trying_no_intersection_code

2024-12-11

```
options(scipen=999)

#install.packages("sf", configure.args = "--with-gdal-config=C:/gdal-filegdb/bin/gdal-config")
#install.packages("rgdal")
#install.packages("rgdal")
#library("terra")
library(raster)
```

```
## Warning: package 'raster' was built under R version 4.4.2
```

```
## Loading required package: sp
```

```
library(sp)
#library("rgdal")

library("sf")
```

```
## Warning: package 'sf' was built under R version 4.4.2
```

```
## Linking to GEOS 3.12.2, GDAL 3.9.3, PROJ 9.4.1; sf_use_s2() is TRUE
```

```
library("spaMM")
```

```
## Registered S3 methods overwritten by 'registry':
##   method               from
##   print.registry_field proxy
##   print.registry_entry proxy
```

```
## spaMM (Rousset & Ferdy, 2014, version 4.5.0) is loaded.
## Type 'help(spaMM)' for a short introduction,
## 'news(package='spaMM')' for news,
## and 'citation('spaMM')' for proper citation.
## Further infos, slides, etc. at https://gitlab.mbb.univ-montp2.fr/francois/spamm-ref.
```

```
library("readxl")
```

```
## Warning: package 'readxl' was built under R version 4.4.2
```

```
library("ggplot2")
library("Rmisc")
```

```
## Loading required package: lattice
```

```
## Loading required package: plyr
```

```
library("pivottabler")
require("SciViews")
```

```
## Loading required package: SciViews
```

```
require("spdep")
```

```
## Loading required package: spdep
```

```
## Loading required package: spData
```

```
## To access larger datasets in this package, install the spDataLarge
## package with: `install.packages('spDataLarge',
## repos='https://nowosad.github.io/drat/', type='source')`
```

```
require("dplyr")
```

```
## Loading required package: dplyr
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## The following objects are masked from 'package:raster':
##
##     intersect, select, union
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
require("xlsx")
```

```
## Loading required package: xlsx
```

```
## Error: package or namespace load failed for 'xlsx':
##  .onLoad failed in loadNamespace() for 'rJava', details:
##   call: fun(libname, pkgname)
##   error: JAVA_HOME cannot be determined from the Registry
```

```r
require("labelVector")
```

```
## Loading required package: labelVector
```

```r
require("foreign")
```

```
## Loading required package: foreign
```

```r
require("optimx")
```

```
## Loading required package: optimx
```

```
##
## Attaching package: 'optimx'
```

```
## The following object is masked from 'package:spaMM':
##
##     coef<-
```

```r
require("tidyverse")
```

```
## Loading required package: tidyverse
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1
## v readr     2.1.5
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::arrange()   masks plyr::arrange()
## x purrr::compact()   masks plyr::compact()
## x dplyr::count()     masks plyr::count()
## x dplyr::desc()      masks plyr::desc()
## x tidyr::extract()   masks raster::extract()
## x dplyr::failwith()  masks plyr::failwith()
## x dplyr::filter()    masks stats::filter()
## x dplyr::id()        masks plyr::id()
## x dplyr::lag()       masks stats::lag()
## x dplyr::mutate()    masks plyr::mutate()
## x dplyr::rename()    masks plyr::rename()
## x dplyr::select()    masks raster::select()
## x dplyr::summarise() masks plyr::summarise()
## x dplyr::summarize() masks plyr::summarize()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
require("sjPlot")
```

```
## Loading required package: sjPlot
## Learn more about sjPlot with 'browseVignettes("sjPlot")'.
```

```r
require("rgdal")
```

```
## Loading required package: rgdal
```

```
## Warning in library(package, lib.loc = lib.loc, character.only = TRUE,
## logical.return = TRUE, : there is no package called 'rgdal'
```

```r
require("tidyr")
require("reghelper")
```

```
## Loading required package: reghelper
##
## Attaching package: 'reghelper'
##
## The following object is masked from 'package:base':
##
##     beta
```

```r
require("stringr")

# checkong the GDAL version
sf::sf_extSoftVersion()
```

```
##           GEOS           GDAL         proj.4 GDAL_with_GEOS     USE_PROJ_H
##       "3.12.2"       "3.9.3"       "9.4.1"         "true"         "true"
##           PROJ
##        "9.4.1"
```

```r
# listing the GDAL drivers
sf::st_drivers()
```

```
##                           name
## PCIDSK                 PCIDSK
## netCDF                 netCDF
## PDS4                     PDS4
## VICAR                   VICAR
## JP2OpenJPEG       JP2OpenJPEG
## PDF                       PDF
## MBTiles                MBTiles
## BAG                       BAG
## EEDA                     EEDA
## OGCAPI                 OGCAPI
## ESRI Shapefile ESRI Shapefile
## MapInfo File     MapInfo File
## UK .NTF               UK .NTF
```

```
## LVBAG                  LVBAG
## OGR_SDTS             OGR_SDTS
## S57                       S57
## DGN                       DGN
## OGR_VRT               OGR_VRT
## Memory                 Memory
## CSV                       CSV
## GML                       GML
## GPX                       GPX
## KML                       KML
## GeoJSON               GeoJSON
## GeoJSONSeq         GeoJSONSeq
## ESRIJSON             ESRIJSON
## TopoJSON             TopoJSON
## OGR_GMT               OGR_GMT
## GPKG                     GPKG
## SQLite                 SQLite
## ODBC                     ODBC
## WAsP                     WAsP
## PGeo                     PGeo
## MSSQLSpatial     MSSQLSpatial
## PostgreSQL         PostgreSQL
## MySQL                   MySQL
## OpenFileGDB       OpenFileGDB
## DXF                       DXF
## CAD                       CAD
## FlatGeobuf         FlatGeobuf
## Geoconcept         Geoconcept
## GeoRSS                 GeoRSS
## VFK                       VFK
## PGDUMP                 PGDUMP
## OSM                       OSM
## GPSBabel             GPSBabel
## OGR_PDS               OGR_PDS
## WFS                       WFS
## OAPIF                   OAPIF
## EDIGEO                 EDIGEO
## SVG                       SVG
## Idrisi                 Idrisi
## XLS                       XLS
## ODS                       ODS
## XLSX                     XLSX
## Elasticsearch   Elasticsearch
## Carto                   Carto
## AmigoCloud         AmigoCloud
## SXF                       SXF
## Selafin                Selafin
## JML                       JML
## PLSCENES             PLSCENES
## CSW                       CSW
## VDV                       VDV
## MVT                       MVT
## NGW                       NGW
## MapML                   MapML
```

```
## GTFS                     GTFS
## PMTiles                PMTiles
## JSONFG                  JSONFG
## MiraMonVector    MiraMonVector
## TIGER                    TIGER
## AVCBin                  AVCBin
## AVCE00                  AVCE00
## HTTP                      HTTP
##                                                        long_name write  copy
## PCIDSK                                        PCIDSK Database File  TRUE FALSE
## netCDF                                   Network Common Data Format  TRUE  TRUE
## PDS4                                   NASA Planetary Data System 4  TRUE  TRUE
## VICAR                                                MIPL VICAR file  TRUE  TRUE
## JP2OpenJPEG          JPEG-2000 driver based on JP2OpenJPEG library FALSE  TRUE
## PDF                                                  Geospatial PDF  TRUE  TRUE
## MBTiles                                                 MBTiles  TRUE  TRUE
## BAG                                     Bathymetry Attributed Grid  TRUE  TRUE
## EEDA                                         Earth Engine Data API FALSE FALSE
## OGCAPI                                                  OGCAPI FALSE FALSE
## ESRI Shapefile                                    ESRI Shapefile  TRUE FALSE
## MapInfo File                                        MapInfo File  TRUE FALSE
## UK .NTF                                                UK .NTF FALSE FALSE
## LVBAG                                   Kadaster LV BAG Extract 2.0 FALSE FALSE
## OGR_SDTS                                                 SDTS FALSE FALSE
## S57                                                IHO S-57 (ENC)  TRUE FALSE
## DGN                                               Microstation DGN  TRUE FALSE
## OGR_VRT                                  VRT - Virtual Datasource FALSE FALSE
## Memory                                                   Memory  TRUE FALSE
## CSV                                   Comma Separated Value (.csv)  TRUE FALSE
## GML                                Geography Markup Language (GML)  TRUE FALSE
## GPX                                                          GPX  TRUE FALSE
## KML                                  Keyhole Markup Language (KML)  TRUE FALSE
## GeoJSON                                                  GeoJSON  TRUE FALSE
## GeoJSONSeq                                      GeoJSON Sequence  TRUE FALSE
## ESRIJSON                                                ESRIJSON FALSE FALSE
## TopoJSON                                                TopoJSON FALSE FALSE
## OGR_GMT                                  GMT ASCII Vectors (.gmt)  TRUE FALSE
## GPKG                                                  GeoPackage  TRUE  TRUE
## SQLite                                      SQLite / Spatialite  TRUE FALSE
## ODBC                                                             FALSE FALSE
## WAsP                                           WAsP .map format  TRUE FALSE
## PGeo                                   ESRI Personal GeoDatabase FALSE FALSE
## MSSQLSpatial                Microsoft SQL Server Spatial Database  TRUE FALSE
## PostgreSQL                                   PostgreSQL/PostGIS  TRUE FALSE
## MySQL                                                    MySQL  TRUE FALSE
## OpenFileGDB                                         ESRI FileGDB  TRUE FALSE
## DXF                                                   AutoCAD DXF  TRUE FALSE
## CAD                                                AutoCAD Driver FALSE FALSE
## FlatGeobuf                                            FlatGeobuf  TRUE FALSE
## Geoconcept                                            Geoconcept  TRUE FALSE
## GeoRSS                                                    GeoRSS  TRUE FALSE
## VFK                          Czech Cadastral Exchange Data Format FALSE FALSE
## PGDUMP                                       PostgreSQL SQL dump  TRUE FALSE
## OSM                                    OpenStreetMap XML and PBF FALSE FALSE
```

```
## GPSBabel                                                GPSBabel  TRUE FALSE
## OGR_PDS                                Planetary Data Systems TABLE FALSE FALSE
## WFS                                   OGC WFS (Web Feature Service) FALSE FALSE
## OAPIF                                            OGC API - Features FALSE FALSE
## EDIGEO                                French EDIGEO exchange format FALSE FALSE
## SVG                                        Scalable Vector Graphics FALSE FALSE
## Idrisi                                         Idrisi Vector (.vct) FALSE FALSE
## XLS                                                MS Excel format FALSE FALSE
## ODS            Open Document/ LibreOffice / OpenOffice Spreadsheet  TRUE FALSE
## XLSX                                 MS Office Open XML spreadsheet  TRUE FALSE
## Elasticsearch                                      Elastic Search  TRUE FALSE
## Carto                                                      Carto  TRUE FALSE
## AmigoCloud                                            AmigoCloud  TRUE FALSE
## SXF                                     Storage and eXchange Format FALSE FALSE
## Selafin                                                  Selafin  TRUE FALSE
## JML                                                   OpenJUMP JML  TRUE FALSE
## PLSCENES                                    Planet Labs Scenes API FALSE FALSE
## CSW                          OGC CSW (Catalog  Service for the Web) FALSE FALSE
## VDV                             VDV-451/VDV-452/INTREST Data Format  TRUE FALSE
## MVT                                          Mapbox Vector Tiles  TRUE FALSE
## NGW                                                  NextGIS Web  TRUE  TRUE
## MapML                                                      MapML  TRUE FALSE
## GTFS                               General Transit Feed Specification FALSE FALSE
## PMTiles                                              ProtoMap Tiles  TRUE FALSE
## JSONFG                               OGC Features and Geometries JSON  TRUE FALSE
## MiraMonVector                      MiraMon Vectors (.pol, .arc, .pnt)  TRUE FALSE
## TIGER                                       U.S. Census TIGER/Line FALSE FALSE
## AVCBin                                    Arc/Info Binary Coverage FALSE FALSE
## AVCE00                                 Arc/Info E00 (ASCII) Coverage FALSE FALSE
## HTTP                                     HTTP Fetching Wrapper FALSE FALSE
##                is_raster is_vector   vsi
## PCIDSK              TRUE      TRUE  TRUE
## netCDF              TRUE      TRUE FALSE
## PDS4                TRUE      TRUE  TRUE
## VICAR               TRUE      TRUE  TRUE
## JP2OpenJPEG         TRUE      TRUE  TRUE
## PDF                 TRUE      TRUE  TRUE
## MBTiles             TRUE      TRUE  TRUE
## BAG                 TRUE      TRUE  TRUE
## EEDA               FALSE      TRUE FALSE
## OGCAPI              TRUE      TRUE  TRUE
## ESRI Shapefile     FALSE      TRUE  TRUE
## MapInfo File       FALSE      TRUE  TRUE
## UK .NTF            FALSE      TRUE  TRUE
## LVBAG              FALSE      TRUE  TRUE
## OGR_SDTS           FALSE      TRUE  TRUE
## S57                FALSE      TRUE  TRUE
## DGN                FALSE      TRUE  TRUE
## OGR_VRT            FALSE      TRUE  TRUE
## Memory             FALSE      TRUE FALSE
## CSV                FALSE      TRUE  TRUE
## GML                FALSE      TRUE  TRUE
## GPX                FALSE      TRUE  TRUE
## KML                FALSE      TRUE  TRUE
```

```
## GeoJSON            FALSE      TRUE  TRUE
## GeoJSONSeq         FALSE      TRUE  TRUE
## ESRIJSON           FALSE      TRUE  TRUE
## TopoJSON           FALSE      TRUE  TRUE
## OGR_GMT            FALSE      TRUE  TRUE
## GPKG                TRUE      TRUE  TRUE
## SQLite             FALSE      TRUE  TRUE
## ODBC               FALSE      TRUE FALSE
## WAsP               FALSE      TRUE  TRUE
## PGeo               FALSE      TRUE FALSE
## MSSQLSpatial       FALSE      TRUE FALSE
## PostgreSQL         FALSE      TRUE FALSE
## MySQL              FALSE      TRUE FALSE
## OpenFileGDB         TRUE      TRUE  TRUE
## DXF                FALSE      TRUE  TRUE
## CAD                 TRUE      TRUE  TRUE
## FlatGeobuf         FALSE      TRUE  TRUE
## Geoconcept         FALSE      TRUE  TRUE
## GeoRSS             FALSE      TRUE  TRUE
## VFK                FALSE      TRUE FALSE
## PGDUMP             FALSE      TRUE  TRUE
## OSM                FALSE      TRUE  TRUE
## GPSBabel           FALSE      TRUE FALSE
## OGR_PDS            FALSE      TRUE  TRUE
## WFS                FALSE      TRUE  TRUE
## OAPIF              FALSE      TRUE FALSE
## EDIGEO             FALSE      TRUE  TRUE
## SVG                FALSE      TRUE  TRUE
## Idrisi             FALSE      TRUE  TRUE
## XLS                FALSE      TRUE FALSE
## ODS                FALSE      TRUE  TRUE
## XLSX               FALSE      TRUE  TRUE
## Elasticsearch      FALSE      TRUE FALSE
## Carto              FALSE      TRUE FALSE
## AmigoCloud         FALSE      TRUE FALSE
## SXF                FALSE      TRUE  TRUE
## Selafin            FALSE      TRUE  TRUE
## JML                FALSE      TRUE  TRUE
## PLSCENES            TRUE      TRUE FALSE
## CSW                FALSE      TRUE FALSE
## VDV                FALSE      TRUE  TRUE
## MVT                FALSE      TRUE  TRUE
## NGW                 TRUE      TRUE FALSE
## MapML              FALSE      TRUE  TRUE
## GTFS               FALSE      TRUE  TRUE
## PMTiles            FALSE      TRUE  TRUE
## JSONFG             FALSE      TRUE  TRUE
## MiraMonVector      FALSE      TRUE  TRUE
## TIGER              FALSE      TRUE  TRUE
## AVCBin             FALSE      TRUE  TRUE
## AVCE00             FALSE      TRUE  TRUE
## HTTP                TRUE      TRUE FALSE
```

Instructions

' Step 1: Intersect crop shapefile with census tract shapefile Step 3: Export Step 1 as txt file and read it "crop.category.tract" Step 4: Need to make sure the number of crop grown has not changed (80 landuses inlcuding some that were exlcuded) Step 5: Need to make sure the number of pesticides applied has not changed (48 at the time of analysis) '

County name

```
#emily
setwd('C:\\Users\\emily\\OneDrive - Washington State University (email.wsu.edu)\\cpts475-project')

# county FIPS code

#FIPS <- read_excel("P:/Ofer Amram/WA DOH/Pesticide/Data/geographic_codes.xls", sheet = "County")

# emily edited this so hopefully will read from my folder
# county names and codes
FIPS <- read.csv("OREGON/OR_geographic_codes.csv")

# gave up on this part, so used the csv above.
#FIPS <- read_excel("OREGON/OR_geographic_codes.xls", sheet = "County")
#file.exists("OREGON/OR_geographic_codes.xlsx")  # Should return TRUE if the path is correct
#FIPS <- read_excel("OREGON/OR_geographic_codes.xlsx")

FIPS$COUNTY_FIPS_CODE <- as.numeric(FIPS$COUNTYFP)
FIPS <- subset(FIPS, select = c(COUNTY_NAME, COUNTYFPL, COUNTY_FIPS_CODE ))
FIPS
```

```
##    COUNTY_NAME COUNTYFPL COUNTY_FIPS_CODE
## 1        BAKER     41001                1
## 2       BENTON     41003                3
## 3    CLACKAMAS     41005                5
## 4      CLATSOP     41007                7
## 5     COLUMBIA     41009                9
## 6         COOS     41011               11
## 7        CROOK     41013               13
## 8        CURRY     41015               15
## 9    DESCHUTES     41017               17
## 10     DOUGLAS     41019               19
## 11     GILLIAM     41021               21
## 12       GRANT     41023               23
## 13      HARNEY     41025               25
## 14  HOOD RIVER     41027               27
## 15     JACKSON     41029               29
## 16   JEFFERSON     41031               31
## 17   JOSEPHINE     41033               33
## 18     KLAMATH     41035               35
## 19        LAKE     41037               37
## 20        LANE     41039               39
## 21     LINCOLN     41041               41
## 22        LINN     41043               43
## 23     MALHEUR     41045               45
## 24      MARION     41047               47
## 25      MORROW     41049               49
## 26   MULTNOMAH     41051               51
```

```
## 27        POLK     41053              53
## 28     SHERMAN     41055              55
## 29   TILLAMOOK     41057              57
## 30    UMATILLA     41059              59
## 31       UNION     41061              61
## 32     WALLOWA     41063              63
## 33       WASCO     41065              65
## 34  WASHINGTON     41067              67
## 35     WHEELER     41069              69
## 36     YAMHILL     41071              71
```

Census tract area for adjusting pesticide application

county FIPS code

```
#ct_area <- sf::st_read("C:/Users/solmaz.amiri/Documents/Washington State University/Ofer Amram/DOH/Pes

# emily/sarah edit
# tract20 file for washington. Downloaded from https://www.atsdr.cdc.gov/place-health/php/svi/svi-data-

#st_layers("SVI2022_OREGON_censustract.gdb")

ct_area <- sf::st_read("SVI2022_OREGON_censustract.gdb", layer = "SVI2022_OREGON_tract")
```

```
## Reading layer ‘SVI2022_OREGON_tract’ from data source
##   ‘C:\Users\emily\OneDrive - Washington State University (email.wsu.edu)\cpts475-project\SVI2022_ORE
##   using driver ‘OpenFileGDB’
## Simple feature collection with 994 features and 160 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -124.5662 ymin: 41.99179 xmax: -116.4635 ymax: 46.29083
## Geodetic CRS:  NAD83
```

```
#converts the object to a data frame, since we are removing the spatial component
sf::st_geometry(ct_area) <- NULL

#colnames(ct_area)
# replaced ID2 and AREA columns with FIPS and AREA_SQMI because there was neither ID2 and AREA
ct_area <- subset(ct_area, select = c(FIPS, AREA_SQMI))
```

Crop categories

read crop category spreadsheet

```
# ours
crop.category <- read.csv("OREGON/OR_Crops.csv")

crop.category <- subset(crop.category, select = c(Class_Name, Category))
```

this was our vectoized layer. It seems that the only columns from this are the geometry, which we end up taking out, and the DN, which is the color of the pixel. The pixel color should correspond to the crop

```
gpkg_info <- st_read("OR_CROPS_vector.gpkg", layer = "OR_CROPS_vector", query = "SELECT * FROM OR_CROPS
```

```
## Warning in CPL_read_ogr(dsn, layer, query, as.character(options), quiet, :
## argument layer is ignored when query is specified
```

```
## Reading query `SELECT * FROM OR_CROPS_vector LIMIT 5'
## from data source
##   `C:\Users\emily\OneDrive - Washington State University (email.wsu.edu)\cpts475-project\OR_CROPS_ve
##   using driver `GPKG'
## Simple feature collection with 5 features and 1 field
## Geometry type: POLYGON
## Dimension:     XY
## Bounding box:  xmin: -13742220 ymin: 5819247 xmax: -13741290 ymax: 5820537
## Projected CRS: WGS 84 / Pseudo-Mercator
```

```
head(gpkg_info)
```

```
## Simple feature collection with 5 features and 1 field
## Geometry type: POLYGON
## Dimension:     XY
## Bounding box:  xmin: -13742220 ymin: 5819247 xmax: -13741290 ymax: 5820537
## Projected CRS: WGS 84 / Pseudo-Mercator
##    DN                           geom
## 1 111 POLYGON ((-13742221 5820537...
## 2 111 POLYGON ((-13741741 5819907...
## 3 111 POLYGON ((-13741681 5819787...
## 4 111 POLYGON ((-13741591 5819667...
## 5 111 POLYGON ((-13741351 5819307...
```

```
gpkg_info <- st_read("intersection.gpkg", layer = "intersection", query = "SELECT * FROM intersection L
```

```
## Warning in CPL_read_ogr(dsn, layer, query, as.character(options), quiet, :
## argument layer is ignored when query is specified
```

```
## Reading query `SELECT * FROM intersection LIMIT 1'
## from data source
##   `C:\Users\emily\OneDrive - Washington State University (email.wsu.edu)\cpts475-project\intersectio
##   using driver `GPKG'
## Simple feature collection with 1 feature and 162 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -13742220 ymin: 5820507 xmax: -13742160 ymax: 5820537
## Projected CRS: WGS 84 / Pseudo-Mercator
```

```
head(colnames(gpkg_info))
```

```
## [1] "DN"       "OBJECTID" "ST"       "STATE"    "ST_ABBR"  "STCNTY"
```

Crop data

read the intersection of the census tract and the crop. This is not really possible to create in R, since the crop data when downloaded is in raster format. We used the open source QGis to make this. It takes a lot of memory(~20gb) and time (6 hrs) on my (emily) personal computer.

shout out to Annie Kintner, she is the our consulting GIS expert and we could not have done this without her.

```
crop.category.tract <- sf::st_read("intersection.gpkg", layer = "intersection", geometry = "NULL", query
```

```
## Warning in CPL_read_ogr(dsn, layer, query, as.character(options), quiet, :
## argument layer is ignored when query is specified
```

```
## Reading query `SELECT DN, OBJECTID, ST, STATE, COUNTY, FIPS, AREA_SQMI, E_POV150, M_POV150, E_UNEMP,
## from data source
##   `C:\Users\emily\OneDrive - Washington State University (email.wsu.edu)\cpts475-project\intersectio
##   using driver `GPKG'
```

```
## Warning: no simple feature geometries present: returning a data.frame or tbl_df
```

county FIPS

replaced crop.category.tract$ID_1 with FIPS.

```
crop.category.tract$county.FIPS <- substr(crop.category.tract$FIPS, 1, 5)
```

join with crop category and drop exclude and soybeans

```
#grab the DN to crop
pixels <- read.csv("crop_pixels.csv", header = TRUE, sep = ",")
crop.category.tract <- left_join(crop.category.tract, pixels, by = "DN")
crop.category.tract <- left_join(crop.category.tract, crop.category, by = "Class_Name")
```

```
crop.category.tract <- subset(crop.category.tract, Category != "exclude" & Category != "soybeans")

str(crop.category.tract)
```

```
## 'data.frame':    1300728 obs. of  29 variables:
##  $ DN         : int  71 37 37 37 37 37 37 14 14 70 ...
##  $ OBJECTID   : num  125 125 125 125 125 125 125 128 128 128 ...
##  $ ST         : chr  "41" "41" "41" "41" ...
##  $ STATE      : chr  "Oregon" "Oregon" "Oregon" "Oregon" ...
##  $ COUNTY     : chr  "Clatsop County" "Clatsop County" "Clatsop County" "Clatsop County" ...
##  $ FIPS       : chr  "41007951200" "41007951200" "41007951200" "41007951200" ...
##  $ AREA_SQMI  : num  411 411 411 411 411 ...
##  $ E_POV150   : int  837 837 837 837 837 837 837 635 635 635 ...
##  $ M_POV150   : int  325 325 325 325 325 325 325 223 223 223 ...
##  $ E_UNEMP    : int  55 55 55 55 55 55 55 60 60 60 ...
##  $ M_UNEMP    : int  40 40 40 40 40 40 40 61 61 61 ...
##  $ E_NOHSDP   : int  202 202 202 202 202 202 202 195 195 195 ...
##  $ M_NOHSDP   : int  93 93 93 93 93 93 93 145 145 145 ...
##  $ E_AGE65    : int  924 924 924 924 924 924 924 546 546 546 ...
##  $ M_AGE65    : int  198 198 198 198 198 198 198 171 171 171 ...
##  $ E_AGE17    : int  662 662 662 662 662 662 662 630 630 630 ...
##  $ M_AGE17    : int  176 176 176 176 176 176 176 192 192 192 ...
##  $ E_DISABL   : int  735 735 735 735 735 735 735 519 519 519 ...
##  $ M_DISABL   : int  181 181 181 181 181 181 181 190 190 190 ...
##  $ E_SNGPNT   : int  13 13 13 13 13 13 13 0 0 0 ...
##  $ M_SNGPNT   : int  15 15 15 15 15 15 15 18 18 18 ...
##  $ E_HISP     : int  138 138 138 138 138 138 138 109 109 109 ...
##  $ M_HISP     : int  118 118 118 118 118 118 118 127 127 127 ...
##  $ E_LIMENG   : int  7 7 7 7 7 7 7 0 0 0 ...
##  $ M_LIMENG   : int  51 51 51 51 51 51 51 52 52 52 ...
##  $ Shape_Area : num  0.127 0.127 0.127 0.127 0.127 ...
##  $ county.FIPS: chr  "41007" "41007" "41007" "41007" ...
##  $ Class_Name : chr  "Other Tree Crops" "Other Hay/Non Alfalfa" "Other Hay/Non Alfalfa" "Other Hay/N
##  $ Category   : chr  "othercrops" "pastureandhay" "pastureandhay" "pastureandhay" ...
```

```
#this number is wayyyyyyyy too big
sum(crop.category.tract$Shape_Area)
```

```
## [1] 521879.9
```

```
crop.category.tract.test <- crop.category.tract
```

area excluded (this does nothing once replaced with AREA_SQMI)

```
crop.category.tract <- subset(crop.category.tract, AREA_SQMI >= 50)
#sum(crop.category.tract.exclude$Shape_Area)
#summary(crop.category.tract)
```

remove areas <= 18000 m2

crop.category.tract1 <- subset(crop.category.tract, area_m2 > 18000)

not remove based on <= 18000 m2, replaced with AREA_SQMI

```
crop.category.tract1 <- crop.category.tract


crop.category.tract1 %>%
  group_by(Category) %>%
  summarise(AREA_SQMI = sum(Shape_Area))
```

```
## # A tibble: 7 x 2
##   Category           AREA_SQMI
##   <chr>                  <dbl>
## 1 alfalfa              166420.
## 2 corn                   1467.
## 3 orchardsandgrapes      5938.
## 4 othercrops            55254.
## 5 pastureandhay        222118.
## 6 vegetablesandfruit    14760.
## 7 wheat                 53892.
```

Crop area by county

summarize crop area by county

replaced with AREA_SQMI

```
crop_area_county <- crop.category.tract1 %>%
  group_by(county.FIPS,Category) %>%
  summarise(AREA_SQMI = sum(Shape_Area))
```

```
## `summarise()` has grouped output by 'county.FIPS'. You can override using the
## `.groups` argument.
```

```
crop_area_county.flat <- crop.category.tract1 %>%
  group_by(county.FIPS) %>%
  summarise(AREA_SQMI = sum(Shape_Area)) #AREA_SQMI

sum(crop_area_county$AREA_SQMI)
```

```
## [1] 519848.7
```

```r
sum(crop_area_county.flat$AREA_SQMI)
```

```
## [1] 519848.7
```

convert from long to wide

```r
crop_area_county1 <- spread(crop_area_county, key = Category, value = AREA_SQMI)

crop_area_county1[is.na(crop_area_county1)] <- 0
```

State and county pesticide data - High estimates used

pesticide by crop state

```r
#HighEstimate_AgPestUsebyCropGroup <- sf::st_read("P:/Ofer Amram/WA DOH/Pesticide/Data/Cropland.gdb", l

# ours
HighEstimate_AgPestUsebyCropGroup <- read.csv("OREGON/EP_HighEstimate_OR.csv", header = TRUE, sep = ",")

#we already seperated this by hand
#HighEstimate_AgPestUsebyCropGroup <- subset(HighEstimate_AgPestUsebyCropGroup, State_FIPS_code == 53 &

HighEstimate_AgPestUsebyCropGroup <- subset(HighEstimate_AgPestUsebyCropGroup, select = c(State_FIPS_co
HighEstimate_AgPestUsebyCropGroup[is.na(HighEstimate_AgPestUsebyCropGroup)] <- 0
head(HighEstimate_AgPestUsebyCropGroup)
```

```
##   State_FIPS_code Year     Compound   Corn   Wheat Vegetables_and_fruit
## 1              41 2019        2,4-D 3682.5 27942.3                 11.3
## 2              41 2019  ACETAMIPRID    0.0     0.0                  4.3
## 3              41 2019   ACETOCHLOR 2653.6     0.0                  0.0
## 4              41 2019     ATRAZINE 2571.7     0.0              15198.3
## 5              41 2019 AZOXYSTROBIN    0.0  5034.1               2943.8
## 6              41 2019    BENTAZONE    0.0     0.0               6649.4
##   Orchards_and_grapes Alfalfa Pasture_and_hay Other_crops
## 1             29214.9  1269.1         12078.0      5692.9
## 2              1318.4     0.0             0.0         0.0
## 3                 0.0     0.0             0.0         0.0
## 4                 0.0     0.0         55125.3         0.0
## 5              2352.4     0.0             0.0       285.1
## 6                 0.0     0.0             0.0         0.0
```

pesticide use in Or

```r
estimate.2019 <- subset(HighEstimate_AgPestUsebyCropGroup, State_FIPS_code == 41 & Year == 2019)
estimate.2019 <- subset(estimate.2019, select = c(Compound, Corn, Wheat, Vegetables_and_fruit, Orchards_
str(estimate.2019)
```

```
## 'data.frame':    49 obs. of  8 variables:
##  $ Compound            : chr  "2,4-D" "ACETAMIPRID" "ACETOCHLOR" "ATRAZINE" ...
##  $ Corn                : num  3682 0 2654 2572 0 ...
##  $ Wheat               : num  27942 0 0 0 5034 ...
##  $ Vegetables_and_fruit: num  11.3 4.3 0 15198.3 2943.8 ...
##  $ Orchards_and_grapes : num  29215 1318 0 0 2352 ...
##  $ Alfalfa             : num  1269 0 0 0 0 ...
##  $ Pasture_and_hay     : num  12078 0 0 55125 0 ...
##  $ Other_crops         : num  5693 0 0 0 285 ...
```

```r
summary(estimate.2019)
```

```
##    Compound              Corn           Wheat        Vegetables_and_fruit
##  Length:49          Min.   :    0   Min.   :     0   Min.   :    0.0
##  Class :character   1st Qu.:    0   1st Qu.:     0   1st Qu.:    0.0
##  Mode  :character   Median :    0   Median :     0   Median :  423.7
##                     Mean   : 1391   Mean   :  9902   Mean   : 5761.0
##                     3rd Qu.:    0   3rd Qu.:     0   3rd Qu.: 6021.5
##                     Max.   :29719   Max.   :356139   Max.   :63168.6
##  Orchards_and_grapes    Alfalfa      Pasture_and_hay   Other_crops
##  Min.   :    0.0     Min.   :    0   Min.   :     0   Min.   :    0.0
##  1st Qu.:    0.0     1st Qu.:    0   1st Qu.:     0   1st Qu.:    0.0
##  Median :   12.6     Median :    0   Median :     0   Median :    0.0
##  Mean   : 2895.3     Mean   : 2705   Mean   : 12164   Mean   :  585.4
##  3rd Qu.: 1677.4     3rd Qu.:    0   3rd Qu.:     0   3rd Qu.:    9.3
##  Max.   :29214.9     Max.   :52724   Max.   :475794   Max.   :19705.8
```

```r
#unsure why she had 6:15 here since she doesnt even select that many columns to use.
cols = c(1:ncol(estimate.2019))
#cols

estimate.2019[,cols] = apply(estimate.2019[,cols], 2, function(x) as.numeric(as.character(x)));
```

```
## Warning in FUN(newX[, i], ...): NAs introduced by coercion
```

```r
estimate.2019$all <- rowSums(estimate.2019[ , 1:ncol(estimate.2019)], na.rm = T)
sum(estimate.2019$all)
```

```
## [1] 1734812
```

pesticide applied to each crop category

```
HighEstimate_AgPestUsebyCropGroup$Corn <- as.numeric(HighEstimate_AgPestUsebyCropGroup$Corn)  # corn is

HighEstimate_AgPestUsebyCropGroup[,4:ncol(HighEstimate_AgPestUsebyCropGroup)] <- lapply(HighEstimate_Agl

HighEstimate_AgPestUsebyCropGroup$Corn1 <- as.numeric(ifelse(HighEstimate_AgPestUsebyCropGroup$Corn ==
                                                             HighEstimate_AgPestUsebyCropGroup$Corn / ro


HighEstimate_AgPestUsebyCropGroup$Corn1 <-  as.numeric(ifelse(HighEstimate_AgPestUsebyCropGroup$Corn ==
                                                 format(HighEstimate_AgPestUsebyCropGroup$Corn / rowSums(Hi


HighEstimate_AgPestUsebyCropGroup$Wheat1 <-  as.numeric(ifelse(HighEstimate_AgPestUsebyCropGroup$Wheat =
                                                       format(HighEstimate_AgPestUsebyCropGroup$Wheat / row


HighEstimate_AgPestUsebyCropGroup$Vegetables_and_fruit1 <-  as.numeric(ifelse(HighEstimate_AgPestUsebyCi
                                                     format(HighEstimate_AgPestUsebyCropGroup$Vegetables_


HighEstimate_AgPestUsebyCropGroup$Orchards_and_grapes1 <-  as.numeric(ifelse(HighEstimate_AgPestUsebyCro
                                                             format(HighEstimate_AgPestUsebyCropGi


HighEstimate_AgPestUsebyCropGroup$Alfalfa1 <-  as.numeric(ifelse(HighEstimate_AgPestUsebyCropGroup$Alfal
                                                           format(HighEstimate_AgPestUsebyCropGro


HighEstimate_AgPestUsebyCropGroup$Pasture_and_hay1 <-  as.numeric(ifelse(HighEstimate_AgPestUsebyCropGro
                                                       format(HighEstimate_AgPestUsebyCropGroup$Pasture_a


HighEstimate_AgPestUsebyCropGroup$Other_crops1 <-  as.numeric(ifelse(HighEstimate_AgPestUsebyCropGroup$0
                                                             format(HighEstimate_AgPestUsebyCropGroup$0



HighEstimate_AgPestUsebyCropGroup <- subset(HighEstimate_AgPestUsebyCropGroup, select = c(Compound, Corn
```

pesticide data by county

```
#EPest_county_estimates <- sf::st_read("P:/Ofer Amram/WA DOH/Pesticide/Data/Cropland.gdb", layer = "EPes
EPest_county_estimates <- read.csv("OREGON/OR_EPest_county_estimates_2019.csv", header = TRUE, sep = ",
#EPest_county_estimates <- subset(EPest_county_estimates, STATE_FIPS_CODE == 53 )


EPest_county_estimates1 <- subset(EPest_county_estimates, select = c(COMPOUND, COUNTY_FIPS_CODE,EPEST_HI

EPest_county_estimates1 <- left_join(FIPS, EPest_county_estimates1, by = "COUNTY_FIPS_CODE")
```

```
EPest_county_estimates2 <- EPest_county_estimates1 %>%
  group_by(COUNTY_FIPS_CODE) %>%
  summarise(pesticide = sum(EPEST_HIGH_KG))
HighEstimate_AgPestUsebyCropGroup
```

```
##                         Compound       Corn1     Wheat1 Vegetables_and_fruit1
## 1                           2,4-D 0.046094027 0.34975509           0.000141442
## 2                     ACETAMIPRID 0.000000000 0.00000000           0.003250926
## 3                     ACETOCHLOR 0.999623295 0.00000000           0.000000000
## 4                       ATRAZINE 0.035279349 0.00000000           0.208494823
## 5                    AZOXYSTROBIN 0.000000000 0.47422612           0.277301703
## 6                       BENTAZONE 0.000000000 0.00000000           1.000000000
## 7                 BENZOVINDIFLUPYR 0.000000000 0.70018398           0.299667547
## 8                       BOSCALID 0.000000000 0.00000000           0.702461404
## 9                     BROMOXYNIL 0.000000000 0.88724580           0.105546563
## 10                      CARBARYL 0.000000000 0.00000000           0.935570872
## 11            CHLORANTRANILIPROLE 0.000000000 0.00000000           0.000000000
## 12                  CHLORPYRIFOS 0.012867007 0.00000000           0.241601569
## 13               CYANTRANILIPROLE 0.000000000 0.00000000           0.896780966
## 14                    CYPRODINIL 0.000000000 0.00000000           0.672578116
## 15                      DIAZINON 0.000000000 0.00000000           0.000000000
## 16                       DICAMBA 0.120303349 0.15873596           0.000000000
## 17                  DIMETHENAMID-P 0.221370242 0.00000000           0.739674985
## 18 DIMETHENAMID & DIMETHENAMID-P 0.221370242 0.00000000           0.739674985
## 19                     DIMETHOATE 0.000000000 0.08031678           0.323708135
## 20                         DIURON 0.000000000 0.21461988           0.004704313
## 21                     ETHOPROPHOS 0.000000000 0.00000000           1.000000000
## 22                       ETOXAZOLE 0.902261753 0.00000000           0.000000000
## 23                       FIPRONIL 0.000000000 0.00000000           1.000000000
## 24                     GLYPHOSATE 0.029076178 0.34843484           0.061802035
## 25                    HALOSULFURON 0.000000000 0.00000000           0.056250000
## 26                      HEXAZINONE 0.000000000 0.00000000           0.000000000
## 27                     IMAZETHAPYR 0.000000000 0.00000000           0.084179688
## 28                    IMIDACLOPRID 0.000000000 0.00000000           0.753428062
## 29                         LINURON 0.000000000 0.00000000           1.000000000
## 30                      MALATHION 0.000000000 0.00000000           0.475130954
## 31                     METCONAZOLE 0.000000000 0.00000000           0.000000000
## 32                       METHOMYL 0.000000000 0.00000000           1.000000000
## 33                 METHOXYFENOZIDE 0.000000000 0.00000000           0.000000000
## 34                   METOLACHLOR-S 0.221679535 0.00000000           0.778315525
## 35   METOLACHLOR & METOLACHLOR-S 0.221679535 0.00000000           0.778315525
## 36                     METRIBUZIN 0.000000000 0.38624556           0.106929664
## 37                    MYCLOBUTANIL 0.000000000 0.00000000           0.000000000
## 38                       ORYZALIN 0.000000000 0.00000000           0.000000000
## 39                  PROPICONAZOLE 0.009474465 0.74468645           0.000000000
## 40                 PYRACLOSTROBIN 0.000000000 0.58869420           0.138794759
## 41                   PYRIMETHANIL 0.000000000 0.00000000           1.000000000
## 42                       SIMAZINE 0.000000000 0.00000000           0.144186166
## 43                  SULFENTRAZONE 0.000000000 0.00000000           0.041442518
## 44                    SULFOXAFLOR 0.000000000 0.00000000           0.000000000
## 45                    TEBUCONAZOLE 0.000000000 0.74489656           0.249369113
## 46                  TETRACONAZOLE 0.000000000 0.00000000           0.000000000
## 47                   THIAMETHOXAM 0.000000000 0.00000000           0.414164742
```

```
## 48                    TRICLOPYR 0.000000000 0.00000000           0.000000000
## 49                TRIFLOXYSTROBIN 0.000000000 0.00000000           0.000000000
##    Orchards_and_grapes1   Alfalfa1 Pasture_and_hay1 Other_crops1
## 1           0.365682683 0.01588524        0.1511795  0.071257511
## 2           0.996746624 0.00000000        0.0000000  0.000000000
## 3           0.000000000 0.00000000        0.0000000  0.000000000
## 4           0.000000000 0.00000000        0.7562232  0.000000000
## 5           0.221586890 0.00000000        0.0000000  0.026854746
## 6           0.000000000 0.00000000        0.0000000  0.000000000
## 7           0.000000000 0.00000000        0.0000000  0.000000000
## 8           0.297508158 0.00000000        0.0000000  0.000000000
## 9           0.000000000 0.00000000        0.0000000  0.007205404
## 10          0.064423107 0.00000000        0.0000000  0.000000000
## 11          1.000000000 0.00000000        0.0000000  0.000000000
## 12          0.305754205 0.43185437        0.0000000  0.007916342
## 13          0.103111187 0.00000000        0.0000000  0.000000000
## 14          0.327156426 0.00000000        0.0000000  0.000000000
## 15          1.000000000 0.00000000        0.0000000  0.000000000
## 16          0.000000000 0.06903349        0.6201904  0.031725127
## 17          0.000000000 0.00000000        0.0000000  0.038935047
## 18          0.000000000 0.00000000        0.0000000  0.038935047
## 19          0.000000000 0.59593407        0.0000000  0.000000000
## 20          0.522548185 0.25811963        0.0000000  0.000000000
## 21          0.000000000 0.00000000        0.0000000  0.000000000
## 22          0.091170722 0.00000000        0.0000000  0.000000000
## 23          0.000000000 0.00000000        0.0000000  0.000000000
## 24          0.024322676 0.05158368        0.4655008  0.019279481
## 25          0.942645337 0.00000000        0.0000000  0.000000000
## 26          0.000000000 1.00000000        0.0000000  0.000000000
## 27          0.000000000 0.91566976        0.0000000  0.000000000
## 28          0.245185403 0.00000000        0.0000000  0.001359331
## 29          0.000000000 0.00000000        0.0000000  0.000000000
## 30          0.140024548 0.38483220        0.0000000  0.000000000
## 31          0.000000000 0.00000000        0.0000000  1.000000000
## 32          0.000000000 0.00000000        0.0000000  0.000000000
## 33          1.000000000 0.00000000        0.0000000  0.000000000
## 34          0.000000000 0.00000000        0.0000000  0.000000000
## 35          0.000000000 0.00000000        0.0000000  0.000000000
## 36          0.000000000 0.50681881        0.0000000  0.000000000
## 37          1.000000000 0.00000000        0.0000000  0.000000000
## 38          1.000000000 0.00000000        0.0000000  0.000000000
## 39          0.236662819 0.00000000        0.0000000  0.009165734
## 40          0.263332152 0.00000000        0.0000000  0.009156499
## 41          0.000000000 0.00000000        0.0000000  0.000000000
## 42          0.855808554 0.00000000        0.0000000  0.000000000
## 43          0.000000000 0.00000000        0.9585462  0.000000000
## 44          0.156250000 0.00000000        0.0000000  0.835589942
## 45          0.005668556 0.00000000        0.0000000  0.000000000
## 46          0.000000000 0.00000000        0.0000000  1.000000000
## 47          0.585213306 0.00000000        0.0000000  0.000000000
## 48          0.000000000 0.00000000        1.0000000  0.000000000
## 49          1.000000000 0.00000000        0.0000000  0.000000000
```

County calculation

join state with county to calculate

```
#EPest_county_estimates1$COMPOUND <- as.numeric(EPest_county_estimates1$COMPOUND) #this should not be n

state.county <- left_join(EPest_county_estimates1, HighEstimate_AgPestUsebyCropGroup, by = c("COMPOUND"
```

join state compound by county compound

```
crop_area_county1$county.FIPS <-as.numeric(crop_area_county1$county.FIPS)
data <- left_join(crop_area_county1, state.county, by = c("county.FIPS" = "COUNTYFPL"))
```

compound m2 crop

```
data$alfalfa.perm2 <- ifelse( data$alfalfa == 0, 0 , format(data$Alfalfa1 * data$EPEST_HIGH_KG / data$al

data$corn.perm2 <- ifelse( data$corn == 0, 0 , format(data$Corn1 * data$EPEST_HIGH_KG/ data$corn, scien

data$othercrops.perm2 <- ifelse( data$othercrops == 0, 0 , format(data$Other_crops1 * data$EPEST_HIGH_K

data$orchardsandgrapes.perm2 <- ifelse( data$orchardsandgrapes == 0, 0 , format(data$Orchards_and_grapes

data$pastureandhay.perm2 <- ifelse( data$pastureandhay == 0, 0 , format(data$Pasture_and_hay1 * data$EPE

data$vegetablesandfruit.perm2 <- ifelse( data$vegetablesandfruit == 0, 0 , format(data$Vegetables_and_fr

data$wheat.perm2 <- ifelse( data$wheat == 0, 0 ,format(data$Wheat1 * data$EPEST_HIGH_KG/ data$wheat, sci
```

final selection

```
data.county <- subset(data, select = c(county.FIPS, COUNTY_NAME, COMPOUND, alfalfa.perm2, corn.perm2, o
colnames(data.county) <- gsub(".perm2", "", colnames(data.county))
```

from wide to long format & create unique ID

```
data.county1 <- gather(data.county, Category, perm2, alfalfa:wheat, factor_key = TRUE)
data.county1$county.FIPS.Category <- paste(data.county1$county.FIPS, data.county1$Category, "")
```

Census tract calculation

create unique ID (i am not sure this is creating a unique ID)

```
crop.category.tract1$county.FIPS.Category <- paste(crop.category.tract1$county.FIPS, crop.category.trac
summary(crop.category.tract1)
```

```
##        DN            OBJECTID          ST               STATE
##  Min.   :  4.00   Min.   :  1.0   Length:1046806     Length:1046806
##  1st Qu.: 36.00   1st Qu.:253.0   Class :character   Class :character
##  Median : 37.00   Median :510.0   Mode  :character   Mode  :character
##  Mean   : 45.35   Mean   :512.7
##  3rd Qu.: 61.00   3rd Qu.:810.0
##  Max.   :216.00   Max.   :994.0
##     COUNTY              FIPS             AREA_SQMI          E_POV150
##  Length:1046806     Length:1046806     Min.   :  50.38   Min.   :  47.0
##  Class :character   Class :character   1st Qu.: 164.65   1st Qu.: 453.0
##  Mode  :character   Mode  :character   Median : 432.78   Median : 611.0
##                                        Mean   :1704.46   Mean   : 713.3
##                                        3rd Qu.:1562.05   3rd Qu.: 883.0
##                                        Max.   :9489.85   Max.   :2206.0
##     M_POV150        E_UNEMP          M_UNEMP          E_NOHSDP
##  Min.   : 56.0   Min.   :  0.00   Min.   :  3.00   Min.   :   0.0
##  1st Qu.:158.0   1st Qu.: 37.00   1st Qu.: 28.00   1st Qu.: 150.0
##  Median :214.0   Median : 66.00   Median : 41.00   Median : 217.0
##  Mean   :238.4   Mean   : 75.87   Mean   : 54.73   Mean   : 296.3
##  3rd Qu.:293.0   3rd Qu.:117.00   3rd Qu.: 71.00   3rd Qu.: 397.0
##  Max.   :932.0   Max.   :548.00   Max.   :366.00   Max.   :1143.0
##     M_NOHSDP        E_AGE65          M_AGE65         E_AGE17
##  Min.   : 11.0   Min.   :  15.0   Min.   : 39.0   Min.   :  37.0
##  1st Qu.: 68.0   1st Qu.: 597.0   1st Qu.:105.0   1st Qu.: 438.0
##  Median : 89.0   Median : 730.0   Median :138.0   Median : 618.0
##  Mean   :105.6   Mean   : 815.4   Mean   :150.1   Mean   : 726.7
##  3rd Qu.:132.0   3rd Qu.: 950.0   3rd Qu.:169.0   3rd Qu.: 862.0
##  Max.   :362.0   Max.   :2021.0   Max.   :684.0   Max.   :1963.0
##     M_AGE17        E_DISABL         M_DISABL        E_SNGPNT
##  Min.   : 54   Min.   :   0.0   Min.   : 13.0   Min.   :  0.00
##  1st Qu.:110   1st Qu.: 396.0   1st Qu.:107.0   1st Qu.: 26.00
##  Median :166   Median : 569.0   Median :143.0   Median : 43.00
##  Mean   :181   Mean   : 582.5   Mean   :154.9   Mean   : 57.21
##  3rd Qu.:236   3rd Qu.: 744.0   3rd Qu.:195.0   3rd Qu.: 88.00
##  Max.   :648   Max.   :1616.0   Max.   :525.0   Max.   :210.00
##     M_SNGPNT          E_HISP           M_HISP          E_LIMENG
##  Min.   :  9.00   Min.   :   0.0   Min.   :  13.0   Min.   :  0.00
##  1st Qu.: 24.00   1st Qu.: 121.0   1st Qu.:  74.0   1st Qu.:  7.00
##  Median : 33.00   Median : 215.0   Median : 122.0   Median : 34.00
##  Mean   : 43.72   Mean   : 458.8   Mean   : 173.7   Mean   : 77.48
##  3rd Qu.: 52.00   3rd Qu.: 587.0   3rd Qu.: 218.0   3rd Qu.: 65.00
##  Max.   :147.00   Max.   :3271.0   Max.   :1011.0   Max.   :965.00
##     M_LIMENG        Shape_Area      county.FIPS         Class_Name
##  Min.   : 47.00   Min.   :0.01529   Length:1046806     Length:1046806
##  1st Qu.: 52.00   1st Qu.:0.04858   Class :character   Class :character
##  Median : 61.00   Median :0.13133   Mode  :character   Mode  :character
##  Mean   : 78.49   Mean   :0.49660
```

```
##  3rd Qu.: 88.00   3rd Qu.:0.45585
##  Max.   :395.00   Max.   :2.73199
##    Category         county.FIPS.Category
##  Length:1046806     Length:1046806
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
```

```
small_crop<-head(crop.category.tract1,1)
small_data<-head(data.county1,1)
#summary(data.county1)
```

```
crop.category.tract2 <- left_join(small_crop, small_data, by = "county.FIPS.Category" )
crop.category.tract2
```

```
##   DN OBJECTID ST  STATE         COUNTY        FIPS AREA_SQMI E_POV150 M_POV150
## 1 71      125 41 Oregon Clatsop County 41007951200  411.0191      837      325
##   E_UNEMP M_UNEMP E_NOHSDP M_NOHSDP E_AGE65 M_AGE65 E_AGE17 M_AGE17 E_DISABL
## 1      55      40      202       93     924     198     662     176      735
##   M_DISABL E_SNGPNT M_SNGPNT E_HISP M_HISP E_LIMENG M_LIMENG Shape_Area
## 1      181       13       15    138    118        7       51  0.1272869
##   county.FIPS.x       Class_Name Category.x county.FIPS.Category county.FIPS.y
## 1         41007 Other Tree Crops othercrops    41007 othercrops             NA
##   COUNTY_NAME COMPOUND Category.y perm2
## 1        <NA>     <NA>       <NA>  <NA>
```

join with county pesticide data

```
colnames(data.county1)
```

```
## [1] "county.FIPS"          "COUNTY_NAME"          "COMPOUND"
## [4] "Category"             "perm2"                "county.FIPS.Category"
```

```
colnames(crop.category.tract1)
```

```
##  [1] "DN"                  "OBJECTID"            "ST"
##  [4] "STATE"               "COUNTY"              "FIPS"
##  [7] "AREA_SQMI"           "E_POV150"            "M_POV150"
## [10] "E_UNEMP"             "M_UNEMP"             "E_NOHSDP"
## [13] "M_NOHSDP"            "E_AGE65"             "M_AGE65"
## [16] "E_AGE17"             "M_AGE17"             "E_DISABL"
## [19] "M_DISABL"            "E_SNGPNT"            "M_SNGPNT"
## [22] "E_HISP"              "M_HISP"              "E_LIMENG"
## [25] "M_LIMENG"            "Shape_Area"          "county.FIPS"
## [28] "Class_Name"          "Category"            "county.FIPS.Category"
```

```
crop.category.tract2 <- left_join(crop.category.tract1, data.county1, by = "county.FIPS.Category" )
```

```
## Warning in left_join(crop.category.tract1, data.county1, by = "county.FIPS.Category"): Detected an u
## i Row 1 of `x` matches multiple rows in `y`.
## i Row 6037 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

check of any row didn't make it to the join »> 0 is good

```
crop.category.tract2.notjoined <- anti_join(crop.category.tract1, data.county1, by = "county.FIPS.Categ
crop.category.tract2.notjoined
```

```
##  [1] DN                  OBJECTID            ST
##  [4] STATE               COUNTY              FIPS
##  [7] AREA_SQMI           E_POV150            M_POV150
## [10] E_UNEMP             M_UNEMP             E_NOHSDP
## [13] M_NOHSDP            E_AGE65             M_AGE65
## [16] E_AGE17             M_AGE17             E_DISABL
## [19] M_DISABL            E_SNGPNT            M_SNGPNT
## [22] E_HISP              M_HISP              E_LIMENG
## [25] M_LIMENG            Shape_Area          county.FIPS
## [28] Class_Name          Category            county.FIPS.Category
## <0 rows> (or 0-length row.names)
```

pesticide

```
crop.category.tract2$pesticide <- as.numeric(crop.category.tract2$Shape_Area)*as.numeric(crop.category.

pesticide.tract <- crop.category.tract2 %>%
  group_by(FIPS) %>%
  summarise(pesticide = format(sum(pesticide), scientific = FALSE))


pesticide.county <- crop.category.tract2 %>%
  group_by(county.FIPS.x) %>% #COMPOUND
  summarise(pesticidewsu = format(sum(pesticide), scientific = FALSE))

pesticide.county$COUNTY_FIPS_CODE <- as.numeric(str_sub(pesticide.county$county.FIPS.x,start = -2))
```

WSU calculated pesticide vs reported pesticide - double checking numbers

```r
sum(as.numeric(pesticide.tract$pesticide), na.rm = T)
```

```
## [1] 1734362
```

```r
sum(as.numeric(pesticide.county$pesticidewsu))
```

```
## [1] 1734362
```

```r
sum(state.county$EPEST_HIGH_KG)
```

```
## [1] 1734823
```

```r
sum(EPest_county_estimates$EPEST_HIGH_KG)
```

```
## [1] 1734823
```

difference — pesticide application data includes pesticides applied to wheat in Mason, Kitsap and Wahkiakum counties but no wheat is grown in these counties

difference is 465 kg of pesticide or 0.01% of pesticide applied to WA — no further adjustments make to correct for this

#This is probably note relevant for oregon.

```r
county.difference <- left_join(pesticide.county, EPest_county_estimates2, by = "COUNTY_FIPS_CODE")
county.difference$pesticide.diff <- format(county.difference$pesticide - as.numeric(county.difference$pe


str(pesticide.county)
```

```
## tibble [36 x 3] (S3: tbl_df/tbl/data.frame)
##  $ county.FIPS.x   : chr [1:36] "41001" "41003" "41005" "41007" ...
##  $ pesticidewsu    : chr [1:36] "31706.95" "27246.52" "24907.87" "1565.597" ...
##  $ COUNTY_FIPS_CODE: num [1:36] 1 3 5 7 9 11 13 15 17 19 ...
```

```r
str(EPest_county_estimates2)
```

```
## tibble [36 x 2] (S3: tbl_df/tbl/data.frame)
##  $ COUNTY_FIPS_CODE: num [1:36] 1 3 5 7 9 11 13 15 17 19 ...
##  $ pesticide       : num [1:36] 31707 27247 24908 1762 8754 ...
```

Final file with unit conversion from kg to lbs & adjustment based on CT area & create percentile

```
pesticide.tract$ID_1 <- as.numeric(pesticide.tract$FIPS )

#changed ID2 to FIPS due to lack of ID2 column
pesticide.tract <- left_join(ct_area, pesticide.tract, by = c("FIPS" = "FIPS"))


pesticide.tract$pesticide_lbs <- as.numeric(pesticide.tract$pesticide) * 2.2046226218
pesticide.tract$pesticide_lbs_mile2 <- pesticide.tract$pesticide_lbs / pesticide.tract$AREA

#changed ID2 to FIPS due to lack of ID2 column
pesticide.tract <- subset(pesticide.tract, select = c(FIPS, pesticide_lbs, pesticide_lbs_mile2))
pesticide.tract[is.na(pesticide.tract)] <- 0


pesticide.tract$pesticide_lbs_mile2_percentile <- ntile(pesticide.tract$pesticide_lbs_mile2, 10)

#write.csv(pesticide.tract, "Final.csv")
```

Include pesticides that are harmful

read pesticide inclusion file

```
#excel_sheets("OREGON/OR_Crops.xlsx")
#Hazardous_pesticide <- read_excel("OREGON/OR_Crops.xlsx", sheet = "Hazardous Pesticide")
Hazardous_pesticide <- read.csv("OREGON/hazardpest.csv")


Hazardous_pesticide <- subset(Hazardous_pesticide, select = c(COMPOUND, Include))
```

shapefile with selected pesticides

```
crop.category.tract3 <- left_join(Hazardous_pesticide, crop.category.tract2, by ="COMPOUND")
```

```
crop.category.tract3$perm2 = as.numeric(crop.category.tract3$perm2)
crop.category.tract3$pesticide = as.numeric(crop.category.tract3$pesticide)
crop.category.tract3 <- crop.category.tract3 %>%
  group_by(COUNTY,ST, E_POV150, FIPS, M_POV150, E_UNEMP, M_UNEMP, E_NOHSDP, M_NOHSDP, E_AGE65, M_AGE65,
  mutate(
    perm2 = sum(perm2, na.rm = TRUE),
    pesticide = sum(pesticide, na.rm = TRUE)) %>%
      select(COUNTY, ST, E_POV150, FIPS, M_POV150, E_UNEMP, M_UNEMP, E_NOHSDP, M_NOHSDP, E_AGE65, M_AGE
  distinct()
```

our state plots

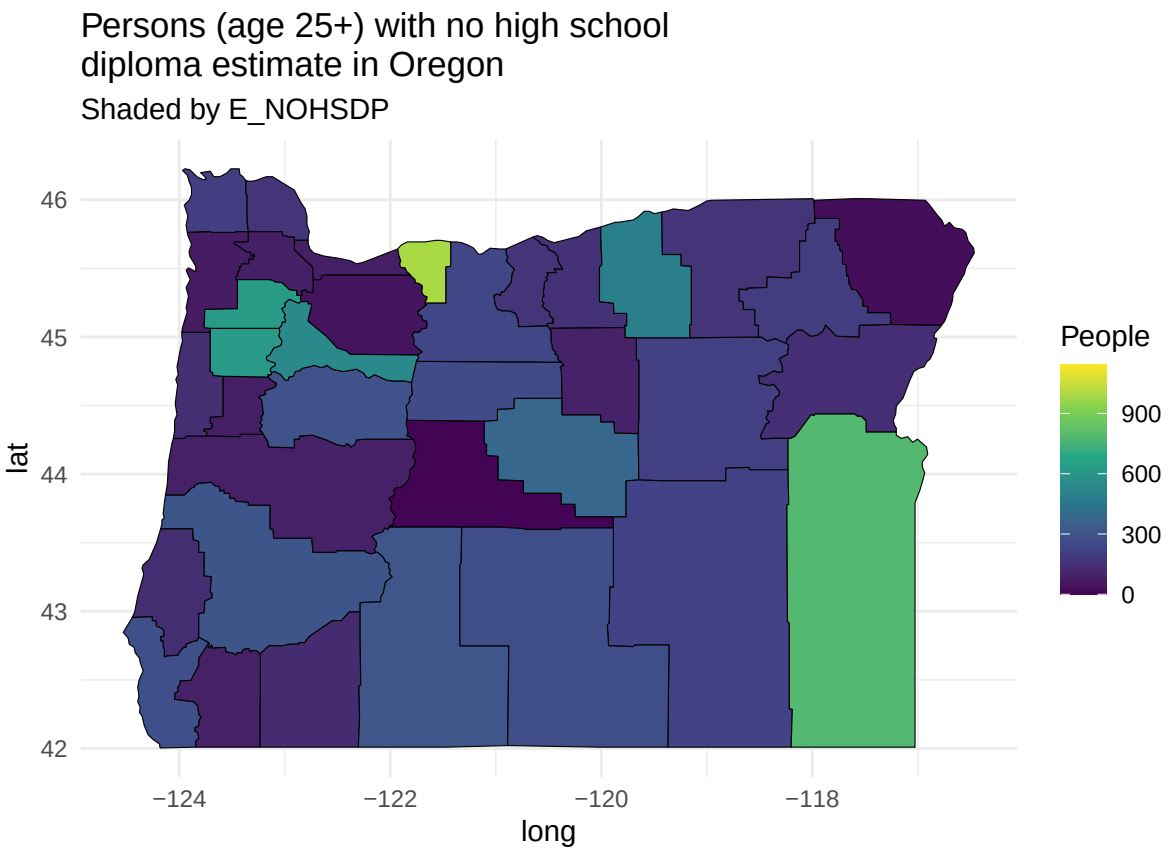
```
crop.category.tract3 <- crop.category.tract3 %>%
  mutate(county = tolower(gsub(" County", "", COUNTY)))
crop.category.tract3
```

```
## # A tibble: 174 x 24
## # Groups:   COUNTY, ST, E_POV150, FIPS, M_POV150, E_UNEMP, M_UNEMP, E_NOHSDP,
## #   M_NOHSDP, E_AGE65, M_AGE65, E_AGE17, M_AGE17, E_DISABL, M_DISABL, E_SNGPNT,
## #   M_SNGPNT, E_HISP, M_HISP, E_LIMENG, M_LIMENG [174]
##    COUNTY        ST    E_POV150 FIPS  M_POV150 E_UNEMP M_UNEMP E_NOHSDP M_NOHSDP
##    <chr>         <chr>    <int> <chr>    <int>   <int>   <int>    <int>    <int>
##  1 Columbia Cou~ 41         699 4100~      235     109      98      165      112
##  2 Columbia Cou~ 41         339 4100~      126     117      52      176       86
##  3 Columbia Cou~ 41         595 4100~      223     129      74      305      109
##  4 Umatilla Cou~ 41         495 4105~      277      92      72      165       80
##  5 Umatilla Cou~ 41         889 4105~      314     118      68      397      132
##  6 Umatilla Cou~ 41        1539 4105~      520      68      57      487      180
##  7 Wallowa Coun~ 41         304 4106~      117      41      27       30       21
##  8 Wallowa Coun~ 41         342 4106~       99      40      24      155       67
##  9 Columbia Cou~ 41         864 4100~      308     212     102      311      152
## 10 Morrow County 41        1145 4104~      332     122      86      491      120
## # i 164 more rows
## # i 15 more variables: E_AGE65 <int>, M_AGE65 <int>, E_AGE17 <int>,
## #   M_AGE17 <int>, E_DISABL <int>, M_DISABL <int>, E_SNGPNT <int>,
## #   M_SNGPNT <int>, E_HISP <int>, M_HISP <int>, E_LIMENG <int>, M_LIMENG <int>,
## #   pesticide <dbl>, perm2 <dbl>, county <chr>
```

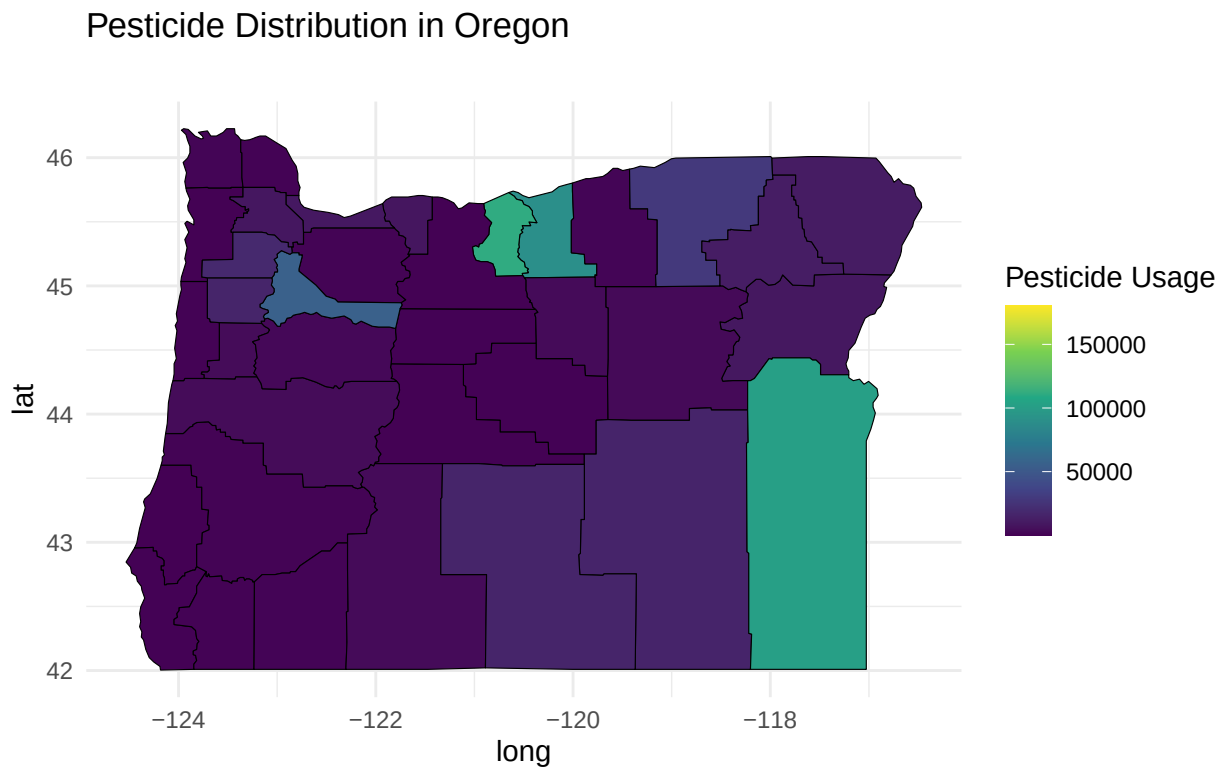
```
oregon_map <- map_data("county") %>%
  filter(region == "oregon")%>%
  left_join(crop.category.tract3, by = c("subregion" = "county"))
```

```
## Warning in left_join(., crop.category.tract3, by = c(subregion = "county")): Detected an unexpected m
## i Row 1 of `x` matches multiple rows in `y`.
## i Row 48 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

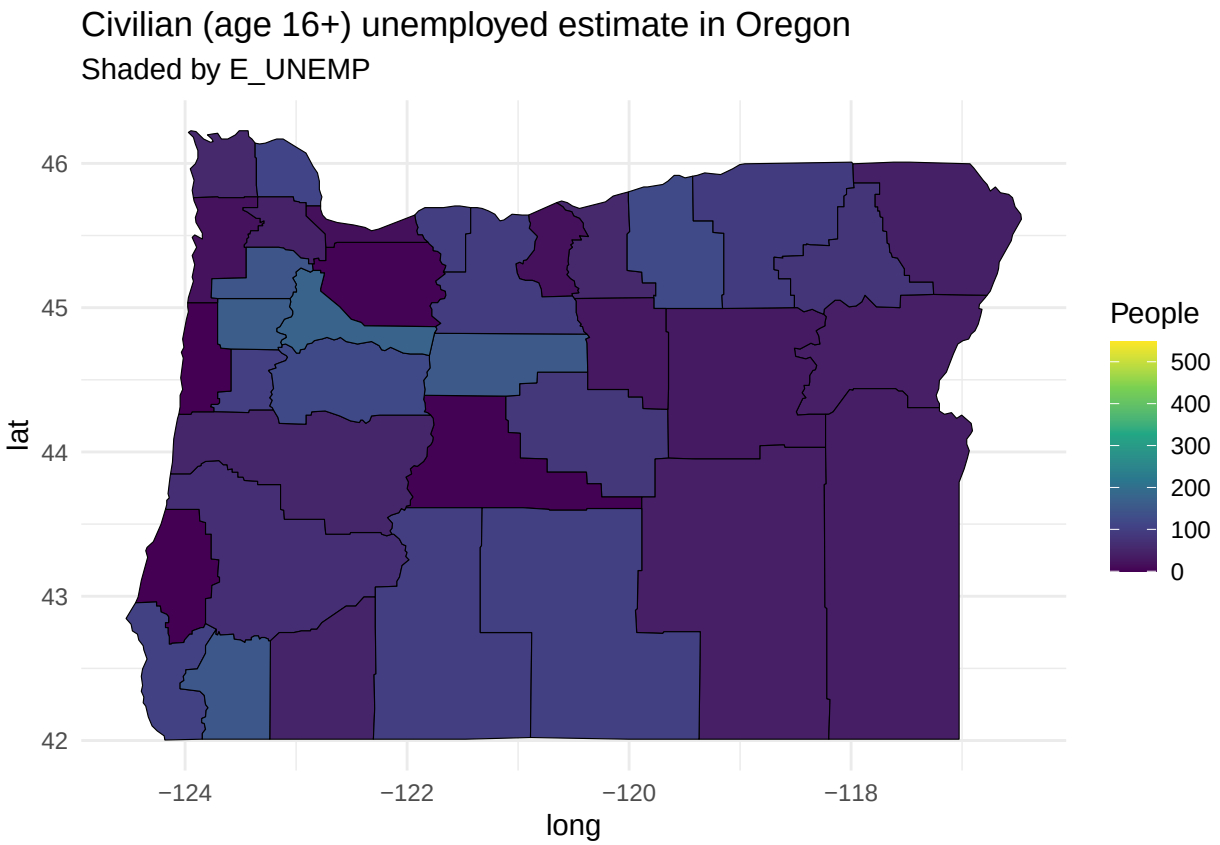
```
plot <- ggplot(data = oregon_map, aes(long, lat, group = group, fill = E_NOHSDP)) +
  geom_polygon(color = "black", linewidth = 0.2) +
  coord_fixed(1.3) +
  scale_fill_viridis_c(name = "People", option = "viridis", na.value = "gray90") +
  theme_minimal() +
  labs(title = "Persons (age 25+) with no high school
diploma estimate in Oregon",
       subtitle = "Shaded by E_NOHSDP")
plot
```



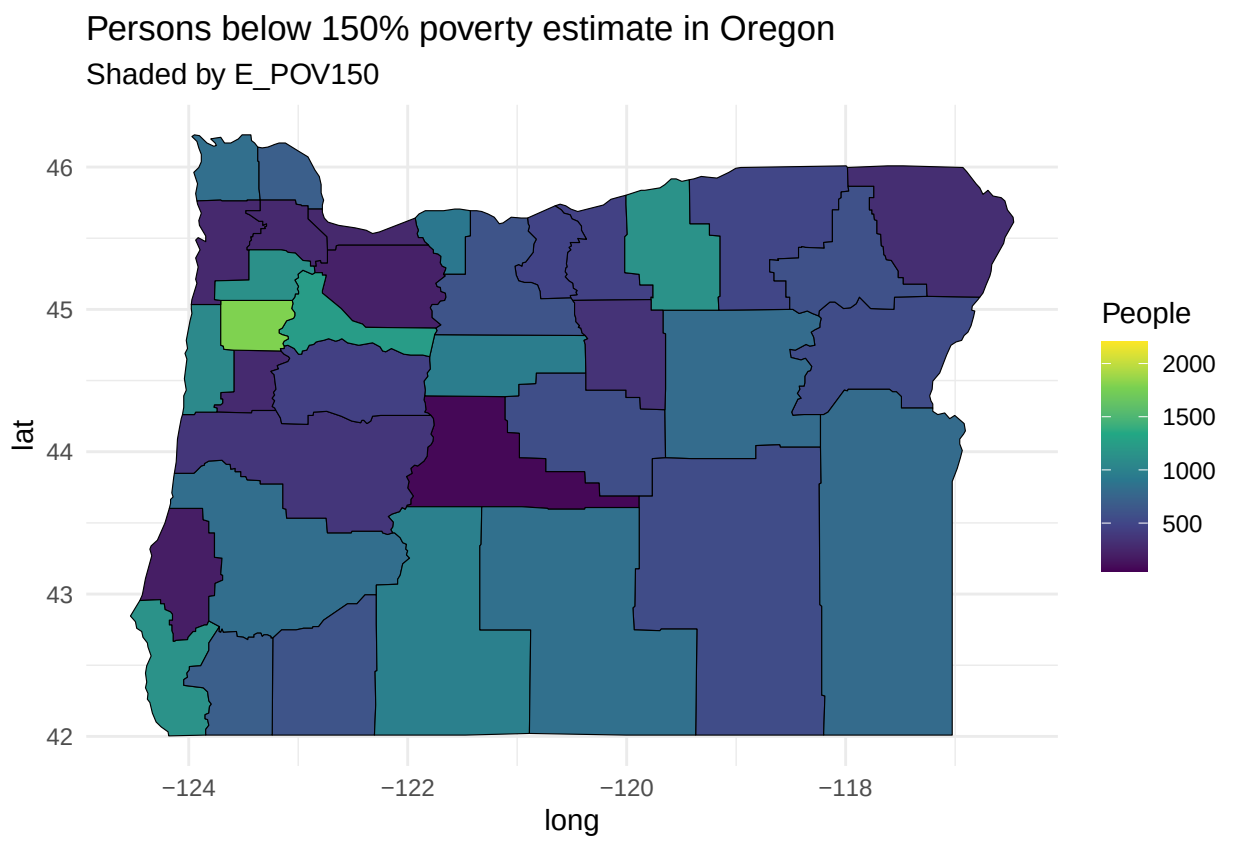

```
plot <- ggplot(data = oregon_map, aes(long, lat, group = group, fill = pesticide)) +
  geom_polygon(color = "black", linewidth = 0.2) +
  coord_fixed(1.3) +
  scale_fill_viridis_c(name = "Pesticide Usage", option = "viridis", na.value = "gray90") +
  theme_minimal() +
  labs(title = "Pesticide Distribution in Oregon",
       subtitle = "")
plot
```



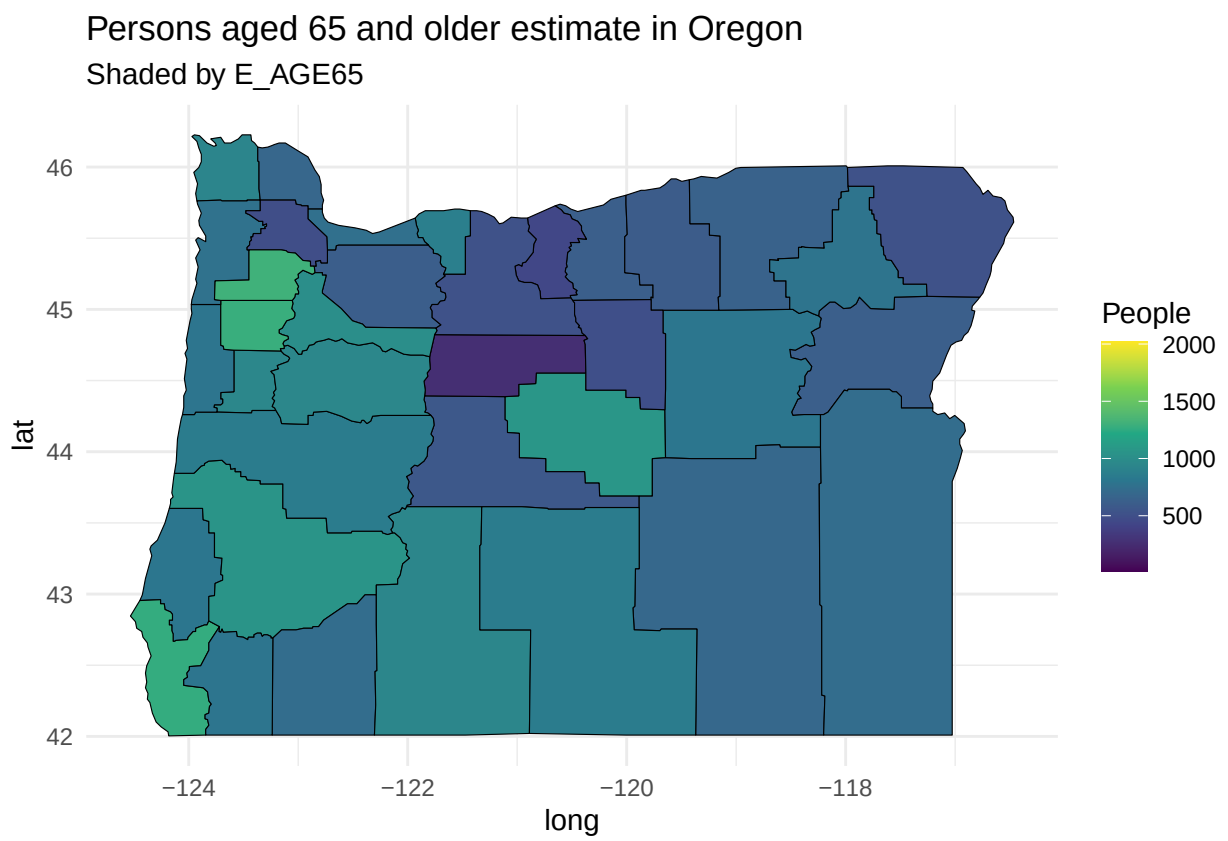

```
plot <- ggplot(data = oregon_map, aes(long, lat, group = group, fill = E_UNEMP)) +
  geom_polygon(color = "black", linewidth = 0.2) +
  coord_fixed(1.3) +
  scale_fill_viridis_c(name = "People", option = "viridis", na.value = "gray90") +
  theme_minimal() +
  labs(title = "Civilian (age 16+) unemployed estimate in Oregon",
       subtitle = "Shaded by E_UNEMP")
plot
```



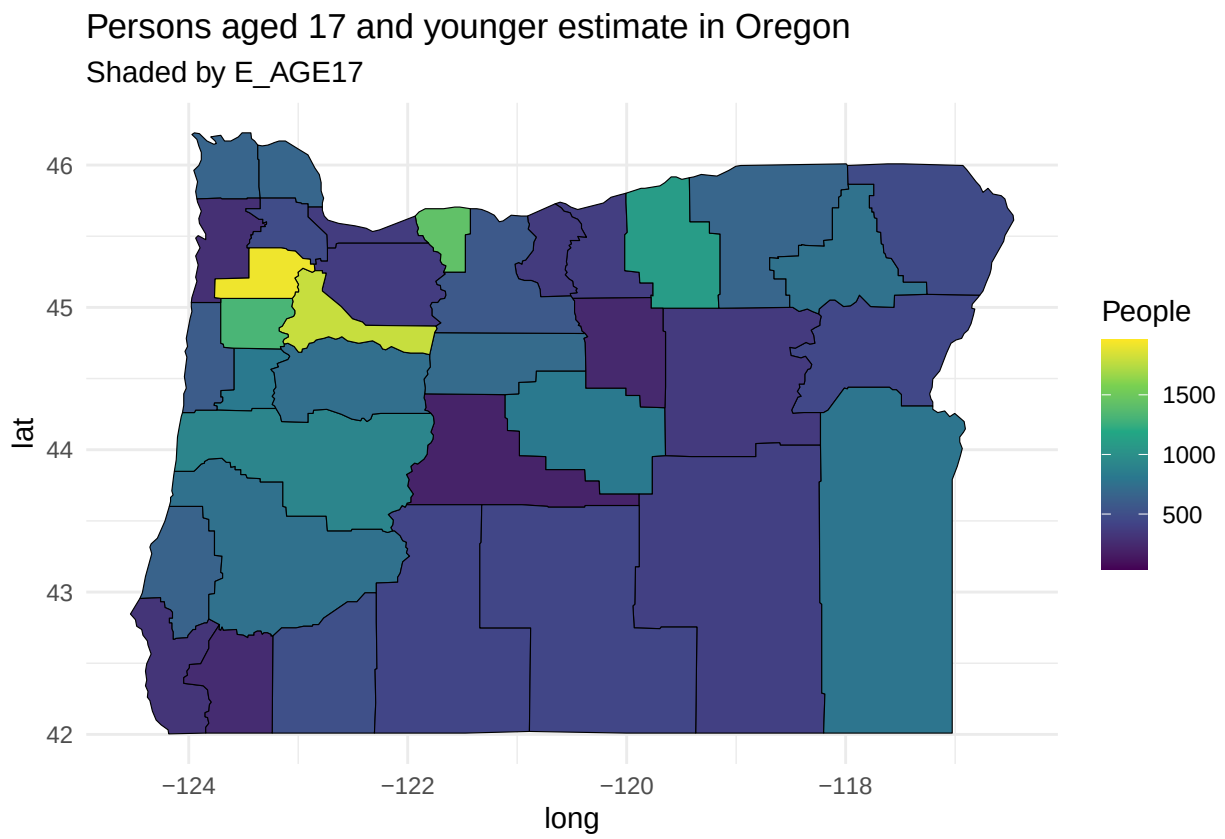

```
plot <- ggplot(data = oregon_map, aes(long, lat, group = group, fill = E_POV150)) +
  geom_polygon(color = "black", linewidth = 0.2) +
  coord_fixed(1.3) +
  scale_fill_viridis_c(name = "People", option = "viridis", na.value = "gray90") +
  theme_minimal() +
  labs(title = "Persons below 150% poverty estimate in Oregon",
       subtitle = "Shaded by E_POV150")
plot
```



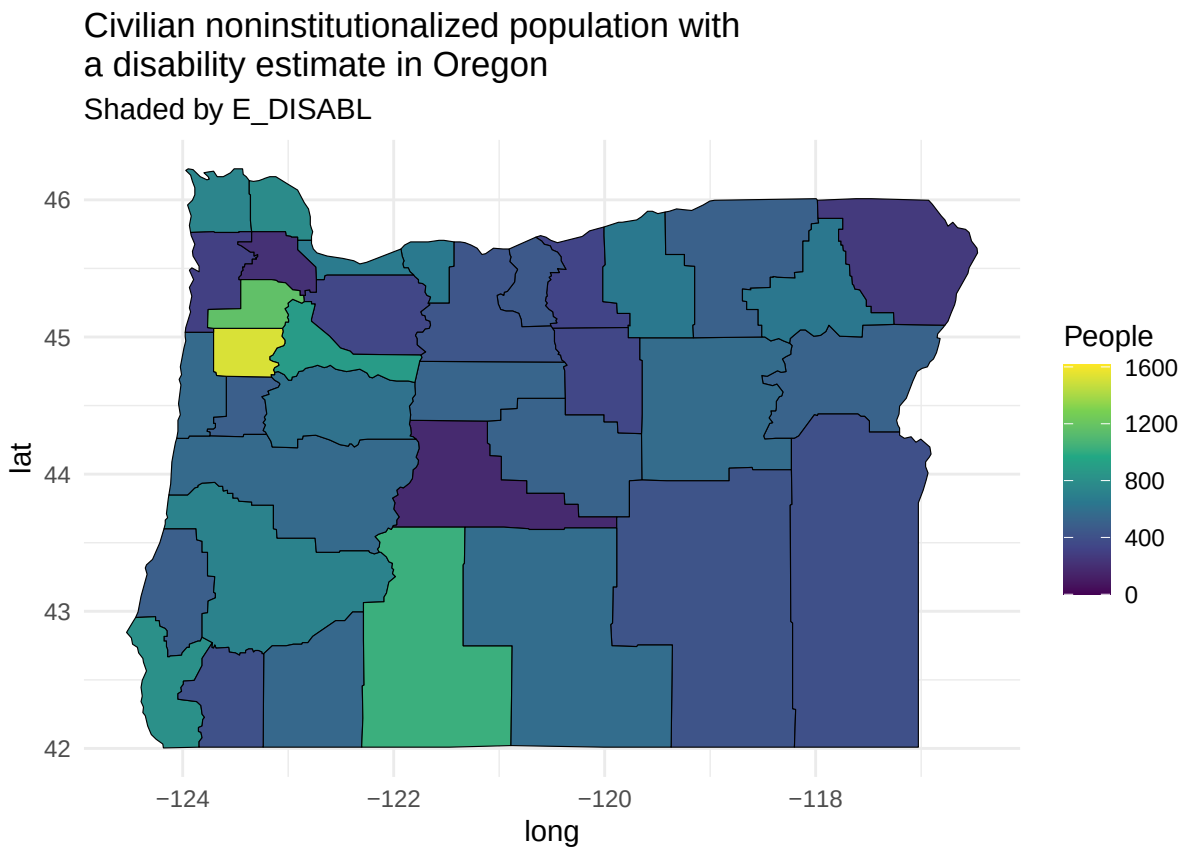

```
plot <- ggplot(data = oregon_map, aes(long, lat, group = group, fill = E_AGE65)) +
  geom_polygon(color = "black", linewidth = 0.2) +
  coord_fixed(1.3) +
  scale_fill_viridis_c(name = "People", option = "viridis", na.value = "gray90") +
  theme_minimal() +
  labs(title = "Persons aged 65 and older estimate in Oregon",
       subtitle = "Shaded by E_AGE65")
plot
```



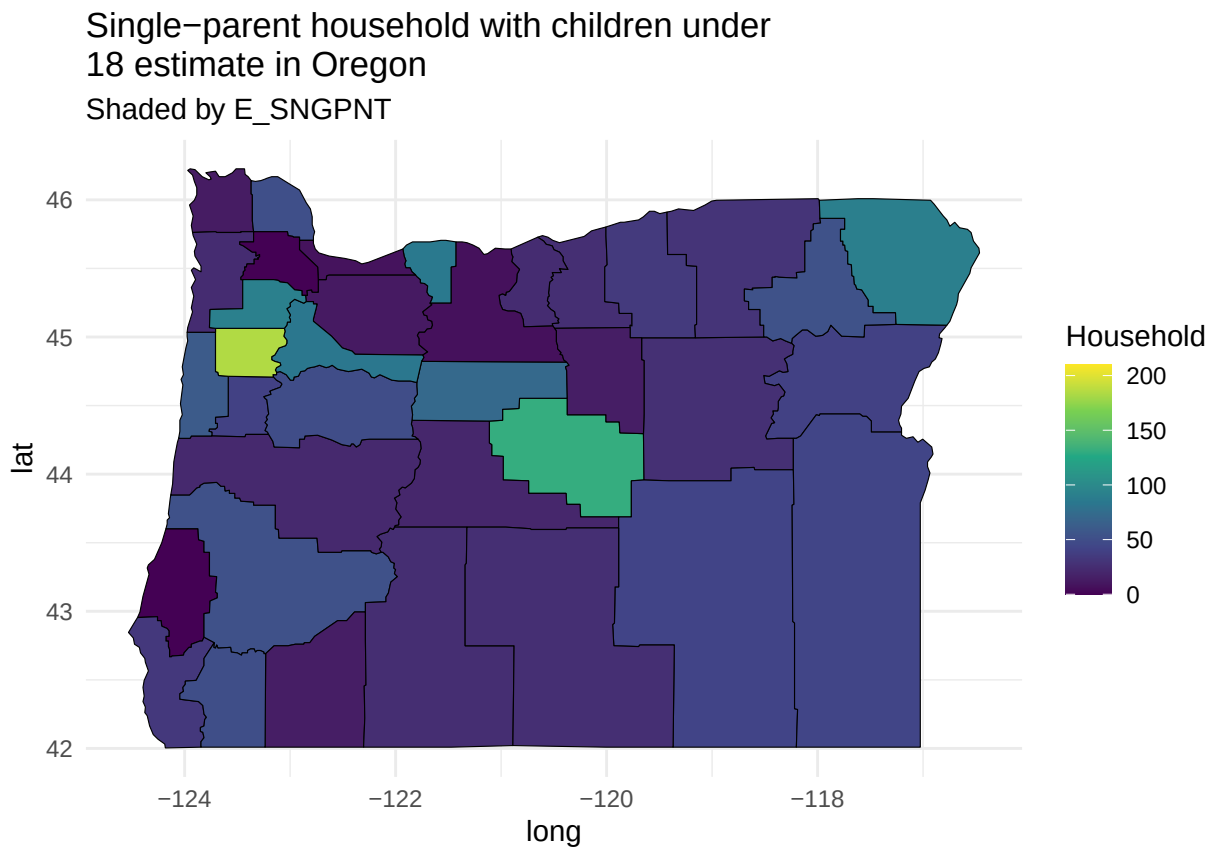

```
plot <- ggplot(data = oregon_map, aes(long, lat, group = group, fill = E_AGE17)) +
  geom_polygon(color = "black", linewidth = 0.2) +
  coord_fixed(1.3) +
  scale_fill_viridis_c(name = "People", option = "viridis", na.value = "gray90") +
  theme_minimal() +
  labs(title = "Persons aged 17 and younger estimate in Oregon",
       subtitle = "Shaded by E_AGE17")
plot
```



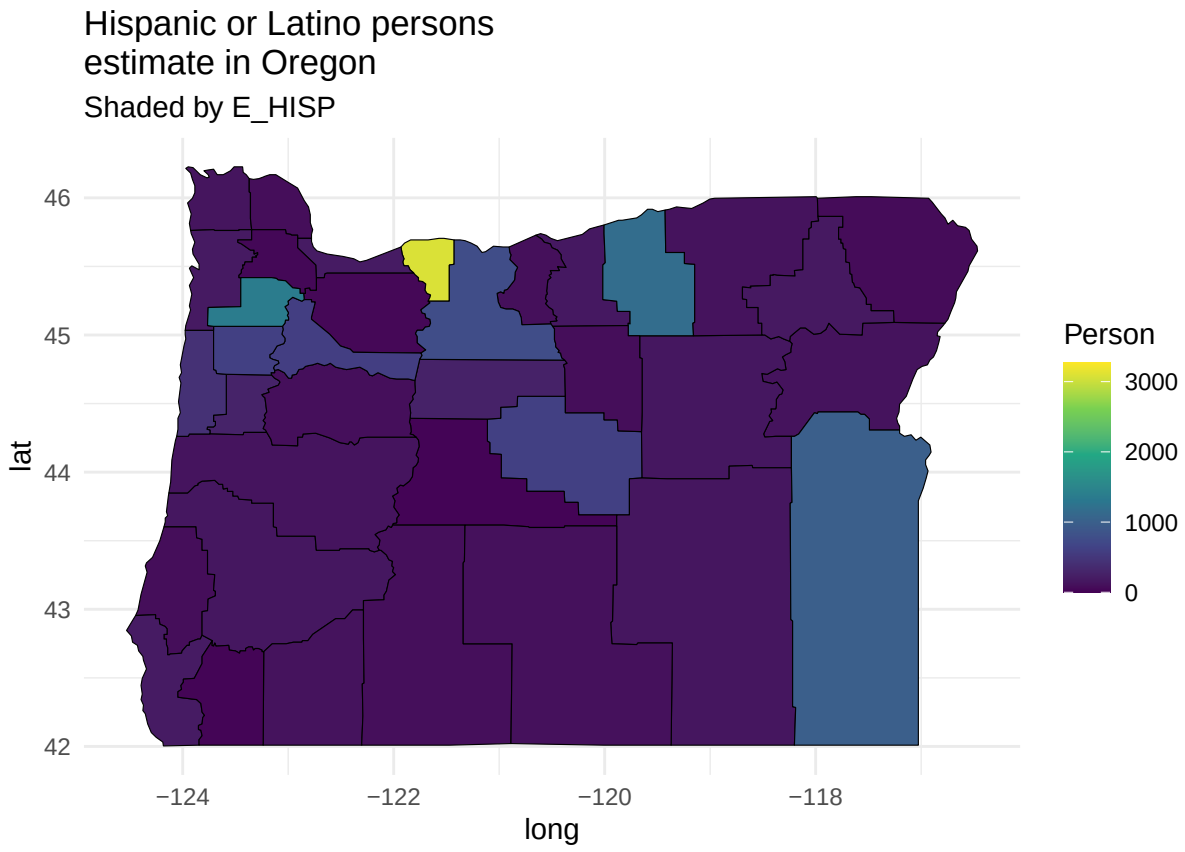

```
plot <- ggplot(data = oregon_map, aes(long, lat, group = group, fill = E_DISABL)) +
  geom_polygon(color = "black", linewidth = 0.2) +
  coord_fixed(1.3) +
  scale_fill_viridis_c(name = "People", option = "viridis", na.value = "gray90") +
  theme_minimal() +
  labs(title = "Civilian noninstitutionalized population with
a disability estimate in Oregon",
       subtitle = "Shaded by E_DISABL")
plot
```



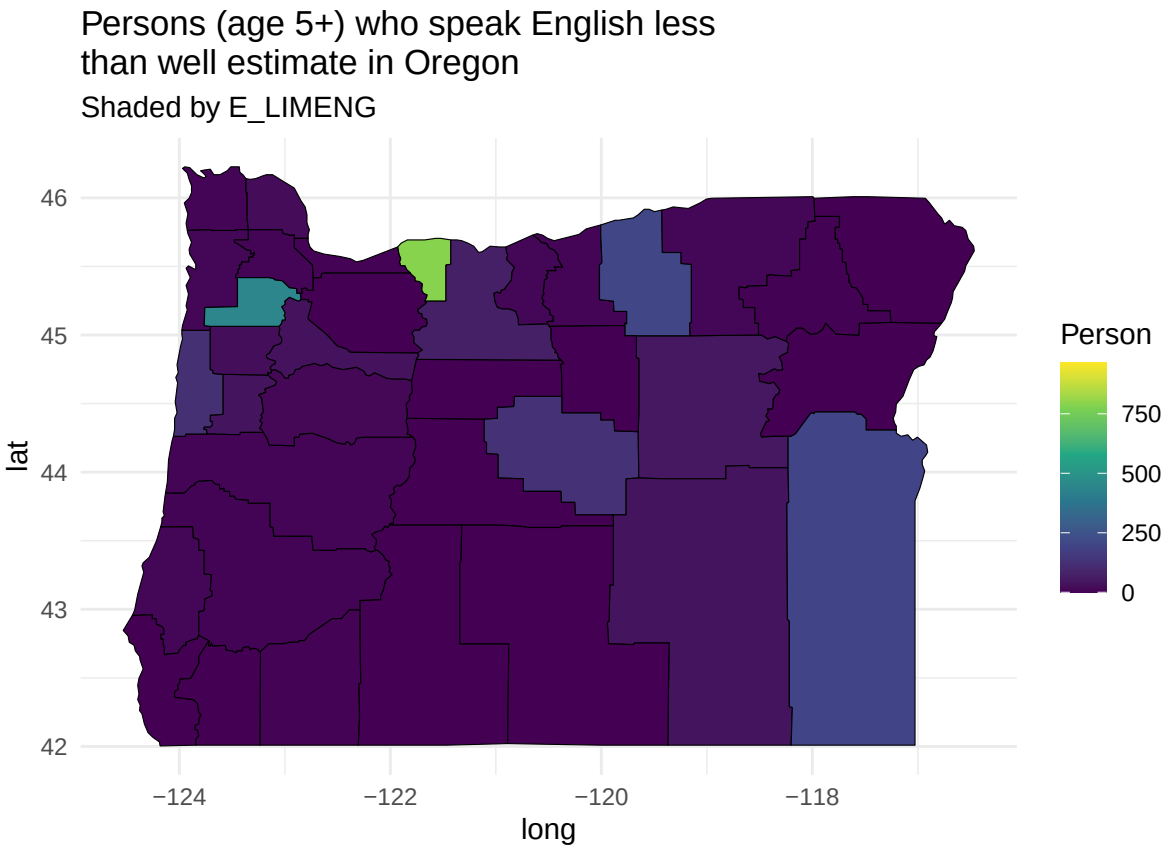

```
plot <- ggplot(data = oregon_map, aes(long, lat, group = group, fill = E_SNGPNT)) +
  geom_polygon(color = "black", linewidth = 0.2) +
  coord_fixed(1.3) +
  scale_fill_viridis_c(name = "Household", option = "viridis", na.value = "gray90") +
  theme_minimal() +
  labs(title = "Single-parent household with children under
18 estimate in Oregon",
       subtitle = "Shaded by E_SNGPNT")
plot
```




```
plot <- ggplot(data = oregon_map, aes(long, lat, group = group, fill = E_HISP)) +
  geom_polygon(color = "black", linewidth = 0.2) +
  coord_fixed(1.3) +
  scale_fill_viridis_c(name = "Person", option = "viridis", na.value = "gray90") +
  theme_minimal() +
  labs(title = "Hispanic or Latino persons
estimate in Oregon",
       subtitle = "Shaded by E_HISP")
plot
```




```
plot <- ggplot(data = oregon_map, aes(long, lat, group = group, fill = E_LIMENG)) +
  geom_polygon(color = "black", linewidth = 0.2) +
  coord_fixed(1.3) +
  scale_fill_viridis_c(name = "Person", option = "viridis", na.value = "gray90") +
  theme_minimal() +
  labs(title = "Persons (age 5+) who speak English less
than well estimate in Oregon",
       subtitle = "Shaded by E_LIMENG")
plot
```


Persons (age 5+) who speak English less than well estimate in Oregon
Shaded by E_LIMENG
46
45
44
43
42
lat
−124
−122
−120
−118
long
Person
750
500
250
0