````
---
title: "data_prep"
format: html
editor: visual
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(
  echo = FALSE,       # Hide code by default
  warning = FALSE,
  message = FALSE
)

# Load packages
library(dplyr)
library(tidyr)
library(readr)
library(janitor)   # for clean_names()
library(tibble)
library(stringr)
library(haven)



```

## 0. Preliminary - Load Data +s Troubleshoot

```{r}
#| eval: false
#| include: false
# Unzip and load data

raw_data <- read_csv("/Users/kristinamensik/Documents/Duke/2024-2025/902 Legislators Pre-Ananlysis Plan/Prelim Folder/Prelim 3-23/data/raw/scotg_results.csv")


raw_data <- clean_names(raw_data)
```

### Create Codebook

```{r}
#| eval: false
#| include: false
generate_q4_base_codebook <- function(raw_data) {
  q4_binary_cols <- names(raw_data)[str_detect(names(raw_data), "^Q4_\\d+_\\d+$")]

  tibble(variable_name = q4_binary_cols) %>%
    mutate(
      statement_number = str_extract(variable_name, "(?<=Q4_)\\d+(?=_)"),
      response_option_code = str_extract(variable_name, "(?<=_)\\d+$"),
      response_option = case_when(
        response_option_code == "1" ~ "You",
        response_option_code == "2" ~ "Someone you know well",
        response_option_code == "3" ~ "No/Don't know",
        TRUE ~ NA_character_
      ),
      type = "Binary (0/1)",
      values = "1 = selected, 0 = not selected, NA = no response",
      notes = "From Q4 matrix question (multiple selection possible)"
    )
}
````
````

```

#### 0a. troubleshooting to figure out how different question types exported

# (Removed early troubleshooting code for block assignment using col indices 998–1000;

# now using named columns like FL_8_DO_Deserving&PowerBlockA)

## 1. Cleaning + Creating Codebook

#### a. Label Deserving & Power Blocks

```{r}
#| eval: false
#| include: false

# DeservingPower Blocks


FL_8_DO_Deserving&PowerBlockA %in% names(raw_data)

# Add participant block label (group a, b, c)
raw_data <- raw_data %>%
  mutate(
    block = case_when(
      !is.na(`FL_8_DO_Deserving&PowerBlockA`) ~ "A",
      !is.na(`FL_8_DO_Deserving&PowerBlockB`) ~ "B",
      !is.na(`FL_8_DO_Deserving&PowerBlockC`) ~ "C",
      TRUE ~ NA_character_
    )
  )

# Save labeled but otherwise unmodified version
write_csv(raw_data, "/Users/kristinamensik/Documents/Duke/2024-2025/902 Legislators Pre-
Ananlysis Plan/Prelim Folder/Prelim 3-23/data/cleaned/labeled_raw_data.csv")

```

b. Create cleaned data frame, deleting un-needed cols

```{r}
#| eval: false
#| include: false

columns_to_drop <- c(
  "Status", "StartDate", "EndDate", "IPAddress", "RecipientLastName",
"RecipientFirstName",
  "RecipientEmail", "ExternalReference", "RecordedDate", "LocationLatitude",
  "LocationLongitude", "DistributionChannel", "UserLanguage"
)

# Drop all columns containing "_DO" (Display Order)
do_columns <- grep("_DO", names(raw_data), value = TRUE)

# Combine and remove
raw_data_cleaned <- raw_data %>%
  select(-all_of(c(columns_to_drop, do_columns)))

```

c. Clean: drop non-consenters

```{r}
#| eval: false
#| include: false
```
```

```
# Filter to those who consented
raw_data_cleaned <- raw_data_cleaned %>%
  filter(`Consent:` == 1)

# Save again
write_csv(raw_data_cleaned, "data/cleaned/cleaned_data_consent.csv")

#remove consent col

raw_data_cleaned <- raw_data_cleaned %>%
  select(-`Consent:`)
```

d. Clean: attn checks

-   first, create flags for passed/not each check and people who failed all

```{r}
#| eval: false
#| include: false

# 1. Individual pass/fail checks (already defined earlier in your script)
attn_check_q9 <- raw_data_cleaned$Q9 == 4

attn_text <- tolower(str_trim(raw_data_cleaned$Q17))
accepted_variants <- c("purple", "\"purple\"", "purpke", "purplee", "prurprle", "purpl",
"purp", "💜", "🟣")
attn_check_q17 <- attn_text %in% accepted_variants

attn_check_q1114 <- raw_data_cleaned$Q114_4 == 6
attn_check_q1114[raw_data_cleaned$Q114_4 == -99] <- FALSE
attn_check_q1114[is.na(raw_data_cleaned$Q114_4)] <- TRUE  # not shown → pass

# 2. Create score + filter full failures
raw_data_cleaned <- raw_data_cleaned %>%
  mutate(
    attn_q9_pass = attn_check_q9,
    attn_q17_pass = attn_check_q17,
    attn_q1114_pass = attn_check_q1114,
    attn_pass_count = as.integer(attn_q9_pass) + as.integer(attn_q17_pass) +
as.integer(attn_q1114_pass)
  ) %>%
  filter(attn_pass_count > 0)  # Remove only people who failed all
```

e. Clean: via survey duration

```{r}
#| eval: false
#| include: false

str(raw_data_cleaned$`Duration (in seconds)`)

raw_data_cleaned$duration_sec <- as.numeric(raw_data_cleaned$`Duration (in seconds)`)

raw_data_cleaned <- raw_data_cleaned %>%
  filter(duration_sec >= 480)
```

e. Clean: ppl who do not answer my IV ??

2. Variables

A. Deservingness and Power consolidation

Objective: consolidate blocks (so only 1 Incarcerated People/Prisoners)

```{r}
#| eval: false
#| include: false
library(readr)
mapping <- read_csv("/Users/kristinamensik/Documents/Duke/2024-2025/902 Legislators Pre-Ananlysis Plan/Prelim Folder/Prelim 3-23/data/Unified_Deserving_Power_Target_Mapping.csv")

raw_data_preserved <- raw_data_cleaned  # back up before modification

```

Prep by converting to numeric

```{r}
#| eval: false
#| include: false
# Identify all Deserving and Power columns
deserving_power_cols <- names(raw_data_cleaned)[str_detect(names(raw_data_cleaned), "^(Deserving|Power) Group")]

raw_data_cleaned <- raw_data_cleaned %>%
  mutate(across(
    all_of(deserving_power_cols),
    as.character
  ))

raw_data_cleaned <- raw_data_cleaned %>%
  mutate(across(
    all_of(deserving_power_cols),
    ~ as.numeric(na_if(.x, "-99"))
  ))

table(raw_data_cleaned$`Deserving Group A_1`, useNA = "always")

```

ALL Evals: Create consolidated variable names for deservingness

