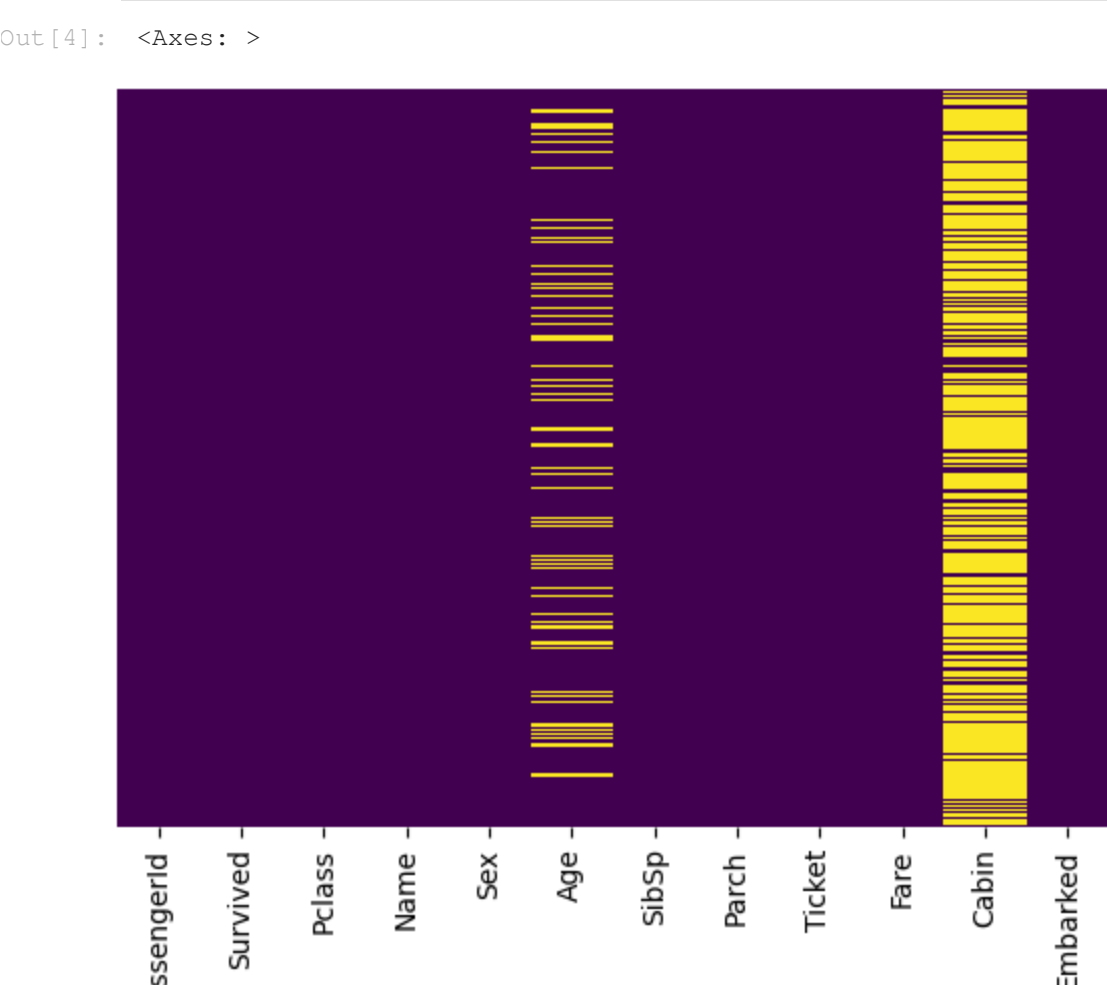


```
In [2]: import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
%matplotlib inline
train=pd.read_csv('train.csv')
train.head()
#importing the necessary libraries for EDA
```

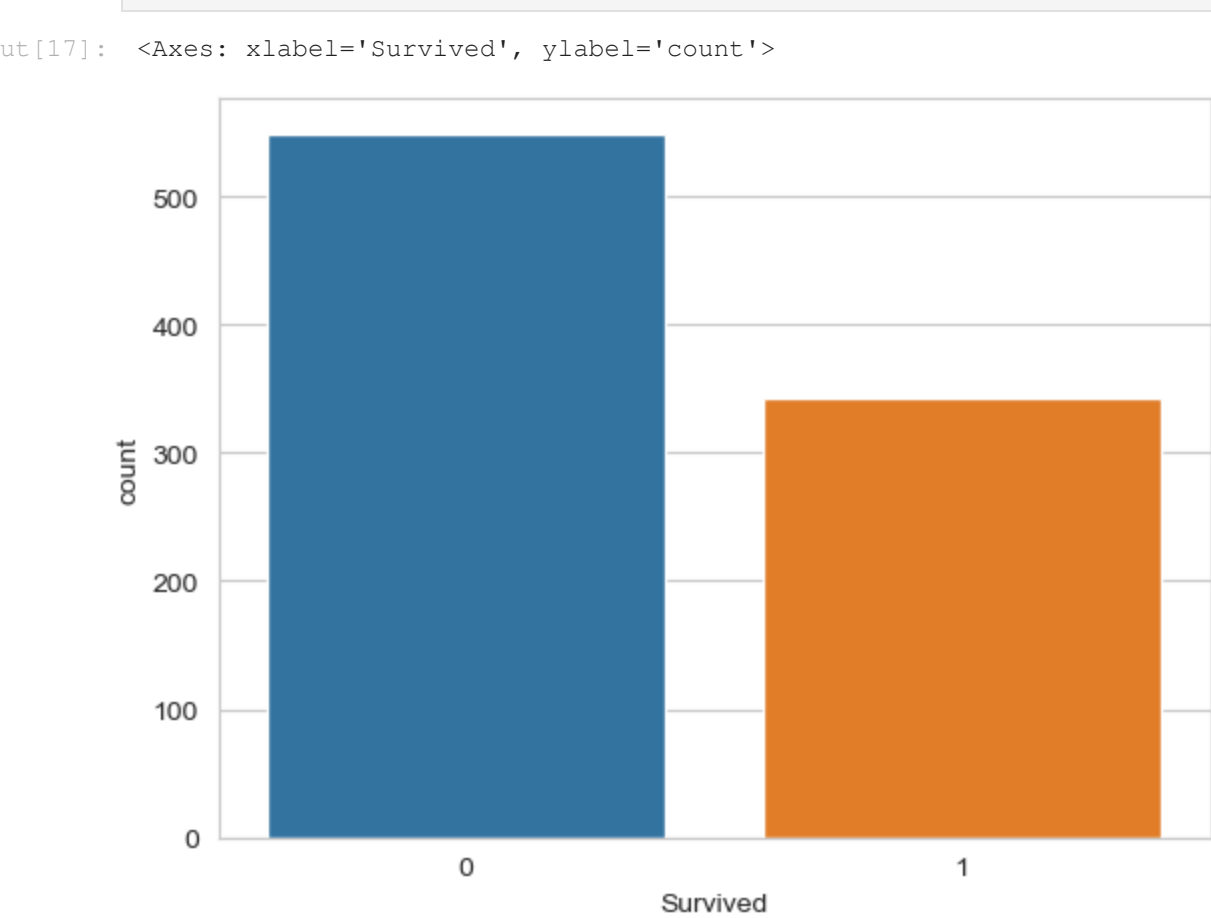
Out[2]:

	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	Cummings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

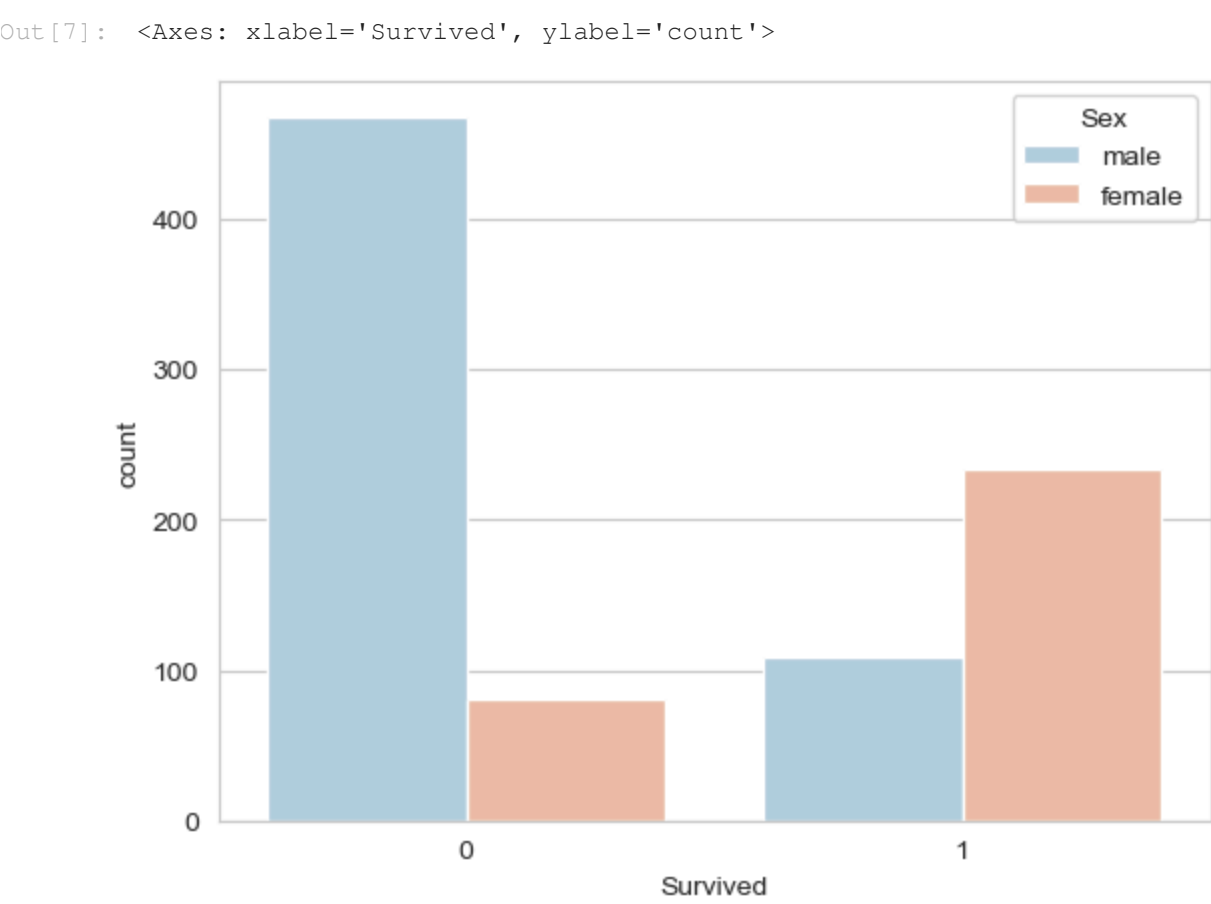
```
In [4]: sns.heatmap(train.isnull(),yticklabels=False,cbar=False,cmap='viridis')
#display of nan values in a column using a heatmap
```



```
In [17]: sns.set_style('whitegrid')
sns.countplot(x='Survived',data=train)
#countplot for displaying persons who have survived and persons who have not
```



```
In [7]: sns.set_style('whitegrid')
sns.countplot(x='Survived',hue='Sex',data=train,palette='RdBu_r')
#gender distribution of number of people survived and people who did not
```



```
In [8]: sns.set_style('whitegrid')
sns.countplot(x='Survived',hue='Pclass',data=train,palette='rainbow')
#countplot for showing survival of number of people distributed according to the passenger-class(Pclass)
```

AttributeError

Traceback (most recent call last)

Cell In[8], line 2

```
1 sns.set_style('whitegrid')
----> 2 sns.countplot(x='Survived',hue='Pclass',data=train,palette='rainbow')
```

File ~\anaconda3\lib\site-packages\seaborn\categorical.py:2955, in countplot(data, x, y, hue, order, hue_order, orient, color, palette, saturation, width, dodge, ax, **kwargs)

```
2952 if ax is None:
2953     ax = plt.gca()
-> 2955 plotter.plot(ax, kwargs)
2956 return ax
```

File ~\anaconda3\lib\site-packages\seaborn\categorical.py:1587, in _BarPlotter.plot(self, ax, bar_kwargs)

```
1585 """Make the plot."""
1586 self.draw_bar(ax, bar_kwargs)
-> 1587 self.annotate_axes(ax)
1588 if self.orient == "h":
1589     ax.invert_yaxis()
```

File ~\anaconda3\lib\site-packages\seaborn\categorical.py:767, in _CategoricalPlotter.annotate_axes(self, ax)

```
764 ax.set_ylim(-.5, len(self.plot_data) - .5, auto=None)
766 if self.hue_names is not None:
-> 767     ax.legend(loc="best", title=self.hue_title)
```

File ~\anaconda3\lib\site-packages\matplotlib\axes.py:322, in Axes.legend(self, *args, **kwargs)

```
204 @docstring.dedent_interpd
205 def legend(self, *args, **kwargs):
206     """
207     Place a legend on the Axes.
208
209     (...)
320     .. plot:: gallery/text_labels_and_annotations/legend.py
321     """
-> 322     handles, labels, kwargs = mlegend._parse_legend_args(self, *args, **kwargs)
323     self.legend_ = mlegend.Legend(self, handles, labels, **kwargs)
324     self.legend._remove_method = self._remove_legend
```

File ~\anaconda3\lib\site-packages\matplotlib\legend.py:1361, in _parse_legend_arg(axs, handles, labels, *args, **kwargs)

```
1357 handles = [handle for handle, label
1358               in zip(_get_legend_handles(axs, handles), labels)]
1360 elif len(args) == 0: # 0 args: automatically detect labels and handles.
-> 1361     handles, labels = _get_legend_handles_labels(axs, handles)
1362     if not handles:
1363         log.warning(
1364             "No artists with labels found to put in legend. Note that "
1365             "artists whose label start with an underscore are ignored "
1366             "when legend() is called with no argument.")
```

File ~\anaconda3\lib\site-packages\matplotlib\legend.py:1291, in _get_legend_handles_labels(axs, legend_handler_map)

```
1289 for handle in _get_legend_handles(axs, legend_handler_map):
1290     label = handle.get_label()
-> 1291     if label and not label.startswith('_'):
1292         handles.append(handle)
1293         labels.append(label)
```

AttributeError: 'numpy.int64' object has no attribute 'startswith'

```
In [9]: sns.distplot(train['Age'],kde=False,color='darkred',bins=40)
#distibution of age
```

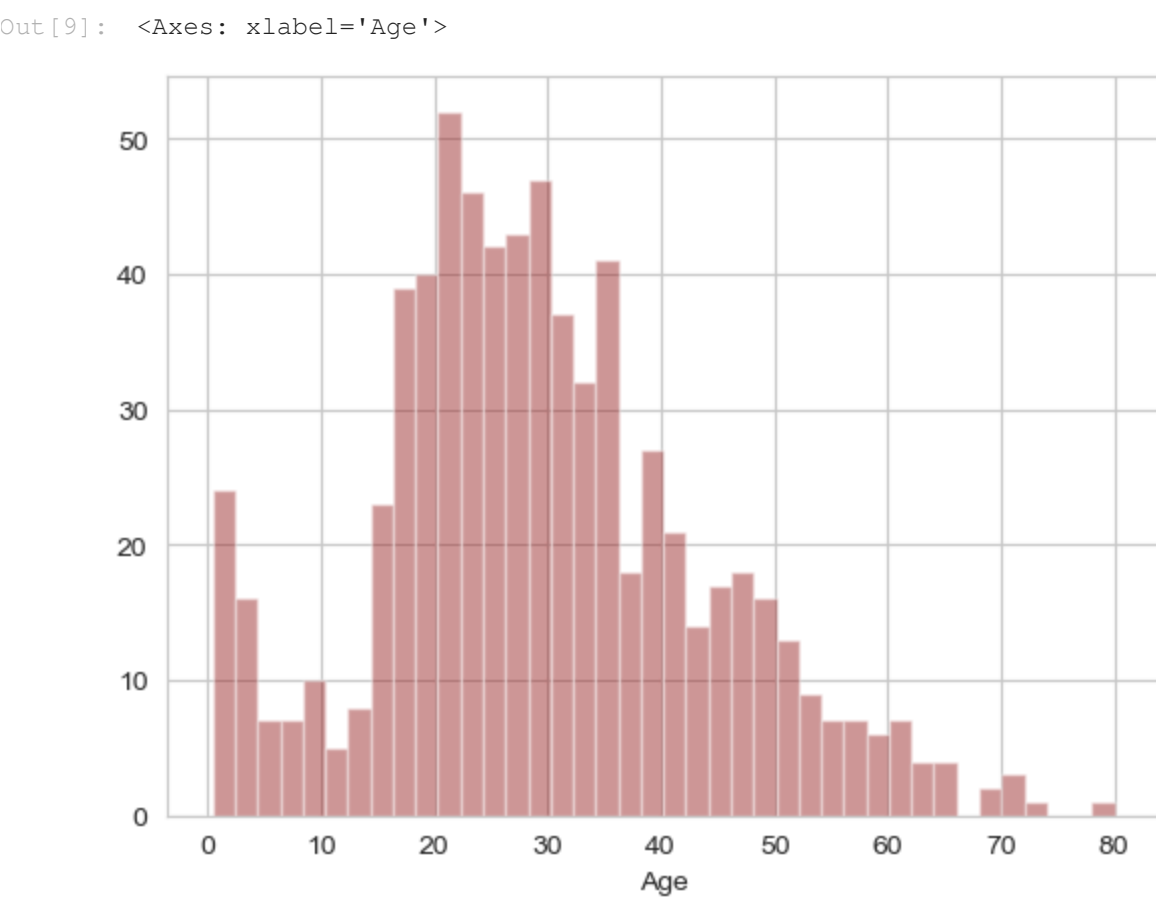
C:\Users\Ajagmeet Kaur\AppData\Local\Temp\ipykernel_7508\2002818437.py:1: UserWarning:

'distplot' is a deprecated function and will be removed in seaborn v0.14.0.

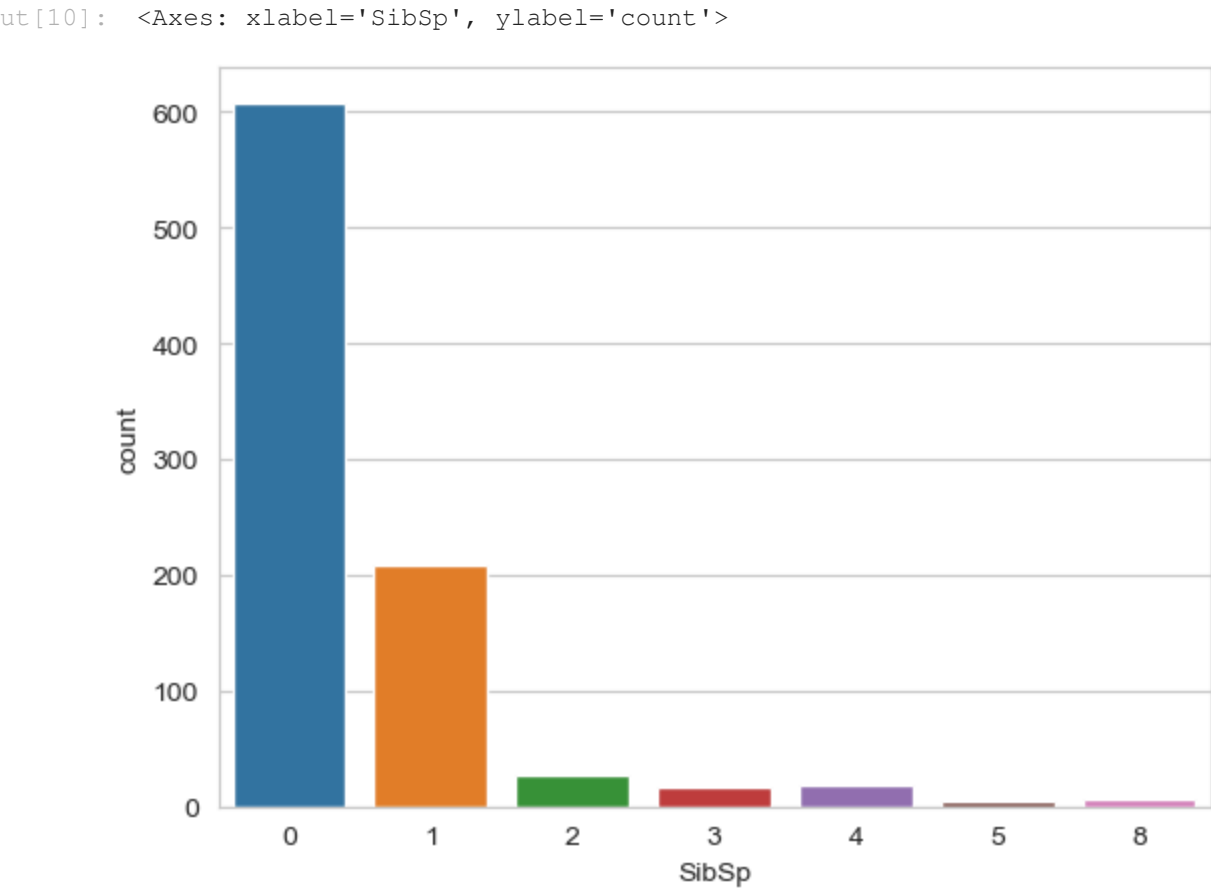
Please adapt your code to use either 'displot' (a figure-level function with similar flexibility) or 'histplot' (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see <https://gist.github.com/awaskin/de4147ed277457ade6372750bbe5751>

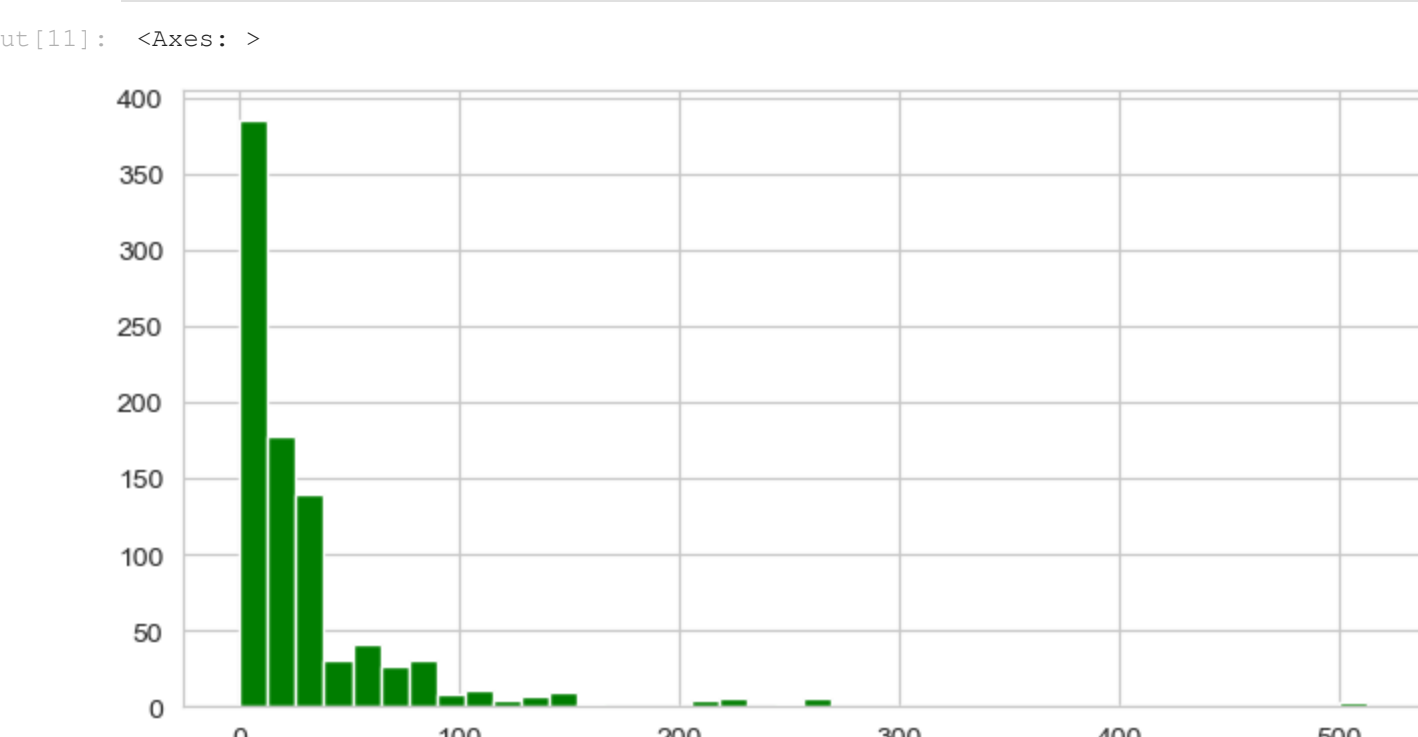
```
sns.distplot(train['Age'],kde=False,color='darkred',bins=40)
```



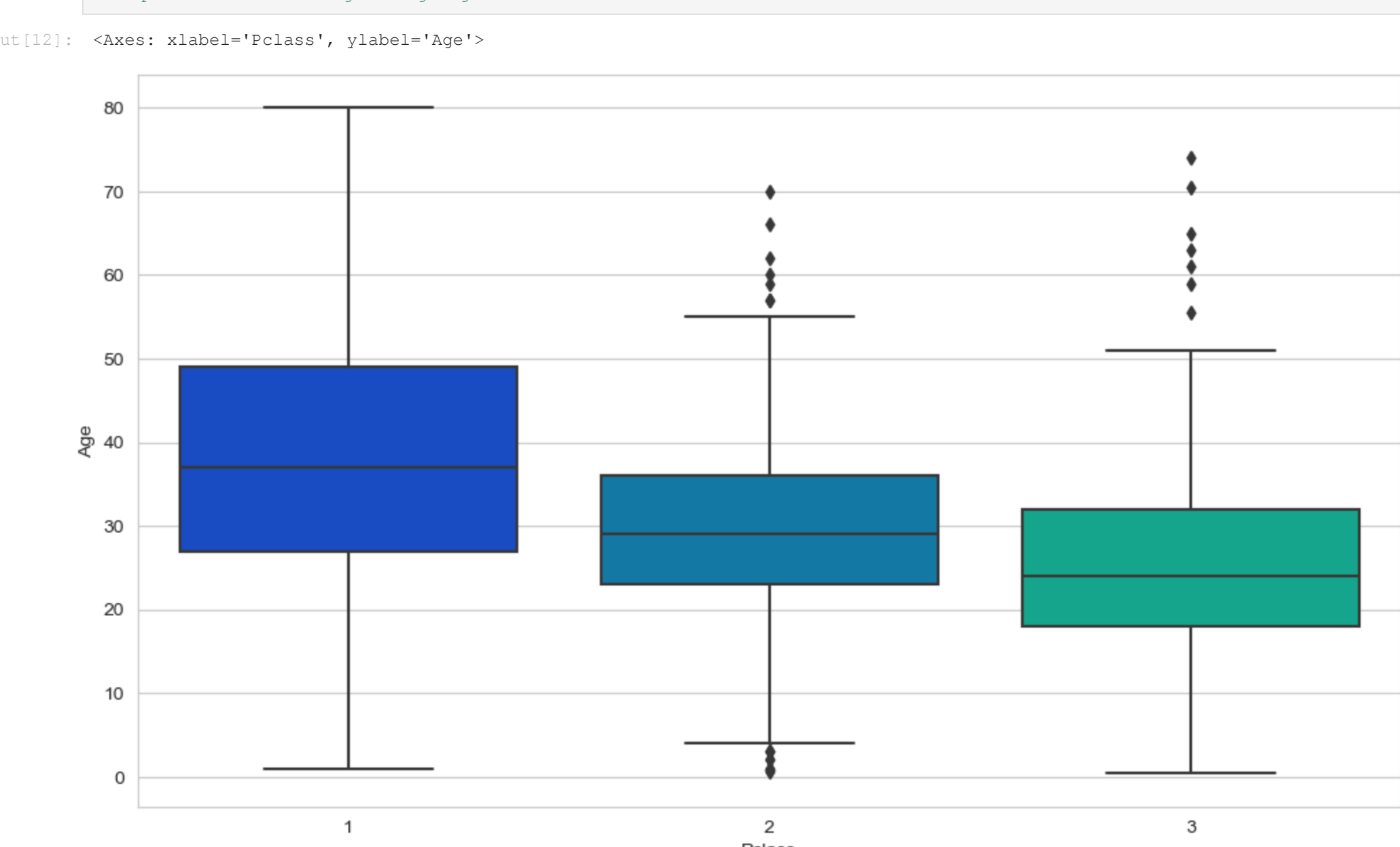
```
In [10]: sns.countplot(x='SibSp',data=train)
#frequency showing the count of siblings
```



```
In [11]: train['Fare'].hist(color='green',bins=40,figsize=(8,4))
#train fare histogram
```



```
In [12]: plt.figure(figsize=(12,7))
sns.boxplot(x='Pclass',y='Age',data=train,palette='winter')
#box plot for visualising average age
```



```
In [13]: def impute_age(cols):
Age=cols[0]
Pclass=cols[1]
if pd.isnull(Age):
    if Pclass==1:
        return 21
    elif Pclass==2:
        return 29
    else:
        return Age
else:
    return Age
```

```
In [14]: train['Age']=train[['Age','Pclass']].apply(impute_age,axis=1)
```

C:\Users\Ajagmeet Kaur\AppData\Local\Temp\ipykernel_7508\2856058251.py:2: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future version, integer keys will always be treated as labels (consistent with DataFrame behavior). To access a value by position, use `ser.iloc[pos]`

```
Age=cols[0]
C:\Users\Ajagmeet Kaur\AppData\Local\Temp\ipykernel_7508\2856058251.py:3: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future version, integer keys will always be treated as labels (consistent with DataFrame behavior). To access a value by position, use `ser.iloc[pos]`
Pclass=cols[1]
```

```
In [16]: sns.heatmap(train.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

Out[16]: <Axes: >

PassengerIdSurvivedPclassNameSexAge SibSp Parch Ticket Fare Cabin Embarked

