Reall Household Income and Affordability Calculator

Methodology Paper: Nigeria 2019

This paper lays out the methodology used to develop the 2019 Nigeria data for Reall's Household Income and Affordability Calculator, which can be found at www.reall.net/calculator.

The Nigeria Living Standards Survey (NLSS) is produced by the Nigeria Bureau of Statistics (NBS). NLSS is a nationally-representative household survey focused on presenting income and consumption expenditure data, with the current round covering 22,110 households. The full published report and tables provide a broad overview of the landscape, with the raw data freely available for more detailed analysis.[1]

The NLSS focuses on consumption expenditure, so this is used as a proxy for income. This is generally seen as good practice, providing more reliable responses from participants. However, it may also result in slightly lower figures than would have been collected for income, particularly for wealthier households.

Data is split into separate data files for different sections of the survey. Key sections of the NLSS survey used for this work are 'totcons', which provides aggregated consumption data per household, and 'sect4a1_labour', which provides data on number of income earners per household. Names and coding for Nigeria's states and local government authorities can be found on the Data Description tab under 'totcons', [state] and [lga]. [2]

Methodology

The following actions were undertaken on these datasets:

Creating Dataset

- state_codes
    - Create new 'state_codes' dataset using state names from the survey webpage's Data Description tab (see above) – [state] and [state_value]
- lga_codes
    - Create new 'lga_codes' dataset using local government authority names from the survey webpage's Data Description tab (see above) – [lga] and [lga_value]
- sect4a1_labour
    - Replace blank rows in following columns with 2 ('No'):
        - [s04aq04] ('During the past 7 days, has [NAME] worked … in a wage job')
        - [s04aq06] ('During the past 7 days, has [NAME] worked … in own agriculture')

[1] Data dictionary, reports and raw data accessible at https://microdata.worldbank.org/index.php/catalog/3827/
[2] Data dictionary, reports and raw data accessible at https://microdata.worldbank.org/index.php/catalog/3827/

        - [s04aq09] ('During the past 7 days, has [NAME] worked … in own non-farm enterprise')
        - [s04aq48] ('… did [NAME] do any wage/salaried job during the last 12 months')
    - Replace blank rows with 3 in [s04aq11] ('During the past 7 days, has [NAME} worked as a trainee or apprenticeship')
    - Create new [is_earner] column, which has a value of 1 if [s04aq04] is 1, or [s04aq06] is 1, or [s04aq09] is 1, or [s04aq11] is 1 or 2, or [s04aq48] is 1
    - Group all data by [hhid] (household ID), summing number of [is_earner] household members in a new [no_of_earners] column
- totcons
    - Create new [household_monthly] column where [totcons_pc] (total consumption per capita) is divided by 12 and then multiplied by [hhsize] (household size)
    - Convert [sector] to new [urban_rural] column, where 1 is 'urban' and 2 is 'rural'
    - Merge [no_of_earners] from 'sect4a1_labour' by [hhid]
    - Merge [state_value] from 'state_codes' by [state]
    - Merge [lga_values] from 'lga_codes' by [lga]

```
# Calculate the number of earners per household
sec4a_dataset <- sec4a_dataset %>%
        mutate(
                s04aq04 = replace_na(s04aq04, 2),
                s04aq48 = replace_na(s04aq48, 2),
                s04aq06 = replace_na(s04aq06, 2),
                s04aq09 = replace_na(s04aq09, 2),
                s04aq11 = replace_na(s04aq11, 3)
                ) %>%
        mutate(is_earner = ifelse(s04aq04 == 1 | s04aq06 == 1 | s04aq09 == 1 | s04aq11 == 1 |
        s04aq11 == 2 | s04aq48 == 1, 1, 0)) %>%
        group_by(hhid) %>%
        mutate(no_of_earners = sum(is_earner)) %>%
        ungroup()

# Prepare total consumption data
totcons_summary <- totcons_dataset %>%
        mutate(
                Household_monthly = (totcons_pc / 12) * hhsize,
                Percent_Income_Spent_on_Housing = (rent33 / totcons_pc) * 100,
                urban_rural = ifelse(sector == 1, "Urban", "Rural")
                ) %>%
        left_join(sec4a_dataset[c('hhid', 'no_of_earners')], by = "hhid") %>%
        left_join(state_codes, by = "state") %>%
        left_join(lga_codes, by = "lga") %>%
        distinct()
```

Calculating Sample Sizes

- Group dataset by [lga_value] and [state_value] to create sample sizes for each district
- Group dataset by [lga_value], [state_value] and [urban_rural] to create separate urban and rural sample sizes for each district

```
sample_size_nigeria <- totcons_summary %>%
        group_by(state_value, urban_rural,lga_value, Year = 2019) %>%
        summarise(sample_size = n())

sample_size_nigeria_u_r <- totcons_summary %>%
        group_by(state_value, urban_rural = "All",lga_value, Year = 2019) %>%
        summarise(sample_size = n())
```

Calculating Percentiles

- Multiply each [hhid] record by [weights] to create full weighted dataset
- Group data by [lga_value] and [state_value], and extracting data for records at 1% increments, creating figures for each percentile of every district
- Repeat the step above but also grouping by [urban_rural], creating separate urban and rural percentile figures for each district

```
# Function to calculate quantiles and related statistics
calculate_quantiles <- function(data, quantiles) {
        do.call(rbind, lapply(quantiles, function(q) {
                data.frame(
                        Quantile = q * 100,  # Convert quantile probability to percentage
                        HH_exp = quantile(data$HH_exp, probs = q, na.rm = TRUE)
                        )
                }))
        }

# Calculate quantiles for each location
all_data_nigeria <- totcons_summary %>%
        uncount(weights = as.integer(wt_final / 100)) %>%
        rename(HH_exp = Household_monthly, HH_size = hhsize)

nigeria_location_quantiles <- all_data_nigeria %>%
        group_by(Country = "Nigeria", urban_rural, Location = state_value,City = lga_value
        ,Year = 2019) %>%
        group_modify(~ calculate_quantiles(.x, quantile_probs)) %>%
        ungroup()

# Additional mixed urban/rural quantiles for each location
quantile_probs <- seq(0.01, 0.99, by = 0.01)
nigeria_location_quantiles_u_r <- all_data_nigeria %>%
        group_by(Country = "Nigeria", urban_rural = "All", Location = state_value, City =
        lga_value, Year = 2019) %>%
```

```
    group_modify(~ calculate_quantiles(.x, quantile_probs)) %>%
    ungroup()
```

Creating Final Dataset

- Define and align common columns to enable merging of quantiles datasets
- Combine relevant 'quantiles' and 'quantiles_u_r' datasets into a single 'summary_dataset'
- Define and align common columns to enable merging of sample size datasets
- Combine relevant 'sample_size' datasets into a single 'combined_sample_size'
- Join 'combined_sample_size' dataset to 'summary_dataset'
- Create a 'state_aggregated' version of 'summary_dataset' by grouping all household data by state
- Create a 'national_aggregated' version of 'summary_dataset' by grouping all household data by country
- Join 'state_aggregated' and 'national_aggregated' datasets to 'summary_dataset'

```
# Define common columns for final summary
common_columns <- c("Country", "urban_rural", "Location", "City", "Year", "Quantile",
"HH_exp", "no_of_earners", "HH_size", "Percent_Income_Spent_on_Housing")

# Function to align columns across datasets
align_columns <- function(df, common_cols) {
    df %>%
    mutate(across(setdiff(common_cols, colnames(df)), ~ NA)) %>%
    select(all_of(common_cols))
    }

# Align columns and combine all datasets
Nigeria_location_quantiles <- align_columns(Nigeria_location_quantiles, common_columns)
Nigeria_location_quantiles_u_r <- align_columns(Nigeria_location_quantiles_u_r,
common_columns)

# Combine all country datasets into a single summary dataset
summary_dataset <- bind_rows(Nigeria_location_quantiles, Nigeria_location_quantiles_u_r)

sample_size_Nigeria <- sample_size_Nigeria %>%
    rename(Location = state, sample_size = sample_size) %>%
    mutate(Country = "Nigeria")

sample_size_Nigeria_u_r <- sample_size_Nigeria_u_r %>%
    rename(Location = state, sample_size = sample_size) %>%
    mutate(Country = "Nigeria")

# Combine the sample size tables into one
combined_sample_size <- bind_rows(
    sample_size_Nigeria,
    sample_size_Nigeria_u_r,
```

```
summary_dataset <- summary_dataset %>%
        left_join(combined_sample_size, by = c ("Country", "urban_rural", "Location", "City",
        "Year"))

# Aggregate data for state level
state_aggregated <- summary_dataset %>%
        group_by(Country, Location, Year, Quantile,urban_rural) %>%
        summarise(
                sample_size = sum(sample_size),
                HH_exp = mean(HH_exp, na.rm = TRUE),
                no_of_earners = mean(no_of_earners, na.rm = TRUE),
                HH_size = mean(HH_size, na.rm = TRUE),
                ) %>%
        ungroup()%>%
        mutate(City = "All")

# Aggregate data for the national level by combining all states within each country
national_aggregated <- summary_dataset %>%
        group_by(Country, Year, Quantile,urban_rural) %>%
        summarise(
                sample_size = sum(sample_size),
                HH_exp = mean(HH_exp, na.rm = TRUE),
                no_of_earners = mean(no_of_earners, na.rm = TRUE),
                HH_size = mean(HH_size, na.rm = TRUE)
                ) %>%
        ungroup() %>%
        mutate(Location = "All", City = "All")

# Combine both the state-level and national-level data
final_dataset <- bind_rows(state_aggregated, national_aggregated,summary_dataset)
```

Calculating Inflation

All data is inflated using median annual inflation rates from 2010-23. Median rates were used rather than actual figures to help compensate for large-scale inflation across many economies in 2022 and 2023.

Consumer Price Index inflation rates were taken from the World Bank[3] and consisted of the following figures, creating a final median rate of 12.73%.

CPI	Nigeria
2000	6.933292
2001	18.87365
2002	12.87658
2003	14.03178
2004	14.99803

[3] https://data.worldbank.org/indicator/FP.CPI.TOTL.ZG

2005	17.86349
2006	8.225222
2007	5.388008
2008	11.58108
2009	12.55496
2010	13.7202
2011	10.84003
2012	12.21778
2013	8.475827
2014	8.062486
2015	9.009387
2016	15.67534
2017	16.52354
2018	12.09473
2019	11.39679
2020	13.24602
2021	16.95285
2022	18.84719
2023	24.65955