Chapter 4. The effect of verb class

Raquel Montero Estebaranz

05.22.2024

Contents

1. Introduction

This file shows the code used to analyse the data for Chapter 4 of the thesis "Mood alternations: a synchronic and diachronic study of negated complement clauses". The data as well as the annotation guidelines can be found at: https://github.com/Raquel-Montero

The following are the Packages that will be used:

```
library(readr)
library(carData)      # for cat package
library(car)          # Anova function
library(dplyr)        # Operations
library(plyr)         # for ddply
library(ggplot2)      # to use ggplot
library(sjPlot)       # to change the font
library(Matrix)       # for lme4 package
library(lme4)         # to calculate lmer models
library(lmerTest)     # for the p values
library(mgcv)         # for Gam model
library(itsadug)      # for Gam model
library(tidygam)      # for Gam plotting
library(tidymv)       # For Gam plotting: https://www.rdocumentation.org/packages/tidymv/versions/3.4.2
library(ggpubr)
library(LaplacesDemon) #for the function invlogit
library(ggpubr)
library(functional)
library(scales)
```

2. General Development: Graphs

Loading the Data:

```
data <- read_csv("DiachronyAllVerbs.csv",
                 show_col_types = FALSE,
                 locale = locale(encoding =  "ISO-8859-1"))
```

Final Data that will be used for analysis:

```
clean.data <- subset(data, Mverbtype =="ind" & # the matrix verb is in indicative
                     Mverbclass!="Na"& # excludes any data that is not one of the four verbs
                     Emood !="Na" & # no embedded verb
                     Emood !="NA" &
                     Emood !="inf" & # embedded verb in infinitive
                     MClauseType=="noninterrogative" & # only takes into account non-interrogative clau
                     EClauseType=="unambiguous" # exclude ambigous sentences
                   )

# Changing the indicative to 1 and subjunctive to 0:
clean.data$Emood2 <- ifelse(clean.data$Emood == "subj", 0, 1)

# Converting the mood into a numeric value:
clean.data$Emood2 <- as.numeric(as.character(clean.data$Emood2))

#changing names of values so that they are clearer:
clean.data$Mverbclass2 <- ifelse(clean.data$Mverbclass == "factive", "semi-factive", "non-factive")
clean.data$Construction <- ifelse(clean.data$Construction == "1st present", "1st present (FP)", "other
```

Data for plotting:

```
# Creates new data-frame with the means of embedded mood per period, verb class/verb and construction:

plot.data.verbtype <- ddply(clean.data, # data frame
                            .(Period, Construction,Mverbclass2), # variables to calculate the mean
                            summarize, #summary
                            mean = mean(as.numeric(as.character(Emood2)), na.rm = T), # mean
                            n = sum(!is.na(as.numeric(as.character(Emood2)))) # number of tokens
                            )
plot.data.verbtype
```

```
##    Period     Construction  Mverbclass2       mean    n
## 1    1200 1st present (FP)  non-factive 0.27272727   11
## 2    1200 1st present (FP) semi-factive 0.33333333    3
## 3    1200        other (0)  non-factive 0.70000000   20
## 4    1200        other (0) semi-factive 0.80645161   31
## 5    1300 1st present (FP)  non-factive 0.36666667   30
## 6    1300 1st present (FP) semi-factive 0.44444444    9
## 7    1300        other (0)  non-factive 0.42307692   26
## 8    1300        other (0) semi-factive 0.59459459   37
## 9    1400 1st present (FP)  non-factive 0.41089109  202
## 10   1400 1st present (FP) semi-factive 0.65714286   35
## 11   1400        other (0)  non-factive 0.66206897  145
## 12   1400        other (0) semi-factive 0.84745763  118
## 13   1500 1st present (FP)  non-factive 0.70249520  521
## 14   1500 1st present (FP) semi-factive 0.34693878   98
## 15   1500        other (0)  non-factive 0.68446602  206
## 16   1500        other (0) semi-factive 0.68627451  204
## 17   1600 1st present (FP)  non-factive 0.66285714  175
## 18   1600 1st present (FP) semi-factive 0.11666667  120
## 19   1600        other (0)  non-factive 0.82051282  156
## 20   1600        other (0) semi-factive 0.80701754  114
```

```
## 21   1700 1st present (FP)  non-factive 0.26666667  120
## 22   1700 1st present (FP) semi-factive 0.01470588   68
## 23   1700        other (O)  non-factive 0.57352941   68
## 24   1700        other (O) semi-factive 0.79310345   29
## 25   1800 1st present (FP)  non-factive 0.15139442  502
## 26   1800 1st present (FP) semi-factive 0.03947368   76
## 27   1800        other (O)  non-factive 0.42924528  212
## 28   1800        other (O) semi-factive 0.83783784  148
## 29   1900 1st present (FP)  non-factive 0.08325171 1021
## 30   1900 1st present (FP) semi-factive 0.06849315   73
## 31   1900        other (O)  non-factive 0.36000000  225
## 32   1900        other (O) semi-factive 0.83141762  261
```

```
plot.data.verbs <- ddply(clean.data,
                         .(Period, Construction,Mverbl),
                         summarize,
                         mean = mean(as.numeric(as.character(Emood2)), na.rm = T),
                         n = sum(!is.na(as.numeric(as.character(Emood2))))
                         )
plot.data.verbs
```

