```
import pandas as pd

# Define the file path
file_path = 'Address Analysis.xlsx'

# Load the data from the first sheet
df = pd.read_excel(file_path, sheet_name='Sheet1')

# Update the affiliation categories with new keywords based on the
additional sample
affiliation_categories_final = {
    'University': [
        'university', 'universität', 'universidad', 'université',
'univ',
        'institute of technology', 'college', 'polytechnic',
'technische', 'schule',
        'trinity college', 'kings college', 'baylor college', 'eastern
virginia medical school',
        'geisel school of medicine', 'indian institute of technology',
'imperial college',
        'simmons college', 'ut health san antonio', 'ucla',
        'leiden university medical center','mit', 'acad med ctr', 'chu
quebec',
        'royal college of surgeons', 'hannover medical school', 'new
york medical college',
        'ucl', 'rhein west university', 'suny stony brook', 'cambridge
university',
        'uam', 'indian institute of science education and research'
    ],
    'Research Institute': [
        'research institute', 'institut', 'institute', 'research
center', 'research centre',
        'leibniz', 'helmholtz', 'irccs', 'natl inst', 'inst', 'the
scripps', 'cea saclay',
        'nci', 'garvan inst', 'karolinska inst', 'pasteur inst',
'pacific nw national laboratory',
        'mayo clinic', 'max planck', 'german cancer', 'vtt tech res',
'tianjin inst hlt',
        'chinese acad sci', 'rajiv gandhi', 'inst basic med sci',
'sanquin research',
        'int agency research on cancer', 'inst hematology', 'inst
gustave roussy',
        'fraunhofer institute', 'murdoch children\'s research
institute', 'cnr',
        'phys tech bundesanstalt', 'inra', 'nichd', 'fom institute',
'van andel research institute',
        'noguchi institute', 'ceinge biotecnologie avanzate',
        'south australian health and medical research institute', 'agr
victoria',
        'austrian academy of sciences', 'erasmus mc', 'irblleida',
```

```
'korea institute of science and technology',
        'slovak academy of sciences', 'institute of myeloma and bone
cancer',
        'csir igib', 'rudjer boskovic institute', 'niser', 'feodor
lynen research institute',
        'victor chang cardiac research institute', 'institute of
cancer research',
        'plant & food research', 'iasri', 'international center for
genetic engineering and biotechnology',
        'center for agriculture and biosciences international'
    ],
    'Hospital': [
        'hospital', 'clinic', 'health', 'medical center', 'medical
centre', 'kantonsspital',
        'osaka med', 'hosp', 'cleveland clinic', 'montreal heart', 'mt
sinai medical',
        'gen infirmary', 'azienda ospedaliera', 'sunnybrook health',
'klin str',
        'st joseph\'s health', 'memorial sloan kettering cancer
center',
        'lombardi comprehensive cancer center', 'ega institute for
women\'s health',
        'md anderson cancer center', 'channing division of network
medicine', 'framingham heart study',
        'natl jewish health', 'beth israel deaconess medical center',
'boston medical center'
    ],
    'Industry': [
        'company', 'corp', 'inc', 'llc', 'ltd', 'co.', 'corporation',
'astrazeneca', 'janssen',
        'mosaiques diagnostics', 'texas biomed', 'ciphergen
biosystems', 'sequenom',
        'roche diagnostics', 'amer health found', 'biosyst
international', 'euroimmun ag', 'nestle',
        'amicus therapeutics', 'sanofi', 'novartis', 'zora biosciences
oy', 'kompetenzzentrum'
    ],
    'Government': [
        'government', 'national laboratory', 'ministry', 'department
of', 'ctr dis control',
        'armed forces', 'nist', 'minist hlth', 'nia, nih', 'usda ars',
'adm nacl lab',
        'natl canc ctr', 'mmc', 'swedish defence research agency',
'nih', 'us fda',
        'european commission', 'california department of public
health', 'national center for global health and medicine'
    ],
    'Other': []  # This will be used as a fallback category
}
```

```python
# Function to categorize addresses based on final refined keywords
def categorize_affiliation_final(address):
    if isinstance(address, str):
        address_lower = address.lower()
        for category, keywords in
affiliation_categories_final.items():
            if any(keyword in address_lower for keyword in keywords):
                return category
    return 'Other'

# Apply the final refined function to the 'Reprint Addresses' column
and create a new column 'Affiliation Type Final Refined'
df['Affiliation Type'] = df['Reprint
Addresses'].apply(categorize_affiliation_final)

# Save the dataframe with the new column to an Excel file
output_file_path = 'Updated_Address_Analysis.xlsx'
df.to_excel(output_file_path, index=False)

# Calculate the final refined distribution of different affiliation
types
affiliation_counts_final = df['Affiliation Type'].value_counts()

# Display the final refined results
print(affiliation_counts_final)
```

```
University            8030
Research Institute    1283
Other                 1153
Hospital               412
Industry               186
Government             172
Name: Affiliation Type, dtype: int64
```

```python
# Calculate the total number of addresses
total_addresses = len(df)

# Calculate the percentage of each affiliation type
affiliation_percentages = (affiliation_counts_final / total_addresses)
* 100

# Print the results
print(f"Total addresses: {total_addresses}")
for affiliation_type, count in affiliation_counts_final.items():
    percentage = affiliation_percentages[affiliation_type]
    print(f"{affiliation_type} addresses: {count} ({percentage:.2f}
%)")
```

```
Total addresses: 11236
University addresses: 8030 (71.47%)
Research Institute addresses: 1283 (11.42%)
Other addresses: 1153 (10.26%)
Hospital addresses: 412 (3.67%)
Industry addresses: 186 (1.66%)
Government addresses: 172 (1.53%)
```