The Development of Polarity Subjunctive

Raquel Montero Estebaranz

03.04.2024

Contents

1. Introduction

This file shows the code used to analyse the data of the verb "say" for Chapter 5 of the thesis entitled "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
```

2. Data

Loading the Data:

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

3. The verb say

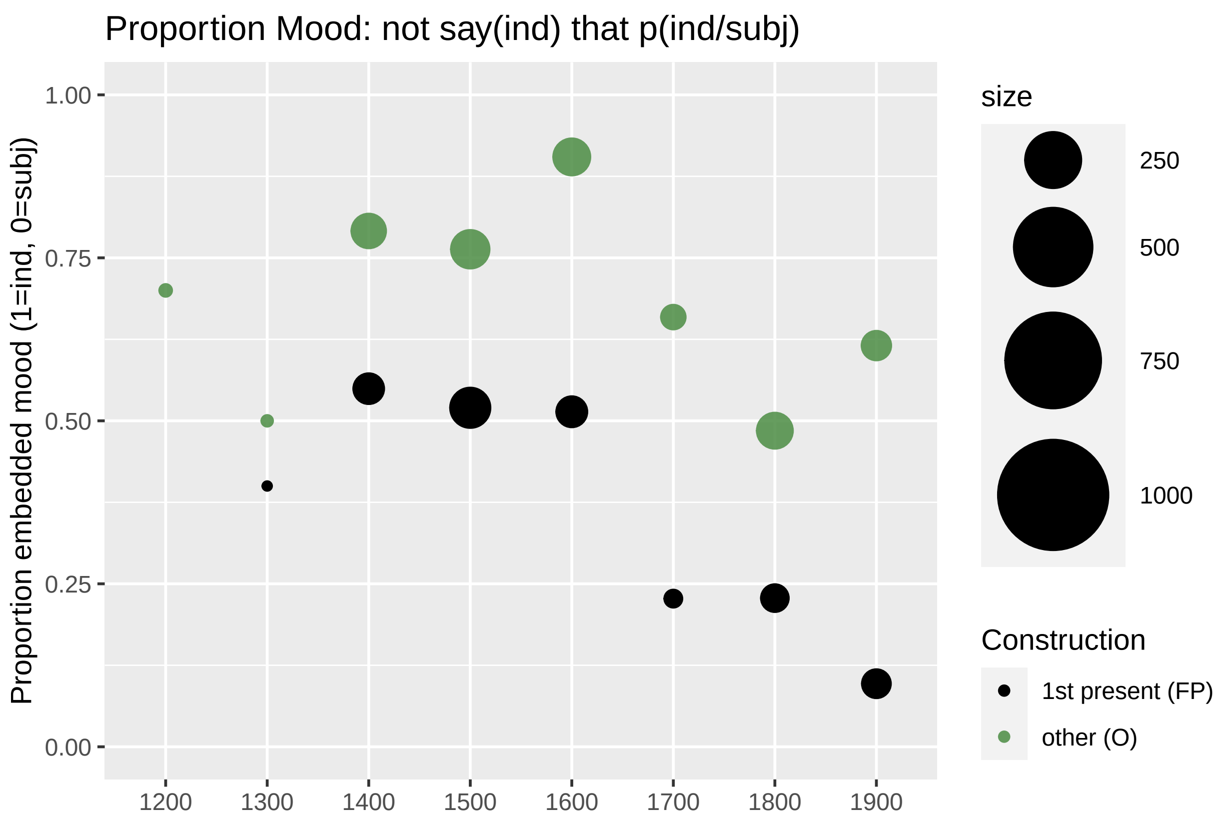
```
ex.data <- subset(data, Mverbtype =="ind" &
                    Mverbclass!="Na"& # not one of the four verbs
                    Emood !="Na" &
                    Emood !="NA" &
                    Emood !="inf" &
                    MClauseType=="noninterrogative" &
                    EClauseType=="unambiguous"&
                    Mverbl=="decir (say)"
                  )

# Changing the indicative to 1 and subjunctive to 0:
ex.data$Emood2 <- ifelse(ex.data$Emood == "subj", 0, 1)
# Converting the into a numeric value:
ex.data$Emood2 <- as.numeric(as.character(ex.data$Emood2))

#changing names of values so that they are better for plotting:
ex.data$Mverbtype2 <- ifelse(ex.data$Mverbclass == "factive", "Semi-factive", "Non-factive)")
ex.data$Construction <- ifelse(ex.data$Construction == "1st present", "1st present (FP)", "other (O)")
```

Data for plotting:

```
# new dataframe with the means of embedded mood per period, verb class/verb and construction:

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

Plotting the results:

```
# Plot 1:
plot.data.say$Period <- as.factor(plot.data.say$Period) # Period as factor

plot.say <- ggplot()+
                geom_point(data=plot.data.say,
                           aes(Period, mean, size = n, color=Construction))+ #main data
                scale_size_area(max_size=20,limits=c(1,1100))+
                labs(title="Proportion Mood: not say(ind) that p(ind/subj)",  # axis
                       x = "",
                      y="Proportion embedded mood (1=ind, 0=subj)")+
                scale_color_manual(values=c("black", "#4c8c44d9" ))+ # colors
                labs(size="size", colour="Construction")+ # labels legends
                ylim(0,1)

plot.say
```




```
#Saving the plot:
ggsave(plot.say, file="plot-say.png", width = 6.5, height= 4)
```

3.1. Speed of change: logistic

```
# Data for Construction O:
ex.dataB <- subset(ex.data, Construction=="other (O)")

# defining a logarithmic function:
logarithmB <- function(Period,k,s,m)(1/((1/m)+exp((Period-k)/s)))

# Fiting the model to the data:
#Initial values: m = 0.79; s = 163; k = 1980 (this values were obtained by using the graphic calculator
# calculating the vales with nls:
modelB <- nls(Emood2 ~ logarithmB(Period,k,s,m), data=ex.dataB, start=list(s=163, k=1980, m=0.79))
summary(modelB)
```

```
##
## Formula: Emood2 ~ logarithmB(Period, k, s, m)
##
## Parameters:
##    Estimate Std. Error t value Pr(>|t|)
## s 1.638e+02  9.035e+01   1.813   0.0704 .
## k 1.946e+03  5.056e+01  38.485   <2e-16 ***
## m 8.286e-01  6.749e-02  12.278   <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4434 on 533 degrees of freedom
##
## Number of iterations to convergence: 13
## Achieved convergence tolerance: 4.732e-06
```

```
#Obtained values -->
                    #K = 1946+-51*2= 102;
                    #s = 164 +- 90*2=180 --> [0-344];
                    #m = 0.82 +-0.07*2 = 14

# Some procedure for Construction FP:

ex.dataA <- subset(ex.data, Construction=="1st present (FP)")

logarithmA <- function(Period,k,s,m)(1/((1/m)+exp((Period-k)/s)))

modelA <- nls(Emood2 ~ logarithmA(Period,k,s,m), data=ex.dataA, start=list(s=88, k=1696, m=0.51))
summary(modelA)
```

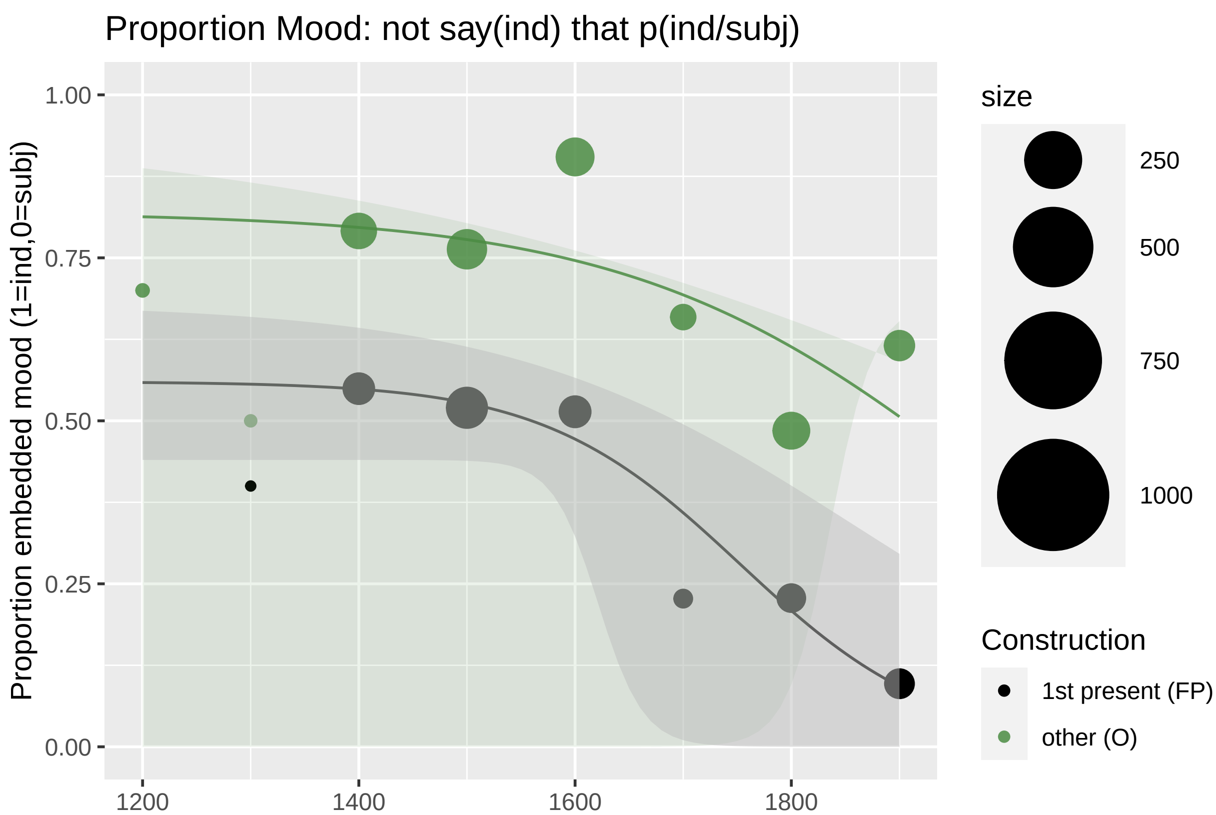
```
##
## Formula: Emood2 ~ logarithmA(Period, k, s, m)
##
## Parameters:
##    Estimate Std. Error t value Pr(>|t|)
## s 9.094e+01  3.482e+01   2.612  0.00933 **
## k 1.700e+03  4.840e+01  35.132  < 2e-16 ***
## m 5.579e-01  5.601e-02   9.961  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4626 on 411 degrees of freedom
##
## Number of iterations to convergence: 6
## Achieved convergence tolerance: 9e-06
```

```
## k =1700 +-48*2= 96=[],
#s = 91+- 35*2 =70,
# m = 0.56 +-0.06*2 =0.12
```

Ploting the model fit to the data:

```
#Main functions:
fitB <- function(x){(1/((1/0.82)+exp((x-1946)/164)))}
fitA <- function(x) {1/((1/0.56)+exp((x-1700)/91))}

#Confidence intervals for the speed of Construction O:

fitBupper <-function(x){(1/((1/0.96)+exp((x-2048)/344)))}
fitBlower <-function(x){(1/((1/0.68)+exp((x-1844)/-20)))}
#Confidence intervals for the speed of construction FP:
fitAupper <- function(x) {1/((1/0.68)+exp((x-1796)/161))}
fitAlower <- function(x) {1/((1/0.44)+exp((x-1604)/21))}

#equations for plotting the confiddence intervals:
```

```
x <- seq(1200,1900,10)
y1O <- fitBupper(x)
y2O <- fitBlower(x)

x <- seq(1200,1900,10)
y1FP <- fitAupper(x)
y2FP <- fitAlower(x)

plot.data.say$Period <- as.numeric(as.character(plot.data.say$Period))


plot.say.fit <- ggplot()+
                geom_point(data=plot.data.say,
                           aes(Period, mean, size = n, color=Construction))+#main data
                scale_size_area(max_size=20,limits=c(1,1100))+  # controls the maximum size of the poin
                geom_function(fun=fitB, color = "#4c8c44d9")+
                geom_function(fun=fitA, color = "black")+
                labs(title="Proportion Mood: not say(ind) that p(ind/subj)",  # axis
                      x = " ",
                       y="Proportion embedded mood (1=ind,0=subj)")+
                scale_color_manual(values=c("black", "#4c8c44d9" ))+ # colors
                labs(size="size", colour="Construction")+ # labels legends
                ylim(0,1)+
                geom_polygon(aes(c(x,rev(x)),c(y2O,rev(y1O))),fill="#4c8c44d9",alpha=0.1)+
                geom_polygon(aes(c(x,rev(x)),c(y2FP,rev(y1FP))),fill="gray",alpha=0.5)
                #geom_polygon(aes(c(x,rev(x)),c(y2decir,rev(y1decir))),fill="black",alpha=0.1)


plot.say.fit
```




```r
ggsave(plot.say.fit, file="say-fit-CI.png", width = 8, height = 5)
```

The confidence interval of fitting the logistic to Construction A show that there might be something wrong with the model.

3.2. The speed of change: Laplace

```r
#Define a function to fit data to:
laplace <- function(Period,m,k,s)(m*exp(-(abs(Period-k)/s)))


#NLLS:
modellaplaceB <- nls(Emood2 ~ laplace(Period,m,k,s), data=ex.dataB, start=list(m=0.43,k=1572, s=200))
summary(modellaplaceB)
```

```
##
## Formula: Emood2 ~ laplace(Period, m, k, s)
##
## Parameters:
##    Estimate Std. Error t value Pr(>|t|)
## m 8.983e-01  3.955e-02  22.716  < 2e-16 ***
## k 1.553e+03  1.736e+01  89.483  < 2e-16 ***
## s 6.373e+02  1.166e+02   5.464 7.15e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

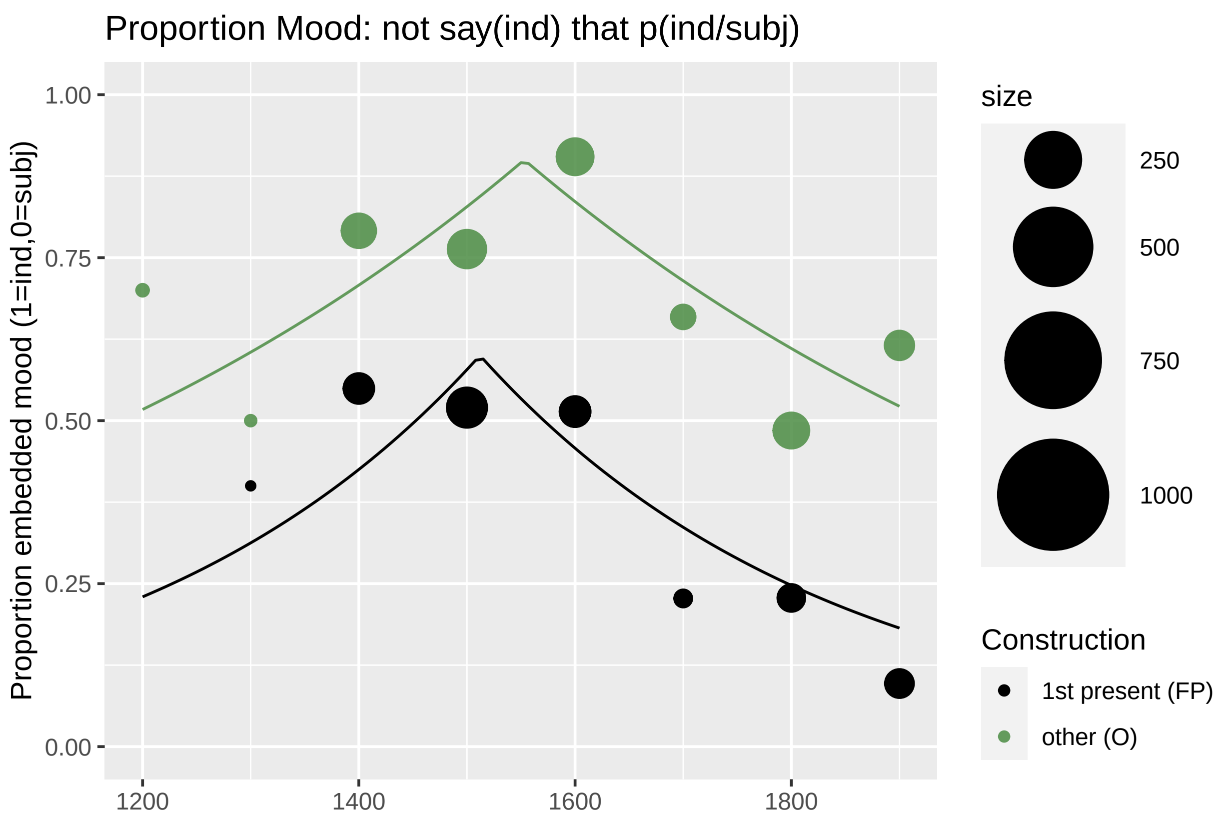
```
## Residual standard error: 0.4398 on 533 degrees of freedom
##
## Number of iterations to convergence: 4
## Achieved convergence tolerance: 6.665e-06
```

```
# m= 0.90 +-(0.04*2)0.08
# k = 1553 +-(17*2)34
# s= 637 +-(116*2)232


modellaplaceA <- nls(Emood2 ~ laplace(Period,m,k,s), data=ex.dataA, start=list(m=0.43,k=1572, s=200))
summary(modellaplaceA)
```

```
##
## Formula: Emood2 ~ laplace(Period, m, k, s)
##
## Parameters:
##    Estimate Std. Error t value Pr(>|t|)
## m 5.957e-01  5.705e-02   10.44  < 2e-16 ***
## k 1.512e+03  2.438e+01   61.99  < 2e-16 ***
## s 3.253e+02  8.341e+01    3.90 0.000112 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4674 on 411 degrees of freedom
##
## Number of iterations to convergence: 5
## Achieved convergence tolerance: 7.84e-06
```

```
# speed of change: 325 +-160(80x2) --> 485
#m= 0.60 +-(0.06*2)0.12
#k= 1512 +-(24*2)48
#s= 325+- (83*2)166
# The speeds of change given by the model also overlap they are not statistically significant from one


fitBLaplace <- function(x){0.90* exp(-(abs(x-1553)/637))}
fitALaplace <- function(x){0.60* exp(-(abs(x-1512)/325))}




plot.say.fit.laplace <- ggplot()+
                geom_point(data=plot.data.say,
                           aes(Period, mean, size = n, color=Construction))+#main data
                scale_size_area(max_size=20,limits=c(1,1100))+  # controls the maximum side of the poin
                geom_function(fun=fitBLaplace, color = "#4c8c44d9")+
                geom_function(fun=fitALaplace, color = "black")+
                labs(title="Proportion Mood: not say(ind) that p(ind/subj)",  # axis
                      x = " ",
                       y="Proportion embedded mood (1=ind,0=subj)")+
                scale_color_manual(values=c("black", "#4c8c44d9" ))+ # colors
                labs(size="size", colour="Construction")+ # labels legends
                ylim(0,1)
                #geom_polygon(aes(c(x,rev(x)),c(y2decir,rev(y1decir))),fill="black",alpha=0.1)
```

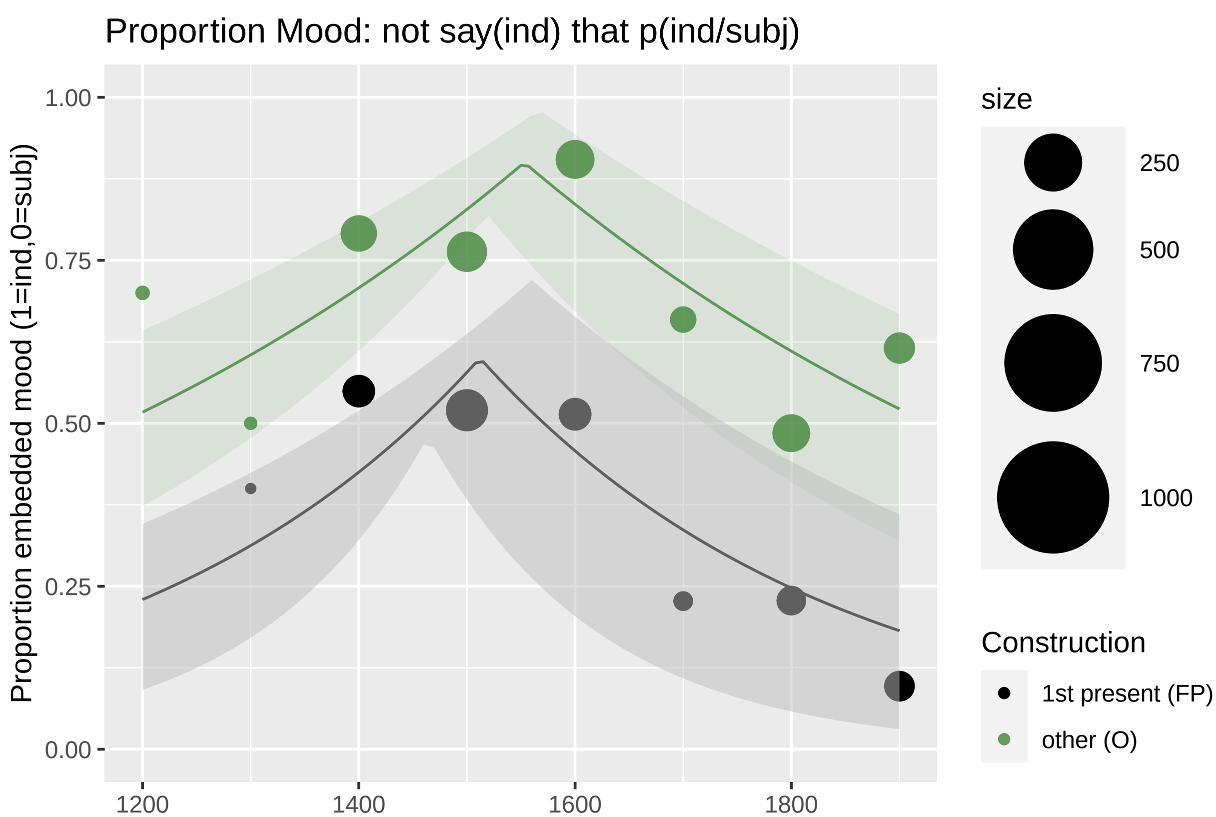
```
plot.say.fit.laplace
```



Laplace with confidence intervals:

```
#Main functions:
fitBLaplace <- function(x){0.90* exp(-(abs(x-1553)/637))}
fitALaplace <- function(x){0.60* exp(-(abs(x-1512)/325))}

#Confidence intervals for the speed of Construction O:
fitBLaplaceupper <-function(x){0.98* exp(-(abs(x-1567)/869))}
fitBLaplacelower <-function(x){0.82* exp(-(abs(x-1519)/405))}
#Confidence intervals for the speed of construction FP:
fitALaplaceupper <- function(x){0.72* exp(-(abs(x-1560)/491))}
fitALaplacelower <- function(x){0.48* exp(-(abs(x-1464)/159))}

#equations for plotting the confiddence intervals:
x <- seq(1200,1900,10)
y1Olaplace <- fitBLaplaceupper(x)
y2Olaplace <- fitBLaplacelower(x)

x <- seq(1200,1900,10)
y1FPlaplace <- fitALaplaceupper(x)
y2FPlaplace <- fitALaplacelower(x)

#plot.data.say$Period <- as.numeric(as.character(plot.data.say$Period))
```

```r
plot.say.fit.laplace.ci <- ggplot()+
                geom_point(data=plot.data.say,
                           aes(Period, mean, size = n, color=Construction))+#main data
                scale_size_area(max_size=20,limits=c(1,1100))+  # controls the maximum side of the poin
                geom_function(fun=fitBLaplace, color = "#4c8c44d9")+
                geom_function(fun=fitALaplace, color = "black")+
                labs(title="Proportion Mood: not say(ind) that p(ind/subj)",  # axis
                      x = " ",
                       y="Proportion embedded mood (1=ind,0=subj)")+
                scale_color_manual(values=c("black", "#4c8c44d9" ))+ # colors
                labs(size="size", colour="Construction")+ # labels legends
                ylim(0,1)+
              geom_polygon(aes(c(x,rev(x)),c(y2Olaplace,rev(y1Olaplace))),fill="#4c8c44d9",alpha=0.1)+
  geom_polygon(aes(c(x,rev(x)),c(y2FPlaplace,rev(y1FPlaplace))),fill="gray",alpha=0.5)
                #geom_polygon(aes(c(x,rev(x)),c(y2decir,rev(y1decir))),fill="black",alpha=0.1)


plot.say.fit.laplace.ci
```



The confidence intervals are better in the case showing that there the model is behaving as expected.

