Comparing PSM products at regional resolution

D G Rossiter

[d.g.rossiter@cornell.edu](mailto:d.g.rossiter@cornell.edu)

12-November-2020

Table of Contents

# Introduction

This script compares PSM products at 250 m grid resolution, which is used by SoilGrids250. We consider this an appropriate resolution for regional studies.

Depending on the [property of interest](#voi), the following can be compared:

* [SoiLGrids250](https://www.isric.org/explore/soilgrids) from ISRIC further abbreviated as *SG2*;
* [Global Soil Map v0.5 for the USA](https://www.nrcs.usda.gov/wps/portal/nrcs/detail/soils/research/?cid=nrcseprd1321715), further abbreviated as *GSMv05*;
* [Intermediate-scale gridded soil property and interpretation maps from averaged and aggregated SSURGO and STATSGO data](https://github.com/ncss-tech/ISSR-800), further abbreviated as *ISSR-800*;
* POLARIS Soil Properties;
* [Soil Properties and Class 100m Grids USA](https://doi.org/10.18113/S1KW2H), further abbreviated as *SPCG100USA*;
* [LandGIS](https://opengeohub.org/about-landgis) from the private company [EnvirometriX](http://envirometrix.nl/).

The PSM products must have been previously imported and restricted to the same area of interest (AOI), typically , to the locations indicated in the directory list.

We use GSM v0.5 as the reference map, since it is based directly on field survey (SSURGO and STATSGO). So this script requires, at a minimum, SoilGrids250 and GSM v0.5 products to have been imported.

This script must follow the import of the various products; these are in directory ../scripts\_importmaps.

To use this script:

1. Ajust the [directory structure](#dirs) to your system
2. [Select a property](#voi) and [select a depth slice](#depth).
3. [Select an Area of Interest](#aoi).
4. Either compile to HTML or PDF (“knit”), or “Run All” within R Markdown.
5. Generated figures will be in directory ./figs/compare\_sg/.
6. Generated tables in format will be in directory ../LaTeX\_tables.
7. Generated harmonized maps will be [saved](#save) to directory PSM\_compare under the base directory

# Setup

## Packages

library(rgdal) # R interface to GDAL  
library(terra) # for raster maps  
library(sf) # Simple Features spatial data  
# library(gridExtra) # arrange multiple plots  
library(knitr) # for fancy tables  
library(xtable) # (same)

## Base directory paths

Set base directories, specific to the local file system.

1. base.dir: This is the location of the PSM tiles that have been cropped to an AOI by an import script.

base.dir <- "/Users/rossiter/ds/"  
base.dir.sg <- paste0(base.dir, "SoilGrids250")  
base.dir.gsm <- paste0(base.dir, "GSM\_USA")  
base.dir.issr8 <- paste0(base.dir, "ISSR8")  
base.dir.polaris <- paste0(base.dir, "POLARIS")  
base.dir.psu <- paste0(base.dir, "SPCG100USA")  
base.dir.landgis <- paste0(base.dir, "LandGIS")

1. base.dir.import: This is where downloaded large GeoTIFF are located. Because of their size they may be on a separate file system, e.g., removable or networked drive. Files may have been downloaded here by an import script, or by direct download from the data provider.

In this script this location is only used for POLARIS, since they are imported directly as tiles by script ./scripts\_importmaps/POLARIS\_import.Rmd.

base.dir.import <- "/Volumes/Pythagoras/ds/"  
base.dir.polaris.import <- paste0(base.dir.import, "POLARIS")

1. base.dir.export: This is where generated harmonized maps will be saved for further processing, e.g., comparing patterns.

base.dir.export <- paste0(base.dir, "Compare\_PSM")

# Property of interest

Property names in various systems. Note that all except GSM v0.5 and SoilGrids250 are missing one or more properties. Such products will be omitted from comparisons for those properties.

Properties that be compared are: clay, silt, sand weight concentrations; pH in 1:1 water; CEC; SOC; bulk density of the fine earth; coarse fragment volume proportion. See the import script for each product for a link to a description of the properties and their units of measure.

voi.list.sg <- c("clay", "silt", "sand", "phh2o", "cec", "soc", "bdod", "cfvo")  
voi.list.gsm <- c("claytotal\_r\_g\_kg", "silttotal\_r\_g\_kg", "sandtotal\_r\_g\_kg",   
 "ph1to1h2o\_r\_ions\_pHx10", "ecec\_r\_cmolc\_kg", "soc\_r\_mr\_g\_gF",  
 "dbthirdbar\_lt2mm\_r\_g\_cm3", "gravel\_r\_vol\_ratio\_m3\_m3")  
voi.list.issr8 <- c("clay", "silt","sand","ph", "cec", "", "", "")  
voi.list.polaris <- c("clay", "silt", "sand", "ph", "", "om", "bd", "")   
voi.list.psu <- c("clay", "", "sand", "ph\_h2o", "", "soc", "bd", "")  
voi.list.landgis <- c("clay.wfraction\_usda.3a1a1a",  
 "silt.wfraction\_usda.3a1a1a",  
 "sand.wfraction\_usda.3a1a1a",  
 "ph.h2o\_usda.4c1a2a",  
 "",  
 "organic.carbon\_usda.6a1c",  
 "bulkdens.fineearth\_usda.4a1h",  
 "coarsefrag.vfraction\_usda\_3b1")

Select the position in these lists

voi <- 4  
voi.sg <- voi.list.sg[voi]  
voi.issr8 <- voi.list.issr8[voi]  
voi.polaris <- voi.list.polaris[voi]  
voi.psu <- voi.list.psu[voi]  
voi.gsm <- voi.list.gsm[voi]

# Depth of interest

Depth slices:

depth.list.sg <- c("0-5", "5-15", "15-30", "30-60", "60-100", "100-200")  
# SPCGUSA100 predicts at points, these were averaged to GSM slices during import  
# --- these have the SG names  
# LandGIS predicts at points, these were averaged to GSM slices during import  
# --- these have the SG names  
depth.list.polaris <- gsub("-", "\_", depth.list.sg)  
depth.list.issr8 <- gsub("-", "", depth.list.sg)  
depth.list.gsm <- c("000\_005", "005\_015", "015\_030", "030\_060", "060\_100", "100\_200")

Select the depth slice:

depth <- 1

# Area of Interest (AOI)

We use a tile, because that is how POLARIS data is served. Specify the lower-right corner, then compute the upper-right corner 1 degree west and north.

Specify the two corners using WGS84 geographic coordinates:

Specify the lower-right corner:

tile.lrc <- c(-76, 42) # lower-right corner  
tile.size <- 1

Compute the upper-left coner:

tile.ulc <- c(tile.lrc[1]-tile.size, tile.lrc[2]+tile.size) # upper-left corner

A prefix for directories and file names, to keep AOI results separate.

AOI.dir.prefix <- paste0("lat", tile.lrc[2], tile.ulc[2],  
 "\_lon", tile.ulc[1], tile.lrc[1])

Bounding box:

m <- matrix(c(tile.ulc[1],tile.lrc[1], #ulc  
 tile.ulc[2], tile.lrc[2]), nrow=2) #lrc  
bb.ll <- st\_sfc(st\_multipoint(m))  
st\_crs(bb.ll) <- 4326 # ESPG code for WGS84 long/lat

Project the bounding box to Goode Interrupted Homolosine (IGH) used by SoilGrids:

# convert to Homolosine. Note epsg=152160 is not in PROJ4 database  
crs.igh <- '+proj=igh +lat\_0=0 +lon\_0=0 +datum=WGS84 +units=m +no\_defs'  
(bb.igh <- st\_transform(bb.ll, crs.igh))

## Geometry set for 1 feature   
## geometry type: MULTIPOINT  
## dimension: XY  
## bbox: xmin: -9234503 ymin: 4674913 xmax: -9132054 ymax: 4785101  
## CRS: +proj=igh +lat\_0=0 +lon\_0=0 +datum=WGS84 +units=m +no\_defs

(bb.igh <- st\_coordinates(bb.igh)[,1:2])

## X Y  
## [1,] -9234503 4785101  
## [2,] -9132054 4674913

(bb <- as.vector(t(bb.igh)))

## [1] -9234503 4785101 -9132054 4674913

Project the bounding box to the AEA. This is the CRS used by ISSR-800 and SPCG100USA:

crs.aea <- "+proj=aea +lat\_0=23 +lon\_0=-96 +lat\_1=29.5 +lat\_2=45.5 +x\_0=0 +y\_0=0 +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +units=m +no\_defs"  
(bb.aea <- st\_transform(bb.ll, crs.aea))

## Geometry set for 1 feature   
## geometry type: MULTIPOINT  
## dimension: XY  
## bbox: xmin: 1530465 ymin: 2283406 xmax: 1633206 ymax: 2376207  
## CRS: +proj=aea +lat\_0=23 +lon\_0=-96 +lat\_1=29.5 +lat\_2=45.5 +x\_0=0 +y\_0=0 +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +units=m +no\_defs

# Products to compare

Load the tiles in their native CRS, as processed in the import scripts.

## SoilGrids250

This is required; all the properties to compare are mapped by SoilGrids250.

# SoilGrids250 -- only the mean prediction in this script  
# Use the EPSG:4326 version  
src.dir <- paste0(base.dir.sg ,"/",   
 AOI.dir.prefix, "/",   
 voi.sg, "/mean/",  
 depth.list.sg[depth], "cm")  
(voi.depth.name <- paste0(voi.sg, "\_", depth.list.sg[depth], "cm\_mean\_4326"))

## [1] "phh2o\_0-5cm\_mean\_4326"

(file.name <- paste0(src.dir, "/", voi.depth.name, '.tif'))

## [1] "/Users/rossiter/ds/SoilGrids250/lat4243\_lon-77-76/phh2o/mean/0-5cm/phh2o\_0-5cm\_mean\_4326.tif"

if (file.exists(file.name)) {  
 r.sg <- terra::rast(file.name)  
 names(r.sg) <- "SoilGrids250"  
 print(r.sg)  
} else { stop("No SoilGrids250 tile, stopping") }

## class : SpatRaster   
## dimensions : 426, 626, 1 (nrow, ncol, nlyr)  
## resolution : 0.002349893, 0.002349893 (x, y)  
## extent : -77.23224, -75.76121, 42.00031, 43.00136 (xmin, xmax, ymin, ymax)  
## coord. ref. : +proj=longlat +datum=WGS84 +no\_defs   
## data source : phh2o\_0-5cm\_mean\_4326.tif   
## names : SoilGrids250

## GSM

# GSM  
src.dir <- paste0(base.dir.gsm ,"/", AOI.dir.prefix)  
voi.depth.name <- paste0(voi.gsm, "\_", depth.list.gsm[depth])  
(file.name <- paste0(src.dir, "/GSM\_mu\_", voi.depth.name, ".tif"))

## [1] "/Users/rossiter/ds/GSM\_USA/lat4243\_lon-77-76/GSM\_mu\_ph1to1h2o\_r\_ions\_pHx10\_000\_005.tif"

if (file.exists(file.name)) {  
 r.gsm <- terra::rast(file.name)  
 names(r.gsm) <- "GSMv0.5"  
 print(r.gsm)  
} else { stop("No GSM v0.5 tile, stopping") }

## class : SpatRaster   
## dimensions : 1200, 1200, 1 (nrow, ncol, nlyr)  
## resolution : 0.0008333333, 0.0008333333 (x, y)  
## extent : -77.00042, -76.00042, 41.99958, 42.99958 (xmin, xmax, ymin, ymax)  
## coord. ref. : +proj=longlat +datum=WGS84 +no\_defs   
## data source : GSM\_mu\_ph1to1h2o\_r\_ions\_pHx10\_000\_005.tif   
## names : GSMv0.5   
## min values : 41.99483   
## max values : 75

## SPCG100USA

# SPCG100USA  
src.dir <- paste0(base.dir.psu ,"/", AOI.dir.prefix)  
(file.name <- paste0(src.dir, "/", voi.psu, "\_", depth.list.sg[depth], '.tif'))

## [1] "/Users/rossiter/ds/SPCG100USA/lat4243\_lon-77-76/ph\_h2o\_0-5.tif"

if (file.exists(file.name)) {  
 r.psu <- terra::rast(file.name)  
 names(r.psu) <- "spcg100usa"  
 print(r.psu)  
}

## class : SpatRaster   
## dimensions : 928, 1027, 1 (nrow, ncol, nlyr)  
## resolution : 100, 100 (x, y)  
## extent : 1530500, 1633200, 2283400, 2376200 (xmin, xmax, ymin, ymax)  
## coord. ref. : +proj=aea +lat\_0=23 +lon\_0=-96 +lat\_1=29.5 +lat\_2=45.5 +x\_0=0 +y\_0=0 +datum=NAD83 +units=m +no\_defs   
## data source : ph\_h2o\_0-5.tif   
## names : spcg100usa   
## min values : 41   
## max values : 72.5

## POLARIS

# POLARIS -- only the mean prediction in this script  
(file.name <- paste0(base.dir.polaris.import, "/",  
 AOI.dir.prefix, "/",  
 voi.list.polaris[voi], "/mean/",  
 depth.list.polaris[depth], "/",  
 AOI.dir.prefix, ".tif"))

## [1] "/Volumes/Pythagoras/ds/POLARIS/lat4243\_lon-77-76/ph/mean/0\_5/lat4243\_lon-77-76.tif"

if (file.exists(file.name)) {  
 r.p <- terra::rast(file.name)  
 names(r.p) <- "polaris"  
 print(r.p)  
}

## class : SpatRaster   
## dimensions : 3600, 3600, 1 (nrow, ncol, nlyr)  
## resolution : 0.0002777778, 0.0002777778 (x, y)  
## extent : -77, -76, 42, 43 (xmin, xmax, ymin, ymax)  
## coord. ref. : +proj=longlat +datum=WGS84 +no\_defs   
## data source : lat4243\_lon-77-76.tif   
## names : polaris

## LandGIS

# LandGIS -- only the mean prediction in this script  
(file.name <- paste0(base.dir.landgis, "/",  
 AOI.dir.prefix, "/",  
 voi.list.landgis[voi], "\_",  
 depth.list.sg[depth],  
 ".tif"))

## [1] "/Users/rossiter/ds/LandGIS/lat4243\_lon-77-76/ph.h2o\_usda.4c1a2a\_0-5.tif"

if (file.exists(file.name)) {  
 r.landgis <- terra::rast(file.name)  
 names(r.landgis) <- "landgis"  
 print(r.landgis)  
}

## class : SpatRaster   
## dimensions : 480, 480, 1 (nrow, ncol, nlyr)  
## resolution : 0.002083333, 0.002083333 (x, y)  
## extent : -77.00002, -76.00002, 41.99917, 42.99917 (xmin, xmax, ymin, ymax)  
## coord. ref. : +proj=longlat +datum=WGS84 +no\_defs   
## data source : ph.h2o\_usda.4c1a2a\_0-5.tif   
## names : landgis   
## min values : 40   
## max values : 74

## ISSR-800

(file.name <- paste0(base.dir.issr8, "/",  
 AOI.dir.prefix, "/",  
 voi.list.issr8[voi], "\_",  
 depth.list.issr8[depth],  
 ".tif"))

## [1] "/Users/rossiter/ds/ISSR8/lat4243\_lon-77-76/ph\_05.tif"

if (file.exists(file.name)) {  
 r.issr8 <- terra::rast(file.name)  
 names(r.issr8) <- "ISSR-800"  
 print(r.issr8)  
}

## class : SpatRaster   
## dimensions : 116, 128, 1 (nrow, ncol, nlyr)  
## resolution : 800, 800 (x, y)  
## extent : 1530800, 1633200, 2283600, 2376400 (xmin, xmax, ymin, ymax)  
## coord. ref. : +proj=aea +lat\_0=23 +lon\_0=-96 +lat\_1=29.5 +lat\_2=45.5 +x\_0=0 +y\_0=0 +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +units=m +no\_defs   
## data source : ph\_05.tif   
## names : ISSR-800   
## min values : 4.68522   
## max values : 7.163799

# Make the units compatible

Depending on the property, data in some coverages need to be converted to the units used in SoilGrids250; we choose this as the base; note these are integers. Here are the units:

df <- data.frame(property=voi.list.sg,   
 #"clay" "silt" "sand" "phh2o" "cec" "soc" "bdod" "cfvo"   
 gsm=c("%%","%%","%%","pHx10","cmol(c)/kg","g/gF","Mg/m3", "m3/m3"), # GSM  
 sg=c("%%","%%","%%","pHx10","mmol(c)/kg","dg/kg","cg/cm3", "cm3/dm3"), #SG  
 p=c("%","%", "%","pH","", "log10(%)", "g/cm3", ""), # POLARIS  
 spcg=c("%","%","%","pHx10", "", "%","g/cm3", ""), # SPCG100USA  
 lgis=c("%","%","%","pHx10","","5g/Kg","10 kg/m3", "%"), # LandGIS  
 issr=c("%","%","%","pH","cmol(+)/kg","","g/cm3", "%") # ISSR-800  
 )  
knitr::kable(  
 df, caption = 'Properties and units of measure',  
 col.names=c("Property", "GlobalSoilMap","SoilGrids","POLARIS", "SPCG100USA", "LandGIS", "ISSR-800"),  
 booktabs = TRUE)

Properties and units of measure

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Property | GlobalSoilMap | SoilGrids | POLARIS | SPCG100USA | LandGIS | ISSR-800 |
| clay | %% | %% | % | % | % | % |
| silt | %% | %% | % | % | % | % |
| sand | %% | %% | % | % | % | % |
| phh2o | pHx10 | pHx10 | pH | pHx10 | pHx10 | pH |
| cec | cmol(c)/kg | mmol(c)/kg |  |  |  | cmol(+)/kg |
| soc | g/gF | dg/kg | log10(%) | % | 5g/Kg |  |
| bdod | Mg/m3 | cg/cm3 | g/cm3 | g/cm3 | 10 kg/m3 | g/cm3 |
| cfvo | m3/m3 | cm3/dm3 |  |  | % | % |

Make a matrix with the conversions to SoilGrids250 units. These factors *multiply* the source, to match SoilGrids250. NA values indicate that the property is not included in the source.

Some conversions are given [here](https://www.isric.org/explore/soilgrids/faq-soilgrids#What_do_the_filename_codes_mean).

conversions <- data.frame(property=voi.list.sg,   
 # sg=c("%%","%%","%%","pHx10","mmol(c)/kg","dg/kg","cg/cm3", "cm3/dm3"), #SG  
 gsm=c(1, 1, 1, 1 , 10, 0.01, 0.01, 0.1), # GSM  
 p=c(10, 10, 10, 10, NA, NA, 100, NA), # POLARIS -- SOM is special case  
 spcg=c(10, 10, 10, 1, NA, 1, 100, NA), # SPCG100USA  
 lgis=c(10, 10, 10, 1, NA, 20, 0.1, 0.1), # LandGIS  
 issr=c(10, 10, 10, 10, 10, NA, 1, 0.1) # ISSR-800  
 )  
knitr::kable(  
 conversions, caption = 'Conversion factors, multiply by these to match SoilGrids250',  
 col.names=c("Property","GlobalSoilMap", "POLARIS", "SPCG100USA", "LandGIS", "ISSR-800"),  
 booktabs = TRUE)

Conversion factors, multiply by these to match SoilGrids250

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Property | GlobalSoilMap | POLARIS | SPCG100USA | LandGIS | ISSR-800 |
| clay | 1.00 | 10 | 10 | 10.0 | 10.0 |
| silt | 1.00 | 10 | 10 | 10.0 | 10.0 |
| sand | 1.00 | 10 | 10 | 10.0 | 10.0 |
| phh2o | 1.00 | 10 | 1 | 1.0 | 10.0 |
| cec | 10.00 | NA | NA | NA | 10.0 |
| soc | 0.01 | NA | 1 | 20.0 | NA |
| bdod | 0.01 | 100 | 100 | 0.1 | 1.0 |
| cfvo | 0.10 | NA | NA | 0.1 | 0.1 |

Convert units as necessary:

(factors <- conversions[match(voi.sg, conversions$property),])

## property gsm p spcg lgis issr  
## 4 phh2o 1 10 1 1 10

# GSM  
fact <- as.numeric(factors[2])  
if (!is.na(fact) && (fact != 1)) { r.gsm <- r.gsm\*fact }  
# POLARIS  
if (exists("r.p")) {  
 fact <- as.numeric(factors[3])  
 if (!is.na(fact) && (fact != 1)) { r.p <- r.p\*fact }  
}  
# SPCG100USA  
if (exists("r.psu")) {  
 fact <- as.numeric(factors[4])  
 if (!is.na(fact) && (fact != 1)) { r.psu <- r.psu\*fact }  
}  
# LandGIS  
if (exists("r.landgis")) {  
 fact <- as.numeric(factors[5])  
 if (!is.na(fact) && (fact != 1)) { r.landgis <- r.landgis\*fact }  
}  
# ISSR-800  
if (exists("r.issr8")) {  
 fact <- as.numeric(factors[6])  
 if (!is.na(fact) && (fact != 1)) { r.issr8 <- r.issr8\*fact }  
}

Note that SOC for POLARIS is a special case, because of the log10-scale, and because it is SOM, not SOC. Use the conventional conversion factor 0.58 = 1/1.724138.

if (exists("r.p") && (voi.sg=="soc")) {  
 r.p <- ((10^r.p)\*0.58\*1000)   
}

# Match resolution and CRS

SoilGrids250, GSM v0.5, POLARIS and LandGIS are in EPSG:4326 (WGS84 long/lat), at different grid resolutions.

rgdal::showP4(crs(r.sg))

## [1] "+proj=longlat +datum=WGS84 +no\_defs"

data.frame(sg=res(r.sg)[1],   
 polaris=ifelse(exists("r.p"), res(r.p)[1], ""),  
 gsm=res(r.gsm)[1],  
 landgis=ifelse(exists("r.landgis"), res(r.landgis)[1], ""))

## sg polaris gsm landgis  
## 1 0.002349893 0.0002777778 0.0008333333 0.002083333

ISSR-800 and SPCG100USA are in a CONUS Albers Equal Area, but with slightly different definitions of the datums, also with different resolutions (800 and 100 m, respectively). The GRS80 ellipsoid is the basis of the NAD83 datum, and centred with WGS84, so there effectively no difference.

if (exists("r.psu")) rgdal::showP4(crs(r.psu))

## [1] "+proj=aea +lat\_0=23 +lon\_0=-96 +lat\_1=29.5 +lat\_2=45.5 +x\_0=0 +y\_0=0 +datum=NAD83 +units=m +no\_defs"

if (exists("r.issr8")) rgdal::showP4(crs(r.issr8))

## [1] "+proj=aea +lat\_0=23 +lon\_0=-96 +lat\_1=29.5 +lat\_2=45.5 +x\_0=0 +y\_0=0 +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +units=m +no\_defs"

Resample products into SoilGrids250 WGS84, using cubic interpolation.

r.gsm.sg <- terra::resample(r.gsm, r.sg)  
if (exists("r.p")) { r.p.sg <- terra::resample(r.p, r.sg) }  
if (exists("r.psu")) { r.psu.sg <- terra::resample(r.psu, r.sg) }  
if (exists("r.landgis")) { r.landgis.sg <- terra::resample(r.landgis, r.sg) }  
if (exists("r.issr8")) { r.issr8.sg <- terra::resample(r.issr8, r.sg) }

# Make all maps cover the same area

POLARIS predicts in the lakes, for example.

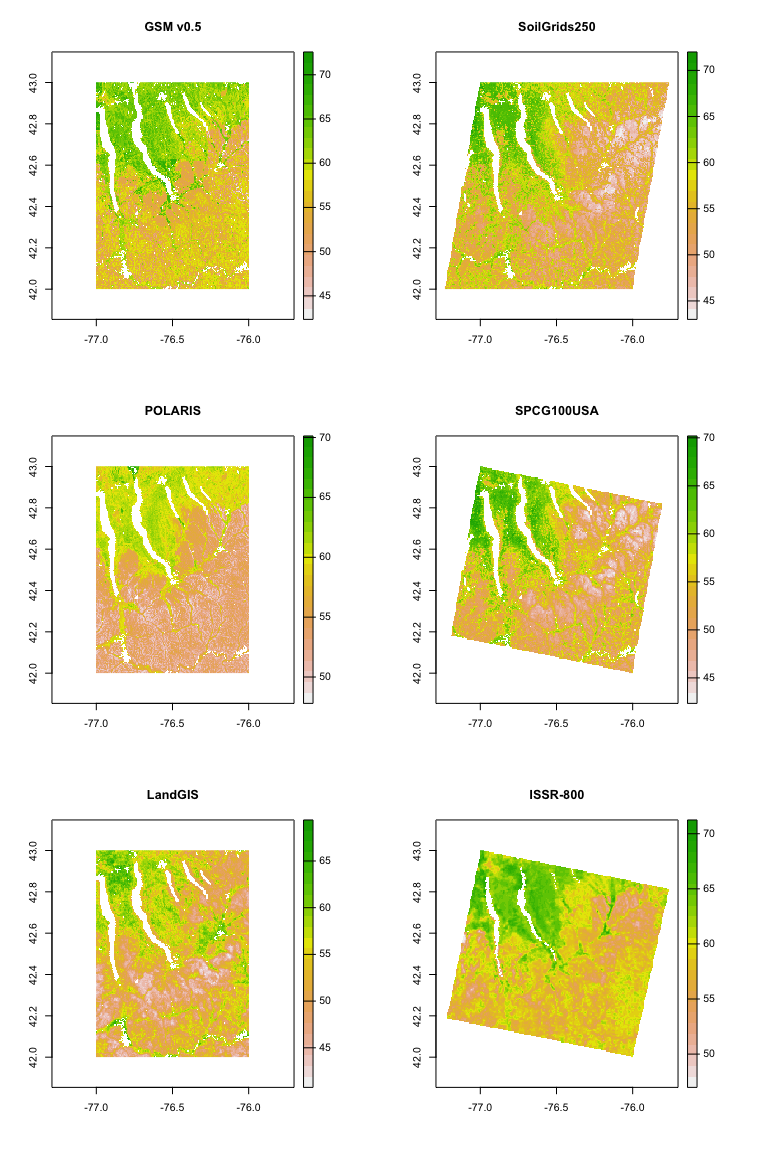
if (exists("r.p.sg")) { r.p.sg <- mask(r.p.sg, r.sg) }

Mask from SoilGrids to get consistent coverge:

r.gsm.sg <- mask(r.gsm.sg, r.sg)  
if (exists("r.issr.sg")) { r.issr.sg <- mask(r.issr8.sg, r.sg) }  
if (exists("r.psu.sg")) { r.psu.sg <- mask(r.psu.sg, r.sg) }  
if (exists("r.landgis.sg")) { r.landgis.sg <- mask(r.landgis.sg, r.sg) }

Check that the maps cover the same area, ignore different stretches for now:

par(mfrow=c(3, 2))  
plot(r.gsm.sg, main="GSM v0.5")  
plot(r.sg, main="SoilGrids250")  
if (exists("r.p.sg")) { plot(r.p.sg, main="POLARIS") }  
if (exists("r.psu.sg")) { plot(r.psu.sg, main="SPCG100USA") }  
if (exists("r.landgis.sg")) { plot(r.landgis.sg, main="LandGIS") }  
if (exists("r.issr8.sg")) { plot(r.issr8.sg, main="ISSR-800") }



par(mfrow=c(1, 1))

# Compare

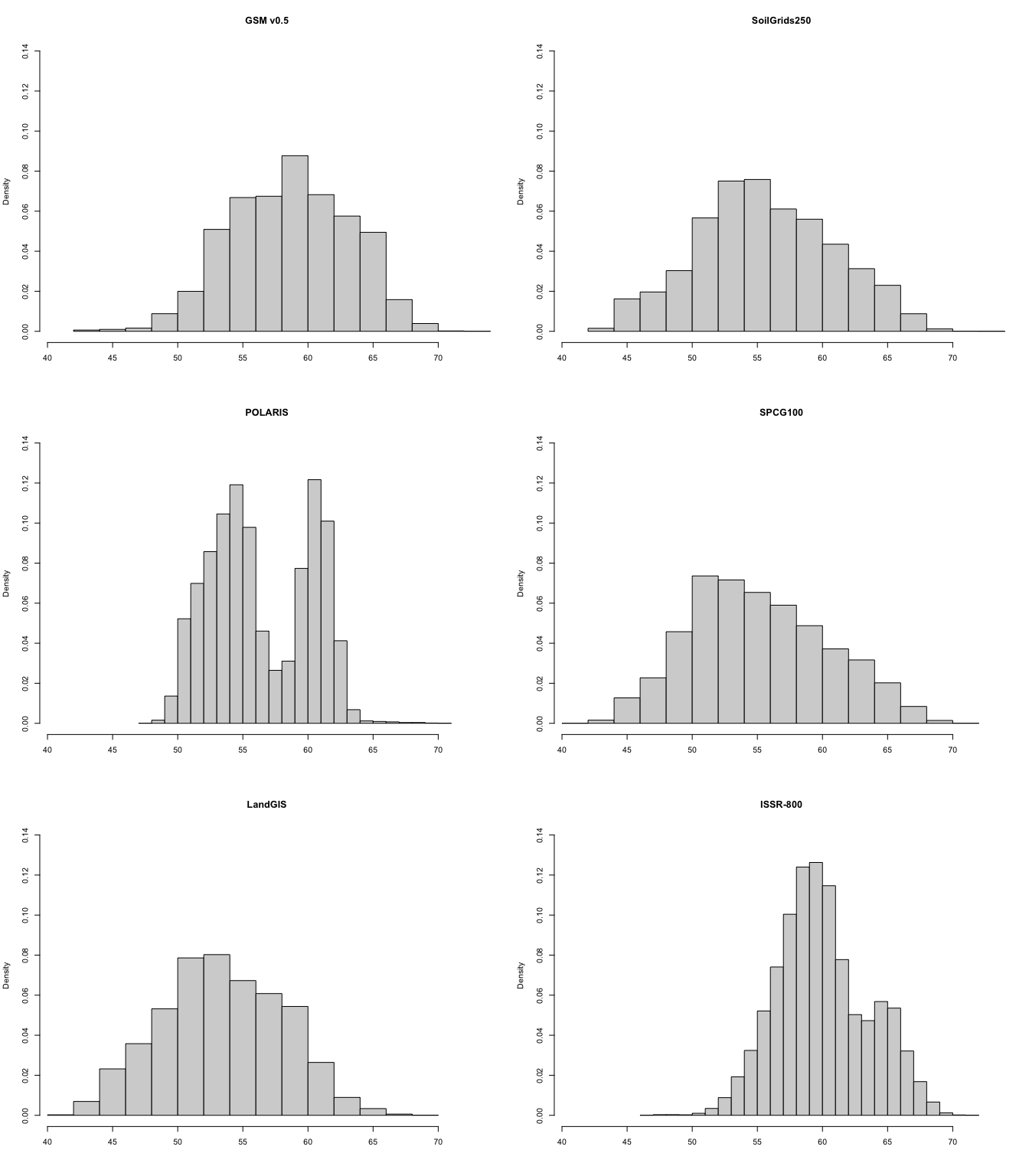
## Compute common range for all products

zlim <- c(min(values(r.sg)\*10, na.rm = TRUE),  
 max(values(r.sg)\*10, na.rm = TRUE))/10  
zlim <- c(floor(min(zlim[1]\*10, values(r.gsm.sg)\*10, na.rm=TRUE)),  
 ceiling(max(zlim[2]\*10, values(r.gsm.sg)\*10, na.rm=TRUE)))/10  
if (exists("r.p.sg")) {  
 zlim <- c(floor(min(zlim[1]\*10, values(r.p.sg)\*10, na.rm=TRUE)),  
 ceiling(max(zlim[2]\*10, values(r.p.sg)\*10, na.rm=TRUE)))/10  
}  
if (exists("r.psu.sg")) {  
 zlim <- c(floor(min(zlim[1]\*10, values(r.psu.sg)\*10, na.rm=TRUE)),  
 ceiling(max(zlim[2]\*10, values(r.psu.sg)\*10, na.rm=TRUE)))/10  
}  
if (exists("r.landgis.sg")) {  
 zlim <- c(floor(min(zlim[1]\*10, values(r.landgis.sg)\*10, na.rm=TRUE)),  
 ceiling(max(zlim[2]\*10, values(r.landgis.sg)\*10, na.rm=TRUE)))/10  
}  
if (exists("r.issr8.sg")) {  
 zlim <- c(floor(min(zlim[1]\*10, values(r.issr8.sg)\*10, na.rm=TRUE)),  
 ceiling(max(zlim[2]\*10, values(r.issr8.sg)\*10, na.rm=TRUE)))/10  
}

## Histograms

The property and depth will be given in the caption, if this is used in a publication.

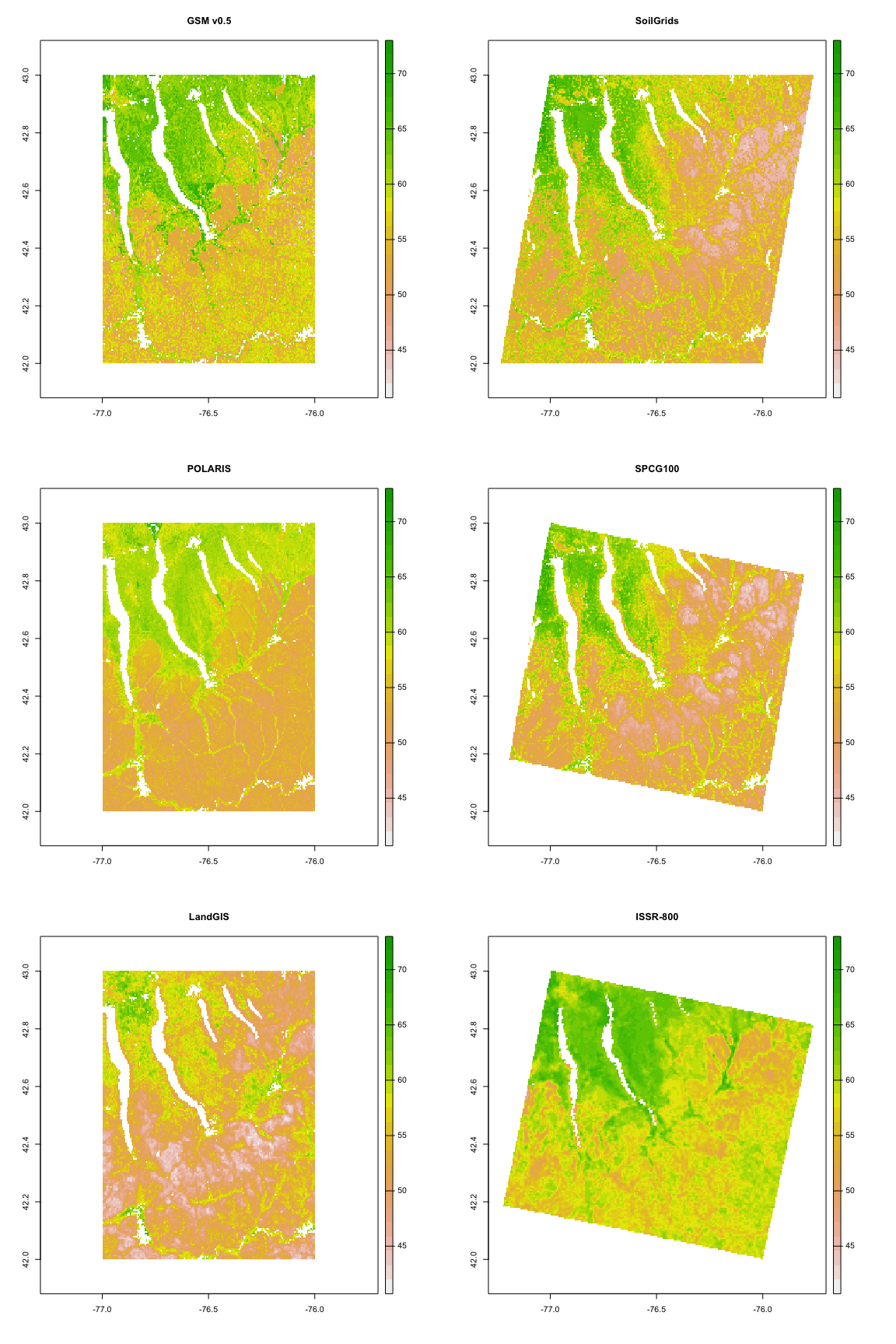
yl <- c(0, 0.14) # standardize the density axis  
par(mfrow=c(3,2))  
hist(r.gsm.sg, main="GSM v0.5",  
 xlim=zlim, xlab="", freq = FALSE, ylim=yl)  
hist(r.sg, main="SoilGrids250",  
 xlim=zlim, xlab="", freq = FALSE, ylim=yl)  
if (exists("r.p.sg")) {  
 hist(r.p.sg, main="POLARIS",  
 xlim=zlim, xlab="", freq = FALSE, ylim=yl)  
}  
if (exists("r.psu.sg")) {  
 hist(r.psu.sg, main="SPCG100",  
 xlim=zlim, xlab="", freq = FALSE, ylim=yl)  
}  
if (exists("r.landgis.sg")) {  
 hist(r.landgis.sg, main="LandGIS",  
 xlim=zlim, xlab="", freq = FALSE, ylim=yl)  
}  
if (exists("r.issr8.sg")) {  
 hist(r.issr8.sg, main="ISSR-800",  
 xlim=zlim, xlab="", freq = FALSE, ylim=yl)  
}



par(mfrow=c(1,1))

## Maps

par(mfrow=c(3,2))  
terra::plot(r.gsm.sg, main="GSM v0.5", range=zlim)  
terra::plot(r.sg, main="SoilGrids", range=zlim)  
if (exists("r.p.sg")) {  
 terra::plot(r.p.sg, main="POLARIS", range=zlim)  
}  
if (exists("r.psu.sg")) {  
 terra::plot(r.psu.sg, main="SPCG100", range=zlim)  
}  
if (exists("r.landgis.sg")) {  
 terra::plot(r.landgis.sg, range=zlim, main="LandGIS")  
}  
if (exists("r.issr8.sg")) {  
 terra::plot(r.issr8.sg, range=zlim, main="ISSR-800")  
}



par(mfrow=c(1,1))

# Differences

## Compute all differences

Relative to GSM v0.5, as the base product closest to the field soil survey.

diff.gsm.sg <- r.gsm.sg - r.sg  
if (exists("r.p.sg")) {  
 diff.gsm.p <- r.gsm.sg - r.p.sg }  
if (exists("r.psu.sg")) {  
 diff.gsm.psu <- r.gsm.sg - r.psu.sg }  
if (exists("r.landgis.sg")) {  
 diff.gsm.landgis <- r.gsm.sg - r.landgis.sg }  
if (exists("r.issr8.sg")) {  
 diff.gsm.issr8 <- r.gsm.sg - r.issr8.sg }

## Statistics

RMSE, ME, RMSE adjusted to ME, for all products compared to GSM v0.5:

stats.compare <- data.frame(PSM\_product = "", MD = 0, RMSD = 0, RMSD.Adjusted = 0)  
rmse <- function(v1, v2) {  
 round(sqrt(mean((v1-v2)^2, na.rm=TRUE)),3)  
}  
me <- function(v1, v2) {   
 round(mean(v1-v2, na.rm=TRUE), 3)  
}  
rmse.adj <- function(v1, v2) { # RMSE adjusted for ME (bias)  
 me <- mean(v1-v2, na.rm=TRUE)  
 v2.adj <- v2 + me  
 round(sqrt(mean((v1-v2.adj)^2, na.rm=TRUE)),3)  
}  
stats.compare[1, ] <- c("SoilGrids250",  
 me(values(r.gsm.sg),values(r.sg)),  
 rmse(values(r.gsm.sg),values(r.sg)),  
 rmse.adj(values(r.gsm.sg),values(r.sg))  
)  
if (exists("r.p.sg")) {  
 stats.compare[2, ] <- c("POLARIS",  
 me(values(r.gsm.sg),values(r.p.sg)),  
 rmse(values(r.gsm.sg),values(r.p.sg)),  
 rmse.adj(values(r.gsm.sg),values(r.p.sg))  
 )  
}  
if (exists("r.psu.sg")) {  
 stats.compare[3, ] <- c("SPCG100",  
 me(values(r.gsm.sg),values(r.psu.sg)),  
 rmse(values(r.gsm.sg),values(r.psu.sg)),  
 rmse.adj(values(r.gsm.sg),values(r.psu.sg))  
 )  
}  
if (exists("r.landgis.sg")) {  
 stats.compare[4, ] <- c("LandGIS",  
 me(values(r.gsm.sg),values(r.landgis.sg)),  
 rmse(values(r.gsm.sg),values(r.landgis.sg)),  
 rmse.adj(values(r.gsm.sg),values(r.landgis.sg))  
 )  
}  
if (exists("r.issr8.sg")) {  
 stats.compare[5, ] <- c("ISSR-800",  
 me(values(r.gsm.sg),values(r.issr8.sg)),  
 rmse(values(r.gsm.sg),values(r.issr8.sg)),  
 rmse.adj(values(r.gsm.sg),values(r.issr8.sg))  
 )  
}

Save this table for incorporation in a LaTeX document:

options(xtable.floating = FALSE)  
options(xtable.timestamp = "")  
x <- xtable(stats.compare, row.names=FALSE, digits=3)  
autoformat(x)

## % latex table generated in R 4.0.3 by xtable 1.8-4 package  
## %   
## \begin{tabular}{lllll}  
## \hline  
## & PSM\\_product & MD & RMSD & RMSD.Adjusted \\   
## \hline  
## 1 & SoilGrids250 & 2.327 & 5.269 & 4.728 \\   
## 2 & POLARIS & 2.373 & 3.664 & 2.792 \\   
## 3 & SPCG100 & 3.303 & 5.663 & 4.6 \\   
## 4 & LandGIS & 5.308 & 7.172 & 4.823 \\   
## 5 & ISSR-800 & -1.61 & 3.469 & 3.073 \\   
## \hline  
## \end{tabular}

capture.output(print(x, include.rownames=FALSE),   
 file=paste0("../LaTeX\_tables/SoilGrids250\_compare\_statistics\_",  
 AOI.dir.prefix, "\_", voi.sg, "\_", depth, ".tex"))

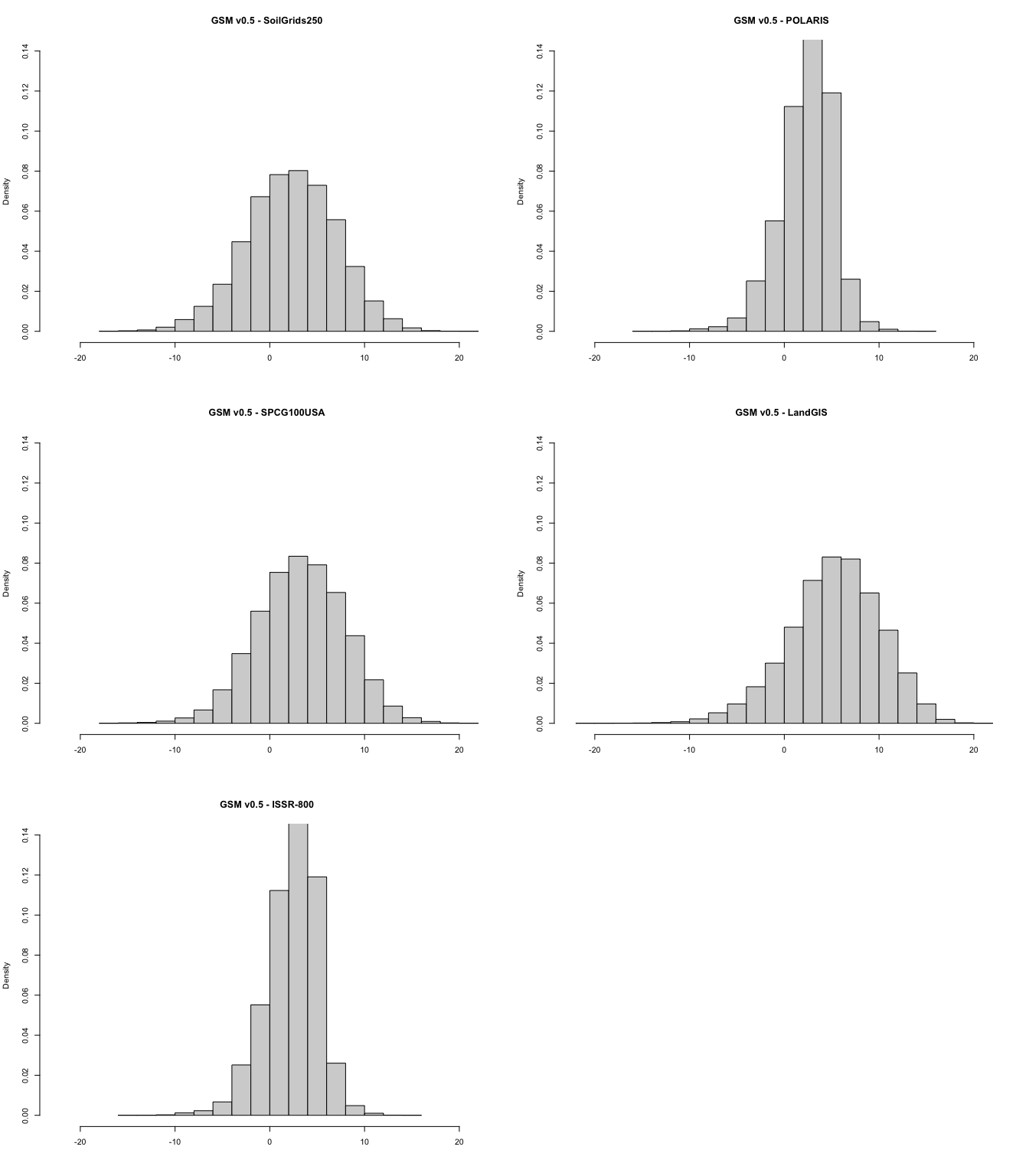
## Compute common range for all differences

zlim <- c(NA, NA)  
zlim <- c(floor(min(zlim[1]\*10, values(diff.gsm.sg)\*10, na.rm=TRUE)),  
 ceiling(max(zlim[2]\*10, values(diff.gsm.sg)\*10, na.rm=TRUE)))/10  
if (exists("diff.gsm.p")) {  
 zlim <- c(floor(min(zlim[1]\*10, values(diff.gsm.p)\*10, na.rm=TRUE)),  
 ceiling(max(zlim[2]\*10, values(diff.gsm.p)\*10, na.rm=TRUE)))/10  
}  
if (exists("diff.gsm.psu")) {  
 zlim <- c(floor(min(zlim[1]\*10, values(diff.gsm.psu)\*10, na.rm=TRUE)),  
 ceiling(max(zlim[2]\*10, values(diff.gsm.psu)\*10, na.rm=TRUE)))/10  
}  
if (exists("diff.gsm.landgis")) {  
 zlim <- c(floor(min(zlim[1]\*10, values(diff.gsm.landgis)\*10, na.rm=TRUE)),  
 ceiling(max(zlim[2]\*10, values(diff.gsm.landgis)\*10, na.rm=TRUE)))/10  
}  
if (exists("diff.gsm.issr8")) {  
 zlim <- c(floor(min(zlim[1]\*10, values(diff.gsm.issr8)\*10, na.rm=TRUE)),  
 ceiling(max(zlim[2]\*10, values(diff.gsm.issr8)\*10, na.rm=TRUE)))/10  
}  
print(zlim)

## [1] -22.5 21.9

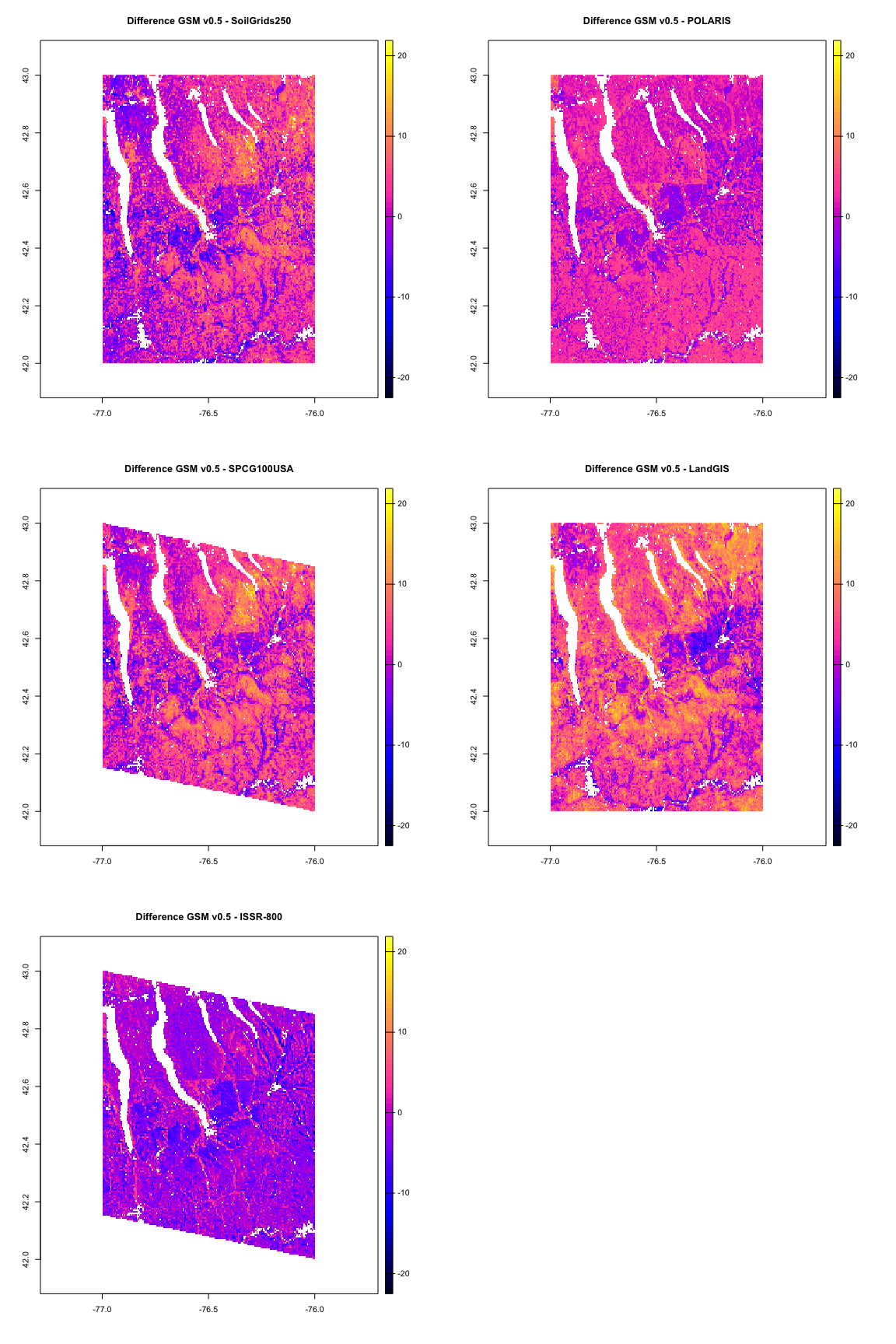
## Histograms

yl <- c(0, 0.14) # Standardize density axis  
par(mfrow=c(3,2))  
hist(diff.gsm.sg, main="GSM v0.5 - SoilGrids250", xlab="",  
 xlim=zlim, freq = FALSE, ylim=yl)  
if (exists("r.p.sg")) {  
 hist(diff.gsm.p, main="GSM v0.5 - POLARIS", xlab="",  
 xlim=zlim, freq = FALSE, ylim=yl)  
}  
if (exists("r.psu.sg")) {  
 hist(diff.gsm.psu, main="GSM v0.5 - SPCG100USA", xlab="",  
 xlim=zlim, freq = FALSE, ylim=yl)  
}  
if (exists("r.landgis.sg")) {  
 hist(diff.gsm.landgis, main="GSM v0.5 - LandGIS", xlab="",  
 xlim=zlim, freq = FALSE, ylim=yl)  
}  
if (exists("r.issr8.sg")) {  
 hist(diff.gsm.p, main="GSM v0.5 - ISSR-800", xlab="",  
 xlim=zlim, freq = FALSE, ylim=yl)  
}  
par(mfrow=c(1,1))



## Maps

par(mfrow=c(3,2))  
terra::plot(diff.gsm.sg, main="Difference GSM v0.5 - SoilGrids250",  
 range=zlim, col=bpy.colors(64))  
if (exists("diff.gsm.p")) {  
 terra::plot(diff.gsm.p, main="Difference GSM v0.5 - POLARIS",  
 range=zlim, col=bpy.colors(64))  
}  
if (exists("diff.gsm.psu")) {  
 terra::plot(diff.gsm.psu, main="Difference GSM v0.5 - SPCG100USA",  
 range=zlim, col=bpy.colors(64))  
}  
if (exists("diff.gsm.landgis")) {  
 terra::plot(diff.gsm.landgis, main="Difference GSM v0.5 - LandGIS",  
 range=zlim, col=bpy.colors(64))  
}  
if (exists("diff.gsm.issr8")) {  
 terra::plot(diff.gsm.issr8, main="Difference GSM v0.5 - ISSR-800",  
 range=zlim, col=bpy.colors(64))  
}  
par(mfrow=c(1,1))



# Save harmonized maps

Set up a directory for these, based on the AOI:

dest.dir.save <- file.path(base.dir.export,  
 AOI.dir.prefix)  
if (!dir.exists(dest.dir.save)) {  
 dir.create(dest.dir.save, recursive = TRUE)  
}

