Data Preparation

```
# Load necessary libraries:
# ...
library(tidyverse)
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0
──
✔ dplyr     1.1.4     ✔ readr     2.1.5
✔ forcats   1.0.1     ✔ stringr   1.5.2
✔ ggplot2   4.0.0     ✔ tibble    3.3.0
✔ lubridate 1.9.4     ✔ tidyr     1.3.1
✔ purrr     1.1.0
── Conflicts ────────────────────────────────────────── tidyverse_conflicts()
──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
conflicts to become errors
```

```
library(readr)
library(dplyr)
library(ggplot2)

# Load CSV to R.
work_dir <- "/home/robot/rlab/er-visits/"
setwd(work_dir)
citbi <- read_csv("data/citbi.csv")
```

```
Rows: 30379 Columns: 26
── Column specification
────────────────────────────────────────────────────────
Delimiter: ","
dbl (26): PatNum, Amnesia_verb, LocLen, Seiz, SeizLen, ActNorm, HA_verb,
Vom...

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this
message.
```

```
spec(citbi)
```

```
cols(
  PatNum = col_double(),
  Amnesia_verb = col_double(),
  LocLen = col_double(),
  Seiz = col_double(),
  SeizLen = col_double(),
  ActNorm = col_double(),
  HA_verb = col_double(),
  Vomit = col_double(),
  Dizzy = col_double(),
  GCSEye = col_double(),
  GCSVerbal = col_double(),
  GCSMotor = col_double(),
  GCSTotal = col_double(),
  AMS = col_double(),
  SFxPalp = col_double(),
  FontBulg = col_double(),
  Hema = col_double(),
  Clav = col_double(),
  NeuroD = col_double(),
  OSI = col_double(),
  CTForm1 = col_double(),
  AgeInMonth = col_double(),
  Gender = col_double(),
  CTDone = col_double(),
  DeathTBI = col_double(),
  PosIntFinal = col_double()
)
```

Part I: Data Cleaning

1. Address Missing Data:

• How is missing data being represented? Hint: Look at the data dictionary. • Notice that some columns have a special 91 value that indicates a pre-verbal patients (children who haven't started speaking yet), some columns have a 92 that could be specific to that variable.

```
# Regard 91 (PV/NV) and 92 as NA.
citbi_na <- read_csv("data/citbi.csv", na = c("91", "92", "NA"))
```

```
Rows: 30379 Columns: 26
── Column specification
────────────────────────────────────────────────────────
```

```
Delimiter: ","
dbl (26): PatNum, Amnesia_verb, LocLen, Seiz, SeizLen, ActNorm, HA_verb,
Vom...

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this
message.
```

2. Change Variable Names:

• Consider changing variable names. For example, names like "Clav" are unclear. A better name could be "clavicle_trauma".

```
citbi_ren <- citbi_na |>
    rename(
        patient_number = PatNum,
        amnesia_verb = Amnesia_verb,
        duration_loss_of_consciousness = LocLen,
        post_traumatic_seizure = Seiz,
        duration_post_traumatic_seizure = SeizLen,
        acting_normally = ActNorm,
        headache_verb = HA_verb,
        vomit = Vomit,
        dizzy = Dizzy,
        gcs_eye = GCSEye,
        gcs_verb = GCSVerbal,
        gcs_motor = GCSMotor,
        gcs_total = GCSTotal,
        altered_mental_status = AMS,
        palpable_skull_fracture = SFxPalp,
        fontanelle_bulging = FontBulg,
        scalp_hematoma = Hema,
        clavicle_trauma = Clav,
        neuro_defect = NeuroD,
        other_substantial_injure = OSI,
        ct_form_filled = CTForm1,
        age_in_month = AgeInMonth,
        gender = Gender,
        ct_done = CTDone,
        death_due_to_tbi = DeathTBI,
        citbi_final = PosIntFinal

    )
```

3. Specify Variable Types:

• Identify character, logical, and numeric vectors. Convert data types where appropriate. * amnesia_verb : logical/character? * duration_loss_of_consciousness: character *

post_traumatic_seizure: logical * duration_post_traumatic_seizure: character * acting_normally : logical * headache_verb : logical * vomit : logical * dizzy : logical * gcs_eye : Character * gcs_verb : Character * gcs_motor : Character
* altered_mental_status : logical * palpable_skull_fracture : Character * fontanelle_bulging : Character * scalp_hematoma : Logical * clavicle_trauma : Logical
* neuro_defect : Logical * other_substantial_injure : Logical * ct_form_filled : Logical
* gender : Character
* ct_done : Logical * death_due_to_tbi : Logical * citbi_final : Logical

• Consider converting character vectors to factors.

```
citbi_update <- citbi_ren |>
    mutate(
        amnesia_verb = as.factor(
            recode(
                amnesia_verb,
                `0` = "No",
                `1` = "Yes",
                `91` = "Pre-verbal/Non-verbal",
                .default = NA_character_,
                .missing = NA_character_
            )
        ),
        duration_loss_of_consciousness = as.factor(
            factor(
                recode(
                    duration_loss_of_consciousness,
                    `1` = "< 5 sec",
                    `2` = "5 sec - < 1 min",
                    `3` = "1 - 5 min",
                    `4` = "> 5 min",
                    .default = NA_character_,
                    .missing = NA_character_
                ),
                levels = c("< 5 sec", "5 sec - < 1 min", "1 - 5 min","> 5
min"),
                ordered = TRUE,
            )
        ),
        post_traumatic_seizure = as.logical(post_traumatic_seizure),
        duration_post_traumatic_seizure = as.factor(
            recode(
                duration_post_traumatic_seizure,
                `1` = "< 1 min",
                `2` = "1 - < 5 min",
                `3` = "5 - `5 min",
                `4` = "> 15 min",
                .default = NA_character_,
```

```
            .missing = NA_character_
        )
    ),
    acting_normally = as.logical(acting_normally),
    headache_verb = as.factor(
        recode(
            headache_verb,
            `0` = "No",
            `1` = "Yes",
            `91` = "Pre-verbal/Non-verbal",
            .default = NA_character_,
            .missing = NA_character_
        )
    ),
    vomit = as.logical(vomit),
    dizzy = as.logical(dizzy),

    gcs_eye = as.factor(
        recode(
            gcs_eye,
            `1` = "None",
            `2` = "Pain",
            `3` = "Verbal",
            `4` = "Spontaneous",
            .default = NA_character_,
            .missing = NA_character_
        )
    ),
    gcs_verb = as.factor(
        recode(
            gcs_verb,
            `1` = "None",
            `2` = "Incomprehensible sounds (moans)",
            `3` = "Inappropriate words(cries to pain)",
            `4` = "Confused (irritable/cries)",
            `5` = "Oriented (coos/babbles)",
            .default = NA_character_,
            .missing = NA_character_
        )
    ),
    gcs_motor = as.factor(
        recode(
            gcs_motor,
            `1` = "None",
            `2` = "Abnormal extension posturing",
            `3` = "Abnormal flexure posituring",
            `4` = "Withdraws to pain",
            `5` = "Localizes pain (withdraws to touch)",
```

```
            `6` = "Follow commands (spontaneous movement)",
            .default = NA_character_,
            .missing = NA_character_
        )
    ),
    altered_mental_status = as.logical(altered_mental_status),
    palpable_skull_fracture = as.factor(
        recode(
            palpable_skull_fracture,
            `0` = "No",
            `1` = "Yes",
            `2` = "Unclear exam",
            .default = NA_character_,
            .missing = NA_character_
        )
    ),
    fontanelle_bulging = as.factor(
        recode(
            fontanelle_bulging,
            `0` = "No/Closed",
            `1` = "Yes",
            .default = NA_character_,
            .missing = NA_character_
        )
    ),
    scalp_hematoma = as.logical(scalp_hematoma),
    clavicle_trauma = as.logical(clavicle_trauma),
    neuro_defect = as.logical(neuro_defect),
    other_substantial_injure = as.logical(other_substantial_injure),
    ct_form_filled = as.logical(ct_form_filled),
    gender = as.factor(
        recode(
            gender,
            `1` = "Male",
            `2` = "Female",
            .default = NA_character_,
            .missing = NA_character_
        )
    ),
    ct_done = as.logical(ct_done),
    death_due_to_tbi = as.logical(death_due_to_tbi),
    citbi_final = as.logical(citbi_final)
)
```

Part II: Exploratory Data Analysis (EDA)

4. Create a Missing Data Summary:

• In Positron, navigate to the session tab in the top right where you can view a list of objects saved in your environment. Click the spreadsheet icon to the right of the data frame name you created in part I. You should be launched into a new tab with a split pane view. On one side, you will see mini-histograms of each variable along with the proportion of missing values. If you click onto the variables you can see some useful summary statistics. On the other pane, you can see the data frame itself.

• Now, create a table that summarizes the number and the proportion of missing values for each variable in the dataset. Does it match up with what you saw in the positron summary tab?

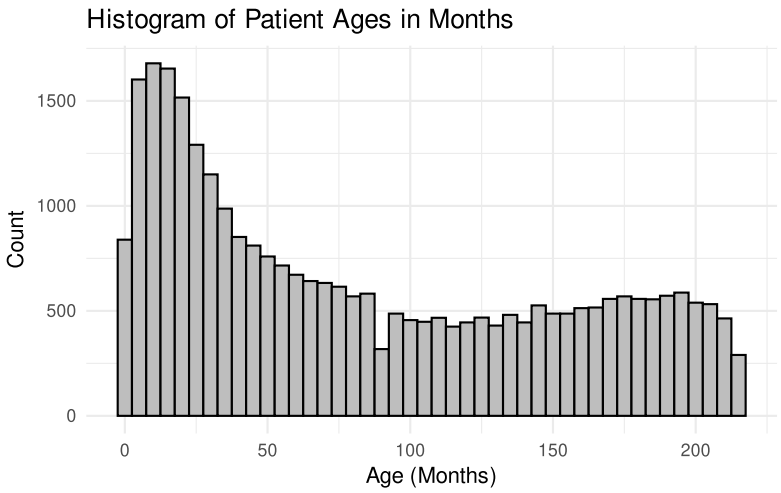
```
missing_table <- citbi_update |>
  summarise(
    cnt_amnesia_verb_missing = sum(is.na(amnesia_verb)),
    prop_amnesia_verb_missing = mean(is.na(amnesia_verb)),
    cnt_duration_loss_of_consciousness_missing =
sum(is.na(duration_loss_of_consciousness)),
    prop_duration_loss_of_consciousness_missing =
mean(is.na(duration_loss_of_consciousness)),
    #Can add more lines of code to check missing values.
  )

print(missing_table)
```

```
# A tibble: 1 × 4
  cnt_amnesia_verb_missing prop_amnesia_verb_missing
cnt_duration_loss_of_cons…¹
                     <int>                     <dbl>
<int>
1                    11933                     0.393
27398
# ℹ abbreviated name: ¹cnt_duration_loss_of_consciousness_missing
# ℹ 1 more variable: prop_duration_loss_of_consciousness_missing <dbl>
```

5. Create a histogram of the patient ages in months. Describe any interesting patterns you see.

```
citbi_update |>
ggplot(aes(x = age_in_month)) +
  geom_histogram(binwidth = 5, fill = "gray", color = "black") +
  labs(title = "Histogram of Patient Ages in Months", x = "Age (Months)", y =
"Count") +
  theme_minimal()
```

```
Warning: Removed 189 rows containing non-finite outside the scale range
(`stat_bin()`).
```




```
# Yonger babies are more dangous to be injured?
```

6. Create a grouped summary table that shows the total count of patients for every combination of the loss of consciousness length and ciTBI outcome columns.

```
citbi_grouped <- citbi_update |>
    filter(!is.na(duration_loss_of_consciousness) & !is.na(citbi_final)) |>
    group_by(duration_loss_of_consciousness, citbi_final) |>
    summarise(count = n())
```

```
`summarise()` has grouped output by 'duration_loss_of_consciousness'. You can
override using the `.groups` argument.
```

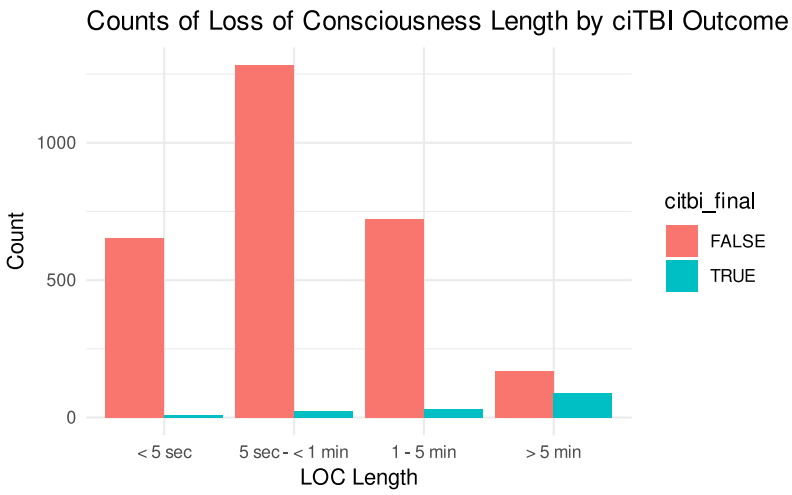
```
print(citbi_grouped)
```

```
# A tibble: 8 × 3
# Groups:   duration_loss_of_consciousness [4]
  duration_loss_of_consciousness citbi_final count
  <ord>                          <lgl>       <int>
1 < 5 sec                        FALSE         652
2 < 5 sec                        TRUE            8
3 5 sec - < 1 min                FALSE        1283
4 5 sec - < 1 min                TRUE           25
```

```
5 1 - 5 min                    FALSE         723
6 1 - 5 min                    TRUE           30
7 > 5 min                      FALSE         168
8 > 5 min                      TRUE           90
```

7. Create a bar chart to visualize the count for each category side-by-side in length of loss of consciousness by ciTBI outcome.

```
citbi_grouped |>
    ggplot( aes(x = duration_loss_of_consciousness, y = count, fill =
citbi_final )) +
    geom_bar(stat = "identity", position = "dodge") +
    labs(title = "Counts of Loss of Consciousness Length by ciTBI Outcome", x
= "LOC Length", y = "Count") +
    theme_minimal()
```



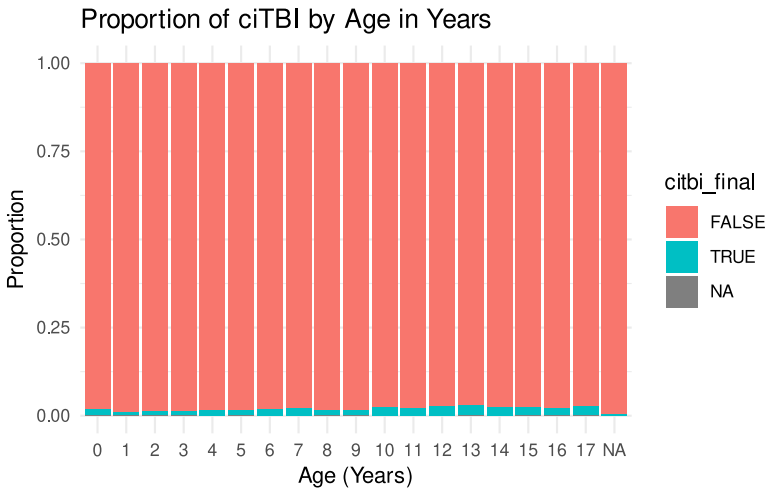
8. TTe variable (originally) named "GCSTotal" refers to the Glasgow Coma Scale (GCS) score, a neurological assessment used to evaluate the patient's level of consciousness. Create three visualizations for the relationship between total GCS score, Age in years, and ciTBI outcome.

```
citbi_gcs_age <- citbi_update |>
    mutate(age_in_year = age_in_month %/%12) |>
    select(age_in_year, gcs_total, citbi_final)

print(citbi_gcs_age)
```

```
# A tibble: 30,379 × 3
   age_in_year gcs_total citbi_final
         <dbl>     <dbl> <lgl>
 1           0        15 FALSE
 2           1        15 FALSE
 3          17        15 FALSE
 4          13        15 FALSE
 5          16        15 FALSE
 6           8        15 FALSE
 7           8        15 FALSE
 8          10        15 FALSE
 9           1        15 FALSE
10           1        15 FALSE
# ℹ 30,369 more rows
```

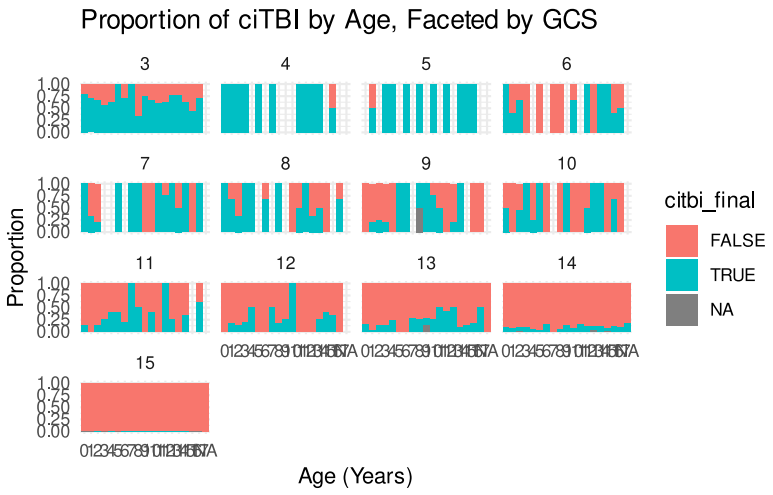
- a. Create a stacked normalized bar chart to visualize how the proportion of patients with ciTBI varies across different ages.

```
citbi_gcs_age |>
    ggplot(aes(x = factor(age_in_year), fill = citbi_final)) +
        geom_bar(position = "fill") +
        labs(title = "Proportion of ciTBI by Age in Years", x = "Age (Years)",
y = "Proportion") +
        theme_minimal()
```



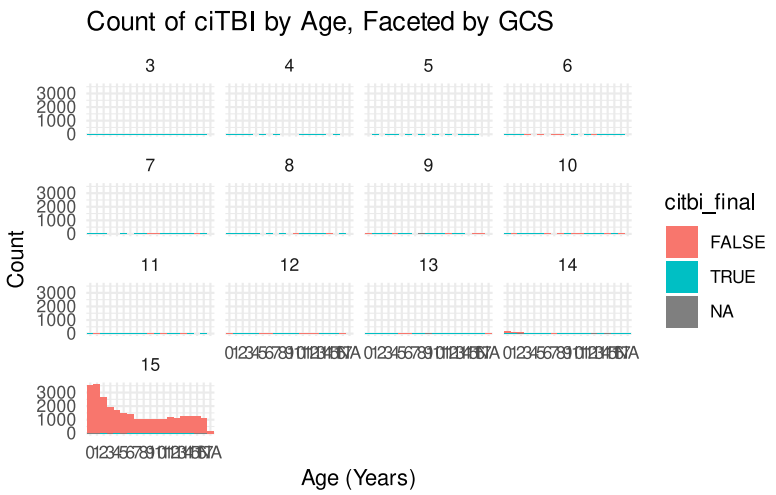
- b. Explore whether the relationship between age and ciTBI differs across Glasgow Coma Scale scores. Create a stacked normalized bar chart that shows the proportion of patients with ciTBI across age, now faceted by total GCS score.

```
citbi_gcs_age |>
    ggplot(aes(x = factor(age_in_year), fill = citbi_final)) +
    geom_bar(position = "fill") +
    facet_wrap(~ gcs_total) +
    labs(title = "Proportion of ciTBI by Age, Faceted by GCS", x = "Age
(Years)", y = "Proportion") +
    theme_minimal()
```



- c. Create a stacked bar chart that shows the number of patients with ciTBI across age faceted by total GCS score.

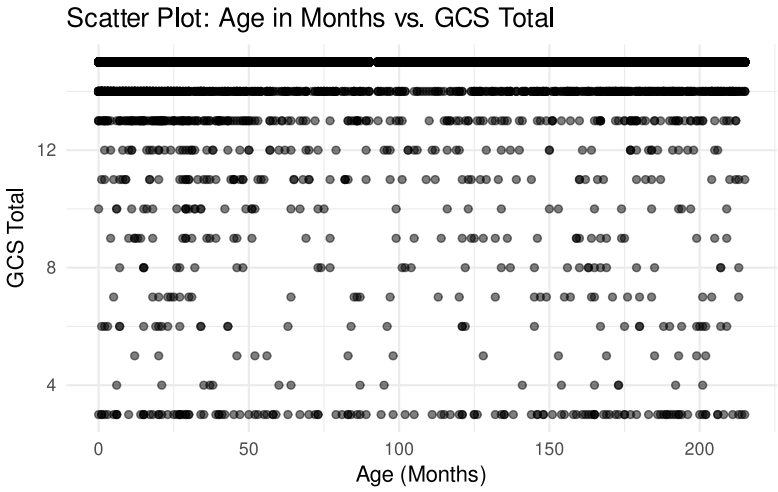
```
citbi_gcs_age |>
    ggplot(aes(x = factor(age_in_year), fill = citbi_final)) +
    geom_bar(position = "stack") +
    facet_wrap(~ gcs_total) +
    labs(title = "Count of ciTBI by Age, Faceted by GCS", x = "Age (Years)", y
= "Count") +
    theme_minimal()
```



9. Create a scatter plot to visualize any two numeric variables in the dataset. Describe any interesting patterns you see.

```
citbi_update |>
    ggplot(aes(x = age_in_month, y = gcs_total)) +
    geom_point(alpha = 0.5) +
    labs(title = "Scatter Plot: Age in Months vs. GCS Total", x = "Age
(Months)", y = "GCS Total") +
    theme_minimal()
```

```
Warning: Removed 189 rows containing missing values or values outside the
scale range
(`geom_point()`).
```



10. Create a table that summarizes at least two statistics grouped by two categorical variables. Describe any interesting patterns you see.

```
citbi_grouped <- citbi_update |>
    filter(!is.na(duration_loss_of_consciousness) & !is.na(citbi_final)) |>
    group_by(duration_loss_of_consciousness, citbi_final) |>
    summarise(count = n())
```

```
`summarise()` has grouped output by 'duration_loss_of_consciousness'. You can
override using the `.groups` argument.
```

```
print(citbi_grouped)
```

```
# A tibble: 8 × 3
# Groups:   duration_loss_of_consciousness [4]
  duration_loss_of_consciousness citbi_final count
  <ord>                          <lgl>       <int>
1 < 5 sec                        FALSE         652
2 < 5 sec                        TRUE            8
3 5 sec - < 1 min                FALSE        1283
4 5 sec - < 1 min                TRUE           25
5 1 - 5 min                      FALSE         723
6 1 - 5 min                      TRUE           30
7 > 5 min                        FALSE         168
8 > 5 min                        TRUE           90
```