In [2]:
```
#1-General view, picture,  (Genel Resim)
```

In [2]:
```
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
pd.set_option('display.max_columns',None)
pd.set_option('display.width',500)
df=sns.load_dataset('titanic')
```

In [3]:
```
df.head()
```

Out[3]:

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

In [4]:
```
df.tail()
```

Out[4]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embarl
886	0	2	male	27.0	0	0	13.00	S	Second	man	True	NaN	Southa
887	1	1	female	19.0	0	0	30.00	S	First	woman	False	B	Southa
888	0	3	female	NaN	1	2	23.45	S	Third	woman	False	NaN	Southa
889	1	1	male	26.0	0	0	30.00	C	First	man	True	C	Che
890	0	3	male	32.0	0	0	7.75	Q	Third	man	True	NaN	Quee

In [6]:
```
df.shape
```

Out[6]:
```
(891, 15)
```

In [7]:
```
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 [8]:
```
df.columns
```

Out[8]:
```
Index(['survived', 'pclass', 'sex', 'age', 'sibsp', 'parch', 'fare', 'embarked', 'clas
s', 'who', 'adult_male', 'deck', 'embark_town', 'alive', 'alone'], dtype='object')
```

In [9]:
```
df.index
```

Out[9]:
```
RangeIndex(start=0, stop=891, step=1)
```

In [10]:
```
df.describe().T
```

Out[10]:

	count	mean	std	min	25%	50%	75%	max
survived	891.0	0.383838	0.486592	0.00	0.0000	0.0000	1.0	1.0000
pclass	891.0	2.308642	0.836071	1.00	2.0000	3.0000	3.0	3.0000
age	714.0	29.699118	14.526497	0.42	20.1250	28.0000	38.0	80.0000
sibsp	891.0	0.523008	1.102743	0.00	0.0000	0.0000	1.0	8.0000
parch	891.0	0.381594	0.806057	0.00	0.0000	0.0000	0.0	6.0000
fare	891.0	32.204208	49.693429	0.00	7.9104	14.4542	31.0	512.3292

In [13]:
```
df.isnull().values.any()
```

Out[13]:
```
True
```

In [14]:
```
df.isnull().sum()
```

Out[14]:
```
survived         0
pclass           0
sex              0
age            177
sibsp            0
parch            0
fare             0
embarked         2
class            0
who              0
adult_male       0
deck           688
embark_town      2
alive            0
alone            0
dtype: int64
```

In [17]:
```
def check_df (dataframe, head=5):
    print("######Shape######")
    print(dataframe.shape)
    print("######Types######")
    print(dataframe.dtypes)
    print("######Head######")
    print(dataframe.head(head))
    print("######Tail######")
    print(dataframe.tail(head))
    print("######NA######")
    print(dataframe.isnull().sum())
    print("######Quantiles#####")
    print(dataframe.describe([0,0.05,0.50,0.95,0.99,1]).T)
```

In [18]:
```
check_df(df)
```

```
######Shape######
(891, 15)
######Types######
survived          int64
pclass            int64
sex              object
age             float64
sibsp             int64
parch             int64
fare            float64
embarked         object
class          category
who              object
adult_male         bool
deck           category
embark_town      object
alive            object
alone              bool
dtype: object
######Head######
   survived  pclass     sex   age  sibsp  parch     fare embarked  class    who  adult_m
ale deck  embark_town alive  alone
0         0       3    male  22.0      1      0   7.2500        S  Third    man        T
rue  NaN  Southampton    no  False
1         1       1  female  38.0      1      0  71.2833        C  First  woman       Fa
lse    C    Cherbourg   yes  False
2         1       3  female  26.0      0      0   7.9250        S  Third  woman       Fa
lse  NaN  Southampton   yes   True
3         1       1  female  35.0      1      0  53.1000        S  First  woman       Fa
lse    C  Southampton   yes  False
4         0       3    male  35.0      0      0   8.0500        S  Third    man        T
rue  NaN  Southampton    no   True
######Tail######
     survived  pclass     sex   age  sibsp  parch   fare embarked   class    who  adult_
male deck  embark_town alive  alone
886         0       2    male  27.0      0      0  13.00        S  Second    man
True  NaN  Southampton    no   True
887         1       1  female  19.0      0      0  30.00        S   First  woman       F
alse    B  Southampton   yes   True
888         0       3  female   NaN      1      2  23.45        S   Third  woman       F
alse  NaN  Southampton    no  False
889         1       1    male  26.0      0      0  30.00        C   First    man
True    C    Cherbourg   yes   True
890         0       3    male  32.0      0      0   7.75        Q   Third    man
True  NaN   Queenstown    no   True
######NA######
survived         0
pclass           0
sex              0
age            177
sibsp            0
parch            0
fare             0
embarked         2
class            0
who              0
adult_male       0
deck           688
embark_town      2
alive            0
alone            0
dtype: int64
```

```
######Quantiles#####
          count        mean         std   min    0%     5%       50%         95%         99%
100%       max
survived  891.0    0.383838    0.486592  0.00  0.00  0.000    0.0000     1.00000     1.00000
1.0000     1.0000
pclass    891.0    2.308642    0.836071  1.00  1.00  1.000    3.0000     3.00000     3.00000
3.0000     3.0000
age       714.0   29.699118   14.526497  0.42  0.42  4.000   28.0000    56.00000    65.87000
80.0000    80.0000
sibsp     891.0    0.523008    1.102743  0.00  0.00  0.000    0.0000     3.00000     5.00000
8.0000     8.0000
parch     891.0    0.381594    0.806057  0.00  0.00  0.000    0.0000     2.00000     4.00000
6.0000     6.0000
fare      891.0   32.204208   49.693429  0.00  0.00  7.225   14.4542   112.07915   249.00622
512.3292   512.3292
```

In [ ]:
```
#2-Analysis of Categorical Variables
```

In [4]:
```
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
pd.set_option('display.max_columns',None)
pd.set_option('display.width',500)
df=sns.load_dataset("titanic")
```

In [20]:
```
df.head()
```

Out[20]:

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

In [28]:
```
df['survived'].value_counts()
```

Out[28]:
```
survived
0    549
1    342
Name: count, dtype: int64
```

In [31]:
```
df['sex'].unique()
```

Out[31]:
```
array(['male', 'female'], dtype=object)
```

In [29]:
```
check_df(df)
```

```
######Shape######
(891, 15)
######Types######
survived          int64
pclass            int64
sex              object
age             float64
sibsp             int64
parch             int64
fare            float64
embarked         object
class          category
who              object
adult_male         bool
deck           category
embark_town      object
alive            object
alone              bool
dtype: object
######Head######
   survived  pclass     sex   age  sibsp  parch     fare embarked  class    who  adult_m
ale deck  embark_town alive  alone
0         0       3    male  22.0      1      0   7.2500        S  Third    man        T
rue  NaN  Southampton    no  False
1         1       1  female  38.0      1      0  71.2833        C  First  woman       Fa
lse    C    Cherbourg   yes  False
2         1       3  female  26.0      0      0   7.9250        S  Third  woman       Fa
lse  NaN  Southampton   yes   True
3         1       1  female  35.0      1      0  53.1000        S  First  woman       Fa
lse    C  Southampton   yes  False
4         0       3    male  35.0      0      0   8.0500        S  Third    man        T
rue  NaN  Southampton    no   True
######Tail######
     survived  pclass     sex   age  sibsp  parch   fare embarked   class    who  adult_
male deck  embark_town alive  alone
886         0       2    male  27.0      0      0  13.00        S  Second    man
True  NaN  Southampton    no   True
887         1       1  female  19.0      0      0  30.00        S   First  woman       F
alse    B  Southampton   yes   True
888         0       3  female   NaN      1      2  23.45        S   Third  woman       F
alse  NaN  Southampton    no  False
889         1       1    male  26.0      0      0  30.00        C   First    man
True    C    Cherbourg   yes   True
890         0       3    male  32.0      0      0   7.75        Q   Third    man
True  NaN   Queenstown    no   True
######NA######
survived         0
pclass           0
sex              0
age            177
sibsp            0
parch            0
fare             0
embarked         2
class            0
who              0
adult_male       0
deck           688
embark_town      2
alive            0
alone            0
dtype: int64
```

```
######Quantiles#####
          count       mean        std   min    0%     5%      50%        95%        99%
100%       max
survived  891.0   0.383838   0.486592  0.00  0.00  0.000   0.0000    1.00000    1.00000
1.0000    1.0000
pclass    891.0   2.308642   0.836071  1.00  1.00  1.000   3.0000    3.00000    3.00000
3.0000    3.0000
age       714.0  29.699118  14.526497  0.42  0.42  4.000  28.0000   56.00000   65.87000
80.0000   80.0000
sibsp     891.0   0.523008   1.102743  0.00  0.00  0.000   0.0000    3.00000    5.00000
8.0000    8.0000
parch     891.0   0.381594   0.806057  0.00  0.00  0.000   0.0000    2.00000    4.00000
6.0000    6.0000
fare      891.0  32.204208  49.693429  0.00  0.00  7.225  14.4542  112.07915  249.00622
512.3292  512.3292
```

In [32]:
```
cat_cols = [col for col in df.columns if str(df[col].dtypes) in ["category","object","boo
num_but_cat = [col for col in df.columns if df[col].nunique()<10 and df[col].dtypes in ["
cat_but_car = [col for col in df.columns if df[col].nunique() > 20 and str(df[col].dtypes
cat_cols=cat_cols+num_but_cat
```

In [33]:
```
cat_cols
```

Out[33]:
```
['sex',
 'embarked',
 'class',
 'who',
 'adult_male',
 'deck',
 'embark_town',
 'alive',
 'alone',
 'survived',
 'pclass',
 'sibsp',
 'parch']
```

In [9]:
```
def cat_summmary (dataframe, col_name):
    print(pd.DataFrame({col_name:dataframe[col_name].value_counts(),"Ratio":100*dataframe
```

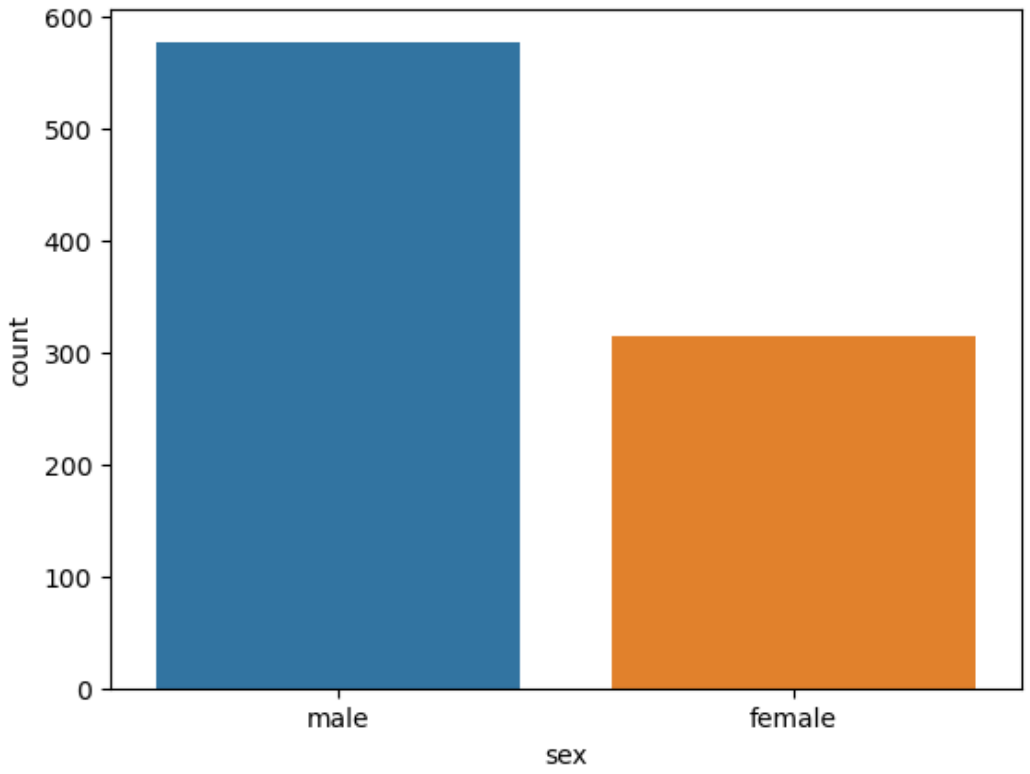
In [10]:
```
cat_summmary(df,"sex")
```

```
        sex      Ratio
sex
male    577  64.758698
female  314  35.241302
```

In [15]:
```
def cat_summary(dataframe, col_name, plot=False):
    print(pd.DataFrame({col_name: dataframe[col_name].value_counts(),
                        "Ratio":  100*dataframe[col_name].value_counts()/len(dataframe)
                       }))
    if plot:
        sns.countplot(x=dataframe[col_name],data=dataframe)
        plt.show(block=True)
```

In [16]:
```
cat_summary(df,"sex",plot=True)
```

```
        sex      Ratio
sex
male    577  64.758698
female  314  35.241302
```



In [17]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
pd.set_option('display.max_columns',None)
pd.set_option('display.width',500)
df=sns.load_dataset("titanic")
df.head()
```

Out[17]:

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

In [18]:
```
df[["age","fare"]].describe().T
```

Out[18]:

	count	mean	std	min	25%	50%	75%	max
age	714.0	29.699118	14.526497	0.42	20.1250	28.0000	38.0	80.0000
fare	891.0	32.204208	49.693429	0.00	7.9104	14.4542	31.0	512.3292

In [20]:
```
num_cols=[col for col in df.columns if df[col].dtypes in ["int","float","int64"]]
```

In [21]:
```
cat_cols = [col for col in df.columns if str(df[col].dtypes) in ["category","object","boo
num_but_cat = [col for col in df.columns if df[col].nunique() < 10 and df[col].dtypes in
cat_but_car = [col for col in df.columns if df[col].nunique() > 20 and str(df[col].dtypes
cat_cols=cat_cols+num_but_cat
```

In [22]:
```
cat_cols= [col for col in cat_cols if col not in cat_but_car]
```

In [23]:
```
num_cols=[col for col in df.columns if df[col].dtypes in ["int","float","int64"]]
num_cols=[col for col in num_cols if col not in cat_cols]
```

In [24]:
```
num_cols
```

Out[24]:
```
['age', 'fare']
```

In [32]:
```
def num_summary (dataframe,num_cols):
    quantiles = [0.1,0.3,0.4,0.75]
    print(dataframe[num_cols].describe(quantiles).T)
```

In [ ]:
```

```

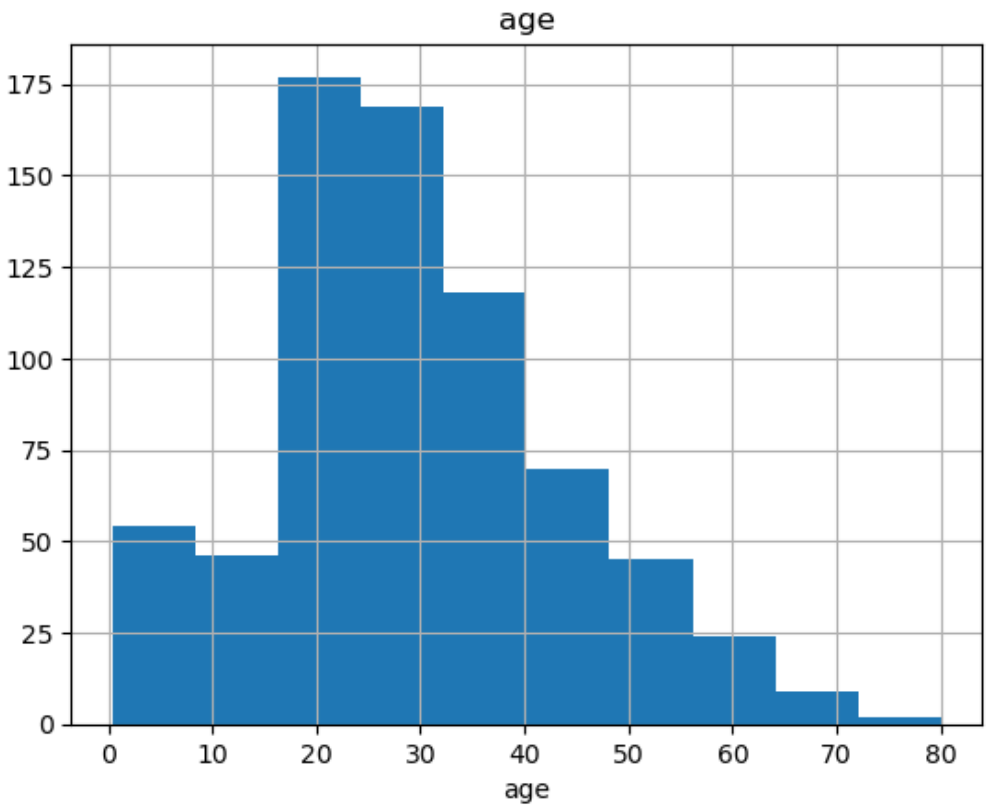
In [28]:
```
num_summary(df,"age")
```

```
count    714.000000
mean      29.699118
std       14.526497
min        0.420000
10%       14.000000
30%       22.000000
40%       25.000000
50%       28.000000
75%       38.000000
max       80.000000
Name: age, dtype: float64
```

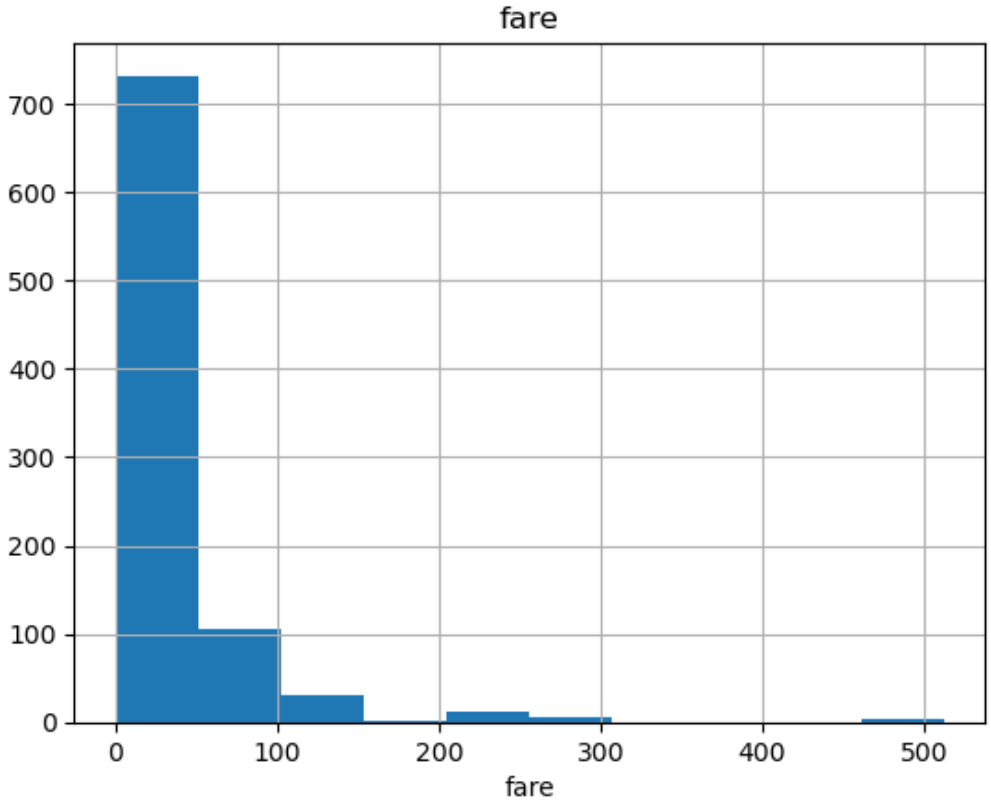
In [34]:
```
def num_summary(dataframe, numerical_cols, plot=False):
    quantiles=[0.05,0.10,0.20,0.30,0.40,0.50,0.60,0.70,0.80,0.90,0.95,0.99]
    print(dataframe[numerical_cols].describe(quantiles).T)
    if plot:
        dataframe[numerical_cols].hist()
        plt.xlabel(numerical_cols)
        plt.title(numerical_cols)
        plt.show(block=True)
```

In [35]:
```
for col in num_cols:
    num_summary(df,col,plot=True)
```

```
count    714.000000
mean      29.699118
std       14.526497
min        0.420000
5%         4.000000
10%       14.000000
20%       19.000000
30%       22.000000
40%       25.000000
50%       28.000000
60%       31.800000
70%       36.000000
80%       41.000000
90%       50.000000
95%       56.000000
99%       65.870000
max       80.000000
Name: age, dtype: float64
```



```
count    891.000000
mean      32.204208
std       49.693429
min        0.000000
5%         7.225000
10%        7.550000
20%        7.854200
30%        8.050000
40%       10.500000
50%       14.454200
60%       21.679200
70%       27.000000
80%       39.687500
90%       77.958300
95%      112.079150
99%      249.006220
max      512.329200
Name: fare, dtype: float64
```



In [36]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
pd.set_option('display.max_columns',None)
pd.set_option('display.width',500)
df=sns.load_dataset("titanic")
df.head()
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 [41]:
```
def grab_col_names(dataframe, cat_th=10, car_th=20):
    cat_cols = [col for col in df.columns if str(df[col].dtypes) in ["category","object",'
    num_but_cat = [col for col in df.columns if df[col].nunique() < 10 and df[col].dtypes
    cat_but_car = [col for col in df.columns if df[col].nunique() > 20 and str(df[col].dty
    cat_cols=cat_cols+num_but_cat
    cat_cols= [col for col in cat_cols if col not in cat_but_car]
    num_cols=[col for col in df.columns if df[col].dtypes in ["int","float","int64"]]
    num_cols=[col for col in num_cols if col not in cat_cols]

    print(f"Observations: {dataframe.shape[0]}") #gözlemler
    print(f"Variables: {dataframe.shape[1]}") #değişkenler
    print(f"cat_cols: {len(cat_cols)}") #kategorik değişkenin boyutu
    print(f"cat_cols: {len(num_cols)}")
    print(f"cat_cols: {len(cat_but_car)}")
    print(f"cat_cols: {len(num_but_cat)}")
    return cat_cols, num_cols,cat_but_car
```

In [38]:
```
help(grab_col_names)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[38], line 1
----> 1 help(grab_col_names)

NameError: name 'grab_col_names' is not defined
```

In [39]:
```
cat_cols = [col for col in df.columns if str(df[col].dtypes) in ["category","object","bool
num_but_cat = [col for col in df.columns if df[col].nunique() < 10 and df[col].dtypes in
cat_but_car = [col for col in df.columns if df[col].nunique() > 20 and str(df[col].dtypes
cat_cols=cat_cols+num_but_cat
cat_cols= [col for col in cat_cols if col not in cat_but_car]
```

In [40]:
```
num_cols=[col for col in df.columns if df[col].dtypes in ["int","float","int64"]]
num_cols=[col for col in num_cols if col not in cat_cols]
```

In [42]:
```
grab_col_names(df)
```

```
Observations: 891
Variables: 15
cat_cols: 13
cat_cols: 2
cat_cols: 0
cat_cols: 4
```

Out[42]:
```
(['sex',
  'embarked',
  'class',
  'who',
  'adult_male',
  'deck',
  'embark_town',
  'alive',
  'alone',
  'survived',
  'pclass',
  'sibsp',
  'parch'],
 ['age', 'fare'],
 [])
```

In [44]:
```
cat_cols,num_cols,cat_but_car= grab_col_names(df)
```

```
Observations: 891
Variables: 15
cat_cols: 13
cat_cols: 2
cat_cols: 0
cat_cols: 4
```

In [ ]:
```
#4 Analysis of Target Variable  Hedef Değişken Analizi
```

In [1]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
pd.set_option('display.max_columns',None)
pd.set_option('display.width',500)
df=sns.load_dataset("titanic")
df.head()
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 [5]:
```
for col in df.columns:
    if df[col].dtypes =="bool":
        df[col]=df[col].astype(int)
```

In [6]:
```
def cat_summary(dataframe, col_name, plot=False):
    print(pd.DataFrame({col_name: dataframe[col_name].value_counts(),
                        "Ratio":  100*dataframe[col_name].value_counts()/len(dataframe)
                       }))
    if plot:
        sns.countplot(x=dataframe[col_name],data=dataframe)
        plt.show(block=True)
```

In [8]:
```
def grab_col_names(dataframe, cat_th=10, car_th=20):   #docstring
    """
    Veri setindeki kategorik, numerik ve kategorik fakat kardinal değişkenlerin isimlerini

    #Parameters

    dataframe: dataframe
        değişkenlerin isimleri alınmak istenen dataframe'dir.
    cat_th: int,float
        numerik fakat kategorik olan değişkenler için sınıf eşik değeri
    car_th: int, float
        kategorik fakat kardinal değişkenler için sınıf eşik değeri
    """
    cat_cols = [col for col in df.columns if str(df[col].dtypes) in ["category","object",'
    num_but_cat = [col for col in df.columns if df[col].nunique() < 10 and df[col].dtypes
    cat_but_car = [col for col in df.columns if df[col].nunique() > 20 and str(df[col].dty
    cat_cols=cat_cols+num_but_cat
    cat_cols= [col for col in cat_cols if col not in cat_but_car]
    num_cols=[col for col in df.columns if df[col].dtypes in ["int","float","int64"]]
    num_cols=[col for col in num_cols if col not in cat_cols]

    print(f"Observations: {dataframe.shape[0]}") #gözlemler
    print(f"Variables: {dataframe.shape[1]}") #değişkenler
    print(f"cat_cols: {len(cat_cols)}") #kategorik değişkenin boyutu
    print(f"num_cols: {len(num_cols)}")
    print(f"cat_but_car_cols: {len(cat_but_car)}")
    print(f"num_but_cat_cols: {len(num_but_cat)}")
    return cat_cols, num_cols,cat_but_car
```

In [9]:
```
cat_cols,num_cols,cat_but_car= grab_col_names(df)
```

```
Observations: 891
Variables: 15
cat_cols: 13
num_cols: 2
cat_but_car_cols: 0
num_but_cat_cols: 6
```

In [10]:
```
df["survived"].value_counts()
```

Out[10]:
```
survived
0    549
1    342
Name: count, dtype: int64
```

In [14]:
```
df.groupby("sex")["survived"].mean()
```

Out[14]:
```
sex
female    0.742038
male      0.188908
Name: survived, dtype: float64
```

In [18]:
```
def target_summary_with_cat(dataframe, target, categorical_col):
    print(pd.DataFrame({"TARGET_MEAN":dataframe.groupby(categorical_col)[target].mean()})
```

In [19]:
```
target_summary_with_cat(df, "survived","pclass")
```

```
        TARGET_MEAN
pclass
1          0.629630
2          0.472826
3          0.242363
```

In [ ]:
```
#5 Analysis of Correlation
```

In [20]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
pd.set_option('display.max_columns',None)
pd.set_option('display.width',500)
df=pd.read_csv("data.csv")
df.head()
df.info()
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
Cell In[20], line 6
      4 pd.set_option('display.max_columns',None)
      5 pd.set_option('display.width',500)
----> 6 df=pd.read_csv("data.csv")
      7 df.head()
      8 df.info()

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\io\parsers\readers.py:912, in rea
d_csv(filepath_or_buffer, sep, delimiter, header, names, index_col, usecols, dtype, engi
ne, converters, true_values, false_values, skipinitialspace, skiprows, skipfooter, nrow
s, na_values, keep_default_na, na_filter, verbose, skip_blank_lines, parse_dates, infer_
datetime_format, keep_date_col, date_parser, date_format, dayfirst, cache_dates, iterato
r, chunksize, compression, thousands, decimal, lineterminator, quotechar, quoting, doubl
equote, escapechar, comment, encoding, encoding_errors, dialect, on_bad_lines, delim_whi
tespace, low_memory, memory_map, float_precision, storage_options, dtype_backend)
    899 kwds_defaults = _refine_defaults_read(
    900     dialect,
    901     delimiter,
   (...)
    908     dtype_backend=dtype_backend,
    909 )
    910 kwds.update(kwds_defaults)
--> 912 return _read(filepath_or_buffer, kwds)

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\io\parsers\readers.py:577, in _re
ad(filepath_or_buffer, kwds)
    574 _validate_names(kwds.get("names", None))
    576 # Create the parser.
--> 577 parser = TextFileReader(filepath_or_buffer, **kwds)
    579 if chunksize or iterator:
    580     return parser

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\io\parsers\readers.py:1407, in Te
xtFileReader.__init__(self, f, engine, **kwds)
   1404     self.options["has_index_names"] = kwds["has_index_names"]
   1406 self.handles: IOHandles | None = None
-> 1407 self._engine = self._make_engine(f, self.engine)

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\io\parsers\readers.py:1661, in Te
xtFileReader._make_engine(self, f, engine)
   1659     if "b" not in mode:
   1660         mode += "b"
-> 1661 self.handles = get_handle(
   1662     f,
   1663     mode,
   1664     encoding=self.options.get("encoding", None),
   1665     compression=self.options.get("compression", None),
   1666     memory_map=self.options.get("memory_map", False),
   1667     is_text=is_text,
   1668     errors=self.options.get("encoding_errors", "strict"),
   1669     storage_options=self.options.get("storage_options", None),
   1670 )
   1671 assert self.handles is not None
   1672 f = self.handles.handle

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\io\common.py:859, in get_handle(p
ath_or_buf, mode, encoding, compression, memory_map, is_text, errors, storage_options)
    854 elif isinstance(handle, str):
    855     # Check whether the filename is to be opened in binary mode.
    856     # Binary mode does not support 'encoding' and 'newline'.
```

```
    857     if ioargs.encoding and "b" not in ioargs.mode:
    858         # Encoding
--> 859         handle = open(
    860             handle,
    861             ioargs.mode,
    862             encoding=ioargs.encoding,
    863             errors=errors,
    864             newline="",
    865         )
    866     else:
    867         # Binary mode
    868         handle = open(handle, ioargs.mode)

FileNotFoundError: [Errno 2] No such file or directory: 'data.csv'
```

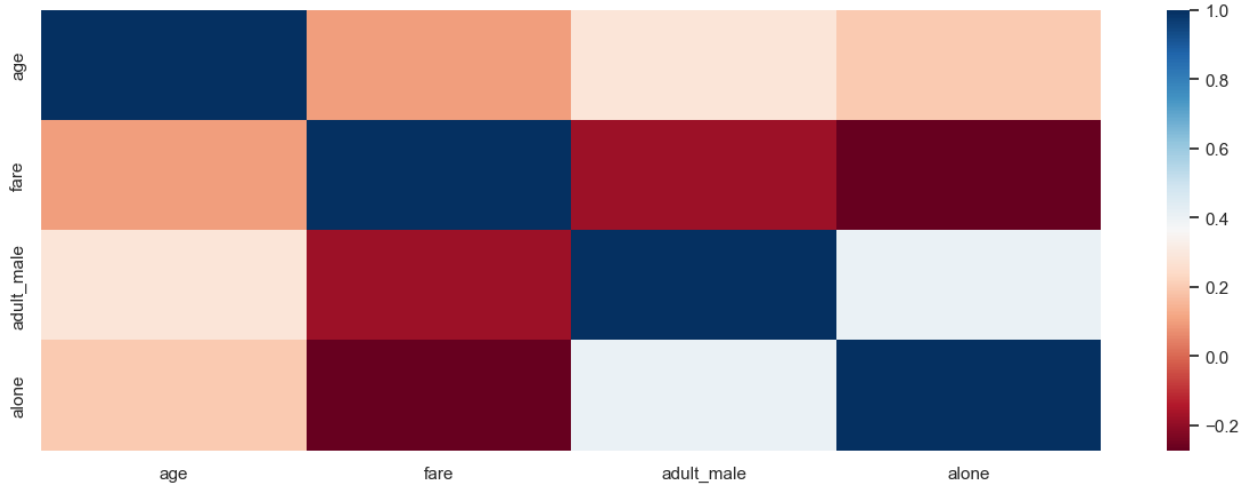
In [21]:
```
num_cols=[col for col in df.columns if df[col].dtype in [int,float]]
num_cols
```

Out[21]:
```
['age', 'fare', 'adult_male', 'alone']
```

In [22]:
```
corr=df[num_cols].corr()
```

In [23]:
```
sns.set(rc={'figure.figsize': (15,5)})
```

In [24]:
```
sns.heatmap(corr, cmap="RdBu")
plt.show()
```



In [ ]:
```

```