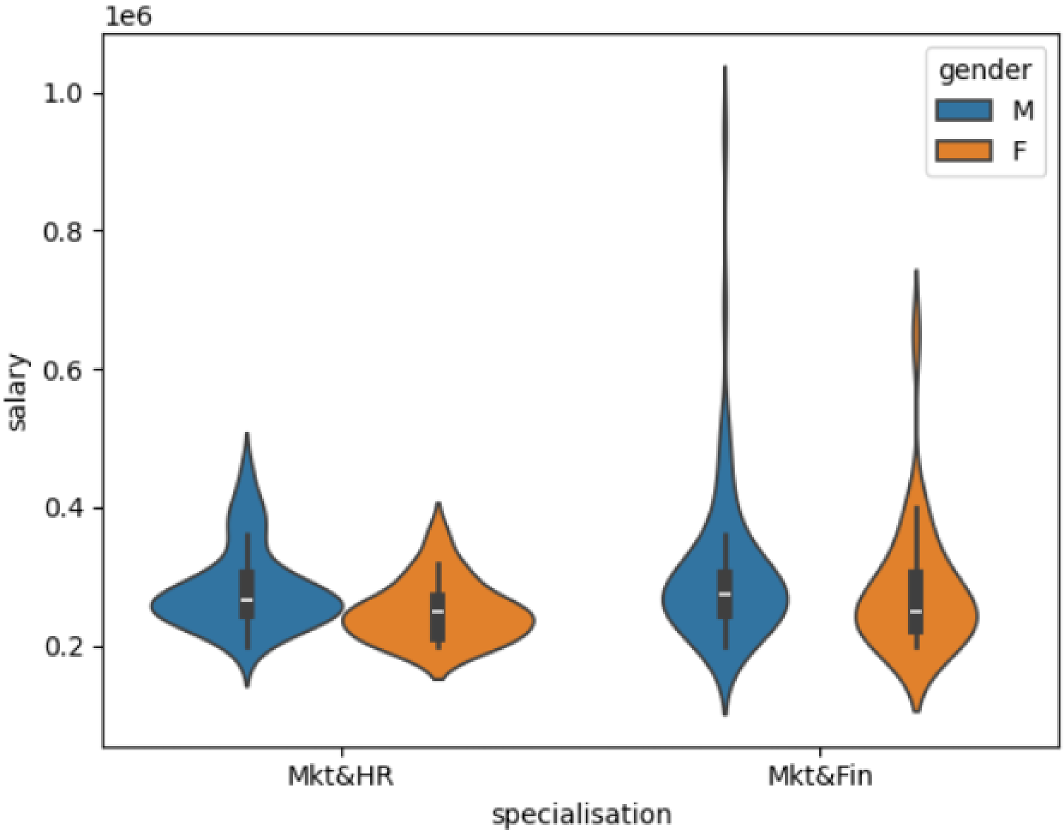


Violineplot

A violin plot is a statistical representation of numerical data that combines elements of a box plot and a kernel density plot, visualizing the distribution of data across different groups.

```
sb.violinplot(x="specialisation",y="salary",data=dataset,
hue="gender")
```

Here x axis is a categorical data & y axis is quantative data.



The above fig shows the salary information of male and female in (Marketing & HR) and (Marketing & Finance)