Explanatory Document

<table>
<tr><td>Code in R</td><td>Equation(s)</td></tr>
<tr><td>

```
abnPol[1, aInd, ptN] <- ((M - 1):0) *
(parMat[ptN, 3] - parMat[ptN, 2]) / (M-1)
+ parMat[ptN, 2]

abnPol[1, nInd, ptN] <- parMat[ptN, 1]

abnPol[1, bInd, ptN] <- abnPol[1, nInd,
ptN] - abnPol[1, aInd, ptN]
```

</td><td>

Initial beliefs for each of the groups are set via free parameters $n^{(0)}$, $a^{(0)}_{min}$ and $a^{(0)}_{max}$, where $a^{(0)}_{min}$ is the α for the least accepting group, $a^{(0)}_{max}$ is the α for the most accepting group, and n is the number of groups. For group M, where M = 1, 2, 3 and 4,

$$\alpha^{(0)}_{M} = \alpha^{(0)}_{min} + (M-1)(\alpha^{(0)}_{max} - \alpha^{(0)}_{min})/3$$
$$n^{(0)}_{M} = n^{(0)}$$
$$\beta^{(0)}_{M} = n^{(0)}_{M} - \alpha^{(0)}_{M}$$

Equation (13)

</td></tr>
<tr><td>

```
abnPol[trN + 1, aInd, ptN] <-
(1 -decayCoeffGroups) *abnPol[trN,
aInd, ptN] + decayCoeffGroups

abnPol[trN + 1, bInd, ptN] <-
(1 - decayCoeffGroups) *abnPol[trN,
bInd, ptN] + decayCoeffGroups

abnPol[trN + 1, nInd, ptN] <-
abnPol[trN + 1, aInd, ptN] +
abnPol[trN + 1, bInd, ptN]
```

</td><td>

On each trial, beliefs decay via the following equations, where λ_{acc} is the decay coefficient for beliefs about groups. The decay rate represents the assumption that participants have limited working memory and thus older observations will be replaced by newer ones.

$$\alpha'_{t} = (1-\lambda_{acc})\,\alpha_{t-1} + \lambda_{acc}$$
$$\beta'_{t} = (1-\lambda_{acc})\,\beta_{t-1} + \lambda_{acc}$$
$$n_{t} = \alpha'_{t} + \beta'_{t}$$

Equation (14)

</td></tr>
<tr><td>

```
abnPol[trN + 1, aInd[gpI], ptN]
<-abnPol[trN + 1, aInd[gpI], ptN] +
apprfb

abnPol[trN + 1, bInd[gpI], ptN] <-
abnPol[trN + 1, bInd[gpI], ptN] + 1 -
apprfb
```

</td><td>

After (14), i.e. decay of existing beliefs, occurs, beliefs are then updated, which will now be described in (16) below. Beliefs about a specific group then got updated if feedback, i.e. approval or disapproval, from that group is encountered (*e*), and remained unchanged if feedback was not encountered (*ne*).

$$\alpha^{(ne)}_{t} = \alpha^{(ne)'}_{t-1}$$
$$\beta^{(ne)}_{t} = \beta^{(ne)'}_{t-1}$$
$$\alpha^{(e)}_{t} = \alpha^{(e)'}_{t-1} + o_{t}$$
$$\beta^{(e)}_{t} = \beta^{(e)'}_{t-1} + 1 - o_{t},$$

</td></tr>
</table>

	where the outcome of a trial o_t = 1 on approval and o_t = 0 on disapproval Equation (16) (note that only the last two equations in this set are shown in the code here)

`ratingP <- 1/(1 + exp((1-2*(abnPol[trN, aInd[gpI], ptN] / abnPol[trN, nInd[gpI], ptN] + Bpred)) / Tpred))`
 	
`DatBelPol[trN+1,'genPred',ptN] <- rbinom(1,1,DatBelPol[trN+1,'predAccP',ptN]);`
	The probability of the participant predicting approval approval is π_L. *T* is a free parameter which represents decision temperature, the magnitude of difference between approval approval probability $G_{acc} = \frac{\alpha}{n}$ and the indifference point (0.5) needed to increase the probability of predicting approval approval by a certain amount. *B* is a free parameter which represents bias.
	
