In the lecture we mentioned the benefits of Data Transformation, but can you think of any problems that might arise with Data Transformation?

Data Transformation can make complex data clearer. For example, when we analyze the information of product reviews, it is difficult to analyze each review. At this time, we can use NLP to process them into emotional dimensions, or eliminate irrelevant information. At this time The data can express their mathematical relationship more clearly.

Do you think data transformation or validation should come first in your pipeline? Why or why not?

Yes, we should load the mess first and see what we have to deal with. For example, we can first load the information into a hashset to see what results are in this column of data, and then decide which method to use.

For example we already know about drop, does insert exist? How can you find more information about a method's arguments or examples of usage?

We can output the distribution map of the information of some columns, such as outputting a line chart, to see whether a certain data is completely centralized, or extremely cluttered, and then compare it with our other data. Check if there is a logical relationship between them, if not, then we can consider deleting it.

Hint: are you dropping rows or columns? Is there an argument for that in the drop method? What types of values does the usecols argument expect?

Usually drop columns. No, I just dropped unneeded extraneous Columns. usecols has been used when loading data into dataframe, and there are no related parameters.

Hint: How can you decode a row into multiple rows in pandas? While it may be tempting to try to iterate through the dataframe and append new rows, instead consider table-level pandas methods that you can use. If any DataFrame methods you want to use are not available on a Series, is there an equivalent method for the Series?

for dataframe method just like this:

```
df2 = (
 df.assign(proportion=df['proportion'].str.split('/'))
   .explode('proportion')
   .reset_index(drop=True)
)
d = df2['proportion'].str.split('-', expand=True)

df2['proportion'] = d[0]
df2['value'] *= d[1].astype(float)
```

Hint: How can you check for bad data like NaN or duplicates in a DataFrame? How can you find all the unique values in a column? For a column named like VALID_FLAG, what do you think are the expected values?

Use pandas.dataframe.isna method to check bad data like NaN or duplicates

fall a column we can use pandas.dataframe.fillna method

for column 'VALID_FLAG' i prefer set all 'NaN' to 'N'

Hint: What is the frequency of the bus datapoints? Do we expect them every minute, every few seconds, etc? Does interpolate achieve this automatically? If not, how can you adjust it to do so?

we can use the `df['ARRIVE_TIME'].value_counts()` method to get frequency.

```
27030.0    180
24079.0    178
21403.0    147
23786.0     34
29139.0     19
```

interpolate cant achieve this automatically, so we need set the index = liner

Could you have used the **interpolate()** method for problem E above?

we can use

```
df = df.interpolate(method='linear', axis=0).ffill().bfill()
```