Testing Carbon Analysis

Ed Maurer

November 14, 2021

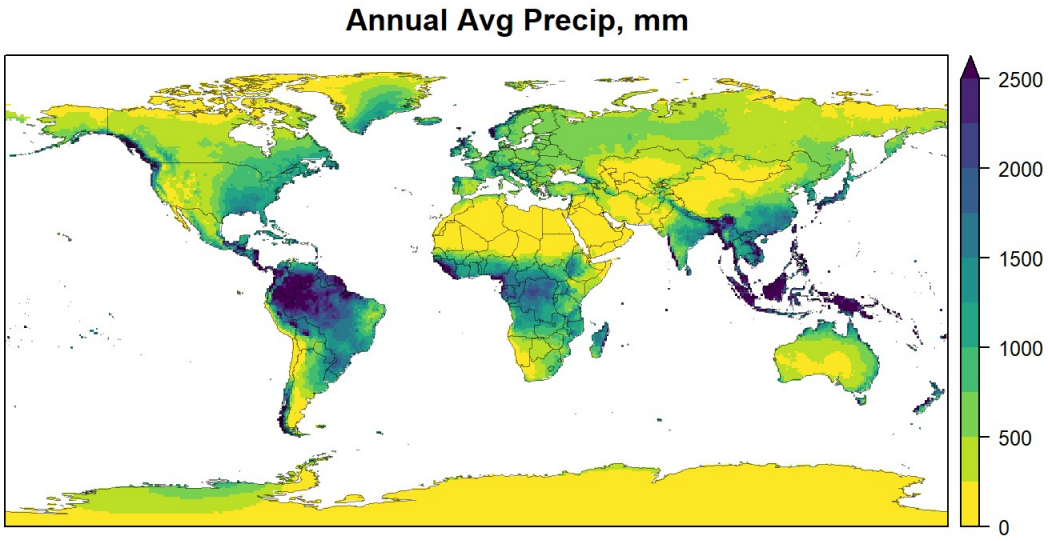
```
options(warn=-1)
suppressMessages(library(stars))
suppressMessages(library(raster))
suppressMessages(library(rasterVis))
suppressMessages(library(RColorBrewer))
suppressMessages(library(maps))
suppressMessages(library(mapdata))
suppressMessages(library(maptools))
suppressMessages(library(lattice))
suppressMessages(library(latticeExtra))
suppressMessages(library(sp))
suppressMessages(library(magrittr))
suppressMessages(library(ggplot2))
suppressMessages(library(viridis))
suppressMessages(library(gridExtra))
suppressMessages(library(spatialEco))
```

Load worldclim mean annual T and P

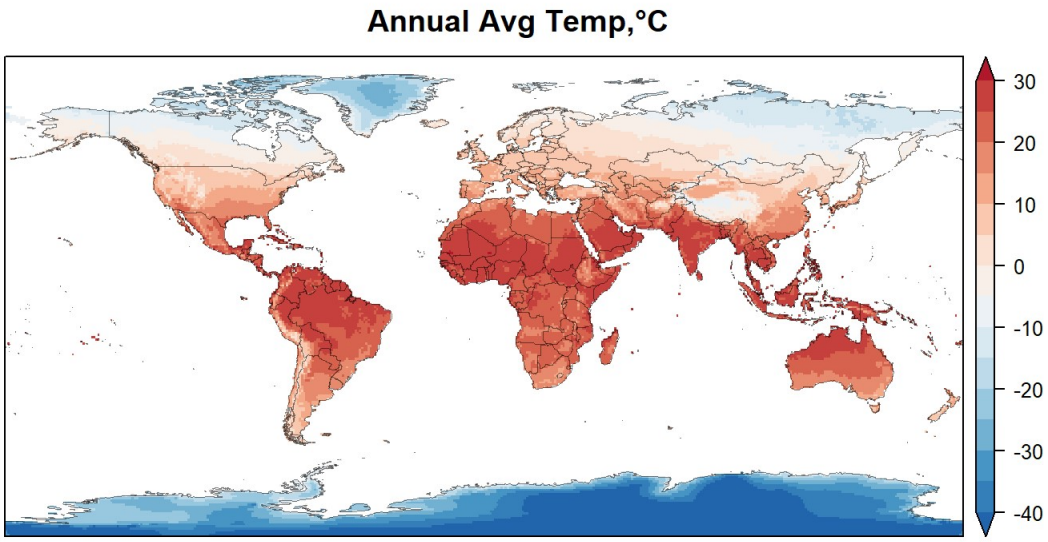
```
#temperature
rt <- raster("data/worldclim/wc2.1_5m_bio_1.tif")

#precip
rp <- raster("data/worldclim/wc2.1_5m_bio_12.tif")
```

