Importing libraries and data

In [1]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
```

In [2]:

```
df=sns.load_dataset('titanic')
df
```

Out[2]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_m
0	0	3	male	22.0	1	0	7.2500	S	Third	man	T
1	1	1	female	38.0	1	0	71.2833	C	First	woman	Fa
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	Fa
3	1	1	female	35.0	1	0	53.1000	S	First	woman	Fa
4	0	3	male	35.0	0	0	8.0500	S	Third	man	T
...	...	...	...	...	...	...	...	...	...	...	
886	0	2	male	27.0	0	0	13.0000	S	Second	man	T
887	1	1	female	19.0	0	0	30.0000	S	First	woman	Fa
888	0	3	female	NaN	1	2	23.4500	S	Third	woman	Fa
889	1	1	male	26.0	0	0	30.0000	C	First	man	T
890	0	3	male	32.0	0	0	7.7500	Q	Third	man	T

891 rows × 15 columns

Introductory Details About Data

In [3]:

```
df.head()
```

Out[3]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True

In [4]:

```
df.tail()
```

Out[4]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_mal
886	0	2	male	27.0	0	0	13.00	S	Second	man	Tru
887	1	1	female	19.0	0	0	30.00	S	First	woman	Fals
888	0	3	female	NaN	1	2	23.45	S	Third	woman	Fals
889	1	1	male	26.0	0	0	30.00	C	First	man	Tru
890	0	3	male	32.0	0	0	7.75	Q	Third	man	Tru

In [5]:

```
df.shape
```

Out[5]:

```
(891, 15)
```

In [6]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 15 columns):
 #   Column       Non-Null Count  Dtype   
---  ------       --------------  -----   
 0   survived     891 non-null    int64   
 1   pclass       891 non-null    int64   
 2   sex          891 non-null    object  
 3   age          714 non-null    float64 
 4   sibsp        891 non-null    int64   
 5   parch        891 non-null    int64   
 6   fare         891 non-null    float64 
 7   embarked     889 non-null    object  
 8   class        891 non-null    category
 9   who          891 non-null    object  
 10  adult_male   891 non-null    bool    
 11  deck         203 non-null    category
 12  embark_town  889 non-null    object  
 13  alive        891 non-null    object  
 14  alone        891 non-null    bool    
dtypes: bool(2), category(2), float64(2), int64(4), object(5)
memory usage: 80.7+ KB
```

In [7]:

```
print(df.columns)
```

```
Index(['survived', 'pclass', 'sex', 'age', 'sibsp', 'parch', 'fare',
       'embarked', 'class', 'who', 'adult_male', 'deck', 'embark_town',
       'alive', 'alone'],
      dtype='object')
```

Statistical Insights

In [8]:

```
df.describe()
```

Out[8]:

	survived	pclass	age	sibsp	parch	fare
count	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

Data cleaning- Checking for null values

checking missing values

In [9]:

```
df.isna().sum          #gives the number of missing values for each variable
```

Out[9]:

```
<bound method NDFrame._add_numeric_operations.<locals>.sum of      survived
pclass    sex    age  sibsp  parch   fare  embarked  class  \
0       False   False  False  False  False  False  False     False  False
1       False   False  False  False  False  False  False     False  False
2       False   False  False  False  False  False  False     False  False
3       False   False  False  False  False  False  False     False  False
4       False   False  False  False  False  False  False     False  False
..        ...     ...    ...    ...    ...    ...    ...       ...    ...
886     False   False  False  False  False  False  False     False  False
887     False   False  False  False  False  False  False     False  False
888     False   False  False   True  False  False  False     False  False
889     False   False  False  False  False  False  False     False  False
890     False   False  False  False  False  False  False     False  False

       who  adult_male   deck  embark_town  alive  alone
0    False       False   True        False  False  False
1    False       False  False        False  False  False
2    False       False   True        False  False  False
3    False       False  False        False  False  False
4    False       False   True        False  False  False
..     ...         ...    ...          ...    ...    ...
886  False       False   True        False  False  False
887  False       False  False        False  False  False
888  False       False   True        False  False  False
889  False       False  False        False  False  False
890  False       False   True        False  False  False

[891 rows x 15 columns]>
```

Removing Null Entries

In [10]:

```
#df.dropna(axis=0,inplace=True)    #  If null entries are there
```

In [11]:

```
#df.shape
```

Filling values in place of Null Entries(If Numerical feature)

Values can either be mean, median or any integer

In [12]:

```
data = df.drop_duplicates(subset ="class",)
data
```

Out[12]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_mal
0	0	3	male	22.0	1	0	7.2500	S	Third	man	Tru
1	1	1	female	38.0	1	0	71.2833	C	First	woman	Fals
9	1	2	female	14.0	1	0	30.0708	C	Second	child	Fals

In [13]:

```
df['class'].value_counts()
```

Out[13]:

```
Third     491
First     216
Second    184
Name: class, dtype: int64
```

In [14]:

```
data = df.drop_duplicates(subset ="sex",)
data
```

Out[14]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False

In [15]:

```
df['sex'].value_counts()
```

Out[15]:

```
male      577
female    314
Name: sex, dtype: int64
```

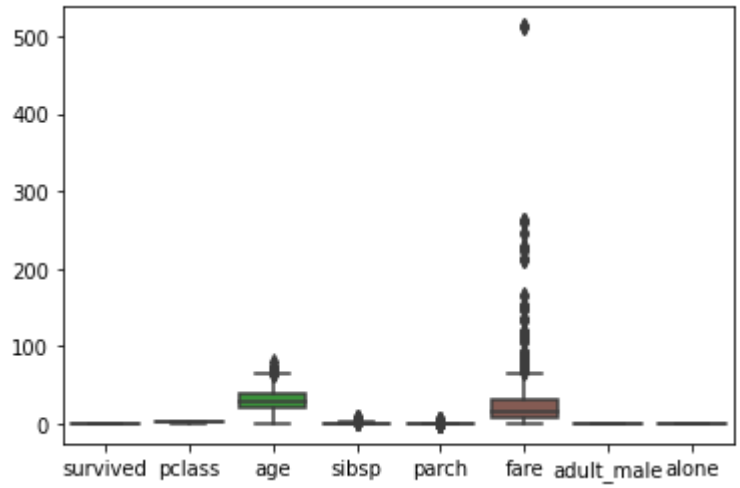
Handling Outliers

In [16]:

```
sns.boxplot(data=df)
```

Out[16]:

<AxesSubplot:>

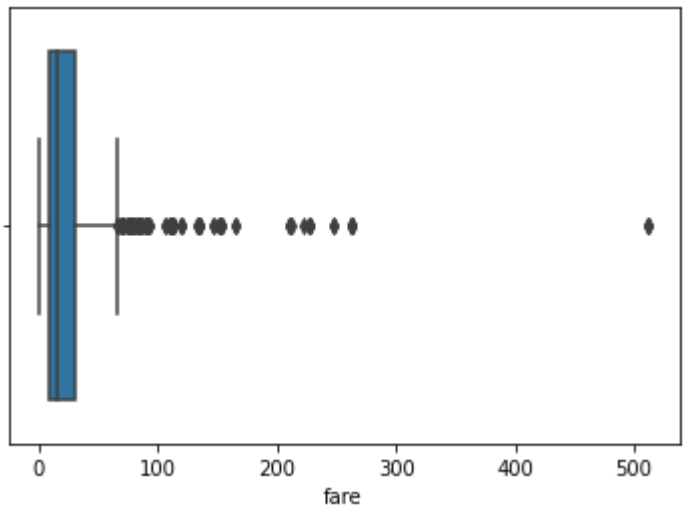


In [17]:

```
sns.boxplot(x ='fare' , data=df)
```

Out[17]:

<AxesSubplot:xlabel='fare'>



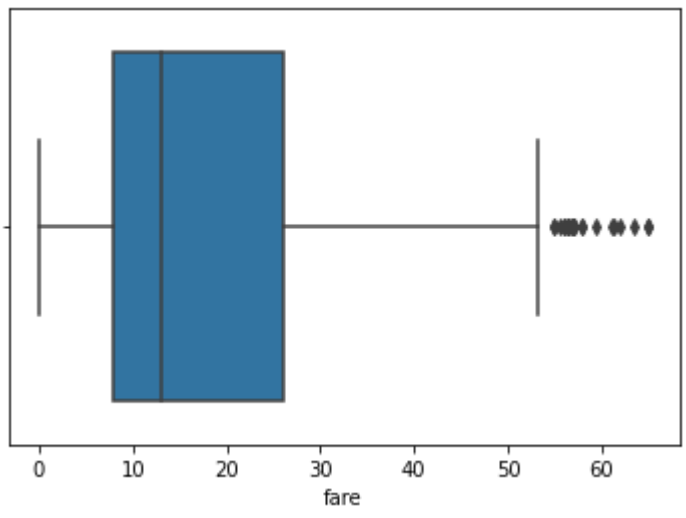
In [18]:

```
# IQR
Q1 = np.percentile(df['fare'], 25,
                   interpolation = 'midpoint')

Q3 = np.percentile(df['fare'], 75,
                   interpolation = 'midpoint')
IQR = Q3 - Q1


print("Old Shape: ", df.shape)

# Upper bound
upper = np.where(df['fare'] >= (Q3+1.5*IQR))

# Lower bound
lower = np.where(df['fare'] <= (Q1-1.5*IQR))

# Removing the Outliers
df.drop(upper[0], inplace = True)
df.drop(lower[0], inplace = True)

print("New Shape: ", df.shape)

sns.boxplot(x='fare', data=df)
```

```
Old Shape:  (891, 15)
New Shape:  (775, 15)
```

Out[18]:

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

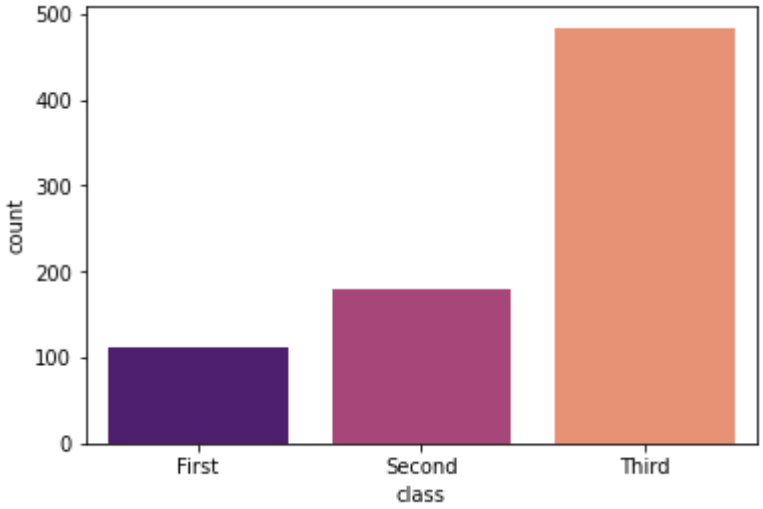


Data Visualization

Visualizing the target column

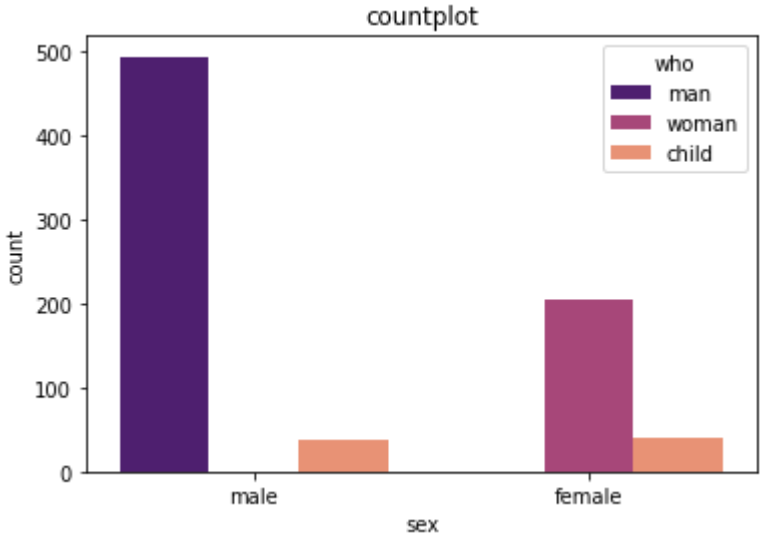
In [19]:

```
sns.countplot(x='class', data=df,palette='magma' )
plt.show()
```



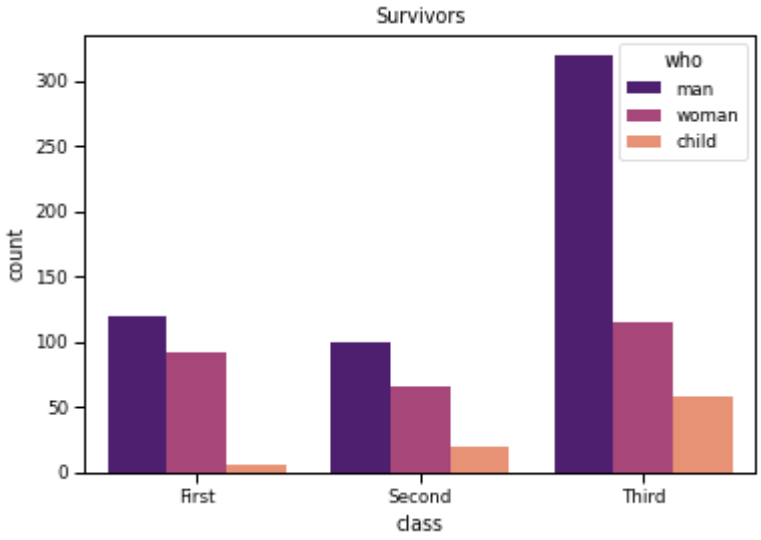
In [20]:

```
sns.countplot(x='sex',hue='who', data=df , palette='magma' )
plt.title('countplot')
plt.show()
```



In [21]:

```
sns.set_context('paper')

# load dataset
titanic = sns.load_dataset('titanic')
# create plot
sns.countplot(x = 'class', hue = 'who', data = titanic, palette = 'magma')
plt.title('Survivors')
plt.show()
```

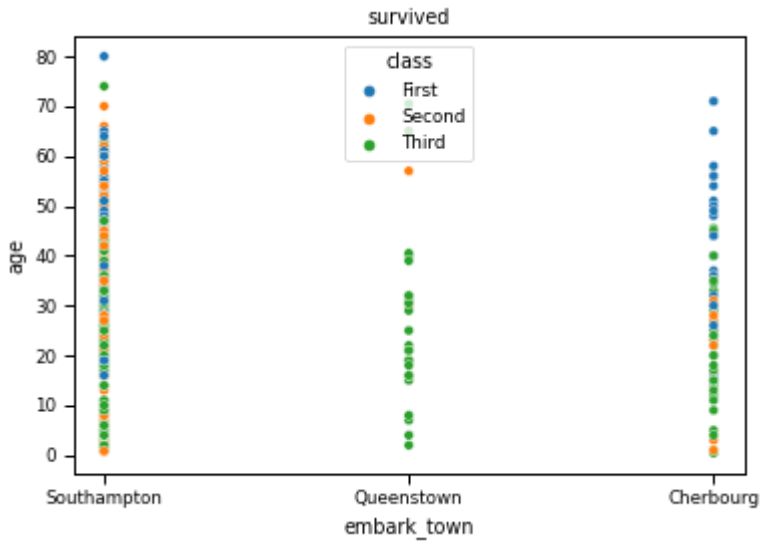


Bivariate Analysis - Scatter plot

Comparing Servival and class

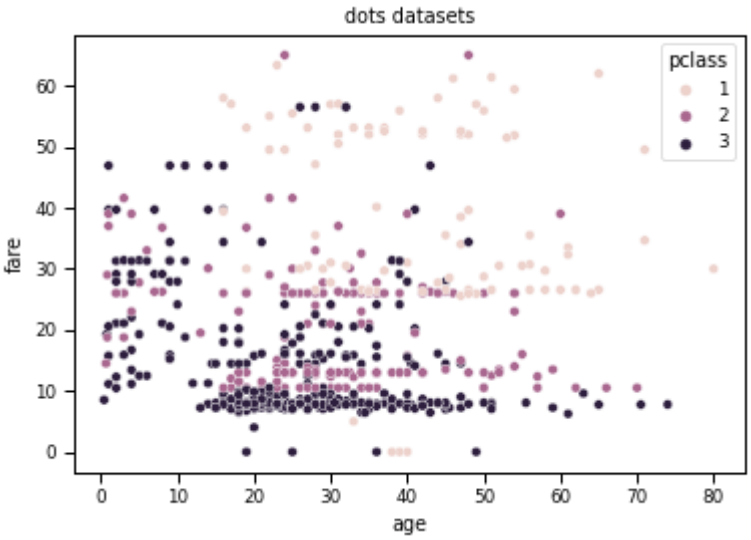
In [26]:

```
sns.scatterplot(x='embark_town', y='age',
                hue='class', data=df, )
plt.title('survived')
plt.show()
```



In [27]:

```
sns.scatterplot(x='age', y='fare',
                hue='pclass', data=df, )
plt.title('dots datasets')
plt.show()
```



Pair Plot - Multivariate Analysis

In [30]:

```
#iris = sns.load_dataset('iris')
sns.set_style("ticks")
sns.pairplot(df,hue = 'pclass',diag_kind = "kde",kind = "scatter",palette = "husl")
plt.show()
```

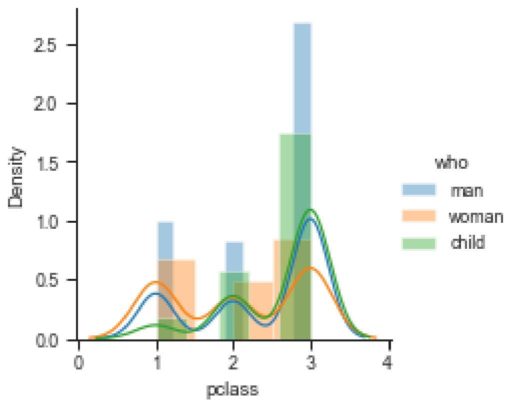
```
---------------------------------------------------------------------------
-
TypeError                                 Traceback (most recent call las
t)
<ipython-input-30-c1dc520fb13d> in <module>
      1 #iris = sns.load_dataset('iris')
      2 sns.set_style("ticks")
----> 3 sns.pairplot(df,hue = 'pclass',diag_kind = "kde",kind = "scatter",
palette = "husl")
      4 plt.show()

~\anaconda3\lib\site-packages\seaborn\_decorators.py in inner_f(*args, **k
wargs)
     44             )
     45         kwargs.update({k: arg for k, arg in zip(sig.parameters, ar
gs)})
---> 46         return f(**kwargs)
     47     return inner_f
     48
```

Histograms with Distplot Plot

In [31]:

```
plot = sns.FacetGrid(titanic , hue="who")
plot.map(sns.distplot, "pclass").add_legend()


plt.show()
```

```
C:\Users\Admin\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
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)
C:\Users\Admin\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
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)
C:\Users\Admin\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
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)
```



Handling Correlation

In [32]:

```
corr=titanic.corr()
corr
```

Out[32]:

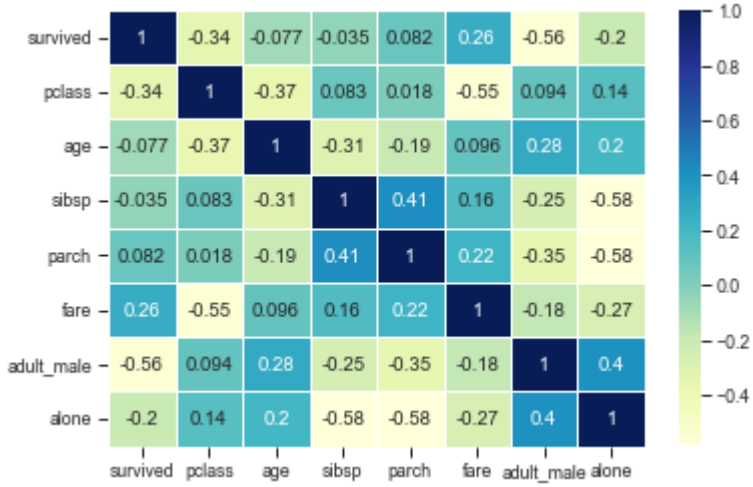
	survived	pclass	age	sibsp	parch	fare	adult_male	alo
survived	1.000000	-0.338481	-0.077221	-0.035322	0.081629	0.257307	-0.557080	-0.2033
pclass	-0.338481	1.000000	-0.369226	0.083081	0.018443	-0.549500	0.094035	0.1352
age	-0.077221	-0.369226	1.000000	-0.308247	-0.189119	0.096067	0.280328	0.1982
sibsp	-0.035322	0.083081	-0.308247	1.000000	0.414838	0.159651	-0.253586	-0.5844
parch	0.081629	0.018443	-0.189119	0.414838	1.000000	0.216225	-0.349943	-0.5833
fare	0.257307	-0.549500	0.096067	0.159651	0.216225	1.000000	-0.182024	-0.2718
adult_male	-0.557080	0.094035	0.280328	-0.253586	-0.349943	-0.182024	1.000000	0.4047
alone	-0.203367	0.135207	0.198270	-0.584471	-0.583398	-0.271832	0.404744	1.0000

In [33]:

```
sns.heatmap(corr,annot=True,linewidths=.5,cmap="YlGnBu")
```

Out[33]:

<AxesSubplot:>



In [34]:

```
#df = sns.load_dataset('titanic')
#ns.set_style("ticks")
#sns.pairplot(df,hue = 'class',diag_kind = "kde",kind = "scatter",palette = "husl")
#plt.show()
```

In [ ]:

```

```