September 30, 2025

The results below are generated from an R script.

```
################################################################################
##
## [ PROJ ] Supplemental Example: Women in STEM
## [ FILE ] womeninstem.r
## [ AUTH ] < YOUR NAME >
## [ INIT ] < September 30, 2025 >
##
################################################################################

## This exercise is intended to help with project brainstorming, exploratory
## data analysis (EDA), and to demonstrate why a cross-country analysis is
## generally *not* advised for studying the effects of public policy.

## Research question:
##  - Do countries with more mandated maternal leave have a greater share of
##    women graduating in STEM fields?
##  - A more informative causal framing of that question:
##     Does maternal leave increase women's representation in STEM fields?

## Data source: World Bank's DataBank Gender Statistics
##  (https://databank.worldbank.org/source/gender-statistics)
```

```
## load libraries
```

```
# install.packages("ggrepel")
# install.packages("ggpmisc")

library(tidyverse)
library(ggrepel)
library(ggpmisc)

getwd()

## [1] "C:/Users/hbs2103/My Drive/Teaching/U6614-drive/Lectures-drive/WomenInSTEM"
```

```
## 1. load & prep input data
```

```
wbgender <- read_csv("worldbank-genderstats.csv", na = "..")
```

```
## Warning:  One or more parsing issues, call `problems()` on your data frame for details,
e.g.:
##   dat <- vroom(...)
##   problems(dat)
## Rows:  242491 Columns:  14
## - Column specification --------------------------------
## Delimiter:  ","
## chr  (4):  Series Name, Series Code, Country Name, Country Code
## dbl (10):  2011 [YR2011], 2012 [YR2012], 2013 [YR2013], 2014 [YR2014], 2015 [YR2015], 2...
##
## 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.
```

```
# wait this data isn't "tidy"!
# each row is not its own observation! there are diff rows for diff vars

# here are the 3 variables we want to work with
  # SE.TER.GRAD.FE.SI.ZS :
    # Y: Female share of graduates from STEM programmes, tertiary (%)
  # SH.MMR.LEVE :
    # X1: Length of paid maternity leave (calendar days)
  # SP.UWT.TFRT :
    # X2: Unmet need for contraception (% of married women ages 15-49)


# first, let's keep *rows* with information for 3 variables:
  keepvars <- c("SE.TER.GRAD.FE.SI.ZS", "SH.MMR.LEVE", "SP.UWT.TFRT")
  stem <- wbgender %>% filter(`Series Code` %in% keepvars)
    # NOTE: use backticks to refer to `Series Code` bc of space in col name
    # NOTE: the %in% operator checks if two vectors contain overlapping values
```

```
## 2. prepare analysis data frame
```

```
## A. create 3 separate df's for each variable
##    - stem.fmshstemgrads, stem.dayspaidmatleave, stem.unmetcontr
##    - each df should be a subset of rows for each variable
##    - look at the distribution of each variable in each year - any concerns?

  stem.fmshstemgrads <- wbgender %>% filter(`Series Code` == "SE.TER.GRAD.FE.SI.ZS")
  stem.dayspaidmatleave <- wbgender %>% filter(`Series Code` == "SH.MMR.LEVE")
  stem.unmetcontr <- wbgender %>% filter(`Series Code` == "SP.UWT.TFRT")

  summary(stem.fmshstemgrads)
```

```
##  Series Name        Series Code        Country Name       Country Code
##  Length:263         Length:263         Length:263         Length:263
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  2011 [YR2011]    2012 [YR2012]    2013 [YR2013]    2014 [YR2014]    2015 [YR2015]
```

```
##  Min.   :14.89   Min.   : 0.00   Min.   :19.45   Min.   :18.58   Min.   : 0.00
##  1st Qu.:29.63   1st Qu.:28.70   1st Qu.:28.44   1st Qu.:28.24   1st Qu.:28.20
##  Median :34.09   Median :33.33   Median :31.86   Median :33.68   Median :33.41
##  Mean   :35.63   Mean   :32.86   Mean   :32.50   Mean   :33.22   Mean   :33.99
##  3rd Qu.:41.22   3rd Qu.:39.94   3rd Qu.:38.40   3rd Qu.:38.07   3rd Qu.:41.25
##  Max.   :75.00   Max.   :49.44   Max.   :42.08   Max.   :46.33   Max.   :75.00
##  NA's   :201     NA's   :198     NA's   :226     NA's   :212     NA's   :169
##  2016 [YR2016]   2017 [YR2017]   2018 [YR2018]   2019 [YR2019]   2020 [YR2020]
##  Min.   :16.04   Min.   :10.56   Min.   : 0.00   Min.   :31.63   Min.   : NA
##  1st Qu.:28.58   1st Qu.:28.66   1st Qu.:28.10   1st Qu.:33.39   1st Qu.: NA
##  Median :34.41   Median :34.25   Median :34.17   Median :35.16   Median : NA
##  Mean   :34.71   Mean   :34.27   Mean   :34.52   Mean   :35.16   Mean   :NaN
##  3rd Qu.:39.97   3rd Qu.:40.08   3rd Qu.:42.46   3rd Qu.:36.92   3rd Qu.: NA
##  Max.   :55.47   Max.   :66.67   Max.   :60.76   Max.   :38.68   Max.   : NA
##  NA's   :169     NA's   :180     NA's   :216     NA's   :261     NA's   :263
```

```
  summary(stem.dayspaidmatleave)
```

```
##  Series Name        Series Code        Country Name       Country Code
##  Length:263         Length:263         Length:263         Length:263
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  2011 [YR2011]   2012 [YR2012]   2013 [YR2013]   2014 [YR2014]   2015 [YR2015]
##  Min.   :  0.0   Min.   :  0.0   Min.   :  0.0   Min.   :  0.0   Min.   :  0.0
##  1st Qu.: 84.0   1st Qu.: 84.0   1st Qu.: 84.0   1st Qu.: 84.0   1st Qu.: 84.0
##  Median : 91.0   Median : 91.0   Median : 91.0   Median : 91.0   Median : 91.0
##  Mean   :101.5   Mean   :100.6   Mean   :101.9   Mean   :102.4   Mean   :101.4
##  3rd Qu.:112.0   3rd Qu.:112.0   3rd Qu.:112.0   3rd Qu.:112.0   3rd Qu.:112.0
##  Max.   :635.0   Max.   :635.0   Max.   :635.0   Max.   :635.0   Max.   :635.0
##  NA's   :74      NA's   :74      NA's   :74      NA's   :74      NA's   :74
##  2016 [YR2016]   2017 [YR2017]   2018 [YR2018]   2019 [YR2019]   2020 [YR2020]
##  Min.   :  0.0   Min.   :  0.0   Min.   :  0.0   Min.   :  0.0   Min.   :  0.0
##  1st Qu.: 84.0   1st Qu.: 84.0   1st Qu.: 84.0   1st Qu.: 84.0   1st Qu.: 84.0
##  Median : 98.0   Median : 98.0   Median : 98.0   Median : 98.0   Median : 98.0
##  Mean   :101.4   Mean   :102.1   Mean   :102.9   Mean   :103.5   Mean   :104.6
##  3rd Qu.:112.0   3rd Qu.:112.0   3rd Qu.:119.0   3rd Qu.:119.0   3rd Qu.:120.0
##  Max.   :635.0   Max.   :635.0   Max.   :635.0   Max.   :635.0   Max.   :635.0
##  NA's   :74      NA's   :74      NA's   :74      NA's   :74      NA's   :74
```

```
  summary(stem.unmetcontr)
```

```
##  Series Name        Series Code        Country Name       Country Code
##  Length:263         Length:263         Length:263         Length:263
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  2011 [YR2011]   2012 [YR2012]    2013 [YR2013]   2014 [YR2014]   2015 [YR2015]
##  Min.   : 4.30   Min.   : 4.900   Min.   : 5.90   Min.   : 5.90   Min.   : 6.5
```

```
##  1st Qu.:10.60   1st Qu.: 9.375   1st Qu.:13.90   1st Qu.:11.40   1st Qu.:12.3
##  Median :19.35   Median :14.200   Median :17.50   Median :17.90   Median :18.9
##  Mean   :18.93   Mean   :16.389   Mean   :19.28   Mean   :19.42   Mean   :20.0
##  3rd Qu.:26.32   3rd Qu.:23.100   3rd Qu.:25.00   3rd Qu.:27.43   3rd Qu.:27.4
##  Max.   :34.30   Max.   :35.300   Max.   :37.20   Max.   :40.20   Max.   :34.7
##  NA's   :231     NA's   :235      NA's   :242     NA's   :225     NA's   :240
##  2016 [YR2016]  2017 [YR2017]   2018 [YR2018]   2019 [YR2019]   2020 [YR2020]
##  Min.   : 6.0   Min.   : 6.50   Min.   : 6.30   Min.   : 8.00   Min.   :12.90
##  1st Qu.:13.9   1st Qu.:14.55   1st Qu.:15.05   1st Qu.:10.30   1st Qu.:21.20
##  Median :22.3   Median :21.99   Median :19.90   Median :21.10   Median :24.20
##  Mean   :21.0   Mean   :20.61   Mean   :19.86   Mean   :19.92   Mean   :23.54
##  3rd Qu.:28.0   3rd Qu.:23.98   3rd Qu.:23.50   3rd Qu.:24.80   3rd Qu.:26.00
##  Max.   :38.0   Max.   :38.00   Max.   :33.60   Max.   :37.60   Max.   :33.40
##  NA's   :232    NA's   :223     NA's   :228     NA's   :246     NA's   :258
```

```
    # uh oh, way too many missing values in any given year!
    # obs come from diff surveys in diff years. values of vars may change slowly.
    # for now let's try using the mean over all years to reduce NAs
      # i.e. get mean fem share of STEM grads for every year (2011-2020)


## B. for each variable, create 'analysis variable' = mean value across all years for each country
##  - 1. start by reshaping data from wide to long form in each df using pivot_longer
##    - new long form df should have 1 obs for every country-year combination
##    - i.e. reshape stem.fmshstemgrads into new df stem.fmshstemgrads_long, etc.
##  - 2. next use group_by + summarise to generate aggregated stats for each group
##    - new df should have 3 columns: `Country Name`, `Country Code`, fmshstemgrads
##    - repeat for each input variable to end up with 3 data frames
##    - round to 2 decimal points if you need to

  # here's a basic pivot_longer example:
    # https://statisticsglobe.com/pivot_longer-and-pivot_wider-functions-in-r

  stem.fmshstemgrads_cross <- stem.fmshstemgrads %>%
    pivot_longer(cols = `2011 [YR2011]`:`2020 [YR2020]`,
                 names_to = "Year",       #ARG FOR TIME VARIABLE IN THIS EXAMPLE
                 values_to = "value") %>% #ARG FOR NEW COL NAME W/VARIABLE VALUES
    group_by(`Country Name`,`Country Code`) %>%
    summarise(fmshstemgrads = round(mean(value, na.rm = TRUE),2) )

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

  stem.dayspaidmatleave_cross  <- stem.dayspaidmatleave  %>%
    pivot_longer(cols = `2011 [YR2011]`:`2020 [YR2020]`,
                 names_to = "Year",
                 values_to = "value") %>%
    group_by(`Country Name`,`Country Code`) %>%
    summarise(dayspaidmatleave = round(mean(value, na.rm = TRUE), 2) )

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

  stem.unmetcontr_cross <- stem.unmetcontr %>%
    pivot_longer(cols = `2011 [YR2011]`:`2020 [YR2020]`,
```

```
                names_to = "Year",
                values_to = "value") %>%
    group_by(`Country Name`,`Country Code`) %>%
    summarise(unmetcontr = round(mean(value, na.rm = TRUE), 2) )
```

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

```
## C. join 3 new df's together to get a single "tidy" dataframe
##    - include 3 analysis variables and `Country Name` and `Country Code`
##    - how many countries have non-missing values for all 3 vars?

  stem.cross <- stem.fmshstemgrads_cross %>%
    inner_join(stem.unmetcontr_cross) %>%
    inner_join(stem.dayspaidmatleave_cross) %>%
    na.omit()
```

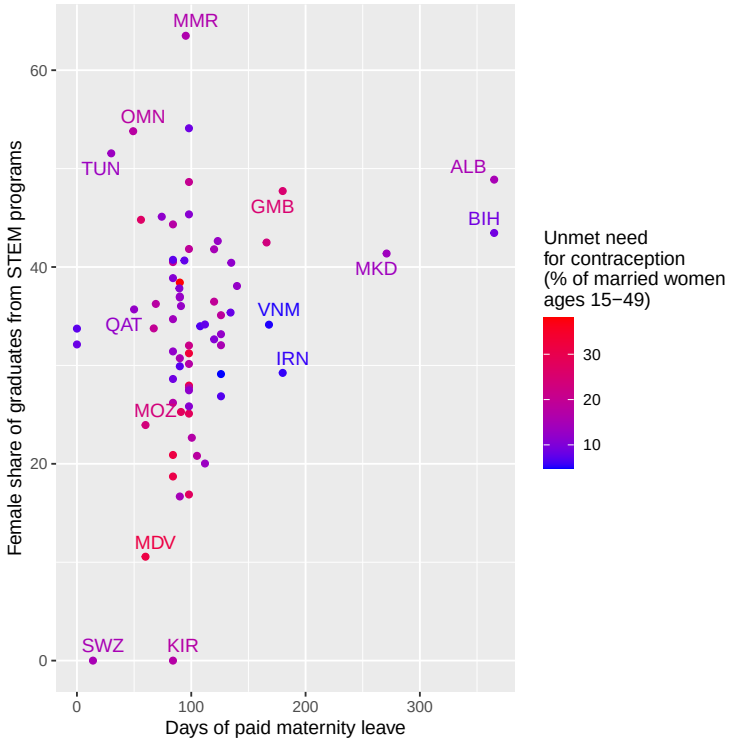
```
## Joining with `by = join_by(`Country Name`, `Country Code`)`
## Joining with `by = join_by(`Country Name`, `Country Code`)`
```

```
  # think about sample selection issues!
  # are missing observations for original variables randomly distributed?
```

```
## 3. exploratory analysis
```

```
## stem.cross has issues w/measurement error and non-random sample selection
## these issues limit the usefulness of this data for credible project work
## but we'll proceed a bit further for this in-class exercise


## A. explore relationship between days paid maternity leave & fem share of STEM grads
    ggplot(stem.cross,
           aes(x = dayspaidmatleave,
               y = fmshstemgrads,
               label = `Country Code`,
               color = unmetcontr)) +
      geom_point() +
      stat_dens2d_filter(geom = "text_repel", keep.fraction = 0.2) +
      scale_colour_gradient(low = "blue", high = "red") +
      ylab("Female share of graduates from STEM programs") + xlab("Days of paid maternity leave") +
      labs(color = "Unmet need \nfor contraception \n(% of married women \nages 15-49)") +
      theme(legend.position = "right")
```




```
  # how would you describe this relationship?
    # how would you describe the variation that we're using?

    cor(stem.cross$dayspaidmatleave, stem.cross$fmshstemgrads)
```

```
## [1] 0.244336
```

```
      # it's is positive (though doesn't appear to be linear)
      # we're using cross-sectional variation across countries


## B. could other country-level differences in part explain this relationship?
##    - what can you check in the data?

    cor(stem.cross$dayspaidmatleave, stem.cross$unmetcontr)
```

```
## [1] -0.1300048
```

```
    # surely there are other confounding factors! this is the usual problem w/cross-sectional variation
    # this cross-sectional variation in X (dayspaidmatleave) is endogenous!
    # i.e. it isn't random w/respect to other determinants of Y, such as unmet contraception need (X2)


## C. what can we do to improve internal validity?
    # i.e. we at least want to identify a more informative correlation, if not an arguably causal effect
    # trying to explicitly control for all of these other factors is usually an uphill battle
    # a stronger research design might focus on...
      # time variation "within-countries" (using panel data & fixed effects)
      # even better: exploit sharper policy changes giving variation in X
```

The R session information (including the OS info, R version and all packages used):

```r
sessionInfo()
```

```
## R version 4.4.2 (2024-10-31 ucrt)
## Platform: x86_64-w64-mingw32/x64
## Running under: Windows 11 x64 (build 22631)
##
## Matrix products: default
##
##
## locale:
## [1] LC_COLLATE=English_United States.utf8  LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8 LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## time zone: America/New_York
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] ggpmisc_0.6.1   ggpp_0.5.8-1    ggrepel_0.9.6   lubridate_1.9.4 forcats_1.0.0
##  [6] stringr_1.5.1   dplyr_1.1.4     purrr_1.0.2     readr_2.1.5     tidyr_1.3.1
## [11] tibble_3.2.1    ggplot2_3.5.2   tidyverse_2.0.0
##
## loaded via a namespace (and not attached):
##  [1] generics_0.1.3     stringi_1.8.4      lattice_0.22-6     hms_1.1.3
##  [5] magrittr_2.0.3     evaluate_1.0.3     grid_4.4.2         timechange_0.3.0
##  [9] Matrix_1.7-1       tinytex_0.57       survival_3.7-0     scales_1.3.0
## [13] cli_3.6.3          crayon_1.5.3       rlang_1.1.5        bit64_4.6.0-1
## [17] munsell_0.5.1      splines_4.4.2      withr_3.0.2        parallel_4.4.2
## [21] tools_4.4.2        SparseM_1.84-2     polynom_1.4-1      tzdb_0.4.0
## [25] MatrixModels_0.5-3 colorspace_2.1-1   vctrs_0.6.5        R6_2.5.1
## [29] lifecycle_1.0.4    bit_4.5.0.1        vroom_1.6.5        MASS_7.3-61
## [33] pkgconfig_2.0.3    pillar_1.10.1      gtable_0.3.6       glue_1.8.0
## [37] Rcpp_1.0.14        xfun_0.50          tidyselect_1.2.1   highr_0.11
## [41] rstudioapi_0.17.1  knitr_1.49         farver_2.1.2       labeling_0.4.3
## [45] compiler_4.4.2     quantreg_6.00
```

```r
Sys.time()
```

```
## [1] "2025-09-30 12:22:06 EDT"
```