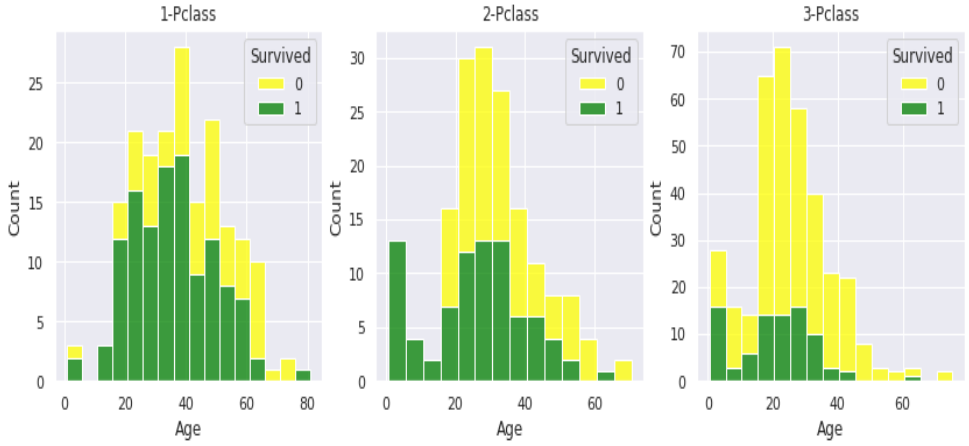
Mastering Data Visualization Techniques (Part 3)

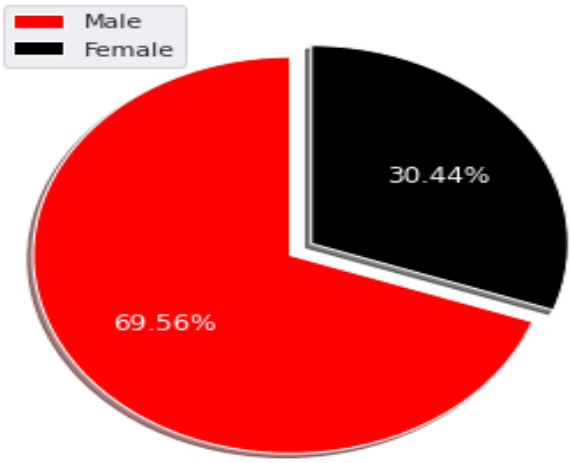
Prepared by: Syed Afroz Ali

```
plot , ax = plt.subplots(1 , 3 , figsize=(14,4))
sns.histplot(data = train_data.loc[train_data["Pclass"]==1] , x = "Age" , hue = "Survived",binwidth=5,ax = ax[0],palette = sns.color_palette(["yellow" , "green"]),multiple = "stack").set_title("1-Pclass")
sns.histplot(data = train_data.loc[train_data["Pclass"]==2] , x = "Age" , hue = "Survived",binwidth=5,ax = ax[1],palette = sns.color_palette(["yellow" , "green"]),multiple = "stack").set_title("2-Pclass")
sns.histplot(data = train_data.loc[train_data["Pclass"]==3] , x = "Age" , hue = "Survived",binwidth=5,ax = ax[2],palette = sns.color_palette(["yellow" , "green"]),multiple = "stack").set_title("3-Pclass")
plt.show()
```



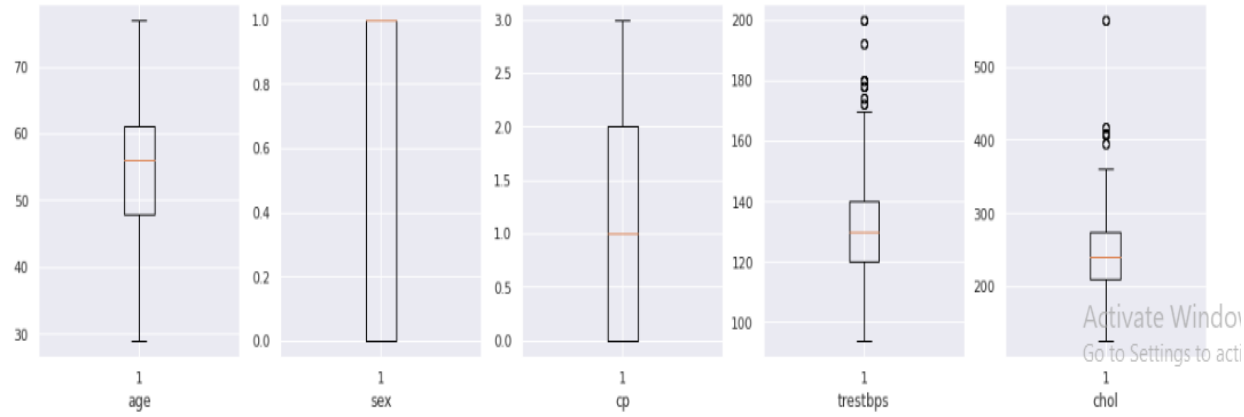
Syed Afroz Ali
Follow me on LinkedIn for more: https://www.linkedin.com/in/syed-afroz-70939914/

```
sex = ["Male", "Female"]
values = data["sex"].value_counts()
color = ["#FF0000", "#000000"]

plt.figure(figsize = (5, 7))
plt.pie(values, labels = sex, colors = color, explode = (0.1, 0),
textprops = {"color":"w"}, autopct = "%.2f%%", shadow = Tru
e, startangle = 90)

plt.legend();
```

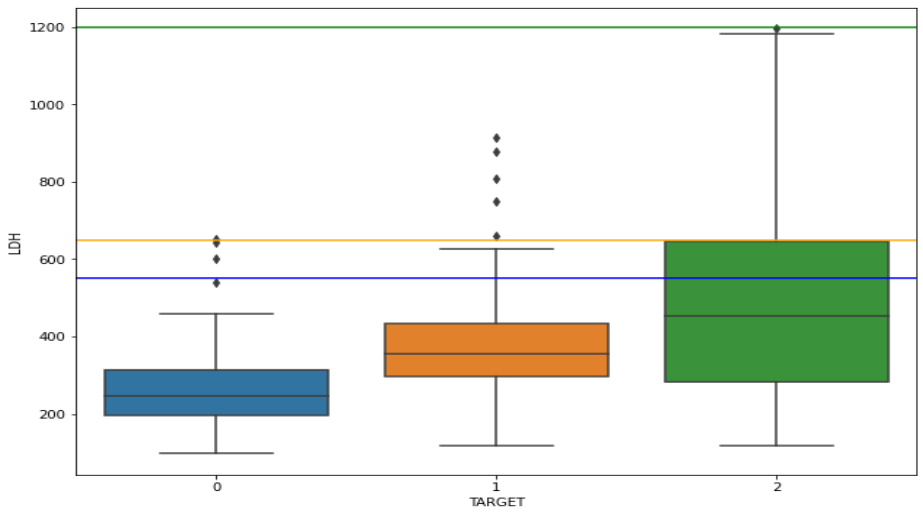



```
# Plotting Outliers
col = 1
plt.figure(figsize = (20, 15))
for i in data.columns:
    if col < 14:
        plt.subplot(3, 5, col)
        plt.boxplot(data[i])
        plt.xlabel(i)
    col = col + 1
```

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

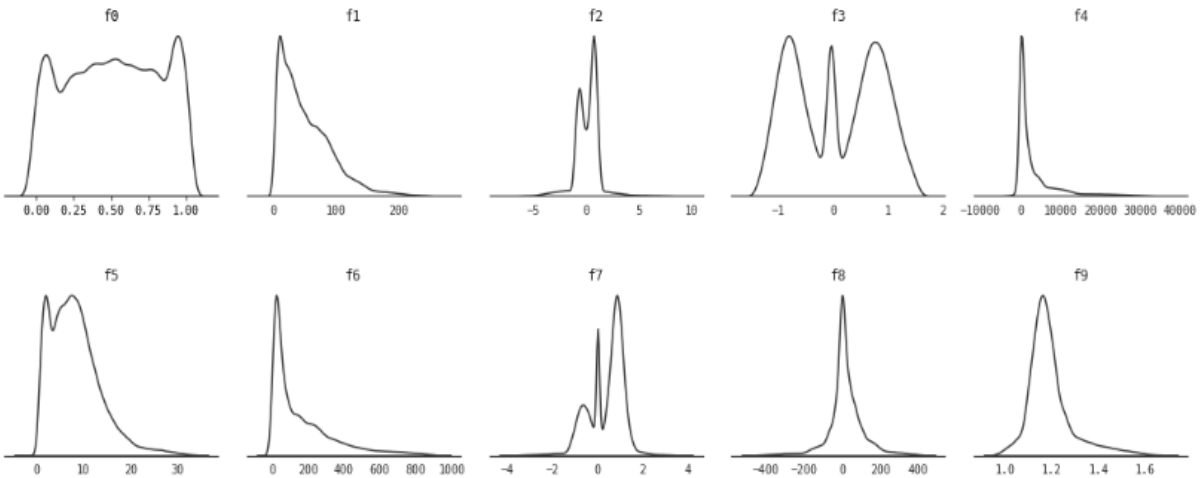



```
fig = plt.figure( figsize=(8, 6))
ax = fig.add_axes([0,0,1,1])
sns.boxplot(ax=ax, data=df, x='TARGET', y='LDH')#,fli
erprops=dict(marker='o', markersize=6),fliersize=2)

ax.axhline(y=550,color='b')
ax.axhline(y=650,color='orange')
ax.axhline(y=1200,color='g')
```



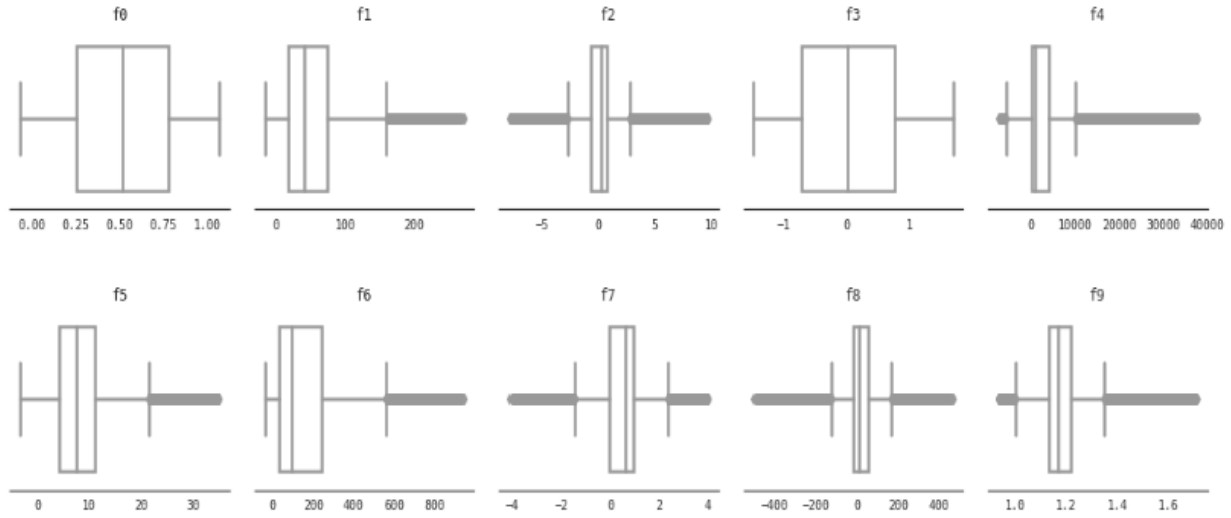
Syed Afroz Ali
Follow me on LinkedIn for more: https://www.linkedin.com/in/syed-afroz-70939914/

```
fig = plt.figure(figsize = (15, 60))
for i in range(len(train.columns.tolist()[:100])):
    plt.subplot(20,5,i+1)
    sns.set_style("white")
    plt.title(train.columns.tolist()[:100][i], size = 12, fontname
= 'monospace')
    a = sns.kdeplot(train[train.columns.tolist()[:100][i]], shade
= True, alpha = 0.9, linewidth = 1.5, facecolor=(1, 1, 1, 0), ed
gecolor=".2")
    plt.ylabel('')
    plt.xlabel('')
    plt.xticks(fontname = 'monospace')
    plt.yticks([])
    for j in ['right', 'left', 'top']:
        a.spines[j].set_visible(False)
        a.spines['bottom'].set_linewidth(1.2)

fig.tight_layout(h_pad = 3)
plt.show()
```



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

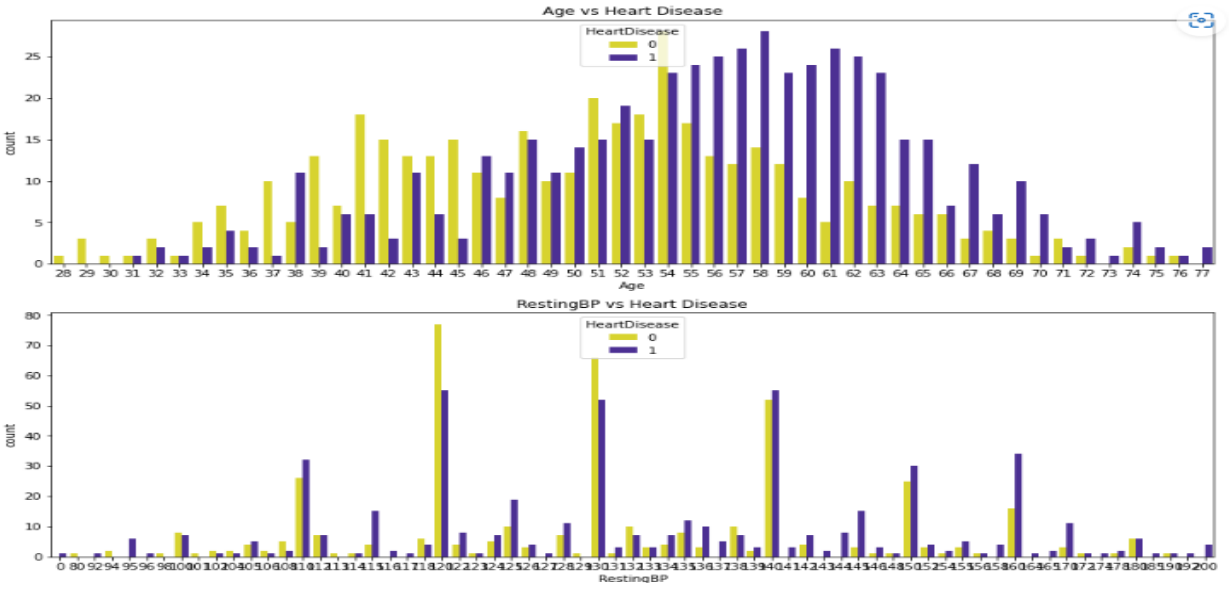
```
fig = plt.figure(figsize = (15, 60))
for i in range(len(train.columns.tolist()[:100])):
    plt.subplot(20,5,i+1)
    sns.set_style("white")
    plt.title(train.columns.tolist()[:100][i], size = 12, fontname
= 'monospace')
    a = sns.boxplot(train[train.columns.tolist()[:100][i]], linewi
dth = 2.5,color = 'white')
    plt.ylabel('')
    plt.xlabel('')
    plt.xticks(fontname = 'monospace')
    plt.yticks([])
    for j in ['right', 'left', 'top']:
        a.spines[j].set_visible(False)
        a.spines['bottom'].set_linewidth(1.2)

fig.tight_layout(h_pad = 3)
plt.show()
```




```
fig, ax = plt.subplots(nrows = 5,ncols = 1,figsize = (15,30))
```

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

```
colors = ['#F3ED13','#451FA4']
for i in range(len(numerical_features)):
    plt.subplot(5,1,i+1)
    sns.countplot(numerical_features[i],data = data,hue = "HeartDisease",palette = colors)
    title = numerical_features[i] + ' vs Heart Disease'
    plt.title(title);
```



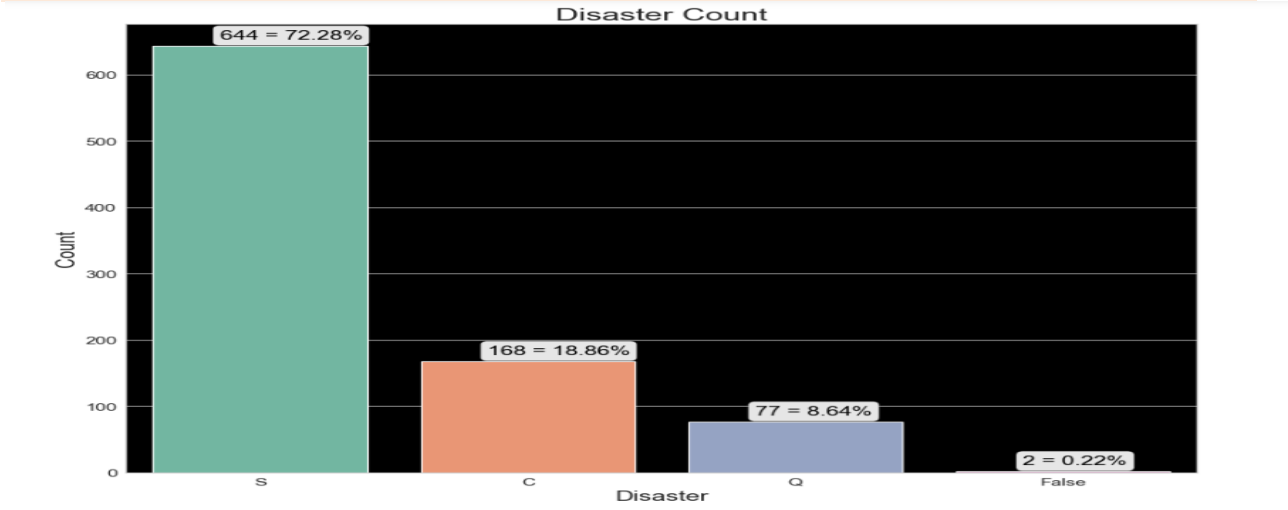
.

```
train.iloc[:, :-1].describe().T.sort_values(by='std' , ascending = False)\
                .style.background_gradient(cmap='GnBu')\
                .bar(subset=["max"], color='#BB0000')\
                .bar(subset=["mean",], color='green')
```

	count	mean	std	min	25%	50%	75%	max
FoodCourt	8510.00	458.08	1611.49	0.00	0.00	0.00	76.00	29813.00
VRDeck	8505.00	304.85	1145.72	0.00	0.00	0.00	46.00	24133.00
Spa	8510.00	311.14	1136.71	0.00	0.00	0.00	59.00	22408.00
RoomService	8512.00	224.69	666.72	0.00	0.00	0.00	47.00	14327.00
ShoppingMall	8485.00	173.73	604.70	0.00	0.00	0.00	27.00	23492.00
Age	8514.00	28.83	14.49	0.00	19.00	27.00	38.00	79.00

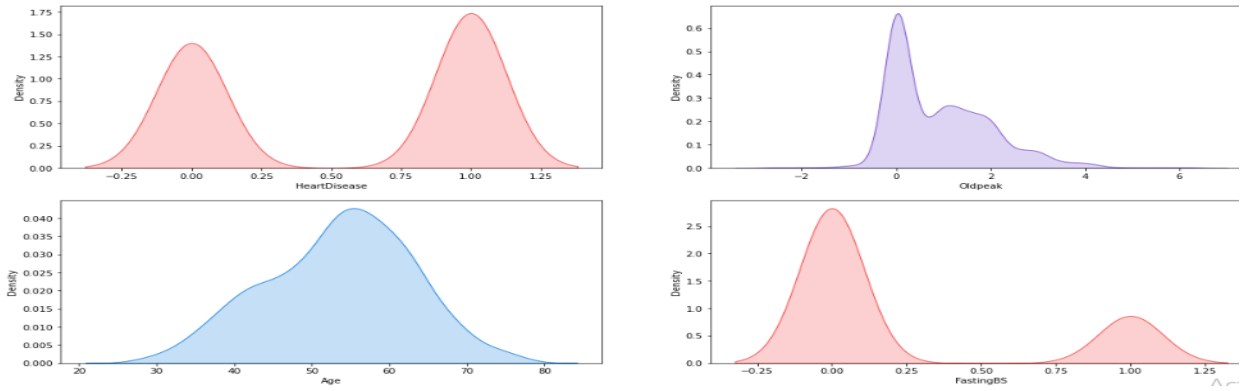
Syed Afroz Ali
Follow me on LinkedIn for more: https://www.linkedin.com/in/syed-afroz-70939914/

```
plt.figure(figsize = (15, 12))
ax = plt.axes()
ax.set_facecolor('black')
ax = sns.countplot(x = 'Embarked', data = titanic, palette = [
custom_colors[2], custom_colors[1]], edgecolor = 'white', lin
ewidth = 1.2)
plt.title('Disaster Count', fontsize = 25)
plt.xlabel('Disaster', fontsize = 20)
plt.ylabel('Count', fontsize = 20)
ax.xaxis.set_tick_params(labelsize = 15)
ax.yaxis.set_tick_params(labelsize = 15)
bbox_args = dict(boxstyle = 'round', fc = '0.9')
for p in ax.patches:
        ax.annotate('{:.0f} = {:.2f}%'.format(p.get_height(), (p.get
_height() / len(titanic['Embarked'])) * 100), (p.get_x() + 0.25, p
.get_height() + 10),
                    color = 'black',
                    bbox = bbox_args,
                    fontsize = 18)
plt.show()
```



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

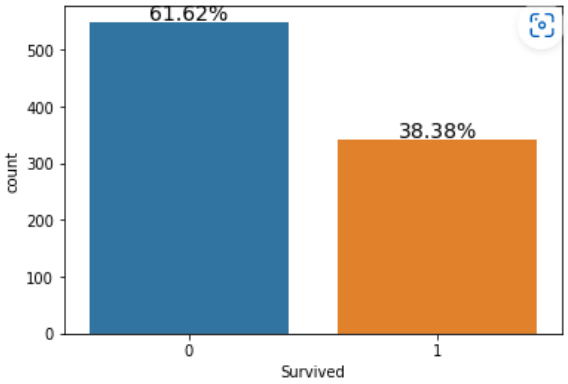
```
#Plotting the distributions of the numerical variables
color_plot = ['#de972c','#74c91e','#1681de','#e069f5','#f54545','#f0ea46','#7950cc']

fig,ax = plt.subplots(4,2,figsize=(20,20))
sns.kdeplot(df['HeartDisease'],color=np.random.choice(color_plot), ax=ax[0][0], shade=True)
sns.kdeplot(df['Oldpeak'],color=np.random.choice(color_plot), ax=ax[0][1], shade=True)
sns.kdeplot(df['Age'],color=np.random.choice(color_plot), ax=ax[1][0], shade=True)
sns.kdeplot(df['FastingBS'],color=np.random.choice(color_plot), ax=ax[1][1], shade=True)
sns.kdeplot(df['RestingBP'],color=np.random.choice(color_plot), ax=ax[2][0],shade=True)
sns.kdeplot(df['Cholesterol'],color=np.random.choice(color_plot), ax=ax[2][1], shade=True)
sns.kdeplot(df['MaxHR'],color=np.random.choice(color_plot), ax=ax[3][0],shade=True)
fig.delaxes(ax[3][1])
```



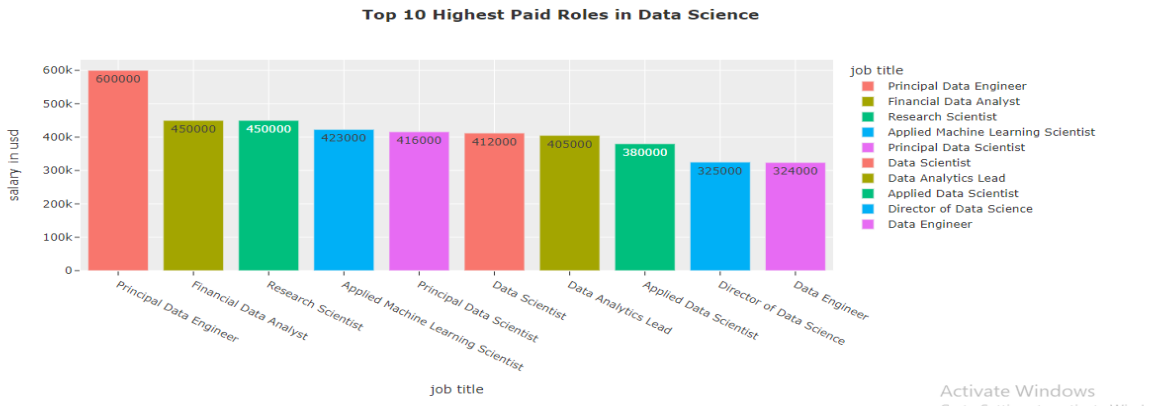

```
s = sns.countplot(x = 'Survived',data = train)
sizes=[]
for p in s.patches:
    height = p.get_height()
```

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

```
    sizes.append(height)
    s.text(p.get_x()+p.get_width()/2.,
           height + 3,
           '{:1.2f}%'.format(height/len(train)*100),
           ha="center", fontsize=14)
```



