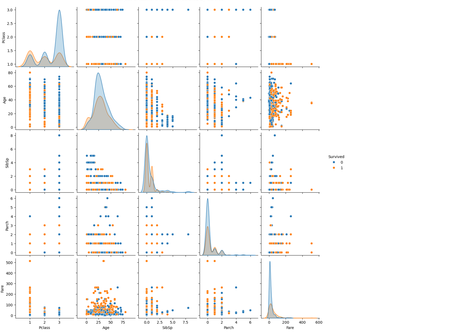
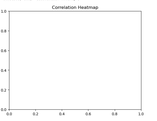
```
In [1]: import pandas as pd
        import numpy as np
        import seaborn as sns
        import matplotlib.pyplot as plt

In [2]: df = pd.read_csv("/train.csv")

In [3]: print(df.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

In [4]: print(df.describe(include="all"))
                PassengerId    Survived      Pclass  \
        count    891.000000  891.000000  891.000000
        unique          NaN         NaN         NaN
        top             NaN         NaN         NaN
        freq            NaN         NaN         NaN
        mean     446.000000    0.383838    2.308642
        std      257.353842    0.486592    0.836071
        min        1.000000    0.000000    1.000000
        25%      223.500000    0.000000    2.000000
        50%      446.000000    0.000000    3.000000
        75%      668.500000    1.000000    3.000000
        max      891.000000    1.000000    3.000000

                                  Name   Sex         Age       SibSp  \
        count                      891   891  714.000000  891.000000
        unique                     891     2         NaN         NaN
        top     Braund, Mr. Owen Harris  male         NaN         NaN
        freq                         1   577         NaN         NaN
        mean                       NaN   NaN   29.699118    0.523008
        std                        NaN   NaN   14.526497    1.102743
        min                        NaN   NaN    0.420000    0.000000
        25%                        NaN   NaN   20.125000    0.000000
        50%                        NaN   NaN   28.000000    0.000000
        75%                        NaN   NaN   38.000000    1.000000
        max                        NaN   NaN   80.000000    8.000000

                     Parch    Ticket        Fare    Cabin  \
        count   891.000000       891  891.000000      204
        unique         NaN       681         NaN      147
        top            NaN  CA. 2343         NaN  B96 B98
        freq           NaN         7         NaN        4
        mean      0.381594       NaN   32.204208      NaN
        std       0.806057       NaN   49.693429      NaN
        min       0.000000       NaN    0.000000      NaN
        25%       0.000000       NaN    7.910400      NaN
        50%       0.000000       NaN   14.454200      NaN
        75%       0.000000       NaN   31.000000      NaN
        max       6.000000       NaN  512.329200      NaN

                Embarked
        count         889
        unique          3
        top             S
        freq          644
        mean          NaN
        std           NaN
        min           NaN
        25%           NaN
        50%           NaN
        75%           NaN
        max           NaN

In [5]: print("Missing Values:\n", df.isnull().sum())
        Missing Values:
         PassengerId      0
        Survived         0
        Pclass           0
        Name             0
        Sex              0
        Age            177
        SibSp            0
        Parch            0
        Ticket           0
        Fare             0
        Cabin          687
        Embarked         2
        dtype: int64

        for col in df.select_dtypes(include='object').columns: print(f"\nValue counts for {col}:\n", df[col].value_counts())

In [6]: sns.pairplot(df[["Survived", "Pclass", "Age", "SibSp", "Parch", "Fare"]], hue="Survived")
        plt.suptitle("Pairplot of Numeric Features Colored by Survival", y=1.02)
        plt.show()
```



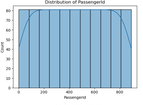
Observation: Higher fares and younger ages show slightly higher survival rates.

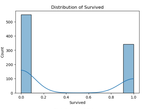
Pclass strongly separates survival chances.

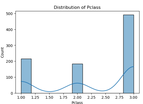
```
In [7]: plt.figure(figsize=(12, 6))
        <Figure size 1200x600 with 0 Axes>
In [8]: <Figure size 1200x600 with 0 Axes>

In [9]: sns.heatmap(df.corr(), annot=True, cmap="coolwarm", fmt=".2f")
        ---------------------------------------------------------------------------
        ValueError                                Traceback (most recent call last)
        Cell In[9], line 1
        ----> 1 sns.heatmap(df.corr(), annot=True, cmap="coolwarm", fmt=".2f")

        File C:\ProgramData\anaconda3\lib\site-packages\pandas\core\frame.py:11049, in DataFrame.corr(self, method, min_periods, numeric_only)
          11047 cols = data.columns
          11048 idx = cols.copy()
        -> 11049 mat = data.to_numpy(dtype=float, na_value=np.nan, copy=False)
          11051 if method == "pearson":
          11052     correl = libalgos.nancorr(mat, minp=min_periods)

        File C:\ProgramData\anaconda3\lib\site-packages\pandas\core\frame.py:1993, in DataFrame.to_numpy(self, dtype, copy, na_value)
           1991 if dtype is not None:
           1992     dtype = np.dtype(dtype)
        -> 1993 result = self._mgr.as_array(dtype=dtype, copy=copy, na_value=na_value)
           1994 if result.dtype is not dtype:
           1995     result = np.asarray(result, dtype=dtype)

        File C:\ProgramData\anaconda3\lib\site-packages\pandas\core\internals\managers.py:1694, in BlockManager.as_array(self, dtype, copy, na_value)
           1692         arr.flags.writeable = False
           1693 else:
        -> 1694     arr = self._interleave(dtype=dtype, na_value=na_value)
           1695     # The underlying data was copied within _interleave, so no need
           1696     # to further copy if copy=True or setting na_value
           1698 if na_value is lib.no_default:

        File C:\ProgramData\anaconda3\lib\site-packages\pandas\core\internals\managers.py:1753, in BlockManager._interleave(self, dtype, na_value)
           1751         else:
           1752             arr = blk.get_values(dtype)
        -> 1753         result[rl.indexer] = arr
           1754         itemmask[rl.indexer] = 1
           1756 if not itemmask.all():

        ValueError: could not convert string to float: 'Braund, Mr. Owen Harris'

In [10]: plt.title("Correlation Heatmap")
         Text(0.5, 1.0, 'Correlation Heatmap')
```

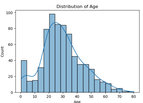



```
In [11]: plt.show()

In [12]: # Observation: Fare and Pclass are negatively correlated (-0.55).
         #              Survived is positively correlated with Fare (0.26) and negatively with Pclass (-0.34).

In [13]: # 4. Histograms

In [14]: for col in df.select_dtypes(include=np.number).columns:

           Cell In[14], line 1
             for col in df.select_dtypes(include=np.number).columns:
         SyntaxError: incomplete input

In [15]: for col in df.select_dtypes(include=np.number).columns:
             plt.figure(figsize=(8, 4))
             sns.histplot(df[col].dropna(), kde=True)
             plt.title(f"Distribution of {col}")
             plt.xlabel(col)
             plt.ylabel("Count")
             # Add observations for each
             if col == "Age":
                 print("Observation: Age distribution is right-skewed with most passengers in 20-40 age range.")
             elif col == "Fare":
                 print("Observation: Fare is heavily right-skewed; a few passengers paid very high fares.")
             elif col == "SibSp":
                 print("Observation: Most passengers traveled without siblings/spouses.")
             elif col == "Parch":
                 print("Observation: Most passengers traveled without parents/children.")
```

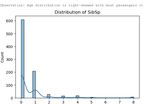




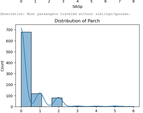




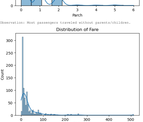
Observation: Age distribution is right-skewed with most passengers in 20-40 age range.



Observation: Most passengers traveled without siblings/spouses.

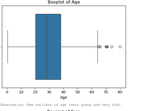


Observation: Most passengers traveled without parents/children.

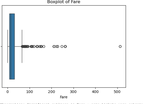


Observation: Fare is heavily right-skewed; a few passengers paid very high fares.

```
In [16]: # 5. Boxplots

In [17]: for col in ["Age", "Fare"]:
             plt.figure(figsize=(8,4))
             sns.boxplot(x=df[col])
             plt.title(f"Boxplot of {col}")
             plt.show()
             if col == "Age":
                 print("Observation: Few outliers in age (very young and very old).")
             elif col == "Fare":
                 print("Observation: Significant outliers in Fare - some tickets were extremely expensive.")
```



Observation: Few outliers in age (very young and very old).



Observation: Significant outliers in Fare - some tickets were extremely expensive.

```
In [18]: plt.figure(figsize=(8,5))
         sns.scatterplot(data=df, x="Fare", hue="Survived", palette="Set1")
         plt.title("Age vs Fare (Colored by Survival)")
         plt.show()
```


```
In [19]: # Observation: Higher fares generally linked to higher survival; survival more common among younger passengers.

In [20]: # 7. Summary of Findings

In [21]: print("""
         Summary:
         1. Dataset has 891 rows, 12 columns, with missing data in Age (177), Cabin (687), Embarked (2).
         2. Most passengers were male (577/891).
         3. Survival rate is higher among females, higher-class passengers, and those paying higher fares.
         4. Pclass is strongly related to Fare and survival chance.
         5. Fare distribution is right-skewed with several outliers.
         6. Pclass is strongly related to Fare and survival chance.
         7. Missing cabin data suggests many passengers traveled in lower-class accommodations.
         """)

         Summary:
         1. Dataset has 891 rows, 12 columns, with missing data in Age (177), Cabin (687), Embarked (2).
         2. Most passengers were male (577/891).
         3. Survival rate is higher among females, higher-class passengers, and those paying higher fares.
         4. Age distribution is concentrated between 20-40 years.
```

5. Pure distribution is right-skewed with extreme outliers.
6. Missing cabin data suggests many passengers traveled in shared or lower-class accommodations.