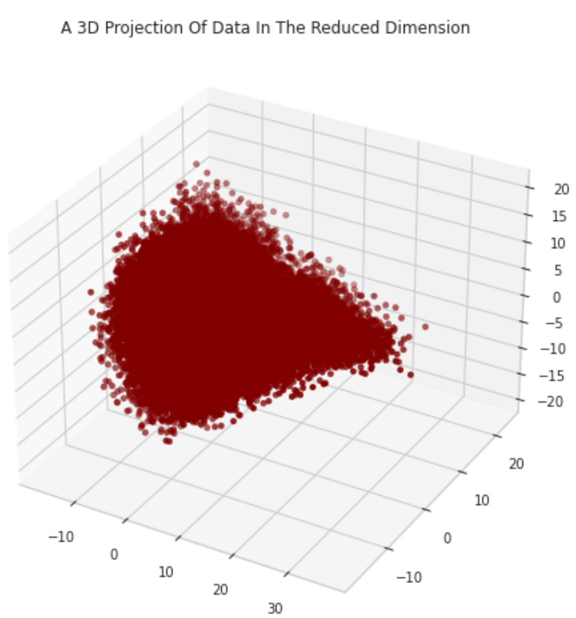
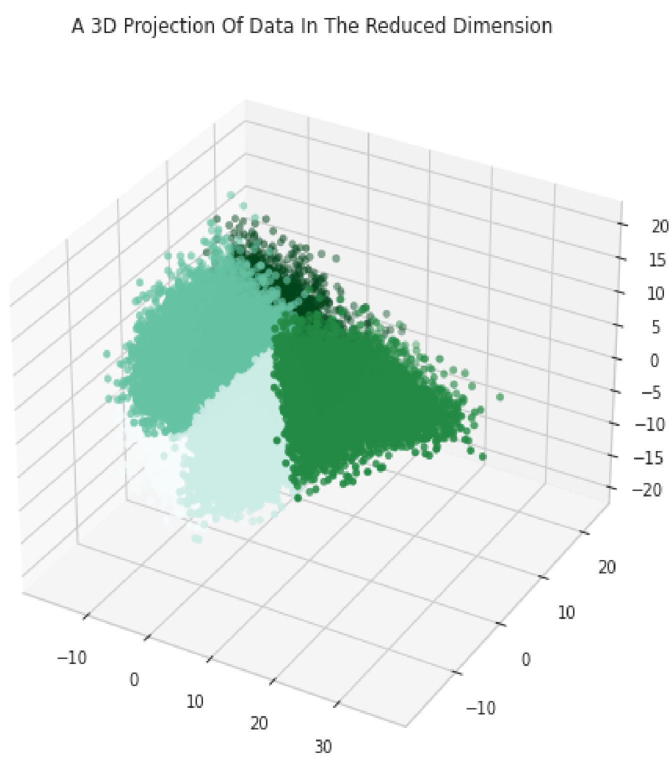
Mastering Data Visualization Techniques (Part 4)

Prepared by: Syed Afroz Ali

```
X = dataset.copy()
from sklearn.decomposition import PCA
pca = PCA(n_components=3)
pca.fit(X)
PCA_ds = pd.DataFrame(pca.transform(X), columns=(["col1","col2", "col3"]))

# A 3D Projection Of Data In The Reduced Dimension
x =PCA_ds["col1"]
y =PCA_ds["col2"]
z =PCA_ds["col3"]
#To plot
fig = plt.figure(figsize=(10,8))
ax = fig.add_subplot(111, projection="3d")
ax.scatter(x, y, z, c="maroon", marker="o" )
ax.set_title("A 3D Projection Of Data In The Reduced Dimension")
plt.show()
```



```
# A 3D Projection Of Data In The Reduced Dimension
x =PCA_ds["col1"]
y =PCA_ds["col2"]
z =PCA_ds["col3"]

#To plot
fig = plt.figure(figsize=(10,8))
ax = fig.add_subplot(111, projection="3d")
ax.scatter(x, y, z, c=labels, marker="o", cmap="BuGn")
ax.set_title("A 3D Projection Of Data In The Reduced Dimension")
plt.show()
```

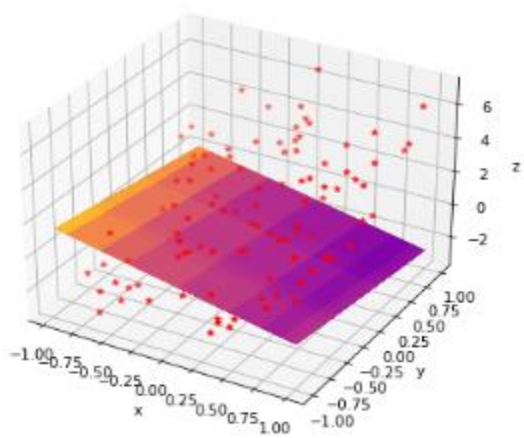



```
for i in range(0, 10):
    fig = plt.figure(figsize=(8, 6))
    ax = plt.axes(projection="3d")

    ax.scatter(x, y, z, marker='*', color='red')

    X, Y = np.meshgrid(x, y)
```

