Sf tutorial session 2

Applying the sf and terra packages to answer scientific questions!

By Alys and Michael

Efficient spatial processing with SF

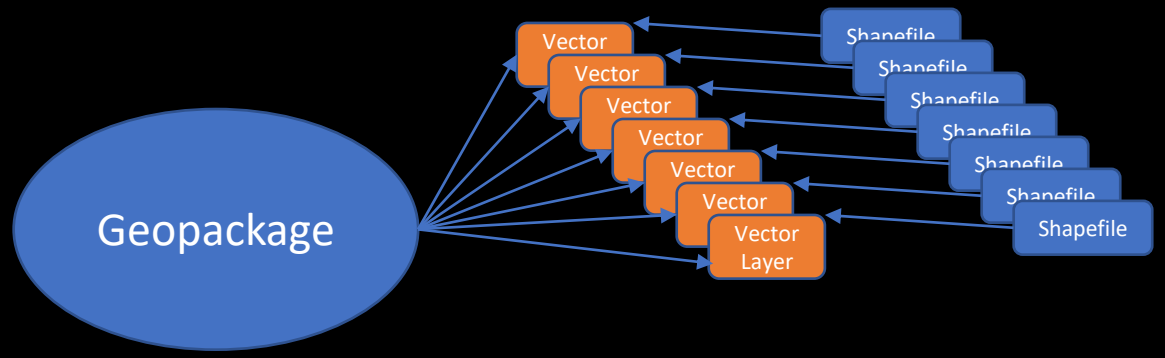
1) What is a Geopackage and why should I care?

2) Reduce output size and processing time by simplifying data

3) Reduce memory requirements by piping data from start to end of operations

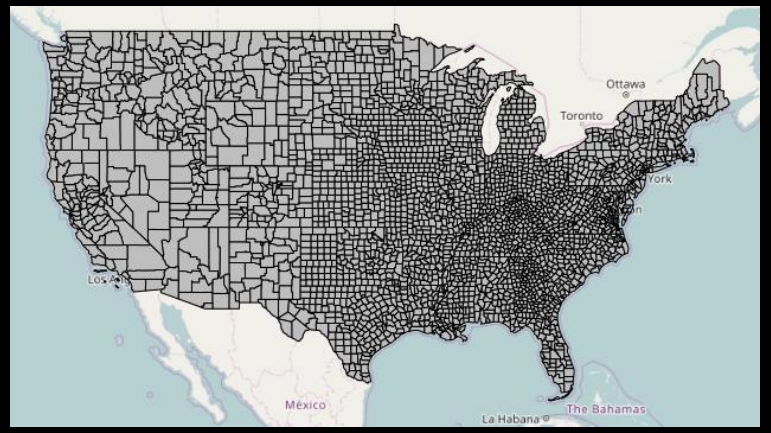
Geopackages instead of shapefiles

- All layers in a single file
- No file size restrictions
- Compatible with R, QGIS and ARC
- Same code in R as with shapefiles

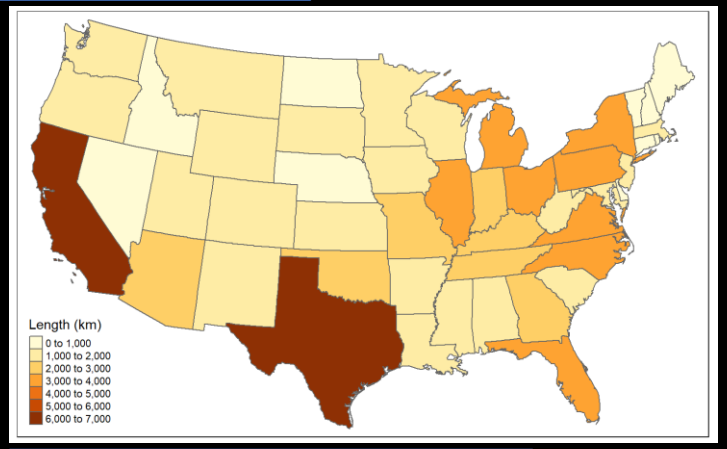



```
> st_layers("sf_tutorial_data.gpkg")
        - 'VirtualXPath'        [XML Path Language - XPath]
Driver: GPKG
Available layers:
  layer_name geometry_type features fields           crs_name
1 EVC_extant                41573     21 GDA94 / MGA zone 55
2    NSW3858                14195      3 GDA94 / MGA zone 55
3    NSW3859                16438      3 GDA94 / MGA zone 55
4   EVC_1750                29768     16 GDA94 / MGA zone 55
>
```

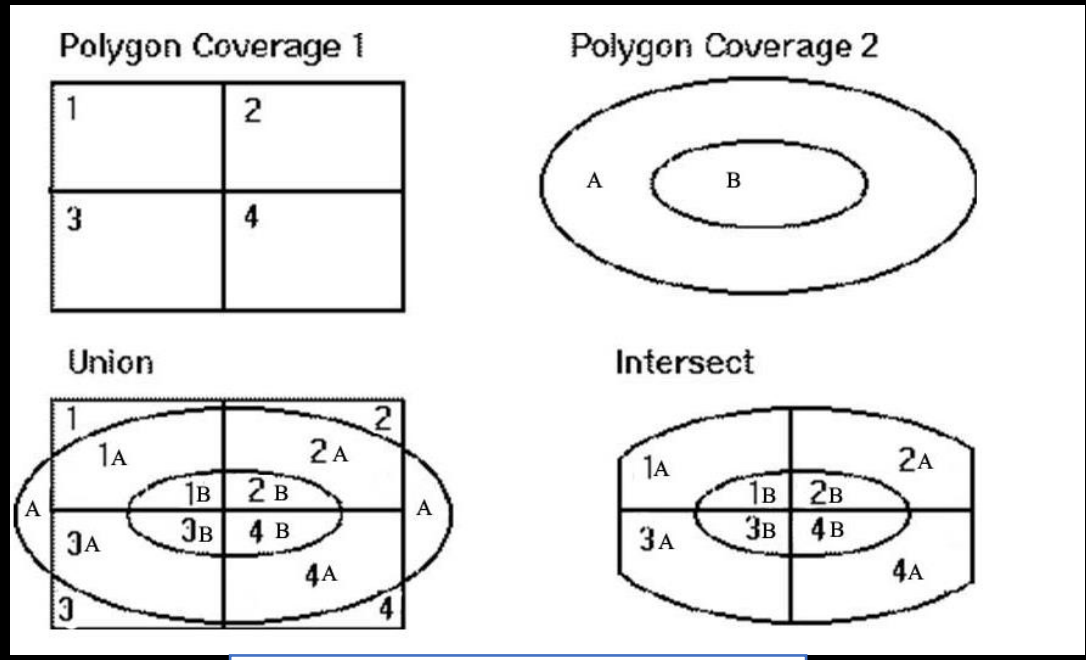
Simplify your data, reduce processing time



https://michaelminn.net/tutorials/r-area-visualization/index.html



http://www.wwview.org/ossa/ossa/13_Vector_Analysis.html



http://web.mit.edu/11.188/www/lectures/lecture7.html

Memory is at a premium, avoid creating objects!

- Spatial operations require a lot of memory, this increases with scale
- Storing spatial objects that are not reused is wasteful

```r
veg_source      <- st_read("source_data.gpkg", layer = "veg")
veg_transformed <- st_transform(veg_source, st_crs(28355))

mainland_border <- st_read("border_data.gpkg", layer = "mainland")
main_bord_trans <- st_transform(mainland_border, st_crs(28355))

veg_mainland    <- st_intersection(veg_source, main_bord_trans)

tas_source      <- st_read("source_data.gpkg", layer = "tas_veg")
tas_transformed <- st_transform(tas_source, st_crs(28355))

all_veg         <- st_union(veg_mainland, tas_transformed)

st_write(all_veg, "outputs.gpkg", layer = "all_veg")
```

```r
st_union(
    st_intersection(
        st_read("source_data.gpkg", layer = "veg") %>% st_transform(st_crs(28355)), #veg_transformed

        st_read("border_data.gpkg", layer = "mainland") %>% st_transform(st_crs(28355))), #main_border

    st_read("source_data.gpkg", layer = "tas_veg") %>% st_transform(st_crs(28355))) %>% #tas_transformed

st_write("outputs.gpkg", layer = "all_veg")
```

What are we doing today

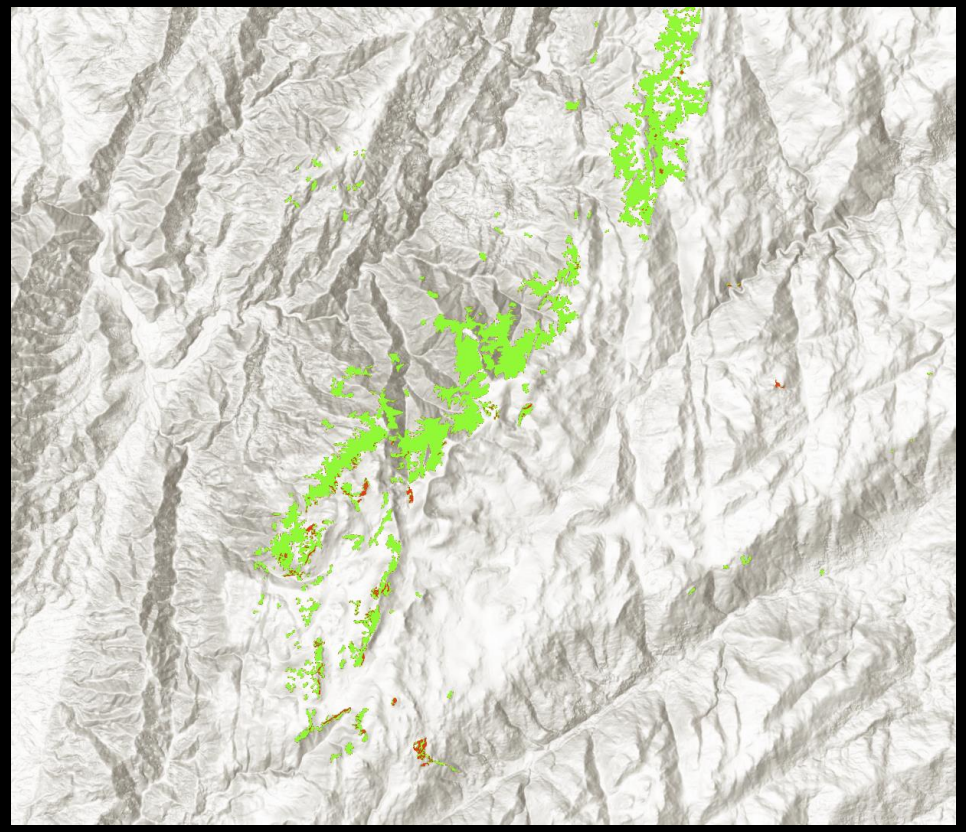
1) Creating a distribution map for an ecosystem or species

2) Analyse population densities around a selection of sites around Victoria

3) Demonstrate the principles of efficient spatial analysis outlined in the previous slides

Building a distribution map from multiple data sources

- Combining selected vegetation classes from two publicly available datasets NSW VISID 3858 from SEED, and EVC mapping from Data Victoria
- Using an Australia wide digital elevation map to reduce distribution to within elevation boundaries
- Analyse changes in our toy ecosystem/species distribution since pre colonization (modelled).



Combining raster and vector data for spatial analysis

- Spatial points layer of random site locations around Victoria
- Population density raster from data for good.
- Using a combination of the sf and terra packages to find population densities within a buffer zone around site locations