```{r}
#| eval: false
#| include: false

new_vars <- c(
  "d_black_afam_abc", "d_native_ind_abc", "d_asianam_abc", "d_latino_his_abc", "d_whiteam_abc",
  "d_mideastam_abc", "d_incwomen_abc", "d_collegewom_abc", "d_enviro_abc", "d_feminists_abc",
  "d_proabort_choice_abc", "d_abortprov_abc", "d_welfare_tanf_abc", "d_fstamps_snap_abc",
  "d_unions_abc", "d_medicaid_abc", "d_medicare_abc", "d_unemployed_abc", "d_police_abc",
  "d_poorfam_abc", "d_wealthy_abc", "d_teachers_abc", "d_sci_experts_abc", "d_child_abc",
  "d_crim_abc", "d_incppl_abc", "d_homeless_abc", "d_immigrants_abc", "d_unauth_undoc_abc",
  "d_gunowners_abc", "d_voters_abc", "d_lgbt_abc", "d_trans_abc", "d_transkids_abc", "d_gunmans_abc",
  "d_ice_abc", "d_socmed_abc", "d_nra_abc", "d_bigbank_abc", "d_bigcorp_abc", "d_uni_abc",
  "d_comcoll_abc", "d_noncit_parent_a", "d_incparents_a", "d_hsteen_a", "d_athiest_a", "d_cath_a",
  "d_evan_a", "d_jews_a", "d_pal_a", "d_muslim_a", "d_blackwom_a", "d_blackmen_a", "d_lds_a",
```

```
  "d_longcov_a", "d_exon_a", "d_weed_a", "d_opioid_a", "d_whitecol_a", "d_tech_a",
"d_smokers_a",
  "d_genx_a", "d_boomer_a", "d_mill_a", "d_genz_a"
)

for (var in new_vars) {
  raw_data_cleaned[[var]] <- NA #to initialize
}


```

```{r blockspecifictargets}
#| eval: false
#| include: false
# Load the spreadsheet with the mappings
library(readxl)
mapping <- read_excel("scratch for consolidation.xlsx")

# Create composite deservingness variables from the correct block column only
for (i in 1:nrow(mapping)) {
  new_var <- mapping$target_slug[i]
  var_a <- mapping$`Deserving Group A`[i]
  var_b <- mapping$`Deserving Group B`[i]
  var_c <- mapping$`Deserving Group C`[i]

  raw_data_cleaned[[new_var]] <- case_when(
    raw_data_cleaned$block == "A" ~ raw_data_cleaned[[var_a]],
    raw_data_cleaned$block == "B" ~ raw_data_cleaned[[var_b]],
    raw_data_cleaned$block == "C" ~ raw_data_cleaned[[var_c]],
    TRUE ~ NA_real_
  )
}

```

LIMITED Evals: recode group specifics

```{r}
#| eval: false
#| include: false

# Load the split mapping
split_map <- read.csv("/Users/kristinamensik/Library/Mobile
Documents/com~apple~CloudDocs/Documents/Duke/2024-2025/902 Legislators Pre-Ananlysis
Plan/Prelim Folder/Prelim 3-23/deserving_split.csv")

# Loop through and rename each variable
for (i in 1:nrow(split_map)) {

  old_var <- split_map$var.name[i]     # e.g., "Deserving Group A_43"
  new_var <- split_map$target_slug[i]  # e.g., "d_noncit_parent_a"

  # Just copy the values over
  raw_data_cleaned[[new_var]] <- raw_data_cleaned[[old_var]]
}

View(raw_data_cleaned)
```

```{r, prisonerdeserves}

# pretty much done, maybe standardize later

```

```{r desor}
#| eval: false
#| include: false

# List of deserving control variables
deserving_control_vars <- c(
  "d_welfare_tanf_abc", "d_fstamps_snap_abc", "d_medicaid_abc", "d_medicare_abc",
  "d_unemployed_abc", "d_poorfam_abc", "d_homeless_abc", "d_immigrants_abc",
  "d_unauth_undoc_abc", "d_trans_abc", "d_transkids_abc",
  "d_noncit_parent_a", "d_unins_c", "d_mentill_c"
)

# Create composite deservingness control variable
raw_data_cleaned <- raw_data_cleaned %>%
  mutate(desor = rowMeans(select(., all_of(deserving_control_vars)), na.rm = TRUE))


```

####A1. Deserving Validation

```{r}
#CHECK INDIVIDUALS
#raw_data_cleaned %>%
#  filter(ResponseId == "R_5DugmlzJSCFvja1") %>%
 # select(block, `Deserving Group A_1`, d_black_afam_abc)

```

B. Q4 - Experiences Block

```{r, cleaningexperiences1}

#view(raw_data_cleaned$Q4_1_1)

# Basic Totals
#I'm counting how many people said they themselves have been incarcerated by checking
where Q4_1_1 equals 1. I'm ignoring missing values
total_incarcerated <- sum(raw_data_cleaned$Q4_1_1 == 1, na.rm = TRUE)

#"I'm counting how many people said they know someone who's been incarcerated (Q4_1_2 ==
1). Again, I'm skipping over any missing answers."
total_knows_incarcerated <- sum(raw_data_cleaned$Q4_1_2 == 1, na.rm = TRUE)

#I'm counting how many people selected 'no / unsure' when asked about incarceration
experience (Q4_1_3 == 1)
total_no_incarcerated <- sum(raw_data_cleaned$Q4_1_3 == 1, na.rm = TRUE)

#I'm counting how many respondents said they themselves were a victim of violent crime.
total_victim <- sum(raw_data_cleaned$Q4_2_1 == 1, na.rm = TRUE)

#I'm counting how many said they know someone who was a victim of violent crime
total_knows_victim<- sum(raw_data_cleaned$Q4_2_2 == 1, na.rm = TRUE)

#I'm counting those who selected 'no / unsure' for victimization experience
total_no_victim <- sum(raw_data_cleaned$Q4_2_3 == 1, na.rm = TRUE)

#counting the total of those who said they were incarcerated, know incarcerated, or
victims)

raw_data_cleaned <- raw_data_cleaned %>%
  mutate(
    impact = if_else(
      Q4_1_1 == 1 | Q4_1_2 == 1 | Q4_2_1 == 1,
```

```
      1, 0,
      missing = NA_real_
    )
  )


```

Create fingerprint

-   create a new variable named contact_pattern

-   for each respondent, create a 6 digit code that summarizes how they answered each Q4_1 and Q4_2 question

    -   if the person said yes (1), use that 1. If they didn't answer, treat it as 0

-   combine all 6 digits starting with incarceration responses then victimization to create pattern to put in var

```{r, experiencesfingerprints1}
raw_data_cleaned <- raw_data_cleaned %>%
  mutate(
    contact_pattern = paste0(
      coalesce(Q4_1_1, 0),
      coalesce(Q4_1_2, 0),
      coalesce(Q4_1_3, 0),
      coalesce(Q4_2_1, 0),
      coalesce(Q4_2_2, 0),
      coalesce(Q4_2_3, 0)
    )
  )
```

Problem Report
Some respondents selected both a contact experience (e.g., "been incarcerated") and also marked “none/unsure”, which is logically inconsistent. These may reflect confusion, misclicks, or low-quality responses.

Conflict Type	Count
Q4_1_1 (Been incarcerated) & Q4_1_3 (None/Unsure)	1
Q4_1_2 (Knows incarcerated) & Q4_1_3 (None/Unsure)	1
Q4_2_1 (Been victimized) & Q4_2_3 (None/Unsure)	10
Q4_2_2 (Knows victim) & Q4_2_3 (None/Unsure)	10

```{r, experiencesfingerprints}
#table(raw_data_cleaned$Q4_1_3, useNA = "always")
#table(raw_data_cleaned$Q4_2_3, useNA = "always")


strange1 <- raw_data_cleaned %>%
  filter(Q4_1_1 == 1 & Q4_1_3 == 1)

strange2 <- raw_data_cleaned %>%
  filter(Q4_1_2 == 1 & Q4_1_3 == 1)

strange3 <- raw_data_cleaned %>%
  filter(Q4_2_1 == 1 & Q4_1_3 == 1)

strange4 <- raw_data_cleaned %>%
  filter(Q4_2_2 == 1 & Q4_1_3 == 1)

print(strange1)
print(strange2)
print(strange3)
```

```
print(strange4)

```

Tabulate unique patterns

```{r, allexperienceuniquecombos}

combo_summary <- raw_data_cleaned %>%
  group_by(contact_pattern) %>%
  summarise(
    N = n(),
    pct_total = 100 * n() / nrow(raw_data_cleaned),
    pct_of_been_incarcerated = 100 * sum(Q4_1_1 == 1, na.rm = TRUE) / total_incarcerated,
    pct_of_knows_incarcerated = 100 * sum(Q4_1_2 == 1, na.rm = TRUE) /
total_knows_incarcerated,
    pct_of_been_victim = 100 * sum(Q4_2_1 == 1, na.rm = TRUE) / total_victim
  ) %>%
  arrange(desc(N))
```



```{r}
# Define labeled combinations
combo_labels <- tibble::tibble(
  contact_pattern = c(
    "100000",  # Incarcerated Only
    "110000",  # Also KI
    "110100",  # Also KI and Vict
    "100100",  # Also Vict
    "100010",  # Also and only Vict
    "110110",  # Also and only KI and Vict
    "010000",  # KI Only
    "010100",  # Also Vict
    "010110",  # Also and only Vict
    "001100",  # Victim Only
    "011100"   # Also and only KI
  ),
  label = c(
    "Incarcerated Only",
    "Also KI",
    "Also KI and Vict",
    "Also Vict",
    "Also and only Vict",
    "Also and only KI and Vict",
    "KI Only",
    "Also Vict",
    "Also and only Vict",
    "Victim Only",
    "Also and only KI"
  )
)

# Join to summary
combo_appendix_data <- combo_summary %>%
  left_join(combo_labels, by = "contact_pattern") %>%
  filter(!is.na(label)) %>%
  mutate(
    pct_of_incarcerated = if_else(
      pct_of_been_incarcerated > 0,
      100 * N / sum(N[pct_of_been_incarcerated > 0], na.rm = TRUE),
      NA_real_
    ),
    pct_of_ki = if_else(
      pct_of_knows_incarcerated > 0,
```

```
      100 * N / sum(N[pct_of_knows_incarcerated > 0], na.rm = TRUE),
      NA_real_
    ),
    pct_of_victim = if_else(
      pct_of_been_victim > 0,
      100 * N / sum(N[pct_of_been_victim > 0], na.rm = TRUE),
      NA_real_
    )
  )

View(combo_appendix_data)

```

charts
```{r}

# Calculate basic totals
table_a <- tibble::tibble(
  Type = c("Been Incarcerated", "Knows Incarcerated", "Been Victimized"),
  Count = c(total_incarcerated, total_knows_incarcerated, total_victim),
  Percent_of_Sample = round(100 * Count / nrow(raw_data_cleaned), 1)
)

table_a %>%
  gt() %>%
  cols_label(
    Type = "Type",
    Count = "Total Count",
    Percent_of_Sample = "Percent of Sample"
  ) %>%
  tab_header(title = "Appendix Table A. Basic Contact Types") %>%
  tab_options(table.font.size = "small")

```

```{r}

combo_appendix_data %>%
  filter(label %in% c(
    "Incarcerated Only",
    "Also KI",
    "Also and only KI",
    "Also Vict",
    "Also and only Vict",
    "Also KI and Vict",
    "Also and only KI and Vict"
  )) %>%
  select(label, N, pct_total, pct_of_incarcerated) %>%
  arrange(desc(N)) %>%
  gt() %>%
  cols_label(
    label = "Contact Combo",
    N = "Count",
    pct_total = "Percent of Sample",
    pct_of_incarcerated = "Percent of Total Incarcerated Count"
  ) %>%
  fmt_number(columns = where(is.numeric), decimals = 1) %>%
  tab_header(title = "Appendix Table B. Incarcerated (Inc) Combos") %>%
  tab_options(table.font.size = "small")

```
```{r}

combo_appendix_data %>%
```

```
  filter(label %in% c(
    "KI Only",
    "Also Vict",
    "Also and only Vict"
  )) %>%
  select(label, N, pct_total, pct_of_ki) %>%
  arrange(desc(N)) %>%
  gt() %>%
  cols_label(
    label = "Contact Combo",
    N = "Count",
    pct_total = "Percent of Sample",
    pct_of_ki = "Percent of Total KI Count"
  ) %>%
  fmt_number(columns = where(is.numeric), decimals = 1) %>%
  tab_header(title = "Appendix Table C. Knows Incarcerated (KI) Combos") %>%
  tab_options(table.font.size = "small")
```
```{r}

combo_appendix_data %>%
  filter(label %in% c(
    "Victim Only",
    "Also KI",
    "Also and only KI"
  )) %>%
  select(label, N, pct_total, pct_of_victim) %>%
  arrange(desc(N)) %>%
  gt() %>%
  cols_label(
    label = "Contact Combo",
    N = "Count",
    pct_total = "Percent of Sample",
    pct_of_victim = "Percent of Total Victim Count"
  ) %>%
  fmt_number(columns = where(is.numeric), decimals = 1) %>%
  tab_header(title = "Appendix Table D. Victim (VI) Combos") %>%
  tab_options(table.font.size = "small")

```



```{r Incarcerated Combos}
#| eval: false
#| include: false

incarcerated_only <- sum(
  raw_data_cleaned$Q4_1_1 == 1 &
  raw_data_cleaned$Q4_1_2 == 0 &
  raw_data_cleaned$Q4_1_3 == 0 &
  raw_data_cleaned$Q4_2_1 == 0 &
  raw_data_cleaned$Q4_2_2 == 0 &
  raw_data_cleaned$Q4_2_3 == 0 &
  !is.na(raw_data_cleaned$Q4_1_1) &
  !is.na(raw_data_cleaned$Q4_1_2) &
  !is.na(raw_data_cleaned$Q4_1_3) &
  !is.na(raw_data_cleaned$Q4_2_1) &
  !is.na(raw_data_cleaned$Q4_2_2) &
  !is.na(raw_data_cleaned$Q4_2_3)
)

incarcerated_knowsinc <- sum(
  raw_data_cleaned$Q4_1_1 == 1 &
  raw_data_cleaned$Q4_1_2 == 1 &
```

```
  raw_data_cleaned$Q4_1_3 == 0 &
  raw_data_cleaned$Q4_2_1 == 0 &
  raw_data_cleaned$Q4_2_2 == 0 &
  raw_data_cleaned$Q4_2_3 == 0 &
  !is.na(raw_data_cleaned$Q4_1_1) &
  !is.na(raw_data_cleaned$Q4_1_2) &
  !is.na(raw_data_cleaned$Q4_1_3) &
  !is.na(raw_data_cleaned$Q4_2_1) &
  !is.na(raw_data_cleaned$Q4_2_2) &
  !is.na(raw_data_cleaned$Q4_2_3)
)

inc_knowsinc_victim <- sum(
  raw_data_cleaned$Q4_1_1 == 1 &
  raw_data_cleaned$Q4_1_2 == 1 &
  raw_data_cleaned$Q4_1_3 == 0 &
  raw_data_cleaned$Q4_2_1 == 1 &
  raw_data_cleaned$Q4_2_2 == 0 &
  raw_data_cleaned$Q4_2_3 == 0 &
  !is.na(raw_data_cleaned$Q4_1_1) &
  !is.na(raw_data_cleaned$Q4_1_2) &
  !is.na(raw_data_cleaned$Q4_1_3) &
  !is.na(raw_data_cleaned$Q4_2_1) &
  !is.na(raw_data_cleaned$Q4_2_2) &
  !is.na(raw_data_cleaned$Q4_2_3)
)

inc_knowsinc_knowsvict <- sum(
  raw_data_cleaned$Q4_1_1 == 1 &
  raw_data_cleaned$Q4_1_2 == 1 &
  raw_data_cleaned$Q4_1_3 == 0 &
  raw_data_cleaned$Q4_2_1 == 0 &
  raw_data_cleaned$Q4_2_2 == 1 &
  raw_data_cleaned$Q4_2_3 == 0 &
  !is.na(raw_data_cleaned$Q4_1_1) &
  !is.na(raw_data_cleaned$Q4_1_2) &
  !is.na(raw_data_cleaned$Q4_1_3) &
  !is.na(raw_data_cleaned$Q4_2_1) &
  !is.na(raw_data_cleaned$Q4_2_2) &
  !is.na(raw_data_cleaned$Q4_2_3)
)


inc_knowsinc_victim_knowsvict <- sum(
  raw_data_cleaned$Q4_1_1 == 1 &
  raw_data_cleaned$Q4_1_2 == 1 &
  raw_data_cleaned$Q4_1_3 == 0 &
  raw_data_cleaned$Q4_2_1 == 1 &
  raw_data_cleaned$Q4_2_2 == 1 &
  raw_data_cleaned$Q4_2_3 == 0 &
  !is.na(raw_data_cleaned$Q4_1_1) &
  !is.na(raw_data_cleaned$Q4_1_2) &
  !is.na(raw_data_cleaned$Q4_1_3) &
  !is.na(raw_data_cleaned$Q4_2_1) &
  !is.na(raw_data_cleaned$Q4_2_2) &
  !is.na(raw_data_cleaned$Q4_2_3)
)

incarcerated_victim <- sum(
  raw_data_cleaned$Q4_1_1 == 1 &
  raw_data_cleaned$Q4_1_2 == 0 &
  raw_data_cleaned$Q4_1_3 == 0 &
  raw_data_cleaned$Q4_2_1 == 1 &
  raw_data_cleaned$Q4_2_2 == 0 &
  raw_data_cleaned$Q4_2_3 == 0 &
```

```
  !is.na(raw_data_cleaned$Q4_1_1) &
  !is.na(raw_data_cleaned$Q4_1_2) &
  !is.na(raw_data_cleaned$Q4_1_3) &
  !is.na(raw_data_cleaned$Q4_2_1) &
  !is.na(raw_data_cleaned$Q4_2_2) &
  !is.na(raw_data_cleaned$Q4_2_3)
)

inc_victim_knowsvict <- sum(
  raw_data_cleaned$Q4_1_1 == 1 &
  raw_data_cleaned$Q4_1_2 == 0 &
  raw_data_cleaned$Q4_1_3 == 0 &
  raw_data_cleaned$Q4_2_1 == 1 &
  raw_data_cleaned$Q4_2_2 == 1 &
  raw_data_cleaned$Q4_2_3 == 0 &
  !is.na(raw_data_cleaned$Q4_1_1) &
  !is.na(raw_data_cleaned$Q4_1_2) &
  !is.na(raw_data_cleaned$Q4_1_3) &
  !is.na(raw_data_cleaned$Q4_2_1) &
  !is.na(raw_data_cleaned$Q4_2_2) &
  !is.na(raw_data_cleaned$Q4_2_3)
)

incarcerated_knowsvict <- sum(
  raw_data_cleaned$Q4_1_1 == 1 &
  raw_data_cleaned$Q4_1_2 == 0 &
  raw_data_cleaned$Q4_1_3 == 0 &
  raw_data_cleaned$Q4_2_1 == 0 &
  raw_data_cleaned$Q4_2_2 == 1 &
  raw_data_cleaned$Q4_2_3 == 0 &
  !is.na(raw_data_cleaned$Q4_1_1) &
  !is.na(raw_data_cleaned$Q4_1_2) &
  !is.na(raw_data_cleaned$Q4_1_3) &
  !is.na(raw_data_cleaned$Q4_2_1) &
  !is.na(raw_data_cleaned$Q4_2_2) &
  !is.na(raw_data_cleaned$Q4_2_3)
)


knowsinc_only <- sum(
  raw_data_cleaned$Q4_1_1 == 0 &
  raw_data_cleaned$Q4_1_2 == 1 &
  raw_data_cleaned$Q4_1_3 == 0 &
  raw_data_cleaned$Q4_2_1 == 0 &
  raw_data_cleaned$Q4_2_2 == 0 &
  raw_data_cleaned$Q4_2_3 == 0 &
  !is.na(raw_data_cleaned$Q4_1_1) &
  !is.na(raw_data_cleaned$Q4_1_2) &
  !is.na(raw_data_cleaned$Q4_1_3) &
  !is.na(raw_data_cleaned$Q4_2_1) &
  !is.na(raw_data_cleaned$Q4_2_2) &
  !is.na(raw_data_cleaned$Q4_2_3)
)

knowsinc_victim <- sum(
  raw_data_cleaned$Q4_1_1 == 0 &
  raw_data_cleaned$Q4_1_2 == 1 &
  raw_data_cleaned$Q4_1_3 == 0 &
  raw_data_cleaned$Q4_2_1 == 1 &
  raw_data_cleaned$Q4_2_2 == 0 &
  raw_data_cleaned$Q4_2_3 == 0 &
  !is.na(raw_data_cleaned$Q4_1_1) &
  !is.na(raw_data_cleaned$Q4_1_2) &
  !is.na(raw_data_cleaned$Q4_1_3) &
  !is.na(raw_data_cleaned$Q4_2_1) &
```

```
  !is.na(raw_data_cleaned$Q4_2_2) &
  !is.na(raw_data_cleaned$Q4_2_3)
)

knowsinc_knowsvict <- sum(
  raw_data_cleaned$Q4_1_1 == 0 &
  raw_data_cleaned$Q4_1_2 == 1 &
  raw_data_cleaned$Q4_1_3 == 0 &
  raw_data_cleaned$Q4_2_1 == 0 &
  raw_data_cleaned$Q4_2_2 == 1 &
  raw_data_cleaned$Q4_2_3 == 0 &
  !is.na(raw_data_cleaned$Q4_1_1) &
  !is.na(raw_data_cleaned$Q4_1_2) &
  !is.na(raw_data_cleaned$Q4_1_3) &
  !is.na(raw_data_cleaned$Q4_2_1) &
  !is.na(raw_data_cleaned$Q4_2_2) &
  !is.na(raw_data_cleaned$Q4_2_3)
)

knowsinc_victim_knowsvict <- sum(
  raw_data_cleaned$Q4_1_1 == 0 &
  raw_data_cleaned$Q4_1_2 == 1 &
  raw_data_cleaned$Q4_1_3 == 0 &
  raw_data_cleaned$Q4_2_1 == 1 &
  raw_data_cleaned$Q4_2_2 == 1 &
  raw_data_cleaned$Q4_2_3 == 0 &
  !is.na(raw_data_cleaned$Q4_1_1) &
  !is.na(raw_data_cleaned$Q4_1_2) &
  !is.na(raw_data_cleaned$Q4_1_3) &
  !is.na(raw_data_cleaned$Q4_2_1) &
  !is.na(raw_data_cleaned$Q4_2_2) &
  !is.na(raw_data_cleaned$Q4_2_3)
)

vict_only <- sum(
  raw_data_cleaned$Q4_1_1 == 0 &
  raw_data_cleaned$Q4_1_2 == 0 &
  raw_data_cleaned$Q4_1_3 == 0 &
  raw_data_cleaned$Q4_2_1 == 1 &
  raw_data_cleaned$Q4_2_2 == 0 &
  raw_data_cleaned$Q4_2_3 == 0 &
  !is.na(raw_data_cleaned$Q4_1_1) &
  !is.na(raw_data_cleaned$Q4_1_2) &
  !is.na(raw_data_cleaned$Q4_1_3) &
  !is.na(raw_data_cleaned$Q4_2_1) &
  !is.na(raw_data_cleaned$Q4_2_2) &
  !is.na(raw_data_cleaned$Q4_2_3)
)

victim_knowsvict <- sum(
  raw_data_cleaned$Q4_1_1 == 0 &
  raw_data_cleaned$Q4_1_2 == 0 &
  raw_data_cleaned$Q4_1_3 == 0 &
  raw_data_cleaned$Q4_2_1 == 1 &
  raw_data_cleaned$Q4_2_2 == 1 &
  raw_data_cleaned$Q4_2_3 == 0 &
  !is.na(raw_data_cleaned$Q4_1_1) &
  !is.na(raw_data_cleaned$Q4_1_2) &
  !is.na(raw_data_cleaned$Q4_1_3) &
  !is.na(raw_data_cleaned$Q4_2_1) &
  !is.na(raw_data_cleaned$Q4_2_2) &
  !is.na(raw_data_cleaned$Q4_2_3)
)

knowsvict_only <- sum(
```

```
  raw_data_cleaned$Q4_1_1 == 0 &
  raw_data_cleaned$Q4_1_2 == 0 &
  raw_data_cleaned$Q4_1_3 == 0 &
  raw_data_cleaned$Q4_2_1 == 1 &
  raw_data_cleaned$Q4_2_2 == 0 &
  raw_data_cleaned$Q4_2_3 == 0 &
  !is.na(raw_data_cleaned$Q4_1_1) &
  !is.na(raw_data_cleaned$Q4_1_2) &
  !is.na(raw_data_cleaned$Q4_1_3) &
  !is.na(raw_data_cleaned$Q4_2_1) &
  !is.na(raw_data_cleaned$Q4_2_2) &
  !is.na(raw_data_cleaned$Q4_2_3)
)

#data frame
contact_counts <- tibble::tibble(
  total_incarcerated = total_incarcerated,
  total_knows_incarcerated = total_knows_incarcerated,
  total_no_incarcerated = total_no_incarcerated,
  total_victim = total_victim,
  total_knows_victim = total_knows_victim,
  total_no_victim = total_no_victim,
  incarcerated_only = incarcerated_only,
  incarcerated_knowsinc = incarcerated_knowsinc,
  inc_knowsinc_victim = inc_knowsinc_victim,
  inc_knowsinc_knowsvict = inc_knowsinc_knowsvict,
  inc_knowsinc_victim_knowsvict = inc_knowsinc_victim_knowsvict,
  incarcerated_victim = incarcerated_victim,
  inc_victim_knowsvict = inc_victim_knowsvict,
  incarcerated_knowsvict = incarcerated_knowsvict,
  knowsinc_only = knowsinc_only,
  knowsinc_victim = knowsinc_victim,
  knowsinc_knowsvict = knowsinc_knowsvict,
  knowsinc_victim_knowsvict = knowsinc_victim_knowsvict,
  vict_only = vict_only,
  victim_knowsvict = victim_knowsvict,
  knowsvict_only = knowsvict_only
)

write_csv(contact_counts, "filled_contact_combo_counts.csv")


#view(contact_counts)

```

FLAGGED FOR DELETION Another Way for Impacted Counts

```{r}
#| eval: false
#| include: false

getwd()
impacted <- read.csv("/Users/kristinamensik/Library/Mobile
Documents/com~apple~CloudDocs/Documents/Duke/2024-2025/902 Legislators Pre-Ananlysis
Plan/Prelim Folder/Prelim 3-23/data/impacted.csv")

view(impacted)

# Count unique combinations
impacted_summary <- impacted %>%
  select(been_inc, know_inc, none_inc, been_vict, know_vict, no_vict) %>%
  group_by(across(everything())) %>%
  summarise(count = n(), .groups = "drop") %>%
  arrange(desc(count))
```

```
# View the result
view(impacted_summary)

#only incarcerated
sum(
  impacted$been_inc == 1 &
  impacted$no_vict == 1 &
  impacted$know_inc == 0 &
  impacted$none_inc == 0 &
  impacted$been_vict == 0 &
  impacted$know_vict == 0
)

#incarcerated & Victimized
sum(
  impacted$been_inc == 1 &
  impacted$no_vict == 0 &
  impacted$know_inc == 0 &
  impacted$none_inc == 0 &
  impacted$been_vict == 1 &
  impacted$know_vict == 0
)

#incarcerated & Victimized & Know Inc
sum(
  impacted$been_inc == 1 &
  impacted$no_vict == 0 &
  impacted$know_inc == 1 &
  impacted$none_inc == 0 &
  impacted$been_vict == 1 &
  impacted$know_vict == 0
)

#only know incarcerated
sum(
  impacted$been_inc == 0 &
  impacted$no_vict == 1 &
  impacted$know_inc == 1 &
  impacted$none_inc == 0 &
  impacted$been_vict == 0 &
  impacted$know_vict == 0
)

#only vict
sum(
  impacted$been_inc == 0 &
  impacted$no_vict == 0 &
  impacted$know_inc == 0 &
  impacted$none_inc == 1 &
  impacted$been_vict == 1 &
  impacted$know_vict == 0
)

#only know vict
sum(
  impacted$been_inc == 0 &
  impacted$no_vict == 0 &
  impacted$know_inc == 0 &
  impacted$none_inc == 1 &
  impacted$been_vict == 0 &
  impacted$know_vict == 1
)

#know vict, know incarc, been vict
```

```
sum(
  impacted$been_inc == 0 &
  impacted$no_vict == 0 &
  impacted$know_inc == 1 &
  impacted$none_inc == 0 &
  impacted$been_vict == 1 &
  impacted$know_vict == 1
)


```

END FLAG

Tables

```{r}
combo_summary_labeled %>%
  select(label, contact_pattern, N, pct_total) %>%
  arrange(desc(N)) %>%
  gt() %>%
  cols_label(
    contact_pattern = "Q4 Pattern",
    label = "Description",
    N = "N",
    pct_total = "% of Sample"
  ) %>%
  fmt_number(columns = where(is.numeric), decimals = 1) %>%
  tab_header(title = "Appendix Table A. Full Distribution of Q4 Contact Combinations") %>%
  tab_options(table.font.size = "small")


print(combo_summary_labeled)f
```

C. Q6, 7, 8 - Ration Block

```{r}

view(raw_data_cleaned)

#inspect

#raw_data %>% select(q6, q7, q8) %>% summarise_all(~ list(unique(.)))

# clean and transform
raw_data_cleaned <- raw_data_cleaned %>%
  mutate(across(
    c(Q6, Q7, Q8),
    ~ na_if(as.numeric(.x), -99)
    )) %>% mutate(across(
    c(Q6, Q7, Q8),
    ~ factor(.x, levels = 1:4,
             labels = c("Frequently", "Sometimes", "Rarely", "Never"),
             ordered = TRUE)
  ))

 raw_data_cleaned %>% count(Q8)

```

D. Prison & Election Questions

```{r, punishorrehab}
# Renamed raw column to a more intuitive name
```

```
#raw_data_cleaned <- raw_data_cleaned %>%
#  rename(PenalPunitiveness = Q62_1)

#cleaned and reverse coded already

# Create ordered factor version of the reverse-coded scale (7 = most punitive)
raw_data_cleaned <- raw_data_cleaned %>%
  mutate(penalpunitiveness_ord = factor(
    PenalPunitiveness,
    levels = 1:7,
    labels = c(
      "Rehab (1)", "Rehab (2)", "Rehab (3)", "Neutral (4)",
      "Punish (5)", "Punish (6)", "Punish (7)"
    ),
    ordered = TRUE
  ))

# Factor version of PenalPunitiveness (reverse-coded already: 1 = least punitive, 7 = most
punitive)


```

```{r, prisonpen}
#Q25_1 and Q25_5, death penalty and LWOP
 #raw_data_cleaned %>% count(Q25_1)

#recode -99
raw_data_cleaned <- raw_data_cleaned %>%
  mutate(Q25_1 = na_if(as.numeric(Q25_1), -99))

#factor version
raw_data_cleaned <- raw_data_cleaned %>%
  mutate(Q25_1_ord = factor(
    Q25_1,
    levels = 1:5,
    labels = c("Strongly favor", "Favor", "Oppose", "Strongly oppose", "Don't know"),
    ordered = TRUE
  ))

# Recode Q25_5 the same way
raw_data_cleaned <- raw_data_cleaned %>%
  mutate(Q25_5 = na_if(as.numeric(Q25_5), -99))

# Code prisonpen subitems, NA if "Don't know" (5)
# Create cleaned subcomponents
raw_data_cleaned <- raw_data_cleaned %>%
  mutate(
    prisonpen_1 = case_when(Q25_1 %in% 1:4 ~ Q25_1, TRUE ~ NA_integer_),
    prisonpen_5 = case_when(Q25_5 %in% 1:4 ~ Q25_5, TRUE ~ NA_integer_)
  )

# Compute index (after subcomponents exist)
raw_data_cleaned <- raw_data_cleaned %>%
  mutate(prisonpen = rowMeans(select(., prisonpen_1, prisonpen_5), na.rm = TRUE))

```

```{r, prisonhelp}

# Define Q25 help-oriented items
q25_help_items <- c("Q25_2", "Q25_3", "Q25_4", "Q25_6")

# Recode -99 to NA
raw_data_cleaned <- raw_data_cleaned %>%
```

```
  mutate(across(all_of(q25_help_items), ~ na_if(as.numeric(.), -99)))


# Create ordered factor versions
raw_data_cleaned <- raw_data_cleaned %>%
  mutate(across(
    all_of(q25_help_items),
    ~ factor(.,
             levels = 1:5,
             labels = c("Strongly favor", "Favor", "Oppose", "Strongly oppose", "Don't
know"),
             ordered = TRUE),
    .names = "{.col}_ord"
  ))

# Create reverse-coded numeric versions
raw_data_cleaned <- raw_data_cleaned %>%
  mutate(across(
    all_of(q25_help_items),
    ~ ifelse(. %in% 1:4, 5 - ., .),
    .names = "{.col}_rev"
  ))

view(raw_data_cleaned)


#election help questions
# Clean and convert to numeric
raw_data_cleaned <- raw_data_cleaned %>%
  mutate(across(c(Q23_2, Q23_7), ~ na_if(as.numeric(.x), -99))
         )

# Create ordered factor versions
raw_data_cleaned <- raw_data_cleaned %>%
  mutate(
    Q23_2_ord = factor(Q23_2, levels = 1:3,
                       labels = c("Favor", "Oppose", "Don't know"),
                       ordered = TRUE),
    Q23_7_ord = factor(Q23_7, levels = 1:3,
                       labels = c("Favor", "Oppose", "Don't know"),
                       ordered = TRUE)
  )

# Create reverse-coded numeric versions (Favor = 2, Oppose = 1, Don't know = NA or 0)
raw_data_cleaned <- raw_data_cleaned %>%
  mutate(
    Q23_2_rev = case_when(
      Q23_2 == 1 ~ 2,  # Favor
      Q23_2 == 2 ~ 1,  # Oppose
      Q23_2 == 3 ~ NA_real_,  # Don't know
      TRUE ~ NA_real_
    ),
    Q23_7_rev = case_when(
      Q23_7 == 1 ~ 2,
      Q23_7 == 2 ~ 1,
      Q23_7 == 3 ~ NA_real_,
      TRUE ~ NA_real_
    )
  )



# prisonhelp index
# Construct composite index (average of 6 items)
raw_data_cleaned <- raw_data_cleaned %>%
```

```
  mutate(prisonhelp = rowMeans(select(.,
    Q25_2_rev, Q25_3_rev, Q25_4_rev, Q25_6_rev, Q23_2_rev, Q23_7_rev
  ), na.rm = TRUE))

```

####D1. Variance and Stand Dev

```{r}

help_policy_items <- c("Q25_2", "Q25_3", "Q25_4", "Q25_6", "Q23_2", "Q23_7")

# Variance and SD across help policy items
raw_data_cleaned <- raw_data_cleaned %>%
  rowwise() %>%
  mutate(
    variancehelp = var(c_across(all_of(help_policy_items)), na.rm = TRUE),
    sdhelp = sd(c_across(all_of(help_policy_items)), na.rm = TRUE)
  ) %>%
  ungroup()


```

###E. Policy Orientation (polor) control

```{r}
#| eval: false
#| include: false

help_policy_items <- c("Q25_2", "Q25_3", "Q25_4", "Q25_6", "Q23_2", "Q23_7")

# Variance and SD across help policy items
raw_data_cleaned <- raw_data_cleaned %>%
  rowwise() %>%
  mutate(
    variancehelp = var(c_across(all_of(help_policy_items)), na.rm = TRUE),
    sdhelp = sd(c_across(all_of(help_policy_items)), na.rm = TRUE)
  ) %>%
  ungroup()


```

###F. FIRE controls

```{r}
#| eval: false
#| include: false

#raw_data_cleaned %>% count(Q32)

#raw_data_cleaned %>% count(Q33)

#raw_data_cleaned %>% count(Q34)

#raw_data_cleaned %>% count(Q111)

fire_vars <- c("Q32", "Q33", "Q34", "Q111")



raw_data_cleaned <- raw_data_cleaned %>%
  mutate(across(
    all_of(fire_vars),
```

```
    ~ as.numeric(na_if(as.character(.x), "-99"))
  ))

#raw_data_cleaned <- raw_data_cleaned %>%
#  rename(
#    fire_rare = Q32,
#    fire_privilege = Q33,
#    fire_angry = Q34,
#    fire_fear = Q111
 # )

```

###G. Race

```{r}
#| eval: false
#| include: false

#convert + clean
race_vars <- paste0("Q51_", 1:8)

raw_data_cleaned <- raw_data_cleaned %>%
  mutate(across(all_of(race_vars), ~ na_if(as.numeric(.x), -99)))

raw_data_cleaned <- raw_data_cleaned %>%
  mutate(
    Black = ifelse(Q51_2 == 1 & is.na(Q51_7), 1, 0),
    Hispanic = ifelse(Q51_7 == 1, 1, 0),
    White = ifelse(Q51_1 == 1 & is.na(Q51_7), 1, 0)
  ) %>%
  mutate(
    OtherRace = ifelse(
      (Black + Hispanic + White) == 0 & rowSums(select(., all_of(race_vars)), na.rm =
TRUE) > 0,
      1, 0
    )
  )



```

###H. Party + Ideology controls

```{r}

#raw_data_cleaned %>% count(Q47)

#clean dem 101
raw_data_cleaned <- raw_data_cleaned %>%
  mutate(dem101 = na_if(as.numeric(dem101), -99))

#party ID dummies
raw_data_cleaned <- raw_data_cleaned %>%
  mutate(
    Republican = ifelse(dem101 == 1, 1, 0),
    Democrat = ifelse(dem101 == 2, 1, 0)
  )


#clean ideology
raw_data_cleaned <- raw_data_cleaned %>%
  mutate(
    Q46 = na_if(as.numeric(Q46), -99),
```

```
    Q47 = na_if(as.numeric(Q47), -99)
  )

raw_data_cleaned <- raw_data_cleaned %>%
  mutate(
    fiscal_ideology = ifelse(!is.na(Q46), 6 - Q46, NA),
    social_ideology = ifelse(!is.na(Q47), 6 - Q47, NA)
  )


```

###I. Gender

```{r, gender}
#raw_data_cleaned %>% count(Q55)

#trans clean

raw_data_cleaned <- raw_data_cleaned %>%
  mutate(Q55 = na_if(as.numeric(Q55), -99))

#trans indicator
raw_data_cleaned <- raw_data_cleaned %>%
  mutate(trans_id = case_when(
    Q55 == 1 ~ 1,
    Q55 %in% c(2, 3) ~ 0,  # Conservative approach: counts only self-identified trans as 1
    TRUE ~ NA_real_
  ))


#gender clean
gender_vars <- paste0("Q56_", 1:4)

raw_data_cleaned <- raw_data_cleaned %>%
  mutate(across(all_of(gender_vars), ~ na_if(as.numeric(.x), -99)))


#constructing the gender binary
raw_data_cleaned <- raw_data_cleaned %>%
  mutate(gender = case_when(
    Q56_2 == 1 ~ 1,  # Selected man
    Q56_1 == 1 & (Q56_2 != 1 | is.na(Q56_2)) ~ 0,  # Woman only or with genderqueer/other
    Q56_3 == 1 | Q56_4 == 1 ~ 0,  # Genderqueer or other, no man selected
    TRUE ~ NA_real_
  ))



```

###. LAST OF CLEANING - WRITE CSV

```{r}
write_csv(raw_data_cleaned, "data/cleaned/cleaned_data_may8.csv")
```

3. Preliminary Analyses

Decision	Notes
which version of table?	prisonerdeserves (block aware and cleaned)
NAs?	drop silently for plot but log
scale?	raw 0-100 for now
compare subgroups?	for now, overall distribution of just incppl

summary overlays?	mean line, kernal density, histogram bars
plot aesthetics?	clear x-axis label
Output	

1.1. Main DV: Deservingness of Incarcerated People/Prisoners

```{r, prechecks}
#check counts: raw_data_cleaned %>% count(d_incppl_pris_abc)

#summary:
summary(raw_data_cleaned$d_incppl_pris_abc)

#histogram
hist(raw_data_cleaned$d_incppl_pris_abc, breaks = 20,
     main = "Deservingness of Incarcerated People/Prisoners",
     xlab = "Deservingness (0–100)", col = "gray80", border = "white")


mean(raw_data_cleaned$d_incppl_pris_abc, na.rm = TRUE)

sd(raw_data_cleaned$d_incppl_pris_abc, na.rm = TRUE)


```

1.2. Control: Generalized Deservingness (desor)

```{r}

# Histogram for generalized deservingness control variable
hist(raw_data_cleaned$desor, breaks = 20,
     main = "Generalized Deservingness",
     xlab = "Average Deservingness of Help-Target Groups (0–100)",
     col = "gray80", border = "white")

summary(raw_data_cleaned$desor)
sd(raw_data_cleaned$desor, na.rm = TRUE)

```

2. Cross-Tabs of Contact Types

```{r}
library(dplyr)
library(gt)
library(stringr)

# Optional: define custom labels for key combinations
pattern_labels <- tibble::tibble(
  contact_pattern = c(
    "001001", "000000", "100000", "010000", "100100",
    "010100", "110110", "110100", "011010", "001100"
  ),
  label = c(
    "No or unsure only",
    "Skipped all Q4 items",
    "Been incarcerated only",
    "Knows someone incarcerated only",
    "Been incarcerated & victimized",
    "Knows incarcerated & victimized",
    "All four: incarcerated, knows, victim, knows victim",
    "Incarcerated, knows, victim",
    "Knows incarcerated & knows victim",
    "Victimized only"
  )
```

```
)

# Join with summary table
combo_summary_labeled <- combo_summary %>%
  left_join(pattern_labels, by = "contact_pattern") %>%
  mutate(label = ifelse(is.na(label), "Other", label))
```

A. All Contact Combos

pull from 2B

```{r}

```

3. Regression Model 1: Predicting deservingness

####3.1 Without Controls

```{r}
m1 <- lm(d_incppl_pris_abc ~ impact, data = raw_data_cleaned)
summary(m1)

```

M1 Interpretation:

- Q: does any carceral contact or victimization change perceived deservingness of incarcerated people (`d_incppl_pris_abc`, 0–100 scale) compared to those without any such contact (`impact = 0`).

- Coefficients:

    - Intercept = 43.31

        - The average deservingness score of incarcerated people, for respondents with no contact (`impact = 0`), is on average 43.31 out of 100 on the deservingness scale. This is higher than I might have thought.

    - impact = 7.17

        - People who have had contact (direct, indirect, or victimization) rate incarcerated people 7.17 points higher on the 0–100 deservingness scale – on average – than those without contact.

        - Significance: The t-val is 5.881 and p is less than 0.001, suggesting this effect is significantly significant and the probability that it is due to chance is very low.

        - Model fit: The R-squared is 0.0156, meaning only 1.6% of the variation in deservingness is explained by this binary variable alone.

```{r, m1plot}
library(ggplot2)
# Summarize means and confidence intervals
plot_data <- raw_data_cleaned %>%
  filter(!is.na(impact)) %>%  # <- filter out NA values
  group_by(impact) %>%
  summarize(
    mean_deserving = mean(d_incppl_pris_abc, na.rm = TRUE),
    se = sd(d_incppl_pris_abc, na.rm = TRUE) / sqrt(n()),
    .groups = "drop"
  ) %>%
  mutate(
```

```
    impact_label = ifelse(impact == 1, "Any Contact", "No Contact"),
    ci_lower = mean_deserving - 1.96 * se,
    ci_upper = mean_deserving + 1.96 * se
  )

# Plot
ggplot(plot_data, aes(x = impact_label, y = mean_deserving)) +
  geom_col(fill = "gray60") +
  geom_errorbar(aes(ymin = ci_lower, ymax = ci_upper), width = 0.2) +
  labs(
    x = NULL,
    y = "Mean Deservingness Rating (0–100)",
    title = "Deservingness of Incarcerated People by Contact Experience"
  ) +
  theme_minimal(base_size = 14)


```

####3.2 With Controls

```{r}
m2 <- lm(d_incppl_pris_abc ~ impact +
    Republican + Democrat +
    fiscal_ideology + social_ideology +
    PenalPunitiveness + desor +
    fire_rare + fire_privilege + fire_angry + fire_fear +
    gender + Black + Hispanic + OtherRace,
  data = raw_data_cleaned
)
summary(m2)
```

M2 Interpretation:

- Coefficients:

    - Intercept = 0.63 (p = 0.91)

        - The predicted deservingness score when all predictors are 0, so not particularly meaningful. Means respondent:
            - is independent or other
            - has the most conservative ideology
            - ...is not impacted
            - ...(add remaining)

    - impact = 4.70 (p \< 0.001

        - Being impacted by the CJ system (incarcerated, knows, victimized) increases the predicted deservingness rating by 4.7 points on average, holding all else constant.

            - Remains statistically significant, particularly because the outcome range is 0–100

            - supports H1

    - Republican = 0.39 (p = 0.81)

        - Compared to Independents/Others, Republicans rate incarcerated people just 0.39 points higher, and this is not statistically significant.

        - **No evidence** that Republican identification alone predicts harsher or more generous deservingness scores in this model (when other factors are controlled).

        - May reflect that **ideology and punitiveness** absorb more of the variance

that would otherwise be attributed to party label.

- Democrat = −0.49 (p = 0.72)
    - Compared to Independents/Others, Democrats score incarcerated people 0.49 points lower on deservingness, but this is not statistically significant.
    - There is **no reliable difference** in prisoner deservingness between Democrats and Independents/Others once we account for ideology, contact, and other controls.
    - Coupled with the Republican coefficient, this suggests **very weak direct effects of party ID per se**, which may be surprising given elite cues—but again, could reflect ideological and experiential absorption.
- fiscal_ideology = 0.17 (p = 0.84)
    - This is the effect of being more fiscally conservative (scale reversed: higher = more conservative).
    - Not statistically significant, and the coefficient is small.
    - **Fiscal ideology appears unrelated** to how respondents rate the deservingness of incarcerated people, net of party, social ideology, punitiveness, and racial attitudes.
    - This aligns with your expectation that deservingness is more influenced by *social/cultural* orientations than by economic/fiscal ideology.
- social_ideology = −1.49 (0.072)
    - each unit increase in social conservativism predicts a 1.49 point decrease in deservingness of incarcerated people (on a 0–100 scale)
    - marginally significant (p\~0.07) and in the expected direction
    - some evidence social conservativism is associated with lower perceived deservingness
    - effect size is moderate – but this all goes into the pile of "do further testing to see which controls are meaningful
- PenalPunitiveness = −1.14 (p \<0.001)
    - participant view on whether purpose of incarceration is to punish or rehabilitate.
    - a one unit increase in belief that incarceration is primarily about punishment is associated with a 1.14–point decrease in the perceived deservingness of incarcerated people.
    - **substantive significance:** a full–scale shift (from 1 to 7) predicts a 6.8 point drop in deservingness on a 0–100 scale – modest but meaningful, especially in the context of other predictors.
    - this supports the claim that punitive orientations towards the purpose of incarceration shape how people evaluate the deservingness of targets.
    - **modeling implication**: it's not entirely endogenous to deservingness ratings, bc it comes from a distinct question about the penal telos. I think this tells us that deservingness ratings don't just reflect beliefs about what prison is for.
        - However, I'm not sure this variable captures "punitivenness." In part, it bleeds into my questions about affective reactions versus judgement.

- desor = +0.76 (p \< 0.001)
    - Interpretation:
        - For each one-point increase in the average deservingness rating of *help-target groups*, perceived deservingness of incarcerated people increases by 0.76
    - Substantive scale:
        - ranges from 0-100, so a 10-pt increase corresponds to a 7.6 point increase in prisoner deservingness — comperable to but in opp direction of penalpun effect
    - conceptual interpretation:
        - captures individual-level variation in generalized generosity twoard marginalized/help needing populations.
        - inclusion seems important because it helps isolate target-specific effects, shows that prisoner deservingness is partially embedded in broader affective/moral schemas about who deserves help.
    - Modeling implication: controlling for desor might let me argue that the effect of contact on prisoner deservingness isn't just a byproduct of being generally generous — it's targeted.
- fire_rare = 0.39734 (p = 0.60)
    - agreement that racial problems are rare and isolated, so more rejection of racial minimization (racial empathy) predicts lightly higher deservingness perceptions of incarcerated people.
    - takeaway — not statistically significant, so can't rule out the possibility the relationship is due to chance. But direction aligns with theory.
- fire_privilege = 0.05353 (p = 0.94)
    - Meaning: Agreement that white people benefit from unearned racial advantages
    - Interpretation: The near-zero, nonsignificant coefficient indicates this belief does not meaningfully predict differences in perceived deservingness of incarcerated people in this model.
    - Note: This item may be less emotionally activated than the others or more subject to social desirability bias, dampening its predictive power
- fire_angry = 1.18110 (p = 0.16)
    - Meaning: Agreement with *“I am angry that racism exists.”*
    - Interpretation: The positive direction suggests that respondents who are more emotionally angry about racism tend to see incarcerated people as more deserving, consistent with frameworks linking racial empathy to support for equity-enhancing policies.
        - Note: While not significant, this coefficient is substantively large and may become more stable with a larger sample or better scaling.
- fire_fear (0.30112) (P = 0.73)
    - Meaning: Agreement with *“I am fearful of people of other races.”*
    - Interpretation: Respondents who express more racial fear are (slightly) more likely to rate incarcerated people as less deserving, though the effect is small and nonsignificant

        - This aligns with theories linking racial threat or fear to punitive or exclusionary attitudes.

    - gender = -0.3833 (p=0.74)

        - men estimated to rate incarcerated people as slightly less deserving, but difference is small and not significant

    - Black = DROPPED! colinearity? (p=)

    - Hispanic = -3.03639 (p= 0.057)

        - rate incarcerated people 3 points less deserving than non-hispanic whites, holding all else constant

    - OtherRace = dropped

####3.3 With Weights

```{r}
# weights don't exist yet
# m1_w <- lm(prisonerdeserves ~ impact, data = raw_data_cleaned, weights = weight_var)
# summary(m1_w)
```