In [2]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
df_customer = pd.read_excel(r"C:\Users\User\Downloads\KB DS Case 1 (1).xlsx",sheet_name = 1)
```

In [4]:
```
df_customer
```

Out[4]:

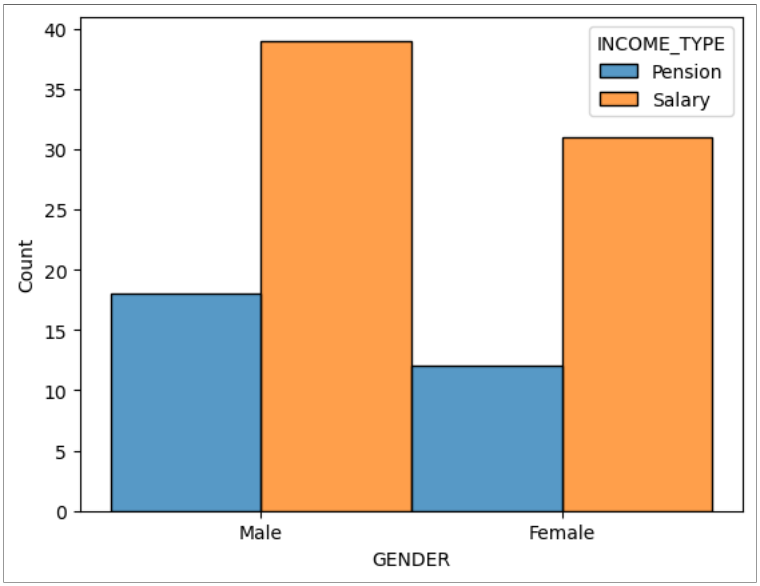
	CIF	BIRTH_DATE	GENDER	REGISTERED_CITY_REGION	INCOME_TYPE	AVERAGE_SALARY	AVERAGE_PENSION
0	1000001	1962-09-09	Male	GEDEBEY	Pension	0	257
1	1000002	1966-12-05	Female	ZARDAB	Salary	1033	0
2	1000003	1988-03-02	Male	ASTARA	Salary	488	0
3	1000004	1987-06-10	Male	BAKU	Salary	1860	0
4	1000005	1977-06-26	Female	BEYLAQAN	Salary	1654	0
...	...	...	...	...	...	...	...
95	1000096	1991-02-21	Female	AGCABEDI	Salary	1641	0
96	1000097	1977-09-28	Female	AGJABADI	Salary	1375	0
97	1000098	1997-01-27	Male	BAKU	Salary	1633	0
98	1000099	1967-09-22	Male	MASSALI	Salary	1946	0
99	1000100	1979-05-23	Female	BAKU	Salary	1508	0

100 rows × 7 columns

In [5]:
```
sns.histplot(data = df_customer,x = 'GENDER',hue = 'INCOME_TYPE',multiple="dodge")
```

Out[5]:

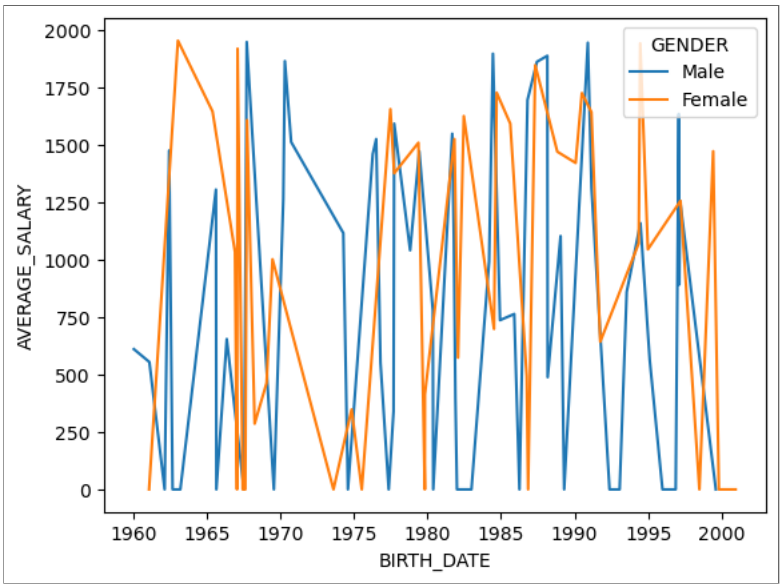
```
<AxesSubplot:xlabel='GENDER', ylabel='Count'>
```



In [6]:
```
sns.lineplot(data = df_customer,x = 'BIRTH_DATE',y = 'AVERAGE_SALARY',hue = 'GENDER')
```

Out[6]:

```
<AxesSubplot:xlabel='BIRTH_DATE', ylabel='AVERAGE_SALARY'>
```



In [7]:
```
df_contracts = pd.read_excel(r"C:\Users\User\Downloads\KB DS Case 1 (1).xlsx",sheet_name = 2)
df_contracts
```

Out[7]:

	CIF	CONTRACT_NUMBER	PRODUCT_CODE	STATUS	AMOUNT/LIMIT	INTEREST	DURATION	OPEN_DATE	CLOSE_DATE	
0	1000094	BBLK0001	BBLK	ACTIVE	7100	25	36	2021-08-06	2024-08-05	3.88
1	1000043	BBLK0002	BBLK	ACTIVE	5000	25	36	2021-01-28	2024-01-28	3.88
2	1000093	BBLK0003	BBLK	ACTIVE	5500	25	36	2020-07-21	2023-07-21	3.88
3	1000096	BBLK0004	BBLK	ACTIVE	2500	25	36	2021-01-01	2024-01-01	3.88
4	1000081	BBLK0005	BBLK	CLOSED	6400	25	36	2017-11-07	2020-11-06	3.88
...	...	...	...	...	...	...	...	...	...	
115	1000097	GTKR0066	GTKR	ACTIVE	9300	27	36	2019-09-15	2022-09-14	
116	1000016	GTKR0067	GTKR	ACTIVE	2800	28	48	2018-12-09	2022-12-08	
117	1000023	GTKR0068	GTKR	CLOSED	3900	21	12	2019-10-10	2020-10-09	
118	1000022	GTKR0069	GTKR	CLOSED	5800	21	12	2018-08-29	2019-08-29	
119	1000100	GTKR0070	GTKR	ACTIVE	6800	28	48	2020-04-06	2024-04-05	

120 rows × 10 columns

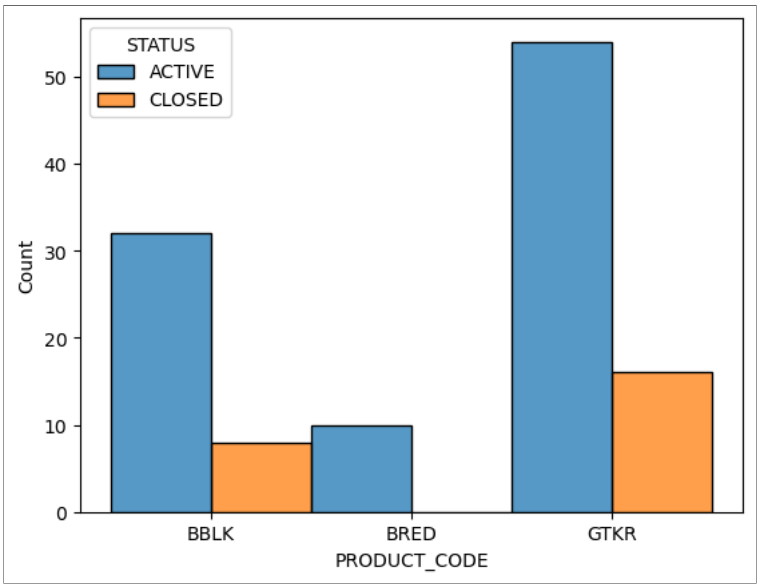
In [8]:
```
df_contracts.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 120 entries, 0 to 119
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype         
---  ------           --------------  -----         
 0   CIF              120 non-null    int64         
 1   CONTRACT_NUMBER  120 non-null    object        
 2   PRODUCT_CODE     120 non-null    object        
 3   STATUS           120 non-null    object        
 4   AMOUNT/LIMIT     120 non-null    int64         
 5   INTEREST         120 non-null    int64         
 6   DURATION         120 non-null    int64         
 7   OPEN_DATE        120 non-null    datetime64[ns]
 8   CLOSE_DATE       120 non-null    datetime64[ns]
 9   ACCOUNT_NUMBER   50 non-null     float64       
dtypes: datetime64[ns](2), float64(1), int64(4), object(3)
memory usage: 9.5+ KB
```

In [9]:
```
sns.histplot(data = df_contracts,x = 'PRODUCT_CODE',hue = 'STATUS',multiple="dodge")
```

Out[9]:

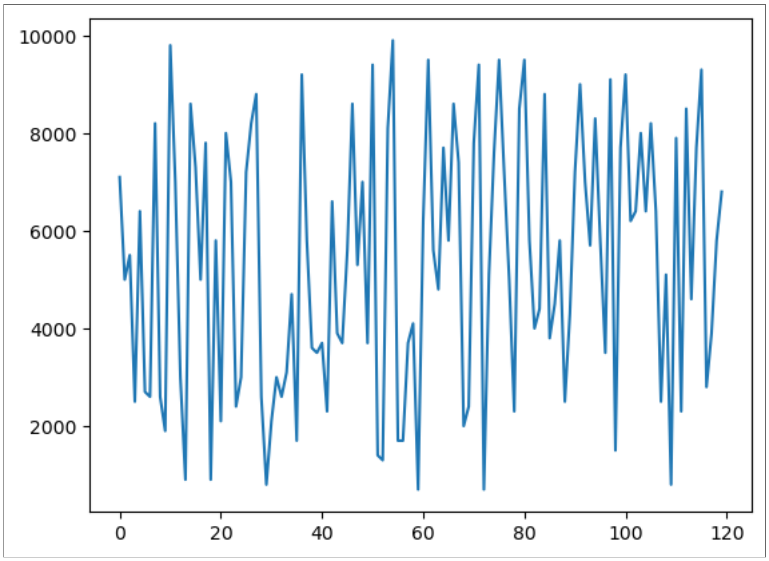
```
<AxesSubplot:xlabel='PRODUCT_CODE', ylabel='Count'>
```



In [10]:
```
plt.plot(df_contracts['AMOUNT/LIMIT'])
```

Out[10]:

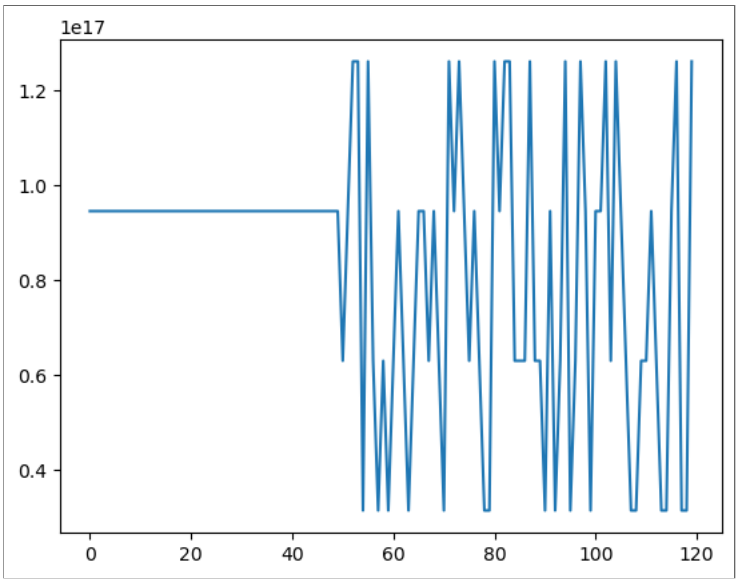
```
[<matplotlib.lines.Line2D at 0x1a141b0ffd0>]
```



In [11]:
```
plt.plot((df_contracts['CLOSE_DATE'] - df_contracts['OPEN_DATE']))
```

Out[11]:

```
[<matplotlib.lines.Line2D at 0x1a141ad54f0>]
```



In [12]:
```
df_contracts['PRODUCT_CODE'].value_counts()
```

Out[12]:

```
GTKR    70
BBLK    40
BRED    10
Name: PRODUCT_CODE, dtype: int64
```

In [13]:
```
df_product = pd.read_excel(r"C:\Users\User\Downloads\KB DS Case 1 (1).xlsx",sheet_name = 3)
df_product
```

Out[13]:

	PRODUCT_CODE	PRODUCT_NAME	PRODUCT_TYPE
0	BRED	Birkart Red	Card
1	BBLK	Birkart Black	Card
2	GTKR	Gündəlik Tələbat	Cash

In [14]:
```
df_transaction = pd.read_excel(r"C:\Users\User\Downloads\KB DS Case 1 (1).xlsx",sheet_name = 4)
df_transaction
```

Out[14]:

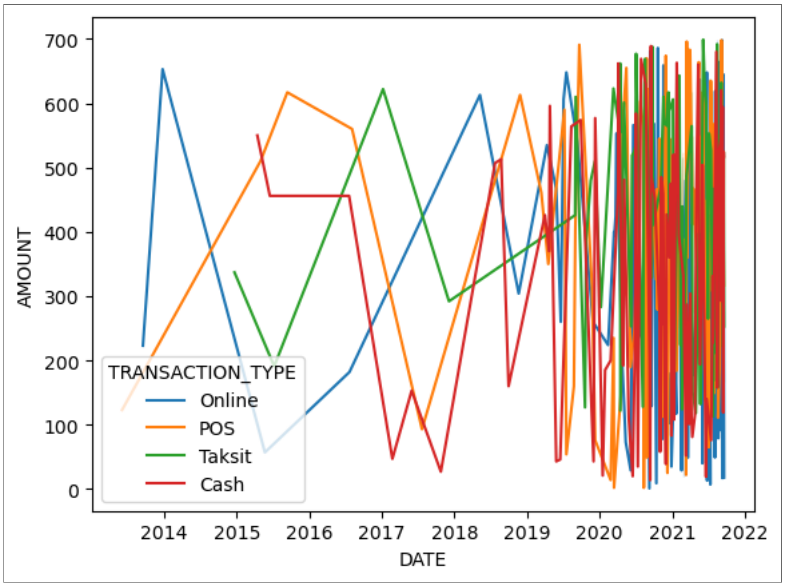
	ACCOUNT_NUMBER	DATE	AMOUNT	TRANSACTION_TYPE
0	3881790981	2021-04-05	261	Online
1	3881790930	2020-03-12	235	POS
2	3881790981	2021-04-14	549	POS
3	3881791455	2020-07-13	473	POS
4	3881791398	2019-09-03	610	Taksit
...	...	...	...	...
399	3881790526	2019-12-09	510	Taksit
400	3881791572	2021-08-20	111	POS
401	3881790550	2021-09-18	517	Cash
402	3881791273	2018-05-09	613	Online
403	3881790981	2021-01-14	213	POS

404 rows × 4 columns

In [15]:
```
sns.lineplot(data = df_transaction,x = 'DATE',y = 'AMOUNT',hue = 'TRANSACTION_TYPE')
```

Out[15]:

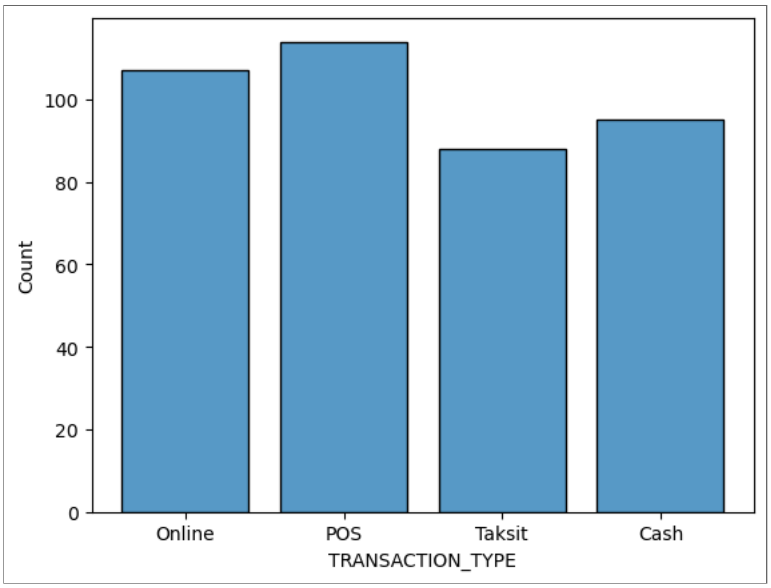
```
<AxesSubplot:xlabel='DATE', ylabel='AMOUNT'>
```



In [16]:
```
sns.histplot(data = df_transaction,x = 'TRANSACTION_TYPE',shrink=.8)
```

Out[16]:

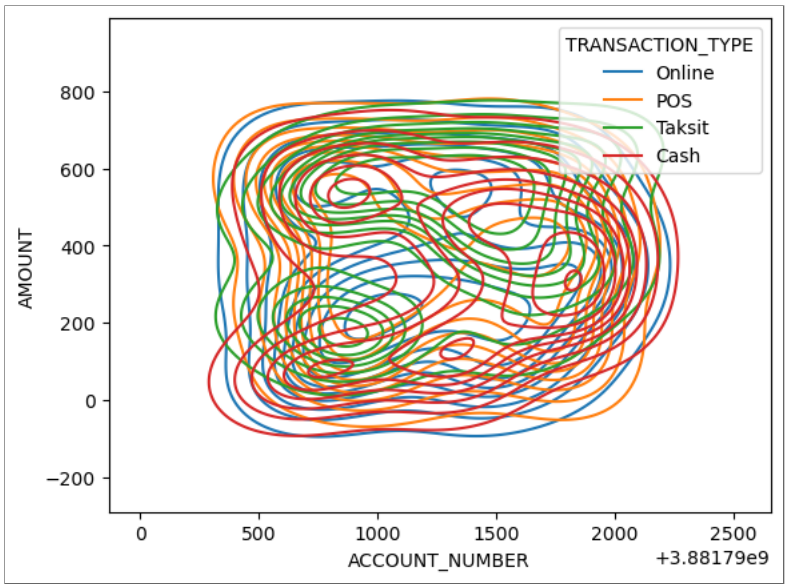
```
<AxesSubplot:xlabel='TRANSACTION_TYPE', ylabel='Count'>
```



In [17]:
```
sns.kdeplot(
    data=df_transaction,
    x="ACCOUNT_NUMBER",
    y="AMOUNT",
    hue="TRANSACTION_TYPE",
    thresh=.1,
)
```

Out[17]:

```
<AxesSubplot:xlabel='ACCOUNT_NUMBER', ylabel='AMOUNT'>
```



In [18]:
```
df = pd.read_csv(r"C:\Users\User\Documents\kpdata.csv")
df = df.drop(['CIF_1','ACCOUNT_NUMBER_1'],axis = 1)
```

In [19]:
```
df
```

Out[19]:

	CIF	BIRTH_DATE	GENDER	REGISTERED_CITY_REGION	INCOME_TYPE	AVERAGE_SALARY	AVERAGE_PENSION	CONTRACT_NUMB
0	1000084	7-Feb-67	Female	GOBUSTAN	Salary	1917	0	BRED0004
1	1000018	26-Aug-67	Male	BAKU	Salary	1048	0	BRED0002
2	1000084	7-Feb-67	Female	GOBUSTAN	Salary	1917	0	BRED0004
3	1000025	24-Feb-62	Male	NAKHCHIVAN	Pension	0	152	BBLK0006
4	1000040	5-Aug-99	Male	KHACHMAZ	Pension	0	229	BBLK0011
...	...	...	...	...	...	...	...	...
505	1000048	12-Nov-96	Male	BAKU	Pension	0	165	NaN
506	1000037	7-Aug-74	Male	BEYLAQAN	Pension	0	165	BBLK0007
507	1000089	28-Sep-77	Male	BAKU	Salary	1591	0	BRED0010
508	1000055	30-Aug-65	Male	GAKH	Pension	0	204	BBLK0022
509	1000064	17-Jan-89	Male	BAKU	Salary	1102	0	BBLK0026

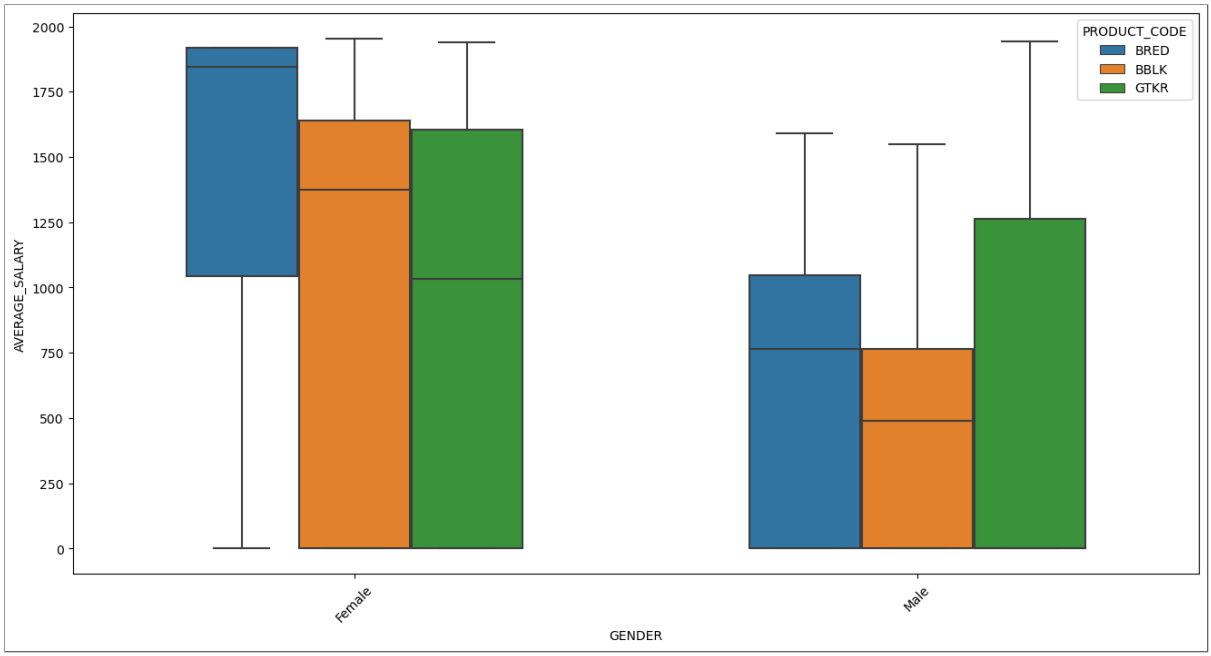
510 rows × 19 columns

In [20]:
```
df.columns
```

Out[20]:

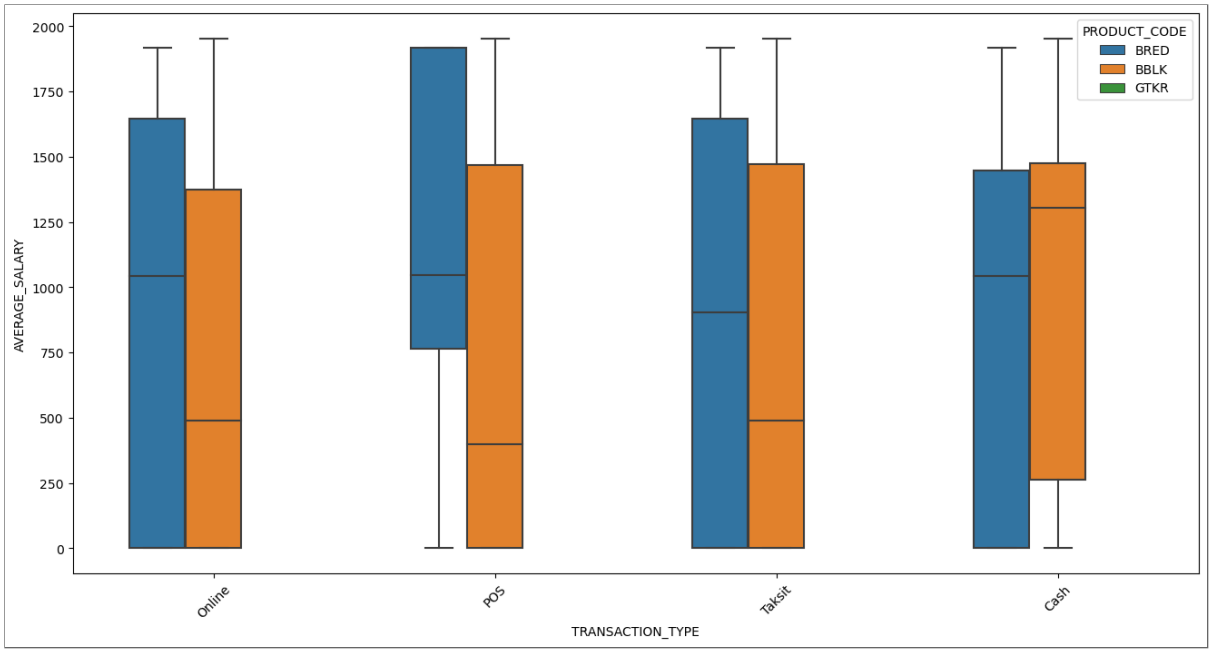
```
Index(['CIF', 'BIRTH_DATE', 'GENDER', 'REGISTERED_CITY_REGION', 'INCOM
E_TYPE',
       'AVERAGE_SALARY', 'AVERAGE_PENSION', 'CONTRACT_NUMBER', 'PRODUC
T_CODE',
       'STATUS', 'AMOUNT_LIMITS', 'INTEREST', 'DURATION', 'OPEN_DATE',
       'CLOSE_DATE', 'ACCOUNT_NUMBER', 'DATES', 'AMOUNT', 'TRANSACTION
_TYPE'],
      dtype='object')
```

In [21]:
```
plt.figure(figsize=[16,8])
sns.boxplot(x="GENDER",y = 'AVERAGE_SALARY', hue = 'PRODUCT_CODE',width=0.6,data=df)
plt.xticks(rotation=45)
plt.show()
```

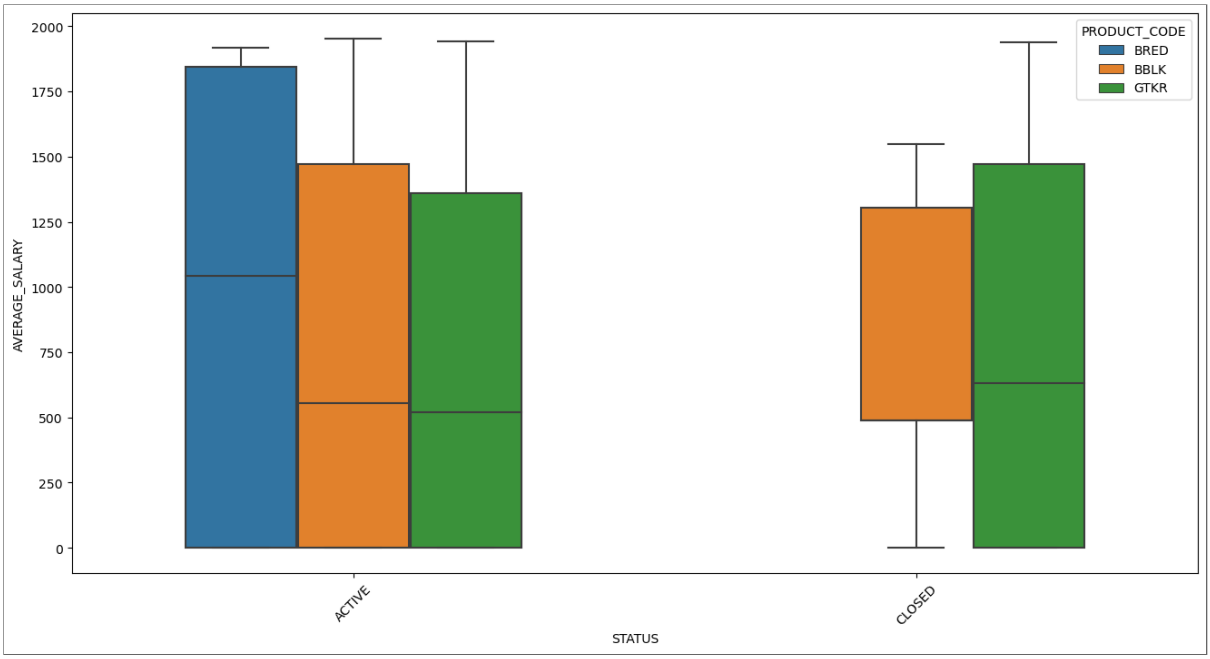


In [22]:

```
plt.figure(figsize=[16,8])
sns.boxplot(x="TRANSACTION_TYPE",y = 'AVERAGE_SALARY', hue = 'PRODUCT_CODE',width=0.6,data=df)
plt.xticks(rotation=45)
plt.show()
```



In [23]:
```
plt.figure(figsize=[16,8])
sns.boxplot(x="STATUS",y = 'AVERAGE_SALARY', hue = 'PRODUCT_CODE',width=0.6,data=df)
plt.xticks(rotation=45)
plt.show()
```



In [24]:

```
gelir_seviyyesi = []
for i in df['AVERAGE_SALARY'].iloc[:510]:
    if i < 600:
        gelir_seviyyesi.append('asagi')
    elif i > 600 and i < 1200:
        gelir_seviyyesi.append('orta')
    else:
        gelir_seviyyesi.append('yuxari')
```

In [25]:
```
df['gelir_seviyyesi'] =gelir_seviyyesi
df['gelir_seviyyesi']
```

Out[25]:

```
0      yuxari
1        orta
2      yuxari
3       asagi
4       asagi
        ...
505     asagi
506     asagi
507    yuxari
508     asagi
509      orta
Name: gelir_seviyyesi, Length: 510, dtype: object
```

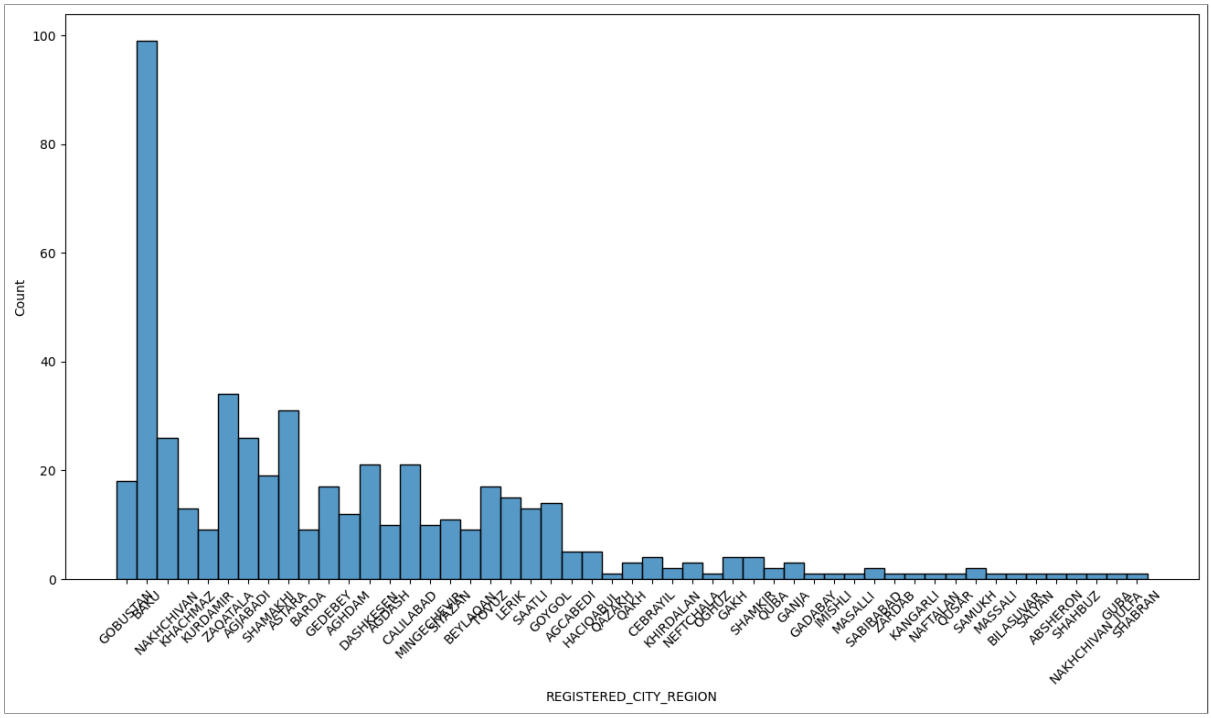
In [26]:
```
df['contract_duration'] = (pd.to_datetime(df['CLOSE_DATE']).dt.year - pd.to_datetime(df['OPEN_DATE']).dt.year)
```

In [27]:
```
"""Hansi yas araliginda hansi salary e malik olan insanlar daha cox contractda qalir,hansi regionlardaki insanlar daha
maraqlidi buna,aktiv musterilerin salaryisi nece deyisir,hansi yas qrupu ucun hansi transaction novu daha cox istifade
olunur(amounta gore),pensiya alan insanlarin maraq dairesi nedir,hansi transaction novune ustunluk verirler,ne qeder
muddet erzinde contracda qaliblar,hansi average salaryie malik olan musteriye ne qeder credit ne qeder muddete verilir,
tebiiki ne qeder interest rate i ile.Butun bunlari analiz etdikden sonra ister visuallarla ister domain knowledge ile
mueyyen neticeler cixarmaq mumkundur.Bundan elave meseleye classification meselesi kimi baxib
Machine Learning de istifade etmek olar evvelki recordlara esasen ki hansi musteriye birkart verilib hansina verilmeyib.
Machine Learning algoritmlerinden Random forest,logistic regression,boostiong,bagging modelleri ve onlarin nece performance
gosterdiyini bildirmek ucun recall,precious,accuracy ve f1 score kimi metriclerden istifade edib neticeni deyerlendirmek olar,
Ilk once ise neticeye tesir eden (netice birkartin verilib verilmemesi ) featurelar ya arasdirilib tapilmaldir (data varsa),
yoxdursa,domain knowledge esasen feature mining etmek daha elverislidir."""
```

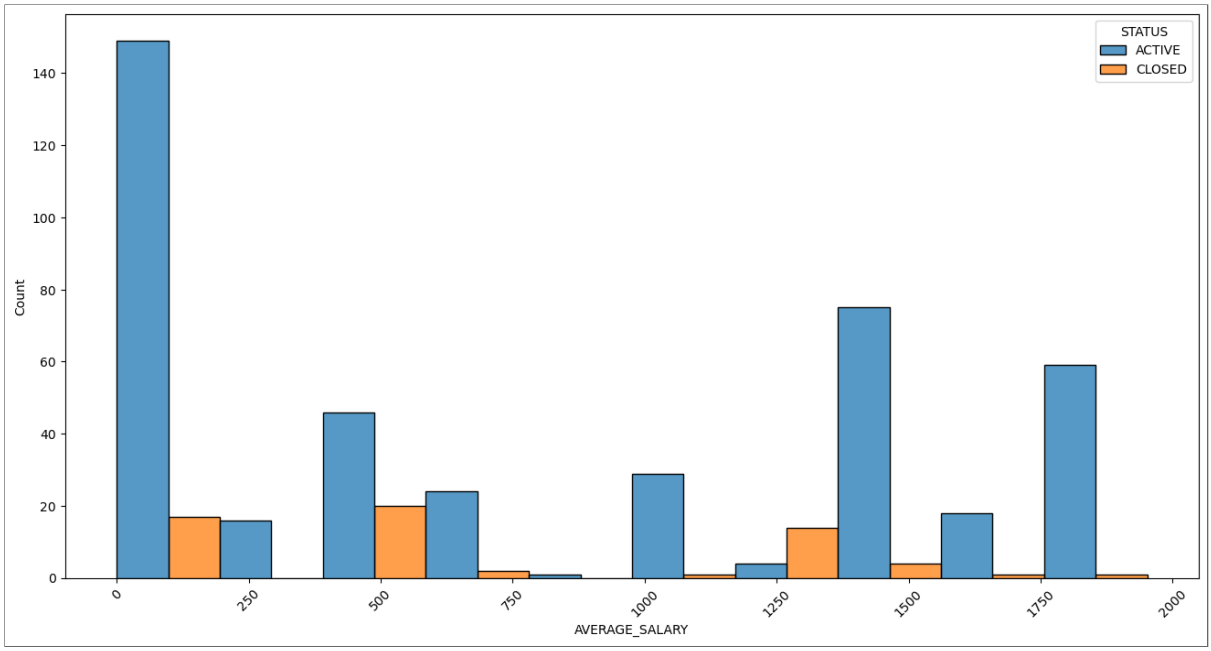
Out[27]:

```
'Hansi yas araliginda hansi salary e malik olan insanlar daha cox cont
ractda qalir,hansi regionlardaki insanlar daha \nmaraqlidi buna,aktiv
musterilerin salaryisi nece deyisir,hansi yas qrupu ucun hansi transac
tion novu daha cox istifade \nolunur(amounta gore),pensiya alan insanl
arin maraq dairesi nedir,hansi transaction novune ustunluk verirler,ne
qeder \nmuddet erzinde contracda qaliblar,hansi average salaryie malik
olan musteriye ne qeder credit ne qeder muddete verilir,\ntebiiki ne q
eder interest rate i ile.Butun bunlari analiz etdikden sonra ister vis
uallarla ister domain knowledge ile \nmueyyen neticeler cixarmaq mumku
ndur.Bundan elave meseleye classification meselesi kimi baxib \nMachin
e Learning de istifade etmek olar evvelki recordlara esasen ki hansi m
usteriye birkart verilib hansina verilmeyib.\nMachine Learning algorit
mlerinden Random forest,logistic regression,boostiong,bagging modeller
i ve onlarin nece performance \ngosterdiyini bildirmek ucun recall,pre
cious,accuracy ve f1 score kimi metriclerden istifade edib neticeni de
yerlendirmek olar,\nIlk once ise neticeye tesir eden (netice birkartin
verilib verilmemesi ) featurelar ya arasdirilib tapilmaldir (data vars
a),\nyoxdursa,domain knowledge esasen feature mining etmek daha elveri
slidir.'
```

In [28]:
```
plt.figure(figsize=[16,8])
sns.histplot(data = df,x = 'REGISTERED_CITY_REGION')
plt.xticks(rotation=45)
plt.show()
```

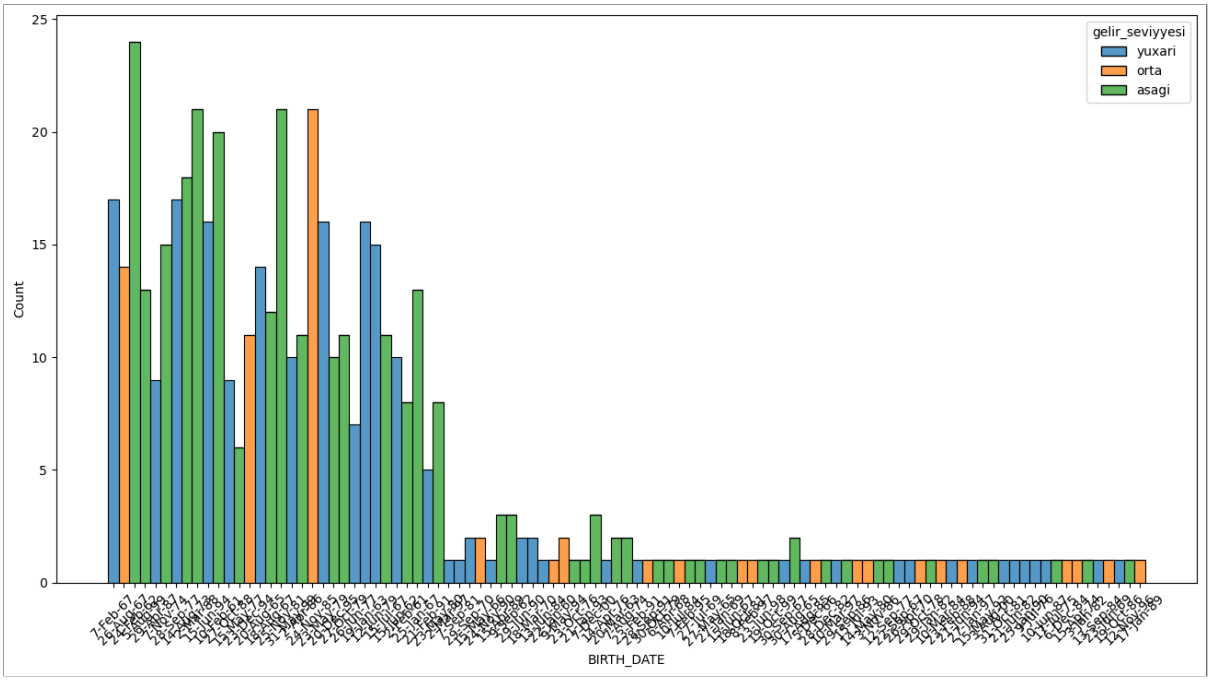


In [29]:
```
plt.figure(figsize=[16,8])
sns.histplot(data = df,x = 'AVERAGE_SALARY',hue = 'STATUS',multiple="dodge")
plt.xticks(rotation=45)
plt.show()
```



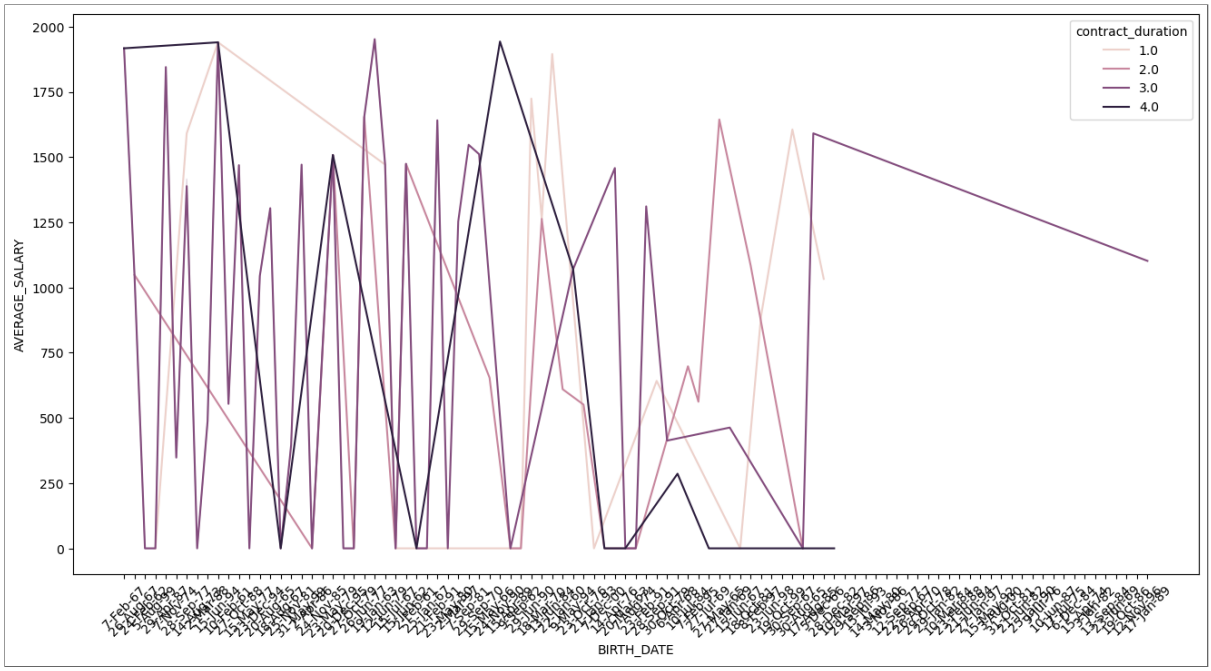
In [30]:

```
plt.figure(figsize=[16,8])
sns.histplot(data = df,x = 'BIRTH_DATE',hue = 'gelir_seviyyesi',multiple="stack")
plt.xticks(rotation=45)
plt.show()
```



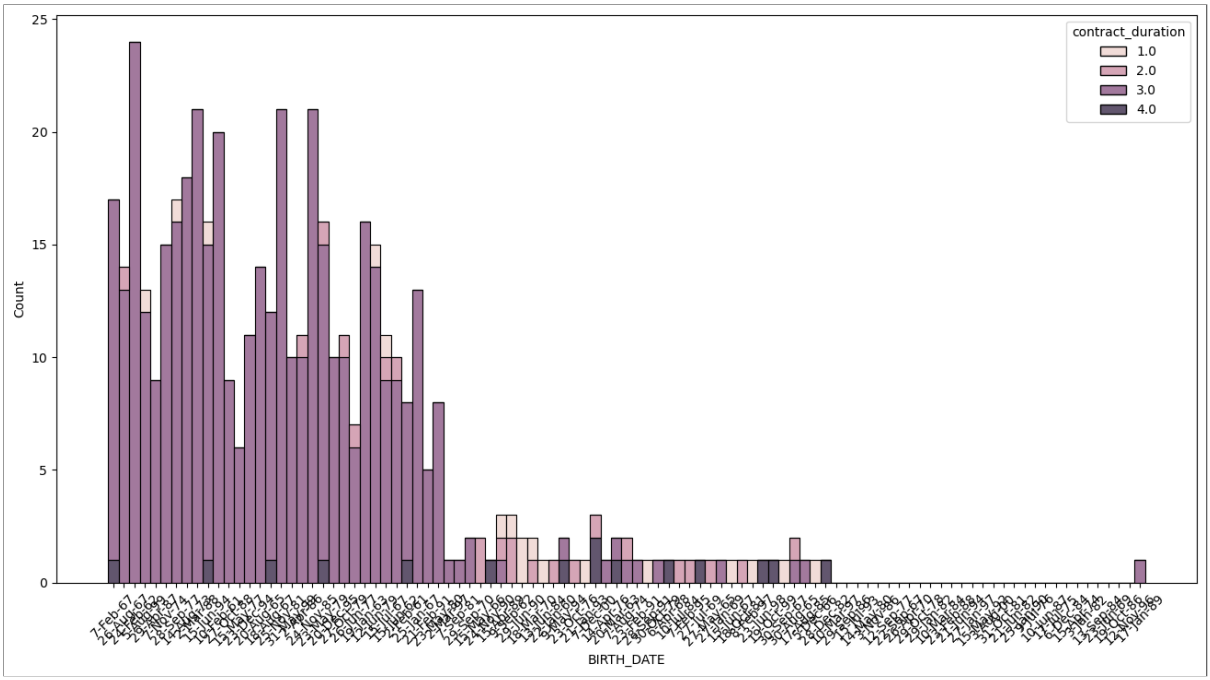
In [31]:

```
plt.figure(figsize=[16,8])
sns.lineplot(data = df,x = 'BIRTH_DATE',y = 'AVERAGE_SALARY',hue = 'contract_duration')
plt.xticks(rotation=45)
plt.show()
```

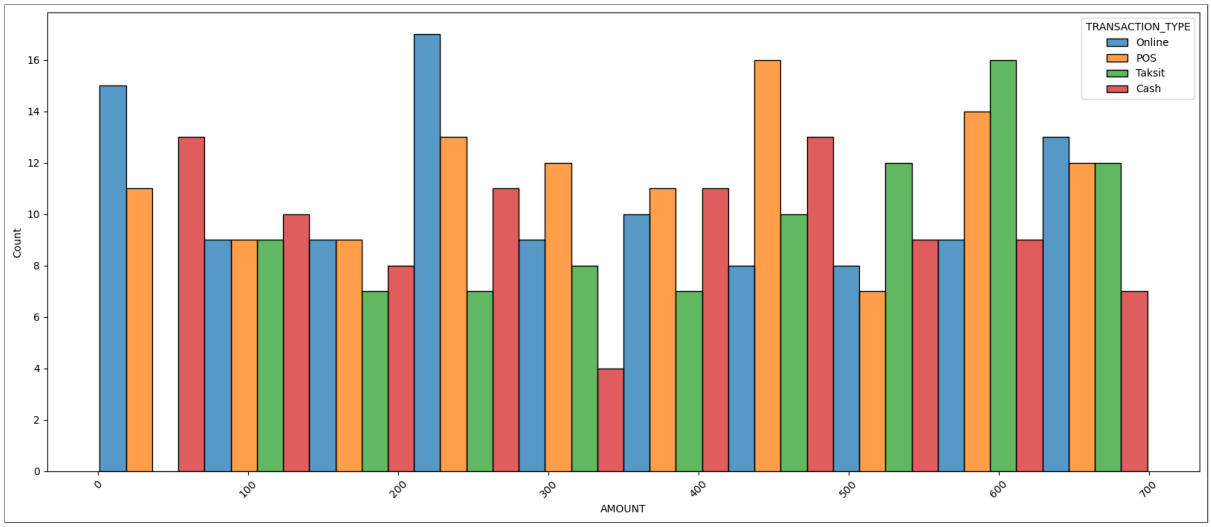


In [32]:

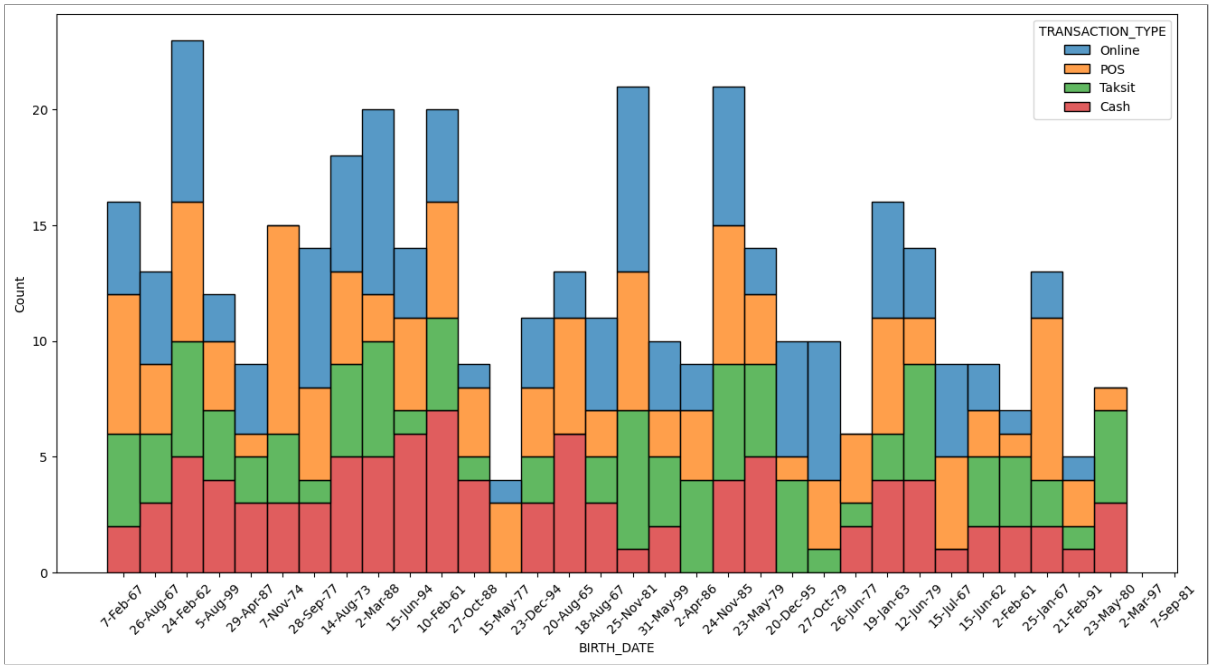
```
plt.figure(figsize=[16,8])
sns.histplot(data = df,x = 'BIRTH_DATE',hue = 'contract_duration',multiple="stack")
plt.xticks(rotation=45)
plt.show()
```



In [33]:
```
plt.figure(figsize=[20,8])
sns.histplot(data = df,x = 'AMOUNT',hue = 'TRANSACTION_TYPE',multiple="dodge")
plt.xticks(rotation=45)
plt.show()
```



In [34]:
```
plt.figure(figsize=[16,8])
sns.histplot(data = df,x = 'BIRTH_DATE',hue = 'TRANSACTION_TYPE',multiple="stack")
plt.xticks(rotation=45)
plt.show()
```



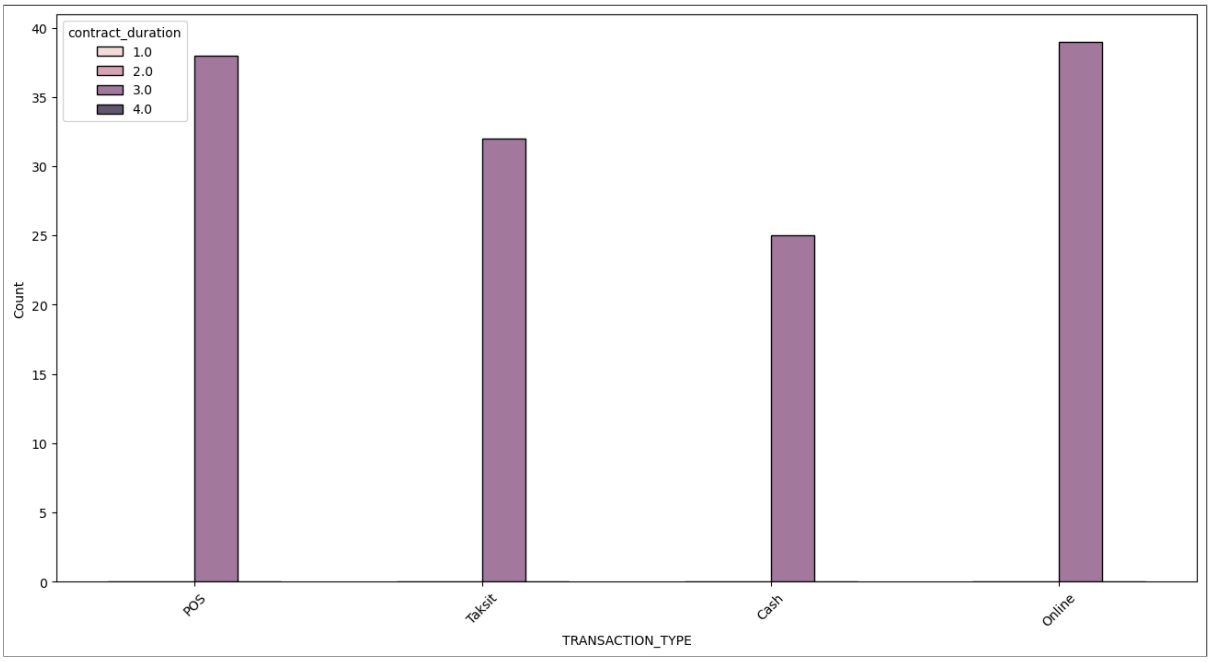
In [35]:
```
df[df['AVERAGE_PENSION'] != 0]
```

Out[35]:

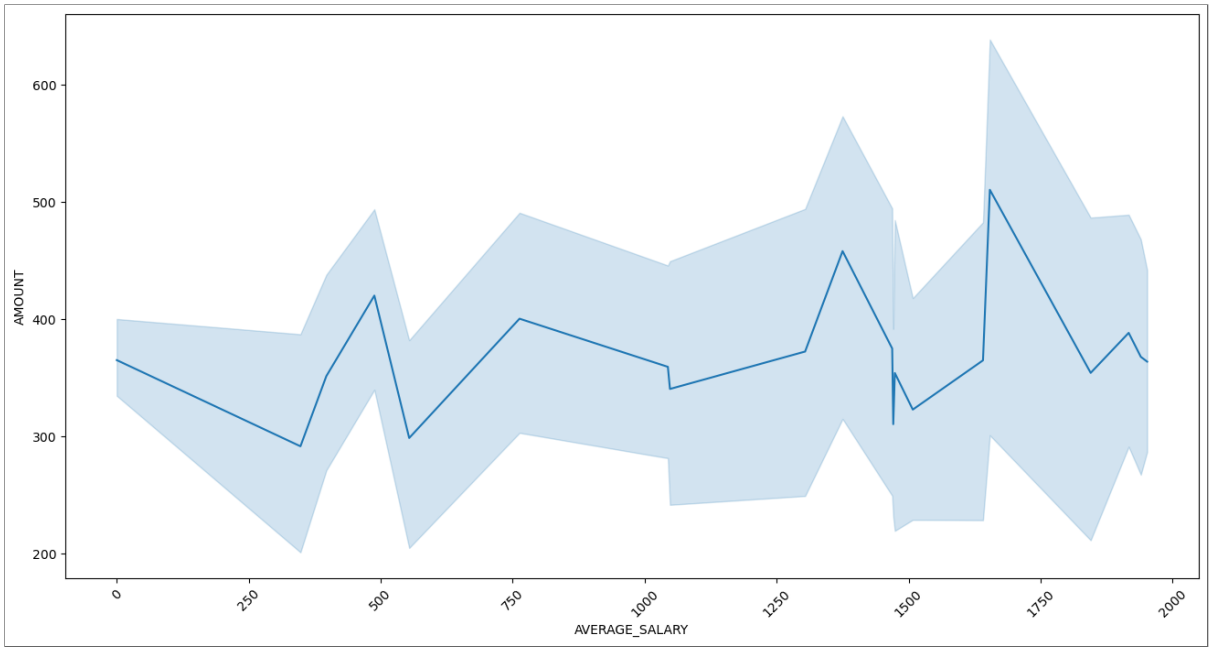
	CIF	BIRTH_DATE	GENDER	REGISTERED_CITY_REGION	INCOME_TYPE	AVERAGE_SALARY	AVERAGE_PENSION	CONTRACT_NUMBI
3	1000025	24-Feb-62	Male	NAKHCHIVAN	Pension	0	152	BBLK0006
4	1000040	5-Aug-99	Male	KHACHMAZ	Pension	0	229	BBLK0011
8	1000040	5-Aug-99	Male	KHACHMAZ	Pension	0	229	BBLK0011
9	1000087	14-Aug-73	Female	SHAMAKHI	Pension	0	311	BBLK003C
11	1000087	14-Aug-73	Female	SHAMAKHI	Pension	0	311	BBLK003C
...	...	...	...	...	...	...	...	..
498	1000014	17-Jul-75	Female	TOVUZ	Pension	0	150	NaN
501	1000017	3-Jan-82	Male	GAKH	Pension	0	356	NaN
505	1000048	12-Nov-96	Male	BAKU	Pension	0	165	NaN
506	1000037	7-Aug-74	Male	BEYLAQAN	Pension	0	165	BBLK0007
508	1000055	30-Aug-65	Male	GAKH	Pension	0	204	BBLK0022

172 rows × 21 columns

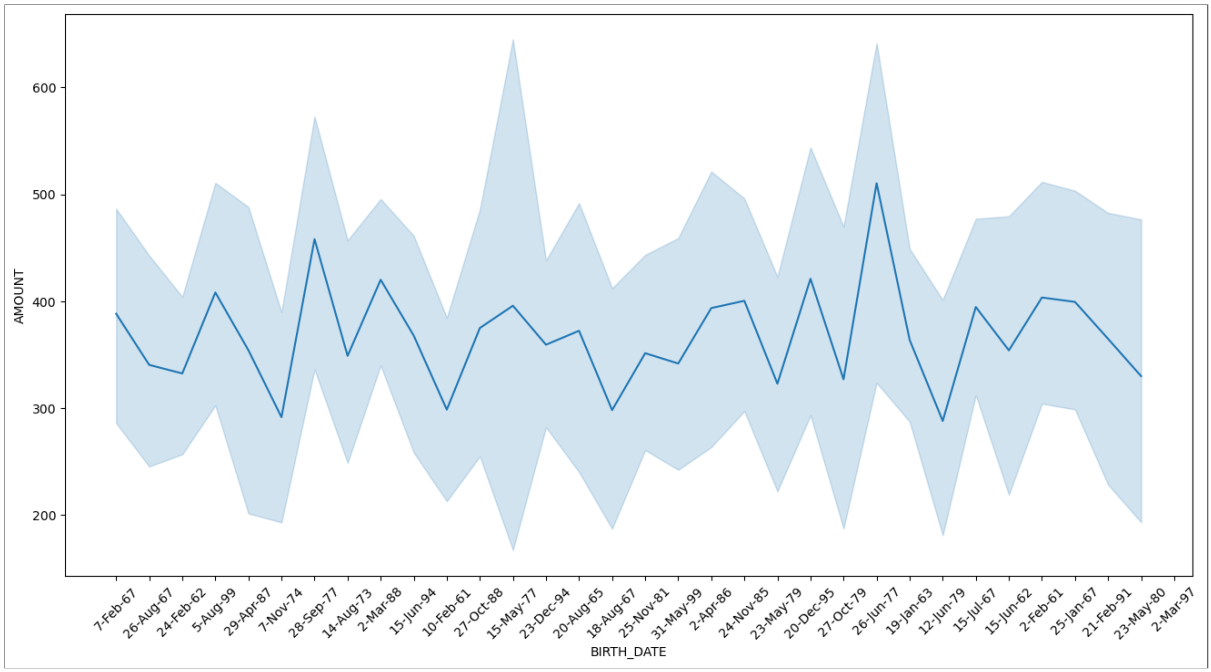
In [36]:
```
#pensiyacilar hamisi 3 illik muqaile bagliyib.
plt.figure(figsize=[16,8])
sns.histplot(data = df[df['AVERAGE_PENSION'] != 0],x = 'TRANSACTION_TYPE',hue = 'contract_duration',multiple="dodge",shrink=.6)
plt.xticks(rotation=45)
plt.show()
```



In [37]:
```
plt.figure(figsize=[16,8])
sns.lineplot(data = df,x = 'AVERAGE_SALARY',y = 'AMOUNT')
plt.xticks(rotation=45)
plt.show()
```



In [38]:
```
plt.figure(figsize=[16,8])
sns.lineplot(data = df,x = 'BIRTH_DATE',y = 'AMOUNT')
plt.xticks(rotation=45)
plt.show()
```

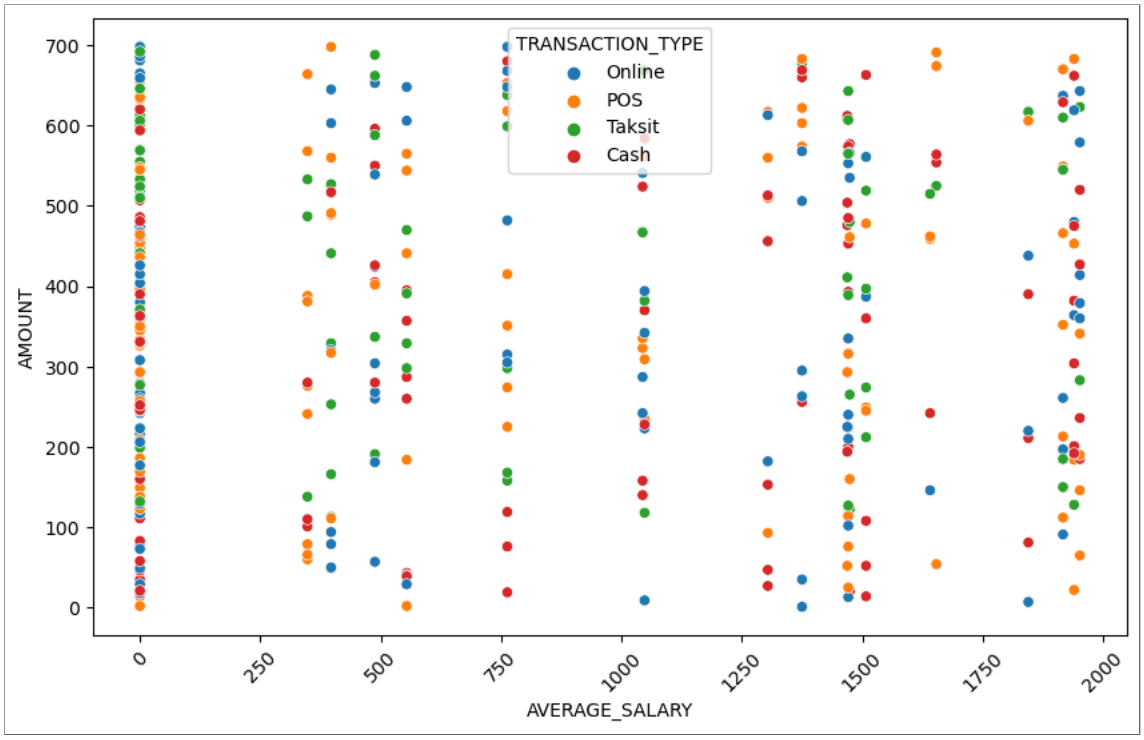


In [39]:
```
df.corr()
```

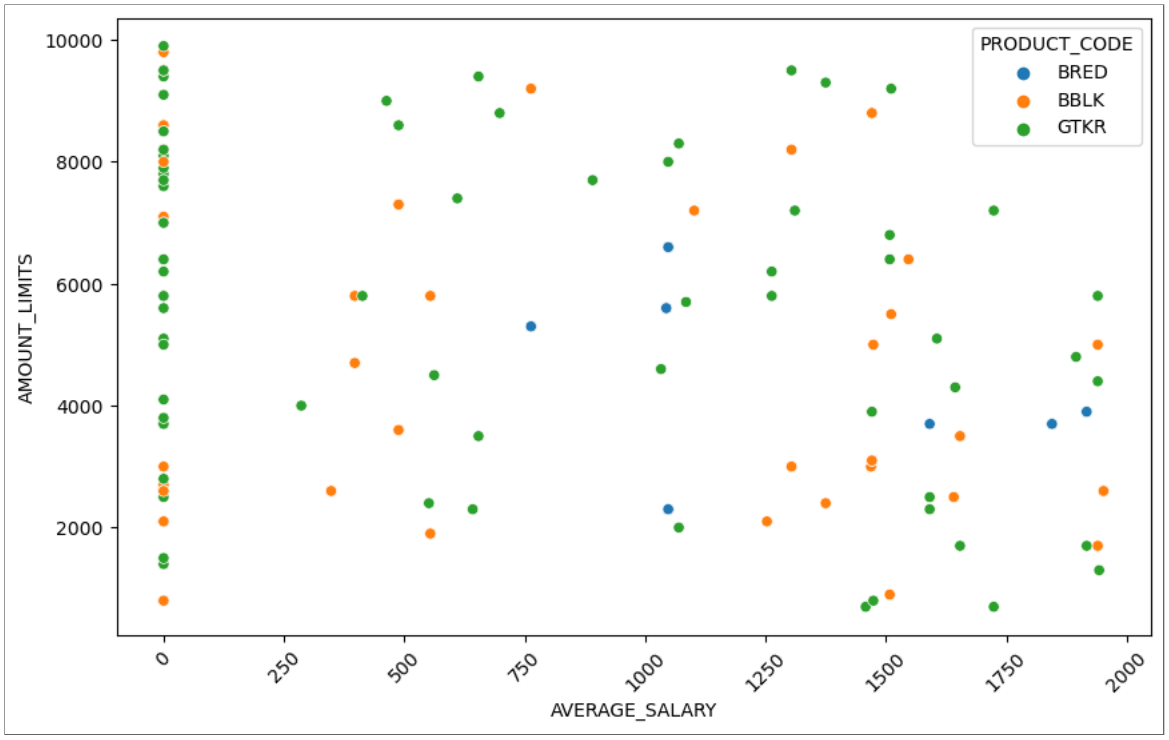
Out[39]:

	CIF	AVERAGE_SALARY	AVERAGE_PENSION	AMOUNT_LIMITS	INTEREST	DURATION	AC
CIF	1.000000	0.252560	-0.087841	-0.210024	0.008604	-0.001789	-0.
AVERAGE_SALARY	0.252560	1.000000	-0.725823	-0.307181	-0.074327	-0.027789	-0.
AVERAGE_PENSION	-0.087841	-0.725823	1.000000	0.179566	0.074336	0.031578	0.
AMOUNT_LIMITS	-0.210024	-0.307181	0.179566	1.000000	0.035232	-0.035248	0.
INTEREST	0.008604	-0.074327	0.074336	0.035232	1.000000	0.724052	
DURATION	-0.001789	-0.027789	0.031578	-0.035248	0.724052	1.000000	
ACCOUNT_NUMBER	-0.309860	-0.067824	0.131324	0.039680	NaN	NaN	1.
AMOUNT	0.011785	0.017616	0.019304	0.049672	NaN	NaN	0.
contract_duration	-0.001789	-0.027789	0.031578	-0.035248	0.724052	1.000000	

In [40]:
```
plt.figure(figsize=[10,6])
sns.scatterplot(x="AVERAGE_SALARY",y = 'AMOUNT', hue = 'TRANSACTION_TYPE',data=df)
plt.xticks(rotation=45)
plt.show()
```



In [41]:
```
plt.figure(figsize=[10,6])
sns.scatterplot(x="AVERAGE_SALARY",y = 'AMOUNT_LIMITS', hue = 'PRODUCT_CODE',data=df)
plt.xticks(rotation=45)
plt.show()
```



In [ ]:

In [ ]:

In [42]:
```
jupyter nbconvert kapital.ipynb --to slides --post serve
```

```
  File "C:\Users\User\AppData\Local\Temp\ipykernel_8672\1411504507.p
y", line 1
    jupyter nbconvert kapital.ipynb --to slides --post serve
            ^
SyntaxError: invalid syntax
```

In [ ]: