Dear Sarah,

Sorry for my slow response. Thank you for your interest on our research. To create land transition flow, we use R-Studio software and ggalluvial package (<https://cran.r-project.org/web/packages/ggalluvial/vignettes/ggalluvial.html>).

First, load all required packages

library("sf", lib.loc = "~/R-dev")

library("dplyr", lib.loc="~/R/win-library/3.5")

library("ggplot2", lib.loc="~/R-dev")

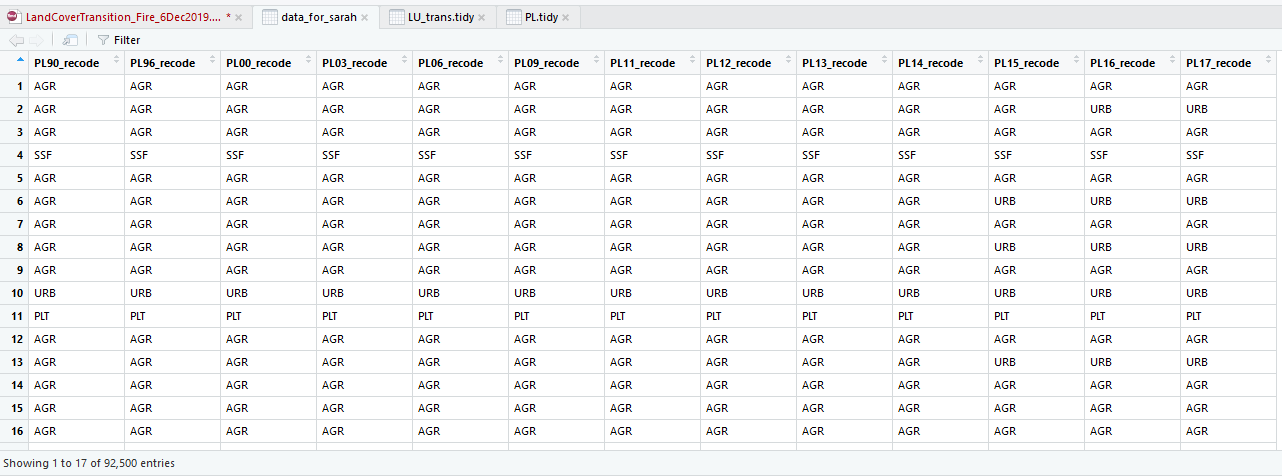
library("magrittr", lib.loc="~/R/win-library/3.5")

library("ggalluvial", lib.loc="~/R/win-library/3.5")

library("Gmisc", lib.loc="~/R/win-library/3.5")

Next, read the attached dataset (data\_for\_sarah.rds) which consist of 17 columns and 92500 rows. The “PL” prefix means Indonesian term *Perubahan Lahan* for Land cover type while number in column name refers to year

data\_for\_sarah <- readRDS(“data\_for\_sarah.rds")



Then reformat the dataset aas PL.tidy variable. It is consist of PL\_year column which refers to year of land cover change, PL\_recode column which indicate land cover type and Freq column shows number of certain land cover type in specific year.

PL.tidy <- data\_for\_sarah %>% tidyr::gather(key = "PL\_year", value = "PL\_recode") %>% table() %>% as.data.frame() %>% dplyr::mutate(PL\_year = as.numeric(substr(PL\_year, start=3, stop=4))) %>% dplyr::mutate( PL\_year = case\_when( PL\_year >= 90 ~ PL\_year + 1900, PL\_year <= 90 ~ PL\_year + 2000))

