32.23123° N / -110.94909°W

UA Libraries Data Cooperative Unit's

GIS TUTORIALS

CREATING BASIC MAPS

R

SOFTWARE USED

3

TUTORIAL NUMBER

★★★☆

DIFFICULTY LEVEL

★★★★

LEVEL OF STOKE

HARDWARE NEEDED:

desktop or laptop computer

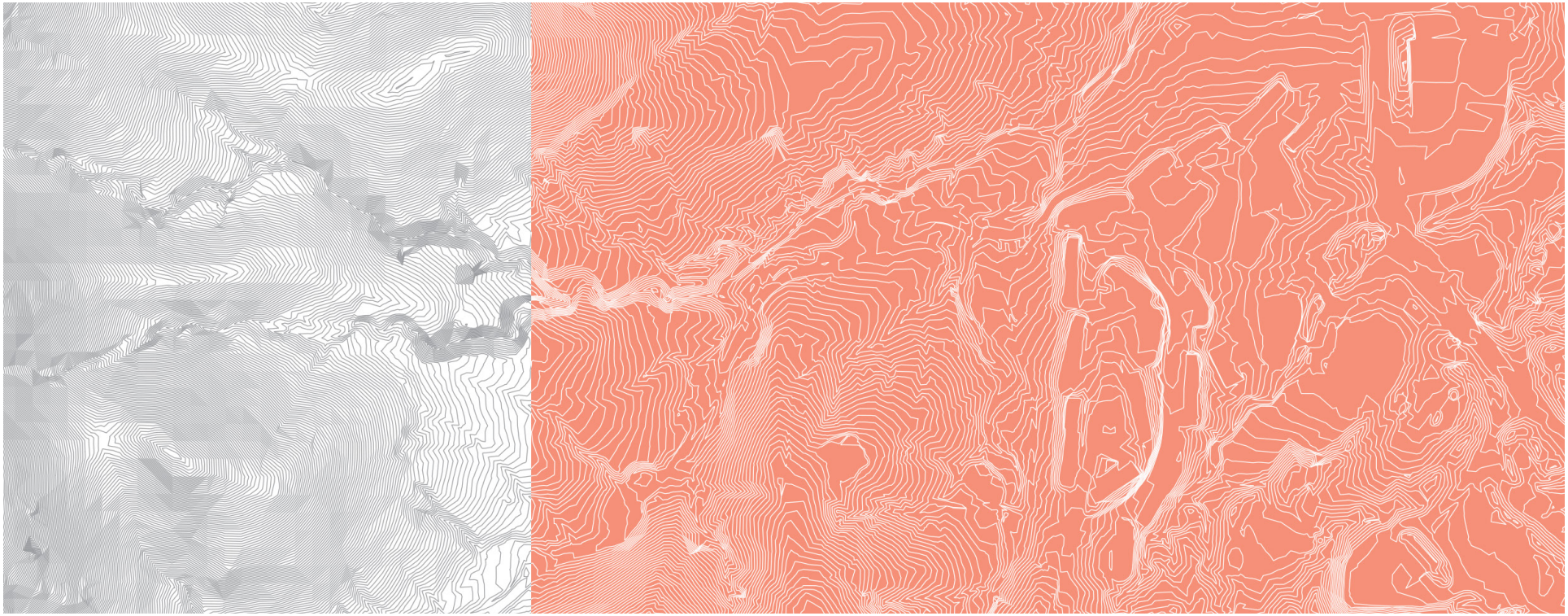
internet connection

SOFTWARE NEEDED:

R

RStudio

UNIVERSITY LIBRARIES
Data Cooperative



INTRODUCTION

3

The purpose of this tutorial is to teach you how to create a basic map in R that contains the necessary cartographic elements. Generally the type of map layout that you will create will be based on where your final map will be used (e.g. publication, in a report, as a stand alone layout, etc...) and this tutorial will serve as a basic guide to get you familiar with creating maps in a general context.

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

Upon completion of this tutorial, you should be comfortable:

1. Changing aesthetics of sf objects in a plot.
2. Designing a map layout with all of the proper cartographic elements.
3. Gaining an understanding of all the cartographic elements that are necessary for map layout.

REOPENING A PREVIOUS PROJECT

1. Open the project you created in the previous tutorial.

Load the packages you used in the previous tutorial.

Install and load the `tmap` and `RColorBrewer` packages.

Run all of your previous code to load all the sf objects previously created:
`tucson_dams_sf` `tucson_majtribs_sf` `tucson_basin_sf`

SCRIPT EDITOR

```
## load needed packages

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

## install and load the tmap package

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

PLOT

```
none
```

TMAP PACKAGE:
The tmaps packaage is used to create layer-based thematics maps using the grammar of graphics like the syntax used in the ggplot2 package. It is a dedicated map making package and allows more direct control of cartographic elements than other R packages.

PREPARING TO MAP

1. Create and plot a map of the `tucson_basin_sf` object using the `tmap` package.

When using the `tmap` package there are two arguments that are needed to plot an sf object on a map known as the map layer and the visual variable. The `tm_shape()` function declare the sf object as the map layer and the `tm_polygon()` function assigns the visual variables including the `fill` (color fill) and `lwd` (line width) arguments.

You can save plots as objects, but need to run the plot for it to appear in the Plots panel.

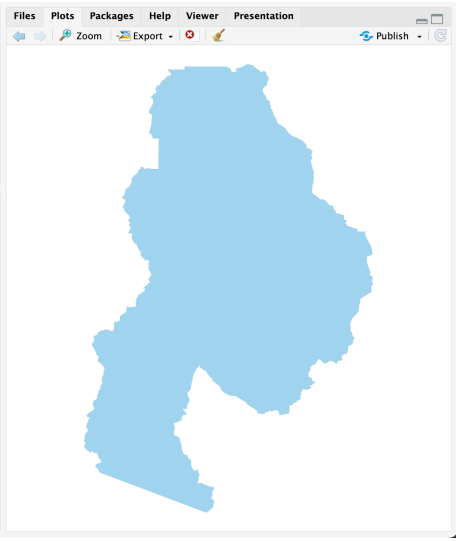
SCRIPT EDITOR

```
## create a map of the tucson basin

basin_map <- tm_shape(tucson_basin_sf) +
   tm_polygons(fill = "lightskyblue2", lwd = 0) +
   tm_layout(frame = FALSE)

basin_map
```

PLOT



2. Create and plot a map of the `tucson_majtribs_sf` object using the `tmap` package.

When using the `tmap` package there are two arguments that are needed to plot an sf object on a map known as the map layer and the visual variable. The `tm_shape()` function declare the sf object as the map layer and the `tm_lines()` function assigns the visual variables including the `col` (outline color) and `lwd` (line width) arguments.

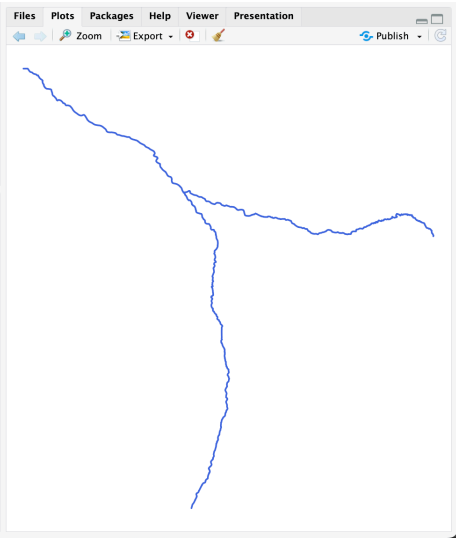
SCRIPT EDITOR

```
## create a map of the tucson major tributaries

majtribs_map <- tm_shape(tucson_majtribs_sf) +
   tm_lines(col = "royalblue", lwd = 2) +
   tm_layout(frame = FALSE)

majtribs_map
```

PLOT



TMAP PACKAGE:
When you are plotting multiple sf objects with different geometry types on a map the order of how the objects should be polygons (bottom), lines (middle), and points (top). The layers will be plotted in the order that they appear in the code.

COLOR CHOICES IN R:
Base R (with no packages loaded) comes l oaded with around 657 color choices for you to choose from when you are making maps. There are many websites that have a visual of these colors. You can also user packages such as RColorBrewer that expands the color choices you have.

HELPFUL HINT:
There are many arguments that you can use to change the visuals of each map layer and these different arguments are available on various online websites.

3. Change the tucson_dams_sf HAZARD variable (field) from character to factor and relevel the factors in the correct order of dam hazard risk levels.

```
[1] "LOW"          "HIGH"         "SIGNIFICANT"
```

Character variables, also known as strings, are used to store textual data while factors are used to store categorical (group) data. While character variables can take on innumerable values, factor variables are constrained to a set or fixed number of known values (in this case the dam hazard risk leevl).

SCRIPT EDITOR

```
## turn HAZARD variable to factor and check level order

tucson_dams_sf$HAZARD <- factor(tucson_dams_sf$HAZARD,
                                levels = c("LOW", "HIGH", "SIGNIFICANT"))

## check the level order of the HAZARD variable

levels(tucson_dams_sf$HAZARD)
```

PLOT

```
none
```

TMAP PACKAGE:
When you are plotting multiple sf objects with different geometry types on a map the order of how the objects should be polygons (bottom), lines (middle), and points (top). The layers will be plotted in the order that they appear in the code.

4. Create and plot a map of the `tucson_dams_sf` object using the `tmap` package.

When using the `tmap` package there are two arguments that are needed to plot an sf object on a map known as the map layer and the visual variable. The `tm_shape()` function declare the sf object as the map layer and the `tm_dots()` function assigns the visual variables including the `fill` (the field used to fill the layer), `fill.scale` (the colors to use for the factors), and `size` (dot size) arguments.

Also, because the map is using the HAZARD field a legend is automatically created.

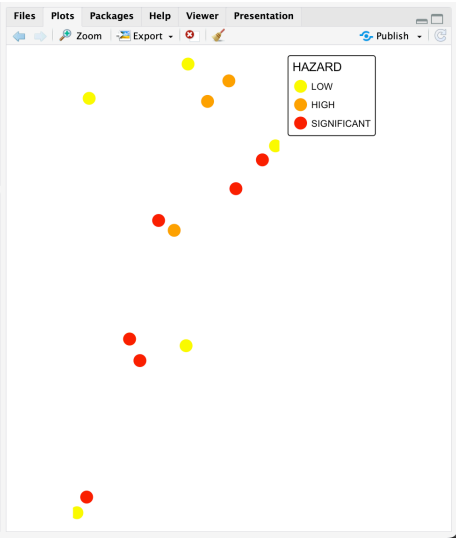
SCRIPT EDITOR

```
## create a map of the tucson basin dams

dams_map <- tm_shape(tucson_dams_sf) +
   tm_dots(fill = "HAZARD",
   fill.scale = tm_scale_categorical(values =
               c("yellow", "orange", "red")), size = 1) +
   tm_layout(frame = FALSE)

dams_map
```

PLOT



5. Combine the three maps to create one map.

If you want to make visual adjustments to the layers after combining the three map objects into one map you can make those in the individual map code, rerun the code, and then run the code below again.

The size of the dam points were changed in the plot to the right.

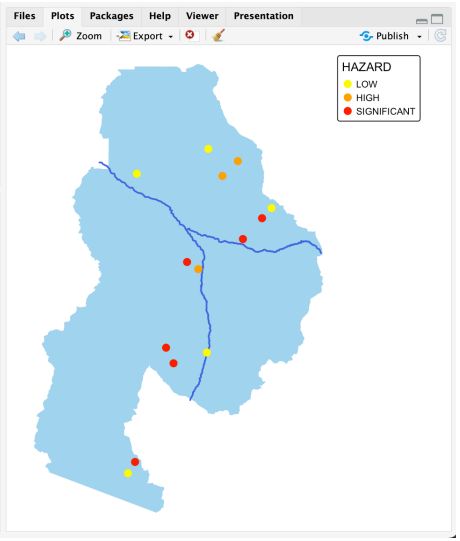
SCRIPT EDITOR

```
## create a tucson map from the three individual maps

tucson_map <- basin_map +
   majtribs_map +
   dams_map

tucson_map
```

PLOT



TMAP PACKAGE:
When you are plotting multiple sf objects with different geometry types on a map the order of how the objects should be polygons (bottom), lines (middle), and points (top). The layers will be plotted in the order that they appear in the code.

COLOR CHOICES IN R:
Base R (with no packages loaded) comes loaded with around 657 color choices for you to choose from when you are making maps. There are many websites that have a visual of these colors. You can also user packages such as RColorBrewer that expands the color choices you have.

HELPFUL HINT:
There are many arguments that you can use to change the visuals of each map layer and these different arguments are available on various online websites. This tutorial contains only basic examples of arguments you can make for different visualization options.

CARTOGRAPHIC ELEMENTS

1. Create a legend for the dams layer in a new map.

Within the `tm_dots` visualization arguments you can control the legend using the `fill.legend` and `tm_legend` arguments. Within `tm_legend` you can define a number of legend options including `title` (changes the default title), `position` (changes the default location of the legend), and `frame` (removes the legend frame).

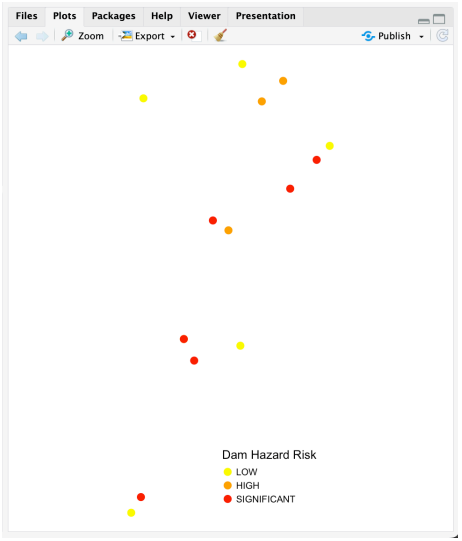
SCRIPT EDITOR

```
## create a dam map with a designed legend

dams_map_lgd <- tm_shape(tucson_dams_sf) +
   tm_dots(fill = "HAZARD", size = 0.6,
   fill.scale = tm_scale_categorical(values = c("yellow", "orange", "red")),
   fill.legend = tm_legend(title = "Dam Hazard Risk",
                           position = tm_pos_in("right", "bottom"),
                           frame = FALSE)) +
   tm_layout(frame = FALSE)

dams_map_lgd
```

PLOT



2. Replace the newly created `dams_map_lgd` map for the `dams_map` in your combined map code created in previous steps to create a new `tucson_map` map with a designed legend.

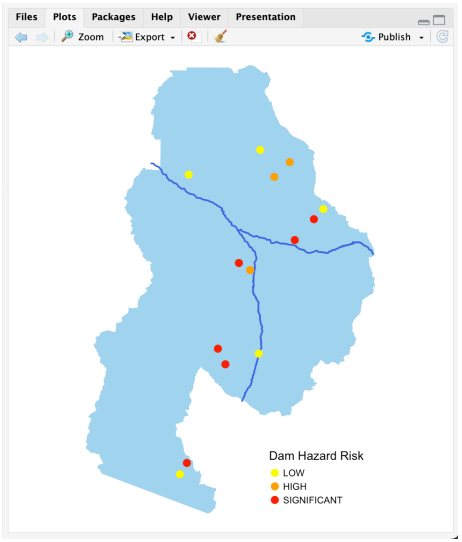
SCRIPT EDITOR

```
## create a tucson map from the three individual maps

tucson_map <- basin_map +
   majtribs_map +
   dams_map_lgd +
   tm_layout(legend.show = TRUE)

tucson_map
```

PLOT



CARTOGRAPHIC ELEMENTS:
All maps contain cartographic elements that provide viewers with context about what is happening on the map.

LEGENDS:
Legends allow viewers of the map to understand the meaning of the symbols used to represent features on the map. If there are multiple feature layers they are represented in the legend as individual legend items. By default legends are dynamic and will update show the feature layers that are visible in the current map frame's extent if new feature layers are added or removed.

3. Use the `tm_compass` argument to add a north arrow to your map.

Within `tm_compass` you can define a number of north arrow options including `north` (the orientation of the north arrow), `type` (the style of the north arrow), `size` (the size of the north arrow), `show.labels` (labels of the north arrow), and `position` (the position of the north arrow on the layout).

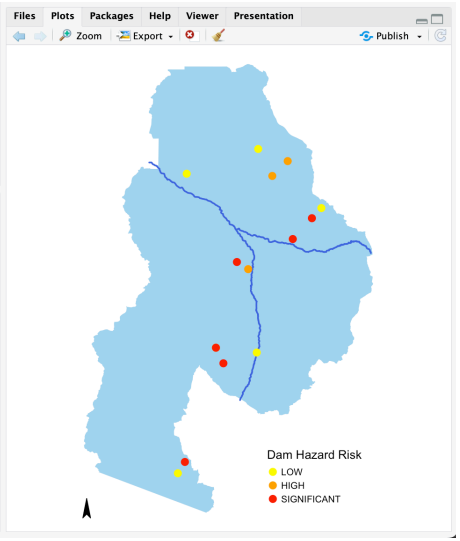
SCRIPT EDITOR

```
## create a tucson map from the three individual maps

tucson_map <- basin_map +
   majtribs_map +
   dams_map_lgd +
   tm_layout(legend.show = TRUE) +
   tm_compass(north = 0, type = "arrow", size = 1.5, show.labels = 0,
              position = c("BOTTOM", "LEFT"))

tucson_map
```

PLOT



4. Use the `tm_scalebar` argument to add a north arrow to your map.

Within `tm_scalebar` you can define a number of scale bar options including `breaks` (the numeric breaks), and `position` (the position of the scalebar on the layout).

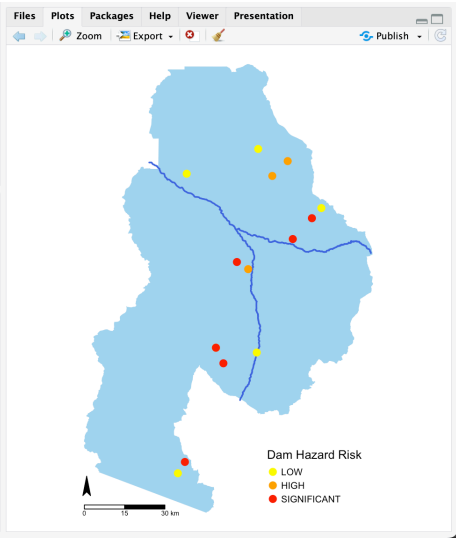
SCRIPT EDITOR

```
## create a tucson map from the three individual maps

tucson_map <- basin_map +
   majtribs_map +
   dams_map_lgd +
   tm_layout(legend.show = TRUE) +
   tm_compass(north = 0, type = "arrow", size = 1.5, show.labels = 0,
              position = c("BOTTOM", "LEFT")) +
   tm_scalebar(breaks = c(0, 15, 30),
               position = c("LEFT", "BOTTOM"))

tucson_map
```

PLOT



NORTH ARROW:
The north arrow indicates the orientation of the map and when placed on the map is automatically associated with the default map frame.

SCALE BAR:
The scale bar are used to visually identify the distance and feature layer sizes on the map. The scale bar is dynamically linked to the default map frame and if that map's extent changes the scale bar will automatically update to represent these changes. Scale bars should be displayed in measures that are appropriate for the scale of the analysis.

5. Use the `tm_title` argument to add a title to your map.

Within `tm_title` you can define a number of title options including `text` (the title of your map).

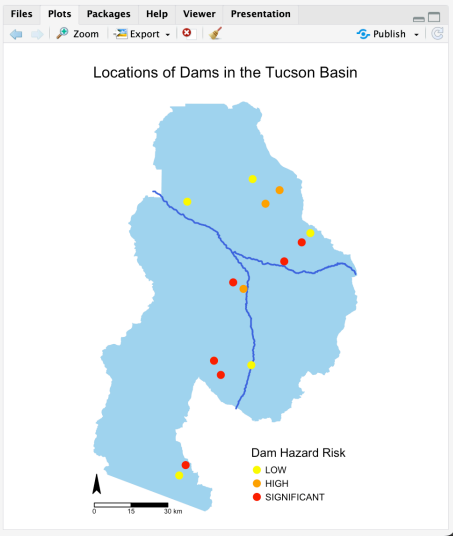
SCRIPT EDITOR

```
## create a tucson map from the three individual maps

tucson_map <- basin_map +
   majtribs_map +
   dams_map_lgd +
   tm_layout(legend.show = TRUE) +
   tm_compass(north = 0, type = "arrow", size = 1.5, show.labels = 0,
              position = c("BOTTOM", "LEFT")) +
   tm_scalebar(breaks = c(0, 15, 30),
               position = c("LEFT", "BOTTOM")) +
   tm_title(text = "Locations of Dams in the Tucson Basin")

tucson_map
```

PLOT



TITLE:
Map titles need to be concise and descriptive generally mentioning the location of the analysis.

HELPFUL HINT:
As stated previously, there are many more visualization options that are available through additional arguments that are not addressed in this tutorial. The elements that you need for your map layout as well as their location will be determined by where your map will be used.

EXPORTING A TMAP OBJECT

1. Use the `tmap_save()` function to export your `tucson_map` object to a .pdf file.

```
Map saved to /Users/lbry-smithg/Desktop/stuff/UA/UAL/gis/rstudio/R_GIS_
intro/plots/tucson_dams_map.pdf
Size: 5 by 7 inches
```

SCRIPT EDITOR

```
## export the tucson map layout as a .pdf

tmap_save(tm = tucson_map,
          filename = "plots/tucson_dams_map.pdf",
          width = 5, height = 7, units = "in",
          dpi = 600)
```

PLOT

```
none
```

EXPORTING MAPS:
The type of map that you will export will be based on how your map layout will be used. There are a number of different options for exporting maps and each has its own utility.

END