BS_Rep_D1

Beza & Solome

2023-03-14

Data Wrangling

```
##Loading Both Waves of pew_ATP_W64 Data
pew_ATP_W64 <- read_dta("ATP_W64.dta") #this is in haven label type


#subetting relevant portion of wave 64
pew_ATP_W64 <- subset(pew_ATP_W64,
                      select=c(QKEY,COVID_COMFORT_a_W64,COVID_COMFORT_b_W64,
                            COVID_COMFORT_c_W64,COVID_COMFORT_d_W64,COVID_COMFORT_e_W64,
                            COVID_RESTRICTION_a_W64,COVID_RESTRICTION_b_W64,
                            COVID_RESTRICTION_c_W64,COVID_RESTRICTION_d_W64,
                            COVID_RESTRICTION_e_W64,COVID_RESTRICTION_f_W64,
                            COVID_RESTRICTION_g_W64,F_IDEO,F_PARTY_FINAL,F_SEX,
                            F_AGECAT,F_EDUCCAT2,F_MARITAL,F_INCOME,F_CREGION,
                            F_RACETHN,WEIGHT_W64))
```

Coding and scaling ideological dependent variable (DV) of interest (policy support)

```
#subsetting DV data
covid_ideo_scales <- subset(pew_ATP_W64,
                            select=c(QKEY,COVID_RESTRICTION_a_W64,COVID_RESTRICTION_b_W64,
                                     COVID_RESTRICTION_c_W64,COVID_RESTRICTION_d_W64,
                                     COVID_RESTRICTION_e_W64,COVID_RESTRICTION_f_W64,
                                     COVID_RESTRICTION_g_W64,F_IDEO))

#renaming column names of subsetted data
colnames(covid_ideo_scales) <- c("QKEY","restriction_intl_travel","restriction_most_business",
                                 "restriction_large_gatherings","restriction_sporting_events",
                                 "restriction_closing_k12","restriction_carry_out_only",
                                 "restriction_postponing_primary","libcon")

#transforming observations into characters (had to redone long way, orginal code did not work)
covid_ideo_scales[,2] <- as.character(covid_ideo_scales$restriction_intl_travel)
covid_ideo_scales[,3] <- as.character(covid_ideo_scales$restriction_most_business)
covid_ideo_scales[,4] <- as.character(covid_ideo_scales$restriction_large_gatherings)
covid_ideo_scales[,5] <- as.character(covid_ideo_scales$restriction_sporting_events)
covid_ideo_scales[,6] <- as.character(covid_ideo_scales$restriction_closing_k12)
covid_ideo_scales[,7] <- as.character(covid_ideo_scales$restriction_carry_out_only)
```

```r
covid_ideo_scales[,8] <- as.character(covid_ideo_scales$restriction_postponing_primary)


#recoding observations into binary --> don't use if else because you don't want to improperly code the
covid_ideo_scales$restriction_intl_travel <- recode(covid_ideo_scales$restriction_intl_travel,
                            "1"= 1, "2"= 0)
covid_ideo_scales$restriction_most_business <- recode(covid_ideo_scales$restriction_most_business,
                            "1"= 1, "2"= 0)
covid_ideo_scales$restriction_large_gatherings <- recode(covid_ideo_scales$restriction_large_gatherings
                            "1"= 1, "2"= 0)
covid_ideo_scales$restriction_sporting_events <- recode(covid_ideo_scales$restriction_sporting_events,
                            "1"= 1, "2"= 0)
covid_ideo_scales$restriction_closing_k12 <- recode(covid_ideo_scales$restriction_closing_k12,
                            "1"= 1, "2"= 0)
covid_ideo_scales$restriction_carry_out_only <- recode(covid_ideo_scales$restriction_carry_out_only,
                            "1"= 1, "2"= 0)
covid_ideo_scales$restriction_postponing_primary <- recode(covid_ideo_scales$restriction_postponing_pri
                            "1"= 1, "2"= 0)

#recoding observations into binary --> this currently (3/17) doesn't work because the variables have no
# covid_ideo_scales[covid_ideo_scales == "Necessary"] <- 1
# covid_ideo_scales[covid_ideo_scales == "Unnecessary"] <- 0

#recoding political ideology variable --> currently (3/17) this yields a bunch of NA's that end up maki
# covid_ideo_scales$libcon <- ifelse(covid_ideo_scales$libcon %in% "Very Conservative",1,
#                                    ifelse(covid_ideo_scales$libcon %in% "Conservative",2,
#                                           ifelse(covid_ideo_scales$libcon %in% "Moderate",3,
#                                                  ifelse(covid_ideo_scales$libcon %in% "Liberal",4,
#                                                         ifelse(covid_ideo_scales$libcon %in% "Liberal
#                                                                ifelse(covid_ideo_scales$libcon %in% "

#transforming observations into numeric --> this doesn't work (as of 3/17) and I am not sure it is nece
# for(i in 2:ncol(covid_ideo_scales)){
#   covid_ideo_scales[,i] <- as.numeric(covid_ideo_scales[,i])
# }


#working on the political ideology variable
library(haven)
library(dplyr)

# Convert to regular factor vector
covid_ideo_scales$libcon <- as_factor(covid_ideo_scales$libcon)

# Recode using case_when()
covid_ideo_scales <- covid_ideo_scales %>%
  mutate(
    libcon = case_when(
      libcon == "Very conservative" ~ 1,
      libcon == "Conservative" ~ 2,
      libcon == "Moderate" ~ 3,
      libcon == "Liberal" ~ 4,
      libcon == "Very liberal" ~ 5,
```

```
      TRUE ~ NA_real_
    )
  )

# convert back to numeric vector
covid_ideo_scales$libcon <- as.numeric(covid_ideo_scales$libcon)

# converting columns to numeric
covid_ideo_scales[, 2:ncol(covid_ideo_scales)] <- lapply(covid_ideo_scales[, 2:ncol(covid_ideo_scales)]

# removing incomplete observations
covid_ideo_scales <- na.omit(covid_ideo_scales)
```

Factor Analysis (policy support)

```
library(FactoMineR)
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

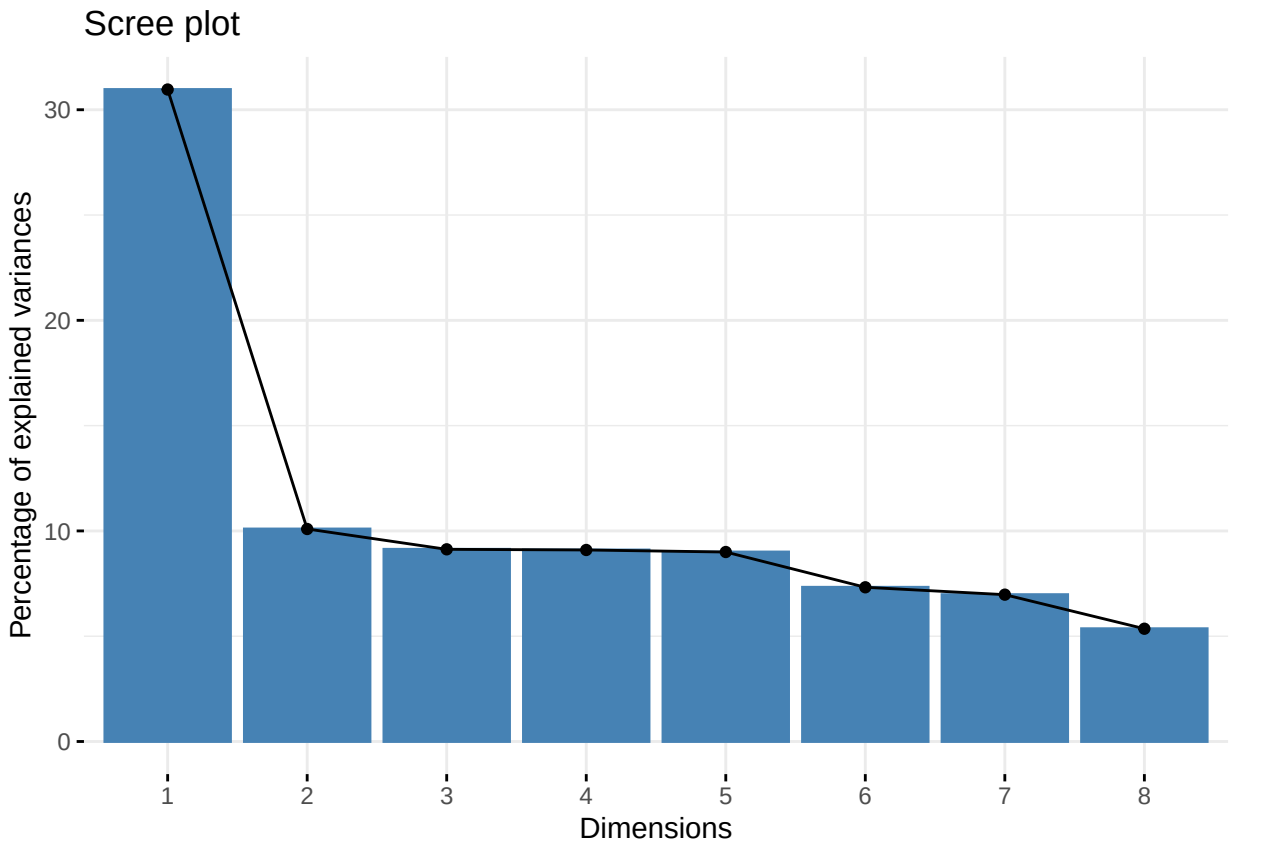
```
#transforming observations into factors for factor analysis
x <- covid_ideo_scales[,c(2:9)]
# for(i in 1:8){
#   x[,i] <- as.factor(x[,i])
# }

#trying changing observations into factors the long way
x[,1] <- as.factor(covid_ideo_scales$restriction_intl_travel)
x[,2] <- as.factor(covid_ideo_scales$restriction_most_business)
x[,3] <- as.factor(covid_ideo_scales$restriction_large_gatherings)
x[,4] <- as.factor(covid_ideo_scales$restriction_sporting_events)
x[,5] <- as.factor(covid_ideo_scales$restriction_closing_k12)
x[,6] <- as.factor(covid_ideo_scales$restriction_carry_out_only)
x[,7] <- as.factor(covid_ideo_scales$restriction_postponing_primary)
x[,8] <- as.factor(covid_ideo_scales$libcon)

factanal <- FAMD(x, graph = FALSE,ncp=8,sup.var = NULL)
library(rstatix)
```

```
##
## Attaching package: 'rstatix'
```

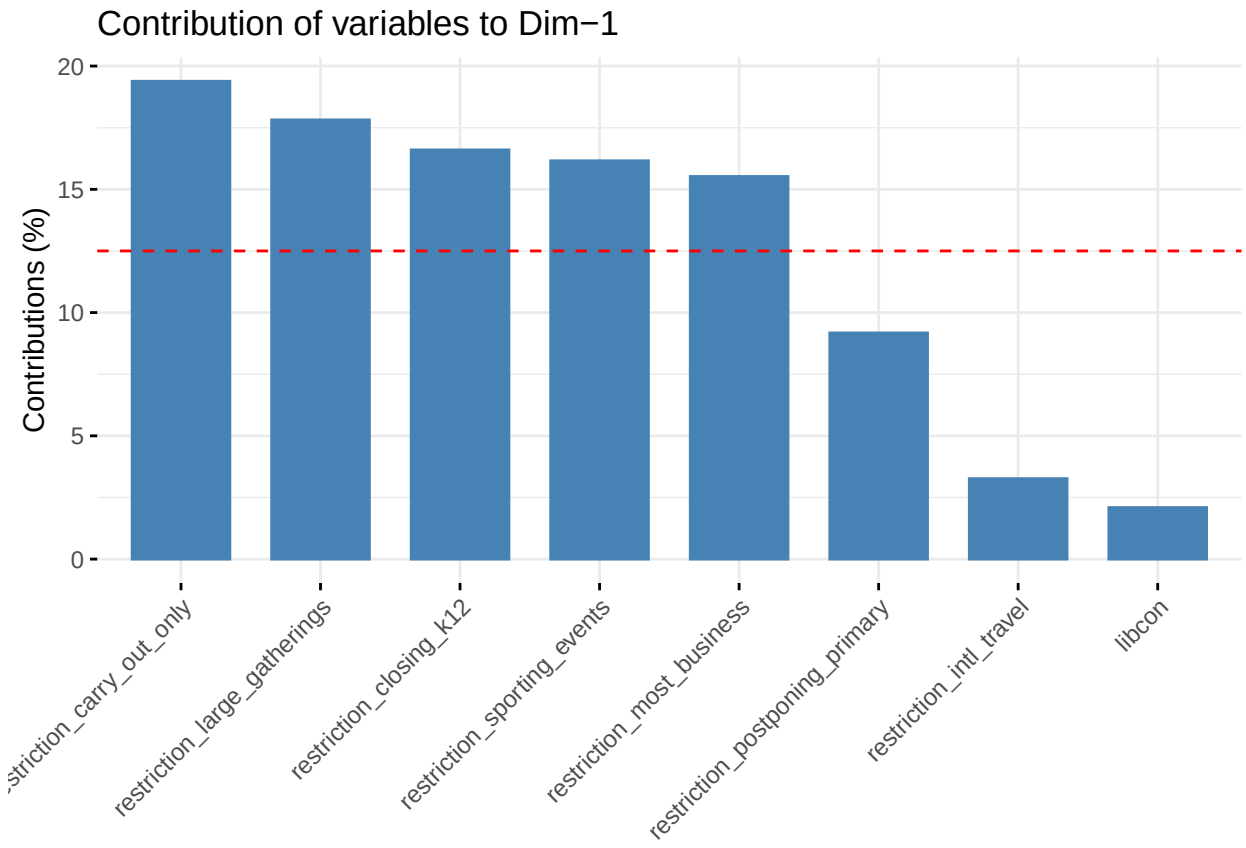
```
## The following objects are masked from 'package:plyr':
##
##     desc, mutate
```

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

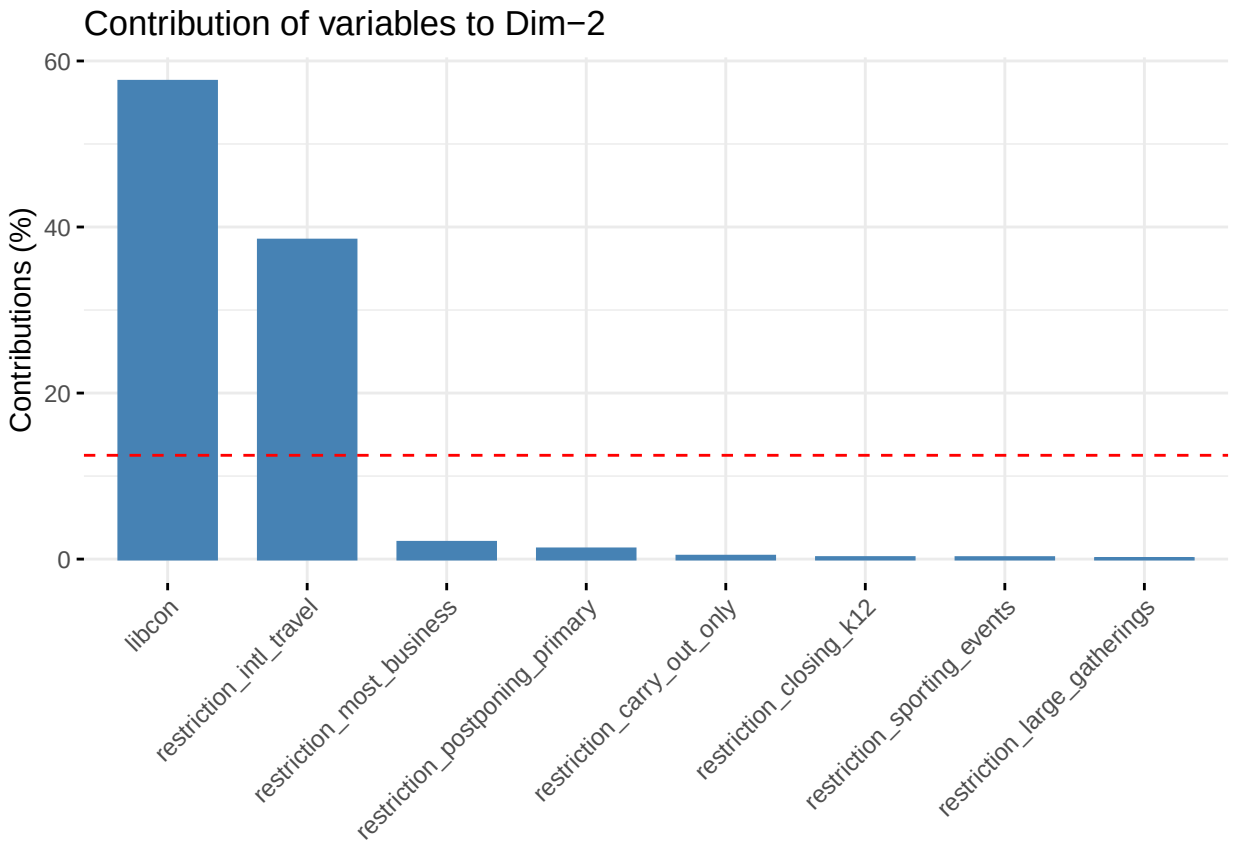
```
fviz_screeplot(factanal)
```



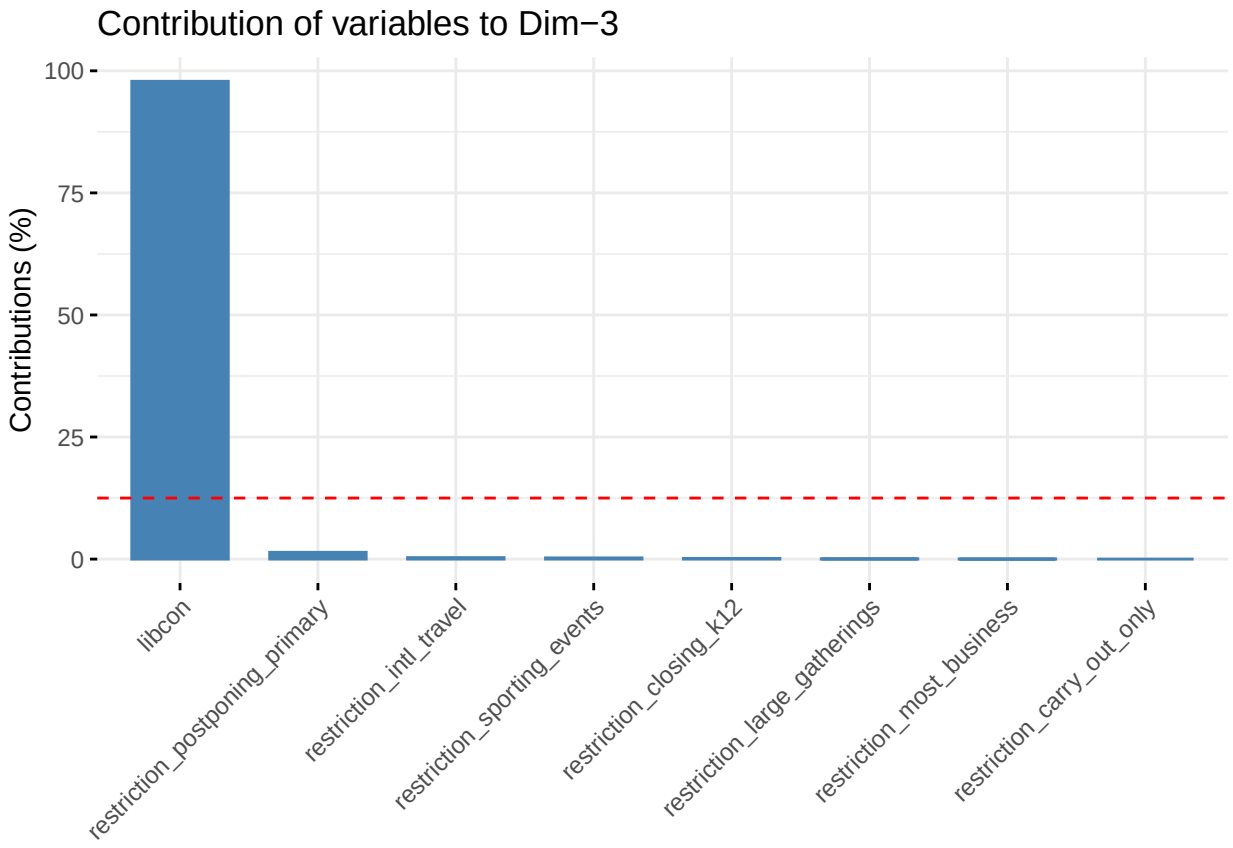

```
fviz_contrib(factanal, "var", axes = 1)
```



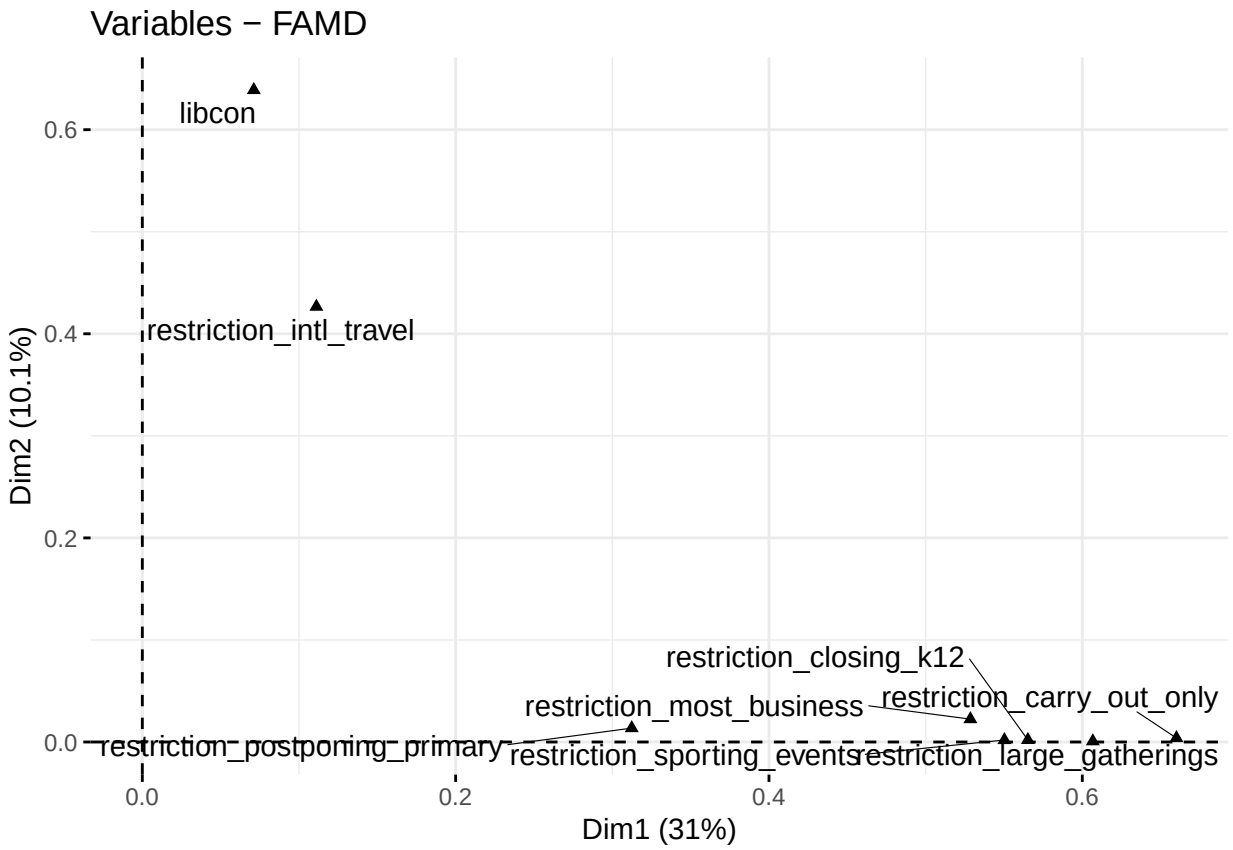

```
fviz_contrib(factanal, "var", axes = 2)
```




```
fviz_contrib(factanal, "var", axes = 3)
```




```
fviz_famd_var(factanal, "var", repel = TRUE, col.var = "black")
```

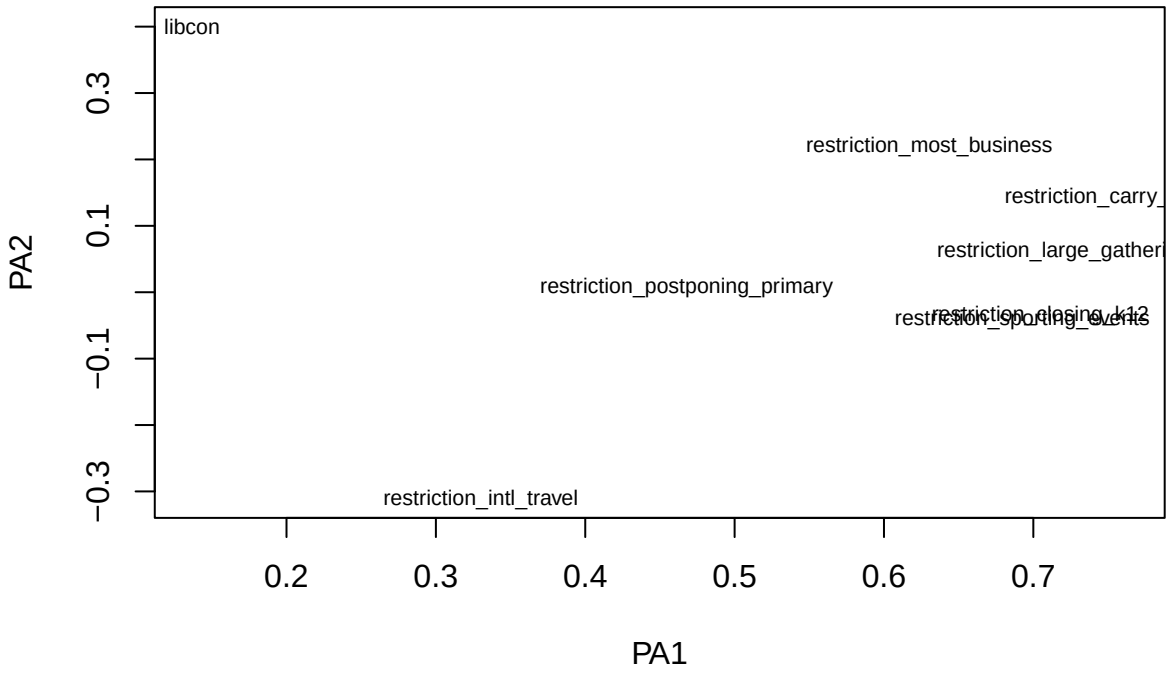



```
# Run factor analysis
factanal <- fa(covid_ideo_scales[,c(2:9)], nfactors=2, rotate="promax", fm="pa")


factanal <- fa(covid_ideo_scales[,c(2:9)], nfactors=2, rotate="promax", fm="pa")
scores <- data.frame(factanal$scores)
loadings(factanal)
```

```
##
## Loadings:
##                               PA1    PA2
## restriction_intl_travel        0.330 -0.311
## restriction_most_business      0.630  0.220
## restriction_large_gatherings   0.724
## restriction_sporting_events    0.693
## restriction_closing_k12        0.705
## restriction_carry_out_only     0.763  0.142
## restriction_postponing_primary 0.468
## libcon                         0.137  0.401
##
##                  PA1   PA2
## SS loadings    2.827 0.333
## Proportion Var 0.353 0.042
## Cumulative Var 0.353 0.395
```

```
loadings <- factanal$loadings
loadings <- data.frame(f1 = loadings[,1],f2=loadings[,2])
```

```
plot(factanal$loadings,type="n") # set up plot
text(factanal$loadings,labels=names(covid_ideo_scales)[2:9],cex=.7) # add variable names
```




```
library(ggrepel)

alpha(covid_ideo_scales[,c(2:9)])
```

```
##
## Reliability analysis
## Call: alpha(x = covid_ideo_scales[, c(2:9)])
##
##   raw_alpha std.alpha G6(smc) average_r S/N    ase mean   sd median_r
##        0.6      0.78    0.78      0.31 3.5 0.0056  1.1 0.25      0.3
##
##     95% confidence boundaries
##          lower alpha upper
## Feldt     0.59   0.6  0.61
## Duhachek  0.59   0.6  0.61
##
##  Reliability if an item is dropped:
##                               raw_alpha std.alpha G6(smc) average_r S/N
## restriction_intl_travel            0.61      0.80    0.79      0.36 3.9
## restriction_most_business          0.51      0.74    0.74      0.29 2.8
## restriction_large_gatherings       0.54      0.73    0.73      0.28 2.7
## restriction_sporting_events        0.55      0.74    0.74      0.28 2.8
## restriction_closing_k12            0.55      0.73    0.73      0.28 2.8
## restriction_carry_out_only         0.52      0.72    0.72      0.27 2.6
## restriction_postponing_primary     0.56      0.77    0.77      0.32 3.3
## libcon                             0.79      0.81    0.80      0.37 4.2
##                               alpha se var.r med.r
## restriction_intl_travel          0.0057 0.028  0.38
## restriction_most_business        0.0069 0.035  0.22
## restriction_large_gatherings     0.0064 0.030  0.22
```

```
## restriction_sporting_events      0.0062 0.033  0.22
## restriction_closing_k12          0.0063 0.032  0.21
## restriction_carry_out_only       0.0066 0.028  0.22
## restriction_postponing_primary   0.0061 0.038  0.22
## libcon                           0.0029 0.021  0.38
##
##  Item statistics
##                                    n raw.r std.r r.cor r.drop mean   sd
## restriction_intl_travel        11104  0.24  0.41  0.25   0.14 0.96 0.20
## restriction_most_business      11104  0.68  0.71  0.67   0.54 0.76 0.43
## restriction_large_gatherings   11104  0.64  0.75  0.72   0.54 0.91 0.29
## restriction_sporting_events    11104  0.59  0.72  0.68   0.50 0.94 0.24
## restriction_closing_k12        11104  0.60  0.73  0.69   0.51 0.93 0.26
## restriction_carry_out_only     11104  0.69  0.78  0.77   0.59 0.89 0.32
## restriction_postponing_primary 11104  0.54  0.57  0.47   0.34 0.70 0.46
## libcon                         11104  0.66  0.35  0.18   0.17 3.03 1.06
##
## Non missing response frequency for each item
##                                   0    1    2    3    4   5 miss
## restriction_intl_travel        0.04 0.96 0.00 0.00 0.00 0.0    0
## restriction_most_business      0.24 0.76 0.00 0.00 0.00 0.0    0
## restriction_large_gatherings   0.09 0.91 0.00 0.00 0.00 0.0    0
## restriction_sporting_events    0.06 0.94 0.00 0.00 0.00 0.0    0
## restriction_closing_k12        0.07 0.93 0.00 0.00 0.00 0.0    0
## restriction_carry_out_only     0.11 0.89 0.00 0.00 0.00 0.0    0
## restriction_postponing_primary 0.30 0.70 0.00 0.00 0.00 0.0    0
## libcon                         0.00 0.08 0.23 0.39 0.21 0.1    0
```

```
library(grid)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
loadings$vars <- ifelse(rownames(loadings) %in% "restriction_intl_travel","International Travel",
                        ifelse(rownames(loadings) %in% "restriction_most_business","Most Businesses",
                               ifelse(rownames(loadings) %in% "restriction_large_gatherings","Large Gatl
                                      ifelse(rownames(loadings) %in% "restriction_sporting_events","Spo
                                             ifelse(rownames(loadings) %in% "restriction_closing_k12","
                                                    ifelse(rownames(loadings) %in% "restriction_carry_o
                                                           ifelse(rownames(loadings) %in% "restriction_
                                                                  ifelse(rownames(loadings) %in% "libcor


#plotting factor analysis plot
plot <- ggplot(loadings,aes(x = f1, y=f2,label=vars)) +
  theme_minimal() +
  geom_label_repel() +
  scale_x_continuous("First Dimension Factor (Proportion of Variance: 36%)") +
  scale_y_continuous("Second Dimension Factor (Propoportion of Variance 4%)") +
  geom_segment(aes(x = 0,y = 0,xend = f1,yend = f2),arrow=arrow())

grid.newpage()
```

```
footnote <- expression("Cronbach's Standardized"~alpha~"="~0.78)
g <- arrangeGrob(plot,
                 bottom = textGrob(footnote, x = 0.025, hjust = 0, vjust= 0,
                                   y=0.75, gp = gpar(fontface = "italic",
                                                     fontsize = 9, col = "black")))
grid.draw(g)
```

Cronbach's Standardized α = 0.78

```
ggsave(file="factor_analysis_covid19_policies.png", g, width = 8, height = 5.43, units = "in")

covid_ideo_scales <- cbind(covid_ideo_scales,scores)
colnames(covid_ideo_scales)[10:11] <- c("covid_restriction_fa_dim1","covid_restriction_fa_dim2")
covid_ideo_scales$summated_restriction_scale <- rowSums(covid_ideo_scales[2:8],na.rm=T)
```

Item-Response Theory Model= "The item response theory (IRT), also known as the latent response theory refers to a family of mathematical models that attempt to explain the relationship between latent traits (unobservable characteristic or attribute) and their manifestations (i.e. observed outcomes, responses or performance). They establish a link between the properties of items on an instrument, individuals responding to these items and the underlying trait being measured. IRT assumes that the latent construct (e.g. stress, knowledge, attitudes) and items of a measure are organized in an unobservable continuum. Therefore, its main purpose focuses on establishing the individual's position on that continuum."

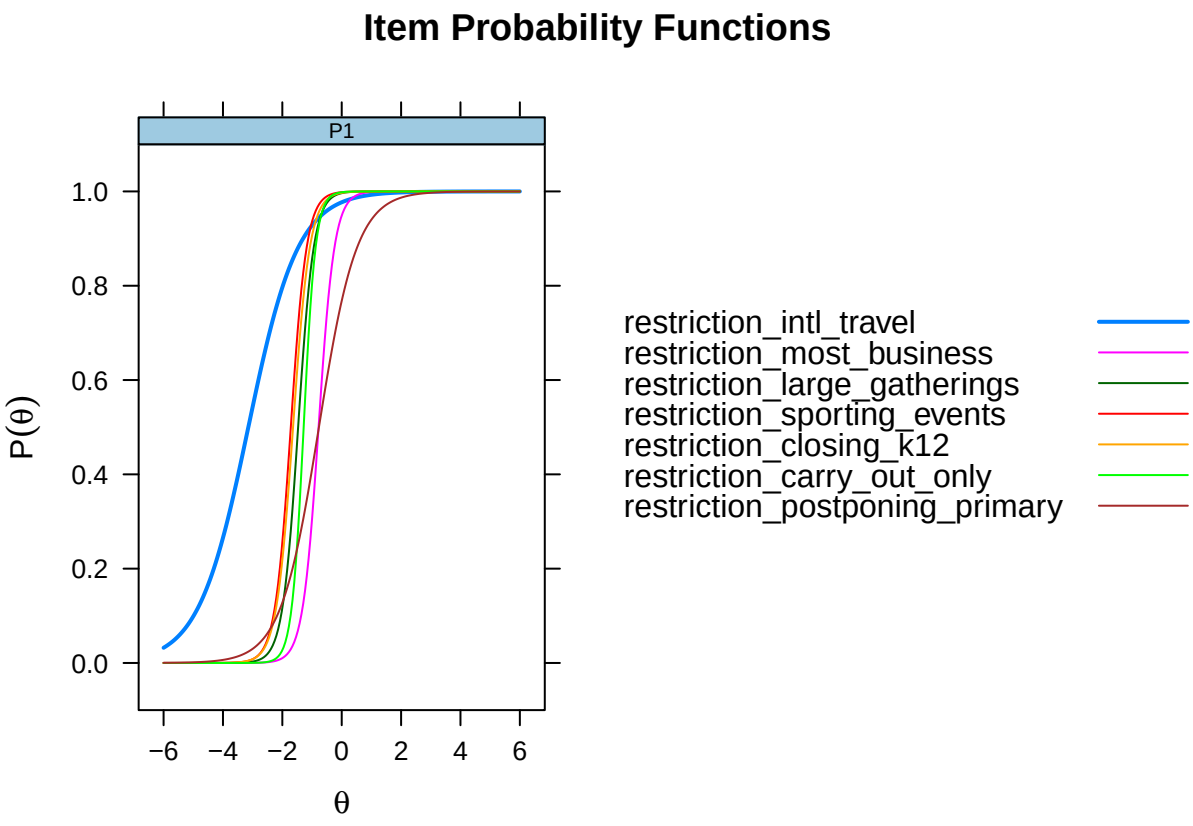
In this context, IRT is used to create a continuum with the various COVID policy questions and the aggregate score of a respondant's answers to the questions will proxy as an overall measure of support (which is a latent variable) for social distancing policies

NOTE: this whole chunk works but I want to know the more granular mechanics of how/why, but leave that to me

```
irt <- mirt(covid_ideo_scales[2:8], model = 1, itemtype = "graded", SE = T,
            verbose = T,removeEmptyRows = TRUE)
```

```
## Iteration: 1, Log-Lik: -24114.323, Max-Change: 1.49614Iteration: 2, Log-Lik: -21899.147, Max-Change:
##
## Calculating information matrix...
```

```
plt <- plot(irt, type = 'trace', facet_items=F) #store the object
print(plt) #plot the object
```



```
#str(plt) #find the data
pltdata <- data.frame(lapply(plt$panel.args, function(x) do.call(cbind, x))[[1]])

groups <- plot(irt, type = 'trace', facet_items=T)
#groups$packet.sizes
pltdata$item <- rep(colnames(covid_ideo_scales)[2:8], each = 200)
pltdata$response <- groups$panel.args.common$groups

#plt$panel.args.common$groups

pltdata$item2 <- factor(pltdata$item,levels=c("restriction_carry_out_only",
                                              "restriction_closing_k12",
                                              "restriction_intl_travel",
                                              "restriction_large_gatherings",
                                              "restriction_most_business",
                                              "restriction_postponing_primary",
                                              "restriction_sporting_events"),
                        labels=c("Carry Out Only","Close K-12 Schools",
                                 "Restrict Intl. Travel","Restrict Large Gatherings",
                                 "Restrict Most Businesses","Postpone Primary Elections",
                                 "Restrict Sporting Events"))

pltdata$item2 <- factor(pltdata$item2,levels=c("Restrict Intl. Travel",
                                               "Restrict Sporting Events",
                                               "Close K-12 Schools",
                                               "Restrict Large Gatherings",
                                               "Carry Out Only","Restrict Most Businesses",
                                               "Postpone Primary Elections"))

plot <- ggplot(pltdata, aes(x, y,linetype=item2,color=item2)) +
  geom_line() +
  scale_x_continuous(expression(theta)) +
  scale_y_continuous("Pr (Support)") +
  geom_hline(aes(yintercept = 0.5))  +
  theme_minimal() +
  labs(color="Policy",linetype="Policy") +
  theme(legend.position="bottom")
#+ scale_colour_grey(start = 0, end = .5) +
#  ggtitle("Ordinal IRT Model Characteric Curves for Emphatic Racism Scale")

#ggsave(file="covid19_restrictions_irt_curves_probs.png", plot, width = 8, height = 5.43, units = "in")
```

Scores= the scores generated from the IRT continuum for each of the respondants (this is the ultimate support for social distancing policy that serves as the DV)

whole thing runs!

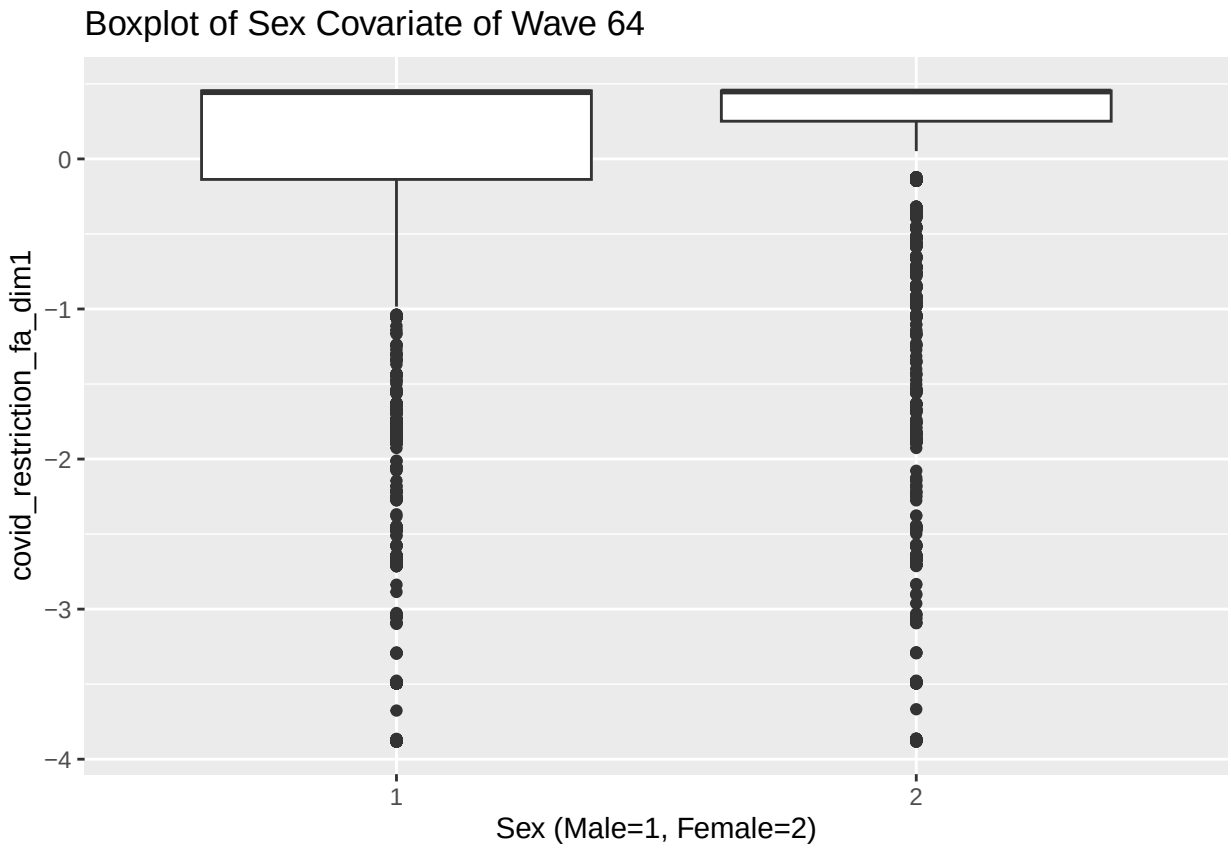
```
#storing scores from IRT analysis for each respondent in original covid scales data
covid_ideo_scales$covid_restriction_fa_dim2 <- fscores(irt, full.scores = TRUE, full.scores.SE = F)
colnames(covid_ideo_scales)[11] <- "covid_restriction_irt"

# colnames(covid_ideo_scales)
# colnames(pew_ATP_W64)
```

```
#merging with original data set
pew_ATP_W64 <- merge(pew_ATP_W64,covid_ideo_scales,by=c("QKEY"),all=T)
pew_ATP_W64$covid_restriction_fa_dim1 <- pew_ATP_W64$covid_restriction_fa_dim1- mean(pew_ATP_W64$covid_
```

Covariates

whole thing runs!

```
#transforming gender covariate & plotting
pew_ATP_W64$female <- as.character(pew_ATP_W64$F_SEX)
pew_ATP_W64$female <- recode(pew_ATP_W64$female, "1" = "Male", "2"= "Female")
pew_ATP_W64$female <- factor(pew_ATP_W64$female,levels=c("Male","Female"))

df_64 <- as.data.frame(cbind(pew_ATP_W64$female, pew_ATP_W64$covid_restriction_fa_dim1))

ggplot(data= na.omit(df_64),
       aes(x=as.factor(V1),y= V2)) + geom_boxplot() +
  labs(title = "Boxplot of Sex Covariate of Wave 64",
       x= "Sex (Male=1, Female=2)",
       y= "covid_restriction_fa_dim1")
```




```
#transforming political identification & using to create weights
pew_ATP_W64$pid3 <- as.character(pew_ATP_W64$F_PARTY_FINAL)
pew_ATP_W64$pid3 <- factor(pew_ATP_W64$pid3,levels=c("Republican","Independent","Democrat"))

pew_ATP_W64$weight <- as.numeric(pew_ATP_W64$WEIGHT_W64)
```

```
#transforming & cleaning age covariate
pew_ATP_W64$age_linear <- as.numeric(factor(pew_ATP_W64$F_AGECAT))
pew_ATP_W64$age_linear[pew_ATP_W64$age_linear %in% 5] <- NA # Get rid of refused

#transforming & cleaning education covariate
pew_ATP_W64$educ_linear <- as.numeric(factor(pew_ATP_W64$F_EDUCCAT2))
pew_ATP_W64$educ_linear[pew_ATP_W64$educ_linear %in% 7] <- NA # Get rid of refused

#transforming & cleaning marital covariate
pew_ATP_W64$marital_status <- ifelse(pew_ATP_W64$F_MARITAL %in% "Married",1,0)
pew_ATP_W64$marital_status[pew_ATP_W64$F_MARITAL %in% "Refused"] <- NA # Get rid of refused

#transforming & cleaning income covariate
pew_ATP_W64$income_linear <- as.numeric(factor(pew_ATP_W64$F_INCOME))
pew_ATP_W64$income_linear[pew_ATP_W64$income_linear %in% 10] <- NA # Get rid of refused

#transforming & cleaning region & white race covariates
pew_ATP_W64$region_factor <- pew_ATP_W64$F_CREGION
pew_ATP_W64$white_respondent <- ifelse(pew_ATP_W64$F_RACETHN %in% "White non-Hispanic",1,0)
pew_ATP_W64$white_respondent[pew_ATP_W64$F_RACETHN %in% "Refused"] <- NA # Get rid of refused

#getting other race covariate
hold <- read_dta("ATP_W42.dta")
hold <- hold[,c("QKEY", "F_RACECMB")]

pew_ATP_W64 <- merge(pew_ATP_W64,hold,by=c("QKEY"),all=T)

pew_ATP_W64$race3 <- ifelse(pew_ATP_W64$F_RACETHN %in% "White non-Hispanic","white",ifelse(pew_ATP_W64$F
pew_ATP_W64$race3 <- ifelse(is.na(pew_ATP_W64$race3) & pew_ATP_W64$F_RACECMB %in% "Asian or Asian-Americ

pew_ATP_W64$race3 <- factor(pew_ATP_W64$race3,levels=c("white","black","hispanic","asian"))
```

Trust

```
#loading trust data from wave 42
trust <- read_dta("ATP_W42.dta")
trust <- subset(trust,select=c("QKEY","CONFa_W42","CONFb_W42","CONFd_F1_W42",
                               "CONFd_F2_W42", "POLICY1_W42","POLICY2_W42",
                               "POLICY3_W42","SCM2_W42","SCM3_W42","SCM4a_W42",
                               "SCM4b_W42"))

colnames(trust) <- c("QKEY","trust_elected_officials","trust_media",
                     "trust_medial_scientists","trust_scientists",
                     "scientists_active_role_policy","scientists_pivotal_policy",
                     "scientists_better_policy","scientific_method",
                     "scientists_judgement_facts","research_essential_immediate_applications",
                     "research_essential_advance_knowledge")

#omitting non-responses # this doesn't work because it's already coded as NA, but "refused"
#trust[trust == "Refused"] <- NA

#adding levels
for(i in 2:ncol(trust)){
```

```
  print(table(trust[,i]))
  print(levels(trust[,i]))
}
for(i in 2:ncol(trust)){
  trust[,i] <- as.character(trust[,i])
  print(table(trust[,i]))
}

trust$research_essential_immediate_applications
#recoding responses to each trust survey question
trust[trust == "Essential"] <- 4
trust[trust == "Important, but not essential"] <- 3
trust[trust == "Not too important"] <- 2
trust[trust == "Not important at all"] <- 1

trust[trust == "A great deal of confidence"] <- 4
trust[trust == "A fair amount of confidence"] <- 3
trust[trust == "Not too much confidence"] <- 2
trust[trust == "No confidence at all"] <- 1

trust[trust == "Scientists should take an active role in public policy debates about scientific issues"]
trust[trust == "Scientists should focus on establishing sound scientific facts and stay out of public po

trust[trust == "Public opinion should NOT play an important role to guide policy decisions about scienti
trust[trust == "Public opinion should play an important role to guide policy decisions about scientific

trust[trust == "Usually BETTER at making good policy decisions about scientific issues than other people
trust[trust == "NEITHER BETTER NOR WORSE at making good policy decisions about scientific issues than ot
trust[trust == "Usually WORSE at making good policy decisions about scientific issues than other people

trust[trust == "The scientific method generally produces accurate conclusions"] <- 2
trust[trust == "The scientific method can be used to produce any conclusion the researcher wants"] <- 1

trust[trust == "Scientists make judgments based solely on the facts"] <- 2
trust[trust == "Scientists' judgments are just as likely to be biased as other people's"] <- 1

#indicating incomplete observations as NA
trust$trust_scientists <- ifelse(is.na(trust$trust_medial_scientists),trust$trust_scientists,
                                 ifelse(is.na(trust$trust_scientists),trust$trust_medial_scientists,NA))

trust$trust_medial_scientists <- NULL
trust$scientists_pivotal_policy <- NULL

for(i in 2:ncol(trust)){
  trust[,i] <- as.numeric(trust[,i])
  print(table(trust[,i]))
}

factanal <- fa(trust[,c(4:10)], nfactors=2, rotate="promax", fm="pa")
scores <- data.frame(factanal$scores)
loadings(factanal)
loadings <- factanal$loadings
loadings <- data.frame(f1 = loadings[,1],f2=loadings[,2])
```

```
plot(factanal$loadings,type="n") # set up plot
text(factanal$loadings,labels=names(trust)[4:10],cex=.7) # add variable names

library(ggrepel)

alpha(trust[,c(4:8)])
skewness(loadings$f1, na.rm=T)
kurtosis(loadings$f1, na.rm=T)

alpha(trust[,c(9:10)])
skewness(loadings$f2, na.rm=T)
kurtosis(loadings$f2, na.rm=T)

loadings$vars <- ifelse(rownames(loadings) %in% "trust_scientists","Trust Scientists",ifelse(rownames(l

plot <- ggplot(loadings,aes(x = f1, y=f2,label=vars)) + theme_minimal() + geom_label_repel() + scale_x_
grid.newpage()
footnote <- expression("Cronbach's Standardized"~alpha~"="~0.66)
g <- arrangeGrob(plot, bottom = textGrob(footnote, x = 0.025, hjust = 0, vjust= 0, y=0.75, gp = gpar(fo
grid.draw(g)
#ggsave(file="factor_analysis_scientific_trust.png", g, width = 8, height = 5.43, units = "in")

trust <- cbind(trust,scores)
colnames(trust)[11:12] <- c("trust_scientists_fa_dim1","trust_scientists_fa_dim2")

pew_ATP_W64 <- merge(pew_ATP_W64,trust,by=c("QKEY"),all=T)

x <- subset(pew_ATP_W64,select=c(pid3,trust_scientists_fa_dim1,trust_scientists_fa_dim2))
x <- na.omit(x)
x$pid <- ifelse(x$pid3 %in% "Democrat","D",ifelse(x$pid3 %in% "Republican","R",ifelse(x$pid3 %in% "Indep

plot <- ggplot(x,aes(x=trust_scientists_fa_dim1,y=trust_scientists_fa_dim2,label=pid,color=pid)) + geom_

x1 <- x
x1$pid3 <- "Full Sample"
x <- rbind(x,x1)
x$pid3 <- factor(x$pid3,levels=c("Republican","Independent","Democrat","Full Sample"),labels=c("Republi

print(summary(aov(trust_scientists_fa_dim1 ~ pid3, data = x)))

plot <- ggplot(x, aes(x=pid3,y=trust_scientists_fa_dim1, group=pid3,fill=pid3)) + geom_boxplot(alpha=0.
#ggsave(file="scientific_trust_boxplots_by_party.png", plot, width = 8, height = 5.43, units = "in")
```

Data Analysis: COVID Policy ~ Scientific Trust: Baseline Effects

Baseline Trust Effects

```
baseline_trust_effects <- list()
for(i in which(colnames(pew) == "restriction_intl_travel"):which(colnames(pew) == "restriction_postponi
  summary(model <- glm(pew[,i] ~ trust_scientists_fa_dim1 + trust_media + trust_elected_officials + fem
  mes <- summary(margins(model, variables=c("trust_scientists_fa_dim1","trust_media","trust_elected_off
  mes$model <- colnames(pew)[i]
```

```
  mes$category <- "Full Sample Baseline"
  baseline_trust_effects[[i]] <- mes
}
baseline_trust_effects <- ldply(baseline_trust_effects,data.frame)

baseline_trust_effects$pid3 <- "Full Baseline Sample"
baseline_trust_effects$category <- NULL

effects <- baseline_trust_effects
effects$ylo90 <- (effects$AME - (qt(.95, 100) * effects$SE))
effects$yhi90 <- (effects$AME + (qt(.95, 100) * effects$SE))
effects$model2 <- ifelse(effects$model %in% "restriction_carry_out_only","Dependent Variable: Restauran
                         ifelse(effects$model %in% "restriction_closing_k12","Dependent Variable: Close
                                ifelse(effects$model %in% "restriction_intl_travel","Dependent Variable
                                       ifelse(effects$model %in% "restriction_large_gatherings","Depend
                                              ifelse(effects$model %in% "restriction_most_business","De
                                                     ifelse(effects$model %in% "restriction_postponing_
                                                            ifelse(effects$model %in% "restriction_spor

effects$factor <- ifelse(effects$factor %in% "trust_elected_officials","Elected Officials Trust",
                         ifelse(effects$factor %in% "trust_media","Media Trust",
                                ifelse(effects$factor %in% "trust_scientists_fa_dim1","Scientific Trust

effects$label <- ifelse(effects$p < 0.01,paste(round(effects$AME,2),"***",sep=""),
                        ifelse(effects$p < 0.05,paste(round(effects$AME,2),"**",sep=""),
                               ifelse(effects$p < 0.10,paste(round(effects$AME,2),"*",sep=""),NA)))


for(i in unique(effects$model)){
  x <- subset(effects,effects$model %in% i)
  plot <- ggplot(x,aes(x=factor,y=AME,factor=factor,group=factor,color=factor,shape=factor,label=label,
    facet_wrap(~model2) + coord_flip() +
    geom_linerange(aes(x= factor, ymin = ylo90, ymax = yhi90), position = position_dodge(width=0.75), lw
    geom_pointrange(aes(x= factor, ymin = lower, ymax = upper), lwd = 1/2, position = position_dodge(wi
    theme_minimal() + scale_x_discrete("") + scale_y_continuous("Min/Max First Difference Marginal Effe
    geom_hline(yintercept = 0, colour = gray(1/2), lty = 2) +
    labs(color="Trust Effects",shape="Trust Effects") +
    theme(legend.position = "none") +
    theme(axis.text.x = element_text(hjust = 0.5),axis.text.y = element_text(hjust = 0.5)) +
    geom_label(vjust=-0.5,hjust=0.25,fill="white") + labs(caption="* p < 0.1; ** p < 0.05; *** p < 0.01
    scale_shape_manual("",values=c(23,22,21))
  print(plot)
  #ggsave(file=paste(i,"_model",".png",sep=""), plot, width = 8, height = 5.43, units = "in")
}
```

Distribution of Summated Rating Scales

```
x <- subset(pew,select=c(summated_restriction_scale,trust_scientists_fa_dim1,pid3,trust_media,trust_ele
x1 <- na.omit(x)
x <- na.omit(x)
x1$race3 <- "Full Sample"

x <- subset(x,select=c(summated_restriction_scale,race3))
```

```
x$n <- 1
xs <- ddply(x,.(summated_restriction_scale,race3),summarise,total=sum(n,na.rm=T))
x <- ddply(x,.(race3),summarise,total_race3=sum(n,na.rm=T))
xs <- merge(xs,x,by=c("race3"))
xs$prop <- xs$total/xs$total_race3

x1 <- subset(x1,select=c(summated_restriction_scale,race3))
x1$n <- 1
xs1 <- ddply(x1,.(summated_restriction_scale),summarise,total=sum(n,na.rm=T))
xs1$total_race3 <- sum(x1$n,na.rm=T)
xs1$prop <- xs1$total/xs1$total_race3
xs1$race3 <- "Full Sample"

x <- rbind(xs,xs1)
x$race3 <- factor(x$race3,levels=c("asian","black","hispanic","white","Full Sample"),labels=c("Asian Res

plot <- ggplot(x, aes(x=factor(summated_restriction_scale), y=prop, label=round(prop,2))) + geom_point(s
#ggsave(file="number_policies_dotplot.png", plot, width = 8, height = 5.43, units = "in")
```

Conditioned by Race

Data Analysis: COVID Policy ~ Scientific Trust: Baseline Effects

Baseline Trust Effects

```
baseline_trust_effects.race <- list()
for(i in which(colnames(pew) == "restriction_intl_travel"):which(colnames(pew) == "restriction_postponi
  summary(model <- glm(pew[,i] ~ trust_scientists_fa_dim1*race3 + trust_media + trust_elected_officials
  mes <- summary(margins(model, variables=c("trust_scientists_fa_dim1","trust_media","trust_elected_off
  mes$model <- colnames(pew)[i]
  mes$category <- "Full Sample Baseline"
  baseline_trust_effects.race[[i]] <- mes
}
baseline_trust_effects.race <- ldply(baseline_trust_effects.race,data.frame)


effects <- baseline_trust_effects.race
effects$ylo90 <- (effects$AME - (qt(.95, 100) * effects$SE))
effects$yhi90 <- (effects$AME + (qt(.95, 100) * effects$SE))
effects$model2 <- ifelse(effects$model %in% "restriction_carry_out_only","Dependent Variable: Restauran
effects$factor <- ifelse(effects$factor %in% "trust_elected_officials","Elected Officials Trust",ifelse
effects$race3 <- factor(effects$race3,levels=c("asian","black","white","hispanic"), labels=c("Asian Resp

for(i in unique(effects$model)){
  x <- subset(effects,effects$model %in% i)
  plot <- ggplot(x,aes(x=race3,y=AME,factor=factor,group=factor,color=factor,shape=factor)) + facet_wrap
  print(plot)
  #ggsave(file=paste(i,"_race3_model",".png",sep=""), plot, width = 8, height = 5.43, units = "in")
}
```

Data Analysis Figures: OLS Composite Models

```
summary(model <- glm(covid_restriction_irt ~ trust_scientists_fa_dim1*race3 + trust_media*race3 + trust_
baseline_trust_effects.4 <- summary(margins(model, variables=c("trust_scientists_fa_dim1","trust_media"
baseline_trust_effects.4$model <- "DV: Latent Policy Scale"
baseline_trust_effects.4$pid3 <- "Full Sample"

effects <- baseline_trust_effects.4
effects$race3 <- factor(effects$race3,levels=c("asian","black","white","hispanic"), labels=c("Asian Resp

effects$ylo90 <- (effects$AME - (qt(.95, 100) * effects$SE))
effects$yhi90 <- (effects$AME + (qt(.95, 100) * effects$SE))

effects$factor <- ifelse(effects$factor %in% "trust_elected_officials","Elected Officials Trust",ifelse

plot <- ggplot(effects,aes(x=race3,y=AME,factor=factor,group=factor,color=factor,shape=factor)) + facet_
print(plot)
#ggsave(file="latent_policy_scale_race3_model.png", plot, width = 8, height = 5.43, units = "in")

summary(model <- glm(summated_restriction_scale ~ trust_scientists_fa_dim1*race3 + trust_media*race3 + t
baseline_trust_effects.5 <- summary(margins(model, variables=c("trust_scientists_fa_dim1","trust_media"
baseline_trust_effects.5$model <- "DV: Summated Policy Scale"
baseline_trust_effects.5$pid3 <- "Full Sample"

effects <- baseline_trust_effects.5
effects$race3 <- factor(effects$race3,levels=c("asian","black","white","hispanic"), labels=c("Asian Resp

effects$ylo90 <- (effects$AME - (qt(.95, 100) * effects$SE))
effects$yhi90 <- (effects$AME + (qt(.95, 100) * effects$SE))

effects$factor <- ifelse(effects$factor %in% "trust_elected_officials","Elected Officials Trust",ifelse

plot <- ggplot(effects,aes(x=race3,y=AME,factor=factor,group=factor,color=factor,shape=factor)) + facet_
print(plot)
#ggsave(file="summated_policy_scale_race3_model.png", plot, width = 8, height = 5.43, units = "in")
```

Boxplot by Race

```
x <- subset(pew,select=c(race3,trust_scientists_fa_dim1))
x <- na.omit(x)
print(summary(aov(trust_scientists_fa_dim1 ~ race3, data = x)))
y <- subset(pew,select=c(race3,trust_scientists_fa_dim1))
y$race3 <- "Full Sample"
x <- rbind(x,y)

plot <- ggplot(x, aes(x=race3,y=trust_scientists_fa_dim1, group=race3,fill=race3)) + geom_boxplot(alpha=
#ggsave(file="scientific_trust_boxplots_by_race3.png", plot, width = 8, height = 5.43, units = "in")
```

Data Analysis Figures: OLS Composite Models

```
summary(model <- lm(trust_scientists_fa_dim1 ~ female + pid3 + libcon + age_linear + educ_linear + incom
```

```
mes <- summary(margins(model, type="response", change="minmax"))
mes$race3 <- factor(mes$factor,levels=c("race3asian","race3black","race3hispanic"),labels=c("Asian Respo

mes$label <- ifelse(mes$p < 0.01,paste(round(mes$AME,2),"***",sep=""),ifelse(mes$p < 0.05,paste(round(m

mes$ylo90 <- (mes$AME - (qt(.95, 100) * mes$SE))
mes$yhi90 <- (mes$AME + (qt(.95, 100) * mes$SE))
mes$model <- "DV: Latent Scientific Trust "

plot <- ggplot(subset(mes,!is.na(mes$race3)),aes(x=race3,y=AME,factor=race3,group=race3,color=race3,shap
ggsave(file="latent_scientific_trust_model.png", plot, width = 8, height = 5.43, units = "in")

mes <- subset(mes,!(mes$factor %in% c("region_factorWest","region_factorSouth","region_factorMidwest")))

plot <- ggplot(mes,aes(x=factor,y=AME,factor=factor,group=factor,label=label)) + facet_wrap(~model) + co
ggsave(file="latent_scientific_trust_model_full.png", plot, width = 8, height = 5.43, units = "in")
```