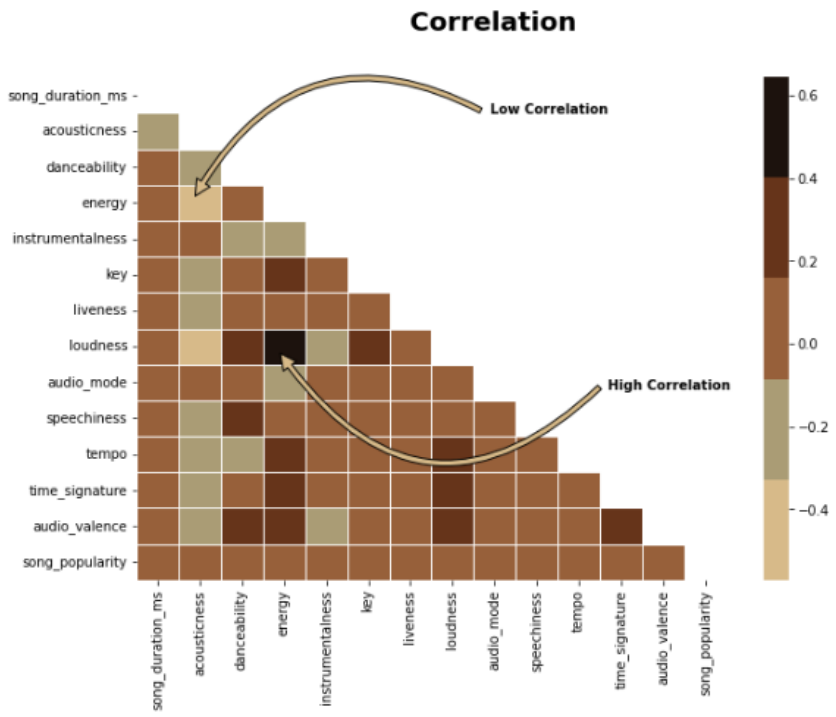

```
z=df['job_title'].value_counts().head(10)
fig=px.bar(z,x=z.index,y=z.values,color=z.index,text=z.values,labels={'index':'job title','y':'count','text':'count'},template='seaborn',title='<b> Top 10 Popular Roles in Data Sceince')
fig.show()
```



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

```
hm= df.drop('id', axis =1)
mask = np.zeros_like(hm.corr(), dtype=np.bool)
mask[np.triu_indices_from(mask)]= True

plt.suptitle('Correlation', size = 20, weight='bold')

ax = sns.heatmap(hm.corr(), linewidths = 0.9, linecolor = 'whi
te', cbar = True,mask=mask, cmap=heatmap)

ax.annotate('Low Correlation',
                fontsize=10,fontweight='bold',
                xy=(1.3, 3.5), xycoords='data',
                xytext=(0.6, 0.95), textcoords='axes fraction',
                arrowprops=dict(
                facecolor=heatmap[0], shrink=0.025,
                connectionstyle='arc3, rad=0.50'),
                horizontalalignment='left', verticalalignment='to
p'
)
ax.annotate('High Correlation',
                fontsize=10,fontweight='bold',
                xy=(3.3, 7.5), xycoords='data',
                xytext=(0.8, 0.4), textcoords='axes fraction',
                arrowprops=dict(
                facecolor=heatmap[0], shrink=0.025,
                connectionstyle='arc3, rad=-0.6'),
                horizontalalignment='left', verticalalignment='to
p'
)
plt.show()
```

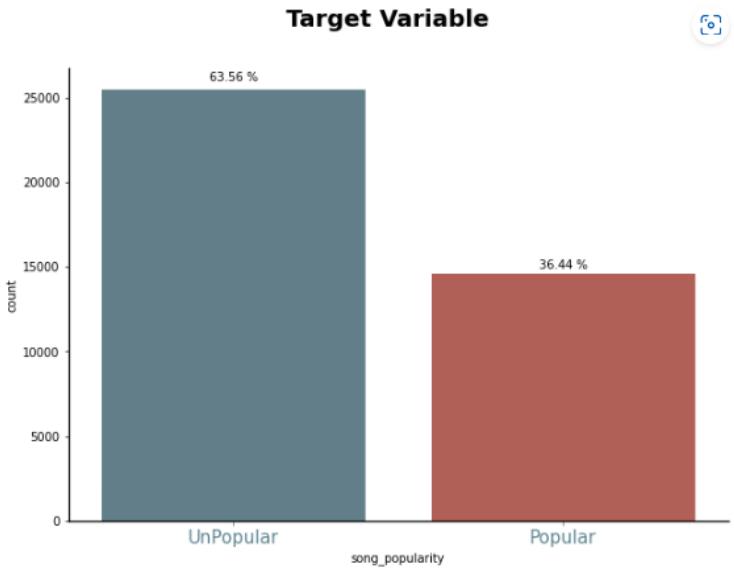
Syed Afroz Ali
Follow me on LinkedIn for more: https://www.linkedin.com/in/syed-afroz-70939914/




```
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=them
e)
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()
```

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

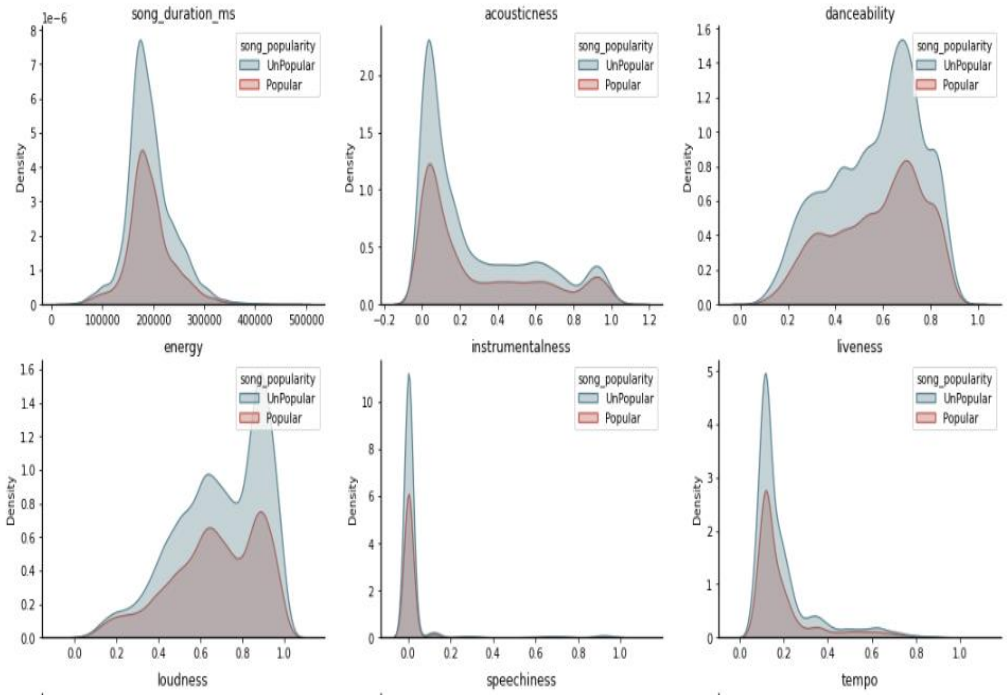
```
    a.annotate(f'{height/df.shape[0]*100} %', (x + width/2, y + h
eight*1.02), ha='center')

plt.show()
```




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

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/

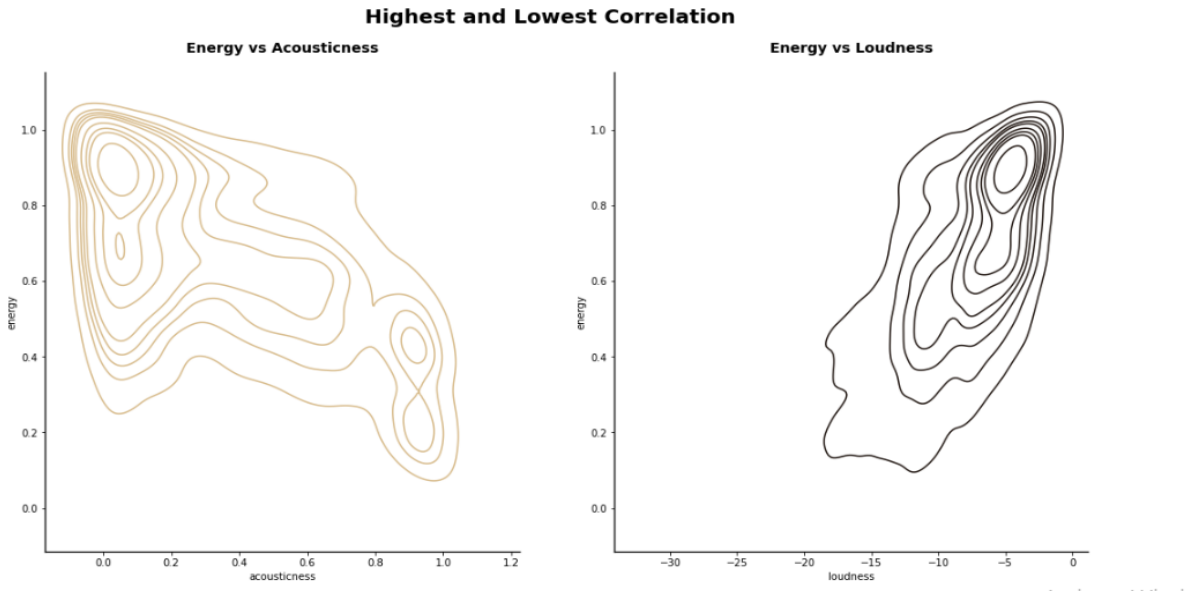
```
for i in cont:
        plt.suptitle('Distribution of Features', size = 20, weight='bold')
        plt.subplot(a, b, c)
        A=sns.kdeplot(data= df, x=i,hue=song_popularity,palette=theme[:-2], linewidth = 1.3,shade=True, alpha=0.35)
        plt.title(i)
        plt.xlabel(" ")
        c = c + 1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




```
#plotting
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(18, 9))
fig.suptitle(' Highest and Lowest Correlation ', size = 20, weight='bold')
```

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

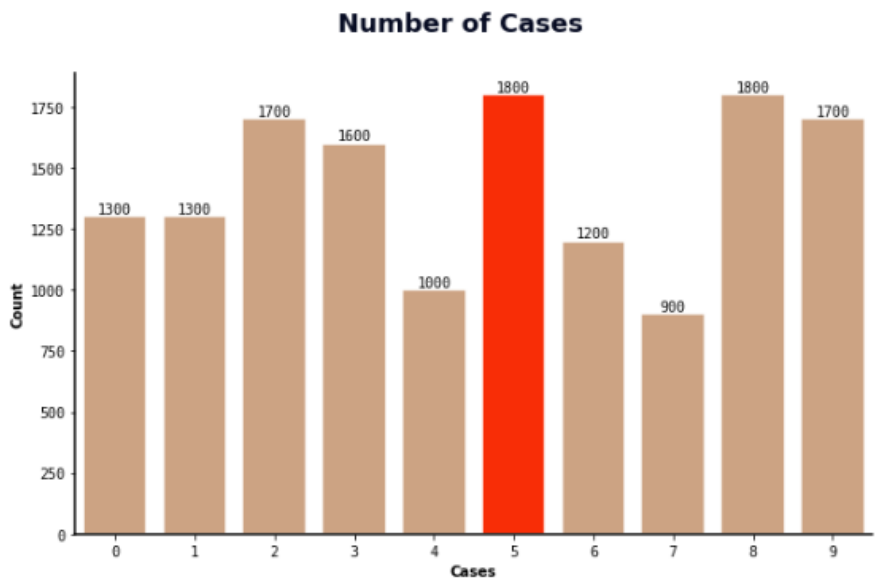
```
axs = [ax1, ax2]

#kdeplot
sns.kdeplot(data=df, y='energy', x='acousticness', ax
=ax1, color=heatmap[0])
ax1.set_title('Energy vs Acousticness', size = 14, wei
ght='bold', pad=20)

#kdeplot
sns.kdeplot(data=df, y='energy', x='loudness', ax=ax2
, color=heatmap[4])
ax2.set_title('Energy vs Loudness', size = 14, weight=
'bold', pad=20);
```



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

```
colors = ["#e9d9c8","#cca383","#070c23","#f82d06",
"#e8c195","#cd7551","#a49995","#a3a49c","#6c7470
"]
sns.palplot(sns.color_palette(colors))


#plot
A = sns.countplot(train_df['case_num'],
            color=colors[1],
            edgecolor='white',
            linewidth=1.5,
            saturation=1.5)

#Patch
patch_h = []
for patch in A.patches:
    reading = patch.get_height()
    patch_h.append(reading)

idx_tallest = np.argmax(patch_h)
A.patches[idx_tallest].set_facecolor(colors[3])

