Supply Demand Gap Analysis

2024-09-28

```r
# This code reads the LMI jobs demand data by region and occupation from the ODJFS website.

#Paths
common_path <- getwd()
target_folder <- paste0(common_path, "/data/lmi-data/")

# Create the target folder, this will be helpful so all groupmemebers will automatically have a folder
  dir.create(target_folder, recursive = TRUE)
```

```
## Warning in dir.create(target_folder, recursive = TRUE):
## ’C:\Users\marko\7250-Project\data\lmi-data’ already exists
```

```r
# URLs for the different regions, these are all the excell sheets on the Ohio LMI website for each regi
url_northeast <- "https://ohiolmi.com/_docs/PROJ/JobsOhio/Northeast.xlsx"
url_central <- "https://ohiolmi.com/_docs/PROJ/JobsOhio/Central.xlsx"
url_west <- "https://ohiolmi.com/_docs/PROJ/JobsOhio/West.xlsx"
url_southeast <- "https://ohiolmi.com/_docs/PROJ/JobsOhio/Southeast.xlsx"
url_northwest <- "https://ohiolmi.com/_docs/PROJ/JobsOhio/Northwest.xlsx"
url_southwest <- "https://ohiolmi.com/_docs/PROJ/JobsOhio/Southwest.xlsx"




# Process Northeast region First
temp_northeast <- tempfile(fileext = ".xlsx")
response_northeast <- GET(url_northeast, write_disk(temp_northeast, overwrite = TRUE)) #Calls the url,
  headers_northeast <- suppressMessages(read_excel(temp_northeast, range = cell_rows(3:6)))
  #Have to get rid of bad headers
  headers_northeast <- apply(headers_northeast, 2, function(x) paste(na.omit(x), collapse = " "))
  headers_northeast <- c(headers_northeast, "med wage symbol")
  data_northeast <- suppressMessages(read_excel(temp_northeast, skip = 5))
  #Skip the first 5 rows! all headers of white space.
  colnames(data_northeast) <- headers_northeast[1:12]
  #grab the names from only these headers
  rows_all_na_northeast <- rowSums(is.na(data_northeast)) == ncol(data_northeast)
  first_all_na_row_northeast <- which(rows_all_na_northeast)[1]
  data_northeast <- data_northeast[1:(first_all_na_row_northeast - 1), ]
  #that's annoying, but this should give us JUST the headers and not weird splits or missing headers.
  data_northeast$jobsohioregion <- "Northeast"


#OKAY, now do the same thing for all the other 5 regions, just past the above and change the region nam
# Process Central region___________________________________________________________
temp_central <- tempfile(fileext = ".xlsx")
response_central <- GET(url_central, write_disk(temp_central, overwrite = TRUE))
```

```r
  headers_central <- suppressMessages(read_excel(temp_central, range = cell_rows(3:6)))
  headers_central <- apply(headers_central, 2, function(x) paste(na.omit(x), collapse = " "))
  headers_central <- c(headers_central, "med wage symbol")
  data_central <- suppressMessages(read_excel(temp_central, skip = 5))
  colnames(data_central) <- headers_central[1:12]
  rows_all_na_central <- rowSums(is.na(data_central)) == ncol(data_central)
  first_all_na_row_central <- which(rows_all_na_central)[1]
    data_central <- data_central[1:(first_all_na_row_central - 1), ]
  data_central$jobsohioregion <- "Central"



# Process West region_______________________________________________________________
temp_west <- tempfile(fileext = ".xlsx")
response_west <- GET(url_west, write_disk(temp_west, overwrite = TRUE))
  headers_west <- suppressMessages(read_excel(temp_west, range = cell_rows(3:6)))
  headers_west <- apply(headers_west, 2, function(x) paste(na.omit(x), collapse = " "))
  headers_west <- c(headers_west, "med wage symbol")
  data_west <- suppressMessages(read_excel(temp_west, skip = 5))
  colnames(data_west) <- headers_west[1:12]
  rows_all_na_west <- rowSums(is.na(data_west)) == ncol(data_west)
  first_all_na_row_west <- which(rows_all_na_west)[1]
    data_west <- data_west[1:(first_all_na_row_west - 1), ]
  data_west$jobsohioregion <- "West"



# Process Southeast region___________________________________________________________________
temp_southeast <- tempfile(fileext = ".xlsx")
response_southeast <- GET(url_southeast, write_disk(temp_southeast, overwrite = TRUE))
  headers_southeast <- suppressMessages(read_excel(temp_southeast, range = cell_rows(3:6)))
  headers_southeast <- apply(headers_southeast, 2, function(x) paste(na.omit(x), collapse = " "))
  headers_southeast <- c(headers_southeast, "med wage symbol")
  data_southeast <- suppressMessages(read_excel(temp_southeast, skip = 5))
  colnames(data_southeast) <- headers_southeast[1:12]
  rows_all_na_southeast <- rowSums(is.na(data_southeast)) == ncol(data_southeast)
  first_all_na_row_southeast <- which(rows_all_na_southeast)[1]
    data_southeast <- data_southeast[1:(first_all_na_row_southeast - 1), ]
  data_southeast$jobsohioregion <- "Southeast"


# Process Northwest region___________________________________________________________________________
temp_northwest <- tempfile(fileext = ".xlsx")
response_northwest <- GET(url_northwest, write_disk(temp_northwest, overwrite = TRUE))
  headers_northwest <- suppressMessages(read_excel(temp_northwest, range = cell_rows(3:6)))
  headers_northwest <- apply(headers_northwest, 2, function(x) paste(na.omit(x), collapse = " "))
  headers_northwest <- c(headers_northwest, "med wage symbol")
  data_northwest <- suppressMessages(read_excel(temp_northwest, skip = 5))
  colnames(data_northwest) <- headers_northwest[1:12]
  rows_all_na_northwest <- rowSums(is.na(data_northwest)) == ncol(data_northwest)
  first_all_na_row_northwest <- which(rows_all_na_northwest)[1]
    data_northwest <- data_northwest[1:(first_all_na_row_northwest - 1), ]
  data_northwest$jobsohioregion <- "Northwest"
```

```r
# Process Southwest region____________________________________________________________________
temp_southwest <- tempfile(fileext = ".xlsx")
response_southwest <- GET(url_southwest, write_disk(temp_southwest, overwrite = TRUE))
  headers_southwest <- suppressMessages(read_excel(temp_southwest, range = cell_rows(3:6)))
  headers_southwest <- apply(headers_southwest, 2, function(x) paste(na.omit(x), collapse = " "))
  headers_southwest <- c(headers_southwest, "med wage symbol")
  data_southwest <- suppressMessages(read_excel(temp_southwest, skip = 5))
  colnames(data_southwest) <- headers_southwest[1:12]
  rows_all_na_southwest <- rowSums(is.na(data_southwest)) == ncol(data_southwest)
  first_all_na_row_southwest <- which(rows_all_na_southwest)[1]
    data_southwest <- data_southwest[1:(first_all_na_row_southwest - 1), ]
  data_southwest$jobsohioregion <- "Southwest"

# Combine all region datasets into a single data frame
lmi_oews <- bind_rows(data_northeast, data_central, data_west, data_southeast, data_northwest, data_sou
#OKAY! all Regions loaded.




#Ohio overall data_____________________________________________________________________________
# Define the column names manually, including the new 'median_wage_symbol'. This is because I cannot ge
column_names <- c(
  "soc_code",                     # SOC Code
  "soc_lmi_title",                # Occupational Title
  "employment",                   # Employment* 2020 Annual
  "projected_2030",               # 2030 Projected
  "change_employment",            # Change in Employment 2020-2030
  "percent_change",               # Percent
  "annual_openings_growth",       # Annual Openings Growth
  "exits",                        # Exits
  "transfers",                    # Transfers
  "total_openings",               # Total
  "median_wage",                  # Median Wage May 2021
  "median_wage_symbol",           # med wage symbol
  "Typical Education Needed for Entry",     # Not used in the select list
  "Work Experience in a Related Occupation",    # Not used in the select list
  "Typical On-The-Job Training Needed to Attain Competency" # Not used in the select list
)


# Read the data from the Excel file, skipping the first three rows. I could not get the url to read in
ohio_data <- read_excel(paste0("./data/lmi-data/OccOH30_raw.xlsx"),
                        sheet = "Occupational Detail", skip = 3, col_names = FALSE)
```

```
## New names:
## * ‘‘ -> ‘...1‘
## * ‘‘ -> ‘...2‘
## * ‘‘ -> ‘...3‘
## * ‘‘ -> ‘...4‘
## * ‘‘ -> ‘...5‘
## * ‘‘ -> ‘...6‘
## * ‘‘ -> ‘...7‘
## * ‘‘ -> ‘...8‘
## * ‘‘ -> ‘...9‘
## * ‘‘ -> ‘...10‘
## * ‘‘ -> ‘...11‘
## * ‘‘ -> ‘...12‘
## * ‘‘ -> ‘...13‘
## * ‘‘ -> ‘...14‘
## * ‘‘ -> ‘...15‘
```

```
ohio_data <- as.data.frame(ohio_data)
# Assign the manually defined column names to the data, these are defined above
colnames(ohio_data) <- column_names
# Add a new column 'jobsohioregion' with all values set to 'Ohio', this will give us the same manually
ohio_data <- ohio_data %>%
  mutate(jobsohioregion = 'Ohio')




#Combine Ohio and Region Data__________________________________________________________________________
# Ensure consistent column names and types for `ohio_data_trimmed`
ohio_data_trimmed <- ohio_data %>%
  select(
    soc_code, soc_lmi_title, employment, projected_2030,
    change_employment, percent_change, annual_openings_growth,
    exits, transfers, total_openings, median_wage,
    median_wage_symbol, jobsohioregion
  ) %>%
  mutate(
    employment = as.numeric(employment),  # Convert to numeric
    change_employment = as.numeric(change_employment),
    median_wage = as.numeric(median_wage),
    projected_2030 = as.numeric(projected_2030),
    percent_change = as.numeric(percent_change),
    annual_openings_growth = as.numeric(annual_openings_growth),
    exits = as.numeric(exits),
    transfers = as.numeric(transfers),
    total_openings = as.numeric(total_openings)
  )
```

```
## Warning: There were 9 warnings in ‘mutate()‘.
## The first warning was:
## i In argument: ‘employment = as.numeric(employment)‘.
## Caused by warning:
## ! NAs introduced by coercion
## i Run ‘dplyr::last_dplyr_warnings()‘ to see the 8 remaining warnings.
```

```
# Ensure column names and types match for `lmi_oews`
lmi_oews <- lmi_oews %>%
  rename(
    soc_code = `SOC Code`,
```

```r
    soc_lmi_title = `Occupational Title`,
    employment = `Employment* 2020 Annual`,
    projected_2030 = `2030 Projected`,
    change_employment = `Change in Employment 2020-2030`,
    percent_change = `Percent`,
    annual_openings_growth = `Annual Openings Growth`,
    exits = `Exits`,
    transfers = `Transfers`,
    total_openings = `Total`,
    median_wage = `Median Wage May 2021`,
    median_wage_symbol = `med wage symbol`
  ) %>% mutate(
    employment = as.numeric(employment),  # Convert to numeric
    projected_2030 = as.numeric(projected_2030),
    change_employment = as.numeric(change_employment),
    percent_change = as.numeric(percent_change),
    annual_openings_growth = as.numeric(annual_openings_growth),
    exits = as.numeric(exits),
    transfers = as.numeric(transfers),
    total_openings = as.numeric(total_openings),
    median_wage = as.numeric(median_wage)
  )
```

```
## Warning: There was 1 warning in ‘mutate()‘.
## i In argument: ‘median_wage = as.numeric(median_wage)‘.
## Caused by warning:
## ! NAs introduced by coercion
```

```r
# Ensure standardized column names for both data frames
colnames(ohio_data_trimmed) <- tolower(trimws(colnames(ohio_data_trimmed)))
colnames(lmi_oews) <- tolower(trimws(colnames(lmi_oews)))

# Combine the two datasets
ohio_region_lmi_data <- bind_rows(lmi_oews, ohio_data_trimmed)%>%
  mutate(
    jobsohioregion = case_when( #casewhen easiest in this case
      jobsohioregion == "Northwest" ~ 1L,
       jobsohioregion == "West" ~ 2L,
      jobsohioregion == "Southwest" ~ 3L,
      jobsohioregion == "Northeast" ~ 4L,
      jobsohioregion == "Central" ~ 5L,
      jobsohioregion == "Southeast" ~ 6L,
      jobsohioregion == "Ohio" ~ 39L, #ohio to 39, check this is true for all
      TRUE ~ NA_integer_  # For any unmatched regions, set to NA, should removed these or see why they
     )
    )
#Will have to fix manual vs hourly wage data later on it looks like. Pay attention to the wage symbol.
#SAVE the data
rda_file_path <- paste0(target_folder, "ohio_region_lmi_data.rda") #rda's always better (I think?)
save(ohio_region_lmi_data, file = rda_file_path)
```

```
#FOR FUTURE, ADD THE BELOW INSTRUCTIONS AND THE NEXT CHUNK'S INSTRUCTIONS TO A READ_ME FILE
# IPEDS Directory data -----
# https://nces.ed.gov/ipeds/use-the-data
# Survey Data > Custom Data Files
# Use provisional release data, continue
# Step 1 - Select Instituitions:
# Select "By Variables", "Browse/Search Variables"
#  Institutional Characteristics, Directory Information, select most recent year and "State abbreviatio
#   Under "Variable Title (Table Name)" click the link "State abbreviation - (17)" and check the box fo
# Click Continue to Step 2 - Select Variables
# + Institutional Characteristics
# + Directory information, response status and frequently used variables
# + Directory information and response status:
# NOT NEEDED: Institution (entity) name -- they give you this by default, and if you request it, you ge
# Institution name alias
# Street address or post office box
# City location of institution
# State abbreviation
# ZIP code
# General information telephone number
# Institution's internet website address
# Employer Identification Number
# Fips County code
# County name
# Longitude location of institution
# Latitude location of institution
# UNITID for merged schools
# Year institution was deleted from IPEDS
# Date institution closed
# Institution is active in current year
# + Institution Classifications:
# Sector of institution
# Level of institution
# Control of institution
# Highest level of offering


# Hit Continue to move on to a page listing the requested data.
# Select "STATA", which actually will produce a CSV but uses codes instead of value labels, which is go
# .do files are also provided for each, should there be any question about value labels.
# Get JOR codes to attach to the IPEDS directory data
load('data/cross-walks/jobsohioregions.rda')

ipeds_directory <- read_csv('data/ipeds-institution-detail/STATA_RV_7162021-493.zip') %>%
  left_join(jobsohioregions, by = c('countycd' = 'statefips')) %>%
  transmute(
    ipeds_code = unitid,
    institutionname = instnm,
    street_address = addr,
    city = city,
    state = stabbr,
    zip = zip,
    web_address = webaddr,
    regionId = jobsohioregion,
    lat = latitude,
    lng = longitud
  )
```

```
## Multiple files in zip: reading 'STATA_RV_7162021-493.csv'
## Rows: 296 Columns: 23
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (10): instnm, ialias, addr, city, stabbr, zip, webaddr, ein, countynm, c...
## dbl (13): unitid, year, gentele, countycd, longitud, latitude, newid, deathy...
##
## 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.
```

```
save(ipeds_directory, file = 'data/ipeds-institution-detail/ipeds_directory.rda')
```

```
# IPEDS Data -----
# https://nces.ed.gov/ipeds/use-the-data
# Survey Data > Custom Data Files
# Step 1 - Select Instituitions:
# Select "By Variables", "Browse/Search Variables"
#  Institutional Characteristics, Directory Information, select most recent year and "State abbreviatio
#   Under "Variable Title (Table Name)" click the link "State abbreviation - (##)" and check the box fo
# Click Continue to Step 2 - Select Variables
# For each year wanted under Available Year(s) do:
#   click the year (each year refers to school year ending June 30 of that year)
#   Completions, Awards/degrees conferred by program (CIP), check Grand total
#   (repeat)
# Hit Continue to move on to a page listing each of the requested data sets.
# For each one, select "STATA", which actually will produce CSVs but uses codes instead of value labels
# .do files are also provided for each, should there be any question about value labels.



#Usefull to get the remappings for award level, ie. these are the education levels.
## From the STATA .do file from IPEDS for 1997:
# label values cipcode         label_cipcode
# label define label_awlevel         15 "Degrees/certificates total"
# label define label_awlevel         12 "Degrees total", add
# label define label_awlevel         3 "Associate''s degree", add
# label define label_awlevel         5 "Bachelor''s degree", add
# label define label_awlevel         7 "Master''s degree", add
# label define label_awlevel         9 "Doctor''s degree", add
# label define label_awlevel         10 "First-professional degree", add
# label define label_awlevel         13 "Certificates below the bacculaureate total", add
# label define label_awlevel         1 "Award of less than 1 academic year", add
# label define label_awlevel         2 "Award of at least 1 but less than 2 academic years", add
# label define label_awlevel         4 "Award of at least 2 but less than 4 academic years", add
# label define label_awlevel         14 "Certificates above the bacculaureate total", add
# label define label_awlevel         6 "Postbaccalaureate certificate", add
# label define label_awlevel         8 "Post-master''s certificate", add
# label define label_awlevel         11 "First-professional certificate", add
```

```
#Using the above category definitions from the STATA file you can download from IPEDS, let's remap to l
#so we actually know what is goin on
ipeds_degree_remapping <- tribble(
  ~awlevel, ~degree_group_logord,
  '1',        1L,
  '2',        1L,
  '3',        2L,
  '4',        1L,
  '5',        3L,
  '6',        1L,
  '7',        4L,
  '8',        5L,    # grad certificate, has not been included in the Supply Tool
  '9',        4L,
  '10',         4L,
  '11',         5L,    # grad certificate, has not been included in the Supply Tool
  '12',       NA,    # subtotals
  '13',       NA,    # subtotals
  '14',       NA,    # subtotals
  '15',       NA,    # subtotals
  '17',         4L,
  '18',         4L,
  '19',         4L
)

# Read files, keep only 6-digit CIP, address some variable name changes (crace24/ctotalt)
# Using default character because it is easier to start from there, keep CIP codes correct,

#First, use list.files to find the .zip files that download from IPEDS, better to store them as .zip, b
ipeds_completions <- list.files('data/ipeds-completions', '.*zip$', full.names = TRUE) %>%
  map_dfr(~ read_csv(., col_types = cols(.default = col_character()))) %>%
  filter(nchar(cipcode) == 7) %>%  # 7 because of the "." in the number, e.g. "15.0101"
  mutate(grads = as.integer(ctotalt)) %>% #this is the grads count column
  left_join(ipeds_degree_remapping, by = 'awlevel') %>%
  filter(!is.na(degree_group_logord) & grads > 0) %>% # drop subtotals and zero rows
  group_by(unitid, year, cipcode, degree_group_logord) %>%  # this is for combining majornum = 1 and ma
  summarise(graduates = sum(grads), .groups = 'drop') %>%
  left_join(transmute(ipeds_directory, unitid = as.character(ipeds_code), regionId), by = 'unitid') %>%
  select(ipeds_code = unitid,
         cip_code = cipcode,
         degree_group_logord,
         academic_year = year,
         jobsohioregion = regionId,
         graduates)
```

```
## Multiple files in zip: reading ’STATA_RV_3172022-1009.csv’
## Multiple files in zip: reading ’STATA_RV_3172022-1030.csv’
## Multiple files in zip: reading ’STATA_RV_3172022-141.csv’
## Multiple files in zip: reading ’STATA_RV_3172022-185.csv’
## Multiple files in zip: reading ’STATA_RV_3172022-301.csv’
## Multiple files in zip: reading ’STATA_RV_3172022-502.csv’
## Multiple files in zip: reading ’STATA_RV_3172022-620.csv’
## Multiple files in zip: reading ’STATA_RV_3172022-893.csv’
## Multiple files in zip: reading ’STATA_RV_3172022-949.csv’
## Multiple files in zip: reading 'STATA_RV_3172022-974.csv'
## Multiple files in zip: reading 'STATA_RV_582024-207.csv'
## Multiple files in zip: reading 'STATA_RV_962022-18.csv'
```

```
save(ipeds_completions, file = 'data/ipeds-completions/ipeds_completions.rda')
```

##End OF data Import, now need to Combine according to CIP-SOC Crosswalk

Final Datasets Created:

ohio_region_lmi_data: Occupation demand dataset that includes six Ohio regions and statewide data (jobsohioregion coded numerically for each region).

Main Variables: -soc_code: Standard Occupational Classification code. -soc_lmi_title: Occupation title based on LMI. -employment: Employment count for 2020. -projected_2030: Projected employment count for 2030. -change_employment: Change in employment from 2020 to 2030. -percent_change: Percentage change in employment. -annual_openings_growth: Annual growth in job openings. -median_wage: Median wage in 2021. -jobsohioregion: Region identifier (1-6 for regions, 39 for Ohio).

ipeds_completions.rda:IPEDS completions data for institutions in Ohio, linked to LMI regions.

Main Variables: -ipeds_code: Unique identifier for institutions. -cip_code: Classification of Instructional Programs code for program areas. -degree_group_logord: Ordinal representation of degree levels (e.g., 1 for -certificates, 2 for associate degrees, 3 for bachelor's degrees). -academic_year: Year of data collection. -jobsohioregion: Region identifier linked to LMI regions. -graduates: Number of graduates in a given program and year.

Including Plots

You can also embed plots, for example:

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.