Supplementary materials for: Consonant stability in Portuguese-based creoles

Steven Moran and Carlos Silva

(14 June, 2024)

Contents

1 Overview

Supplementary materials for, "Consonant Stability in Portuguese-based creoles", by Carlos Silva and Steven Moran, to appear in Diachronica.

These supplementary materials are available online at:

- https://github.com/CreoPhon/consonant_stability_portuguese_creoles

In this report, we provide code in R (R Core Team 2023) and we use these R libraries (Wickham et al. 2019; Xie 2021; Slowikowski 2022; Kuznetsova, Brockhoff, and Christensen 2017; Wood 2004; Pohlert 2023; Kassambara 2023a, 2023b; Hennig and Hausdorf 2023; Becker et al. 2023):

```
library(tidyverse)
library(knitr)
library(ggrepel)
library(lmerTest)
library(mgcv)
library(PMCMRplus)
library(ggpubr)
library(rstatix)
library(stats)
library(prabclus)
library(maps)

# Set the theme for all figures
theme_set(theme_bw())
```

Load the data set.

```
database <- read_csv("database.csv")
```

We extend the database with some additional variables. First, duration of contact.

```
database$duration <- database$`EndOfInfluence` - database$`FirstMajorSettlement`
```

Next, a variable of global stability.

```
database <- mutate(database, GlobalStability = (PlaceStability + MannerStability) / 2)
```

Also, a categorical variable for duration.

```
database <- database %>%
  mutate(duration_group = ifelse(duration <= 250, "short", "long"))
```

And a categorical variable for changes in manner and/or place. Stability in the database is '1' (no change) and '0' (change).

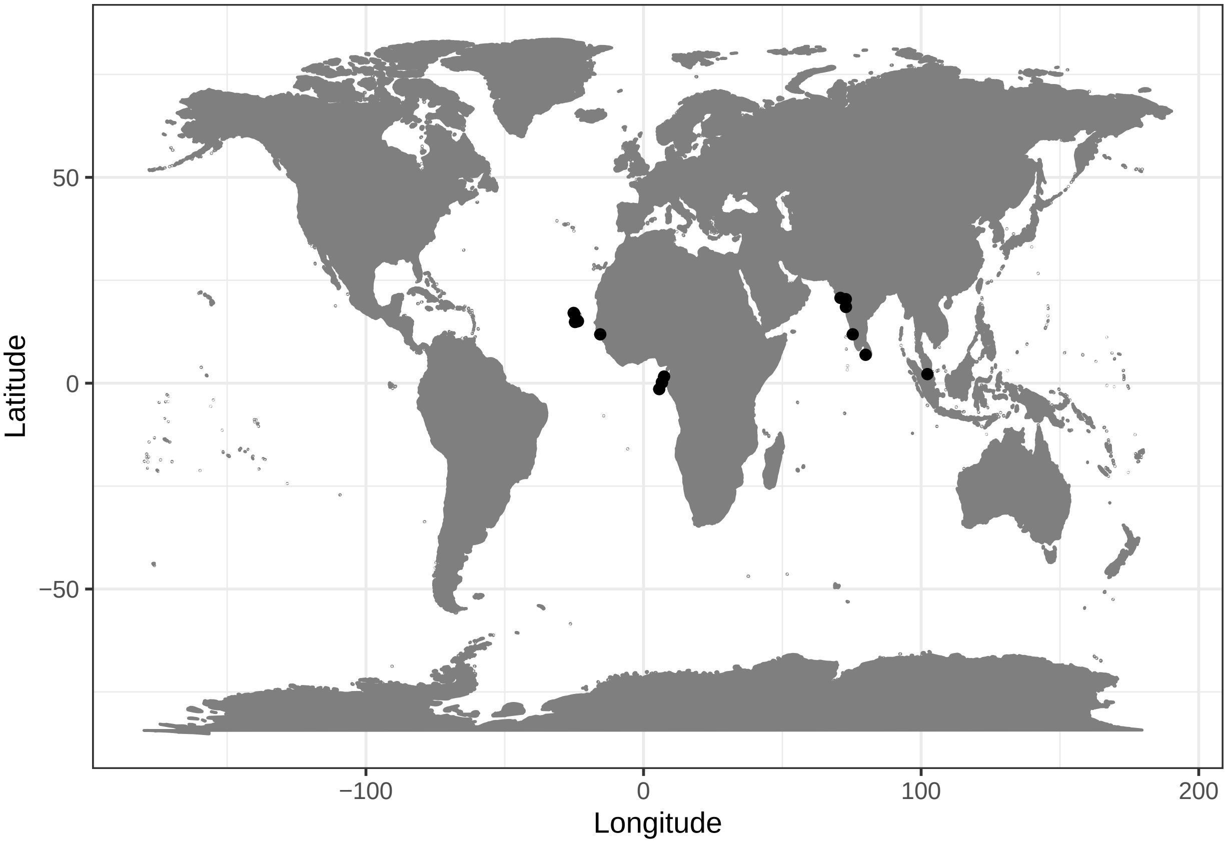
```
database <- database %>%
  mutate(categorical_stability = ifelse(PlaceStability == 1 & MannerStability == 1,
                                        "no manner/no place", NA))

database <- database %>%
  mutate(categorical_stability = ifelse(PlaceStability == 1 & MannerStability == 0,
                                        "manner/no place", categorical_stability))

database <- database %>%
  mutate(categorical_stability = ifelse(PlaceStability == 0 & MannerStability == 1,
                                        "no manner/place", categorical_stability))

database <- database %>%
  mutate(categorical_stability = ifelse(PlaceStability == 0 & MannerStability == 0,
                                        "manner/place", categorical_stability))

table(database$categorical_stability)
```

```
##
##    manner/no place       manner/place no manner/no place    no manner/place
##                 49                 58                553                 25
```

2 Language sample

Here are the languages plotted on a world map.

```
locations <- read_csv('creole_locations.csv')

ggplot(data = locations, aes(x = Longitude, y = Latitude)) +
  borders("world", colour="gray50", fill="gray50") +
  geom_point()
```



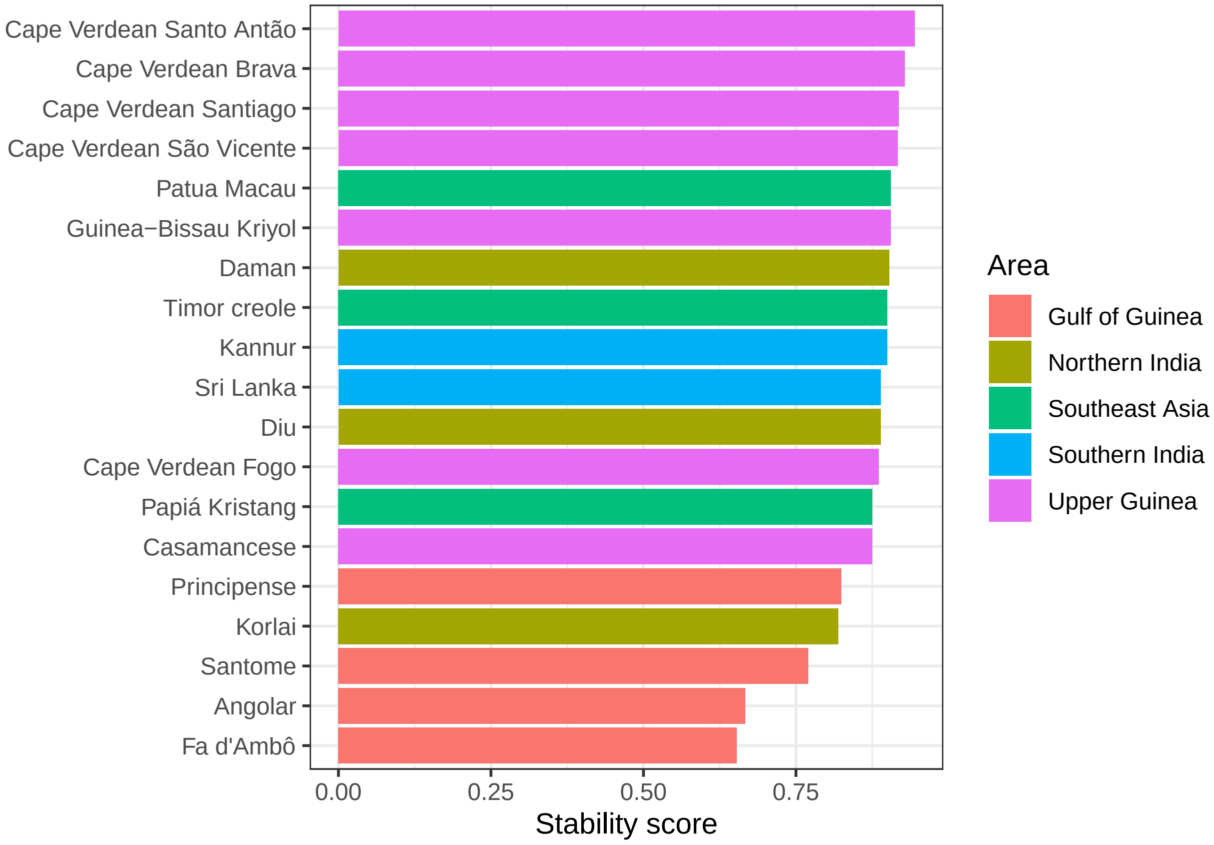
3 Creole stability

Which creoles in the sample are more or less stable overall?

```
creole_stability <- database %>%
  group_by(Language, Area, duration, duration_group, ContactConditions) %>%
  summarize(MeanStability = mean(GlobalStability, na.rm = TRUE))

write_csv(creole_stability, 'creole_stability.csv')
```

Plot it by area.

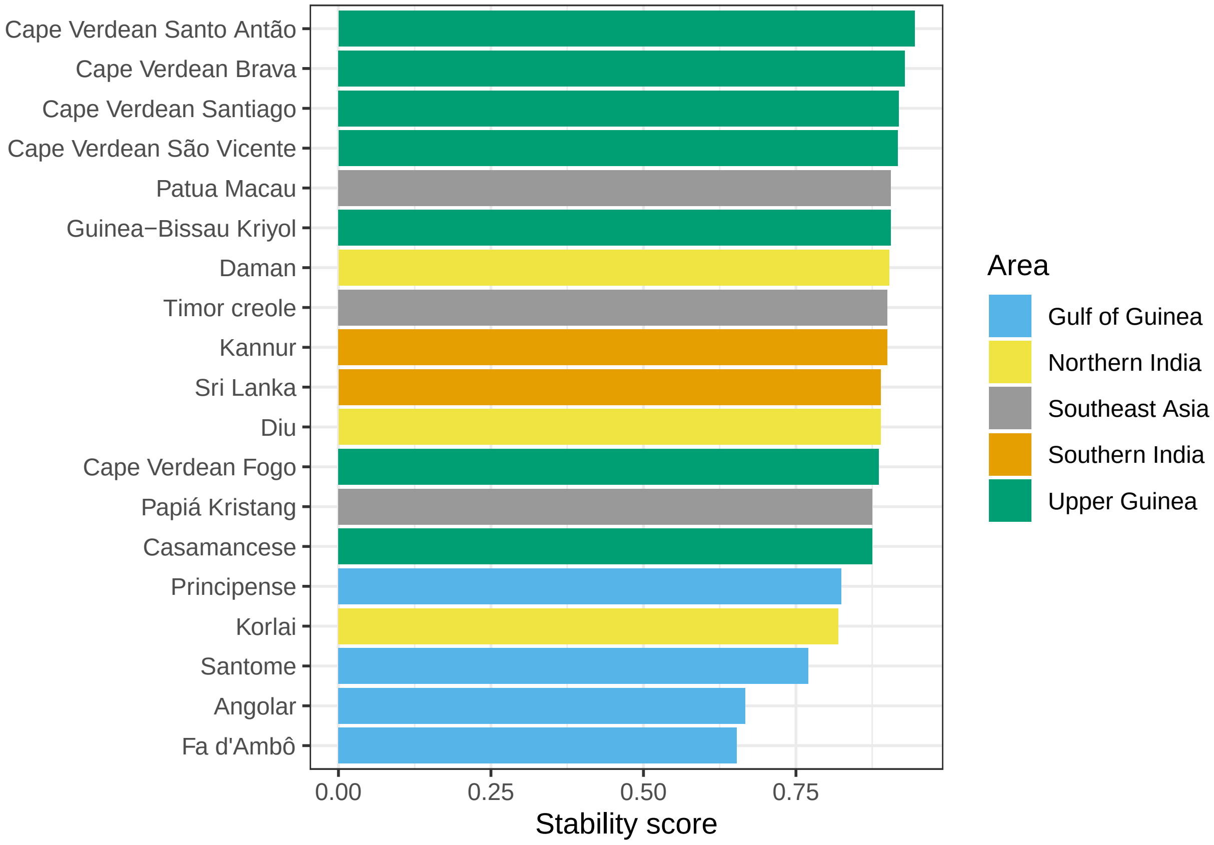
```
ggplot(creole_stability) +
  geom_bar(aes(x = MeanStability, y = reorder(Language, MeanStability),
               fill = Area), stat = "identity", show.legend = TRUE)  +
  theme(axis.title.y = element_blank()) +
  labs(x = "Stability score")
```



Plot it by area with a color-blind friendly palette.

```r
# cbPalette <- c("#999999", "#E69F00", "#56B4E9", "#009E73", "#F0E442", "#0072B2", "#D55E00", "#CC79A7"
cbPalette <- c("#56B4E9", "#F0E442", "#999999", "#E69F00", "#009E73", "#0072B2", "#D55E00", "#CC79A7")
# cbbPalette <- c("#000000", "#E69F00", "#56B4E9", "#009E73", "#F0E442", "#0072B2", "#D55E00", "#CC79A7
```

```r
ggplot(creole_stability) +
  geom_bar(aes(x = MeanStability, y = reorder(Language, MeanStability),
               fill = Area), stat = "identity", show.legend = TRUE)  +
  theme(axis.title.y = element_blank()) +
  labs(x = "Stability score") +
  scale_fill_manual(values=cbPalette)
```



```r
table(creole_stability$Area)
```

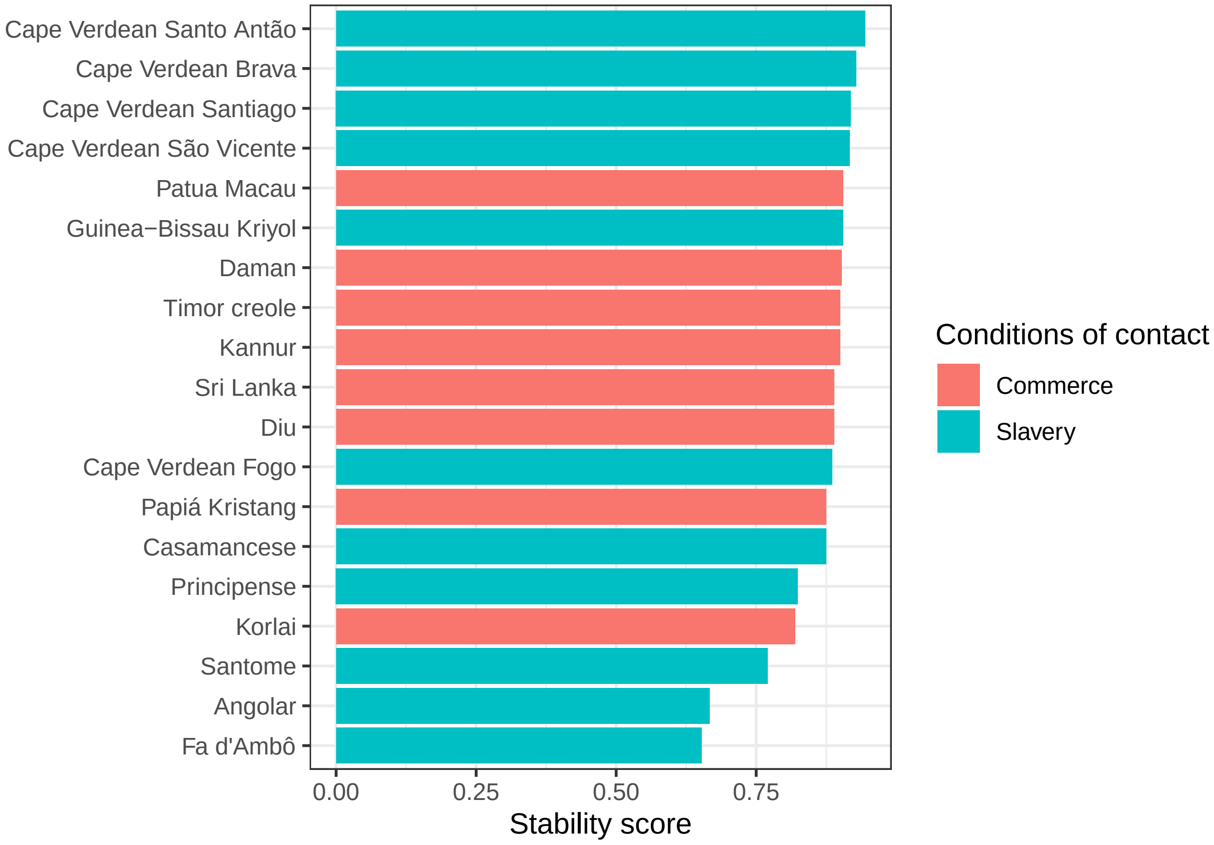
```
##
## Gulf of Guinea Northern India Southeast Asia Southern India   Upper Guinea
##              4              3              3              2              7
```

3.1 Conditions of contact

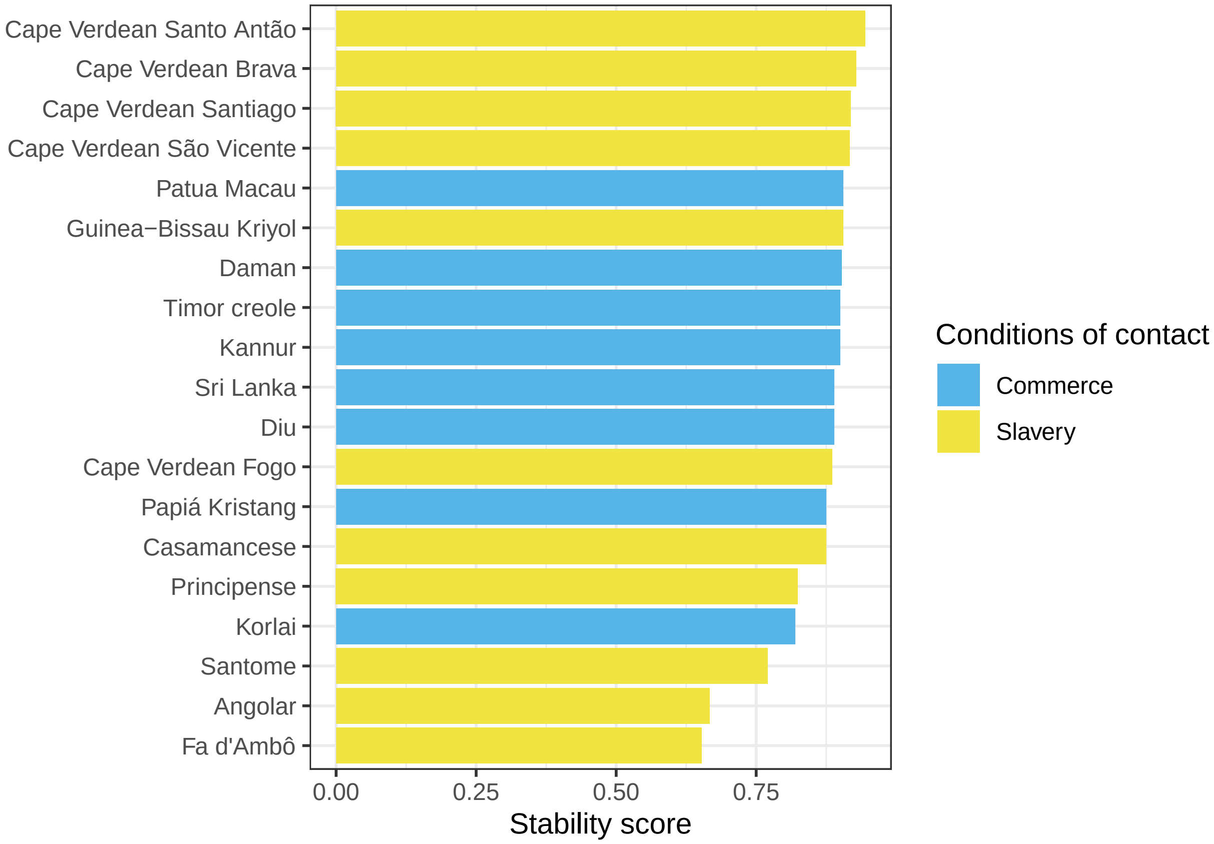
We have the overall stability values. What are these in relation to the conditions of contact?

The finding that "slavery has a negative impact on stability" was mainly observational (e.g. Faraclas et al. (2007); Carvalho and Lucchesi (2016)). Upper Guinea are considered "light" creoles, i.e., enslaved people with "lighter" contact conditions. Gulf of Guinea are considered "hard" creole, i.e., enslaved people under "harder" contact conditions.

```r
ggplot(creole_stability) +
  geom_bar(aes(x = MeanStability, y = reorder(Language, MeanStability),
             fill = ContactConditions), stat = "identity", show.legend = TRUE
  ) +
  theme(axis.title.y = element_blank()) +
  labs(x = "Stability score", fill = "Conditions of contact")
```



```r
ggplot(creole_stability) +
  geom_bar(aes(x = MeanStability, y = reorder(Language, MeanStability),
               fill = ContactConditions), stat = "identity", show.legend = TRUE
  ) +
  theme(axis.title.y = element_blank()) +
  labs(x = "Stability score", fill = "Conditions of contact") +
  scale_fill_manual(values=cbPalette)
```



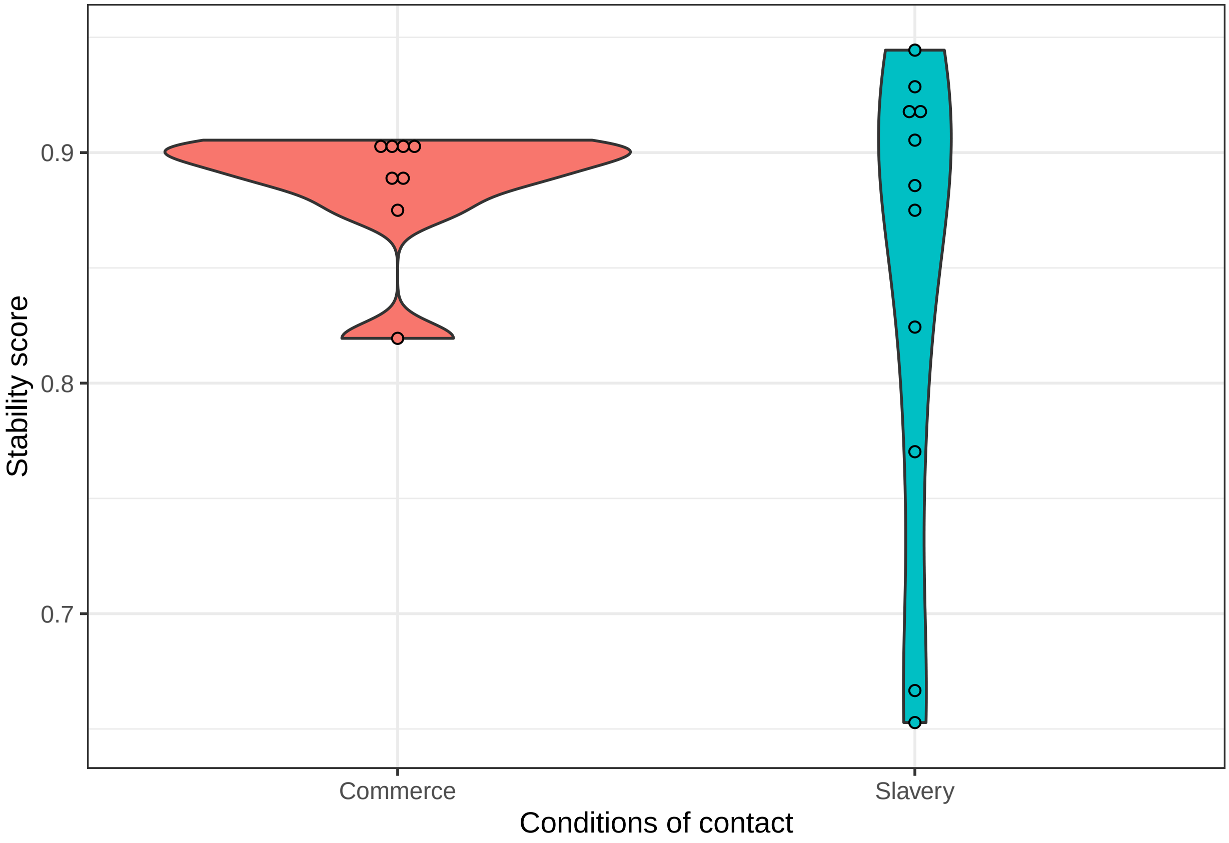
Test whether there is a relation between type of contact situation and overall mean stability with a linear model.

```
m <- lm(MeanStability ~ ContactConditions, data = creole_stability)
summary(m)
```

```
##
## Call:
## lm(formula = MeanStability ~ ContactConditions, data = creole_stability)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.19165 -0.01508  0.01495  0.05113  0.10001
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)               0.88505    0.02896  30.562 2.68e-16 ***
## ContactConditionsSlavery -0.04062    0.03806  -1.067    0.301
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08191 on 17 degrees of freedom
## Multiple R-squared:  0.06279,    Adjusted R-squared:  0.007662
## F-statistic: 1.139 on 1 and 17 DF,  p-value: 0.3008
```

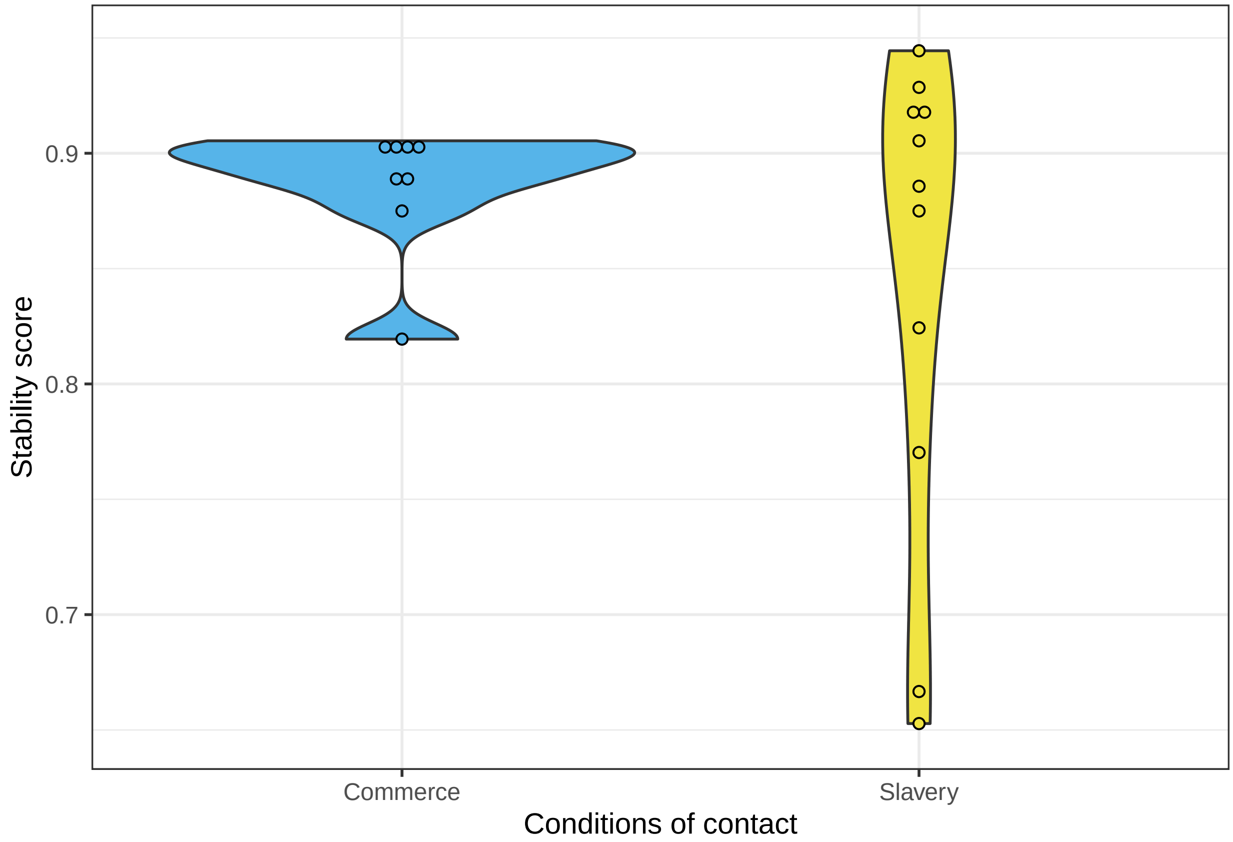
Visualize the results.

```
ggplot(creole_stability, aes(x = ContactConditions, y = MeanStability,
                             fill = ContactConditions)) +
  geom_smooth(method = "lm", colour="black") +
  geom_violin() +
  geom_dotplot(binaxis = "y",
               stackdir = "center",
               dotsize = 0.5)  +
  theme(legend.position="none") +
  labs(y = "Stability score", x = "Conditions of contact")
```



```
ggplot(creole_stability, aes(x = ContactConditions, y = MeanStability,
                              fill = ContactConditions)) +
  geom_smooth(method = "lm", colour="black") +
  geom_violin() +
  geom_dotplot(binaxis = "y",
               stackdir = "center",
               dotsize = 0.5)  +
  theme(legend.position="none") +
  labs(y = "Stability score", x = "Conditions of contact") +
  scale_fill_manual(values=cbPalette)
```

```
## `geom_smooth()` using formula = 'y ~ x'
## Bin width defaults to 1/30 of the range of the data. Pick better value with
## `binwidth`.
```

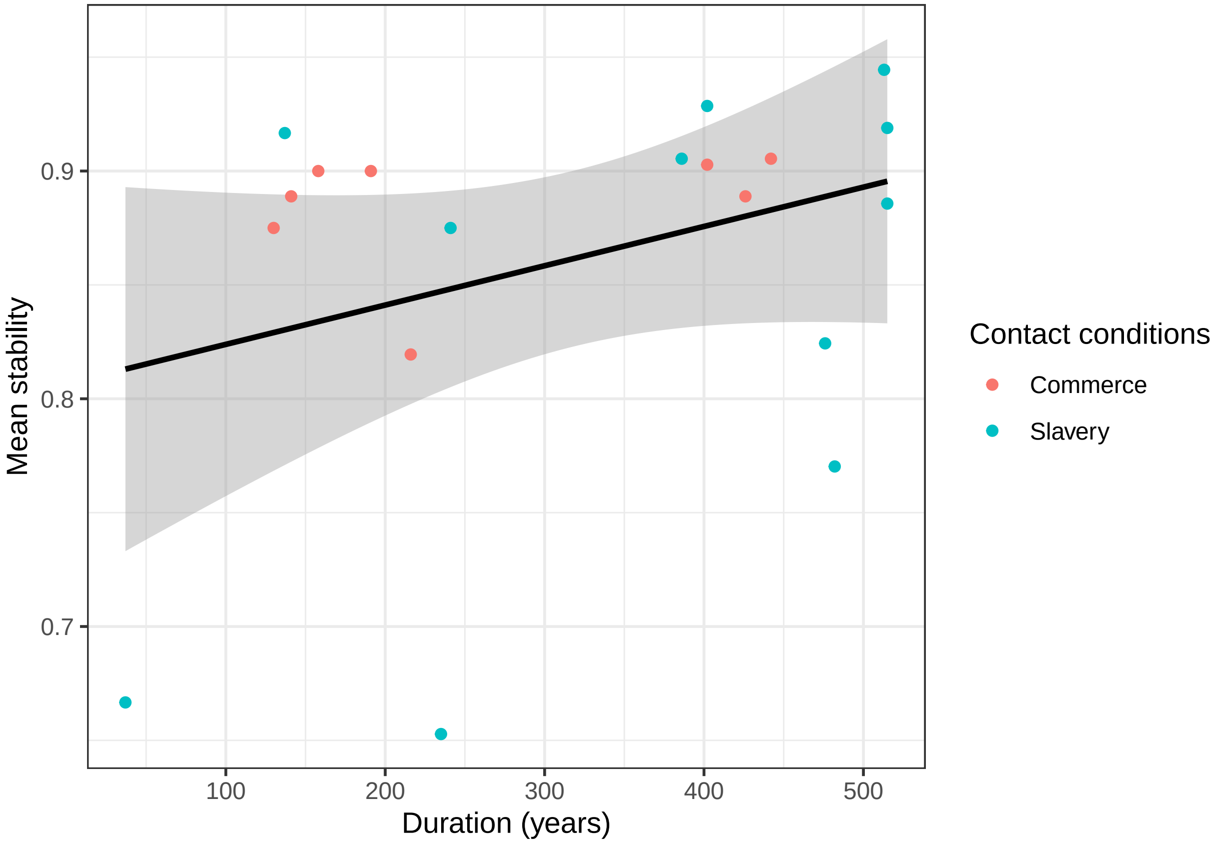


Statistical summary.

```r
m <- lm(MeanStability ~ duration + ContactConditions * duration,
        data = creole_stability)
summary(m)
```

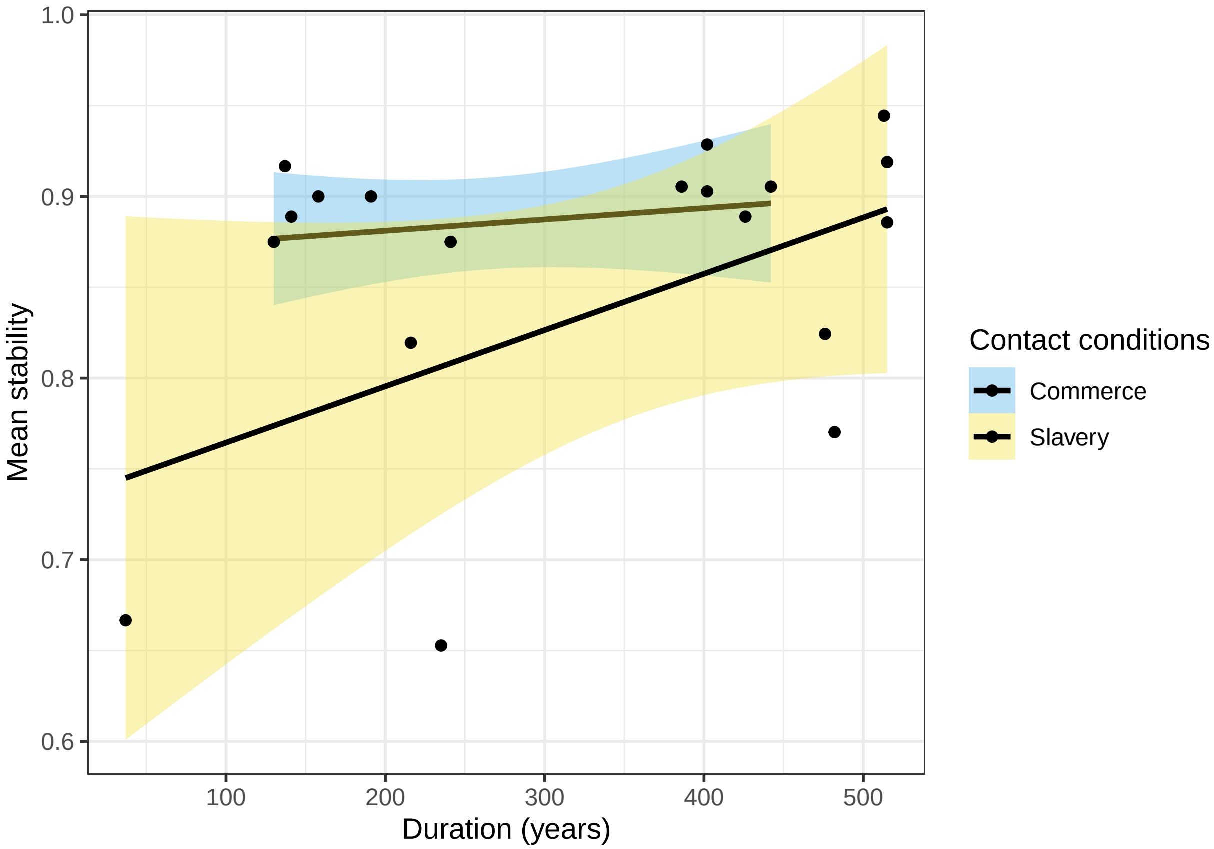
```
##
## Call:
## lm(formula = MeanStability ~ duration + ContactConditions * duration,
##     data = creole_stability)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.153522 -0.031982  0.009208  0.038949  0.140726
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       8.686e-01  6.167e-02  14.086  4.7e-10 ***
## duration                          6.236e-05  2.110e-04   0.296    0.772
## ContactConditionsSlavery         -1.351e-01  8.305e-02  -1.627    0.125
## duration:ContactConditionsSlavery  2.474e-04  2.541e-04   0.974    0.346
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07576 on 15 degrees of freedom
## Multiple R-squared:  0.2925, Adjusted R-squared:  0.151
## F-statistic: 2.067 on 3 and 15 DF,  p-value: 0.1477
```

```r
ggplot(creole_stability, aes(x = duration, y = MeanStability, color = ContactConditions)) +
  geom_smooth(method = "lm", colour="black") +
  geom_point() +
  xlab("Duration (years)") +
  ylab("Mean stability") +
  labs(color = "Contact conditions")
```



```r
ggplot(creole_stability, aes(x = duration, y = MeanStability, fill = ContactConditions)) +
  geom_smooth(method = "lm", colour="black") +
  geom_point() +
  xlab("Duration (years)") +
  ylab("Mean stability") +
  labs(fill = "Contact conditions") +
  scale_fill_manual(values=cbPalette)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

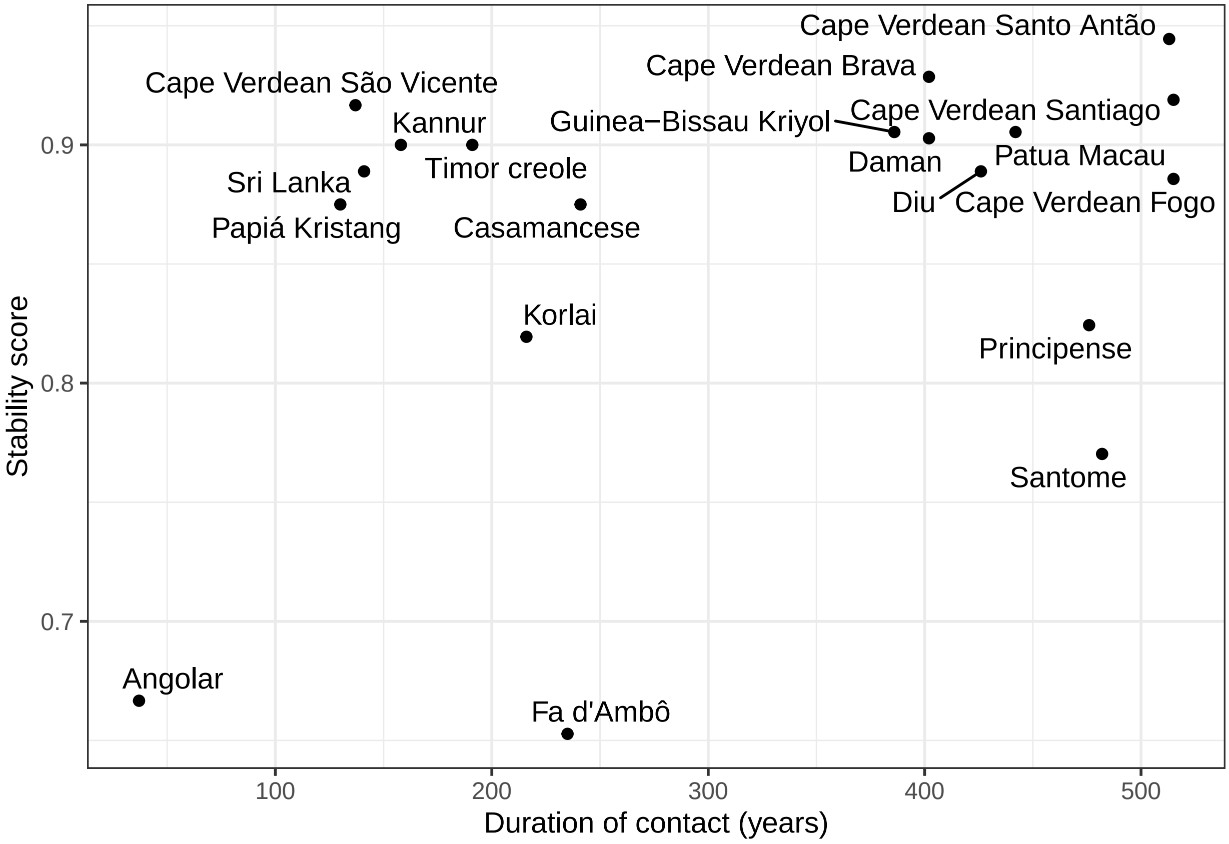


3.2 Duration of contact

We have the overall stability values. What are these in relation to the duration of contact?

There does not seem to be a relationship between overall duration and overall stability.

```
ggplot(creole_stability, aes(x = duration, y = MeanStability)) +
  geom_point() +
  geom_text_repel(aes(label = creole_stability$Language)) +
  xlab("Duration of contact (years)") +
  ylab("Stability score")
```



Results from a linear regression.

```
msd <- lm(MeanStability ~ duration, data = creole_stability)
summary(msd)
```

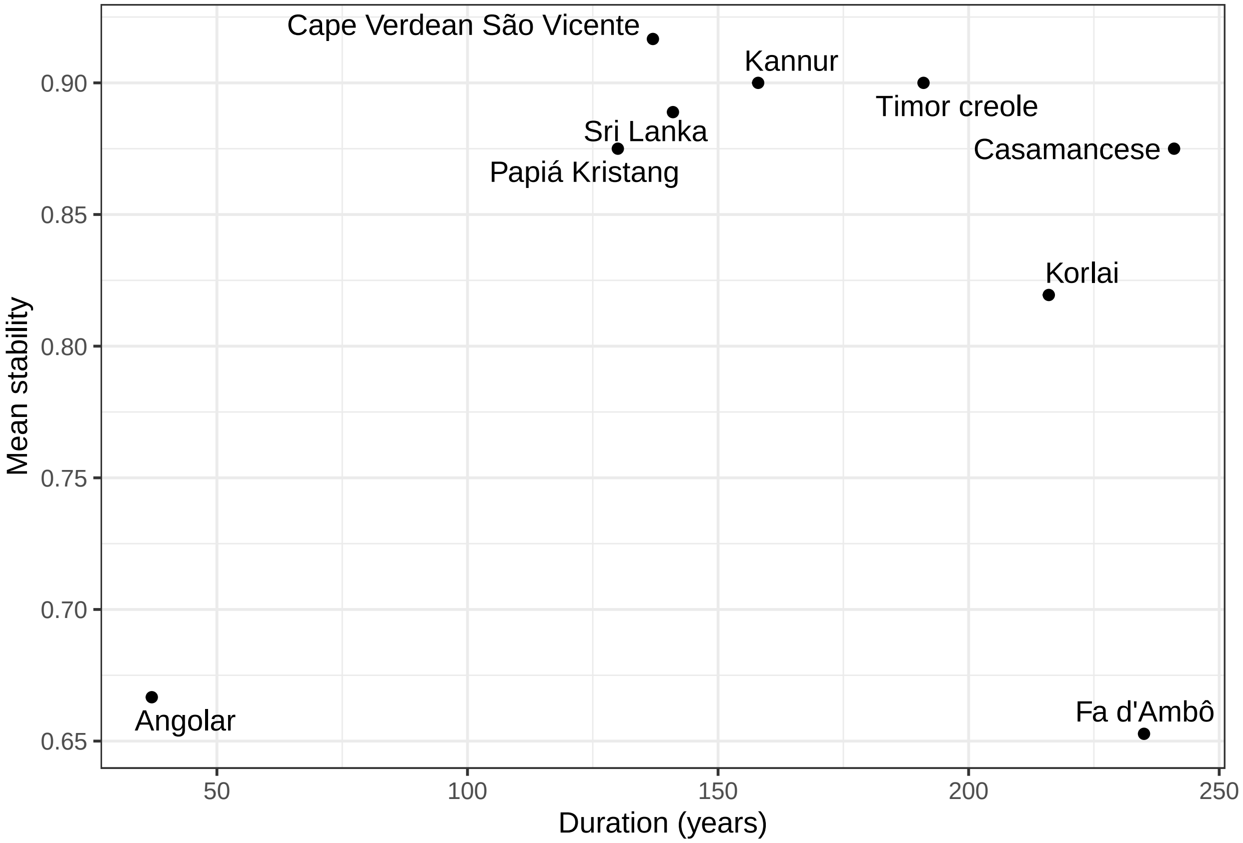
```
##
## Call:
## lm(formula = MeanStability ~ duration, data = creole_stability)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.19440 -0.01713  0.02677  0.05092  0.08640
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.8066224  0.0417628  19.314 5.29e-13 ***
## duration    0.0001726  0.0001180   1.463    0.162
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07974 on 17 degrees of freedom
## Multiple R-squared:  0.1118, Adjusted R-squared:  0.05953
## F-statistic: 2.139 on 1 and 17 DF,  p-value: 0.1618
```

However, there does seem to be two groups of languages – ones that belong to “long duration” (>= 400 years) and those that belong to “short duration” (<= 250 years).

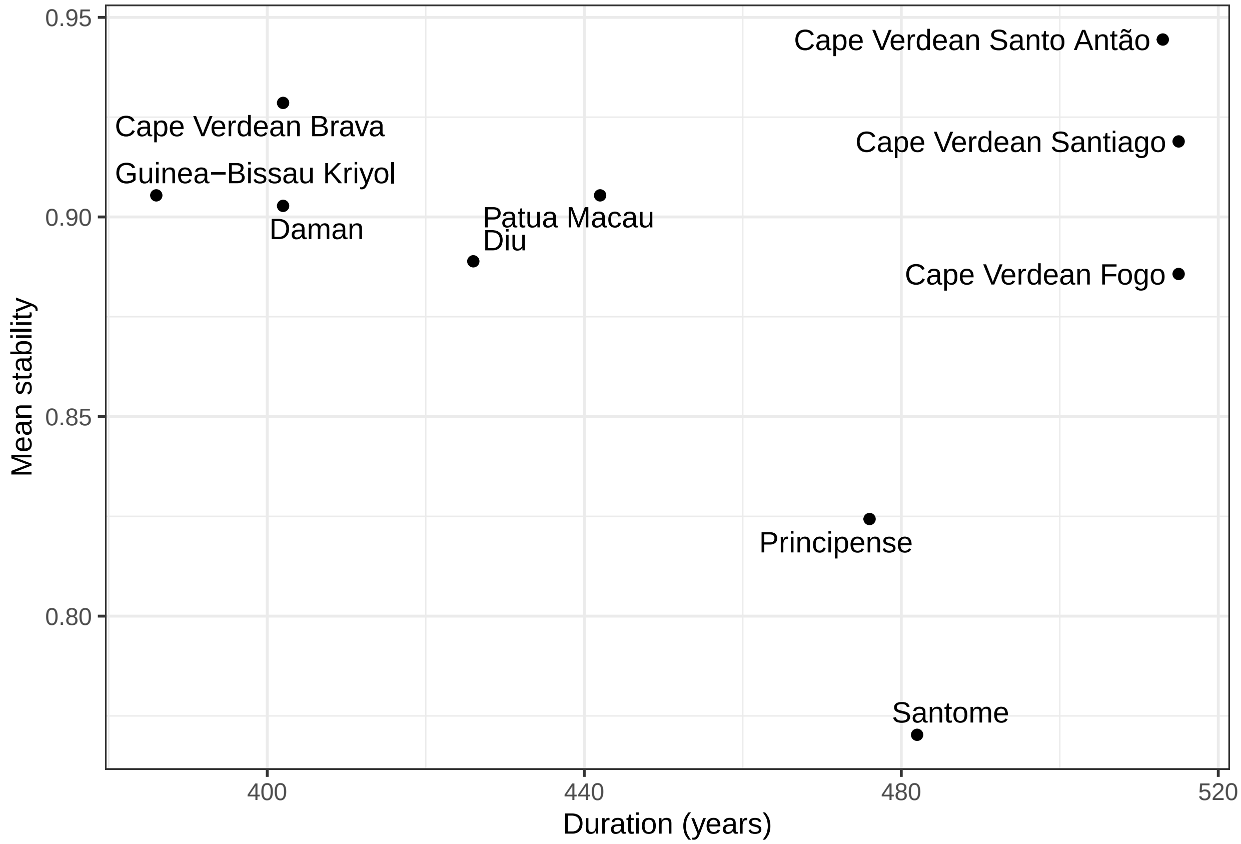
We split the data and rerun the models, but we note that there are very few data points.

```
tmp_short <- creole_stability %>% filter(duration <= 250)
tmp_long <- creole_stability %>% filter(duration > 250)
```

```
ggplot(tmp_short, aes(x = duration, y = MeanStability)) +
  geom_point() +
  geom_text_repel(aes(label = tmp_short$Language)) +
  xlab("Duration (years)") +
  ylab("Mean stability")
```



```
ggplot(tmp_long, aes(x = duration, y = MeanStability)) +
  geom_point() +
  geom_text_repel(aes(label = tmp_long$Language)) +
  xlab("Duration (years)") +
  ylab("Mean stability")
```

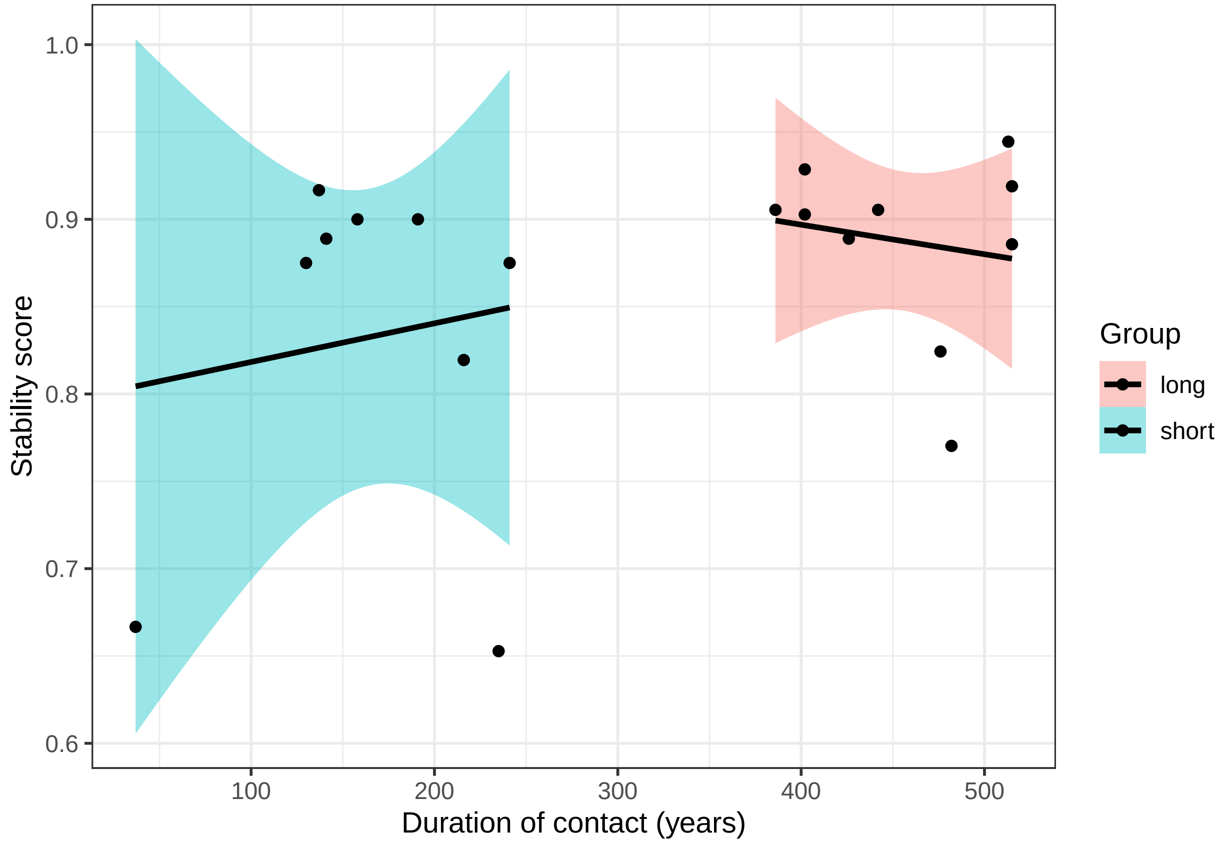


Here is a single model with an interaction term MeanSim ~ duration, group * duration.

```
msd <- lm(MeanStability ~ duration + duration_group * duration, data = creole_stability)
summary(msd)
```

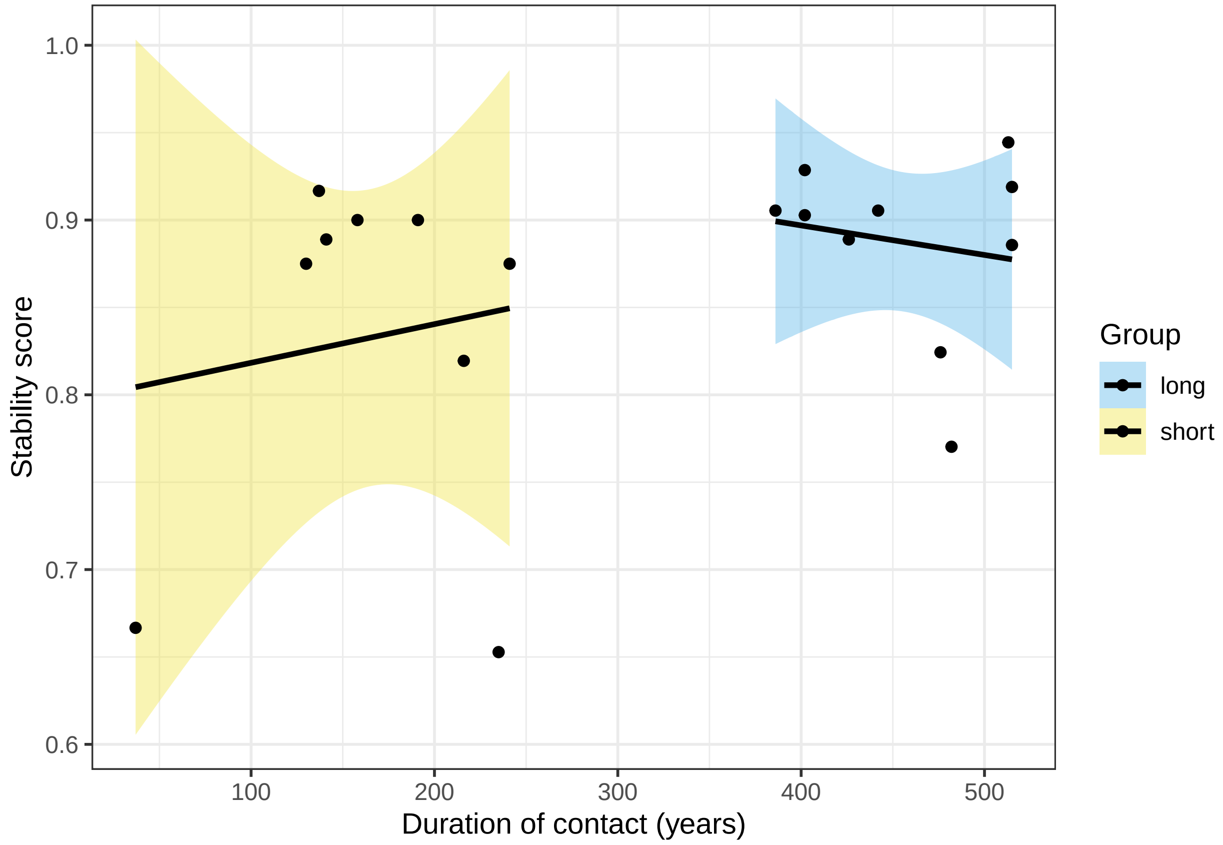
```
##
## Call:
## lm(formula = MeanStability ~ duration + duration_group * duration,
##     data = creole_stability)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.19540 -0.01408  0.01558  0.05578  0.09017
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     0.9645090  0.2525341   3.819  0.00168 **
## duration                       -0.0001690  0.0005509  -0.307  0.76325
## duration_groupshort            -0.1683250  0.2652407  -0.635  0.53524
## duration:duration_groupshort    0.0003902  0.0007185   0.543  0.59501
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08375 on 15 degrees of freedom
## Multiple R-squared:  0.1354, Adjusted R-squared:  -0.03753
## F-statistic: 0.783 on 3 and 15 DF,  p-value: 0.5218
```

```
ggplot(creole_stability, aes(x = duration, y = MeanStability, fill = duration_group)) +
  geom_smooth(method = "lm", colour="black") +
  geom_point() +
  xlab("Duration of contact (years)") +
  ylab("Stability score") +
  labs(fill = "Group")
```



```
ggplot(creole_stability, aes(x = duration, y = MeanStability, fill = duration_group)) +
  geom_smooth(method = "lm", colour="black") +
  geom_point() +
  xlab("Duration of contact (years)") +
  ylab("Stability score") +
  labs(fill = "Group") +
  scale_fill_manual(values=cbPalette)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



The variability in the two groups is very different. The direction of the effect is interesting: shorter durations yield more stability more consistently. Over time, the variability in mean stability increases. Time is “destabillizing the pattern of stability” (pc Nicholas Lester).

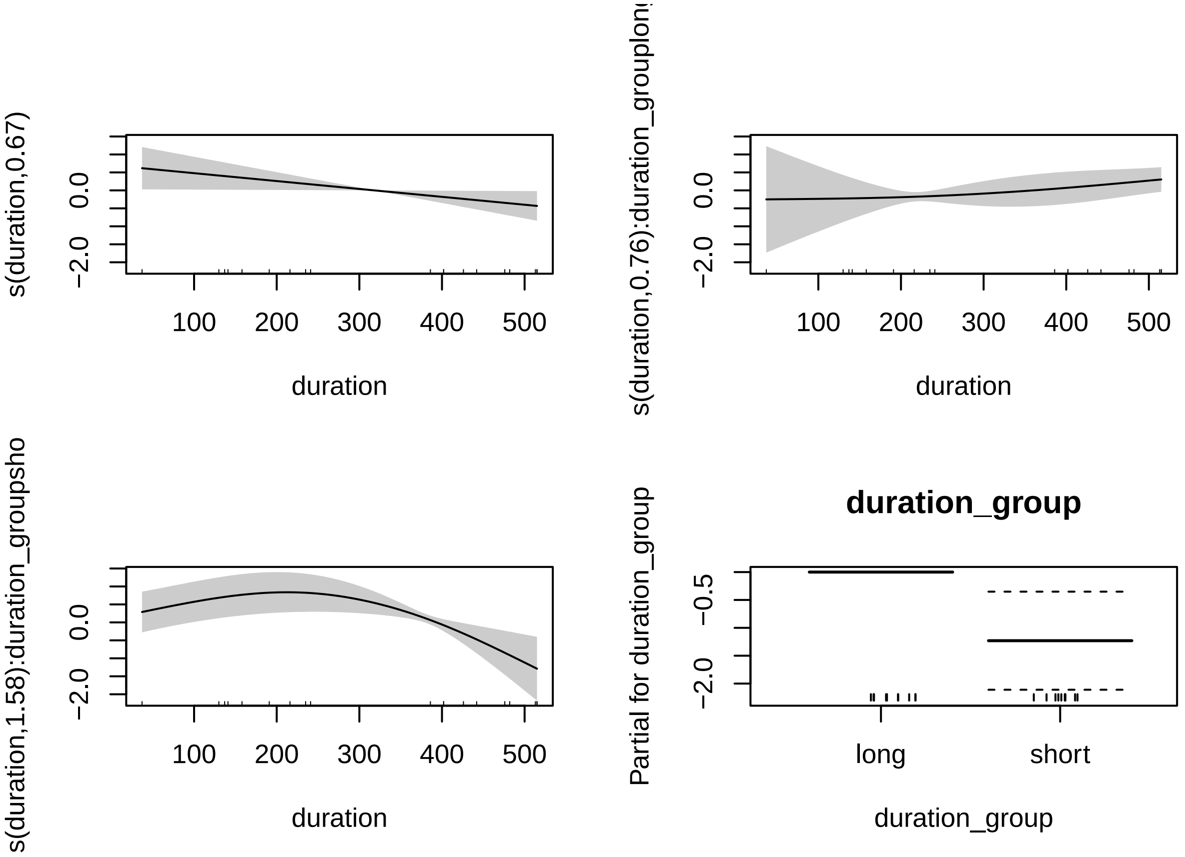
And we can also increase the number of observations by running the analysis at the segment level, rather than on mean stability.

Here is an exploratory analysis with a generalized additive model (GAM).

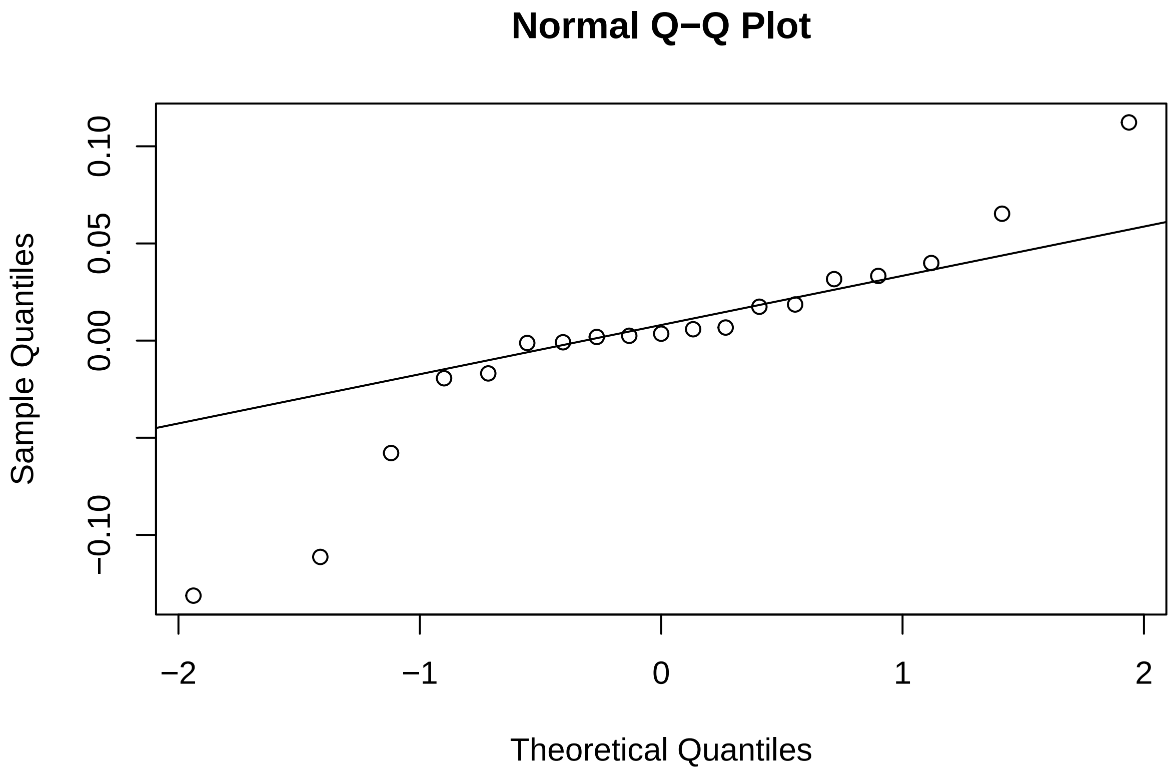
```
# Factorize duration_group
creole_stability$duration_group <- as.factor(creole_stability$duration_group)

# Model with an interaction between duration_group and duration
# (with maximum of cubic-spline fit)
msd.gam <- gam(MeanStability ~ duration_group + s(duration, k = 3) +
                 s(duration, by = duration_group, k = 3), data = creole_stability)

summary(msd.gam)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## MeanStability ~ duration_group + s(duration, k = 3) + s(duration,
##     by = duration_group, k = 3)
##
## Parametric coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           1.0073     0.2965   3.397  0.00434 **
## duration_groupshort  -1.2305     0.4398  -2.798  0.01424 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                   edf Ref.df     F p-value
## s(duration)                    0.6667 0.6667 6.605  0.0545 .
## s(duration):duration_grouplong  0.7592 0.8431 0.895  0.3997
## s(duration):duration_groupshort 1.5840 1.6598 8.549  0.0288 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Rank: 7/8
## R-sq.(adj) =  0.419   Deviance explained = 54.8%
## GCV = 0.0053365  Scale est. = 0.0039294  n = 19
```

```r
plot(msd.gam, all.terms = T, shade = T, pages = 1)
```



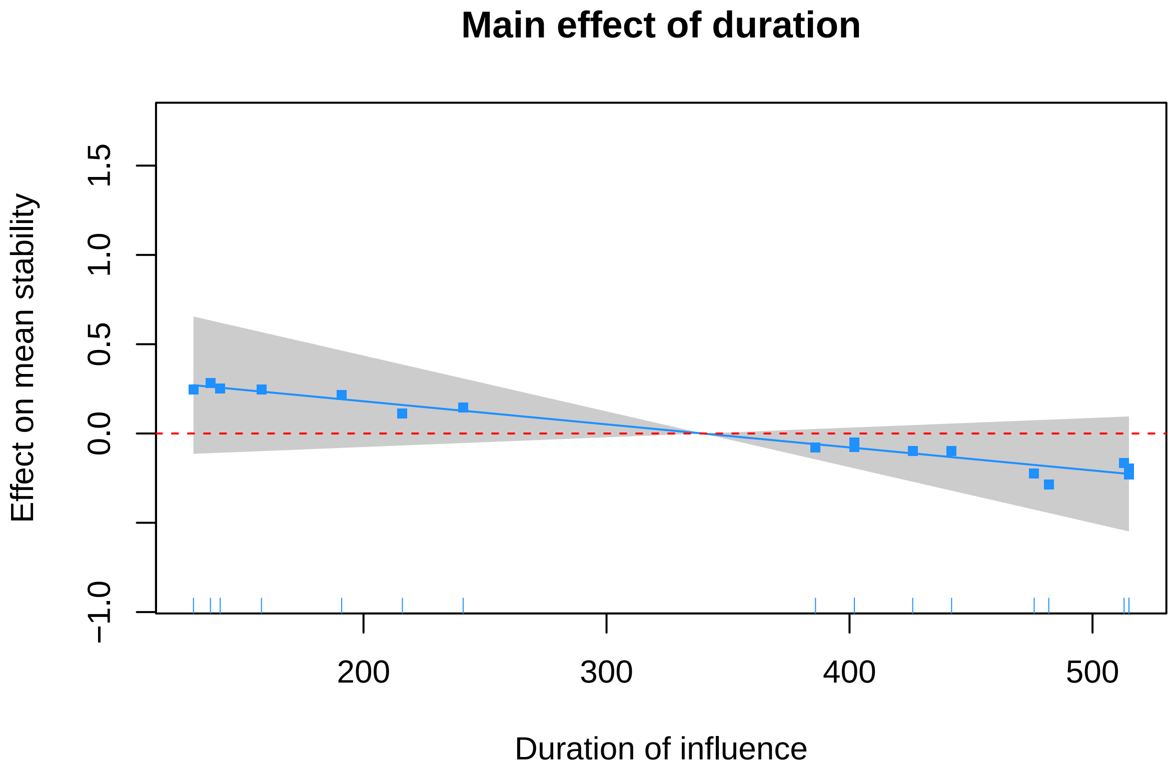
```r
qqnorm(resid(msd.gam))
qqline(resid(msd.gam))
```



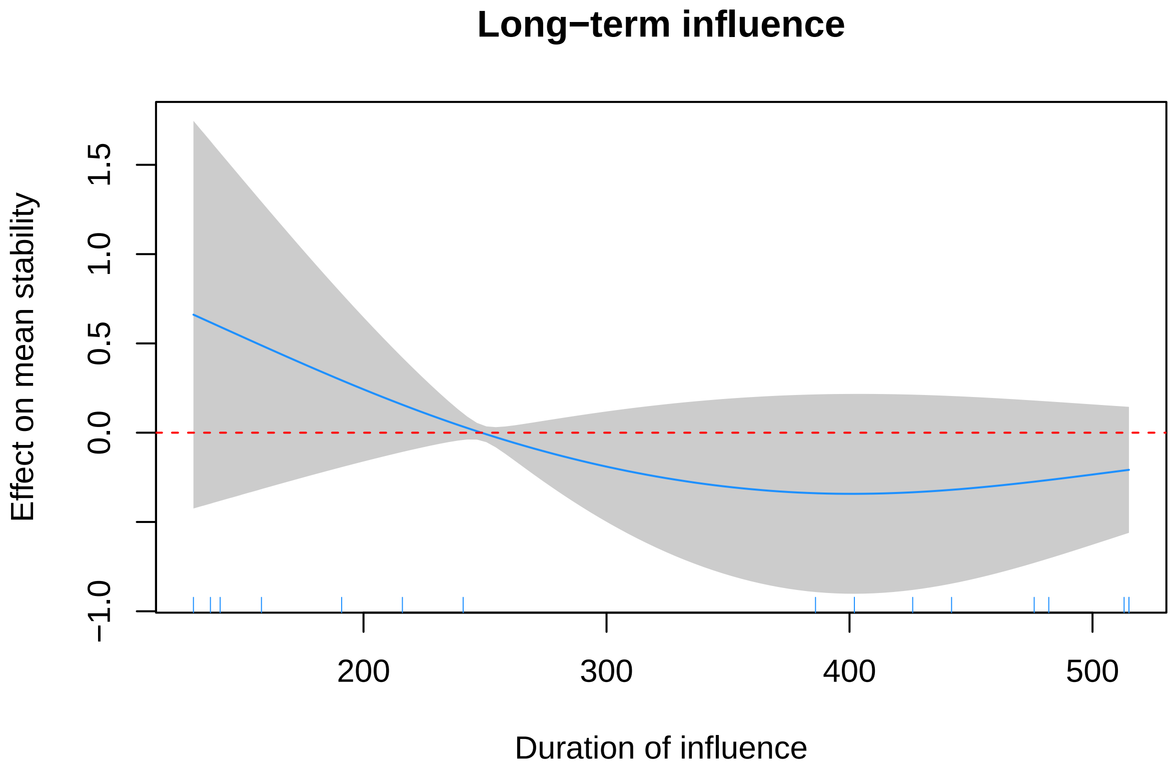
```
msd.gam.trimmed <- gam(MeanStability ~ duration_group + s(duration, k = 3)
                       + s(duration, by = duration_group, k = 3),
                       data = creole_stability %>% filter(MeanStability > 0.7))

summary(msd.gam.trimmed)
```

```
## 
## Family: gaussian 
## Link function: identity 
## 
## Formula:
## MeanStability ~ duration_group + s(duration, k = 3) + s(duration, 
##     by = duration_group, k = 3)
## 
## Parametric coefficients:
##                     Estimate Std. Error t value Pr(>|t|)   
## (Intercept)           1.3221     0.3396   3.893  0.00202 **
## duration_groupshort  -0.5035     0.3470  -1.451  0.17162   
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Approximate significance of smooth terms:
##                                    edf Ref.df     F p-value
## s(duration)                     0.6667 0.6667 2.975   0.184
## s(duration):duration_grouplong  1.2683 1.5080 1.101   0.374
## s(duration):duration_groupshort 0.6667 0.6667 1.379   0.357
## 
## Rank: 7/8
## R-sq.(adj) =  -0.0156   Deviance explained = 21.3%
## GCV = 0.0026593  Scale est. = 0.0019395  n = 17
```

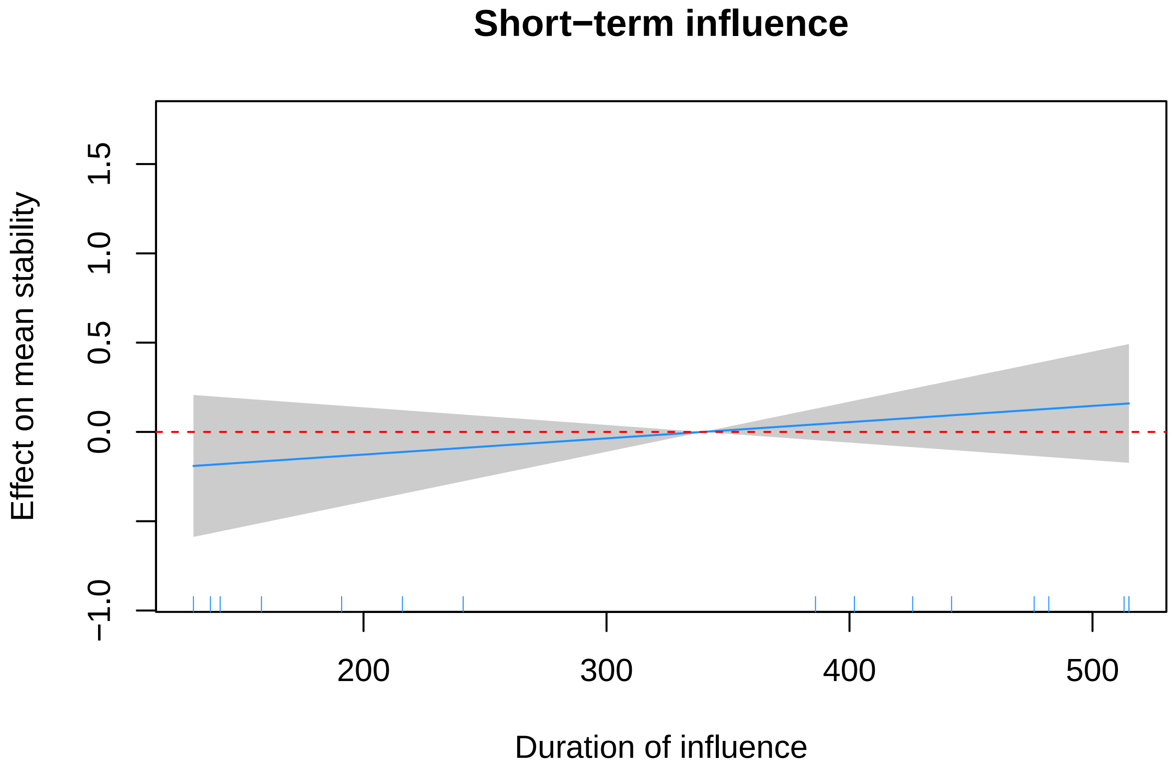
```r
plot(msd.gam.trimmed, sel = 1, shade = T, ylab = "Effect on mean stability",
     xlab = "Duration of influence", residuals = T, main = "Main effect of duration",
     cex = 5, pch = ".", col = "dodgerblue")
abline(h = 0, lty = 2, col = "red")
```



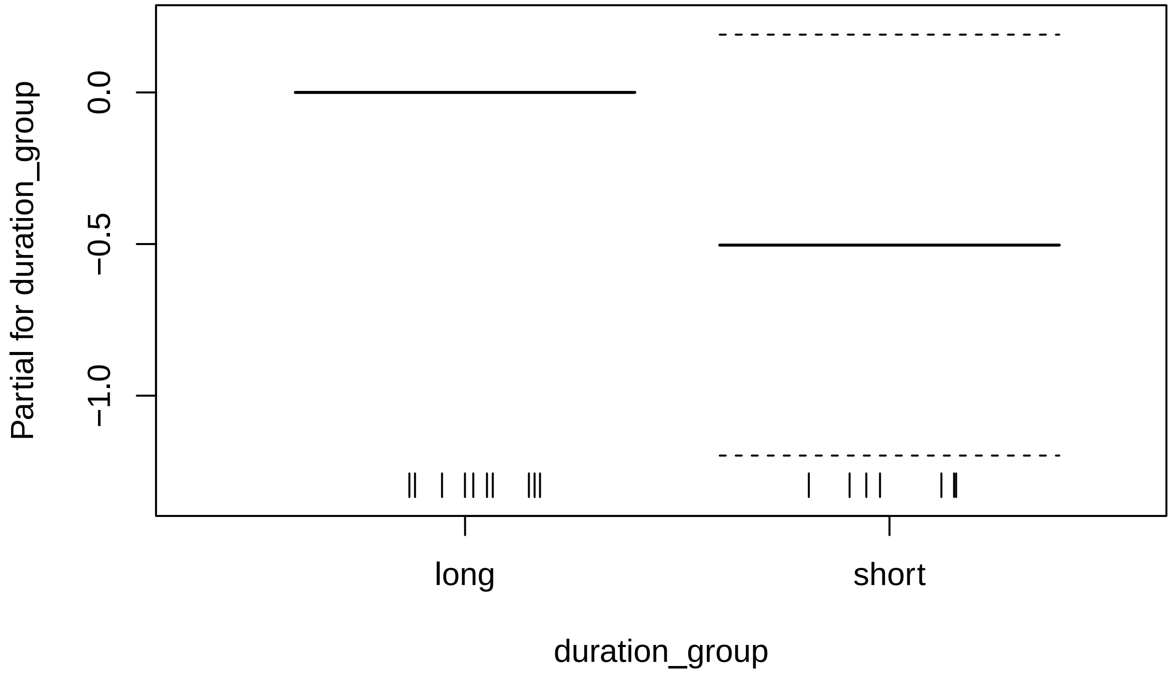
```r
plot(msd.gam.trimmed, sel = 2, shade = T, ylab = "Effect on mean stability",
     xlab = "Duration of influence", main = "Long-term influence", col = "dodgerblue")
abline(h = 0, lty = 2, col = "red")
```



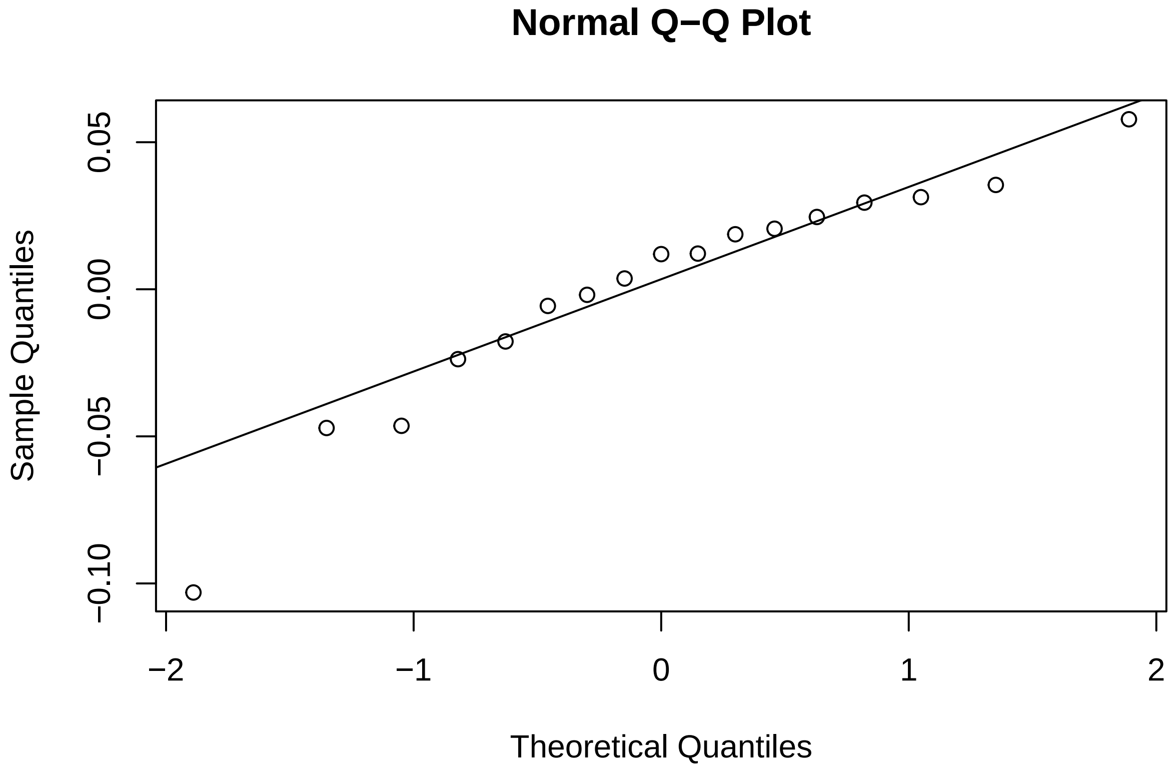
```
plot(msd.gam.trimmed, sel = 3, shade = T, ylab = "Effect on mean stability",
     xlab = "Duration of influence", main = "Short-term influence", col = "dodgerblue")
abline(h = 0, lty = 2, col = "red")
```



```r
# (dotted lines indicate error)
plot(msd.gam.trimmed, all.terms = T, sel = 4, ylab = "Effect on mean stability",
     xlab = "Duration group", main = "Main effect of duration group")
```



```r
# checking out the model performance
qqnorm(resid(msd.gam.trimmed))
qqline(resid(msd.gam.trimmed)) # meh
```

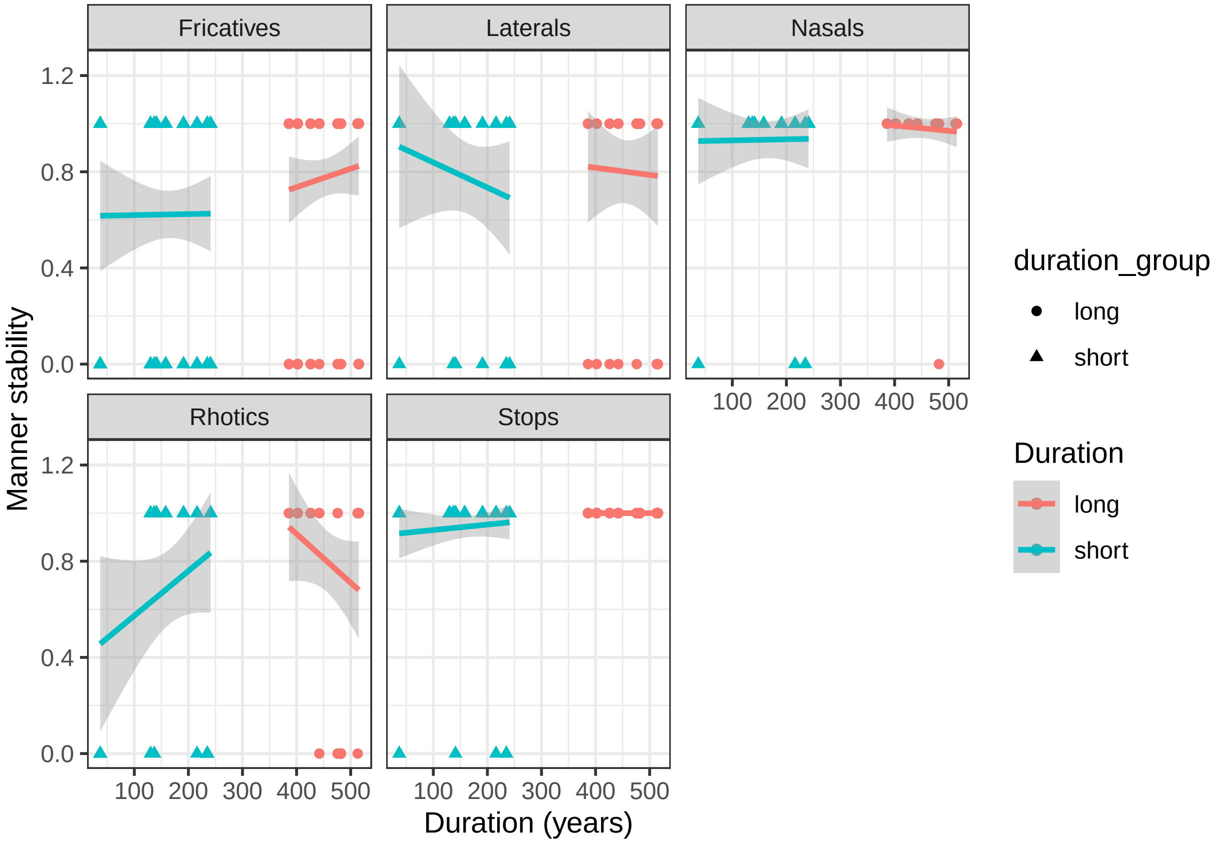


Removing the two creoles with the lowest scores produces significant effects. This does not seem very reliable though, especially given the small sample size. Also, the pattern is strange: a negative trend of duration for long-term influence and a positive one for short-term influence? Note that the model detected a mean difference between duration groups, with the short group having (slightly) lower mean stability. This appears to be the case – but again – there are very few observations.

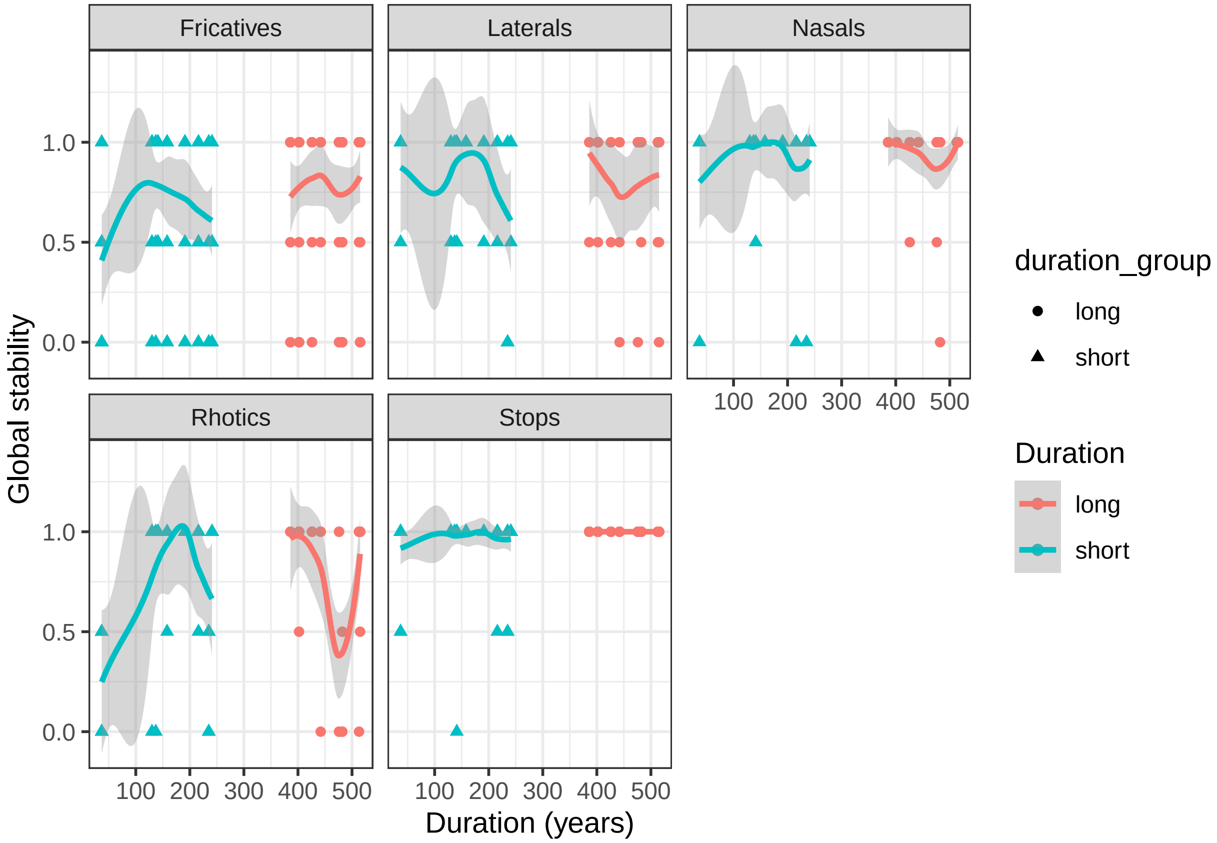
3.3 Duration effects on the segment level

Does duration affect the stability values of specific segments or segment classes?

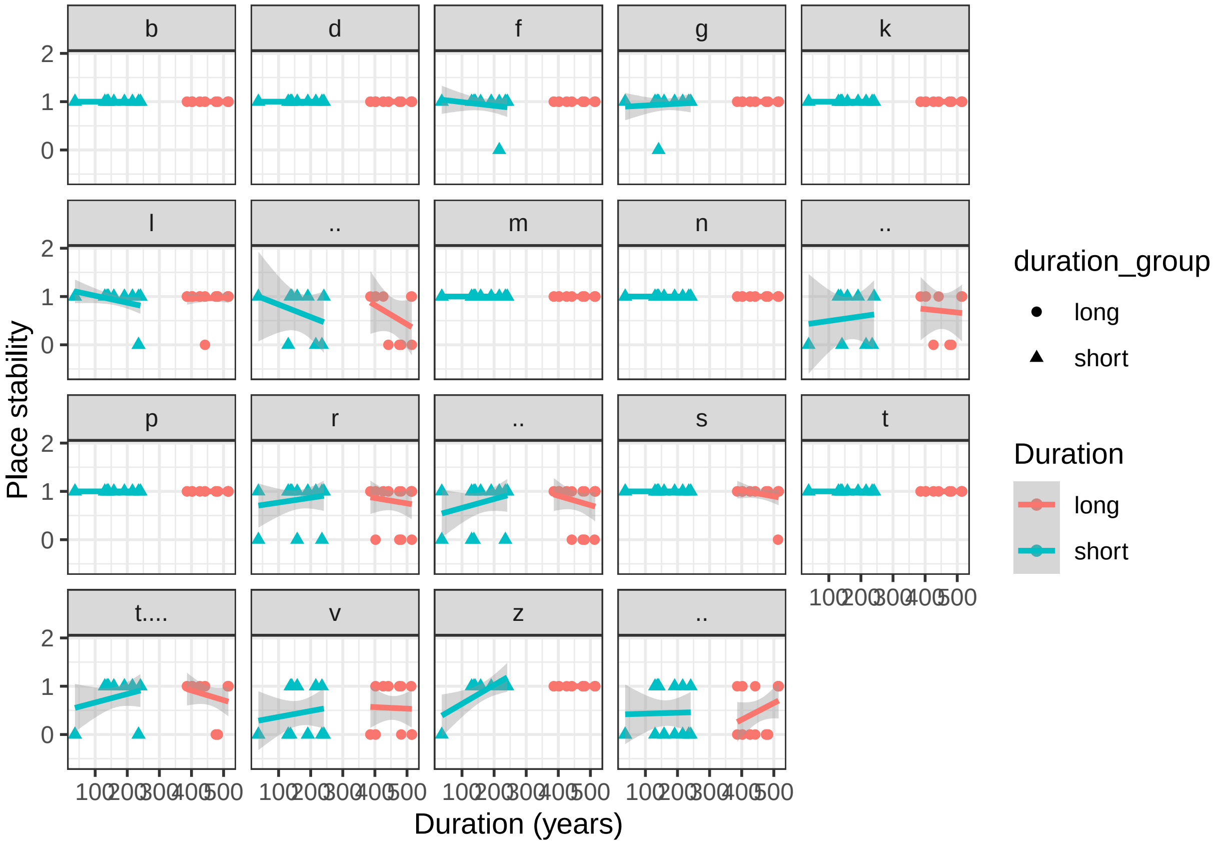
```
ggplot(database, aes(duration, MannerStability, colour = duration_group, shape = duration_group)) +
  geom_point() +
  geom_smooth(method = "lm") +
  facet_wrap(~Class) +
  xlab("Duration (years)") +
  ylab("Manner stability") +
  labs(color = "Duration")
```



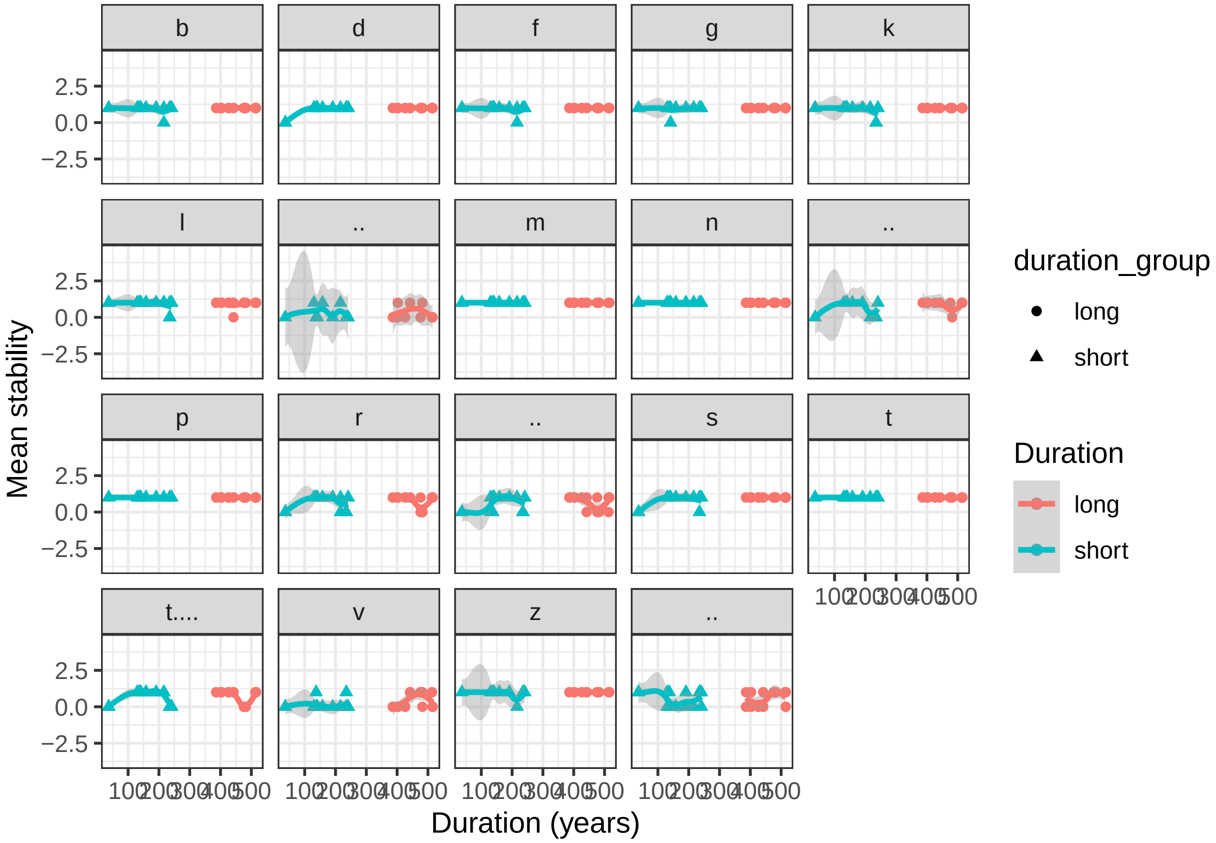
```
ggplot(database, aes(duration, GlobalStability, colour = duration_group, shape = duration_group)) +
  geom_point() +
  geom_smooth() +
  facet_wrap(~Class) +
  xlab("Duration (years)") +
  ylab("Global stability") +
  labs(color = "Duration")
```



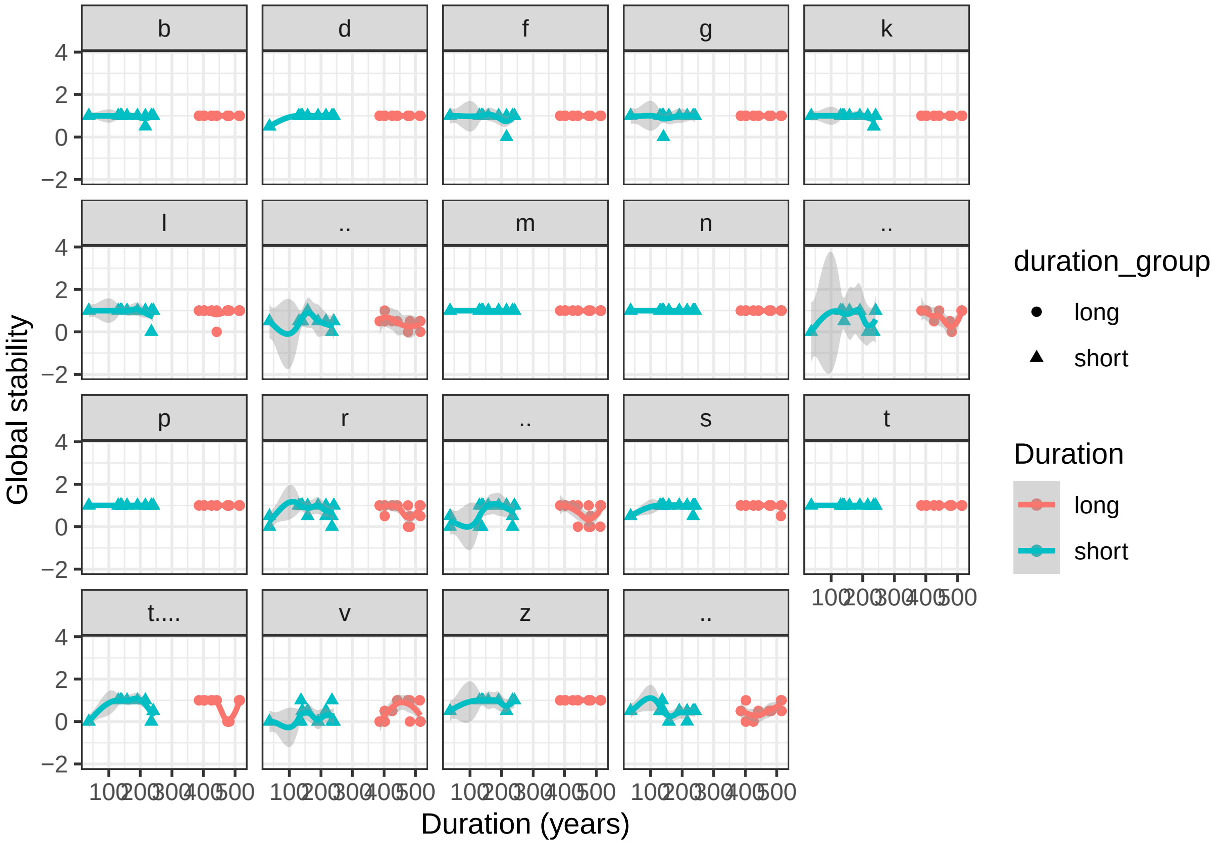
```r
ggplot(database, aes(duration, PlaceStability, colour = duration_group, shape = duration_group)) +
  geom_point() +
  geom_smooth(method = "lm") +
  facet_wrap(~LexifierPhoneme) +
  xlab("Duration (years)") +
  ylab("Place stability") +
  labs(color = "Duration")
```



```
ggplot(database, aes(duration, MannerStability, colour = duration_group, shape = duration_group)) +
  geom_point() +
  geom_smooth() +
  facet_wrap(~LexifierPhoneme) +
  xlab("Duration (years)") +
  ylab("Mean stability") +
  labs(color = "Duration")
```



```
ggplot(database, aes(duration, GlobalStability, colour = duration_group, shape = duration_group)) +
  geom_point() +
  geom_smooth() +
  facet_wrap(~LexifierPhoneme) +
  xlab("Duration (years)") +
  ylab("Global stability") +
  labs(color = "Duration")
```



3.4 Jaccard distance between inventories

Here we explore the Jaccard distance between phonological inventories.

```
df_jac <- read_csv("Inventories.csv")
#df_jac <- df_jac %>% subset(Category != 'creole')
df_jac <- df_jac %>% dplyr::select(c('Language','Phoneme'))
df_jac$presence <- 1
df_wide <- df_jac %>% spread(Phoneme, presence)
df_wide <- df_wide %>% replace(is.na(.), 0)
head(df_wide)
```

```
## # A tibble: 6 x 116
##   Language         b   `b`     b     b   `b `     c     ç    cç    cç     c     d
##   <chr>        <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Angolar          1     0     0     0     0     0     0     0     0     0     1
## 2 Cantonese        0     0     0     0     0     0     0     0     0     0     0
## 3 Cape Verdea~     1     0     0     0     0     0     0     0     0     0     1
## 4 Cape Verdea~     1     0     0     0     0     0     0     0     0     0     1
## 5 Cape Verdea~     1     0     0     0     0     0     0     0     0     0     1
## 6 Cape Verdea~     1     0     0     0     0     0     0     0     0     0     1
## # i 104 more variables: `d` <dbl>, `d` <dbl>, ð <dbl>, d  <dbl>, `d ` <dbl>,
## #   dz <dbl>, `dz` <dbl>, `d ` <dbl>, `d  ` <dbl>,   <dbl>, ` ` <dbl>,
## #     <dbl>, ` ` <dbl>, f <dbl>, g <dbl>, g  <dbl>,   <dbl>, b <dbl>,
## #     <dbl>,    <dbl>, ` ` <dbl>,    <dbl>, h <dbl>,    <dbl>, j <dbl>,
## #    <dbl>,    <dbl>,    <dbl>,    <dbl>, ` ` <dbl>, k <dbl>, k  <dbl>,
## #   kp <dbl>, k  <dbl>, k   <dbl>, l <dbl>, `l` <dbl>, l  <dbl>,   <dbl>,
## #    <dbl>, m <dbl>, `m` <dbl>, m  <dbl>, mb <dbl>, mp  <dbl>,  f <dbl>, ...
```

```
tmp <- df_wide %>%
  column_to_rownames(var = "Language")
jac_dist <- jaccard(t(tmp))
```

The Jaccard distance values were manually extracted into a new table, so we could visualize those values according to the relevant language in contact (see file `jaccard_results.csv`). Then, we created a new table which summaries those results for creoles and joins their stability values, so we can assess if there is or if there is not a correlation between the Jaccard distances and the overall stability os creoles.

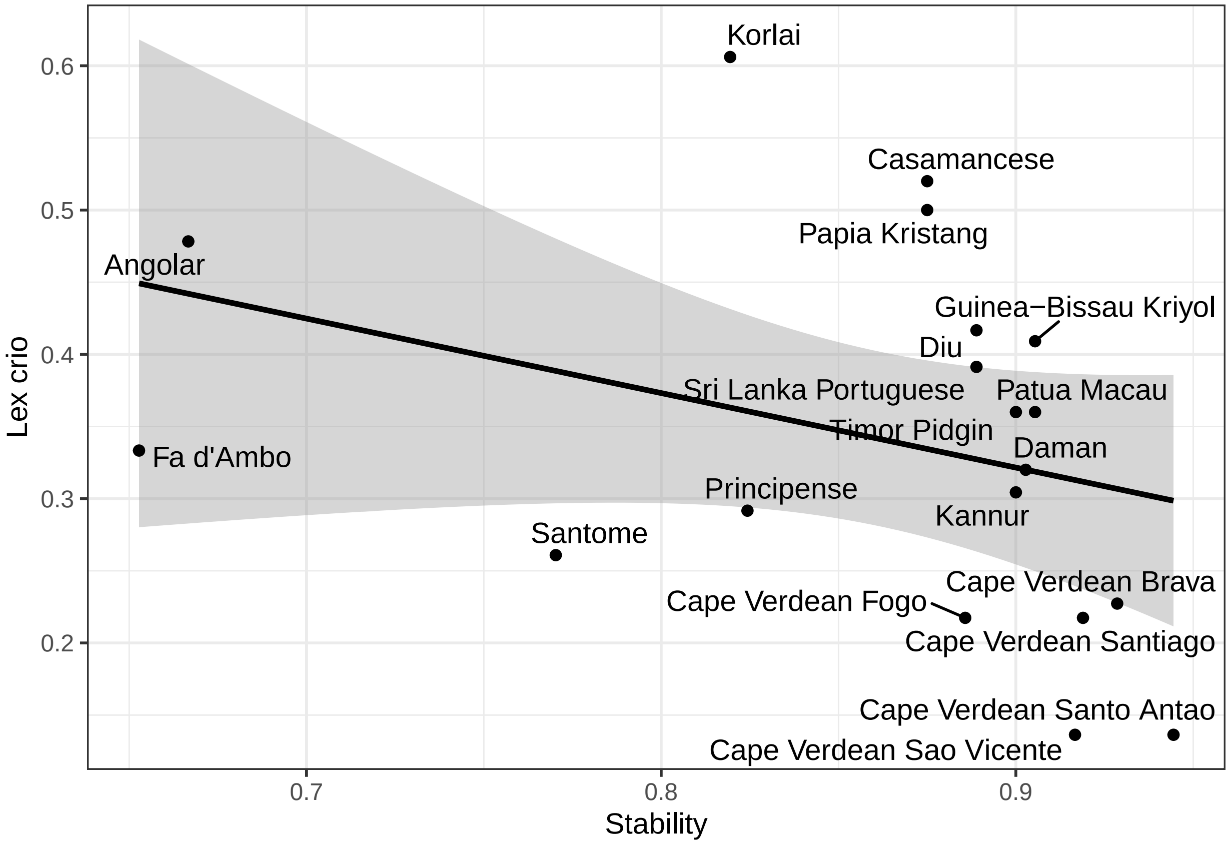
```
df_cor <- read.csv("jaccard_summary.csv")
```

We create a linear model of inventory distance creoles ~ Portuguese.

```
cl <- lm(stability ~ lex_crio, data=df_cor)
summary(cl)
```

```
## 
## Call:
## lm(formula = stability ~ lex_crio, data = df_cor)
## 
## Residuals:
##       Min        1Q    Median        3Q       Max 
## -0.210456  0.004963  0.036732  0.043100  0.058147 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)   0.9335     0.0532  17.546 2.51e-12 ***
## lex_crio     -0.2109     0.1463  -1.441    0.168    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.07987 on 17 degrees of freedom
## Multiple R-squared:  0.1089, Adjusted R-squared:  0.05647 
## F-statistic: 2.077 on 1 and 17 DF,  p-value: 0.1677
```

```
ggplot(df_cor, aes(x = stability, y = lex_crio, label = Language)) +
  geom_smooth(method = "lm", colour="black") +
  geom_point() +
  geom_text_repel(aes(label = Language)) +
  xlab("Stability") +
  ylab("Lex crio") # CS: FIX THIS
```

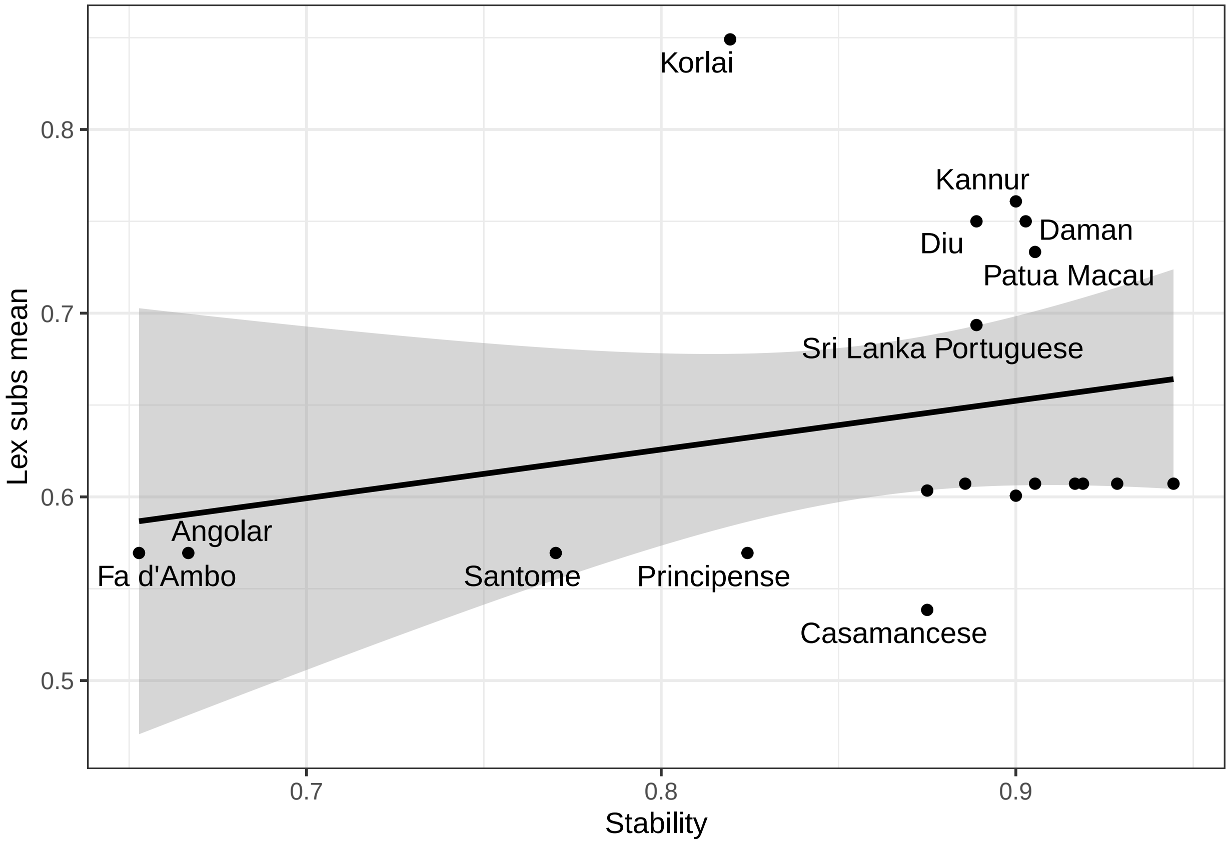


Next, inventory distance substrates ~ Portuguese.

```
sl_mean <- lm(stability ~ lex_subs_mean, data=df_cor)
summary(sl_mean)
```

```
##
## Call:
## lm(formula = stability ~ lex_subs_mean, data = df_cor)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.191196 -0.009182  0.021829  0.050399  0.091351
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.7064     0.1450   4.871 0.000144 ***
## lex_subs_mean   0.2416     0.2240   1.079 0.295663
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08185 on 17 degrees of freedom
## Multiple R-squared:  0.06409,    Adjusted R-squared:  0.009041
## F-statistic: 1.164 on 1 and 17 DF,  p-value: 0.2957
```

```
ggplot(df_cor, aes(x = stability, y = lex_subs_mean, label = Language)) +
  geom_smooth(method = "lm", colour="black") +
  geom_point() +
  geom_text_repel(aes(label = Language)) +
  xlab("Stability") +
  ylab("Lex subs mean") # CS: FIX THIS
```

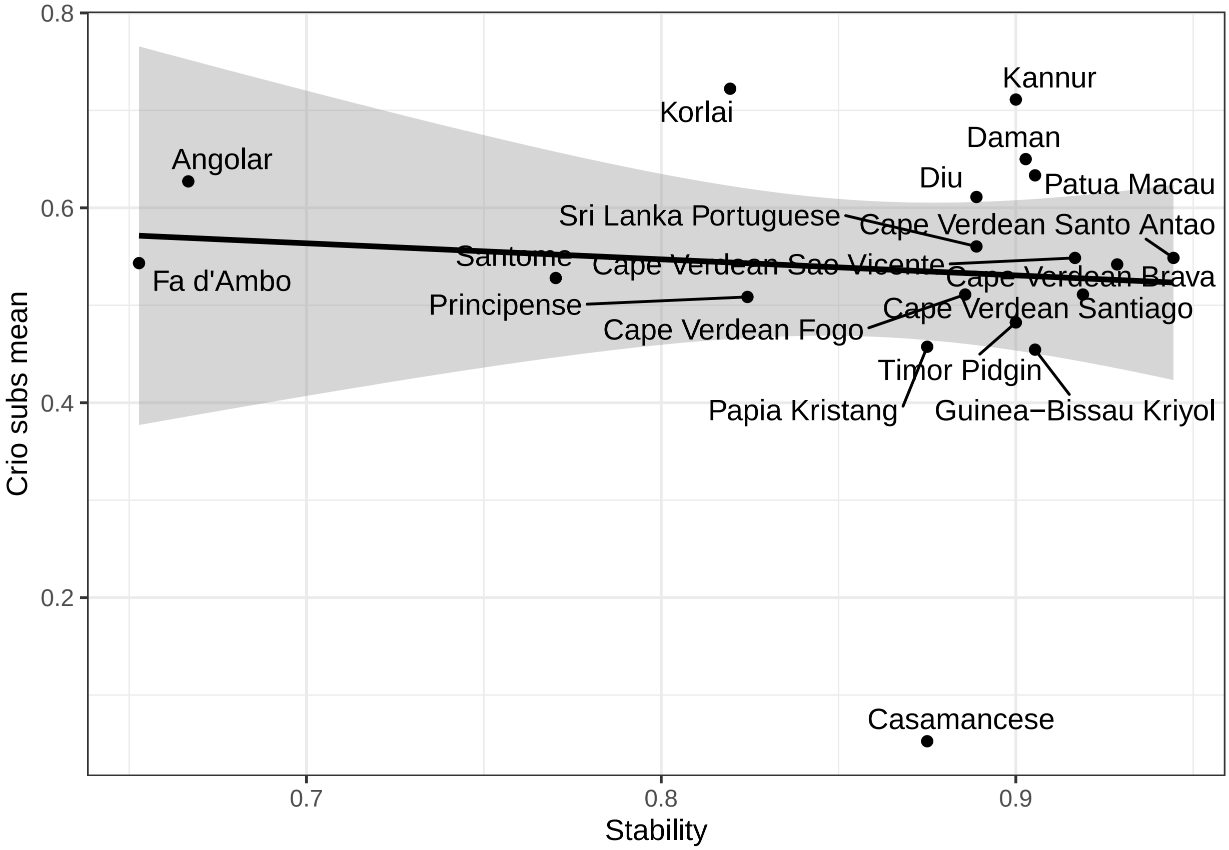


Next, inventory distance substrates ~ creoles.

```
sc_mean <- lm(stability ~ crio_subs_mean, data=df_cor)
summary(sc_mean)
```

```
##
## Call:
## lm(formula = stability ~ crio_subs_mean, data = df_cor)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.20840 -0.02278  0.03155  0.04882  0.08357
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.89192    0.07841   11.38 2.27e-09 ***
## crio_subs_mean -0.05658    0.14151   -0.40    0.694
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08421 on 17 degrees of freedom
## Multiple R-squared:  0.009317,   Adjusted R-squared:  -0.04896
## F-statistic: 0.1599 on 1 and 17 DF,  p-value: 0.6942
```

```
ggplot(df_cor, aes(x = stability, y = crio_subs_mean, label = Language)) +
  geom_smooth(method = "lm", colour="black") +
  geom_point() +
  geom_text_repel(aes(label = Language)) +
  xlab("Stability") +
  ylab("Crio subs mean") # CS: FIX THIS
```



4 Consonant stability

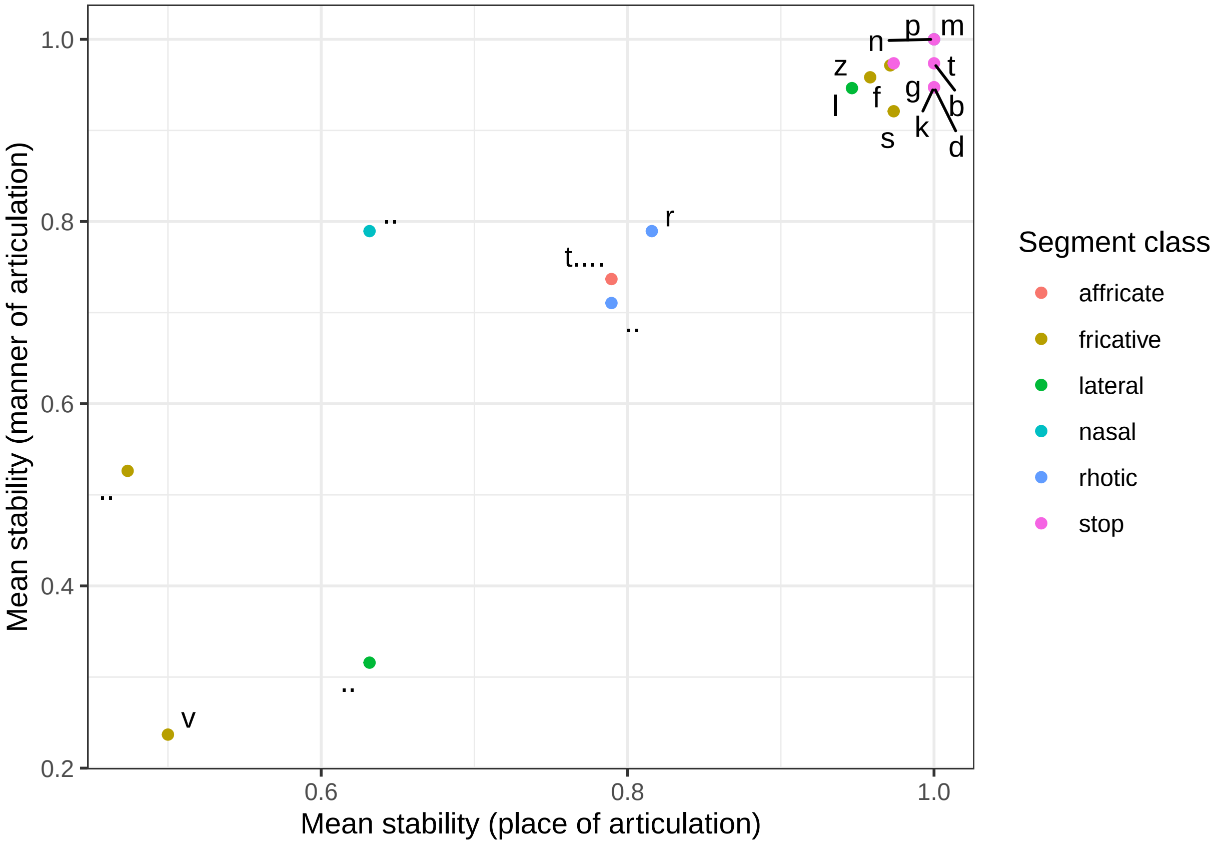
Which segments are the most stable across creoles in the language sample? We calculate stability of place and manner for each phoneme.

```
place_results <- database %>%
  group_by(LexifierPhoneme) %>%
  summarize(mplace = mean(PlaceStability, na.rm = TRUE))
manner_results <- database %>%
  group_by(LexifierPhoneme) %>%
  summarize(mmanner = mean(MannerStability, na.rm = TRUE))

consonant_stability <- left_join(place_results, manner_results, by = "LexifierPhoneme")

class <- c("stop", "stop", "fricative", "stop", "stop", "lateral",
           "nasal", "nasal", "stop", "rhotic","fricative", "stop",
           "affricate", "fricative", "fricative","nasal", "rhotic",
           "lateral", "fricative")
```

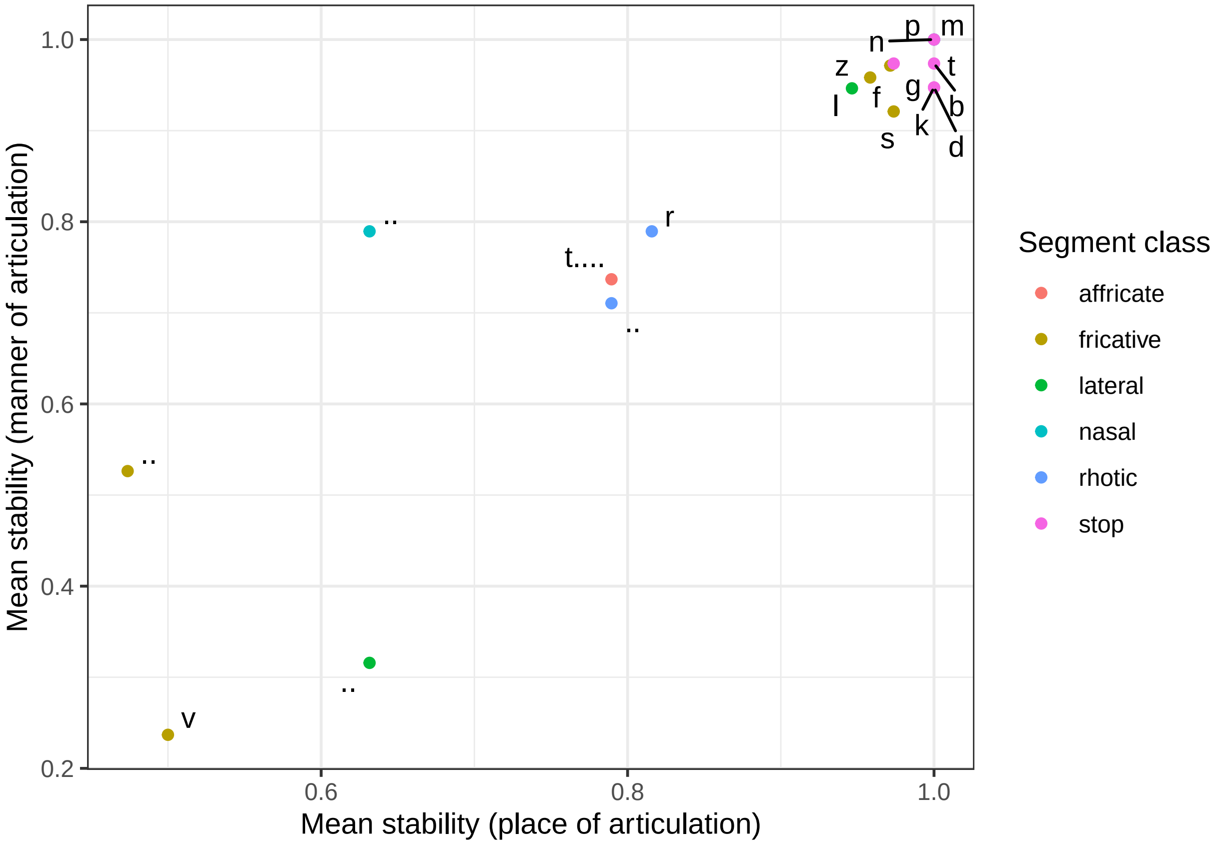
```
consonant_stability_class <- cbind(consonant_stability, class)
```

Next, we plot the results.

```
ggplot(consonant_stability, aes(y = mmanner, x = mplace)) +
  geom_point(position = "dodge", aes(color = class)) +
  geom_text_repel(aes(label = LexifierPhoneme), size = 4) +
  xlab("Mean stability (place of articulation)") +
  ylab("Mean stability (manner of articulation)") +
  labs(color = "Segment class")
```



```
ggplot(consonant_stability, aes(y = mmanner, x = mplace)) +
  geom_point(position = "dodge", aes(color = class)) +
  geom_text_repel(aes(label = LexifierPhoneme), size = 4) +
  xlab("Mean stability (place of articulation)") +
  ylab("Mean stability (manner of articulation)") +
  labs(color = "Segment class") +
  scale_fill_manual(values=cbPalette)
```

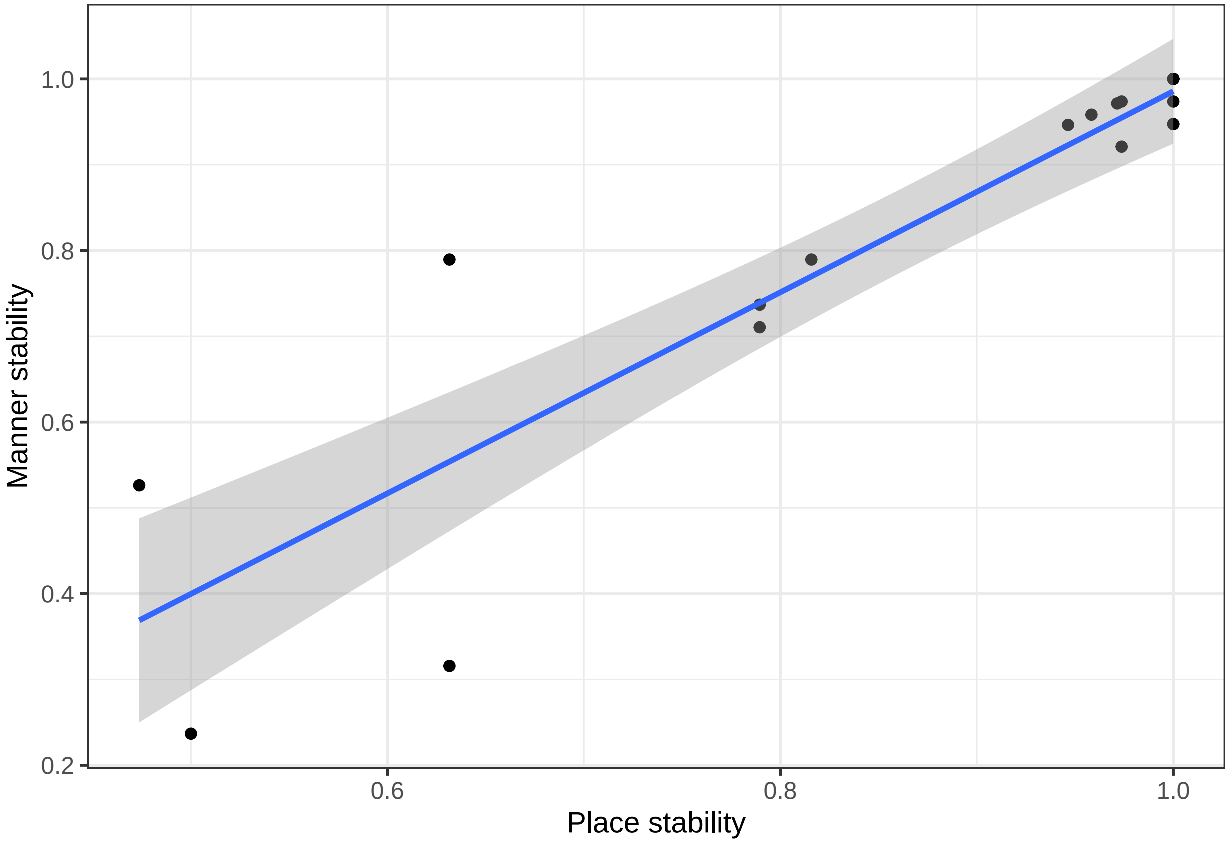


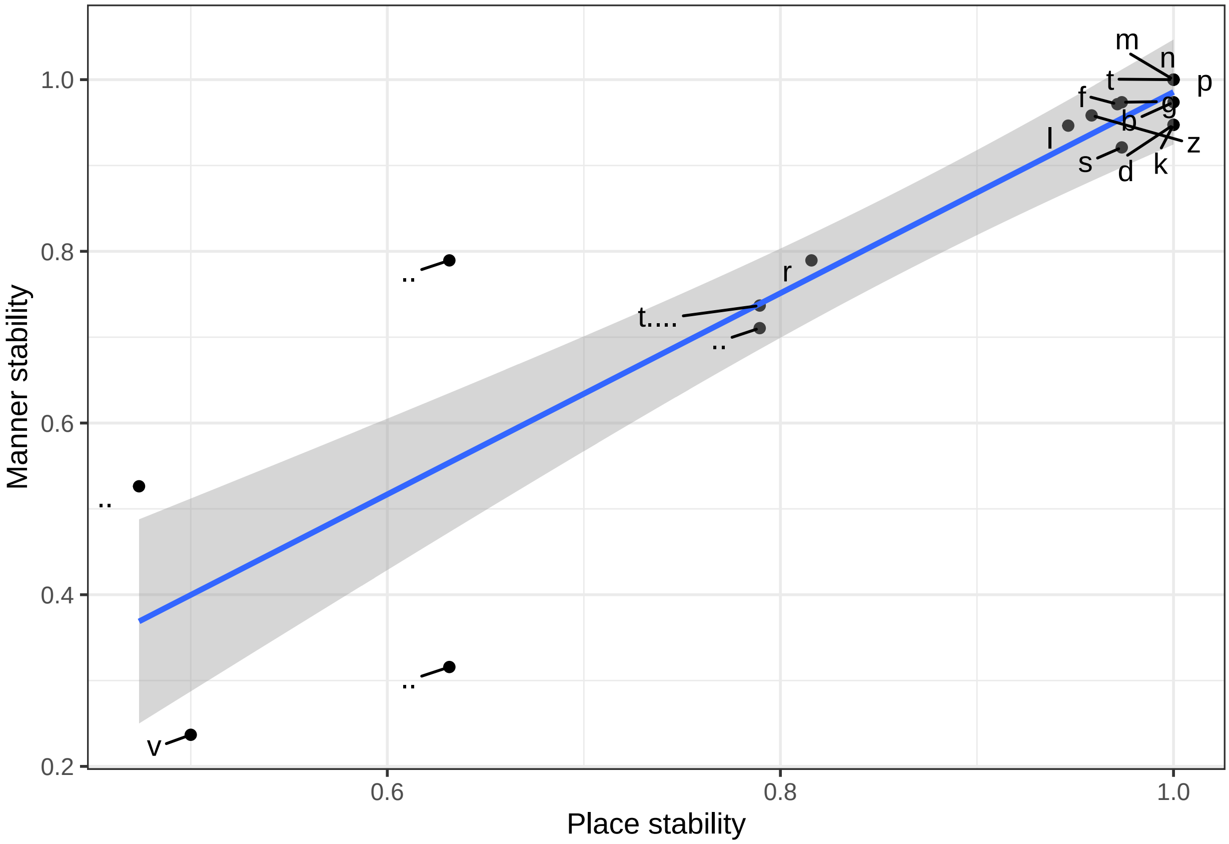
We make a linear model to assess the relationship between manner and place stability.

```
lm_manner_place <- lm(mplace~mmanner, data=consonant_stability)

summary(lm_manner_place)
```

```
##
## Call:
## lm(formula = mplace ~ mmanner, data = consonant_stability)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.206903 -0.000738  0.013372  0.037250  0.126426
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.28293    0.06764   4.183 0.000624 ***
## mmanner      0.70370    0.07873   8.938 7.81e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07783 on 17 degrees of freedom
## Multiple R-squared:  0.8246, Adjusted R-squared:  0.8142
## F-statistic:  79.9 on 1 and 17 DF,  p-value: 7.811e-08
```

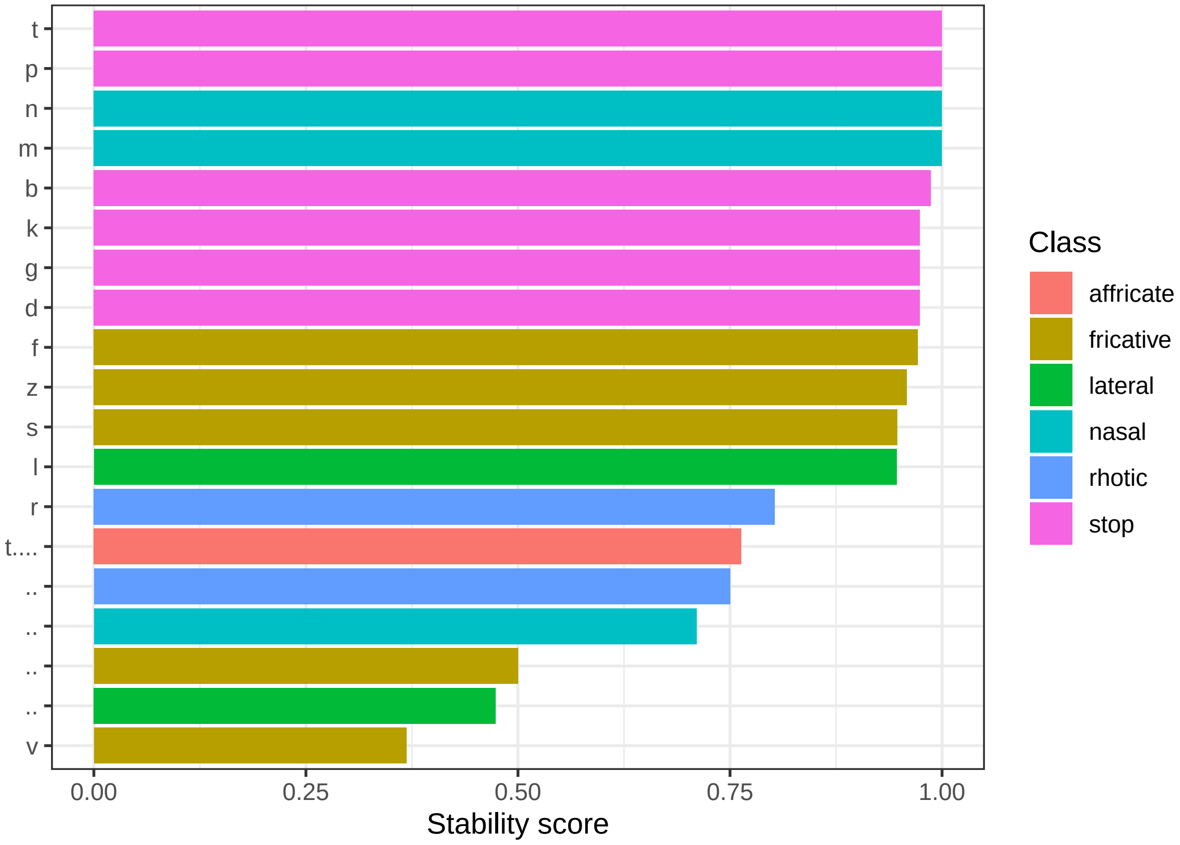
```
manner_place_lmplot <- ggplot(consonant_stability,
                              aes(y = mmanner, x = mplace,
                                  label = LexifierPhoneme)) +
  geom_point(position= "dodge") +
  geom_smooth(method = lm) #+
  #geom_text(aes(label=LexifierPhoneme), hjust=3, vjust=0)

print(manner_place_lmplot + labs(y = "Manner stability", x = "Place stability")) +
  geom_text_repel(aes(label=LexifierPhoneme), hjust=3, vjust=1)
```





Here is an alternative view for the global results.

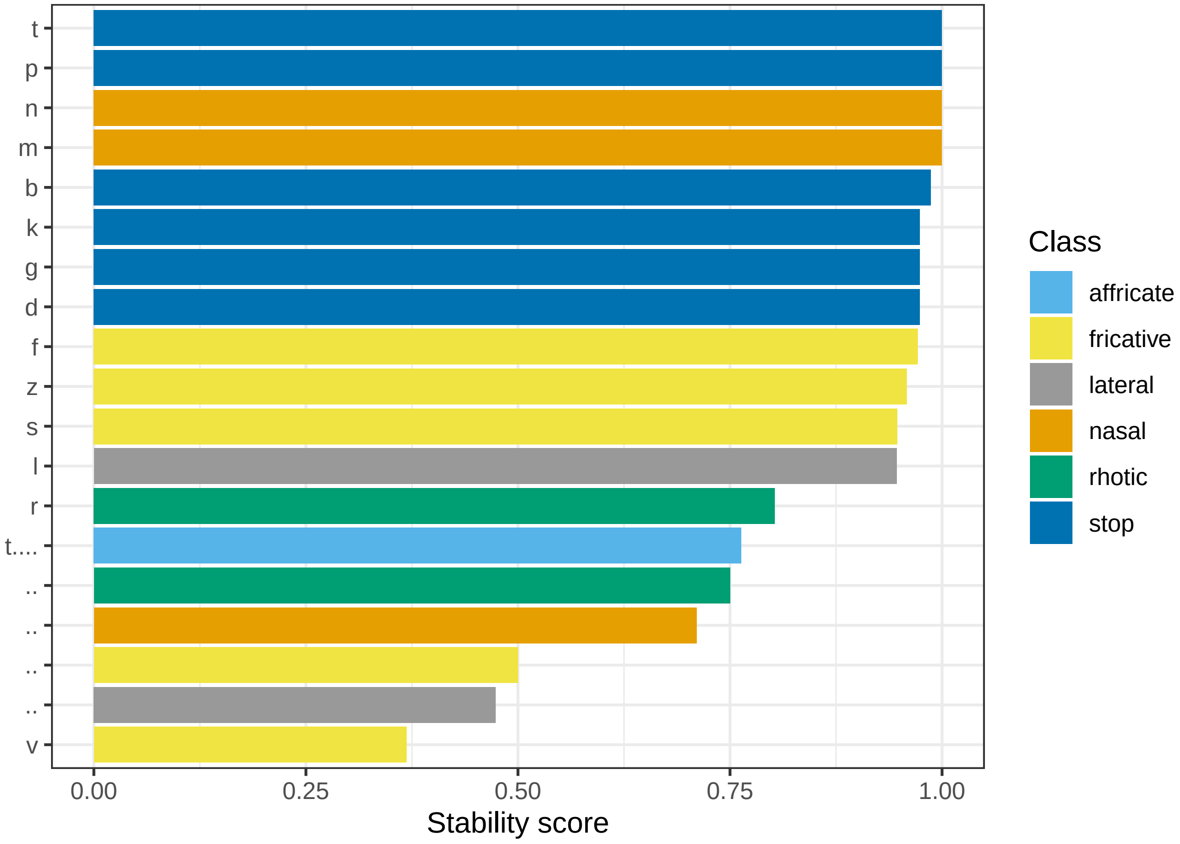
```r
consonant_global_stability <- mutate(consonant_stability_class,
                                     mglobal = (mmanner + mplace) / 2)

write.csv(consonant_global_stability, "consonant_global_stability.csv", row.names=FALSE)

ggplot(consonant_global_stability) +
  geom_bar(aes(
    x = mglobal,
    y = reorder(LexifierPhoneme, mglobal),
    fill = class
  ), stat = "identity", show.legend = TRUE) +
  labs(x = "Stability score", y="", fill = "Class")
```



```
ggplot(consonant_global_stability) +
  geom_bar(aes(
    x = mglobal,
    y = reorder(LexifierPhoneme, mglobal),
    fill = class
  ), stat = "identity", show.legend = TRUE) +
  labs(x = "Stability score", y="", fill = "Class") +
  scale_fill_manual(values=cbPalette)
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <ca>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <8e>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <ca>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <92>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <c9>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <b2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <c9>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
```

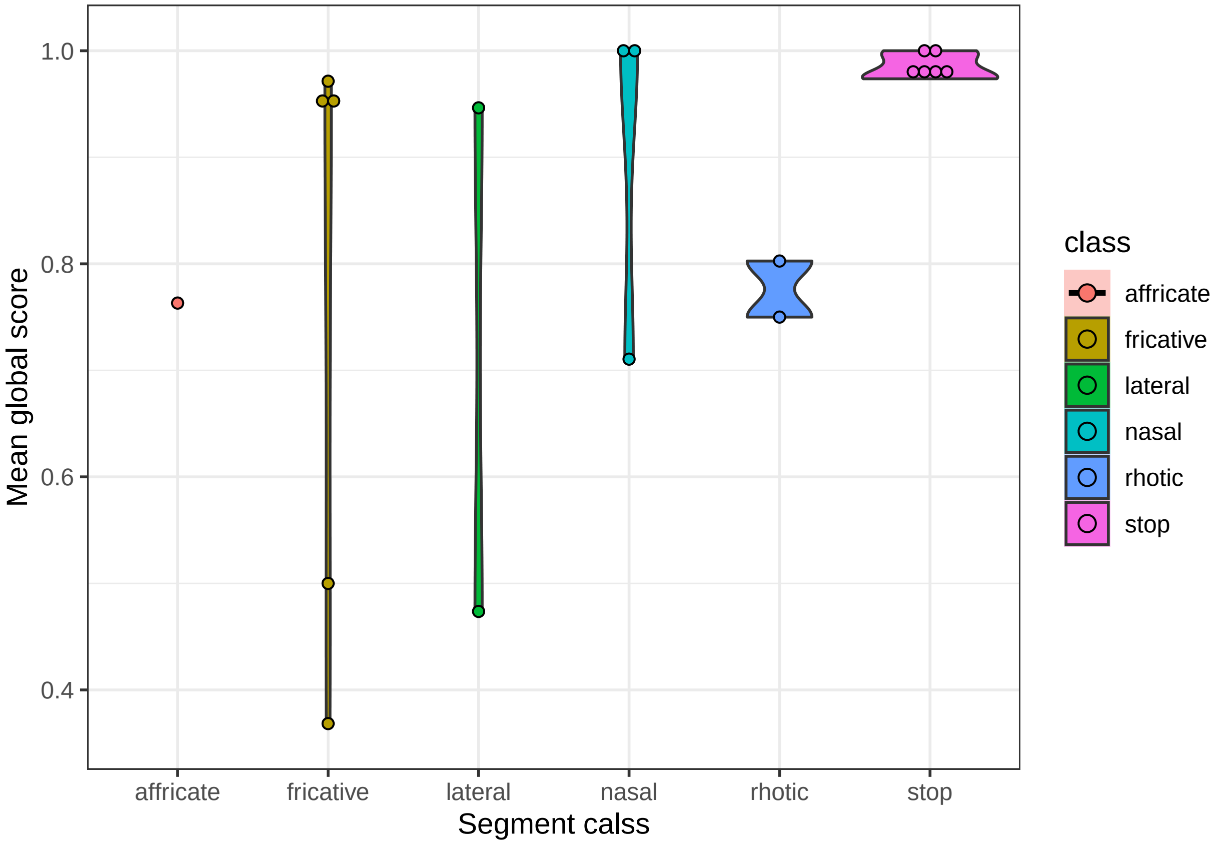
```
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <be>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 't ' in 'mbcsToSbcs': dot substituted for <cc>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 't ' in 'mbcsToSbcs': dot substituted for <a0>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 't ' in 'mbcsToSbcs': dot substituted for <ca>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 't ' in 'mbcsToSbcs': dot substituted for <83>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <ca>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <8e>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <ca>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <92>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <c9>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <b2>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <c9>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <be>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 't ' in 'mbcsToSbcs': dot substituted for <cc>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 't ' in 'mbcsToSbcs': dot substituted for <a0>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 't ' in 'mbcsToSbcs': dot substituted for <ca>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 't ' in 'mbcsToSbcs': dot substituted for <83>
```



4.1 Manner stability

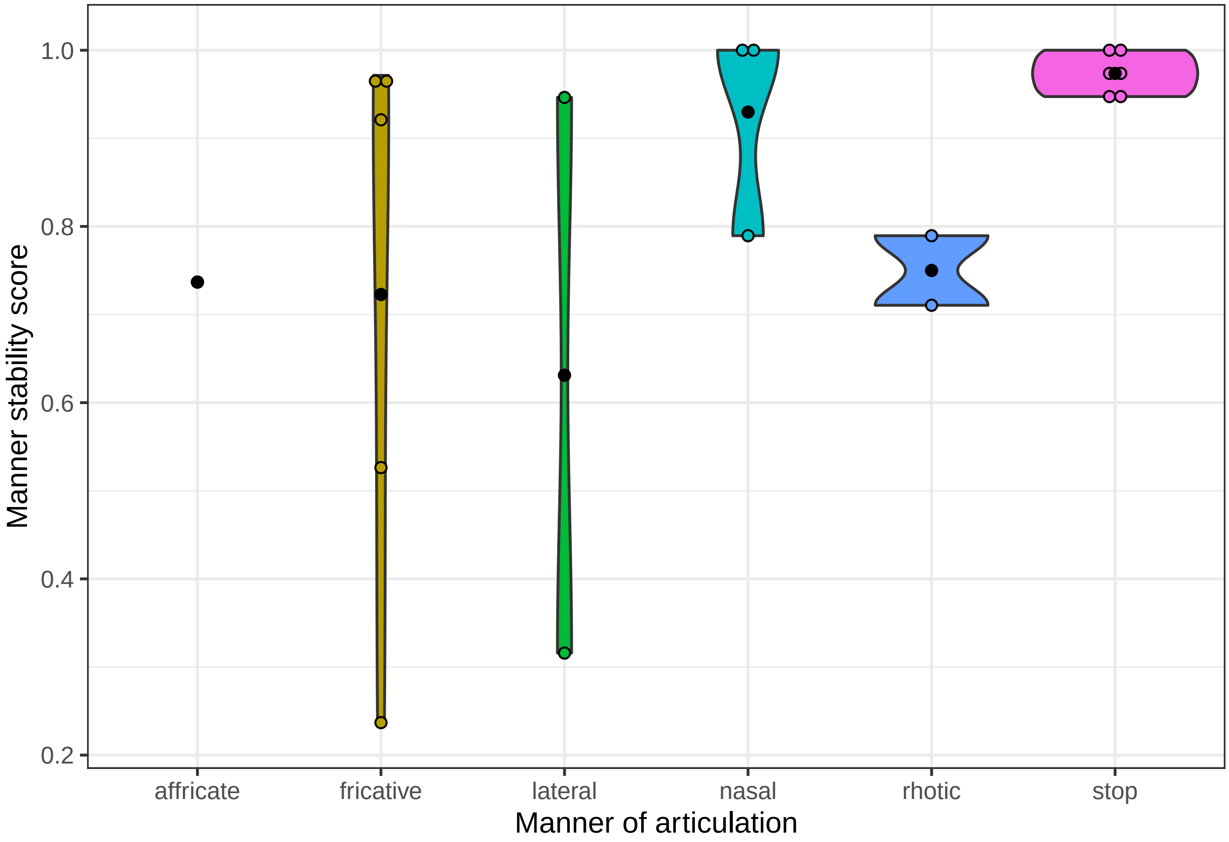
Check for class effects on the global stability of consonants.

```
ggplot(consonant_global_stability, aes(x = class, y = mglobal, fill = class)) +
  geom_smooth(method = "lm", colour="black") +
  geom_violin() +
   geom_dotplot(binaxis = "y",
               stackdir = "center",
               dotsize = 0.5) +
  xlab("Segment calss") +
  ylab("Mean global score") # CS: FIX THIS
```



Now, just plotting the relation manner to manner.

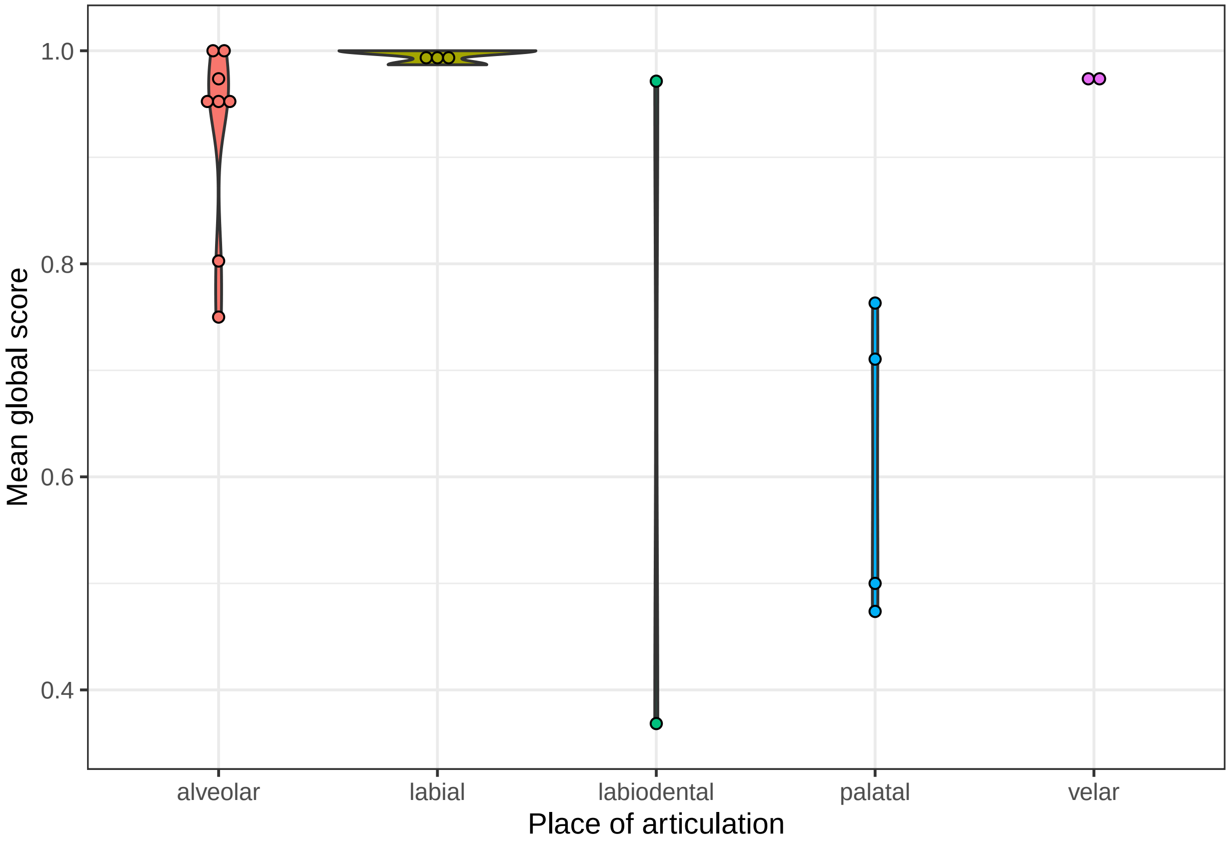
```
ggplot(consonant_global_stability, aes(x = class, y = mmanner, fill = class)) +
  geom_smooth(method = "lm", colour="black") +
  geom_violin() +
   geom_dotplot(binaxis = "y",
               stackdir = "center",
               dotsize = 0.5) +
  theme(legend.position="none") +
  ylab("Manner stability score") +
  xlab("Manner of articulation") +
 stat_summary(fun.y=mean, geom="point", size=2, shape=16)
```



4.2 Place stability

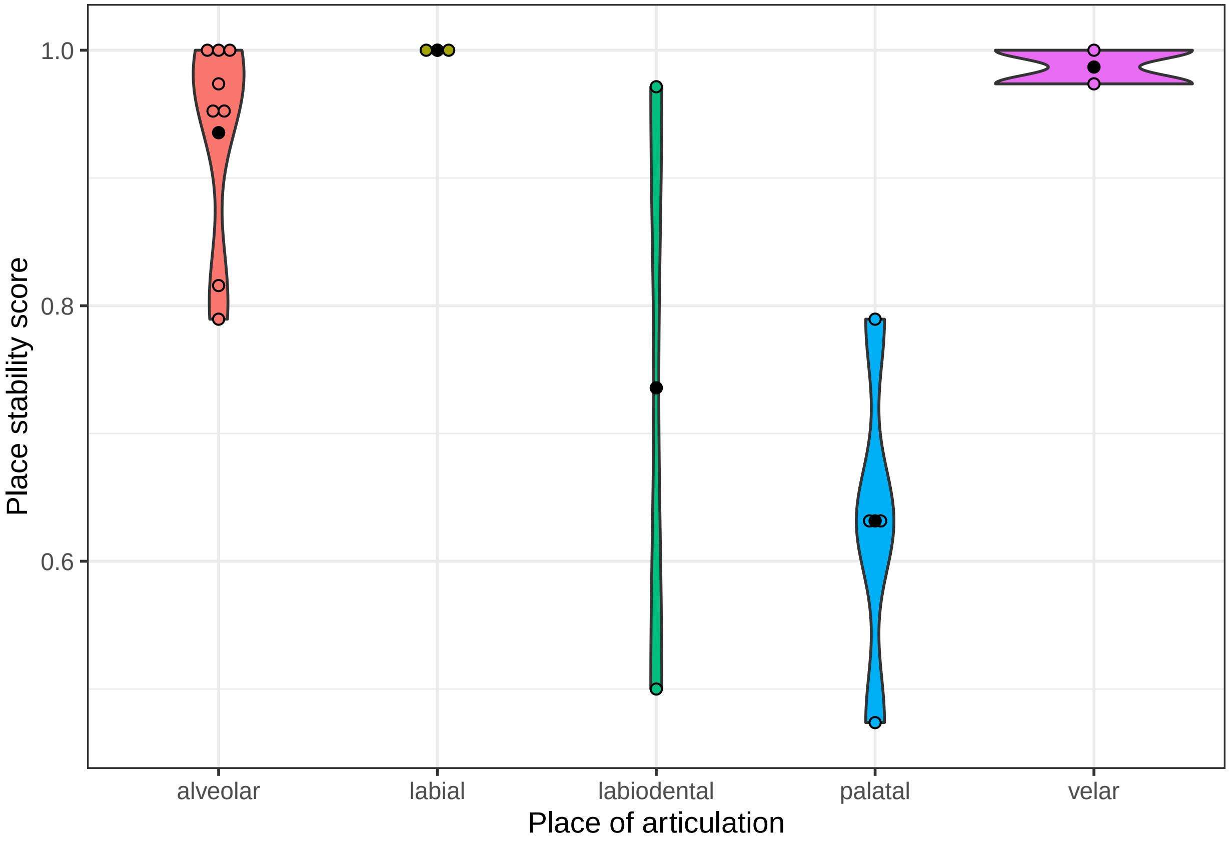
We check place effects on the global stability of the consonants.

```
place <- c("labial", "alveolar", "labiodental", "velar", "velar",
           "alveolar", "labial", "alveolar", "labial", "alveolar",
           "alveolar", "alveolar", "palatal", "labiodental", "alveolar",
           "palatal", "alveolar", "palatal", "palatal")

consonant_stability_place <- cbind(consonant_global_stability, place)

ggplot(consonant_stability_place, aes(x = place, y = mglobal, fill = place)) +
  geom_smooth(method = "lm", colour="black") +
  geom_violin() +
   geom_dotplot(binaxis = "y",
               stackdir = "center",
               dotsize = 0.5) +
  theme(legend.position="none") +
  xlab("Place of articulation") +
  ylab("Mean global score") # CS: FIX THIS
```



Now, just with the mean for place stability.

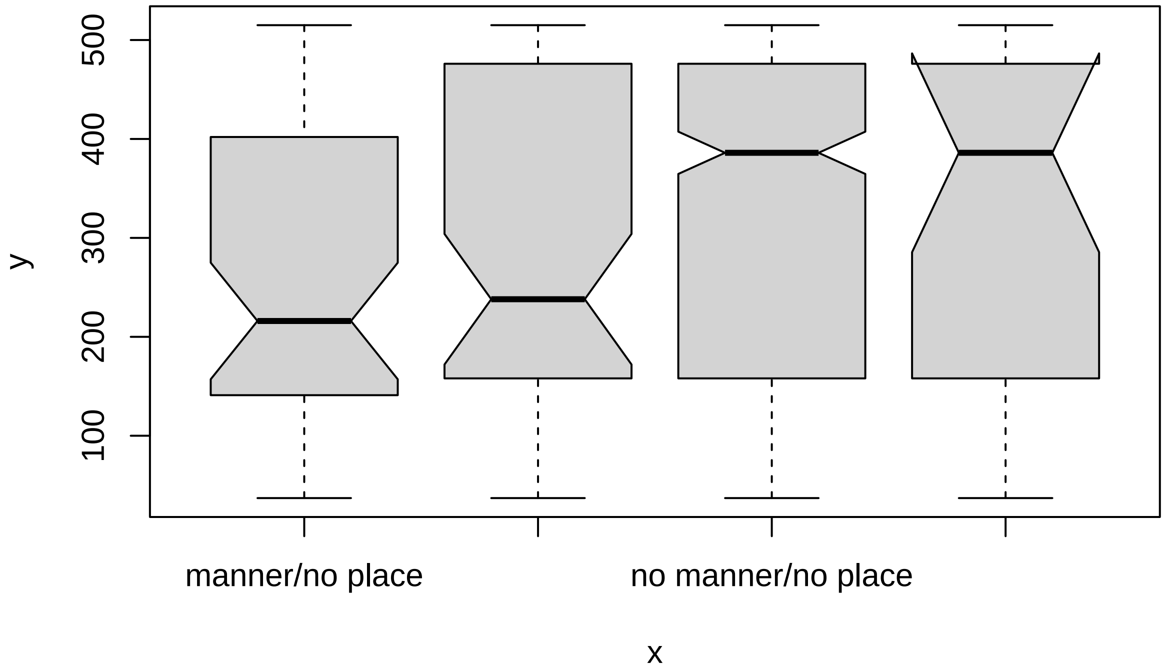
```
ggplot(consonant_stability_place, aes(x = place, y = mplace, fill = place)) +
  geom_smooth(method = "lm", colour="black") +
  geom_violin()  +
   geom_dotplot(binaxis = "y",
               stackdir = "center",
               dotsize = 0.5)  +
  theme(legend.position="none") +
  ylab("Place stability score") +
  xlab("Place of articulation") +
 stat_summary(fun.y=mean, geom="point", size=2, shape=16)
```



We can also increase the number of observations in duration regression by running the analysis at the segment level, rather than on mean stability.

```r
# Factorizing
mod.db <- database %>%
  as.data.frame() %>%
  mutate(
    categorical_stability = as.factor(categorical_stability),
    Lexifier = as.factor(Lexifier),
    CreolePhoneme = as.factor(CreolePhoneme),
    Language = as.factor(Language)
  )

# Remove singletons/doubletons
# goodies = names(table(mod.db$CreolePhoneme)>2)

# mod.db = mod.db %>%
#         filter(CreolePhoneme %in% goodies)
```

```r
plot(mod.db$categorical_stability, mod.db$duration, notch = T)
```



Hugely skewed in favor of no manner/place (10x as frequent as the next most frequent level; note this may cause problems for the models).

```
table(mod.db$categorical_stability)
```

```
##
##    manner/no place       manner/place no manner/no place    no manner/place
##                 49                 58                553                 25
```

```
# Place stability
cat.mod.place <- glmer(PlaceStability ~ log(duration) + (1 | CreolePhoneme),
                       data = mod.db, family = "binomial")
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.0989799 (tol = 0.002, component 1)

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model is nearly unider
##  - Rescale variables?
```

```
summary(cat.mod.place)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: PlaceStability ~ log(duration) + (1 | CreolePhoneme)
##    Data: mod.db
##
##      AIC      BIC   logLik deviance df.resid
##    305.4    319.0   -149.7    299.4      682
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -6.4584  0.0282  0.0300  0.1610  2.4148
##
## Random effects:
```

```
##  Groups        Name        Variance Std.Dev.
##  CreolePhoneme (Intercept) 40.56    6.369
## Number of obs: 685, groups:  CreolePhoneme, 34
##
## Fixed effects:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)   5.4884037  0.0007144  7682.9   <2e-16 ***
## log(duration) 0.0643798  0.0007145    90.1   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## log(duratn) 0.000
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 0.0989799 (tol = 0.002, component 1)
## Model is nearly unidentifiable: very large eigenvalue
##  - Rescale variables?
```

```
# Manner stability
cat.mod.manner <- glmer(MannerStability ~ log(duration) + (1 | CreolePhoneme),
                        data = mod.db, family = "binomial")

summary(cat.mod.manner)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: MannerStability ~ log(duration) + (1 | CreolePhoneme)
##    Data: mod.db
##
##      AIC      BIC   logLik deviance df.resid
##    255.8    269.4   -124.9    249.8      682
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -6.2314  0.0036  0.0044  0.1687  1.2034
##
## Random effects:
##  Groups        Name        Variance Std.Dev.
##  CreolePhoneme (Intercept) 405.7    20.14
## Number of obs: 685, groups:  CreolePhoneme, 34
##
## Fixed effects:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)     7.6512     2.5177   3.039  0.00237 **
## log(duration)   0.5459     0.2683   2.035  0.04185 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## log(duratn) -0.566
```

```r
# Duration group
cat.mod.group <- glmer(as.factor(duration_group) ~ PlaceStability +
                         MannerStability + (1 | CreolePhoneme),
                       data = mod.db, family = "binomial", nAGQ = 0)
```

```
## boundary (singular) fit: see help('isSingular')
```

```r
summary(cat.mod.group)
```

```
## Generalized linear mixed model fit by maximum likelihood (Adaptive
##   Gauss-Hermite Quadrature, nAGQ = 0) [glmerMod]
##  Family: binomial  ( logit )
## Formula: as.factor(duration_group) ~ PlaceStability + MannerStability +
##     (1 | CreolePhoneme)
##    Data: mod.db
##
##      AIC      BIC   logLik deviance df.resid
##    944.6    962.7   -468.3    936.6      681
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -1.342 -0.900 -0.900  1.111  1.276
##
## Random effects:
##  Groups        Name        Variance Std.Dev.
##  CreolePhoneme (Intercept) 0        0
## Number of obs: 685, groups:  CreolePhoneme, 34
##
## Fixed effects:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)       0.3120     0.2363   1.321  0.18665
## PlaceStability    0.2769     0.2897   0.956  0.33922
## MannerStability  -0.7995     0.2621  -3.050  0.00229 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) PlcStb
## PlaceStblty -0.543
## MannrStblty -0.326 -0.570
## optimizer (bobyqa) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

There is some indication that place stability is more often associated with shorter periods of influence.

Numerically, the manner/place category has 50% of its observations in the longest duration from the sample. At the same time, no manner/no place is associated with the shortest duration.

4.3 Word position

Next we ask, does word position influence stability?

First, data preparation.

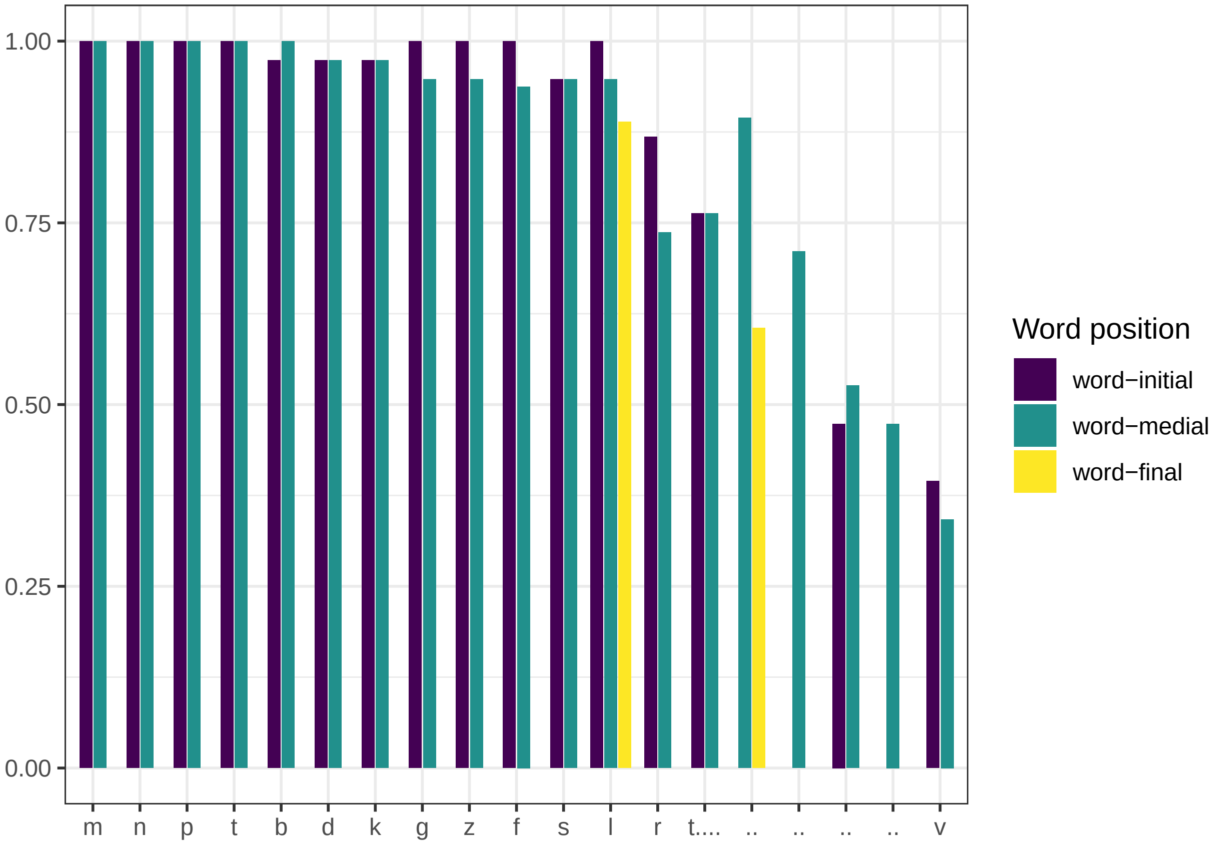
```r
data_by_position <- database %>%
  dplyr::select(Position, LexifierPhoneme, PlaceStability, MannerStability) %>%
  mutate(Position = tolower(Position))

data_by_position$PlaceStability <- as.numeric(data_by_position$PlaceStability)

data_by_position$MannerStability <- as.numeric(data_by_position$MannerStability)
```

Next, calculate stability for each segment according to its word position.

```
position_stability <- mutate(data_by_position, GlobalStability =
                               (PlaceStability + MannerStability) / 2)

position_results <- position_stability %>%
  group_by(LexifierPhoneme, Position) %>%
  summarize(m = mean(GlobalStability, na.rm = TRUE))
```

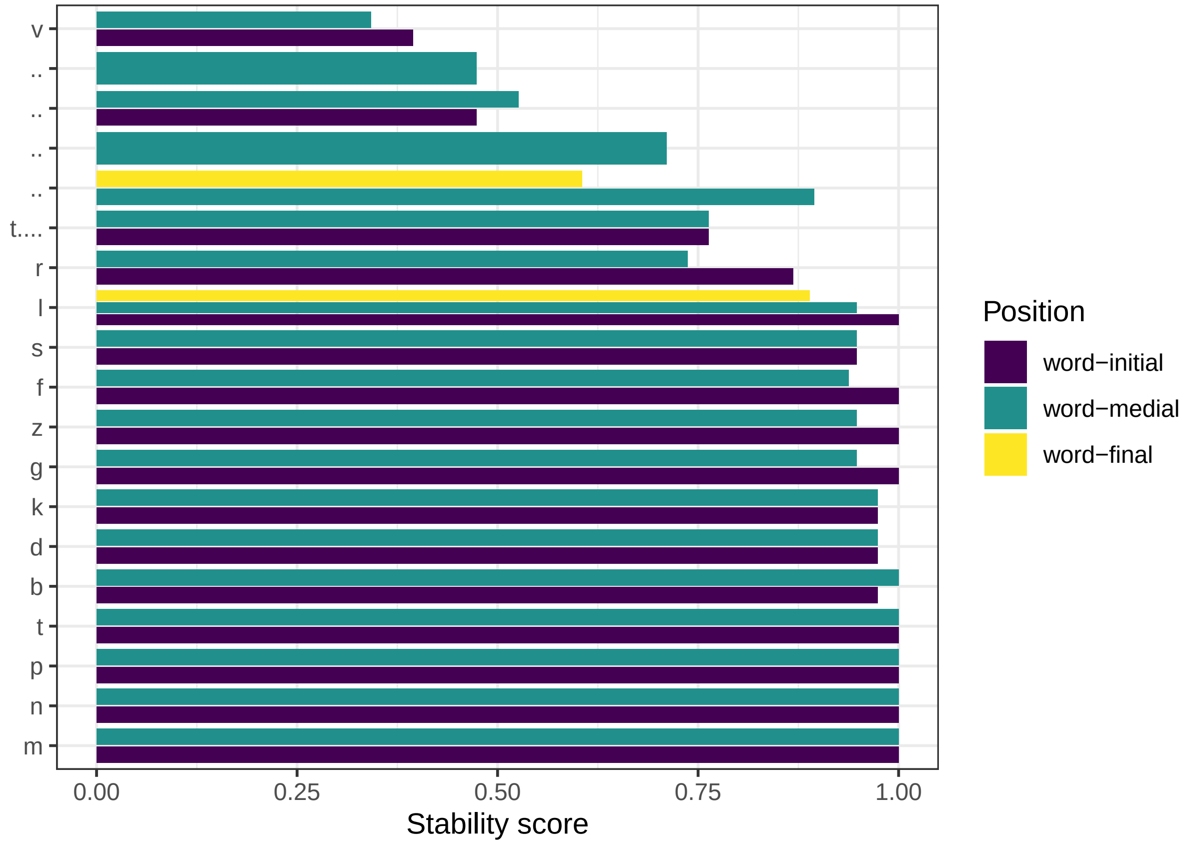
And plot the results for all segments.

```
position_results$Position <- factor(position_results$Position,
                                    levels = c("word-initial",
                                               "word-medial",
                                               "word-final"),
                                    ordered = TRUE)

ggplot(position_results, aes(x = reorder(LexifierPhoneme, -m), y = m, fill = Position)) +
  geom_col(position = position_dodge2(width = 0.9, preserve = "single")) +
  theme(
    axis.title.x = element_blank(),
    axis.title.y = element_blank()
  ) +
  labs(x = "Lexifier phoneme", y = "Mean stability", fill = "Word position")
```

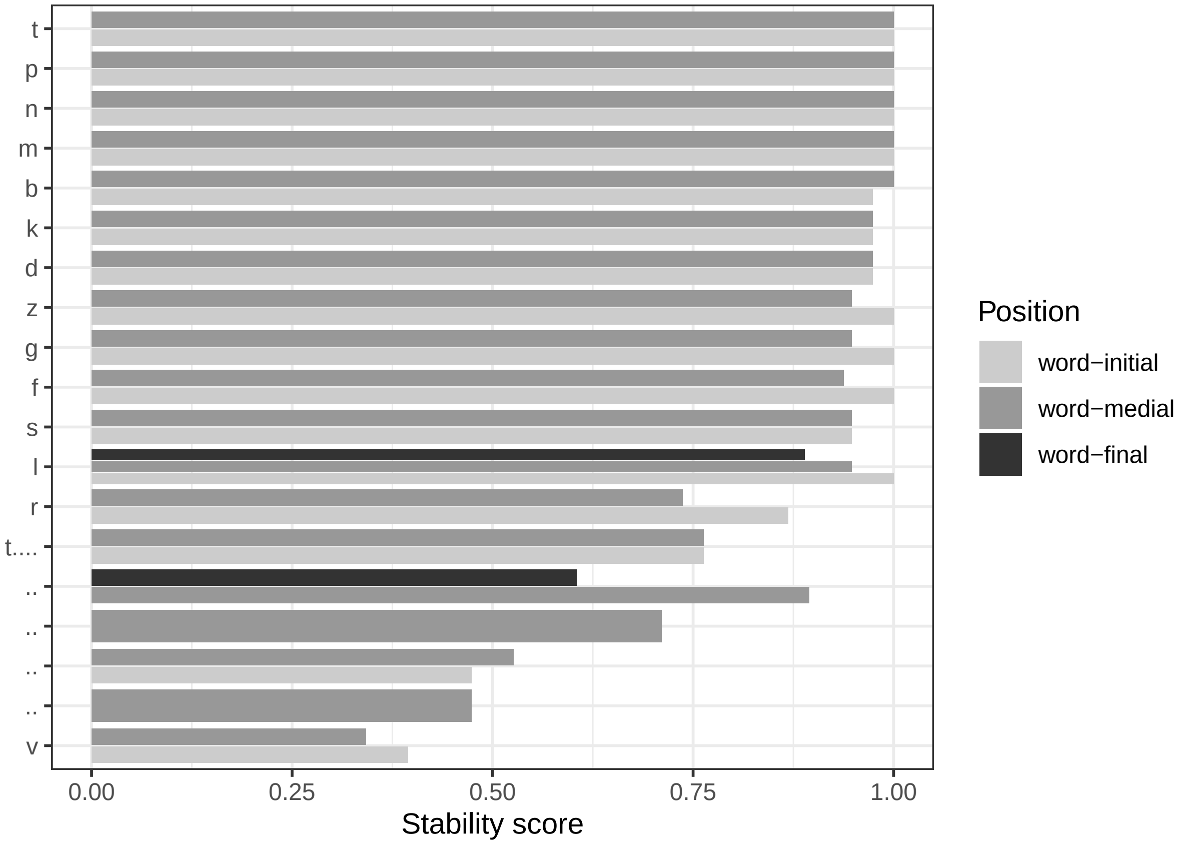


Flip horizontally.

```
ggplot(position_results) +
  geom_bar(
    aes(
      x = m,
      y = reorder(LexifierPhoneme, -m),
      # y = LexifierPhoneme,
      fill = Position
    ),
    stat = "identity",
    show.legend = TRUE,
    position = "dodge2"
  ) +
  labs(x = "Stability score", y = "", fill = "Position")
```

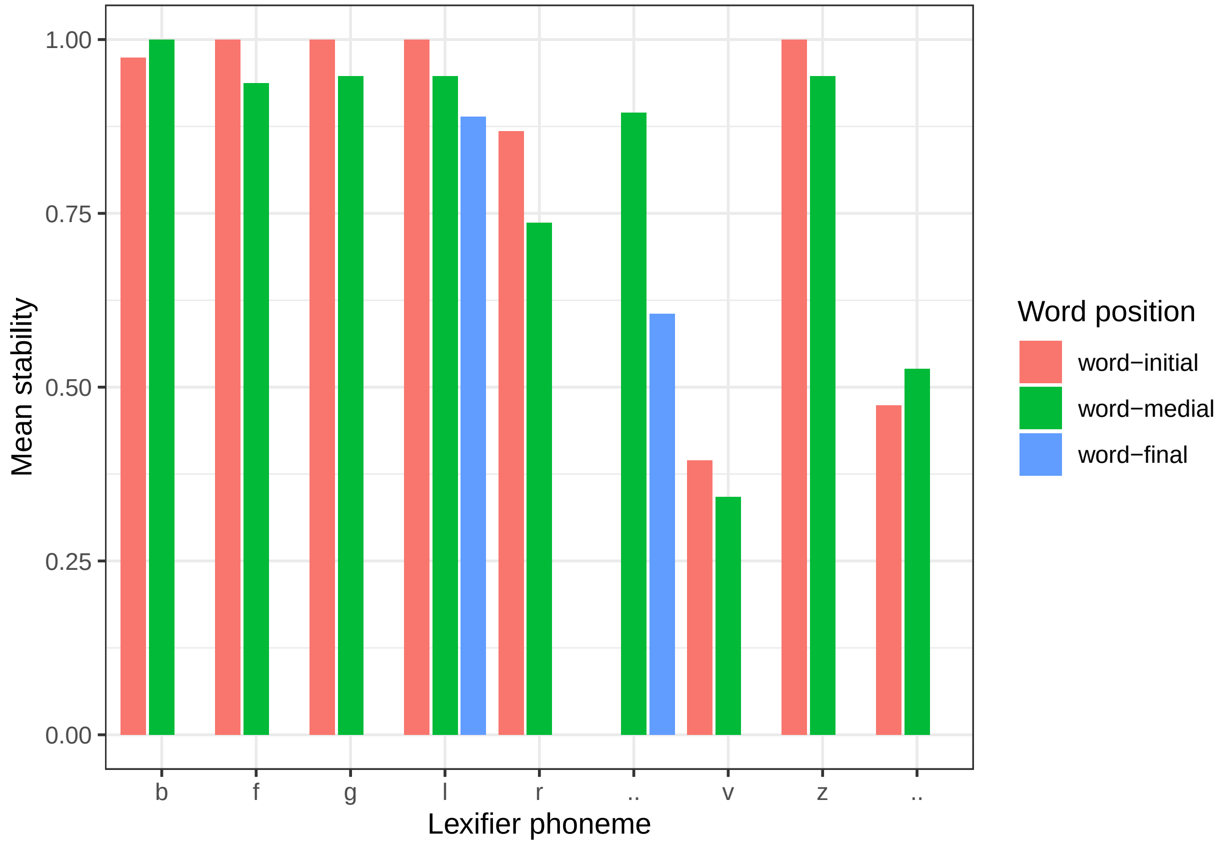


```
ggplot(position_results) +
  geom_bar(
    aes(
      x = m,
      y = reorder(LexifierPhoneme, m),
      fill = Position
    ),
    stat = "identity",
    show.legend = TRUE,
    position = "dodge2"
  ) +
  labs(x = "Stability score", y = "", fill = "Position") +
  scale_fill_grey(start = 0.8, end = 0.2)
```



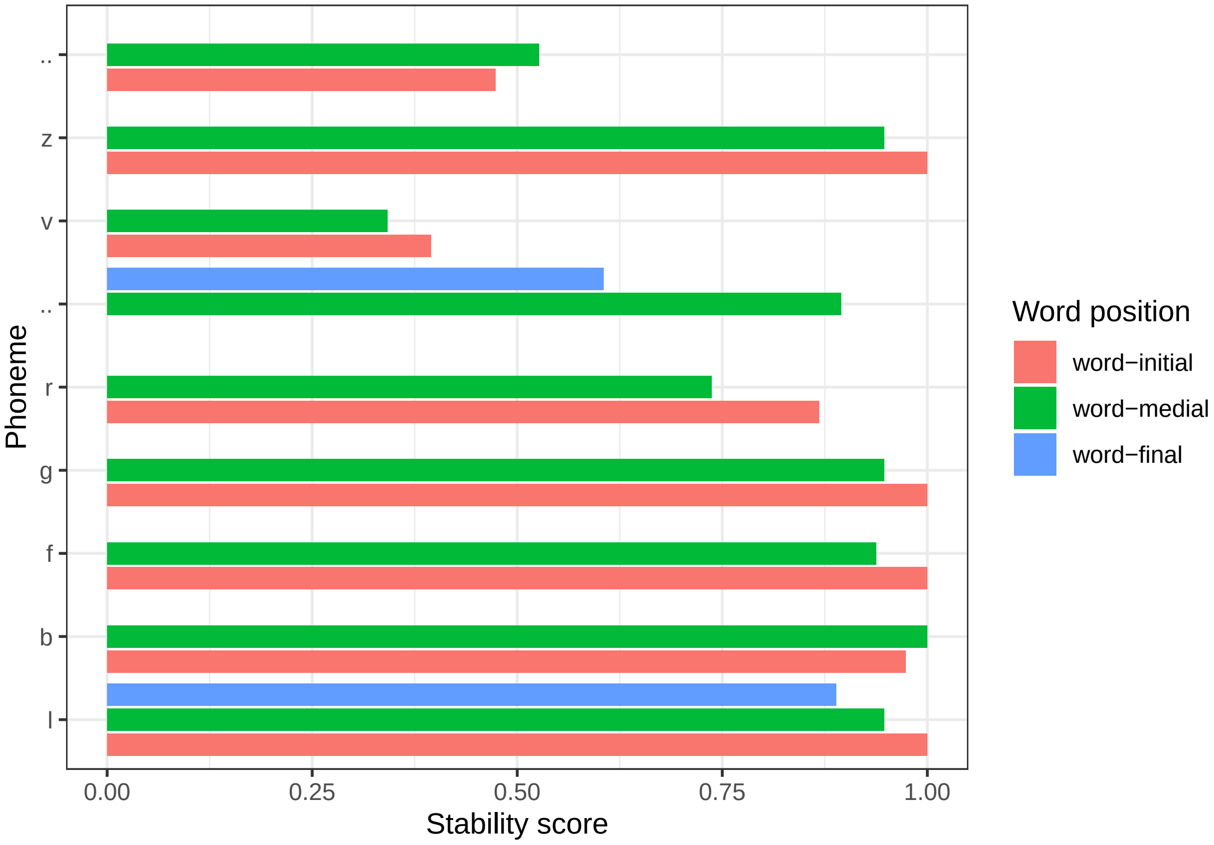
Plot the results for segments that show differences.

```
position_results1 <- position_results %>%
  pivot_wider(names_from = Position, values_from = m)

different_position <- subset(position_results1, position_results1$`word-initial`
                             != position_results1$`word-medial` |
                               position_results1$`word-final`
                             != position_results1$`word-medial`)

different_position_results <- different_position %>%
  pivot_longer(c(`word-initial`, `word-medial`, `word-final`),
               names_to = "Position", values_to = "m")

different_position_results$Position <- factor(different_position_results$Position,
                                              levels = c("word-initial",
                                                         "word-medial",
                                                         "word-final"))

ggplot(
  different_position_results,
  aes(x = LexifierPhoneme, y = m, fill = Position)
) +
  geom_col(position = position_dodge2(width = 0.9, preserve = "single")) +
  labs(x = "Lexifier phoneme", y = "Mean stability", fill = "Word position")
```



Flip horizontally.

```
ggplot(different_position_results) +
  geom_bar(
    aes(
      x = m,
      y = reorder(LexifierPhoneme, m),
      fill = Position
    ),
    stat = "identity",
    show.legend = TRUE,
    position = "dodge2"
  ) +
  labs(x = "Stability score", y = "Phoneme", fill = "Word position")
```



4.4 Typological frequency and borrowability

First, we turn the data into ordinal values. Ordinal data was generated by ranking the percentage values of stability, borrowability and typological frequency from 1 to 19. Duplicate values summed and averaged in the ranking.

Cross linguistic frequency

```
Typology <- c(1, 2, 3, 4, 5, 5, 7, 8, 9, 10, 11.5, 11.5, 13, 14, 15, 16, 17, 18, 19)
consonant <- c("m", "k", "p", "n", "t", "l", "s", "b", "g", "d", "f", "r", " ",
               "t ","z","v"," "," "," ")
df_typ <- data.frame(Typology, consonant)
```

2) Borrowability

```
Borrowability <- c(1, 2, 3, 4, 5, 6, 6, 8, 9, 10, 11, 12, 13, 14.5, 14.5, 16, 17, 18, 19)
consonant <- c("f", "g", "t ","b","z","v","d","r","p","l","s"," "," "," ","k"
               ," ","t","n","m")
df_bor <- data.frame(Borrowability, consonant)
```

3) Stability values

```
Stability <- c(2, 2, 2, 2, 5, 7, 7, 7, 8, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19)
consonant <- c("t", "p", "n", "m", "f", "b", "k", "g", "d", "z", "s", "l", "r",
               "t " ," "  ," " ," " ," " ,"v")
df_sta <- data.frame(Stability, consonant)
```

Then, we create the data frames and prepare them for non-parametric tests. Long format with joint groups.

```
df_friedman <- left_join(df_sta, df_bor, by="consonant")

order_df <- left_join(df_friedman, df_typ, by="consonant")
head(order_df)
```

```
##   Stability consonant Borrowability Typology
## 1         2         t            17      5.0
## 2         2         p             9      3.0
## 3         2         n            18      4.0
## 4         2         m            19      1.0
## 5         5         f             1     11.5
## 6         7         b             4      8.0
```

```
df_long <- order_df %>%
  gather(key = "conditions",value = "order", Borrowability, Stability, Typology)

head(df_long)
```

```
##   consonant    conditions order
## 1         t Borrowability    17
## 2         p Borrowability     9
## 3         n Borrowability    18
## 4         m Borrowability    19
## 5         f Borrowability     1
## 6         b Borrowability     4
```

For the Spearman's rank correlation coefficient, we use large format and separated groups.

```
df_sta_bor <- left_join(df_sta, df_bor, by="consonant")
df_sta_typ <- left_join(df_sta, df_typ, by="consonant")
```

Convert to long format.

```
sta_bor_long <- df_sta_bor %>%
  gather(key = "conditions", value = "order", Borrowability, Stability)

sta_typ_long <- df_sta_typ %>%
  gather(key = "conditions", value = "order", Typology, Stability)
```

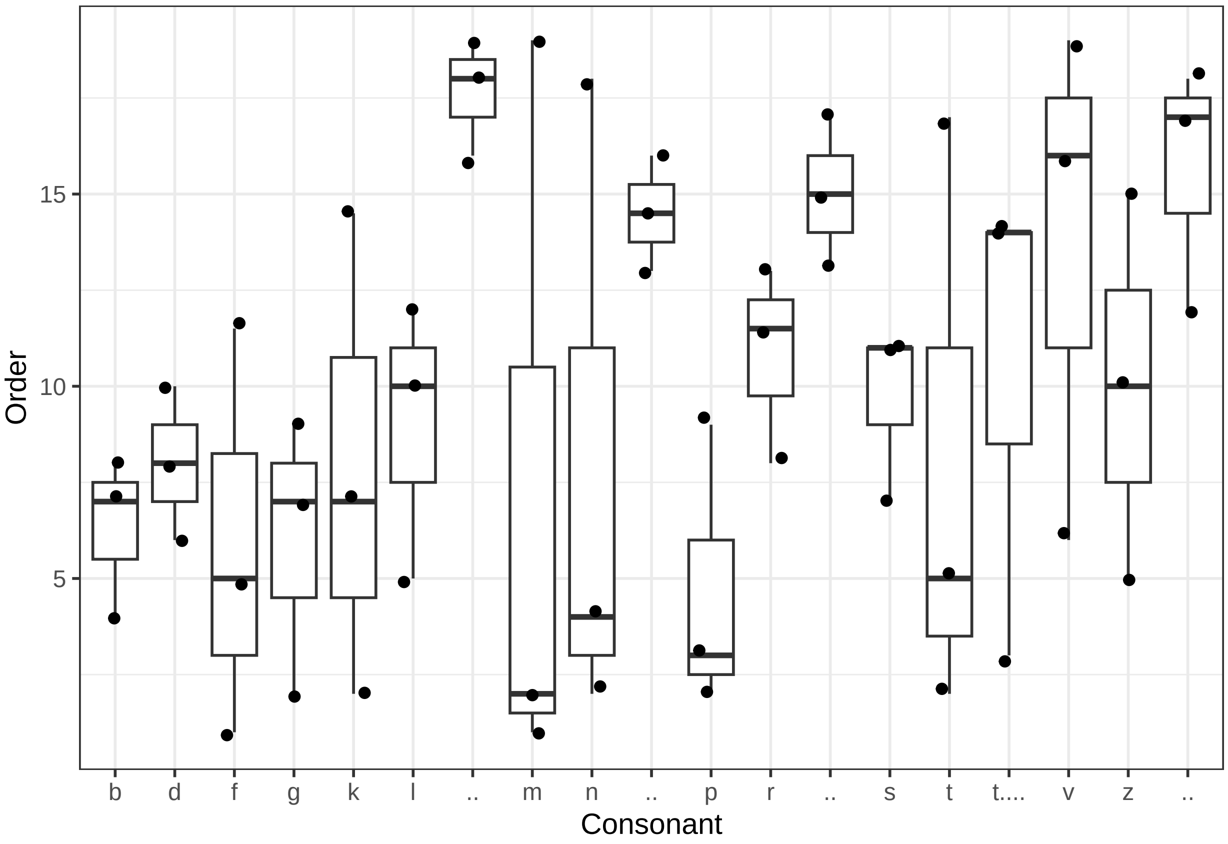
Finally, we perform the non-parametric tests. The statistical summary:

```
df_long %>% group_by(conditions) %>%
  summarise(n = n(), mean = mean(order), sd = sd(order))
```

```
## # A tibble: 3 x 4
##   conditions        n  mean    sd
##   <chr>         <int> <dbl> <dbl>
## 1 Borrowability    19  9.95  5.66
## 2 Stability        19  9.84  5.76
## 3 Typology         19  9.95  5.67
```

A first plot.

```
ggplot(df_long, aes(x = consonant, y = order)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(width = 0.2) +
  theme(legend.position="top") +
  ylab("Order") +
  xlab("Consonant") # @CS: FIX
```



1. Friedman test.

```
friedman.test(y = df_long$order, groups = df_long$conditions, blocks = df_long$consonant)
```

```
##
##  Friedman rank sum test
##
## data:  df_long$order, df_long$conditions and df_long$consonant
## Friedman chi-squared = 2.5135, df = 2, p-value = 0.2846
```

```
df_long %>% friedman_effsize(order ~ conditions | consonant)
```

```
## # A tibble: 1 x 5
##   .y.         n effsize method    magnitude
## * <chr> <int>   <dbl> <chr>     <ord>
## 1 order    19  0.0661 Kendall W small
```

2. Conover's all-pairs test.

```
frdAllPairsConoverTest(
  y = df_long$order,
  groups = df_long$conditions,
  blocks = df_long$consonant,
  p.adjust.method = "bonf")
```

```
##           Borrowability Stability
## Stability 0.67          -
## Typology  0.43          1.00
```

3. Durbin's all-pairs test.

```
durbinAllPairsTest(
  y      = df_long$order,
  groups = df_long$conditions,
  blocks = df_long$consonant,
  p.adjust.method = "holm")
```

```
##           Borrowability Stability
## Stability 0.44          -        
## Typology  0.43          0.81     
```

4. Spearman's correlation coefficient.

4.1 Stability ~ Borrowability.

```
cor.test(x=df_sta_bor$Borrowability,
         y=df_sta_bor$Stability,
         method = 'spearman')
```

```
## 
##  Spearman's rank correlation rho
## 
## data:  df_sta_bor$Borrowability and df_sta_bor$Stability
## S = 1252.8, p-value = 0.687
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##         rho 
## -0.09894132
```

4.2 Stability ~ Typological frequency.

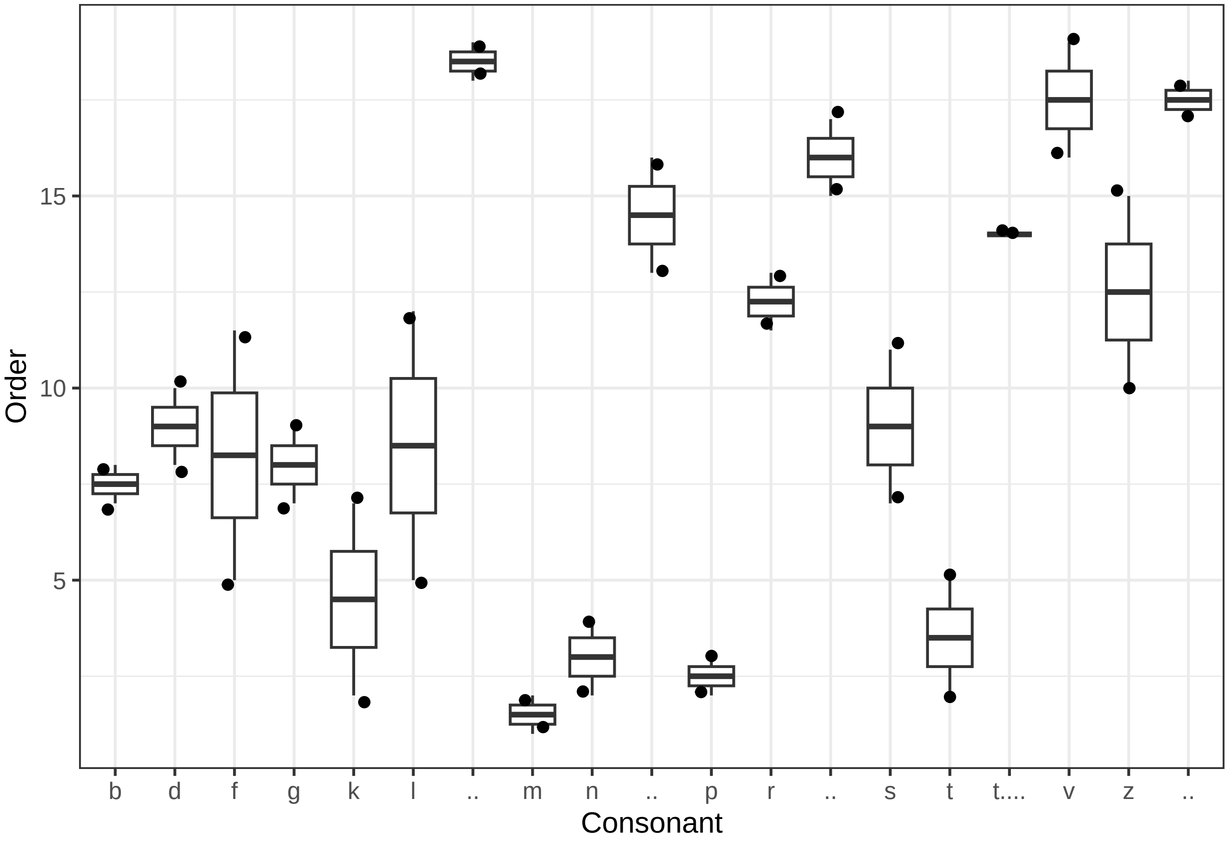
```
cor.test(x=df_sta_typ$Typology,
         y=df_sta_typ$Stability,
         method = 'spearman')
```

```
## 
##  Spearman's rank correlation rho
## 
## data:  df_sta_typ$Typology and df_sta_typ$Stability
## S = 197.88, p-value = 1.295e-05
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho 
## 0.826425
```

Next, visualize the results. Box plots.

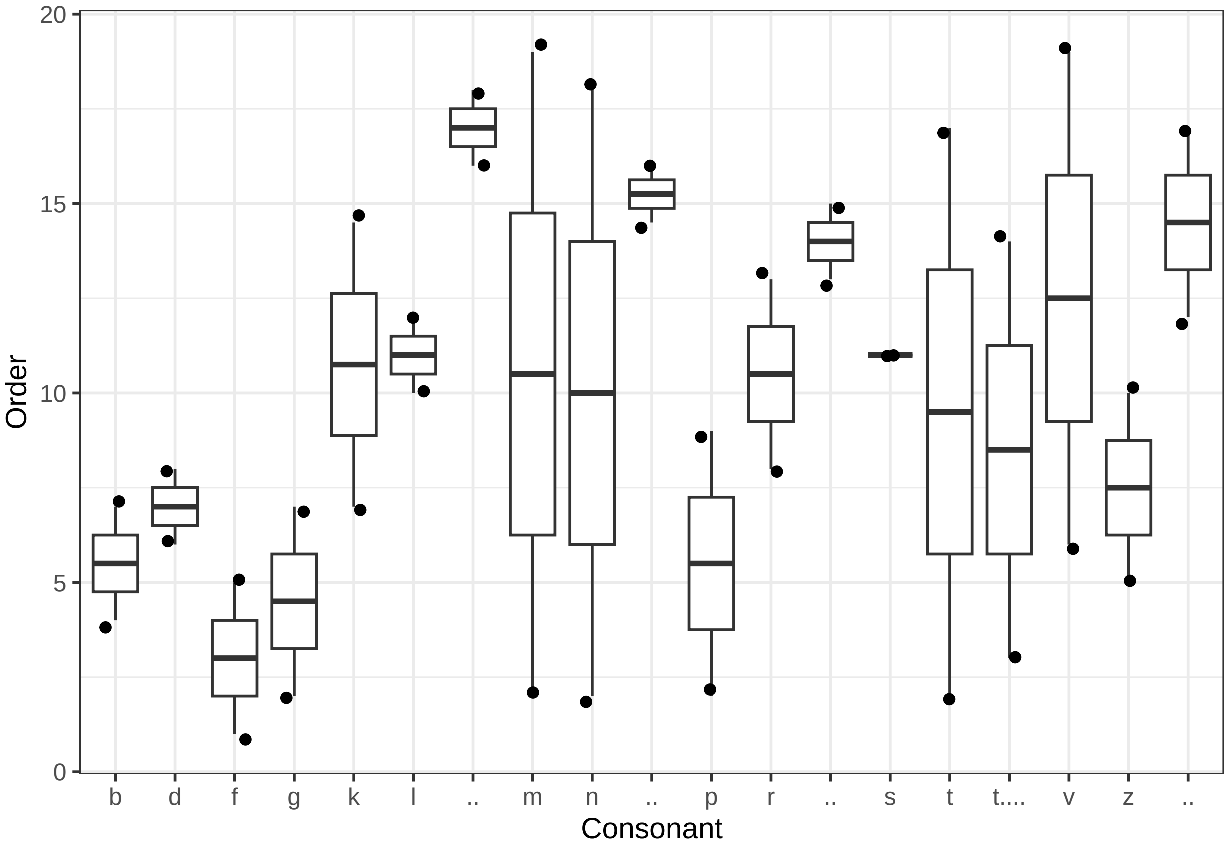
Stability vs typological frequency.

```
ggplot(sta_typ_long, aes(x = consonant, y = order)) + geom_boxplot(outlier.shape = NA) +
  geom_jitter(width = 0.2) + theme(legend.position="top") +
  ylab("Order") +
  xlab("Consonant") # @CS: FIX
```



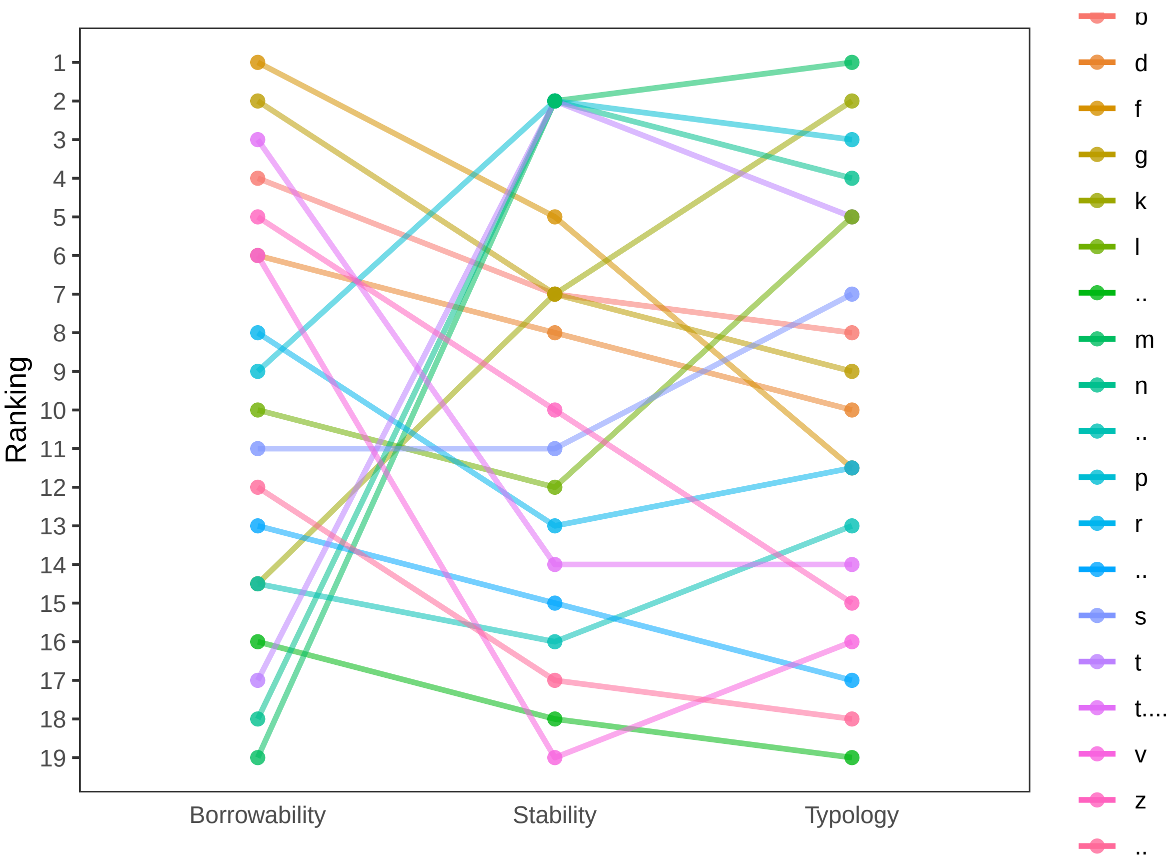
Stability vs borrowability.

```
ggplot(sta_bor_long, aes(x = consonant, y = order)) + geom_boxplot(outlier.shape = NA) +
  geom_jitter(width = 0.2) + theme(legend.position="top") +
  ylab("Order") +
  xlab("Consonant") # @CS: FIX
```



Bump chart.

```
ggplot(data = df_long, aes(x = conditions, y = order, group = consonant)) +
  geom_line(aes(color = consonant, alpha = 1), size = 1) +
  geom_point(aes(color = consonant, alpha = 1), size = 2,  alpha = 0.8) +
  scale_y_reverse(breaks = 1:nrow(df_long)) +
  scale_alpha(guide = 'none') +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.ticks.x = element_blank())+
  ylab("Ranking")+
  xlab("")
```



4.5 Inventory size and frequency across substrates

Get the data.

```
inv <- read.csv("Inventories.csv")
head(inv)
```

```
##   ID   Language Category Phoneme Notes              Source PhoibleID
## 1 19 Portuguese Lexifier       p       Castro2013[242-248]        NA
## 2 19 Portuguese Lexifier       b       Castro2013[242-248]        NA
## 3 19 Portuguese Lexifier       t       Castro2013[242-248]        NA
## 4 19 Portuguese Lexifier       d       Castro2013[242-248]        NA
## 5 19 Portuguese Lexifier       k       Castro2013[242-248]        NA
## 6 19 Portuguese Lexifier       g       Castro2013[242-248]        NA
```

Prepare the data.

```
df_inv_long <- inv %>% dplyr::select(Language, Phoneme) %>%  mutate(newcol = 1)

df_inv <- df_inv_long %>% pivot_wider(names_from = Language,
                                      values_from = newcol, values_fill = 0)

df_total_inv <- df_inv %>% mutate_at(c(2:38), as.numeric)

head(df_total_inv)
```

```
## # A tibble: 6 x 38
##   Phoneme Portuguese `Timor Pidgin` `Papia Kristang` `Patua Macau` Tetum
##   <chr>        <dbl>          <dbl>            <dbl>         <dbl> <dbl>
## 1 p                1              1                1             1     0
## 2 b                1              1                1             1     1
## 3 t                1              1                0             1     1
## 4 d                1              1                1             1     1
## 5 k                1              1                1             1     1
## 6 g                1              1                1             1     0
## # i 32 more variables: `Larantuka Malay` <dbl>, `Standard Malay` <dbl>,
## #   `Hokkien Chinese` <dbl>, Cantonese <dbl>, Malayalam <dbl>,
## #   `Sri Lanka Portuguese` <dbl>, Diu <dbl>, Daman <dbl>, Korlai <dbl>,
## #   Kannur <dbl>, Sinhala <dbl>, Tamil <dbl>, Gujarati <dbl>, Marathi <dbl>,
## #   Santome <dbl>, Principense <dbl>, Angolar <dbl>, `Fa d'Ambo` <dbl>,
## #   `Cape Verdean Brava` <dbl>, `Cape Verdean Sao Vicente` <dbl>,
## #   `Cape Verdean Santo Antao` <dbl>, `Cape Verdean Fogo` <dbl>, ...
```

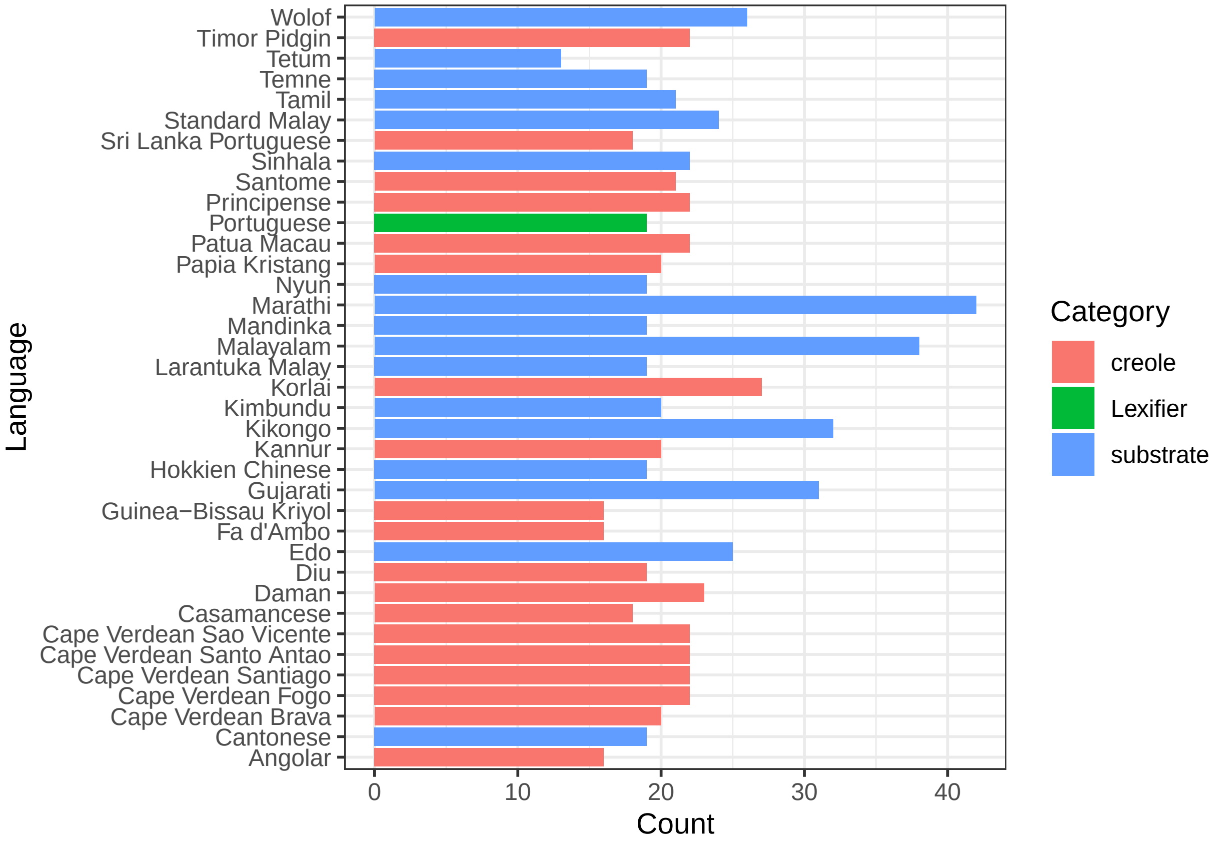
Measuring the inventory size. Get the consonant inventory size for all languages.

```r
cons_count <- df_total_inv %>% dplyr::select(c(2:38)) %>% mutate_at(c(1:37), as.numeric)

count <- colSums(cons_count [,c(1:37)]) #%>% unname(colSums(count))

cons_lg <- dplyr::select(df_inv_long, "Language")

Language <- unique(cons_lg$Language)

category <- inv %>% dplyr::select(Language, Category)

count_lg <- data.frame(cbind(Language, count))

count_lg_1 <- inner_join(count_lg, category, by = "Language") %>% distinct()

inv_size <- transform(count_lg_1, count = as.numeric(count))

head(inv_size)
```

```
##          Language count  Category
## 1      Portuguese    19  Lexifier
## 2    Timor Pidgin    22    creole
## 3  Papia Kristang    20    creole
## 4     Patua Macau    22    creole
## 5           Tetum    13 substrate
## 6 Larantuka Malay    19 substrate
```

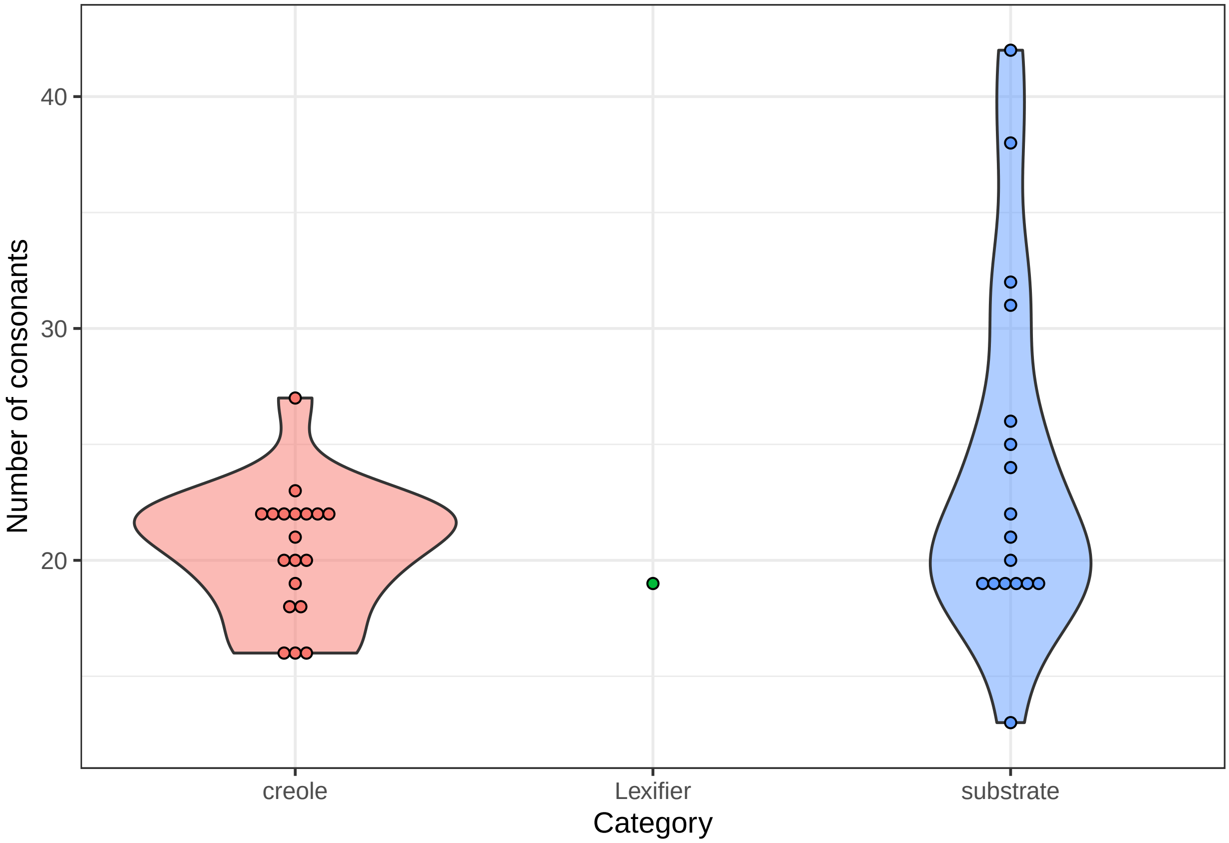
Which languages have bigger inventories?

```r
ggplot(inv_size) +
  geom_bar(aes(x = Language,
  y = count,
  fill = Category),
  stat = "identity",
  show.legend = TRUE,
  position = "dodge2") + coord_flip() +
  xlab("Language") +
  ylab("Count") # @CS: FIX
```



Violin plot shows that the majority of creoles have larger consonant inventories than Portuguese.

```
ggplot(inv_size, aes(x = Category, y = count, fill = Category)) +
  geom_violin(alpha = 0.5) +
  geom_dotplot(binaxis = "y",
               stackdir = "center",
               dotsize = 0.5) +
  theme(legend.position = "none") +
  ylab("Number of consonants")
```



Consonant frequency in all languages involved. Count frequent consonants.

```r
total <- rowSums(cons_count)

cons_freq <- data.frame(cbind(df_total_inv$Phoneme, total))

cons_freq <- transform(cons_freq, total = as.numeric(total))

colnames(cons_freq)[1] <- "LexifierPhoneme"
```

Is there a relationship between this frequency and the stability values? Portuguese consonants only.

```r
cons_freq_pt <- cons_freq %>% subset(LexifierPhoneme %in%
                                        c('b','d','f','g','k','l',' ','m','n',
                                          ' ','p','t ',' ','s','t','v','z',' ', 'r'))
```

First dataset: relative frequency values.

```r
cons_freq_rel <- cons_freq_pt %>% mutate(frequency = total/37)
```

Second dataset: stability values.

```r
consonant_global_stability <- read.csv("consonant_global_stability.csv")
```

Merge the datasets.

```r
cor_freq_sta <- left_join(consonant_global_stability, cons_freq_rel, by='LexifierPhoneme')
```

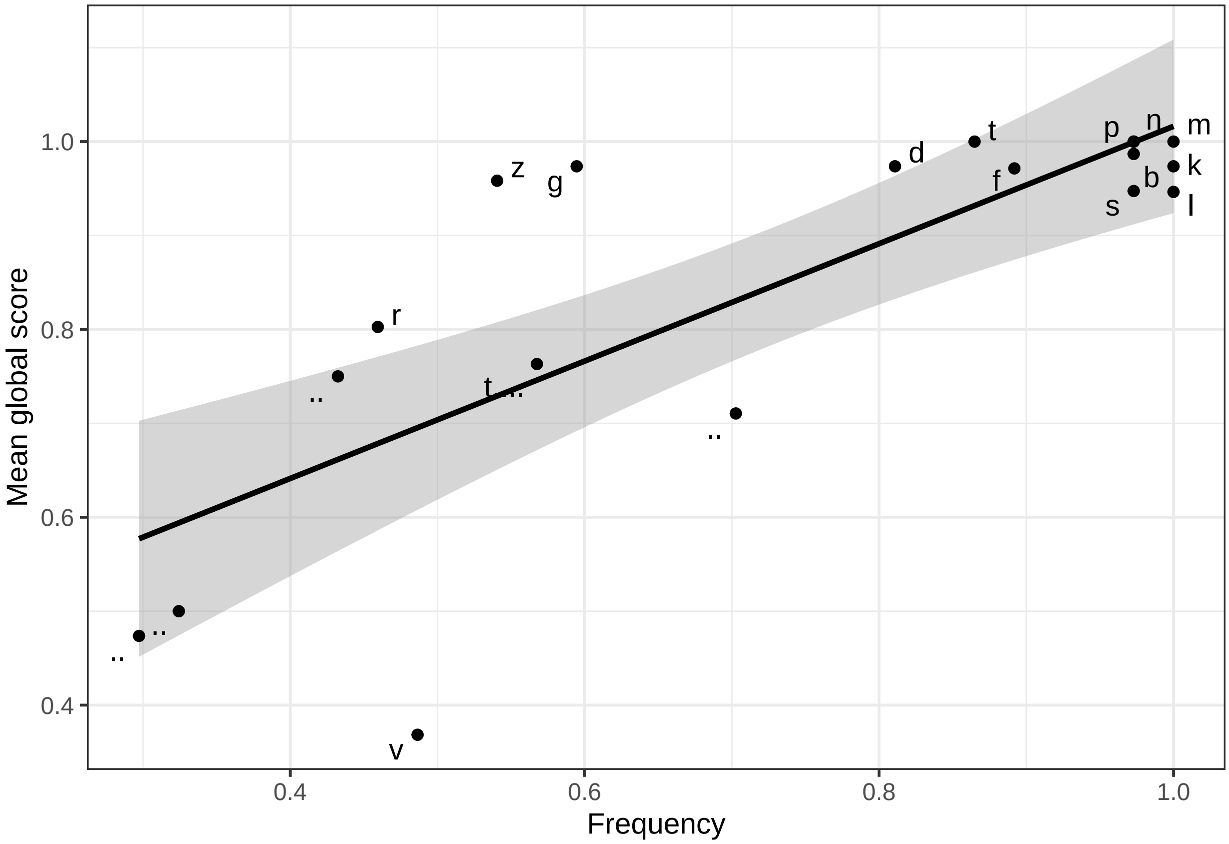
Results of a regression analysis.

```r
fs <- lm(frequency ~ mglobal, data=cor_freq_sta)
summary(fs)
```

```
##
## Call:
## lm(formula = frequency ~ mglobal, data = cor_freq_sta)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.29863 -0.07151  0.03980  0.11383  0.22911
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.1060     0.1642  -0.645    0.527
## mglobal       0.9862     0.1887   5.226 6.85e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1618 on 17 degrees of freedom
## Multiple R-squared:  0.6163, Adjusted R-squared:  0.5938
## F-statistic: 27.31 on 1 and 17 DF,  p-value: 6.851e-05
```

Plot the results.

```r
fs_plot <- ggplot(cor_freq_sta, aes(x = frequency, y = mglobal, label = LexifierPhoneme)) +
  geom_smooth(method = "lm", colour="black") +
  geom_point() +
  ylab("Mean global score") +  # @CS: FIX
  xlab("Frequency") # +
  #geom_text(aes(label=V1), hjust=3, vjust=0)

fs_plot + geom_text_repel(aes(label=LexifierPhoneme))
```



There is a relationship between stability and frequency across all languages in the sample. But does it make sense? We are measuring the consonants in all categories and the present of the lexifier and the creoles may influence the results. Thus, we subset and measure the frequency across substrates only, but this procedure may just increase the lack of correlation.

Is there a relationship between consonant stability and their presence in the inventories of the substrate languages?

Data preparation (substrates only).

```
inv_subs <- inv %>% subset(Category == 'substrate') %>%
  dplyr::select(Language, Phoneme) %>%
  mutate(newcol = 1)

inv_subs_long <- inv_subs %>%
  pivot_wider(names_from = Language, values_from = newcol, values_fill = 0)

inv_subs_long <- inv_subs_long %>%
  mutate_at(c(2:18), as.numeric)

head(inv_subs_long)
```

```
## # A tibble: 6 x 18
##   Phoneme Tetum `Larantuka Malay` `Standard Malay` `Hokkien Chinese` Cantonese
##   <chr>   <dbl>             <dbl>            <dbl>             <dbl>     <dbl>
## 1 m           1                 1                1                 1         1
## 2 k           1                 1                1                 1         1
## 3 w           1                 1                1                 1         1
## 4 n           1                 1                1                 1         1
## 5 t           1                 1                1                 1         1
## 6 l           1                 1                1                 1         1
## # i 12 more variables: Malayalam <dbl>, Sinhala <dbl>, Tamil <dbl>,
## #   Gujarati <dbl>, Marathi <dbl>, Edo <dbl>, Kikongo <dbl>, Kimbundu <dbl>,
## #   Wolof <dbl>, Temne <dbl>, Mandinka <dbl>, Nyun <dbl>
```

Sum row values and subset to consonants which have correspondents in Portuguese.

```
subs_count <- inv_subs_long %>%
  dplyr::select(c(2:18))

total_subs <- rowSums(subs_count)

subs_freq <- data.frame(cbind(inv_subs_long$Phoneme, total_subs))

subs_freq <- transform(subs_freq, total_subs = as.numeric(total_subs))

colnames(subs_freq)[1] <- "LexifierPhoneme"

subs_freq <- subs_freq %>%
  subset(LexifierPhoneme %in% c('b','d','f','g','k','l',' ','m','n',' ','p',
                                't ',' ','s', 't','v','z',' ', 'r'))
```

Get relative values.

```
subs_freq_rel <- subs_freq %>% mutate(frequency = total_subs/17)
```

Merge datasets.

```
subs_sta <- left_join(consonant_global_stability, subs_freq_rel, by='LexifierPhoneme')
```

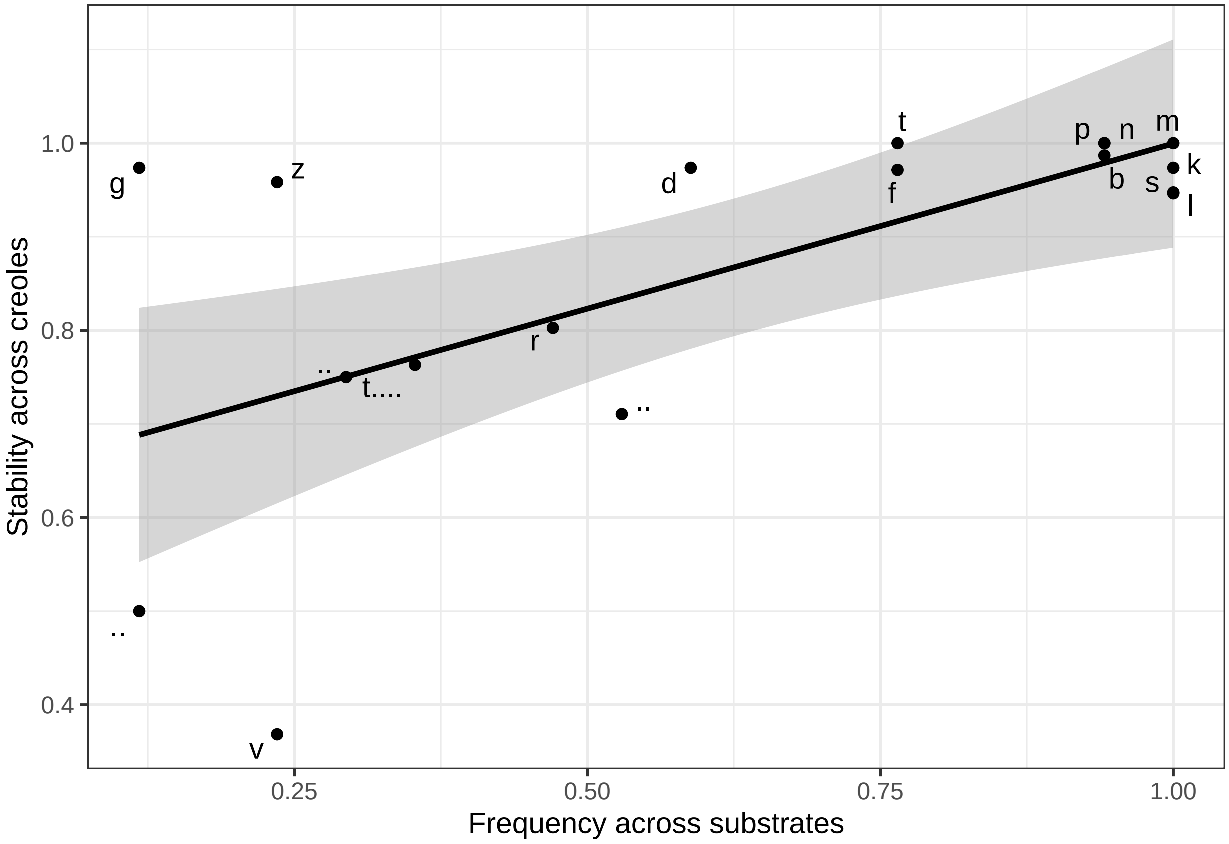
Results of a simple regression.

```
subs_sta_lm <- lm(frequency ~ mglobal, data=subs_sta)
summary(subs_sta_lm)
```

```
##
## Call:
## lm(formula = frequency ~ mglobal, data = subs_sta)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.63261 -0.13468  0.05119  0.18579  0.28145
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.3825     0.3091  -1.238  0.23374
## mglobal       1.1634     0.3486   3.338  0.00418 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2672 on 16 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.4105, Adjusted R-squared:  0.3736
## F-statistic: 11.14 on 1 and 16 DF,  p-value: 0.004175
```

Plot the results.

```
subsfreq_sta_cor <- ggplot(subs_sta, aes(x = frequency, y = mglobal, label = LexifierPhoneme)) +
  geom_smooth(method = "lm", colour="black") +
  geom_point() +
  xlab("Frequency across substrates") + ylab("Stability across creoles") # +
 # geom_text(aes(label=V1), hjust=3, vjust=0)


subsfreq_sta_cor + geom_text_repel(aes(label=LexifierPhoneme))
```



When we consider the correlation between the consonant stability in creoles and the frequency of these consonants in the substrates only, we find that there is a weaker correlation (if we compare with the results above). However, this correlation is statistically significant (p-value: 0.004). Nothing the outliers which are normally voiced consonants.

Typological frequency vs. Substrate frequency.

Typological frequency data, extrated from PHOIBLE.

```
typ_freq <- read.csv("typ_freq.csv")
```

Merge the datasets.

```
#consonant_global_stability <- read.csv("consonant_global_stability.csv")
typ_sta <- left_join(typ_freq, consonant_global_stability, by='LexifierPhoneme')
```

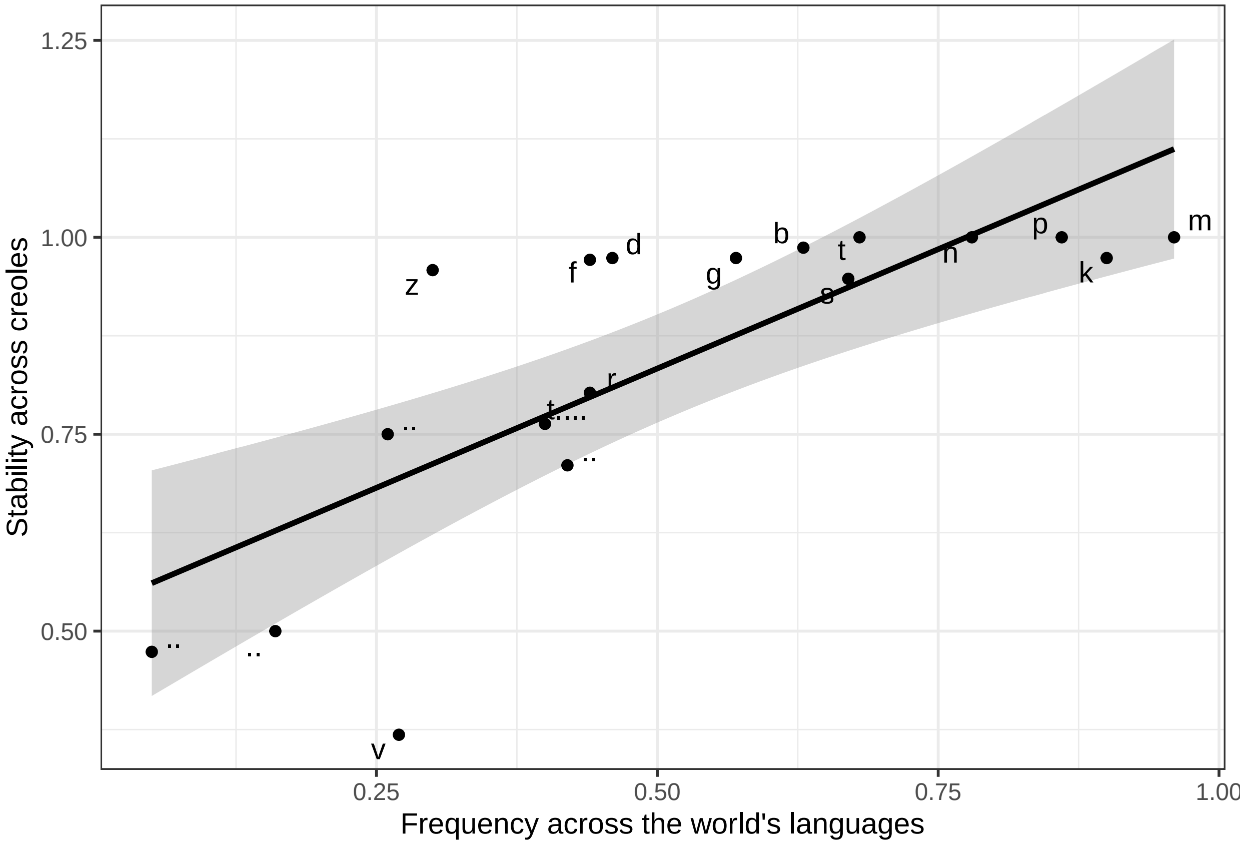
Results of a linear regression.

```
typ_sta_lm <- lm(TypologicalFreq ~ mglobal, data=typ_sta)
summary(typ_sta_lm)
```

```
##
## Call:
## lm(formula = TypologicalFreq ~ mglobal, data = typ_sta)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.32614 -0.09953 -0.02402  0.09888  0.29370
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.2975     0.1761  -1.689 0.110531
## mglobal       0.9638     0.2035   4.737 0.000223 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1732 on 16 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.5838, Adjusted R-squared:  0.5577
## F-statistic: 22.44 on 1 and 16 DF,  p-value: 0.0002233
```

Plot the results.

```
typ_sta_cor <- ggplot(typ_sta, aes(x = TypologicalFreq, y = mglobal, label = LexifierPhoneme)) +
  geom_smooth(method = "lm", colour="black") +
  geom_point() +
  xlab("Frequency across the world's languages") + ylab("Stability across creoles") # +
 # geom_text(aes(label=V1), hjust=3, vjust=0)

typ_sta_cor + geom_text_repel(aes(label=LexifierPhoneme))
```



References

Becker, Richard A., Allan R. Wilks, Ray Brownrigg, Thomas P Minka, and Alex Deckmyn. 2023. *Maps: Draw Geographical Maps*. https://CRAN.R-project.org/package=maps.

Carvalho, Ana Maria, and Dante Lucchesi. 2016. "Portuguese in Contact." In *The Handbook of Portuguese Linguistics*, 41–55. Wiley Blackwell. https://doi.org/10.1002/9781118791844.ch3.

Faraclas, Nicholas, Don Walicek, Mervyn Alleyne, Wilfredo Geigel, and Luis Ortiz. 2007. "The Complexity That Really Matters: The Role of Political Economy in Creole Genesis." In *Deconstructing Creole: New Horizons in Language Creation*, edited by U. Ansaldo, S. J. Matthews, and L. Lim, 227–64. John Benjamins. https://doi.org/10.1075/tsl.73.12far.

Hennig, Christian, and Bernhard Hausdorf. 2023. *Prabclus: Functions for Clustering and Testing of Presence-Absence, Abundance and Multilocus Genetic Data*. https://CRAN.R-project.org/package=prabclus.

Kassambara, Alboukadel. 2023a. *Ggpubr: 'Ggplot2' Based Publication Ready Plots*. https://CRAN.R-project.org/package=ggpubr.

———. 2023b. *Rstatix: Pipe-Friendly Framework for Basic Statistical Tests*. https://CRAN.R-project.org/package=rstatix.

Kuznetsova, Alexandra, Per B. Brockhoff, and Rune H. B. Christensen. 2017. "lmerTest Package: Tests in Linear Mixed Effects Models." *Journal of Statistical Software* 82 (13): 1–26. https://doi.org/10.18637/jss.v082.i13.

Pohlert, Thorsten. 2023. *PMCMRplus: Calculate Pairwise Multiple Comparisons of Mean Rank Sums Extended*. https://CRAN.R-project.org/package=PMCMRplus.

R Core Team. 2023. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Slowikowski, Kamil. 2022. *Ggrepel: Automatically Position Non-Overlapping Text Labels with 'Ggplot2'*. https://CRAN.R-project.org/package=ggrepel.

Wickham, Hadley, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, et al. 2019. "Welcome to the tidyverse." *Journal of Open Source Software* 4 (43): 1686. https://doi.org/10.21105/joss.01686.

Wood, S. N. 2004. "Stable and Efficient Multiple Smoothing Parameter Estimation for Generalized Additive Models." *Journal of the American Statistical Association* 99 (467): 673–86.

Xie, Yihui. 2021. *Knitr: A General-Purpose Package for Dynamic Report Generation in r.* https://yihui.org/knitr/.