Summarizing Data Tutorial

Grace Arkfeld

2024-03-05

```
rm(list = ls())
```

```
# Load the tidyverse package
library(tidyverse)
```

```
## -- 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   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- 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
```

```
# Load the knitr package
library(knitr)
```

```
la_mort <-
  read_csv("https://www.dropbox.com/scl/fi/fzsnhfd3lq80v2o3sag6c/la_mort.csv?rlkey=h1vyjm2b8ppgejgsg3e8
```

```
## Rows: 642696 Columns: 29
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): stocr, strsd, stbrth, brthr, sex, marstat, ucod
## dbl (22): restatus, cntyocr, popcntyocr, cntyrsd, popcntyresd, educ1989, edu...
##
## 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.
```

```
la_pop <-
  read_csv("https://www.dropbox.com/scl/fi/650k1obpczky6bwa19ex6/la_county_pop.csv?rlkey=0aokd9m76q7mxw
```

```
## Rows: 24320 Columns: 23
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (3): stname, ctyname, agegrp
## dbl (20): state, county, year, tot_pop, tot_male, tot_female, wa_male, wa_fe...
##
## 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.
```

```r
stnrd_pop <-
  read_csv("https://www.dropbox.com/scl/fi/xzd2o5lza237so6vamqwb/stnrd_pop.csv?rlkey=zp90au2tuq6eptvi1y
```

```
## Rows: 18 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): agegrp
## dbl (1): stnrd_pop
##
## 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.
```

```r
# Step 4: Define Cancer Alley Parishes
la_mort$cancer_alley_parish <- ifelse(la_mort$cntyrsd %in% c(5, 47, 89, 93, 95, 121), 1, 0)
```

```r
# Step 5: Define Cancer Deaths by Cancer Site
la_mort$stomach_cancer <- ifelse(la_mort$ucr39 == 5, 1, 0)
la_mort$colon_cancer <- ifelse(la_mort$ucr39 == 6, 1, 0)
la_mort$pancreas_cancer <- ifelse(la_mort$ucr39 == 7, 1, 0)
la_mort$lung_cancer <- ifelse(la_mort$ucr39 == 8, 1, 0)
la_mort$breast_cancer <- ifelse(la_mort$ucr39 == 9, 1, 0)
la_mort$cervix_cancer <- ifelse(la_mort$ucr39 == 10, 1, 0)
la_mort$prostate_cancer <- ifelse(la_mort$ucr39 == 11, 1, 0)
la_mort$bladder_cancer <- ifelse(la_mort$ucr39 == 12, 1, 0)
la_mort$lymphoma_cancer <- ifelse(la_mort$ucr39 == 13, 1, 0)
la_mort$leukemia_cancer <- ifelse(la_mort$ucr39 == 14, 1, 0)
la_mort$other_site_cancer <- ifelse(la_mort$ucr39 == 15, 1, 0)
la_mort$total_cancer <- ifelse(la_mort$ucr39 %in% 5:15, 1, 0)
```

```r
# Step 6: Adjust Age Groupings
la_mort <- la_mort %>% filter(age != 9999)
age_breaks <- c(0, 1, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100, 1
age_labels <- c("0", "1-4", "5-9", "10-14", "15-19", "20-24", "25-29", "30-34", "35-39", "40-44", "45-4
la_mort$agegrp <- cut(la_mort$age, breaks = age_breaks, labels = age_labels, right = FALSE)
```

```r
# Step 7: Define Race in the Mortality File
la_mort <- la_mort %>% filter(racer5 %in% c(1, 2))
la_mort$black <- ifelse(la_mort$racer5 == 2, 1, 0)
```

```r
# Step 8: Create Parish Counts of Cancer Deaths by Cancer Site and by Race
parish_counts <- la_mort %>%
  group_by(cntyrsd, black, year, agegrp) %>%
  summarize(
    stomach_cancer_deaths = sum(stomach_cancer, na.rm = TRUE),
    colon_cancer_deaths = sum(colon_cancer, na.rm = TRUE),
    pancreas_cancer_deaths = sum(pancreas_cancer, na.rm = TRUE),
    lung_cancer_deaths = sum(lung_cancer, na.rm = TRUE),
    breast_cancer_deaths = sum(breast_cancer, na.rm = TRUE),
    cervix_cancer_deaths = sum(cervix_cancer, na.rm = TRUE),
    prostate_cancer_deaths = sum(prostate_cancer, na.rm = TRUE),
    bladder_cancer_deaths = sum(bladder_cancer, na.rm = TRUE),
    lymphoma_cancer_deaths = sum(lymphoma_cancer, na.rm = TRUE),
```

```
    leukemia_cancer_deaths = sum(leukemia_cancer, na.rm = TRUE),
    other_site_cancer_deaths = sum(other_site_cancer, na.rm = TRUE),
    total_cancer_deaths = sum(total_cancer, na.rm = TRUE)
  )
```

```
## `summarise()` has grouped output by 'cntyrsd', 'black', 'year'. You can
## override using the `.groups` argument.
```

```
# Step 9: Define Race in the Population File
la_pop <- la_pop %>%
  mutate(
    black_pop = rowSums(select(., c("ba_male", "ba_female"))),
    white_pop = rowSums(select(., c("wa_male", "wa_female"))),
  )
la_pop_black <- select(la_pop, county, year, agegrp, black_pop)
la_pop_white <- select(la_pop, county, year, agegrp, white_pop)
```

```
# Step 10: Join the Mortality and Population Data Frames
la_joined_black <- parish_counts %>%
  filter(black == 1) %>%
  inner_join(la_pop_black, by = c("cntyrsd" = "county", "year", "agegrp")) %>%
  rename(tot_pop = black_pop)

la_joined_white <- parish_counts %>%
  filter(black == 0) %>%
  inner_join(la_pop_white, by = c("cntyrsd" = "county", "year", "agegrp")) %>%
  rename(tot_pop = white_pop)

la_bind <- rbind(la_joined_black, la_joined_white)
```

```
# Step 11: Join the Mortality/Population Data to the Standard Population Data
# Assuming stnrd_pop is the standard population data frame
la_bind <- la_bind %>%
  inner_join(stnrd_pop, by = "agegrp")
```

```
# Step 12: Calculate Population Weights
la_bind$stnrd_pop_weight <- (la_bind$stnrd_pop) / (sum(stnrd_pop$stnrd_pop))
```

```
# Step 13: Calculate Cancer Mortality Rates by Cancer Site and Race
cancer_sites <- c("stomach", "colon", "pancreas", "lung", "breast", "cervix", "prostate", "bladder", "ly
for (site in cancer_sites) {
  rate_col <- paste(site, "cancer_rate_adj", sep = "_")
  death_col <- paste(site, "cancer_deaths", sep = "_")
  la_bind[[rate_col]] <- ((la_bind[[death_col]]) / (la_bind$tot_pop / 100000)) * la_bind$stnrd_pop_weigh
}
```

```
# Replace "inf" values with NA
for (col in names(la_bind)) {
  la_bind[[col]][is.infinite(la_bind[[col]])] <- NA
}
```

```r
# Step 14: Aggregate to the Parish-Year Level
parish_rates <- la_bind %>%
  group_by(cntyrsd, black, year) %>%
  summarize(across(ends_with("cancer_rate_adj"), sum, na.rm = TRUE), tot_pop = sum(tot_pop))
```

```
## Warning: There was 1 warning in `summarize()`.
## i In argument: `across(ends_with("cancer_rate_adj"), sum, na.rm = TRUE)`.
## Caused by warning:
## ! The `...` argument of `across()` is deprecated as of dplyr 1.1.0.
## Supply arguments directly to `.fns` through an anonymous function instead.
##
##   # Previously
##   across(a:b, mean, na.rm = TRUE)
##
##   # Now
##   across(a:b, \(x) mean(x, na.rm = TRUE))
```

```
## `summarise()` has grouped output by 'cntyrsd', 'black'. You can override using
## the `.groups` argument.
```

```r
# Step 15: Weight by Parish Population
for (site in cancer_sites) {
  rate_col <- paste(site, "cancer_rate_adj", sep = "_")
  weight_col <- paste(site, "pop_weight", sep = "_")
  parish_rates[[weight_col]] <- parish_rates[[rate_col]] * parish_rates$tot_pop
}
```

```r
# Step 15: Weight by Parish Population
for (site in cancer_sites) {
  rate_col <- paste(site, "cancer_rate_adj", sep = "_")
  weight_col <- paste(site, "pop_weight", sep = "_")
  parish_rates[[weight_col]] <- parish_rates[[rate_col]] * parish_rates$tot_pop
}
```

```r
# Create a data frame with cancer_alley_parish information
cancer_alley_info <- la_mort %>%
  select(cntyrsd, cancer_alley_parish) %>%
  distinct()

# Add cancer_alley_parish information to parish_rates
parish_rates <- parish_rates %>%
  left_join(cancer_alley_info, by = "cntyrsd")
```

```r
# Step 16: Aggregate to Cancer Alley and non-Cancer Alley Parishes
cancer_alley_rates <- parish_rates %>%
  group_by(cancer_alley_parish, black, year) %>%
  summarize(across(ends_with("pop_weight"), sum), tot_pop = sum(tot_pop)) %>%
  mutate(across(ends_with("pop_weight"), ~ .x / tot_pop))
```

```
## `summarise()` has grouped output by 'cancer_alley_parish', 'black'. You can
## override using the `.groups` argument.
```