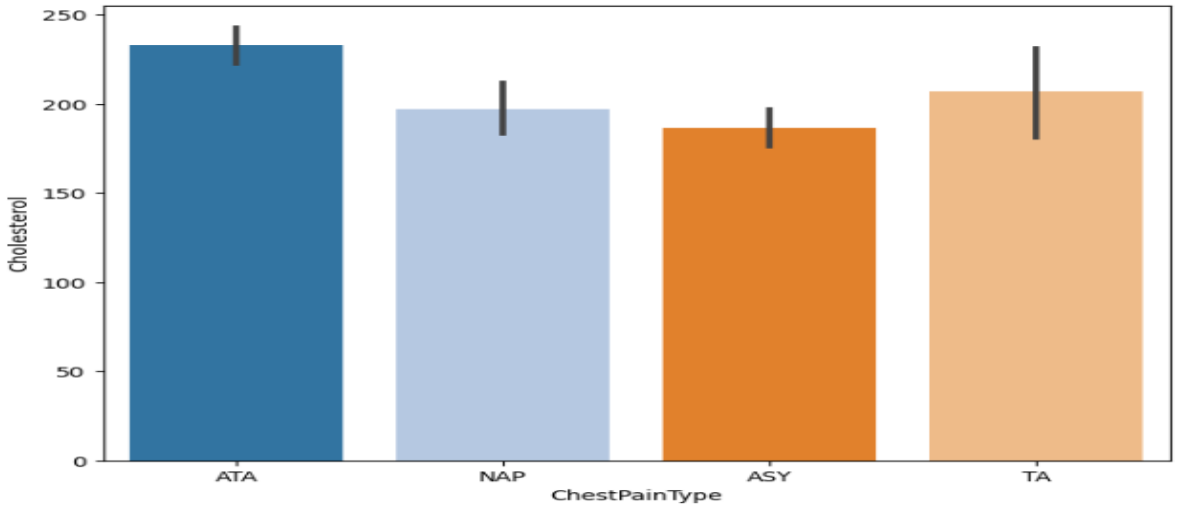
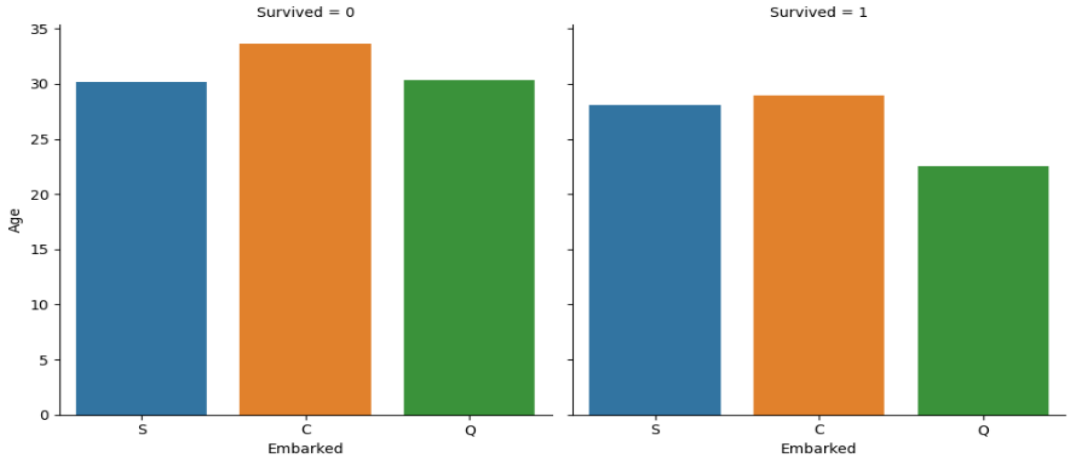
Mastering Data Visualization Techniques (Part 1)

Prepared by: Syed Afroz Ali

```
plt.figure(figsize = (8, 6))
plt.ticklabel_format(style = 'plain')
sns.barplot(x = heart["ChestPainType"], y = heart["Cholesterol"], palette = "tab20");
```

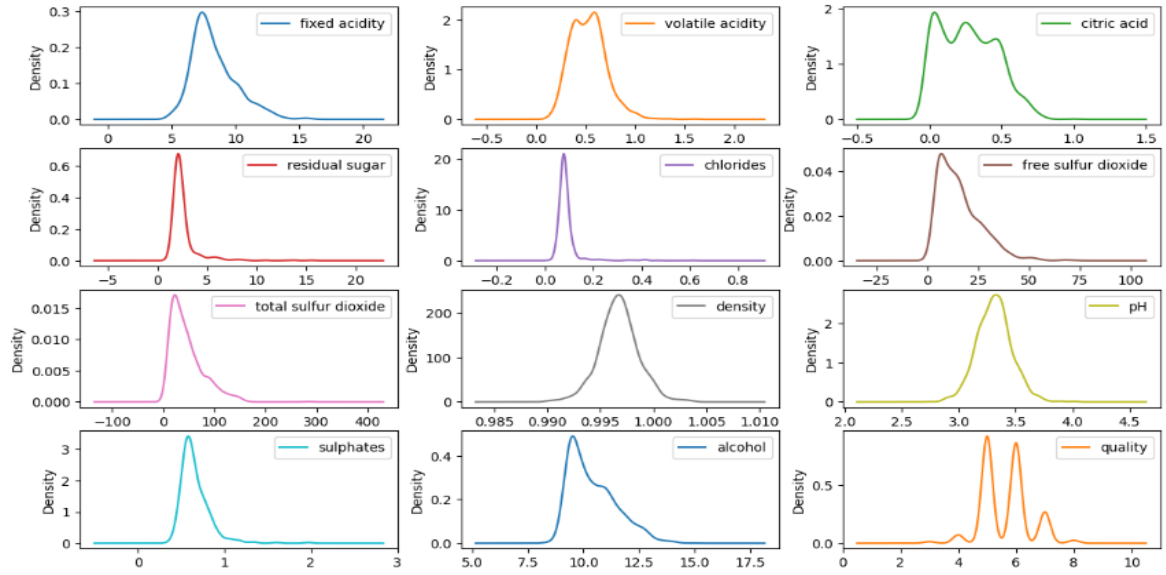



```
sns.catplot(data = titanic , x ="Embarked" , y ="Age" , col ="Survived" , kind="bar" , ci =None)
plt.show()
```

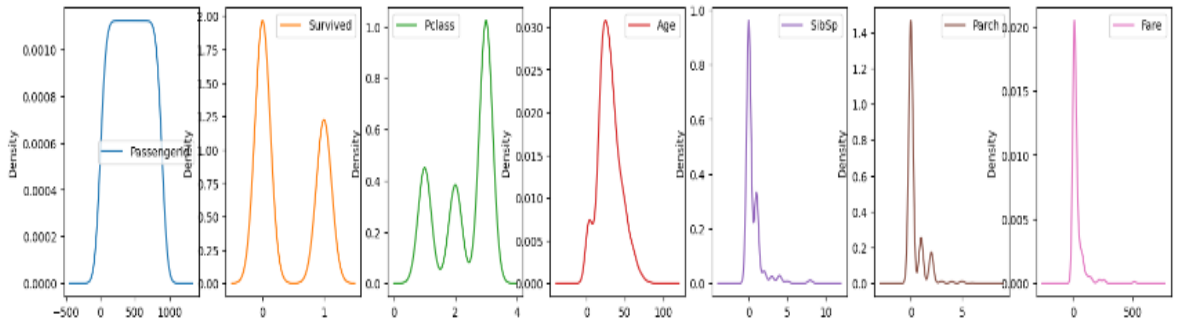



```
wine.plot(kind='density', subplots=True, layout=(4,3), sharex=False, figsize= (14,8))
```

