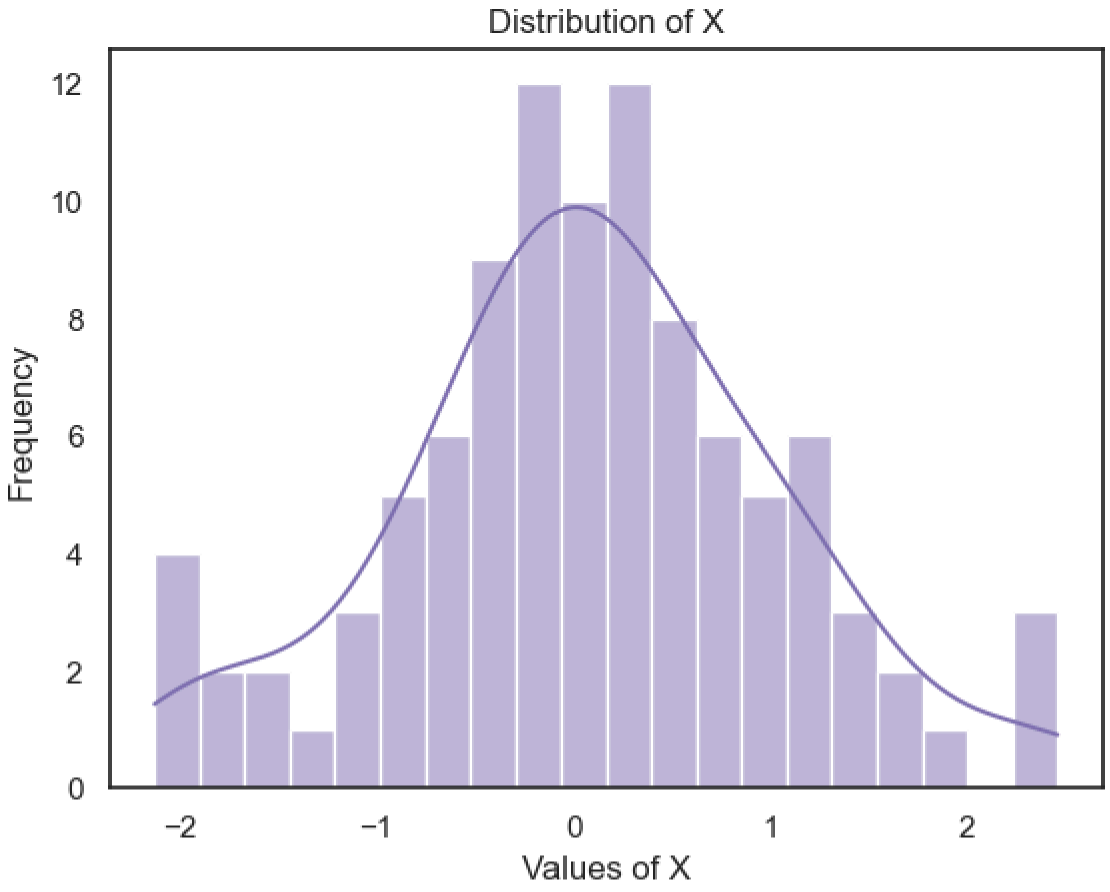
Histogram Plot

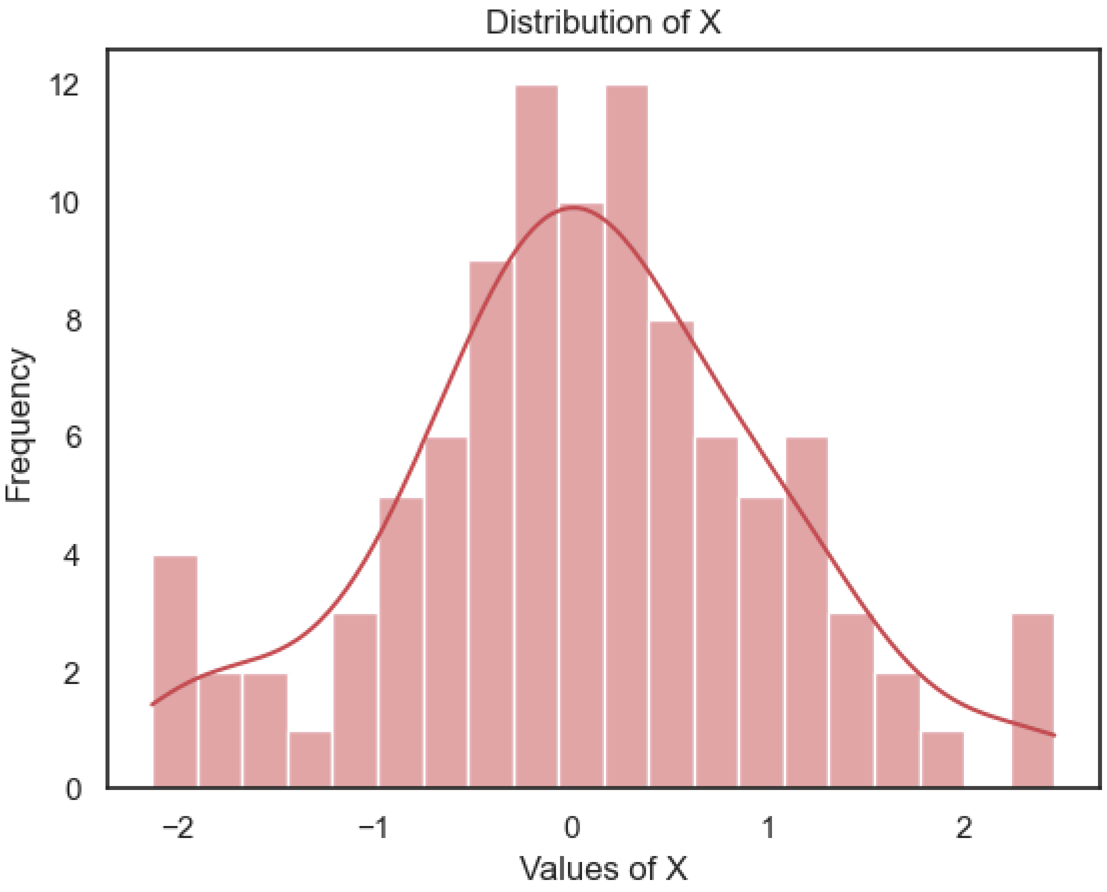
In [1]:

```python
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
sns.set(style="white")
## Grnerate a Random Sample from Univariate Normal Distribution
rs=np.random.RandomState(10)
x=rs.normal(size=100)
# Plot a simple histogram and kde
sns.histplot(x,kde=True,color="m",bins=20)
plt.xlabel("Values of X")
plt.ylabel("Frequency")
plt.title("Distribution of X")
plt.show()
```



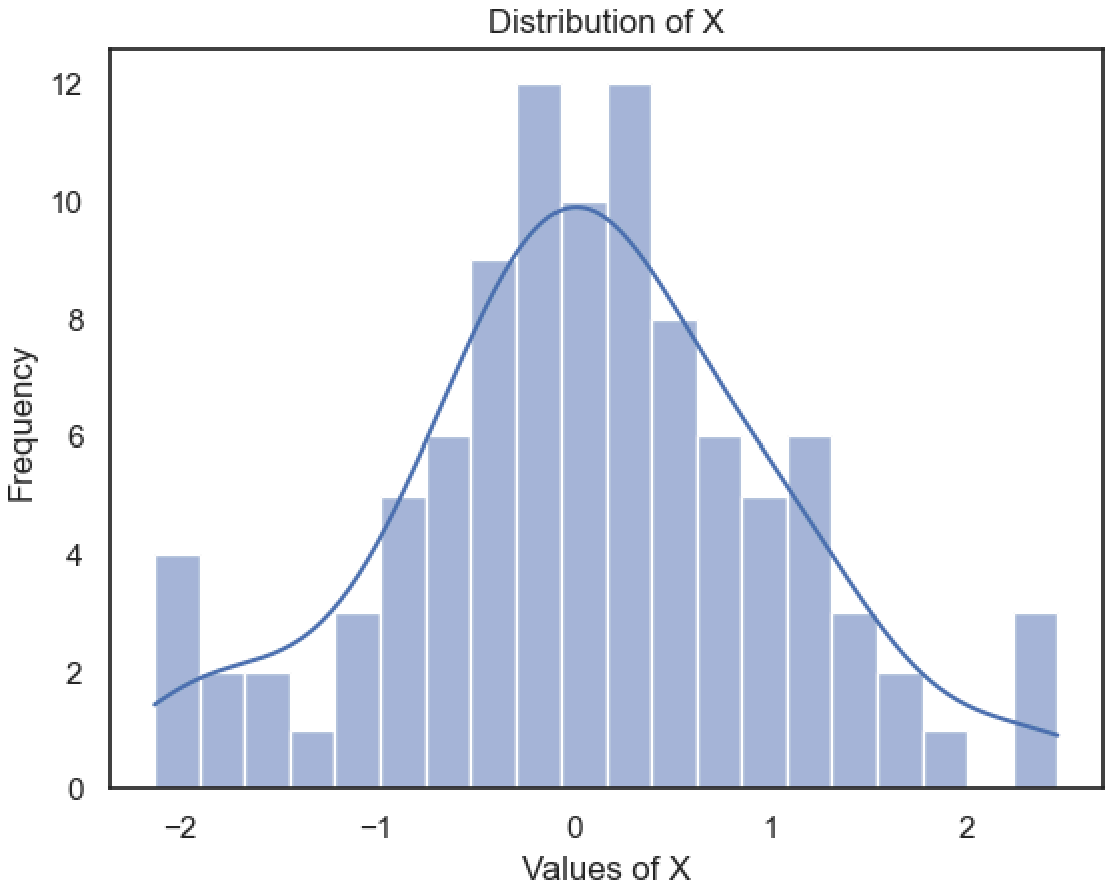
In [2]:

```
import numpy as np
import seaborn as sns
sns.set(style="white")
## Grnerate a Random Sample from Univariate Normal Distribution
rs=np.random.RandomState(10)
x=rs.normal(size=100)
# Plot a simple histogram and kde
sns.histplot(x,kde=True,color="r",bins=20)
plt.xlabel("Values of X")
plt.ylabel("Frequency")
plt.title("Distribution of X")
plt.show()
```



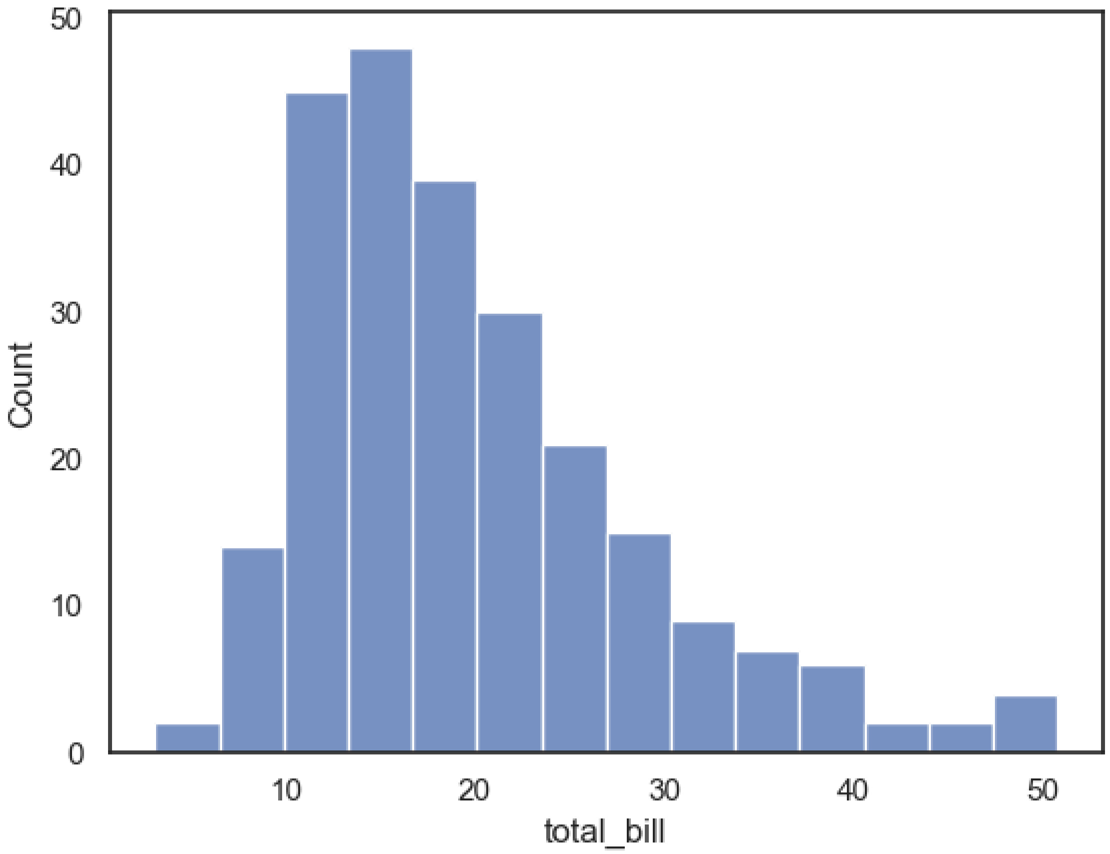
In [3]:

```
import numpy as np
import seaborn as sns
sns.set(style="white")
## Grnerate a Random Sample from Univariate Normal Distribution
rs=np.random.RandomState(10)
x=rs.normal(size=100)
# Plot a simple histogram and kde
sns.histplot(x,kde=True,color="b",bins=20)
plt.xlabel("Values of X")
plt.ylabel("Frequency")
plt.title("Distribution of X")
plt.show()
```



```
In [4]: import seaborn as sns
        # Load the Tips Dataset
        tips=sns.load_dataset("tips")
        # Create a histogram of the total bill amounts
        sns.histplot(data=tips,x="total_bill")
```

```
Out[4]: <Axes: xlabel='total_bill', ylabel='Count'>
```



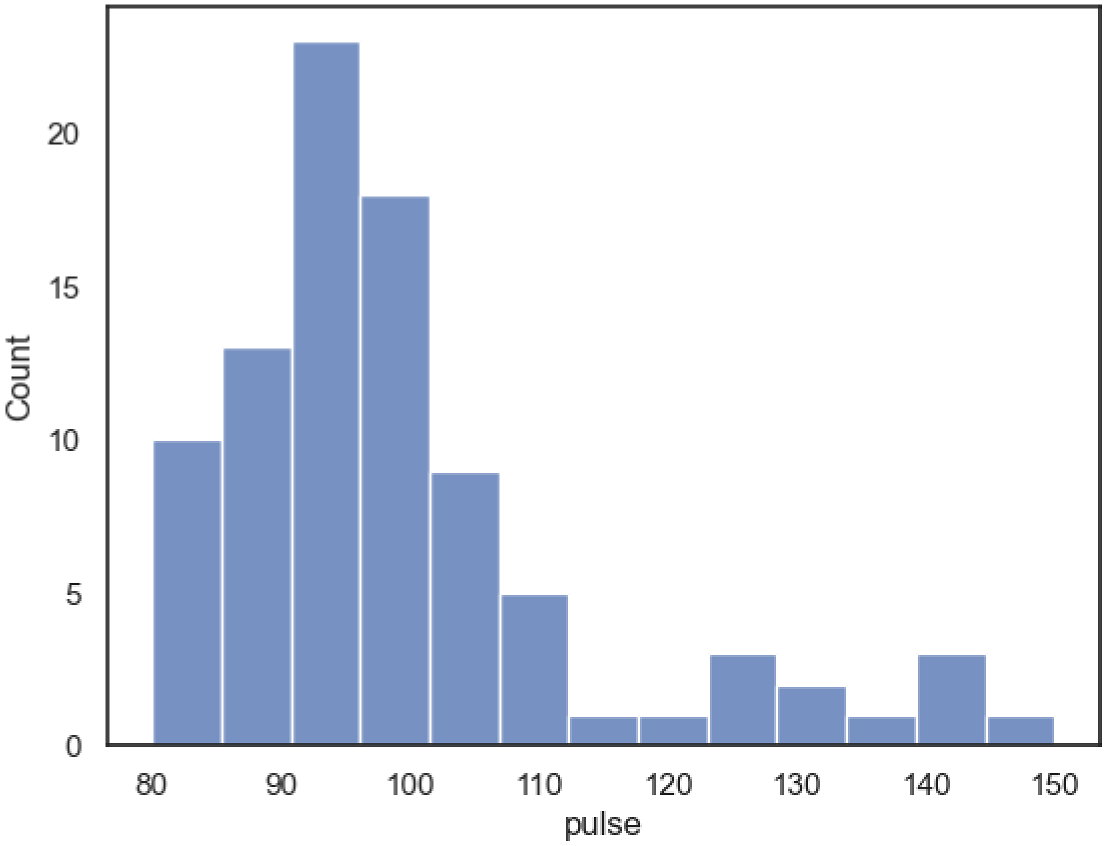
```
In [5]: import seaborn as sns
        # Load the exercise dataset
        exercise = sns.load_dataset("exercise")
        # check the head
        exercise.head()
```

Out[5]:

