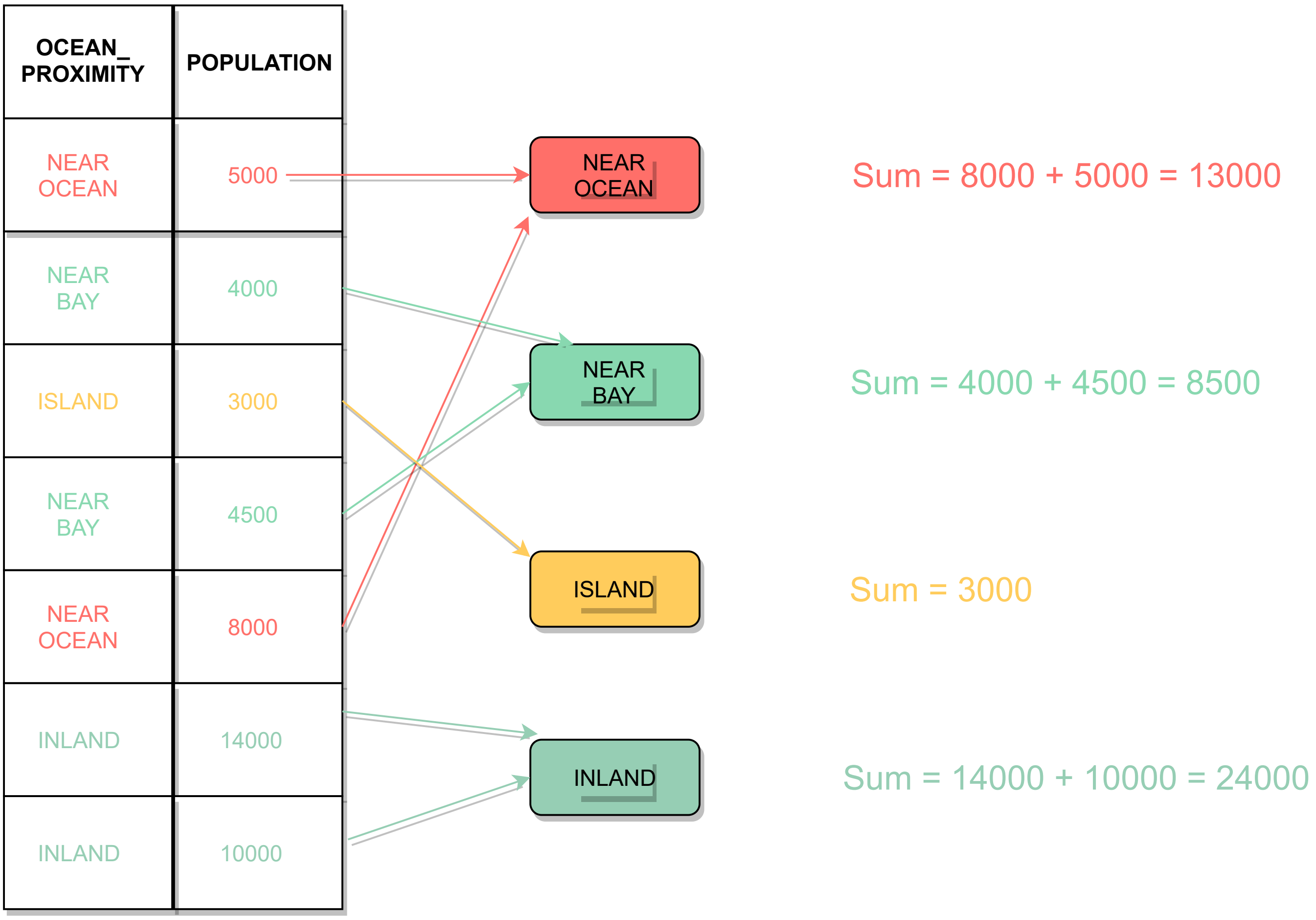
Grouping Data

This lesson introduces us to grouping data and focuses on how grouping can be done with Pandas in Python.

Grouping

During analysis, we may want to gather information about specific types of items in a column. For instance, we may want to separate the data for household blocks based on their proximity to the ocean in our California Housing Dataset and calculate the total population of household blocks in each type of area.

We simply want to *group* data for each type of area and then aggregate population. This can be easily done with Pandas using the function `groupby`. Below is an illustration of how grouping is done.



OCEAN_ PROXIMITY	POPULATION
NEAR OCEAN	5000
NEAR BAY	4000
ISLAND	3000
NEAR BAY	4500
NEAR OCEAN	8000
INLAND	14000
INLAND	10000

Illustration of how grouping and aggregating works

As we can see in the above illustration, for every row that has the value **NEAR OCEAN** in the `ocean_proximity` column, the `population` values go to the red block and these values are aggregated to calculate a sum.

The same process is followed for all other values of `ocean_proximity` column. Groups are created and we say that the data was *grouped by* `ocean_proximity`.

Run the code below to see how easy it is to do this with Pandas. All of the data is in the file *housing.csv*.

housing.csv

```
import pandas as pd
df = pd.read_csv('housing.csv')

# filter data since we only need population adn ocean proximity
cols_select = ['population','ocean_proximity']
filtered_df = df[cols_select]
# groupby and aggregate
sums = filtered_df.groupby('ocean_proximity').sum()
print(sums)
```

We first filter the data in **line 6** since we only need two columns. Then we use `groupby()` to group data by `ocean_proximity` and take the sum in **line 8**.

The output of **line 9** gives us another insight that `<1H OCEAN` areas have the highest population in California.

Usually, we use aggregation functions with grouped data, since we are interested in summarizing data for different groups.

Grouping by more than one variable

We will be using the Student Alcohol Consumption Dataset. This dataset was made to understand how alcohol consumption and other factors influence the grades of school students. We have grades for math class in the file *student-mat.csv*. Let's look at the dataset.

student-mat.csv

```
import pandas as pd
df = pd.read_csv('student-mat.csv')
print(df.head())
```

We might be interested in finding the average grade for all males and females. The final grades are given in the column `G3`. We can do that using a *two-level* groupby. We will group the data by `school` and `gender` and then find the average of the final grade (`G3`).

```
import pandas as pd
df = pd.read_csv('student-mat.csv')

# Filter dataset
df = df[['school','gender','G3']]

# Groupby and aggregation
avgs = df.groupby(['school','gender']).mean()
print(avgs)
```

After reading the dataset, we filter the dataset in **line 5** since we only need three columns. In the next line, we use the `groupby` function and pass it a list of attributes for which we want to group the data. Then we find the average using the `mean` function. We can see from the output of **line 8**, that 4 groups have been made on the basis of school and gender which are:

- Females from school `GP`
- Males from school `GP`
- Females from school `MS`
- Males from school `MS`

Now that we know how to group and aggregate data, we will look at *pivot tables* in the next lesson.