DSCI 401 HW 4

Isabel Heard

10/20/2023

Contents

1

A

Use the HELPrct data from the mosaicData to calculate the mean of all numeric variables (be sure to exclude missing values).

```r
library(mosaicData)
library(tidyverse)
```

```
## Warning: package 'tidyr' was built under R version 4.0.5

## Warning: package 'readr' was built under R version 4.0.5

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.0     v readr     2.1.2
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.1     v tibble    3.1.8
## v lubridate 1.8.0     v tidyr     1.2.0
## v purrr     1.0.1
## -- 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 error
```

```r
library(dplyr)
data(HELPrct)
#head(HELPrct)
#summary(HELPrct)

HELPrct %>%
  summarize(across(where(is.numeric), mean, na.rm = TRUE))
```

```
## Warning: There was 1 warning in `summarize()`.
## i In argument: `across(where(is.numeric), mean, 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))
```

```
##        age anysubstatus     cesd       d1 daysanysub dayslink drugrisk      e2b
## 1 35.65342    0.7723577 32.84768 3.059603   75.30738 255.6056 1.887168 2.504673
##      female       i1       i2       id   indtot linkstatus      mcs      pcs
## 1 0.2362031 17.90728 24.54746 233.4018 35.72848  0.3781903 31.67668 48.04854
##     pss_fr  sexrisk avg_drinks max_drinks hospitalizations
## 1 6.706402 4.642384   17.90728   24.54746         3.059603
```

B

Find the mean of all the numeric variables stratified by sex and age group where age groups are defined as ranges of 10 years (i.e. 0-10, 10-20, 20-30, etc).

```
#Age groups and labels
age_groups <- seq(0, max(HELPrct$age), by = 10)
age_labels <- paste(age_groups, age_groups + 10, sep = "-")
#Add a label for ages greater than the maximum age - helps with error
age_labels[length(age_labels)] <- paste(age_labels[length(age_labels)], "and above")
df <- HELPrct %>% mutate(age_group = cut(age, breaks = c(age_groups, Inf), labels = age_labels, include

#Mean for each numeric variable, stratified by sex and age group
result <- df %>%
  group_by(sex, age_group) %>%
  summarize(across(where(is.numeric), mean, na.rm = TRUE)) #same code from previous problem
```

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

```
print(result)
```

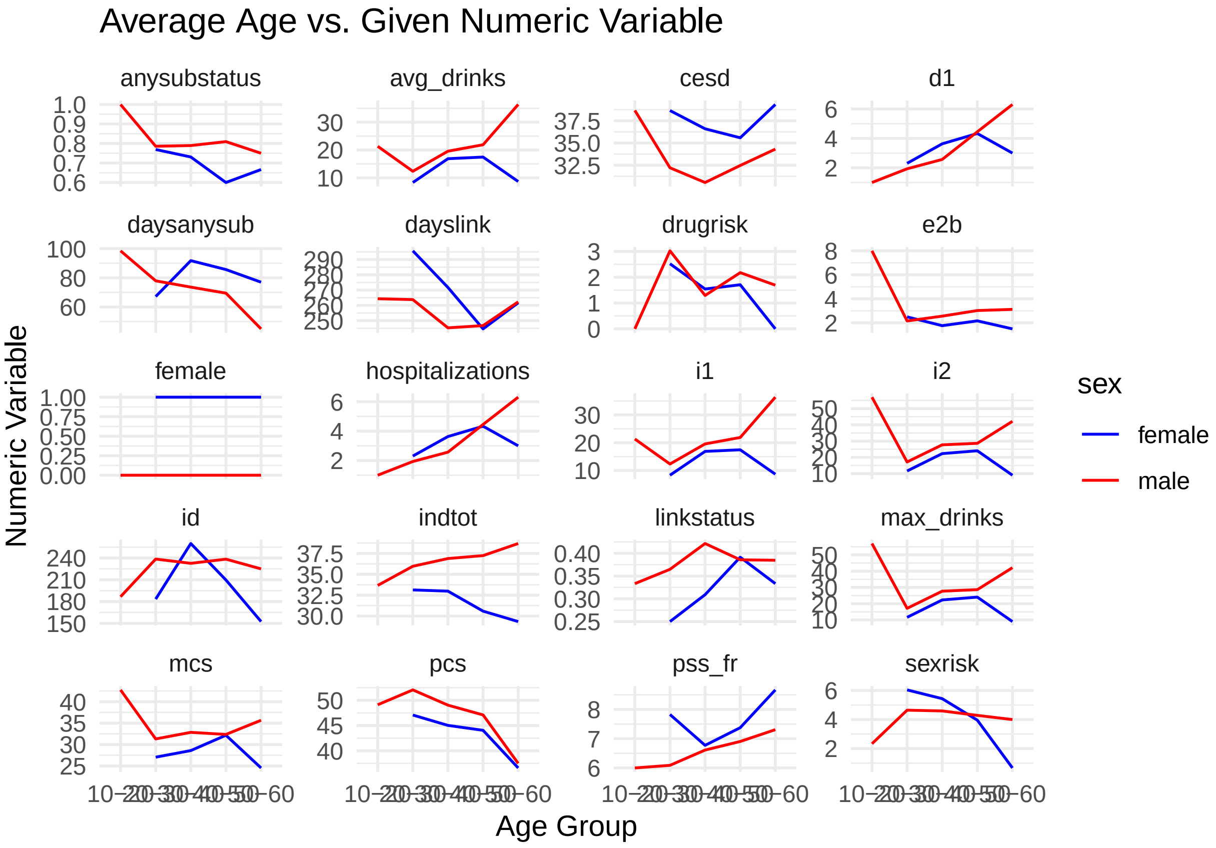
```
## # A tibble: 9 x 23
## # Groups:   sex [2]
##   sex    age_gr~1   age anysu~2  cesd    d1 daysa~3 daysl~4 drugr~5   e2b female
##   <fct>  <fct>    <dbl>   <dbl> <dbl> <dbl>   <dbl>   <dbl>   <dbl> <dbl>  <dbl>
## 1 female 20-30     27.1   0.769  38.7  2.30    67.2    296.    2.52  2.5       1
## 2 female 30-40     35.0   0.731  36.6  3.63    91.8    272.    1.54  1.76      1
## 3 female 40-50     45.5   0.6    35.6  4.33    85.7    245.    1.71  2.17      1
## 4 female 50-60     56.7   0.667  39.3  3       77      262.    0     1.5       1
## 5 male   10-20     19.7   1      38.7  1       98.5    264.    0     8         0
## 6 male   20-30     26.7   0.786  32.2  1.93    77.9    264.    3.02  2.16      0
## 7 male   30-40     35.1   0.789  30.6  2.57    73.6    245.    1.29  2.56      0
## 8 male   40-50     44.1   0.810  32.5  4.45    69.5    247.    2.18  3.03      0
## 9 male   50-60     55.4   0.75   34.3  6.31    45      262.    1.69  3.12      0
## # ... with 12 more variables: i1 <dbl>, i2 <dbl>, id <dbl>, indtot <dbl>,
## #   linkstatus <dbl>, mcs <dbl>, pcs <dbl>, pss_fr <dbl>, sexrisk <dbl>,
## #   avg_drinks <dbl>, max_drinks <dbl>, hospitalizations <dbl>, and abbreviated
## #   variable names 1: age_group, 2: anysubstatus, 3: daysanysub, 4: dayslink,
## #   5: drugrisk
```

C

Using the data set created in the previous problem, create a set of line plots with the average age of the age group on the x-axis and each of other numeric variables on the y-axis in separate plots stratified by sex. (Note: You are not allowed to use a for loop here or simply copy-and- paste 20 times!)

```
library(ggplot2)
df_long <- result %>%
  pivot_longer(cols = c("anysubstatus", "cesd", "d1", "daysanysub", "dayslink", "drugrisk", "e2b", "fema
                        "indtot", "linkstatus", "mcs", "pcs", "pss_fr", "sexrisk", "avg_drinks", "max_d

plots <- df_long %>%
  ggplot(aes(x = age_group, y = value, group = interaction(variable, sex), color = sex, linetype = sex)
  geom_line() +
  facet_wrap(vars(variable), scales = "free_y", ncol = 4, nrow = 5) +
  labs(x = "Age Group", y = "Numeric Variable", title = "Average Age vs. Given Numeric Variable") +
  scale_linetype_manual(values = c("solid", "solid")) +
  scale_color_manual(values = c("blue", "red")) +
  theme_minimal()
print(plots)
```



2

The team IDs corresponding to Brooklyn baseball teams from the Teams data frame from the Lahman package are listed below. Use map int() to find the number of seasons in which each of those teams played by calling a function called count seasons.

```
library(Lahman)
data(Teams)

#List of Brooklyn baseball teams
bk_teams <- c("BR1", "BR2", "BR3", "BR4", "BRO", "BRP", "BRF")

#Count seasons by teamID/yearID
count_seasons <- function(team_id) {
  Teams %>%
    filter(teamID == team_id) %>%
    distinct(yearID) %>%
    nrow()}

#Map to count seasons for each team
season_counts <- map_int(bk_teams, count_seasons)

#Make df
result_df <- data.frame(teamID = bk_teams, seasons_played = season_counts)
print(result_df)
```

```
##   teamID seasons_played
## 1    BR1              1
## 2    BR2              4
## 3    BR3              6
## 4    BR4              1
## 5    BRO             68
## 6    BRP              1
## 7    BRF              2
```

Colab Link

https://colab.research.google.com/drive/1Yx9utHdcMaggM7X69Yn5h0Jm9ljwOHVC?usp=sharing