Surveillance Endorsement in Multiple Cases

Maya Cratsley

## Prospectus

Question: When and why do people endorse surveillance?

Previous academic research has examined and identified several factors driving people’s willingness to be monitored, including: ignorance (not knowing the extent to which we are being watched; Hoofnagle et al. 2010), mutual benefit (getting something in return; PEW Research, 2014), perceived inevitability (Fast & Jago 2020), desire to divulge (mixed motives; John, 2016), lack of immediate negative outcomes (Fast & Jago 2020), and intangibility of privacy as a concept (John, 2016).

However, most research thus far on people’s willingness to be monitored has not made clear distinctions between surveillance and other passive data collection, and may have been intended to apply specifically to the latter. While all data collection represents a reduction in privacy, some can be seen as serving a purely informational function, while other data collection also has a control function. We thus define surveillance as: “observation of an individual by an institution or individual who holds power over them, for the purpose of influence, management, and control.” This definition is important because it highlights a key feature of surveillance technologies: their potential to infringe on not only people’s privacy, but also people’s autonomy.

The potential for surveillance to infringe on autonomy is important because, theoretically speaking,it means that people should oppose surveillance that applies to themselves. Basic Psychological Needs Theory, a component of Self-Determination Theory, posits that autonomy is one of the three fundamental psychological desires of human beings, meaning that infringements on autonomy, such as surveillance, would be avoided if at all possible (Deci & Ryan, 1985; Deci & Ryan 2000, Ryan & Deci 2000). However, in real life we have seen a steady increase in the degree to which we allow both private and public institutions to surveil us. This is the puzzle that this research adresses. We ultimately argue that people will endorse their own surveillance, and will even make sacrifices in order to be surveilled, when social contextual factors allow them to perceive the surveillance as not presenting any threat to their autonomy.

Individuals typically have access to social-relational information that they use to infer the impact a given initiative will have on themselves, and whether it may result in a decrease to their autonomy.Specifically, the perceived threat to autonomy that is posed by surveillance may be impacted by individuals’ relationship to the organization or person doing the surveillance. The degree to which we see the person or group surveilling us as aligned with ourselves (perceived self-other alignment) informs our response to surveillance initiatives, by encouraging individuals to believe that the surveiller will likely take little issue with their current behaviors. This leads people to believe that they will not have to change their behaviors, and thus give up their autonomy, in order to avoid punishment. For example, if you are a member of the LGBTQ+ community, and you see your government as aligned with you, you may not be concerned about needing to avoid being seen with your partner in order to avoid governmental sanctions. Thus, governmental surveillance may be seen as not infringing upon your autonomy. This study aims to test the claim that the degree to which we see ourselves as aligned with a given institution will predict our willingness to be surveilled by that institution.

Design:

The study was hosted on Qualtrics, and distributed through the USC SONA system. Participants who consented to participate first read some information about “new technological tools” for tracking their internet activity. They were then asked to rate for 10 different groups (the police, google, the CDC, etc): the degree to which they see themselves as aligned with those groups (SOA- IV, mod. from Aron, Aron, & Smollan, 1992) and how much they would endorse the use of these tools by those groups (endorsement- DV). The order in which the SOA and endorsement measures will be presented was randomized, such that half of the participants saw the IV first and half the DV. Additionally, for each measure (SOA and Endorsement), the groups they were asked to evaluate were presented in a random order.

The data has a cross-classified structure with level 2a representing clustering by participant and level 2b representing clustering by cases. The cluster size for participant= 10, and the cluster size for cases=~200.

Primary analyses (preregistered):

We will use the lme4 package (Bates, Maechler & Bolker, 2012) in R to fit a linear mixed model using the restricted maximum likelihood estimation method. The only fixed effect in the model will be that of SOA on Endorsement. Participant id and target case (the police, Amazon, etc.) will be included as random intercepts to account for across-participant variations in overall endorsement of surveillance, and variation in endorsement across the various cases. Prior to running hypothesis-testing analyses, we will examine the residuals plot of the data to ensure that the data does not demonstrate any obvious deviations from homoskedasticity and normality. Provided the data doesn’t violate assumptions of our analyses and we can proceed as planned, we will then perform a likelihood ratio test to compare the full model with the single fixed effect and two random effects to a null model that is identical but excludes the fixed effect of SOA on Endorsement. If the likelihood ratio test yields a p value <0.05, this will be interpreted as indicating statistical significance of the overall effect of SOA on Endorsement, and support for our hypothesis.

Additional analyses (not pre-registered):

As an exploratory measure, we will estimate two additional models, one that includes a random slopes for participant id, and one that includes a random slope for cases. Each of these models will be tested to see whether the random slopes are significant. If either/both of the random slopes are significant, we will estimate a model that includes the significant slopes, and perform likelihood ratio tests comparing that model to the null model, as well as to the pre-registered hypothesized model. We will also calculate and report ICCs for both participant id and cases, to demonstrate that the variance across cases and across participants was non-negligible.

## Load Packages

#Loading Mark's function for marginal model plots  
mmps\_lmer <- function(object) {  
 plot\_df <- object@frame  
 form <- formula(object)  
 xvar <- attr(attr(plot\_df, "terms"), "varnames.fixed")[-1]  
 plot\_df$.fitted\_x <- fitted(object)  
 plot\_df$.fitted <- plot\_df$.fitted\_x  
 plot\_df$.rowid <- seq\_len(nrow(plot\_df))  
 plot\_df\_long <- reshape(plot\_df, direction = "long",  
 varying = c(xvar, ".fitted\_x"),  
 v.names = "xvar",  
 idvar = ".rowid")  
 plot\_df\_long$varname <- rep(c(xvar, ".fitted"),  
 each = nrow(plot\_df))  
 ggplot(  
 data = plot\_df\_long,  
 aes\_string(x = "xvar", y = paste(form[[2]]))  
 ) +  
 geom\_point(size = 0.5, alpha = 0.3) +  
 geom\_smooth(aes(col = "data"), se = FALSE) +  
 geom\_smooth(aes(y = .fitted, col = "model"),  
 linetype = "dashed", se = FALSE  
 ) +  
 facet\_wrap(~ varname, scales = "free\_x") +  
 labs(color = NULL, x = NULL) +  
 scale\_color\_manual(values = c("data" = "blue",  
 "model" = "red")) +  
 theme(legend.position = "bottom")  
}

## Load data

Convert to long format

surveillancedt <- read.csv("~/Desktop/Surveillance/Endorsement/MultCase1.csv")  
#load the data into a dataframe  
  
# Convert to long format using the new `tidyr::pivot\_longer()` function  
surveillancedt\_long <- surveillancedt %>%  
 pivot\_longer(  
 c(SOA\_1:SOA\_10,Endorsement\_1:Endorsement\_10), # variables that are repeated measures  
 # Convert 20 columns to 3: 1 columns each for SOA/Endorsement (.value), and  
 # one column for case  
 names\_to = c(".value", "Case"),  
 # Extract the names "SOA"/"Endorsement" from the names of the variables for the  
 # value columns, and then the number to the "Case" column  
 names\_pattern = "(SOA|Endorsement)\_(.\*)",  
 # Convert the "Case" column to integers- NOTE that the .\* indicated to just use everything after the \_  
 #this works in this case because the only thing after the \_ are the numbers 1-10, but wont always work  
 names\_transform = list(Case = as.integer)  
 )  
surveillancedt\_long %>%   
 select(ResponseId, SOA, Endorsement, Case, everything())