	$\pi_L = (1 + exp(-(G_{acc} + B)/T))^{-1}$
	
	Equation (17)
	
	The prediction the participant makes (approval prediction) is given by the binomial distribution $X \sim Bin(1, \pi_L)$.

`if (apprfb == 1) {`
`PE <-`
`abnPol[trN + 1, bInd[gpI], ptN] /`
`abnPol[trN + 1, nInd[gpI], ptN]`
`}`
 	
`else {`
`PE <- -abnPol[trN + 1, aInd[gpI], ptN] / abnPol[trN + 1, nInd[gpI], ptN]`
`}`
	Prediction errors for beliefs about groups are calculated upon encountering social approval approval or disapproval disapproval (**Figure 2b**). Positive prediction errors δ^+ occur on approval (o_t = 1), negative prediction errors δ^- occur on 'disapproval' (o_t = 0), and the group-specific expectations are taken from Equation (11):
	
	$\delta = o_t - \alpha^{(e)'} / n^{(e)'} \Rightarrow$
	$\delta^+ = \beta / n^{(e)'}$
	$\delta^- = -\alpha / n^{(e)'}$
	Equation (18)

`abnPol[trN + 1, SEaI, ptN] <- decayCoeffSelf* (abnPol[trN + 1, SEaI, ptN] - 1) + 1 + weightSelf * max(PE, 0)`
 	
`abnPol[trN + 1, SEbI, ptN] <- decayCoeffSelf* (abnPol[trN + 1, SEbI, ptN] - 1) + 1 - weightSelf * min(PE, 0)`
	Similar to beliefs about groups, beliefs about the self (*S*) are represented in a beta distribution with parameters α and β, and α + β = *n*. The prediction errors above are thus incorporated into the 'beta-belief' about the self via free parameters *w* and ζ, which weigh the prediction error and allow for trial-by-trial decay of beliefs respectively:

	$\alpha_t^{(S)} = (\alpha_{t-1}^{(S)} - 1)\varsigma + 1 + w\,max\,(\delta,\ 0)$ $\beta_t^{(S)} = (\beta_{t-1}^{(S)} - 1)\varsigma + 1 + w\,max\,(\delta,\ 0)$ Equation (12) (copied from Box 4 for clarity)

`accP2SE <- function(p, a, b){`
`  return( 1/(1+((1-p)/(p*b))^a) );`
`} # end accP2SE`
	Self-esteem ratings are generated as follows. First, a value is drawn randomly from this beta distribution; this value is termed P_{acc}. Then, a sigmoidal response function translates P_{acc} into self-esteem ratings via two free parameters, *m* and *B*, which represent the sensitivity of self-esteem changes and the participant's shift or bias respectively:
	
	$SE = \frac{1}{1+[(1-P_{acc})/(P_{acc}B)]^m}$
	
	Equation (19)

`abnPol[1, SEaI, ptN] <- mean(abnPol[1, aInd, ptN])`
`abnPol[1, SEbI, ptN] <- mean(abnPol[1, bInd, ptN])`
`abnPol[1, SEnI, ptN] <- mean(abnPol[1, nInd, ptN])`
 	
`n0   <- parMat[ptN,3]`
 	
`accP <- sum(abnPol[1, aInd, ptN]) / (n0*4)`
 	
`abnPol[1, 'expSE', ptN] <- accP2SE(accP, A, B)`
	Initial self-esteem levels are set by $P_{acc} = \frac{\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4}{n_1 + n_2 + n_3 + n_4}$ and applying the above equation (19).