R Notebook

I. Chechuro, P. Kasyanova

19 06 2017

Udi belongs to the Lezgic branch of the Northeast-Caucasian language family. It is spoken in a small area in the Northern Azerbaidzhan by ca. 10.000 people (Wikipedia). In Udi, there are two markers of participial relative clauses: -al and -i. The purpose of this study os to determine the principles of their use. There are two main hypotheses concerning the use of the markers. The first one is called temporal and implies that the relative tense determines the use of the marker. The second one is called aspectual and implies that the markers mark perfective and imperfective aspect. Since aspect is an extremely vague meaning that cannot be understood outside the utterance and without additional work with a native speaker, we limit this study to the first hypothesis which we are going to test. Thus, the main purpose of this study is to test the temporal hypothesis and to find out if there are any other factors that influence the choice of the marker. Our data comes from the corpus of Udi (only available as .doc files) and contains all 273 examples from both oral and written texts that are attested. Though the data is rather limited, there are no other data available for Udi. This study is organized as follows. First, we test single predictors. Then we create MDS-based semantic maps for the markers. Third, we perform a GLM analysis and build an S-curve for our data. After that we draw some conclusions. As the possible predictors we have taken the text, the type of text (oral/written), relative tense (pst/npst/NA), nominalization (1/0), the use of a postposition (0/oša/belšI), actionality (stat/dyn). It is also important that the oral corpus consists of novels written by only one person, thus, the possible (though not attested, as we will show below) significance of the type of the text could be explained by the bias of one speaker towards one form. First, let us read the data and introduce all the necessary libraries:

```
library(tidyverse)
```

```
## Loading tidyverse: ggplot2
## Loading tidyverse: tibble
## Loading tidyverse: tidyr
## Loading tidyverse: readr
## Loading tidyverse: purrr
## Loading tidyverse: dplyr
```

```
## Conflicts with tidy packages ----------------------------------------------
```

```
## filter(): dplyr, stats
## lag():    dplyr, stats
```

```
library(cluster)
library(smacof)
```

```
## Loading required package: plotrix
```

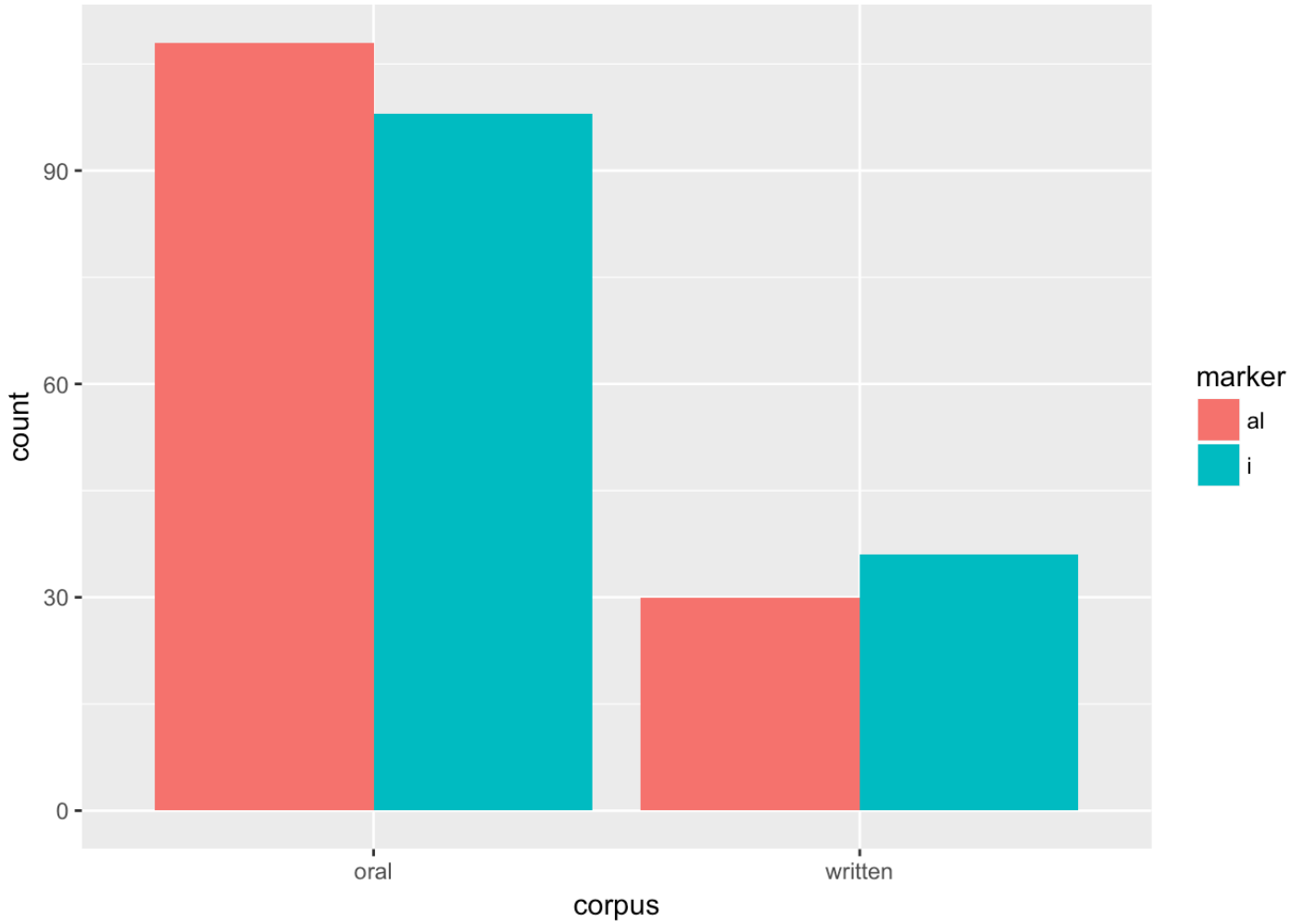
```
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

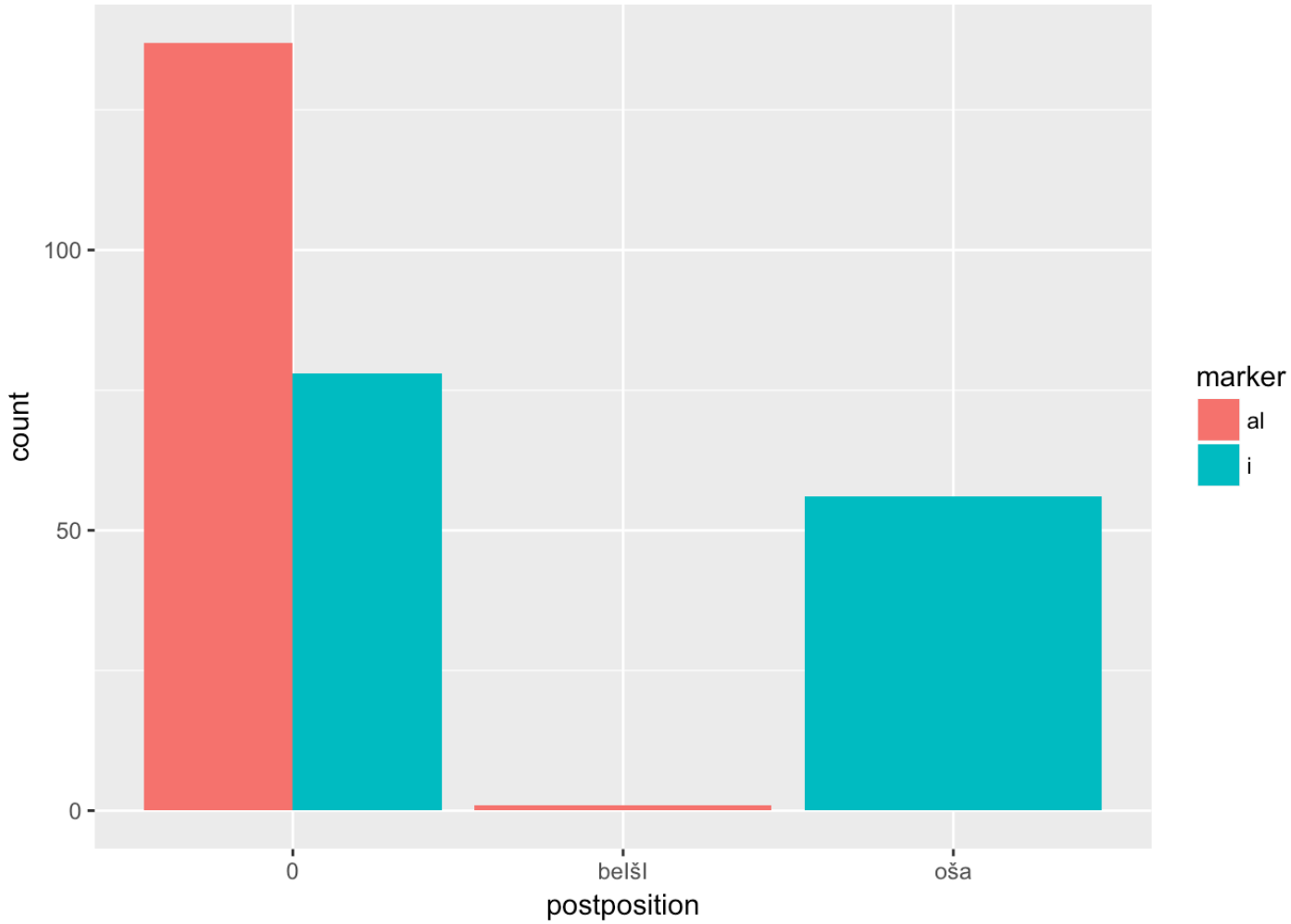
```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
aop <- read.csv("/Users/ilchec/Documents/Учеба/R_Masters/R Project/udi_data.csv",
sep=";")
aop_full <- aop
aop <- aop[,-c(3,4)]
View(aop_full)
```

To begin with, consider some simple plots showing the correspondence between the use of the markers and (1) text type and (2) postposition use. The distribution by the text type shows that both markers are used in oral speech as well as in written.

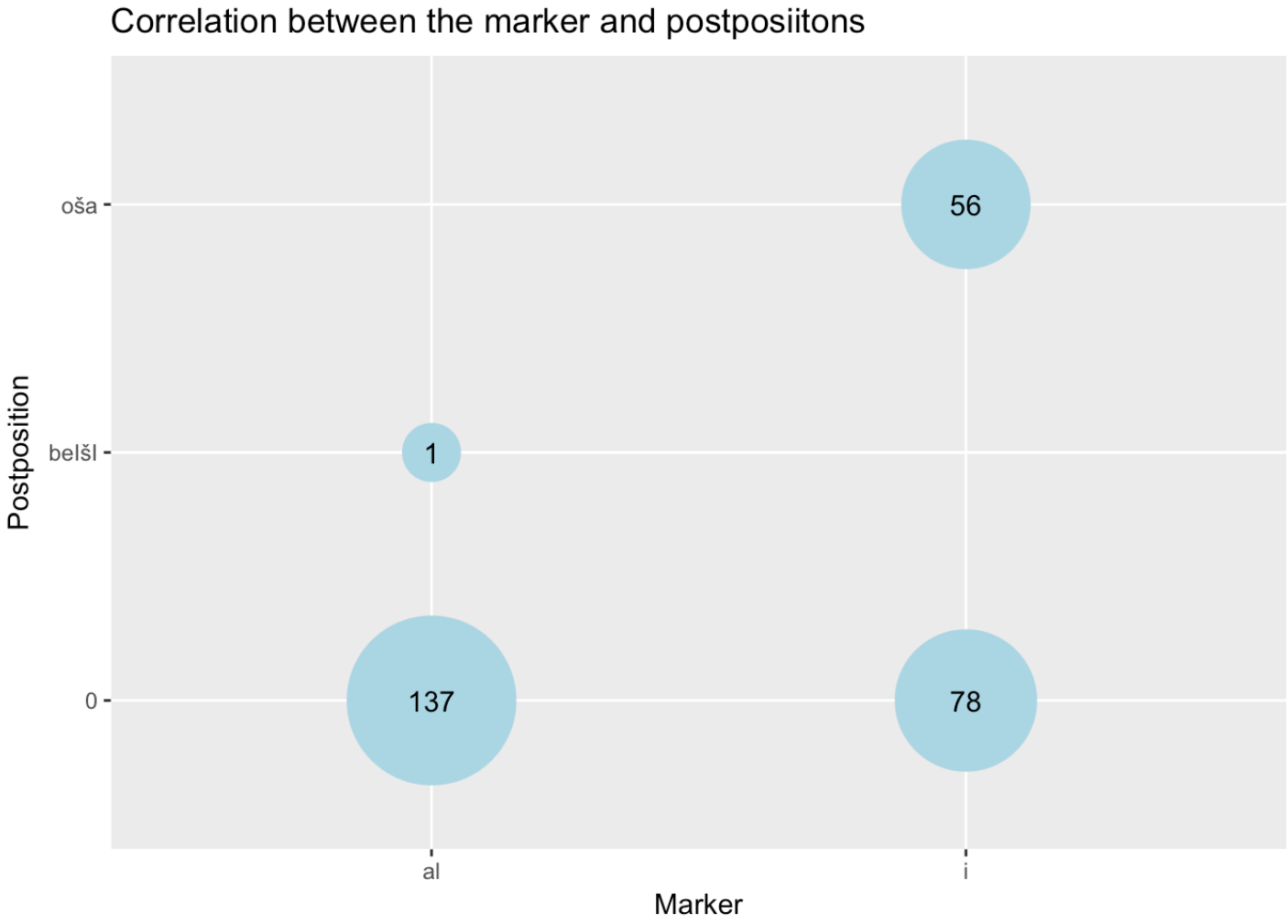


The distribution by the postpositions shows that oša is only possible with -i. We do not have enough data to make any judgement about belšI.



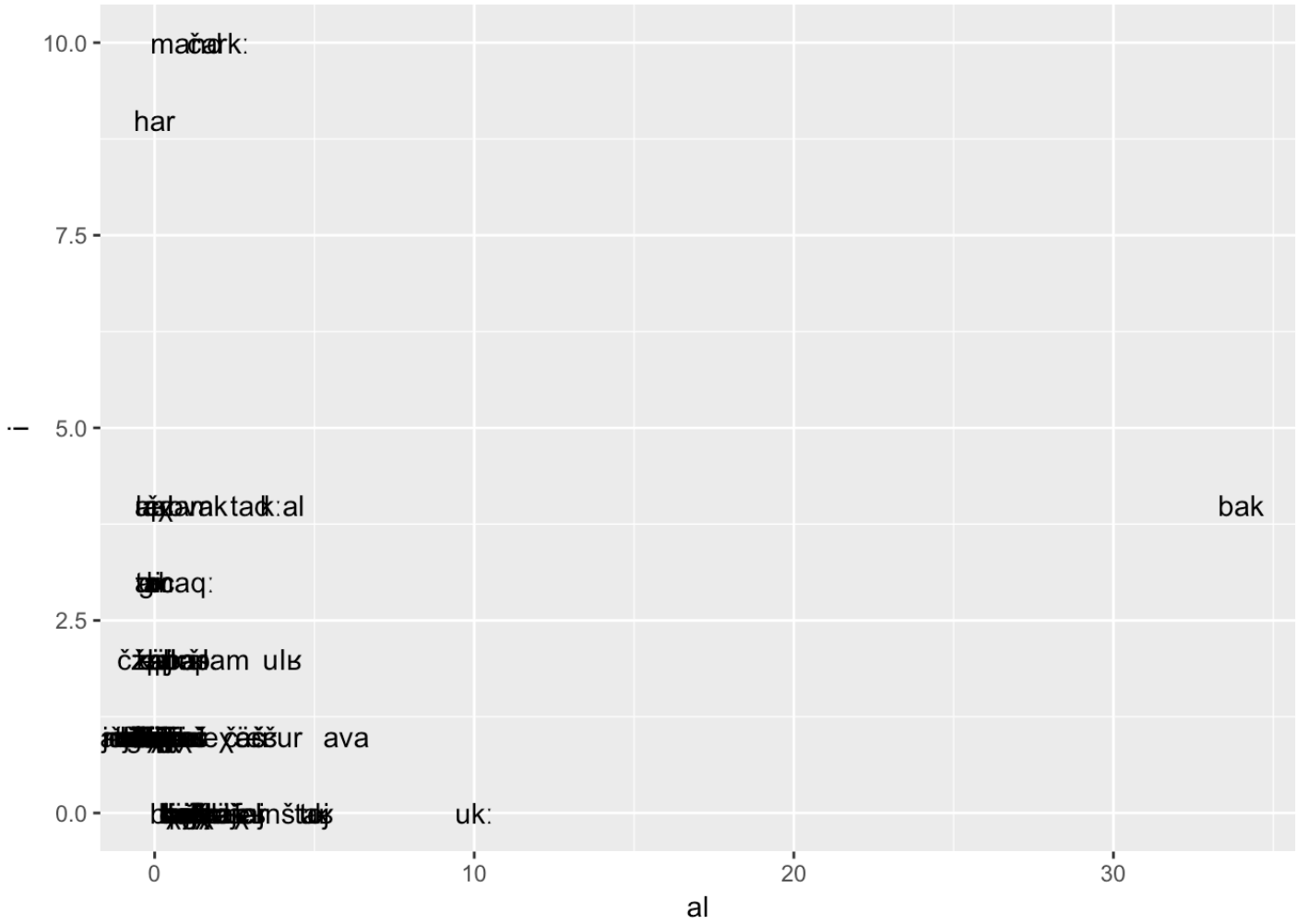
The same with a different type of plot:

```
aop %>%
  group_by(marker, postposition) %>%
  summarise(number = n()) %>%
  ggplot(aes(marker, postposition, label = number))+
  geom_point(aes(size = number), color = "light blue")+
  geom_text()+
  scale_size(range = c(10, 30))+
  guides(size = F)+
  xlab("Marker")+
  ylab("Postposition")+
  ggtitle("Correlation between the marker and postposiitons")
```



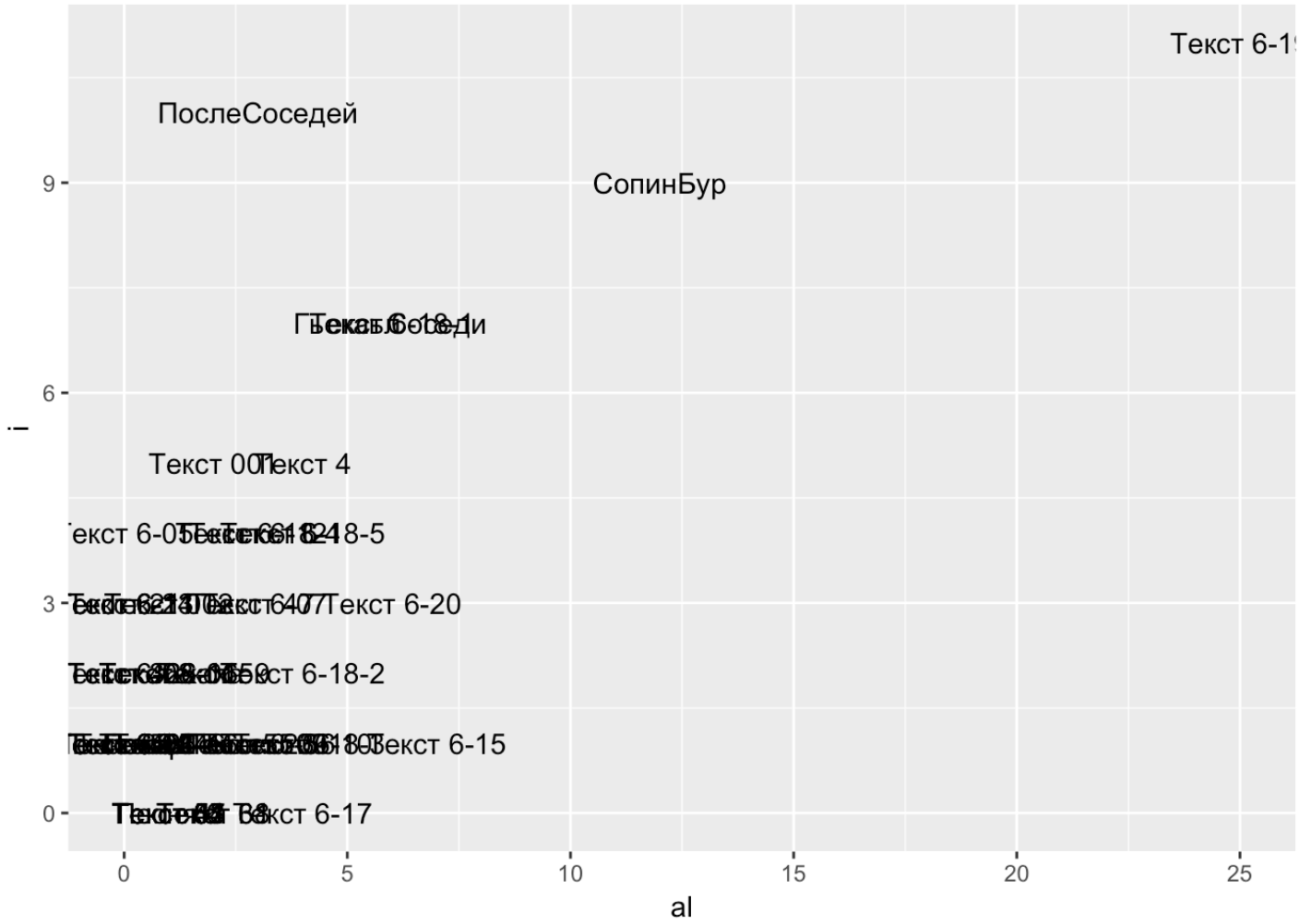
The following plot depicts how the verbs form clusters according to the use of the markers. The only verb that only attaches “-al” is uk: ‘say’. There are verbs that tend to attach -i: har ‘come’, mand ‘stay’, and čark: ‘to end/to cease’. Other verbs attach both suffixes. The graph also shows that a clearly stative verb bak ‘be’ is used with -al considerably more often than with -i.

```
aop_stem <- aop[,c(3,1)]
aop_stem_t <- table(aop_stem)
#View(aop_stem_t)
aop_stem_t <- as.data.frame(aop_stem_t)
aop_stem_t_al <- filter(aop_stem_t, marker == "al")
aop_stem_t_i <- filter(aop_stem_t, marker == "i")
aop_stem_tf <- cbind(aop_stem_t, aop_stem_t_al$Freq, aop_stem_t_i$Freq)
#View(aop_stem_tf)
aop_stem_tf <- aop_stem_tf[,c(1,4,5)]
aop_stem_tf <- aop_stem_tf[c(1:92),]
colnames(aop_stem_tf) <- c("stem", "al", "i")
aop_stem_tf %>%
  ggplot(aes(al, i, label=stem))+
  geom_text()
```



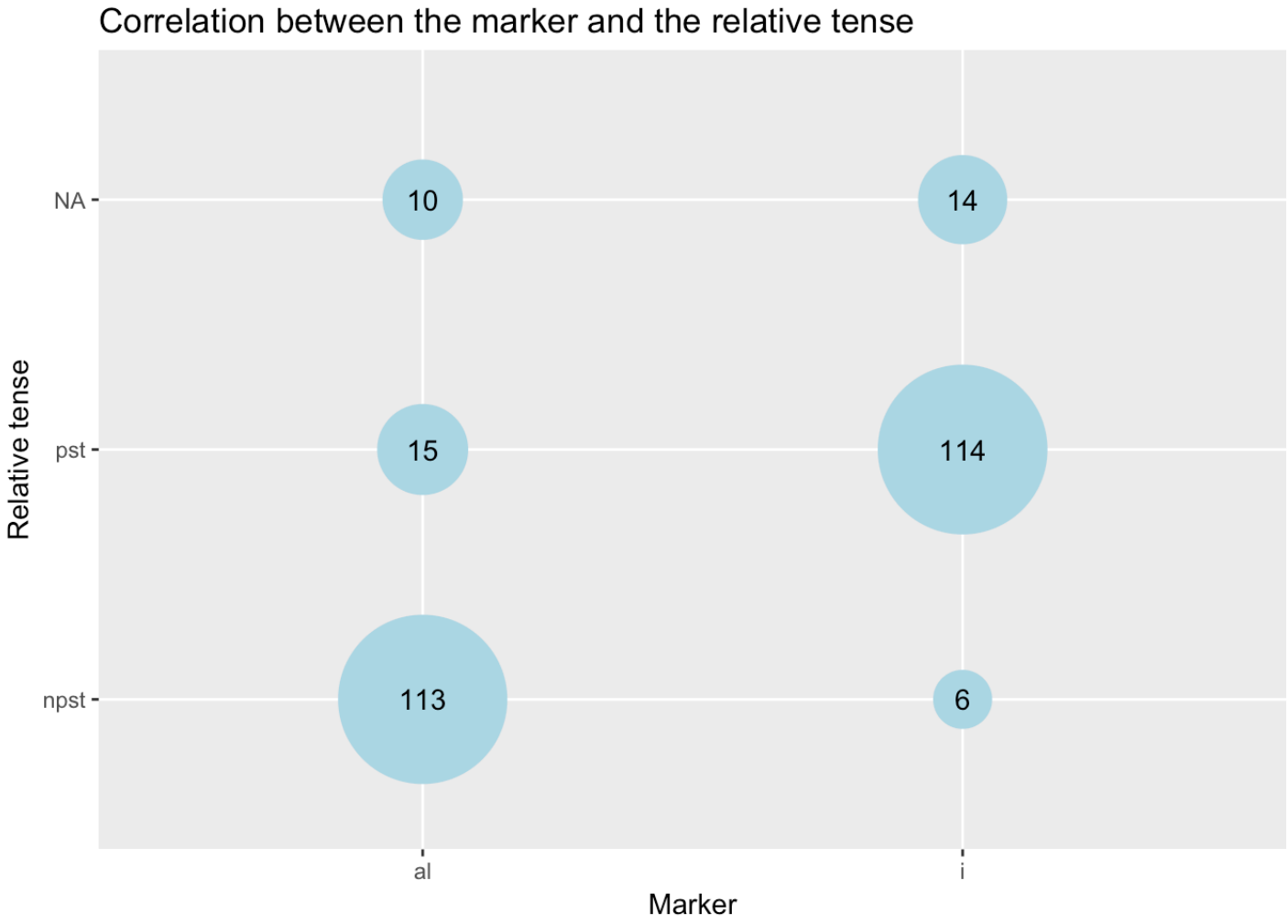
The next plot shows how texts form clusters according to the use of the markers. We do not see any correspondences between the types of the texts and the use of markers.

```
aop_text <- aop[,c(2,1)]
aop_text_t <- table(aop_text)
aop_text_t <- as.data.frame(aop_text_t)
aop_text_t_al <- filter(aop_text_t, marker == "al")
aop_text_t_i <- filter(aop_text_t, marker == "i")
aop_text_tf <- cbind(aop_text_t, aop_text_t_al$Freq, aop_text_t_i$Freq)
aop_text_tf <- aop_text_tf[,c(1,4,5)]
aop_text_tf <- aop_text_tf[c(1:54),]
colnames(aop_text_tf) <- c("text", "al", "i")
aop_text_tf %>%
  ggplot(aes(al, i, label=text))+
  geom_text()
```



The next plot shows how the markers are distributed by the relative tense. The majority of -al uses are in the ‘non-past’ clauses. The majority of -i uses are in the ‘past’ clauses. There are also some clauses that we marked NA. This means that we were indecisive and the relative tense could not be established undoubtedly. In these clauses the markers are distributed equally.

```
aop %>%
  group_by(marker, rtense) %>%
  summarise(number = n()) %>%
  ggplot(aes(marker, rtense, label = number))+
  geom_point(aes(size = number), color = "light blue")+
  geom_text()+
  scale_size(range = c(10, 30))+
  guides(size = F)+
  xlab("Marker")+
  ylab("Relative tense")+
  ggtitle("Correlation between the marker and the relative tense")
```

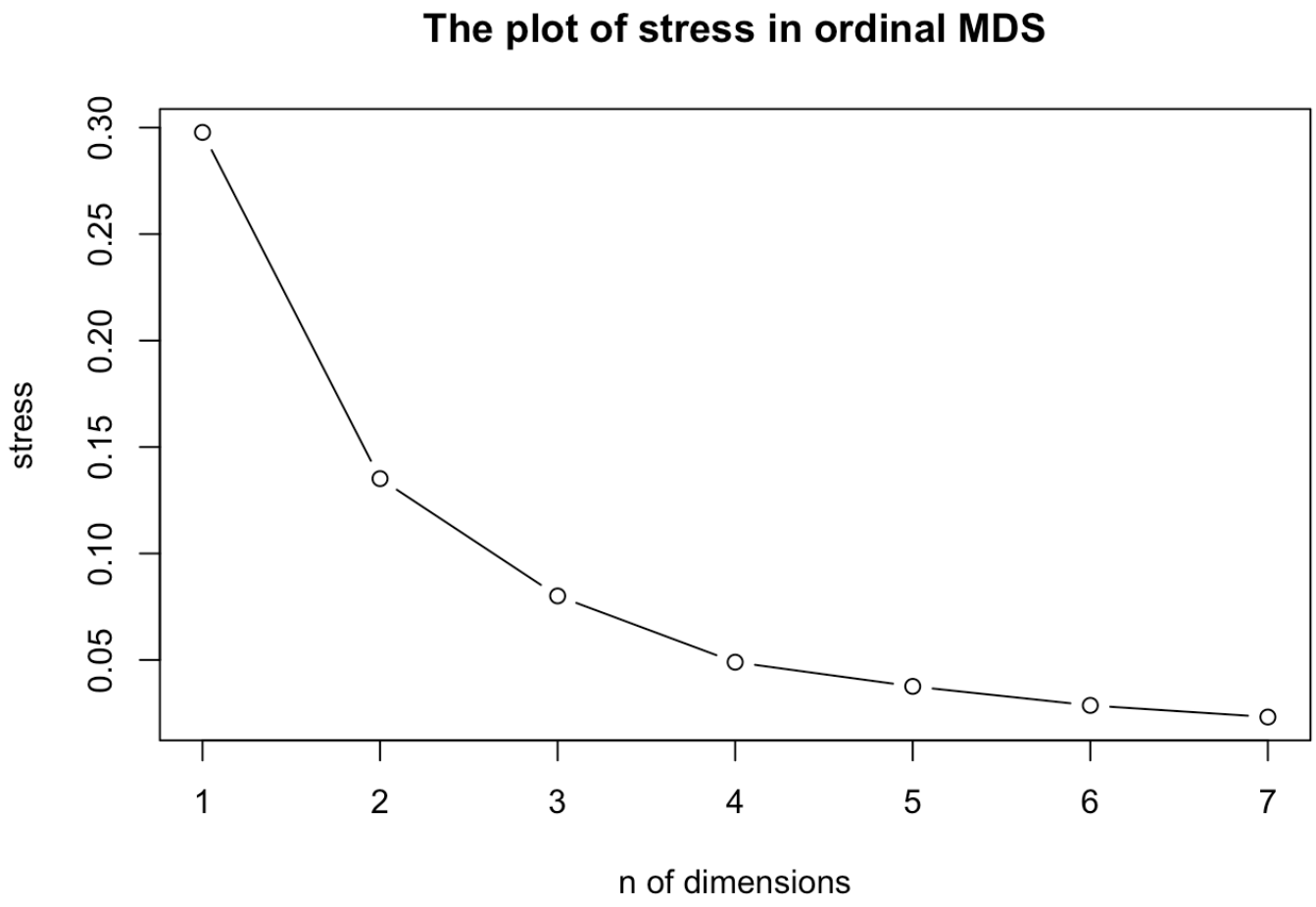


The MDS clusterization for our variables. This part is based on [Levshina 2015]. The plot of stress shows that 3 or 4 dimensions explain enough variance, and we age going to use 4 because the difference between the 3rd and 4th variables is very small.

```
aop.dist <- daisy(aop[,c(3,4,5,6,7,8,9)])
```

```
## Warning in daisy(aop[, c(3, 4, 5, 6, 7, 8, 9)]): binary variable(s) 5
## treated as interval scaled
```

```
stress <- sapply(1:7, function(x) smacofSym(aop.dist, type = "ordinal", ndim = x)$
stress)
plot(1:7, stress, type = "b", xlab = "n of dimensions", ylab = "stress", main = "T
he plot of stress in ordinal MDS")
```




```
aop.mds <- smacofSym(aop.dist, type = "ordinal", ndim = 4)
#plot(aop.mds$conf, main = "Exemplars of al and i: Dim 1 and 2")
#plot(aop.mds$conf[, 2:3], main = "Exemplars of al and i: Dim 2 and 3")
```

After interpreting the dimensions, it becomes clear that the most important predictors are: postposition, stem, rtense, and nominalization.

```
y <- aop.mds$conf[, 1]
adjR2 <- sapply(2:9, function(x) summary(lm(y ~ aop[, x]))$adj.r.squared)
res <- data.frame(colnames(aop[,-1]), adjR2)
View(res)
```

The negative coefficients of postposition = “oša” and postposition = “belšl” show that the examples with postpositions have lower values on Dimension 1 than the observations with the reference level (postposition = “0”).

```
summary(lm(y ~ postposition, data = aop))
```

```
##
## Call:
## lm(formula = y ~ postposition, data = aop)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.6733 -0.1564 -0.0293  0.1681  0.7516
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.18898    0.01943   9.725   <2e-16 ***
## postpositionbeIšI -0.48927    0.28561  -1.713   0.0879 .
## postpositionoša   -0.90918    0.04275 -21.267   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.285 on 269 degrees of freedom
## Multiple R-squared:  0.6276, Adjusted R-squared:  0.6249
## F-statistic: 226.7 on 2 and 269 DF,  p-value: < 2.2e-16
```

There is no need to interpret these values.

```
s <- summary(lm(y ~ stem, data = aop))
View(s$coefficients)
```

The negative coefficients of postposition = “pst” show that the examples with past relative tense have lower values on Dimension 3 then the observations with the reference level (rtense = “prs”).

```
summary(lm(y ~ rtense, data = aop))
```

```
##
## Call:
## lm(formula = y ~ rtense, data = aop)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.66069 -0.32710 -0.03588  0.28812  0.77233
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.36041    0.02809   12.83   <2e-16 ***
## rtensepst   -0.72772    0.03895  -18.69   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3064 on 246 degrees of freedom
##   (24 observations deleted due to missingness)
## Multiple R-squared:  0.5867, Adjusted R-squared:  0.585
## F-statistic: 349.2 on 1 and 246 DF,  p-value: < 2.2e-16
```

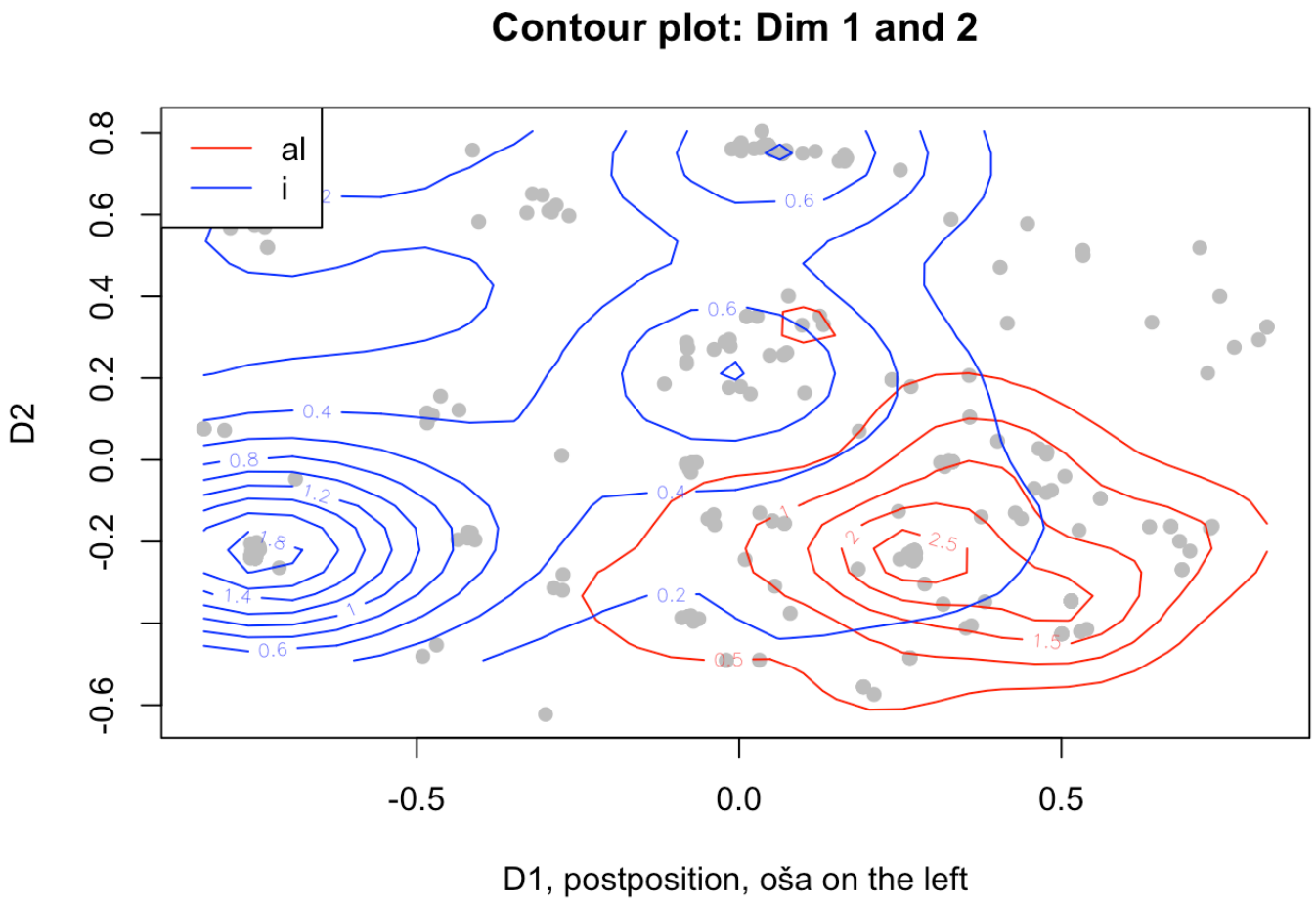
Nominalization. The examples with nominalization have lower values in D4 than the ones without it.

```
summary(lm(y ~ nominalization, data = aop))
```

```
##
## Call:
## lm(formula = y ~ nominalization, data = aop)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.71581 -0.31435 -0.03172  0.21279  0.75843
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.30230    0.02604   11.61   <2e-16 ***
## nominalization -0.67955    0.03904  -17.41   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3199 on 270 degrees of freedom
## Multiple R-squared:  0.5288, Adjusted R-squared:  0.5271
## F-statistic:   303 on 1 and 270 DF,  p-value: < 2.2e-16
```

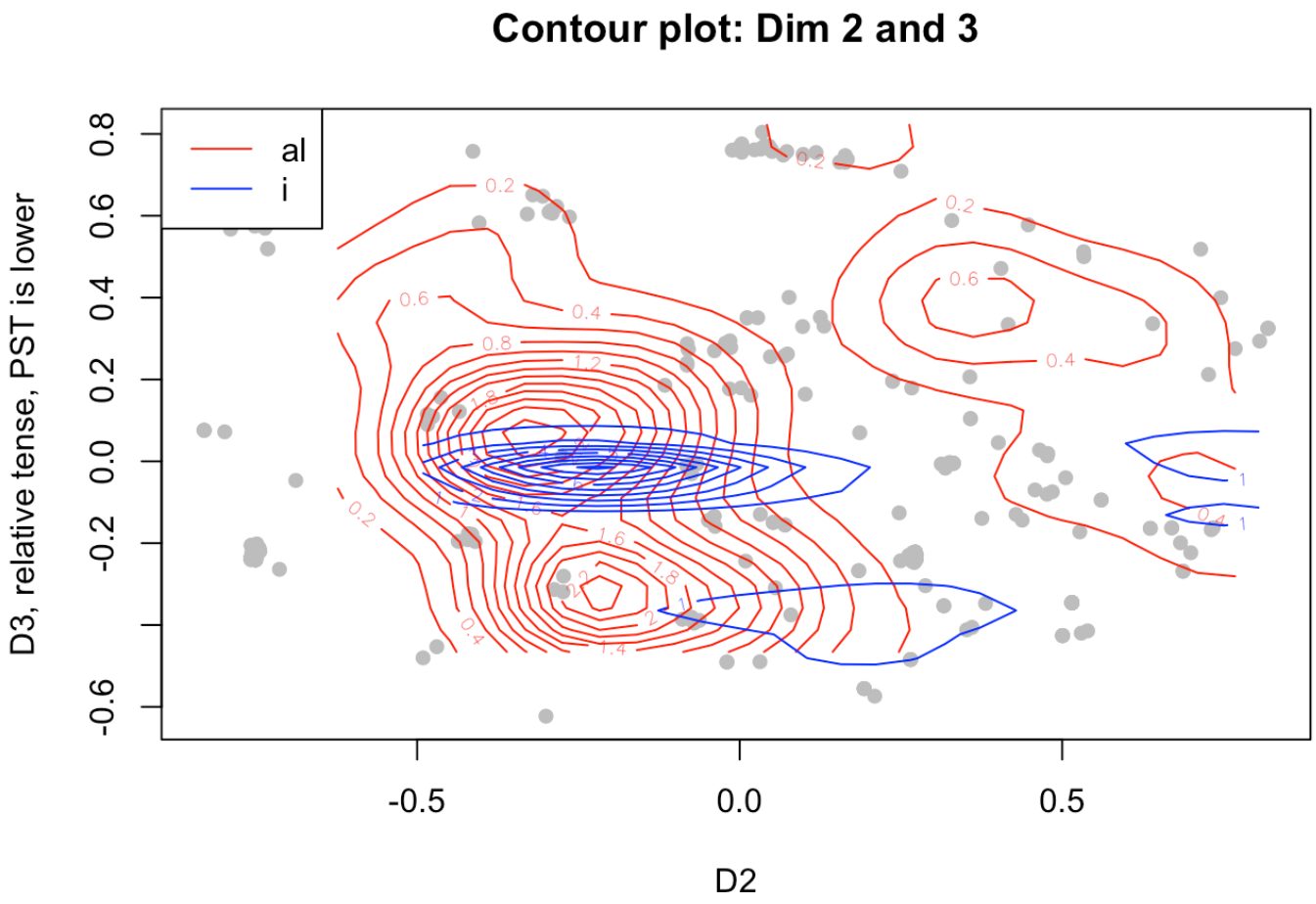
Plotting the semantic map, dimensions 1 and 2:

```
dens.al <- kde2d(aop.mds$conf[aop$marker == "al", 1],
                 aop.mds$conf[aop$marker == "al", 2])
dens.i <- kde2d(aop.mds$conf[aop$marker == "i", 1],
                  aop.mds$conf[aop$marker == "i", 2],)
plot(aop.mds$conf, main = "Contour plot: Dim 1 and 2", pch = 16, col = "grey", xla
b = "D1, postposition, oša on the left", ylab = "D2")
contour(dens.al, add = TRUE, col = "red")
contour(dens.i, add = TRUE, col = "blue")
legend("topleft", c("al", "i"), col = c("red", "blue"), lty = 1)
```



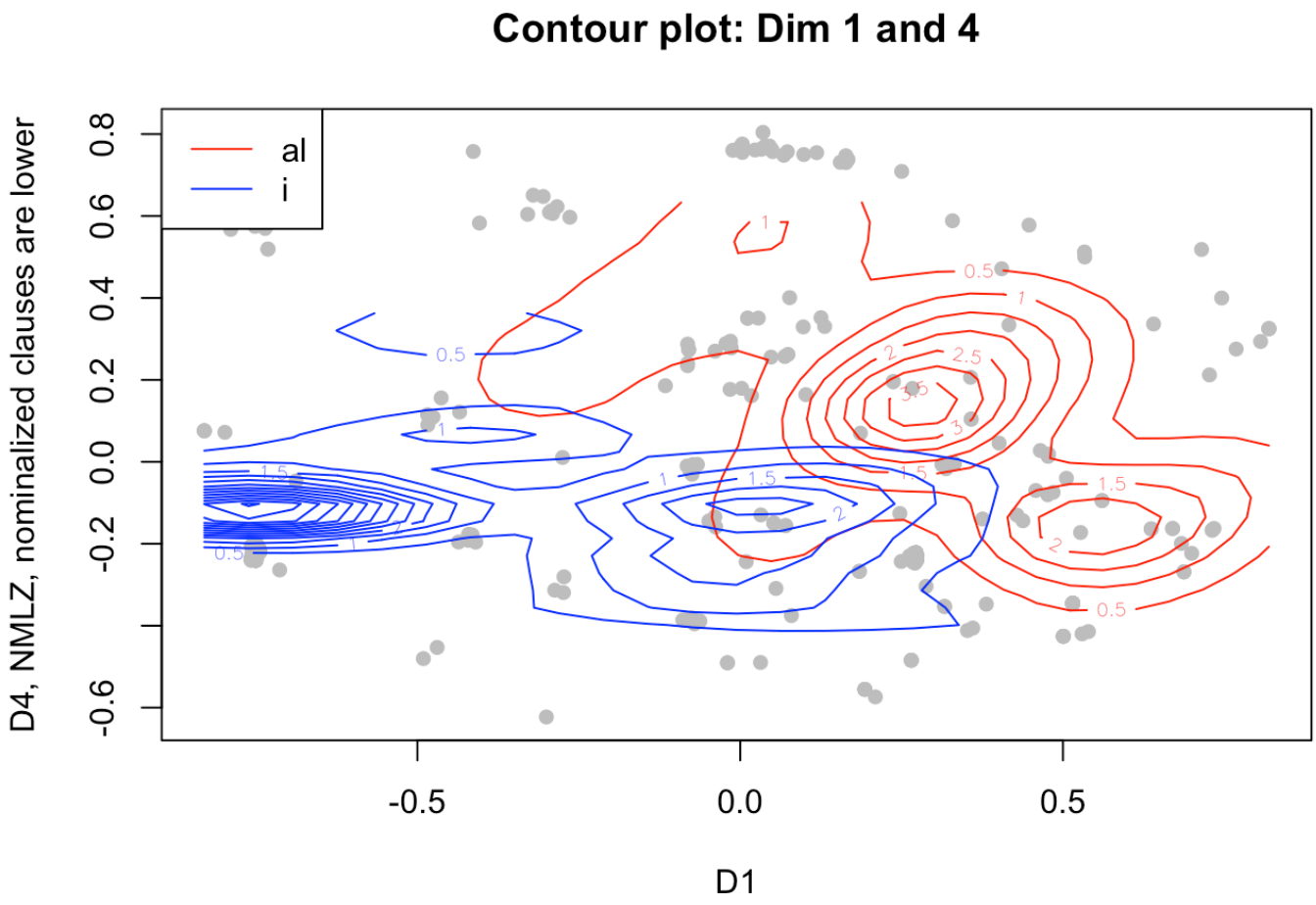
Dimensions 2 and 3.

```
dens.al <- kde2d(aop.mds$conf[aop$marker == "al", 2],
                  aop.mds$conf[aop$marker == "al", 3])
dens.i <- kde2d(aop.mds$conf[aop$marker == "i", 2],
                aop.mds$conf[aop$marker == "i", 3],)
plot(aop.mds$conf, main = "Contour plot: Dim 2 and 3", pch = 16, col = "grey", yla
b = "D3, relative tense, PST is lower", xlab = "D2")
contour(dens.al, add = TRUE, col = "red")
contour(dens.i, add = TRUE, col = "blue")
legend("topleft", c("al", "i"), col = c("red", "blue"), lty = 1)
```



The next chunk performs a GLM based on our predictors:

```
dens.al <- kde2d(aop.mds$conf[aop$marker == "al", 1],
                 aop.mds$conf[aop$marker == "al", 4])
dens.i <- kde2d(aop.mds$conf[aop$marker == "i", 1],
                aop.mds$conf[aop$marker == "i", 4],)
plot(aop.mds$conf, main = "Contour plot: Dim 1 and 4", pch = 16, col = "grey", yla
b = "D4, NMLZ, nominalized clauses are lower", xlab = "D1")
contour(dens.al, add = TRUE, col = "red")
contour(dens.i, add = TRUE, col = "blue")
legend("topleft", c("al", "i"), col = c("red", "blue"), lty = 1)
```



Now let us try to predict the use of the marker from all the other factors:

```
fit <- glm(marker~., data = aop[, -c(2, 3, 4)], family = "binomial")
options(scipen = 999)
all.comb <- aggregate(marker ~  stem + rtense + actionality + matrix_tense + nomin
alization + postposition + corpus, data = aop, length)
colnames(all.comb)[8] <- "number"
final <- cbind.data.frame(all.comb, as.data.frame(predict(fit, all.comb, type="res
ponse", se.fit = T)))
final$fit <- round(final$fit, 2)
final$se.fit <- round(final$se.fit, 2)
final %>%
  arrange(desc(fit)) ->
  final
```

```
lapply(aop[setdiff(names(aop), "marker")],
       function(x) chisq.test(x, aop$marker)$expected) -> expected_values
#expected_values
```

The chisq is applied only if the number is >5.

```
data.frame(params = names(expected_values),
           less_than_5 = sapply(1:length(expected_values),
                                function(x){
                                  sum(expected_values[[x]] < 5) > 0})) ->
  chi_vs_fisher
chi_vs_fisher
```

```
##           params less_than_5
## 1           text        TRUE
## 2           stem        TRUE
## 3         rtense       FALSE
## 4    actionality       FALSE
## 5   matrix_tense        TRUE
## 6 nominalization       FALSE
## 7   postposition        TRUE
## 8         corpus       FALSE
```

We do not perform the exact Fisher's test because for some unknown reason it runs out of memory and crashes the R session. Nonetheless, the number of predictors is low enough to just perform the chisq.

```
aop %>%
summarise_each(
funs(chisq.test(., aop$marker)$p.value),
-one_of("marker","text", "stem", "matrix_tense", "postposition")) ->
chisq.p.values
View(chisq.p.values)

#aop %>%
#summarise_each(
#funs(fisher.test(., aop$marker)$p.value),
#-one_of("marker","rtense", "actionality", "nominalization", "corpus")) ->
#fisher.p.values
#View(fisher.p.values)

chisq.p.values.adjust <- p.adjust(unlist(chisq.p.values), method = 'bonferroni')
chisq.p.values.adjust
```

```
##                                         rtense
## 0.00000000000000000000000000000000005446106
##                                    actionality
## 0.00005882598866897116108090615149706081865588
##                                  nominalization
## 0.00111691548364166079974901801818987223668955
##                                         corpus
## 1.00000000000000000000000000000000000000000000
```

```
#data.frame(st = unlist(chisq_statistic), df = unlist(chisq_df), p = #unlist(chisq
.p.values.adjust)) -> chisq_results
```

```
chisq.p.values.adjust < 0.05
```

```
##         rtense    actionality nominalization         corpus
##           TRUE           TRUE           TRUE          FALSE
```

Making the GLM. Relative tense and nominalization are significant.

```
aop$marker <- factor(aop$marker)
aop$text <- factor(aop$text)
aop$stem <- factor(aop$stem)
aop$rtense <- factor(aop$rtense)
aop$actionality <- factor(aop$actionality)
aop$matrix_tense <- factor(aop$matrix_tense)
aop$nominalization <- factor(aop$nominalization)
aop$postposition <- factor(aop$postposition)
aop$corpus <- factor(aop$corpus)
fit <- glm(marker~., data = aop[, -c(2, 3, 9)], family = "binomial")
summary(fit)
```

```
##
## Call:
## glm(formula = marker ~ ., family = "binomial", data = aop[, -c(2,
##     3, 9)])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2010  -0.2511  -0.1077   0.0001   2.7786
##
## Coefficients:
##                     Estimate  Std. Error z value       Pr(>|z|)
## (Intercept)          -3.50983     0.98895  -3.549       0.000387 ***
## rtensepst             5.54277     0.88348   6.274 0.000000000352 ***
## actionalitystat       0.71133     0.75796   0.938       0.347997
## matrix_tensecond      1.73496     2.70014   0.643       0.520520
## matrix_tensefut      17.53312 10754.01296   0.002       0.998699
## matrix_tensehort     18.36177  7377.82423   0.002       0.998014
## matrix_tenseimp       2.75636     1.52528   1.807       0.070745 .
## matrix_tensenvp       0.29637     1.07007   0.277       0.781806
## matrix_tenseperf     -0.32927     0.94491  -0.348       0.727490
## matrix_tensepot      -0.24281     2.07687  -0.117       0.906931
## matrix_tenseprs       0.06852     0.81678   0.084       0.933142
## matrix_tensepst      -1.34229     0.86540  -1.551       0.120884
## matrix_tensesubj      1.80459     2.05125   0.880       0.378993
## matrix_tenseterm    -16.05624 10754.01302  -0.001       0.998809
## nominalization1      -1.35218     0.63290  -2.136       0.032640 *
## postpositionbeIšI   -14.77258 10754.01301  -0.001       0.998904
## postpositionoša      18.88594  1438.49196   0.013       0.989525
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 334.89  on 241  degrees of freedom
## Residual deviance: 101.24  on 225  degrees of freedom
##   (30 observations deleted due to missingness)
## AIC: 135.24
##
## Number of Fisher Scoring iterations: 18
```

Preparing the data frame for plotting the S-curve.

```
options(scipen = 999)
all.comb <- aggregate(marker ~  rtense + actionality + matrix_tense + nominalizati
on + postposition, data = aop, length)
colnames(all.comb)[6] <- "number"
final <- cbind.data.frame(all.comb, as.data.frame(predict(fit, all.comb, type="res
ponse", se.fit = T)))
final$fit <- round(final$fit, 2)
final$se.fit <- round(final$se.fit, 2)
final %>%
  arrange(desc(fit)) ->
  final
#final
```

Plotting the S-curve. All combinations with npst values are grouped in the upper part, while the ones with the value pst can be found within the lower part of the graph. The combinations with postposition osa also form a group. By looking at the distribution of combinations on the vertical axis, we can conclude that it is the relative tense and postposition that clearly influence the choice of the marker.

```
final$rtense <- as.character(final$rtense)
final$actionality <- as.character(final$actionality)
final$matrix_tense <- as.character(final$matrix_tense)
final$nominalization <- as.character(final$nominalization)
final$postposition <- as.character(final$postposition)

final_table <- sapply(1:nrow(final), function(x){
  paste(final[x,1:5], collapse = "-")})

new_df <- data.frame(name = final_table,
                     p = final$fit,
                     se = final$se.fit)
new_df
```

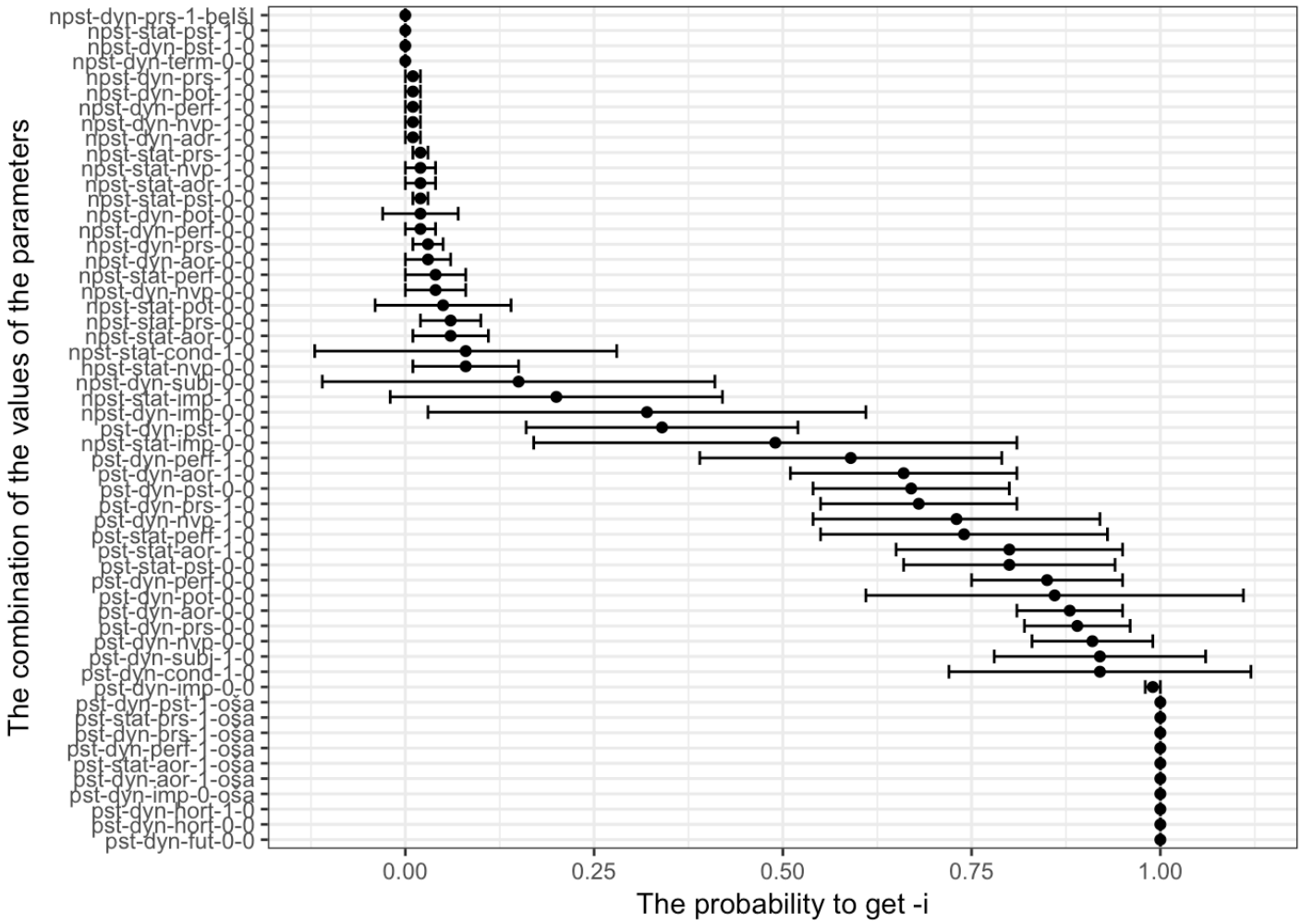
```
##                     name    p   se
## 1       pst-dyn-fut-0-0 1.00 0.00
## 2      pst-dyn-hort-0-0 1.00 0.00
## 3      pst-dyn-hort-1-0 1.00 0.00
## 4     pst-dyn-imp-0-oša 1.00 0.00
## 5     pst-dyn-aor-1-oša 1.00 0.00
## 6    pst-stat-aor-1-oša 1.00 0.00
## 7    pst-dyn-perf-1-oša 1.00 0.00
## 8     pst-dyn-prs-1-oša 1.00 0.00
## 9    pst-stat-prs-1-oša 1.00 0.00
## 10    pst-dyn-pst-1-oša 1.00 0.00
## 11      pst-dyn-imp-0-0 0.99 0.01
## 12     pst-dyn-cond-1-0 0.92 0.20
## 13     pst-dyn-subj-1-0 0.92 0.14
## 14      pst-dyn-nvp-0-0 0.91 0.08
## 15      pst-dyn-prs-0-0 0.89 0.07
```

```
## 16      pst-dyn-aor-0-0 0.88 0.07
## 17      pst-dyn-pot-0-0 0.86 0.25
## 18     pst-dyn-perf-0-0 0.85 0.10
## 19     pst-stat-pst-0-0 0.80 0.14
## 20     pst-stat-aor-1-0 0.80 0.15
## 21    pst-stat-perf-1-0 0.74 0.19
## 22      pst-dyn-nvp-1-0 0.73 0.19
## 23      pst-dyn-prs-1-0 0.68 0.13
## 24      pst-dyn-pst-0-0 0.67 0.13
## 25      pst-dyn-aor-1-0 0.66 0.15
## 26     pst-dyn-perf-1-0 0.59 0.20
## 27    npst-stat-imp-0-0 0.49 0.32
## 28      pst-dyn-pst-1-0 0.34 0.18
## 29     npst-dyn-imp-0-0 0.32 0.29
## 30    npst-stat-imp-1-0 0.20 0.22
## 31    npst-dyn-subj-0-0 0.15 0.26
## 32    npst-stat-nvp-0-0 0.08 0.07
## 33   npst-stat-cond-1-0 0.08 0.20
## 34    npst-stat-aor-0-0 0.06 0.05
## 35    npst-stat-prs-0-0 0.06 0.04
## 36    npst-stat-pot-0-0 0.05 0.09
## 37     npst-dyn-nvp-0-0 0.04 0.04
## 38   npst-stat-perf-0-0 0.04 0.04
## 39     npst-dyn-aor-0-0 0.03 0.03
## 40     npst-dyn-prs-0-0 0.03 0.02
## 41    npst-dyn-perf-0-0 0.02 0.02
## 42     npst-dyn-pot-0-0 0.02 0.05
## 43    npst-stat-pst-0-0 0.02 0.01
## 44    npst-stat-aor-1-0 0.02 0.02
## 45    npst-stat-nvp-1-0 0.02 0.02
## 46    npst-stat-prs-1-0 0.02 0.01
## 47     npst-dyn-aor-1-0 0.01 0.01
## 48     npst-dyn-nvp-1-0 0.01 0.01
## 49    npst-dyn-perf-1-0 0.01 0.01
## 50     npst-dyn-pot-1-0 0.01 0.01
## 51     npst-dyn-prs-1-0 0.01 0.01
## 52    npst-dyn-term-0-0 0.00 0.00
## 53     npst-dyn-pst-1-0 0.00 0.00
## 54    npst-stat-pst-1-0 0.00 0.00
## 55 npst-dyn-prs-1-beIšI 0.00 0.00
```

```
new_df %>%
  arrange(desc(p)) -> new_df
new_df$name %>%
  unlist() %>%
  unname() ->
  levels

new_df$name <- factor(new_df$name, levels = levels)

new_df %>%
  mutate(min = p-se,
         max = p+se) %>%
  ggplot(aes(name, p)) +
  theme_bw() +
  geom_point() +
  geom_errorbar(aes(ymin = min, ymax = max)) +
  coord_flip() +
  ylab("The probability to get -i") +
  xlab("The combination of the values of the parameters")
```



Conclusions. Thus, the temporal hypothesis was confirmed, which does not exclude the possibility of perfective vs. imperfective aspect that we did not check within this study. Other factors that influence the choice of the marker are postposition, stem, and nominalization. The latter two influence the choice to a slightly lesser extent. The significant impact of postposition can also be explained by the relative tense hypothesis, as these postpositions exactly designate the order of events (‘before’ or ‘after’). The interpretation of the influence of the type of the stem and presence of nominalization is more complicated. We assume that each stem has its own intrinsic aspectual properties (lexical aspect) that may have an impact on the marker choice. However, not much is known about different lexical aspectual classes in Udi, so this hypothesis cannot be checked based on the current knowledge. If lexical aspect indeed plays a role, the relation is more complicated than just the opposition between stative and dynamic verbs, we proposed at the beginning of our investigation. The factor of nominalization is also hard to interpret because the nominalization patterns and properties have not yet been studied. We leave these issues open for the further research.