seaborn

What is Seaborn?

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

IMPORTING THE LIBRARIES

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

IMPORTING DATA

```
•sns.get_dataset_names() #returns the data sets that seaborn provides.
•planets_ds = sns.load_dataset('planets')
•penguins_ds = sns.load_dataset('penguins')
#loading the data sets that seaborn provides
```

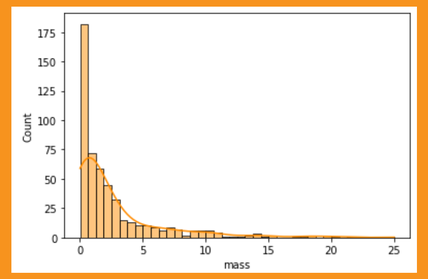
PLOTS

Planets Dataset

	method	number	orbital_period	mass	distance	year
0	Radial Velocity	1	269.300000	7.10	77.40	2006
1	Radial Velocity	1	874.774000	2.21	56.95	2008
2	Radial Velocity	1	763.000000	2.60	19.84	2011
3	Radial Velocity	1	326.030000	19.40	110.62	2007
4	Radial Velocity	1	516.220000	10.50	119.47	2009
...	...	...	...	...	...	...
1030	Transit	1	3.941507	NaN	172.00	2006
1031	Transit	1	2.615864	NaN	148.00	2007
1032	Transit	1	3.191524	NaN	174.00	2007
1033	Transit	1	4.125083	NaN	293.00	2008
1034	Transit	1	4.187757	NaN	260.00	2008

Histogram Plot

```
sns.histplot(data=planets_ds,#creating the plot
             x='mass',#giving the x value to plot
             bins=40,#arranging the bin size
             kde=True,#enabling kernel density
                      #estimate
             color='orange')#arranging the color
```

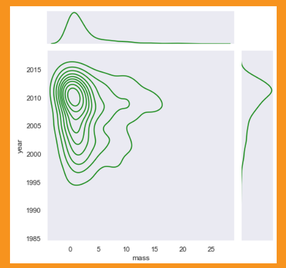


Scatter Plot

```
sns.scatterplot(data=planets_ds,#creating plot
                x='year',#giving x value
                y='orbital_period',#giving y value
                hue='mass')#color mapping
```

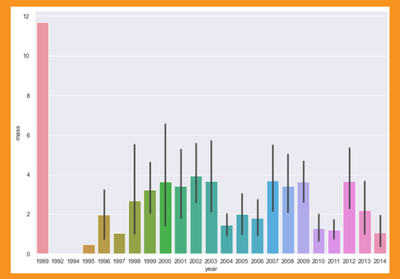
Joint Plot

```
sns.jointplot(data=planets_ds,#creating the plot
              x='mass',#giving the x value to plot
              y='year',#giving the y value
              color='green'#arranging the color
              kind='kde')#changing the kind to
                         #kernel density
                         #estimate - kde
```



Bar Plot

```
sns.barplot(data=planets_ds,#creating the plot
            x='year',#giving the x value
            y='mass')#giving the y value
```



OTHER PLOTS

```
•sns.boxplot(x="day",
             y="total_bill",
             hue="smoker",
             palette["m", "g"],
             data=tips)

•sns.catplot(x="time",
             y="pulse",
             hue="kind",#color mapping
             col="diet",#column
             palette="YlGnBu_d",#color palette
             height=6,#changing figure size
             aspect=.75,#changing figure size
             kind="point",#changing plot kind
             data=df)

•sns.countplot(data=planets_ds,
               x='year')

•sns.distplot(data.y,
              kde=False,
              color="b")

•sns.dogplot()#shows a cute face of a dog:)

•sns.ecdfplot(data=planets_ds,
              x='orbital_period')

•sns.factorplot(data=planets_ds,
                x='year',
                y='mass')

•sns.kdeplot(data=planets_ds,
             x='mass')

•sns.lineplot(data=penguin_ds,
              x='body_mass_g',
              y='flipper_length_mm')

•sns.pairplot(planets_ds)

•sns.palplot(sns.color_palette("hls", 8))

•sns.pointplot(data=planets_ds,
               x='year',
               y='orbital_period'
               markers["^","o"],
               linestyles["-","--"])

•sns.regplot(data=planets_ds,
             x='mass',
             y='orbital_period')

•sns.relplot(data=planets_ds,
             x='mass',
             y='orbital_period')

•sns.rugplot(data=planets_ds,
             x='mass')

•sns.stripplot(x="value",
               y="measurement",
               hue="species",
               data=iris,
               dodge=True,
               alpha=.25,
               zorder=1)
```

OTHER PLOTS

```
•sns.swarmplot(data=df,
               x="body_mass_g",
               y="sex",
               hue="species")

•sns.violinplot(data=planets_ds,
                x='year',
                y='mass')
```

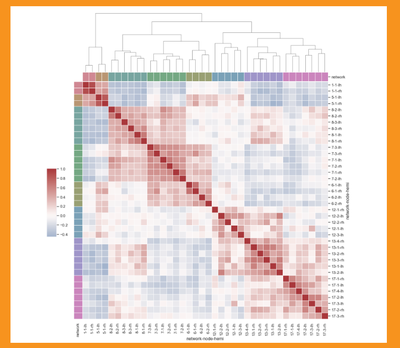
HEAT MAP

```
#Load the example flights dataset and convert
#to long-form
flights_long = sns.load_dataset("flights")
flights = flights_long.pivot("month",
                             "year",
                             "passengers")

sns.heatmap(flights,
            annot=True,#If True, write the data
                       #value in each cell.
            linewidths=.5,
            ax=ax)
```

CLUSTER MAP

```
sns.clustermap(df.corr(),
               center=0,
               cmap="vlag",
               row_colors=network_colors,
               col_colors=network_colors,
               dendrogram_ratio=(.1, .2),
               #Proportion of the figure size
               #devoted to the two marginal
               #elements.
               cbar_pos=(.02, .32, .03, .2),
               #Position of the colorbar axes
               #in the figure.
               linewidths=.75,
               figsize=(12, 13))
```



STYLING

```
•sns.set_style('darkgrid')
#changing the plot style

•sns.set_context('paper',
                 font_scale=1.4,
                 rc={"grid.linewidth": 0.6})
#to change the labels, font scales, line width
#of the grid, etc.
#context can have 3 options besides
#paper(talk-poster)

•sns.despine(left=False,
             bottom=False)
#manages the right, left, top and bottom axes'
#looks.

•sns.set_palette("husl",3)
#Define the color palette
•my_cpalette=["#ED1C24",
              "#00B837",
              "#FCEE21",
              "#662D91"]
 sns.set_palette(my_cpalette)
#we can use our own color palette by giving
#hexadecimal numbers in a list
```

Categorical Plots

- Scatter Plot
- Bar Chart
- Count Plot
- Point Plot
- Box Plot
- Violin Plot

Show or Save the Plot

```
•plt.show()#show the plot
•plt.savefig("gimli.jpg")#save the plot
•plt.savefig("ahsoka.jpg",
              transparent=True)#save transparent
                               #figure
```

Close and Clear

```
•plt.cla()#clear an axis
•plt.clf()#clear an entire figure
•plt.close()#close a window
```

STUDENT

FURKAN ÖZTÜRK
200315002