World Maps

Chris Bentz and Steven Moran

(02 April, 2024)

Session Info

Give the session info (reduced).

```
## [1] "R version 4.3.2 (2023-10-31)"
```

```
## [1] "aarch64-apple-darwin20"
```

Load Libraries

If the libraries are not installed yet, you need to install them using, for example, the command: install.packages("ggplot2").

```
library(readr)
library(ggmap)
library(maps)
library(gridExtra)
library(ggrepel)
library(RCurl)
```

Give the package versions.

```
##       RCurl     ggrepel   gridExtra        maps       ggmap     ggplot2
## "1.98-1.14"     "0.9.5"       "2.3"     "3.4.2"     "4.0.0"     "3.5.0"
##       readr
##     "2.1.4"
```

Load the language info

Load language info file directly from the github repo.

```
languages <- as.data.frame(read_csv("../../LangInfo/langInfo_TeDDi.csv"))
```

All languages of 100 WALS sample

Simple Stats

```
length(unique(languages$iso639_3)) # number of languages according to iso
```

```
## [1] 100
```

```
length(unique(languages$glottocode)) # number of languages according to glottolog
```

```
## [1] 100
```

```
length(unique(languages$top_level_family)) # number of top level language families according to glottol
```

```
## [1] 61
```

```
length(unique(languages$family_wals)) # number of language families according to wals
```

```
## [1] 68
```

```
unique(languages$macroarea_glotto) # number of macroareas according to glottolog (same as for WALS)
```

```
## [1] "Eurasia"       "Papunesia"     "South America" "Africa"
## [5] "North America" "Australia"
```

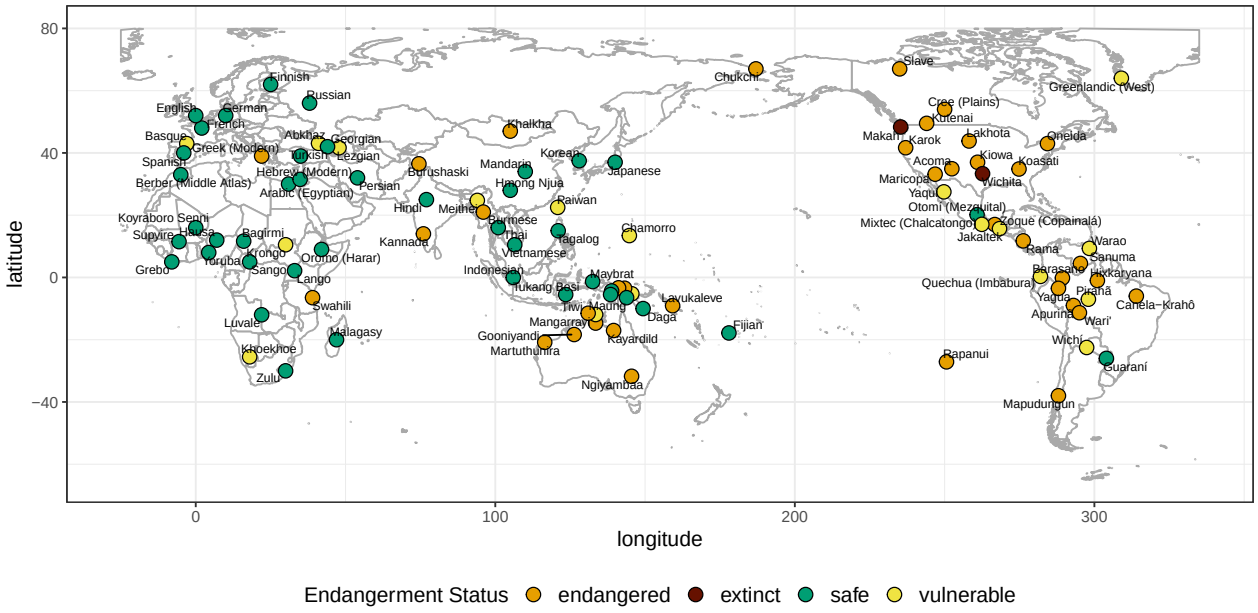
Pre-Processing

```
# add 360 to longitudes for languages with longitudes < -25
# (this is necessary to create a Pacific centered map)
languages$longitude_wals[languages$longitude_wals < -25] <-
  languages$longitude_wals[languages$longitude_wals < -25] + 360
# collapse status information into fewer factors
languages$status[languages$status %in% c("definitely endangered", "severely endangered", "critically en
```

World Map

World maps with endangerment status information from Glottolog.

```
# create world map
world <- map_data("world", wrap = c(-25, 335))
status.map <- ggplot() +
  geom_polygon(data = world, aes(x = long, y = lat, group = group),
               fill = "white", colour = "darkgrey") +
  geom_point(data = languages, aes(x = longitude_wals, y = latitude_wals,
                                   fill = status),
             alpha = 1, size = 3.5, pch = 21) +
             # select colours manually to be color blind safe
             scale_fill_manual(values = c("#E69F00", "#661100", "#009E73", "#F0E442")) +
  geom_text_repel(data = languages, aes(x = longitude_wals, y = latitude_wals,
                                        label = name_wals), size = 2.5,
                  box.padding = unit(0.1, 'lines'), force = 0.5) +
  scale_y_continuous(limits = c(-65, 80)) +
  labs(x = "longitude", y = "latitude", fill = "Endangerment Status") +
  theme_bw() +
  theme(axis.title.x = element_text(size = 12),
        axis.title.y = element_text(size = 12),
        title = element_text(size = 12),
        legend.title = element_text(size = 12),
        legend.text = element_text(size = 12),
        legend.position = "bottom")
status.map
```



Save to file.

```
ggsave("WorldMap_TeDDi.pdf", status.map,
       dpi = 300, scale = 1, width = 12, height = 6, device = cairo_pdf)
```

Languages for which texts are currently available in the corpus

Exclude languages for which there is currently no data.

```
# give selection of languages which are currently not represented by text data
# (see folder Reports/line_counts/line_counts.csv on github)
missing <- c("Karok", "Koasati", "Koyraboro Senni", "Krongo", "Lakhota", "Lezgian",
             "Mangarrayi", "Meithei", "Maricopa", "Slave", "Oneida", "Supiyre", "Tukang Besi")
# define a "no tin" operator
`%notin%` <- Negate(`%in%`)
# exclude these languages from the language info data frame
languages.corpus <- languages[languages$name_wals %notin% missing, ]
```

Simple Stats

```
length(unique(languages.corpus$iso639_3)) # number of languages according to iso
```

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

```
length(unique(languages.corpus$glottocode)) # number of languages according to glottolog
```

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

```
length(unique(languages.corpus$top_level_family)) # number of top level language families according to
```

```
## [1] 52
```

```
length(unique(languages.corpus$family_wals)) # number of language families according to wals
```

```
## [1] 58
```

```r
unique(languages.corpus$macroarea_glotto) # number of macroareas according to glottolog (same as for WA
```

```
## [1] "Eurasia"       "Papunesia"     "South America" "Africa"
## [5] "North America" "Australia"
```

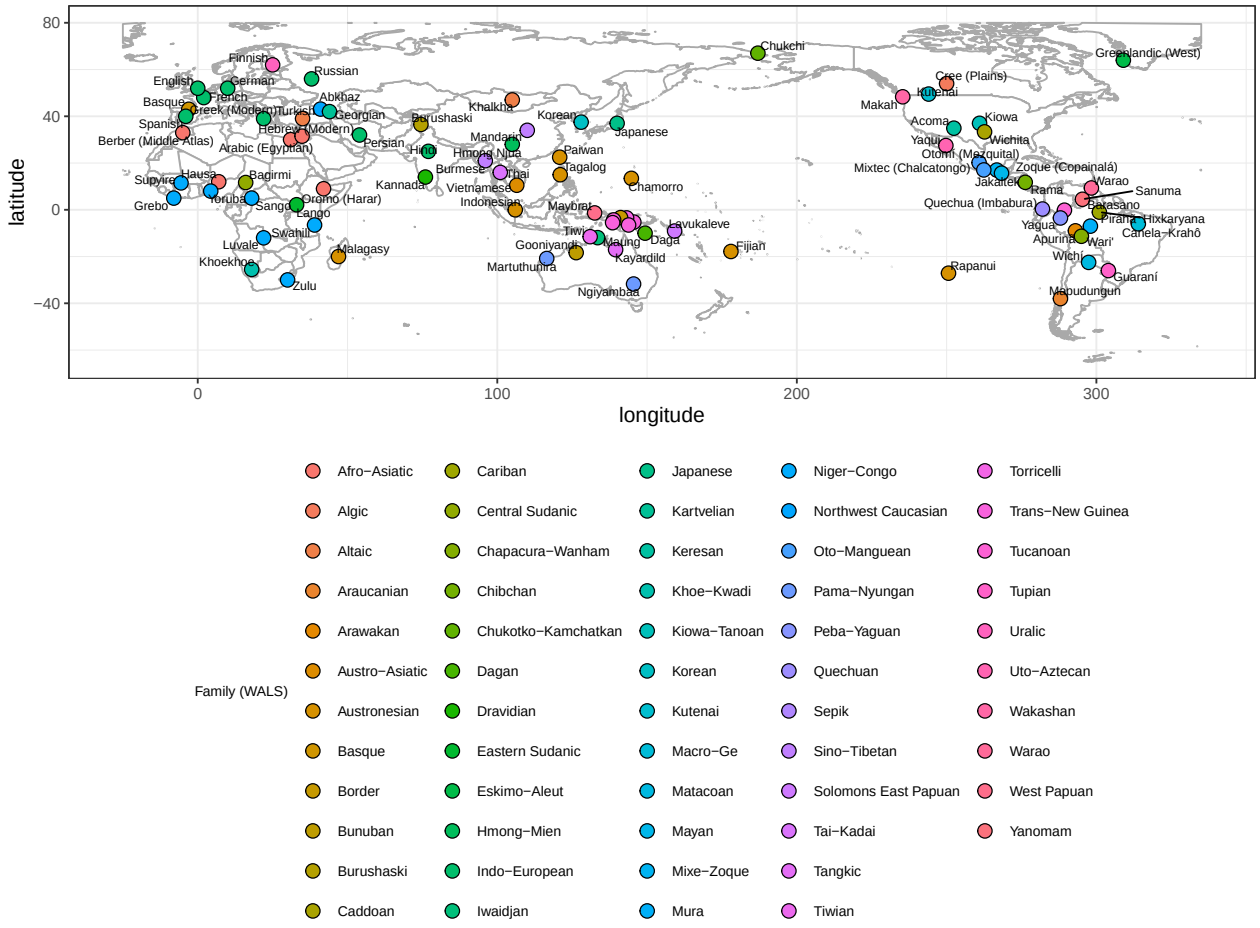
Pre-Processing

```r
# add 360 to longitudes for languages with longitudes < -25
# (this is necessary to create a Pacific centered map)
languages.corpus$longitude_wals[languages.corpus$longitude_wals < -25] <-
  languages.corpus$longitude_wals[languages.corpus$longitude_wals < -25] + 360
```

World Map

World maps with family information.

```r
# create world map
world <- map_data("world", wrap = c(-25, 335))
family.map <- ggplot() +
  geom_polygon(data = world, aes(x = long, y = lat, group = group),
               fill = "white", colour = "darkgrey") +
  geom_point(data = languages.corpus, aes(x = longitude_wals, y = latitude_wals,
                                          fill = family_wals),
             alpha = 1, size = 3.5, pch = 21) +
             # select colors manually to be color blind safe
             # scale_fill_manual(values = c("#E69F00", "#661100", "#009E73", "#F0E442")) +
  geom_text_repel(data = languages.corpus, aes(x = longitude_wals, y = latitude_wals,
                                               label = name_wals), size = 2.5,
                  box.padding = unit(0.1, 'lines'), force = 0.5) +
  scale_y_continuous(limits = c(-65, 80)) +
  labs(x = "longitude", y = "latitude", fill = "Family (WALS)") +
  theme_bw() +
  theme(axis.title.x = element_text(size = 12),
        axis.title.y = element_text(size = 12),
        title = element_text(size = 8),
        legend.title = element_text(size = 8),
        legend.text = element_text(size = 8),
        legend.position = "bottom")
family.map
```




```
ggsave("worldMap_CurrentCorpus.pdf", family.map,
       dpi = 300, scale = 1, width = 10, height = 7.5, device = cairo_pdf)
```