32.23123° N / -110.94909°W

UA Libraries Data Cooperative Unit's

GIS TUTORIALS

DATA WRANGLING AND CREATING NEW GEOSPATIAL DATA

NONE

SOFTWARE USED

2

TUTORIAL NUMBER

★★★☆

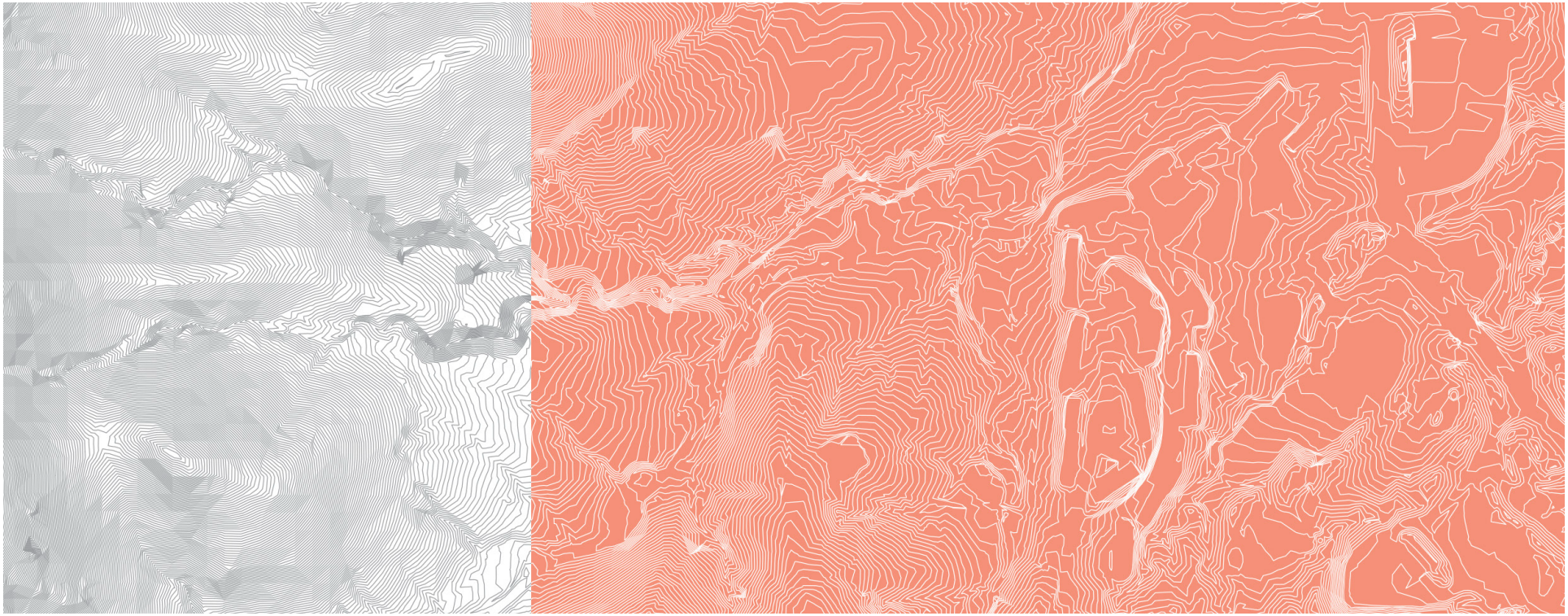
DIFFICULTY LEVEL

★★★★

LEVEL OF STOKE

HARDWARE NEEDED:	SOFTWARE NEEDED:
desktop or laptop computer	R
internet connection	RStudio

UNIVERSITY LIBRARIES
Data Cooperative



INTRODUCTION

2

The purpose of this tutorial is to teach you how to do basic data wrangling tasks in R. Oftentimes when you download secondary (data that is collected by some-one else) GIS data it may be larger than the study area that you are interested in, or contain observations that may not be useful to the analysis that you are going to undertake. Most, if not all, GIS projects contain some form of data wrangling tasks that will provide you with geospatial data that is more relevant to your intended project.

Please note: This tutorial is a continuation of the previous tutorial, please refer to this tutorial in order to follow this tutorial in its entirety.

Upon completion of this tutorial, you should be comfortable:

1. Use the data table to create a subset of geospatial data based on different attribute subsetting options.
2. Use geographic locations to create a subset of data based on different location subsetting options.

REOPENING A PREVIOUS PROJECT

1. Open and load the packages you used in the previous tutorial.

Install and load the `dplyr` package.

Run all of your previous code to load all the sf objects previously created:
`west_dams_sf` `az_tributaries_sf` `az_basins_sf`

SCRIPT EDITOR

```
## load needed packages

library(sf)
library(ggplot2)
library(mapview)

## install and load the dplyr package

install.packages("dplyr")
library(dplyr)
```

PLOT

```
none
```

DPLYR PACKAGE:
The dplyr package is a grammar of data manipulation that uses a consistent and intuitive set of "verbs" of functions to help with data manipulation tasks.

ATTRIBUTE SUBSETTING AND DATA CREATION

1. Open the `west_dams_sf` object's attribute table.

When you use the `View()` function the object's attribute table (`tidyverse data.frame`) will automatically open in the Data Viewer tab next to your Script Editor pane. You can explore this data table as you would in other table-based software programs (e.g. Microsoft Excel or Apple Numbers).

SCRIPT EDITOR

```
## explore the west dams attribute table

View(west_dams_sf)
```

PLOT

```
none
```

2. Select only the dams that are found in the state of Arizona and create a new sf object.

When using the `filter()` function you are asking R to look for only those rows that contain the value of AZ in the field STATE. When exploring the attribute table you could see that there was a STATE field and that if you wanted to select only the dams in Arizona you could use the state's abbreviation since those are the values found in the STATE field.

SCRIPT EDITOR

```
## create an Arizona dams sf object

az_dams_sf <- filter(west_dams_sf, STATE == "AZ")
```

PLOT

```
none
```

ATTRIBUTE TABLES:
Attribute tables are the non-spatial information contained within geospatial features and consist of the following characteristics:

Tables that contains rows.

All rows in the table have the same fields (variables).

Each column has a data type.

In an sf object the attribute table is a tidyverse compatible data.frame.

filter:
The filter function selects only the rows from within your data that meet the argument that you supply it. It uses the values of the rows to select them.

ATTRIBUTE SUBSETTING:
In most geospatial projects you will use datasets that contain more features (rows) that you will need for your analysis. In R using the attribute table to create a new object is referred to as attribute subsetting.

3. Check to ensure that your new sf object only contains dams found in Arizona.

Check the console after running the `count()` function to ensure only AZ appears in the STATE column.

```
* <chr>                <int>       <MULTIPOINT [m]>
1 AZ                    310 ((172934.4 3637111), (174687.7 3737943...
```

You can also plot the sf object on an interactive map

SCRIPT EDITOR

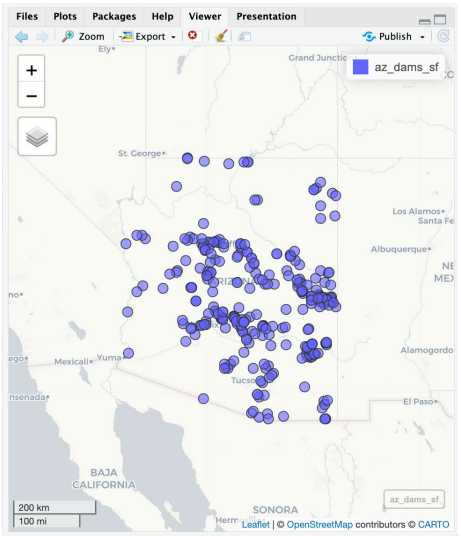
```
## check to make sure only STATE = AZ were selected

az_dams_sf %>%
   count(az_dams_sf$STATE)

## check arizona dams on an interactive map

mapview(az_dams_sf)
```

PLOT



count:
The count function allows you to quickly check the number of unique values that are present in one or more variables (fields).

HELPFUL HINT:
One of the advantages of writing your R code in a script is that you can copy and paste code that you have already written and reuse it for tasks that you have completed on other data-sets. The key to copying and pasting code from your own script or from other sources to use in your own project is understanding what goes where and how the data you are workign with can fit within the code block(s).

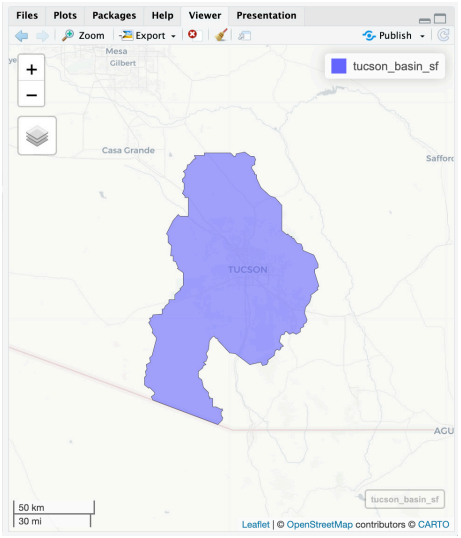
4. Repeat the previous steps to create a `tucson_basin_sf` object from the `az_basins_sf` object.

Hint: Explore your attribute table before you start writing your code so that you can find the field (column) and value within that field that you will use in your `filter()` function.

SCRIPT EDITOR

```
## on your own
```

PLOT



SPATIAL SUBSETTING AND DATA CREATION

1. Use your `az_dams_sf` and `tucson_basin_sf` objects to select the dams that are located within the Tucson water basin.

When using the command `x[y, ]` you are subsetting the features from `az_dams_sf` using the source object `tucson_basin_sf`. This is a spatial subset because both objects used in the command are both geographic sf objects. The `op` argument allows you to select the topological relation that you want to use for the spatial subsetting and you are looking for dams within the Tucson basin.

SCRIPT EDITOR

```
## create a Tucson dams sf object

tucson_dams_sf <- az_dams_sf[tucson_basin_sf, op = st_within]
```

PLOT

```
none
```

2. Check to ensure that you have only selected dams found within the Tucson water basin.

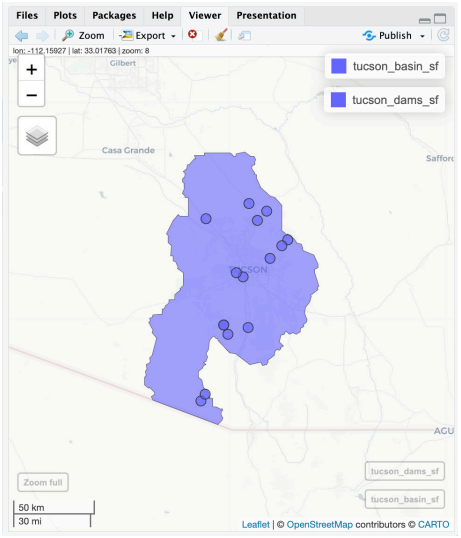
As with the previous tutorial, you can add multiple layers to interactive `mapview()` plots using the `+` operator.

SCRIPT EDITOR

```
## check tucson dams on an interactive map

mapview(tucson_basin_sf) +
   mapview(tucson_dams_sf)
```

PLOT



SPATIAL SUBSETTING:
In most geospatial projects you will use datasets that contain data that is outside of your study area or area of interest. In R using an original object and returning a new object containing only the features that are within another object is called spatial subsetting.

st_within:
The st_within argument determines whether one geometric object is completely within another geometric object.

3. Repeat the previous steps to create a `tucson_tributaries_sf` object selecting all tributaries in the Tucson water basin from the `az_tributaries_basin_sf` object using the `tucson_basin_sf` as the source object.

SCRIPT EDITOR

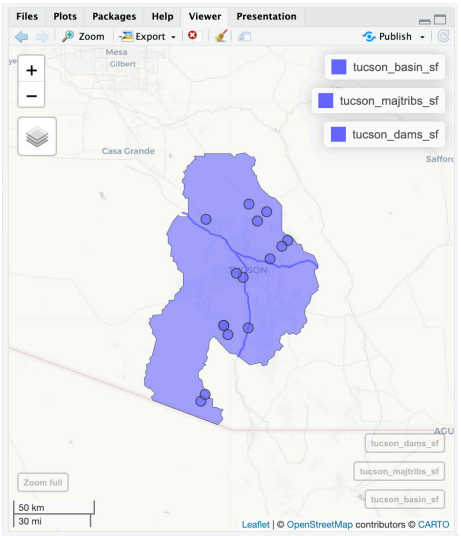
```
## create a Tucson tributaries sf object

tucson_tributaries_sf <- az_tributaries_sf[tucson_basin_sf,
                                           op = st_intersects]

## check tucson tributaries on an interactive map

mapview(tucson_basin_sf) +
   mapview(tucson_tributaries_sf) +
   mapview(tucson_dams_sf)
```

PLOT



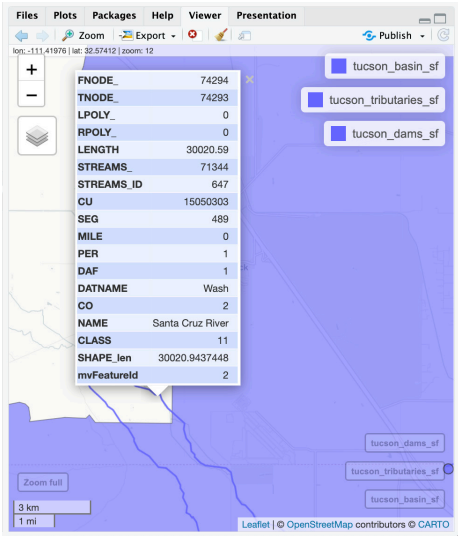
st_intersects:
The st_Intersects argument is used to determine whether two geometries intersect with one another.

4. In the interactive map in the Viewer panel zoom into the area where the the Tucson tributaries extends past the Tucson basin boundary. If you click on one of the tributaries you will notice that this is one of the segments of the Santa Cruz River that extends beyonds the Tucson basin boundary because you used `st_intersect` as your topological geographic relationship. So this segment of the Santa Cruz River intersects with the boundary. You can also view the `data.frame` of the `tucson_tributaries_sf` object to see that there are multiple segments for each of the tributaries in the dataset. If you used `st_within` this segment would have not been included since it is not completely within the basin boundary. You can leave this as is.

SCRIPT EDITOR

```
## none
```

PLOT



MULTIPLE ATTRIBUTE SUBSETTING AND DATA CREATION

1. Create an sf object only containing the Santa Cruz River, Tanque Verde Creek, and Rillito Creek.

The `%in%` operator is used to determine if specific values are present in a field, so you are asking R to look for the three different water courses in the `NAME` field. If these values are TRUE R then will use the `subset()` function to create a new sf object that contains only the TRUE results.

SCRIPT EDITOR

```
## create a Tucson major tributaries sf object

tucson_majtribs_sf <- subset(tucson_tributaries_sf,
   subset = NAME %in% c("Santa Cruz River",
                        "Tanque Verde Creek",
                        "Rillito Creek"))
```

PLOT

none

subset:
The subset function gets the rows and columns from a larger data.frame that meet certain criteria outlined in an argument.

2. Check to ensure that you have only selected dams found within the Tucson water basin.

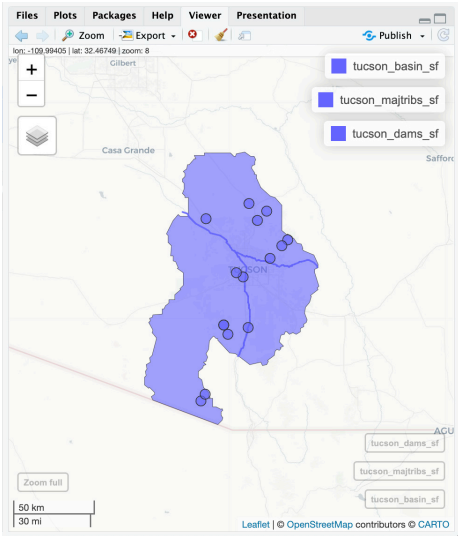
As with the previous tutorial, you can add multiple layers to interactive `mapview()` plots using the `+` operator.

SCRIPT EDITOR

```
## cheeck tuscon major tributaries on an interactive map

mapview(tucson_basin_sf) +
   mapview(tucson_majtribs_sf) +
   mapview(tucson_dams_sf)
```

PLOT



END