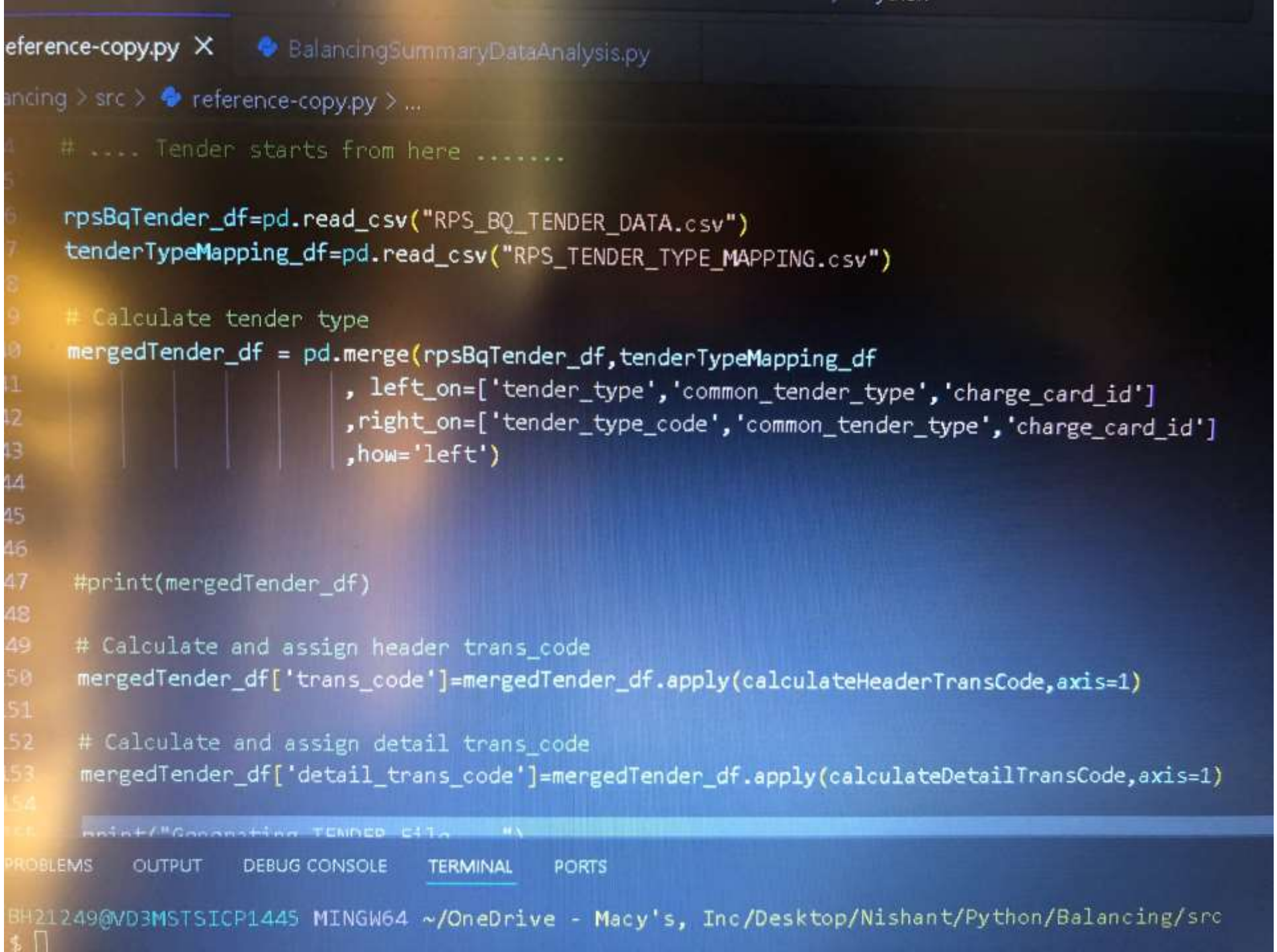

eference-copy.py
BalancingSummaryDataAnalysis.py
ancing > src > reference-copy.py > ...
# .... Tender starts from here .....
rpsBqTender_df=pd.read_csv("RPS_BQ_TENDER_DATA.csv")
tenderTypeMapping_df=pd.read_csv("RPS_TENDER_TYPE_MAPPING.csv")
# Calculate tender type
mergedTender_df = pd.merge(rpsBqTender_df,tenderTypeMapping_df
, left_on=['tender_type','common_tender_type','charge_card_id']
,right_on=['tender_type_code','common_tender_type','charge_card_id']
,how='left')
#print(mergedTender_df)
Calculate and assign header trans_code
mergedTender_df['trans_code']=mergedTender_df.apply(calculateHeaderTransCode,axis=1)
Calculate and assign detail trans_code
mergedTender_df['detail_trans_code']=mergedTender_df.apply(calculateDetailTransCode,axis=1)
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
BH21249@VD3MSTSICP1445 MINGW64 ~/OneDrive - Macy's, Inc/Desktop/Nishant/Python/Balancing/src
$

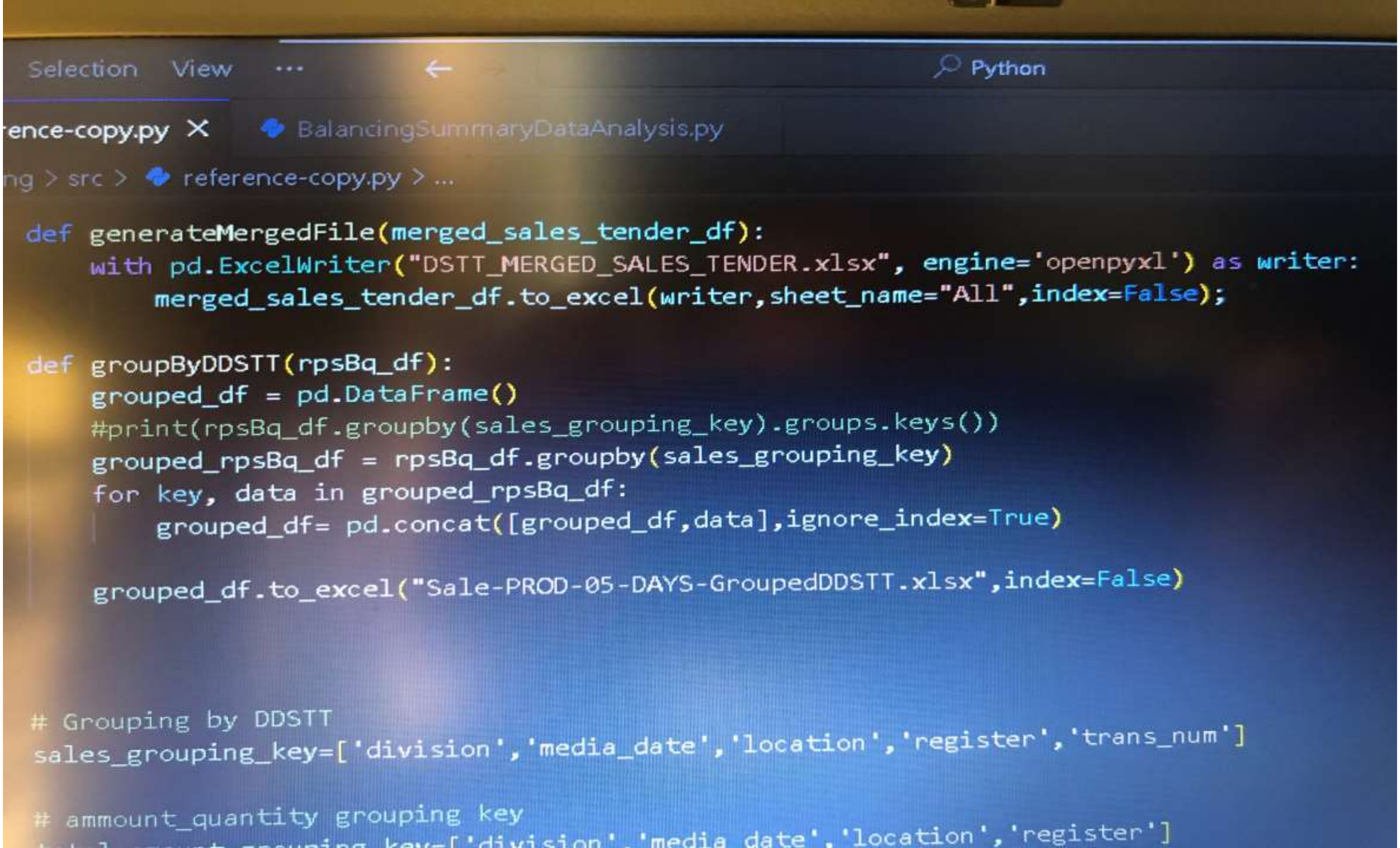
reference-copy.py

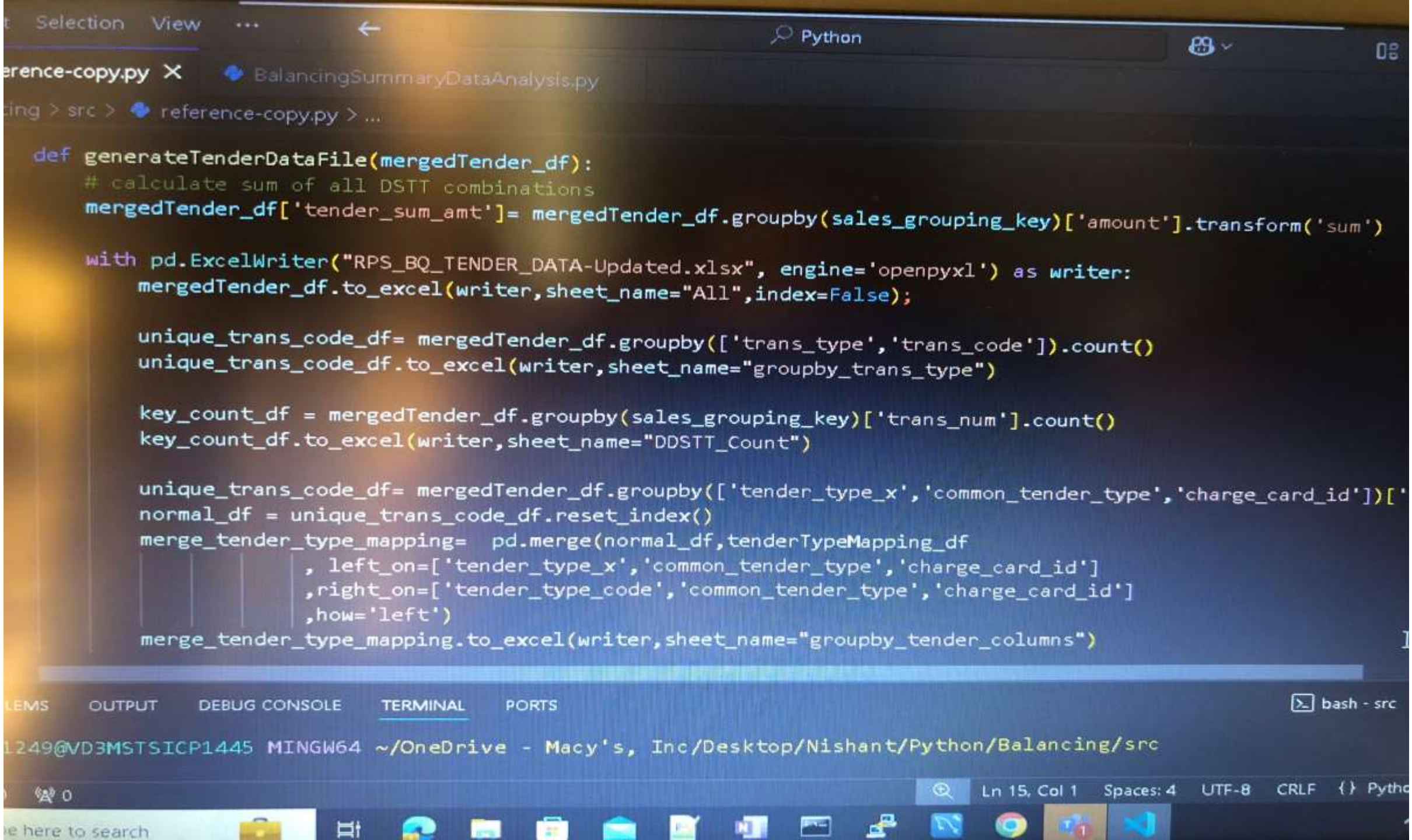
lancing > src > reference-copy.py > ...

```
#rpsBq_df=pd.read_excel("Sale-400Days-PROD.xlsx");
rpsBq_df=pd.read_csv("Sale-PROD-05-DAYS.csv");

# Calculate and assign header trans_code
rpsBq_df['trans_code']=rpsBq_df.apply(calculateHeaderTransCode,axis=1)

# Calculate and assign detail trans_code
rpsBq_df['detail_trans_code']=rpsBq_df.apply(calculateDetailTransCode,axis=1)

# Calculate and shipment sale_amt
rpsBq_df['shipment_sale_amt']=rpsBq_df.apply(calculateShipmentSaleAmt,axis=1)

# Calculate and shipment return_amt
rpsBq_df['shipment_return_amt']=rpsBq_df.apply(calculateShipmentReturnAmt,axis=1)

# Add column amount*quantity
rpsBq_df['amount_quantity']=rpsBq_df['amount']*rpsBq_df['quantity']

# sort by key
# rpsBq_df.sort_values(by=sales_grouping_key)

#print(rpsBq_df);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Inc/Desktop/Nishant/Python/Balancin


Selection View ···
Python
rence-copy.py
BalancingSummaryDataAnalysis.py
ng > src > reference-copy.py > ...
def generateMergedFile(merged_sales_tender_df):
with pd.ExcelWriter("DSTT_MERGED_SALES_TENDER.xlsx", engine='openpyxl') as writer:
merged_sales_tender_df.to_excel(writer,sheet_name="All",index=False);
def groupByDDSTT(rpsBq_df):
grouped_df = pd.DataFrame()
#print(rpsBq_df.groupby(sales_grouping_key).groups.keys())
grouped_rpsBq_df = rpsBq_df.groupby(sales_grouping_key)
for key, data in grouped_rpsBq_df:
grouped_df= pd.concat([grouped_df,data],ignore_index=True)
grouped_df.to_excel("Sale-PROD-05-DAYS-GroupedDDSTT.xlsx",index=False)
# Grouping by DDSTT
sales_grouping_key=['division','media_date','location','register','trans_num']
# ammount_quantity grouping key


Selection View ...
Python
erence-copy.py
BalancingSummaryDataAnalysis.py
cing > src > reference-copy.py > ...
def generateTenderDataFile(mergedTender_df):
# calculate sum of all DSTT combinations
mergedTender_df['tender_sum_amt']= mergedTender_df.groupby(sales_grouping_key)['amount'].transform('sum')
with pd.ExcelWriter("RPS_BQ_TENDER_DATA-Updated.xlsx", engine='openpyxl') as writer:
mergedTender_df.to_excel(writer,sheet_name="All",index=False);
unique_trans_code_df= mergedTender_df.groupby(['trans_type','trans_code']).count()
unique_trans_code_df.to_excel(writer,sheet_name="groupby_trans_type")
key_count_df = mergedTender_df.groupby(sales_grouping_key)['trans_num'].count()
key_count_df.to_excel(writer,sheet_name="DDSTT_Count")
unique_trans_code_df= mergedTender_df.groupby(['tender_type_x','common_tender_type','charge_card_id'])['
normal_df = unique_trans_code_df.reset_index()
merge_tender_type_mapping= pd.merge(normal_df,tenderTypeMapping_df
, left_on=['tender_type_x','common_tender_type','charge_card_id']
,right_on=['tender_type_code','common_tender_type','charge_card_id']
,how='left')
merge_tender_type_mapping.to_excel(writer,sheet_name="groupby_tender_columns")
LEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash - src
1249@VD3MSTSICP1445 MINGW64 ~/OneDrive - Macy's, Inc/Desktop/Nishant/Python/Balancing/src
Ln 15, Col 1 Spaces: 4 UTF-8 CRLF
e here to search

ference-copy.py ✕ BalancingSummaryDataAnalysis.py

cing > src > reference-copy.py > ...

```
def generateSalesFile(rpsBq_df):
    with pd.ExcelWriter("Sale-PROD-05-DAYS-Updated.xlsx", engine='openpyxl') as writer:
        rpsBq_df.to_excel(writer,sheet_name="All",index=False);

        key_count_df = rpsBq_df.groupby(sales_grouping_key)['trans_num'].count()
        key_count_df.to_excel(writer,sheet_name="DDSTT_Count")

        unique_trans_code_df= rpsBq_df.groupby(['trans_type','trans_code'])['item_key'].count()
        unique_trans_code_df.to_excel(writer,sheet_name="groupby_trans_type")

        groupby_loc_reg= rpsBq_df.groupby(['location','register'])['item_key'].count()
        groupby_loc_reg.to_excel(writer,sheet_name='groupby_loc_reg')

        groupby_sum_amount= rpsBq_df.groupby(total_amount_grouping_key)['amount_quantity'].sum()
        groupby_sum_amount.to_excel(writer,sheet_name='groupby_sum_amount')

def generateTenderDataFile(mergedTender_df):
    # calculate sum of all DSTT combinations
    mergedTender_df['tender_sum_amt']= mergedTender_df.groupby(sales_grouping_key)['amount'].transform
```

BLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

21249@VD3MSTSICP1445 MINGW64 ~/OneDrive - Macy's, Inc/Desktop/Nishant/Python/Balancing/src

0 0 Ln 15, Col 1 Spaces: 4 UTF-8 CR

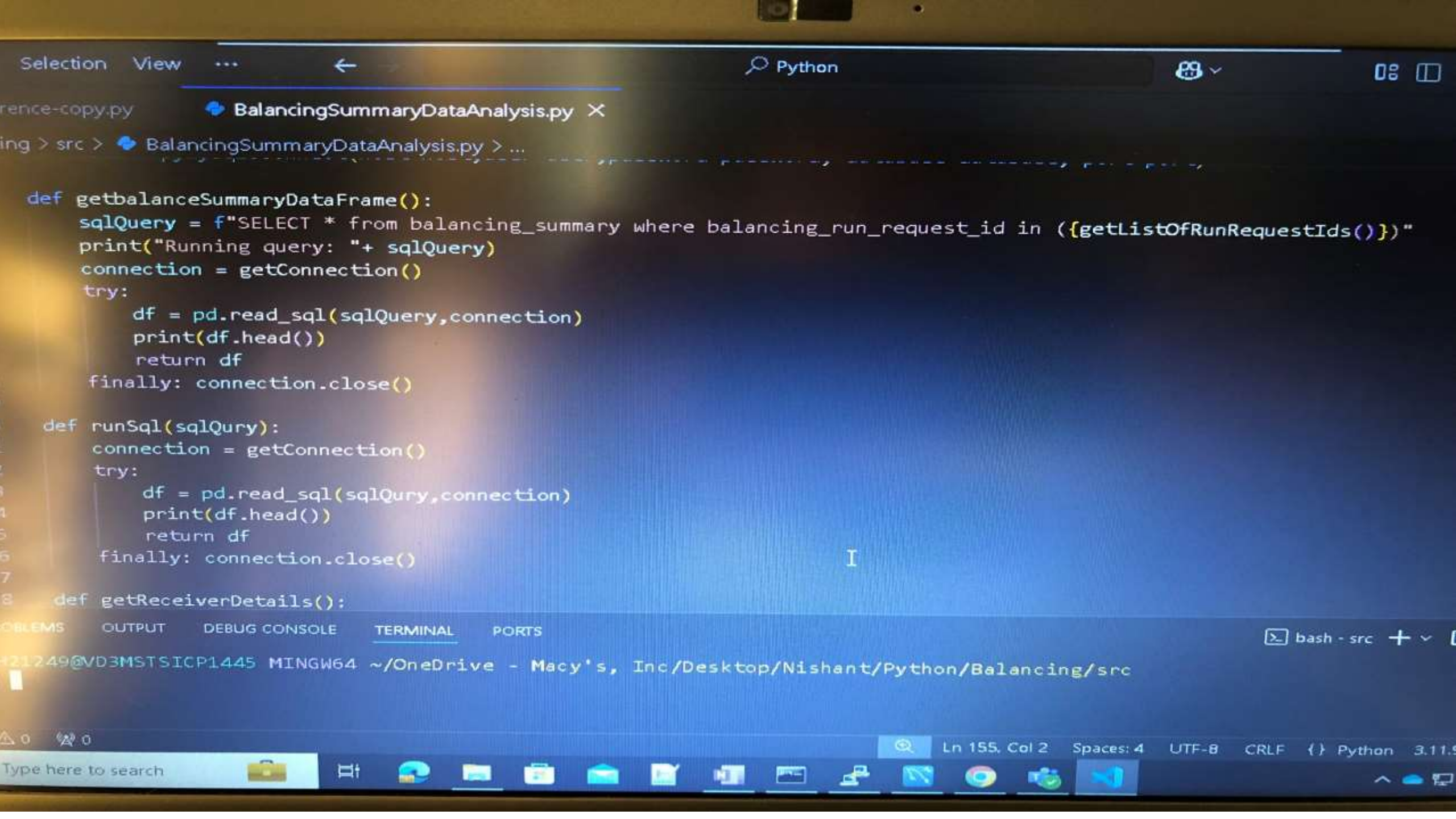
ile Edit Selection View ··· ← Python

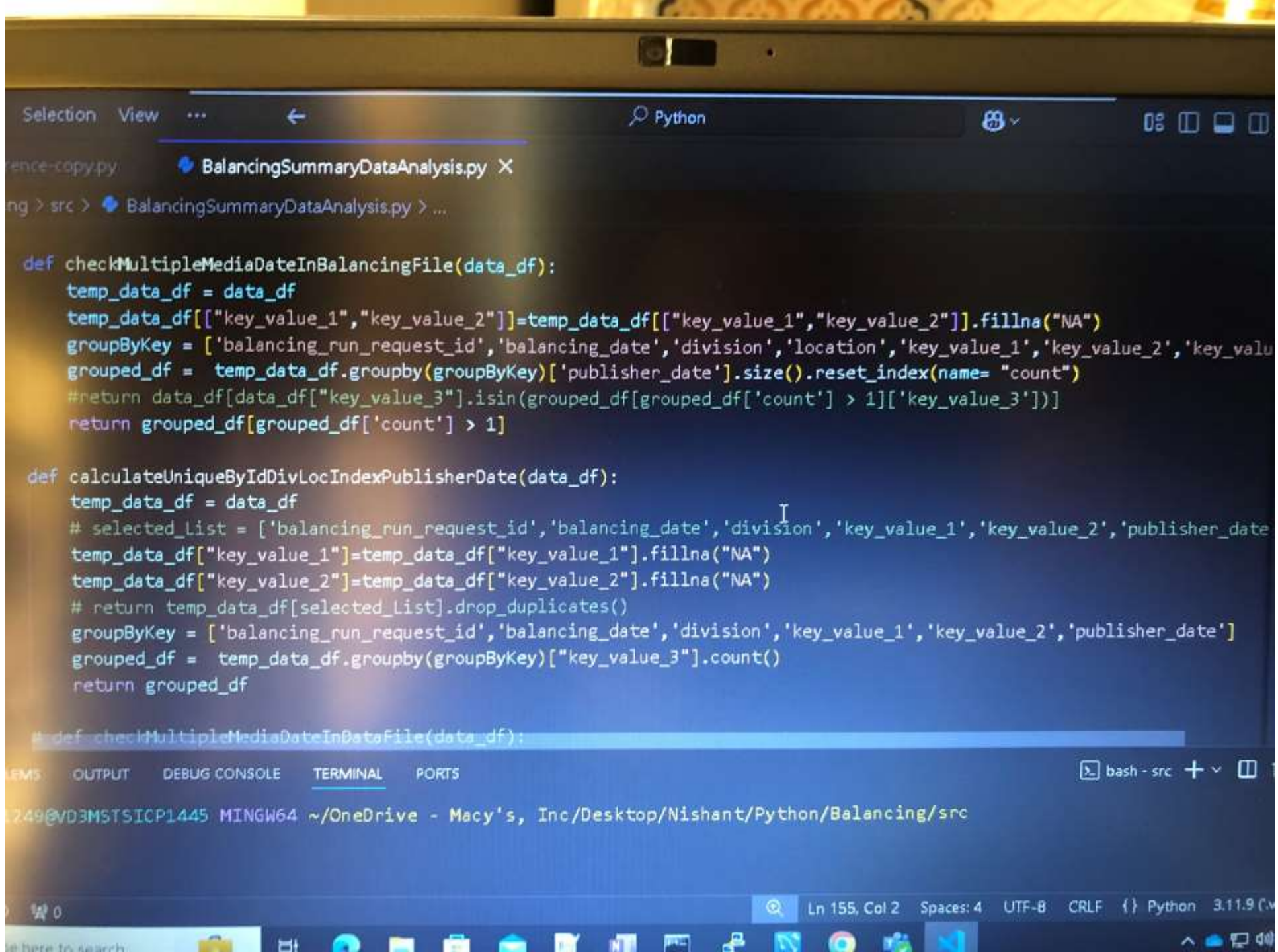
reference-copy.py | BalancingSummaryDataAnalysis.py ×

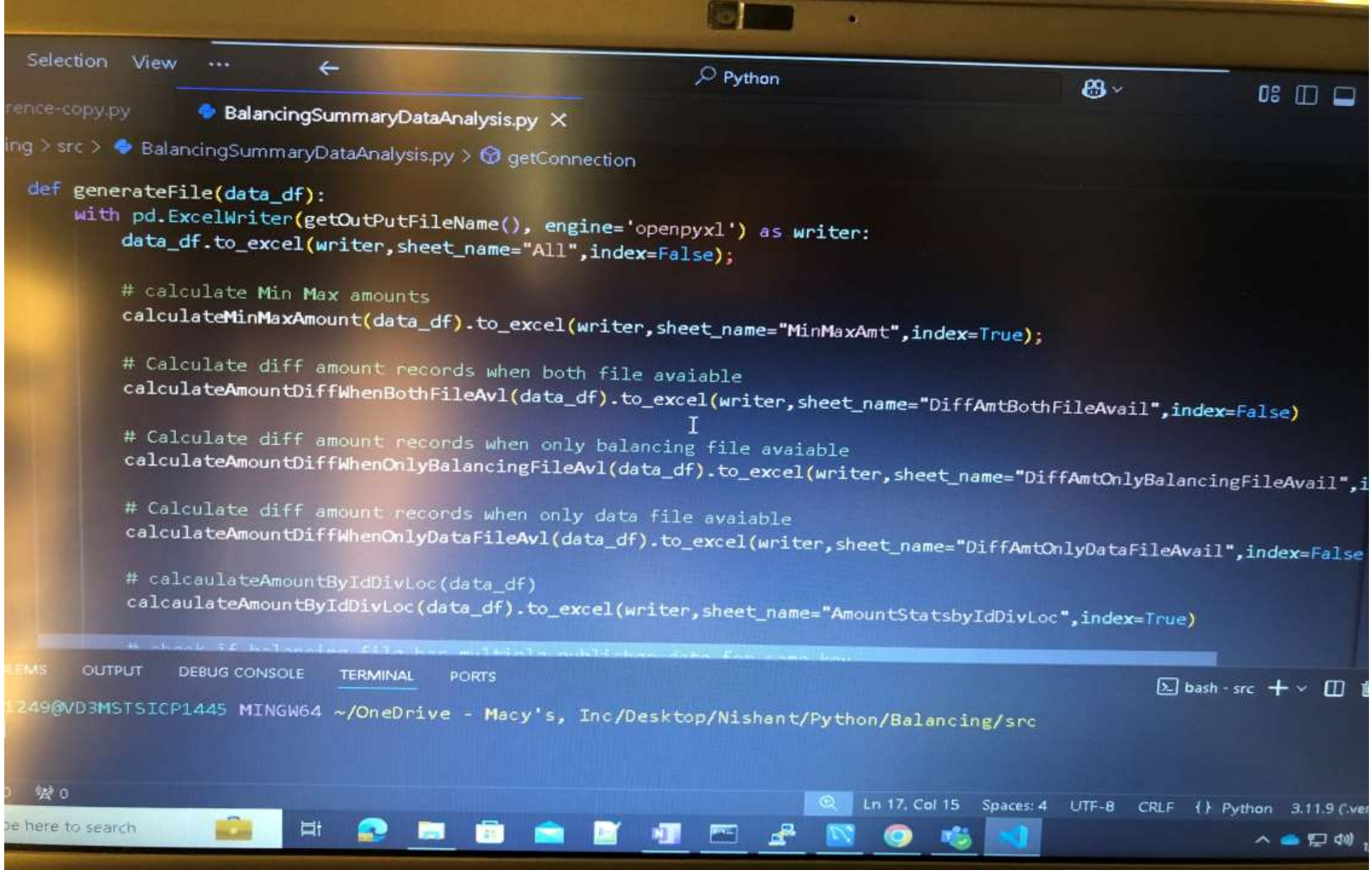
Balancing > src > BalancingSummaryDataAnalysis.py > ...

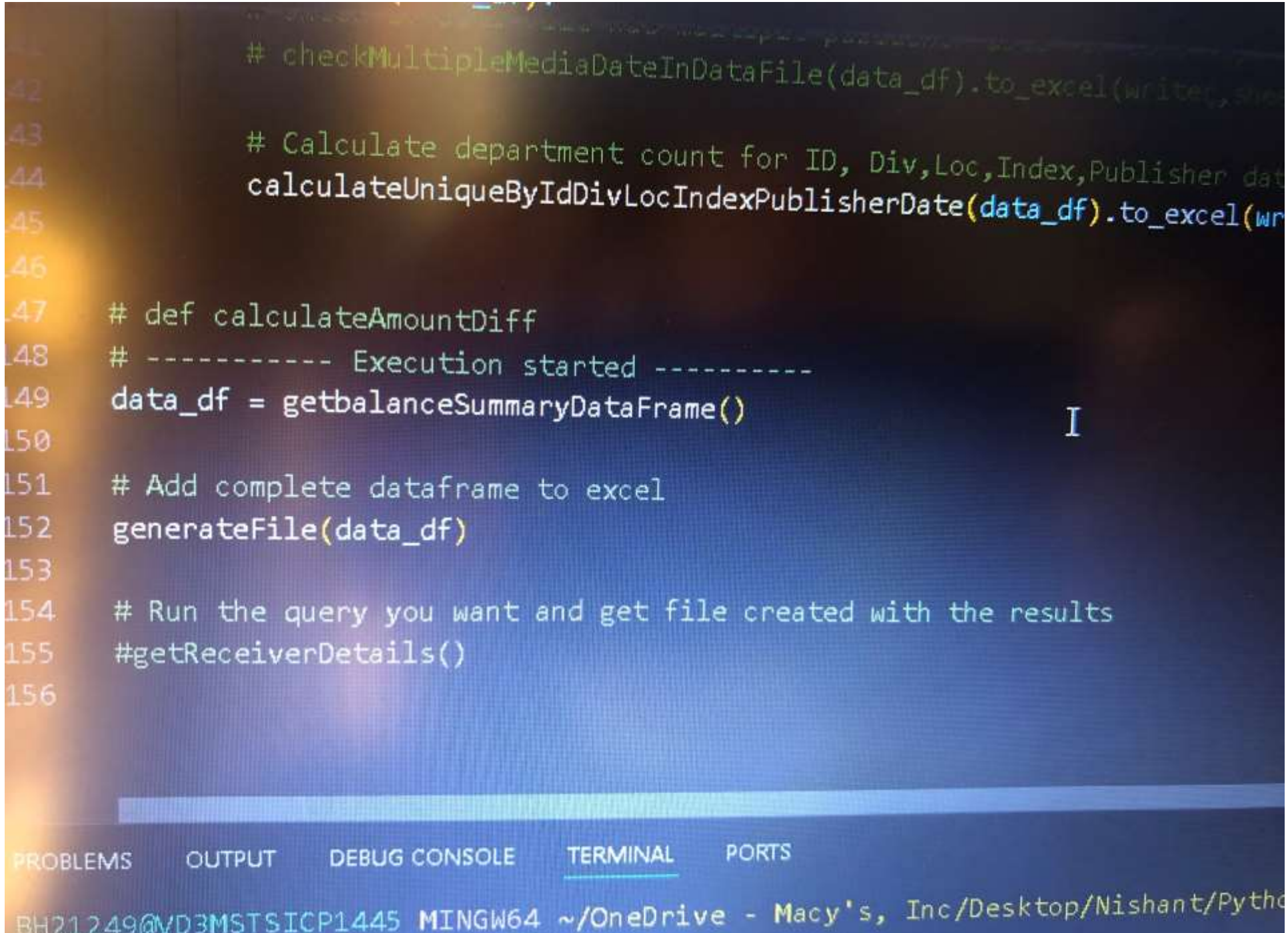
```python
import pandas as pd
import pymysql
import sys
#import openpyxl

def getOutPutFileName():
    return "BalanceSummaryOutPutFile.xlsx"

def getListOfRunRequestIds():
    return sys.argv[1]

def getConnection():
    host = "127.0.0.1"
    user = "osr_user"
    password = "Z3cp1mTk"
    database= "osr"
    port= 6708
    return pymysql.connect(host=host,user=user,password=password, database=databas
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS


Selection View ···
Python
rence-copy.py
BalancingSummaryDataAnalysis.py
ing > src > BalancingSummaryDataAnalysis.py > ...
def getbalanceSummaryDataFrame():
sqlQuery = f"SELECT * from balancing_summary where balancing_run_request_id in ({getListOfRunRequestIds()})"
print("Running query: "+ sqlQuery)
connection = getConnection()
try:
df = pd.read_sql(sqlQuery,connection)
print(df.head())
return df
finally: connection.close()
def runSql(sqlQury):
connection = getConnection()
try:
df = pd.read_sql(sqlQury,connection)
print(df.head())
return df
finally: connection.close()
def getReceiverDetails():
OBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash - src
421249@VD3MSTSICP1445 MINGW64 ~/OneDrive - Macy's, Inc/Desktop/Nishant/Python/Balancing/src
Ln 155, Col 2 Spaces: 4 UTF-8 CRLF {} Python 3.11.
Type here to search


Selection View ...
Python
BalancingSummaryDataAnalysis.py
ng > src > BalancingSummaryDataAnalysis.py > ...
def checkMultipleMediaDateInBalancingFile(data_df):
temp_data_df = data_df
temp_data_df[["key_value_1","key_value_2"]]=temp_data_df[["key_value_1","key_value_2"]].fillna("NA")
groupByKey = ['balancing_run_request_id','balancing_date','division','location','key_value_1','key_value_2','key_valu
grouped_df = temp_data_df.groupby(groupByKey)['publisher_date'].size().reset_index(name= "count")
#return data_df[data_df["key_value_3"].isin(grouped_df[grouped_df['count'] > 1]['key_value_3'])]
return grouped_df[grouped_df['count'] > 1]
def calculateUniqueByIdDivLocIndexPublisherDate(data_df):
temp_data_df = data_df
# selected_List = ['balancing_run_request_id','balancing_date','division','key_value_1','key_value_2','publisher_date
temp_data_df["key_value_1"]=temp_data_df["key_value_1"].fillna("NA")
temp_data_df["key_value_2"]=temp_data_df["key_value_2"].fillna("NA")
# return temp_data_df[selected_List].drop_duplicates()
groupByKey = ['balancing_run_request_id','balancing_date','division','key_value_1','key_value_2','publisher_date']
grouped_df = temp_data_df.groupby(groupByKey)["key_value_3"].count()
return grouped_df
#def checkMultipleMediaDateInDatafile(data_df):
OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash - src
249@VD3MSTSICP1445 MINGW64 ~/OneDrive - Macy's, Inc/Desktop/Nishant/Python/Balancing/src
Ln 155, Col 2 Spaces: 4 UTF-8 CRLF {} Python 3.11.9


Selection View ...
Python
rence-copy.py
BalancingSummaryDataAnalysis.py
ing > src > BalancingSummaryDataAnalysis.py > getConnection
def generateFile(data_df):
with pd.ExcelWriter(getOutPutFileName(), engine='openpyxl') as writer:
data_df.to_excel(writer,sheet_name="All",index=False);
# calculate Min Max amounts
calculateMinMaxAmount(data_df).to_excel(writer,sheet_name="MinMaxAmt",index=True);
# Calculate diff amount records when both file avaiable
calculateAmountDiffWhenBothFileAvl(data_df).to_excel(writer,sheet_name="DiffAmtBothFileAvail",index=False)
# Calculate diff amount records when only balancing file avaiable
calculateAmountDiffWhenOnlyBalancingFileAvl(data_df).to_excel(writer,sheet_name="DiffAmtOnlyBalancingFileAvail",i
Calculate diff amount records when only data file avaiable
calculateAmountDiffWhenOnlyDataFileAvl(data_df).to_excel(writer,sheet_name="DiffAmtOnlyDataFileAvail",index=False
# calcaulateAmountByIdDivLoc(data_df)
calcaulateAmountByIdDivLoc(data_df).to_excel(writer,sheet_name="AmountStatsbyIdDivLoc",index=True)
OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash - src
1249@VD3MSTSICP1445 MINGW64 ~/OneDrive - Macy's, Inc/Desktop/Nishant/Python/Balancing/src
Ln 17, Col 15 Spaces: 4 UTF-8 CRLF {} Python 3.11.9
e here to search


142 # checkMultipleMediaDateInDataFile(data_df).to_excel(writer,
143
144 # Calculate department count for ID, Div,Loc,Index,Publisher dat
145 calculateUniqueByIdDivLocIndexPublisherDate(data_df).to_excel(wr
146
147 # def calculateAmountDiff
148 # ---------- Execution started ----------
149 data_df = getbalanceSummaryDataFrame()
150
151 # Add complete dataframe to excel
152 generateFile(data_df)
153
154 # Run the query you want and get file created with the results
155 #getReceiverDetails()
156
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
BH21249@VD3MSISICP1445 MINGW64 ~/OneDrive - Macy's, Inc/Desktop/Nishant/Pytho

Edit Selection View ··· ← Python

reference-copy.py × BalancingSummaryDataAnalysis.py

Balancing > src > reference-copy.py > ...

```
def calculateHeaderTransCode(record):
    transType=record['trans_type']
    #print(transType)
    filtered_df=rpsOsrTranseTypeMapping_df[rpsOsrTranseTypeMapping_df['RPS TYPE']==transType]
    #print(filtered_df)
    if(filtered_df.empty):
        return 'NO_MATCH_FOUND'
    #print(filtered_df[['OSR Mapping']])
    transCode=filtered_df['OSR Mapping-description'].values[0]

    return transCode

def calculateDetailTransCode(record):
    txn_amount=record['amount']
    if(txn_amount>0):
        return 'SALE'
    elif(txn_amount<0):
        return 'RTN'
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

BH21249@VD3MSTSICP1445 MINGW64 ~/OneDrive - Macy's, Inc/Desktop/Nishant/Python/Balancing/src
$

0 0 0 Ln 15, Col 1 Spaces: 4 UTF-8

File Edit Selection View ··· ← Python

reference-copy.py × BalancingSummaryDataAnalysis.py

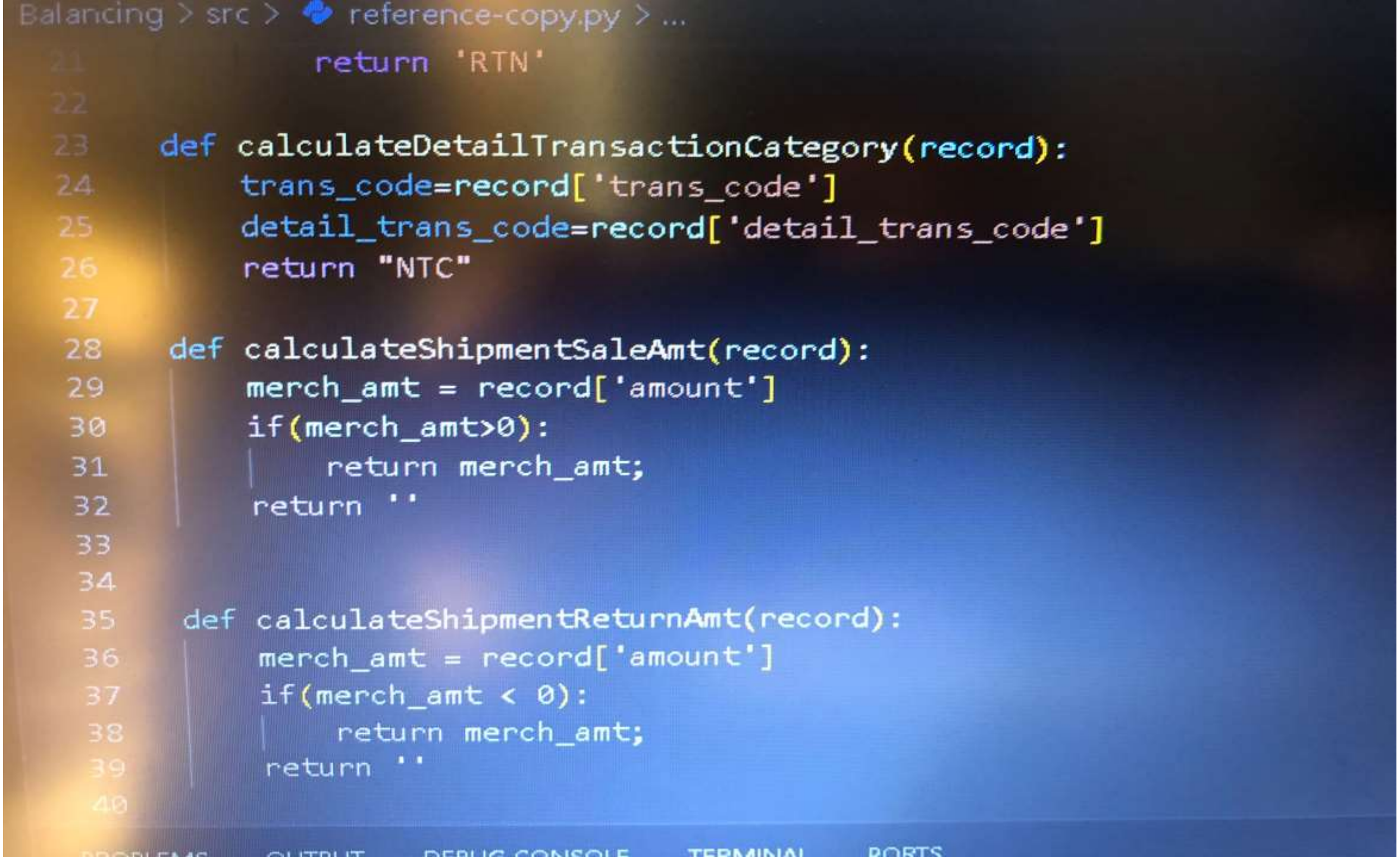
Balancing > src > reference-copy.py > ...

```python
def calculateHeaderTransCode(record):
    transType=record['trans_type']
    #print(transType)
    filtered_df=rpsOsrTranseTypeMapping_df[rpsOsrTranseTypeMapping_df['RPS TYPE']==transType]
    #print(filtered_df)
    if(filtered_df.empty):
        return 'NO_MATCH_FOUND'
    #print(filtered_df[['OSR Mapping']])
    transCode=filtered_df['OSR Mapping-description'].values[0]

    return transCode

def calculateDetailTransCode(record):
    txn_amount=record['amount']
    if(txn_amount>0):
        return 'SALE'
    elif(txn_amount<0):
        return 'RTN'
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

BH21249@VD3MSTSICP1445 MINGW64 ~/OneDrive - Macy's, Inc/Desktop/Nishant/Python/Balancing/src
$

0 0 0 Ln 15, Col 1 Spaces: 4 UTF-8 CR


Balancing > src > reference-copy.py > ...
return 'RTN'
def calculateDetailTransactionCategory(record):
trans_code=record['trans_code']
detail_trans_code=record['detail_trans_code']
return "NTC"
def calculateShipmentSaleAmt(record):
merch_amt = record['amount']
if(merch_amt>0):
return merch_amt;
return ''
def calculateShipmentReturnAmt(record):
merch_amt = record['amount']
if(merch_amt < 0):
return merch_amt;
return ''