Syed Afroz Ali
Follow me on LinkedIn for more: https://www.linkedin.com/in/syed-afroz-70939914/

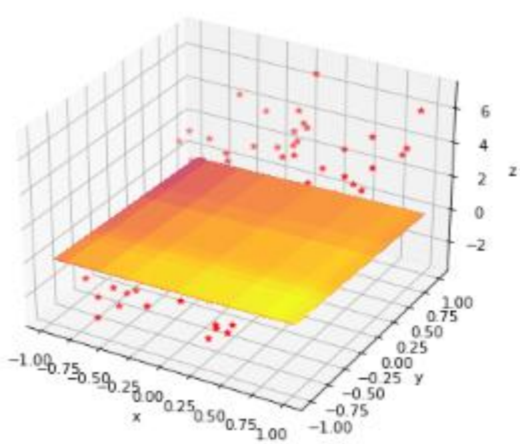
```
    Z = theta_0[i]*X + theta_1[i]*Y + theta_2[i]
    ax.plot_surface(X, Y, Z, cmap='plasma')

    ax.set_xlabel("x")
    ax.set_ylabel("y")
    ax.set_zlabel("z")
    ax.set_title("Thetas: {},{},{}".format(theta_0[i], theta_1[i], theta_2[i]))
    plt.show()
    print(40*"=")
```

Thetas: -1.4035055897613047,-0.763083683174633,-0.4754543787231061




```
========================================
```

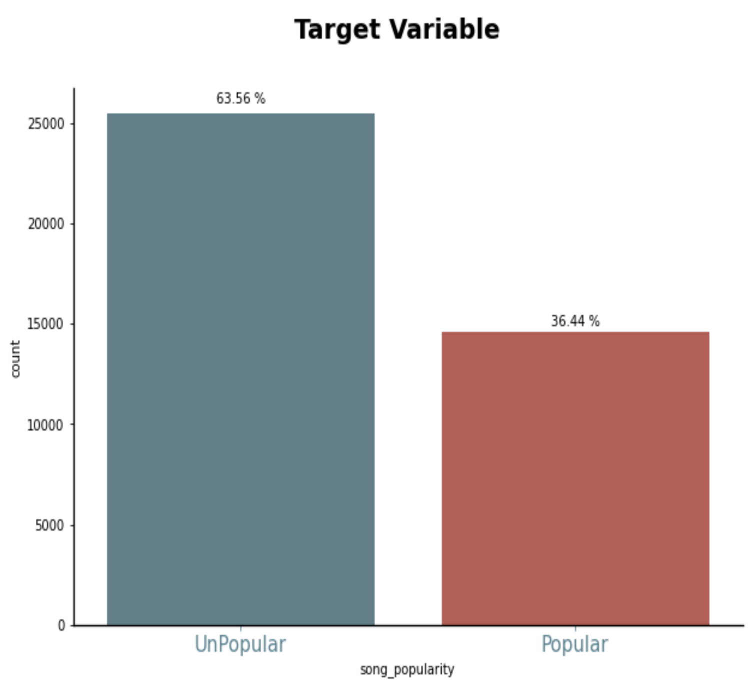
Thetas: 0.12226339350069884,-0.22698056361690117,0.006858537493075969



