Transaction Monitoring Task

Customer segmentation and outlier detection based on transactional behavior.

Product type

Number of customers that have used all 3 product types:

```
wire_cheque_cash_customers = reduce(np.intersect1d, [df_wire.customer_id, df_cheque.customer_id, df_cash.customer_id])

print(len(wire_cheque_cash_customers))
9934
```

66 customers haven't used at least one transaction method.

How many of them have used one product type less than 4 times?

```
wire_or_cheque_df = df[df['customer_id'].isin(wire_or_cheque)]
wire_or_cash_df = df[df['customer_id'].isin(wire_or_cash)]
cheque_or_cash_df = df[df['customer_id'].isin(cheque_or_cash)]
```

```
wire_or_cheque_df.groupby('customer_id')['product_type'].value_counts().loc[lambda x : x<4]
wire_or_cash_df.groupby('customer_id')['product_type'].value_counts().loc[lambda x : x<4]
cheque_or_cash_df.groupby('customer_id')['product_type'].value_counts().loc[lambda x : x<4]
```

```
less_than_four_trans = [1223, 1467, 5936, 8244, 8816, 41, 3400, 3692, 4272, 8643]
```

Country and credit/debit

Customers that made one transaction to one country and more than 10 to another country.

```
one_trans_per_country = df.groupby('customer_id')['CPCC'].value_counts().loc[lambda x : x<2]
tenplus_trans_per_country = df.groupby('customer_id')['CPCC'].value_counts().loc[lambda x : x>10]

one_and_tenplus_trans = reduce(np.intersect1d, [one_trans_per_country.customer_id, tenplus_trans_per_country.customer_id])
```

```
array([ 159,  304,  308,  478,  515,  541,  627,  696,  771,  882, 1280, 1285, 1470, 1495, 1502, 1503, 1649, 1711, 1785, 2102, 2399, 3041, 3053, 3100, 3359, 3487,
3532, 3762, 4188, 4316, 4327, 4423, 4463, 4779, 4823, 5042, 5126, 5173, 5568, 5616, 5783, 6013, 6189, 6212, 6540, 6621, 7060, 7080, 7135, 7221, 7377, 7395,
8096, 8157, 8252, 8403, 9286, 9397])
```

Customers who made only 1 debit transaction to one country, but more than 10 credit transactions to a different one (and viceversa)

```
one_cd_per_country = df_credit.groupby('customer_id')['CPCC'].value_counts().loc[lambda x : x<2]
one_db_per_country = df_debit.groupby('customer_id')['CPCC'].value_counts().loc[lambda x : x<2]
tenplus_cd_per_country = df_credit.groupby('customer_id')['CPCC'].value_counts().loc[lambda x : x>10]
tenplus_db_per_country = df_debit.groupby('customer_id')['CPCC'].value_counts().loc[lambda x : x>10]
```

```
one_cd_tenplus_db = reduce(np.intersect1d, [one_cd_per_country.customer_id, tenplus_db_per_country.customer_id])
one_db_tenplus_cd = reduce(np.intersect1d, [one_db_per_country.customer_id, tenplus_cd_per_country.customer_id])
```

```
one_cd_tenplus_db → 9986
one_db_tenplus_cd → 4905
```

Amount

Which customers made transactions higher than a certain threshold?

```
def find_high_amount_customer_ids (df,col1, col2):
    df_high_amount = df[df[col1]> 15000.0]
    return df_high_amount[col2].unique()

high_amount_trans = find_high_amount_customer_ids(df,'amount','customer_id')
```

```
array([ 659, 3041,  661, 4109, 1071,  664,  793, 4078,  529, 1409, 7407,
       4046, 2845,  674,  668, 1828, 1791, 1531, 7134,  699,  570,  509,
       1275, 3452,  964, 4459, 7212, 1411, 1519, 3288,  856, 4070])
```

One customer (3041) made a high amount transaction and also made one transaction with one country plus more than 10 with a different one.

```
reduce(np.intersect1d, [high_amount_trans,one_and_tenplus_trans]) → 3041
```

Customers that made more than 25 monthly transactions

```
df['customer_id'].value_counts().loc[lambda x : x>25] →

array([5860,  695,  895,  899,  581,  866,  923,  682,  933,  635,  563,
        943,  538,  643,  883, 4070,  514,  534,  932,  981])
```

High risk countries

Which are the risk countries?

```
risk_countries = df2[df2['risk']==
'Y'].country_abb.array

['HE', 'IS', 'PF', 'RO', 'TL', 'YT']
```

Create a dataframe that contains only the rows relevant the risk countries

```
df_risk_countries =
df[df['CPCC'].isin(risk_countries)]
```

Now I can define a monthly transaction threshold and see which customers would trigger a warning.

```
monthly_trans_threshold = 10000.0
trans_over_thresh = df_risk_countries[df_risk_countries['amount']>= monthly_trans_threshold]
trans_over_thresh.customer_id.unique()
```

```
array([ 659, 1746, 1520, 1626, 1493, 1058, 1761, 1808, 1289, 1696, 1964,1329, 1064,  588, 1067, 1409, 1365, 1402, 1627, 1577, 1589,  918, 824, 1161, 1430,  609, 1117, 1549,  600,
1789,  570, 1377, 1226, 1173, 1672, 1175, 1921, 1250, 1877,  534, 1071,  691, 1434, 1999,
1339, 1667, 1719, 1521, 1782,  689, 1468, 1225, 1972, 1183, 1248, 1821, 1608, 1284, 1509,  643,  695, 1894,  892, 1918,  647,  803,
1314, 1777, 1454, 1896, 1174,  806, 1662,  572,  580, 1232, 1315,1749, 1823, 1815, 1612,  631, 1412,  665, 1988, 1827,  694,  648, 1881,  542, 1096, 1291,  802,  611,  879, 1552,
1163,  962, 1006,622,  874, 1048, 1295, 1361, 1712,  947,  838, 2845, 1660, 1393,
1864,  668, 1900, 1322,  540, 1581,  850, 1978,  558, 1837,  598,1767, 1701, 1559, 1041,  561, 1033, 1757, 1009, 1149, 1922, 1119,1868, 1915, 1357, 1395, 1555,  590,  866,  500,
1825, 1210, 1479,1828, 1171,  678, 1396, 1791, 1436, 1255, 1081, 1531, 1170, 1389,683, 1993, 1074, 1673,  601, 1012, 1139,  624,  576, 1554,  804,677, 1958,  841, 1543,  872, 1386,
699, 1423,  525,  686,  509, 1275, 1753, 1945,  644, 1806, 1787, 1729, 1407,  585, 1651, 1308,1156, 1817, 1032, 1060,  531, 1676,  612,  682, 1700,  933,  827, 664, 1024, 1205, 1301,
633, 1680,  537, 1421, 1411, 1501, 1681, 1792, 1216, 1450,  981, 1417, 1732, 1370,  675,  511, 1963, 1179, 1265, 1570, 1775, 1312, 1536, 1708, 1425,  987, 1850, 1858, 1960,650, 1107,
696, 1128,  618,  673,  893, 1857,  594, 1387, 1306,1269,  645, 1324, 1360, 1923, 1611,  698,  821,  932, 1630, 1980, 920, 1464,  641, 1774,  906, 1372])
```