Semantic alignments in number words

Bill Thompson, Seán Roberts & Gary Lupyan

Contents

Introduction **47**
Variables: 47

Load libraries, graphing theme **50**

Load data **50**

Overview **53**
Numeric value 53
Number line 57
Variation by language 58
Variation by frequency 60

Decision tree **61**

Run a GAM **65**
Summary of main model 68
Controlling for linguistic history 70

Conclusion **73**

References **73**

Introduction

What predicts the semantic alignment of number words?

Although translations may be more direct for number words, there are still differences between languages regarding their semantic associations. Data was available for various analyses of numerals in 16 langauges (mainly constrained by the data from Calude & Verkerk, 2016). These were anlaysed using binary decision trees to find coherent clusters then a Generalised Additive Model to explain differences in alignment.

Variables:

- *l1*: Iso2 code for language 1
- *l2*: Iso2 code for language 2
- *wordform_l1*: Orthographic word form for language 1
- *wordform_l2*: Orthographic word form for language 2
- *local_alignment*: Local semantic alignment
- *global_alignment*: Global semantic alignment
- *freq_l1*: Frequency of orthographic form in l1
- *freq_l2*: Frequency of orthographic form in l1
- *neighbour_overlap*: Number of neighbours in common

- *global_density_l1*: Density
- *global_density_l2*: Density
- *local_density_l1*: Density
- *local_density_l2*: Density
- *editdistance*: Edit distance between orthographic forms
- *k*: Parameter for number of neighbours (constant at k=100).
- *n*: Number of possible comparisons
- *number*: Number concept
- *number_numeric*: Numeric value of number
- *name_l1*: Name of language l1
- *family_l1*: Family of language l1
- *name_l2*: Name of language l2
- *family_l2*: Family of language l2
- *same.family*: Compared languages are part of same family?
- *hist.dist*: Historical distance according to phylogenetic tree.
- *l1_typology*: Numeral typology for L1 according to Calude & Verkerk (2016) (see the section on "Number line" below).
- *l2_typology*: Same as above for L2.
- *sameNumeralTypology*: Comparison of numeral typology. This is 1 if the typology is the same in L1 and L2 (i.e. both numbers are atoms or use the same composition), and zero if the typology is different.
- *seven*: True if the numeral is 7. (see below)
- *homophone*: True if the form of the word has an alternative referential class in the North Euralex database. This includes mainly homophones (unrelated meanings) but also some synonyms (words with different but related meanings). For legacy reasons, the variable is named 'homophone'.
- *freqDiff*: Difference in orthographic frequency (absolute, log scale)

The Estimated Degrees of Freedom (EDF) is an indication of how non-linear a smooth term is (higher = less linear, see e.g. Wood, 2008). In general, a curve with an EDF of around 2 will look like a quadratic curve, and an EDF of around 3 will look like a cubic curve. However, this does not have to be the case: a smooth term could have a strong linear term, and a very weak non-linear term. The EDF captures this possibility as a continuous value. The simplest way to actually assess the smooth term is to plot it.

Random effects in the GAM implementation we use are treated just like a smooth term with the identity matrix as the penalty coefficient matrix. When entering a language pair as a random (intercept) effect, coefficients are created for each pair, modelled as independent and identically distributed normal random variables. The values are defined as discrete points along a smooth function. So, just like in a mixed effects model, the predicted alignment can be adjusted by a random intercept (the coefficients), e.g. the model can represent the alignment between English and French as higher overall, and the alignment between English and Bulgarian as being slightly lower etc. Stronger differences between levels of the random effect would need be represented by more complex functions, which would be penalised (similar to how a linear mixed effect model penalises random effect coefficient estimates which deviate from a normal distribution). The EDF value for the random effects relates to the 'wiggliness' of these coefficients when plotted in a regular space. This makes the EDF difficult to interpret. A random effect where there were no differences between levels would have an EDF of 1 (a flat line), but it would also be 1 when there were consistent distances between each level. So a high EDF would indicate something like an imbalance in the distribution of coefficients, e.g. a range of values that does not fit a normal distribution. In fact, there are several language pairs with lower alignment (pairs from different language families), and few with very high alignment (possibly a ceiling effect).

See the SI of Monaghan & Roberts (2019) for further explanation of EDF and random effects applied to linguistic data.

Monaghan & Roberts (2019) Cognitive influences in language evolution: Psycholinguistic predictors of loan word borrowing. *Cognition*, 186, 147-158.

Wood, S. N. (2008). Fast stable direct fitting and smoothness selection for generalized additive models.

Journal of the Royal Statistical Society: Series B (Statistical Methodology), 70(3), 495-518.

Load libraries, graphing theme

```
library(mgcv) # for gam
library(lmtest) # for model comparison
source("GAM_derivaties.R") # for derivitives plot
library(lme4)
library(tidyverse)
#library(langcog)
library(boot)
library(ggplot2)
library(lazyeval)
library(data.table)
library(MuMIn)
library(REEMtree)
library(rpart)
library(rpart.plot)
library(gridExtra)
library(grid)
library(gridBase)


myThemeBasic =
  theme_bw()+
  theme(panel.grid.minor=element_blank(),
        panel.grid.major=element_blank(),
        panel.background=element_blank())+
  theme(axis.text.x=element_text(size=13),
        axis.text.y=element_text(size=13),
        axis.title.x=element_text(size=12),
        axis.title.y=element_text(size=12))+
  theme(legend.background = element_rect(fill="transparent"))+
  theme(legend.text = element_text(size=13))+
  theme(legend.title = element_text(size=13))+
  theme(axis.title.y=element_text(vjust=0.9,size=20))+
  theme(axis.title.x=element_text(vjust=0.9,size=20))+
  theme(axis.title.y=element_text(vjust=0))+
  theme(axis.title.x=element_text(vjust=0))
```

Load data

```
PATH <- "../../data/numbers"
numbers <- read.csv(file.path(PATH,"number-alignments.csv"),
                    encoding = "UTF-8",fileEncoding = "UTF-8") %>%
    left_join(read.csv(file.path(PATH,"word_to_number.csv")),by=c("Number"="number"))

data.table::setnames(numbers,tolower(names(numbers))) #lowercase column names for consistency

language_info <- read.csv("../../data/FAIR_langauges_glotto_xdid.csv",
                          encoding = "UTF-8",fileEncoding = "UTF-8")

numbers <- left_join(numbers,select(language_info,Language,family,iso2),by=c("l1"="iso2"))
```

```
numbers <- numbers %>% rename(name_l1=Language,family_l1=family)


numbers <- left_join(numbers,select(language_info,Language,family,iso2),by=c("l2"="iso2"))

numbers <- numbers %>% rename(name_l2=Language,family_l2=family)

irreg_in_danish = c(50,60,70,80,90)
numbers <- numbers %>% mutate(
  same.family = family_l1==family_l2,
    is_danish = l1=="da" | l2=="da",
    is_single = (number_numeric < 10),
    is_decade = (number_numeric < 100 & number_numeric %% 10 ==0),
    is_hundred = (number_numeric == 100),
    is_thousand = (number_numeric == 1000),
    irreg_in_danish = (number_numeric %in% irreg_in_danish),
    lang_pair = paste(pmax(l1,l2),pmin(l1,l2),sep="-"))

histDistance = read.csv("../../data/trees/IndoEuropean_historical_distances_long.csv",
                        stringsAsFactors = F,encoding = "UTF-8",fileEncoding = "UTF-8")
numbers$hist.dist = histDistance[match(paste(numbers$name_l1,numbers$name_l2),
                            paste(histDistance$Var1,histDistance$Var2)),]$value

# Refactorise to get rid of non-existant categories
numbers$family_l1 = factor(numbers$family_l1)
numbers$family_l2 = factor(numbers$family_l2)

# Add typology data
cnv = read.csv("../../data/numbers/Calude_Verkerk_NumberData.csv",stringsAsFactors = F)
numbers <- left_join(numbers,cnv,by=c('l1','l2','number_numeric'))

numbers$isUralic = numbers$family_l1 == "Uralic" | numbers$family_l2 == "Uralic"
numbers$danish_irregular = (numbers$number_numeric %in% irreg_in_danish) &
                                    (numbers$name_l1=="Danish" | numbers$name_l2=="Danish")

# historic distance, setting Uralic languages to maximum
numbers$hist.dist2 = numbers$hist.dist
numbers$hist.dist2[is.na(numbers$hist.dist2)] = max(numbers$hist.dist,na.rm = T)
numbers$seven = numbers$number_numeric==7

#some rows got duplicated; remove
numbers <- distinct(numbers)

# Data on homophones
h = read.csv("../../data/numbers/NumberHomophones.csv",
             stringsAsFactors = F,encoding = "UTF-8",fileEncoding = "UTF-8")
h$code = paste(h$l,h$word)
numbers$homophone = (paste(numbers$l1,numbers$wordform_l1) %in% h$code) |
                    (paste(numbers$l2,numbers$wordform_l2) %in% h$code)

numbers$lang_pair.f = factor(numbers$lang_pair)

# Frequency difference (already in log scale)
```

```
numbers$freqDiff = abs(numbers$freq_l1-numbers$freq_l2)
# 78 frequency observations (3%) are missing, so impute:
freqM = bam(I(1+freqDiff)~
              #s(number_numeric) +
              s(lang_pair.f,bs='re') +
              s(editdistance) +
              s(hist.dist2),
            family = Gamma(link="identity"),
            data = numbers[!is.na(numbers$freqDiff),])
freqMPred = predict(freqM,newdata=numbers)-1
#plot(freqMPred,numbers$freqDiff)
#abline(0,1)
numbers[is.na(numbers$freqDiff),]$freqDiff =
  freqMPred[is.na(numbers$freqDiff)]
```

Group data by language:

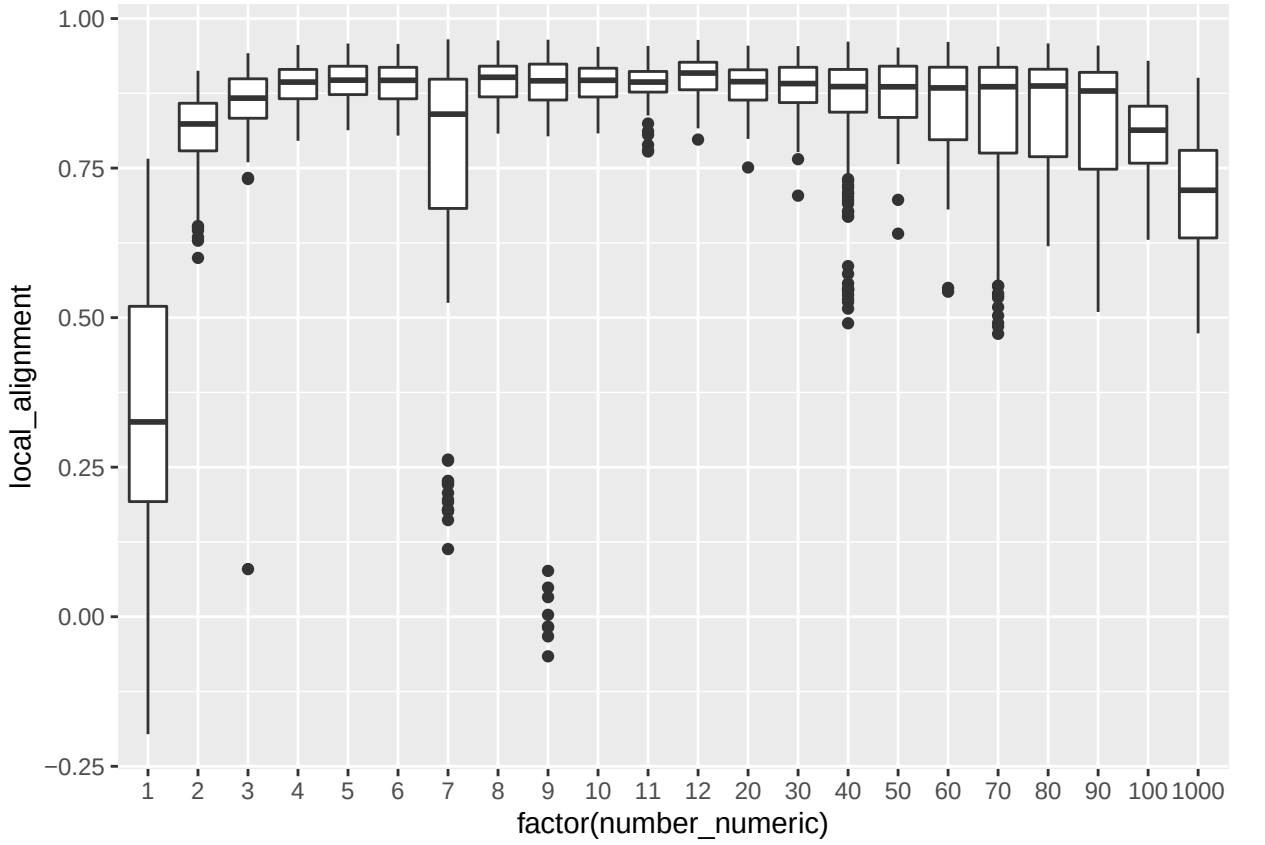
```
langAverages = data.frame()
for(l in unique(c(numbers$l1,numbers$l2))){
  dx = numbers[numbers$l1==l | numbers$l2==l,]
  langAverages = rbind(langAverages,
                  data.frame(
                   local_alignment = dx$local_alignment,
                    number_numeric = dx$number_numeric,
                    l = l,
                    l1 = dx$name_l1,
                    l2 = dx$name_l2))
}
langAverages$l = factor(langAverages$l,
     levels = names(sort(tapply(langAverages$local_alignment,langAverages$l,mean))))
```

Overview

Numeric value

Plot by numeric value:

```
ggplot(numbers,aes(x=factor(number_numeric),y=local_alignment)) + geom_boxplot()
```



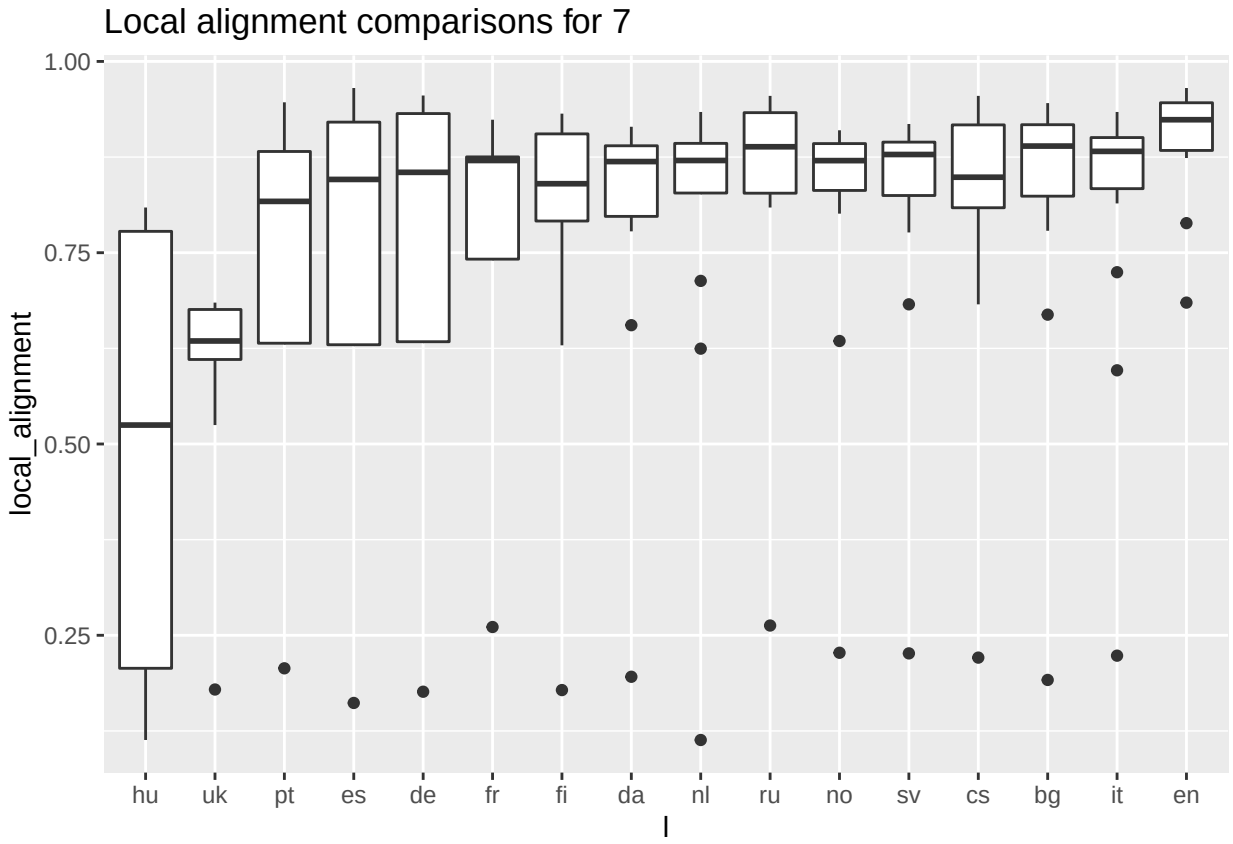
- 1 has a low numeric alignment.
- Alignment rises 2-3.
- 7 has a lower alignment.
- Drop in alignment for 100, 1000.
- Outliers for 7 and 9.
- Outliers for 40 and 70?

1 and 2 are often sources for grammaticalised indefinite/duel markers (Givón, 1981).

The slight decline from 10 to 1000 could be due to the declining frequency of occurrences of these numbers (Dehaene & Mehler, 1992), which might affect convergence on meanings, but more directly would affect the co-occurrence statistics.

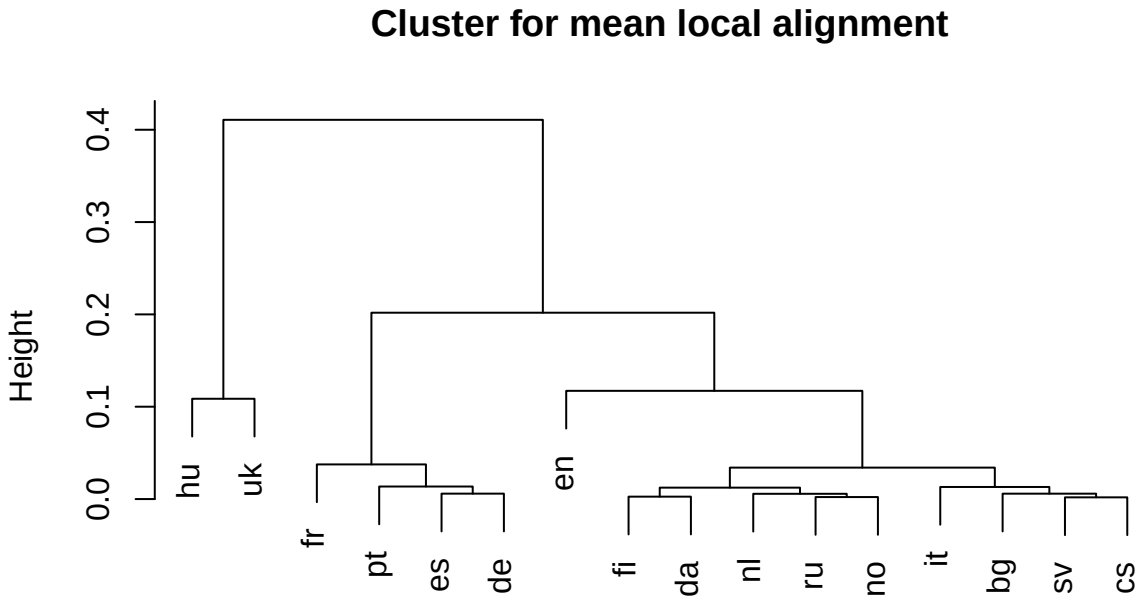
What's driving the difference with 7 and 9? 7 might be linked to there being 7 days in the week, so semantic differences in time might be reflected. Let's look at individual languages:

```
la7 = langAverages[langAverages$number_numeric==7,]
la7$l = factor(la7$l,levels=names(sort(tapply(
         la7$local_alignment,la7$l,mean))))
ggplot(la7,aes(x=l,y=local_alignment)) +
  geom_boxplot() +
  ggtitle("Local alignment comparisons for 7")
```



The plot above shows that the numeral 7 has lower alignment when it involves a comparison with Hungarian or Ukrainian. For example, all the outliers around 0.25 are comparisons with Hungarian. Also, the means are clearly different from the other languages, as shown by hierarchical clustering:

```
hc = hclust(dist(sort(tapply(la7$local_alignment,la7$l,mean))))
plot(hc,main="Cluster for mean local alignment",xlab="",sub = "")
```



This might be because Hungarian is a Uralic language, but maybe also because the Hungarian word for '7' also directly means "week". Ukrainian is also low. We note that forms for 7 and 8 are very similar in

Ukrainian (7 = sim and 8 = visim).

What are the outliers for 9? These all include comparisons to French:

```
numbers[numbers$number_numeric==9 & numbers$local_alignment<0.25,c("l1",'l2',"local_alignment")]
```

```
##      l1 l2 local_alignment
## 1    fr uk     0.003192686
## 483  fr da    -0.016457496
## 786  fr cs     0.076614888
## 903  fr bg    -0.032815537
## 1231 fr fi     0.032734709
## 1366 fr no    -0.066181331
## 1436 fr sv     0.048704798
## 2401 fr ru    -0.017702025
```

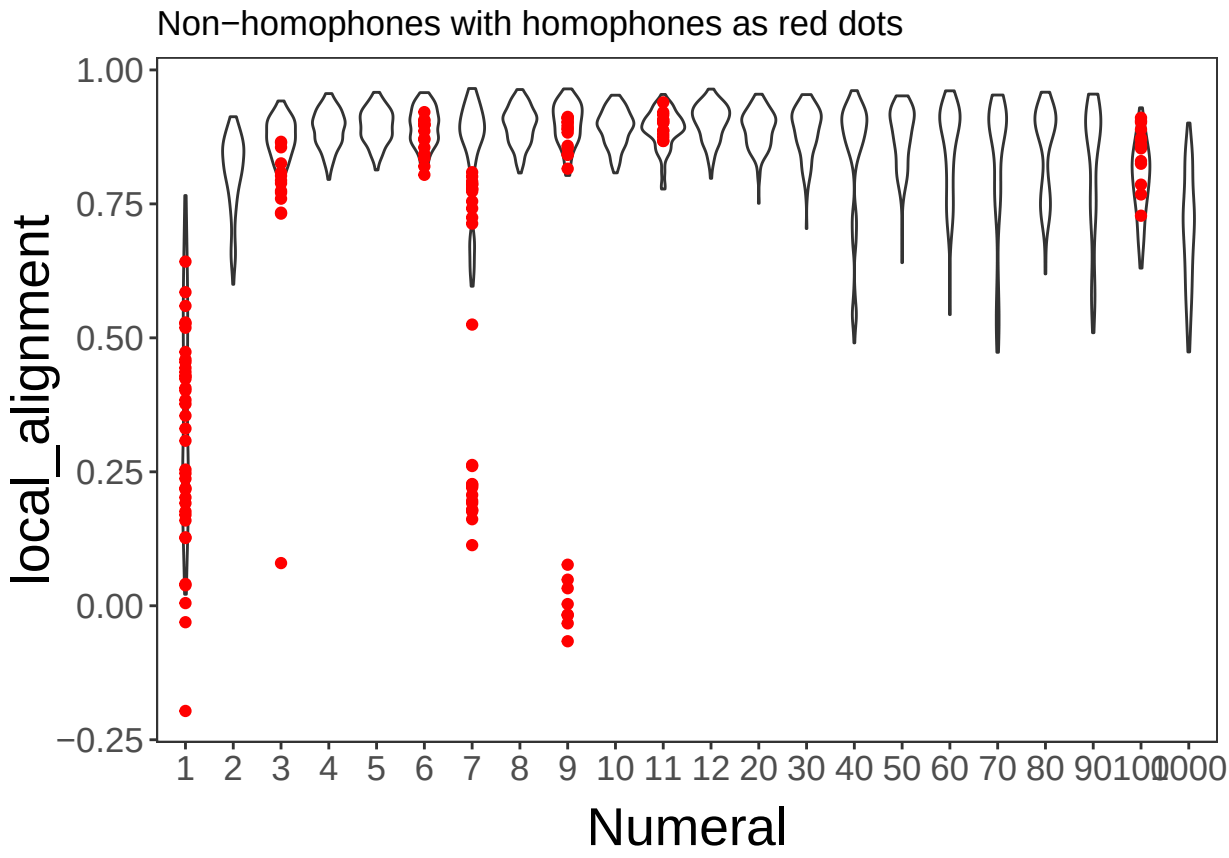
This may be because French 9 ("neuf") can mean '9' or "new". We used the North Euralex dictionary to find numbers which have alternative referential classes (see the file `NumberHomophones.csv`). Some of these will have referential classes with no related meanings (e.g. Norwegian 'tre' meaning '3' or 'tree'), and are thus homonyms. Some of these cases will be polysemous (the different meanings are related), as in the case of Hungarian 'hét' meaning '7' or 'week' (see below).

```
h[,c("l","number","otherMeanings")]
```

```
##    l number   otherMeanings
## 1 no      3 Baum::N;Holz::N
## 2 sv     11      zittern::V
## 3 fr    100         Blut::N
## 4 fr      9          neu::A
## 5 uk      1     allein::ADV
## 6 ru      1     allein::ADV
## 7 fi      6        Tanne::N
## 8 hu      7        Woche::N
```

The plot below shows the distribution of non-homophones, with homophones drawn as dots. For 7 and 9, these fall outside of the general distribution, but there are several other cases where homophones look similar to the rest of the distribution:

```
ggplot(numbers[!numbers$homophone,],
       aes(y=local_alignment,
           x=factor(number_numeric))) +
  geom_violin() +
  geom_point(data=numbers[numbers$homophone,],
             aes(y=local_alignment,
           x=factor(number_numeric)),
           colour="red") +
  myThemeBasic +
  xlab("Numeral") +
  ggtitle("Non-homophones with homophones as red dots")
```


Non−homophones with homophones as red dots
local_alignment
1.00
0.75
0.50
0.25
0.00
−0.25
1 2 3 4 5 6 7 8 9 10 11 12 20 30 40 50 60 70 80 90 100 1000
Numeral

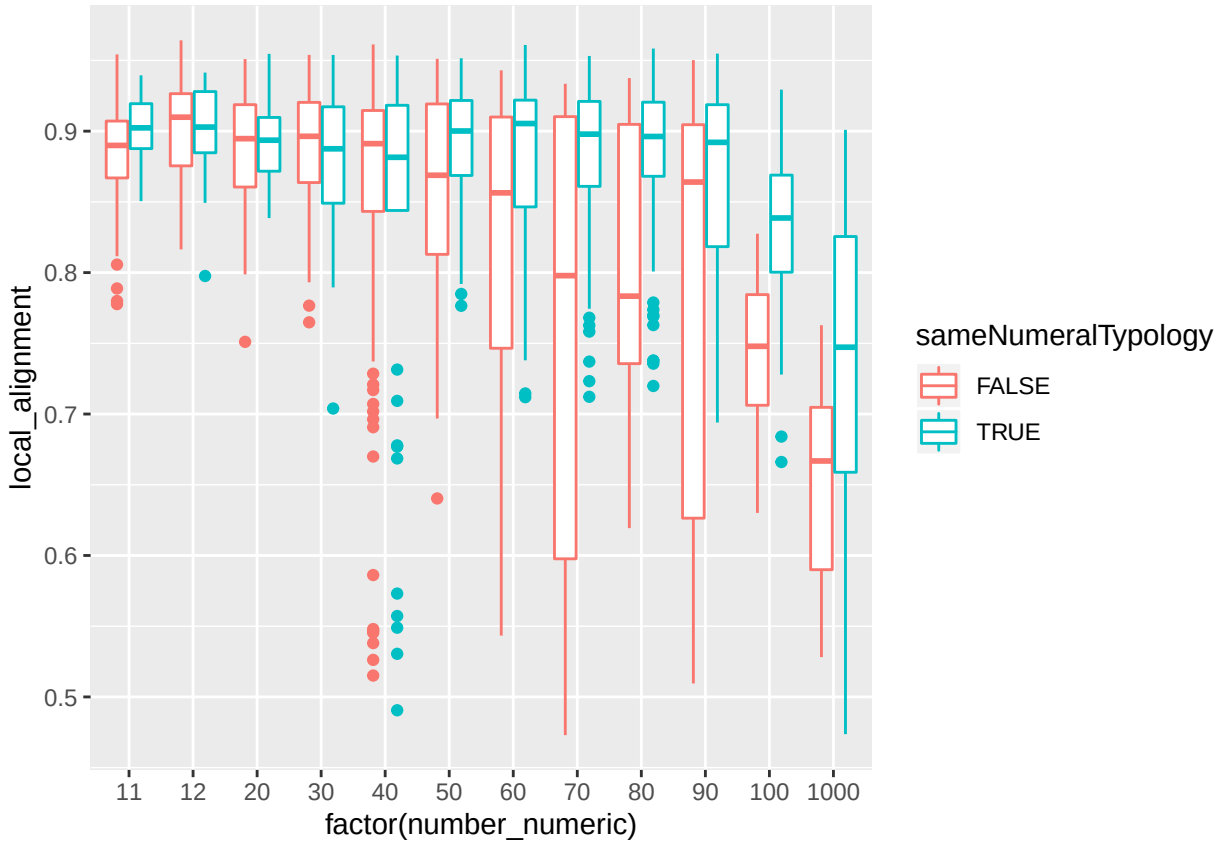
Number line

We can look at the importance of the different ways that numeral systems are composed. We use data from Calude & Verkerk (2016)'s typology which describe the composition of numerals in many Indo-European languages (what Calude & Verkerk call the 'number line'). For example, the numeral 'four' in English is an 'atom' (it is not decomposable into smaller parts), while 'fourteen' is composed of two atoms (four + ten). There are differences between languages in terms of whether they use an atom or a composition of two atoms. For example, English uses a unique atom to represent 12 ('twelve'), while Bulgarian uses a form that is composed as "2 + 10" (2 = dve, 10 = deset, 12 = dvanadeset). There are also differences between languages in how they compose some numerals. For example, 'eighty' in English is constructed as 8 x 10 but in French 'quatre-vingts' is 4 x 20. We used this data to identify whether two languages have the same system for forming a particular numeral.

We note that, in our sample, there are no differences between languages in the number line for numerals from 0 to ten. This is because all numbers below 10 for all languages in the sample are atoms. Therefore, the measure of numeral composition is only explains variation for higher numerals.

It looks like there's an interaction between numeral typology and numeric value:

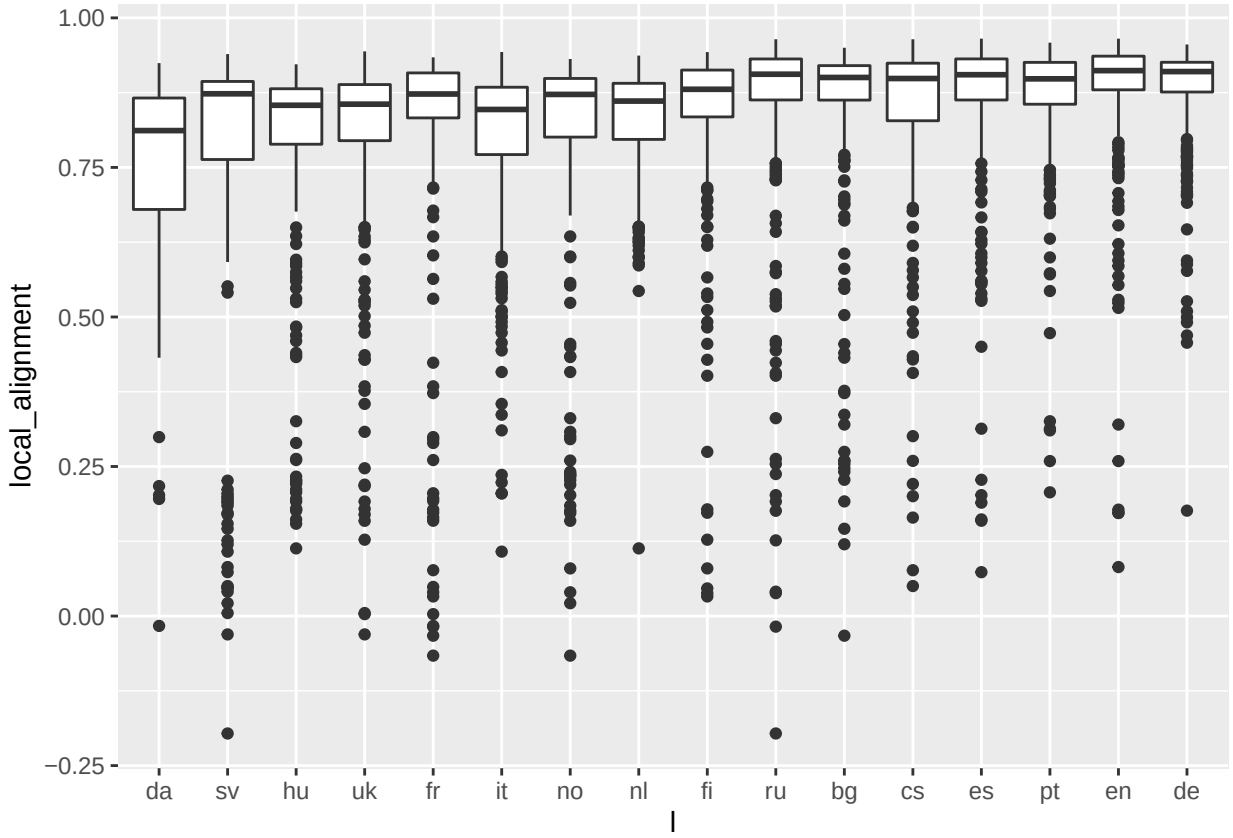
```
ggplot(numbers[numbers$number_numeric>10,],
       aes(y=local_alignment,
           x=factor(number_numeric),
           colour=sameNumeralTypology)) +
  geom_boxplot()
```



Variation by language

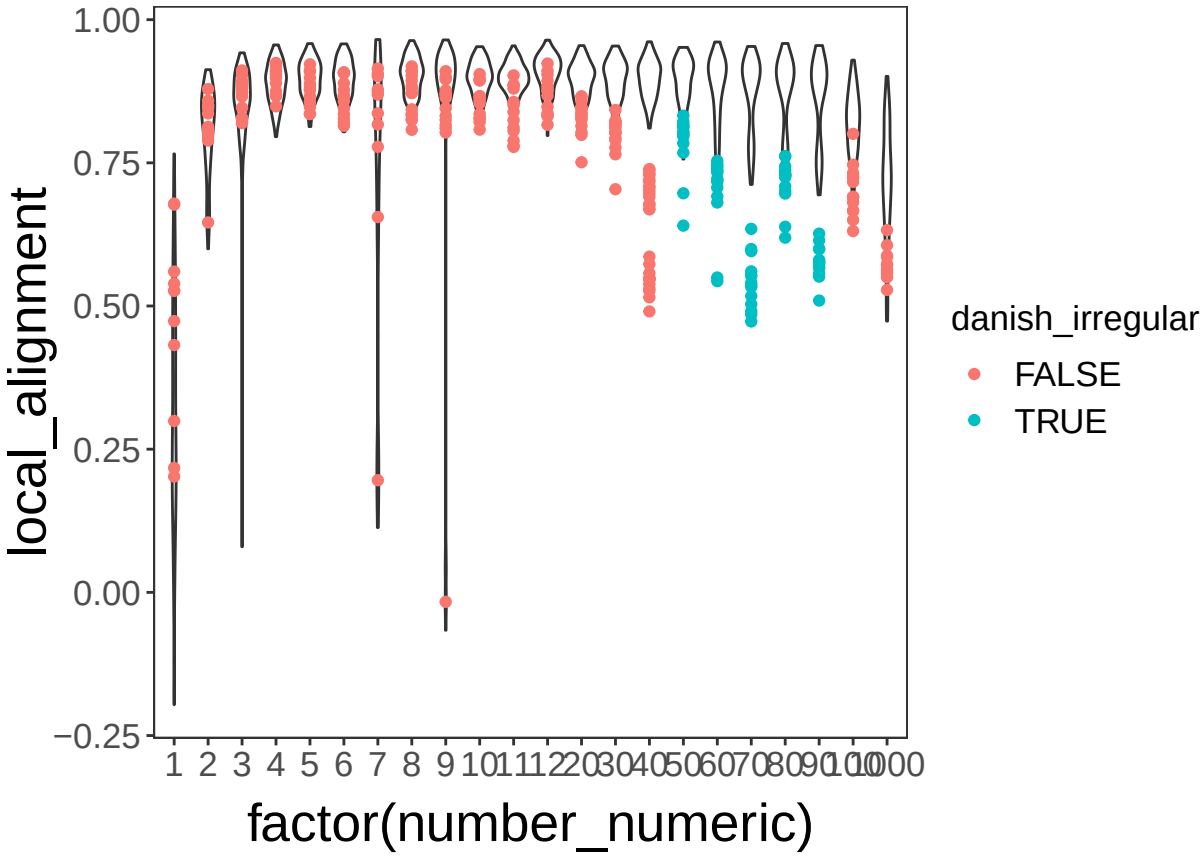
The plot below shows the overview of comparisons by language:

```
ggplot(langAverages,aes(x=l,y=local_alignment)) + geom_boxplot()
```



Danish seems to have a lower average. We note that some Danish 'crowns' are irregular (50,60,70,80,90). These pick out many outliers (dots are danish, blue dots are irregular, violin plots are the rest of the data):

```
ggplot(numbers[!numbers$is_danish,],
       aes(x=factor(number_numeric),y=local_alignment)) +
  geom_violin() + myThemeBasic +
  geom_point(data=numbers[numbers$is_danish,],
             aes(colour=danish_irregular))
```



Formal test of effect of difference between regular and irregular numbers within Danish:

```
summary(lm(local_alignment~irreg_in_danish, data=
  numbers[numbers$is_danish,]))
```

```
##
## Call:
## lm(formula = local_alignment ~ irreg_in_danish, data = numbers[numbers$is_danish,
##     ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.79901 -0.07008  0.04585  0.08937  0.16970
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           0.782552   0.008702  89.931  < 2e-16 ***
## irreg_in_danishTRUE -0.119891   0.018609  -6.442  4.5e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1356 on 309 degrees of freedom
## Multiple R-squared:  0.1184, Adjusted R-squared:  0.1156
## F-statistic: 41.51 on 1 and 309 DF,  p-value: 4.5e-10
```

Variation by frequency

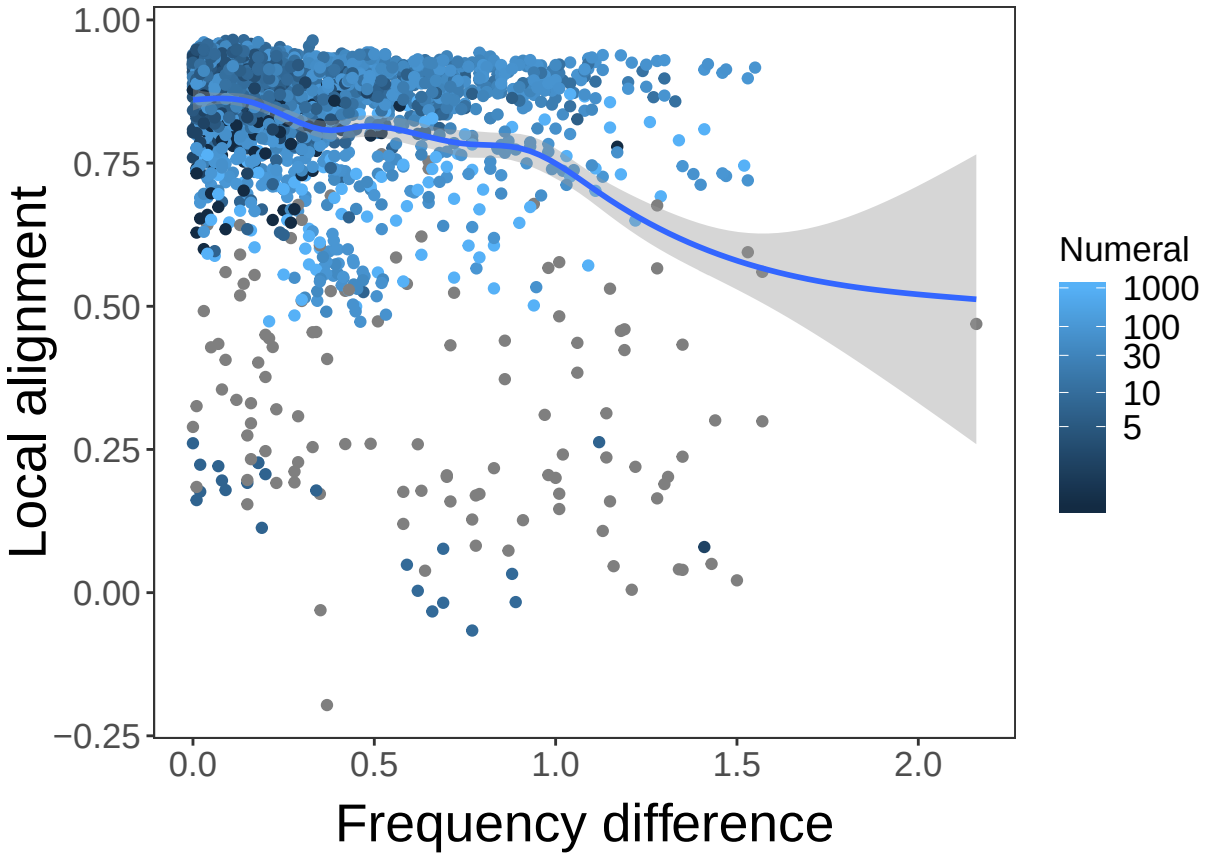
Semantic alignment by frequency (brighter colours are higher numeric values):

```
ggplot(numbers,
       aes(x=freqDiff,y=local_alignment,colour=log10(number_numeric))) +
  geom_point() +
  stat_smooth() +
  scale_colour_gradient(name = "Numeral", trans = "log",
                        breaks = log10(c(1,5,10,30,100,1000)), labels =c(1,5,10,30,100,1000)) +
  myThemeBasic + xlab("Frequency difference") +
  ylab("Local alignment")
```

```
## Warning: Transformation introduced infinite values in discrete y-axis
```

```
## Warning: Transformation introduced infinite values in discrete y-axis
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



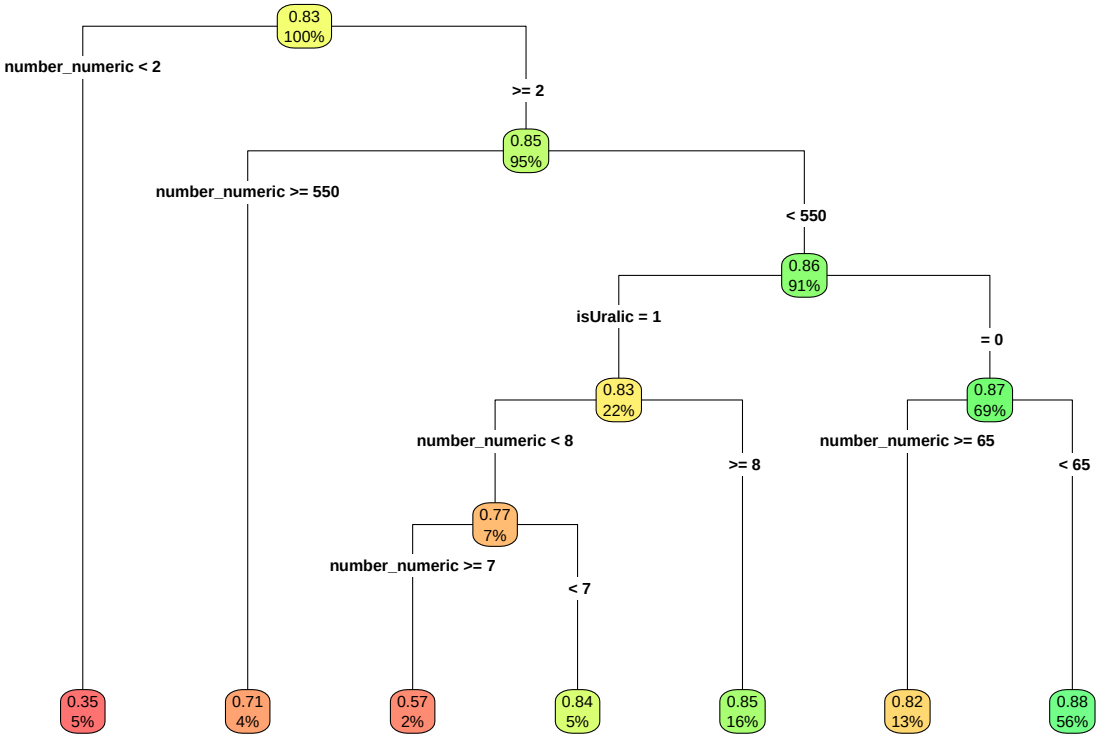
There appears to be a slight effect for larger frequency differences to be associated with lower alignment.

Decision tree

We use a decision tree to explore the data and find coherent clusters in the data. We try to predict local alignment based on various properties.

First we show that Uralic "7s" are a coherent cluster: A decision tree divides the data into categories that represent '1', '1000', and then combines divisions in the numeric value with 'isUralic' to find the cluster of Uralic 7s:

```
rt = REEMtree(local_alignment~
                number_numeric + isUralic,
              random = ~1|lang_pair,
              data=numbers)
rp.rt = tree(rt)
rp.rt$model = numbers
rpart.plot(rp.rt, type=4, branch.lty=1, clip.facs = F, box.palette="RdYlGn")
```



To help this, we include an explicit factor for seven. The final factors in the mode are:

- numeric value
- is a Uralic language
- is a Danish irregular
- is '7'
- has a homophone (an alternative referential class)
- the 'n' variable: number of comparisons possible between language pairs in the whole corpus
- whether the numbers have the same underling compositional structure

```
set.seed(1283)
rt = REEMtree(local_alignment~
                number_numeric + isUralic +
                danish_irregular + hist.dist2 +
                seven + n + sameNumeralTypology + homophone,
              random = ~1|lang_pair,
```

```
              data=numbers,
              MaxIterations=1000000)
rp.rt = tree(rt)
rp.rt$model = numbers
plot1 = rpart.plot(rp.rt, type=4, branch.lty=1, clip.facs = F, box.palette="RdYlGn")

cluster = factor(rp.rt$where,
                 labels = c("One",
                            "Hungarian 7",
                            "French 9",
                            "Small\nhomophones",
                            "Large\nhomophones",
                            "100,1000",
                            "Danish\nirregulars",
                            "Two",
                            "3-90"))

plot2 = ggplot(numbers,aes(y=local_alignment,
                    x=cluster)) +
  geom_violin() + myThemeBasic +
  xlab("") + ylab("Semantic alignment")

pdf("rDecisionTree.pdf",width=12,height=8)
  layout(t(t(c(1,2))), heights=c(2.5,1))
  par(mar=c(1,10,1,1))
  rpart.plot(rp.rt, type=4, branch.lty=2,
             clip.facs = F, box.palette="RdYlGn",
             mar=c(1,4,1,1.5),cex = 1.2,split.yshift=1)
  plot.new()
  vps <- baseViewports()
  pushViewport(vps$figure)
  vp1 <-plotViewport(c(0,0,0,0))
  print(plot2,vp = vp1)
dev.off()
```

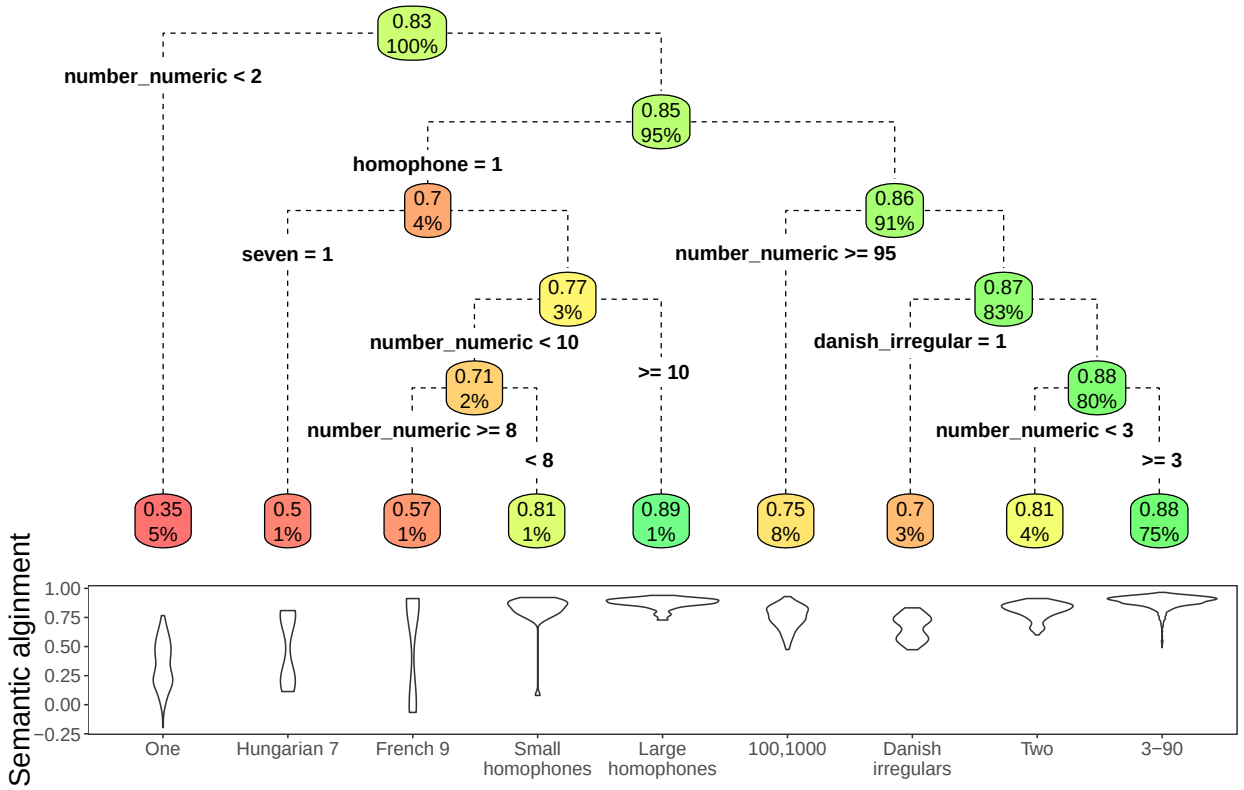


Figure 1: Decision tree predicting the semantic alignment of numerals (upper panel). Each node shows the average alignment of data under that node, and the proportion of data that is represented under that node. Each split in the tree splits the data according to the labelled criteria (e.g. the first split divides the number one from all other numbers). The lower panel shows the distribution of semantic alignment values for the data at each tip of the tree.

```
varimp = sort(rt$Tree$variable.importance)
varimp.plot = ggplot(data.frame(importance=varimp,
                  variable=factor(names(varimp),levels = names(varimp))),
       aes(y=importance,x=variable))+
  geom_col() + coord_flip()
pdf("../../results/numbers/VarImp.pdf")
varimp.plot
dev.off()
```

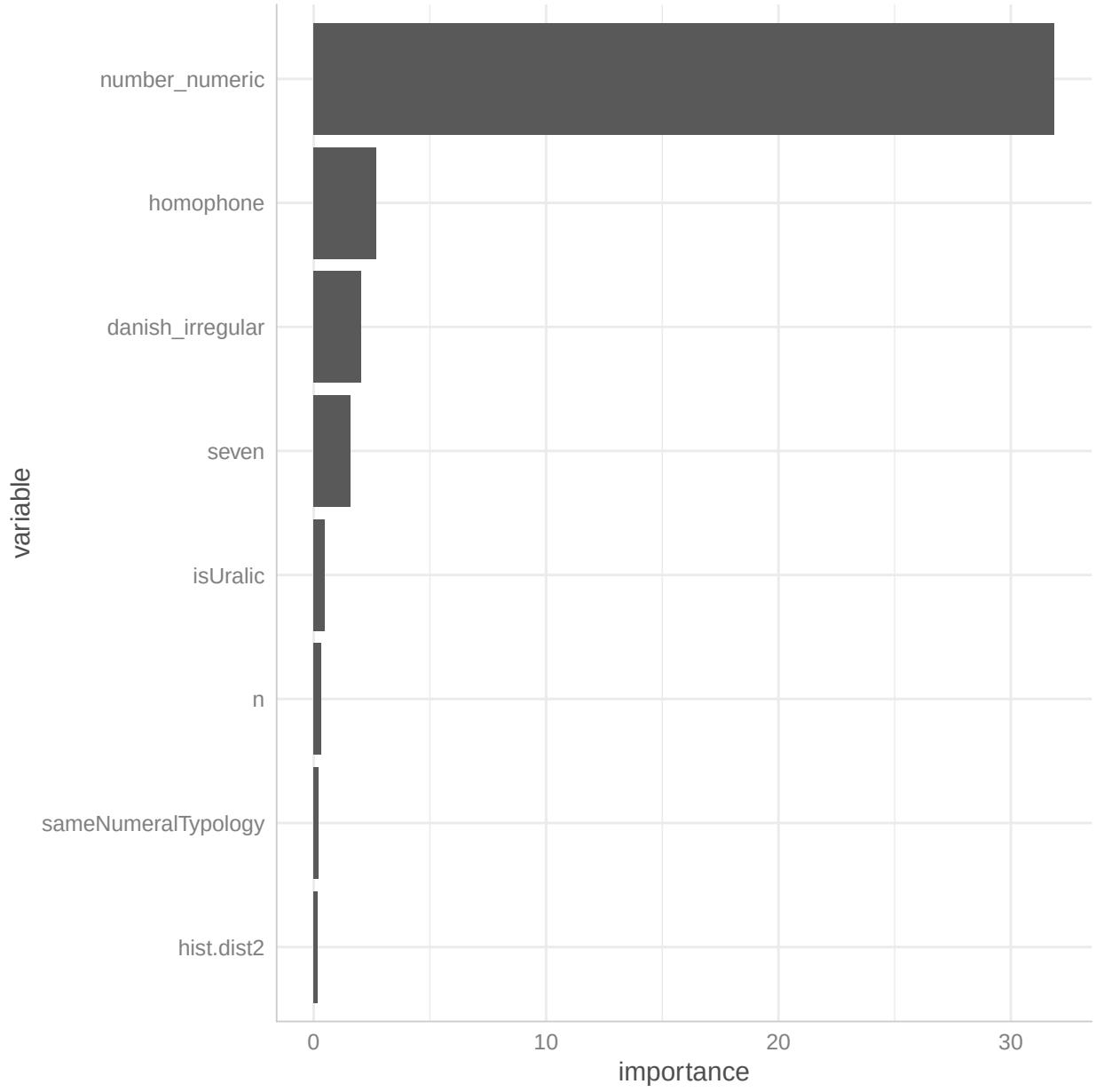


Figure 2: Variable importance measures for the decision tree above

Run a GAM

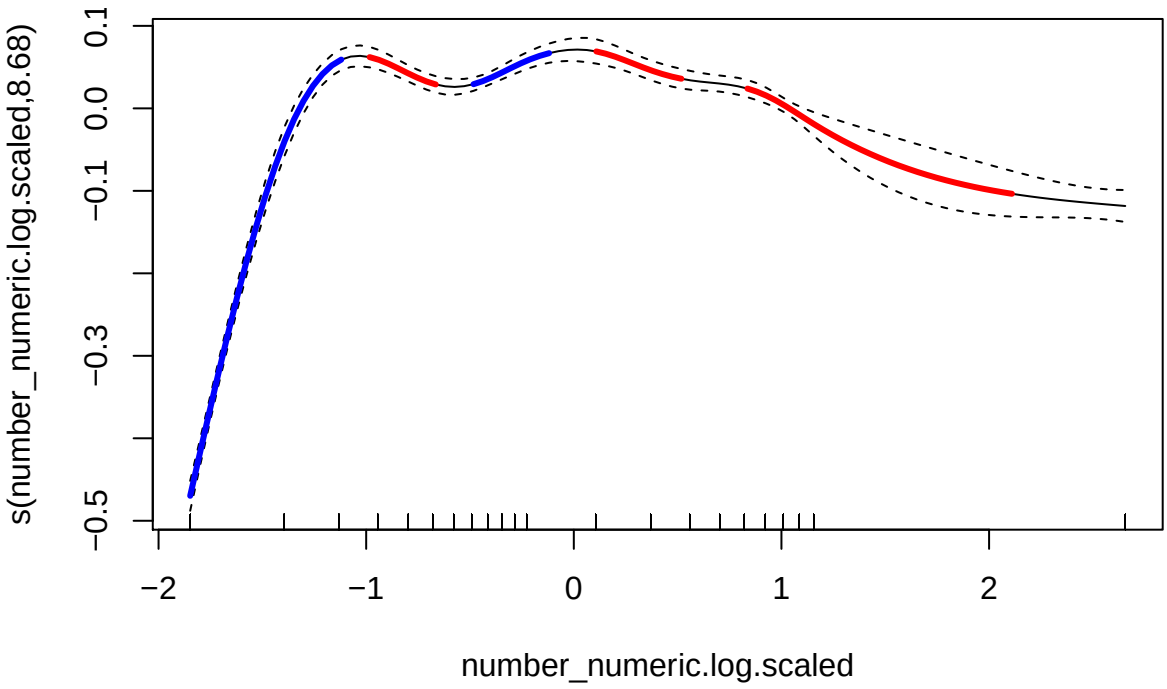
Convert lang_pair to factor and scale variables:

```
numbers$lang_pair = factor(numbers$lang_pair)

numbers$number_numeric.log = log(numbers$number_numeric)
numbers$number_numeric.log.scaled = scale(numbers$number_numeric.log)

numbers$local_alignment.center = scale(numbers$local_alignment,scale=F)

numbers$isUralic = factor(numbers$family_l1 == "Uralic" | numbers$family_l2 == "Uralic")
numbers$differentNumeralTypology = factor(!numbers$sameNumeralTypology)
numbers$seven = factor(numbers$seven)
numbers$danish_irregular = factor(numbers$danish_irregular)
numbers$homophone = factor(numbers$homophone)
```

We start by looking at a simple model that has a random effect for language pair and a main smooth term for number. Note that since this is a non-linear model, a random effect for numeric value is very similar to a fully articulated smooth slope, so we just model numeric as a fixed effect.

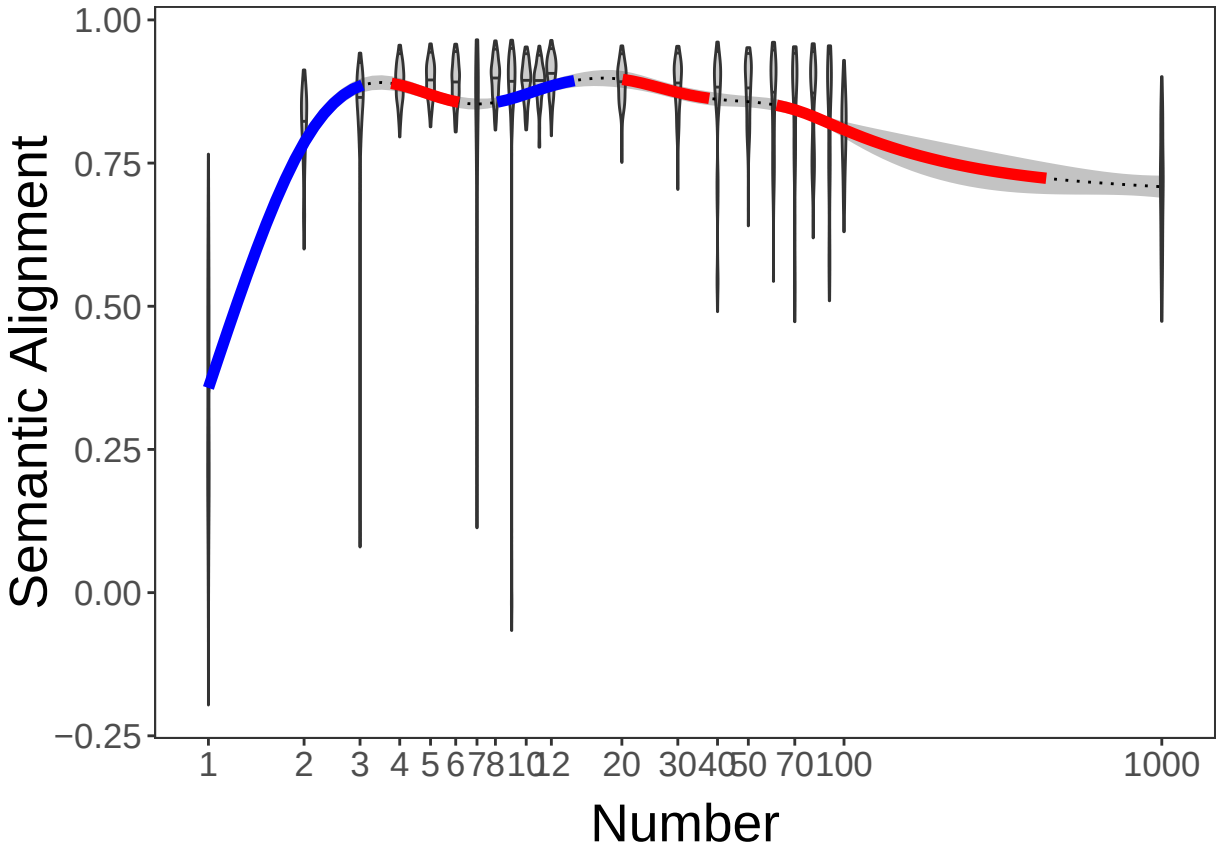
```
m0 = bam(local_alignment.center ~
           s(lang_pair,bs='re') +
           s(number_numeric.log.scaled),
         data = numbers)
```

We plot the fit of the model below:




```
## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to unique
## 'x' values

## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to unique
## 'x' values

## Warning: Removed 61 rows containing missing values (geom_path).
```

```
## Warning: Removed 31 rows containing missing values (geom_path).
```



This shows the difference for 1, a dip for 7 (which the analyses above suggest is due to Uralic languages) and a decrease for 1000.

We now fit a full model with many other predictors:

- Number typology
- Interaction between numeric value and typology
- Historical distance (assuming Uralic is maximum distance)
- Whether the comparison is with a Uralic language
- Whether the number is 7
- Interaction between Uralic and 7
- Whether the number is a Danish irregular
- Whether the number word has a frequent alternative referential class (homophone or synonym)
- The frequency difference between the forms

```
m1 = bam(local_alignment.center~
           s(number_numeric.log.scaled, by=differentNumeralTypology) +
           s(hist.dist2) +
           s(lang_pair,bs='re') +
           isUralic*seven + danish_irregular +
           differentNumeralTypology +
           homophone +
           s(hist.dist2) +
           s(freqDiff),
         data = numbers)
```

Compare models. Adding the extra factors makes a difference.

```r
lrtest(m0,m1)
```

```
## Likelihood ratio test
##
## Model 1: local_alignment.center ~ s(lang_pair, bs = "re") + s(number_numeric.log.scaled)
## Model 2: local_alignment.center ~ s(number_numeric.log.scaled, by = differentNumeralTypology) +
##     s(hist.dist2) + s(lang_pair, bs = "re") + isUralic * seven +
##     danish_irregular + differentNumeralTypology + homophone +
##     s(hist.dist2) + s(freqDiff)
##      #Df LogLik     Df  Chisq Pr(>Chisq)
## 1 109.15 2199.7
## 2 117.38 2518.2 8.2261 637.08  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(m1)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## local_alignment.center ~ s(number_numeric.log.scaled, by = differentNumeralTypology) +
##     s(hist.dist2) + s(lang_pair, bs = "re") + isUralic * seven +
##     danish_irregular + differentNumeralTypology + homophone +
##     s(hist.dist2) + s(freqDiff)
##
## Parametric coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   0.017116   0.004897   3.495 0.000483 ***
## isUralicTRUE                  0.001801   0.009772   0.184 0.853816
## sevenTRUE                    -0.029093   0.013742  -2.117 0.034367 *
## danish_irregularTRUE         -0.144641   0.013097 -11.044  < 2e-16 ***
## differentNumeralTypologyTRUE  0.024844   0.032019   0.776 0.437883
## homophoneTRUE                -0.112477   0.008755 -12.847  < 2e-16 ***
## isUralicTRUE:sevenTRUE       -0.191216   0.020627  -9.270  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                                                edf  Ref.df
## s(number_numeric.log.scaled):differentNumeralTypologyFALSE  8.598   8.944
## s(number_numeric.log.scaled):differentNumeralTypologyTRUE   3.986   4.514
## s(hist.dist2)                                                1.000   1.000
## s(lang_pair)                                                92.082 117.000
## s(freqDiff)                                                  1.000   1.000
##                                                                  F  p-value
## s(number_numeric.log.scaled):differentNumeralTypologyFALSE 305.027  < 2e-16 ***
## s(number_numeric.log.scaled):differentNumeralTypologyTRUE   45.087  < 2e-16 ***
## s(hist.dist2)                                                0.671    0.413
## s(lang_pair)                                                 3.532  < 2e-16 ***
## s(freqDiff)                                                 24.254 9.02e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## R-sq.(adj) =  0.699   Deviance explained = 71.3%
## fREML = -2301.9  Scale est. = 0.0076339  n = 2415
```

Summary of main model

The final model explains 71.33% of the deviance, compared to 62.67% in the baseline model.

Parametric effects:

- Being an Uralic language is not a strong predictor.
- The number 7 has lower alignment in general (β = -0.0291 , p = 0.0344)
- Danish irregulars have lower alignment (β = -0.145 , p = $<$ 0.001))
- Overall, there is no difference for comparisons between numbers with different numeral typologies (β = 0.0248 , p = 0.4379)
- Alignment is lower for comparisons between words were at least one has an alternative referential class (homophone or synonym) (β = -0.112 , p = $<$ 0.001)
- Comparisons with Uralic sevens are significantly lower in alignment (β = -0.191 , p = $<$ 0.001)

Smooth effects:

- The EDF for numeral is high, indicating strong non-linearities.
- There is an interaction between numeral value and numeral typology. The graph below shows that alignment is lower for comparisons between numbers with different typologies for numbers above 50. Also note that there is a slight increase in the semantic alignment for number 20 if the two languages have the same numeral typology.
- There was no effect of historical distance.
- There were significant random effects for particular language pairs.
- There was a significant linear effect of frequency difference: Alignment was lower when the frequency difference was larger.

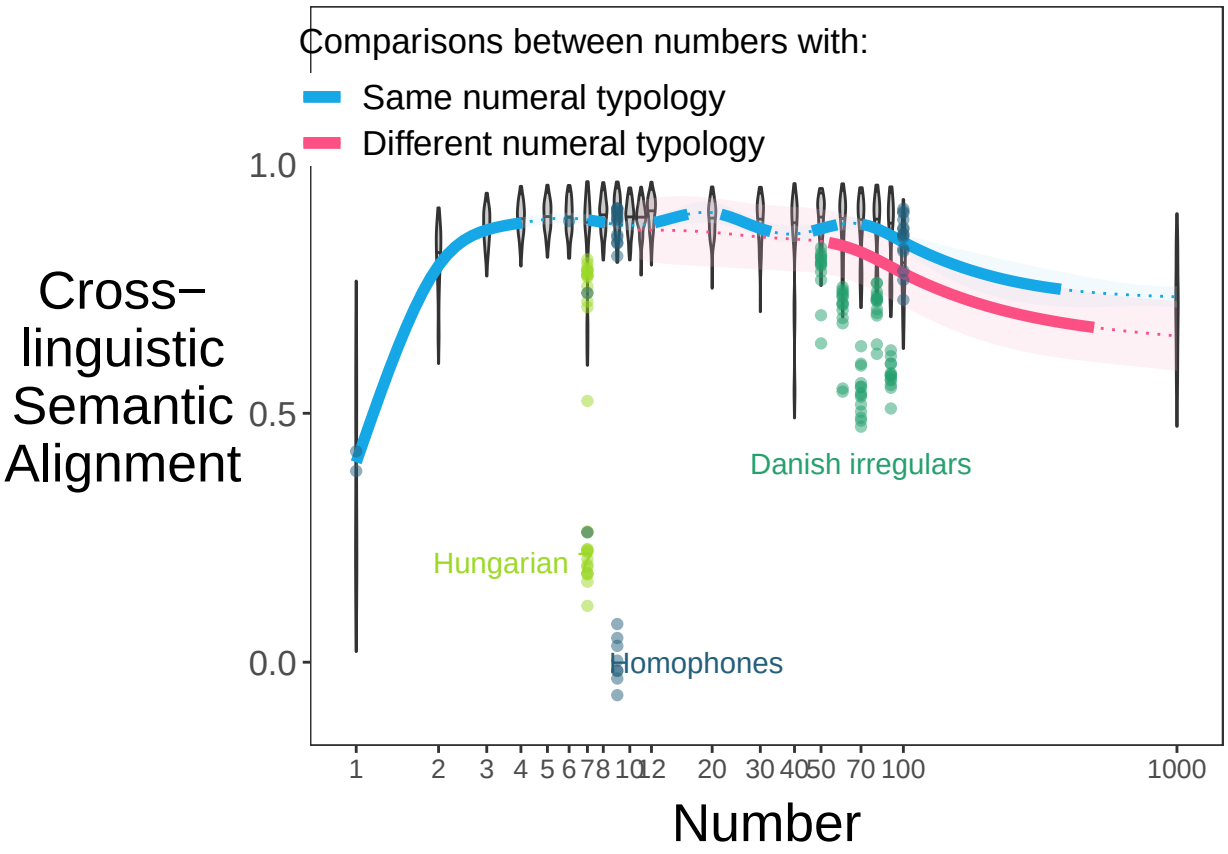
The hidden code below (see the Rmd file) detects significantly steep slopes in the GAM curve. Thick line segments indicate significant rises or decreases. We see that 1 and 2 have lower alignment, then numbers 3-20 are fairly constant. Beyond 20, there is a decrease in alignment, especially for numbers with different numeral typologies. Various outliers captured by the model are drawn on top. Note that there are no numbers below 10 that have different numeral typologies, so we have truncated the curve accordingly.

```
gamPlot
```

```
## Warning: Removed 34 rows containing missing values (geom_path).
```

```
## Warning: Removed 141 rows containing missing values (geom_path).
```

```
## Warning: Removed 109 rows containing missing values (geom_path).
```




```
pdf("../../results/numbers/FinalGamModel.pdf", width=7.5,height=4.5)
gamPlot
```

```
## Warning: Removed 34 rows containing missing values (geom_path).
```

```
## Warning: Removed 141 rows containing missing values (geom_path).
```

```
## Warning: Removed 109 rows containing missing values (geom_path).
```

```
dev.off()
```

```
## pdf
##   2
```

Controlling for linguistic history

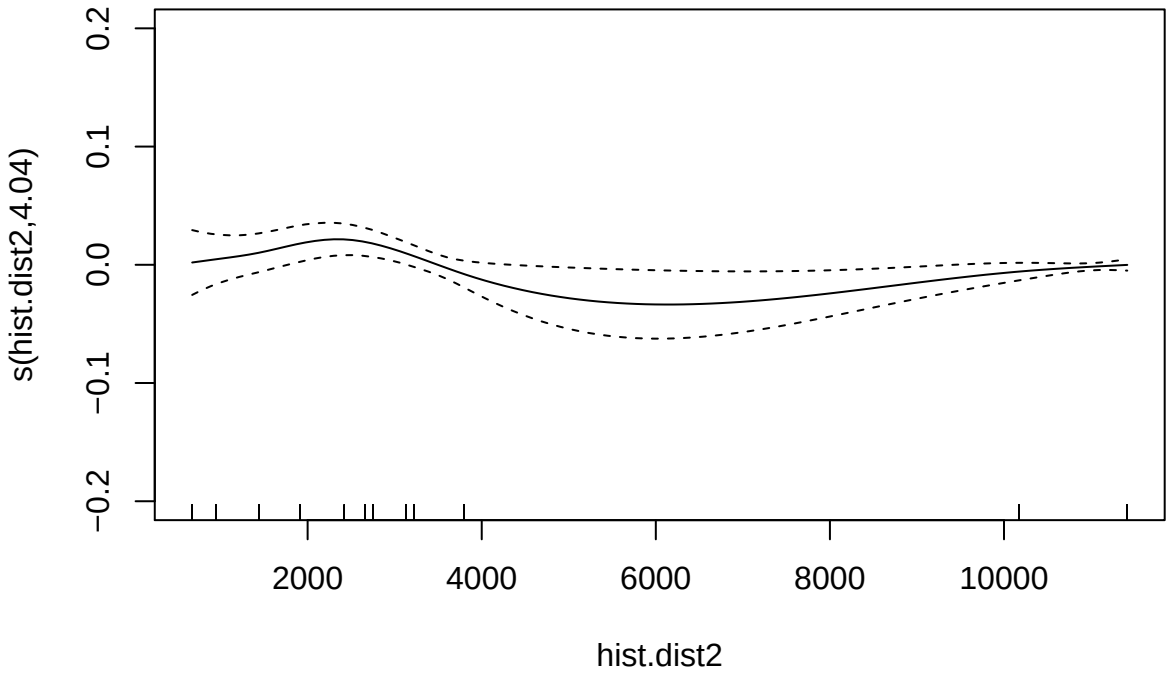
In the model above, the effects of historical distance are minimal: the fit is linear and not significant. This might be because there are random effects for language pairs which are taking up the variance. In the section below, we use only historical distance:

```
m1.phylo = bam(local_alignment.center~
           s(number_numeric.log.scaled, by=differentNumeralTypology) +
           s(hist.dist2) +
           seven + danish_irregular +
           homophone +
           differentNumeralTypology +
           s(freqDiff),
         data = numbers[!is.na(numbers$hist.dist),])
# Model with numeric
summary(m1.phylo)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## local_alignment.center ~ s(number_numeric.log.scaled, by = differentNumeralTypology) +
##     s(hist.dist2) + seven + danish_irregular + homophone + differentNumeralTypology +
##     s(freqDiff)
##
## Parametric coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   0.017078   0.002922   5.845    6e-09 ***
## sevenTRUE                    -0.036295   0.014328  -2.533   0.0114 *
## danish_irregularTRUE         -0.189497   0.013823 -13.709   <2e-16 ***
## homophoneTRUE                -0.112117   0.010836 -10.347   <2e-16 ***
## differentNumeralTypologyTRUE  0.048179   0.072356   0.666   0.5056
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                                        edf Ref.df       F
## s(number_numeric.log.scaled):differentNumeralTypologyFALSE 8.535  8.927 215.732
## s(number_numeric.log.scaled):differentNumeralTypologyTRUE  4.697  5.104  40.902
## s(hist.dist2)                                              4.042  4.587   2.498
## s(freqDiff)                                                1.000  1.000  62.586
##                                                         p-value
## s(number_numeric.log.scaled):differentNumeralTypologyFALSE  < 2e-16 ***
## s(number_numeric.log.scaled):differentNumeralTypologyTRUE   < 2e-16 ***
## s(hist.dist2)                                                0.0271 *
## s(freqDiff)                                                4.31e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.674   Deviance explained = 67.8%
## fREML = -1731.5  Scale est. = 0.0081138  n = 1819
```

The effects are very similar. The effect of history is significant, but not large. Semantic alignment is lower for more distantly related languages:

```
plot(m1.phylo,select=3,ylim=c(-0.2,0.2))
```

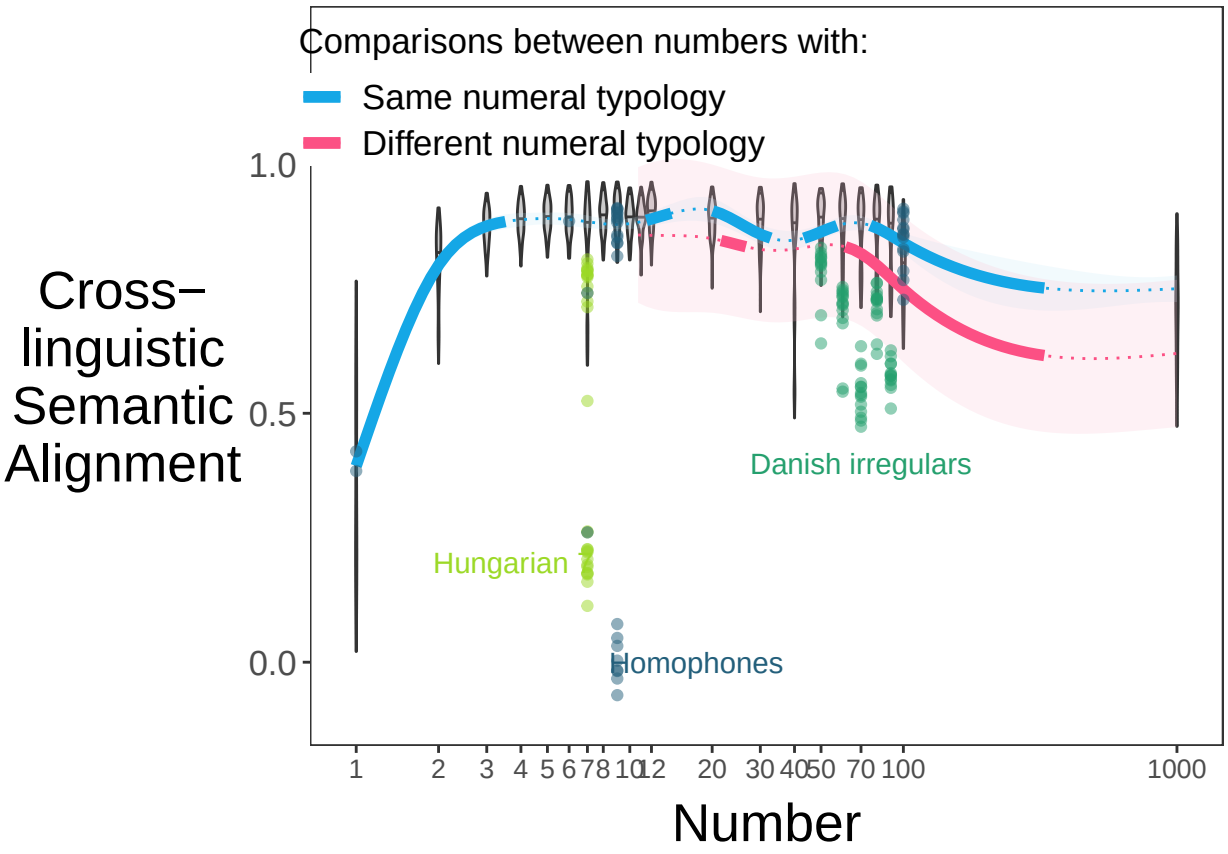


Plot the main effects, which look much like the main graph above.

```
## Warning: Removed 34 rows containing missing values (geom_path).
```

```
## Warning: Removed 141 rows containing missing values (geom_path).
```

```
## Warning: Removed 119 rows containing missing values (geom_path).
```


Comparisons between numbers with:
Same numeral typology
Different numeral typology
Cross−linguistic Semantic Alignment
1.0
0.5
0.0
Danish irregulars
Hungarian
Homophones
1
2
3
4
5
6
7
8
10
12
20
30
40
50
70
100
1000
Number

Conclusion

Semantic alignment for number words is generally high, though there are some differences that can be explained (the model explained 71.33% of the deviance). 1 and 2 have lower alignment due to often being grammaticalised as indefinite or dual marker (Givon, 1981). Numbers 3-12 generally have high alignment (mean local alignment = 0.87), and higher numbers decline in alignment up to 1000. There are also language-specific differences due to how numerals are constructed (e.g. base, combination rules, see Calude & Verkerk, 2016), or for irregular forms (e.g. 50, 60, 70, 80 and 90 in Danish). Some number words have alternative associations due to homophones or polysemies (e.g. the Hungarian 7 is used directly to mean 'week', and 'neuf' in French means '9' or 'new'). The historical distance between languages did not predict much of the variation.

The differences in semantic alignment may appear either because (A) the semantic associations are different for different languages, or (B) as a side-effect of numbers appearing with different words skewing the alignment metric. Effects that support (B) could include:

- Difference in frequency. Lower frequency terms will appear in a smaller range of contexts and the semantic alignment estimates may be more stochastic.

- Alternative referential class (homophone or synonym). Different meanings will contribute different semantic relations.

However, it is more difficult to explain why different numeral typologies would lead to semantic differences, unless the way numbers are constructed affects the way people think about the numbers.

References

Calude, A. S., & Verkerk, A. (2016). The typology and diachrony of higher numerals in Indo-European: a phylogenetic comparative study. Journal of Language Evolution, 1(2), 91-108.

Dehaene, S., & Mehler, J. (1992). Cross-linguistic regularities in the frequency of number words. Cognition, 43(1), 1-29.

Givón, T. (1981). On the development of the numeral 'one'as an indefinite marker. Folia Linguistica Historica, 15(Historica vol. 2, 1), 35-54.