



DATA DICTIONARY

FVT/GE FINANCIAL AID DATA MERGE (PYTHON)

Revisions

Change reference	Date	Version
Initial documentation	06/20/2025	1.0

Contacts

Data Services

dataservices@sbctc.edu

Table of Contents

Revisions.....	2
Contacts	2
Table of Contents	3
Financial Value Transparency/Gainful Employment (FVT/GE) (Python).....	4

Financial Value Transparency/Gainful Employment (FVT/GE) (Python)

```
{
  "cells": [
    {
      "cell_type": "markdown",
      "id": "5e23cf1c-d6d2-4fdc-9a3f-038e2b6ae216",
      "metadata": {},
      "source": [
        "# **Student Level Report Process**\n",
        "\n",
        "***Objective:**\n",
        "This script combines data from the GE FVT Query and the National Student Clearinghouse (NSC) Student-  
Level Report to generate a finalized report for analysis.\n",
        "\n",
        "## **Outline**\n",
        "\n",
        "1. [Run queries](#1.-Run-Queries) \n",
        "2. [Download NCH Student Level Report](#2.-Download-NCH-Student-Level-Report) \n",
        "3. [Import Packages](#3.-Import-Packages) \n",
        "4. [Import Data](#4.-Import-Data) \n",
        "5. [Data Cleaning](#5.-Data-Cleaning) \n",
        "6. [Merge Data](#6.-Merge-Data) \n",
        "7. [Finalize Report](#7.-Finalize-Report)\n",
        "\n",
        "\n",
        "Prepared by Anna Fenner (Anna.Fenner@lwtech.edu) \n",
        "Adapted from R code by Kelsey Anderson (kanderson@highline.edu)"
      ]
    },
    {
      "cell_type": "markdown",
      "id": "1ab8441c-94af-4cc2-ac1f-ac24bd0d23e0",
      "metadata": {},
      "source": [
        "## **1. Run Queries**"
      ]
    },
    {
      "cell_type": "markdown",
      "id": "f24e3d54-b997-4d04-ab0d-f67113f1427a",
      "metadata": {},
      "source": [
```

```

"Run the following three queries for **both**:\n",
"\n",
"* **All students** and\n",
"* **Award year only**\n",
"\n",
"> ⚠️ **Important:** Keep the exported data in its original Excel format. Data cleaning will be handled
later in **Step 5**.\n",
"\n",
"### **Required Queries**\n",
"\n",
"* `QCS_FA_GE_FVT_GRANTS` - *Financial Aid Disbursement Data*\n",
"* `QCS_FA_GE_FVT_COA` - *Cost of Attendance Data*\n",
"* `QCS_FA_GE_FVT_DEBT` - *Institutional Debt, Tuition, and Fees*"
]
},
{
"cell_type": "markdown",
"id": "f058adad-1a0e-4c45-8fa0-990afd41592d",
"metadata": {},
"source": [
"## **2. Download NCH Student Level Report**"
]
},
{
"cell_type": "markdown",
"id": "a3940fc7-5633-43c3-ba58-0503b0b671ab",
"metadata": {},
"source": [
"## **3. Import Packages**"
]
},
{
"cell_type": "code",
"execution_count": 1,
"id": "7207b7fa-e8ae-4416-982d-05fc1416ff50",
"metadata": {},
"outputs": [],
"source": [
"import pandas as pd\n",
"import numpy as np"
]
},
{
"cell_type": "markdown",
"id": "e60f94a3-e988-417e-9177-d3a30bb3c411",
"metadata": {},

```

```

"source": [
  "## **4. Import Data**"
]
},
{
  "cell_type": "code",
  "execution_count": 4,
  "id": "5c930fee-de9f-4fb1-945f-b4f1cbf47c90",
  "metadata": {},
  "outputs": [],
  "source": [
    "# Enter Academic Year (YYYYYYYY format):\n",
    "#Update this every year\n",
    "ayr = 20232024\n",
    "\n",
    "# Enter the start date of the first term for which your campus has\n",
    "# ctcLink FinAid data (YYYYMMDD format):\n",
    "convert_date = 20210923"
  ]
},
{
  "cell_type": "code",
  "execution_count": 6,
  "id": "c0599b27-63e3-48fe-9d30-fb3c19894819",
  "metadata": {},
  "outputs": [],
  "source": [
    "# Financial Aid Disbursement Data (all)\n",
    "grants_all = pd.read_excel('../Documents/src/GRANTS_UPDATE_2024_ALL.xlsx')
  ]
},
{
  "cell_type": "code",
  "execution_count": 8,
  "id": "080c7f3f-06b8-45df-ac28-255c7a94d1fc",
  "metadata": {},
  "outputs": [],
  "source": [
    "# Financial Aid Disbursement Data (award year only)\n",
    "# data = pd.read_excel('../folder/file')\n",
    "# update this to match your file pathway\n",
    "grants_award = pd.read_excel('../Documents/src/GRANTS_UPDATE_2024_AWARD.xlsx')
  ]
},
{
  "cell_type": "code",

```

```

"execution_count": 10,
"id": "db30b0d7-e2ef-4691-abd4-d500dc0736c3",
"metadata": {},
"outputs": [],
"source": [
    "# Cost of Attendance Data (all)\n",
    "coa_all = pd.read_excel('../Documents/src/COA_UPDATE_2024_ALL.xlsx')"
]
},
{
    "cell_type": "code",
    "execution_count": 12,
    "id": "c9b08c76-61f9-43a2-99ed-5887d6d5ca90",
    "metadata": {},
    "outputs": [],
    "source": [
        "# Cost of Attendance Data (award year only)\n",
        "coa_award = pd.read_excel('../Documents/src/COA_UPDATE_2024_AWARD.xlsx')"
    ]
},
{
    "cell_type": "code",
    "execution_count": 14,
    "id": "a3838003-8e46-4fe1-8da0-042c2cf15206",
    "metadata": {},
    "outputs": [],
    "source": [
        "# Institutional Debt, and Tuition and Fees (all)\n",
        "debt_all = pd.read_excel('../Documents/src/DEBT_UPDATE_2024_ALL.xlsx')"
    ]
},
{
    "cell_type": "code",
    "execution_count": 16,
    "id": "306ecba4-c08f-475e-903d-ece67858ba02",
    "metadata": {},
    "outputs": [],
    "source": [
        "# Institutional Debt, and Tuition and Fees (award year only)\n",
        "debt_award = pd.read_excel('../Documents/src/DEBT_UPDATE_2024_AWARD.xlsx')"
    ]
},
{
    "cell_type": "code",
    "execution_count": 18,
    "id": "203471b8-b206-41ba-99c3-d72d3bdee83a",

```

```

"metadata": {},
"outputs": [],
"source": [
    "# NCH Data\n",
    "nch_data = pd.read_csv('./src/NCH_23_24.csv')\n",
    ]
},
{
    "cell_type": "code",
    "execution_count": 74,
    "id": "178e307b-3dc2-42b2-a575-e45fa43985fc",
    "metadata": {},
    "outputs": [],
    "source": [
        "#print length of data so to check length at the end to make sure it matches\n",
        "\n",
        "#print(\\"Length of NCH dataset: \\" +str(len(nch_data)))\n",
    ]
},
{
    "cell_type": "markdown",
    "id": "28b5906a-7858-48cd-bbf4-c7d90fb7fb89",
    "metadata": {},
    "source": [
        "## **5. Data Cleaning**\n",
    ]
},
{
    "cell_type": "markdown",
    "id": "975f9092-54e7-48bd-a974-7f3a3cfce805",
    "metadata": {},
    "source": [
        "### 5a. Cleaning Function\n",
    ]
},
{
    "cell_type": "code",
    "execution_count": 24,
    "id": "c488ba08-7856-4c24-89e9-4f3a4b6aea2d",
    "metadata": {},
    "outputs": [],
    "source": [
        "def clean_data(df, column_name='CIP Code', award_year=ayr):\n",
        "    \\\n",
        "    Cleans and preprocesses the query data:\n",
        "    - Sets the second row as headers\n",
    ]

```

```

" - Drops the first row\n",
" - Resets the index\n",
" - Drops rows with null values in the specified column\n",
" - Removes decimal from the specified column and converts to int\n",
" - Adds an 'Award Year' column\n",
" - Converts 'NID' to numeric, fills nulls with -1, and converts to int\n",
" - Extracts 'AA' or 'TA' from 'AA or TA' column into a new column 'Record Type'\n",
"\n",
" Parameters:\n",
" df (pd.DataFrame): The DataFrame to process.\n",
" column_name (str): The column to clean decimals from.\n",
" award_year (int): The award year to assign to the new column.\n",
"\n",
" Returns:\n",
" pd.DataFrame: The cleaned DataFrame.\n",
" \n\n",
" # Set second row as header\n",
" df.columns = df.iloc[0]\n",
" df = df.drop(0).reset_index(drop=True)\n",
"\n",
" # Drop rows with null in the specified column\n",
" df = df.dropna(subset=[column_name])\n",
"\n",
" # Remove decimal and convert to int\n",
" if column_name in df.columns:\n",
"     df[column_name] = df[column_name].astype(str).str.replace('.', "", regex=False)\n",
"     df[column_name] = df[column_name].astype(int)\n",
" else:\n",
"     print(f"Column '{column_name}' not found in the DataFrame.")\n",
"\n",
" # Add Award Year column\n",
" df['Award Year'] = award_year\n",
"\n",
" # Clean NID column\n",
" if 'NID' in df.columns:\n",
"     df['NID'] = pd.to_numeric(df['NID'], errors='coerce').fillna(-1).astype(int)\n",
" else:\n",
"     print("Column 'NID' not found in the DataFrame.")\n",
"\n",
" # Create 'Record Type' column from 'AA or TA'\n",
" if 'AA or TA' in df.columns:\n",
"     df['Record Type'] = df['AA or TA'].astype(str).str[:2]\n",
" else:\n",
"     print("Column 'AA or TA' not found in the DataFrame.")\n",
"\n",
" return df\n"

```

```

]
},
{
  "cell_type": "markdown",
  "id": "809b982d-16b9-4212-bc55-9605d679a110",
  "metadata": {},
  "source": [
    "### 5b. Cleaning Financial Aid Disbursement Data "
  ]
},
{
  "cell_type": "code",
  "execution_count": 27,
  "id": "87514076-3386-4136-9ff9-2bc2d2e20789",
  "metadata": {},
  "outputs": [],
  "source": [
    "grants_all = clean_data(grants_all)\n",
    "grants_award = clean_data(grants_award)"
  ]
},
{
  "cell_type": "markdown",
  "id": "7116c433-d0c2-4ff7-9872-848b9843d809",
  "metadata": {},
  "source": [
    "### 5c. Cleaning Cost of Attendance Data"
  ]
},
{
  "cell_type": "code",
  "execution_count": 30,
  "id": "3344d3b8-b7a7-4725-aa18-f1dcb4dbc6fd",
  "metadata": {},
  "outputs": [],
  "source": [
    "coa_all = clean_data(coa_all)\n",
    "coa_award = clean_data(coa_award)"
  ]
},
{
  "cell_type": "markdown",
  "id": "1fcd4fc5-c77c-4446-9159-48568d68d07d",
  "metadata": {},
  "source": [
    "### 5d. Cleaning Institutional Debt, and Tuition and Fees"
  ]
}

```

```

]
},
{
  "cell_type": "code",
  "execution_count": 33,
  "id": "bab26d46-3d38-4109-9d05-1231244b667a",
  "metadata": {},
  "outputs": [],
  "source": [
    "debt_all = clean_data(debt_all)\n",
    "debt_award = clean_data(debt_award)"
  ]
},
{
  "cell_type": "markdown",
  "id": "6665f39c-1347-4699-9012-2018c9ce3bb0",
  "metadata": {},
  "source": [
    "### 5e. Combine query data into 3 data sources. "
  ]
},
{
  "cell_type": "code",
  "execution_count": 36,
  "id": "bcc35b6b-d2ec-4aaf-89d2-ce33aa3c1cc2",
  "metadata": {},
  "outputs": [],
  "source": [
    "grants_data = pd.concat([grants_all, grants_award], axis=0, ignore_index=True)\n",
    "coa_data = pd.concat([coa_all, coa_award], axis=0, ignore_index=True)\n",
    "debt_data = pd.concat([debt_all, debt_award], axis=0, ignore_index=True)"
  ]
},
{
  "cell_type": "markdown",
  "id": "c466f3b3-3741-49b9-b933-dafd87199663",
  "metadata": {},
  "source": [
    "Checking the length of each data source before and after combining to make sure numbers are correct. "
  ]
},
{
  "cell_type": "code",
  "execution_count": 78,
  "id": "044a6543-4ee3-4fb6-a36c-335275eebd60",
  "metadata": {},

```

```

"outputs": [],
"source": [
    "#print(len(coa_all), len(coa_award))\n",
    "#print(len(debt_all), len(debt_award))\n",
    "#print(len(grants_all), len(grants_award))\n",
    "\n",
    "#printing length of each combined data source to ensure added correctly\n",
    "#print(len(grants_data), len(coa_data), len(debt_data))"
]
},
{
    "cell_type": "markdown",
    "id": "5a3c4022-3394-4a78-b1f1-a86e22836fa8",
    "metadata": {},
    "source": [
        "### 5f. Changing data types to be like NCH"
    ]
},
{
    "cell_type": "code",
    "execution_count": 42,
    "id": "e8bdb2e6-fbf1-4f1d-a01a-2d27103d9296",
    "metadata": {},
    "outputs": [],
    "source": [
        "# List of columns to convert\n",
        "coa_to_convert = [\n",
        "    'COA Books, Supplies, Equipment',\n",
        "    'COA Food and Housing',\n",
        "    'COA Total'\n",
        "]\n",
        "\n",
        "debt_to_convert = [\n",
        "    'Total Institutional Debt',\n",
        "    'Total Tuition and Fees'\n",
        "]\n",
        "\n",
        "grants_to_convert = [\n",
        "    'Total Grants and Scholarships',\n",
        "    'Institutional Grants and Schol',\n",
        "    'Other, State or Private Grants',\n",
        "    'Private Loans'\n",
        "]\n",
        "\n",
        "# Convert each column to float

```

```

"for col in coa_to_convert:\n",
"  coa_data[col] = coa_data[col].astype(float)\n",
"\n",
"for col in debt_to_convert:\n",
"  debt_data[col] = debt_data[col].astype(float)\n",
"\n",
"for col in grants_to_convert:\n",
"  grants_data[col] = grants_data[col].astype(float)\n"
]
},
{
"cell_type": "markdown",
"id": "6e546ebd-c726-450d-ae31-ac0167da147e",
"metadata": {},
"source": [
"## **6. Merge Data**"
]
},
{
"cell_type": "code",
"execution_count": 80,
"id": "2440728f-5d41-4c72-b669-759de13daacc",
"metadata": {},
"outputs": [],
"source": [
"# Perform the first left join with coa_data\n",
"#NCH Data\n",
"merged_data = pd.merge(\n",
"  nch_data, \n",
"  coa_data,\n",
"  how='left', \n",
"  left_on=['Student Social Security Number', 'CIP Code', 'Record Type', 'Award Year'],\n",
"  right_on=['NID', 'CIP Code', 'Record Type', 'Award Year']\n",
")\n",
"\n",
"# Perform the second left join with debt_data\n",
"merged_data = pd.merge(\n",
"  merged_data, # Merge into the already merged DataFrame\n",
"  debt_data,\n",
"  how='left', \n",
"  left_on=['Student Social Security Number', 'CIP Code', 'Record Type', 'Award Year'],\n",
"  right_on=['NID', 'CIP Code', 'Record Type', 'Award Year']\n",
")\n",
"\n",
"# Perform the third left join with grants_data\n",
"merged_data = pd.merge(\n",

```

```

" merged_data, # Merge into the already merged DataFrame\n",
" grants_data, \n",
" how='left', \n",
" left_on=['Student Social Security Number', 'CIP Code', 'Record Type', 'Award Year'],\n",
" right_on=['NID', 'CIP Code', 'Record Type', 'Award Year']\n",
")\n",
"\n",
"\n",
"# The merged_data DataFrame now contains all the columns from the three data sources\n",
"#len(merged_data)"
]
},
{
"cell_type": "markdown",
"id": "aef5d35e-33ac-4eec-8990-314336cdd09d",
"metadata": {},
"source": [
"COA:\n",
"['COA Books, Supplies, Equipment', ['COA Food and Housing'], ['COA Total']\n",
"Debt: \n",
"['Total Institutional Debt'], ['Total Tuition and Fees']\n",
"Grants:\n",
"['Total Grants and Scholarships'], ['Total Tuition and Fees'], ['Institutional Grants and Schol'], ['Other, State or Private Grants'], ['Private Loans']"
]
},
{
"cell_type": "markdown",
"id": "dcea2d56-d35f-48a7-9893-5c612bae673f",
"metadata": {},
"source": [
"## **7. Finalize Report**"
]
},
{
"cell_type": "code",
"execution_count": 49,
"id": "ca81d47e-fa30-4ce2-aa8d-fb12f455773d",
"metadata": {},
"outputs": [],
"source": [
"#sort values by record type\n",
"merged_data = merged_data.sort_values(by='Record Type')"
]
},
{

```

```

"cell_type": "code",
"execution_count": 51,
"id": "45d12546-9026-40c9-8aa5-915fe4fed8f3",
"metadata": {},
"outputs": [],
"source": [
"#add residency data to NCH column\n",
"def determine_residency_status(row):\n",
"    if row['Title IV Aid?'] == 'N' or pd.isna(row['Title IV Aid?']):\n",
"        return np.nan\n",
"    elif row['Residency'] == 'IS':\n",
"        return 'IS'\n",
"    else:\n",
"        return 'OS'\n",
"\n",
"merged_data['Residency Tuition Status by State or District'] =
merged_data.apply(determine_residency_status, axis=1)\n"
],
{
"cell_type": "markdown",
"id": "63186330-f276-446c-9e65-b6dfa06f3781",
"metadata": {},
"source": [
"***Need to copy data from the columns from the query data to NCH columns***"
]
},
{
"cell_type": "markdown",
"id": "00efd44a-e92e-4fda-ba43-b63e6d9ce5ac",
"metadata": {},
"source": [
"### **TA Column Mapping Notes**\n",
"\n",
"Need to copy the query column values and put them into the NSC Columns. \n",
"\n",
"| **NSC Column Names** | **Query Column Names** |\n",
"| ----- | ----- |\n",
"| *Total Amount Student Received in Private Education Loans During Student's Entire Enrollment in the Program* | `Private Loans` |\n",
"| *Total Amount of Institutional Debt During Student's Entire Enrollment in the Program* | `Total Institutional Debt` |\n",
"| *Total Amount of Tuition & Fees Assessed During Student's Entire Enrollment in the Program* | `Total Tuition and Fees` |\n",
"| *Total Amount of Allowance for books, supplies, and equipment included in the student's Title IV, HEA"
]
}

```

```

COA*      | `COA Books, Supplies, Equipment` |\n",
    "| *Total Amount of Grants and Scholarships the student received During Student's Entire Enrollment in the
Program* | `Total Grants and Scholarships` |\n"
]
},
{
    "cell_type": "code",
    "execution_count": 55,
    "id": "a73a527a-1bf1-48ca-bf4e-ab71e02d5a48",
    "metadata": {},
    "outputs": [],
    "source": [
        "# Define a mask for rows where Record Type is 'TA' and Title IV Aid? is 'Y'\n",
        "mask = (merged_data['Record Type'] == 'TA') & (merged_data['Title IV Aid?'] == 'Y')\n",
        "\n",
        "# Private Loans\n",
        "merged_data.loc[mask, \n",
        "    \"Total Amount Student Received in Private Education Loans During Student's Entire Enrollment in the
Program\"\n",
        "] = merged_data.loc[mask, 'Private Loans']\n",
        "\n",
        "# Institutional Debt\n",
        "merged_data.loc[mask & (merged_data['Total Institutional Debt'] > 0),\n",
        "    \"Total Amount of Institutional Debt During Student's Entire Enrollment in the Program\"\n",
        "] = merged_data.loc[mask & (merged_data['Total Institutional Debt'] > 0), 'Total Institutional Debt']\n",
        "\n",
        "merged_data.loc[mask & (merged_data['Total Institutional Debt'] <= 0),\n",
        "    \"Total Amount of Institutional Debt During Student's Entire Enrollment in the Program\"\n",
        "] = 0.0\n",
        "\n",
        "# Tuition & Fees\n",
        "merged_data.loc[mask, \n",
        "    \"Total Amount of Tuition & Fees Assessed During Student's Entire Enrollment in the Program\"\n",
        "] = merged_data.loc[mask, 'Total Tuition and Fees']\n",
        "\n",
        "# COA Books, Supplies, Equipment\n",
        "merged_data.loc[mask, \n",
        "    \"Total Amount of Allowance for books, supplies, and equipment included in the student's title IV, HEA
COA During Student's Entire Enrollment in the Program\"\n",
        "] = merged_data.loc[mask, 'COA Books, Supplies, Equipment']\n",
        "\n",
        "# Grants and Scholarships\n",
        "merged_data.loc[mask, \n",
        "    \"Total Amount of Grants and Scholarships the student received During Student's Entire Enrollment in the
Program\"\n",
        "] = merged_data.loc[mask, 'Total Grants and Scholarships']\n"

```

```

]
},
{
  "cell_type": "markdown",
  "id": "c8eabd3f-779a-4a3f-8b86-e99a0697860d",
  "metadata": {},
  "source": [
    "### **AA Column Mapping Notes**\n",
    "\n",
    "Need to copy the query column values and put them into the NSC Columns. \n",
    "\n",
    "| **NSC Column** | **Source in `merged_data`** | \n",
    "| ----- | ----- | \n",
    "| *Annual Cost of Attendance (COA)* | `COA Total` | \n",
    "| *Tuition and Fees Amount for Award Year being Reported* | `Total Tuition and Fees` | \n",
    "| *Allowance for Books, Supplies, and Equipment* | `COA Books, Supplies, Equipment` | \n",
    "| *Allowance for Housing and Food* | `COA Food and Housing` | \n",
    "| *Institutional Grants and Scholarships* | `Institutional Grants and Schol` | \n",
    "| *Other State, Tribal, or Private Grants* | `Other, State or Private Grants` | \n",
    "| *Private Loans Amount* | `Private Loans` | "
  ]
},
{
  "cell_type": "code",
  "execution_count": 58,
  "id": "45a68de5-b0c2-4f94-883a-3f1977e0e5b9",
  "metadata": {},
  "outputs": [],
  "source": [
    "# Define a mask for rows where Record Type is 'AA' and Title IV Aid? is 'Y'\n",
    "mask_aa = (merged_data['Record Type'] == 'AA') & (merged_data['Title IV Aid?'] == 'Y')\n",
    "\n",
    "# Private Loans Amount\n",
    "merged_data.loc[mask_aa, 'Private Loans Amount'] = merged_data.loc[mask_aa, 'Private Loans']\n",
    "\n",
    "# Allowance for Books, Supplies, and Equipment\n",
    "merged_data.loc[mask_aa, 'Allowance for Books, Supplies, and Equipment'] = merged_data.loc[mask_aa, 'COA Books, Supplies, Equipment']\n",
    "\n",
    "# Allowance for Housing and Food\n",
    "merged_data.loc[mask_aa, 'Allowance for Housing and Food'] = merged_data.loc[mask_aa, 'COA Food and Housing']\n",
    "\n",
    "# Annual Cost of Attendance (COA)\n",
    "merged_data.loc[mask_aa, 'Annual Cost of Attendance (COA)'] = merged_data.loc[mask_aa, 'COA Total']\n",
    "\n"
  ]
}

```

```

"# Tuition and Fees Amount for Award Year being Reported\n",
"merged_data.loc[mask_aa, 'Tuition and Fees Amount for Award Year being Reported'] =
merged_data.loc[mask_aa, 'Total Tuition and Fees']\n",
"\n",
"# Institutional Grants and Scholarships\n",
"merged_data.loc[mask_aa, 'Institutional Grants and Scholarships'] = merged_data.loc[mask_aa,
'Institutional Grants and Schol']\n",
"\n",
"# Other State, Tribal, or Private Grants\n",
"merged_data.loc[mask_aa, 'Other State, Tribal, or Private Grants'] = merged_data.loc[mask_aa, 'Other,
State or Private Grants']\n"
]
},
{
"cell_type": "code",
"execution_count": 60,
"id": "ceeab66e-0943-4196-8cb5-19021ee0086e",
"metadata": {},
"outputs": [],
"source": [
"# Update Invalid Flag\n",
"merged_data['Invalid Flag'] = merged_data['Title IV Aid?'].apply(\n",
"    lambda x: ' ' if x == 'Y' else 'T'\n",
")"
]
},
{
"cell_type": "code",
"execution_count": 62,
"id": "6c782315-8a3b-4765-ba07-9bf5a510ee13",
"metadata": {},
"outputs": [],
"source": [
"# Filter rows where 'Program Enrollment Begin Date' is before convert_date\n",
"check_legacy = merged_data[merged_data['Program Enrollment Begin Date'] < convert_date].copy()\n",
"\n",
"# Define the action based on 'Record Type'\n",
"check_legacy['Action'] = np.where(\n",
"    check_legacy['Record Type'] == 'AA',\n",
"    'Check legacy for Aid, use annual values from queries, remove invalid flag, add residency',\n",
"    'Check legacy for Aid, sum legacy values with totals from queries, remove invalid flag, add residency',\n",
")"
]
},
{
"cell_type": "code",

```

```

"execution_count": 64,
"id": "be51f122-ecaa-4084-8c7b-50024175158a",
"metadata": {},
"outputs": [],
"source": [
    "#Removing the columns from the query data\n",
    "common_columns = [col for col in nch_data.columns if col in merged_data.columns]\n",
    "merged_output = merged_data[common_columns]"
]
},
{
    "cell_type": "code",
    "execution_count": 76,
    "id": "ddac6a8b-a724-4335-b8d4-a7ff6081cf2f",
    "metadata": {},
    "outputs": [],
    "source": [
        "#check the length to ensure the same length before merging NCH with the query data\n",
        "#len(merged_data)"
    ]
},
{
    "cell_type": "code",
    "execution_count": 68,
    "id": "b65d8263-34f2-47c7-b517-f3367a40c07a",
    "metadata": {},
    "outputs": [],
    "source": [
        "#save data as an excel file\n",
        "merged_output.to_excel(\"../Documents/src/NCH_Data_UPDATE.xlsx\", index=False)"
    ]
},
{
    "cell_type": "code",
    "execution_count": 70,
    "id": "62b0f902-098d-4d3b-bafa-31ff9a2c7771",
    "metadata": {},
    "outputs": [],
    "source": [
        "#data to check in legacy\n",
        "check_legacy.to_excel(\"../Documents/src/Check_Legacy_data.xlsx\", index=False)"
    ]
}
],
"metadata": {
    "kernelspec": {

```

```
"display_name": "Python 3 (ipykernel)",
"language": "python",
"name": "python3"
},
"language_info": {
  "codemirror_mode": {
    "name": "ipython",
    "version": 3
  },
  "file_extension": ".py",
  "mimetype": "text/x-python",
  "name": "python",
  "nbconvert_exporter": "python",
  "pygments_lexer": "ipython3",
  "version": "3.12.2"
}
},
"nbformat": 4,
"nbformat_minor": 5
}
```



Content is licensed under a Creative Commons Attribution 4.0 International License, unless noted otherwise.

Washington State Board for Community and Technical Colleges