SI for the paper: The Semantic Origins of Gender Assignment

Nour Efrat-Kowalsky

2025-07-23

1. Data:

In some cases two variants are found in the extant material for a lexeme: one feminine and one masculine. In the case of Arawak and Indo-European, both variants were coded in the data, but the more common variant is used in the analysis. Because the Semitic data includes long-dead languages and the corpus is limited, there is no way to determine the more common variant in the language. For these lexemes, both variants were coded and used in the analysis, such that for that lexeme in that language there are two entries. In total, there are 55 such lexemes in the dataset. Moreover, in the Semitic dataset, in a handful of cases there were two lexemes (forms) in one language that corresponded to a concept. In cases where these were identical in all other respects (as we are not interested in the form of a lexeme in this study), the duplicate was deleted. This was the case for only five out of 1,480 lexemes, leaving the dataset with 1475 lexemes.

```
gender_assignment <- read_excel("data/gender_assignment_ie_ar_sem.xlsx") %>%
  rename(Glottocode = GlottoCode)
```

create a dataframe per language family

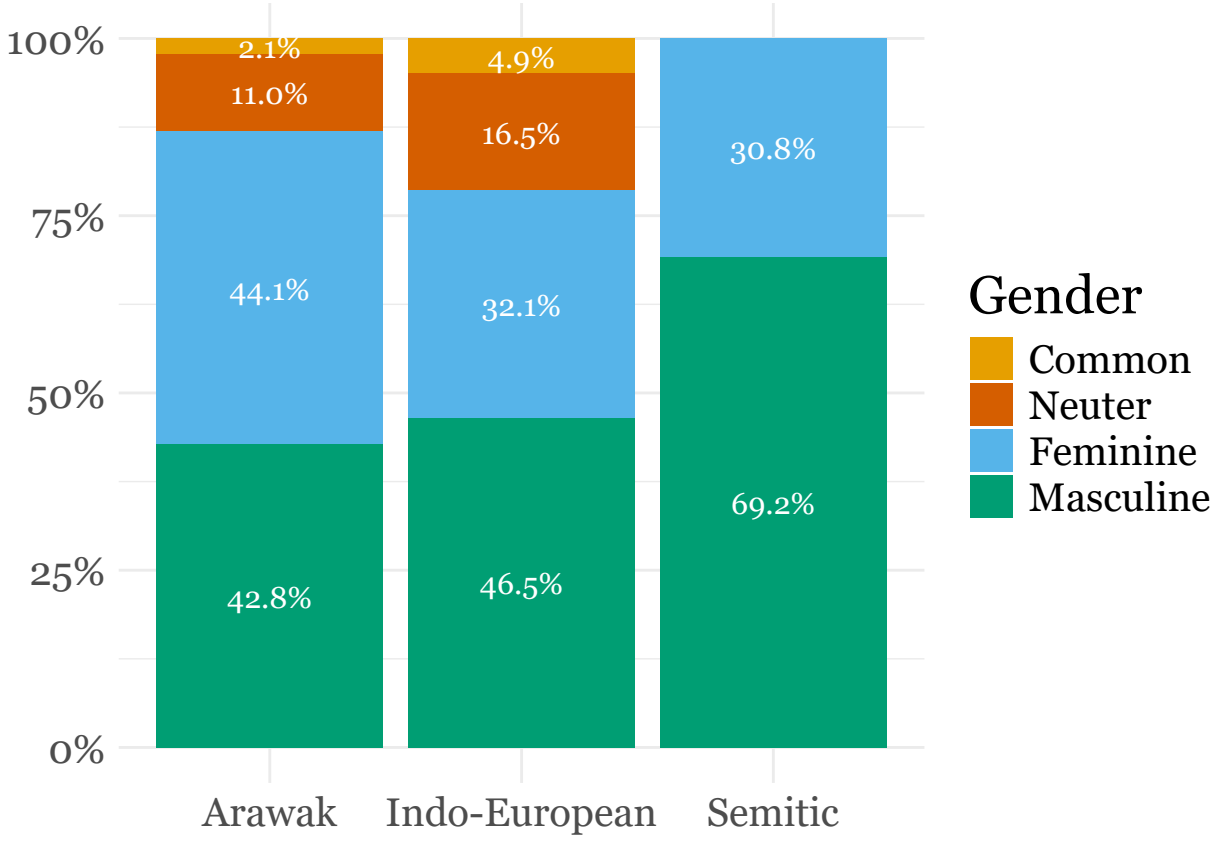
```
IE <- gender_assignment %>% filter(Family == "Indo-European")
Arawak <- gender_assignment %>% filter(Family == "Arawak")
Semitic <- gender_assignment %>% filter(Family == "Semitic")
```

1.1. Data Visualisation:

plot distributions of gender per family

```
stack <- gender_assignment %>%   mutate(Gender = factor(Gender, levels = c("C", "N", "F", "M")))
ggplot(stack, aes(x = Family, fill = Gender)) +
  geom_bar(position = "fill") +
  geom_text(stat = "count",
            aes(label = scales::percent(..count../tapply(..count.., ..x.., sum)[..x..])),
            position = position_fill(vjust = 0.5),
            color = "white", size = 4,
            family = "Georgia") +
  labs(title = NULL,
       x = NULL,
       y = NULL,
       fill = "Gender") +
  scale_fill_manual(
    values = c("C" = "#E69F00", "F" = "#56B4E9", "M" = "#009E73", "N" = "#D55E00"),
```

```
    labels = c("C" = "Common", "F" = "Feminine", "M" = "Masculine", "N" = "Neuter")
  ) +
  scale_y_continuous(labels = scales::percent_format()) +
  theme_minimal()+
  theme(text = element_text(family = "Georgia"),
        plot.title = element_text(size = 20, face = "bold"),
        axis.title = element_text(size = 15),
        axis.text = element_text(size = 15),
        legend.title = element_text(size = 20),
        legend.text = element_text(size = 15))
```




```
ggsave("plots/Proportional Distribution of Gender by Family.png", width = 8, height = 4, dpi = 300)

#We can see the numbers here:
IE %>%
  count(Gender) %>%
  mutate(Proportion = n / sum(n))
```

```
## # A tibble: 4 x 3
##   Gender     n Proportion
##   <chr>  <int>      <dbl>
## 1 C        576     0.0492
## 2 F       3753     0.321
## 3 M       5441     0.465
## 4 N       1931     0.165
```

```
Arawak %>%
  count(Gender) %>%
  mutate(Proportion = n / sum(n))
```

```
## # A tibble: 4 x 3
##   Gender     n Proportion
##   <chr>  <int>      <dbl>
## 1 C         59     0.0214
## 2 F       1218     0.441
## 3 M       1181     0.428
## 4 N        303     0.110
```

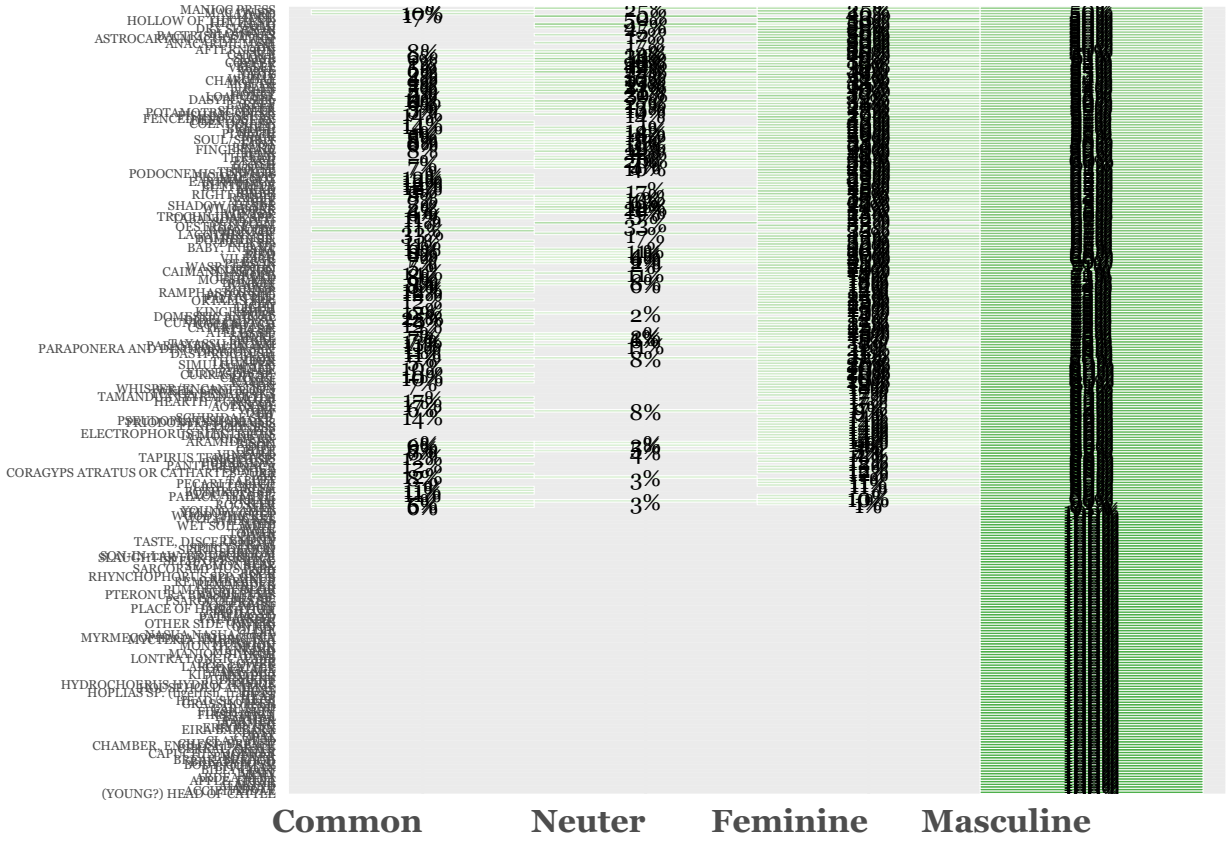
```
Semitic %>%
  count(Gender) %>%
  mutate(Proportion = n / sum(n))
```

```
## # A tibble: 2 x 3
##   Gender     n Proportion
##   <chr>  <int>      <dbl>
## 1 F        454      0.308
## 2 M       1021      0.692
```

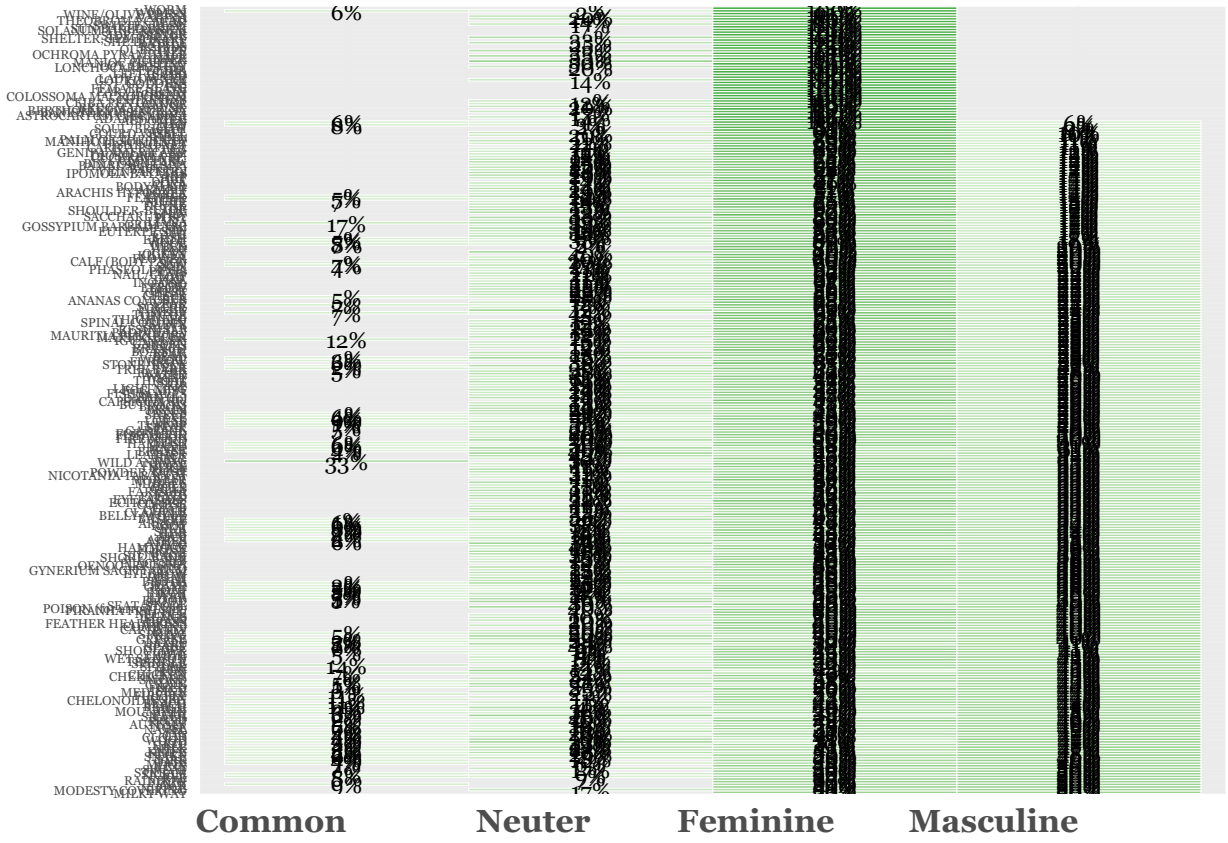
To visualise the proportion of each gender in each noun in the dataset we plot a heatmap of the concept list per gender

```
heatmap_data <- gender_assignment %>%
  mutate(Gender = recode(Gender,
                         "C" = "Common",
                         "F" = "Feminine",
                         "M" = "Masculine",
                         "N" = "Neuter")) %>%
  mutate(Gender = factor(Gender, levels = rev(c("Masculine", "Feminine", "Neuter", "Common")))) %>%
  count(Concept, Gender) %>%
  group_by(Concept) %>%
  mutate(Percentage = n / sum(n)) %>%
  ungroup()

# Calculate order of Concepts by Masculine proportion (descending)
concept_order <- heatmap_data %>%
  filter(Gender == "Masculine") %>%
  arrange(desc(Percentage)) %>%
  pull(Concept) %>%
  unique()

# Reorder Concept factor in heatmap_data
all_concepts <- unique(heatmap_data$Concept)
heatmap_data <- heatmap_data %>%
  mutate(Concept = factor(Concept, levels = c(concept_order, setdiff(all_concepts, concept_order))))


#the plot is too long, need to split it up
# Split concept levels into two halves
```

```r
concept_levels <- levels(heatmap_data$Concept)
n <- length(concept_levels)
half <- ceiling(n / 2)

concepts_part1 <- concept_levels[1:half]
concepts_part2 <- concept_levels[(half + 1):n]

# Subsets
heatmap_part1 <- heatmap_data %>%
  filter(Concept %in% concepts_part1)

heatmap_part2 <- heatmap_data %>%
  filter(Concept %in% concepts_part2)

ggplot(heatmap_part1, aes(x = Gender, y = Concept, fill = Percentage)) +
  geom_tile(color = "white") +
  geom_text(aes(label = scales::percent(Percentage, accuracy = 1)),
            color = "black", size = 3, family = "Georgia") +
  scale_fill_gradient(low = "#e0f3db", high = "#4daf4a") +
  labs(
    title = NULL,
    x = NULL,
    y = NULL
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 0, hjust = 1, size = 12, face = "bold"),
    text = element_text(family = "Georgia", size = 6),
    plot.title = element_text(size = 20, face = "bold"),
    legend.position = "none"
  )
```




```
ggsave("plots/ie_ar_sem_Heatmap Concepts2.png", width = 12, height = 18, dpi = 300)

ggplot(heatmap_part2, aes(x = Gender, y = Concept, fill = Percentage)) +
  geom_tile(color = "white") +
  geom_text(aes(label = scales::percent(Percentage, accuracy = 1)),
            color = "black", size = 3, family = "Georgia") +
  scale_fill_gradient(low = "#e0f3db", high = "#4daf4a") +
  labs(
    title = NULL,
    x = NULL,
    y = NULL
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 0, hjust = 1, size = 12, face = "bold"),
    text = element_text(family = "Georgia", size = 6),
    plot.title = element_text(size = 20, face = "bold"),
    legend.position = "none"
  )
```

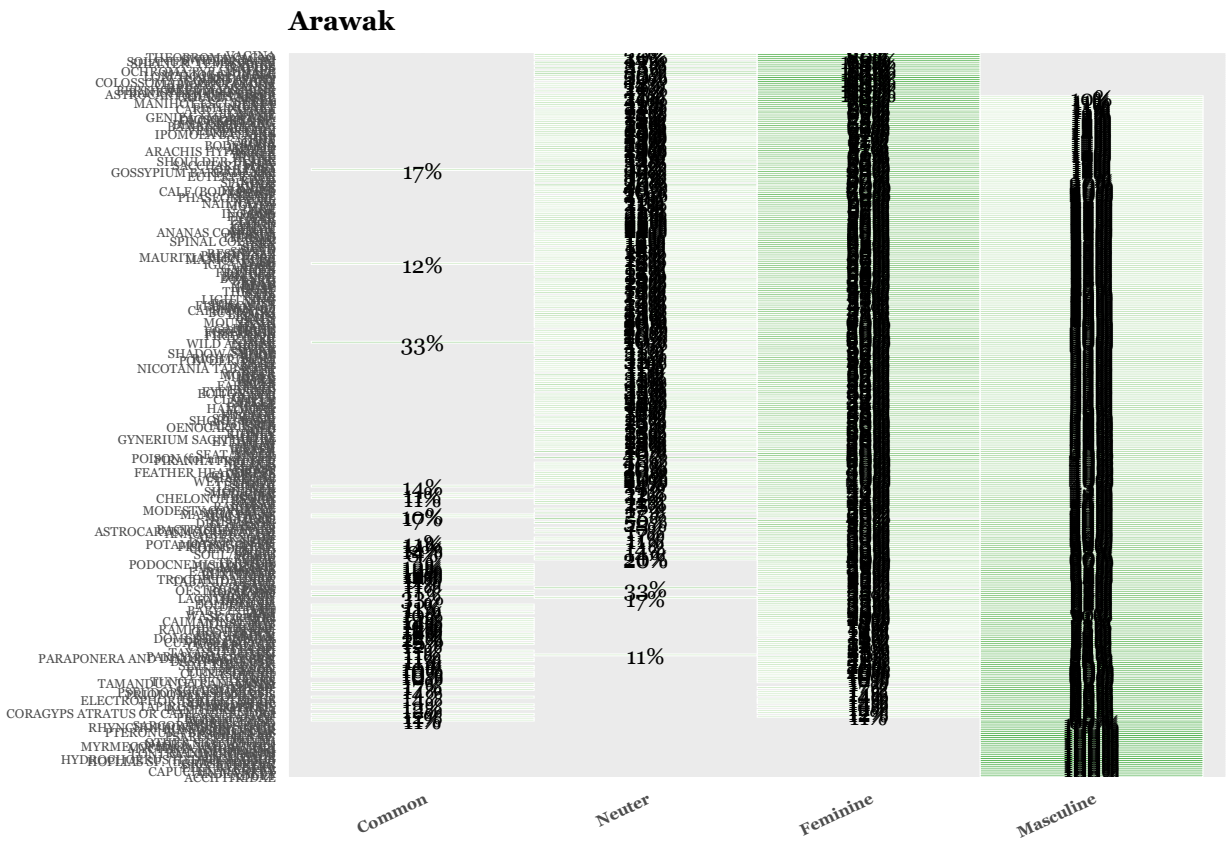



```
ggsave("plots/ie_ar_sem_Heatmap Concepts1.png", width = 12, height = 18, dpi = 300)
```

Heatmap of the concept list per gender - per family

```
#Arawak
arawak_heatmap <- Arawak %>%
  mutate(Gender = recode(Gender,
                         "C" = "Common",
                         "F" = "Feminine",
                         "M" = "Masculine",
                         "N" = "Neuter")) %>%
  mutate(Gender = factor(Gender, levels = rev(c("Masculine", "Feminine", "Neuter", "Common")))) %>%
  count(Concept, Gender) %>%
  group_by(Concept) %>%
  mutate(Percentage = n / sum(n)) %>%
  ungroup()

# Calculate order of Concepts by Masculine proportion (descending)
concept_order_ar <- arawak_heatmap %>%
  filter(Gender == "Masculine") %>%
  arrange(desc(Percentage)) %>%
  pull(Concept) %>%
  unique()

# Reorder Concept factor in heatmap_data
```

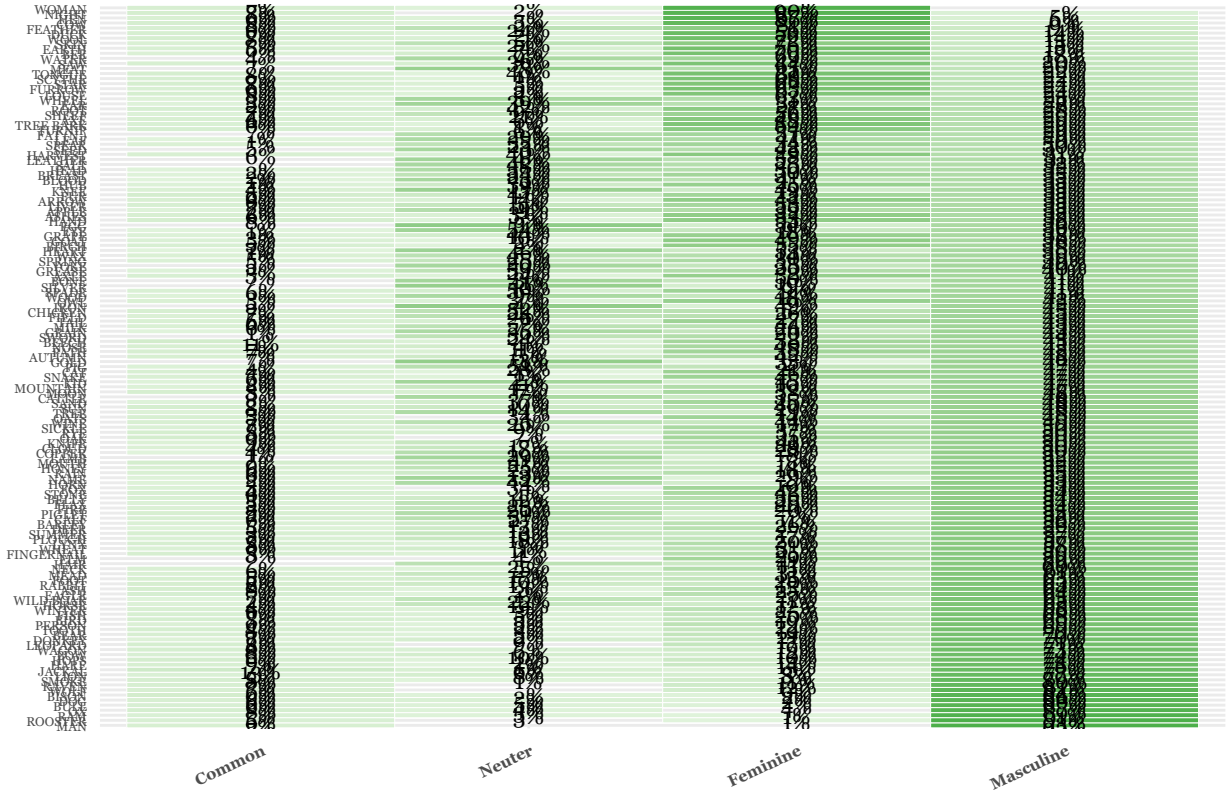
```
all_concepts_ar <- unique(arawak_heatmap$Concept)
arawak_heatmap <- arawak_heatmap %>%
  mutate(Concept = factor(Concept, levels = c(concept_order_ar, setdiff(all_concepts_ar, concept_order_a

#Plot
ggplot(arawak_heatmap, aes(x = Gender, y = Concept, fill = Percentage)) +
  geom_tile(color = "white") +
  geom_text(aes(label = scales::percent(Percentage, accuracy = 1)),
            color = "black", size = 3, family = "Georgia") +
  scale_fill_gradient(low = "#e0f3db", high = "#4daf4a") +
  labs(
    title = "Arawak",
    x = NULL,
    y = NULL
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 25, hjust = 1, size = 6, face = "bold"),
    text = element_text(family = "Georgia", size = 6),
    plot.title = element_text(size = 10, face = "bold"),
    legend.position = "none"  # remove legend
  )
```




```
ggsave("plots/Heatmap concept Arawak.png", width = 12, height = 18, dpi = 300)
```

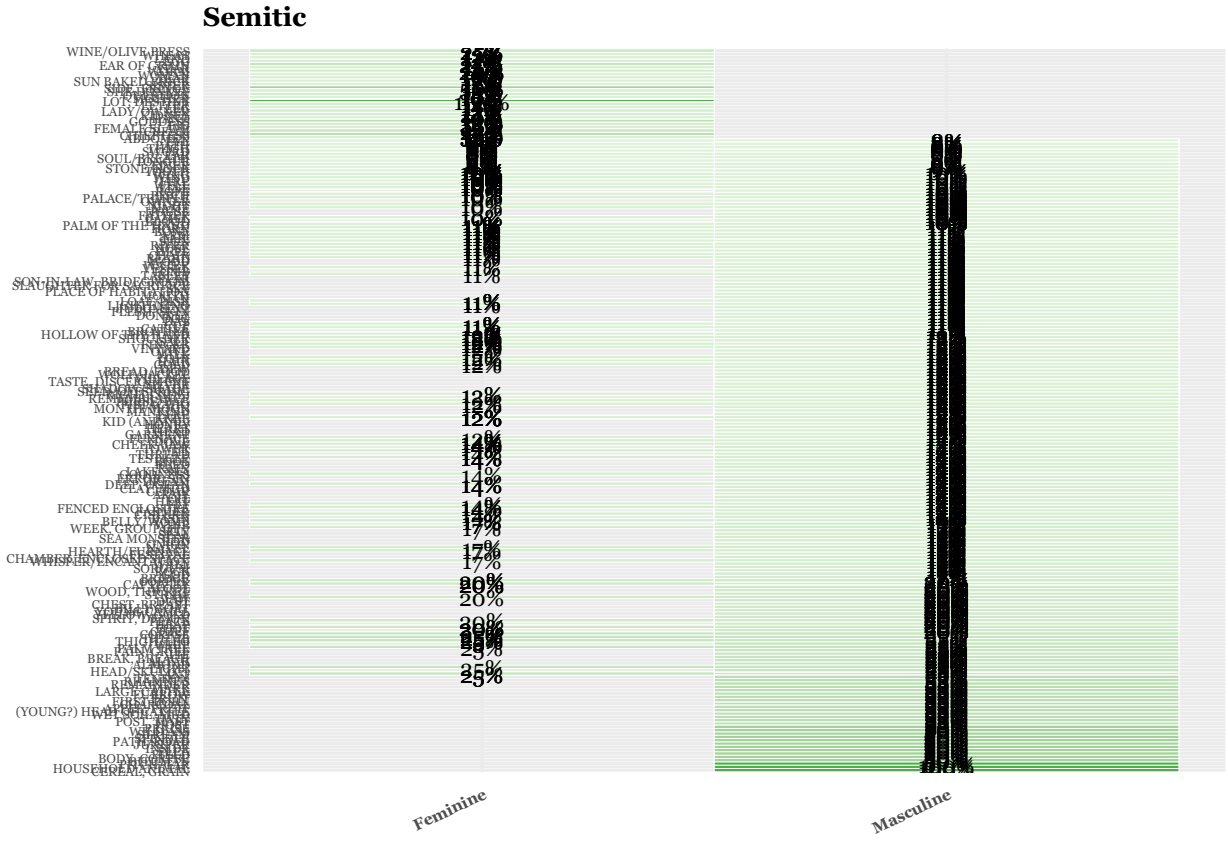
```r
#Indo-European
ie_heatmap <- IE %>%
  mutate(Gender = recode(Gender,
                         "C" = "Common",
                         "F" = "Feminine",
                         "M" = "Masculine",
                         "N" = "Neuter")) %>%
  mutate(Gender = factor(Gender, levels = rev(c("Masculine", "Feminine", "Neuter", "Common")))) %>%
  count(Concept, Gender) %>%
  group_by(Concept) %>%  # same as before, global proportion within this family
  mutate(Percentage = n / sum(n)) %>%
  ungroup()

# Calculate order of Concepts by Masculine proportion (descending)
concept_order_ie <- ie_heatmap %>%
  filter(Gender == "Masculine") %>%
  arrange(desc(Percentage)) %>%
  pull(Concept) %>%
  unique()

# Reorder Concept factor in heatmap_data
all_concepts_ie <- unique(ie_heatmap$Concept)
ie_heatmap <- ie_heatmap %>%
  mutate(Concept = factor(Concept, levels = c(concept_order_ie, setdiff(all_concepts_ie, concept_order_

#Plot
ggplot(ie_heatmap, aes(x = Gender, y = Concept, fill = Percentage)) +
  geom_tile(color = "white") +
  geom_text(aes(label = scales::percent(Percentage, accuracy = 1)),
            color = "black", size = 3, family = "Georgia") +
  scale_fill_gradient(low = "#e0f3db", high = "#4daf4a") +
  labs(
    title = "Indo-European",
    x = NULL,
    y = NULL
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 25, hjust = 1, size = 6, face = "bold"),
    text = element_text(family = "Georgia", size = 6),
    plot.title = element_text(size = 10, face = "bold"),
    legend.position = "none"  # remove legend
  )
```

Indo-European




```
ggsave("plots/Heatmap concept indo-european.png", width = 12, height = 18, dpi = 300)


#Semitic
semitic_heatmap <- Semitic %>%
  mutate(Gender = recode(Gender,
                         "C" = "Common",
                         "F" = "Feminine",
                         "M" = "Masculine",
                         "N" = "Neuter")) %>%
  mutate(Gender = factor(Gender, levels = rev(c("Masculine", "Feminine", "Neuter", "Common")))) %>%
  count(Family, Language, Concept, Gender) %>%
  group_by(Concept) %>%
  mutate(Percentage = n / sum(n)) %>%
  ungroup()

# Calculate order of Concepts by Masculine proportion (descending)
concept_order_sem <- semitic_heatmap %>%
  filter(Gender == "Masculine") %>%
  arrange(desc(Percentage)) %>%
  pull(Concept) %>%
  unique()

# Reorder Concept factor in heatmap_data
all_concepts_sem <- unique(semitic_heatmap$Concept)
semitic_heatmap <- semitic_heatmap %>%
```

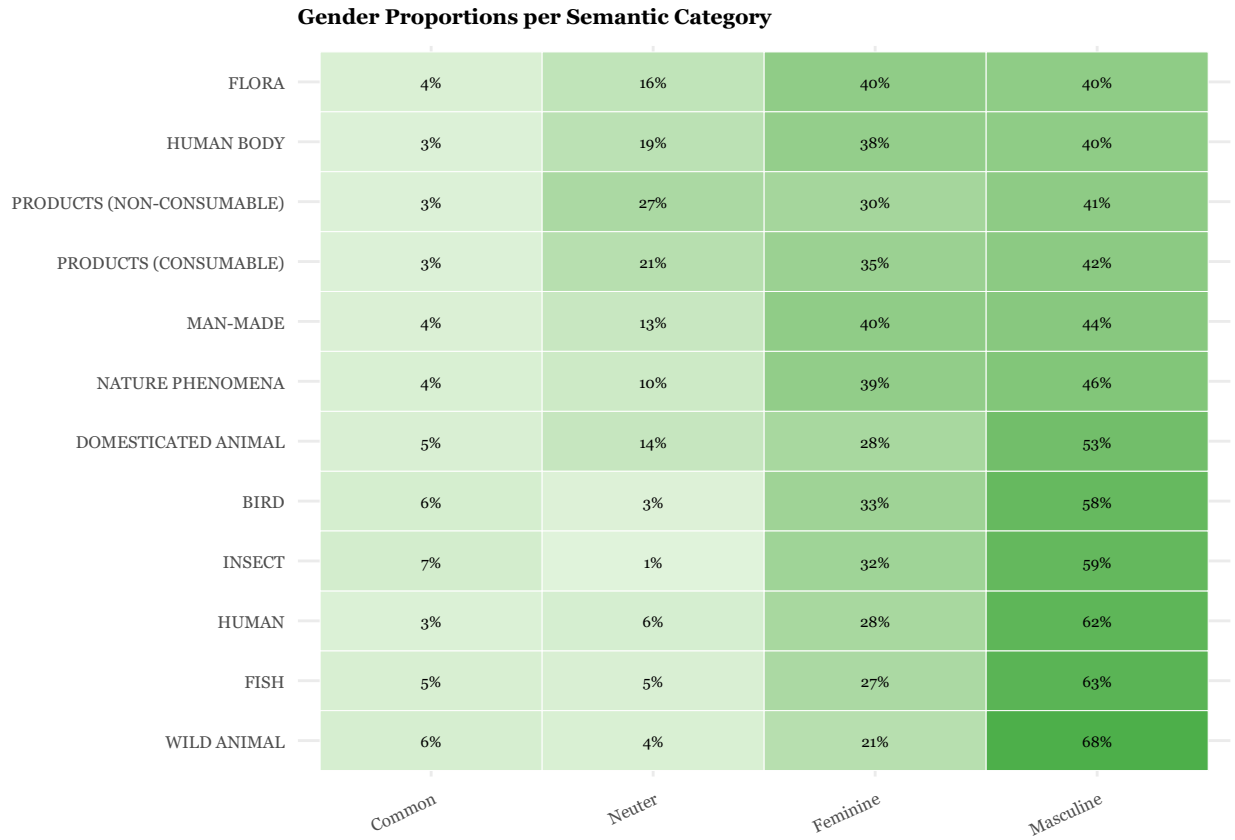
```
  mutate(Concept = factor(Concept, levels = c(concept_order_sem, setdiff(all_concepts_sem, concept_orde

#Plot
ggplot(semitic_heatmap, aes(x = Gender, y = Concept, fill = Percentage)) +
  geom_tile(color = "white") +
  geom_text(aes(label = scales::percent(Percentage, accuracy = 1)),
            color = "black", size = 3, family = "Georgia") +
  scale_fill_gradient(low = "#e0f3db", high = "#4daf4a") +
  labs(
    title = "Semitic",
    x = NULL,
    y = NULL
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 25, hjust = 1, size = 6, face = "bold"),
    text = element_text(family = "Georgia", size = 6),
    plot.title = element_text(size = 10, face = "bold"),
    legend.position = "none"  # remove legend
  )
```




```
ggsave("plots/Heatmap concept semitic.png", width = 12, height = 18, dpi = 300)
```

Heatmap of the Semantic classes per gender

```r
heatmap_data_rep <- gender_assignment %>%
  mutate(Gender = recode(Gender,
                         "C" = "Common",
                         "F" = "Feminine",
                         "M" = "Masculine",
                         "N" = "Neuter")) %>%
  mutate(Gender = factor(Gender, levels = rev(c("Masculine", "Feminine", "Neuter", "Common")))) %>%
  count(ReplaceConcept, Gender) %>%
  group_by(ReplaceConcept) %>%
  mutate(Percentage = n / sum(n)) %>%
  ungroup()

# Calculate order of Concepts by Masculine proportion (descending)
concept_order_rep <- heatmap_data_rep %>%
  filter(Gender == "Masculine") %>%
  arrange(desc(Percentage)) %>%
  pull(ReplaceConcept) %>%
  unique()

# Reorder ReplaceConcept factor in heatmap_data_rep
all_concepts_rep <- unique(heatmap_data_rep$ReplaceConcept)
heatmap_data_rep <- heatmap_data_rep %>%
  mutate(ReplaceConcept = factor(ReplaceConcept, levels = c(concept_order_rep, setdiff(all_concepts_rep

ggplot(heatmap_data_rep, aes(x = Gender, y = ReplaceConcept, fill = Percentage)) +
  geom_tile(color = "white") +
  geom_text(aes(label = scales::percent(Percentage, accuracy = 1)),
            color = "black", size = 2, family = "Georgia") +
  scale_fill_gradient(low = "#e0f3db", high = "#4daf4a") +
  labs(
    title = "Gender Proportions per Semantic Category",
    y = NULL,
    x = NULL
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 25, hjust = 1),
    text = element_text(family = "Georgia", size = 8),
    plot.title = element_text(size = 8 , face = "bold"),
    legend.position = "none"  # remove legend
  )
```

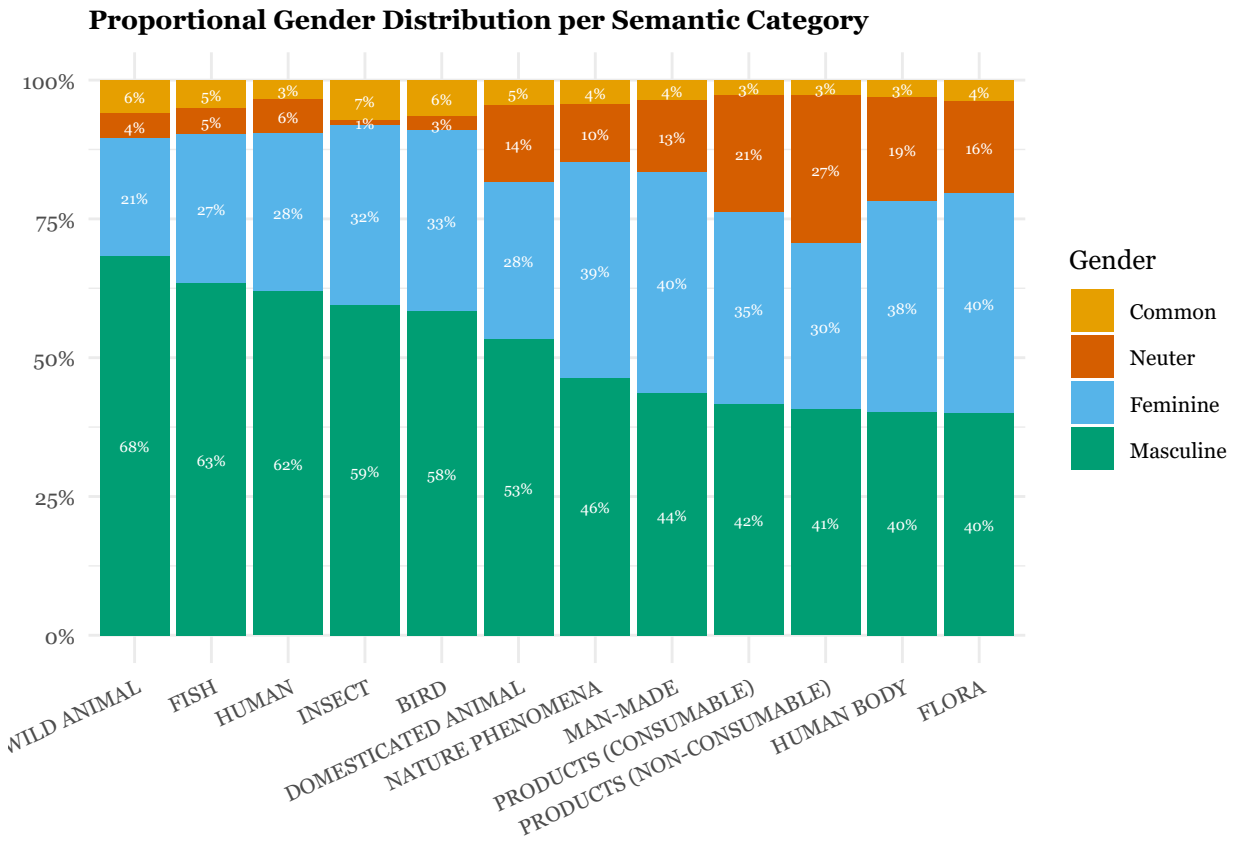



```
ggsave("plots/Heatmap ReplaceConcept.png", width = 8, height = 6, dpi = 300)
```

plot the same thing, but in a bar plot, as appears in the paper

```
#Stacked bar plot
ggplot(heatmap_data_rep, aes(x = ReplaceConcept, y = Percentage, fill = Gender)) +
  geom_col(position = "fill") +  # fill scales bars to 100%
  geom_text(
    aes(label = scales::percent(Percentage, accuracy = 1)),
    position = position_fill(vjust = 0.5),
    color = "white",
    size = 2,
    family = "Georgia"
  ) +
  scale_y_continuous(labels = scales::percent_format()) +
  scale_fill_manual(values = c(
    "Masculine" = "#009E73",
    "Feminine" = "#56B4E9",
    "Neuter"    = "#D55E00",
    "Common"    = "#E69F00"
  )) +
  labs(
    title = "Proportional Gender Distribution per Semantic Category",
    x = NULL,
    y = NULL,
```

```
    fill = "Gender"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 25, hjust = 1),
    text = element_text(family = "Georgia", size = 10),
    plot.title = element_text(size = 10, face = "bold")
  )
```



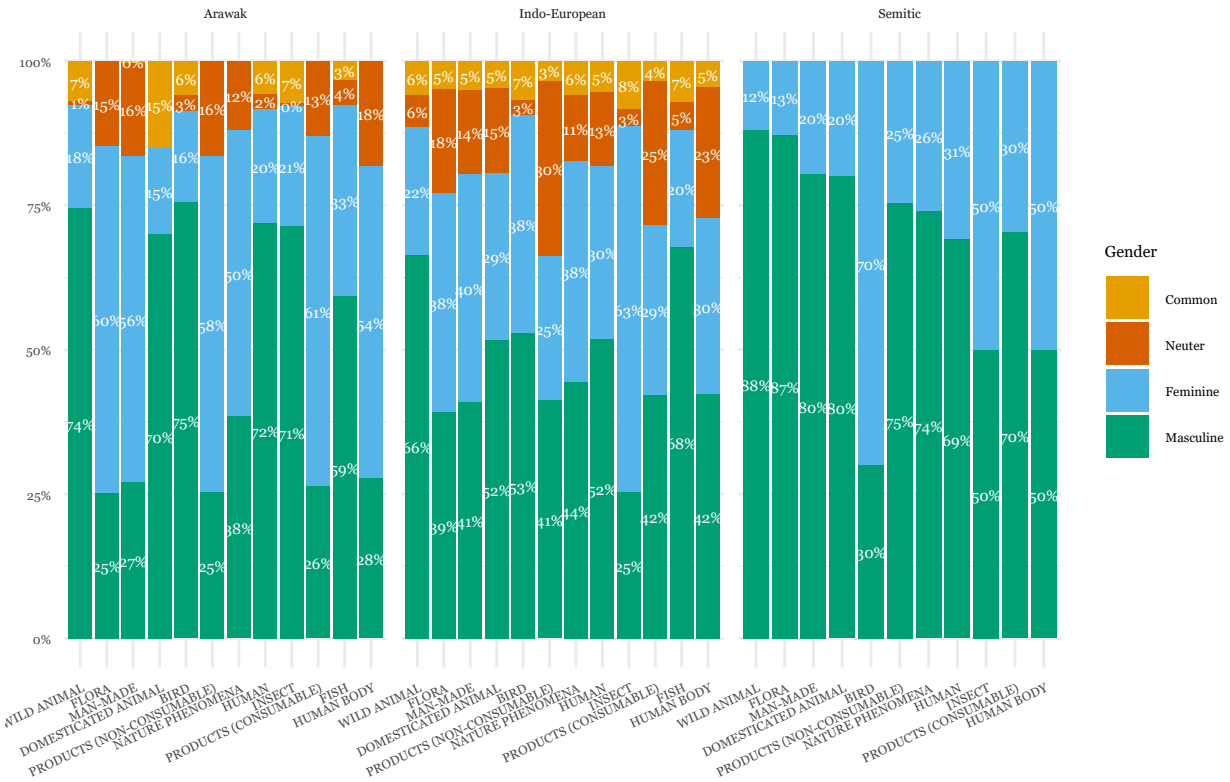

```
ggsave("plots/barplot ReplaceConcept.png", width = 8, height = 6, dpi = 300)
```

Plot per family

```
#plot per family:
bar_data_rep_fam <- gender_assignment %>%
  mutate(Gender = recode(Gender,
                         "C" = "Common",
                         "F" = "Feminine",
                         "M" = "Masculine",
                         "N" = "Neuter")) %>%
  mutate(Gender = factor(Gender, levels = rev(c("Masculine", "Feminine", "Neuter", "Common")))) %>%
  count(Family, ReplaceConcept, Gender) %>%
  group_by(Family, ReplaceConcept) %>%
  mutate(Percentage = n / sum(n)) %>%
  ungroup()
```

```r
# Calculate order of Concepts by Masculine proportion (descending)
concept_order_rep_fam <- bar_data_rep_fam %>%
  filter(Gender == "Masculine") %>%
  arrange(desc(Percentage)) %>%
  pull(ReplaceConcept) %>%
  unique()

# Reorder ReplaceConcept factor in bar_data_rep_fam
all_concepts_rep_fam <- unique(bar_data_rep_fam$ReplaceConcept)
bar_data_rep_fam <- bar_data_rep_fam %>%
  mutate(ReplaceConcept = factor(ReplaceConcept, levels = c(concept_order_rep_fam, setdiff(all_concepts

ggplot(bar_data_rep_fam, aes(x = ReplaceConcept, y = Percentage, fill = Gender)) +
  geom_col(position = "fill") +
  geom_text(aes(label = scales::percent(Percentage, accuracy = 1)),
            position = position_fill(vjust = 0.5),
            color = "white", size = 2, family = "Georgia") +
  scale_y_continuous(labels = scales::percent_format()) +
  scale_fill_manual(values = c(
    "Masculine" = "#009E73",
    "Feminine" = "#56B4E9",
    "Neuter" = "#D55E00",
    "Common" = "#E69F00"
  )) +
  labs(
    title = "Proportional Gender Distribution per Semantic Category",
    x = NULL,
    y = NULL,
    fill = "Gender"
  ) +
  facet_wrap(~ Family, scales = "free_x") +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 25, hjust = 1),
    text = element_text(family = "Georgia", size = 6),
    plot.title = element_text(size = 10, face = "bold")
  )
```

Proportional Gender Distribution per Semantic Category




```
ggsave("plots/barplot ReplaceConcept per family.png", width = 8, height = 6, dpi = 300)
```

2. Statistical Analysis

In this section we compare the level of gender cohesion of each of the 12 Semantic classes against the level of gender cohesion in the full dataset. First, we use mutual information which measures how much knowing the Concept reduces uncertainty about Gender (and vice versa). The level of MI in the full dataset serves as the Null Hypothesis, against which we compare the MI of the semantic classes.

First, we generate the full dataset

```
gen_sem_con <- gender_assignment %>%
  select("Glottocode", "Concept", "Gender", "ReplaceConcept", "Family", "Animacy", "Culture") %>%
  # assign all genders into a new gender column as 4 values of the same variable
  mutate(Gender = replace(Gender, Gender == "M", 1)) %>%
  mutate(Gender = replace(Gender, Gender == "F", 2)) %>%
  mutate(Gender = replace(Gender, Gender == "N", 3)) %>%
  mutate(Gender = replace(Gender, Gender == "C", 4)) %>%
  # change to data frame
  as.data.frame() %>%
  drop_na() #remove NAs
```

2.1. Semantic Classes

2.1.1 Mutual Information

Compute MI for the full dataset and the 12 semantic classes. Interpretation: Bars above 0 means that the group is more cohesive than full dataset. Bars below 0 means that it is less cohesive.

```
# Ensure both are factors
gender <- as.factor(gen_sem_con$Gender)
replace_concept <- as.factor(gen_sem_con$ReplaceConcept)

# Observed MI
obs_mi <- mutinformation(replace_concept, gender)
obs_mi
```

```
## [1] 0.03293121
```

Sanity check: I shuffle the Gender column many times (e.g., 1000 permutations) and recompute MI each time, to compare my data to a random distribution. If $p < 0.05 \rightarrow$ the MI is significantly higher than chance, i.e., knowing ReplaceConcept actually reduces uncertainty about Gender.

```
set.seed(123)
n_perm <- 1000
perm_mi <- numeric(n_perm)

for (i in 1:n_perm) {
  shuffled_gender <- sample(gen_sem_con$Gender)
  perm_mi[i] <- mutinformation(
    as.factor(gen_sem_con$ReplaceConcept),
    as.factor(shuffled_gender)
  )
}

mean(perm_mi >= obs_mi)  # gives a p-value
```

```
## [1] 0
```

```
# p=0
#mean(perm_mi >= obs_mi) = 0 means that in none of the 1000 permutations did random shuffling produce a
#more detailed results:
mean(perm_mi)      # average MI under permutation
```

```
## [1] 0.001040978
```

```
range(perm_mi)     # min and max
```

```
## [1] 0.0004310876 0.0019033926
```

```
obs_mi             # observed MI
```

```
## [1] 0.03293121
```

```r
p_value <- (sum(perm_mi >= obs_mi) + 1)/(length(perm_mi) + 1)
p_value
```

```
## [1] 0.000999001
```

2.1.2. Entropy

Second, to check differences between categories, we compute entropy of Gender within each ReplaceConcept: Lower entropy → more predictable gender within that category → more cohesive. Higher entropy → more variation → less cohesive.

```r
# Make sure Gender is a factor
gender_factor <- as.factor(gen_sem_con$Gender)

# Compute Shannon entropy in bits:
#Shannon entropy of the Gender variable within that ReplaceConcept. Measures how unpredictable Gender i
#0 → all items have the same Gender (perfect cohesion)
#Maximum → all Genders are equally frequent (least cohesion)

full_entropy <- {
  probs <- table(gender_factor) / length(gender_factor)
  -sum(probs * log2(probs))
}

full_entropy
```

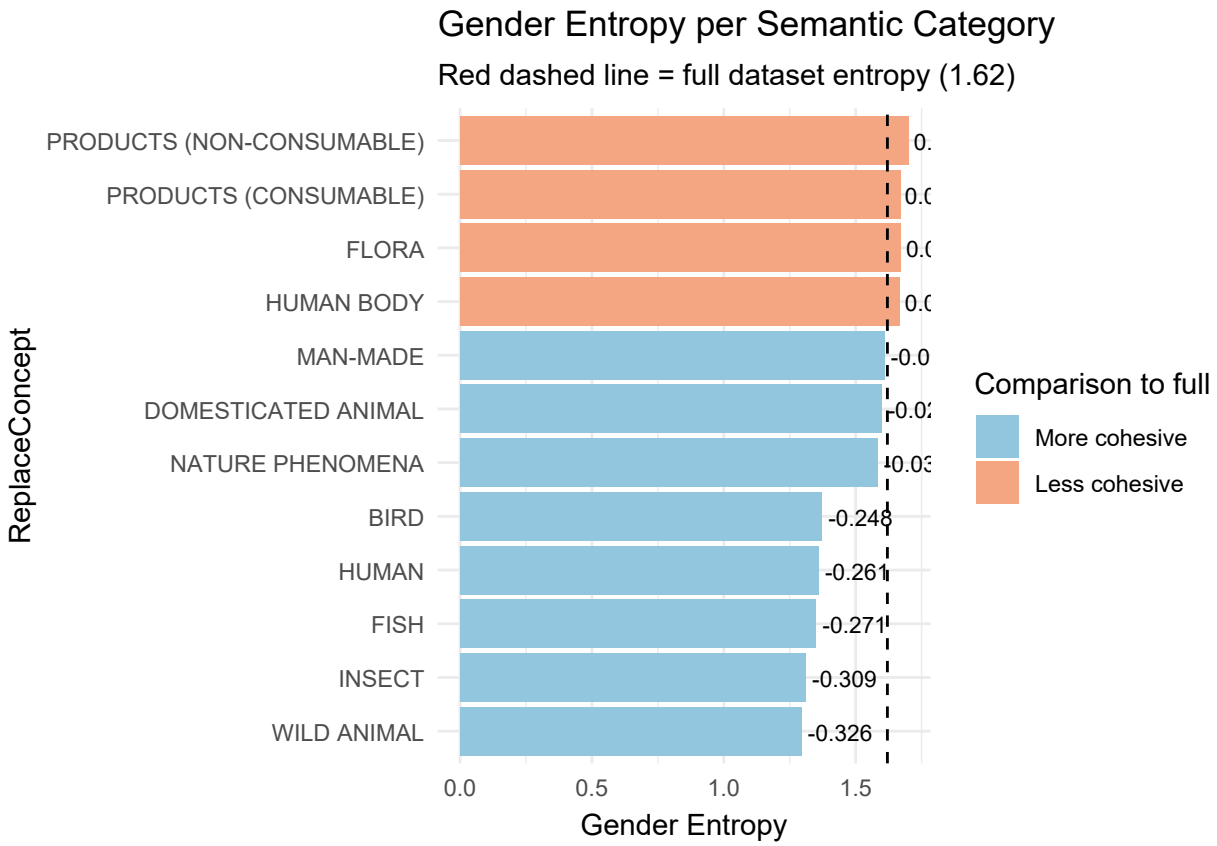
```
## [1] 1.620263
```

```r
group_entropy <- gen_sem_con %>%
  group_by(ReplaceConcept) %>%
  summarise(
    gender_entropy = {
      probs <- table(Gender) / n()
      -sum(probs * log2(probs))
    },
    n = n(),
    .groups = "drop"
  ) %>%
  mutate(diff_from_full = gender_entropy - full_entropy) %>%
  arrange(gender_entropy)

group_entropy
```

```
## # A tibble: 12 x 4
##    ReplaceConcept            gender_entropy     n diff_from_full
##    <chr>                              <dbl> <int>          <dbl>
##  1 WILD ANIMAL                         1.29  1494        -0.326
##  2 INSECT                              1.31   306        -0.309
##  3 FISH                                1.35   175        -0.271
##  4 HUMAN                               1.36   665        -0.261
##  5 BIRD                                1.37   586        -0.248
```

```
##  6 NATURE PHENOMENA                   1.58  2179        -0.0377
##  7 DOMESTICATED ANIMAL                1.60  1663        -0.0232
##  8 MAN-MADE                           1.61  2449        -0.0115
##  9 HUMAN BODY                         1.66  2603         0.0447
## 10 FLORA                              1.67   791         0.0490
## 11 PRODUCTS (CONSUMABLE)              1.67  1851         0.0497
## 12 PRODUCTS (NON-CONSUMABLE)          1.70  1175         0.0787
```

```r
#Difference from full dataset: Interpreted as:
#Negative → group is more cohesive than the overall dataset
#Positive → group is less cohesive than the overall dataset


ggplot(group_entropy, aes(
  x = reorder(ReplaceConcept, gender_entropy),
  y = gender_entropy,
  fill = diff_from_full > 0  # TRUE = less cohesive, FALSE = more cohesive
)) +
  geom_col() +
  geom_hline(yintercept = full_entropy, color = "black", linetype = "dashed") +
  coord_flip() +
  scale_fill_manual(
    values = c("TRUE" = "#f4a582", "FALSE" = "#92c5de"),
    labels = c("More cohesive","Less cohesive")
  ) +
  geom_text(aes(label = round(diff_from_full,3)), hjust = -0.1, size = 3) +
  labs(
    title = "Gender Entropy per Semantic Category",
    subtitle = paste0("Red dashed line = full dataset entropy (", round(full_entropy, 3), ")"),
    x = "ReplaceConcept",
    y = "Gender Entropy",
    fill = "Comparison to full"
  ) +
  theme_minimal()
```

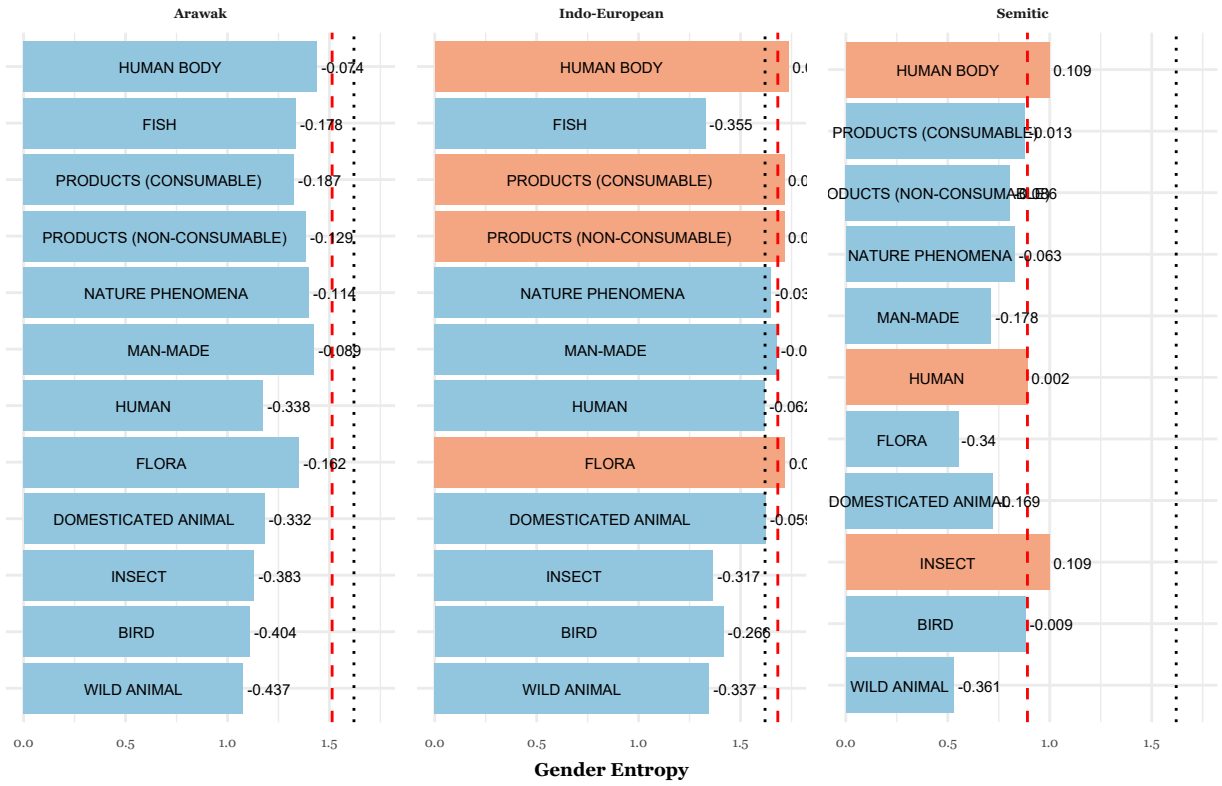


Per language family:

```r
# Function to compute Shannon entropy
shannon_entropy <- function(x) {
  probs <- table(x) / length(x)
  -sum(probs * log2(probs))
}

# Full dataset entropy (all families)
full_dataset_entropy <- shannon_entropy(gen_sem_con$Gender)

# Compute full entropy per family and per-group entropy
all_entropy <- gen_sem_con %>%
  group_by(Family) %>%
  summarise(full_entropy = shannon_entropy(Gender), .groups = "drop") %>%
  left_join(
    gen_sem_con %>%
      group_by(Family, ReplaceConcept) %>%
      summarise(
        gender_entropy = shannon_entropy(Gender),
        n = n(),
        .groups = "drop"
      ),
    by = "Family"
  ) %>%
  mutate(
    diff_from_full = gender_entropy - full_entropy
```

```
  )


ggplot(all_entropy, aes(
  x = reorder(ReplaceConcept, gender_entropy),
  y = gender_entropy,
  fill = diff_from_full > 0
)) +
  geom_col() +
  # Put ReplaceConcept names inside the bars
  geom_text(aes(label = ReplaceConcept),
            position = position_stack(vjust = 0.5),  # vertically centered
            hjust = 0.5,                             # horizontally centered inside bar
            size = 2,
            color = "black") +                        # black text for contrast
  # Family-specific entropy line
  geom_hline(aes(yintercept = full_entropy), color = "red", linetype = "dashed") +
  # Full-dataset entropy line
  geom_hline(yintercept = full_dataset_entropy, color = "black", linetype = "dotted") +
  coord_flip() +
  scale_fill_manual(
    values = c("TRUE" = "#f4a582", "FALSE" = "#92c5de"),
    labels = c("More cohesive","Less cohesive")
  ) +
  geom_text(aes(label = round(diff_from_full,3)),
            hjust = -0.1, size = 2) +  # keeps diff labels outside the bars
  facet_wrap(~Family, scales = "free_y") +
  labs(
    title = "Gender Entropy per semantic catefory",
    #subtitle = paste0("Red dashed line = full-family entropy, Black dotted line = full-dataset entropy
    #                  round(full_dataset_entropy,3), ")"),
    x = NULL,
    y = "Gender Entropy",
    fill = "Comparison to full"
  ) +
  theme_minimal() +
  theme(
    legend.position = "none",
    axis.text.y = element_blank(),  # hide default y-axis labels
    plot.title = element_text(size = 10, family = "Georgia"),
    axis.title.x = element_text(size = 7, face = "bold", family = "Georgia"),
    axis.text.x = element_text(size = 5, family = "Georgia"),
    strip.text = element_text(size = 5, face = "bold", family = "Georgia")
  )
```

Gender Entropy per semantic catefory




```
ggsave("plots/Entropy semantic category per family.png", width = 8, height = 4, dpi = 300)
```

2.1.3. Chi-Square

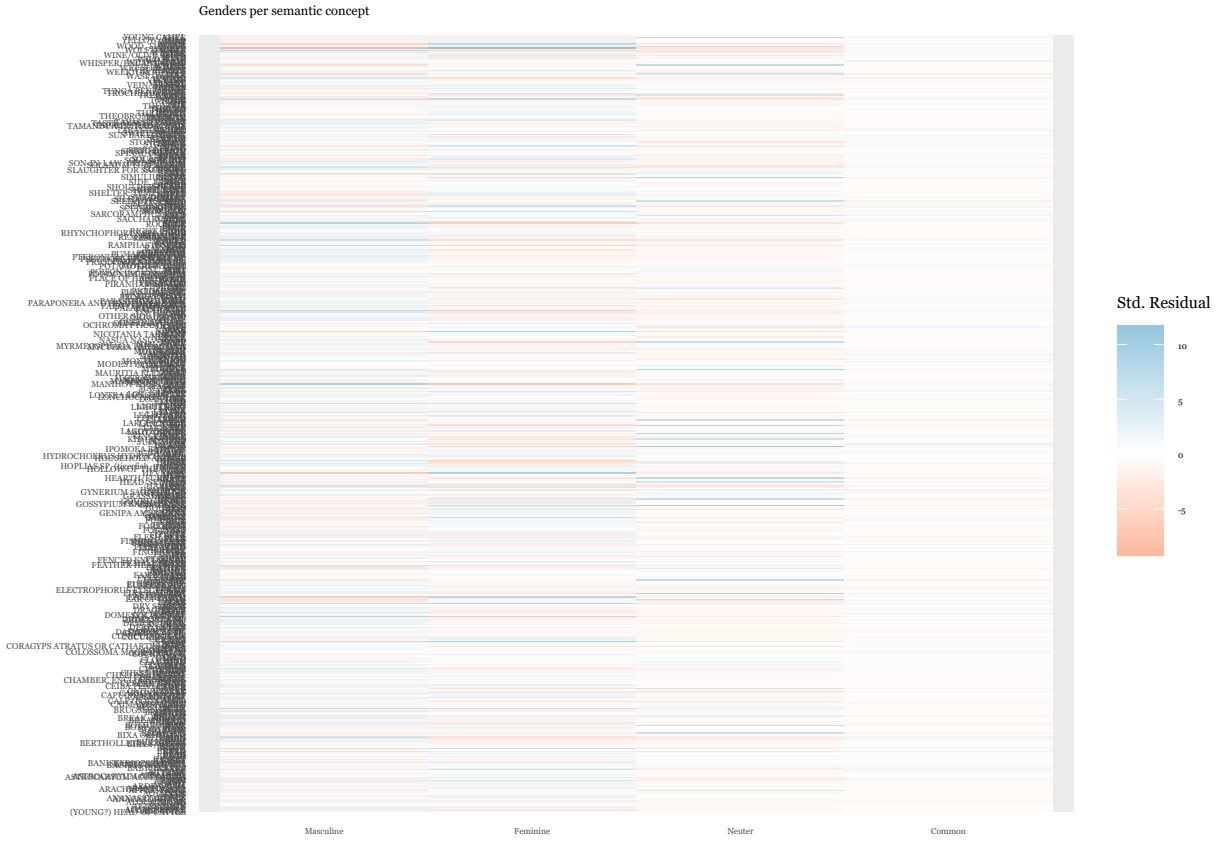
Next, we want to see if the differences in distributions are significant, per concept

```
contingency_table <- heatmap_data %>%
  select(Concept, Gender, n) %>%
  pivot_wider(names_from = Gender, values_from = n, values_fill = 0) %>%
  column_to_rownames("Concept") %>%
  as.matrix()

chisq_result <- chisq.test(contingency_table)
print(chisq_result)
```

```
##
##  Pearson's Chi-squared test
##
## data:  contingency_table
## X-squared = 6239.7, df = 1779, p-value < 2.2e-16
```

```
residuals_df <- as.data.frame(as.table(chisq_result$stdres)) %>%
  rename(Concept = Var1, Gender = Var2, Residual = Freq)
```

```
ggplot(residuals_df, aes(x = Gender, y = Concept, fill = Residual)) +
  geom_tile(color = NA) +
  scale_fill_gradient2(
    low = "#f4a582", mid = "white", high = "#92c5de", midpoint = 0,
    name = "Std. Residual"
  ) +
  labs(
    title = "Genders per semantic concept",
    x = NULL,
    y = NULL
  ) +
  theme_minimal() +
  theme(
    text = element_text(family = "Georgia", size = 4),
    axis.text.x = element_text(angle = 0, hjust = 0.5),
    legend.title = element_text(size = 6)
  )
```




```
ggsave("plots/chi-sq residuals per concept.png", width = 8, height = 4, dpi = 300)
```

Now we want to see if the differences in distributions are significant, within the Semantic Classes

```
contingency_table_rep <- heatmap_data_rep %>%
  select(ReplaceConcept, Gender, n) %>%
```

```r
  pivot_wider(names_from = Gender, values_from = n, values_fill = 0) %>%
  column_to_rownames("ReplaceConcept") %>%
  as.matrix()

# Run chi-square test
chisq_result_rep <- chisq.test(contingency_table_rep)
print(chisq_result_rep)
```

```
##
##  Pearson's Chi-squared test
##
## data:  contingency_table_rep
## X-squared = 995.59, df = 33, p-value < 2.2e-16
```

```r
##Because we have some concepts with very few data points, the chisq test spits out a warning. To make
filtered_table <- contingency_table_rep[rowSums(contingency_table_rep) >= 5, ]
chisq_result_rep_filtered <- chisq.test(filtered_table)
print(chisq_result_rep_filtered)
```

```
##
##  Pearson's Chi-squared test
##
## data:  filtered_table
## X-squared = 995.59, df = 33, p-value < 2.2e-16
```

```r
# Extract standardized residuals
residuals_df <- as.data.frame(as.table(chisq_result_rep$stdres))
colnames(residuals_df) <- c("ReplaceConcept", "Gender", "Residual")

# Order by strongest deviation
concept_order <- residuals_df %>%
  group_by(ReplaceConcept) %>%
  summarize(mean_abs_res = mean(abs(Residual), na.rm = TRUE)) %>%
  arrange(desc(mean_abs_res)) %>%
  pull(ReplaceConcept)

residuals_df <- residuals_df %>%
  mutate(ReplaceConcept = factor(ReplaceConcept, levels = concept_order))

# Plot standardized residuals with numbers
ggplot(residuals_df, aes(x = Gender, y = ReplaceConcept, fill = Residual)) +
  geom_tile(color = "grey95", size = 0.1) +
  geom_text(
    aes(label = round(Residual, 1)),
    size = 2, color = "gray30", family = "Georgia"
  ) +
  scale_fill_gradient2(
    low = "#f4a582", mid = "white", high = "#92c5de", midpoint = 0,
    name = "Std. Residual"
  ) +
  labs(
    title = "Standardized Residuals: Gender Distribution by Semantic Category",
```

```
    x = NULL, y = NULL
  ) +
  theme_minimal(base_family = "Georgia") +
  theme(
    axis.text.x = element_text(angle = 0, hjust = 0.5, size = 8, face = "bold"),
    axis.text.y = element_text(size = 5),
    plot.title = element_text(size = 10, face = "bold"),
    legend.position = "right",
    legend.title = element_text(size = 6)
  )
```


```
ggsave("plots/chi-sq residuals per ReplaceConcept.png", width = 8, height = 4, dpi = 300)
```

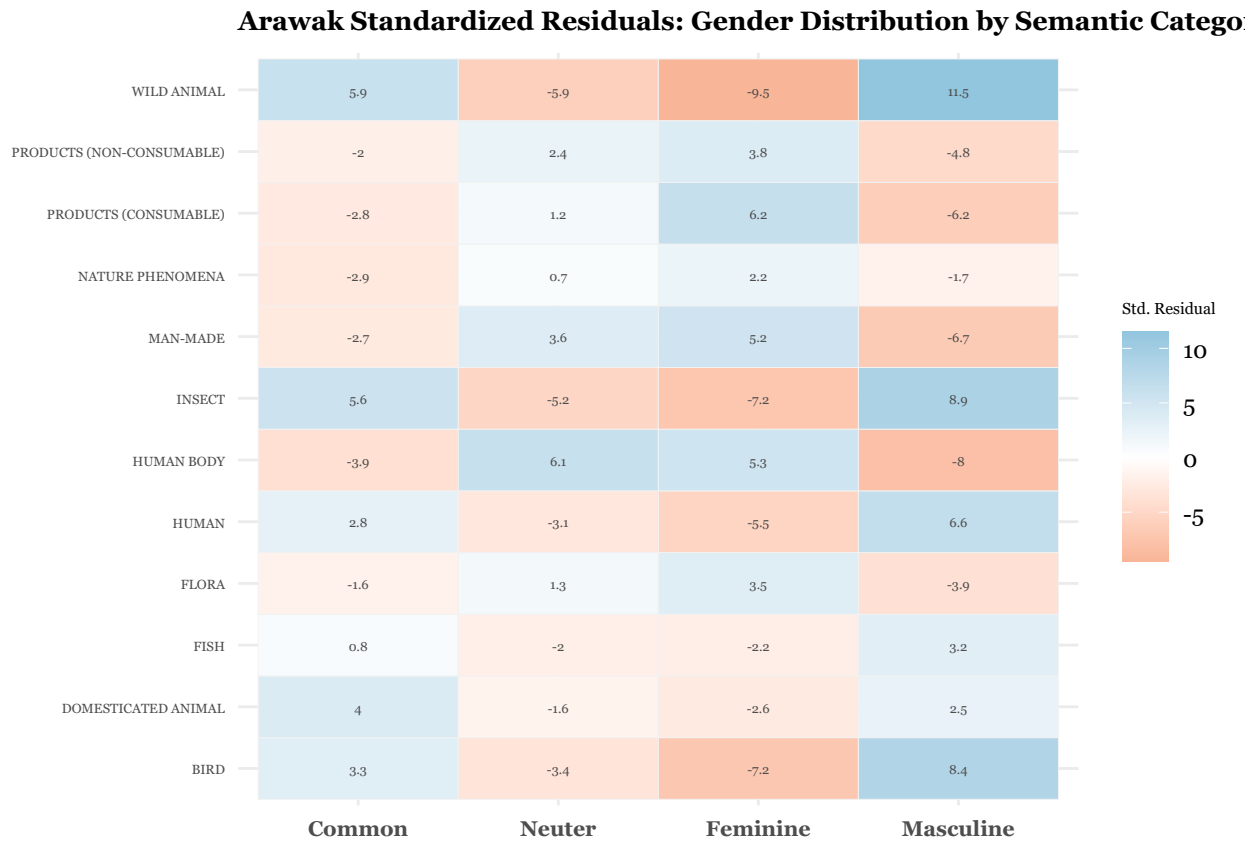
Per family:

```
#### Arawak #####
arawak_rep <- Arawak %>%
  mutate(Gender = recode(Gender,
                         "C" = "Common",
                         "F" = "Feminine",
                         "M" = "Masculine",
                         "N" = "Neuter")) %>%
  mutate(Gender = factor(Gender, levels = rev(c("Masculine", "Feminine", "Neuter", "Common")))) %>%
  count(ReplaceConcept, Gender) %>%
  group_by(ReplaceConcept) %>%
```

```r
  mutate(Percentage = n / sum(n)) %>%
  ungroup()

contingency_table_rep_ar <- arawak_rep %>%
  select(ReplaceConcept, Gender, n) %>%
  pivot_wider(names_from = Gender, values_from = n, values_fill = 0) %>%
  column_to_rownames("ReplaceConcept") %>%
  as.matrix()

# Run chi-square test
chisq_result_rep_ar <- chisq.test(contingency_table_rep_ar)
print(chisq_result_rep_ar)
```

```
##
##  Pearson's Chi-squared test
##
## data:  contingency_table_rep_ar
## X-squared = 704.69, df = 33, p-value < 2.2e-16
```

```r
# Extract standardized residuals
residuals_rep_ar <- as.data.frame(as.table(chisq_result_rep_ar$stdres))
colnames(residuals_rep_ar) <- c("ReplaceConcept", "Gender", "Residual")

# Plot standardized residuals with numbers
ggplot(residuals_rep_ar, aes(x = Gender, y = ReplaceConcept, fill = Residual)) +
  geom_tile(color = "grey95", size = 0.1) +
  geom_text(
    aes(label = round(Residual, 1)),
    size = 1.7, color = "gray30", family = "Georgia"
  ) +
  scale_fill_gradient2(
    low = "#f4a582", mid = "white", high = "#92c5de", midpoint = 0,
    name = "Std. Residual"
  ) +
  labs(
    title = "Arawak Standardized Residuals: Gender Distribution by Semantic Category",
    x = NULL, y = NULL
  ) +
  theme_minimal(base_family = "Georgia") +
  theme(
    axis.text.x = element_text(angle = 0, hjust = 0.5, size = 8, face = "bold"),
    axis.text.y = element_text(size = 5),
    plot.title = element_text(size = 10, face = "bold"),
    legend.position = "right",
    legend.title = element_text(size = 6)
  )
```

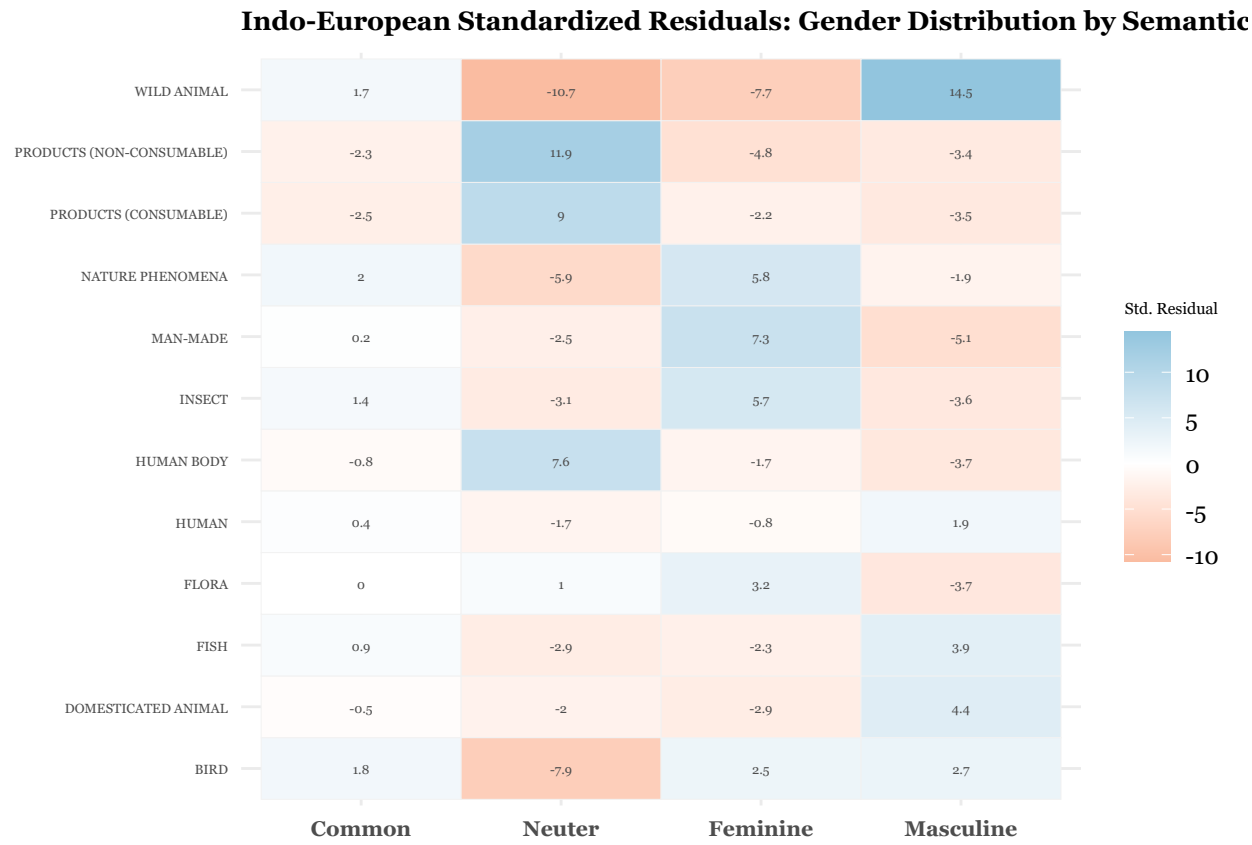



```
ggsave("plots/Arawak chi-sq residuals per ReplaceConcept.png", width = 8, height = 4, dpi = 300)

#### Indo-European #####
ie_rep <- IE %>%
  mutate(Gender = recode(Gender,
                         "C" = "Common",
                         "F" = "Feminine",
                         "M" = "Masculine",
                         "N" = "Neuter")) %>%
  mutate(Gender = factor(Gender, levels = rev(c("Masculine", "Feminine", "Neuter", "Common")))) %>%
  count(ReplaceConcept, Gender) %>%
  group_by(ReplaceConcept) %>%
  mutate(Percentage = n / sum(n)) %>%
  ungroup()

contingency_table_rep_ie <- ie_rep %>%
  select(ReplaceConcept, Gender, n) %>%
  pivot_wider(names_from = Gender, values_from = n, values_fill = 0) %>%
  column_to_rownames("ReplaceConcept") %>%
  as.matrix()

# Run chi-square test
chisq_result_rep_ie <- chisq.test(contingency_table_rep_ie)
print(chisq_result_rep_ie)
```

```
##
```

```
##  Pearson's Chi-squared test
##
## data:  contingency_table_rep_ie
## X-squared = 733.51, df = 33, p-value < 2.2e-16
```

```r
# Extract standardized residuals
residuals_rep_ie <- as.data.frame(as.table(chisq_result_rep_ie$stdres))
colnames(residuals_rep_ie) <- c("ReplaceConcept", "Gender", "Residual")

# Plot standardized residuals with numbers
ggplot(residuals_rep_ie, aes(x = Gender, y = ReplaceConcept, fill = Residual)) +
  geom_tile(color = "grey95", size = 0.1) +
  geom_text(
    aes(label = round(Residual, 1)),
    size = 1.7, color = "gray30", family = "Georgia"
  ) +
  scale_fill_gradient2(
    low = "#f4a582", mid = "white", high = "#92c5de", midpoint = 0,
    name = "Std. Residual"
  ) +
  labs(
    title = "Indo-European Standardized Residuals: Gender Distribution by Semantic Category",
    x = NULL, y = NULL
  ) +
  theme_minimal(base_family = "Georgia") +
  theme(
    axis.text.x = element_text(angle = 0, hjust = 0.5, size = 8, face = "bold"),
    axis.text.y = element_text(size = 5),
    plot.title = element_text(size = 10, face = "bold"),
    legend.position = "right",
    legend.title = element_text(size = 6)
  )
```




```
ggsave("plots/IE chi-sq residuals per ReplaceConcept.png", width = 8, height = 4, dpi = 300)

#### Semitic #####
semitic_rep <- Semitic %>%
  mutate(Gender = recode(Gender,
                         "C" = "Common",
                         "F" = "Feminine",
                         "M" = "Masculine",
                         "N" = "Neuter")) %>%
  mutate(Gender = factor(Gender, levels = rev(c("Masculine", "Feminine", "Neuter", "Common")))) %>%
  count(ReplaceConcept, Gender) %>%
  group_by(ReplaceConcept) %>%
  mutate(Percentage = n / sum(n)) %>%
  ungroup()

contingency_table_rep_sem <- semitic_rep %>%
  select(ReplaceConcept, Gender, n) %>%
  pivot_wider(names_from = Gender, values_from = n, values_fill = 0) %>%
  column_to_rownames("ReplaceConcept") %>%
  as.matrix()

# Run chi-square test
chisq_result_rep_sem <- chisq.test(contingency_table_rep_sem)
print(chisq_result_rep_sem)
```

```
##
```

```
##  Pearson's Chi-squared test
##
## data:  contingency_table_rep_sem
## X-squared = 105.52, df = 10, p-value < 2.2e-16
```

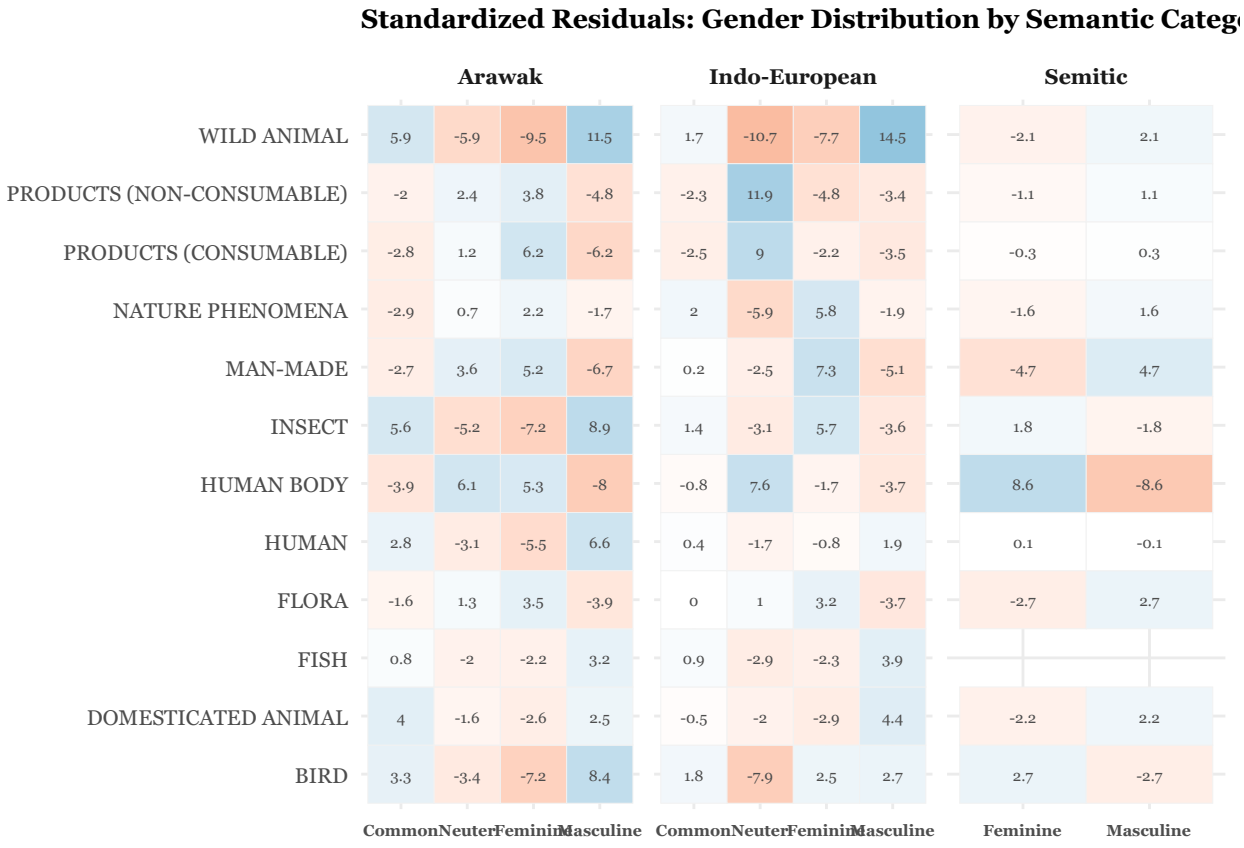
```r
# Extract standardized residuals
residuals_rep_sem <- as.data.frame(as.table(chisq_result_rep_sem$stdres))
colnames(residuals_rep_sem) <- c("ReplaceConcept", "Gender", "Residual")

# Plot standardized residuals with numbers
ggplot(residuals_rep_sem, aes(x = Gender, y = ReplaceConcept, fill = Residual)) +
  geom_tile(color = "grey95", size = 0.1) +
  geom_text(
    aes(label = round(Residual, 1)),
    size = 1.7, color = "gray30", family = "Georgia"
  ) +
  scale_fill_gradient2(
    low = "#f4a582", mid = "white", high = "#92c5de", midpoint = 0,
    name = "Std. Residual"
  ) +
  labs(
    title = "Semitic Standardized Residuals: Gender Distribution by Semantic Category",
    x = NULL, y = NULL
  ) +
  theme_minimal(base_family = "Georgia") +
  theme(
    axis.text.x = element_text(angle = 0, hjust = 0.5, size = 8, face = "bold"),
    axis.text.y = element_text(size = 5),
    plot.title = element_text(size = 10, face = "bold"),
    legend.position = "right",
    legend.title = element_text(size = 6)
  )
```


```
ggsave("plots/Semitic chi-sq residuals per ReplaceConcept.png", width = 8, height = 4, dpi = 300)
```

in one plot:

```
# Combine residuals from all families
residuals_all <- bind_rows(
  residuals_rep_ar %>% mutate(Family = "Arawak"),
  residuals_rep_ie %>% mutate(Family = "Indo-European"),
  residuals_rep_sem %>% mutate(Family = "Semitic")
)

# Remove zero or missing combinations if needed
residuals_all <- residuals_all %>% filter(!is.na(Residual))

# Make ReplaceConcept a factor with shared order
residuals_all$ReplaceConcept <- factor(residuals_all$ReplaceConcept,
                                       levels = unique(residuals_all$ReplaceConcept))

# Plot
ggplot(residuals_all, aes(x = Gender, y = ReplaceConcept, fill = Residual)) +
  geom_tile(color = "grey95", size = 0.1) +
  geom_text(aes(label = round(Residual, 1)),
            size = 2, color = "gray30", family = "Georgia") +
  scale_fill_gradient2(low = "#f4a582", mid = "white", high = "#92c5de", midpoint = 0) +
  facet_wrap(~Family, scales = "free_x") +
```

```
  theme_minimal(base_family = "Georgia") +
  theme(
    axis.text.x = element_text(angle = 0, hjust = 0.5, size = 6, face = "bold"),
    axis.text.y = element_text(size = 8),
    strip.text = element_text(size = 8, face = "bold"),
    plot.title = element_text(size = 10, face = "bold"),
    legend.position = "none"  # removes the legend
  ) +
  labs(
    title = "Standardized Residuals: Gender Distribution by Semantic Category",
    x = NULL, y = NULL
  )
```




```
ggsave("plots/chi-sq residuals per ReplaceConcept all families.png", width = 8, height = 4, dpi = 300)
```

2.2. Animacy/Culture

Now we do the same procedure for the Animacy and Culture distinctions.

2.2.1.Mutual Information:

```r
###animacy###
# Ensure both are factors
gender <- as.factor(gen_sem_con$Gender)
mi_animacy <- as.factor(gen_sem_con$Animacy)

# Observed MI
obs_mi_an <- mutinformation(mi_animacy, gender)
obs_mi_an
```

```
## [1] 0.0201449
```

```r
#I shuffle the Gender column many times (e.g., 1000 permutations) and recompute MI each time, to compar
set.seed(123)
n_perm <- 1000
perm_mi <- numeric(n_perm)

for (i in 1:n_perm) {
  shuffled_gender <- sample(gen_sem_con$Gender)
  perm_mi[i] <- mutinformation(
    as.factor(gen_sem_con$Animacy),
    as.factor(shuffled_gender)
  )
}

mean(perm_mi >= obs_mi_an)  # gives a p-value
```

```
## [1] 0
```

```r
# p=0
#mean(perm_mi >= obs_mi) = 0 means that in none of the 1000 permutations did random shuffling produce a

#more detailed results:
mean(perm_mi)      # average MI under permutation
```

```
## [1] 9.312318e-05
```

```r
range(perm_mi)     # min and max
```

```
## [1] 1.407165e-06 5.675086e-04
```

```r
obs_mi_an             # observed MI
```

```
## [1] 0.0201449
```

```r
p_value <- (sum(perm_mi >= obs_mi_an) + 1)/(length(perm_mi) + 1)
p_value
```

```
## [1] 0.000999001
```

```r
###culture###
# Ensure both are factors
mi_culture <- as.factor(gen_sem_con$Culture)

# Observed MI
obs_mi_cul <- mutinformation(mi_culture, gender)
obs_mi_cul
```

```
## [1] 0.01007617
```

```r
#I shuffle the Gender column many times (e.g., 1000 permutations) and recompute MI each time, to compar
set.seed(123)
n_perm <- 1000
perm_mi <- numeric(n_perm)

for (i in 1:n_perm) {
  shuffled_gender <- sample(gen_sem_con$Gender)
  perm_mi[i] <- mutinformation(
    as.factor(gen_sem_con$Culture),
    as.factor(shuffled_gender)
  )
}

mean(perm_mi >= obs_mi_cul)  # gives a p-value
```

```
## [1] 0
```

```r
# p=0
#mean(perm_mi >= obs_mi) = 0 means that in none of the 1000 permutations did random shuffling produce a

#more detailed results:
mean(perm_mi)      # average MI under permutation
```

```
## [1] 9.442509e-05
```

```r
range(perm_mi)     # min and max
```

```
## [1] 8.804306e-07 4.452407e-04
```

```r
obs_mi_cul            # observed MI
```

```
## [1] 0.01007617
```

```r
p_value <- (sum(perm_mi >= obs_mi_cul) + 1)/(length(perm_mi) + 1)
p_value
```

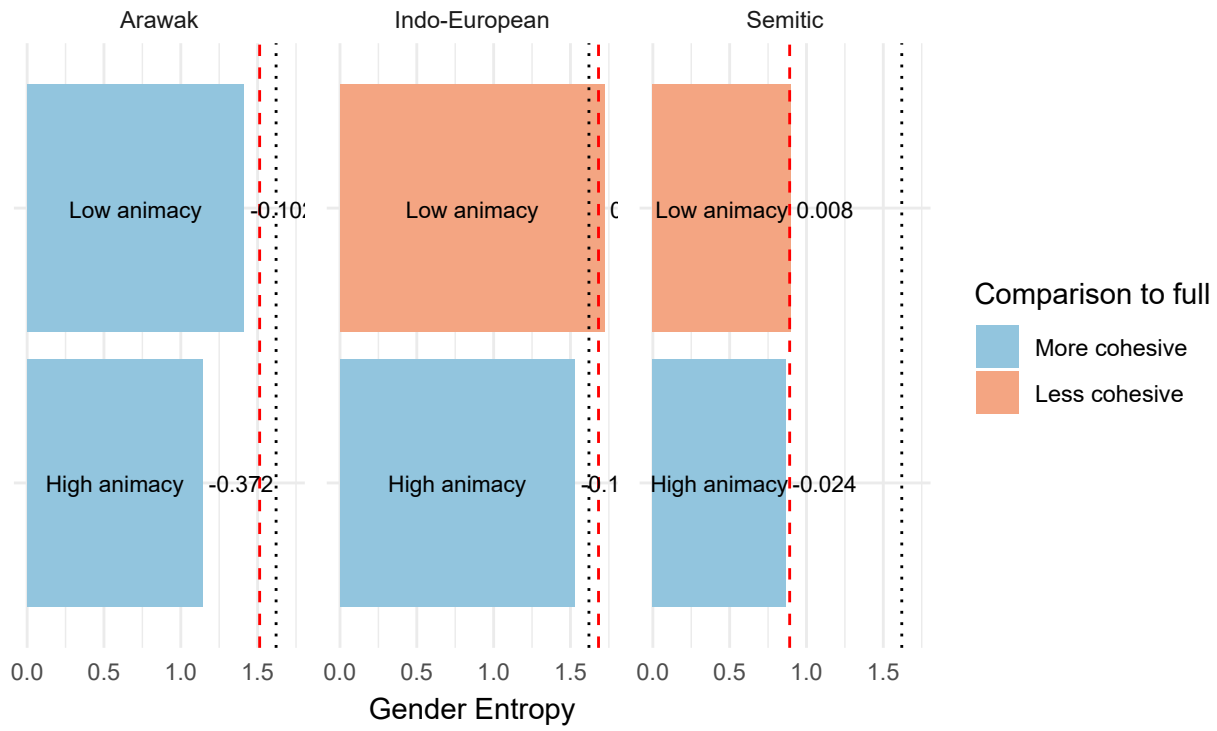
```
## [1] 0.000999001
```

2.2.2. Entropy

Nest, We calculate the Entropy:

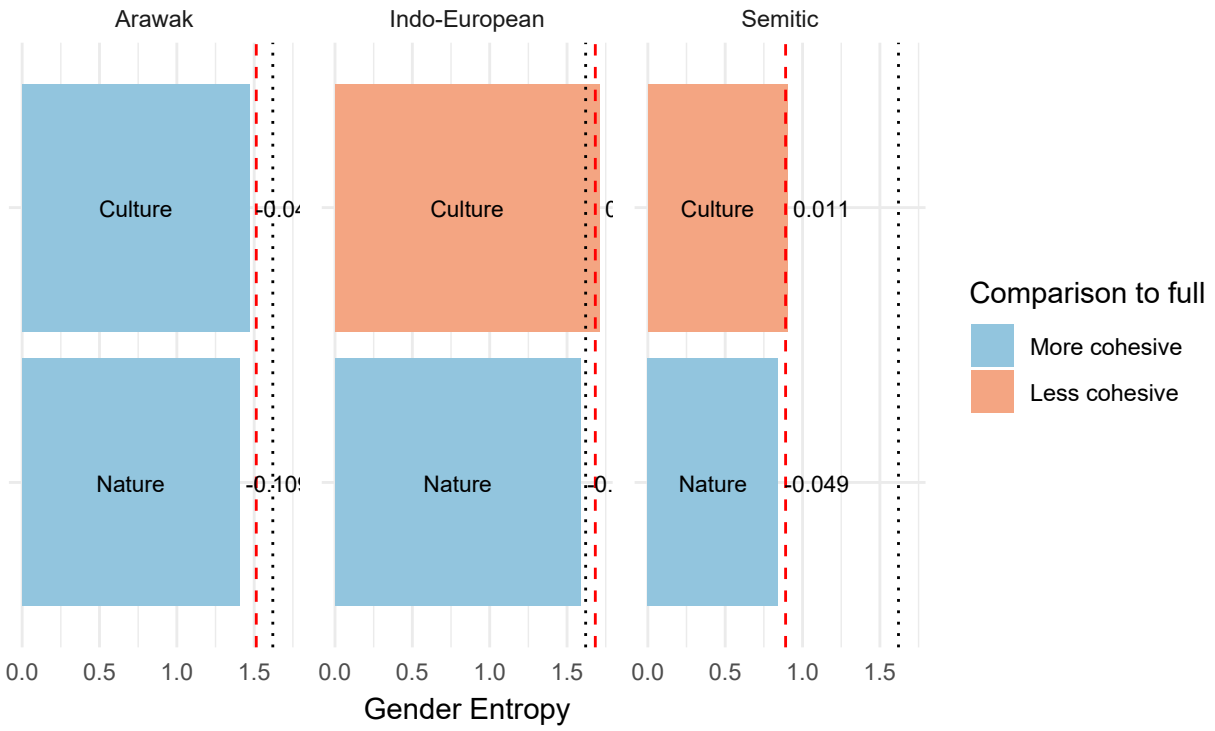
```
# Function to compute Shannon entropy
shannon_entropy <- function(x) {
  probs <- table(x) / length(x)
  -sum(probs * log2(probs))
}

# Full dataset entropy (all families)
full_dataset_entropy <- shannon_entropy(gen_sem_con$Gender)

# Compute full entropy per family and per-group entropy
animacy_entropy <- gen_sem_con %>%
  group_by(Family) %>%
  summarise(full_entropy = shannon_entropy(Gender), .groups = "drop") %>%
  left_join(
    gen_sem_con %>%
      group_by(Family, Animacy) %>%
      summarise(
        gender_entropy = shannon_entropy(Gender),
        n = n(),
        .groups = "drop"
      ),
    by = "Family"
  ) %>%
  mutate(
    diff_from_full = gender_entropy - full_entropy
  )


ggplot(animacy_entropy, aes(
  x = reorder(Animacy, gender_entropy),
  y = gender_entropy,
  fill = diff_from_full > 0
)) +
  geom_col() +
  # Put Animacy names inside the bars
  geom_text(aes(label = Animacy),
            position = position_stack(vjust = 0.5),  # vertically centered
            hjust = 0.5,                             # horizontally centered inside bar
            size = 3,
            color = "black") +                        # white text for contrast
  # Family-specific entropy line
  geom_hline(aes(yintercept = full_entropy), color = "red", linetype = "dashed") +
  # Full-dataset entropy line
  geom_hline(yintercept = full_dataset_entropy, color = "black", linetype = "dotted") +
  coord_flip() +
  scale_fill_manual(
    values = c("TRUE" = "#f4a582", "FALSE" = "#92c5de"),
    labels = c("More cohesive","Less cohesive")
  ) +
  geom_text(aes(label = round(diff_from_full,3)),
```

```
            hjust = -0.1, size = 3) +  # keeps your diff labels outside the bars
  facet_wrap(~Family, scales = "free_y") +
  labs(
    title = "Gender Entropy per Animacy across Language Families",
    subtitle = paste0("Red dashed line = full-family entropy, Black dotted line = full-dataset entropy
                      round(full_dataset_entropy,3), ")"),
    x = NULL,
    y = "Gender Entropy",
    fill = "Comparison to full"
  ) +
  theme_minimal() +
  theme(
    axis.text.y = element_blank()  # hide default y-axis labels
  )
```




```
# Compute full entropy per family and per-group entropy
culture_entropy <- gen_sem_con %>%
  group_by(Family) %>%
  summarise(full_entropy = shannon_entropy(Gender), .groups = "drop") %>%
  left_join(
    gen_sem_con %>%
      group_by(Family, Culture) %>%
      summarise(
        gender_entropy = shannon_entropy(Gender),
        n = n(),
```

```
        .groups = "drop"
      ),
    by = "Family"
  ) %>%
  mutate(
    diff_from_full = gender_entropy - full_entropy
  )


ggplot(culture_entropy, aes(
  x = reorder(Culture, gender_entropy),
  y = gender_entropy,
  fill = diff_from_full > 0
)) +
  geom_col() +
  # Put Culture names inside the bars
  geom_text(aes(label = Culture),
            position = position_stack(vjust = 0.5),  # vertically centered
            hjust = 0.5,                             # horizontally centered inside bar
            size = 3,
            color = "black") +                        # white text for contrast
  # Family-specific entropy line
  geom_hline(aes(yintercept = full_entropy), color = "red", linetype = "dashed") +
  # Full-dataset entropy line
  geom_hline(yintercept = full_dataset_entropy, color = "black", linetype = "dotted") +
  coord_flip() +
  scale_fill_manual(
    values = c("TRUE" = "#f4a582", "FALSE" = "#92c5de"),
    labels = c("More cohesive","Less cohesive")
  ) +
  geom_text(aes(label = round(diff_from_full,3)),
            hjust = -0.1, size = 3) +  # keeps your diff labels outside the bars
  facet_wrap(~Family, scales = "free_y") +
  labs(
    title = "Gender Entropy per Culture-Nature across Language Families",
    subtitle = paste0("Red dashed line = full-family entropy, Black dotted line = full-dataset entropy
                      round(full_dataset_entropy,3), ")"),
    x = NULL,
    y = "Gender Entropy",
    fill = "Comparison to full"
  ) +
  theme_minimal() +
  theme(
    axis.text.y = element_blank()  # hide default y-axis labels
  )
```

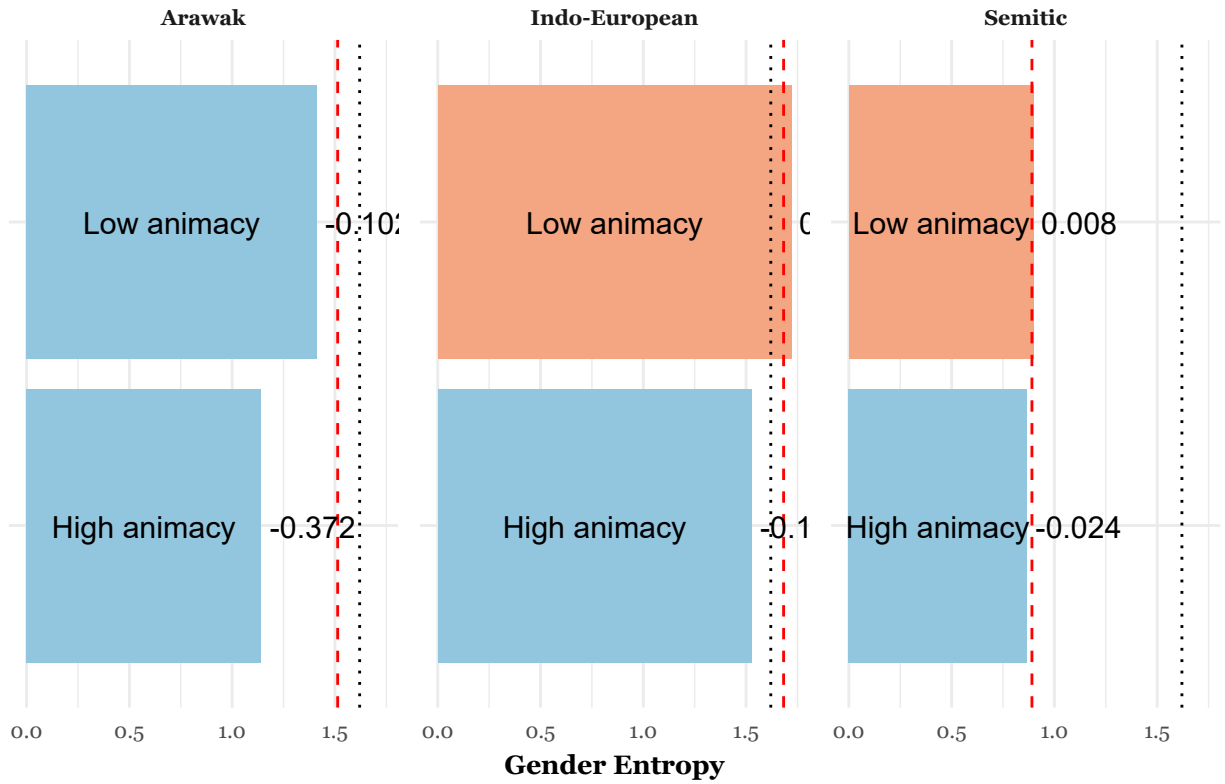


Per family

```
##Animacy##
# Function to compute Shannon entropy
shannon_entropy <- function(x) {
  probs <- table(x) / length(x)
  -sum(probs * log2(probs))
}

# Full dataset entropy (all families)
full_dataset_entropy <- shannon_entropy(gen_sem_con$Gender)

# Compute full entropy per family and per-group entropy
all_entropy <- gen_sem_con %>%
  group_by(Family) %>%
  summarise(full_entropy = shannon_entropy(Gender), .groups = "drop") %>%
  left_join(
    gen_sem_con %>%
      group_by(Family, Animacy) %>%
      summarise(
        gender_entropy = shannon_entropy(Gender),
        n = n(),
        .groups = "drop"
      ),
    by = "Family"
  ) %>%
  mutate(
```

```
    diff_from_full = gender_entropy - full_entropy
  )


ggplot(all_entropy, aes(
  x = reorder(Animacy, gender_entropy),
  y = gender_entropy,
  fill = diff_from_full > 0
)) +
  geom_col() +
  # Put Animacy names inside the bars
  geom_text(aes(label = Animacy),
            position = position_stack(vjust = 0.5),  # vertically centered
            hjust = 0.5,                             # horizontally centered inside bar
            size = 4,
            color = "black") +                        # black text for contrast
  # Family-specific entropy line
  geom_hline(aes(yintercept = full_entropy), color = "red", linetype = "dashed") +
  # Full-dataset entropy line
  geom_hline(yintercept = full_dataset_entropy, color = "black", linetype = "dotted") +
  coord_flip() +
  scale_fill_manual(
    values = c("TRUE" = "#f4a582", "FALSE" = "#92c5de"),
    labels = c("More cohesive","Less cohesive")
  ) +
  geom_text(aes(label = round(diff_from_full,3)),
            hjust = -0.1, size = 4) +  # keeps your diff labels outside the bars
  facet_wrap(~Family, scales = "free_y") +
  labs(
    title = "Gender Entropy per Animacy",
    #subtitle = paste0("Red dashed line = full-family entropy, Black dotted line = full-dataset entropy
    #                  round(full_dataset_entropy,3), ")"),
    x = NULL,
    y = "Gender Entropy",
    fill = "Comparison to full"
  ) +
  theme_minimal() +
  theme(
    legend.position = "none",
    axis.text.y = element_blank(),  # hide default y-axis labels
    plot.title = element_text(size = 10, family = "Georgia"),
    axis.title.x = element_text(size = 10, face = "bold", family = "Georgia"),
    axis.text.x = element_text(size = 8, family = "Georgia"),
    strip.text = element_text(size = 8, face = "bold", family = "Georgia")
  )
```

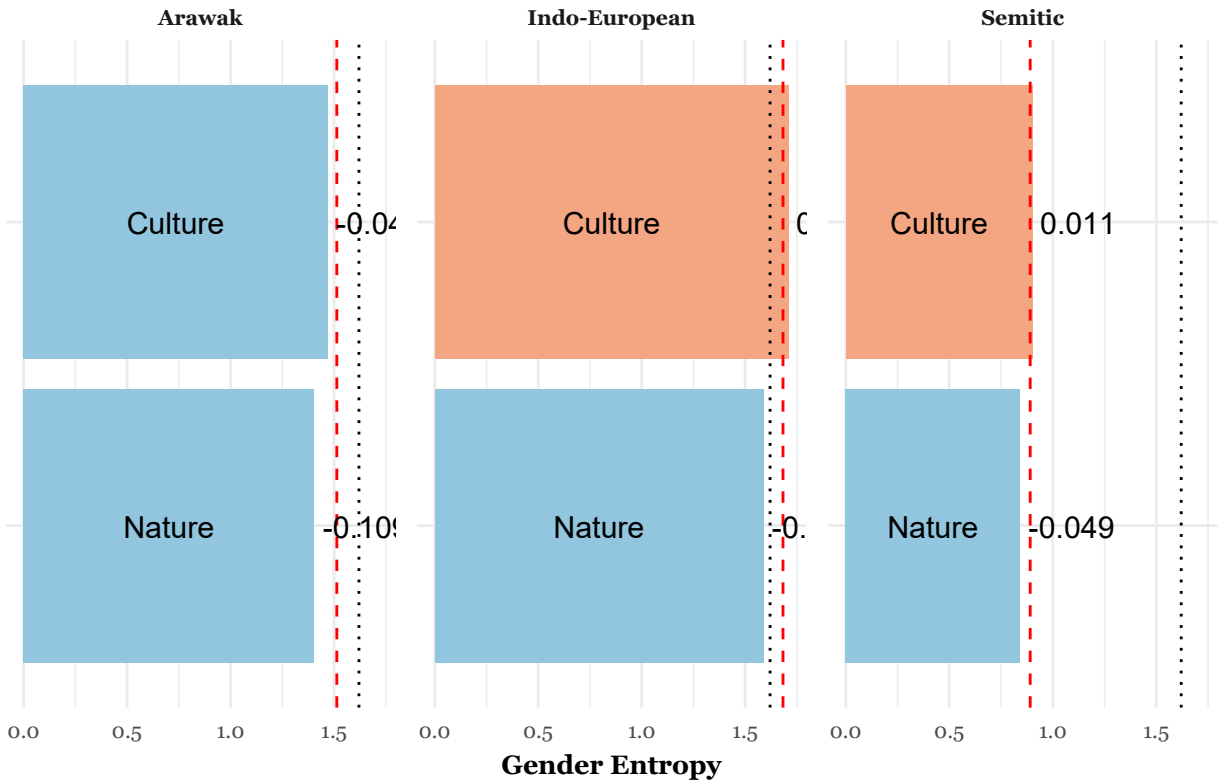
Gender Entropy per Animacy




```
ggsave("plots/Entropy Animacy per family.png", width = 8, height = 4, dpi = 300)



###Culture###
# Compute full entropy per family and per-group entropy
all_entropy <- gen_sem_con %>%
  group_by(Family) %>%
  summarise(full_entropy = shannon_entropy(Gender), .groups = "drop") %>%
  left_join(
    gen_sem_con %>%
      group_by(Family, Culture) %>%
      summarise(
        gender_entropy = shannon_entropy(Gender),
        n = n(),
        .groups = "drop"
      ),
    by = "Family"
  ) %>%
  mutate(
    diff_from_full = gender_entropy - full_entropy
  )


ggplot(all_entropy, aes(
  x = reorder(Culture, gender_entropy),
```

```
  y = gender_entropy,
  fill = diff_from_full > 0
)) +
  geom_col() +
  # Put Culture names inside the bars
  geom_text(aes(label = Culture),
            position = position_stack(vjust = 0.5),  # vertically centered
            hjust = 0.5,                             # horizontally centered inside bar
            size = 4,
            color = "black") +                        # black text for contrast
  # Family-specific entropy line
  geom_hline(aes(yintercept = full_entropy), color = "red", linetype = "dashed") +
  # Full-dataset entropy line
  geom_hline(yintercept = full_dataset_entropy, color = "black", linetype = "dotted") +
  coord_flip() +
  scale_fill_manual(
    values = c("TRUE" = "#f4a582", "FALSE" = "#92c5de"),
    labels = c("More cohesive","Less cohesive")
  ) +
  geom_text(aes(label = round(diff_from_full,3)),
            hjust = -0.1, size = 4) +  # keeps your diff labels outside the bars
  facet_wrap(~Family, scales = "free_y") +
  labs(
    title = "Gender Entropy per Naturalness",
    #subtitle = paste0("Red dashed line = full-family entropy, Black dotted line = full-dataset entropy
    #                  round(full_dataset_entropy,3), ")"),
    x = NULL,
    y = "Gender Entropy",
    fill = "Comparison to full"
  ) +
  theme_minimal() +
  theme(
    legend.position = "none",
    axis.text.y = element_blank(),  # hide default y-axis labels
    plot.title = element_text(size = 10, family = "Georgia"),
    axis.title.x = element_text(size = 10, face = "bold", family = "Georgia"),
    axis.text.x = element_text(size = 8, family = "Georgia"),
    strip.text = element_text(size = 8, face = "bold", family = "Georgia")
  )
```

Gender Entropy per Naturalness




```
ggsave("plots/Entropy Culture per family.png", width = 8, height = 4, dpi = 300)
```

2.2.3. Chi-Square

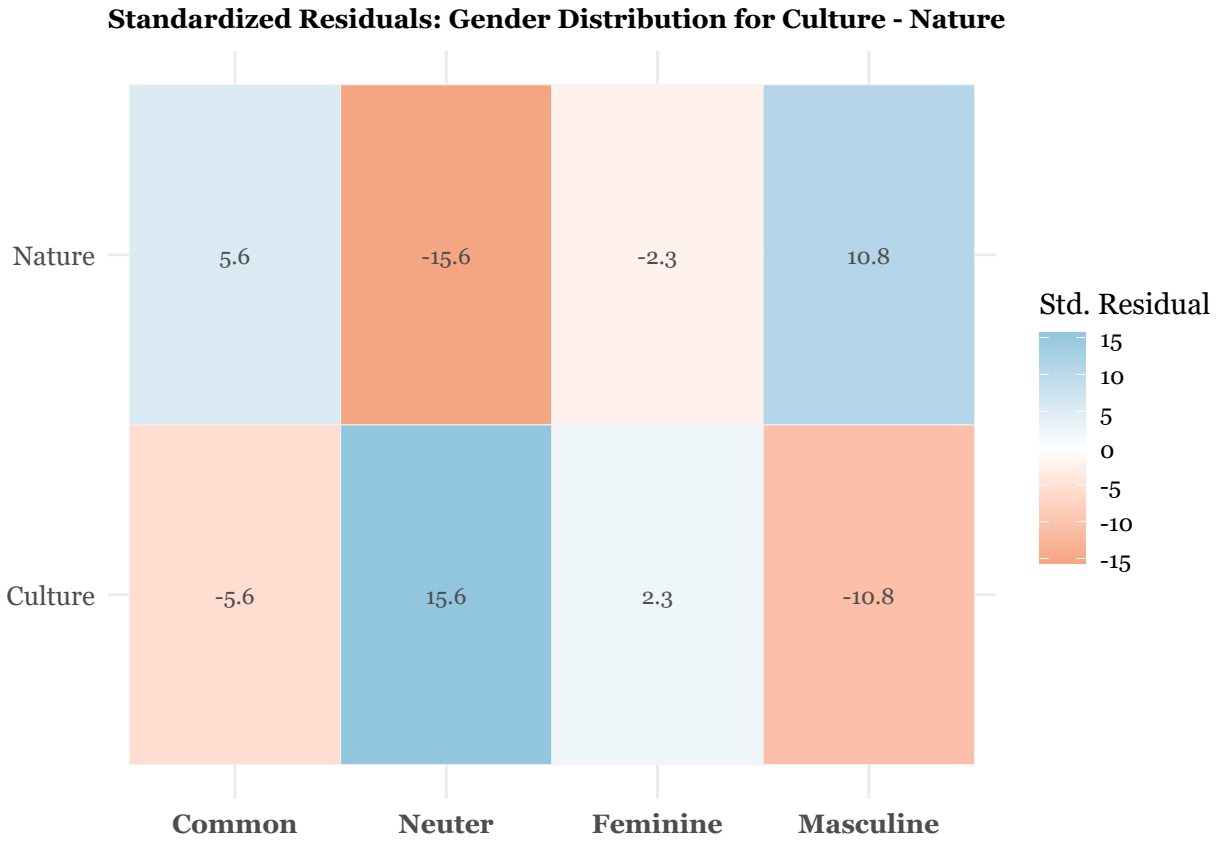
Lastly, we calculate the chi-squared residuals For Animacy and Culture. #### 2.2.3.1. Culture:

```
cul_nat <- gender_assignment %>%
  mutate(Gender = recode(Gender,
                         "C" = "Common",
                         "F" = "Feminine",
                         "M" = "Masculine",
                         "N" = "Neuter")) %>%
  mutate(Gender = factor(Gender, levels = rev(c("Masculine", "Feminine", "Neuter", "Common")))) %>%
  count(Culture, Gender) %>%
  group_by(Culture) %>%
  mutate(Percentage = n / sum(n)) %>%
  ungroup()

contingency_table_cul <- cul_nat %>%
  select(Culture, Gender, n) %>%
  pivot_wider(names_from = Gender, values_from = n, values_fill = 0) %>%
  column_to_rownames("Culture") %>%
  as.matrix()

# Run chi-square test
```

```r
chisq_result_cul <- chisq.test(contingency_table_cul)
print(chisq_result_cul)
```

```
##
##  Pearson's Chi-squared test
##
## data:  contingency_table_cul
## X-squared = 303.66, df = 3, p-value < 2.2e-16
```

```r
# Extract standardized residuals
residuals_cul <- as.data.frame(as.table(chisq_result_cul$stdres))
colnames(residuals_cul) <- c("Culture", "Gender", "Residual")

# Plot standardized residuals with numbers
ggplot(residuals_cul, aes(x = Gender, y = Culture, fill = Residual)) +
  geom_tile(color = "grey95", size = 0.1) +
  geom_text(
    aes(label = round(Residual, 1)),
    size = 3, color = "gray30", family = "Georgia"
  ) +
  scale_fill_gradient2(
    low = "#f4a582", mid = "white", high = "#92c5de", midpoint = 0,
    name = "Std. Residual"
  ) +
  labs(
    title = "Standardized Residuals: Gender Distribution for Culture - Nature",
    x = NULL, y = NULL
  ) +
  theme_minimal(base_family = "Georgia") +
  theme(
    axis.text.x = element_text(angle = 0, hjust = 0.5, size = 10, face = "bold"),
    axis.text.y = element_text(size = 10),
    plot.title = element_text(size = 10, face = "bold"),
    legend.position = "right"
  )
```




```
ggsave("plots/chi-sq residuals Culture Nature.png", width = 6, height = 3, dpi = 300)
```

Per family analysis:

```
### Arawak ###
cul_nat_ar <- Arawak %>%
  mutate(Gender = recode(Gender,
                         "C" = "Common",
                         "F" = "Feminine",
                         "M" = "Masculine",
                         "N" = "Neuter")) %>%
  mutate(Gender = factor(Gender, levels = rev(c("Masculine", "Feminine", "Neuter", "Common")))) %>%
  count(Culture, Gender) %>%
  group_by(Culture) %>%
  mutate(Percentage = n / sum(n)) %>%
  ungroup()

contingency_table_cul_ar <- cul_nat_ar %>%
  select(Culture, Gender, n) %>%
  pivot_wider(names_from = Gender, values_from = n, values_fill = 0) %>%
  column_to_rownames("Culture") %>%
  as.matrix()

# Run chi-square test
chisq_result_cul_ar <- chisq.test(contingency_table_cul_ar)
print(chisq_result_cul_ar)
```

```
##
##  Pearson's Chi-squared test
##
## data:  contingency_table_cul_ar
## X-squared = 271.64, df = 3, p-value < 2.2e-16
```

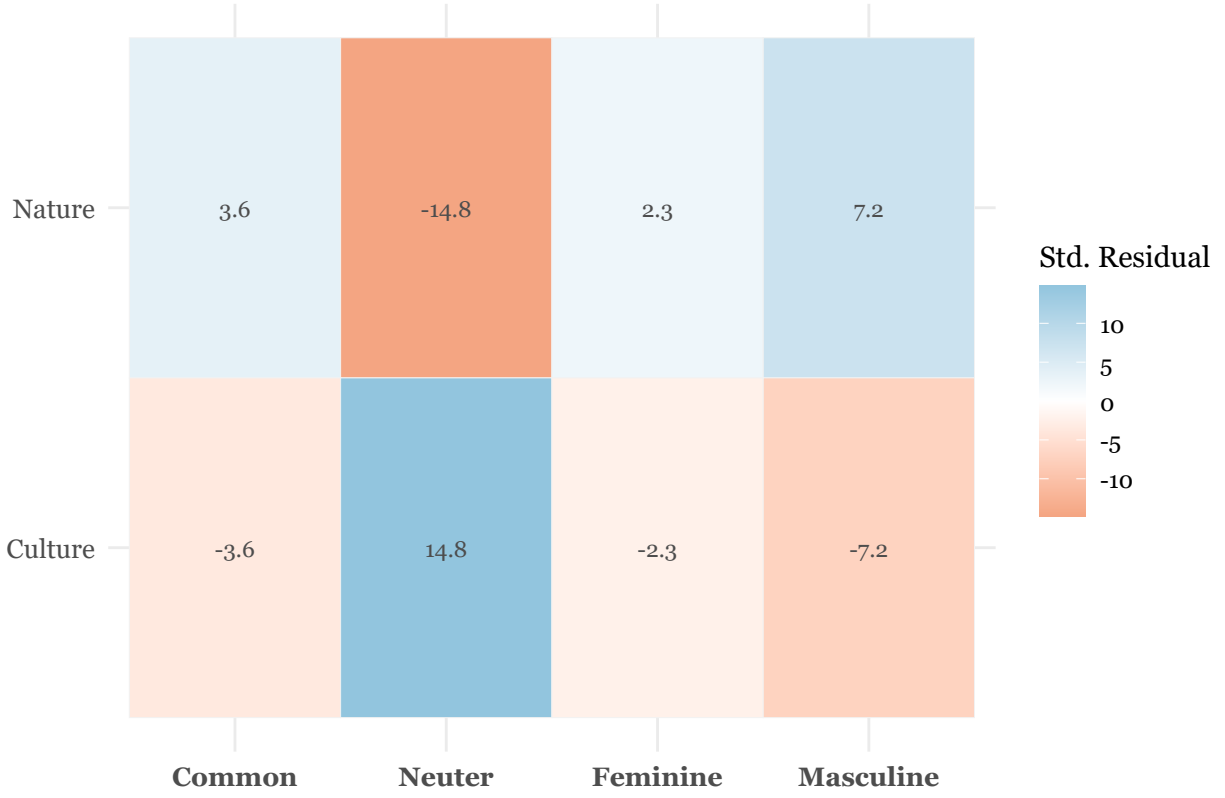
```r
# Extract standardized residuals
residuals_cul_ar <- as.data.frame(as.table(chisq_result_cul_ar$stdres))
colnames(residuals_cul_ar) <- c("Culture", "Gender", "Residual")

# Plot standardized residuals with numbers
ggplot(residuals_cul_ar, aes(x = Gender, y = Culture, fill = Residual)) +
  geom_tile(color = "grey95", size = 0.1) +
  geom_text(
    aes(label = round(Residual, 1)),
    size = 3, color = "gray30", family = "Georgia"
  ) +
  scale_fill_gradient2(
    low = "#f4a582", mid = "white", high = "#92c5de", midpoint = 0,
    name = "Std. Residual"
  ) +
  labs(
    title = "Arawak Standardized Residuals: Gender Distribution for Culture - Nature",
    x = NULL, y = NULL
  ) +
  theme_minimal(base_family = "Georgia") +
  theme(
    axis.text.x = element_text(angle = 0, hjust = 0.5, size = 10, face = "bold"),
    axis.text.y = element_text(size = 10),
    plot.title = element_text(size = 10, face = "bold"),
    legend.position = "right"
  )
```


```
ggsave("plots/Arawak chi-sq residuals Culture Nature.png", width = 6, height = 3, dpi = 300)

### Indo-European ###
cul_nat_ie <- IE %>%
  mutate(Gender = recode(Gender,
                         "C" = "Common",
                         "F" = "Feminine",
                         "M" = "Masculine",
                         "N" = "Neuter")) %>%
  mutate(Gender = factor(Gender, levels = rev(c("Masculine", "Feminine", "Neuter", "Common")))) %>%
  count(Culture, Gender) %>%
  group_by(Culture) %>%
  mutate(Percentage = n / sum(n)) %>%
  ungroup()

contingency_table_cul_ie <- cul_nat_ie %>%
  select(Culture, Gender, n) %>%
  pivot_wider(names_from = Gender, values_from = n, values_fill = 0) %>%
  column_to_rownames("Culture") %>%
  as.matrix()

# Run chi-square test
chisq_result_cul_ie <- chisq.test(contingency_table_cul_ie)
print(chisq_result_cul_ie)
```

```
##
```

```
##  Pearson's Chi-squared test
##
## data:  contingency_table_cul_ie
## X-squared = 226.38, df = 3, p-value < 2.2e-16
```

```r
# Extract standardized residuals
residuals_cul_ie <- as.data.frame(as.table(chisq_result_cul_ie$stdres))
colnames(residuals_cul_ie) <- c("Culture", "Gender", "Residual")

# Plot standardized residuals with numbers
ggplot(residuals_cul_ie, aes(x = Gender, y = Culture, fill = Residual)) +
  geom_tile(color = "grey95", size = 0.1) +
  geom_text(
    aes(label = round(Residual, 1)),
    size = 3, color = "gray30", family = "Georgia"
  ) +
  scale_fill_gradient2(
    low = "#f4a582", mid = "white", high = "#92c5de", midpoint = 0,
    name = "Std. Residual"
  ) +
  labs(
    title = "Indo-European Standardized Residuals: Gender Distribution for Culture - Nature",
    x = NULL, y = NULL
  ) +
  theme_minimal(base_family = "Georgia") +
  theme(
    axis.text.x = element_text(angle = 0, hjust = 0.5, size = 10, face = "bold"),
    axis.text.y = element_text(size = 10),
    plot.title = element_text(size = 10, face = "bold"),
    legend.position = "right"
  )
```

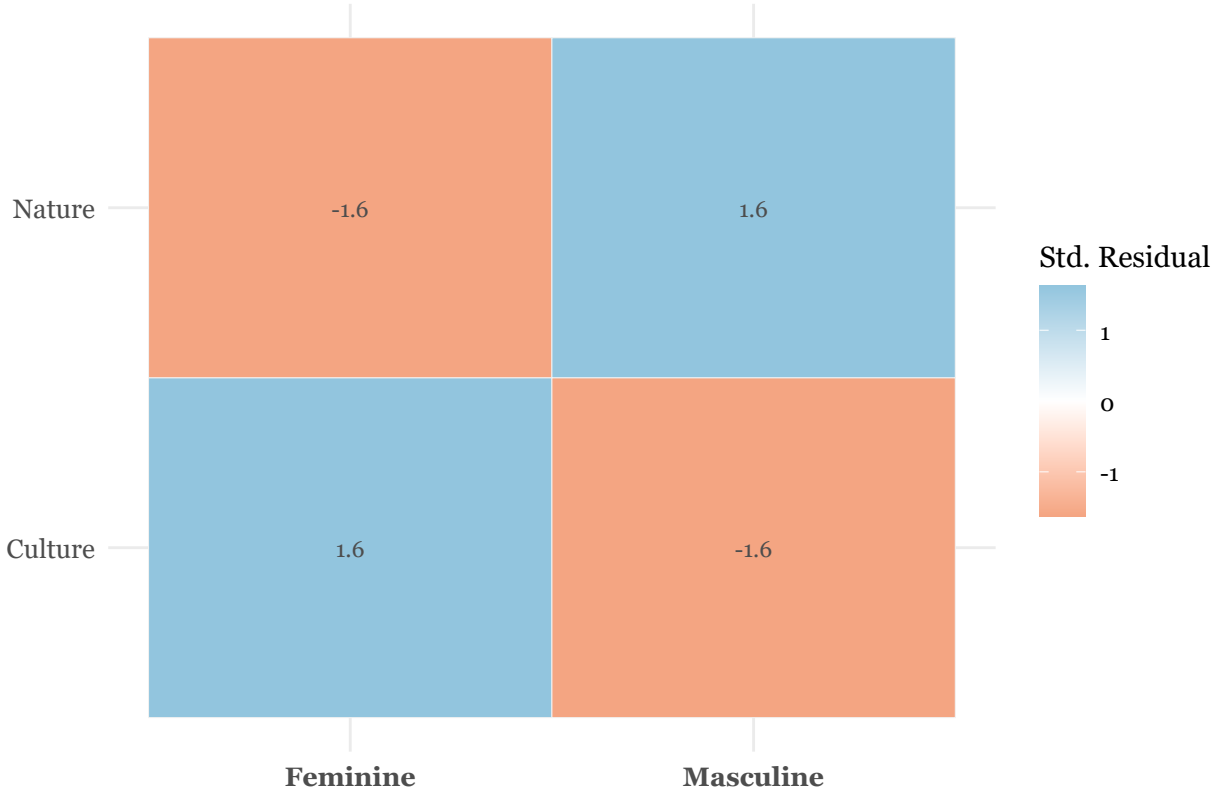



```r
ggsave("plots/IE chi-sq residuals Culture Nature.png", width = 6, height = 3, dpi = 300)

### Semitic ###
cul_nat_sem <- Semitic %>%
  mutate(Gender = recode(Gender,
                         "C" = "Common",
                         "F" = "Feminine",
                         "M" = "Masculine",
                         "N" = "Neuter")) %>%
  mutate(Gender = factor(Gender, levels = rev(c("Masculine", "Feminine", "Neuter", "Common")))) %>%
  count(Culture, Gender) %>%
  group_by(Culture) %>%
  mutate(Percentage = n / sum(n)) %>%
  ungroup()

contingency_table_cul_sem <- cul_nat_sem %>%
  select(Culture, Gender, n) %>%
  pivot_wider(names_from = Gender, values_from = n, values_fill = 0) %>%
  column_to_rownames("Culture") %>%
  as.matrix()

# Run chi-square test
chisq_result_cul_sem <- chisq.test(contingency_table_cul_sem)
print(chisq_result_cul_sem)
```

```
##
```

```
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  contingency_table_cul_sem
## X-squared = 2.4095, df = 1, p-value = 0.1206
```

```r
# Extract standardized residuals
residuals_cul_sem <- as.data.frame(as.table(chisq_result_cul_sem$stdres))
colnames(residuals_cul_sem) <- c("Culture", "Gender", "Residual")

# Plot standardized residuals with numbers
ggplot(residuals_cul_sem, aes(x = Gender, y = Culture, fill = Residual)) +
  geom_tile(color = "grey95", size = 0.1) +
  geom_text(
    aes(label = round(Residual, 1)),
    size = 3, color = "gray30", family = "Georgia"
  ) +
  scale_fill_gradient2(
    low = "#f4a582", mid = "white", high = "#92c5de", midpoint = 0,
    name = "Std. Residual"
  ) +
  labs(
    title = "Semitic Standardized Residuals: Gender Distribution for Culture - Nature",
    x = NULL, y = NULL
  ) +
  theme_minimal(base_family = "Georgia") +
  theme(
    axis.text.x = element_text(angle = 0, hjust = 0.5, size = 10, face = "bold"),
    axis.text.y = element_text(size = 10),
    plot.title = element_text(size = 10, face = "bold"),
    legend.position = "right"
  )
```



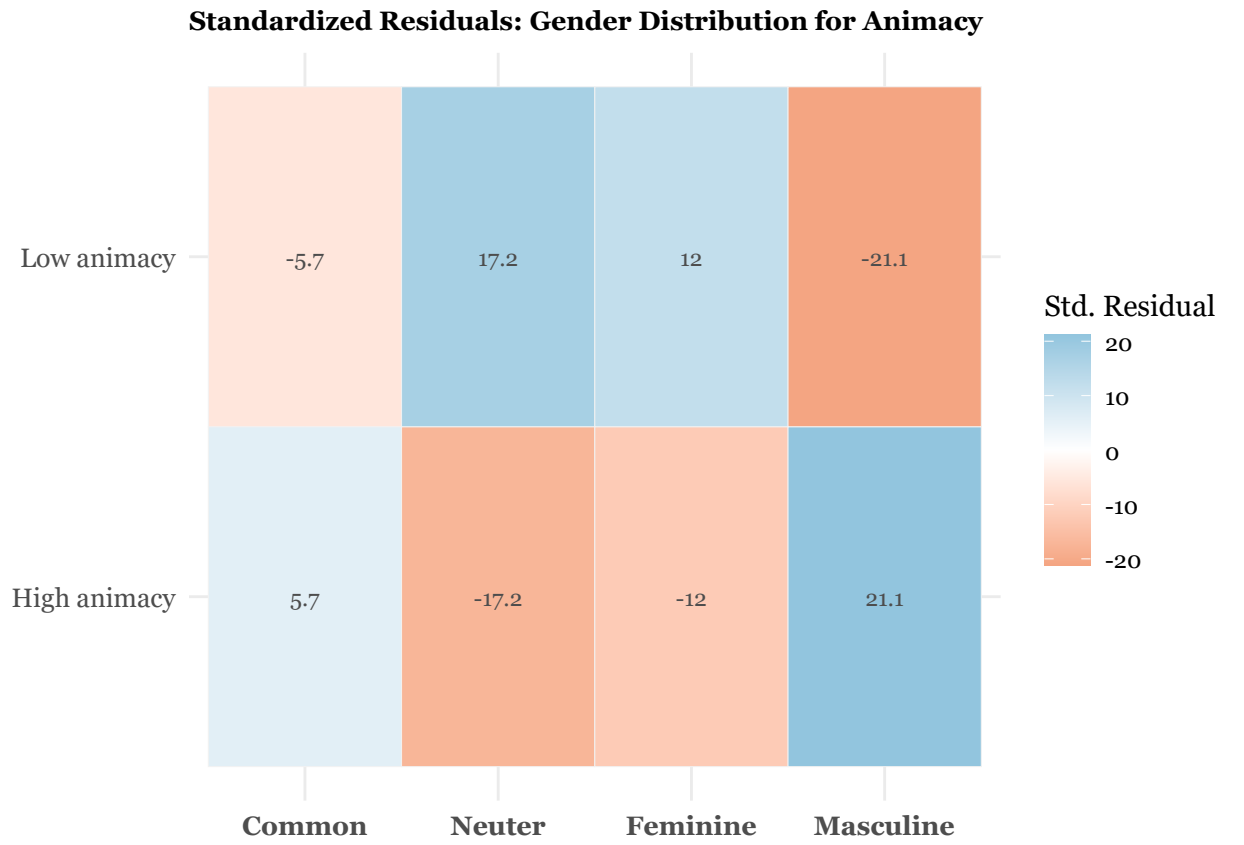

```
ggsave("plots/Semitic chi-sq residuals Culture Nature.png", width = 6, height = 3, dpi = 300)
```

```
anim <- gender_assignment %>%
  mutate(Gender = recode(Gender,
                         "C" = "Common",
                         "F" = "Feminine",
                         "M" = "Masculine",
                         "N" = "Neuter")) %>%
  mutate(Gender = factor(Gender, levels = rev(c("Masculine", "Feminine", "Neuter", "Common")))) %>%
  count(Animacy, Gender) %>%
  group_by(Animacy) %>%
  mutate(Percentage = n / sum(n)) %>%
  ungroup()

contingency_table_anim <- anim %>%
  select(Animacy, Gender, n) %>%
  pivot_wider(names_from = Gender, values_from = n, values_fill = 0) %>%
  column_to_rownames("Animacy") %>%
  as.matrix()

# Run chi-square test
chisq_result_anim <- chisq.test(contingency_table_anim)
print(chisq_result_anim)
```

2.2.3.2. Animacy:

```
##
##  Pearson's Chi-squared test
##
## data:  contingency_table_anim
## X-squared = 612.5, df = 3, p-value < 2.2e-16
```

```r
# Extract standardized residuals
residuals_anim <- as.data.frame(as.table(chisq_result_anim$stdres))
colnames(residuals_anim) <- c("Animacy", "Gender", "Residual")

# Plot standardized residuals with numbers
ggplot(residuals_anim, aes(x = Gender, y = Animacy, fill = Residual)) +
  geom_tile(color = "grey95", size = 0.1) +
  geom_text(
    aes(label = round(Residual, 1)),
    size = 3, color = "gray30", family = "Georgia"
  ) +
  scale_fill_gradient2(
    low = "#f4a582", mid = "white", high = "#92c5de", midpoint = 0,
    name = "Std. Residual"
  ) +
  labs(
    title = "Standardized Residuals: Gender Distribution for Animacy",
    x = NULL, y = NULL
  ) +
  theme_minimal(base_family = "Georgia") +
  theme(
    axis.text.x = element_text(angle = 0, hjust = 0.5, size = 10, face = "bold"),
    axis.text.y = element_text(size = 10),
    plot.title = element_text(size = 10, face = "bold"),
    legend.position = "right"
  )
```



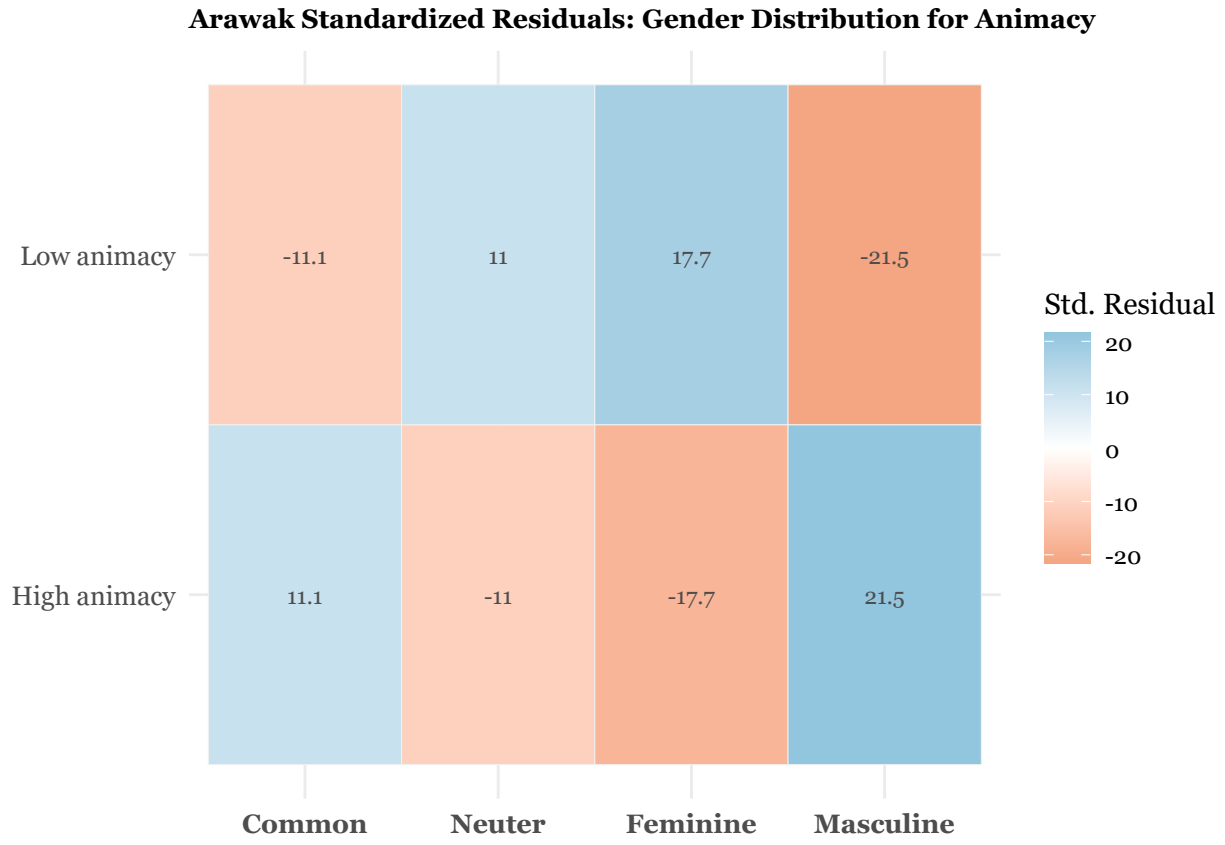

```
ggsave("plots/chi-sq residuals Animacy.png", width = 6, height = 3, dpi = 300)
```

Per family analysis:

```
### Arawak ###
anim_ar <- Arawak %>%
  mutate(Gender = recode(Gender,
                         "C" = "Common",
                         "F" = "Feminine",
                         "M" = "Masculine",
                         "N" = "Neuter")) %>%
  mutate(Gender = factor(Gender, levels = rev(c("Masculine", "Feminine", "Neuter", "Common")))) %>%
  count(Animacy, Gender) %>%
  group_by(Animacy) %>%
  mutate(Percentage = n / sum(n)) %>%
  ungroup()

contingency_table_anim_ar <- anim_ar %>%
  select(Animacy, Gender, n) %>%
  pivot_wider(names_from = Gender, values_from = n, values_fill = 0) %>%
  column_to_rownames("Animacy") %>%
  as.matrix()

# Run chi-square test
chisq_result_anim_ar <- chisq.test(contingency_table_anim_ar)
print(chisq_result_anim_ar)
```

```
##
##  Pearson's Chi-squared test
##
## data:  contingency_table_anim_ar
## X-squared = 668.95, df = 3, p-value < 2.2e-16
```

```r
# Extract standardized residuals
residuals_anim_ar <- as.data.frame(as.table(chisq_result_anim_ar$stdres))
colnames(residuals_anim_ar) <- c("Animacy", "Gender", "Residual")

# Plot standardized residuals with numbers
ggplot(residuals_anim_ar, aes(x = Gender, y = Animacy, fill = Residual)) +
  geom_tile(color = "grey95", size = 0.1) +
  geom_text(
    aes(label = round(Residual, 1)),
    size = 3, color = "gray30", family = "Georgia"
  ) +
  scale_fill_gradient2(
    low = "#f4a582", mid = "white", high = "#92c5de", midpoint = 0,
    name = "Std. Residual"
  ) +
  labs(
    title = "Arawak Standardized Residuals: Gender Distribution for Animacy",
    x = NULL, y = NULL
  ) +
  theme_minimal(base_family = "Georgia") +
  theme(
    axis.text.x = element_text(angle = 0, hjust = 0.5, size = 10, face = "bold"),
    axis.text.y = element_text(size = 10),
    plot.title = element_text(size = 10, face = "bold"),
    legend.position = "right"
  )
```

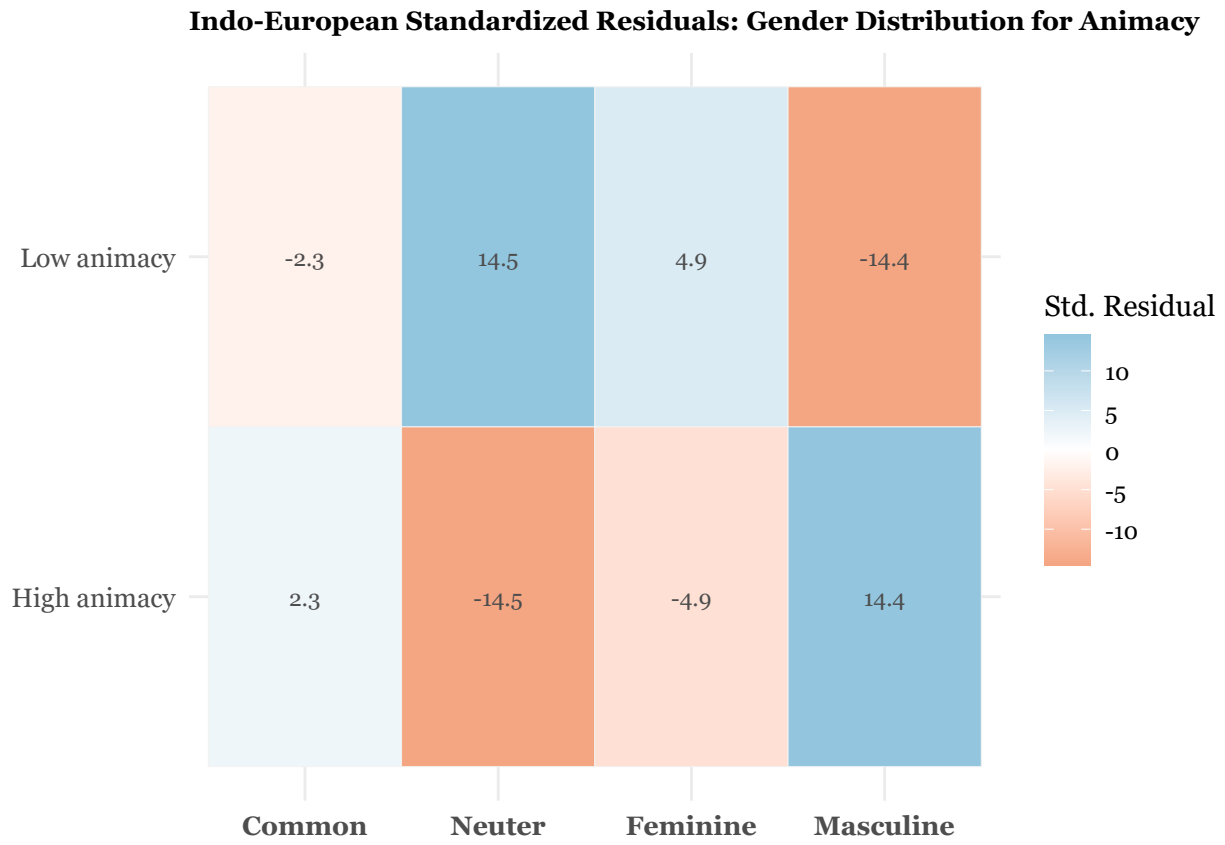



```
ggsave("plots/Arawak chi-sq residuals Animacy.png", width = 6, height = 3, dpi = 300)

### Indo-European ###
anim_ie <- IE %>%
  mutate(Gender = recode(Gender,
                         "C" = "Common",
                         "F" = "Feminine",
                         "M" = "Masculine",
                         "N" = "Neuter")) %>%
  mutate(Gender = factor(Gender, levels = rev(c("Masculine", "Feminine", "Neuter", "Common")))) %>%
  count(Animacy, Gender) %>%
  group_by(Animacy) %>%
  mutate(Percentage = n / sum(n)) %>%
  ungroup()

contingency_table_anim_ie <- anim_ie %>%
  select(Animacy, Gender, n) %>%
  pivot_wider(names_from = Gender, values_from = n, values_fill = 0) %>%
  column_to_rownames("Animacy") %>%
  as.matrix()

# Run chi-square test
chisq_result_anim_ie <- chisq.test(contingency_table_anim_ie)
print(chisq_result_anim_ie)
```

```
##
```

```
##  Pearson's Chi-squared test
##
## data:  contingency_table_anim_ie
## X-squared = 307.34, df = 3, p-value < 2.2e-16
```

```r
# Extract standardized residuals
residuals_anim_ie <- as.data.frame(as.table(chisq_result_anim_ie$stdres))
colnames(residuals_anim_ie) <- c("Animacy", "Gender", "Residual")

# Plot standardized residuals with numbers
ggplot(residuals_anim_ie, aes(x = Gender, y = Animacy, fill = Residual)) +
  geom_tile(color = "grey95", size = 0.1) +
  geom_text(
    aes(label = round(Residual, 1)),
    size = 3, color = "gray30", family = "Georgia"
  ) +
  scale_fill_gradient2(
    low = "#f4a582", mid = "white", high = "#92c5de", midpoint = 0,
    name = "Std. Residual"
  ) +
  labs(
    title = "Indo-European Standardized Residuals: Gender Distribution for Animacy",
    x = NULL, y = NULL
  ) +
  theme_minimal(base_family = "Georgia") +
  theme(
    axis.text.x = element_text(angle = 0, hjust = 0.5, size = 10, face = "bold"),
    axis.text.y = element_text(size = 10),
    plot.title = element_text(size = 10, face = "bold"),
    legend.position = "right"
  )
```




```
ggsave("plots/IE chi-sq residuals Animacy.png", width = 6, height = 3, dpi = 300)

### Semitic ###
anim_sem <- Semitic %>%
  mutate(Gender = recode(Gender,
                         "C" = "Common",
                         "F" = "Feminine",
                         "M" = "Masculine",
                         "N" = "Neuter")) %>%
  mutate(Gender = factor(Gender, levels = rev(c("Masculine", "Feminine", "Neuter", "Common")))) %>%
  count(Animacy, Gender) %>%
  group_by(Animacy) %>%
  mutate(Percentage = n / sum(n)) %>%
  ungroup()

contingency_table_anim_sem <- anim_sem %>%
  select(Animacy, Gender, n) %>%
  pivot_wider(names_from = Gender, values_from = n, values_fill = 0) %>%
  column_to_rownames("Animacy") %>%
  as.matrix()

# Run chi-square test
chisq_result_anim_sem <- chisq.test(contingency_table_anim_sem)
print(chisq_result_anim_sem)
```

```
##
```

```
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  contingency_table_anim_sem
## X-squared = 0.83513, df = 1, p-value = 0.3608
```

```r
# Extract standardized residuals
residuals_anim_sem <- as.data.frame(as.table(chisq_result_anim_sem$stdres))
colnames(residuals_anim_sem) <- c("Animacy", "Gender", "Residual")

# Plot standardized residuals with numbers
ggplot(residuals_anim_sem, aes(x = Gender, y = Animacy, fill = Residual)) +
  geom_tile(color = "grey95", size = 0.1) +
  geom_text(
    aes(label = round(Residual, 1)),
    size = 3, color = "gray30", family = "Georgia"
  ) +
  scale_fill_gradient2(
    low = "#f4a582", mid = "white", high = "#92c5de", midpoint = 0,
    name = "Std. Residual"
  ) +
  labs(
    title = "Semitic Standardized Residuals: Gender Distribution for Animacy",
    x = NULL, y = NULL
  ) +
  theme_minimal(base_family = "Georgia") +
  theme(
    axis.text.x = element_text(angle = 0, hjust = 0.5, size = 10, face = "bold"),
    axis.text.y = element_text(size = 10),
    plot.title = element_text(size = 10, face = "bold"),
    legend.position = "right"
  )
```

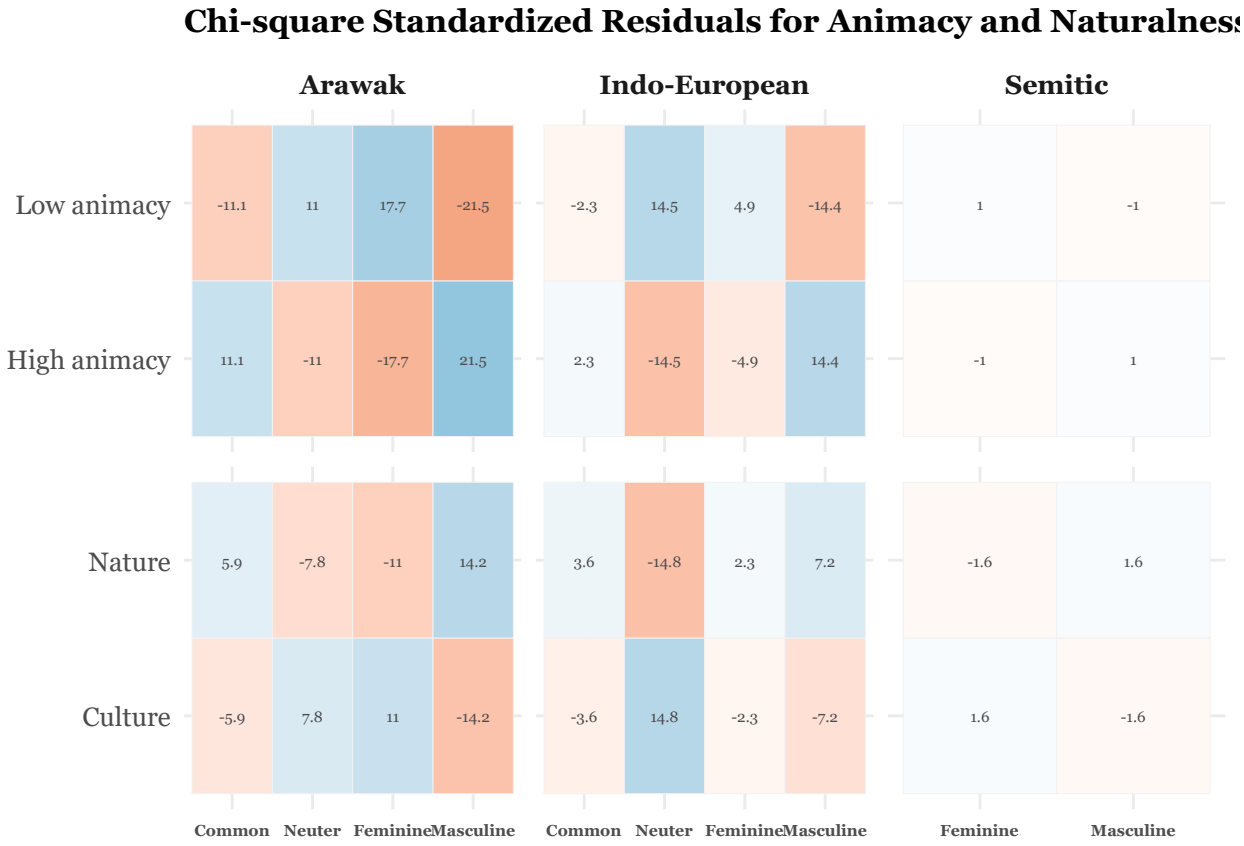

```
ggsave("plots/Semitic chi-sq residuals Animacy.png", width = 6, height = 3, dpi = 300)
```

Plot Animacy and Culture together for all 3 families

```
make_residuals_df <- function(df, family_name, var) {

  df_clean <- df %>%
    mutate(Gender = recode(Gender,
                           "C" = "Common",
                           "F" = "Feminine",
                           "M" = "Masculine",
                           "N" = "Neuter")) %>%
    mutate(Gender = factor(Gender,
                           levels = rev(c("Masculine","Feminine","Neuter","Common")))) %>%
    count(!!sym(var), Gender) %>%
    rename(Group = !!sym(var)) %>%
    group_by(Group) %>%
    mutate(Percentage = n/sum(n)) %>%
    ungroup()

  contingency <- df_clean %>%
    select(Group, Gender, n) %>%
    tidyr::pivot_wider(names_from = Gender, values_from = n, values_fill = 0) %>%
    tibble::column_to_rownames("Group") %>%
    as.matrix()
```

```
  chi <- chisq.test(contingency)

  res <- as.data.frame(as.table(chi$stdres))
  colnames(res) <- c("Group", "Gender", "Residual")

  res$Family <- family_name
  res$VariableType <- var

  return(res)
}

res_anim_ar <- make_residuals_df(Arawak, "Arawak", "Animacy")
res_anim_ie <- make_residuals_df(IE, "Indo-European", "Animacy")
res_anim_sem <- make_residuals_df(Semitic, "Semitic", "Animacy")

res_cult_ar <- make_residuals_df(Arawak, "Arawak", "Culture")
res_cult_ie <- make_residuals_df(IE, "Indo-European", "Culture")
res_cult_sem <- make_residuals_df(Semitic, "Semitic", "Culture")

all_residuals <- dplyr::bind_rows(
  res_anim_ar, res_anim_ie, res_anim_sem,
  res_cult_ar, res_cult_ie, res_cult_sem
)

ggplot(all_residuals, aes(x = Gender, y = Group, fill = Residual)) +
  geom_tile(color = "grey95", size = 0.1) +
  geom_text(
    aes(label = round(Residual, 1)),
    size = 2, color = "gray30", family = "Georgia"
  ) +
  scale_fill_gradient2(
    low = "#f4a582", mid = "white", high = "#92c5de",
    midpoint = 0, name = "Std. Residual"
  ) +
  facet_grid(VariableType ~ Family, scales = "free") +
  labs(
    title = "Chi-square Standardized Residuals for Animacy and Naturalness",
    x = NULL, y = NULL
  ) +
  theme_minimal(base_family = "Georgia") +
  theme(
    axis.text.x = element_text(size = 6, face = "bold"),
    axis.text.y = element_text(size = 10),

    # Remove legend
    legend.position = "none",

    # Remove row facet labels (Animacy / Culture)
    strip.text.y = element_blank(),

    # Keep Family labels
    strip.text.x = element_text(size = 10, face = "bold"),
```

```
    plot.title = element_text(size = 12, face = "bold")
  )
```




```
ggsave("plots/chi-sq residuals animacy culture per family.png", width = 10, height = 6, dpi = 300)
```

3. Sensitivity analysis

Here we want to see whether the type of gender system plays a role in the semantic mapping of gender assignment. We create a subset of the Arawak and Indo-European data that only includes languages with Masculine-Feminine gender systems, similar to the Semitic family. Then we run the same analyses on this data set. We see that the patterns that we had with the four genders are kept when we test the smaller two-gender dataset.

```
gender_assignment_mf <- gender_assignment %>%
  group_by(Language) %>%
  filter(!any(Gender %in% c("C", "N"))) %>%
  ungroup()

#or:
#gender_assignment_mf <- read_excel("data/gender_assignment_mf_systems.xlsx")
```

create a dataframe per language family (Semitic of course stays the same)

```r
IE_mf <- gender_assignment_mf %>% filter(Family == "Indo-European")
Arawak_mf <- gender_assignment_mf %>% filter(Family == "Arawak")
```

```r
#### Arawak #####
arawak_rep_mf <- Arawak_mf %>%
  mutate(Gender = recode(Gender,
                         "C" = "Common",
                         "F" = "Feminine",
                         "M" = "Masculine",
                         "N" = "Neuter")) %>%
  mutate(Gender = factor(Gender, levels = rev(c("Masculine", "Feminine", "Neuter", "Common")))) %>%
  count(ReplaceConcept, Gender) %>%
  group_by(ReplaceConcept) %>%
  mutate(Percentage = n / sum(n)) %>%
  ungroup()

contingency_table_rep_ar_mf <- arawak_rep_mf %>%
  select(ReplaceConcept, Gender, n) %>%
  pivot_wider(names_from = Gender, values_from = n, values_fill = 0) %>%
  column_to_rownames("ReplaceConcept") %>%
  as.matrix()

# Run chi-square test
chisq_result_rep_ar_mf <- chisq.test(contingency_table_rep_ar_mf)
print(chisq_result_rep_ar_mf)
```

```
##
##  Pearson's Chi-squared test
##
## data:  contingency_table_rep_ar_mf
## X-squared = 451.25, df = 11, p-value < 2.2e-16
```

```r
# Extract standardized residuals
residuals_rep_ar_mf <- as.data.frame(as.table(chisq_result_rep_ar_mf$stdres))
colnames(residuals_rep_ar_mf) <- c("ReplaceConcept", "Gender", "Residual")

# Plot standardized residuals with numbers
ggplot(residuals_rep_ar_mf, aes(x = Gender, y = ReplaceConcept, fill = Residual)) +
  geom_tile(color = "grey95", size = 0.1) +
  geom_text(
    aes(label = round(Residual, 1)),
    size = 1.7, color = "gray30", family = "Georgia"
  ) +
  scale_fill_gradient2(
    low = "#f4a582", mid = "white", high = "#92c5de", midpoint = 0,
    name = "Std. Residual"
  ) +
  labs(
    title = "Arawak_MF Standardized Residuals: Gender Distribution by Semantic Category",
    x = NULL, y = NULL
  ) +
  theme_minimal(base_family = "Georgia") +
  theme(
```

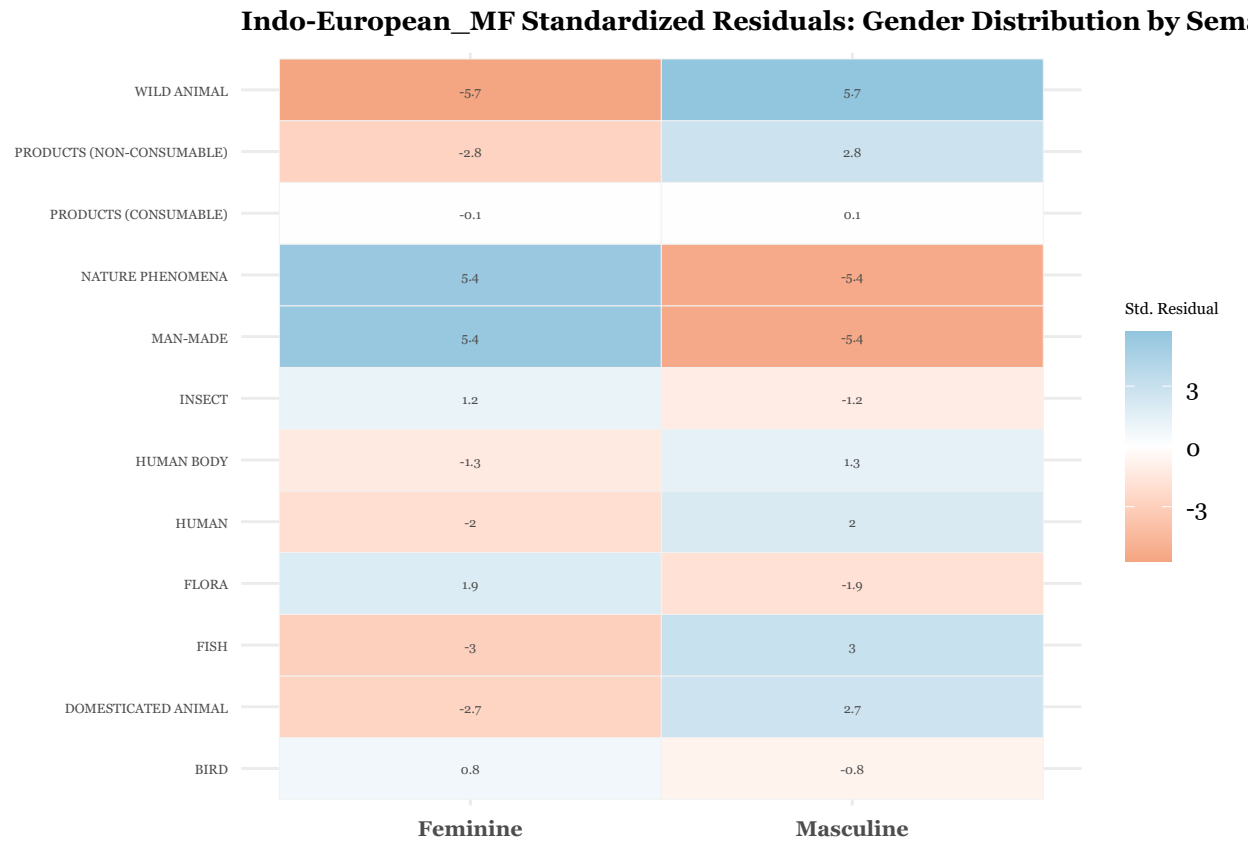
```
    axis.text.x = element_text(angle = 0, hjust = 0.5, size = 8, face = "bold"),
    axis.text.y = element_text(size = 5),
    plot.title = element_text(size = 10, face = "bold"),
    legend.position = "right",
    legend.title = element_text(size = 6)
  )
```


```
ggsave("plots/Sensitivity/Arawak_MF chi-sq residuals per ReplaceConcept.png", width = 8, height = 4, dp

#### Indo-European #####
ie_rep_mf <- IE_mf %>%
  mutate(Gender = recode(Gender,
                         "C" = "Common",
                         "F" = "Feminine",
                         "M" = "Masculine",
                         "N" = "Neuter")) %>%
  mutate(Gender = factor(Gender, levels = rev(c("Masculine", "Feminine", "Neuter", "Common")))) %>%
  count(ReplaceConcept, Gender) %>%
  group_by(ReplaceConcept) %>%
  mutate(Percentage = n / sum(n)) %>%
  ungroup()

contingency_table_rep_ie_mf <- ie_rep_mf %>%
  select(ReplaceConcept, Gender, n) %>%
  pivot_wider(names_from = Gender, values_from = n, values_fill = 0) %>%
```

```r
  column_to_rownames("ReplaceConcept") %>%
  as.matrix()

# Run chi-square test
chisq_result_rep_ie_mf <- chisq.test(contingency_table_rep_ie_mf)
print(chisq_result_rep_ie_mf)
```

```
##
##  Pearson's Chi-squared test
##
## data:  contingency_table_rep_ie_mf
## X-squared = 112.35, df = 11, p-value < 2.2e-16
```

```r
# Extract standardized residuals
residuals_rep_ie_mf <- as.data.frame(as.table(chisq_result_rep_ie_mf$stdres))
colnames(residuals_rep_ie_mf) <- c("ReplaceConcept", "Gender", "Residual")

# Plot standardized residuals with numbers
ggplot(residuals_rep_ie_mf, aes(x = Gender, y = ReplaceConcept, fill = Residual)) +
  geom_tile(color = "grey95", size = 0.1) +
  geom_text(
    aes(label = round(Residual, 1)),
    size = 1.7, color = "gray30", family = "Georgia"
  ) +
  scale_fill_gradient2(
    low = "#f4a582", mid = "white", high = "#92c5de", midpoint = 0,
    name = "Std. Residual"
  ) +
  labs(
    title = "Indo-European_MF Standardized Residuals: Gender Distribution by Semantic Category",
    x = NULL, y = NULL
  ) +
  theme_minimal(base_family = "Georgia") +
  theme(
    axis.text.x = element_text(angle = 0, hjust = 0.5, size = 8, face = "bold"),
    axis.text.y = element_text(size = 5),
    plot.title = element_text(size = 10, face = "bold"),
    legend.position = "right",
    legend.title = element_text(size = 6)
  )
```




```
ggsave("plots/Sensitivity/IE_MF chi-sq residuals per ReplaceConcept.png", width = 8, height = 4, dpi = 3
```

4. Sanity check

4.1. Subset of the dataset

We want to know whether the results are an artifact of the random noun selection that we have. Whether we had a different set of nouns we would still get the same pattern or not. The results hold for both sampling half the data from each language and when sampling a balanced sample from all families.

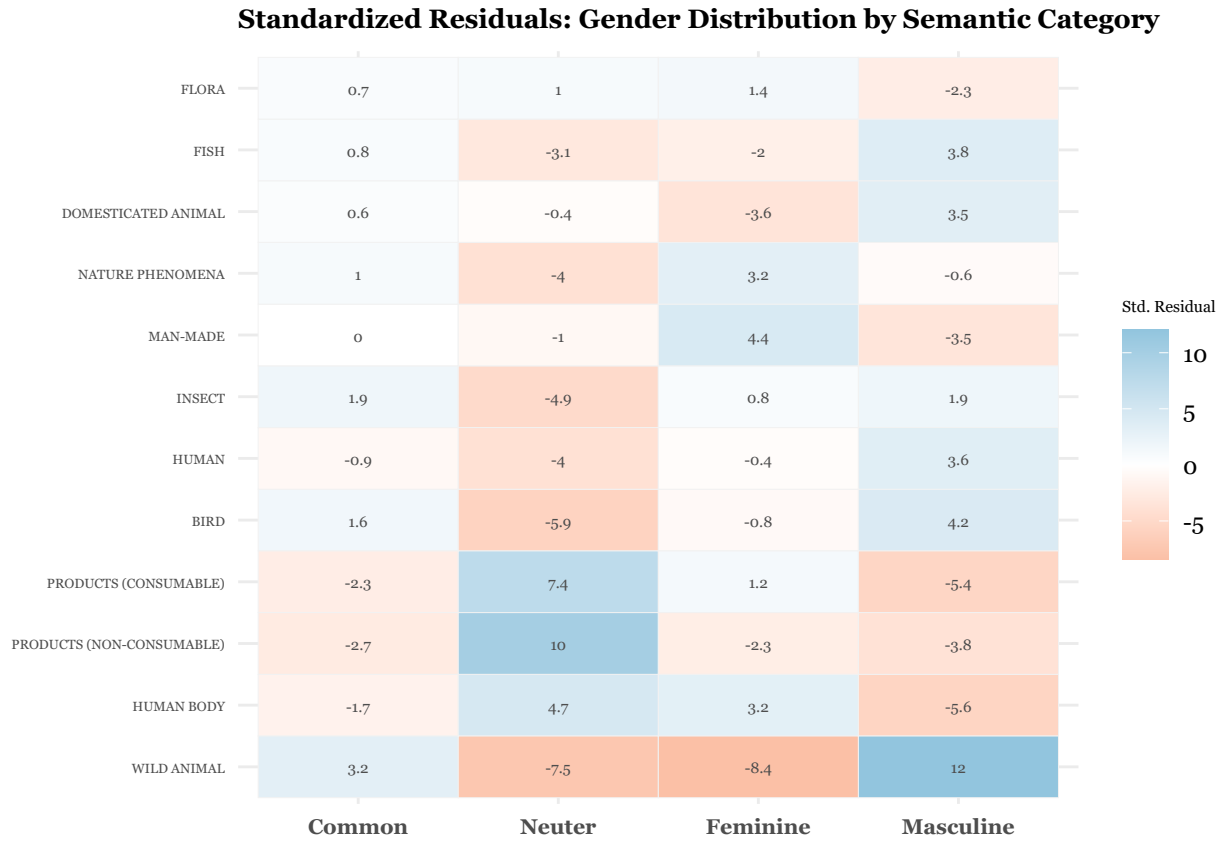
```
set.seed(123)  # for reproducibility

### Randomly sample 50% of data per family ###
sampled_data_per <- gender_assignment %>%
  group_by(Family) %>%           # group by Family
  sample_frac(0.5) %>%           # sample 50% of each group
  ungroup()

heatmap_data_rep_per <- sampled_data_per %>%
  mutate(Gender = recode(Gender,
                         "C" = "Common",
                         "F" = "Feminine",
                         "M" = "Masculine",
                         "N" = "Neuter")) %>%
```

```
  mutate(Gender = factor(Gender, levels = rev(c("Masculine", "Feminine", "Neuter", "Common")))) %>%
  count(ReplaceConcept, Gender) %>%
  group_by(ReplaceConcept) %>%
  mutate(Percentage = n / sum(n)) %>%
  ungroup()

contingency_table_rep_per <- heatmap_data_rep_per %>%
  select(ReplaceConcept, Gender, n) %>%
  pivot_wider(names_from = Gender, values_from = n, values_fill = 0) %>%
  column_to_rownames("ReplaceConcept") %>%
  as.matrix()

# Run chi-square test
chisq_result_rep_per <- chisq.test(contingency_table_rep_per)
print(chisq_result_rep_per)
```

```
##
##  Pearson's Chi-squared test
##
## data:  contingency_table_rep_per
## X-squared = 517.1, df = 33, p-value < 2.2e-16
```

```
# Extract standardized residuals
residuals_df_per <- as.data.frame(as.table(chisq_result_rep_per$stdres))
colnames(residuals_df_per) <- c("ReplaceConcept", "Gender", "Residual")

# Order by strongest deviation
concept_order_per <- residuals_df_per %>%
  group_by(ReplaceConcept) %>%
  summarize(mean_abs_res = mean(abs(Residual), na.rm = TRUE)) %>%
  arrange(desc(mean_abs_res)) %>%
  pull(ReplaceConcept)

residuals_df_per <- residuals_df_per %>%
  mutate(ReplaceConcept = factor(ReplaceConcept, levels = concept_order))

# Plot standardized residuals with numbers
ggplot(residuals_df_per, aes(x = Gender, y = ReplaceConcept, fill = Residual)) +
  geom_tile(color = "grey95", size = 0.1) +
  geom_text(
    aes(label = round(Residual, 1)),
    size = 2, color = "gray30", family = "Georgia"
  ) +
  scale_fill_gradient2(
    low = "#f4a582", mid = "white", high = "#92c5de", midpoint = 0,
    name = "Std. Residual"
  ) +
  labs(
    title = "Standardized Residuals: Gender Distribution by Semantic Category",
    x = NULL, y = NULL
  ) +
  theme_minimal(base_family = "Georgia") +
  theme(
```

```
    axis.text.x = element_text(angle = 0, hjust = 0.5, size = 8, face = "bold"),
    axis.text.y = element_text(size = 5),
    plot.title = element_text(size = 10, face = "bold"),
    legend.position = "right",
    legend.title = element_text(size = 6)
  )
```




```
ggsave("plots/Sensitivity/Sampled 0.5 chi-sq residuals ReplaceConcept.png", width = 8, height = 4, dpi =
```

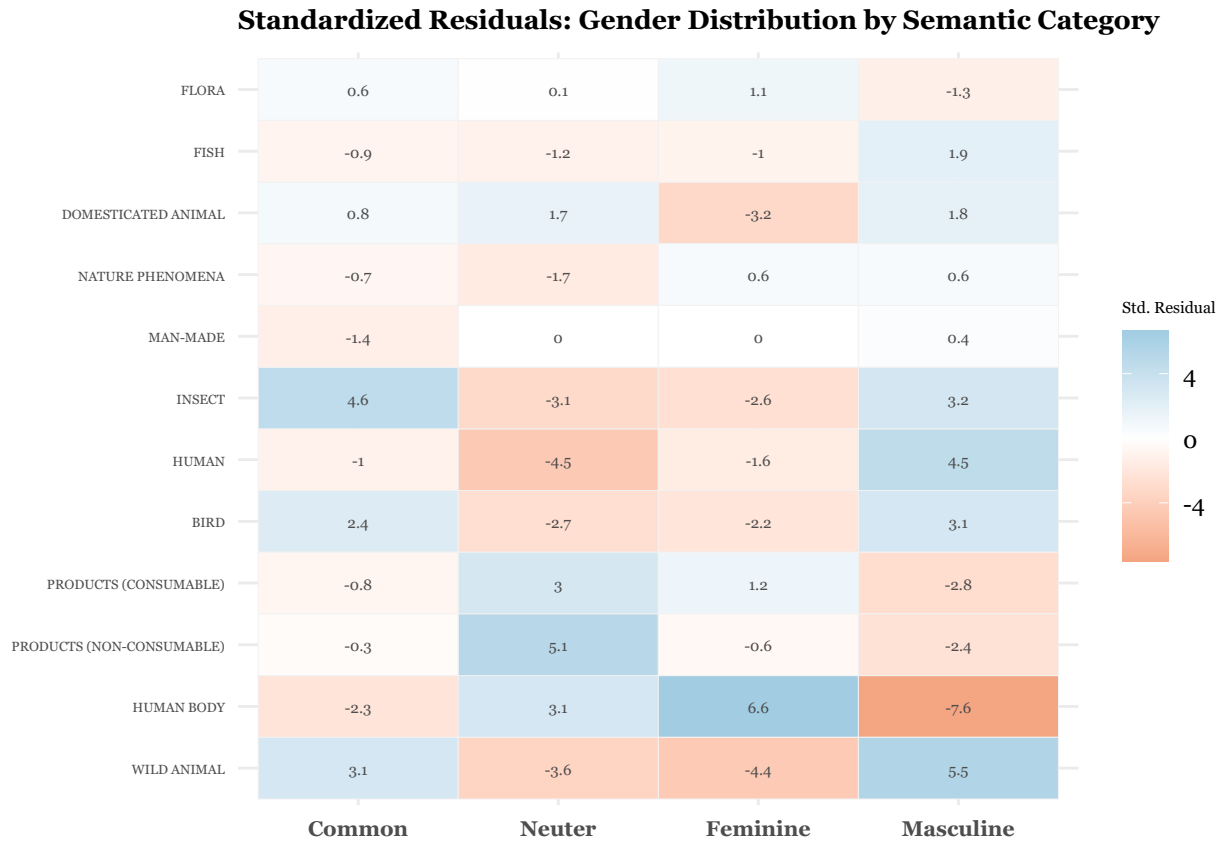
```
###Randomly sample 1,000 data points per language family##
sampled_data_num <- gender_assignment %>%
  group_by(Family) %>%
  sample_n(1000) %>%
  ungroup()

heatmap_data_rep_num <- sampled_data_num %>%
  mutate(Gender = recode(Gender,
                         "C" = "Common",
                         "F" = "Feminine",
                         "M" = "Masculine",
                         "N" = "Neuter")) %>%
  mutate(Gender = factor(Gender, levels = rev(c("Masculine", "Feminine", "Neuter", "Common")))) %>%
  count(ReplaceConcept, Gender) %>%
```

```r
  group_by(ReplaceConcept) %>%
  mutate(Percentage = n / sum(n)) %>%
  ungroup()

contingency_table_rep_num <- heatmap_data_rep_num %>%
  select(ReplaceConcept, Gender, n) %>%
  pivot_wider(names_from = Gender, values_from = n, values_fill = 0) %>%
  column_to_rownames("ReplaceConcept") %>%
  as.matrix()

# Run chi-square test
chisq_result_rep_num <- chisq.test(contingency_table_rep_num)
print(chisq_result_rep_num)
```

```
##
##  Pearson's Chi-squared test
##
## data:  contingency_table_rep_num
## X-squared = 243, df = 33, p-value < 2.2e-16
```

```r
# Extract standardized residuals
residuals_df_num <- as.data.frame(as.table(chisq_result_rep_num$stdres))
colnames(residuals_df_num) <- c("ReplaceConcept", "Gender", "Residual")

# Order by strongest deviation
concept_order_num <- residuals_df_num %>%
  group_by(ReplaceConcept) %>%
  summarize(mean_abs_res = mean(abs(Residual), na.rm = TRUE)) %>%
  arrange(desc(mean_abs_res)) %>%
  pull(ReplaceConcept)

residuals_df_num <- residuals_df_num %>%
  mutate(ReplaceConcept = factor(ReplaceConcept, levels = concept_order))

# Plot standardized residuals with numbers
ggplot(residuals_df_num, aes(x = Gender, y = ReplaceConcept, fill = Residual)) +
  geom_tile(color = "grey95", size = 0.1) +
  geom_text(
    aes(label = round(Residual, 1)),
    size = 2, color = "gray30", family = "Georgia"
  ) +
  scale_fill_gradient2(
    low = "#f4a582", mid = "white", high = "#92c5de", midpoint = 0,
    name = "Std. Residual"
  ) +
  labs(
    title = "Standardized Residuals: Gender Distribution by Semantic Category",
    x = NULL, y = NULL
  ) +
  theme_minimal(base_family = "Georgia") +
  theme(
    axis.text.x = element_text(angle = 0, hjust = 0.5, size = 8, face = "bold"),
    axis.text.y = element_text(size = 5),
```

```
    plot.title = element_text(size = 10, face = "bold"),
    legend.position = "right",
    legend.title = element_text(size = 6)
  )
```




```
ggsave("plots/Sensitivity/Sampled 1000 chi-sq residuals ReplaceConcept.png", width = 8, height = 4, dpi
```

4.2. Inflated dataset

Instead of reducing my data set, now we inflate it artificially to check the same thing.

```
set.seed(143)
inflated_data <- gender_assignment %>%
  group_by(Family) %>%
  sample_n(1000, replace = TRUE) %>%
  ungroup()

# Suppose inflated_data is your bootstrapped dataset
# (already sampled with replacement, keeping all columns)

heatmap_data_rep_inf <- inflated_data %>%
  mutate(Gender = recode(Gender,
                         "C" = "Common",
                         "F" = "Feminine",
```

```
                          "M" = "Masculine",
                          "N" = "Neuter")) %>%
  mutate(Gender = factor(Gender, levels = rev(c("Masculine", "Feminine", "Neuter", "Common")))) %>%
  count(ReplaceConcept, Gender) %>%
  group_by(ReplaceConcept) %>%
  mutate(Percentage = n / sum(n)) %>%
  ungroup()

contingency_table_rep_inf <- heatmap_data_rep_inf %>%
  select(ReplaceConcept, Gender, n) %>%
  pivot_wider(names_from = Gender, values_from = n, values_fill = 0) %>%
  column_to_rownames("ReplaceConcept") %>%
  as.matrix()

# Run chi-square test
chisq_result_rep_inf <- chisq.test(contingency_table_rep_inf)
print(chisq_result_rep_inf)
```

```
##
##  Pearson's Chi-squared test
##
## data:  contingency_table_rep_inf
## X-squared = 223.93, df = 33, p-value < 2.2e-16
```

```
# Extract standardized residuals
residuals_df_inf <- as.data.frame(as.table(chisq_result_rep_inf$stdres))
colnames(residuals_df_inf) <- c("ReplaceConcept", "Gender", "Residual")

# Order by strongest deviation
concept_order_inf <- residuals_df_inf %>%
  group_by(ReplaceConcept) %>%
  summarize(mean_abs_res = mean(abs(Residual), na.rm = TRUE)) %>%
  arrange(desc(mean_abs_res)) %>%
  pull(ReplaceConcept)

residuals_df_inf <- residuals_df_inf %>%
  mutate(ReplaceConcept = factor(ReplaceConcept, levels = concept_order_inf))

# Plot standardized residuals with numbers
ggplot(residuals_df_inf, aes(x = Gender, y = ReplaceConcept, fill = Residual)) +
  geom_tile(color = "grey95", size = 0.1) +
  geom_text(
    aes(label = round(Residual, 1)),
    size = 2, color = "gray30", family = "Georgia"
  ) +
  scale_fill_gradient2(
    low = "#f4a582", mid = "white", high = "#92c5de", midpoint = 0,
    name = "Std. Residual"
  ) +
  labs(
    title = "Standardized Residuals: Gender Distribution by Semantic Category (Inflated Data)",
    x = NULL, y = NULL
  ) +
```

```
  theme_minimal(base_family = "Georgia") +
  theme(
    axis.text.x = element_text(angle = 0, hjust = 0.5, size = 8, face = "bold"),
    axis.text.y = element_text(size = 5),
    plot.title = element_text(size = 10, face = "bold"),
    legend.position = "right",
    legend.title = element_text(size = 6)
  )
```


```
ggsave("plots/Sensitivity/Inflated chi-sq residuals ReplaceConcept.png", width = 8, height = 4, dpi = 3
```

4.3. Sampled data for Animacy and Culture

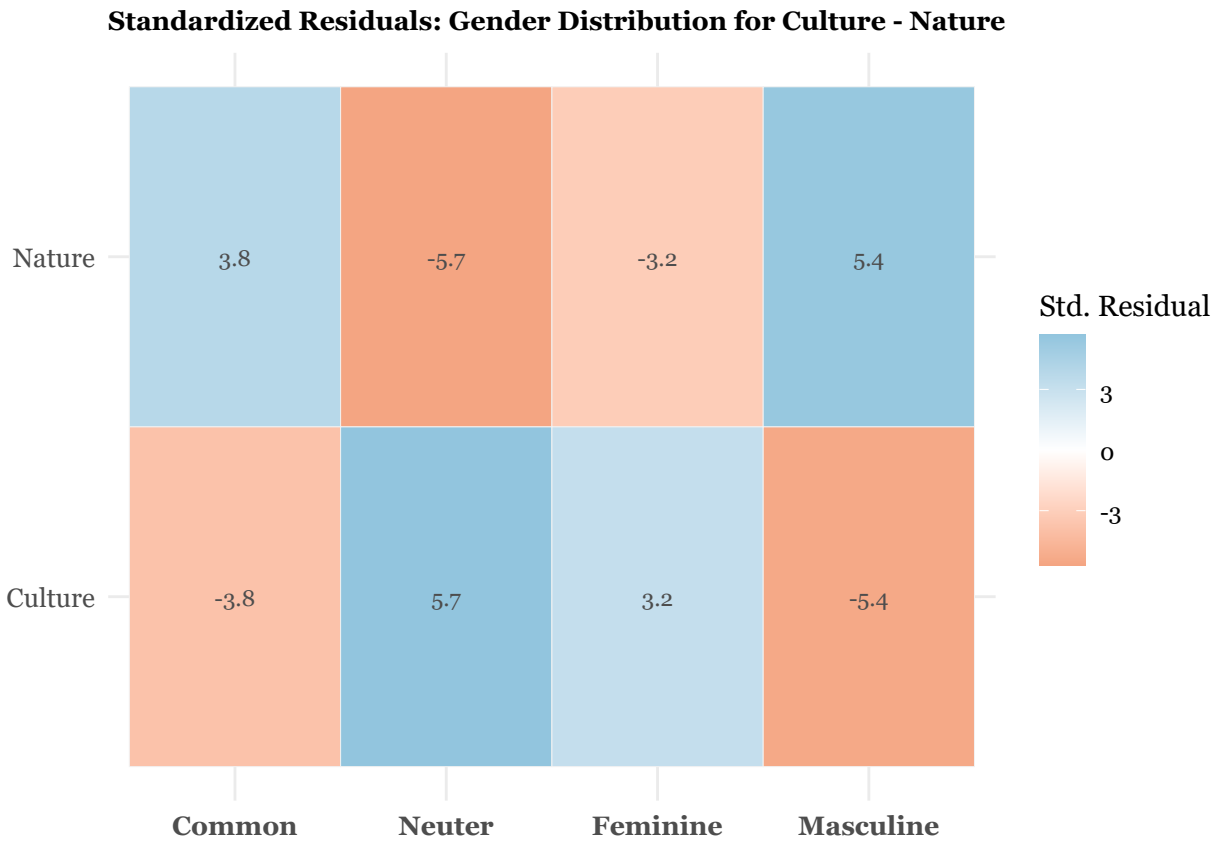
Culture/Nature and Animate/Inanimate Sampled data (1000 points per family):

```
#Culture
cul_nat_num <- sampled_data_num %>%
  mutate(Gender = recode(Gender,
                         "C" = "Common",
                         "F" = "Feminine",
                         "M" = "Masculine",
                         "N" = "Neuter")) %>%
  mutate(Gender = factor(Gender, levels = rev(c("Masculine", "Feminine", "Neuter", "Common")))) %>%
  count(Culture, Gender) %>%
```

```
  group_by(Culture) %>%
  mutate(Percentage = n / sum(n)) %>%
  ungroup()

contingency_table_cul_num <- cul_nat_num %>%
  select(Culture, Gender, n) %>%
  pivot_wider(names_from = Gender, values_from = n, values_fill = 0) %>%
  column_to_rownames("Culture") %>%
  as.matrix()

# Run chi-square test
chisq_result_cul_num <- chisq.test(contingency_table_cul_num)
print(chisq_result_cul_num)
```

```
##
##  Pearson's Chi-squared test
##
## data:  contingency_table_cul_num
## X-squared = 63.591, df = 3, p-value = 1.004e-13
```

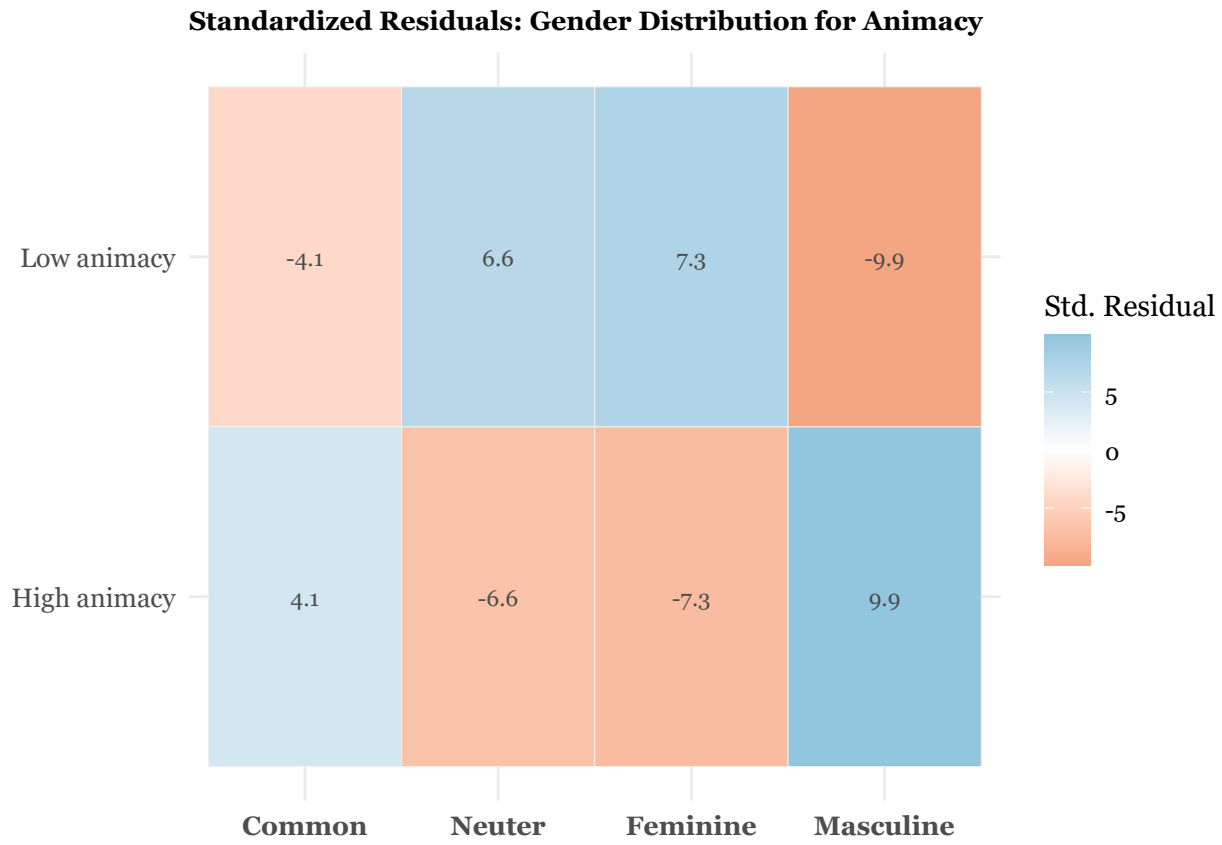
```
# Extract standardized residuals
residuals_cul_num <- as.data.frame(as.table(chisq_result_cul_num$stdres))
colnames(residuals_cul_num) <- c("Culture", "Gender", "Residual")

# Plot standardized residuals with numbers
ggplot(residuals_cul_num, aes(x = Gender, y = Culture, fill = Residual)) +
  geom_tile(color = "grey95", size = 0.1) +
  geom_text(
    aes(label = round(Residual, 1)),
    size = 3, color = "gray30", family = "Georgia"
  ) +
  scale_fill_gradient2(
    low = "#f4a582", mid = "white", high = "#92c5de", midpoint = 0,
    name = "Std. Residual"
  ) +
  labs(
    title = "Standardized Residuals: Gender Distribution for Culture - Nature",
    x = NULL, y = NULL
  ) +
  theme_minimal(base_family = "Georgia") +
  theme(
    axis.text.x = element_text(angle = 0, hjust = 0.5, size = 10, face = "bold"),
    axis.text.y = element_text(size = 10),
    plot.title = element_text(size = 10, face = "bold"),
    legend.position = "right"
  )
```




```
ggsave("plots/Sensitivity/Sampled 0.5 chi-sq residuals Culture.png", width = 8, height = 4, dpi = 300)


#Animate
anim_num <- sampled_data_num %>%
  mutate(Gender = recode(Gender,
                         "C" = "Common",
                         "F" = "Feminine",
                         "M" = "Masculine",
                         "N" = "Neuter")) %>%
  mutate(Gender = factor(Gender, levels = rev(c("Masculine", "Feminine", "Neuter", "Common")))) %>%
  count(Animacy, Gender) %>%
  group_by(Animacy) %>%
  mutate(Percentage = n / sum(n)) %>%
  ungroup()

contingency_table_anim_num <- anim_num %>%
  select(Animacy, Gender, n) %>%
  pivot_wider(names_from = Gender, values_from = n, values_fill = 0) %>%
  column_to_rownames("Animacy") %>%
  as.matrix()

# Run chi-square test
chisq_result_anim_num <- chisq.test(contingency_table_anim_num)
print(chisq_result_anim_num)
```

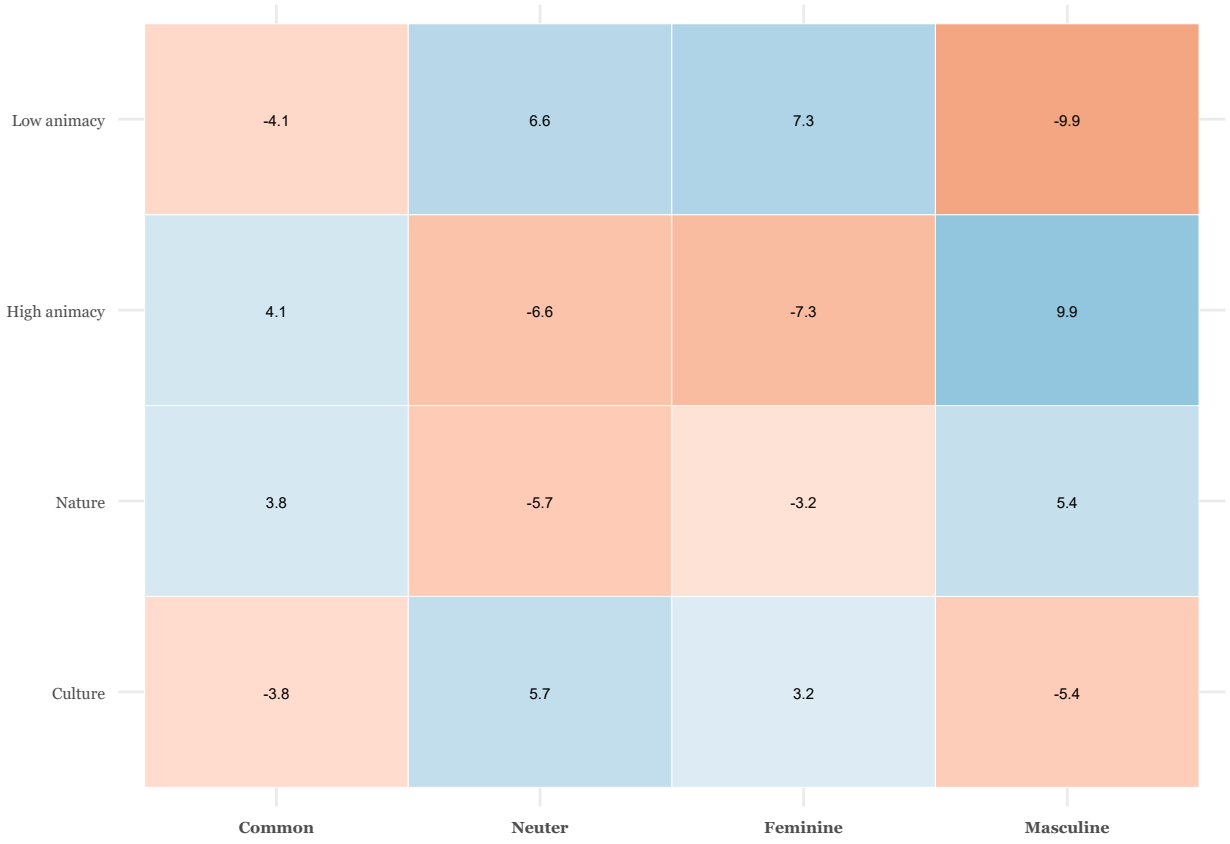
```
##
##  Pearson's Chi-squared test
##
## data:  contingency_table_anim_num
## X-squared = 136.6, df = 3, p-value < 2.2e-16
```

```r
# Extract standardized residuals
residuals_anim_num <- as.data.frame(as.table(chisq_result_anim_num$stdres))
colnames(residuals_anim_num) <- c("Animacy", "Gender", "Residual")

# Plot standardized residuals with numbers
ggplot(residuals_anim_num, aes(x = Gender, y = Animacy, fill = Residual)) +
  geom_tile(color = "grey95", size = 0.1) +
  geom_text(
    aes(label = round(Residual, 1)),
    size = 3, color = "gray30", family = "Georgia"
  ) +
  scale_fill_gradient2(
    low = "#f4a582", mid = "white", high = "#92c5de", midpoint = 0,
    name = "Std. Residual"
  ) +
  labs(
    title = "Standardized Residuals: Gender Distribution for Animacy",
    x = NULL, y = NULL
  ) +
  theme_minimal(base_family = "Georgia") +
  theme(
    axis.text.x = element_text(angle = 0, hjust = 0.5, size = 10, face = "bold"),
    axis.text.y = element_text(size = 10),
    plot.title = element_text(size = 10, face = "bold"),
    legend.position = "right"
  )
```




```r
ggsave("plots/Sensitivity/Sampled 0.5 chi-sq residuals Animacy.png", width = 8, height = 4, dpi = 300)


#Plot them together
# Standardize and add Category column
residuals_anim_num_std <- residuals_anim_num %>%
  rename(Dimension = Animacy) %>%
  mutate(Category = Dimension) %>%   # Store original Animacy value
  select(Category, Gender, Residual, Dimension)

residuals_cul_num_std <- residuals_cul_num %>%
  rename(Dimension = Culture) %>%
  mutate(Category = Dimension) %>%   # Store original Culture value
  select(Category, Gender, Residual, Dimension)

# Combine both
residuals_both_num <- bind_rows(residuals_cul_num_std, residuals_anim_num_std)


# Plot standardized residuals with numbers
ggplot(residuals_both_num, aes(x = Gender, y = Category, fill = Residual)) +
  geom_tile(color = "white") +
  geom_text(aes(label = round(Residual, 1)), size = 2, color = "black") +
  scale_fill_gradient2(
    low = "#f4a582", mid = "white", high = "#92c5de", midpoint = 0,
    name = "Std. Residual"
```

```
  ) +
  theme_minimal(base_family = "Georgia") +
  theme(
    axis.title = element_blank(),
    axis.text.x = element_text(size = 6, face = "bold"),
    axis.text.y = element_text(size = 6),
    axis.ticks = element_blank(),
     legend.position = "none"
  )
```




```
ggsave("plots/Sensitivity/Sampled 0.5 chi-sq residuals Culture and Animacy.png", width = 8, height = 4,
```