Access to Public Transit in Benton, Linn, and Lincoln Counties

Detailed methodology

Resources needed to recreate this methodology:

1. ArcGIS ArcMap
2. Microsoft Excel
3. R statistical software (RStudio recommended)
4. Microsoft Power BI

Data sources needed:

Filename/tables/data frames use this font

Field/Column names use this font

1. Blocks2010.shp: Shapefile with 2010 U.S. Census Blocks, containing
   1. GEOID10: 15 digit Geographic ID for each block
   2. ALAND10: Land area of each block
   3. POP10: 2010 Census block population
2. BlockGroups2010.xlsx: Excel file with 2010 U.S. Census Blockgroups, containing
   1. BG\_GEOID10: 12 digit Geographic ID for each blockgroup
   2. BG\_POP10: 2010 Census blockgroup population
3. BlockGroups2017.xlsx: Excel file with 2013-2017 U.S. Census Blockgroups, containing
   1. BG\_GEOID10: 12 digit Geographic ID for each blockgroup
   2. BG\_POP17: 2013-2017 Census blockgroup population
   3. Any additional subpopulations (e.g. Children, Seniors, Disability Status, etc.). See Appendix 1 for FactFinder table numbers
4. Cities.shp: Shapefile with city boundaries
5. Transit\_stops.xlsx: Excel file with transit stops, containing
   1. Latitude and Longitude
   2. Metadata, such as transit system, route type, average frequency, and other desired characteristics
   3. Note: If you have access to a shapefile with transit stops already, you can export the data with the latitude and longitude and add the other metadata. Otherwise you can laboriously find all the transit stops and use Google Maps to get their latitude and longitude.
6. Populations\_table.xlsx: Excel file with metadata about the subpopulations, containing
   1. The names of the subpopulations (Total Population, Children, Seniors, etc.)
   2. The order you want the subpopulations to show up in the Power BI visuals.
   3. Other metadata you may want
7. CityList.xlsx: Excel file with the cities that will be included when filtering to urban areas, containing:
   1. The cities
   2. The county/geography each city is in.

Step 1: Geoprocessing in ArcGis

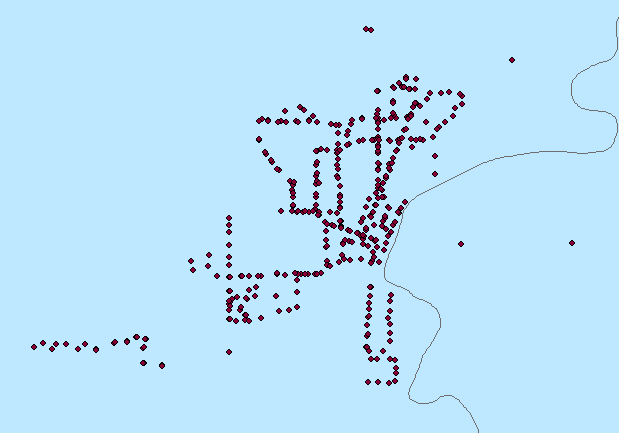
This first step will geographically identify the blocks and/or partial blocks that are within 1/4 mile of a transit stop and calculate the area of these blocks and/or partial blocks.

Files used in geoprocessing step

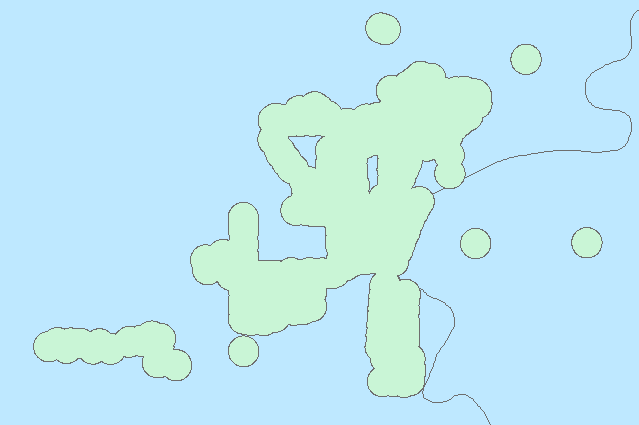
* Transit\_stops.xlsx
* Blocks2010.shp
* Cities.shp

Steps

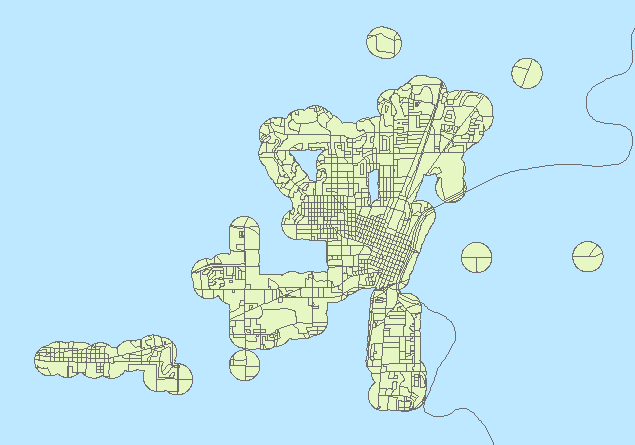
1. Load Transit\_stops.xslx into ArcGis
2. Rightclick on the Transit\_stops.xlsx and select “Display XY Data”. This will calculate the Latitude and Longitude and create a layer.



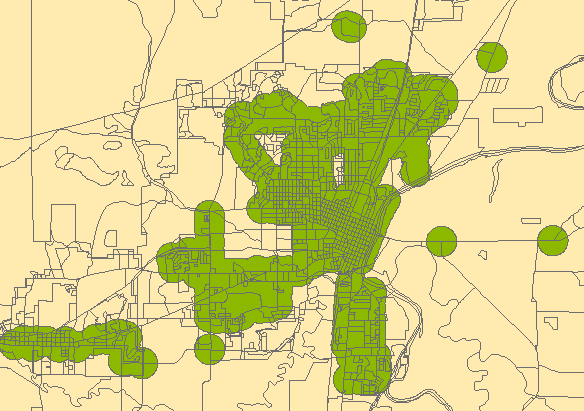
1. Rightclick on the layer and export the data as Transit\_stops.shp.
2. To select a subset of the full set of transit stops, use Select by Attribute. For example, you can select just the local transit stops. Create a layer from the selection and export the data as its own shapefile, e.g. Transit\_local\_stops.shp.
3. For each type of transit stops you have selected:
   1. In the menu bar, click Geoprocessing:Buffer. Calculate a buffer around the transit stops, choosing your buffer radius and setting Dissolve Type to All. Name the buffer layer Transit\_buffer.shp.



1. Load Blocks2010.shp into ArcGIS.
2. In the menu bar, click Geoprocessing:Clip. Clip Blocks2010 using the buffer. This will create a layer of Census blocks that are clipped to the boundary of the buffer. If a block is completely contained within the buffer, the whole block will be kept. If the block straddles the boundary of the buffer, it will be geographically cut to shape. Save this layer as Transit\_blocks.shp.



1. In the menu bar, click Geoprocessing:Union. Union Transit\_blocks.shp with Blocks2010.shp. This will recreate the full layer of census blocks, except any census blocks that straddled the buffer boundary will create two new partial census blocks. This will allow you to calculate the proportionate area within and outside the buffer for each census block. Save this layer as Transit\_blocks\_union.shp.



1. Some blocks may geographically separated (non-contiguous) because of the buffer clipping. In ArcToolbox, use DataManagement:Features:Multipart to Singlepart to separate non-contiguous blocks so that they are each their own element. This will require saving it as a new shapefile, called Transit\_blocks\_mp.shp
2. Open the attribute table of Transit\_blocks\_mp.shp. Add a new field as Double, called ClipArea
   1. You can also delete extraneous fields if you want, but be sure to keep all fields that start with FID, with GEOID10, with ALAND10, and with POP10.
3. Populate ClipArea using Calculate Geometry with Area in square meters (assuming ALAND10 is in square meters). This file is now ready for processing in R.
4. If you are analyzing subsets of the transit system, like local routes:
   1. Make buffers for each of the subsets, e.g. Transit\_local\_buffer.shp.
   2. Clip Transit\_blocks\_mp.shp with Transit\_local\_buffer.shp and save it as Transit\_blocks\_local.shp. These files are now ready for processing in R.
5. If you are analyzing city versus rural:
   1. Load Cities.Shp into ArcGIS
   2. In the menu bar, click Selection:Select by location. Select Blocks2010 with Cities as the source layer. Create a new layer from this selection, save it as CityBlocks.shp. Open the attribute table and Export the table as CityBlocks.xlsx.
   3. You can delete all fields other than the 15-digit GEOID10 and POP10. This file is now ready for processing in R.

Output from geoprocessing steps:

* Transit\_stops.shp
* Transit\_buffer.shp
* Transit\_blocks.shp
* Transit\_blocks\_union.shp
* Transit\_blocks\_mp.shp
* CityBlocks.xlsx
* Any subsets of Transit\_stops.shp, Transit\_buffer.shp, and Transit\_blocks\_mp.shp, like Transit\_blocks\_local.shp, etc.

Step 2: Processing in R

These steps will estimate the proportion of the population within 1/4 mile of a transit stop in 2017. Below is a summary of the steps. The full code is available here:

Files used in R processing step:

* Transit\_blocks\_mp.shp, and any subset like Transit\_blocks\_local.shp
* BlockGroups2010.xlsx. In the code this is called lbl\_bg\_pop10.csv
* BlockGroups2017.xlsx. In the code this is called blockgroup\_lbl\_transit\_2017.txt
* CityBlocks.xlsx. In the code this is called city\_blocks.csv

Steps:

1. Load the following packages:

maptools, rgdal, data.table, reshape2, plyr

1. Read in BlockGroups2010.xlsx, BlockGroups2017.xlsx, CityBlocks.xslx, and Populations\_table.xlsx
2. Merge CityBlocks with BlockGroups2017 and calculate the percent of blockgroup population inside cities.
3. Calculate populations and proportions across the largest geography using BlockGroups2017, by subpopulation. Merge the proportions to Populations\_table and save the table. This will be used when making the map of transit access.
4. Set the coordinate system to match the shapefiles.
5. Read in Transit\_blocks\_mp.shp and store it as a data frame (or data table if you prefer). I stored it as transit\_df.
6. Since transit\_df is built from a union of two shapefiles, many of the columns have blank entries (like ALAND10). Combine ALAND10 with ALAND10\_1 to get a column with no blanks. Repeat this with COUNTYFP10, GEOID10, and POP10.
7. Create the following variables in transit\_df:
   1. PctArea = ClipArea/ALAND10 # The proportion of land area in each block or partial block. Blocks that are wholly contained in or wholly excluded from the buffer will have PctArea = 1. Blocks that are split will have PctArea < 1.
   2. transit\_type = 1 if the block or partial block is in the buffer and 0 if it is not in the buffer.
8. Store this new data frame as tcomb.
9. Read in each of the subset transit blocks shape files like Transit\_blocks\_local.shp. Merge them to tcomb to get a data frame that contains one column for each route type, flagging blocks for inclusion or exclusion.

The next steps estimate the percent of 2017 population in each block/partial block. There are some assumptions:

* 1. People were uniformly distributed across blocks in 2010. This assumption was used in step 8a.
  2. Population growth was equal across all blockgroups, so that the percentage increase in any given blockgroup from 2010 to 2017 was the same. This assumption is used in step 12.
  3. People are uniformly distributed across blocks in 2017. This assumption is used in step 13.
  4. These assumptions hold for each population subgroup.

1. Merge tcomb with BlockGroups2010 and calculate each block/partial blocks proportion of the 2010 Blockgroup population. Call this blockPop10Pct = POP10\*BlockGroupPOP10.
2. Multiply blockPop10Pct by the blockgroup population in 2017 to get the block population in 2017. Call this Pop17block=blockPop10Pct\*BlockGroupPOP17.
3. Multiply Pop17block by PctArea to get the population in each block/partial block in 2017. Call this BufferPop17. This will be used to calculate the proportion of the population with transit access.
4. For each county/large geography, and for each population type, and for each route type, sum the population in BufferPop17 and divide it by the total geography population for that population type. This is the proportion of the population with access to that route type. Specifically: ProportionAccess = Sum(BufferPop17)/Sum(BlockGroupPop17). Save these proportions as an excel or text file: transit\_access.csv.

The next steps create mappable points for each person in the geography. This is used for the Power BI map. In the code, these steps are actually before step 14.

1. Use the spsample command on the spatial dataset to generate a random set of points within each polygon, where the size of each set is equal to the population in each polygon. Call this dataset bap\_coords.
2. Create a new data frame called bap\_exp (blocks-as-points expanded) with one row for each person in the geography by merging bap\_coords with tcomb.
3. For each subpopulation type, randomly select rows within each block to generate a column of subpopulation individuals. Merge the column with bap\_exp to get a very tall (i.e. many rows) data frame with a column for total population and each subpopulation. Use all.x to include all of the population rows; the subpopulation columns will have NA when the row was not selected.
4. Referring to the population ranks in Population\_table, use the following steps to iteratively assign a population rank to each row:
   1. Create a column pop\_rank = 0.
   2. Starting with the largest subpopulation (e.g. rank = 14, total population), if the total population row = 1, make pop\_rank = 14.
   3. With the next largest subpopulation (e.g. rank = 13, white), if the white row = 1, make pop\_rank = 13. This will overwrite some of the total population ranks, but that is okay.
   4. Repeat this all the down to rank = 1.
   5. This will assign a population rank to each row, where the rank is 14 if the row is part of the total population and no other subpopulation, 13 if it is part of the total population and white population and no other subpopulation, …, 1 if is part of every population.
5. Randomly sample the data frame to reduce the size. Choose the sample size to be manageable by Power BI yet still show enough data to interpret. 5000 rows is adequate, 10,000 may be reasonable for large populations.
6. Add columns that flags access to each route type (all routes, local routes, etc.).
7. Duplicate the data frame so that access by route type is listed vertically in one column called Access instead of across many columns. If the old data frame had 5000 rows and 6 route types, the new data frame will have 5000\*6 = 30000 rows. Save the data frame as bap\_transit.csv.

Output from R processing steps:

* Transit\_access.csv. This table includes the proportion of populations that live within 1/4 mile of a transit stop, stratified across three dimensions:
  + Geography (multiple counties)
  + Subpopulation (total population, children, seniors, etc.)
  + Route type (all routes, local routes, etc.)
* Bap\_transit.csv. This table includes individual level mappable points with columns that identify the following columns:
  + Latitude and Longitude
  + Access to route type flagged as 1 or 0. Each route type is a replicate of the full data frame.
  + Subpopulation type displayed as pop\_rank. If pop\_rank is X, then the person belongs to all populations with rank X or lower.
* Population\_table.csv. This table has the population ranks associated with the different subpopulations and will be used to filter the map in Power Bi.

Step 3: Power BI

These steps visualize the data in an interactive form. Most of the work can be done by someone familiar with Power BI graphs and slicers, so those details are skipped. The creation of the map visual is explained in more detail. The slicers I use are a custom visual called Chiclet Slicer.

Files used in Power BI visuals:

* Transit\_access.csv
* CityList.xlsx
* Bap\_transit.csv
* Population\_table.csv

Steps:

1. Load the four tables into Power BI. Transform columns if desired to change value names.
2. For the Compare Counties, Compare Populations, and Compare Route types pages, use Transit\_access and CityList to create the graphs and slicers.
3. To map the data, first create a new measure called FilterMeasure using the following DAX code:

FilterMeasure =

VAR ranks = max(Population\_table[pop\_rank])

RETURN(IF(COUNTROWS(FILTER(bap\_transit,bap\_transit[bap\_rank]<=ranks))>0,1,0))

This measure will filter the bap\_transit data to rows where the population rank is less than or equal to the rank of the population selected in the population slicer.

1. Create a population slicer using Population\_table[pop\_rank]. Make sure the slicer does not allow multiple selections.
2. Add the MapBox custom visual to Power BI. Add a MapBox Visual to the page. Drag the Bap\_transit Latitude and Longitude into their corresponding fields and Access to the color field. Follow the instructions to get an access token.
3. Add Bap\_transit[FilterMeasure] as visual level filter to the MapBox visual. Set FilterMeasure = 1 as the condition. Now when you select different subpopulations the number of dots will reflect the proportionate number of people.

That’s it!

Any further questions email

[peter.banwarth@co.benton.or.us](mailto:peter.banwarth@co.benton.or.us)

Appendix: List of ACS tables. Many of these tables require preprocessing to get the correct population numbers.

|  |
| --- |
| B01001 |
| B03002 |
| B08301 |
| C17002 |
| B22010 |
| B25003 |
| B25010 |
| B25070 |
| B25091 |
| B27010 |