#Lables
plt.ylabel('Count', weight='semibold', fontname = 'Georgia')
plt.xlabel('Cases', weight='semibold', fontname = 'Georgia')
plt.suptitle('Number of Cases', fontname = 'Georgia', weight=
'bold', size = 18, color = colors[2])
A.bar_label(A.containers[0], label_type='edge')

plt.show()
```

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

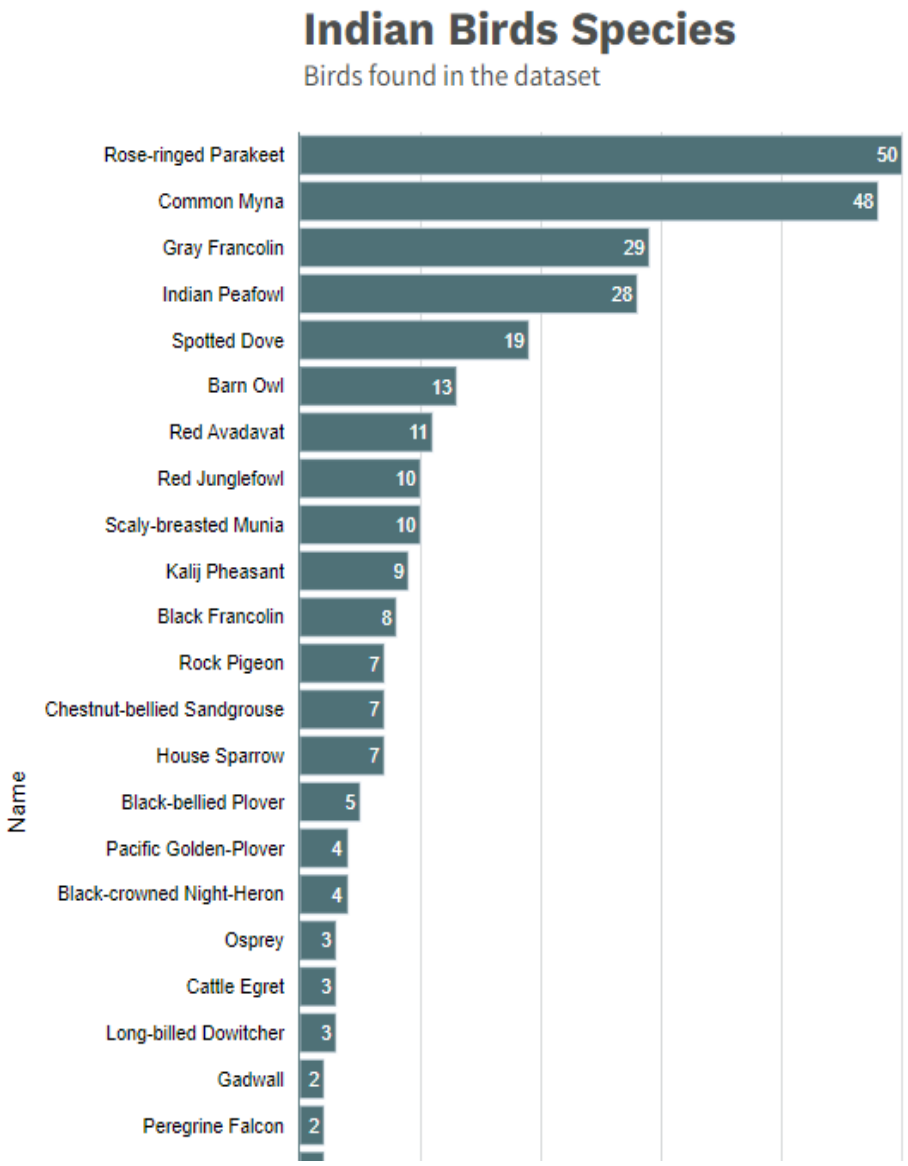



```
Datas = india_df["common_name"].value_counts().reset_inde
x().sort_values(by='common_name')

# Creating the bar chart
trace = go.Bar(
    y = Datas["index"],
    x = Datas["common_name"],
    orientation = "h",
    marker_color= "#4F7177",
    text = Datas["common_name"],
)

layout = dict(
    width = 600,
    height= 1000,
    plot_bgcolor = "#FFFFFF",
    font=dict(family='Arial',
              size=12,
              color='black'),
```

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

```
   margin = dict(
      l=0,
      r=0,
      b=100,
      t=100,
      pad=0
   ),
   xaxis = dict(showline=True, linewidth=1.45, linecolor="#4
F7177",gridcolor='#D5D7D8',
               #griddash='dot',
               title_text='Counts'),
   yaxis = dict(showline=True, linewidth=1.45, linecolor="#4F
7177",ticksuffix = "  ",title_text='Name'),
   bargap = 0.15,hoverlabel_bgcolor="#4F7177",hovermode="
x"
)

fig = go.Figure(data = trace, layout = layout)
fig.layout.xaxis.fixedrange = True
fig.layout.yaxis.fixedrange = True

#text
texter("Indian Birds Species",0.000,1.10,28,"Work Sans")
texter("Birds found in the dataset",0.000,1.06,18,"Source Sa
ns Pro")
texter("heyRobin!",1.00,-0.06,16,"Playfair Display")

fig.show()
```

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

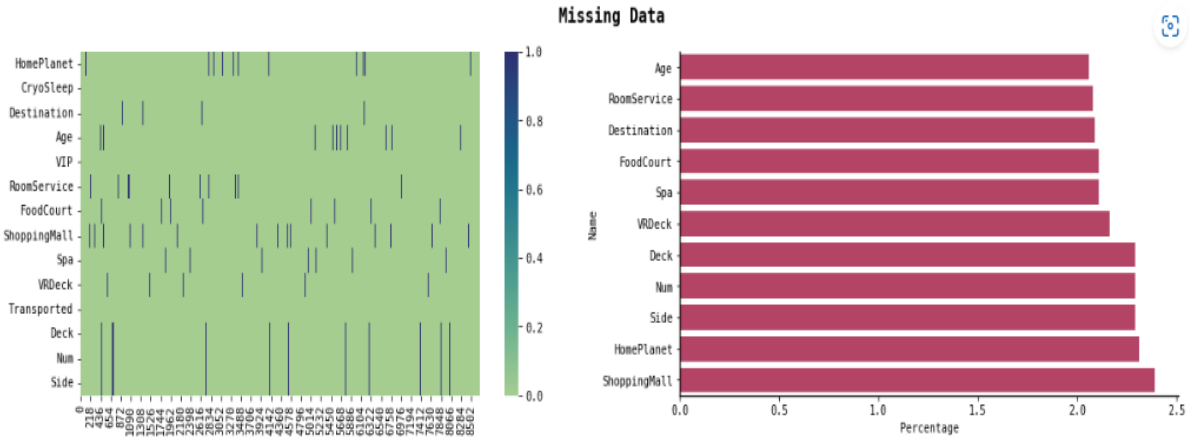


Missing Values:

```
fig, axes = plt.subplots(1,2, figsize=(20,5))
fig.suptitle('Missing Data', size = 15, weight='bold')

#first plot
sns.heatmap(train.isna().transpose(),
```

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

```
        cmap="crest",ax=axes[0])

#missing data
missing = round(train.isna().sum()/train.shape[0]* 100
,2)
missing = missing[missing>0].sort_values().to_frame(
)
missing.columns = ['Percentage']
missing.index.names = ['Name']
missing = missing.reset_index()
sns.barplot(data = missing, y= 'Name', x = 'Percentag
e',ax=axes[1],color=pal[0])
plt.show()
```



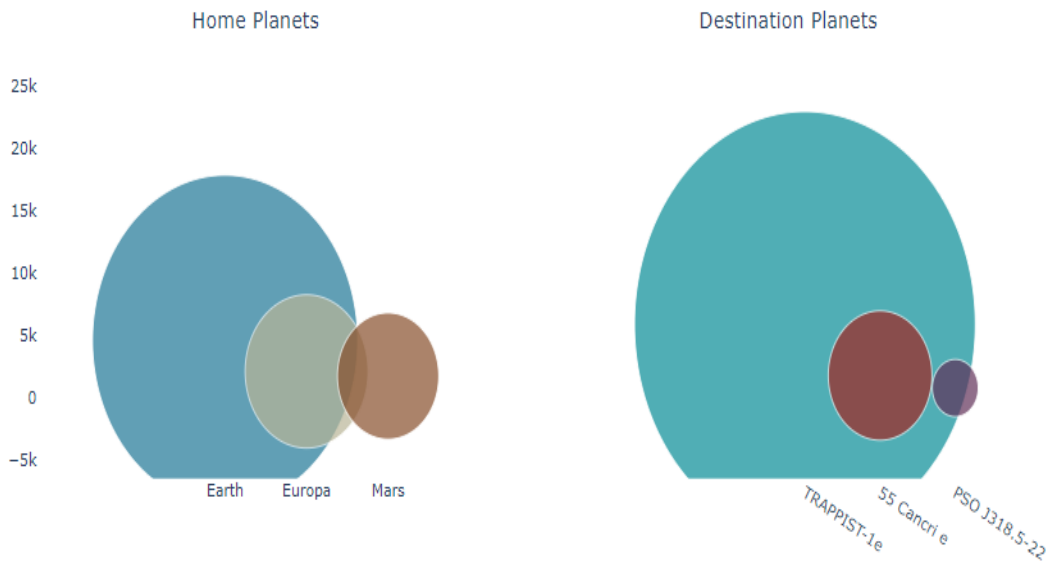

```
from plotly.subplots import make_subplots
import plotly.graph_objects as go
#Data
cnt_srshp =train['HomePlanet'].value_counts()
cnt_srsdes =train['Destination'].value_counts()
```

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

```
fig = make_subplots(rows=2, cols=2, shared_yaxes=True,
    subplot_titles=("Home Planets","Destination Planets","VIP
","CryoSleep"))

#figure1
fig.add_trace(go.Scatter(
    x=cnt_srshp.index,
    y=cnt_srshp.values,
    mode='markers',
    marker=dict(
        sizemode = 'diameter',
        sizeref = 20,
        size = cnt_srshp.values,
        color = ['#1D7595','#B9B596','#864D29'])), 1, 1)
#figure2
fig.add_trace(go.Scatter(
    x=cnt_srsdes.index,
    y=cnt_srsdes.values,
    mode='markers',
    marker=dict(
        sizemode = 'diameter',
        sizeref = 20,
        size = cnt_srsdes.values,
        color = ['#048B95','#A1231F','#602F58'])),
            1, 2)

#figure3
fig.add_trace(go.Histogram(x=train["VIP"],
                           marker=dict(color=pal)),
              row=2, col=1)
```

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

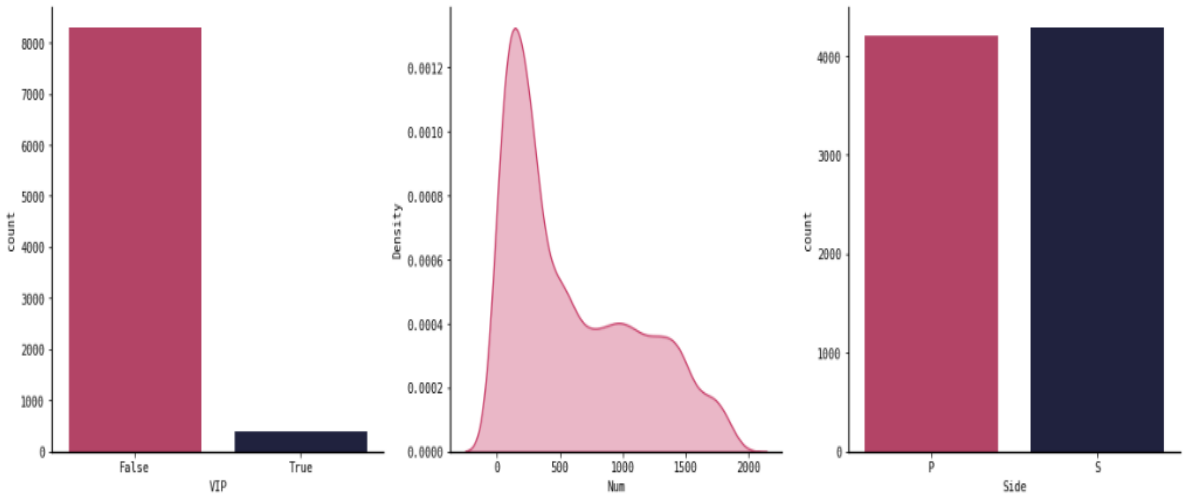
```
#figure4
fig.add_trace(go.Histogram(x=train["CryoSleep"],
                           marker=dict(color=pal)),
              row=2, col=2)

fig.update_layout(height=1000,width=1000, coloraxis=dict(c
olorscale='Bluered_r'), showlegend=False,
    title_x=0.9,
    titlefont=dict(size = 2, color='black', family='Space Mono'),
    plot_bgcolor='rgba(0,0,0,0)'
)
fig.show()
```




```
fig, axes = plt.subplots(1,3, figsize=(20,6))
```

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

```
sns.countplot(train["VIP"], ax=axes[0],palette =pal)
sns.kdeplot(train["Num"],linewidth = 1.3,shade=True,
alpha=0.35, ax=axes[1],color=pal[0])
sns.countplot(train["Side"], ax=axes[2],palette =pal)

plt.show()
```



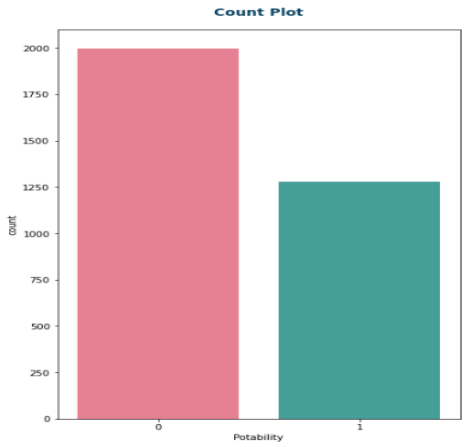

```
import matplotlib as mlb
import matplotlib.image as mpimg
from matplotlib.offsetbox import AnnotationBbox, OffsetImage

#plotting
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(16, 11))
fig.suptitle(' Potablity of Water Quality ', size = 26, color = th
eme[3], weight='bold')
axs = [ax1, ax2]

