

First Analysis 4-30

Alicia

4/30/2021

Step 1: Import & clean data

Click to expand and see the initial data import and cleaning steps & full table from the data extraction form.

```
library(tidyverse)
library(metafor)
library(clubSandwich)
library(robumeta)
library(kableExtra)
library(table1)
```

Import data, making sure everything is the correct data type.

```
all_data = read_csv("Data/FirstPassData4-30.csv",
                    # up to outcome + up to exposure_reliability +
                    col_types = paste0("nncccccfffcfffcfffcncccc", "ffffffffffffllclllcllfc", "fcccc"))
```

Fill in information in columns that are aggregates of others.

```
domestic_perpetrator_strings = c("husband",
                                "Family violence",
                                "family",
                                "Intimate partner",
                                "IPV")

non_domestic_known_perpetrator_strings = c("known nonintimate",
                                             "acquaintance",
                                             "friend")

all_data = all_data %>%
  group_by(Study_ID) %>%
  mutate(Combined_survey_surveillance = paste0(unique(Survey_surveillance), collapse = " & "),
         Combined_outcome_data_source = paste0(unique(Outcome_data_source), collapse = " & "),
         Combined_exposure_level = paste0(unique(Exposure_level), collapse = " & "),
         Combined_outcome_type_factor = paste0(unique(Outcome_type_factor), collapse = " & "),
         Num_EMs_from_this_study = length(unique(Effect_ID))
  ) %>%
  ungroup() %>%
```

```

mutate(Exposure_category_economic = Exposure_category_employment | Exposure_category_financial,
       Exposure_combo = (as.integer(!is.na(Exposure_labor_force_participation)) +
                        as.integer(!is.na(Exposure_employment_unemployment)) +
                        as.integer(!is.na(Exposure_income)) +
                        as.integer(!is.na(Exposure_management_professional_roles)) +
                        as.integer(!is.na(Exposure_women_owned_businesses)) +
                        as.integer(!is.na(Exposure_education)) +
                        as.integer(!is.na(Exposure_poverty)) +
                        as.integer(!is.na(Exposure_representation_in_government)) +
                        as.integer(!is.na(Exposure_voter_registration_civic_participation)) +
                        as.integer(!is.na(Exposure_maternal_mortality)) +
                        as.integer(!is.na(Exposure_teen_pregnancy)) +
                        as.integer(!is.na(Exposure_other_health_outcomes)) +
                        as.integer(!is.na(Exposure_women_legal_rights))) > 1,
       Unadjusted = Confounders == "NONE (UNADJUSTED)",
       Victimization_type = as.factor(
         case_when(Outcome_type_factor %in% c("Intimate partner homicide",
                                              "Female homicide") ~ "Homicide",
                   Outcome_type_factor %in% c("Rape",
                                              "Sexual assault") ~ "Rape & sexual assault",
                   Outcome_type_factor %in% c("Assault",
                                              "IPV",
                                              "Family violence",
                                              "Harassment") ~ "IPV, harassment, & assault",
                   TRUE ~ "999")),
       Perpetrator_relationship = as.factor(
         case_when(str_detect(Outcome, paste0(domestic_perpetrator_strings, collapse = "|")) ~ "Domestic",
                   str_detect(Outcome, paste0(non_domestic_known_perpetrator_strings, collapse = "|")) ~ "Non-domestic known",
                   TRUE ~ "Unknown or not specified")),
       Exposure_level_condensed = as.factor(
         case_when(Exposure_level %in% c("City", "MSA") ~ "City/MSA",
                   Exposure_level %in% c("County") ~ "County",
                   Exposure_level %in% c("State") ~ "State",
                   TRUE ~ "999")),
       Race = as.factor(
         case_when(str_detect(Outcome, "White") ~ "White",
                   str_detect(Outcome, "Black") ~ "Black",
                   TRUE ~ "999"))
) %>%
group_by(Study_ID) %>%
mutate(Combined_exposure_types = paste0(c(rep("Economic", min(sum(Exposure_category_economic, na.rm = TRUE))),
rep("Education", min(sum(Exposure_category_education, na.rm = TRUE))),
rep("Political representation", min(sum(Exposure_category_representation, na.rm = TRUE))),
rep("Legal rights", min(sum(Exposure_category_legal_rights, na.rm = TRUE))),
rep("Health indicators", min(sum(Exposure_category_womens_health_indicators, na.rm = TRUE))),
collapse = " & "))
) %>%
ungroup() %>%
mutate_if(is.logical, ~replace(., is.na(.), FALSE)) %>%
mutate(Decade = fct_relevel(Decade, c("1970s", "1980s", "1990s", "2000s", "2010s")),
       Survey_surveillance = fct_relevel(Survey_surveillance, c("Survey", "Surveillance")),
       Study_design = fct_relevel(Study_design, c("Cross-sectional", "Serial cross-sectional", "Longitudinal")),
       Outcome_data_source_condensed = fct_relevel(Outcome_data_source_condensed, c("Survey", "Hospital", "Clinic", "Community", "Other")))

```

```

    Exposure_level = fct_relevel(Exposure_level, c("State", "City", "MSA", "Neighborhood", "County", "Other"))
    Ecological_cross_level = ifelse(Study_ID == 17, "Unable to determine", Ecological_cross_level)
    Ecological_cross_level = fct_recode(Ecological_cross_level, "Ecological" = "1", "Cross-level" = "2", "Unable to determine" = "3")
    Ecological_cross_level = fct_relevel(Ecological_cross_level, c("Ecological", "Cross-level", "Unable to determine"))
  )

```

```
summary(filter(all_data, Ecological_cross_level == "Ecological")$Sample_size)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      25.0   50.0   148.0   215.5   177.0   3083.0
```

```
summary(filter(all_data, Ecological_cross_level == "Cross-level")$Sample_size)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      610   1414   3916   83665   5390   487166
```

```
summary(filter(all_data, Ecological_cross_level == "Unable to determine")$Sample_size)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       50 1449494 2898939 1932643 2898939 2898939
```

TODOs & options moving forward:

- Double-check the accuracy of the `Outcome` & `Outcome_type_factor` columns (esp. info about victim sex, if we want to use that)
- Are the `Outcome` categories what make the most sense for analysis?
- Are the `Exposure` categories what make the most sense for analysis?
- Do we want to do anything quantitative with `Confounders`?
- Anything else that would make sense as a predictor that we haven't captured?
 - Maybe: location (South / not South? Within 1 state vs. across many?)
 - Maybe: exposure type

Step 2: Describe the data

Click to setup code.

```

# **Summary stats, study-level:**

study_level = all_data %>%
  distinct(Study_ID, .keep_all = TRUE)

# table1(~ as.numeric(PubYear) +
#         Combined_outcome_data_source + Combined_outcome_type_factor +
#         Combined_exposure_level + Combined_exposure_types + Exposure_combo +
#         Unadjusted + Model_type + Num_EMs_from_this_study, data = study_level)

```

Summary stats, EM-level:

```

em_level = all_data %>%
  mutate(Included_in_meta_regression = ifelse(Include_in_regression %in% c("Yes", "Maybe - can we treat
    Included_in_meta_regression = factor(Included_in_meta_regression, levels=c(1, 0), labels=c("In

label(em_level$Decade) <- "Decade of majority of datapoints"
label(em_level$Exposure_level) <- "Exposure: level of analysis"
label(em_level$Survey_surveillance) <- "Outcome: survey or surveillance?"
label(em_level$Outcome_data_source_condensed) <- "Outcome: data source"
label(em_level$Victimization_type) <- "Outcome: victimization type"
label(em_level$Perpetrator_relationship) <- "Outcome: victim/offender relationship"

# Table with more variables, that didn't make it into the thesis table 1
table1(~ Study_design + Ecological_cross_level + Exposure_level + Exposure_combo + Exposure_category_emp
  Exposure_category_political + Exposure_category_legal_rights + Exposure_category_womens_health
  Exposure_absolute_or_relative + Outcome_type_factor +
  Sample_size + Unadjusted | Included_in_meta_regression, data = em_level)

```

```
## [1] "<table class=\"Rtable1\">\n<thead>\n<tr>\n<th class='rowlabel firstrow lastrow'></th>\n<th class="

```

```

#Table 1, for thesis
table1(~ Decade + Exposure_level + Survey_surveillance + Outcome_data_source_condensed +
  Victimization_type + Perpetrator_relationship | Included_in_meta_regression,
  data = em_level,
  overall = FALSE
)

```

```
## [1] "<table class=\"Rtable1\">\n<thead>\n<tr>\n<th class='rowlabel firstrow lastrow'></th>\n<th class="

```

Click to see Table listing all studies

```

for_summary = study_level %>%
  mutate(Included_in_meta_regression = ifelse(Include_in_regression %in% c("Yes", "Maybe - can we treat
  select(Authors_abbr,
    PubYear,
    #Title,
    #Hypothesis,
    Source_population,
    Combined_survey_surveillance,
    Combined_outcome_data_source,
    Combined_outcome_type_factor,
    Combined_exposure_level,
    Combined_exposure_types,
    # NOTE: this isn't necessarily consistent per study
    Sample_size,
    Num_EMs_from_this_study,
    Included_in_meta_regression)

#kable_paper(
  kable(for_summary,
    col.names = c("Authors", "Publication Year", "Source population",
      "Outcome: survey or surveillance?", "Outcome: data source", "Outcome: victimization",
      "Exposure: level of analysis", "Exposure: gender inequality types",
      "Sample size", "Number of effects contributed", "Included in meta-regression?"))#

```

Authors	Publication Year	Source population
Aizer	2010	California counties
Avakame	1998	All U.S. (surveillance)
Bailey	1999	Cities (population > 100,000)
Baron & Straus	1989	All U.S. (surveillance)
Boyle et al.	2017	Students at 413 "top" colleges in 47 states
Brewer & Smith	1995	Cities (population > 250,000)
Campbell et al.	2019	All U.S. (surveillance)
D'Alessio & Stolzenberg	2010	Cities (population > 50,000)
DeWees & Parker	2003	Cities (population > 100,000)
Dugan et al.	1999	Cities (in the 25 largest MSAs)
Ellis & Beattie	1983	Large cities
Eschholz & Vieraitis	2004	Cities (population > 100,000 & Black residents > 2,000)
Gillespie & Reckdenwald	2015	North Carolina counties
Goodson & Bouffard	2019	Texas counties
Gressard et al.	2015	Adolescents - All U.S. (nationally representative survey)
Haynie & Armstrong	2006	Cities (population > 100,000 & Black residents > 5,000)
Henke & Lin-chi	2020	All U.S. (surveillance)
Jackson	2016	Chicago neighborhoods (female primary caregivers of children, married or c
Johnson	2013	Kansas counties
Kawachi et al.	1999	All U.S. (surveillance)
Kearns et al.	2020	All U.S. (nationally representative survey)
Lee et al.	2009	Large MSAs (75 of the 100 largest)
Martin et al.	2006	Cities (population > 100,000)
Okeke et al.	2019	Adolescents - All U.S. (nationally representative survey)
Pazzani	2007	All U.S. (nationally representative survey)
Peterson & Bailey	1992	Cities (population > 250,000)
Powers et al.	2018	Los Angeles neighborhoods
Pridemore & Freilich	2005	All U.S. (surveillance)
Raj et al.	2020	All employed U.S. adults (nationally representative survey)
Reckdenwald & Parker	2012	Cities (population > 100,000)
Smith & Chiricos	2003	Florida counties
Stout	1992	All U.S. (surveillance)
Straus	1994	All U.S. (nationally representative survey)
Titterington	2006	Central cities
Vieraitis & Williams	2002	Cities (population > 100,000 & Black residents > 2,000)
Vieraitis et al. (2007)	2007	All U.S. (surveillance)
Vieraitis et al. (2008)	2008	Cities (population > 100,000)
Vieraitis et al. (2015)	2015	Cities (population > 100,000)
Whaley	2001	Cities (population > 50,000)
Whaley & Messner	2002	Cities (population > 100,000)
Whaley et al.	2013	Cities (population > 100,000)
Willie & Kershaw	2019	All U.S. (nationally representative survey)
Xie et al. (2011)	2011	Largest MSAs
Xie et al. (2012)	2012	Largest MSAs
Yllö (1983)	1983	National survey (30 states, further details unavailable)
Yllö (1984)	1984	National survey (30 states, further details unavailable)

```
#      bootstrap_options = "condensed"
#)
```

Step 3: Transform effect sizes into a uniform statistic

Restrict to only the easiest measures to manage for now:

Some questions:

- DeWees & Parker use “Seemingly Unrelated Regression” - can we treat this like everything else?

```
easy_set = all_data %>%  
  filter(Include_in_regression %in% c("Yes",  
                                       "Maybe - can we treat SUR like normal?",  
                                       "Maybe - no variance with partial correlation")) %>%  
  #filter(!is.na(ES_beta) | !is.na(ES_exponentiated_beta))
```

Convert between different types of Standard Errors & Betas:

To calculate a partial correlation from `escalc`, we need: **T-test statistic, sample size, the number of predictors in the regression model.**

Therefore, we need to do some conversions of the information we have, when it doesn't conform completely:

1. *If we're given an exponentiated Beta (ex. an Odds Ratio):* take the natural log to get Beta.
2. *If we're given a Standard Error:* calculate the t-test with: $t = \text{Beta} / \text{SE}$.
3. *If we're given a 95% CI (in our case, this only happens with reported exponentiated Betas):* calculate the SE with: $(\ln(\text{CI_upper}) - \text{Beta}) / 1.96$. Then calculate the t-test using the formula from #2.
4. *If we're given a z-score:* I think this is actually just the same as a t-test statistic??
5. *If we're given a p-value:* calculate the standard error using the formula: $\text{SE} = \text{Beta} / Z$, and using $z = \text{abs}(qnorm(p))$.

- **QUESTION: does this still work if it wasn't linear regression? Are the defaults correct?**

- <https://www.statology.org/p-value-of-z-score-r/>
- <https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/Normal>
- <https://www.math.arizona.edu/~rsims/ma464/standardnormaltable.pdf>
- <https://www.socscistatistics.com/pvalues/normaldistribution.aspx>
- <https://www.bmj.com/content/343/bmj.d2090>
- https://handbook-5-1.cochrane.org/chapter_7/7_7_7_2_obtaining_standard_errors_from_confidence_intervals_and.htm

Similarly, we can calculate t-tests for correlation coefficients, and partial correlation coefficients:

6. *If we're given a Pearson's correlation coefficient:* calculate the t-test with: $t = r * \sqrt{(n - 2) / (1 - r^2)}$.

- **QUESTION: does this seem to be reasonable based on `escalc`?**
- **NOTE: p-values were given for each correlation coefficient, so can we double check?**
- <https://online.stat.psu.edu/stat501/lesson/1/1.9>
- https://en.wikipedia.org/wiki/Pearson_correlation_coefficient

7. *If we're given a partial correlation coefficient:* calculate the t-test with: $t = r * \sqrt{(n - 2 - k) / (1 - r^2)}$, where k is the number of other variables we're conditioning on.

- **NOTE: this doesn't work, because r isn't always between 0 and 1. I don't think we actually are getting “partial correlation coefficients” from Ellis & Beattie**

- **QUESTION:** *does this seem to be reasonable based on escalc?*
- <https://online.stat.psu.edu/stat505/lesson/6/6.3>

Finally, for every measure, we need to capture the direction of the association, by taking into account whether a higher reported exposure was defined to indicate more or less sexism.

```
easy_set = easy_set %>%
  mutate(
    ES_beta = ifelse(is.na(ES_beta), log(ES_exponentiated_beta), ES_beta),
    VAR_standard_error = case_when(
      !is.na(VAR_standard_error) ~ VAR_standard_error, !is.na(VAR_CI_upper) ~ (log(VAR_CI_upper) - ES_beta) / 1.96,
      TRUE ~ 999
    ),
    # There are a few options for where we get the t-test values from:
    # Just use from the study if they reported a beta & t-test:
    VAR_t_test = case_when(!is.na(ES_beta) &
      !is.na(VAR_t_test)) ~ VAR_t_test,
      # Treat any z-scores like t-test statistics:
      (!is.na(ES_beta) &
        !is.na(VAR_z_score)) ~ VAR_z_score,
      # If we have a beta and a SE (reported or calculated), we can use this simple formula:
      (!is.na(ES_beta) &
        !is.na(VAR_standard_error)) ~ ES_beta / VAR_standard_error,
      # If we have a correlation, we can use this formula along with sample size:
      (!is.na(ES_correlation)) ~ ES_correlation * sqrt((Sample_size - 2) / (1 - ES_correlation^2)),
      # If we have a partial correlation coefficient, we can use this formula:
      (!is.na(ES_partial_correlation_coefficient)) ~ ES_partial_correlation_coefficient * sqrt((Sample_size - 2) / (1 - ES_partial_correlation_coefficient^2)),
      TRUE ~ 999
    ),
    # (!is.na(ES_partial_correlation_coefficient)) ~ 5,
    TRUE ~ 999
  ),
  VAR_t_test = ifelse(
    Higher_exposure_means == "MORE sexism",
    VAR_t_test,
    VAR_t_test * -1
  )
)

# Taking a look at these newly-calculated statistics:
check = easy_set %>%
  select(
    Authors_abbr,
    ES_beta,
    ES_exponentiated_beta,
    ES_partial_correlation_coefficient,
    ES_correlation,
    VAR_CI_upper,
    VAR_z_score,
    VAR_p_value_exact,
    VAR_standard_error,
    VAR_t_test,
    Higher_exposure_means,
```

```

    Sample_size,
    Num_predictors_total
  )

```

Calculate partial correlation coefficients from EMs using SE & Beta:

Note: I had to remove the Reckdenwald study, because `escalc` gave me the error that the degrees of freedom were < 1 . This measure was weird in other ways (reported z-scores rather than t-tests or SEs), so I wouldn't be surprised if there's something we don't understand.

```

remove_reckdenwald = easy_set %>% filter(Study_ID != 30)

paste0("Number of effects, total: ", nrow(all_data))

```

```
## [1] "Number of effects, total: 235"
```

```
paste0("Number of effects left after exclusions: ", nrow(remove_reckdenwald))
```

```
## [1] "Number of effects left after exclusions: 190"
```

```

for_analysis = escalc(
  measure = "PCOR",
  data = remove_reckdenwald,
  ti = VAR_t_test,
  ni = as.numeric(Sample_size),
  mi = as.numeric(Num_predictors_total)
)

```

TODOs:

- DONE: Figure out what to do with correlations & partial correlations
- DONE: Calculate t-test from CIs, z-scores, & p-values
- Figure out what to do when only a p-value range is given
- Add in studies where we had trouble understanding the model, and convert accordingly

Step 4: Build our regression model, conduct statistical tests, create plots

Click for references, instructions, some choices we're making, & set-up code.

(Using the `metafor` and `clubSandwich` packages)

References:

1. This working paper provides a very useful explanation and guide: <https://www.jepusto.com/#working-papers>
2. This is a reference for the statistical motivation, from *Research Synthesis Methods*: <https://onlinelibrary.wiley.com/doi/abs/10.1002/jrsm.5>
3. Documentation for `clubSandwich`: <https://cran.r-project.org/web/packages/clubSandwich/clubSandwich.pdf>
4. Documentation for `metafor`: <https://wviechtb.github.io/metafor/reference/rma.mv.html>

An inventory of choices we're making to decide how to build the meta-regression model:

- What type of model?
 - Multivariate/multi-level, because some studies contributed multiple effect sizes (hence, `rma.mv`)
 - Random effects, because our studies represent a distribution of *different* true effects
- What is the correlation structure between estimates from the same study?
 - We don't know! (not enough info provided per study) → so we estimate the fixed effects component as constant, $\rho = ??$ (some references suggest 0.6, some 0.8, so we'll think about it)
 - Since we might be wrong with the assumption above, we use Robust Variance Estimation (RVE) when building our model and conducting our statistical tests
- What is the nested/hierarchical structure of our data?
 - For now, it's: `studies --> effect sizes`
 - Another option would be `"research group"/author --> studies --> effect sizes` (this is suggested when the same research group is looking at multiple different cohorts for different studies. there aren't really "research groups", although some authors frequently publish together or have contributed multiple papers to our analysis) *Note: If we want to do this, the supplementary materials here have some guidance in S3.1: <https://osf.io/nyv4u/>*

Create stratified datasets:

```
for_analysis = for_analysis %>%
  mutate(Authors_abbr = case_when(Study_ID %in% 36:38 ~ "Vieraitis et al.",
                                   Study_ID %in% 43:44 ~ "Xie et al.",
                                   Study_ID %in% 45:46 ~ "Yllö",
                                   TRUE ~ Authors_abbr),
         Nickname = ifelse(is.na(Nickname), "", paste0(":", Nickname)))

for_analysis_ALL = for_analysis

for_analysis_BLACK_WHITE = for_analysis %>%
  filter(Race != 999)

for_analysis_SURVEY = for_analysis %>%
  filter(Survey_surveillance == "Survey")

for_analysis_SURVEILLANCE = for_analysis %>%
  filter(Survey_surveillance == "Surveillance")

for_analysis_EXPOSURE_LEVEL_CONDENSED = for_analysis %>%
  filter(Exposure_level_condensed != 999)
```

Create a covariance matrix. Notes: I'm not sure if this rho is correct, and I'm not sure whether I need to impute a separate covariance matrix for each subsetted dataset, but I'm doing it just in case for now.

```
# constant sampling correlation assumption
rho <- 0.6 #is this a good assumption for us? In other places it's 0.8....

# create a covariance matrix assuming constant sampling correlation (working model)
cov_matrix_ALL <- impute_covariance_matrix(
  vi = for_analysis_ALL$vi, #vector of variances
  cluster = for_analysis_ALL$Study_ID,
  r = rho, #assuming constant. make sure we pick the right value.
  smooth_vi = FALSE #this is the default. how would we know if we want to smooth?
```

```

    #subgroup = for_analysis$authors, #i *think* this is where we'd do heirarchical clustering. but not s
)

cov_matrix_BLACK_WHITE <- impute_covariance_matrix(
  vi = for_analysis_BLACK_WHITE$vi, cluster = for_analysis_BLACK_WHITE$Study_ID, r = rho, smooth_vi = F
)

cov_matrix_SURVEY <- impute_covariance_matrix(
  vi = for_analysis_SURVEY$vi, cluster = for_analysis_SURVEY$Study_ID, r = rho, smooth_vi = FALSE
)

cov_matrix_SURVEILLANCE <- impute_covariance_matrix(
  vi = for_analysis_SURVEILLANCE$vi, cluster = for_analysis_SURVEILLANCE$Study_ID, r = rho, smooth_vi =
)

cov_matrix_EXPOSURE_LEVEL_CONDENSED <- impute_covariance_matrix(
  vi = for_analysis_EXPOSURE_LEVEL_CONDENSED$vi, cluster = for_analysis_EXPOSURE_LEVEL_CONDENSED$Study_ID
)

```

Fit models with no moderators, to estimate the summary statistic and for use in funnel and forest plots:

```

no_moderators_ALL <- rma.mv(yi ~ 1,
  V = cov_matrix_ALL, random = ~ 1 | Study_ID / Effect_ID, data = for_analysis_ALL
)

no_moderators_SURVEY <- rma.mv(yi ~ 1,
  V = cov_matrix_SURVEY, random = ~ 1 | Study_ID / Effect_ID, data = for_analysis_SURVEY
)

no_moderators_SURVEILLANCE <- rma.mv(yi ~ 1,
  V = cov_matrix_SURVEILLANCE, random = ~ 1 | Study_ID / Effect_ID, data = for_analysis_SURVEILLANCE
)

```

Fit models to test each variable of interest, together and separately:

```

##### Every variable of interest in the model #####
full_model_ALL <- rma.mv(yi ~ 1 + Decade + Survey_surveillance + Exposure_level_condensed +
  Outcome_data_source_condensed + Victimization_type + Perpetrator_relationship
  Unadjusted,
  V = cov_matrix_ALL, random = ~ 1 | Study_ID / Effect_ID, data = for_analysis_ALL
)

```

```

## Warning in rma.mv(yi ~ 1 + Decade + Survey_surveillance +
## Exposure_level_condensed + : Redundant predictors dropped from the model.

```

```

full_model_SURVEY <- rma.mv(yi ~ 1 + Decade + Survey_surveillance + Exposure_level_condensed +
  Outcome_data_source_condensed + Victimization_type + Perpetrator_relationship
  Unadjusted,
  V = cov_matrix_SURVEY, random = ~ 1 | Study_ID / Effect_ID, data = for_analysis_SURVEY
)

```

```

## Warning in rma.mv(yi ~ 1 + Decade + Survey_surveillance +
## Exposure_level_condensed + : Redundant predictors dropped from the model.

```

```
full_model_SURVEILLANCE <- rma.mv(yi ~ 1 + Decade + Survey_surveillance + Exposure_level_condensed +
                                Outcome_data_source_condensed + Victimization_type + Perpetrator_relationships +
                                Unadjusted,
                                V = cov_matrix_SURVEILLANCE, random = ~ 1 | Study_ID / Effect_ID, data = for_analysis_SURVEILLANCE
)
```

```
## Warning in rma.mv(yi ~ 1 + Decade + Survey_surveillance +
## Exposure_level_condensed + : Redundant predictors dropped from the model.
```

```
##### Decade #####
decade_model_ALL <- rma.mv(yi ~ 0 + Decade,
                          V = cov_matrix_ALL, random = ~ 1 | Study_ID / Effect_ID, data = for_analysis_ALL
)

decade_model_SURVEY <- rma.mv(yi ~ 0 + Decade,
                              V = cov_matrix_SURVEY, random = ~ 1 | Study_ID / Effect_ID, data = for_analysis_SURVEY
)
```

```
## Warning in rma.mv(yi ~ 0 + Decade, V = cov_matrix_SURVEY, random = ~1 | :
## Redundant predictors dropped from the model.
```

```
decade_model_SURVEILLANCE <- rma.mv(yi ~ 0 + Decade,
                                     V = cov_matrix_SURVEILLANCE, random = ~ 1 | Study_ID / Effect_ID, data = for_analysis_SURVEILLANCE
)

##### Exposure level #####
level_ALL <- rma.mv(yi ~ 0 + Exposure_level_condensed,
                   V = cov_matrix_ALL, random = ~ 1 | Study_ID / Effect_ID, data = for_analysis_ALL
)

level_SURVEY <- rma.mv(yi ~ 0 + Exposure_level_condensed,
                       V = cov_matrix_SURVEY, random = ~ 1 | Study_ID / Effect_ID, data = for_analysis_SURVEY
)
```

```
## Warning in rma.mv(yi ~ 0 + Exposure_level_condensed, V = cov_matrix_SURVEY, :
## Redundant predictors dropped from the model.
```

```
level_SURVEILLANCE <- rma.mv(yi ~ 0 + Exposure_level_condensed,
                              V = cov_matrix_SURVEILLANCE, random = ~ 1 | Study_ID / Effect_ID, data = for_analysis_SURVEILLANCE
)

##### Outcome_data_source #####
data_ALL <- rma.mv(yi ~ 0 + Outcome_data_source_condensed,
                  V = cov_matrix_ALL, random = ~ 1 | Study_ID / Effect_ID, data = for_analysis_ALL
)
```

```
## Warning in rma.mv(yi ~ 0 + Outcome_data_source_condensed, V = cov_matrix_ALL, :
## Redundant predictors dropped from the model.
```

```
data_SURVEILLANCE <- rma.mv(yi ~ 0 + Outcome_data_source_condensed,
  V = cov_matrix_SURVEILLANCE, random = ~ 1 | Study_ID / Effect_ID, data = for_analysis_SURVEILLANCE
)
```

```
## Warning in rma.mv(yi ~ 0 + Outcome_data_source_condensed, V =
## cov_matrix_SURVEILLANCE, : Redundant predictors dropped from the model.
```

```
##### Victimization type #####
vic_ALL <- rma.mv(yi ~ 0 + Victimization_type,
  V = cov_matrix_ALL, random = ~ 1 | Study_ID / Effect_ID, data = for_analysis_ALL
)

vic_SURVEY <- rma.mv(yi ~ 0 + Victimization_type,
  V = cov_matrix_SURVEY, random = ~ 1 | Study_ID / Effect_ID, data = for_analysis_SURVEY
)
```

```
## Warning in rma.mv(yi ~ 0 + Victimization_type, V = cov_matrix_SURVEY, random =
## ~1 | : Redundant predictors dropped from the model.
```

```
vic_SURVEILLANCE <- rma.mv(yi ~ 0 + Victimization_type,
  V = cov_matrix_SURVEILLANCE, random = ~ 1 | Study_ID / Effect_ID, data = for_analysis_SURVEILLANCE
)
```

```
##### Perpetrator relationship #####
perp_ALL <- rma.mv(yi ~ 0 + Perpetrator_relationship,
  V = cov_matrix_ALL, random = ~ 1 | Study_ID / Effect_ID, data = for_analysis_ALL
)
```

```
perp_SURVEY <- rma.mv(yi ~ 1 + 0 + Perpetrator_relationship,
  V = cov_matrix_SURVEY, random = ~ 1 | Study_ID / Effect_ID, data = for_analysis_SURVEY
)
```

```
perp_SURVEILLANCE <- rma.mv(yi ~ 0 + Perpetrator_relationship,
  V = cov_matrix_SURVEILLANCE, random = ~ 1 | Study_ID / Effect_ID, data = for_analysis_SURVEILLANCE
)
```

```
##### Exposure_level_condensed #####
exp_EXPOSURE_LEVEL_CONDENSED <- rma.mv(yi ~ 0 + Exposure_level_condensed,
  V = cov_matrix_EXPOSURE_LEVEL_CONDENSED, random = ~ 1 | Study_ID / Effect_ID, data = for_analysis_EXP
)
```

```
## Warning in rma.mv(yi ~ 0 + Exposure_level_condensed, V =
## cov_matrix_EXPOSURE_LEVEL_CONDENSED, : Redundant predictors dropped from the
## model.
```

```
##### Survey / surveillance #####
survey_surveillance_ALL <- rma.mv(yi ~ 0 + Survey_surveillance,
  V = cov_matrix_ALL, random = ~ 1 | Study_ID / Effect_ID, data = for_analysis_ALL
)
```

```
##### Race #####
```

```

race_BLACK_WHITE <- rma.mv(yi ~ 0 + Race,
  V = cov_matrix_BLACK_WHITE, random = ~ 1 | Study_ID / Effect_ID, data = for_analysis_BLACK_WHITE
)

```

```

## Warning in rma.mv(yi ~ 0 + Race, V = cov_matrix_BLACK_WHITE, random = ~1 | :
## Redundant predictors dropped from the model.

```

Meta-regression output:

```

### T-tests for each variable:

# #these tests are RVE based and are robust to misspecification of the variances and covariances
# coef_test(
#   obj = multilevel_model, #estimation model above
#   #cluster = for_analysis$study_id, #define cluster IDs (not needed, already specified in model)
#   vcov = "CR2" #estimation method (CR2 is best)
# )

# coef_test(full_model_ALL, vcov = "CR2")
# coef_test(full_model_SURVEY, vcov = "CR2")
# coef_test(full_model_SURVEILLANCE, vcov = "CR2")

##### No moderators #####
#coef_test(no_moderators_ALL, vcov = "CR2")
conf_int(no_moderators_ALL, vcov = "CR2")

##      Coef Estimate      SE d.f. Lower 95% CI Upper 95% CI
## 1 intrcpt -0.00987 0.0375 30.9      -0.0863      0.0665

#coef_test(no_moderators_SURVEY, vcov = "CR2")
conf_int(no_moderators_SURVEY, vcov = "CR2")

##      Coef Estimate      SE d.f. Lower 95% CI Upper 95% CI
## 1 intrcpt  0.0786 0.0296 5.85      0.0056      0.152

#coef_test(no_moderators_SURVEILLANCE, vcov = "CR2")
conf_int(no_moderators_SURVEILLANCE, vcov = "CR2")

##      Coef Estimate      SE d.f. Lower 95% CI Upper 95% CI
## 1 intrcpt  -0.056 0.0496 21.5      -0.159      0.0469

##### Survey / surveillance #####
coef_test(survey_surveillance_ALL, vcov = "CR2")

##      Coef. Estimate      SE t-stat d.f. p-val (Satt)
## 1 Survey_surveillanceSurvey  0.0945 0.0373  2.54 8.63  0.0329
## 2 Survey_surveillanceSurveillance -0.0566 0.0489 -1.16 21.12  0.2599
## Sig.
## 1 *
## 2

```

```
#conf_int(survey_surveillance_ALL, vcov = "CR2")
```

```
##### Exposure_level_condensed #####
coef_test(exp_EXPOSURE_LEVEL_CONDENSED, vcov = "CR2")
```

```
##              Coef. Estimate      SE t-stat d.f. p-val (Satt)
## 1 Exposure_level_condensedCity/MSA -0.0417 0.0214 -1.952 13.7      0.0717
## 2 Exposure_level_condensedCounty -0.0714 0.0443 -1.612 2.9      0.2083
## 3 Exposure_level_condensedState 0.0455 0.1131 0.403 9.6      0.6959
## Sig.
## 1 .
## 2
## 3
```

```
##### Race #####
coef_test(race_BLACK_WHITE, vcov = "CR2")
```

```
##              Coef. Estimate      SE t-stat d.f. p-val (Satt) Sig.
## 1 RaceBlack -0.234 0.212 -1.10 2.95      0.352
## 2 RaceWhite -0.233 0.226 -1.03 2.99      0.379
```

```
##### Decade #####
coef_test(decade_model_ALL, vcov = "CR2")
```

```
##              Coef. Estimate      SE t-stat d.f. p-val (Satt) Sig.
## 1 Decade1970s 0.0227 0.0673 0.337 1.61      0.775
## 2 Decade1980s 0.0615 0.0417 1.474 5.86      0.192
## 3 Decade1990s -0.0349 0.0251 -1.391 10.97      0.192
## 4 Decade2000s -0.0774 0.0804 -0.963 7.56      0.365
## 5 Decade2010s 0.0876 0.0559 1.567 3.94      0.193
```

```
#conf_int(decade_model_ALL, vcov = "CR2")
#decade_model_ALL
```

```
#coef_test(decade_model_SURVEY, vcov = "CR2")
#coef_test(decade_model_SURVEILLANCE, vcov = "CR2")
```

```
##### Exposure level #####
# coef_test(level_ALL, vcov = "CR2")
# coef_test(level_SURVEY, vcov = "CR2")
# coef_test(level_SURVEILLANCE, vcov = "CR2")
```

```
##### Outcome_data_source #####
#coef_test(data_ALL, vcov = "CR2")
coef_test(data_SURVEILLANCE, vcov = "CR2")
```

```
##              Coef. Estimate      SE t-stat
## 1 Outcome_data_source_condensedFBI -0.0330 0.0292 -1.129
## 2 Outcome_data_source_condensedLocal law enforcement -0.0102 0.0520 -0.197
## 3 Outcome_data_source_condensedCDC -0.2289 0.2543 -0.900
## d.f. p-val (Satt) Sig.
```

```
## 1 17.25      0.274
## 2  1.85      0.864
## 3  2.89      0.437
```

```
##### Victimization type #####
coef_test(vic_ALL, vcov = "CR2")
```

```
##                               Coef. Estimate      SE t-stat  d.f.
## 1      Victimization_typeHomicide -0.0975 0.0786 -1.241 11.66
## 2 Victimization_typeIPV, harassment, & assault  0.0671 0.0325  2.064 11.72
## 3      Victimization_typeRape & sexual assault  0.0266 0.0361  0.736  9.15
##  p-val (Satt) Sig.
## 1      0.2391
## 2      0.0619
## 3      0.4804
```

```
#coef_test(vic_SURVEY, vcov = "CR2")
#coef_test(vic_SURVEILLANCE, vcov = "CR2")
```

```
##### Perpetrator relationship #####
coef_test(perp_ALL, vcov = "CR2")
```

```
##                               Coef. Estimate      SE t-stat
## 1      Perpetrator_relationshipDomestic/intimate partner  0.0269 0.0388  0.694
## 2 Perpetrator_relationshipNon-domestic known perpetrator  0.0227 0.0341  0.667
## 3      Perpetrator_relationshipUnknown or not specified -0.0368 0.0452 -0.815
##  d.f. p-val (Satt) Sig.
## 1 16.68      0.497
## 2  4.42      0.538
## 3 23.91      0.423
```

```
#coef_test(perp_SURVEY, vcov = "CR2")
#coef_test(perp_SURVEILLANCE, vcov = "CR2")
```

Funnel plot: publication bias

```
pdf(file = 'PlotsTables/funnel_plot.pdf')
funnel(no_moderators_ALL)
dev.off()
```

```
## pdf
## 2
```

```
ranktest(no_moderators_ALL)
```

```
## Warning in cor.test.default(yi.star, vi, method = "kendall", exact = TRUE):
## Cannot compute exact p-value with ties
```

```
##  
## Rank Correlation Test for Funnel Plot Asymmetry  
##  
## Kendall's tau = 0.0124, p = 0.7991
```

QUESTION: is this still appropriately down-weighting estimates within the same study? I don't think so

Forest plots

(saved as PDFs)

```
## pdf  
## 2
```

```
## pdf  
## 2
```

```
## pdf  
## 2
```

```
## pdf  
## 2
```