# 02. 데이터 전처리

## 학습 내용

- 데이터 확인 및 결측치 처리를 수행한다.
- 범주형, 수치형 변수에 대한 값을 확인해 본다.
- 자료형 변환 및 수치로 변환(라벨 인코딩)을 알아본다.

## 나이와 승선항을 결측치 처리 후, 확인해 보자.

In [4]:

```
## 설치가 안되어 있을 경우, 설치
# !pip install missingno
```

In [5]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import missingno as msno   # No module named 'missingno' 발생시, 위의 pip install missingno 설치 필
```

# 01. 데이터 불러오기

- 데이터 탐색

In [7]:

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

In [8]:

```
print(train.shape, test.shape)    # 데이터의 행과열
```

```
(891, 12) (418, 11)
```

In [9]:

```
## 데이터 확인
train.head()
```

Out[9]:

| | 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 |

## 02 하나 하나 열을 불러와서 5행씩 확인하기

In [10]:

```
# 만약 전체 열이 확인 안 될 때,
for col in train.columns:
    print("column : ", col)
    print(train[col].head())
    print()
```

```
column :  PassengerId
0    1
1    2
2    3
3    4
4    5
Name: PassengerId, dtype: int64

column :  Survived
0    0
1    1
2    1
3    1
4    0
Name: Survived, dtype: int64

column :  Pclass
0    3
1    1
2    3
3    1
4    3
Name: Pclass, dtype: int64

column :  Name
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
Name: Name, dtype: object

column :  Sex
0      male
1    female
2    female
3    female
4      male
Name: Sex, dtype: object

column :  Age
0    22.0
1    38.0
2    26.0
3    35.0
4    35.0
Name: Age, dtype: float64

column :  SibSp
0    1
1    1
2    0
```

```
3    1
4    0
Name: SibSp, dtype: int64

column :  Parch
0    0
1    0
2    0
3    0
4    0
Name: Parch, dtype: int64

column :  Ticket
0           A/5 21171
1            PC 17599
2    STON/O2. 3101282
3              113803
4              373450
Name: Ticket, dtype: object

column :  Fare
0     7.2500
1    71.2833
2     7.9250
3    53.1000
4     8.0500
Name: Fare, dtype: float64

column :  Cabin
0     NaN
1     C85
2     NaN
3    C123
4     NaN
Name: Cabin, dtype: object

column :  Embarked
0    S
1    C
2    S
3    S
4    S
Name: Embarked, dtype: object
```

## 03 데이터 요약

In [11]:

```
train.describe()
```

Out[11]:

| | PassengerId | Survived | Pclass | Age | SibSp | Parch | Fare |
|---|---|---|---|---|---|---|---|
| **count** | 891.000000 | 891.000000 | 891.000000 | 714.000000 | 891.000000 | 891.000000 | 891.000000 |
| **mean** | 446.000000 | 0.383838 | 2.308642 | 29.699118 | 0.523008 | 0.381594 | 32.204208 |
| **std** | 257.353842 | 0.486592 | 0.836071 | 14.526497 | 1.102743 | 0.806057 | 49.693429 |
| **min** | 1.000000 | 0.000000 | 1.000000 | 0.420000 | 0.000000 | 0.000000 | 0.000000 |
| **25%** | 223.500000 | 0.000000 | 2.000000 | 20.125000 | 0.000000 | 0.000000 | 7.910400 |
| **50%** | 446.000000 | 0.000000 | 3.000000 | 28.000000 | 0.000000 | 0.000000 | 14.454200 |
| **75%** | 668.500000 | 1.000000 | 3.000000 | 38.000000 | 1.000000 | 0.000000 | 31.000000 |
| **max** | 891.000000 | 1.000000 | 3.000000 | 80.000000 | 8.000000 | 6.000000 | 512.329200 |

## 데이터 결측치 확인

In [12]:

```
train.info()
```

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

## 결측치 확인

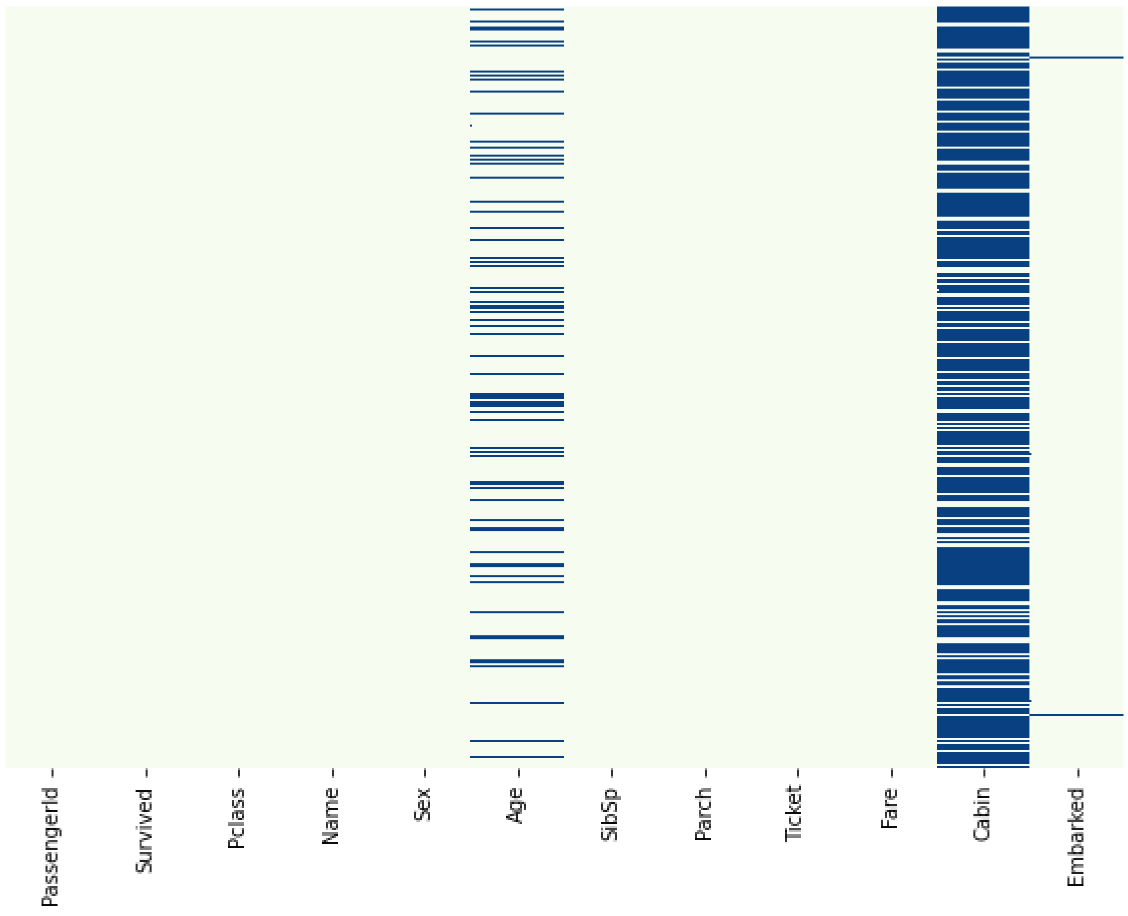
- figsize로 크기 설정
- seaborn의 heatmap 이용 결측치 확인 (cbar : colorbar, cmap : 색 지정, yticklabels : y축 유무)

In [15]:

```
plt.figure(figsize=(10,7)) # cmpa : summer도 해보기
sns.heatmap(train.isnull(), yticklabels=False, cbar=False, cmap="GnBu")  # cbar : colorbar를 그리지
```

Out[15]:

```
<AxesSubplot:>
```



## 04 범주형 변수, 수치형 변수

In [19]:

```
len(train.columns)
```

Out[19]:

```
12
```

## 수치형 변수 살펴보기

In [20]:

```
num_cols = [col for col in train.columns[:12] if train[col].dtype in ['int64', 'float64'] ]
train[num_cols].describe()
```

Out[20]:

| | PassengerId | Survived | Pclass | Age | SibSp | Parch | Fare |
|---|---|---|---|---|---|---|---|
| **count** | 891.000000 | 891.000000 | 891.000000 | 714.000000 | 891.000000 | 891.000000 | 891.000000 |
| **mean** | 446.000000 | 0.383838 | 2.308642 | 29.699118 | 0.523008 | 0.381594 | 32.204208 |
| **std** | 257.353842 | 0.486592 | 0.836071 | 14.526497 | 1.102743 | 0.806057 | 49.693429 |
| **min** | 1.000000 | 0.000000 | 1.000000 | 0.420000 | 0.000000 | 0.000000 | 0.000000 |
| **25%** | 223.500000 | 0.000000 | 2.000000 | 20.125000 | 0.000000 | 0.000000 | 7.910400 |
| **50%** | 446.000000 | 0.000000 | 3.000000 | 28.000000 | 0.000000 | 0.000000 | 14.454200 |
| **75%** | 668.500000 | 1.000000 | 3.000000 | 38.000000 | 1.000000 | 0.000000 | 31.000000 |
| **max** | 891.000000 | 1.000000 | 3.000000 | 80.000000 | 8.000000 | 6.000000 | 512.329200 |

## 범주형 변수 살펴보기

In [21]:

```
cat_cols = [col for col in train.columns[:12] if train[col].dtype in ['O'] ]
train[cat_cols].describe()
```

Out[21]:

| | Name | Sex | Ticket | Cabin | Embarked |
|---|---|---|---|---|---|
| **count** | 891 | 891 | 891 | 204 | 889 |
| **unique** | 891 | 2 | 681 | 147 | 3 |
| **top** | Strom, Mrs. Wilhelm (Elna Matilda Persson) | male | 347082 | B96 B98 | S |
| **freq** | 1 | 577 | 7 | 4 | 644 |

## 05 unique()한 값을 확인해 보기

In [23]:

```
for col in cat_cols:
    uniq = np.unique(train[col].astype(str))
    print("colname : {}, uniq : {}".format(col, uniq), end="₩n₩n")
```

```
colname : Name, uniq : ['Abbing, Mr. Anthony' 'Abbott, Mr. Rossmore Edward'
 'Abbott, Mrs. Stanton (Rosa Hunt)' 'Abelson, Mr. Samuel'
 'Abelson, Mrs. Samuel (Hannah Wizosky)' 'Adahl, Mr. Mauritz Nils Martin'
 'Adams, Mr. John' 'Ahlin, Mrs. Johan (Johanna Persdotter Larsson)'
 'Aks, Mrs. Sam (Leah Rosen)' 'Albimona, Mr. Nassef Cassem'
 'Alexander, Mr. William' 'Alhomaki, Mr. Ilmari Rudolf' 'Ali, Mr. Ahmed'
 'Ali, Mr. William' 'Allen, Miss. Elisabeth Walton'
 'Allen, Mr. William Henry' 'Allison, Master. Hudson Trevor'
 'Allison, Miss. Helen Loraine'
 'Allison, Mrs. Hudson J C (Bessie Waldo Daniels)'
 'Allum, Mr. Owen George'
 'Andersen-Jensen, Miss. Carla Christine Nielsine' 'Anderson, Mr. Harry'
 'Andersson, Master. Sigvard Harald Elias'
 'Andersson, Miss. Ebba Iris Alfrida' 'Andersson, Miss. Ellis Anna Maria'
 'Andersson, Miss. Erna Alexandra' 'Andersson, Miss. Ingeborg Constanzia'
 'Andersson, Miss. Sigrid Elisabeth' 'Andersson, Mr. Anders Johan'
 'Andersson, Mr. August Edvard ("Wennerstrom")'
 'Andersson, Mrs. Anders Johan (Alfrida Konstantia Brogren)'
 'Andreasson, Mr. Paul Edvin' 'Andrew, Mr. Edgardo Samuel'
```

## 나이에 대해 살펴보자

- plt.subplots(행, 열, figsize=(크기지정))

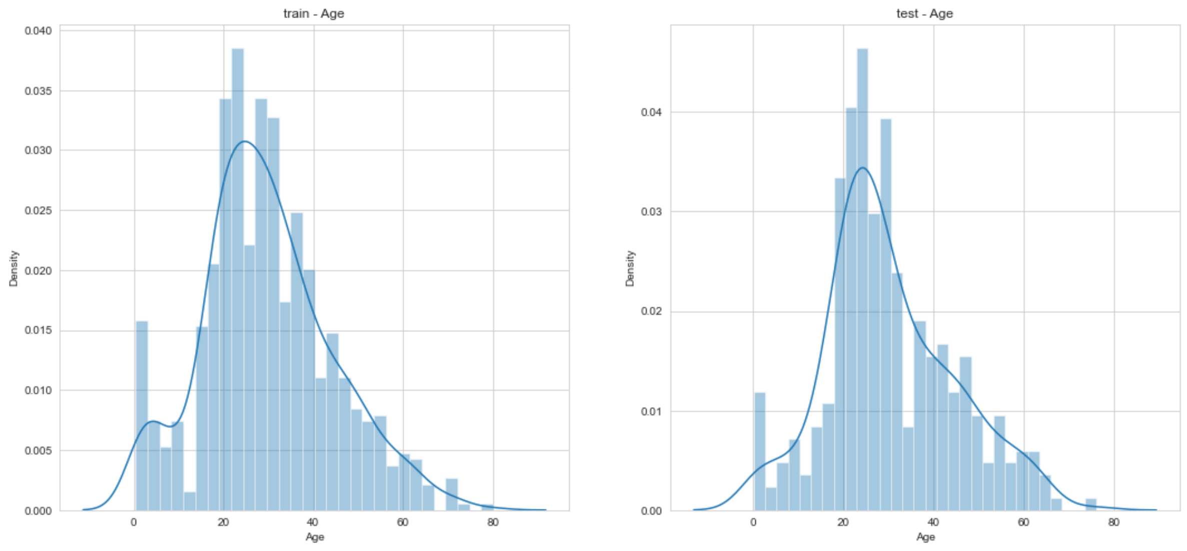
In [31]:

```
f,ax=plt.subplots(1,2,figsize=(18,8))

# 첫번째 그래프
sns.distplot(train['Age'].dropna(), bins=30, ax=ax[0])
ax[0].set_title('train - Age')

# 두번째 그래프
sns.distplot(test['Age'].dropna(), bins=30, ax=ax[1])
ax[1].set_title('test - Age')
plt.show()
```

```
C:\Users\toto\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarni
ng: `distplot` is a deprecated function and will be removed in a future version. Ple
ase adapt your code to use either `displot` (a figure-level function with similar fl
exibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\toto\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarni
ng: `distplot` is a deprecated function and will be removed in a future version. Ple
ase adapt your code to use either `displot` (a figure-level function with similar fl
exibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



## 06 결측치 처리 - 나이, 요금

- 나이는 평균값으로 처리하자.
- 결측치 값을 채우기 - usage : data['열이름'].fillna(값)

In [32]:

```
train['Age'] = train['Age'].fillna(train['Age'].mean())
test['Age'] = test['Age'].fillna(test['Age'].mean())
```

In [33]:

```
## 요금 결측치 처리 - 실습
test['Fare'] = test['Fare'].fillna(test['Fare'].mean())
```

In [34]:

```
print(train.isnull().sum())
print(test.isnull().sum())
```

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age              0
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
PassengerId      0
Pclass           0
Name             0
Sex              0
Age              0
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          327
Embarked         0
dtype: int64
```

## 07 결측치 처리 - Embarked(승선항)

- 가장 많이 나온 값으로 결측치 처리를 하자
- 범주(구분,종류)별 데이터 개수 => [Syntax] 데이터셋명['컬럼명'].value_counts()

In [35]:

```
val_Embarked = train['Embarked'].value_counts()
val_Embarked
```

Out[35]:

```
S    644
C    168
Q     77
Name: Embarked, dtype: int64
```

In [36]:

```
val_Embarked.index[0]   #  행 이름 첫번째
```

Out[36]:

```
'S'
```

In [37]:

```
train['Embarked'] = train['Embarked'].fillna('S')
```

In [38]:

```
print(train.isnull().sum())
print(test.isnull().sum())
```

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age              0
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         0
dtype: int64
PassengerId      0
Pclass           0
Name             0
Sex              0
Age              0
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          327
Embarked         0
dtype: int64
```

## 데이터 전처리

In [39]:

```
train.info()
```

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

In [40]:

```
print( train['Sex'].value_counts() )
print( train['Embarked'].value_counts() )
```

```
male      577
female    314
Name: Sex, dtype: int64
S    646
C    168
Q     77
Name: Embarked, dtype: int64
```

## 08. 자료형 변환 및 숫자로 변경(라벨 인코딩)

- 데이터 자료형 변환
- 데이터.astype(변환될 자료형명)

In [41]:

```
train['Sex'] = train['Sex'].map( {'female': 0, 'male': 1} ).astype(int)
test['Sex'] = test['Sex'].map( {'female': 0, 'male': 1} ).astype(int)

train['Embarked'] = train['Embarked'].map( {'S': 0, 'C': 1, 'Q': 2} ).astype(int)
test['Embarked']= test['Embarked'].map( {'S': 0, 'C': 1, 'Q': 2} ).astype(int)
```

In [42]:

```
## 나이에 대한 int 처리
train['Age'] = train['Age'].astype('int')
test['Age'] = test['Age'].astype('int')
```

In [43]:

```python
print(train.columns)
print(train.info())
```

```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object 
 4   Sex          891 non-null    int32  
 5   Age          891 non-null    int32  
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object 
 11  Embarked     891 non-null    int32  
dtypes: float64(1), int32(3), int64(5), object(3)
memory usage: 73.2+ KB
None
```

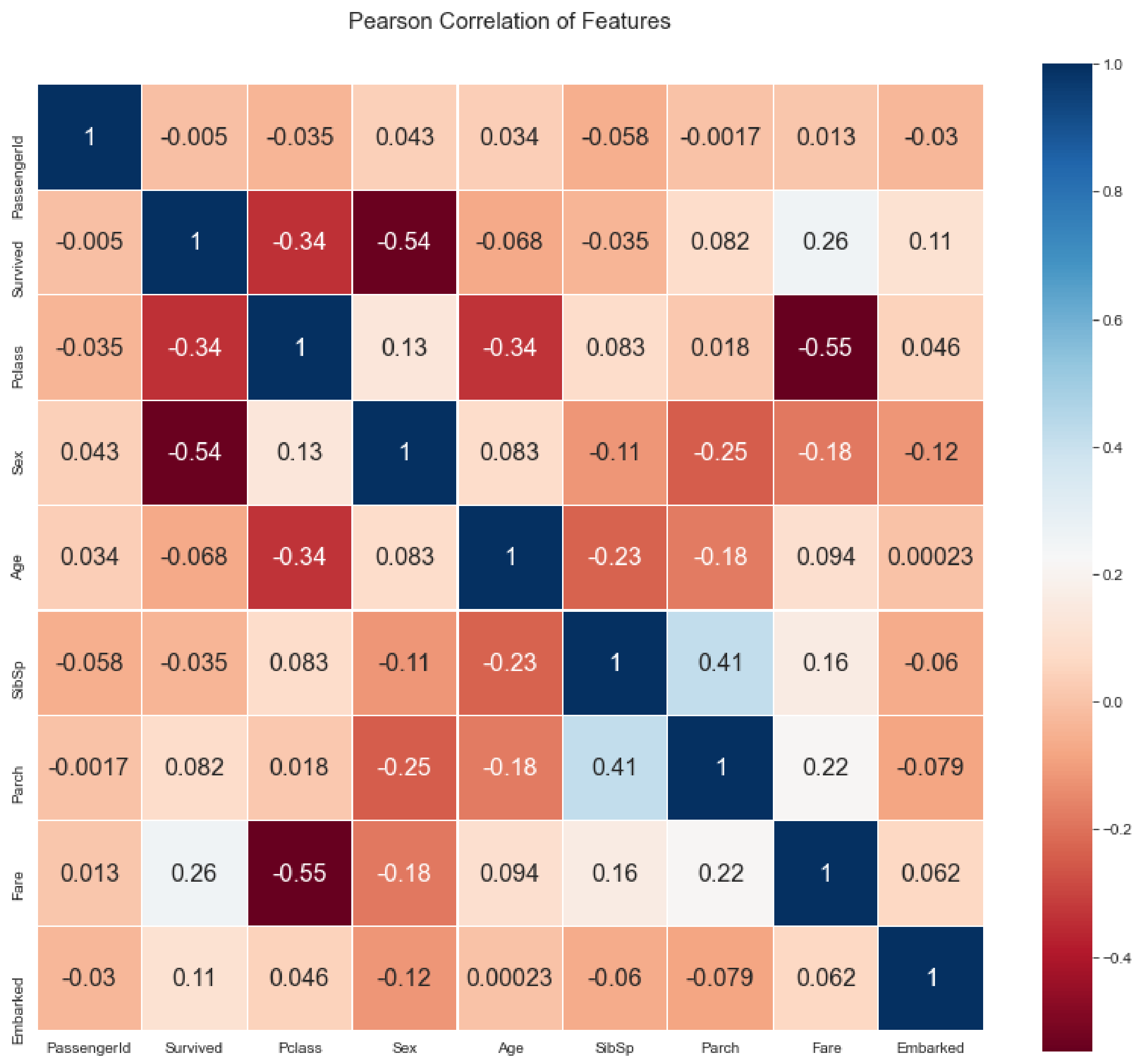
## 컬럼과 컬럼 사이의 관계 확인(상관계수 Heatmap)

In [45]:

```
colormap = plt.cm.RdBu
plt.figure(figsize=(14, 12))
plt.title('Pearson Correlation of Features', y=1.05, size=15)
sns.heatmap(train.corr(), linewidths=0.1, vmax=1.0,
            square=True, cmap=colormap, linecolor='white', annot=True, annot_kws={"size": 16})
```

Out[45]:

```
<AxesSubplot:title={'center':'Pearson Correlation of Features'}>
```



| | PassengerId | Survived | Pclass | Sex | Age | SibSp | Parch | Fare | Embarked |
|---|---|---|---|---|---|---|---|---|---|
| PassengerId | 1 | -0.005 | -0.035 | 0.043 | 0.034 | -0.058 | -0.0017 | 0.013 | -0.03 |
| Survived | -0.005 | 1 | -0.34 | -0.54 | -0.068 | -0.035 | 0.082 | 0.26 | 0.11 |
| Pclass | -0.035 | -0.34 | 1 | 0.13 | -0.34 | 0.083 | 0.018 | -0.55 | 0.046 |
| Sex | 0.043 | -0.54 | 0.13 | 1 | 0.083 | -0.11 | -0.25 | -0.18 | -0.12 |
| Age | 0.034 | -0.068 | -0.34 | 0.083 | 1 | -0.23 | -0.18 | 0.094 | 0.00023 |
| SibSp | -0.058 | -0.035 | 0.083 | -0.11 | -0.23 | 1 | 0.41 | 0.16 | -0.06 |
| Parch | -0.0017 | 0.082 | 0.018 | -0.25 | -0.18 | 0.41 | 1 | 0.22 | -0.079 |
| Fare | 0.013 | 0.26 | -0.55 | -0.18 | 0.094 | 0.16 | 0.22 | 1 | 0.062 |
| Embarked | -0.03 | 0.11 | 0.046 | -0.12 | 0.00023 | -0.06 | -0.079 | 0.062 | 1 |

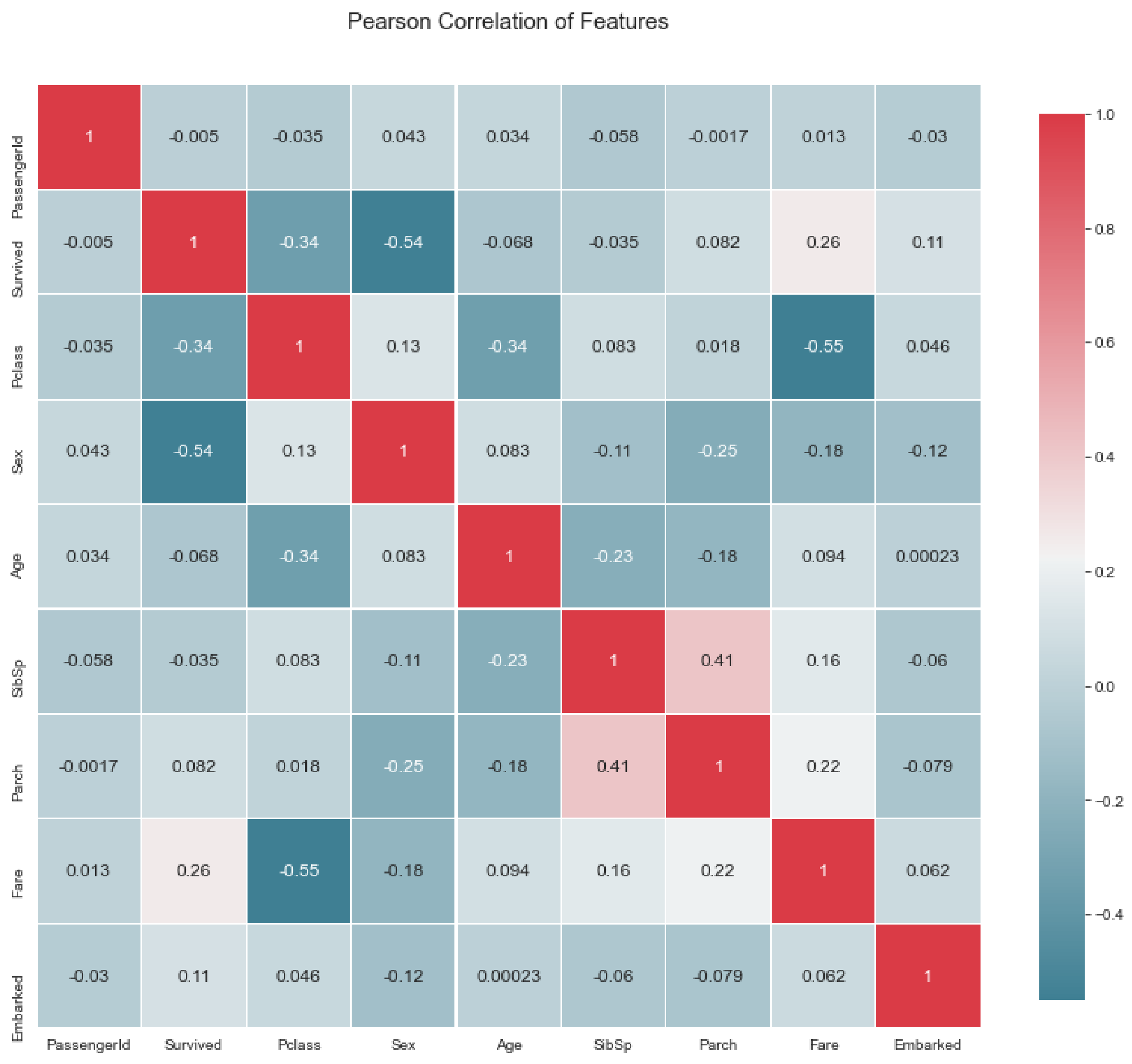
In [46]:

```
#correlation heatmap of dataset
def correlation_heatmap(df):
    _ , ax = plt.subplots(figsize =(14, 12))
    colormap = sns.diverging_palette(220, 10, as_cmap = True)

    _ = sns.heatmap(
        df.corr(),
        cmap = colormap,
        square=True,
        cbar_kws={'shrink':.9 },
        ax=ax,
        annot=True,
        linewidths=0.1,vmax=1.0, linecolor='white',
        annot_kws={'fontsize':12 }
    )

    plt.title('Pearson Correlation of Features', y=1.05, size=15)

correlation_heatmap(train)
```



## 2-1 모델 만들고 제출해 보기

- 모델을 생성 후, 학습

- 그리고 예측을 수행 후, 제출한다.

In [53]:

```
# 'Name', 'Ticket' =>  문자포함
sel = ['Pclass', 'Sex', 'Age', 'SibSp', 'SibSp','Parch', 'Embarked' ]

# 학습에 사용될 데이터 준비 X_train, y_train
X_train = train[sel]
y_train = train['Survived']
X_test = test[sel]
```

In [54]:

```
from sklearn.linear_model import LogisticRegression
log_r = LogisticRegression()
log_r.fit(X_train, y_train)
```

Out[54]:

```
LogisticRegression()
```

In [55]:

```
# 예측
pred = log_r.predict(X_test)
pred[:15]
```

Out[55]:

```
array([0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1], dtype=int64)
```

In [56]:

```
test_passengerId = test['PassengerId']
pred = pred.astype(int)
df_pred = pd.DataFrame({'PassengerID':test_passengerId, 'Survived':pred})
df_pred.to_csv("log_second_model.csv", index=False)
```

## 2-2 모델 개선 - 'Fare'변수 추가

- 모델을 생성 후, 학습
- 그리고 예측을 수행 후, 제출한다.

In [57]:

```
# 'Name', 'Ticket' =>  문자포함으로 제외
sel = ['Pclass', 'Sex', 'Age', 'SibSp', 'SibSp','Parch', 'Embarked', 'Fare' ]

# 학습에 사용될 데이터 준비 X_train, y_train
X_train = train[sel]
y_train = train['Survived']
X_test = test[sel]
```

In [58]:

```
from sklearn.linear_model import LogisticRegression
log_r = LogisticRegression()
log_r.fit(X_train, y_train)

# 예측
pred = log_r.predict(X_test)
pred[:15]
```

Out[58]:

```
array([0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1], dtype=int64)
```

In [59]:

```
test_pid = test['PassengerId']
pred = pred.astype(int)
df_pred = pd.DataFrame({'PassengerID':test_pid, 'Survived':pred})
df_pred.to_csv("log_third_model.csv", index=False)
```

## REF

seaborn heatmap cmap : https://pod.hatenablog.com/entry/2018/09/20/212527 (https://pod.hatenablog.com/entry/2018/09/20/212527)
seaborn set_style : https://www.codecademy.com/articles/seaborn-design-i (https://www.codecademy.com/articles/seaborn-design-i)