Furthermore, specify the colour for each land cover type

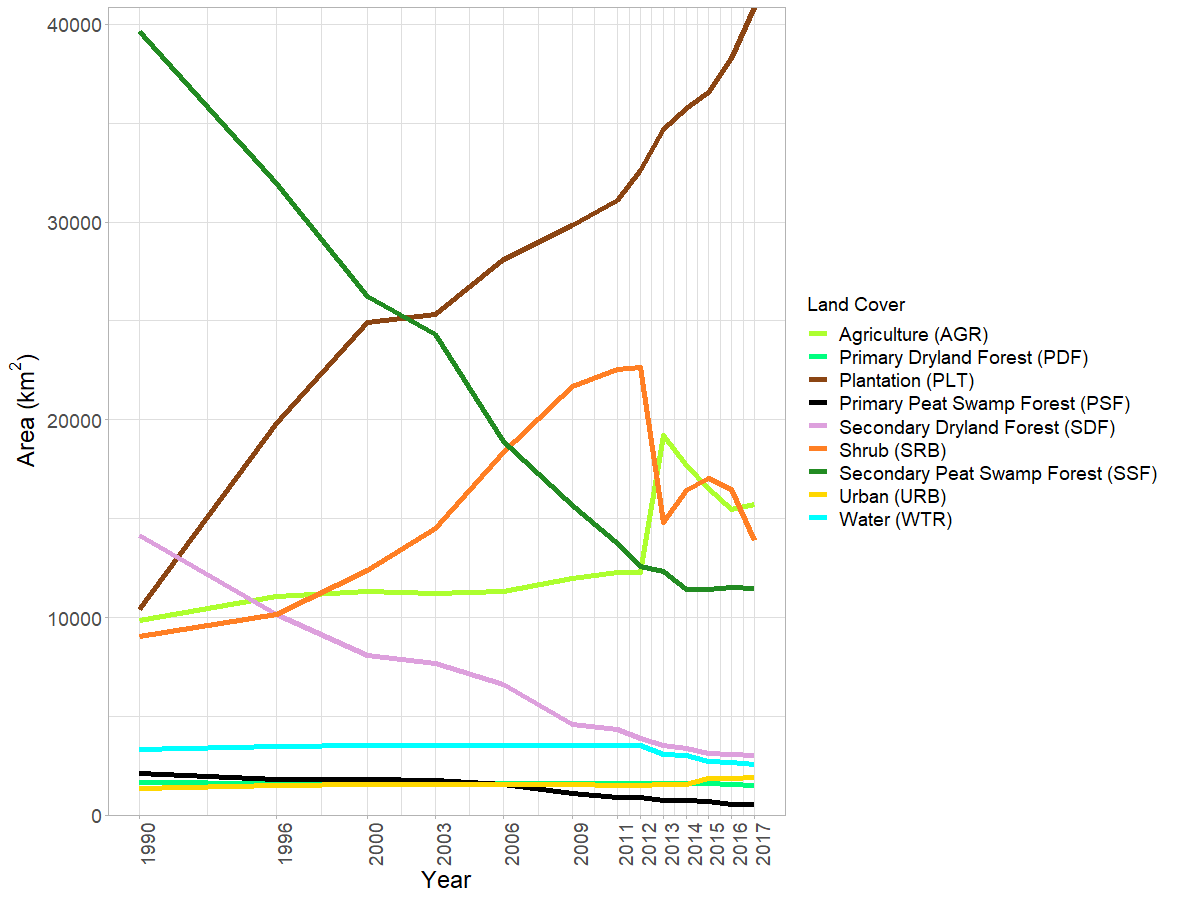
cols = c( "PDF" = "springgreen", "PSF" = "black", "SDF" = "plum", "SSF" = "forestgreen", "PLT" = "chocolate4", "SRB" = "chocolate1", "WTR" = "cyan", "AGR" = "greenyellow", "URB" = "gold" )



Finally, create the graph

s <- ggplot() + geom\_line(data = PL.tidy , aes(x = PL\_year, y = Freq, color= PL\_recode), size = 2) + theme\_light()+ xlab("Year") + ylab(bquote('Area ('\*km^2\*')')) + scale\_colour\_manual(name="Land Cover", values = cols, labels= c("Agriculture (AGR)", "Primary Dryland Forest (PDF)", "Plantation (PLT)", "Primary Peat Swamp Forest (PSF)", "Secondary Dryland Forest (SDF)", "Shrub (SRB)","Secondary Peat Swamp Forest (SSF) ", "Urban (URB)", "Water (WTR)") ) + theme(legend.text=element\_text(size=14) , legend.title = element\_text(size = 14), axis.text.x = element\_text(size = 14, angle = 90), axis.text.y = element\_text(size = 14) , axis.title.x = (element\_text(size = 18)), axis.title.y = (element\_text(size = 18)), legend.position = "right") + scale\_y\_continuous(limits = c(0,NA), expand = c(0,0)) + scale\_x\_continuous(breaks = c(1990, 1996, 2000, 2003, 2006, 2009, 2011, 2012, 2013, 2014, 2015, 2016, 2017))

> print(s)



In order to analyse land cover transition, we select only large land cover type which has frequency greater than 1000. Furthermore, the land type for all years are concatenated by \* symbol and stored as LU\_trans column.

LU\_trans.df <- data\_for\_sarah %>%dplyr::select("PL90\_recode", "PL96\_recode", "PL00\_recode", "PL06\_recode", "PL12\_recode", "PL17\_recode") %>% table() %>% as.data.frame() %>% dplyr::filter(Freq > 1000) %>% dplyr::mutate(LU\_trans = paste(PL90\_recode, PL96\_recode, PL00\_recode, PL06\_recode, PL12\_recode, PL17\_recode, sep="\*") ) %>% `colnames<-`(c("1990","1996","2000","2006","2012","2017","Freq", "LU\_trans"))

LU\_trans.tidy <- LU\_trans.df %>% tidyr::gather(key = "Year", value= "Landcover", -c("Freq", "LU\_trans"))

Finally, draw the graph

>g <- ggplot(LU\_trans.tidy , aes(x = Year, y=Freq, stratum = Landcover, alluvium = LU\_trans, fill = Landcover, label = Landcover)) + scale\_fill\_manual(values = cols, labels= c("Agriculture (AGR)", "Primary Dryland Forest (PDF)", "Plantation (PLT)", "Secondary Dryland Forest (SDF)", "Shrub (SRB)","Secondary Peat Swamp Forest (SSF) ", "Urban (URB)", "Water (WTR)")) + geom\_flow(stat = "alluvium", lode.guidance = "leftright", color = "darkgray") + geom\_stratum() + scale\_y\_continuous(expand = c(0,0))+ ylab(bquote('Area ('\*km^2\*')')) + theme(legend.title = element\_text(size = 14),legend.text=element\_text(size=14) , axis.text.x = element\_text(size = 14, angle = 0), axis.text.y = element\_text(size = 14) , axis.title.x = (element\_text(size = 18)), axis.title.y = (element\_text(size = 18)), legend.position = "right")

> print(g)

