

Lab 8: Importing, Joining Tables

What You'll Learn: This lesson introduces importing text files into an ArcMap table, combining rows, and navigating tricky joins. The work is organized as two small projects, the first with step-by-step instructions, the second less so, as most operations have been introduced previously, and we expect you to be familiar with them by now. This project requires synthesis of what you've learned in this and previous labs.

Data are in the L8\ directory, with the \Project1 subdirectory containing a *lwr48.shp* file and a *states.shp* file, both in NAD83 geographic coordinates, and the \Project2 subdirectory containing a *Cal.shp* file in WGS84 geographic coordinates.

What You'll Produce: Three maps, one of U.S. NASS data, one of California county income, and one of California counties with parks or forests.

Project 1

This project introduces something quite common, joining ASCII tabular data with a shapefile. Here, we will combine a text file on corn production for US counties with a county shapefile, but there are many types of tabular data that are available as text files, summarized on a county basis, including population, voting, education, income, crime, air pollution, and many other social, political, and environmental data.

Here we import a text file, convert it to a ArcMap compatible table, and edit the table, deleting columns, creating join items, and combining rows before joining it with a polygon shape file. These are all common operations when ingesting tabular data.

Start ArcMap, and add *lwr48.shp* from the L8\Project1\ subdirectory.

Now add the text file *cnty26.csv* to this data view, and open the table for viewing.

(Video: L8_1_add_texttable.mov)

This file contains 1996 seed corn production, in bushels, for counties in the United States. These data were downloaded from the National Agricultural Statistical Service website, www.nass.usda.gov/, and we're most interested in the columns:

Stfips: the state Federal Information Processing System (FIPS) code

CoFips: county FIPS code

Harvested: the acres harvested for a given yield category in a county

Yield: Bushels per acre harvested for the yield category

Production: Total bushels produced (yield times harvested) for the given yield level.

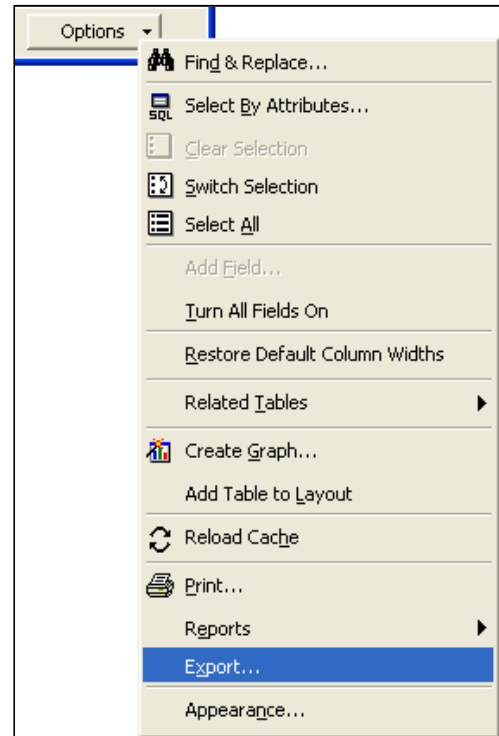
Unfortunately, we can't directly edit the .csv file, so we must convert it to a dbf file.

In the Attributes of *cnty26.csv* window, left click on the **Options** button in the lower-right portion of the table window frame, then **Export** in the dropdown menu, and save all records to the Project1 subdirectory, naming it something like “*raw_corn_dat*”.

Remove the *cnty26.csv* from the data frame, to reduce clutter, and add and open *raw_corn_dat.dbf* table in the data frame.

Delete all the columns except the following: OID, Stfips, CoFips, Harvested, Yield, Production.

Delete columns by right clicking in the column heading, and selecting **Delete** near the bottom of the dropdown menu. Left click on the Yes button in response to the warning about deleting outside of an edit session.



We now want to join these data with the county shapefile, *lwr48.shp*. Unfortunately, there are two problems. First, we don’t have a ready-made key for the join. There is no column that maps cleanly from the *raw_corn_dat.dbf* file to the *lwr48.dbf* file.

Let’s first fix this problem (**Video: L8_2_create_index.mov**).

Open the data table for the *lwr48* shapefile.

Notice that *lwr48.shp* also has the county and state FIPS codes, in the COUNTY and STATE columns, respectively. Each state has a unique FIPS code, and each county within a STATE has a unique code. If we combine the STATE-COUNTY codes, we can create a unique ID for each county in the country.

Add a column to the *lwr48* data table (**Options > Add Field** in the “Attributes of *lwr48*” window).

Make this field a long integer, with at least 8 columns (precision), and name it something like *sta_count*.

Use the field calculator to assign *sta_count* a value according to the formula:

$[STATE] * 10000 + [COUNTY]$

Multiplying the STATE by 10000 and adding to COUNTY creates a unique 5-digit code, with the value for STATE in the first two digits, and the value for COUNTY in the next three digits.

Open the raw_corn_dat.dbf file, add a sta_count column similar to the one in lwr48, and create and value for a new column using the field calculator, according to:

$[Stfips] * 10000 + [CoFips]$

Now, sort the raw_corn_dat table in ascending order, by right clicking/selecting in the sta_count column.

You should have a window that looks something like the figure at right:

We can now see the second problem with this data set.

Note that there are multiply entries for sta_count, each state/county combination. This is because yield was reported at various levels for each county.

We must aggregate the rows before we join this table to the lwr48 shapefile. A join matches the rows by a key. If we don't somehow summarize the multiple rows that have the same sta_count value, then we can't be sure which will be chosen for the join – many to one joins are ambiguous.

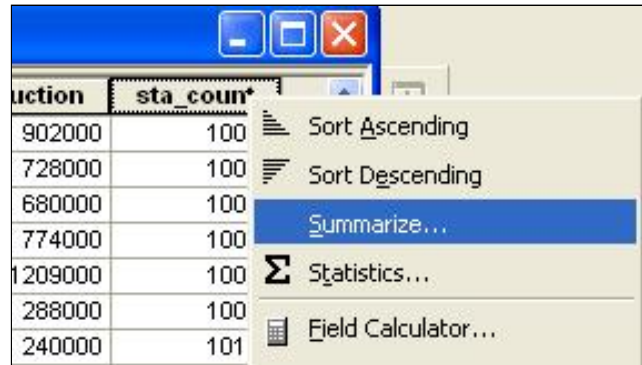
Fortunately, ArcMap provides a tool for aggregating rows.

(Video: L8_3_summarize_rows.mov)

OID	Stfips	CoFips	Harvested	Yield	Production	sta_count
28	1	1	2000	38	76000	10001
2649	1	1	1200	75	90000	10001
5250	1	1	900	33	30000	10001
7843	1	1	1600	76	121000	10001
38	1	3	11600	102	1180000	10003
2662	1	3	9200	99	914000	10003
5262	1	3	4600	68	315000	10003
7854	1	3	5300	112	593000	10003
48	1	5	4300	54	232000	10005
2672	1	5	3400	88	300000	10005
5272	1	5	1000	70	70000	10005
7864	1	5	2600	110	286000	10005
9	1	9	3400	94	320000	10009
2630	1	9	1800	94	169000	10009
5230	1	9	1800	83	150000	10009
7824	1	9	1800	119	214000	10009
2650	1	11	500	60	30000	10011
39	1	13	2600	68	177000	10013
2663	1	13	2900	77	223000	10013

Right click on the column heading for `sta_count` in the `raw_corn_dat.dbf` table.

Left click on the **Summarize** option in the dropdown menu:



In the resultant Summarize window, verify that the `sta_count` field is displayed in subwindow 1. *Select a field to summarize*

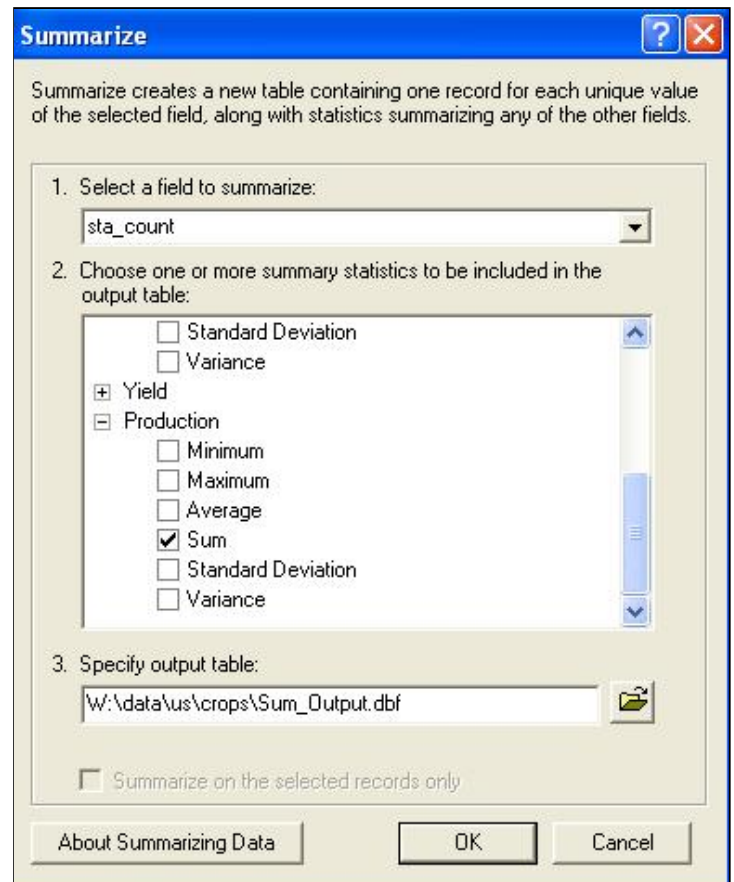
In subwindow 2. *Choose one or more....*, you will see a list of available items. Left clicking on the +/- to the left of each item displays a list of summary statistics you may request.

Request the Sum summary statistic for the Harvested and Production variables.

Specify the `L8\Project1\` subdirectory as an output location, with a filename something like “`Sum_crops`”.

Left click OK, and when asked, add the data to the view.

Open the output `Sum_crops` file, sort ascending by `sta_count`, and verify that `sta_count` with a value of 10001 has a `Sum_Harvested` of 5700, and a `Sum_Production` of 317000.



Now, join the summary table you just created to the `lwr48.shp` file (Right click on `lwr48.shp` in TOC > **joins and relates** > **Join**, than select appropriate columns in the shapefile and the summary table)

We want to further process this combined data. Because many operations are restricted on joined files, it is best to save a copy to a new file, so

Right click on the *lw48.shp* in the TOC, then left click **Data > Export Data**, and name and save the file appropriately, something like *US_corn*.

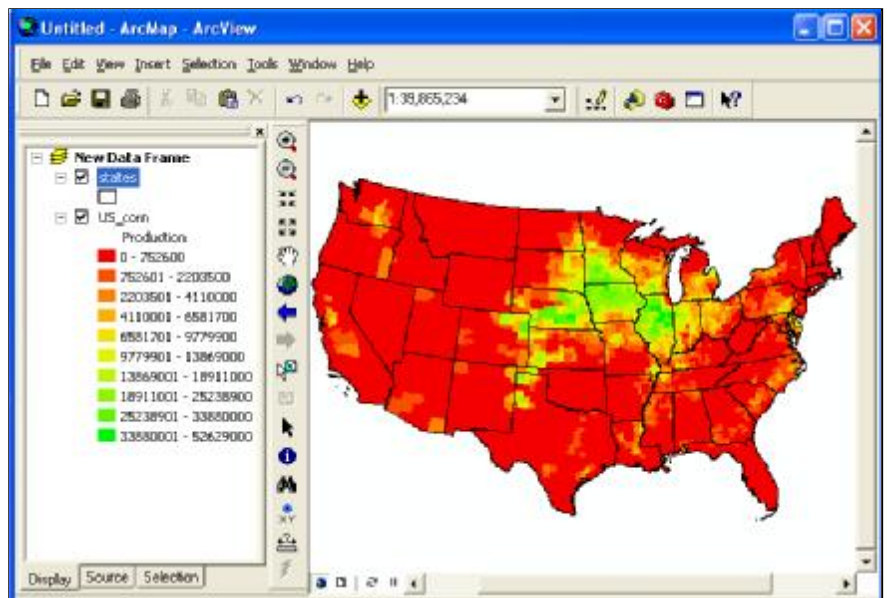
Add this new file to your data frame, and remove the *lwr48* file and summary production tables.

Now, prepare your data for output. First, change the data frame projection to something more appropriate for viewing by right clicking on the data frame name in the TOC, then **Predefined > Projected > Continental > North America > USA_Contiguous_Albers_Equal_Area_Conic**.

Symbolize the *US_corn* file as Quantities, Graduated colors, with a gradient color ramp between two distinct colors (below from red to green), specifying about 10 classes in a Natural Breaks scheme.

Display the states shapefile with a single symbol, hollow symbolization. This should result in a data view similar to that shown to the right.

Create an appropriately annotated layout, with title, north arrow, name, legend, and other descriptive elements, and produce a pdf of the layout.

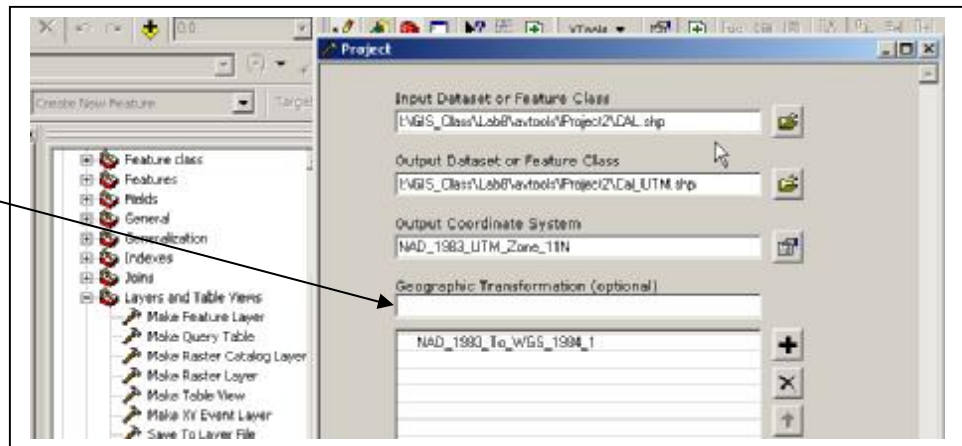


Project 2

Your second task is to produce two (2) maps on the same layout, one showing average income by county in California, and a second map showing counties with state parks or forests. These instructions will be less detailed than previous lessons, as we omit most steps we've covered in previous lessons. The goal is for you to synthesize these previously-taught tools on your own.

Maps must be produced in a UTM Zone 11, NAD83 projection. The original data are with map units of decimal degrees. Use ArcToolbox to reproject the data, or set the data frame properties to the UTM coordinate system.

Hint: When you reproject to UTM – NAD83—Zone 11 you will be asked for a Transformation, just accept the first on the list NAD_1983 to WGS_1984_1



- Data for this project are in L8\project2\ subdirectory.
- California county boundaries are in *Cal.shp*.
- Income.dbf is a database file that lists average per capita income by county.
- Rec.dbf is a database file that list recreation features an properties, including county location

The income file, recreation file and the county file have a common attribute – cnty_name. This common field allows you to join these files together as needed.

Income Map

You need to produce a map showing only those counties with an average per capita income greater than \$16,000. Create this map in a new data frame.

[Hint: – use a binary indicator attribute. Create a new attribute that either has the value 0 (counties below 16,000 per capita income) or the value 1 (counties above 16,000)].

Park/Forest Map

Create this map in a new data frame. It should be a map of counties that contain a Parks **or** Forest. The database file named *rec.dbf* lists many recreation types, not just parks and forest. You may use this database, selecting as appropriate to create a binary indicator for those that meet the condition above. This attribute should indicate if either park or forest recreational areas are present.

You then need to combine the *rec.dbf* file to your county coverage, and create a map displaying the counties that have Park or Forest Lands. Note that there is a **many-to-one** relationship for the counties in *rec.dbf* with counties in the *Cal.shp* file. There may be multiple entries including for each county, one for each park, forest, reservoir, or other features found in a county. You need to develop a list of counties with parks or forests from this *rec.dbf* (**Video: L8_4_join_warning1.mov**).

One way is to open the *rec.dbf* table and select all those with parks, and save your selected records to a Parks table. Repeat this process, selecting and saving only the records with forests to a Forests table.

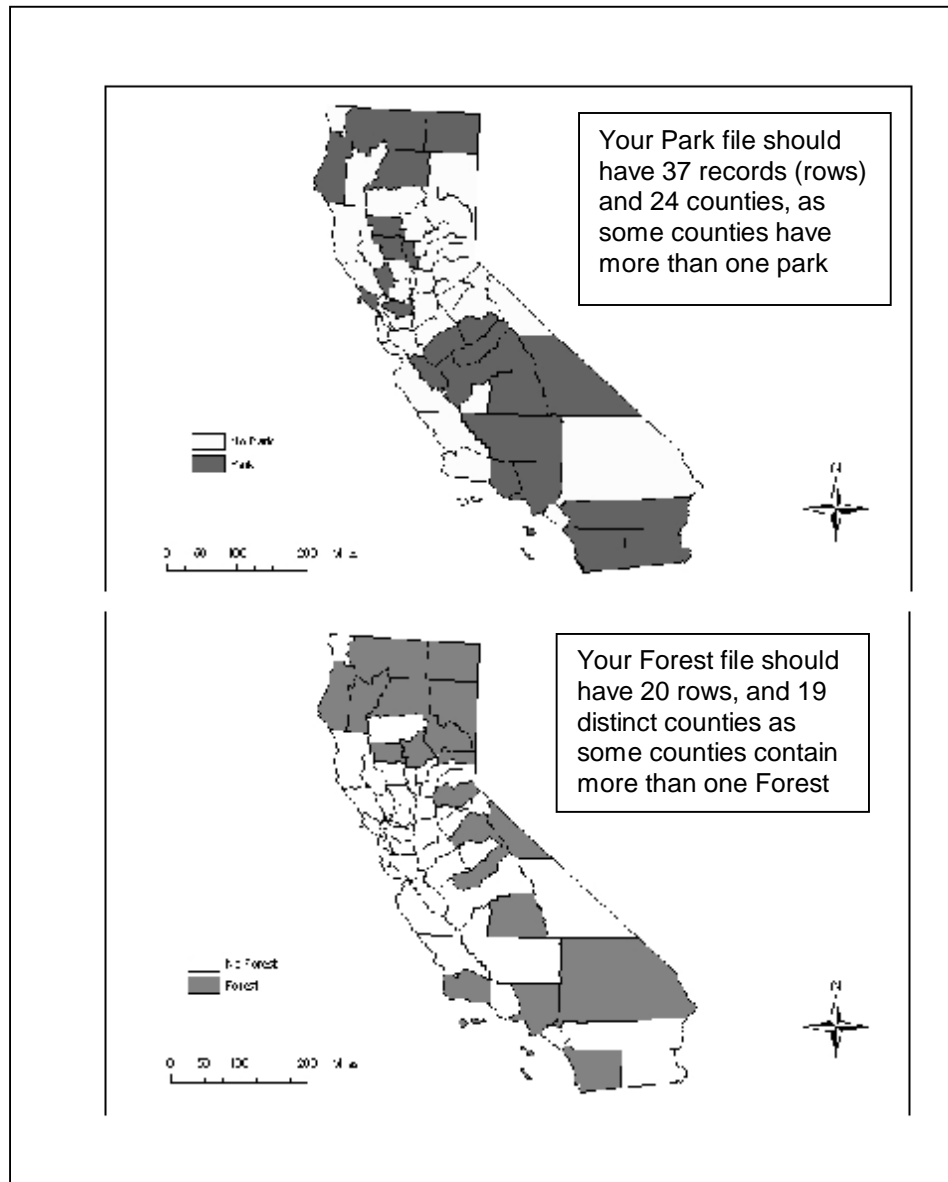
Then create indicator variables in both tables, for example, create a new field named something like “has_forest” in the forest table, and set this value equal to one for each record in the table, and create a similar “has_park” column of ones in the parks table.

Now join the parks and forests tables to *Cal.shp*, and then select those rows with a park or a forest. There is a map of the correct set near the end of these instructions.

Another, bit trickier option is to create both the parks table and the forest table, as above, but instead of adding the 0/1 binary variable to each, do a serial join. First join the parks.dbf you created with the list of counties that have parks to the *Cal.shp* table, then join the forests.dbf table to the *Cal.shp* table (with parks still joined). Then select the records that have either a park or forest in the respective columns, and assign these an indicator variable that you’ll use to symbolize your map (**Video: L8_5_join_warning2.mov**).

A note of caution; this is a tricky exercise, and many students do not produce this final map correctly. The main problem comes from multiple entries (parks or forests) for each county. You need to be very careful in the table joins, and look at the maps you produce. Make sure your final product makes sense. One helpful guide may be the flowchart or the maps of the respective component maps; in this case a map of those counties with forests, and a separate map of those counties with parks. The “OR” condition should include data from both joined files, so your final project 2 map should have both colored in.

The figure below as a guide to ensure you've identified counties correctly, but these are not guides for your final layouts – this is just to illustrate some characteristics of the selection. The figure after this one is an example of what your final map might look like.



Remember to both the income and park/forest maps on the same layout (see example below). Each map should be in its own data frame, and both using the UTM Zone 11 projection

Include the following elements in your maps:

A descriptive title (not Lesson 8. What does your map contain?)

A descriptive legends. The legend should not list file names, for example, cal.shp.

Scale bar and north arrow- use the same scale for each data frame and only 1 north arrow.