	Unnamed: 0	id	diet	pulse	time	kind
0	0	1	low fat	85	1 min	rest
1	1	1	low fat	85	15 min	rest
2	2	1	low fat	88	30 min	rest
3	3	2	low fat	90	1 min	rest
4	4	2	low fat	92	15 min	rest

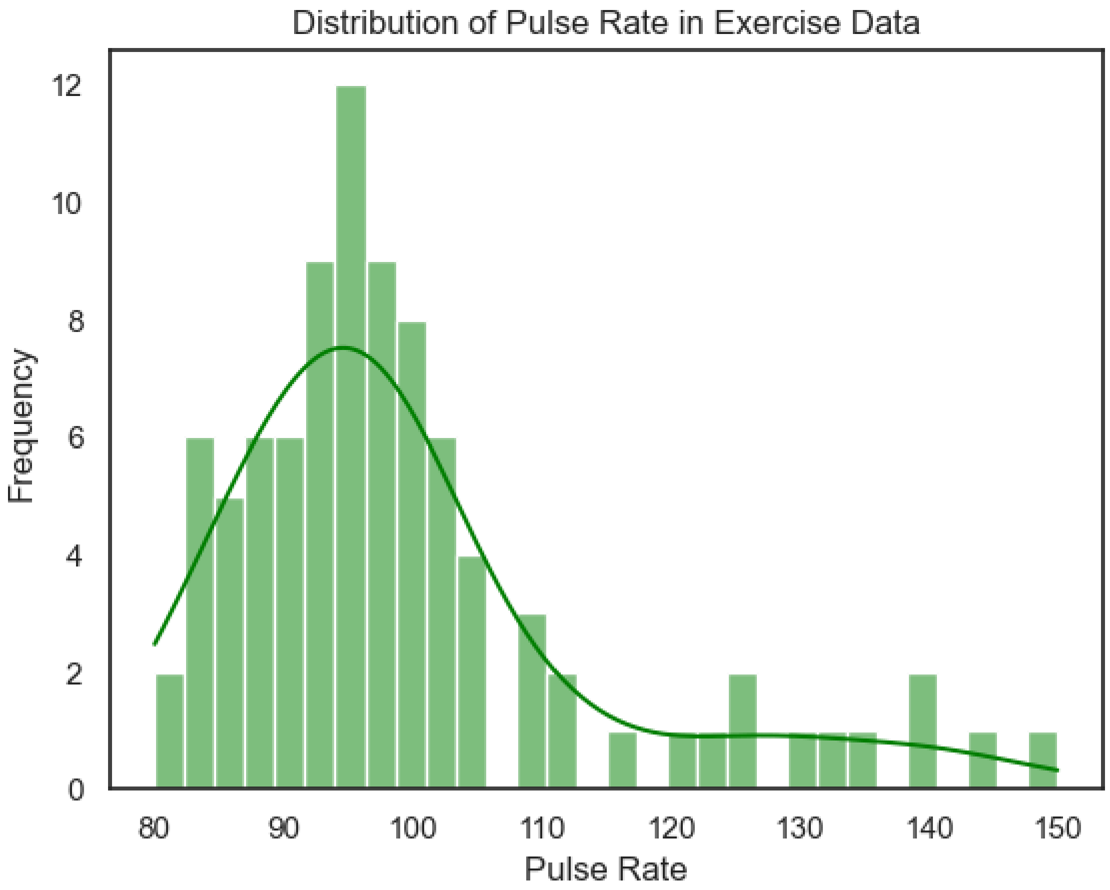
In [6]:
```
sns.histplot(data=exercise,x="pulse")
```

Out[6]:
```
<Axes: xlabel='pulse', ylabel='Count'>
```



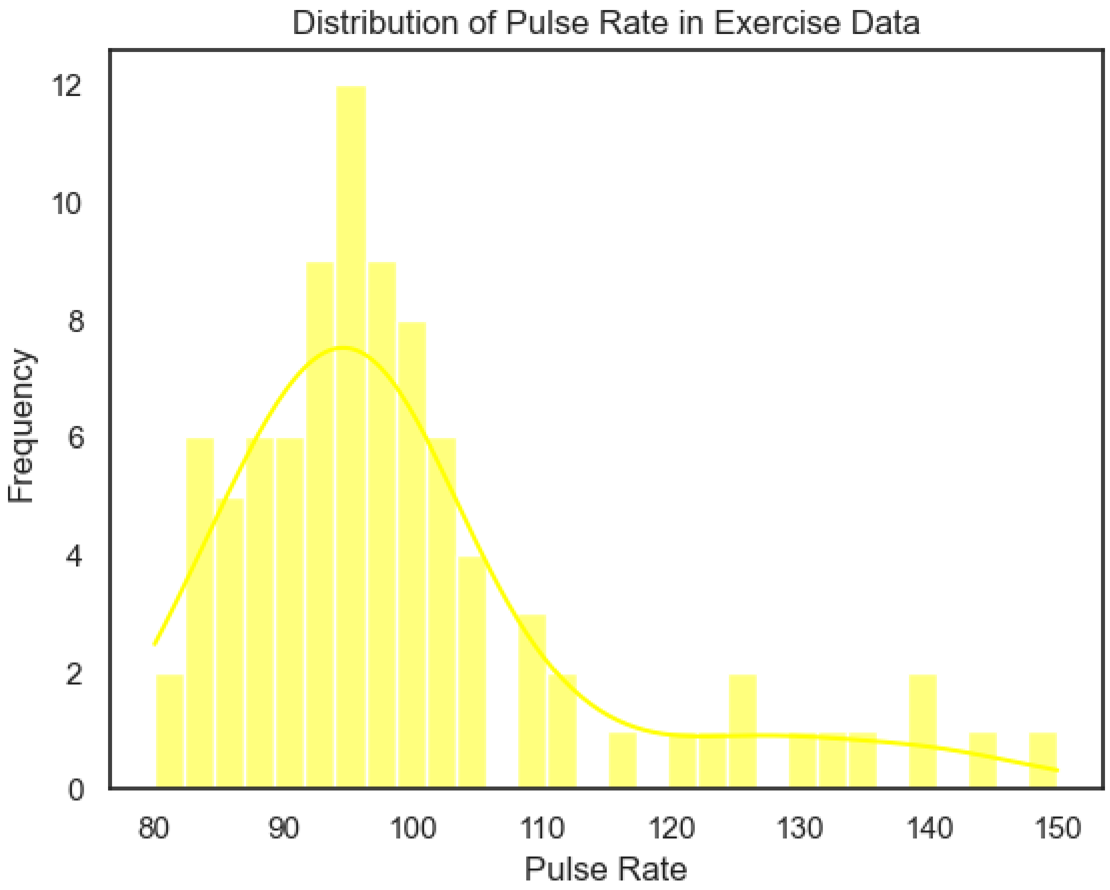
In [7]:

```
import seaborn as sns
import matplotlib.pyplot as plt
# Load the exercise dataset
exercise = sns.load_dataset("exercise")
sns.histplot(data=exercise,x="pulse",bins=30,kde=True,color="green")
## Add Labels and Titles
plt.xlabel("Pulse Rate")
plt.ylabel("Frequency")
plt.title("Distribution of Pulse Rate in Exercise Data")
# Display the Plot
plt.show()
```



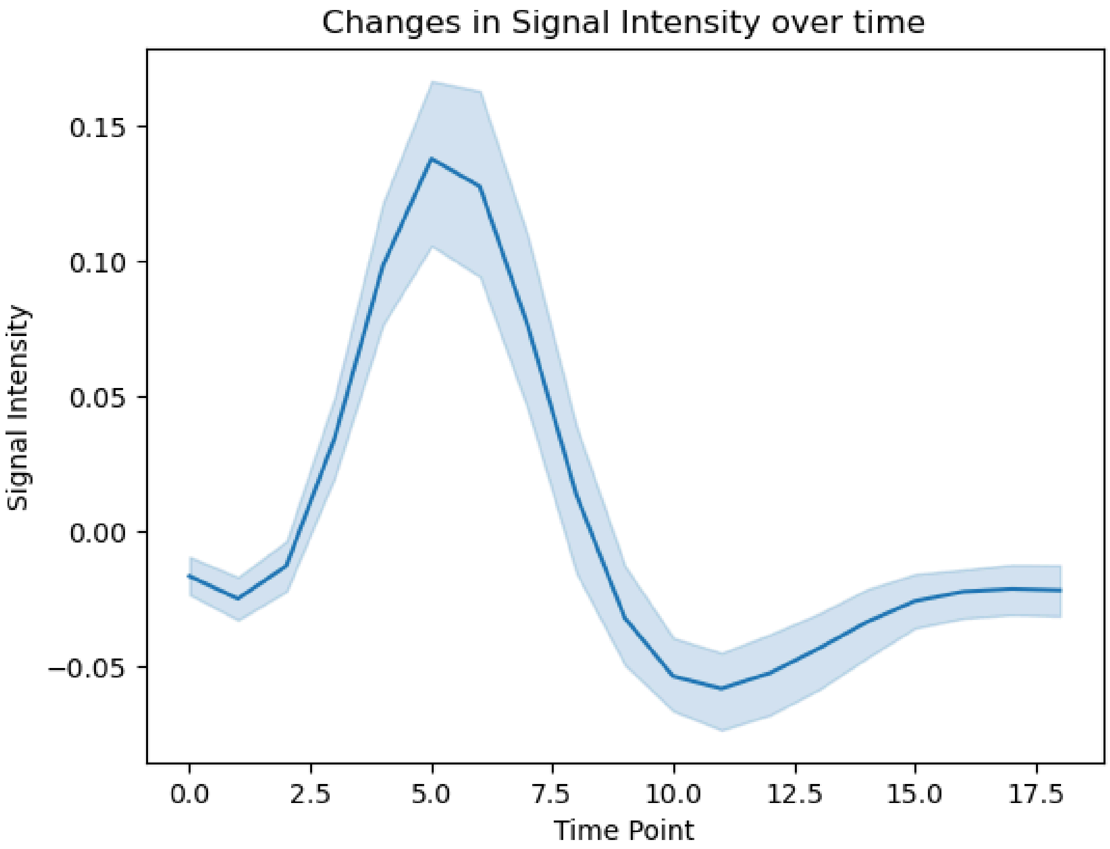
In [8]:

```
import seaborn as sns
import matplotlib.pyplot as plt
# Load the exercise dataset
exercise = sns.load_dataset("exercise")
sns.histplot(data=exercise,x="pulse",bins=30,kde=True,color="yellow")
## Add Labels and Titles
plt.xlabel("Pulse Rate")
plt.ylabel("Frequency")
plt.title("Distribution of Pulse Rate in Exercise Data")
# Display the Plot
plt.show()
```



Line Diagram or Line Plot

In [4]:
```
import seaborn as sns
import matplotlib.pyplot as plt
fmri=sns.load_dataset("fmri")
sns.lineplot(x="timepoint",y="signal",data=fmri)
plt.xlabel("Time Point")
plt.ylabel("Signal Intensity")
plt.title("Changes in Signal Intensity over time")
plt.show()
```



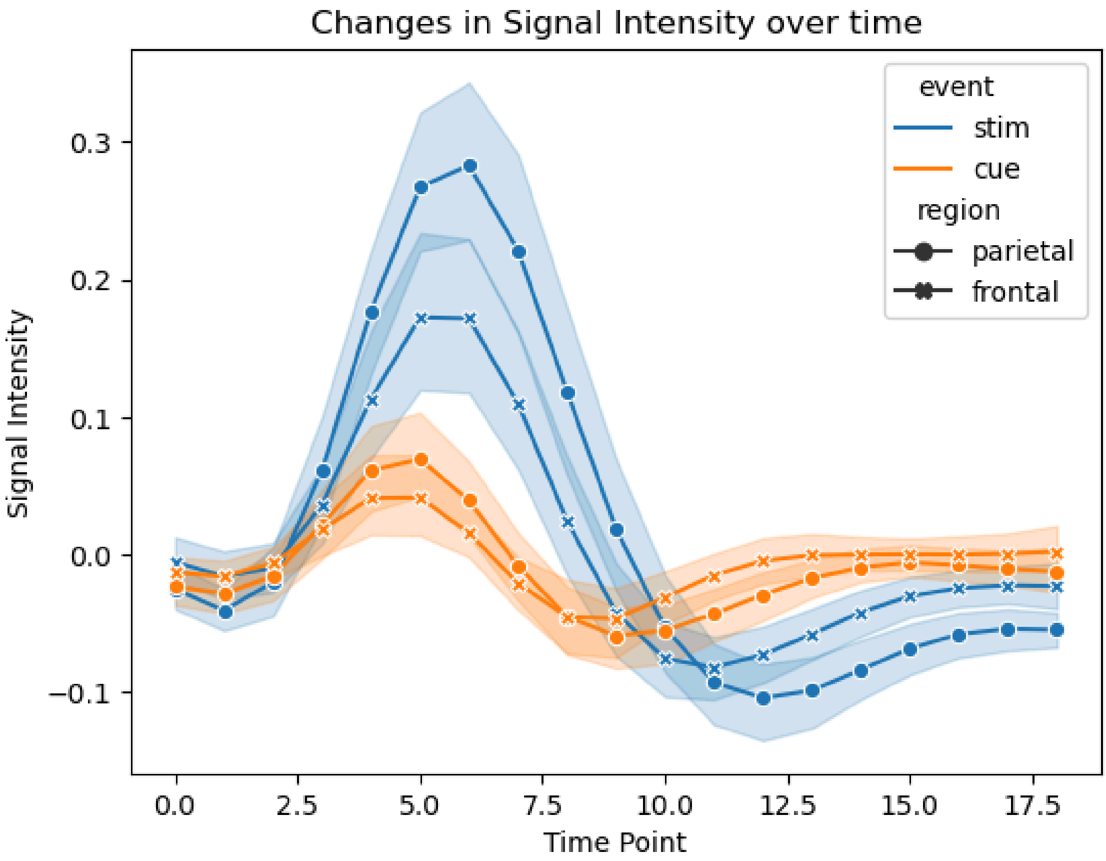
In [1]:

```
import seaborn as sns
import matplotlib.pyplot as plt
fmri=sns.load_dataset("fmri")
fmri
```

Out[1]:

```
      subject  timepoint  event    region     signal
   0      s13         18   stim  parietal  -0.017552
   1       s5         14   stim  parietal  -0.080883
   2      s12         18   stim  parietal  -0.081033
   3      s11         18   stim  parietal  -0.046134
   4      s10         18   stim  parietal  -0.037970
 ...      ...        ...    ...       ...        ...
1059       s0          8    cue   frontal   0.018165
1060      s13          7    cue   frontal  -0.029130
1061      s12          7    cue   frontal  -0.004939
1062      s11          7    cue   frontal  -0.025367
1063       s0          0    cue  parietal  -0.006899

1064 rows × 5 columns
```

In [3]:
```
import seaborn as sns
import matplotlib.pyplot as plt
fmri=sns.load_dataset("fmri")
sns.lineplot(x="timepoint",y="signal",hue="event",style="region",markers=T
plt.xlabel("Time Point")
plt.ylabel("Signal Intensity")
plt.title("Changes in Signal Intensity over time")
plt.show()
```



Bar Diagram or Bar Plot

Qualitative color palettes

tab10/default,deep,muted,pastel,bright,dark,colorblind,tab20,tab20b,tab20c

Sequential Color Brewer palettes

Single-hue and multi-hue (up to three) options, which are: Greys, Reds, Greens, Blues, Oranges, Purples, BuGn, BuPu, GnBu, OrRd, PuBu, PuRd, RdPu, YlGn, PuBuGn, YlGnBu, YlOrBr and YlOrRd.

Perceptually uniform palettes

This category includes the original Seaborn palettes rocket, mako, flare, and crest, as well as the matplotlib palettes viridis, plasma, inferno, magma and cividis.

Qualitative Color Brower palettes

The palettes in this category are: Set1, Set2, Set3, Paired, Accent, Pastel1, Pastel2 and Dark2.

Diverging Color Brewer palettes

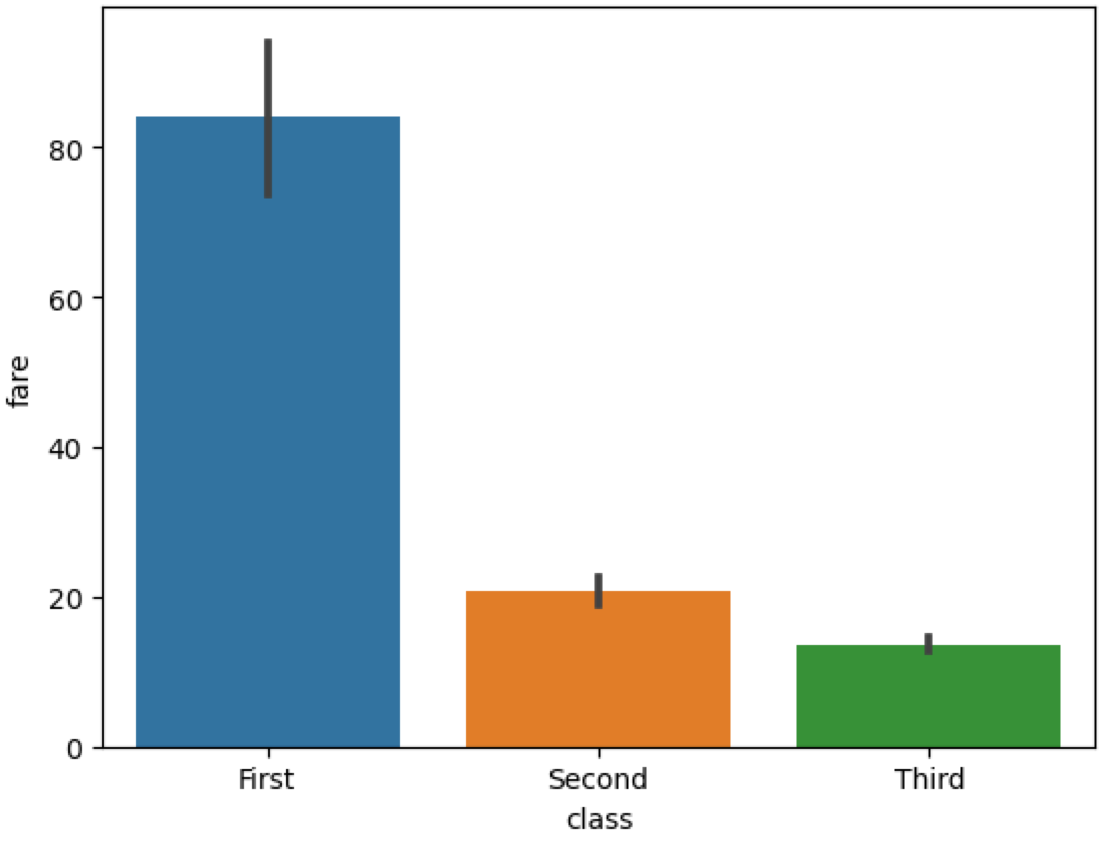
These include RdBu, RdGy, PRGn, PiYG, BrBG, RdYlBu, RdYlGn and Spectral, and their reversed (_r) variations.

Creating a Customized Color Palette

"#F72585", "#7209B7", "#3A0CA3", "#4361EE", "#4CC9F0"

In [5]:
```
import seaborn as sns
titanic = sns.load_dataset("titanic")
sns.barplot(x="class",y="fare",data=titanic)
```

Out[5]:
```
<Axes: xlabel='class', ylabel='fare'>
```

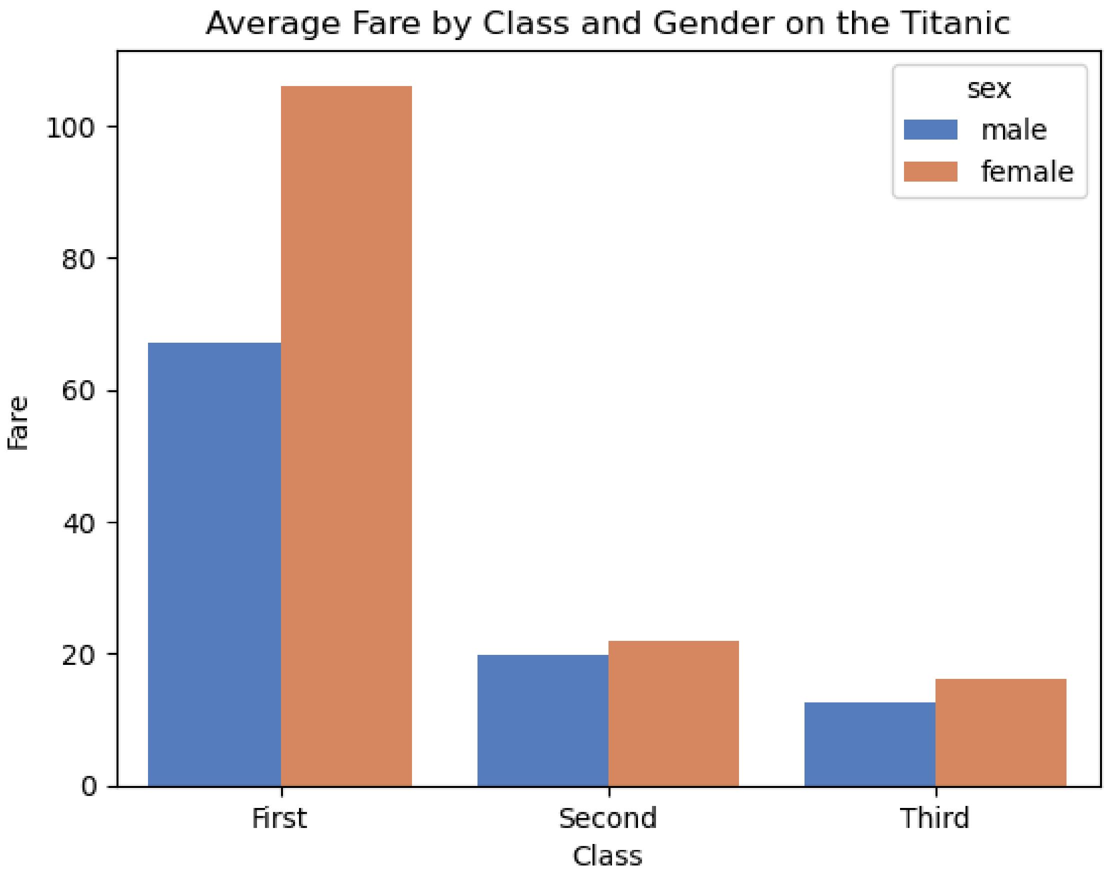


Let’s customize this plot by including `sex` column from the dataset.

In [7]:

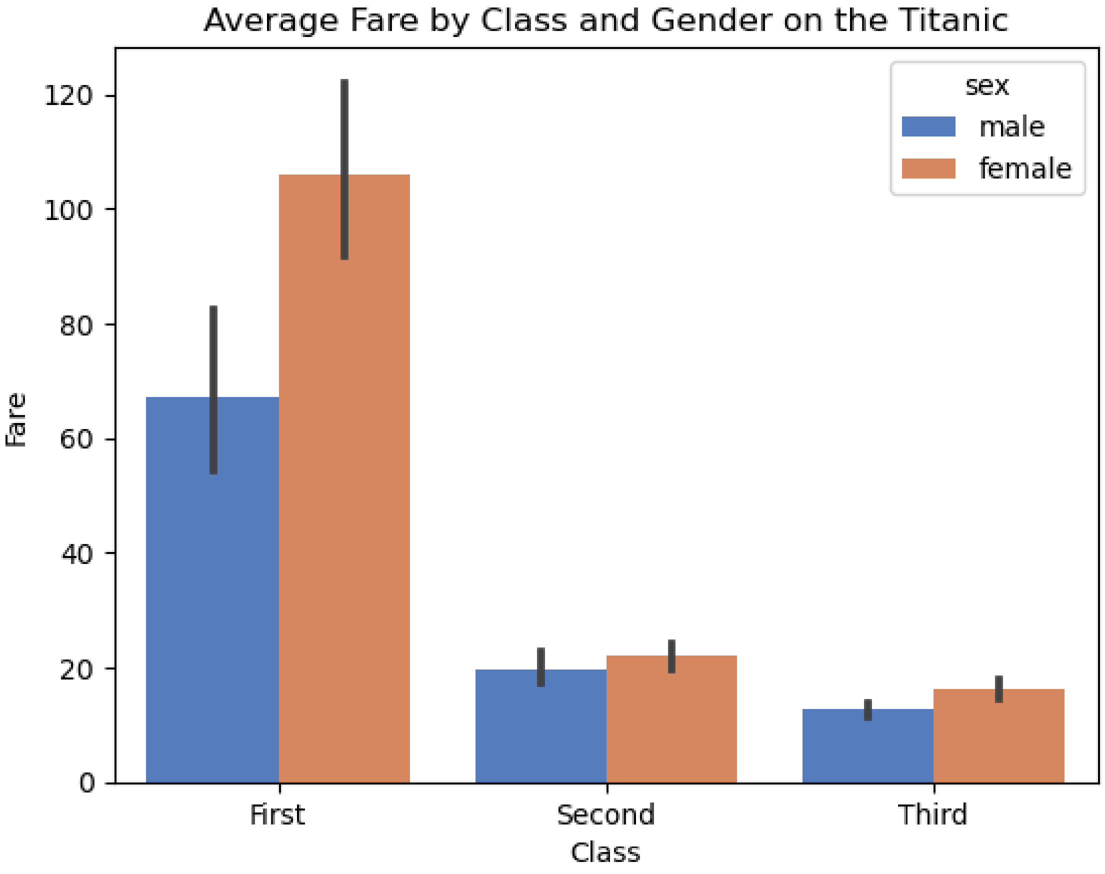
```
import matplotlib.pyplot as plt
import seaborn as sns
titanic=sns.load_dataset("titanic")
sns.barplot(x="class",y="fare",hue="sex",ci=None,palette="muted",data=tita
plt.xlabel("Class")
plt.ylabel("Fare")
plt.title("Average Fare by Class and Gender on the Titanic")
plt.show()
```

```
C:\Users\SAGNIK SAMANTA\AppData\Local\Temp\ipykernel_17680\3479664489.py:
4: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effec
t.

  sns.barplot(x="class",y="fare",hue="sex",ci=None,palette="muted",data=t
itanic)
```



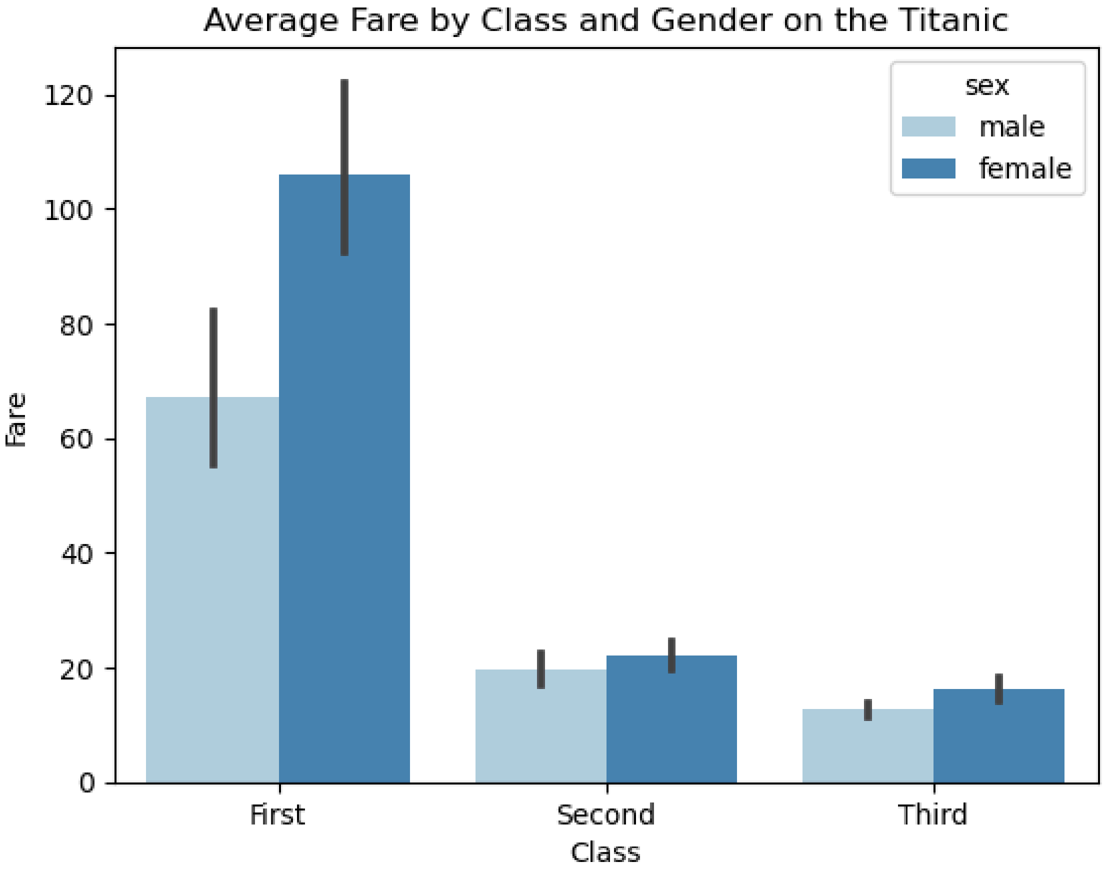
In [9]:

```python
import matplotlib.pyplot as plt
import seaborn as sns
titanic=sns.load_dataset("titanic")
sns.barplot(x="class",y="fare",hue="sex",palette="muted",data=titanic)
plt.xlabel("Class")
plt.ylabel("Fare")
plt.title("Average Fare by Class and Gender on the Titanic")
plt.show()
```



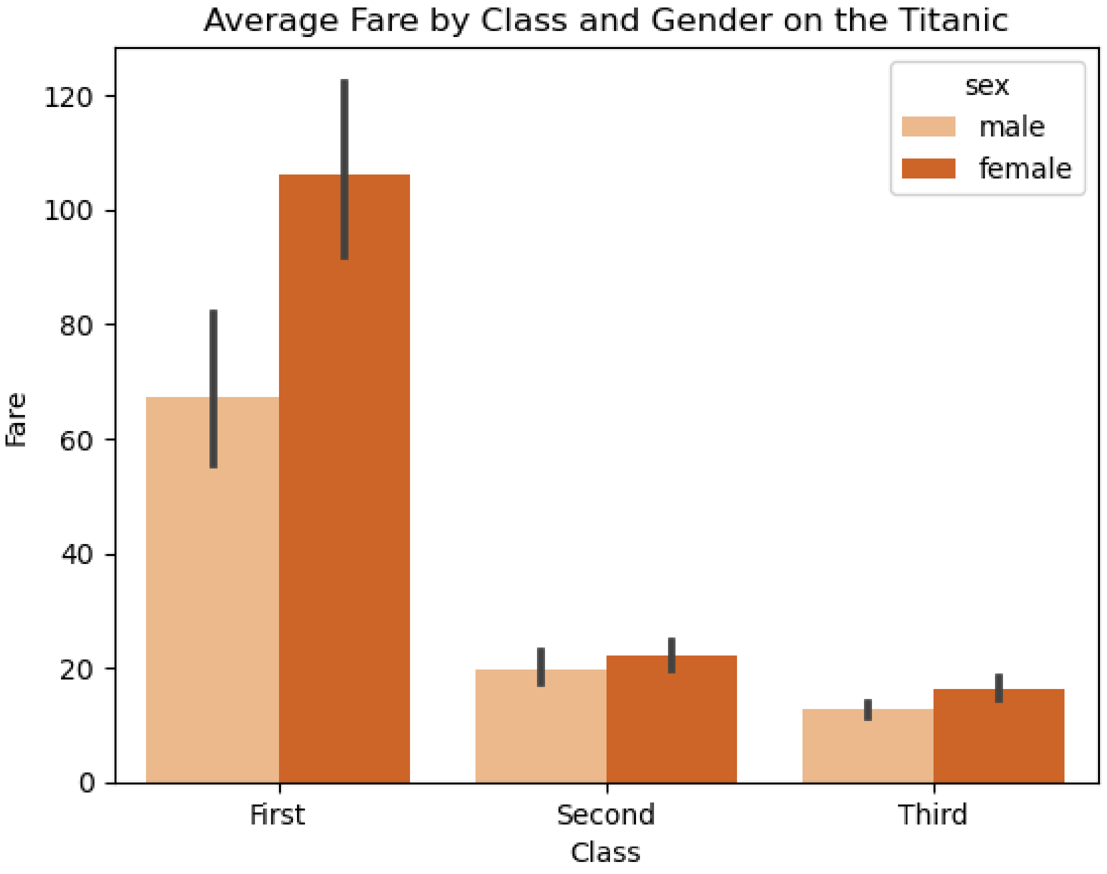
In [16]:

```
import matplotlib.pyplot as plt
import seaborn as sns
titanic=sns.load_dataset("titanic")
sns.barplot(x="class",y="fare",hue="sex",palette="Blues",data=titanic)
plt.xlabel("Class")
plt.ylabel("Fare")
plt.title("Average Fare by Class and Gender on the Titanic")
plt.show()
```

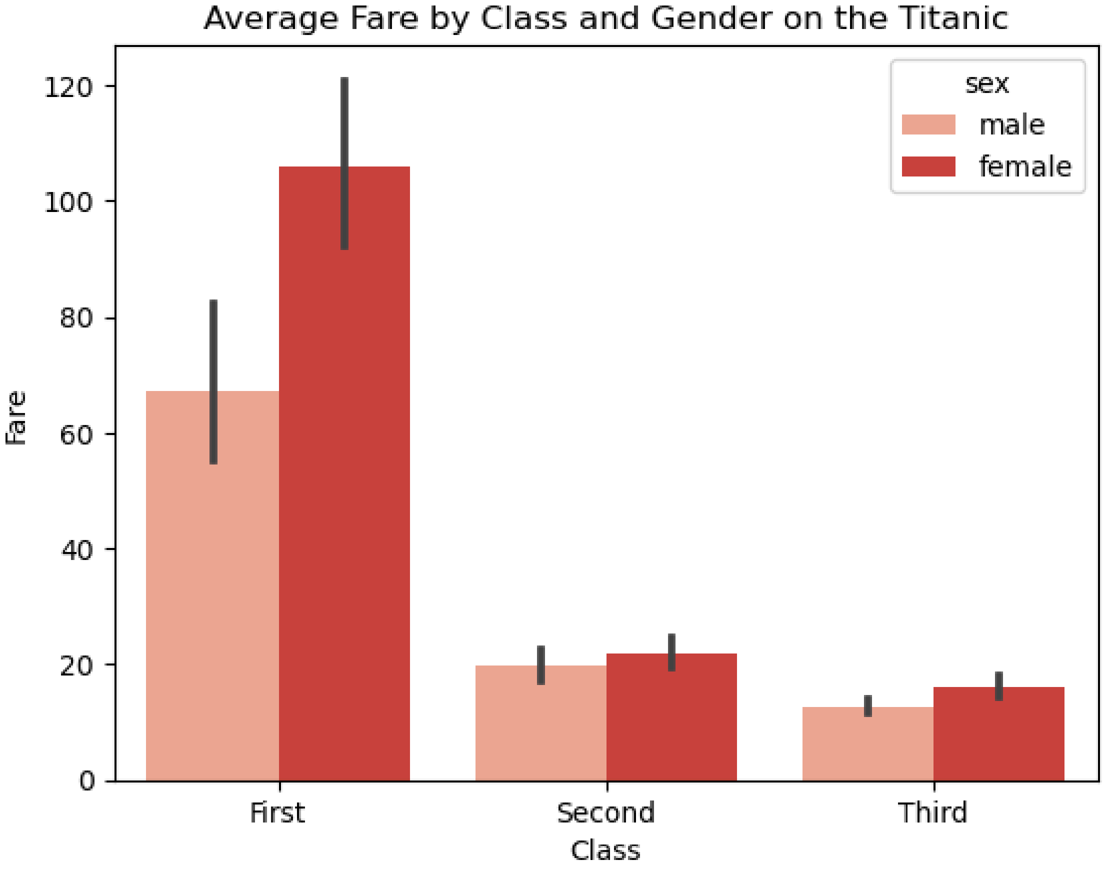


In [17]:

```
import matplotlib.pyplot as plt
import seaborn as sns
titanic=sns.load_dataset("titanic")
sns.barplot(x="class",y="fare",hue="sex",palette="Oranges",data=titanic)
plt.xlabel("Class")
plt.ylabel("Fare")
plt.title("Average Fare by Class and Gender on the Titanic")
plt.show()
```

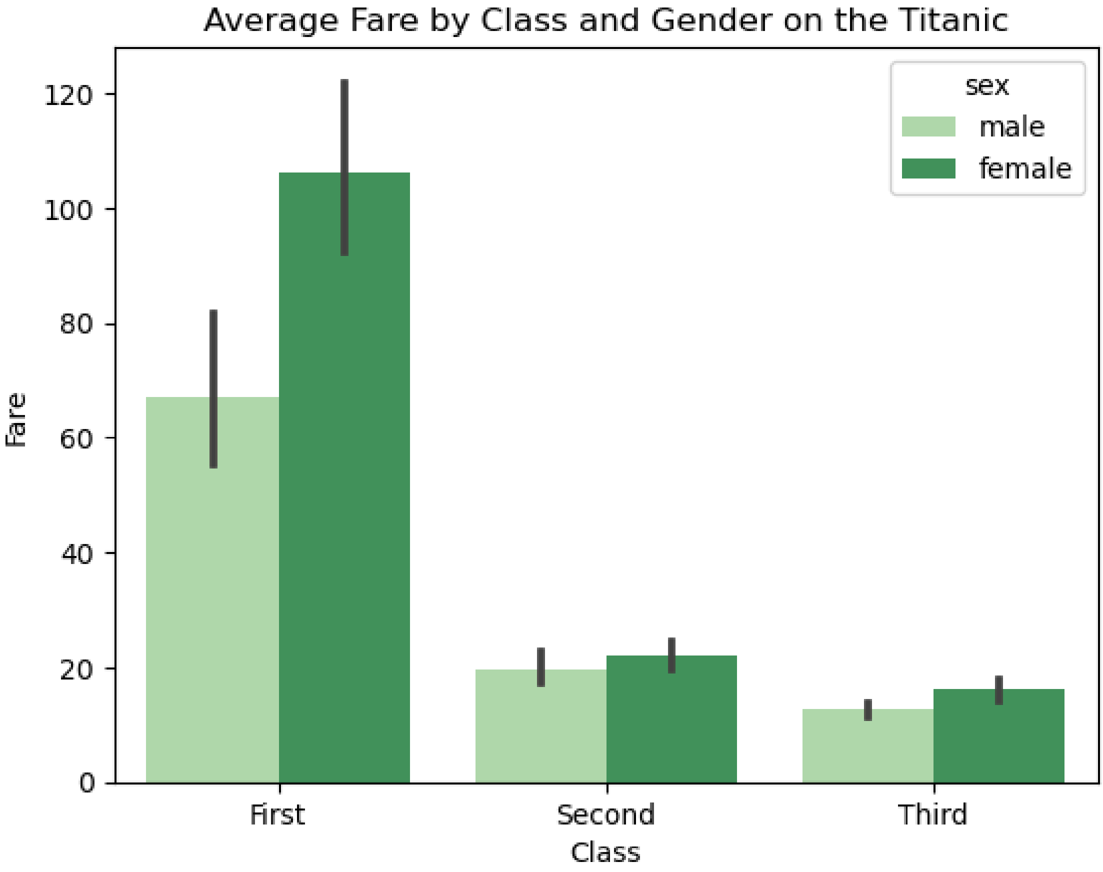


In [18]:
```python
import matplotlib.pyplot as plt
import seaborn as sns
titanic=sns.load_dataset("titanic")
sns.barplot(x="class",y="fare",hue="sex",palette="Reds",data=titanic)
plt.xlabel("Class")
plt.ylabel("Fare")
plt.title("Average Fare by Class and Gender on the Titanic")
plt.show()
```



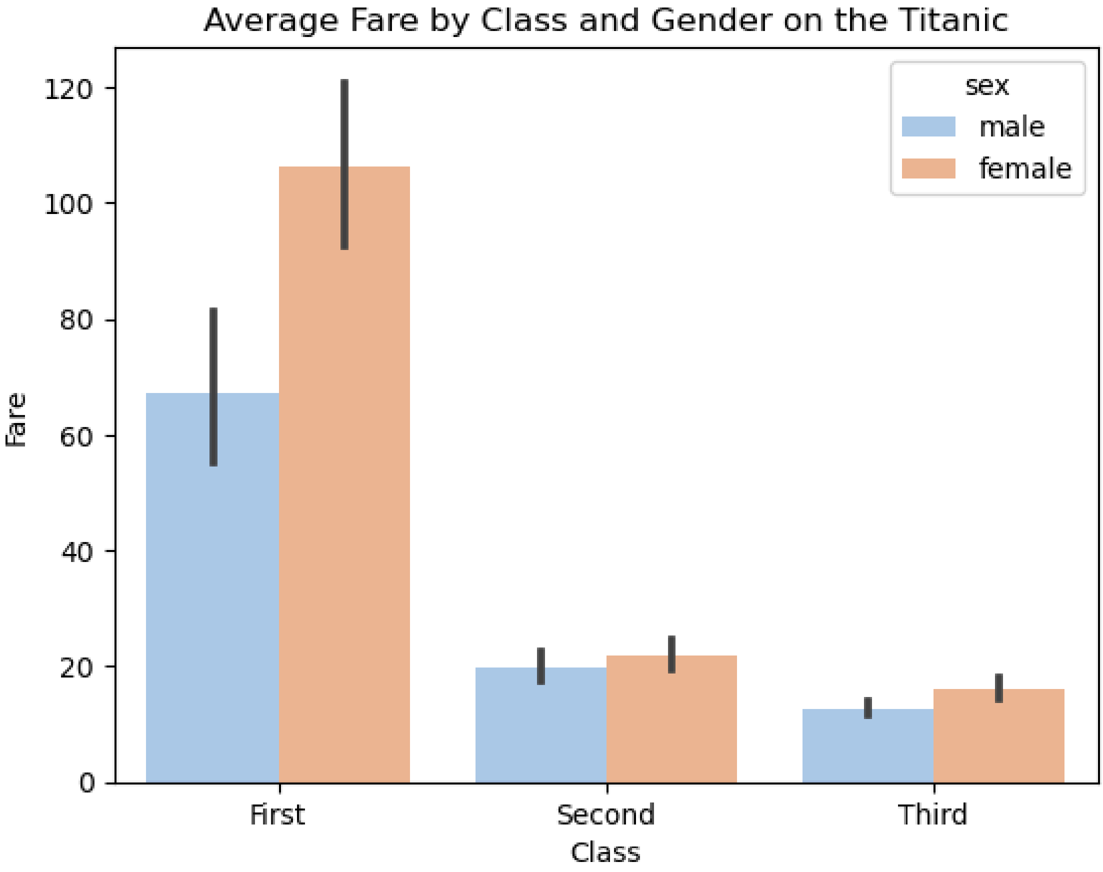
In [20]:

```
import matplotlib.pyplot as plt
import seaborn as sns
titanic=sns.load_dataset("titanic")
sns.barplot(x="class",y="fare",hue="sex",palette="Greens",data=titanic)
plt.xlabel("Class")
plt.ylabel("Fare")
plt.title("Average Fare by Class and Gender on the Titanic")
plt.show()
```

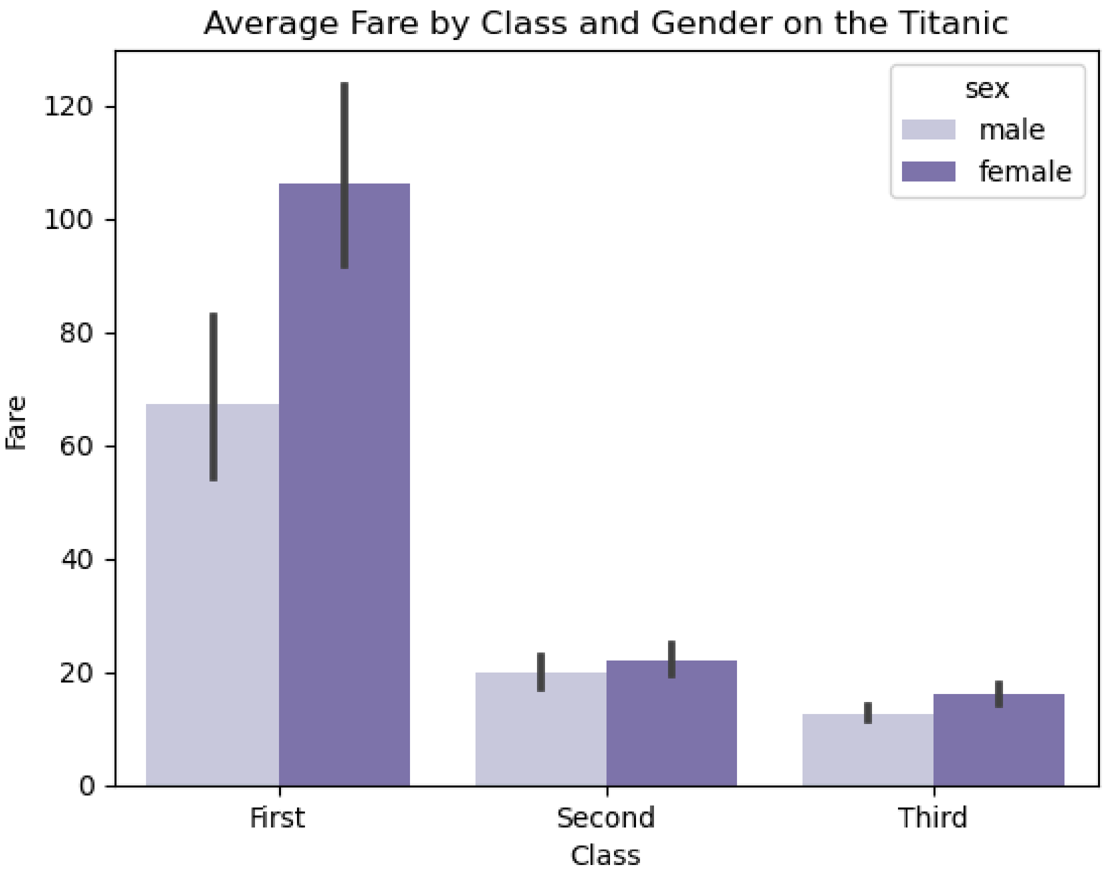


In [30]:

```
import matplotlib.pyplot as plt
import seaborn as sns
titanic=sns.load_dataset("titanic")
sns.barplot(x="class",y="fare",hue="sex",palette="pastel",data=titanic)
plt.xlabel("Class")
plt.ylabel("Fare")
plt.title("Average Fare by Class and Gender on the Titanic")
plt.show()
```

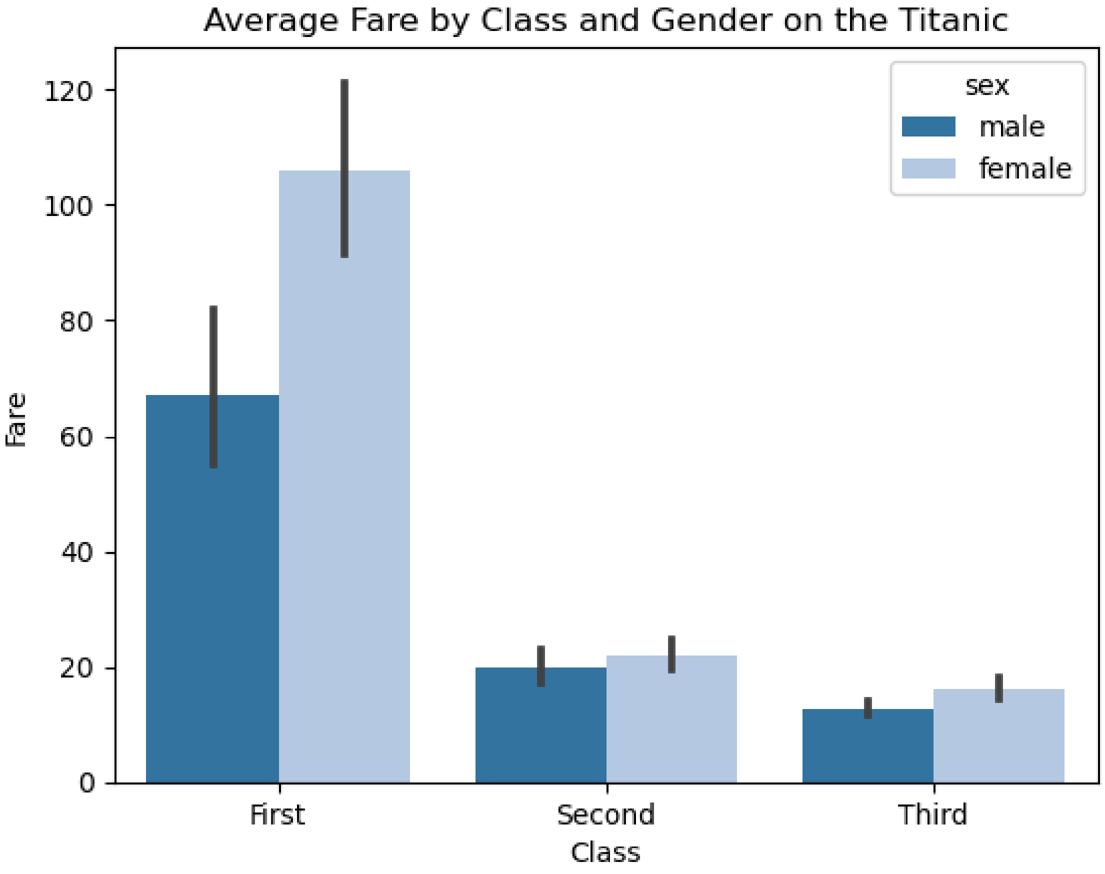


In [31]:
```
import matplotlib.pyplot as plt
import seaborn as sns
titanic=sns.load_dataset("titanic")
sns.barplot(x="class",y="fare",hue="sex",palette="Purples",data=titanic)
plt.xlabel("Class")
plt.ylabel("Fare")
plt.title("Average Fare by Class and Gender on the Titanic")
plt.show()
```



