Importing Required Libraries

In [ ]:
```
from mpl_toolkits.mplot3d import Axes3D
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt # plotting
import numpy as np # linear algebra
import os # accessing directory structure
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
```

In [45]:
```
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

Defining Functions For Data Plotting

In [29]:
```
# Distribution graphs (histogram/bar graph) of column data
def plotPerColumnDistribution(df, nGraphShown, nGraphPerRow):
    nunique = df.nunique()
    df = df[[col for col in df if nunique[col] > 1 and nunique[col] < 50]] # For displayi
    nRow, nCol = df.shape
    columnNames = list(df)
    nGraphRow = (nCol + nGraphPerRow - 1) / nGraphPerRow
    plt.figure(num = None, figsize = (6 * nGraphPerRow, 8 * nGraphRow), dpi = 80, facecolo
    for i in range(min(nCol, nGraphShown)):
        plt.subplot(nGraphRow, nGraphPerRow, i + 1)
        columnDf = df.iloc[:, i]
        if (not np.issubdtype(type(columnDf.iloc[0]), np.number)):
            valueCounts = columnDf.value_counts()
            valueCounts.plot.bar()
        else:
            columnDf.hist()
        plt.ylabel('counts')
        plt.xticks(rotation = 90)
        plt.title(f'{columnNames[i]} (column {i})')
    plt.tight_layout(pad = 1.0, w_pad = 1.0, h_pad = 1.0)
    plt.show()
```

In [30]:
```
# Correlation matrix
def plotCorrelationMatrix(df, graphWidth):
    filename = df.dataframeName
    df = df.dropna('columns') # drop columns with NaN
    df = df[[col for col in df if df[col].nunique() > 1]] # keep columns where there are m
    if df.shape[1] < 2:
        print(f'No correlation plots shown: The number of non-NaN or constant columns ({df
        return
    corr = df.corr()
    plt.figure(num=None, figsize=(graphWidth, graphWidth), dpi=80, facecolor='w', edgecolo
    corrMat = plt.matshow(corr, fignum = 1)
    plt.xticks(range(len(corr.columns)), corr.columns, rotation=90)
    plt.yticks(range(len(corr.columns)), corr.columns)
    plt.gca().xaxis.tick_bottom()
    plt.colorbar(corrMat)
    plt.title(f'Correlation Matrix for {filename}', fontsize=15)
    plt.show()
```

In [31]:
```
def plotScatterMatrix(df, plotSize, textSize):
    df = df.select_dtypes(include =[np.number]) # keep only numerical columns
    rows and columns that would lead to df being singular
```

```
    df = df.dropna('columns')
    df = df[[col for col in df if df[col].nunique() > 1]] # keep columns where there are m
    columnNames = list(df)
    if len(columnNames) > 10: # reduce the number of columns for matrix inversion of kerne
        columnNames = columnNames[:10]
    df = df[columnNames]
    ax = pd.plotting.scatter_matrix(df, alpha=0.75, figsize=[plotSize, plotSize], diagonal
    corrs = df.corr().values
    for i, j in zip(*plt.np.triu_indices_from(ax, k = 1)):
        ax[i, j].annotate('Corr. coef = %.3f' % corrs[i, j], (0.8, 0.2), xycoords='axes fr
    plt.suptitle('Scatter and Density Plot')
    plt.show()
```

Reading the First Dataset File (Test.csv)

In [34]:

```
nRowsRead = 1000 # specify 'None' if want to read whole file
# test.csv may have more rows in reality, but we are only loading/previewing the first 100
df1 = pd.read_csv('C:/Users/Tanisha/Downloads/archive (1)/test.csv', delimiter=',', nrows
df1.dataframeName = 'test.csv'
nRow, nCol = df1.shape
print(f'There are {nRow} rows and {nCol} columns')
```

```
There are 100 rows and 17 columns
```

In [35]:

```
#glimpse of data of first file
df1.head(5)
```

Out[35]:

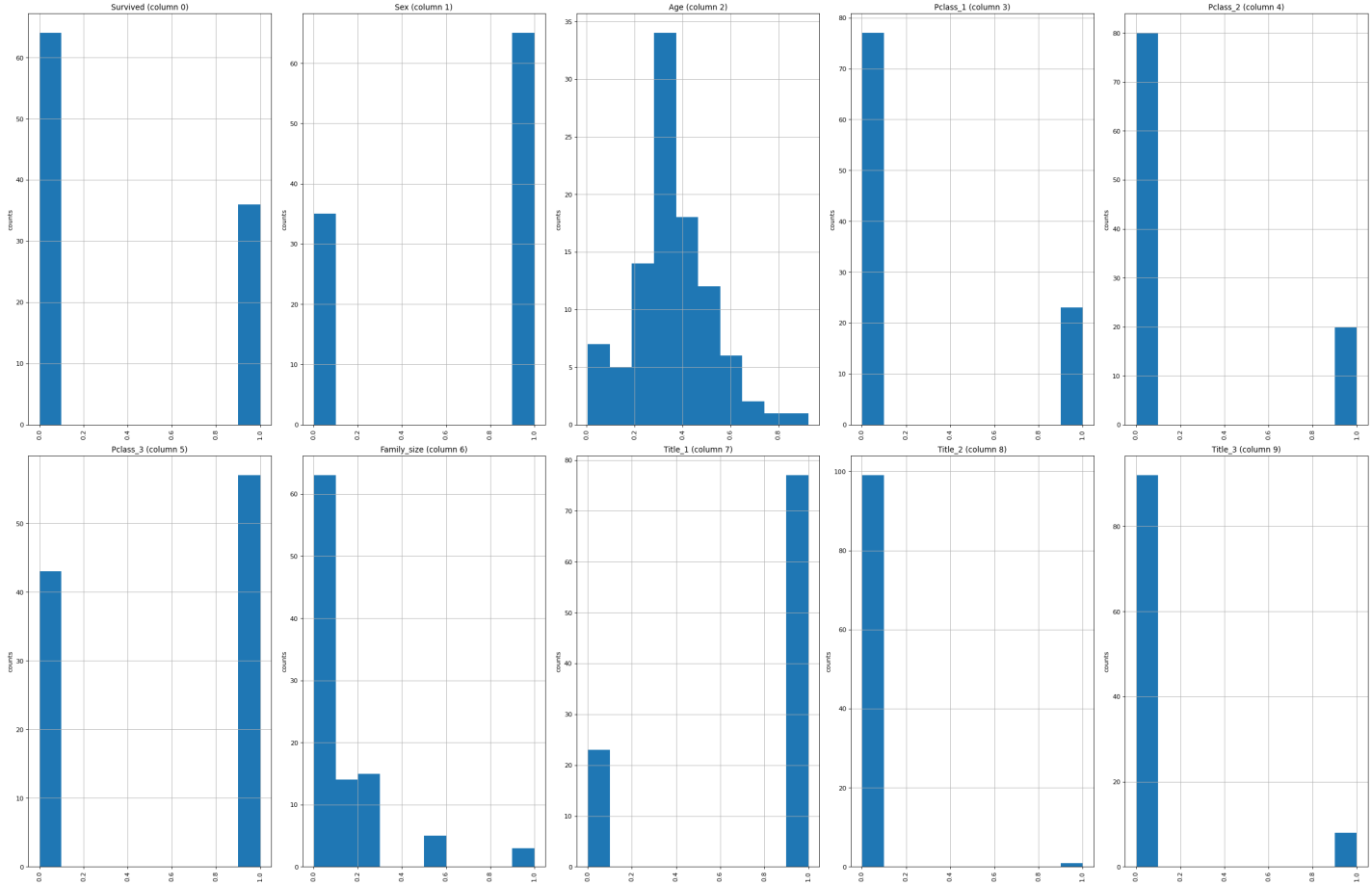
| | Unnamed: 0 | PassengerId | Survived | Sex | Age | Fare | Pclass_1 | Pclass_2 | Pclass_3 | Family_size | Title_1 | T |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 791 | 792 | 0 | 1 | 0.2000 | 0.050749 | 0 | 1 | 0 | 0.0 | 1 | |
| **1** | 792 | 793 | 0 | 0 | 0.3500 | 0.135753 | 0 | 0 | 1 | 1.0 | 0 | |
| **2** | 793 | 794 | 0 | 1 | 0.3500 | 0.059914 | 1 | 0 | 0 | 0.0 | 1 | |
| **3** | 794 | 795 | 0 | 1 | 0.3125 | 0.015412 | 0 | 0 | 1 | 0.0 | 1 | |
| **4** | 795 | 796 | 0 | 1 | 0.4875 | 0.025374 | 0 | 1 | 0 | 0.0 | 1 | |

Distribution Graphs of Sampled Columns

In [36]:

```
plotPerColumnDistribution(df1, 10, 5)
```

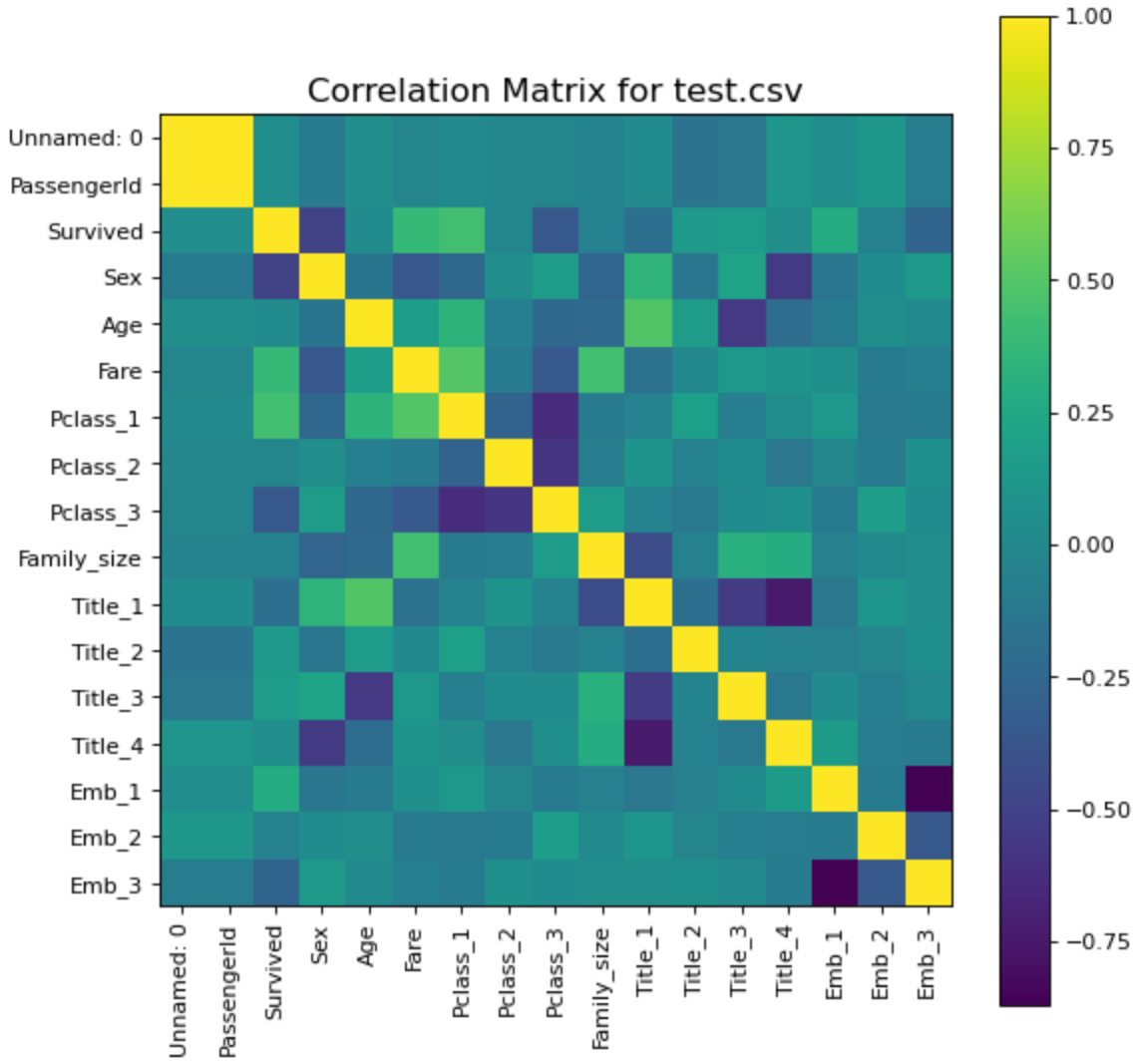
```
<ipython-input-29-0121bf3d9d74>:10: MatplotlibDeprecationWarning: Passing non-integers as
three-element position specification is deprecated since 3.3 and will be removed two minor
releases later.
  plt.subplot(nGraphRow, nGraphPerRow, i + 1)
```




## Correlation Matrix

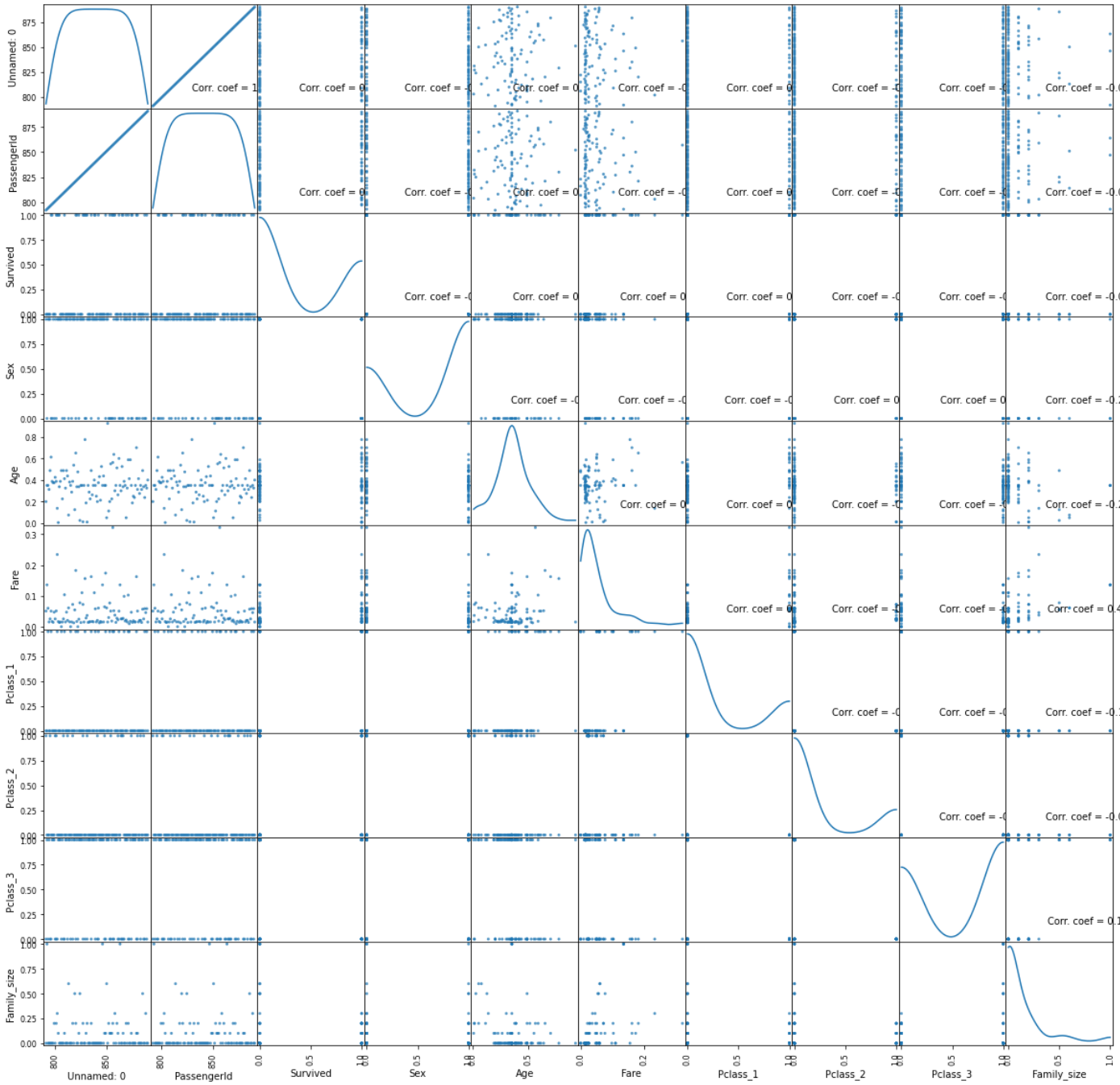
In [37]:

```
plotCorrelationMatrix(df1, 8)
```




Scatter and Density plots of the given data

In [38]:
```
plotScatterMatrix(df1, 20, 10)
```




Reading the second file

In [40]:

```
nRowsRead = 1000 # specify 'None' if want to read whole file
# train_data.csv may have more rows in reality, but we are only loading/previewing the fir
df2 = pd.read_csv('C:/Users/Tanisha/Downloads/archive (1)/train.csv', delimiter=',', nrows
df2.dataframeName = 'train_data.csv'
nRow, nCol = df2.shape
print(f'There are {nRow} rows and {nCol} columns')
```

```
There are 792 rows and 17 columns
```

In [41]:

```
#glimpse of data of second file
df2.head(5)
```

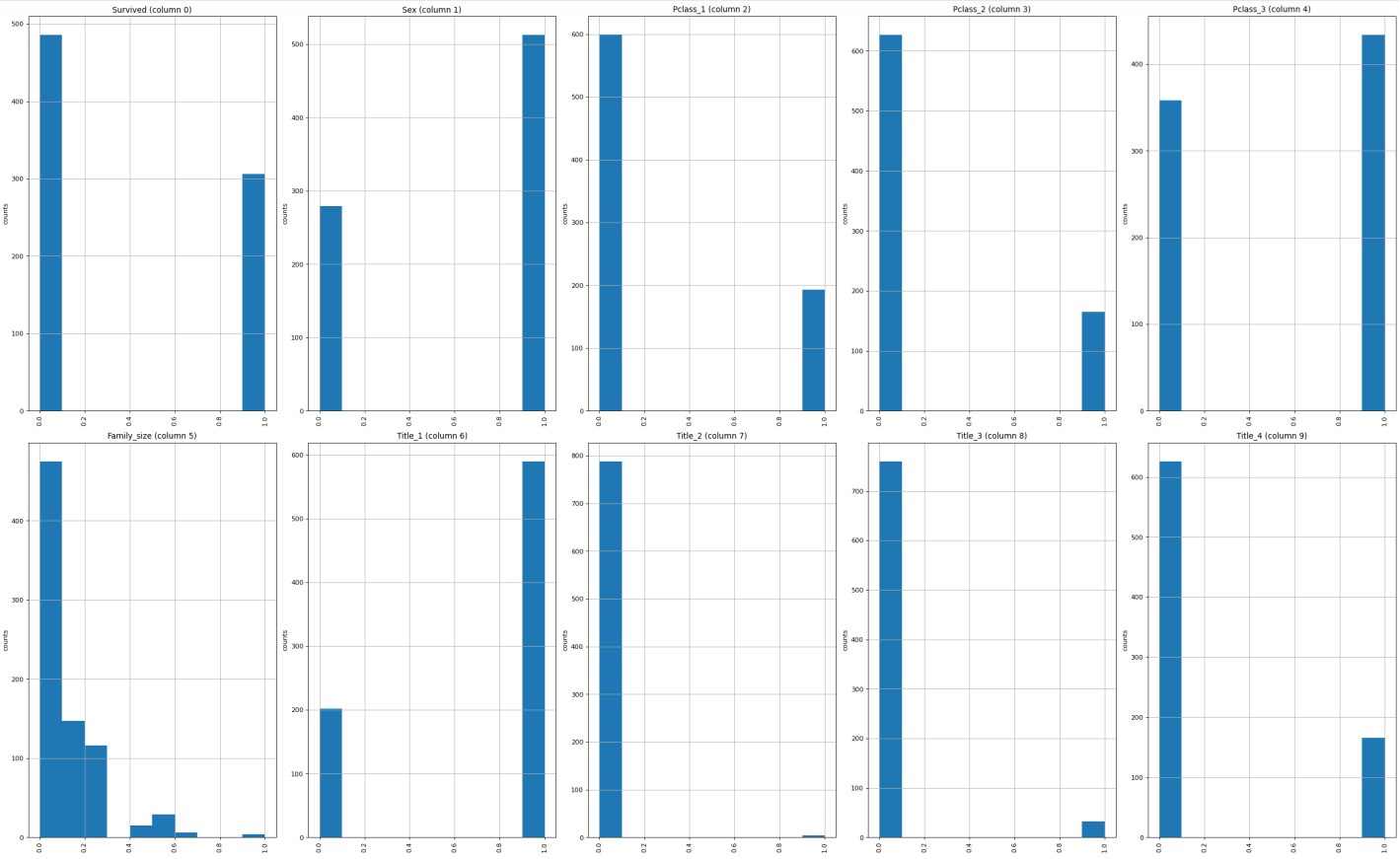
Out[41]:

| | Unnamed: 0 | PassengerId | Survived | Sex | Age | Fare | Pclass_1 | Pclass_2 | Pclass_3 | Family_size | Title_1 | T |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 1 | 0 | 1 | 0.2750 | 0.014151 | 0 | 0 | 1 | 0.1 | 1 | |
| **1** | 1 | 2 | 1 | 0 | 0.4750 | 0.139136 | 1 | 0 | 0 | 0.1 | 1 | |
| **2** | 2 | 3 | 1 | 0 | 0.3250 | 0.015469 | 0 | 0 | 1 | 0.0 | 0 | |
| **3** | 3 | 4 | 1 | 0 | 0.4375 | 0.103644 | 1 | 0 | 0 | 0.1 | 1 | |
| **4** | 4 | 5 | 0 | 1 | 0.4375 | 0.015713 | 0 | 0 | 1 | 0.0 | 1 | |

## Distribution Graphs

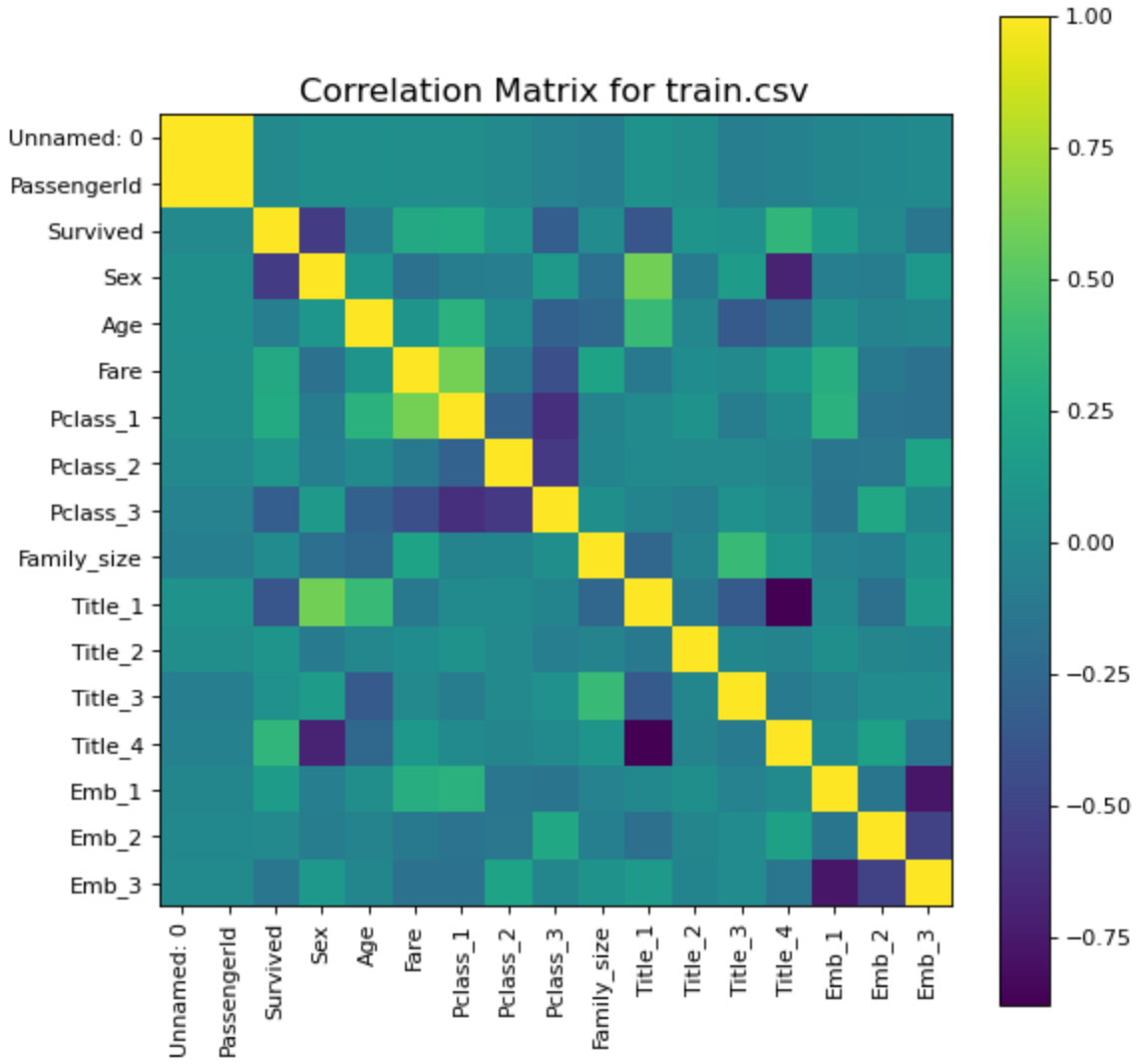
In [42]:
```
plotPerColumnDistribution(df2, 10, 5)
```

```
<ipython-input-29-0121bf3d9d74>:10: MatplotlibDeprecationWarning: Passing non-integers as three-element position specification is deprecated since 3.3 and will be removed two minor releases later.
  plt.subplot(nGraphRow, nGraphPerRow, i + 1)
```



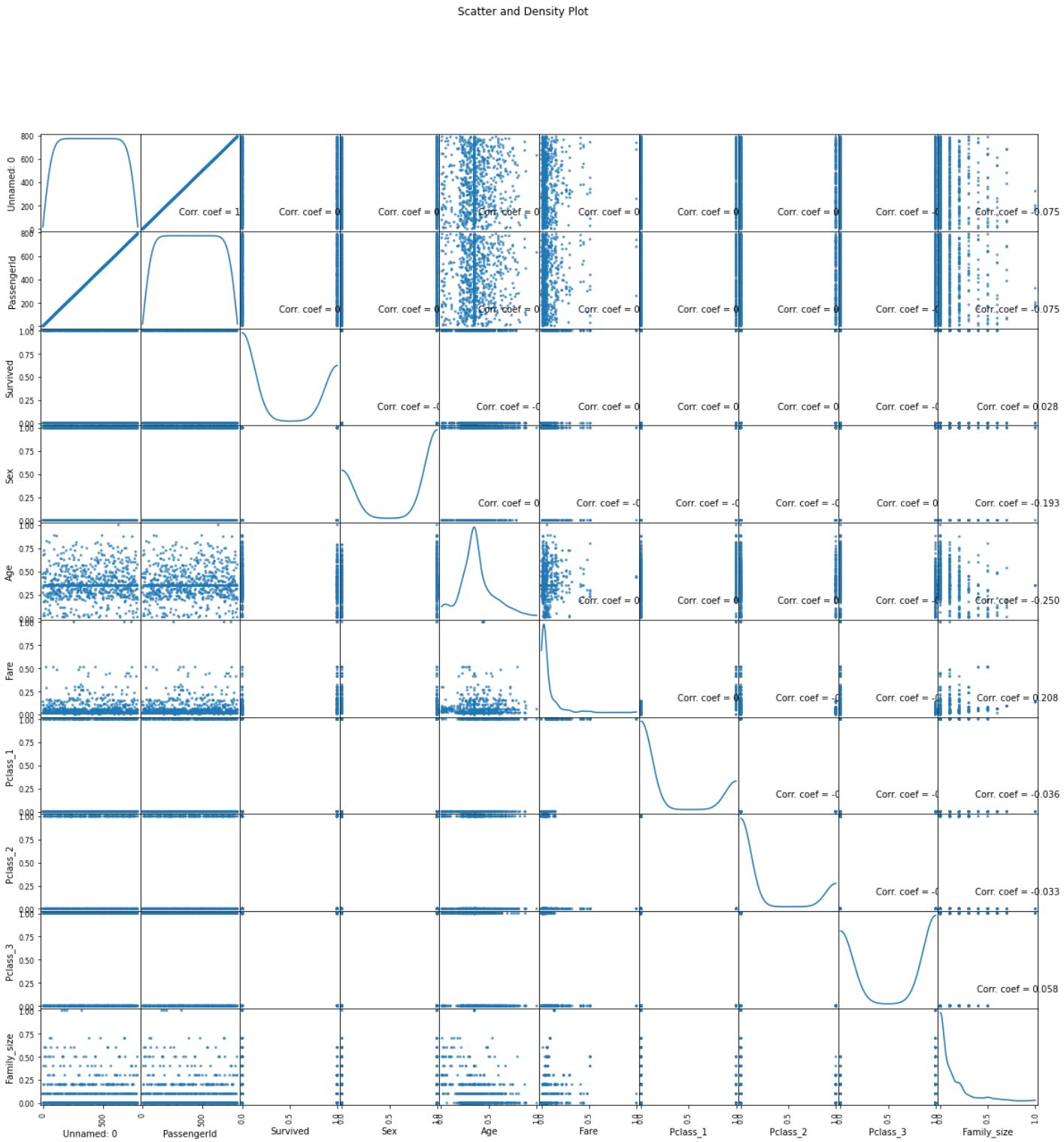

## Correlation Matrix

In [43]:
```
plotCorrelationMatrix(df2, 8)
```




Scatter and Density plots

In [44]:
```
plotScatterMatrix(df2, 20, 10)
```

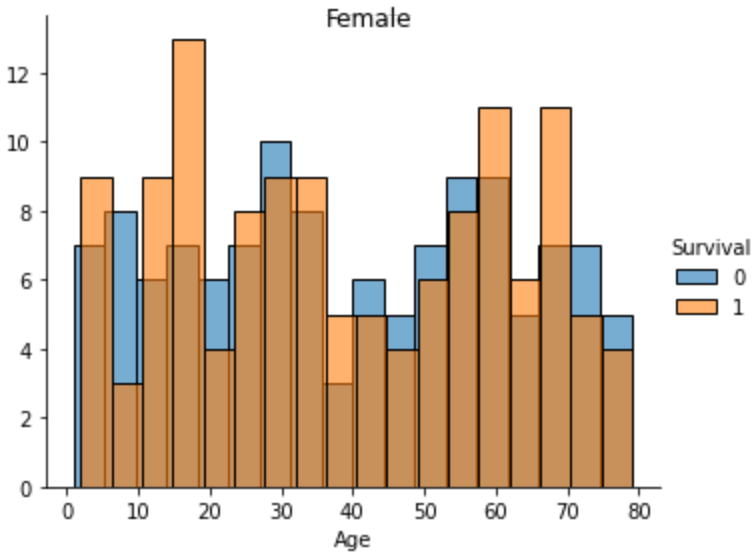


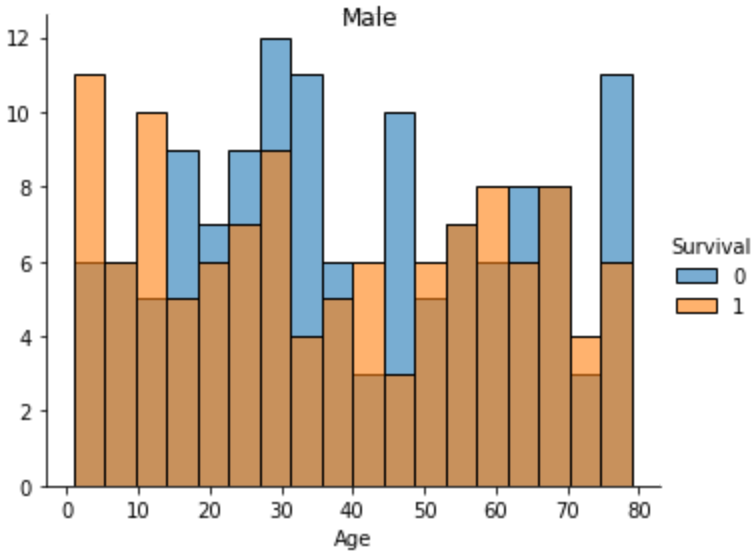

Age and Sex plotting

In [51]:

```
survived = 'survived'
not_survived = 'not survived'


women = train_df[train_df['Sex'] == 'female']
men = train_df[train_df['Sex'] == 'male']

# Female subplot
g_female = sns.FacetGrid(women, hue="Survived", height=4, aspect=1.2)
g_female.map(sns.histplot, "Age", bins=18, multiple="stack", alpha=0.6)
g_female.add_legend(title="Survival")
g_female.set_titles(col_template="{col_name}")
g_female.fig.suptitle("Female")
```

```
# Male subplot
g_male = sns.FacetGrid(men, hue="Survived", height=4, aspect=1.2)
g_male.map(sns.histplot, "Age", bins=18, multiple="stack", alpha=0.6)
g_male.add_legend(title="Survival")
g_male.set_titles(col_template="{col_name}")
g_male.fig.suptitle("Male")

plt.show()
```




In [ ]: