Source Code Notebook

Background of the Dataset

The **Philippine Statistics Authority (PSA)** is the central statistical authority of the Philippines, responsible for collecting, compiling, and disseminating official data across various sectors. Established through the Philippine Statistical Act of 2013, the PSA integrates four previously separate agencies: the National Statistics Office (NSO), the National Statistical Coordination Board (NSCB), the Bureau of Agricultural Statistics (BAS), and the Bureau of Labor and Employment Statistics (BLES). Its primary mission is to provide timely, accurate, and relevant statistics for policy-making, planning, and research.

The **Family Income and Expenditure Survey (FIES)** is a nationwide survey of households undertaken every three years. It is the main source of data on family income and expenditure, which include among others, levels of consumption by item of expenditure as well as sources of income in cash and in kind. The results of FIES provide information on the levels of living and disparities in income of Filipino families, as well as their spending patterns. The survey aims to provide a detailed picture of the economic conditions of households, measuring disparities in income and spending patterns across various regions and socio-economic groups. While usually, the FIES is a survey conducted every three years, the 2023 FIES, the dataset used in this project, is the first biennial survey of the FIES. According to an article by Dr. Mapa in July 10, 2023, this change in standard procedure is due to respond to the clamor for more frequent and timely income and expenditure statistics, as well as poverty statistics. The results of the 2023 FIES will provide inputs to the 2023 Official Poverty Statistics, which will aid the government in planning, programming, policy formulation and decision-making.

Additional information about the 2023 FIES is that for the first time, the 2023 Geo-enabled Master Sample with sample size of about 180,000 households, which includes separate domains for Maguindanao del Norte and Maguindanao del Sur. Further, the 2023 FIES is the second in the series to implement the Computer-Aided Personal Interviewing (CAPI) system, which replaces the traditional Paper and Pencil Interviewing method. The use of the CAPI system eliminates the manual encoding of data obtained through paper questionnaires. It facilitates the data cleaning as consistency checks, skipping patterns and error detection are already embedded in the system. These features serve as safeguard to data quality and shortens the survey's timetable of operations. The 2023 FIES Visit 1 was conducted July 9 to July 31 in 2023, while the 2023 FIES Visit 2 was conducted from January 8 to 31 in 2024. Attached here is link used to download the FIES dataset along with the metadata where the questionnaire is located. https://psada.psa.gov.ph/catalog/FIES/about

The dataset used for this project is found on the **PSADA Microdata Catalogue** accessed here: https://psada.psa.gov.ph/home. ALong with the FIES database, there are also numerous databases made open for public use with focuses on ICT, Tourism, Poverty Indicators, Wage Rates, and much more. Do note that you will need to have an account registered to access these resources, and along with the account, you will need to be subjected to the PSADA Microdata Catalogue Terms and Conditions.

References:

1. https://rsso01.psa.gov.ph/statistics/fies/about
2. https://www.psa.gov.ph/statistics/income-expenditure/fies/node/1684059988

In [40]:
```
import pandas as pd
import numpy as np
fies_df = pd.read_csv('datasets/fies_2023_volume1_494887610821.csv')
fies_df
```

```
/var/folders/mp/c7pgmq8j0472f05vnx5h16000000gn/T/ipykernel_81756/1495670288.py:3: DtypeWarning: Col
umns (77) have mixed types. Specify dtype option on import or set low_memory=False.
  fies_df = pd.read_csv('datasets/fies_2023_volume1_494887610821.csv')
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Out[40]:

	RDMD_ID	Region	Province	Household ID	RECODED PROVINCE	Family Size	Salaries/ Wages from Regular Employment	Salaries/ Wages from Seasonal Employment	Income from Salaries and Wages	Ne Share o Crops Fruits etc (Tot Ne Valu o Share
0	1	1	28	1	2800	2.5	119000	0	119000	(
1	2	1	28	2	2800	6.0	154400	0	154400	(
2	3	1	28	3	2800	3.5	683452	0	683452	(
3	4	1	28	4	2800	2.5	48200	0	48200	1000(
4	5	1	28	5	2800	3.0	400994	0	400994	(
...	...	...	...	...	...	...	...	...	...	..
163263	163264	17	59	163264	5900	3.0	42600	5984	48584	(
163264	163265	17	59	163265	5900	7.0	117600	56800	174400	(
163265	163266	17	59	163266	5900	3.5	0	65800	65800	(
163266	163267	17	59	163267	5900	4.0	121400	0	121400	(
163267	163268	17	59	163268	5900	3.0	0	0	0	(

163268 rows × 91 columns

Data Dictionary

In [41]:
```
fies_df.columns
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Out[41]:
```
Index(['RDMD_ID', 'Region', 'Province', 'Household ID', 'RECODED PROVINCE',
       'Family Size', 'Salaries/Wages from Regular Employment',
       'Salaries/Wages from Seasonal Employment',
       'Income from Salaries and Wages',
       'Net Share of Crops, Fruits, etc. (Tot. Net Value of Share)',
       'Cash Receipts, Support, etc. from Abroad',
       'Cash Receipts, Support, etc. from Domestic Source',
       'Rentals Received from Non-Agri Lands, etc.', 'Unnamed: 13',
       'Pension and Retirement Benefits', 'Dividends from Investment',
       'Other Sources of Income NEC', 'Family Sustenance Activities',
       'Total Received as Gifts', 'Crop Farming and Gardening',
       'Livestock and Poultry Raising', 'Fishing', 'Forestry and Hunting',
       'Wholesale and Retail', 'Manufacturing',
       'Transportation, Storage Services', 'Entrep. Activities NEC',
       'Entrep. Activities NEC.1', 'Entrep. Activities NEC.2',
       'Hhld, Income from Entrepreneurial Activities, Total', 'Losses from EA',
       'Cereal and Cereal Preparations (Total)', 'Meat and Meat Preparations',
       'Fish and Marine Products (Total)', 'Dairy Products and Eggs (Total)',
       'Oils and Fats (Total)', 'Fruits and Vegetables', 'Vegetables (Total)',
       'Sugar, Jam and Honey (Total)', 'Food Not Elsewhere Classified (Total)',
       'Fruit and vegetable juices', 'Coffee, Cocoa and Tea (Total)',
       'Tea (total)  expenditure', 'Cocoa (total)  expenditure',
       'Main Source of Water Supply (2nd visit only)', 'Softdrinks',
       'Other Non Alcoholic Beverages', 'Alcoholic Beverages (Total)',
       'Tobacco (Total)', 'Other Vegetables (Total)', 'Services_Primary_Goods',
       'Alcohol Procduction Services', 'Total Food Consumed at Home (Total)',
       'Food Regularly Consumed Outside The Home (Total)', 'Hhld, Food',
       'Clothing, Footwear and Other Wear', 'Housing and water (Total)',
       'Actual House Rent', 'Imputed House Rental Value',
       'Imputed Housing Benefit Rental Value', 'House Rent/Rental Value',
       'Furnishings, Household Equipment & Routine Household Mainte',
       'Health (Total)', 'Transportation (Total)', 'Communication (Total)',
       'Recreation and Culture (Total)', 'Education (Total)', 'Insurance',
       'Miscellaneous Goods and Services (Total)', 'Durable Furniture',
       'Special Family Occasion',
       'Other Expenditure (inc. Value Consumed, Losses)',
       'Other Disbursements', 'Accomodation Services',
       'Total Non-Food Expenditure', 'Hhld, Income, Total',
       'Hhld, Expenditures, Total', 'Total Household Disbursements',
       'Other Receipts', 'Total Receipts', 'Psu (Recode)', 'Raising Factor',
       'Final Population Weights', 'Urban / Rural', 'Per Capita Income',
       'NPCINC', 'RPCINC', 'Per Capita Income Decile (Province)', 'pPCINC',
       'Per Capita Income Decile (Region with Negros Island Region (NIR))',
       'Region (with NIR)'],
      dtype='object')
```