#Count-Plot
sns.countplot(water_df['Potability'], ax=ax1, palette='husl')
```

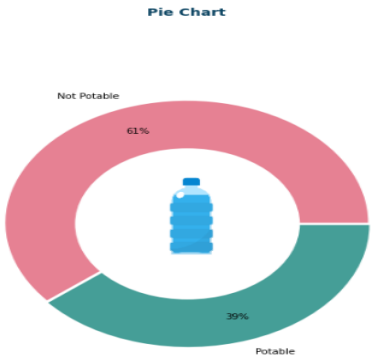
Syed Afroz Ali
Follow me on LinkedIn for more: https://www.linkedin.com/in/syed-afroz-70939914/

```
ax1.set_title('Count Plot', size = 14, color = theme[3], weight
='bold', pad=20)

#Data-2
names = ["Not Potable", "Potable"]
values = water_df['Potability'].value_counts()
colors = ["#E68193","#459E97"]
explode = (0.01, 0.01)

#Doughnut-chart
ax2.pie(x= values,labels =names, colors=colors,autopct='%1.
0f%%', pctdistance=0.8,explode=explode)

#draw-circle
centre_circle = plt.Circle((0,0),0.62,fc='white')
ax2.add_artist(centre_circle)
ax2.axis('equal')

ax2.set_title('Pie Chart', size = 14, color = theme[3], weight='
bold', pad=20)

#Image

path = mpimg.imread('../input/water/water bottle.png')
imagebox = OffsetImage(path , zoom=0.3)
xy = (0.5, 0.7)
ab = AnnotationBbox(imagebox, xy, frameon=False, pad=1, x
ybox=(0.02, 0.05))
ax2.add_artist(ab)
```

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

```
plt.subplots_adjust(left=None, bottom=None, right=None, top
=0.8, wspace=0.4, hspace=None);
```





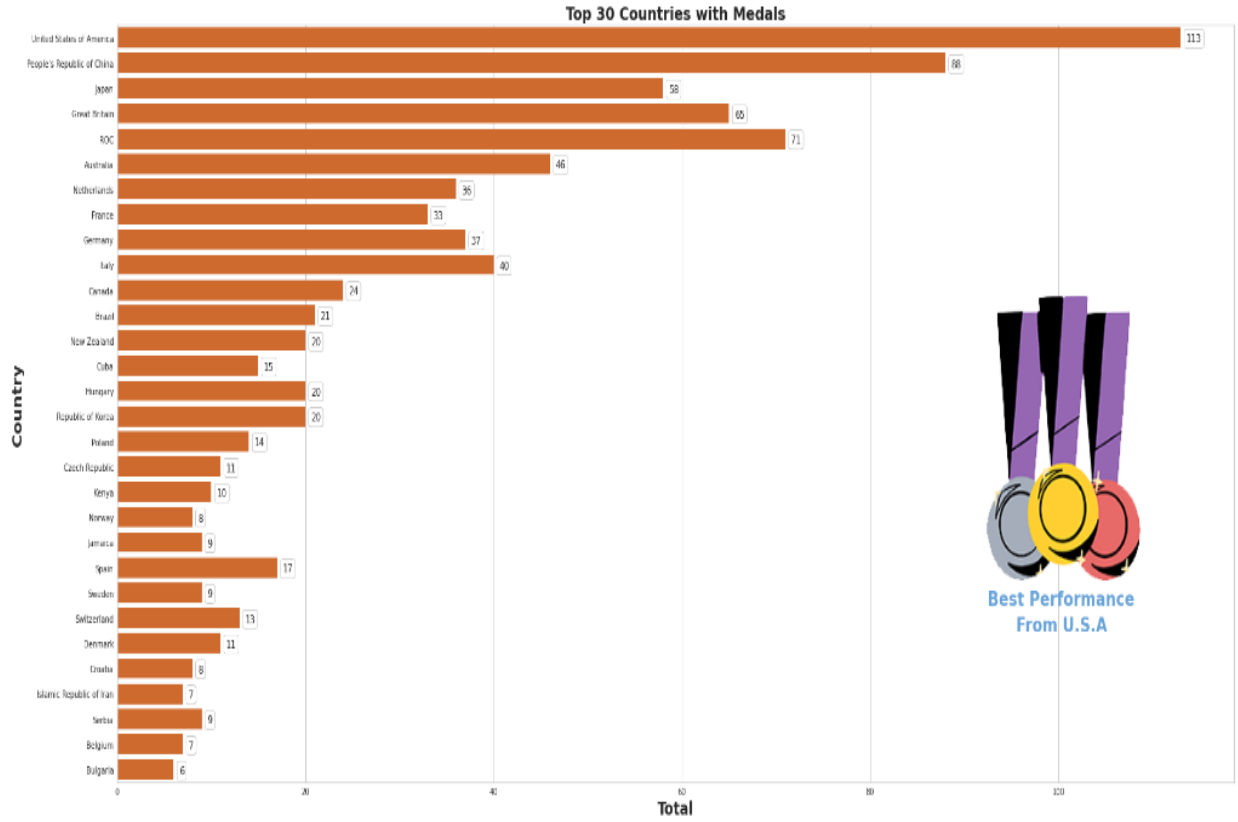

```
#Figure with Image
import matplotlib as mlb
import matplotlib.image as mpimg
from matplotlib.offsetbox import AnnotationBbox, Off
setImage

plt.figure(figsize=(27,15));
ax = sns.barplot(y ='Country',
                 x='Total',
                 data=medals[:30],
                 color=olympics_col[3],
                 zorder=2,
                 linewidth=0,
                 orient='h',
                 saturation=1,
                 alpha=1,
                 )
```

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

```
#Lables
ax.set_xlabel("Total",fontsize=20, weight='bold')
ax.set_ylabel("Country",fontsize=20, weight='bold')
ax.tick_params(labelsize=10, width=0.5, length=1.5)
plt.title("Top 30 Countries with Medals",size=20,weig
ht='bold')

#Patches
for a in ax.patches:
    value = f'{a.get_width():.0f}'
    x = a.get_x() + a.get_width() + 0.60
    y = a.get_y() + a.get_height() / 1.8
    ax.text(x, y, value, ha='left', va='center', fontsize=12,
            bbox=dict(facecolor='none', edgecolor='black', box
style='round', linewidth=0.2))

#image
path = mpimg.imread('../input/font-worksans/medal-cr
op.png')
imagebox = OffsetImage(path , zoom=1.6)
xy = (0.5, 0.7)
ab = AnnotationBbox(imagebox, xy, frameon=False, p
ad=1, xybox=(100.5, 16))

ax.add_artist(ab)
ax.text(x = 92.5, y = 22.5, s = 'Best Performance', fon
tsize=22, weight = 'bold',color=olympics_col[1])
```

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

```
ax.text(x = 95.5, y = 23.5, s = 'From U.S.A', fontsize=22, weight = 'bold',color=olympics_col[1]);
```

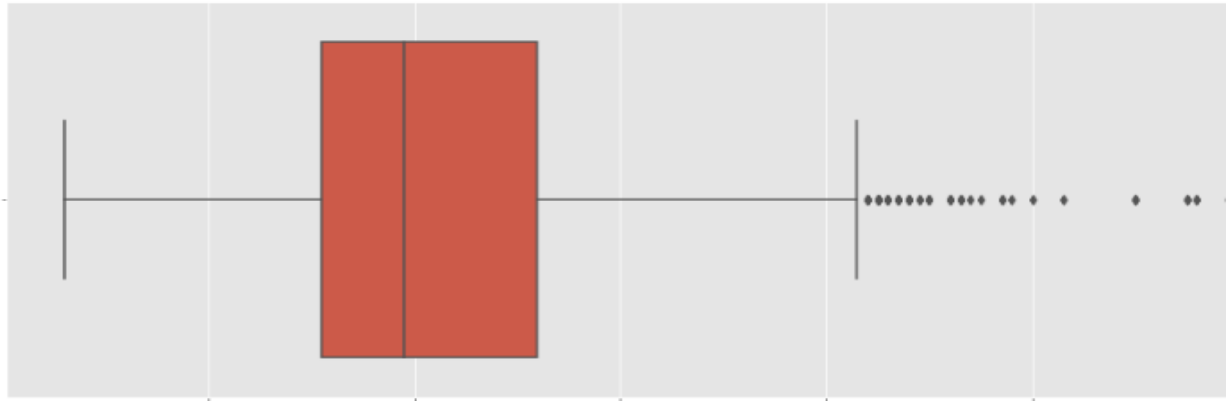



```
for col in numeric_features[1:]:
    fig = plt.figure(figsize=(9, 6))
    ax = fig.gca()
    feature = data[col]
    feature.hist(bins=50, ax = ax)
    ax.axvline(feature.mean(), color='magenta', linestyle='dashed', linewidth=2)
    ax.axvline(feature.median(), color='cyan', linestyle='dashed', linewidth=2)
    ax.set_title(col)
```

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

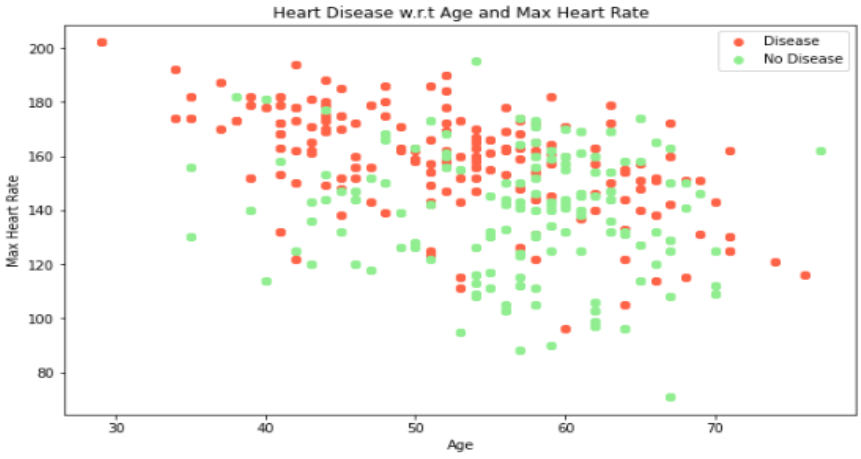
```
plt.show()
```


```
data1=mydata[['Parameter 1']]
for i in data1.columns:
    plt.figure(figsize=(15,6))
    sns.boxplot(data1[i])
    plt.xticks(rotation=90)
    plt.show()
```



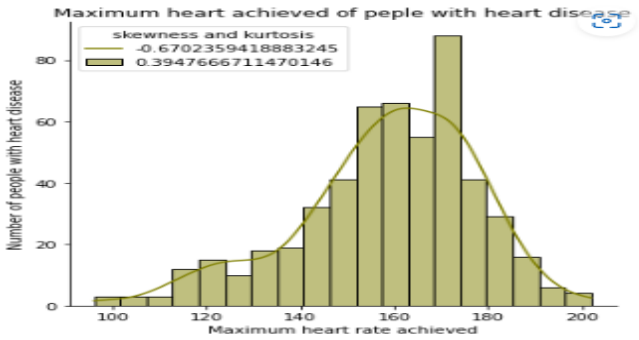
Syed Afroz Ali
Follow me on LinkedIn for more: https://www.linkedin.com/in/syed-afroz-70939914/

```
# Creating a figure
plt.figure(figsize=(10,6))

#plotting the values for people who have heart disease
plt.scatter(df.age[df.target==1],
            df.thalach[df.target==1],
            c="tomato")

#plotting the values for people who doesn't have heart disea
se
plt.scatter(df.age[df.target==0],
            df.thalach[df.target==0],
            c="lightgreen")

# Addind info
plt.title("Heart Disease w.r.t Age and Max Heart Rate")
plt.xlabel("Age")
plt.legend(["Disease", "No Disease"])
plt.ylabel("Max Heart Rate");
```



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

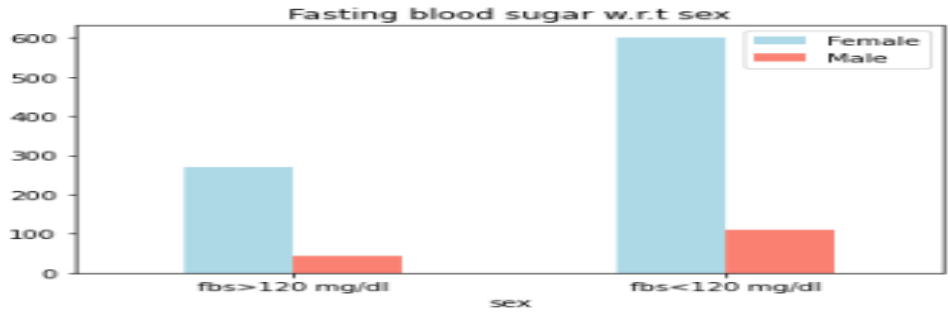
```
sns.displot(x = df.thalach[df.target==1], data = df, kde = True, color= 'olive')#
skewness=str(df.thalach[df.target==1].skew())
kurtosis=str(df.thalach[df.target==1].kurt())
plt.legend([skewness,kurtosis],title=("skewness and kurtosis"))
plt.title("Maximum heart achieved of peple with heart disease")
plt.xlabel("Maximum heart rate achieved")
plt.ylabel("Number of people with heart disease");
```




```
pd.crosstab(df.sex, df.fbs)
```

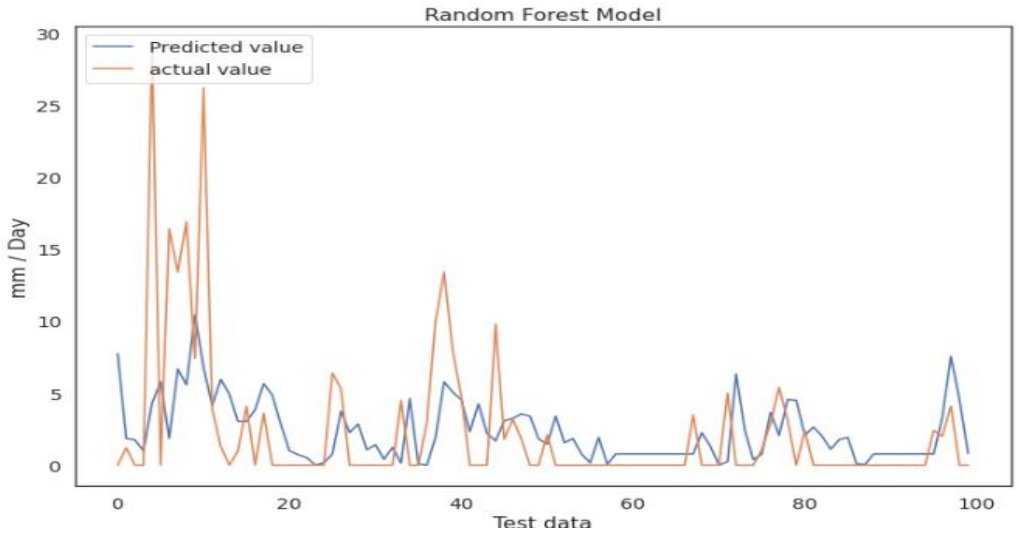
fbs	0	1
sex		
0	270	42
1	602	111

```
fig = pd.crosstab(df.sex, df.fbs).plot(kind = 'bar', color = ['lightblue', 'salmon'])
plt.title("Fasting blood sugar w.r.t sex")
fig.set_xticklabels(labels=['fbs>120 mg/dl', 'fbs<120 mg/dl'], rotation=0)
plt.legend(['Female', 'Male']);
```



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

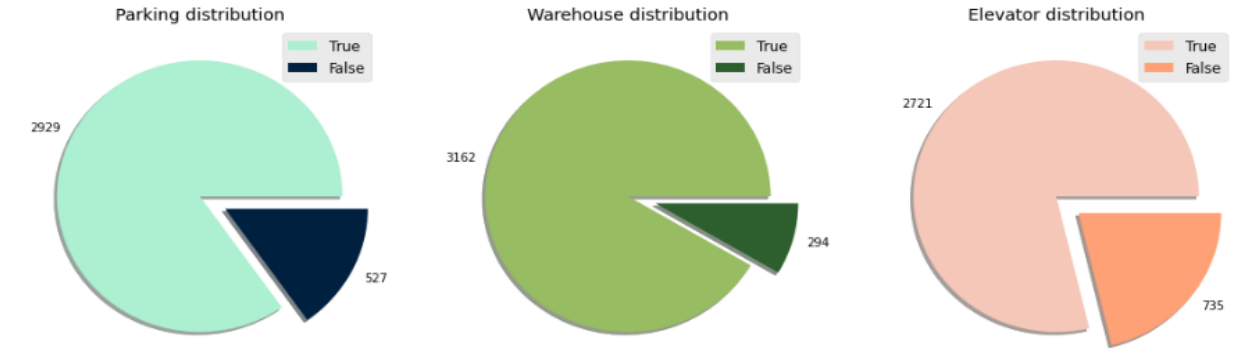
```
pred = rf_model.predict(x_test)
plt.rcParams['figure.figsize'] = (12,8)
plt.plot(pred, label='Predicted value')
plt.plot(y_test, label='actual value')
plt.legend(loc ="upper left")
plt.title('Random Forest Model')
plt.xlabel('Test data')
plt.ylabel('mm / Day')
plt.show()
```




```
fig, ax = plt.subplots(ncols=3, figsize=(18,6))

colors = [['#ADEFD1FF', '#00203FFF'], ['#97BC62FF', '
#2C5F2D'], ['#F5C7B8FF', '#FFA177FF']]
explode = [0, 0.2]
columns = ['Parking', 'Warehouse', 'Elevator']
for i in range(3):
        data = df[columns[i]].value_counts()
```

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

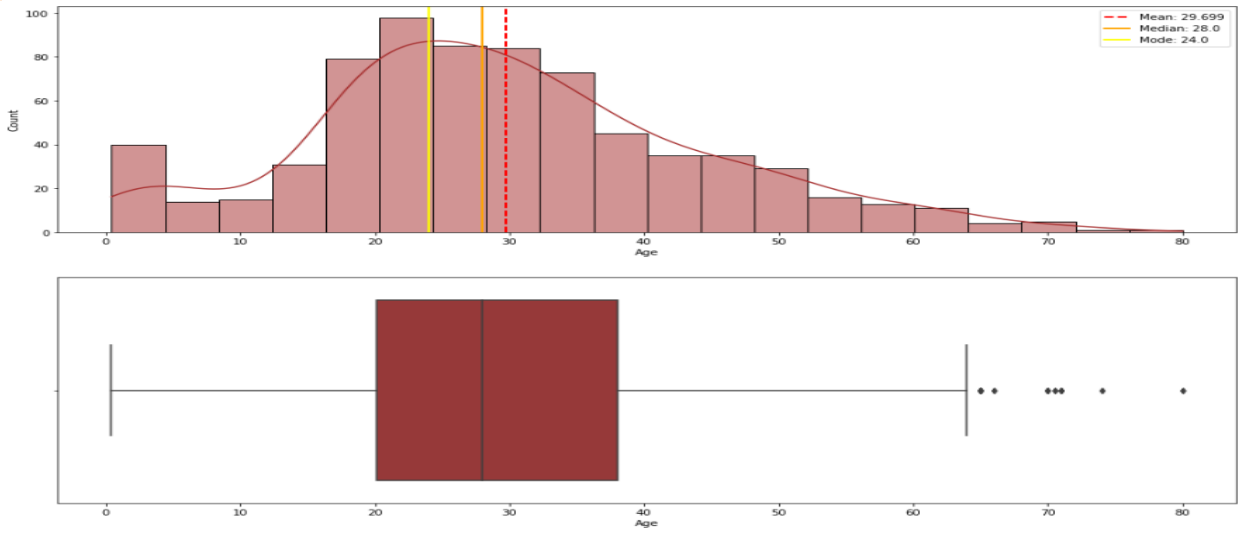
```
        ax[i].pie(data, labels=data.values, explode=expl
ode, colors=colors[i], shadow=True)
        ax[i].legend(labels=data.index, fontsize='large')
        ax[i].set_title('{} distribution'.format(columns[i]))
```



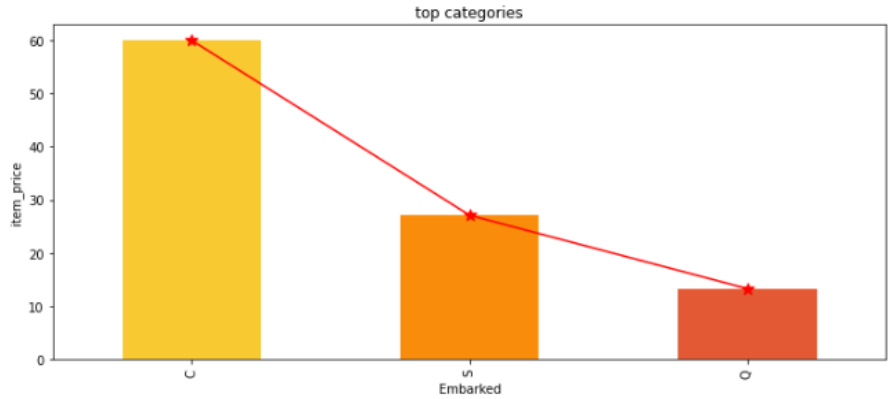
```
def plot_hist(feature):
    fig, ax = plt.subplots(2, 1, figsize=(17, 12))

    sns.histplot(data = titanic[feature], kde = True, ax =
ax[0],color="Brown")

    ax[0].axvline(x = titanic[feature].mean(), color = 'r',
linestyle = '--', linewidth = 2, label = 'Mean:
{}'.format(round(titanic[feature].mean(), 3)))
    ax[0].axvline(x = titanic[feature].median(), color =
'orange', linewidth = 2, label = 'Median:
{}'.format(round(titanic[feature].median(), 3)))
    ax[0].axvline(x = statistics.mode(titanic[feature]), color =
'yellow', linewidth = 2, label = 'Mode:
{}'.format(statistics.mode(titanic[feature])))
    ax[0].legend()

    sns.boxplot(x = titanic[feature], ax = ax[1],color="Brown")
```

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

```
    plt.show()
plot_hist('Age')
```

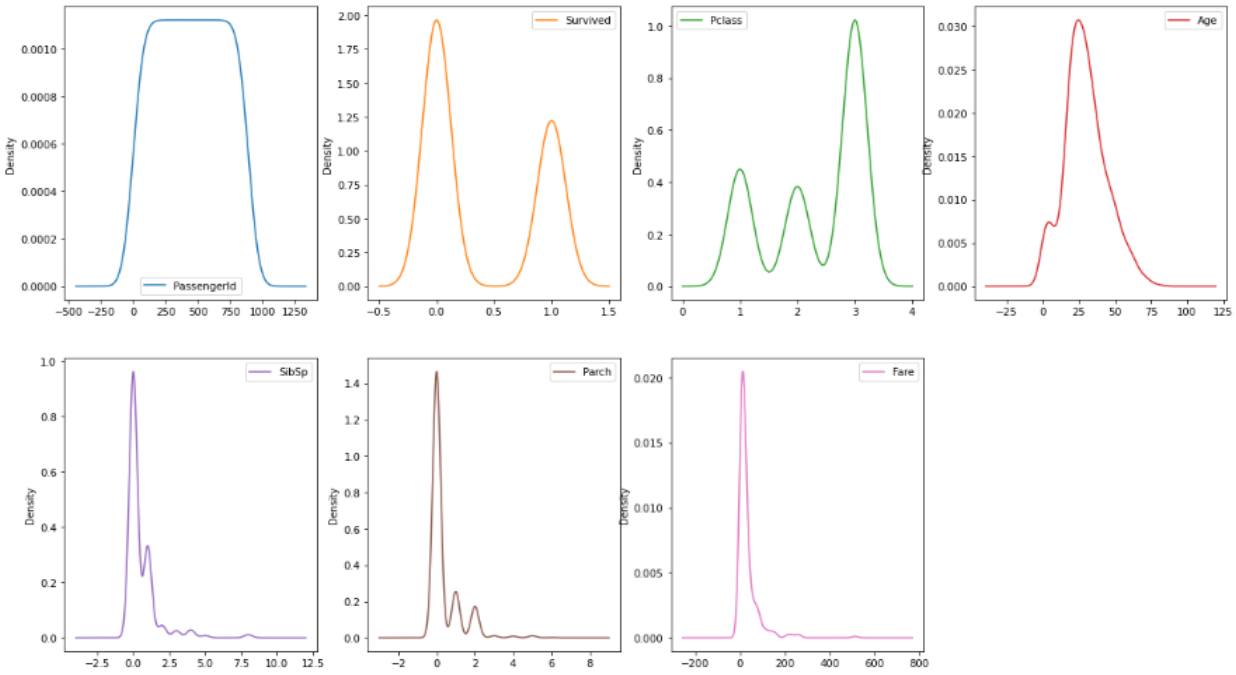



```
plt.figure(figsize=(12,5))
plt.title('top categories')
plt.ylabel('item_price')
titanic.groupby('Embarked')['Fare'].mean().sort_values(ascending=False)[0:15].plot(kind='line', marker='*', color='red', ms=10)
titanic.groupby('Embarked')['Fare'].mean().sort_values(ascending=False)[0:15].plot(kind='bar',color=sns.color_palette("inferno_r", 7))
plt.show()
```



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

```
numeric_feature = titanic.dtypes!=object
final_numeric_feature =
titanic.columns[numeric_feature].tolist()

titanic[final_numeric_feature].plot(kind='density',
subplots=True, layout=(2,4), sharex=False, figsize= (20,12))
plt.show()
```




```
df.describe().round(2).T.sort_values(by='std' , ascending = False)\
                .style.background_gradient(cmap='GnBu')\
                .bar(subset=["max"], color='#BB0000')\
                .bar(subset=["min",], color='green')\
                .bar(subset=["mean",], color='Orange')\
                .bar(subset=['std'], color='pink')\
                .bar(subset=['50%'], color='magenta')
```

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

	count	mean	std	min	25%	50%	75%	max
motor_speed	1330816.000000	2202.080000	1859.660000	-275.550000	317.110000	1999.980000	3760.640000	6000.020000
i_q	1330816.000000	37.410000	92.180000	-293.430000	1.100000	15.770000	100.610000	301.710000
torque	1330816.000000	31.110000	77.140000	-246.470000	-0.140000	10.860000	91.600000	261.010000
i_d	1330816.000000	-68.720000	64.930000	-278.000000	-115.410000	-51.090000	-2.980000	0.050000
u_d	1330816.000000	-25.130000	63.090000	-131.530000	-78.690000	-7.430000	1.470000	131.470000
u_q	1330816.000000	54.280000	44.170000	-25.290000	12.070000	48.940000	90.030000	133.040000
stator_winding	1330816.000000	66.340000	28.670000	18.590000	42.790000	65.110000	88.140000	141.360000
profile_id	1330816.000000	40.790000	25.050000	2.000000	17.000000	43.000000	65.000000	81.000000
stator_tooth	1330816.000000	56.880000	22.950000	18.130000	38.420000	56.040000	75.590000	111.950000
coolant	1330816.000000	36.230000	21.790000	10.620000	18.700000	26.900000	49.860000	101.600000
stator_yoke	1330816.000000	48.190000	19.990000	18.080000	31.990000	45.630000	61.460000	101.150000
pm	1330816.000000	58.510000	19.000000	20.860000	43.150000	60.270000	72.010000	113.610000
ambient	1330816.000000	24.570000	1.930000	8.780000	23.180000	24.800000	26.220000	30.710000


```
def highlight_min(s, props=''):
    return np.where(s == np.nanmin(s.values), props, '')
titanic.describe().style.apply(highlight_min, props='color:yellow;background-color:Grey', axis=0)
```

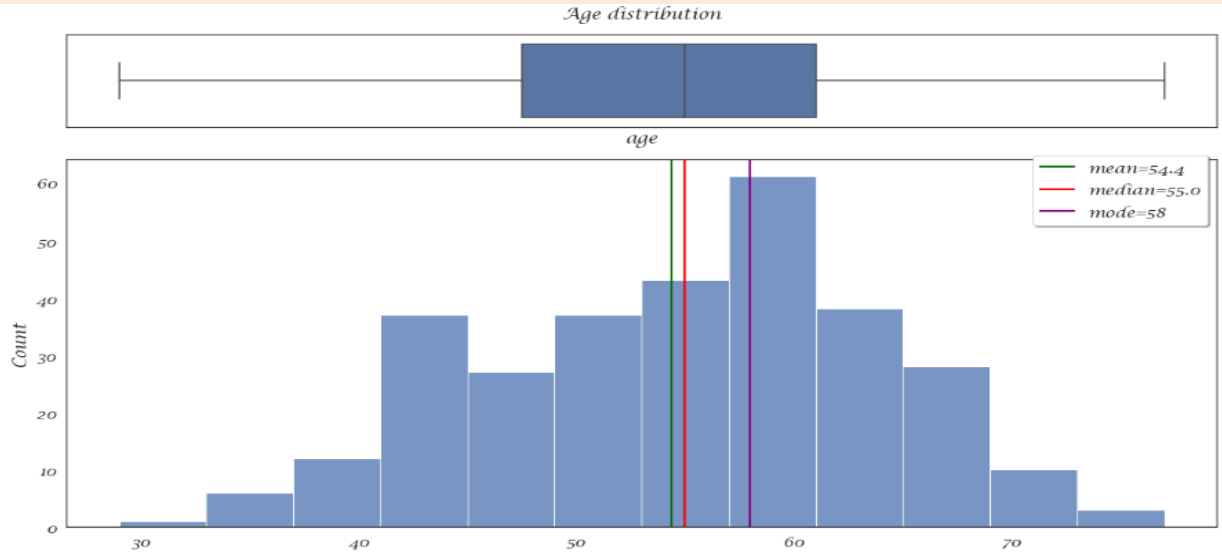
	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

```
titanic[titanic["Age"] >= 50].describe().style.background_gradient(cmap='RdPu')
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	74.000000	74.000000	74.000000	74.000000	74.000000	74.000000	74.000000
mean	445.675676	0.364865	1.554054	57.540541	0.270270	0.283784	46.364415
std	242.330133	0.484678	0.742854	6.767042	0.504698	0.692826	49.480722
min	7.000000	0.000000	1.000000	50.000000	0.000000	0.000000	6.237500
25%	250.750000	0.000000	1.000000	52.000000	0.000000	0.000000	12.643750
50%	483.500000	0.000000	1.000000	56.000000	0.000000	0.000000	27.631250
75%	631.750000	1.000000	2.000000	61.750000	0.000000	0.000000	68.744800
max	880.000000	1.000000	3.000000	80.000000	2.000000	4.000000	263.000000

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

```
fig, ax = plt.subplots(2, 1, sharex=True,
figsize=(17,10),gridspec_kw={"height_ratios": (.2, .8)})
ax[0].set_title('Age distribution',fontsize=18,pad=20)
sns.boxplot(x='age', data=heart, ax=ax[0])
ax[0].set(yticks=[])
sns.histplot(x='age', data=heart, ax=ax[1])
ax[1].set_xlabel(col, fontsize=16)
plt.axvline(heart['age'].mean(), color='darkgreen',
linewidth=2.2, label='mean=' +
str(np.round(heart['age'].mean(),1)))
plt.axvline(heart['age'].median(), color='red', linewidth=2.2,
label='median='+ str(np.round(heart['age'].median(),1)))
plt.axvline(heart['age'].mode()[0], color='purple',
linewidth=2.2, label='mode='+ str(heart['age'].mode()[0]))
plt.legend(bbox_to_anchor=(1, 1.03), ncol=1, fontsize=17,
fancybox=True, shadow=True, frameon=True)
plt.tight_layout()
plt.show()
```




```
plt.rcParams['font.family'] = 'Lucida Calligraphy'
```

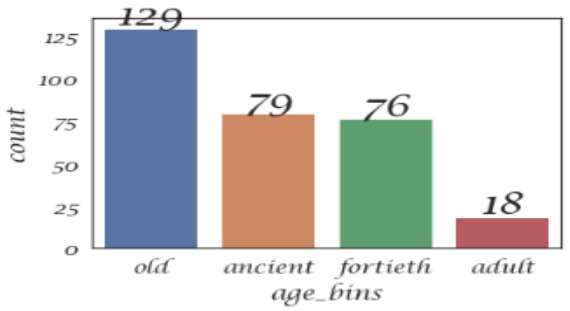
Syed Afroz Ali
Follow me on LinkedIn for more: https://www.linkedin.com/in/syed-afroz-70939914/

```
plt.rcParams['font.size'] = 30

heart["age_bins"]= pd.cut(heart["age"] , bins=[29 , 40 , 50 , 60 ,  80] , labels=["adult" , "fortieth" , "old" , "ancient"] )

def count_plot(data  , x=None , y=None  , figsize =None , title =None , color =None , prop=False , rotation_x =0 ):
    if x is None and y is None :
        raise("Expected y or x")
    if x is not None and y is not None:
        raise("Expected y or x not both")
    count_type = data[y if x is None else x].value_counts(ascending =False)
    Sum = count_type.sum()
    type_order = count_type.index
    plt.figure(figsize=figsize if figsize is None else (12 , 7))
    if x is None:
        sns.countplot(data = data , y=y , color = color ,order=type_order)
        if prop==True:
            for i in range(len(count_type)):
                count = count_type[i]
                pct_string ="{:0.1f}%".format(100*count/Sum)
                plt.text(count+1 , i , pct_string , va="center")
        if prop==False:
            for i in range(len(count_type)):
                count = count_type[i]
                pct_string ="{}".format(count)
                plt.text(count+1 , i , pct_string , va="center")
        plt.title(title)
        plt.show()
```

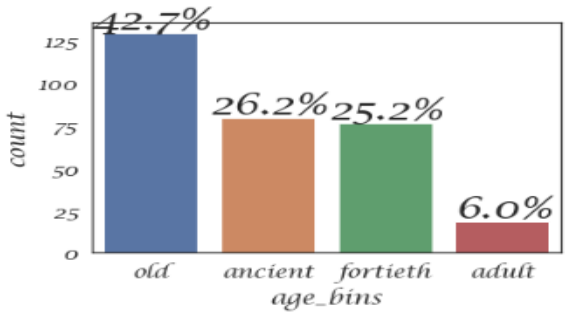
Syed Afroz Ali
Follow me on LinkedIn for more: https://www.linkedin.com/in/syed-afroz-70939914/

```
    if y is None :
        sns.countplot(data = data , x = x , color = color , order =
type_order)
        locs , labels =plt.xticks(rotation = rotation_x)
        if prop == True :
            for loc , label in zip(locs , labels):
                count = count_type[label.get_text()]
                pct_string ="{:0.1f}%".format(100*count/Sum)
                plt.text(loc , count+2 ,pct_string,ha ="center")
        if prop==False :
            for loc , label in zip(locs , labels):
                count = count_type[label.get_text()]
                pct_string ="{}".format(count)
                plt.text(loc , count+2 ,pct_string,ha ="center")
        plt.title(title)
        plt.show()
```

```
1 count_plot(data = heart , x ="age_bins")
```

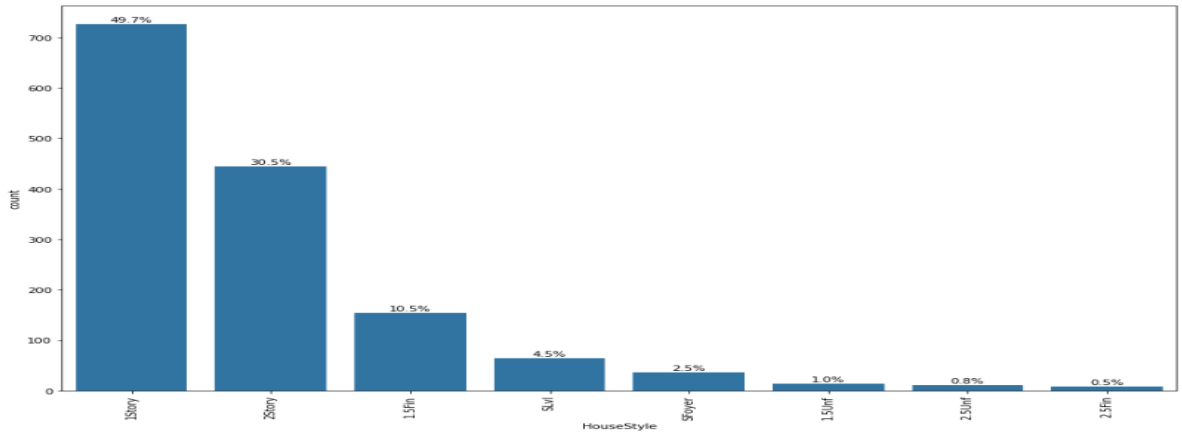



```
1 count_plot(data = heart ,prop=True, x ="age_bins")
```



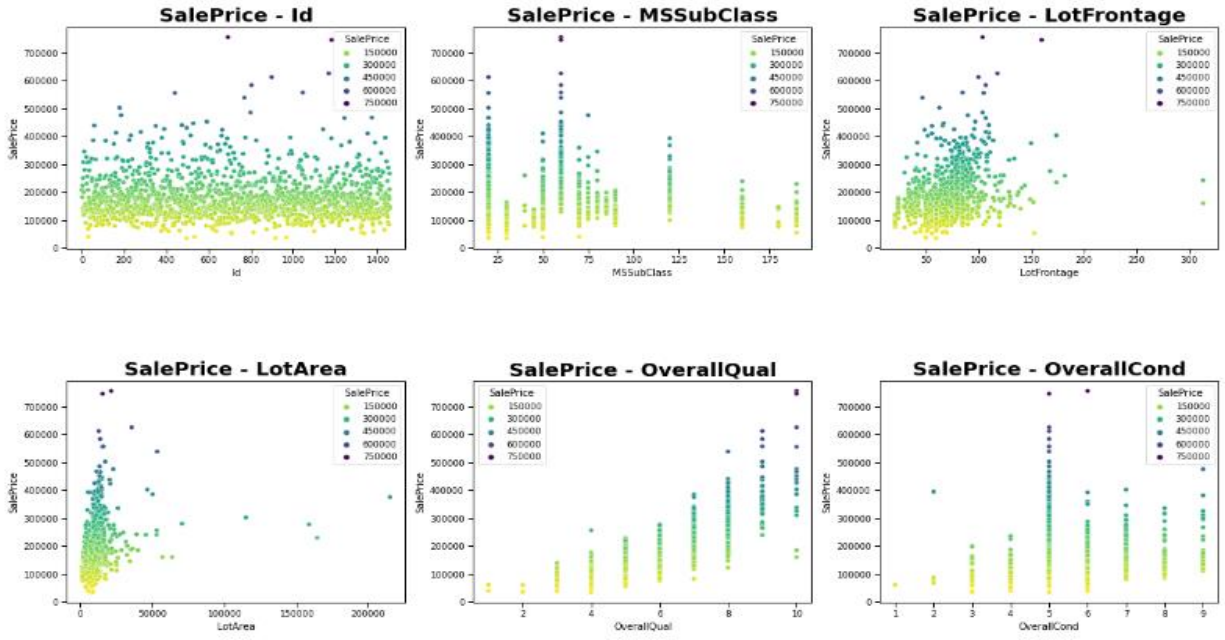
Syed Afroz Ali
Follow me on LinkedIn for more: https://www.linkedin.com/in/syed-afroz-70939914/

```
# Barchart sorted by frequency
base_color = sns.color_palette()[0]
cat_order = train_eda[col_name].value_counts().index
plt.figure(figsize=(15,10))
plt.xticks(rotation = 90)
sns.countplot(data = train_eda, x = col_name, order = cat_order, color =
base_color);

# add annotations
n_points = train_eda.shape[0]
cat_counts = train_eda[col_name].value_counts()
locs, labels = plt.xticks() # get the current tick locations and labels

# loop through each pair of locations and labels
for loc, label in zip(locs, labels):

    # get the text property for the label to get the correct count
    count = cat_counts[label.get_text()]
    pct_string = '{:0.1f}%'.format(100*count/n_points)

    # print the annotation just below the top of the bar
    plt.text(loc, count+4, pct_string, ha = 'center', color = 'black')
```



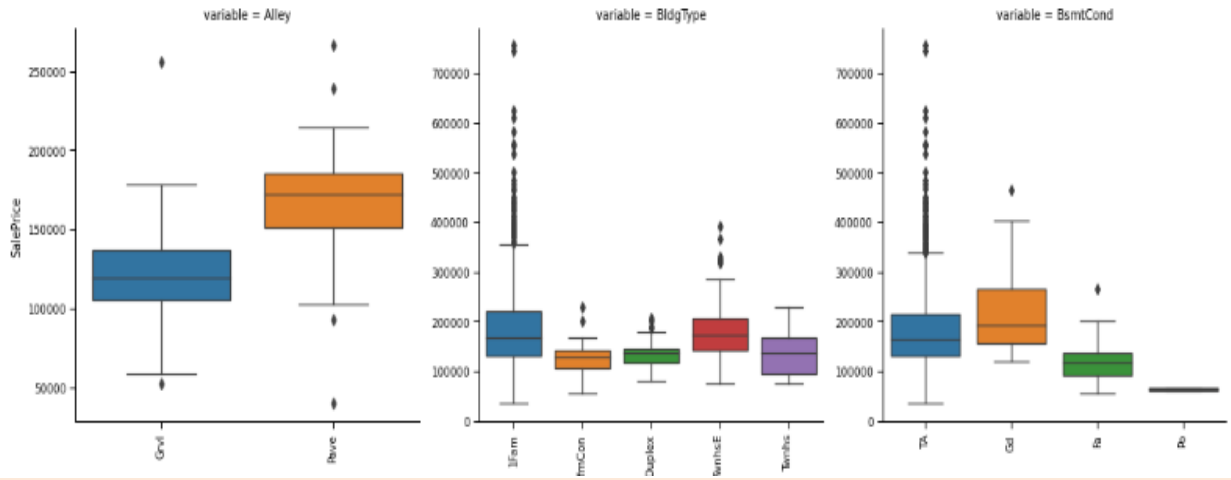
Syed Afroz Ali
Follow me on LinkedIn for more: https://www.linkedin.com/in/syed-afroz-70939914/

```
train = pd.read_csv('train_housing.csv')

#Visualising numerical predictor variables with Target Variables
train_num = train.select_dtypes(include=['int64','float64'])
fig,axs= plt.subplots(12,3,figsize=(20,80))
#adjust horizontal space between plots
fig.subplots_adjust(hspace=0.6)
for i,ax in zip(train_num.columns,axs.flatten()):
    sns.scatterplot(x=i, y='SalePrice',
hue='SalePrice',data=train_num,ax=ax,palette='viridis_r')
    plt.xlabel(i,fontsize=12)
    plt.ylabel('SalePrice',fontsize=12)
    #ax.set_yticks(np.arange(0,900001,100000))
    ax.set_title('SalePrice'+' -
'+str(i),fontweight='bold',size=20)
```



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

```
train = pd.read_csv('train_housing.csv')
categorical = train.select_dtypes(include=['object'])
##Visualising Categorical predictor variables with Target Variables
def facetgrid_boxplot(x, y, **kwargs):
    sns.boxplot(x=x, y=y)
    x=plt.xticks(rotation=90)

f = pd.melt(train, id_vars=['SalePrice'],
value_vars=sorted(train[categorical.columns]))
g = sns.FacetGrid(f, col="variable", col_wrap=3,
sharex=False, sharey=False, size=5)
g = g.map(facetgrid_boxplot, "value", "SalePrice")
```

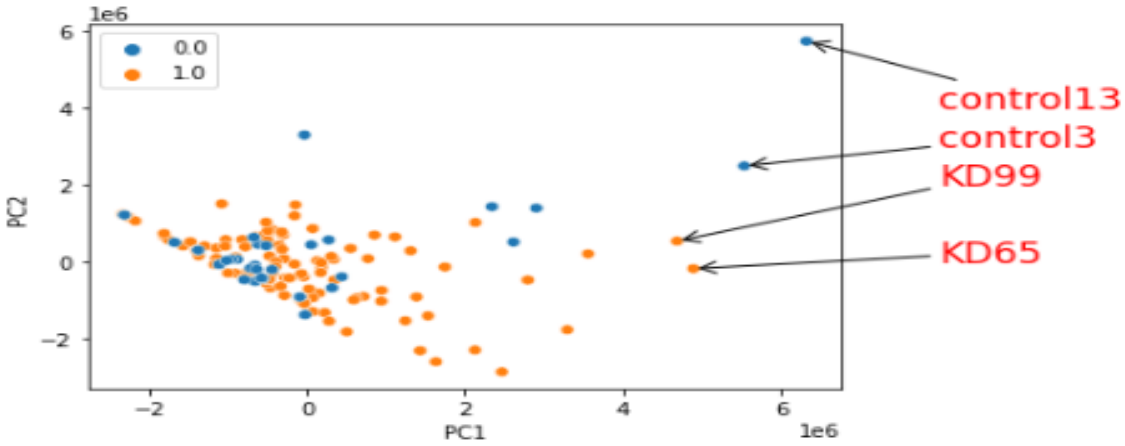



```
import matplotlib.pyplot as plt
import seaborn as sns

sns.scatterplot(x=df.iloc[:,0], y=df.iloc[:,1], hue=y)
plt.annotate("KD65", (df.iloc[64,0], df.iloc[64,1]), (8*1e6, 1),
arrowprops=dict(arrowstyle="->"), fontsize="xx-large",c='red'
)
```

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

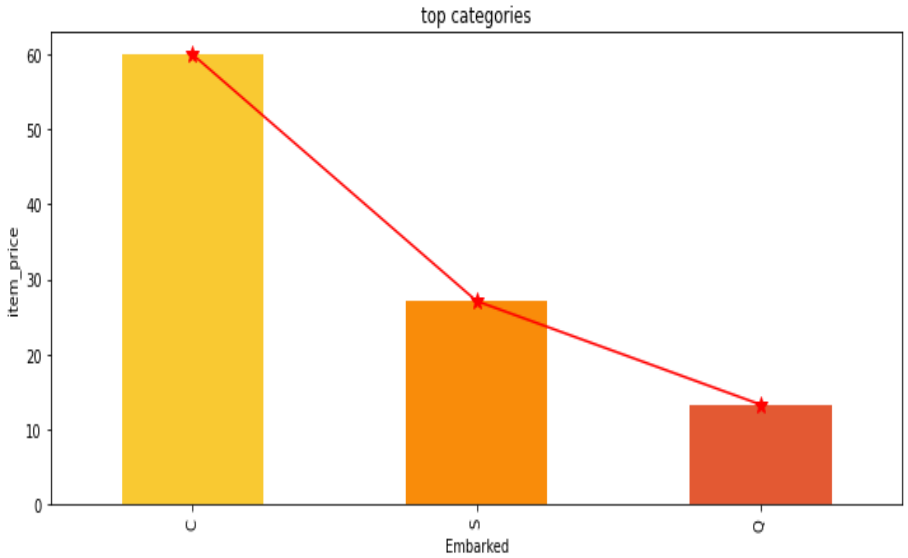
```
plt.annotate("KD99", (df.iloc[98,0], df.iloc[98,1]), (8*1e6, 2*1e6), arrowprops=dict(arrowstyle="->"), fontsize="xx-large",c='red')
plt.annotate("control3", (df.iloc[107,0], df.iloc[107,1]), (8*1e6, 3*1e6), arrowprops=dict(arrowstyle="->"), fontsize="xx-large",c='red')
plt.annotate("control13", (df.iloc[117,0], df.iloc[117,1]), (8*1e6, 4*1e6), arrowprops=dict(arrowstyle="->"), fontsize="xx-large",c='red')
```

```
Text(8000000.0, 4000000.0, 'control13')
```

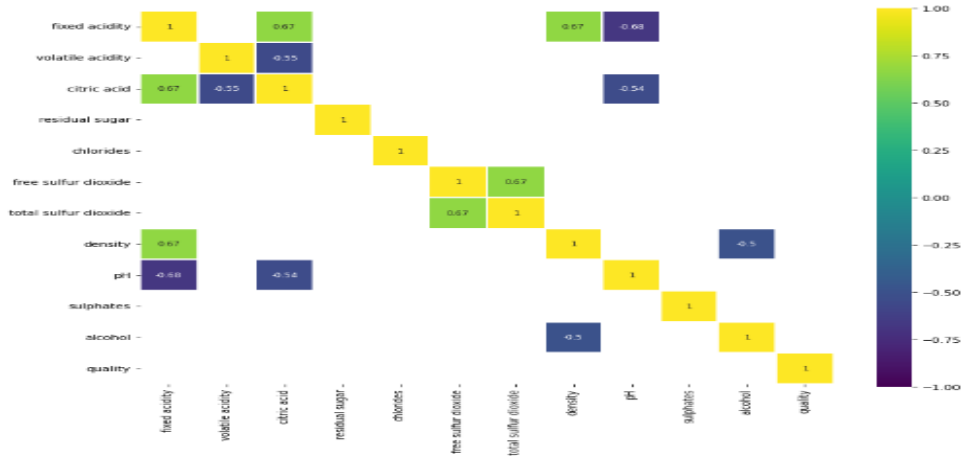



```
plt.figure(figsize=(12,5))
plt.title('top categories')
plt.ylabel('item_price')
titanic.groupby('Embarked')['Fare'].mean().sort_values(ascending=False)[0:15].plot(kind='line', marker='*', color='red', ms=10)
titanic.groupby('Embarked')['Fare'].mean().sort_values(ascending=False)[0:15].plot(kind='bar',color=sns.color_palette("inferno_r", 7))
plt.show()
```

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



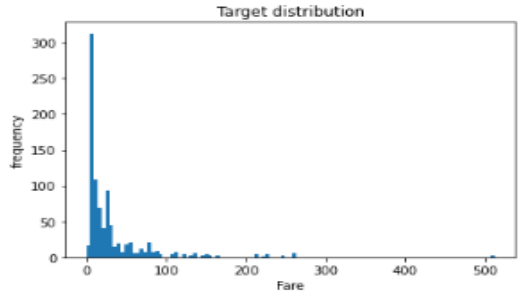

```
corr = wine.corr() # We already examined SalePrice correlations
plt.figure(figsize=(12, 10))

sns.heatmap(corr[(corr >= 0.5) | (corr <= -0.4)],
cmap='viridis', vmax=1.0, vmin=-1.0,
linewidths=0.1,annot=True, annot_kws={"size": 8},
square=True);
```

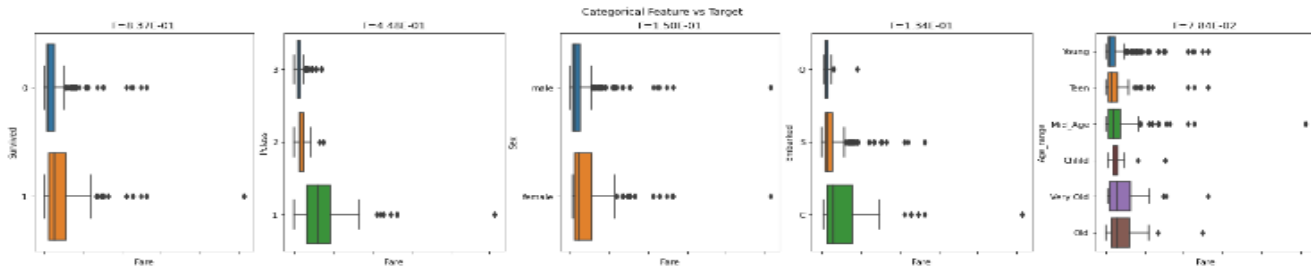


Syed Afroz Ali
Follow me on LinkedIn for more: https://www.linkedin.com/in/syed-afroz-70939914/

```
import dabl
dabl.plot(titanic, 'Fare');
```

```
Target looks like regression
```





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