Report 1: Shared plant knowledge

Pascal

2023-03-22

load packages and files

```
# set wd
setwd("~/Library/CloudStorage/OneDrive-Personal/Dokumente/BSc_UZH/UZH_23FS/BIO206/Reports/1")
know <- read.csv("plant_knowledge.csv")
people <- read.csv("plant_participants.csv")
```

1. Plant knowledge file

a) Dataset dimensions

```
print(paste("Subjects: ", nrow(know)))
```

```
## [1] "Subjects:  219"
```

```
print(paste("Plants: ", ncol(know) - 1))
```

```
## [1] "Plants:  33"
```

b) Sum of knowledge by plant

```
frame_know_plant <- data.frame(know_count = sort(colSums(know[,-1]), decreasing = T) )
print(frame_know_plant)
```

```
##             know_count
## Mobey              216
## Ekoka              215
## Mokakake           214
## Guka               213
## Banga              212
## Boyo               211
## Kombo              208
## Kokosa             201
## Kungu            199
## Imbanda          198
## Ngata            198
## Mongangai        196
## Jongo            195
## Indengo          189
## Bulaki           185
## Iboko            183
## Mosombo          183
## Moba             180
## Embondo          178
## Mokula           177
## Mongo            173
## Juese            171
## Mongamba         164
## Mopo             162
## Njobe            161
## Imbenya          149
## Mototoko         149
## Toko             146
## Muese            144
## Imbi             134
## Mokata           132
## Somboli          101
## Euey              92
```

```r
print(paste("Plant knowledge avg:", mean(frame_know_plant$know_count)))
```

```
## [1] "Plant knowledge avg: 176.636363636364"
```

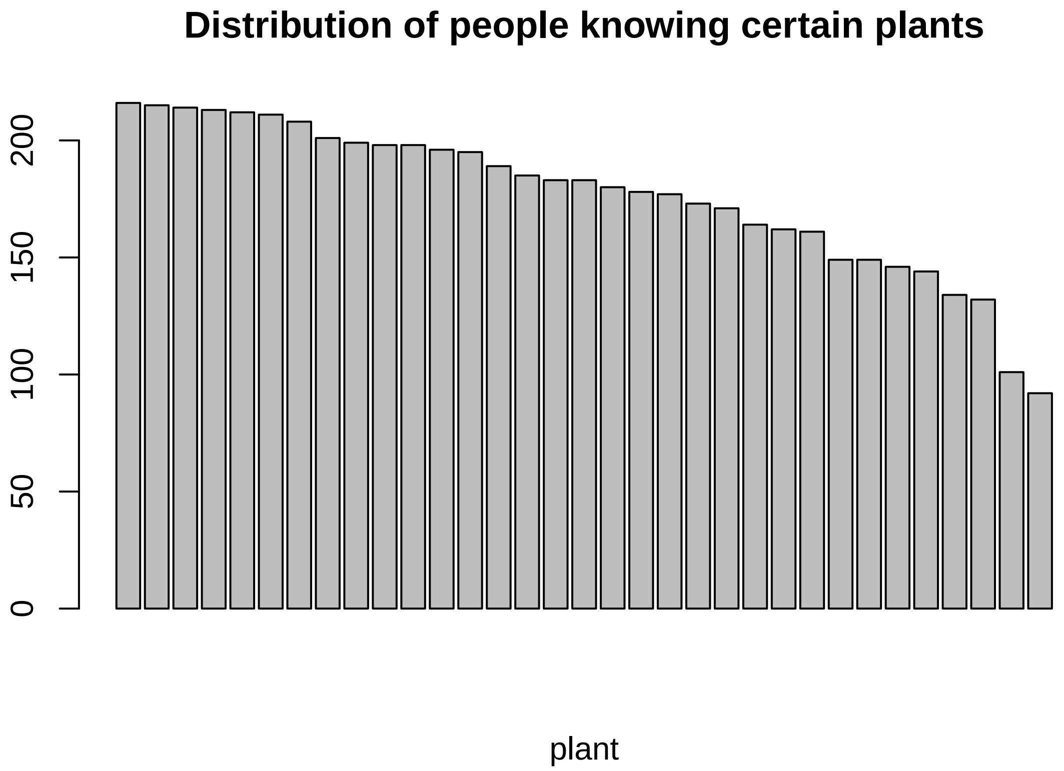
```r
print("Plants known by over 200 people:")
```

```
## [1] "Plants known by over 200 people:"
```

```r
print(rownames(frame_know_plant %>% filter(know_count > 200)))
```

```
## [1] "Mobey"    "Ekoka"    "Mokakake" "Guka"     "Banga"    "Boyo"     "Kombo"
## [8] "Kokosa"
```

```r
barplot(frame_know_plant$know_count,
     main = "Distribution of people knowing certain plants",
     xlab = "plant")
```



```
# ggplot(mapping = aes(x = sum_know_plant)) +
#   geom_histogram(bins = 8)
```

c) Plant knowledge by individual

```
frame_know_people <- data.frame(plant_know = sort(rowSums(know[,-1]), decreasing = T), age = people$age
rownames(frame_know_people) <- know$ID
print("ID and age of people knowing 33 plants:")
```

```
## [1] "ID and age of people knowing 33 plants:"
```

```
print(frame_know_people %>% filter(plant_know > 32))
```

```
##      plant_know   age
## M448         33 60-80
## M441         33 40-60
## M539         33 25-30
## M527         33 20-25
## M456         33 60-80
## M457         33 10-15
## M453         33 25-30
## M405         33 15-20
## M416         33 15-20
## M452         33 15-20
## M478         33 35-40
## M482         33 35-40
## M500         33 25-30
## M502         33 25-30
## M398         33 30-35
## M407         33 25-30
## M439         33 40-60
## M505         33 60-80
## M455         33 25-30
## M445         33 40-60
## M465         33 40-60
## M538         33 40-60
## M461         33 10-15
```