```r
ggsave(plot.say.fit.laplace, file="say-fit-laplace.png", width = 8, height = 5)
ggsave(plot.say.fit.laplace.ci, file="say-fit-laplace-ci.png", width = 8, height = 5)
```

3.3. Comparing the AIC of the models.

```
#AIC:
AIC(modelB,modellaplaceB)#Laplace is a better model with AIC 645 than logistic with 654
```

```
##                df      AIC
## modelB          4 654.2196
## modellaplaceB  4 645.4752
```

```
AIC(modelA,modellaplaceA)# Laplace is a worse model with AIC 550 vs logistic 541
```

```
##                df      AIC
## modelA          4 541.5058
## modellaplaceA  4 550.1515
```

3.4. The speed of change: Hubbert Curve

```
#Define a function to fit data to:
hubbert <- function(Period,k,m,j,r){(exp(-(Period-k)/(m)))/(1+exp(-(Period-j)/(r)))^2}



#NLLS:
modelhubbertB <- nls(Emood2 ~ hubbert(Period,k,m,j,r), data=ex.dataB, start=list(k=1349,j=1465, m=-140,
summary(modelhubbertB)
```

```
##
## Formula: Emood2 ~ hubbert(Period, k, m, j, r)
##
## Parameters:
##   Estimate Std. Error t value Pr(>|t|)
## k   1144.0      123.9   9.235  < 2e-16 ***
## j   1335.6      411.2   3.248  0.00123 **
## m   -178.3      192.3  -0.927  0.35424
## r   -237.1      108.3  -2.189  0.02900 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4411 on 532 degrees of freedom
##
## Number of iterations to convergence: 11
## Achieved convergence tolerance: 7.047e-06
```

```
# Values of the model:
# K 1144 +- 123*2
# j 1335 +- 411*2
# m -178 +- 192*2  slope for the first half of the curve (there is too little data)
# r -237 +- 108*2

modelhubbertA <- nls(Emood2 ~ hubbert(Period,k,m,j,r), data=ex.dataA, start=list(k=1349,j=1465, m=-140,
summary(modelhubbertA)
```

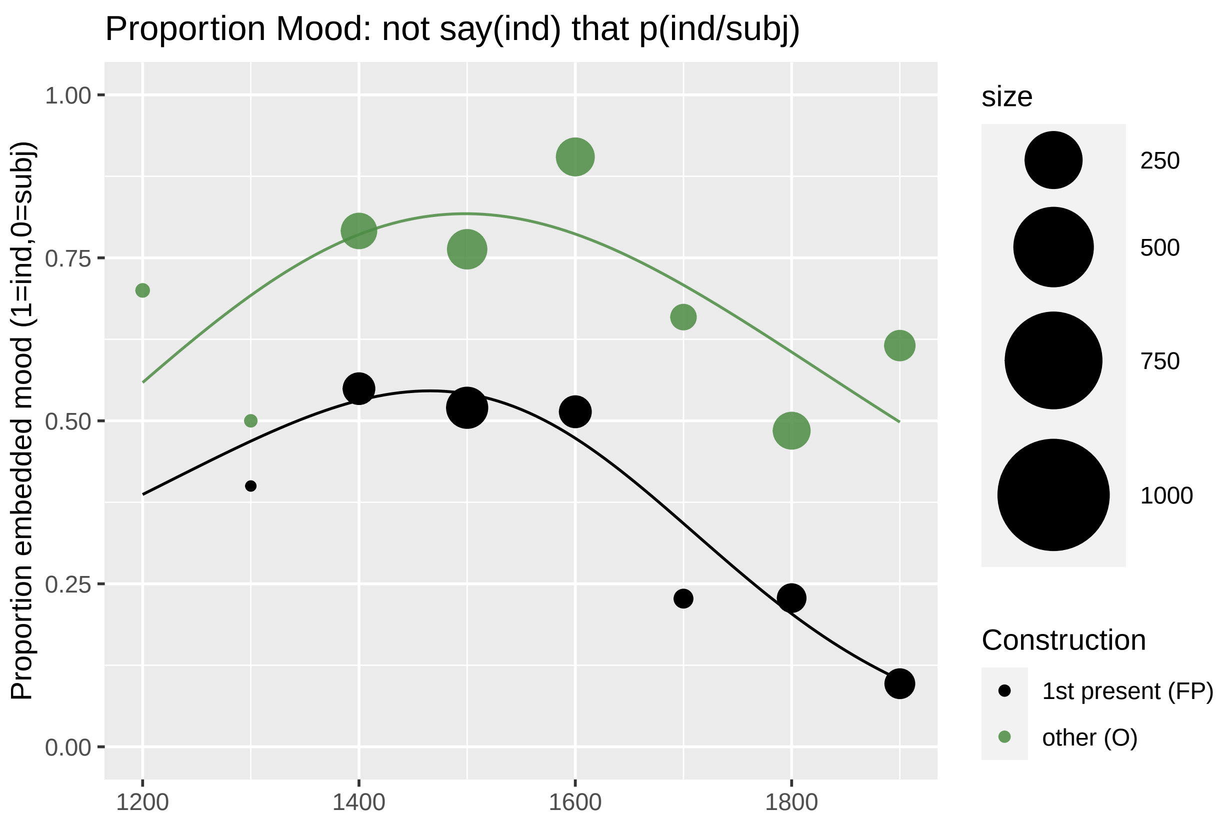
```
##
## Formula: Emood2 ~ hubbert(Period, k, m, j, r)
##
## Parameters:
##   Estimate Std. Error t value Pr(>|t|)
```

```
## k  1505.87     563.05   2.674  0.00778 **
## j  1666.13     409.73   4.066 5.73e-05 ***
## m  -360.57     765.81  -0.471  0.63801
## r  -158.09      52.87  -2.990  0.00296 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4628 on 410 degrees of freedom
##
## Number of iterations to convergence: 12
## Achieved convergence tolerance: 2.618e-06
```

```r
# k 1506 +- 563*2
# j 1666 +- 410*2
# m -361 +- 766*2 --> in the early period there is too little data
# r -158 +- 53*2
```

Plotting the curves:

```r
#Plotting the curves:

#Main functions:
fitBhubbert <- function(x){(exp(-(x-1144)/(-178)))/(1+exp(-(x-1335)/(-237)))^2}
fitAhubbert <- function(x){(exp(-(x-1505)/(-360)))/(1+exp(-(x-1666)/(-158)))^2}


plot.say.fit.hubbert <- ggplot()+
                geom_point(data=plot.data.say,
                           aes(Period, mean, size = n, color=Construction))+#main data
                scale_size_area(max_size=20,limits=c(1,1100))+  # controls the maximum side of the poin
                geom_function(fun=fitBhubbert, color = "#4c8c44d9")+
                geom_function(fun=fitAhubbert, color = "black")+
                labs(title="Proportion Mood: not say(ind) that p(ind/subj)",  # axis
                      x = " ",
                       y="Proportion embedded mood (1=ind,0=subj)")+
                scale_color_manual(values=c("black", "#4c8c44d9" ))+ # colors
                labs(size="size", colour="Construction")+ # labels legends
                ylim(0,1)


plot.say.fit.hubbert
```




```
ggsave(plot.say.fit.hubbert, file="say-fit-hubbert.png", width = 8, height = 5)
```

3.5. Comparing the models

```
AIC(modelA,modelhubbertA) # no difference in the models: 541 vs 542 less than 2 point difference
```

```
##                df      AIC
## modelA          4 541.5058
## modelhubbertA   5 542.9941
```

```
AIC(modelB, modelhubbertB) # hubbert model is preffered
```

```
##                df      AIC
## modelB          4 654.2196
## modelhubbertB   5 649.7801
```