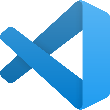
**CS506 Programming for Computing**

**HOP06E– Data Transformation**

06/06/2020 Created by Apiwat Chuaphan

11/08/2020 Revised by Kim Nguyen

Center for Information Assurance (CIAE) @City University of Seattle (CityU)

 A close up of a screen

Description automatically generatedA close up of a sign

Description automatically generated

**Before You Start**

* The directory path shown in screenshots may be different from yours.
* Some steps might not be explained in the tutorial.  If you are not sure what to do:
  + Consult the resources listed below.
  + If you cannot solve the problem after a few tries, ask a TA for help.

**Learning Outcomes**

* Understand data transformation
* Understand data discretization and bin
* Understand the difference between cut and qcut

**Resources**

* Pandas Documentation: <https://pandas.pydata.org/docs/user_guide/index.html>

**Data Transformation**

The process of changing the format, structure, or values of data so that the result may be more efficient and easier to understand. This involves converting data from one structure to another so you can integrate it with different applications.

With Pandas, we can perform transformation easily, we actually did some of transformations in previous HOPs, we will instead use **transform** function in this module.

Open Jupyter Notebook:

1. Create a new file named **data\_transformation.ipynb** under Module folder
2. Create a DataFrame with the following rows and columns.

A screenshot of a cell phone

Description automatically generated

1. Now we will use **transform()** function to add 10 to each element of the dataframe.

DataFrame.transform call **func** on self-producing a DataFrame with transformed values. Produced DataFrame will have same axis length as self.

A screenshot of a computer screen

Description automatically generated

Notice that we used anonymous function (lambda) to add 10 to each value. So, all none-empty values have been added successfully.

**Data Discretization**

Defined as a process of converting continuous data attribute values into a finite set of intervals with minimal loss of information. For example, you have height data and want to discretize it to 0 and 1 interval depending on if the height is below or above a certain value of height.

Open Jupyter Notebook:

1. Within the same file, create a numpy array with 10 random integers between 10 and 200.

A screenshot of a cell phone screen with text

Description automatically generated

1. Numpy provides some of the functions you can perform discretize with, one is using **digitize()** for doing so.

In this step, we will discretize where 50 is the threshold to divide the data into two categories: one less than 50 and ones above than 50.

A picture containing player, meter

Description automatically generated

The bins argument is a list and so we can specify multiple binning or discretizing conditions.

As you can see, there are only two numbers lower than 50 represented as 0. The other are more than 50 represented as 1.

1. Let say if we want to have 3 categories, how many values should we specify in the bins array? You got it, 2 values.

A close up of a sign

Description automatically generated

We have what we wanted. There are 3 categories here: 0, 1, and 2. 0 category is the values less than 50, 1 category is the value less than 100 and category 3 has the value more than 100.

1. Now let’s use a function from Pandas called **cut()** to discretize our data.

It does the same thing as what Numpy’s **digitize()** does, but the way it work is a bit different. Let’s first create a DataFrame with random number from provius.

A screenshot of a video game

Description automatically generated

We created 10 values in height columns with index 0-9.

1. Categorize the height variable into four categories using Pandas **cut** function.

A screenshot of a cell phone

Description automatically generated

The height values between 0 and 25 are in one category, height between 25 and 50 are in the second category, 50-100 in third category, and 100-200 are in the fourth category.

1. We also can label them as follow.

A screen shot of a computer

Description automatically generated

As my random data does not have values below 25 so there is no category 1 in this example. Your data may be different.

You can change labels to string instead of number.

1. Pandas also have another function called **qcut** which discretize variable into equal-sized buckets. You only need to tell the function the number of quantiles, pandas will figure out how to bin that data.

A screenshot of a cell phone

Description automatically generated

Notice that qcut discretized the data into equal size but the interval is not equal. Binned into 5 groups with 2 members each. Unlike what cut function does which is no guarantee about the distribution of items in each bin, but the interval pretty much the same for all bin.

A screenshot of a cell phone

Description automatically generated

1. Let’s look at a familiar example below of how to use discretize with data.

A screenshot of a cell phone

Description automatically generated

Line number 3, we sorted for easier to see in each category. We categorized what grades belong to which categories. Then, we put the data into DataFrame.

1. If you are visual person, you need to see the chart. You can definitely do that with a basic plotting provided by pandas like below image.

We will do a bar plot based on the number of students in each group.

A screenshot of a cell phone

Description automatically generated

*Note: In case you have the following error: "matplotlib is required for plotting”*

*Then do the following step:*

In the terminal, type the following to install matplotlib library and rerun your code



If you want to see what **value\_counts()** does, remove .plot.bar()

**Push your work to GitHub**

Open the terminal from the VSCode by hitting the “control” + “~” key and type the following command:

>>> git add .

>>> git commit -m “Submission for Module 5 – Your Name”

>>> git push origin master