Data Collapsing

2023-10-06

```
#replace with your filename
filename <- "defrtdata_original_RH"
sav <- read_sav(paste0("./", filename, ".sav"))
#getwd() # this is the folder it will save into unless you specify otherwise in the path
below
write_csv(x = sav, file = paste0("./", filename, ".csv"))
data <- read_csv(paste0("./", filename, ".csv"), show_col_types = F)

merged_data <- data %>%
  group_by(record_id) %>%
  summarize_all(list(~ first(.[!is.na(.)])))
```

```
#Used to find the variables in this data set which contain the word date.
date_variables <- merged_data %>%
  select(contains("date"))
#Used to find the indices in this data set for the variables which contain the word dat
e.
date_indices <- grep("date", names(merged_data))
#View date variables and their indices
#date_variables
#date_indices
# if (!is.na(merged_data[,date_indices[1]])) {
#   for (i in date_indices[1]:(date_indices[2] - 1)) {
#     for j in (1:nrow(merged_data)) {
#       if is.na(merged_data[j,i]) {
#         merged_data[j,i] <- -777
#       }
#     }
#   }
# } else if {
#   for i in (date_indices[1]:(date_indices[2] - 1)) {
#     for j in (1:nrow(merged_data)) {
#       if is.na(merged_data[j,i]) {
#         merged_data[j,i] <- -888
# }}}}

#for session_date_virtual
for (i in seq(date_indices[1] + 1, date_indices[3] - 1)) {
  # Check if the column is numeric or character
  is_numeric <- is.numeric(merged_data[[i]])
  is_character <- is.character(merged_data[[i]])

  # Flag to check if a date is encountered for each row
  date_encountered <- rep(FALSE, nrow(merged_data))

  # Iterate through rows
  for (j in 1:nrow(merged_data)) {
    # If a date is encountered in the first date column
    if (!is.na(merged_data[j, date_indices[1]])) {
      date_encountered[j] <- TRUE
    }

    # If a date has been encountered, update NA values after the first date column
    if (date_encountered[j]) {
      if (is.na(merged_data[j, i])) {
        # Fill in -777 if date has been encountered, -888 otherwise
        if (is_numeric) {
          merged_data[j, i] <- -777
        } else if (is_character) {
          merged_data[j, i] <- "-777"
        }
      }
    } else if (is.na(merged_data[j, i])) {
      # Fill in -888 if date has not been encountered
```

```
      if (is_numeric) {
        merged_data[j, i] <- -888
      } else if (is_character) {
        merged_data[j, i] <- "-888"
      }
    }
  }
}

#Do the same for eeg_date
for (i in seq(date_indices[3] + 1, date_indices[4] - 1)) {
  # Check if the column is numeric or character
  is_numeric <- is.numeric(merged_data[[i]])
  is_character <- is.character(merged_data[[i]])

  # Flag to check if a date is encountered for each row
  date_encountered <- rep(FALSE, nrow(merged_data))

  # Iterate through rows
  for (j in 1:nrow(merged_data)) {
    # If a date is encountered in the first date column
    if (!is.na(merged_data[j, date_indices[1]])) {
      date_encountered[j] <- TRUE
    }

    # If a date has been encountered, update NA values after the first date column
    if (date_encountered[j]) {
      if (is.na(merged_data[j, i])) {
        # Fill in -777 if date has been encountered, -888 otherwise
        if (is_numeric) {
          merged_data[j, i] <- -777
        } else if (is_character) {
          merged_data[j, i] <- "-777"
        }
      }
    } else if (is.na(merged_data[j, i])) {
      # Fill in -888 if date has not been encountered
      if (is_numeric) {
        merged_data[j, i] <- -888
      } else if (is_character) {
        merged_data[j, i] <- "-888"
      }
    }
  }
}

#Do the same for smk_date
for (i in seq(date_indices[4] + 1, date_indices[5] - 1)) {
  # Check if the column is numeric or character
  is_numeric <- is.numeric(merged_data[[i]])
  is_character <- is.character(merged_data[[i]])
```

```
  # Flag to check if a date is encountered for each row
  date_encountered <- rep(FALSE, nrow(merged_data))

  # Iterate through rows
  for (j in 1:nrow(merged_data)) {
    # If a date is encountered in the first date column
    if (!is.na(merged_data[j, date_indices[1]])) {
      date_encountered[j] <- TRUE
    }

    # If a date has been encountered, update NA values after the first date column
    if (date_encountered[j]) {
      if (is.na(merged_data[j, i])) {
        # Fill in -777 if date has been encountered, -888 otherwise
        if (is_numeric) {
          merged_data[j, i] <- -777
        } else if (is_character) {
          merged_data[j, i] <- "-777"
        }
      }
    } else if (is.na(merged_data[j, i])) {
      # Fill in -888 if date has not been encountered
      if (is_numeric) {
        merged_data[j, i] <- -888
      } else if (is_character) {
        merged_data[j, i] <- "-888"
      }
    }
  }
}

#Do the same for session_date
for (i in seq(date_indices[5] + 1, ncol(merged_data))) {
  # Check if the column is numeric or character
  is_numeric <- is.numeric(merged_data[[i]])
  is_character <- is.character(merged_data[[i]])

  # Flag to check if a date is encountered for each row
  date_encountered <- rep(FALSE, nrow(merged_data))

  # Iterate through rows
  for (j in 1:nrow(merged_data)) {
    # If a date is encountered in the first date column
    if (!is.na(merged_data[j, date_indices[1]])) {
      date_encountered[j] <- TRUE
    }

    # If a date has been encountered, update NA values after the first date column
    if (date_encountered[j]) {
      if (is.na(merged_data[j, i])) {
        # Fill in -777 if date has been encountered, -888 otherwise
        if (is_numeric) {
```

```
          merged_data[j, i] <- -777
        } else if (is_character) {
          merged_data[j, i] <- "-777"
        }
      }
    } else if (is.na(merged_data[j, i])) {
      # Fill in -888 if date has not been encountered
      if (is_numeric) {
        merged_data[j, i] <- -888
      } else if (is_character) {
        merged_data[j, i] <- "-888"
      }
    }
  }
}

# View the updated merged_data
#View(merged_data)
write_sav(data = merged_data, path = paste0("./", filename, "_errorcode.sav"))
```