UIC Weight and Veterans' Environments Study

Employee density in select industries as an indicator of walkable destination density: Weight and Veterans' Environments Study GIS protocol

Xiang W, Jones KK, Matthews SA, Zenk SN.

UIC Neighborhoods + Health

Overview

This protocol describes the construction of employment measures based on data from the Longitudinal Employer-Household Dynamics program.

Acknowledgements

This protocol was developed with grant support from the National Cancer Institute (R21CA195543) and the Department of Veterans Affairs (IIR 13-085), co-led by Shannon Zenk and Elizabeth Tarlov. Haytham Abu Zayd, Shubhayan Ukil and Abby Klemp helped to edit the protocol for dissemination.

Suggested Citation

Xiang W, Jones K, Matthews SA, Zenk SN. (2018). Employee density in select industries as an indicator of walkable destination density: Weight and Veterans' Environments Study GIS protocol, Version 1. Retrieved from Weight and Veterans' Environments Study website: https://waves.uic.edu/.

Table of Contents

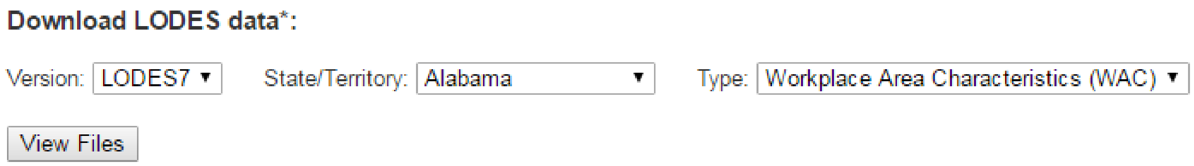
Background

This document describes the work process for constructing Longitudinal Employer-Household Dynamics (LEHD) measures.

Data

Sources

The data source is 2008 through 2014 LEHD Origin-Destination Employment Statistics (LODES) data downloaded from https://lehd.ces.census.gov/data/#lodes (Date of Access: Winter 2016/2017). The data is separated by state, and at census block level. The version is LODES7, which, according to the documentation is based on 2010 census geography, however, we found that it was based on 2014 geographies. Type is Workplace Area Characteristics (WAC), as seen in the following screenshot.



Download LODES data*:

Version: LODES7 ▼ State/Territory: Alabama ▼ Type: Workplace Area Characteristics (WAC) ▼

The technical documentation is available at https://lehd.ces.census.gov/data/lodes/LODES7/LODESTechDoc7.2.pdf.

Definitions

LEHD variables definition and years

The following variables were chosen to represent total and service industry jobs from LEHD:

- C000 "Total number of jobs"
- CNS07 "Number of jobs in NAICS sector 44-45 (Retail Trade)"
- CNS18 "Number of jobs in NAICS sector 72 (Accommodation and Food Services)"
- CNS17 "Number of jobs in NAICS sector 71 (Arts, Entertainment, and Recreation)"

From these variables, we make two variables for grids:

- Total number of jobs (C000)
- Number of jobs in Retail + Accommodation and Food Services + Arts, Entertainment, and Recreation (CNS07+CNS18+CNS17)

For states/territories with missing data in given years, the following data years were used:

DC: 2008 and 2009 (used 2010 for missing years)
MA: 2008, 2009, 2010 (used 2011 for missing years)
WY: 2014 (used 2013 for missing year)

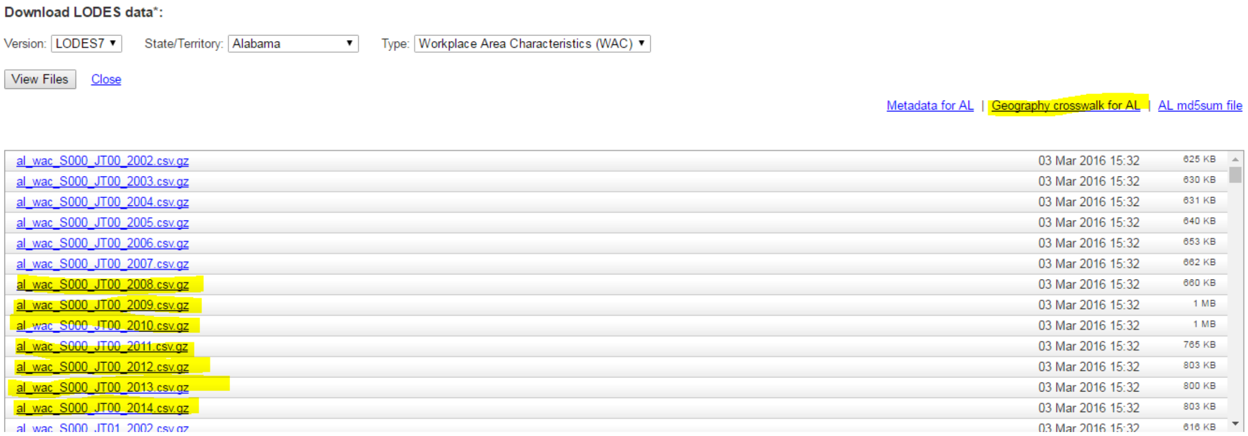
Cleaning

Raw data download

From https://lehd.ces.census.gov/data/#lodes, the data to download is the All Job data - "JT00" file.

Because the raw data is at the census block level, and we need the data aggregated to census block group level, so there is also a crosswalk table for each state that needs to be downloaded as well.

For example, in the screenshot below, the highlighted are the files needed for Alabama, and the same process repeats for every other state.



Download LODES data*:

Version: LODES7 ▾ State/Territory: Alabama ▾ Type: Workplace Area Characteristics (WAC) ▾

View Files Close

Metadata for AL | Geography crosswalk for AL | AL md5sum file

al_wac_S000_JT00_2002.csv.gz	03 Mar 2016 15:32	625 KB
al_wac_S000_JT00_2003.csv.gz	03 Mar 2016 15:32	630 KB
al_wac_S000_JT00_2004.csv.gz	03 Mar 2016 15:32	631 KB
al_wac_S000_JT00_2005.csv.gz	03 Mar 2016 15:32	640 KB
al_wac_S000_JT00_2006.csv.gz	03 Mar 2016 15:32	653 KB
al_wac_S000_JT00_2007.csv.gz	03 Mar 2016 15:32	662 KB
al_wac_S000_JT00_2008.csv.gz	03 Mar 2016 15:32	660 KB
al_wac_S000_JT00_2009.csv.gz	03 Mar 2016 15:32	1 MB
al_wac_S000_JT00_2010.csv.gz	03 Mar 2016 15:32	1 MB
al_wac_S000_JT00_2011.csv.gz	03 Mar 2016 15:32	765 KB
al_wac_S000_JT00_2012.csv.gz	03 Mar 2016 15:32	803 KB
al_wac_S000_JT00_2013.csv.gz	03 Mar 2016 15:32	800 KB
al_wac_S000_JT00_2014.csv.gz	03 Mar 2016 15:32	803 KB
al_wac_S000_JT01_2002.csv.gz	03 Mar 2016 15:32	616 KB

The downloaded file format is .gz file, and all .gz files can be batch unzipped using WinZip or 7-Zip to .csv format. The downloaded file is named as LongitudinalEmployerHouseholdDynamics(LEHD)\Data\RawData\

Append separate files to create one national file per year and collapse the data by block group

Append files to create national file

Separate downloaded files were merged using Script A of Appendix: Scripts.

For reference in further scripts, the output is saved as "full_YEAR.csv".

The same process is done to append the crosswalk tables as well.

Collapse the data by block group

Because the data is at census block level, and we would like the data to be aggregated to census block group level, so the crosswalk table provides the information of which census block geoid corresponds to which census block group geoid. The data is processed in Stata to aggregate the values for the same block group together from block level.

This process is done in Stata, and the DO file is given in Script B, Appendix: Scripts

For reference in future scripts, the output for the block group level files are saved as "full_YEAR_byBG.csv".

Create cell value for making grids

After data is aggregated to block group level, we need to create the "cell value" for each block group, for the purpose of creating a single raster surface covering the entire continental US. Because in all grids measures, a cell is defined as 30 by 30 meters square, the process is to distribute the value for one block group to each 900 sq meters cell, assuming the value is distributed evenly.

This process is done in Stata, and the DO file is given in Script C, Appendix: Scripts.

The DO file does the following process:

- Calculate the 30x30 meter cell value from census block group value and census block area, detailed formula as:

  Cell value within one block group = (LEHD variable value for this block group/total area in sq. meters for this block group) *900

- For LEHD variables, the geography is 2014 census geography, so there are also geography changes applied in the DO file script in order to join the processed cell value to 2014 geography in GIS
- From all LEHD variables, keep Total number of jobs (C000), calculate the sum of Number of jobs in Retail + Accommodation and Food Services + Arts, Entertainment, and Recreation (CNS07+CNS18+CNS17), and also keep these 3 as separate variables.

Decisions

Join cell value to GIS block group file

To create the LEHD grids, the first step is to join the LEHD cell value derived from the last step to the GIS census block group polygon file.

The join cell value to block group process and is done manually in the following steps:

1. Add the cell value .csv file in ArcMap, export it to a geodatabase as a table. Run "Add Attribute Index" tool to add index to "geoid10" field.
2. Locate the block group polygon file and create a copy of this file for each year. For example, the LEHD data has 7 years, so we need 7 block group polygon files. Include the year variable (e.g. 2008) in the polygon feature class name.
3. For each year of polygon file, Run "Join Field" tool based on "geoid10" to join the LEHD same year's cell value data permanently to the block group GIS file.
4. After join is completed, run "Add Attribute Index" tool to add index to "geoid10", and all the cell value fields.

5. *This process needs to be run once for each cell value field:* Use the joined block group polygon, use cell value field as the value field, run "Polygon to Raster" tool, store the raster in a folder (NOT in geodatabase). This takes about 2-5 hours for one raster. This process can also be run in Python script, which is given in Script D, Appendix: Scripts.

The raster files derived from the above step 5 will be used as input to create the grids.

LEHD grids generation process

Use the base raster files generated from the step 5. Python script (given in Script E, Appendix: Scripts) creates the LEHD grids for 400m and 1600m

The script performs the following function:

- The script needs to be run once for each year's measure by changing the parameters
- Create 2 grids per buffer – 400m and 1600m. The 2 grids are 1) Total number of jobs and 2) Number of jobs in Retail + Accommodation and Food Services + Arts, Entertainment, and Recreation

Appendix

Software

The software used is ArcGIS 10.3.1 and Python 2.7.

Scripts

Note: Script will need to be adjusted with project-specific file names and locations.

A: Csv Combine

```
# This script appends multiple csv files together as one long file
# only keep the column header from the 1st file, remove the others
# Written by Aster Xiang
import os
import csv, sys, fileinput, itertools
from datetime import datetime, date, time
from collections import defaultdict

years = ["2008","2009","2010","2011","2012","2013","2014","xwalk"]

for year in years:
	dir = {insert file location}
```

```
        output_dir = {insert file location}
        outfile = "full_"+year+".csv"
        input = []

        #store all csv file in list
        for file in os.listdir(dir):
                if file.endswith(".csv"):
                        input.append(file)

        with open(output_dir+outfile,'wb') as out:
                writer = csv.writer(out)
                for count,file in enumerate(input):
                        with open(dir+file,'rb') as inp:
                                reader = csv.reader(inp)
                                all = []

                                if count==0:# add the column header for the first file only
                                        row = next(reader)#row equals to first line right now
                                        all.append(row)
                                if count>0: # starting from the 2nd file, skip the column header
                                        reader.next()
                                for row in reader:#loop second line and after
                                        all.append(row)
                                writer.writerows(all)
                print outfile
```

B: Collapse by BG

```
******************************************************************************
* pgm: collapse_byBG.do
* purpose:  (1) use xwalk file to merge BG to CT
*                       (2) collapse variables by block group
******************************************************************************

capture log close _all
log using {insert file location}, replace
clear all
set more off

cd {insert location}

import delimited xwalk\full_xwalk.csv, stringcols(1 9) clear
keep tabblk2010 bgrp
rename tabblk2010 w_geocode
sort w_geocode
```

```
*tempfile xwalk
*save `xwalk'
save xwalk\full_xwalk_withBG.dta, replace

foreach year in 2008 2009 2010 2011 2012 2013 2014 {
        clear
        import delimited `year'\full_`year'.csv, stringcols(1) clear
        sort w_geocode
        *merge w_geocode using `xwalk'
        merge w_geocode using xwalk\full_xwalk_withBG
        drop createdate _merge
        collapse (sum) c000 - cfs05, by(bgrp)
        * export to .csv
        export delimited `year'\full_`year'_byBG.csv, replace}

log close _all
```

C: LED by Cell

```
*Housing Distribution by 30m cell

clear all
cap log close
set more off

log using {insert file location}, replace

******
*This DO file allows the user to assign block groups population density per 900sq meters. This is necessary so that there can be a grid made of 30mX30m cell size
******



***The LEHD data is always based on 2014 geography.

**For 2008, use 2010 geographies, apply 2011, 2012, 2014 changes.

import delimited {insert LEHD file location},  stringcols(1) clear
la var bgrp "Block Group 2010 geoid"
la var c000 "Total number of jobs"
rename c000 ttl_job
la var cns07 "Number of jobs in NAICS sector 44-45 (Retail Trade)"
rename cns07 retail
la var cns18 "Number of jobs in NAICS sector 72 (Accommodation and Food Services)"
rename cns18 accom_food
```

```
la var cns17 "Number of jobs in NAICS sector 71 (Arts, Entertainment, and Recreation)"
rename cns17 arts


gen str blockgroup2010 = bgrp
gen str blockgroupnumber = substr(bgrp,12,1)
gen str censustract2013 = substr(bgrp,1,11)

*2011 changes:
replace blockgroup2010 = "36053940101"+blockgroupnumber if censustract2013 == "36053030101"
replace blockgroup2010 = "36053940102"+blockgroupnumber if censustract2013 == "36053030102"
replace blockgroup2010 = "36053940103"+blockgroupnumber if censustract2013 == "36053030103"
replace blockgroup2010 = "36053940200"+blockgroupnumber if censustract2013 == "36053030200"
replace blockgroup2010 = "36053940300"+blockgroupnumber if censustract2013 == "36053030300"
replace blockgroup2010 = "36053940401"+blockgroupnumber if censustract2013 == "36053030401"
replace blockgroup2010 = "36053940403"+blockgroupnumber if censustract2013 == "36053030403"
replace blockgroup2010 = "36053940600"+blockgroupnumber if censustract2013 == "36053030600"
replace blockgroup2010 = "36053940700"+blockgroupnumber if censustract2013 == "36053030402"

*Oneida County, NY
replace blockgroup2010 = "36065940000"+blockgroupnumber if censustract2013 == "36065024800"
replace blockgroup2010 = "36065940100"+blockgroupnumber if censustract2013 == "36065024700"
replace blockgroup2010 = "36065940200"+blockgroupnumber if censustract2013 == "36065024900"

*2012 changes:
*1. Numbering of 7 census tracts in Pima County, AZ is changes with no geography changes

        replace blockgroup2010 = "04019002701"+blockgroupnumber if censustract2013 ==
"04019002704"
        replace blockgroup2010 = "04019002903"+blockgroupnumber if censustract2013 ==
"04019002906"
        replace blockgroup2010 = "04019410501"+blockgroupnumber if censustract2013 ==
"04019004118"
        replace blockgroup2010 = "04019410502"+blockgroupnumber if censustract2013 ==
"04019004121"
        replace blockgroup2010 = "04019410503"+blockgroupnumber if censustract2013 ==
"04019004125"
        replace blockgroup2010 = "04019470400"+blockgroupnumber if censustract2013 ==
"04019005200"
        replace blockgroup2010 = "04019470500"+blockgroupnumber if censustract2013 ==
"04019005300"


        *2. The deletion of Census 2000 tract 1370.00 is corrected in Los Angeles County, CA
        replace blockgroup2010 = "060378002043" if blockgroup2010 == "060371370002"
        replace blockgroup2010 = "060379304011" if blockgroup2010 == "060371370001"


*2014 Changes
*Bedford City was absorbed into Bedford County, Virginia, and its 5-digit FIPS code has been eliminated.
gen county = substr(bgrp,1,5)
gen noncounty = substr(bgrp,6,7)
replace blockgroup2010 = "51019"+noncounty if county == "51515"

*other changes
replace blockgroup2010 = "46113940500"+blockgroupnumber if censustract2013 == "46102940500"
replace blockgroup2010 = "46113940800"+blockgroupnumber if censustract2013 == "46102940800"
replace blockgroup2010 = "46113940900"+blockgroupnumber if censustract2013 == "46102940900"

replace blockgroup2010 = "51515050100"+blockgroupnumber if censustract2013 == "51019050100"


rename blockgroup2010 geoid10
merge 1:1 geoid10 using {insert BG area file location here}

drop if _merge==2

*create # jobs Retail+Food+Arts
gen rtl_fd_art = retail+accom_food+arts
*create cell value
* # total jobs
gen cell_ttl_job=(ttl_job/BG_sqmeter)*900
* # retail jobs
gen cell_retail = (retail/BG_sqmeter)*900
* # accommodation food
gen cell_accom_food = (accom_food/BG_sqmeter)*900
* # Arts
gen cell_arts = (arts/BG_sqmeter)*900
* # Retail+Food+Arts
gen cell_rtl_fd_art = (rtl_fd_art/BG_sqmeter)*900

*drop state county tract blkgrp geoid name BG_ttl_hu BG_pct_vhu BG_pct_oh BG_pct_ooh BG_0HU
BG_0OH b25004e1 BG_ttl_ooh bg_ACS_2007_2011 blockgroupnumber censustract2011 BG_sqmeter
strlength zero _merge
keep geoid10 cell_ttl_job cell_retail cell_accom_food cell_arts cell_rtl_fd_art


clear
```

*Follow same procedure for subsequent years

D: LED Polygon to Raster

```
#This script converts polygon to raster
import arcpy,sys, os, time
from arcpy.sa import *


in_workspace = {insert file location}
out_workspace = {insert file location}
arcpy.env.workspace = in_workspace
arcpy.env.extent = "-2493045.0 -1429501.25 2342655.0 1703218.75"
arcpy.env.overwriteOutput = True
arcpy.CheckOutExtension("Spatial")


featureList = arcpy.ListFeatureClasses()
del featureList[:3]
print featureList

for count,feature in enumerate(featureList):
                start = time.time()
                year = feature[20:22]
                out_name_ttl_job = year+"ttl_job"
                out_name_rtl_fd_art = year+"rtl_fd_art"
                lyr_feature = "lyr"+str(count)
                arcpy.MakeFeatureLayer_management(feature, lyr_feature)

        arcpy.PolygonToRaster_conversion(lyr_feature,"cell_ttl_job",out_workspace+out_name_ttl_job
,"CELL_CENTER","",30)

        arcpy.PolygonToRaster_conversion(lyr_feature,"cell_rtl_fd_art",out_workspace+out_name_rtl_
fd_art,"CELL_CENTER","",30)
                end = time.time()
                print str((end-start)/60)+" minutes"
                print feature
```

E: LED Grids Generation

```
#housing grids generation
#only need 400m and 1600m
import arcpy, time
from arcpy.sa import *
```

```
in_workspace = {insert location}
out_workspace_14 = {insert location}

arcpy.env.workspace = in_workspace
arcpy.env.extent = "-2493045.0 -1429501.25 2342655.0 1703218.75"
arcpy.env.overwriteOutput = True
arcpy.CheckOutExtension("Spatial")

start = time.time()

#set focal statistics variables

neighborhood_400 = NbrCircle(400, "MAP")
neighborhood_1600 = NbrCircle(1600, "MAP")

#2014
# total job
# Set focal statistics variables
lyr_ttl_job = "Total_job_2014"
arcpy.MakeRasterLayer_management("14ttl_job", lyr_ttl_job)
inRaster = lyr_ttl_job
# Execute FocalStatistics
outFocalStatistics = FocalStatistics(inRaster, neighborhood_400, "SUM","")
outFocalStatistics.save(out_workspace_14+"LED_2014_ttl_job_400")
print inRaster
outFocalStatistics = FocalStatistics(inRaster, neighborhood_1600, "SUM","")
outFocalStatistics.save(out_workspace_14+"LED_2014_ttl_job_1600")
print inRaster
end = time.time()
print str((end-start)/60)+" minutes"

# total retail+food+art
# Set focal statistics variables
lyr_rtl_fd_art = "rtl_fd_art_2014"
arcpy.MakeRasterLayer_management("14rtl_fd_art", lyr_rtl_fd_art)
inRaster = lyr_rtl_fd_art
# Execute FocalStatistics
outFocalStatistics = FocalStatistics(inRaster, neighborhood_400, "SUM","")
outFocalStatistics.save(out_workspace_14+"LED_2014_rtlfdart_400")
print inRaster
outFocalStatistics = FocalStatistics(inRaster, neighborhood_1600, "SUM","")
outFocalStatistics.save(out_workspace_14+"LED_2014_rtlfdart_1600")
print inRaster
```

```
end = time.time()
print str((end-start)/60)+" minutes"
```