In [163…

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import warnings
import warnings
warnings.filterwarnings('ignore')
```

Heart Disease Kaggle Dataset:

https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset

In [3]:

```python
data=pd.read_csv(r'E:\Data Analyst Project\heart.csv')
```

note: age sex chest pain type (4 values) Value 0: typical angina Value 1: atypical angina Value 2: non-anginal pain Value 3: asymptomatic trestbps: resting blood pressure (in mm Hg on admission to the hospital) chol: serum cholestoral in mg/dl fbs: (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false) restecg: resting electrocardiographic results Value 0: normal Value 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV) Value 2: showing probable or definite left ventricular hypertrophy by Estes' criteria thalach: maximum heart rate achieved exang: exercise induced angina (1 = yes; 0 = no) oldpeak = ST depression induced by exercise relative to rest slope: the slope of the peak exercise ST segment Value 1: upsloping Value 2: flat Value 3: downsloping ca: number of major vessels (0-3) colored by flourosopy thal: 3 = normal; 6 = fixed defect; 7 = reversable defect target : 0=less chance of heart attack, 1=more chance of heart attack

1. Show top and bottom rows of dataset

In [4]:

```python
data.head(5)
```

Out[4]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0

In [5]:

```python
data.tail(5)
```

Out[5]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	targ
1020	59	1	1	140	221	0	1	164	1	0.0	2	0	2	
1021	60	1	0	125	258	0	0	141	1	2.8	1	1	3	
1022	47	1	0	110	275	0	0	118	1	1.0	1	1	2	
1023	50	0	0	110	254	0	0	159	0	0.0	2	0	2	
1024	54	1	0	120	188	0	1	113	0	1.4	1	1	3	

2. Check the null values of dataset

In [6]:

```
data.isnull().sum()
```

Out[6]:

```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0
dtype: int64
```

3. Drop duplicate values

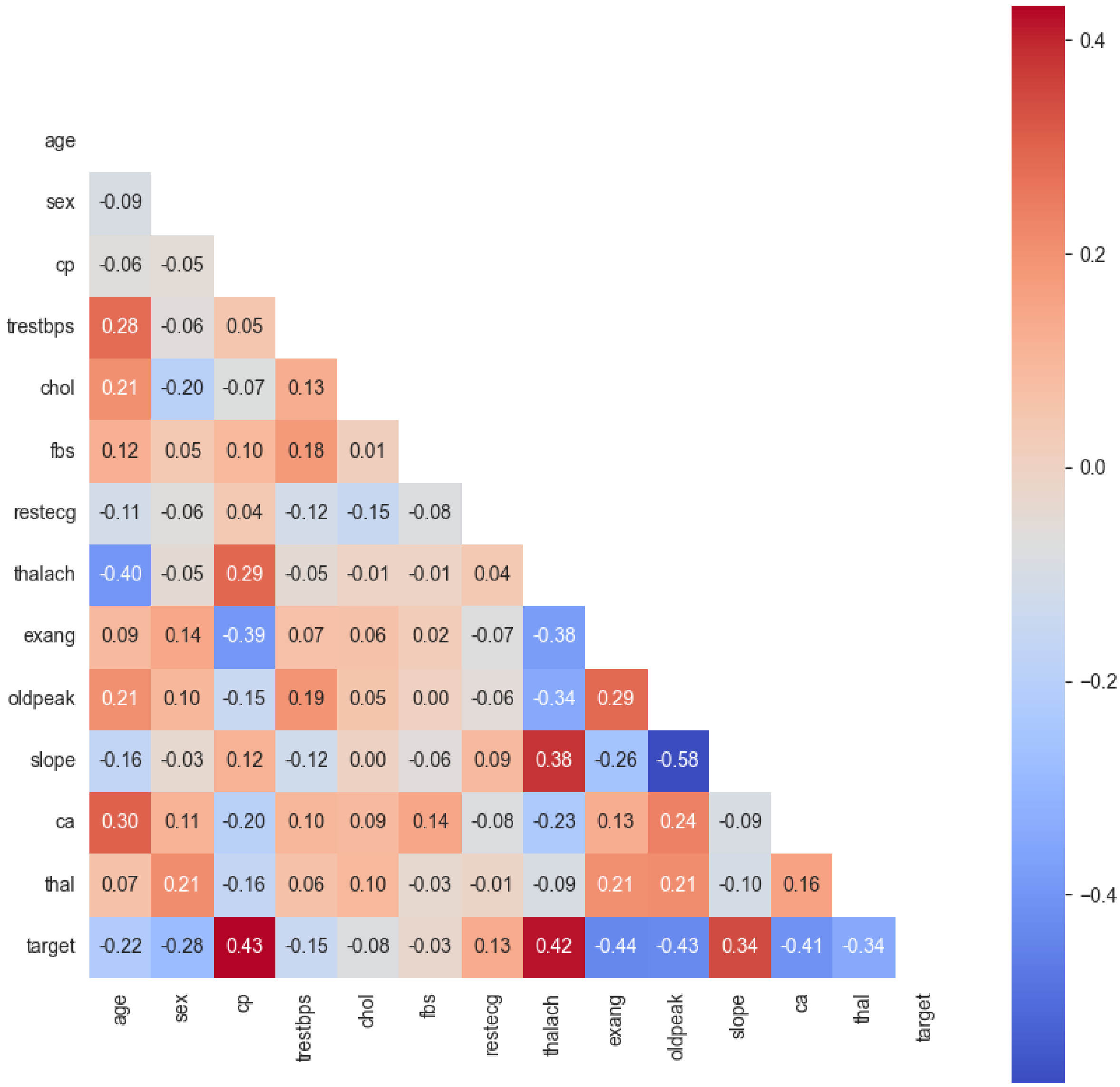
In [7]:

```
data=data.drop_duplicates()
```

4. Visualization Data Correlation

In [8]:

```
correlation_matrix=data.corr()
sns.set_style("white")
mask = np.triu(np.ones_like(correlation_matrix, dtype=bool))
fig, ax = plt.subplots(figsize=(10, 10))
sns.heatmap(correlation_matrix, annot=True, fmt='.2f', cmap='coolwarm', mask
plt.show()
```

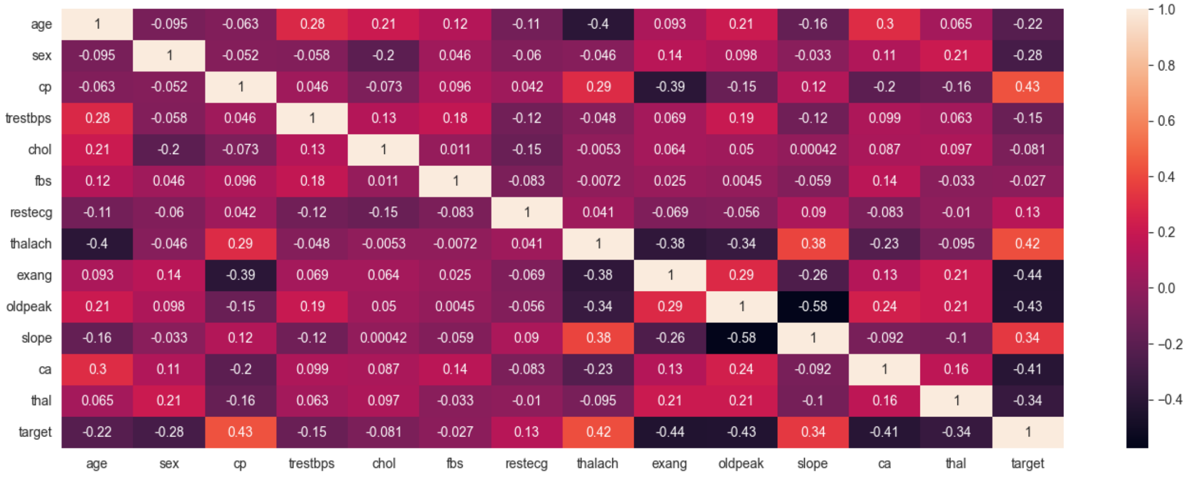


or we can use heatmap

```
In [10]: plt.figure(figsize=(17,6))
         sns.heatmap(correlation_matrix, annot=True)
```

```
Out[10]: <Axes: >
```

Loading [MathJax]/extensions/Safe.js



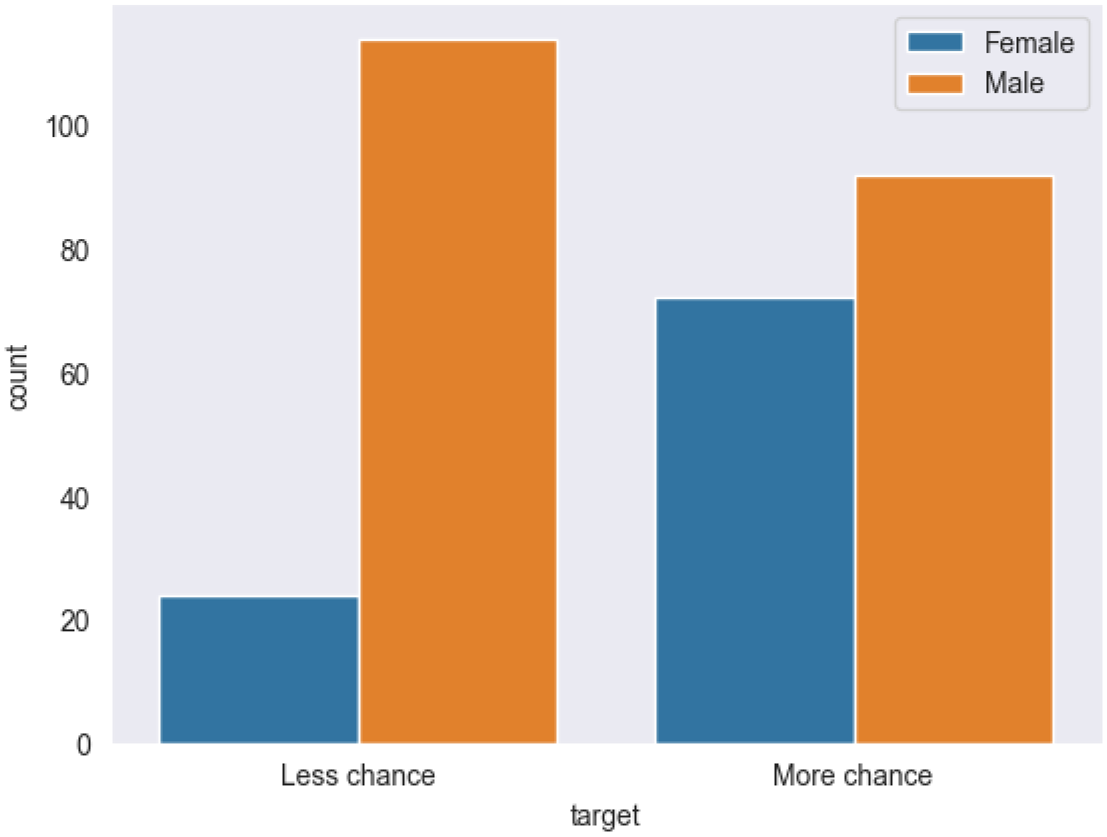
	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
age	1	-0.095	-0.063	0.28	0.21	0.12	-0.11	-0.4	0.093	0.21	-0.16	0.3	0.065	-0.22
sex	-0.095	1	-0.052	-0.058	-0.2	0.046	-0.06	-0.046	0.14	0.098	-0.033	0.11	0.21	-0.28
cp	-0.063	-0.052	1	0.046	-0.073	0.096	0.042	0.29	-0.39	-0.15	0.12	-0.2	-0.16	0.43
trestbps	0.28	-0.058	0.046	1	0.13	0.18	-0.12	-0.048	0.069	0.19	-0.12	0.099	0.063	-0.15
chol	0.21	-0.2	-0.073	0.13	1	0.011	-0.15	-0.0053	0.064	0.05	0.00042	0.087	0.097	-0.081
fbs	0.12	0.046	0.096	0.18	0.011	1	-0.083	-0.0072	0.025	0.0045	-0.059	0.14	-0.033	-0.027
restecg	-0.11	-0.06	0.042	-0.12	-0.15	-0.083	1	0.041	-0.069	-0.056	0.09	-0.083	-0.01	0.13
thalach	-0.4	-0.046	0.29	-0.048	-0.0053	-0.0072	0.041	1	-0.38	-0.34	0.38	-0.23	-0.095	0.42
exang	0.093	0.14	-0.39	0.069	0.064	0.025	-0.069	-0.38	1	0.29	-0.26	0.13	0.21	-0.44
oldpeak	0.21	0.098	-0.15	0.19	0.05	0.0045	-0.056	-0.34	0.29	1	-0.58	0.24	0.21	-0.43
slope	-0.16	-0.033	0.12	-0.12	0.00042	-0.059	0.09	0.38	-0.26	-0.58	1	-0.092	-0.1	0.34
ca	0.3	0.11	-0.2	0.099	0.087	0.14	-0.083	-0.23	0.13	0.24	-0.092	1	0.16	-0.41
thal	0.065	0.21	-0.16	0.063	0.097	-0.033	-0.01	-0.095	0.21	0.21	-0.1	0.16	1	-0.34
target	-0.22	-0.28	0.43	-0.15	-0.081	-0.027	0.13	0.42	-0.44	-0.43	0.34	-0.41	-0.34	1

5. 'People which has heart disease or not' visualization

In [12]:
```
data.columns
```

Out[12]:
```
Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
       'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
      dtype='object')
```

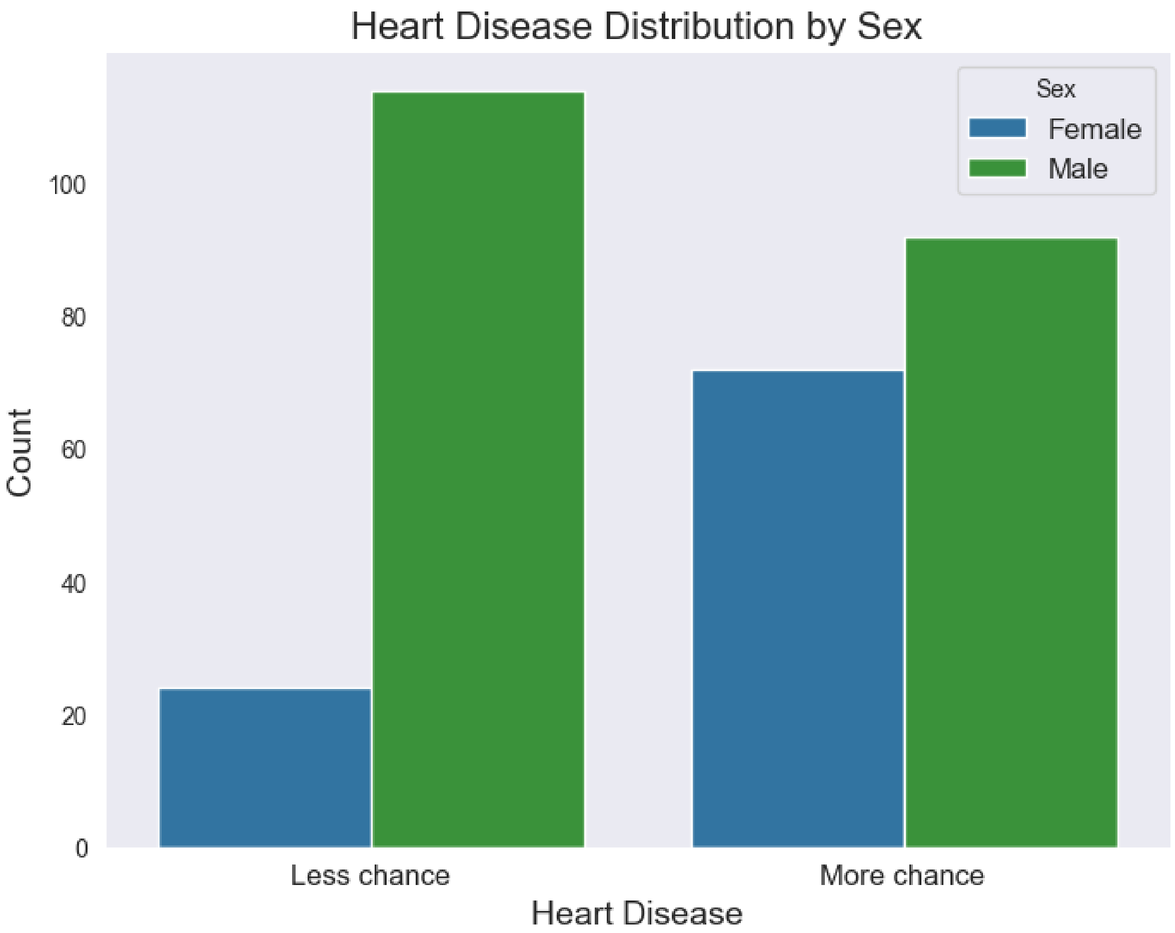
In [57]:
```
sns.countplot(x=data['target'], hue=data['sex'])
plt.xticks([0,1],['Less chance','More chance'])
plt.legend(labels=['Female','Male'])
plt.show()
```



Let's change the style

In [56]:

```
# set the style of the plot
sns.set_style("dark")
custom_palette = sns.color_palette(['#1f77b4', '#2ca02c'])

# create a figure and axis object
fig, ax = plt.subplots(figsize=(8, 6))

# create the countplot
sns.countplot(x=data['target'], hue=data['sex'], palette=custom_palette, ax=

# set the labels and legend
ax.set_xticklabels(['Less chance', 'More chance'], fontsize=12)
ax.set_xlabel('Heart Disease', fontsize=14)
ax.set_ylabel('Count', fontsize=14)
ax.legend(title='Sex', labels=['Female', 'Male'], fontsize=12)

# add a title to the plot
ax.set_title('Heart Disease Distribution by Sex', fontsize=16)

# remove the top and right spines
sns.despine()

# show the plot
plt.show()
```



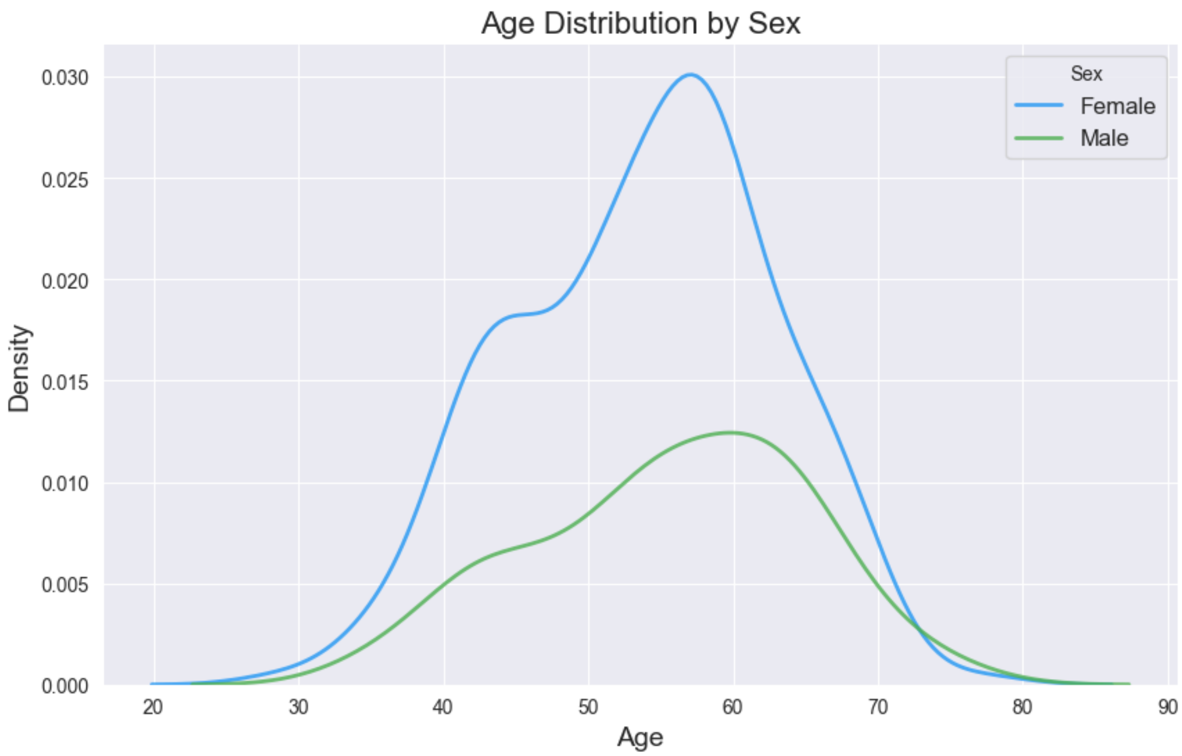
6. Age distribution of Heart Disease

In [101…

```
# set the style and color palette of the plot
sns.set_style("darkgrid")
palette = sns.color_palette(["#4CAF50", "#2196F3"])
sns.set_palette(palette)

# create a figure and axis object
fig, ax = plt.subplots(figsize=(10, 6))

# create the distribution plot
sns.kdeplot(data=data, x="age", hue="sex", multiple='layer', palette=palette

# set the labels and legend
ax.set_xlabel('Age', fontsize=14)
ax.set_ylabel('Density', fontsize=14)
ax.legend(title='Sex', labels=['Female', 'Male'], fontsize=12)

# add a title to the plot
ax.set_title('Age Distribution by Sex', fontsize=16)

# remove the top and right spines
sns.despine()
```

Loading [MathJax]/extensions/Safe.js

```
# show the plot
plt.show()
```

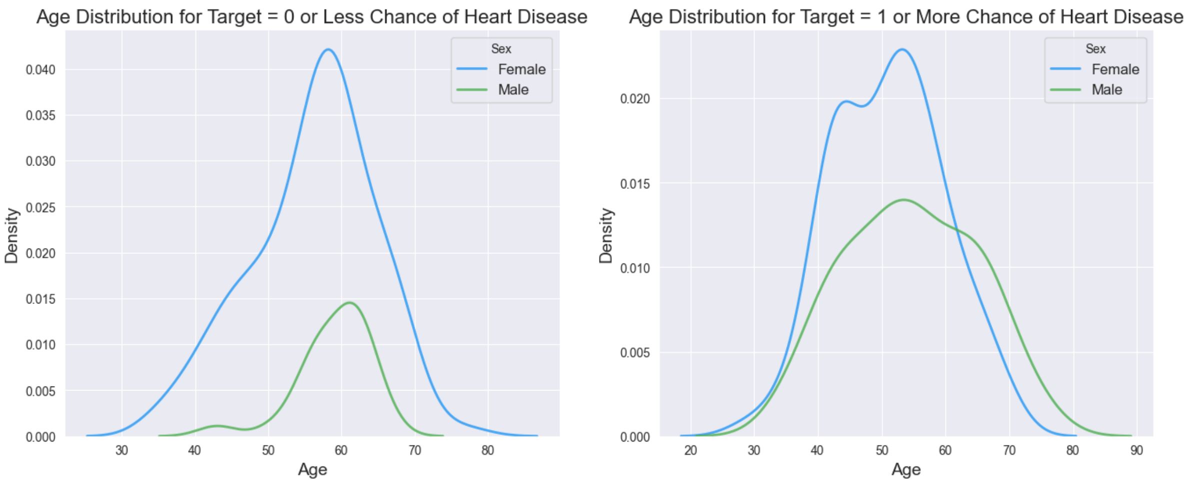


In [103…

```
# set the custom color palette
palette = sns.color_palette(["#4CAF50", "#2196F3"])

# set the style and color palette of the plot
sns.set_style("darkgrid")
sns.set_palette(palette)

# create a figure and axis object with two subplots
fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(16, 6))

# create the distribution plots for target = 0 and target = 1
sns.kdeplot(data=data[data['target']==0], x="age", hue="sex", multiple="laye
sns.kdeplot(data=data[data['target']==1], x="age", hue="sex", multiple="laye

# set the labels and legends for each subplot
ax1.set_xlabel('Age', fontsize=14)
ax1.set_ylabel('Density', fontsize=14)
ax1.set_title('Age Distribution for Target = 0 or Less Chance of Heart Disea
ax1.legend(title='Sex', labels=['Female', 'Male'], fontsize=12)

ax2.set_xlabel('Age', fontsize=14)
ax2.set_ylabel('Density', fontsize=14)
ax2.set_title('Age Distribution for Target = 1 or More Chance of Heart Disea
ax2.legend(title='Sex', labels=['Female', 'Male'], fontsize=12)

# remove the top and right spines from each subplot
sns.despine(ax=ax1)
sns.despine(ax=ax2)
```

Loading [MathJax]/extensions/Safe.js

```
# show the plot
plt.show()
```



7. Chest pain type visualization

chest pain type (4 values) Value 0: typical angina Value 1: atypical angina Value 2: non-anginal pain Value 3: asymptomatic

In [104…

```
data.columns
```

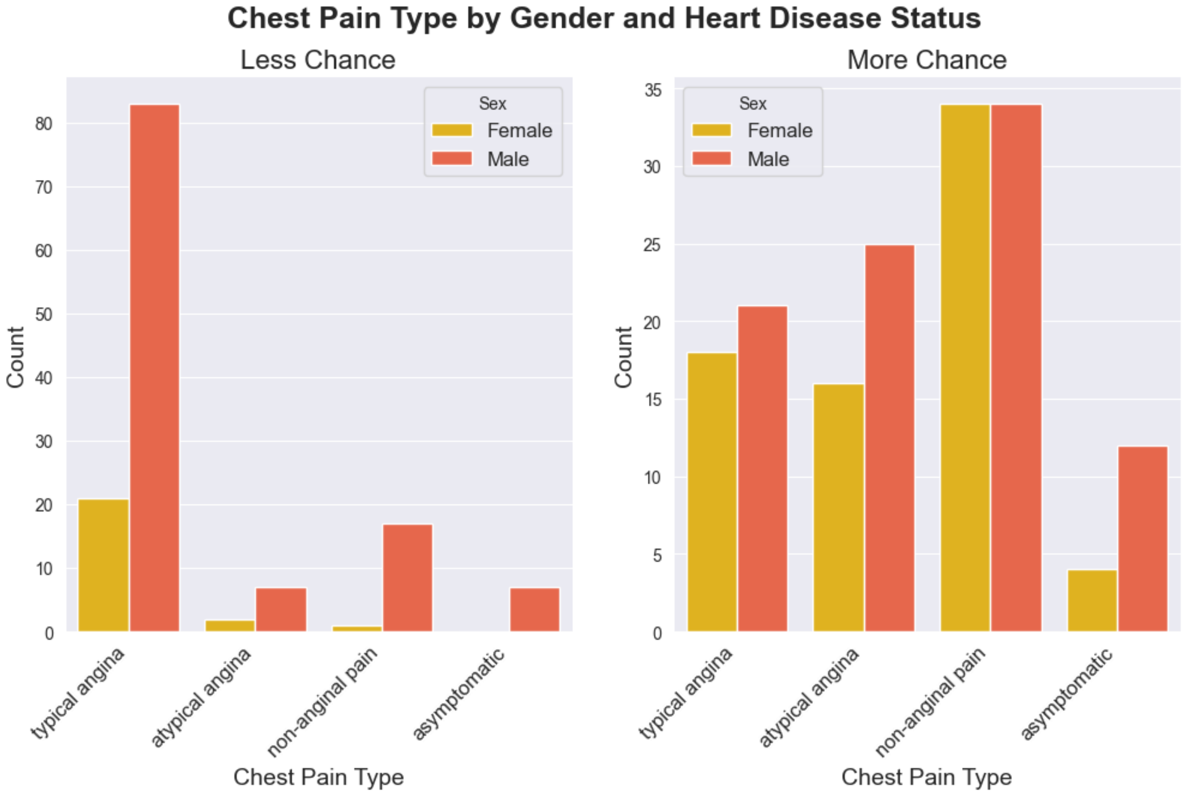
Out[104]:

```
Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalac
h',
       'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
      dtype='object')
```

In [122…

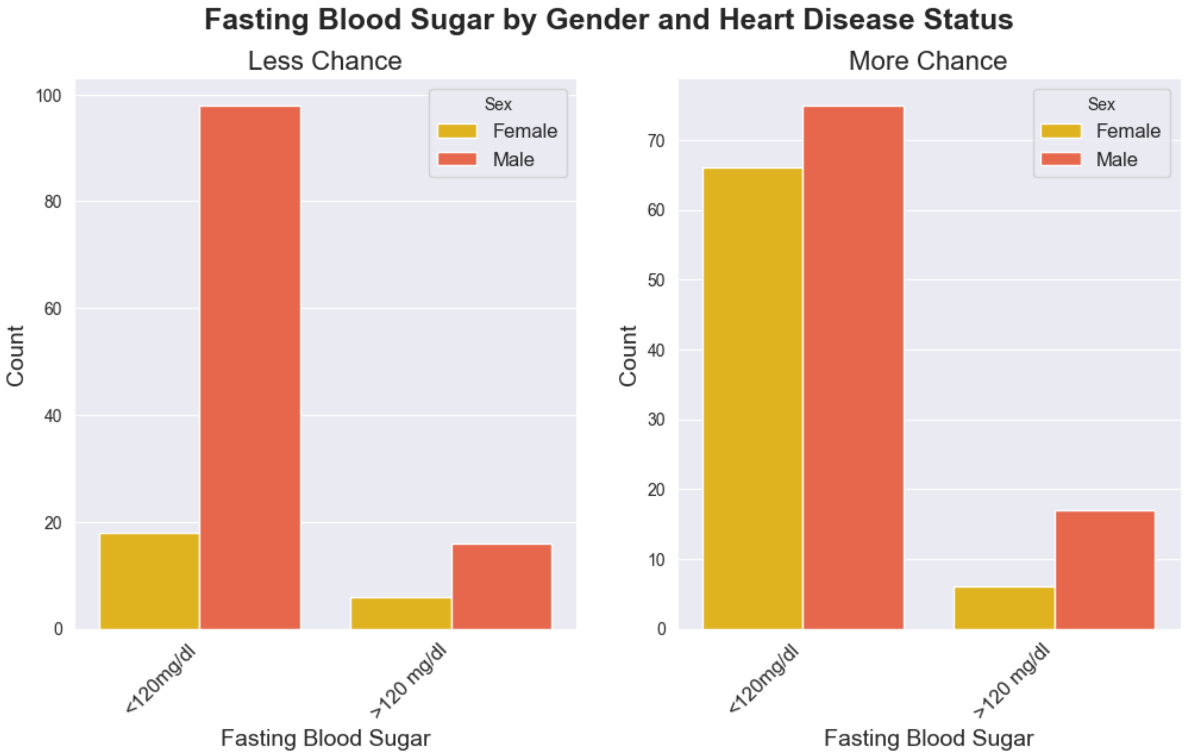
```
# create a custom color palette with shades of orange and red
palette = sns.color_palette(["#FFC300", "#FF5733"])

# set the style and color palette of the plot
sns.set_style("darkgrid")
sns.set_palette(palette)

# create a figure and axis object with two subplots
fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(12, 6))

# create the count plots for target = 0 and target = 1
sns.countplot(x=data[data['target']==0]['cp'], hue=data[data['target']==0]['
sns.countplot(x=data[data['target']==1]['cp'], hue=data[data['target']==1]['

# set the labels and legends for each subplot
ax1.set_xlabel('Chest Pain Type', fontsize=14)
ax1.set_ylabel('Count', fontsize=14)
ax1.set_title('Less Chance', fontsize=16)
ax1.legend(title='Sex', labels=['Female', 'Male'], fontsize=12)

ax2.set_xlabel('Chest Pain Type', fontsize=14)
ax2.set_ylabel('Count', fontsize=14)
ax2.set_title('More Chance', fontsize=16)
ax2.legend(title='Sex', labels=['Female', 'Male'], fontsize=12)
```

```
# remove the top and right spines from each subplot
sns.despine(ax=ax1)
sns.despine(ax=ax2)

# set the x-tick labels for the chest pain types
for ax in [ax1, ax2]:
    ax.set_xticklabels(['typical angina','atypical angina','non-anginal pair

# set the title
plt.suptitle('Chest Pain Type by Gender and Heart Disease Status', fontsize=

# show the plot
plt.show()
```



8. Fasting blood pressure visualization

```
In [127…
# create a custom color palette with shades of orange and red
palette = sns.color_palette(["#FFC300", "#FF5733"])

# set the style and color palette of the plot
sns.set_style("darkgrid")
sns.set_palette(palette)

# create a figure and axis object with two subplots
fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(12, 6))

# create the count plots for target = 0 and target = 1
sns.countplot(x=data[data['target']==0]['fbs'], hue=data[data['target']==0][
sns.countplot(x=data[data['target']==1]['fbs'], hue=data[data['target']==1][
# set the labels and legends for each subplot
ax1.set_xlabel('Fasting Blood Sugar', fontsize=14)
ax1.set_ylabel('Count', fontsize=14)
ax1.set_title('Less Chance', fontsize=16)
ax1.legend(title='Sex', labels=['Female', 'Male'], fontsize=12)

ax2.set_xlabel('Fasting Blood Sugar', fontsize=14)
ax2.set_ylabel('Count', fontsize=14)
ax2.set_title('More Chance', fontsize=16)
ax2.legend(title='Sex', labels=['Female', 'Male'], fontsize=12)

# remove the top and right spines from each subplot
sns.despine(ax=ax1)
sns.despine(ax=ax2)

# set the x-tick labels for the fasting blood sugar values
for ax in [ax1, ax2]:
    ax.set_xticklabels(['<120mg/dl','>120 mg/dl'], fontsize=12, rotation=45,

# set the title
plt.suptitle('Fasting Blood Sugar by Gender and Heart Disease Status', fonts

# show the plot
plt.show()
```

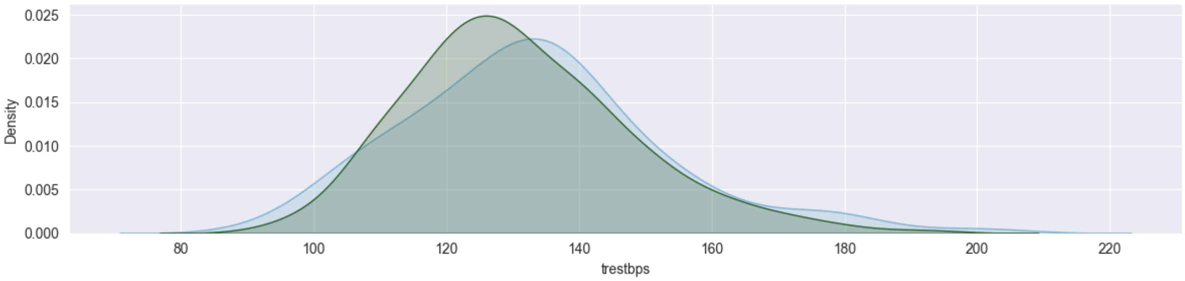


9. Resting Blood Pressure visualization

In [166…

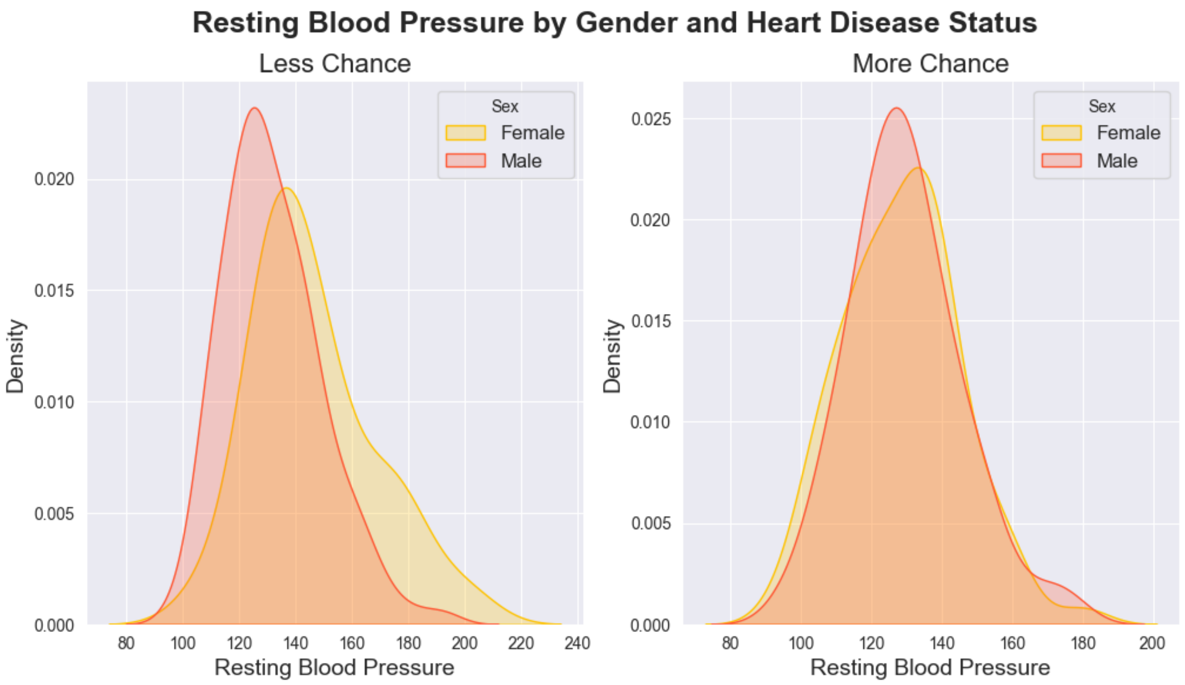
```
g = sns.FacetGrid(data,hue='sex',aspect=4)
g.map(sns.kdeplot, 'trestbps',shade=True )
```

```
Out[166]: <seaborn.axisgrid.FacetGrid at 0x1e989a93a00>
```



```
In [164…
# create a custom color palette with shades of orange and red
palette = sns.color_palette(["#FFC300", "#FF5733"])

# set the style and color palette of the plot
sns.set_style("darkgrid")
sns.set_palette(palette)

# create a figure and axis object with two subplots
fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(12, 6))

# create the kde plots for target = 0 and target = 1
sns.kdeplot(data[data['target']==0][data['sex']==0]['trestbps'], shade=True,
sns.kdeplot(data[data['target']==0][data['sex']==1]['trestbps'], shade=True,
sns.kdeplot(data[data['target']==1][data['sex']==0]['trestbps'], shade=True,
sns.kdeplot(data[data['target']==1][data['sex']==1]['trestbps'], shade=True,

# set the labels and legends for each subplot
ax1.set_xlabel('Resting Blood Pressure', fontsize=14)
ax1.set_ylabel('Density', fontsize=14)
ax1.set_title('Less Chance', fontsize=16)
ax1.legend(title='Sex', fontsize=12)

ax2.set_xlabel('Resting Blood Pressure', fontsize=14)
ax2.set_ylabel('Density', fontsize=14)
ax2.set_title('More Chance', fontsize=16)
ax2.legend(title='Sex', fontsize=12)

# remove the top and right spines from each subplot
sns.despine(ax=ax1)
sns.despine(ax=ax2)

# set the title
plt.suptitle('Resting Blood Pressure by Gender and Heart Disease Status', fc

# show the plot
plt.show()
```

Loading [MathJax]/extensions/Safe.js

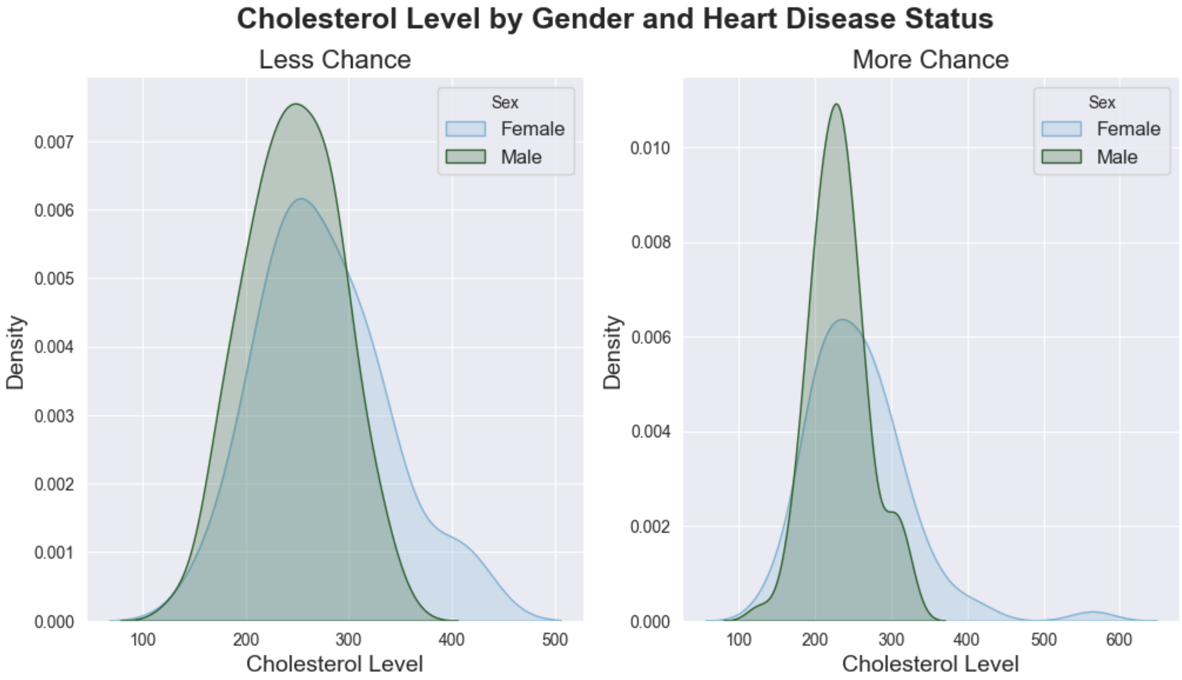


In [165…

```
# create a custom color palette with shades of blue and green
palette = sns.color_palette(["#7FB3D5", "#2C5F2D"])

# set the style and color palette of the plot
sns.set_style("darkgrid")
sns.set_palette(palette)

# create a figure and axis object with two subplots
fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(12, 6))

# create the KDE plots for target = 0 and target = 1
sns.kdeplot(x=data[data['target']==0][data['sex']==0]['chol'], shade=True, l
sns.kdeplot(x=data[data['target']==0][data['sex']==1]['chol'], shade=True, l

sns.kdeplot(x=data[data['target']==1][data['sex']==0]['chol'], shade=True, l
sns.kdeplot(x=data[data['target']==1][data['sex']==1]['chol'], shade=True, l

# set the labels and legends for each subplot
ax1.set_xlabel('Cholesterol Level', fontsize=14)
ax1.set_ylabel('Density', fontsize=14)
ax1.set_title('Less Chance', fontsize=16)
ax1.legend(title='Sex', fontsize=12)

ax2.set_xlabel('Cholesterol Level', fontsize=14)
ax2.set_ylabel('Density', fontsize=14)
ax2.set_title('More Chance', fontsize=16)
ax2.legend(title='Sex', fontsize=12)

# remove the top and right spines from each subplot
sns.despine(ax=ax1)
sns.despine(ax=ax2)

# set the title
plt.suptitle('Cholesterol Level by Gender and Heart Disease Status', fontsiz
```

Loading [MathJax]/extensions/Safe.js

```python
# show the plot
plt.show()
```



In [ ]:

Loading [MathJax]/extensions/Safe.js