

NeuroMex Analysis

Hayden Mountcastle

2025-01-30

Purpose

The purpose of this document is to run analysis for the NeuroMEX CIDI/LEC paper/ The main aim is to determine whether exposure to interpersonal and/or noninterpersonal traumas impact chronic condition outcomes.

We use NeuroMex Freeze 6 data, located in the Koenen Lab dropbox: /Dropbox (Harvard University)/NeuroMex/6. Data Freezes/Data Freeze 6

The latest models and figures start at: **Regression update 1-13-25**

Libraries

```
library(tidyverse)
library(ggeasy)
library(table1)
library(confintr)
library(lme4)
library(gtsummary)
library(kableExtra)
library(sandwich)
library(lmtest)
library(AER)
library(MASS)
```

Import data

We are using the latest NeuroMex Datafreeze 6

```
setwd("~/Dropbox (Harvard University)/NeuroMex/6. Data Freezes/Data Freeze 6")
nmex <- read.csv("NeuroMex_Freeze6_Draft1.csv")
```

C. Cleaning Data

Removing participants

We remove participants who did not complete UBACC or left the interview prematurely.

```

#These participants are missing the UBACC
nmex <- nmex %>% filter(!(partid %in%
                        c("CAP5649381", "CAP9216055",
                          "GEA7332459", "QUP2253899", "QUP5901822",
                          "QUP6859222")))

#These participants are missing all or 50% of the CIDI,
#will exclude for simplicity

nmex <- nmex %>% filter(!(partid %in%
                        c("INP2366191", "INP4404950",
                          "JAP1320792", "INP1389432")))

```

Demographics

Cleaning basic variables

```

#Factorize is_case and is_male

nmex$is_case_f <- factor(nmex$is_case)
nmex$is_case_f <- fct_recode(nmex$is_case_f, "Control" = "0", "Case" = "1")
nmex$is_case_f <- factor(nmex$is_case_f, c("Case", "Control"))

kable(table(nmex$is_case_f), col.names = c("Case Status", "N"), caption = "Frequency of Case Categories")

```

Case status

Table 1: Frequency of Case Categories

Case Status	N
Case	4675
Control	4611

```

nmex$is_male_f <- factor(nmex$is_male)
nmex$is_male_f <- fct_recode(nmex$is_male_f, "Male" = "1", "Female" = "0")
nmex$is_male_f <- factor(nmex$is_male_f, c("Male", "Female"))

kable(table(nmex$is_male_f), col.names = c("Sex", "N"), caption = "Frequency of Case Categories")

```

Sex

Table 2: Frequency of Case Categories

Sex	N
Male	4545

Sex	N
Female	4741

SES status In our 12-11-24 meeting, we decided to use Low-Middle Low and Middle-High classifications.

```
nmex <- nmex %>%
  mutate(ses_status_f = factor(case_when(
    ses_status %in% c(1,2) ~ "Low-Middle Low",
    ses_status %in% c(3,4, 5) ~ "Middle-High",
    TRUE ~ NA_character_
  ), levels = c("Low-Middle Low", "Middle-High")))

nmex <- nmex %>%
  mutate(ses_status_opt2_f = factor(case_when(
    ses_status == 1 ~ "Low",
    ses_status == 2 ~ "Middle Low",
    ses_status %in% c(3,4, 5) ~ "Middle-High",
    TRUE ~ NA_character_
  ), levels = c("Low", "Middle Low", "Middle-High")))

kable(table(nmex$ses_status), col.names = c("SES", "N"), caption = "Frequency of SES Status")
```

Table 3: Frequency of SES Status

SES	N
1	2281
2	2839
3	519
4	3418
5	56
777	158
888	14

```
kable(table(nmex$ses_status_f), col.names = c("SES Recoded", "N"), caption = "Frequency of SES Status Recoded")
```

Table 4: Frequency of SES Status Recoded

SES Recoded	N
Low-Middle Low	5120
Middle-High	3993

#Age Categorical

```
nmex <- nmex %>% mutate(age_cat = case_when(
```

```

age_at_interview < 30 ~ "18-29",
age_at_interview >= 30 & age_at_interview < 40 ~ "30-39",
age_at_interview >= 40 & age_at_interview < 50 ~ "40-49",
age_at_interview >= 50 & age_at_interview < 60 ~ "50-59",
age_at_interview >= 60 ~ "60+"

))

kable(table(nmex$age_cat), col.names = c("Age", "N"), caption = "Frequency of Age Cat.")

```

Age

Table 5: Frequency of Age Cat.

Age	N
18-29	2077
30-39	2191
40-49	2247
50-59	1911
60+	860

Location In our 12-11-24 meeting, we decided to use urbanicity (site_location_group_f)

```

#Factorize site location

nmex$site_location_f <- factor(nmex$site_location)
nmex$site_location_f <- fct_recode(nmex$site_location_f,
                                   "CDMX" = "1",
                                   "Campeche" = "2",
                                   "Queretaro" = "3",
                                   "Leon" = "4",
                                   "Guadalajara" = "5",
                                   "Morelia" = "6")

kable(table(nmex$site_location_f), col.names = c("Site", "N"), caption = "Frequency of Site")

```

Table 6: Frequency of Site

Site	N
CDMX	5507
Campeche	132
Queretaro	608
Leon	392
Guadalajara	1917
Morelia	730

```

#Factorize site location

nmex <- nmex %>% mutate(

```

```

site_location_group_f = case_when(
  site_location %in% c(1,3,5) ~ "Urban",
  site_location %in% c(2,4,6) ~ "Rural"
))

kable(table(nmex$site_location_group_f), col.names = c("Urbanicity (Site Recoded)", "N"), caption = "Fr

```

Table 7: Frequency of Urbanicity (Site Recoded)

Urbanicity (Site Recoded)	N
Rural	1254
Urban	8032

```

nmex <- nmex %>% mutate(edu_ord = case_when(
  education_gradepassed == 0 ~ "None",
  education_gradepassed == 8 ~ "Literate",
  education_gradepassed %in% c(1,2) ~ "Elementary or middle schol",
  education_gradepassed %in% c(3,4) ~ "High school or technical school",
  education_gradepassed %in% c(5,6,7) ~ "Post secondary school",
  TRUE ~ NA_character_
))

kable(table(nmex$education_gradepassed), col.names = c("Education", "N"), caption = "Frequency of Educa

```

Education

Table 8: Frequency of Education

Education	N
0	112
1	948
2	2071
3	2172
4	839
5	2331
6	449
7	97
8	267

```

kable(table(nmex$edu_ord), col.names = c("Education (Recoded)", "N"), caption = "Frequency of Education

```

Table 9: Frequency of Education Recoded

Education (Recoded)	N
Elementary or middle schol	3019
High school or technical school	3011

Education (Recoded)	N
Literate	267
None	112
Post secondary school	2877

```
nmex <- nmex %>% mutate(liv_arrange_ord = case_when(
  liv_arrange == 1 ~ "Lives alone",
  liv_arrange %in% c(2, 4) ~ "Lives with parents or relatives",
  liv_arrange == 3 ~ "Lives with nuclear family",
  liv_arrange == 5 ~ "Lives with friends",
  liv_arrange %in% c(6,7,999) ~ "Other",
  liv_arrange == 888 ~ "Does not want to respond",
  TRUE ~ NA_character_
))

kable(table(nmex$liv_arrange), col.names = c("Living Arrangement", "N"), caption = "Frequency of Living
```

Living arrangements

Table 10: Frequency of Living Arrangement

Living Arrangement	N
1	827
2	4337
3	3360
4	292
5	160
6	45
7	167
888	46
999	52

```
kable(table(nmex$liv_arrange_ord), col.names = c("Living Arrangement (Recoded)", "N"), caption = "Freque
```

Table 11: Frequency of Living Arrangement Recoded

Living Arrangement (Recoded)	N
Does not want to respond	46
Lives alone	827
Lives with friends	160
Lives with nuclear family	3360
Lives with parents or relatives	4629
Other	264

```
nmex <- nmex %>% mutate(civil_status_ord = case_when(
  civil_status == 1 ~ "Single",
  civil_status %in% c(2,3) ~ "Married/Open Union",
  civil_status == 4 ~ "Divorced/Separated",
  civil_status == 5 ~ "Widowed",
  civil_status %in% c(777,888) ~ "Does not know/Does not want to respond",
  TRUE ~ NA_character_
))

kable(table(nmex$civil_status), col.names = c("Civil status", "N"), caption = "Frequency of Civil Status")
```

Civil status

Table 12: Frequency of Civil Status

Civil status	N
1	5274
2	2149
3	809
4	834
5	219
999	1

```
kable(table(nmex$civil_status_ord), col.names = c("Civil status (Recoded)", "N"), caption = "Frequency of Civil Status Recoded")
```

Table 13: Frequency of Civil Status Recoded

Civil status (Recoded)	N
Divorced/Separated	834
Married/Open Union	2958
Single	5274
Widowed	219

LEC

Trauma Loads Here, we code for interpersonal trauma, noninterpersonal trauma. We take the add of each trauma (binary) for “happened to me only”, which will give a sum of traumas experienced.

We ignore violent death and accidental death as these can not “happen to me”

```
#Interpersonal Trauma Load

nmex <- nmex %>% mutate(
  trauma_load_interpersonal_1 = rowSums(dplyr::select(.,
    phys_assault_1,
    weapon_assault_1,
```

```

sexual_assault_1,
other_sexual_1,
captivity_1,
harm_else_1),
na.rm=TRUE),

#removing human suffering
trauma_load_noninter_1_nhs = rowSums(dplyr::select(.,
  natural_disaster_1,
  fire_exp_1,
  transp_accident_1,
  serious_accident_1,
  toxic_sub_1,
  combat_1,
  illness_injury_1,
  #human_suffering_1,
  # violent_death_2,
  # accidental_death_2
), na.rm=TRUE),

trauma_load_any_1 = rowSums(dplyr::select(.,
  phys_assault_1,
  weapon_assault_1,
  sexual_assault_1,
  other_sexual_1,
  captivity_1,
  harm_else_1,
  natural_disaster_1,
  fire_exp_1,
  transp_accident_1,
  serious_accident_1,
  toxic_sub_1,
  combat_1,
  illness_injury_1),
na.rm=TRUE))

```

Trauma categorical Here we used the previously made sums of trauma to create a categorical variable of traumas experienced.

```

nmex <- nmex %>% mutate(
  trauma_load_inter_cat = factor(case_when(
    trauma_load_interpersonal_1 == 0 ~ "0",
    trauma_load_interpersonal_1 == 1 ~ "1",
    trauma_load_interpersonal_1 == 2 ~ "2",
    trauma_load_interpersonal_1 >= 3 ~ ">=3",
    TRUE ~ NA_character_
  ), levels=c("0", "1", "2", ">=3")),

  trauma_load_noninter_nhs_cat = factor(case_when(
    trauma_load_noninter_1_nhs == 0 ~ "0",
    trauma_load_noninter_1_nhs == 1 ~ "1",
    trauma_load_noninter_1_nhs == 2 ~ "2",

```



```

trauma_load_noninter_1_nhs >= 3 ~ ">=3",
TRUE ~ NA_character_),levels=c("0","1","2",">=3"))
)

kable(table(nmex$trauma_load_inter_cat), col.names = c("Trauma Load IT Cat.", "N"), caption = "Frequency")

```

Table 14: Frequency Interpersonal Trauma Load

Trauma Load IT Cat.	N
0	4420
1	2295
2	1510
>=3	1061

```

kable(table(nmex$trauma_load_noninter_nhs_cat), col.names = c("Trauma Load NIT Cat.", "N"), caption = "Frequency")

```

Table 15: Frequency Noninterpersonal Trauma Load

Trauma Load NIT Cat.	N
0	4311
1	2724
2	1394
>=3	857

CIDI

Here we make a binary variable for all CIDI variables (1-18). If a participant responded 0 or 777 (unknown) then they are coded as 0, if they responded 1, then they are coded as 1. If the data is NA, it stays NA.

```

#Clean CIDI variables

#Create a list of CIDI variables 1-18
cidi_q <- paste0("cidi_q", seq(1,18))

#Chronic conditions load

#Make all variables 0,1,NA
for (q in cidi_q) {
  new_col <- paste0(q, "_binary")

  nmex <- nmex %>%
    mutate(!new_col := case_when(
      .data[[q]] %in% c(0,777) ~ 0,
      .data[[q]] == 1 ~ 1,
      TRUE ~ NA_real_
    ))
}

```

```

tab <- kable(table(nmex[[q]]), col.names = c(q, "N"), caption = paste0("Frequency of ", q))
tab
tab_binary <- kable(table(nmex[[new_col]]), col.names = c(q, "N"), caption = paste0("Frequency of Recoded: ", q))
tab_binary
print(tab)
print(tab_binary)
}

```

```

##
##
## Table: Frequency of cidi_q1
##
## |cidi_q1 |    N|
## |:-----|----:|
## |0       | 8749|
## |1       |  482|
## |777     |   54|
##
##
## Table: Frequency of Recoded: cidi_q1_binary
##
## |cidi_q1 |    N|
## |:-----|----:|
## |0       | 8803|
## |1       |  482|
##
##
## Table: Frequency of cidi_q2
##
## |cidi_q2 |    N|
## |:-----|----:|
## |0       | 7779|
## |1       | 1490|
## |777     |   16|
##
##
## Table: Frequency of Recoded: cidi_q2_binary
##
## |cidi_q2 |    N|
## |:-----|----:|
## |0       | 7795|
## |1       | 1490|
##
##
## Table: Frequency of cidi_q3
##
## |cidi_q3 |    N|
## |:-----|----:|
## |0       | 7670|
## |1       | 1605|
## |777     |   10|
##
##

```

```

## Table: Frequency of Recoded: cidi_q3_binary
##
## |cidi_q3 |    N|
## |:-----|----:|
## |0       | 7680|
## |1       | 1605|
##
##
## Table: Frequency of cidi_q4
##
## |cidi_q4 |    N|
## |:-----|----:|
## |0       | 8390|
## |1       |  875|
## |777     |   20|
##
##
## Table: Frequency of Recoded: cidi_q4_binary
##
## |cidi_q4 |    N|
## |:-----|----:|
## |0       | 8410|
## |1       |  875|
##
##
## Table: Frequency of cidi_q5
##
## |cidi_q5 |    N|
## |:-----|----:|
## |0       | 7803|
## |1       | 1455|
## |777     |   27|
##
##
## Table: Frequency of Recoded: cidi_q5_binary
##
## |cidi_q5 |    N|
## |:-----|----:|
## |0       | 7830|
## |1       | 1455|
##
##
## Table: Frequency of cidi_q6
##
## |cidi_q6 |    N|
## |:-----|----:|
## |0       | 9205|
## |1       |   65|
## |777     |   15|
##
##
## Table: Frequency of Recoded: cidi_q6_binary
##
## |cidi_q6 |    N|

```

```

## |:-----|----:|
## |0       | 9220|
## |1       |   65|
##
##
## Table: Frequency of cidi_q7
##
## |cidi_q7 |    N|
## |:-----|----:|
## |0       | 9186|
## |1       |   73|
## |777     |   26|
##
##
## Table: Frequency of Recoded: cidi_q7_binary
##
## |cidi_q7 |    N|
## |:-----|----:|
## |0       | 9212|
## |1       |   73|
##
##
## Table: Frequency of cidi_q8
##
## |cidi_q8 |    N|
## |:-----|----:|
## |0       | 8919|
## |1       |  342|
## |777     |   24|
##
##
## Table: Frequency of Recoded: cidi_q8_binary
##
## |cidi_q8 |    N|
## |:-----|----:|
## |0       | 8943|
## |1       |  342|
##
##
## Table: Frequency of cidi_q9
##
## |cidi_q9 |    N|
## |:-----|----:|
## |0       | 7930|
## |1       | 1290|
## |777     |   65|
##
##
## Table: Frequency of Recoded: cidi_q9_binary
##
## |cidi_q9 |    N|
## |:-----|----:|
## |0       | 7995|
## |1       | 1290|

```

```

##
##
## Table: Frequency of cidi_q10
##
## |cidi_q10 |    N|
## |:-----|----:|
## |0        | 8821|
## |1        |  445|
## |777      |   19|
##
##
## Table: Frequency of Recoded: cidi_q10_binary
##
## |cidi_q10 |    N|
## |:-----|----:|
## |0        | 8840|
## |1        |  445|
##
##
## Table: Frequency of cidi_q11
##
## |cidi_q11 |    N|
## |:-----|----:|
## |0        | 9223|
## |1        |   45|
## |777      |   17|
##
##
## Table: Frequency of Recoded: cidi_q11_binary
##
## |cidi_q11 |    N|
## |:-----|----:|
## |0        | 9240|
## |1        |   45|
##
##
## Table: Frequency of cidi_q12
##
## |cidi_q12 |    N|
## |:-----|----:|
## |0        | 9060|
## |1        |   202|
## |777      |    23|
##
##
## Table: Frequency of Recoded: cidi_q12_binary
##
## |cidi_q12 |    N|
## |:-----|----:|
## |0        | 9083|
## |1        |   202|
##
##
## Table: Frequency of cidi_q13

```

```

##
## |cidi_q13 |    N|
## |:-----|----:|
## |0        | 7949|
## |1        | 1261|
## |777      |   75|
##
##
## Table: Frequency of Recoded: cidi_q13_binary
##
## |cidi_q13 |    N|
## |:-----|----:|
## |0        | 8024|
## |1        | 1261|
##
##
## Table: Frequency of cidi_q14
##
## |cidi_q14 |    N|
## |:-----|----:|
## |0        | 6360|
## |1        | 2897|
## |777      |   28|
##
##
## Table: Frequency of Recoded: cidi_q14_binary
##
## |cidi_q14 |    N|
## |:-----|----:|
## |0        | 6388|
## |1        | 2897|
##
##
## Table: Frequency of cidi_q15
##
## |cidi_q15 |    N|
## |:-----|----:|
## |0        | 9180|
## |1        |   53|
## |777      |   52|
##
##
## Table: Frequency of Recoded: cidi_q15_binary
##
## |cidi_q15 |    N|
## |:-----|----:|
## |0        | 9232|
## |1        |   53|
##
##
## Table: Frequency of cidi_q16
##
## |cidi_q16 |    N|
## |:-----|----:|

```

```

## |0          | 8961|
## |1          |  298|
## |777        |   26|
##
##
## Table: Frequency of Recoded: cidi_q16_binary
##
## |cidi_q16 |    N|
## |:-----|----:|
## |0          | 8987|
## |1          |  298|
##
##
## Table: Frequency of cidi_q17
##
## |cidi_q17 |    N|
## |:-----|----:|
## |0          | 9118|
## |1          |  145|
## |777        |   22|
##
##
## Table: Frequency of Recoded: cidi_q17_binary
##
## |cidi_q17 |    N|
## |:-----|----:|
## |0          | 9140|
## |1          |  145|
##
##
## Table: Frequency of cidi_q18
##
## |cidi_q18 |    N|
## |:-----|----:|
## |0          | 8544|
## |1          |  697|
## |777        |   44|
##
##
## Table: Frequency of Recoded: cidi_q18_binary
##
## |cidi_q18 |    N|
## |:-----|----:|
## |0          | 8588|
## |1          |  697|

```

```

nmex <- nmex %>% mutate(cidi_any_binary = case_when(
  (cidi_q1_binary == 1 |
   cidi_q2_binary == 1 |
   cidi_q3_binary == 1 |
   cidi_q4_binary == 1 |
   cidi_q5_binary == 1 |
   cidi_q6_binary == 1 |
   cidi_q7_binary == 1 |

```

```

    cidi_q8_binary == 1 |
    cidi_q9_binary == 1 |
    cidi_q10_binary == 1 |
    cidi_q11_binary == 1 |
    cidi_q12_binary == 1 |
    cidi_q13_binary == 1 |
    cidi_q14_binary == 1 |
    #cidi_q15_binary == 1 | Exclude HIV
    cidi_q16_binary == 1 |
    #cidi_q17_binary == 1 | Exclude Cancer
    cidi_q18_binary == 1) ~ 1,

(cidi_q1_binary %in% c(0,777) |
cidi_q2_binary %in% c(0,777) |
cidi_q3_binary %in% c(0,777) |
cidi_q4_binary %in% c(0,777) |
cidi_q5_binary %in% c(0,777) |
cidi_q6_binary %in% c(0,777) |
cidi_q7_binary %in% c(0,777) |
cidi_q8_binary %in% c(0,777) |
cidi_q9_binary %in% c(0,777) |
cidi_q10_binary %in% c(0,777) |
cidi_q11_binary %in% c(0,777) |
cidi_q12_binary %in% c(0,777) |
cidi_q13_binary %in% c(0,777) |
cidi_q14_binary %in% c(0,777) |
cidi_q15_binary %in% c(0,777) |
cidi_q16_binary %in% c(0,777) |
cidi_q17_binary %in% c(0,777) |
cidi_q18_binary %in% c(0,777)) ~ 0,

TRUE ~ NA_real_
))

```

CIDI Load Here we sum all CIDI binary variables to create a count of chronic conditions, similar to above with trauma.

We removed HIV (cidi_q15) and cancer (cidi_q17) after discussion on 1-8-25. This was to maintain consistency with the other models.

```

#Add all "binary" variables together for a total sum of cidi conditions.
nmex <- nmex %>% mutate(
  cidi_load = rowSums(dplyr::select(.,
cidi_q1_binary, cidi_q2_binary, cidi_q3_binary, cidi_q4_binary,
cidi_q5_binary, cidi_q6_binary, cidi_q7_binary, cidi_q8_binary,
cidi_q9_binary, cidi_q10_binary, cidi_q11_binary, cidi_q12_binary,
cidi_q13_binary, cidi_q14_binary,
#cidi_q15_binary,
cidi_q16_binary,
#cidi_q17_binary,
cidi_q18_binary
), na.rm=TRUE))

```


CIDI categorical

Similar to above, we create a categorical variable of 0 to 4+ CIDI experienced.

```
#Create variable for 0,1,2,3, or 4+ traumas
nmex <- nmex %>% mutate(
  number_cidi = case_when(
    cidi_load == 0 ~ "0", #If have experienced at least one cidi
    cidi_load == 1 ~ "1",
    cidi_load == 2 ~ "2",
    cidi_load == 3 ~ "3",
    cidi_load >= 4 ~ "4+",
    TRUE ~ NA_character_
  )
)

kable(table(nmex$number_cidi), col.names = c("CIDI Load Cat.", "N"), caption = "Frequency CIDI Loads")
```

Table 16: Frequency CIDI Loads

CIDI Load Cat.	N
0	2907
1	2652
2	1818
3	981
4+	928

Grouped Chronic conditions Here we create sums of specific CIDI variables in the following classifications: pain, cardiometabolic, respiratory, neurological,

```
###ALED###
# Create pain load and associated binary variable
nmex <- nmex %>% mutate(
  cidi_pain_load = rowSums(dplyr::select(., cidi_q1_binary,
    cidi_q2_binary, cidi_q3_binary, cidi_q4_binary), na.rm=TRUE)) %>%
  mutate(cidi_pain_binary = ifelse(cidi_pain_load>0, 1, 0))

# Create cardiac load and associated binary variable
nmex <- nmex %>% mutate(
  cidi_cardiac_load = rowSums(dplyr::select(., cidi_q7_binary,
    cidi_q8_binary, cidi_q9_binary, cidi_q13_binary), na.rm=TRUE)) %>%
  mutate(cidi_cardiac_binary = ifelse(cidi_cardiac_load>0, 1, 0))

# Create respiratory load and associated binary variable
nmex <- nmex %>% mutate(
  cidi_resp_load = rowSums(dplyr::select(.,
    cidi_q10_binary, cidi_q11_binary, cidi_q12_binary),
    na.rm=TRUE)) %>%
  mutate(cidi_resp_binary = ifelse(cidi_resp_load>0, 1, 0))
```

```

# Create neurological load and associated binary variable
nmex <- nmex %>% mutate(
  cidi_neur_load = rowSums(dplyr::select(.,
    cidi_q6_binary, cidi_q16_binary),
    na.rm=TRUE)) %>%
  mutate(cidi_neur_binary = ifelse(cidi_neur_load>0, 1, 0))

# Create overall CIDI binary variable (any condition)
nmex <- nmex %>% mutate(cidi_binary = ifelse(cidi_load>0, 1, 0))

```

ASSIST

Here we recode alcohol, tobacco use, and cannabis use as Never/Once or twice”, “Monthly/Weekly”, and “Daily”. We decided to do this to allow for more granularity (relative to a binary “Ever”/“Never”).

```

nmex <- nmex %>% mutate(assist_alcohol_amt_recode =
  case_when(

    (assist_alcohol == 0 | (assist_alcohol == 1 &
      assist_alcohol_amt %in% c(1,2))) ~ "Never/Once or twice",

    (assist_alcohol == 1 &
      assist_alcohol_amt %in% c(3,4)) ~ "Monthly/Weekly",

    (assist_alcohol == 1 &
      assist_alcohol_amt == 5) ~ "Daily",
    TRUE ~ NA_character_),

  assist_tobacco_amt_recode =
  case_when(

    assist_tobacco == 0 | (assist_tobacco == 1 &
      assist_tobacco_amt %in% c(1,2)) ~ "Never/Once or twice",

    (assist_tobacco == 1 &
      assist_tobacco_amt %in% c(3,4)) ~ "Monthly/Weekly",

    (assist_tobacco == 1 &
      assist_tobacco_amt == 5) ~ "Daily",

    TRUE ~ NA_character_),

  assist_cannabis_amt_recode =
  case_when(

    assist_cannabis == 0 | (assist_cannabis == 1 &
      assist_cannabis_amt %in% c(1,2)) ~ "Never/Once or twice",

    (assist_cannabis == 1 &
      assist_cannabis_amt %in% c(3,4)) ~ "Monthly/Weekly",

```

```
(assist_cannabis == 1 &
assist_cannabis_amt == 5) ~ "Daily",

TRUE ~ NA_character_))
```

```
kable(table(nmex$assist_alcohol), col.names = c("Alcohol", "N"), caption = "Frequency Ever use alcohol")
```

Table 17: Frequency Ever use alcohol

Alcohol	N
0	1262
1	8021

```
kable(table(nmex$assist_alcohol_amt), col.names = c("Alcohol AMT", "N"), caption = "Frequency alcohol in
```

Table 18: Frequency alcohol in past 3 months

Alcohol AMT	N
1	3814
2	1806
3	1167
4	1158
5	76

```
kable(table(nmex$assist_alcohol_amt_recode), col.names = c("Alcohol Recode", "N"), caption = "Frequency
```

Table 19: Frequency Alcohol Recoded

Alcohol Recode	N
Daily	76
Monthly/Weekly	2325
Never/Once or twice	6882

```
kable(table(nmex$assist_tobacco), col.names = c("Tobacco", "N"), caption = "Frequency Ever use tobacco")
```

Table 20: Frequency Ever use tobacco

Tobacco	N
0	3011
1	6272

```
kable(table(nmex$assist_tobacco_amt), col.names = c("Tobacco AMT", "N"), caption = "Frequency of tobacco
```

Table 21: Frequency of tobacco in past 3 months

Tobacco AMT	N
1	3438
2	532
3	221
4	424
5	1657

```
kable(table(nmex$assist_tobacco_amt_recode), col.names = c("Tobacco Recode", "N"), caption = "Frequency
```

Table 22: Frequency Interpersonal Trauma Load

Tobacco Recode	N
Daily	1657
Monthly/Weekly	645
Never/Once or twice	6981

```
kable(table(nmex$assist_cannabis), col.names = c("Cannabis", "N"), caption = "Frequency Ever use cannabis
```

Table 23: Frequency Ever use cannabis

Cannabis	N
0	6263
1	3020

```
kable(table(nmex$assist_cannabis_amt), col.names = c("Cannabis AMT", "N"), caption = "Frequency of cannabis
```

Table 24: Frequency of cannabis in past 3 months

Cannabis AMT	N
1	2418
2	323
3	110
4	95
5	74

```
kable(table(nmex$assist_cannabis_amt_recode), col.names = c("Cannabis Recode", "N"), caption = "Frequency
```

Table 25: Frequency of cannabis Recoded

Cannabis Recode	N
Daily	74
Monthly/Weekly	205
Never/Once or twice	9004

Filter data to cases

Here we filter the data to only cases. This is what we will use for the analysis.

```
nmex_cases <- nmex %>% filter(is_case == 1)
```

Tables

Demographics

Here we produce Table 1: Demographics

```
label(nmex_cases$age_at_interview) <- "Age at interview"
label(nmex_cases$age_cat) <- "Age Categories"
nmex_cases$psychosis_primary_f <- as.factor(nmex_cases$psychosis_primary)
label(nmex_cases$psychosis_primary_f) <- "Primary Psychosis"
label(nmex_cases$site_location_f) <- "Sites"
label(nmex_cases$site_location_group_f) <- "Urbanicity"
label(nmex_cases$ses_status_f) <- "SES"
label(nmex_cases$edu_ord) <- "Education"

table1(~ age_at_interview + age_cat + psychosis_primary_f +
  site_location_f + site_location_group_f + ses_status_f + factor(ses_status) + edu_ord
  | is_male_f, data= nmex_cases,
  caption = "Table 1: Demographics")
```

Interpersonal trauma: CIDI and Substance Use

Here we create a table of outcomes by interpersonal trauma.(NOT SHOWN)

Noninterpersonal trauma: CIDI and Substance Use

Here we create a table of outcomes by noninterpersonal trauma. (NOT SHOWN)

Table Sex X LEC

Here we make table 2 which shows sex with interpersonal and non-interpersonal chronic conditions.

Table 26: Table 1: Demographics

	Male	Female	Overall
	(N=2407)	(N=2268)	(N=4675)
Age at interview			
Mean (SD)	40.5 (12.4)	43.7 (12.6)	42.0 (12.6)
Median [Min, Max]	40.0 [18.0, 84.0]	44.0 [18.0, 82.0]	42.0 [18.0, 84.0]
Age Categories			
18-29	558 (23.2%)	352 (15.5%)	910 (19.5%)
30-39	618 (25.7%)	512 (22.6%)	1130 (24.2%)
40-49	596 (24.8%)	618 (27.2%)	1214 (26.0%)
50-59	486 (20.2%)	564 (24.9%)	1050 (22.5%)
60+	149 (6.2%)	222 (9.8%)	371 (7.9%)
Primary Psychosis			
1	406 (16.9%)	935 (41.2%)	1341 (28.7%)
2	98 (4.1%)	197 (8.7%)	295 (6.3%)
3	52 (2.2%)	90 (4.0%)	142 (3.0%)
4	1773 (73.7%)	932 (41.1%)	2705 (57.9%)
5	53 (2.2%)	80 (3.5%)	133 (2.8%)
7	24 (1.0%)	33 (1.5%)	57 (1.2%)
Missing	1 (0.0%)	1 (0.0%)	2 (0.0%)
Sites			
CDMX	1334 (55.4%)	1374 (60.6%)	2708 (57.9%)
Campeche	28 (1.2%)	38 (1.7%)	66 (1.4%)
Queretaro	176 (7.3%)	150 (6.6%)	326 (7.0%)
Leon	103 (4.3%)	111 (4.9%)	214 (4.6%)
Guadalajara	524 (21.8%)	345 (15.2%)	869 (18.6%)
Morelia	242 (10.1%)	250 (11.0%)	492 (10.5%)
Urbanicity			
Rural	373 (15.5%)	399 (17.6%)	772 (16.5%)
Urban	2034 (84.5%)	1869 (82.4%)	3903 (83.5%)
SES			
Low-Middle Low	1632 (67.8%)	1460 (64.4%)	3092 (66.1%)
Middle-High	744 (30.9%)	767 (33.8%)	1511 (32.3%)
Missing	31 (1.3%)	41 (1.8%)	72 (1.5%)
factor(ses__status)			
1	863 (35.9%)	770 (34.0%)	1633 (34.9%)
2	769 (31.9%)	690 (30.4%)	1459 (31.2%)
3	75 (3.1%)	77 (3.4%)	152 (3.3%)
4	651 (27.0%)	678 (29.9%)	1329 (28.4%)
5	18 (0.7%)	12 (0.5%)	30 (0.6%)
777	26 (1.1%)	38 (1.7%)	64 (1.4%)
888	4 (0.2%)	3 (0.1%)	7 (0.1%)
Missing	1 (0.0%)	0 (0%)	1 (0.0%)
Education			
Elementary or middle schol	1153 (47.9%)	910 (40.1%)	2063 (44.1%)
High school or technical school	729 (30.3%)	694 (30.6%)	1423 (30.4%)
Literate	111 (4.6%)	101 (4.5%)	212 (4.5%)
None	31 (1.3%)	58 (2.6%)	89 (1.9%)
Post secondary school	383 (15.9%)	505 (22.3%)	888 (19.0%)

```

pvalue <- function(x, ...) {
  # Construct vectors of data y, and groups (strata) g
  y <- unlist(x)
  g <- factor(rep(1:length(x), times=apply(x, length)))
  if (is.numeric(y)) {
    # For numeric variables, perform a standard 2-sample t-test
    p <- t.test(y ~ g)$p.value
  } else {
    # For categorical variables, perform a chi-squared test of independence
    p <- chisq.test(table(y, g))$p.value
  }
  # Format the p-value, using an HTML entity for the less-than sign.
  # The initial empty string places the output on the line below the variable label.
  c("", sub("<", "&lt;", format.pval(p, digits=3, eps=0.001)))
}

table1(~ factor(phys_assault_1) + factor(weapon_assault_1) +
factor(sexual_assault_1) + factor(other_sexual_1) + factor(captivity_1) + factor(harm_else_1) +
factor(natural_disaster_1) + factor(fire_exp_1) +
factor(transp_accident_1) + factor(serious_accident_1) + factor(toxic_sub_1) + factor(combat_1) +
factor(illness_injury_1) +
trauma_load_inter_cat +
trauma_load_noninter_nhs_cat +
trauma_load_interpersonal_1 +
trauma_load_noninter_1_nhs +
trauma_load_any_1 | is_male_f, data=nmx_cases)

