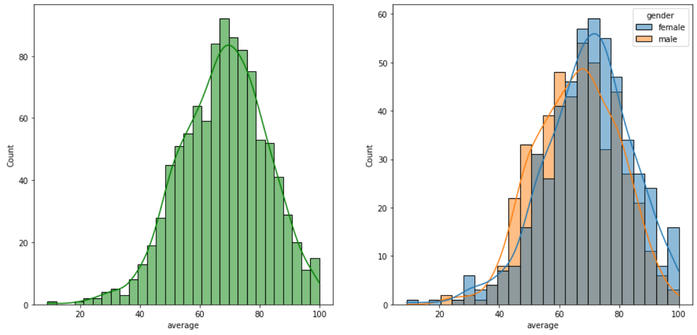
EDA (Exploratory Data Analysis) Cheatsheet

1. For Missing values

- df.isna().sum()
- df.duplicated().sum()   # check duplicated
- df.nunique()   # check number of unique values

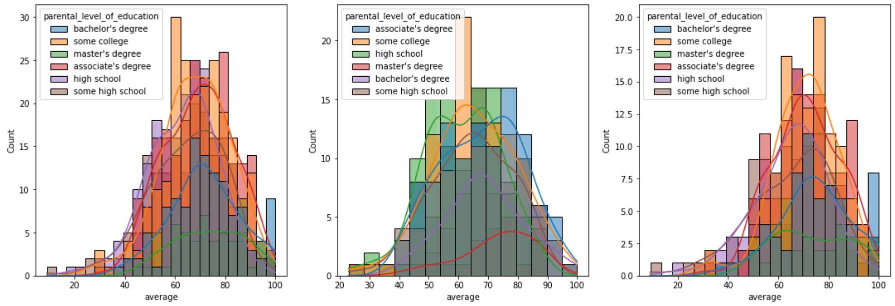
2. Histograms

-
  ```
  fig, axs = plt.subplots(1, 2, figsize=(15, 7))
  plt.subplot(121)
  sns.histplot(data=df,x='average',bins=30,kde=True,color='g')
  plt.subplot(122)
  sns.histplot(data=df,x='average',kde=True,hue='gender')
  plt.show()
  ```



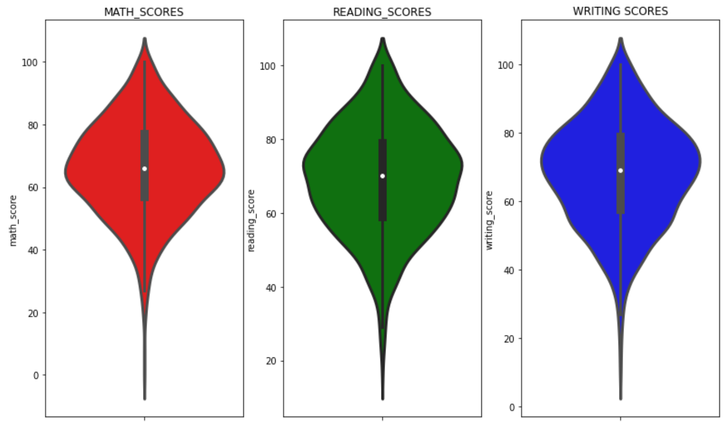
# another one with 3 in row

```
plt.subplots(1,3,figsize=(25,6))
plt.subplot(141)
ax =sns.histplot(data=df,x='average',kde=True,hue='race_ethnicity')
plt.subplot(142)
ax =sns.histplot(data=df[df.gender=='female'],x='average',kde=True,hue='race_ethnicity')
plt.subplot(143)
ax =sns.histplot(data=df[df.gender=='male'],x='average',kde=True,hue='race_ethnicity')
plt.show()
```



# For the violin Plot

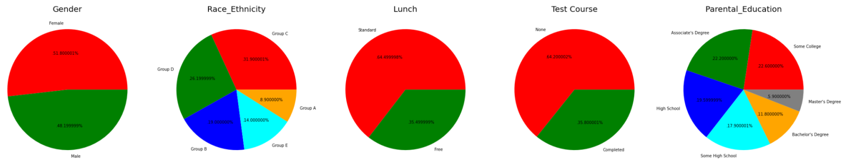
```
plt.figure(figsize=(18,8))
plt.subplot(1, 4, 1)
plt.title('MATH_SCORES')
sns.violinplot(y='math_score',data=df,color='red',linewidth=3)
plt.subplot(1, 4, 2)
plt.title('READING_SCORES')
sns.violinplot(y='reading_score',data=df,color='green',linewidth=3)
plt.subplot(1, 4, 3)
plt.title('WRITING SCORES')
sns.violinplot(y='writing_score',data=df,color='blue',linewidth=3)
plt.show()
```



