SEABORN

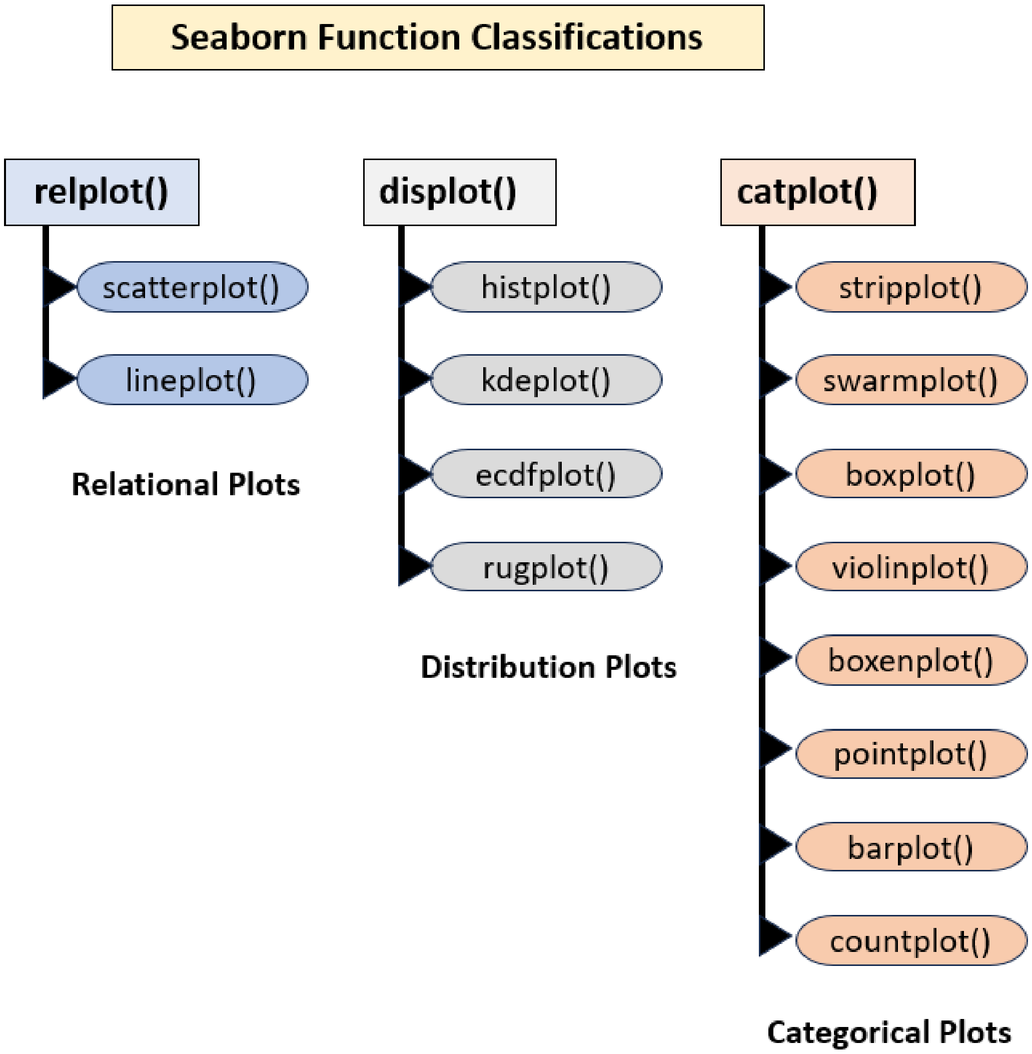
DEFINITION:

- Seaborn is an amazing visualization library for statistical graphics plotting in Python.
- It provides beautiful default styles and colour palettes to make statistical plots more attractive.
- It is built on top matplotlib library and is also closely integrated with the data structures from pandas.
- Seaborn aims to make visualization the central part of exploring and understanding data.
- It provides dataset-oriented APIs so that we can switch between different visual representations for the same variables for a better understanding of the dataset.

CATEGORIES OF PLOT IN SEABORN:

Plots are basically used for visualizing the relationship between variables. Those variables can be either completely numerical or a category like a group, class, or division. Seaborn divides the plot into the below categories:

1. **Relational plots:** This plot is used to understand the relation between two variables.
2. **Categorical plots:** This plot deals with categorical variables and how they can be visualized.
3. **Distribution plots:** This plot is used for examining univariate and bivariate distributions
4. **Regression plots:** The regression plots in Seaborn are primarily intended to add a visual guide that helps to emphasize patterns in a dataset during exploratory data analyses.
5. **Matrix plots:** A matrix plot is an array of scatterplots.
6. **Multi-plot grids:** It is a useful approach to draw multiple instances of the same plot on different subsets of the dataset.



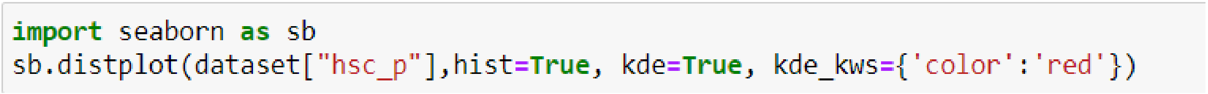
UNIVARIATE ANALYSIS

DISTPLOT:

This function provides access to several approaches for visualizing the univariate or bivariate distribution of data, including subsets of data defined by semantic mapping and faceting across multiple subplots. The kind parameter selects the approach to use:

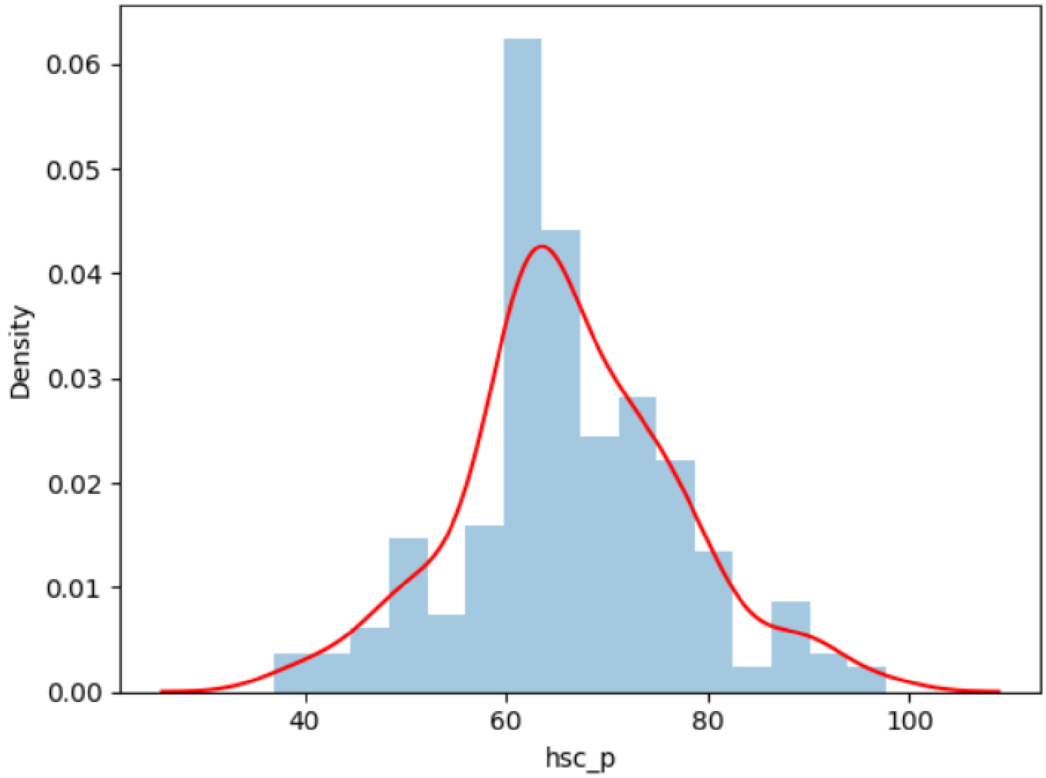
- histplot() (with kind="hist"; the default)
- kdeplot() (with kind="kde")
- ecdfplot() (with kind="ecdf"; univariate-only)

CODE:



```
import seaborn as sb
sb.distplot(dataset["hsc_p"],hist=True, kde=True, kde_kws={'color':'red'})
```

IMAGE:



EXPLANATION:

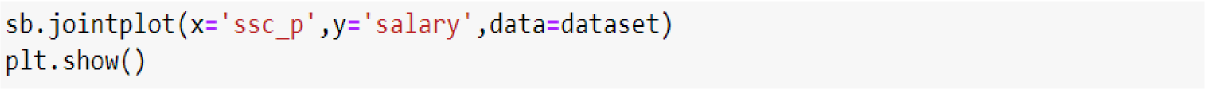
- The above graph is plotted for the sample dataset ‘Placement.csv’.
- The graph of hsc marks is plotted in histogram (hist =true) with respect to density because kde = True is given, by this kernel density estimation.
- By passing the parameter kde_kws we are specifying that kde plot should be present in red colour.

BIVARIATE ANALYSIS

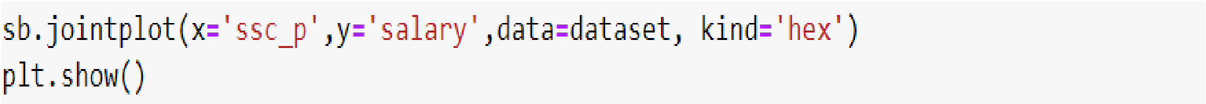
JOINTPLOT:

- A Joint plot is very helpful to plot two variables. On the one hand, the individual distribution of each variable is visualized. On the other hand, the relationship between the two variables is visualized.
- It provides a convenient interface to the JointGrid class, with several canned plot kinds.

CODE:



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



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

IMAGE 1:

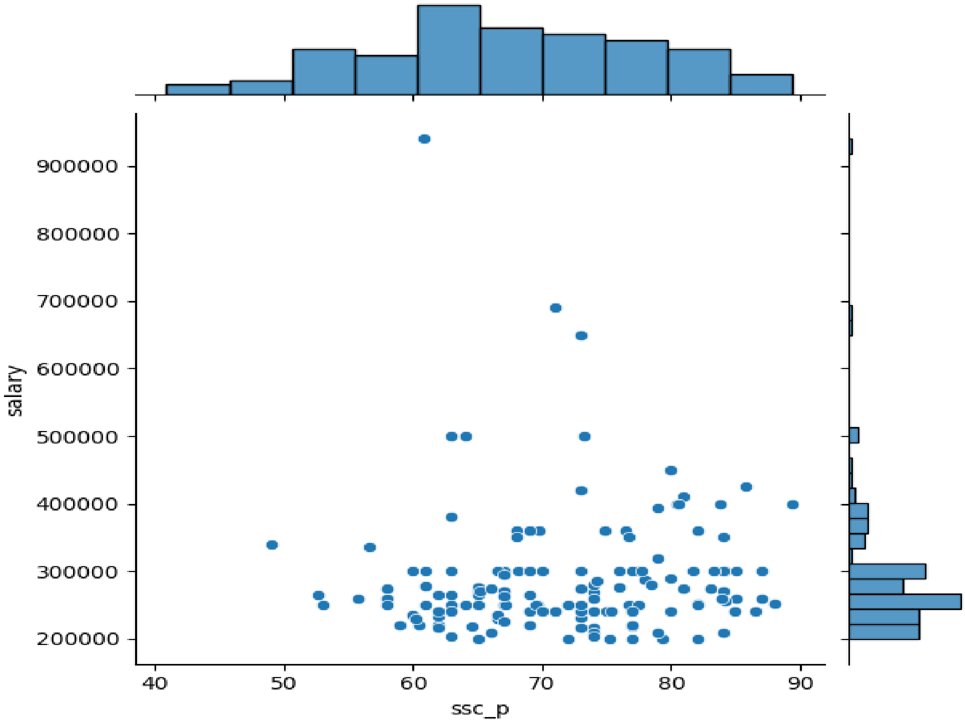
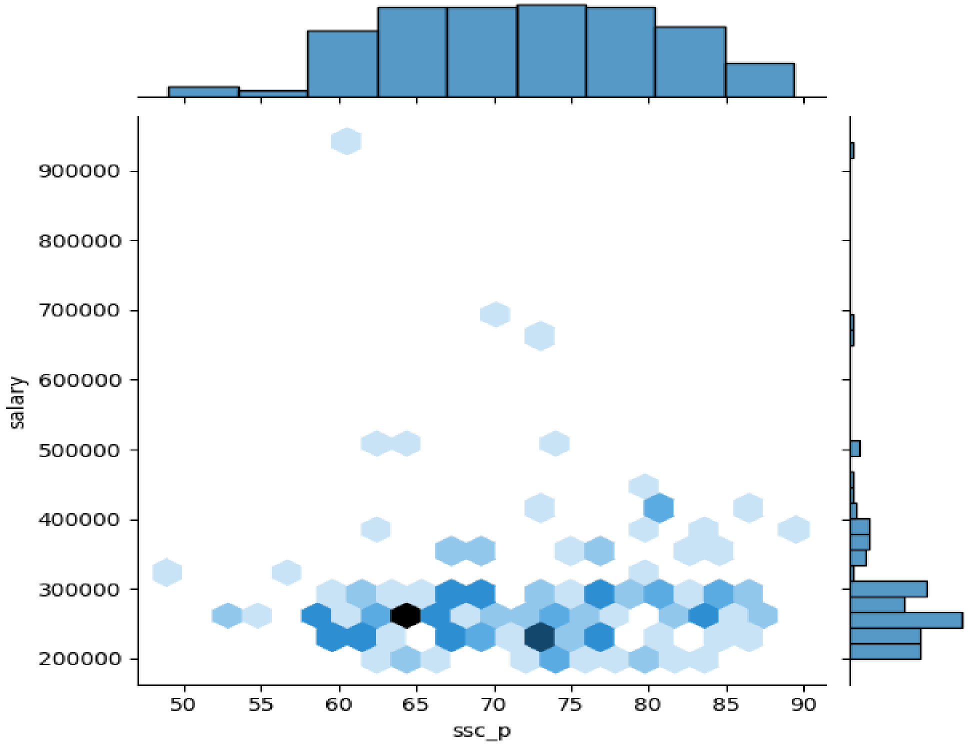


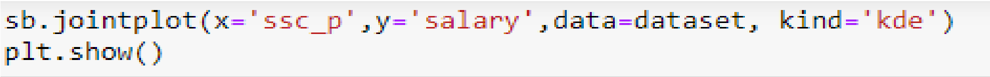
IMAGE 2:



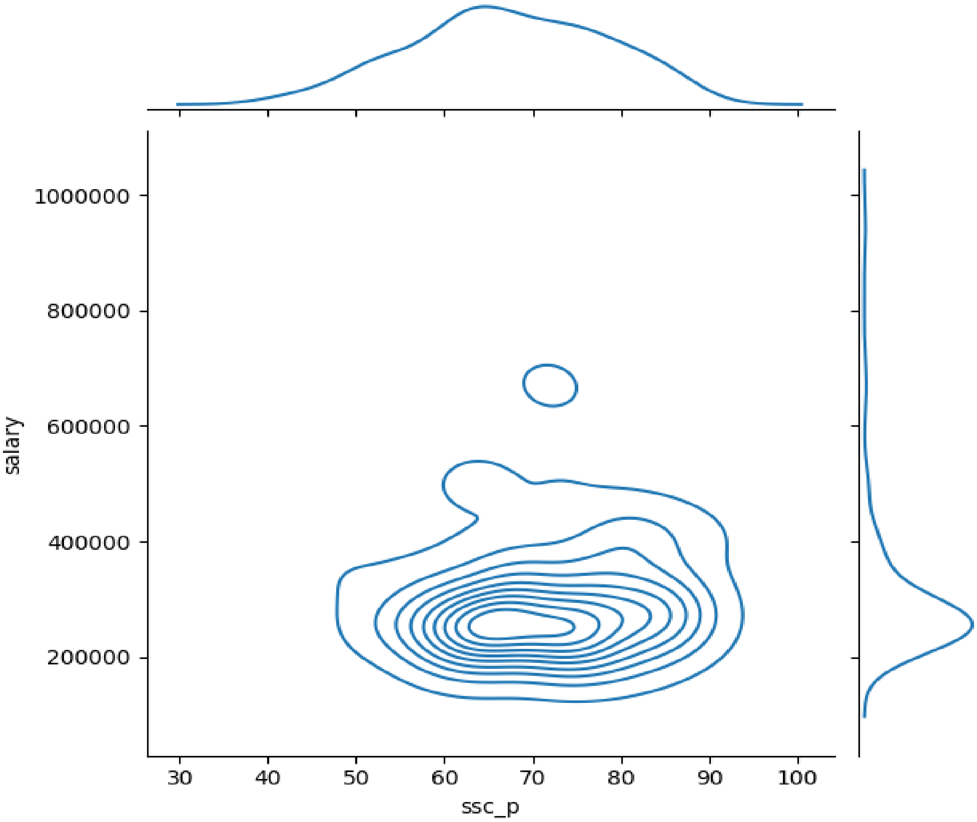
EXPLANATION:

- In the above jointplot image 1 graph, we can easily see both scatter plot and histogram side by side
- The graph is plotted between hsc_p marks and salary of students, we can easily analysis the salary according to students marks here.
- Here in the above image 2 graph, we have changed the graph type to hex by giving kind= 'hex'.
- Hex having a concept of heatmap, heatmap is used to visualize the datas easily, when data comes to over repetition, it is represented by dark shades; we can understand that data is not repeated, when it is in light shade.
- The colour shade increases or decreases according to the repetition of data.
- Both histogram and heatmap is plotted by using jointplot.

IMAGE 3:



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

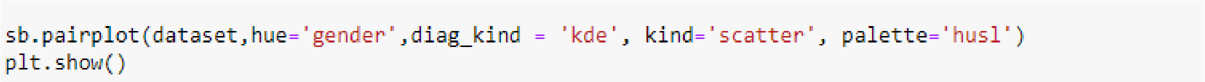


- The graph is plotted with kernel density estimation.

PAIRPLOT:

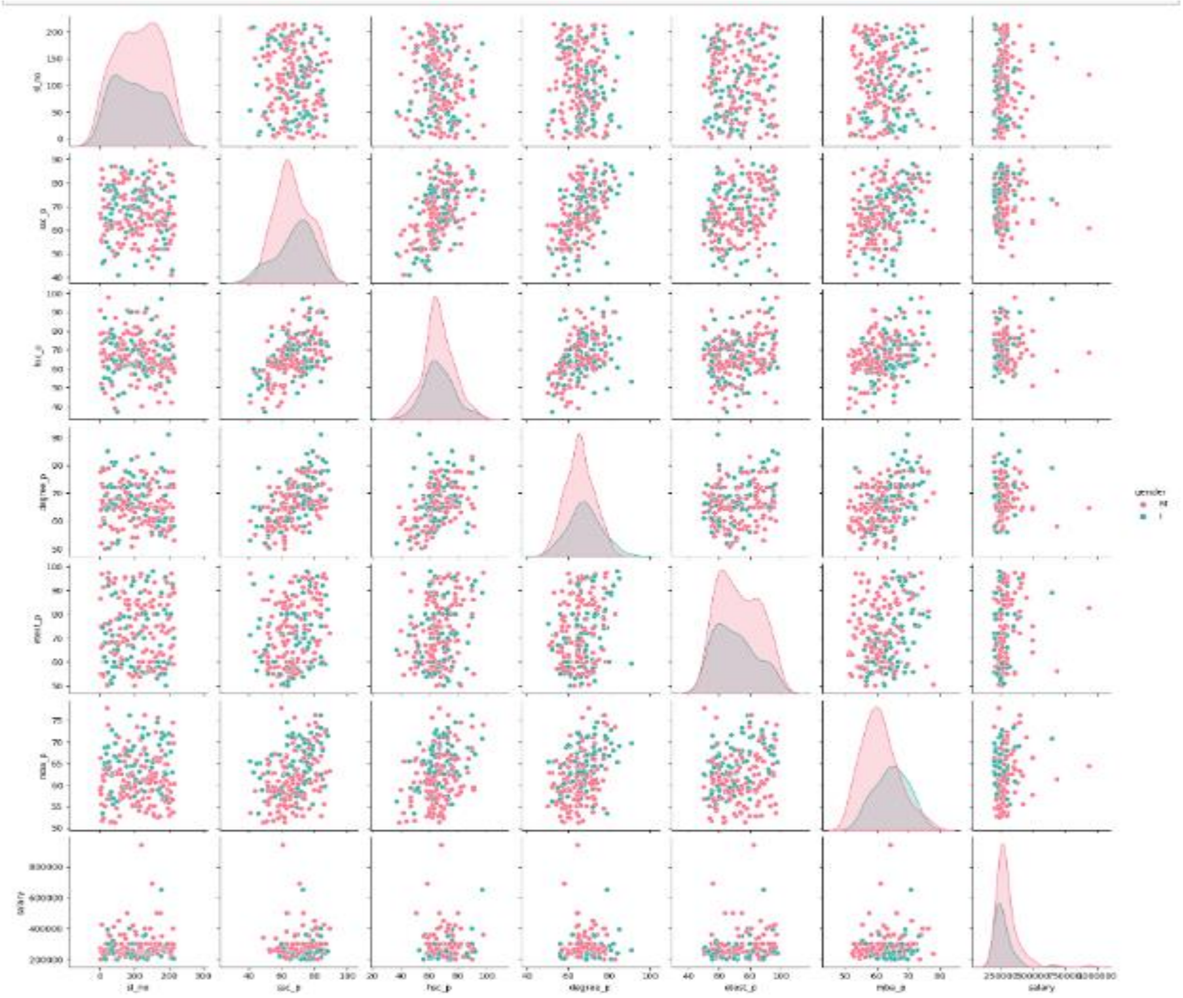
- Pairplot visualizes given data to find the relationship between them where the variables can be continuous or categorical.
- Plot pairwise relationships in a data-set.
- Pairplot is a module of seaborn library which provides a high-level interface for drawing attractive and informative statistical graphics.

CODE:



```
sb.pairplot(dataset,hue='gender',diag_kind = 'kde', kind='scatter', palette='husl')
plt.show()
```

IMAGE:



EXPLANATION:

- By default, this function will create a grid of Axes such that each numeric variable in data will by shared across the y-axes across a single row and the x-axes across a single column.

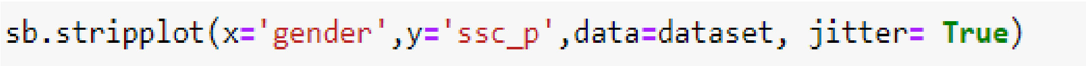
- The diagonal plots are treated differently: a univariate distribution plot is drawn to show the marginal distribution of the data in each column.
- Diagonal plots are differed by denoting diag_kind='kde', we can prefer other method instead of kde.
- It is also possible to show a subset of variables or plot different variables on the rows and columns.
- Pairplot is plotted for the whole dataset 'Placement.csv' in the above graph.

CATEGORICAL PLOT

STRIP PLOT:

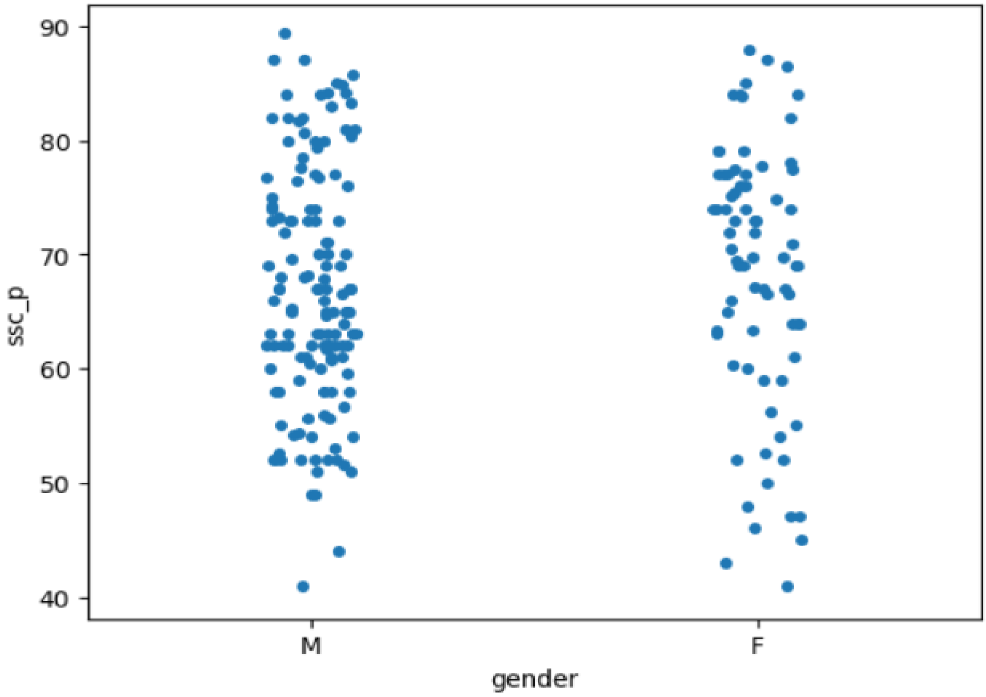
- A strip plot is created entirely on its own. In circumstances when all data are given together with some representation of the underlying distribution, it is a nice complement to a boxplot or violinplot.
- It is used to generate a scatter plot depending on a category.
- In strip plot, we can plot graph between quantitative and qualitative datas.

CODE:



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

IMAGE:



EXPLANATION:

- Strip plot is used to do analysis between quantitative and qualitative data.
- Here in this graph, we can easily identify the ssc mark range of students segregated male and female by using strip plot.
- By analysing this graph, male students performed well in ssc exam than female students because male students mark range is 90 but in female it ended with below 90.

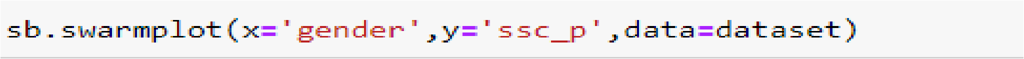
SWARMPLOT:

- This function is similar to stripplot(), but the points are adjusted (only along the categorical axis) so that they don't overlap.
- This gives a better representation of the distribution of values, but it does not scale well to large numbers of observations.
- Seaborn.swarmplot () method is used to draw a non-overlapping scatter plot where one of the variables is a categorical variable.
- A swarm plot can be drawn on its own, but it is also a good complement to a box, preferable because the associated names will be used to annotate the axes.
- This style of plot is sometimes called a "beeswarm".

Grouping variables in Seaborn Swarmplot with different attributes,

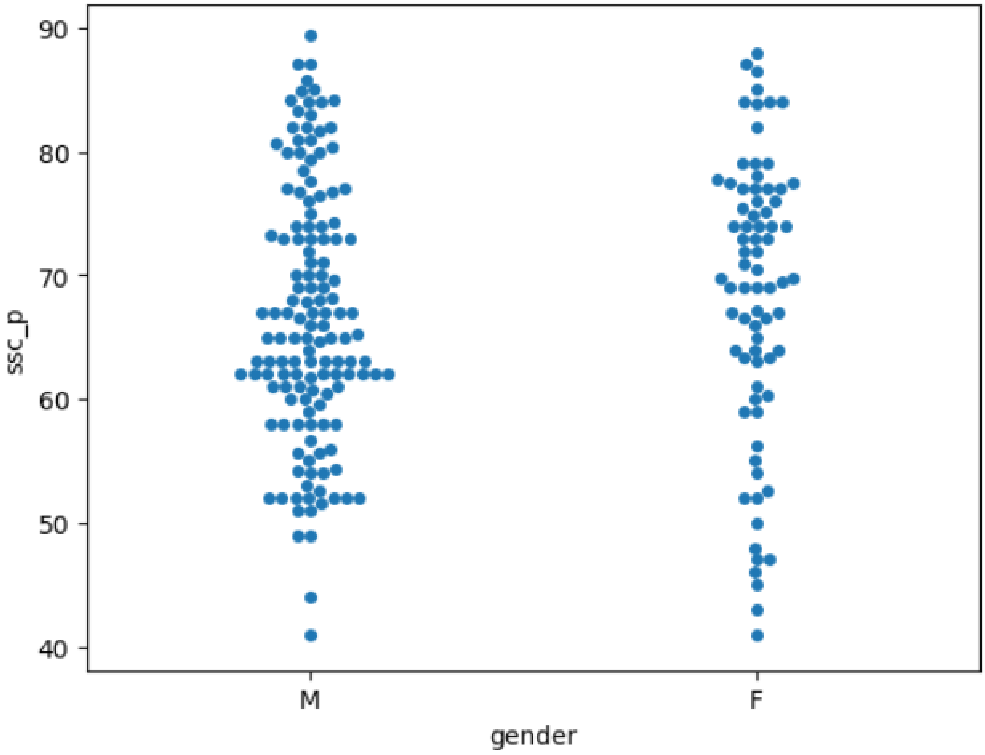
1. Single horizontal swarm plot using only one axis
2. horizontal swarms
3. Using hue parameter
4. Draw outlines around the data points using linewidth
5. Draw each level of the hue variable at different locations on the major categorical axis
6. Plotting large points and different aesthetics with marker and alpha parameter
7. Control swarm order by passing an explicit order
8. Adding size attributes
9. Adding the palette attributes

CODE:



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

IMAGE:



EXPLANATION:

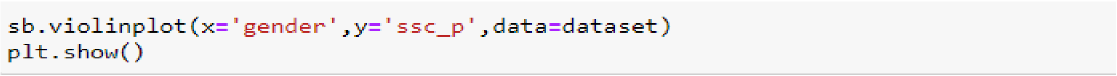
- Swarmplot is plotted between ssc marks of students and categorical data ‘gender’.
- By analysing the graph, Male students mark range starts from 40 and ends at the range of 90 marks.
- Female students mark range also starts from 40 and ends at the 86 marks.
- So comparatively, male students performed well than female students in ssc exam.
- Swarmplot points are adjusted and arranged, so that they won’t overlap.

VIOLIN PLOT:

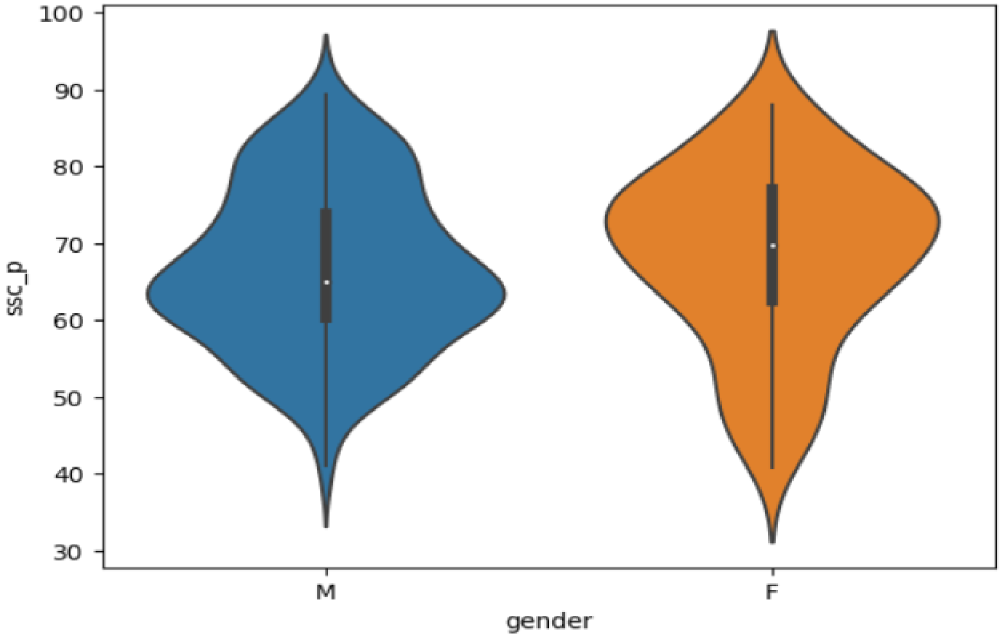
- A violin plot plays a similar role as a box-and-whisker plot. It shows the distribution of data points after grouping by one (or more) variables.

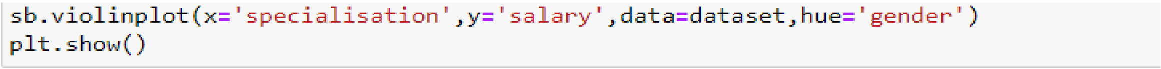
- Unlike a box plot, each violin is drawn using a kernel density estimate of the underlying distribution.
- Violin plot shows data range using box plot and density distribution of data.

CODE AND IMAGE:

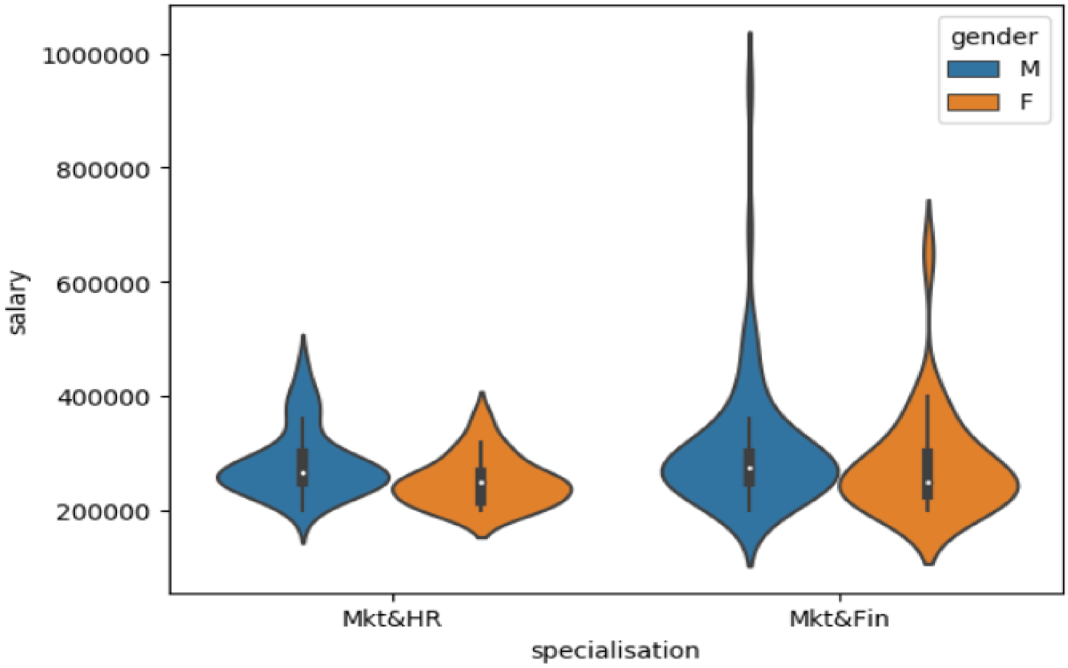


```
sb.violinplot(x='gender',y='ssc_p',data=dataset)
plt.show()
```





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



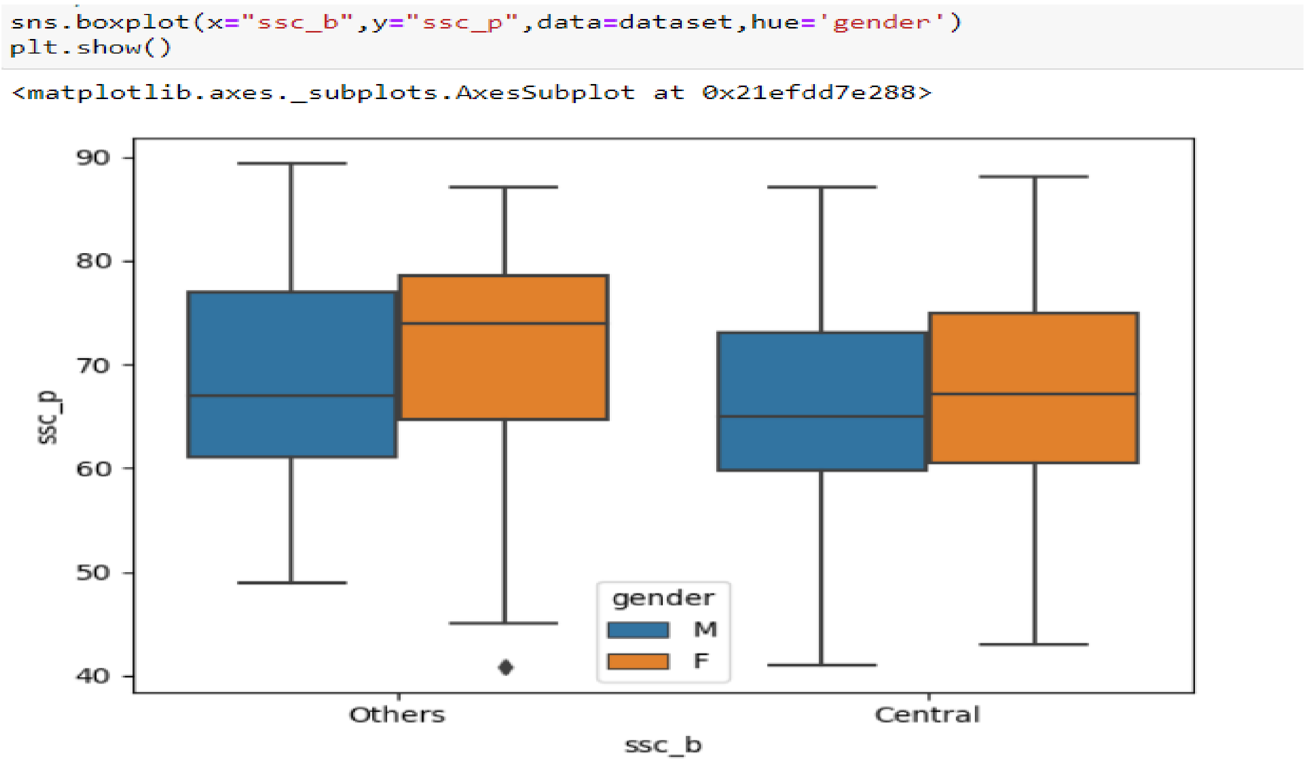
EXPLANATION:

- In violin plot, we can see both density and box plot range between input and output.
- The above graph is plotted to find which specialisation gets more salary.
- In Mkt and HR specialisation, salary range of male starts from 0 to 6 lakhs and female starts from 0 to 5 lakhs.
- In Mkt and Finance specialisation, salary range of male starts from 2 lakhs to 10 lakhs and in female starts from 2 lakhs to 8 lakhs.
- By analysing the violin plot graph, we can easily conclude that Mkt and Finance specialisation gets more salary than Mkt and HR.

BOX PLOT:

- A box plot (or box-and-whisker plot) shows the distribution of quantitative data in a way that facilitates comparisons between variables or across levels of a categorical variable.
- The box shows the quartiles of the dataset while the whiskers extend to show the rest of the distribution, except for points that are determined to be “outliers” using a method that is a function of the inter-quartile range.

CODE AND IMAGE:



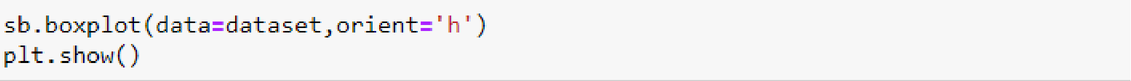
```
sns.boxplot(x="ssc_b",y="ssc_p",data=dataset,hue='gender')
plt.show()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x21efdd7e288>
```

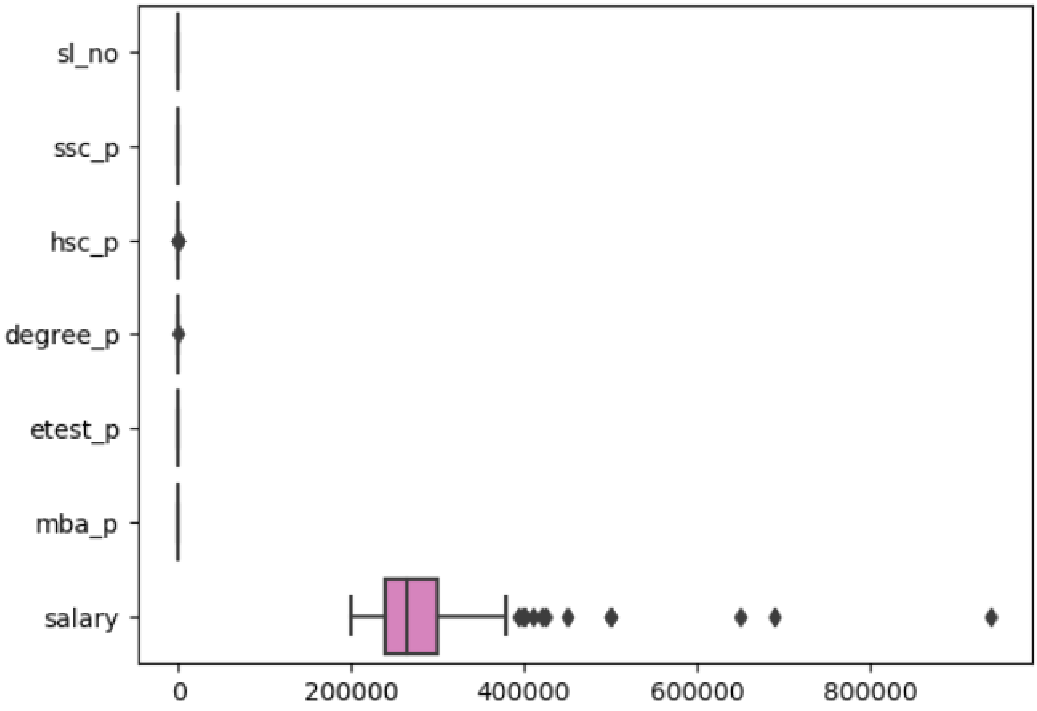
EXPLANATION:

- Box plot resembles the concept of percentile, the box shaped graph presents in between the start range and end range
- Bottom line of boxplot represents Q1, mid line of box represents Q2, the upper third line is Q3 and end range is Q4
- In this above graph is plotted between boards of ssc and students marks of ssc.
- hue=gender is used to different the marks of male and female in others and central.
- In others, male students mark range is 50 to 90 and female students mark range starts from 45 to 86, so male students performed well.
- In central board, male students mark range is 42 to 86 and female students mark range starts from 45 to 88, so female students performed well.
- In overall performance, others board students performed well in ssc than central board students.

BOX PLOT IN HORIZONTAL:



```
sb.boxplot(data=dataset,orient='h')
plt.show()
```

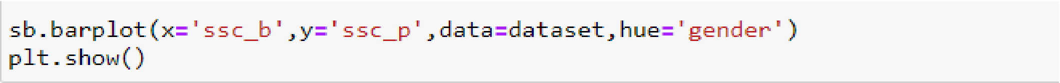


This shows box plot of all data in the dataset but readability is poor.

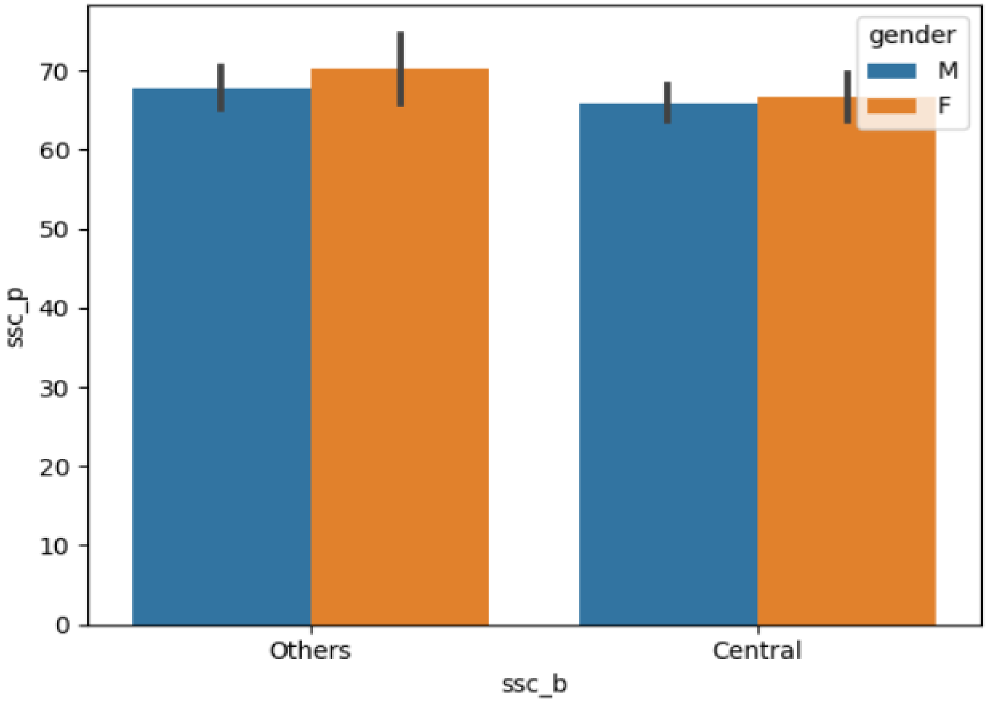
BAR PLOT:

- Bar plot is a visualization tool provided by the Seaborn library in Python for creating bar plots with enhanced aesthetics.
- With Seaborn, you can easily create bar plots by specifying the data, categorical variables, and optional parameters for customizing the appearance, such as colors, annotations, and error bars.
- Seaborn's bar plots are particularly useful for visualizing relationships between categorical variables or comparing distributions across different categories.
- They provide a visually appealing way to explore and communicate patterns in your data.

CODE AND IMAGE:



```
sb.barplot(x='ssc_b',y='ssc_p',data=dataset,hue='gender')
plt.show()
```



EXPLANATION:

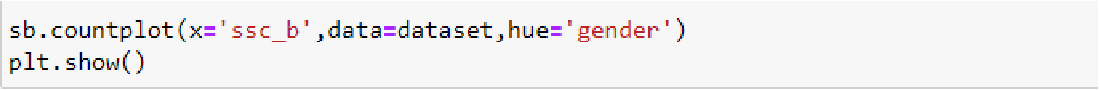
- Barplot is plotted between ssc boards and ssc student marks, to find out who is performed better.
- Barplot shows that, male students scored up to 68 range and female students scored up to 70 range in others board section.
- In central board, male students scored up to 66 range and female students scored up to 67 range.

- In comparision, female students of others board performed well.

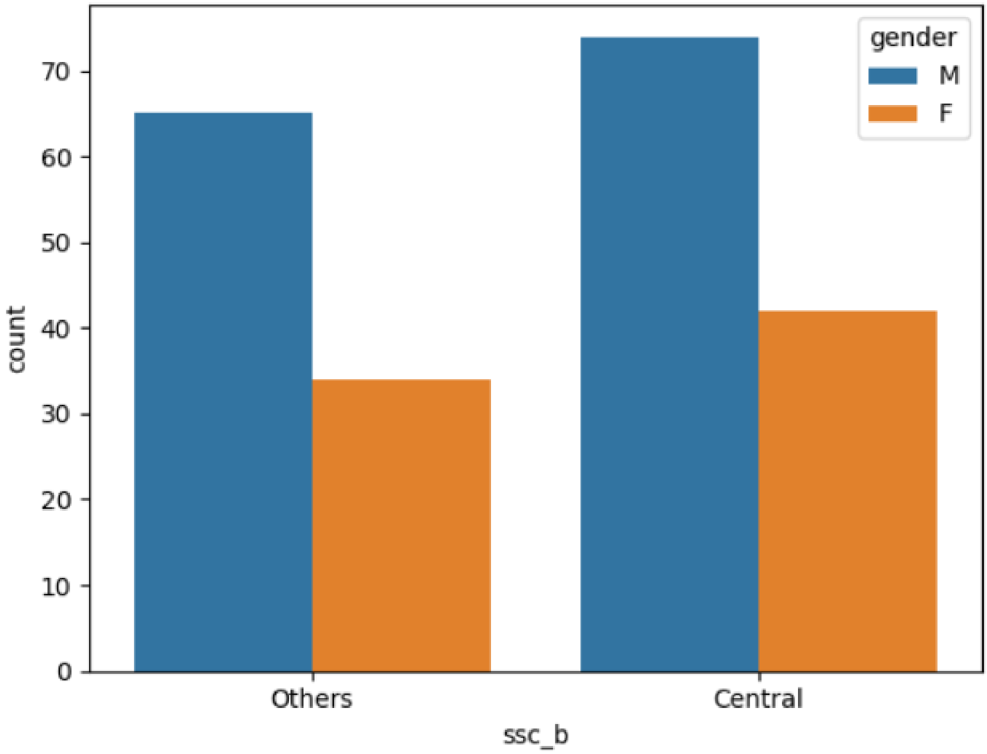
COUNT PLOT:

- Count plot is a type of bar plot that displays the count of observations in each category of a categorical variable. It's a convenient way to visualize the distribution of categorical data.
- With Seaborn's count plot, you can quickly examine the frequency or occurrence of different categories within a dataset.
- The count plot function in Seaborn automatically aggregates the data and creates a bar plot with counts on the y-axis and categories on the x-axis. It's a powerful tool for understanding the distribution and frequency of categorical variables in your dataset, providing insights into patterns and trends.

CODE AND IMAGE:



```
sb.countplot(x='ssc_b',data=dataset,hue='gender')
plt.show()
```



EXPLANATION:

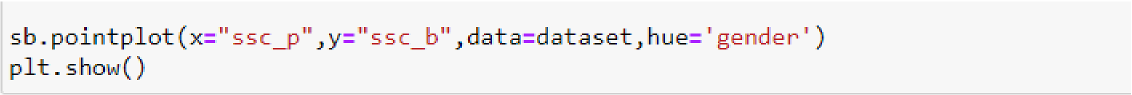
- By analysing the above count plot, we can identify the students strength and we can get count of each section.

- In others board, male students count is 65 and female students count is 32.
- In central board, male students count is 75 and female students count is 40.

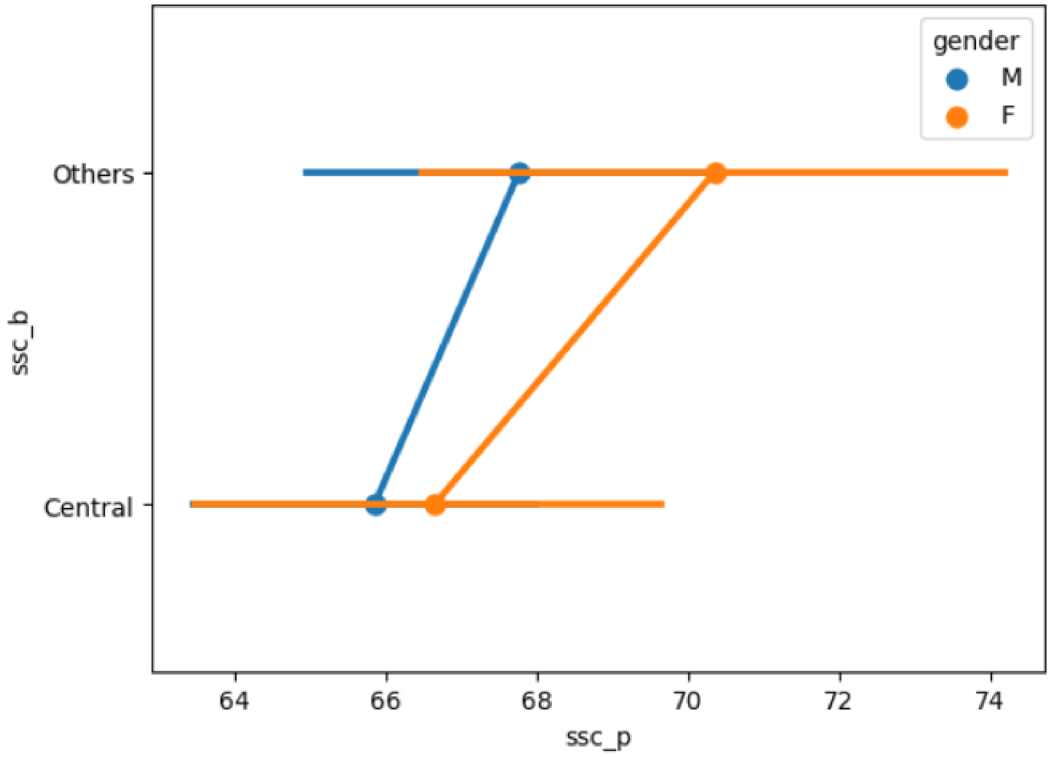
POINTPLOT:

- Point plot is a type of plot used to visualize the relationship between two categorical variables and one numeric variable. It displays the mean or median of the numeric variable for each category, along with confidence intervals or error bars.
- The categorical variables are typically plotted on the x-axis, while the numeric variable is plotted on the y-axis.
- Point plots are useful for comparing the central tendency of the numeric variable across different categories and for identifying trends or patterns in the data.
- They provide a concise way to visualize relationships between variables and are particularly effective when there are relatively few data points per category.

CODE AND IMAGE:



```
sb.pointplot(x="ssc_p",y="ssc_b",data=dataset,hue='gender')
plt.show()
```



EXPLANATION:

- Point plot is plotted between 2 categorical variables and one qualitative variable.
- In the above graph, point plot is plotted between ssc board and ssc pass marks of students and it is differentiated through two different coloured lines.
- Ssc board is classified into two, others and central
- Through point plot, we can analyse the relationship between input and output variables. Here others board female students performed well.

CATPLOT:

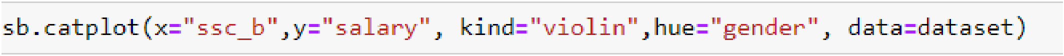
Seaborn's catplot is a high-level function that allows for the creation of categorical plots with ease. It provides a versatile interface for generating various types of categorical plots such as strip plots, swarm plots, box plots, violin plots, point plots, bar plots, and more.

With catplot, you can:

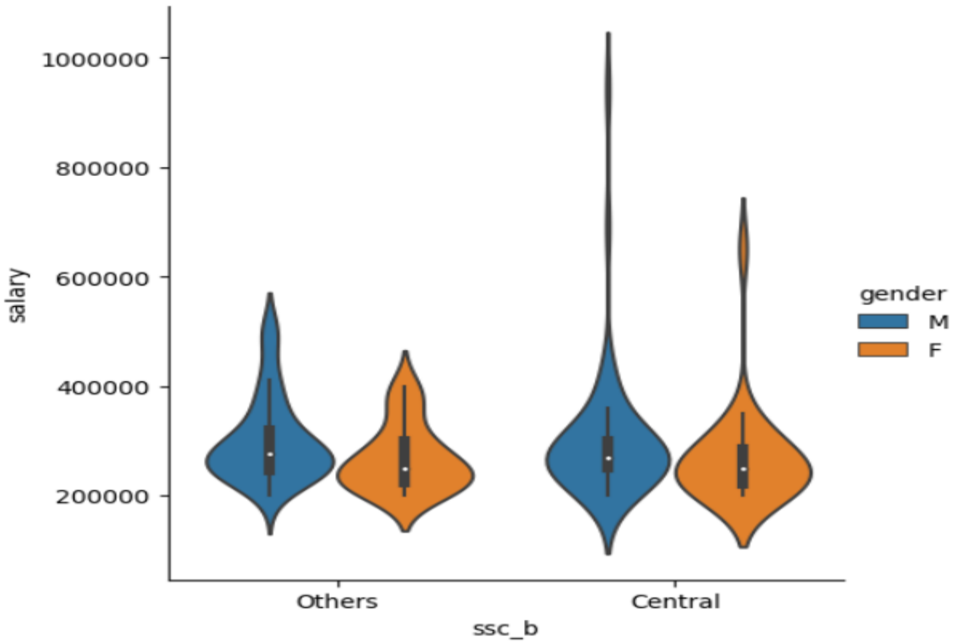
- Visualize Categorical Data: Explore relationships and distributions within categorical variables.
- Customize Plots: Easily adjust plot aesthetics including colors, markers, and error bars.
- Facet Plots: Create facet grids to visualize additional dimensions of the data.
- Statistical Estimation: Automatically compute and visualize statistical estimates like mean, median, or confidence intervals.
- Handle Different Data Formats: Work seamlessly with different data formats including pandas DataFrames and long-form or wide-form data.

In summary, catplot is a powerful tool for visualizing categorical data, providing insights into relationships and distributions within your dataset while offering flexibility and customization options.

CODE AND IMAGE:



```
sb.catplot(x="ssc_b",y="salary", kind="violin",hue="gender", data=dataset)
```



EXPLANATION:

- Catplot is drawn between ssc boards and salary, to find out which board gets high salary.
- In the above catplot we gave kind='violin', so the graph is plotted in violin plot and we can also change this by giving other plot graph in the kind parameter.
- With the above graph, we can easily find out central board male students got the highest salary than others.

REGRESSION PLOT:

A regression plot, often created using Seaborn's regplot function, is a type of scatter plot that includes a fitted regression line to show the relationship between two variables. It's particularly useful for visualizing linear relationships between continuous variables and for identifying trends or patterns in the data.

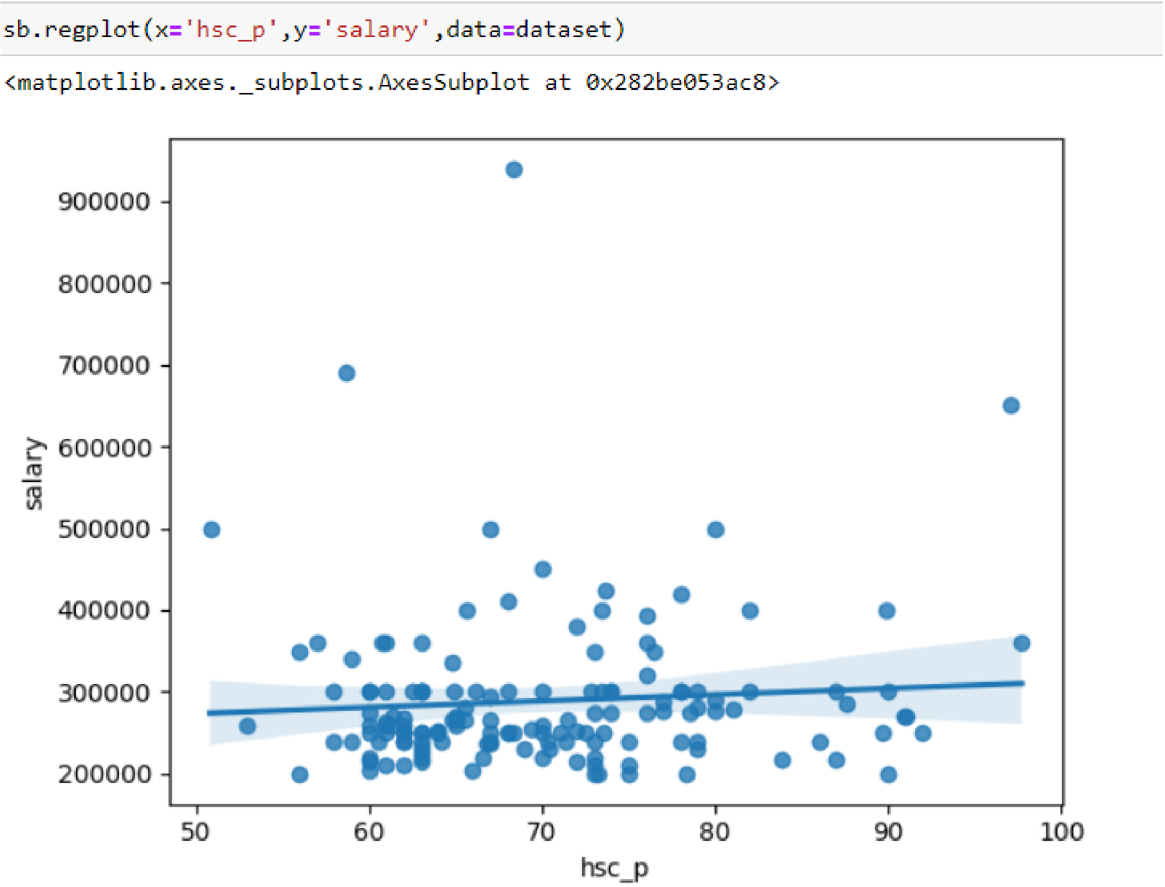
Here's a brief explanation of regression plots:

- Scatter Plot: The data points are plotted on a Cartesian plane, with one variable on the x-axis and the other on the y-axis.
- Regression Line: A line of best fit is drawn through the data points, representing the relationship between the two variables. This line is determined by a regression model fitted to the data.
- Trend Analysis: The regression line helps to visualize the general trend or direction of the relationship between the variables. For example, if the

line slopes upwards, it suggests a positive correlation, while a downward slope indicates a negative correlation.

- Prediction: Regression plots can also be used for making predictions about one variable based on the value of another, using the fitted regression equation.

CODE AND IMAGE:



```
sb.regplot(x='hsc_p',y='salary',data=dataset)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x282be053ac8>
```

LMPLOT:

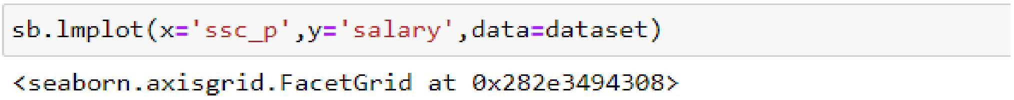
lmplot is a function provided by the Seaborn library in Python for creating scatter plots with a fitted regression line. It stands for "Linear Model Plot" and is a higher-level interface that combines functionality from regplot and FacetGrid.

Here's a concise explanation of lmplot:

- Scatter Plot: lmplot creates a scatter plot where data points are plotted based on two continuous variables.
- Regression Line: It includes a fitted regression line to visualize the linear relationship between the variables.

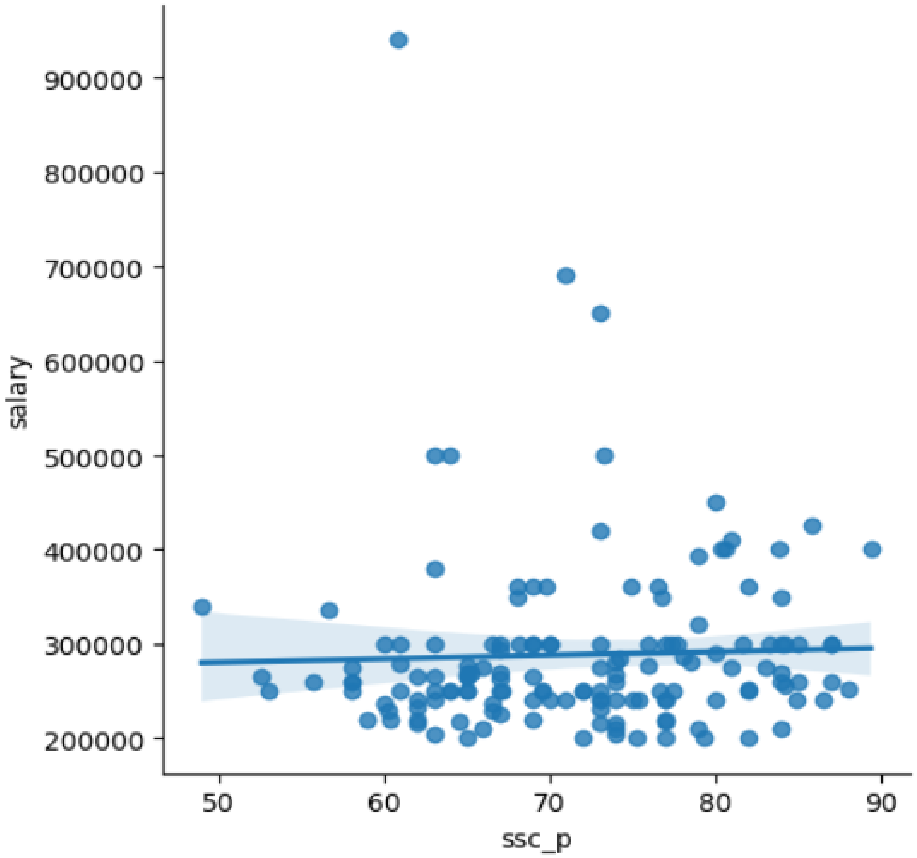
- Faceting: lmplot also supports faceting, allowing you to create multiple plots arranged in a grid, each representing subsets of the data based on one or more categorical variables. This is particularly useful for comparing relationships across different categories.
- Customization: Similar to other Seaborn plots, lmplot provides options for customizing the appearance of the plot, such as adjusting markers, colors, and error bars.

CODE AND IMAGE:



```
sb.lmplot(x='ssc_p',y='salary',data=dataset)
```

```
<seaborn.axisgrid.FacetGrid at 0x282e3494308>
```


salary
ssc_p