

Statistical analysis for ‘Comparison of wild mushroom use by ethnic groups surrounding the Upper Ouémé Reserve Forest in Benin, West Africa.’

anonymous for submission

This R Markdown notebook includes code and output for all statistical analyses mentioned in the main text, including all main and supplementary figures. The lists of figures and tables are hyperlinked for easy access to the supplemental figures and tables.

Data inputs for this notebook are prepared from raw data files by `1_data.Rmd`.

Contents

Setup	4
0.1 Symbolology	4
0.2 Species abbreviations	6
1 Vernacular names	6
1.1 Read data	6
1.2 Name stats	6
1.2.1 Number of vernacular names	6
1.2.2 Number of vernacular names per language	7
1.3 Name categories	8
1.4 Name characteristics	10
2 Interview stats	13
2.1 Load data	13
2.2 Number of interviews	14
2.3 Number of mushroom citations	14
2.4 Mean and median number of citations per interviewee	14
2.5 Number of specimens cited in interviews	14
2.6 Number of specimens cited in focus groups	14
2.7 Number of specimens cited in focus groups and interviews combined	14
2.7.1 Number of species represented by cited specimens	15
2.8 Number of cited “sp.” specimens	15

2.8.1	Number of “sp.” species represented by cited specimens	15
2.9	Number of cited “cf.” specimens	15
2.9.1	Number of “cf.” species represented by cited specimens	15
2.10	Number of “nom. prov.” specimens	15
2.10.1	Number of “nom. prov.” species represented by cited specimens	16
2.11	Number of specimens confidently assigned to a species:	16
2.11.1	Number of confidently assigned species represented by cited specimens:	16
3	Number of recognized species	16
3.1	Load data	16
3.2	Sampling balance	16
3.2.1	Number of residents interviewed from each ethnic group and village	16
3.2.2	Number of residents interviewed from each gender and village	17
3.2.3	Number of residents interviewed from each gender and ethnic group	17
3.2.4	Number of residents interviewed from each gender and age	18
3.2.5	Number of residents interviewed from each age and ethnic group	18
3.2.6	Number of residents interviewed from each age and village	19
3.2.7	Number of residents interviewed from each gender, ethnic group, and village	19
3.2.8	VIF	21
3.3	Generalized linear model (GLM)	22
3.4	Reduced GLM for comparison of ethnic groups	24
3.5	Marginal effects	26
3.6	Model predictions	34
4	“Community” composition analysis	37
4.1	Load data	37
4.2	Knowledge CCA	37
4.3	Preference CCA	49
5	Knowledge exchange within villages	59
6	Edible species	64
	Package versions	66

List of Tables

A	Number of vernacular names recorded by language.	7
B	Number of names in each language which belong to each semantic category.	8
C	Number of names in each language which belong to each characteristic.	11
D	Number of male and female interviewees from each village and ethnic group. (Table 2 in main text.)	20
E	Variance inflation factors for gender, age, and ethnic group/village combination.	21
F	Coefficients and associated confidence intervals for Poisson GLM.	23
G	Coefficients and confidence intervals for Poisson GLM, fit without village as a predictor. . . .	25
H	Estimated response ratios for Poisson GLM. (Table 5 in main text.)	29
I	Estimated response ratios for Poisson GLM without village effect.	30
J	ANOVA table for CCA axes of knowledge CCA.	39
K	ANOVA table for explanatory terms in knowledge CCA.	39
L	Centroids of predictor values for knowledge CCA	41
M	ANOVA table for CCA axes of preference CCA.	50
N	ANOVA table for explanatory variables in preference CCA.	51
O	Centroids of predictor values for preference CCA	53
P	Species regarded as edible.	65

List of Figures

A	Correspondence Analysis (CA) plot for semantic categories used in mushroom names in different languages.	10
B	Correspondence Analysis (CA) plot for characteristics used in mushroom names in different languages.	13
C	Model checking results for generalized linear model of number of species recognized.	24
D	Model checking results for generalized linear model of number of species recognized, without using village as a variable.	26
E	Estimated marginal effect contrasts based on Poisson GLM. (Fig 2 in main text.)	33
F	Observed data (points) and model predictions (lines) for the Poisson GLM.	36
G	Biplot of CCA axes 1 and 2 for species known. (Fig 3 in main text.)	44
H	Biplot of CCA axes 1 and 3 for species known. (Fig 4 in main text.)	46
I	Biplot of CCA axes 2 and 3 for species known.	48
J	Biplot of CCA axes 1 and 2 for species preferences.	56
K	Biplot of CCA axes 3 and 4 for species preferences.	58
L	Pairwise knowledge CCA differences between Lokpa and Yom respondents in Angaradebou and Bio Sika villages. (Fig 5 in main text.)	61
M	Pairwise preference CCA differences between Lokpa and Yom respondents in Angaradebou and Bio Sika villages.	63

Setup

This section loads data and defines some R functions and settings which will be used later in the document

```
# output figures as PDF, TIFF, and PNG
knitr::opts_chunk$set(dev = if (knitr::is_latex_output()) c("pdf", "tiff", "png") else "png", dpi = 300)
# silence dplyr::summarize's "helpful" messages
options(dplyr.summarise.inform = FALSE)
```

Load libraries.

```
library(magrittr) # %>% and %T>%
library(ggplot2) # plotting
library(vegan) # community ecology
theme_set(theme_bw())
```

0.1 Symbolology

In this section, we define some helper functions that allow us to apply consistent formatting to multiple ggplot2 plots.

Load plotting aesthetics for different villages, ethnic groups, and their combinations.

```
groups.file <- here::here("data", "groups.csv")
groups <- readr::read_csv(groups.file, col_types = "cccc")
villages.file <- here::here("data", "villages.csv")
villages <- readr::read_csv(villages.file, col_types = "ccccii")
villagegroups.file <- here::here("data", "villagegroups.csv")
villagegroups <-
  readr::read_csv(villagegroups.file, col_types = "cc") %>%
  dplyr::left_join(groups, by = c("Group")) %>%
  dplyr::left_join(villages, by = c("Village"), suffix = c("Group", "Village")) %>%
  dplyr::mutate(
    VillageGroup = paste0(Village, Group),
    Label = paste(Village, Group, sep = " / ")
  )
```

Make a ggplot2 object that applies color, shape, and fill to village/group combinations. This is used in main CCA plots.

```
scale_villagegroup <- list(
  scale_shape_manual(
    values = tibble::deframe(
      dplyr::select(villagegroups, VillageGroup, ShapeFill)
    ),
    name = NULL,
    breaks = villagegroups$VillageGroup,
    labels = villagegroups$Label,
    guide = guide_legend(ncol = 4)
  ),
  scale_color_manual(
    values = tibble::deframe(
```

```

    dplyr::select(villagegroups, VillageGroup, Color)
  ),
  name = NULL,
  breaks = villagegroups$VillageGroup,
  labels = villagegroups$Label,
  guide = guide_legend(ncol = 4)
),
scale_fill_manual(
  values = tibble::deframe(
    dplyr::select(villagegroups, VillageGroup, Fill)
  ),
  name = NULL,
  breaks = villagegroups$VillageGroup,
  labels = villagegroups$Label,
  guide = guide_legend(ncol = 4)
)
)

```

A theme which styles the rest of the main CCA plot.

```

theme_cca_main <- list(
  coord_fixed(),
  scale_x_continuous(sec.axis = dup_axis(name = NULL, labels = NULL),
    expand = expansion(mult = c(0.1, 0.1))),
  scale_y_continuous(sec.axis = dup_axis(name = NULL, labels = NULL),
    expand = expansion(mult = c(0.1, 0.1))),
  theme(
    legend.position = "bottom",
    legend.direction = "vertical",
    legend.justification = c(0,0),
    legend.text = element_text(size = 7),
    legend.key.size = unit(5, "mm"),
    legend.margin = margin(0, 3, 0, 0),
    legend.box.margin = margin(0, 0, 0, 0),
    legend.spacing.y = unit(0, "pt"),
    legend.spacing.x = unit(4, "pt"),
    plot.margin = margin(0, 0, 0, 0),
    panel.grid = element_blank()
  )
)

```

Apply just color and shape based on only village. This is for CCA sidebar plots.

```

scale_village <- list(
  scale_shape_manual(
    values = tibble::deframe(dplyr::select(villages, Village, ShapeNofill)),
    breaks = c("Son", "Fab", "Gan", "Ang", "Bio")
  ),
  scale_color_manual(
    values = tibble::deframe(dplyr::select(villages, Village, Color)),
    breaks = c("Son", "Fab", "Gan", "Ang", "Bio")
  )
)

```

A theme which styles the rest of the sidebar CCA plots.

```
theme_cca_side <- list(  
  coord_fixed(),  
  scale_x_continuous(label = NULL, name = NULL,  
                     sec.axis = dup_axis(name = NULL, labels = NULL)),  
  scale_y_continuous(label = NULL, name = NULL,  
                     sec.axis = dup_axis(name = NULL, labels = NULL)),  
  lemon::facet_rep_wrap(~Ethnic.group, ncol = 1, strip.position = "right"),  
  theme(  
    strip.placement = "outside",  
    legend.text = element_text(size = 7),  
    legend.title = element_text(size = 9),  
    legend.key.size = unit(2.5, "mm"),  
    legend.box.spacing = unit(5, "pt"),  
    legend.box.margin = margin(3, 3, 3, 3),  
    legend.margin = margin(0, 0, 0, 0),  
    plot.margin = margin(0, 0, 0, 5),  
    panel.grid = element_blank()  
  )  
)
```

0.2 Species abbreviations

Load species names and abbreviations.

```
species.file <- here::here("data", "species.csv")  
specieskey <- readr::read_csv(species.file, col_types = "cccc")  
abbrevkey <- dplyr::select(specieskey, Abbrev, canonicalName) %>%  
  unique()
```

1 Vernacular names

1.1 Read data

Read the data.

```
name_table <- readRDS(here::here("output", "nametable.rds"))
```

1.2 Name stats

Output counts of names which are mentioned in the text.

1.2.1 Number of vernacular names

```
dplyr::distinct(name_table, Vernacular.name, Language) %>% nrow()
```

```
## [1] 180
```

1.2.2 Number of vernacular names per language

Count the number of distinct vernacular names given in each language (Table A).

```
dplyr::distinct(name_table, Vernacular.name, Language) %>%  
  dplyr::count(Language)
```

Table A: Number of vernacular names recorded by language.

Language	n
Bariba	51
Gando	40
Lokpa	24
Peuhl	9
Yom	56

1.3 Name categories

Make a table of name categories by ethnic group. Only Bariba, Gando, Lokpa, and Yom names are included, because the other language (Peulh) was not well sampled.

```
name_categories <-
  dplyr::group_by(name_table, Language, type) %>%
  dplyr::filter(Language %in% c("Bariba", "Gando", "Lokpa", "Yom")) %>%
  dplyr::tally() %>%
  tidyr::pivot_wider(
    names_from = type,
    values_from = n,
    values_fill = 0L
  )
name_categories
name_categories %<>%
  tibble::column_to_rownames("Language") %>%
  as.matrix()
```

Table B: Number of names in each language which belong to each semantic category.

Language	ecological	physical	practical	unknown
Bariba	14	27	17	6
Gando	9	24	10	6
Lokpa	17	10	0	0
Yom	14	33	19	6

Test whether the distribution of names is uniform, given the marginal totals, using the Fisher test.

```
set.seed(8934567)
fisher.test(name_categories, simulate.p.value = TRUE, B = 1e6)
```

```
##
## Fisher's Exact Test for Count Data with simulated p-value (based on
## 1e+06 replicates)
##
## data: name_categories
## p-value = 0.00096
## alternative hypothesis: two.sided
```

The test strongly rejects the null hypothesis.

Do correspondence analysis (CA) of name categories and languages to visualize the variation.

```
category_ca <- vegan::cca(name_categories)
category_ca
```

```
## Call: cca(X = name_categories)
##
##              Inertia Rank
## Total          0.1339
## Unconstrained 0.1339    3
## Inertia is scaled Chi-square
##
## Eigenvalues for unconstrained axes:
##      CA1      CA2      CA3
## 0.12656 0.00643 0.00088
```

95% of the inertia is explained by the first axis, but we'll look at the first two (99% of explained inertia) because it's easier to visualize two dimensions at once (Fig A).

```
type_lang_scores <- scores(category_ca, scaling = "symmetric") %>%
  purrr::map2_dfr(
    c("category", "language"),
    ~ tibble::tibble(
      label = rownames(.x),
      CA1 = .x[,1],
      CA2 = .x[,2],
      type = .y
    )
  )
```

```
ggplot(aes(x = CA1, y = CA2, label = label,
           color = type, shape = type),
       data = type_lang_scores) +
  geom_vline(xintercept = 0, color = "grey80") +
  geom_hline(yintercept = 0, color = "grey80") +
  geom_point() +
  ggrepel::geom_text_repel(show.legend = FALSE) +
  scale_color_manual(values = list(language = "red", category = "grey40")) +
  scale_shape_manual(values = list(language = 16, category = 1)) +
  scale_x_continuous(sec.axis = dup_axis(name = NULL)) +
  scale_y_continuous(sec.axis = dup_axis(name = NULL)) +
  coord_fixed() +
  theme(panel.grid = element_blank())
```

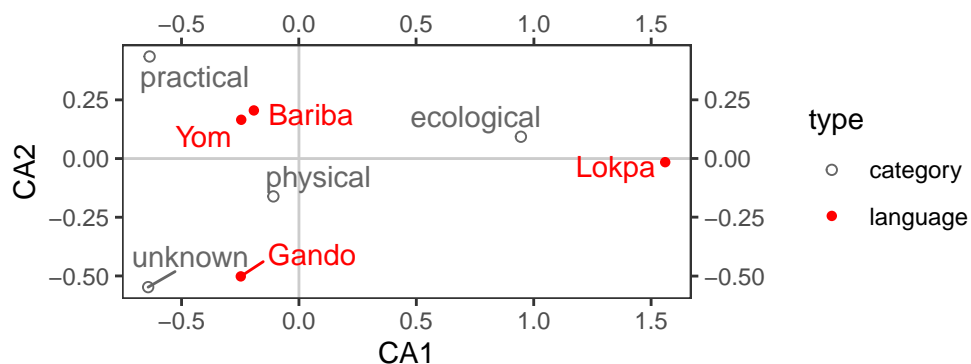


Fig A: Correspondence Analysis (CA) plot for semantic categories used in mushroom names in different languages.

1.4 Name characteristics

Make a table of name characteristics by ethnic group (Table C). Only Bariba, Gando, Lokpa, and Yom names are included, because the other language (Peulh) was not well sampled.

```
name_characteristics <- name_table %>%
  dplyr::filter(Language %in% c("Bariba", "Gando", "Lokpa", "Yom")) %>%
  dplyr::group_by(Language, characteristics) %>%
  dplyr::tally() %>%
  tidyr::pivot_wider(
    names_from = characteristics,
    values_from = n,
    values_fill = 0L
  )
name_characteristics
name_characteristics %<>%
  tibble::column_to_rownames("Language") %>%
  as.matrix()
```

Table C: Number of names in each language which belong to each characteristic.

Language	changes	color	edibility	growth habit	habitat	shape	size	technique	texture	unknown
Bariba	2	6	13	4	10	9	7	4	3	6
Gando	3	8	7	5	4	8	3	3	2	6
Lokpa	1	2	0	2	15	4	0	0	3	0
Yom	1	8	14	3	11	12	6	5	6	6

Test whether the distribution of names is uniform, given the marginal totals, using the Fisher test.

```
set.seed(2387)
name_characteristics %>%
  fisher.test(simulate.p.value = TRUE, B = 1e6)

##
## Fisher's Exact Test for Count Data with simulated p-value (based on
## 1e+06 replicates)
##
## data: .
## p-value = 0.02909
## alternative hypothesis: two.sided
```

The test strongly rejects the null hypothesis.

Do correspondence analysis (CA) of name characteristics and languages to visualize the variation.

```
characteristic_ca <- vegan::cca(name_characteristics)
characteristic_ca

## Call: cca(X = name_characteristics)
##
##              Inertia Rank
## Total          0.2094
## Unconstrained 0.2094    3
## Inertia is scaled Chi-square
##
## Eigenvalues for unconstrained axes:
##      CA1      CA2      CA3
## 0.16844 0.03319 0.00780
```

80% of the inertia is explained by the first axis, but we'll look at the first two (96% of explained inertia) because it's easier to visualize two dimensions at once (Fig B).

```

characteristic_lang_scores <-
  vegan::scores(characteristic_ca, scaling = "symmetric", choices = 1:3) %>%
  purrr::map2_dfr(
    c("characteristic", "language"),
    ~ tibble::tibble(
      label = rownames(.x),
      CA1 = .x[,1],
      CA2 = .x[,2],
      CA3 = .x[,3],
      type = .y
    )
  )

ggplot(aes(x = CA1, y = CA2, label = label,
           color = type, shape = type),
       data = characteristic_lang_scores) +
  geom_vline(xintercept = 0, color = "grey80") +
  geom_hline(yintercept = 0, color = "grey80") +
  geom_point() +
  ggrepel::geom_text_repel(show.legend = FALSE) +
  scale_color_manual(values = list(
    language = "red",
    characteristic = "grey40"
  )) +
  scale_shape_manual(values = list(
    language = 16,
    characteristic = 1
  )) +
  scale_x_continuous(sec.axis = dup_axis(name = NULL)) +
  scale_y_continuous(sec.axis = dup_axis(name = NULL),
                     expand = expansion(mult = 0.1)) +
  coord_fixed() +
  theme()

```

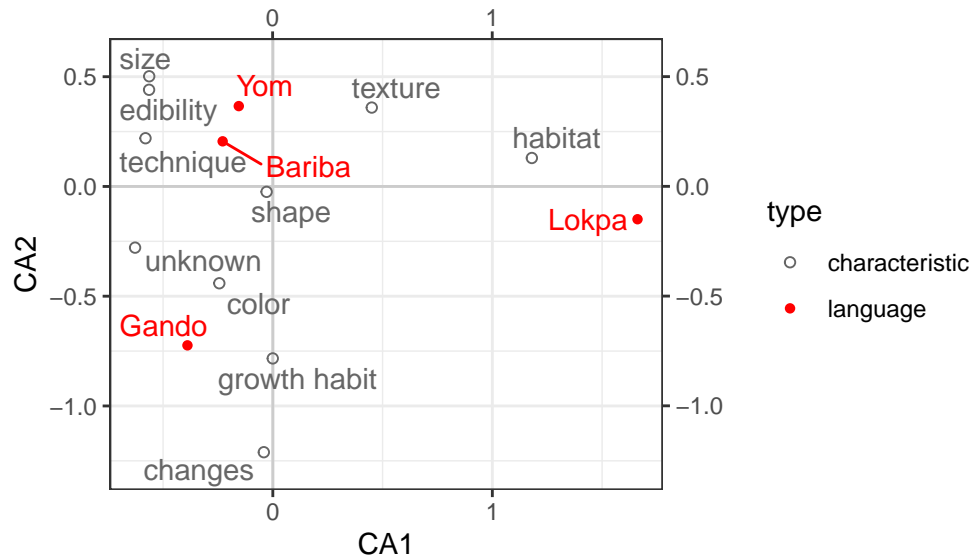


Fig B: Correspondence Analysis (CA) plot for characteristics used in mushroom names in different languages.

Write the name table to a file (S2 Table).

```
name_table %>%
  dplyr::group_by(dplyr::across(!characteristics & !type)) %>%
  dplyr::summarize(
    characteristics = paste(type, characteristics,
                           sep = ":", collapse = "; ")
  ) %>%
  dplyr::rename_all(chartr, old = ".", new = " ") %>%
  dplyr::rename_all(stringr::str_to_sentence) %>%
  openxlsx::write.xlsx(here::here("output", "name_table.xlsx"))
```

Note: `zip::zip()` is deprecated, please use `zip::zipr()` instead

2 Interview stats

These values are given in the results section of the paper.

2.1 Load data

```
interviews <- readRDS(here::here("output", "interviews.rds"))
focusgroups <- readRDS(here::here("output", "focusgroups.rds"))
```

2.2 Number of interviews

```
dplyr::n_distinct(interviews$ID)
```

```
## [1] 70
```

2.3 Number of mushroom citations

```
nrow(interviews)
```

```
## [1] 825
```

2.4 Mean and median number of citations per interviewee

```
dplyr::count(interviews, ID) %>%  
c(mean = mean(n), median = median(n))
```

```
##      mean      median  
## 11.78571 10.00000
```

2.5 Number of specimens cited in interviews

```
interview_specimens <-  
  dplyr::filter(interviews, Specimen.photo == "Specimen") %>%  
  dplyr::select(Species.photo, canonicalName, scientificName) %>%  
  unique()  
  
nrow(interview_specimens)
```

```
## [1] 50
```

2.6 Number of specimens cited in focus groups

```
focusgroup_specimens <-  
  dplyr::filter(focusgroups, Specimen.photo == "Specimen") %>%  
  dplyr::select(Species.photo, canonicalName, scientificName) %>%  
  unique()  
  
nrow(focusgroup_specimens)
```

```
## [1] 54
```

2.7 Number of specimens cited in focus groups and interviews combined

```
cited_specimens <- dplyr::union(interview_specimens, focusgroup_specimens)
nrow(cited_specimens)
```

```
## [1] 88
```

2.7.1 Number of species represented by cited specimens

```
dplyr::n_distinct(cited_specimens$canonicalName)
```

```
## [1] 53
```

2.8 Number of cited “sp.” specimens

(identified to genus but not species)

```
is_sp <- grepl("sp\\.\"", cited_specimens$canonicalName)
sum(is_sp)
```

```
## [1] 29
```

2.8.1 Number of “sp.” species represented by cited specimens

```
dplyr::n_distinct(cited_specimens$canonicalName[is_sp])
```

```
## [1] 20
```

2.9 Number of cited “cf.” specimens

(identified to species with low certainty, or to a species which is unlikely to occur in tropical Africa).

```
is_cf <- grepl("cf\\.\"", cited_specimens$canonicalName)
sum(is_cf)
```

```
## [1] 7
```

2.9.1 Number of “cf.” species represented by cited specimens

```
dplyr::n_distinct(cited_specimens$canonicalName[is_cf])
```

```
## [1] 4
```

2.10 Number of “nom. prov.” specimens

```
is_nomprov <- grepl("nom\\. prov\\.", cited_specimens$scientificName)
sum(is_nomprov)
```

```
## [1] 9
```

2.10.1 Number of “nom. prov.” species represented by cited specimens

```
dplyr::n_distinct(cited_specimens$canonicalName[is_nomprov])
```

```
## [1] 4
```

2.11 Number of specimens confidently assigned to a species:

```
is_confident <- !(is_sp | is_cf | is_nomprov)
sum(is_confident)
```

```
## [1] 43
```

2.11.1 Number of confidently assigned species represented by cited specimens:

```
dplyr::n_distinct(cited_specimens$canonicalName[is_confident])
```

```
## [1] 25
```

3 Number of recognized species

This analysis focuses strictly on the number of different species recognized by each respondent. The analogy in community ecology is species richness.

3.1 Load data

Load data for number of species known per participant.

```
knowledge <- readRDS(here::here("output", "knowledge.rds"))
```

3.2 Sampling balance

Make contingency tables for different pairs of factors, and test for independence using Fisher’s exact test.

3.2.1 Number of residents interviewed from each ethnic group and village


```
knowledge %>%
  dplyr::select(Ethnic.group, Village) %>%
  table() %T>%
  print() %>%
  fisher.test(workspace = 2e6)
```

```
##           Village
## Ethnic.group Ang Bio Fab Gan Son
##           Bar   0   0  10   0  10
##           Gan   0   0   0  11   0
##           Lok   6  10   0   0   0
##           Yom  10  10   0   0   0
##
## Fisher's Exact Test for Count Data
##
## data:  .
## p-value < 2.2e-16
## alternative hypothesis: two.sided
```

This table is highly unbalanced due to the presence of 0's for many ethnic group/village combinations, but it is still fairly balanced for the pairs that exist, so we can test for Village and Ethnic.group effects at the same time, at least in the small number of cases where it is relevant.

3.2.2 Number of residents interviewed from each gender and village

```
knowledge %>%
  dplyr::select(Gender, Village) %>%
  table() %T>%
  print() %>%
  fisher.test()
```

```
##           Village
## Gender Ang Bio Fab Gan Son
##      F   8  10   5   6   5
##      M   8  10   5   5   5
##
## Fisher's Exact Test for Count Data
##
## data:  .
## p-value = 1
## alternative hypothesis: two.sided
```

This was intentionally balanced in the sampling design.

3.2.3 Number of residents interviewed from each gender and ethnic group

```
knowledge %>%
  dplyr::select(Gender, Ethnic.group) %>%
  table() %T>%
  print() %>%
  fisher.test()
```

```
##      Ethnic.group
## Gender Bar Gan Lok Yom
##      F   10    6    8   10
##      M   10    5    8   10
##
## Fisher's Exact Test for Count Data
##
## data:  .
## p-value = 1
## alternative hypothesis: two.sided
```

This was intentionally balanced in the sampling design.

3.2.4 Number of residents interviewed from each gender and age

```
knowledge %>%
  dplyr::select(Gender, Age.group) %>%
  table() %T>%
  print() %>%
  fisher.test()
```

```
##      Age.group
## Gender <35 35+
##      F     9  25
##      M     3  30
##
## Fisher's Exact Test for Count Data
##
## data:  .
## p-value = 0.1092
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.7723891 22.4768972
## sample estimates:
## odds ratio
##  3.533855
```

This is not very balanced, so we avoid Gender \times Age interactions.

3.2.5 Number of residents interviewed from each age and ethnic group

```
knowledge %>%
  dplyr::select(Age.group, Ethnic.group) %>%
  table() %T>%
  print() %>%
  fisher.test(workspace = 2e6)
```

```
##           Ethnic.group
## Age.group Bar Gan Lok Yom
##      <35   6   2   3   1
##      35+  14   9  13  19
##
## Fisher's Exact Test for Count Data
##
## data:  .
## p-value = 0.2168
## alternative hypothesis: two.sided
```

This is quite unbalanced. This means that Ethnic.group \times Age.group interactions are problematic.

3.2.6 Number of residents interviewed from each age and village

```
knowledge %>%
  dplyr::select(Age.group, Village) %>%
  table() %T>%
  print() %>%
  fisher.test(workspace = 2e6)
```

```
##           Village
## Age.group Ang Bio Fab Gan Son
##      <35   3   1   5   2   1
##      35+  13  19   5   9   9
##
## Fisher's Exact Test for Count Data
##
## data:  .
## p-value = 0.05425
## alternative hypothesis: two.sided
```

This is very unbalanced, so we also avoid Village \times Age.group interactions

3.2.7 Number of residents interviewed from each gender, ethnic group, and village

```
knowledge %>%
  dplyr::select(Gender, Village, Ethnic.group) %>%
  table() %>%
  as.data.frame() %>%
  tibble::as_tibble() %>%
  dplyr::filter(Freq > 0) %>%
```

```

tidyr::pivot_wider(names_from = Gender, values_from = Freq) %>%
dplyr::arrange(Village, Ethnic.group) %>%
# put an X if there was a focus group
dplyr::left_join(
  dplyr::transmute(
    focusgroups,
    Village = Village,
    Ethnic.group = Ethnic.group,
    Focus.group = "X",
    Sex = Sex
  ) %>%
  dplyr::group_by(Village, Ethnic.group, Focus.group) %>%
  dplyr::summarize(star = if(dplyr::n_distinct(Sex) > 1) "*" else ""),
  by = c("Village", "Ethnic.group")
) %>%
tidyr::replace_na(list(Focus.group = "", star = "")) %>%
tidyr::unite("Focus.group", Focus.group, star, sep = "")

```

Table D: **Number of male and female interviewees from each village and ethnic group.** (Table 2 in main text.) Due to the small number of Nagot and Peulh respondents, these groups were not included in statistical analyses, but information from these interviews is included in Table 3 and S5 Table. X: Groups for which a focus group was conducted. *Seperate focus groups for men and women were conducted.

Village	Ethnic.group	F	M	Focus.group
Ang	Lok	3	3	
Ang	Yom	5	5	X
Bio	Lok	5	5	X
Bio	Yom	5	5	X
Fab	Bar	5	5	
Gan	Gan	6	5	X*
Son	Bar	5	5	X

This is close to balanced for all the combinations which exist.

3.2.8 VIF

Calculate (generalized) variance inflation factors for all predictors (Table E). This gives an error if we treat Ethnic group and Village independently, since these are aliased (e.g., every interviewee from Gando village is a member of the Gando ethnic group, and *vice versa*). Instead, for this calculation we combine the two factors into one factor with 7 levels.

```
car::vif(
  glm(
    Recognize ~ Gender + Age.group + paste(Ethnic.group, Village),
    data = knowledge,
    family = "poisson"
  )
) %>%
tibble::as_tibble(rownames = "Term") %>%
dplyr::mutate("GVIF^(1/Df)" = `GVIF^(1/(2*Df))`^2)
```

Table E: Variance inflation factors for gender, age, and ethnic group/village combination.

Term	GVIF	Df	GVIF ^{1/(2*Df)}	GVIF ^{1/Df}
Gender	1.034351	1	1.017030	1.034351
Age.group	1.264115	1	1.124329	1.264115
paste(Ethnic.group, Village)	1.233702	6	1.017656	1.035623

Despite the not-too-encouraging Fisher tests for combinations involving age group, the variance inflation factors are fine; in all cases $\text{GVIF}^{1/D_f} < 4$.

3.3 Generalized linear model (GLM)

Fit a generalized linear model using the Poisson distribution and log link function, using backwards selection by AIC.

The full model includes Gender, Ethnic.group, and Village, as well as all their interactions, as well as Age.group. At each step, the terms are removed one at a time, and the term whose removal leads to the lowest (best) AIC is removed. This is continued until removing any of the remaining terms leads to a higher (worse) AIC.

```
knowledge.glm <- glm(
  formula = Recognize ~ (Gender + Ethnic.group + Village)^2 + Age.group,
  data = knowledge,
  family = poisson(link = "log")
)
knowledge.glm <- step(
  knowledge.glm,
  scope = Recognize ~ 1,
  direction = "backward",
  trace = TRUE
)
```

```
## Start:  AIC=399.27
## Recognize ~ (Gender + Ethnic.group + Village)^2 + Age.group
##
##              Df Deviance    AIC
## - Gender:Village      2   106.39 396.14
## - Ethnic.group:Village 1   106.68 398.42
## <none>                  105.52 399.27
## - Age.group           1   113.12 404.87
## - Gender:Ethnic.group  1   117.78 409.53
##
## Step:  AIC=396.14
## Recognize ~ Gender + Ethnic.group + Village + Age.group + Gender:Ethnic.group +
##           Ethnic.group:Village
##
##              Df Deviance    AIC
## - Ethnic.group:Village 1   107.33 395.08
## <none>                  106.39 396.14
## - Age.group           1   114.05 401.79
## - Gender:Ethnic.group  3   120.99 404.74
##
## Step:  AIC=395.08
## Recognize ~ Gender + Ethnic.group + Village + Age.group + Gender:Ethnic.group
##
##              Df Deviance    AIC
```

```
## <none>                107.33 395.08
## - Village              2    111.39 395.14
## - Age.group            1    115.69 401.43
## - Gender:Ethnic.group  3    122.20 403.94
```

The final model is `Recognize ~ Gender + Ethnic.group + Village + Age.group + Gender:Ethnic.group` (AIC = 395.0799884).

Run a Type II (aka marginal; testing removal of each term vs. the full model) ANOVA table for the GLM.

```
car::Anova(knowledge.glm)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: Recognize
##              LR Chisq Df Pr(>Chisq)
## Gender          29.6659  1 5.133e-08 ***
## Ethnic.group    28.5657  1 9.057e-08 ***
## Village          4.0607  2  0.131288
## Age.group        8.3536  1  0.003849 **
## Gender:Ethnic.group 14.8637  3  0.001937 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Show the fit coefficients of the model (Table F).

```
broom::tidy(knowledge.glm, conf.int = TRUE)
```

Table F: Coefficients and associated confidence intervals for Poisson GLM.

term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	2.3687920	0.1226358	19.3156661	0.0000000	2.1216671	2.6026951
GenderM	-0.5308104	0.1450336	-3.6599121	0.0002523	-0.8184745	-0.2490229
Ethnic.groupGan	0.1137400	0.1627603	0.6988190	0.4846652	-0.2063990	0.4326856
Ethnic.groupLok	-0.3266890	0.1840173	-1.7753167	0.0758456	-0.6896585	0.0325737
Ethnic.groupYom	-0.0555042	0.1571576	-0.3531753	0.7239571	-0.3625300	0.2542262
VillageBio	0.1261335	0.1111680	1.1346204	0.2565344	-0.0908171	0.3453519
VillageFab	0.2356062	0.1460652	1.6130199	0.1067402	-0.0507143	0.5226292
Age.group.L	0.2419765	0.0853658	2.8345825	0.0045886	0.0769208	0.4118657
GenderM:Ethnic.groupGan	-0.1769147	0.2495320	-0.7089861	0.4783331	-0.6727075	0.3074618
GenderM:Ethnic.groupLok	-0.4085274	0.2591302	-1.5765335	0.1149029	-0.9248123	0.0934157
GenderM:Ethnic.groupYom	0.4512424	0.1948697	2.3156107	0.0205795	0.0703347	0.8348231

Display model checking results (Fig C).

```
# This could be done in less code,
# i.e. plot(knowledge.glm)
# but we want to put the individual plot headers in the figure caption.
par(mfrow = c(2, 2), mar = c(3, 3, 1.2, 0), oma = c(0, 0, 0, 0),
    mgp = c(1.5, 0.5, 0))
for (i in 1:4) {
  plot(knowledge.glm, caption = NULL, which = c(1:3, 5)[i])
  mtext(text = paste0("(", letters[i], ")"), side = 3, line = 0.2, adj = 0)
}
```

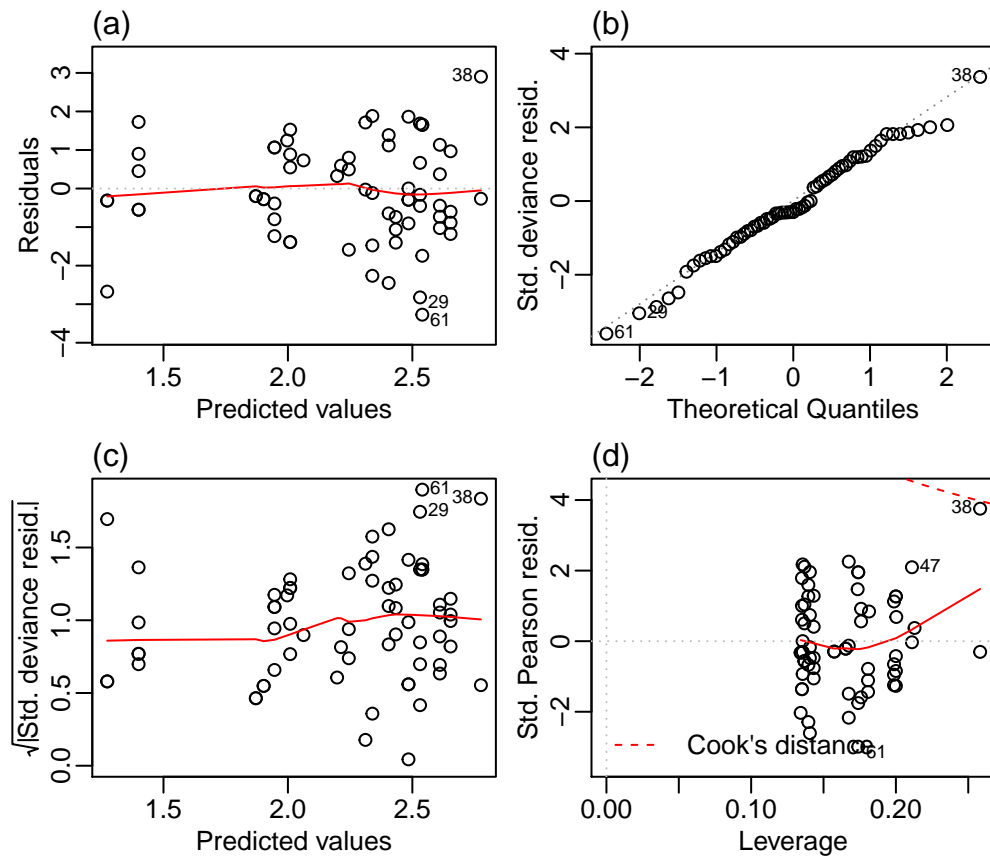


Fig C: **Model checking results for generalized linear model of number of species recognized.** (a) Residuals vs. fitted values; (b) Normal Q-Q plot; (c) Scale-location plot; (d) Residuals vs. leverage.

3.4 Reduced GLM for comparison of ethnic groups

Contrasts between all ethnic groups are not estimable using the above model, because the different groups inhabit different villages. (However, they are also not nested!)

Fit a model without Village, so that we can estimate contrasts between all ethnic groups.


```
knowledge.glm.novill <-  
  update(knowledge.glm, ~ . - Village)
```

The model is Recognize ~ Gender + Ethnic.group + Age.group + Gender:Ethnic.group (AIC = 395.1407077).

Show the fit coefficients of the no-village model (Table G).

```
broom::tidy(knowledge.glm.novill, conf.int = TRUE)
```

Table G: Coefficients and confidence intervals for Poisson GLM, fit without village as a predictor.

term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	2.4922435	0.0911950	27.3287381	0.0000000	2.3082196	2.6659828
GenderM	-0.5239803	0.1448774	-3.6167154	0.0002984	-0.8113488	-0.2425077
Ethnic.groupGan	-0.0010648	0.1448119	-0.0073527	0.9941335	-0.2886152	0.2800118
Ethnic.groupLok	-0.3577829	0.1484160	-2.4106756	0.0159230	-0.6531537	-0.0703106
Ethnic.groupYom	-0.0954543	0.1310237	-0.7285271	0.4662910	-0.3521216	0.1620568
Age.group.L	0.2158881	0.0808447	2.6704060	0.0075760	0.0600868	0.3773223
GenderM:Ethnic.groupGan	-0.1739442	0.2495428	-0.6970518	0.4857704	-0.6697585	0.3104529
GenderM:Ethnic.groupLok	-0.4085905	0.2591560	-1.5766198	0.1148830	-0.9249256	0.0934161
GenderM:Ethnic.groupYom	0.4435452	0.1947213	2.2778466	0.0227357	0.0629282	0.8268360

Show the model checking results (Fig D).

```
par(mfrow = c(2, 2), mar = c(3, 3, 1.2, 0), oma = c(0, 0, 0, 0),
    mgp = c(1.5, 0.5, 0))
for (i in 1:4) {
  plot(knowledge.glm.novill, caption = NULL, which = c(1:3, 5)[i])
  mtext(text = paste0("(", letters[i], ")"), side = 3, line = 0.2, adj = 0)
}
```

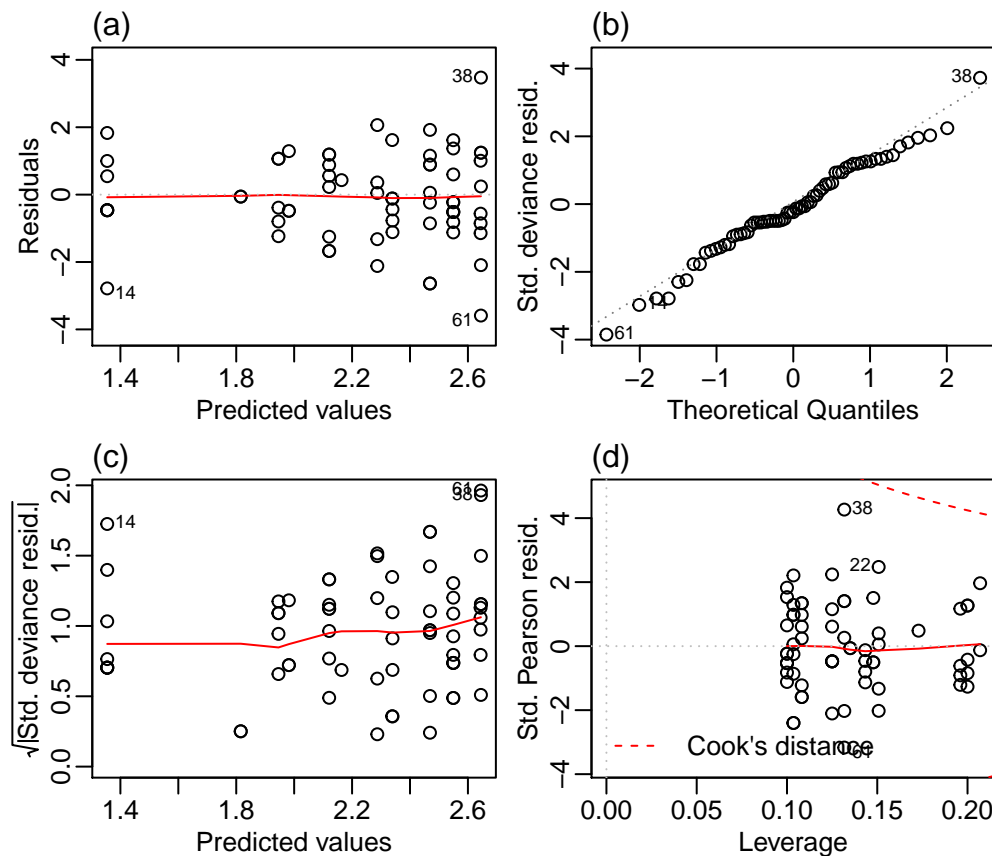


Fig D: **Model checking results for generalized linear model of number of species recognized, without using village as a variable.** (a) Residuals vs. fitted values; (b) Normal Q-Q plot; (c) Scale-location plot; (d) Residuals vs. leverage.

3.5 Marginal effects

Calculate pairwise contrasts for the full GLM.

```
coef_plot_table <-
  purrr::pmap_dfr(
```

```

# define the order we will use for each contrast,
# and that we need a difference Gender contrast for each Ethnic group
tibble::tribble(
  ~specs,      ~contrast,    ~by,
  "Gender",    "pairwise",    "Ethnic.group",
  "Ethnic.group", "revpairwise", NULL,
  "Village",   "revpairwise", NULL,
  "Age.group", "revpairwise", NULL
),
# calculate marginal means
~ emmeans::emmeans(
  knowledge.glm,
  specs = ..1,
  by = ..3,
  type = "response"
) %>%
# calculate contrasts
emmeans::contrast(..2) %>%
# add confidence intervals and compact letter display
purrr::invoke_map(
  .f = list(broom::tidy, confint, emmeans::CLD),
  .x = list(NULL, NULL, list(letters = letters, reversed = TRUE))
) %>%
# Not all elements have the same columns, so we can't specify
# the "by" columns in this join. Suppress the messages.
purrr::reduce(~suppressMessages(dplyr::left_join(.x, .y))) %>%
dplyr::mutate(term = ..1, type = ..2)
) %>%
# clean up the output for display
dplyr::mutate_at("Ethnic.group", forcats::fct_explicit_na, "") %>%
dplyr::filter(
  complete.cases(.)
) %>%
dplyr::mutate_at("contrast", stringr::str_remove, " effect") %>%
dplyr::mutate_at("contrast", forcats::as_factor) %>%
dplyr::mutate_at("term", forcats::as_factor) %>%
dplyr::mutate_at(
  "term", dplyr::recode_factor,
  Age.group = "Age group",
  Ethnic.group = "Ethnic group"
) %>%
dplyr::select(term, Ethnic.group, contrast, type, ratio,
  asymp.LCL, asymp.UCL, p.value, Group = .group)

```

NOTE: Results may be misleading due to involvement in interactions

Show the contrasts in a table (Table H).

```

# For the table, do pairwise contrasts for everything.
coef_plot_table %>%
  dplyr::filter(endsWith(type, "pairwise")) %>%

```

```

dplyr::transmute(
  Term = term,
  "Ethnic Group" = ifelse(
    Ethnic.group == "",
    "-",
    as.character(Ethnic.group)
  ),
  Contrast = contrast,
  Ratio = paste0(
    formatC(ratio, digits = 2, format = "f"),
    " (",
    formatC(asymp.LCL, digits = 2, format = "f"),
    "-",
    formatC(asymp.UCL, digits = 2, format = "f"),
    ")"
  ),
  P = formatC(p.value, digits = 2)
) %T>%
readr::write_csv(here::here("output/table5.csv"))

```

Table H: **Estimated response ratios for Poisson GLM.** (Table 5 in main text.) Pairwise contrast ratios relating number of mushroom species known to respondent gender, age group, ethnic group, and village. The effect of respondent gender was modeled as an interaction with ethnic group, so different values are given for the four ethnic groups; Bar = Bariba, Gan = Gando, Lok = Lokpa, Yom = Yom. For each contrast, the ratio is given along with its 95% confidence interval in parentheses.

Term	Ethnic Group	Contrast	Ratio	P
Gender	Bar	F / M	1.70 (1.28–2.26)	0.00025
Gender	Gan	F / M	2.03 (1.35–3.04)	0.0006
Gender	Lok	F / M	2.56 (1.67–3.92)	1.7e-05
Gender	Yom	F / M	1.08 (0.84–1.39)	0.54
Ethnic group	-	Yom / Lok	2.02 (1.45–2.79)	2.1e-07
Village	-	Bio / Ang	1.13 (0.84–1.54)	0.79
Village	-	Son / Fab	0.79 (0.53–1.18)	0.49
Age group	-	35+ / <35	1.41 (1.11–1.78)	0.0046

Now do the same calculations, but only for differences between ethnic groups in the no-village model.

```
coef_table_novill <-
  # calculate marginal means
  emmeans::emmeans(
    knowledge.glm.novill,
    "Ethnic.group",
    type = "response"
  ) %>%
  # It will be easier to read four effect contrasts than six pairwise
  # contrasts
  emmeans::contrast("eff") %>%
  # Add confidence intervals and compact letter display.
  purrr::invoke_map(
    .f = list(broom::tidy, confint, emmeans::CLD),
    .x = list(NULL, NULL, list(Letters = letters, reversed = TRUE))
  ) %>%
  # Not all elements have the same columns, so we can't specify
  # the "by" columns in this join. Suppress the messages.
  purrr::reduce(~suppressMessages(dplyr::left_join(.x, .y))) %>%
  dplyr::mutate_at(
    "contrast",
    stringr::str_remove,
    pattern = " effect"
  ) %>%
  # Clean up output for display
  dplyr::select(contrast, ratio, asymp.LCL, asymp.UCL, p.value, Group = .group) %>%
  dplyr::mutate(term = "Ethnic group", type = "eff")
```

NOTE: Results may be misleading due to involvement in interactions

```

coef_table_novill %>%
  dplyr::transmute(
    `Ethnic Group` = contrast,
    Ratio = paste0(
      formatC(ratio, digits = 2, format = "f"),
      " (",
      formatC(asymp.LCL, digits = 2, format = "f"),
      "-",
      formatC(asymp.UCL, digits = 2, format = "f"),
      ")"
    ),
    P = formatC(p.value, digits = 2),
    Group = Group
  )

```

Table I: **Estimated response ratios for Poisson GLM without village effect.** Effect ratios relating number of mushroom species known to respondent ethnic group vs. the all groups mean; Bar = Bariba, Gan = Gando, Lok = Lokpa, Yom = Yom. For each ethnic group, the ratio is given along with its 95% confidence interval in parentheses. P indicates the result of a test that the ethnic group differs from the overall mean. Ethnic groups with different letters in the “Groups” column differed in a pairwise test at $p < 0.05$.

Ethnic Group	Ratio	P	Group
Bar	1.14 (0.96–1.35)	0.073	a
Gan	1.04 (0.85–1.29)	0.61	a
Lok	0.65 (0.52–0.81)	3.9e-06	b
Yom	1.29 (1.10–1.52)	0.00013	a

Make a figure showing the effects (Fig E).

```
# put together contrast tables for the model with and without Village
dplyr::bind_rows(
  coef_plot_table,
  coef_table_novill
) %>%
# For Gender, Age group, and Village, we want pairwise contrasts.
# For Ethnic Group, looking at effect contrasts with group indicators is
# more succinct.
dplyr::filter(
  !(term %in% c("Gender", "Age group") & endsWith(type, "eff")),
  !(term %in% c("Ethnic group") & endsWith(type, "pairwise"))
) %>%
# don't display "NA" when there is no interaction with Ethnic group
dplyr::mutate_at("Ethnic.group", forcats::fct_explicit_na, "") %>%
# reverse the ordering of contrasts.
# They will be on the vertical axis,
# but we want them ordered from top to bottom.
dplyr::mutate_at("contrast", forcats::as_factor) %>%
dplyr::mutate_at("contrast", forcats::fct_rev) %>%
# Define the order that terms will be presented, and abbreviate Age.
dplyr::mutate_at(
  "term",
  factor,
  levels = c("Gender", "Age group", "Village", "Ethnic group"),
  labels = c("Gender", "Age", "Village", "Ethnic group")
) %>%
# remove the group indicators when there is only one group
# and for pairwise contrasts
dplyr::group_by(Ethnic.group, term, type) %>%
dplyr::mutate(
  Group = if (dplyr::n_distinct(Group) > 1 && all(type == "eff"))
    Group else ""
) %>%
# make the plot
ggplot(aes(
  x = contrast,
  y = ratio,
  ymin = asymp.LCL,
  ymax = asymp.UCL,
  # color significant pairwise contrasts red
  color = ifelse(
    endsWith(type, "pairwise") & (p.value < 0.05),
    "red",
    "black"
  ),
  label = ifelse(type == "eff", Group, "")
)) +
# vertical line at 0 (hline because coordinates are swapped)
geom_hline(
  aes(yintercept = 1),
```

```

    alpha = 0.3,
    linetype = 2
) +
# draw points and confidence intervals
geom_pointrange(fatten = 3) +
# interpret the color values ("red" and "black") as literal colors.
scale_color_identity() +
# draw group identifiers
geom_text(mapping = aes(y = asymp.UCL), nudge_y = 0.1) +
# flip axes
coord_flip() +
# make sure 0 is included in the range
scale_y_continuous(limits = c(0, NA)) +
# separate the different terms
ggh4x::facet_nested(
  term + Ethnic.group ~ .,
  scales = "free_y",
  space = "free_y",
  nest_line = TRUE
) +
# label axes
xlab(NULL) +
ylab("Ratio") +
# remove gray rectangles around term labels
theme(strip.background = element_blank())

```

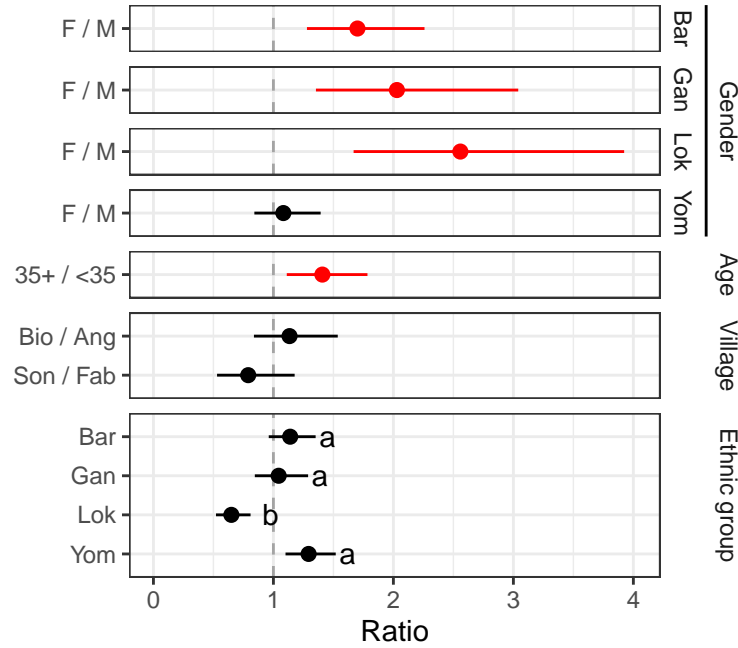



Fig E: **Estimated marginal effect contrasts based on Poisson GLM.** (Fig 2 in main text.) Multiplicative effects on the number of mushroom species recognized based on respondent gender, ethnic group, age group, and village. For gender, age, and village, pairwise contrast ratios are shown. For ethnic group, the effect of each group is contrasted with the all-group mean. Points represent the estimated contrast ratio for each variable with each other variable held constant. Horizontal lines represent the 95% confidence interval. Dashed vertical line at 1 represents no effect. Points and confidence intervals are colored red for pairwise contrasts which were significant at $p < 0.05$. The effect of respondent gender was found to vary significantly among ethnic groups, so different values are given for each ethnic group; Bar = Bariba, Gan = Gando, Lok = Lokpa, Yom = Yom. For villages, only contrasts between villages which were comparable due to containing respondents from the same ethnic group are included. Neither of these contrasts were statistically significant at $p < 0.05$. Village abbreviations: Ang = Angaradebou, Bio = Bio Sika, Fab = Faba, Son = Sonnoumon. Comparisons between ethnic groups are based on a GLM which did not include village as a predictor. For ethnic group, the results of pairwise tests are indicated by letters, where groups which have the same letter do not differ significantly.

3.6 Model predictions

Plot the model predictions along with observed data (Fig F).

```
kn.plot <-
  dplyr::select(knowledge, Age.group, Gender, Ethnic.group, Village) %>%
  unique() %>%
  tidyr::complete(
    tidyr::nesting(Ethnic.group, Village),
    Age.group,
    Gender
  ) %>%
  cbind(predict(knowledge.glm, ., type = "link", se.fit = TRUE)) %>%
  dplyr::mutate(
    lower = fit - 2 * se.fit,
    upper = fit + 2 * se.fit
  ) %>%
  dplyr::mutate_at(
    c("fit", "lower", "upper"),
    family(knowledge.glm)$linkinv
  ) %>%
  dplyr::mutate(
    Ethnic.group = dplyr::recode_factor(
      Ethnic.group,
      Lok = "Lokpa",
      Yom = "Yom",
      Bar = "Bariba",
      Gan = "Gando"
    ),
    Village = dplyr::recode_factor(
      Village,
      Ang = "Angaradebou",
      Bio = "Bio Sika",
      Fab = "Faba",
      Son = "Sonnoumon",
      Gan = "Gando"
    )
  ) %>%
  ggplot(
    aes(y = fit, x = Age.group, group = Gender, color = Gender,
        linetype = Gender, shape = Gender)
  ) +
  geom_ribbon(
    aes(x = Age.group, ymin = lower, ymax = upper,
        group = Gender, fill = Gender),
    alpha = 0.4,
    inherit.aes = FALSE
  ) +
  geom_line() +
  facet_wrap(
    ~ Village + Ethnic.group,
```

```

ncol = 2,
strip.position = "top",
dir = "h"
) +
geom_jitter(
  data = knowledge %>%
    dplyr::mutate(
      Ethnic.group = dplyr::recode_factor(
        Ethnic.group,
        Lok = "Lokpa",
        Yom = "Yom",
        Bar = "Bariba",
        Gan = "Gando"
      ),
      Village = dplyr::recode_factor(
        Village,
        Ang = "Angaradebou",
        Bio = "Bio Sika",
        Fab = "Faba",
        Son = "Sonnoumon",
        Gan = "Gando"
      )
    ),
  aes(x = Age.group, y = Recognize),
  width = 0.1
) +
scale_shape_manual(values = c(F = 1, M = 2)) +
xlab("Age group") +
ylab("Number of species recognized") +
guides(
  fill = guide_legend(
    direction = "horizontal",
    label.position = "bottom",
    title.position = "top"
  ),
  color = guide_legend(
    direction = "horizontal",
    label.position = "bottom",
    title.position = "top"
  ),
  shape = guide_legend(
    direction = "horizontal",
    label.position = "bottom",
    title.position = "top"
  ),
  linetype = guide_legend(
    direction = "horizontal",
    label.position = "bottom",
    title.position = "top"
  )
) +
theme(
  panel.grid.major.y = element_line(color = "gray70"),

```

```
axis.text.x = element_text(angle = -30, hjust = 0, vjust = 1)
)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = residual.scale, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```
lemon::reposition_legend(kn.plot, position = "center", panel = "panel-2-4")
```

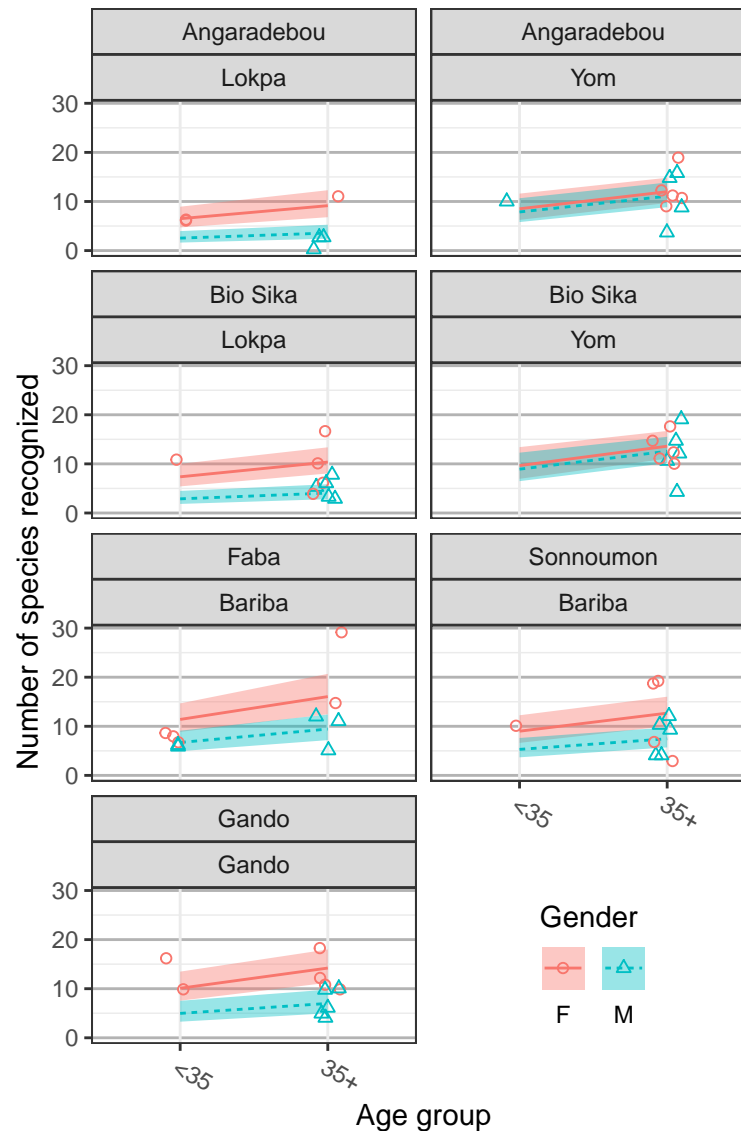


Fig F: **Observed data (points) and model predictions (lines) for the Poisson GLM.** Number of mushroom species recognized vs. respondent gender, age group, village, and ethnic group. Shaded areas represent 95% confidence intervals of model predictions.

4 “Community” composition analysis

This analysis focuses on which particular species are recognized by different respondents, or the preference ratings given to each species by different respondents, but not the total number of species recognized. The analogy in community ecology is community composition.

4.1 Load data

Load knowledge and preference matrices, as well as associated biographical data.

```
knowledge.matrix <- readRDS(here::here("output", "knowledge_matrix.rds"))
knowledge.biodata <- readRDS(here::here("output", "knowledge_biodata.rds"))
preference.matrix <- readRDS(here::here("output", "preference_matrix.rds"))
preference.biodata <- readRDS(here::here("output", "preference_biodata.rds"))
```

4.2 Knowledge CCA

This analysis focuses on differences in which species are known by different interviewees. The knowledge matrix has respondents along the rows, and mushroom species along the columns. The values are 1 if the respondent recognized that mushroom species, and 0 if they did not.

Do constrained correspondence analysis (CCA) on the knowledge matrix, constrained by the biographical data.

```
set.seed(2184761)
k.cca <- vegan::cca(
  formula = knowledge.matrix ~ Age.group + (Ethnic.group + Gender + Village)^2,
  data = knowledge.biodata
)
k.cca <- vegan::ordistep(
  k.cca,
  scope = knowledge.matrix ~ 1,
  direction = "backward",
  trace = TRUE,
  permutations = how(nperm = 999)
)
```

```
##
## Start: knowledge.matrix ~ Age.group + (Ethnic.group + Gender + Village)^2
##
##              Df      AIC      F Pr(>F)
## - Ethnic.group:Gender  1 210.62 0.8671  0.653
## - Gender:Village      2 210.53 1.2128  0.125
## - Ethnic.group:Village 1 211.19 1.3277  0.112
## - Age.group           1 211.44 1.5277  0.045 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: knowledge.matrix ~ Age.group + Ethnic.group + Gender + Village +      Ethnic.group:Village + G
##
##              Df      AIC      F Pr(>F)
## - Ethnic.group:Village 1 210.21 1.2907  0.153
```

```
## - Age.group          1 210.35 1.4106 0.078 .
## - Gender:Village     4 208.60 1.2576 0.049 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: knowledge.matrix ~ Age.group + Ethnic.group + Gender + Village +      Gender:Village
##
##              Df      AIC      F Pr(>F)
## - Gender:Village 4 207.92 1.2209 0.082 .
## - Age.group      1 209.93 1.4292 0.074 .
## - Ethnic.group   1 211.07 2.3990 0.001 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The selected model is $\text{knowledge.matrix} \sim \text{Age.group} + \text{Ethnic.group} + \text{Gender} + \text{Village} + , \text{Gender:Village}$ ($R^2 = 0.278$; $R^2_{adj} = 0.132$)

Find a way to orient the CCA so that the Gando ethnic group is positive on all axes. Because the direction of each axis is arbitrary in CCA, this does not effect the interpretation of results. However, it ensures some consistency between different methods used during analysis or review. The choice of Gando was arbitrary, based on early results.

```
k.cca.orient <- diag(sign(k.cca$CCA$biplot["Ethnic.groupGan",])) %>%
  magrittr::set_rownames(colnames(k.cca$CCA$biplot)) %>%
  magrittr::set_colnames(colnames(k.cca$CCA$biplot))
```

Do per-axis ANOVA (Table J). This tests each axis sequentially.

```
set.seed(2184761)
k.anova <- anova(k.cca, by = "axis", permutations = 9999, cutoff = 0.05) %>%
  broom::tidy()
```

```
## Warning in tidy.anova(.): The following column names in ANOVA output were not
## recognized or transformed: ChiSquare
```

```
k.anova
```

Table J: ANOVA table for CCA axes of knowledge CCA.

term	df	ChiSquare	statistic	p.value
CCA1	1	0.1550241	4.7266030	0.0001
CCA2	1	0.1324464	4.0382201	0.0002
CCA3	1	0.1136788	3.4660065	0.0021
CCA4	1	0.0767233	2.3392533	0.1609
CCA5	1	0.0519421	1.5836861	NA
CCA6	1	0.0424078	1.2929921	NA
CCA7	1	0.0329295	1.0040018	NA
CCA8	1	0.0293139	0.8937653	NA
CCA9	1	0.0233014	0.7104473	NA
CCA10	1	0.0137503	0.4192380	NA
CCA11	1	0.0096839	0.2952556	NA
Residual	54	1.7711037	NA	NA

Choose axes which were significant at $p \leq 0.05$.

```
k.choices <- which(k.anova$p.value <= 0.05)
k.cca.orient <- k.cca.orient[k.choices, k.choices]
```

Do marginal per-term ANOVA (Table K). This tests the full model vs. the model with each individual term removed.

```
set.seed(2184761)
anova(k.cca, by = "margin", permutations = 9999) %>%
  broom::tidy()
```

```
## Warning in tidy.anova(.): The following column names in ANOVA output were not
## recognized or transformed: ChiSquare
```

Table K: ANOVA table for explanatory terms in knowledge CCA.

term	df	ChiSquare	statistic	p.value
Age.group	1	0.0468757	1.429215	0.0684
Ethnic.group	1	0.0786816	2.398959	0.0002
Gender:Village	4	0.1601702	1.220876	0.0641
Residual	54	1.7711037	NA	NA

Display the projections of the predictors on the significant axes, after reflection to “standard” orientation with Gando ethnicity positive (Table L).

```
scores(  
  k.cca,  
  display = "cn",  
  choices = k.choices  
) %>%  
  magrittr::multiply_by_matrix(k.cca.orient) %>%  
  tibble::as_tibble(rownames = "Category")
```


Table L: Centroids of predictor values for knowledge CCA.

Category	CCA1	CCA2	CCA3
Age.group<35	0.4820417	0.7324646	0.4844077
Age.group35+	-0.0927356	-0.1409122	-0.0931908
Ethnic.groupBar	-0.9060251	1.0044064	-0.0380426
Ethnic.groupGan	1.5325401	0.8604514	0.7071044
Ethnic.groupLok	0.1489670	-0.9718772	0.9877812
Ethnic.groupYom	-0.0137681	-0.7987715	-0.6958660
GenderF	0.2558404	0.1793563	-0.3618414
GenderM	-0.3899502	-0.2733737	0.5515164
VillageAng	-0.1778572	-0.6329423	-0.5114109
VillageBio	0.1853490	-1.0043450	0.0184292
VillageFab	-0.6487072	0.6137513	0.4546604
VillageSon	-1.2238884	1.4869803	-0.6466756

Make various calculations that will help in plotting the CCA.

```
# coordinates for each interviewee ("site") and each species.
k.scores <- vegan::scores(
  k.cca,
  display = c("sites", "species"),
  scaling = "symmetric",
  choices = k.choices
)
# reflect into standard orientation
k.scores$sites <- k.scores$sites %*% k.cca.orient
k.scores$species <- k.scores$species %*% k.cca.orient

# scores for interviewees ("sites")
k.people <- k.scores$sites %>%
  tibble::as_tibble(rownames = "ID") %>%
  dplyr::left_join(knowledge.biodata, by = "ID")

# calculate species scores
k.species <- k.scores$species %>%
  tibble::as_tibble(rownames = "sp") %>%
  dplyr::mutate(
    length12 = sqrt(CCA1*CCA1 + CCA2*CCA2),
    length13 = sqrt(CCA1*CCA1 + CCA3*CCA3),
    length23 = sqrt(CCA2*CCA2 + CCA3*CCA3)
  )

# Choose the top 10 most extreme species in each axis combination.
k.species12 <- k.species %>%
  dplyr::arrange(dplyr::desc(length12)) %>%
  dplyr::mutate(sp = ifelse(dplyr::cur_group_rows() <= 10, sp, ""))

k.species13 <- k.species %>%
```

```

dplyr::arrange(dplyr::desc(length13)) %>%
dplyr::mutate(sp = ifelse(dplyr::cur_group_rows() <= 10, sp, ""))

k.species23 <- k.species %>%
dplyr::arrange(dplyr::desc(length23)) %>%
dplyr::mutate(sp = ifelse(dplyr::cur_group_rows() <= 10, sp, ""))

# Species abbreviation text to include in captions
k.species12.abbrev <-
dplyr::filter(k.species12, sp != "") %>%
dplyr::arrange(sp) %>%
dplyr::left_join(abbrevkey, by = c("sp" = "Abbrev")) %>%
glue::glue_data("{sp} = {canonicalName}") %>%
glue::glue_collapse(", ")

k.species13.abbrev <-
dplyr::filter(k.species13, sp != "") %>%
dplyr::arrange(sp) %>%
dplyr::left_join(abbrevkey, by = c("sp" = "Abbrev")) %>%
glue::glue_data("{sp} = {canonicalName}") %>%
glue::glue_collapse(", ")

k.species23.abbrev <-
dplyr::filter(k.species23, sp != "") %>%
dplyr::arrange(sp) %>%
dplyr::left_join(abbrevkey, by = c("sp" = "Abbrev")) %>%
glue::glue_data("{sp} = {canonicalName}") %>%
glue::glue_collapse(", ")

# background points for each facet in the sidebar plots
k.bg <- k.people %>%
dplyr::select(dplyr::starts_with("CCA"), Ethnic.group) %>%
dplyr::mutate(EG = TRUE) %>%
tidyr::spread(Ethnic.group, EG, fill = FALSE) %>%
dplyr::mutate_at(c("Bar", "Gan", "Lok", "Yom"), `!`) %>%
tidyr::gather(Ethnic.group, EG, Bar:Yom) %>%
dplyr::filter(EG)

k.bg.vill <- k.people %>%
dplyr::select(dplyr::starts_with("CCA"), Village) %>%
dplyr::mutate(V = TRUE) %>%
tidyr::spread(Village, V, fill = FALSE) %>%
dplyr::mutate_at(c("Ang", "Bio", "Gan", "Fab", "Son"), `!`) %>%
tidyr::gather(Village, V, Ang:Son) %>%
dplyr::filter(V)

```

Create a biplot for the first two CCA axes, with a sidebar to more easily read ethnic group/village combinations (Fig G).

```

cca12 <-
k.people %>%
ggplot(aes(
  x = CCA1,

```

```

    y = CCA2,
    color = VillageGroup,
    shape = VillageGroup,
    fill = VillageGroup
  )) +
  # light axes
  geom_hline(yintercept = 0, color = "grey80") +
  geom_vline(xintercept = 0, color = "grey80") +
  # points for "sites" (i.e. interviewees)
  geom_point(stroke = 1.5, alpha = 0.8) +
  # points for species
  geom_point(
    aes(x = CCA1, y = CCA2),
    data = k.species12,
    inherit.aes = FALSE,
    shape = 1
  ) +
  # labels for the species
  ggrepel::geom_text_repel(
    aes(CCA1, CCA2, label = sp),
    data = dplyr::bind_rows(
      k.species12,
      dplyr::mutate(k.people, sp = "")
    ),
    size = 3,
    inherit.aes = FALSE,
    box.padding = 0.5,
    point.padding = 0.2,
    segment.alpha = 0.2,
    min.segment.length = 0
  ) +
  theme_cca_main +
  scale_villagegroup

cca12_side <- k.people %>%
  ggplot(aes(CCA1, CCA2, shape = Village, color = Village)) +
  geom_hline(yintercept = 0, color = "grey80") +
  geom_vline(xintercept = 0, color = "grey80") +
  geom_point(aes(CCA1, CCA2), color = "gray50", shape = ".", data = k.bg, inherit.aes = FALSE) +
  geom_point() +
  theme_cca_side +
  scale_village

ggpubr::ggarrange(cca12, cca12_side, widths = c(2.05, 1.06))

```

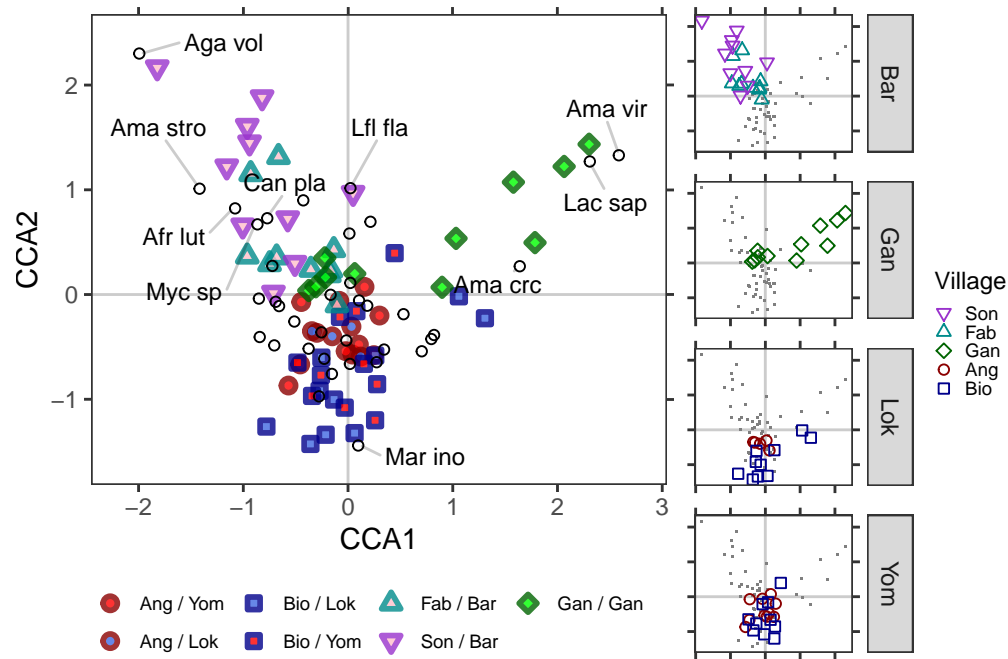


Fig G: Biplot of CCA axes 1 and 2 for species known. (Fig 3 in main text.) Individual respondents are shown as colored shapes, while mushroom species empty circles colored shapes. Only the ten mushroom species with the strongest axis associations are labeled. Village abbreviations: Ang = Angaradebou, Bio = Bio Sika, Fab = Faba, Gan = Gando, Son = Sonnoumon. Ethnic group abbreviations: Bar = Bariba, Gan = Gando, Lok = Lokpa, Yom = Yom. Species abbreviations: Afr lut = *Afroboletus luteolus*, Aga vol = *Agaricus volvatulus*, Ama crc = *Amanita "crassiconus"*, Ama stro = *Amanita strobilaceovolvata*, Ama vir = *Amanita "virido-odorata"*, Can pla = *Cantharellus platyphyllus*, Lac sap = *Lactarius saponaceus*, Lfl fla = *Lactifluus flammans*, Mar ino = *Marasmiellus inoderma*, Myc sp = *Mycoamaranthus sp.*. Side plots: respondents for each ethnic group are plotted separately, with point shape and color representing village.

Create a biplot for CCA axes 1 and 3 (Fig ??). These are the ones which show the Gender x Village interaction most clearly, so the sidebar is for Gender/Village combinations.

```
cca13 <-
  k.people %>%
  ggplot(aes(
    x = CCA1,
    y = CCA3,
    color = VillageGroup,
    shape = VillageGroup,
    fill = VillageGroup
  )) +
  # light axes
  geom_hline(yintercept = 0, color = "grey80") +
  geom_vline(xintercept = 0, color = "grey80") +
  # points for "sites" (i.e. interviewees)
  geom_point(stroke = 1.5, alpha = 0.8) +
  # points for species
  geom_point(
    aes(x = CCA1, y = CCA3),
    data = k.species13,
    inherit.aes = FALSE,
    shape = 1
  ) +
  # labels for the species
  ggrepel::geom_text_repel(
    aes(CCA1, CCA3, label = sp),
    data = dplyr::bind_rows(
      k.species13,
      dplyr::mutate(k.people, sp = "")
    ),
    size = 3,
    inherit.aes = FALSE,
    box.padding = 0.5,
    point.padding = 0.2,
    segment.alpha = 0.2,
    min.segment.length = 0
  ) +
  theme_cca_main +
  scale_villagegroup

cca13_side <- k.people %>%
  ggplot(aes(CCA1, CCA3, shape = Gender, color = Gender)) +
  geom_hline(yintercept = 0, color = "grey80") +
  geom_vline(xintercept = 0, color = "grey80") +
  geom_point(aes(CCA1, CCA3), color = "gray50", shape = ".",
    data = k.bg.vill, inherit.aes = FALSE) +
  geom_point() +
  scale_shape_manual(values = c("F" = 1, "M" = 2)) +
  scale_color_manual(values = c("F" = "red", "M" = "blue")) +
  coord_fixed() +
  scale_x_continuous(label = NULL, name = NULL,
```

```

sec.axis = dup_axis(name = NULL, labels = NULL)) +
scale_y_continuous(label = NULL, name = NULL,
sec.axis = dup_axis(name = NULL, labels = NULL)) +
lemon::facet_rep_wrap(~Village, ncol = 1, strip.position = "right") +
theme(
strip.placement = "outside",
legend.box.spacing = unit(5, "pt"),
legend.box.margin = margin(3, 3, 3, 3),
legend.margin = margin(0, 0, 0, 0),
legend.text = element_text(size = 7),
legend.title = element_text(size = 9),
legend.key.size = unit(3, "mm"),
plot.margin = margin(0, 0, 0, 5),
panel.grid = element_blank()
)

ggpubr::ggarrange(cca13, cca13_side, nrow = 1, widths = c(2.05, 0.94))

```

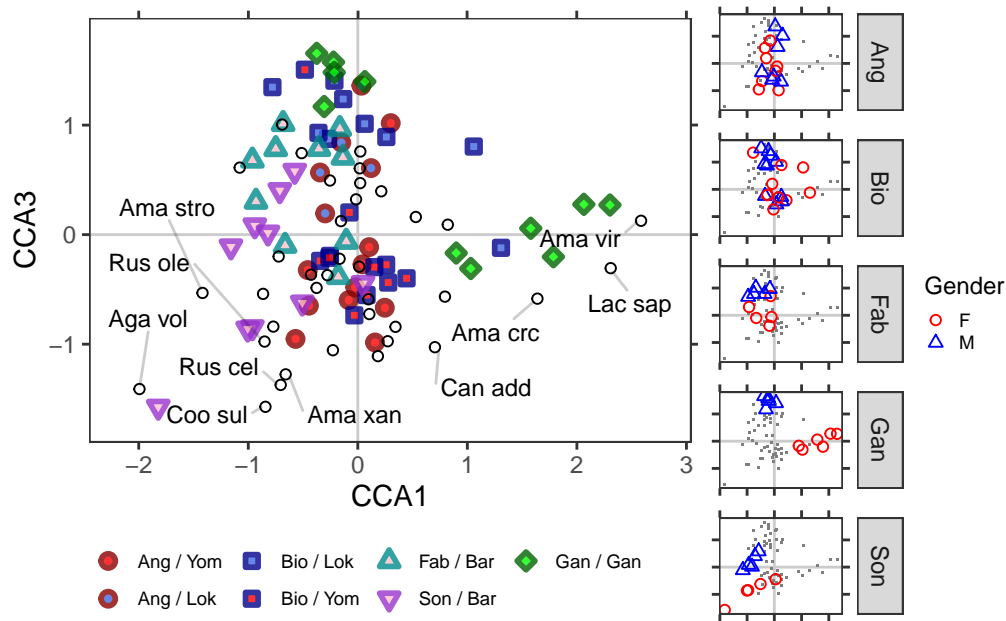


Fig H: Biplot of CCA axes 1 and 3 for species known. (Fig 4 in main text.) Individual respondents are shown as colored shapes, while mushroom species empty circles colored shapes. Only the ten mushroom species with the strongest axis associations are labeled. Village abbreviations: Ang = Angaradebou, Bio = Bio Sika, Fab = Faba, Gan = Gando, Son = Sonnoumon. Ethnic group abbreviations: Bar = Bariba, Gan = Gando, Lok = Lokpa, Yom = Yom. Species abbreviations: Aga vol = *Agaricus volvatulus*, Ama crc = *Amanita "crassiconus"*, Ama stro = *Amanita strobilaceovolvata*, Ama vir = *Amanita "virido-odorata"*, Ama xan = *Amanita xanthogala*, Can add = *Cantharellus addaiensis*, Coo sul = *Cookeina sulcipes*, Lac sap = *Lactarius saponaceus*, Rus cel = *Russula cellulata*, Rus ole = *Russula oleifera*. Side plots: respondents for each ethnic group are plotted separately, with point shape and color representing gender.

Make a plot of CCA axes 2 and 3 (Fig H). These axes show the variation between the Lokpa and Yom in Angaradebou and Bio Sika most clearly, so the sidebar is again Ethnic group/Village.

```

cca23 <-
  k.people %>%
  ggplot(aes(
    x = CCA2,
    y = CCA3,
    color = VillageGroup,
    shape = VillageGroup,
    fill = VillageGroup
  )) +
  # light axes
  geom_hline(yintercept = 0, color = "grey80") +
  geom_vline(xintercept = 0, color = "grey80") +
  # points for "sites" (i.e. interviewees)
  geom_point(stroke = 1.5, alpha = 0.8) +
  # points for species
  geom_point(
    aes(x = CCA2, y = CCA3),
    data = k.species23,
    inherit.aes = FALSE,
    shape = 1
  ) +
  # labels for the species
  ggrepel::geom_text_repel(
    aes(CCA2, CCA3, label = sp),
    data = dplyr::bind_rows(
      k.species23,
      dplyr::mutate(k.people, sp = "")
    ),
    size = 3,
    inherit.aes = FALSE,
    box.padding = 0.5,
    point.padding = 0.2,
    segment.alpha = 0.2,
    min.segment.length = 0
  ) +
  theme_cca_main +
  scale_villagegroup

cca23_side <- k.people %>%
  ggplot(aes(CCA2, CCA3, shape = Village, color = Village)) +
  geom_hline(yintercept = 0, color = "grey80") +
  geom_vline(xintercept = 0, color = "grey80") +
  geom_point(aes(CCA2, CCA3), color = "gray50", shape = ".", data = k.bg, inherit.aes = FALSE) +
  geom_point() +
  theme_cca_side +
  scale_village

ggpubr::ggarrange(cca23, cca23_side, widths = c(2.05, 1.072))

```

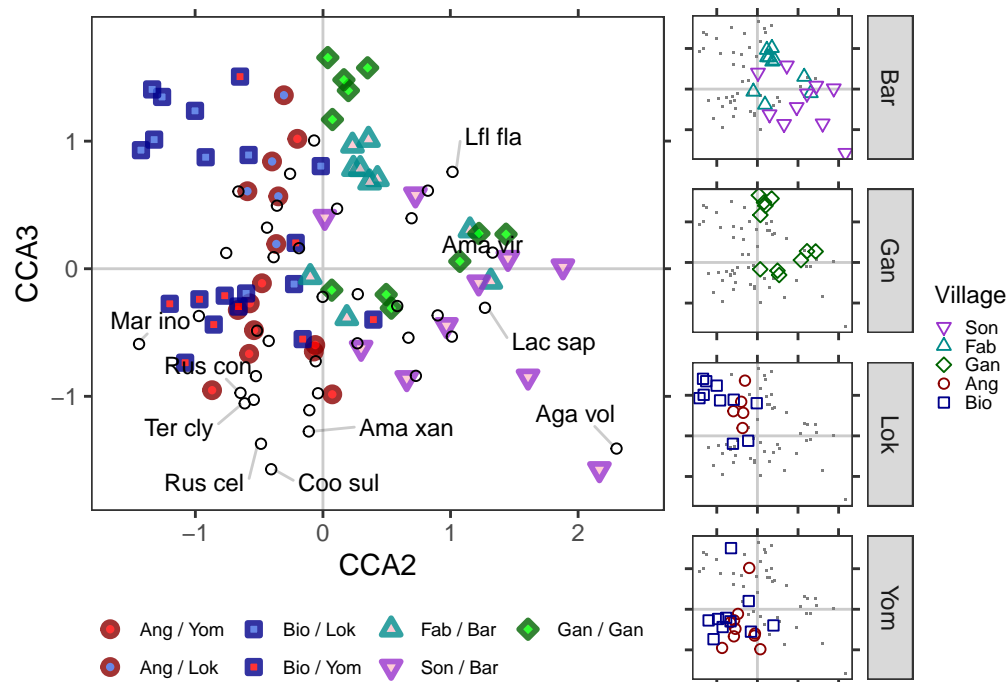


Fig I: **Biplot of CCA axes 2 and 3 for species known.** (Fig 4 in main text.) Individual respondents are shown as colored shapes, while mushroom species empty circles colored shapes. Only the ten mushroom species with the strongest axis associations are labeled. Village abbreviations: Ang = Angaradebou, Bio = Bio Sika, Fab = Faba, Gan = Gando, Son = Sonnoumon. Ethnic group abbreviations: Bar = Bariba, Gan = Gando, Lok = Lokpa, Yom = Yom. Species abbreviations: Aga vol = *Agaricus volvatulus*, Ama vir = *Amanita* “virido-odorata”, Ama xan = *Amanita xanthogala*, Coo sul = *Cookeina sulcipes*, Lac sap = *Lactarius saponaceus*, Lfl fla = *Lactifluus flammans*, Mar ino = *Marasmiellus inoderma*, Rus cel = *Russula cellulata*, Rus con = *Russula congoana*, Ter cly = *Termitomyces clypeatus*. Side plots: respondents for each ethnic group are plotted separately, with point shape and color representing gender.

4.3 Preference CCA

This analysis focuses on differences in the preference ratings given to different species by different interviewees. The preference matrix has respondents along the rows, and mushroom species along the columns. The values are the preference given by that respondent for that mushroom on a scale of 0-4. Mushrooms which were not recognized are also given a score of 0 (inedible).

Do constrained correspondence analysis (CCA) for the preference matrix, constrained by the biographical data, including interactions. Then remove variables or interactions which are given a p-value > 0.10 by a permutation test.

```
set.seed(32195761)
p.cca <- vegan::cca(
  formula = preference.matrix ~ Age.group + (Ethnic.group + Gender + Village)^2,
  data = preference.biodata
)
p.cca <- vegan::ordistep(
  p.cca,
  scope = preference.matrix ~ 1,
  direction = "backward",
  trace = TRUE,
  permutations = how(nperm = 999)
)
```

```
##
## Start: preference.matrix ~ Age.group + (Ethnic.group + Gender + Village)^2
##
##              Df      AIC      F Pr(>F)
## - Ethnic.group:Gender  1 276.75 0.8367 0.699
## - Ethnic.group:Village 1 277.11 1.1204 0.302
## - Gender:Village       2 276.47 1.1126 0.246
## - Age.group            1 277.97 1.8164 0.008 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: preference.matrix ~ Age.group + Ethnic.group + Gender + Village +      Ethnic.group:Village +
##
##              Df      AIC      F Pr(>F)
## - Ethnic.group:Village 1 275.94 0.9601 0.497
## - Gender:Village       4 273.96 1.0873 0.246
## - Age.group            1 276.84 1.7002 0.025 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: preference.matrix ~ Age.group + Ethnic.group + Gender + Village +      Gender:Village
##
##              Df      AIC      F Pr(>F)
## - Gender:Village     4 273.01 1.0782 0.282
## - Age.group          1 276.03 1.7388 0.014 *
## - Ethnic.group       1 277.25 2.7732 0.001 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: preference.matrix ~ Age.group + Ethnic.group + Gender + Village
##
##           Df      AIC      F Pr(>F)
## - Age.group   1 273.05 1.8192 0.013 *
## - Gender       1 273.01 1.7882 0.008 **
## - Village      2 272.74 1.6861 0.004 **
## - Ethnic.group 1 274.10 2.7791 0.001 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The selected model is preference.matrix ~ Age.group + Ethnic.group + Gender + Village ($R^2 = 0.232$; $R^2_{adj} = 0.14$)

Find a way to orient the CCA so that the Gando ethnic group is positive on all axes. Because the direction of each axis is arbitrary in CCA, this does not effect the interpretation of results. However, it ensures some consistency between different methods used during analysis or review. The choice of Gando was arbitrary, based on early results.

```
p.cca.orient <- diag(sign(p.cca$CCA$biplot["Ethnic.groupGan",])) %>%
  magrittr::set_rownames(colnames(p.cca$CCA$biplot)) %>%
  magrittr::set_colnames(colnames(p.cca$CCA$biplot))
```

Do per-axis ANOVA (Table M). This tests each axis sequentially.

```
set.seed(32195761)
p.anova <- anova(p.cca, by = "axis", permutations = 9999, cutoff = 0.05) %>%
  broom::tidy()
```

```
## Warning in tidy.anova(.): The following column names in ANOVA output were not
## recognized or transformed: ChiSquare
```

```
p.anova
```

Table M: ANOVA table for CCA axes of preference CCA.

term	df	ChiSquare	statistic	p.value
CCA1	1	0.2118260	5.8484199	0.0001
CCA2	1	0.1168735	3.2268249	0.0004
CCA3	1	0.0975981	2.6946387	0.0027
CCA4	1	0.0834703	2.3045784	0.0122
CCA5	1	0.0608335	1.6795852	0.1536
CCA6	1	0.0416862	1.1509384	NA
CCA7	1	0.0217567	0.6006917	NA
Residual	58	2.1007223	NA	NA

Choose axes which were significant at $p \leq 0.05$.

```
p.choices <- which(p.anova$p.value <= 0.05)
p.cca.orient <- p.cca.orient[p.choices, p.choices]
```

Do marginal per-term ANOVA (Table N). This tests the full model vs. the model with each individual term removed.

```
set.seed(32195761)
anova(p.cca, by = "margin", permutations = 9999) %>%
  broom::tidy()
```

```
## Warning in tidy.anova(.): The following column names in ANOVA output were not
## recognized or transformed: ChiSquare
```

Table N: **ANOVA table for explanatory variables in preference CCA.**

term	df	ChiSquare	statistic	p.value
Age.group	1	0.0658918	1.819242	0.0109
Ethnic.group	1	0.1006559	2.779064	0.0001
Gender	1	0.0647662	1.788166	0.0054
Village	2	0.1221361	1.686061	0.0019
Residual	58	2.1007223	NA	NA

Display the projections of the predictors on the significant axes, after reflection to “standard” orientation with Gando ethnicity positive (Table O).

```
scores(  
  p.cca,  
  display = "cn",  
  choices = p.choices  
) %>%  
  magrittr::multiply_by_matrix(p.cca.orient) %>%  
  tibble::as_tibble(rownames = "Category")
```

Table O: Centroids of predictor values for preference CCA.

Category	CCA1	CCA2	CCA3	CCA4
Age.group<35	0.7667950	0.5441357	0.8252103	-0.6386825
Age.group35+	-0.1374195	-0.0975161	-0.1478883	0.1144601
Ethnic.groupBar	1.0434631	-0.5161519	-0.5364880	0.0382995
Ethnic.groupGan	0.9451329	1.5245661	1.0722695	0.2794563
Ethnic.groupLok	-0.9964760	1.3651032	-1.4433138	-0.1893467
Ethnic.groupYom	-0.8753114	-0.6536537	0.6086067	-0.0662598
GenderF	0.1043842	-0.1936648	0.0027108	0.2827459
GenderM	-0.1472796	0.2732490	-0.0038247	-0.3989367
VillageAng	-0.6345110	-0.4514018	0.1111425	0.2098917
VillageBio	-1.1298566	0.1916357	-0.0153724	-0.3502499
VillageFab	0.8457558	-0.2108519	-0.1473823	1.2859117
VillageSon	1.2722989	-0.8695204	-0.9868571	-1.4057452

Set up some additional variables for plotting the preference CCA.

```
# coordinates for each interviewee ("site") and each species.
p.scores <- vegan::scores(
  p.cca,
  display = c("sites", "species"),
  scaling = "symmetric",
  choices = p.choices
)
# reflect into standard orientation
p.scores$sites <- p.scores$sites %*% p.cca.orient
p.scores$species <- p.scores$species %*% p.cca.orient

# scores for interviewees ("sites")
p.people <- p.scores$sites %>%
  tibble::as_tibble(rownames = "ID") %>%
  dplyr::left_join(preference.biodata, by = "ID")

# calculate species scores
p.species <- p.scores$species %>%
  tibble::as_tibble(rownames = "sp") %>%
  dplyr::mutate(
    length12 = sqrt(CCA1*CCA1 + CCA2*CCA2),
    length13 = sqrt(CCA3*CCA3 + CCA1*CCA1),
    length34 = sqrt(CCA3*CCA3 + CCA4*CCA4)
  )

# Choose the top 10 most extreme species in each axis combination.
p.species12 <- p.species %>%
  dplyr::arrange(dplyr::desc(length12)) %>%
  dplyr::mutate(sp = ifelse(dplyr::cur_group_rows() <= 10, sp, ""))

p.species13 <- p.species %>%
```

```

dplyr::arrange(dplyr::desc(length13)) %>%
dplyr::mutate(sp = ifelse(dplyr::cur_group_rows() <= 10, sp, ""))

p.species34 <- p.species %>%
dplyr::arrange(dplyr::desc(length34)) %>%
dplyr::mutate(sp = ifelse(dplyr::cur_group_rows() <= 10, sp, ""))

# Species abbreviation text to include in captions
p.species12.abbrev <-
dplyr::filter(k.species12, sp != "") %>%
dplyr::arrange(sp) %>%
dplyr::left_join(abbrevkey, by = c("sp" = "Abbrev")) %>%
glue::glue_data("{sp} = {canonicalName}") %>%
glue::glue_collapse(", ")

p.species13.abbrev <-
dplyr::filter(p.species13, sp != "") %>%
dplyr::arrange(sp) %>%
dplyr::left_join(abbrevkey, by = c("sp" = "Abbrev")) %>%
glue::glue_data("{sp} = {canonicalName}") %>%
glue::glue_collapse(", ")

p.species34.abbrev <-
dplyr::filter(p.species34, sp != "") %>%
dplyr::arrange(sp) %>%
dplyr::left_join(abbrevkey, by = c("sp" = "Abbrev")) %>%
glue::glue_data("{sp} = {canonicalName}") %>%
glue::glue_collapse(", ")

```

Show biplot for the first two axes, with sidebar for Ethnic group/Village combinations (Fig J).

```

p.cca12 <-
p.people %>%
ggplot(aes(
  x = CCA1,
  y = CCA2,
  color = VillageGroup,
  shape = VillageGroup,
  fill = VillageGroup
)) +
# light axes
geom_hline(yintercept = 0, color = "grey80") +
geom_vline(xintercept = 0, color = "grey80") +
# points for "sites" (i.e. interviewees)
geom_point(stroke = 1.5, alpha = 0.8) +
# points for species
geom_point(
  aes(x = CCA1, y = CCA2),
  data = p.species12,
  inherit.aes = FALSE,
  shape = 1
)

```

```

) +
# labels for the species
ggrepel::geom_text_repel(
  aes(CCA1, CCA2, label = sp),
  data = dplyr::bind_rows(
    p.species12,
    dplyr::mutate(p.people, sp = "")
  ),
  size = 3,
  inherit.aes = FALSE,
  box.padding = 0.5,
  point.padding = 0.2,
  segment.alpha = 0.2,
  min.segment.length = 0
) +
theme_cca_main +
scale_villagegroup

p.bg <- p.people %>%
  dplyr::select(dplyr::starts_with("CCA"), Ethnic.group) %>%
  dplyr::mutate(EG = TRUE) %>%
  tidyr::spread(Ethnic.group, EG, fill = FALSE) %>%
  dplyr::mutate_at(c("Bar", "Gan", "Lok", "Yom"), `!`) %>%
  tidyr::gather(Ethnic.group, EG, Bar:Yom) %>%
  dplyr::filter(EG)

p.cca12_side <- p.people %>%
  ggplot(aes(CCA1, CCA2, shape = Village, color = Village)) +
  geom_hline(yintercept = 0, color = "grey80") +
  geom_vline(xintercept = 0, color = "grey80") +
  geom_point(aes(CCA1, CCA2), color = "gray50", shape = ".",
    data = p.bg, inherit.aes = FALSE) +
  geom_point() +
  theme_cca_side +
  scale_village

ggpubr::ggarrange(p.cca12, p.cca12_side, widths = c(2.05, 1.13))

```

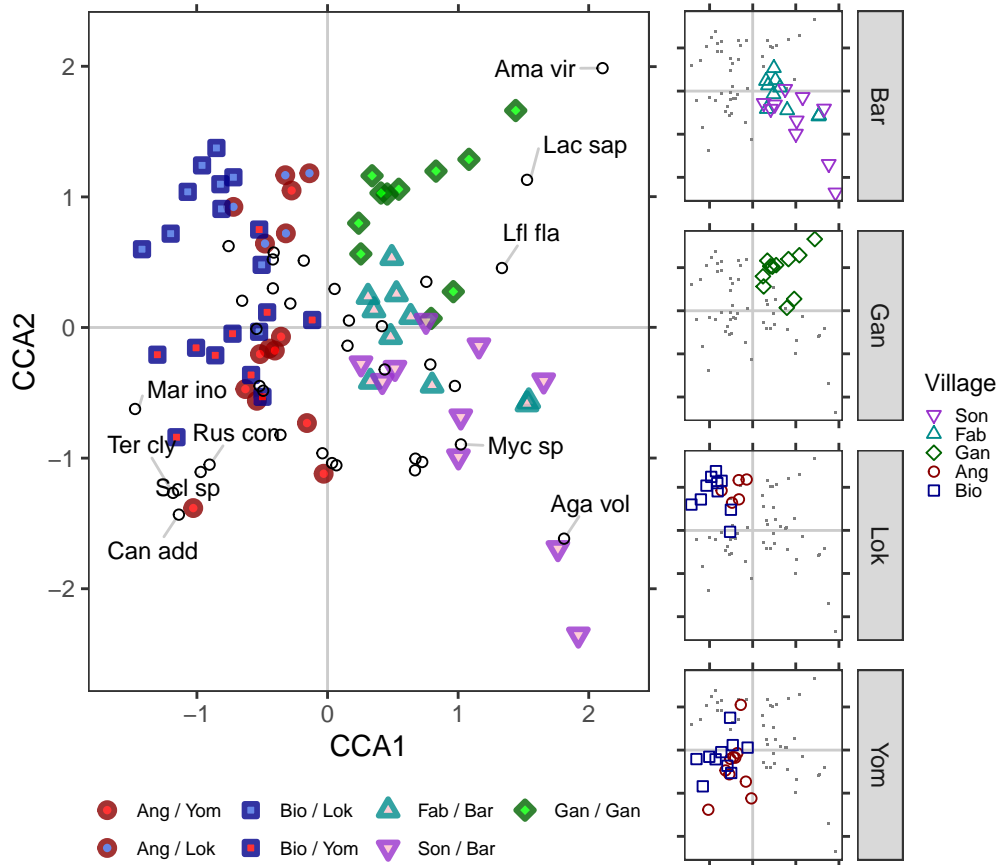


Fig J: **Biplot of CCA axes 1 and 2 for species preferences.** Individual respondents are shown as colored shapes, while mushroom species empty circles colored shapes. Only the ten mushroom species with the strongest axis associations are labeled. Village abbreviations: Ang = Angaradebou, Bio = Bio Sika, Fab = Faba, Gan = Gando, Son = Sonnoumon. Ethnic group abbreviations: Bar = Bariba, Gan = Gando, Lok = Lokpa, Yom = Yom. Species abbreviations: Afr lut = *Afroboletus luteolus*, Aga vol = *Agaricus volvatulus*, Ama crc = *Amanita* “crassiconus”, Ama stro = *Amanita strobilaceovolvata*, Ama vir = *Amanita* “virido-odorata”, Can pla = *Cantharellus platyphyllus*, Lac sap = *Lactarius saponaceus*, Lfl fla = *Lactifluus flammans*, Mar ino = *Marasmiellus inoderma*, Myc sp = *Mycoamaranthus* sp.. Side plots: respondents for each ethnic group are plotted separately, with point shape and color representing village.

Show the biplot for CCA axes 3 and 4, again with Ethnic Group/Village sidebar (Fig K).

```
p.cca34 <-
p.people %>%
  ggplot(aes(
    x = CCA3,
    y = CCA4,
    color = VillageGroup,
    shape = VillageGroup,
    fill = VillageGroup
  )) +
  # light axes
  geom_hline(yintercept = 0, color = "grey80") +
  geom_vline(xintercept = 0, color = "grey80") +
  # points for "sites" (i.e. interviewees)
  geom_point(stroke = 1.5, alpha = 0.8) +
  # points for species
  geom_point(
    aes(x = CCA3, y = CCA4),
    data = p.species34,
    inherit.aes = FALSE,
    shape = 1
  ) +
  # labels for the species
  ggrepel::geom_text_repel(
    aes(CCA3, CCA4, label = sp),
    data = dplyr::bind_rows(
      p.species34,
      dplyr::mutate(p.people, sp = "")
    ),
    size = 3,
    inherit.aes = FALSE,
    box.padding = 0.5,
    point.padding = 0.2,
    segment.alpha = 0.2,
    min.segment.length = 0
  ) +
  theme_cca_main +
  scale_villagegroup

p.cca34_side <- p.people %>%
  ggplot(aes(CCA3, CCA4, shape = Village, color = Village)) +
  geom_hline(yintercept = 0, color = "grey80") +
  geom_vline(xintercept = 0, color = "grey80") +
  geom_point(aes(CCA3, CCA4), color = "gray50", shape = ".", data = p.bg, inherit.aes = FALSE) +
  geom_point() +
  theme_cca_side +
  scale_village

ggpubr::ggarrange(p.cca34, p.cca34_side, widths = c(2.06, 1))
```

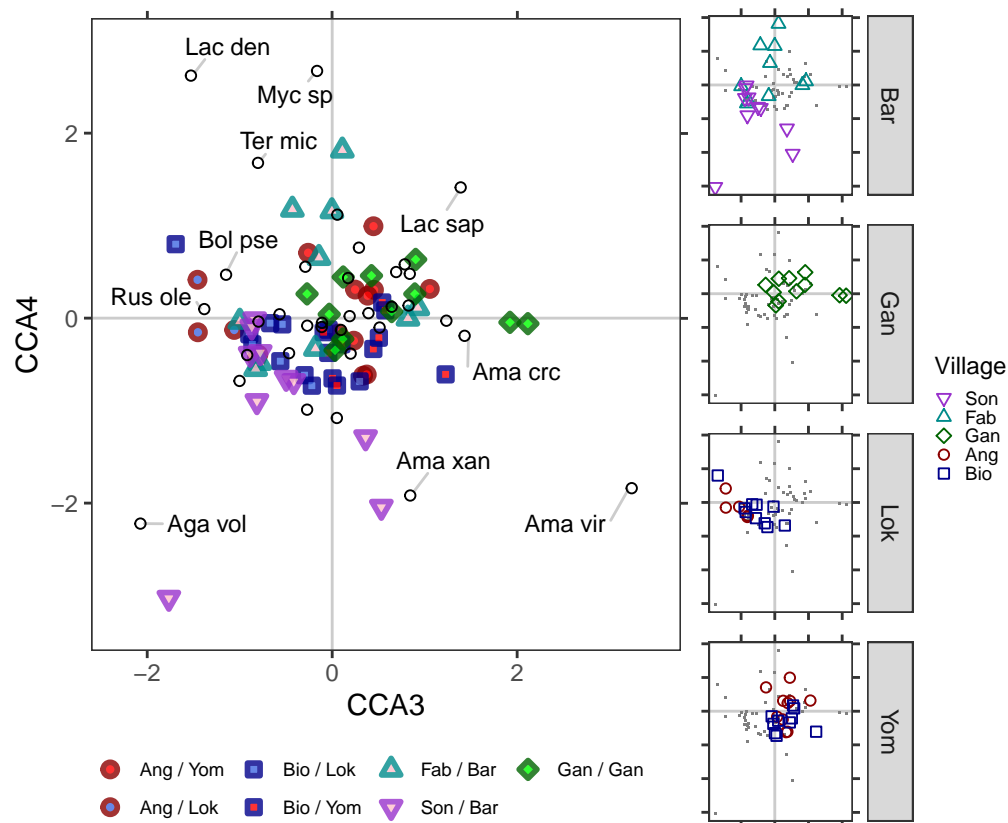


Fig K: **Biplot of CCA axes 3 and 4 for species preferences.** Individual respondents are shown as colored shapes, while mushroom species empty circles colored shapes. Only the ten mushroom species with the strongest axis associations are labeled. Village abbreviations: Ang = Angaradebou, Bio = Bio Sika, Fab = Faba, Gan = Gando, Son = Sonnoumon. Ethnic group abbreviations: Bar = Bariba, Gan = Gando, Lok = Lokpa, Yom = Yom. Species abbreviations: Aga vol = *Agaricus volvatulus*, Ama crc = *Amanita* “crassiconus”, Ama vir = *Amanita* “virido-odorata”, Ama xan = *Amanita xanthogala*, Bol pse = *Boletus pseudoloosii*, Lac den = *Lactarius denigricans*, Lac sap = *Lactarius saponaceus*, Myc sp = *Mycoamaranthus* sp., Rus ole = *Russula oleifera*, Ter mic = *Termitomyces microcarpus*. Side plots: respondents for each ethnic group are plotted separately, with point shape and color representing village.

5 Knowledge exchange within villages

The Lokpa and Yom ethnic groups both inhabit the villages of Angaradebou and Bio Sika. The Yom make up the demographic majority in Angaradebou, and the Lokpa make up the demographic majority in Bio Sika.

Here we compare the differences in the profile of species known between pairs of interviewees who belong to:

1. The same village and same ethnic group (AngLok / AngLok; AngYom / AngYom; BioLok / BioLok; BioYom / BioYom)
 2. The same village but different ethnic groups (AngLok / AngYom; BioLok / BioYom)
 3. Different villages but the same ethnic group (AngLok / BioLok; AngYom / BioYom)
 4. Different villages and ethnic groups, each the minority in their respective village (AngLok / BioYom)
 5. Different villages and ethnic groups, each the majority in their respective village (AngYom / BioLok)
-

Put together the data and do a Kruskal-Wallis test.

```
k_lokyom_pairwise <- k.people %>%  
  # choose only Lokpa and Yom,  
  # because these are the groups that inhabit the same villages.  
  dplyr::filter(Ethnic.group %in% c("Lok", "Yom")) %>%  
  dplyr::select(VillageGroup, Ethnic.group, Village, CCA1:CCA3) %>%  
  # Yom are the majority in Anagaradebou, Lokpa are the majority in Bio Sika  
  dplyr::mutate(  
    ID = seq_len(nrow(.)),  
    prevalence = dplyr::case_when(  
      Village == "Ang" & Ethnic.group == "Yom" ~ "Majority",  
      Village == "Bio" & Ethnic.group == "Lok" ~ "Majority",  
      TRUE ~ "Minority"  
    )  
  ) %>%  
  # make all unique pairs  
  tidyr::crossing(A = ., B = .) %>%  
  tidyr::unpack(c(A, B), names_sep = "_") %>%  
  dplyr::filter(A_ID > B_ID) %>%  
  dplyr::mutate(  
    # calculate pairwise distances  
    dist = sqrt(  
      (A_CCA1 - B_CCA1)^2 +  
      (A_CCA2 - B_CCA2)^2 +  
      (A_CCA3 - B_CCA3)^2  
    ),  
    # categorize comparisons  
    type = dplyr::case_when(  
      A_Village == B_Village & A_Ethnic.group == B_Ethnic.group ~  
        "Same",  
      A_Village == B_Village & A_Ethnic.group != B_Ethnic.group ~  
        "Diff Ethn",
```

```

    A_Village != B_Village & A_Ethnic.group == B_Ethnic.group ~
      "Diff Vill",
    TRUE ~ A_prevalence
  ),
  type = factor(
    type,
    c("Same", "Diff Vill", "Diff Ethn", "Minority", "Majority")
  )
)

k_lokyom_kruskal <- k_lokyom_pairwise %$%
  agricolae::kruskal(y = dist, trt = type, p.adj = "holm")

k_lokyom_kruskal$statistics

##      Chisq p.chisq
## 115.5213      0

```

Plot the distribution of pairwise distances within each category, along with the K-W test results (Fig ??).

```

k_lokyom_groups <-
  dplyr::left_join(
    # kruskal-wallis groupings
    k_lokyom_kruskal$groups %>%
      tibble::rownames_to_column("type") %>%
      dplyr::select(-dist),
    # max value for each group (for plotting)
    dplyr::select(k_lokyom_pairwise, type, dist) %>%
      dplyr::group_by(type) %>%
      dplyr::summarize_all(max),
    by = "type"
  )
ggplot(k_lokyom_pairwise, aes(x = type, y = dist)) +
  geom_boxplot() +
  geom_text(
    aes(x = type, y = dist, label = groups),
    data = k_lokyom_groups,
    inherit.aes = FALSE,
    nudge_y = 0.2
  ) +
  xlab(NULL) +
  ylab("CCA distance")

```

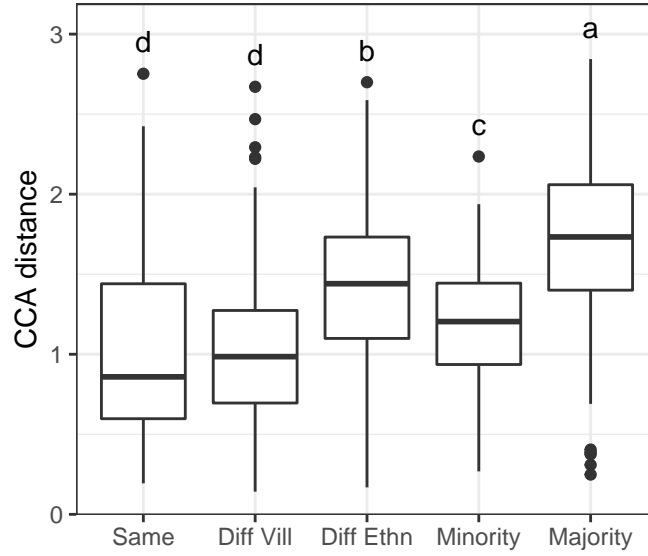


Fig L: **Pairwise knowledge CCA differences between Lokpa and Yom respondents in Anga-radebou and Bio Sika villages.** (Fig 5 in main text.) Distance represents dissimilarity in the profile of which mushroom species are known, not necessarily a difference in the absolute number of species known. Categories are *Same*: interviewees belonging to the same ethnic group, living in the same village; *Diff Ethn*: interviewees belonging to different ethnic groups, but living in the same village; *Diff Vill*: interviewees belonging to the same ethnic group, but living in different villages; *Minority*: interviewees belonging to different ethnic groups and living in different villages, each in the minority ethnic group in their respective village; *Majority*: interviewees belonging to different ethnic groups and living in different villages, each in the majority ethnic group in their respective village.

Now we repeat the same analysis, based on preferences rather than the set of species known.

```
p_lokyom_pairwise <- p.people %>%
  # choose only Lokpa and Yom,
  # because these are the groups that inhabit the same villages.
  dplyr::filter(Ethnic.group %in% c("Lok", "Yom")) %>%
  dplyr::select(VillageGroup, Ethnic.group, Village, CCA1:CCA4) %>%
  # Yom are the majority in Anagaradebou, Lokpa are the majority in Bio Sika
  dplyr::mutate(
    ID = seq_len(nrow()),
    prevalence = dplyr::case_when(
      Village == "Ang" & Ethnic.group == "Yom" ~ "Majority",
      Village == "Bio" & Ethnic.group == "Lok" ~ "Majority",
      TRUE ~ "Minority"
    )
  ) %>%
  # make all unique pairs
  tidyr::crossing(A = ., B = .) %>%
  tidyr::unpack(c(A, B), names_sep = "_") %>%
  dplyr::filter(A_ID > B_ID) %>%
  dplyr::mutate(
    # calculate pairwise distances
    dist = sqrt(
      (A_CCA1 - B_CCA1)^2 +
      (A_CCA2 - B_CCA2)^2 +
      (A_CCA3 - B_CCA3)^2 +
      (A_CCA4 - B_CCA4)^2
    ),
    # categorize comparisons
    type = dplyr::case_when(
      A_Village == B_Village & A_Ethnic.group == B_Ethnic.group ~
        "Same",
      A_Village == B_Village & A_Ethnic.group != B_Ethnic.group ~
        "Diff Ethn",
      A_Village != B_Village & A_Ethnic.group == B_Ethnic.group ~
        "Diff Vill",
      TRUE ~ A_prevalence
    ),
    type = factor(
      type,
      c("Same", "Diff Vill", "Diff Ethn", "Minority", "Majority")
    )
  )

p_lokyom_kruskal <- p_lokyom_pairwise %$%
  agricolae::kruskal(y = dist, trt = type, p.adj = "holm")

p_lokyom_kruskal$statistics
```

```
##      Chisq p.chisq
## 226.089      0
```

Plot the dissimilarities and K-W results (Fig M).

```
p_lokyom_groups <-
  dplyr::left_join(
    # kruskal-wallis groupings
    p_lokyom_kruskal$groups %>%
      tibble::rownames_to_column("type") %>%
      dplyr::select(-dist),
    # max value for each group (for plotting)
    dplyr::select(p_lokyom_pairwise, type, dist) %>%
      dplyr::group_by(type) %>%
      dplyr::summarize_all(max),
    by = "type"
  )
ggplot(p_lokyom_pairwise, aes(x = type, y = dist)) +
  geom_boxplot() +
  geom_text(
    aes(x = type, y = dist, label = groups),
    data = p_lokyom_groups,
    inherit.aes = FALSE,
    nudge_y = 0.2
  ) +
  xlab(NULL) +
  ylab("CCA distance")
```

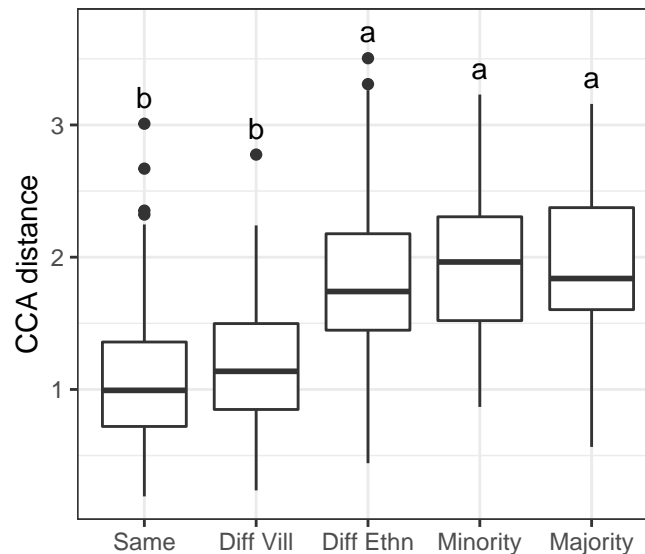


Fig M: **Pairwise CCA differences between Lokpa and Yom respondents in Angaradebou and Bio Sika villages.** Distance represents dissimilarity in the preference for different mushroom species. Categories are *Same*: interviewees belonging to the same ethnic group, living in the same village; *Diff Ethn*: interviewees belonging to different ethnic groups, but living in the same village; *Diff Vill*: interviewees belonging to the same ethnic group, but living in different villages; *Minority*: interviewees belonging to different ethnic groups and living in different villages, each in the minority ethnic group in their respective village; *Majority*: interviewees belonging to different ethnic groups and living in different villages, each in the majority ethnic group in their respective village.

6 Edible species

Make a list of species commonly regarded as edible, and information about the preference values each was given (Table P).

```
interviews %>%
  dplyr::group_by(scientificName) %>%
  dplyr::filter(!is.na(Preference)) %>%
  dplyr::summarize(
    n = dplyr::n(),
    min = min(Preference),
    q25 = quantile(Preference, 0.25),
    mean = mean(Preference),
    median = median(Preference),
    q75 = quantile(Preference, 0.75),
    max = max(Preference),
    toxic = sum(stringr::str_detect(Remarks, "[Tt]oxic"), na.rm = TRUE)
  ) %>%
  dplyr::filter(n >= 5, median >= 2, toxic < 2) %>%
  dplyr::select(Species = scientificName, n, min, q25, mean, median, q75, max) %>%
  dplyr::arrange(dplyr::desc(mean))
```


Table P: **Species regarded as edible.** Includes all species with at least 5 appraisals, median preference value ≥ 2 , and regarded as toxic by < 2 interviewees. *n*: number of preference appraisals; *min*: minimum preference value; *q25*: first quartile preference appraisal; *mean*: mean preference appraisal; *median*: median preference appraisal; *q75*: third quartile preference appraisal; *max*: max preference appraisal.

Species	n	min	q25	mean	median	q75	max
Termitomyces schimperi (Pat.) R. Heim, 1942	27	2	3.0	3.592593	4.0	4.00	4
Lactifluus edulis (Verbeken & Buyck) Buyck, 2011	22	0	3.0	3.227273	4.0	4.00	4
Macrocybe lobayensis (R. Heim) Pegler & Lodge, 1998	9	2	3.0	3.222222	3.0	4.00	4
Chlorophyllum palaeotropicum Z.W. Ge & A. Jacobs 2018	40	2	3.0	3.200000	3.0	4.00	4
Psathyrella tuberculata (Pat.) A.H. Sm., 1972	54	0	3.0	3.185185	3.0	4.00	4
Termitomyces le-testui (Pat.) R. Heim, 1942	39	2	3.0	3.153846	3.0	4.00	4
Lactifluus flammans (Verbeken) Verbeken, 2012	38	0	2.0	2.842105	3.0	4.00	4
Amanita strobilaceovolvata Beeli, 1935	8	1	2.0	2.750000	3.0	3.25	4
Termitomyces fuliginosus R. Heim, 1942	25	0	2.0	2.680000	3.0	3.00	4
Lentinus squarrosulus Mont., 1842	52	0	2.0	2.538461	3.0	3.00	4
Boletus pseudoloosii De Kesel, Codjia & Yourou	6	1	2.0	2.500000	2.0	3.50	4
Russula oleifera Buyck, 1990	8	0	2.0	2.375000	2.5	3.00	4
Cantharellus platyphyllus Heinem., 1966	14	1	2.0	2.357143	2.0	2.00	4
Lactarius tenellus Verbeken & Walley, 2000	17	0	2.0	2.294118	3.0	3.00	4
Pleurotus cystidiosus O.K. Mill., 1969	35	0	1.5	2.285714	3.0	3.00	4
Amanita masasiensis Härk. & Saarim., 1994	34	0	1.0	2.117647	2.0	3.00	4
Termitomyces clypeatus R. Heim, 1951	9	0	0.0	2.111111	3.0	3.00	4
Lactifluus gymnocarpoides (Verbeken) Verbeken, 2012	32	0	2.0	2.093750	2.0	3.00	4
Amanita subviscosa Beeli, 1931	27	0	1.5	2.074074	2.0	3.00	3
Amanita craseoderma Bas, 1978	27	0	1.0	2.037037	2.0	3.00	4
Lentinus tuber-regium (Fr.) Fr., 1836	25	0	1.0	1.920000	2.0	3.00	4
Russula congoana Pat., 1914	18	0	1.0	1.777778	2.0	3.00	4
Lactarius saponaceus Verbeken, 1996	6	0	0.5	1.666667	2.0	2.00	4

Package versions

Show version numbers of R and key packages that were used.

R:

```
R.version.string  
citation("base")
```

```
## [1] "R version 3.5.1 (2018-07-02)"  
##  
## To cite R in publications use:  
##  
## R Core Team (2018). R: A language and environment for statistical  
## computing. R Foundation for Statistical Computing, Vienna, Austria.  
## URL https://www.R-project.org/.  
##  
## A BibTeX entry for LaTeX users is  
##  
## @Manual{,  
## title = {R: A Language and Environment for Statistical Computing},  
## author = {{R Core Team}},  
## organization = {R Foundation for Statistical Computing},  
## address = {Vienna, Austria},  
## year = {2018},  
## url = {https://www.R-project.org/},  
## }  
##  
## We have invested a lot of time and effort in creating R, please cite it  
## when using it for data analysis. See also 'citation("pkgname")' for  
## citing R packages.
```

vegan:

```
packageVersion("vegan")  
citation("vegan")
```

```
## [1] '2.5.4'  
##  
## To cite package 'vegan' in publications use:  
##  
## Jari Oksanen, F. Guillaume Blanchet, Michael Friendly, Roeland Kindt,  
## Pierre Legendre, Dan McGlinn, Peter R. Minchin, R. B. O'Hara, Gavin  
## L. Simpson, Peter Solymos, M. Henry H. Stevens, Eduard Szoecs and  
## Helene Wagner (2019). vegan: Community Ecology Package. R package  
## version 2.5-4. https://CRAN.R-project.org/package=vegan  
##  
## A BibTeX entry for LaTeX users is  
##  
## @Manual{,  
##
```

```
## title = {vegan: Community Ecology Package},
## author = {Jari Oksanen and F. Guillaume Blanchet and Michael Friendly and Roeland Kindt and Pier
## year = {2019},
## note = {R package version 2.5-4},
## url = {https://CRAN.R-project.org/package=vegan},
## }
##
## ATTENTION: This citation information has been auto-generated from the
## package DESCRIPTION file and may need manual editing, see
## 'help("citation")'.
```

emmeans:

```
packageVersion("emmeans")
citation("emmeans")
```

```
## [1] '1.3.4'
##
## To cite package 'emmeans' in publications use:
##
## Russell Lenth (2019). emmeans: Estimated Marginal Means, aka
## Least-Squares Means. R package version 1.3.4.
## https://CRAN.R-project.org/package=emmeans
##
## A BibTeX entry for LaTeX users is
##
## @Manual{,
## title = {emmeans: Estimated Marginal Means, aka Least-Squares Means},
## author = {Russell Lenth},
## year = {2019},
## note = {R package version 1.3.4},
## url = {https://CRAN.R-project.org/package=emmeans},
## }
```

car:

```
packageVersion("car")
citation("car")
```

```
## [1] '3.0.2'
##
## To cite the car package in publications use:
##
## John Fox and Sanford Weisberg (2011). An {R} Companion to Applied
## Regression, Second Edition. Thousand Oaks CA: Sage. URL:
## http://socserv.socsci.mcmaster.ca/jfox/Books/Companion
##
## A BibTeX entry for LaTeX users is
##
## @Book{,
## title = {An {R} Companion to Applied Regression},
## edition = {Second},
## author = {John Fox and Sanford Weisberg},
```

```
##   year = {2011},
##   publisher = {Sage},
##   address = {Thousand Oaks {CA}},
##   url = {http://socserv.socsci.mcmaster.ca/jfox/Books/Companion},
## }
```

agricolae:

```
packageVersion("agricolae")
citation("agricolae")
```

```
## [1] '1.3.1'
##
## To cite package 'agricolae' in publications use:
##
## Felipe de Mendiburu (2019). agricolae: Statistical Procedures for
## Agricultural Research. R package version 1.3-1.
## https://CRAN.R-project.org/package=agricolae
##
## A BibTeX entry for LaTeX users is
##
## @Manual{,
##   title = {agricolae: Statistical Procedures for Agricultural Research},
##   author = {Felipe {de Mendiburu}},
##   year = {2019},
##   note = {R package version 1.3-1},
##   url = {https://CRAN.R-project.org/package=agricolae},
## }
##
## ATTENTION: This citation information has been auto-generated from the
## package DESCRIPTION file and may need manual editing, see
## 'help("citation")'.
```

ggplot2:

```
packageVersion("ggplot2")
citation("ggplot2")
```

```
## [1] '3.3.0'
##
## To cite ggplot2 in publications, please use:
##
## H. Wickham. ggplot2: Elegant Graphics for Data Analysis.
## Springer-Verlag New York, 2016.
##
## A BibTeX entry for LaTeX users is
##
## @Book{,
##   author = {Hadley Wickham},
##   title = {ggplot2: Elegant Graphics for Data Analysis},
##   publisher = {Springer-Verlag New York},
##   year = {2016},
```

```
##      isbn = {978-3-319-24277-4},  
##      url = {https://ggplot2.tidyverse.org},  
##    }
```