In [2]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [5]:
```
#Importing the dataset using pandas read_csv
df= pd.read_csv('train.csv')
df.head()
```

Out[5]:

	PassengerId	Survived	Pclass	Name	Gender	Age	SibSp	Parch	Ticket	Fare
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

Data Cleaning

(Dropping column with null values, statistical analysis)

From the table, we can see that mean of the survived column is 0.38, but since this is not a complete dataset we cannot conclude on that.

The count for the ‘Age’ column is 714, which means the dataset has some missing values. We will have to clean up the data before start exploring.

In [6]:
```
df.describe()
```

Out[6]:

	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 [7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object 
 4   Gender       891 non-null    object 
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

In [8]:
```
# Check number of null values in a column
df.isnull().sum()
```

Out[8]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Gender           0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

In [9]:
```
#dropping column not in use and having maximum number of null values i.e. C
df_cleaned = df.drop(['PassengerId', 'Name', 'Ticket', 'Cabin'], axis=1)
df_cleaned.head()
```

Out[9]:

	Survived	Pclass	Gender	Age	SibSp	Parch	Fare	Embarked
0	0	3	male	22.0	1	0	7.2500	S
1	1	1	female	38.0	1	0	71.2833	C
2	1	3	female	26.0	0	0	7.9250	S
3	1	1	female	35.0	1	0	53.1000	S
4	0	3	male	35.0	0	0	8.0500	S

In [10]:
```
df_cleaned.describe()
```

Out[10]:

	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

In [11]:
```
df_cleaned.isnull().sum()
```

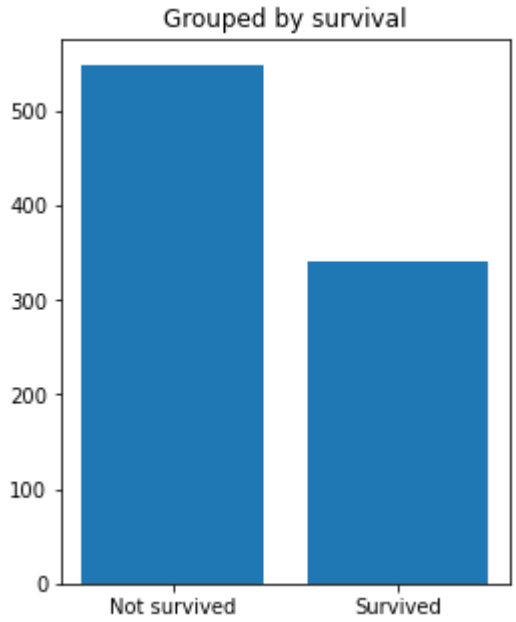
Out[11]:
```
Survived      0
Pclass        0
Gender        0
Age         177
SibSp         0
Parch         0
Fare          0
Embarked      2
dtype: int64
```

In [ ]:
```

```

In [12]:
```
# Group the data frame by values in Survived column, and count the number o
survived_count = df.groupby('Survived')['Survived'].count()
survived_count
```

Out[12]:
```
Survived
0    549
1    342
Name: Survived, dtype: int64
```

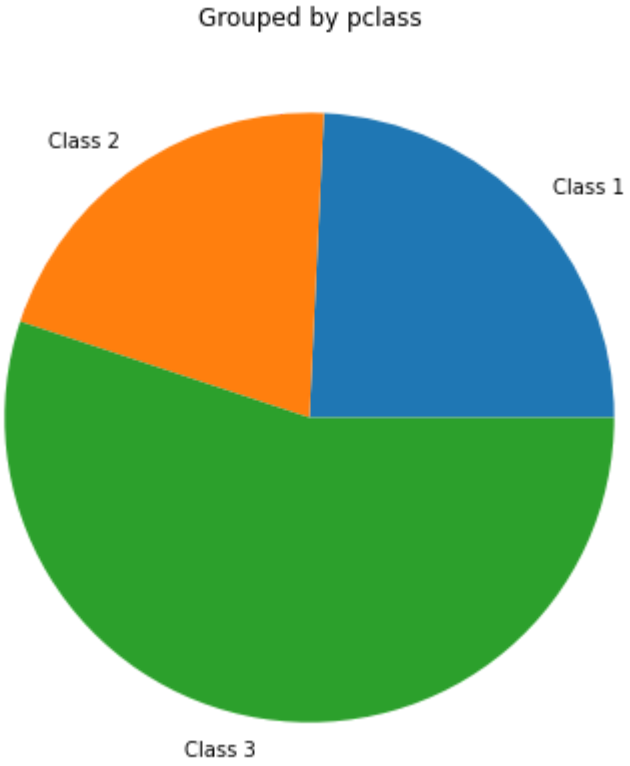
In [16]:
```
# Grouped by survival
plt.figure(figsize=(4,5))
plt.bar(survived_count.index, survived_count.values)
plt.title('Grouped by survival')
plt.xticks([0,1],['Not survived', 'Survived'])


plt.show()
```



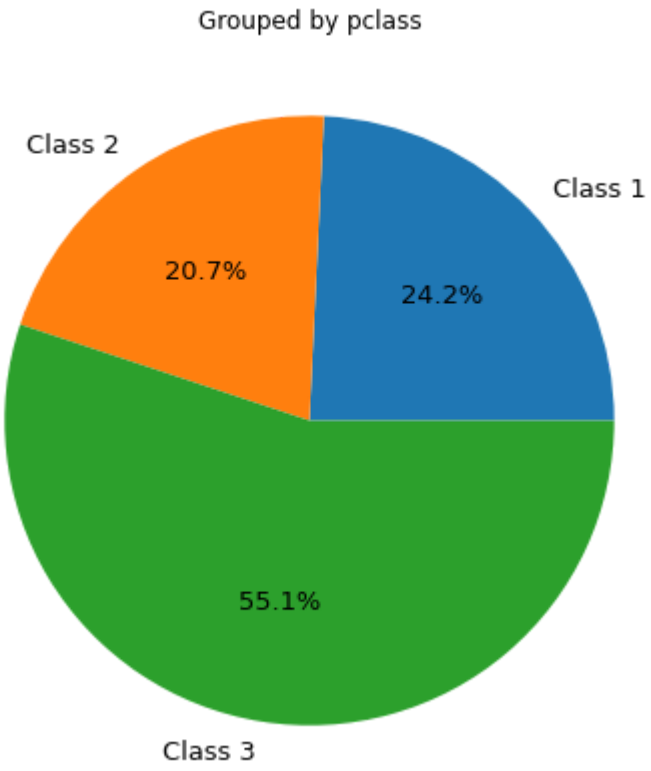
In [17]:
```
# Group the data frame by classes in the pclass column, and count the numbe
pclass_count = df.groupby('Pclass')['Pclass'].count()
pclass_count
```

Out[17]:
```
Pclass
1    216
2    184
3    491
Name: Pclass, dtype: int64
```

In [20]:
```
plt.figure(figsize=(7,7))
plt.title('Grouped by pclass')
plt.pie(pclass_count.values, labels=['Class 1', 'Class 2', 'Class 3'])
plt.show()
```



In [27]:
```
plt.figure(figsize=(7,7))
plt.title('Grouped by pclass')
plt.pie(pclass_count.values, labels=['Class 1', 'Class 2', 'Class 3'],
 autopct='%1.1f%%', textprops={'fontsize':13})
plt.show()
```

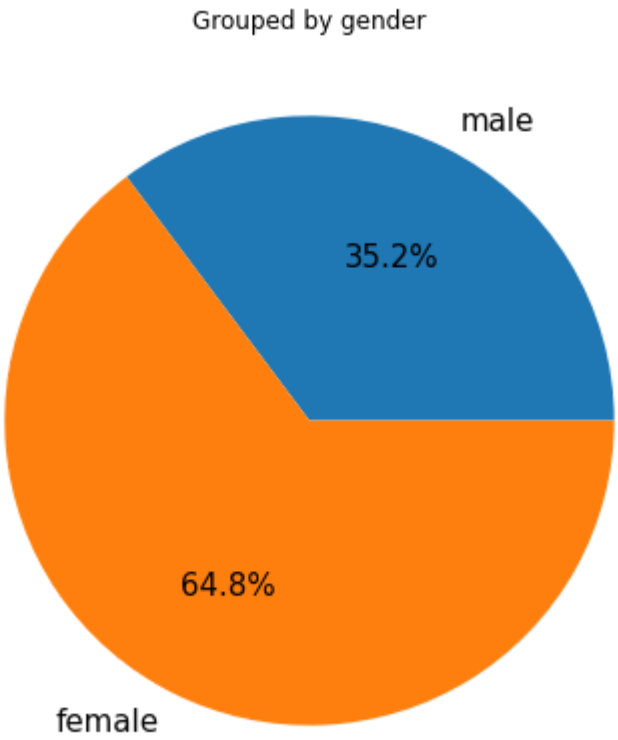


In [ ]:
```
```

In [30]:
```
# Group the data frame by classes in the pclass column, and count the numbe
gender_count = df.groupby('Gender')['Gender'].count()
gender_count
```

Out[30]:
```
Gender
female    314
male      577
Name: Gender, dtype: int64
```

In [34]:
```
plt.figure(figsize=(7,7))
plt.title('Grouped by gender')
plt.pie(gender_count.values, labels=['male', 'female'],
 autopct='%1.1f%%', textprops={'fontsize':15})
plt.show()
```

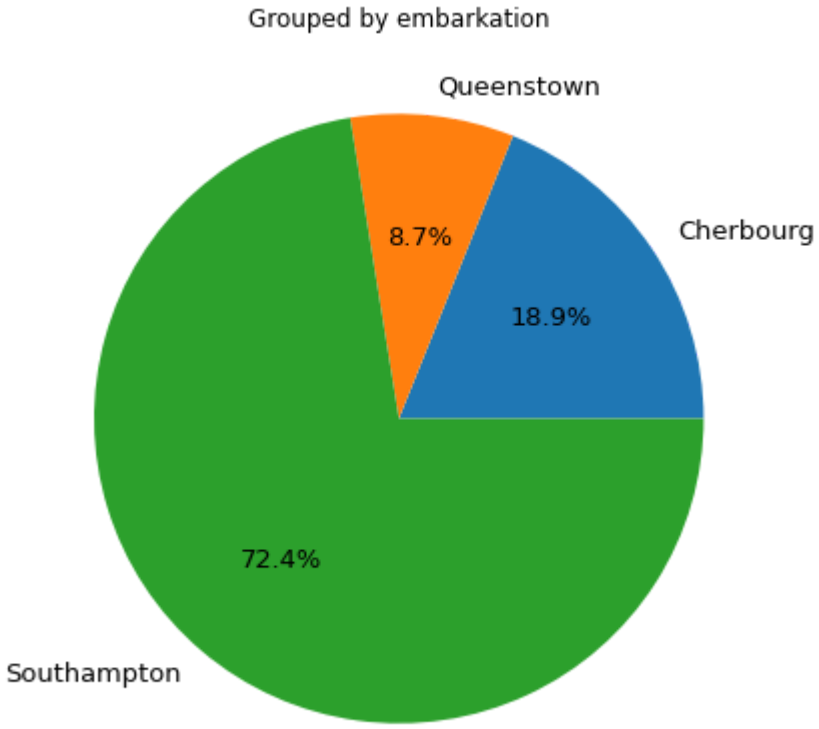


In [ ]:
```

```

In [35]:
```
# Group the data frame by classes in the pclass column, and count the numbe
embark_count = df.groupby('Embarked')['Embarked'].count()
embark_count
```

Out[35]:
```
Embarked
C    168
Q     77
S    644
Name: Embarked, dtype: int64
```

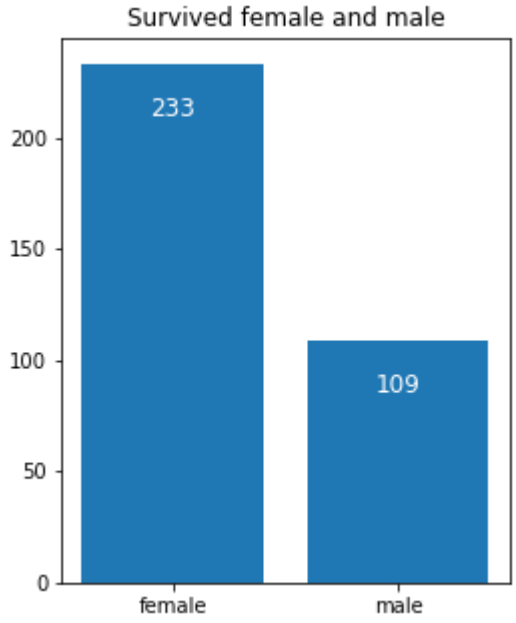
In [36]:
```
plt.figure(figsize=(7,7))
plt.title('Grouped by embarkation')
plt.pie(embark_count.values, labels=['Cherbourg', 'Queenstown', 'Southampto
        autopct='%1.1f%%', textprops={'fontsize':13})
plt.show()
```



visualize the following questions:

1. Did Gender play a role in Survival?
2. Did class played role in survival?
3. How does Embarkation vary across different ports?

In [43]:
```
#Survivial number according to gender or sex i.e. Male and Female
survived_gender = df.groupby('Gender')['Survived'].sum()
plt.figure(figsize=(4,5))
plt.bar(survived_gender.index, survived_gender.values)
plt.title('Survived female and male')

for i, value in enumerate(survived_gender.values):
    plt.text(i, value-20, str(value), fontsize=12, color='white',
             horizontalalignment='center', verticalalignment='center')
plt.show()
```



In [ ]:
```
```

In [44]:
```
grouped_by_pclass = df_cleaned.groupby(['Pclass', 'Survived', 'Gender'])
grouped_by_pclass.size()
```

Out[44]:
```
Pclass  Survived  Gender
1       0         female      3
                  male       77
        1         female     91
                  male       45
2       0         female      6
                  male       91
        1         female     70
                  male       17
3       0         female     72
                  male      300
        1         female     72
                  male       47
dtype: int64
```

In [45]:
```
df_cleaned.groupby(['Pclass'])['Survived'].sum()/df_cleaned.groupby(['Pclas
```

Out[45]:
```
Pclass
1    62.962963
2    47.282609
3    24.236253
Name: Survived, dtype: float64
```

In [49]:

```
# Define your data
# Assuming you have data similar to the 'df_cleaned' DataFrame

# Define the figure and subplots
fig, axes = plt.subplots(1, 3, figsize=(20, 8), sharey=True)

# Iterate through passenger classes
for i, pclass in enumerate([1, 2, 3]):
    # Filter data for the specific class
    class_data = df_cleaned[df_cleaned['Pclass'] == pclass]

    # Create a subplot for the current class
    ax = axes[i]

    # Create the counts for Survived and Sex
    counts = class_data.groupby(['Survived', 'Gender']).size().unstack()

    # Plot the counts
    counts.plot(kind='bar', stacked=True, ax=ax)

    # Set labels and title
    ax.set_title(f'Pclass {pclass}')
    ax.set_xlabel('Survived')
    ax.set_ylabel('Count')

# Adjust the layout
plt.subplots_adjust(top=0.9)
plt.suptitle('Class and gender wise segregation of passengers', fontsize=16

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