Make some initial plots

```
#create world boundaries layer
ext <- as.vector(extent(rt))
boundaries <- map('world', xlim = ext[1:2], ylim = ext[3:4], plot = FALSE)
boundaries <- map2SpatialLines(boundaries, proj4string = CRS(projection(rt)))

# Plot of mean annual P and T
my.at <- c(seq(0, 2500, by = 250), Inf)
revViridis <- rasterTheme(region = rev(viridis(length(my.at)+1)))
levelplot(rp, margin=F, main = "Annual Avg Precip, mm",
          par.settings=revViridis, at=my.at, labels=list(at=my.at),
          colorkey=list(colorkey=T,tri.lower=FALSE, tri.upper=TRUE, width=1.0),
          xlab=NULL,ylab=NULL,scales = list(draw=F,tck = 0)) +
  latticeExtra::layer(sp.lines(boundaries, lwd = 0.5, col = 'black', alpha = 0.5))
```

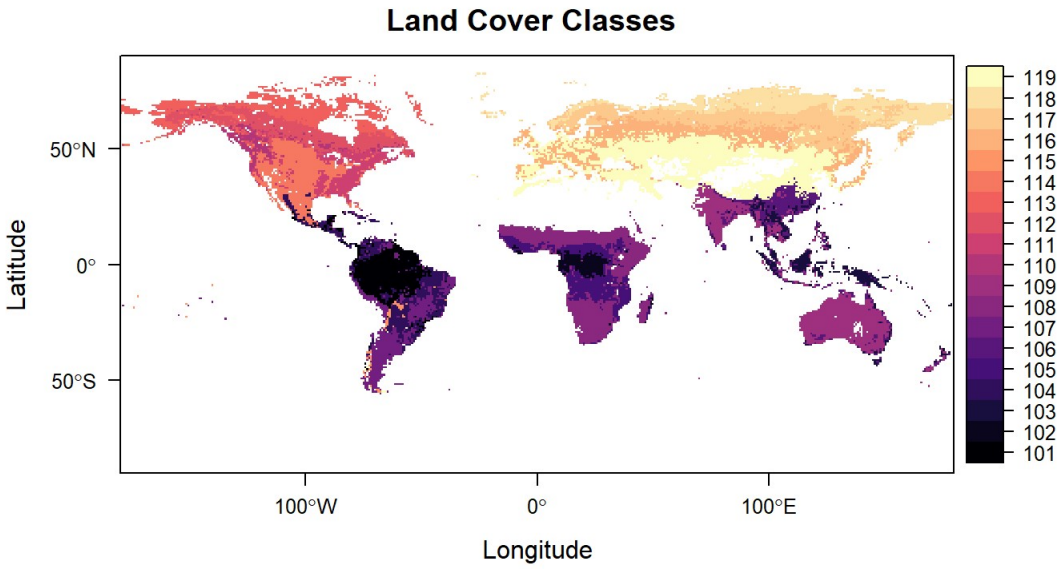



```
my.at <- c(-Inf,seq(-40, 30, by = 5), Inf)
levelplot(rt, margin=F, main = "Annual Avg Temp,\u00B0C",
          par.settings=BuRdTheme(), at=my.at, labels=list(at=my.at),
          colorkey=list(colorkey=T,tri.lower=TRUE, tri.upper=TRUE, width=1.0),
          xlab=NULL,ylab=NULL,scales = list(draw=F,tck = 0)) +
  latticeExtra::layer(sp.lines(boundaries, lwd = 0.5, col = 'black', alpha = 0.5))
```



read in ecoregion map and plot it

```
re <- raster("data/xu_carbon/global_ecoregions.tif")
#remove values not in classification
re[ re[] <= 100 ] <- NA

re <- ratify(re[[1]])
re_rat <- levels(re)[[1]]
re_rat$legend <- as.character(seq(101,119))
levels(re) <- re_rat
levelplot(re, main="Land Cover Classes")
```



read in DEM

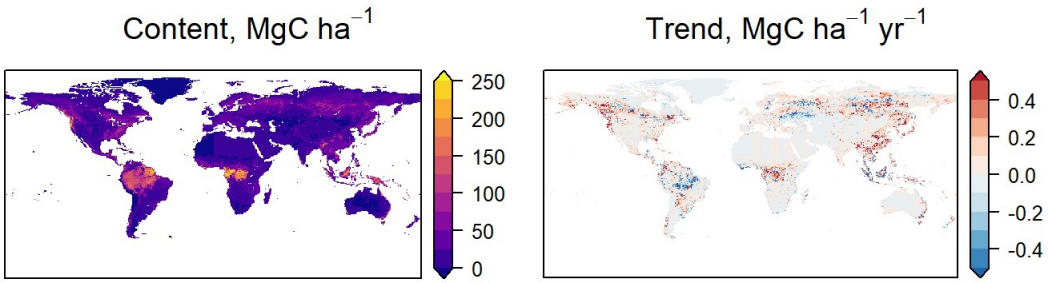
```
rdem <- raster("data/worldclim/wc2.1_5m_elev.tif")
```

read in carbon data

```
#units: MgC ha-1
rc <- stack("data/xu_carbon/test10a_cd_ab_pred_corr_2000_2019_v2.tif")

tm <- seq(as.Date('2000-01-01'), as.Date('2019-01-01'), 'year')
rc <- setZ(rc, tm, 'years')

rc_mean <- calc(rc, fun = mean, na.rm = T)

#Do a linear regression
fun <- function(x) {if (all(is.na(x))) {
    return(c(NA, NA))
  }
  m <- lm(x ~ seq(1,length(x)))
  slope <- summary(m)$coefficients[2,1]
  p.value <- summary(m)$coefficients[2,4]
  return(c(slope,p.value))
}
#rc_lm_trend <- calc(rc, fun)

#save output files
#writeRaster(rc_lm.trend[[1]], filename="C.lm.trend.tif", format="GTiff", overwrite=TRUE)
#writeRaster(rc_lm.trend[[2]], filename="C.lm.p-value.tif", format="GTiff", overwrite=TRUE)
rc_lm_trend <- raster("C.lm.trend.tif")
###NEED TO REMOVE DESERT?
```

plot Carbon and trends

```
s <- stack(rc_mean,rc_lm_trend)
names(s) <- c("Mean","Trend")
lbl1 <- expression("Content, MgC" ~ "ha"^{-1})
lbl2 <- expression("Trend, MgC" ~ "ha"^{-1}~"yr"^{-1})
my.at <- c(-Inf, seq(0, 250, by = 25), Inf)
p1 <- levelplot(s[[1]], margin=F,main=lbl1,
                par.settings=plasmaTheme(), at=my.at, labels=list(at=my.at),
                colorkey=list(tri.lower = TRUE, tri.upper = TRUE, width=1.0),
                xlab=NULL,ylab=NULL,scales = list(draw=F,tck = 0))
my.at <- c(-Inf,seq(-.5, .5, by = 0.1), Inf)
p2 <- levelplot(s[[2]], margin=F,main=lbl2,
                par.settings=BuRdTheme(), at=my.at, labels=list(at=my.at),
                colorkey=list(tri.lower = TRUE, tri.upper = TRUE, width=1.0),
                xlab=NULL,ylab=NULL,scales = list(draw=F,tck = 0))
grid.arrange(p1, p2, ncol=2)
```



resample precip, temp, land cover to equal area grid, 10 km x 10 km

```
EqualEarthProj= "+proj=eqearth +lon_0=0 +x_0=0 +y_0=0 +ellps=WGS84 +datum=WGS84 +units=m +no_defs"

rt_resamp <- projectRaster(rt, method = 'bilinear',
                           crs = EqualEarthProj, res = c(10000,10000))
rt1 = setNames(rt_resamp[[1]], "MAT")
rp_resamp <- projectRaster(rp, rt_resamp, method = 'bilinear')
rp1 = setNames(rp_resamp[[1]], "MAP")
re_resamp <- projectRaster(re, rt_resamp, method = 'ngb')
re1 = setNames(re_resamp[[1]], "LC_Code")
rdem_resamp <- projectRaster(rdem, rt_resamp, method = 'bilinear')
rdem1 = setNames(rdem_resamp[[1]], "Elev")
rc_mean_resamp <- projectRaster(rc_mean, rt_resamp, method = 'bilinear')
rc_mean1 = setNames(rc_mean_resamp[[1]], "Carbon")
rc_lm_trend_resamp <- projectRaster(rc_lm_trend, rt_resamp, method = 'bilinear')
rc_lm_trend1 = setNames(rc_lm_trend_resamp[[1]], "CarbonTrend")
rm(rp_resamp, re_resamp, rdem_resamp, rc_mean_resamp, rc_lm_trend_resamp)

s_all <- stack(rt1, rp1, re1, rdem1, rc_mean1, rc_lm_trend1)
s_xy <- as.data.frame(raster::coordinates(s_all))
s_df <- as.data.frame(getValues(s_all))
df_all <- na.omit(cbind(s_xy, s_df))

#write.csv(df_all, "All_data.csv", row.names = FALSE)
```

Make a plot

```
p1 <- ggplot(df_all, aes(MAT, MAP, z = Carbon)) +
  stat_summary_hex(fun = mean) +
  labs(fill = lbl1) +
  scale_fill_viridis() +
  theme_bw() +
  theme(legend.position = "right")
p2 <- ggplot(df_all, aes(MAT, MAP, z = CarbonTrend)) +
  stat_summary_hex(fun = mean) +
  labs(fill = lbl2) +
  scale_fill_viridis() +
  theme_bw() +
  theme(legend.position = "right")

grid.arrange(p1, p2, ncol=1)
```