Experiment - 2

```
import pandas as pd
df=pd.read_csv('https://raw.githubusercontent.com/datasciencedojo/datasets/
master/titanic.csv')
print("Shape of dataset:", df.shape)
print("\nInfo:")
print(df.info())
print("\nDescribe:")
print(df.describe(include='all'))
```

```
Shape of dataset: (891, 12)

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
None

Describe:
        PassengerId    Survived      Pclass                Name   Sex  \
count    891.000000  891.000000  891.000000                 891   891
unique          NaN         NaN         NaN                 891     2
top             NaN         NaN         NaN  Dooley, Mr. Patrick  male
freq            NaN         NaN         NaN                   1   577
mean     446.000000    0.383838    2.308642                 NaN   NaN
std      257.353842    0.486592    0.836071                 NaN   NaN
min        1.000000    0.000000    1.000000                 NaN   NaN
25%      223.500000    0.000000    2.000000                 NaN   NaN
50%      446.000000    0.000000    3.000000                 NaN   NaN
75%      668.500000    1.000000    3.000000                 NaN   NaN
max      891.000000    1.000000    3.000000                 NaN   NaN

                Age       SibSp       Parch  Ticket        Fare Cabin Embarked
count   714.000000  891.000000  891.000000     891  891.000000   204      889
unique         NaN         NaN         NaN     681         NaN   147        3
top            NaN         NaN         NaN  347082         NaN    G6        S
freq           NaN         NaN         NaN       7         NaN     4      644
mean     29.699118    0.523008    0.381594     NaN   32.204208   NaN      NaN
std      14.526497    1.102743    0.806057     NaN   49.693429   NaN      NaN
min       0.420000    0.000000    0.000000     NaN    0.000000   NaN      NaN
25%      20.125000    0.000000    0.000000     NaN    7.910400   NaN      NaN
50%      28.000000    0.000000    0.000000     NaN   14.454200   NaN      NaN
75%      38.000000    1.000000    0.000000     NaN   31.000000   NaN      NaN
max      80.000000    8.000000    6.000000     NaN  512.329200   NaN      NaN
```

```
from sklearn.impute import SimpleImputer

age_imputer = SimpleImputer(strategy='mean')
df['Age'] = age_imputer.fit_transform(df[['Age']])

df['Cabin'].fillna('Unknown', inplace=True)

mode_embarked = df['Embarked'].mode()[0]
df['Embarked'].fillna(mode_embarked, inplace=True)
```

```
/tmp/ipython-input-3-2053958636.py:6: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through cha
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are set

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = c


  df['Cabin'].fillna('Unknown', inplace=True)
/tmp/ipython-input-3-2053958636.py:9: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through cha
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are set

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = c
```

```
df['Embarked'].fillna(mode_embarked, inplace=True)
```

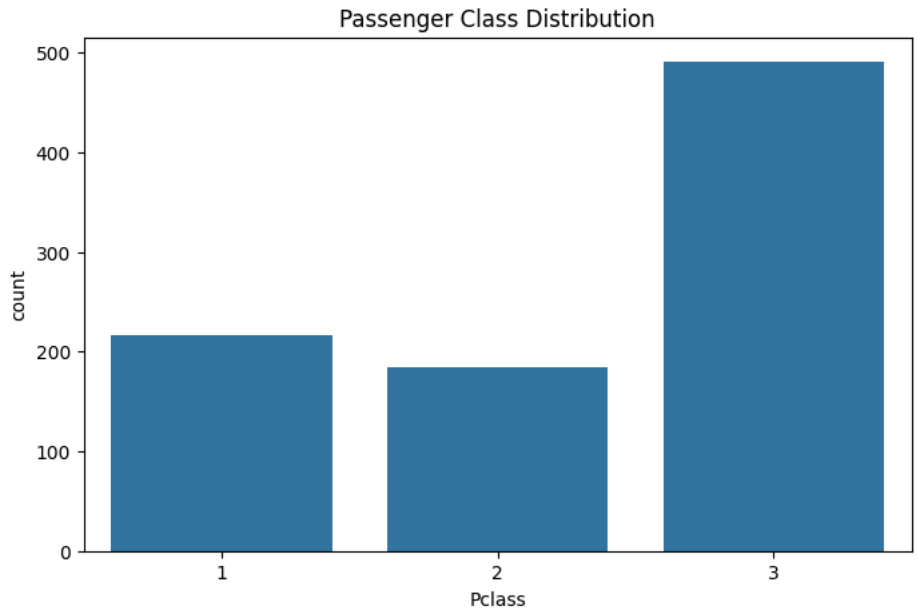
```
female_survived = df[(df['Sex'] == 'female') & (df['Survived'] == 1)]
print(female_survived[['Name', 'Survived']])
```

```
                                                 Name  Survived
1    Cumings, Mrs. John Bradley (Florence Briggs Th...         1
2                               Heikkinen, Miss. Laina         1
3         Futrelle, Mrs. Jacques Heath (Lily May Peel)         1
8    Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)         1
9                  Nasser, Mrs. Nicholas (Adele Achem)         1
..                                                 ...       ...
874              Abelson, Mrs. Samuel (Hannah Wizosky)         1
875                  Najib, Miss. Adele Kiamie "Jane"          1
879      Potter, Mrs. Thomas Jr (Lily Alexenia Wilson)         1
880       Shelley, Mrs. William (Imanita Parrish Hall)         1
887                         Graham, Miss. Margaret Edith         1

[233 rows x 2 columns]
```

```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(8, 5))
sns.countplot(x='Pclass', data=df)
plt.title('Passenger Class Distribution')
plt.show()
```




```
young_3rd_class = df[(df['Pclass'] == 3) & (df['Age'] < 18)]
print(young_3rd_class[['Name', 'Age', 'Pclass']])
```

```
                                     Name   Age  Pclass
7              Palsson, Master. Gosta Leonard   2.0       3
10           Sandstrom, Miss. Marguerite Rut   4.0       3
14      Vestrom, Miss. Hulda Amanda Adolfina  14.0       3
16                      Rice, Master. Eugene   2.0       3
22                McGowan, Miss. Anna "Annie"  15.0       3
..                                        ...   ...     ...
844                     Culumovic, Mr. Jeso  17.0       3
850  Andersson, Master. Sigvard Harald Elias   4.0       3
852                  Boulos, Miss. Nourelain   9.0       3
869           Johnson, Master. Harold Theodor   4.0       3
875          Najib, Miss. Adele Kiamie "Jane"  15.0       3

[78 rows x 3 columns]
```

```
old_1st_class = df[(df['Pclass'] == 1) & (df['Age'] > 40)]
print(old_1st_class[['Name', 'Age', 'Pclass']])
```

```
                                    Name   Age  Pclass
6                 McCarthy, Mr. Timothy J  54.0       1
```

```
11                            Bonnell, Miss. Elizabeth  58.0       1
35                         Holverson, Mr. Alexander Oskar  42.0       1
52              Harper, Mrs. Henry Sleeper (Myna Haxtun)  49.0       1
54                         Ostby, Mr. Engelhart Cornelius  65.0       1
..                                                   ...   ...     ...
856           Wick, Mrs. George Dennick (Mary Hitchcock)  45.0       1
857                              Daly, Mr. Peter Denis   51.0       1
862  Swift, Mrs. Frederick Joel (Margaret Welles Ba...  48.0       1
871   Beckwith, Mrs. Richard Leonard (Sallie Monypeny)  47.0       1
879      Potter, Mrs. Thomas Jr (Lily Alexenia Wilson)  56.0       1

[76 rows x 3 columns]
```

```
survived_old_1st_class = old_1st_class[old_1st_class['Survived'] == 1]
print(survived_old_1st_class[['Name', 'Age', 'Pclass', 'Survived']])
```

```
                                             Name   Age  Pclass  Survived
11                            Bonnell, Miss. Elizabeth  58.0       1         1
52              Harper, Mrs. Henry Sleeper (Myna Haxtun)  49.0       1         1
187      Romaine, Mr. Charles Hallace ("Mr C Rolmane")  45.0       1         1
194          Brown, Mrs. James Joseph (Margaret Tobin)  44.0       1         1
195                                Lurette, Miss. Elise  58.0       1         1
268      Graham, Mrs. William Thompson (Edith Junkins)  58.0       1         1
275                  Andrews, Miss. Kornelia Theodosia  63.0       1         1
299    Baxter, Mrs. James (Helene DeLaudeniere Chaput)  50.0       1         1
337                     Burns, Miss. Elizabeth Margaret  41.0       1         1
366   Warren, Mrs. Frank Manley (Anna Sophia Atkinson)  60.0       1         1
380                              Bidois, Miss. Rosalie  42.0       1         1
449                      Peuchen, Major. Arthur Godfrey  52.0       1         1
453                            Goldenberg, Mr. Samuel L  49.0       1         1
460                                 Anderson, Mr. Harry  48.0       1         1
496                      Eustis, Miss. Elizabeth Mussey  54.0       1         1
513     Rothschild, Mrs. Martin (Elizabeth L. Barrett)  54.0       1         1
523    Hippach, Mrs. Louis Albert (Ida Sophia Fischer)  44.0       1         1
556  Duff Gordon, Lady. (Lucille Christiana Sutherl...  48.0       1         1
571      Appleton, Mrs. Edward Dale (Charlotte Lamson)  53.0       1         1
587                    Frolicher-Stehli, Mr. Maxmillian  60.0       1         1
591    Stephenson, Mrs. Walter Bertram (Martha Eustis)  52.0       1         1
599       Duff Gordon, Sir. Cosmo Edmund ("Mr Morgan")  49.0       1         1
621                         Kimball, Mr. Edwin Nelson Jr  42.0       1         1
630                Barkworth, Mr. Algernon Henry Wilson  80.0       1         1
645                           Harper, Mr. Henry Sleeper  48.0       1         1
647                  Simonius-Blumer, Col. Oberst Alfons  56.0       1         1
660                        Frauenthal, Dr. Henry William  50.0       1         1
707                     Calderhead, Mr. Edward Pennington  42.0       1         1
712                               Taylor, Mr. Elmer Zebley  48.0       1         1
765                Hogeboom, Mrs. John C (Anna Andrews)  51.0       1         1
779  Robert, Mrs. Edward Scott (Elisabeth Walton Mc...  43.0       1         1
796                          Leader, Dr. Alice (Farnham)  49.0       1         1
820  Hays, Mrs. Charles Melville (Clara Jennings Gr...  52.0       1         1
829           Stone, Mrs. George Nelson (Martha Evelyn)  62.0       1         1
856          Wick, Mrs. George Dennick (Mary Hitchcock)  45.0       1         1
857                              Daly, Mr. Peter Denis   51.0       1         1
862  Swift, Mrs. Frederick Joel (Margaret Welles Ba...  48.0       1         1
871   Beckwith, Mrs. Richard Leonard (Sallie Monypeny)  47.0       1         1
879      Potter, Mrs. Thomas Jr (Lily Alexenia Wilson)  56.0       1         1
```

```
male_fare_gt_100 = df[(df['Sex'] == 'male') & (df['Fare'] > 100)]
print(male_fare_gt_100[['Name', 'Fare', 'Sex']])
```

```
                                        Name      Fare   Sex
27                Fortune, Mr. Charles Alexander  263.0000  male
118                     Baxter, Mr. Quigg Edmond  247.5208  male
305               Allison, Master. Hudson Trevor  151.5500  male
332                    Graham, Mr. George Edward  153.4625  male
373                          Ringhini, Mr. Sante  135.6333  male
377                    Widener, Mr. Harry Elkins  211.5000  male
390                  Carter, Mr. William Ernest  120.0000  male
438                             Fortune, Mr. Mark  263.0000  male
505  Penasco y Castellana, Mr. Victor de Satode  108.9000  male
527                           Farthing, Mr. John  221.7792  male
544                   Douglas, Mr. Walter Donald  106.4250  male
550                  Thayer, Mr. John Borland Jr  110.8833  male
557                          Robbins, Mr. Victor  227.5250  male
659                   Newell, Mr. Arthur Webster  113.2750  male
660                Frauenthal, Dr. Henry William  133.6500  male
679           Cardeza, Mr. Thomas Drake Martinez  512.3292  male
698                     Thayer, Mr. John Borland  110.8833  male
737                       Lesurer, Mr. Gustave J  512.3292  male
802          Carter, Master. William Thornton II  120.0000  male
```

```
embarked_C_class2 = df[(df['Embarked'] == 'C') & (df['Pclass'] == 2)]
print(embarked_C_class2[['Name', 'Embarked', 'Pclass']])
```

```
                                                Name Embarked  Pclass
9                  Nasser, Mrs. Nicholas (Adele Achem)        C       2
43            Laroche, Miss. Simonne Marie Anne Andree        C       2
122                               Nasser, Mr. Nicholas        C       2
135                                 Richard, Mr. Emile        C       2
181                                   Pernot, Mr. Rene        C       2
292                              Levy, Mr. Rene Jacques        C       2
308                                 Abelson, Mr. Samuel        C       2
361                            del Carlo, Mr. Sebastiano        C       2
389                                Lehmann, Miss. Bertha        C       2
473       Jerwan, Mrs. Amin S (Marie Marthe Thuillard)        C       2
547                           Padro y Manent, Mr. Julian        C       2
608  Laroche, Mrs. Joseph (Juliette Marie Louise La...        C       2
685             Laroche, Mr. Joseph Philippe Lemercier        C       2
817                                 Mallet, Mr. Albert        C       2
827                               Mallet, Master. Andre        C       2
866                         Duran y More, Miss. Asuncion        C       2
874               Abelson, Mrs. Samuel (Hannah Wizosky)        C       2
```

```
sibsp_gt_2 = df[df['SibSp'] > 2]
print(sibsp_gt_2[['Name', 'SibSp']])
```

```
                                          Name  SibSp
7                        Palsson, Master. Gosta Leonard      3
16                                Rice, Master. Eugene      4
24                         Palsson, Miss. Torborg Danira      3
27                        Fortune, Mr. Charles Alexander      3
50                            Panula, Master. Juha Niilo      4
59                   Goodwin, Master. William Frederick      5
63                                Skoog, Master. Harald      3
68                      Andersson, Miss. Erna Alexandra      4
71                           Goodwin, Miss. Lillian Amy      5
85   Backstrom, Mrs. Karl Alfred (Maria Mathilda Gu...      3
88                            Fortune, Miss. Mabel Helen      3
119                   Andersson, Miss. Ellis Anna Maria      4
159                           Sage, Master. Thomas Henry      8
164                         Panula, Master. Eino Viljami      4
171                                 Rice, Master. Arthur      4
176                       Lefebre, Master. Henry Forbes      3
180                         Sage, Miss. Constance Gladys      8
182               Asplund, Master. Clarence Gustaf Hugo      4
201                                 Sage, Mr. Frederick      8
229                              Lefebre, Miss. Mathilde      3
233                         Asplund, Miss. Lillian Gertrud      4
261                     Asplund, Master. Edvin Rojj Felix      4
266                              Panula, Mr. Ernesti Arvid      4
278                                     Rice, Master. Eric      4
324                               Sage, Mr. George John Jr      8
341                         Fortune, Miss. Alice Elizabeth      3
374                             Palsson, Miss. Stina Viola      3
386                        Goodwin, Master. Sidney Leonard      5
409                                   Lefebre, Miss. Ida      3
480                         Goodwin, Master. Harold Victor      5
485                                 Lefebre, Miss. Jeannie      3
541                  Andersson, Miss. Ingeborg Constanzia      4
542                     Andersson, Miss. Sigrid Elisabeth      4
634                                   Skoog, Miss. Mabel      3
642                          Skoog, Miss. Margit Elizabeth      3
683                            Goodwin, Mr. Charles Edward      5
686                               Panula, Mr. Jaako Arnold      4
726         Renouf, Mrs. Peter Henry (Lillian Jefferys)      3
787                               Rice, Master. George Hugh      4
792                                 Sage, Miss. Stella Anna      8
813                   Andersson, Miss. Ebba Iris Alfrida      4
819                         Skoog, Master. Karl Thorsten      3
824                         Panula, Master. Urho Abraham      4
846                             Sage, Mr. Douglas Bullen      8
850               Andersson, Master. Sigvard Harald Elias      4
863                      Sage, Miss. Dorothy Edith "Dolly"      8
```

```
no_family_not_survived = df[(df['Survived'] == 0) & (df['SibSp'] == 0) & (df['Parch'] == 0)]
print(no_family_not_survived[['Name', 'SibSp', 'Parch', 'Survived']])
```

```
                                    Name  SibSp  Parch  Survived
4                  Allen, Mr. William Henry      0      0         0
5                          Moran, Mr. James      0      0         0
6                   McCarthy, Mr. Timothy J      0      0         0
12          Saundercock, Mr. William Henry      0      0         0
14   Vestrom, Miss. Hulda Amanda Adolfina      0      0         0
..                                    ...    ...    ...       ...
882            Dahlberg, Miss. Gerda Ulrika      0      0         0
883          Banfield, Mr. Frederick James      0      0         0
884                 Sutehall, Mr. Henry Jr      0      0         0
886                  Montvila, Rev. Juozas      0      0         0
```

```
890                  Dooley, Mr. Patrick      0      0         0

[374 rows x 4 columns]
```

```
survived_passengers = df[df['Survived'] == 1]

top5_oldest_survived = survived_passengers.sort_values(by='Age', ascending=False).head(5)

print("Top 5 oldest passengers who survived:")
print(top5_oldest_survived[['Name', 'Age', 'Survived']])
```

```
Top 5 oldest passengers who survived:
                                     Name   Age  Survived
630       Barkworth, Mr. Algernon Henry Wilson  80.0         1
275          Andrews, Miss. Kornelia Theodosia  63.0         1
483                     Turkula, Mrs. (Hedwig)  63.0         1
570                         Harris, Mr. George  62.0         1
829  Stone, Mrs. George Nelson (Martha Evelyn)  62.0         1
```

```
zero_fare_passengers = df[df['Fare'] == 0]

print("Passengers who paid zero fare:")
print(zero_fare_passengers[['Name', 'Fare']])
```

```
Passengers who paid zero fare:
                               Name  Fare
179              Leonard, Mr. Lionel   0.0
263            Harrison, Mr. William   0.0
271      Tornquist, Mr. William Henry   0.0
277       Parkes, Mr. Francis "Frank"   0.0
302   Johnson, Mr. William Cahoone Jr   0.0
413    Cunningham, Mr. Alfred Fleming   0.0
466            Campbell, Mr. William   0.0
481  Frost, Mr. Anthony Wood "Archie"   0.0
597              Johnson, Mr. Alfred   0.0
633     Parr, Mr. William Henry Marsh   0.0
674        Watson, Mr. Ennis Hastings   0.0
732             Knight, Mr. Robert J   0.0
806           Andrews, Mr. Thomas Jr   0.0
815                Fry, Mr. Richard   0.0
822   Reuchlin, Jonkheer. John George   0.0
```

```
from sklearn.model_selection import train_test_split
train_df, test_df = train_test_split(df, test_size=0.2, random_state=42)

print(f"Training set shape: {train_df.shape}")
print(f"Testing set shape: {test_df.shape}")
```

```
Training set shape: (712, 12)
Testing set shape: (179, 12)
```