Transforming Data

Candace Grant

2025-10-05

```
library(tidyverse)
```

The dataset for this project can be classified as untidy because the columns have multiple variables and the rows have multiple observations. In this project I will tidy the dataset by transforming the data.

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   4.0.0     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.1.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(readr)
```

Read the csv file and view overall structure

```
#Read the csv file and view overall structure
employment_wide <- read_csv("employment_stats.csv", skip = 1)
```

```
## New names:
## Rows: 34 Columns: 5
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (1): ...1 num (4): Aug. 2024...2, Aug. 2025...3, Aug. 2024...4, Aug. 2025...5
## 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.
## * `` -> `...1`
## * `Aug. 2024` -> `Aug. 2024...2`
## * `Aug. 2025` -> `Aug. 2025...3`
## * `Aug. 2024` -> `Aug. 2024...4`
## * `Aug. 2025` -> `Aug. 2025...5`
```

```r
head(employment_wide, 10)
```

```
## # A tibble: 10 x 5
##    ...1          `Aug. 2024...2` `Aug. 2025...3` `Aug. 2024...4` `Aug. 2025...5`
##    <chr>                   <dbl>           <dbl>           <dbl>           <dbl>
##  1 <NA>                     NA              NA              NA              NA
##  2 Civilian non~         33649           35129          235207          238872
##  3 Civilian lab~          8030            8809          160733          162226
##  4 Participatio~            23.9            25.1            68.3            67.9
##  5 Employed               7362            8052          153987          155236
##  6 Employment-p~            21.9            22.9            65.5            65
##  7 Unemployed              669             757            6746            6990
##  8 Unemployment~             8.3             8.6             4.2             4.3
##  9 Not in labor~         25619           26321           74474           76646
## 10 Men, 16 to 6~            NA              NA              NA              NA
```

```r
glimpse(employment_wide)
```

```
## Rows: 34
## Columns: 5
## $ ...1            <chr> NA, "Civilian noninstitutional population", "Civilian ~
## $ `Aug. 2024...2` <dbl> NA, 33649.0, 8030.0, 23.9, 7362.0, 21.9, 669.0, 8.3, 2~
## $ `Aug. 2025...3` <dbl> NA, 35129.0, 8809.0, 25.1, 8052.0, 22.9, 757.0, 8.6, 2~
## $ `Aug. 2024...4` <dbl> NA, 235207.0, 160733.0, 68.3, 153987.0, 65.5, 6746.0, ~
## $ `Aug. 2025...5` <dbl> NA, 238872.0, 162226.0, 67.9, 155236.0, 65.0, 6990.0, ~
```

```r
str(employment_wide)
```

```
## spc_tbl_ [34 x 5] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ...1         : chr [1:34] NA "Civilian noninstitutional population" "Civilian labor force" "Parti
##  $ Aug. 2024...2: num [1:34] NA 33649 8030 23.9 7362 ...
##  $ Aug. 2025...3: num [1:34] NA 35129 8809 25.1 8052 ...
##  $ Aug. 2024...4: num [1:34] NA 235207 160733 68.3 153987 ...
##  $ Aug. 2025...5: num [1:34] NA 238872 162226 67.9 155236 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ...1 = col_character(),
##   ..   `Aug. 2024...2` = col_number(),
##   ..   `Aug. 2025...3` = col_number(),
##   ..   `Aug. 2024...4` = col_number(),
##   ..   `Aug. 2025...5` = col_number()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```r
names(employment_wide)
```

```
## [1] "...1"          "Aug. 2024...2" "Aug. 2025...3" "Aug. 2024...4"
## [5] "Aug. 2025...5"
```

```r
tail(employment_wide, 5)
```

```
## # A tibble: 5 x 5
##   ...1           `Aug. 2024...2` `Aug. 2025...3` `Aug. 2024...4` `Aug. 2025...5`
##   <chr>                    <dbl>           <dbl>           <dbl>           <dbl>
## 1 Employment-po~             7.6             7.8            23.7            22.9
## 2 Unemployed                60              113             330             386
## 3 Unemployment ~             4.5             7.4             3.1             3.7
## 4 Not in labor ~         15607           16343           32479           33622
## 5 NOTE: A perso~            NA              NA              NA              NA
```

```r
unique_categories <- unique(employment_wide$Category)
```

```
## Warning: Unknown or uninitialised column: `Category`.
```

```r
print(unique_categories)
```

```
## NULL
```

Inspect the dataset

```r
data_types <- sapply(employment_wide, class)
print(data_types)
```

```
##          ...1 Aug. 2024...2 Aug. 2025...3 Aug. 2024...4 Aug. 2025...5
##   "character"     "numeric"     "numeric"     "numeric"     "numeric"
```

```r
numeric_cols <- names(employment_wide)[sapply(employment_wide, is.numeric)]
print(numeric_cols)
```

```
## [1] "Aug. 2024...2" "Aug. 2025...3" "Aug. 2024...4" "Aug. 2025...5"
```

```r
character_cols <- names(employment_wide)[sapply(employment_wide, is.character)]
print(character_cols)
```

```
## [1] "...1"
```

```r
cat("Duplicate rows:", sum(duplicated(employment_wide)))
```

```
## Duplicate rows: 0
```

```r
empty_rows <- employment_wide %>%
  filter(if_all(everything(), is.na))
cat("Completely empty rows:", nrow(empty_rows), "\n")
```

```
## Completely empty rows: 1
```

#####Transformation Goal The goal of this transformation is to convert the dataset from wide format to long (tidy) format. In the original wide format, column headers contain data values rather than variable names, violating tidy data principles. The transformed dataset will have five columns: Category, Disability_Status, Year, Month, and Value.

Transformation Approach My approach follows these steps:

- Clean the wide format data - Rename columns for clarity and remove empty rows
- Convert to long format - Use pivot_longer() to reshape the data
- Separate combined variables - Split compound column names into distinct variables (Disability_Status, Year, Month)
- Refine data types - Convert values to appropriate numeric formats and clean text fields
- Validate the structure - Ensure the final dataset adheres to tidy data principles

```
# Rename columns in the wide datasetto be more descriptive
employment_wide <- employment_wide %>%
  rename(
    Category = 1,                              # First column
    Disability_Aug2024 = 2,                    # People with disability, Aug 2024
    Disability_Aug2025 = 3,                    # People with disability, Aug 2025
    NoDisability_Aug2024 = 4,                  # People with no disability, Aug 2024
    NoDisability_Aug2025 = 5                   # People with no disability, Aug 2025
  ) %>%
  filter(!is.na(Category))                     # Remove empty rows

# View cleaned wide format
print("\nCleaned wide format:")
```

```
## [1] "\nCleaned wide format:"
```

```
print(head(employment_wide, 10))
```

```
## # A tibble: 10 x 5
##    Category           Disability_Aug2024 Disability_Aug2025 NoDisability_Aug2024
##    <chr>                           <dbl>              <dbl>                <dbl>
##  1 Civilian noninsti~              33649              35129               235207
##  2 Civilian labor fo~               8030               8809               160733
##  3 Participation rate                 23.9               25.1                 68.3
##  4 Employed                         7362               8052               153987
##  5 Employment-popula~                 21.9               22.9                 65.5
##  6 Unemployed                        669                757                 6746
##  7 Unemployment rate                   8.3                8.6                  4.2
##  8 Not in labor force              25619              26321                74474
##  9 Men, 16 to 64 yea~                 NA                 NA                   NA
## 10 Civilian labor fo~              3377               3837                79333
## # i 1 more variable: NoDisability_Aug2025 <dbl>
```

Check for empty rows and remove

```
empty_rows <- employment_wide %>%
  filter(if_all(everything(), is.na))
cat("Completely empty rows:", nrow(empty_rows), "\n")
```

```
## Completely empty rows: 0
```

#####Convert from wide to long format

```
library(tidyverse)

print(head(employment_wide, 15))
```

```
## # A tibble: 15 x 5
##    Category           Disability_Aug2024 Disability_Aug2025 NoDisability_Aug2024
##    <chr>                           <dbl>              <dbl>                <dbl>
##  1 Civilian noninsti~              33649              35129               235207
##  2 Civilian labor fo~               8030               8809               160733
##  3 Participation rate                 23.9               25.1                 68.3
##  4 Employed                         7362               8052               153987
##  5 Employment-popula~                 21.9               22.9                 65.5
##  6 Unemployed                        669                757                 6746
##  7 Unemployment rate                   8.3                8.6                  4.2
##  8 Not in labor force               25619              26321                74474
##  9 Men, 16 to 64 yea~                 NA                 NA                   NA
## 10 Civilian labor fo~               3377               3837                79333
## 11 Participation rate                 41.4               44.3                 83
## 12 Employed                         3065               3496                76097
## 13 Employment-popula~                 37.6               40.4                 79.6
## 14 Unemployed                        311                341                 3237
## 15 Unemployment rate                   9.2                8.9                  4.1
## # i 1 more variable: NoDisability_Aug2025 <dbl>
```

```
# Step 1: Pivot to long
employment_long <- employment_wide %>%
  pivot_longer(
    cols = -Category,
    names_to = "Group_Year",
    values_to = "Value"
  )

print(head(employment_long, 20))
```

```
## # A tibble: 20 x 3
##    Category                              Group_Year              Value
##    <chr>                                 <chr>                   <dbl>
##  1 Civilian noninstitutional population Disability_Aug2024    33649
##  2 Civilian noninstitutional population Disability_Aug2025    35129
##  3 Civilian noninstitutional population NoDisability_Aug2024 235207
##  4 Civilian noninstitutional population NoDisability_Aug2025 238872
##  5 Civilian labor force                 Disability_Aug2024     8030
##  6 Civilian labor force                 Disability_Aug2025     8809
##  7 Civilian labor force                 NoDisability_Aug2024 160733
##  8 Civilian labor force                 NoDisability_Aug2025 162226
##  9 Participation rate                   Disability_Aug2024       23.9
## 10 Participation rate                   Disability_Aug2025       25.1
## 11 Participation rate                   NoDisability_Aug2024     68.3
## 12 Participation rate                   NoDisability_Aug2025     67.9
```

```
## 13 Employed                              Disability_Aug2024     7362
## 14 Employed                              Disability_Aug2025     8052
## 15 Employed                              NoDisability_Aug2024 153987
## 16 Employed                              NoDisability_Aug2025 155236
## 17 Employment-population ratio           Disability_Aug2024       21.9
## 18 Employment-population ratio           Disability_Aug2025       22.9
## 19 Employment-population ratio           NoDisability_Aug2024     65.5
## 20 Employment-population ratio           NoDisability_Aug2025     65
```

Continue separating and cleaning the data

```
# Separate and clean
employment_tidy <- employment_long %>%
  separate(
    Group_Year,
    into = c("Disability_Status", "Month_Year"),
    sep = "_"
  ) %>%
  mutate(
    # Clean disability status
    Disability_Status = case_when(
      Disability_Status == "Disability" ~ "With Disability",
      Disability_Status == "NoDisability" ~ "No Disability",
      TRUE ~ Disability_Status
    ),
    # Extract year and month
    Year = str_extract(Month_Year, "\\d{4}"),
    Month = str_remove(str_extract(Month_Year, "[A-Za-z]+\\.?"), "\\."),
    # Convert value to numeric (handle commas)
    Value = case_when(
      is.character(Value) ~ as.numeric(gsub(",", "", Value)),
      TRUE ~ as.numeric(Value)
    )
  ) %>%
  select(Category, Disability_Status, Year, Month, Value)

print("\n=== FINAL TIDY FORMAT ===")
```

```
## [1] "\n=== FINAL TIDY FORMAT ==="
```

```
print(head(employment_tidy, 30))
```

```
## # A tibble: 30 x 5
##    Category                              Disability_Status Year  Month    Value
##    <chr>                                 <chr>             <chr> <chr>    <dbl>
##  1 Civilian noninstitutional population With Disability   2024  Aug    33649
##  2 Civilian noninstitutional population With Disability   2025  Aug    35129
##  3 Civilian noninstitutional population No Disability     2024  Aug   235207
##  4 Civilian noninstitutional population No Disability     2025  Aug   238872
##  5 Civilian labor force                With Disability   2024  Aug     8030
##  6 Civilian labor force                With Disability   2025  Aug     8809
##  7 Civilian labor force                No Disability     2024  Aug   160733
##  8 Civilian labor force                No Disability     2025  Aug   162226
```

```
##  9 Participation rate                  With Disability   2024  Aug       23.9
## 10 Participation rate                  With Disability   2025  Aug       25.1
## # i 20 more rows
```

Check transformation

```
# Show all unique categories
print(unique(employment_tidy$Category))
```

```
##  [1] "Civilian noninstitutional population"
##  [2] "Civilian labor force"
##  [3] "Participation rate"
##  [4] "Employed"
##  [5] "Employment-population ratio"
##  [6] "Unemployed"
##  [7] "Unemployment rate"
##  [8] "Not in labor force"
##  [9] "Men, 16 to 64 years"
## [10] "Women, 16 to 64 years"
## [11] "Both sexes, 65 years and over"
## [12] "NOTE: A person with a disability has at least one of the following conditions: is deaf or has s
```

```
# Overall employment comparison - people with and without disabilities
overall_employment <- employment_tidy %>%
  filter(Category == "Employed") %>%
  pivot_wider(names_from = Disability_Status, values_from = Value)
```

Demonstrating Tidy Data Analysis

```
## Warning: Values from `Value` are not uniquely identified; output will contain list-cols.
## * Use `values_fn = list` to suppress this warning.
## * Use `values_fn = {summary_fun}` to summarise duplicates.
## * Use the following dplyr code to identify duplicates.
##   {data} |>
##   dplyr::summarise(n = dplyr::n(), .by = c(Category, Year, Month,
##   Disability_Status)) |>
##   dplyr::filter(n > 1L)
```

```
print(overall_employment)
```

```
## # A tibble: 2 x 5
##   Category Year  Month `With Disability` `No Disability`
##   <chr>    <chr> <chr> <list>            <list>
## 1 Employed 2024  Aug   <dbl [4]>         <dbl [4]>
## 2 Employed 2025  Aug   <dbl [4]>         <dbl [4]>
```

```
# Participation rates - percentage of the working-age population that is either employed or actively lo
participation <- employment_tidy %>%
  filter(Category == "Participation rate") %>%
  pivot_wider(names_from = c(Disability_Status, Year), values_from = Value)
```

```
## Warning: Values from `Value` are not uniquely identified; output will contain list-cols.
## * Use `values_fn = list` to suppress this warning.
## * Use `values_fn = {summary_fun}` to summarise duplicates.
## * Use the following dplyr code to identify duplicates.
##   {data} |>
##   dplyr::summarise(n = dplyr::n(), .by = c(Category, Month, Disability_Status,
##   Year)) |>
##   dplyr::filter(n > 1L)
```

```
print(participation)
```

```
## # A tibble: 1 x 6
##   Category           Month `With Disability_2024` `With Disability_2025`
##   <chr>              <chr> <list>                 <list>
## 1 Participation rate Aug   <dbl [4]>              <dbl [4]>
## # i 2 more variables: `No Disability_2024` <list>, `No Disability_2025` <list>
```

```
# Example 3: By demographic group
demographics <- employment_tidy %>%
  filter(Category %in% c("Men, 16 to 64 years", "Women, 16 to 64 years",
                         "Both sexes, 65 years and over"))
print(head(demographics, 12))
```

```
## # A tibble: 12 x 5
##    Category                     Disability_Status Year  Month Value
##    <chr>                        <chr>             <chr> <chr> <dbl>
##  1 Men, 16 to 64 years          With Disability   2024  Aug      NA
##  2 Men, 16 to 64 years          With Disability   2025  Aug      NA
##  3 Men, 16 to 64 years          No Disability     2024  Aug      NA
##  4 Men, 16 to 64 years          No Disability     2025  Aug      NA
##  5 Women, 16 to 64 years        With Disability   2024  Aug      NA
##  6 Women, 16 to 64 years        With Disability   2025  Aug      NA
##  7 Women, 16 to 64 years        No Disability     2024  Aug      NA
##  8 Women, 16 to 64 years        No Disability     2025  Aug      NA
##  9 Both sexes, 65 years and over With Disability   2024  Aug      NA
## 10 Both sexes, 65 years and over With Disability   2025  Aug      NA
## 11 Both sexes, 65 years and over No Disability     2024  Aug      NA
## 12 Both sexes, 65 years and over No Disability     2025  Aug      NA
```

Save tidy format in a csv

```
write_csv(employment_tidy, "employment_status_tidy.csv")
cat("\nTidy data saved to: employment_status_tidy.csv\n")
```

```
##
## Tidy data saved to: employment_status_tidy.csv
```