Syed Afroz Ali

Follow me on LinkedIn for more: https://www.linkedin.com/in/syed-afroz-70939914/

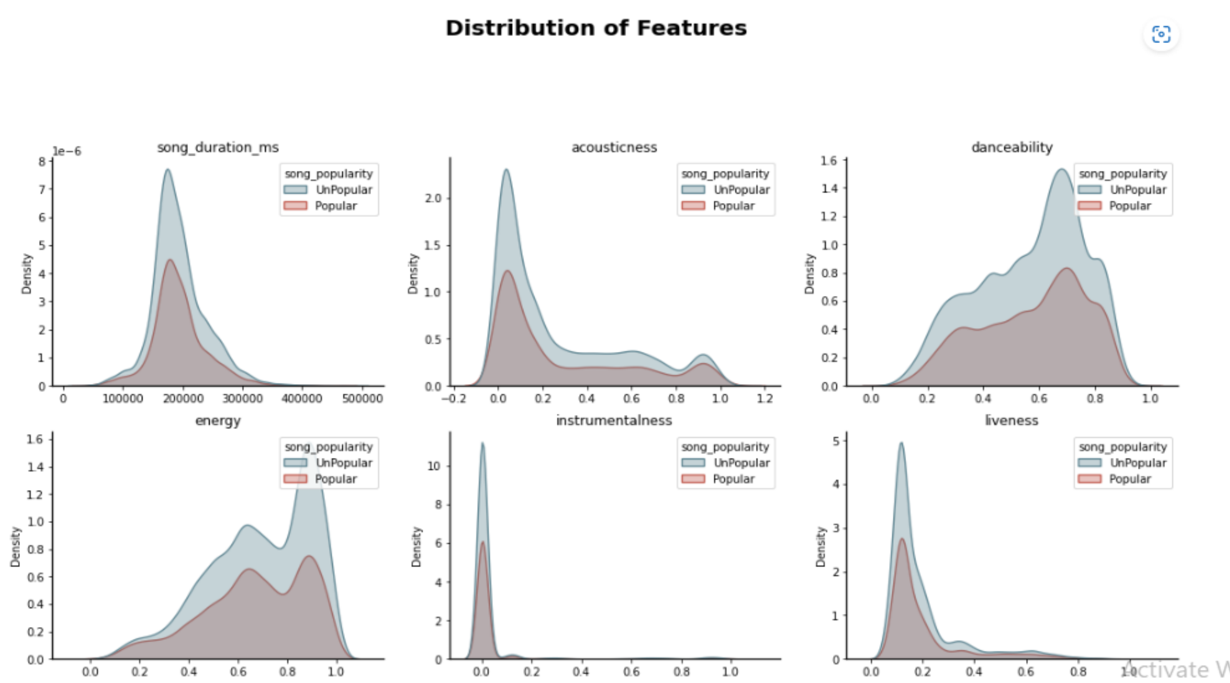
```
plt.suptitle('Target Variable', size = 20, weight='bold')

song_popularity = df['song_popularity'].map({0:'UnPopular', 1:'Popular'})

a = sns.countplot(data = df, x =song_popularity,palette=theme)
plt.tick_params(axis="x", colors=theme[0],labelsize=15)

for p in a.patches:
    width = p.get_width()
    height = p.get_height()
    x, y = p.get_xy()
    a.annotate(f'{height/df.shape[0]*100} %', (x + width/2, y + height*1.0
2), ha='center')

plt.show()
```




```
cont = ['song_duration_ms', 'acousticness', 'danceability', 'energy',
       'instrumentalness', 'liveness', 'loudness',
       'speechiness', 'tempo', 'audio_valence']
cat = [ 'key', 'audio_mode', 'time_signature']
```

Syed Afroz Ali
Follow me on LinkedIn for more: https://www.linkedin.com/in/syed-afroz-70939914/

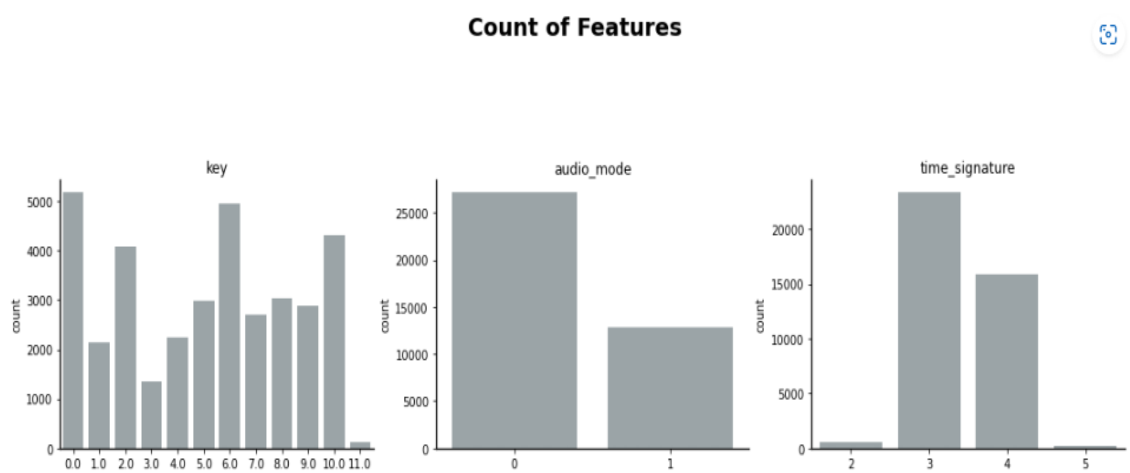
```
a = 4  # number of rows
b = 3  # number of columns
c = 1  # initialize plot counter

plt.figure(figsize= (18,18))

for i in cont:
	plt.suptitle('Distribution of Features', size = 20, weight='bold')
	plt.subplot(a, b, c)
	A=sns.kdeplot(data= df, x=i,hue=song_popularity,palette=theme[:-2], linewidth = 1.3,shade=True, alpha=0.35)
	plt.title(i)
	plt.xlabel(" ")
	c = c + 1
```



