

Baseline long survey

Abigail

2025-01-20

Installing packages and calling out libraries

Importing data set

The data was exported from CommCare on 23/12/2024.

```
Mangwana_Baseline_Survey <- read_excel("Mangwana - Baseline Survey.xlsx",  
                                         sheet = "Forms")
```

Selecting indicators of interest

A sub set of the indicators were selected depending on the goal of the calculation and the indicators needed.

Number of people with diverse and adequate diet

I made a separate calculation for the child and the adult.

```
Selected_Child_indicators <- Mangwana_Baseline_Survey %>%  
  select(form.Section_II_group.voc_tem_um_filho_entre_0_e_24_meses, form.Section_II_group.llquidos_formula_infantil,  
         form.Section_II_group.sumos_ou_sumos_em_p, `form.Section_II_group.refrescos_como_coca-cola_fanta_frozy_ou_fizz_ou_bebidas_ervas`,  
         form.Section_II_group.xima_ou_papas_de_milho_no_processado_mexoeira_mapira_maaroca_po_integral,  
         form.Section_II_group.tomate_repolho_quiabo_beringela_ou_cogumelos, form.Section_II_group.alfafa,  
         form.Section_II_group.ovos, form.Section_II_group.queijo, form.Section_II_group.paloni_salsicha,  
         form.Section_II_group.amendoim_manteiga_de_amendoim_amendoim_modo_castanhas_de_caj_ou_sementes,  
  
         rename(have_child = form.Section_II_group.voc_tem_um_filho_entre_0_e_24_meses,  
                infant_formula = form.Section_II_group.llquidos_formula_infantil,  
                animal_milk = form.Section_II_group.leite_de_origem_animal_incluindo_lquido_ou_em_p,  
                chocolate_milk = form.Section_II_group.achocolatado_como_milo_cremora_ou_leite_condensado,  
                juices = form.Section_II_group.sumos_ou_sumos_em_p,  
                soft_drinks = `form.Section_II_group.refrescos_como_coca-cola_fanta_frozy_ou_fizz_ou_bebidas_ervas`,  
                plant_base_drink = form.Section_II_group.ch_caf_ou_bebidas__base_de plantas_ou_ervas,  
                soups_broth = form.Section_II_group.sopa_ou_caldo,  
                yoghurt = form.Section_II_group.iogurte,  
                processed_porridge = form.Section_II_group.arroz_po_massa_espaguete_xima_de_farinha_de_milho,  
                green_leafy_vegetables = form.Section_II_group.bruculos_espinafres_folhas_de_moringa_folhas_de_morango,  
                shrimp_snails = form.Section_II_group.camaro_voador_caracol_da_terra_trmita_ou_ishwa, # Flying Snails)
```

```

unprocessed_porridge = form.Section_II_group.xima_ou_papas_de_milho_no_processado_mexoeira_mapa
root_crop_porridge = form.Section_II_group.batata_batata_doce_branca_inhame_mandioca_xima_de_ma
beans = form.Section_II_group.feijo_comum_feijo_nhamba_ervilha_feijo_boer_feijo_jugo_feijo_sorv
orange_vegetables = form.Section_II_group.cenoura_abbora_ou_batata_doce_de_polpa_alaranjada,#
leafy_vegetables = form.Section_II_group.couve_folhas_de_abobora_folhas_de_cacana_folhas_de_ba
tomato_cabbage_mushroom = form.Section_II_group.tomate_repolho_quiabo_beringela_ou_cogumelos,
salad_vegetables = form.Section_II_group.alface_pepino_feijo_verde_ou_pimento_verde, # Lettuce
yellow_fruits = form.Section_II_group.manga_madura_papaia_madura_ou_maracuj, # Ripe mango, rip
citrus = form.Section_II_group.laranja_ou_tangerinas,
long_cycle_fruits = form.Section_II_group.bananas_melancia_abacate_lichia_caju_anans_ou_malamb
apples = form.Section_II_group.ma_pra_goiaba_massala_ata_mapfilo_tintsiva_ou_mafurra, # Apple,
cakes = form.Section_II_group.bolos_bolinhas_bolachas_doces_argolas_fritas_bolas_de_berlim_ou_
eggs = form.Section_II_group.ovos,
cheese = form.Section_II_group.queijo,
sausages = form.Section_II_group.paloni_salsichas_ou_enchidos_como_chourio_lingua_rachel_russ
meat = form.Section_II_group.carne_de_vaca_carne_de_cabrito_carne_de_ovelha_gazela_ou_bufalo,
pork_rabbit = form.Section_II_group.carne_de_porco_carne_de_ratazana_ou_coelho, # Pork, rat or
poultry = form.Section_II_group.galinha_pato_peru_codorniz_ou_passarinhos, # Chicken, duck, tu
fish = form.Section_II_group.peixe_fresco_ou_seco_camaro_fresco_ou_seco_caranguejo_lulas_ou_am
nuts = form.Section_II_group.amendoim_manteiga_de_amendoim_amendoim_modos_castanhas_de_caj_ou_s
popcorns = form.Section_II_group.chips_pipocas_de_pacote_ou_niknaks, # Chips, popcorn or NikNa
instant_pasta = form.Section_II_group.massa_instantanea,
fries = form.Section_II_group.batatas_fritas_badjia_peixe_frito_ou_frango_frito, # Deep fried
palm_oil = form.Section_II_group.leo_de_palma_vermelho,
fast_foods = `form.Section_II_group.copy-1-of-leo_de_palma_vermelho`, # KFC, Debonairs Pizza,
sweets = form.Section_II_group.chupa_chupa_ou_doces_bombom_chocolate_sorvete_ou_gelinho) %>%

mutate(
  across(c(have_child,infant_formula,animal_milk,chocolate_milk,juices,soft_drinks,plant_base_drink,s
    beans, orange_vegetables,leafy_vegetables,green_leafy_vegetables,tomato_cabbage_mushroom,s
    fish,nuts,popcorns,instant_pasta,fries,palm_oil,fast_foods,sweets), ~ ifelse(. == "sim", 1
) %>%

mutate(

mdd_w = (rowSums(tibble(processed_porridge, unprocessed_porridge, root_crop_porridge) == 1) > 0) +
  (rowSums(tibble(bean == 1) > 0) + # Pulses
  (rowSums(tibble(nuts == 1) > 0) + # Nuts and seeds
  (rowSums(tibble(animal_milk, cheese, yoghurt) == 1) > 0) + # Dairy
  (rowSums(tibble(meat, sausages, pork_rabbit, poultry, fish, shrimp_snails) == 1) > 0) + # Meat, p
  (rowSums(tibble(green_leafy_vegetables, leafy_vegetables) == 1) > 0) + # Dark green leafy vegetab
  (rowSums(tibble(eggs) == 1) > 0) + # Eggs
  (rowSums(tibble(orange_vegetables, tomato_cabbage_mushroom, yellow_fruits, citrus, apples) == 1) > 0) +
  (rowSums(tibble(long_cycle_fruits) == 1) > 0),# Other fruits
all_5 = (rowSums(tibble(processed_porridge, unprocessed_porridge, root_crop_porridge) == 1) > 0) +
  (rowSums(tibble(green_leafy_vegetables, leafy_vegetables, orange_vegetables, tomato_cabbage_mushr
  (rowSums(tibble(yellow_fruits, citrus,long_cycle_fruits, apples) == 1) > 0) + # Fruits
  (rowSums(tibble(bean,nuts) == 1) > 0) + # Pulses, nuts and seeds
  (rowSums(tibble(animal_milk, yoghurt, eggs, cheese, sausages, meat, poultry, fish, shrimp_snails)
ncd_protect = (rowSums(tibble(unprocessed_porridge) == 1) > 0) + # Whole grain
  (rowSums(tibble(bean == 1) > 0) + # Pulses
  (rowSums(tibble(nuts == 1) > 0) + # Nuts and seeds
  (rowSums(tibble(citrus) == 1) > 0) + # Citrus

```

```

    (rowSums(tibble(orange_vegetables) == 1) > 0) + # Vitamin A-rich orange vegetable
    (rowSums(tibble(green_leafy_vegetables, leafy_vegetables) == 1) > 0) + # Dark green leafy vegetable
    (rowSums(tibble(apples, yellow_fruits) == 1) > 0) + # Vitamin A-rich fruits
    (rowSums(tibble(tomato_cabbage_mushroom) == 1) > 0) + # Other vegetables
    (rowSums(tibble(long_cycle_fruits) == 1) > 0), # Other fruits
  ncd_risk = (rowSums(tibble(chocolate_milk, juices, soft_drinks) == 1) > 0) + # Represents: Soft drinks
    (rowSums(tibble(cakes) == 1) > 0) + # Baked/grain-based sweets
    (rowSums(tibble(sweets) == 1) > 0) + # Other sweets
  2 * (rowSums(tibble(sausages) == 1) > 0) + # Processed meat
    (rowSums(tibble(meat, pork_rabbit) == 1) > 0) + # Unprocessed meat
    (rowSums(tibble(fries) == 1) > 0) + # Deep fried food
    (rowSums(tibble(fast_foods, instant_pasta) == 1) > 0) + # Fast food & instant noodles
    (rowSums(tibble(popcorns) == 1) > 0), # Packaged ultra-processed salty snacks

  mdd_w_binary = ifelse(mdd_w >= 5, 1, 0), # Calculating binary MDD_W

  all_5_binary = ifelse(all_5 == 5, 1, 0), # Calculating binary All- 5

  gdr_score = (ncd_protect - ncd_risk) + 9, # Calculating General Dietary Recommendation (GDR score)

)

```

Selecting indicators for child's diverse and adequate diet

```
head(Selected_Child_indicators)
```

Displaying updated data and calculation

```

## # A tibble: 6 x 45
##   have_child infant_formula animal_milk chocolate_milk juices soft_drinks
##   <dbl>      <dbl>      <dbl>      <dbl> <dbl>      <dbl>
## 1         0         0         0         0     0         0
## 2         1         0         0         0     0         0
## 3         0         0         0         0     0         0
## 4         1         0         0         0     0         0
## 5         1         0         0         0     0         0
## 6         0         0         0         0     0         0
## # i 39 more variables: plant_base_drink <dbl>, soups_broth <dbl>,
## #   yoghurt <dbl>, processed_porridge <dbl>, green_leafy_vegetables <dbl>,
## #   shrimp_snails <dbl>, unprocessed_porridge <dbl>, root_crop_porridge <dbl>,
## #   beans <dbl>, orange_vegetables <dbl>, leafy_vegetables <dbl>,
## #   tomato_cabbage_mushroom <dbl>, salad_vegetables <dbl>, yellow_fruits <dbl>,
## #   citrus <dbl>, long_cycle_fruits <dbl>, apples <dbl>, cakes <dbl>,
## #   eggs <dbl>, cheese <dbl>, sausages <dbl>, meat <dbl>, ...

```

```

mean_mdd_w_score <- round(mean(Selected_Child_indicators$mdd_w), 1) # Calculating the average MDD-W score
print(mean_mdd_w_score)

```

```
## [1] 1
```

```
mean_all_5_score <- round(mean(Selected_Child_indicators$all_5),1) # Calculating the average All 5 score
print(mean_all_5_score)
```

```
## [1] 0.9
```

```
mean_gdr_score <- round(mean(Selected_Child_indicators$gdr_score),1) # Calculating the average GDR score
print(mean_gdr_score)
```

```
## [1] 9.7
```

Selecting indicators to calculate adult diverse and adequate diet

Note: I did not find any adult data in the Excel file downloaded or ODataFeed. I tried answering survey questions but did not see the questions in the mobile App. Therefore questions under this section “ler__agora_gostaria_de_lhe_fazer_algunas_perguntas_de_sim_ou_no_sobre_os_al” do not appear in the survey. I confirmed it by filling a form myself. After the “KFC, Debonairs Pizza, Teka Famba, Burger House, Flaming Dogs or other places that serve pizza or burgers?” question, the survey moves to Section III.

Selecting indicators for Resilience to shocks

```
Resilience <- Mangwana_Baseline_Survey %>%
  select(form.Section_III_group.missed_food, form.Section_III_group.Months_missed_food, form.Section_III_group.how_many_times_HH_not_have_enough_food,
         form.Section_III_group.not_enough_food,form.Section_III_group.how_many_times_not_enough_food,
         form.Section_III_group.hunger_scores)

mutate(
  months_adequate_food = 12 - apply(strsplit(ifelse(form.Section_III_group.Months_missed_food == "--",
                                                    form.Section_III_group.how_many_times_HH_not_have_enough_food, form.Section_III_group.how_many_times_not_enough_food),
                                     " ",
                                     FUN = function(x) as.numeric(x)),
                                     MARGIN = 2,
                                     FUN = function(x) sum(x != 0)),
  hhs_score = form.Section_III_group.how_many_times_HH_not_have_enough_food + form.Section_III_group.how_many_times_not_enough_food,
  hunger_scores = case_when(hhs_score <= 1 ~ "Little to no hunger in the household",
                             hhs_score <= 3 ~ "Moderate hunger in the household",
                             hhs_score <= 6 ~ "Severe hunger in the household") # Classified the household into three categories of hunger
)
```

```
Average_months_of_adequate_food <- round(mean(Resilience$months_adequate_food),1) # Average months household has adequate food
print(Average_months_of_adequate_food)
```

```
## [1] 10.7
```

```
Median_hhs_score <- median(Resilience$hhs_score) # Median value of HHS score
print(Median_hhs_score)
```

```
## [1] 0
```

```
head(Resilience)
```

Displaying updated data and calculation

```
## # A tibble: 6 x 11
##   form.Section_III_group.missed_~1 form.Section_III_gro~2 form.Section_III_gro~3
##   <chr>                                <chr>                                <chr>
## 1 yes                                august                                yes
## 2 yes                                october                               yes
## 3 yes                                september                             yes
## 4 no                                  ---                                no
## 5 yes                                september                             yes
## 6 no                                  ---                                no
## # i abbreviated names: 1: form.Section_III_group.missed_food,
## #   2: form.Section_III_group.Months_missed_food,
## #   3: form.Section_III_group.you_feared_HH_not_have_enough_food
## # i 8 more variables:
## #   form.Section_III_group.how_may_times_HH_not_have_enough_food <dbl>,
## #   form.Section_III_group.not_enough_food <chr>,
## #   form.Section_III_group.how_many_times_not_enough_food <dbl>, ...
```

```
print(Average_months_of_adequate_food)
```

```
## [1] 10.7
```

```
print(Median_hhs_score)
```

```
## [1] 0
```

Agriculture indicators

```
## Merging the general data sheet and the repeat agriculture sheet to filter to baseline data
```

```
## Importing dry season dataset
```

```
Drought_repeat_Survey <- read_excel("Mangwana - Baseline Survey.xlsx",
                                     sheet = "Repeat- crop_drought_season")
```

```
Mangwana_Baseline_Survey <- read_excel("Mangwana - Baseline Survey.xlsx",
                                         sheet = "Forms")
```

```
## Constant variables
```

```
number_dryseason_weeks <- 23
```

```
Standard_workhours <- 8
```

```
PPP <- 24.95 ## Current (2025) PPP. Source: https://www.imf.org/external/datamapper/PPPEX@WEO/MOZ?zoom=1
```

Data manipulation

```
Baseline_drought <- Drought_repeat_Survey %>%
  mutate(number = as.character(number__0)) %>%

  inner_join(Mangwana_Baseline_Survey %>% select(number,form.Section_IV_ProductionAreas.que_tipo_de_inq

  filter(form.Section_IV_ProductionAreas.que_tipo_de_inquirito_est_a_realizar == "inquirito_de_base_para_

  mutate(across(everything(), ~ case_when(. == "---" ~ "0",
                                           . == " " ~ "0",
                                           is.na(.) ~ "0",
                                           TRUE ~ as.character(.)))) %>%

  rename(qty_crops_produced = form.Section_IV_ProductionAreas.Section_IV_I_DroughtSeason.crop_drought_s
        days_per_week_fieldwork = `form.Section_IV_ProductionAreas.Section_IV_I_DroughtSeason.crop_dro
        hours_per_day_fieldwork = `form.Section_IV_ProductionAreas.Section_IV_I_DroughtSeason.crop_dro
        assist_days_perweek_fieldwork = form.Section_IV_ProductionAreas.Section_IV_I_DroughtSeason.crop
        assist_hoursper_fieldwork = form.Section_IV_ProductionAreas.Section_IV_I_DroughtSeason.crop_dro
        paidlabour_days_perweek_fieldwork = form.Section_IV_ProductionAreas.Section_IV_I_DroughtSeason
        paidlabour_hours_per_fieldwork = form.Section_IV_ProductionAreas.Section_IV_I_DroughtSeason.cro
        price_per_unit_sold = form.Section_IV_ProductionAreas.Section_IV_I_DroughtSeason.crop_drought_
        seed_expense = form.Section_IV_ProductionAreas.Section_IV_I_DroughtSeason.crop_drought_season.
        equipment_expense = form.Section_IV_ProductionAreas.Section_IV_I_DroughtSeason.crop_drought_se
        fert_expense = form.Section_IV_ProductionAreas.Section_IV_I_DroughtSeason.crop_drought_season.
        labour_expense = form.Section_IV_ProductionAreas.Section_IV_I_DroughtSeason.crop_drought_season
        other_expense = form.Section_IV_ProductionAreas.Section_IV_I_DroughtSeason.crop_drought_season
        pesticide_expense = form.Section_IV_ProductionAreas.Section_IV_I_DroughtSeason.crop_drought_se

  mutate(across(c(qty_crops_produced, days_per_week_fieldwork, hours_per_day_fieldwork, assist_days_per
        paidlabour_hours_per_fieldwork,price_per_unit_sold, seed_expense,equipment_expense, fert

## Calculating Labour days and productivity

mutate(total_weeks_fieldwork = rowSums(across(c(days_per_week_fieldwork,assist_days_perweek_fieldwork
  personallabour_time= days_per_week_fieldwork*hours_per_day_fieldwork,
  assistedlabour_time = assist_days_perweek_fieldwork * assist_hoursper_fieldwork,
  paidlabour_time = paidlabour_days_perweek_fieldwork * paidlabour_hours_per_fieldwork,
  personallabour_days = (personallabour_time*number_dryseason_weeks)/Standard_workhours,
  assistedlabour_days = (assistedlabour_time*number_dryseason_weeks)/Standard_workhours,
  paidlabour_days = (paidlabour_time*number_dryseason_weeks)/Standard_workhours,
  total_labour_days = rowSums(across(c(personallabour_days, assistedlabour_days, paidlabour_days)
  crops_income = qty_crops_produced*price_per_unit_sold,
  productivity = crops_income/total_labour_days,
  total_expense = rowSums(across(c(seed_expense,pesticide_expense,fert_expense,equipment_expense)
  gross_agric_income = crops_income - total_expense)

## Productivity

total_crop_income <-sum(Baseline_drought$crops_income)
print(total_crop_income)
```

```
## [1] 8592384
```

```
## Agriculture gross income
```

```
agric_gross <- sum(Baseline_drought$gross_agric_income)
agric_gross
```

```
## [1] 7667173
```

Agroforestry indicators

```
## Calculating agroforestry indicators
```

```
## Importing data
```

```
Agroforestry_repeat_Survey <- read_excel("Mangwana - Baseline Survey.xlsx",
                                          sheet = "Repeat- product_silviculture")
```

```
## Selecting a subset of data
```

```
Baseline_drought_agroforestry <- Agroforestry_repeat_Survey %>%
```

```
  mutate(number = as.character(number__0)) %>%
```

```
  inner_join(Mangwana_Baseline_Survey %>% select(number,form.Section_IV_ProductionAreas.que_tipo_de_in
```

```
  filter(form.Section_IV_ProductionAreas.que_tipo_de_inquirito_est_a_realizar == "inquirito_de_base_para_
```

```
  mutate(across(everything(), ~ case_when(. == "---" ~ "0",
                                           . == " " ~ "0",
                                           is.na(.) ~ "0",
                                           TRUE ~ as.character(.)))) %>%
```

```
  rename(agro_sale_price = form.Section_IV_ProductionAreas.Section_IV_III_Silviculture.product_silvicult
          agro_qty_sold = form.Section_IV_ProductionAreas.Section_IV_III_Silviculture.product_silvicultu
          seed_cost = form.Section_IV_ProductionAreas.Section_IV_III_Silviculture.product_silviculture.e
          pesticide_cost = form.Section_IV_ProductionAreas.Section_IV_III_Silviculture.product_silvicult
          fert_cost = form.Section_IV_ProductionAreas.Section_IV_III_Silviculture.product_silviculture.e
          labour_cost = form.Section_IV_ProductionAreas.Section_IV_III_Silviculture.product_silviculture
          equipment_expense = form.Section_IV_ProductionAreas.Section_IV_III_Silviculture.product_silvic
          other_expense = form.Section_IV_ProductionAreas.Section_IV_III_Silviculture.product_silvicultu
```

```
  mutate(across(c(agro_sale_price,agro_qty_sold,seed_cost,pesticide_cost,fert_cost,labour_cost,equipmen
```

```
  mutate(crop_income = agro_sale_price * agro_qty_sold,
          total_expenses = rowSums(across(c(seed_cost,pesticide_cost,fert_cost,labour_cost,equipment_exp
          gross_income = crop_income - total_expenses)
```

```
## Agroforestry gross income
```

```
agroforestry_gross <- sum(Baseline_drought_agroforestry$gross_income)
agroforestry_gross
```

```
## [1] -1750
```


Livestock indicators

```
## Importing data
```

```
Animal_repeat_Survey <- read_excel("Mangwana - Baseline Survey.xlsx",  
                                   sheet = "Repeat- Animals_Raised")
```

```
Baseline_drought_animals <- Animal_repeat_Survey %>%  
  mutate(number = as.character(number__0)) %>%
```

```
  inner_join(Mangwana_Baseline_Survey %>% select(number,form.Section_IV_ProductionAreas.que_tipo_de_in
```

```
  filter(form.Section_IV_ProductionAreas.que_tipo_de_inquirito_est_a_realizar == "inquirito_de_base_para_
```

```
  mutate(across(everything(), ~ case_when(. == "---" ~ "0",  
                                           . == " " ~ "0",  
                                           is.na(.) ~ "0",  
                                           TRUE ~ as.character(.)))) %>%
```

```
  rename(qty_sold = form.Section_IV_ProductionAreas.Section_IV_IV_Livestock.Animals_Raised.qty_sale_liv  
         price_livestock = form.Section_IV_ProductionAreas.Section_IV_IV_Livestock.Animals_Raised.price  
         feeding_expense = form.Section_IV_ProductionAreas.Section_IV_IV_Livestock.Animals_Raised.expense  
         labour_expense = form.Section_IV_ProductionAreas.Section_IV_IV_Livestock.Animals_Raised.expense  
         medical_expense = form.Section_IV_ProductionAreas.Section_IV_IV_Livestock.Animals_Raised.expense  
         equipment_expense = form.Section_IV_ProductionAreas.Section_IV_IV_Livestock.Animals_Raised.exp  
         maintain_expense = form.Section_IV_ProductionAreas.Section_IV_IV_Livestock.Animals_Raised.exp
```

```
  mutate(across(c(qty_sold,price_livestock,feeding_expense, labour_expense, medical_expense, equipment_
```

```
  mutate(animal_income = qty_sold * price_livestock,  
         total_expense = rowSums(across(c(feeding_expense,labour_expense, medical_expense, equipment_exp  
         gross_animal_income = animal_income - total_expense)
```

```
gross_animal <- sum(Baseline_drought_animals$gross_animal_income)  
gross_animal
```

```
## [1] -172886
```