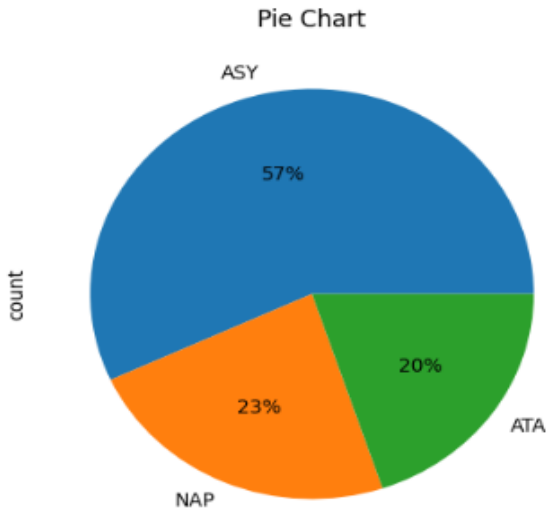
```
plt.show()
```



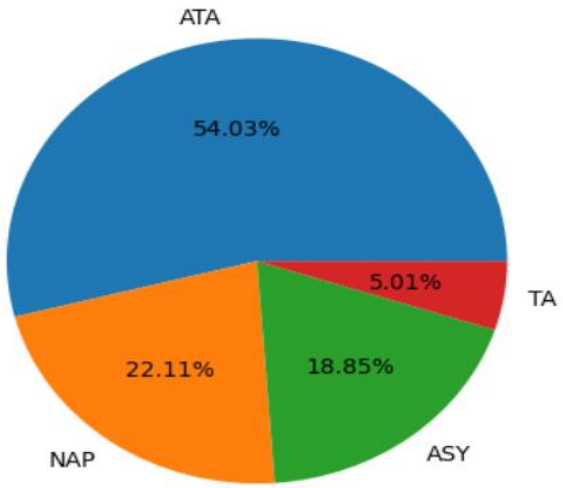

```
numeric_feature = titanic.dtypes!=object
final_numeric_feature = titanic.columns[numeric_feature].tolist()

titanic[final_numeric_feature].plot(kind='density', subplots=True, layout=(1,7), sharex=False, figsize= (20,4))
plt.show()
```



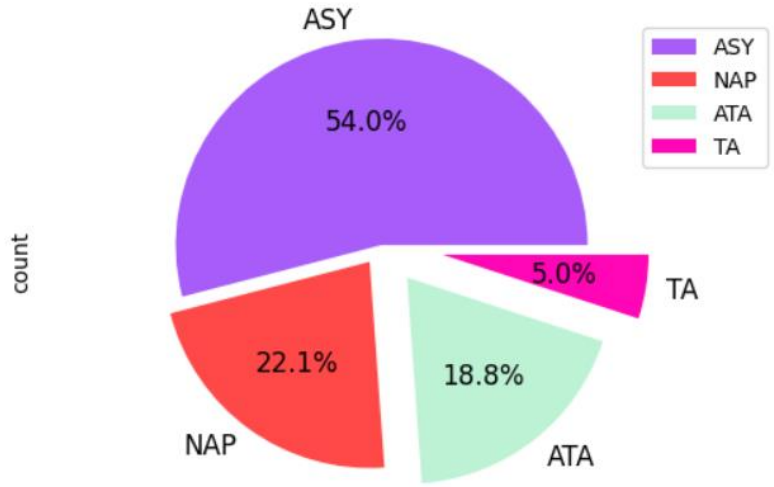

```
heart["ChestPainType"].value_counts()[:3].plot.pie(figsize = (5, 5),
                                                   autopct = '%1.0f%%')
plt.title("Pie Chart")
plt.xticks(rotation = 90)
plt.show()
```



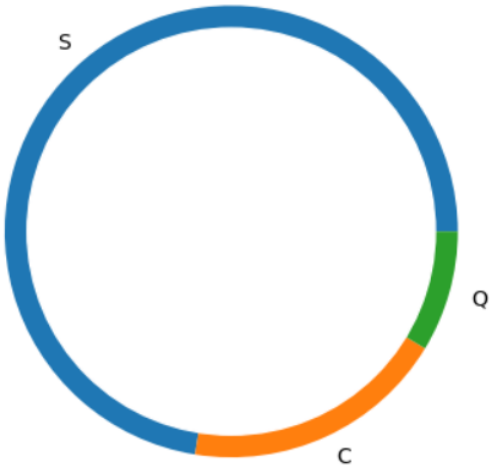

```
plt.pie(heart['ChestPainType'].value_counts(),labels=heart['ChestPainType'].unique(),autopct = '%1.2f%%');
```



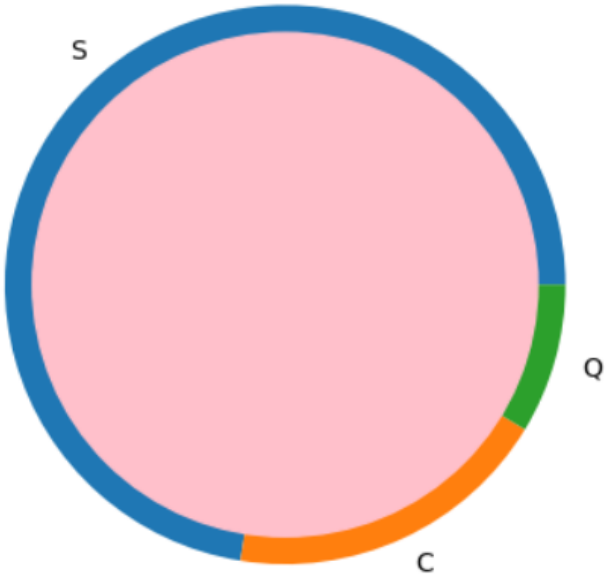

```
plt.figure(figsize = (6, 4))
counts = heart["ChestPainType"].value_counts()
explode = (0, 0.1, 0.2, 0.3)
colors = ['#A85CF9', '#FF4949', '#BDF2D5', '#FF06B7', '#4B7BE5', '#FF5D5D', '#FAC213', '#37E2D5', '#6D8B74', '#E9D5CA']
counts.plot(kind = 'pie', fontsize = 12, colors = colors, explode = explode, autopct = '%1.1f%%')
plt.axis('equal')
plt.legend(labels = counts.index, loc = "best")
plt.show()
```



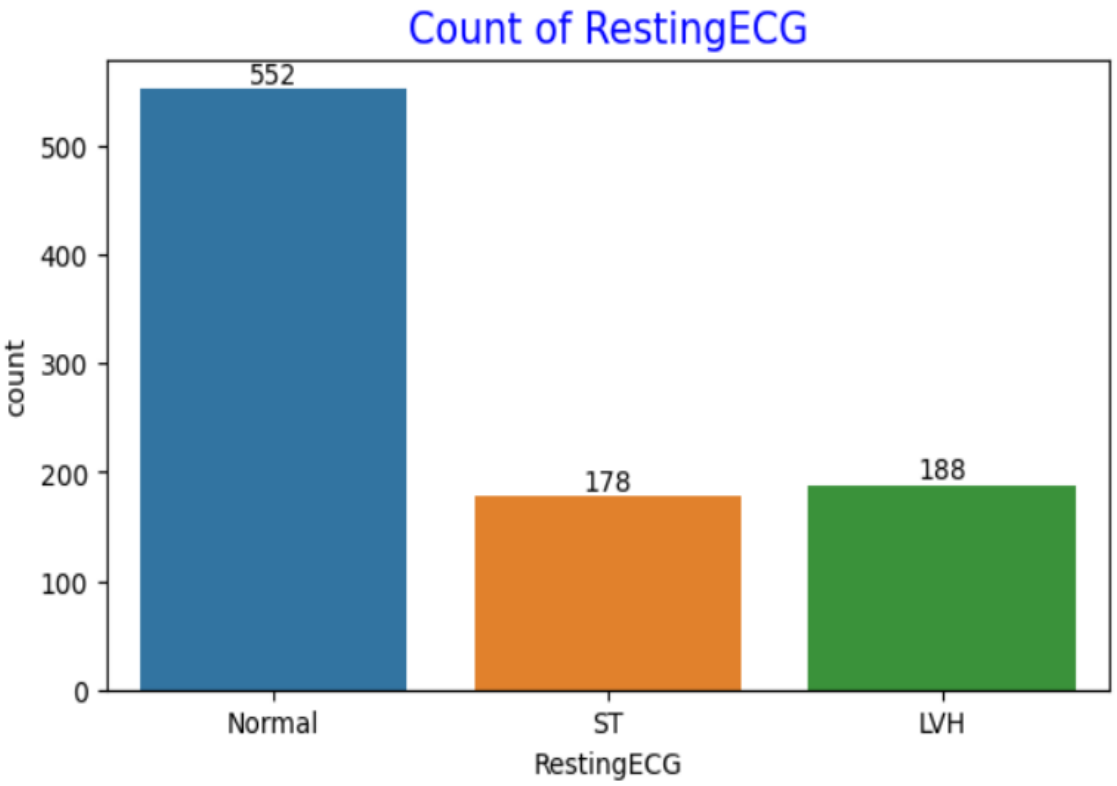

```
my_circle=plt.Circle( (0,0), 0.9, color='white')
plt.pie(titanic['Embarked'].value_counts()[:10].values, label
s = titanic['Embarked'].value_counts()[:10].index)
p=plt.gcf()
p.gca().add_artist(my_circle)
plt.show()
```



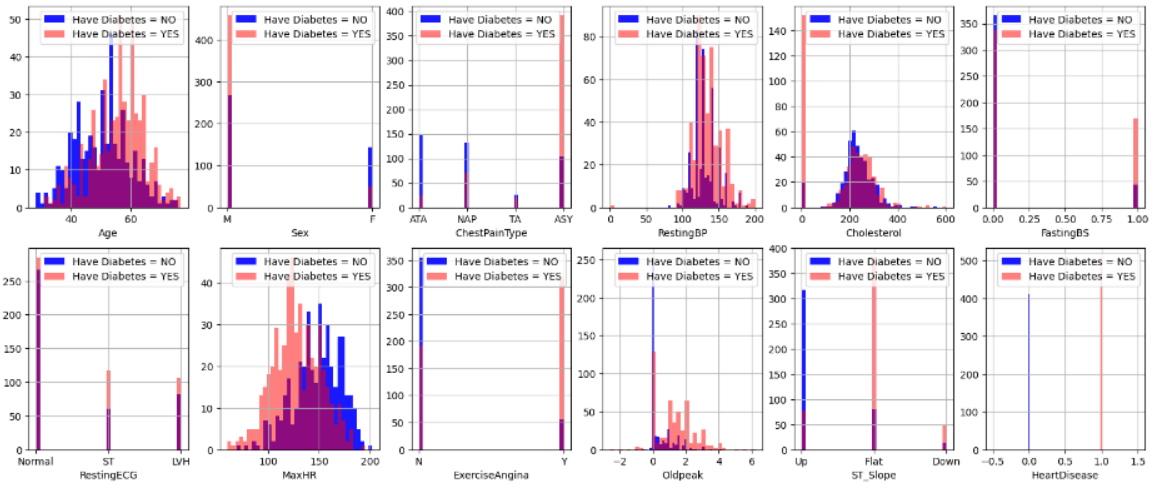

```
my_circle=plt.Circle( (0,0), 0.9, color='pink')
plt.pie(titanic['Embarked'].value_counts()[:10].values, label
s = titanic['Embarked'].value_counts()[:10].index)
p=plt.gcf()
p.gca().add_artist(my_circle)
plt.show()
```



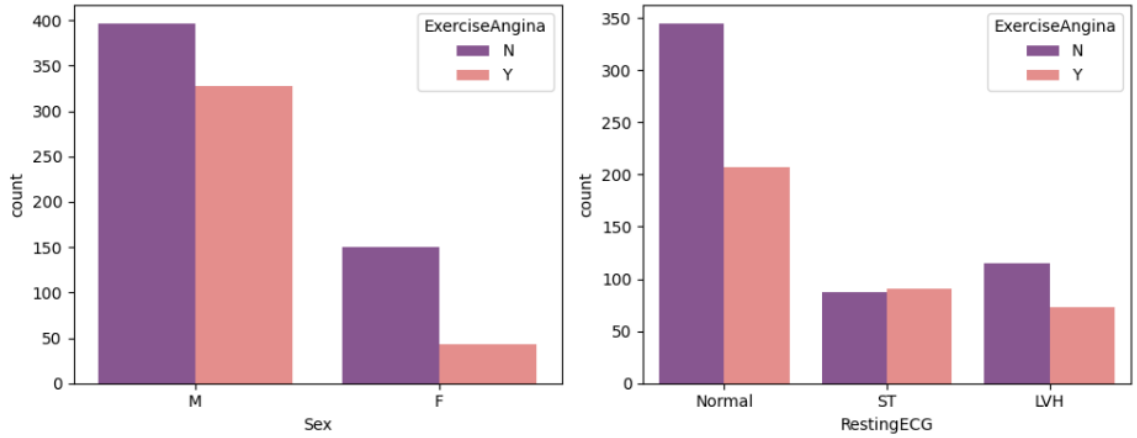

```
plt.figure(figsize = (7,4))
ax = sns.countplot(x=heart['RestingECG'])
for bars in ax.containers:
    ax.bar_label(bars)
plt.title("Count of RestingECG", fontsize = 15,color='Blue');
```



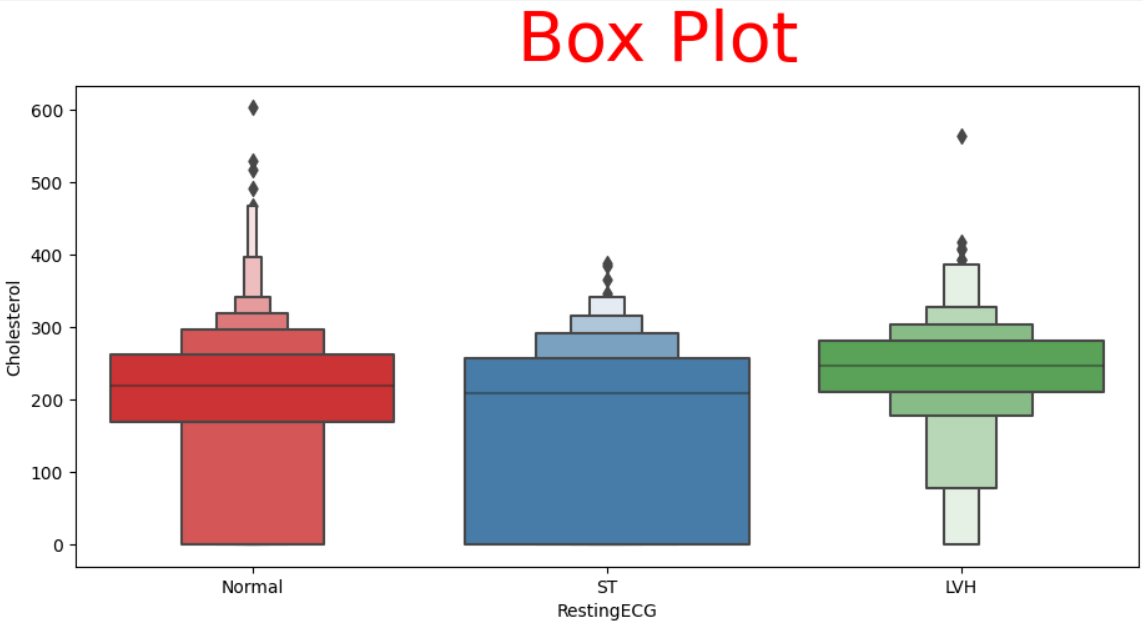
```
# Visulazing the distibution of the data for every feature
plt.figure(figsize=(20, 8))

for i, column in enumerate(heart.columns, 1):
    plt.subplot(2, 6, i)
    heart[heart["HeartDisease"] == 0][column].hist(bins=35,
color='blue', label='Have Diabetes = NO', alpha=0.9)
    heart[heart["HeartDisease"] == 1][column].hist(bins=35,
color='red', label='Have Diabetes = YES', alpha=0.5)
    plt.legend()
    plt.xlabel(column)
```



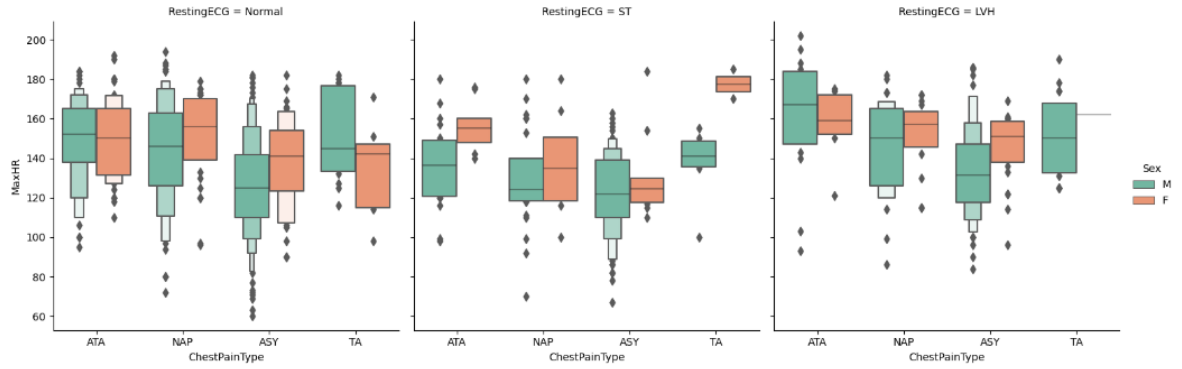

```
cat = ['Sex','RestingECG']

fig, ax = plt.subplots(1, 2, figsize = (10, 4))
for indx, (column, axes) in list(enumerate(list(zip(cat,
                                        ax.flatten())))):
    sns.countplot(ax = axes, x = heart[column], hue = heart[
'ExerciseAngina'],
                  palette = 'magma', alpha = 0.8)

else:
    [axes.set_visible(False) for axes in ax.flatten()[indx + 1:]
]
plt.tight_layout()
plt.show()
```



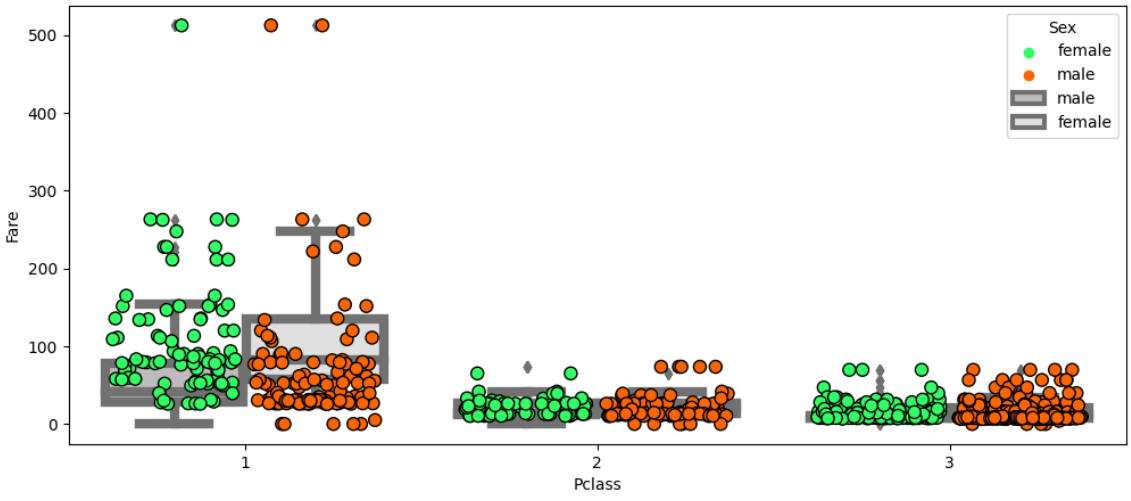

```
plt.figure(figsize=(11,5))
plt.gcf().text(.55, .95, "Box Plot", fontsize = 40, color='Red'
,ha='center', va='center')
sns.boxenplot(x=heart['RestingECG'] , y = heart['Cholester
ol'],palette="Set1")
plt.show()
```



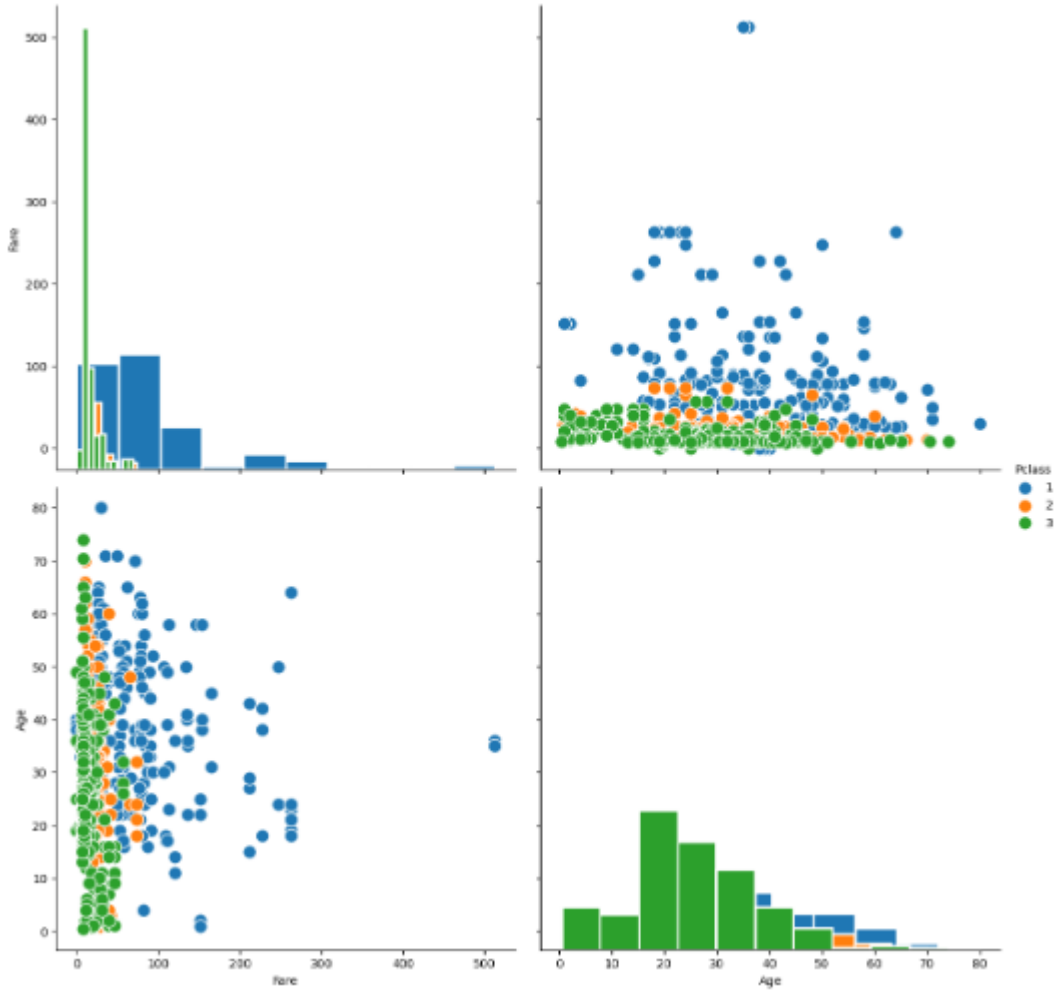

```
# Facet along the columns to show a categorical variable
using "col" parameter
plt.figure(figsize=(11,5))
sns.catplot(x="ChestPainType" , y = "MaxHR", hue= "Sex",
            col="RestingECG", kind="boxen",palette="Set2" , h
eight=5, aspect=1 ,data=heart)
plt.show();
```



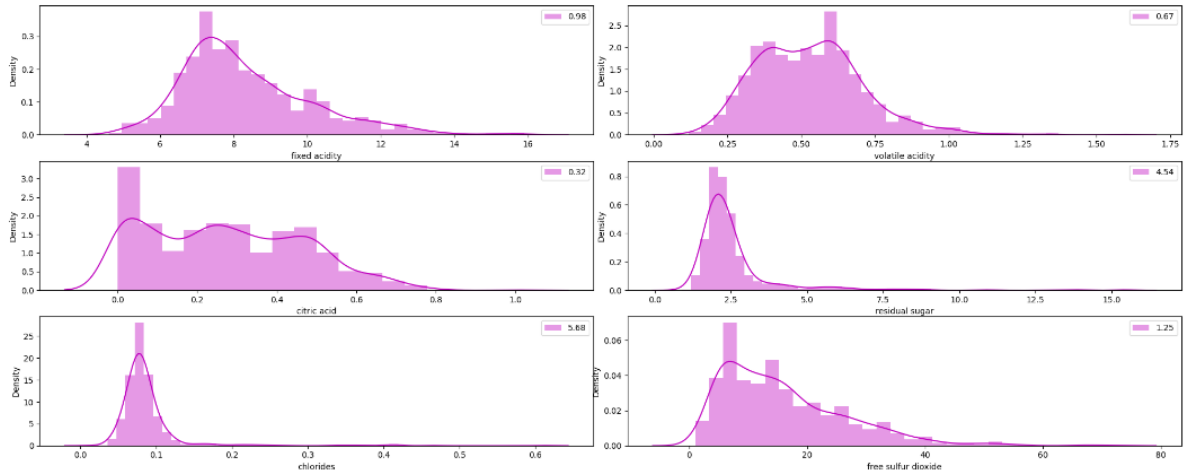

```
plt.figure(figsize=(12,5))
params = dict(data=titanic ,x = titanic.Pclass ,y = titanic.Fare ,hue=titanic.Sex,dodge=True)
sns.stripplot(**params , size=8,jitter=0.35,palette=['#33FF66','#FF6600','Blue'],edgecolor='black',linewidth=1)
sns.boxplot(**params ,palette=['#BDBDBD','#E0E0E0'],linewidth=6)
plt.show()
```



