Term Project: Is AI taking our jobs or transforming them?

Lana Geissinger
Bellevue University

DSC540_T303 Data Preparation (2257-1)

Professor Catherine Williams

Milestone 2

June 29, 2025

Cleaning/Formatting Flat File Source

```python
import os
import pandas as pd
from dotenv import load_dotenv
```

Load and preview data files with SOC and NAICS codes

```
# Load environment variables
load_dotenv('../env_var.env')
NAICS_codes_path = os.getenv('NAICS_codes_path')
SOC_codes_path = os.getenv('SOC_codes_path')

# Preview data
if NAICS_codes_path and SOC_codes_path:
    try:
        df_NAICS = pd.read_csv(NAICS_codes_path, encoding='Windows-1252')
        df_SOC = pd.read_csv(SOC_codes_path, encoding='Windows-1252')
        print("DataFrame for NAICS Data:")
        print(df_NAICS.head(20))
        print(df_NAICS.info())
        print("DataFrame for SOC Data:")
        print(df_SOC.head(20))
        print(df_SOC.info())
    except FileNotFoundError as e:
        print(f"Error: {e}")
    except Exception as e:
        print(f"An unexpected error occurred: {e}")
else:
    print("Error: One or both environment variables for file paths are not set or invali
d.")
```

```
DataFrame for NAICS Data:
   Sector  \
0     NaN
1      11
2      21
3      22
4      23
5   31-33
6      42
7   44-45
8   48-49
9      51
10     52
11     53
12     54
13     55
14     56
15     61
16     62
17     71
18     72
19     81

                                                                          Name  
\
0                                                                          NaN
1                                   Agriculture, Forestry, Fishing and Hunting
2                                Mining, Quarrying, and Oil and Gas Extraction
3                                                                    Utilities
4                                                                 Construction
5                                                                Manufacturing
6                                                              Wholesale Trade
7                                                                 Retail Trade
8                                               Transportation and Warehousing
9                                                                  Information
10                                                       Finance and Insurance
11                                           Real Estate and Rental and Leasing
12                            Professional, Scientific, and Technical Services
13                                     Management of Companies and Enterprises
14  Administrative and Support and Waste Management and Remediation Services
15                                                        Educational Services
16                                           Health Care and Social Assistance
17                                          Arts, Entertainment, and Recreation
18                                             Accommodation and Food Services
19                               Other Services (except Public Administration)

    Subsectors (3-digit)  Industry Groups (4-digit)  \
0                    NaN                        NaN
1                    5.0                       19.0
2                    3.0                        5.0
3                    1.0                        3.0
4                    3.0                       10.0
5                   21.0                       86.0
6                    3.0                       19.0
7                    9.0                       24.0
8                   11.0                       29.0
9                    6.0                       11.0
```

```
10                 5.0                      11.0
11                 3.0                       8.0
12                 1.0                       9.0
13                 1.0                       1.0
14                 2.0                      11.0
15                 1.0                       7.0
16                 4.0                      18.0
17                 3.0                       9.0
18                 2.0                       6.0
19                 4.0                      14.0

    NAICS Industries (5-digit) 6-digit Industries       Unnamed: 6 Unnamed: 7
0                          NaN        U.S. Detail  Same as 5-digit      Total
1                         42.0                 32               32         64
2                         11.0                 14                7         21
3                          6.0                 10                4         14
4                         28.0                  4               27         31
5                        176.0                249               97        346
6                         69.0                  0               69         69
7                         48.0                 16               41         57
8                         42.0                 25               32         57
9                         24.0                 10               19         29
10                        27.0                 13               22         35
11                        17.0                 11               13         24
12                        35.0                 20               29         49
13                         1.0                  3                0          3
14                        29.0                 25               19         44
15                        12.0                  7               10         17
16                        30.0                 16               23         39
17                        23.0                  3               22         25
18                        10.0                  8                7         15
19                        30.0                 24               20         44
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 22 entries, 0 to 21
Data columns (total 8 columns):
 #   Column                      Non-Null Count  Dtype  
---  ------                      --------------  -----  
 0   Sector                      20 non-null     object 
 1   Name                        21 non-null     object 
 2   Subsectors (3-digit)        21 non-null     float64
 3   Industry Groups (4-digit)   21 non-null     float64
 4   NAICS Industries (5-digit)  21 non-null     float64
 5   6-digit Industries          22 non-null     object 
 6   Unnamed: 6                  22 non-null     object 
 7   Unnamed: 7                  22 non-null     object 
dtypes: float64(3), object(5)
memory usage: 1.5+ KB
None
DataFrame for SOC Data:

U.S. Bureau of Labor Statistics  \
0   On behalf of the Office of Management and Budget (OMB) and the Standard O
ccupational Classification Policy Committee (SOCPC)
1
NaN
2
November 2017 (for reference year January 2018)
```

```
3                                 ***This is the final structure for the 2018
SOC. Questions should be emailed to soc@bls.gov***
4
NaN
5
NaN
6
Major Group
7
11-0000
8
NaN
9
NaN
10
NaN
11
NaN
12
NaN
13
NaN
14
NaN
15
NaN
16
NaN
17
NaN
18
NaN
19
NaN

     Unnamed: 1   Unnamed: 2          Unnamed: 3  \
0           NaN          NaN                 NaN
1           NaN          NaN                 NaN
2           NaN          NaN                 NaN
3           NaN          NaN                 NaN
4           NaN          NaN                 NaN
5           NaN          NaN                 NaN
6   Minor Group  Broad Group  Detailed Occupation
7           NaN          NaN                 NaN
8       11-1000          NaN                 NaN
9           NaN      11-1010                 NaN
10          NaN          NaN             11-1011
11          NaN      11-1020                 NaN
12          NaN          NaN             11-1021
13          NaN      11-1030                 NaN
14          NaN          NaN             11-1031
15       Nov-00          NaN                 NaN
16          NaN      11-2010                 NaN
17          NaN          NaN              Nov-11
18          NaN      11-2020                 NaN
19          NaN          NaN              Nov-21
```

```
                                                          Unnamed: 4
0                                                                NaN
1                                                                NaN
2                                                                NaN
3                                                                NaN
4                                                                NaN
5                                                                NaN
6                                                                NaN
7                                               Management Occupations
8                                                       Top Executives
9                                                     Chief Executives
10                                                    Chief Executives
11                                     General and Operations Managers
12                                     General and Operations Managers
13                                                         Legislators
14                                                         Legislators
15  Advertising, Marketing, Promotions, Public Relations, and Sales Managers
16                                   Advertising and Promotions Managers
17                                   Advertising and Promotions Managers
18                                          Marketing and Sales Managers
19                                                  Marketing Managers
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1454 entries, 0 to 1453
Data columns (total 5 columns):
 #   Column                          Non-Null Count  Dtype 
---  ------                          --------------  ----- 
 0   U.S. Bureau of Labor Statistics  27 non-null     object
 1   Unnamed: 1                      99 non-null     object
 2   Unnamed: 2                      460 non-null    object
 3   Unnamed: 3                      868 non-null    object
 4   Unnamed: 4                      1447 non-null   object
dtypes: object(5)
memory usage: 56.9+ KB
None
```

Cleaning and Formatting SOC Data

```
# Step 1: Remove first 7 rows with metadata and whitespace
print(df_SOC.iloc[:7])
df_SOC = df_SOC.iloc[7:].copy()
df_SOC = df_SOC.apply(lambda x: x.str.strip() if x.dtype == "object" else x)
```

```
U.S. Bureau of Labor Statistics  \
0  On behalf of the Office of Management and Budget (OMB) and the Standard Oc
cupational Classification Policy Committee (SOCPC)
1
NaN
2
November 2017 (for reference year January 2018)
3                                 ***This is the final structure for the 2018
SOC. Questions should be emailed to soc@bls.gov***
4
NaN
5
NaN
6
Major Group

    Unnamed: 1   Unnamed: 2          Unnamed: 3 Unnamed: 4
0          NaN          NaN                 NaN        NaN
1          NaN          NaN                 NaN        NaN
2          NaN          NaN                 NaN        NaN
3          NaN          NaN                 NaN        NaN
4          NaN          NaN                 NaN        NaN
5          NaN          NaN                 NaN        NaN
6  Minor Group  Broad Group  Detailed Occupation        NaN
```

```python
# Step 2: Rename columns
df_SOC = df_SOC.rename(columns={
    'U.S. Bureau of Labor Statistics': 'major_group',
    'Unnamed: 1': 'minor_group',
    'Unnamed: 2': 'broad_group',
    'Unnamed: 3': 'detailed_occupation',
    'Unnamed: 4': 'occupation_title'
})

# Display the SOC Structure after renaming
print("\nSOC Structure:")
print(df_SOC.head())
```

```
SOC Structure:
   major_group minor_group broad_group detailed_occupation  \
7      11-0000         NaN         NaN                 NaN   
8          NaN     11-1000         NaN                 NaN   
9          NaN         NaN     11-1010                 NaN   
10         NaN         NaN         NaN             11-1011   
11         NaN         NaN     11-1020                 NaN   

                     occupation_title  
7              Management Occupations  
8                      Top Executives  
9                    Chief Executives  
10                   Chief Executives  
11  General and Operations Managers  
```

```python
# Step 4: Forward fill the hierarchy levels
df_SOC['major_group'] = df_SOC['major_group'].ffill()
df_SOC['minor_group'] = df_SOC['minor_group'].ffill()
df_SOC['broad_group'] = df_SOC['broad_group'].ffill()
df_SOC['detailed_occupation'] = df_SOC['detailed_occupation'].ffill()
df_SOC['occupation_title'] = df_SOC['occupation_title'].ffill()

# Display the SOC Structure after filling hierarchy levels
print("\nSOC Structure:")
print(df_SOC.head())
```

```
SOC Structure:
   major_group minor_group broad_group detailed_occupation  \
7      11-0000         NaN         NaN                 NaN   
8      11-0000     11-1000         NaN                 NaN   
9      11-0000     11-1000     11-1010                 NaN   
10     11-0000     11-1000     11-1010             11-1011   
11     11-0000     11-1000     11-1020             11-1011   

                     occupation_title  
7              Management Occupations  
8                      Top Executives  
9                    Chief Executives  
10                   Chief Executives  
11  General and Operations Managers  
```

```python
# Check for null values
print("\nNull values:")
print(df_SOC.isnull().sum())
```

```
Null values:
major_group           0
minor_group           1
broad_group           2
detailed_occupation   3
occupation_title      0
dtype: int64
```

```python
# Step 5.1: Remove rows where occupation_title is missing
df_SOC = df_SOC.dropna(subset=['occupation_title'])

# Reset index after removing rows
df_SOC = df_SOC.reset_index(drop=True)

# Display the result
print("SOC DF Shape:", df_SOC.shape)
print(df_SOC.head())
```

```
SOC DF Shape: (1447, 5)
  major_group minor_group broad_group detailed_occupation  \
0     11-0000         NaN         NaN                 NaN
1     11-0000     11-1000         NaN                 NaN
2     11-0000     11-1000     11-1010                 NaN
3     11-0000     11-1000     11-1010             11-1011
4     11-0000     11-1000     11-1020             11-1011

                  occupation_title
0           Management Occupations
1                   Top Executives
2                 Chief Executives
3                 Chief Executives
4  General and Operations Managers
```

```python
# Step 5.2: Remove rows where detailed_occupation is missing
df_SOC = df_SOC.dropna(subset=['detailed_occupation'])

# Reset index after removing rows
df_SOC = df_SOC.reset_index(drop=True)

# Display the result
print("SSOC DF Shape:", df_SOC.shape)
print(df_SOC.head(10))
```

```
SSOC DF Shape: (1444, 5)
  major_group minor_group broad_group detailed_occupation  \
0     11-0000     11-1000     11-1010             11-1011   
1     11-0000     11-1000     11-1020             11-1011   
2     11-0000     11-1000     11-1020             11-1021   
3     11-0000     11-1000     11-1030             11-1021   
4     11-0000     11-1000     11-1030             11-1031   
5     11-0000      Nov-00     11-1030             11-1031   
6     11-0000      Nov-00     11-2010             11-1031   
7     11-0000      Nov-00     11-2010              Nov-11   
8     11-0000      Nov-00     11-2020              Nov-11   
9     11-0000      Nov-00     11-2020              Nov-21   

                                                            occupation_title  
0                                                           Chief Executives  
1                                            General and Operations Managers  
2                                            General and Operations Managers  
3                                                                Legislators  
4                                                                Legislators  
5   Advertising, Marketing, Promotions, Public Relations, and Sales Managers  
6                                        Advertising and Promotions Managers  
7                                        Advertising and Promotions Managers  
8                                               Marketing and Sales Managers  
9                                                         Marketing Managers  
```

```
# Step 6: Make sure all occupation codes look like XX-XXXX (not changed to dates like 'Nov
-00')

# Create function to convert to standard format
def standardize_soc_code(code, major_group):

    if pd.isna(code):
        return code

    code = str(code).strip()

# If it's in "Nov-XX" format covert to standard format XX-XXXX where
# Major Group: XX-0000 (first 2 digits significant, rest zeros)
# Minor Group: XX-X000 (first 3 digits significant, rest zeros)
# Broad Group: XX-XX00 (first 4 digits significant, rest zeros)
# Detailed Occupation: XX-XXXX (all digits significant)

    if 'Nov' in code or not '-' in code:
        prefix = str(major_group)[:2]
        numbers = ''.join(filter(str.isdigit, code))
        numbers = numbers.zfill(4)
        return f"{prefix}-{numbers}"

    parts = code.split('-')
    if len(parts) == 2:
        prefix = str(major_group)[:2]
        numbers = parts[1].zfill(4)
        return f"{prefix}-{numbers}"

    return code

# Apply the standardization to each column
df_SOC['minor_group'] = df_SOC.apply(
    lambda row: standardize_soc_code(row['minor_group'], row['major_group']), axis=1)

df_SOC['broad_group'] = df_SOC.apply(
    lambda row: standardize_soc_code(row['broad_group'], row['major_group']), axis=1)

df_SOC['detailed_occupation'] = df_SOC.apply(
    lambda row: standardize_soc_code(row['detailed_occupation'], row['major_group']), axis
=1)

# Show result
print(df_SOC[['major_group', 'minor_group', 'broad_group', 'detailed_occupation']].head(2
0))
```

```
   major_group minor_group broad_group detailed_occupation
0      11-0000     11-1000     11-1010             11-1011
1      11-0000     11-1000     11-1020             11-1011
2      11-0000     11-1000     11-1020             11-1021
3      11-0000     11-1000     11-1030             11-1021
4      11-0000     11-1000     11-1030             11-1031
5      11-0000     11-0000     11-1030             11-1031
6      11-0000     11-0000     11-2010             11-1031
7      11-0000     11-0000     11-2010             11-0011
8      11-0000     11-0000     11-2020             11-0011
9      11-0000     11-0000     11-2020             11-0021
10     11-0000     11-0000     11-2020             11-0022
11     11-0000     11-0000     11-2030             11-0022
12     11-0000     11-0000     11-2030             11-0032
13     11-0000     11-0000     11-2030             11-0033
14     11-0000     11-0000     11-2030             11-0033
15     11-0000     11-0000     11-3010             11-0033
16     11-0000     11-0000     11-3010             11-0012
17     11-0000     11-0000     11-3010             11-0013
18     11-0000     11-0000     11-3020             11-0013
19     11-0000     11-0000     11-3020             11-0021
```

```python
# Step 6: Save the cleaned file to output folder for loading into SQL DB in Milestone 5

# Define the output file path
output_dir = os.path.join('..', 'output')
output_file = os.path.join(output_dir, 'SOC_DB.csv')

# Save as CSV
df_SOC.to_csv(output_file, index=False)

# Verify the file was created
if os.path.exists(output_file):
    print(f"File successfully saved to: {output_file}")

else:
    print("Error: File was not created")
```

```
File successfully saved to: ..\output\SOC_DB.csv
```

```
# Preview the output file
output_file = os.path.join('..', 'output', 'SOC_DB.csv')

try:

    df_preview = pd.read_csv(output_file)
    print("\nSOC Structure Final:")
    print(df_preview.head(15))

except FileNotFoundError:
    print(f"Error: File not found at {output_file}")
except Exception as e:
    print(f"An error occurred while reading the file: {e}")
```

```
SOC Structure Final:
   major_group minor_group broad_group detailed_occupation  \
0      11-0000     11-1000     11-1010             11-1011
1      11-0000     11-1000     11-1020             11-1011
2      11-0000     11-1000     11-1020             11-1021
3      11-0000     11-1000     11-1030             11-1021
4      11-0000     11-1000     11-1030             11-1031
5      11-0000     11-0000     11-1030             11-1031
6      11-0000     11-0000     11-2010             11-1031
7      11-0000     11-0000     11-2010             11-0011
8      11-0000     11-0000     11-2020             11-0011
9      11-0000     11-0000     11-2020             11-0021
10     11-0000     11-0000     11-2020             11-0022
11     11-0000     11-0000     11-2030             11-0022
12     11-0000     11-0000     11-2030             11-0032
13     11-0000     11-0000     11-2030             11-0033
14     11-0000     11-0000     11-2030             11-0033


                                           occupation_title
0                                          Chief Executives
1                           General and Operations Managers
2                           General and Operations Managers
3                                               Legislators
4                                               Legislators
5  Advertising, Marketing, Promotions, Public Relations, and Sales Managers
6                        Advertising and Promotions Managers
7                        Advertising and Promotions Managers
8                               Marketing and Sales Managers
9                                        Marketing Managers
10                                           Sales Managers
11                 Public Relations and Fundraising Managers
12                                Public Relations Managers
13                                     Fundraising Managers
14                          Operations Specialties Managers
```

Cleaning and Formatting NAICS Data

```
# Step 1.1: Remove whitespace
df_NAICS = df_NAICS.apply(lambda x: x.str.strip() if x.dtype == "object" else x)
```

```python
# Step 1.2: Remove rows where the "Sector" column is empty
df_NAICS = df_NAICS.loc[~df_NAICS['Sector'].isna()].copy()
```

```python
# Step 2: Modify column names to remove sub-columns under "6-digit Industries"
df_NAICS = df_NAICS.rename(columns={
    'U. S. Census Bureau – NAICS structure by industry': 'Sector',
    'Unnamed: 1': 'Name',
    'Unnamed: 2': 'Subsectors (3-digit)',
    'Unnamed: 3': 'detailed_occupation',
    'Unnamed: 4': 'occupation_title',
    'Unnamed: 5': '6-digit Industries - U.S. Detail',
    'Unnamed: 6': '6-digit Industries - Same as 5-digit',
    'Unnamed: 7': '6-digit Industries - Total'
})


# Display the NAICS Structure after renaming
print("\nNAICS structure by industry:")
print(df_NAICS.head(30))
```

```
NAICS structure by industry:
   Sector  \
1      11
2      21
3      22
4      23
5   31-33
6      42
7   44-45
8   48-49
9      51
10     52
11     53
12     54
13     55
14     56
15     61
16     62
17     71
18     72
19     81
20     92

                                                                     Name
\
1                                  Agriculture, Forestry, Fishing and Hunting
2                               Mining, Quarrying, and Oil and Gas Extraction
3                                                                  Utilities
4                                                               Construction
5                                                              Manufacturing
6                                                            Wholesale Trade
7                                                               Retail Trade
8                                             Transportation and Warehousing
9                                                                Information
10                                                     Finance and Insurance
11                                         Real Estate and Rental and Leasing
12                           Professional, Scientific, and Technical Services
13                                    Management of Companies and Enterprises
14  Administrative and Support and Waste Management and Remediation Services
15                                                      Educational Services
16                                         Health Care and Social Assistance
17                                       Arts, Entertainment, and Recreation
18                                           Accommodation and Food Services
19                              Other Services (except Public Administration)
20                                                     Public Administration

    Subsectors (3-digit)  Industry Groups (4-digit)  \
1                    5.0                       19.0
2                    3.0                        5.0
3                    1.0                        3.0
4                    3.0                       10.0
5                   21.0                       86.0
6                    3.0                       19.0
7                    9.0                       24.0
8                   11.0                       29.0
9                    6.0                       11.0
10                   5.0                       11.0
```

```
11                    3.0                        8.0
12                    1.0                        9.0
13                    1.0                        1.0
14                    2.0                       11.0
15                    1.0                        7.0
16                    4.0                       18.0
17                    3.0                        9.0
18                    2.0                        6.0
19                    4.0                       14.0
20                    8.0                        8.0

    NAICS Industries (5-digit) 6-digit Industries  \
1                         42.0                 32   
2                         11.0                 14   
3                          6.0                 10   
4                         28.0                  4   
5                        176.0                249   
6                         69.0                  0   
7                         48.0                 16   
8                         42.0                 25   
9                         24.0                 10   
10                        27.0                 13   
11                        17.0                 11   
12                        35.0                 20   
13                         1.0                  3   
14                        29.0                 25   
15                        12.0                  7   
16                        30.0                 16   
17                        23.0                  3   
18                        10.0                  8   
19                        30.0                 24   
20                        29.0                  0   

   6-digit Industries - Same as 5-digit 6-digit Industries - Total
1                                    32                         64
2                                     7                         21
3                                     4                         14
4                                    27                         31
5                                    97                        346
6                                    69                         69
7                                    41                         57
8                                    32                         57
9                                    19                         29
10                                   22                         35
11                                   13                         24
12                                   29                         49
13                                    0                          3
14                                   19                         44
15                                   10                         17
16                                   23                         39
17                                   22                         25
18                                    7                         15
19                                   20                         44
20                                   29                         29
```

```python
# Step 3: Create function to expand ranges in the dataframe
def expand_ranges(df):
    expanded_rows = []

    for _, row in df.iterrows():
        name = str(row['Sector'])
        if '-' in name:
            try:
                start, end = map(int, name.split('-'))

                for num in range(start, end + 1):
                    new_row = row.copy()
                    new_row['Sector'] = str(num)
                    expanded_rows.append(new_row)
            except ValueError:
                expanded_rows.append(row)
        else:
            expanded_rows.append(row)

    # Create new dataframe with expanded rows
    return pd.DataFrame(expanded_rows)

# Apply the expansion to the NAICS dataframe
df_NAICS = expand_ranges(df_NAICS)

# Reset index
df_NAICS = df_NAICS.reset_index(drop=True)


print("\nNAICS structure by industry:")
print(df_NAICS)
```

```
NAICS structure by industry:
   Sector  \
0      11
1      21
2      22
3      23
4      31
5      32
6      33
7      42
8      44
9      45
10     48
11     49
12     51
13     52
14     53
15     54
16     55
17     56
18     61
19     62
20     71
21     72
22     81
23     92

                                                                      Name
\
0                                 Agriculture, Forestry, Fishing and Hunting
1                              Mining, Quarrying, and Oil and Gas Extraction
2                                                                  Utilities
3                                                               Construction
4                                                              Manufacturing
5                                                              Manufacturing
6                                                              Manufacturing
7                                                            Wholesale Trade
8                                                               Retail Trade
9                                                               Retail Trade
10                                            Transportation and Warehousing
11                                            Transportation and Warehousing
12                                                               Information
13                                                    Finance and Insurance
14                                         Real Estate and Rental and Leasing
15                          Professional, Scientific, and Technical Services
16                                   Management of Companies and Enterprises
17  Administrative and Support and Waste Management and Remediation Services
18                                                      Educational Services
19                                         Health Care and Social Assistance
20                                       Arts, Entertainment, and Recreation
21                                           Accommodation and Food Services
22                             Other Services (except Public Administration)
23                                                     Public Administration

    Subsectors (3-digit)  Industry Groups (4-digit)  \
0                    5.0                       19.0
1                    3.0                        5.0
```

```
2                    1.0                        3.0
3                    3.0                       10.0
4                   21.0                       86.0
5                   21.0                       86.0
6                   21.0                       86.0
7                    3.0                       19.0
8                    9.0                       24.0
9                    9.0                       24.0
10                  11.0                       29.0
11                  11.0                       29.0
12                   6.0                       11.0
13                   5.0                       11.0
14                   3.0                        8.0
15                   1.0                        9.0
16                   1.0                        1.0
17                   2.0                       11.0
18                   1.0                        7.0
19                   4.0                       18.0
20                   3.0                        9.0
21                   2.0                        6.0
22                   4.0                       14.0
23                   8.0                        8.0

    NAICS Industries (5-digit) 6-digit Industries  \
0                         42.0                 32
1                         11.0                 14
2                          6.0                 10
3                         28.0                  4
4                        176.0                249
5                        176.0                249
6                        176.0                249
7                         69.0                  0
8                         48.0                 16
9                         48.0                 16
10                        42.0                 25
11                        42.0                 25
12                        24.0                 10
13                        27.0                 13
14                        17.0                 11
15                        35.0                 20
16                         1.0                  3
17                        29.0                 25
18                        12.0                  7
19                        30.0                 16
20                        23.0                  3
21                        10.0                  8
22                        30.0                 24
23                        29.0                  0

   6-digit Industries - Same as 5-digit 6-digit Industries - Total
0                                    32                         64
1                                     7                         21
2                                     4                         14
3                                    27                         31
4                                    97                        346
5                                    97                        346
6                                    97                        346
```

```
7                                     69                          69
8                                     41                          57
9                                     41                          57
10                                    32                          57
11                                    32                          57
12                                    19                          29
13                                    22                          35
14                                    13                          24
15                                    29                          49
16                                     0                           3
17                                    19                          44
18                                    10                          17
19                                    23                          39
20                                    22                          25
21                                     7                          15
22                                    20                          44
23                                    29                          29
```

```python
# Step 4: Save the cleaned file to output folder for loading into SQL DB in Milestone 5

# Output file path
output_dir = os.path.join('..', 'output')
output_file = os.path.join(output_dir, 'NAICS_DB.csv')

# Save as CSV
df_NAICS.to_csv(output_file, index=False)

# Verify the file was created
if os.path.exists(output_file):
    print(f"File successfully saved to: {output_file}")

else:
    print("Error: File was not created")
```

```
File successfully saved to: ..\output\NAICS_DB.csv
```

```python
# Preview the output file
output_file = os.path.join('..', 'output', 'NAICS_DB.csv')
try:
    df_preview = pd.read_csv(output_file)
    print("\nNAICS Structure:")
    pd.set_option('display.max_columns', None)
    pd.set_option('display.width', None)
    pd.set_option('display.max_colwidth', None)
    print(df_preview.head().to_string(index=False))
except FileNotFoundError:
    print(f"Error: File not found at {output_file}")
except Exception as e:
    print(f"An error occurred while reading the file: {e}")
```

```
NAICS Structure:
 Sector                                          Name  Subsectors (3-digit)
Industry Groups (4-digit)  NAICS Industries (5-digit)  6-digit Industries  6-
digit Industries - Same as 5-digit  6-digit Industries - Total
     11    Agriculture, Forestry, Fishing and Hunting                   5.0
19.0                        42.0                   32
32                          64
     21 Mining, Quarrying, and Oil and Gas Extraction                   3.0
5.0                        11.0                   14
7                          21
     22                                     Utilities                   1.0
3.0                         6.0                   10
4                          14
     23                                  Construction                   3.0
10.0                        28.0                    4
27                          31
     31                                 Manufacturing                  21.0
86.0                       176.0                  249
97                         346
```

Ethical Implications Of Data Wrangling SOC and NAICS Codes Data

While working with SOC and NAICS datasets, I performed the following cleaning and formating steps.

SOC (Standard Occupational Classification) Data Cleaning and formating steps: - Removed first 7 rows containing metadata
- Stripped whitespace from all string columns
- Renamed columns
- Forward filled hierarchy levels for all group columns
- Removed rows with missing occupation titles and missing detailed occupations
- Standardized occupation codes
- Saved cleaned data to 'SOC_DB.csv" to output folder for loading into SQL DB in Milestone 5.
-
NAICS (North American Industry Classification System) Data Cleaning and Formating Steps: - Stripped whitespace from all string columns
- REmoved rows where the 'Sector' column was empty
- Renamed columns to remove sub-columns under "6-digit Industries"
- Created function to expand ranges in 'Sector' column
- Expanded ranges into individual rows
- Saved cleaned data to 'NAICS_DB.csv" to output folder for loading into SQL DB in Milestone 5.
-
Ethical Implications: These datasets are public and come from trusted government sources. Therefore, they are ethically safe to use for my research. However, during the wrangling process, there was a small risk that I made incorrect assumptions during forward-filling missing values or labeling split sectors. All changes to the original data were documented for future reference to avoid misinterpretation and stay responsible.