In [1]:

```
# ==============================================================================
# Village Income and Wealth July 2023
# ==============================================================================
# INCLUDES ============
# agriculture: output (in kg and monetary value) and inputs (including land, labor, 
# non-agric income: labor, ganyu, business, other.
# Wealth: farming capital, hh assets, etc.
# labor supply.
# shocks
# coupons, conditional cash transfer program.

# Checks and correction of the data:

# OUTPUT ==============
    # income_wealth_23_rainseas.csv
    # income_wealth_23_rainseas.csv
    # income_wealth_23_year.csv

## MISSING ===========

root_path = 'C:/Users/rodri/Dropbox/Malawi/SIEG2021 (1)/2023 July'
path_22 = 'C:/Users/rodri/Dropbox/Malawi/SIEG2021 (1)/2022 July/Data/Clean data/Phas
path_feb23 = 'C:/Users/rodri/Dropbox/Malawi/SIEG2021 (1)/2023 Feb/Data/Clean data/Ph
folder_fig = root_path+'Figures'
save=False

import numpy as np
import pandas as pd
import os
os.chdir(root_path+'/Code/Phase 3/Auxiliary files')
from data_functions_albert import remove_outliers, gini

import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")



# Set the working directory
os.chdir(root_path+'/Data/Clean data/Phase 3 - Consumption, Transfers, Income/Income

## Display  set-up
pd.options.display.float_format = '{:,.2f}'.format
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)


# July 1st 2023 MWK vs dollar (official)
dollar_MWK = 1052


# Import village 19 data
data22 = pd.read_csv(path_22+'income_wealth_22_rainseas.csv')


# ==============================================================================
# Import data
# ==============================================================================

data = pd.read_stata(root_path+"/Data/Raw data/[3]-SIEG-Consumption + Agriculture + 

data.rename(columns={'householdid':'hhid'}, inplace=True)
```

```
# roster
roster = pd.read_csv(root_path+"/Data/Clean data/Phase 1 - Roster/roster_july23.csv"
roster = roster[['hhid','intervieweename']]

# Check households in the data but not in the roster: None
merge_rost = data[['hhid','enumerator']].merge(roster,on='hhid', how='inner')
missing_roster = data[~data.hhid.isin(merge_rost.hhid)]

# Households in the roster but not in the data:
missing_data =  roster[~roster.hhid.isin(merge_rost.hhid)]

print('Check missing households')
print(missing_data[['hhid','intervieweename']].to_string(index=False))
print('No missing households')

## consumption prices in the village 2023
p_23 = pd.read_csv(root_path+'/Data/Clean data/Phase 3 - Consumption, Transfers, Inc

# Isa-Lsms prices old survey (for missing prices)
p_isalsms = pd.read_stata(root_path+'/Data/Clean data/Phase 3 - Consumption, Transfe
p_isalsms = p_isalsms.groupby(by=['crop_code']).median()




## Look at duplicates:
duplicates = pd.value_counts(data['hhid'])
print('================================================================')
print('These households are duplicate')
print('================================================================')
print(duplicates[duplicates>1])


# Check households in the data but not in the roster
merge_rost = data[['hhid','enumerator']].merge(roster,on='hhid', how='inner')
missing = data[~data.hhid.isin(merge_rost.hhid)]
print('all households interviewed are also in the roster')

# merge to get old hhids and be able to see panel
data = data.merge(roster[['hhid','intervieweename']], on='hhid', how='left')

percentiles = [ 0.1, .25, .5, .75, 0.9, 0.99]
list_crops = ['maize', 'groundnut', 'groundbean', 'sweetpotatoe', 'fingermillet', 's

# Rename some variables
data.rename(columns={'unitssoldpearlmillet2':'unitssoldpearlmilletout2'}, inplace=Tr
data.rename(columns={'unitssoldsoyabean2':'unitssoldsoyabeanout2'}, inplace=True)
data.rename(columns={'soldquantitygroundbeanin':'soldquantitygroundbeanin'}, inplace


## Remove 9999 observations=====================================
data.replace([9999, 9999.00], np.nan, inplace=True)


# ============================================================================
# Check Land
# ============================================================================
print('      ')
print('      ')
print('   LAND SIZE, VALUE, AND RIGHTS   ')
print('      ')
data.rename(columns={'areaallplot':'hh_area_plots','rentoutallplot':'hh_rentout_plot
```

```
data['hh_p_acre_plots']=  data['hh_value_plots'] / data['hh_area_plots']
data['hh_ratio_value_rent']  = data['hh_value_plots'] / data['hh_rentout_plots']


# ===============================================================================
# Check: Land area, rentout value, and land value (acres and MWK)
# ===============================================================================
sum_hhplots = data[['area_cultivated','hh_area_plots','hh_rentout_plots','hh_value_p
print('')
print('================================================================')
print('Check: Distribution land at household level')
print('================================================================')
print(sum_hhplots)

print('Outliers land?')
print(data.loc[data['hh_area_plots']>10,['hhid','hh_area_plots']])
print('Last year Augus confirmed hhid 1211 has 21 acres')
# - 1211: Augus confirmed has 21 acres.


# ===============================================================================
#  Summarize Land rights
# ===============================================================================

data[['rightsellland', 'rightbequeathplot', 'chiefpreventsell', 'chiefpreventbequeat
sum_landrights = (data[['rightsellland', 'rightbequeathplot', 'chiefpreventsell', 'c
print(sum_landrights)


del missing_roster, missing, missing_data, merge_rost,  sum_hhplots, sum_landrights


#%% ===============================================================================
# INPUTS: Capital and Livestock
# ===============================================================================
print('      ')
print('      ')
print('   FARM CAPITAL AND LIVESTOCK  ')
print('      ')


#Livestock
data['hhlivestock'] =0
for i in range(1,16):
    if (i==2) or (i==5):
        continue
    data['hhlivestock'] += (data['selllivstck_'+str(i)].replace(9999,np.nan)).fillna

# Farm Equipment
data['hhfarmequip'] =0
for i in range(1,15):
    data['hhfarmequip'] += (data['sellfrmeqp_'+str(i)].replace(9999,np.nan)).fillna(

# Farm Structure
data['hhfarmstruct'] =0
for i in range(1,10):
    data['hhfarmstruct'] += (data['sellfrmstrc_'+str(i)].replace(9999,np.nan)).filln



## farming capital
data['k_farm'] = data['hhfarmequip'].fillna(0)+data['hhfarmstruct'].fillna(0)

print('================================================================')
```

```
print('Check: Farm Capital Value (in $)')
print('===============================================================')
print((data[['k_farm','hhlivestock','hhfarmequip','hhfarmstruct']]/dollar_MWK).descr


outliers_kfarm = data.loc[(data['k_farm']>200*dollar_MWK) | (data['hhlivestock']>300


print('I checked the two households with extreme values. The reason of the high valu

del outliers_kfarm

# ============================================================================
#%% Convert agricultural outputs to kgs and MWK. total quantities reported
# ============================================================================

print('      ')
print('      ')
print('  AGRICULTURAL PRODUCTION  ')
print('      ')

# Import conversion rates
crop_unit = pd.read_csv("conversions/crop_conversions_kg.csv")
crop_unit.set_index('unit', inplace=True)


#Check units
tab_units = []
for crop in list_crops:
    unitscrop = pd.value_counts(data['unitstotal'+crop])
    tab_units.append(unitscrop)

tab_units = pd.DataFrame(tab_units)

print(tab_units.to_string())
print('We might want to remove some units for a next time.')
## unit 10 is other units: we have 4 cases of other units
data.loc[data['unitstotalpigeonpeas']==10,'otherunitspigeonpeas']
data.loc[data['unitstotalcotton']==10,'otherunitscotton']
data.loc[data['unitstotalcassava']==10,'otherunitscassava']

#Generate empty variables
for crop in list_crops:
    data['total_kg_'+crop] = np.nan
    data['sold_kg_'+crop] = np.nan
    data['sold_insiders_kg_'+crop] = np.nan
    data['store_kg_'+crop] = np.nan
    data['total2_kg_'+crop] =np.nan
    data['sold_bigger_total_'+crop] = 0
    data['store_bigger_total_'+crop] = 0
    data['soldstore_bigger_total_'+crop] = 0
    data['p_'+crop] = np.nan
    data['y_'+crop] = 0
    data['y_agric'] = 0
    data['sold_MWK_'+crop] = 0
    data['sold_agric'] = 0
    data['sold_insiders_MWK_'+crop] = 0
    data['sold_insiders_agric'] = 0
    data['store_MWK_'+crop] = 0
    data['store_agric'] = 0
    data[['unitstotal'+crop, 'unitssold'+crop, 'unitsstore'+crop]].replace(np.nan, 0


# ============================================================================
```

```
# Main Loop: Conversion to kgs for all crops and questions


### change guys that reported other units and there wasnt question other units appea

data.replace(np.nan, 0, inplace=True)
for i in range(len(data)):
    for crop in list_crops:
        data.iloc[i, data.columns.get_loc('total_kg_'+crop)] = data.iloc[i,data.colu
        data.iloc[i, data.columns.get_loc('sold_kg_'+crop)] = data.iloc[i,data.colum
        data.iloc[i, data.columns.get_loc('sold_insiders_kg_'+crop)] = data.iloc[i,d
        data.iloc[i, data.columns.get_loc('store_kg_'+crop)] = data.iloc[i,data.colu

for crop in list_crops:
    data['total2_kg_'+crop] = data['sold_kg_'+crop].fillna(0) +data['store_kg_'+crop


#Summary total output kg:
pd.options.display.float_format = '{:,.0f}'.format
sum_kg = (data[['total_kg_maize', 'total_kg_groundnut', 'total_kg_groundbean', 'tota


## NON-PRODUCED CROPS
# pearl millet

# tomatoes: 1 hh, therereokra: 1 hh, tanaposi: 2 hh. Strange?
print('================================================================')
print('Crop production: Number of households harvested crops')
print('================================================================')
print((data[['total_kg_maize', 'total_kg_groundnut', 'total_kg_groundbean', 'total_k


print('================================================================')
print(' Distribution of crop production (in kg)')
print('================================================================')
sum_kg = sum_kg.dropna(axis=1, how='any')
N_prodcrops = sum_kg.iloc[0,:]
T_prodcrops = sum_kg.iloc[0,:]*sum_kg.iloc[1,:]
T_prod = T_prodcrops.sum()
print(sum_kg)
## STOP RUN

print('Check top maize producers:')
big_kg = data.loc[(data['total_kg_maize']>2000),[ 'hhid','total_kg_maize', 'area_cul

print(big_kg)
print('Not sure if 6100 kg per 5 acres is a lot in the village context')

data22_maize = data22[['hhid','total_kg_maize', 'area_cultivated']]
data22_maize.columns = ['hhid','total_kg_maize22', 'area_cultivated22']
data22_maize['maizeyield22'] = data22_maize['total_kg_maize22']/data22_maize['area_c

data_maize = data[['hhid','total_kg_maize','area_cultivated']]

data_maize['maizeyield'] = data['total_kg_maize']/data['area_cultivated']


panel_maize = data_maize.merge(data22_maize, how='inner', on='hhid')
panel_maize['maize_diff'] = panel_maize['total_kg_maize'] - panel_maize['total_kg_ma
panel_maize['maizeyield_diff'] =  panel_maize['maizeyield'] - panel_maize['maizeyiel


check_bigdrops = panel_maize.nsmallest(n=5, columns=['maize_diff'])
```

```
#print(check_bigdrops[['hhid','oldhhid', 'maize_diff', 'maizeyield_diff','total_kg_m

check_bigdrops2 = panel_maize.nsmallest(n=8, columns=['maizeyield_diff'])

#print(check_bigdrops2[['hhid','oldhhid', 'maize_diff', 'maizeyield_diff','total_kg_

print(check_bigdrops2)



# Summary total sellings kg:
sum_sold_kg= (data[['sold_kg_maize', 'sold_kg_groundnut', 'sold_kg_groundbean', 'sol
print('==============================================================')
print('Check: Distribution of crop Sellings (in kg)')
print('==============================================================')
sum_sold_kg.dropna(axis=1, how='any', inplace=True)
N_sellcrops = sum_sold_kg.iloc[0,:]
T_sellcrops = sum_sold_kg.iloc[0,:]*sum_sold_kg.iloc[1,:]
T_sell = T_sellcrops.sum()
print(sum_sold_kg)
## STOP RUN


#Summary sellings inside kg:
sum_sold_kg_inside = (data[['sold_insiders_kg_maize', 'sold_insiders_kg_groundnut',
print('==============================================================')
print('Check: Distribution of crop Sellings to Villagers')
print('==============================================================')
sum_sold_kg_inside.dropna(axis=1, how='any', inplace=True)

print(sum_sold_kg_inside)
## STOP RUN


list_cropssell = ['maize','groundnut','sweetpotatoe','soyabean','pigeonpeas','cotton



share_didsell = []
share_sell = []
share_sell.append(T_sell/T_prod)
for crop in list_cropssell:
    share_didsell.append(N_sellcrops['sold_kg_'+crop]/N_prodcrops['total_kg_'+crop])
    share_sell.append(T_sellcrops['sold_kg_'+crop]/T_prodcrops['total_kg_'+crop])

print('Share of sellings across crops')  # dont show it for all the crops. Just for
print(list_crops)
print(share_didsell)
print(share_sell)
# Sum transportation costs

# Summary Store kg:
sum_store_kg= (data[['store_kg_maize', 'store_kg_groundnut', 'store_kg_groundbean',
print('==============================================================')
print('Check: Distribution of crop store (in kg)')
print('==============================================================')
sum_store_kg.dropna(axis=1, how='any')
## STOP RUN


# ============================================================================
# Check quantity sold, store, not larger than total
# ============================================================================
for crop in list_crops:
```

```
    data['sold_bigger_total_'+crop] = 1*(data['sold_kg_'+crop].fillna(0)> data['tota
    data['store_bigger_total_'+crop] = 1*(data['store_kg_'+crop].fillna(0)> data['to

check_sold_bigger_total = data[['sold_bigger_total_maize', 'sold_bigger_total_ground


#Get the households that reported larger amounts than total:
list_hh_check_sell = []
list_hh_check_lost = []
list_hh_check_store = []
list_hh_check = []

for crop in list_crops:
    liers_sell = data.loc[data['sold_bigger_total_'+crop]==1, 'hhid'] # 'interviewee
    liers_store = data.loc[data['store_bigger_total_'+crop]==1, 'hhid'] #
    liers = data.loc[data['soldstore_bigger_total_'+crop]==1, 'hhid'] #

    list_hh_check_sell.append(liers_sell)
    list_hh_check_store.append(liers_store)
    list_hh_check.append(liers)

# sellings check:
hh_to_check_sell = pd.concat(list_hh_check_sell, axis=1)
hh_to_check_sell.columns = list_crops
print('')
print('===============================================================')
print('Check: Households-crop combination where SELLINGS larger than total produced'
print('===============================================================')
print(hh_to_check_sell.dropna(axis=1,  how='all'))
###STOP RUN

print('2 cases where sellings higher than total: Replace total by quantitiy sold (if

data.loc[data['total_kg_soyabean']<data['sold_kg_soyabean'],['total_kg_soyabean']] =
data.loc[data['total_kg_tomatoes']<data['sold_kg_tomatoes'],['total_kg_tomatoes']] =


# Store quantity check:
hh_to_check_store = pd.concat(list_hh_check_store, axis=1)
hh_to_check_store.columns = list_crops
print('')
print('===============================================================')
print('Check: Households-crop combination where STORED larger than total produced')
print('===============================================================')
print(hh_to_check_store.dropna(axis=1, how='all'))
### STOP RUN






hh_to_check = pd.concat(list_hh_check, axis=1)
hh_to_check.columns = list_crops
print('')
print('===============================================================')
print('Check: Households-crop combination where SELL+STORED larger than total produc
print('===============================================================')
hh_to_check.dropna(axis=1).to_string()

### Check each household that reported some amount bigger. Look at values, units and
# Write a note per each household and sent them to the enumerators.
#data_elia = data.loc[data['hhid']==93,]
#data_sell_outliers = data.loc[(data['hhid']==93) | (data['hhid']==56) | (data['hhid
#data_store_outliers = data.loc[(data['hhid']==62) | (data['hhid']==13) | (data['hhi
```

```
pd.options.display.float_format = '{:,.2f}'.format
# ==============================================================================
#  get selling PRICES per kg
# ==============================================================================
for crop in list_crops:
    data['p_'+crop] = (data['soldvalue'+crop].replace(0,np.nan)).dropna()  / (data['
    #DF = data[['soldvalue'+crop, 'sold_kg_'+crop]].dropna()

print('')
print('================================================================')
print('Check: Distribution of prices')
print('================================================================')
print((data[['p_maize', 'p_groundnut', 'p_groundbean', 'p_sweetpotatoe', 'p_fingermi
).dropna(axis=1))

list_crops_price = ['maize','groundnut','sweetpotatoe','pigeonpeas','cotton','tomato
price_data = pd.DataFrame(list_crops_price, columns=['crop'])
price_data['p_sell'] = np.nan

for item in list_crops_price:
    price_data.loc[price_data['crop']==item,'p_sell'] = np.nanmedian(data['p_'+item]



### UPLOAD SET OF SELLING AND CONSUMPTION PRICES:


### For the missing prices I use the ones from ISA-LSMS 2017. I use the maize price

print('WE NEED AN UPDATED ISA-LSMS TO USE PRICES AND KILOGRAMS CONVERSIONS. FOR THE
print('This is only for the few crops we do not have consupmtion price')
maize_isavillage = 297/57

# instead of using ISA-LSMS prices I use prices from google. They are minium retail
# https://www.africannewsagency.com/times-group-malawi/government-sets-2023-farm-gat
# for nkhwani I use price of greeny vegetables

prices = pd.DataFrame({'crop':list_crops})

# merge selling prices
prices = prices.merge(price_data,on='crop',how='left')

p_23['crop'] = p_23['good']

print('Assign consumption prices. Use more similar goods (ie maize grain) or higher
print(list_crops)
print(p_23.good.to_list())

# crops with same name, no need to repeat: groundnut, fingermillet,  'sugarcane'
# crops not in  consumption: groundbean, 'sorghum', 'pearlmillet', 'soyabean',cotton
rename_mapping ={'maizegrain':'maize', 'cassavatubers':'cassava',  'osweetpotatoes':

p_23['crop'].replace(rename_mapping, inplace=True)

# for items without price (ASK AUGUSTINE ABOUT PRICE OF THESE ITEMS)


prices = prices.merge(p_23[['crop','p_c']],on='crop',how='left')

prices.loc[prices['crop']=='groundbean','p_c'] = 600
prices.loc[prices['crop']=='sorghum','p_c'] = 400
```

```
prices.loc[prices['crop']=='pearlmillet','p_c'] = 675 # similar to finger millet
prices.loc[prices['crop']=='soyabean','p_c'] = 1000
prices.loc[prices['crop']=='nkhwani','p_c'] = 693 # similar to green vegs



## tanaposi and okra I couldnt find a price in ISA-LSMS (neither on cons nor prod).
prices.loc[prices['crop']=='therereokra','p_c'] = 575
prices.loc[prices['crop']=='tanaposi','p_c'] = 575
#no price for there okra in village or ISA-LSMS. Also not in internet. I use price o
prices['p_c'].fillna(prices['p_sell'], inplace=True)

if save==True:
    prices.to_csv('prices/prices_23.csv', index=False)

#Get monetary value:



# Using consumption prices. To use selling ones replace p_c for p_sell.
for crop in list_crops:
    data['y_'+crop] =  float(prices.loc[prices['crop']==crop, 'p_c'])*data['total_kg
    data['sold_MWK_'+crop] = float(prices.loc[prices['crop']==crop, 'p_sell'])*data[
    data['sold2_MWK_'+crop] =  data['soldvalue'+crop]
    data['sold_insiders_MWK_'+crop] = data['sold_insiders_kg_'+crop]
    data['store_MWK_'+crop] = float(prices.loc[prices['crop']==crop, 'p_c'])*data['s
   ### without loss production there is not an easy way to value all the production
    data['y_agric'] += data['y_'+crop].fillna(0)
    data['sold_agric'] += data['sold_MWK_'+crop].fillna(0)
    data['sold_insiders_agric'] += data['sold_insiders_MWK_'+crop].fillna(0)
    data['store_agric'] += data['store_MWK_'+crop].fillna(0)
    data['y_'+crop].replace(0,np.nan,inplace=True)
    data['sold_MWK_'+crop].replace(0,np.nan,inplace=True)
    data['store_MWK_'+crop].replace(0,np.nan,inplace=True)


data[['y_agric','sold_agric','sold_insiders_agric','store_agric']] = data[['y_agric'




print('')
print('================================================================')
print('Agricultural Output (rainy season) in $')
print('(================================================================')
print((data[['y_agric','y_maize', 'y_groundnut', 'y_pigeonpeas','y_tomatoes']]/dolla
print('Agricultural Output (rainy season) in Kgs')
print('(================================================================')
sum_ykg = data[['total_kg_maize', 'total_kg_groundnut','total_kg_pigeonpeas','total_

print(sum_ykg)
#STOP RUN



del sum_kg, check_bigdrops, check_bigdrops2, check_sold_bigger_total, data22_maize,


#%%  AGRICULTURAL INPUTS
print('      ')
print('      ')
print('   AGRICULTURAL INPUTS:   ')
print('      ')
print('LABOR INPUT')
```

```
print('        ')


### LABOR
#### Loop for labor input
for member in range(1,int(np.max(data['manyhhlaborplot'])+1)):
    data['months_member_'+str(member)] = np.nan
    data['weeks_member_'+str(member)] = np.nan
    data['days_member_'+str(member)] = np.nan
    data['hours_member_'+str(member)] = np.nan


for member in range(1,int(np.max(data['manyhhlaborplot'])+1)):
    data['months_member_'+str(member)] = data['monthshhplot_'+str(member)]
    data['weeks_member_'+str(member)] = data['weekshhplot_'+str(member)]
    data['days_member_'+str(member)] = data['dayshhplot_'+str(member)]
    data['hours_member_'+str(member)] = data['hourshhplot_'+str(member)]



for member in range(1,int(np.max(data['manyhhlaborplot'])+1)):
    data['months_member_'+str(member)].replace([0,0.0],np.nan,inplace=True)
    data['weeks_member_'+str(member)].replace([0,0.0],np.nan,inplace=True)
    data['days_member_'+str(member)].replace([0,0.0],np.nan,inplace=True)
    data['hours_member_'+str(member)].replace([0,0.0],np.nan,inplace=True)


sum_member = data[['months_member_1', 'weeks_member_1', 'days_member_1', 'hours_memb
print('================================================================')
print('Agriculture hh labor member 1 and 2')
print('================================================================')
print(sum_member)
### STOP RUN



data['hh_labor_days'] = 0
data['hh_labor_hours'] = 0

for member in range(1,int(np.max(data['manyhhlaborplot'])+1)):
    data['member_'+str(member)+'_labor_days'] =  (data['months_member_'+str(member)]
    data['member_'+str(member)+'_labor_hours'] = data['member_'+str(member)+'_labor_

for member in range(1,int(np.max(data['manyhhlaborplot'])+1)):

    data['hh_labor_days'] += data['member_'+str(member)+'_labor_days'].fillna(0)
    data['hh_labor_hours'] += data['member_'+str(member)+'_labor_hours'].fillna(0)


print('================================================================')
print('Distribution Agric Household Labor in days')
print('================================================================')
sum_labor_days = data[['hh_labor_days', 'member_1_labor_days', 'member_2_labor_days'
print(sum_labor_days)



print('================================================================')
print('Distribution Agric Household Labor in hours')
print('================================================================')
sum_labor_hours = data[['hh_labor_hours', 'member_1_labor_hours', 'member_2_labor_ho
print(sum_labor_hours)
```

```
# Hired labor
list_persons = ['men','women','kids']
data['w_men']=np.nan
data['w_women']=np.nan
data['w_kids']=np.nan
data['hired_N'] = 0

for person in list_persons:
    data['hired_N'] += data['manyhired'+str(person)].fillna(0)
    data['hired_'+str(person)+'_avg_hours'] = (data['hireplotmotnhs'+str(person)]*da
    data['hired_'+str(person)+'_L'] = data['manyhired'+str(person)]*data['hired_'+st
    data['w_'+str(person)] = (data['hireplotwage'+str(person)].replace(0,np.nan) / d
    data['weight_'+str(person)] = np.nanmedian(data['w_'+str(person)])

sum_hiredlabor = data[['w_men','w_women','w_kids' , 'hired_men_avg_hours', 'hired_wo
print('=========== Summary Hired Labor ===============')
print(sum_hiredlabor)


data['hhlabor_N'] = data['manyhhlaborplot']
data['labor_N'] = (data['hhlabor_N'].fillna(0) +data['hired_N'].fillna(0))
data['labor_h'] = (data['hh_labor_hours'].fillna(0) +data['hired_men_L'].fillna(0) +


sum_agriclabor = data[['labor_N', 'labor_h', 'hired_N', 'hh_labor_hours','hired_men_
print('=========== Summary Household + Hired Agricultural Labor input =============
print(sum_agriclabor)
print('Where _N denotes in supply number of persons, _h or _L in total hours')


### NEEED TO CLEAN YING VARIABLES


# FERTILIZER AND OTHER INTERMEDIATES
print('       ')
print('       ')
print(' FERTILIZER AND INTERMEDIATES ')
print('       ')
# obtain value non-bought fertilizer. Use median price

(data[['fertilizerbuymarketkg', 'buyfertilizierpay']].replace(0,np.nan)).describe()

data[['fertilizerbuymarketkg','buyfertilizierpay']] = remove_outliers(data[['fertili

data['p_fert']  = pd.to_numeric(data['buyfertilizierpay'].divide(data['fertilizerbuy

p_fertmed =np.nanmedian(data['p_fert'])
p_fertmean =np.nanmean(data['p_fert'])

# median price of a 50kg bag
print('mean price 50kg fertilizer bag:', p_fertmean*50)
print('med price 50kg fertilizer bag:', p_fertmed*50)

print('Note prices increases a lot wrt to 2022 (600%). in 2022 I got a number from t


## Use kg of fertilizer by total report
data['value_fertilizer'] = p_fertmed*data['fertilizerkg']

data[['govcoupon','fertilizeryes','fertilizerbuymarketyes']] = (data[['govcoupon','f

data[['fertilizerkg','fertilizerbuymarketkg','buyfertilizierpay']] = data[['fertiliz
## Coupons summary
```

```
print('Summary of Coupons =======================')
print((data[['govcoupon', 'govcouponmany' ]].describe()))

sum_fertilizer = data[['fertilizeryes','fertilizerkg','fertilizerbuymarketyes','fert

print('Summary fertilizer =========================')
print(sum_fertilizer)


print('NOTES: 52% hhs received coupons. 63% used feritlizer with an avg of 36 kg use
print('Rememeber though that Konje told us that fertilizer arrived in the village to

print('Top extreme values fertilizer ===================')
print(data.loc[data['fertilizerkg']>200,['hhid','fertilizerkg','area_cultivated']])
print('doesnt seem that extreme')


print(' other intermediates')

print('note the measure of intermediates does not use value fertilizer but fertilize

#intermediates
data['interm'] = (data['spendseeds'].fillna(0) +data['buyfertilizierpay'].fillna(0)

sum_interm = data[['interm','value_fertilizer','fertilizerkg','spendseeds', 'spendpe

print('=========== Summary Intermediate inputs ===============')
print('All variables in MWK except kg_fertilizer.')
print(sum_interm)





### ===== SUMMARY AGRICULTURAL INPUTS ================
data_inp = data[['hh_area_plots','hh_value_plots','k_farm','interm','labor_N', 'labo

data_inp[['hh_value_plots','k_farm','interm','value_fertilizer','spendseeds', 'spend
sum_inp = data_inp[['hh_area_plots','hh_value_plots','area_cultivated','k_farm','lab

print('===== SUMMARY AGRICULTURAL INPUTS in $ ================')
print(sum_inp)

sum_inp2 = data_inp[[ 'hired_kids_L', 'interm','value_fertilizer', 'spendseeds', 'sp
print(sum_inp2)

'hh_area_plots','hh_value_plots','area_cultivated','k_farm','labor_N', 'labor_h', 'h

#%% Check cashtransfer subsidy
print('      ')
print('      ')
print(' CASH TRANSFER ')
print('      ')

data['cashtrans_yes'] = data['other_sour_income_3'].replace(2,0)
data['cashtrans_value'] = data['other_sour_income_4'].replace(0,np.nan)

sum_subsidy = data[['cashtrans_yes','cashtrans_value']].describe()
print('================================================================')
print('Conditional Cash Transfer Program Implementation in the Village.')
print('================================================================')
print(sum_subsidy)
```

```
#%% ============================================================================
#  Shocks
# ============================================================================
print('      ')
print('      ')
print('     SHOCKS ')
print('      ')


list_abcd = ['a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r'

data['shocks'] = 0
for a in list_abcd:
    data['shocks'] += (data['shocks_'+a+'1'].replace(2,0)).fillna(0)

data['shock_flood'] = data['shocks_a1'].replace(2,0)
data['shock_drought'] = data['shocks_b1'].replace(2,0)
data['shock_lndslide'] = data['shocks_c1'].replace(2,0)
data['shock_covid'] = data['shocks_d1'].replace(2,0)
data['shock_adultill'] = data['shocks_e1'].replace(2,0)
data['shock_kidill'] = data['shocks_f1'].replace(2,0)
data['shock_death_earner'] = data['shocks_g1'].replace(2,0)
data['shock_death_othermemb'] = data['shocks_h1'].replace(2,0)
data['shock_inp_p'] = data['shocks_i1'].replace(2,0)
data['shock_out_p'] = data['shocks_j1'].replace(2,0)
data['shock_pests'] = data['shocks_k1'].replace(2,0)
data['shock_lvstk'] = data['shocks_l1'].replace(2,0)
data['shock_theft'] = data['shocks_m1'].replace(2,0)
data['shock_theft_agric'] = data['shocks_n1'].replace(2,0)
data['shock_business'] = data['shocks_o1'].replace(2,0)
data['shock_unemp'] = data['shocks_p1'].replace(2,0)
data['shock_wage_decr'] = data['shocks_q1'].replace(2,0)
data['shock_other'] = data['shocks_r1'].replace(2,0)





shocks = data['shocks'].value_counts()/len(data)
#Proportion of individuals that reported each shock
shocks_avg= np.array(np.mean(data[['shock_flood','shock_drought','shock_lndslide','s
]],axis=0))


p_shocks = np.sum(shocks_avg)



labels = ['flood', 'drought', 'lndslide', 'covid', 'adult ill', 'kid/elder ill', 'de


#Bar Plot
fig, ax = plt.subplots(figsize=(8,8))
ax.barh(np.arange(len(shocks_avg)), shocks_avg, tick_label=labels)
plt.title('Reported Shocks last rainy season Pilot village 2022')
plt.xlabel('Proportion Households Experienced the Shock')
plt.show()
if save==True:
    fig.savefig(folder_fig+'village_shocks.png', bbox_inches='tight')



### LABOR INCOME: SALARY LABOR (last month) ----------------------
print('      ')
```

```
print('      ')
print(' lABOR INCOME: SALARY LABOR')
print('      ')


## propotion households getting sallary work:
data['lobor_inc1'].replace(2,0, inplace=True)

print('===============================================================')
print(' Salary labor (1 month)')
print('===============================================================')
print('Numb. households with a sallary job:', pd.value_counts(data['lobor_inc1'])[1]


data['wlabor'] = 0
## labor supply in hours
data['wlabor_supply'] = (data['lobor_inc4'].multiply(data['lobor_inc5'],axis=0, fill
data['wlabor_supply'].replace(0,np.nan, inplace=True)
## Construct wlabor income for the last month
pd.value_counts(data['lobor_inc8'])

#households only reported salary in months

## Let's compute the total labor income at the month level
# wage  periods
# month: 1
data.loc[data['lobor_inc8']==1,'wlabor_inc'] = data.loc[data['lobor_inc8']==1, 'lobo
# weeks
data.loc[data['lobor_inc8']==2, 'wlabor_inc'] = (data.loc[data['lobor_inc8']==2, 'lo
# Days
#data.loc[data['lobor_inc8']==3, 'wlabor_inc'] = (data.loc[data['lobor_inc8']==3, 'l

data['wlabor_inc'] = pd.to_numeric(data['wlabor_inc'])
data['wlabor_supply'] = pd.to_numeric(data['wlabor_supply'])
data['wlabor_inc_dollar'] = data['wlabor_inc']/dollar_MWK

print(data[['wlabor_supply','wlabor_inc_dollar']].describe())


## To rainy season
data['wlabor_inc'] = pd.to_numeric(data['wlabor_inc'])*7
data['wlabor_supply'] = pd.to_numeric(data['wlabor_supply'])*7

print('      ')
print('      ')
print(' lABOR INCOME: GANYU LABOR')
print('      ')

print('Ganyu income coming from network data')

## GANYU INCOME MISSING SINCE IT IS INSIDE THE NETWORK. LAST MONTH

transfers = pd.read_csv('C:/Users/rodri/Dropbox/Malawi/SIEG2021 (1)/2023 July/Data/C

ganyu = transfers.loc[transfers['good']=='ganyu',['hhid','ganyu.cash','ganyu.days','

ganyu = ganyu.groupby(by='hhid').sum()
ganyu.reset_index(inplace=True)

ganyu.rename(columns={'ganyu.cash':'ganyu_inc','ganyu.cash_dollar':'ganyu_inc_dollar

data = pd.merge(data,ganyu,on='hhid',how='left')

data['ganyu_yes'] = 1*(data['ganyu_inc']>0)
```

```
print('Numb. households did Ganyu:',sum(data['ganyu_yes']))
print('in 2023 we had 103 hhs in ganyu. In february we had 160.')
print('Asking via network might make us lose some observations in ganyu. Note timing
print('=============================================================')
print('Ganyu summary at the household level. Last month')
print('=============================================================')
print(data[['ganyu_yes','ganyu_inc','ganyu.days', 'ganyu_inc_dollar']].describe())

data['ganyu_inc'] = pd.to_numeric(data['ganyu_inc'])*7
data['ganyu_supply'] = pd.to_numeric(data['ganyu.days'])*7
data['ganyu_inc_dollar'] = pd.to_numeric(data['ganyu_inc_dollar'])*7

### BUSINESS INCOME (last month) -------------------------------------------

print(' BUSINESS INCOME')
print('      ')

data['busin_income_1'].replace(2,0)
pd.value_counts(data['busin_income_1'])
type_business = pd.value_counts(data['busin_income_2'])
data['business_type'] = data['busin_income_2']

data['business_months'] = data['busin_income_3']
data['business_revenue'] = data['busin_income_5']
data['business_profits'] = data['busin_income_4']
data['business_costs'] = data['busin_income_6'] + data['busin_income_7']
data['business_profits2'] = data['business_revenue'] - data['business_costs']

business_data = data.loc[data['busin_income_1']==1, ['hhid', 'business_type', 'busin
business_data[['business_revenue', 'business_costs', 'business_profits']] = business

data[['business_revenue', 'business_costs', 'business_profits','business_profits2']]

print('types of business in the village. Values in $, last month')
print(business_data)

data['business_revenue'] =  pd.to_numeric(data['business_revenue'])
data['business_costs'] =  pd.to_numeric(data['business_costs'])
data['business_profits'] =  pd.to_numeric(data['business_profits'])



sum_business = (data[['business_revenue', 'business_costs', 'business_profits', 'bus
print('=============================================================')
print('Summary Business income, month level, in dollars')
print('=============================================================')
print(sum_business)

## to rainy season:


data[['business_revenue']] =data['business_revenue']*data['business_months']*7/12
data[['business_costs']] =data['business_costs']*data['business_months']*7/12
data[['business_profits']] =data['business_profits']*data['business_months']*7/12
data[['business_profits2']] =data['business_profits2']*data['business_months']*7/12



print('=============================================================')
print('Salary Labor, Ganyu labor, and Business summary (at rainy season, 7 months)')
print('=============================================================')
print('income in dollars')
print(data[['wlabor_inc_dollar','wlabor_supply', 'ganyu_inc', 'ganyu_supply','busine
```

```
## compare with agriculture: vey low values in agric...
(data['y_agric'].replace(0,np.nan)/dollar_MWK).describe()

print('      ')
print('      ')
print(' GOV, NGO TRANSFERS AND REMITTANCES')
print('      ')
### OTHER SOURCES OF INCOME -------------------------------------------------
data[['NGO_yes', 'cashtrans_yes', 'gov_yes', 'remittances_yes']] = data[['other_sour
data[['NGO_yes', 'cashtrans_yes', 'gov_yes', 'remittances_yes']] = data[['NGO_yes',
sum_other_prop = np.mean(data[['NGO_yes', 'cashtrans_yes', 'gov_yes', 'remittances_y

data['cashtrans_value'] = data['cashtrans_value']
data['NGO_trans'] = data['other_sour_income_2']
data['gov_trans'] = data['other_sour_income_6']
data['remittances'] = data['other_sour_income_8']
data['other_inc'] = data[['cashtrans_value', 'NGO_trans', 'gov_trans', 'remittances'
sum_other = (data[['cashtrans_value', 'NGO_trans', 'gov_trans', 'remittances']].repl

print(sum_other)

print('1000$ of cash transfer might be too much? This is a government transfer')

### AGGREGATE INCOME ------------------------------------------------------------

### FOR THE MOMENT I DO NOT SUBSTRACT FOR INTERMEDIATES COSTS. NEED TO BE SURE HOW W

print('  AGGREGATE INCOME ' )
print('      ')
print(' For the moment agricultural income is output minus hired wages. I do not sub


data['y_net'] = data['y_agric'].fillna(0) -data['hireplotwagemen'].fillna(0) -data['


## inctotal using agric revenues not profits
data['inctotal'] = data[['y_agric' ,'wlabor_inc', 'ganyu_inc', 'business_profits']].

data['inctotal_trans'] = data[['y_agric' ,'wlabor_inc', 'ganyu_inc', 'business_profi


income = data[['hhid','inctotal','inctotal_trans','y_net','y_agric','y_maize', 'y_gr

sum_inc = (income.loc[:, income.columns != 'hhid']/dollar_MWK).describe(percentiles=

var_list = ['inctotal','inctotal_trans', 'y_net', 'y_agric','y_maize', 'y_groundnut'
gini_stat= np.empty((1, len(var_list)))

for i,state in enumerate(var_list):
    gini_stat[:,i] = gini(income[state].dropna().values)

data_gini = pd.DataFrame(gini_stat, columns=var_list)
data_gini.reset_index(inplace=True)
data_gini['index'] = 'gini'
sum_inc.reset_index(inplace=True)
sum_inc = sum_inc.append(data_gini, ignore_index=True)

print('================================================================')
print('Summary total Income (rainy season)')
print('================================================================')
```

```
print('values in $')
print(sum_inc)

print('AGAIN A VERY LOW INCOME... DUE TO VERY LOW AGRICULTURAL PRODUCTION.')
print('This is consistent though with the bad harvest in Malawi, the expensive and m
# quite low numbers of income....

data

#sum_inc.to_csv('C:/Users/rodri/Dropbox/Chied_Field_June_19/Data/Income/outputs/summ

#%% ===========================================================================
#   WEALTH
# ============================================================================
print('      ')
print('      ')
print('  WEALTH    ')
print('      ')

data['housing'] = data['selldwell']

data['hh_assets'] = 0
for i in range(1,12):
    data['hh_assets'] += data['sellhhasset_'+str(i)]


sum_assets = data[['housing','hh_assets']].describe(percentiles=percentiles)


#STOP RUN


data['wtotal'] = data[['housing','hh_assets','hh_value_plots','k_farm','hhlivestock'

print('======== Summary Wealth ==========')
print((data[['wtotal','housing','hh_assets','hh_value_plots','k_farm','hhlivestock']

#%% Save dataset

### let's do some checks (before we get augustine corrections). Remove observations
data[['y_agric', 'interm','labor_h', 'k_farm', 'hh_area_plots']].describe(percentile


data['wave'] = 2023



data_short =  data[['hhid',  'wave','rightsellland', 'chiefpreventsell', 'chiefpreve
                     'inctotal', 'inctotal_trans','y_net','y_agric', 'y_maize', 'y_g
                      'y_cassava',  'y_soyabean', 'y_sorghum',  'y_fingermillet', 'y
                    'sold_agric','sold_insiders_agric','store_agric',

                    'hh_area_plots', 'hh_ratio_value_rent', 'hh_p_acre_plots', 'area
                    'labor_N', 'labor_h', 'hh_labor_hours','hired_men_L', 'hired_wom
                   'wlabor_inc', 'wlabor_supply', 'ganyu_yes', 'ganyu_inc', 'ganyu_s

                 'NGO_yes', 'cashtrans_yes', 'gov_yes', 'remittances_yes', 'other_in
                    'wtotal','housing','hh_assets','hh_value_plots','k_farm','hhlive

                    'shocks', 'shock_flood','shock_drought','shock_lndslide','shock_
                    'shock_adultill','shock_kidill','shock_death_earner','shock_deat


## data with income at the year level -----------------------------------
```

```
data_short_year =  data_short
data_short_year[['wlabor_inc', 'wlabor_supply','ganyu_inc','ganyu_supply','business_


## inctotal using agric revenues not profits
data_short_year['inctotal'] = data_short_year[['y_agric' ,'wlabor_inc', 'ganyu_inc',
data_short_year['inctotal_trans'] = data_short_year[['y_agric' ,'wlabor_inc', 'ganyu



# summary

income = data_short_year[['hhid','inctotal','inctotal_trans','y_net','y_agric','y_ma
sum_inc = (income.loc[:, income.columns != 'hhid']/dollar_MWK).describe(percentiles=
var_list = ['inctotal','inctotal_trans', 'y_net', 'y_agric','y_maize', 'y_groundnut'
gini_stat= np.empty((1, len(var_list)))

for i,state in enumerate(var_list):
    gini_stat[:,i] = gini(income[state].dropna().values)

data_gini = pd.DataFrame(gini_stat, columns=var_list)
data_gini.reset_index(inplace=True)
data_gini['index'] = 'gini'
sum_inc.reset_index(inplace=True)
sum_inc = sum_inc.append(data_gini, ignore_index=True)

print('===============================================================')
print('Summary total Income (year level)')
print('===============================================================')
print('values in $')
print(sum_inc)




if save==True:
    data.to_csv('income_wealth_22_LONG_rainseas.csv')
    data_short.to_csv('income_wealth_23_rainseas.csv', index=False)
    data_short_year.to_csv('income_wealth_23_year.csv', index=False)
print('      ')
print('      ')
print("-----------------------------")
print(' DATA SAVED in July 2023/Data/Clean data/income')
print("-----------------------------")
print('in rainy season level (7 months) income_wealth_23_rainseas.csv')
print('in yearly level income_wealth_23_year.csv')
print("-----------------------------")
print('dataset contains the variables: ')
print(" 'hhid',  'wave','rightsellland', 'chiefpreventsell', 'chiefpreventbequeat',
print(" 'inctotal', 'inctotal_trans','y_net','y_agric', 'y_maize', 'y_groundnut', 'y
print(" 'y_cassava',  'y_soyabean', 'y_sorghum',  'y_fingermillet', 'y_cotton', 'y_t
print(" 'sold_agric','sold_insiders_agric','store_agric',")
print(" 'hh_area_plots', 'hh_ratio_value_rent', 'hh_p_acre_plots' 'area_cultivated',
print(" 'labor_N', 'labor_h', 'hh_labor_hours','hired_men_L', 'hired_women_L', 'hire
print(" 'wlabor_inc', 'wlabor_supply', 'ganyu_yes', 'ganyu_inc', 'ganyu_supply','bus
print(" 'NGO_yes', 'cashtrans_yes', 'gov_yes', 'remittances_yes', 'other_inc', 'cash
print(" 'wtotal','housing','hh_assets','hh_value_plots','k_farm','hhlivestock',")
print(" 'shocks', 'shock_flood','shock_drought','shock_lndslide','shock_covid',")
print(" 'shock_adultill','shock_kidill','shock_death_earner','shock_death_othermemb'

print("-----------------------------")
print("y_net is agricultucal net income (minus intermediates). (MWK)")
print("y_agric is gross agricultural income (MWK)   ")
```

```
print("Labor variables: N denotes unit is number of persons. labor_h, denotes total
print("Shock variables: whether households reported the shock or not.")
```

```
Check missing households
Empty DataFrame
Columns: [hhid, intervieweename]
Index: []
No missing households
===============================================================
These households are duplicate
===============================================================
Series([], Name: hhid, dtype: int64)
all households interviewed are also in the roster


   LAND SIZE, VALUE, AND RIGHTS


===============================================================
Check: Distribution land at household level
===============================================================
       area_cultivated  hh_area_plots  hh_rentout_plots  hh_value_plots  \
count           273.00         269.00            260.00          245.00
mean              1.90           2.23         49,492.31      792,538.77
std               1.22           1.81         48,490.67    1,022,381.34
min               0.50           0.50          5,000.00          999.00
25%               1.00           1.00         20,000.00      200,000.00
50%               1.50           2.00         37,500.00      500,000.00
75%               2.50           3.00         60,000.00    1,000,000.00
max               7.00          21.00        480,000.00    6,000,000.00

       hh_ratio_value_rent  hh_p_acre_plots
count               245.00           245.00
mean                 16.51       338,158.50
std                  15.47       293,244.82
min                   0.07           999.00
25%                   6.67       150,000.00
50%                  12.50       266,666.67
75%                  20.00       450,000.00
max                 133.33     2,000,000.00
Outliers land?
    hhid  hh_area_plots
89  1211          21.00
Last year Augus confirmed hhid 1211 has 21 acres
       rightsellland  rightbequeathplot  chiefpreventsell  \
count         269.00             269.00            269.00
mean            0.41               0.22              0.12
std             0.49               0.41              0.32

       chiefpreventbequeat  landdispute
count               269.00       269.00
mean                  0.08         0.11
std                   0.27         0.32


   FARM CAPITAL AND LIVESTOCK

===============================================================
Check: Farm Capital Value (in $)
===============================================================
       k_farm  hhlivestock  hhfarmequip  hhfarmstruct
count  284.00       284.00       284.00        284.00
mean    18.46        50.30        11.04          7.42
std     43.46       135.00        10.44         40.92
min      0.00         0.00         0.00          0.00
25%      4.28         0.00         4.28          0.00
50%      9.03         9.51         8.56          0.00
75%     17.11        55.61        14.59          0.00
max    596.96     1,948.67        89.35        570.34
```

```
I checked the two households with extreme values. The reason of the high values is b
ecause they have cows.


  AGRICULTURAL PRODUCTION

                        1.00   2.00   3.00   4.00   5.00   6.00   7.00   8.00   9.00
10.00  11.00  12.00
unitstotalmaize          3.00    nan 224.00    nan   3.00    nan   1.00   3.00   6.0
0    nan    nan  10.00
unitstotalgroundnut      1.00 103.00   3.00   2.00    nan    nan    nan   5.00  10.0
0    nan   6.00   1.00
unitstotalgroundbean      nan  15.00   2.00    nan    nan   1.00    nan   4.00  14.0
0    nan   1.00   1.00
unitstotalsweetpotatoe    nan  22.00   5.00    nan   1.00    nan    nan    nan   2.0
0   1.00    nan    nan
unitstotalfingermillet   1.00    nan    nan    nan    nan    nan    nan   2.00   4.0
0    nan    nan   2.00
unitstotalsorghum         nan    nan  10.00    nan    nan    nan    nan   2.00   7.0
0    nan    nan   4.00
unitstotalpearlmillet     nan    nan    nan    nan    nan    nan    nan    nan    na
n    nan    nan    nan
unitstotalsoyabean       3.00    nan  11.00    nan   1.00    nan    nan   2.00  10.0
0    nan    nan   2.00
unitstotalpigeonpeas     3.00   4.00 144.00    nan   1.00    nan    nan  13.00  34.0
0    nan   1.00  40.00
unitstotalcotton         2.00    nan   7.00    nan    nan    nan    nan    nan    na
n   5.00    nan    nan
unitstotalnkhwani         nan    nan   1.00    nan    nan    nan    nan  29.00  21.0
0   1.00    nan   1.00
unitstotalcassava         nan    nan   5.00    nan    nan    nan    nan    nan   1.0
0    nan    nan    nan
unitstotalsugarcane       nan    nan    nan    nan    nan    nan    nan    nan    na
n   2.00    nan    nan
unitstotaltomatoes        nan    nan    nan    nan    nan    nan    nan   6.00   3.0
0    nan    nan    nan
unitstotalthereokra      2.00    nan    nan    nan    nan    nan    nan   1.00  19.0
0    nan    nan    nan
unitstotaltanaposi        nan    nan    nan    nan    nan    nan    nan   1.00    na
n   1.00    nan    nan
We might want to remove some units for a next time.
================================================================
Crop production: Number of households harvested crops
================================================================
total_kg_maize          250
total_kg_groundnut      131
total_kg_groundbean      38
total_kg_sweetpotatoe    30
total_kg_fingermillet     9
total_kg_sorghum         23
total_kg_pearlmillet      0
total_kg_soyabean        29
total_kg_pigeonpeas     240
total_kg_cotton           9
total_kg_nkhwani         51
total_kg_cassava          6
total_kg_sugarcane        0
total_kg_tomatoes         9
total_kg_therereokra     22
total_kg_tanaposi         1
dtype: int64
================================================================
 Distribution of crop production (in kg)
================================================================
       total_kg_maize  total_kg_groundnut  total_kg_groundbean  \
count             250                 131                   38
mean              213                 140                   35
std               437                 176                   35
min                10                   5                    2
```

10%	25	25	5
25%	50	50	10
50%	150	100	25
75%	250	150	50
90%	400	300	80
99%	1,275	658	141
max	6,100	1,500	150

	total_kg_sweetpotatoe	total_kg_fingermillet	total_kg_sorghum \
count	30	9	23
mean	147	14	38
std	170	8	34
min	5	2	2
10%	25	4	5
25%	50	10	22
50%	75	10	25
75%	188	20	50
90%	400	25	70
99%	678	25	139
max	750	25	150

	total_kg_soyabean	total_kg_pigeonpeas	total_kg_cotton \
count	29	240	9
mean	63	59	350
std	79	60	202
min	5	2	100
10%	10	10	180
25%	15	25	200
50%	35	50	350
75%	75	75	450
90%	120	126	550
99%	322	311	730
max	350	500	750

	total_kg_nkhwani	total_kg_cassava	total_kg_tomatoes \
count	51	6	9
mean	31	84	160
std	30	67	322
min	5	5	1
10%	10	28	4
25%	20	50	10
50%	20	75	20
75%	40	100	100
90%	60	150	360
99%	150	195	936
max	200	200	1,000

	total_kg_therereokra
count	22
mean	9
std	9
min	2
10%	2
25%	5
50%	5
75%	13
90%	15
99%	36
max	40

Check top maize producers:

	hhid	total_kg_maize	area_cultivated
190	1429	2,050	5
191	1430	6,100	5

Not sure if 6100 kg per 5 acres is a lot in the village context

	hhid	total_kg_maize	area_cultivated	maizeyield	total_kg_maize22 \
149	1338	250	3	91	250
21	1025	600	1	600	1,175
195	1440	250	4	62	600
61	1123	420	2	168	1,200

213	1506	125	2	83	250
17	1021	50	0	100	250
192	1437	250	2	167	275
73	1140	20	1	20	200

	area_cultivated22	maizeyield22	maize_diff	maizeyield_diff
149	0	1,000	0	-909
21	1	1,175	-575	-575
195	1	600	-350	-538
61	2	600	-780	-432
213	0	500	-125	-417
17	0	500	-200	-400
192	0	550	-25	-383
73	0	400	-180	-380

===
Check: Distribution of crop Sellings (in kg)
===

	sold_kg_maize	sold_kg_groundnut	sold_kg_sweetpotatoe \
count	18	18	8
mean	64	107	128
std	82	98	143
min	5	25	5
25%	10	50	20
50%	50	62	62
75%	100	138	212
max	350	420	400

	sold_kg_soyabean	sold_kg_pigeonpeas	sold_kg_cotton	sold_kg_tomatoes
count	14	32	9	4
mean	64	30	333	305
std	68	36	208	403
min	5	2	100	20
25%	13	9	200	80
50%	38	15	200	150
75%	100	50	450	375
max	225	175	750	900

===
Check: Distribution of crop Sellings to Villagers
===

	sold_insiders_kg_maize	sold_insiders_kg_groundnut \
count	13	7
mean	67	89
std	93	76
min	5	25
10%	5	40
25%	10	50
50%	50	50
75%	100	100
90%	100	160
99%	320	241
max	350	250

	sold_insiders_kg_sweetpotatoe	sold_insiders_kg_soyabean \
count	5	2
mean	133	8
std	168	4
min	20	5
10%	20	6
25%	20	6
50%	25	8
75%	200	9
90%	320	10
99%	392	10
max	400	10

	sold_insiders_kg_pigeonpeas	sold_insiders_kg_tomatoes
count	7	2
mean	24	110
std	20	127

```
min                              2                         20
10%                              4                         38
25%                              8                         65
50%                             25                        110
75%                             38                        155
90%                             50                        182
99%                             50                        198
max                             50                        200
Share of sellings across crops
['maize', 'groundnut', 'groundbean', 'sweetpotatoe', 'fingermillet', 'sorghum', 'pea
rlmillet', 'soyabean', 'pigeonpeas', 'cotton', 'nkhwani', 'cassava', 'sugarcane', 't
omatoes', 'therereokra', 'tanaposi']
[0.072, 0.13740458015267176, 0.26666666666666666, 0.4827586206896552, 0.133333333333
33333, 1.0, 0.4444444444444444]
[0.10072956945753579, 0.021816813992853113, 0.1049610496104961, 0.23155505107832008,
0.49696969696969695, 0.06753228424246702, 0.9523809523809523, 0.8494342906875544]
===============================================================
Check: Distribution of crop store (in kg)
===============================================================

===============================================================
Check: Households-crop combination where SELLINGS larger than total produced
===============================================================
     sweetpotatoe
251          1550
2 cases where sellings higher than total: Replace total by quantitiy sold (if necess
ary)

===============================================================
Check: Households-crop combination where STORED larger than total produced
===============================================================
     maize  pigeonpeas
181    nan       1,419
191  1,430         nan

===============================================================
Check: Households-crop combination where SELL+STORED larger than total produced
===============================================================

===============================================================
Check: Distribution of prices
===============================================================
       p_maize  p_groundnut  p_sweetpotatoe  p_soyabean  p_pigeonpeas  \
count    18.00        18.00            8.00       14.00         32.00
mean    400.00       319.04          213.12      452.13        440.39
std      97.01        90.98          100.96      109.35         99.38
min     200.00       116.00           80.00      360.00        200.00
25%     400.00       285.00          155.00      400.00        400.00
50%     400.00       320.00          187.50      414.55        500.00
75%     400.00       400.00          262.50      482.14        500.00
max     600.00       400.00          400.00      800.00        600.00

       p_cotton  p_tomatoes
count      9.00        4.00
mean     397.22      370.83
std      230.64      105.74
min      200.00      250.00
25%      220.00      312.50
50%      300.00      366.67
75%      580.00      425.00
max      750.00      500.00
WE NEED AN UPDATED ISA-LSMS TO USE PRICES AND KILOGRAMS CONVERSIONS. FOR THE MOMENT
I USE 2017 WAVE WITH THE MAIZE REFERENCE IN THE VILLAGE
This is only for the few crops we do not have consupmtion price
Assign consumption prices. Use more similar goods (ie maize grain) or higher price
(capture better value not selling)
['maize', 'groundnut', 'groundbean', 'sweetpotatoe', 'fingermillet', 'sorghum', 'pea
rlmillet', 'soyabean', 'pigeonpeas', 'cotton', 'nkhwani', 'cassava', 'sugarcane', 't
omatoes', 'therereokra', 'tanaposi']
```

```
['maizemgaiwa', 'maizerefined', 'maizemadeya', 'maizegrain', 'greenmaize', 'rice',
'cassavatubers', 'wsweetpotatoes', 'osweetpotatoes', 'ipotatoes', 'potatocrisps', 'b
bean', 'pigeonpea', 'groundnut', 'groundnutf', 'onion', 'cabbage', 'tanaposi', 'leaf
yvegetables', 'tomato', 'eggs', 'driedfish', 'fleshfish', 'goat', 'chicken', 'otherp
oultry', 'smockedfish', 'mango', 'banana', 'guava', 'wildfruits', 'sugar', 'sugarcan
e', 'cookingoil', 'softdrinks', 'thobwa', 'locallybrewed', 'salt', 'fingermillet',
'mandazidou']

================================================================
Agricultural Output (rainy season) in $
(================================================================
```

	y_agric	y_maize	y_groundnut	y_pigeonpeas	y_tomatoes
count	267.00	250.00	131.00	240.00	9.00
mean	255.01	142.95	132.74	32.27	34.53
std	383.41	293.78	167.56	32.72	69.62
min	2.73	6.72	4.75	1.37	0.27
10%	43.94	16.80	23.76	5.47	0.92
25%	80.56	33.61	47.53	13.66	2.16
50%	161.75	100.82	95.06	27.33	4.33
75%	290.44	168.03	142.59	40.99	21.64
90%	514.36	268.85	285.17	69.14	77.90
99%	1,600.52	857.29	625.48	169.99	202.54
max	4,639.38	4,099.93	1,425.86	273.29	216.39

```
Agricultural Output (rainy season) in Kgs
(================================================================
```

	total_kg_maize	total_kg_groundnut	total_kg_pigeonpeas \
count	284.00	284.00	284.00
mean	187.22	64.41	49.90
std	415.79	138.33	59.02
min	0.00	0.00	0.00
25%	50.00	0.00	10.00
50%	100.00	0.00	27.50
75%	202.50	75.00	51.25
max	6,100.00	1,500.00	500.00

	total_kg_tomatoes
count	284.00
mean	5.06
std	60.91
min	0.00
25%	0.00
50%	0.00
75%	0.00
max	1,000.00

```
   AGRICULTURAL INPUTS:

LABOR INPUT

================================================================
Agriculture hh labor member 1 and 2
================================================================
```

	months_member_1	weeks_member_1	days_member_1	hours_member_1 \
count	269.00	269.00	269.00	269.00
mean	5.05	3.59	5.68	3.90
std	1.63	0.75	1.14	1.50
min	1.00	1.00	1.00	1.00
50%	5.00	4.00	6.00	4.00
max	7.00	4.00	7.00	12.00

	months_member_2	weeks_member_2	days_member_2	hours_member_2
count	203.00	203.00	203.00	203.00
mean	4.71	3.66	5.07	3.65
std	1.61	0.64	1.72	1.58
min	1.00	1.00	1.00	1.00
50%	4.00	4.00	6.00	3.00
max	7.00	4.00	7.00	9.00

```
================================================================
```

Distribution Agric Household Labor in days

==

	hh_labor_days	member_1_labor_days	member_2_labor_days
count	284.00	269.00	203.00
mean	211.46	104.60	89.57
std	165.67	46.69	48.81
min	0.00	2.00	8.00
10%	60.00	48.00	25.40
25%	100.00	72.00	55.00
50%	168.00	96.00	84.00
75%	274.00	144.00	120.00
90%	404.60	168.00	168.00
99%	844.76	196.00	196.00
max	1,176.00	196.00	196.00

	member_3_labor_days
count	108.00
mean	62.65
std	51.50
min	4.00
10%	16.00
25%	26.50
50%	46.50
75%	84.00
90%	168.00
99%	196.00
max	196.00

==

Distribution Agric Household Labor in hours

==

	hh_labor_hours	member_1_labor_hours	member_2_labor_hours
count	284.00	269.00	203.00
mean	770.11	421.41	343.10
std	641.35	271.78	264.75
min	0.00	8.00	16.00
10%	131.40	144.00	72.00
25%	336.00	216.00	144.00
50%	598.50	360.00	288.00
75%	1,051.00	560.00	480.00
90%	1,556.40	784.00	672.00
99%	2,914.00	1,271.52	1,176.00
max	3,920.00	1,568.00	1,568.00

	member_3_labor_hours
count	108.00
mean	177.29
std	195.07
min	0.00
10%	36.00
25%	70.00
50%	120.00
75%	197.00
90%	360.00
99%	828.24
max	1,372.00

=========== Summary Hired Labor ===============

	w_men	w_women	w_kids	hired_men_avg_hours	hired_women_avg_hours
count	46.00	28.00	16.00	47.00	29.00
mean	433.98	582.20	343.10	111.32	54.83
std	378.58	590.86	317.42	145.59	75.80
min	26.79	48.61	41.67	1.00	1.00
25%	133.48	157.55	107.81	14.00	8.00
50%	416.67	358.33	208.33	32.00	21.00
75%	625.00	800.00	562.50	186.00	50.00
max	1,666.67	2,500.00	1,000.00	560.00	288.00

	hired_kids_avg_hours	hireplotwagemen	hireplotwagewomen
count	17.00	284.00	284.00
mean	23.59	2,931.34	1,258.80

std	39.42	8,053.58	4,722.97
min	1.00	0.00	0.00
25%	3.00	0.00	0.00
50%	10.00	0.00	0.00
75%	24.00	0.00	0.00
max	160.00	56,000.00	34,000.00

	hireplotwagekids
count	284.00
mean	153.35
std	814.30
min	0.00
25%	0.00
50%	0.00
75%	0.00
max	8,000.00

=========== Summary Household + Hired Agricultural Labor input ===============

	labor_N	labor_h	hired_N	hh_labor_hours	hired_men_L	hired_women_L \
count	284.00	284.00	284.00	284.00	47.00	29.00
mean	3.38	836.25	0.95	770.11	275.47	168.79
std	2.67	688.31	2.25	641.35	429.17	300.74
min	0.00	0.00	0.00	0.00	1.00	1.00
25%	2.00	384.00	0.00	336.00	24.00	16.00
50%	3.00	661.00	0.00	598.50	64.00	48.00
75%	4.00	1,152.00	0.00	1,051.00	372.00	150.00
max	18.00	3,992.00	16.00	3,920.00	2,240.00	1,200.00

	hired_kids_L
count	17.00
mean	55.47
std	60.44
min	1.00
25%	14.00
50%	36.00
75%	48.00
max	192.00

Where _N denotes in supply number of persons, _h or _L in total hours

FERTILIZER AND INTERMEDIATES

mean price 50kg fertilizer bag: 59491.777713865726
med price 50kg fertilizer bag: 50000.0
Note prices increases a lot wrt to 2022 (600%). in 2022 I got a number from the data of 15000 MWK, but Auga corrected me and told me it was 35000 MWK
Summary of Coupons ======================

	govcoupon	govcouponmany
count	273.00	284.00
mean	0.52	0.66
std	0.50	0.73
min	0.00	0.00
25%	0.00	0.00
50%	1.00	1.00
75%	1.00	1.00
max	1.00	2.00

Summary fertilizer ========================

	fertilizeryes	fertilizerkg	fertilizerbuymarketyes \
count	273.00	171.00	171.00
mean	0.63	36.72	0.73
std	0.48	49.93	0.44
min	0.00	2.00	0.00
10%	0.00	5.00	0.00
25%	0.00	10.00	0.00
50%	1.00	25.00	1.00
75%	1.00	50.00	1.00
90%	1.00	75.00	1.00
99%	1.00	150.00	1.00
max	1.00	550.00	1.00

	fertilizerbuymarketkg	buyfertilizierpay
count	114.00	114.00
mean	37.12	28,266.71
std	26.90	21,945.02
min	1.00	5.00
10%	7.15	9,000.00
25%	15.00	15,000.00
50%	47.50	17,250.00
75%	50.00	30,000.00
90%	50.00	70,000.00
99%	100.00	85,000.00
max	100.00	90,000.00

NOTES: 52% hhs received coupons. 63% used feritlizer with an avg of 36 kg used Rememeber though that Konje told us that fertilizer arrived in the village too late... Not be surpsise if production/productivity is very low despite fertilizer

Top extreme values fertilizer ===================

	hhid	fertilizerkg	area_cultivated
191	1430	550.00	5.00

doesnt seem that extreme

other intermediates

note the measure of intermediates does not use value fertilizer but fertilizer expenditure.

=========== Summary Intermediate inputs ===============

All variables in MWK except kg_fertilizer.

	interm	value_fertilizer	fertilizerkg	spendseeds	spendpesticides
count	237.00	284.00	171.00	284.00	284.00
mean	24,322.28	22,110.92	36.72	8,344.96	605.65
std	28,338.14	42,679.00	49.93	14,264.83	4,197.35
min	5.00	0.00	2.00	0.00	0.00
10%	3,212.00	0.00	5.00	0.00	0.00
25%	5,700.00	0.00	10.00	0.00	0.00
50%	15,000.00	10,000.00	25.00	4,650.00	0.00
75%	33,500.00	30,000.00	50.00	10,000.00	0.00
90%	64,940.00	50,000.00	75.00	19,350.00	0.00
99%	122,944.00	150,000.00	150.00	61,870.00	12,425.00
max	210,000.00	550,000.00	550.00	150,000.00	60,000.00

===== SUMMARY AGRICULTURAL INPUTS in $ ================

	hh_area_plots	hh_value_plots	area_cultivated	k_farm	labor_N	\
count	284.00	284.00	284.00	284.00	284.00	
mean	2.11	649.91	0.00	18.46	3.38	
std	1.83	939.04	0.00	43.46	2.67	
min	0.00	0.00	0.00	0.00	0.00	
25%	1.00	95.06	0.00	4.28	2.00	
50%	2.00	380.23	0.00	9.03	3.00	
75%	3.00	760.46	0.00	17.11	4.00	
max	21.00	5,703.42	0.01	596.96	18.00	

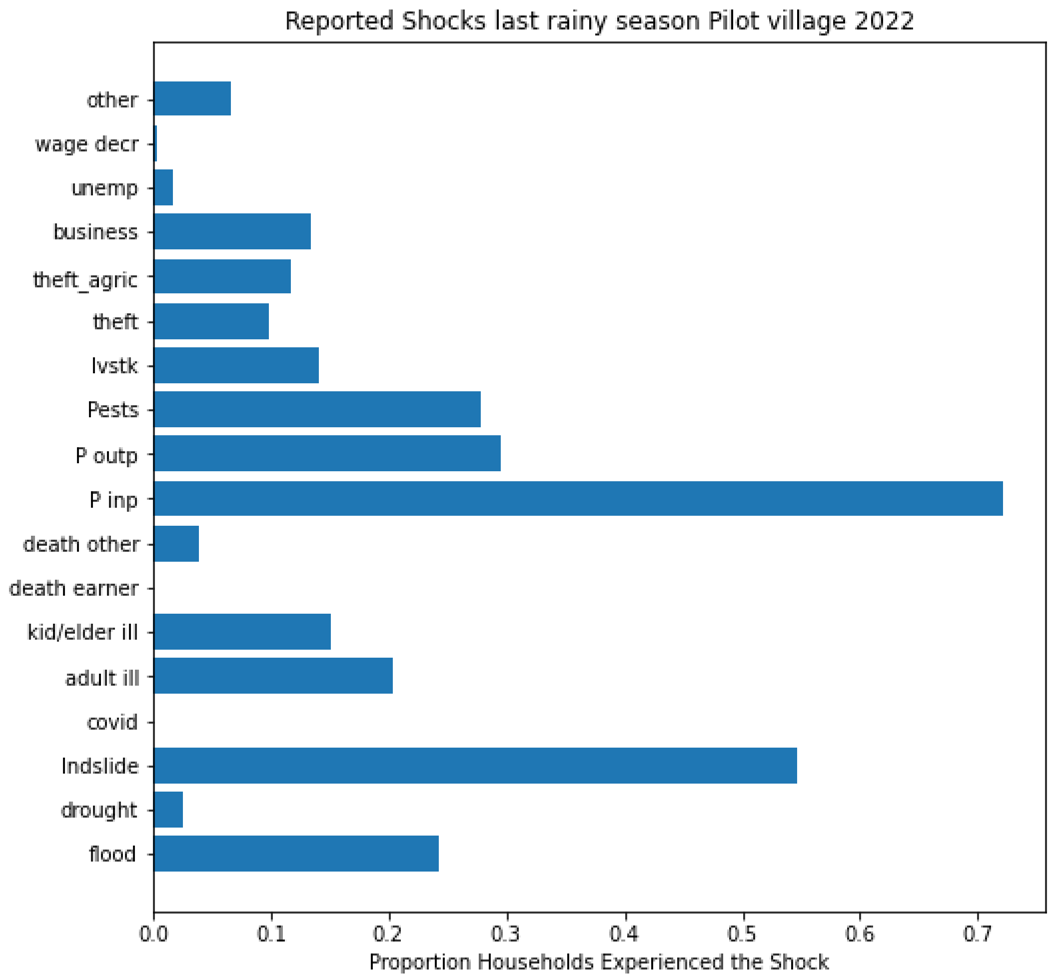
	labor_h	hh_labor_hours	hired_men_L	hired_women_L
count	284.00	284.00	47.00	29.00
mean	836.25	770.11	275.47	168.79
std	688.31	641.35	429.17	300.74
min	0.00	0.00	1.00	1.00
25%	384.00	336.00	24.00	16.00
50%	661.00	598.50	64.00	48.00
75%	1,152.00	1,051.00	372.00	150.00
max	3,992.00	3,920.00	2,240.00	1,200.00

	hired_kids_L	interm	value_fertilizer	spendseeds	spendpesticides
count	17.00	237.00	284.00	284.00	284.00
mean	55.47	23.12	21.02	7.93	0.58
std	60.44	26.94	40.57	13.56	3.99
min	1.00	0.00	0.00	0.00	0.00
25%	14.00	5.42	0.00	0.00	0.00
50%	36.00	14.26	9.51	4.42	0.00
75%	48.00	31.84	28.52	9.51	0.00
max	192.00	199.62	522.81	142.59	57.03

CASH TRANSFER

```
==============================================================
Conditional Cash Transfer Program Implementation in the Village.
==============================================================
       cashtrans_yes  cashtrans_value
count         284.00            26.00
mean            0.09       194,576.92
std             0.29       203,803.08
min             0.00        14,000.00
25%             0.00       100,000.00
50%             0.00       164,000.00
75%             0.00       248,000.00
max             1.00     1,100,000.00
```

SHOCKS



lABOR INCOME: SALARY LABOR

```
==============================================================
 Salary labor (1 month)
==============================================================
Numb. households with a sallary job: 6
       wlabor_supply  wlabor_inc_dollar
count           6.00               6.00
mean          135.67              68.92
std            75.73              64.13
min            48.00              19.01
25%            76.50              25.43
50%           132.00              38.97
75%           186.00              97.43
max           240.00             178.71
```

lABOR INCOME: GANYU LABOR

```
Ganyu income coming from network data
Numb. households did Ganyu: 69
in 2023 we had 103 hhs in ganyu. In february we had 160.
Asking via network might make us lose some observations in ganyu. Note timing interv
iew was also diferent, july 2022 was in September
===============================================================
Ganyu summary at the household level. Last month
===============================================================
```

	ganyu_yes	ganyu_inc	ganyu.days	ganyu_inc_dollar
count	284.00	69.00	69.00	69.00
mean	0.24	17,985.07	10.10	17.46
std	0.43	18,457.78	9.56	17.91
min	0.00	250.00	1.00	0.24
25%	0.00	4,000.00	3.00	3.88
50%	0.00	12,000.00	6.00	11.65
75%	0.00	25,000.00	14.00	24.26
max	1.00	80,000.00	40.00	77.64

BUSINESS INCOME

types of business in the village. Values in $, last month

	hhid	business_type	business_revenue \
16	1020	Selling fish	47.53
17	1021	Selling airtime	99.81
18	1022	Selling washing soap	39.35
21	1025	Selling fish	34.22
22	1026	Selling groceries	66.54
25	1029	Selling soap	23.29
30	1036	Selling charcoal	76.05
35	1042	Selling liquor	59.89
57	1118	Grocery	33.27
69	1133	Grocery shop	35.17
72	1138	Selling fish	193.92
80	1201	Selling fish, selling charcoal.	47.53
81	1202	Selling banana flitters and sugarcane	7.60
84	1205	Selling banana flitters	50.38
86	1207	Motorbike taxi	28.52
87	1208	Selling tomatoes and fish	104.56
89	1211	Grocery shop	527.57
102	1226	Selling of sweet potatoes	142.59
104	1228	Selling of fish	30.42
109	1233	Motorbike taxi	285.17
127	1309	Selling fish	57.03
134	1318	Has a Grocery Shop	427.76
135	1319	Selling wrappers, selling groundnuts	237.64
138	1322	Selling fish	47.53
139	1323	Selling of charcoal	13.31
152	1338	Kukhoma zidebe	28.52
153	1339	Selling of tondido	48.48
154	1340	Glocery	90.30
159	1346	Selling tomatoes	68.44
162	1349	Selling of fish	142.59
172	1409	Selling fish	51.33
175	1412	Selling fish,selling charcoal.	161.60
178	1416	Selling fish	17.11
188	1427	Motorbike taxi	190.11
190	1429	Selling of shoes	247.15
194	1433	Selling charcoal	0.00
195	1434	Sewing clothes	0.00
199	1438	Grocery	142.59
202	1441	Selling of clothes and glocery	570.34
204	1443	Selling smoked fish	28.90
214	1500	Carpentry	95.06
215	1501	Airtel money Agent	190.11
217	1503	Selling samsa, zigege, kanyenya, mandazi	23.76
218	1504	Selling of fish, mandazi, and kanyenya	52.28
221	1507	Selling bananas	0.00
223	1510	Fixing bicycles	85.55
228	1518	Selling dry fish	128.33

229	1520	Selling of matabwa	22.81
230	1521	Selling fish	47.53
235	1527	Selling of fish, beans and rice	66.54
242	1538	Carpentery	114.07
244	1543	Selling tomatoes	26.62
245	1544	Selling charcoal	91.25
246	1545	Selling Mops	26.62
247	1546	Selling of bananas	11.41
252	2001	Restaurant	46.91
261	2016	Selling fish	17.11
264	2020	Selling of charcoal	45.63
265	2022	Selling flour coated fish	114.07
266	2023	Motorbike taxi	142.59
267	2026	Kabaza	47.53
273	3005	Selling of rice, and samsa	118.82
277	3009	Bicycle taxi	142.59
279	3011	Motor bike taxi	399.24

	business_costs	business_profits
16	1.90	9.51
17	95.06	4.75
18	31.75	7.60
21	5.70	9.51
22	47.53	19.01
25	19.01	4.28
30	19.01	57.03
35	42.78	17.11
57	7.13	14.26
69	19.01	16.16
72	136.88	57.03
80	0.00	19.01
81	2.85	4.75
84	28.52	21.39
86	14.26	14.26
87	38.02	57.03
89	319.39	208.17
102	142.59	85.55
104	22.81	11.41
109	133.08	152.09
127	34.22	22.81
134	5.70	47.53
135	4.75	38.02
138	33.27	14.26
139	3.80	9.51
152	17.11	11.41
153	39.92	8.56
154	38.02	52.28
159	0.00	14.26
162	114.07	47.53
172	3.80	11.41
175	15.21	66.54
178	1.90	9.51
188	118.82	71.29
190	190.11	57.03
194	0.00	0.00
195	0.00	0.00
199	3.80	38.02
202	570.34	190.11
204	21.29	7.60
214	53.23	41.83
215	0.00	33.27
217	14.26	9.51
218	30.42	21.86
221	0.00	1.90
223	30.42	52.28
228	52.28	76.05
229	7.60	15.21
230	28.52	19.01
235	47.53	19.01

242	57.03	57.03
244	19.01	7.60
245	60.84	30.42
246	9.51	11.41
247	3.80	7.60
252	27.90	19.01
261	7.60	9.51
264	26.62	19.01
265	85.55	28.52
266	30.89	104.09
267	19.01	47.53
273	95.06	23.76
277	47.53	95.06
279	52.28	346.96

==

Summary Business income, month level, in dollars

==

	business_revenue	business_costs	business_profits	business_profits2
count	61.00	58.00	62.00	59.00
mean	107.55	54.32	42.02	57.80
std	119.14	88.19	57.15	80.82
min	7.60	1.90	1.90	4.28
25%	34.22	14.26	11.41	14.73
50%	59.89	29.47	19.01	26.14
75%	142.59	52.28	52.28	57.03
max	570.34	570.34	346.96	422.05

==

Salary Labor, Ganyu labor, and Business summary (at rainy season, 7 months)

==

income in dollars

	wlabor_inc_dollar	wlabor_supply	ganyu_inc	ganyu_supply \
count	6.00	6.00	69.00	69.00
mean	68.92	949.67	125,895.51	70.71
std	64.13	530.11	129,204.47	66.94
min	19.01	336.00	1,750.00	7.00
25%	25.43	535.50	28,000.00	21.00
50%	38.97	924.00	84,000.00	42.00
75%	97.43	1,302.00	175,000.00	98.00
max	178.71	1,680.00	560,000.00	280.00

	business_profits	business_profits2
count	62.00	59.00
mean	207,991.26	300,283.19
std	368,040.48	556,330.28
min	1,166.67	4,666.67
25%	35,000.00	44,333.33
50%	84,000.00	87,500.00
75%	275,625.00	285,833.33
max	2,342,083.33	3,108,000.00

GOV, NGO TRANSFERS AND REMITTANCES

	cashtrans_value	NGO_trans	gov_trans	remittances
count	26.00	10.00	6.00	99.00
mean	184.96	77.72	40.40	61.98
std	193.73	79.71	39.66	124.23
min	13.31	0.00	5.70	1.90
25%	95.06	18.77	27.09	14.26
50%	155.89	62.64	28.52	28.52
75%	235.74	118.82	32.79	68.92
max	1,045.63	253.80	118.82	1,140.68

1000$ of cash transfer might be too much? This is a government transfer

AGGREGATE INCOME

For the moment agricultural income is output minus hired wages. I do not substract for values intermediates (need to discuss how we measure them)

==

Summary total Income (rainy season)

```
================================================================
values in $
    index  inctotal  inctotal_trans    y_net  y_agric  y_maize  y_groundnut  \
0   count    274.00          278.00   267.00   267.00   250.00       131.00
1    mean    333.94          372.17   250.62   255.01   142.95       132.74
2     std    435.61          461.57   378.51   383.41   293.78       167.56
3     min      3.80            4.75   -43.71     2.73     6.72         4.75
4      1%     11.29           10.04     4.90     5.47     8.37         6.18
5     10%     57.39           62.90    42.98    43.94    16.80        23.76
6     25%    113.28          119.56    78.83    80.56    33.61        47.53
7     50%    234.98          253.69   160.33   161.75   100.82        95.06
8     75%    371.57          433.94   284.27   290.44   168.03       142.59
9     90%    675.25          731.02   514.36   514.36   268.85       285.17
10    99%  2,228.66        2,274.87 1,563.96 1,600.52   857.29       625.48
11    max  4,639.38        4,829.50 4,574.74 4,639.38 4,099.93     1,425.86
12   gini      0.51            0.50     0.64     0.53     0.55         0.52

    wlabor_inc  ganyu_inc  business_profits  other_inc
0         6.00      69.00             62.00     128.00
1       482.41     119.67            197.71      93.48
2       448.89     122.82            349.85     151.67
3       133.08       1.66              1.11       0.00
4       134.74       3.70              3.14       2.54
5       149.71      13.31              9.09       9.51
6       177.99      26.62             33.27      19.01
7       272.81      79.85             79.85      40.40
8       682.03     166.35            262.00     118.82
9     1,024.71     306.08            365.80     215.40
10    1,228.33     509.70          1,680.05     879.83
11    1,250.95     532.32          2,226.31   1,140.68
12        0.44        nan              0.65       0.61
AGAIN A VERY LOW INCOME... DUE TO VERY LOW AGRICULTURAL PRODUCTION.
This is consistent though with the bad harvest in Malawi, the expensive and messy di
stribution of fertilizers, and the stories villagers reported us


  WEALTH

======== Summary Wealth ==========
         wtotal  housing  hh_assets  hh_value_plots  k_farm  hhlivestock
count    284.00   284.00     284.00          284.00  284.00       284.00
mean   1,381.38   576.30      86.42          649.91   18.46        50.30
std    1,511.01   830.23     174.38          939.04   43.46       135.00
min        0.00     0.00       0.00            0.00    0.00         0.00
25%      443.23    95.06       2.85           95.06    4.28         0.00
50%      855.51   285.17      25.19          380.23    9.03         9.51
75%    1,757.37   760.46      83.06          760.46   17.11        55.61
max   11,263.59 7,604.56   1,457.22        5,703.42  596.96     1,948.67
================================================================
Summary total Income (year level)
================================================================
values in $
    index  inctotal  inctotal_trans    y_net  y_agric  y_maize  y_groundnut  \
0   count    274.00          278.00   267.00   267.00   250.00       131.00
1    mean    394.96          463.06   250.62   255.01   142.95       132.74
2     std    527.90          575.47   378.51   383.41   293.78       167.56
3     min      3.80            5.47   -43.71     2.73     6.72         4.75
4      1%     11.29           12.22     4.90     5.47     8.37         6.18
5     10%     59.74           73.23    42.98    43.94    16.80        23.76
6     25%    120.30          139.79    78.83    80.56    33.61        47.53
7     50%    255.65          298.41   160.33   161.75   100.82        95.06
8     75%    436.85          554.18   284.27   290.44   168.03       142.59
9     90%    787.83          901.10   514.36   514.36   268.85       285.17
10    99%  2,717.41        2,734.53 1,563.96 1,600.52   857.29       625.48
11    max  4,639.38        4,965.29 4,574.74 4,639.38 4,099.93     1,425.86
12   gini      0.53            0.52     0.64     0.53     0.55         0.52

    wlabor_inc  ganyu_inc  business_profits  other_inc
0         6.00      69.00             62.00     128.00
```

```
1       827.00     205.15           338.93     160.24
2       769.53     210.55           599.74     260.00
3       228.14       2.85             1.90       0.00
4       230.99       6.34             5.38       4.36
5       256.65      22.81            15.59      16.30
6       305.13      45.63            57.03      32.59
7       467.68     136.88           136.88      69.26
8     1,169.20     285.17           449.14     203.69
9     1,756.65     524.71           627.09     369.26
10    2,105.70     873.76         2,880.09   1,508.28
11    2,144.49     912.55         3,816.54   1,955.46
12        0.44       0.52             0.65       0.61


-----------------------------
 DATA SAVED in July 2023/Data/Clean data/income
-----------------------------
in rainy season level (7 months) income_wealth_23_rainseas.csv
in yearly level income_wealth_23_year.csv
-----------------------------
dataset contains the variables:
 'hhid',  'wave','rightsellland', 'chiefpreventsell', 'chiefpreventbequeat', 'cashtr
ans_yes', 'govcoupon' ,
 'inctotal', 'inctotal_trans','y_net','y_agric', 'y_maize', 'y_groundnut', 'y_pigeon
peas', 'total_kg_maize', 'total_kg_groundnut',  'total_kg_pigeonpeas',
 'y_cassava',  'y_soyabean', 'y_sorghum',  'y_fingermillet', 'y_cotton', 'y_tanapos
i', 'y_groundbean', 'y_nkhwani', 'y_sugarcane', 'y_sweetpotatoe',
 'sold_agric','sold_insiders_agric','store_agric',
 'hh_area_plots', 'hh_ratio_value_rent', 'hh_p_acre_plots' 'area_cultivated','k_far
m',
 'labor_N', 'labor_h', 'hh_labor_hours','hired_men_L', 'hired_women_L', 'hired_kids_
L', 'interm','fertilizerkg','p_fert','value_fertilizer', 'spendseeds', 'spendpestici
des',
 'wlabor_inc', 'wlabor_supply', 'ganyu_yes', 'ganyu_inc', 'ganyu_supply','business_r
evenue', 'business_profits','business_profits2',
 'NGO_yes', 'cashtrans_yes', 'gov_yes', 'remittances_yes', 'other_inc', 'cashtrans_v
alue', 'NGO_trans', 'gov_trans', 'remittances',
 'wtotal','housing','hh_assets','hh_value_plots','k_farm','hhlivestock',
 'shocks', 'shock_flood','shock_drought','shock_lndslide','shock_covid',
 'shock_adultill','shock_kidill','shock_death_earner','shock_death_othermemb','shock
_inp_p','shock_out_p','shock_pests','shock_lvstk','shock_theft','shock_theft_agri
c','shock_business','shock_unemp','shock_wage_decr','shock_other'
-----------------------------
y_net is agricultucal net income (minus intermediates). (MWK)
y_agric is gross agricultural income (MWK)
Labor variables: N denotes unit is number of persons. labor_h, denotes total labor i
nput (hh+hired) in hours.
Shock variables: whether households reported the shock or not.
```

In [ ]: