Covid Model Analysis

Joseph Heffner

26 May, 2021

# Setup

# Data

There are two data files: covid\_models\_data which holds all variables used in the cross-validated model of emotional distress and covid\_emo\_distress which holds the item-level responses for the emotional distress scale. The supplement of the manuscript holds more detailed information about all of the questionnaires included.

## Covid models data

## Covid emotion distress data

# Tables

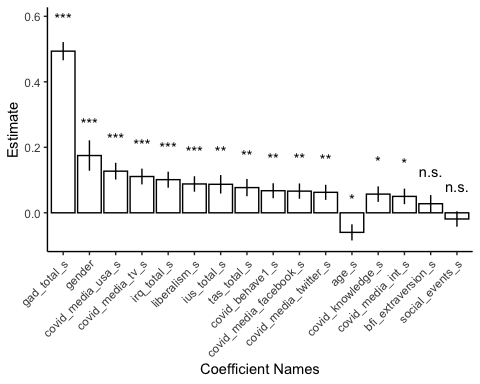
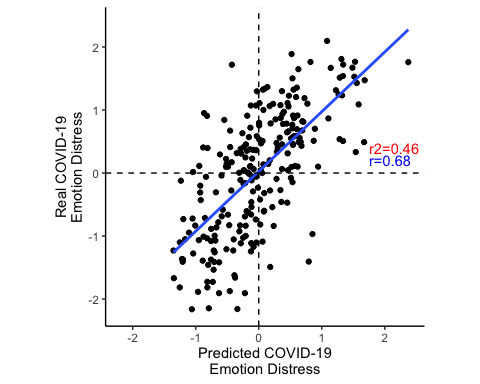
## Participant sample

##   
## level Overall   
## n 948   
## age (mean (SD)) 44.85 (15.94)  
## gender (%) female 491 (51.8)   
## male 457 (48.2)   
## race (%) black 126 (13.3)   
## east\_asian 42 ( 4.4)   
## hispanic\_latinx 37 ( 3.9)   
## middle\_eastern 3 ( 0.3)   
## mixed 15 ( 1.6)   
## native\_american 4 ( 0.4)   
## other 9 ( 0.9)   
## south\_asian 20 ( 2.1)   
## white 692 (73.0)   
## personal\_income (%) <$20,000 183 (19.3)   
## >$200,000 29 ( 3.1)   
## $100,000 to $149,999 88 ( 9.3)   
## $150,000 to $199,999 35 ( 3.7)   
## $20,000 to $34,999 161 (17.0)   
## $35,000 to $49,999 156 (16.5)   
## $50,000 to $74,999 179 (18.9)   
## $75,000 to $99,999 117 (12.3)   
## education (mean (SD)) 15.48 (2.59)

# Stepwise CV model

## Fit model

## Test model



Estimates

Predictors

Estimates

SE

t

p

Intercept

-0.09

0.03

-2.80

0.005

Alexithymia

0.08

0.03

2.93

0.004

Anxiety

0.49

0.03

17.75

<0.001

Interpersonal (IRQ)

0.10

0.02

4.14

<0.001

Extraversion

0.03

0.03

1.05

0.294

IUS

0.09

0.03

3.09

0.002

Liberalism

0.09

0.02

3.73

<0.001

Age

-0.06

0.02

-2.42

0.016

Covid Knowledge

0.06

0.02

2.41

0.016

Covid Behavior

0.07

0.02

2.95

0.003

Covid Media: Facebook

0.07

0.02

2.83

0.005

Covid Media: Twitter

0.06

0.02

2.70

0.007

Covid Media: TV

0.11

0.02

4.61

<0.001

Covid Media: USA

0.13

0.03

4.98

<0.001

Covid Media: International

0.05

0.02

2.11

0.035

Gender

0.17

0.05

3.77

<0.001

Social Events

-0.02

0.02

-0.80

0.425

Observations

948

(#tab:test\_model)

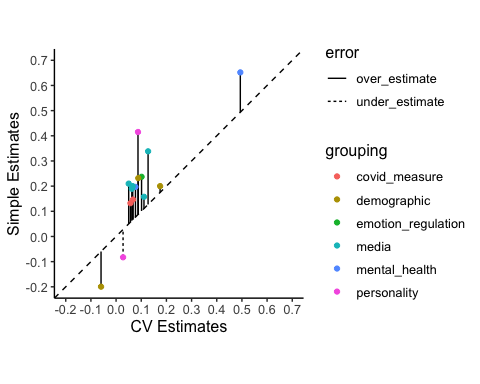
*Cross-validated regression model*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| predictor |  | 95% CI |  |  |
| Intercept | -0.09 | , | -2.80 | .005 |
| Alexithymia | 0.08 | , | 2.93 | .004 |
| Anxiety | 0.49 | , | 17.75 | < .001 |
| Interpersonal (IRQ) | 0.10 | , | 4.14 | < .001 |
| Extraversion | 0.03 | , | 1.05 | .294 |
| IUS | 0.09 | , | 3.09 | .002 |
| Liberalism | 0.09 | , | 3.73 | < .001 |
| Age | -0.06 | , | -2.42 | .016 |
| Covid Knowledge | 0.06 | , | 2.41 | .016 |
| Covid Behavior | 0.07 | , | 2.95 | .003 |
| Covid Media: Facebook | 0.07 | , | 2.83 | .005 |
| Covid Media: Twitter | 0.06 | , | 2.70 | .007 |
| Covid Media: TV | 0.11 | , | 4.61 | < .001 |
| Covid Media: USA | 0.13 | , | 4.98 | < .001 |
| Covid Media: International | 0.05 | , | 2.11 | .035 |
| Gender | 0.17 | , | 3.77 | < .001 |
| Social Events | -0.02 | , | -0.80 | .425 |

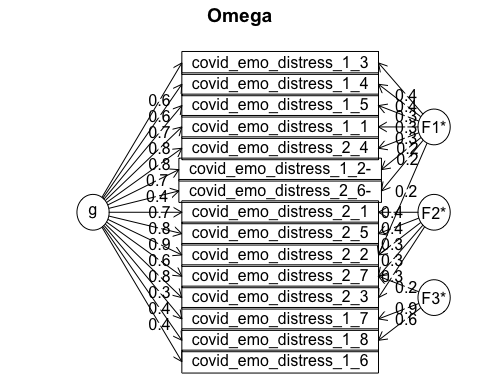
## Random forest

## Lasso

# Simple correlations



# Covid Emotion Distress Reliabilty



# Session Info

## R version 4.0.2 (2020-06-22)  
## Platform: x86\_64-apple-darwin17.0 (64-bit)  
## Running under: macOS Mojave 10.14.6  
##   
## Matrix products: default  
## BLAS: /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRblas.dylib  
## LAPACK: /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRlapack.dylib  
##   
## locale:  
## [1] en\_US.UTF-8/en\_US.UTF-8/en\_US.UTF-8/C/en\_US.UTF-8/en\_US.UTF-8  
##   
## attached base packages:  
## [1] stats graphics grDevices utils datasets methods base   
##   
## other attached packages:  
## [1] igraph\_1.2.5 ggraph\_2.0.2 corrr\_0.4.2 papaja\_0.1.0.9997  
## [5] sjPlot\_2.8.4 tableone\_0.12.0 corrplot\_0.84 caret\_6.0-86   
## [9] lattice\_0.20-41 psych\_1.9.12.31 ggrepel\_0.8.2 kableExtra\_1.1.0   
## [13] broom.mixed\_0.2.5 knitr\_1.28 forcats\_0.5.0 stringr\_1.4.0   
## [17] dplyr\_1.0.2 purrr\_0.3.4 readr\_1.3.1 tidyr\_1.1.2   
## [21] tibble\_3.0.3 ggplot2\_3.3.0 tidyverse\_1.3.0 conflicted\_1.0.4   
## [25] here\_0.1   
##   
## loaded via a namespace (and not attached):  
## [1] snow\_0.4-3 readxl\_1.3.1 backports\_1.1.6   
## [4] plyr\_1.8.6 GPArotation\_2014.11-1 TMB\_1.7.16   
## [7] splines\_4.0.2 TH.data\_1.0-10 digest\_0.6.25   
## [10] foreach\_1.5.0 htmltools\_0.5.1.1 viridis\_0.5.1   
## [13] fansi\_0.4.1 magrittr\_1.5 memoise\_1.1.0   
## [16] graphlayouts\_0.7.0 recipes\_0.1.11 modelr\_0.1.6   
## [19] gower\_0.2.1 sandwich\_2.5-1 colorspace\_1.4-1   
## [22] blob\_1.2.1 rvest\_0.3.5 mitools\_2.4   
## [25] haven\_2.3.1 xfun\_0.22 crayon\_1.3.4   
## [28] jsonlite\_1.6.1 lme4\_1.1-23 survival\_3.1-12   
## [31] zoo\_1.8-8 iterators\_1.0.12 glue\_1.4.0   
## [34] polyclip\_1.10-0 gtable\_0.3.0 ipred\_0.9-9   
## [37] emmeans\_1.4.8 webshot\_0.5.2 sjstats\_0.18.0   
## [40] sjmisc\_2.8.5 shape\_1.4.4 scales\_1.1.0   
## [43] mvtnorm\_1.1-0 DBI\_1.1.0 ggeffects\_0.15.1   
## [46] Rcpp\_1.0.4.6 viridisLite\_0.3.0 xtable\_1.8-4   
## [49] performance\_0.4.8 stats4\_4.0.2 lava\_1.6.7   
## [52] survey\_4.0 prodlim\_2019.11.13 glmnet\_4.0-2   
## [55] httr\_1.4.1 ellipsis\_0.3.0 farver\_2.0.3   
## [58] pkgconfig\_2.0.3 nnet\_7.3-14 dbplyr\_1.4.4   
## [61] labeling\_0.3 tidyselect\_1.1.0 rlang\_0.4.7   
## [64] reshape2\_1.4.4 effectsize\_0.3.2 munsell\_0.5.0   
## [67] cellranger\_1.1.0 tools\_4.0.2 cli\_2.0.2   
## [70] generics\_0.0.2 ranger\_0.12.1 sjlabelled\_1.1.6   
## [73] broom\_0.5.6 evaluate\_0.14 yaml\_2.2.1   
## [76] ModelMetrics\_1.2.2.2 fs\_1.4.1 tidygraph\_1.2.0   
## [79] nlme\_3.1-148 xml2\_1.3.2 compiler\_4.0.2   
## [82] rstudioapi\_0.11 e1071\_1.7-3 reprex\_0.3.0   
## [85] tweenr\_1.0.1 statmod\_1.4.34 stringi\_1.4.6   
## [88] highr\_0.8 parameters\_0.8.2 Matrix\_1.2-18   
## [91] nloptr\_1.2.2.1 vctrs\_0.3.4 pillar\_1.4.3   
## [94] lifecycle\_0.2.0 estimability\_1.3 data.table\_1.13.0   
## [97] insight\_0.9.0 R6\_2.4.1 gridExtra\_2.3   
## [100] codetools\_0.2-16 boot\_1.3-25 MASS\_7.3-51.6   
## [103] assertthat\_0.2.1 rprojroot\_1.3-2 withr\_2.2.0   
## [106] mnormt\_1.5-6 multcomp\_1.4-13 mgcv\_1.8-31   
## [109] doSNOW\_1.0.18 bayestestR\_0.7.2 parallel\_4.0.2   
## [112] hms\_0.5.3 labelled\_2.8.0 grid\_4.0.2   
## [115] rpart\_4.1-15 timeDate\_3043.102 coda\_0.19-3   
## [118] class\_7.3-17 minqa\_1.2.4 rmarkdown\_2.7   
## [121] ggforce\_0.3.1 pROC\_1.16.2 lubridate\_1.7.8

# Appendix

Code used for all figures and analyses.

# Knitr options  
knitr::opts\_chunk$set(echo = FALSE, warning = FALSE, message = FALSE)  
  
reg\_ouput <- TRUE # set to TRUE if want to knit document with regression table output  
cor\_output <- FALSE # knitting will not work with these graphs   
  
# Libraries  
library(here) # relative path  
library(conflicted) # resolve conflicts  
library(tidyverse) # tidy functions  
library(knitr) # knit functions  
library(broom.mixed) # tidy()  
library(kableExtra) # extra markdown functions  
library(ggrepel) # geom\_text\_repel  
library(psych) # alpha(), omega()  
library(caret) # train-test functions  
library(corrplot) # correlation plot  
library(tableone) # tables  
library(sjPlot) # regression tables  
library(papaja) # apa regression tables, requires an install from github  
library(corrr) # correlations  
library(ggraph) # graph visualizations  
library(igraph) # graph visualizations  
  
  
# Resolve conflicts  
conflict\_prefer("select", "dplyr")  
conflict\_prefer("filter", "dplyr")  
# Relative paths  
dir\_parent <- here()  
dir\_data <- str\_c(dir\_parent, "/data")  
dir\_graphs <- str\_c(dir\_parent, "/graphs")  
  
# Data  
df\_models <- read\_csv(str\_c(dir\_data, "/covid\_models\_data.csv")) %>% select(-sub)   
df\_emo\_distress <- read\_csv(str\_c(dir\_data, "/covid\_emo\_distress.csv")) %>% select(-sub)  
df\_demo <- read\_csv(str\_c(dir\_data, "/demographics.csv"))  
table1 <- tableone::CreateTableOne(data = df\_demo %>% select(age, gender, race, personal\_income, education))  
  
table1\_matrix <- print(table1, showAllLevels = TRUE, formatOptions = list(big.mark = ","))  
  
write.csv(table1\_matrix, file = str\_c(dir\_graphs, "/table1.csv")) # note this was later converted into latex  
#### Test-train split  
set.seed(123)  
  
index\_train <- as.numeric(caret::createDataPartition(y = df\_models$covid\_emo\_distress\_s, p = 0.75, list = FALSE, times = 1))  
df\_train\_raw <- df\_models[index\_train, ]  
df\_test\_raw <- df\_models[-index\_train, ]  
  
#### Stepwise lm   
df\_train\_lm <- df\_train\_raw  
df\_test\_lm <- df\_test\_raw  
  
numCores <- parallel::detectCores() - 1 # never want to use all cores   
cl <- parallel::makeCluster(numCores, type = "SOCK")  
doSNOW::registerDoSNOW(cl) # register cluster so caret knows to train in parallel  
  
train\_control <- trainControl(method = "cv", number = 10)  
lm\_fit <- train(covid\_emo\_distress\_s ~ .,   
 data = df\_train\_lm,   
 method = "lmStepAIC",   
 trControl = train\_control, trace = FALSE, # prevent output  
 direction = "both") # forward and backward  
parallel::stopCluster(cl) # stop cluster after training  
#### Test model  
df\_test\_lm$model\_fits <- predict(lm\_fit, newdata = df\_test\_lm)  
df\_test\_results <- postResample(pred = df\_test\_lm$model\_fits, obs = df\_test\_lm$covid\_emo\_distress\_s)  
cor\_value <- cor(df\_test\_lm$model\_fits, df\_test\_lm$covid\_emo\_distress\_s)  
var\_value <- cor\_value^2  
  
#### Calibration check   
m1 <- lm(covid\_emo\_distress\_s ~ model\_fits, data = df\_test\_lm)  
#summary(m1)  
  
df\_test\_plot <- ggplot(df\_test\_lm, aes(x = model\_fits, y = covid\_emo\_distress\_s)) +   
 geom\_point() +   
 geom\_smooth(method = "lm", se = F) +   
 scale\_y\_continuous(name = "Real COVID-19\nEmotion Distress", limits = c(-2.2, 2.4)) +   
 scale\_x\_continuous(name = "Predicted COVID-19\nEmotion Distress", limits = c(-2.2, 2.4)) +   
 coord\_fixed() +   
 geom\_hline(aes(yintercept = 0), linetype = "dashed") +   
 geom\_vline(aes(xintercept = 0), linetype = "dashed") +   
 annotate(geom="text", y = 0.2, x = 1.75, label = str\_c("r=", round(cor\_value, 2)), color = "blue", hjust = 0) +   
 annotate(geom="text", y = 0.4, x = 1.75, label = str\_c("r2=", round(var\_value, 2)), color = "red", hjust = 0) +   
 theme\_classic()  
df\_test\_plot  
ggsave(filename = str\_c(dir\_graphs, "/fig2a.pdf"), plot = df\_test\_plot, width = 4, height = 4)  
  
#### Refit on entire data set  
best\_model <- as.character(lm\_fit$finalModel$call[2]) %>% str\_replace(., ".outcome", "covid\_emo\_distress\_s")  
final\_model <- lm(formula = as.formula(best\_model), data = df\_models)  
final\_summary <- summary(final\_model)  
  
final\_results <- as.data.frame(final\_summary$coefficients) %>%  
 tibble::rownames\_to\_column(var = "coefficient") %>%  
 mutate(p\_values = case\_when(`Pr(>|t|)` <= .001 ~ "\*\*\*",   
 `Pr(>|t|)` <= .01 ~ "\*\*",   
 `Pr(>|t|)` <= .05 ~ "\*",   
 `Pr(>|t|)` > .05 ~ "n.s.")) %>%  
 filter(coefficient != "(Intercept)")  
  
final\_betas <- ggplot(final\_results, aes(x = reorder(coefficient, -abs(Estimate)), y = Estimate)) +   
 geom\_bar(stat = "identity", position = position\_dodge(.9), fill = "white", color = "black") +   
 geom\_errorbar(aes(ymin = Estimate - `Std. Error`, ymax = Estimate + `Std. Error`), width = 0) +   
 geom\_text(aes(label = p\_values, y = Estimate + .1)) +  
 xlab("Coefficient Names") +   
 theme\_classic() +  
 theme(axis.text.x = element\_text(angle = 45, hjust= 1))  
final\_betas  
ggsave(filename = str\_c(dir\_graphs, "/fig2b.pdf"), plot = final\_betas, width = 7, height = 4)  
  
#### Export regression results  
tab\_model(final\_model,   
 pred.labels = c("Intercept", "Alexithymia", "Anxiety", "Interpersonal (IRQ)",   
 "Extraversion", "IUS", "Liberalism", "Age", "Covid Knowledge",   
 "Covid Behavior", "Covid Media: Facebook", "Covid Media: Twitter",   
 "Covid Media: TV", "Covid Media: USA", "Covid Media: International", "Gender", "Social Events"),  
 dv.labels = c("Estimates"), string.se = "SE", string.stat = "t",  
 show.se = TRUE, show.stat = TRUE, show.ci = FALSE,   
 show.re.var = FALSE, show.aic = FALSE,  
 show.r2 = FALSE, show.icc = FALSE, show.obs = TRUE,  
 CSS = css\_theme("regression"), file = str\_c(dir\_graphs, "/table2.html"))  
  
if (reg\_ouput == TRUE) {  
 apa\_lm <- papaja::apa\_print(final\_model)   
 apa\_lm$table$predictor <- c("Intercept", "Alexithymia", "Anxiety", "Interpersonal (IRQ)",   
 "Extraversion", "IUS", "Liberalism", "Age", "Covid Knowledge",   
 "Covid Behavior", "Covid Media: Facebook", "Covid Media: Twitter",   
 "Covid Media: TV", "Covid Media: USA", "Covid Media: International", "Gender", "Social Events")  
 papaja::apa\_table(apa\_lm$table,   
 caption = "Cross-validated regression model")  
}  
## RF specific   
df\_train\_rf <- df\_train\_raw  
df\_test\_rf <- df\_test\_raw  
  
numCores <- parallel::detectCores() - 1   
cl <- parallel::makeCluster(numCores, type = "SOCK")  
doSNOW::registerDoSNOW(cl)   
  
train\_control <- trainControl(method = "cv", number = 10)  
forest\_fit <- train(covid\_emo\_distress\_s ~ .,   
 data = df\_train\_rf,   
 method = "ranger",   
 trControl = train\_control,   
 importance = "impurity")  
parallel::stopCluster(cl)   
  
## Test correlation   
df\_test\_rf$model\_fits <- predict(forest\_fit, newdata = df\_test\_rf)  
df\_test\_results\_rf <- postResample(pred = df\_test\_rf$model\_fits, obs = df\_test\_rf$covid\_emo\_distress\_s)  
cor\_rf <- cor(df\_test\_rf$model\_fits, df\_test\_rf$covid\_emo\_distress\_s)  
## Lasso specific  
df\_train\_ls <- df\_train\_raw  
df\_test\_ls <- df\_test\_raw  
  
numCores <- parallel::detectCores() - 1   
cl <- parallel::makeCluster(numCores, type = "SOCK")  
doSNOW::registerDoSNOW(cl)   
  
train\_control <- trainControl(method = "cv", number = 10)  
tune\_control <- expand.grid(alpha = 1, lambda = 10^seq(-3, 3, length = 100))  
lasso\_fit <- train(covid\_emo\_distress\_s ~ .,   
 data = df\_train\_ls,   
 method = "glmnet",   
 trControl = train\_control,  
 tuneGrid = tune\_control)  
parallel::stopCluster(cl)  
  
## Test correlation   
df\_test\_ls$model\_fits <- predict(lasso\_fit, newdata = df\_test\_ls)  
df\_test\_results\_ls <- postResample(pred = df\_test\_ls$model\_fits, obs = df\_test\_ls$covid\_emo\_distress\_s)  
cor\_ls <- cor(df\_test\_ls$model\_fits, df\_test\_ls$covid\_emo\_distress\_s)  
# Final model is cv  
df\_cv <- tidy(final\_model, conf.int = TRUE) %>%  
 mutate(p.labels = case\_when(p.value <= .001 ~ "\*\*\*",   
 p.value <= .01 ~ "\*\*",   
 p.value <= .05 ~ "\*",   
 p.value > .05 ~ "n.s."))  
  
# Simple relationships  
df\_simples <- df\_models %>%  
 pivot\_longer(cols = -covid\_emo\_distress\_s, names\_to = "term", values\_to = "value") %>%   
 group\_by(term) %>%   
 do(tidy(cor.test(.$covid\_emo\_distress\_s, .$value))) %>%  
 select(term:conf.high) %>%  
 mutate(p.labels = case\_when(p.value <= .001 ~ "\*\*\*",   
 p.value <= .01 ~ "\*\*",   
 p.value <= .05 ~ "\*",   
 p.value > .05 ~ "n.s."))  
  
# Full  
df\_cv\_full <- df\_cv %>%  
 filter(term != "(Intercept)") %>%  
 full\_join(., df\_simples, by = "term", suffix = c("\_cross\_valid", "\_simple")) %>%  
 mutate(grouping = case\_when(term == "gad\_total\_s" ~ "mental\_health",   
 term == "ius\_total\_s" ~ "personality",   
 term == "covid\_media\_usa\_s" ~ "media",   
 term == "covid\_media\_facebook\_s" ~ "media",   
 term == "covid\_media\_int\_s" ~ "media",   
 term == "covid\_media\_twitter\_s" ~ "media",   
 term == "covid\_media\_tv\_s" ~ "media",   
 term == "tas\_total\_s" ~ "mental\_health",   
 term == "irq\_total\_s" ~ "emotion\_regulation",   
 term == "covid\_behave1\_s" ~ "covid\_measure",   
 term == "covid\_knowledge\_s" ~ "covid\_measure",   
 term == "bfi\_extraversion\_s" ~ "personality",   
 term == "gender" ~ "demographic",   
 term == "age\_s" ~ "demographic",   
 term == "liberalism\_s" ~ "demographic",   
 term == "delta\_social\_s" ~ "social"),   
 error = case\_when(((estimate\_cross\_valid < estimate\_simple) & estimate\_cross\_valid > 0) ~ "over\_estimate",   
 ((estimate\_cross\_valid > estimate\_simple) & estimate\_cross\_valid < 0) ~ "over\_estimate",   
 ((estimate\_cross\_valid < estimate\_simple) & estimate\_cross\_valid < 0) ~ "under\_estimate",   
 ((estimate\_cross\_valid > estimate\_simple) & estimate\_cross\_valid > 0) ~ "under\_estimate"))  
  
# Comparison  
df\_compare <- df\_cv\_full %>% filter(!is.na(grouping))  
  
model\_compare <- ggplot(df\_compare, aes(x = estimate\_cross\_valid, y = estimate\_simple)) +   
 geom\_segment(aes(x = estimate\_cross\_valid, y = estimate\_simple, xend = estimate\_cross\_valid, yend = estimate\_cross\_valid, linetype = error)) +   
 geom\_point(aes(color = grouping)) +   
 #ggrepel::geom\_label\_repel(aes(label = term), size = 2) +   
 geom\_abline(slope = 1, intercept = 0, linetype = 2) +   
 scale\_x\_continuous(name = "CV Estimates", limits = c(-.2, 0.7), breaks = seq(-.2, .7, .1)) +   
 scale\_y\_continuous(name = "Simple Estimates", limits = c(-.2, 0.7), breaks = seq(-.2, .7, .1)) +   
 coord\_fixed() +   
 theme\_classic() +   
 theme(text = element\_text(size = 12))  
model\_compare  
ggsave(filename = str\_c(dir\_graphs, "/fig1b.pdf"), plot = model\_compare, width = 5, height = 4)  
  
if (cor\_output == TRUE) {  
 ## Correlation plot  
 final\_variables <- as.data.frame(final\_summary$coefficients) %>%  
 tibble::rownames\_to\_column(var = "coefficient") %>%  
 filter(coefficient != "(Intercept)") %>%  
 pull(coefficient)  
   
 cor\_data <- cor(df\_models %>% select(all\_of(final\_variables)), use = "na.or.complete") # correlation of all variables   
 res1 <- cor.mtest(df\_models %>% select(all\_of(final\_variables)), conf.level = .95) # significance.  
   
 ## Correlation network  
 cor\_df <- df\_models %>%   
 select(all\_of(final\_variables), covid\_emo\_distress\_s) %>%   
 correlate() %>%  
 stretch() %>%  
 mutate(x = case\_when(x == "gad\_total\_s" ~ "Anxiety",   
 x == "ius\_total\_s" ~ "IUS",   
 x == "covid\_media\_usa\_s" ~ "Media: USA",   
 x == "covid\_media\_facebook\_s" ~ "Media: Facebook",   
 x == "covid\_media\_int\_s" ~ "Media: International",   
 x == "covid\_media\_twitter\_s" ~ "Media: Twitter",   
 x == "covid\_media\_tv\_s" ~ "Media: TV",   
 x == "tas\_total\_s" ~ "Alexithymia",   
 x == "irq\_total\_s" ~ "Interpersonal (IRQ)",   
 x == "covid\_behave1\_s" ~ "Covid Behavior",   
 x == "covid\_knowledge\_s" ~ "Covid Knowledge",   
 x == "bfi\_extraversion\_s" ~ "Extraversion",   
 x == "gender" ~ "Gender",   
 x == "age\_s" ~ "Age",   
 x == "liberalism\_s" ~ "Liberalism",   
 x == "social\_events\_s" ~ "Social Events",   
 x == "covid\_emo\_distress\_s" ~ "Covid Emotional Distress"),   
 y = case\_when(y == "gad\_total\_s" ~ "Anxiety",   
 y == "ius\_total\_s" ~ "IUS",   
 y == "covid\_media\_usa\_s" ~ "Media: USA",   
 y == "covid\_media\_facebook\_s" ~ "Media: Facebook",   
 y == "covid\_media\_int\_s" ~ "Media: International",   
 y == "covid\_media\_twitter\_s" ~ "Media: Twitter",   
 y == "covid\_media\_tv\_s" ~ "Media: TV",   
 y == "tas\_total\_s" ~ "Alexithymia",   
 y == "irq\_total\_s" ~ "Interpersonal (IRQ)",   
 y == "covid\_behave1\_s" ~ "Covid Behavior",   
 y == "covid\_knowledge\_s" ~ "Covid Knowledge",   
 y == "bfi\_extraversion\_s" ~ "Extraversion",   
 y == "gender" ~ "Gender",   
 y == "age\_s" ~ "Age",   
 y == "liberalism\_s" ~ "Liberalism",   
 y == "social\_events\_s" ~ "Social Events",   
 y == "covid\_emo\_distress\_s" ~ "Covid Emotional Distress"))  
   
 fig3\_data <- cor\_df %>%  
 filter(abs(r) > .2) %>%  
 graph\_from\_data\_frame(directed = FALSE)  
   
 fig3\_plot <- ggraph(fig3\_data) +  
 geom\_edge\_link(aes(color = r)) +  
 guides(edge\_alpha = "none", edge\_width = "none") +  
 scale\_edge\_colour\_gradientn(limits = c(-1, 1), colors = c("firebrick2", "dodgerblue2")) +  
 geom\_node\_point(size = 3) +  
 geom\_node\_text(aes(label = name), repel = TRUE) +  
 theme\_graph(base\_family = 'Helvetica')  
 fig3\_plot  
   
 ggsave(filename = str\_c(dir\_graphs, "/fig3.pdf"), plot = fig3\_plot, width = 6, height = 4)  
 ggsave(filename = str\_c(dir\_graphs, "/fig3.png"), plot = fig3\_plot, width = 6.666, height = 5)  
}  
# Alpha  
alpha\_emo <- psych::alpha(df\_emo\_distress, keys = c("covid\_emo\_distress\_1\_2", "covid\_emo\_distress\_2\_6")) # keys are reverse coded items  
#alpha\_emo$total$raw\_alpha  
  
# Omega (ERF)  
emo\_r\_matrix <- cor(df\_emo\_distress)  
omega\_emo <- omega(emo\_r\_matrix) # bifactor factor solution   
#omegaSem(m = emo\_r\_matrix, n.obs = 948)  
  
## Plot  
#pdf(file = str\_c(dir\_graphs, "omega\_plot.pdf"))  
#omega(emo\_r\_matrix)  
#dev.off()  
sessionInfo()