Intro to Data Science - Grouping Data

March 25, 2020

[1]:
```
# This Python 3 environment comes with many helpful analytics libraries␣
 ↪installed
# It is defined by the kaggle/python docker image: https://github.com/kaggle/
 ↪docker-python
# For example, here's several helpful packages to load in

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the "../input/" directory.
# For example, running this (by clicking run or pressing Shift+Enter) will list␣
 ↪all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# Any results you write to the current directory are saved as output.
```

```
/kaggle/input/student-alcohol-consumption/student-por.csv
/kaggle/input/student-alcohol-consumption/student-mat.csv
/kaggle/input/student-alcohol-consumption/student-merge.R
/kaggle/input/california-housing-prices/housing.csv
```

This is the post about **Introduction to Data Scinece**, I am going to write about* handling tabular/dataframe* data in python3.

0.0.1 Importing Data

[2]:
```
import pandas as pd
df = pd.read_csv('/kaggle/input/california-housing-prices/housing.csv')
display(df.head())
```

```
   longitude  latitude  housing_median_age  total_rooms  total_bedrooms  \
0    -122.23     37.88                41.0        880.0           129.0
1    -122.22     37.86                21.0       7099.0          1106.0
2    -122.24     37.85                52.0       1467.0           190.0
```

```
3    -122.25     37.85                52.0       1274.0           235.0
4    -122.25     37.85                52.0       1627.0           280.0

   population  households  median_income  median_house_value ocean_proximity
0       322.0       126.0         8.3252            452600.0        NEAR BAY
1      2401.0      1138.0         8.3014            358500.0        NEAR BAY
2       496.0       177.0         7.2574            352100.0        NEAR BAY
3       558.0       219.0         5.6431            341300.0        NEAR BAY
4       565.0       259.0         3.8462            342200.0        NEAR BAY
```

0.1 ### Grouping Data

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.

[3]:
```
cols = ['ocean_proximity', 'population']
filterd_df = df[cols]
total = filterd_df.groupby('ocean_proximity').sum()
print(total)
```

```
                population
ocean_proximity
<1H OCEAN        13889374.0
INLAND            9112744.0
ISLAND               3340.0
NEAR BAY          2817427.0
NEAR OCEAN        3598955.0
```

0.2 ### 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

[4]:
```
import pandas as pd
df = pd.read_csv('/kaggle/input/student-alcohol-consumption/student-mat.csv')
display(df.head())
```

```
  school sex  age address famsize Pstatus  Medu  Fedu     Mjob      Fjob  ...  \
0     GP   F   18       U     GT3       A     4     4  at_home   teacher  ...
1     GP   F   17       U     GT3       T     1     1  at_home     other  ...
2     GP   F   15       U     LE3       T     1     1  at_home     other  ...
3     GP   F   15       U     GT3       T     4     2   health  services  ...
4     GP   F   16       U     GT3       T     3     3    other     other  ...

  famrel freetime  goout  Dalc  Walc health absences  G1  G2  G3
```

```
0      4        3      4     1     1      3        6   5   6   6
1      5        3      3     1     1      3        4   5   5   6
2      4        3      2     2     3      3       10   7   8  10
3      3        2      2     1     1      5        2  15  14  15
4      4        3      2     1     2      5        4   6  10  10

[5 rows x 33 columns]
```

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)

[5]:
```
cols = ['school', 'sex', 'G3']
new_df = df[cols]
print(new_df.head())

grp = new_df.groupby(['school', 'sex']).mean()
print(grp)
```

```
  school sex  G3
0     GP   F   6
1     GP   F   6
2     GP   F  10
3     GP   F  15
4     GP   F  10
                    G3
school sex
GP     F     9.972678
       M    11.060241
MS     F     9.920000
       M     9.761905
```