In [42]:
```
fies_column_descriptions = {
    'RDMD_ID': 'Unique identifier for the record',
    'Region': 'Region code',
    'Province': 'Province code',
    'Household ID': 'Unique household identifier',
    'RECODED PROVINCE': 'Recoded province information',
    'Family Size': 'Number of people in the household',
    'Salaries/Wages from Regular Employment': 'Income from regular employment',
    'Salaries/Wages from Seasonal Employment': 'Income from seasonal employment',
    'Income from Salaries and Wages': 'Total income from salaries and wages',
    'Net Share of Crops, Fruits, etc. (Tot. Net Value of Share)': 'Net value from crop and fruit sh
    'Cash Receipts, Support, etc. from Abroad': 'Cash support received from abroad',
    'Cash Receipts, Support, etc. from Domestic Source': 'Cash support received domestically',
    'Rentals Received from Non-Agri Lands, etc.': 'Income from land rentals (non-agricultural)',
    'Unnamed: 13': 'Unknown or unnamed column',
    'Pension and Retirement Benefits': 'Income from pensions and retirement',
    'Dividends from Investment': 'Income from dividends',
    'Other Sources of Income NEC': 'Other sources of income not elsewhere classified',
    'Family Sustenance Activities': 'Income from family sustenance activities',
    'Total Received as Gifts': 'Total gifts received by the household',
    'Crop Farming and Gardening': 'Income from crop farming and gardening',
    'Livestock and Poultry Raising': 'Income from livestock and poultry raising',
    'Fishing': 'Income from fishing activities',
    [illegible]: 'Income from forestry and hunting',
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
    'Wholesale and Retail': 'Income from wholesale and retail business',
    'Manufacturing': 'Income from manufacturing activities',
    'Transportation, Storage Services': 'Income from transportation and storage services',
    'Entrep. Activities NEC': 'Income from entrepreneurial activities (not elsewhere classified)',
    'Entrep. Activities NEC.1': 'Income from entrepreneurial activities (additional category 1)',
    'Entrep. Activities NEC.2': 'Income from entrepreneurial activities (additional category 2)',
    'Hhld, Income from Entrepreneurial Activities, Total': 'Total household income from entrepreneu
    'Losses from EA': 'Losses from entrepreneurial activities',
    'Cereal and Cereal Preparations (Total)': 'Expenditure on cereals and cereal preparations',
    'Meat and Meat Preparations': 'Expenditure on meat and meat preparations',
    'Fish and Marine Products (Total)': 'Expenditure on fish and marine products',
    'Dairy Products and Eggs (Total)': 'Expenditure on dairy products and eggs',
    'Oils and Fats (Total)': 'Expenditure on oils and fats',
    'Fruits and Vegetables': 'Expenditure on fruits and vegetables',
    'Vegetables (Total)': 'Expenditure on vegetables',
    'Sugar, Jam and Honey (Total)': 'Expenditure on sugar, jam, and honey',
    'Food Not Elsewhere Classified (Total)': 'Expenditure on other food items',
    'Fruit and vegetable juices': 'Expenditure on fruit and vegetable juices',
    'Coffee, Cocoa and Tea (Total)': 'Expenditure on coffee, cocoa, and tea',
    'Tea (total)  expenditure': 'Expenditure on tea',
    'Cocoa (total)  expenditure': 'Expenditure on cocoa',
    'Main Source of Water Supply (2nd visit only)': 'Main source of water supply (second visit)',
    'Softdrinks': 'Expenditure on soft drinks',
    'Other Non Alcoholic Beverages': 'Expenditure on other non-alcoholic beverages',
    'Alcoholic Beverages (Total)': 'Expenditure on alcoholic beverages',
    'Tobacco (Total)': 'Expenditure on tobacco products',
    'Other Vegetables (Total)': 'Expenditure on other types of vegetables',
    'Services_Primary_Goods': 'Expenditure on services and primary goods',
    'Alcohol Procduction Services': 'Expenditure on alcohol production services',
    'Total Food Consumed at Home (Total)': 'Total food consumed at home',
    'Food Regularly Consumed Outside The Home (Total)': 'Food consumed outside the home',
    'Hhld, Food': 'Household expenditure on food',
    'Clothing, Footwear and Other Wear': 'Expenditure on clothing, footwear, and other wear',
    'Housing and water (Total)': 'Expenditure on housing and water',
    'Actual House Rent': 'Expenditure on actual house rent',
    'Imputed House Rental Value': 'Imputed value of house rental',
    'Imputed Housing Benefit Rental Value': 'Imputed value of housing benefit rental',
    'House Rent/Rental Value': 'Expenditure on house rent/rental value',
    'Furnishings, Household Equipment & Routine Household Mainte': 'Expenditure on furnishings and
    'Health (Total)': 'Expenditure on health services and products',
    'Transportation (Total)': 'Expenditure on transportation',
    'Communication (Total)': 'Expenditure on communication services',
    'Recreation and Culture (Total)': 'Expenditure on recreation and culture',
    'Education (Total)': 'Expenditure on education',
    'Insurance': 'Expenditure on insurance',
    'Miscellaneous Goods and Services (Total)': 'Expenditure on miscellaneous goods and services',
    'Durable Furniture': 'Expenditure on durable furniture',
    'Special Family Occasion': 'Expenditure on special family occasions',
    'Other Expenditure (inc. Value Consumed, Losses)': 'Other expenditures including losses',
    'Other Disbursements': 'Other household disbursements',
    'Accomodation Services': 'Expenditure on accommodation services',
    'Total Non-Food Expenditure': 'Total non-food expenditure',
    'Hhld, Income, Total': 'Total household income',
    'Hhld, Expenditures, Total': 'Total household expenditures',
    'Total Household Disbursements': 'Total household disbursements',
    'Other Receipts': 'Other household receipts',
    'Total Receipts': 'Total receipts',
    'Psu (Recode)': 'Primary Sampling Unit (recoded)',
    'Raising Factor': 'Raising factor for survey results',
    'Final Population Weights': 'Final weights for population data',
    'Urban / Rural': 'Urban or rural classification',
    'Per Capita Income': 'Household per capita income',
    'NPCINC': 'National per capita income',
    'RPCINC': 'Regional per capita income',
    'Per Capita Income Decile (Province)': 'Per capita income decile in the province',
    'pPCINC': 'Provincial per capita income decile',
    'Per Capita Income Decile (Region with Negros Island Region (NIR))': 'Per capita income decile
    'Region (with NIR)': 'Region code including NIR'
}
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

In [43]:
```
fies_derivations = {
    'Total Receipts': 'Total Household Income + Other Receipts',
    'Hhld, Income, Total': 'Net Share of Crops, Fruits, etc. + Cash Receipts, Support, etc. from Ab
    'Hhld, Income from Entrepreneurial Activities, Total': 'Crop Farming and Gardening + Livestock
    'Total Household Disbursements': 'Total Household Expenditure + Other Disbursements',
    'Hhld, Expenditures, Total': 'Household Food + Total Non-Food Expenditure',
    'Hhld, Food': 'Total Food Consumed at Home + Food Regularly Consumed Outside The Home',
    'Total Food Consumed at Home (Total)': 'Cereal and Cereal Preparations + Meat and Meat Preparat
    'Total Non-Food Expenditure': 'Alcoholic Beverages + Tobacco + Other Vegetables + Services_Prim
}
```

In [44]:
```
fies_dataset_data_dict = pd.DataFrame({
    'Column Name': fies_df.columns,
    'Data Type': fies_df.dtypes,
    'Non-Null Count': fies_df.notnull().sum(),
    'Unique Values': fies_df.nunique(),
    'Description': [fies_column_descriptions.get(col, 'No desciption available') for col in fies_df
    'Derivations from other columns': [fies_derivations.get(col, '') for col in fies_df.columns]
})

fies_dataset_data_dict.to_csv('fies_dataset_data_dict.csv', index=False)
```

Preliminary Data Analysis

Data Cleaning and Wrangling

Handling Null Values

From the data dictionary, the Total Household Disbursements column has is the only one with an object datatype, suggesting mixed values of numbers, strings, etc.

In [45]:
```
def is_numeric(val):
    try:
        float(val)  # Try converting to float
        return True
    except (ValueError, TypeError):
        return False

non_numeric_mask = fies_df['Total Household Disbursements'].map(lambda x: not is_numeric(x))
non_numeric_values = fies_df[non_numeric_mask]
# for row_idx, value in non_numeric_values['Total Household Disbursements'].items():
#     print(f"Non-numeric value at Row {row_idx}: {value}") # commented out to save space on the ou
```

Upon further inspection, we see that the column contains whitespace values. The next step is to find out if the column has zero values to ascertain that null values and zeroes are different

In [46]:
```
## Find out if there are zero values in the total_household_disbursements column
print(fies_df['Total Household Disbursements'].value_counts().where(fies_df['Total Household Disbur
```

```
144056
```

There are zeroes, which means that the null values cannot be attributed to zero. To preserve data quality, we drop these columns

In [47]:
```
# Function to check if a value is whitespace or empty
def has_whitespace(val):
    return isinstance(val, str) and val.strip() == ''

whitespace_rows = fies_df.map(has_whitespace).any(axis=1)

whitespace_count = whitespace_rows.sum()

# Drop the columns
fies_df = fies_df[~whitespace_rows]
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
print(f"Shape after removing rows with whitespaces: {fies_df.shape}")
```

```
Shape after removing rows with whitespaces: (155536, 91)
```

In [48]:
```
# Display all columns with missing values
print(fies_df.isnull().sum())
```

```
RDMD_ID                                                            0
Region                                                             0
Province                                                           0
Household ID                                                       0
RECODED PROVINCE                                                   0
                                                                  ..
RPCINC                                                             0
Per Capita Income Decile (Province)                                0
pPCINC                                                             0
Per Capita Income Decile (Region with Negros Island Region (NIR))  0
Region (with NIR)                                                  0
Length: 91, dtype: int64
```

No more null values

Renaming Columns

In [49]:
```
# Renaming columns to follow standard snake_case naming convention
fies_df = fies_df.rename(columns={
    'RDMD_ID': 'rdmd_id',
    'Region': 'region',
    'Province': 'province',
    'Household ID': 'household_id',
    'RECODED PROVINCE': 'recoded_province',
    'Family Size': 'family_size',
    'Salaries/Wages from Regular Employment': 'regular_salaries_wages',
    'Salaries/Wages from Seasonal Employment': 'seasonal_salaries_wages',
    'Income from Salaries and Wages': 'total_salaries_wages',
    'Net Share of Crops, Fruits, etc. (Tot. Net Value of Share)': 'net_crop_fruit_share',
    'Cash Receipts, Support, etc. from Abroad': 'cash_receipts_abroad',
    'Cash Receipts, Support, etc. from Domestic Source': 'cash_receipts_domestic',
    'Rentals Received from Non-Agri Lands, etc.': 'non_agri_land_rentals',
    'Unnamed: 13': 'unnamed_13',
    'Pension and Retirement Benefits': 'pension_retirement_benefits',
    'Dividends from Investment': 'dividends_from_investment',
    'Other Sources of Income NEC': 'other_income_nec',
    'Family Sustenance Activities': 'family_sustenance_activities',
    'Total Received as Gifts': 'total_gifts_received',
    'Crop Farming and Gardening': 'income_crop_farming',
    'Livestock and Poultry Raising': 'income_livestock_poultry',
    'Fishing': 'income_fishing',
    'Forestry and Hunting': 'income_forestry_hunting',
    'Wholesale and Retail': 'income_wholesale_retail',
    'Manufacturing': 'income_manufacturing',
    'Transportation, Storage Services': 'income_transport_storage',
    'Entrep. Activities NEC': 'entrepreneurial_activities_nec',
    'Entrep. Activities NEC.1': 'entrepreneurial_activities_nec_1',
    'Entrep. Activities NEC.2': 'entrepreneurial_activities_nec_2',
    'Hhld, Income from Entrepreneurial Activities, Total': 'total_income_entrepreneurial_activities
    'Losses from EA': 'losses_from_entrepreneurial_activities',
    'Cereal and Cereal Preparations (Total)': 'expenditure_cereal_preparations',
    'Meat and Meat Preparations': 'expenditure_meat_preparations',
    'Fish and Marine Products (Total)': 'expenditure_fish_marine_products',
    'Dairy Products and Eggs (Total)': 'expenditure_dairy_eggs',
    'Oils and Fats (Total)': 'expenditure_oils_fats',
    'Fruits and Vegetables': 'expenditure_fruits_vegetables',
    'Vegetables (Total)': 'expenditure_vegetables',
    'Sugar, Jam and Honey (Total)': 'expenditure_sugar_jam_honey',
    'Food Not Elsewhere Classified (Total)': 'expenditure_other_food',
    'Fruit and vegetable juices': 'expenditure_fruit_vegetable_juices',
    'Coffee, Cocoa and Tea (Total)': 'expenditure_coffee_cocoa_tea',
    'Tea (total)  expenditure': 'expenditure_tea',
    iture': 'expenditure_cocoa',
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
    'Main Source of Water Supply (2nd visit only)': 'main_water_supply_second_visit',
    'Softdrinks': 'expenditure_softdrinks',
    'Other Non Alcoholic Beverages': 'expenditure_non_alcoholic_beverages',
    'Alcoholic Beverages (Total)': 'expenditure_alcoholic_beverages',
    'Tobacco (Total)': 'expenditure_tobacco',
    'Other Vegetables (Total)': 'expenditure_other_vegetables',
    'Services_Primary_Goods': 'expenditure_services_primary_goods',
    'Alcohol Procduction Services': 'expenditure_alcohol_production_services',
    'Total Food Consumed at Home (Total)': 'total_food_consumed_home',
    'Food Regularly Consumed Outside The Home (Total)': 'food_consumed_outside_home',
    'Hhld, Food': 'household_food_expenditure',
    'Clothing, Footwear and Other Wear': 'expenditure_clothing_footwear',
    'Housing and water (Total)': 'expenditure_housing_water',
    'Actual House Rent': 'actual_house_rent',
    'Imputed House Rental Value': 'imputed_house_rental_value',
    'Imputed Housing Benefit Rental Value': 'imputed_housing_benefit_rental_value',
    'House Rent/Rental Value': 'house_rent_rental_value',
    'Furnishings, Household Equipment & Routine Household Mainte': 'expenditure_furnishings_househo
    'Health (Total)': 'expenditure_health',
    'Transportation (Total)': 'expenditure_transportation',
    'Communication (Total)': 'expenditure_communication',
    'Recreation and Culture (Total)': 'expenditure_recreation_culture',
    'Education (Total)': 'expenditure_education',
    'Insurance': 'expenditure_insurance',
    'Miscellaneous Goods and Services (Total)': 'expenditure_miscellaneous_goods_services',
    'Durable Furniture': 'expenditure_durable_furniture',
    'Special Family Occasion': 'expenditure_special_family_occasion',
    'Other Expenditure (inc. Value Consumed, Losses)': 'other_expenditure',
    'Other Disbursements': 'other_disbursements',
    'Accomodation Services': 'expenditure_accommodation_services',
    'Total Non-Food Expenditure': 'total_non_food_expenditure',
    'Hhld, Income, Total': 'total_household_income',
    'Hhld, Expenditures, Total': 'total_household_expenditures',
    'Total Household Disbursements': 'total_household_disbursements',
    'Other Receipts': 'other_receipts',
    'Total Receipts': 'total_receipts',
    'Psu (Recode)': 'psu_recode',
    'Raising Factor': 'raising_factor',
    'Final Population Weights': 'final_population_weights',
    'Urban / Rural': 'urban_rural',
    'Per Capita Income': 'per_capita_income',
    'NPCINC': 'national_per_capita_income',
    'RPCINC': 'regional_per_capita_income',
    'Per Capita Income Decile (Province)': 'per_capita_income_decile_province',
    'pPCINC': 'provincial_per_capita_income_decile',
    'Per Capita Income Decile (Region with Negros Island Region (NIR))': 'per_capita_income_decile_
    'Region (with NIR)': 'region_with_nir'
})
```

Unnamed_13 Column

We opted to total the unnamed column and Other sources of income NEC because the unnamed column contributes to the total income of the household as some of the total income were inaccurate if the unnamed column wans't included

In [50]:
```
# Adding the value of unnamed_13 to the other_income_nec column
fies_df['other_income_nec'] = fies_df['other_income_nec'] + fies_df['unnamed_13']
fies_df = fies_df.drop(columns=['unnamed_13'])
```

Datatype Standardization

In [51]:
```
fies_df.info()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
<class 'pandas.core.frame.DataFrame'>
Index: 155536 entries, 0 to 163267
Data columns (total 90 columns):
 #   Column                                          Non-Null Count   Dtype
---  ------                                          --------------   -----
 0   rdmd_id                                         155536 non-null  int64
 1   region                                          155536 non-null  int64
 2   province                                        155536 non-null  int64
 3   household_id                                    155536 non-null  int64
 4   recoded_province                                155536 non-null  int64
 5   family_size                                     155536 non-null  float64
 6   regular_salaries_wages                          155536 non-null  int64
 7   seasonal_salaries_wages                         155536 non-null  int64
 8   total_salaries_wages                            155536 non-null  int64
 9   net_crop_fruit_share                            155536 non-null  int64
 10  cash_receipts_abroad                            155536 non-null  int64
 11  cash_receipts_domestic                          155536 non-null  int64
 12  non_agri_land_rentals                           155536 non-null  int64
 13  pension_retirement_benefits                     155536 non-null  int64
 14  dividends_from_investment                       155536 non-null  int64
 15  other_income_nec                                155536 non-null  int64
 16  family_sustenance_activities                    155536 non-null  int64
 17  total_gifts_received                            155536 non-null  float64
 18  income_crop_farming                             155536 non-null  int64
 19  income_livestock_poultry                        155536 non-null  int64
 20  income_fishing                                  155536 non-null  int64
 21  income_forestry_hunting                         155536 non-null  int64
 22  income_wholesale_retail                         155536 non-null  int64
 23  income_manufacturing                            155536 non-null  int64
 24  income_transport_storage                        155536 non-null  int64
 25  entrepreneurial_activities_nec                  155536 non-null  int64
 26  entrepreneurial_activities_nec_1                155536 non-null  int64
 27  entrepreneurial_activities_nec_2                155536 non-null  int64
 28  total_income_entrepreneurial_activities         155536 non-null  int64
 29  losses_from_entrepreneurial_activities          155536 non-null  int64
 30  expenditure_cereal_preparations                 155536 non-null  float64
 31  expenditure_meat_preparations                   155536 non-null  float64
 32  expenditure_fish_marine_products                155536 non-null  float64
 33  expenditure_dairy_eggs                          155536 non-null  float64
 34  expenditure_oils_fats                           155536 non-null  float64
 35  expenditure_fruits_vegetables                   155536 non-null  float64
 36  expenditure_vegetables                          155536 non-null  float64
 37  expenditure_sugar_jam_honey                     155536 non-null  float64
 38  expenditure_other_food                          155536 non-null  float64
 39  expenditure_fruit_vegetable_juices              155536 non-null  float64
 40  expenditure_coffee_cocoa_tea                    155536 non-null  float64
 41  expenditure_tea                                 155536 non-null  float64
 42  expenditure_cocoa                               155536 non-null  float64
 43  main_water_supply_second_visit                  155536 non-null  float64
 44  expenditure_softdrinks                          155536 non-null  float64
 45  expenditure_non_alcoholic_beverages             155536 non-null  float64
 46  expenditure_alcoholic_beverages                 155536 non-null  float64
 47  expenditure_tobacco                             155536 non-null  float64
 48  expenditure_other_vegetables                    155536 non-null  float64
 49  expenditure_services_primary_goods              155536 non-null  int64
 50  expenditure_alcohol_production_services         155536 non-null  int64
 51  total_food_consumed_home                        155536 non-null  float64
 52  food_consumed_outside_home                      155536 non-null  float64
 53  household_food_expenditure                      155536 non-null  float64
 54  expenditure_clothing_footwear                   155536 non-null  int64
 55  expenditure_housing_water                       155536 non-null  int64
 56  actual_house_rent                               155536 non-null  int64
 57  imputed_house_rental_value                      155536 non-null  int64
 58  imputed_housing_benefit_rental_value            155536 non-null  int64
 59  house_rent_rental_value                         155536 non-null  int64
 60  expenditure_furnishings_household_maintenance   155536 non-null  int64
 61  expenditure_health                              155536 non-null  int64
 62  expenditure_transportation                      155536 non-null  int64
 63  expenditure_communication                       155536 non-null  int64
 64  expenditure_recreation_culture                  155536 non-null  int64
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
 65  expenditure_education                          155536 non-null  int64
 66  expenditure_insurance                          155536 non-null  int64
 67  expenditure_miscellaneous_goods_services       155536 non-null  int64
 68  expenditure_durable_furniture                  155536 non-null  int64
 69  expenditure_special_family_occasion            155536 non-null  int64
 70  other_expenditure                              155536 non-null  int64
 71  other_disbursements                            155536 non-null  int64
 72  expenditure_accommodation_services             155536 non-null  int64
 73  total_non_food_expenditure                     155536 non-null  float64
 74  total_household_income                         155536 non-null  float64
 75  total_household_expenditures                   155536 non-null  float64
 76  total_household_disbursements                  155536 non-null  object
 77  other_receipts                                 155536 non-null  int64
 78  total_receipts                                 155536 non-null  float64
 79  psu_recode                                     155536 non-null  int64
 80  raising_factor                                 155536 non-null  float64
 81  final_population_weights                       155536 non-null  float64
 82  urban_rural                                    155536 non-null  int64
 83  per_capita_income                              155536 non-null  float64
 84  national_per_capita_income                     155536 non-null  int64
 85  regional_per_capita_income                     155536 non-null  int64
 86  per_capita_income_decile_province              155536 non-null  int64
 87  provincial_per_capita_income_decile            155536 non-null  int64
 88  per_capita_income_decile_region_nir            155536 non-null  int64
 89  region_with_nir                                155536 non-null  int64
dtypes: float64(31), int64(58), object(1)
memory usage: 108.0+ MB
```

In [52]:

```python
# Select float columns
float_columns = fies_df.select_dtypes(include=['float']).columns

# Find columns where all values are integers (i.e., decimal part is 0)
integer_float_columns = [col for col in float_columns if (fies_df[col] == fies_df[col].astype(int))

# Check if there are any integer float columns
if not integer_float_columns:
    print("No float columns with all values having a decimal part of 0.")
else:
    # Print the columns that meet the criteria and their data
    for col in integer_float_columns:
        print(f"Column: {col}")
        print(fies_df[col].to_string(index=False))  # Printing without the index for better readabi
        print()  # Add a blank line for separation
```

```
No float columns with all values having a decimal part of 0.
```

Feature Engineering

Household Classification based on Income

This article by The Philippine Star cited the Philippine Institute for Development Studies (PIDA), to which this organization used the PSA's FIES data to classify seven income groups in the Philippines with the following

```
- poor: with per capita incomes less than the official poverty threshold
- low (but not poor): with per capita incomes between the poverty line and twice
the poverty line
- lower middle: with per capita incomes between twice the poverty line and four
times the poverty line
- middle middle: with per capita incomes between four times the poverty line and
seven times the poverty line
- upper middle: with per capita incomes between seven times the poverty line and
12 times the poverty line
- upper middle (but not rich): with per capita incomes between 12 times the
poverty line and 20 times the poverty line
- rich: with per capita incomes at least equal to 20 times the poverty line
```

We try to make this feature in the dataset to potentially have a variable for machine learning.

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

FILIPINO INCOME GROUPS *and* HOW MUCH THEY'RE EARNING

INCOME GROUP	RANGE OF MONTHLY FAMILY INCOMES (for a family size of five members) IN 2021 PRICES
POOR	Less than P12,030 per month
LOW INCOME (BUT NOT POOR)	Between P12,030 to P24,060 per month
LOWER MIDDLE INCOME	Between P24,060 to P48,120 per month
MIDDLE MIDDLE CLASS	Between P48,120 to P84,210 per month
UPPER MIDDLE INCOME	Between P84,210 to P144,360 per month
UPPER INCOME (BUT NOT RICH)	Between P144,360 to P240,600
RICH	At least P240,600

l!fe
THE PHILIPPINE STAR

References:

1. https://philstarlife.com/news-and-views/847218-how-much-filipino-income-groups-earning?page=2

In [53]:
```
# We first derive the monthly income from the total household income
fies_df['monthly_income'] = fies_df['total_household_income'] / 12
print(fies_df['monthly_income'].head(5))
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
0    50589.166667
1    34331.666667
2    68184.333333
3    21544.833333
4    45893.666667
Name: monthly_income, dtype: float64
```

In [54]:
```python
def classify_income_group(income):
    if income < 12030:
        return 'Poor'
    elif income <= 24060:
        return 'Low Income (but not poor)'
    elif income <= 48120:
        return 'Lower Middle Income'
    elif income <= 84120:
        return 'Middle Middle Class'
    elif income <= 144360:
        return 'Upper Middle Income'
    elif income < 240600:
        return 'Upper Income (but not rich)'
    elif income >= 240600:
        return 'Rich'

# Apply
fies_df['income_group'] = fies_df['monthly_income'].apply(classify_income_group)

# Check
print(fies_df['income_group'].value_counts())
```

```
income_group
Low Income (but not poor)      65309
Lower Middle Income            43182
Poor                           28379
Middle Middle Class            13800
Upper Middle Income             3804
Upper Income (but not rich)      805
Rich                             257
Name: count, dtype: int64
```

Income-Based Features

In [55]:
```python
income_columns = ['regular_salaries_wages',
                  'seasonal_salaries_wages',
                  'total_salaries_wages',
                  'net_crop_fruit_share',
                  'cash_receipts_abroad',
                  'cash_receipts_domestic',
                  'non_agri_land_rentals',
                  'pension_retirement_benefits',
                  'dividends_from_investment',
                  'other_income_nec',
                  'family_sustenance_activities',
                  'total_gifts_received',
                  'income_crop_farming',
                  'income_livestock_poultry',
                  'income_fishing',
                  'income_forestry_hunting',
                  'income_wholesale_retail',
                  'income_manufacturing',
                  'income_transport_storage',
                  'entrepreneurial_activities_nec',
                  'entrepreneurial_activities_nec_1',
                  'entrepreneurial_activities_nec_2',
                  'other_receipts',
]
```

A. Income Diversity

```
This variable measures the number of income sources a household has, reflecting
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
their income diversification. A household with multiple income sources might be
more financially resilient, as it’s not entirely dependent on one source of
income.
```

B. Income Stability

```
It measures the ratio of regular to seasonal wages, assessing how consistent a
household's income is. A higher ratio suggests a more stable income source. INF in
this case
means that the household is stable with regular wages
```

C. Income Shares

```
These variables measure the contribution of various cash flows to the overall
income of the household, ranging from salaries, cash receipts, entrep. activities,
and more.
```

In [56]:

```
# Income Diversity
fies_df['income_diversity'] = fies_df[income_columns].notnull().sum(axis=1)

# Income Stability
fies_df['income_stability'] = np.where(fies_df['seasonal_salaries_wages'] == 0,
                                       float('inf'),  # flag for infinite stability
                                       fies_df['regular_salaries_wages'] / fies_df['seasonal_

# Income Shares
fies_df['income_salary_share'] = (fies_df['total_salaries_wages'] / fies_df['total_household_income
fies_df['income_cash_receipts_share'] = ((fies_df['cash_receipts_abroad'] + fies_df['cash_receipts_
fies_df['income_entrepreneurial_share'] = (fies_df['total_income_entrepreneurial_activities'] / fie
fies_df['income_rent_share'] = (fies_df['imputed_house_rental_value'] / fies_df['total_household_in
fies_df['income_gifts_share'] = (fies_df['total_gifts_received'] / fies_df['total_household_income'
fies_df['income_family_and_pension_share'] = ((fies_df['family_sustenance_activities'] + fies_df['p
fies_df['other_income_and_investments_share'] = ((fies_df['other_income_nec'] + fies_df['dividends_
```

Expenditure-Based Features

A. Food to Non-Food Ratio

```
This ratio compares spending on food versus non-food items. A higher ratio
suggests that the household allocates more of its budget to food, which may
indicate lower-income status, where more income is used for basic needs.
```

B. Expenditure Shares

```
Same gist as Income Shares, but for expenditure
```

In [57]:

```
# Food to Non-Food Ratio
fies_df['expenditure_food_to_nonfood_ratio'] = fies_df['household_food_expenditure'] / fies_df['tot

# Expenditure Shares
fies_df['expenditure_protein_share'] = ((fies_df['expenditure_meat_preparations'] +
                                         fies_df['expenditure_fish_marine_products'] +
                                         fies_df['expenditure_dairy_eggs']) / fies_df['total_househ

fies_df['expenditure_carbohydrates_share'] = ((fies_df['expenditure_cereal_preparations'] +
                                               fies_df['expenditure_sugar_jam_honey']) / fies_df['t

fies_df['expenditure_fruits_and_veggies_share'] = ((fies_df['expenditure_fruits_vegetables'] + fies
                                                    + fies_df['expenditure_fruit_vegetable_juices']

fies_df['expenditure_other_foods_share'] = ((fies_df['expenditure_oils_fats'] + fies_df['expenditur
                                             fies_df['expenditure_coffee_cocoa_tea'] + fies_df['exp
                                             fies_df['expenditure_cocoa'] + fies_df['expenditure_so
                                             fies_df['expenditure_non_alcoholic_beverages'] +
                                             fies_df['main_water_supply_second_visit']) / fies_df['
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
fies_df['expenditure_food_outside_home_share'] = (fies_df['food_consumed_outside_home'] / fies_df['

fies_df['expenditure_essential_goods_and_services'] = ((fies_df['expenditure_services_primary_goods
                                                        fies_df['expenditure_alcohol_production_ser
                                                        fies_df['expenditure_housing_water'] +
                                                        fies_df['expenditure_furnishings_household_
                                                        fies_df['expenditure_transportation'] + fie
                                                        fies_df['total_household_expenditures']) *

fies_df['expenditure_discretionary_goods_and_services'] = ((fies_df['expenditure_alcoholic_beverage
                                                            fies_df['expenditure_tobacco'] + fies_d
                                                            fies_df['expenditure_recreation_culture
                                                            fies_df['expenditure_insurance'] + fies
                                                            fies_df['expenditure_durable_furniture'
                                                            fies_df['other_expenditure'] + fies_df[
                                                            / fies_df['total_household_expenditures
```

Economic Stability and Vulnerability

A. Income to Expenditure Ratio

```
This measures whether a household's income is sufficient to cover its
expenditures. A ratio above 1 indicates that the household is earning more than it
spends, suggesting potential savings, while a ratio below 1 implies financial
strain.
```

B. Potential Savings

```
The difference between total household income and total household expenditures.
This variable suggests the possible value for savings relative to the income
```

C. Potential Debt

```
The difference between total household expenditures and total household income.
This variable suggests the possible value for debt if expenditures is larger than
income
```

D. Economically Vulnerable

```
This variable is a flag for households which have higher expenditure than income.
```

E. Below Poverty Line

```
This variable is a flag for household below the poverty line as per the PSA's
report on Poverty in 2023.
```

References

1. https://www.psa.gov.ph/statistics/poverty

In [58]:
```
fies_df['income_to_expenditure_ratio'] = fies_df['total_household_income'] / fies_df['total_househo

fies_df['potential_savings'] = fies_df['total_household_income'] - fies_df['total_household_expendi

fies_df['potential_debt'] = fies_df['total_household_expenditures'] - fies_df['total_household_inco
fies_df['potential_debt'] = fies_df['potential_debt'].apply(lambda x: x if x > 0 else 0)  # Only po

fies_df['is_economically_vulnerable'] = fies_df['total_household_expenditures'] > fies_df['total_ho

POVERTY_THRESHOLD_FOR_PH = 13873
fies_df['is_below_poverty_line'] = fies_df['total_household_income'] < POVERTY_THRESHOLD_FOR_PH
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Consumption Patterns and Living Standards

A. Protein to Carbohydrate Ratio

```
This ratio compares household spending on protein-rich foods (meat, fish, dairy)
to carbohydrate-rich foods (cereals, sugar). It helps understand dietary choices,
with higher values indicating a more protein-heavy diet, which can reflect higher
income or health-conscious behavior.
```

B. Processed to Fresh Food Ratio

```
This compares spending on processed food (e.g., soft drinks, oils) to fresh food
(e.g., fruits, vegetables, fish). Higher values suggest a diet that relies more on
processed items, which could indicate convenience or economic constraints.
```

C. Non-Essential Food Ratio

```
This variable reflects how much a household spends on non-essential food items
like soft drinks, alcohol, and coffee. Higher values indicate discretionary
spending, suggesting more disposable income.
```

D. Education Spending Ratio

```
These show the proportion of household expenditure on education. High spending on
education may indicate investment in long-term opportunities.
```

E. Health Spending Ratio

```
These show the proportion of household expenditure on health. High spending on
health can reflect the household's vulnerability to medical expenses or priority
on health.
```

F. Tobacco / Alcohol Spending Ratio

```
These show the proportion of household expenditure on tobacco and alcohol. High
spending on these items may reflect some potential problems in terms of vices like
drinking and smoking
```

In [59]:

```
protein_expenditure = ['expenditure_meat_preparations', 'expenditure_fish_marine_products', 'expend
carbs_expenditure = ['expenditure_cereal_preparations', 'expenditure_sugar_jam_honey']
fies_df['protein_to_carbohydrate_spending_ratio'] = fies_df[protein_expenditure].sum(axis=1) / fies

processed_food_expenditure = ['expenditure_sugar_jam_honey', 'expenditure_oils_fats', 'expenditure_

fresh_food_expenditure = ['expenditure_fruits_vegetables', 'expenditure_fish_marine_products', 'exp

fies_df['processed_to_fresh_food_ratio'] = fies_df[processed_food_expenditure].sum(axis=1) / fies_d

non_essential_food_expenditure = ['expenditure_softdrinks', 'expenditure_coffee_cocoa_tea', 'expend

fies_df['non_essential_food_ratio'] = fies_df[non_essential_food_expenditure].sum(axis=1) / fies_df

fies_df['education_spending_ratio'] = fies_df['expenditure_education'] / fies_df['total_household_e

fies_df['health_spending_ratio'] = fies_df['expenditure_health'] / fies_df['total_household_expendi

fies_df['tobacco_alcohol_ratio'] = (fies_df['expenditure_tobacco'] + fies_df['expenditure_alcoholic
```

Summary Statistics

We calculate the summary statistics for each income and expenditure columns.

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

In [60]:
```
income_columns = ['regular_salaries_wages',
                  'seasonal_salaries_wages',
                  'total_salaries_wages',
                  'net_crop_fruit_share',
                  'cash_receipts_abroad',
                  'cash_receipts_domestic',
                  'non_agri_land_rentals',
                  'pension_retirement_benefits',
                  'dividends_from_investment',
                  'other_income_nec',
                  'family_sustenance_activities',
                  'total_gifts_received',
                  'income_crop_farming',
                  'income_livestock_poultry',
                  'income_fishing',
                  'income_forestry_hunting',
                  'income_wholesale_retail',
                  'income_manufacturing',
                  'income_transport_storage',
                  'entrepreneurial_activities_nec',
                  'entrepreneurial_activities_nec_1',
                  'entrepreneurial_activities_nec_2',
                  'other_receipts',
                  'total_receipts',
                  'total_income_entrepreneurial_activities',
                  'total_household_income',
]

expenditure_columns = ['losses_from_entrepreneurial_activities',
                       'expenditure_cereal_preparations',
                       'expenditure_meat_preparations',
                       'expenditure_fish_marine_products',
                       'expenditure_dairy_eggs',
                       'expenditure_oils_fats',
                       'expenditure_fruits_vegetables',
                       'expenditure_vegetables',
                       'expenditure_sugar_jam_honey',
                       'expenditure_other_food',
                       'expenditure_fruit_vegetable_juices',
                       'main_water_supply_second_visit',
                       'expenditure_coffee_cocoa_tea',
                       'expenditure_tea',
                       'expenditure_cocoa',
                       'expenditure_softdrinks',
                       'expenditure_non_alcoholic_beverages',
                       'total_food_consumed_home',
                       'food_consumed_outside_home',
                       'expenditure_alcoholic_beverages',
                       'expenditure_tobacco',
                       'expenditure_other_vegetables',
                       'expenditure_services_primary_goods',
                       'expenditure_alcohol_production_services',
                       'household_food_expenditure',
                       'expenditure_clothing_footwear',
                       'expenditure_housing_water',
                       'expenditure_furnishings_household_maintenance',
                       'expenditure_health',
                       'expenditure_transportation',
                       'expenditure_communication',
                       'expenditure_recreation_culture',
                       'expenditure_education',
                       'expenditure_insurance',
                       'expenditure_miscellaneous_goods_services',
                       'expenditure_durable_furniture',
                       'expenditure_special_family_occasion',
                       'other_expenditure',
                       'other_disbursements',
                       'expenditure_accommodation_services',
                       'actual_house_rent',
                       'total_non_food_expenditure',
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
#                        'total_household_expenditures',
#                        'total_household_disbursements',
 ]

id_columns = ['rdmd_id',
              'region',
              'province',
              'household_id',
              'recoded_province',
              'region_with_nir'
]

misc_columns = ['family_size',
                'imputed_house_rental_value',
                'imputed_housing_benefit_rental_value',
                'house_rent_rental_value',
                'psu_recode',
                'raising_factor',
                'final_population_weights',
                'urban_rural',
                'per_capita_income',
                'national_per_capita_income',
                'regional_per_capita_income',
                'per_capita_income_decile_province',
                'provincial_per_capita_income_decile',
                'per_capita_income_decile_region_nir',
]
```

For reference only:

total food consumed at home:

```
    'expenditure_cereal_preparations',
    'expenditure_meat_preparations',
    'expenditure_fish_marine_products',
    'expenditure_dairy_eggs',
    'expenditure_oils_fats',
    'expenditure_fruits_vegetables',
    'expenditure_vegetables',
    'expenditure_sugar_jam_honey',
    'expenditure_other_food',
    'expenditure_fruit_vegetable_juices',
    'main_water_supply_second_visit',
    'expenditure_coffee_cocoa_tea',
    'expenditure_tea',
    'expenditure_cocoa',
    'expenditure_softdrinks',
    'expenditure_non_alcoholic_beverages',
```

Total household Income:

In [61]:
```
# Summary Statistics for Income (Mean, Median, Mode, Standard Deviation)

income_summary = fies_df[income_columns].describe()
print(income_summary)
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
       regular_salaries_wages  seasonal_salaries_wages  total_salaries_wages  \
count            1.555360e+05             1.555360e+05          1.555360e+05
mean             1.509929e+05             2.874662e+04          1.797396e+05
std              2.398292e+05             6.979412e+04          2.511435e+05
min              0.000000e+00             0.000000e+00          0.000000e+00
25%              0.000000e+00             0.000000e+00          3.000000e+04
50%              7.239600e+04             0.000000e+00          1.180000e+05
75%              2.045000e+05             4.250000e+04          2.350000e+05
max              1.547640e+07             1.700000e+07          3.247640e+07

       net_crop_fruit_share  cash_receipts_abroad  cash_receipts_domestic  \
count          1.555360e+05          1.555360e+05            1.555360e+05
mean           1.716062e+03          2.385233e+04            1.635780e+04
std            1.250161e+04          8.447856e+04            2.968692e+04
min            0.000000e+00          0.000000e+00            0.000000e+00
25%            0.000000e+00          0.000000e+00            0.000000e+00
50%            0.000000e+00          0.000000e+00            6.400000e+03
75%            0.000000e+00          1.500000e+03            2.120000e+04
max            1.200000e+06          4.500000e+06            1.700000e+06

       non_agri_land_rentals  pension_retirement_benefits  \
count           1.555360e+05                 1.555360e+05
mean            2.137240e+03                 8.887319e+03
std             2.519769e+04                 6.075362e+04
min             0.000000e+00                 0.000000e+00
25%             0.000000e+00                 0.000000e+00
50%             0.000000e+00                 0.000000e+00
75%             0.000000e+00                 0.000000e+00
max             3.900000e+06                 5.100000e+06

       dividends_from_investment  other_income_nec  ...  \
count               1.555360e+05     155536.000000  ...
mean                9.061123e+02        612.385988  ...
std                 6.832362e+04       6900.809717  ...
min                 0.000000e+00          0.000000  ...
25%                 0.000000e+00          0.000000  ...
50%                 0.000000e+00          0.000000  ...
75%                 0.000000e+00          0.000000  ...
max                 2.000000e+07     800000.000000  ...

       income_wholesale_retail  income_manufacturing  \
count             1.555360e+05          1.555360e+05
mean              2.555829e+04          2.354684e+03
std               2.288451e+05          2.940363e+04
min              -2.950000e+04         -2.000000e+03
25%               0.000000e+00          0.000000e+00
50%               0.000000e+00          0.000000e+00
75%               0.000000e+00          0.000000e+00
max               7.766400e+07          3.000660e+06

       income_transport_storage  entrepreneurial_activities_nec  \
count              1.555360e+05                    1.555360e+05
mean               6.463486e+03                    5.804513e+03
std                3.485627e+04                    8.336695e+04
min               -1.660000e+04                   -4.877000e+04
25%                0.000000e+00                    0.000000e+00
50%                0.000000e+00                    0.000000e+00
75%                0.000000e+00                    0.000000e+00
max                3.235800e+06                    1.592320e+07

       entrepreneurial_activities_nec_1  entrepreneurial_activities_nec_2  \
count                      1.555360e+05                      1.555360e+05
mean                       3.670353e+02                      3.743294e+02
std                        1.697473e+04                      1.816846e+04
min                       -2.000000e+00                      0.000000e+00
25%                        0.000000e+00                      0.000000e+00
50%                        0.000000e+00                      0.000000e+00
75%                        0.000000e+00                      0.000000e+00
max                        4.469398e+06                      5.569280e+06
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
       other_receipts  total_receipts  \
count    1.555360e+05    1.555360e+05
mean     1.018184e+04    3.439300e+05
std      1.112867e+05    4.449579e+05
min      0.000000e+00    1.403000e+04
25%      0.000000e+00    1.679300e+05
50%      0.000000e+00    2.496095e+05
75%      1.500000e+03    4.006250e+05
max      2.644000e+07    8.666120e+07

       total_income_entrepreneurial_activities  total_household_income
count                             1.555360e+05            1.555360e+05
mean                              6.192258e+04            3.337481e+05
std                               2.696420e+05            4.093804e+05
min                               0.000000e+00            1.403000e+04
25%                               0.000000e+00            1.638390e+05
50%                               1.750000e+04            2.430000e+05
75%                               8.059000e+04            3.915030e+05
max                               8.094431e+07            8.246120e+07

[8 rows x 26 columns]
```

In [62]:

```
# Summary Statistics for Expenditure (Mean, Median, Mode, Standard Deviation)

expenditure_summary = fies_df[expenditure_columns].describe()
print(expenditure_summary)
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
       losses_from_entrepreneurial_activities  \
count                           155536.000000
mean                                 6.946058
std                                546.407497
min                                  0.000000
25%                                  0.000000
50%                                  0.000000
75%                                  0.000000
max                             136448.000000

       expenditure_cereal_preparations  expenditure_meat_preparations  \
count                    155536.000000                  155536.000000
mean                      27220.986942                   16213.612726
std                       12902.768897                   13795.044346
min                           0.000000                       0.000000
25%                       18279.000000                    6802.000000
50%                       25135.000000                   12480.000000
75%                       33953.000000                   21528.000000
max                      367182.000000                  362100.000000

       expenditure_fish_marine_products  expenditure_dairy_eggs  \
count                     155536.000000           155536.000000
mean                       14389.271158             6145.896146
std                        10377.031551             6327.668685
min                            0.000000                0.000000
25%                         7456.000000             2782.000000
50%                        11937.000000             4680.000000
75%                        18460.000000             7560.000000
max                       301740.000000           626024.000000

       expenditure_oils_fats  expenditure_fruits_vegetables  \
count          155536.000000                  155536.000000
mean             1722.389313                    4615.230797
std              1570.568516                    4269.061334
min                 0.000000                       0.000000
25%               960.000000                    1990.000000
50%              1430.000000                    3440.000000
75%              2090.000000                    5845.000000
max            240480.000000                  263530.000000

       expenditure_vegetables  expenditure_sugar_jam_honey  \
count           155536.000000                155536.000000
mean              7078.460560                  2145.303171
std               5583.559762                  2050.757852
min                  0.000000                     0.000000
25%               3795.000000                   983.000000
50%               5870.000000                  1621.000000
75%               8890.125000                  2704.250000
max             811295.000000                131445.000000

       expenditure_other_food  ...  expenditure_recreation_culture  \
count           155536.000000  ...                    1.555360e+05
mean              4133.526558  ...                    1.993140e+03
std               6183.479430  ...                    9.175600e+03
min                  0.000000  ...                    0.000000e+00
25%               1660.000000  ...                    5.000000e+01
50%               2587.000000  ...                    6.500000e+02
75%               4236.000000  ...                    1.700000e+03
max             405697.000000  ...                    2.011000e+06

       expenditure_education  expenditure_insurance  \
count           1.555360e+05          155536.000000
mean            8.035793e+03            6382.939782
std             2.410755e+04           17183.425748
min             0.000000e+00               0.000000
25%             0.000000e+00               0.000000
50%             5.000000e+02               0.000000
75%             5.300000e+03            6000.000000
max             1.800000e+06          906000.000000
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
       expenditure_miscellaneous_goods_services  \
count                              155536.000000
mean                                 9088.059999
std                                 10555.815845
min                                    64.000000
25%                                  4215.000000
50%                                  6690.000000
75%                                 10658.000000
max                                881775.000000

       expenditure_durable_furniture  expenditure_special_family_occasion  \
count                   1.555360e+05                        155536.000000
mean                    5.142340e+03                          6231.273821
std                     3.765973e+04                         13754.655359
min                     0.000000e+00                             0.000000
25%                     0.000000e+00                             0.000000
50%                     0.000000e+00                          2500.000000
75%                     0.000000e+00                          6800.000000
max                     4.000000e+06                        905000.000000

       other_expenditure  other_disbursements  \
count       1.555360e+05         1.555360e+05
mean        4.944876e+03         2.725642e+04
std         2.695376e+04         1.766135e+05
min         0.000000e+00         0.000000e+00
25%         0.000000e+00         0.000000e+00
50%         3.500000e+02         0.000000e+00
75%         2.600000e+03         1.260000e+04
max         3.769969e+06         3.704800e+07

       expenditure_accommodation_services  actual_house_rent
count                       155536.000000       1.555360e+05
mean                           221.874119       3.779537e+03
std                           2636.134137       1.819488e+04
min                              0.000000       0.000000e+00
25%                              0.000000       0.000000e+00
50%                              0.000000       0.000000e+00
75%                              0.000000       0.000000e+00
max                         288000.000000       2.100000e+06

[8 rows x 41 columns]
```

Data Visualization

In [63]:
```
expenditure_columns_1 = [
    'losses_from_entrepreneurial_activities',
    'expenditure_services_primary_goods',
    'expenditure_alcohol_production_services',
    'expenditure_clothing_footwear',
    'expenditure_housing_water',
    'expenditure_furnishings_household_maintenance',
    'expenditure_health',
    'expenditure_transportation',
    'expenditure_communication',
    'expenditure_recreation_culture',
    'expenditure_education',
    'expenditure_insurance',
    'expenditure_miscellaneous_goods_services',
    'expenditure_durable_furniture',
    'expenditure_special_family_occasion',
    'other_expenditure',
    'other_disbursements',
    'expenditure_accommodation_services',
    'actual_house_rent',
    'total_food_consumed_home',
    'food_consumed_outside_home',
    'expenditure_alcoholic_beverages',
    'expenditure_other_vegetables',
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
]

expenditure_labels = {
    'losses_from_entrepreneurial_activities': 'Losses from Entrepreneurial Activities',
    'expenditure_services_primary_goods': 'Expenditure of Services and Goods',
    'expenditure_alcohol_production_services': 'Expenditure of Alcohol and Production Services',
    'expenditure_clothing_footwear': 'Expenditure of Clothing and Footwear',
    'expenditure_housing_water': 'Expenditure of Housing and Water',
    'expenditure_furnishings_household_maintenance': 'Expenditure of Furnishings and Household Main
    'expenditure_health': 'Expenditure of Health',
    'expenditure_transportation': 'Expenditure of Transportation',
    'expenditure_communication': 'Expenditure of Communication',
    'expenditure_recreation_culture': 'Expenditure of Recreation and Culture',
    'expenditure_education': 'Expenditure of Education',
    'expenditure_insurance': 'Expenditure of Insurance',
    'expenditure_miscellaneous_goods_services': 'Expenditure of Miscellaneous Goods and Services',
    'expenditure_durable_furniture': 'Expenditure of Durable Furniture',
    'expenditure_special_family_occasion': 'Expenditure of Special Family Occasion',
    'other_expenditure': 'Other Expenditure',
    'other_disbursements': 'Other Disbursements',
    'expenditure_accommodation_services': 'Expenditure of Accommodation Services',
    'actual_house_rent': 'Actual House Rent',
    'total_food_consumed_home': 'Total Food Consumed at Home',
    'food_consumed_outside_home': 'Food Consumed Outside Home',
    'expenditure_alcoholic_beverages': 'Expenditure on Alcoholic Beverages',
    'expenditure_other_vegetables': 'Expenditure on Other Vegetables',

}
```

In [64]:

```
import matplotlib.pyplot as plt
# Step 1: Convert expenditure columns to numeric, coercing errors (non-numeric values will be set a
fies_df[expenditure_columns_1] = fies_df[expenditure_columns_1].apply(pd.to_numeric, errors='coerce

# Step 2: Calculate mean and total for each expenditure column
mean_expenditures = fies_df[expenditure_columns_1].mean().sort_values(ascending=True)
total_expenditures = fies_df[expenditure_columns_1].sum().sort_values(ascending=True)

# Replace column names with friendly labels
mean_expenditures_friendly = mean_expenditures.rename(index=expenditure_labels)
total_expenditures_friendly = total_expenditures.rename(index=expenditure_labels)

# Step 3: Plot the mean expenditures
plt.figure(figsize=(12, 6))
mean_expenditures_friendly.plot(kind='barh', color='skyblue')
plt.title('Mean Expenditure per Category')
plt.xlabel('Mean Expenditure')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()

# Step 4: Plot the total expenditures
plt.figure(figsize=(12, 6))
total_expenditures_friendly.plot(kind='barh', color='salmon')
plt.title('Total Expenditure per Category')
plt.xlabel('Total Expenditure')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Justification:

- We used a map to make the categories more readable.

Insights:

1. Targeted Marketing: Businesses can identify which expenditure categories (e.g., food, transportation, health) have the highest average spending or overall spending, allowing them to tailor their products or services to meet consumer demand in specific sectors.

2. Market Trends: By analyzing expenditure data over time, businesses can spot trends in consumer behavior, which can inform strategic planning and investment decisions.

3. Budget Planning: For policymakers and organizations, this data aids in budget allocation and financial planning, ensuring that funding aligns with actual consumer needs and priorities.

4. Competitive Advantage: Companies can gain insights into competitors' market positions by understanding where consumers are spending their money, enabling them to adjust their strategies accordingly.

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

In [65]:
```
region_mapping = {
    1: 'I',             # Region I - Ilocos
    2: 'II',            # Region II - Cagayan Valley
    3: 'III',           # Region III - Central Luzon
    4: 'IV-A',          # Region IV-A - CALABARZON
    5: 'V',             # Region V - Bicol
    6: 'VI',            # Region VI - Western Visayas
    7: 'VII',           # Region VII - Central Visayas
    8: 'VIII',          # Region VIII - Eastern Visayas
    9: 'IX',            # Region IX - Zamboanga Peninsula
    10: 'X',            # Region X - Northern Mindanao
    11: 'XI',           # Region XI - Davao
    12: 'XII',          # Region XII - SOCCSKSARGEN
    13: 'NCR',          # National Capital Region
    14: 'CAR',          # Cordillera Administrative Region
    15: 'ARMM',         # Bangsamoro Autonomous Region in Muslim Mindanao
    16: 'XIII',         # Region XIII - Caraga
    17: 'IV-B',          # Region IV-B - MIMAROPA
    18: 'XVIII'         # Region XVIII - Negros Island Region
}
```

In [66]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Assuming fies_df is already defined and renamed

# Calculate average house rent by region (keep using integer values)
average_rent_by_region = fies_df.groupby('region_with_nir')['house_rent_rental_value'].mean().reset

# Rename the region index using the mapping
average_rent_by_region_friendly = average_rent_by_region.rename(columns={'region_with_nir': 'region
average_rent_by_region_friendly['region_with_nir'] = average_rent_by_region_friendly['region_index'

# Sort values for better visualization
average_rent_by_region_friendly = average_rent_by_region_friendly.sort_values(by='house_rent_rental

# Set up the matplotlib figure
plt.figure(figsize=(12, 6))

# Create a bar plot
sns.barplot(x='region_with_nir', y='house_rent_rental_value', data=average_rent_by_region_friendly)

# Add titles and labels
plt.title('Average House Rent/Rental Value by Region', fontsize=16)
plt.xlabel('Region', fontsize=14)
plt.ylabel('Average Rental Value', fontsize=14)

# Show the plot
plt.show()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Justification:

- We used a map to display in the graph the regions number specifically instead of the original 0-17 values which can cause confusions as there are regions such as NCR that is not numbered like region 1, 2, and so on. The region names are based on the PSA's record of 'Summary of Changes made in the Philippines Standard Geographic Code Since 2001'. In this record, the code for each region is showed by the code's first two digits. For instance, Region 9 starting code is 09 followed by the more specific place (municipality / city).

Insights:

1. Investment Decisions: The average rental values provide critical insights for potential investors looking to purchase properties, as they highlight regions with higher or lower rental income potential. for example, NCR has the highest average of house rent which can mean the demand for a place to stay in NCR is higher compared to other regions.

2. Regional Differences: The visualization illustrates the disparities in housing rental values across different regions, aiding in identifying profitable investment areas.

In [67]:
```
# Step 1: Convert the relevant columns to numeric, forcing non-numeric values to NaN
fies_df['total_household_income'] = pd.to_numeric(fies_df['total_household_income'], errors='coerce
fies_df['total_household_disbursements'] = pd.to_numeric(fies_df['total_household_disbursements'],

# Step 2: Group the data by family size and calculate the mean for income and disbursements
grouped_data = fies_df.groupby('family_size').agg({
    'total_household_income': 'mean',
    'total_household_disbursements': 'mean'
}).reset_index()

# Step 3: Plot the grouped bar chart with thinner bars
plt.figure(figsize=(18, 6))
bar_width = 0.2  # Adjust this value for thinner bars

# Plot Total Income
plt.bar(grouped_data['family_size'] - bar_width/2, grouped_data['total_household_income'],
        width=bar_width, label='Total Income', color='skyblue')

# Plot Total Disbursements
plt.bar(grouped_data['family_size'] + bar_width/2, grouped_data['total_household_disbursements'],
        width=bar_width, label='Total Disbursements', color='salmon')
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
# Add labels and title
plt.xlabel('Family Size')
plt.ylabel('Amount (mean)')
plt.title('Mean Total Income vs Total Disbursements by Family Size')
plt.xticks(grouped_data['family_size'])
plt.legend()
plt.tight_layout()
plt.show()
```

Justification:

FSIZE_VS1	Average Family Size		
	1	0	1
	2	1.5	2
	3	2.5	3
	4	3.5	4
	5	4.5	5
	6	5.5	6
	7	6.5	7
	8	7.5	8
	9	8.5	9
	10 and ove	9.5	24

- Here are the meanings of the family size in the graph. Since there are .5 values in the family size. For now, we opted in the original value.

Insights:

- The visual comparison allows for quick identification of which family sizes manage to save more. The relationship between household income and disbursements across different family sizes helps businesses assess consumer purchasing power in various segments.

In [68]:

```
# Average Monthly Income Per Region

average_income_by_region = fies_df.groupby('region_with_nir')['monthly_income'].mean().reset_index(

# Rename the region index using the mapping
average_income_by_region_friendly = average_income_by_region.rename(columns={'region_with_nir': 're
average_income_by_region_friendly['region_with_nir'] = average_income_by_region_friendly['region_in

# Sort values for better visualization
average_income_by_region_friendly = average_income_by_region_friendly.sort_values(by='monthly_incom

# Set up the matplotlib figure
plt.figure(figsize=(12, 6))

# Create a bar plot
sns.barplot(x='region_with_nir', y='monthly_income', data=average_income_by_region_friendly)
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
plt.title('Average Monthly Income by Region', fontsize=16)
plt.xlabel('Region', fontsize=14)
plt.ylabel('Average Monthly Income', fontsize=14)

# Show the plot
plt.show()
```

Insights for Average Monthly Income Per Region

The graph above showcases the distribution of the average monthly income. NCR have the highest average monthly invome as this region is the central location of business and economic activities. Furthermore, the minimum wage is higher in comparison to the other regions so in average, it will provide a higher mean. NCR is also the home of the richer individuals which will help increase the mean. The second highest average income is from Region IV-A which is the Calabarzon region. The region is known as the industrial powerhouse of the Philippines. It is also in close proximity of Manila which provides an enormous economic opportunity. Land doesn't depreciated which means that in this case, the neighboring regions of NCR will also have their respective lands appreciate in value. As for BARMM, it was stated in an article by Inquirer that even with the rich natural endowment, the region is still poor due to poor governance, feudalism, clan fueds, weak infrastructure, land dispute and, unstable peace and order.

In [69]:

```
# Average Expenditure on Alcoholic Beverages and Tobacco by Family Size
family_expenditure = fies_df.groupby('family_size')[['expenditure_alcoholic_beverages', 'expenditur

# Calculate the sample size for each family size
family_counts = fies_df['family_size'].value_counts().reset_index()
family_counts.columns = ['family_size', 'count']

# Merge to get counts with expenditures
family_expenditure = family_expenditure.merge(family_counts, on='family_size')

# Filter family sizes with a sample size greater than 20
filtered_expenditure = family_expenditure[family_expenditure['count'] > 5000]

# Set the figure size
plt.figure(figsize=(10, 6))

# Plot the average data
filtered_expenditure.set_index('family_size')[['expenditure_alcoholic_beverages', 'expenditure_toba

# Add labels and title
plt.title('Average Expenditure on Alcoholic Beverages and Tobacco by Family Size', fontsize=16)
plt.xlabel('Family Size', fontsize=14)
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
plt.ylabel('Average Expenditure', fontsize=14)
plt.legend(loc='best', bbox_to_anchor=(1.05, 1), title="Expenditure Type")  # Legend outside the pl

# Show the plot
plt.tight_layout()
plt.show()
```

```
<Figure size 1000x600 with 0 Axes>
```

In [70]:
```
decile_expenditure = fies_df.groupby('per_capita_income_decile_region_nir')[['expenditure_alcoholic

# Set the figure size
plt.figure(figsize=(12, 8))

# Plot the data as a vertical bar chart
ax = decile_expenditure.set_index('per_capita_income_decile_region_nir')[['expenditure_alcoholic_be

# Customize the y-axis labels
ax.set_yticks(ax.get_yticks())  # Get current y ticks
ax.set_yticklabels([f'{int(tick):,}' for tick in ax.get_yticks()])  # Format labels with commas

# Add labels and title
plt.title('Average Expenditure on Alcoholic Beverages and Tobacco by Per Capita Income Decile', fon
plt.xlabel('Per Capita Income Decile', fontsize=14)
plt.ylabel('Average Expenditure', fontsize=14)
plt.legend(loc='best', bbox_to_anchor=(1.05, 1), title="Expenditure Type")  # Legend outside the pl

# Show the plot
plt.tight_layout()
plt.show()
```

```
<Figure size 1200x800 with 0 Axes>
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

In [71]:

```
# Average Expenditure on Alcoholic Beverages and Tobacco by Urban/Rural Areas
fies_df['urban_rural'] = fies_df['urban_rural'].map({1: 'Urban', 2: 'Rural'})

# Group by urban_rural and calculate the average expenditure for alcoholic beverages and tobacco
urban_rural_expenditure = fies_df.groupby('urban_rural')[[
    'expenditure_alcoholic_beverages',
    'expenditure_tobacco'
]].mean().reset_index()

# Set the urban_rural as the index for proper labeling
urban_rural_expenditure.set_index('urban_rural', inplace=True)

# Plot the data
urban_rural_expenditure.plot(kind='bar', figsize=(10, 6))

# Add labels and title
plt.title('Average Expenditure on Alcoholic Beverages and Tobacco by Urban/Rural Areas', fontsize=1
plt.xlabel('Urban/Rural', fontsize=14)
plt.ylabel('Average Expenditure', fontsize=14)
plt.legend(loc='best', bbox_to_anchor=(1.05, 1), title="Expenditure Type")  # Legend outside the pl

# Show the plot
plt.tight_layout()
plt.show()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Insights for Average Expenditure on Alcoholic Beverages and Tobacco by Family Size and by Urban/Rural Areas

By Family Size, it was limited to family sizes >5000 as it represents a very small percentage and it may be skewed so it won't show accurate and proper distribution. In this graph, we see an increasing number, which could point to having more family members, leads to someone embracing more and more vices. But it could also just represent that more family members are smoking tobacco and consuming alcohol.

By per capita income decile. We can see a big difference from the 10th decile and 9th decile for the alcoholic beverages as it could mean that with having more disposable income, they have a better access to more expensive and more refined alcohols as compared to the rest. As for the tobacco, it shows a normal distribution. Nicotine provides dopamine to the body. For the low income earners, they need to save as tobacco isn't really a necessity but is still somewhat is, as for some, it will be their leisure time. and as we go up the income decile, we see an increase. They have enough income to support their vices, which is why they have an increased consumption. And for the 10th decile, maybe smoking tobacco is still a leisurely time for them but with even more disposable income, they would rather opt to spend their time elswhere. Or we can dig deeper to the psyche of the others, where they would prefer to stay healthy to enjoy their wealth and whatnot.

By Urban/Rural Areas. By 2020, 54% of the population lives in urban barangays as stated by the PSA. And it is just natural that a higher population with higher monthly income, they will have an easier access to there vices.

In [72]:
```
# Total Agricultural Income
import matplotlib.ticker as mticker

# List of agricultural income columns
agri_columns = [
    'income_crop_farming',
    'income_livestock_poultry',
    'income_fishing',
    'income_forestry_hunting',
]

# Group by region and calculate the average income for each agricultural activity
average_agri_income_by_region = fies_df.groupby('region')[agri_columns].mean().reset_index()

# Renaming and mapping region names
average_agri_income_by_region_friendly = average_agri_income_by_region.rename(columns={'region': 'r
average_agri_income_by_region_friendly['region'] = average_agri_income_by_region_friendly['region_i

                                                                                    void showing it in the legend
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
average_agri_income_by_region_friendly = average_agri_income_by_region_friendly.drop(columns='regio

# Set the figure size larger
plt.figure(figsize=(12, 8))

# Plot average agricultural income by region
average_agri_income_by_region_friendly.set_index('region').plot(kind='bar', figsize=(12, 8))

# Add labels and title
plt.title('Average Agricultural Income by Region', fontsize=16)
plt.xlabel('Region', fontsize=14)
plt.ylabel('Average Income', fontsize=14)
plt.xticks(rotation=45, ha='right')  # Rotate x labels for better readability

# Customize y-axis labels to show a suitable range
plt.gca().yaxis.set_major_formatter(mticker.StrMethodFormatter('{x:,.0f}'))  # Format y-ticks with
plt.gca().yaxis.set_major_locator(mticker.MaxNLocator(integer=True))  # Ensure y-ticks are integers

plt.legend(title="Agricultural Activities", fontsize=12)  # Legend without region_index

# Show the plot
plt.tight_layout()
plt.show()
```

```
<Figure size 1200x800 with 0 Axes>
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Insights for Average Agricultural Income

Here we Can clearly see what each region's strength is. As stated earlier, ARMM is the poorest region but to some that engages in agricultural activities, we can see how proficient they are. This is also due to some certain household who provided such a high number of income. Region II is also home to one of the biggest crop farming industry here in the Philippines which is why when natural calamities strike, where they often strike the northen part of the Philippines, the northern regions are greatly affected. However, there are possible misrepresentation as there are some regions that are also big in the fishing industry, but we are looking at the average per household, and some entries will provide data to skew the graph. On the other hand, NCR is clearly seen to have close to zero contribution. As mentioned earlier, NCR is the central location of business activities and that its geographical location won't provide much help to the said agricultural activities.

Insights for Average House Rent/Rental Value by Family Size

In [73]:
```
import matplotlib.ticker as ticker

average_expenditures = fies_df.groupby('per_capita_income_decile_region_nir')['total_household_expe
average_income = fies_df.groupby('per_capita_income_decile_region_nir')['total_household_income'].m

# Merge the two DataFrames on 'per_capita_income_decile_region_nir'
combined_df = pd.merge(average_expenditures, average_income, on='per_capita_income_decile_region_ni

# Calculate the average difference (income - expenditures)
combined_df['average_expenditure_income_difference'] = combined_df['total_household_income'] - comb

# Set the figure size
plt.figure(figsize=(12, 8))

# Plot the average difference as a bar chart
ax = combined_df.set_index('per_capita_income_decile_region_nir')['average_expenditure_income_diffe

# Add labels and title
plt.title('Average Difference between Total Household Income and Expenditures by Per Capita Income
plt.xlabel('Per Capita Income Decile', fontsize=14)
plt.ylabel('Average Income - Expenditure Difference', fontsize=14)

# Customize y-axis to show more labels
ax.yaxis.set_major_locator(ticker.MaxNLocator(nbins=15))  # Set the number of y-ticks

# Show the plot
plt.tight_layout()
plt.show()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Insights for Average Difference between Total Household Income and Expenditures by Per Capita Income Decile

In other words, Savings. The most obvious thing that could be said in accordance to this graph is that top percentile of the rich are just too rich. This graph is in an annual timeframe. This graph also represents all of the households income after all the expenditures are deducted. What we're surprised with is that even the lowest decile still have some savings. We all know that a number of Pilipinos are living paycheck to paycheck, some don't even have enough to go by some of the days. So the data isn't as accurate is it would have seemed or that there is a low representation of the poor. We also had the chance to go through the survey and some of its contents won't be understandable to the uneducated and less fortunate thus a justification for the previous statement. As for the average Pilipino, PSA stated that P75k is what the normal Pilipino would have in his/her savings, which is accurate for the data. What it says is that the average Pilipino will slave away till the end of their times because having 75k annual savings is just not enough to retire. In 20 years time, without any inflation adjustments, 75k would only amount to 1.5 million. Taking into account any kind of emergency, that 1.5M would easily vanish.

In [74]:
```
import matplotlib.pyplot as plt

# Calculate mean values for food consumed at home and outside
mean_food_home = fies_df['household_food_expenditure'].mean() # Mean of means of all home food colu
mean_food_outside = fies_df['food_consumed_outside_home'].mean() # Mean of means of all outside foo


# Create a pie chart comparing mean food consumed at home vs outside
labels = ['Food Consumed at Home', 'Food Consumed Outside']
sizes = [mean_food_home, mean_food_outside]
colors = ['seagreen', 'coral']

plt.figure(figsize=(7, 7))
plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', startangle=90, counterclock=False)

# Add a title
plt.title('Mean Expenditure: Food Consumed at Home vs Outside')

# Show the plot
plt.tight_layout()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Insights:

- Dominance of Home-Cooked Meals: The chart shows that 89.7% of food expenditure is directed toward food consumed at home, highlighting a strong preference for home-prepared meals.
- Lower Spending on Dining Out: Only 10.3% of the food budget is spent on food consumed outside the home, indicating that dining out is relatively less common or prioritized in the given household data.
- Insight: This could suggest that households are more likely to allocate their food budget toward meals prepared at home due to cost-saving reasons or limited access to dining out options.

In [75]:

```
# Group 1: With income from salaries and wages
with_income = fies_df[fies_df['total_salaries_wages'] > 0]

# Group 2: No income from salaries and wages
no_income = fies_df[fies_df['total_salaries_wages'] == 0]

# Summing expenditures for households with income
with_income_summary = with_income[expenditure_columns].sum()

# Summing expenditures for households without income
no_income_summary = no_income[expenditure_columns].sum()

import numpy as np

# Create index for the expenditure categories
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
ind = np.arange(len(expenditure_columns))

# Bar width
width = 0.5

# Plotting
fig, ax = plt.subplots(figsize=(12, 8))

# Bar for households with income
ax.bar(ind - width/2, with_income_summary, width, label='With Income')

# Bar for households without income
ax.bar(ind + width/2, no_income_summary, width, label='No Income')

# Add labels, title, and custom ticks for x-axis
ax.set_xlabel('Expenditure Categories')
ax.set_ylabel('Total Expenditure')
ax.set_title('Expenditure Breakdown: With vs No Income from Salaries and Wages')
ax.set_xticks(ind)
ax.set_xticklabels([col.replace('expenditure_', '').replace('_', ' ').title() for col in expenditur
ax.legend()
plt.tight_layout()
plt.show()
```

Insights:

- Food Expenditures: Households with income spend significantly more on food compared to those without income, where food expenses are minimal.
- Non-Food Essentials: Families with income also spend more on non-food items like housing, transportation, and health. Households without income struggle to afford these essentials.
- Disparity: There is a clear divide in spending between households with and without income, with food and non-food expenses being drastically lower for families without income.

In [76]:
```
import matplotlib.pyplot as plt
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```python
# Create a temporary DataFrame to avoid modifying the original fies_df
temp_df = fies_df.copy()

# Sum the food and non-food expenditures across the entire dataset
total_food_expenditure = temp_df['household_food_expenditure'].sum()
total_non_food_expenditure = temp_df['total_non_food_expenditure'].sum()

# Prepare data for the pie chart
labels = ['Food Expenditure', 'Non-Food Expenditure']
sizes = [total_food_expenditure, total_non_food_expenditure]
colors = ['blue', 'orange']

# Plotting pie chart for food vs non-food expenditure
plt.figure(figsize=(8, 8))
plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', startangle=90)

# Adding title
plt.title('Food vs Non-Food Expenditures (Overall)')

# Show the plot
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
plt.tight_layout()
plt.show()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Insights:

- Non-Food Expenditures Dominate (58.2%): Families spend more on non-food essentials, indicating a higher financial strain in these areas.
- Food Expenditures (41.8%): Food remains a significant portion of household budgets, highlighting its importance for overall well-being.
- Food security and affordability remain critical concerns.
- Non-food expenditure insights can guide strategies in housing, healthcare, and transportation support.
- Households can focus on reducing non-food spending.
- Businesses targeting these sectors can adjust offerings based on consumer priorities.

In [77]:
```
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

# Calculate the correlation matrix
corr_matrix = fies_df[expenditure_columns].corr()

# Plotting the heatmap with better label handling and larger cells
plt.figure(figsize=(24, 18))  # Increase figure size for larger cells
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewidths=1.2, vmin=-1, vmax=1, c

# Improve label readability with adjustments
plt.title('Correlation Heatmap Between Expenditure Categories', fontsize=24, pad=20)

# Adjust x-axis labels
plt.xticks(ticks=np.arange(len(expenditure_columns)) + 0.5,
           labels=[col.replace('expenditure_', '').replace('_', ' ').title() for col in expenditure
           rotation=45, ha='right', fontsize=14)

# Adjust y-axis labels
plt.yticks(ticks=np.arange(len(expenditure_columns)) + 0.5,
           labels=[col.replace('expenditure_', '').replace('_', ' ').title() for col in expenditure
           rotation=0, fontsize=14)

# Adjust layout to ensure everything fits and labels are readable
plt.tight_layout()
plt.show()
```

Insights

Strong Correlations:

- Household Food Expenditure and Total Food Consumed at Home have a high positive correlation (0.97), suggesting that food consumed at home is a major driver of household food expenditures.
- Clothing Footwear and Furnishings Household Maintenance also show a notable positive correlation (0.87). Low/Negative Correlations:

Low Correlations:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

- Losses From Entrepreneurial Activities and most other categories show very low or negative correlations, indicating minimal spending overlap in these areas.
- Accommodation Services and Other Expenditures have a low positive correlation (0.13), suggesting some but not strong financial link.

In [78]:
```
non_food_expenditure_columns = [
    'losses_from_entrepreneurial_activities',
    'expenditure_services_primary_goods',
    'expenditure_alcohol_production_services',
    'household_food_expenditure',
    'expenditure_clothing_footwear',
    'expenditure_housing_water',
    'expenditure_furnishings_household_maintenance',
    'expenditure_health',
    'expenditure_transportation',
    'expenditure_communication',
    'expenditure_recreation_culture',
    'expenditure_education',
    'expenditure_insurance',
    'expenditure_miscellaneous_goods_services',
    'expenditure_durable_furniture',
    'expenditure_special_family_occasion',
    'other_expenditure',
    'other_disbursements',
    'expenditure_accommodation_services',
    'actual_house_rent',
]

# Calculate the correlation matrix
corr_matrix = fies_df[non_food_expenditure_columns].corr()

# Plotting the heatmap with better label handling and larger cells
plt.figure(figsize=(24, 18))  # Increase figure size for larger cells
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewidths=1.2, vmin=-1, vmax=1, c

# Improve label readability with adjustments
plt.title('Correlation Heatmap Between Non-Food Expenditure Categories', fontsize=24, pad=20)

# Adjust x-axis labels
plt.xticks(ticks=np.arange(len(non_food_expenditure_columns)) + 0.5,
           labels=[col.replace('expenditure_', '').replace('_', ' ').title() for col in non_food_ex
           rotation=45, ha='right', fontsize=14)

# Adjust y-axis labels
plt.yticks(ticks=np.arange(len(non_food_expenditure_columns)) + 0.5,
           labels=[col.replace('expenditure_', '').replace('_', ' ').title() for col in non_food_ex
           rotation=0, fontsize=14)

# Adjust layout to ensure everything fits and labels are readable
plt.tight_layout()
plt.show()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Insights:

Strong Correlations:

- "Household Food Expenditure" is strongly correlated with "Clothing Footwear" (0.45) and "Housing Water" (0.47).
- "Furnishings Household Maintenance" has a moderate correlation with "Housing Water" (0.48).
- "Communication" and "Recreation Culture" show a correlation of 0.55.

Weak Correlations:

- "Losses from Entrepreneurial Activities" shows almost no correlation with other categories, all values near 0.
- Most categories like "Alcohol Production Services" and "Special Family Occasion" exhibit weak correlations.

Outliers:

- "Actual House Rent" and "Accommodation Services" have a very high correlation (0.99), suggesting a close relationship between these expenditures.
- Overall, most expenditures are weakly to moderately correlated, with few strong relationships evident.

Preliminary Machine Learning Model

We plan to make two models to use the FIES dataset.

Classification Neural Network based on Income Group

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

We plan to make a Neural Network to classify the income group of each household. Neural Networks work well with high-dimensional data, most especially with this one with having many expenditure features.

Classification Artificial Neural Network

- **Use case**: Using Expenditures, the ANN will try to classify households to the seven income groups from the Feature Engineering section.
- **Why**: ANN's handle complex, high-dimensional data well, as well as having enough complexity to be scalable for more data later on.
- **Why Expenditures**: Using income results in potential data leakage, so using variables like region, expenditures and other factors will challenge the model to find any interesting insights and predict well the household income group.

Clustering for Household Segmentation

Group households into clusters based on similar income patterns and characteristics without labels. This can visualize the demographic of the FIES dataset.

DBSCAN (Density-Based Spatial Clustering)

- **Use case**: DBSCAN is good for irregularly shaped data, where households can have similar incomes but have different expenditures. It is also more complex than the usual K-Means and Hierarchical Clustering.
- **Why**: DBSCAN can identify clusters of varying densities, including outliers, and doesn't require specifying the number of clusters.

Standard Process

1. **Data Preprocessing**: Normalizing/standardizing continuous features (like income, expenditure) and encoding categorical variables (e.g., *Region*, *Urban/Rural*).
2. **Model Training**: Train the two models using training data, and find the best hyperparameter or network architecture.
3. **Model Evaluation**: Use cross-validation and metrics like **F1-score** (for classification) to assess model performance.
4. **Visualization**: Visualize the results of the models and compare their performance. For clustering models, visualize the clusters found by the model.

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js