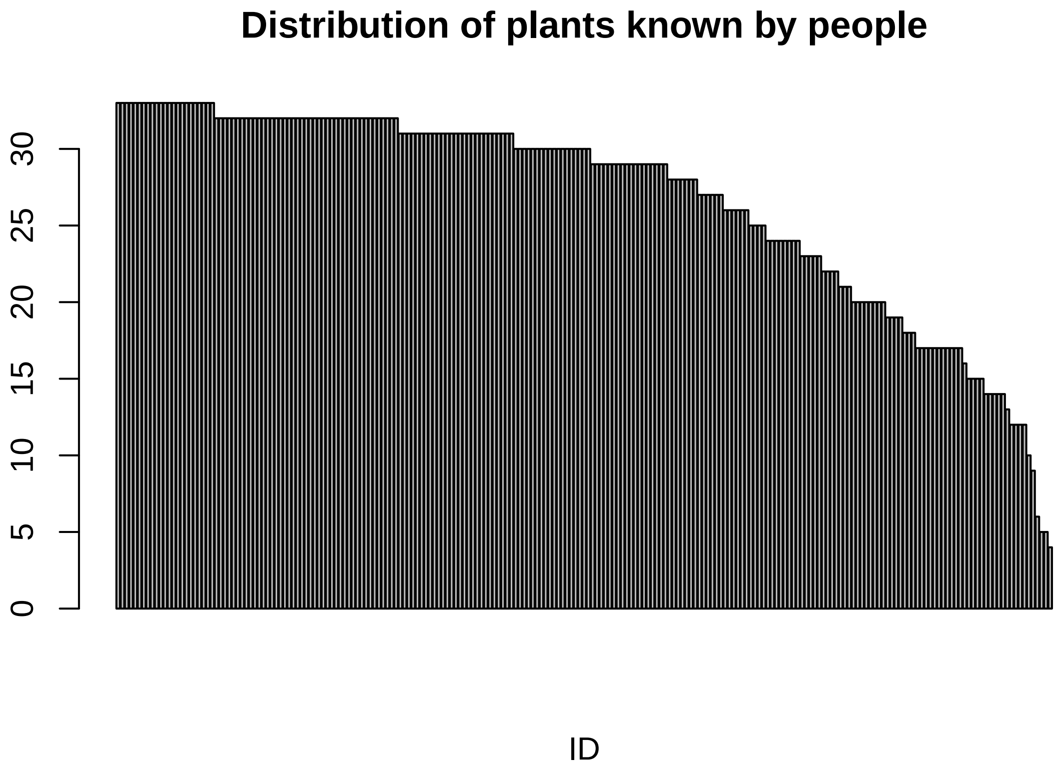
```
print("We see that the great plant knowledge is not bound to old people and varies among individuals")
```

```
## [1] "We see that the great plant knowledge is not bound to old people and varies among individuals"
```

```
print(paste("People knowledge avg:", mean(frame_know_people$plant_know)))
```

```
## [1] "People knowledge avg: 26.6164383561644"
```

```
barplot(frame_know_people$plant_know,
        main = "Distribution of plants known by people",
        xlab = "ID")
```



2. Plant participants file

a) age distribution

```
print(table(people$age)) # use count for digits and table for factors
```

```
##
## 05-10 10-15 15-20 20-25 25-30 30-35 35-40 40-60 60-80
##     7    15    26    29    30    24    23    41    21
```

b) fraction of pre-adults

```
count_pre_20 <- people %>% filter(age == "05-10" | age == "10-15") %>% nrow(.)
print(count_pre_20 / nrow(people))
```

```
## [1] 0.1004566
```

c) sex ratio

```
count_males <- people %>% filter(sex == "M") %>% nrow(.)
count_females <-  people %>% filter(sex == "F") %>% nrow(.)
print(paste("Sex ratio M/F:", count_males / count_females))
```

```
## [1] "Sex ratio M/F: 0.831932773109244"
```

3. Dyads

merge files to a dyad frame

```
# create dyads
dyads = data.frame(t(combn(people$ID, 2)))
# dim of resulting frame
no_dyads = length(people$ID)* (length(people$ID)-1)/2
no_plants = ncol(know) - 1
# add dyad names
dyads$dyad_ID = paste(dyads$X1, dyads$X2, sep ="_")
colnames(dyads) = c("ID1", "ID2", "dyad_ID")

# merge with plant knowledge
dyads_people <- merge(dyads, people, by.x = "ID2", by.y = "ID")
dyads_people <- merge(dyads_people, people ,by.x = "ID1", by.y = "ID")


dyads_people$dyadsex <- paste0(pmin(dyads_people$sex.x, dyads_people$sex.y),
                               pmax(dyads_people$sex.x, dyads_people$sex.y))

dyads_people$samesex <- ifelse(dyads_people$dyadsex == "FM", 1, 0) # ifelse to see where they match

know <- know %>% gather("plant", "know", -ID)    # switch from wide to long format



dyads_merged <- merge(dyads_people, know, by.x = "ID1", by.y = "ID")
dyads_merged <- merge(dyads_merged, know, by.x = c("ID2", "plant"), by.y = c("ID", "plant"))

print(paste("Check dimension of frame:", nrow(dyads_merged) == no_dyads * no_plants))
```

```
## [1] "Check dimension of frame: TRUE"
```

```
dyads_merged <- dyads_merged %>%
  select(c(3, 1, 4, 5, 11, 6, 12, 18, 17, 7, 13, 8, 14, 9, 15, 2, 19, 20, 10, 16)) %>%
  rename_with(~gsub(".x","1", .x, fixed = T)) %>%
  rename_with(~gsub(".y","2", .x, fixed = T))
```

a) dyad sex

```
table((dyads_merged %>% filter(dyadsex != "NANA"))$dyadsex) / no_plants
```

```
##
##    FF    FM    MM
##  7021 11781  4851
```

b) dyad age

young

```
no_young <- dyads_merged %>%
  filter((age1 == "05-10" & age2 == "05-10") |
          (age1 == "05-10" & age2 == "10-15") |
          (age1 == "10-15" & age2 == "05-10") |
          (age1 == "10-15" & age2 == "10-15")) %>%
  nrow(.)
print(paste("Young:", no_young / no_plants))
```

```
## [1] "Young: 231"
```

old

```
no_old <- dyads_merged %>%
  filter(age1 == "60-80" & age2 == "60-80") %>%
  nrow(.)
print(paste("old:", no_old / no_plants))
```

```
## [1] "old: 210"
```

c) dyad born

```
# account for NAs!
dyads_merged$sameborn <- ifelse(dyads_merged$born1 == dyads_merged$born2 &
                                  !is.na(dyads_merged$born1) &
                                  !is.na(dyads_merged$born2), 1, 0)

dyads_merged$diffborn <- ifelse(dyads_merged$born1 != dyads_merged$born2 &
                                  !is.na(dyads_merged$born1) &
                                  !is.na(dyads_merged$born2), 1, 0)
# omit nas otherwise na comparison
print(paste("Born in same camp:", sum(dyads_merged$sameborn) / 33))
```

```
## [1] "Born in same camp: 3638"
```

```
print(paste("Born in different camp:", sum(dyads_merged$diffborn) / 33))
```

```
## [1] "Born in different camp: 15083"
```

