SUBMITTED BY : ABHISHEK | abhishekpooja1408@gmail.com | +91-6006450679

ALL THE CODES AND RESULTS ARE AVILABLE AT THE REPO:
abhi-1408-shek/Innovate_with_GoIStats

Documentation of Software Tools and Code

Software Tools and Libraries

The analysis was conducted using **Python 3.10**, with the following libraries:

- **pandas (v2.0.3)** for data manipulation, merging, and cleaning.
- **numpy (v1.24.3)** for numerical operations and handling large datasets.
- **openpyxl (v3.1.2)** to parse the Excel layout file (Layout_HCES_2022-23.xls).
- **concurrent.futures** (built-in) for parallel processing of hierarchical data files.
- **plotly (v5.18.0)** and **seaborn (v0.12.2)** for interactive and statistical visualizations.
- **streamlit (v1.28.0)** to deploy the policy-ready dashboard.

Data Processing Workflow

Reading Layout File

The Layout_HCES_2022-23.xls file defined the structure of raw fixed-width text files. Columns were decoded using byte positions and variable names specified in the layout.

```
#CREATING LAYOUT.csv for FURTHER STEPS.
import pandas as pd

layout_data = pd.read_excel("./Documentation/Layout_HCES 2022-23_modified.xlsx", engine="openpyxl")

new_names = ["Slno", "Item", "QSec", "QItem", "QCol", "Length", "Byte_Start_position", "Dash", "Byte_End_position", "Remarks"]
layout_data.columns = new_names

levels = layout_data[layout_data["Item"].astype(str).str.contains("LEVEL", na=False)].index.tolist()

common_id_rows = layout_data.iloc[4:19][["Slno", "Item", "Length"]].dropna(subset=["Length"])

def extract_level_layout(start_row, next_level_start=None):
    if next_level_start is None:
        level_layout = layout_data.iloc[start_row:][["Slno", "Item", "Length"]]
    else:
        level_layout = layout_data.iloc[start_row:next_level_start][["Slno", "Item", "Length"]]
    return level_layout.dropna(subset=["Length"])

combined_layouts = []
```

Parallel Processing of Fixed-Width Files

The 15 hierarchical files were processed in parallel to optimize efficiency.

```python
from concurrent.futures import ProcessPoolExecutor

layout = pd.read_csv("./Output/Layout.csv")


print(layout.head())

levels = layout["Level"].unique()

State_list = pd.read_excel("Documentation/tabulation_state_code.xlsx", usecols=["st", "stn"])

State_list["st"] = pd.to_numeric(State_list["st"], errors='coerce')
print(State_list.head())

def read_fwf_level(level):
    file_name = f"./RawData/hces22_lvl_{level:02d}.TXT" #creting csv's


    if not os.path.exists(file_name):
        print(f"File not found: {file_name}, skipping level {level}")
        return None

    current_layout = layout[layout["Level"] == level]

    column_widths = current_layout["Length"].tolist()
    column_names = current_layout["Item"].apply(lambda x: x.replace(" ", "_")).tolist()


    df = pd.read_fwf(file_name, widths=column_widths, names=column_names, dtype=str)

    print(f"Columns in level {level} file:", df.columns)
```

Merging and Cleaning Data

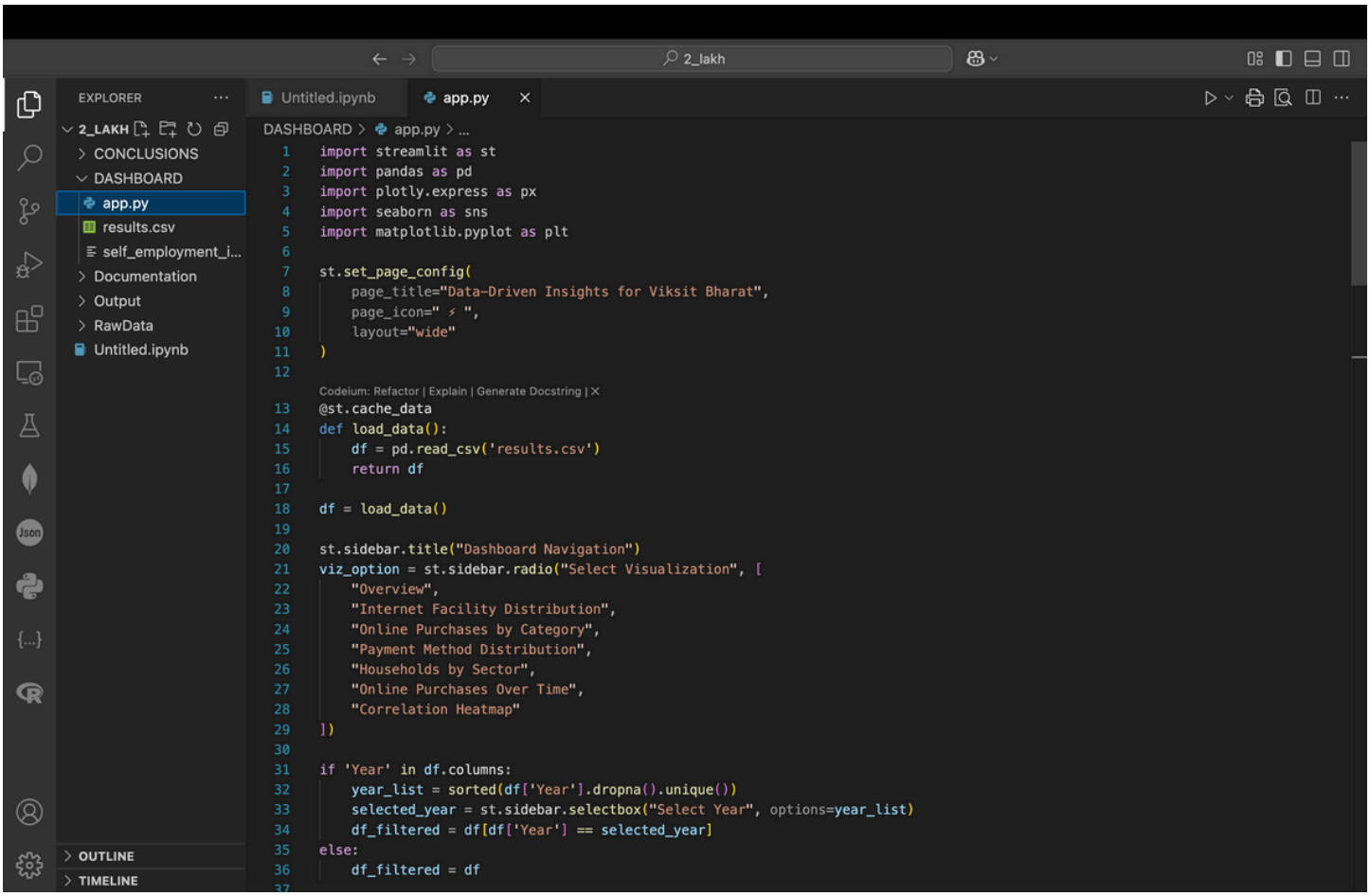
The consolidated dataset was cleaned to handle missing values and categorical responses.

```python
for file_path in file_paths:
    try:
        df = pd.read_csv(file_path, dtype=str)
        df.drop_duplicates(inplace=True)

        if 'HH_ID' in df.columns:
            if final_df is None:
                final_df = df
            else:
                final_df = pd.merge(final_df, df, on='HH_ID', how='outer', suffixes=('', '_dup'))


                final_df = final_df.loc[:, ~final_df.columns.duplicated()]
        else:
            print(f"Skipping {file_path} - 'HH_ID' column missing")
    except Exception as e:
        print(f"Error processing {file_path}: {e}")

output_file = os.path.join(data_dir, "results.csv")
if final_df is not None:
    final_df.to_csv(output_file, index=False)
    print(f"Consolidation complete. File saved to {output_file}")
else:
    print("No valid data to save.")
```

Data Visualization

Interactive Streamlit Dashboard

The dashboard enables policymakers to explore data dynamically.



Reproducibility

Dependency Installation--->In the app.py file; open console and run:

```
pip install -r requirements.txt
```

Running the Dashboard---> in the same terminal

```
streamlit run app.py
```

Code Repository

All code, datasets, and outputs are available at **Innovate_with_GoIStats**. The repository includes:

- **Untitled.py**: Script for data preprocessing.
- **app.py**: Streamlit dashboard code.
- **CONCLUSION**: Policy recommendations and reports.

- **DOCUMENTATION**: Layout and state code files.
- **OUTPUT**: Processed datasets and visualizations.

Ethical Compliance

- **Anonymization**: Removed personal identifiers (HH_ID, Person_Srl_No.).
- **Data Use**: Complied with MoSPI's non-disclosure terms.

Note: The full code (Untitled.py, app.py) is embedded in the subsequent pages of this document for review.

In [ ]:
```
#ABHISHEK SHARMA - Data-Driven Insight

#KINDLY REFER TO THE PROJECT FILE IN THE CONCULSIONS FOLDER FOR MORE DETAILS
```

In [ ]:
```
#CREATING LAYOUT.csv for FURTHER STEPS.
import pandas as pd

layout_data = pd.read_excel("./Documentation/Layout_HCES 2022-23_modified.xl

new_names = ["Slno", "Item", "QSec", "QItem", "QCol", "Length", "Byte_Start_
layout_data.columns = new_names

levels = layout_data[layout_data["Item"].astype(str).str.contains("LEVEL", r

common_id_rows = layout_data.iloc[4:19][["Slno", "Item", "Length"]].dropna(s

def extract_level_layout(start_row, next_level_start=None):
    if next_level_start is None:
        level_layout = layout_data.iloc[start_row:][["Slno", "Item", "Length
    else:
        level_layout = layout_data.iloc[start_row:next_level_start][["Slno",
    return level_layout.dropna(subset=["Length"])

combined_layouts = []

for i in range(len(levels)):
    start_row = levels[i] + 3
    next_level_start = levels[i + 1] if i < len(levels) - 1 else len(layout_
    level_layout = extract_level_layout(start_row, next_level_start)

    if i == 0:
        combined_layout = level_layout
    else:
        combined_layout = pd.concat([common_id_rows, level_layout], ignore_i

    combined_layouts.append(combined_layout)

final_combined_layout = pd.concat(combined_layouts, keys=range(1, len(combir

final_combined_layout = final_combined_layout[final_combined_layout["Item"]

final_combined_layout.to_csv("./Output/Layout.csv", index=False)
#SAVING IN OUTPUT FOLDER
```

In [ ]:
```
!pip install pandas numpy openpyxl pyreadr multiprocess
#installing dependencies
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-pack
ages (2.2.2)
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packa
ges (2.0.2)
Requirement already satisfied: openpyxl in /usr/local/lib/python3.11/dist-pa
ckages (3.1.5)
Collecting pyreadr
  Downloading pyreadr-0.5.3-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_
x86_64.whl.metadata (1.4 kB)
Collecting multiprocess
  Downloading multiprocess-0.70.17-py311-none-any.whl.metadata (7.2 kB)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/pyth
on3.11/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dis
t-packages (from pandas) (2025.1)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/d
ist-packages (from pandas) (2025.1)
Requirement already satisfied: et-xmlfile in /usr/local/lib/python3.11/dist-
packages (from openpyxl) (2.0.0)
Collecting dill>=0.3.9 (from multiprocess)
  Downloading dill-0.3.9-py3-none-any.whl.metadata (10 kB)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-pa
ckages (from python-dateutil>=2.8.2->pandas) (1.17.0)
Downloading pyreadr-0.5.3-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x8
6_64.whl (411 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 411.7/411.7 kB 9.1 MB/s eta 0:0
0:00
Downloading multiprocess-0.70.17-py311-none-any.whl (144 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 144.3/144.3 kB 9.2 MB/s eta 0:0
0:00
Downloading dill-0.3.9-py3-none-any.whl (119 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 119.4/119.4 kB 8.5 MB/s eta 0:0
0:00
Installing collected packages: dill, multiprocess, pyreadr
Successfully installed dill-0.3.9 multiprocess-0.70.17 pyreadr-0.5.3
```

In [ ]:

```python
#DATA PREPROCESSING MULTI-THREADING
import pandas as pd
import numpy as np
import os
from concurrent.futures import ProcessPoolExecutor

layout = pd.read_csv("./Output/Layout.csv")


print(layout.head())

levels = layout["Level"].unique()

State_list = pd.read_excel("Documentation/tabulation_state_code.xlsx", usecc

State_list["st"] = pd.to_numeric(State_list["st"], errors='coerce')
print(State_list.head())

def read_fwf_level(level):
    file_name = f"./RawData/hces22_lvl_{level:02d}.TXT" #creting csv's
```

```
    if not os.path.exists(file_name):
        print(f"File not found: {file_name}, skipping level {level}")
        return None

    current_layout = layout[layout["Level"] == level]

    column_widths = current_layout["Length"].tolist()
    column_names = current_layout["Item"].apply(lambda x: x.replace(" ", "_"

    df = pd.read_fwf(file_name, widths=column_widths, names=column_names, dt

    print(f"Columns in level {level} file:", df.columns)

    df["Multiplier"] = pd.to_numeric(df.get("Multiplier"), errors="coerce")
    df["Weights"] = df["Multiplier"] / 100

    fsu_col = next((col for col in df.columns if "FSU" in col), None)
    stratum_col = next((col for col in df.columns if "Second" in col), None)
    hhld_col = next((col for col in df.columns if "Sample" in col), None)

    if fsu_col and stratum_col and hhld_col:
        df["HH_ID"] = df[fsu_col].fillna("").astype(str) + \
                      df[stratum_col].fillna("").astype(str) + \
                      df[hhld_col].fillna("").astype(str)

    if "State" in df.columns:
        df["State"] = pd.to_numeric(df["State"], errors="coerce")
        df = df.merge(State_list, left_on="State", right_on="st", how="left"

    return df
num_workers = max(os.cpu_count() - 2, 1)
with ProcessPoolExecutor(max_workers=num_workers) as executor:
    data_frames = list(executor.map(read_fwf_level, levels))

valid_data = [(level, df) for level, df in zip(levels, data_frames) if df is

output_dir = "./Output"
os.makedirs(output_dir, exist_ok=True)

for level, df in valid_data:
    file_prefix = f"level_{level}"
    df.to_csv(os.path.join(output_dir, f"{file_prefix}.csv"), index=False)
    df.to_pickle(os.path.join(output_dir, f"{file_prefix}.pkl"))

del layout, State_list, data_frames

print("Processing complete.")
```

```
   Level  Slno            Item  Length
0      1     1     Survey Name       4
1      1     2            Year       4
2      1     3  FSU Serial No.       5
3      1     4          Sector       1
4      1     5           State       2
   st                stn
0  28     Andhra Pradesh
1  12  Arunachal Pradesh
2  18              Assam
3  10              Bihar
4  22        Chattisgarh
Columns in level 1 file: Index(['Survey_Name', 'Year', 'FSU_Serial_No.', 'Se
ctor', 'State',
       'NSS-Region', 'District', 'Stratum', 'Sub-stratum', 'Panel',
       'Sub-sample', 'FOD-Sub-Region', 'Sample_SU_No.',
       'Sample_Sub-division_no.', 'Second-stage-stratum_no.',
       'Sample_hhld._No.', 'Questionnaire_No.', 'Level_', 'Survey_Code',
       'Reason_for_substitution_Code', 'Multiplier'],
      dtype='object')
Columns in level 2 file: Index(['Year', 'FSU_Serial_No.', 'Sector', 'State',
'NSS-Region', 'District',
       'Stratum', 'Sub-stratum', 'Panel', 'Sub-sample', 'FOD-Sub-Region',
       'Sample_SU_No.', 'Sample_Sub-division_no.', 'Second-stage-stratum_n
o.',
       'Sample_hhld._No.', 'Questionnaire_No.', 'Level', 'Person_Srl_No.',
       'Relation_to_head_(code)', 'Gender', 'Age(in_years)',
       'Marital_Status_(code)', 'Highest_educational_level_attained_(code)',
       'Total_year_of_education_completed',
       'Whether_used_internet_from_any_location_during_last_30_days',
       'No._of_days_stayed_away_from_home_during_last_30_days',
       'No._of_meals_usually_taken_in_a_day',
       'No._of_meals_taken_during_last_30_days_from_school,_balwadi_etc.',
       'No._of_meals_taken_during_last_30_days_from_employer_as\nperquisites
_or_part_of_wage',
       'No._of_meals_taken_during_last_30_days__others',
       'No._of_meals_taken_during_last_30_days_on_payment',
       'No._of_meals_taken_during_last_30_days_at_home',
       'Status_of_Member_as_on_revisit',
       'FDQ_original_member(generated_field)', 'Multiplier'],
      dtype='object')
Columns in level 3 file: Index(['Year', 'FSU_Serial_No.', 'Sector', 'State',
'NSS-Region', 'District',
       'Stratum', 'Sub-stratum', 'Panel', 'Sub-sample', 'FOD-Sub-Region',
       'Sample_SU_No.', 'Sample_Sub-division_no.', 'Second-stage-stratum_n
o.',
       'Sample_hhld._No.', 'Questionnaire_No.', 'Level', 'HH_Size_(For_FD
Q)',
       'Whether_any_household_member_(excluding_those_employed_by_the_househ
old_and_paying_guests)_was_engaged_in_economic_activities_during_last_365_da
ys?_',
       'NCO-2015_Code(3-digit)', 'NIC-2008_Code(5-digit)',
       'Broad_activities_from_which_maximum_income_was_derived_by_the_househ
old_during_last_365_days',
       'Whether_major_source_of_income_from_self-_employment_was_from_agricu
ltural_/_non-_agricultural_sector',
```

```
       'Whether_the_major_income_from_regular_wage/salary_earning_from_agric
ultural_/non-_agricultural_sector',
       'Whether_the_major_income_from_casual_labour_was_from_agriculturalsec
tor/_non-_agricultural_sector',
       'Household_Type', 'Religion_of_the_head_of_the_household',
       'Social_Group_of_the_head_of_the_household',
       'Does_the_household_own_(owned_&_possessed_or_leased_out)_any_land_(w
ithin_the_country)_as_on_the_date_of_survey?',
       'Type_of_land_owned',
       'What_is_the_total_area_of_all_owned_(owned_and_possessed_or_leased_o
ut)_land_(within_the_country)_by_the_household_as_on_the_date_of_survey_(are
a_in_acre)?(upto_two_places_of_decimal)_',
       'Does_the_household_have_a_dwelling_unit_at_present_place_of_enumerat
ion?__',
       'Type_of_Dwelling_Unit',
       'Basic_building_Material_used_for_major_portion_of_the_wall_of_the_dw
elling_Unit',
       'Basic_building_Material_used_for_construction_of_the_major_portion_o
f_the_outer_exposed_part_of_the_roof_of_the_dwelling_unit',
       'Basic_Building_Material_used_for_construction_of_the_major_portion_o
f_the_floor_of_the_dwelling_Unit``',
       '_Primary_source_of_energy_of_the_household_for_cooking',
       '_Primary_source_of_energy_of_the_household_for_Lighting_',
       'Source_of_Drinking_Water_(Last_365_days)',
       'Time_taken_by_the_household_for_a_single_trip_to_reach_the_source_(f
rom_which_most_of_the_drinking_water_is_fetched),_obtain_water_and_back_to_h
ousehold_(in_Minutes)',
       'Type_of_access_of_the_household_to_latrine',
       'Type_of_latrine_in_which_the_household_has_access',
       'Type_of_ration_card_possessed_by_the_household_as_on_the_date_of_sur
vey',
       'Prevailing_rate_of_rent_in_the_locality_is_available_(FOR_Rural_onl
y)',
       'Benefitted_from_PMGKY_as_on_the_date_on_the_survey',
       'Any_member_of_the_household_of_age_0_-_18_years_died_during_the_peri
od_of_last_5_years_preceding_the_date_of_survey',
       'No_of_members_of_the_household_of_age_0_–_18_years_died_during_the_p
eriod_of_last_5_years_preceding_the_date_of_survey?',
       'Multiplier'],
      dtype='object')
Columns in level 4 file: Index(['Year', 'FSU_Serial_No.', 'Sector', 'State',
'NSS-Region', 'District',
       'Stratum', 'Sub-stratum', 'Panel', 'Sub-sample', 'FOD-Sub-Region',
       'Sample_SU_No.', 'Sample_Sub-division_no.', 'Second-stage-stratum_n
o.',
       'Sample_hhld._No.', 'Questionnaire_No.', 'Level',
       'Whether_the_household_procured_any_item_using_the_ration_card_during
_the_last_30_days',
       'Which_item_procured_using_ration_card_during_the_last_30_days_-_Ric
e',
       'Which_item_procured_using_ration_card_during_the_last_30_days-_Whea
t',
       'Which_item_procured_using_ration_card_during_the_last_30_days_-_Coar
se_Grain',
       'Which_item_procured_using_ration_card_during_the_last_30_days_-_Suga
```

```
       'Which_item_procured_using_ration_card_during_the_last_30_days-_Pulse
s',
       'Which_item_procured_using_ration_card_during_the_last_30_days-_Edibl
e_Oil',
       'Which_item_procured_using_ration_card_during_the_last_30_days-_Other
_food_item_(including_salt,_gram,_etc.)',
       'Whether_any_online_purchase/payment_has_been_made_during_the_referen
ce_period_to_buy_-_Groceries',
       'Whether_any_online_purchase/payment_has_been_made_during_the_referen
ce_period_to_buy_-_Milk_&_its_products',
       'Whether_any_online_purchase/payment_has_been_made_during_the_referen
ce_period_to_buy_-_Vegetables',
       'Whether_any_online_purchase/payment_has_been_made_during_the_referen
ce_period_to_buy_-_Fresh_Fruits',
       'Whether_any_online_purchase/payment_has_been_made_during_the_referen
ce_period_to_buy_-_Dry_Fruits',
       'Whether_any_online_purchase/payment_has_been_made_during_the_referen
ce_period_to_buy_-_Egg,fish_&_meat',
       'Whether_any_online_purchase/payment_has_been_made_during_the_referen
ce_period_to_buy_-_Served_processed_food',
       'Whether_any_online_purchase/payment_has_been_made_during_the_referen
ce_period_to_buy_-_Packed_processed_food',
       'Whether_any_online_purchase/payment_has_been_made_during_the_referen
ce_period_to_buy_-_Other_food_items',
       'Ceremony_Pereformed_during_last_30_days',
       'Meals_served_to_non-household_members_during_the_last_30_days',
       'Multiplier'],
      dtype='object')
Columns in level 5 file: Index(['Year', 'FSU_Serial_No.', 'Sector', 'State',
'NSS-Region', 'District',
       'Stratum', 'Sub-stratum', 'Panel', 'Sub-sample', 'FOD-Sub-Region',
       'Sample_SU_No.', 'Sample_Sub-division_no.', 'Second-stage-stratum_n
o.',
       'Sample_hhld._No.', 'Questionnaire_No.', 'Level', 'Item_Code_',
       'Consumption_out_of_home_produce-Quantity(0.000)',
       'Consumption_out_of_home_produce-Value(Rs.)',
       'Total_Consumption--Quantity(0.000)', 'Total_Consumption--Value(R
s.)',
       'Source', 'Multiplier'],
      dtype='object')
Columns in level 6 file: Index(['Year', 'FSU_Serial_No.', 'Sector', 'State',
'NSS-Region', 'District',
       'Stratum', 'Sub-stratum', 'Panel', 'Sub-sample', 'FOD-Sub-Region',
       'Sample_SU_No.', 'Sample_Sub-division_no.', 'Second-stage-stratum_n
o.',
       'Sample_hhld._No.', 'Questionnaire_No.', 'Level', 'Item_Code_',
       'Total_Consumption-Quantity(0.000)', 'Total_Consumption-Value(Rs.)',
       'Source', 'Multiplier'],
      dtype='object')
Columns in level 7 file: Index(['Year', 'FSU_Serial_No.', 'Sector', 'State',
'NSS-Region', 'District',
       'Stratum', 'Sub-stratum', 'Panel', 'Sub-sample', 'FOD-Sub-Region',
       'Sample_SU_No.', 'Sample_Sub-division_no.', 'Second-stage-stratum_n
o.',
       'Sample_hhld._No.', 'Questionnaire_No.', 'Level',
       'Whether_the_household_procured_kerosene_using_ration_card_during_las
```

```
t_30_days_-_Kerosene',
       'Whether_household_received_subsidy_on_LPG_cylinder_during_the_last_3
_months?',
       'If_yes_in_Q4.2.1_Number_of_subsidized_LPG_cylinder_received_during_t
he_last_3_months_preceding_the_date_of_survey_(number)_',
       'Whether_household_received_free_electricity_during_the_last_30_day
s',
       'Whether_any_household_member_is_attending/attended_educational_insti
tution_during_last_365_days',
       'If_Code_1_in_Q4.2.3_Number_attending_/attended_Govt._Institution_?',
       'If_Code_1_in_Q4.2.3_Number__attending_/attended_Private_Institution
_?',
       '[Checkbox]_If_Code_1_in_Q4.2.3_whether_any_member_of_the_household_r
eceived_following_items_free_in_last_365_days_:_Textbooks_?_',
       'Total_no._of_free_textbooks_received',
       '[Checkbox]_If_Code_1_in_Q4.2.3_whether_any_member_of_the_household_r
eceived_following_items_free_in_last_365_days_:_Stationary_(pen,notepad_et
c.)_?_',
       'Total_no._of_free_stationaries_received',
       '[Checkbox]_If_Code_1_in_Q4.2.3_whether_any_member_of_the_household_r
eceived_following_items_free_in_last_365_days_:_School_Bag_?_',
       'Total_no._of_free_school_bags_received',
       '[Checkbox]_If_Code_1_in_Q4.2.3_whether_any_member_of_the_household_r
eceived_following_items_free_in_last_365_days_:_Others_?_',
       'Total_no._of_free_other_items_received',
       'If_Code_1_in_Q4.2.3_Whether_Any_member_received_reimbursement/waiver
_of_school/clg._Fee_during_last_365_days_?',
       'If_Code_1_in_Q4.2.6_Number_of_member_received_reimbursement/waiver_
_?',
       'Is_one_or_more_member_of_the_household_a_benificiary_of_Pradhan_Mant
ri_Jan_Aarogya_Yojana_(Ayushman_Bharat)_or_any_other_state_specific_public_h
ealth_scheme_as_on_the_date_of_survey',
       'If_Code_1_in_Q4.2.7_Number_of_beneficiaries_?',
       'Whether_there_was_any_case_of_hospitalization_in_the_household_durin
g_last_365_days?',
       'Whether_one_or_more_member_of_the_household_has_received_benefits_of
_medical_treatment_(medical_–_hospitalisation)under_Pradhan_Mantri_Jan_Aarog
ya_Yojana_Card_(Ayushman_Bharat)_or_any_other_state_specific_public_health_s
cheme_during_the_last_365_days',
       'If_yes_in_Q4.2.9_number_of_member_received_benifit_?',
       'If_yes_in_Q4.2.9_Amount_?',
       'Whether_any_online_purchase/payment_has_been_made_during_the_referen
ce_period_to_buy_-_Fuel_&_light',
       'Whether_any_online_purchase/payment_has_been_made_during_the_referen
ce_period_to_buy_-_Toilet_articles_&_other_household_consumables',
       'Whether_any_online_purchase/payment_has_been_made_during_the_referen
ce_period_to_buy_-_Education',
       'Whether_any_online_purchase/payment_has_been_made_during_the_referen
ce_period_to_buy-_Medicine_&_other_medical_services',
       'Whether_any_online_purchase/payment_has_been_made_during_the_referen
ce_period_to_buy-_Services_(Travel,_Recharges,_Bill_payment,_Cinema/Theatre,
_internet,_etc.)_',
       'Household_has_internet_facility_as_on_the_date_of_the_survey',
       'Multiplier'],
      dtype='object')
Columns in level 8 file: Index(['Year', 'FSU_Serial_No.', 'Sector', 'State',
```

```
'NSS-Region', 'District',
       'Stratum', 'Sub-stratum', 'Panel', 'Sub-sample', 'FOD-Sub-Region',
       'Sample_SU_No.', 'Sample_Sub-division_no.', 'Second-stage-stratum_n
o.',
       'Sample_hhld._No.', 'Questionnaire_No.', 'Level', 'Item_Code_',
       'Consumption_out_of_home_produce--Quantity(0.000)',
       'Consumption_out_of_home_produce--Value(Rs.)',
       'Total_Consumption--Quantity(0.000)', 'Total_Consumption--Value(R
s.)',
       'Source', 'Multiplier'],
      dtype='object')
Columns in level 9 file: Index(['Year', 'FSU_Serial_No.', 'Sector', 'State',
'NSS-Region', 'District',
       'Stratum', 'Sub-stratum', 'Panel', 'Sub-sample', 'FOD-Sub-Region',
       'Sample_SU_No.', 'Sample_Sub-division_no.', 'Second-stage-stratum_n
o.',
       'Sample_hhld._No.', 'Questionnaire_No.', 'Level', 'Item_Code_',
       'Value(Rs.)', 'Multiplier'],
      dtype='object')
Columns in level 10 file: Index(['Year', 'FSU_Serial_No.', 'Sector', 'Stat
e', 'NSS-Region', 'District',
       'Stratum', 'Sub-stratum', 'Panel', 'Sub-sample', 'FOD-Sub-Region',
       'Sample_SU_No.', 'Sample_Sub-division_no.', 'Second-stage-stratum_n
o.',
       'Sample_hhld._No.', 'Questionnaire_No.', 'Level', 'Item_Code_',
       'Consumption_out_of_home_produce--Quantity(0.000)',
       'Consumption_out_of_home_produce--Value(Rs.)',
       'Total_Consumption--Quantity(0.000)', 'Total_Consumption--Value(R
s.)',
       'Source', 'Multiplier'],
      dtype='object')
Columns in level 11 file: Index(['Year', 'FSU_Serial_No.', 'Sector', 'Stat
e', 'NSS-Region', 'District',
       'Stratum', 'Sub-stratum', 'Panel', 'Sub-sample', 'FOD-Sub-Region',
       'Sample_SU_No.', 'Sample_Sub-division_no.', 'Second-stage-stratum_n
o.',
       'Sample_hhld._No.', 'Questionnaire_No.', 'Level',
       'Which_online_item_purchased/Paid_during_last_365_days_-_Clothing',
       'Which_online_item_purchased/Paid_during_last_365_days-_Footwear',
       'Which_online_item_purchased/Paid_during_last_365_days_-_Furniture_&_
fixtures',
       'Which_online_item_purchased/Paid_during_last_365_days_-_Mobile_hands
et',
       'Which_online_item_purchased/Paid_during_last_365_days-_Personal_good
s_(laptop/PC,_tablet,_clock,_watch,_spectacles,_contact_\nlenses,_etc.)',
       'Which_online_item_purchased/Paid_during_last_365_days-_Goods_for_rec
reation_(TV,_camera,_pen-drive,_musical_instruments,_\netc.)_\n',
       'Which_online_item_purchased/Paid_during_last_365_days_-Cooking_&_oth
er_household_appliances_',
       'Which_online_item_purchased/Paid_during_last_365_days_-_Crockery_&_u
tensils',
       'Which_online_item_purchased/Paid_during_last_365_days_-_Sports_good
s',
       'Which_online_item_purchased/Paid_during_last_365_days-_Medical_equip
ment',
       'Which_online_item_purchased/Paid_during_last_365_days-_Bedding',
```

```
       '[Checkbox]_:_Whether_One_or_more_member_of_the_household_received_it
ems_free_of_cost__during_last_365_days:_Laptop/PC',
       'Total_number_of_free_Laptop/PC',
       '[Checkbox]_:_Whether_One_or_more_member_of_the_household_received_it
ems_free_of_cost__during_last_365_days:_Tablet',
       'Total_number_of_free_Tablet',
       '[Checkbox]_:_Whether_One_or_more_member_of_the_household_received_it
ems_free_of_cost__during_last_365_days_:_Mobile',
       'Total_number_of_free_Mobile',
       '[Checkbox]_:_Whether_One_or_more_member_of_the_household_received_it
ems_free_of_cost__during_last_365_days_:_Bicycle',
       'Total_number_of_free_Bicycle',
       '[Checkbox]_:_Whether_One_or_more_member_of_the_household_received_it
ems_free_of_cost__during_last_365_days_:_Motorcycle/Scooty',
       'Total_number_of_free_Motorcycle/Scooty',
       '[Checkbox]_:_Whether_One_or_more_member_of_the_household_received_it
ems_free_of_cost__during_last_365_days:_Clothing_(Schooling_uniform_etc.)',
       'Total_number_of_free_Clothing_(Schooling_uniform_etc.)',
       '[Checkbox]_:_Whether_One_or_more_member_of_the_household_received_it
ems_free_of_cost__during_last_365_days:_Footwear_(School_shoe_etc.)',
       'Total_number_of_free_Footwear_(School_shoe_etc.)',
       '[Checkbox]_:_Whether_One_or_more_member_of_the_household_received_it
ems_free_of_cost__during_last_365_days:_Other',
       'Total_number_of_free_Other_items',
       'Whether_household_possessed_one_or_more_item_as_on_the_date_of_the_s
urvey-_Television_',
       'Whether_household_possessed_one_or_more_item_as_on_the_date_of_the_s
urvey-_Radio',
       'Whether_household_possessed_one_or_more_item_as_on_the_date_of_the_s
urvey_-_Laptop/PC',
       'Whether_household_possessed_one_or_more_item_as_on_the_date_of_the_s
urvey-_Mobile_handset',
       'Whether_household_possessed_one_or_more_item_as_on_the_date_of_the_s
urvey-_Bicycle',
       'Whether_household_possessed_one_or_more_item_as_on_the_date_of_the_s
urvey-_Motorcycle,_scooter_',
       'Whether_household_possessed_one_or_more_item_as_on_the_date_of_the_s
urvey-_Motor_car/jeep/van',
       'Whether_household_possessed_one_or_more_item_as_on_the_date_of_the_s
urvey-_Trucks',
       'Whether_household_possessed_one_or_more_item_as_on_the_date_of_the_s
urvey_-_Animal_cart',
       'Whether_household_possessed_one_or_more_item_as_on_the_date_of_the_s
urvey-_Refrigerator',
       'Whether_household_possessed_one_or_more_item_as_on_the_date_of_the_s
urvey-_Washing_machine',
       'Whether_household_possessed_one_or_more_item_as_on_the_date_of_the_s
urvey-_Air_conditioner/air_cooler_',
       'Type_of_multichannel_television_facility_is_used_by_the_household_as
_on_the_date_of_the_survey',
       'Multiplier'],
      dtype='object')
Columns in level 12 file: Index(['Year', 'FSU_Serial_No.', 'Sector', 'Stat
e', 'NSS-Region', 'District',
       'Stratum', 'Sub-stratum', 'Panel', 'Sub-sample', 'FOD-Sub-Region',
       'Sample_SU_No.', 'Sample_Sub-division_no.', 'Second-stage-stratum_n
```

```
o.',
       'Sample_hhld._No.', 'Questionnaire_No.', 'Level', 'Item_Code',
       'Quantity_(0.00)', 'Value_(Rs.)', 'Multiplier'],
      dtype='object')
Columns in level 13 file: Index(['Year', 'FSU_Serial_No.', 'Sector', 'Stat
e', 'NSS-Region', 'District',
       'Stratum', 'Sub-stratum', 'Panel', 'Sub-sample', 'FOD-Sub-Region',
       'Sample_SU_No.', 'Sample_Sub-division_no.', 'Second-stage-stratum_n
o.',
       'Sample_hhld._No.', 'Questionnaire_No.', 'Level', 'Item_Code_',
       '_First-hand_purchase:number', 'Whether_purchased_on_hire',
       '_First-hand_purchase:Value(Rs.)',
       'Cost_of_repair_&_maintenance_/cost_of_raw_material_and_services_for_
construction_and_repair(Rs.)',
       '2nd-hand_purchase:Number', '_2nd-hand_purchase:Value(Rs.)',
       'Total_expenditure(Rs.)', 'Multiplier'],
      dtype='object')
Columns in level 14 file: Index(['Year', 'FSU_Serial_No.', 'Sector', 'Stat
e', 'NSS-Region', 'District',
       'Stratum', 'Sub-stratum', 'Panel', 'Sub-sample', 'FOD-Sub-Region',
       'Sample_SU_No.', 'Sample_Sub-division_no.', 'Second-stage-stratum_n
o.',
       'Sample_hhld._No.', 'Questionnaire_No.', 'Level', 'Section',
       'Item_Code', 'Value_(in_Rs)', 'Multiplier'],
      dtype='object')
Columns in level 15 file: Index(['Year', 'FSU_Serial_No.', 'Sector', 'Stat
e', 'NSS-Region', 'District',
       'Stratum', 'Sub-stratum', 'Panel', 'Sub-sample', 'FOD-Sub-Region',
       'Sample_SU_No.', 'Sample_Sub-division_no.', 'Second-stage-stratum_n
o.',
       'Sample_hhld._No.', 'Questionnaire_No.', 'Level', 'Section',
       'Time_taken_to_canvass_the_questionnaire_(in_minutes)',
       'Household's_usual_consumption_expenditure_in_a_month_(in_Rs.)',
       'Total_expenditure_incurred_on_online_purchase/payment_in_last_30_day
s',
       'Informant_code', 'Response_code', 'Household_size', 'Multiplier'],
      dtype='object')
Processing complete.
```

In [ ]:

```python
#CONSOLIDATING CSV'S TO A SINGLE FILE i.e Results.csv
import pandas as pd
import os
from glob import glob

data_dir = "./Output"

file_paths = sorted(glob(os.path.join(data_dir, "level_*.csv")))

if not file_paths:
    print("No CSV files found in", data_dir)
else:
    print("Found CSV files:", file_paths)

final_df = None

for file_path in file_paths:
```

```
    try:
        df = pd.read_csv(file_path, dtype=str)
        df.drop_duplicates(inplace=True)

        if 'HH_ID' in df.columns:
            if final_df is None:
                final_df = df
            else:
                final_df = pd.merge(final_df, df, on='HH_ID', how='outer', s

                final_df = final_df.loc[:, ~final_df.columns.duplicated()]
        else:
            print(f"Skipping {file_path} - 'HH_ID' column missing")
    except Exception as e:
        print(f"Error processing {file_path}: {e}")

output_file = os.path.join(data_dir, "results.csv")
if final_df is not None:
    final_df.to_csv(output_file, index=False)
    print(f"Consolidation complete. File saved to {output_file}")
else:
    print("No valid data to save.")
```

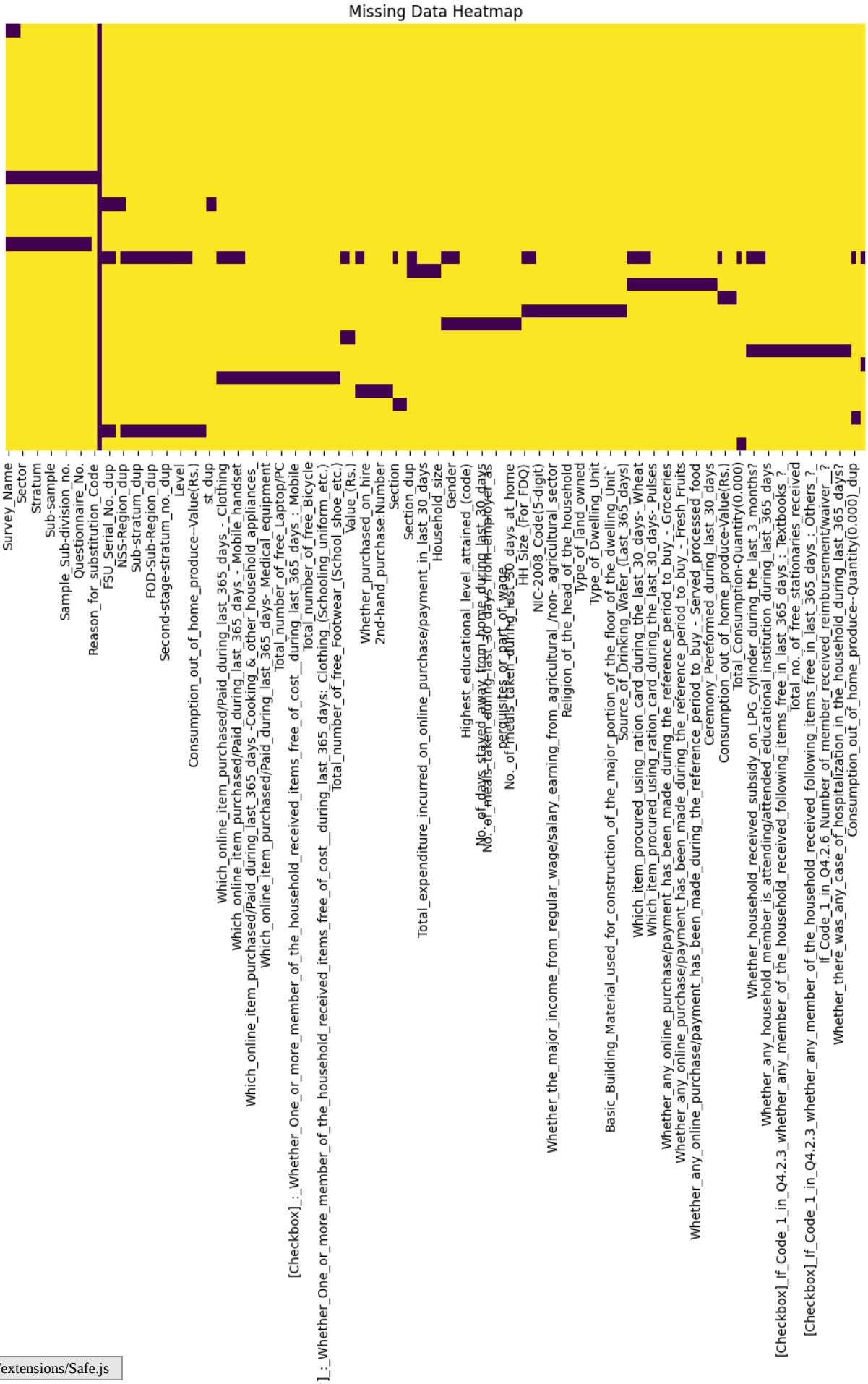
```
Found CSV files: ['./Output/level_1.csv', './Output/level_10.csv', './Outpu
t/level_11.csv', './Output/level_12.csv', './Output/level_13.csv', './Outpu
t/level_14.csv', './Output/level_15.csv', './Output/level_2.csv', './Output/
level_3.csv', './Output/level_4.csv', './Output/level_5.csv', './Output/leve
l_6.csv', './Output/level_7.csv', './Output/level_8.csv', './Output/level_9.
csv']
Consolidation complete. File saved to ./Output/results.csv
```

In [ ]:
```
#checking for duplicates
import pandas as pd

file_path = "./Output/results.csv"
df = pd.read_csv(file_path)

df = df.dropna(axis=1, how="all")

df = df.loc[:, ~df.columns.duplicated()]

print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32 entries, 0 to 31
Columns: 180 entries, Survey_Name to Value(Rs.)
dtypes: float64(56), object(124)
memory usage: 45.1+ KB
None
```

In [ ]:
```
# Convert columns that should be numeric
for col in df.columns:
    if df[col].dtype == "object":
        try:
            df[col] = pd.to_numeric(df[col])
        except ValueError:
            pass
```

In [ ]:
```
#HEATMAP TO CHECK FOR MISSING VALUES BEFORE DATA VISUALIZATION
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(12, 6))
sns.heatmap(df.isnull(), cmap="viridis", cbar=False, yticklabels=False)
plt.title("Missing Data Heatmap")
plt.show()
```



[Checkbox

In [ ]:
```
# CHECKING THE COLUMN VALUES
import pandas as pd
file_path = './Output/results.csv'
df = pd.read_csv(file_path)
df.head()
```

Out[ ]:

	Survey_Name	Year	FSU_Serial_No.	Sector	State	NSS-Region	District	Stratu
0	size	157	04760	NaN	NaN	NaN	NaN	Na
1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na

5 rows × 219 columns

In [16]:
```
# Display column names
print(df.columns)
```

```
Index(['Survey_Name', 'Year', 'FSU_Serial_No.', 'Sector', 'State',
       'NSS-Region', 'District', 'Stratum', 'Sub-stratum', 'Panel',
       ...
       'If_yes_in_Q4.2.9_Amount_?',
       'Whether_any_online_purchase/payment_has_been_made_during_the_referen
ce_period_to_buy_-_Fuel_&_light',
       'Whether_any_online_purchase/payment_has_been_made_during_the_referen
ce_period_to_buy_-_Toilet_articles_&_other_household_consumables',
       'Whether_any_online_purchase/payment_has_been_made_during_the_referen
ce_period_to_buy_-_Education',
       'Whether_any_online_purchase/payment_has_been_made_during_the_referen
ce_period_to_buy-_Medicine_&_other_medical_services',
       'Whether_any_online_purchase/payment_has_been_made_during_the_referen
ce_period_to_buy-_Services_(Travel,_Recharges,_Bill_payment,_Cinema/Theatre,
_internet,_etc.)_',
       'Household_has_internet_facility_as_on_the_date_of_the_survey',
       'Consumption_out_of_home_produce--Quantity(0.000)_dup',
       'Consumption_out_of_home_produce--Value(Rs.)_dup', 'Value(Rs.)'],
      dtype='object', length=219)
```

In [17]:
```
# Check for columns that might be related to employment
employment_cols = [col for col in df.columns if 'employment' in col.lower()]
print(employment_cols)
```

```
['Whether_major_source_of_income_from_self-_employment_was_from_agricultural
_/_non-_agricultural_sector']
```

In [ ]:
```
#SAVED THE "results.csv" file and use it for data visualization further, che
```

DASHBOARD/app.py

```python
import streamlit as st
import pandas as pd
import plotly.express as px
import seaborn as sns
import matplotlib.pyplot as plt

st.set_page_config(
    page_title="Data-Driven Insights for Viksit Bharat",
    page_icon="📊",
    layout="wide"
)

@st.cache_data
def load_data():
    df = pd.read_csv('results.csv')
    return df

df = load_data()

st.sidebar.title("Dashboard Navigation")
viz_option = st.sidebar.radio("Select Visualization", [
    "Overview",
    "Internet Facility Distribution",
    "Online Purchases by Category",
    "Payment Method Distribution",
    "Households by Sector",
    "Online Purchases Over Time",
    "Correlation Heatmap"
])

if 'Year' in df.columns:
    year_list = sorted(df['Year'].dropna().unique())
    selected_year = st.sidebar.selectbox("Select Year", options=year_list)
    df_filtered = df[df['Year'] == selected_year]
else:
    df_filtered = df

online_purchase_categories = [
    'Whether_any_online_purchase/payment_has_been_made_during_the_reference_period_to_buy_-_Fuel_&_light',
    'Whether_any_online_purchase/payment_has_been_made_during_the_reference_period_to_buy_-_Toilet_articles_&_other_household_consumables',
    'Whether_any_online_purchase/payment_has_been_made_during_the_reference_period_to_buy_-_Education',
    'Whether_any_online_purchase/payment_has_been_made_during_the_reference_period_to_buy-_Medicine_&_other_medical_services',
```

```
    'Whether_any_online_purchase/payment_has_been_made_during_the_reference_period_to_buy-_Services_(Travel,_Recharges,_Bill_payment,_Cinema/Theatre,_internet,_etc.)_'
]

if viz_option == "Overview":
    st.title("Data-Driven Insights for Viksit Bharat")
    st.markdown("### Overview")
    st.dataframe(df.head(10))
    st.markdown(f"**Total Records:** {df.shape[0]}  |  **Columns:** {df.shape[1]}")

elif viz_option == "Internet Facility Distribution":
    st.title("Internet Facility Distribution in Households")
    if 'Household_has_internet_facility_as_on_the_date_of_the_survey' in df_filtered.columns:
        internet_counts = df_filtered['Household_has_internet_facility_as_on_the_date_of_the_survey'].value_counts()
        fig = px.pie(
            names=internet_counts.index,
            values=internet_counts.values,
            title="Internet Facility Availability",
            color_discrete_sequence=px.colors.sequential.RdBu
        )
        st.plotly_chart(fig, use_container_width=True)

elif viz_option == "Online Purchases by Category":
    st.title("Online Purchases by Category")

    for col in online_purchase_categories:
        if col in df_filtered.columns:
            df_filtered[col] = df_filtered[col].map({'Yes': 1, 'No': 0}).fillna(0)

    category_counts = df_filtered[online_purchase_categories].apply(pd.Series.value_counts).fillna(0)

    if category_counts.shape[1] == 2:
        category_counts.columns = ['No', 'Yes']
    elif category_counts.shape[1] == 1:
        only_response = category_counts.columns[0]
        if only_response == 0:
            category_counts = category_counts.rename(columns={0: 'No'})
            category_counts['Yes'] = 0
        elif only_response == 1:
            category_counts = category_counts.rename(columns={1: 'Yes'})
            category_counts['No'] = 0

    fig = px.bar(
```

```
        category_counts,
        barmode='stack',
        title="Online Purchases Made During the Reference Period",
        labels={"value": "Number of Households", "index": "Category"},
        color_discrete_sequence=px.colors.qualitative.Set3
    )
    fig.update_layout(xaxis_tickangle=-45)
    st.plotly_chart(fig, use_container_width=True)

elif viz_option == "Payment Method Distribution":
    st.title("Payment Method Distribution")
    payment_col = 'If_yes_in_Q4.2.9_Amount_?'
    if payment_col in df_filtered.columns:
        payment_counts = df_filtered[payment_col].value_counts()
        fig = px.pie(
            names=payment_counts.index,
            values=payment_counts.values,
            title="Distribution of Online Payment Methods",
            color_discrete_sequence=px.colors.sequential.Blues
        )
        st.plotly_chart(fig, use_container_width=True)

elif viz_option == "Households by Sector":
    st.title("Households by Sector")
    if 'Sector' in df_filtered.columns:
        sector_counts = df_filtered['Sector'].value_counts().reset_index()
        sector_counts.columns = ['Sector', 'Count']
        fig = px.bar(
            sector_counts,
            x='Sector',
            y='Count',
            title="Households by Sector (Urban vs Rural)",
            color='Sector',
            color_discrete_sequence=px.colors.qualitative.Pastel
        )
        st.plotly_chart(fig, use_container_width=True)

elif viz_option == "Online Purchases Over Time":
    st.title("Online Purchases Over Time")
    if 'Year' in df.columns:
        time_trend = df.groupby('Year').size().reset_index(name='Purchases')
        fig = px.line(
            time_trend,
            x='Year',
            y='Purchases',
            markers=True,
            title="Trend of Online Purchases Over the Years",
            color_discrete_sequence=['#17becf']
```

```
        )
        st.plotly_chart(fig, use_container_width=True)

elif viz_option == "Correlation Heatmap":
    st.title("Correlation Between Purchase Categories")
    df_corr = df_filtered.copy()
    for col in online_purchase_categories:
        if col in df_corr.columns:
            df_corr[col] = pd.to_numeric(df_corr[col].map({'Yes': 1, 'No': 0}),
errors='coerce')
    corr_matrix = df_corr[online_purchase_categories].corr()
    fig, ax = plt.subplots(figsize=(10, 8))
    sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt='.2f', ax=ax)
    st.pyplot(fig)

st.sidebar.markdown("---")
st.sidebar.markdown("Download Cleaned Data")
st.sidebar.download_button(
    label="Download CSV",
    data=df.to_csv(index=False).encode('utf-8'),
    file_name="online_purchase_data.csv",
    mime="text/csv"
)

st.sidebar.text("Created by Abhishek Sharma")

```