**Titanic Dataset Analysis using Hierarchical Indexing and Grouping**

**1. Introduction**

This report showcases the analysis of a Titanic dataset using pandas in Python, focusing on hierarchical indexing, cross-section selection, and group aggregation. The dataset, stored in a file named tested.csv, contains information about passengers on the Titanic, such as class (Pclass), sex (Sex), age (Age), fare (Fare), and embarkation port (Embarked).

**2. Data Preprocessing**

* **Loading the Data**: The dataset was loaded into a pandas DataFrame using pd.read\_csv('tested.csv').
* **Column Inspection**: The column names were checked with df.columns, and any unnecessary spaces were removed using df.columns.str.strip() to ensure clean column headers.
* **Displaying Data**: A preview of the first few rows of the dataset was shown using df.head().

**3. Hierarchical Indexing**

* **Setting Hierarchical Index**: The dataset was indexed using three columns: Pclass, Sex, and Embarked. This hierarchical index allows for efficient multi-level grouping and selection of data.

python

Copy code

d = df.set\_index(['Pclass', 'Sex', 'Embarked'])

* **Accessing Specific Groups**: A specific group, such as female passengers in class 1 who embarked at port 'Q', was accessed using the loc method:

python

Copy code

d.loc[(1, 'female', 'Q')]

* **Cross-section Selection**: A cross-section of all female passengers was extracted using the xs method:

python

Copy code

females = d.xs('female', level='Sex')

**4. Group Aggregation**

* **Aggregating Statistics**: The dataset was grouped by the hierarchical index (Pclass, Sex, and Embarked), and various statistics were calculated for Age and Fare:
  + Mean, maximum, and minimum values for Age.
  + Sum of Fare.

python

Copy code

agg\_stats = d.groupby(['Pclass', 'Sex', 'Embarked']).agg({

'Age': ['mean', 'max', 'min'],

'Fare': 'sum'

})

**5. Resetting Hierarchical Index**

* **Resetting Index**: After performing the necessary operations, the hierarchical index was reset back to the default integer index using the reset\_index() method. This is helpful for easier manipulation of the DataFrame after grouping and aggregation.

**6. Conclusion**

This analysis demonstrates how hierarchical indexing can simplify data manipulation and aggregation tasks, especially when dealing with multi-dimensional datasets. By setting multiple columns as indices, it became easy to access and analyze specific groups, such as female passengers from certain classes or embarkation points. Grouping and aggregating statistics based on these indices provided valuable insights into the dataset.