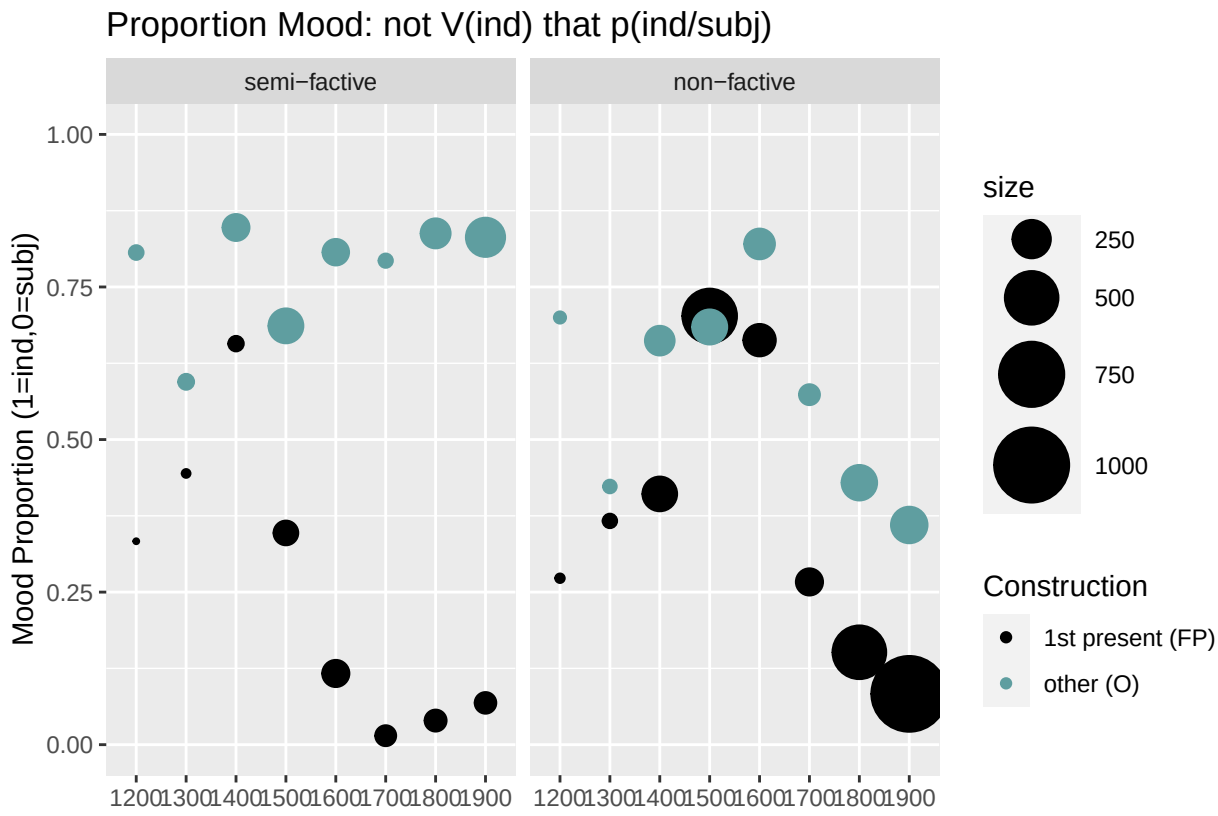
```
##    Period     Construction          Mverbl       mean   n
## 1    1200 1st present (FP) creer (believe) 0.27272727  11
## 2    1200 1st present (FP)    saber (know) 0.00000000   2
## 3    1200 1st present (FP)       ver (see) 1.00000000   1
## 4    1200        other (O) creer (believe) 0.70000000  10
## 5    1200        other (O)     decir (say) 0.70000000  10
## 6    1200        other (O)    saber (know) 0.81481481  27
## 7    1200        other (O)       ver (see) 0.75000000   4
## 8    1300 1st present (FP) creer (believe) 0.36000000  25
## 9    1300 1st present (FP)     decir (say) 0.40000000   5
## 10   1300 1st present (FP)    saber (know) 0.33333333   3
## 11   1300 1st present (FP)       ver (see) 0.50000000   6
## 12   1300        other (O) creer (believe) 0.38888889  18
## 13   1300        other (O)     decir (say) 0.50000000   8
## 14   1300        other (O)    saber (know) 0.58064516  31
## 15   1300        other (O)       ver (see) 0.66666667   6
## 16   1400 1st present (FP) creer (believe) 0.33587786 131
## 17   1400 1st present (FP)     decir (say) 0.54929577  71
## 18   1400 1st present (FP)    saber (know) 0.72222222  18
## 19   1400 1st present (FP)       ver (see) 0.58823529  17
## 20   1400        other (O) creer (believe) 0.44444444  54
## 21   1400        other (O)     decir (say) 0.79120879  91
## 22   1400        other (O)    saber (know) 0.85416667  96
## 23   1400        other (O)       ver (see) 0.81818182  22
## 24   1500 1st present (FP) creer (believe) 0.76010101 396
## 25   1500 1st present (FP)     decir (say) 0.52000000 125
## 26   1500 1st present (FP)    saber (know) 0.17543860  57
## 27   1500 1st present (FP)       ver (see) 0.58536585  41
## 28   1500        other (O) creer (believe) 0.58695652  92
## 29   1500        other (O)     decir (say) 0.76315789 114
## 30   1500        other (O)    saber (know) 0.60135135 148
## 31   1500        other (O)       ver (see) 0.91071429  56
## 32   1600 1st present (FP) creer (believe) 0.76699029 103
## 33   1600 1st present (FP)     decir (say) 0.51388889  72
```

```
## 34   1600 1st present (FP)    saber (know) 0.05050505  99
## 35   1600 1st present (FP)       ver (see) 0.42857143  21
## 36   1600        other (O) creer (believe) 0.64705882  51
## 37   1600        other (O)     decir (say) 0.90476190 105
## 38   1600        other (O)    saber (know) 0.80769231  78
## 39   1600        other (O)       ver (see) 0.80555556  36
## 40   1700 1st present (FP) creer (believe) 0.27551020  98
## 41   1700 1st present (FP)     decir (say) 0.22727273  22
## 42   1700 1st present (FP)    saber (know) 0.00000000  49
## 43   1700 1st present (FP)       ver (see) 0.05263158  19
## 44   1700        other (O) creer (believe) 0.41666667  24
## 45   1700        other (O)     decir (say) 0.65909091  44
## 46   1700        other (O)    saber (know) 0.76190476  21
## 47   1700        other (O)       ver (see) 0.87500000   8
## 48   1800 1st present (FP) creer (believe) 0.14157303 445
## 49   1800 1st present (FP)     decir (say) 0.22807018  57
## 50   1800 1st present (FP)    saber (know) 0.05769231  52
## 51   1800 1st present (FP)       ver (see) 0.00000000  24
## 52   1800        other (O) creer (believe) 0.38053097 113
## 53   1800        other (O)     decir (say) 0.48484848  99
## 54   1800        other (O)    saber (know) 0.82352941 119
## 55   1800        other (O)       ver (see) 0.89655172  29
## 56   1900 1st present (FP) creer (believe) 0.08237748 959
## 57   1900 1st present (FP)     decir (say) 0.09677419  62
## 58   1900 1st present (FP)    saber (know) 0.10638298  47
## 59   1900 1st present (FP)       ver (see) 0.00000000  26
## 60   1900        other (O) creer (believe) 0.25625000 160
## 61   1900        other (O)     decir (say) 0.61538462  65
## 62   1900        other (O)    saber (know) 0.82832618 233
## 63   1900        other (O)       ver (see) 0.85714286  28
```

Plotting the results:

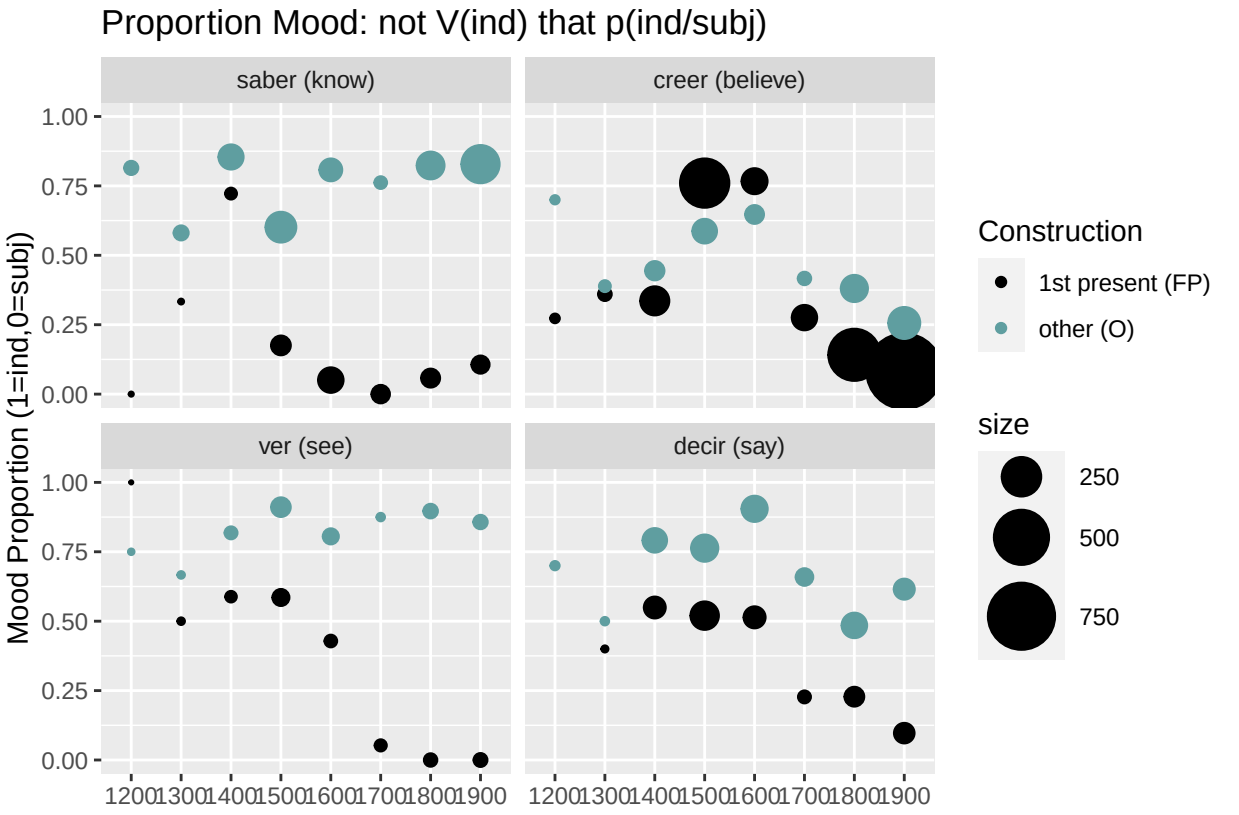
```
# Plot 1: Mood selection of different verb classes

plot.data.verbtype$Period <- as.factor(plot.data.verbtype$Period) # Period as factor

verbtype.diachrony <- ggplot()+
                geom_point(data=plot.data.verbtype, #main data
                           aes(Period, # x-axis
                               mean, # y-axis
                               size = n, # size of the dots
                               color=Construction #colors
                               )
                           )+
                facet_wrap(~factor(Mverbclass2, # divide into facets
                                   levels = c('semi-factive', 'non-factive') # order of the facets
                                   )
                           )+
                scale_size_area(max_size=13)+  # controls the maximum side of the points.
                labs(title="Proportion Mood: not V(ind) that p(ind/subj)",  # axis labels
                      x = " ",
                       y="Mood Proportion (1=ind,0=subj)")+
                scale_color_manual(values=c("black", "#5F9EA0" ))+ # color values
                labs(size="size", colour="Construction")+ # labels legends
```

```
                ylim(0,1) # minimum/maximum values of the y-axis

verbtype.diachrony
```




```
# Plot 2: mood selection after each individual verb

plot.data.verbs$Period <- as.factor(plot.data.verbs$Period) # Period as factor

verbs.diachrony <- ggplot()+
              geom_point(data=plot.data.verbs, #main data
                         aes(Period, # x-axis
                             mean, # y-axis
                             size = n, # size of the dots
                             color=Construction) # colors
                         )+
             facet_wrap(~factor(Mverbl, #divide into facets
                                levels=c('saber (know)', 'creer (believe)', 'ver (see)', 'decir (say)'))
                                ncol=2 # number of columns
                        )+
              scale_size_area(max_size=13)+ # maximum size of the dots
              labs(title="Proportion Mood: not V(ind) that p(ind/subj)", #labels
                    x = " ",
                    y="Mood Proportion (1=ind,0=subj)")+
             scale_color_manual(values=c("black", "#5F9EA0" ))+ # colors plot
            ylim(0,1)+ # minimum/maximum value of the y-axis
            labs(size="size", colour="Construction")
```

```
verbs.diachrony
```




```
#Saving the plots:
ggsave(verbtype.diachrony, file="verb-classes-diachrony.png", width = 8, height= 4)
ggsave(verbs.diachrony, file="verbs-diachrony.png", width = 8, height= 6)
```

3. Statistical Analysis

```
# changing verbclass and construction to factor:
clean.data$Mverbclass <- as.factor(clean.data$Mverbclass)#
clean.data$Construction <- as.factor(clean.data$Construction)

# change year into a numeric variable:
clean.data$Year2 <- as.numeric(clean.data$Year2)

# create a new column with Class and Construction as one combined factor: (for the GAM model)
clean.data <- mutate(clean.data, classContruction = interaction(Mverbclass, Construction))

# use a contrast treatment and order the variable:
clean.data$classContruction <- as.ordered(clean.data$classContruction)
contrasts(clean.data$classContruction) <- "contr.treatment"
```

3.1. GAM: Exploring the verb class effect

```
# No effect of Verb class:
```

```
model <- bam(Emood2 ~ Construction+
                      s(Year2)+
                     s(Year2,by=Construction),
                    method = "REML",
                   family="binomial",
                    data = clean.data)

# Main efect of verb class but no interaction with it: -->
model2 <- bam(Emood2 ~ Construction+Mverbclass+
                      s(Year2)+
                      s(Year2, by=Construction),
                    method = "REML",
                   family="binomial",
                    data = clean.data)


# Three way interaction:
model3 <- bam(Emood2 ~ classContruction+
                      s(Year2)+
                      s(Year2, by=classContruction),
                    method = "REML",
                   family="binomial",
                    data = clean.data)
```

Gam models

```
# Comparing the models:
AIC(model2, model) # no effect of verb class vs. main effect of verb class
```

Comparision of the models

```
##              df      AIC
## model2 15.41804 5363.737
## model  14.36729 5370.142
```

```
AIC(model3, model2) # main effect of verb class vs. three way interaction
```

```
##              df      AIC
## model3 20.15714 5045.599
## model2 15.41804 5363.737
```

```
final.model <- bam(Emood2 ~ classContruction+
                      s(Year2)+
                      s(Year2, by=classContruction),
                    method = "REML",
                   family="binomial",
                    data = clean.data)

summary(final.model)
```

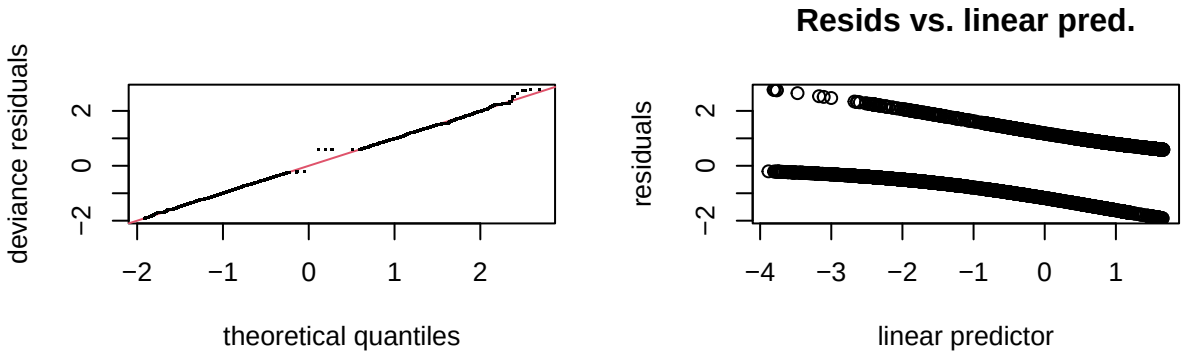
Gam final model

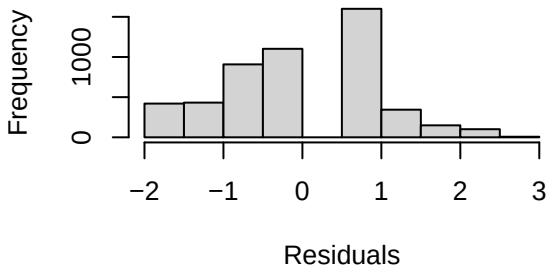
```
##
## Family: binomial
```

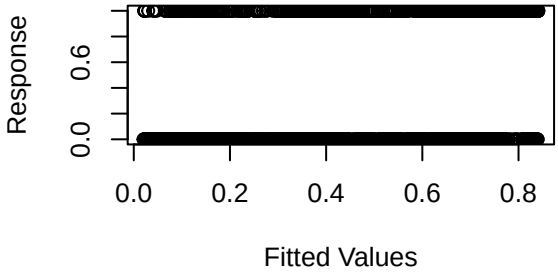
```
## Link function: logit
##
## Formula:
## Emood2 ~ classContruction + s(Year2) + s(Year2, by = classContruction)
##
## Parametric coefficients:
##                                             Estimate Std. Error z value
## (Intercept)                                  -2.1695     0.1914 -11.335
## classContructionnon-factive.1st present (FP)   1.2379     0.1990   6.221
## classContructionfactive.other (O)              3.5317     0.2091  16.888
## classContructionnon-factive.other (O)          2.3310     0.2031  11.478
##                                             Pr(>|z|)
## (Intercept)                                  < 2e-16 ***
## classContructionnon-factive.1st present (FP) 4.94e-10 ***
## classContructionfactive.other (O)             < 2e-16 ***
## classContructionnon-factive.other (O)         < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                                       edf Ref.df Chi.sq
## s(Year2)                                            1.648  2.017  52.34
## s(Year2):classContructionnon-factive.1st present (FP) 6.428  7.508 143.56
## s(Year2):classContructionfactive.other (O)            1.088  1.148  55.55
## s(Year2):classContructionnon-factive.other (O)        4.499  5.485  52.87
##                                                       p-value
## s(Year2)                                             <2e-16 ***
## s(Year2):classContructionnon-factive.1st present (FP)  <2e-16 ***
## s(Year2):classContructionfactive.other (O)             <2e-16 ***
## s(Year2):classContructionnon-factive.other (O)         <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =   0.34   Deviance explained = 27.8%
## -REML = 7235.7  Scale est. = 1         n = 5063
```

Checking the model results:

```
gam.check(final.model)
```








```
##
## Method: REML   Optimizer: outer newton
## full convergence after 7 iterations.
## Gradient range [-0.005552765,-0.0002713495]
## (score 7235.688 & scale 1).
## Hessian positive definite, eigenvalue range [0.002436445,1.880846].
## Model rank =  40 / 40
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##                                                       k'  edf k-index p-value
## s(Year2)                                            9.00 1.65    0.91  <2e-16
## s(Year2):classContructionnon-factive.1st present (FP) 9.00 6.43    0.91  <2e-16
## s(Year2):classContructionfactive.other (O)          9.00 1.09    0.91  <2e-16
## s(Year2):classContructionnon-factive.other (O)      9.00 4.50    0.91  <2e-16
##
## s(Year2)                                            ***
## s(Year2):classContructionnon-factive.1st present (FP) ***
## s(Year2):classContructionfactive.other (O)          ***
## s(Year2):classContructionnon-factive.other (O)      ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

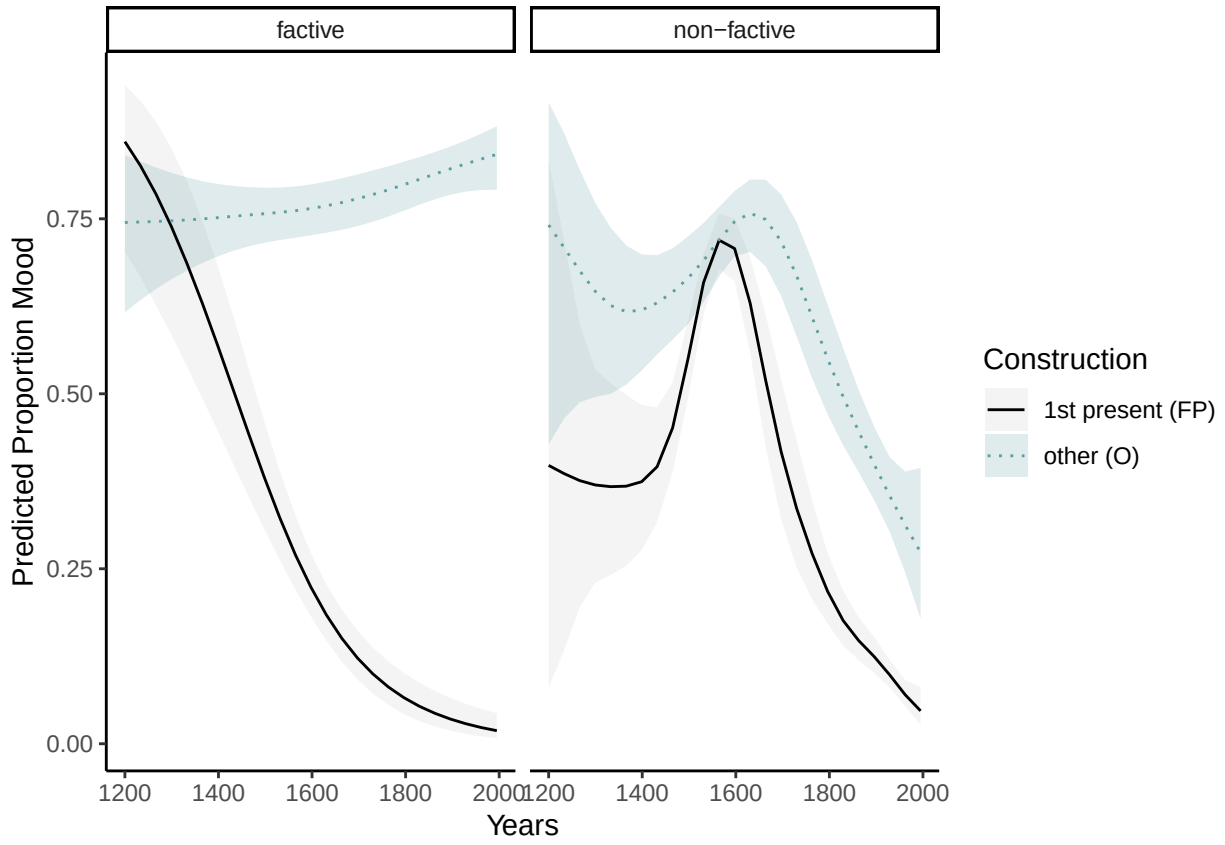
```
k.check(final.model)
```

```
##                                                       k'      edf   k-index
```

```
## s(Year2)                                                 9 1.647515 0.9149969
## s(Year2):classContructionnon-factive.1st present (FP)  9 6.427591 0.9149969
## s(Year2):classContructionfactive.other (O)             9 1.087964 0.9149969
## s(Year2):classContructionnon-factive.other (O)         9 4.498703 0.9149969
##                                                       p-value
## s(Year2)                                                    0
## s(Year2):classContructionnon-factive.1st present (FP)       0
## s(Year2):classContructionfactive.other (O)                  0
## s(Year2):classContructionnon-factive.other (O)              0
```

```
plot.binomial <- plot_smooths(model = final.model,
             series = Year2,
             comparison = Construction,
             facet_terms = Mverbclass2,
             split = list(classContruction = c( "Mverbclass2","Construction")),
             transform = invlogit # the model log transforms the data, to plot the odds ratio we use th
             ) +
  scale_color_manual(values = c("black", "#5F9EA0"), name = "Construction") +
  scale_fill_manual(values = c("gray80", "#5F9EA0"), name = "Construction") +
  scale_linetype_manual(values = c("solid", "dotted"), name = "Construction") +
  theme_classic() +
  ylab("Predicted Proportion Mood") +
  xlab("Years")

plot.binomial
```

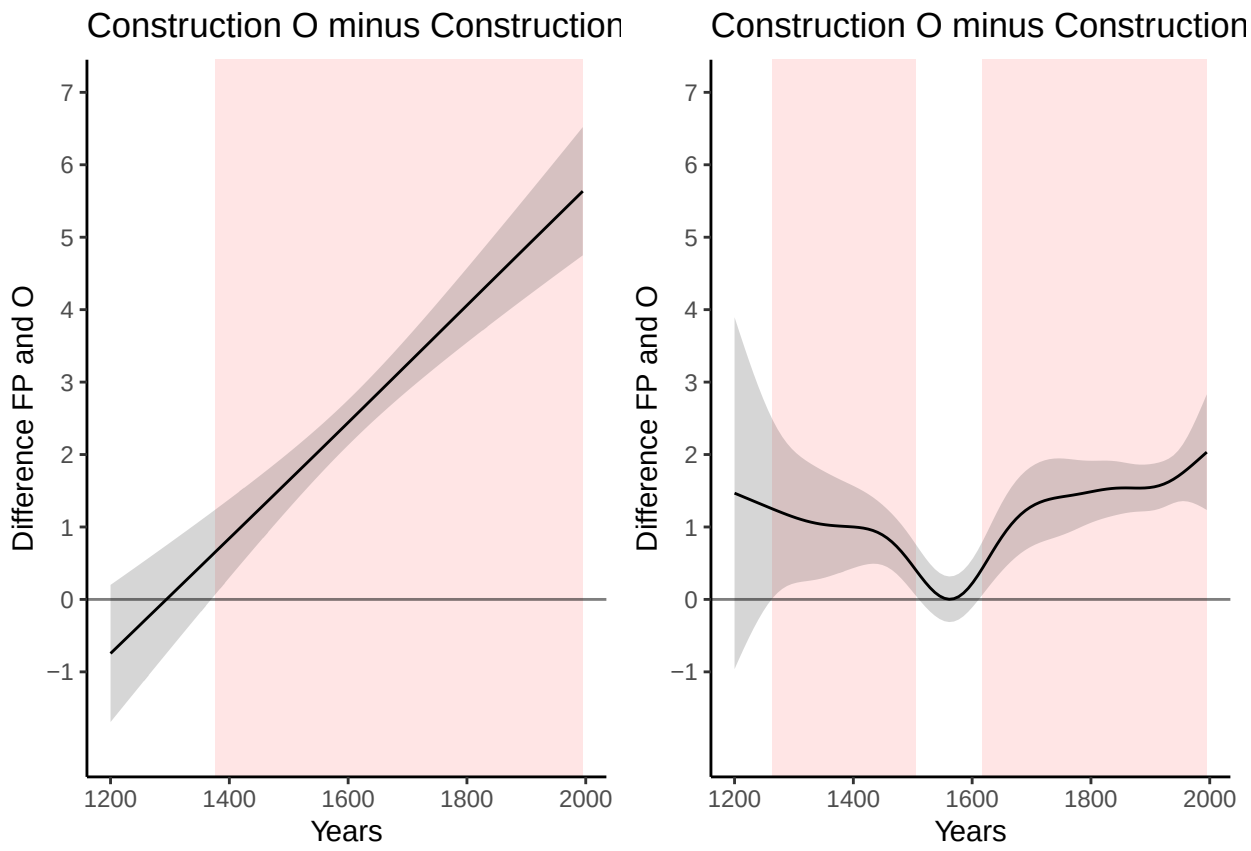
Ploting the GAM




```
#saving the plot:
ggsave(plot.binomial, file="plot-gam-binomial.png", width = 8, height = 4)
```

```
# Plotting the difference curve:

combined <- ggarrange(plot_difference(final.model, Year2,
                      difference= list(classContruction =
                                         c("factive.other (O)",
                                           "factive.1st present (FP)"))) +
            scale_y_continuous(limits=c(-2, 7), breaks = seq(-1, 7, by = 1))+
            ylab("Difference FP and O") +
            xlab("Years") +
            theme_classic() +
            ggtitle("Construction O minus Construction FP | semi-factive verbs"),
            plot_difference(final.model, Year2,
                      list(classContruction = c("non-factive.other (O)",
                                                "non-factive.1st present (FP)"))) +
             scale_y_continuous(limits=c(-2, 7),breaks = seq(-1, 7, by = 1))+
            ylab("Difference FP and O") +
            xlab("Years") +
            theme_classic() +
            ggtitle("Construction O minus Construction FP | non-factive verbs"))
combined
```

Difference curve



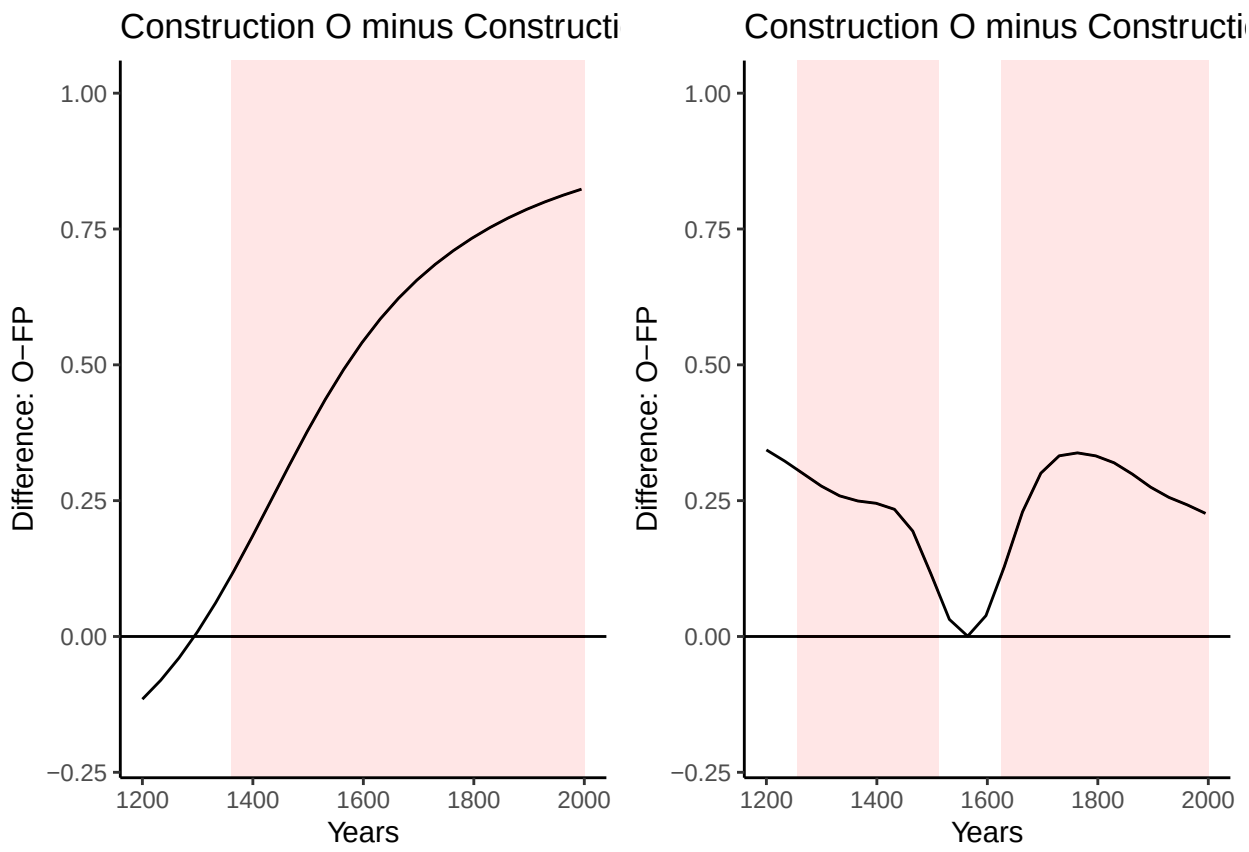

```
ggsave(combined, file="combined.png", width = 11, height = 4)
```

Plotting the Difference Curve with a change in the scale

```
#Predictions of the model:
preds <- get_gam_predictions(final.model,
                             Year2,
                             split =
                  list(classContruction = c("Verb Class",
                                            "Constrcution")))

# Backtransforming the values:
preds$Emood2<-invlogit(preds$Emood2)

# Change into a data-frame:
my_df <- as.data.frame(preds)

# diff: 2-1 by Year2
my_df1 <- subset(my_df, .idx=="1")
my_df2 <- subset(my_df, .idx=="2")
differenceV1 <- (my_df2$Emood2 - my_df1$Emood2)

# Diff: 4-3 by Year2
my_df3 <- subset(my_df, .idx=="3")
my_df4 <- subset(my_df, .idx=="4")
differenceV2 <- (my_df4$Emood2 - my_df3$Emood2)
```

```r
# Years
Years <- my_df3$Year2

# V1 significance level: 1360 onwards significant
significanceV1 <- c(FALSE,FALSE,FALSE,FALSE,FALSE,
                    TRUE, TRUE, TRUE, TRUE, TRUE,
                    TRUE, TRUE, TRUE, TRUE, TRUE,
                    TRUE, TRUE, TRUE, TRUE, TRUE,
                    TRUE, TRUE, TRUE, TRUE, TRUE)

# V2 significance level:
significanceV2 <- c(FALSE,FALSE,FALSE,
                    TRUE, TRUE, TRUE, TRUE,
                    TRUE, TRUE, TRUE,
                    FALSE,FALSE,FALSE,
                    TRUE, TRUE, TRUE, TRUE, TRUE,
                    TRUE, TRUE, TRUE, TRUE, TRUE,
                    TRUE,TRUE)

# combining all the above
newdata <- cbind(Years,differenceV1,significanceV1,differenceV2,significanceV2)
newdata <- as.data.frame(newdata) #transform into data frame


#we can now plot these differences:
#Plot 1:
 PlotV1 <- ggplot(newdata, aes(Years, differenceV1))+
   geom_line()+
   geom_hline(aes(yintercept = 0), colour = "black") +
  geom_rect(aes(xmin = 1360, xmax = 2000, ymin = -Inf, ymax = Inf),
                  fill = "red", alpha = 0.005)+
   ylim(-0.2,1)+
   ylab("Difference: O-FP")+
   ggtitle("Construction O minus Construction FP|semi-factive verbs")+
   theme_bw() +
   theme(panel.border = element_blank(), panel.grid.major = element_blank(),
panel.grid.minor = element_blank(), axis.line = element_line(colour = "black"))

 # Plot 2:
 PlotV2 <- ggplot(newdata, aes(Years, differenceV2))+
   geom_line()+
    geom_hline(aes(yintercept = 0), colour = "black") +
   geom_rect(aes(xmin = 1256, xmax = 1512, ymin = -Inf, ymax = Inf),
                  fill = "red", alpha = 0.005)+
   geom_rect(aes(xmin = 1625, xmax = 2000, ymin = -Inf, ymax = Inf),
                  fill = "red", alpha = 0.005)+
   ylim(-0.2,1)+
   ylab("Difference: O-FP")+
   ggtitle("Construction O minus Construction FP|non-factive verbs")+
   theme_bw() +
   theme(panel.border = element_blank(), panel.grid.major = element_blank(),
panel.grid.minor = element_blank(), axis.line = element_line(colour = "black"))
```

```r
#combining the plots
combinedraw <- ggarrange(PlotV1,PlotV2)
combinedraw
```




```r
ggsave(combinedraw,file="combined-difference-curve.png", width = 11, height = 4)
```

3.2. GAM: exploring the effect of individual predicates

```r
# Create a new column with Class and Construction as one combined factors:
clean.data <- mutate(clean.data, PredicateConstruction = interaction(Mverbl, Construction))

#use a contrast treatment and order the variable:
clean.data$PredicateContruction  <- as.ordered(clean.data$PredicateConstruction )
contrasts(clean.data$PredicateConstruction ) <- "contr.treatment"

model.predicates <- bam(Emood2 ~ PredicateConstruction  +
                      s(Year2) +
                      s(Year2, by = PredicateConstruction),
                    method = "REML",
                    family = "binomial",
                    data = clean.data)

#comparing the models:
AIC(final.model,model.predicates)
```

```
##                          df      AIC
```

```
## final.model      20.15714 5045.599
## model.predicates 43.01642 4873.139
```

The model in which individual predicates rather than verb class is significantly better. It must be said that the data is considerably less for each individual predicate. The work leaves open the possibility of exploring theoretical specific difference across verbs within the same verbal class.

4. Construction FP: semi-factive

Data for first person construction semi-factive verbs:

```
# Subseting the data:

clean.data.factive <- subset(clean.data,
                      Mverbclass=="factive"& # Only factive verbs
                      Construction=="1st present (FP)"
                     )

plot.data.verbs.factive <- ddply(clean.data.factive, .(Period,Mverbl),
                         summarize,
                         mean = mean(as.numeric(as.character(Emood2)), na.rm = T),
                         n = sum(!is.na(as.numeric(as.character(Emood2))))
                         )
# know:
clean.data.know <- subset(clean.data,
                      Mverbl=="saber (know)"& # Only the verb know
                      Construction=="1st present (FP)"
                     )
# see:
clean.data.see <- subset(clean.data,
                      Mverbl=="ver (see)"& # Only the verb see
                      Construction=="1st present (FP)"
                     )
```

```
# Logarithm:
logarithm <- function(Period,k,s)(1/(1+exp(s*(Period-k))))

# model:
model <- nls(Emood2 ~ logarithm(Period,k,s), data=clean.data.know, start=list(s=0.024, k=1439))
summary(model)
```

First attempt: fitting all data

```
##
## Formula: Emood2 ~ logarithm(Period, k, s)
##
## Parameters:
##    Estimate Std. Error t value Pr(>|t|)
## s 1.864e-02  3.094e-03   6.026 4.54e-09 ***
## k 1.428e+03  1.170e+01 122.069  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2959 on 325 degrees of freedom
##
```

```
## Number of iterations to convergence: 16
## Achieved convergence tolerance: 9.899e-06
```

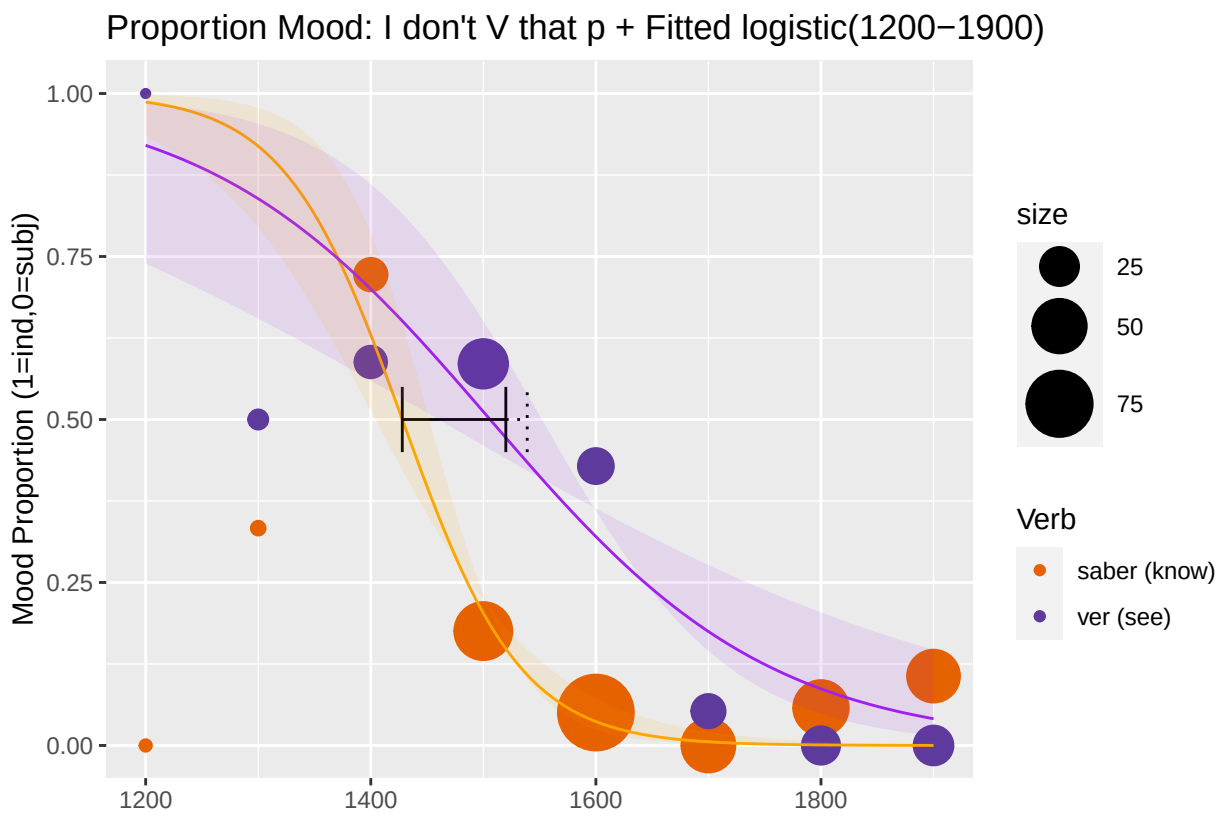
```
#values of the model: 0.019+-0.003 [0.22-0.016], k=1428+-12 [1440-]

modelsee <- nls(Emood2 ~ logarithm(Period,k,s), data=clean.data.see, start=list(s=0.008, k=1505))
summary(modelsee)
```

```
##
## Formula: Emood2 ~ logarithm(Period, k, s)
##
## Parameters:
##    Estimate Std. Error t value Pr(>|t|)
## s 7.881e-03  1.656e-03   4.759 4.46e-06 ***
## k 1.506e+03  2.267e+01  66.438  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3922 on 153 degrees of freedom
##
## Number of iterations to convergence: 8
## Achieved convergence tolerance: 3.692e-06
```

```
#values of the model: s=0.008+-0.002[0.01-0.006] k=1506.+-23
```

```
clean.data.know$Period<-as.numeric(clean.data.know$Period)
plot.data.verbs.factive$Period <-as.numeric(plot.data.verbs.factive$Period)

# Functions for plotting
logsaber <-function(x) {1/(1+exp(0.019*(x-1428)))}
logsaberupper <-function(x) {1/(1+exp(0.025*(x-1452)))}
logsaberlower <-function(x) {1/(1+exp(0.013*(x-1404)))}


logver <-function(x) {1/(1+exp(0.008*(x-1506)))}
logverupper <-function(x) {1/(1+exp(0.012*(x-1552)))}
logverlower <-function(x) {1/(1+exp(0.004*(x-1460)))}

#Plotting parameters:

x <- seq(1200,1900,10)
y1ver <- logverlower(x)
y2ver <- logverupper(x)

y1know <-logsaberlower(x)
y2know <- logsaberupper(x)

# Plot:

mood.verbs.factive <- ggplot()+
              geom_point(data=plot.data.verbs.factive,
                         aes(Period, mean,size = n, color=Mverbl))+scale_size_area(max_size=13)+
              labs(title="Proportion Mood: I don't V that p + Fitted logistic(1200-1900)",
                    x = " ",
                    y="Mood Proportion (1=ind,0=subj)")+
```

```
              scale_color_manual(values=c("#E66100", "#5D3A9B" ))+
              geom_function(fun=logsaber, color = "orange")+
               geom_function(fun=logver, color = "purple")+
             ylim(0,1)+
             labs(size="size", colour="Verb")+
             geom_segment(aes(x = 1428, y = 0.50, xend = 1520, yend = 0.50))+
             geom_segment(aes(x=1428, y=0.45, xend = 1428, yend=0.55))+
             geom_segment(aes(x=1520, y=0.45, xend = 1520, yend=0.55))+
             geom_segment(aes(x = 1520, y = 0.50, xend = 1539, yend = 0.50), linetype="dotted")+
             geom_segment(aes(x=1539, y=0.45, xend = 1539, yend=0.55),linetype="dotted")+
             geom_polygon(aes(c(x,rev(x)),c(y2ver,rev(y1ver))),fill="purple",alpha=0.1)+
             geom_polygon(aes(c(x,rev(x)),c(y2know,rev(y1know))),fill="orange",alpha=0.1)
mood.verbs.factive
```




```
#Fit the data from 1400 onwards only:

clean.data.know.1400 <- subset(clean.data,
                    Mverbl=="saber (know)"& # Only know
                    Construction=="1st present (FP)"& # only first person
                      Period!="1200"& # data from 1400 onwards
                    Period!="1300"
                   )

modelknow1400 <- nls(Emood2 ~ logarithm(Period,k,s), data=clean.data.know.1400, start=list(s=0.024, k=1
```

```
summary(modelknow1400)
```

Second Attempt: fitting only the drop

```
##
## Formula: Emood2 ~ logarithm(Period, k, s)
##
## Parameters:
##    Estimate Std. Error t value Pr(>|t|)
## s 2.326e-02  3.647e-03   6.378 6.29e-10 ***
## k 1.438e+03  9.284e+00 154.847  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2778 on 320 degrees of freedom
##
## Number of iterations to convergence: 4
## Achieved convergence tolerance: 6.72e-06
```

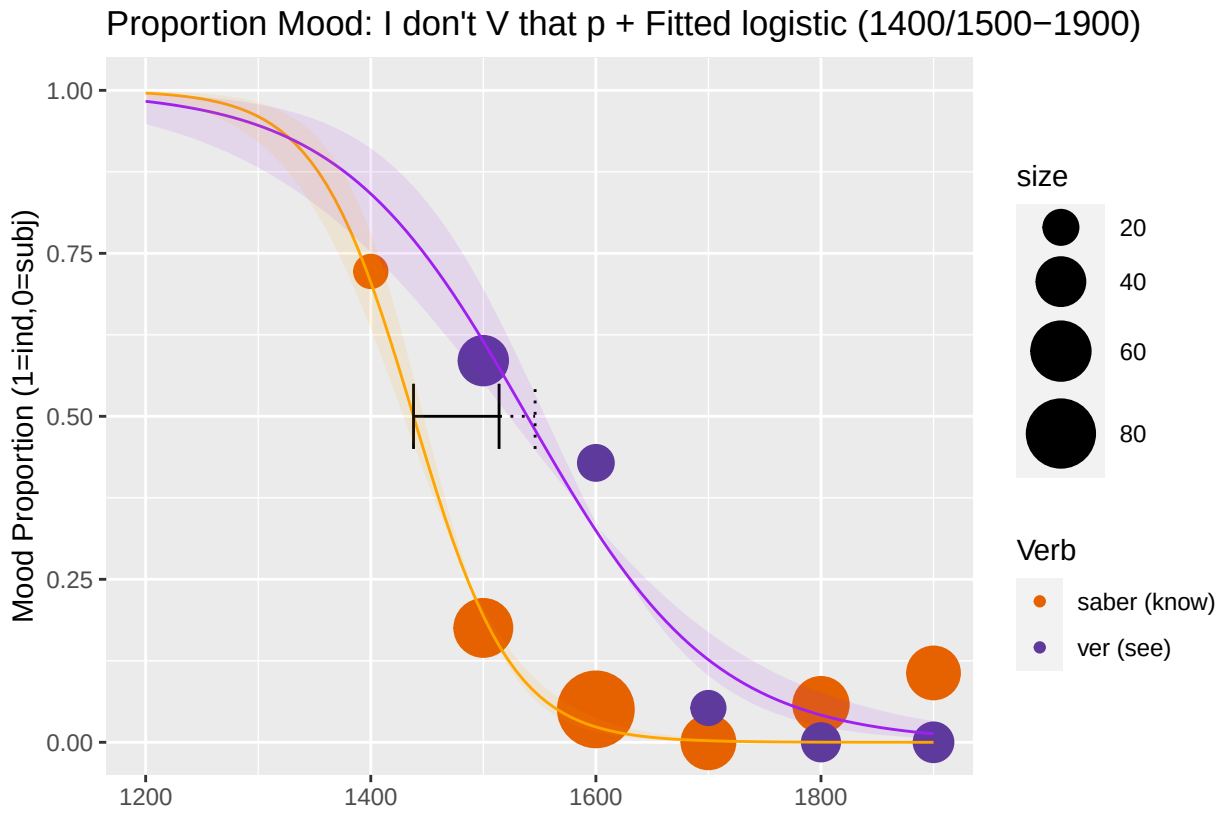
```
clean.data.see.1500<- subset(clean.data,
                    Mverbl=="ver (see)"& # Only the verb see
                    Construction=="1st present (FP)"& # only first person constructions
                    Period!="1200"& # data from 1500 onwards.
                    Period!="1300"&
                    Period!="1400"
                   )

modelsee <- nls(Emood2 ~ logarithm(Period,k,s), data=clean.data.see.1500, start=list(s=0.008, k=1505))
summary(modelsee)
```

```
##
## Formula: Emood2 ~ logarithm(Period, k, s)
##
## Parameters:
##    Estimate Std. Error t value Pr(>|t|)
## s 1.184e-02  2.758e-03   4.294 3.42e-05 ***
## k 1.539e+03  1.595e+01  96.458  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3571 on 129 degrees of freedom
##
## Number of iterations to convergence: 9
## Achieved convergence tolerance: 3.911e-06
```

Plotting the results

```
logsaber1500 <-function(x) {1/(1+exp(0.023*(x-1438)))}
logsaber1500upper <-function(x) {1/(1+exp(0.027*(x-1447)))}
logsaber1500lower <-function(x) {1/(1+exp(0.019*(x-1429)))}

logver1500 <-function(x) {1/(1+exp(0.012*(x-1539)))}
logver1500lower <-function(x) {1/(1+exp(0.009*(x-1523)))}
logver1500upper <-function(x) {1/(1+exp(0.015*(x-1555)))}

#Calculating the error upper bouns: 0.019=> 92 years(1429+117=1546), 0.023=>76 years(1438+76=1514);
```

```r
x <- seq(1200,1900,10)
y1ver1500 <- logver1500lower(x)
y2ver1500 <- logver1500upper(x)

y1know1400 <-logsaber1500lower(x)
y2know1400 <- logsaber1500upper(x)

#delete points not used
plot.data.verbs.factive2 <- subset(plot.data.verbs.factive, Period!="1200" &
                                     Period!="1300"
                                   )
plot.data.verbs.factive3 <- filter(plot.data.verbs.factive2, (Period!="1400"| Mverbl=="saber (know)"))

# Plotting the graph:

moodverbs.factive1400 <- ggplot()+
              geom_point(data=plot.data.verbs.factive3,
                         aes(Period, mean,size = n, color=Mverbl))+scale_size_area(max_size=13)+
              labs(title="Proportion Mood: I don't V that p + Fitted logistic (1400/1500-1900)",
                    x = " ",
                    y="Mood Proportion (1=ind,0=subj)")+
             scale_color_manual(values=c("#E66100", "#5D3A9B" ))+
              geom_function(fun=logsaber1500, color = "orange")+
              #geom_function(fun=logsaber1500upper, color = "orange", linetype="dotted")+
              #geom_function(fun=logsaber1500lower, color = "orange", linetype="dotted")+
              geom_function(fun=logver1500, color = "purple")+
              #geom_function(fun=logver1500lower, color = "purple", linetype="dotted")+
              #geom_function(fun=logver1500upper, color = "purple", linetype="dotted")+
              #stat_function(fun=logver1500upper, geom="area", fill="#84CA72", alpha=0.2)+
            ylim(0,1)+
            labs(size="size", colour="Verb")+
            geom_segment(aes(x = 1438, y = 0.50, xend = 1514, yend = 0.50))+
            geom_segment(aes(x=1438, y=0.45, xend = 1438, yend=0.55))+
            geom_segment(aes(x=1514, y=0.45, xend = 1514, yend=0.55))+
            geom_segment(aes(x = 1514, y = 0.50, xend = 1546, yend = 0.50),color="black",linetype="dotte
            geom_segment(aes(x=1546, y=0.45, xend = 1546, yend=0.55),linetype="dotted")+
            geom_polygon(aes(c(x,rev(x)),c(y2ver1500,rev(y1ver1500))),fill="purple",alpha=0.1)+
  geom_polygon(aes(c(x,rev(x)),c(y2know1400,rev(y1know1400))),fill="orange",alpha=0.1)
moodverbs.factive1400
```




```
combined.models <- ggarrange(mood.verbs.factive,moodverbs.factive1400)
```

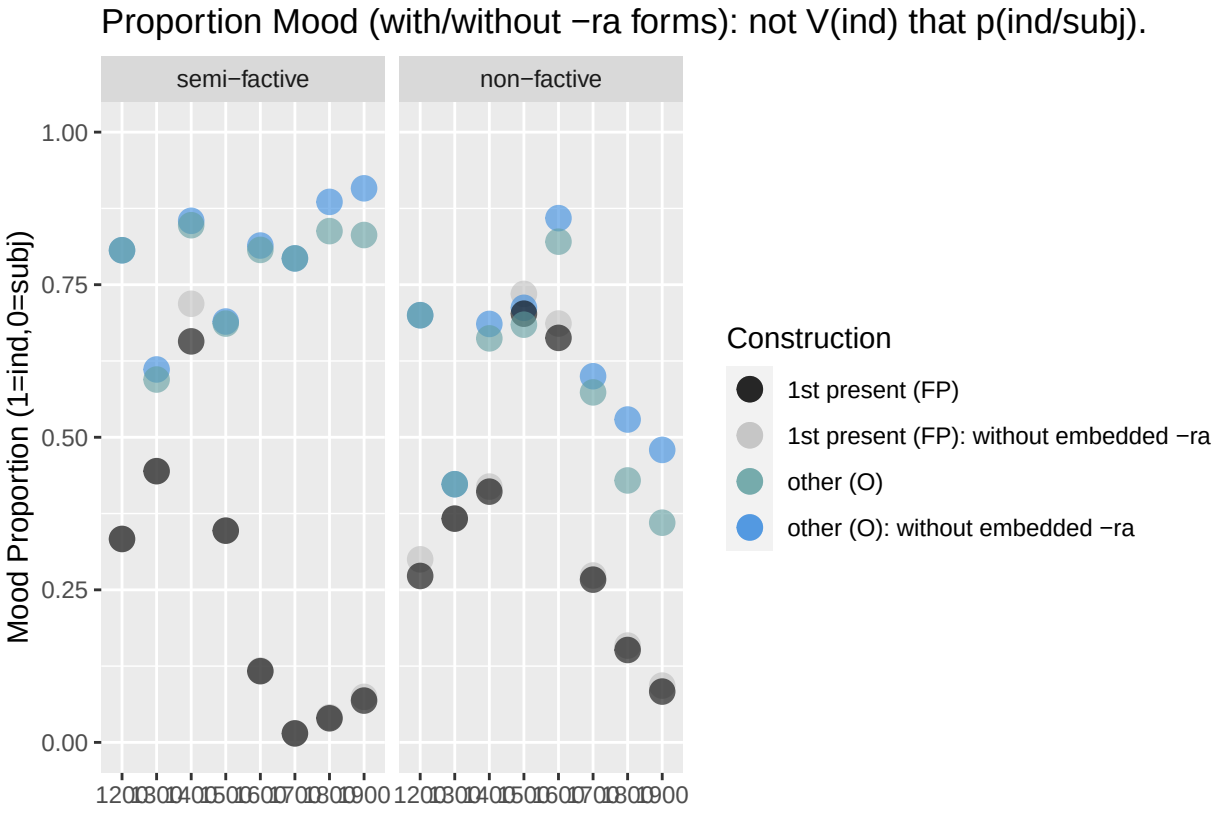
```
ggsave(combined.models, file="fitted-loss-1stperson2.png",width = 13,height = 4)
```

Combining the two attempts

Appendix I: -ra subjunctive forms

```
ex.datarase <- subset(data, Mverbtype =="ind" &
                      Emood !="Na" &
                      Emood !="NA" &
                      Emood !="inf" &
                      Emoodrase !="ra" & # excludes the -ra forms
                      MClauseType=="noninterrogative" &
                      EClauseType=="unambiguous"
                    )

# Changing the indicative to 1 and subjunctive to 0:
ex.datarase$Emood2 <- ifelse(ex.datarase$Emood == "subj", 0, 1)

#changing name factive and non-factive
ex.datarase$Mverbclass2 <- ifelse(ex.datarase$Mverbclass == "factive", "semi-factive", "non-factive")
ex.datarase$Construction <- ifelse(ex.datarase$Construction == "1st present", "1st present (FP): withou
```

```r
# Converting the into a numeric value:
ex.datarase$Emood2 <- as.numeric(as.character(ex.datarase$Emood2))
ex.datarase$Period <- as.factor(as.character(ex.datarase$Period))

# new dataframe with the means of embedded mood per period, verb and type of matrix subject(persona)
plot.datafactiverase <- ddply(ex.datarase, .(Period, Construction,Mverbclass2
),summarize, mean = mean(as.numeric(as.character(Emood2)), na.rm = T), n = sum(!is.na(as.numeric(as.cha


# original data:
clean.data$Mverbclass2 <- ifelse(clean.data$Mverbclass == "factive", "semi-factive", "non-factive")
plot.datafactive <- ddply(clean.data, .(Period, Construction,Mverbclass2),
                          summarize,
                          mean = mean(as.numeric(as.character(Emood2)), na.rm = T),
                          n = sum(!is.na(as.numeric(as.character(Emood2))))
                          )
plot.datafactive$Period <- as.factor(plot.datafactive$Period)


moodfactiverase <- ggplot()+
  geom_point(data=plot.datafactiverase, aes(Period, mean, color=Construction), alpha=0.6, size=4)+
  geom_point(data=plot.datafactive, aes(Period, mean, color=Construction), alpha=0.6, size=4)+
  facet_wrap(~factor(Mverbclass2, c("semi-factive","non-factive")), ncol=2)+scale_size_area(max_size=13
  labs(title="Proportion Mood (with/without -ra forms): not V(ind) that p(ind/subj).",
       x = " ",
       y="Mood Proportion (1=ind,0=subj)")+
  scale_color_manual(values=c("black","gray", "#5F9EA0", "#3589de"))+
  ylim(0,1)+
  labs(colour="Construction")


moodfactiverase
```

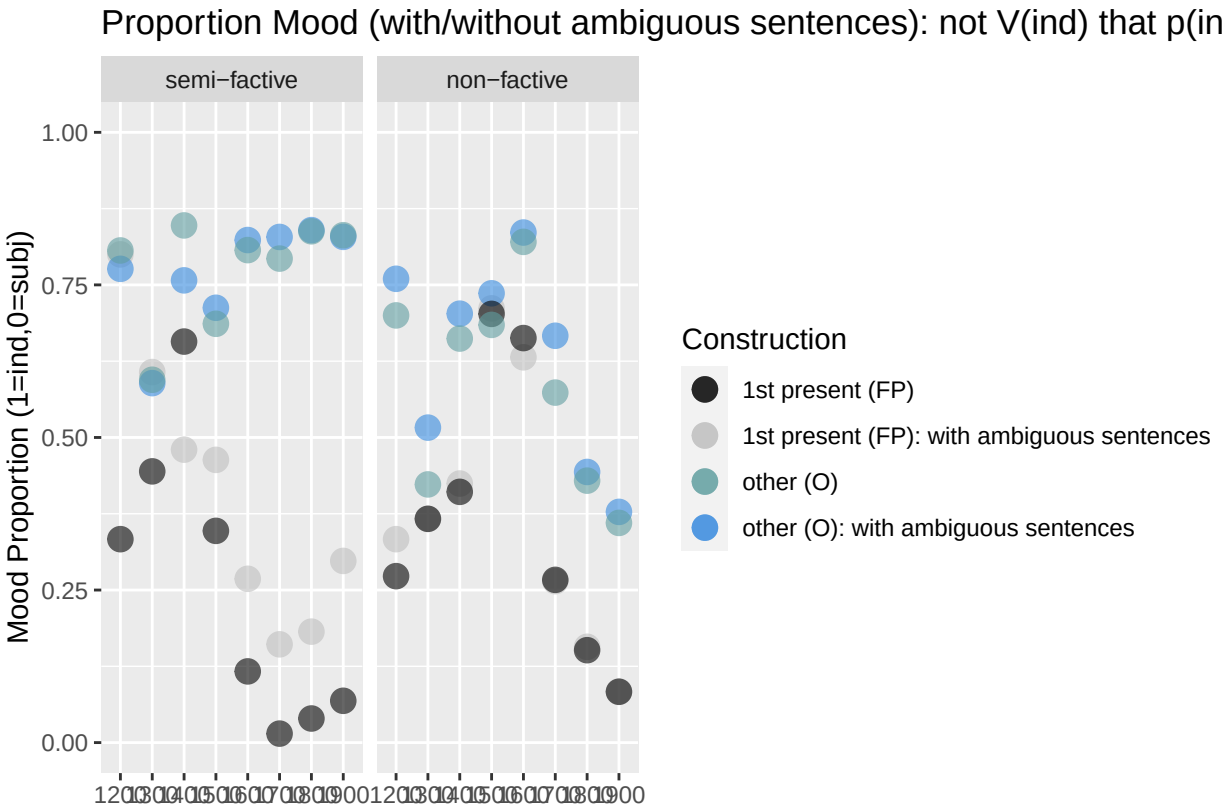



```
ggsave(moodfactiverase, file="diachrony-verb-classes-rase2.png", width = 11, height = 4)
```

Appendix II: ambiguous sentences

```
ex.data.ambiguous <- subset(data, Mverbtype =="ind" &
                      Emood !="Na" &
                      Emood !="NA" &
                      Emood !="inf" &
                      MClauseType=="noninterrogative"
                    )

# Changing the indicative to 1 and subjunctive to 0:
ex.data.ambiguous$Emood2 <- ifelse(ex.data.ambiguous$Emood == "subj", 0, 1)

#changing name factive and non-factive
ex.data.ambiguous$Mverbclass2 <- ifelse(ex.data.ambiguous$Mverbclass == "factive", "semi-factive", "non
ex.data.ambiguous$Construction <- ifelse(ex.data.ambiguous$Construction == "1st present", "1st present

# Converting the into a numeric value:
ex.data.ambiguous$Emood2 <- as.numeric(as.character(ex.data.ambiguous$Emood2))
ex.data.ambiguous$Period <- as.factor(as.character(ex.data.ambiguous$Period))

# new dataframe with the means of embedded mood per period, verb and type of matrix subject(persona)
plot.datafactive.ambiguous <- ddply(ex.data.ambiguous, .(Period, Construction,Mverbclass2
),summarize, mean = mean(as.numeric(as.character(Emood2)), na.rm = T), n = sum(!is.na(as.numeric(as.cha
```

```
moodfactiveambiguous <- ggplot()+
  geom_point(data=plot.datafactive.ambiguous, aes(Period, mean, color=Construction), alpha=0.6, size=4)+
  geom_point(data=plot.datafactive, aes(Period, mean, color=Construction), alpha=0.6, size=4)+
  facet_wrap(~factor(Mverbclass2, c("semi-factive","non-factive")), ncol=2)+scale_size_area(max_size=13)
  labs(title="Proportion Mood (with/without ambiguous sentences): not V(ind) that p(ind/subj).",
       x = " ",
       y="Mood Proportion (1=ind,0=subj)")+
  scale_color_manual(values=c("black","gray", "#5F9EA0", "#3589de"))+
  ylim(0,1)+
  labs(colour="Construction")

moodfactiveambiguous
```




```
ggsave(moodfactiveambiguous, file="diachrony-verb-classes-ambiguous2.png", width = 11, height = 4)
```