Notebook test

Customs Data Aggregation

In [ ]:
```
import pandas as pd
```

In [ ]:
```
data = pd.read_csv('rt_57394.csv')
```

In [ ]:
```
# data processing and rename dataframe columns
data['ADG_CODE'] = data['GOODSTNVEDCODE'].astype(str).apply(lambda x:
'0' + x if len(x) == 10 else str(x))
data['ADG_CODE'] = data['ADG_CODE'].str.extract('(\d{4})')

data['TOTAL_WITH_TAXES'] = data['INVOICED_COST']
data['UNIT_PRICE'] = data['INVOICED_COST']/data['GOODS_QUANTITY']
data['UNIT'] = data['MEASUR_UNIT_QUALIFIER_NAME']
data = data.rename({'COD':'TIN', 'GOODS_DSC':'GOOD_NAME'}, axis=1)
data['APPLICATION_DATE_DT'] = pd.to_datetime(data['APPLICATION_DATE'])
# filter for the needed timeframe
data = data[
    (pd.to_datetime('2023-10-
01')>=pd.to_datetime(data['APPLICATION_DATE_DT']))&
    (pd.to_datetime(data['APPLICATION_DATE_DT'])<=pd.to_datetime('2024-
06-01'))
]
```

In [ ]:
```
df_gr = data.groupby(['TIN', 'GOOD_NAME']).agg({
    'TOTAL_WITH_TAXES': ['min', 'max', 'mean', 'sum'],
    'UNIT_PRICE': ['min', 'max', 'mean', 'sum'],
    'UNIT': [lambda x: x.mode()[0] if len(x.mode()) else None, lambda
x:set(x) if len(x) else None],
    'ADG_CODE': [lambda x:set(x) if len(x) else None, lambda x: x.mode()
[0] if len(x.mode()) else None]
})
```

In [ ]:
```
df_gr.columns = ['TOTAL_WITH_TAXES_min',
'TOTAL_WITH_TAXES_max','TOTAL_WITH_TAXES_mean','TOTAL_WITH_TAXES_sum',
                 'UNIT_PRICE_min',
'UNIT_PRICE_max','UNIT_PRICE_mean','UNIT_PRICE_sum',
                  'UNIT_mode', 'UNIT_set',
                  'ADG_CODE_set','ADG_CODE_mode'
                 ]

df_gr = df_gr.reset_index()
```

In [ ]:
```
df_gr.to_csv('D:/data/decl_grouped_data.csv', index=False)
```

Hdm Data Aggregation

In [ ]:
```
import pandas as pd
```

In [ ]:
```
data = pd.read_csv('hdm_data.csv')
```

In [ ]:
```
df_gr = data.groupby(['TIN', 'GOOD_NAME']).agg({
    'TOTAL_WITH_TAXES': ['min', 'max', 'mean', 'sum'],
    'UNIT_PRICE': ['min', 'max', 'mean', 'sum'],
    'UNIT': [lambda x: x.mode()[0] if len(x.mode()) else None,  lambda
x:set(x) if len(x) else None],
    'ADG_CODE': [lambda x:set(x) if len(x) else None, lambda x: x.mode()
[0] if len(x.mode()) else None]
})
```

In [ ]:
```
df_gr.columns = ['TOTAL_WITH_TAXES_min',
'TOTAL_WITH_TAXES_max','TOTAL_WITH_TAXES_mean','TOTAL_WITH_TAXES_sum',
                 'UNIT_PRICE_min',
'UNIT_PRICE_max','UNIT_PRICE_mean','UNIT_PRICE_sum',
                  'UNIT_mode', 'UNIT_set',
                  'ADG_CODE_set','ADG_CODE_mode'
                 ]
df_gr = df_gr.reset_index()
df_gr.to_csv('D:/data/hdm_data_grouped.csv', index=False)
```

In [ ]:
```

```