Week 9 Workbook

Odelia

5/5/2021

library

library("tidyverse")

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.0 ──

## ✓ ggplot2 3.3.3 ✓ purrr 0.3.4  
## ✓ tibble 3.1.1 ✓ dplyr 1.0.5  
## ✓ tidyr 1.1.2 ✓ stringr 1.4.0  
## ✓ readr 1.4.0 ✓ forcats 0.5.1

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library("patchwork")  
library("lubridate")

##   
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':  
##   
## date, intersect, setdiff, union

library("kableExtra")

##   
## Attaching package: 'kableExtra'

## The following object is masked from 'package:dplyr':  
##   
## group\_rows

library("gtsummary")  
library("lubridate")  
library("equatiomatic")  
library("ggdag")

##   
## Attaching package: 'ggdag'

## The following object is masked from 'package:stats':  
##   
## filter

library("brms")

## Loading required package: Rcpp

## Loading 'brms' package (version 2.15.0). Useful instructions  
## can be found by typing help('brms'). A more detailed introduction  
## to the package is available through vignette('brms\_overview').

##   
## Attaching package: 'brms'

## The following object is masked from 'package:stats':  
##   
## ar

library("rstan")

## Loading required package: StanHeaders

## rstan (Version 2.26.1, GitRev: 2e1f913d3ca3)

## For execution on a local, multicore CPU with excess RAM we recommend calling  
## options(mc.cores = parallel::detectCores()).  
## To avoid recompilation of unchanged Stan programs, we recommend calling  
## rstan\_options(auto\_write = TRUE)  
## For within-chain threading using `reduce\_sum()` or `map\_rect()` Stan functions,  
## change `threads\_per\_chain` option:  
## rstan\_options(threads\_per\_chain = 1)

##   
## Attaching package: 'rstan'

## The following object is masked from 'package:tidyr':  
##   
## extract

library("rstanarm")

## This is rstanarm version 2.21.1

## - See https://mc-stan.org/rstanarm/articles/priors for changes to default priors!

## - Default priors may change, so it's safest to specify priors, even if equivalent to the defaults.

## - For execution on a local, multicore CPU with excess RAM we recommend calling

## options(mc.cores = parallel::detectCores())

##   
## Attaching package: 'rstanarm'

## The following object is masked from 'package:rstan':  
##   
## loo

## The following objects are masked from 'package:brms':  
##   
## dirichlet, exponential, get\_y, lasso, ngrps

library("bayesplot")

## This is bayesplot version 1.8.0

## - Online documentation and vignettes at mc-stan.org/bayesplot

## - bayesplot theme set to bayesplot::theme\_default()

## \* Does \_not\_ affect other ggplot2 plots

## \* See ?bayesplot\_theme\_set for details on theme setting

library("easystats")

## # Attaching packages (red = needs update)  
## ⚠ insight 0.13.1.1 ⚠ bayestestR 0.8.3.1   
## ⚠ performance 0.7.0.1 ⚠ parameters 0.12.0.1  
## ⚠ see 0.6.2.1 ✔ effectsize 0.4.4.1   
## ⚠ correlation 0.6.0.1 ⚠ modelbased 0.5.9   
## ⚠ report 0.2.0   
## Warnings or errors in CRAN checks for package(s) 'bayestestR', 'modelbased', 'correlation'.  
## Restart the R-Session and update packages in red with 'easystats::easystats\_update()'.

library("kableExtra")  
library("broom")  
library("tidybayes")

##   
## Attaching package: 'tidybayes'

## The following object is masked from 'package:parameters':  
##   
## parameters

## The following object is masked from 'package:bayestestR':  
##   
## hdi

## The following objects are masked from 'package:brms':  
##   
## dstudent\_t, pstudent\_t, qstudent\_t, rstudent\_t

library("bmlm")  
# rstan options  
rstan\_options(auto\_write = TRUE)  
options(mc.cores = parallel::detectCores ())  
theme\_set(theme\_classic())

read data

nz\_0 <- as.data.frame(readr::read\_csv2(  
 url(  
 "https://raw.githubusercontent.com/go-bayes/psych-447/main/data/nzj.csv"  
 )  
))

## ℹ Using ',' as decimal and '.' as grouping mark. Use `read\_delim()` for more control.

##   
## ── Column specification ────────────────────────────────────────────────────────  
## cols(  
## .default = col\_double(),  
## Male = col\_character(),  
## BigDoms = col\_character(),  
## GenCohort = col\_character(),  
## Religious\_Group = col\_character(),  
## Religious = col\_character(),  
## Believe.God = col\_character(),  
## Believe.Spirit = col\_character(),  
## Env.SacMade = col\_logical(),  
## FeelHopeless = col\_character(),  
## FeelDepressed = col\_character(),  
## FeelRestless = col\_character(),  
## EverythingIsEffort = col\_character(),  
## FeelWorthless = col\_character(),  
## FeelNervous = col\_character()  
## )  
## ℹ Use `spec()` for the full column specifications.

# to relevel kessler 6 variables  
f <-  
 c(  
 "None Of The Time",  
 "A Little Of The Time",  
 "Some Of The Time",  
 "Most Of The Time",  
 "All Of The Time"  
 )  
nz\_cr <- nz\_0 %>%  
 dplyr::mutate\_if(is.character, factor) %>%  
 select(  
 -c(  
 SWB.Kessler01,  
 SWB.Kessler02,  
 SWB.Kessler03,  
 SWB.Kessler04,  
 SWB.Kessler05,  
 SWB.Kessler06  
 )  
 ) %>%  
 dplyr::mutate(Wave = as.factor(Wave)) %>%  
 dplyr::mutate(FeelHopeless = forcats::fct\_relevel(FeelHopeless, f)) %>%  
 dplyr::mutate(FeelDepressed = forcats::fct\_relevel(FeelDepressed, f)) %>%  
 dplyr::mutate(FeelRestless = forcats::fct\_relevel(FeelRestless, f)) %>%  
 dplyr::mutate(EverythingIsEffort = forcats::fct\_relevel(EverythingIsEffort, f)) %>%  
 dplyr::mutate(FeelWorthless = forcats::fct\_relevel(FeelWorthless, f)) %>%  
 dplyr::mutate(FeelNervous = forcats::fct\_relevel(FeelNervous, f)) %>%  
 dplyr::mutate(Wave = as.factor(Wave)) %>%  
 dplyr::mutate(male\_id = as.factor(Male)) %>%  
 dplyr::mutate(date = make\_date(year = 2009, month = 6, day = 30) + TSCORE) %>%  
 dplyr::mutate(  
 FeelWorthless\_int = as.integer(FeelWorthless),  
 FeelNervous\_int = as.integer(FeelNervous),  
 FeelHopeless\_int = as.integer(FeelHopeless),  
 EverythingIsEffort\_int = as.integer(EverythingIsEffort),  
 FeelRestless\_int = as.integer(FeelRestless),  
 FeelDepressed\_int = as.integer(FeelDepressed),  
 HLTH.Fatigue\_int = as.integer(HLTH.Fatigue + 1)  
 ) %>%  
 dplyr::mutate(yearS = TSCORE - min(TSCORE, na.rm = TRUE)) %>%  
 dplyr::mutate(KESSLER6sum = as.integer(KESSLER6sum))  
nz2018 <- nz\_cr %>%  
 dplyr::filter(Wave == 2018)  
ord\_dates\_class\_2019\_only <- c("PreCOVID",  
 "JanFeb",  
 "EarlyMarch",  
 "Lockdown",  
 "PostLockdown")  
  
nz2019 <- nz\_cr %>%  
 dplyr::filter(YearMeasured == 1) %>%  
 dplyr::filter(Wave == 2019) %>%  
 dplyr::group\_by(Id) %>%  
 dplyr::ungroup(Id) %>%  
 dplyr::mutate(Covid\_Timeline\_cr =  
 as.factor(ifelse(  
 TSCORE %in% 3896:3921,  
 # feb 29 - march 25th  
 "EarlyMarch",  
 ifelse(  
 TSCORE %in% 3922:3954,  
 "Lockdown",  
 #march 26- Mon 27 April 2020  
 ifelse(  
 TSCORE > 3954,  
 # after april 27th 20202  
 "PostLockdown",  
 ifelse(TSCORE %in% 3842:3895,  
 # jan 6 to feb 28  
 "JanFeb",  
 "PreCOVID")  
 )  
 )  
 ))) %>%  
 dplyr::mutate(Covid\_Timeline\_cr = forcats::fct\_relevel(Covid\_Timeline\_cr, ord\_dates\_class\_2019\_only))  
ord\_dates\_class <- c("Baseline",  
 "PreCOVID",  
 "JanFeb",  
 "EarlyMarch",  
 "Lockdown",  
 "PostLockdown")  
nzl <- nz\_cr %>%  
 dplyr::filter(YearMeasured == 1) %>%  
 dplyr::filter(Wave == 2018 | Wave == 2019) %>%  
 dplyr::group\_by(Id) %>%  
 dplyr::filter(n() > 1) %>%  
 dplyr::filter(n() != 0) %>%  
 dplyr::ungroup(Id) %>%  
 dplyr::mutate(yearS = (TSCORE - min(TSCORE)/365)) %>%  
 dplyr::mutate(WSCORE = as.factor(WSCORE)) %>%  
 dplyr::mutate(Covid\_Timeline =  
 as.factor(ifelse(  
 TSCORE %in% 3896:3921,  
 # feb 29 - march 25th  
 "EarlyMarch",  
 ifelse(  
 TSCORE %in% 3922:3954,  
 "Lockdown",  
 #march 26- Mon 27 April 2020  
 ifelse(  
 TSCORE > 3954,  
 # after april 27th 20202  
 "PostLockdown",  
 ifelse(  
 TSCORE %in% 3842:3895,  
 # jan 6 to feb 28  
 "JanFeb",  
 ifelse(TSCORE %in% 3665:3841 &  
 Wave == 2019,  
 "PreCOVID",  
 "Baseline" # 3672 TSCORE or 20 July 2019))))))))  
 )  
 )  
 )  
 ))))%>%  
 dplyr::mutate(Covid\_Timeline = forcats::fct\_relevel(Covid\_Timeline, ord\_dates\_class))

**Write brief report that in which you estimating demographic and ideological predictors of attitudes to government surveillance (Issue.GovtSurveillance) and attitudes to (Issue.RegulateAI) in New Zealand in 2018**

To center:

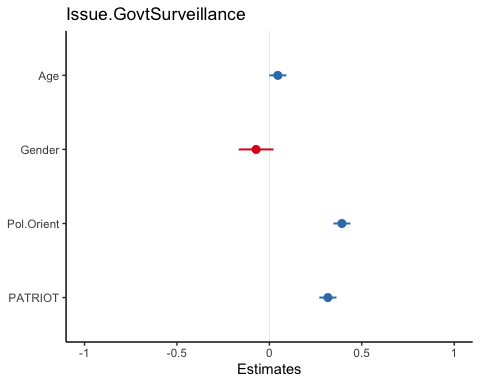
nz2018\_c <- nz2018   
nz2018\_c['Age'] = as.data.frame(scale(nz2018$Age), scale = FALSE)  
nz2018\_c['Pol.Orient'] = as.data.frame(scale(nz2018$Pol.Orient), scale = FALSE)  
nz2018\_c['PATRIOT'] = as.data.frame(scale(nz2018$PATRIOT), scale = FALSE)

Demographic and ideological predictors of attitudes towards government surveillance

plot1 <- lm(Issue.GovtSurveillance ~ Age + Gender + Pol.Orient + PATRIOT, data = nz2018\_c)  
parameters::parameters(plot1)

## Parameter | Coefficient | SE | 95% CI | t(5530) | p  
## -------------------------------------------------------------------  
## (Intercept) | 4.41 | 0.03 | [ 4.36, 4.47] | 151.58 | < .001  
## Age | 0.05 | 0.02 | [ 0.00, 0.09] | 1.94 | 0.052   
## Gender | -0.07 | 0.05 | [-0.17, 0.02] | -1.50 | 0.134   
## Pol.Orient | 0.39 | 0.02 | [ 0.35, 0.44] | 16.80 | < .001  
## PATRIOT | 0.32 | 0.02 | [ 0.27, 0.36] | 13.38 | < .001

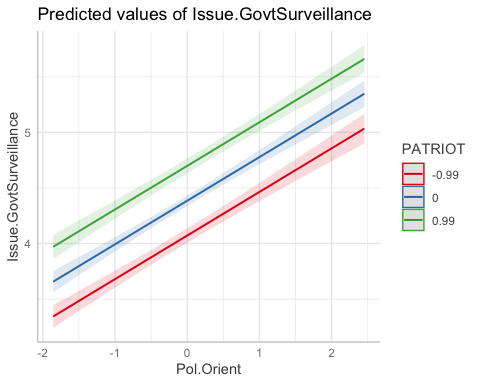
sjPlot::plot\_model(plot1)



plot(ggeffects::ggpredict(plot1, terms = c("Age","Gender")))



plot(ggeffects::ggpredict(plot1, terms = c("Pol.Orient","PATRIOT")))

 Model: Issue.GovtSurveillance = 4.41 + 0.05(Age) - 0.07(Gender) + 0.39(Pol.Orient) + 0.32(PATRIOT)

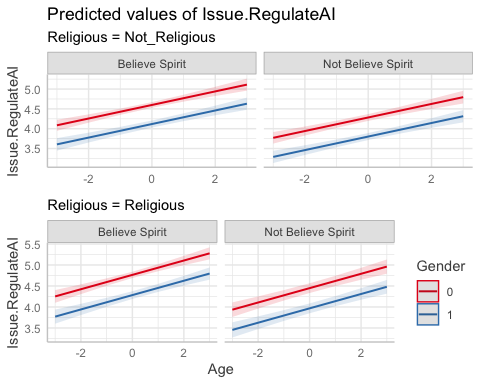
Age, Gender, Political Orientation, and level of Patriotism were plotted against attitudes towards government surveillance in a linear regression model. The effect of Age is non-significantly positive at p = 0.052, indicating that with the increase of age, positive attitudes towards government surveillance decreases although this effect is not statistically above chance. The effect of Gender is also non-significantly negative at p = 0.134, indicating that females tend to have a more positive outlook than males on government surveillance although this is effect is also not statistically above chance. The effect of Pol.Orient and PATRIOT are both significantly positive at p < .001, indicating that the more right-wing/conservative people are in their political ideologies, and also the more patriotic feelings they have towards the country, the more positive their attitudes are towards government surveillance. The linear trends of these relationships and their representativeness are clearly expressed in the plots. The parameters illustrate the size of predictor effects where political orientation has the largest effect in this model for predicting attitudes towards government surveillance, followed by patriotism, gender, and lastly age.

Demographic and ideological predictors of attitudes towards development and use of Artificial Intelligence (AI)

plot2 <- lm(Issue.RegulateAI ~ Age + Gender + Believe.Spirit + Religious, data = nz2018\_c)  
parameters::parameters(plot2)

## Parameter | Coefficient | SE | 95% CI | t(5571) | p  
## --------------------------------------------------------------------------------------------  
## (Intercept) | 4.60 | 0.04 | [ 4.53, 4.67] | 125.02 | < .001  
## Age | 0.17 | 0.02 | [ 0.13, 0.21] | 7.93 | < .001  
## Gender | -0.48 | 0.04 | [-0.57, -0.39] | -10.71 | < .001  
## Believe.Spirit [Not Believe Spirit] | -0.32 | 0.05 | [-0.41, -0.22] | -6.48 | < .001  
## Religious [Religious] | 0.17 | 0.05 | [ 0.07, 0.26] | 3.52 | < .001

plot(ggeffects::ggpredict(plot2, terms = c("Age","Gender","Believe.Spirit","Religious")))

 Model: Issue.RegulateAI = 4.60 + 0.17(Age) - 0.48(Gender) - 0.32(Believe.Spirit) + 0.17(Religious)

Age, Gender, Belief in Spirit, and Religiosity were plotted against attitudes towards development and use of Artificial Intelligence in a linear regression model. The effect of Age is significantly positive at p < .001, indicating that with the increase of age, attitudes for having strict and limiting regulations towards the development and use of AI increases as well. The effect of Gender is significantly negative at p < .001, indicating that males tend to be more against having strict and limiting regulations than females towards the development and use of AI. The effect of Believe.Spirit is significantly negative at p < .001, indicating that people who believes in spirits tend to be more supportive for strict and limiting regulations than people who do not believe in spirits towards the development and use of AI. The effect of Religious is significantly positive at p < .001, indicating that people who are more religious tend to be more suppotive for strict and limiting regulations than people who are less religious towards the development and use of AI. The linear trends of these relationships and their representativeness are clearly expressed in the plots. The parameters illustrate the size of predictor effects where gender has the largest effect in this model for predicting attitudes towards the development and use of AI, followed by belief in spirits, age, and lastly religiosity.