## # A tibble: 2,660 x 40  
## ResponseId SOA Endorsement Case StartDate EndDate Status IPAddress   
## <chr> <int> <int> <int> <chr> <chr> <int> <chr>   
## 1 R\_1hQIEbHWrd… 4 2 1 9/14/21 13… 9/14/21 … 0 207.151.5…  
## 2 R\_1hQIEbHWrd… 6 3 2 9/14/21 13… 9/14/21 … 0 207.151.5…  
## 3 R\_1hQIEbHWrd… 2 1 3 9/14/21 13… 9/14/21 … 0 207.151.5…  
## 4 R\_1hQIEbHWrd… 2 1 4 9/14/21 13… 9/14/21 … 0 207.151.5…  
## 5 R\_1hQIEbHWrd… 2 1 5 9/14/21 13… 9/14/21 … 0 207.151.5…  
## 6 R\_1hQIEbHWrd… 1 1 6 9/14/21 13… 9/14/21 … 0 207.151.5…  
## 7 R\_1hQIEbHWrd… 5 2 7 9/14/21 13… 9/14/21 … 0 207.151.5…  
## 8 R\_1hQIEbHWrd… 2 1 8 9/14/21 13… 9/14/21 … 0 207.151.5…  
## 9 R\_1hQIEbHWrd… 5 1 9 9/14/21 13… 9/14/21 … 0 207.151.5…  
## 10 R\_1hQIEbHWrd… 5 1 10 9/14/21 13… 9/14/21 … 0 207.151.5…  
## # … with 2,650 more rows, and 32 more variables: Progress <int>,  
## # Duration..in.seconds. <int>, Finished <int>, RecordedDate <chr>,  
## # RecipientLastName <lgl>, RecipientFirstName <lgl>, RecipientEmail <lgl>,  
## # ExternalReference <lgl>, LocationLatitude <dbl>, LocationLongitude <dbl>,  
## # DistributionChannel <chr>, UserLanguage <chr>, Consent <int>,  
## # PassCheck <int>, Privacy\_1 <int>, Privacy\_2 <int>, Privacy\_3 <int>,  
## # Privacy\_4 <int>, Gender <int>, Gender\_other <chr>, Age <int>,  
## # Education <int>, Race <int>, Employment <int>, Employment\_other <chr>,  
## # Ladder <int>, Poli\_affil <int>, Poli\_party <int>, Poli\_other <chr>,  
## # id <int>, Order <int>, Exclude <int>

### Equations

Pre-registered Model

Repeated-Measure level (Lv 1):

Between-cell (Subject Item) level (Lv 2):

In this equation, represents the deviation of the intercept of participant from the average intercept (i.e. ), and represents the deviation of the intercept of case from the average intercept.

Pre-registered Model Equation:

Final Model:

Repeated-Measure level (Lv 1):

Between-cell (Subject Item) level (Lv 2):

We’ve now introduced two new terms, with, represents the deviation of the slope of participant from the average slope (i.e., ), and represents the deviation of the slope of case from the average slope. Final Model Equation:

## Building the Model

I’ll estimate the preregistered model (equation above).

surveillance.model <- lmer( Endorsement ~ SOA + (1 | ResponseId)+ (1 | Case), data=surveillancedt\_clean)  
#create and view our pre-registered mixed effects model with 1 fixed effect and 2 random intercepts  
  
summary(surveillance.model)

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [  
## lmerModLmerTest]  
## Formula: Endorsement ~ SOA + (1 | ResponseId) + (1 | Case)  
## Data: surveillancedt\_clean  
##   
## REML criterion at convergence: 9059.1  
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -3.2908 -0.6235 -0.0685 0.5896 3.7471   
##   
## Random effects:  
## Groups Name Variance Std.Dev.  
## ResponseId (Intercept) 1.1750 1.0840   
## Case (Intercept) 0.1449 0.3806   
## Residual 1.4406 1.2003   
## Number of obs: 2634, groups: ResponseId, 266; Case, 10  
##   
## Fixed effects:  
## Estimate Std. Error df t value Pr(>|t|)   
## (Intercept) 2.1136 0.1499 19.9457 14.10 7.79e-12 \*\*\*  
## SOA 0.2976 0.0165 2552.3726 18.04 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Correlation of Fixed Effects:  
## (Intr)  
## SOA -0.365

#view the summary of our model to help with interpretation  
#intercept estimate tells you the anticipated endorsement level at alignment=0  
#alignment values ~i think~ tells you the expected change in endorsement per point of aligment change  
confint(surveillance.model, parm = "beta\_")

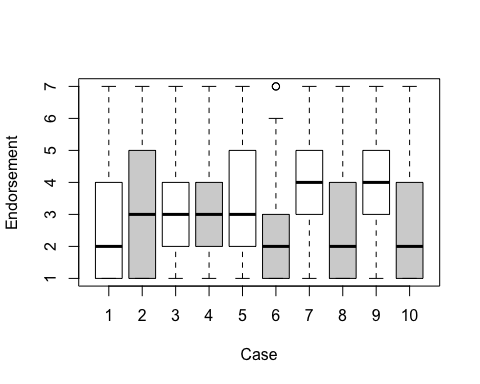
## Computing profile confidence intervals ...

## 2.5 % 97.5 %  
## (Intercept) 1.8144171 2.413881  
## SOA 0.2653464 0.330049

#confidence interval does not cross 0, so null is rejected ~but by how much~?

## Visualizing the Data

boxplot(Endorsement ~ Case,  
 col=c("white","lightgray"),surveillancedt\_clean)



#look at the variation in mean Endorsement between cases

On this chart, the labels are as follows:

1- Your place of employment

2-the police

3- apple

4-amazon

5-google

6-facebook

7-The National Science Foundation

8- The US Military

9- The CDC

10- The US government

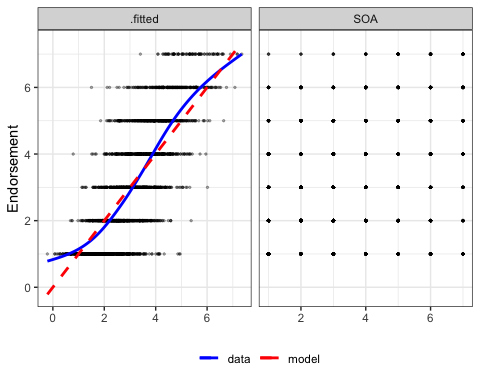
#testing for linearity by making a residuals plot using Mark's function  
mmps\_lmer(surveillance.model)

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

## Warning: Computation failed in `stat\_smooth()`:  
## x has insufficient unique values to support 10 knots: reduce k.

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

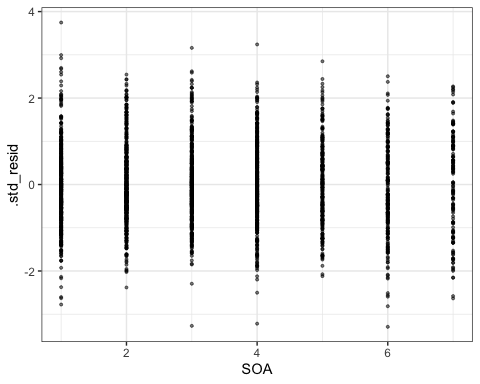
## Warning: Computation failed in `stat\_smooth()`:  
## x has insufficient unique values to support 10 knots: reduce k.



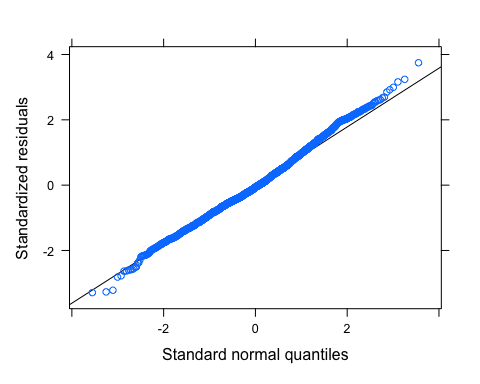
#Testing for homoscedasticity across Alignment Values (level 1)  
augment(surveillance.model) %>%  
 mutate(.std\_resid = resid(surveillance.model, scaled = TRUE)) %>%  
 ggplot(aes(x = SOA, y = .std\_resid)) +  
 geom\_point(size = 0.7, alpha = 0.5) +  
 geom\_smooth(se = FALSE)

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

## Warning: Computation failed in `stat\_smooth()`:  
## x has insufficient unique values to support 10 knots: reduce k.

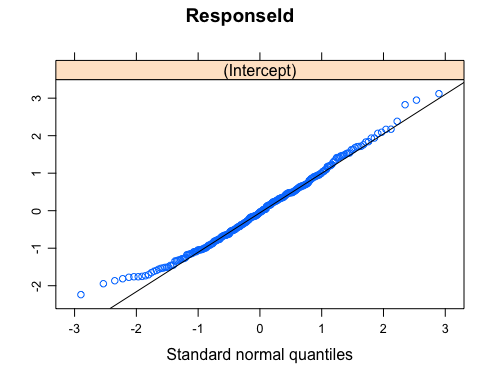


#Testing for normality at level 1  
library(lattice) # need this package to use the built-in functions  
qqmath(surveillance.model) # just use the `qqmath()` function on the fitted model

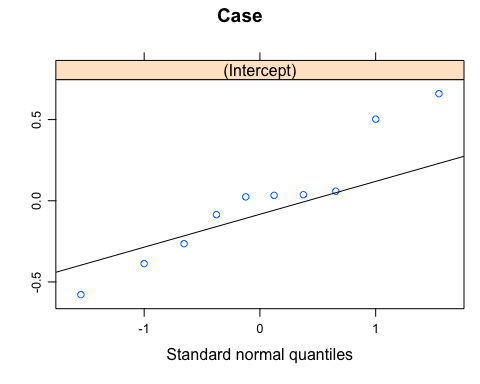


#Testing for normality at other levels  
qqmath(ranef(surveillance.model, condVar = FALSE),  
 panel = function(x) {  
 panel.qqmath(x)  
 panel.qqmathline(x)  
 })

## $ResponseId



##   
## $Case



#plot(fitted(surveillance.model),residuals(surveillance.model))  
#another plot I could use to check for linearity, normality & homoskedasticity

The test for linearity shows some minor deviations from linearity. Homoscedasticity is demonstrated, confirming the equal variance assumption. The normality assumption is also confirmed at the data and participant level, however there is some variance across cases.

## Model Estimates

#calculate ICC for Endorsement based on our two intercepts  
  
  
surveillance.null <- lmer(Endorsement ~ (1 | ResponseId) + (1 | Case), data=surveillancedt\_clean)  
#create our null model, which excludes the fixed effect of alignment  
  
vc\_null <- as.data.frame(VarCorr(surveillance.null))  
# Proportion of variance at the within-cell level  
icc\_e <- vc\_null$vcov[3] / sum(vc\_null$vcov)  
  
# ICC/Deff (person; cluster size= 266  
icc\_person <- vc\_null$vcov[1] / sum(vc\_null$vcov)  
c("ICC(person)" = icc\_person,  
"Deff(person)" = icc\_e + 266 \* icc\_person)

## ICC(person) Deff(person)   
## 0.4137133 110.5658892

# ICC/Deff (Case; cluster size = 10)  
icc\_case <- vc\_null$vcov[2] / sum(vc\_null$vcov)  
c("ICC(case)" = icc\_case,  
"Deff(case)" = icc\_e + 10 \* icc\_case)

## ICC(case) Deff(case)   
## 0.06814422 1.19958465

The ICC for “the variance between”ResponseId" is 0.4137, indicating that 41% of the variance in endorsement is explained by between-person differences. The ICC for “Case” is 0.0681, indicating that 6% of the variance in endorsement is explained by differences between cases. Thus, it seems that it was appropriate to include these two random intercepts into the model.

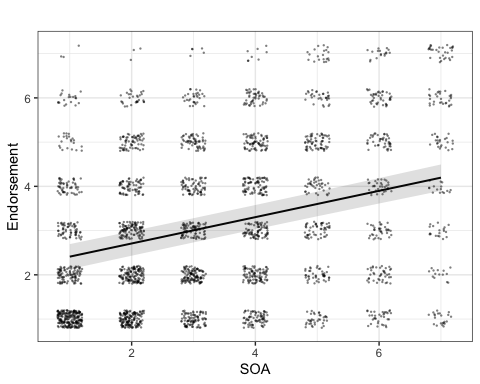
msummary(list(Estimate = surveillance.model, `95% CI` = surveillance.model),  
estimate = c("estimate", "[{conf.low}, {conf.high}]"),  
statistic = NULL, # suppress the extra rows for SEs  
title = "Parameter Estimates of Multilevel Model",  
gof\_omit = ".\*")

## Warning in msummary(list(Estimate = surveillance.model, `95% CI` =  
## surveillance.model), : The table includes duplicate term names. This can happen  
## when `coef\_map` or `coef\_rename` are misused. This can also happen when a  
## model produces "grouped" terms, such as in multinomial logit or gamlss models.  
## You may want to call `get\_estimates(model)` to see how estimates are labelled  
## internally, and use the `group` argument of the `modelsummary` function.

Table :Parameter Estimates of Multilevel Model

|  | Estimate | 95% CI |
| --- | --- | --- |
| (Intercept) | 2.114 | [1.801, 2.426] |
| SOA | 0.298 | [0.265, 0.330] |
| sd\_\_(Intercept) | 0.381 |  |
|  | 0.381 |  |
|  | 1.084 |  |
|  | 1.084 |  |
| sd\_\_Observation | 1.200 |  |

#plotting the model predictions next to the data  
sjPlot::plot\_model(surveillance.model, type = "pred", terms = "SOA",   
 show.data = TRUE, title = "",   
 dot.size = 0.5, jitter =0.2)



## Alternative Models

I will also test random slopes for Case and ResponseId to see whether they are worth including in the model.

surveillance.model2 <- lmer(Endorsement ~ SOA + (SOA | ResponseId)+ (1 | Case), data=surveillancedt\_clean)  
#create a model that includes random slopes for participants  
ranova(surveillance.model2)

## ANOVA-like table for random-effects: Single term deletions  
##   
## Model:  
## Endorsement ~ SOA + (SOA | ResponseId) + (1 | Case)  
## npar logLik AIC LRT Df Pr(>Chisq)   
## <none> 7 -4474.4 8962.7   
## SOA in (SOA | ResponseId) 5 -4529.6 9069.1 110.40 2 < 2.2e-16 \*\*\*  
## (1 | Case) 6 -4576.1 9164.1 203.37 1 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#test whether the random slope is significant  
  
surveillance.model3 <- lmer(Endorsement ~ SOA + (1 | ResponseId)+ (SOA | Case), data=surveillancedt\_clean)  
#create a model that includes random slopes for case  
ranova(surveillance.model3)

## ANOVA-like table for random-effects: Single term deletions  
##   
## Model:  
## Endorsement ~ SOA + (1 | ResponseId) + (SOA | Case)  
## npar logLik AIC LRT Df Pr(>Chisq)   
## <none> 7 -4525.7 9065.4   
## (1 | ResponseId) 6 -5019.4 10050.8 987.38 1 < 2e-16 \*\*\*  
## SOA in (SOA | Case) 5 -4529.6 9069.1 7.76 2 0.02061 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#test whether the random slope is significant

Both random slopes were significant, so I will build a model with both included.

surveillance.model4 <- lmer(Endorsement ~ SOA + (SOA | ResponseId)+ (SOA | Case), data=surveillancedt\_clean)  
  
summary(surveillance.model4)

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [  
## lmerModLmerTest]  
## Formula: Endorsement ~ SOA + (SOA | ResponseId) + (SOA | Case)  
## Data: surveillancedt\_clean  
##   
## REML criterion at convergence: 8937.3  
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -3.3522 -0.5885 -0.0731 0.5535 3.4847   
##   
## Random effects:  
## Groups Name Variance Std.Dev. Corr   
## ResponseId (Intercept) 1.246308 1.11638   
## SOA 0.058227 0.24130 -0.40  
## Case (Intercept) 0.140430 0.37474   
## SOA 0.004704 0.06858 -0.35  
## Residual 1.271466 1.12759   
## Number of obs: 2634, groups: ResponseId, 266; Case, 10  
##   
## Fixed effects:  
## Estimate Std. Error df t value Pr(>|t|)   
## (Intercept) 2.09714 0.14919 16.04865 14.057 1.93e-10 \*\*\*  
## SOA 0.30412 0.03136 19.34052 9.697 7.29e-09 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Correlation of Fixed Effects:  
## (Intr)  
## SOA -0.478

#creating and viewing a model with random slopes  
  
  
confint(surveillance.model4, parm = "beta\_")

## Computing profile confidence intervals ...

## 2.5 % 97.5 %  
## (Intercept) 1.7960131 2.3983012  
## SOA 0.2411932 0.3673834

## Likelihood Ratio Test

surveillance.null

## Linear mixed model fit by REML ['lmerModLmerTest']  
## Formula: Endorsement ~ (1 | ResponseId) + (1 | Case)  
## Data: surveillancedt\_clean  
## REML criterion at convergence: 9359.4  
## Random effects:  
## Groups Name Std.Dev.  
## ResponseId (Intercept) 1.1372   
## Case (Intercept) 0.4615   
## Residual 1.2726   
## Number of obs: 2634, groups: ResponseId, 266; Case, 10  
## Fixed Effects:  
## (Intercept)   
## 3.101

#view our null model, which excludes the fixed effect of alignment  
  
anova(surveillance.null, surveillance.model)

## refitting model(s) with ML (instead of REML)

## Data: surveillancedt\_clean  
## Models:  
## surveillance.null: Endorsement ~ (1 | ResponseId) + (1 | Case)  
## surveillance.model: Endorsement ~ SOA + (1 | ResponseId) + (1 | Case)  
## npar AIC BIC logLik deviance Chisq Df Pr(>Chisq)   
## surveillance.null 4 9365.6 9389.1 -4678.8 9357.6   
## surveillance.model 5 9060.6 9090.0 -4525.3 9050.6 306.94 1 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#using the anova function to perform the Likelihood Ratio Test which will compare our model to the null  
# The hypothesized model is a significant improvement on the null  
#reporting: Alignment significantly increased endorsement (chisq("DF")="chisq", p="PR(>Chisq)").   
  
anova(surveillance.model,surveillance.model4)

## refitting model(s) with ML (instead of REML)

## Data: surveillancedt\_clean  
## Models:  
## surveillance.model: Endorsement ~ SOA + (1 | ResponseId) + (1 | Case)  
## surveillance.model4: Endorsement ~ SOA + (SOA | ResponseId) + (SOA | Case)  
## npar AIC BIC logLik deviance Chisq Df Pr(>Chisq)  
## surveillance.model 5 9060.6 9090.0 -4525.3 9050.6   
## surveillance.model4 9 8947.9 9000.8 -4465.0 8929.9 120.72 4 < 2.2e-16  
##   
## surveillance.model   
## surveillance.model4 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#the model with random slopes included for both Case and Participant is a significant improvement on the hypothesized model  
  
msummary(list(surveillance.model, surveillance.model4))

## Warning in msummary(list(surveillance.model, surveillance.model4)): The  
## table includes duplicate term names. This can happen when `coef\_map` or  
## `coef\_rename` are misused. This can also happen when a model produces "grouped"  
## terms, such as in multinomial logit or gamlss models. You may want to call  
## `get\_estimates(model)` to see how estimates are labelled internally, and use the  
## `group` argument of the `modelsummary` function.

|  | Model 1 | Model 2 |
| --- | --- | --- |
| (Intercept) | 2.114 | 2.097 |
|  | (0.150) | (0.149) |
| SOA | 0.298 | 0.304 |
|  | (0.016) | (0.031) |
| sd\_\_(Intercept) | 0.381 | 0.375 |
|  | 0.381 | 1.116 |
|  | 1.084 | 0.375 |
|  | 1.084 | 1.116 |
| sd\_\_Observation | 1.200 | 1.128 |
| cor\_\_(Intercept).SOA |  | -0.354 |
|  |  | -0.404 |
| sd\_\_SOA |  | 0.069 |
|  |  | 0.241 |
| AIC | 9069.1 | 8955.3 |
| BIC | 9098.5 | 9008.2 |
| Log.Lik. | -4529.574 | -4468.645 |
| REMLcrit | 9059.149 | 8937.290 |

The likelihood ration tests demonstrate that the hypothesized model is a significant improvement on the null model, but the alternative model, which includes random slopes, is a significant improvement on the hypothesized model.These results largely indicate that Alignment was a significant predictor of Endorsement, such that a 1 point increase in Alignment predicted about a 0.3 point increase in endorsement. This provides support for the theoretical perspective being tested.

r.squaredGLMM(surveillance.model) #returns the marginal and conditional Rsquared

## Warning: 'r.squaredGLMM' now calculates a revised statistic. See the help page.

## R2m R2c  
## [1,] 0.09311078 0.5267319

boo\_r2 <- bootstrap\_mer(surveillance.model, MuMIn::r.squaredGLMM, nsim = 999,  
type = "residual")

## 1 warning(s) in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, lbound = lower) : Model failed to converge with max|grad| = 0.00202435 (tol = 0.002, component 1)  
## 1 warning(s) in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, lbound = lower) : Model failed to converge with max|grad| = 0.00241255 (tol = 0.002, component 1)  
## 1 warning(s) in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, lbound = lower) : Model failed to converge with max|grad| = 0.00243979 (tol = 0.002, component 1)  
## 1 warning(s) in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, lbound = lower) : Model failed to converge with max|grad| = 0.00378375 (tol = 0.002, component 1)  
## 1 warning(s) in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, lbound = lower) : Model failed to converge with max|grad| = 0.00537729 (tol = 0.002, component 1)

boot.ci(boo\_r2, index = 1, type = "perc")

## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS  
## Based on 999 bootstrap replicates  
##   
## CALL :   
## boot.ci(boot.out = boo\_r2, type = "perc", index = 1)  
##   
## Intervals :   
## Level Percentile   
## 95% ( 0.0735, 0.1143 )   
## Calculations and Intervals on Original Scale