Processing and Encoding PRISM Data

Set up

Load the required libraries:

```
library(data.table)
library(rcahelpr) # remotes::install_github("cllghn/rcahelpr")
```

```
## {rcahelpr} loaded! Happy nerding.
```

Read Data

The data comes to us in two tables. The first, being the PRIMS files. The second, the supervision snapshot. Each is stored in a separate folder on SharePoint.

Both directories include data on a monthly basis. As such, we will read in bulk files for the available months in 2023 and then left join both tables using the appropriate key.

To begin bulk reading, first list the paths to the appropriate directory:

```
prism <- csgjcr::csg_sp_path("JR_Monitoring/Wyoming/Data/DOC/Excel Raw Files/PRISM")
pp <- csgjcr::csg_sp_path(
  "JR_Monitoring/Wyoming/Data/DOC/Excel Raw Files/Supervision Snapshot")
```

To ingest the PRISM data for 2023 (`prism_2023`), begin by bulk list monthly files for 2023. Then proceed read over the list of file paths and row bind the output. Keep in mind that in order to account for when the data was produced, a new variable is added (`reportedfor`):

```
# List all files for 2023.
prism_2023_paths <- list.files(path = prism, pattern = "\\s(2023)\\s",
                               full.names = TRUE)
# Read them all, add a month identifier, bind them.
prism_2023 <- lapply(prism_2023_paths, function(x){
  out <- readxl::read_xlsx(x)
  out['reportedfor'] <- gsub("\\s(2023).*", "_2023",
                             strsplit(x, '/')[[1]][length(strsplit(x, '/')[[1]])])
  return(out)
  }) |> rbindlist()
```

To ingest the supervision data (`pp_2023`), we repeat a version of the process outline above:

```
pp_2023_paths <- list.files(path = pp, pattern = "\\s(2023)\\s",
                               full.names = TRUE)

# Read them all, add a month identifier, bind them.
pp_2023 <- lapply(pp_2023_paths, function(x){
  out <- readxl::read_xlsx(x,
                           col_types = c("numeric", "numeric", "text", "text",
                                         "text", "text", "text", "text", "text",
                                         "text","text","text","text"))
  out['reportedfor'] <- paste0(
    strsplit(
```

```
      strsplit(x, '/')[[1]][length(strsplit(x, '/')[[1]])],
      " ")[[1]][1],
    "_2023")
  # Remove missing date strings for NA
  if (inherits(out[['BIRTH_DATE']], "character")) {
    out[['BIRTH_DATE']] <-  gsub("NULL", NA_character_, out[['BIRTH_DATE']])
    }
  return(out)
  }) |> rbindlist()
```

Examine the Data

Before joining the data, we asked two key questions.

1. Are there any multiple `EVTID` per report month in the PP data?

```
any(pp_2023[, .N, by = .(EVTID, reportedfor)][, N] |> unlist() > 1)
```

```
## [1] FALSE
```

2. Are there any multiple `EVTID` per report month in the PRISM data?

```
any(prism_2023[, .N, by = .(EVTID, reportedfor)][, N] |> unlist() > 1)
```

```
## [1] TRUE
```

There are multiple `EVTID` in a given month b/c a person can have multiple interactions (e.g., sanctions and rewards). Thus, the `reportedfor` variable is used to ensure that we link the PRISM data with the appropriate PP monthly snapshot.

Final Touches

Let's left join the `pp_2023` table to the `prism_2023` table. Next, transform the `EVENTDATE` variable to a date. Then, encode PII using a bespoke tool with the **{rcahelpr}** package.

```
prism_pp <- merge(x = prism_2023, y = pp_2023, by = c("EVTID", "reportedfor"),
                  all.x = TRUE)[, EVENTDATE := as.Date(EVENTDATE)] |>
  rcahelpr::encode_variables(c("EVTID", "CHSID"), tag = TRUE) |>
  rcahelpr::encode_variables("BIRTH_DATE", tag = FALSE)
```

Save the Data

List the destination:

```
output_path <- csgjcr::csg_sp_path("JR_WDOC/data/raw/%s")
```

Write as CSV:

```
fwrite(prism_pp, file = sprintf(output_path, "prism_pp_2023.csv"))
```