d) dyad camp

```
samecamp <- ifelse(dyads_merged$camp1 == dyads_merged$camp2 &
                    !is.na(dyads_merged$camp1) &
                    !is.na(dyads_merged$camp2), 1, 0)

diffcamp <- ifelse(dyads_merged$camp1 != dyads_merged$camp2 &
                    !is.na(dyads_merged$camp1) &
                    !is.na(dyads_merged$camp2), 1, 0)
# omit nas otherwise na comparison
print(paste("Interviewed in same camp", sum(samecamp) / 33))
```

```
## [1] "Interviewed in same camp 6215"
```

```
print(paste("Interviewed in different camp", sum(diffcamp) / 33))
```

```
## [1] "Interviewed in different camp 17656"
```

4) Total knowledge score

shared knowledge column

```
dyads_merged$dyadknow <- ifelse(dyads_merged$know1 == 1 & dyads_merged$know2 == 1, 1, 0)
```

a) age

create levels and check distribution

```
dyads_merged$dyadagelevels <- ifelse((dyads_merged$age1 == "05-10" & dyads_merged$age2 == "05-10") |
                                     (dyads_merged$age1 == "05-10" & dyads_merged$age2 == "10-15") |
                                     (dyads_merged$age1 == "10-15" & dyads_merged$age2 == "05-10") |
                                     (dyads_merged$age1 == "10-15" & dyads_merged$age2 == "10-15"),
                                   "young",
                                   ifelse(dyads_merged$age1 == "60-80" & dyads_merged$age2 == "60-80",
                                          "old", ifelse(is.na(dyads_merged$age1) | is.na(dyads_merged$ag
                                                        "others", "others")))
# dyads_merged$dyadagelevels <- as.factor(dyads_merged$dyadagelevels)
print(table(dyads_merged$dyadagelevels))
```

```
##
##    old others  young
##   6930 768834   7623
```

dyplyr summary and Kruskal-Walis test

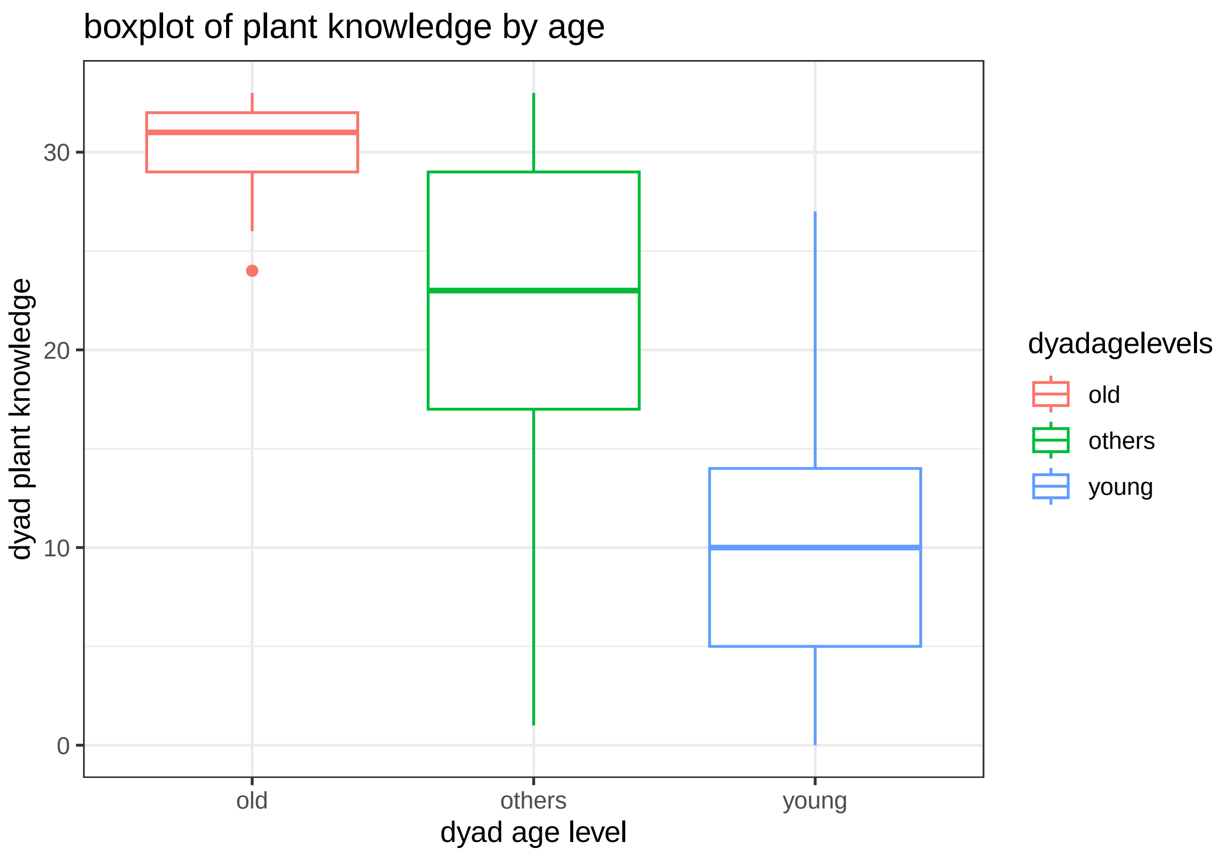
```r
know_age <- dyads_merged %>%
  filter(!is.na(dyadagelevels)) %>%
  group_by(dyad_ID, dyadagelevels) %>%
  summarise(sum_know = sum(dyadknow), n= n()) # %>%
  # group_by(dyadagelevels) %>%
  # summarise(mean_know = mean(sum_know))

know_age <- data.frame(know_age)
kruskal.test(sum_know ~ dyadagelevels, data = know_age)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  sum_know by dyadagelevels
## Kruskal-Wallis chi-squared = 731.73, df = 2, p-value < 2.2e-16
```

boxplot

```r
ggplot() +
  geom_boxplot(data = know_age, mapping = aes(x = dyadagelevels, y = sum_know, colour = dyadagelevels))
  labs(x = "dyad age level", y = "dyad plant knowledge",
       title = "boxplot of plant knowledge by age") +
  theme_bw()
```



### Interpretation

From the p-value in the Kruskal-Walis test one sees that age levels are a valid predictor for total plant knowledge. Furthermore, the boxplot implies that age positively corelates with plant knowledge, meaning that older people have more knowledge.

b) sex

Kruskal Walis test

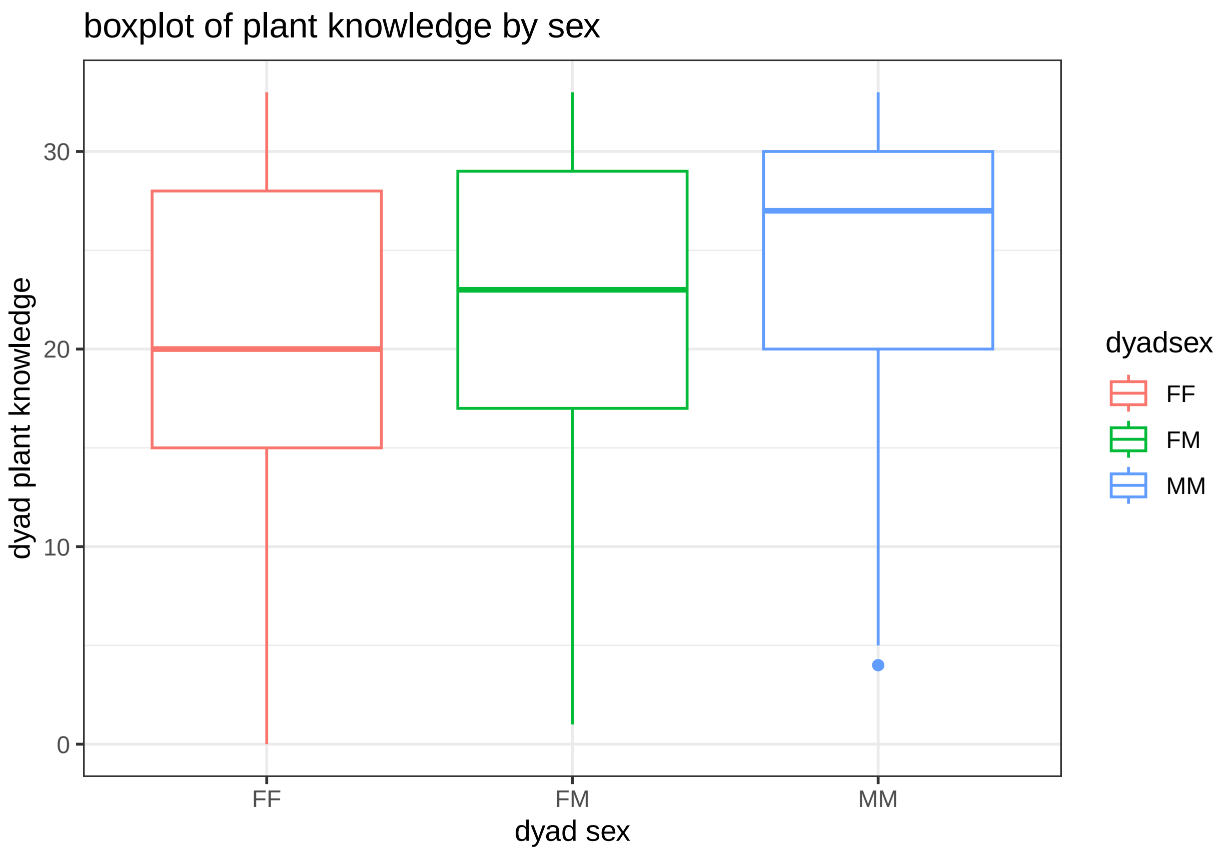
```
# exclude NAs
know_sex <- dyads_merged %>%
  filter(dyadsex != "NANA") %>%
  group_by(dyad_ID, dyadsex) %>%
  summarise(sum_know = sum(dyadknow), n= n())

kruskal.test(sum_know ~ dyadsex, data = know_sex)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  sum_know by dyadsex
## Kruskal-Wallis chi-squared = 881.85, df = 2, p-value < 2.2e-16
```

boxplot

```
ggplot() +
  geom_boxplot(data = know_sex, mapping = aes(x = dyadsex, y = sum_know, colour = dyadsex)) +
  labs(x = "dyad sex", y = "dyad plant knowledge",
       title = "boxplot of plant knowledge by sex") +
  theme_bw()
```



### Interpretation

From the p-value in the Kruskal-Walis test one sees that dyad sex is a valid predictor for total plant knowledge. Furthermore, the boxplot implies that men share more plant knowledge.

c) camp

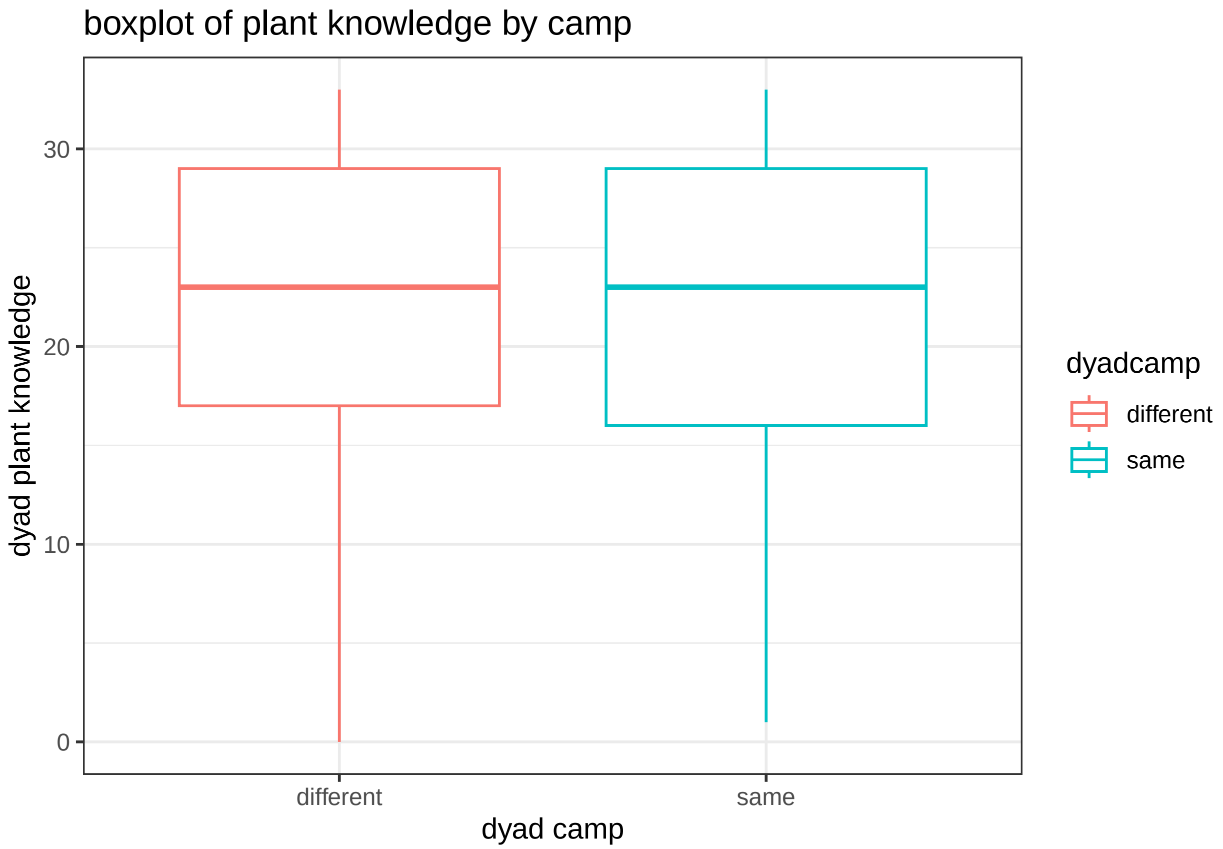
Kruskal-Walis test

```
dyads_merged$dyadcamp <- ifelse(dyads_merged$camp1 == dyads_merged$camp2, "same", "different")

# how do I get rid of the NAs?
know_camp <- dyads_merged %>%
  # filter(!is.na(dyadcamp)) %>%
  group_by(dyad_ID, dyadcamp) %>%
  summarise(sum_know = sum(dyadknow), n= n()) # %>%
kruskal.test(sum_know ~ dyadcamp, data = know_camp)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  sum_know by dyadcamp
## Kruskal-Wallis chi-squared = 9.6007, df = 1, p-value = 0.001945
```

```
ggplot() +
  geom_boxplot(data = know_camp, mapping = aes(x = dyadcamp, y = sum_know, colour = dyadcamp)) +
  labs(x = "dyad camp", y = "dyad plant knowledge",
       title = "boxplot of plant knowledge by camp") +
  theme_bw()
```



### Interpretation

The Kruskal Walis test implies a significant relationship between camp and sum of shared knowledge. This result needs to be treated with care, because the boxplot implies little significance.

5. Regression analysis

```
# column shared knowledge is called dyadknow
# function to convert odds into probabilities
```

```
odds2P <- function (odds){
  return (odds / (1 + odds))
}
```

a) age

Regression

```
simplem_logreg_age <- glm(dyadknow ~ dyadagelevels, binomial, data = dyads_merged)
summary(simplem_logreg_age)
```

```
##
## Call:
## glm(formula = dyadknow ~ dyadagelevels, family = binomial, data = dyads_merged)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2439  -1.4946   0.8904   0.8904   1.5350
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)            2.43336    0.04409   55.19   <2e-16 ***
## dyadagelevelsothers -1.71282    0.04416  -38.79   <2e-16 ***
## dyadagelevelsyoung  -3.24339    0.05059  -64.11   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 992103  on 783386  degrees of freedom
## Residual deviance: 985462  on 783384  degrees of freedom
##   (4356 observations deleted due to missingness)
## AIC: 985468
##
## Number of Fisher Scoring iterations: 4
```

Analysis

```
# coeficicents of model
coef_age <- coef(simplem_logreg_age) ; names(coef_age) <- c("baseline", "others", "young")
# odds baseline = old: P(know) / P(not know)
odds_base <- exp(coef_age["baseline"])
# odds ratios of who is more likely to know plant vs baseline
ratio_other <- exp(coef_age["others"]) # others:old
ratio_young <- exp(coef_age["young"]) # young:old
# odds exposure groups: P(know) / P(not know)
odds_others <- exp(coef_age["others"] + coef_age["baseline"])
odds_young <- exp(coef_age["young"] + coef_age["baseline"])
# probabilities that groups share knowledge
print(paste("P(other share) =", odds2P(odds_others)))
```

```
## [1] "P(other share) = 0.67272519165544"
```

```
print(paste("P(young share) =", odds2P(odds_young)))
```

```
## [1] "P(young share) = 0.307884035161342"
```

```
print(paste("P(old share) =", odds2P(odds_base)))
```

```
## [1] "P(old share) = 0.919336169340131"
```

Interpretations

The probability that knowledge is shared in a dyad rises with the age level and the p-value indicates that this relationship is significant.

b) sex

Regression

```
simplem_logreg_sex <- glm(dyadknow ~ dyadsex, binomial, data = dyads_merged %>% filter(dyadsex != "NANA
summary(simplem_logreg_sex)
```

```
## 
## Call:
## glm(formula = dyadknow ~ dyadsex, family = binomial, data = dyads_merged %>% 
##     filter(dyadsex != "NANA"))
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -1.6484  -1.3873   0.8854   0.8854   0.9811  
## 
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)    
## (Intercept) 0.481026   0.004276  112.50   <2e-16 ***
## dyadsexFM   0.253254   0.005479   46.22   <2e-16 ***
## dyadsexMM   0.580572   0.007141   81.30   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 987349  on 780548  degrees of freedom
## Residual deviance: 980515  on 780546  degrees of freedom
## AIC: 980521
## 
## Number of Fisher Scoring iterations: 4
```

Analysis

```
coef_sex <- coef(simplem_logreg_sex)[1:3] ; names(coef_sex) <- c("baseline", "FM", "MM")
# odds baseline: P(know) / P(not know)
odds_FF <- exp(coef_sex["baseline"])
# odds ratios of who is more likely to know plant vs baseline
ratio_FM <- exp(coef_sex["FM"]) # FM:FF
ratio_MM <- exp(coef_sex["MM"]) # MM:FF
# odds exposure groups: P(know) / P(not know)
odds_FM <- exp(coef_sex["FM"] + coef_sex["baseline"])
odds_MM <- exp(coef_sex["MM"] + coef_sex["baseline"])
# probalities that groups shares knowledge
print(paste("P(FF share) =", odds2P(odds_FF)))
```

```
## [1] "P(FF share) = 0.617990185287495"
```

```
print(paste("P(FM share) =", odds2P(odds_FM)))
```

```
## [1] "P(FM share) = 0.675743943122912"
```

```
print(paste("P(MM share) =", odds2P(odds_MM)))
```

```
## [1] "P(MM share) = 0.742995820917068"
```

Interpretation

The probability that knowledge is shared rises when a man is part of the dyad.

c) camp

Regression

```
simplem_logreg_camp <- glm(dyadknow ~ dyadcamp, binomial, data = dyads_merged)
summary(simplem_logreg_camp)
```

```
##
## Call:
## glm(formula = dyadknow ~ dyadcamp, family = binomial, data = dyads_merged)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4972  -1.4725   0.8883   0.8883   0.9086
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.726272   0.002795  259.86   <2e-16 ***
## dyadcampsame -0.054878   0.005440  -10.09   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 998225  on 787742  degrees of freedom
## Residual deviance: 998123  on 787741  degrees of freedom
## AIC: 998127
##
## Number of Fisher Scoring iterations: 4
```

Analysis

```
coef_camp <- coef(simplem_logreg_camp) ; names(coef_camp) <- c("baseline", "same")
# odds baseline: P(know) / P(not know)
odds_diff <- exp(coef_camp["baseline"])
# odds ratios of who is more likely to know plant vs baseline
ratio_sampe <- exp(coef_camp["same"]) # same:diff
# odds exposure groups: P(know) / P(not know)
odds_same <- exp(coef_camp["same"] + coef_camp["baseline"])
# probalities that groups share knowledge
print(paste("P(diff camp share) =", odds2P(odds_diff)))
```

```
## [1] "P(diff camp share) = 0.673986695227569"
```

```
print(paste("P(same camp share) =", odds2P(odds_same)))
```

```
## [1] "P(same camp share) = 0.661815256346314"
```

Interpretation

Camp doesn't seem to be a valid predictor for shared knowledge because probability that knowledge is shared by people in same comp is similar to the one of dyads not belonging to the same camp.

d) multiplicative

Regression

```
simplem_logreg_agexsex <- glm(dyadknow ~ dyadagelevels * dyadsex, binomial, data = dyads_merged %>% fil
summary(simplem_logreg_agexsex)
```

```
##
## Call:
## glm(formula = dyadknow ~ dyadagelevels * dyadsex, family = binomial,
##     data = dyads_merged %>% filter(dyadsex != "NANA"))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5433  -1.3813   0.8813   0.8813   1.6313
##
## Coefficients:
```

```
##                               Estimate Std. Error z value Pr(>|z|)    
## (Intercept)                    2.28138    0.05480  41.631  < 2e-16 ***
## dyadagelevelsothers           -1.81390    0.05497 -32.998  < 2e-16 ***
## dyadagelevelsyoung            -3.30507    0.08563 -38.598  < 2e-16 ***
## dyadsexFM                      0.33956    0.09485   3.580 0.000344 ***
## dyadsexMM                      0.91257    0.28816   3.167 0.001541 ** 
## dyadagelevelsothers:dyadsexFM -0.06161    0.09501  -0.649 0.516660    
## dyadagelevelsyoung:dyadsexFM  -0.15738    0.12065  -1.304 0.192079    
## dyadagelevelsothers:dyadsexMM -0.27123    0.28825  -0.941 0.346724    
## dyadagelevelsyoung:dyadsexMM  -0.56112    0.29849  -1.880 0.060129 .  
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 981221  on 776192  degrees of freedom
## Residual deviance: 966471  on 776184  degrees of freedom
##   (4356 observations deleted due to missingness)
## AIC: 966489
## 
## Number of Fisher Scoring iterations: 5
```

Analysis

```
# the baseline is oldFF
coef_mult <- coef(simplem_logreg_agexsex)[1:5] ; names(coef_mult) <-c("baseline", "otherFF", "youngFF",
# basic probabilities
print(paste("P(share young,FF) =", odds2P(exp(coef_mult["youngFF"] + coef_mult["baseline"]))))
```

```
## [1] "P(share young,FF) = 0.264309764317572"
```

```
print(paste("P(share other,FF) =", odds2P(exp(coef_mult["otherFF"] + coef_mult["baseline"]))))
```

```
## [1] "P(share other,FF) = 0.61478734909351"
```

```
print(paste("P(share old,FF) =", odds2P(exp(coef_mult["baseline"]))))
```

```
## [1] "P(share old,FF) = 0.907323232322887"
```

```
print(paste("P(share old,FM) =", odds2P(exp(coef_mult["oldFM"] + coef_mult["baseline"]))))
```

```
## [1] "P(share old,FM) = 0.932196969695637"
```

```
print(paste("P(share old,MM) =", odds2P(exp(coef_mult["oldMM"] + coef_mult["baseline"]))))
```

```
## [1] "P(share old,MM) = 0.960606053627194"
```

Interpretation

Because the interaction is not significant, we see that the probabilities for P(share young, FM) or P(share young, MM) do not differ significantly from the calculated P(share young, FF). In the same sense for P(share others, FF) and the combined probabilities for the two other age levels or all other possible combinations. This is the conceptual meaning for a non-significant interaction for logistic regression performed up on multiple levels!

Optimization

```
summary(step(glm(dyadknow ~ dyadagelevels + dyadsex + dyadcamp, binomial, data = dyads_merged %>% filte
```

```
## Start:  AIC=966396.7
## dyadknow ~ dyadagelevels + dyadsex + dyadcamp
##
##                 Df Deviance    AIC
## <none>               966385 966397
## - dyadcamp       1   966491 966501
## - dyadagelevels  2   973982 973990
## - dyadsex        2   974474 974482

##
## Call:
## glm(formula = dyadknow ~ dyadagelevels + dyadsex + dyadcamp,
##     family = binomial, data = dyads_merged %>% filter(dyadsex !=
##         "NANA"))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4540  -1.3880   0.8755   0.8965   1.7160
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)          2.321543   0.044189   52.54   <2e-16 ***
## dyadagelevelsothers -1.839046   0.044229  -41.58   <2e-16 ***
## dyadagelevelsyoung  -3.476119   0.050781  -68.45   <2e-16 ***
## dyadsexFM            0.278780   0.005517   50.53   <2e-16 ***
## dyadsexMM            0.638983   0.007242   88.23   <2e-16 ***
## dyadcampsame        -0.057205   0.005534  -10.34   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 981221  on 776192  degrees of freedom
## Residual deviance: 966385  on 776187  degrees of freedom
##   (4356 observations deleted due to missingness)
## AIC: 966397
##
## Number of Fisher Scoring iterations: 4
```

The step function suggests that the best AIC is optained if all features are kept in the model.

6) Mixed Effect Models

a) included effect: learned from the same realtionship

```
dyads_merged$samelearned <- ifelse(dyads_merged$learned1 == dyads_merged$learned2, 1, 0)
# dyads_merged$sameborn <- ifelse(dyads_merged$born1 == dyads_merged$born2, 1, 0)
# variance components analysis?
```

b) age

```
mixedm_logreg_age <- glmer(dyadknow ~ dyadagelevels + (1|samelearned), family = binomial, data = dyads_
summary(mixedm_logreg_age)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: dyadknow ~ dyadagelevels + (1 | samelearned)
##    Data: dyads_merged
##
##       AIC       BIC    logLik  deviance  df.resid
##  859793.3  859839.1 -429892.7  859785.3    692831
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.3177 -1.4823  0.6746  0.6746  1.4550
##
## Random effects:
##  Groups      Name        Variance Std.Dev.
##  samelearned (Intercept) 0.001515 0.03893
## Number of obs: 692835, groups:  samelearned, 2
##
## Fixed effects:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)          2.35993    0.03109   75.90   <2e-16 ***
## dyadagelevelsothers -1.61139    0.02642  -60.99   <2e-16 ***
## dyadagelevelsyoung  -3.07138    0.03083  -99.63   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) dydglvlst
## dydglvlsthr -0.703
## dydglvlsyng -0.573  0.698
```

Analysis

```
# coeficicents of model
coef_age <- coef(mixedm_logreg_age)$samelearned[1,]; names(coef_age) <- c("baseline", "others", "young"
```

```r
# odds baseline = old: P(know) / P(not know)
odds_base <- exp(coef_age["baseline"])
# odds ratios of who is more likely to know plant vs baseline
ratio_other <- exp(coef_age["others"]) # others:old
ratio_young <- exp(coef_age["young"]) # young:old
# odds exposure groups: P(know) / P(not know)
odds_others <- exp(coef_age["others"] + coef_age["baseline"])
odds_young <- exp(coef_age["young"] + coef_age["baseline"])
# probabilities that groups share knowledge
print(paste("P(other share) =", odds2P(odds_others)))
```

```
## [1] "P(other share) = 0.687222937382532"
```

```r
print(paste("P(young share) =", odds2P(odds_young)))
```

```
## [1] "P(young share) = 0.337865541063698"
```

```r
print(paste("P(old share) =", odds2P(odds_base)))
```

```
## [1] "P(old share) = 0.916717585245083"
```

Interpretetion

Even though the chosen random effect doesn't explain a lot of variance in the data, the mixed model results in different probabilities than the simple one in 5). The probabiilities are more certain and increase in ~ 1%.

c) sex

```r
mixedm_logreg_sex <- glmer(dyadknow ~ dyadsex + (1|samelearned), family = binomial, data = dyads_merged
summary(mixedm_logreg_sex)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: dyadknow ~ dyadsex + (1 | samelearned)
##    Data: dyads_merged %>% filter(dyadsex != "NANA")
##
##       AIC       BIC    logLik  deviance  df.resid
##  854077.6  854123.4 -427034.8  854069.6    690026
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.7582 -1.3228  0.6721  0.6721  0.8030
##
## Random effects:
##  Groups      Name        Variance Std.Dev.
##  samelearned (Intercept) 0.003679 0.06066
## Number of obs: 690030, groups:  samelearned, 2
```

```
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 0.499134   0.035612   14.02   <2e-16 ***
## dyadsexFM   0.235037   0.005913   39.75   <2e-16 ***
## dyadsexMM   0.568969   0.007471   76.15   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##           (Intr) dydsFM
## dyadsexFM -0.082
## dyadsexMM -0.058  0.478
```

Analysis

```
coef_sex <- coef(mixedm_logreg_sex)$samelearned[1,] ; names(coef_sex) <- c("baseline", "FM", "MM")
# odds baseline: P(know) / P(not know)
odds_FF <- exp(coef_sex["baseline"])
# odds ratios of who is more likely to know plant vs baseline
ratio_FM <- exp(coef_sex["FM"]) # FM:FF
ratio_MM <- exp(coef_sex["MM"]) # MM:FF
# odds exposure groups: P(know) / P(not know)
odds_FM <- exp(coef_sex["FM"] + coef_sex["baseline"])
odds_MM <- exp(coef_sex["MM"] + coef_sex["baseline"])
# probalities that groups shares knowledge
print(paste("P(FF share) =", odds2P(odds_FF)))
```

```
## [1] "P(FF share) = 0.636350059107171"
```

```
print(paste("P(FM share) =", odds2P(odds_FM)))
```

```
## [1] "P(FM share) = 0.688816886979458"
```

```
print(paste("P(MM share) =", odds2P(odds_MM)))
```

```
## [1] "P(MM share) = 0.755566826529226"
```

Interpretation

The random effect samelearned explains 3 times as much variability of differences in dyadsex as in dyadage, but still very little. Nevertheless, the probabilities increase again in ~1% on average, which means that the random effect led to gained certainty.

d) camp

```
mixedm_logreg_camp <- glmer(dyadknow ~ dyadcamp + (1|samelearned), family = binomial, data = dyads_merg
summary(mixedm_logreg_camp)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: dyadknow ~ dyadcamp + (1 | samelearned)
##    Data: dyads_merged
##
##       AIC       BIC    logLik  deviance  df.resid
##  870101.3  870135.7 -435047.7  870095.3    696792
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.4862 -1.4579  0.6728  0.6859  0.7150
##
## Random effects:
##  Groups      Name        Variance Std.Dev.
##  samelearned (Intercept) 0.001749 0.04182
## Number of obs: 696795, groups:  samelearned, 2
##
## Fixed effects:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.750920   0.020547  36.547  < 2e-16 ***
## dyadcampsame -0.038422   0.005757  -6.674 2.48e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## dyadcampsam -0.065
```

Analysis

```
coef_camp <- coef(mixedm_logreg_camp)$samelearned[1,] ; names(coef_camp) <- c("baseline", "same")
# odds baseline: P(know) / P(not know)
odds_diff <- exp(coef_camp["baseline"])
# odds ratios of who is more likely to know plant vs baseline
ratio_sampe <- exp(coef_camp["same"]) # same:diff
# odds exposure groups: P(know) / P(not know)
odds_same <- exp(coef_camp["same"] + coef_camp["baseline"])
# probalities that groups share knowledge
print(paste("P(diff camp share) =", odds2P(odds_diff)))
```

```
## [1] "P(diff camp share) = 0.688359525225556"
```

```
print(paste("P(same camp share) =", odds2P(odds_same)))
```

```
## [1] "P(same camp share) = 0.680058173057575"
```

e) multiplicative

```
mixedm_logreg_agexsex <- glmer(dyadknow ~ dyadagelevels * dyadsex + (1|samelearned), family = binomial,
summary(mixedm_logreg_agexsex)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: dyadknow ~ dyadagelevels * dyadsex + (1 | samelearned)
##    Data: dyads_merged %>% filter(dyadsex != "NANA")
##
##       AIC       BIC    logLik  deviance  df.resid
##  842671.0  842785.4 -421325.5  842651.0    686060
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.9381 -1.3112  0.6701  0.6701  1.5367
##
## Random effects:
##  Groups      Name        Variance Std.Dev.
##  samelearned (Intercept) 0.00329  0.05736
## Number of obs: 686070, groups:  samelearned, 2
##
## Fixed effects:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   2.17133    0.07814  27.787  < 2e-16 ***
## dyadagelevelsothers          -1.68661    0.06142 -27.460  < 2e-16 ***
## dyadagelevelsyoung           -2.97341    0.03464 -85.829  < 2e-16 ***
## dyadsexFM                     0.36329    0.08117   4.475 7.62e-06 ***
## dyadsexMM                     1.07979    0.12236   8.825  < 2e-16 ***
## dyadagelevelsothers:dyadsexFM -0.10448    0.07940  -1.316 0.188245
## dyadagelevelsyoung:dyadsexFM  -0.32081    0.04391  -7.307 2.74e-13 ***
## dyadagelevelsothers:dyadsexMM -0.45245    0.12402  -3.648 0.000264 ***
## dyadagelevelsyoung:dyadsexMM  -0.92201    0.13973  -6.598 4.16e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr) dydglvlst dydglvlsy dydsFM dydsMM dydglvlst:FM dydglvlsy:FM
## dydglvlsthr  -0.880
## dydglvlsyng  -0.442  0.602
## dyadsexFM    -0.861  0.868     0.416
## dyadsexMM     0.847 -0.853    -0.485    -0.908
## dydglvlst:FM  0.860 -0.869    -0.415    -0.997  0.906
## dydglvlsy:FM  0.510 -0.502    -0.424    -0.660  0.616  0.658
## dydglvlst:MM -0.847  0.852     0.485     0.909 -0.998 -0.905       -0.617
## dydglvlsy:MM -0.863  0.865     0.424     0.910 -0.953 -0.907       -0.600
##              dydglvlst:MM
## dydglvlsthr
## dydglvlsyng
## dyadsexFM
## dyadsexMM
## dydglvlst:FM
## dydglvlsy:FM
## dydglvlst:MM
## dydglvlsy:MM  0.953
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 0.00215034 (tol = 0.002, component 1)
```

Analysis

```
# the baseline is oldFF
coef_mult <- coef(mixedm_logreg_agexsex)$samelearned[1,] ; names(coef_mult) <-c("baseline", "otherFF",
                                                                                 "youngFM", "othersMM",
# basic probabilities
print(paste("P(share young,FF) =", odds2P(exp(coef_mult["youngFF"] + coef_mult["baseline"]))))
```

```
## [1] "P(share young,FF) = 0.321928219412141"
```

```
print(paste("P(share other,FF) =", odds2P(exp(coef_mult["otherFF"] + coef_mult["baseline"]))))
```

```
## [1] "P(share other,FF) = 0.632248787054176"
```

```
print(paste("P(share old,FF) =", odds2P(exp(coef_mult["baseline"]))))
```

```
## [1] "P(share old,FF) = 0.902778264603097"
```

```
print(paste("P(share old,FM) =", odds2P(exp(coef_mult["oldFM"] + coef_mult["baseline"]))))
```

```
## [1] "P(share old,FM) = 0.930330191157814"
```

```
print(paste("P(share old,MM) =", odds2P(exp(coef_mult["oldMM"] + coef_mult["baseline"]))))
```

```
## [1] "P(share old,MM) = 0.96471163384014"
```

```
print("P(share others,FM) doesn't vary significantly from P(share young, FM) or P(share old, FM)")
```

```
## [1] "P(share others,FM) doesn’t vary significantly from P(share young, FM) or P(share old, FM)"
```

```
# print(paste("P(share others,FM) =", odds2P(exp(coef_mult["othersFM"] + coef_mult["otherFF"] + coef_mu
print(paste("P(share young,FM) =", odds2P(exp(coef_mult["youngFM"] + coef_mult["youngFF"] + coef_mult["
```

```
## [1] "P(share young,FM) = 0.331271512456108"
```

```
print(paste("P(share others,MM) =", odds2P(exp(coef_mult["othersMM"] + coef_mult["otherFF"] + coef_mult
```

```
## [1] "P(share others,MM) = 0.763003721226337"
```

```
print(paste("P(share young,MM) =", odds2P(exp(coef_mult["youngMM"] + coef_mult["youngFF"] + coef_mult["
```

```
## [1] "P(share young,MM) = 0.357291066743476"
```

Interpretation

The random effect had a significant effect on the model even though it explains little variance amount. The interaction terms are now classified as significant, which means that for example P(share young,FM) does vary significantly from P(share old,FF) and P(others, FF). Thus, the random effect gave the model more explanatory power and certainty.

7) Conclusion

Overall, the result seems to be coherent over the analysis through basic test and boxplots over simple logistic regression to the mixed regression models. The hypothesis resulting from exploring the data in 4) are confirmed in the built model of 5) and 6). From the simple model it becomes clear that sex and age are valid predictors for shared plant knowledge, since probability of shared knowledge in dyads increase with age or fraction of males in the dyad. However, being in the same camp doesn't increase the probability for sharing knowledge. General trends are that the models with interactions show lower AICs and thus have greater explanatory power than single-level ones. Moreover, in the mixed models, the random effect describing the shared source of learnt content brings more certainty in the probabilities and further lowers the AIC in the model. Thus, the model with the greatest explanatory power is the one of 6e), a two level mixed effect model. From the probabilities resutling from the single level models I would rank the factors in importance as follows: 1. age, 2. sex and 3. camp.