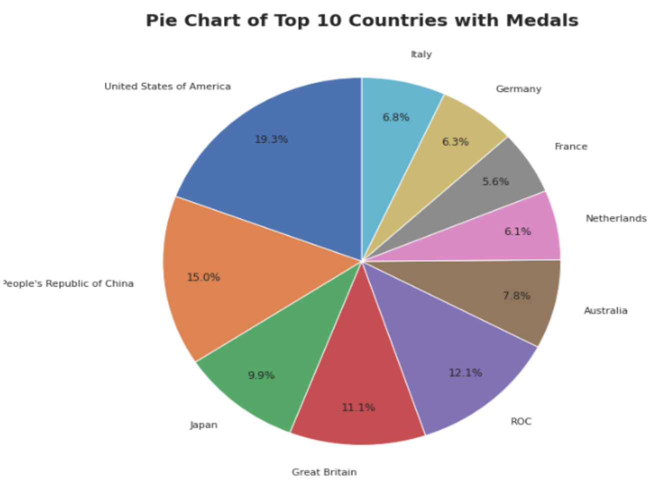

```
a = 4  # number of rows
b = 3  # number of columns
c = 1  # initialize plot counter

plt.figure(figsize= (18,18))
```

Syed Afroz Ali
Follow me on LinkedIn for more: https://www.linkedin.com/in/syed-afroz-70939914/

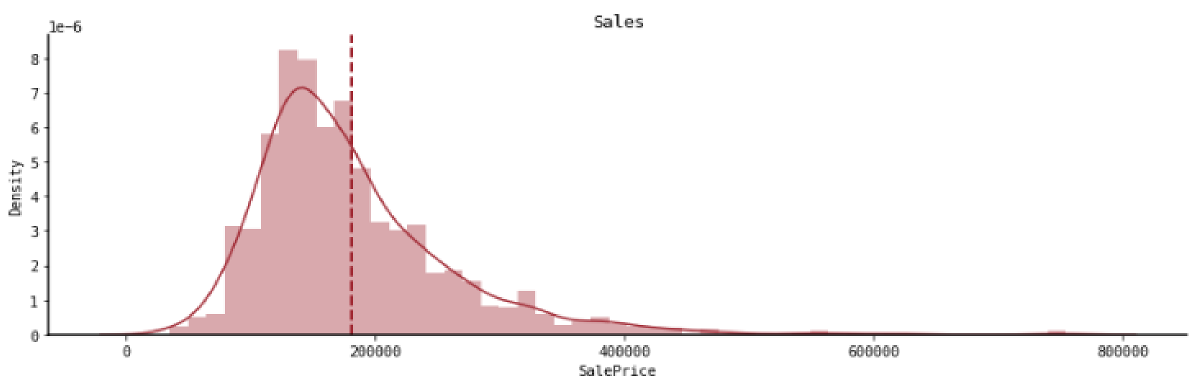
```python
for i in cat:

    plt.suptitle('Count of Features', size = 20, weight='bold')

    plt.subplot(a, b, c)
    A=sns.countplot(df[i],color=theme[3], alpha=0.5)
    plt.title(i)
    plt.xlabel(" ")
    plt.tick_params(axis="x", colors='black',labelsize=10)
    c = c + 1
```




```python
figure = plt.figure(figsize=(30,10))

A = plt.pie(medals['Total'][:10],
    labels=medals['Country'][:10],
    startangle=90,
    labeldistance=1.15,
    pctdistance=0.8,
    autopct='%1.1f%%')
plt.title("Pie Chart of Top 10 Countries with Medals",size=20,weight='b
old')
plt.show();
```