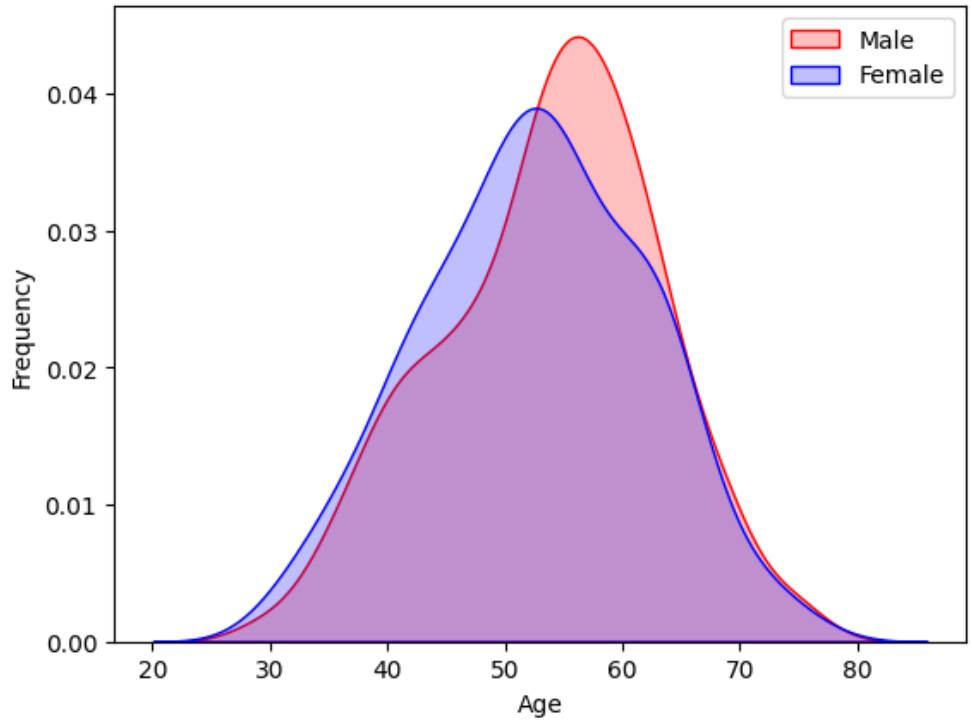

```
# Plot a subset of variables
g = sns.PairGrid(titanic, hue='Pclass' ,x_vars=["Fare" , "Age"],y_vars=["Fare" , "Age"],
                 height=6, aspect=1)
g = g.map_offdiag(plt.scatter , edgecolor="w", s=130)
g = g.map_diag(plt.hist , edgecolor ='w', linewidth=2)
g = g.add_legend()
plt.show()
```



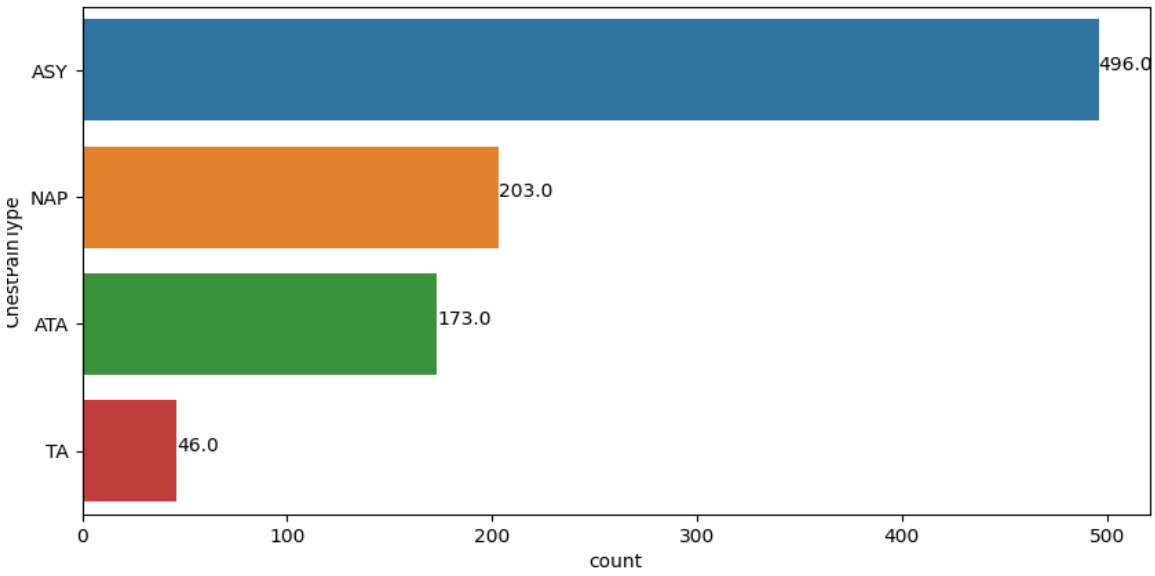

```
features_mean= list(wine.columns[:6])

num_rows, num_cols = 3,2

fig, axes = plt.subplots(num_rows, num_cols, figsize=(20, 8
))
fig.tight_layout()

for index, column in enumerate(wine[features_mean].colu
mns):
    i,j = (index // num_cols, index % num_cols)
    g = sns.distplot(wine[column], color="m", label="%.2f"%
(wine[column].skew()), ax=axes[i,j])
    g = g.legend(loc="best")
```



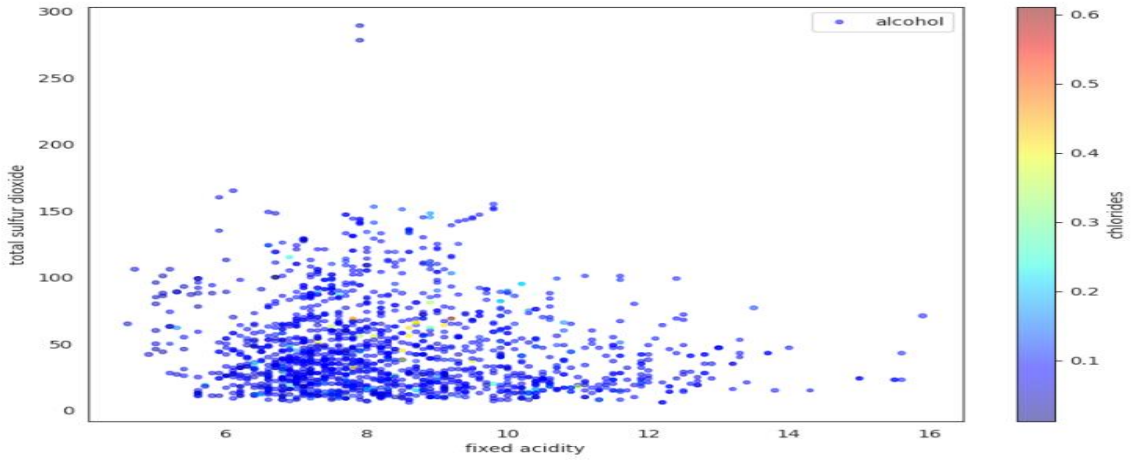

```
y = heart['Sex']

# Explore Age distibution
g = sns.kdeplot(heart["Age"][(y == 'M') & (heart["Age"].notnull())], color="Red", shade=True)
g = sns.kdeplot(heart["Age"][(y == 'F') & (heart["Age"].notnull())], ax=g, color="Blue", shade=True)
g.set_xlabel("Age")
g.set_ylabel("Frequency")
g = g.legend(["Male","Female"])
```



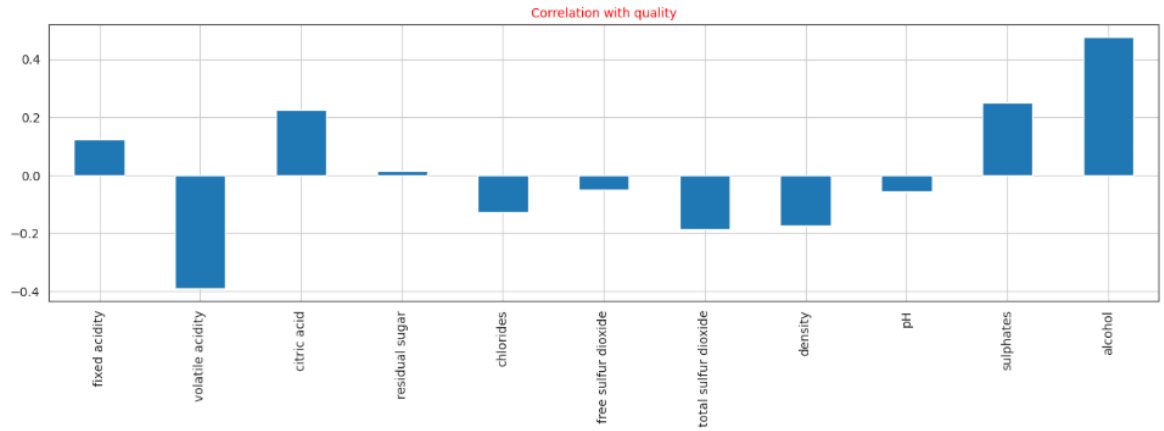
```
raw_df = heart [['Age', 'Sex', 'ChestPainType', 'RestingBP', 'Cholesterol', 'FastingBS',
       'RestingECG', 'MaxHR', 'ExerciseAngina', 'Oldpeak', 'ST_Slope',
       'HeartDisease']]
# Function to print width of barcharts on the bars
def barw(ax):
    for p in ax.patches:
        val = p.get_width() #height of the bar
        x = p.get_x()+ p.get_width() # x- position
        y = p.get_y() + p.get_height()/2 #y-position
        ax.annotate(round(val,2),(x,y))
plt.figure(figsize=(10,5))
ax0 = sns.countplot(data = heart, y ='ChestPainType', order = heart['ChestPainType'].value_counts().index)
barw(ax0)
plt.show()
```



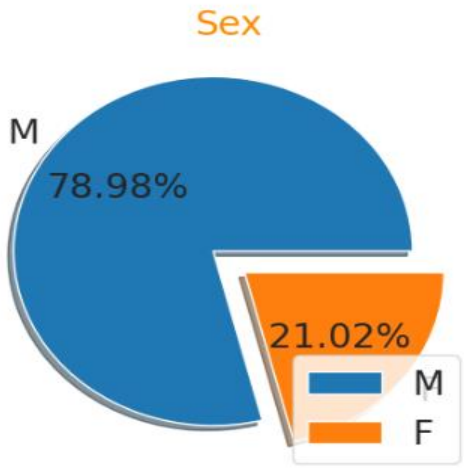

```
sns.set_style('white')
wine.plot(kind="scatter", x="fixed acidity", y="total sulfur dioxide", alpha=.5,
          s=wine["alcohol"], label="alcohol", figsize=(10,7),
          c="chlorides", cmap=plt.get_cmap("jet"), colorbar=True,
          sharex=False)
plt.legend()
plt.show()
```



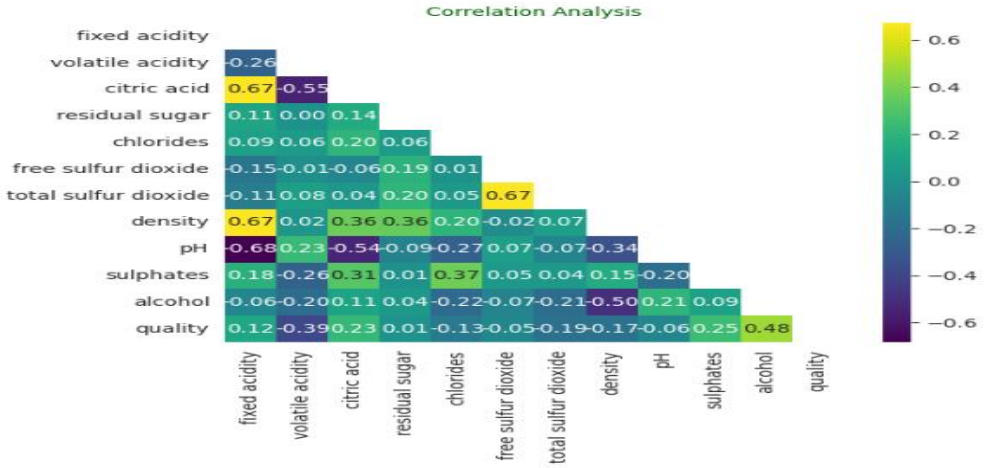

```
#Correlation with Response Variable class
X = wine.drop(['quality'], axis=1)
y = wine['quality']

X.corrwith(y).plot.bar(figsize=(16, 4), rot=90, grid=True)
plt.title('Correlation with quality',
          fontsize=30,
          color='Red',
          font='Times New Roman')
plt.show()
```



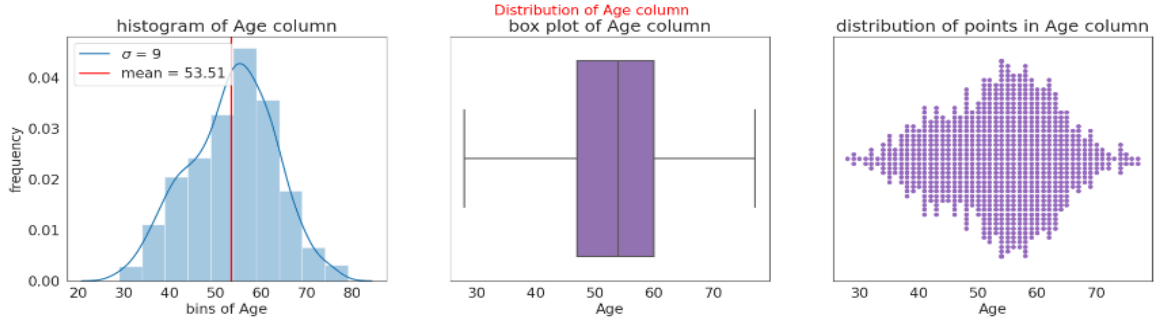

```
import matplotlib
matplotlib.rcParams.update({'font.size': 20})
ax=heart['Sex'].value_counts().plot.pie(explode=[0.1, 0.1],autopct='
%1.2f%%',shadow=True);
ax.set_title(label = "Sex", fontsize = 40,color='DarkOrange',font='Luc
ida Calligraphy');
plt.legend(labels=['M','F'])
plt.axis('off');
```



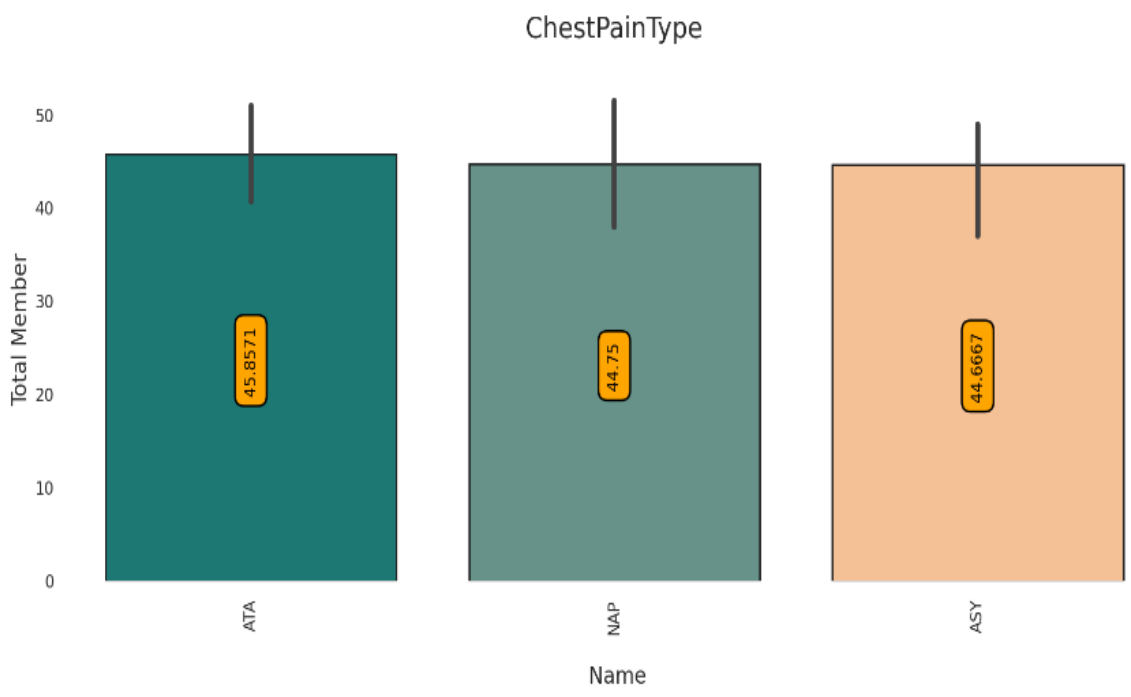

```
matplotlib.rcParams.update({'font.size': 10})
corr = wine.corr()
mask = np.triu(np.ones_like(corr, dtype=bool))
plt.title('Correlation Analysis',
          fontsize=25,
          color='DarkGreen',
          font='Times New Roman')
sns.heatmap(corr,
            mask=mask,
            annot=True,
            lw=0,
            linecolor='white',
            cmap='viridis',
            fmt="0.2f")
plt.xticks(rotation=90)
plt.yticks(rotation=0)
plt.show()
```



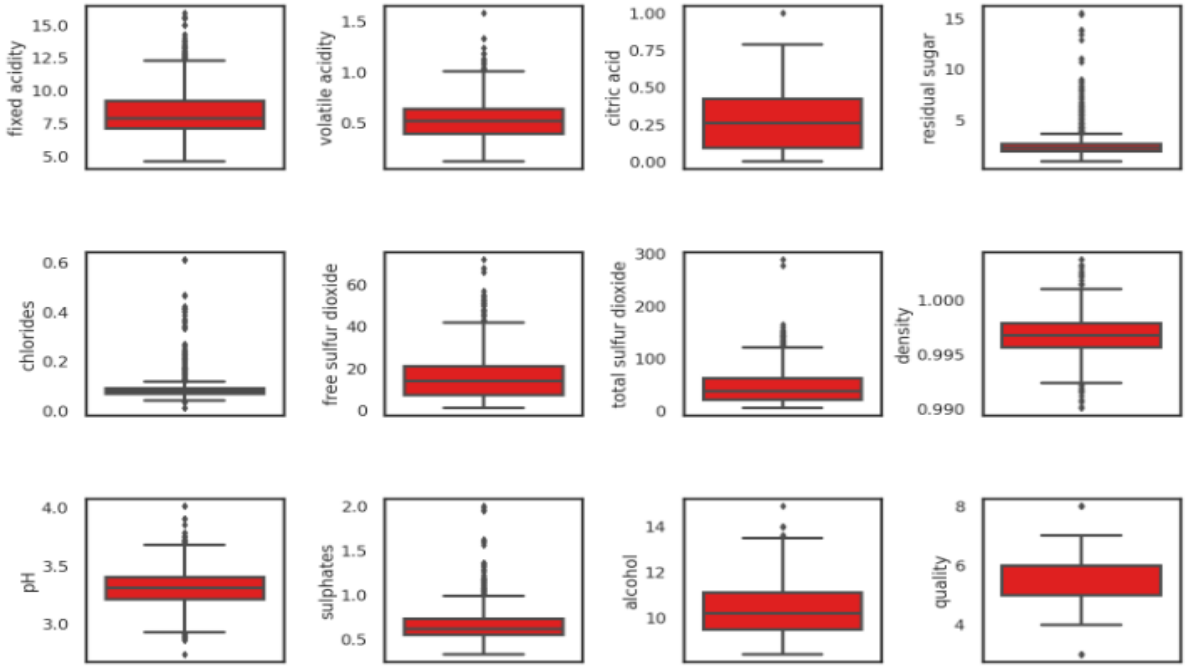
```
#set configuration for charts
plt.rcParams["figure.figsize"]=[20 , 5]
plt.rcParams["font.size"]=15
plt.rcParams["legend.fontsize"]="medium"
plt.rcParams["figure.titlesize"]="medium"

def plot_disribution(data , x ,color,bins ):
    mean = data[x].mean()
    std = data[x].std()
    info=dict(data = data , x = x , color = color)
    plt.subplot(1 , 3 , 1 , title =f"Ditstribution of {x} column")
    sns.distplot(a=data[x] , bins = bins)
    plt.xlabel(f"bins of {x}")
    plt.axvline(mean , label ="mean" , color ="red")
    plt.ylabel("frequency")
    plt.legend(["${\sigma}$ = %d"%std , f"mean = {mean:.2f}"])
    plt.title(f"histogram of {x} column")
    plt.subplot(1 , 3 , 2)
    sns.boxplot(**info)
    plt.xlabel(f"{x}")
    plt.title(f"box plot of {x} column")
    plt.subplot(1 , 3 , 3)
    sns.swarmplot(**info)
    plt.xlabel(f"{x}")
    plt.title(f"distribution of points in {x} column")
    plt.suptitle(f"Distribution of {x} column" , fontsize =15 , color="red
")
plt.show()

age_bins = np.arange(29 , 77+5 , 5)
base_color = sns.color_palette()[4]
plot_disribution(data = heart , x ="Age" , color = base_color , bins=ag
e_bins)
```



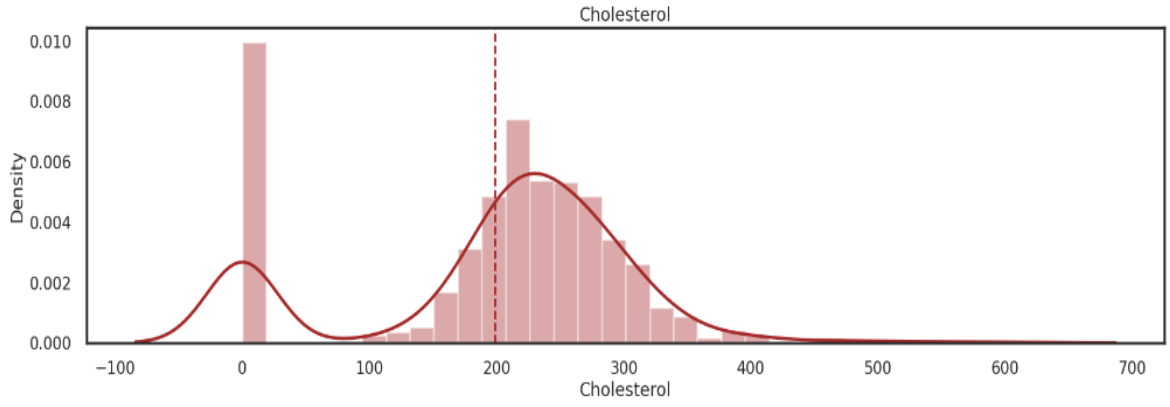
```
sns.set_style("white")
sns.set_context("poster",font_scale = .7)
palette = ["#1d7874","#679289","#f4c095","#ee2e31","#ffb563","#918450","#f85e00","#a41623","#9a031e","#d6d6d6","#ffee32","#ffd100","#333533","#202020"]
# sns.palplot(sns.color_palette(palette))
# plt.show()

plt.subplots(figsize=(20,8))
p = sns.barplot(x=heart["ChestPainType"][:14],y=heart["Age"],palette=palette, saturation=1, edgecolor = "#1c1c1c", linewidth = 2)
p.axes.set_title("\n ChestPainType \n", fontsize=25)
plt.ylabel("Total Member" , fontsize = 20)
plt.xlabel("\n Name" , fontsize = 20)
# plt.yscale("log")
plt.xticks(rotation = 90)
for container in p.containers:
    p.bar_label(container,label_type = "center",padding = 6,size = 15,color = "black",rotation = 90,
    bbox={"boxstyle": "round", "pad": 0.6, "facecolor": "orange", "edgecolor": "black", "alpha": 1})

sns.despine(left=True, bottom=True)
plt.show()
```



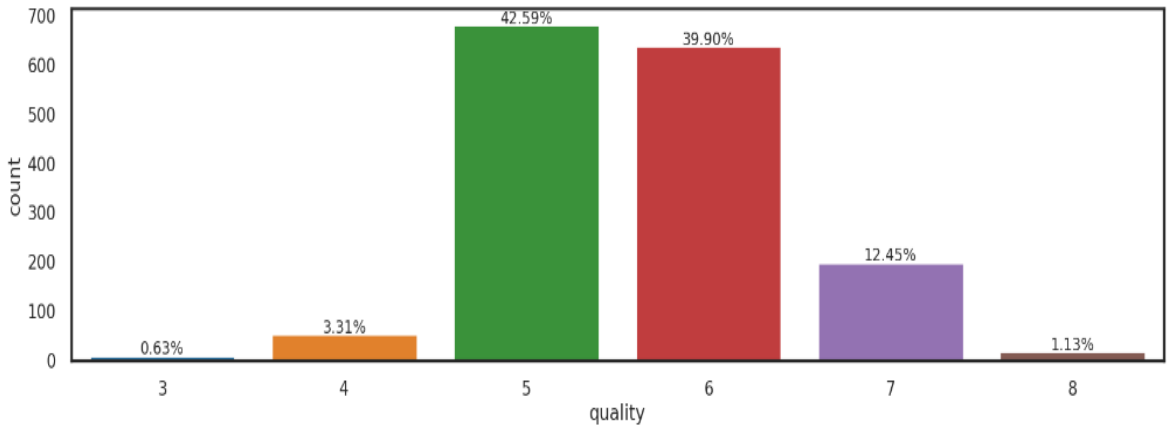
```
fig,axis=plt.subplots(ncols=4,nrows=3,figsize=(15,10))
index=0
axis=axis.flatten()

for col,values in wine.items():
    sns.boxplot(y=col,data=wine,color='r',ax=axis[index])
    index+=1
plt.tight_layout(pad=0.5,w_pad=0.7,h_pad=5.0);
```