Same AOI, CRS, resolution, units of measure:

# SoilGrids250  
voi.depth.sg <- paste0(voi.sg, "\_", depth.list.sg[depth])  
dest.name <- paste0(dest.dir.save,"/sg\_tile\_250\_", voi.depth.sg, ".tif")  
f <- terra::writeRaster(r.sg, file=dest.name,  
 overwrite=TRUE,  
 wopt=list(gdal=c("TFW=YES"), datatype="FLT4S"),  
 filetype="GTiff")  
GDALinfo(dest.name)

## rows 426   
## columns 626   
## bands 1   
## lower left origin.x -77.23224   
## lower left origin.y 42.00031   
## res.x 0.002349893   
## res.y 0.002349893   
## ysign -1   
## oblique.x 0   
## oblique.y 0   
## driver GTiff   
## projection +proj=longlat +datum=WGS84 +no\_defs   
## file /Users/rossiter/ds/Compare\_PSM/lat4243\_lon-77-76/sg\_tile\_250\_phh2o\_0-5.tif   
## apparent band summary:  
## GDType hasNoDataValue NoDataValue blockSize1 blockSize2  
## 1 Float32 TRUE NaN 3 626  
## apparent band statistics:  
## Bmin Bmax Bmean Bsd  
## 1 43 73 -9999 -9999  
## Metadata:  
## AREA\_OR\_POINT=Area

# GSM v0.5  
dest.name <- paste0(dest.dir.save,"/gsm\_tile\_250\_", voi.depth.sg, ".tif")  
f <- terra::writeRaster(r.gsm.sg, file=dest.name,  
 overwrite=TRUE,  
 wopt=list(gdal=c("TFW=YES"), datatype="FLT4S"),  
 filetype="GTiff")  
GDALinfo(dest.name)

## rows 426   
## columns 626   
## bands 1   
## lower left origin.x -77.23224   
## lower left origin.y 42.00031   
## res.x 0.002349893   
## res.y 0.002349893   
## ysign -1   
## oblique.x 0   
## oblique.y 0   
## driver GTiff   
## projection +proj=longlat +datum=WGS84 +no\_defs   
## file /Users/rossiter/ds/Compare\_PSM/lat4243\_lon-77-76/gsm\_tile\_250\_phh2o\_0-5.tif   
## apparent band summary:  
## GDType hasNoDataValue NoDataValue blockSize1 blockSize2  
## 1 Float32 TRUE NaN 3 626  
## apparent band statistics:  
## Bmin Bmax Bmean Bsd  
## 1 42.00703 72.59309 -9999 -9999  
## Metadata:  
## AREA\_OR\_POINT=Area

# SPCG100USA  
if (exists("r.psu.sg")) {  
 dest.name <- paste0(dest.dir.save,"/psu\_tile\_250\_", voi.depth.sg, ".tif")  
 f <- terra::writeRaster(r.psu.sg, file=dest.name,  
 overwrite=TRUE,   
 wopt=list(gdal=c("TFW=YES"), datatype="FLT4S"),  
 filetype="GTiff")  
 GDALinfo(dest.name)  
}

## rows 426   
## columns 626   
## bands 1   
## lower left origin.x -77.23224   
## lower left origin.y 42.00031   
## res.x 0.002349893   
## res.y 0.002349893   
## ysign -1   
## oblique.x 0   
## oblique.y 0   
## driver GTiff   
## projection +proj=longlat +datum=WGS84 +no\_defs   
## file /Users/rossiter/ds/Compare\_PSM/lat4243\_lon-77-76/psu\_tile\_250\_phh2o\_0-5.tif   
## apparent band summary:  
## GDType hasNoDataValue NoDataValue blockSize1 blockSize2  
## 1 Float32 TRUE NaN 3 626  
## apparent band statistics:  
## Bmin Bmax Bmean Bsd  
## 1 41.91402 70.60033 -9999 -9999  
## Metadata:  
## AREA\_OR\_POINT=Area

# POLARIS  
if (exists("r.p.sg")) {  
 dest.name <- paste0(dest.dir.save,"/polaris\_tile\_250\_", voi.depth.sg, ".tif")  
 f <- terra::writeRaster(r.p.sg, file=dest.name,  
 overwrite=TRUE,  
 wopt=list(gdal=c("TFW=YES"), datatype="FLT4S"),  
 filetype="GTiff")  
 GDALinfo(dest.name)  
}

## rows 426   
## columns 626   
## bands 1   
## lower left origin.x -77.23224   
## lower left origin.y 42.00031   
## res.x 0.002349893   
## res.y 0.002349893   
## ysign -1   
## oblique.x 0   
## oblique.y 0   
## driver GTiff   
## projection +proj=longlat +datum=WGS84 +no\_defs   
## file /Users/rossiter/ds/Compare\_PSM/lat4243\_lon-77-76/polaris\_tile\_250\_phh2o\_0-5.tif   
## apparent band summary:  
## GDType hasNoDataValue NoDataValue blockSize1 blockSize2  
## 1 Float32 TRUE NaN 3 626  
## apparent band statistics:  
## Bmin Bmax Bmean Bsd  
## 1 47.44973 70.77708 -9999 -9999  
## Metadata:  
## AREA\_OR\_POINT=Area

# LandGIS  
if (exists("r.landgis.sg")) {  
 dest.name <- paste0(dest.dir.save,"/landgis\_tile\_250\_", voi.depth.sg, ".tif")  
 f <- terra::writeRaster(r.landgis.sg, file=dest.name,  
 overwrite=TRUE,   
 wopt=list(gdal=c("TFW=YES"), datatype="FLT4S"),  
 filetype="GTiff")  
 GDALinfo(dest.name)  
}

## rows 426   
## columns 626   
## bands 1   
## lower left origin.x -77.23224   
## lower left origin.y 42.00031   
## res.x 0.002349893   
## res.y 0.002349893   
## ysign -1   
## oblique.x 0   
## oblique.y 0   
## driver GTiff   
## projection +proj=longlat +datum=WGS84 +no\_defs   
## file /Users/rossiter/ds/Compare\_PSM/lat4243\_lon-77-76/landgis\_tile\_250\_phh2o\_0-5.tif   
## apparent band summary:  
## GDType hasNoDataValue NoDataValue blockSize1 blockSize2  
## 1 Float32 TRUE NaN 3 626  
## apparent band statistics:  
## Bmin Bmax Bmean Bsd  
## 1 40.76328 69.7853 -9999 -9999  
## Metadata:  
## AREA\_OR\_POINT=Area

# ISSR-800  
if (exists("r.issr8.sg")) {  
 dest.name <- paste0(dest.dir.save,"/issr8\_tile\_250\_", voi.depth.sg, ".tif")  
 f <- terra::writeRaster(r.issr8.sg, file=dest.name,  
 overwrite=TRUE,  
 wopt=list(gdal=c("TFW=YES"), datatype="FLT4S"),  
 filetype="GTiff")  
 GDALinfo(dest.name)  
}

## rows 426   
## columns 626   
## bands 1   
## lower left origin.x -77.23224   
## lower left origin.y 42.00031   
## res.x 0.002349893   
## res.y 0.002349893   
## ysign -1   
## oblique.x 0   
## oblique.y 0   
## driver GTiff   
## projection +proj=longlat +datum=WGS84 +no\_defs   
## file /Users/rossiter/ds/Compare\_PSM/lat4243\_lon-77-76/issr8\_tile\_250\_phh2o\_0-5.tif   
## apparent band summary:  
## GDType hasNoDataValue NoDataValue blockSize1 blockSize2  
## 1 Float32 TRUE NaN 3 626  
## apparent band statistics:  
## Bmin Bmax Bmean Bsd  
## 1 46.95297 71.3209 -9999 -9999  
## Metadata:  
## AREA\_OR\_POINT=Area