**Overview**

Adviser Direct is working on a capability called Datafeeds where Adviser, Client and Account data is fetched, validated, and eventually sent via SFTP to third party platforms such as Xplan. With Xplan, after receiving the data, they will perform their own quality checks and send Vanguard a daily email confirming the receipt of data and any issues encountered. We need to have a proactive approach where we need to ensure that the data we are sending to Xplan has been properly validated before it gets sent.

**Problem statement**

One of the common problems encountered with a data feed is related to transactions and portfolio balances. There were cases where the total value of the transactions for a particular account does not equate to his/her portfolio balance.

Create a script where it will take 2 CSVs, one for a list of accounts and their holding balances and another for a list of transactions. Ensure that the total value of the transactions per holding is equal to their holding balance. If not, produce a report stating which holding for a particular account is not correct.

Please create a personal Bitbucket/GitHub repository to store the code. Ensure that the code is runnable and executable before sending it.

Name = datafeeds-validation

1. Understand
   1. Script in repo
   2. 2 csv files as input
   3. Run check on the transactions vs holdings at an account and product level
   4. Produce report for accounts and their product holdings to show any that don’t reconcile
2. Approach
   1. Read the csvs
   2. Transform the transactions (abs, rounding)
   3. Merge into holdings and isolate the accounts/products that do not equal each other
   4. Produce a csv report and one bar chart showing the results
3. Psuedo
   1. data reader – tries to import the raw data files (more of a central tool)
   2. data transform – totals the deposits and withdrawals and adds a netcashflow column, saves csv to a transformed folder
   3. compare financials – merges the input dataframes and creates a valid\_balance column, saves the script\_output to an output folder
   4. run\_report – uses the script\_output to generate an exceptions report (account, product, balance, cashflow, difference) and waterfall chart showing the total number of unique accounts, total number of holdings, valid holdings, invalid holdings
4. Considerations

**Sample dataset:**

**Account CSV**

account\_id,product\_id,total\_product\_balance   
1,101,5000   
1,102,3000   
2,103,7000   
3,101,2000   
3,104,3000

**Transactions CSV**

account\_id,transaction\_id,product\_id,transaction\_type,amount  
1,1,101,deposit,2000  
1,2,101,withdrawal,1000  
1,3,102,deposit,3000  
1,4,102,withdrawal,2000  
2,5,103,deposit,5000  
2,6,103,withdrawal,2000  
2,7,103,deposit,3000  
2,8,103,withdrawal,1000  
3,9,101,deposit,1500  
3,10,101,withdrawal,500  
3,11,104,deposit,2000  
3,12,104,withdrawal,1000  
3,13,104,deposit,1000  
3,14,104,withdrawal,500  
3,15,104,deposit,2500  
3,16,104,withdrawal,2000  
3,17,104,deposit,1500  
3,18,104,withdrawal,1000  
3,19,104,deposit,1800  
3,20,104,withdrawal,1500