Effect of providing livestock feed during critical dry periods on the household milk yield among pastoralist communities in northern Kenya

Josphat Muema1,2,3, Nyamai Mutono2,3,4, Joseph Njuguna5, Christine Jost6,7, Erin Boyd6, Daniel Tewoldeberhan8, Immaculate Mutua9, George Gacharamu8, Francis Wambua8, Emmanuella Olesambu10, Abdal Monium Osman10, Darana Souza10, Irene Kimani5, Julius Oyugi1,2, Zipporah Bukania2,11, Anita makori3,4, Guy H. Palmer1,3,12, Jonathan Yoder 2,13, SM Thumbi 1,2,4,12#

1 Institute of Tropical and Infectious Diseases, University of Nairobi, Nairobi, Kenya

2 Feed the Future Innovation Lab for Animal Health, Washington State University

3 Washington State University Global Health Program- Kenya, Nairobi, Kenya

4 Center for Epidemiological Modelling and Analysis, University of Nairobi

5 Food and Agriculture Organization of the United Nations, Nairobi, Kenya

6 United States Agency for International Development’s Bureau for Humanitarian Assistance (USAID/BHA), Washington, DC

7 Global Health Support Initiative III, Social Solutions International, Washington DC

8 United Nations Children’s Fund, Nairobi, Kenya

9 Ministry of Health, Government of Marsabit County, Marsabit, Kenya

10 Emergency and Resilience Division, Food and Agriculture Organization of the United Nations, Rome, Italy

11 Centre for Public Health, Kenya Medical Research Institute, Nairobi, Kenya

12 Paul G. Allen School for Global Health, Washington State University, Pullman, USA

13 School of Economic Sciences, Washington state University, Pullman, USA

# Corresponding author: SM Thumbi ([thumbi.mwangi@wsu.edu](mailto:thumbi.mwangi@wsu.edu))

# Abstract

# Introduction

# Methods

## Ethics statement

## Study setting

## Study design and participants

## Data collection

To understand the household milk yield in litres and the factors associated with it, we collected periodic data on the household demographics, socio-demographics, herd size (milk and non-milking), herd dynamics (births), veterinary interventions, periods of expected droughts, period the household received the intervention of feed, and the amount of milk produced at a household level. The data was collected every six weeks, from September 2019 till December 2021. Baseline data was collected during the first six weeks of the study period.

## Data analysis

We carried out descriptive analysis by assessing the daily milk yield for households in both the intervention and control arms for the different seasons (critical dry periods and non-dry seasons). The milk yield was also assessed for the different animal species, and the tropical livestock unit (TLU) at a household level. The tropical livestock unit is a way of standardising comparison of animals where one cow/camel represents 10 goats/ sheep. The animal births for each species were also calculated during the study period.

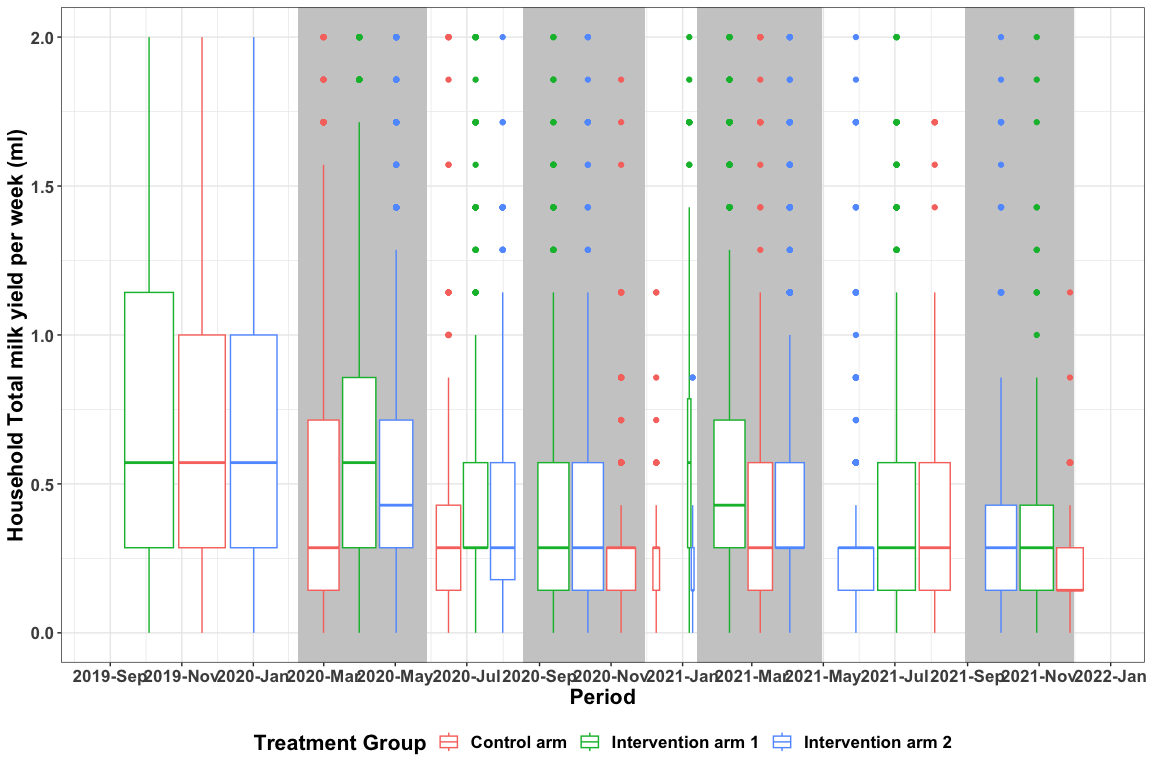
To assess the factors associated with milk yield at a household level, we carried out a mixed effects multivariate analysis. A univariate model as conducted for variables that included the period a household received feed, number of milked and non-milked TLUs, number of births, number of people in the household, whether the household herds received any veterinary intervention, the occupation of the household head, and the socio-economic index. Variables that had a P value of <0.2 were included in the mixed effects multivariate model which employed a gaussian distribution. The household identification number and number of days since the household was recruited into the study were used as fixed effects.

The socio-economic index was calculated using assets collected during the study period which included ownership of a mobile phone (household head), small electronics (e.g. radio), large electronics (e.g. television), simple transport (e.g. bicycle), large transport (e.g. motorbike), main source of lighting, main toilet facility, main material for external roofind, exterior walls and floor. Principal component analysis using the singular value decomposition method which assesses correlations between individuals was implemented. Factor score for each variable during each visit was calculated. Quantiles for each visit were used to categorise households into five quantiles, ranging from quantile 1 (highest)to quantile 5 (lowest).

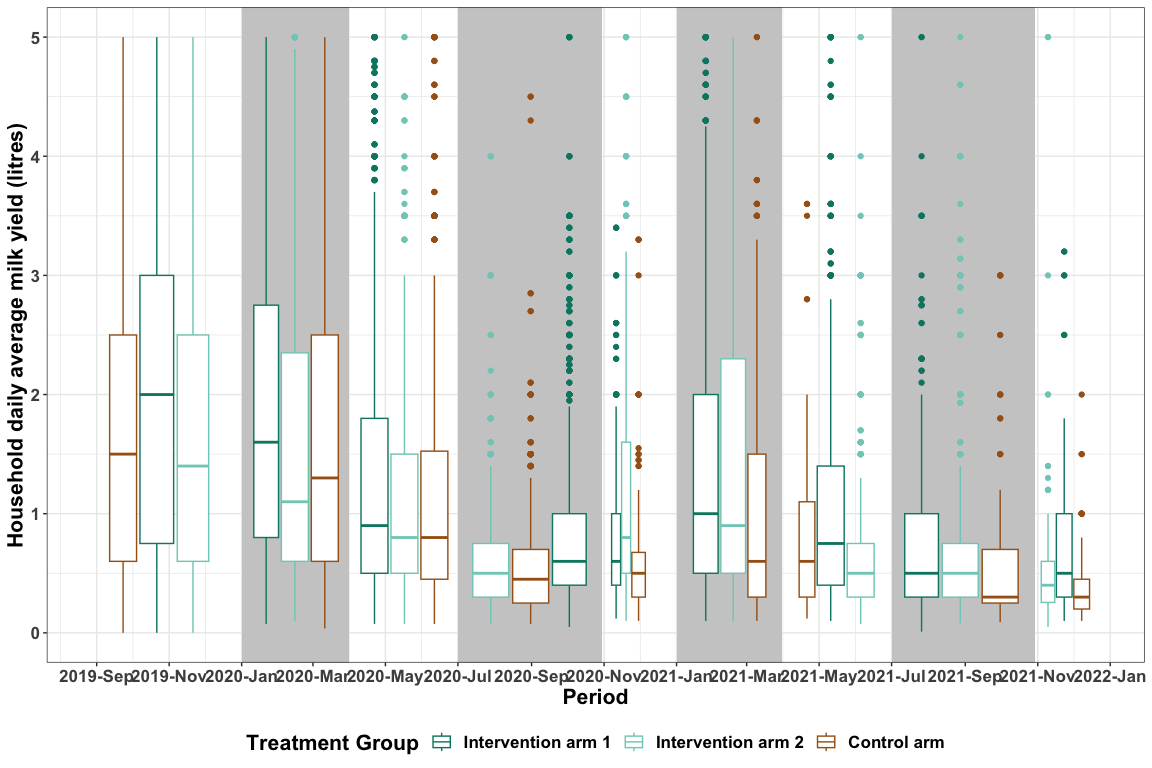
# Results

## Feed distribution

## Warning: Removed 916 rows containing non-finite values (`stat\_boxplot()`).



## Warning: Removed 1322 rows containing non-finite values (`stat\_boxplot()`).



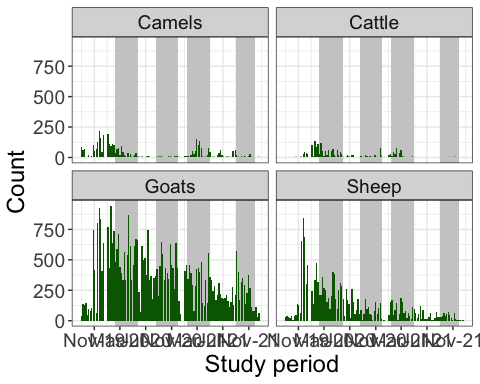
## milking animals   
  
lct\_animals1 <- household\_baseline|>  
 clean\_names()|>  
 #filter(hh\_eligible%in%"Yes")|>  
 mutate(visit=1)|>  
 # mutate(interview\_date=dmy(interview\_date)) |>  
 dplyr::select(interview\_date,household\_id,cwslactatn,shplactatn,gtslactatn,cmlslactatn, visit)#|>  
   
lct\_animals2 <- household\_quarterly|>  
# filter(hh\_eligible%in%"Yes")|>  
 # mutate(interview\_date=dmy(interview\_date)) |>  
 mutate(visit=ifelse(interview\_date<as.Date("2020-02-15"), 3, ifelse(interview\_date>as.Date("2020-04-05") & interview\_date<as.Date("2020-05-20"), 5, ifelse(interview\_date>as.Date("2020-05-20") & interview\_date<as.Date("2020-09-12"), 7, ifelse(interview\_date>as.Date("2020-09-13") & interview\_date<as.Date("2020-12-16"), 9, ifelse(interview\_date>as.Date("2021-02-24") & interview\_date<as.Date("2021-04-23"), 11, ifelse(interview\_date>as.Date("2021-06-13") & interview\_date<as.Date("2021-08-25"), 13, ifelse(interview\_date>as.Date("2021-10-11"), 15, 99))))))))|>  
 dplyr::select(interview\_date,household\_id,cwslactatn,shplactatn,gtslactatn,cmlslactatn, visit)#|>  
  
## followup  
lct\_animals3 <- followup\_data|>  
 # filter(hh\_eligible%in%"Yes")|>  
 # mutate(interview\_date=dmy(interview\_date)) |>  
 mutate(visit=ifelse(interview\_date<as.Date("2020-01-16"), 2, ifelse(interview\_date>as.Date("2020-01-16") & interview\_date<as.Date("2020-04-03"), 4, ifelse(interview\_date>as.Date("2020-05-01") & interview\_date<as.Date("2020-07-28"), 6, ifelse(interview\_date>as.Date("2020-09-10") & interview\_date<as.Date("2020-10-28"), 8, ifelse(interview\_date>as.Date("2021-01-05") & interview\_date<as.Date("2021-02-25"), 10, ifelse(interview\_date>as.Date("2021-02-25") & interview\_date<as.Date("2021-07-07"), 12, ifelse(interview\_date>as.Date("2021-08-17"), 14, 99))))))))|>  
 dplyr::select(interview\_date,household\_id,cwslactatn,shplactatn,gtslactatn,cmlslactatn, visit)#|>  
  
## combine all the data  
  
lct\_animals <- lct\_animals1|>  
 rbind(lct\_animals2, lct\_animals3)|>  
 group\_by(household\_id)|>  
 arrange(interview\_date)|>  
 # mutate(interview\_date=dmy(interview\_date)) |>  
 mutate(days\_recruitment=interview\_date-dplyr::lag(interview\_date))|>  
 mutate(days\_recruitment=ifelse(is.na(days\_recruitment), 0, days\_recruitment))|>  
 ungroup()|>  
 gather(c(cwslactatn,shplactatn,gtslactatn,cmlslactatn), key="milking\_species", value="milking\_animals")|>  
 mutate(milking\_species=recode(milking\_species,"cwslactatn"="cattle","shplactatn"="sheep","gtslactatn"="goats","cmlslactatn"="camels"))|>  
 mutate(milking\_animals=as.numeric(milking\_animals)) |>  
 mutate(tlu=ifelse(milking\_species%in%"cattle",(milking\_animals\*1),ifelse(milking\_species%in%"camels",(milking\_animals\*1),(milking\_animals\*0.1))))|>  
 # group\_by(days\_recruitment,household\_id,milking\_species)|>  
 # mutate(milking\_animals1=round(sum(milking\_animals, na.rm=T)))|>  
 # mutate(tlu1=sum(tlu, na.rm=T))|>  
 # ungroup()|>  
 group\_by(days\_recruitment,household\_id)|>  
 mutate(total\_milk\_animals=round(sum(milking\_animals, na.rm=T)))|>  
 mutate(total\_tlu=sum(tlu, na.rm=T))|>  
 ungroup()|>  
 dplyr::select(household\_id,total\_milk\_animals,total\_tlu, days\_recruitment, visit)|>  
 distinct()

## Warning: There was 1 warning in `mutate()`.  
## ℹ In argument: `milking\_animals = as.numeric(milking\_animals)`.  
## Caused by warning:  
## ! NAs introduced by coercion

lct\_animals\_a <- lct\_animals1|>  
 rbind(lct\_animals2, lct\_animals3)|>  
 gather(c(cwslactatn,shplactatn,gtslactatn,cmlslactatn), key="milking\_species", value="milking\_animals")|>  
 mutate(milking\_species=recode(milking\_species,"cwslactatn"="cattle","shplactatn"="sheep","gtslactatn"="goats","cmlslactatn"="camels"))|>  
 mutate(milking\_animals=as.numeric(milking\_animals)) |>  
 mutate(tlu=ifelse(milking\_species%in%"cattle",(milking\_animals\*1),ifelse(milking\_species%in%"camels",(milking\_animals\*1),(milking\_animals\*0.1))))|>  
 # mutate(interview\_date=dmy(interview\_date)) |>  
 mutate(interview\_week=floor\_date(interview\_date, "week"))|>  
 group\_by(interview\_week, milking\_species)|>  
 mutate(milking\_animals1=sum(milking\_animals, na.rm=T))|>  
 ungroup()|>  
 dplyr::select(interview\_week, milking\_species, milking\_animals1)|>  
 distinct()

## Warning: There was 1 warning in `mutate()`.  
## ℹ In argument: `milking\_animals = as.numeric(milking\_animals)`.  
## Caused by warning:  
## ! NAs introduced by coercion

lct\_animals\_a$milking\_species<- str\_to\_title(lct\_animals\_a$milking\_species)  
  
ggplot(lct\_animals\_a, aes(x=interview\_week, y=milking\_animals1))+geom\_rect(aes(xmin=as.Date("2020-08-18"), xmax=as.Date("2020-11-30"), ymin=-Inf, ymax=Inf ), fill="grey80", color=NA)+geom\_rect(aes(xmin=as.Date("2021-01-13"), xmax=as.Date("2021-04-30"), ymin=-Inf, ymax=Inf ),fill="grey80", color=NA)+geom\_rect(aes(xmin=as.Date("2020-02-08"), xmax=as.Date("2020-05-28"), ymin=-Inf, ymax=Inf),fill="grey80", color=NA)+geom\_rect(aes(xmin=as.Date("2021-08-30"), xmax=as.Date("2021-12-01"), ymin=-Inf, ymax=Inf ),fill="grey80", color=NA)+geom\_col(fill="darkgreen")+facet\_wrap(.~milking\_species)+theme\_bw()+scale\_x\_date(date\_breaks="4 months", date\_labels = "%b-%y")+theme(text=element\_text(size=18))+labs(x="Study period", y="Count", fill="")



#ggsave("animals\_milked.png", width=18, height=10)  
###herd dynamics  
dynamics1 <- household\_baseline|>  
 clean\_names()|>  
 dplyr::select(interview\_date,household\_id,cwsbrth,shpbrth,goatsbrth,cmlsbrth,calves\_death,cows\_death,bulls\_death,sheep\_death,msheep\_death,fsheep\_death,goats\_death,mgoats\_death,fgoats\_death,camels\_death,mcamels\_death,fcamels\_death,cowsgft,sheepgfts,goatsgft,cmlsgft,cowsgvnout,sheepgvnout,goatsgvnout,cmlsgvnout,cttlebght,sheepbght,goatsbgh,cmlsbght,cttlesld,sheepsld,goatsld,cmlsold)|>  
 mutate(visit=1) |>  
 mutate\_at(vars(cwsbrth,shpbrth,goatsbrth,cmlsbrth,calves\_death,cows\_death,bulls\_death,sheep\_death,msheep\_death,fsheep\_death,goats\_death,mgoats\_death,fgoats\_death,camels\_death,mcamels\_death,fcamels\_death,cowsgft,sheepgfts,goatsgft,cmlsgft,cowsgvnout,sheepgvnout,goatsgvnout,cmlsgvnout,cttlebght,sheepbght,goatsbgh,cmlsbght,cttlesld,sheepsld,goatsld,cmlsold), funs(as.numeric(.)))

## Warning: `funs()` was deprecated in dplyr 0.8.0.  
## ℹ Please use a list of either functions or lambdas:  
##   
## # Simple named list: list(mean = mean, median = median)  
##   
## # Auto named with `tibble::lst()`: tibble::lst(mean, median)  
##   
## # Using lambdas list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

## Warning: There were 32 warnings in `mutate()`.  
## The first warning was:  
## ℹ In argument: `cwsbrth = as.numeric(cwsbrth)`.  
## Caused by warning:  
## ! NAs introduced by coercion  
## ℹ Run `dplyr::last\_dplyr\_warnings()` to see the 31 remaining warnings.

#mutate(cattle\_death=colSums(calves\_death,cows\_death,bulls\_death),na.rm=T)  
dynamics1$cattle\_out <- rowSums(dynamics1[ , c("calves\_death","cows\_death","bulls\_death","cowsgvnout","cttlesld")],na.rm = T)  
dynamics1$cattle\_in <- rowSums(dynamics1[ , c("cwsbrth","cowsgft","cttlebght")],na.rm = T)  
dynamics1$sheep\_out <- rowSums(dynamics1[ , c("sheep\_death","msheep\_death","fsheep\_death","sheepgvnout","sheepsld")],na.rm = T)  
dynamics1$sheep\_in <- rowSums(dynamics1[ , c("shpbrth","sheepgfts","sheepbght")],na.rm = T)  
dynamics1$goat\_out <- rowSums(dynamics1[ , c("goats\_death","mgoats\_death","fgoats\_death","goatsgvnout","goatsld")],na.rm = T)  
dynamics1$goat\_in <- rowSums(dynamics1[ , c("goatsbrth","goatsgft","goatsbgh")],na.rm = T)  
dynamics1$camel\_out <- rowSums(dynamics1[ , c("camels\_death","mcamels\_death","fcamels\_death","cmlsgvnout","cmlsold")],na.rm = T)  
dynamics1$camel\_in <- rowSums(dynamics1[ , c("cmlsbrth","cmlsgft","cmlsbght")],na.rm = T)  
#  
dynamics1 <- dynamics1|>  
 dplyr::select(interview\_date,household\_id,cattle\_out,cattle\_in,sheep\_out,sheep\_in,goat\_out,goat\_in,camel\_out,camel\_in, visit)  
  
dynamics2 <- followup\_data|>  
 clean\_names()|>  
 dplyr::select(interview\_date,household\_id,cwsbrth,shpbrth,goatsbrth,cmlsbrth,calves\_death,cows\_death,bulls\_death,sheep\_death,msheep\_death,fsheep\_death,goats\_death,mgoats\_death,fgoats\_death,camels\_death,mcamels\_death,fcamels\_death,calves\_gift,cows\_gifts,bulls\_gifts,sheep\_gifts,msheep\_gifts,fsheep\_gifts,goats\_gifts,mgoats\_gifts,fgoats\_gifts,camels\_gifts,mcamels\_gifts,fcamels\_gifts,calves\_givenout,bulls\_givenout,cows\_givenout,sheep\_givenout,msheep\_givenout,fsheep\_givenout,goats\_givenout,mgoats\_givenout,fgoats\_givenout,camels\_givenout,mcamels\_givenout,fcamels\_givenout,cttlebght,sheepbght,goatsbgh,cmlsbght,calves\_sold,bulls\_sold,cows\_sold,sheep\_sold,msheep\_sold,fsheep\_sold,goats\_sold,mgoats\_sold,fgoats\_sold,camels\_sold,mcamels\_sold,fcamels\_sold)|>  
 # mutate(interview\_date=dmy(interview\_date)) |>  
 mutate(visit=ifelse(interview\_date<as.Date("2020-01-16"), 2, ifelse(interview\_date>as.Date("2020-01-16") & interview\_date<as.Date("2020-04-03"), 4, ifelse(interview\_date>as.Date("2020-05-01") & interview\_date<as.Date("2020-07-28"), 6, ifelse(interview\_date>as.Date("2020-09-10") & interview\_date<as.Date("2020-10-28"), 8, ifelse(interview\_date>as.Date("2021-01-05") & interview\_date<as.Date("2021-02-25"), 10, ifelse(interview\_date>as.Date("2021-02-25") & interview\_date<as.Date("2021-07-07"), 12, ifelse(interview\_date>as.Date("2021-08-17"), 14, 99)))))))) |>  
 mutate\_at(vars(contains("death"),contains("givenout"),contains("sold"),contains("brth"), contains("bgh"), contains("gift") ), funs(as.numeric(.)))

## Warning: `funs()` was deprecated in dplyr 0.8.0.  
## ℹ Please use a list of either functions or lambdas:  
##   
## # Simple named list: list(mean = mean, median = median)  
##   
## # Auto named with `tibble::lst()`: tibble::lst(mean, median)  
##   
## # Using lambdas list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

## Warning: There were 56 warnings in `mutate()`.  
## The first warning was:  
## ℹ In argument: `calves\_death = as.numeric(calves\_death)`.  
## Caused by warning:  
## ! NAs introduced by coercion  
## ℹ Run `dplyr::last\_dplyr\_warnings()` to see the 55 remaining warnings.

#mutate(cattle\_death=colSums(calves\_death,cows\_death,bulls\_death),na.rm=T)  
dynamics2$cattle\_out <- rowSums(dynamics2[ , c("calves\_death","cows\_death","bulls\_death","calves\_givenout","bulls\_givenout","cows\_givenout","calves\_sold","bulls\_sold" ,"cows\_sold")],na.rm = T)  
dynamics2$cattle\_in <- rowSums(dynamics2[ , c("cwsbrth","calves\_gift","cows\_gifts","bulls\_gifts","cttlebght")],na.rm = T)  
dynamics2$sheep\_out <- rowSums(dynamics2[ , c("sheep\_death","msheep\_death","fsheep\_death","sheep\_givenout","msheep\_givenout","fsheep\_givenout","sheep\_sold","sheep\_sold","fsheep\_sold")],na.rm = T)  
dynamics2$sheep\_in <- rowSums(dynamics2[ , c("shpbrth","sheep\_gifts","msheep\_gifts","fsheep\_gifts","sheepbght")],na.rm = T)  
dynamics2$goat\_out <- rowSums(dynamics2[ , c("goats\_death","mgoats\_death","fgoats\_death","goats\_givenout","mgoats\_givenout","fgoats\_givenout","goats\_sold","mgoats\_sold","fgoats\_sold")],na.rm = T)  
dynamics2$goat\_in <- rowSums(dynamics2[ , c("goatsbrth","goats\_gifts","mgoats\_gifts","fgoats\_gifts","goatsbgh")],na.rm = T)  
dynamics2$camel\_out <- rowSums(dynamics2[ , c("camels\_death","mcamels\_death","fcamels\_death","camels\_givenout","mcamels\_givenout","fcamels\_givenout","camels\_sold","mcamels\_sold","fcamels\_sold")],na.rm = T)  
dynamics2$camel\_in <- rowSums(dynamics2[ , c("cmlsbrth","camels\_gifts","mcamels\_gifts","fcamels\_gifts","cmlsbght")],na.rm = T)  
#  
dynamics2 <- dynamics2|>  
 dplyr::select(interview\_date,household\_id,cattle\_out,cattle\_in,sheep\_out,sheep\_in,goat\_out,goat\_in,camel\_out,camel\_in, visit)  
  
dynamics3 <- household\_quarterly|>  
 clean\_names()|>  
 dplyr::select(interview\_date,household\_id,cwsbrth,shpbrth,goatsbrth,cmlsbrth,calves\_death,cows\_death,bulls\_death,sheep\_death,msheep\_death,fsheep\_death,goats\_death,mgoats\_death,fgoats\_death,camels\_death,mcamels\_death,fcamels\_death,calves\_gift,cows\_gifts,bulls\_gifts,sheep\_gifts,msheep\_gifts,fsheep\_gifts,goats\_gifts,mgoats\_gifts,fgoats\_gifts,camels\_gifts,mcamels\_gifts,fcamels\_gifts,calves\_givenout,bulls\_givenout,cows\_givenout,sheep\_givenout,msheep\_givenout,fsheep\_givenout,goats\_givenout,mgoats\_givenout,fgoats\_givenout,camels\_givenout,mcamels\_givenout,fcamels\_givenout,calves\_prchsd,bulls\_purchased,cows\_purchased,sheep\_purchased,msheep\_purchased,fsheep\_purchased,goats\_purchased,mgoats\_purchased,fgoats\_purchased,camels\_purchased,mcamels\_purchased,fcamels\_purchased,calves\_sold,bulls\_sold,cows\_sold,sheep\_sold,msheep\_sold,fsheep\_sold,goats\_sold,mgoats\_sold,fgoats\_sold,camels\_sold,mcamels\_sold,fcamels\_sold)|>  
 mutate(visit=ifelse(interview\_date<as.Date("2020-02-15"), 3, ifelse(interview\_date>as.Date("2020-04-05") & interview\_date<as.Date("2020-05-20"), 5, ifelse(interview\_date>as.Date("2020-05-20") & interview\_date<as.Date("2020-09-12"), 7, ifelse(interview\_date>as.Date("2020-09-13") & interview\_date<as.Date("2020-12-16"), 9, ifelse(interview\_date>as.Date("2021-02-24") & interview\_date<as.Date("2021-04-23"), 11, ifelse(interview\_date>as.Date("2021-06-13") & interview\_date<as.Date("2021-08-25"), 13, ifelse(interview\_date>as.Date("2021-10-11"), 15, 99))))))))|>  
 mutate\_at(vars(contains("death"),contains("givenout"),contains("sold"),contains("brth"), contains("bgh"), contains("gift") ), funs(as.numeric(.)))

## Warning: `funs()` was deprecated in dplyr 0.8.0.  
## ℹ Please use a list of either functions or lambdas:  
##   
## # Simple named list: list(mean = mean, median = median)  
##   
## # Auto named with `tibble::lst()`: tibble::lst(mean, median)  
##   
## # Using lambdas list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

#mutate(cattle\_death=colSums(calves\_death,cows\_death,bulls\_death),na.rm=T)  
dynamics3$cattle\_out <- rowSums(dynamics3[ , c("calves\_death","cows\_death","bulls\_death","calves\_givenout","bulls\_givenout","cows\_givenout","calves\_sold","bulls\_sold" ,"cows\_sold")],na.rm = T)  
dynamics3$cattle\_in <- rowSums(dynamics3[ , c("cwsbrth","calves\_gift","cows\_gifts","bulls\_gifts","calves\_prchsd","bulls\_purchased","cows\_purchased")],na.rm = T)  
dynamics3$sheep\_out <- rowSums(dynamics3[ , c("sheep\_death","msheep\_death","fsheep\_death","sheep\_givenout","msheep\_givenout","fsheep\_givenout","sheep\_sold","sheep\_sold","fsheep\_sold")],na.rm = T)  
dynamics3$sheep\_in <- rowSums(dynamics3[ , c("shpbrth","sheep\_gifts","msheep\_gifts","fsheep\_gifts","sheep\_purchased","msheep\_purchased","fsheep\_purchased")],na.rm = T)  
dynamics3$goat\_out <- rowSums(dynamics3[ , c("goats\_death","mgoats\_death","fgoats\_death","goats\_givenout","mgoats\_givenout","fgoats\_givenout","goats\_sold","mgoats\_sold","fgoats\_sold")],na.rm = T)  
dynamics3$goat\_in <- rowSums(dynamics3[ , c("goatsbrth","goats\_gifts","mgoats\_gifts","fgoats\_gifts","goats\_purchased","mgoats\_purchased","fgoats\_purchased")],na.rm = T)  
dynamics3$camel\_out <- rowSums(dynamics3[ , c("camels\_death","mcamels\_death","fcamels\_death","camels\_givenout","mcamels\_givenout","fcamels\_givenout","camels\_sold","mcamels\_sold","fcamels\_sold")],na.rm = T)  
dynamics3$camel\_in <- rowSums(dynamics3[ , c("cmlsbrth","camels\_gifts","mcamels\_gifts","fcamels\_gifts","camels\_purchased","mcamels\_purchased","fcamels\_purchased")],na.rm = T)  
#  
dynamics3 <- dynamics3|>  
 dplyr::select(interview\_date,household\_id,cattle\_out,cattle\_in,sheep\_out,sheep\_in,goat\_out,goat\_in,camel\_out,camel\_in,visit)  
dynamics <- rbind(dynamics1,dynamics2,dynamics3)|>  
 group\_by(household\_id)|>  
 arrange(interview\_date)|>  
 mutate(days\_recruitment=interview\_date-dplyr::lag(interview\_date))|>  
 ungroup()|>  
 group\_by(household\_id, days\_recruitment)|>  
 mutate\_at(vars(cattle\_in, cattle\_out, sheep\_in, sheep\_out, goat\_in, goat\_out, camel\_in, camel\_out), funs(sum(., na.rm=T)))|>  
 ungroup()|>  
 distinct()

## Warning: `funs()` was deprecated in dplyr 0.8.0.  
## ℹ Please use a list of either functions or lambdas:  
##   
## # Simple named list: list(mean = mean, median = median)  
##   
## # Auto named with `tibble::lst()`: tibble::lst(mean, median)  
##   
## # Using lambdas list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

##livestock ownership (herd structure)  
lvstckown1 <- household\_baseline|>  
 clean\_names()|>  
 # filter(hh\_eligible%in%"Yes")|>  
 dplyr::select(interview\_date,household\_id,calvsnum,bullsnum,cowsnum,sheep1num,shpmlenum,shpfmenum,goatsnum,gtsmlenum,gtsfmlenum,cmels1num,cmelsmlenum,cmelsfmlenum,study\_arm)|>  
 pivot\_longer(c(calvsnum,bullsnum,cowsnum,sheep1num,shpmlenum,shpfmenum,goatsnum,gtsmlenum,gtsfmlenum,cmels1num,cmelsmlenum,cmelsfmlenum), names\_to="species\_owned", values\_to="animals\_owned")|>  
 #mutate(interview\_week=floor\_date(interview\_date,"week"))|>  
 filter(!is.na(animals\_owned))|>  
 mutate(species\_owned=recode(species\_owned,"calvsnum"="cattle","bullsnum"="cattle","cowsnum"="cattle","sheep1num"="sheep","shpmlenum"="sheep","shpfmenum"="sheep","goatsnum"="goats","gtsmlenum"="goats","gtsfmlenum"="goats","cmels1num"="camels","cmelsmlenum"="camels","cmelsfmlenum"="camels"))|>  
 group\_by(household\_id, species\_owned)|>  
 mutate(animals\_owned=sum(as.numeric(animals\_owned), na.rm=T))|>  
 ungroup()|>  
 distinct()

## Warning: There were 3260 warnings in `mutate()`.  
## The first warning was:  
## ℹ In argument: `animals\_owned = sum(as.numeric(animals\_owned), na.rm = T)`.  
## ℹ In group 2: `household\_id = "KOHFGAHH001"`, `species\_owned = "cattle"`.  
## Caused by warning:  
## ! NAs introduced by coercion  
## ℹ Run `dplyr::last\_dplyr\_warnings()` to see the 3259 remaining warnings.

## herd dynamics  
   
dynamics1 <- household\_baseline|>  
 clean\_names()|>  
 dplyr::select(interview\_date,household\_id,cwsbrth,shpbrth,goatsbrth,cmlsbrth,calves\_death,cows\_death,bulls\_death,sheep\_death,msheep\_death,fsheep\_death,goats\_death,mgoats\_death,fgoats\_death,camels\_death,mcamels\_death,fcamels\_death,cowsgft,sheepgfts,goatsgft,cmlsgft,cowsgvnout,sheepgvnout,goatsgvnout,cmlsgvnout,cttlebght,sheepbght,goatsbgh,cmlsbght,cttlesld,sheepsld,goatsld,cmlsold)#|>  
   
dynamics2 <- followup\_data|>  
 clean\_names()|>  
 dplyr::select(interview\_date,household\_id,cwsbrth,shpbrth,goatsbrth,cmlsbrth,calves\_death,cows\_death,bulls\_death,sheep\_death,msheep\_death,fsheep\_death,goats\_death,mgoats\_death,fgoats\_death,camels\_death,mcamels\_death,fcamels\_death,calves\_gift,cows\_gifts,bulls\_gifts,sheep\_gifts,msheep\_gifts,fsheep\_gifts,goats\_gifts,mgoats\_gifts,fgoats\_gifts,camels\_gifts,mcamels\_gifts,fcamels\_gifts,calves\_givenout,bulls\_givenout,cows\_givenout,sheep\_givenout,msheep\_givenout,fsheep\_givenout,goats\_givenout,mgoats\_givenout,fgoats\_givenout,camels\_givenout,mcamels\_givenout,fcamels\_givenout,cttlebght,sheepbght,goatsbgh,cmlsbght,calves\_sold,bulls\_sold,cows\_sold,sheep\_sold,msheep\_sold,fsheep\_sold,goats\_sold,mgoats\_sold,fgoats\_sold,camels\_sold,mcamels\_sold,fcamels\_sold)#|>  
 # mutate(visit=ifelse(interview\_date<as.Date("2020-01-16"), 2, ifelse(interview\_date>as.Date("2020-01-16") & interview\_date<as.Date("2020-04-03"), 4, ifelse(interview\_date>as.Date("2020-05-01") & interview\_date<as.Date("2020-07-28"), 6, ifelse(interview\_date>as.Date("2020-09-10") & interview\_date<as.Date("2020-10-28"), 8, ifelse(interview\_date>as.Date("2021-01-05") & interview\_date<as.Date("2021-02-25"), 10, ifelse(interview\_date>as.Date("2021-02-25") & interview\_date<as.Date("2021-07-07"), 12, ifelse(interview\_date>as.Date("2021-08-17"), 14, 99))))))))  
 #mutate(cattle\_death=colSums(calves\_death,cows\_death,bulls\_death),na.rm=T)  
  
dynamics3 <- household\_quarterly|>  
 clean\_names()|>  
 dplyr::select(interview\_date,household\_id,cwsbrth,shpbrth,goatsbrth,cmlsbrth,calves\_death,cows\_death,bulls\_death,sheep\_death,msheep\_death,fsheep\_death,goats\_death,mgoats\_death,fgoats\_death,camels\_death,mcamels\_death,fcamels\_death,calves\_gift,cows\_gifts,bulls\_gifts,sheep\_gifts,msheep\_gifts,fsheep\_gifts,goats\_gifts,mgoats\_gifts,fgoats\_gifts,camels\_gifts,mcamels\_gifts,fcamels\_gifts,calves\_givenout,bulls\_givenout,cows\_givenout,sheep\_givenout,msheep\_givenout,fsheep\_givenout,goats\_givenout,mgoats\_givenout,fgoats\_givenout,camels\_givenout,mcamels\_givenout,fcamels\_givenout,calves\_prchsd,bulls\_purchased,cows\_purchased,sheep\_purchased,msheep\_purchased,fsheep\_purchased,goats\_purchased,mgoats\_purchased,fgoats\_purchased,camels\_purchased,mcamels\_purchased,fcamels\_purchased,calves\_sold,bulls\_sold,cows\_sold,sheep\_sold,msheep\_sold,fsheep\_sold,goats\_sold,mgoats\_sold,fgoats\_sold,camels\_sold,mcamels\_sold,fcamels\_sold)#|>  
 # mutate(visit=ifelse(interview\_date<as.Date("2020-02-15"), 3, ifelse(interview\_date>as.Date("2020-04-05") & interview\_date<as.Date("2020-05-20"), 5, ifelse(interview\_date>as.Date("2020-05-20") & interview\_date<as.Date("2020-09-12"), 7, ifelse(interview\_date>as.Date("2020-09-13") & interview\_date<as.Date("2020-12-16"), 9, ifelse(interview\_date>as.Date("2021-02-24") & interview\_date<as.Date("2021-04-23"), 11, ifelse(interview\_date>as.Date("2021-06-13") & interview\_date<as.Date("2021-08-25"), 13, ifelse(interview\_date>as.Date("2021-10-11"), 15, 99))))))))  
 #mutate(cattle\_death=colSums(calves\_death,cows\_death,bulls\_death),na.rm=T)  
  
dynamics1a<- dynamics1|>  
 ungroup()|>  
 group\_by(interview\_date, household\_id)|>  
 mutate(cattle\_death=sum(as.numeric(c(calves\_death, cows\_death, bulls\_death)), na.rm=T),  
 sheep\_death=sum(as.numeric(c(sheep\_death, msheep\_death, fsheep\_death)), na.rm=T),  
 goats\_death=sum(as.numeric(c(goats\_death, mgoats\_death, fgoats\_death)), na.rm=T),  
 camels\_death=sum(as.numeric(c(camels\_death, mcamels\_death, fcamels\_death)), na.rm=T))|>  
 ungroup()|>  
 rename(cattle\_birth=cwsbrth, sheep\_birth=shpbrth, goats\_birth=goatsbrth, camels\_birth=cmlsbrth,  
 cattle\_gifts=cowsgft, sheep\_gifts=sheepgfts, goats\_gifts=goatsgft, camels\_gifts=cmlsgft,   
 cattle\_givenout=cowsgvnout, sheep\_givenout=sheepgvnout, goats\_givenout=goatsgvnout,   
 camels\_givenout=cmlsgvnout, cattle\_bought=cttlebght, sheep\_bought=sheepbght,   
 goats\_bought=goatsbgh, camels\_bought=cmlsbght, cattle\_sold=cttlesld, sheep\_sold=sheepsld,   
 goats\_sold=goatsld, camels\_sold=cmlsold)|>  
 dplyr::select(interview\_date, household\_id, cattle\_death, sheep\_death, goats\_death, camels\_death, cattle\_birth, sheep\_birth, goats\_birth, camels\_birth,  
 cattle\_gifts, sheep\_gifts, goats\_gifts, camels\_gifts, cattle\_givenout, sheep\_givenout, goats\_givenout,  
 camels\_givenout, cattle\_bought, sheep\_bought, goats\_bought, camels\_bought, cattle\_sold, sheep\_sold,   
 goats\_sold, camels\_sold)

## Warning: There were 6845 warnings in `mutate()`.  
## The first warning was:  
## ℹ In argument: `cattle\_death = sum(as.numeric(c(calves\_death, cows\_death,  
## bulls\_death)), na.rm = T)`.  
## ℹ In group 1: `interview\_date = 2019-09-04`, `household\_id = "KONMMLHH005"`.  
## Caused by warning:  
## ! NAs introduced by coercion  
## ℹ Run `dplyr::last\_dplyr\_warnings()` to see the 6844 remaining warnings.

dynamics2a<- dynamics2|>  
 ungroup()|>  
 group\_by(interview\_date, household\_id)|>  
 mutate(cattle\_death=sum(as.numeric(c(calves\_death, cows\_death, bulls\_death)), na.rm=T),  
 sheep\_death=sum(as.numeric(c(sheep\_death, msheep\_death, fsheep\_death)), na.rm=T),  
 goats\_death=sum(as.numeric(c(goats\_death, mgoats\_death, fgoats\_death)), na.rm=T),  
 camels\_death=sum(as.numeric(c(camels\_death, mcamels\_death, fcamels\_death)), na.rm=T),  
 cattle\_gifts=sum(as.numeric(c(calves\_gift, cows\_gifts, bulls\_gifts)), na.rm=T),  
 sheep\_gifts=sum(as.numeric(c(sheep\_gifts, msheep\_gifts, fsheep\_gifts)), na.rm=T),  
 goats\_gifts=sum(as.numeric(c(goats\_gifts, mgoats\_gifts, fgoats\_gifts)), na.rm=T),   
 camels\_gifts=sum(as.numeric(c(camels\_gifts, mcamels\_gifts, fcamels\_gifts)), na.rm=T),  
 cattle\_givenout=sum(as.numeric(c(calves\_givenout, bulls\_givenout, cows\_givenout)), na.rm=T),  
 sheep\_givenout=sum(as.numeric(c(sheep\_givenout, msheep\_givenout,fsheep\_givenout)), na.rm=T),  
 goats\_givenout=sum(as.numeric(c(goats\_givenout, mgoats\_givenout, fgoats\_givenout)), na.rm=T),  
 camels\_givenout=sum(as.numeric(c(camels\_givenout,mcamels\_givenout,fcamels\_givenout)), na.rm=T),  
 cattle\_sold=sum(as.numeric(c(calves\_sold, bulls\_sold, cows\_sold)), na.rm=T),  
 sheep\_sold=sum(as.numeric(c(sheep\_sold,msheep\_sold,fsheep\_sold)) , na.rm=T),  
 goats\_sold=sum(as.numeric(c(goats\_sold,mgoats\_sold,fgoats\_sold)), na.rm=T),  
 camels\_sold=sum(as.numeric(c(camels\_sold,mcamels\_sold, fcamels\_sold)), na.rm=T)  
 )|>  
 ungroup()|>  
 rename(cattle\_birth=cwsbrth, sheep\_birth=shpbrth, goats\_birth=goatsbrth, camels\_birth=cmlsbrth,  
 cattle\_bought=cttlebght, sheep\_bought=sheepbght,   
 goats\_bought=goatsbgh, camels\_bought=cmlsbght)|>  
 dplyr::select(interview\_date, household\_id, cattle\_death, sheep\_death, goats\_death, camels\_death, cattle\_birth, sheep\_birth, goats\_birth, camels\_birth,  
 cattle\_gifts, sheep\_gifts, goats\_gifts, camels\_gifts, cattle\_givenout, sheep\_givenout, goats\_givenout,  
 camels\_givenout, cattle\_bought, sheep\_bought, goats\_bought, camels\_bought, cattle\_sold, sheep\_sold,   
 goats\_sold, camels\_sold)

## Warning: There were 116072 warnings in `mutate()`.  
## The first warning was:  
## ℹ In argument: `cattle\_death = sum(as.numeric(c(calves\_death, cows\_death,  
## bulls\_death)), na.rm = T)`.  
## ℹ In group 1: `interview\_date = 2019-10-24`, `household\_id = "KONGLPHH011"`.  
## Caused by warning:  
## ! NAs introduced by coercion  
## ℹ Run `dplyr::last\_dplyr\_warnings()` to see the 116071 remaining warnings.

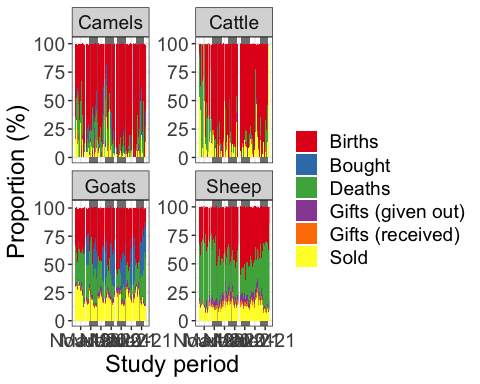
dynamics3a<- dynamics3|>  
 ungroup()|>  
 group\_by(interview\_date, household\_id)|>  
 mutate(cattle\_death=sum(as.numeric(c(calves\_death, cows\_death, bulls\_death)), na.rm=T),  
 sheep\_death=sum(as.numeric(c(sheep\_death, msheep\_death, fsheep\_death)), na.rm=T),  
 goats\_death=sum(as.numeric(c(goats\_death, mgoats\_death, fgoats\_death)), na.rm=T),  
 camels\_death=sum(as.numeric(c(camels\_death, mcamels\_death, fcamels\_death)), na.rm=T),  
 cattle\_gifts=sum(as.numeric(c(calves\_gift, cows\_gifts, bulls\_gifts)), na.rm=T),  
 sheep\_gifts=sum(as.numeric(c(sheep\_gifts, msheep\_gifts, fsheep\_gifts)), na.rm=T),  
 goats\_gifts=sum(as.numeric(c(goats\_gifts, mgoats\_gifts, fgoats\_gifts)), na.rm=T),   
 camels\_gifts=sum(as.numeric(c(camels\_gifts, mcamels\_gifts, fcamels\_gifts)), na.rm=T),  
 cattle\_givenout=sum(as.numeric(c(calves\_givenout, bulls\_givenout, cows\_givenout)), na.rm=T),  
 sheep\_givenout=sum(as.numeric(c(sheep\_givenout, msheep\_givenout,fsheep\_givenout)), na.rm=T),  
 goats\_givenout=sum(as.numeric(c(goats\_givenout, mgoats\_givenout, fgoats\_givenout)), na.rm=T),  
 camels\_givenout=sum(as.numeric(c(camels\_givenout,mcamels\_givenout,fcamels\_givenout)), na.rm=T),  
 cattle\_sold=sum(as.numeric(c(calves\_sold, bulls\_sold, cows\_sold)), na.rm=T),  
 sheep\_sold=sum(as.numeric(c(sheep\_sold,msheep\_sold,fsheep\_sold )), na.rm=T),  
 goats\_sold=sum(as.numeric(c(goats\_sold,mgoats\_sold,fgoats\_sold)), na.rm=T),  
 camels\_sold=sum(as.numeric(c(camels\_sold,mcamels\_sold, fcamels\_sold)), na.rm=T),  
 cattle\_bought=sum(as.numeric(c(calves\_prchsd,bulls\_purchased,cows\_purchased)), na.rm=T),  
 sheep\_bought=sum(as.numeric(c(sheep\_purchased,msheep\_purchased,fsheep\_purchased)), na.rm=T),  
 goats\_bought=sum(as.numeric(c(goats\_sold,mgoats\_sold,fgoats\_sold)), na.rm=T),  
 camels\_bought=sum(as.numeric(c(camels\_sold, mcamels\_sold,fcamels\_sold)), na.rm=T )  
 )|>  
 ungroup()|>  
 rename(cattle\_birth=cwsbrth, sheep\_birth=shpbrth, goats\_birth=goatsbrth, camels\_birth=cmlsbrth  
 )|>  
 dplyr::select(interview\_date, household\_id,cattle\_death, sheep\_death, goats\_death, camels\_death, cattle\_birth, sheep\_birth, goats\_birth, camels\_birth,  
 cattle\_gifts, sheep\_gifts, goats\_gifts, camels\_gifts, cattle\_givenout, sheep\_givenout, goats\_givenout,  
 camels\_givenout, cattle\_bought, sheep\_bought, goats\_bought, camels\_bought, cattle\_sold, sheep\_sold,   
 goats\_sold, camels\_sold)  
  
dynamics\_ <- rbind(dynamics1a,dynamics2a,dynamics3a)|>  
 mutate\_at(vars(cattle\_death, sheep\_death, goats\_death, camels\_death, cattle\_birth, sheep\_birth, goats\_birth, camels\_birth,  
 cattle\_gifts, sheep\_gifts, goats\_gifts, camels\_gifts, cattle\_givenout, sheep\_givenout, goats\_givenout,  
 camels\_givenout, cattle\_bought, sheep\_bought, goats\_bought, camels\_bought, cattle\_sold, sheep\_sold,   
 goats\_sold, camels\_sold), funs(as.numeric(.)))|>  
 pivot\_longer(c(cattle\_death, sheep\_death, goats\_death, camels\_death, cattle\_birth, sheep\_birth, goats\_birth, camels\_birth,  
 cattle\_gifts, sheep\_gifts, goats\_gifts, camels\_gifts, cattle\_givenout, sheep\_givenout, goats\_givenout,  
 camels\_givenout, cattle\_bought, sheep\_bought, goats\_bought, camels\_bought, cattle\_sold, sheep\_sold,   
 goats\_sold, camels\_sold), names\_to="movement", values\_to="movement1")|>  
 mutate(dyn=ifelse(grepl("death", movement), "Deaths", ifelse(grepl("birth", movement), "Births", ifelse(grepl("gifts", movement), "Gifts (received)", ifelse(grepl("givenout", movement), "Gifts (given out)", ifelse(grepl("bought", movement), "Bought", ifelse(grepl("sold", movement), "Sold", movement)))))))|>  
 mutate(interview\_week=floor\_date(interview\_date, "week"))|>  
 group\_by(interview\_week, movement)|>  
 mutate(movement1=sum(movement1, na.rm=T))|>  
 ungroup()|>  
 dplyr::select(-household\_id, -interview\_date)|>  
 distinct()|>  
 mutate(movement=ifelse(grepl("cattle", movement), "Cattle", ifelse(grepl("sheep", movement), "Sheep", ifelse(grepl("goats", movement), "Goats", ifelse(grepl("camel", movement), "Camels", movement)))))

## Warning: `funs()` was deprecated in dplyr 0.8.0.  
## ℹ Please use a list of either functions or lambdas:  
##   
## # Simple named list: list(mean = mean, median = median)  
##   
## # Auto named with `tibble::lst()`: tibble::lst(mean, median)  
##   
## # Using lambdas list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

## Warning: There were 20 warnings in `mutate()`.  
## The first warning was:  
## ℹ In argument: `cattle\_birth = as.numeric(cattle\_birth)`.  
## Caused by warning:  
## ! NAs introduced by coercion  
## ℹ Run `dplyr::last\_dplyr\_warnings()` to see the 19 remaining warnings.

dynamics\_<- dynamics\_|>  
 mutate(movement1=ifelse(movement%in%"Cattle" & movement1==124, 12, movement1))|>  
 group\_by(interview\_week, movement)|>  
 mutate(tot=sum(movement1, na.rm=T))|>  
 ungroup()|>  
 mutate(prop=round(movement1/tot\*100))  
  
ggplot(dynamics\_, aes(x=interview\_week, y=prop, group=dyn, fill=dyn))+geom\_rect(aes(xmin=as.Date("2020-08-18"), xmax=as.Date("2020-11-30"), ymin=-Inf, ymax=Inf ), fill="grey50", color=NA)+geom\_rect(aes(xmin=as.Date("2021-01-13"), xmax=as.Date("2021-04-30"), ymin=-Inf, ymax=Inf ),fill="grey50", color=NA)+geom\_rect(aes(xmin=as.Date("2020-02-08"), xmax=as.Date("2020-05-28"), ymin=-Inf, ymax=Inf),fill="grey50", color=NA)+geom\_rect(aes(xmin=as.Date("2021-08-30"), xmax=as.Date("2021-12-01"), ymin=-Inf, ymax=Inf ),fill="grey50", color=NA)+geom\_col()+facet\_wrap(.~movement, scales = "free\_y")+theme\_bw()+scale\_x\_date(date\_breaks="4 months", date\_labels = "%b-%y")+theme(text=element\_text(size=18))+scale\_fill\_brewer(palette="Set1")+labs(x="Study period", y="Proportion (%)", fill="")

## Warning: Removed 24 rows containing missing values (`position\_stack()`).



#ggsave("herd\_dynamics2a.png", width = 18, height=10)  
  
# lvstckown2 <- dynamics|>full\_join(lvstckown1|>dplyr::select(-interview\_date),by=c("household\_id"))|>distinct()|>  
# gather(c(cattle\_out:camel\_in), key="movement", value="animals\_moved")|>  
# mutate(dyn=ifelse(grepl("in", movement), "In", "Out"))|>  
# mutate(movement=ifelse(grepl("camel", movement), "Camels", ifelse(grepl("cattle", movement), "Cattle", ifelse(grepl("goat", movement), "Goats",ifelse(grepl("sheep", movement), "Sheep", movement)))))|>  
# mutate(interview\_week=floor\_date(interview\_date, "week"))|>  
# #mutate(visit=1)|>  
# group\_by(interview\_week, study\_arm, movement, dyn)|>  
# mutate(movement1=sum(animals\_moved))|>  
# ungroup()|>  
# dplyr::select(interview\_week, study\_arm, movement1, movement, dyn)|>  
# distinct()  
#   
# ggplot(lvstckown2, aes(x=interview\_week, y=movement1, group=dyn, fill=dyn))+geom\_col()+facet\_wrap(.~movement, scales = "free\_y")+theme\_bw()+labs(x="Period", y="Count", fill="")+scale\_x\_date(date\_breaks="4 months", date\_labels = "%b-%y")+theme(text=element\_text(size=16))  
  
#ggsave("herd\_dynamics.png", width = 18, height=10)  
  
  
  
lvstckown3 <- dynamics|>  
 full\_join(lvstckown1|>dplyr::select(-interview\_date),by=c("household\_id"))|>distinct()|>  
 gather(c(cattle\_out:camel\_in), key="movement", value="animals\_moved")|>  
 mutate(species\_moved=ifelse(movement%in%"camel\_in" | movement%in%"camel\_out","camels",ifelse(movement%in%"cattle\_in" | movement%in%"cattle\_out","cattle",ifelse(movement%in%"goat\_in" | movement%in%"goat\_out","goats","sheep"))))|>  
 mutate(interview\_week=floor\_date(interview\_date, "week"))|>  
 group\_by(visit, household\_id, movement, species\_moved)|>  
 mutate(movement1=sum(animals\_moved))|>  
 ungroup()|>  
 dplyr::select(visit, household\_id, animals\_owned,species\_owned,movement1, movement, species\_moved,)|>  
 distinct()|>  
 mutate(movement2=ifelse(grepl("out",movement), "Out", "In"))|>  
 mutate(species\_moved=str\_to\_title(species\_moved))|>dplyr::select(-movement)

## Warning in full\_join(dynamics, dplyr::select(lvstckown1, -interview\_date), : Detected an unexpected many-to-many relationship between `x` and `y`.  
## ℹ Row 1 of `x` matches multiple rows in `y`.  
## ℹ Row 5709 of `y` matches multiple rows in `x`.  
## ℹ If a many-to-many relationship is expected, set `relationship =  
## "many-to-many"` to silence this warning.

lvstckown4 <- lvstckown3|>  
 group\_by(visit, household\_id, species\_owned, species\_moved)|>  
 mutate(animals\_owned=ifelse(movement2%in%"In",(animals\_owned+movement1),animals\_owned))|>  
 ungroup()|>  
 group\_by(visit,household\_id,species\_owned,species\_moved)|>  
 mutate(animals\_owned=ifelse(movement2%in%"Out",(animals\_owned-movement1),animals\_owned))|>  
 ungroup()  
   
  
dynamics\_1<- dynamics|>  
 #dplyr::select(-interview\_date)|>  
 #filter(days\_recruitment>)|>  
 distinct()  
   
   
lvstckown1a <- household\_baseline|>  
 clean\_names()|>  
 # filter(hh\_eligible%in%"Yes")|>  
 dplyr::select(interview\_date,household\_id,calvsnum,bullsnum,cowsnum,sheep1num,shpmlenum,shpfmenum,goatsnum,gtsmlenum,gtsfmlenum,cmels1num,cmelsmlenum,cmelsfmlenum,study\_arm)|>  
 gather(c(calvsnum,bullsnum,cowsnum,sheep1num,shpmlenum,shpfmenum,goatsnum,gtsmlenum,gtsfmlenum,cmels1num,cmelsmlenum,cmelsfmlenum), key="species\_owned", value="animals\_owned")|>  
 #mutate(interview\_week=floor\_date(interview\_date,"week"))|>  
 #filter(!is.na(animals\_owned))|>  
 mutate(species\_owned=recode(species\_owned,"calvsnum"="cattle","bullsnum"="cattle","cowsnum"="cattle","sheep1num"="sheep","shpmlenum"="sheep","shpfmenum"="sheep","goatsnum"="goats","gtsmlenum"="goats","gtsfmlenum"="goats","cmels1num"="camels","cmelsmlenum"="camels","cmelsfmlenum"="camels"))|>  
 group\_by(household\_id, species\_owned)|>  
 mutate(animals\_owned=sum(as.numeric(animals\_owned), na.rm=T))|>  
 ungroup()|>  
 distinct()|>  
 pivot\_wider(names\_from=species\_owned, values\_from=animals\_owned)|>  
 mutate\_at(vars(cattle, sheep, goats, camels), funs(ifelse(is.na(.), 0, .)))|>  
 dplyr::select(-interview\_date)

## Warning: There were 3260 warnings in `mutate()`.  
## The first warning was:  
## ℹ In argument: `animals\_owned = sum(as.numeric(animals\_owned), na.rm = T)`.  
## ℹ In group 2: `household\_id = "KOHFGAHH001"`, `species\_owned = "cattle"`.  
## Caused by warning:  
## ! NAs introduced by coercion  
## ℹ Run `dplyr::last\_dplyr\_warnings()` to see the 3259 remaining warnings.

## Warning: `funs()` was deprecated in dplyr 0.8.0.  
## ℹ Please use a list of either functions or lambdas:  
##   
## # Simple named list: list(mean = mean, median = median)  
##   
## # Auto named with `tibble::lst()`: tibble::lst(mean, median)  
##   
## # Using lambdas list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

lvstckown3a <- dynamics\_1|>  
 mutate\_at(vars(cattle\_in, cattle\_out, sheep\_out, sheep\_in, camel\_out, camel\_in, goat\_out, goat\_in), funs(ifelse(days\_recruitment==0, 0, .)))|>  
 full\_join(lvstckown1a,by=c("household\_id"))|>distinct()|>  
 arrange(interview\_date)|>  
 mutate\_at(vars(cattle\_out, cattle\_in,cattle, sheep\_out,sheep\_in,sheep, goat\_in,goat\_out,goats,camels, camel\_in, camel\_out), funs(ifelse(is.na(.), 0, .)))|>  
 #mutate(visit=ifelse(is.na(visit), 1, visit))|>  
 group\_by( household\_id)|>  
 arrange(interview\_date)|>  
 # mutate\_at(vars(cattle, sheep, goats, camels), funs(ifelse(visit==2, dplyr::lag(.), .)))|>  
 mutate(cattle=abs(cattle-cattle\_out+cattle\_in))|>  
 mutate(sheep=abs(sheep-sheep\_out+sheep\_in))|>  
 mutate(goats=abs(goats-goat\_out+goat\_in))|>  
 mutate(camels=abs(camels-camel\_out+camel\_in))|>  
 ungroup()|>  
 # group\_by( household\_id)|>  
 # arrange(interview\_date)|>  
 # mutate\_at(vars(cattle, sheep, goats, camels), funs(ifelse(visit==3, dplyr::lag(.), .)))|>  
 # mutate(cattle=ifelse(visit==3, cattle-cattle\_out+cattle\_in, cattle))|>  
 # mutate(sheep=ifelse(visit==3, sheep-sheep\_out+sheep\_in, sheep))|>  
 # mutate(goats=ifelse(visit==3, goats-goat\_out+goat\_in, goats))|>  
 # mutate(camels=ifelse(visit==3, camels-camel\_out+camel\_in, camels))|>  
 # ungroup()|>  
 # group\_by( household\_id)|>  
 # arrange(interview\_date)|>  
 # mutate\_at(vars(cattle, sheep, goats, camels), funs(ifelse(visit==4, dplyr::lag(.), .)))|>  
 # mutate(cattle=ifelse(visit==4, cattle-cattle\_out+cattle\_in, cattle))|>  
 # mutate(sheep=ifelse(visit==4, sheep-sheep\_out+sheep\_in, sheep))|>  
 # mutate(goats=ifelse(visit==4, goats-goat\_out+goat\_in, goats))|>  
 # mutate(camels=ifelse(visit==4, camels-camel\_out+camel\_in, camels))|>  
 # ungroup()|>  
 # group\_by( household\_id)|>  
 # arrange(interview\_date)|>  
 # mutate\_at(vars(cattle, sheep, goats, camels), funs(ifelse(visit==5, dplyr::lag(.), .)))|>  
 # mutate(cattle=ifelse(visit==5, cattle-cattle\_out+cattle\_in, cattle))|>  
 # mutate(sheep=ifelse(visit==5, sheep-sheep\_out+sheep\_in, sheep))|>  
 # mutate(goats=ifelse(visit==5, goats-goat\_out+goat\_in, goats))|>  
 # mutate(camels=ifelse(visit==5, camels-camel\_out+camel\_in, camels))|>  
 # ungroup()|>  
 # group\_by( household\_id)|>  
 # arrange(interview\_date)|>  
 # mutate\_at(vars(cattle, sheep, goats, camels), funs(ifelse(visit==6, dplyr::lag(.), .)))|>  
 # mutate(cattle=ifelse(visit==6, cattle-cattle\_out+cattle\_in, cattle))|>  
 # mutate(sheep=ifelse(visit==6, sheep-sheep\_out+sheep\_in, sheep))|>  
 # mutate(goats=ifelse(visit==6, goats-goat\_out+goat\_in, goats))|>  
 # mutate(camels=ifelse(visit==6, camels-camel\_out+camel\_in, camels))|>  
 # ungroup()|>  
 # group\_by( household\_id)|>  
 # arrange(interview\_date)|>  
 # mutate\_at(vars(cattle, sheep, goats, camels), funs(ifelse(visit==7, dplyr::lag(.), .)))|>  
 # mutate(cattle=ifelse(visit==7, cattle-cattle\_out+cattle\_in, cattle))|>  
 # mutate(sheep=ifelse(visit==7, sheep-sheep\_out+sheep\_in, sheep))|>  
 # mutate(goats=ifelse(visit==7, goats-goat\_out+goat\_in, goats))|>  
 # mutate(camels=ifelse(visit==7, camels-camel\_out+camel\_in, camels))|>  
 # ungroup()|>  
 # group\_by( household\_id)|>  
 # arrange(interview\_date)|>  
 # mutate\_at(vars(cattle, sheep, goats, camels), funs(ifelse(visit==8, dplyr::lag(.), .)))|>  
 # mutate(cattle=ifelse(visit==8, cattle-cattle\_out+cattle\_in, cattle))|>  
 # mutate(sheep=ifelse(visit==8, sheep-sheep\_out+sheep\_in, sheep))|>  
 # mutate(goats=ifelse(visit==8, goats-goat\_out+goat\_in, goats))|>  
 # mutate(camels=ifelse(visit==8, camels-camel\_out+camel\_in, camels))|>  
 # ungroup()|>  
 # group\_by( household\_id)|>  
 # arrange(interview\_date)|>  
 # mutate\_at(vars(cattle, sheep, goats, camels), funs(ifelse(visit==9, dplyr::lag(.), .)))|>  
 # mutate(cattle=ifelse(visit==9, cattle-cattle\_out+cattle\_in, cattle))|>  
 # mutate(sheep=ifelse(visit==9, sheep-sheep\_out+sheep\_in, sheep))|>  
 # mutate(goats=ifelse(visit==9, goats-goat\_out+goat\_in, goats))|>  
 # mutate(camels=ifelse(visit==9, camels-camel\_out+camel\_in, camels))|>  
 # ungroup()|>  
 # group\_by( household\_id)|>  
 # arrange(interview\_date)|>  
 # mutate\_at(vars(cattle, sheep, goats, camels), funs(ifelse(visit==10, dplyr::lag(.), .)))|>  
 # mutate(cattle=ifelse(visit==10, cattle-cattle\_out+cattle\_in, cattle))|>  
 # mutate(sheep=ifelse(visit==10, sheep-sheep\_out+sheep\_in, sheep))|>  
 # mutate(goats=ifelse(visit==10, goats-goat\_out+goat\_in, goats))|>  
 # mutate(camels=ifelse(visit==10, camels-camel\_out+camel\_in, camels))|>  
 # ungroup()|>  
 # group\_by( household\_id)|>  
 # arrange(interview\_date)|>  
 # mutate\_at(vars(cattle, sheep, goats, camels), funs(ifelse(visit==11, dplyr::lag(.), .)))|>  
 # mutate(cattle=ifelse(visit==11, cattle-cattle\_out+cattle\_in, cattle))|>  
 # mutate(sheep=ifelse(visit==11, sheep-sheep\_out+sheep\_in, sheep))|>  
 # mutate(goats=ifelse(visit==11, goats-goat\_out+goat\_in, goats))|>  
 # mutate(camels=ifelse(visit==11, camels-camel\_out+camel\_in, camels))|>  
 # ungroup()|>  
 # group\_by( household\_id)|>  
 # arrange(interview\_date)|>  
 # mutate\_at(vars(cattle, sheep, goats, camels), funs(ifelse(visit==12, dplyr::lag(.), .)))|>  
 # mutate(cattle=ifelse(visit==12, cattle-cattle\_out+cattle\_in, cattle))|>  
 # mutate(sheep=ifelse(visit==12, sheep-sheep\_out+sheep\_in, sheep))|>  
 # mutate(goats=ifelse(visit==12, goats-goat\_out+goat\_in, goats))|>  
 # mutate(camels=ifelse(visit==12, camels-camel\_out+camel\_in, camels))|>  
 # ungroup()|>  
 # group\_by( household\_id)|>  
 # arrange(interview\_date)|>  
 # mutate\_at(vars(cattle, sheep, goats, camels), funs(ifelse(visit==13, dplyr::lag(.), .)))|>  
 # mutate(cattle=ifelse(visit==13, cattle-cattle\_out+cattle\_in, cattle))|>  
 # mutate(sheep=ifelse(visit==13, sheep-sheep\_out+sheep\_in, sheep))|>  
 # mutate(goats=ifelse(visit==13, goats-goat\_out+goat\_in, goats))|>  
 # mutate(camels=ifelse(visit==13, camels-camel\_out+camel\_in, camels))|>  
 # ungroup()|>  
 # group\_by( household\_id)|>  
 # arrange(interview\_date)|>  
 # mutate\_at(vars(cattle, sheep, goats, camels), funs(ifelse(visit==14, dplyr::lag(.), .)))|>  
 # mutate(cattle=ifelse(visit==14, cattle-cattle\_out+cattle\_in, cattle))|>  
 # mutate(sheep=ifelse(visit==14, sheep-sheep\_out+sheep\_in, sheep))|>  
 # mutate(goats=ifelse(visit==14, goats-goat\_out+goat\_in, goats))|>  
 # mutate(camels=ifelse(visit==14, camels-camel\_out+camel\_in, camels))|>  
 # ungroup()|>  
 # group\_by( household\_id)|>  
 # arrange(interview\_date)|>  
 # mutate\_at(vars(cattle, sheep, goats, camels), funs(ifelse(visit==15, dplyr::lag(.), .)))|>  
 # mutate(cattle=ifelse(visit==15, cattle-cattle\_out+cattle\_in, cattle))|>  
 # mutate(sheep=ifelse(visit==15, sheep-sheep\_out+sheep\_in, sheep))|>  
 # mutate(goats=ifelse(visit==15, goats-goat\_out+goat\_in, goats))|>  
 # mutate(camels=ifelse(visit==15, camels-camel\_out+camel\_in, camels))|>  
 # ungroup()|>  
 dplyr::select( interview\_date, household\_id, days\_recruitment, study\_arm, cattle, sheep, goats, camels)|>  
 filter(!is.na(cattle))|>  
 distinct()|>  
 mutate(days\_recruitment=ifelse(is.na(days\_recruitment), 0, days\_recruitment))

## Warning: `funs()` was deprecated in dplyr 0.8.0.  
## ℹ Please use a list of either functions or lambdas:  
##   
## # Simple named list: list(mean = mean, median = median)  
##   
## # Auto named with `tibble::lst()`: tibble::lst(mean, median)  
##   
## # Using lambdas list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

## Warning: `funs()` was deprecated in dplyr 0.8.0.  
## ℹ Please use a list of either functions or lambdas:  
##   
## # Simple named list: list(mean = mean, median = median)  
##   
## # Auto named with `tibble::lst()`: tibble::lst(mean, median)  
##   
## # Using lambdas list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

lvstckown4 <- lvstckown3|>  
 group\_by(visit, household\_id, species\_owned, species\_moved)|>  
 mutate(animals\_owned=ifelse(movement2%in%"In",(animals\_owned+movement1),animals\_owned))|>  
 ungroup()|>  
 group\_by(visit,household\_id,species\_owned,species\_moved)|>  
 mutate(animals\_owned=ifelse(movement2%in%"Out",(animals\_owned-movement1),animals\_owned))|>  
 ungroup()  
   
  
########  
### births  
#####  
births1 <- household\_baseline|>  
 clean\_names()|>  
 dplyr::select(interview\_date,household\_id,cwsbrth,shpbrth,goatsbrth,cmlsbrth) |>  
 mutate\_at(vars(cwsbrth,shpbrth,goatsbrth,cmlsbrth), funs(as.numeric(.)))

## Warning: `funs()` was deprecated in dplyr 0.8.0.  
## ℹ Please use a list of either functions or lambdas:  
##   
## # Simple named list: list(mean = mean, median = median)  
##   
## # Auto named with `tibble::lst()`: tibble::lst(mean, median)  
##   
## # Using lambdas list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

## Warning: There were 4 warnings in `mutate()`.  
## The first warning was:  
## ℹ In argument: `cwsbrth = as.numeric(cwsbrth)`.  
## Caused by warning:  
## ! NAs introduced by coercion  
## ℹ Run `dplyr::last\_dplyr\_warnings()` to see the 3 remaining warnings.

#mutate(visit=1)  
 #mutate(cattle\_death=colSums(calves\_death,cows\_death,bulls\_death),na.rm=T)  
births1$no\_births <- rowSums(births1[ , c("cwsbrth","shpbrth","goatsbrth","cmlsbrth")],na.rm = T)  
  
births2 <- followup\_data|>  
 clean\_names()|>  
 dplyr::select(interview\_date,household\_id,cwsbrth,shpbrth,goatsbrth,cmlsbrth)|>  
 mutate\_at(vars(cwsbrth,shpbrth,goatsbrth,cmlsbrth), funs(as.numeric(.)))

## Warning: `funs()` was deprecated in dplyr 0.8.0.  
## ℹ Please use a list of either functions or lambdas:  
##   
## # Simple named list: list(mean = mean, median = median)  
##   
## # Auto named with `tibble::lst()`: tibble::lst(mean, median)  
##   
## # Using lambdas list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))  
## There were 4 warnings in `mutate()`. The first warning was: ℹ In argument:  
## `cwsbrth = as.numeric(cwsbrth)`. Caused by warning: ! NAs introduced by  
## coercion ℹ Run `dplyr::last\_dplyr\_warnings()` to see the 3 remaining warnings.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

births2$no\_births <- rowSums(births2[ , c("cwsbrth","shpbrth","goatsbrth","cmlsbrth")],na.rm = T)  
  
births3 <- household\_quarterly|>  
 clean\_names()|>  
 dplyr::select(interview\_date,household\_id,cwsbrth,shpbrth,goatsbrth,cmlsbrth)|>  
 mutate\_at(vars(cwsbrth,shpbrth,goatsbrth,cmlsbrth), funs(as.numeric(.)))

## Warning: `funs()` was deprecated in dplyr 0.8.0.  
## ℹ Please use a list of either functions or lambdas:  
##   
## # Simple named list: list(mean = mean, median = median)  
##   
## # Auto named with `tibble::lst()`: tibble::lst(mean, median)  
##   
## # Using lambdas list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

births3$no\_births <- rowSums(births3[ , c("cwsbrth","shpbrth","goatsbrth","cmlsbrth")],na.rm = T)  
  
births <- rbind(births1,births2,births3)|>  
 group\_by(household\_id)|>  
 arrange(interview\_date)|>  
 mutate(days\_recruitment=interview\_date-dplyr::lag(interview\_date))|>  
 ungroup()|>  
 group\_by(household\_id, days\_recruitment)|>  
 mutate\_at(vars(no\_births), funs(sum(., na.rm=T)))|>  
 ungroup()|>  
 distinct()

## Warning: `funs()` was deprecated in dplyr 0.8.0.  
## ℹ Please use a list of either functions or lambdas:  
##   
## # Simple named list: list(mean = mean, median = median)  
##   
## # Auto named with `tibble::lst()`: tibble::lst(mean, median)  
##   
## # Using lambdas list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

#write\_csv(births, "l4h\_births.csv")  
  
##disease event frequency  
animal\_sick1 <- household\_baseline|>  
 clean\_names()|>  
 # filter(hh\_eligible%in%"Yes")|>  
 dplyr::select(interview\_date,household\_id,anmlsick)|>  
 filter(!is.na(anmlsick))|>  
 # mutate(visit=1)|>  
 dplyr::select(interview\_date, household\_id, anmlsick)  
animal\_sick2 <- household\_quarterly|>  
 dplyr::select(interview\_date,household\_id,anmlsick)|>  
 filter(!is.na(anmlsick))|>  
 # mutate(visit=ifelse(interview\_date<as.Date("2020-02-15"), 3, ifelse(interview\_date>as.Date("2020-04-05") & interview\_date<as.Date("2020-05-20"), 5, ifelse(interview\_date>as.Date("2020-05-20") & interview\_date<as.Date("2020-09-12"), 7, ifelse(interview\_date>as.Date("2020-09-13") & interview\_date<as.Date("2020-12-16"), 9, ifelse(interview\_date>as.Date("2021-02-24") & interview\_date<as.Date("2021-04-23"), 11, ifelse(interview\_date>as.Date("2021-06-13") & interview\_date<as.Date("2021-08-25"), 13, ifelse(interview\_date>as.Date("2021-10-11"), 15, 99))))))))|>  
 # dplyr::select(-interview\_date)|>  
 distinct()  
  
  
animal\_sick3 <- followup\_data|>  
 #filter(hh\_eligible%in%"Yes")|>  
 rename(anmlsick=form\_anmlsick) |>  
 dplyr::select(interview\_date,household\_id,anmlsick)|>  
 filter(!is.na(anmlsick))|>  
#mutate(visit=ifelse(interview\_date<as.Date("2020-01-16"), 2, ifelse(interview\_date>as.Date("2020-01-16") & interview\_date<as.Date("2020-04-03"), 4, ifelse(interview\_date>as.Date("2020-05-01") & interview\_date<as.Date("2020-07-28"), 6, ifelse(interview\_date>as.Date("2020-09-10") & interview\_date<as.Date("2020-10-28"), 8, ifelse(interview\_date>as.Date("2021-01-05") & interview\_date<as.Date("2021-02-25"), 10, ifelse(interview\_date>as.Date("2021-02-25") & interview\_date<as.Date("2021-07-07"), 12, ifelse(interview\_date>as.Date("2021-08-17"), 14, 99))))))))|>  
 #dplyr::select(-interview\_date)|>  
 distinct()  
animal\_sick <- animal\_sick1|>  
 rbind(animal\_sick2, animal\_sick3)|>  
 group\_by(household\_id)|>  
 arrange(interview\_date)|>  
 mutate(days\_recruitment=interview\_date-dplyr::lag(interview\_date))|>  
 ungroup()|>  
 mutate(anmlsick=ifelse(anmlsick%in%"Yes", 1, 0))|>  
 group\_by(household\_id, days\_recruitment)|>  
 mutate(anmlsick=max(anmlsick, na.rm=T))|>  
 ungroup()|>  
 distinct()|>  
 mutate(anmlsick=recode(anmlsick, "1"="Yes", "0"="No"))  
 #mutate(interview\_week=floor\_date(interview\_date,"week"))|>  
 #dplyr::select(-interview\_date)  
  
##number animals sick  
sick\_animals1 <- household\_baseline|>  
 clean\_names()|>  
 dplyr::select(interview\_date,household\_id,sickcttle,sickgts,sickshp,sickcmls)  
  
sick\_animals2 <- household\_quarterly|>  
 clean\_names()|>  
 dplyr::select(interview\_date,household\_id,sickcttle,sickgts,sickshp,sickcmls)  
  
names(followup\_data)<- gsub("form\_", "",names(followup\_data))  
sick\_animals3 <- followup\_data|>  
 clean\_names()|>  
 dplyr::select(interview\_date,household\_id,sickcttle,sickgts,sickshp,sickcmls)  
  
sick\_animals <- sick\_animals1|>  
 rbind(sick\_animals2, sick\_animals3)|>  
 group\_by(household\_id)|>  
 arrange(interview\_date)|>  
 mutate(days\_recruitment=interview\_date-dplyr::lag(interview\_date))|>  
 mutate(days\_recruitment=ifelse(is.na(days\_recruitment), 0, days\_recruitment))|>  
 ungroup()|>  
 gather(c(sickcttle,sickgts,sickshp,sickcmls), key="sick\_species", value="sick\_animals")|>  
 filter(!is.na(sick\_animals))|>  
 mutate(sick\_species=recode(sick\_species,"sickcttle"="cattle","sickshp"="sheep","sickgts"="goats","sickcmls"="camels"))|>  
 mutate(sick\_animals=as.numeric(sick\_animals)) |>  
 mutate(tlu=ifelse(sick\_species%in%"cattle",(sick\_animals\*1),ifelse(sick\_species%in%"camels",(sick\_animals\*1),(sick\_animals\*0.1))))|>  
 group\_by(days\_recruitment,household\_id,sick\_species)|>  
 mutate(sick\_animals=round(sum(sick\_animals)))|>  
 mutate(tlu=sum(tlu))|>  
 ungroup()|>  
 group\_by(days\_recruitment,household\_id)|>  
 mutate(total\_sick\_animals=round(sum(sick\_animals)))|>  
 mutate(total\_tlu=sum(tlu))|>  
 ungroup()|>  
 dplyr::select(household\_id,total\_tlu, days\_recruitment)|>  
 distinct()

## Warning: There was 1 warning in `mutate()`.  
## ℹ In argument: `sick\_animals = as.numeric(sick\_animals)`.  
## Caused by warning:  
## ! NAs introduced by coercion

sick\_animals2 <- household\_quarterly|>  
 clean\_names()|>  
 dplyr::select(interview\_date,household\_id,sickcttle,sickgts,sickshp,sickcmls)|>  
 gather(c(sickcttle,sickgts,sickshp,sickcmls), key="sick\_species", value="sick\_animals")|>  
 mutate(interview\_week=floor\_date(interview\_date,"week"))|>  
 mutate(visit=ifelse(interview\_date<as.Date("2020-02-15"), 3, ifelse(interview\_date>as.Date("2020-04-05") & interview\_date<as.Date("2020-05-20"), 5, ifelse(interview\_date>as.Date("2020-05-20") & interview\_date<as.Date("2020-09-12"), 7, ifelse(interview\_date>as.Date("2020-09-13") & interview\_date<as.Date("2020-12-16"), 9, ifelse(interview\_date>as.Date("2021-02-24") & interview\_date<as.Date("2021-04-23"), 11, ifelse(interview\_date>as.Date("2021-06-13") & interview\_date<as.Date("2021-08-25"), 13, ifelse(interview\_date>as.Date("2021-10-11"), 15, 99))))))))|>  
 filter(!is.na(sick\_animals))|>  
 mutate(sick\_species=recode(sick\_species,"sickcttle"="cattle","sickshp"="sheep","sickgts"="goats","sickcmls"="camels"))|>  
 mutate(tlu=ifelse(sick\_species%in%"cattle",(sick\_animals\*1),ifelse(sick\_species%in%"camels",(sick\_animals\*1),(sick\_animals\*0.1))))|>  
 group\_by(visit,household\_id,sick\_species)|>  
 mutate(sick\_animals=round(sum(sick\_animals)))|>  
 mutate(tlu=sum(tlu))|>  
 ungroup()|>  
 group\_by(visit,household\_id)|>  
 mutate(total\_sick\_animals=round(sum(sick\_animals)))|>  
 mutate(total\_sicktlu=sum(tlu))|>  
 ungroup()|>  
 dplyr::select(household\_id,total\_sicktlu, visit)|>  
 distinct()  
  
# sick\_animals3 <- individual\_followup|>  
# clean\_names()|>  
# filter(hh\_eligible%in%"Yes")|>  
# dplyr::select(interview\_date,household\_id,sickcttle,sickgts,sickshp,sickcmls)|>  
# gather(c(sickcttle,sickgts,sickshp,sickcmls), key="sick\_species", value="sick\_animals")|>  
# mutate(interview\_week=floor\_date(interview\_date,"week"))|>  
# mutate(visit=ifelse(interview\_date<as.Date("2020-01-16"), 2, ifelse(interview\_date>as.Date("2020-01-16") & interview\_date<as.Date("2020-04-03"), 4, ifelse(interview\_date>as.Date("2020-05-01") & interview\_date<as.Date("2020-07-28"), 6, ifelse(interview\_date>as.Date("2020-09-10") & interview\_date<as.Date("2020-10-28"), 8, ifelse(interview\_date>as.Date("2021-01-05") & interview\_date<as.Date("2021-02-25"), 10, ifelse(interview\_date>as.Date("2021-02-25") & interview\_date<as.Date("2021-07-07"), 12, ifelse(interview\_date>as.Date("2021-08-17"), 14, 99))))))))|>  
# filter(!is.na(sick\_animals))|>  
# mutate(sick\_species=recode(sick\_species,"sickcttle"="cattle","sickshp"="sheep","sickgts"="goats","sickcmls"="camels"))|>  
# mutate(tlu=ifelse(sick\_species%in%"cattle",(sick\_animals\*1),ifelse(sick\_species%in%"camels",(sick\_animals\*1),(sick\_animals\*0.1))))|>  
# group\_by(visit,household\_id,sick\_species)|>  
# mutate(sick\_animals=round(sum(sick\_animals)))|>  
# mutate(tlu=sum(tlu))|>  
# ungroup()|>  
# group\_by(visit,household\_id)|>  
# mutate(total\_sick\_animals=round(sum(sick\_animals)))|>  
# mutate(total\_tlu=sum(tlu))|>  
# ungroup()|>  
# dplyr::select(household\_id,total\_tlu, visit)|>  
# distinct()  
# sick\_animals <- sick\_animals1|>  
# rbind(sick\_animals2, sick\_animals3)|>  
# rename(sick\_tlu=total\_tlu)|>  
# distinct()  
  
##animal abortions  
abortions1 <- household\_baseline|>  
 clean\_names()|>  
 dplyr::select(interview\_date,household\_id,abortnscttle,abrtngts,abrtnshp,abrtncmls)  
  
abortions2 <- household\_quarterly|>  
 clean\_names()|>  
 dplyr::select(interview\_date,household\_id,abortnscttle,abrtngts,abrtnshp,abrtncmls)  
   
abortions3 <- followup\_data|>  
 clean\_names()|>  
 dplyr::select(interview\_date,household\_id,abortnscttle,abrtngts,abrtnshp,abrtncmls)  
  
abortions <- abortions1|>  
 rbind(abortions2, abortions3)|>  
 group\_by(household\_id)|>  
 arrange(interview\_date)|>  
 mutate(days\_recruitment=interview\_date-dplyr::lag(interview\_date))|>  
 mutate(days\_recruitment=ifelse(is.na(days\_recruitment), 0, days\_recruitment))|>  
 ungroup()|>  
 gather(c(abortnscttle,abrtngts,abrtnshp,abrtncmls), key="species\_aborted", value="number\_aborted")|>  
 mutate(interview\_week=floor\_date(interview\_date,"week"))|>  
 filter(!is.na(number\_aborted))|>  
 mutate(species\_aborted=recode(species\_aborted,"abortnscttle"="cattle","abrtnshp"="sheep","abrtngts"="goats","abrtncmls"="camels"))|>  
 mutate(number\_aborted=as.numeric(number\_aborted)) |>  
 mutate(tlu=ifelse(species\_aborted%in%"cattle",(number\_aborted\*1),ifelse(species\_aborted%in%"camels",(number\_aborted\*1),(number\_aborted\*0.1))))|>  
 group\_by(days\_recruitment,household\_id,species\_aborted)|>  
 mutate(number\_aborted=round(sum(number\_aborted)))|>  
 mutate(tlu=sum(tlu))|>  
 ungroup()|>  
 group\_by(days\_recruitment,household\_id)|>  
 mutate(total\_number\_aborted=round(sum(number\_aborted)))|>  
 mutate(aborted\_tlu=sum(tlu))|>  
 ungroup()|>  
 dplyr::select(days\_recruitment, household\_id,aborted\_tlu)|>  
 distinct()

## Warning: There was 1 warning in `mutate()`.  
## ℹ In argument: `number\_aborted = as.numeric(number\_aborted)`.  
## Caused by warning:  
## ! NAs introduced by coercion

##vet intervention frequency  
vet\_freq1 <- household\_baseline|>  
 clean\_names()|>  
 dplyr::select(interview\_date,household\_id,trtaniml)|>  
 filter(!is.na(trtaniml))  
  
vet\_freq2 <- household\_quarterly|>  
 clean\_names()|>  
 dplyr::select(interview\_date,household\_id,trtaniml)|>  
 filter(!is.na(trtaniml))#|>  
 # mutate(visit=ifelse(interview\_date<as.Date("2020-02-15"), 3, ifelse(interview\_date>as.Date("2020-04-05") & interview\_date<as.Date("2020-05-20"), 5, ifelse(interview\_date>as.Date("2020-05-20") & interview\_date<as.Date("2020-09-12"), 7, ifelse(interview\_date>as.Date("2020-09-13") & interview\_date<as.Date("2020-12-16"), 9, ifelse(interview\_date>as.Date("2021-02-24") & interview\_date<as.Date("2021-04-23"), 11, ifelse(interview\_date>as.Date("2021-06-13") & interview\_date<as.Date("2021-08-25"), 13, ifelse(interview\_date>as.Date("2021-10-11"), 15, 99))))))))  
  
vet\_freq3 <- followup\_data|>  
 clean\_names()|>  
 dplyr::select(interview\_date,household\_id,trtaniml)|>  
 filter(!is.na(trtaniml))#|>  
 # mutate(visit=ifelse(interview\_date<as.Date("2020-01-16"), 2, ifelse(interview\_date>as.Date("2020-01-16") & interview\_date<as.Date("2020-04-03"), 4, ifelse(interview\_date>as.Date("2020-05-01") & interview\_date<as.Date("2020-07-28"), 6, ifelse(interview\_date>as.Date("2020-09-10") & interview\_date<as.Date("2020-10-28"), 8, ifelse(interview\_date>as.Date("2021-01-05") & interview\_date<as.Date("2021-02-25"), 10, ifelse(interview\_date>as.Date("2021-02-25") & interview\_date<as.Date("2021-07-07"), 12, ifelse(interview\_date>as.Date("2021-08-17"), 14, 99))))))))  
vet\_freq <- vet\_freq1|>  
 rbind(vet\_freq2, vet\_freq3)|>  
 distinct()|>  
 mutate(trtaniml=ifelse(trtaniml%in%"Yes", 1, 0))|>  
 group\_by(household\_id)|>  
 arrange(interview\_date)|>  
 mutate(days\_recruitment=interview\_date-dplyr::lag(interview\_date))|>  
 mutate(days\_recruitment=ifelse(is.na(days\_recruitment), 0, days\_recruitment))|>  
 dplyr::select(-interview\_date)|>  
 ungroup()|>  
 group\_by(days\_recruitment, household\_id)|>  
 mutate(trtaniml=max(trtaniml, na.rm=T))|>  
 ungroup()|>  
 distinct()|>  
 mutate(trtaniml=ifelse(trtaniml%in%1, "Yes", "No"))  
  
#write\_csv(vet\_freq, "l4h\_vet\_frequency.csv")  
##grazing distance  
grazedist1 <- household\_baseline|>  
 clean\_names()|>  
 dplyr::select(interview\_date,household\_id,grazedist)|>  
 filter(!is.na(grazedist))  
  
grazedist2 <- household\_quarterly|>  
 clean\_names()|>  
 dplyr::select(interview\_date,household\_id,grazedist)|>  
 filter(!is.na(grazedist))#|>  
 # mutate(visit=ifelse(interview\_date<as.Date("2020-02-15"), 3, ifelse(interview\_date>as.Date("2020-04-05") & interview\_date<as.Date("2020-05-20"), 5, ifelse(interview\_date>as.Date("2020-05-20") & interview\_date<as.Date("2020-09-12"), 7, ifelse(interview\_date>as.Date("2020-09-13") & interview\_date<as.Date("2020-12-16"), 9, ifelse(interview\_date>as.Date("2021-02-24") & interview\_date<as.Date("2021-04-23"), 11, ifelse(interview\_date>as.Date("2021-06-13") & interview\_date<as.Date("2021-08-25"), 13, ifelse(interview\_date>as.Date("2021-10-11"), 15, 99))))))))  
  
grazedist3 <- followup\_data|>  
 clean\_names()|>  
 dplyr::select(interview\_date,household\_id,grazedist)|>  
 filter(!is.na(grazedist))|>  
 #mutate(visit=ifelse(interview\_date<as.Date("2020-01-16"), 2, ifelse(interview\_date>as.Date("2020-01-16") & interview\_date<as.Date("2020-04-03"), 4, ifelse(interview\_date>as.Date("2020-05-01") & interview\_date<as.Date("2020-07-28"), 6, ifelse(interview\_date>as.Date("2020-09-10") & interview\_date<as.Date("2020-10-28"), 8, ifelse(interview\_date>as.Date("2021-01-05") & interview\_date<as.Date("2021-02-25"), 10, ifelse(interview\_date>as.Date("2021-02-25") & interview\_date<as.Date("2021-07-07"), 12, ifelse(interview\_date>as.Date("2021-08-17"), 14, 99))))))))|>  
 mutate(grazedist=recode(grazedist, "1"="Below 1 km", "2"="1-5 km", "3"="5-10km", "4"="More than 10 km", "5"="migration"))  
  
grazedist <- grazedist1|>  
 rbind(grazedist2, grazedist3)|>  
 distinct()|>  
 mutate(grazedist=recode(grazedist,"Below 1 km"= "1", "1-5 km"="2", "5-10km"="3", "More than 10 km"="4", "migration"="5"))|>  
 mutate(grazedist=as.numeric(grazedist))|>  
 group\_by(household\_id)|>  
 arrange(interview\_date)|>  
 mutate(days\_recruitment=interview\_date-dplyr::lag(interview\_date))|>  
 mutate(days\_recruitment=ifelse(is.na(days\_recruitment), 0, days\_recruitment))|>  
 dplyr::select(-interview\_date)|>  
 ungroup()|>  
 group\_by(days\_recruitment, household\_id)|>  
 mutate(grazedist=max(grazedist, na.rm=T))|>  
 ungroup()|>  
 distinct()|>  
 mutate(grazedist=recode(grazedist, "1"="Below 1 km", "2"="1-5 km", "3"="5-10km", "4"="More than 10 km", "5"="migration"))  
  
##rcvd fds  
hh\_rcvd\_fd1 <- household\_baseline|>  
 clean\_names()|>  
 dplyr::select(interview\_date,household\_id,rcvdfds, fdsrc1)|>  
 mutate(fdscr\_project=ifelse(rcvdfds%in%"Yes" & fdsrc1%in%"provided by project", 1, 0))|>  
 mutate(fdscr\_other=ifelse(rcvdfds%in%"Yes" & fdsrc1%in%c("bought", "other", "provided by neighbor/family"), 1, 0))|>  
 filter(!is.na(rcvdfds))|>  
 dplyr::select(-fdsrc1)|>  
 distinct()  
  
  
hh\_rcvd\_fd2 <- household\_quarterly|>  
 clean\_names()|>  
 dplyr::select(interview\_date,household\_id,rcvdfds, fdsrc1)|>  
 mutate(fdscr\_project=ifelse(rcvdfds%in%"Yes" & fdsrc1%in%"provided by project", 1, 0))|>  
 mutate(fdscr\_other=ifelse(rcvdfds%in%"Yes" & fdsrc1%in%c("provided by neighbor/family", "other"), 1, 0))|>  
 filter(!is.na(rcvdfds))|>  
 dplyr::select(-fdsrc1)|>  
 distinct()#|>  
# mutate(visit=ifelse(interview\_date<as.Date("2020-02-15"), 3, ifelse(interview\_date>as.Date("2020-04-05") & interview\_date<as.Date("2020-05-20"), 5, ifelse(interview\_date>as.Date("2020-05-20") & interview\_date<as.Date("2020-09-12"), 7, ifelse(interview\_date>as.Date("2020-09-13") & interview\_date<as.Date("2020-12-16"), 9, ifelse(interview\_date>as.Date("2021-02-24") & interview\_date<as.Date("2021-04-23"), 11, ifelse(interview\_date>as.Date("2021-06-13") & interview\_date<as.Date("2021-08-25"), 13, ifelse(interview\_date>as.Date("2021-10-11"), 15, 99))))))))|>  
 #dplyr::select(-interview\_date)|>  
 #distinct()  
  
hh\_rcvd\_fd3 <- followup\_data|>  
 clean\_names()|>  
 dplyr::select(interview\_date,household\_id,rcvdfds, fdsrc)|>  
 mutate(fdscr\_project=ifelse(rcvdfds%in%"Yes" & fdsrc%in%"2", 1, 0))|>  
 mutate(fdscr\_other=ifelse(rcvdfds%in%"Yes" & fdsrc%in%c("3", "4"), 1, 0))|>  
 filter(!is.na(rcvdfds))|>  
 dplyr::select(-fdsrc)|>  
 distinct()  
 # mutate(visit=ifelse(interview\_date<as.Date("2020-01-16"), 2, ifelse(interview\_date>as.Date("2020-01-16") & interview\_date<as.Date("2020-04-03"), 4, ifelse(interview\_date>as.Date("2020-05-01") & interview\_date<as.Date("2020-07-28"), 6, ifelse(interview\_date>as.Date("2020-09-10") & interview\_date<as.Date("2020-10-28"), 8, ifelse(interview\_date>as.Date("2021-01-05") & interview\_date<as.Date("2021-02-25"), 10, ifelse(interview\_date>as.Date("2021-02-25") & interview\_date<as.Date("2021-07-07"), 12, ifelse(interview\_date>as.Date("2021-08-17"), 14, 99))))))))|>  
 # dplyr::select(-interview\_date)|>  
 # distinct()  
   
  
rcvd\_fds <- hh\_rcvd\_fd1|>  
 rbind(hh\_rcvd\_fd2, hh\_rcvd\_fd3)|>  
 distinct()|>  
 mutate(rcvdfds=ifelse(rcvdfds%in%"Yes", 1, 0))|>  
 group\_by(household\_id)|>  
 arrange(interview\_date)|>  
 mutate(days\_recruitment=interview\_date-dplyr::lag(interview\_date))|>  
 mutate(days\_recruitment=ifelse(is.na(days\_recruitment), 0, days\_recruitment))|>  
 dplyr::select(-interview\_date)|>  
 ungroup()|>  
 group\_by(days\_recruitment, household\_id)|>  
 mutate(rcvdfds=max(rcvdfds, na.rm=T))|>  
 mutate(fdscr\_project=max(fdscr\_project, na.rm=T))|>  
 mutate(fdscr\_other=max(fdscr\_other, na.rm=T))|>  
 ungroup()|>  
 distinct()|>  
 mutate(rcvdfds=ifelse(rcvdfds%in%1, "Yes", "No"))  
  
#write\_csv(rcvd\_fds, "l4h\_rcvdfds.csv")  
   
##feed type bought  
#only one hh bought hay  
  
##feed amount (provided by project-range cubes)  
#0.3 0.4 0.5 1 150 200   
 #1 1 1 1 1 11  
   
##household illness  
hh\_sick1 <- household\_baseline|>  
 clean\_names()|>  
 dplyr::select(interview\_date,household\_id,hhmmbrsick)|>  
 filter(!is.na(hhmmbrsick))  
  
hh\_sick2 <- household\_quarterly|>  
 clean\_names()|>  
 dplyr::select(interview\_date,household\_id,hhmmbrsick)|>  
 filter(!is.na(hhmmbrsick))#|>  
 # mutate(visit=ifelse(interview\_date<as.Date("2020-02-15"), 3, ifelse(interview\_date>as.Date("2020-04-05") & interview\_date<as.Date("2020-05-20"), 5, ifelse(interview\_date>as.Date("2020-05-20") & interview\_date<as.Date("2020-09-12"), 7, ifelse(interview\_date>as.Date("2020-09-13") & interview\_date<as.Date("2020-12-16"), 9, ifelse(interview\_date>as.Date("2021-02-24") & interview\_date<as.Date("2021-04-23"), 11, ifelse(interview\_date>as.Date("2021-06-13") & interview\_date<as.Date("2021-08-25"), 13, ifelse(interview\_date>as.Date("2021-10-11"), 15, 99))))))))|>  
 #dplyr::select(-interview\_date)|>  
 #distinct()  
   
hh\_sick3 <- followup\_data|>  
 clean\_names()|>  
 dplyr::select(interview\_date,household\_id,hhmmbrsick)|>  
 filter(!is.na(hhmmbrsick))|>  
 # mutate(visit=ifelse(interview\_date<as.Date("2020-01-16"), 2, ifelse(interview\_date>as.Date("2020-01-16") & interview\_date<as.Date("2020-04-03"), 4, ifelse(interview\_date>as.Date("2020-05-01") & interview\_date<as.Date("2020-07-28"), 6, ifelse(interview\_date>as.Date("2020-09-10") & interview\_date<as.Date("2020-10-28"), 8, ifelse(interview\_date>as.Date("2021-01-05") & interview\_date<as.Date("2021-02-25"), 10, ifelse(interview\_date>as.Date("2021-02-25") & interview\_date<as.Date("2021-07-07"), 12, ifelse(interview\_date>as.Date("2021-08-17"), 14, 99))))))))|>  
 # dplyr::select(-interview\_date)|>  
 mutate(hhmmbrsick=recode(hhmmbrsick, "1"="Yes", "0"="No", "2"="Don't know"))|>  
 distinct()  
  
hh\_sick <- hh\_sick1|>  
 rbind(hh\_sick2, hh\_sick3)|>  
 mutate(hhmmbrsick=recode(hhmmbrsick, "Don't know"="No"))|>  
 distinct()|>  
 mutate(hhmmbrsick=ifelse(hhmmbrsick%in%"Yes", 1, 0))|>  
 group\_by(household\_id)|>  
 arrange(interview\_date)|>  
 mutate(days\_recruitment=interview\_date-dplyr::lag(interview\_date))|>  
 mutate(days\_recruitment=ifelse(is.na(days\_recruitment), 0, days\_recruitment))|>  
 dplyr::select(-interview\_date)|>  
 ungroup()|>  
 group\_by(days\_recruitment, household\_id)|>  
 mutate(hhmmbrsick=max(hhmmbrsick, na.rm=T))|>  
 ungroup()|>  
 distinct()|>  
 mutate(hhmmbrsick=ifelse(hhmmbrsick%in%1, "Yes", "No"))  
   
#write\_csv(hh\_sick, "l4h\_hh\_sick.csv")   
### average milk yield: species  
#milk:baseline  
household\_bl1 <- household\_baseline|>  
 dplyr::select(interview\_date,household\_id,village,cwslactatn, shplactatn,gtslactatn,cmlslactatn, cows\_litres,sheep\_litres,goats\_litres,camels\_litres, dtefdprvdd)  
  
#milk:followup  
followup\_bl1 <- followup\_data|>  
 clean\_names()|>  
 dplyr::select(interview\_date,household\_id,village,cwslactatn, shplactatn,gtslactatn,cmlslactatn,cows\_litres,sheep\_litres,goats\_litres,camels\_litres, dtefdprvdd)#|>  
 #rename(dtefdprvdd=form\_dtefdprvdd)|>  
 # mutate(visit=ifelse(interview\_date<as.Date("2020-01-16"), 2, ifelse(interview\_date>as.Date("2020-01-16") & interview\_date<as.Date("2020-04-03"), 4, ifelse(interview\_date>as.Date("2020-05-01") & interview\_date<as.Date("2020-07-28"), 6, ifelse(interview\_date>as.Date("2020-09-10") & interview\_date<as.Date("2020-10-28"), 8, ifelse(interview\_date>as.Date("2021-01-05") & interview\_date<as.Date("2021-02-25"), 10, ifelse(interview\_date>as.Date("2021-02-25") & interview\_date<as.Date("2021-07-07"), 12, ifelse(interview\_date>as.Date("2021-08-17"), 14, 99))))))))  
household\_q1 <- household\_quarterly|>  
 dplyr::select(interview\_date,household\_id,village,cwslactatn, shplactatn,gtslactatn,cmlslactatn,cows\_litres,sheep\_litres,goats\_litres,camels\_litres,dtefdprvdd)#|>  
 # mutate(visit=ifelse(interview\_date<as.Date("2020-02-15"), 3, ifelse(interview\_date>as.Date("2020-04-05") & interview\_date<as.Date("2020-05-20"), 5, ifelse(interview\_date>as.Date("2020-05-20") & interview\_date<as.Date("2020-09-12"), 7, ifelse(interview\_date>as.Date("2020-09-13") & interview\_date<as.Date("2020-12-16"), 9, ifelse(interview\_date>as.Date("2021-02-24") & interview\_date<as.Date("2021-04-23"), 11, ifelse(interview\_date>as.Date("2021-06-13") & interview\_date<as.Date("2021-08-25"), 13, ifelse(interview\_date>as.Date("2021-10-11"), 15, 99))))))))  
household\_comb <- household\_bl1|>  
 rbind(household\_q1|>mutate(dtefdprvdd=NA), followup\_bl1)|>  
 mutate(study\_arm=ifelse(village%in%c("Lependera","Gobb Arbelle", "Nahgan-ngusa","Sulate", "Saale-Sambakah", "Namarei", "Manyatta Lengima","Lokoshula", "TubchaDakhane", "Rengumo-Gargule"), "Intervention arm 1", ifelse(village%in%c("Galthelian-Torrder", "Uyam village", "Galthelan Elemo", "Nebey", "Rongumo\_kurkum", "Urawen\_Kurkum", "Eisimatacho", "Manyatta K.A.G", "Ltepes Ooodo", "Lorokushu", "Marti", "Manyatta Juu West/East", "Lbaarok1"), "Intervention arm 2","Control arm")))  
  
###check ward arms  
# wards <- households\_baseline2|>  
# dplyr::select(ward,village)|>  
# mutate(study\_arm=ifelse(village%in%c("Lependera","Gobb Arbelle", "Nahgan-ngusa","Sulate", "Saale-Sambakah", "Namarei", "Manyatta Lengima","Lokoshula", "TubchaDakhane", "Rengumo-Gargule"), "Intervention arm 1", ifelse(village%in%c("Galthelian-Torrder", "Uyam village", "Galthelan Elemo", "Nebey", "Rongumo\_kurkum", "Urawen\_Kurkum", "Eisimatacho", "Manyatta K.A.G", "Ltepes Ooodo", "Lorokushu", "Marti", "Manyatta Juu West/East", "Lbaarok1"), "Intervention arm 2","Control arm")))|>  
# distinct()  
  
xx<- household\_baseline|>clean\_names()|>  
 dplyr::select(interview\_date,household\_id,hhpple, h\_hoccuptn1, hhead, cshtrnsfrprog1,h\_heducation)|>  
 group\_by(household\_id)|>  
 dplyr::slice(1L)  
### milk yield  
hh\_milk<- distinct(household\_comb)|>  
 #dplyr::select(-interview\_date)|>  
 distinct()|>  
 left\_join(xx, by=c("household\_id", "interview\_date"))|>  
 group\_by(household\_id)|>  
 arrange(interview\_date)|>  
 mutate(days\_recruitment=interview\_date-dplyr::lag(interview\_date))|>  
 mutate(days\_recruitment=ifelse(is.na(days\_recruitment), 0, days\_recruitment))|>  
 #mutate(interview\_week=floor\_date(interview\_date,"week"))|>  
 pivot\_longer(c(cows\_litres, sheep\_litres, camels\_litres, goats\_litres), names\_to="animals\_ltrs", values\_to="animal\_litres")|>  
 pivot\_longer(c(cwslactatn, shplactatn,gtslactatn,cmlslactatn), names\_to="spcs", values\_to="species1")|>  
 group\_by(household\_id, days\_recruitment)|>  
 mutate(animal\_litres=as.numeric(animal\_litres)) |>  
 mutate(total\_litres= sum(animal\_litres, na.rm=T))|>  
 mutate(hhpple=max(hhpple))|>  
 ungroup()|>  
 #group\_by(household\_id, days\_recruitment, animals\_ltrs)|>  
 #mutate(total\_litres\_anim= sum(animal\_litres, na.rm=T))|>  
 #ungroup()|>  
 dplyr::select(household\_id,hhpple,study\_arm,animals\_ltrs, total\_litres,animal\_litres, dtefdprvdd, species1, h\_hoccuptn1, hhead, cshtrnsfrprog1, days\_recruitment, interview\_date, h\_heducation)|>  
 filter(!(is.na(animal\_litres)))|>  
 separate(animals\_ltrs, into = c("species","ltrs"),sep = "\_")|>  
 dplyr::select(household\_id,hhpple,study\_arm,total\_litres,animal\_litres, dtefdprvdd, h\_hoccuptn1, hhead, cshtrnsfrprog1, days\_recruitment, interview\_date, h\_heducation, species)|>  
 mutate(dtefdprvdd=ifelse(is.na(dtefdprvdd), 1, 0))|>  
 ungroup()|>  
 distinct()|>  
 # mutate(spcs=recode(spcs, "cwslactatn"="Cows", "shplactatn"="Sheep","gtslactatn"="Goats","cmlslactatn"="Camels"))|>  
 distinct()

## Warning: There were 8754 warnings in `mutate()`.  
## The first warning was:  
## ℹ In argument: `animal\_litres = as.numeric(animal\_litres)`.  
## ℹ In group 1: `household\_id = "KOHFGAHH001"`, `days\_recruitment = 0`.  
## Caused by warning:  
## ! NAs introduced by coercion  
## ℹ Run `dplyr::last\_dplyr\_warnings()` to see the 8753 remaining warnings.

#write\_csv(hh\_milk,"l4h\_hh\_milk.csv")   
  
hh\_milk1 <- hh\_milk|>  
 mutate(interview\_week=floor\_date(interview\_date, "week"))|>  
 mutate(season= ifelse(interview\_week<as.Date("2020-02-08"), "Season 1", ifelse(interview\_week>as.Date("2020-02-07") & interview\_week<as.Date("2020-05-29"), "Season 2- (Dry)", ifelse(interview\_week>as.Date("2020-05-28") & interview\_week<as.Date("2020-08-19"), "Season 3", ifelse(interview\_week>as.Date("2020-08-18") & interview\_week<as.Date("2020-12-01"), "Season 4 (Dry)", ifelse(interview\_week>as.Date("2020-11-29") & interview\_week<as.Date("2021-01-13"), "Season 5", ifelse(interview\_week>as.Date("2021-01-12") & interview\_week<as.Date("2021-05-01"),"Season 6 (Dry)", ifelse(interview\_week>as.Date("2021-04-30") & interview\_week<as.Date("2021-08-30"), "Season 7", "Season 8-(Dry)"))))))))|>  
 group\_by(season, household\_id)|>  
 mutate(feeds\_date1=min(dtefdprvdd))|>  
 mutate(feeds\_date2=feeds\_date1+90)|>  
 mutate(feeds\_date3=ifelse(is.na(feeds\_date1), "No", "Yes"))|>  
 group\_by(season,household\_id)|>  
 mutate(total\_milk=sum(animal\_litres))|>  
 ungroup()|>  
 distinct(interview\_week,household\_id,hhpple,study\_arm,animal\_litres,total\_milk, season)|>  
 mutate(treatment=ifelse(study\_arm%in%"Control arm", "No", "Yes"))|>  
 group\_by(season,study\_arm)|>  
 mutate(avrg\_milk=mean(total\_milk,na.rm=T))|>  
 ungroup()|>  
 mutate(study\_arm=recode(study\_arm, "1"="Intervention arm 1 (feeds)", "2"="Intervention arm 2 (feeds+ counselling)", "3"="Control arm"))  
  
  
# hh\_milk1a<- hh\_milk|>  
# mutate(species=str\_to\_title(species))|>  
# group\_by( visit, household\_id)|>  
# mutate(species2= ifelse(species==spcs, species1, NA))|>  
# ungroup()|>  
# dplyr::select(-species1, -spcs)|>  
# distinct()|>  
# filter(!is.na(species2))|>  
# group\_by(household\_id, visit, species)|>  
# mutate(animal\_litres=sum(animal\_litres, na.rm=T))|>  
# #mutate(species2=mean(species2, na.rm=T))|>  
# mutate(average=animal\_litres/species2, na.rm=T)|>  
# ungroup()|>  
# dplyr::select(household\_id,hhpple,study\_arm, interview\_week,species, animal\_litres, average, visit)|>  
# mutate(season= ifelse(interview\_week<as.Date("2020-02-08"), "Season 1", ifelse(interview\_week>as.Date("2020-02-07") & interview\_week<as.Date("2020-05-29"), "Season 2- (Dry)", ifelse(interview\_week>as.Date("2020-05-28") & interview\_week<as.Date("2020-08-19"), "Season 3", ifelse(interview\_week>as.Date("2020-08-18") & interview\_week<as.Date("2020-12-01"), "Season 4 (Dry)", ifelse(interview\_week>as.Date("2020-11-29") & interview\_week<as.Date("2021-01-13"), "Season 5", ifelse(interview\_week>as.Date("2021-01-12") & interview\_week<as.Date("2021-05-01"),"Season 6 (Dry)", ifelse(interview\_week>as.Date("2021-04-30") & interview\_week<as.Date("2021-08-30"), "Season 7", "Season 8-(Dry)"))))))))  
  
#write\_csv(hh\_milk1,"milk\_yield.csv")  
ses\_data<- scores1a|>  
 dplyr::select(household\_id, visit, income)|>  
 distinct()  
  
ses\_data1<- ses\_data|>  
 mutate(visit=visit+1)|>  
 filter(visit<15)  
  
ses\_data2<- rbind(ses\_data, ses\_data1)  
rs<- ses\_data|>  
 group\_by(household\_id, visit, income)|>  
 count()  
#write\_csv(ses\_data2, "ses\_scores.csv")  
## nutritional\_counselling  
  
nutr1<- mother\_baseline|>  
 dplyr::select(interview\_date,household\_id, nutrtn\_cnslng, tms\_nutrcouns)  
  
xx<- followup\_data |>  
 select(number, household\_id, interview\_date)  
  
mother\_followup<- left\_join(mother\_followup, xx, by="number")  
nutr2 <- mother\_followup|>  
 dplyr::select(interview\_date,household\_id,nutrtn\_cnslng, tms\_nutrcouns)#|>  
 # mutate(nut\_counseling= recode(nutrtn\_cnslng,"1"="Yes", "0"="No"))|>  
# mutate(interview\_week=floor\_date(interview\_date,"week"))|>  
 # dplyr::select(-interview\_date,-nutrtn\_cnslng)  
nutr3 <- mother\_quarterly|>  
 dplyr::select(interview\_date,household\_id,nutrtn\_cnslng, tms\_nutrcouns)#|>  
 # mutate(nut\_counseling= recode(nutrtn\_cnslng,"1"="Yes", "0"="No"))|>  
# mutate(interview\_week=floor\_date(interview\_date,"week"))|>  
 # dplyr::select(-interview\_date,-nutrtn\_cnslng)  
nutr <- rbind(nutr1,nutr2,nutr3)|>  
 group\_by(household\_id)|>  
 arrange(interview\_date)|>  
 mutate(days\_recruitment=interview\_date-dplyr::lag(interview\_date))|>  
 mutate(days\_recruitment=ifelse(is.na(days\_recruitment), 0, days\_recruitment))|>  
 ungroup()|>  
 mutate(village=str\_sub(household\_id, 5,6))|>  
 mutate(study\_arm=ifelse(village%in%c("LP","GA", "NN","SL", "SS", "NM", "ML","LK", "TD", "RG"), "Intervention arm 1", ifelse(village%in%c("GT", "UY", "GE", "NB", "RK", "UK", "ES", "MK", "LO", "LR", "MR", "MJ", "LB"), "Intervention arm 2","Control arm")))|>  
 mutate(normal\_counseling=ifelse(study\_arm%in%c("Intervention arm 1", "Control arm") & tms\_nutrcouns>0, 1,0))|>  
 mutate(normal\_counseling=ifelse(study\_arm%in%c("Intervention arm 1", "Control arm") & is.na(normal\_counseling), 0, normal\_counseling))|>  
 mutate(enhanced\_counseling=ifelse(study\_arm%in%"Intervention arm 2" & tms\_nutrcouns>2, 1, 0))|>  
 mutate(tms\_nutrcouns=ifelse(normal\_counseling==1 & tms\_nutrcouns>2, 2, tms\_nutrcouns))|>  
 mutate(tms\_nutrcouns=ifelse(tms\_nutrcouns==(-2), 2, tms\_nutrcouns))|>  
 dplyr::select(-village, -nutrtn\_cnslng, -interview\_date)  
  
   
  
  
###dataset for model  
obj1\_data <- lvstckown3a|>dplyr::select(household\_id, cattle, sheep, goats, camels, days\_recruitment)|>mutate(days\_recruitment=as.numeric(days\_recruitment))|>left\_join(hh\_milk|>distinct()|>mutate(days\_recruitment=as.numeric(days\_recruitment)), by=c("days\_recruitment","household\_id"))|>mutate\_at(vars(cattle, sheep, goats, camels), funs(abs(.)))

## Warning in left\_join(mutate(dplyr::select(lvstckown3a, household\_id, cattle, : Detected an unexpected many-to-many relationship between `x` and `y`.  
## ℹ Row 12 of `x` matches multiple rows in `y`.  
## ℹ Row 94 of `y` matches multiple rows in `x`.  
## ℹ If a many-to-many relationship is expected, set `relationship =  
## "many-to-many"` to silence this warning.

## Warning: `funs()` was deprecated in dplyr 0.8.0.  
## ℹ Please use a list of either functions or lambdas:  
##   
## # Simple named list: list(mean = mean, median = median)  
##   
## # Auto named with `tibble::lst()`: tibble::lst(mean, median)  
##   
## # Using lambdas list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

obj1\_data <- obj1\_data|>left\_join(lct\_animals|>distinct()|>mutate(days\_recruitment=as.numeric(days\_recruitment)),by=c("days\_recruitment","household\_id"))|>rename(milk\_tlu=total\_tlu)

## Warning in left\_join(obj1\_data, mutate(distinct(lct\_animals), days\_recruitment = as.numeric(days\_recruitment)), : Detected an unexpected many-to-many relationship between `x` and `y`.  
## ℹ Row 12 of `x` matches multiple rows in `y`.  
## ℹ Row 12 of `y` matches multiple rows in `x`.  
## ℹ If a many-to-many relationship is expected, set `relationship =  
## "many-to-many"` to silence this warning.

obj1\_data <- obj1\_data|>left\_join(ses\_data,by=c("visit","household\_id"))

## Warning in left\_join(obj1\_data, ses\_data, by = c("visit", "household\_id")): Detected an unexpected many-to-many relationship between `x` and `y`.  
## ℹ Row 13 of `x` matches multiple rows in `y`.  
## ℹ Row 381 of `y` matches multiple rows in `x`.  
## ℹ If a many-to-many relationship is expected, set `relationship =  
## "many-to-many"` to silence this warning.

obj1\_data <- obj1\_data|>left\_join(animal\_sick|>distinct()|>mutate(days\_recruitment=as.numeric(days\_recruitment)),by=c("days\_recruitment","household\_id"))

## Warning in left\_join(obj1\_data, mutate(distinct(animal\_sick), days\_recruitment = as.numeric(days\_recruitment)), : Detected an unexpected many-to-many relationship between `x` and `y`.  
## ℹ Row 1327 of `x` matches multiple rows in `y`.  
## ℹ Row 8975 of `y` matches multiple rows in `x`.  
## ℹ If a many-to-many relationship is expected, set `relationship =  
## "many-to-many"` to silence this warning.

obj1\_data <- obj1\_data|>left\_join(sick\_animals|>distinct()|>mutate(days\_recruitment=as.numeric(days\_recruitment)),by=c("days\_recruitment","household\_id"))|>rename(sick\_tlu=total\_tlu)  
obj1\_data <- obj1\_data|>left\_join(abortions|>mutate(days\_recruitment=as.numeric(days\_recruitment))|>distinct(),by=c("days\_recruitment","household\_id"))  
obj1\_data <- obj1\_data|>left\_join(vet\_freq|>mutate(days\_recruitment=as.numeric(days\_recruitment))|>distinct(),by=c("days\_recruitment","household\_id"))  
obj1\_data <- obj1\_data|>left\_join(grazedist|>mutate(days\_recruitment=as.numeric(days\_recruitment))|>distinct(),by=c("days\_recruitment","household\_id"))  
obj1\_data <- obj1\_data|>left\_join(rcvd\_fds|>distinct()|>mutate(days\_recruitment=as.numeric(days\_recruitment)),by=c("days\_recruitment","household\_id"))  
obj1\_data <- obj1\_data|>left\_join(hh\_sick|>distinct()|>mutate(days\_recruitment=as.numeric(days\_recruitment)),by=c("days\_recruitment","household\_id"))  
obj1\_data <- obj1\_data|>left\_join(nutr|>distinct()|>mutate(days\_recruitment=as.numeric(days\_recruitment)),by=c("days\_recruitment","household\_id"))

## Warning in left\_join(obj1\_data, mutate(distinct(nutr), days\_recruitment = as.numeric(days\_recruitment)), : Detected an unexpected many-to-many relationship between `x` and `y`.  
## ℹ Row 4 of `x` matches multiple rows in `y`.  
## ℹ Row 5 of `y` matches multiple rows in `x`.  
## ℹ If a many-to-many relationship is expected, set `relationship =  
## "many-to-many"` to silence this warning.

obj1\_data <- obj1\_data|>left\_join(births|>distinct()|>mutate(days\_recruitment=as.numeric(days\_recruitment)),by=c("days\_recruitment","household\_id"))

## Warning in left\_join(obj1\_data, mutate(distinct(births), days\_recruitment = as.numeric(days\_recruitment)), : Detected an unexpected many-to-many relationship between `x` and `y`.  
## ℹ Row 95 of `x` matches multiple rows in `y`.  
## ℹ Row 7770 of `y` matches multiple rows in `x`.  
## ℹ If a many-to-many relationship is expected, set `relationship =  
## "many-to-many"` to silence this warning.

#obj1\_data<- obj1\_data|>left\_join(hh\_milk1a|>dplyr::select(species, household\_id, days\_recruitment,average)|>mutate(days\_recruitment=as.numeric(days\_recruitment)), by=c("days\_recruitment","household\_id"))|>distinct()|>  
 # mutate(species=recode(species, "Camels"="camels\_milk", "Cows"="cows\_milk", "Goats"="goats\_milk", "Sheep"="sheep\_milk"))|>  
# pivot\_wider(names\_from=species,values\_from = average )|>  
# distinct()|>  
# unnest()  
  
  
# obj1\_data<- obj1\_data|>  
# mutate\_at(vars(total\_litres, camels\_milk, goats\_milk, sheep\_milk, cows\_milk, hhpple), funs(ifelse(is.na(.), 0,.)))|>  
# group\_by(household\_id, days\_recruitment)|>  
# mutate\_at(vars(total\_litres, cattle, sheep, goats, camels, camels\_milk, goats\_milk, sheep\_milk, cows\_milk, hhpple), funs(max(., na.rm=T)))|>  
# #mutate(study\_arm=ifelse(is.na(study\_arm), dplyr::lag(study\_arm), study\_arm))|>  
# #mutate(hhmmbrsick=ifelse(hhmmbrsick%in%"Yes", "Yes", "No"))|>  
# ungroup()|>  
# dplyr::select(-grazedist)|>  
# distinct()  
  
  
 #mutate(grazedist=recode(grazedist, "1"="Below 1 km", "2"="1-5 km", "3"="5-10km", "4"="More than 10 km", "5"="migration"))  
  
obj1\_data<- obj1\_data|>  
 #dplyr::select(-interview\_date)|>  
 mutate(village=str\_sub(household\_id, 5,6))|>  
 mutate(study\_arm=ifelse(village%in%c("LP","GA", "NN","SL", "SS", "NM", "ML","LK", "TD", "RG"), "Intervention arm 1", ifelse(village%in%c("GT", "UY", "GE", "NB", "RK", "UK", "ES", "MK", "LO", "LR", "MR", "MJ", "LB"), "Intervention arm 2", "Control arm")))|>  
 dplyr::select(-village)|>  
 distinct()|>  
 mutate(dtefdprvdd=ifelse(is.na(dtefdprvdd), 0, dtefdprvdd))|>  
 group\_by(household\_id, days\_recruitment)|>  
 mutate(dtefdprvdd=max(dtefdprvdd, na.rm=T))|>  
 ungroup()|>  
 distinct()|>  
 group\_by(household\_id, days\_recruitment)|>  
 mutate(tlu=sum(cattle+camels+(sheep/10)+(goats/10)))|>  
 dplyr::select(household\_id, days\_recruitment, total\_litres, study\_arm, hhpple, h\_hoccuptn1, hhead, cshtrnsfrprog1, total\_milk\_animals, milk\_tlu,sick\_tlu, grazedist, rcvdfds, hhmmbrsick, tlu , fdscr\_other, fdscr\_project, h\_hoccuptn1, h\_heducation, normal\_counseling , enhanced\_counseling, tms\_nutrcouns, trtaniml, income, no\_births )  
   
  
  
###combine intervention arms  
#obj1\_data$treatment\_arm <- ifelse(obj1\_data$study\_arm=="Control arm","Control arm","Treatment arm (1&2)")  
  
###set parameter references  
obj1\_data$study\_arm <- fct\_relevel(obj1\_data$study\_arm, "Control arm")  
#obj1\_data$anmlsick <- fct\_relevel(obj1\_data$anmlsick, "No")  
obj1\_data$trtaniml <- fct\_relevel(obj1\_data$trtaniml, "No")  
obj1\_data$grazedist <- fct\_relevel(obj1\_data$grazedist, "migration")  
obj1\_data$rcvdfds <- fct\_relevel(obj1\_data$rcvdfds, "No")  
obj1\_data$hhmmbrsick <- fct\_relevel(obj1\_data$hhmmbrsick, "Yes")  
  
#obj1\_data$season<- fct\_relevel(obj1\_data$season, "Non-dry")  
  
###convert milk to litres  
#obj1\_data <- obj1\_data|>  
 #mutate(total\_milk1=(total\_milk/1000))|>  
 #mutate(animal\_litres1=animal\_litres/1000)  
#obj1\_data$total\_milk1 <- as.numeric(obj1\_data$total\_milk1)  
  
###check milk yield distribution  
#hist(obj1\_data$total\_milk1) #positively/right skewed. Therefore, negbinomial model  
  
###subset  
#obj1\_data1 <- obj1\_data|>  
 #filter(!study\_arm=="Intervention arm 1")  
#obj1\_data1$study\_arm <- factor(obj1\_data1$study\_arm)  
#obj1\_data1 <- within(obj1\_data1,study\_arm <- relevel(study\_arm,ref= "Intervention arm 2"))  
# data2<- obj1\_data|>  
# filter(study\_arm%in%c("Intervention arm 1", "Intervention arm 2"))  
  
### panel data  
  
# obj1\_data<- obj1\_data|>  
# dplyr::select( -rcvdfds, -hhmmbrsick, -anmlsick, -trtaniml)|>  
# distinct()  
obj1\_data$days\_recruitment<- ifelse(is.na(obj1\_data$days\_recruitment), 0, obj1\_data$days\_recruitment)  
  
obj1\_data<- obj1\_data|>  
 mutate(h\_hoccuptn1=ifelse(h\_hoccuptn1%in%"Livestock herding", 1, 2))|>  
 mutate(hhead=ifelse(hhead%in%"Yes", 1, 0))|>  
 mutate(cshtrnsfrprog1=ifelse(cshtrnsfrprog1%in%"None", 1, 0))|>  
 mutate(h\_heducation=ifelse(h\_heducation%in%"No", 0, 1))|>  
 mutate(grazedist=recode(grazedist,"Below 1 km"= "1", "1-5 km"="2", "5-10km"="3", "More than 10 km"="4", "migration"="5"))|>  
 mutate(grazedist=as.numeric(grazedist))|>  
 mutate(rcvdfds=ifelse(rcvdfds%in%"Yes", 1, 0))|>  
 mutate(hhmmbrsick=ifelse(hhmmbrsick%in%"Yes", 1, 0))|>  
 mutate(income=recode(income, "Forth"=4, "Middle"=3, "Poorest"=1, "Richest"=5, "Second"=2))|>  
 mutate\_at(vars(total\_litres, hhpple, sick\_tlu, milk\_tlu, hhmmbrsick, rcvdfds, grazedist, cshtrnsfrprog1, hhead, h\_hoccuptn1, tlu, fdscr\_other, fdscr\_project, h\_heducation, total\_milk\_animals, tms\_nutrcouns, enhanced\_counseling, normal\_counseling, income, no\_births), funs(ifelse(is.na(.), 0, .)))|>  
 group\_by(household\_id, days\_recruitment)|>  
 mutate\_at(vars(total\_litres, hhpple, sick\_tlu, milk\_tlu, hhmmbrsick, rcvdfds, grazedist, cshtrnsfrprog1, hhead, h\_hoccuptn1, tlu, fdscr\_other, fdscr\_project, h\_heducation, total\_milk\_animals, tms\_nutrcouns, enhanced\_counseling, normal\_counseling, income, no\_births), funs(max(., na.rm=T)))|>  
 ungroup()|>  
 # group\_by(household\_id, days\_recruitment, animals\_ltrs)|>  
 # mutate(total\_litres\_anim=max(total\_litres\_anim, na.rm=T))|>  
 #ungroup()|>  
 distinct()|>  
 mutate(h\_hoccuptn1=ifelse(h\_hoccuptn1%in%1, "Livestock herding", "Non-livestock"))|>  
 mutate(hhead=ifelse(hhead%in%1, "Female", "Male"))|>  
 mutate(cshtrnsfrprog1=ifelse(cshtrnsfrprog1%in%0, "Yes", "No"))|>  
 mutate(grazedist=recode(grazedist,"1"="Below 1 km", "2"="1-5 km", "3"="5-10km","4"="More than 10 km","5"="migration"))|>  
 mutate(rcvdfds=ifelse(rcvdfds%in%1, "Yes", "No"))|>  
 mutate(hhmmbrsick=ifelse(hhmmbrsick%in%1, "Yes","No"))|>  
 mutate(fdscr\_other=ifelse(fdscr\_other%in%1, "Yes", "No"))|>  
 mutate(fdscr\_project=ifelse(fdscr\_project%in%1, "No", "Yes"))|>  
 mutate(income=recode(income, "4"="Forth", "3"="Middle", "1"="Poorest", "5"="Richest", "2"="Second"))

## Warning: `funs()` was deprecated in dplyr 0.8.0.  
## ℹ Please use a list of either functions or lambdas:  
##   
## # Simple named list: list(mean = mean, median = median)  
##   
## # Auto named with `tibble::lst()`: tibble::lst(mean, median)  
##   
## # Using lambdas list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

## Warning: `funs()` was deprecated in dplyr 0.8.0.  
## ℹ Please use a list of either functions or lambdas:  
##   
## # Simple named list: list(mean = mean, median = median)  
##   
## # Auto named with `tibble::lst()`: tibble::lst(mean, median)  
##   
## # Using lambdas list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

## Warning: There was 1 warning in `mutate()`.  
## ℹ In argument: `income = recode(...)`.  
## Caused by warning:  
## ! Unreplaced values treated as NA as `.x` is not compatible.  
## Please specify replacements exhaustively or supply `.default`.

obj1\_data$fdscr\_other<- ifelse(obj1\_data$days\_recruitment<40, "No",obj1\_data$fdscr\_other )  
obj1\_data$total\_litres<- ifelse(obj1\_data$fdscr\_other%in%"Yes", NA, obj1\_data$total\_litres )  
  
pp<-data.frame(lapply(obj1\_data[, "milk\_tlu"], # variable indices 3:5  
 function(y) ave(y, obj1\_data$days\_recruitment, FUN=function(x)   
 ifelse(x < quantile(x, .01) | x > quantile(x, .99), NA, x))))   
  
mm<-data.frame(lapply(obj1\_data[, "total\_litres"], # variable indices 3:5  
 function(y) ave(y, obj1\_data$days\_recruitment, FUN=function(x)   
 ifelse(x < quantile(x, .01, na.rm = T) | x > quantile(x, .99, na.rm = T), NA, x))))   
  
  
  
  
  
obj1\_data$milk\_tlu1<- pp$milk\_tlu  
obj1\_data$total\_litres1<- mm$total\_litres  
  
  
obj1\_data|>  
 distinct()|>  
 dplyr::select(household\_id, days\_recruitment)|>  
 is.pbalanced()

## [1] FALSE

# camels<- obj1\_data|>  
# filter(animals\_ltrs%in%"camels\_litres")|>  
# distinct()  
#   
# sheep<- obj1\_data|>  
# filter(animals\_ltrs%in%"sheep\_litres")|>  
# distinct()  
#   
# goats<- obj1\_data|>  
# filter(animals\_ltrs%in%"goats\_litres")|>  
# distinct()  
#   
# cows<- obj1\_data|>  
# filter(animals\_ltrs%in%"cows\_litres")|>  
# distinct()  
  
  
dta<-obj1\_data|>dplyr::select(household\_id, days\_recruitment)|>  
 group\_by(household\_id, days\_recruitment)|>  
 dplyr::count()  
  
  
obj1\_data1 <- make.pbalanced(obj1\_data, balance.type = "fill")  
obj1\_data$grazedist<- ifelse(obj1\_data$grazedist%in%"migration", "Fora", "At home")  
obj1\_data1<- distinct(obj1\_data1)  
  
#grazedist+hhmmbrsick+trtaniml  
obj1\_data$hhead<-fct\_relevel(obj1\_data$hhead, "Female")  
  
obj1\_data$h\_hoccuptn1<- fct\_relevel(obj1\_data$h\_hoccuptn1, "Non-livestock")  
  
obj1\_data$income<- fct\_relevel(obj1\_data$income, "Poorest")  
  
obj1\_data$fdscr\_project1<- ifelse(obj1\_data$fdscr\_project%in%"Yes", 1, 0)  
obj1\_data$enhanced\_counseling1<- ifelse(obj1\_data$enhanced\_counseling%in%"1", 0, 1)  
obj1\_data$fdscr\_project1<- as.numeric(obj1\_data$fdscr\_project1)  
  
obj1\_data<- obj1\_data|>  
 group\_by(household\_id, days\_recruitment)|>  
 mutate(nonmilk\_tlu=tlu-milk\_tlu)|>  
 ungroup()  
  
obj1\_data$milk\_tlu<- obj1\_data$milk\_tlu\*7  
#obj1\_data1$total\_litres<- obj1\_data1$total\_litres\*1000  
#+hhpple+sick\_tlu+cshtrnsfrprog1+h\_hoccuptn1+hhead+milk\_tlu+sick\_tlu+aborted\_tlu+trtaniml+grazedist+hhmmbrsick+rcvdfds  
  
  
random\_model1a <- plm(total\_litres~study\_arm, data=obj1\_data , index=c("household\_id", "days\_recruitment"), effect = "individual", model="pooling")  
  
  
random\_model1 <- plm(total\_litres~cshtrnsfrprog1+nonmilk\_tlu+milk\_tlu1+fdscr\_project+enhanced\_counseling, data=obj1\_data , index=c("household\_id", "days\_recruitment"), effect = "individual", model="random")  
  
within\_model1\_uni <- plm(total\_litres1~milk\_tlu, data=obj1\_data, index=c("household\_id", "days\_recruitment"), effect = "individual", model="within")  
  
within\_model1 <- plm(total\_litres1~study\_arm+milk\_tlu+nonmilk\_tlu+fdscr\_project1+sick\_tlu+cshtrnsfrprog1+no\_births+cshtrnsfrprog1+hhpple, data=obj1\_data, index=c("household\_id", "days\_recruitment"), effect = "individual", model="within")  
  
within\_model1a <- plm(total\_litres1~tlu+fdscr\_project1+enhanced\_counseling1+sick\_tlu+cshtrnsfrprog1+no\_births+cshtrnsfrprog1+hhpple, data=obj1\_data, index=c("household\_id", "days\_recruitment"), effect = "individual", model="within")  
  
  
pooled\_model1 <- plm(total\_litres~study\_arm, data=obj1\_data, index=c("household\_id", "days\_recruitment"), effect = "individual", model="pooling")  
pooled\_model1a <- plm(total\_litres1~h\_hoccuptn1, data=obj1\_data, index=c("household\_id", "days\_recruitment"), effect = "individual", model="pooling")  
  
  
obj1\_data$fdscr\_project2<-ifelse(obj1\_data$fdscr\_project1%in%"0", "Yes", "No")  
##   
milk\_tlumodel<- plm(milk\_tlu1~fdscr\_project2, data=obj1\_data, index=c("household\_id", "days\_recruitment"), effect = "individual", model="within")  
  
## arm 1 and 2  
  
arm12\_obj1<- obj1\_data|>  
 filter(study\_arm%in%c("Intervention arm 1", "Intervention arm 2"))  
  
  
arm12\_model1<- plm(total\_litres1~enhanced\_counseling1, data=arm12\_obj1, index=c("household\_id", "days\_recruitment"), effect = "individual", model="pooling")   
#feed\_data1<- objective1\_data|>  
# dplyr::select(days\_recruitment, fdscr\_other)|>  
# group\_by(days\_recruitment, fdscr\_other)|>  
# count()|>  
# ungroup()|>  
# filter(fdscr\_other%in%"Yes")  
   
#feed\_data2<- objective1\_data|>  
# dplyr::select(days\_recruitment, fdscr\_project)|>  
# group\_by(days\_recruitment, fdscr\_project)|>  
# count()|>  
# ungroup()|>  
# filter(fdscr\_project%in%"Yes")  
  
  
  
#ggplot()+geom\_line(data=feed\_data1, aes(x=days\_recruitment, y=n, group=1), color="Red", size=1)+theme\_bw()+geom\_line(data=feed\_data2, aes(x=days\_recruitment, y=n, group=1), color="blue")  
  
  
#all\_anim<- obj1\_data|>  
# dplyr::select(household\_id, days\_recruitment, total\_litres\_anim1,fdscr\_project,animals\_ltrs)|>  
# mutate(fdscr\_project=ifelse(fdscr\_project%in%"Yes", 1, 0))|>  
# mutate\_at(vars(total\_litres\_anim1, fdscr\_project), funs(ifelse(is.na(.), 0, .)))|>  
# group\_by(household\_id, days\_recruitment, animals\_ltrs)|>  
 # mutate\_at(vars(total\_litres\_anim1, fdscr\_project), funs(max(., na.rm=T)))|>  
 # ungroup()|>  
 # distinct()  
  
# al<- all\_anim|>  
# group\_by(household\_id, days\_recruitment)|>  
# count()  
  
  
# camels<- all\_anim|>  
# filter(animals\_ltrs%in%"camels\_litres")|>  
# distinct()  
  
# sheep<- all\_anim|>  
# filter(animals\_ltrs%in%"sheep\_litres")|>  
# distinct()  
  
# goats<- all\_anim|>  
# filter(animals\_ltrs%in%"goats\_litres")|>  
# distinct()  
  
# cows<- all\_anim|>  
# filter(animals\_ltrs%in%"cows\_litres")|>  
# distinct()  
# ## species models  
# all\_anim  
# cattle\_model<- plm(total\_litres\_anim1~fdscr\_project+animals\_ltrs, data=all\_anim, index=c("household\_id", "days\_recruitment"), effect = "individual", model="pooling")  
  
  
# sheep\_model<- plm(total\_litres\_anim1~fdscr\_project, data=sheep, index=c("household\_id", "days\_recruitment"), effect = "individual", model="pooling")  
#   
# goats\_model<- plm(total\_litres\_anim1~fdscr\_project, data=goats, index=c("household\_id", "days\_recruitment"), effect = "individual", model="pooling")  
#   
# camels\_model<- plm(total\_litres\_anim1~fdscr\_project, data=camels, index=c("household\_id", "days\_recruitment"), effect = "individual", model="pooling")