# for making pie chart

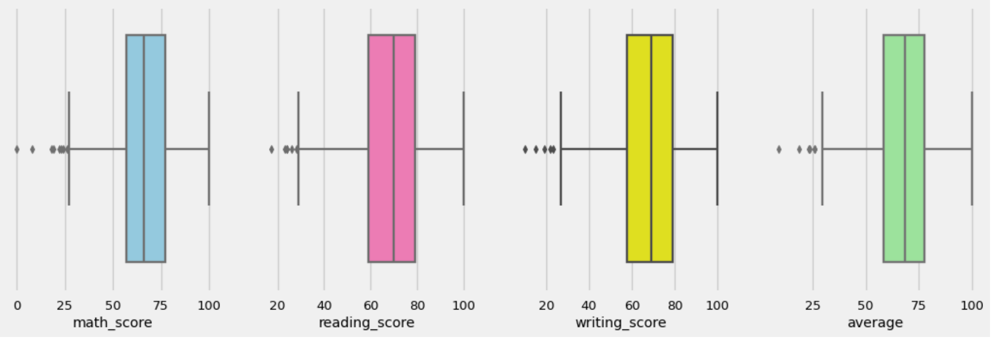
```
plt.subplot(1, 5, 1)
size = df['gender'].value_counts()
labels = 'Female', 'Male'
color = ['red','green']

plt.pie(size, colors = color, labels = labels,autopct = '.%2f%%')
plt.title('Gender', fontsize = 20)
plt.axis('off')
```



3. Checking the Outliers

```
plt.subplots(1,4,figsize=(16,5))
plt.subplot(141)
sns.boxplot(df['math_score'],color='skyblue')
plt.subplot(142)
sns.boxplot(df['reading_score'],color='hotpink')
plt.subplot(143)
sns.boxplot(df['writing_score'],color='yellow')
plt.subplot(144)
sns.boxplot(df['average'],color='lightgreen')
plt.show()
```

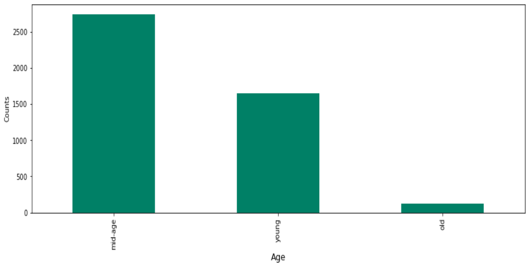


4. Pairplot

```
sns.pairplot(df,hue = 'gender')
plt.show()
```

5. Defining the functions to get counts

```
def get_counts(data):
   plt.figure(figsize = (15,5))
   plt.ylabel("Counts",fontsize = 10)
   return  data.value_counts().plot(kind = "bar",cmap='summer')
   plt.show()
```

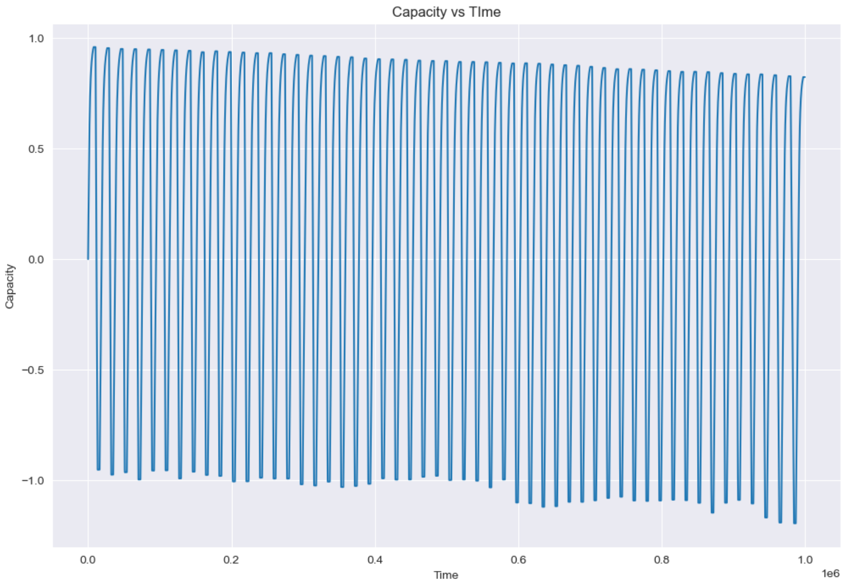


6. How to do One Hot Encoding

```
one_hot_encoded_data = pd.get_dummies(df, columns =
['lost_reason','room_type','des_country','des_city','source_country'])
```

7. How to plot normal graphs

```
sns.set_style("darkgrid")
plt.figure(figsize=(12, 8))
plt.plot(df['Time'],df['Capacity'])
plt.ylabel('Capacity')
adf = plt.gca().get_xaxis().get_major_formatter()
plt.xlabel('Time')
plt.title('Capacity vs TIme')
```



8. Removing 0 from the columns

```
df= df[df['status'] != 0] # removing 0 from status
```