타이타닉 데이터 살펴보기

학습 내용

- 데이터 확인 및 전처리
- 데이터 시각화 해 보기
- 머신러닝 모델 만들고 제출하기

In [8]:

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

```
Collecting missingno
  Downloading missingno-0.4.2-py3-none-any.whl (9.7 kB)
Requirement already satisfied: numpy in c:₩users₩toto₩anaconda3₩lib₩site-packages (f
rom missingno) (1.19.2)
Requirement already satisfied: seaborn in c:₩users₩toto₩anaconda3₩lib₩site-packages
(from missingno) (0.11.0)
Requirement already satisfied: scipy in c:₩users₩toto₩anaconda3₩lib₩site-packages (f
rom missingno) (1.5.2)
Requirement already satisfied: matplotlib in c:₩users₩toto₩anaconda3₩lib₩site-packag
es (from missingno) (3.3.2)
Requirement already satisfied: pandas>=0.23 in c:₩users₩toto₩anaconda3₩lib₩site-pack
ages (from seaborn->missingno) (1.1.3)
Requirement already satisfied: certifi>=2020.06.20 in c:₩users₩toto₩anaconda3₩lib₩si
te-packages (from matplotlib->missingno) (2020.6.20)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in c:₩users
₩toto₩anaconda3₩lib₩site-packages (from matplotlib->missingno) (2.4.7)
Requirement already satisfied: pillow>=6.2.0 in c:₩users₩toto₩anaconda3₩lib₩site-pac
kages (from matplotlib->missingno) (8.0.1)
Requirement already satisfied: python-dateutil>=2.1 in c:₩users₩toto₩anaconda3₩lib₩s
ite-packages (from matplotlib->missingno) (2.8.1)
Requirement already satisfied: kiwisolver>=1.0.1 in c:₩users₩toto₩anaconda3₩lib₩site
-packages (from matplotlib->missingno) (1.3.0)
Requirement already satisfied: cycler>=0.10 in c:₩users₩toto₩anaconda3₩lib₩site-pack
ages (from matplotlib->missingno) (0.10.0)
Requirement already satisfied: pytz>=2017.2 in c:₩users₩toto₩anaconda3₩lib₩site-pack
ages (from pandas>=0.23->seaborn->missingno) (2020.1)
Requirement already satisfied: six>=1.5 in c:₩users₩toto₩anaconda3₩lib₩site-packages
(from python-dateutil>=2.1->matplotlib->missingno) (1.15.0)
Installing collected packages: missingno
Successfully installed missingno-0.4.2
```

In [9]:

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

01. EDA(탐색적 데이터 탐색)

- 데이터에 익숙해 지기
- 데이터 자료형에 대해 알아가기
- 데이터 컬럼명 알아보기

1-1 나이와 승선항을 결측치 처리 후, 확인해 보자.

In [41]:

```
train = pd.read_csv("../data/titanic/train.csv")
test = pd.read_csv("../data/titanic/test.csv")
```

In [42]:

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

(891, 12) (418, 11)

In [43]:

```
## 데이터 확인
train.head()
```

Out[43]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	C
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 [44]:

```
# 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/02. 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
```

데이터 요약

In [45]:

```
train.describe()
```

Out[45]:

	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 [46]:

```
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

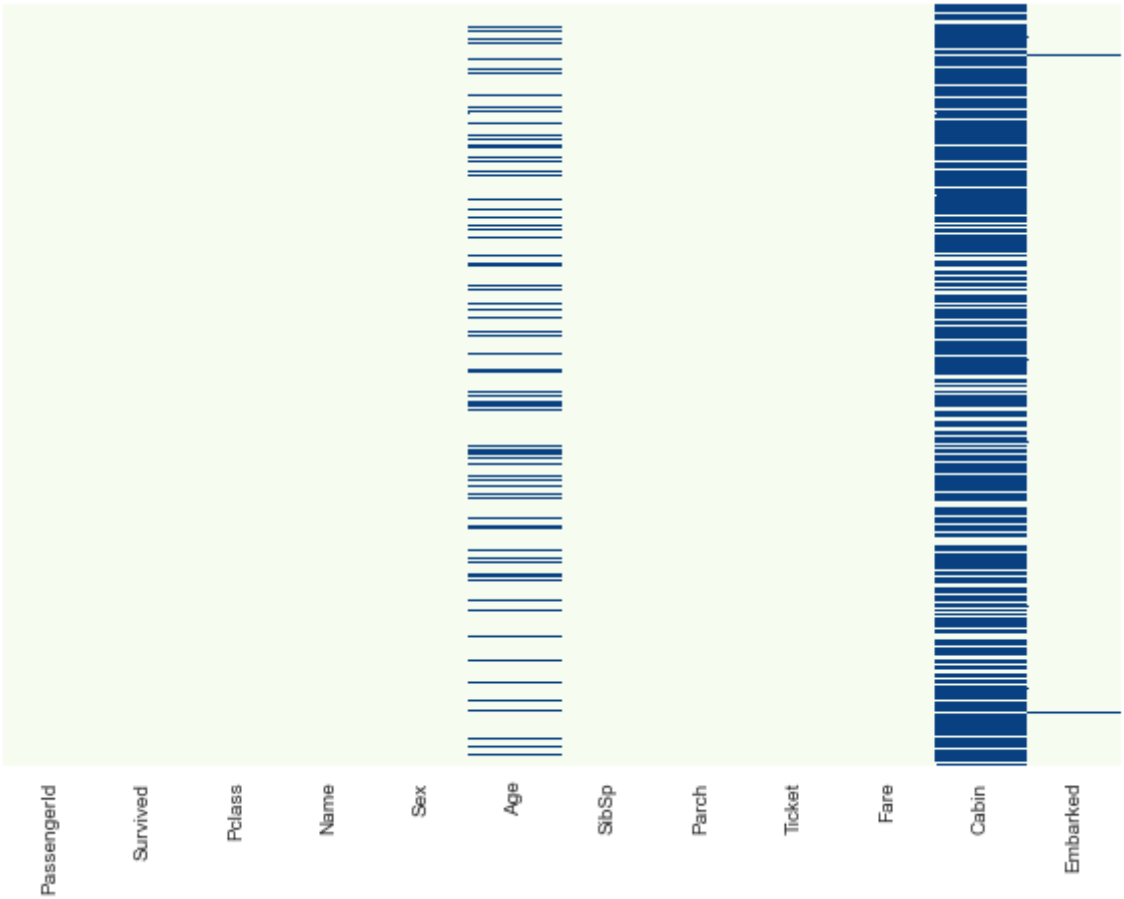
- figsize로 크기 설정
- seaborn의 heatmap 이용 결측치 확인 (cbar : colorbar, cmap : 색 지정, yticklabels : y축 유무)

In [47]:

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

Out[47]:

<AxesSubplot:>

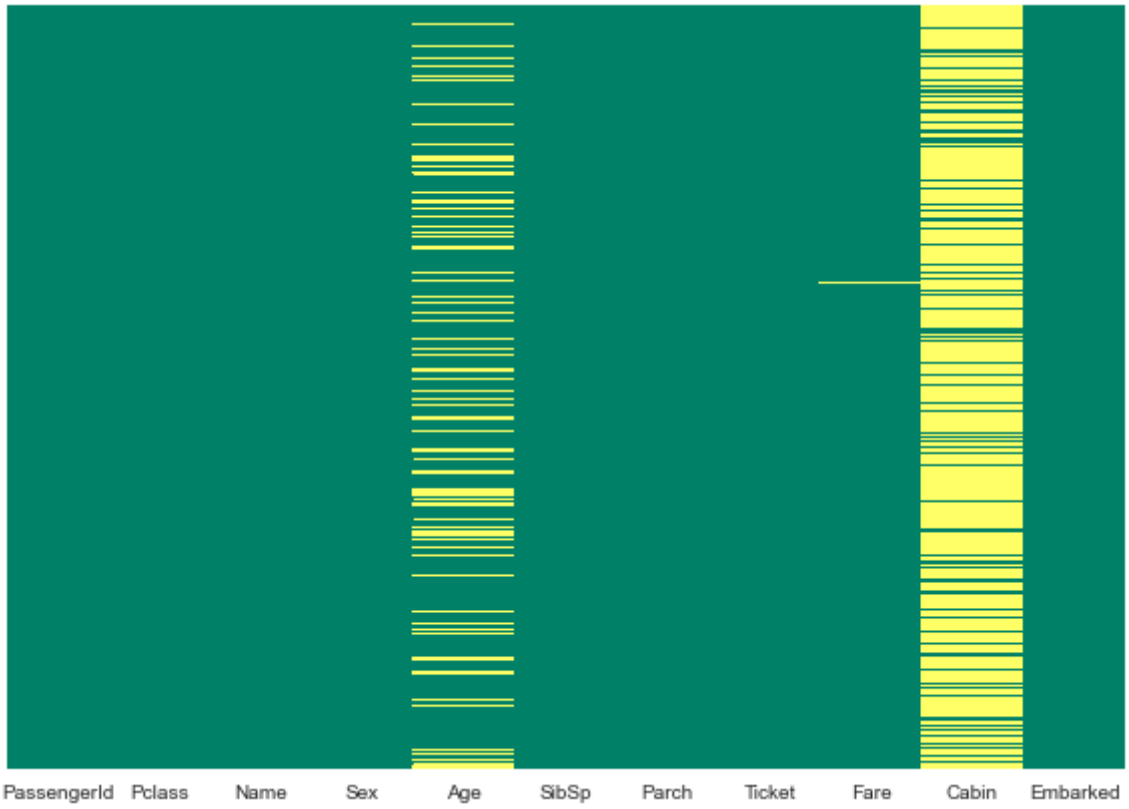


In [48]:

```
plt.figure(figsize=(10,7))
sns.heatmap(test.isnull(), yticklabels=False, cbar=False, cmap="summer")  # cbar : colorbar를 그리
```

Out[48]:

<AxesSubplot:>



[더 알아보기] 데이터의 수치형 변수, 범주형 변수 살펴보기

In [49]:

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

Out[49]:

12

1-2 수치형 변수 살펴보기

In [50]:

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

Out[50]:

	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

1-3 범주형 변수 살펴보기

In [51]:

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

Out[51]:

	Name	Sex	Ticket	Cabin	Embarked
count	891	891	891	204	889
unique	891	2	681	147	3
top	McCarthy, Mr. Timothy J	male	CA. 2343	B96 B98	S
freq	1	577	7	4	644

02 데이터 이해해가기

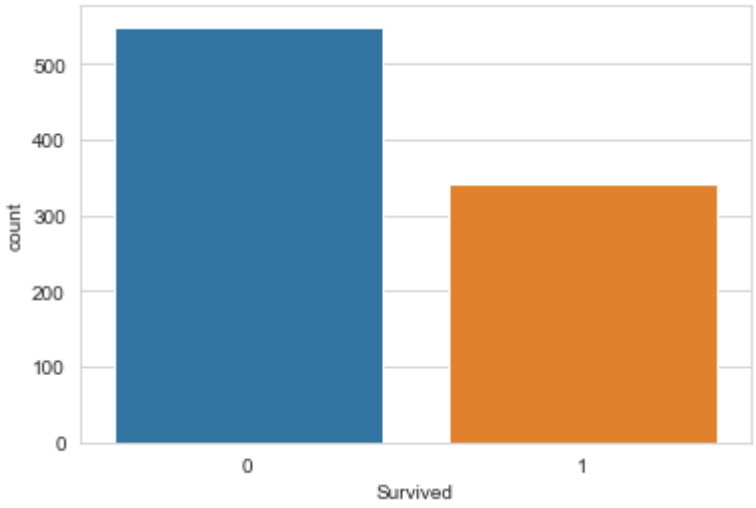
2-1 생존자 사망자의 비율이 얼마나 될까?

In [52]:

```
sns.set_style('whitegrid')   # seaborn 스타일 지정
sns.countplot(x='Survived', data=train)
```

Out[52]:

<AxesSubplot:xlabel='Survived', ylabel='count'>



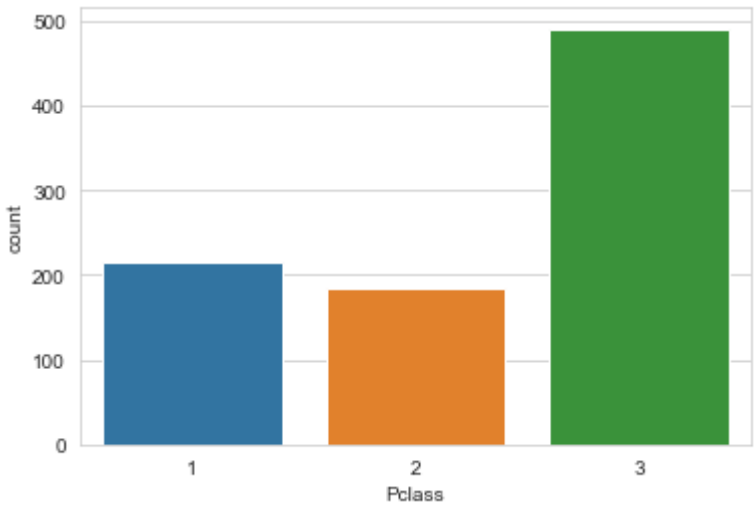
2-2 PClass별 생존자는 얼마나 될까?

In [54]:

```
## 해보기 : PClass 별 Count
sns.countplot(x='Pclass', data=train)
```

Out[54]:

<AxesSubplot:xlabel='Pclass', ylabel='count'>



2-3 나이에 대해 살펴보자

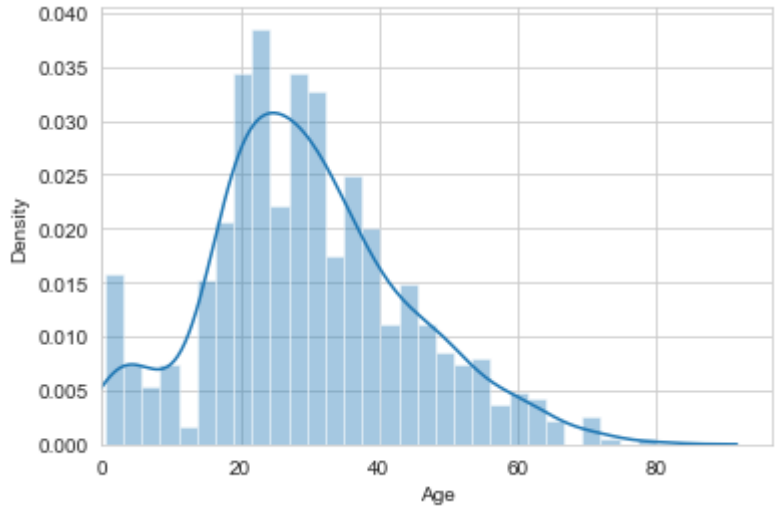
In [56]:

```
sns.distplot(train['Age'].dropna(), bins=30).set_xlim(0,)
```

```
C:₩Users₩toto₩anaconda3₩lib₩site-packages₩seaborn₩distributions.py:2551: 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)
```

Out[56]:

```
(0.0, 96.85957367917433)
```



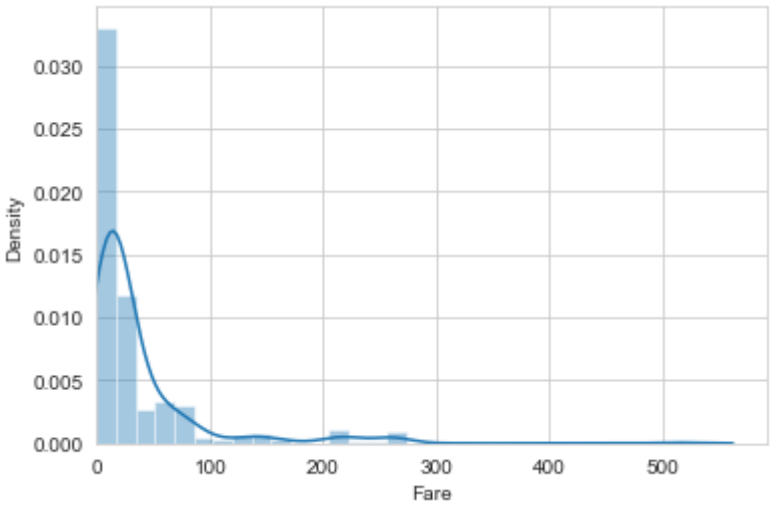
In [57]:

```
## 해보기 Fare
sns.distplot(test['Fare'].dropna(), bins=30).set_xlim(0,)
```

```
C:₩Users₩toto₩anaconda3₩lib₩site-packages₩seaborn₩distributions.py:2551: 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)
```

Out[57]:

```
(0.0, 593.1479879851557)
```



- plt.subplots(행, 열, figsize=(크기지정))

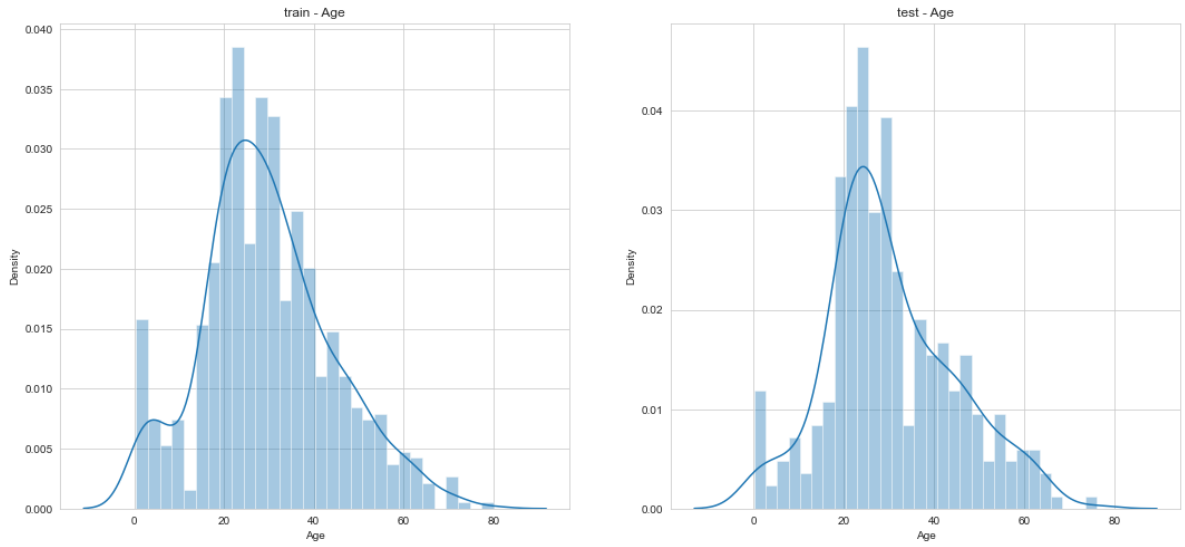
In [58]:

```
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:2551: 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:2551: 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)
```



03 데이터 전처리

3-1 결측치 처리 첫번째

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

In [59]:

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

In [60]:

```
## 해보기
test['Fare'] = test['Fare'].fillna(test['Fare'].mean())
```

In [61]:

```
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
```

3-2 결측치 처리 두번째 Embarked(승선항)

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

In [62]:

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

Out[62]:

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

In [63]:

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

Out[63]:

'S'

In [64]:

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

In [65]:

```
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 [66]:

```
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 [67]:

```
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
```

- 데이터 자료형 변환
- 데이터.astype(변환될 자료형명)

In [68]:

```
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 [69]:

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

In [70]:

```
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
```

In [71]:

```
# 'Name', 'Ticket' =>  문자포함
sel = ['PassengerId', '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]
```

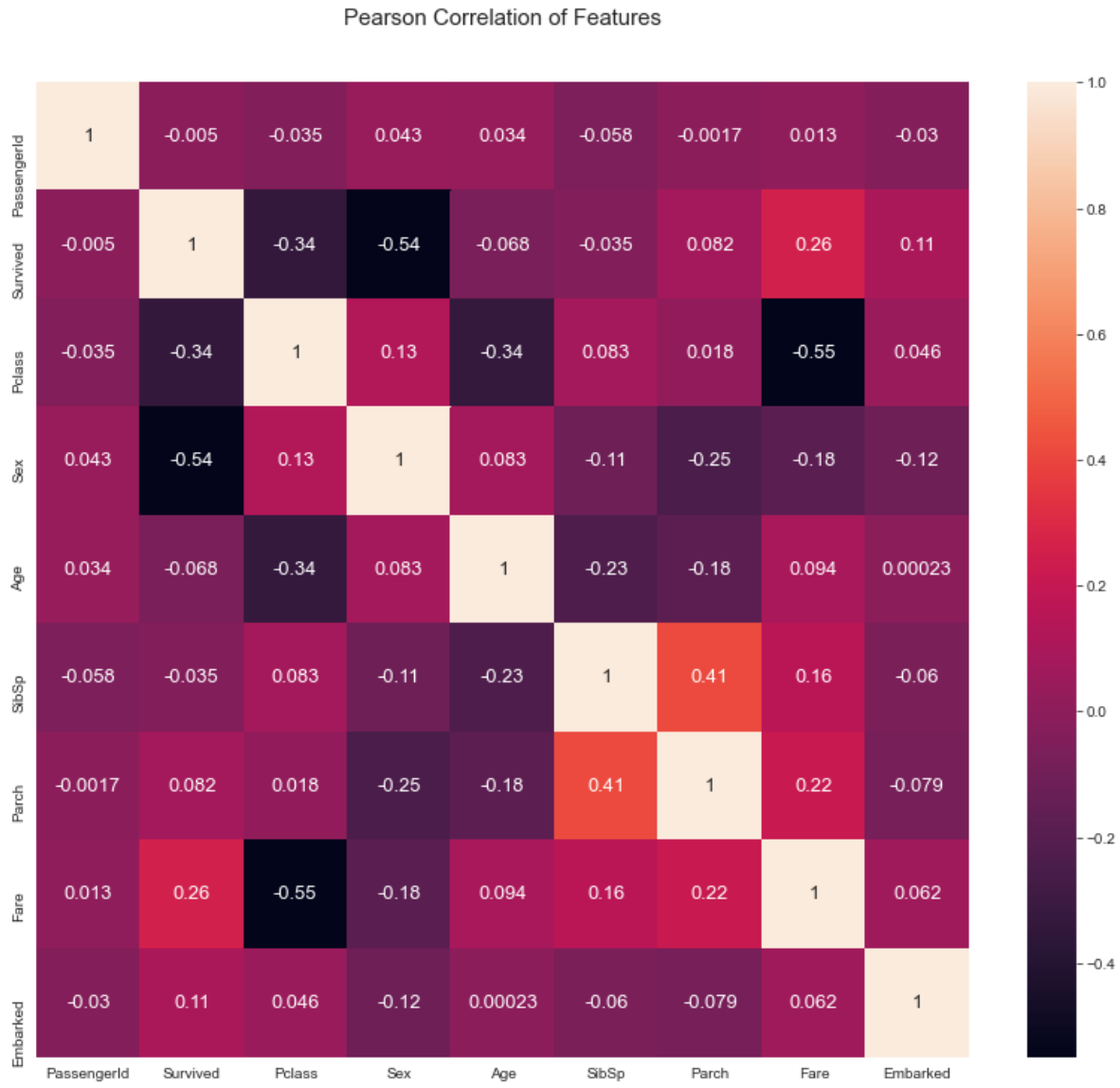
04 컬럼과 컬럼 사이의 관계 확인(상관계수 Heatmap)

In [72]:

```
colormap = plt.cm.RdBu
plt.figure(figsize=(14, 12))
plt.title('Pearson Correlation of Features', y=1.05, size=15)
sns.heatmap(train.corr(), annot=True, annot_kws={"size": 13})
```

Out[72]:

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



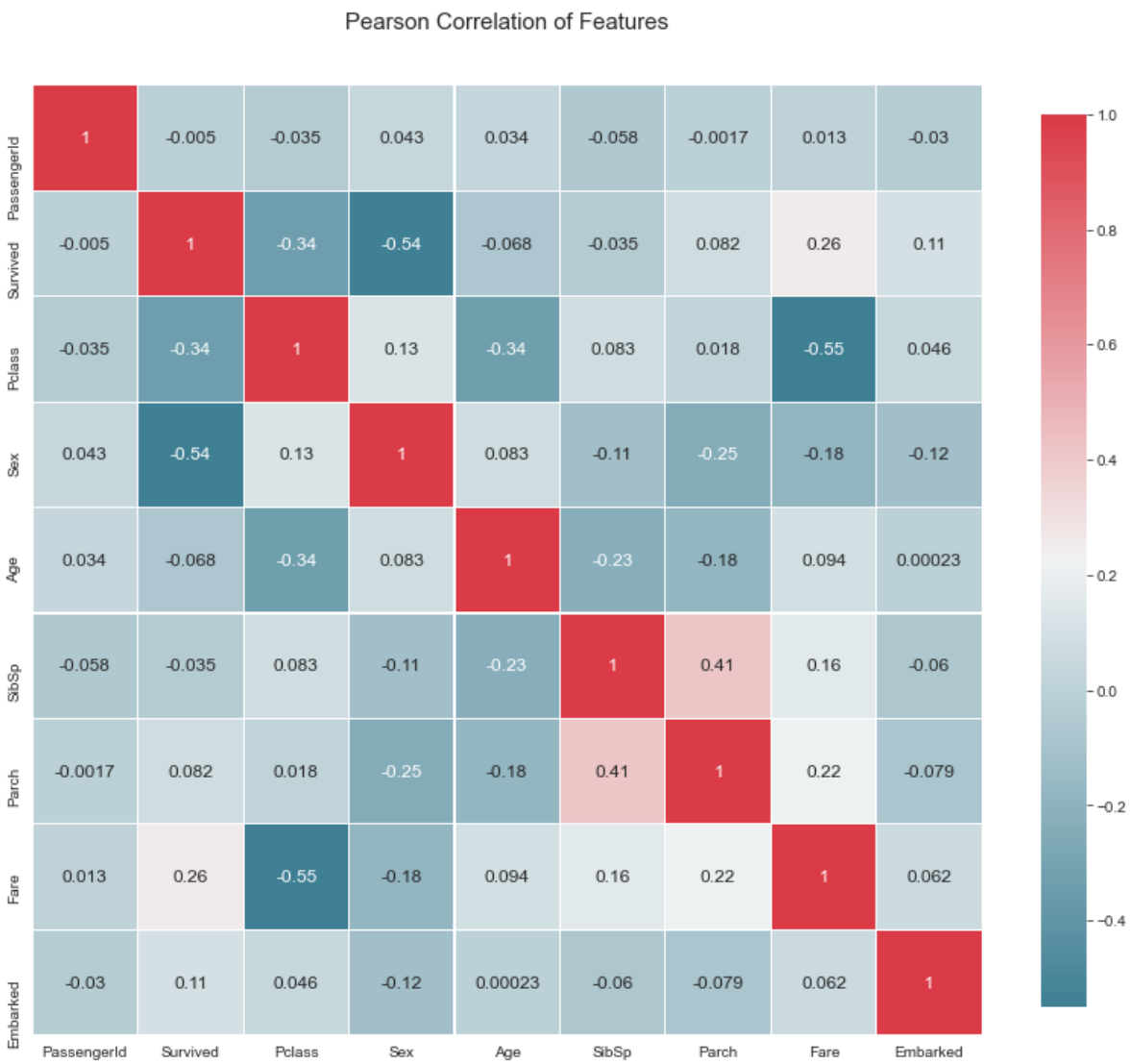
In [40]:

```
#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)
```



05 의사결정 트리 모델 만들고 제출해 보기

- 모델을 생성 후, 학습
- 그리고 예측을 수행 후, 제출한다.

5-1 첫모델 만들기

In [77]:

```
print(X_train.columns)
print(X_test.columns)
```

```
Index(['PassengerId', 'Pclass', 'Sex', 'Age', 'SibSp', 'SibSp', 'Parch',
       'Embarked'],
      dtype='object')
Index(['PassengerId', 'Pclass', 'Sex', 'Age', 'SibSp', 'SibSp', 'Parch',
       'Embarked'],
      dtype='object')
```

In [78]:

```
from sklearn.tree import DecisionTreeClassifier
decisiontree = DecisionTreeClassifier()
decisiontree.fit(X_train, y_train)
```

Out[78]:

```
DecisionTreeClassifier()
```

In [79]:

```
# 예측
predictions = decisiontree.predict(X_test)
predictions[:15]
```

Out[79]:

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

In [80]:

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

5-2 의사결정 트리 모델 - 'Fare'변수 추가

- 모델을 생성 후, 학습
- 그리고 예측을 수행 후, 제출한다.

In [81]:

```
# 'Name', 'Ticket' =>  문자포함
sel = ['PassengerId', '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 [82]:

```
from sklearn.tree import DecisionTreeClassifier
decisiontree = DecisionTreeClassifier()
decisiontree.fit(X_train, y_train)
# 예측
predictions = decisiontree.predict(X_test)
predictions[:15]
```

Out[82]:

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

In [83]:

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

In [ ]:

```
```