Let's start by adding some libraries. ¶

In [1]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

We load the dataset.

In [2]:

```
train = pd.read_csv("train.csv")
```

Let's see the data a little bit.

In [3]:

```
train.head()
```

Out[3]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	C
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	

In [4]:

```
train.count()
```

Out[4]:

```
PassengerId    891
Survived       891
Pclass         891
Name           891
Sex            891
Age            714
SibSp          891
Parch          891
Ticket         891
Fare           891
Cabin          204
Embarked       889
dtype: int64
```

In [5]:

```
train.describe()
```

Out[5]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

In [6]:

```
train[train['Sex'].str.match("female")].count()
```

Out[6]:

```
PassengerId    314
Survived       314
Pclass         314
Name           314
Sex            314
Age            261
SibSp          314
Parch          314
Ticket         314
Fare           314
Cabin           97
Embarked       312
dtype: int64
```

Missing Data

- We can use seaborn to create a simple heatmap to see where we are missing data!

In [7]:

```
train.isnull()
```

Out[7]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	I
0	False	False	False	False	False	False	False	False	False	False	True	
1	False	False	False	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	False	False	True	
3	False	False	False	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	False	False	True	
...	...	...	...	...	...	...	...	...	...	...	...	
886	False	False	False	False	False	False	False	False	False	False	True	
887	False	False	False	False	False	False	False	False	False	False	False	
888	False	False	False	False	False	True	False	False	False	False	True	
889	False	False	False	False	False	False	False	False	False	False	False	
890	False	False	False	False	False	False	False	False	False	False	True	

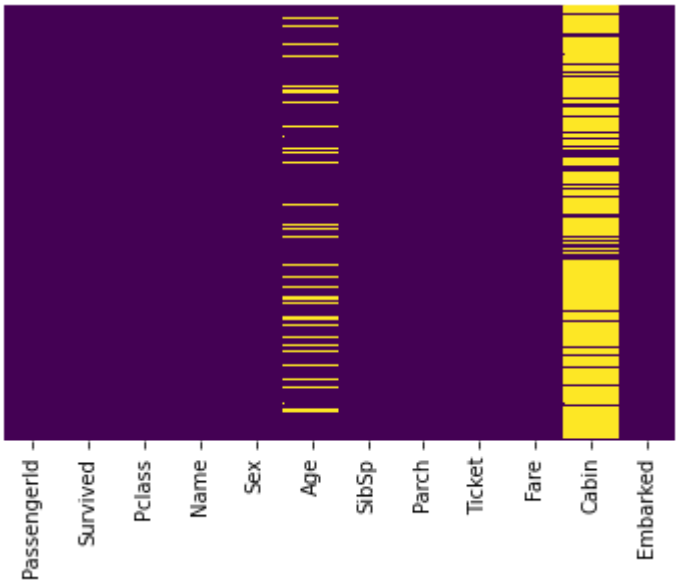
891 rows × 12 columns

In [8]:

```
sns.heatmap(train.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

Out[8]:

```
<AxesSubplot:>
```



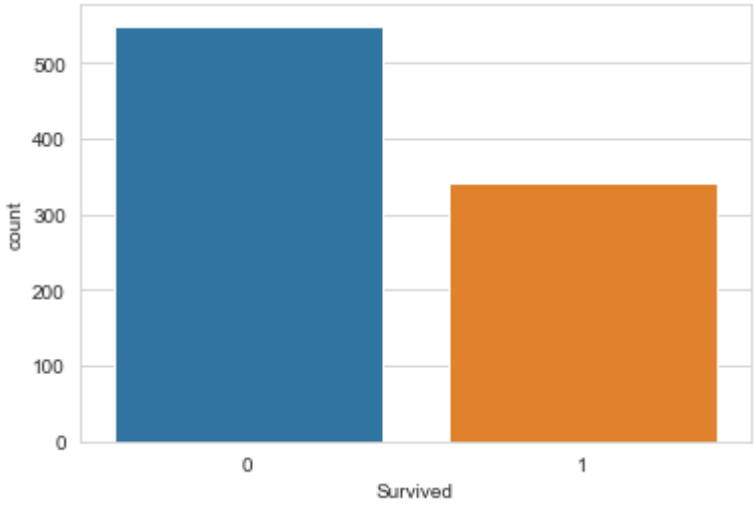
- Roughly 20 percent of the Age data is missing. The proportion of Age missing is likely small enough for reasonable replacement with some form of imputation. Looking at the Cabin column, it looks like we are just missing too much of that data to do something useful with at a basic level. We'll probably drop this later, or change it to another feature like "Cabin Known: 1 or 0"
- Let's continue on by visualizing some more of the data! Check out the video for full explanations over these plots, this code is just to serve as reference.

In [9]:

```
sns.set_style('whitegrid')
sns.countplot(x='Survived',data=train)
```

Out[9]:

```
<AxesSubplot:xlabel='Survived', ylabel='count'>
```

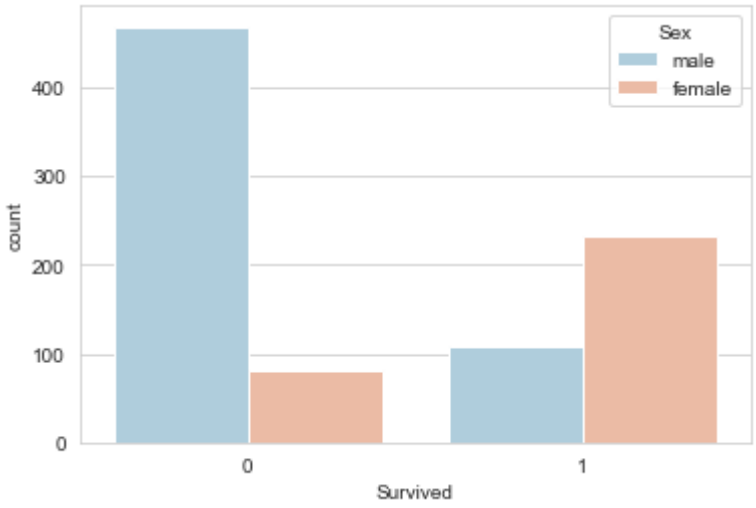


In [10]:

```
sns.set_style('whitegrid')
sns.countplot(x='Survived',hue='Sex',data=train,palette='RdBu_r')
```

Out[10]:

```
<AxesSubplot:xlabel='Survived', ylabel='count'>
```

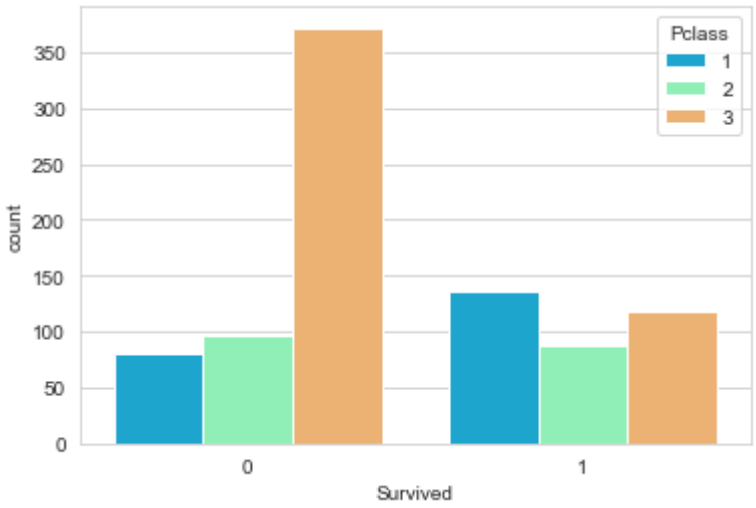


In [11]:

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

Out[11]:

```
<AxesSubplot:xlabel='Survived', ylabel='count'>
```



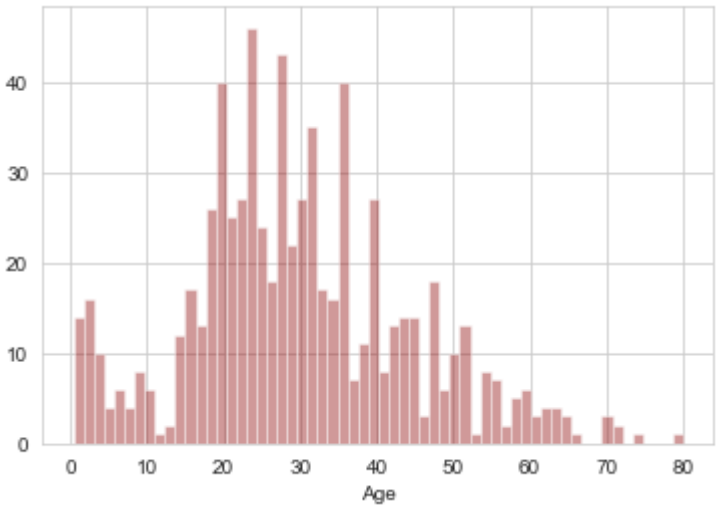
In [12]:

```
sns.distplot(train['Age'].dropna(),kde=False,color='darkred',bins=60)
```

```
C:\Users\Admin\anaconda3\lib\site-packages\seaborn\distributions.py:2551: F
tureWarning: `distplot` is a deprecated function and will be removed in a fu
ture version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

Out[12]:

```
<AxesSubplot:xlabel='Age'>
```

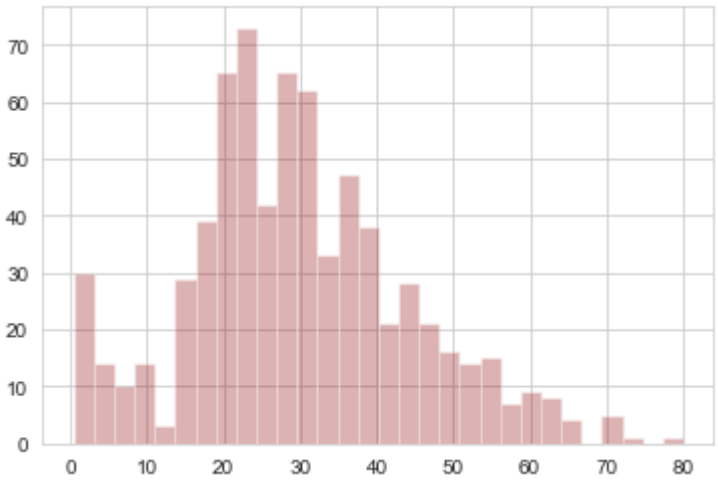


In [13]:

```
train['Age'].hist(bins=30,color='darkred',alpha=0.3)
```

Out[13]:

```
<AxesSubplot:>
```

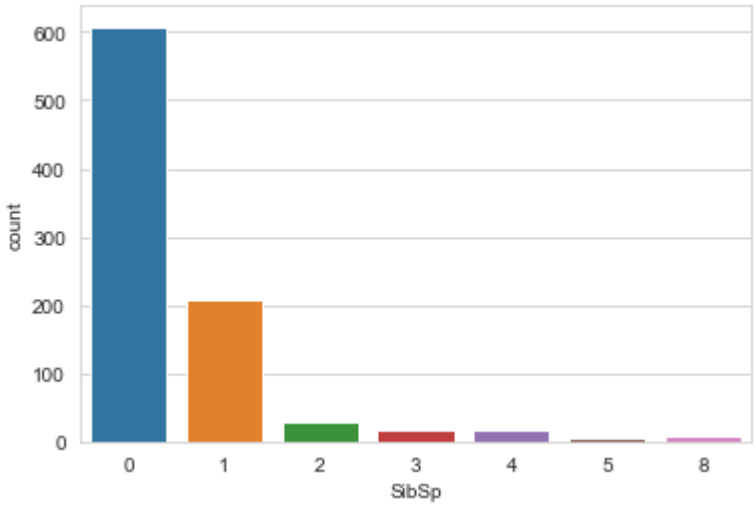


In [14]:

```
sns.countplot(x='SibSp',data=train)
```

Out[14]:

```
<AxesSubplot:xlabel='SibSp', ylabel='count'>
```

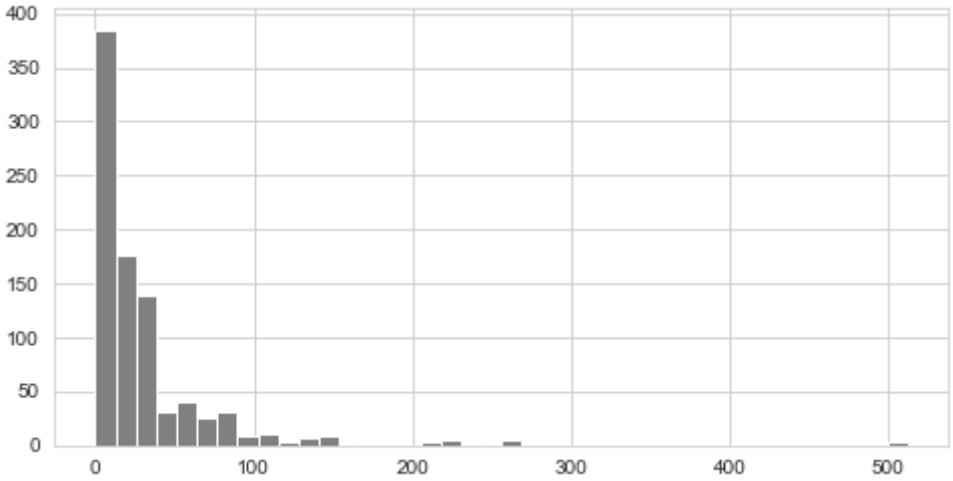


In [15]:

```
train['Fare'].hist(color='grey',bins=40,figsize=(8,4))
```

Out[15]:

```
<AxesSubplot:>
```



Data Cleaning

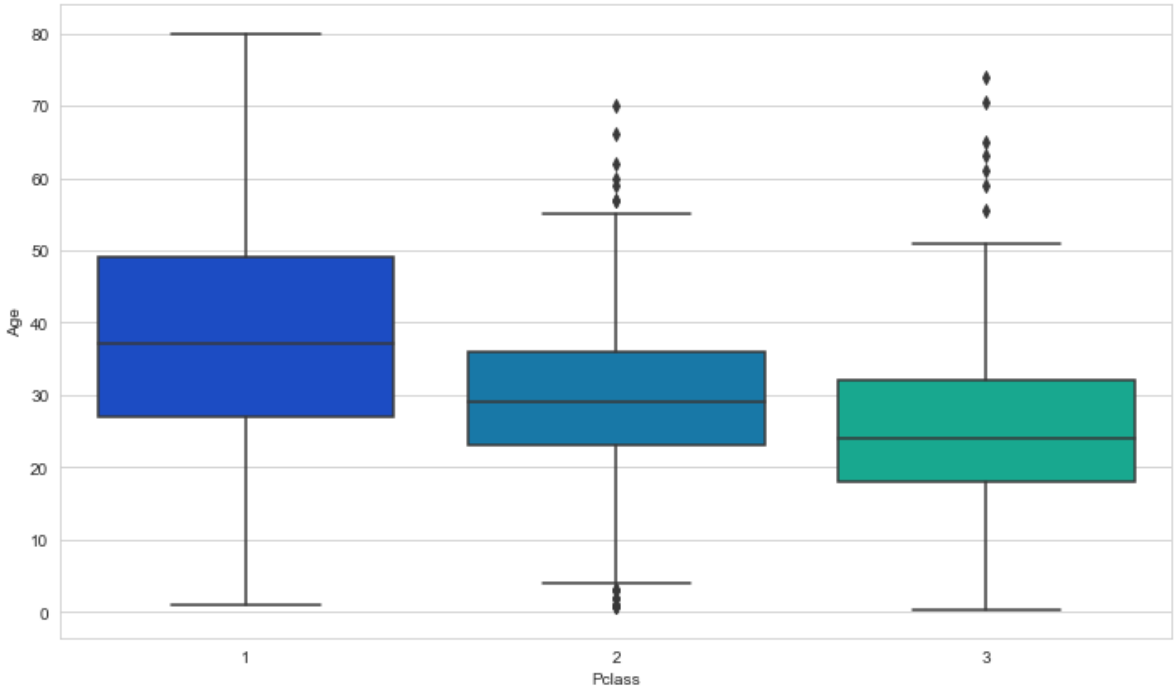
- We want to fill in missing age data instead of just dropping the missing age data rows. One way to do this is by filling in the mean age of all the passengers . However we can be smarter about this and check the average age by passenger class.

In [16]:

```
plt.figure(figsize=(12, 7))
sns.boxplot(x='Pclass',y='Age',data=train,palette='winter')
```

Out[16]:

```
<AxesSubplot:xlabel='Pclass', ylabel='Age'>
```



- We can see the wealthier passengers in the higher classes tend to be older, which makes sense.
- We'll use these average age values to impute based on Pclass for Age.

In [17]:

```
def impute_age(cols):
    Age = cols[0]
    Pclass = cols[1]

    if pd.isnull(Age):

        if Pclass == 1:
            return 37

        elif Pclass == 2:
            return 29

        else:
            return 24

    else:
        return Age
```

- Now apply that function!

In [18]:

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

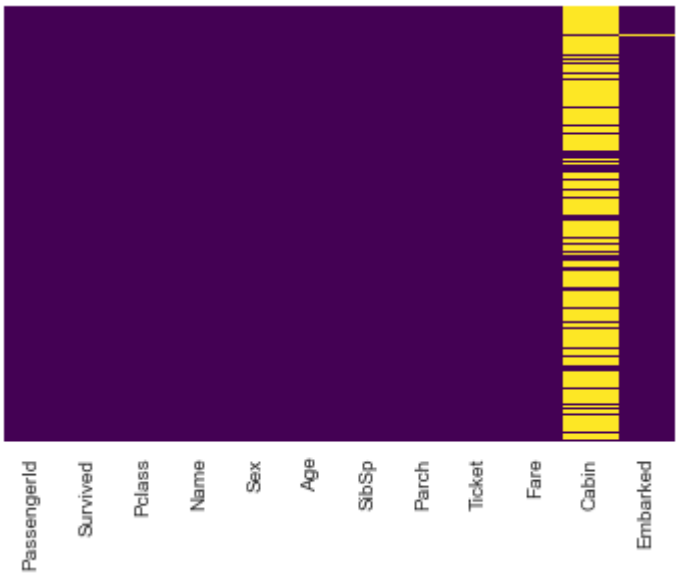
- Now let's check that heat map again!

In [19]:

```
sns.heatmap(train.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

Out[19]:

```
<AxesSubplot:>
```



- Great! Let's go ahead and drop the Cabin column and the row in Embarked that is NaN.

In [20]:

```
train.drop('Cabin',axis=1,inplace=True)
```

In [21]:

```
train.head()
```

Out[21]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	I
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	

In [22]:

```
train.dropna(inplace=True)
```

In [ ]:

```

```

- Lets perform modeling in the next note book

In [ ]:

```

```