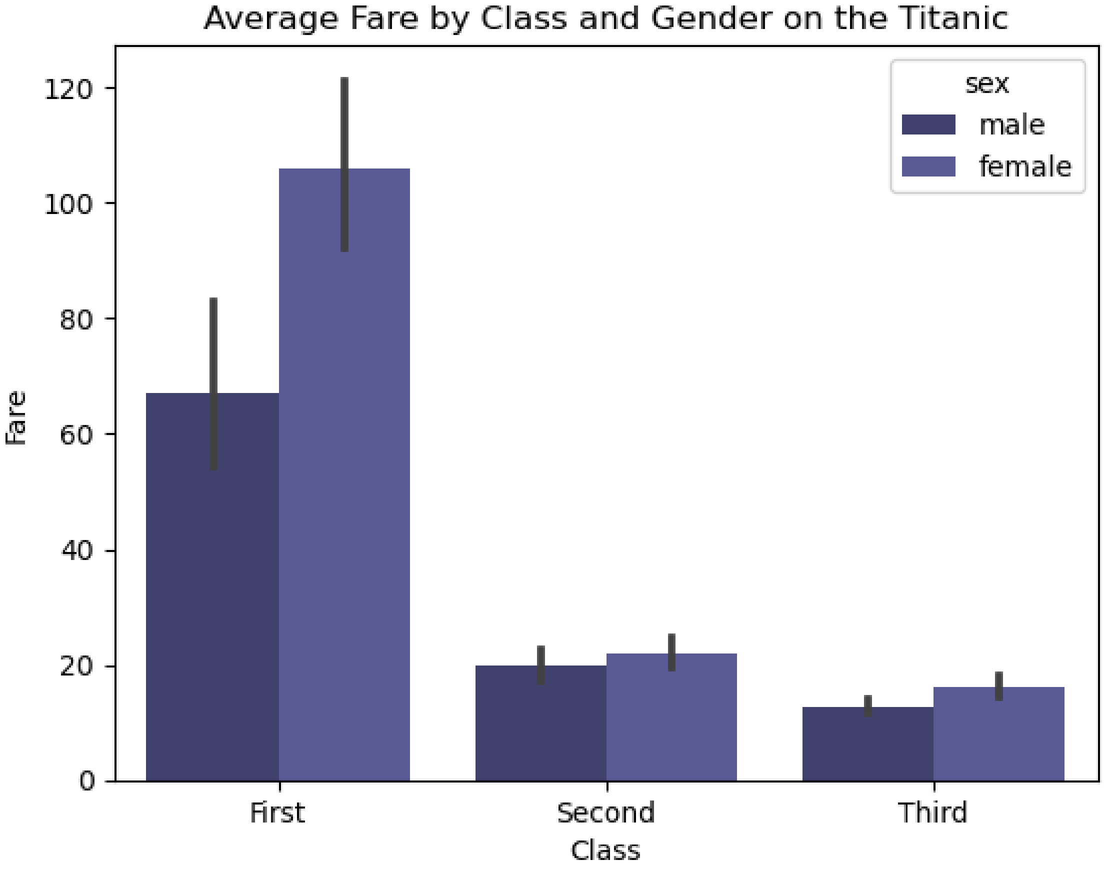
In [32]:

```python
import matplotlib.pyplot as plt
import seaborn as sns
titanic=sns.load_dataset("titanic")
sns.barplot(x="class",y="fare",hue="sex",palette="tab20",data=titanic)
plt.xlabel("Class")
plt.ylabel("Fare")
plt.title("Average Fare by Class and Gender on the Titanic")
plt.show()
```



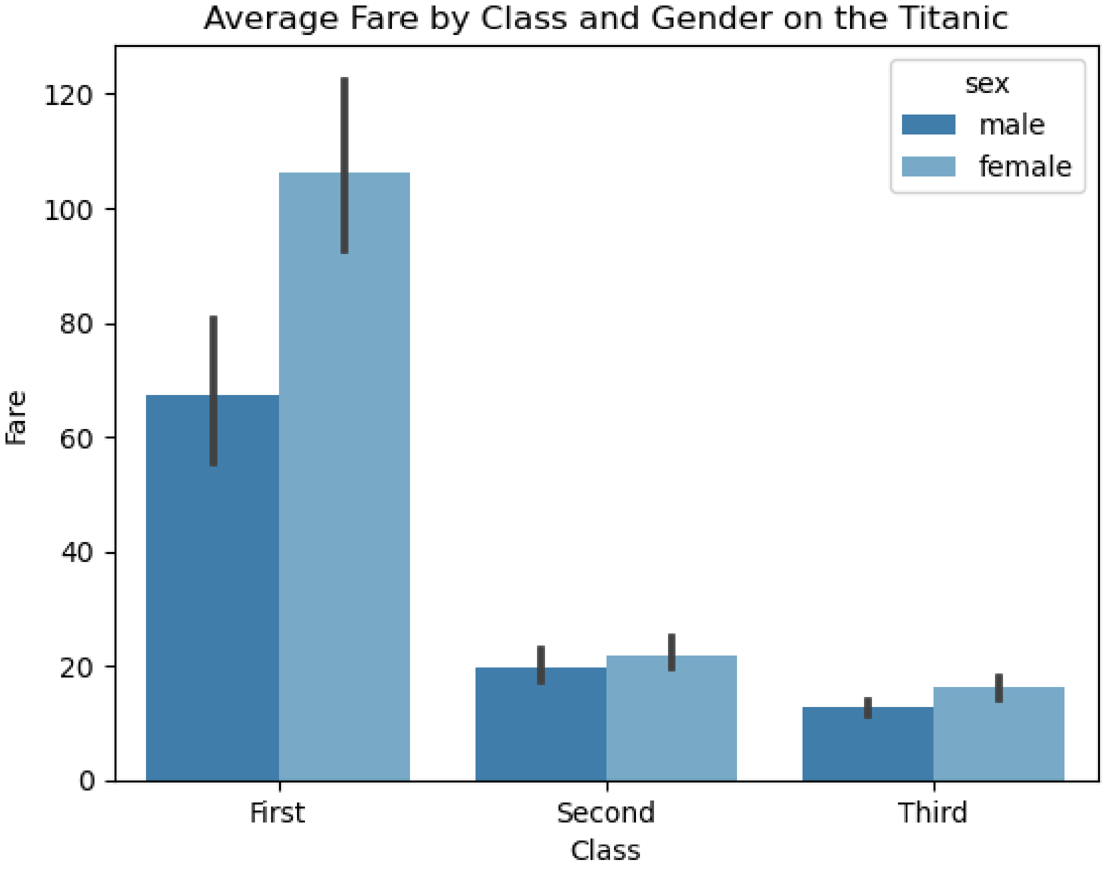
In [33]:

```
import matplotlib.pyplot as plt
import seaborn as sns
titanic=sns.load_dataset("titanic")
sns.barplot(x="class",y="fare",hue="sex",palette="tab20b",data=titanic)
plt.xlabel("Class")
plt.ylabel("Fare")
plt.title("Average Fare by Class and Gender on the Titanic")
plt.show()
```



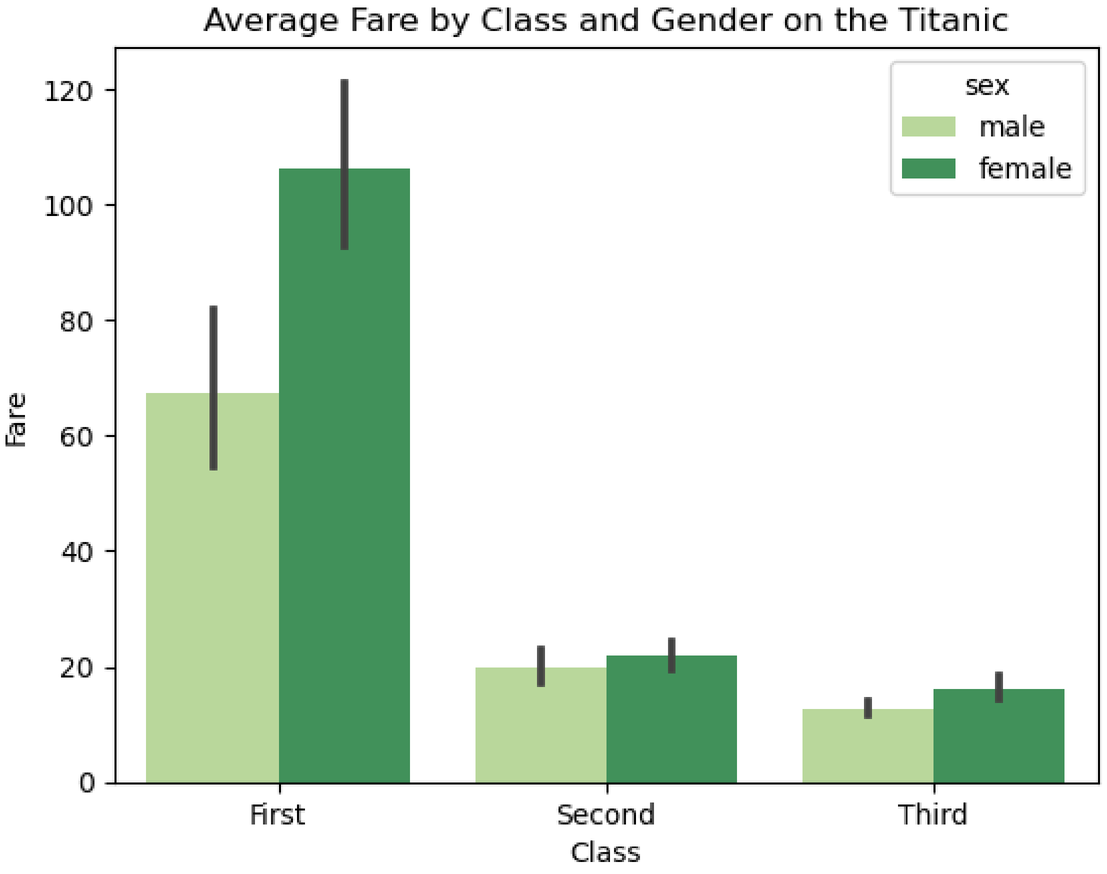
In [34]:

```
import matplotlib.pyplot as plt
import seaborn as sns
titanic=sns.load_dataset("titanic")
sns.barplot(x="class",y="fare",hue="sex",palette="tab20c",data=titanic)
plt.xlabel("Class")
plt.ylabel("Fare")
plt.title("Average Fare by Class and Gender on the Titanic")
plt.show()
```

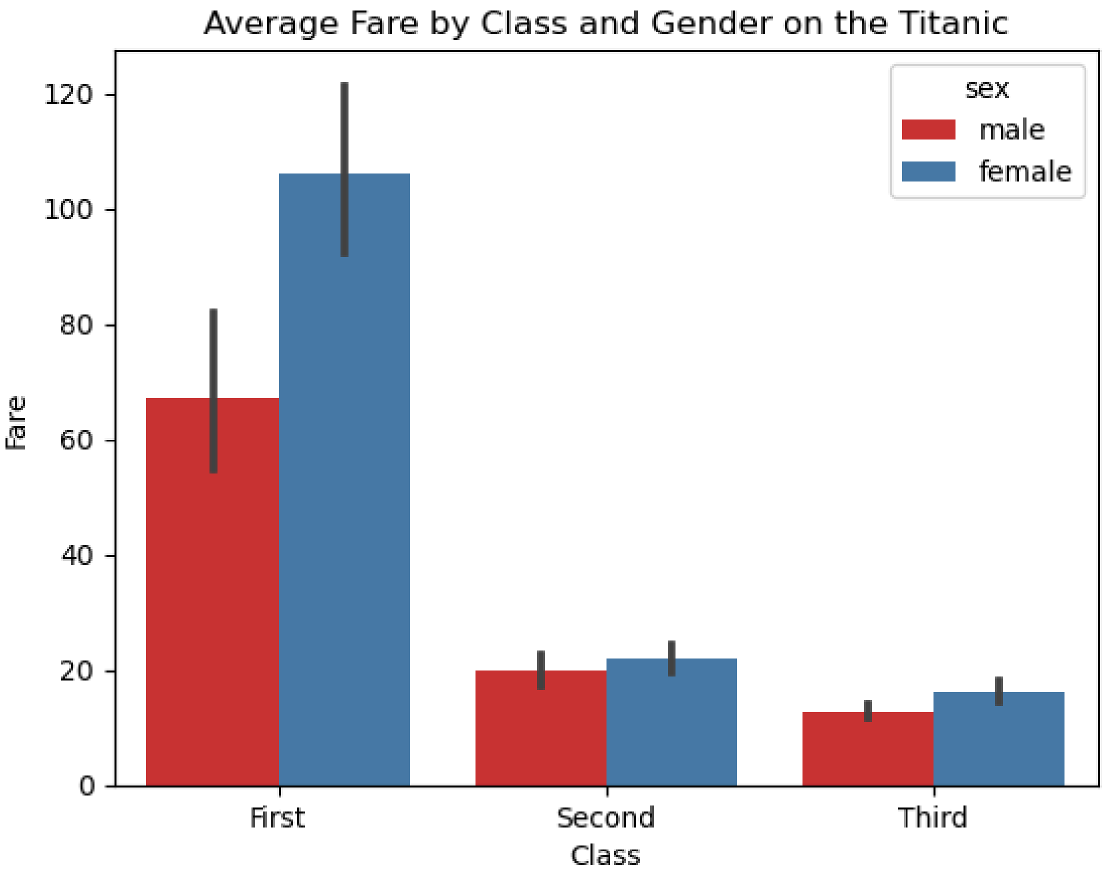


In [35]:

```
import matplotlib.pyplot as plt
import seaborn as sns
titanic=sns.load_dataset("titanic")
sns.barplot(x="class",y="fare",hue="sex",palette="YlGn",data=titanic)
plt.xlabel("Class")
plt.ylabel("Fare")
plt.title("Average Fare by Class and Gender on the Titanic")
plt.show()
```

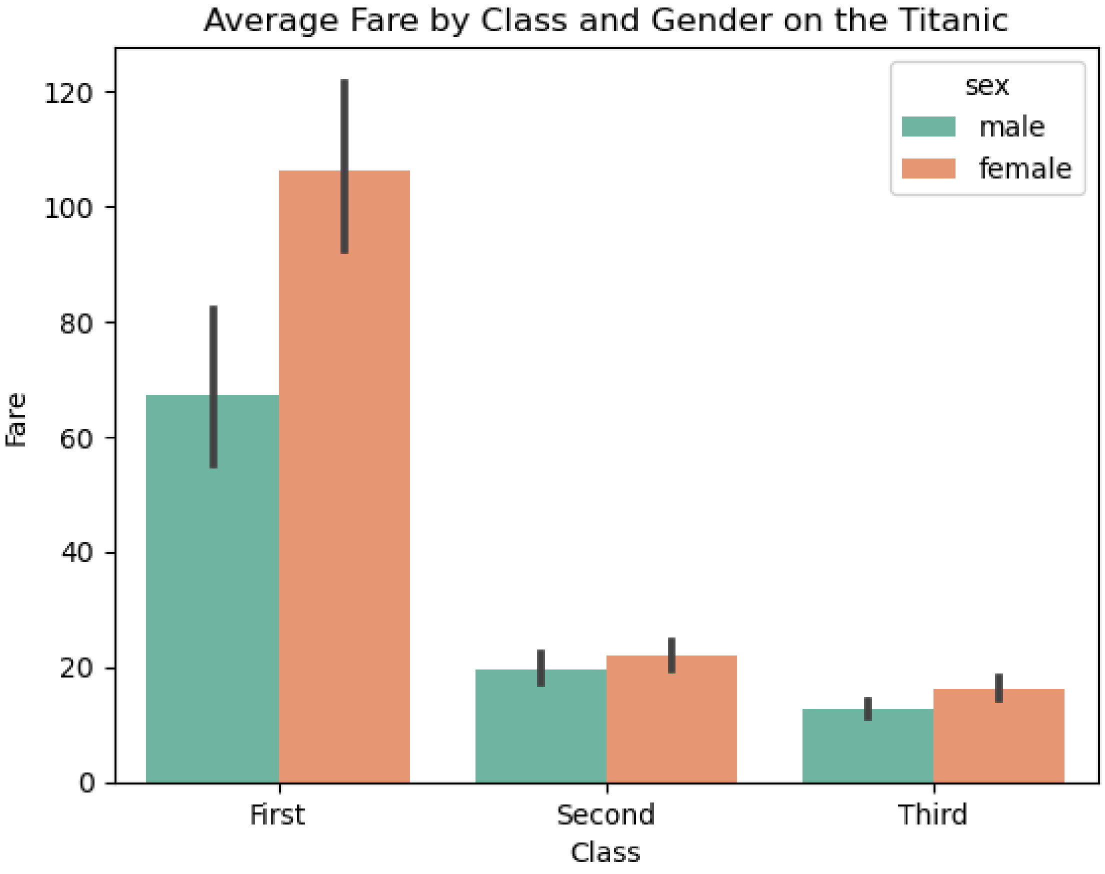


```
In [37]: import matplotlib.pyplot as plt
         import seaborn as sns
         titanic=sns.load_dataset("titanic")
         sns.barplot(x="class",y="fare",hue="sex",palette="Set1",data=titanic)
         plt.xlabel("Class")
         plt.ylabel("Fare")
         plt.title("Average Fare by Class and Gender on the Titanic")
         plt.show()
```

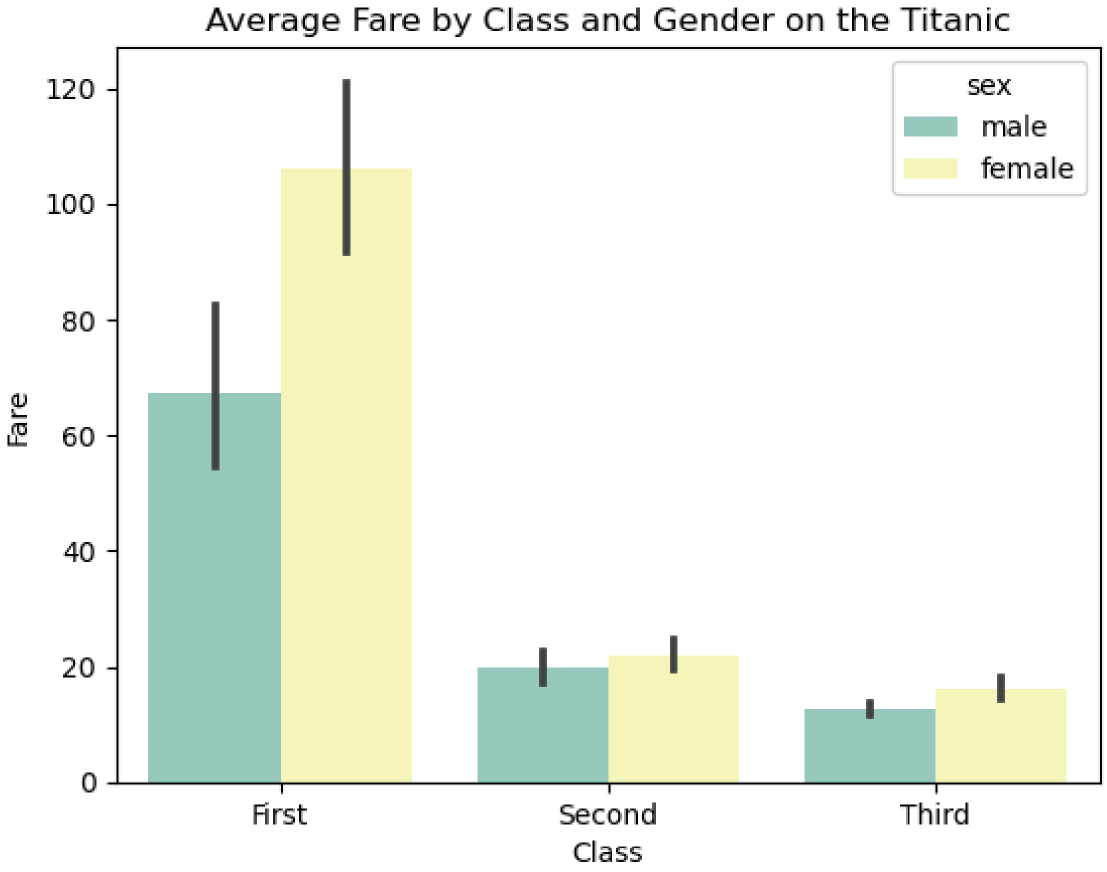


In [38]:

```
import matplotlib.pyplot as plt
import seaborn as sns
titanic=sns.load_dataset("titanic")
sns.barplot(x="class",y="fare",hue="sex",palette="Set2",data=titanic)
plt.xlabel("Class")
plt.ylabel("Fare")
plt.title("Average Fare by Class and Gender on the Titanic")
plt.show()
```

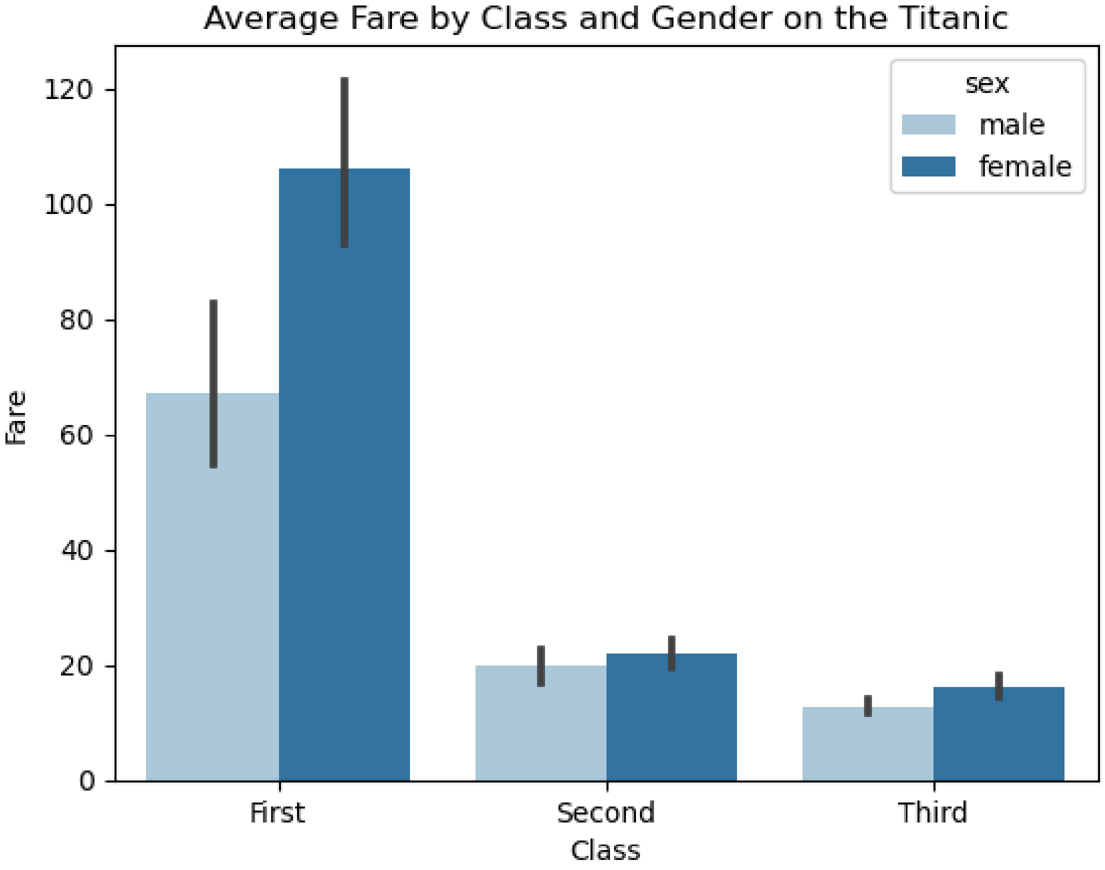


In [39]:
```
import matplotlib.pyplot as plt
import seaborn as sns
titanic=sns.load_dataset("titanic")
sns.barplot(x="class",y="fare",hue="sex",palette="Set3",data=titanic)
plt.xlabel("Class")
plt.ylabel("Fare")
plt.title("Average Fare by Class and Gender on the Titanic")
plt.show()
```

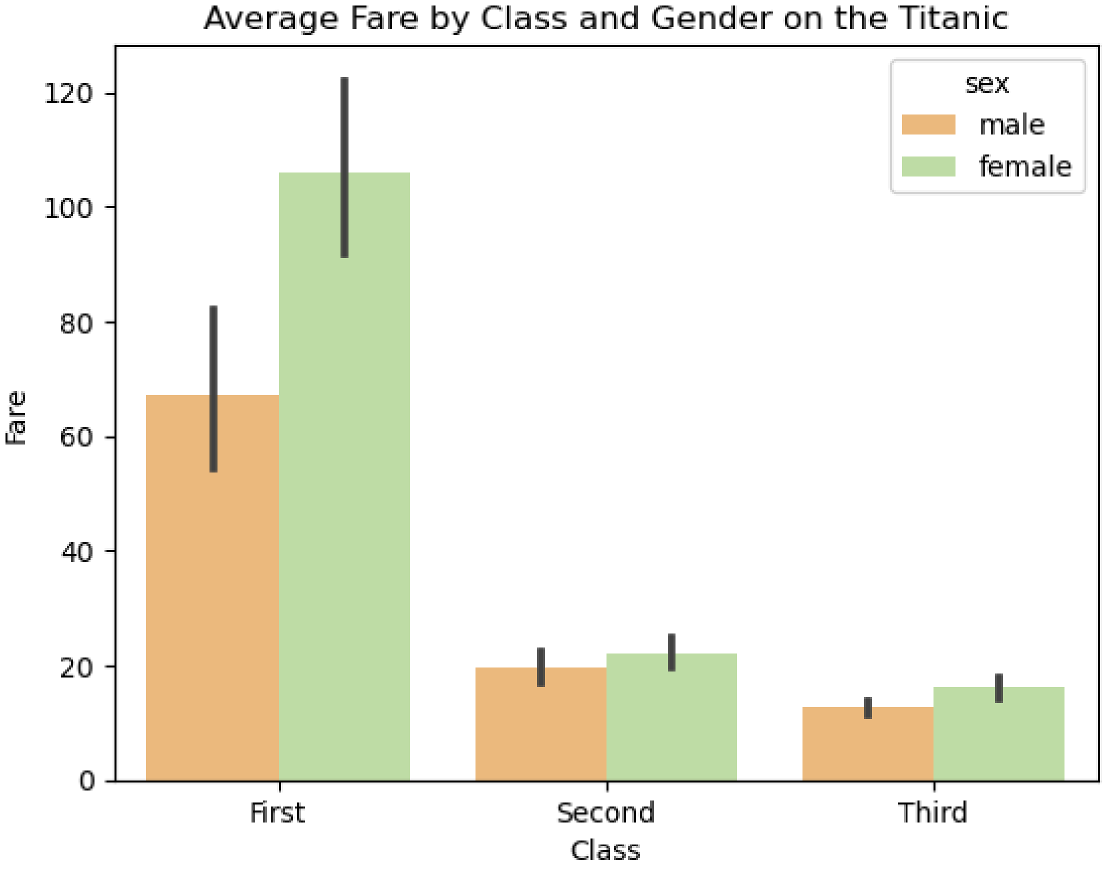


In [62]:

```
import matplotlib.pyplot as plt
import seaborn as sns
titanic=sns.load_dataset("titanic")
sns.barplot(x="class",y="fare",hue="sex",palette="Paired",data=titanic)
plt.xlabel("Class")
plt.ylabel("Fare")
plt.title("Average Fare by Class and Gender on the Titanic")
plt.show()
```

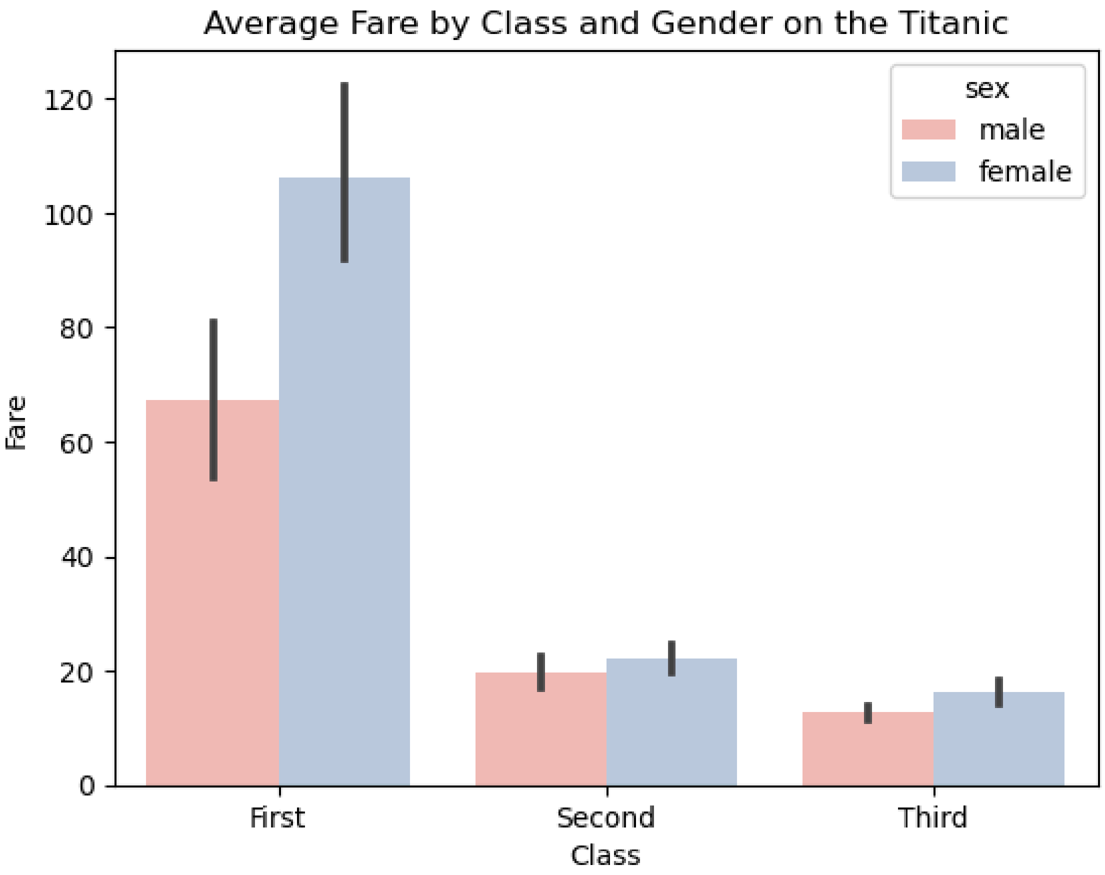


```
In [42]: import matplotlib.pyplot as plt
         import seaborn as sns
         titanic=sns.load_dataset("titanic")
         sns.barplot(x="class",y="fare",hue="sex",palette="Spectral",data=titanic)
         plt.xlabel("Class")
         plt.ylabel("Fare")
         plt.title("Average Fare by Class and Gender on the Titanic")
         plt.show()
```



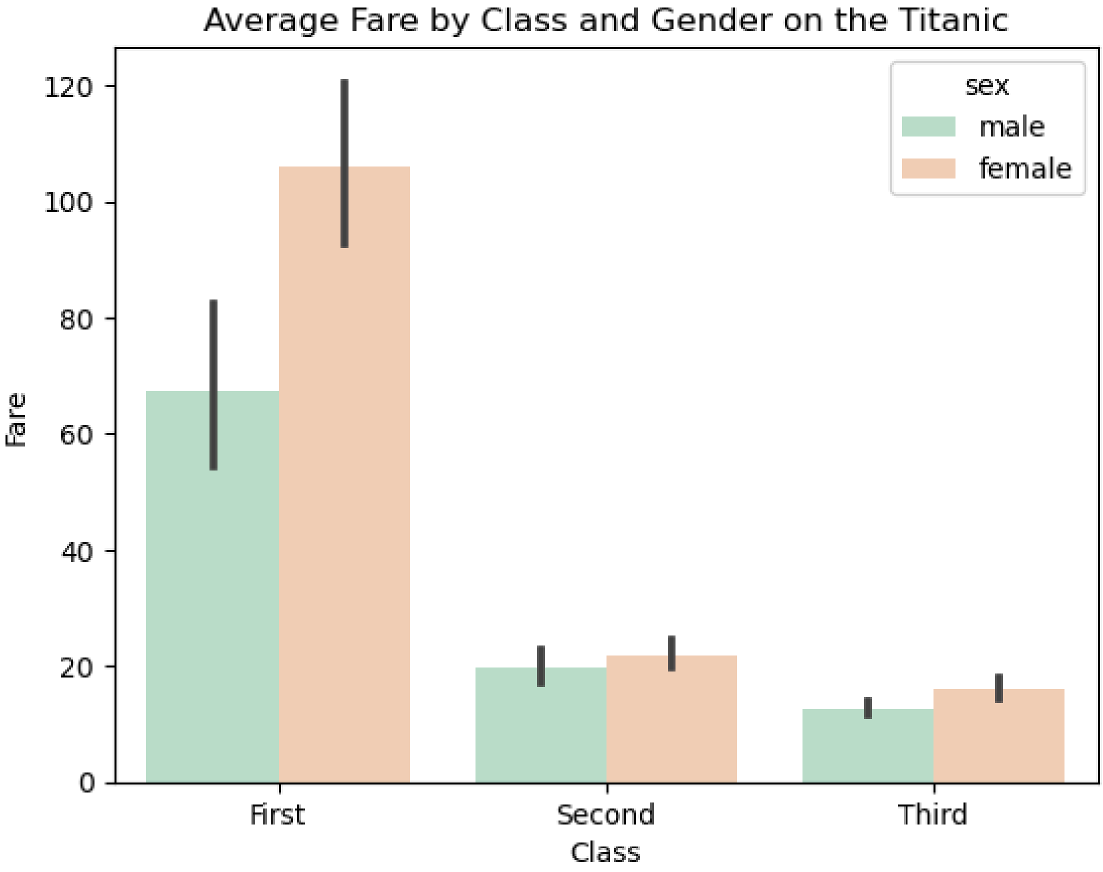
In [44]:

```
import matplotlib.pyplot as plt
import seaborn as sns
titanic=sns.load_dataset("titanic")
sns.barplot(x="class",y="fare",hue="sex",palette="Pastel1",data=titanic)
plt.xlabel("Class")
plt.ylabel("Fare")
plt.title("Average Fare by Class and Gender on the Titanic")
plt.show()
```



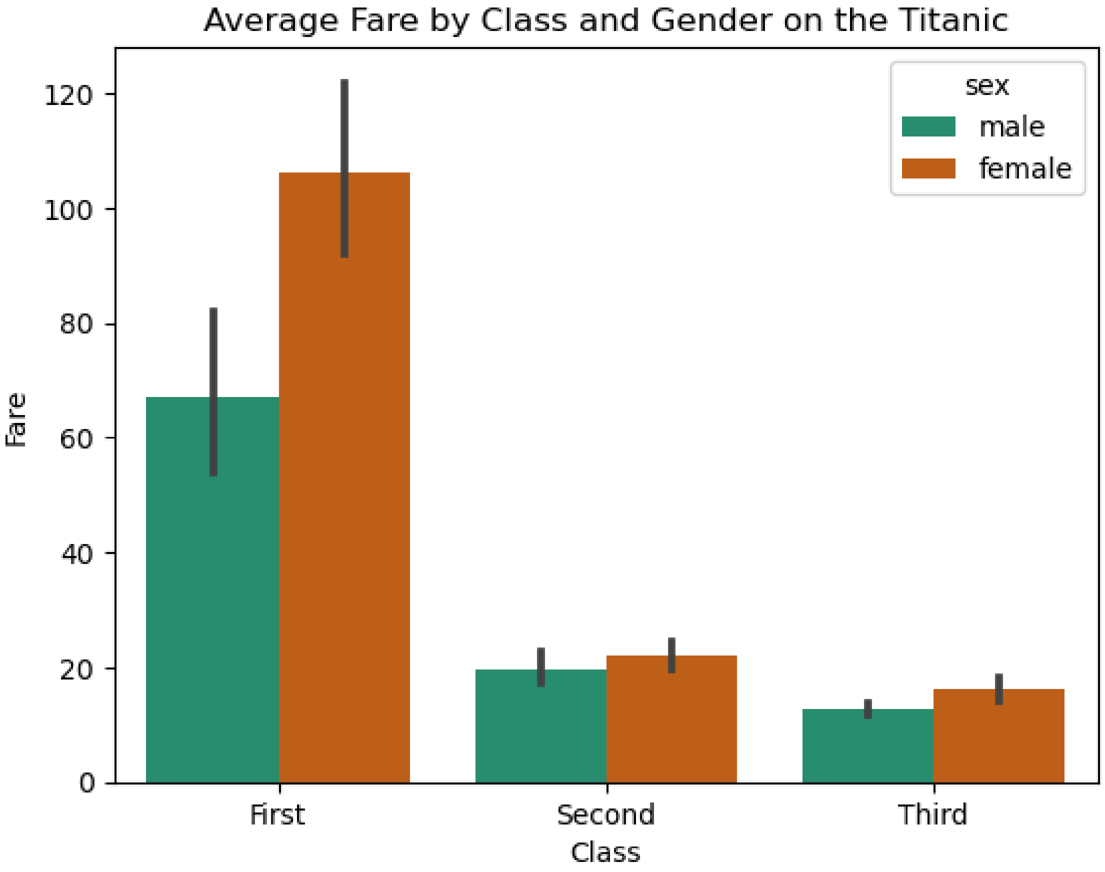
In [45]:

```
import matplotlib.pyplot as plt
import seaborn as sns
titanic=sns.load_dataset("titanic")
sns.barplot(x="class",y="fare",hue="sex",palette="Pastel2",data=titanic)
plt.xlabel("Class")
plt.ylabel("Fare")
plt.title("Average Fare by Class and Gender on the Titanic")
plt.show()
```



In [47]:

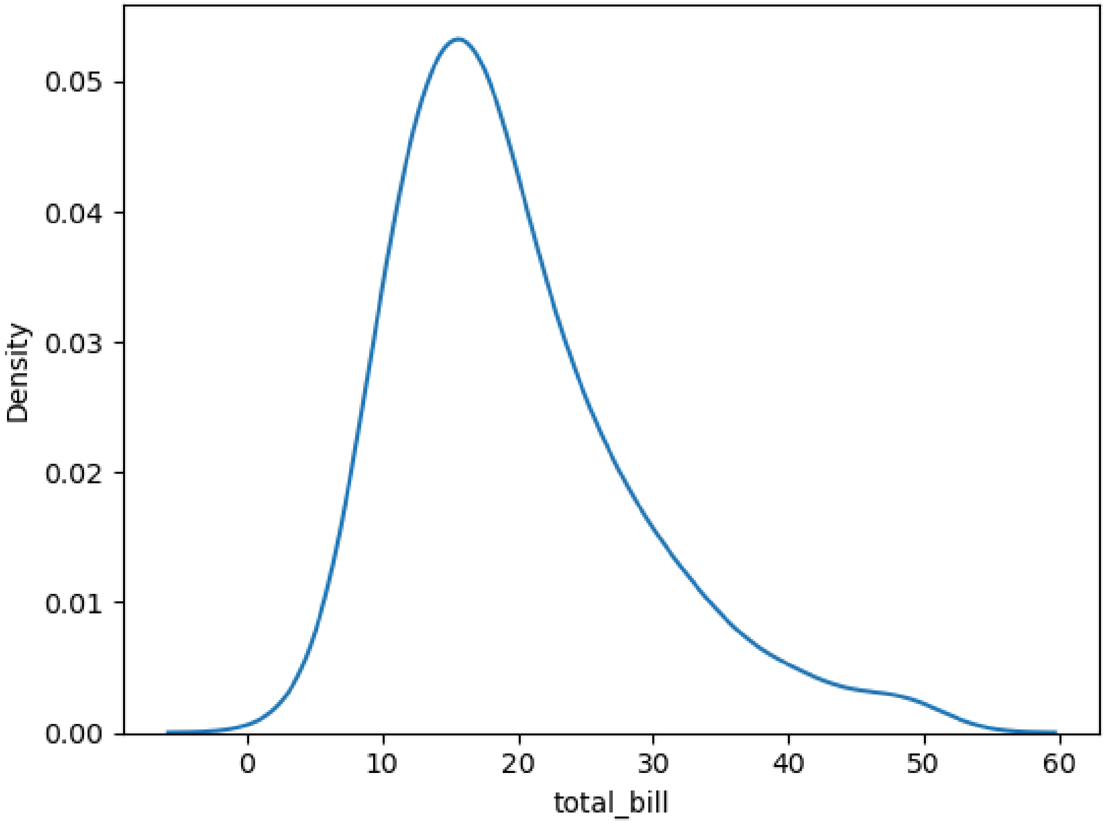
```
import matplotlib.pyplot as plt
import seaborn as sns
titanic=sns.load_dataset("titanic")
sns.barplot(x="class",y="fare",hue="sex",palette="Dark2",data=titanic)
plt.xlabel("Class")
plt.ylabel("Fare")
plt.title("Average Fare by Class and Gender on the Titanic")
plt.show()
```



Seaborn density plots

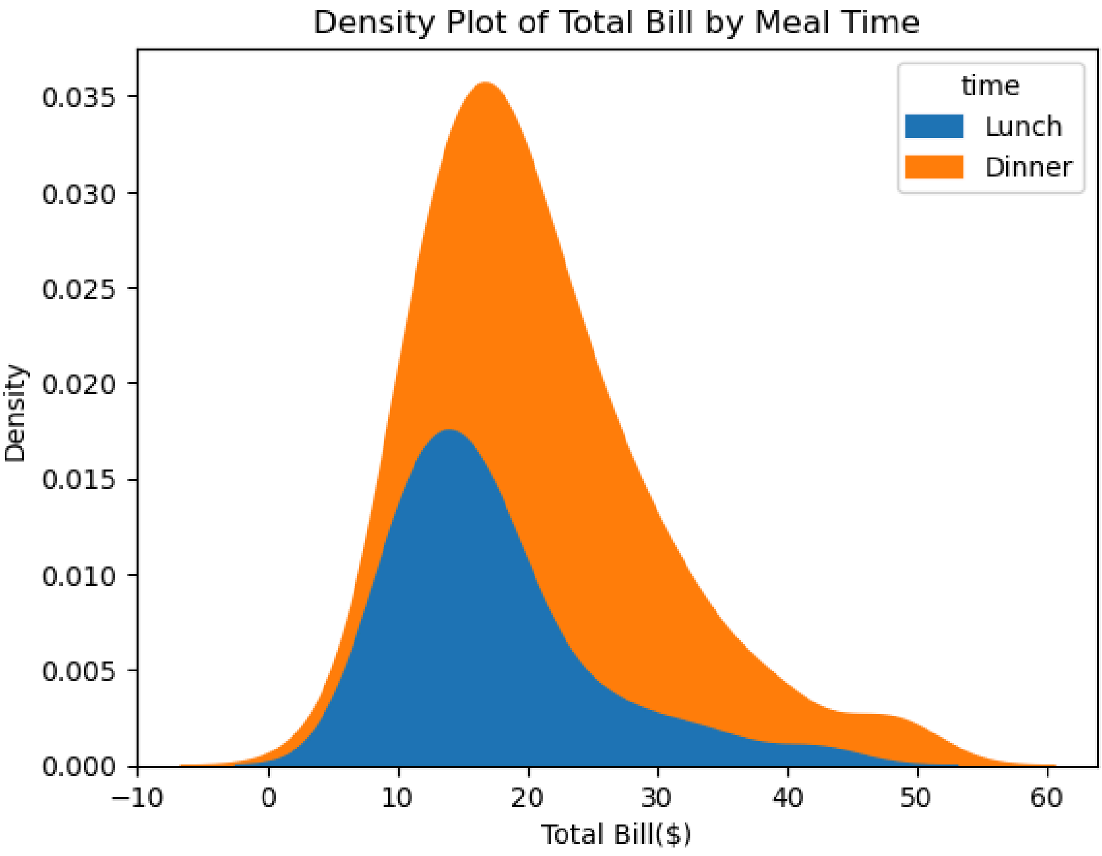
Density plots, also known as kernel density plots, are a type of data visualization that display the distribution of a continuous variable. They are similar to histograms, but instead of representing the data as bars, density plots use a smooth curve to estimate the density of the data. In Seaborn, density plots can be created using the kdeplot() function.

```
In [48]: import seaborn as sns
         tips = sns.load_dataset("tips")
         sns.kdeplot(data=tips,x="total_bill")

Out[48]: <Axes: xlabel='total_bill', ylabel='Density'>
```



Create a density plot of the "total_bill" column from the "tips" dataset We use the "hue" parameter to differentiate between "lunch" and "dinner" meal times We use the "fill" parameter to fill the area under the curve We adjust the "alpha" and "linewidth" parameters to make the plot more visually appealing

In [57]:
```
import seaborn as sns
import matplotlib.pyplot as plt
# Load the "tips" dataset from Seaborn
tips = sns.load_dataset("tips")
sns.kdeplot(data=tips,x="total_bill",hue="time",fill=True,alpha=1,linewidt
plt.xlabel("Total Bill($)")
plt.ylabel("Density")
plt.title("Density Plot of Total Bill by Meal Time")
plt.show()
```



In [3]:
```
import seaborn as sns
import matplotlib.pyplot as plt
tips=sns.load_dataset("tips")
tips.head()
```

Out[3]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

In [36]:
```
tips.shape
```

Out[36]:
```
(244, 7)
```

In [7]:

```
## Statistical Analysis
## In statistical analysis, first, we use the df.describe() which will give
tips.describe()
## The above table shows the count, mean, standard deviation, min, 25%, 50%
```

Out[7]:

	total_bill	tip	size
count	244.000000	244.000000	244.000000
mean	19.785943	2.998279	2.569672
std	8.902412	1.383638	0.951100
min	3.070000	1.000000	1.000000
25%	13.347500	2.000000	2.000000
50%	17.795000	2.900000	2.000000
75%	24.127500	3.562500	3.000000
max	50.810000	10.000000	6.000000

In [8]:

```
tips.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 244 entries, 0 to 243
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype   
---  ------      --------------  -----   
 0   total_bill  244 non-null    float64 
 1   tip         244 non-null    float64 
 2   sex         244 non-null    category
 3   smoker      244 non-null    category
 4   day         244 non-null    category
 5   time        244 non-null    category
 6   size        244 non-null    int64   
dtypes: category(4), float64(2), int64(1)
memory usage: 7.4 KB
```

In [11]:

```
## We can Check null values
tips.isnull().sum()
```

Out[11]:

```
total_bill    0
tip           0
sex           0
smoker        0
day           0
time          0
size          0
dtype: int64
```

In [13]:

```
tips.columns
```

Out[13]:

```
Index(['total_bill', 'tip', 'sex', 'smoker', 'day', 'time', 'size'], dtyp
e='object')
```

In [31]:

```
tips["sex"].unique()
```

Out[31]:

```
['Female', 'Male']
Categories (2, object): ['Male', 'Female']
```

In [39]:

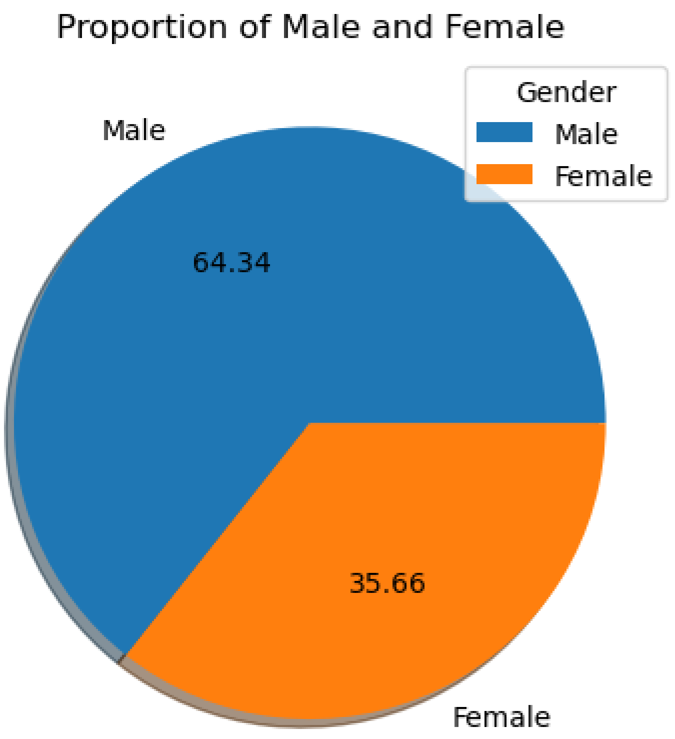
```
tips["sex"].value_counts()
```

Out[39]:

```
sex
Male      157
Female     87
Name: count, dtype: int64
```

In [52]:

```
plt.pie(tips["sex"].value_counts(),labels=["Male","Female"],autopct="%0.2f
plt.title("Proportion of Male and Female")
plt.legend(title="Gender")
plt.show()
```



In [32]:

```
tips["smoker"].unique()
```

Out[32]:

```
['No', 'Yes']
Categories (2, object): ['Yes', 'No']
```

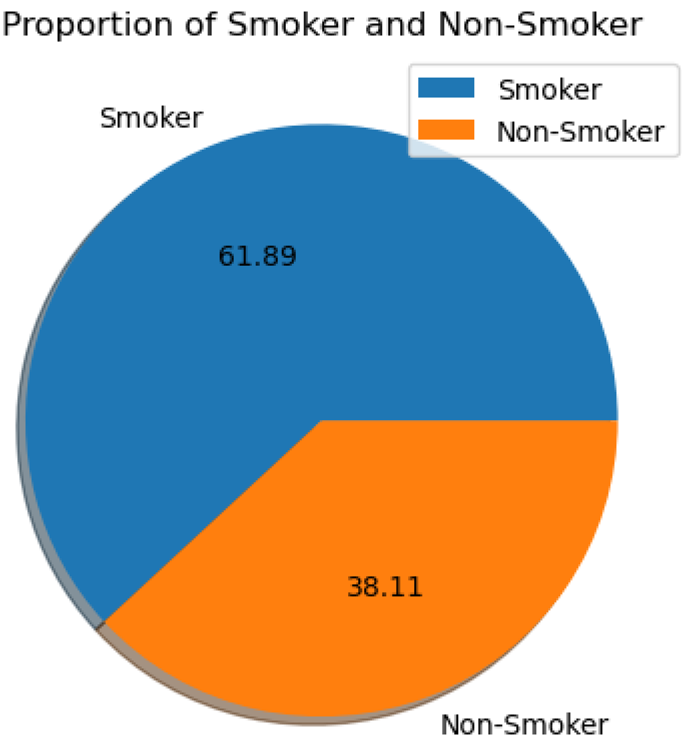
In [40]:

```
tips["smoker"].value_counts()
```

Out[40]:

```
smoker
No     151
Yes     93
Name: count, dtype: int64
```

In [49]:
```
plt.pie(tips["smoker"].value_counts(),labels=["Smoker","Non-Smoker"],autop
plt.title("Proportion of Smoker and Non-Smoker")
plt.legend()
plt.show()
```



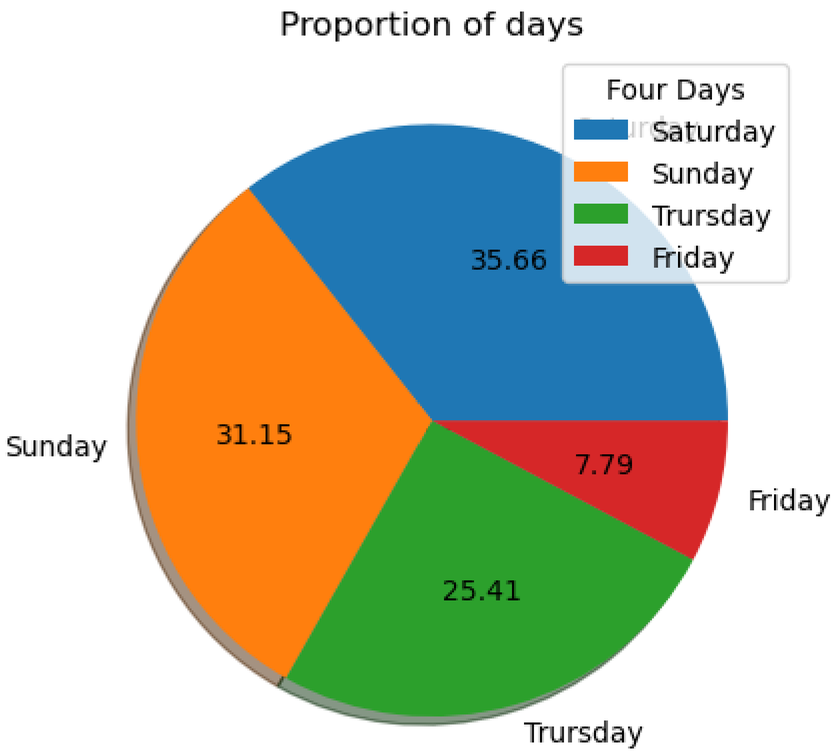
In [33]:
```
tips["day"].unique()
```

Out[33]:
```
['Sun', 'Sat', 'Thur', 'Fri']
Categories (4, object): ['Thur', 'Fri', 'Sat', 'Sun']
```

In [41]:
```
tips["day"].value_counts()
```

Out[41]:
```
day
Sat     87
Sun     76
Thur    62
Fri     19
Name: count, dtype: int64
```

In [50]:
```
plt.pie(tips["day"].value_counts(),labels=["Saturday","Sunday","Trursday","
plt.title("Proportion of days")
plt.legend(title="Four Days")
plt.show()
```



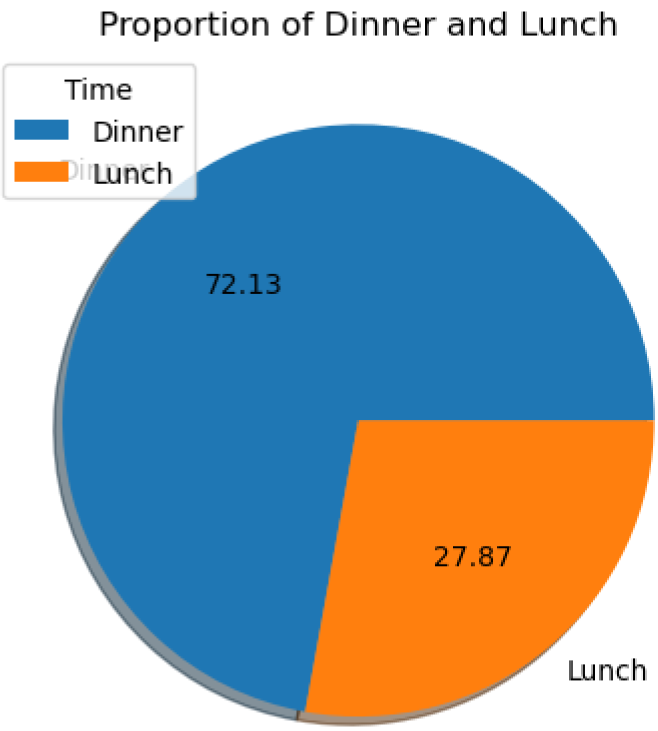
In [34]:
```
tips["time"].unique()
```

Out[34]:
```
['Dinner', 'Lunch']
Categories (2, object): ['Lunch', 'Dinner']
```

In [42]:
```
tips["time"].value_counts()
```

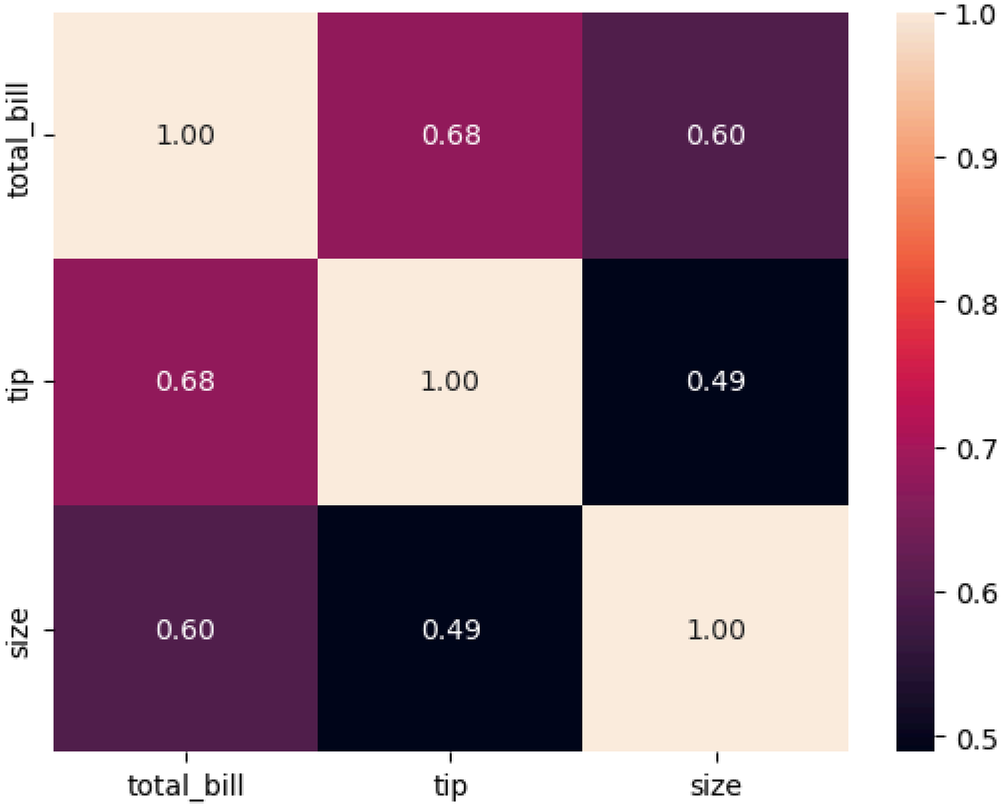
Out[42]:
```
time
Dinner    176
Lunch      68
Name: count, dtype: int64
```

In [51]:
```
plt.pie(tips["time"].value_counts(),labels=["Dinner","Lunch"],autopct="%0.
plt.title("Proportion of Dinner and Lunch")
plt.legend(title="Time")
plt.show()
```



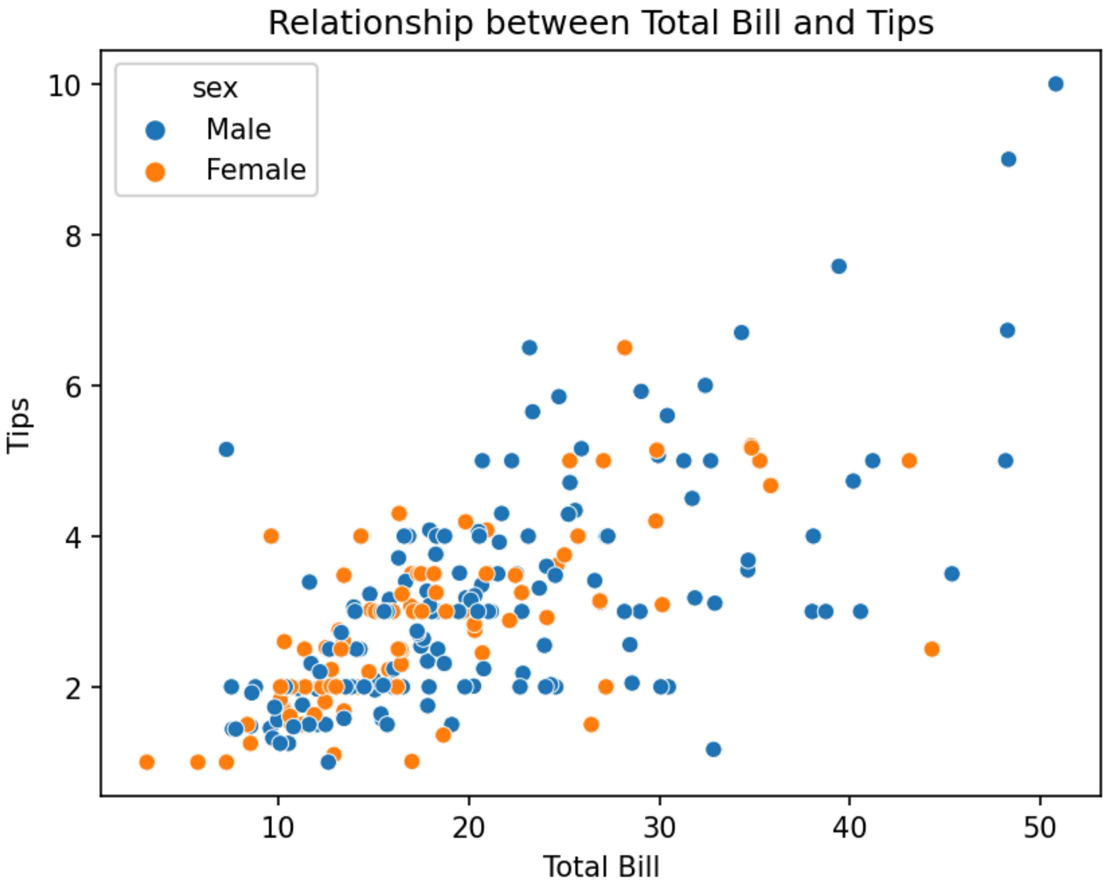
In [25]:

```
df=tips[['total_bill', 'tip',"size"]]
#correlation
corr = df.corr()

plt.figure(dpi=100)
sns.heatmap(df.corr(),annot=True,fmt="0.2f")
plt.show()
```



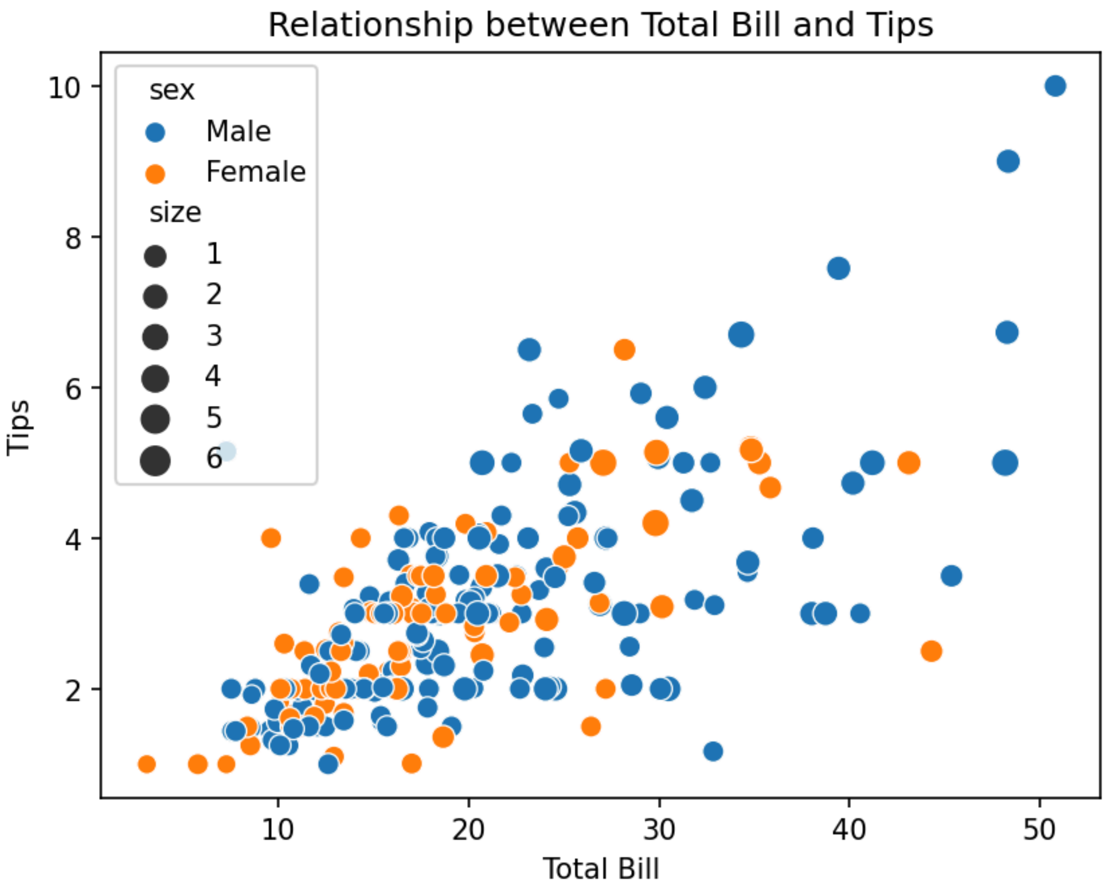
In [27]:

```
import seaborn as sns
import matplotlib.pyplot as plt
tips=sns.load_dataset("tips")
plt.figure(dpi=150)
sns.scatterplot(x="total_bill",y="tip",hue="sex",sizes=(50,100),data=tips)
plt.xlabel("Total Bill")
plt.ylabel("Tips")
plt.title("Relationship between Total Bill and Tips")
plt.show()
```



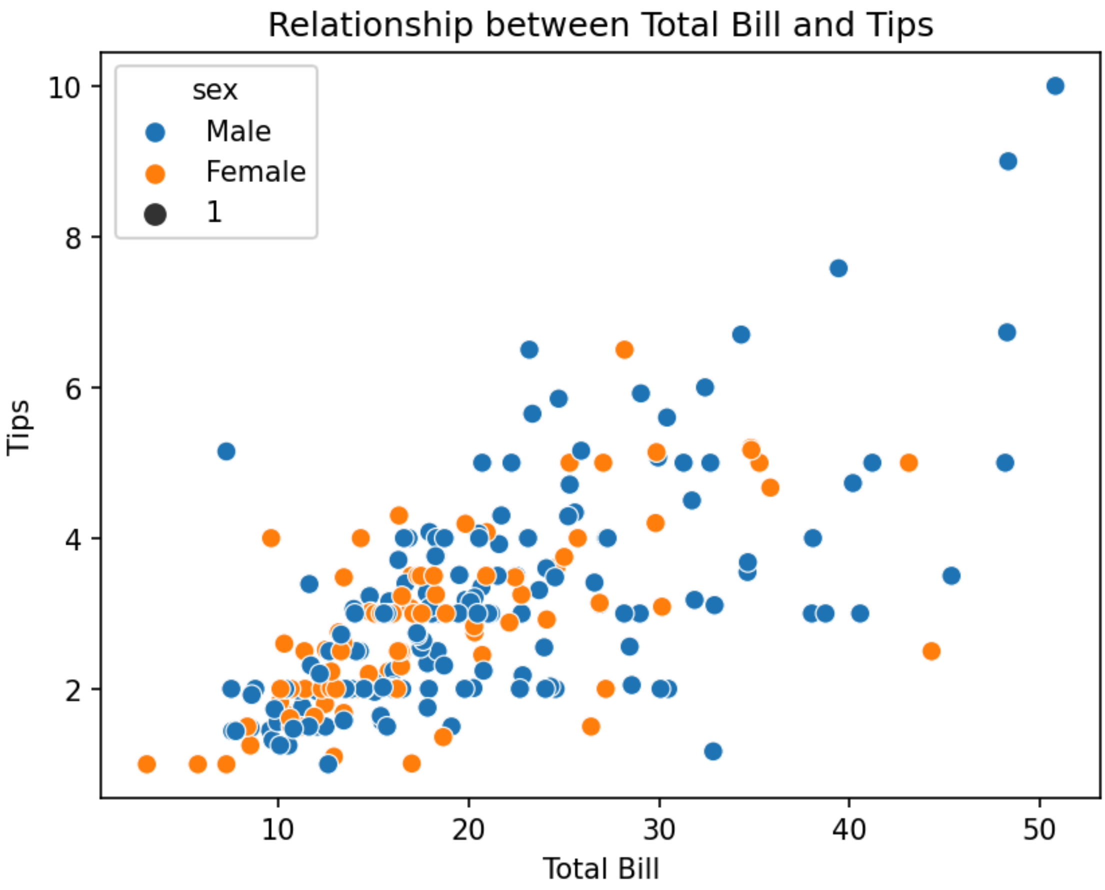
In [28]:

```
import seaborn as sns
import matplotlib.pyplot as plt
tips=sns.load_dataset("tips")
plt.figure(dpi=150)
sns.scatterplot(x="total_bill",y="tip",hue="sex",size="size",sizes=(50,100
plt.xlabel("Total Bill")
plt.ylabel("Tips")
plt.title("Relationship between Total Bill and Tips")
plt.show()
```



In [30]:

```
import seaborn as sns
import matplotlib.pyplot as plt
tips=sns.load_dataset("tips")
plt.figure(dpi=150)
sns.scatterplot(x="total_bill",y="tip",hue="sex",size=1,sizes=(50,100),dat
plt.xlabel("Total Bill")
plt.ylabel("Tips")
plt.title("Relationship between Total Bill and Tips")
plt.show()
```



In [ ]: