Before you turn this problem in, make sure everything runs as expected. First, **restart the kernel** (in the menu bar, select Kernel→Restart) and then **run all cells** (in the menu bar, select Cell→Run All).

Make sure that in addition to the code, you provide written answers for all questions of the assignment.

Below, please fill in your name and collaborators:

In [173]:
```
NAME = "Jacqueline Bungay"
COLLABORATORS = ""
```

Assignment 2 - Data Analysis using Pandas

(15 points total)

For this assignment, we will analyze the open dataset with data on the passengers aboard the Titanic.

The data file for this assignment can be downloaded from Kaggle website: https://www.kaggle.com/c/titanic/data (https://www.kaggle.com/c/titanic/data), file `train.csv` . It is also attached to the assignment page. The definition of all variables can be found on the same Kaggle page, in the Data Dictionary section.

Read the data from the file into pandas DataFrame. Analyze, clean and transform the data to answer the following question:

What categories of passengers were most likely to survive the Titanic disaster?

Question 1. *(4 points)*

- The answer to the main question - What categories of passengers were most likely to survive the Titanic disaster? *(2 points)*
- The detailed explanation of the logic of the analysis *(2 points)*

Question 2. *(3 points)*

- What other attributes did you use for the analysis? Explain how you used them and why you decided to use them.
- Provide a complete list of all attributes used.

Question 3. *(3 points)*

- Did you engineer any attributes (created new attributes)? If yes, explain the rationale and how the new attributes were used in the analysis?
- If you have excluded any attributes from the analysis, provide an explanation why you believe they can be excluded.

Question 4. *(5 points)*

- How did you treat missing values for those attributes that you included in the analysis (for example, `age` attribute)? Provide a detailed explanation in the comments

Information on Titanic Dataset

Kaggle Titanic Data Dictionary

Variable	Definition	Key	Notes
survival	Survival	0 = No, 1 = Yes	
pclass	Ticket class	1 = 1st, 2 = 2nd, 3 = 3rd	1st = Upper, 2nd = Middle, 3rd = Lower
sex	Sex	male, female	
Age	Age in years		Age is fractional if less than 1. If the age is estimated, is it in the form of xx.5
sibsp	# of siblings / spouses aboard the Titanic		Defines family relations in this way... Sibling = brother, sister, stepbrother, stepsister Spouse = husband, wife (mistresses and fiancés were ignored)
parch	# of parents / children aboard the Titanic		Dataset defines family relations in this way... Parent = mother, father Child = daughter, son, stepdaughter, stepson Some children travelled only with a nanny, therefore parch=0 for them
ticket	Ticket number		
fare	Passenger fare		
cabin	Cabin number		
embarked	Port of Embarkation	C = Cherbourg, Q = Queenstown, S = Southampton	Cherbourg, Franch Queenstown, Ireland Southampton, England

Results and Analysis Summary

After analysis of summary descriptive statistics of the Titanic dataset. Categories that were analysed to see which categories of passengers were most likely to survive the Titanic are:

- **Passenger title**
  - Out of all the passenger titles, the highest number that survived were those with title of 'Miss'.
  - The highest number that survived is 127
  - Top 4 titles are 'Miss', 'Mrs', 'Mr', 'Master'.
- **Ticket class**

    - The Upper class passengers had the highest number of survivers(136) with a survival rate of 63%.
    - Followed by Lower (119) and then Middle (87).

- **Age group**
    - The highest number of survivers by age group were adults (272). They were also the highest age group of passengers (748 out of 891 or 84% of passengers were adults).
    - However children had the highest survival rate (58%) while adults survival rate was (36%).

- **Passenger gender (Sex)**
    - The highest number of survivers by 'Sex' were females(233).
    - Females survival rate was 74% of all females compared to males which was only 19% of all males, even though they were 1.8 times more males than females.

- **Where passengers embarked**
    - The highest number of survivers embarked from Southhampton (219).
    - The total number of passengers embarked from Southampton (646) with a survival rate of 34%

- **--- The 4 categories with the highest number of passengers that survived ---**
    - Age Group of Adults (272) who were 84% of all passengers.
    - Females (233)
    - Passengers that embarked from Southampton (219)
    - Upper class passengers (136)

- **--- The categories with the highest survival rates ---**
    - Female (74%)
    - Upper Class (63%)
    - Children (58%)

Initial Dataset analysis and cleaning: (See detailed comments in Detailed Analysis Performed section below)

- Update column headings to be more descriptive
- Change column values to be more descriptive for columns:
    - 'Survived' - Yes/No
    - 'Embarked' - Cherbourg/Queenstown/Southampton
    - 'Passenger_class' - Upper/Middle/Lower

- Attributes excluded after initial analysis were removed due to limited time available to do all the analysis. The 5 categories above are focused on analysing individual passenger survival and not relationships between the passengers. The excluded attributes can be analysed in future when time permits.

    - 'Sibling_Spouse'
    - 'Parent_Child'
    - 'Fare' - after analysis of passengers that did not pay a fare. Passenger class is deemed more significant.

- Engineered attributes
    - 'Title' - to analyse survival based on a passenger's title. (Example, Mr, Mrs., Miss, Master, Dr.)
    - 'Age_Group' - to analyse survival based on age group passenger belonged in.
    - 'Total_Passengers' - for age group, Sex, port embarked category analysis
    - 'Survival rate' - for age group, Sex, port embarkedcategory analysis
    - 'Overall_%_of_Passengers' - for age group category analysis

- Missing values - how they were dealt with as they are needed for category analysis.
    - The 'Age' column had 177 passenger ages that were missing. Estimated the missing age values by taking the average of existing age values in the same group/category and assign that age to the missing age value for that passenger.
        - Average age of passengers with title 'Master'. Young males who are 17 years old or younger.
        - Average age of passengers with title 'Miss'. Ususally young and unmarried females.
        - Average age of passengers with title 'Mrs'. Married females.
        - Average age of passengers with title 'Mr'. Adult men who are 18 years or older.
    - The 'Port_of_Embarkation', 2 upper class passengers had missing values. Assigned 'Southampton' to both as they onboareded with with the same ticket number and shared the same cabin. Most passengers onboarded from Southampton

Other Observations

During initial analysis:

- Each row of the dataset represents information on a passenger and the dataset contains information on 891 passengers.
- Overall 38% of passengers survived.
- Cabin locations
    - There are 204 rows of 'Cabin_number' numbers which means 687 rows has missing cabin number data.
    - There are 147 (does not include NaN) unique cabin numbers. We know there are 204 non-null cabin numbers which means 57 cabin numbers are repeated in the dataset because some passengers (eg. a family) shared the same cabin.
    - 94% of all missing cabin numbers are in the 'Middle' and 'Lower' classes.
    - 77% of passengers cabin information is missing in the overall dataset.
    - Due to time constraints a clear approach was not determined in time in order to do analysis by cabin number/location.
- 65% percentage of all passengers were male but only 19% of all males survived.
- Passengers that did not pay a fare
    - There were 15 passengers that did not pay a 'Fare' and were most likely crew.

- They were all males that did not have any family members onboard across the 3 classes.
- They all boarded from Southhampton and only 1 survived.

Initializations and Custom functions

In [174]:
```
import numpy as np
import pandas as pd

Total_Passengers = 0
highest_that_survived = 0     #index to Series or DataFrame to the highest survival value.

upper_class  = 1
middle_class = 2
lower_class  = 3

#-------------------------------------------------------------------------------------------
def city_onboard_from (x):
    ''' This function will return the port city name where passengers boarded the Titanic
        based on the single character letter passed.

        Input:
            x: Single character value of 'C' or 'Q' or 'S'.

        Output:
            a string: City name of 'Cherbourg' or 'Queenstown' or 'Southampton' or NaN
    '''

    if x =='C':
        return 'Cherbourg'
    elif x == 'Q':
        return 'Queenstown'
    elif x == 'S':
        return 'Southampton'
    else:
        return np.nan
#-------------------------------------------------------------------------------------------
def YesNo_Survived (x):
    ''' This function will update 'Survived'column values to be more descriptive .

        Input:
            x: '0' or '1'

        Output:
            a string: 'Yes' when x = '1'
                      'No'  when x = '0'
    '''
    if x == '1':
        return 'Yes'
    else:
        return 'No'
#-------------------------------------------------------------------------------------------
```

```
def UpperMiddleLow_class(x):
    ''' This function will update 'Passenger_class' column values to be more descriptive .

        Input:
            x: '1' or '2' or '3'

        Output:
            a string: 'Upper'  when x = '1'
                      'Middle' when x = '2'
                      'Lower'  when x = '3'
    '''
    if x == '1':
        return 'Upper'
    elif x == '2':
        return 'Middle'
    else:
        return 'Lower'

#-------------------------------------------------------------------------------------------
def get_passenger_title(passenger_name):
    ''' This function will return the passenger title from the passenger name in the dataset.
        Format:
            Passenger name : "lastname, title. firstname middlename".
         For married women : "lastname, title. firstname middlename(maiden name with 1st middle lastname)"

        Input:
            passenger_name: Passenger 'Name' from the dataset.

        Output:
            a string: The passenger title.  Such as 'Mr', 'Mrs', 'Miss'
    '''
    return passenger_name[(passenger_name.find(',')+2):(passenger_name.find('.'))]
#-------------------------------------------------------------------------------------------
def age_group_category(age):
    ''' This function will return the age group category of the passenger based on the age parameter.

        Input:
            age: The age of the passenger. Float number with 2 decimal places.

        Output:
            a string: The age group category
                      age less than 13        'Child'
                      age between 13 and 17.x 'Teen'
                      age between 18 and 59.x 'Adult'
                      age greater than 60     'Senior'
    '''
    if age < 13:
```

```
        return 'Child'
    elif age >= 13 and age < 18:
        return 'Teen'
    elif age >= 18 and age < 60:
        return 'Adult'
    else:
        return 'Senior'
#-------------------------------------------------------------------------------------
```

In [175]:
```
# Load Titanic train dataset
Titanic_passengers = pd.read_csv("train.csv")
```

In [ ]:
```

```

Detailed Analysis Performed

Analysis of what dataset looks like and potential categories to analyse

In [176]:
```
# Each row of the dataset represents a passenger that was onboard the Titanic.

# The headings - some headings can be more descriptive -
#              - 'Pclass' change to 'Passenger_class',
#              - 'SibSp'  change to 'Sibling_Spouse',
#              - 'Parch'  change to 'Parent_Child',
#              - 'Cabin'  change to 'Cabin_number',
#              - 'Embarked' change to 'Port_of_Embarkation'

# Data shown - 'Survived' could be changed to Yes/No instead of 1/0.
#            - 'Embarked' could be changed to 'Port_of_Embarkation' and values could be changed to
#                - 'Cherbourg' when value = 'C',
#                - 'Queenstown'when value = 'Q',
#                - 'Southampton' when value = 'S'
#            - 'Cabin' seems mostly 1st class has cabin numbers - verify this.
#            - 'Age' has some missing values.
#            - 'Name' format is "lastname, title. firstname middlename".
#                     for married women the format is
#            -           "lastname, title. firstname middlename(maiden name - 1st mid last)"

Titanic_passengers.rename(columns={'Pclass':'Passenger_class','SibSp':'Sibling_Spouse','Parch':'Parent_Child',
                                   'Cabin':'Cabin_number', 'Embarked':'Port_of_Embarkation'}, inplace=True)
Titanic_passengers.head(3)
```

Out[176]:

	PassengerId	Survived	Passenger_class	Name	Sex	Age	Sibling_Spouse	Parent_Child	Ticket	Fare	Cabin_number	Port_of_Embarkati
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	

In [177]:
```
# Change 'Port_of_Embarkation' from single letter 'C', 'Q' or 'S' to name of the city that passengers embarked f
Titanic_passengers['Port_of_Embarkation'] = Titanic_passengers['Port_of_Embarkation'].apply(city_onboard_from)
Titanic_passengers.head(3)
```

Out[177]:

	PassengerId	Survived	Passenger_class	Name	Sex	Age	Sibling_Spouse	Parent_Child	Ticket	Fare	Cabin_number	Port_of_Embarkati
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	Southampt
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	Cherbou
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	Southampt

In [178]:
```
Titanic_passengers.tail()
```

Out[178]:

	PassengerId	Survived	Passenger_class	Name	Sex	Age	Sibling_Spouse	Parent_Child	Ticket	Fare	Cabin_number	Port_of_Embarkation
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00	NaN	Southampton
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00	B42	Southampton
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45	NaN	Southampton
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00	C148	Cherbourg
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75	NaN	Queenstown

In [179]:
```
# Analysing passenger names format in the dataset.
#-----------------------------------------------
#  'Name' format is "lastname, title. firstname middlename".
#         for married women the format is
#          -  "lastname, title. firstname middlename(maiden name - 1st middle last)"

Titanic_passengers['Name'].head(10)
```

Out[179]:
```
0                              Braund, Mr. Owen Harris
1    Cumings, Mrs. John Bradley (Florence Briggs Th...
2                               Heikkinen, Miss. Laina
3         Futrelle, Mrs. Jacques Heath (Lily May Peel)
4                             Allen, Mr. William Henry
5                                     Moran, Mr. James
6                              McCarthy, Mr. Timothy J
7                       Palsson, Master. Gosta Leonard
8    Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)
9                  Nasser, Mrs. Nicholas (Adele Achem)
Name: Name, dtype: object
```

In [180]:
```
# DataFrame information
#-----------------------
# There are 891 entries of passenger information
#
# There are 12 columns
# Each column contains 891 non-null data except for
#     'Age'                 which has 714 entries
#     'Cabin_number'        which has 204 entries
#     'Port_of_Embarkation' which has 889 entries
#
# Column Data types
#        integer - PassengerId, Survived, Passenger_class, Sibling_Spouse, Parent_Child
#  string object - Name, Sex, Ticket, Cabin_number, Port_of_Embarkation
#          float - Age, Fare

Titanic_passengers.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   Passenger_class      891 non-null    int64  
 3   Name                 891 non-null    object 
 4   Sex                  891 non-null    object 
 5   Age                  714 non-null    float64
 6   Sibling_Spouse       891 non-null    int64  
 7   Parent_Child         891 non-null    int64  
 8   Ticket               891 non-null    object 
 9   Fare                 891 non-null    float64
 10  Cabin_number         204 non-null    object 
 11  Port_of_Embarkation  889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

In [181]:
```
# Summary descriptive statistics of all columns - See analysis below in this section
#-------------------------------------------

# -- Count --
# There are 891 rows of Titanic passenger information.
# There are 714 rows of passenger 'Age' information which means 177 rows has missing age data.
# There are 204 rows of 'Cabin_number' numbers which means 687 rows has missing cabin number data.
# There are 889 rows of 'Port_of_Embarkation' data which means 2 rows has missing port of embarkation data.

# -- Unique --
# There are 891 unique passenger names in the dataset. So, each row represents a passenger.
# There are 681 unique 'Ticket' numbers which means 210 are repeat ticket numbers. This could mean
#    groups of passengers (example, a family) boarded under the same ticket number. To be verified.

# There are 147 (does not include NaN) unique cabin numbers.  We know there are 204 non-null cabin numbers
# which means 57 cabin numbers are repeated in the dataset because some passengers (eg. a family)
#    shared the same cabin.
#
# --- Top & Frequency ---
# The most common 'Sex' in the dataset is 'male', where 577 of the 891 passengers were male (65%).
# The most common 'Ticket' number is '347082' which 7 passengers had, meaning they were in the same group/family
# The most common 'Cabin' number is 'B96 B98' which appears to be 2 cabins shared by 4 passengers.
# The most common 'Port_of_Embarkation' was Southampton where 644 passengers boarded the Titanic.
#
# --- Mean ---
# The mean of the 'Survived' column is actually the percentage of passengers that survived (38%).
#    Since the 'Survived' column only has 1s and 0s, only the 1s (passenger survived) were summed and
#    then divided by the total number of passengers.

#--- min ---
# Some passengers did not pay a 'Fare'. They were most likely crew.


descript_stats = Titanic_passengers.describe(include = 'all')

descript_stats
```

Out[181]:

	PassengerId	Survived	Passenger_class	Name	Sex	Age	Sibling_Spouse	Parent_Child	Ticket	Fare	Cabin_number	Port_of
count	891.000000	891.000000	891.000000	891	891	714.000000	891.000000	891.000000	891	891.000000	204	
unique	NaN	NaN	NaN	891	2	NaN	NaN	NaN	681	NaN	147	
top	NaN	NaN	NaN	Braund, Mr. Owen Harris	male	NaN	NaN	NaN	347082	NaN	B96 B98	

	PassengerId	Survived	Passenger_class	Name	Sex	Age	Sibling_Spouse	Parent_Child	Ticket	Fare	Cabin_number	Port_of
freq	NaN	NaN	NaN	1	577	NaN	NaN	NaN	7	NaN	4	
mean	446.000000	0.383838	2.308642	NaN	NaN	29.699118	0.523008	0.381594	NaN	32.204208	NaN	
std	257.353842	0.486592	0.836071	NaN	NaN	14.526497	1.102743	0.806057	NaN	49.693429	NaN	
min	1.000000	0.000000	1.000000	NaN	NaN	0.420000	0.000000	0.000000	NaN	0.000000	NaN	
25%	223.500000	0.000000	2.000000	NaN	NaN	20.125000	0.000000	0.000000	NaN	7.910400	NaN	
50%	446.000000	0.000000	3.000000	NaN	NaN	28.000000	0.000000	0.000000	NaN	14.454200	NaN	
75%	668.500000	1.000000	3.000000	NaN	NaN	38.000000	1.000000	0.000000	NaN	31.000000	NaN	

In [182]:
```
# Total number of passengers in the dataset.
Total_Passengers = len(Titanic_passengers)
Total_Passengers
```

Out[182]: 891

In [183]:
```
# The 'Survived' column in the Titanic dataset contains non-null values for each passenger and is an integer dty
# Confirm that it only has 1 or 0 and no other integer values.
Titanic_passengers['Survived'].unique()
```

Out[183]: array([0, 1])

In [184]:
```
# Passenger age information that are missing (177).
Titanic_passengers[Titanic_passengers['Age'].isna()]
```

Out[184]:

	PassengerId	Survived	Passenger_class	Name	Sex	Age	Sibling_Spouse	Parent_Child	Ticket	Fare	Cabin_number	Port_of_Embarka
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Queens
17	18	1	2	Williams, Mr. Charles Eugene	male	NaN	0	0	244373	13.0000	NaN	Southam
19	20	1	3	Masselmani, Mrs. Fatima	female	NaN	0	0	2649	7.2250	NaN	Cherb
26	27	0	3	Emir, Mr. Farred Chehab	male	NaN	0	0	2631	7.2250	NaN	Cherb
28	29	1	3	O'Dwyer, Miss. Ellen "Nellie"	female	NaN	0	0	330959	7.8792	NaN	Queens
...	...	...	...	...	...	...	...	...	...	...	...	
859	860	0	3	Razi, Mr. Raihed	male	NaN	0	0	2629	7.2292	NaN	Cherb
863	864	0	3	Sage, Miss. Dorothy Edith "Dolly"	female	NaN	8	2	CA. 2343	69.5500	NaN	Southam
868	869	0	3	van Melkebeke, Mr. Philemon	male	NaN	0	0	345777	9.5000	NaN	Southam
878	879	0	3	Laleff, Mr. Kristo	male	NaN	0	0	349217	7.8958	NaN	Southam
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	Southam

177 rows × 12 columns

Passenger cabin numbers that are missing and classes

In [185]:
```
# All the unique cabin numbers + NaN
unique_cabins = Titanic_passengers['Cabin_number'].unique()

unique_cabins
```

Out[185]:
```
array([nan, 'C85', 'C123', 'E46', 'G6', 'C103', 'D56', 'A6',
       'C23 C25 C27', 'B78', 'D33', 'B30', 'C52', 'B28', 'C83', 'F33',
       'F G73', 'E31', 'A5', 'D10 D12', 'D26', 'C110', 'B58 B60', 'E101',
       'F E69', 'D47', 'B86', 'F2', 'C2', 'E33', 'B19', 'A7', 'C49', 'F4',
       'A32', 'B4', 'B80', 'A31', 'D36', 'D15', 'C93', 'C78', 'D35',
       'C87', 'B77', 'E67', 'B94', 'C125', 'C99', 'C118', 'D7', 'A19',
       'B49', 'D', 'C22 C26', 'C106', 'C65', 'E36', 'C54',
       'B57 B59 B63 B66', 'C7', 'E34', 'C32', 'B18', 'C124', 'C91', 'E40',
       'T', 'C128', 'D37', 'B35', 'E50', 'C82', 'B96 B98', 'E10', 'E44',
       'A34', 'C104', 'C111', 'C92', 'E38', 'D21', 'E12', 'E63', 'A14',
       'B37', 'C30', 'D20', 'B79', 'E25', 'D46', 'B73', 'C95', 'B38',
       'B39', 'B22', 'C86', 'C70', 'A16', 'C101', 'C68', 'A10', 'E68',
       'B41', 'A20', 'D19', 'D50', 'D9', 'A23', 'B50', 'A26', 'D48',
       'E58', 'C126', 'B71', 'B51 B53 B55', 'D49', 'B5', 'B20', 'F G63',
       'C62 C64', 'E24', 'C90', 'C45', 'E8', 'B101', 'D45', 'C46', 'D30',
       'E121', 'D11', 'E77', 'F38', 'B3', 'D6', 'B82 B84', 'D17', 'A36',
       'B102', 'B69', 'E49', 'C47', 'D28', 'E17', 'A24', 'C50', 'B42',
       'C148'], dtype=object)
```

In [186]:
```
# Number of unique cabins including NaN
unique_cabins.shape
```

Out[186]:
```
(148,)
```

In [187]:
```
# The total number of 1st, 2nd, and 3rd class passengers
Titanic_passengers.groupby('Passenger_class')['Name'].count().to_frame()
```

Out[187]:

Passenger_class	Name
1	216
2	184
3	491

In [188]:
```
# Which class had the highest number of survivers?
# Answer: The 1st class.
Titanic_passengers.groupby('Passenger_class')['Survived'].sum().to_frame()
```

Out[188]:

Passenger_class	Survived
1	136
2	87
3	119

In [189]:
```
# The number of missing 'Cabin' numbers for 1st, 2nd and 3rd class tickets.

no_cabin_num_by_class = Titanic_passengers[Titanic_passengers['Cabin_number'].isna()].groupby('Passenger_class')

no_cabin_num_by_class
```

Out[189]:

Passenger_class	Name	Cabin_number
1	40	0
2	168	0
3	479	0

In [190]:
```
# 94% of all missing cabin numbers are in the 'Middle' and 'Lower' classes.
total_missing_cabin_nums = no_cabin_num_by_class['Name'].sum()
round((no_cabin_num_by_class['Name'][middle_class] + no_cabin_num_by_class['Name'][lower_class]) / total_missing
```

Out[190]: 0.94

In [191]:
```
# 77% of passengers cabin information is missing in the overall dataset. Due to time constraints a clear approac
# was not determined in time in order to do analysis by cabin number/location.
round(total_missing_cabin_nums / descript_stats['Name']['count'],2)
```

Out[191]: 0.77

Other analysis - % males, family of 7, overall % passengers that survived

In [192]:
```
# Percentage of passengers that are males (65%).
round(descript_stats.loc['freq','Sex'] / descript_stats.loc['count','Sex'],2)
```

Out[192]:
```
0.65
```

In [193]:
```
# Analysis of group/family of 7 passengers with ticket number '347082'.
#   - 3rd/lower class ticket.
#   - Family of 7 onboarded at Southampton
#   - Mr. & Mrs. Anders Johan Andersson and their 4 daughters and 1 son.
#   - None survived
Titanic_passengers[Titanic_passengers['Ticket'] == '347082']
```

Out[193]:

	PassengerId	Survived	Passenger_class	Name	Sex	Age	Sibling_Spouse	Parent_Child	Ticket	Fare	Cabin_number	Port_of_Embarkatic
13	14	0	3	Andersson, Mr. Anders Johan	male	39.0	1	5	347082	31.275	NaN	Southampto
119	120	0	3	Andersson, Miss. Ellis Anna Maria	female	2.0	4	2	347082	31.275	NaN	Southampto
541	542	0	3	Andersson, Miss. Ingeborg Constanzia	female	9.0	4	2	347082	31.275	NaN	Southampto
542	543	0	3	Andersson, Miss. Sigrid Elisabeth	female	11.0	4	2	347082	31.275	NaN	Southampto
610	611	0	3	Andersson, Mrs. Anders Johan (Alfrida Konstant...	female	39.0	1	5	347082	31.275	NaN	Southampto
813	814	0	3	Andersson, Miss. Ebba Iris Alfrida	female	6.0	4	2	347082	31.275	NaN	Southampto
850	851	0	3	Andersson, Master. Sigvard Harald Elias	male	4.0	4	2	347082	31.275	NaN	Southampto

In [194]:
```
# Percentage of passengers that survived. This is also the mean in the descriptive statistics because the 'Survi
# column only has 1s and 0s for survived and did not survive.  Since the mean is to add all the values in the co
# divided by the number of values, it is also the percentage of passengers that survived (38%)
round(Titanic_passengers['Survived'].sum() / Titanic_passengers['Name'].count(),2)
```

Out[194]: 0.38

Data transform - Update columns 'Survived' and 'Passenger_class' values to be more descriptive

In [195]:
```
# Update columns 'Survived' and 'Passenger_class' values to be more descriptive

# First change the column data types from int64 to string.
Titanic_passengers['Survived'] = Titanic_passengers['Survived'].astype('string')
Titanic_passengers['Passenger_class'] = Titanic_passengers['Passenger_class'].astype('string')

# Change'Survived' to 'No'  when '0'
#                    'Yes' when '1'
Titanic_passengers['Survived'] = Titanic_passengers['Survived'].apply(YesNo_Survived)

Titanic_passengers.head()
```

Out[195]:

	PassengerId	Survived	Passenger_class	Name	Sex	Age	Sibling_Spouse	Parent_Child	Ticket	Fare	Cabin_number	Port_of_Embarkati
0	1	No	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	Southampt
1	2	Yes	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	Cherbou
2	3	Yes	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	Southampt
3	4	Yes	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	Southampt
4	5	No	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	Southampt

In [196]:
```
# Change'Passenger_class' to 'Upper'  when '1'
#                           'Middle' when '2'
#                           'Lower'  when '3'

Titanic_passengers['Passenger_class'] = Titanic_passengers['Passenger_class'].apply(UpperMiddleLow_class)

Titanic_passengers.tail()
```

Out[196]:

	PassengerId	Survived	Passenger_class	Name	Sex	Age	Sibling_Spouse	Parent_Child	Ticket	Fare	Cabin_number	Port_of_Embarkation
886	887	No	Middle	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00	NaN	Southampton
887	888	Yes	Upper	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00	B42	Southampton
888	889	No	Lower	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45	NaN	Southampton
889	890	Yes	Upper	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00	C148	Cherbourg
890	891	No	Lower	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75	NaN	Queenstown

Analysis - Passengers that did not pay fares

In [197]:
```
# There were 15 passengers that did not pay a 'Fare' and were most likely crew.
# They were all males that did not have any family members onboard across the 3 classes.
# They all boarded from Southhampton and only 1 survived.
Titanic_passengers[Titanic_passengers['Fare'] == 0.00]
```

Out[197]:

	PassengerId	Survived	Passenger_class	Name	Sex	Age	Sibling_Spouse	Parent_Child	Ticket	Fare	Cabin_number	Port_of_Embarkation
179	180	No	Lower	Leonard, Mr. Lionel	male	36.0	0	0	LINE	0.0	NaN	Southampton
263	264	No	Upper	Harrison, Mr. William	male	40.0	0	0	112059	0.0	B94	Southampton
271	272	Yes	Lower	Tornquist, Mr. William Henry	male	25.0	0	0	LINE	0.0	NaN	Southampton
277	278	No	Middle	Parkes, Mr. Francis "Frank"	male	NaN	0	0	239853	0.0	NaN	Southampton
302	303	No	Lower	Johnson, Mr. William Cahoone Jr	male	19.0	0	0	LINE	0.0	NaN	Southampton
413	414	No	Middle	Cunningham, Mr. Alfred Fleming	male	NaN	0	0	239853	0.0	NaN	Southampton
466	467	No	Middle	Campbell, Mr. William	male	NaN	0	0	239853	0.0	NaN	Southampton
481	482	No	Middle	Frost, Mr. Anthony Wood "Archie"	male	NaN	0	0	239854	0.0	NaN	Southampton
597	598	No	Lower	Johnson, Mr. Alfred	male	49.0	0	0	LINE	0.0	NaN	Southampton
633	634	No	Upper	Parr, Mr. William Henry Marsh	male	NaN	0	0	112052	0.0	NaN	Southampton
674	675	No	Middle	Watson, Mr. Ennis Hastings	male	NaN	0	0	239856	0.0	NaN	Southampton
732	733	No	Middle	Knight, Mr. Robert J	male	NaN	0	0	239855	0.0	NaN	Southampton
806	807	No	Upper	Andrews, Mr. Thomas Jr	male	39.0	0	0	112050	0.0	A36	Southampton

	PassengerId	Survived	Passenger_class	Name	Sex	Age	Sibling_Spouse	Parent_Child	Ticket	Fare	Cabin_number	Port_of_Embarkation
815	816	No	Upper	Fry, Mr. Richard	male	NaN	0	0	112058	0.0	B102	Southampton

Reuchlin,

In [198]:
```
# Number of passengers that did not pay a 'Fare'.
Titanic_passengers[Titanic_passengers['Fare'] == 0.00]['PassengerId'].count()
```

Out[198]: 15

In [199]:
```
# Remove the columns 'Sibling_Spouse', 'Parent_Child', and 'Fare' as they will not be part
# of analysis going forward.
Titanic_passengers.drop(['Sibling_Spouse','Parent_Child','Fare'], axis=1, inplace=True)

Titanic_passengers.head(2)
```

Out[199]:

	PassengerId	Survived	Passenger_class	Name	Sex	Age	Ticket	Cabin_number	Port_of_Embarkation
0	1	No	Lower	Braund, Mr. Owen Harris	male	22.0	A/5 21171	NaN	Southampton
1	2	Yes	Upper	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	PC 17599	C85	Cherbourg

Adding passenger 'Title' as a category to analyse

In [200]:
```
# Add passenger title column to the dataset
Titanic_passengers['Title'] = Titanic_passengers['Name'].apply(get_passenger_title)

Titanic_passengers
```

Out[200]:

	PassengerId	Survived	Passenger_class	Name	Sex	Age	Ticket	Cabin_number	Port_of_Embarkation	Title
0	1	No	Lower	Braund, Mr. Owen Harris	male	22.0	A/5 21171	NaN	Southampton	Mr
1	2	Yes	Upper	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	PC 17599	C85	Cherbourg	Mrs
2	3	Yes	Lower	Heikkinen, Miss. Laina	female	26.0	STON/O2. 3101282	NaN	Southampton	Miss
3	4	Yes	Upper	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	113803	C123	Southampton	Mrs
4	5	No	Lower	Allen, Mr. William Henry	male	35.0	373450	NaN	Southampton	Mr
...	...	...	...	...	...	...	...	...	...	...
886	887	No	Middle	Montvila, Rev. Juozas	male	27.0	211536	NaN	Southampton	Rev
887	888	Yes	Upper	Graham, Miss. Margaret Edith	female	19.0	112053	B42	Southampton	Miss
888	889	No	Lower	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	W./C. 6607	NaN	Southampton	Miss
889	890	Yes	Upper	Behr, Mr. Karl Howell	male	26.0	111369	C148	Cherbourg	Mr
890	891	No	Lower	Dooley, Mr. Patrick	male	32.0	370376	NaN	Queenstown	Mr

891 rows × 10 columns

Highest number of survivers by category 'Title' of the passenger

In [201]:
```
# What are the passenger titles and the number of passengers for each ?
Titanic_passengers.groupby('Title').count().sort_values('PassengerId', ascending=False)['PassengerId']
```

Out[201]:
```
Title
Mr              517
Miss            182
Mrs             125
Master           40
Dr                7
Rev               6
Major             2
Col               2
Mlle              2
Sir               1
Ms                1
Capt              1
Mme               1
Lady              1
Jonkheer          1
Don               1
the Countess      1
Name: PassengerId, dtype: int64
```

In [202]:
```
# Group by category of passenger title and count those who survived.
# Passengers with title of 'Miss' survived the most followed by 'Mrs'

titles_survive_info = Titanic_passengers[['Title','Survived']]

titles_survived = titles_survive_info[titles_survive_info['Survived'] == 'Yes'].groupby('Title').count().sort_va
                                                                                    ascending=False).head
titles_survived
```

Out[202]:

Title	Survived
Miss	127
Mrs	99
Mr	81
Master	23
Dr	3

In [203]:
```
titles_survived.reset_index()
```

Out[203]:

	Title	Survived
0	Miss	127
1	Mrs	99
2	Mr	81
3	Master	23
4	Dr	3

In [204]:
```
# Out of all the passenger titles, the highest number that survived were those with title of 'Miss'.
titles_survived.reset_index().loc[highest_that_survived]
```

Out[204]:
```
Title       Miss
Survived     127
Name: 0, dtype: object
```

In [205]:
```
# The highest number that survived is 127.
titles_survived.reset_index().loc[highest_that_survived,'Survived']
```

Out[205]: 127

Highest number of survivers by category 'Passenger_class'

In [206]:
```
# Which passenger ticket class had the highest number of survivers?
# The Upper class passengers had the highest number, followed by Lower and then Middle.
class_survive_info = Titanic_passengers[['Passenger_class','Survived']]
class_survived = class_survive_info[class_survive_info['Survived'] == 'Yes'].groupby('Passenger_class').count().
                                                                                    ascending=False)
class_survived
```

Out[206]:

	Survived
Passenger_class	
Upper	136
Lower	119
Middle	87

In [207]:
```
# The highest number that survived is 136.
class_survived.reset_index().loc[highest_that_survived,'Survived']
```

Out[207]: 136

In [208]:
```
# After submitted as was running out of time. What was the total number of Upper class tickets?
tot_by_class = class_survive_info.groupby('Passenger_class').count()
tot_by_class.rename(columns={'Survived':'Total_passengers'},inplace=True)
tot_by_class
```

Out[208]:

Passenger_class	Total_passengers
Lower	491
Middle	184
Upper	216

In [209]:
```
# Join into 1 table the passengers survived by 'class' with total passengers by 'class'
joined_tot_class = class_survived.join(tot_by_class)
joined_tot_class
```

Out[209]:

Passenger_class	Survived	Total_passengers
Upper	136	216
Lower	119	491
Middle	87	184

In [210]:
```
#  Add column'Survival rate' which contains survival rate of each 'Passenger_class'
# The highest number of survivers by 'Passenger_class' was 136.
# However, the survival rate for Upper class passengers was 63%
joined_tot_class['Survival rate'] = joined_tot_class['Survived'] / joined_tot_class['Total_passengers']

joined_tot_class
```

Out[210]:

Passenger_class	Survived	Total_passengers	Survival rate
Upper	136	216	0.629630
Lower	119	491	0.242363
Middle	87	184	0.472826

In [ ]:

Highest number of survivers by category 'Age_group'

Some **'Age'** values are missing. So, in order to analyse survival by 'Age_group', will estimate the missing age values by taking the average of existing age values in the same group/category and assign that age to the missing age value for that passenger.

Passengers with title **'Master'** usually refers to young males who are 17 years old or younger.
Estimate the missing age values by taking the average of existing age values in passengers with title 'Master' and assign that average age to the missing age value for that passenger.

In [211]:
```
# Passengers with title 'Master' usually refers to young males who are 17 years old or younger.
# Once they turn 18, they are considered to be an adult and are referred to as 'Mr.'

# Show passengers with title 'Master'
title_master = Titanic_passengers[Titanic_passengers['Title'] == 'Master']

title_master
```

Out[211]:

	PassengerId	Survived	Passenger_class	Name	Sex	Age	Ticket	Cabin_number	Port_of_Embarkation	Title
7	8	No	Lower	Palsson, Master. Gosta Leonard	male	2.00	349909	NaN	Southampton	Master
16	17	No	Lower	Rice, Master. Eugene	male	2.00	382652	NaN	Queenstown	Master
50	51	No	Lower	Panula, Master. Juha Niilo	male	7.00	3101295	NaN	Southampton	Master
59	60	No	Lower	Goodwin, Master. William Frederick	male	11.00	CA 2144	NaN	Southampton	Master
63	64	No	Lower	Skoog, Master. Harald	male	4.00	347088	NaN	Southampton	Master
65	66	Yes	Lower	Moubarek, Master. Gerios	male	NaN	2661	NaN	Cherbourg	Master
78	79	Yes	Middle	Caldwell, Master. Alden Gates	male	0.83	248738	NaN	Southampton	Master
125	126	Yes	Lower	Nicola-Yarred, Master. Elias	male	12.00	2651	NaN	Cherbourg	Master
159	160	No	Lower	Sage, Master. Thomas Henry	male	NaN	CA. 2343	NaN	Southampton	Master
164	165	No	Lower	Panula, Master. Eino Viljami	male	1.00	3101295	NaN	Southampton	Master
165	166	Yes	Lower	Goldsmith, Master. Frank John William "Frankie"	male	9.00	363291	NaN	Southampton	Master
171	172	No	Lower	Rice, Master. Arthur	male	4.00	382652	NaN	Queenstown	Master
176	177	No	Lower	Lefebre, Master. Henry Forbes	male	NaN	4133	NaN	Southampton	Master
182	183	No	Lower	Asplund, Master. Clarence Gustaf Hugo	male	9.00	347077	NaN	Southampton	Master
183	184	Yes	Middle	Becker, Master. Richard F	male	1.00	230136	F4	Southampton	Master
193	194	Yes	Middle	Navratil, Master. Michel M	male	3.00	230080	F2	Southampton	Master
261	262	Yes	Lower	Asplund, Master. Edvin Rojj Felix	male	3.00	347077	NaN	Southampton	Master
278	279	No	Lower	Rice, Master. Eric	male	7.00	382652	NaN	Queenstown	Master
305	306	Yes	Upper	Allison, Master. Hudson Trevor	male	0.92	113781	C22 C26	Southampton	Master
340	341	Yes	Middle	Navratil, Master. Edmond Roger	male	2.00	230080	F2	Southampton	Master
348	349	Yes	Lower	Coutts, Master. William Loch "William"	male	3.00	C.A. 37671	NaN	Southampton	Master

	PassengerId	Survived	Passenger_class	Name	Sex	Age	Ticket	Cabin_number	Port_of_Embarkation	Title
386	387	No	Lower	Goodwin, Master. Sidney Leonard	male	1.00	CA 2144	NaN	Southampton	Master
407	408	Yes	Middle	Richards, Master. William Rowe	male	3.00	29106	NaN	Southampton	Master
445	446	Yes	Upper	Dodge, Master. Washington	male	4.00	33638	A34	Southampton	Master
480	481	No	Lower	Goodwin, Master. Harold Victor	male	9.00	CA 2144	NaN	Southampton	Master
489	490	Yes	Lower	Coutts, Master. Eden Leslie "Neville"	male	9.00	C.A. 37671	NaN	Southampton	Master
549	550	Yes	Middle	Davies, Master. John Morgan Jr	male	8.00	C.A. 33112	NaN	Southampton	Master
709	710	Yes	Lower	Moubarek, Master. Halim Gonios ("William George")	male	NaN	2661	NaN	Cherbourg	Master
751	752	Yes	Lower	Moor, Master. Meier	male	6.00	392096	E121	Southampton	Master
755	756	Yes	Middle	Hamalainen, Master. Viljo	male	0.67	250649	NaN	Southampton	Master
787	788	No	Lower	Rice, Master. George Hugh	male	8.00	382652	NaN	Queenstown	Master
788	789	Yes	Lower	Dean, Master. Bertram Vere	male	1.00	C.A. 2315	NaN	Southampton	Master
802	803	Yes	Upper	Carter, Master. William Thornton II	male	11.00	113760	B96 B98	Southampton	Master
803	804	Yes	Lower	Thomas, Master. Assad Alexander	male	0.42	2625	NaN	Cherbourg	Master
819	820	No	Lower	Skoog, Master. Karl Thorsten	male	10.00	347088	NaN	Southampton	Master
824	825	No	Lower	Panula, Master. Urho Abraham	male	2.00	3101295	NaN	Southampton	Master
827	828	Yes	Middle	Mallet, Master. Andre	male	1.00	S.C./PARIS 2079	NaN	Cherbourg	Master
831	832	Yes	Middle	Richards, Master. George Sibley	male	0.83	29106	NaN	Southampton	Master
850	851	No	Lower	Andersson, Master. Sigvard Harald Elias	male	4.00	347082	NaN	Southampton	Master

In [212]:
```
# How many are there?
title_master['PassengerId'].count()
```

Out[212]: 40

In [213]:
```
# Take the average of the ages that are available and round to 2 decimal places
# NOTE: the mean() function will not include values that = NaN
avg_age_title_master = round(title_master['Age'].mean(),2)
avg_age_title_master
```

Out[213]: 4.57

In [214]:
```
# Which passenger ages are missing?
boys_missing_age = title_master[title_master['Age'].isna()]
boys_missing_age
```

Out[214]:

	PassengerId	Survived	Passenger_class	Name	Sex	Age	Ticket	Cabin_number	Port_of_Embarkation	Title
65	66	Yes	Lower	Moubarek, Master. Gerios	male	NaN	2661	NaN	Cherbourg	Master
159	160	No	Lower	Sage, Master. Thomas Henry	male	NaN	CA. 2343	NaN	Southampton	Master
176	177	No	Lower	Lefebre, Master. Henry Forbes	male	NaN	4133	NaN	Southampton	Master
709	710	Yes	Lower	Moubarek, Master. Halim Gonios ("William George")	male	NaN	2661	NaN	Cherbourg	Master

In [215]:
```
list_index_bma = boys_missing_age['PassengerId'].index
list_index_bma
```

Out[215]:
```
Int64Index([65, 159, 176, 709], dtype='int64')
```

In [216]:
```
# Update the missing ages in Titanic_passengers dataset with the average age calculated for passengers
# with title of 'Master'.

for i in list_index_bma:
    Titanic_passengers.loc[i,'Age'] = avg_age_title_master

#Titanic_passengers
```

In [217]:
```
# Verify a passenger age value has been updated.
Titanic_passengers.loc[65, 'Age']
```

Out[217]:
```
4.57
```

Estimate the missing age values of passengers with title **'Miss'** by taking the average of existing age values in passengers with that title and assign that average age to the missing age value for that passenger.

In [218]:
```
# Get passengers with title 'Miss'
title_Miss = Titanic_passengers[Titanic_passengers['Title'] == 'Miss']

title_Miss.head()
```

Out[218]:

	PassengerId	Survived	Passenger_class	Name	Sex	Age	Ticket	Cabin_number	Port_of_Embarkation	Title
2	3	Yes	Lower	Heikkinen, Miss. Laina	female	26.0	STON/O2. 3101282	NaN	Southampton	Miss
10	11	Yes	Lower	Sandstrom, Miss. Marguerite Rut	female	4.0	PP 9549	G6	Southampton	Miss
11	12	Yes	Upper	Bonnell, Miss. Elizabeth	female	58.0	113783	C103	Southampton	Miss
14	15	No	Lower	Vestrom, Miss. Hulda Amanda Adolfina	female	14.0	350406	NaN	Southampton	Miss
22	23	Yes	Lower	McGowan, Miss. Anna "Annie"	female	15.0	330923	NaN	Queenstown	Miss

In [219]:
```
# How many are there?
title_Miss['PassengerId'].count()
```

Out[219]: 182

In [220]:
```
# Take the average of the ages that are available and round to 2 decimal places

avg_age_title_Miss = round(title_Miss['Age'].mean(),2)

avg_age_title_Miss
```

Out[220]: 21.77

In [221]:
```
# Which passenger ages are missing?
young_ladies_missing_age = title_Miss[title_Miss['Age'].isna()]

young_ladies_missing_age
```

Out[221]:

	PassengerId	Survived	Passenger_class	Name	Sex	Age	Ticket	Cabin_number	Port_of_Embarkation	Title
28	29	Yes	Lower	O'Dwyer, Miss. Ellen "Nellie"	female	NaN	330959	NaN	Queenstown	Miss
32	33	Yes	Lower	Glynn, Miss. Mary Agatha	female	NaN	335677	NaN	Queenstown	Miss
47	48	Yes	Lower	O'Driscoll, Miss. Bridget	female	NaN	14311	NaN	Queenstown	Miss
82	83	Yes	Lower	McDermott, Miss. Brigdet Delia	female	NaN	330932	NaN	Queenstown	Miss
109	110	Yes	Lower	Moran, Miss. Bertha	female	NaN	371110	NaN	Queenstown	Miss
128	129	Yes	Lower	Peter, Miss. Anna	female	NaN	2668	F E69	Cherbourg	Miss
180	181	No	Lower	Sage, Miss. Constance Gladys	female	NaN	CA. 2343	NaN	Southampton	Miss
198	199	Yes	Lower	Madigan, Miss. Margaret "Maggie"	female	NaN	370370	NaN	Queenstown	Miss
229	230	No	Lower	Lefebre, Miss. Mathilde	female	NaN	4133	NaN	Southampton	Miss
235	236	No	Lower	Harknett, Miss. Alice Phoebe	female	NaN	W./C. 6609	NaN	Southampton	Miss
240	241	No	Lower	Zabour, Miss. Thamine	female	NaN	2665	NaN	Cherbourg	Miss
241	242	Yes	Lower	Murphy, Miss. Katherine "Kate"	female	NaN	367230	NaN	Queenstown	Miss
264	265	No	Lower	Henry, Miss. Delia	female	NaN	382649	NaN	Queenstown	Miss
274	275	Yes	Lower	Healy, Miss. Hanora "Nora"	female	NaN	370375	NaN	Queenstown	Miss
300	301	Yes	Lower	Kelly, Miss. Anna Katherine "Annie Kate"	female	NaN	9234	NaN	Queenstown	Miss
303	304	Yes	Middle	Keane, Miss. Nora A	female	NaN	226593	E101	Queenstown	Miss
306	307	Yes	Upper	Fleming, Miss. Margaret	female	NaN	17421	NaN	Cherbourg	Miss
330	331	Yes	Lower	McCoy, Miss. Agnes	female	NaN	367226	NaN	Queenstown	Miss
358	359	Yes	Lower	McGovern, Miss. Mary	female	NaN	330931	NaN	Queenstown	Miss
359	360	Yes	Lower	Mockler, Miss. Helen Mary "Ellie"	female	NaN	330980	NaN	Queenstown	Miss
368	369	Yes	Lower	Jermyn, Miss. Annie	female	NaN	14313	NaN	Queenstown	Miss
409	410	No	Lower	Lefebre, Miss. Ida	female	NaN	4133	NaN	Southampton	Miss
485	486	No	Lower	Lefebre, Miss. Jeannie	female	NaN	4133	NaN	Southampton	Miss

	PassengerId	Survived	Passenger_class	Name	Sex	Age	Ticket	Cabin_number	Port_of_Embarkation	Title
502	503	No	Lower	O'Sullivan, Miss. Bridget Mary	female	NaN	330909	NaN	Queenstown	Miss
564	565	No	Lower	Meanwell, Miss. (Marion Ogden)	female	NaN	SOTON/O.Q. 392087	NaN	Southampton	Miss
573	574	Yes	Lower	Kelly, Miss. Mary	female	NaN	14312	NaN	Queenstown	Miss
593	594	No	Lower	Bourke, Miss. Mary	female	NaN	364848	NaN	Queenstown	Miss
596	597	Yes	Middle	Leitch, Miss. Jessie Wills	female	NaN	248727	NaN	Southampton	Miss
612	613	Yes	Lower	Murphy, Miss. Margaret Jane	female	NaN	367230	NaN	Queenstown	Miss
653	654	Yes	Lower	O'Leary, Miss. Hanora "Norah"	female	NaN	330919	NaN	Queenstown	Miss
680	681	No	Lower	Peters, Miss. Katie	female	NaN	330935	NaN	Queenstown	Miss
697	698	Yes	Lower	Mullens, Miss. Katherine "Katie"	female	NaN	35852	NaN	Queenstown	Miss
727	728	Yes	Lower	Mannion, Miss. Margareth	female	NaN	36866	NaN	Queenstown	Miss
792	793	No	Lower	Sage, Miss. Stella Anna	female	NaN	CA. 2343	NaN	Southampton	Miss
863	864	No	Lower	Sage, Miss. Dorothy Edith "Dolly"	female	NaN	CA. 2343	NaN	Southampton	Miss

In [222]:
```
# How many passengers with title 'Miss' are there with missing ages?
young_ladies_missing_age['PassengerId'].count()
```

Out[222]:
```
36
```

In [223]:
```
# List of indeces where age is missing for passengers with title 'Miss'
list_index_ylma = young_ladies_missing_age['PassengerId'].index

list_index_ylma
```

Out[223]:
```
Int64Index([ 28,  32,  47,  82, 109, 128, 180, 198, 229, 235, 240, 241, 264,
            274, 300, 303, 306, 330, 358, 359, 368, 409, 485, 502, 564, 573,
            593, 596, 612, 653, 680, 697, 727, 792, 863, 888],
           dtype='int64')
```

In [224]:
```
# Update the missing ages in Titanic_passengers dataset with the average age calculated for passengers
# with title of 'Miss'.

for i in list_index_ylma:
    Titanic_passengers.loc[i,'Age'] = avg_age_title_Miss

Titanic_passengers
```

Out[224]:

	PassengerId	Survived	Passenger_class	Name	Sex	Age	Ticket	Cabin_number	Port_of_Embarkation	Title
0	1	No	Lower	Braund, Mr. Owen Harris	male	22.00	A/5 21171	NaN	Southampton	Mr
1	2	Yes	Upper	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.00	PC 17599	C85	Cherbourg	Mrs
2	3	Yes	Lower	Heikkinen, Miss. Laina	female	26.00	STON/O2. 3101282	NaN	Southampton	Miss
3	4	Yes	Upper	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.00	113803	C123	Southampton	Mrs
4	5	No	Lower	Allen, Mr. William Henry	male	35.00	373450	NaN	Southampton	Mr
...	...	...	...	...	...	...	...	...	...	...
886	887	No	Middle	Montvila, Rev. Juozas	male	27.00	211536	NaN	Southampton	Rev
887	888	Yes	Upper	Graham, Miss. Margaret Edith	female	19.00	112053	B42	Southampton	Miss
888	889	No	Lower	Johnston, Miss. Catherine Helen "Carrie"	female	21.77	W./C. 6607	NaN	Southampton	Miss
889	890	Yes	Upper	Behr, Mr. Karl Howell	male	26.00	111369	C148	Cherbourg	Mr
890	891	No	Lower	Dooley, Mr. Patrick	male	32.00	370376	NaN	Queenstown	Mr

891 rows × 10 columns

In [225]:
```
# Verify a passenger age value has been updated.
Titanic_passengers.loc[28, 'Age']
```

Out[225]: 21.77

Estimate the missing age values of passengers with title **'Mrs'** by taking the average of existing age values in passengers with that title and assign the average age to the missing age value for that passenger.

In [226]:
```
# Get passengers with title 'Mrs'
title_Mrs = Titanic_passengers[Titanic_passengers['Title'] == 'Mrs']

title_Mrs.head()
```

Out[226]:

	PassengerId	Survived	Passenger_class	Name	Sex	Age	Ticket	Cabin_number	Port_of_Embarkation	Title
1	2	Yes	Upper	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	PC 17599	C85	Cherbourg	Mrs
3	4	Yes	Upper	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	113803	C123	Southampton	Mrs
8	9	Yes	Lower	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	347742	NaN	Southampton	Mrs
9	10	Yes	Middle	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	237736	NaN	Cherbourg	Mrs
15	16	Yes	Middle	Hewlett, Mrs. (Mary D Kingcome)	female	55.0	248706	NaN	Southampton	Mrs

In [227]:
```
# How many are there?
title_Mrs['PassengerId'].count()
```

Out[227]: 125

In [228]:
```
# Take the average of the ages that are available and round to 2 decimal places

avg_age_title_Mrs = round(title_Mrs['Age'].mean(),2)

avg_age_title_Mrs
```

Out[228]: 35.9

In [229]:
```
# Which passenger ages are missing?
title_Mrs_missing_age = title_Mrs[title_Mrs['Age'].isna()]

title_Mrs_missing_age.head()
```

Out[229]:

	PassengerId	Survived	Passenger_class	Name	Sex	Age	Ticket	Cabin_number	Port_of_Embarkation	Title
19	20	Yes	Lower	Masselmani, Mrs. Fatima	female	NaN	2649	NaN	Cherbourg	Mrs
31	32	Yes	Upper	Spencer, Mrs. William Augustus (Marie Eugenie)	female	NaN	PC 17569	B78	Cherbourg	Mrs
140	141	No	Lower	Boulos, Mrs. Joseph (Sultana)	female	NaN	2678	NaN	Cherbourg	Mrs
166	167	Yes	Upper	Chibnall, Mrs. (Edith Martha Bowerman)	female	NaN	113505	E33	Southampton	Mrs
186	187	Yes	Lower	O'Brien, Mrs. Thomas (Johanna "Hannah" Godfrey)	female	NaN	370365	NaN	Queenstown	Mrs

In [230]:
```
# How many passengers with title 'Mrs' are there with missing ages?
title_Mrs_missing_age['PassengerId'].count()
```

Out[230]: 17

In [231]:
```
# List of indeces where age is missing for passengers with title 'Mrs'
list_index_tmma = title_Mrs_missing_age['PassengerId'].index

list_index_tmma
```

Out[231]:
```
Int64Index([ 19,  31, 140, 166, 186, 256, 334, 347, 367, 375, 415, 431, 457,
            533, 578, 669, 849],
           dtype='int64')
```

In [232]:
```
# Update the missing ages in Titanic_passengers dataset with the average age calculated for passengers
# with title of 'Mrs'.

for i in list_index_tmma:
    Titanic_passengers.loc[i,'Age'] = avg_age_title_Mrs

Titanic_passengers
```

Out[232]:

	PassengerId	Survived	Passenger_class	Name	Sex	Age	Ticket	Cabin_number	Port_of_Embarkation	Title
0	1	No	Lower	Braund, Mr. Owen Harris	male	22.00	A/5 21171	NaN	Southampton	Mr
1	2	Yes	Upper	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.00	PC 17599	C85	Cherbourg	Mrs
2	3	Yes	Lower	Heikkinen, Miss. Laina	female	26.00	STON/O2. 3101282	NaN	Southampton	Miss
3	4	Yes	Upper	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.00	113803	C123	Southampton	Mrs
4	5	No	Lower	Allen, Mr. William Henry	male	35.00	373450	NaN	Southampton	Mr
...	...	...	...	...	...	...	...	...	...	...
886	887	No	Middle	Montvila, Rev. Juozas	male	27.00	211536	NaN	Southampton	Rev
887	888	Yes	Upper	Graham, Miss. Margaret Edith	female	19.00	112053	B42	Southampton	Miss
888	889	No	Lower	Johnston, Miss. Catherine Helen "Carrie"	female	21.77	W./C. 6607	NaN	Southampton	Miss
889	890	Yes	Upper	Behr, Mr. Karl Howell	male	26.00	111369	C148	Cherbourg	Mr
890	891	No	Lower	Dooley, Mr. Patrick	male	32.00	370376	NaN	Queenstown	Mr

891 rows × 10 columns

In [233]:
```
# Verify a passenger age value has been updated.
Titanic_passengers.loc[140, 'Age']
```

Out[233]: 35.9

Estimate the missing age values of passengers with title **'Mr'** by taking the average of existing age values in passengers with that title and assign the average age to the missing age value for that passenger.

In [234]:
```
# Get passengers with title 'Mr'
title_Mr = Titanic_passengers[Titanic_passengers['Title'] == 'Mr']

title_Mr.head()
```

Out[234]:

	PassengerId	Survived	Passenger_class	Name	Sex	Age	Ticket	Cabin_number	Port_of_Embarkation	Title
0	1	No	Lower	Braund, Mr. Owen Harris	male	22.0	A/5 21171	NaN	Southampton	Mr
4	5	No	Lower	Allen, Mr. William Henry	male	35.0	373450	NaN	Southampton	Mr
5	6	No	Lower	Moran, Mr. James	male	NaN	330877	NaN	Queenstown	Mr
6	7	No	Upper	McCarthy, Mr. Timothy J	male	54.0	17463	E46	Southampton	Mr
12	13	No	Lower	Saundercock, Mr. William Henry	male	20.0	A/5. 2151	NaN	Southampton	Mr

In [235]:
```
# How many are there?
title_Mr['PassengerId'].count()
```

Out[235]: 517

In [236]:
```
# Take the average of the ages that are available and round to 2 decimal places

avg_age_title_Mr = round(title_Mr['Age'].mean(),2)

avg_age_title_Mr
```

Out[236]: 32.37

In [237]:
```
# Which passenger ages are missing?
title_Mr_missing_age = title_Mr[title_Mr['Age'].isna()]

title_Mr_missing_age.head()
```

Out[237]:

	PassengerId	Survived	Passenger_class	Name	Sex	Age	Ticket	Cabin_number	Port_of_Embarkation	Title
5	6	No	Lower	Moran, Mr. James	male	NaN	330877	NaN	Queenstown	Mr
17	18	Yes	Middle	Williams, Mr. Charles Eugene	male	NaN	244373	NaN	Southampton	Mr
26	27	No	Lower	Emir, Mr. Farred Chehab	male	NaN	2631	NaN	Cherbourg	Mr
29	30	No	Lower	Todoroff, Mr. Lalio	male	NaN	349216	NaN	Southampton	Mr
36	37	Yes	Lower	Mamee, Mr. Hanna	male	NaN	2677	NaN	Cherbourg	Mr

In [238]:
```
# How many passengers with title 'Mr' are there with missing ages?
title_Mr_missing_age['PassengerId'].count()
```

Out[238]:
```
119
```

In [239]:
```
# List of indeces where age is missing for passengers with title 'Mr'
list_index_tmrma = title_Mr_missing_age['PassengerId'].index

list_index_tmrma
```

Out[239]:
```
Int64Index([  5,  17,  26,  29,  36,  42,  45,  46,  48,  55,
             ...
             825, 826, 828, 832, 837, 839, 846, 859, 868, 878],
            dtype='int64', length=119)
```

In [240]:
```
# Update the missing ages in Titanic_passengers dataset with the average age calculated for passengers
# with title of 'Mr'.

for i in list_index_tmrma:
    Titanic_passengers.loc[i,'Age'] = avg_age_title_Mr

Titanic_passengers
```

Out[240]:

	PassengerId	Survived	Passenger_class	Name	Sex	Age	Ticket	Cabin_number	Port_of_Embarkation	Title
0	1	No	Lower	Braund, Mr. Owen Harris	male	22.00	A/5 21171	NaN	Southampton	Mr
1	2	Yes	Upper	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.00	PC 17599	C85	Cherbourg	Mrs
2	3	Yes	Lower	Heikkinen, Miss. Laina	female	26.00	STON/O2. 3101282	NaN	Southampton	Miss
3	4	Yes	Upper	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.00	113803	C123	Southampton	Mrs
4	5	No	Lower	Allen, Mr. William Henry	male	35.00	373450	NaN	Southampton	Mr
...	...	...	...	...	...	...	...	...	...	...
886	887	No	Middle	Montvila, Rev. Juozas	male	27.00	211536	NaN	Southampton	Rev
887	888	Yes	Upper	Graham, Miss. Margaret Edith	female	19.00	112053	B42	Southampton	Miss
888	889	No	Lower	Johnston, Miss. Catherine Helen "Carrie"	female	21.77	W./C. 6607	NaN	Southampton	Miss
889	890	Yes	Upper	Behr, Mr. Karl Howell	male	26.00	111369	C148	Cherbourg	Mr
890	891	No	Lower	Dooley, Mr. Patrick	male	32.00	370376	NaN	Queenstown	Mr

891 rows × 10 columns

In [241]:
```
# Verify a passenger age value has been updated.
Titanic_passengers.loc[828, 'Age']
```

Out[241]:
```
32.37
```

Which passengers are left that do not have an age value?

In [242]:
```
# Any remaining passengers without an age value?
Titanic_passengers[Titanic_passengers['Age'].isna()]
```

Out[242]:

	PassengerId	Survived	Passenger_class	Name	Sex	Age	Ticket	Cabin_number	Port_of_Embarkation	Title
766	767	No	Upper	Brewe, Dr. Arthur Jackson	male	NaN	112379	NaN	Cherbourg	Dr

In [243]:
```
# Only 1 passenger remains without an age value.
# Since this passenger is a male who is a doctor 'Dr', will use the average age value for 'Mr'.
Titanic_passengers.loc[766,'Age'] = avg_age_title_Mr
```

In [244]:
```
Titanic_passengers.loc[766]
```

Out[244]:
```
PassengerId                          767
Survived                              No
Passenger_class                    Upper
Name                 Brewe, Dr. Arthur Jackson
Sex                                 male
Age                                32.37
Ticket                            112379
Cabin_number                         NaN
Port_of_Embarkation             Cherbourg
Title                                  Dr
Name: 766, dtype: object
```

In [245]:
```
# Verify there are no more passengers without an age value.
Titanic_passengers[Titanic_passengers['Age'].isna()]
```

Out[245]:

	PassengerId	Survived	Passenger_class	Name	Sex	Age	Ticket	Cabin_number	Port_of_Embarkation	Title

Add 'Age_group' column to Titanic_passengers dataset

In [246]:
```
# Add column 'Age_group' which will have values of the age group category based on the passenger age
# The age group category:
#            age less than 13         'Child'
#            age between 13 and 17.x  'Teen'
#            age between 18 and 59.x  'Adult'
#            age greater than 60      'Senior'

Titanic_passengers['Age_group'] = Titanic_passengers['Age'].apply(age_group_category)
```

In [247]:
```
Titanic_passengers
```

Out[247]:

	PassengerId	Survived	Passenger_class	Name	Sex	Age	Ticket	Cabin_number	Port_of_Embarkation	Title	Age_group
0	1	No	Lower	Braund, Mr. Owen Harris	male	22.00	A/5 21171	NaN	Southampton	Mr	Adult
1	2	Yes	Upper	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.00	PC 17599	C85	Cherbourg	Mrs	Adult
2	3	Yes	Lower	Heikkinen, Miss. Laina	female	26.00	STON/O2. 3101282	NaN	Southampton	Miss	Adult
3	4	Yes	Upper	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.00	113803	C123	Southampton	Mrs	Adult
4	5	No	Lower	Allen, Mr. William Henry	male	35.00	373450	NaN	Southampton	Mr	Adult
...	...	...	...	...	...	...	...	...	...	...	...
886	887	No	Middle	Montvila, Rev. Juozas	male	27.00	211536	NaN	Southampton	Rev	Adult
887	888	Yes	Upper	Graham, Miss. Margaret Edith	female	19.00	112053	B42	Southampton	Miss	Adult
888	889	No	Lower	Johnston, Miss. Catherine Helen "Carrie"	female	21.77	W./C. 6607	NaN	Southampton	Miss	Adult
889	890	Yes	Upper	Behr, Mr. Karl Howell	male	26.00	111369	C148	Cherbourg	Mr	Adult
890	891	No	Lower	Dooley, Mr. Patrick	male	32.00	370376	NaN	Queenstown	Mr	Adult

891 rows × 11 columns

Highest number of survivers by category 'Age_group' & survival rate

In [248]:
```
# How many passengers survived by 'Age_group'
age_group_survive_info = Titanic_passengers[['Age_group','Survived']]

age_group_survived = age_group_survive_info[age_group_survive_info['Survived'] == 'Yes'].groupby('Age_group').co

# Number of passengers that survived by 'Age_group'.
total_survived_by_age_group = age_group_survived.sort_values('Survived', ascending=False)
total_survived_by_age_group
```

Out[248]:

Age_group	Survived
Adult	272
Child	42
Teen	21
Senior	7

In [249]:
```
# Total number of passengers by age group.
total_by_age_group = age_group_survive_info.groupby(['Age_group']).count()
total_by_age_group.rename(columns ={'Survived':'Total_Passengers'},inplace=True)
total_by_age_group
```

Out[249]:

Age_group	Total_Passengers
Adult	748
Child	73
Senior	26
Teen	44

In [250]:
```
# Join into 1 table the passengers survived by 'Age_group' with total passengers by 'Age_group'
joined_tot_by_age_group = total_survived_by_age_group.join(total_by_age_group)

joined_tot_by_age_group
```

Out[250]:

Age_group	Survived	Total_Passengers
Adult	272	748
Child	42	73
Teen	21	44
Senior	7	26

In [251]:
```
# Add column'Survival rate' which contains survival rate of each age group.
joined_tot_by_age_group['Survival rate'] = joined_tot_by_age_group['Survived'] / joined_tot_by_age_group['Total_

joined_tot_by_age_group
```

Out[251]:

Age_group	Survived	Total_Passengers	Survival rate
Adult	272	748	0.363636
Child	42	73	0.575342
Teen	21	44	0.477273
Senior	7	26	0.269231

In [252]:
```
# Add column of 'Overall_%_of_Passengers' in the dataset by 'Age_group'
# The highest number of survivers by age group were adults (272). They were also the highest age group of
# passengers (748 out of 891 or 84% of passengers were adults).
# However children had the highest survival rate.

joined_tot_by_age_group['Overall_%_of_Passengers'] =  joined_tot_by_age_group['Total_Passengers'] / Total_Passen

joined_tot_by_age_group
```

Out[252]:

Age_group	Survived	Total_Passengers	Survival rate	Overall_%_of_Passengers
Adult	272	748	0.363636	0.839506
Child	42	73	0.575342	0.081930
Teen	21	44	0.477273	0.049383
Senior	7	26	0.269231	0.029181

Highest number of survivors by category 'Sex' & survival rate

In [253]:
```
# The total number of males and females passengers that survived.

gender_survive_info = Titanic_passengers[['Sex','Survived']]
gender_survived = gender_survive_info[gender_survive_info['Survived'] == 'Yes'].groupby('Sex').count().sort_valu
                                                                                ascending=False)

gender_survived
```

Out[253]:

Sex	Survived
female	233
male	109

In [254]:
```
# The overall total of passengers by male and female.
total_by_Sex = Titanic_passengers.groupby('Sex').count()['PassengerId'].to_frame()
total_by_Sex.rename(columns={'PassengerId':'Total_passengers'},inplace=True)
total_by_Sex
```

Out[254]:

Sex	Total_passengers
female	314
male	577

In [255]:
```
# Join into 1 table the passengers survived by 'Sex' with total passengers by 'Sex'
joined_tot_by_Sex = gender_survived.join(total_by_Sex)
joined_tot_by_Sex
```

Out[255]:

Sex	Survived	Total_passengers
female	233	314
male	109	577

In [256]:
```
# Add column'Survival rate' which contains survival rate of each 'Sex'.
# The highest number of survivers by 'Sex' were females(233).
# Females survival rate was 74% compared to males which was only 19%, even though there were 1.8 times more
# males than females.

joined_tot_by_Sex['Survival rate'] = joined_tot_by_Sex['Survived'] / joined_tot_by_Sex['Total_passengers']

joined_tot_by_Sex
```

Out[256]:

Sex	Survived	Total_passengers	Survival rate
female	233	314	0.742038
male	109	577	0.188908

In [257]:
```
# How many more times are there males than females?
round((joined_tot_by_Sex.loc['male','Total_passengers'] / joined_tot_by_Sex.loc['female','Total_passengers']), 1
```

Out[257]: 1.8

Highest number of survivors by category 'Port_of_Embarkation' & survival rate

First assign missing **'Port_of_Embarkation'** for 2 passengers.

In [258]:
```
# Two passengers with same ticket number 'Port_of_Embarkation' do not have a value.
missing_port_embarked = Titanic_passengers[Titanic_passengers['Port_of_Embarkation'].isna()]

missing_port_embarked
```

Out[258]:

	PassengerId	Survived	Passenger_class	Name	Sex	Age	Ticket	Cabin_number	Port_of_Embarkation	Title	Age_group
61	62	Yes	Upper	Icard, Miss. Amelie	female	38.0	113572	B28	NaN	Miss	Adult
829	830	Yes	Upper	Stone, Mrs. George Nelson (Martha Evelyn)	female	62.0	113572	B28	NaN	Mrs	Senior

In [259]:
```
# Is there anyone with the same cabin number that has 'Port_of_Embarkation' information?  No.
Titanic_passengers[Titanic_passengers['Cabin_number'] == 'B28']
```

Out[259]:

	PassengerId	Survived	Passenger_class	Name	Sex	Age	Ticket	Cabin_number	Port_of_Embarkation	Title	Age_group
61	62	Yes	Upper	Icard, Miss. Amelie	female	38.0	113572	B28	NaN	Miss	Adult
829	830	Yes	Upper	Stone, Mrs. George Nelson (Martha Evelyn)	female	62.0	113572	B28	NaN	Mrs	Senior

In [260]:
```
# From where did other 'Upper' class passengers embarked?
Titanic_passengers[Titanic_passengers['Passenger_class'] == 'Upper'].groupby('Port_of_Embarkation')['PassengerId
```

Out[260]:
```
Port_of_Embarkation
Cherbourg       85
Queenstown       2
Southampton    127
Name: PassengerId, dtype: int64
```

In [261]:
```
# Is there any other passenger with last name 'Icard' or 'Stone'? If so, then can use where they embarked.
# Else use 'Southampton' as it is where most passengers embarked.

any_other_passenger = Titanic_passengers['Name'].str.contains('Icard')
Titanic_passengers[any_other_passenger]
```

Out[261]:

	PassengerId	Survived	Passenger_class	Name	Sex	Age	Ticket	Cabin_number	Port_of_Embarkation	Title	Age_group
61	62	Yes	Upper	Icard, Miss. Amelie	female	38.0	113572	B28	NaN	Miss	Adult

In [262]:
```
# No Mr. Stone or other 'Icard' found, so use 'Southhampton'.
any_other_passenger = Titanic_passengers['Name'].str.contains('Stone')
Titanic_passengers[any_other_passenger]
```

Out[262]:

	PassengerId	Survived	Passenger_class	Name	Sex	Age	Ticket	Cabin_number	Port_of_Embarkation	Title	Age_group
319	320	Yes	Upper	Spedden, Mrs. Frederic Oakley (Margaretta Corn...	female	40.0	16966	E34	Cherbourg	Mrs	Adult
829	830	Yes	Upper	Stone, Mrs. George Nelson (Martha Evelyn)	female	62.0	113572	B28	NaN	Mrs	Senior

In [263]:
```
# List of indeces where 'Port_of_Embarkation' is missing for 2 passengers.
list_index_pe = missing_port_embarked['PassengerId'].index

list_index_pe
```

Out[263]:
```
Int64Index([61, 829], dtype='int64')
```

In [264]:
```
# Update the missing 'Port_of_Embarkation' in Titanic_passengers dataset with 'Southampton'.
for i in list_index_pe:
    Titanic_passengers.loc[i,'Port_of_Embarkation'] = 'Southampton'

# Test the change is there for 'Icard, Miss. Amelie'
Titanic_passengers.loc[61]
```

Out[264]:
```
PassengerId                          62
Survived                            Yes
Passenger_class                   Upper
Name                 Icard, Miss. Amelie
Sex                              female
Age                                38.0
Ticket                           113572
Cabin_number                        B28
Port_of_Embarkation          Southampton
Title                               Miss
Age_group                          Adult
Name: 61, dtype: object
```

In [265]:
```
# Verify no more missing embarked city location. Verfied.
Titanic_passengers[Titanic_passengers['Port_of_Embarkation'].isna()]
```

Out[265]:

PassengerId	Survived	Passenger_class	Name	Sex	Age	Ticket	Cabin_number	Port_of_Embarkation	Title	Age_group

All embarked from cities are assigned a value.

The total number of passengers who survived based on cities they boarded the Titanic from **(port of embarkation)**.

In [266]:
```
# The total number of passengers who survived based on cities they boarded the Titanic from (port of embarkation
port_survive_info = Titanic_passengers[['Port_of_Embarkation','Survived']]
port_survived = port_survive_info[port_survive_info['Survived'] == 'Yes'].groupby('Port_of_Embarkation').count()
                                                                                    ascending=False)
port_survived
```

Out[266]:

Port_of_Embarkation	Survived
Southampton	219
Cherbourg	93
Queenstown	30

In [267]:
```
# Total number of passengers that boarded from each city.
total_by_City = Titanic_passengers.groupby('Port_of_Embarkation').count()['PassengerId'].to_frame()
total_by_City.sort_values('PassengerId', ascending=False, inplace=True)
total_by_City.rename(columns={'PassengerId':'Total_passengers'},inplace=True)
```

In [268]:
```
total_by_City
```

Out[268]:

Port_of_Embarkation	Total_passengers
Southampton	646
Cherbourg	168
Queenstown	77

In [269]:
```
# Join into 1 table the passengers survived by 'Port_of_Embarkation' with total passengers by 'Port_of_Embarkati
joined_total_by_city = port_survived.join(total_by_City)

# Add column'Survival rate' which contains survival rate based on each city passengers embarked from.
joined_total_by_city['Survival_rate'] = joined_total_by_city['Survived'] / joined_total_by_city['Total_passenger

# The highest number of survivers embarked from Southhampton (219). The total number of passengers embarked
# from Southampton (646) with a survival rate of 34%.
joined_total_by_city
```

Out[269]:

Port_of_Embarkation	Survived	Total_passengers	Survival_rate
Southampton	219	646	0.339009
Cherbourg	93	168	0.553571
Queenstown	30	77	0.389610

In [ ]:

In [ ]: