Data Exploration using Non-Parametric Methods ¶

Objective: To perform Data Exploration using Non-Parametric Methods on the given dataset.

In this experiment, we'll follow the following steps:

1. Load data
2. Identify variables
3. Variable analysis
4. Handling missing values
5. Handling outliers
6. Feature engineering

In [5]:

```
#Import the required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Load the dataset

In [1]:

```
import os
os.getcwd()
```

Out[1]:

```
'C:\\Users\\jitfr\\Desktop\\ML Python Lab Experiments'
```

In [3]:

```
os.chdir('C:\\Users\\jitfr\\Desktop\\ML Python Lab Experiments\\DataSets')
```

In [6]:

```
test = pd.read_csv('test.csv')
train = pd.read_csv('train.csv')

df = pd.concat([train, test])
```

In [8]:

```
df.sample(10)
```

Out[8]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fa
91	92	0.0	3	Andreasson, Mr. Paul Edvin	male	20.0	0	0	347466	7.85
21	913	NaN	3	Olsen, Master. Artur Karl	male	9.0	0	1	C 17368	3.17
360	1252	NaN	3	Sage, Master. William Henry	male	14.5	8	2	CA. 2343	69.55
317	1209	NaN	2	Rogers, Mr. Reginald Harry	male	19.0	0	0	28004	10.50
359	1251	NaN	3	Lindell, Mrs. Edvard Bengtsson (Elin Gerda Per...	female	30.0	1	0	349910	15.55
681	682	1.0	1	Hassab, Mr. Hammad	male	27.0	0	0	PC 17572	76.72
212	1104	NaN	2	Deacon, Mr. Percy William	male	17.0	0	0	S.O.C. 14879	73.50
192	193	1.0	3	Andersen-Jensen, Miss. Carla Christine Nielsine	female	19.0	1	0	350046	7.85
373	1265	NaN	2	Harbeck, Mr. William H	male	44.0	0	0	248746	13.00
67	959	NaN	1	Moore, Mr. Clarence Bloomfield	male	47.0	0	0	113796	42.40

In [9]:

```
print(df.size)
print(df.shape)
print(df.ndim)
```

```
15708
(1309, 12)
2
```

In [9]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1309 entries, 0 to 417
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  1309 non-null   int64  
 1   Survived     891 non-null    float64
 2   Pclass       1309 non-null   int64  
 3   Name         1309 non-null   object 
 4   Sex          1309 non-null   object 
 5   Age          1046 non-null   float64
 6   SibSp        1309 non-null   int64  
 7   Parch        1309 non-null   int64  
 8   Ticket       1309 non-null   object 
 9   Fare         1308 non-null   float64
 10  Cabin        295 non-null    object 
 11  Embarked     1307 non-null   object 
dtypes: float64(3), int64(4), object(5)
memory usage: 132.9+ KB
```

In [14]:

```
df.describe(include='all').T
```

Out[14]:

	count	unique	top	freq	mean	std	min	25%	50%
PassengerId	1309.0	NaN	NaN	NaN	655.0	378.020061	1.0	328.0	655.0
Survived	891.0	NaN	NaN	NaN	0.383838	0.486592	0.0	0.0	0.0
Pclass	1309.0	NaN	NaN	NaN	2.294882	0.837836	1.0	2.0	3.0
Name	1309	1307	Connolly, Miss. Kate	2	NaN	NaN	NaN	NaN	NaN
Sex	1309	2	male	843	NaN	NaN	NaN	NaN	NaN
Age	1046.0	NaN	NaN	NaN	29.881138	14.413493	0.17	21.0	28.0
SibSp	1309.0	NaN	NaN	NaN	0.498854	1.041658	0.0	0.0	0.0
Parch	1309.0	NaN	NaN	NaN	0.385027	0.86556	0.0	0.0	0.0
Ticket	1309	929	CA. 2343	11	NaN	NaN	NaN	NaN	NaN
Fare	1308.0	NaN	NaN	NaN	33.295479	51.758668	0.0	7.8958	14.4542
Cabin	295	186	C23 C25 C27	6	NaN	NaN	NaN	NaN	NaN
Embarked	1307	3	S	914	NaN	NaN	NaN	NaN	NaN

In [17]:

```
#Load the data
titanic_df=pd.read_csv("https://raw.githubusercontent.com/pandas-dev/pandas/main/do
titanic_df.head()
```

Out[17]:

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

In [18]:

```
titanic_df.shape
```

Out[18]:

(891, 12)

In [19]:

```
titan = pd.read_csv('titan.csv')
```

In [20]:

```
titan.head()
```

Out[20]:

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

In [21]:

```
print(titanic_df.size)
print(titanic_df.shape)
print(titanic_df.ndim)
```

```
10692
(891, 12)
2
```

In [3]:

```
titanic_df.tail()
```

Out[3]:

	PassengerId	Survived	Pclass	Lname	Name	Sex	Age	SibSp	Parch	Ticket
151	152	1	1	Pears	Mrs. Thomas (Edith Wearne)	female	22.0	1	0	113776
152	153	0	3	Meo	Mr. Alfonzo	male	55.5	0	0	A.5. 11206
153	154	0	3	van Billiard	Mr. Austin Blyler	male	40.5	0	2	A/5. 851
154	155	0	3	Olsen	Mr. Ole Martin	male	NaN	0	0	Fa 265302
155	156	0	1	Williams	Mr. Charles Duane	male	51.0	0	1	PC 17597

In [22]:

```
titanic_df.sample(10)
```

Out[22]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fa
453	454	1	1	Goldenberg, Mr. Samuel L	male	49.0	1	0	17453	89.10
705	706	0	2	Morley, Mr. Henry Samuel ("Mr Henry Marshall")	male	39.0	0	0	250655	26.00
486	487	1	1	Hoyt, Mrs. Frederick Maxfield (Jane Anne Forby)	female	35.0	1	0	19943	90.00
857	858	1	1	Daly, Mr. Peter Denis	male	51.0	0	0	113055	26.55
39	40	1	3	Nicola-Yarred, Miss. Jamila	female	14.0	1	0	2651	11.24
818	819	0	3	Holm, Mr. John Fredrik Alexander	male	43.0	0	0	C 7075	6.45
294	295	0	3	Mineff, Mr. Ivan	male	24.0	0	0	349233	7.89
603	604	0	3	Torber, Mr. Ernst William	male	44.0	0	0	364511	8.05
143	144	0	3	Burke, Mr. Jeremiah	male	19.0	0	0	365222	6.75
415	416	0	3	Meek, Mrs. Thomas (Annie Louise Rowley)	female	NaN	0	0	343095	8.05

Identify variable type

In [23]:

```
titanic_df.dtypes
```

Out[23]:

```
PassengerId      int64
Survived         int64
Pclass           int64
Name            object
Sex             object
Age            float64
SibSp            int64
Parch            int64
Ticket          object
Fare           float64
Cabin           object
Embarked        object
dtype: object
```

In [25]:

```
titanic_df.dtypes.nunique()
```

Out[25]:

```
3
```

In [7]:

```
titanic_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 156 entries, 0 to 155
Data columns (total 13 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  156 non-null    int64  
 1   Survived     156 non-null    int64  
 2   Pclass       156 non-null    int64  
 3   Lname        156 non-null    object 
 4   Name         156 non-null    object 
 5   Sex          156 non-null    object 
 6   Age          126 non-null    float64
 7   SibSp        156 non-null    int64  
 8   Parch        156 non-null    int64  
 9   Ticket       156 non-null    object 
 10  Fare         156 non-null    float64
 11  Cabin        31 non-null     object 
 12  Embarked     155 non-null    object 
dtypes: float64(2), int64(5), object(6)
memory usage: 16.0+ KB
```

In [26]:

```
titanic_df.describe().T
```

Out[26]:

	count	mean	std	min	25%	50%	75%	max
PassengerId	891.0	446.000000	257.353842	1.00	223.5000	446.0000	668.5	891.0000
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

Varaible Analysis: univariate, bivariate, and multivariate analysis.

- Univariate analysis is performed to find out missing and outlier values.
- Any variable may be diveded into categorical or continuous variables.
- In the case of categorical variables, we can use frequency table to understand distribution of each category.
- For continuous variables, we have to understand the central tendency and spread of the variable. It can be measured using mean, median, mode, etc. It can be visualized using box plot or histogram.

In [27]:

```
#Understand various summary statistics of the data
dataTypes =['object', 'float', 'int']
titanic_df.describe(include=dataTypes).T
```

Out[27]:

	count	unique	top	freq	mean	std	min	25%	50%	7
PassengerId	891.0	NaN	NaN	NaN	446.0	257.353842	1.0	223.5	446.0	66
Survived	891.0	NaN	NaN	NaN	0.383838	0.486592	0.0	0.0	0.0	
Pclass	891.0	NaN	NaN	NaN	2.308642	0.836071	1.0	2.0	3.0	
Name	891	891	Braund, Mr. Owen Harris	1	NaN	NaN	NaN	NaN	NaN	N
Sex	891	2	male	577	NaN	NaN	NaN	NaN	NaN	N
Age	714.0	NaN	NaN	NaN	29.699118	14.526497	0.42	20.125	28.0	3
SibSp	891.0	NaN	NaN	NaN	0.523008	1.102743	0.0	0.0	0.0	
Parch	891.0	NaN	NaN	NaN	0.381594	0.806057	0.0	0.0	0.0	
Ticket	891	681	347082	7	NaN	NaN	NaN	NaN	NaN	N
Fare	891.0	NaN	NaN	NaN	32.204208	49.693429	0.0	7.9104	14.4542	3
Cabin	204	147	B96 B98	4	NaN	NaN	NaN	NaN	NaN	N
Embarked	889	3	S	644	NaN	NaN	NaN	NaN	NaN	N

In [28]:

```
#Select the values in a categorical variable
titanic_df.select_dtypes(include='object').head()
```

Out[28]:

	Name	Sex	Ticket	Cabin	Embarked
0	Braund, Mr. Owen Harris	male	A/5 21171	NaN	S
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	PC 17599	C85	C
2	Heikkinen, Miss. Laina	female	STON/O2. 3101282	NaN	S
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	113803	C123	S
4	Allen, Mr. William Henry	male	373450	NaN	S

In [36]:

```python
titanic_df.isna().sum()/(titanic_df.shape[0])*100
```

Out[36]:

```
PassengerId     0.000000
Survived        0.000000
Pclass          0.000000
Name            0.000000
Sex             0.000000
Age            19.865320
SibSp           0.000000
Parch           0.000000
Ticket          0.000000
Fare            0.000000
Cabin          77.104377
Embarked        0.224467
dtype: float64
```

In [37]:

```python
print(titanic_df.size)
print(titanic_df.shape)
print(titanic_df.ndim)
```

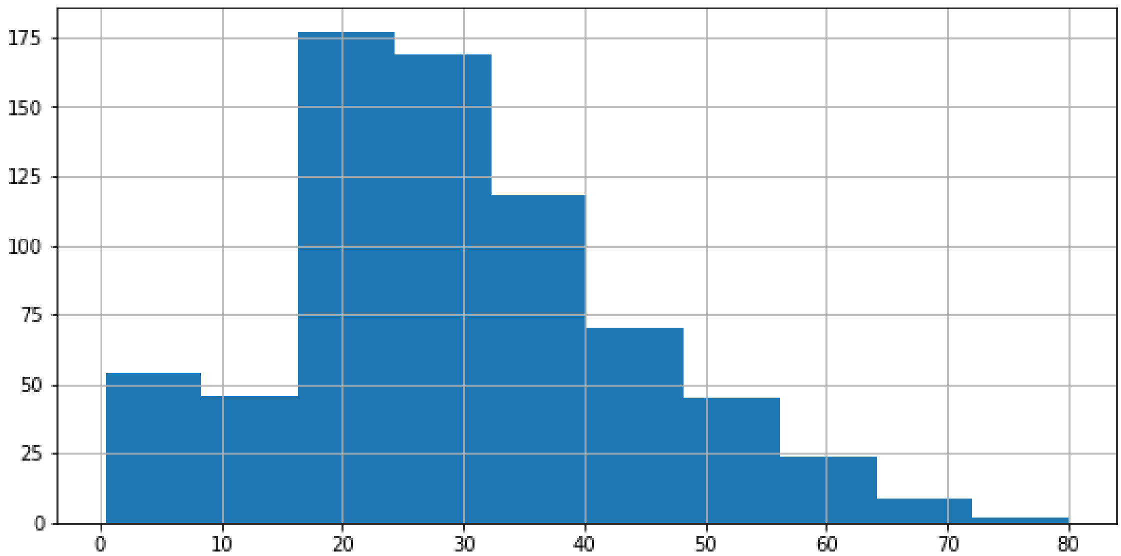
```
10692
(891, 12)
2
```

In [38]:

```python
titanic_df.Age.hist(figsize=(10,5))
```

Out[38]:

```
<AxesSubplot:>
```

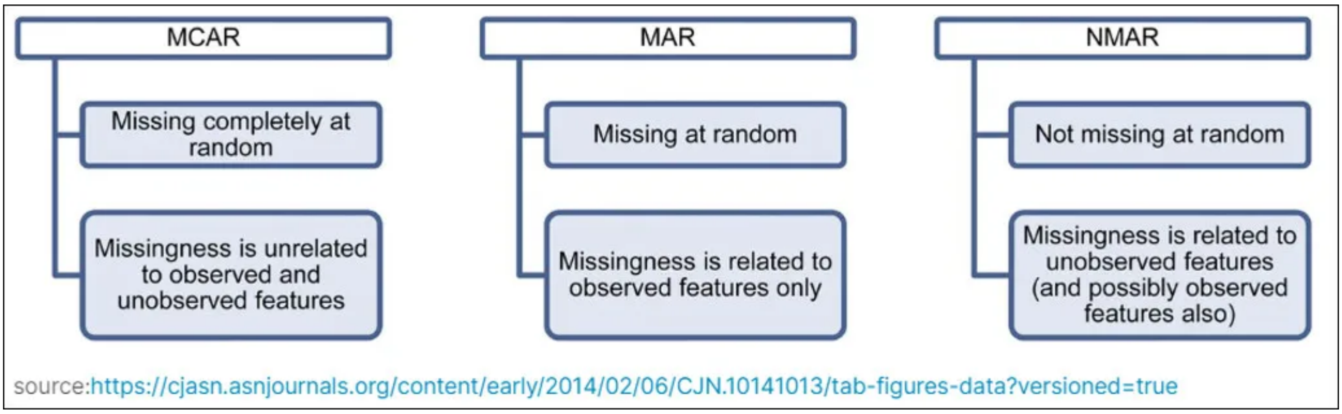


- Bivariate Analysis is used to find the relationship between two variables.
- Analysis can be performed for combination of categorical and continuous variables.
- Scatter plot is suitable for analyzing two continuous variables.

- It indicates the linear or non-linear relationship between the variables.
- Bar charts helps to understand relation between two categorical variables.

Handling Missing Values:

- Missing values in the dataset can affect model fit.
- It can lead to a biased model as the data cannot be analysed completely.
- Behavior and relationship with other variables cannot be deduced correctly.
- It can lead to wrong prediction or classification.
- Missing values may occur due to problems in data extraction or data collection, which can be categorized as
    - MCAR: Missing completely at random,
    - MAR: Missing at random, or
    - MNAR: Missing not at random.



Missing values can be treated by deletion, mean/mode/median imputation, KNN imputation, or using prediction models.

We can visually analyse the missing data using a library called as Missingno in Python.

In [32]:

```
!pip install missingno
```

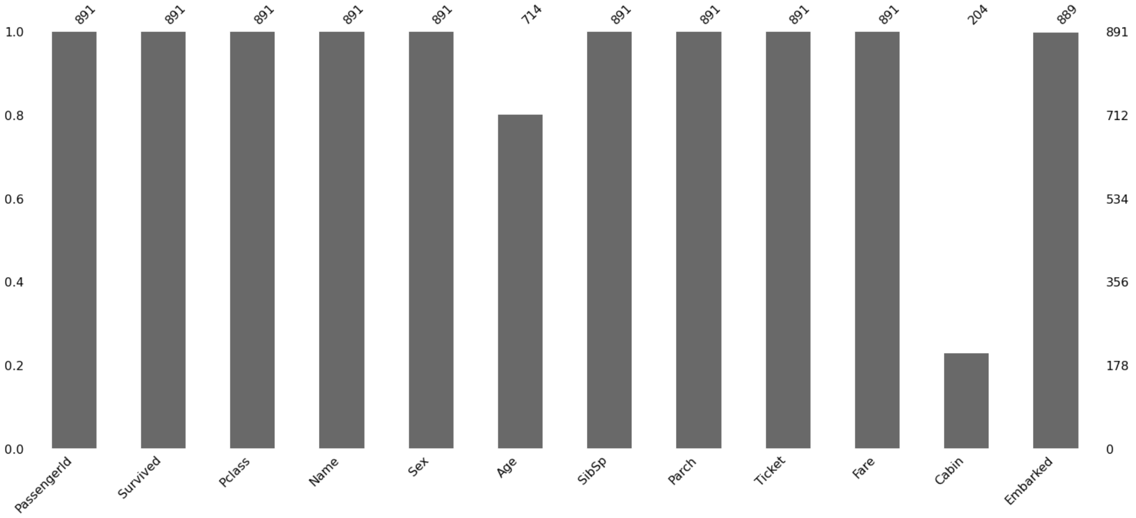
```
Collecting missingno
  Downloading missingno-0.5.1-py3-none-any.whl (8.7 kB)
Requirement already satisfied: numpy in c:\users\jitfr\anaconda3\lib\site
-packages (from missingno) (1.21.5)
Requirement already satisfied: seaborn in c:\users\jitfr\anaconda3\lib\si
te-packages (from missingno) (0.11.2)
Requirement already satisfied: scipy in c:\users\jitfr\anaconda3\lib\site
-packages (from missingno) (1.7.3)
Requirement already satisfied: matplotlib in c:\users\jitfr\anaconda3\lib
\site-packages (from missingno) (3.5.1)
Requirement already satisfied: cycler>=0.10 in c:\users\jitfr\anaconda3\l
ib\site-packages (from matplotlib->missingno) (0.11.0)
Requirement already satisfied: pillow>=6.2.0 in c:\users\jitfr\anaconda3
\lib\site-packages (from matplotlib->missingno) (9.0.1)
Requirement already satisfied: packaging>=20.0 in c:\users\jitfr\anaconda
3\lib\site-packages (from matplotlib->missingno) (21.3)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\jitfr\anacond
a3\lib\site-packages (from matplotlib->missingno) (3.0.4)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\jitfr\ana
conda3\lib\site-packages (from matplotlib->missingno) (2.8.2)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\jitfr\anacon
da3\lib\site-packages (from matplotlib->missingno) (1.3.2)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\jitfr\anacon
da3\lib\site-packages (from matplotlib->missingno) (4.25.0)
Requirement already satisfied: six>=1.5 in c:\users\jitfr\anaconda3\lib\s
ite-packages (from python-dateutil>=2.7->matplotlib->missingno) (1.16.0)
Requirement already satisfied: pandas>=0.23 in c:\users\jitfr\anaconda3\l
ib\site-packages (from seaborn->missingno) (1.4.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\jitfr\anaconda3\l
ib\site-packages (from pandas>=0.23->seaborn->missingno) (2021.3)
Installing collected packages: missingno
Successfully installed missingno-0.5.1
```

In [39]:

```
import missingno as msno
msno.bar(titanic_df)
```

Out[39]:

<AxesSubplot:>

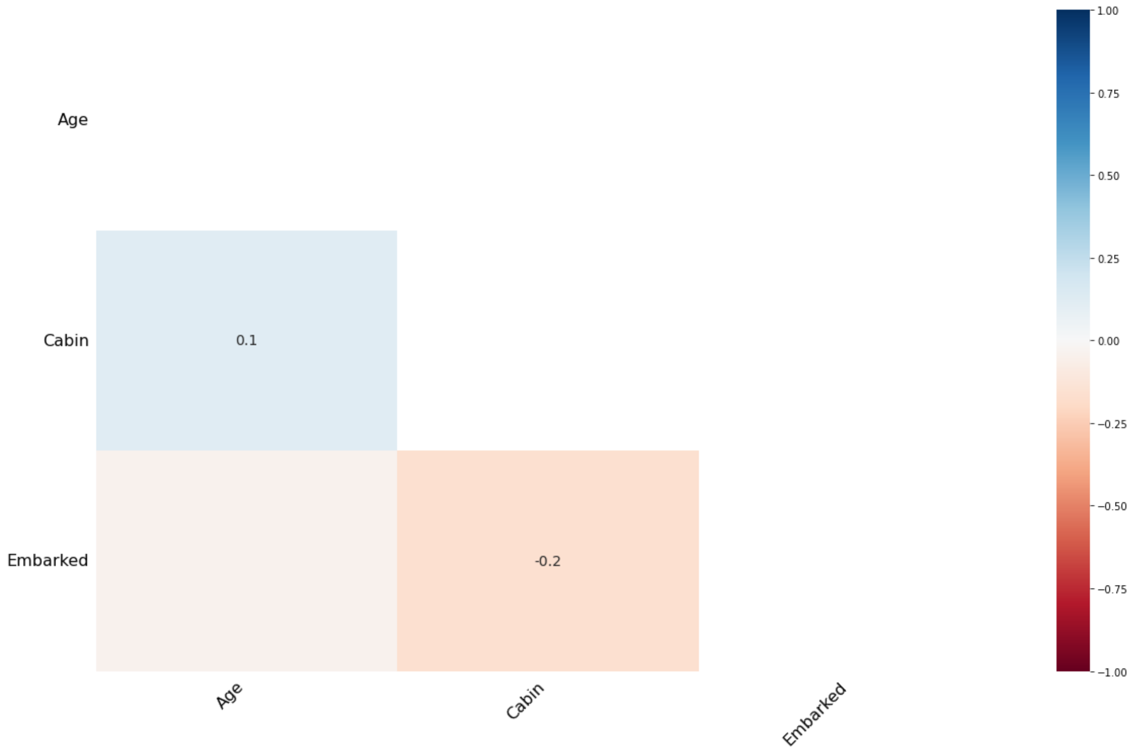


In [34]:

```
msno.heatmap(titanic_df)
```

Out[34]:

<AxesSubplot:>



In [4]:

```
np.mean(titanic_df['Age'])
```

Out[4]:

```
28.141507936507935
```

In [5]:

```
from scipy import stats
stats.mode(titanic_df['Embarked'])
```

Out[5]:

```
ModeResult(mode=array(['S'], dtype=object), count=array([110]))
```

In [6]:

```
titanic_df['Age'].fillna(29,inplace=True)
titanic_df['Embarked'].fillna("S", inplace=True)
```

Handling Outliers

- Outliers can occur naturally in a data or due to data entry errors.
- They can drastically change the results of the data analysis and statistical modeling.
- Outliers are easily detected by visualization methods, like box-plot, histogram, and scatter plot.
- Outliers are handled like missing values by deleting observations, transforming them, binning or grouping them, treating them as a separate group, or imputing values.

In [19]:

```
titanic_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 156 entries, 0 to 155
Data columns (total 13 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  156 non-null    int64  
 1   Survived     156 non-null    int64  
 2   Pclass       156 non-null    int64  
 3   Lname        156 non-null    object 
 4   Name         156 non-null    object 
 5   Sex          156 non-null    object 
 6   Age          126 non-null    float64
 7   SibSp        156 non-null    int64  
 8   Parch        156 non-null    int64  
 9   Ticket       156 non-null    object 
 10  Fare         156 non-null    float64
 11  Cabin        31 non-null     object 
 12  Embarked     155 non-null    object 
dtypes: float64(2), int64(5), object(6)
memory usage: 16.0+ KB
```

In [7]:

```
import plotly.express as px
fig = px.box(titanic_df,x='Survived',y='Age', color='Pclass')
fig.show()
```

In [8]:

```
px.box(titanic_df, y='Age')
px.box(titanic_df,x='Survived',y='Fare', color='Pclass')
```

Feature Engineering:

- Feature engineering is the process of extracting more information from existing data.
- Feature selection also can be part of it.
- Two common techniques of feature engineering are variable transformation and variable creation.
- In variable transformation existing variable is transformed using certain functions.
- For example, a number can be replaced by its logarithmic value.
- Another technique is to create a new variable from the existing variable.
- For example, breaking the date field in the format of dd/mm/yy to date, month and year columns.

In [9]:

```
titanic_copy = titanic_df.copy()
```

In [10]:

```
#variable transformation
titanic_copy['Embarked'].replace({'S':0,'Q':1,'C':2}, inplace=True)
```

In [11]:

```
#Convert boolean to integer
titanic_copy['Survived']=titanic_copy['Survived'].astype(int)
```

In [12]:

```
titanic_copy.sample(10)
```

Out[12]:

	PassengerId	Survived	Pclass	Lname	Name	Sex	Age	SibSp	Parch	Tick
0	1	0	3	Braund	Mr. Owen Harris	male	22.0	1	0	A 2117
95	96	0	3	Shorney	Mr. Charles Joseph	male	29.0	0	0	37491
23	24	1	1	Sloper	Mr. William Thompson	male	28.0	0	0	11378
3	4	1	1	Futrelle	Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	11380
61	62	1	1	Icard	Miss. Amelie	female	38.0	0	0	11357
111	112	0	3	Zabour	Miss. Hileni	female	14.5	1	0	266
70	71	0	2	Jenkin	Mr. Stephen Curnow	male	32.0	0	0	C./ 331
30	31	0	1	Uruchurtu	Don. Manuel E	male	40.0	0	0	P 1760
11	12	1	1	Bonnell	Miss. Elizabeth	female	58.0	0	0	11378
132	133	0	3	Robins	Mrs. Alexander A (Grace Charity Laury)	female	47.0	1	0	A/ 333

- Two other data transformation techniques are
- encoding categorical variables and scaling continuous variables to normalize the data.
- This depends on the model that is used for evaluation, as some models accept categorical variables.
- Irrelevant features can decrease the accuracy of the model.
- Feature selection can be done automatically or manually.

- A correlation matrix is used to visualize how the features are related to each other or with the target variable.

In [52]:

```
1 titanic_copy.corr()
```

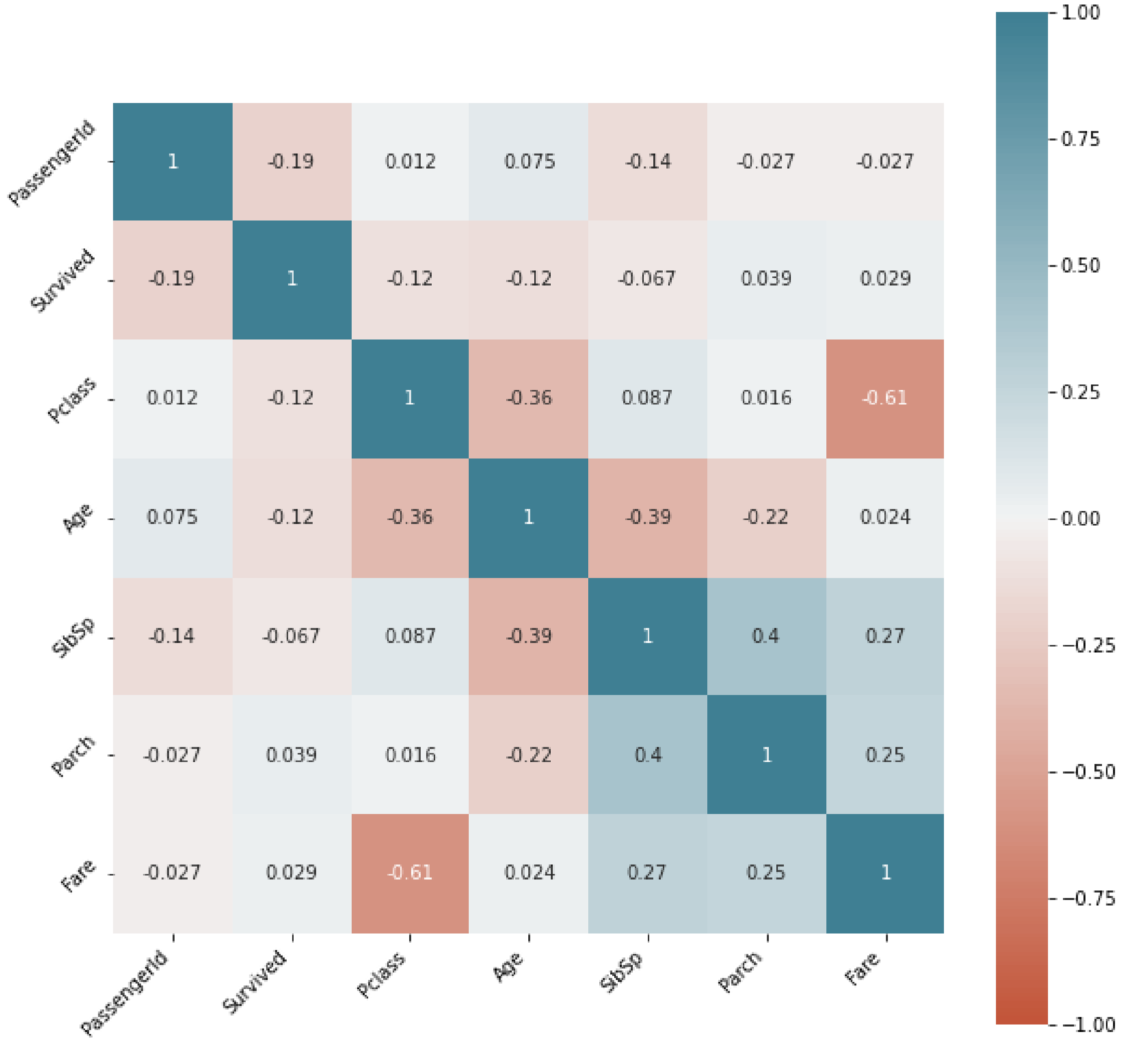
Out[52]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
PassengerId	1.000000	-0.192991	0.012208	0.065909	-0.136420	-0.027243	-0.027122
Survived	-0.192991	1.000000	-0.116340	-0.104230	-0.066943	0.039435	0.029343
Pclass	0.012208	-0.116340	1.000000	-0.326655	0.087420	0.016491	-0.607256
Age	0.065909	-0.104230	-0.326655	1.000000	-0.383121	-0.211423	0.019475
SibSp	-0.136420	-0.066943	0.087420	-0.383121	1.000000	0.399040	0.271997
Parch	-0.027243	0.039435	0.016491	-0.211423	0.399040	1.000000	0.254822
Fare	-0.027122	0.029343	-0.607256	0.019475	0.271997	0.254822	1.000000
Embarked	-0.064032	0.072072	-0.115187	0.037916	-0.085307	-0.087342	0.112638

In [37]:

```
plt.figure(figsize=(10,10))
corr = titanic_df.corr()
ax = sns.heatmap(
    corr,
    vmin=-1, vmax=1, center=0,
    cmap=sns.diverging_palette(20, 220, n=200),
    square=True, annot=True
)
ax.set_xticklabels(
    ax.get_xticklabels(),
    rotation=45,
    horizontalalignment='right'
)
ax.set_yticklabels(
    ax.get_yticklabels(),
    rotation=45,

);
```



Ref: https://www.kaggle.com/code/krrai77/exploratory-data-analysis-and-visualization/data [(https://www.kaggle.com/code/krrai77/exploratory-data-analysis-and-visualization/data)](https://www.kaggle.com/code/krrai77/exploratory-data-analysis-and-visualization/data)

In [ ]:

```

```