Exploratory Data Analysis (Part II)

Querying Data Frame in Pandas

Projecting one column

- Project ’Name’ and apply head()
- Project ‘Age’ and apply mean, min
- Project ‘Sex’ and apply value_counts()
- Projecting columns and rows with iloc[] and loc[] methods.

Indexing and Row Selection

Print the name of the oldest passenger travelling on ship

```
df[df['Age']== df[df['Sex'] =='male']['Age'].max()]['Name']
```

Sorting

1. sort_values() function
2. Arguments are
    a. By=
    b. ascending=

sort_values(by=[])

```
df.sort_values(by=['Name','Age'],ascending=[True,False]).head(
)
```

Replacing values in column

1. Through map() method
2. Through replace method

```
d={'male':0, 'female':1}
df['Sex'] = df['Sex'].map(d)
df.head()
```

```
o={0: 'male', 1: 'female'}
df= df.replace({'Sex':o})
df.head()
```

Groupby

Describe age of persons grouped by survived attribute.

```
df.groupby(by='Survived')['Age'].describe()
```

Contingency Table: Relating two variables

```
pd.crosstab(df['Sex'],df['Survived'], normalize=True)
```

```
df.pivot_table(['Age', 'Fare'],['Survived'], aggfunc='mean')
```

Sr. no	Task	Method
1	Projecting a column i.e. a feature	df.[‘fname’]
2	Selecting rows and columns	Df.loc[0:5, ‘fanme’:fname] df.iloc[0:5,0:5]
3	Sorting	df.sort_values(by=[],ascending=[])
4	Replacing values in a column	Map and replace method
5	Grouping values	df.groupby()[].describe()
6	Relating two or more than two variables	df.crosstab() df.pivottable()