```

	Male	Female	Overall
	(N=2407)	(N=2268)	(N=4675)
factor(phys__assault_1)			
0	1425 (59.2%)	1532 (67.5%)	2957 (63.3%)
1	982 (40.8%)	736 (32.5%)	1718 (36.7%)
factor(weapon__assault_1)			
0	1775 (73.7%)	1919 (84.6%)	3694 (79.0%)
1	632 (26.3%)	349 (15.4%)	981 (21.0%)
factor(sexual__assault_1)			
0	2130 (88.5%)	1618 (71.3%)	3748 (80.2%)
1	277 (11.5%)	650 (28.7%)	927 (19.8%)
factor(other__sexual_1)			
0	2197 (91.3%)	1793 (79.1%)	3990 (85.3%)
1	210 (8.7%)	475 (20.9%)	685 (14.7%)
factor(captivity_1)			
0	2255 (93.7%)	2125 (93.7%)	4380 (93.7%)
1	152 (6.3%)	143 (6.3%)	295 (6.3%)
factor(harm__else_1)			
0	2221 (92.3%)	2097 (92.5%)	4318 (92.4%)
1	186 (7.7%)	171 (7.5%)	357 (7.6%)
factor(natural__disaster_1)			
0	1848 (76.8%)	1668 (73.5%)	3516 (75.2%)
1	559 (23.2%)	600 (26.5%)	1159 (24.8%)
factor(fire__exp_1)			
0	2249 (93.4%)	2134 (94.1%)	4383 (93.8%)
1	158 (6.6%)	134 (5.9%)	292 (6.2%)
factor(transp__accident_1)			
0	1885 (78.3%)	1938 (85.4%)	3823 (81.8%)
1	522 (21.7%)	330 (14.6%)	852 (18.2%)
factor(serious__accident_1)			
0	2046 (85.0%)	2023 (89.2%)	4069 (87.0%)
1	361 (15.0%)	245 (10.8%)	606 (13.0%)
factor(toxic__sub_1)			
0	2252 (93.6%)	2196 (96.8%)	4448 (95.1%)
1	155 (6.4%)	72 (3.2%)	227 (4.9%)
factor(combat_1)			
0	2298 (95.5%)	2187 (96.4%)	4485 (95.9%)
1	109 (4.5%)	81 (3.6%)	190 (4.1%)
factor(illness__injury_1)			
0	2024 (84.1%)	1916 (84.5%)	3940 (84.3%)
1	383 (15.9%)	352 (15.5%)	735 (15.7%)
trauma_load_inter_cat			
0	1062 (44.1%)	1003 (44.2%)	2065 (44.2%)
1	638 (26.5%)	546 (24.1%)	1184 (25.3%)
2	440 (18.3%)	356 (15.7%)	796 (17.0%)
>=3	267 (11.1%)	363 (16.0%)	630 (13.5%)
trauma_load_noninter_nhs_cat			
0	1137 (47.2%)	1147 (50.6%)	2284 (48.9%)
1	691 (28.7%)	644 (28.4%)	1335 (28.6%)
2	326 (13.5%)	320 (14.1%)	646 (13.8%)
>=3	253 (10.5%)	157 (6.9%)	410 (8.8%)
trauma_load_interpersonal_1			
Mean (SD)	1.01 (1.16)	1.11 (1.29)	1.06 (1.22)
Median [Min, Max]	1.00 [0, 6.00]	1.00 [0, 6.00]	1.00 [0, 6.00]


```

# Getting p values
table1(~ factor(phys_assault_1) + factor(weapon_assault_1) +
factor(sexual_assault_1) + factor(other_sexual_1) + factor(captivity_1) + factor(harm_else_1) +
factor(natural_disaster_1) + factor(fire_exp_1) +
factor(transp_accident_1) + factor(serious_accident_1) + factor(toxic_sub_1) + factor(combat_1) +
factor(illness_injury_1) +
trauma_load_inter_cat +
trauma_load_noninter_nhs_cat +
trauma_load_interpersonal_1 +
trauma_load_noninter_1_nhs +
trauma_load_any_1 | is_male_f, overall=F, data=nmx_cases,
extra.col=list(`P-value`=pvalue))

```

	Male	Female	P-value
	(N=2407)	(N=2268)	
factor(phys__assault_1)			
0	1425 (59.2%)	1532 (67.5%)	<0.001
1	982 (40.8%)	736 (32.5%)	
factor(weapon__assault_1)			
0	1775 (73.7%)	1919 (84.6%)	<0.001
1	632 (26.3%)	349 (15.4%)	
factor(sexual__assault_1)			
0	2130 (88.5%)	1618 (71.3%)	<0.001
1	277 (11.5%)	650 (28.7%)	
factor(other__sexual_1)			
0	2197 (91.3%)	1793 (79.1%)	<0.001
1	210 (8.7%)	475 (20.9%)	
factor(captivity_1)			
0	2255 (93.7%)	2125 (93.7%)	1
1	152 (6.3%)	143 (6.3%)	
factor(harm__else_1)			
0	2221 (92.3%)	2097 (92.5%)	0.852
1	186 (7.7%)	171 (7.5%)	
factor(natural__disaster_1)			
0	1848 (76.8%)	1668 (73.5%)	0.0116
1	559 (23.2%)	600 (26.5%)	
factor(fire__exp_1)			
0	2249 (93.4%)	2134 (94.1%)	0.387
1	158 (6.6%)	134 (5.9%)	
factor(transp__accident_1)			
0	1885 (78.3%)	1938 (85.4%)	<0.001
1	522 (21.7%)	330 (14.6%)	
factor(serious__accident_1)			
0	2046 (85.0%)	2023 (89.2%)	<0.001
1	361 (15.0%)	245 (10.8%)	
factor(toxic__sub_1)			
0	2252 (93.6%)	2196 (96.8%)	<0.001
1	155 (6.4%)	72 (3.2%)	
factor(combat_1)			
0	2298 (95.5%)	2187 (96.4%)	0.114
1	109 (4.5%)	81 (3.6%)	
factor(illness__injury_1)			
0	2024 (84.1%)	1916 (84.5%)	0.743
1	383 (15.9%)	352 (15.5%)	
trauma_load_inter_cat			
0	1062 (44.1%)	1003 (44.2%)	<0.001
1	638 (26.5%)	546 (24.1%)	
2	440 (18.3%)	356 (15.7%)	
>=3	267 (11.1%)	363 (16.0%)	
trauma_load_noninter_nhs_cat			
0	1137 (47.2%)	1147 (50.6%)	<0.001
1	691 (28.7%)	644 (28.4%)	
2	326 (13.5%)	320 (14.1%)	
>=3	253 (10.5%)	157 (6.9%)	
trauma_load_interpersonal_1			
Mean (SD)	1.01 (1.16)	1.11 (1.29)	0.00553
Median [Min, Max]	1.00 [0, 6.00]	1.00 [0, 6.00]	
trauma_load_noninter_1_nhs			

```
chi_cap <- table(as.factor(nmex_cases$captivity_1), nmex_cases$is_male_f)
chi_cap
```

```
##
##      Male Female
##    0 2255    2125
##    1  152     143
```

```
chisq.test(chi_cap, )
```

```
##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data:  chi_cap
## X-squared = 1.7541e-28, df = 1, p-value = 1
```

Table of chronic conditions

```
table1(~ factor(cidi_any_binary) + #Any

      factor(cidi_pain_binary) + #Pain
      factor(cidi_q1_binary) + # Arthritis
      factor(cidi_q2_binary) + #Back or neck problems
      factor(cidi_q3_binary) + #Headaches
      factor(cidi_q4_binary) + #Other chronic pain

      factor(cidi_cardiac_binary) + #Cardiometabolic
      factor(cidi_q7_binary) + #Heart attack
      factor(cidi_q8_binary) + #Heart disease
      factor(cidi_q9_binary) + #High blood pressure
      factor(cidi_q13_binary) + #Diabetes

      factor(cidi_resp_binary) + #Respiratory
      factor(cidi_q10_binary) + #Asthma
      factor(cidi_q11_binary) + #Tuberculosis
      factor(cidi_q12_binary) + #Other lung

      factor(cidi_neur_binary) + #Neurological
      factor(cidi_q6_binary) + #Stroke
      factor(cidi_q16_binary) + #Epilepsy

      factor(cidi_q14_binary) + #Ulcer
      factor(cidi_q5_binary) + #Allergies
      factor(cidi_q18_binary) + #Hypothyroidism
| is_male_f, data=nmex_cases)
```

	Male	Female	Overall
	(N=2407)	(N=2268)	(N=4675)
factor(cidi_any_binary)			
0	972 (40.4%)	652 (28.7%)	1624 (34.7%)
1	1435 (59.6%)	1616 (71.3%)	3051 (65.3%)
factor(cidi_pain_binary)			
0	1789 (74.3%)	1428 (63.0%)	3217 (68.8%)
1	618 (25.7%)	840 (37.0%)	1458 (31.2%)
factor(cidi_q1_binary)			
0	2342 (97.3%)	2137 (94.2%)	4479 (95.8%)
1	65 (2.7%)	131 (5.8%)	196 (4.2%)
factor(cidi_q2_binary)			
0	2116 (87.9%)	1902 (83.9%)	4018 (85.9%)
1	291 (12.1%)	366 (16.1%)	657 (14.1%)
factor(cidi_q3_binary)			
0	2106 (87.5%)	1808 (79.7%)	3914 (83.7%)
1	301 (12.5%)	460 (20.3%)	761 (16.3%)
factor(cidi_q4_binary)			
0	2251 (93.5%)	2049 (90.3%)	4300 (92.0%)
1	156 (6.5%)	219 (9.7%)	375 (8.0%)
factor(cidi_cardiac_binary)			
0	1923 (79.9%)	1678 (74.0%)	3601 (77.0%)
1	484 (20.1%)	590 (26.0%)	1074 (23.0%)
factor(cidi_q7_binary)			
0	2394 (99.5%)	2254 (99.4%)	4648 (99.4%)
1	13 (0.5%)	14 (0.6%)	27 (0.6%)
factor(cidi_q8_binary)			
0	2327 (96.7%)	2186 (96.4%)	4513 (96.5%)
1	80 (3.3%)	82 (3.6%)	162 (3.5%)
factor(cidi_q9_binary)			
0	2125 (88.3%)	1937 (85.4%)	4062 (86.9%)
1	282 (11.7%)	331 (14.6%)	613 (13.1%)
factor(cidi_q13_binary)			
0	2196 (91.2%)	1929 (85.1%)	4125 (88.2%)
1	211 (8.8%)	339 (14.9%)	550 (11.8%)
factor(cidi_resp_binary)			
0	2227 (92.5%)	2128 (93.8%)	4355 (93.2%)
1	180 (7.5%)	140 (6.2%)	320 (6.8%)
factor(cidi_q10_binary)			
0	2300 (95.6%)	2183 (96.3%)	4483 (95.9%)
1	107 (4.4%)	85 (3.7%)	192 (4.1%)
factor(cidi_q11_binary)			
0	2392 (99.4%)	2258 (99.6%)	4650 (99.5%)
1	15 (0.6%)	10 (0.4%)	25 (0.5%)
factor(cidi_q12_binary)			
0	2338 (97.1%)	2217 (97.8%)	4555 (97.4%)
1	69 (2.9%)	51 (2.2%)	120 (2.6%)
factor(cidi_neur_binary)			
0	2283 (94.8%)	2158 (95.2%)	4441 (95.0%)
1	124 (5.2%)	110 (4.9%)	234 (5.0%)
factor(cidi_q6_binary)			
0	2393 (99.4%)	2251 (99.3%)	4644 (99.3%)
1	14 (0.6%)	17 (0.7%)	31 (0.7%)
factor(cidi_q16_binary)			

```

table1(~ factor(cidi_any_binary) + #Any

      factor(cidi_pain_binary) + #Pain
      factor(cidi_q1_binary) + # Arthritis
      factor(cidi_q2_binary) + #Back or neck problems
      factor(cidi_q3_binary) + #Headaches
      factor(cidi_q4_binary) + #Other chronic pain

      factor(cidi_cardiac_binary) + #Cardiometabolic
      factor(cidi_q7_binary) + #Heart attack
      factor(cidi_q8_binary) + #Heart disease
      factor(cidi_q9_binary) + #High blood pressure
      factor(cidi_q13_binary) + #Diabetes

      factor(cidi_resp_binary) + #Respiratory
      factor(cidi_q10_binary) + #Asthma
      factor(cidi_q11_binary) + #Tuberculosis
      factor(cidi_q12_binary) + #Other lung

      factor(cidi_neur_binary) + #Neurological
      factor(cidi_q6_binary) + #Stroke
      factor(cidi_q16_binary) + #Epilepsy

      factor(cidi_q14_binary) + #Ulcer
      factor(cidi_q5_binary) + #Allergies
      factor(cidi_q18_binary) + #Hypothyroidism
| is_male_f, overall=F, data=nmx_cases,
extra.col=list(`P-value`=pvalue))

```

	Male	Female	P-value
	(N=2407)	(N=2268)	
factor(cidi_any_binary)			
0	972 (40.4%)	652 (28.7%)	<0.001
1	1435 (59.6%)	1616 (71.3%)	
factor(cidi_pain_binary)			
0	1789 (74.3%)	1428 (63.0%)	<0.001
1	618 (25.7%)	840 (37.0%)	
factor(cidi_q1_binary)			
0	2342 (97.3%)	2137 (94.2%)	<0.001
1	65 (2.7%)	131 (5.8%)	
factor(cidi_q2_binary)			
0	2116 (87.9%)	1902 (83.9%)	<0.001
1	291 (12.1%)	366 (16.1%)	
factor(cidi_q3_binary)			
0	2106 (87.5%)	1808 (79.7%)	<0.001
1	301 (12.5%)	460 (20.3%)	
factor(cidi_q4_binary)			
0	2251 (93.5%)	2049 (90.3%)	<0.001
1	156 (6.5%)	219 (9.7%)	
factor(cidi_cardiac_binary)			
0	1923 (79.9%)	1678 (74.0%)	<0.001
1	484 (20.1%)	590 (26.0%)	
factor(cidi_q7_binary)			
0	2394 (99.5%)	2254 (99.4%)	0.877
1	13 (0.5%)	14 (0.6%)	
factor(cidi_q8_binary)			
0	2327 (96.7%)	2186 (96.4%)	0.642
1	80 (3.3%)	82 (3.6%)	
factor(cidi_q9_binary)			
0	2125 (88.3%)	1937 (85.4%)	0.00409
1	282 (11.7%)	331 (14.6%)	
factor(cidi_q13_binary)			
0	2196 (91.2%)	1929 (85.1%)	<0.001
1	211 (8.8%)	339 (14.9%)	
factor(cidi_resp_binary)			
0	2227 (92.5%)	2128 (93.8%)	0.0875
1	180 (7.5%)	140 (6.2%)	
factor(cidi_q10_binary)			
0	2300 (95.6%)	2183 (96.3%)	0.26
1	107 (4.4%)	85 (3.7%)	
factor(cidi_q11_binary)			
0	2392 (99.4%)	2258 (99.6%)	0.514
1	15 (0.6%)	10 (0.4%)	
factor(cidi_q12_binary)			
0	2338 (97.1%)	2217 (97.8%)	0.214
1	69 (2.9%)	51 (2.2%)	
factor(cidi_neur_binary)			
0	2283 (94.8%)	2158 (95.2%)	0.685
1	124 (5.2%)	110 (4.9%)	
factor(cidi_q6_binary)			
0	2393 (99.4%)	2251 (99.3%)	0.598
1	14 (0.6%)	17 (0.7%)	
factor(cidi_q16_binary)			

Models and Figures

As per: <https://docs.google.com/document/d/1g42GgDfTURCFaz44x83fNtuOiDNBV2NGff1iqdeEl6A/edit?tab=t.0>

1. **Model B:** {Health outcome grouping} ~ trauma_load_inter_categorical (coded as 0, 1, 2, 3+) + trauma_load_noninter_nhs_categorical (coded as 0, 1, 2, 3+) + age + sex + SES status + urbanicity
2. **Model C:** {Health outcome grouping} ~ trauma_load_inter_continuous + trauma_load_noninter_nhs_continuous + age + sex + SES status + urbanicity
3. **Model D:** {Health outcome grouping} ~ trauma_load_inter_cat (coded as 0, 1, 2, 3+) + trauma_load_noninter_nhs_cat (coded as 0, 1, 2, 3+) + age + sex + SES status + urbanicity + alcohol + tobacco + cannabis
4. **Model E:** {Health outcome grouping} ~ trauma_load_inter_continuous + trauma_load_noninter_nhs_continuous + age + sex + SES status + urbanicity + Alcohol + Cannabis + Tobacco

Regression (old)

Here I make a df with all data from all tables.

We loop through each model with each specification and then we loop through each chronic condition in list `cidi_list_binary`.

For variables `cidiq1-q18` (excluding `q15,q17`) we use Bonferonni correction ($\alpha/16$). Else (grouped chronic conditions) we do not.

```
cidi_list_binary <- c(paste0("cidi_q", 1:14, "_binary"), "cidi_q16_binary",
                     "cidi_q18_binary", "cidi_pain_binary",
                     "cidi_cardiac_binary", "cidi_resp_binary",
                     "cidi_neur_binary")

results <- NULL
df <- NULL

models <- c("Model B", "Model C", "Model D", "Model E")

#All Trauma

for (model in models) {

  if (model == "Model B") {
    trauma_var <- "trauma_load_inter_cat + trauma_load_noninter_nhs_cat"
    covar <- "+ age_at_interview + is_male_f + ses_status_f +
site_location_group_f"
    type <- "Categorical"
  } else if (model == "Model C") {
    trauma_var <- "trauma_load_interpersonal_1 + trauma_load_noninter_1_nhs"
    covar <- "+ age_at_interview + is_male_f + ses_status_f +
site_location_group_f"
    type <- "Numeric"
  } else if (model == "Model D") {
    trauma_var <- "trauma_load_inter_cat + trauma_load_noninter_nhs_cat"
```

```

covar <- "+ age_at_interview + is_male_f + ses_status_f +
site_location_group_f + factor(assist_tobacco_amt_recode) +
factor(assist_alcohol_amt_recode) + factor(assist_cannabis_amt_recode)"
type <- "Numeric"
} else if (model == "Model E") {
trauma_var <- "trauma_load_interpersonal_1 + trauma_load_noninter_1_nhs"
covar <- "+ age_at_interview + is_male_f + ses_status_f +
          factor(assist_tobacco_amt_recode) +
          factor(assist_alcohol_amt_recode) +
          factor(assist_cannabis_amt_recode) + site_location_group_f"
type <- "Numeric"
}

for (ii in 1:length(cidi_list_binary)) {

  fmla <- as.formula(paste(cidi_list_binary[ii], " ~ ",
                           trauma_var, covar))

  # Fit the generalized linear regression model
  fit <- glm(fmla, data = nmex_cases, family = binomial)

  #####
  cidi_var <- cidi_list_binary[ii]
  #print(cidi_list_binary[ii])

  cidi_var_n <- table(nmex_cases[[cidi_var]])[2]

  fixef <- exp(coef(fit))

  #Bonferonni correction for single variables
  if (cidi_list_binary[ii] %in% c("Pain", "Cardiometabolic",
                                "Respiratory", "Neurological")) {
    conf <- exp(confint(fit))
    group <- "Not grouped"
  } else {
    conf <- exp(confint(fit, level = (1 - 0.05 / 16)))
    group <- "Single"
  }

  conf_complete <- conf[complete.cases(conf), ]

  var <- rownames(conf_complete)

  for (j in 1:length(var)) {

    var_temp <- var[j]

    est      <- round(exp(summary(fit)[["coefficients"]][var[j],
                                         "Estimate"])), 3)
    #est <- round(exp(summary(fit)[var[j]]), 3)
    ci_ll    <- round(conf_complete[var[j], 1], 3)
  }
}

```



```

ci_ul <- round(conf_complete[var[j], 2],3)
ci <- paste0("(", ci_ll, ", ", ci_ul, ")")

p <- summary(fit)$coefficients[var[j], "Pr(>|z|)"]

#Bonferonni correction for single variables
if (!(cidi_list_binary[ii] %in% c("Pain","Cardiometabolic",
                                "Respiratory","Neurological"))) {
  p <- p/16
}

model <- model

if (p < 0.001) {
  pval <- "<0.001"
} else if (p < 0.01) {
  pval <- "<0.01"
} else if (p < 0.05) {
  pval <- "<0.05"
} else {
  pval <- round(p, 3)
}

results <- rbind(results,c(cidi_var, var_temp, est, ci_ll, ci_ul, pval,
                           model, cidi_var_n, type, group))
}
}
}

#Here we edit the results tibble into a dataframe and modify some variables in prep to make the figures

df <- as.data.frame(results)

colnames(df)<-c("cidi_q", "Coefficient", "OR", "ci_ll", "ci_ul", "pval",
               "model", "cidi_var_n", "type","group")

df <- df %>% mutate(cidi_name = case_when(
  cidi_q == "cidi_q1_binary" ~ "Arthritis or rheumatism",
  cidi_q == "cidi_q2_binary" ~ "Chronic back or neck problems",
  cidi_q == "cidi_q3_binary" ~ "Frequent or severe headaches",
  cidi_q == "cidi_q4_binary" ~ "Any other chronic pain",
  cidi_q == "cidi_q5_binary" ~ "Seasonal allergies like hay fever",
  cidi_q == "cidi_q6_binary" ~ "A stroke",
  cidi_q == "cidi_q7_binary" ~ "A heart attack",
  cidi_q == "cidi_q8_binary" ~ "Heart disease",
  cidi_q == "cidi_q9_binary" ~ "High Blood Pressure",
  cidi_q == "cidi_q10_binary" ~ "Asthma",
  cidi_q == "cidi_q11_binary" ~ "Tuberculosis",
  cidi_q == "cidi_q12_binary" ~ "Any other chronic lung disease",
  cidi_q == "cidi_q13_binary" ~ "Diabetes or high blood sugar",
  cidi_q == "cidi_q14_binary" ~ "An ulcer in your stomach or intestine",
  cidi_q == "cidi_q15_binary" ~ "HIV infection or AIDS",

```

```

cidi_q == "cidi_q16_binary" ~ "Epilepsy or seizures",
cidi_q == "cidi_q17_binary" ~ "Cancer",
cidi_q == "cidi_q18_binary" ~ "Other",
cidi_q == "cidi_pain_binary" ~ "Pain",
cidi_q == "cidi_cardiac_binary" ~ "Cardiometabolic",
cidi_q == "cidi_resp_binary" ~ "Respiratory",
cidi_q == "cidi_neur_binary" ~ "Neurological",

),

coef_name = factor(case_when(
  Coefficient == "trauma_load_inter_cat1" ~ "1 Trauma",
  Coefficient == "trauma_load_inter_cat2" ~ "2 Traumas",
  Coefficient == "trauma_load_inter_cat>=3" ~ "3+ Traumas",
  Coefficient == "trauma_load_noninter_nhs_cat1" ~ "1 Trauma",
  Coefficient == "trauma_load_noninter_nhs_cat2" ~ "2 Traumas",
  Coefficient == "trauma_load_noninter_nhs_cat>=3" ~ "3+ Traumas",
  Coefficient == "trauma_load_noninter_nhs_cat>=3" ~ "3+ Traumas",
  Coefficient == "trauma_load_interpersonal_1" ~ "Continuous",
  Coefficient == "trauma_load_noninter_1_nhs" ~ "Continuous"
), levels = c("1 Trauma",
              "2 Traumas",
              "3+ Traumas",
              "Continuous")),

type = factor(type, levels = c("Numeric", "Categorical")),

trauma_type = case_when(
  grepl("noninter", Coefficient) ~ "Non-interpersonal",
  grepl("inter", Coefficient) ~ "Interpersonal"
)

)

df_trauma_only <- df %>%
  filter(str_starts(Coefficient, "trauma_load"))

```

Figures: ORs by model by chronic condition Here we loop through each of the 18 chronic conditions as well as the grouped chronic conditions and pool all results for Model B, C, D, and E.

In the title there will be the number of participants who endorsed the conditions.

```

outs<-unique(df_trauma_only$cidi_name)

pdf("NMEX_modelBCDE_allconditions.pdf")

for (i in outs) {

  tdf<-df_trauma_only[df_trauma_only$cidi_name==i, ]
  tdf$OR <- as.numeric(tdf$OR)
  tdf$ci_ll <- as.numeric(tdf$ci_ll)
  tdf$ci_ul <- as.numeric(tdf$ci_ul)
}

```

```

cidi_var_n <- tdf$cidi_var_n

p <- ggplot(tdf, aes(x = coef_name, y = OR, ymin = ci_ll, ymax = ci_ul,
                    color = trauma_type)) +
  geom_pointrange(position = position_dodge(width = 0.7)) +
  facet_wrap(~ model, nrow = 1, scales = 'free_x') +
  geom_hline(yintercept = 1, lty = 2) +
  theme_bw() +
  theme(
    axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1),
    legend.position = "right"
  ) +
  #scale_fill_manual(name = "Coefficient Type", ) +

  labs(
    y = "OR (99.7% CI) / OR (95% CI) for grouped",
    x = "Model",
    color = "Trauma Type"
  ) +
  ggtitle(paste(i, "; N =", cidi_var_n))

print(p)
}

dev.off()

```

```

## pdf
## 2

```

Figures: OR Model B and Model C (all CIDI together) Here we show all chronic conditions in the same figure for easy comparison for all four models. We separate outcomes for interpersonal and noninterpersonal trauma.

```

cidi_name_order = c("Arthritis or rheumatism",
  "Chronic back or neck problems",
  "Frequent or severe headaches",
  "Any other chronic pain",
  "Seasonal allergies like hay fever",
  "A stroke",
  "A heart attack",
  "Heart disease",
  "High Blood Pressure",
  "Asthma",
  "Tuberculosis",
  "Any other chronic lung disease",
  "Diabetes or high blood sugar",
  "An ulcer in your stomach or intestine",
  "Epilepsy or seizures",
  "Other",
  "Pain",

```

```

      "Cardiometabolic",
      "Respiratory",
      "Neurological"

    )

model_nosub <- df_trauma_only %>% filter(model %in% c("Model C")) %>%
  mutate(OR = as.numeric(OR),
         ci_ll = as.numeric(ci_ll),
         ci_ul = as.numeric(ci_ul),
         cidi_name = factor(cidi_name, levels = cidi_name_order))

outs<-unique(model_nosub$cidi_name)

pdf("Compiled results_modelBCDE.pdf")

# model_nosub <- df_trauma_only %>% filter(model %in% c("Model B")) %>%
# mutate(OR = as.numeric(OR),
#        ci_ll = as.numeric(ci_ll),
#        ci_ul = as.numeric(ci_ul),
#        cidi_name = factor(cidi_name, levels = cidi_name_order))

df_trauma_only %>% filter(model %in% c("Model B")) %>%
mutate(OR = as.numeric(OR),
      ci_ll = as.numeric(ci_ll),
      ci_ul = as.numeric(ci_ul),
      cidi_name = factor(cidi_name, levels = cidi_name_order)) %>%
  ggplot(aes(x = cidi_name, y = OR, ymin = ci_ll, ymax = ci_ul,
            color = coef_name)) +
  facet_wrap(~trauma_type) +
  geom_pointrange(position = position_dodge(width = 0.7)) +
  geom_hline(yintercept = 1, lty = 2) +
  theme_bw() +
  theme(
    legend.position = "right"
  ) +
  labs(
    y = "OR (99.7% CI) / OR (95% CI) for grouped ",
    x = "",
    color = "Trauma Type",
    title = "Model B (Main categorical model)" +
    ylim(0,5) +
    coord_flip()

df_trauma_only %>% filter(model %in% c("Model C")) %>%
mutate(OR = as.numeric(OR),

```

```

    ci_ll = as.numeric(ci_ll),
    ci_ul = as.numeric(ci_ul),
    cidi_name = factor(cidi_name, levels = cidi_name_order)) %>%
  ggplot(aes(x = cidi_name, y = OR, ymin = ci_ll, ymax = ci_ul,
             color = trauma_type)) +
  geom_pointrange(position = position_dodge(width = 0.7)) +
  geom_hline(yintercept = 1, lty = 2) +
  theme_bw() +
  theme(
    legend.position = "right"
  ) +
  labs(
    y = "OR (99.7% CI) / OR (95% CI) for grouped ",
    x = "",
    color = "Trauma Type",
    title = "Model C (Main cont. model)"
  ) + coord_flip()

df_trauma_only %>% filter(model %in% c("Model D")) %>%
mutate(OR = as.numeric(OR),
       ci_ll = as.numeric(ci_ll),
       ci_ul = as.numeric(ci_ul),
       cidi_name = factor(cidi_name, levels = cidi_name_order)) %>%
  ggplot(aes(x = cidi_name, y = OR, ymin = ci_ll, ymax = ci_ul,
             color = coef_name)) +
  facet_wrap(~trauma_type) +
  geom_pointrange(position = position_dodge(width = 0.7)) +
  geom_hline(yintercept = 1, lty = 2) +
  theme_bw() +
  theme(
    legend.position = "right"
  ) +
  labs(
    y = "OR (99.7% CI) / OR (95% CI) for grouped ",
    x = "",
    color = "Trauma Type",
    title = "Model D (Main categorical model)" +
    ylim(0,5) +
    coord_flip()

df_trauma_only %>% filter(model %in% c("Model E")) %>%
mutate(OR = as.numeric(OR),
       ci_ll = as.numeric(ci_ll),
       ci_ul = as.numeric(ci_ul),
       cidi_name = factor(cidi_name, levels = cidi_name_order)) %>%
  ggplot(aes(x = cidi_name, y = OR, ymin = ci_ll, ymax = ci_ul,
             color = trauma_type)) +
  geom_pointrange(position = position_dodge(width = 0.7)) +
  geom_hline(yintercept = 1, lty = 2) +
  theme_bw() +

```

```

facet_wrap(~trauma_type) +
theme(
  legend.position = "right"
) +
labs(
  y = "OR (99.7% CI) / OR (95% CI) for grouped ",
  x = "",
  color = "Trauma Type",
  title = "Model E (Substance use cont. model)"
) + coord_flip()

dev.off()

```

```

## pdf
## 2

```

Regression - update 1-13-25 (current)

Update from 1-13-25

Redoing the above model such that:

Here I make a df with all data from all tables.

We loop through each model with each specification and then we loop through each chronic condition in list `cidi_list_binary`.

We look at the following outcomes: Chronic pain grouping, Cardiometabolic conditions grouping, Respiratory conditions grouping, Neurological conditions grouping, Stomach ulcer, Allergies, Hypothyroidism

For these 7 plus the poisson model we do an Bonferonni correction with $\alpha/8$. For outcomes with prevalence > 10%, we run a relative risk regression. Otherwise we run logistic regression.

We then edit the results tibble into a dataframe and modify some variables in prep to make the figures.

Andrew R Notes on Poisson for Relative Risk Regression - 01-13-25:

Hi Hayden,

Confirmed, you run relative risk regression with the command:

```
fit<-glm(fmla, data=df, family=poisson(link="log"))
```

Note: I think I mentioned in a previous email that when you do relative risk regression using a Poisson model that you need to use robust standard errors. I sent a link previously but there are a couple more at the end of the email. This means you cannot use the `confint()` command to calculate the confidence intervals. You have to manually calculate the standard errors and critical values. For robust standard errors, the specific command is:

```
rse <- coeftest(fit, vcov = vcovHC(fit, type="HC1"))
```

That's going to get you the standard error. To calculate the Bonferroni CI you need to get the right critical value (it is not 1.96; that's the 95% confidence level). The correct critical value will be

```
qnorm(p=.05/(2*# of hypotheses tested), lower.tail=FALSE)
```

I think the # of hypotheses tested is 8, correct? 7 grouped outcomes and then the count? If so, your critical value is 2.734369 and you can calculate the CIs as $\beta \pm 2.734369 \times \text{robust standard error}$.

I would not hard code the value 2.734369. As before, make sure you define the number of tests and calculate the alpha based on that in case things change in the future.

Let me know if you have any questions about the above.

Best,

Andrew

<https://cran.r-project.org/web/packages/sandwich/sandwich.pdf> <https://data.princeton.edu/wws509/r/robust> <https://stats.stackexchange.com/questions/117052/replicating-stata-robust-option-in-r>

```
cidi_list_binary <- c(
  "cidi_q5_binary", "cidi_q14_binary", "cidi_q18_binary",
  "cidi_pain_binary", "cidi_cardiac_binary",
  "cidi_resp_binary", "cidi_neur_binary"
)

bon_cor <- length(cidi_list_binary) + 1

results <- NULL
df <- NULL

models <- c("Model B", "Model C", "Model D", "Model E")

# All Trauma

for (model in models) {
  if (model == "Model B") {
    trauma_var <- "trauma_load_inter_cat + trauma_load_noninter_nhs_cat"
    covar <- "+ age_at_interview + is_male_f + ses_status_f +
      site_location_group_f"
    type <- "Categorical"
  } else if (model == "Model C") {
    trauma_var <- "trauma_load_interpersonal_1 + trauma_load_noninter_1_nhs"
    covar <- "+ age_at_interview + is_male_f + ses_status_f +
      site_location_group_f"
    type <- "Numeric"
  } else if (model == "Model D") {
    trauma_var <- "trauma_load_inter_cat + trauma_load_noninter_nhs_cat"
    covar <- "+ age_at_interview + is_male_f + ses_status_f +
      site_location_group_f + factor(assist_tobacco_amt_recode) +
      factor(assist_alcohol_amt_recode) + factor(assist_cannabis_amt_recode)"
    type <- "Numeric"
  } else if (model == "Model E") {
    trauma_var <- "trauma_load_interpersonal_1 + trauma_load_noninter_1_nhs"
    covar <- "+ age_at_interview + is_male_f + ses_status_f +
      factor(assist_tobacco_amt_recode) +
      factor(assist_alcohol_amt_recode) +
      factor(assist_cannabis_amt_recode) + site_location_group_f"
    type <- "Numeric"
  }
}

for (ii in 1:length(cidi_list_binary)) {
```

```

#print(model)
#print(cidi_list_binary[ii])
tab <- table(nmex_cases[cidi_list_binary[ii]])
prev <- tab[2] / (tab[1] + tab[2])
prev_round <- round(prev,3)
cidi_var <- cidi_list_binary[ii]
# Get number of cases
cidi_var_n <- table(nmex_cases[[cidi_var]])[2]

if (prev <= 0.1) {
  fmla <- as.formula(paste(
    cidi_list_binary[ii], " ~ ",
    trauma_var, covar
  ))
  # Fit the model for logistic regression
  fit <- glm(fmla, data = nmex_cases, family = binomial)

  # get OR
  fixef <- exp(coef(fit))

  # Get confidence interval based on bonferonni
  conf <- exp(confint(fit, level = (1 - 0.05 / bon_cor)))

  # Extract CI of all values
  conf_complete <- conf[complete.cases(conf), ]

  # Get all coefficient names and levels
  var <- rownames(conf_complete)

  # For j in the number of coefficients (var)
  for (j in 1:length(var)) {
    reg_type <- "Logistic Regression"
    var_temp <- var[j]

    est <- round(exp(summary(fit)[["coefficients"]][
      var[j],
      "Estimate"
    ]), 3)
    # est <- round(exp(summary(fit)[var[j]]), 3)
    ci_ll <- round(conf_complete[var[j], 1], 3)
    ci_ul <- round(conf_complete[var[j], 2], 3)
    ci <- paste0("(", ci_ll, ", ", ci_ul, ")")

    p <- summary(fit)$coefficients[var[j], "Pr(>|z|)"]
    #print(paste("P val log reg:", p))

    model <- model

    if (p < (0.05 / bon_cor)) {
      pval <- "Significant"
    } else {
      pval <- "Not significant"
    }
  }
}

```



```

      results <- rbind(results, c(
        cidi_var, var_temp, prev_round,
        est, ci_ll, ci_ul, p, pval,
        model, cidi_var_n, type, reg_type
      ))
    }
  } else {
    # if prev > 10%, do relative risk regression
    reg_type <- "Relative Risk"
    fmla <- as.formula(paste(
      cidi_list_binary[ii], " ~ ",
      trauma_var, covar
    ))
    # Fit the model for relative risk regression (poisson, log link).
    fit <- glm(fmla, data = nmex_cases, family = poisson(link = "log"))

    rse <- coeftest(fit, vcov = vcovHC(fit, type = "HC1"))
    q <- qnorm(p = .05 / (2 * bon_cor), lower.tail = FALSE)

    # Extract beta, st_error, and p_val

    estimate_complete <- rse[complete.cases(rse), 1]
    rob_std_error_complete <- rse[complete.cases(rse), 2]
    p_val_rob <- rse[complete.cases(rse), 4]

    var <- rownames(rse)

    for (j in 1:length(var)) {
      var_temp <- var[j]
      beta <- estimate_complete[var[j]]
      est <- round(exp(beta),3)
      robust_se <- rob_std_error_complete[var[j]]
      p <- p_val_rob[var[j]]
      #print(paste("P val rel risk reg:", p))

      if (p < (0.05 / bon_cor)) {
        pval <- "Significant"
      } else {
        pval <- "Not significant"
      }

      ci_ll <- round(exp(beta - q * robust_se),3)
      ci_ul <- round(exp(beta + q * robust_se),3)

      results <- rbind(results, c(
        cidi_var, var_temp, prev_round,
        est, ci_ll, ci_ul, p, pval,
        model, cidi_var_n, type, reg_type
      ))
    }
  }
}

```

```

}
}

#Make results a dataframe and relabel
df <- as.data.frame(results)
colnames(df)<-c("cidi_q", "Coefficient", "prevalence",
               "OR_or_RR", "ci_ll", "ci_ul", "p", "significant",
               "model", "cidi_var_n", "type", "regression_type")

df <- df %>% mutate(cidi_name = case_when(
  cidi_q == "cidi_q1_binary" ~ "Arthritis/Rheumatism",
  cidi_q == "cidi_q2_binary" ~ "Back/neck problems",
  cidi_q == "cidi_q3_binary" ~ "Headaches",
  cidi_q == "cidi_q4_binary" ~ "Other pain",
  cidi_q == "cidi_q5_binary" ~ "Seasonal allergies",
  cidi_q == "cidi_q6_binary" ~ "Stroke",
  cidi_q == "cidi_q7_binary" ~ "Heart attack",
  cidi_q == "cidi_q8_binary" ~ "Heart disease",
  cidi_q == "cidi_q9_binary" ~ "High Blood Pressure",
  cidi_q == "cidi_q10_binary" ~ "Asthma",
  cidi_q == "cidi_q11_binary" ~ "Tuberculosis",
  cidi_q == "cidi_q12_binary" ~ "Lung disease",
  cidi_q == "cidi_q13_binary" ~ "Diabetes",
  cidi_q == "cidi_q14_binary" ~ "Stomach/intestine ulcer",
  cidi_q == "cidi_q15_binary" ~ "HIV/AIDS",
  cidi_q == "cidi_q16_binary" ~ "Epilepsy/Seizures",
  cidi_q == "cidi_q17_binary" ~ "Cancer",
  cidi_q == "cidi_q18_binary" ~ "Hypothyroidism",
  cidi_q == "cidi_pain_binary" ~ "Pain",
  cidi_q == "cidi_cardiac_binary" ~ "Cardiometabolic",
  cidi_q == "cidi_resp_binary" ~ "Respiratory",
  cidi_q == "cidi_neur_binary" ~ "Neurological",

),

coef_name = factor(case_when(
  Coefficient == "trauma_load_inter_cat1" ~ "1",
  Coefficient == "trauma_load_inter_cat2" ~ "2",
  Coefficient == "trauma_load_inter_cat>=3" ~ "3+",
  Coefficient == "trauma_load_noninter_nhs_cat1" ~ "1",
  Coefficient == "trauma_load_noninter_nhs_cat2" ~ "2",
  Coefficient == "trauma_load_noninter_nhs_cat>=3" ~ "3+",
  Coefficient == "trauma_load_interpersonal_1" ~ "Continuous",
  Coefficient == "trauma_load_noninter_1_nhs" ~ "Continuous"
), levels = c("1",
              "2",
              "3+",
              "Continuous")),

type = factor(type, levels = c("Numeric", "Categorical")),

```

```

trauma_type = factor(case_when(
  grepl("noninter", Coefficient) ~ "Non-interpersonal",
  grepl("inter_|interpersonal", Coefficient) ~ "Interpersonal"
), levels = c("Non-interpersonal", "Interpersonal"))

df_trauma_only <- df %>%
  filter(str_starts(Coefficient, "trauma_load"))

```

Format data for tables Here I format some of the data for the table which can be found here: https://docs.google.com/spreadsheets/d/1zhTdSART7JBKzzx_rPKCvSLNfKM5NfJTbV29twtnFIk/edit?gid=1299043405#gid=1299043405

Figures: ORs by model by chronic condition Here we loop through each of the 18 chronic conditions as well as the grouped chronic conditions and pool all results for Model B, C, D, and E.

In the title there will be the number of participants who endorsed the conditions.

```

#Ensure correct path
setwd("/Users/ham593/Dropbox (Harvard University)/NeuroMex/Data Analysis/Trauma Abstract_1-24/CIDI_LEC

outs<-unique(df_trauma_only$cidi_name)

pdf(paste0("NMEX_modelBCDE_allconditions_",today(),".pdf"))

for (i in outs) {

  tdf<-df_trauma_only[df_trauma_only$cidi_name==i, ]
  tdf$OR <- as.numeric(tdf$OR_or_RR)
  tdf$ci_ll <- as.numeric(tdf$ci_ll)
  tdf$ci_ul <- as.numeric(tdf$ci_ul)
  cidi_var_n <- tdf$cidi_var_n

  p <- ggplot(tdf, aes(x = coef_name, y = OR, ymin = ci_ll, ymax = ci_ul,
    color = trauma_type)) +
    geom_pointrange(position = position_dodge(width = 0.7)) +
    facet_wrap(~ model, nrow = 1, scales = 'free_x') +
    geom_hline(yintercept = 1, lty = 2) +
    theme_bw() +
    theme(
      axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1),
      legend.position = "right"
    ) +
    #scale_fill_manual(name = "Coefficient Type", ) +

  labs(
    y = "OR/RR (99.7% CI)",
    x = "Model",
    color = "Trauma Type"
  ) +

```

```

  ggtitle(paste(i, "; N =", cidi_var_n))

  print(p)
}

dev.off()

```

```

## pdf
## 2

```

Figures: OR Model B and Model C (all CIDI together) Here we show all chronic conditions in the same figure for easy comparison for all four models. We separate outcomes for interpersonal and nonintepersonal trauma.

```

setwd("/Users/ham593/Dropbox (Harvard University)/NeuroMex/Data Analysis/Trauma Abstract_1-24/CIDI_LEC

cidi_name_order = c( "Arthritis/Rheumatism",
  "Back/neck problems",
  "Headaches",
  "Other pain",
  "Seasonal allergies",
  "Stroke",
  "Heart attack",
  "Heart disease",
  "High Blood Pressure",
  "Asthma",
  "Tuberculosis",
  "Lung disease",
  "Diabetes",
  "Stomach/intestine ulcer",
  "HIV/AIDS",
  "Epilepsy/Seizures",
  "Cancer",
  "Hypothyroidism",
  "Pain",
  "Cardiometabolic",
  "Respiratory",
  "Neurological"
)

model_nosub <- df_trauma_only %>% filter(model %in% c("Model C")) %>%
  mutate(OR = as.numeric(OR_or_RR),
    ci_ll = as.numeric(ci_ll),
    ci_ul = as.numeric(ci_ul),
    cidi_name = factor(cidi_name, levels = cidi_name_order))

outs<-unique(model_nosub$cidi_name)

pdf(paste0("Compiled results_modelBCDE_",today(),".pdf"))

```

```

# model_nosub <- df_trauma_only %>% filter(model %in% c("Model B")) %>%
# mutate(OR = as.numeric(OR),
#        ci_ll = as.numeric(ci_ll),
#        ci_ul = as.numeric(ci_ul),
#        cidi_name = factor(cidi_name, levels = cidi_name_order))

df_trauma_only %>% filter(model %in% c("Model B")) %>%
mutate(OR = as.numeric(OR_or_RR),
       ci_ll = as.numeric(ci_ll),
       ci_ul = as.numeric(ci_ul),
       cidi_name = factor(cidi_name, levels = cidi_name_order)) %>%
ggplot(aes(x = cidi_name, y = OR, ymin = ci_ll, ymax = ci_ul,
           color = coef_name)) +
  facet_wrap(~trauma_type) +
  geom_pointrange(position = position_dodge(width = 0.7)) +
  geom_hline(yintercept = 1, lty = 2) +
  theme_bw() +
  theme(
    legend.position = "right"
  ) +
  labs(
    y = "OR (99.7% CI) / OR (95% CI) for grouped ",
    x = "",
    color = "Trauma Type",
    title = "Model B (Main categorical model)" +
    ylim(0,5) +
    coord_flip()

df_trauma_only %>% filter(model %in% c("Model C")) %>%
mutate(OR = as.numeric(OR_or_RR),
       ci_ll = as.numeric(ci_ll),
       ci_ul = as.numeric(ci_ul),
       cidi_name = factor(cidi_name, levels = cidi_name_order)) %>%
ggplot(aes(x = cidi_name, y = OR, ymin = ci_ll, ymax = ci_ul,
           color = trauma_type)) +
  geom_pointrange(position = position_dodge(width = 0.7)) +
  geom_hline(yintercept = 1, lty = 2) +
  theme_bw() +
  theme(
    legend.position = "right"
  ) +
  labs(
    y = "OR (99.7% CI) / OR (95% CI) for grouped ",
    x = "",
    color = "Trauma Type",
    title = "Model C (Main cont. model)"
  ) + coord_flip()

```

```

df_trauma_only %>% filter(model %in% c("Model D")) %>%
mutate(OR = as.numeric(OR_or_RR),
       ci_ll = as.numeric(ci_ll),
       ci_ul = as.numeric(ci_ul),
       cidi_name = factor(cidi_name, levels = cidi_name_order)) %>%
ggplot(aes(x = cidi_name, y = OR, ymin = ci_ll, ymax = ci_ul,
          color = coef_name)) +
  facet_wrap(~trauma_type) +
  geom_pointrange(position = position_dodge(width = 0.7)) +
  geom_hline(yintercept = 1, lty = 2) +
  theme_bw() +
  theme(
    legend.position = "right"
  ) +
  labs(
    y = "OR (99.7% CI) / OR (95% CI) for grouped ",
    x = "",
    color = "Trauma Type",
    title = "Model D (Main categorical model)" +
    ylim(0,5) +
    coord_flip()

```

```

df_trauma_only %>% filter(model %in% c("Model E")) %>%
mutate(OR = as.numeric(OR_or_RR),
       ci_ll = as.numeric(ci_ll),
       ci_ul = as.numeric(ci_ul),
       cidi_name = factor(cidi_name, levels = cidi_name_order)) %>%
ggplot(aes(x = cidi_name, y = OR, ymin = ci_ll, ymax = ci_ul,
          color = trauma_type)) +
  geom_pointrange(position = position_dodge(width = 0.7)) +
  geom_hline(yintercept = 1, lty = 2) +
  theme_bw() +
  facet_wrap(~trauma_type) +
  theme(
    legend.position = "right"
  ) +
  labs(
    y = "OR (99.7% CI) / OR (95% CI) for grouped ",
    x = "",
    color = "Trauma Type",
    title = "Model E (Substance use cont. model)"
  ) + coord_flip()

```

```
dev.off()
```

```

## pdf
## 2

```

Figure: OR/RR Model B and C with significance Here we modify Andrew R's code to create a figure of all outcomes on one page.

```
setwd("/Users/ham593/Dropbox (Harvard University)/NeuroMex/Data Analysis/Trauma Abstract_1-24/CIDI_LEC

df_gg <- df_trauma_only %>% filter(model == "Model B")
df_gg$OR_or_RR <- as.numeric(as.character(df_gg$OR_or_RR))
df_gg$ci_ll <- as.numeric(as.character(df_gg$ci_ll))
df_gg$ci_ul <- as.numeric(as.character(df_gg$ci_ul))

pdf(paste0("Compiled results_ModelBCDE_",today(),".pdf"), width=8, height=6)

ggplot(df_gg, aes(x=coef_name, y=OR_or_RR, ymin=ci_ll, ymax=ci_ul,
                  shape=trauma_type, color=significant)) +
  geom_pointrange(position=position_dodge(0.5))+
  geom_hline(yintercept=1, lty=2)+
  coord_flip()+
  scale_color_manual(breaks=c("Not significant", "Significant"),
                    values=c("black", "red"),
                    name = "Significance") +
  scale_shape_manual(values = c(16, 17),
                    name = "Trauma Type") +
  scale_y_continuous(breaks = seq(0, 4, by = 1), limits = c(0, 4)) +
  facet_wrap(~cidi_name) +
  theme_bw() +
  guides(color = guide_legend(reverse=T))+
  xlab("Number of Traumatic Events")+
  ylab("OR/RR (99.7% CI)")+
  theme(legend.position = "bottom") +
  ggtitle("Model B: Categorical")

df_gg <- df_trauma_only %>% filter(model == "Model C")
df_gg$OR_or_RR <- as.numeric(as.character(df_gg$OR_or_RR))
df_gg$ci_ll <- as.numeric(as.character(df_gg$ci_ll))
df_gg$ci_ul <- as.numeric(as.character(df_gg$ci_ul))

ggplot(df_gg, aes(x=coef_name, y=OR_or_RR, ymin=ci_ll, ymax=ci_ul,
                  shape=trauma_type, color=significant)) +
  geom_pointrange(position=position_dodge(0.5))+
  geom_hline(yintercept=1, lty=2)+
  coord_flip()+
  scale_color_manual(breaks=c("Not significant", "Significant"),
                    values=c("black", "red"),
                    name = "Significance") +
  scale_shape_manual(values = c(16, 17),
                    name = "Trauma Type") +
  scale_y_continuous(breaks = seq(0, 4, by = 1), limits = c(0, 2)) +
  facet_wrap(~cidi_name) +
  theme_bw() +
  guides(color = guide_legend(reverse=T),
        shape = guide_legend(reverse=T))+
  xlab("Number of Traumatic Events")+
  ylab("OR/RR (99.7% CI)"+
```

```

theme(legend.position = "bottom") +
ggtitle("Model C: Continuous")

df_gg <- df_trauma_only %>% filter(model == "Model D")
df_gg$OR_or_RR <- as.numeric(as.character(df_gg$OR_or_RR))
df_gg$ci_ll <- as.numeric(as.character(df_gg$ci_ll))
df_gg$ci_ul <- as.numeric(as.character(df_gg$ci_ul))

ggplot(df_gg, aes(x=coef_name, y=OR_or_RR, ymin=ci_ll, ymax=ci_ul,
                  shape=trauma_type, color=significant)) +
  geom_pointrange(position=position_dodge(0.5))+
  geom_hline(yintercept=1, lty=2)+
  coord_flip()+
  scale_color_manual(breaks=c("Not significant", "Significant"),
                    values=c("black", "red"),
                    name = "Significance") +
  scale_shape_manual(values = c(16, 17),
                    name = "Trauma Type") +
  scale_y_continuous(breaks = seq(0, 4, by = 1), limits = c(0, 4)) +
  facet_wrap(~cidi_name) +
  theme_bw() +
  guides(color = guide_legend(reverse=T),
         shape = guide_legend(reverse=T))+
  xlab("Number of Traumatic Events")+
  ylab("OR/RR (99.7% CI)")+
  theme(legend.position = "bottom") +
  ggtitle("Model D: Categorical + Substance Use")

df_gg <- df_trauma_only %>% filter(model == "Model E")
df_gg$OR_or_RR <- as.numeric(as.character(df_gg$OR_or_RR))
df_gg$ci_ll <- as.numeric(as.character(df_gg$ci_ll))
df_gg$ci_ul <- as.numeric(as.character(df_gg$ci_ul))

ggplot(df_gg, aes(x=coef_name, y=OR_or_RR, ymin=ci_ll, ymax=ci_ul,
                  shape=trauma_type, color=significant)) +
  geom_pointrange(position=position_dodge(0.5))+
  geom_hline(yintercept=1, lty=2)+
  coord_flip()+
  scale_color_manual(breaks=c("Not significant", "Significant"),
                    values=c("black", "red"),
                    name = "Significance") +
  scale_shape_manual(values = c(16, 17),
                    name = "Trauma Type") +
  scale_y_continuous(breaks = seq(0, 4, by = 1), limits = c(0, 2)) +
  facet_wrap(~cidi_name) +
  theme_bw() +
  guides(color = guide_legend(reverse=T),
         shape = guide_legend(reverse=T))+
  xlab("Number of Traumatic Events")+
  ylab("OR/RR (99.7% CI)")+

```



```

theme(legend.position = "bottom") +
ggtitle("Model E: Continuous + Substance Use")

dev.off()

```

```

## pdf
## 2

```

Model B only (current) Andrew 01-17-25 Model B: I imagine is the main results figure. Model C, D, and E's results will be mentioned in the text. Model B figure needs to be updated in the following ways to be ready for publication: i. Need to add a reference point for 0 traumas ii. Y-axis ticks: Just label 0, 1, 2, 3+. Do NOT label them ("0 Traumas", "1 Trauma", "2 Traumas", etc). iii. Since I think the focus is on Interpersonal trauma, change the order of the trauma types in the plots. Interpersonal trauma should be placed above non-interpersonal trauma. Make sure the order of the names in the legend match the order in the plots so that non-interpersonal is first in the legend. iv. Facet names: "An ulcer in your stomach or intestine" and "Seasonal allergies like hay fever" should be shortened. v. For a final publication version, remove the title ("Model B...") vi. Change Figure to having 2 rows with 4 columns rather than 3x3 to minimize white space.

Andrew - 1-30-25

You are almost there with the Figure. Please make the following changes:

Remove the Title ("Associations of Interpersonal...") Remove the other text in the bottom right Remake the figure to be 2 rows by 4 columns rather than 4 rows by 2 columns Reorder the outcomes by Pain (row 1, column 1) Ulcers (row 1, column 2) Allergies (row 1, column 3) Neurological (row 1, column 4) Cardiometabolic (row 2, column 1) Respiratory (row 2, column 2) Hypothyroidism (row 2 column 3). Chronic conditions (row 2, column 4) In the Trauma Type legend, make sure Interpersonal comes before Non-Interpersonal since Interpersonal is above Non-interpersonal in the plots (which is the way it should be). Relabel the facet "Chronic Condition Count" to "Chronic condition count" to be consistent with the capitalization

Best,

Andrew

Note: df_gg_temp created here – important for full figure

```

#Ensure correct path
setwd("/Users/ham593/Dropbox (Harvard University)/NeuroMex/Data Analysis/Trauma Abstract_1-24/CIDI_LEC

df_gg <- df_trauma_only %>% filter(model == "Model B")

#Add reference rows for all in cidi_name
reference_row_total <- NULL
for (i in unique(df_gg$cidi_name)) {

  reference_row <- data.frame(cidi_q = NA,
                             Coefficient = NA,
                             prevalence = NA,
                             OR_or_RR = 1,
                             ci_ll = 1, ci_ul = 1,
                             p = NA, significant = "Not significant",
                             model="Model B",
                             cidi_var_n = NA,

```

```

      type = NA,
      regression_type = NA,
      cidi_name = i,
      coef_name = "0",
      trauma_type = "Interpersonal")

reference_row2 <- data.frame(cidi_q = NA,
      Coefficient = NA,
      prevalence = NA,
      OR_or_RR = 1,
      ci_ll = 1, ci_ul = 1,
      p = NA, significant = "Not significant",
      model="Model B",
      cidi_var_n = NA,
      type = NA,
      regression_type = NA,
      cidi_name = i,
      coef_name = "0",
      trauma_type = "Non-interpersonal")

reference_row_3 <- rbind(reference_row, reference_row2)
reference_row_total <- rbind(reference_row_total, reference_row_3)
}

#Ensure data is correct type
#df_gg <- df_trauma_only %>% filter(model == "Model B")
df_gg <- rbind(df_gg, reference_row_total)
df_gg$coef_name <- factor(df_gg$coef_name, levels = c("0", "1", "2", "3+"))
df_gg$OR_or_RR <- as.numeric(as.character(df_gg$OR_or_RR))
df_gg$ci_ll <- as.numeric(as.character(df_gg$ci_ll))
df_gg$ci_ul <- as.numeric(as.character(df_gg$ci_ul))

#Adjust to releval the data
cidi_name_order <- c("Pain", "Stomach/intestine ulcer",
      "Seasonal allergies", "Neurological",
      "Cardiometabolic", "Respiratory", "Hypothyroidism")

df_gg$cidi_name <- factor(df_gg$cidi_name, levels = cidi_name_order)

#Save this data for use later
#Will combine this with negative binomial model later
df_gg_temp <- df_gg

#Create the figure
pdf(paste0("Compiled results_ModelB_",today(),".pdf"), width=10, height=8)

ggplot(df_gg, aes(x=coef_name, y=OR_or_RR, ymin=ci_ll, ymax=ci_ul,
      shape=trauma_type, color=significant)) +

```

```

geom_pointrange(position=position_dodge(0.5))+
geom_hline(yintercept=1, lty=2)+
coord_flip()+
scale_color_manual(breaks=c("Not significant", "Significant"),
                    values=c("black", "red"),
                    name = "Significance") +
scale_shape_manual(values = c(16, 17),
                   name = "Trauma Type") +
scale_y_continuous(breaks = seq(0, 4, by = 1), limits = c(0, 4)) +
facet_wrap(~cidi_name, nrow=2) +
theme_bw() +
guides(color = guide_legend(reverse=T),
        shape = guide_legend(reverse=T))+
xlab("Number of Traumatic Events Experienced")+
ylab("OR/RR (99.7% CI)")+
theme(legend.position = "bottom")

dev.off()

```

```

## pdf
## 2

```

Model: Poisson (old)

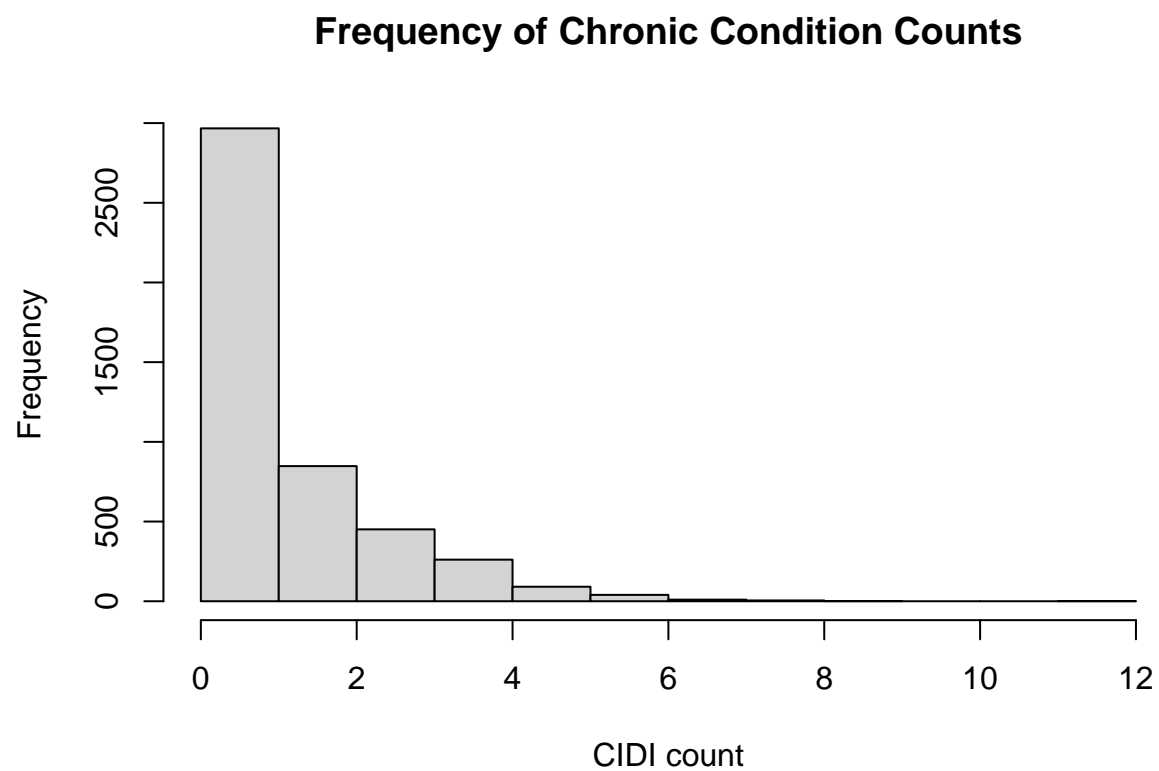
Here we run a poisson regression on chronic condition count

Histogram He we create a quick histogram of chronic condition counts.

```

hist(nmex_cases$cidi_load, main = "Frequency of Chronic Condition Counts",
     xlab = "CIDI count")

```

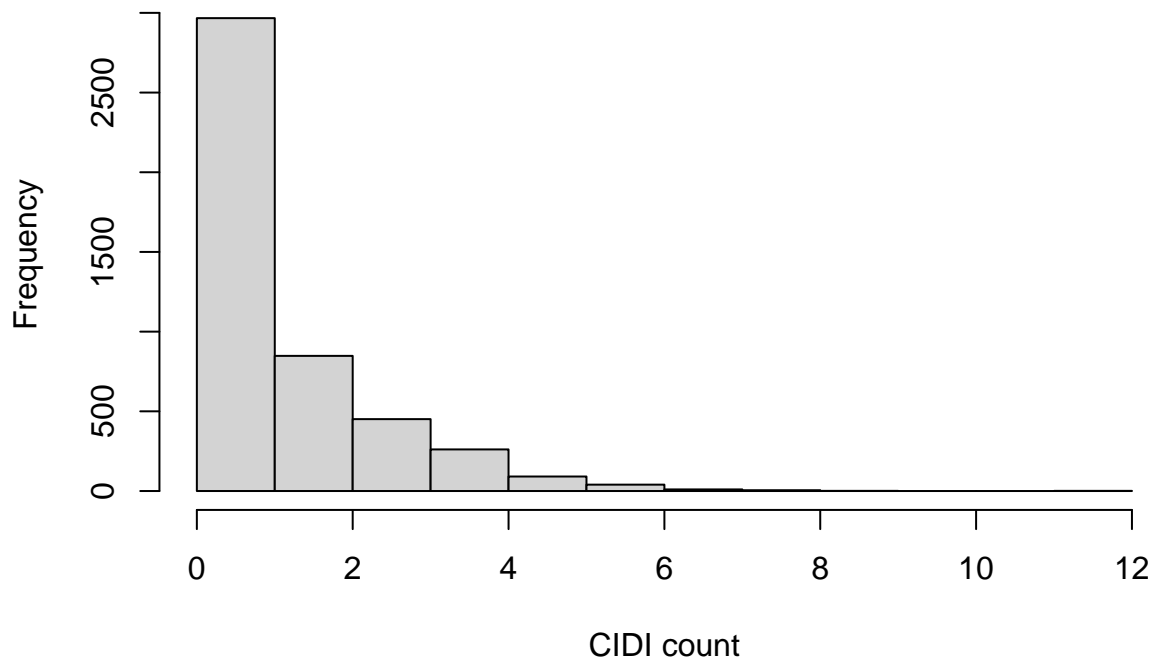


```
hist(nmex_cases$cidi_load, main = "Frequency of Chronic Condition Counts",  
     xlab = "CIDI count")
```

Table 27: Distribution of CIDI Count by Interpersonal Trauma Count

	0	1	2	>=3	Overall
	(N=2065)	(N=1184)	(N=796)	(N=630)	(N=4675)
factor(cidi_load)					
0	794 (38.5%)	428 (36.1%)	236 (29.6%)	166 (26.3%)	1624 (34.7%)
1	603 (29.2%)	356 (30.1%)	218 (27.4%)	166 (26.3%)	1343 (28.7%)
2	366 (17.7%)	198 (16.7%)	157 (19.7%)	127 (20.2%)	848 (18.1%)
3	159 (7.7%)	120 (10.1%)	97 (12.2%)	75 (11.9%)	451 (9.6%)
4	98 (4.7%)	53 (4.5%)	62 (7.8%)	48 (7.6%)	261 (5.6%)
5	27 (1.3%)	19 (1.6%)	18 (2.3%)	27 (4.3%)	91 (1.9%)
6	12 (0.6%)	8 (0.7%)	5 (0.6%)	15 (2.4%)	40 (0.9%)
7	4 (0.2%)	1 (0.1%)	1 (0.1%)	4 (0.6%)	10 (0.2%)
8	1 (0.0%)	1 (0.1%)	1 (0.1%)	2 (0.3%)	5 (0.1%)
9	0 (0%)	0 (0%)	1 (0.1%)	0 (0%)	1 (0.0%)
12	1 (0.0%)	0 (0%)	0 (0%)	0 (0%)	1 (0.0%)

Frequency of Chronic Condition Counts



```
table1(~ factor(cidi_load) | trauma_load_inter_cat, data=nmx_cases, caption = "Distribution of CIDI Co
```

```
table1(~ factor(cidi_load) | trauma_load_noninter_nhs_cat, data=nmx_cases, caption = "Distribution of C
```

Table 28: Distribution of CIDI Count by Non-interpersonal Trauma Count

	0	1	2	>=3	Overall
	(N=2284)	(N=1335)	(N=646)	(N=410)	(N=4675)
factor(cidi_load)					
0	846 (37.0%)	486 (36.4%)	184 (28.5%)	108 (26.3%)	1624 (34.7%)
1	676 (29.6%)	356 (26.7%)	195 (30.2%)	116 (28.3%)	1343 (28.7%)
2	406 (17.8%)	255 (19.1%)	110 (17.0%)	77 (18.8%)	848 (18.1%)
3	201 (8.8%)	126 (9.4%)	78 (12.1%)	46 (11.2%)	451 (9.6%)
4	104 (4.6%)	75 (5.6%)	46 (7.1%)	36 (8.8%)	261 (5.6%)
5	29 (1.3%)	24 (1.8%)	24 (3.7%)	14 (3.4%)	91 (1.9%)
6	13 (0.6%)	10 (0.7%)	7 (1.1%)	10 (2.4%)	40 (0.9%)
7	6 (0.3%)	1 (0.1%)	1 (0.2%)	2 (0.5%)	10 (0.2%)
8	2 (0.1%)	1 (0.1%)	1 (0.2%)	1 (0.2%)	5 (0.1%)
9	0 (0%)	1 (0.1%)	0 (0%)	0 (0%)	1 (0.0%)
12	1 (0.0%)	0 (0%)	0 (0%)	0 (0%)	1 (0.0%)

Poisson regressions (old)

He were create using Poisson regression outcomes for each model covariates

We treat chronic conditions (count) as the outcome variable.

Figure Here we create a figure compiling the above data.

Model Poission 1-13-25 (old)

Here we are updating the above code to match the bonferonni adjustments

```
results <- NULL
df <- NULL

models <- c("Model B", "Model C", "Model D", "Model E")

#All Trauma

for (model in models) {

  if (model == "Model B") {
    trauma_var <- "trauma_load_inter_cat + trauma_load_noninter_nhs_cat"
    covar <- "+ age_at_interview + is_male_f + ses_status_f +
site_location_group_f"
    type <- "Categorical"
  } else if (model == "Model C") {
    trauma_var <- "trauma_load_interpersonal_1 + trauma_load_noninter_1_nhs"
    covar <- "+ age_at_interview + is_male_f + ses_status_f +
site_location_group_f"
    type <- "Numeric"
  } else if (model == "Model D") {
    trauma_var <- "trauma_load_inter_cat + trauma_load_noninter_nhs_cat"
```

```

covar <- "+ age_at_interview + is_male_f + ses_status_f +
site_location_group_f + assist_tobacco_amt_recode +
assist_alcohol_amt_recode + assist_cannabis_amt_recode"
type <- "Categorical"
} else if (model == "Model E") {
trauma_var <- "trauma_load_interpersonal_1 + trauma_load_noninter_1_nhs"
covar <- "+ age_at_interview + is_male_f + ses_status_f +
          assist_tobacco_amt_recode + assist_alcohol_amt_recode +
          assist_cannabis_amt_recode + site_location_group_f"
type <- "Numeric"
}

fmla <- as.formula(paste("cidi_load", " ~ ",
                        trauma_var, covar))

# Fit the generalized linear regression model
fit <- glm(fmla, data = nmex_cases, family = poisson)

#####
print(model)

fixef <- exp(coef(fit))
conf <- exp(confint(fit, level = (1 - 0.05 / bon_cor)))
conf_complete <- conf[complete.cases(conf), ]

var <- rownames(conf_complete)

for (j in 1:length(var)) {
  var_temp <- var[j]

  est      <- round(exp(summary(fit)[["coefficients"]][var[j],
                                         "Estimate"]),3)

  ci_ll    <- round(conf_complete[var[j], 1],3)
  ci_ul    <- round(conf_complete[var[j], 2],3)
  ci <- paste0("(", ci_ll, ", ", ci_ul, ")")

  p        <- summary(fit)$coefficients[var[j], "Pr(>|z|)"]
  model <- model

  if (p < (0.05 / bon_cor)) {
    pval <- "Significant"
  } else {
    pval <- "Not significant"
  }

  results <- rbind(results,c(model, var_temp, est, ci_ll, ci_ul, p, pval,
                             type))
}
}

```

```
## [1] "Model B"
## [1] "Model C"
## [1] "Model D"
## [1] "Model E"

df <- as.data.frame(results)

colnames(df)<-c("Model","Coefficient","RR", "ci_ll", "ci_ul","p", "pval",
              "type")

df <- df %>% mutate(

  coef_name = factor(case_when(
    Coefficient == "trauma_load_inter_cat1" ~ "1 Trauma",
    Coefficient == "trauma_load_inter_cat2" ~ "2 Traumas",
    Coefficient == "trauma_load_inter_cat>=3" ~ "3+ Traumas",
    Coefficient == "trauma_load_noninter_nhs_cat1" ~ "1 Trauma",
    Coefficient == "trauma_load_noninter_nhs_cat2" ~ "2 Traumas",
    Coefficient == "trauma_load_noninter_nhs_cat>=3" ~ "3+ Traumas",
    Coefficient == "trauma_load_noninter_nhs_cat>=3" ~ "3+ Traumas",
    Coefficient == "trauma_load_interpersonal_1" ~ "Continuous",
    Coefficient == "trauma_load_noninter_1_nhs" ~ "Continuous"
  ), levels = c("1 Trauma",
                "2 Traumas",
                "3+ Traumas",
                "Continuous")),

  type = factor(type, levels = c("Numeric", "Categorical")),

  trauma_type = case_when(
    grepl("noninter", Coefficient) ~ "Non-interpersonal",
    grepl("inter_|interpersonal", Coefficient) ~ "Interpersonal"
  )
)

df_trauma_only <- df %>%
  filter(str_starts(Coefficient, "trauma_load"))
```

Format data for tables

Figure of Poisson regression (old) Here we create a figure compiling the above data for only Model B.

Update 01-21-2025 (old)

Here I will run the Poisson analysis for count data on model B only, as this seems like the most likely main model for our paper. I will run tests on the model for overdispersion and heteroskedacity.


```

bon_cor = 9
results <- NULL

models <- c("Model B")

#All Trauma

for (model in models) {

  if (model == "Model B") {
    trauma_var <- "trauma_load_inter_cat + trauma_load_noninter_nhs_cat"
    covar <- "+ age_at_interview + is_male_f + ses_status_f +
    site_location_group_f"
    type <- "Categorical"

    fmla <- as.formula(paste("cidi_load", " ~ ",
                             trauma_var, covar))

    # Fit the generalized linear regression model
    fit <- glm(fmla, data = nmex_cases, family = poisson)

    #####
    print(model)

    fixef <- exp(coef(fit))
    conf <- exp(confint(fit, level = (1 - 0.05 / bon_cor)))
    conf_complete <- conf[complete.cases(conf), ]

    var <- rownames(conf_complete)

    for (j in 1:length(var)) {

      var_temp <- var[j]

      est      <- round(exp(summary(fit)[["coefficients"]][var[j],
                                                "Estimate"]),3)

      ci_ll    <- round(conf_complete[var[j], 1],3)
      ci_ul    <- round(conf_complete[var[j], 2],3)
      ci <- paste0("(", ci_ll, ", ", ci_ul, ")")

      p        <- summary(fit)$coefficients[var[j], "Pr(>|z|)"]
      model <- model

      if (p < (0.05 / bon_cor)) {
        pval <- "Significant"
      } else {
        pval <- "Not significant"
      }
    }
  }
}

```

```

      results <- rbind(results,c(model, var_temp, est, ci_ll, ci_ul, p, pval,
                                type))
    }
  }
}

```

```
## [1] "Model B"
```

```

df <- as.data.frame(results)

colnames(df)<-c("Model","Coefficient","RR", "ci_ll", "ci_ul","p", "pval",
               "type")

df <- df %>% mutate(

  coef_name = factor(case_when(
    Coefficient == "trauma_load_inter_cat1" ~ "1 Trauma",
    Coefficient == "trauma_load_inter_cat2" ~ "2 Traumas",
    Coefficient == "trauma_load_inter_cat>=3" ~ "3+ Traumas",
    Coefficient == "trauma_load_noninter_nhs_cat1" ~ "1 Trauma",
    Coefficient == "trauma_load_noninter_nhs_cat2" ~ "2 Traumas",
    Coefficient == "trauma_load_noninter_nhs_cat>=3" ~ "3+ Traumas",
    Coefficient == "trauma_load_noninter_nhs_cat>=3" ~ "3+ Traumas",
    Coefficient == "trauma_load_interpersonal_1" ~ "Continuous",
    Coefficient == "trauma_load_noninter_1_nhs" ~ "Continuous"
  ), levels = c("1 Trauma",
                "2 Traumas",
                "3+ Traumas",
                "Continuous")),

  type = factor(type, levels = c("Numeric", "Categorical")),

  trauma_type = case_when(
    grepl("noninter", Coefficient) ~ "Non-interpersonal",
    grepl("inter_|interpersonal", Coefficient) ~ "Interpersonal"
  )
)

df_trauma_only <- df %>%
  filter(str_starts(Coefficient, "trauma_load"))

```

Here we see that there is overdispersion

```
dispersiontest(fit)
```

```

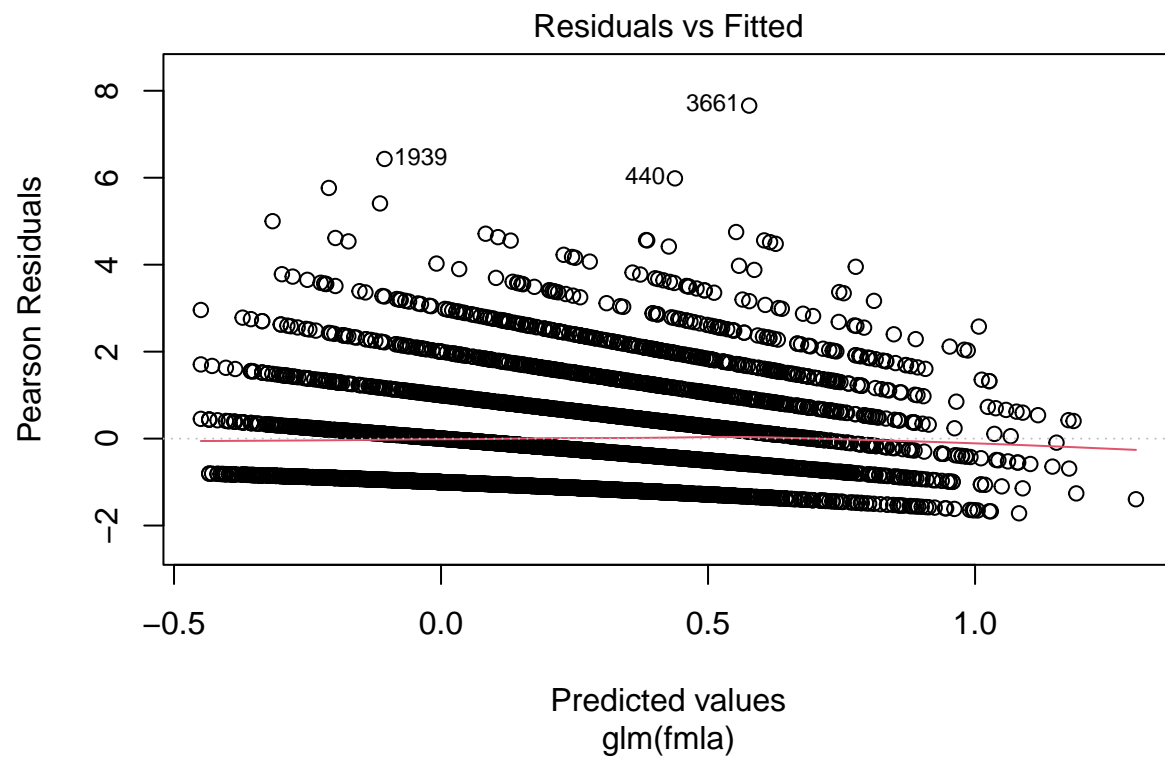
##
## Overdispersion test
##
## data: fit

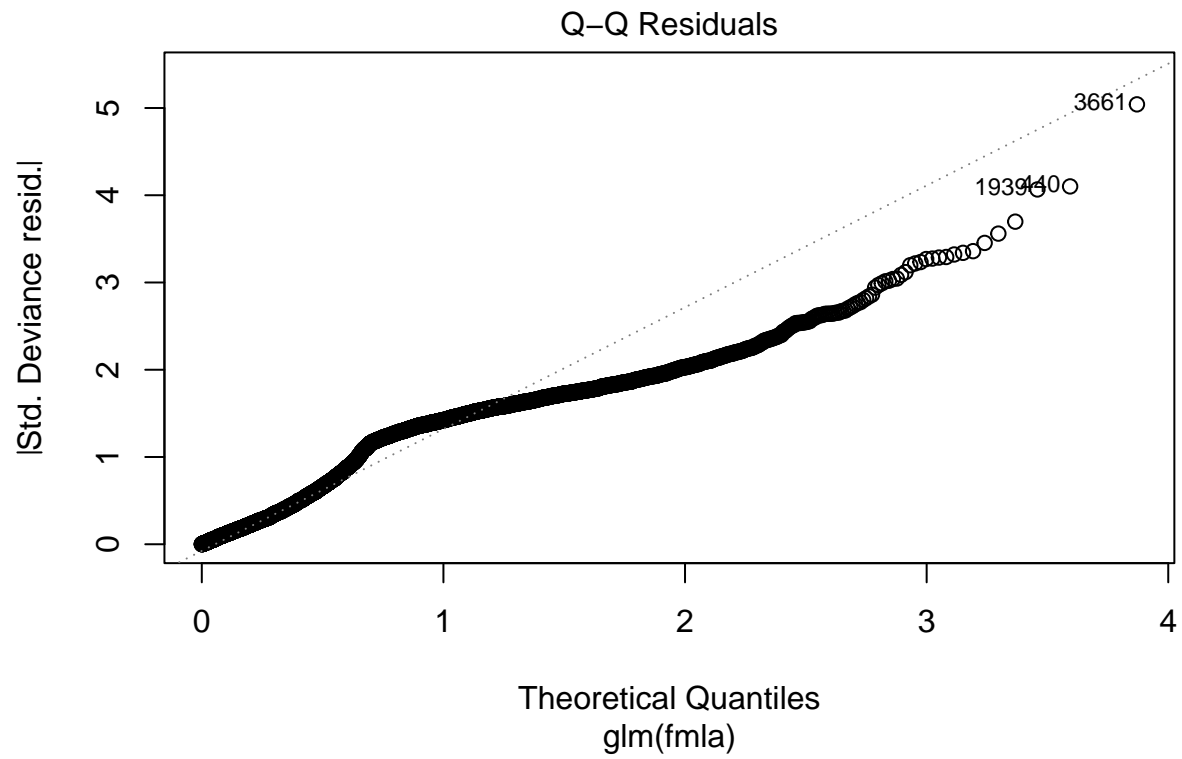
```

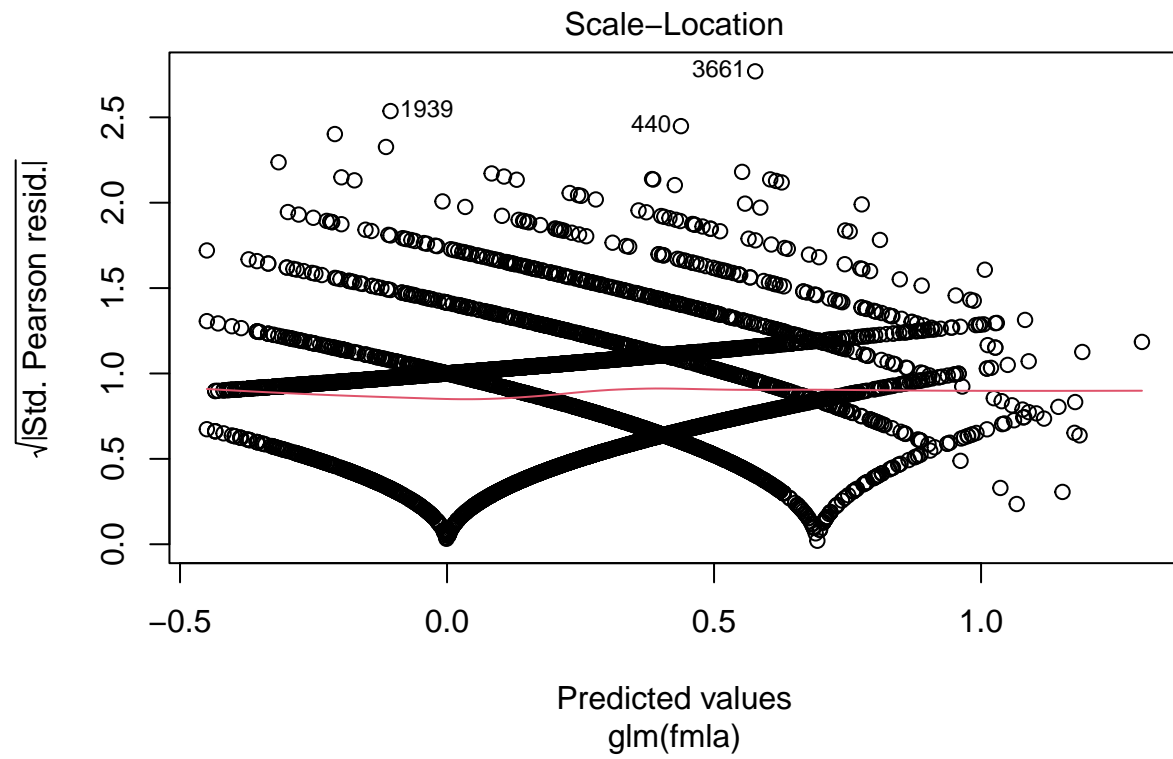
```
## z = 11.559, p-value < 2.2e-16
## alternative hypothesis: true dispersion is greater than 1
## sample estimates:
## dispersion
## 1.355786
```

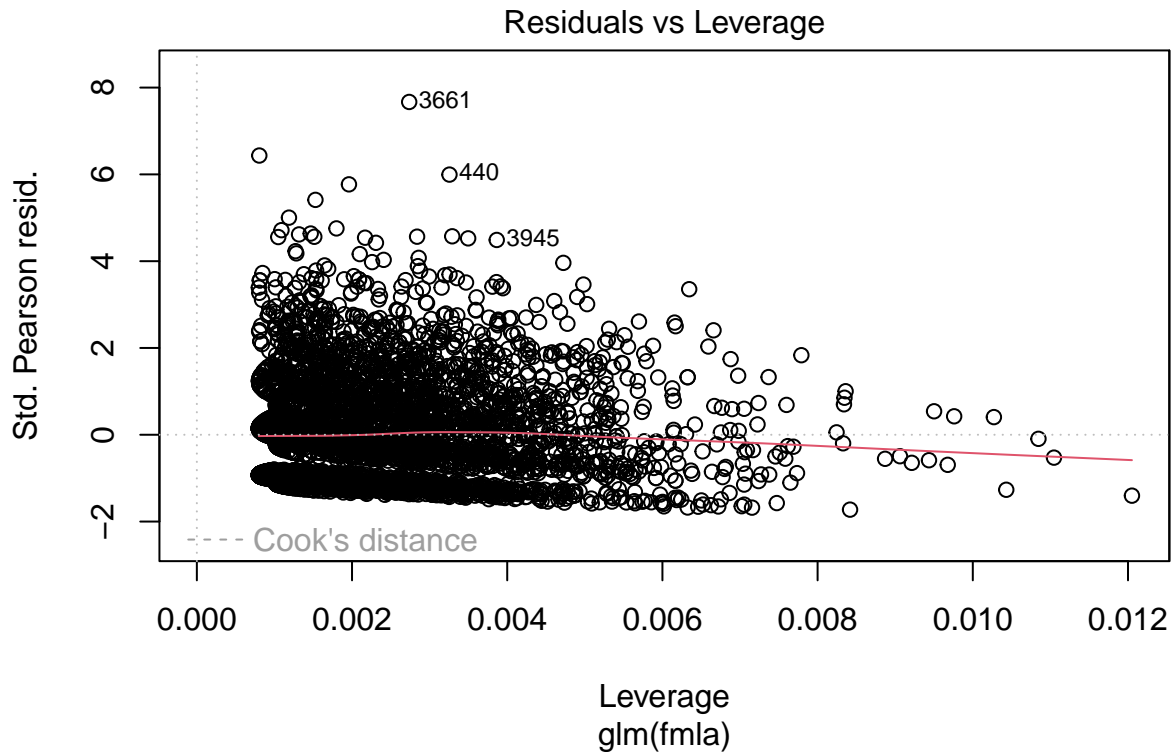
We can confirm this in this plot of Residuals vs Fitted values by the fan-shaped pattern.

```
plot(fit)
```









It might be worth trying a quasipoisson model for account for this overdispersion.

Option 1: Keep poisson and use RSE Option 2: Use quasi-poisson to account for overdispersion Option 3: Option 2 plus RSE? Option 4: Zero-inflated regression model

```
quasi_model <- glm(fmla, family = quasipoisson, data = nmex_cases)
```

```
summary(quasi_model)
```

```
##
## Call:
## glm(formula = fmla, family = quasipoisson, data = nmex_cases)
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.628603   0.070283  -8.944 < 2e-16 ***
## trauma_load_inter_cat1    0.040536   0.039196   1.034  0.30111
## trauma_load_inter_cat2    0.244879   0.042348   5.783 7.84e-09 ***
## trauma_load_inter_cat>=3    0.331103   0.045195   7.326 2.78e-13 ***
## trauma_load_noninter_nhs_cat1    0.027652   0.036370   0.760  0.44712
## trauma_load_noninter_nhs_cat2    0.162644   0.044380   3.665 0.00025 ***
## trauma_load_noninter_nhs_cat>=3    0.236791   0.052115   4.544 5.67e-06 ***
## age_at_interview    0.015635   0.001197  13.066 < 2e-16 ***
## is_male_fFemale    0.298758   0.030497   9.796 < 2e-16 ***
## ses_status_fMiddle-High    0.021067   0.031826   0.662  0.50804
```

```
## site_location_group_fUrban      -0.102590    0.038445   -2.669   0.00765 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 1.357139)
##
##      Null deviance: 7311.7  on 4602  degrees of freedom
## Residual deviance: 6710.0  on 4592  degrees of freedom
##      (72 observations deleted due to missingness)
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
```

```
summary(fit)
```

```
##
## Call:
## glm(formula = fmla, family = poisson, data = nmex_cases)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -0.628603    0.060331 -10.419 < 2e-16 ***
## trauma_load_inter_cat1    0.040536    0.033646   1.205  0.22829
## trauma_load_inter_cat2    0.244879    0.036351   6.736 1.62e-11 ***
## trauma_load_inter_cat>=3    0.331103    0.038795   8.535 < 2e-16 ***
## trauma_load_noninter_nhs_cat1  0.027652    0.031220   0.886  0.37577
## trauma_load_noninter_nhs_cat2  0.162644    0.038095   4.269 1.96e-05 ***
## trauma_load_noninter_nhs_cat>=3 0.236791    0.044735   5.293 1.20e-07 ***
## age_at_interview    0.015635    0.001027  15.222 < 2e-16 ***
## is_male_fFemale    0.298758    0.026179  11.412 < 2e-16 ***
## ses_status_fMiddle-High    0.021067    0.027320   0.771  0.44063
## site_location_group_fUrban    -0.102590    0.033001  -3.109  0.00188 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##      Null deviance: 7311.7  on 4602  degrees of freedom
## Residual deviance: 6710.0  on 4592  degrees of freedom
##      (72 observations deleted due to missingness)
## AIC: 14246
##
## Number of Fisher Scoring iterations: 5
```

```
dispersiontest(fit)
```

```
##
## Overdispersion test
##
## data: fit
## z = 11.559, p-value < 2.2e-16
## alternative hypothesis: true dispersion is greater than 1
## sample estimates:
```

```
## dispersion
## 1.355786
```

NB GLM -01-27-2025 (current)

After showing these above tests to Andrew, he confirmed that there does appear to be overdispersion. He suggests running a negative binomial regression instead. I do this below.

Calculate data NBGLM

Calculate the negative binomial GLM for all models.

```
results <- NULL
df <- NULL
models <- c("Model B", "Model C", "Model D", "Model E")

#All Trauma

for (model in models) {

  if (model == "Model B") {
    trauma_var <- "trauma_load_inter_cat + trauma_load_noninter_nhs_cat"
    covar <- "+ age_at_interview + is_male_f + ses_status_f +
site_location_group_f"
    type <- "Categorical"
  } else if (model == "Model C") {
    trauma_var <- "trauma_load_interpersonal_1 + trauma_load_noninter_1_nhs"
    covar <- "+ age_at_interview + is_male_f + ses_status_f +
site_location_group_f"
    type <- "Numeric"
  } else if (model == "Model D") {
    trauma_var <- "trauma_load_inter_cat + trauma_load_noninter_nhs_cat"
    covar <- "+ age_at_interview + is_male_f + ses_status_f +
site_location_group_f + assist_tobacco_amt_recode +
assist_alcohol_amt_recode + assist_cannabis_amt_recode"
    type <- "Categorical"
  } else if (model == "Model E") {
    trauma_var <- "trauma_load_interpersonal_1 + trauma_load_noninter_1_nhs"
    covar <- "+ age_at_interview + is_male_f + ses_status_f +
assist_tobacco_amt_recode + assist_alcohol_amt_recode +
assist_cannabis_amt_recode + site_location_group_f"
    type <- "Numeric"
  }

  fmla <- as.formula(paste("cidi_load", " ~ ",
trauma_var, covar))

  # Fit the negative binomial regression model
  fit <- glm.nb(fmla, data = nmex_cases)
```



```
#####
print(model)

fixef <- exp(coef(fit))
conf <- exp(confint(fit, level = (1 - 0.05 / bon_cor)))
conf <- exp(confint(fit))
conf_complete <- conf[complete.cases(conf), ]

var <- rownames(conf_complete)

for (j in 1:length(var)) {

  var_temp <- var[j]

  est      <- round(exp(summary(fit)[["coefficients"]][var[j],
                                                                    "Estimate"]),3)

  ci_ll    <- round(conf_complete[var[j], 1],3)
  ci_ul    <- round(conf_complete[var[j], 2],3)
  ci <- paste0("(", ci_ll, ", ", ci_ul, ")")

  p        <- summary(fit)$coefficients[var[j], "Pr(>|z|)"]
  model <- model

  if (p < (0.05 / bon_cor)) {
    significant <- "Significant"
  } else {
    significant <- "Not significant"
  }

  results <- rbind(results,c(model, var_temp, est, ci_ll, ci_ul, p, significant,
                             type))
}
}
```

```
## [1] "Model B"
```

```
## Waiting for profiling to be done...
```

```
## Waiting for profiling to be done...
```

```
## [1] "Model C"
```

```
## Waiting for profiling to be done...
```

```
## Waiting for profiling to be done...
```

```
## [1] "Model D"
```

```
## Waiting for profiling to be done...
```

```
## Waiting for profiling to be done...
```

```
## [1] "Model E"
```

```
## Waiting for profiling to be done...
## Waiting for profiling to be done...
```

```
df <- as.data.frame(results)

colnames(df)<-c("model","Coefficient","RR", "ci_ll", "ci_ul","p", "significant",
              "type")

df <- df %>% mutate(

  coef_name = factor(case_when(
    Coefficient == "trauma_load_inter_cat1" ~ "1",
    Coefficient == "trauma_load_inter_cat2" ~ "2",
    Coefficient == "trauma_load_inter_cat>=3" ~ "3+",
    Coefficient == "trauma_load_noninter_nhs_cat1" ~ "1",
    Coefficient == "trauma_load_noninter_nhs_cat2" ~ "2",
    Coefficient == "trauma_load_noninter_nhs_cat>=3" ~ "3+",
    Coefficient == "trauma_load_noninter_nhs_cat>=3" ~ "3+",
    Coefficient == "trauma_load_interpersonal_1" ~ "Continuous",
    Coefficient == "trauma_load_noninter_1_nhs" ~ "Continuous"
  ), levels = c("1",
                "2",
                "3+",
                "Continuous")),

  type = factor(type, levels = c("Numeric", "Categorical")),

  trauma_type = factor(case_when(
    grepl("noninter", Coefficient) ~ "Non-interpersonal",
    grepl("inter_|interpersonal", Coefficient) ~ "Interpersonal"
  ), levels = c("Non-interpersonal", "Interpersonal"))
)

df_trauma_only <- df %>%
  filter(str_starts(Coefficient, "trauma_load"))
```

Format data

Figure NB GLM Model B (current)

Here I create the figure for Model B's chronic condition count using the output produced above.

```
setwd("/Users/ham593/Dropbox (Harvard University)/NeuroMex/Data Analysis/Trauma Abstract_1-24/CIDI_LEC

pdf("NB_ModelB.pdf", width = 8, height = 6)

df_trauma_onlyB <- df_trauma_only %>% filter(model == "Model B")
df_trauma_onlyB$RR <- as.numeric(df_trauma_onlyB$RR)
df_trauma_onlyB$ci_ll <- as.numeric(df_trauma_onlyB$ci_ll)
df_trauma_onlyB$ci_ul <- as.numeric(df_trauma_onlyB$ci_ul)
```

```

ggplot(df_trauma_onlyB, aes(x = coef_name, y = RR, ymin = ci_ll,
                           ymax = ci_ul, shape=trauma_type, color=significant)) +
  geom_pointrange(position = position_dodge(width = 0.7)) +
  scale_color_manual(breaks=c("Not significant", "Significant"),
                    values=c("black", "red"),
                    name = "Significance") +
  scale_shape_manual(values = c(16, 17),
                    name = "Trauma Type") +
  geom_hline(yintercept = 1, lty = 2) +
  theme_bw() +
  theme(
    axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1),
    legend.position = "right"
  ) +
  guides(color = guide_legend(reverse=T),
         shape = guide_legend(reverse=T)) +
  scale_y_continuous(breaks = seq(0, 1.75, by = 0.25), limits = c(0.75, 1.6)) +
  labs(
    y = "RR (99.7% CI)",
    x = "Number of Traumatic Events Experienced",
    color = "Trauma Type"
  ) + coord_flip() +
  theme(legend.position = "bottom")

dev.off()

```

```

## pdf
## 2

```

Full Figure Model B + NBGLM (current) Here I combine data from the Logistic Regression/Poisson Regression with the data from the negative binomial regression.

I also manually add reference points of “0” traumas with OR/ci_ll/ci_ul of 1. Finally I relevel the data so the facets are in the correct order.

```

df_trauma_onlyB <- df_trauma_only %>% filter(model == "Model B")
df_trauma_onlyB$OR_or_RR <- df_trauma_onlyB$RR
df_trauma_onlyB$prevalence <- NA
df_trauma_onlyB$type <- NA
df_trauma_onlyB$prevalence <- NA
df_trauma_onlyB$regression_type <- NA
df_trauma_onlyB$cidi_name <- "Chronic condition count"
df_trauma_onlyB$cidi_q <- NA
df_trauma_onlyB$cidi_var_n <- NA
df_trauma_onlyB <- df_trauma_onlyB %>% dplyr::select(-RR)

reference_row <- data.frame(cidi_q = NA,
                           Coefficient = NA,
                           prevalence = NA,
                           OR_or_RR = 1,

```

```

      ci_ll = 1, ci_ul = 1,
      p = NA, significant = "Not significant",
      model="Model B",
      cidi_var_n = NA,
      type = NA,
      regression_type = NA,
      cidi_name = "Chronic condition count",
      coef_name = "0",
      trauma_type = "Non-interpersonal")
df_trauma_onlyB <- rbind(df_trauma_onlyB, reference_row)

reference_row <- data.frame(cidi_q = NA,
      Coefficient = NA,
      prevalence = NA,
      OR_or_RR = 1,
      ci_ll = 1, ci_ul = 1,
      p = NA, significant = "Not significant",
      model="Model B",
      cidi_var_n = NA,
      type = NA,
      regression_type = NA,
      cidi_name = "Chronic condition count",
      coef_name = "0",
      trauma_type = "Interpersonal")
df_trauma_onlyB <- rbind(df_trauma_onlyB, reference_row)

df_gg_temp_nb <- rbind(df_gg_temp, df_trauma_onlyB)

df_gg_temp_nb$coef_name <- factor(df_gg_temp_nb$coef_name, levels = c("0", "1", "2", "3+"))
df_gg_temp_nb$OR_or_RR <- as.numeric(as.character(df_gg_temp_nb$OR_or_RR))
df_gg_temp_nb$ci_ll <- as.numeric(as.character(df_gg_temp_nb$ci_ll))
df_gg_temp_nb$ci_ul <- as.numeric(as.character(df_gg_temp_nb$ci_ul))

#Adjust to relelevel the data
cidi_name_order <- c("Pain", "Stomach/intestine ulcer",
      "Seasonal allergies", "Neurological",
      "Cardiometabolic", "Respiratory", "Hypothyroidism", "Chronic condition count")

df_gg_temp_nb$cidi_name <- factor(df_gg_temp_nb$cidi_name, levels = cidi_name_order)

```

Here I produce the figure. The facets will be in 2 rows, with 4 columns.

```

setwd("/Users/ham593/Dropbox (Harvard University)/NeuroMex/Data Analysis/Trauma Abstract_1-24/CIDI_LEC

pdf(paste0("Compiled results_ModelB_withCCcount",today(),".pdf"), width=10, height=8)

ggplot(df_gg_temp_nb, aes(x=coef_name, y=OR_or_RR, ymin=ci_ll, ymax=ci_ul,

```

```

        shape=trauma_type, color=significant)) +
geom_pointrange(position=position_dodge(0.5))+
geom_hline(yintercept=1, lty=2)+
coord_flip()+
scale_color_manual(breaks=c("Not significant", "Significant"),
                    values=c("black", "red"),
                    name = "Significance") +
scale_shape_manual(values = c(16, 17),
                   name = "Trauma Type") +
scale_y_continuous(breaks = seq(0, 4, by = 1), limits = c(0, 4)) +
facet_wrap(~cidi_name, nrow=2) +
theme_bw() +
guides(color = guide_legend(reverse=T),
        shape = guide_legend(reverse=T))+
xlab("Number of Traumatic Events Experienced")+
ylab("OR/RR (99.7% CI)")+
theme(legend.position = "bottom")

dev.off()

```

```

## pdf
## 2

```