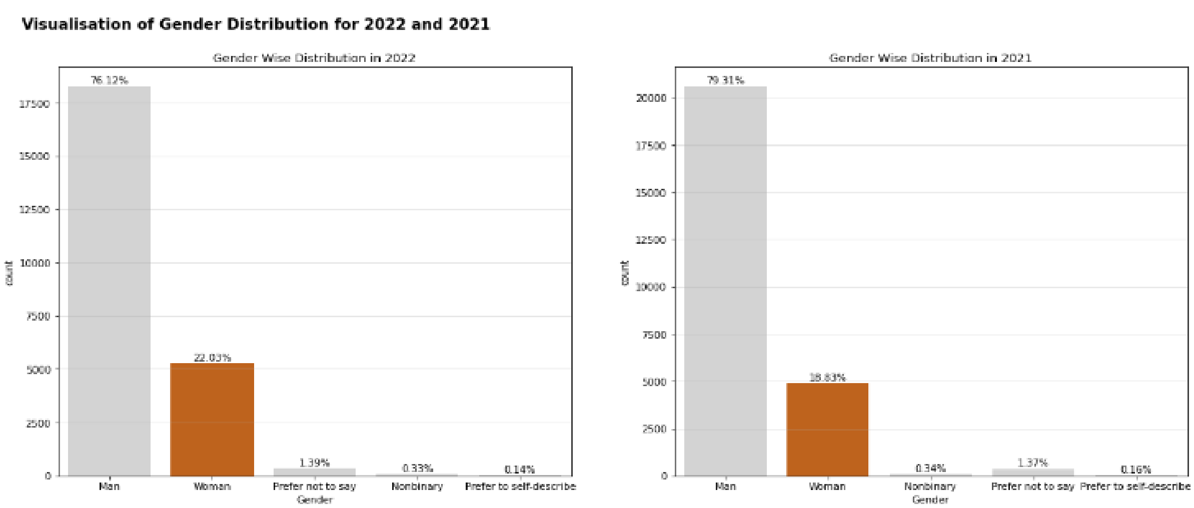

```
#checking the target variables for distribution
sns.distplot(house['SalePrice'],color=colors[7])
plt.axvline(x=house['SalePrice'].mean(), color=colors[7], linestyle='--', li
newidth=2)
plt.title('Sales');
```




```
l = df_current['Q3'].value_counts(normalize=True).mul(100).tolist()[1]-df
_old['Q2'].value_counts(normalize=True).mul(100).values.tolist()[1]

print(5*'\n',"\033[1;32m Increase in Woman is only\033[1;32m",round(l,
2),'%\033[1;32m Over Last Year\033[1;32m',5*'\n')
```

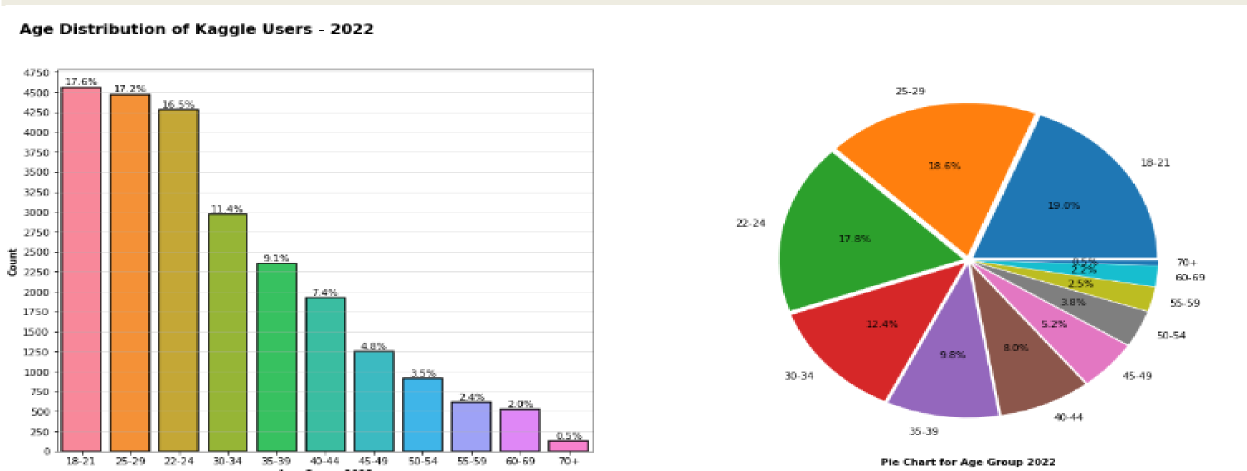
Syed Afroz Ali
Follow me on LinkedIn for more: https://www.linkedin.com/in/syed-afroz-70939914/

```
fig, ax = plt.subplots(1, 2, figsize=(20,8))
fig.text(0.1, 0.95, "Visualisation of Gender Distribution for 2022 and 2021", fontsize=15, fontweight='bold')

sns.countplot(x='Q3', data=df_current,palette="Dark2", ax=ax[0]); #Current Year
sns.countplot(x='Q2', data=df_old,palette="Dark2",ax=ax[1]); #Last Year

for i, ax in enumerate(ax.flatten()):
    ax.grid(axis='y', linestyle='-', alpha=0.4)
    if i==0:t=shape;year = 2022
    else:t=shape_21;year =2021
    for p in ax.patches:
        percentage = f'{100 * p.get_height() / t:.2f}%\n'
        ax.annotate(percentage, (p.get_x() + p.get_width() / 2,p.get_height()), ha='center', va='center')
        ax.set_xlabel('Gender');ax.set_title("Gender Wise Distribution in "+str(year))
        if not(0.5 <= p.get_x() < 1.5):
            p.set_facecolor('lightgrey')

plt.show()
```



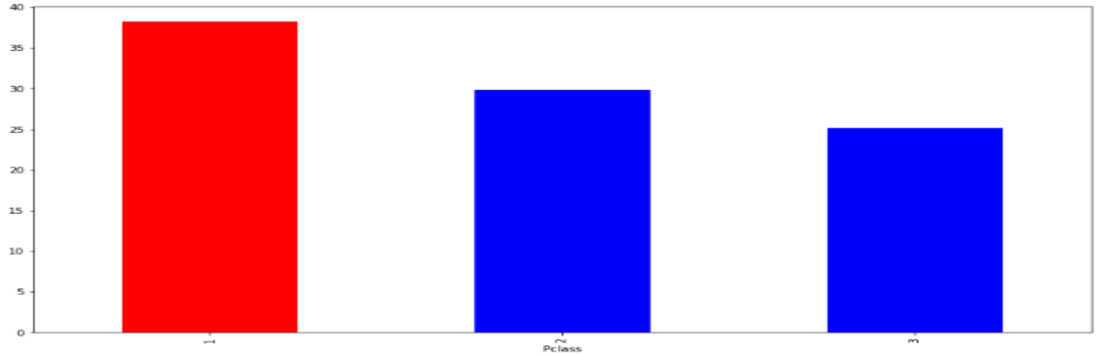
Syed Afroz Ali
Follow me on LinkedIn for more: https://www.linkedin.com/in/syed-afroz-70939914/

```
fig, ax = plt.subplots(1,2, figsize=(20,8))

fig.text(0.1, 0.95, "Age Distribution of Kaggle Users - 2022", fontsize=1
5, fontweight='bold')

sns.barplot(x=df_current['Q2'].value_counts().index, y=df_current['Q2'].
value_counts().values, ax=ax[0],
          edgecolor='black', linewidth=1.5, saturation=1.5)
ax[0].yaxis.set_major_locator(MaxNLocator(nbins=20));ax[0].grid(axis=
'y', linestyle='-', alpha=0.4)
ax[0].set_ylabel('Count', weight='semibold')
ax[0].set_xlabel('Age Group 2022', weight='semibold')
ax[1].set_xlabel('Pie Chart for Age Group 2022', weight='semibold')
for p in ax[0].patches:
       percentage = f'{100 * p.get_height() / t:.1f}%\n'
       ax[0].annotate(percentage, (p.get_x() + p.get_width() / 2,p.get_hei
ght()), ha='center', va='center')

ax[1].pie(df_current['Q2'].value_counts(), labels = df_current['Q2'].value
_counts().index,  autopct='%1.1f%%',
       explode=[0.03 for i in df_current['Q2'].value_counts().index])

plt.show()
```

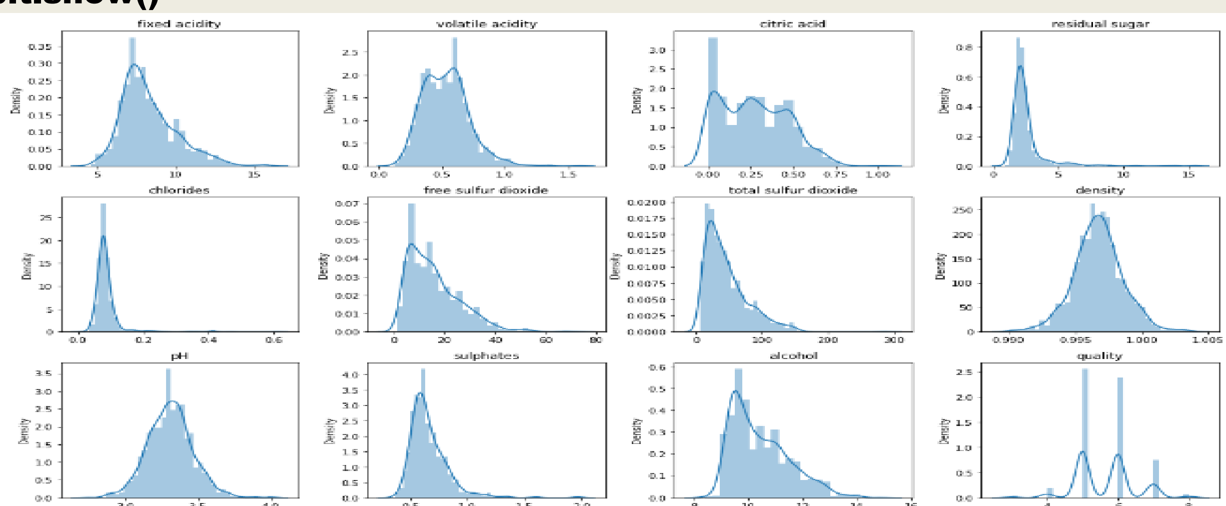


Syed Afroz Ali
Follow me on LinkedIn for more: https://www.linkedin.com/in/syed-afroz-70939914/

```
df2=titanic.groupby('Pclass')['Age'].mean().sort_values(ascending=False)
plt.figure(figsize = (15,8))
color = [('b' if i < 30 else 'r') for i in df2]
df2.plot.bar(color=color);
```



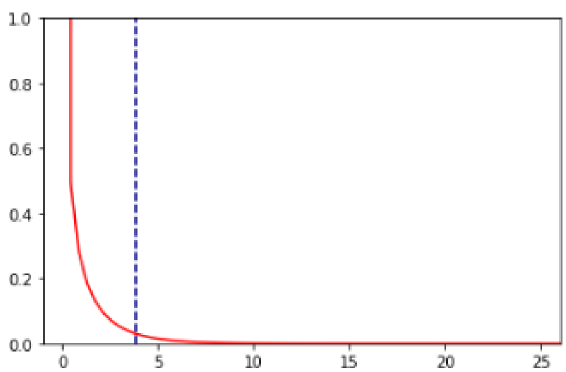

```
col=['fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar',
       'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'density',
       'pH', 'sulphates', 'alcohol', 'quality']

fig = plt.figure(figsize=(15,10))

for i in range(len(col)):
    plt.subplot(3,4,i+1)
    plt.title(col[i])
    sns.distplot(df,x=df[col[i]])
plt.tight_layout()
plt.show()
```



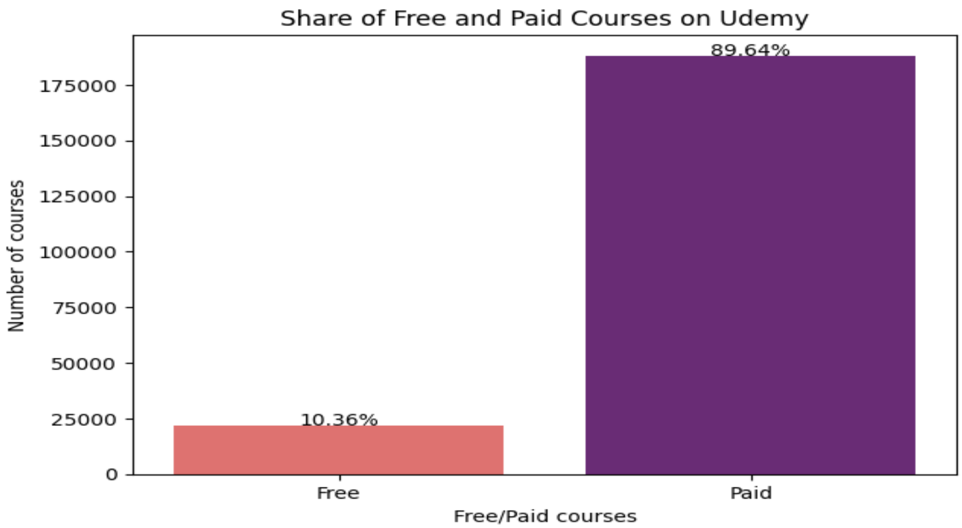
Syed Afroz Ali
Follow me on LinkedIn for more: https://www.linkedin.com/in/syed-afroz-70939914/

```
fig, ax = plt.subplots(1, 1)

plt.xlim(-1,26)
plt.ylim(0,1)
x = np.linspace(f.ppf(0.0000000001, dfn, dfd),f.ppf(0.9999999999, dfn, dfd), 100)
ax.plot(x, f.pdf(x, dfn, dfd), 'r-')
ax.axvline(f.ppf(0.95, dfn, dfd), ls = "--", color = "navy")
print('upper 5%:', f.ppf(0.95, dfn, dfd))
```

```
upper 5%: 3.8426563592313365
```

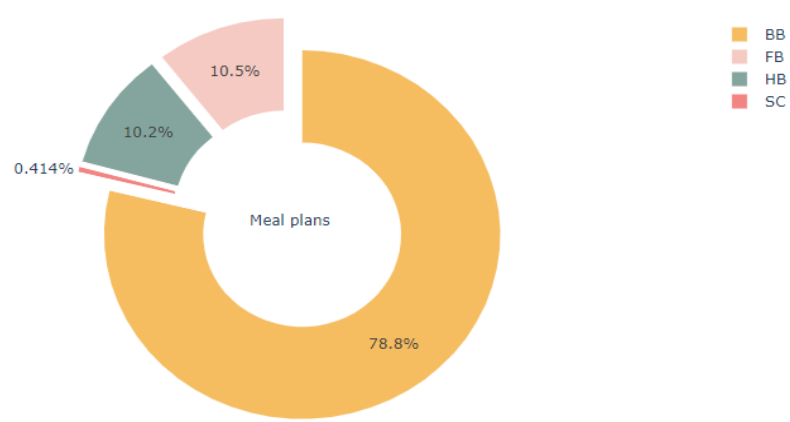



```
# Free or Paid Courses - Countplot
fig, ax = plt.subplots(figsize=(7,5), dpi=100)
ax = sns.countplot(data=courses, x='is_paid', palette='magma_r')
ax.set_xticklabels(labels=['Free', 'Paid'])
ax.set_xlabel("Free/Paid courses")
ax.set_ylabel("Number of courses")
ax.set_title("Share of Free and Paid Courses on Udemy")
percentage = round(courses['is_paid'].value_counts() * 100 /len(courses), 2)
patches = ax.patches
for i in range(len(patches)):
  x = patches[i].get_x() + patches[i].get_width()/2
  y = patches[i].get_height()+.05
  ax.annotate('{:.2f}%'.format(percentage[i]), (x, y), ha='center')
```