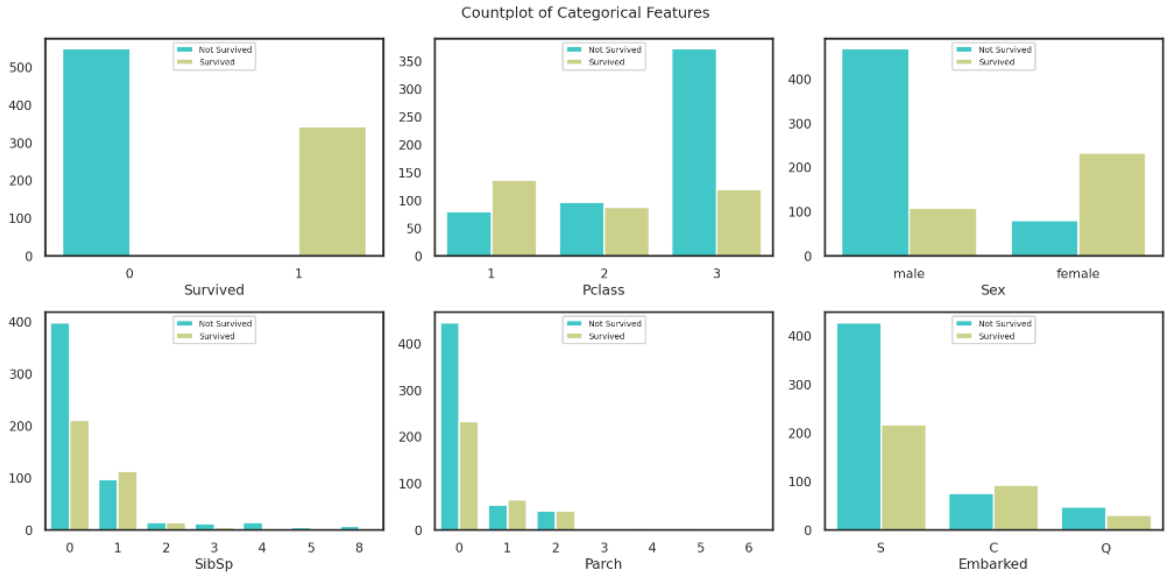

```
#checking the target variables for distribution
sns.distplot(heart['Cholesterol'],color='Brown')
plt.axvline(x=heart['Cholesterol'].mean(), color='Brown', lin
estyle='--', linewidth=2)
plt.title('Cholesterol');
```



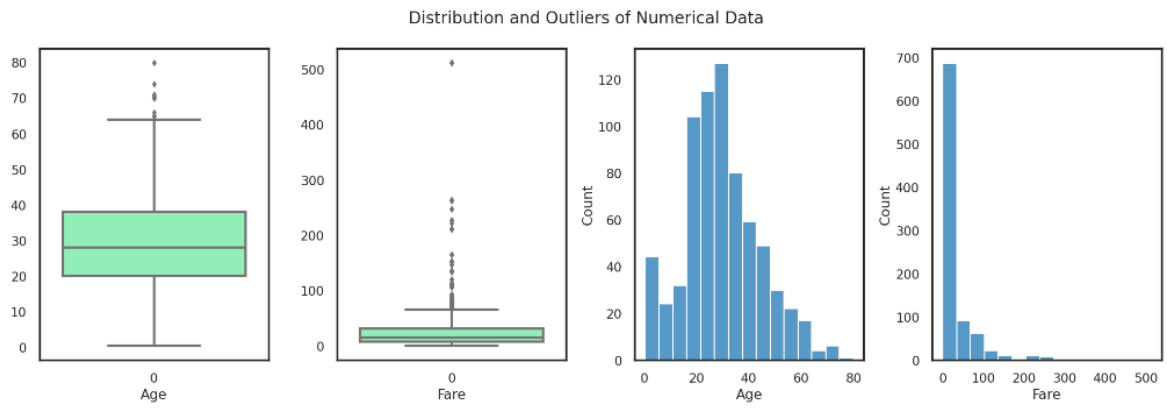
```
s = sns.countplot(x = 'quality',data = wine)
sizes=[]
for p in s.patches:
    height = p.get_height()
    sizes.append(height)
    s.text(p.get_x()+p.get_width()/2.,
           height + 3,
           '{:1.2f}%'.format(height/len(wine)*100),
           ha="center", fontsize=14)
```



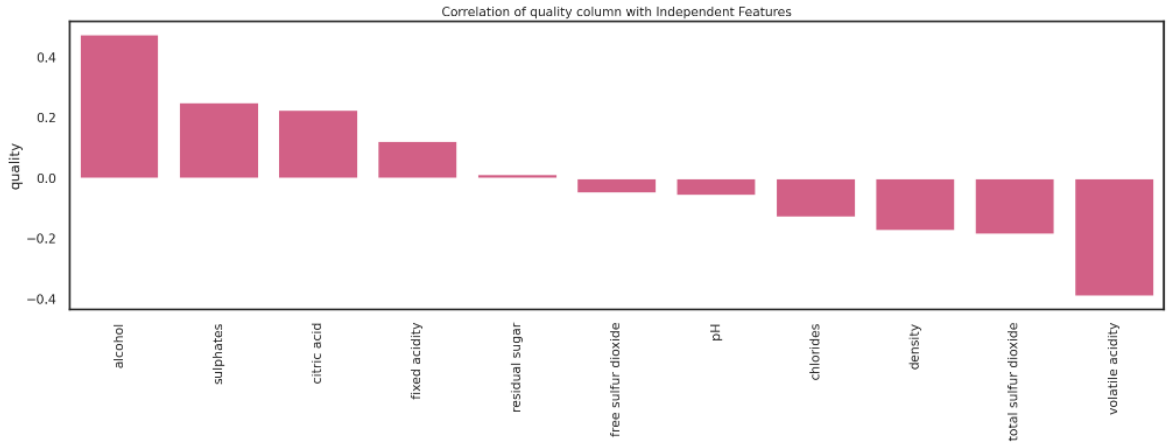

```
countfeature = ["Survived", "Pclass", "Sex", "SibSp", "Parch", "Embarked"]
countlist = list(enumerate(countfeature))

plt.figure(figsize = (20,10))
plt.suptitle("Countplot of Categorical Features", fontsize=18)
for i in countlist:
    plt.subplot(2,3,i[0]+1)
    sns.countplot(data = titanic, x = i[1], hue = "Survived", palette="rainbow")
    plt.ylabel("")
    plt.legend(['Not Survived', 'Survived'], loc='upper center', prop={'size': 10})
plt.tight_layout()
plt.show()
```



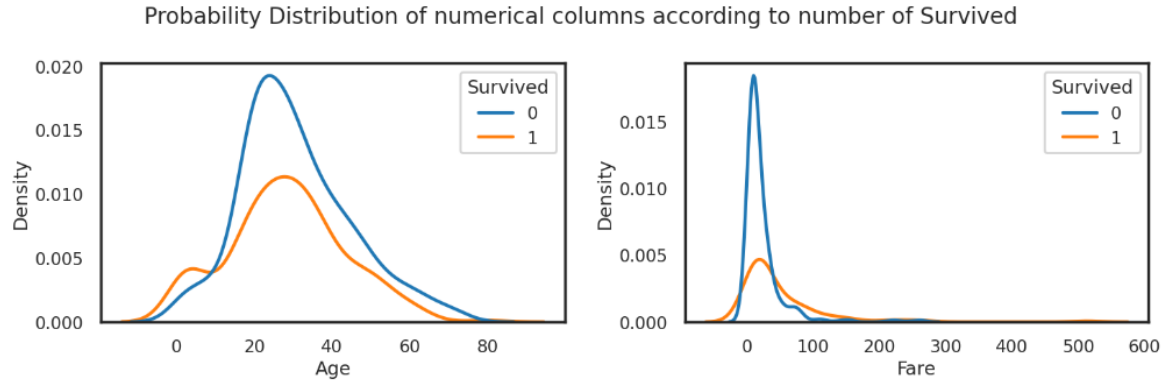

```
numfeature = ["Age", "Fare"]
enumfeat = list(enumerate(numfeature))

plt.figure(figsize=(20,7))
plt.suptitle("Distribution and Outliers of Numerical Data", fontsize=20)
for i in enumfeat:
    plt.subplot(1,4,i[0]+1)
    sns.boxplot(data = titanic[i[1]], palette="rainbow")
    plt.xlabel(str(i[1]))
for i in enumfeat:
    plt.subplot(1,4,i[0]+3)
    sns.histplot(data = titanic[i[1]], palette="rainbow", bins=15)
    plt.xlabel(str(i[1]))
plt.tight_layout()
plt.show()
```



```
plt.figure(figsize=(20,6))
plt.title("Correlation of quality column with Independent Features", f
ontsize=15)
corr = wine.corr()["quality"].sort_values(ascending=False)[1:]
sns.barplot(x=corr.index, y=corr, color=(0.90,0.30,0.50))
plt.tight_layout()
plt.xticks(rotation = 90)
plt.show()
```




```
plt.figure(figsize=(15,5))
plt.suptitle("Probability Distribution of numerical columns
according to number of Survived", fontsize = 20)
for i in enumfeat:
    plt.subplot(1,2,i[0]+1)
    sns.kdeplot(data=titanic, x=i[1], hue="Survived")
plt.tight_layout()

plt.show()
```



```
import missingno as msno
msno.matrix(titanic, color=(0.50,0.30,0.80))
plt.show()
x = titanic.isnull().sum()
for a, b in x.items():
    if b > 0:
        print(f"There are {b} missing values in column: {a}")
```

Titaic Data:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S

wine Data:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5

Heart Data:

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
0	40	M	ATA	140	289	0	Normal	172	N	0.0	Up	0
1	49	F	NAP	160	180	0	Normal	156	N	1.0	Flat	1
2	37	M	ATA	130	283	0	ST	98	N	0.0	Up	0