```
#Case Study 03: Data Cleaning and Transformation for Reproducible Research"
#'
#' 1.  **Identifying common data errors**, such as missing values, incorrect data types, and
inconsistent entries.
#' 2.  **Applying `tidyverse` functions** to correct these errors efficiently and reproducibly.
#' 3.  **Saving the cleaned dataset** to disk.

library(tidyverse)
library(readr)
library(knitr)

messy_data <- tibble(
      ID = 1:7,
      ObservationDate = c("2023-01-15", "Jan 20, 2023", "01/25/2025", "2023-02-05", "missing",
"2023-02-15", "03/01/2023"),
      Temperature_C = c("15.2", "18.0", "14.5", "invalid", "17.3", "20.1", NA),
      Humidity_Percent = c(70, -72, NA, 65, 780, 80, 71),
      Site = c("North", "nirth", "South", "East", "west", "NORTH", "Souther"))
head(messy_data)
#This function shows all of the data, since there are only 7 observations and 5 variables. In a
larger data set, this function would only show the beginning lines. Negative values are
highlighted.
str(messy_data)
#This function tells me that there are 7 rows and 5 columns of data, and shows that there are 7
observations within each column.
summary(messy_data)
#This function shows me a statistical summary of the messy_data, such as values like minimum and
mean.It can't provide values for columns with character values.

messy_data <- messy_data %>%
  mutate(Temperature_C = as.numeric(Temperature_C)) %>%
  mutate(ObservationDate = c(date1, date2, date3, date4, date5, date6, date7)) %>%
  mutate(Site = recode(messy_data$Site, North = "North", nirth = "North", South = "South", East =
"East", west = "West", NORTH = "North", Souther = "South")) %>%
  mutate(Humidity_Percent = if_else(Humidity_Percent >= 0 & Humidity_Percent <= 100,
Humidity_Percent, NA_real_))

#temperature: for temperature, using "as.numeric" converted this column to a numeric format
#observationdate: I saved each date as an object, parsed them with "mdy()" or "ymd()" based on
their format, which then allowed me to combine them into a new column formatted as "ymd()"
#site: I used the "recode()" function and specified the data frame/column, which allowed me to
choose values in the column and rename them
#humidity_percent: I used an "ifelse()" statement to choose humidity percent values between 0 and
100. Using the humidity_percent after specifies that I want the humidity_percent value to show if
the condition is true, and using NA_Real_ has "NA" show up if the condition is false
#my methods for data and site wouldn't be the most efficient for larger data sets.

date1 <- ymd("2023-01-15")
date2 <- mdy("Jan 20, 2023")
date3 <- mdy("01/25/2025")
date4 <- ymd("2023-02-05")
date5 <- "NA"
date6 <- ymd("2023-02-15")
date7 <- mdy("03-01-2023")

summary(messy_data)
str(messy_data)
view(messy_data)

dir.create(path = "data_processed", showWarnings = TRUE, recursive = TRUE, mode = "0777")
write.csv(messy_data, "data_processed/cleaned_field_data.csv", row.names = FALSE)
```