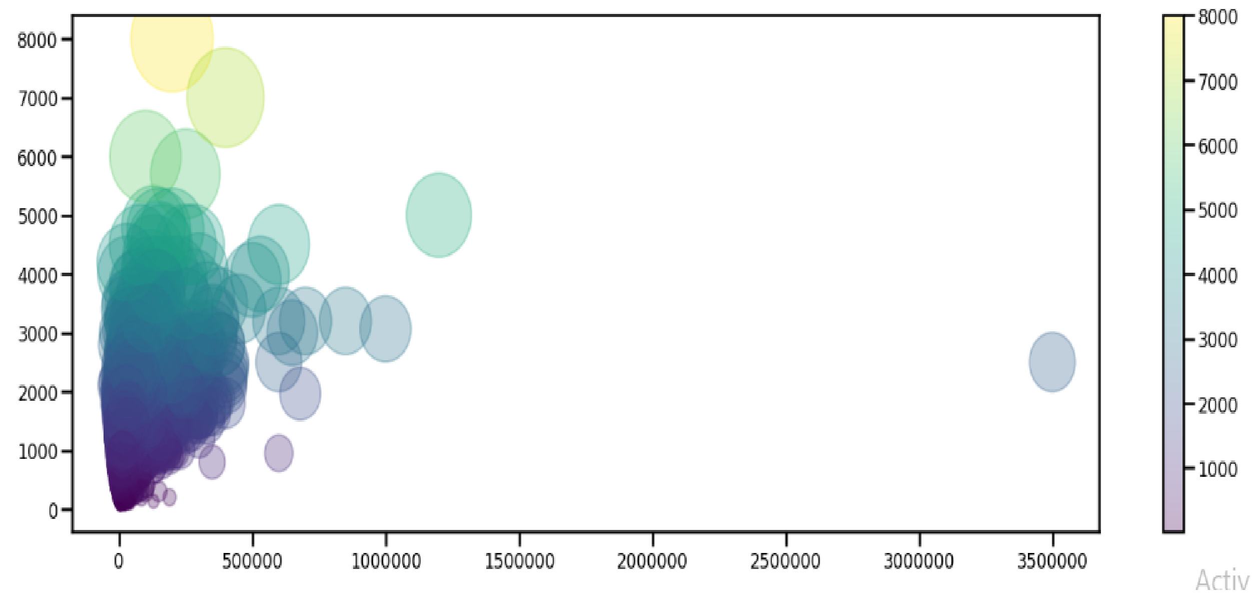

```
#Creating a stripplot to visualize differences in data distribution between hot
els

features = ['lead_time', 'stays_in_weekend_nights', 'stays_in_week_nights', 'ad
ults', 'children', 'babies', 'previous_cancellations', 'previous_bookings_not_can
celed', 'booking_changes', 'adr', 'days_in_waiting_list']

n = 1

sns.set_style('darkgrid')
sns.set(font_scale = 1.2)
plt.figure(figsize = (14, 18))

for feature in features:
    plt.subplot(4,3,n)
    sns.stripplot(x = df['hotel'], y = df[feature], palette = 'summer').set(xlabel =
None, ylabel = None)
    plt.title(f'{feature} strip plot')
    n = n + 1
    plt.tight_layout()
```


```
import plotly.graph_objects as go

labels = confirmed_bookings['meal'].unique()
values = confirmed_bookings['meal'].value_counts()
palette = ["#f6bd60", "#f5cac3", "#84a59d", "#f28482"]

fig = go.Figure(data=[go.Pie(labels = labels,
                             values = values,
                             hole=.5,
                             title = 'Meal plans',
                             legendgroup = True,
                             pull = [0.1, 0.1, 0.1, 0.1]
                            )
                     ]
               )

fig.update_traces(marker = dict(colors = palette));
```



Syed Afroz Ali
Follow me on LinkedIn for more: https://www.linkedin.com/in/syed-afroz-70939914/

```
x = rent_df["Rent"]
y = rent_df["Size"]
colors = rent_df["Size"]
sizes = rent_df["Size"]

plt.figure(figsize = (25, 8))
plt.ticklabel_format(style = 'plain')
plt.scatter(x, y, c = colors, s = sizes, alpha = 0.3, cmap = 'viridis')
plt.colorbar();
```



Syed Afroz Ali
Follow me on LinkedIn for more: https://www.linkedin.com/in/syed-afroz-70939914/