

# Challenge 2

Sarim Rizvi

2025-12-05

## Introduction

This report explores seasonality in farmer questions across Kenya and Uganda, using Wefarm Q&A data. We analyse monthly volumes, topic distributions, and align them with cropping seasons to understand farmer needs and provide actionable insights for stakeholders. All results are generated through R code, which is documented below

## Loading required libraries

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v readr      2.1.5
## v forcats    1.0.0      v stringr   1.5.1
## v ggplot2    3.5.1      v tibble    3.2.1
## v lubridate  1.9.3      v tidyr     1.3.1
## v purrr      1.0.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(arrow)
```

```
##
## Attaching package: 'arrow'
##
## The following object is masked from 'package:lubridate':
##
##     duration
##
## The following object is masked from 'package:utils':
##
##     timestamp
```

```
library(corrgram)
library(data.table)
```

```
##
## Attaching package: 'data.table'
##
## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
##
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
##
## The following object is masked from 'package:purrr':
##
##     transpose
```

```
library(qs)
```

```
## qs 0.27.3. Announcement: https://github.com/qsbase/qs/issues/103
```

```
library(naniar)
library(text2vec)
library(tm)
```

```
## Loading required package: NLP
##
## Attaching package: 'NLP'
##
## The following object is masked from 'package:ggplot2':
##
##     annotate
```

```
library(textstem)
```

```
## Loading required package: koRpus.lang.en
## Loading required package: koRpus
## Loading required package: sylly
## For information on available language packages for 'koRpus', run
##
##     available.koRpus.lang()
##
## and see ?install.koRpus.lang()
##
## Attaching package: 'koRpus'
##
## The following object is masked from 'package:tm':
##
##     readTagged
##
## The following object is masked from 'package:readr':
##
##     tokenize
```

```
library(lstatuning)
library(Matrix)
```

```
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

## Loading the data

The data was pre-saved as RDS file to avoid reloading the CSV file

```
# agridata <- fread("data/agridata.csv", header = T) # Uncomment and run to load csv file.
agridata <- readRDS("agridata.rds")
str(agridata)
```

```
## Classes 'data.table' and 'data.frame': 20304843 obs. of 24 variables:
## $ question_id : int 3849056 3849061 3849077 3849077 3849077 3849077 3849078 3849082 ...
## $ question_user_id : int 519124 521327 307821 307821 307821 307821 174909 417525 417525 6...
## $ question_language : chr "nyn" "eng" "nyn" "nyn" ...
## $ question_content : chr "E ABA WEFARM OFFICES ZABO NIZISHANGWA NKAHI?" "Q this goes to w...
## $ question_topic : chr "" "" "cattle" "cattle" ...
## $ question_sent : POSIXct, format: "2017-11-22 12:25:03" "2017-11-22 12:25:05" ...
## $ response_id : int 20691011 4334249 3849291 3849291 3849291 3849291 3849334 6410097 ...
## $ response_user_id : int 200868 526113 296187 296187 296187 296187 438108 107087 482985 3...
## $ response_language : chr "nyn" "eng" "nyn" "nyn" ...
## $ response_content : chr "E!23 Omubazi Ni Dudu Cipa'" "Q1 which stage is marleks last vac...
## $ response_topic : chr "" "" "tomato" "cattle" ...
## $ response_sent : POSIXct, format: "2019-01-24 17:54:06" "2018-01-04 08:57:28" ...
## $ question_user_type : chr "farmer" "farmer" "farmer" "farmer" ...
## $ question_user_status : chr "live" "live" "zombie" "zombie" ...
## $ question_user_country_code : chr "ug" "ug" "ug" "ug" ...
## $ question_user_gender : chr "" "" "" "" ...
## $ question_user_dob : IDate, format: NA NA ...
## $ question_user_created_at : POSIXct, format: "2017-11-18 13:09:11" "2017-11-20 11:55:48" ...
## $ response_user_type : chr "farmer" "farmer" "farmer" "farmer" ...
## $ response_user_status : chr "live" "zombie" "zombie" "zombie" ...
## $ response_user_country_code : chr "ug" "ug" "ug" "ug" ...
## $ response_user_gender : chr "" "" "" "" ...
## $ response_user_dob : IDate, format: NA NA ...
## $ response_user_created_at : POSIXct, format: "2017-05-09 09:19:33" "2017-11-22 10:13:03" ...
## - attr(*, ".internal.selfref")=<externalptr>
```

We then filter to English questions and responses, as translation is not part of this report

```
agridata_en <- agridata %>% filter(question_language == "eng", response_language == "eng")
agridata_en <- agridata_en %>% select(-c(question_language, response_language)) # language fields are n...
agridata_en <- as.data.frame(agridata_en)
str(agridata_en)
```

```
## 'data.frame':    11523993 obs. of  22 variables:
## $ question_id      : int  3849061 3849084 3849098 3849100 3849100 3849100 3849100 3849100 3849100 3849100 ...
## $ question_user_id  : int  521327 6642 526375 237506 237506 237506 237506 237506 237506 237506 5444 ...
## $ question_content  : chr   "Q this goes to wefarm. is it possible to get for us market for ...
## $ question_topic    : chr   "" "rabbit" "poultry" "pig" ...
## $ question_sent     : POSIXct, format: "2017-11-22 12:25:05" "2017-11-22 12:25:10" ...
## $ response_id       : int  4334249 3852272 3859675 4263505 3852604 4238099 4263505 3852604 4238099 4263505 4 ...
## $ response_user_id  : int  526113 35690 522795 412335 412335 412335 412335 412335 412335 412335 35 ...
## $ response_content  : chr   "Q1 which stage is marleks last vaccinated" "Q165#Ksh120" "Q5 10 ...
## $ response_topic    : chr   "" "" "" "" ...
## $ response_sent     : POSIXct, format: "2018-01-04 08:57:28" "2017-11-22 15:26:07" ...
## $ question_user_type: chr   "farmer" "farmer" "farmer" "farmer" ...
## $ question_user_status: chr   "live" "destroyed" "zombie" "destroyed" ...
## $ question_user_country_code: chr   "ug" "ke" "ug" "ke" ...
## $ question_user_gender: chr   "" "" "" "" ...
## $ question_user_dob  : IDate, format: NA NA ...
## $ question_user_created_at : POSIXct, format: "2017-11-20 11:55:48" "2015-07-28 17:12:04" ...
## $ response_user_type  : chr   "farmer" "farmer" "farmer" "farmer" ...
## $ response_user_status: chr   "zombie" "zombie" "zombie" "destroyed" ...
## $ response_user_country_code: chr   "ug" "ke" "ug" "ke" ...
## $ response_user_gender: chr   "" "" "" "" ...
## $ response_user_dob   : IDate, format: NA NA ...
## $ response_user_created_at : POSIXct, format: "2017-11-22 10:13:03" "2015-11-14 19:59:19" ...
```

## Basic Data Understanding

Summary statistics to get a better understanding of the data.

```
summary(agridata_en)
```

##	question_id	question_user_id	question_content	question_topic
##	Min. : 3849061	Min. : 7	Length:11523993	Length:11523993
##	1st Qu.:14091001	1st Qu.: 868085	Class :character	Class :character
##	Median :23569512	Median :1334506	Mode :character	Mode :character
##	Mean :27206212	Mean :1558422		
##	3rd Qu.:40417956	3rd Qu.:2214034		
##	Max. :59261512	Max. :3832740		
##				
##	question_sent	response_id	response_user_id	
##	Min. :2017-11-22 12:25:05.00	Min. : 3849209	Min. : 7	
##	1st Qu.:2018-10-26 16:05:16.67	1st Qu.:14254353	1st Qu.: 643085	
##	Median :2019-04-05 05:04:46.75	Median :23790767	Median :1170037	
##	Mean :2019-07-17 09:07:35.03	Mean :27396260	Mean :1346732	
##	3rd Qu.:2020-04-16 13:35:12.61	3rd Qu.:40628874	3rd Qu.:1940866	
##	Max. :2022-06-21 14:31:25.47	Max. :59262191	Max. :3832167	
##				
##	response_content	response_topic	response_sent	
##	Length:11523993	Length:11523993	Min. :2017-11-22 12:28:03.00	
##	Class :character	Class :character	1st Qu.:2018-10-29 11:43:55.66	
##	Mode :character	Mode :character	Median :2019-04-09 17:58:41.34	
##			Mean :2019-07-21 12:18:57.20	
##			3rd Qu.:2020-04-19 17:04:10.94	
##			Max. :2022-07-04 14:48:23.90	

```
##
## question_user_type question_user_status question_user_country_code
## Length:11523993      Length:11523993      Length:11523993
## Class :character      Class :character      Class :character
## Mode :character      Mode :character      Mode :character
##
##
##
## question_user_gender question_user_dob      question_user_created_at
## Length:11523993      Min. :1917-01-13      Min. :2014-11-27 15:06:11.00
## Class :character      1st Qu.:1976-02-08      1st Qu.:2018-05-23 16:10:11.71
## Mode :character      Median :1988-01-19      Median :2018-10-11 17:32:26.93
##                      Mean :1984-02-25      Mean :2018-11-18 18:59:58.66
##                      3rd Qu.:1995-03-19      3rd Qu.:2019-06-20 18:01:06.51
##                      Max. :2020-09-23      Max. :2022-04-06 23:16:01.57
##                      NA's :10669761
## response_user_type response_user_status response_user_country_code
## Length:11523993      Length:11523993      Length:11523993
## Class :character      Class :character      Class :character
## Mode :character      Mode :character      Mode :character
##
##
##
## response_user_gender response_user_dob      response_user_created_at
## Length:11523993      Min. :1916-02-07      Min. :2014-11-27 15:06:11.00
## Class :character      1st Qu.:1974-01-19      1st Qu.:2018-01-26 07:41:32.00
## Mode :character      Median :1986-07-15      Median :2018-09-05 13:20:20.26
##                      Mean :1982-12-24      Mean :2018-08-21 17:14:31.65
##                      3rd Qu.:1993-11-20      3rd Qu.:2019-03-06 17:38:40.09
##                      Max. :2020-09-23      Max. :2022-04-06 04:48:37.66
##                      NA's :10325876
```

## Top Questions

The most frequently asked questions

```
n_distinct(agridata_en$question_id)/nrow(agridata_en) # Only 25% are different questions
```

```
## [1] 0.2528083
```

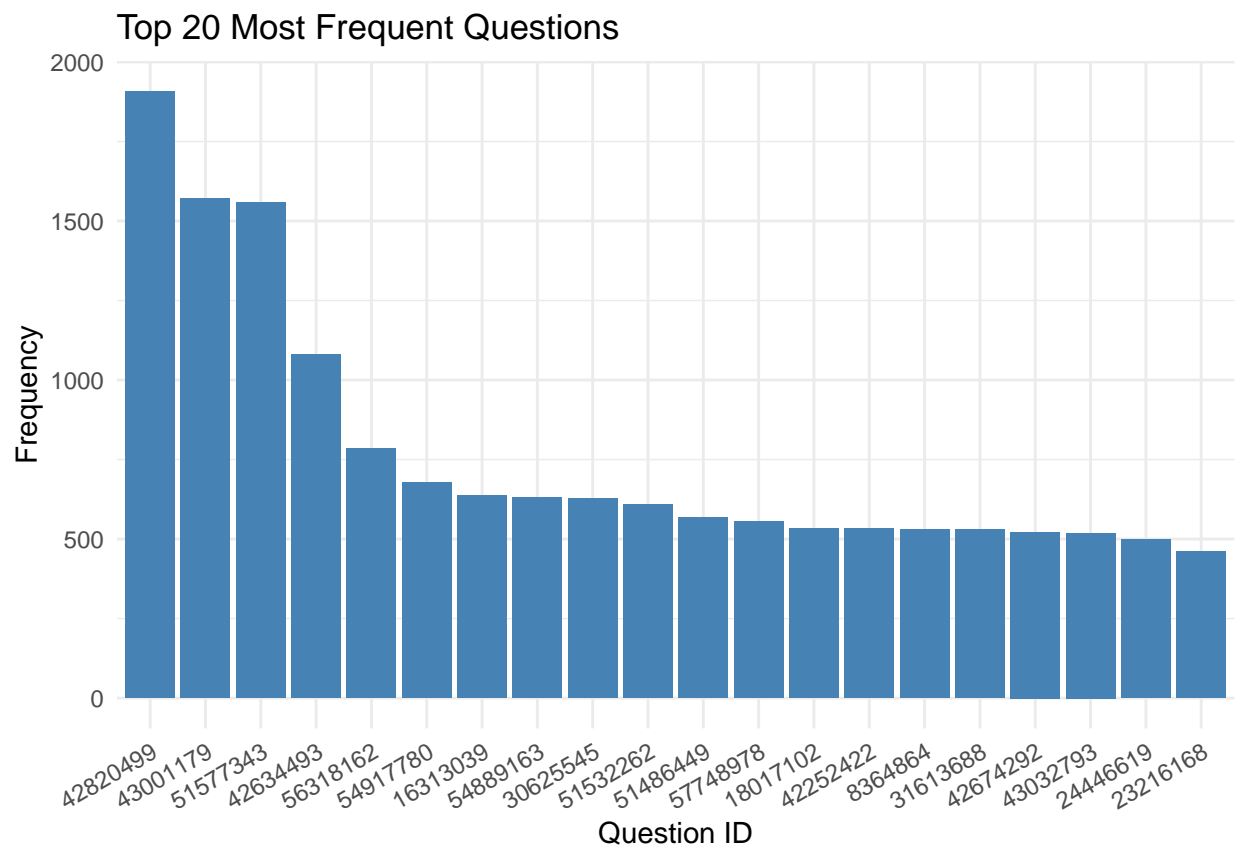
```
top_20_questions <- sort(table(as.factor(agridata_en$question_id)), decreasing=T)[1:20] # 4 questions h
```

```
df_plot <- data.frame(
  question_id = names(top_20_questions),
  count = as.numeric(top_20_questions)
)
```

```
# Convert question_id to a factor to ensure the order is maintained
```

```
df_plot$question_id <- factor(df_plot$question_id, levels = df_plot$question_id)
```

```
# Generate the plot
ggplot(df_plot, aes(x = question_id, y = count)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  labs(
    title = "Top 20 Most Frequent Questions",
    x = "Question ID",
    y = "Frequency"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 30, hjust = 1))
```



```
# Frequency distribution of questions
question_frequencies <- agridata_en %>%
  count(question_id, name = "count")

plot <- ggplot(question_frequencies, aes(x = count)) +
  geom_histogram(binwidth = 10, fill = "steelblue", color = "white") +

  # Using log scale to handle large variation in question count
  scale_y_log10(
    labels = scales::comma
  ) +
  scale_x_continuous(limits = c(1, NA)) +
  labs(
    title = "Log-Scale Distribution of Question Counts",
```

```

    subtitle = "Histogram of Question Frequencies (Binwidth = 10)",
    x = "Number of Times a Question was Asked (Count)",
    y = "Number of Unique Questions (Log Scale)"
  ) +
  theme_minimal() +
  theme(plot.title = element_text(face = "bold"))
plot

```

```

## Warning in scale_y_log10(labels = scales::comma): log-10 transformation
## introduced infinite values.

```

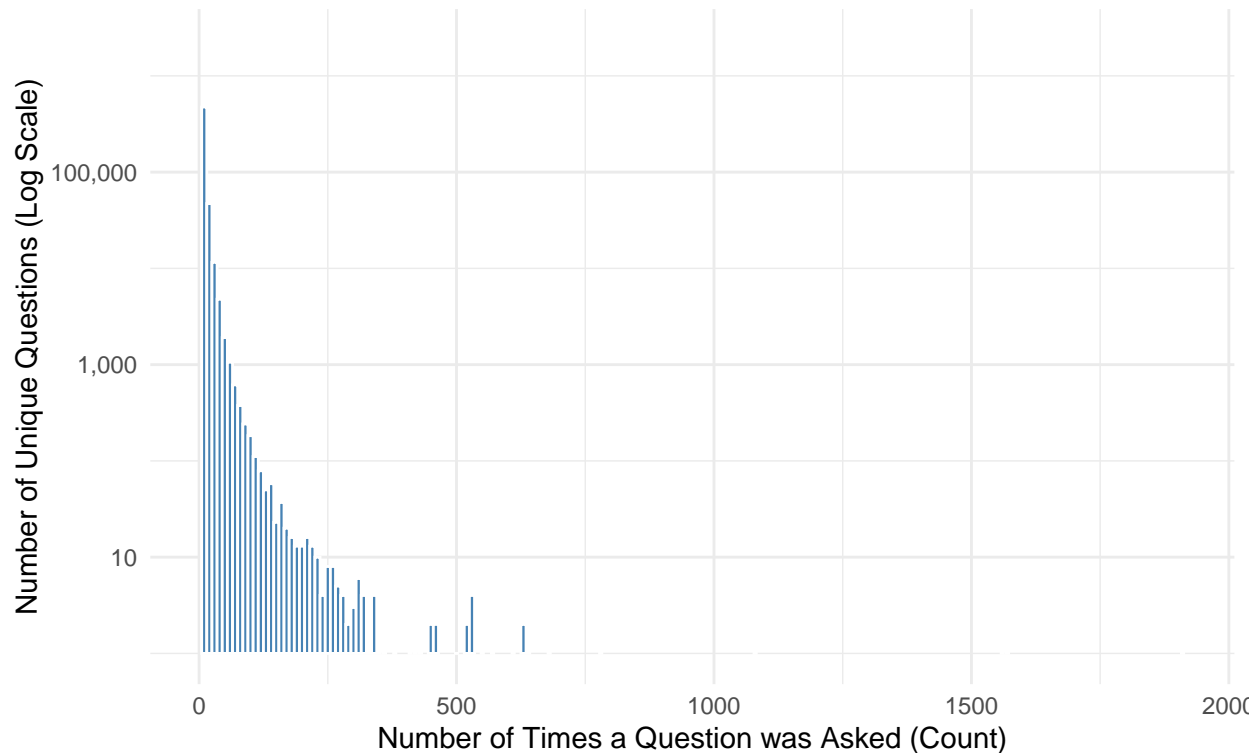
```

## Warning: Removed 136 rows containing missing values or values outside the scale range
## ('geom_bar()').

```

## Log-Scale Distribution of Question Counts

Histogram of Question Frequencies (Binwidth = 10)



Insight: A small number of questions are repeated thousands of times, while most are asked only once.

## Top Question Topics

```

agridata_en$question_topic<-factor(agridata_en$question_topic)

top_20_question_topics <- sort(table(agridata_en$question_topic), decreasing=T)[2:21] # 4 questions hav

df_plot <- data.frame(

```

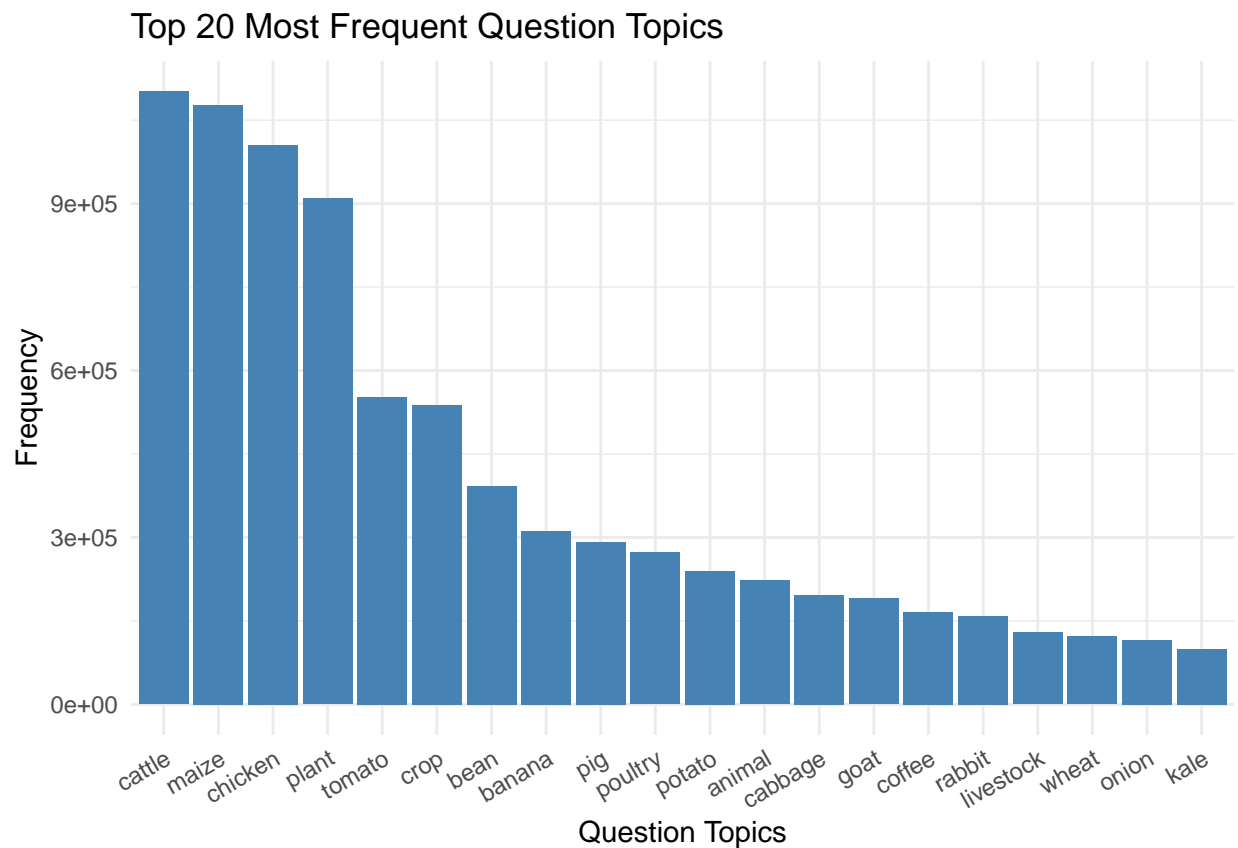
```

question_topic = names(top_20_question_topics),
count = as.numeric(top_20_question_topics)
)

df_plot$question_topic <- factor(df_plot$question_topic, levels = df_plot$question_topic)

# Generate the plot
ggplot(df_plot, aes(x = question_topic, y = count)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  labs(
    title = "Top 20 Most Frequent Question Topics",
    x = "Question Topics",
    y = "Frequency"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 30, hjust = 1))

```



Insight: Cattle, maize, and chicken dominate farmer queries, reflecting key agricultural activities.

## Questions Vs responses

Comparing daily counts of questions and responses.

```

# Create daily counts for questions
question_daily <- agridata_en %>%

```

```

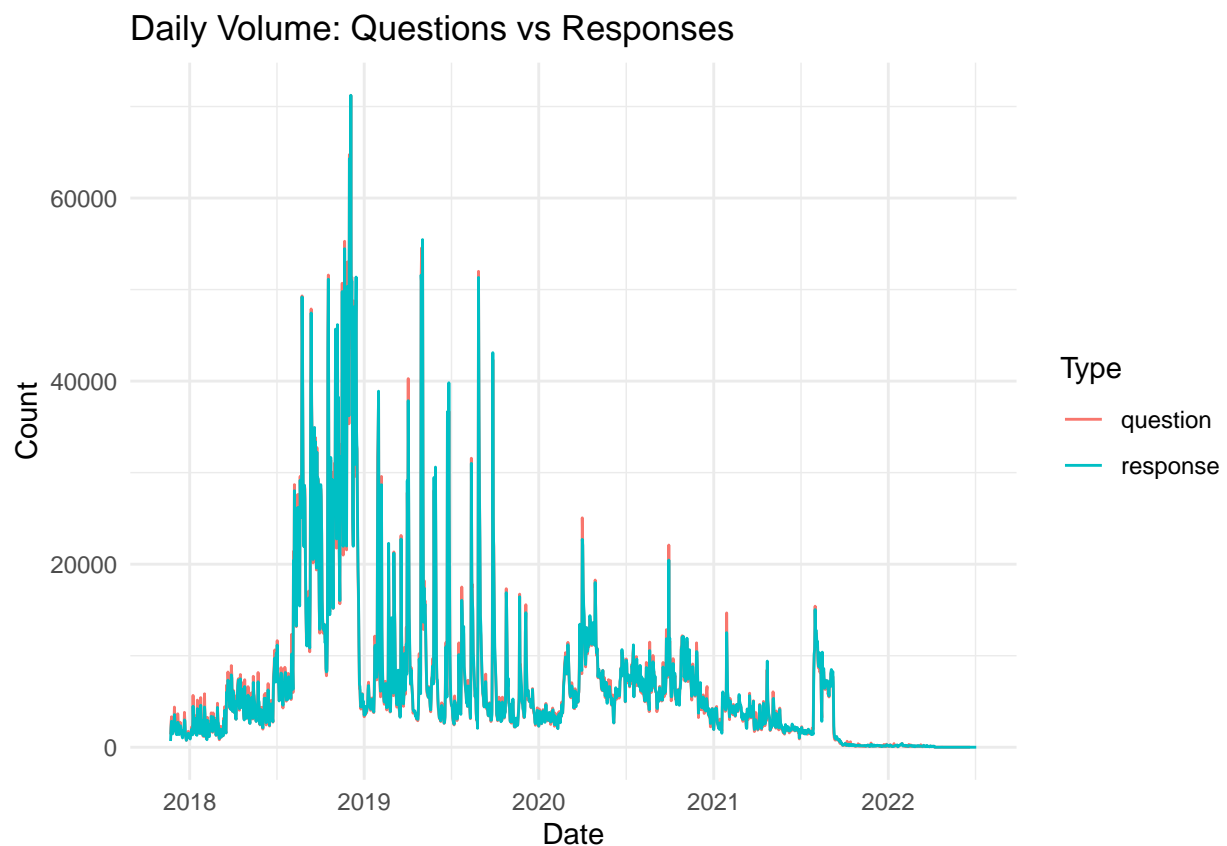
mutate(date = as.Date(question_sent)) %>%
count(date) %>%
mutate(type = "question")

# Create daily counts for responses
response_daily <- agridata_en %>%
  mutate(date = as.Date(response_sent)) %>%
  count(date) %>%
  mutate(type = "response")

# Combine both
daily_combined <- bind_rows(question_daily, response_daily)

# Plot
ggplot(daily_combined, aes(x = date, y = n, color = type)) +
  geom_line() +
  labs(title = "Daily Volume: Questions vs Responses",
       x = "Date", y = "Count", color = "Type") +
  theme_minimal()

```



Insight: Responses generally track question volume, but lag slightly in timing.

## Handling Missing data

Replacing empty values with NA for easier tracking of missing values

```

agridata_en <- agridata_en %>%
  mutate(across(where(is.factor), as.character)) %>% # Convert factors to character
  mutate(across(where(is.character), ~na_if(.x, ""))) # Replace "" with NA

miss_var_summary(agridata_en)

```

```

## # A tibble: 22 x 3
##   variable          n_miss pct_miss
##   <chr>             <int>   <num>
## 1 question_user_gender 11063776    96.0
## 2 response_user_gender 10744844    93.2
## 3 question_user_dob    10669761    92.6
## 4 response_user_dob    10325876    89.6
## 5 response_topic       7527577    65.3
## 6 question_topic      1548673    13.4
## 7 question_id           0         0
## 8 question_user_id      0         0
## 9 question_content      0         0
## 10 question_sent        0         0
## # i 12 more rows

```

Insight: Most columns have complete information

The following columns were removed: question\_user\_gender, response\_user\_gender, question\_user\_dob, response\_user\_dob, question\_user\_created\_at, response\_user\_created\_at, response\_user\_status and question\_user\_status. These have high % of missing values or are not relevant for the analysis.

```

## Removing gender and dob due to high missing%
agridata_en <- agridata_en %>%
  select(-c(question_user_gender, response_user_gender,
            question_user_dob, response_user_dob))

## Removing User creation date - unnecessary information
agridata_en$question_user_created_at <- NULL
agridata_en$response_user_created_at <- NULL
agridata_en$response_user_status <- NULL
agridata_en$question_user_status <- NULL

```

## Feature Engineering

Converting country code to factor as there are only four possible values.

```

agridata_en$question_user_country_code <- factor(agridata_en$question_user_country_code)
agridata_en$response_user_country_code <- factor(agridata_en$response_user_country_code)

```

Breaking down date and time of question and responses into month and year to track seasonality

```

## Breaking down Question and response times
str(agridata_en$question_sent)

```

```

## POSIXct[1:11523993], format: "2017-11-22 12:25:05" "2017-11-22 12:25:10" "2017-11-22 12:25:12" ...

```

```

agridata_en$question_sent_year <- year(agridata_en$question_sent)
agridata_en$question_sent_month <- month(agridata_en$question_sent)
agridata_en$response_sent_year <- year(agridata_en$response_sent)
agridata_en$response_sent_month <- month(agridata_en$response_sent)

## Removing question_sent and response_sent
agridata_en$response_sent <- NULL
agridata_en$question_sent <- NULL

```

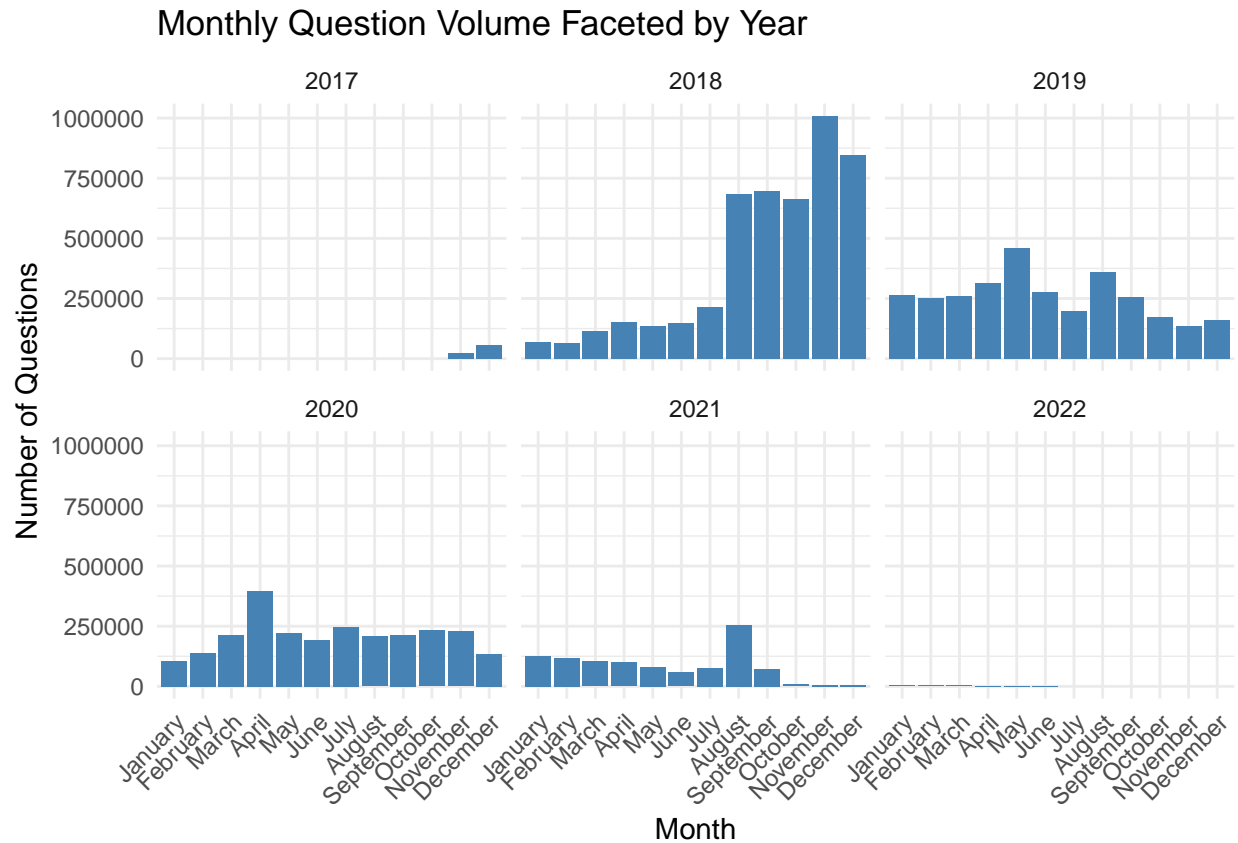
## Yearly question distribution

```

## Plot with faceting by year
monthly_by_year <- agridata_en %>%
  group_by(question_sent_year, question_sent_month) %>%
  summarise(total_questions = n(), .groups = "drop") %>%
  mutate(month_label = factor(month.name[question_sent_month], levels = month.name))

ggplot(monthly_by_year, aes(x = month_label, y = total_questions)) +
  geom_col(fill = "steelblue") +
  facet_wrap(~ question_sent_year, ncol = 3) +
  labs(
    title = "Monthly Question Volume Faceted by Year",
    x = "Month",
    y = "Number of Questions"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

```



Insight: The dataset has the most questions in 2018. 2017 and 2022 have limited coverage over the year.

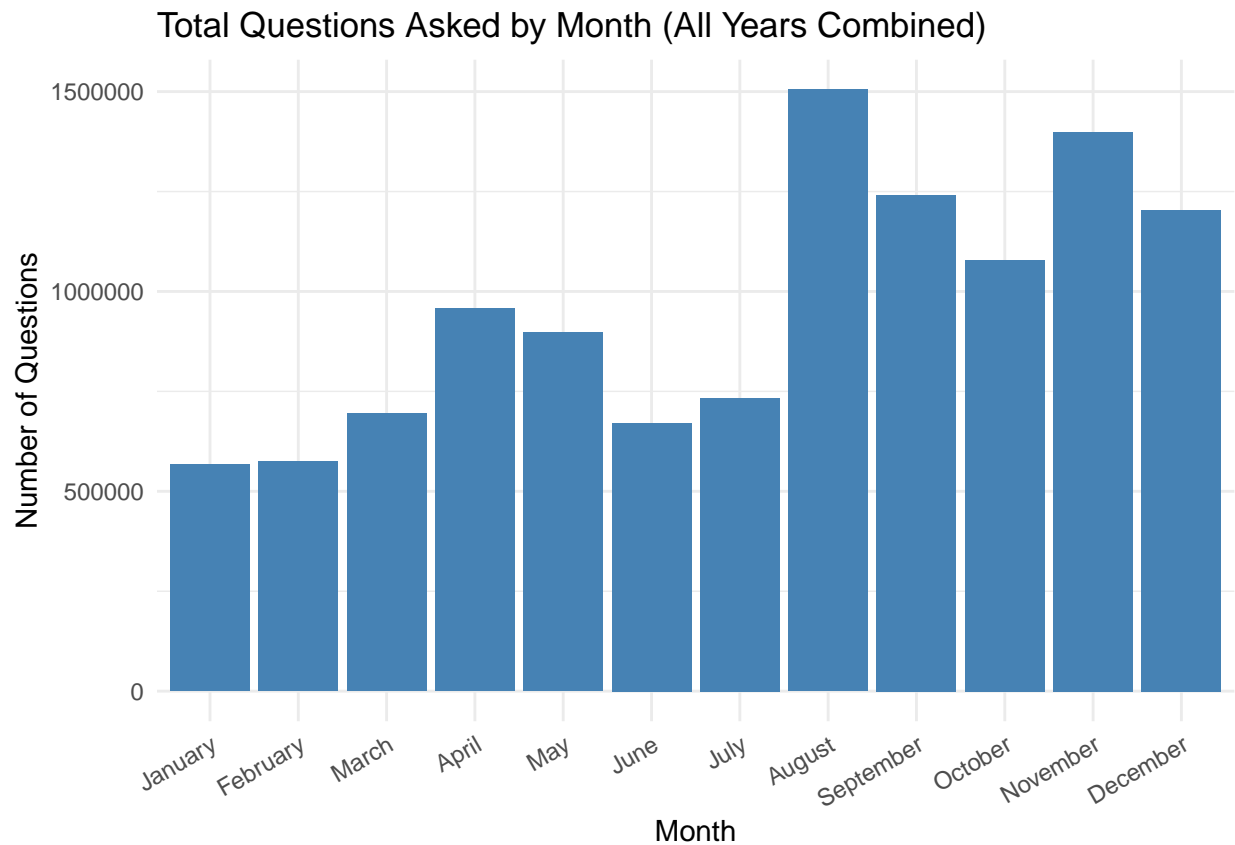
## Monthly Seasonality

Examining monthly totals across all years.

```
monthly_totals <- agridata_en %>%
  group_by(question_sent_month) %>%
  summarise(total_questions = n(), .groups = "drop")

monthly_totals$month_label <- month.name[monthly_totals$question_sent_month]

ggplot(monthly_totals, aes(x = reorder(month_label, question_sent_month), y = total_questions)) +
  geom_col(fill = "steelblue") +
  labs(
    title = "Total Questions Asked by Month (All Years Combined)",
    x = "Month",
    y = "Number of Questions"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 30, hjust = 1))
```



Insight: Peaks in March–May and October–December align with planting seasons.

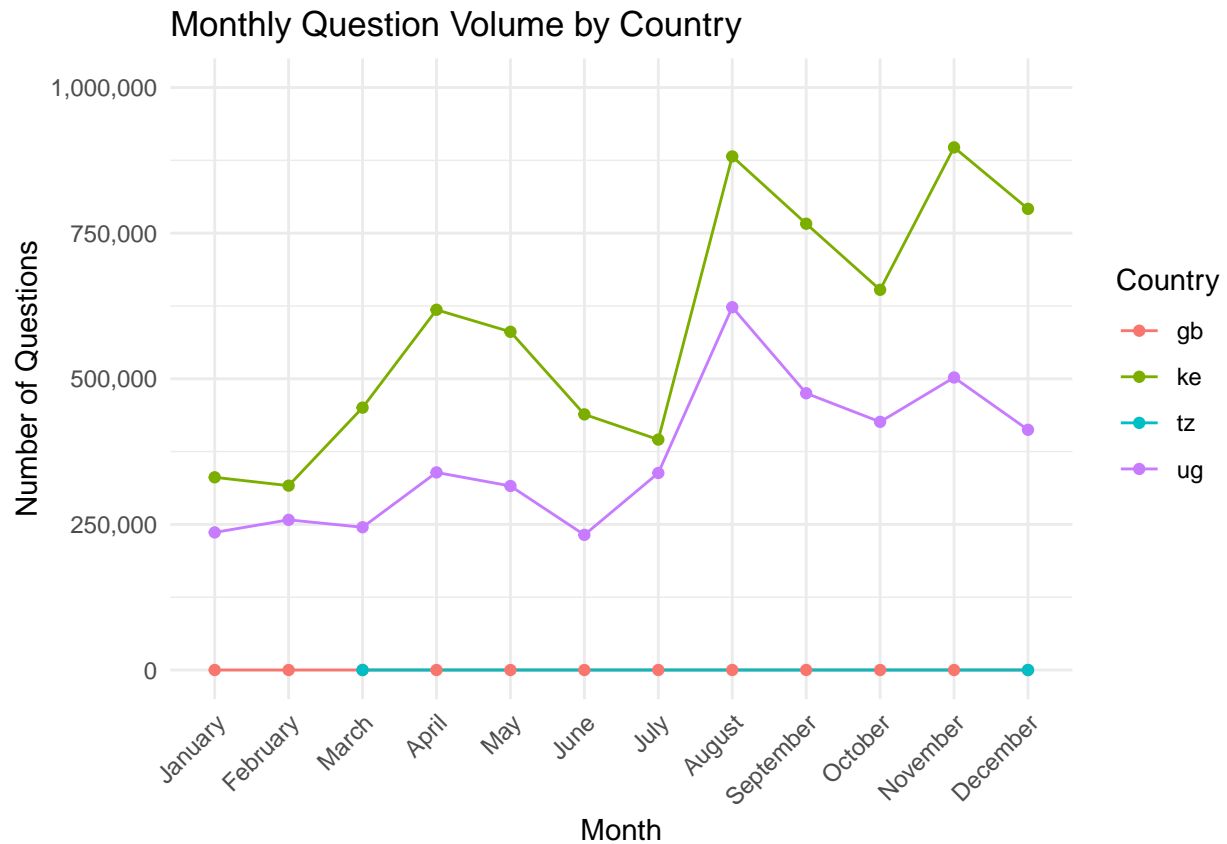
## Country comparison

Comparing seasonal trends across countries

```
monthly_by_country <- agridata_en %>%
  group_by(question_user_country_code, question_sent_month) %>%
  summarise(total_questions = n(), .groups = "drop") %>%
  mutate(month_label = factor(month.name[question_sent_month], levels = month.name))

ggplot(monthly_by_country,
  aes(x = month_label, y = total_questions, color = question_user_country_code, group = question_user_country_code)) +
  geom_line() +
  geom_point() +
  scale_y_continuous(
    limits = c(0, 1000000),
    labels = scales::comma
  ) +
  labs(
    title = "Monthly Question Volume by Country",
    x = "Month",
    y = "Number of Questions",
    color = "Country"
  ) +
```

```
theme_minimal() +
theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Insight: Kenya and Uganda show similar seasonal peaks, but Uganda's Season B creates a second wave.

## Topic Modelling (LDA)

Applying LDA to discover topics in farmer questions.

```
## Creating a new dataframe with just the question ID and question content
lda_input <- agridata_en %>%
  select(question_id, question_content) %>%
  distinct(question_id, .keep_all = TRUE) %>%
  filter(!is.na(question_content))
str(lda_input)
```

```
## 'data.frame': 2913361 obs. of 2 variables:
## $ question_id : int 3849061 3849084 3849098 3849100 3849129 3849153 3849196 3849225 3849246 3849268
## $ question_content: chr "Q this goes to wefarm. is it possible to get for us market for our produc"
```

Preprocessing the data by removing custom words, punctuation and numbers. Also converted data to lower case for easier processing. The data was then lemmatised to get the root words.

```
## Data Preprocessing
custom_stopwords <- c(stopwords("en"), "q", "question", "what", "where", "why", "how", "when", "who", "I")

prep_fun <- function(text) {
  text %>%
    tolower() %>%
    removePunctuation() %>%
    str_replace_all("^q\\s*-*", "") %>%
    removeNumbers() %>%
    removeWords(custom_stopwords) %>%
    stripWhitespace() %>%
    lemmatize_strings()
}

lda_input_clean <- lda_input %>%
  mutate(clean_text = prep_fun(question_content))
```

Tokenisation is done to divide the questions into a table, with each column having a word(token)

```
## Tokenisation and DTM
tokens <- word_tokenizer(lda_input_clean$clean_text)

it <- itoken(tokens, progressbar = TRUE)
vocab <- create_vocabulary(it) %>%
  prune_vocabulary(term_count_min = 10, doc_proportion_max = 0.5)
```

Most common words from the questions.

```
vocab %>%
  arrange(desc(term_count)) %>%
  head(50)
```

```
## Number of docs: 2913361
## 0 stopwords: ...
## ngram_min = 1; ngram_max = 1
## Vocabulary:
##      term term_count doc_count
##      <char>      <int>      <int>
## 1:   good    394218    382634
## 2:  plant    333013    315999
## 3:  maize    231031    223261
## 4:   use    222085    216891
## 5:   cow    181106    175009
## 6:   get    159849    157536
## 7:  type    154391    152713
## 8: tomato    136981    133901
## 9:   farm    134102    130325
## 10: grow     123792    120846
## 11: give     119630    115888
## 12: many     113778    112697
## 13: one      112569    106952
## 14: take     109496    108196
```

## 15:	ask	106594	101650
## 16:	much	101565	101052
## 17:	control	98553	97902
## 18:	bean	95345	92810
## 19:	want	95279	93028
## 20:	crop	94735	92209
## 21:	long	90614	89956
## 22:	seed	88532	85631
## 23:	cause	83948	83542
## 24:	banana	82589	77630
## 25:	hen	82534	80109
## 26:	chick	82090	79156
## 27:	farmer	81497	79544
## 28:	disease	79749	77710
## 29:	chicken	78165	75952
## 30:	start	77730	73203
## 31:	reply	73622	72542
## 32:	price	73504	72562
## 33:	follow	73085	72216
## 34:	pig	68144	65926
## 35:	soil	64449	62436
## 36:	spray	64262	62620
## 37:	need	63668	62317
## 38:	response	63567	62998
## 39:	egg	63330	60050
## 40:	know	62492	61810
## 41:	poultry	61845	60666
## 42:	milk	60791	56812
## 43:	will	58883	57546
## 44:	market	58829	57647
## 45:	season	58697	57322
## 46:	month	58633	57593
## 47:	potato	55581	54548
## 48:	goat	55516	53705
## 49:	time	55030	53589
## 50:	animal	54891	53769
##	term	term_count	doc_count

Insight: Vocabulary confirms key terms like “maize”, “cow”, “tomato”, “seed”, “market” dominate farmer queries.

Creating the LDA model

```
vectorizer <- vocab_vectorizer(vocab)
dtm <- create_dtm(it, vectorizer)
```

```
lda_model <- LDA$new(n_topics = 10, doc_topic_prior = 0.1, topic_word_prior = 0.01)
doc_topic_distr <- lda_model$fit_transform(dtm, n_iter = 1000)
```

Topic assignment

```
topic_assignments <- data.frame(
  question_id = lda_input_clean$question_id,
```

```
topic = max.col(doc_topic_distr) # most probable topic per document
)
```

Merging topics to the question and labelling the topic based on the most common words

```
agridata_en <- agridata_en %>%
  left_join(topic_assignments, by = "question_id")

## Top Words and Topic labelling
top_words <- lda_model$get_top_words(n = 10, lambda = 1)

topic_labels <- c(
  "Crop cultivation basics",      # Topic 1
  "Poultry farming",             # Topic 2
  "Crop management & harvest",   # Topic 3
  "Starting a farm/business",     # Topic 4
  "Market & pricing",            # Topic 5
  "Crop protection",             # Topic 6
  "Other livestock & community", # Topic 7
  "Platform interactions",       # Topic 8
  "Banana/soil management",      # Topic 9
  "Livestock & dairy"            # Topic 10
)

agridata_en$topic_label <- topic_labels[agridata_en$topic]
table(agridata_en$topic_label)
```

```
##
##      Banana/soil management      Crop cultivation basics
##              1365187              764179
##      Crop management & harvest      Crop protection
##              1061717              1191202
##              Livestock & dairy      Market & pricing
##              786236                884170
##      Other livestock & community      Platform interactions
##              1158029              1750512
##              Poultry farming      Starting a farm/business
##              1286920              1275841
```

```
topic_month_counts <- agridata_en %>%
  filter(!is.na(topic_label), !is.na(question_sent_month)) %>%
  group_by(question_sent_year, question_sent_month, topic_label) %>%
  summarise(count = n(), .groups = "drop") %>%
  mutate(month_label = factor(month.name[question_sent_month], levels = month.name))
```

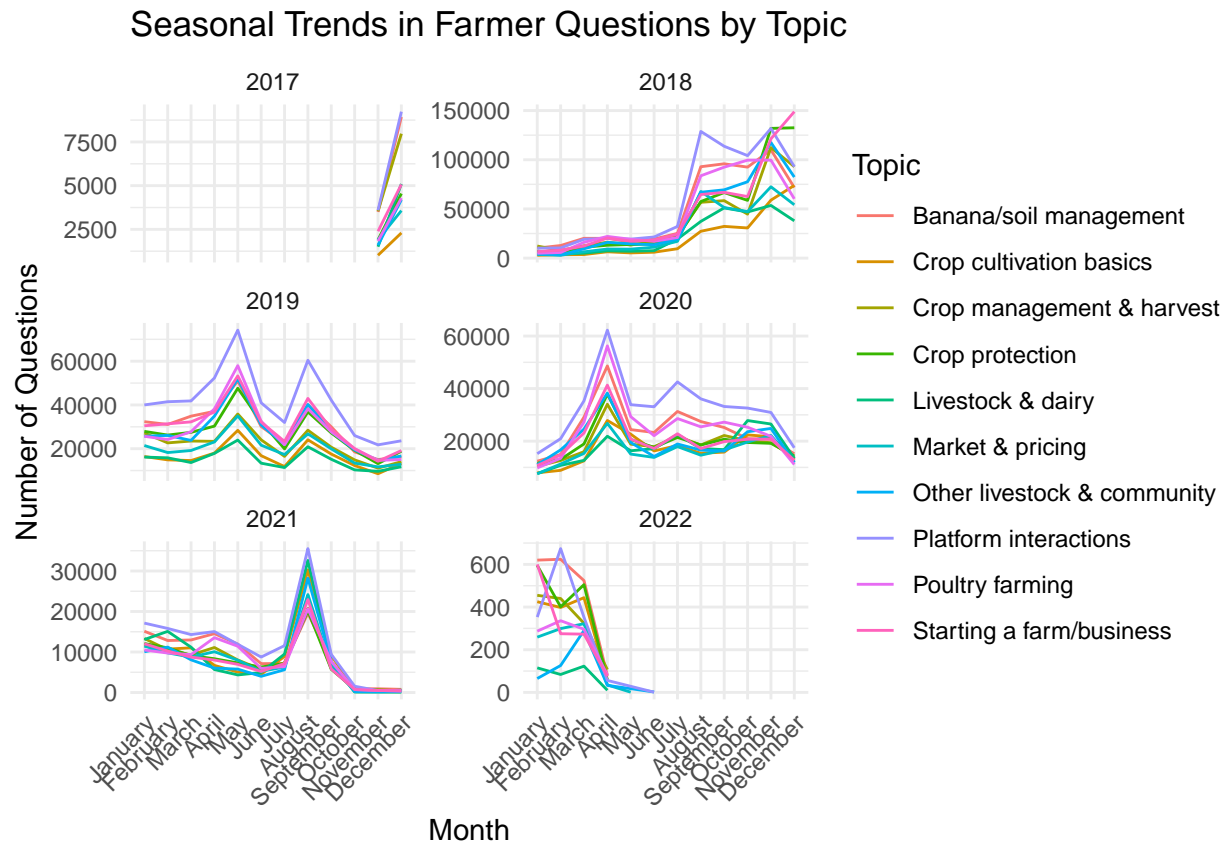
Plot showing the topic trend throughout the year

```
ggplot(topic_month_counts,
  aes(x = month_label, y = count,
    color = topic_label,
    group = topic_label)) +
  geom_line() +
```

```

facet_wrap(~question_sent_year, scales = "free_y", ncol = 2) +
labs(
  title = "Seasonal Trends in Farmer Questions by Topic",
  x = "Month",
  y = "Number of Questions",
  color = "Topic"
) +
theme_minimal() +
theme(axis.text.x = element_text(angle = 45, hjust = 1))

```



Country-wise season

```

season_map <- data.frame(
  country = c(rep("ke", 12), rep("ug", 12)),
  month = rep(1:12, 2),
  season_phase = c(
    # Kenya
    "Harvest", "Harvest", "Planting", "Planting", "Planting",
    "Harvest", "Harvest", "Harvest", "Dry",
    "Planting", "Planting", "Planting",
    # Uganda
    "Harvest", "Harvest", "Planting", "Planting", "Planting",
    "Harvest", "Harvest", "Harvest", "Planting",
    "Planting", "Planting", "Harvest"
  )
)

```

```

topic_month_counts <- agridata_en %>%
  filter(!is.na(topic_label), !is.na(question_sent_month), !is.na(question_user_country_code)) %>%
  group_by(question_user_country_code, question_sent_year, question_sent_month, topic_label) %>%
  summarise(count = n(), .groups = "drop")

topic_season_counts <- topic_month_counts %>%
  left_join(season_map,
            by = c("question_user_country_code" = "country",
                  "question_sent_month" = "month"))

```

Comparison of topic across the year in Kenya and Uganda

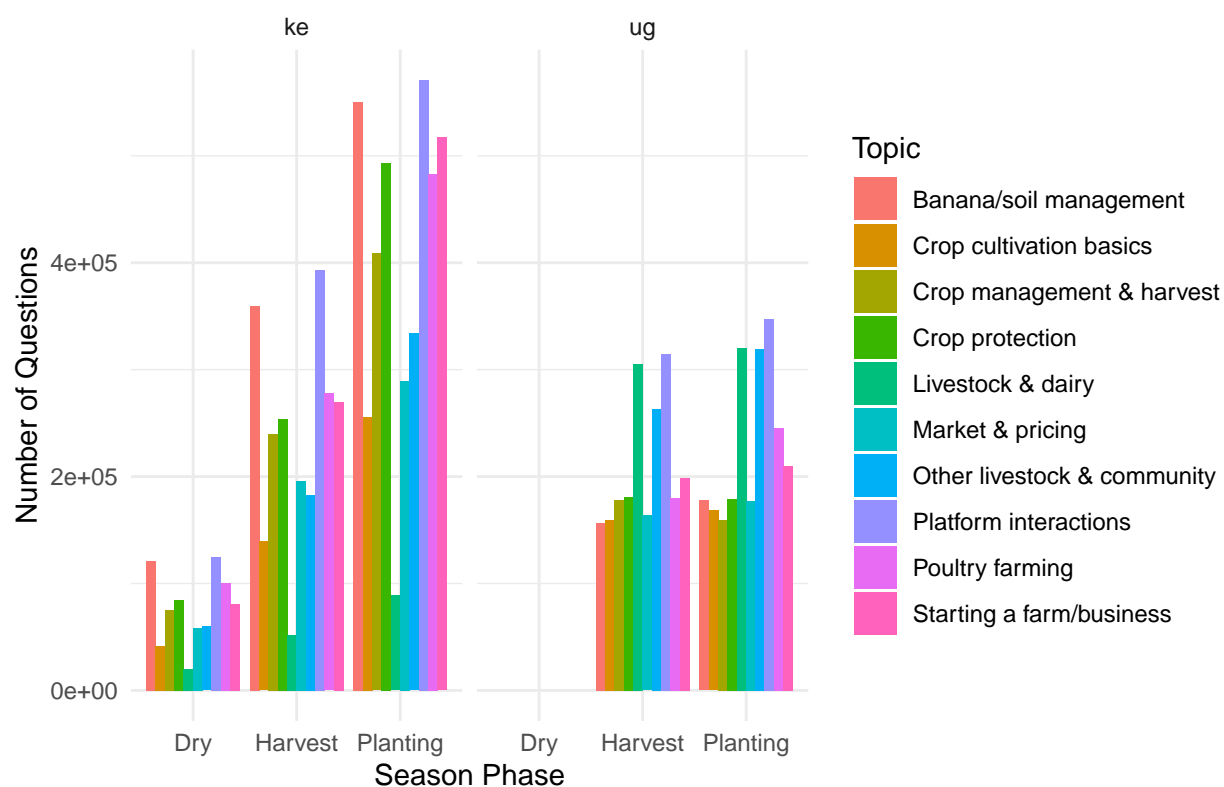
```

seasonal_topics <- topic_season_counts %>% filter(question_user_country_code == 'ke' | question_user_coun
  group_by(question_user_country_code, season_phase, topic_label) %>%
  summarise(total_questions = sum(count), .groups = "drop")

ggplot(seasonal_topics,
       aes(x = season_phase, y = total_questions,
           fill = topic_label)) +
  geom_col(position = "dodge") +
  facet_wrap(~ question_user_country_code) +
  labs(
    title = "Topic Distribution by Cropping Season and Country",
    x = "Season Phase",
    y = "Number of Questions",
    fill = "Topic"
  ) +
  theme_minimal()

```

## Topic Distribution by Cropping Season and Country



## Key Insights

- Farmer engagement is seasonal, peaking in planting and harvest months.
- Kenya: Two peaks (long rains Mar–May, short rains Oct–Dec).
- Uganda: Two cropping seasons (Season A Mar–May, Season B Sep–Nov).
- Topics:
  - Planting → seed varieties, pest control, soil fertility.
  - Harvest → storage, pricing, market access.
  - Livestock/poultry → steady year-round.
- Implication: Support services (advice, market info, pest alerts) should be timed to cropping calendars.

## Conclusion

Seasonality strongly shapes farmer information needs. By aligning support and interventions with planting and harvest cycles, Wefarm and partners can maximize impact.