Remote Sensing Data Processing

NEON airborne remote sensing data exploration

Yingying Xie

March 3, 2020

NEON Airborne Observation Platform (AOP) data

Remote sensing (AOP) - Data collected by the airborne observation platform, e.g. LIDAR, surface reflectance

There are 29 AOP data products available from NEON data portal. We will use six of them:

- Ecosystem structure (canopy height)
- Biomass
- Water indices
- Elevation
- Slope and Aspect
- Vegetation indices

The spatial resolution of these AOP remote sensing data is 1 meter. The data were generated into tiles of 1*1km, which are available at NEON data portal (https://data.neonscience.org/home (https://data.neonscience.org/home)).

To explore information of these data products (29 AOP products from 53 sites), go to this web site: https://data.neonscience.org/data-products/explore (https://data.neonscience.org/data-products/explore).

```
# load packages
library(sf)
library(rgdal)
library(raster)
library(rasterVis)
library(tidyverse)
```

Explore AOP data

There are 81 NEON sites with multiple plots, flux tower observations, and other observation settings. You may check the site information at this web site: https://www.neonscience.org/field-sites/field-sites-map/list (https://www.neonscience.org/field-sites/field-sites-map/list).

In this practice, we will examine three sites:

- Harvard Forest - HARV
- Wind River Experimental Forest - WREF
- Yellowstone Northern Range (Frog Rock) - YELL

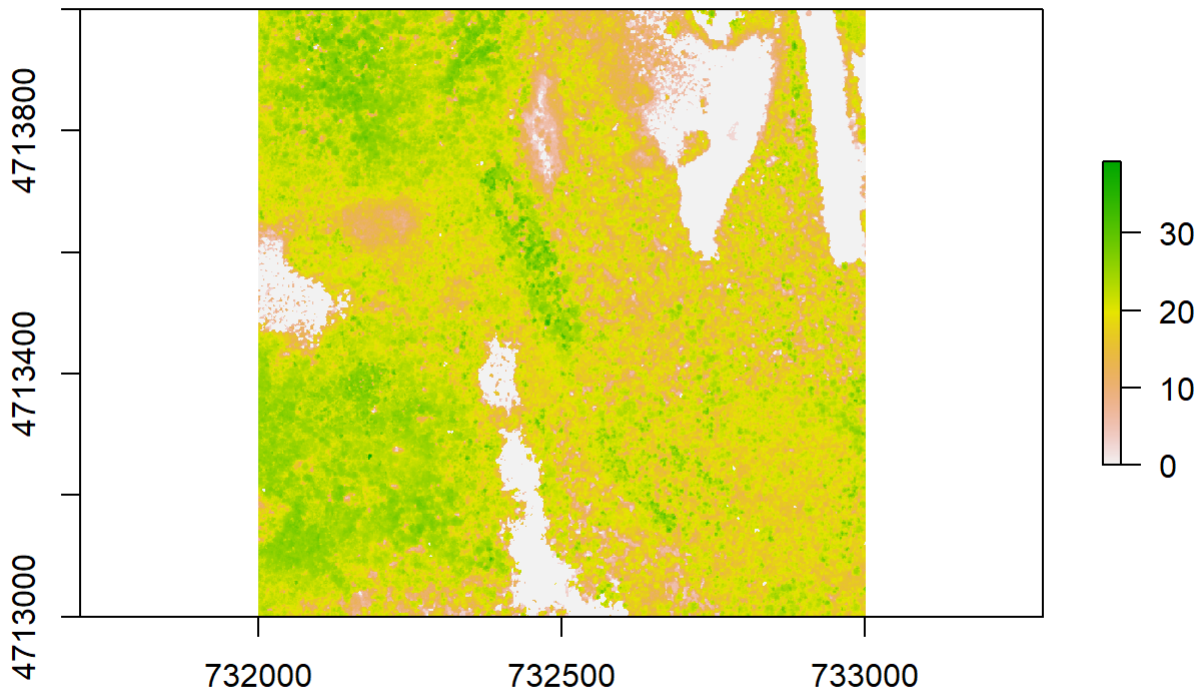
One tile data of six AOP products at these three sites have been downloaded from NEON data portal. You should access the data through the link on Canvas and download them to your local disk. All data files are Tiff files.

Let's first explore the data at Harvard Forest site.

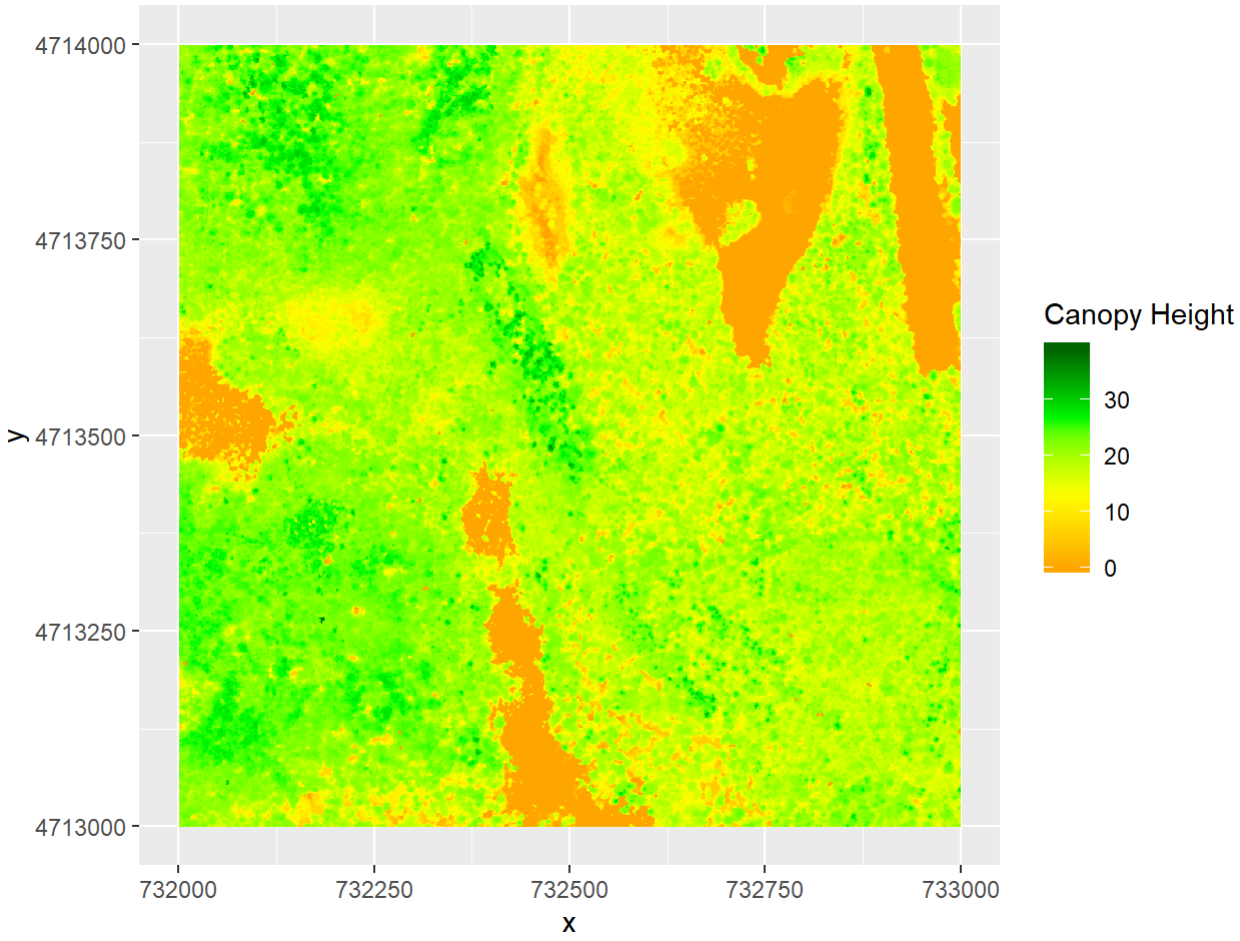
```
# read AOP data
# make sure that you change the path to your file directory
harv_ch2017=raster("09_data/HARV_2017/NEON_D01_HARV_DP3_732000_4713000_CHM.tif")

# check metadata
harv_ch2017
```

```
## class      : RasterLayer
## dimensions : 1000, 1000, 1e+06  (nrow, ncol, ncell)
## resolution : 1, 1  (x, y)
## extent     : 732000, 733000, 4713000, 4714000  (xmin, xmax, ymin, ymax)
## crs        : +proj=utm +zone=18 +datum=WGS84 +units=m +no_defs +ellps=WGS84 +towgs84=0,0,0
## source     : C:/Users/yxa9764/Documents/Courses/2020 Winter/R Data Science/09a RemoteSensing
1/09_data/HARV_2017/NEON_D01_HARV_DP3_732000_4713000_CHM.tif
## names      : NEON_D01_HARV_DP3_732000_4713000_CHM
```

```
# plot it
plot(harv_ch2017)
```



```
# let's use ggplot to graph the data, so we can have more management on the graph
# convert raster data to a data frame
df_harv_ch2017=as.data.frame(harv_ch2017,xy=T,long=T)

# plot it using ggplot
ggplot(df_harv_ch2017)+
  geom_raster(aes(x=x,y=y,fill=value))+
  # use a color scheme with gradients defined by four colors
  scale_fill_gradientn(
    colours=c("orange","yellow","green","darkgreen"),
    name="Canopy Height")+
  coord_equal()
```



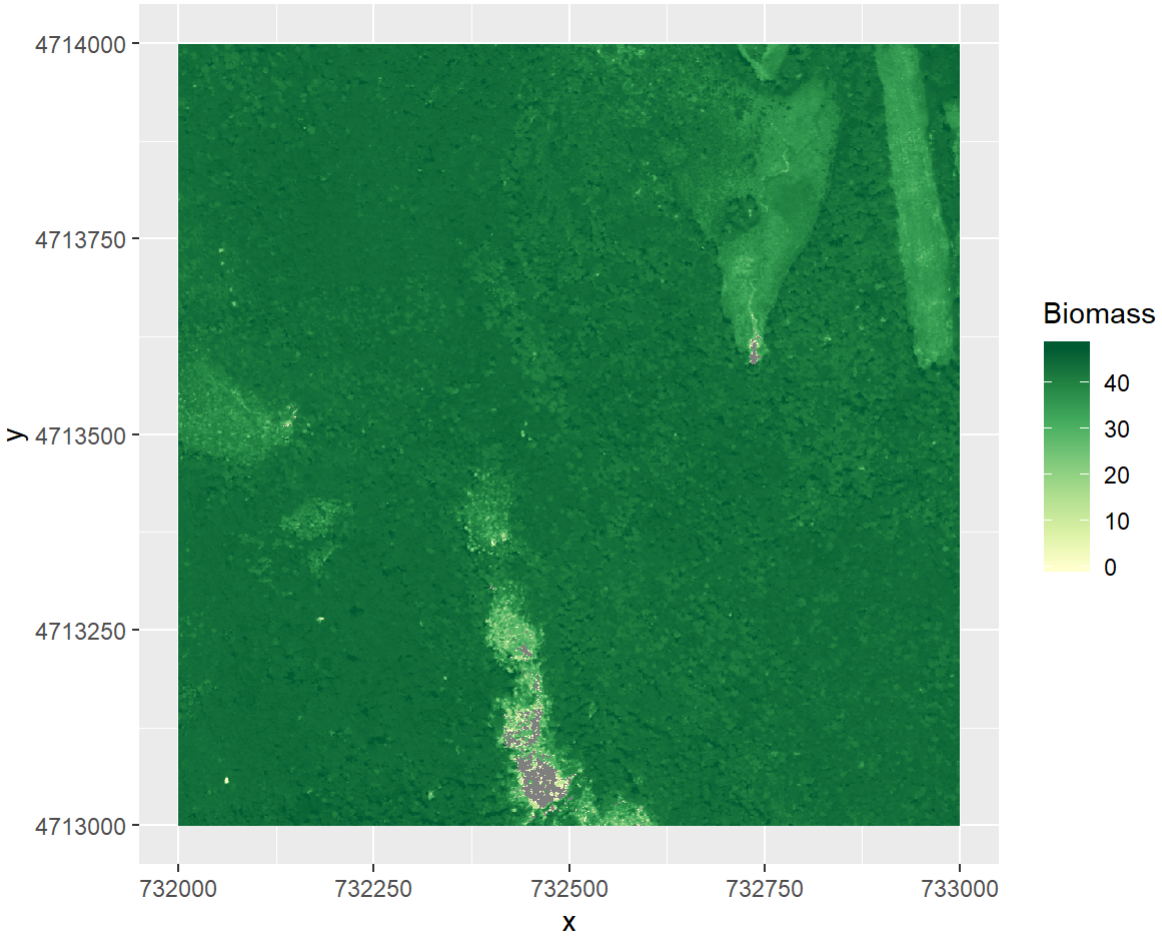
From this plot, we may see the general spatial pattern of canopy height in this 1*1km tile. The area with brown color might be the bare land with no or very low amount of vegetation (low canopy height), and the area with green color represent forested area (high canopy height).

Now let's check the biomass data using the same way shown above.

```
harv_bm2017=raster("09_data/HARV_2017/NEON_D01_HARV_DP3_732000_4713000_Biomass.tif")

harv_bm2017
```

```
## class      : RasterLayer
## dimensions : 1000, 1000, 1e+06  (nrow, ncol, ncell)
## resolution : 1, 1  (x, y)
## extent     : 732000, 733000, 4713000, 4714000  (xmin, xmax, ymin, ymax)
## crs        : +proj=utm +zone=18 +datum=WGS84 +units=m +no_defs +ellps=WGS84 +towgs84=0,0,0
## source     : C:/Users/yxa9764/Documents/Courses/2020 Winter/R Data Science/09a RemoteSensing
1/09_data/HARV_2017/NEON_D01_HARV_DP3_732000_4713000_Biomass.tif
## names      : NEON_D01_HARV_DP3_732000_4713000_Biomass
```

```
df_harv_bm2017=as.data.frame(harv_bm2017,xy=T,long=T)

ggplot(df_harv_bm2017)+
  geom_raster(aes(x=x,y=y,fill=value))+
  # use a sequential color scheme "YlGn"
  scale_fill_distiller( palette = "YlGn", direction = 1, name="Biomass")+
  coord_equal()
```



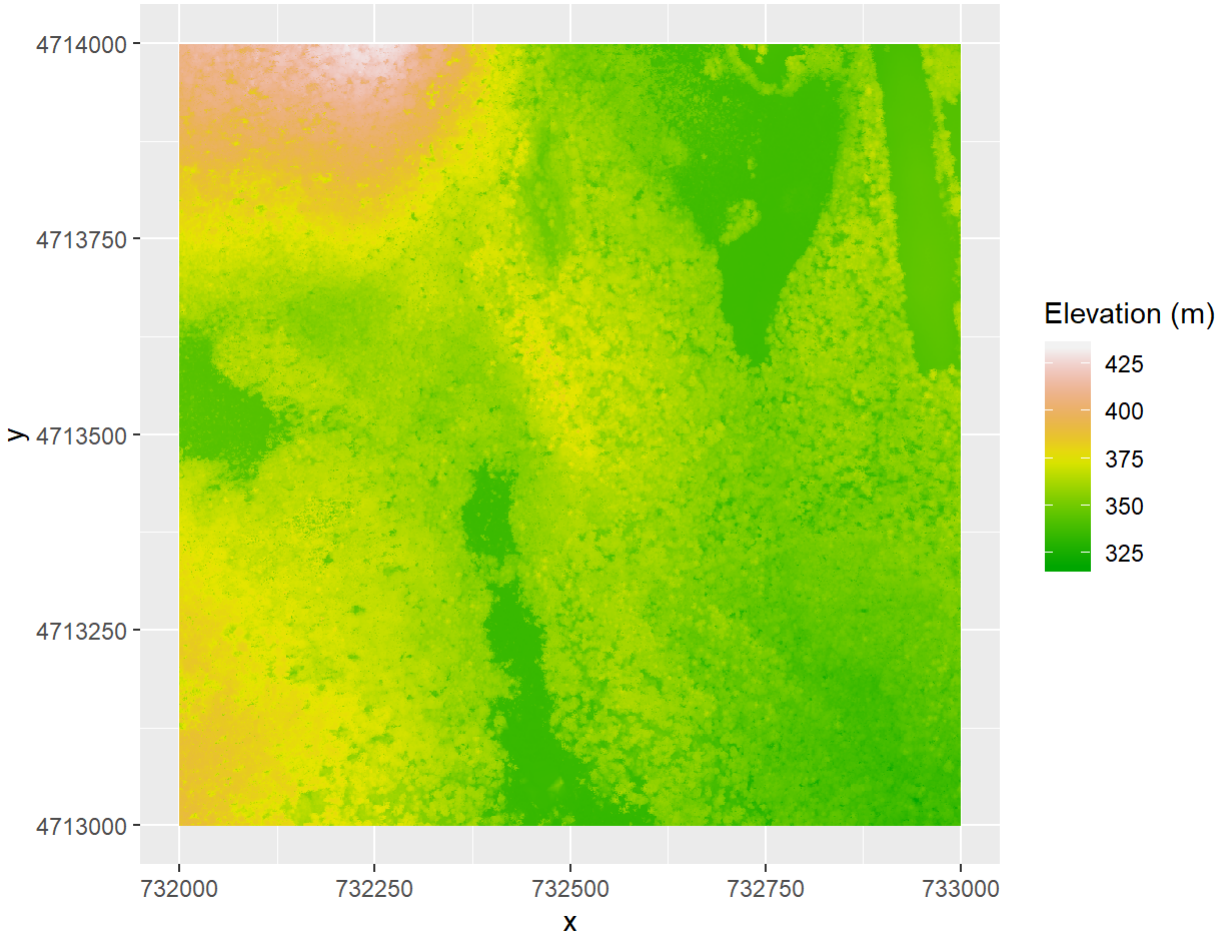
Apparently, most areas have a high amount of biomass as the observation was done in summer time, that's why we see a lot of darkgreen in the plot.

What about the topography in this region?

```
harv_el2017=raster("09_data/HARV_2017/NEON_D01_HARV_DP3_732000_4713000_DSM.tif")

harv_el2017
```

```
## class      : RasterLayer
## dimensions : 1000, 1000, 1e+06  (nrow, ncol, ncell)
## resolution : 1, 1  (x, y)
## extent     : 732000, 733000, 4713000, 4714000  (xmin, xmax, ymin, ymax)
## crs        : +proj=utm +zone=18 +datum=WGS84 +units=m +no_defs +ellps=WGS84 +towgs84=0,0,0
## source     : C:/Users/yxa9764/Documents/Courses/2020 Winter/R Data Science/09a RemoteSensing
1/09_data/HARV_2017/NEON_D01_HARV_DP3_732000_4713000_DSM.tif
## names      : NEON_D01_HARV_DP3_732000_4713000_DSM
```

```
df_harv_el2017=as.data.frame(harv_el2017,xy=T,long=T)

ggplot(df_harv_el2017)+
  geom_raster(aes(x=x,y=y,fill=value))+
  # use a terrain color scheme for visualizing topography
  scale_fill_gradientn( colours=terrain.colors(100),
    name="Elevation (m)")+
  coord_equal()
```



It seems the northwestern area has high elevation and the southeastern area is relatively low in elevation.

Now it's your turn to explore other AOP data at Harvard Forest. Refer to the code above to check the following data. Make the plots and briefly describe the information shown in the plots.

1. Vegetation index - NDVI (Normalized Difference Vegetation Index)

The Normalized Difference Vegetation Index (NDVI) is one of the most widely used indicator that describes the greenness - the relative density and health of vegetation, based on red and near-infrared light waves reflected by land surfaces.

NDVI values range from -1.0 to +1.0, but the common range of NDVI values is 0 to 1. Areas of barren rock, sand, or snow usually show very low NDVI values (for example, 0.1 or less). Sparse vegetation such as shrubs and grasslands or senescing crops may result in moderate NDVI values (approximately 0.2 to 0.5). High NDVI values (approximately 0.6 to 0.9) correspond to dense vegetation such as that found in temperate and tropical forests or crops at their peak growth stage.

```
# let's start from here
harv_ndvi2017=raster("09_data/HARV_2017/VegIndices/NEON_D01_HARV_DP3_732000_4713000_NDVI.tif")
```

Write your code below:

What do you know about the forests from NDVI values shown in your graph?

2. Water index - NDWI (Normalized Difference Water Index)

The Normalized Difference Water Index (NDWI) is a remote sensing derived index estimating the leaf water content at canopy level. NDWI is known to be strongly related to the plant water content. It is therefore a very good proxy for plant water stress.

The NDWI value varies between -1 to +1, depending on the leaf water content but also on the vegetation type and cover. High values of NDWI correspond to high vegetation water content and to high vegetation fraction cover. Low NDWI values correspond to low vegetation water content and low vegetation fraction cover. In period of water stress, NDWI will decrease.

```
# let's start from here
harv_ndwi2017=raster("09_data/HARV_2017/WaterIndices/NEON_D01_HARV_DP3_732000_4713000_NDWI.tif")
```

Write your code below:

What do you know about the forests from NWVI values shown in your graph?

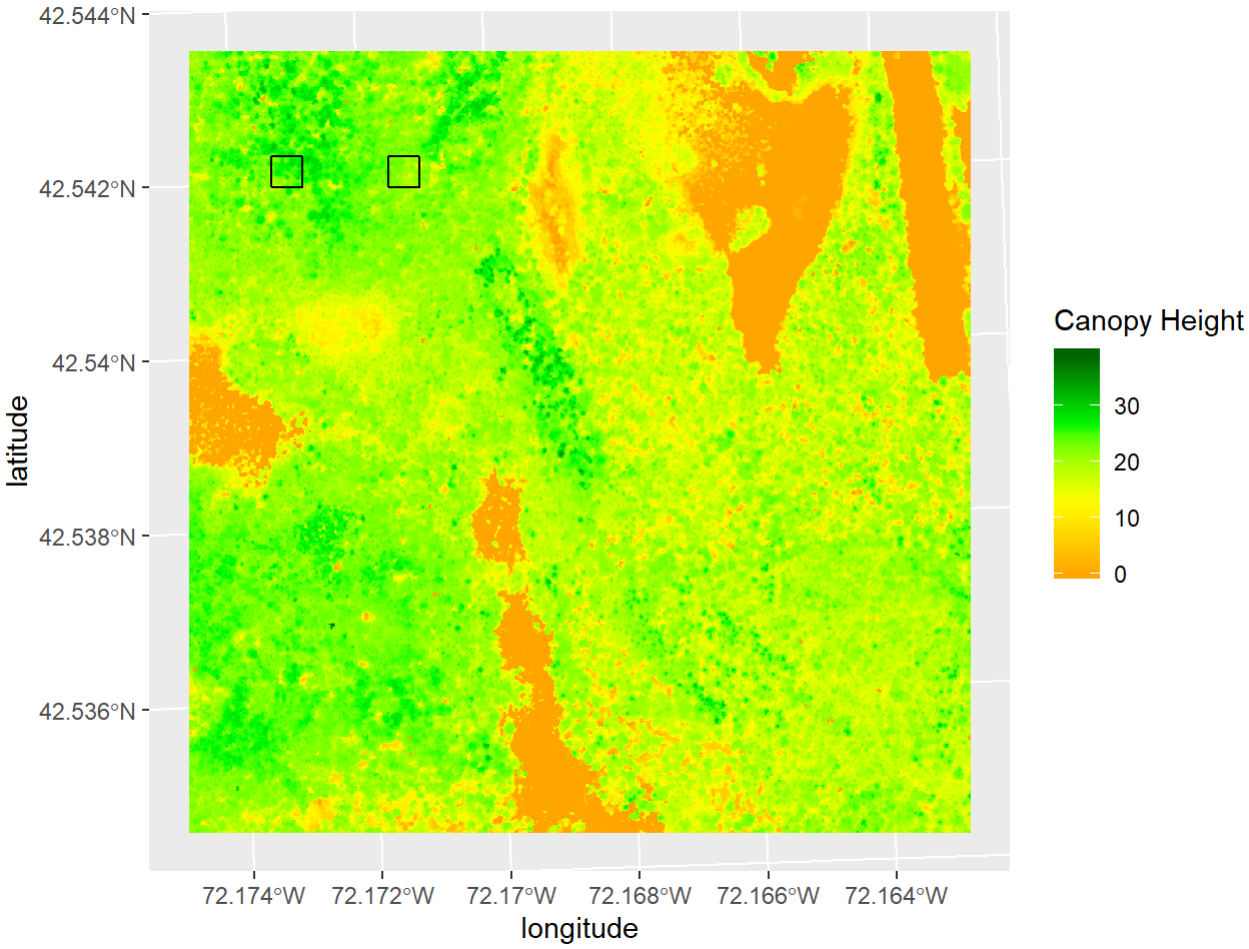
Extract data for observatory plots

At each site, we are interested in one observatory plots representing local dominant vegetation type. For example, dominant vegetation type at Harvard Forest is deciduous forest. They are relatively small sites (40m*40m) to observe specific biolgoical and ecological processes. To extract data for the plots, we need to use the spaital information of the plots from the spatial data provided by NEON.

By exploring the site information of Harvard Forest at https://www.neonscience.org/field-sites/field-sites-map/HARV (https://www.neonscience.org/field-sites/field-sites-map/HARV). We determined two plots that we are interested in: HARV_036, and HARV_045.

```
# read NEON plots shapefile
plot_polygons <- read_sf("09_data/All_NEON_TOS_Plots_V7/All_NEON_TOS_Plot_Polygons.shp",  quiet
 = T)
plot_polygons
colnames(plot_polygons)

# filter plot polygon for two representative plots: HARV_036 and HARV_045
HARV_plots=filter(plot_polygons,plotID %in% c("HARV_036","HARV_045"))

HARV_plots
```

```
# convert datasets for visualization
HARV_plots=st_transform(HARV_plots,crs(harv_ch2017))


# plot the tile and the plots
ggplot(df_harv_ch2017)+
  geom_raster(aes(x=x,y=y,fill=value))+
  scale_fill_gradientn(
    colours=c("orange","yellow","green","darkgreen"),
    name="Canopy Height")+
  geom_sf(data=HARV_plots,fill=NA,color="black")+
  xlab("longitude")+ylab("latitude")
```



To better examine the detailed information for the plots, we may extract the remote sensing data for the plot area.

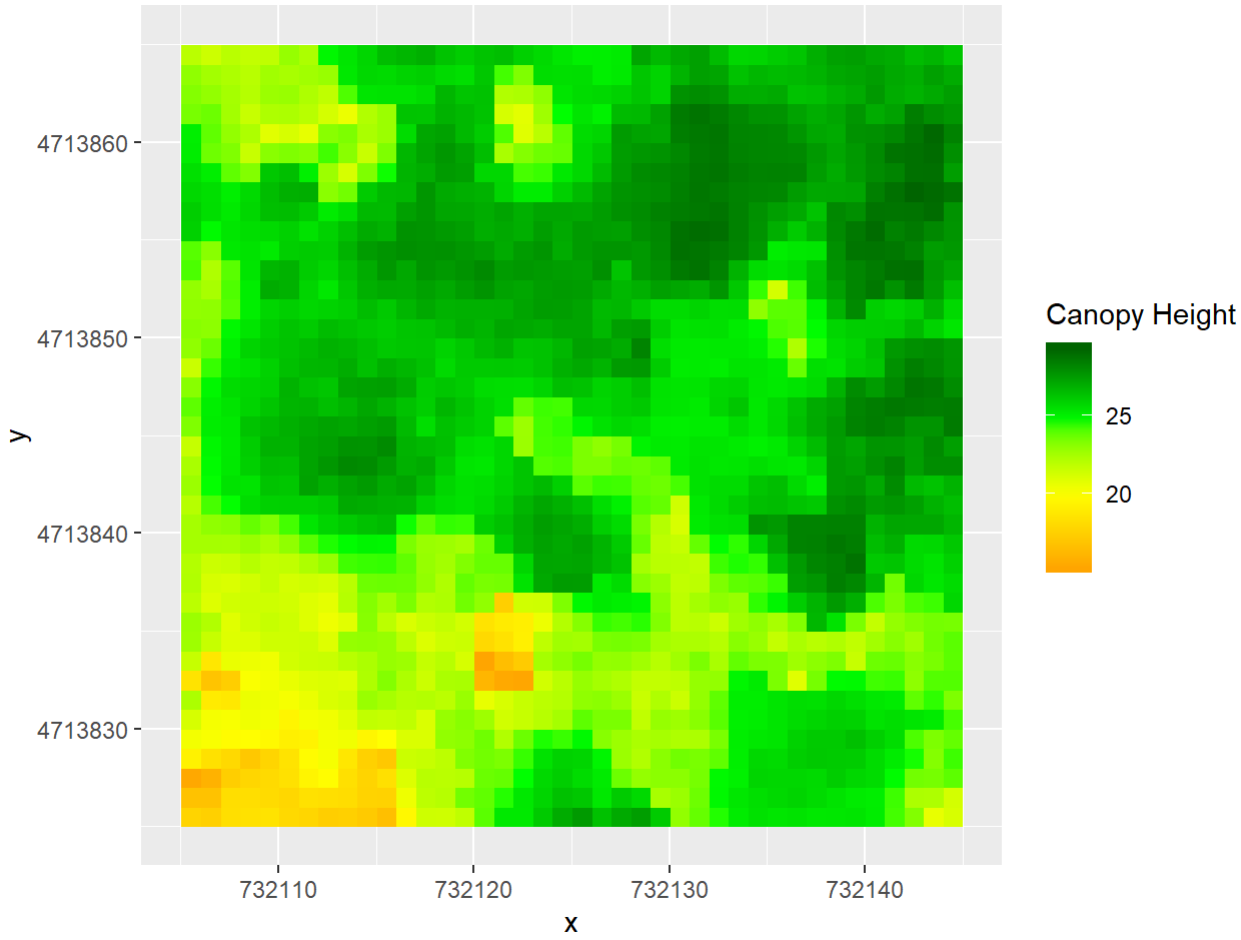
```
# Check plots information
HARV_plots$plotID
```

```
## [1] "HARV_036" "HARV_045"
```

```
# crop canopy height data based on polygon of one plot
harv_036=crop(harv_ch2017,extent(HARV_plots[1,])) # first polygon element of HARV_plots is for H
ARV_036

harv_036
```

```
## class      : RasterLayer
## dimensions : 40, 40, 1600  (nrow, ncol, ncell)
## resolution : 1, 1  (x, y)
## extent     : 732105, 732145, 4713825, 4713865  (xmin, xmax, ymin, ymax)
## crs        : +proj=utm +zone=18 +datum=WGS84 +units=m +no_defs +ellps=WGS84 +towgs84=0,0,0
## source     : memory
## names      : NEON_D01_HARV_DP3_732000_4713000_CHM
## values     : 15.31, 29.33  (min, max)
```

```
# plot canopy height for HARV_036 plot area
df_harv_036=as.data.frame(harv_036,xy=T,long=T)

ggplot(df_harv_036)+
  geom_raster(aes(x=x,y=y,fill=value))+
  scale_fill_gradientn(
    colours=c("orange","yellow","green","darkgreen"),
    name="Canopy Height")+
  coord_equal()
```



From the graph, you can see that this plot is 40 by 40 meters. Each pixel in the graph represent 1 by 1 meter area, as the spatial resolution of the data is 1 meter.

Now it's your turn to extract and graph the canopy height data for HARV_045 plot.

Write your code below:

Stack and manage data for multiple products and plots

As there are remote sensing data from 9 AOP products of 3 sites, data loading and processing can be inconvenient when we read each data to one object, which lead to too many data objects. Stacking similar data into a RasterStack or RasterBrick object with multiple raster layers would be more efficient for data management and processing.

Create multi-layer data

```
#generate a list of input rasters that have the same extent and spaital resolution
#pattern = "*.tif$" - filters for main raster files
files <- list.files("09_data/HARV_2017" , pattern = "*.tif$")
veg <- list.files("09_data/HARV_2017/VegIndices" , pattern = "*.tif$")   # Vegetation indices

#create a raster stack from the input raster files
s1 <- raster::stack(paste0("09_data/HARV_2017/", files))
s2 <- raster::stack(paste0("09_data/HARV_2017/VegIndices/", veg))

# simplify the name for each layer
names=substr(names(s1),34,nchar(names(s1)))
names(s1)=names

names=substr(names(s2),34,nchar(names(s2)))
names(s2)=names
```

Export plot level data

As we have extracted data for observatory plots, we want to export the plot level data to a data file, so we can work on them in the future without extracting them again.

```
#write the output raster to a grid file
#although you can also output to a GeoTiff file, a grid file preserves the names for each layer, so when you read the data next time, you still know what data each layer is
writeRaster(s1, filename = "09_data/HARV_2017/HARV_2017_BioTopo.grd", format="raster", overwrite=TRUE)
writeRaster(s2, filename = "09_data/HARV_2017/HARV_2017_VegIndices.grd", format="raster", overwrite=TRUE)
```

```
# you can test it using the code below
r=brick(("09_data/HARV_2017/HARV_2017_BioTopo.grd"))

r
```

Now it is your turn to stack all water indices layers to a RasterStack object, then export it to a grid file.

```
# let's start from here
water <- list.files("09_data/HARV_2017/WaterIndices" , pattern = "*.tif$")   # Water indices
```

Write your code below:

While we have multi-layer remote sensing data, it is easier to extract plot level data for all layers from multiple data products

```
# extract plot data
harv_036=crop(r,extent(HARV_plots[1,]))
harv_045=crop(r,extent(HARV_plots[2,]))

# check plot data
harv_036
```

```
## class      : RasterBrick
## dimensions : 40, 40, 1600, 6  (nrow, ncol, ncell, nlayers)
## resolution : 1, 1  (x, y)
## extent     : 732105, 732145, 4713825, 4713865  (xmin, xmax, ymin, ymax)
## crs        : +proj=utm +zone=18 +datum=WGS84 +units=m +no_defs +ellps=WGS84 +towgs84=0,0,0
## source     : memory
## names      :  aspect, Biomass,     CHM,     DSM,     DTM,   slope
## min values :  14.754,  41.429,  15.310, 370.160, 361.140,   0.296
## max values : 348.862,  47.024,  29.330, 394.930, 367.470,  18.060
```

Compare plots

Now we can dig details for the observatory plots. For example, although both the two plots represent deciduous forest dominant vegetation, are there any differences in their biological and ecological status?

To visualize the differences between two plots, we need to graph them together using a same color scheme.

```
# convert raster data to data frame, then combine data of two plots
df_plots=rbind(as.data.frame(harv_036,xy=T,long=T), as.data.frame(harv_045,xy=T,long=T))
head(df_plots)
```

	x	y	layer	value
	<dbl>	<dbl>	<chr>	<dbl>
1	732105.5	4713865	aspect	177.363
2	732106.5	4713865	aspect	178.883
3	732107.5	4713865	aspect	183.366
4	732108.5	4713865	aspect	181.811
5	732109.5	4713865	aspect	171.094
6	732110.5	4713865	aspect	158.764

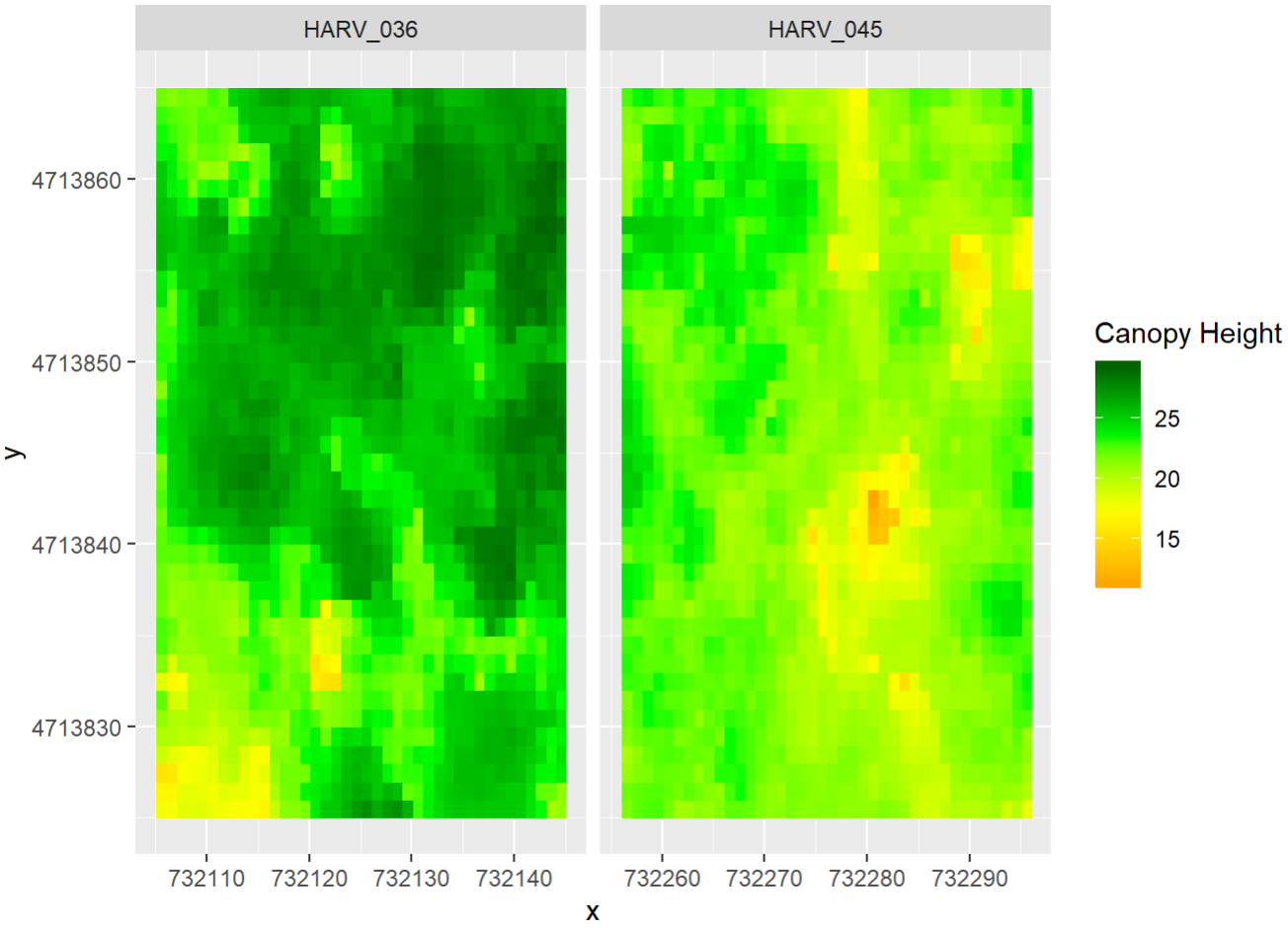
6 rows

```
dim(df_plots)
```

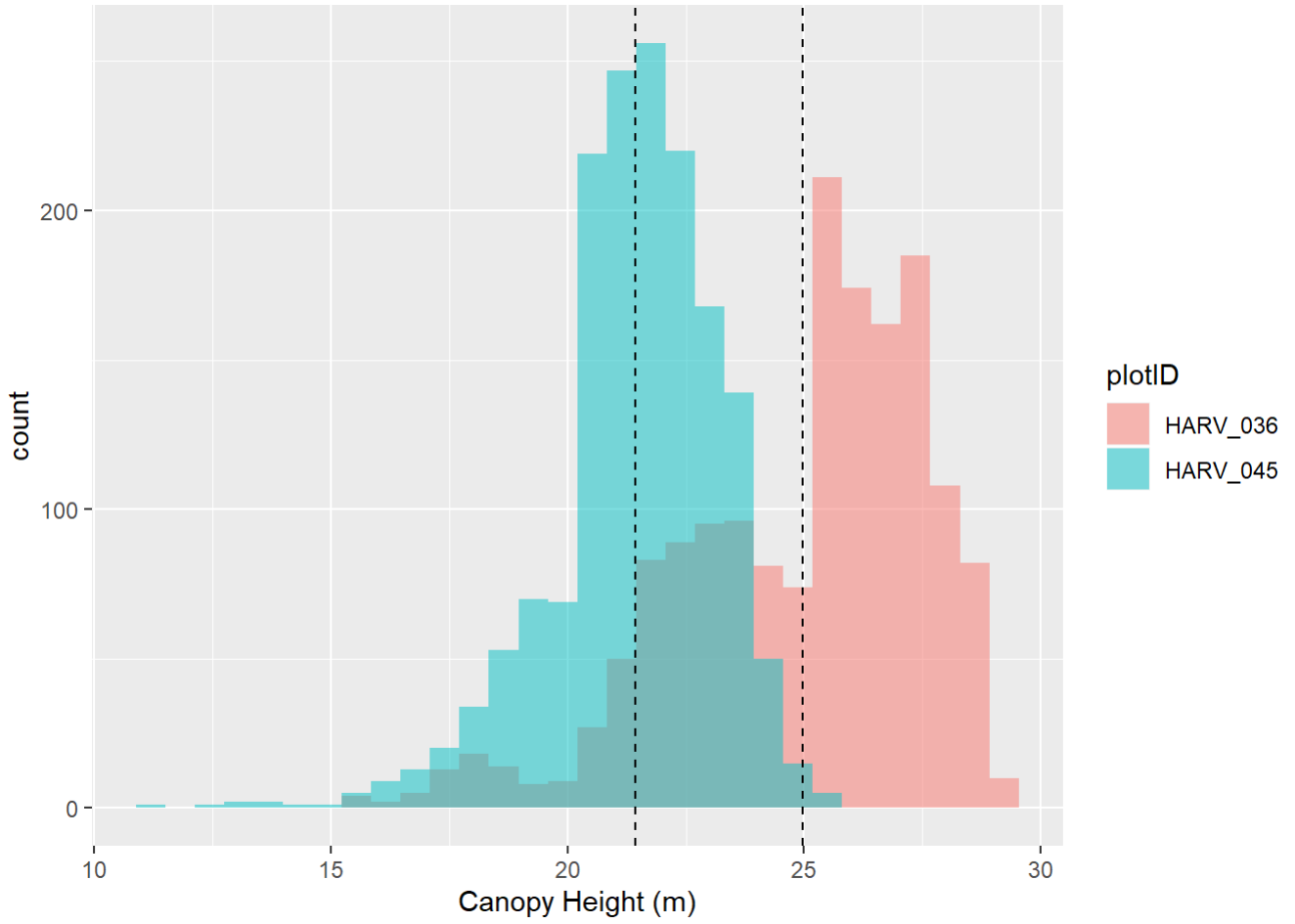
```
## [1] 19200     4
```

```
# add plotID column
df_plots$plotID=c(rep("HARV_036",40*40*6),rep("HARV_045",40*40*6))
```

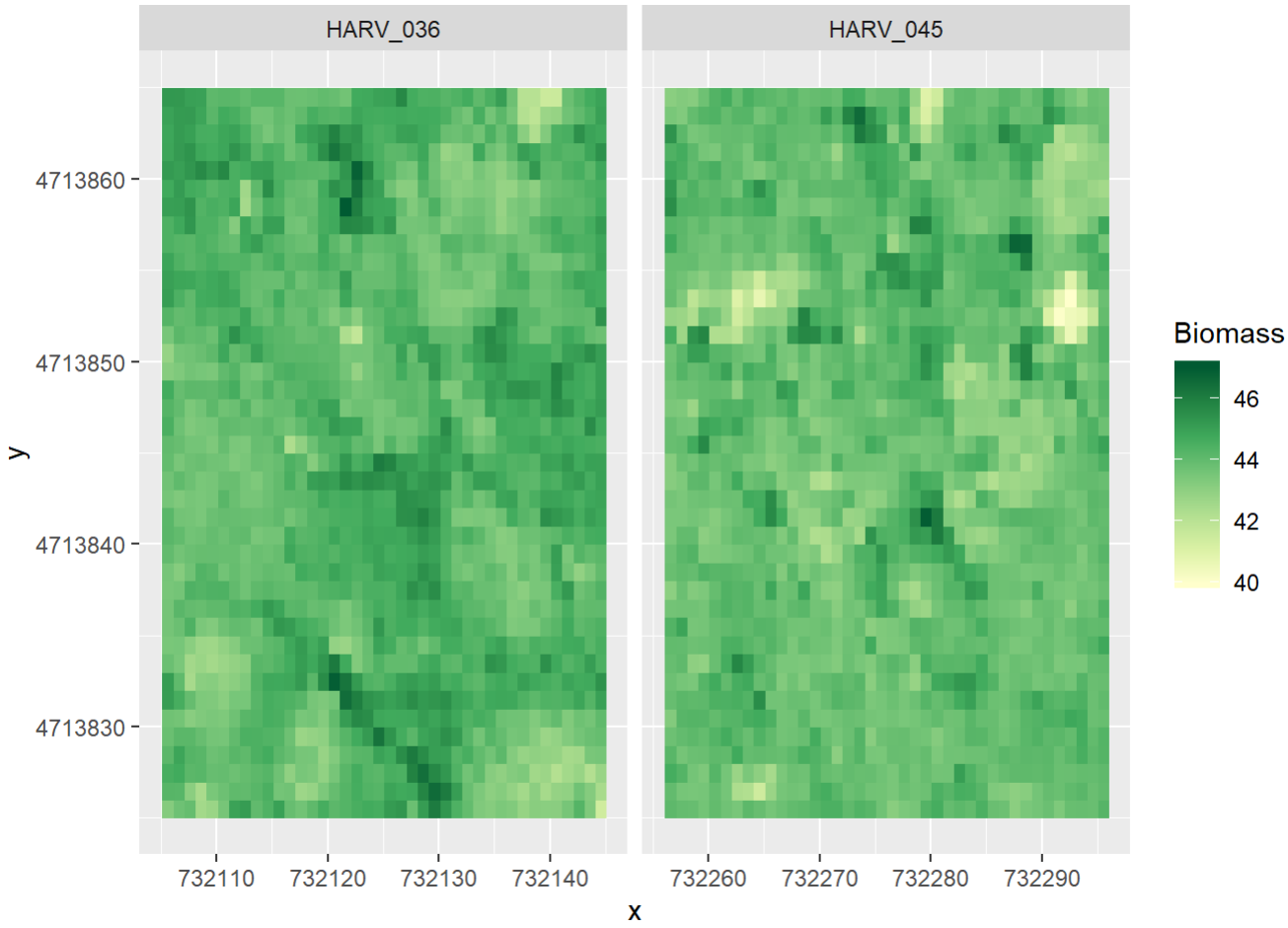
```
# compare canopy height
ggplot(filter(df_plots,layer=="CHM"))+
  geom_raster(aes(x=x,y=y,fill=value))+
  scale_fill_gradientn(
    colours=c("orange","yellow","green","darkgreen"),
    name="Canopy Height")+
  facet_wrap(~plotID, scale="free_x")
```



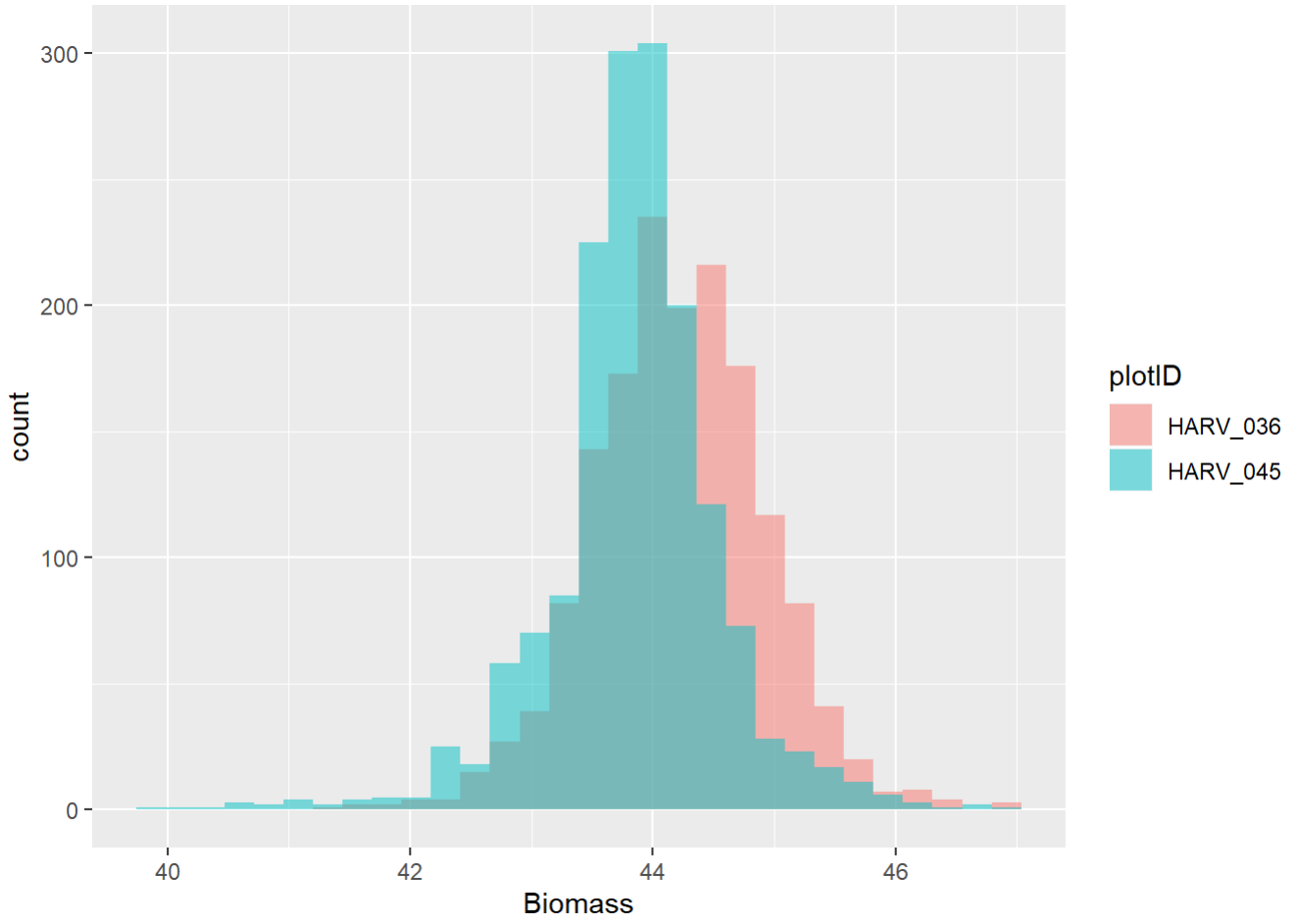

```
# compare the histogram of canopy height
meanch=filter(df_plots,layer=="CHM") %>%
  group_by(plotID) %>%
  summarise(mean(value)) %>%
  as.data.frame()


ggplot(data=filter(df_plots,layer=="CHM"),aes(x=value, fill=plotID)) +
  geom_histogram(position="identity",alpha=0.5)+
  # add mean value
  geom_vline(xintercept=as.numeric(meanch[,2]),linetype=2)+
  xlab("Canopy Height (m)")
```




```
# compare biomass
ggplot(filter(df_plots,layer=="Biomass"))+
  geom_raster(aes(x=x,y=y,fill=value))+
  scale_fill_distiller( palette = "YlGn", direction = 1, name="Biomass")+
  facet_wrap(~plotID, scale="free_x")
```




```
# compare the histogram of biomass
ggplot(data=filter(df_plots,layer=="Biomass"),aes(x=value, fill=plotID)) +
  geom_histogram(position="identity",alpha=0.5)+
  xlab("Biomass")
```



What about the differences of all five water indices between two plots?

```
r1=brick("09_data/HARV_2017/HARV_2017_WaterIndices.grd")

# extract plot data
harv_036=crop(r1,extent(HARV_plots[1,]))
harv_045=crop(r1,extent(HARV_plots[2,]))


# convert raster data to data frame, then combine data of two plots
df_plots1=rbind(as.data.frame(harv_036,xy=T,long=T), as.data.frame(harv_045,xy=T,long=T))
head(df_plots1)
```

	x	y	layer	value
	<dbl>	<dbl>	<chr>	<dbl>
1	732105.5	4713865	MSI	0.4301075
2	732106.5	4713865	MSI	0.3799039
3	732107.5	4713865	MSI	0.3665031
4	732108.5	4713865	MSI	0.3870968
5	732109.5	4713865	MSI	0.3962209
6	732110.5	4713865	MSI	0.3973931

6 rows

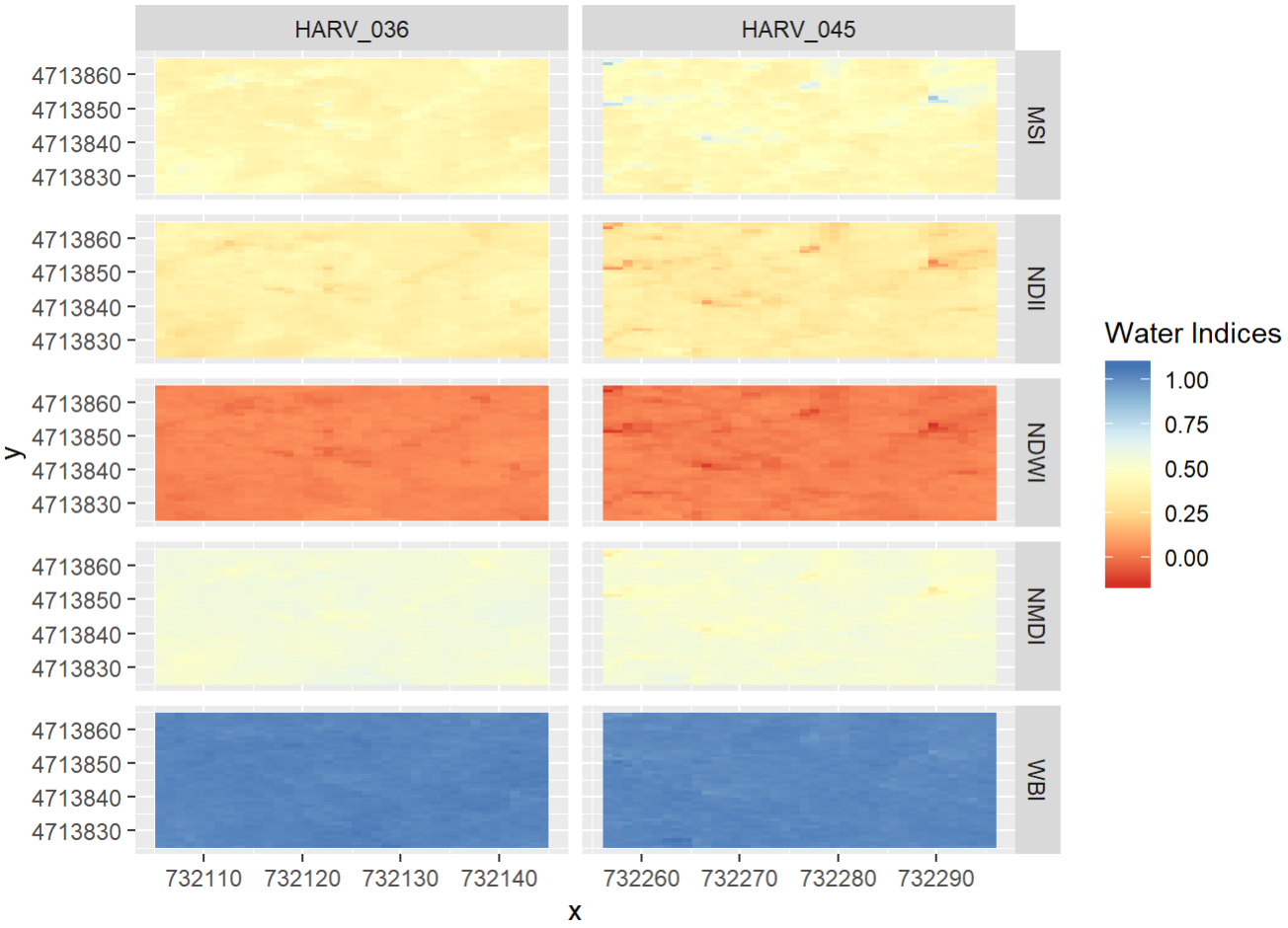
```
dim(df_plots1)
```

```
## [1] 16000     4
```

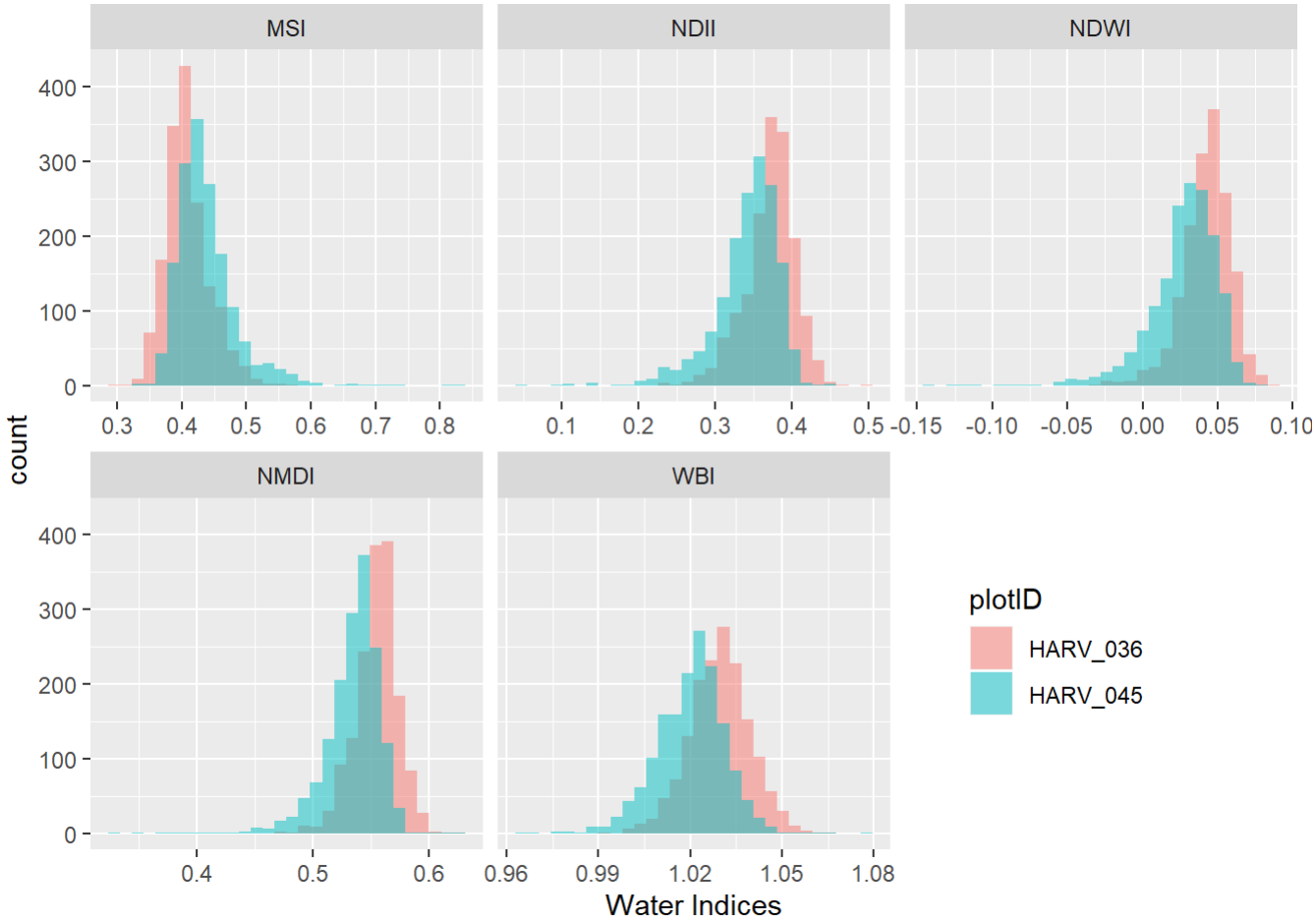
```
# add plotID column
df_plots1$plotID=c(rep("HARV_036",40*40*5),rep("HARV_045",40*40*5))
```

```
# compare all five water indices
ggplot(df_plots1)+
  geom_raster(aes(x=x,y=y,fill=value))+
  scale_fill_distiller( palette = "RdYlBu", direction = 1, name="Water Indices")+
  facet_grid(layer~plotID, scale="free_x")
```

```
## Warning in f(...): Raster pixels are placed at uneven horizontal intervals and
## will be shifted. Consider using geom_tile() instead.
```



```
# compare all five water indices in histograms
ggplot(data=df_plots1,aes(x=value, fill=plotID)) +
  geom_histogram(position="identity",alpha=0.5)+
  facet_wrap(~layer, scale="free_x")+
  theme(legend.position=c(0.8, 0.25))+
  xlab("Water Indices")
```



- Question: Can you do similar comparisons for Vegetation Indices?
- Question: Other than histograms, what else types of graphs can we use for comparisons between observatory plots?

Examine relationships

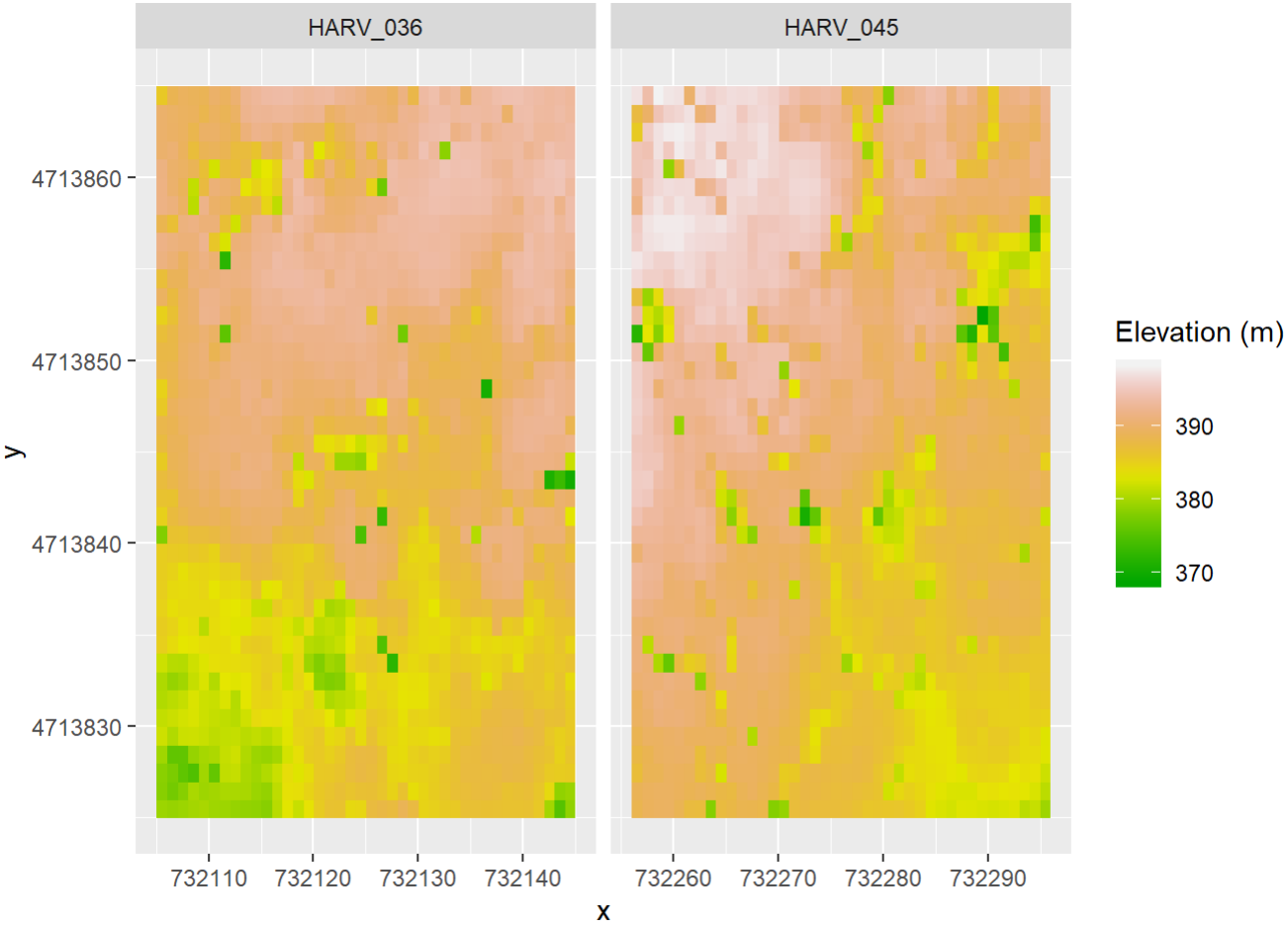
One of main goals in monitoring biological and ecological processes at NEON sites is to examine the relationships between environmental conditions and these processes, so that we can understand how enviornmental changes affect ecosystems and predict how ecosystems respond to future changes.

Here we would like to examine the relationships between topography and biological status of deciduous forests, such as canopy height, water content, etc.

We have seen the biomass and water indices of two plots. What about topography (e.g. elevation, slope and aspect)?

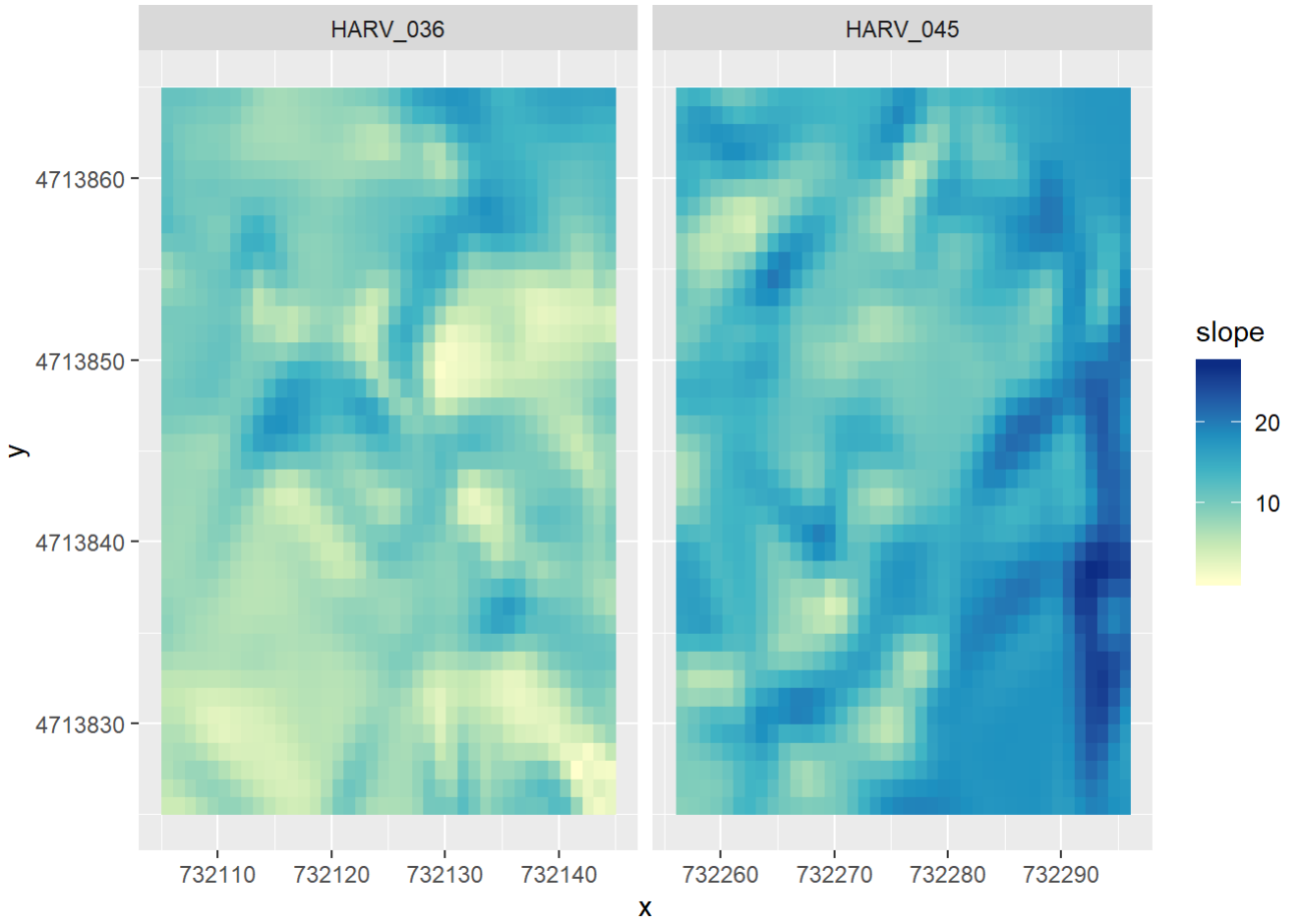
```
# graph elevation, slope and aspect of two plots

# elevation
ggplot(filter(df_plots,layer =="DSM"))+
  geom_raster(aes(x=x,y=y,fill=value))+
  scale_fill_gradientn( colours=terrain.colors(100),
    name="Elevation (m)")+
  facet_wrap(~plotID, scale="free_x")
```

```
## Warning in f(...): Raster pixels are placed at uneven horizontal intervals and
## will be shifted. Consider using geom_tile() instead.
```



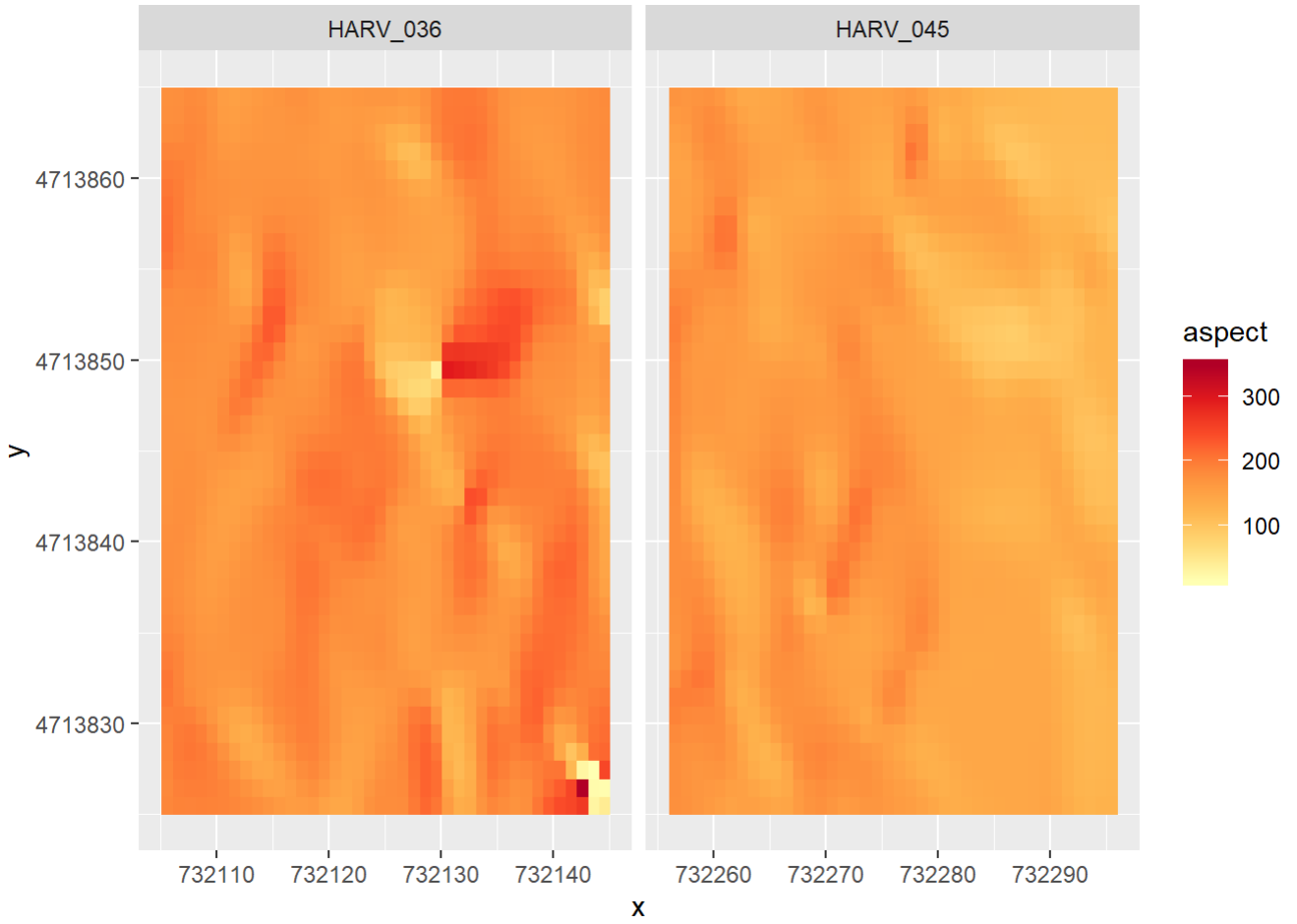

```
# slope
ggplot(filter(df_plots,layer =="slope"))+
  geom_raster(aes(x=x,y=y,fill=value))+
  scale_fill_distiller( palette = "YlGnBu", direction = 1, name="slope")+
  facet_wrap(~plotID, scale="free_x")
```

```
## Warning in f(...): Raster pixels are placed at uneven horizontal intervals and
## will be shifted. Consider using geom_tile() instead.
```




```
# aspect
ggplot(filter(df_plots,layer =="aspect"))+
  geom_raster(aes(x=x,y=y,fill=value))+
  scale_fill_distiller( palette = "YlOrRd", direction = 1, name="aspect")+
  facet_wrap(~plotID, scale="free_x")
```

```
## Warning in f(...): Raster pixels are placed at uneven horizontal intervals and
## will be shifted. Consider using geom_tile() instead.
```




```
# in order to make scatter plots, it would be better to convert the data frame from long format
 to wide format
df_plots_w=df_plots %>%
  spread(layer, value)

# combine water indices
df_plots_w=left_join(df_plots_w,spread(df_plots1,layer,value))
```

```
## Joining, by = c("x", "y", "plotID")
```

```
head(df_plots_w)
```

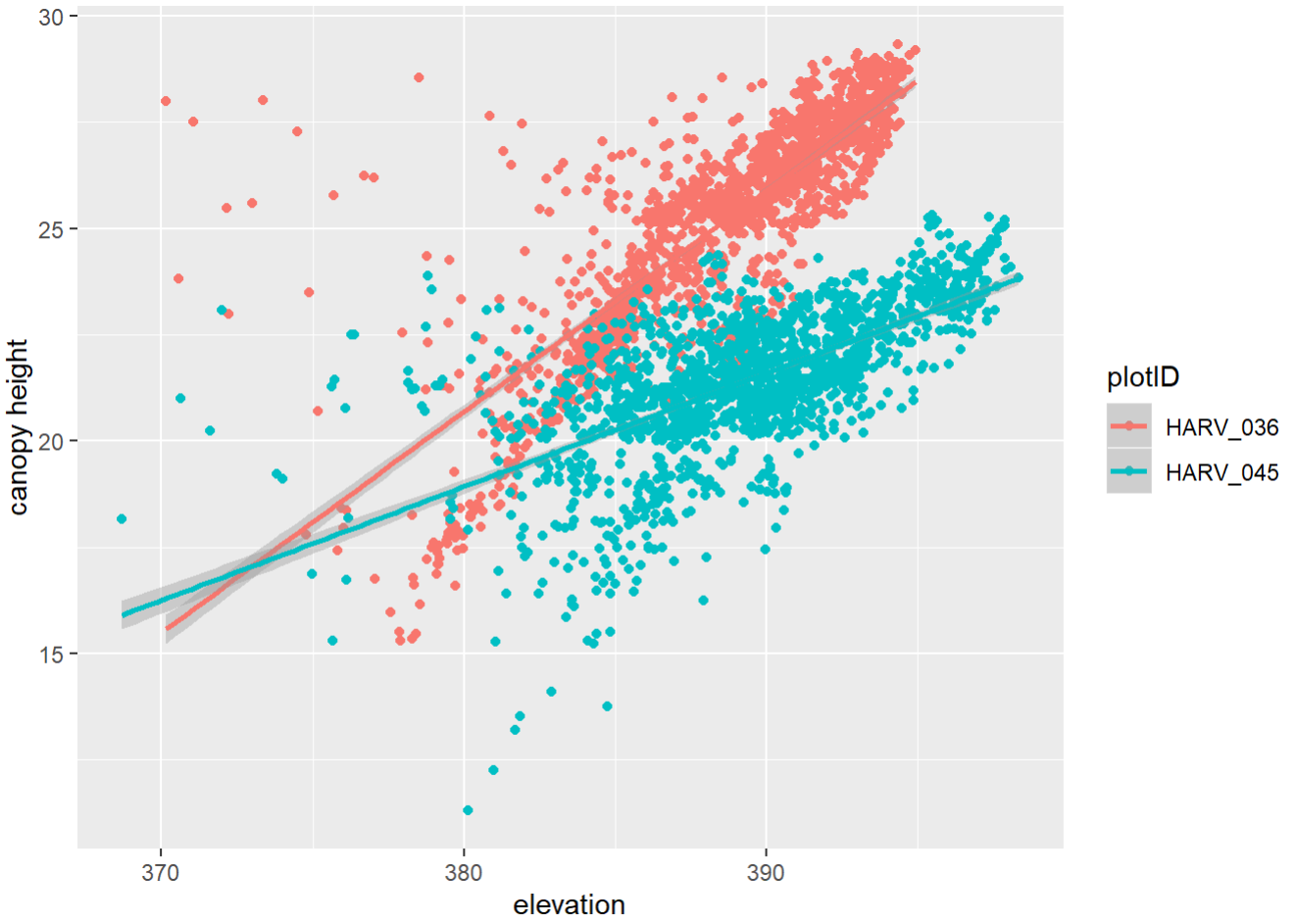
	x <dbl>	y <dbl>	plotID <chr>	aspect <dbl>	Biomass <dbl>	CHM <dbl>	DSM <dbl>	DTM <dbl>	slope <dbl>
1	732105.5	4713826	HARV_036	187.743	43.231	17.73	379.46	361.67	4.360
2	732105.5	4713827	HARV_036	185.431	44.245	16.78	378.33	361.73	5.284
3	732105.5	4713828	HARV_036	187.753	44.551	15.47	378.43	361.84	6.407
4	732105.5	4713829	HARV_036	189.588	44.133	19.28	379.69	361.95	6.517
5	732105.5	4713830	HARV_036	190.736	43.823	20.35	382.12	362.07	6.212

	x <dbl>	y <dbl>	plotID <chr>	aspect <dbl>	Biomass <dbl>	CHM <dbl>	DSM <dbl>	DTM <dbl>	slope <dbl>
6	732105.5	4713831	HARV_036	194.930	43.694	21.23	381.51	362.18	5.420

6 rows | 1-10 of 15 columns

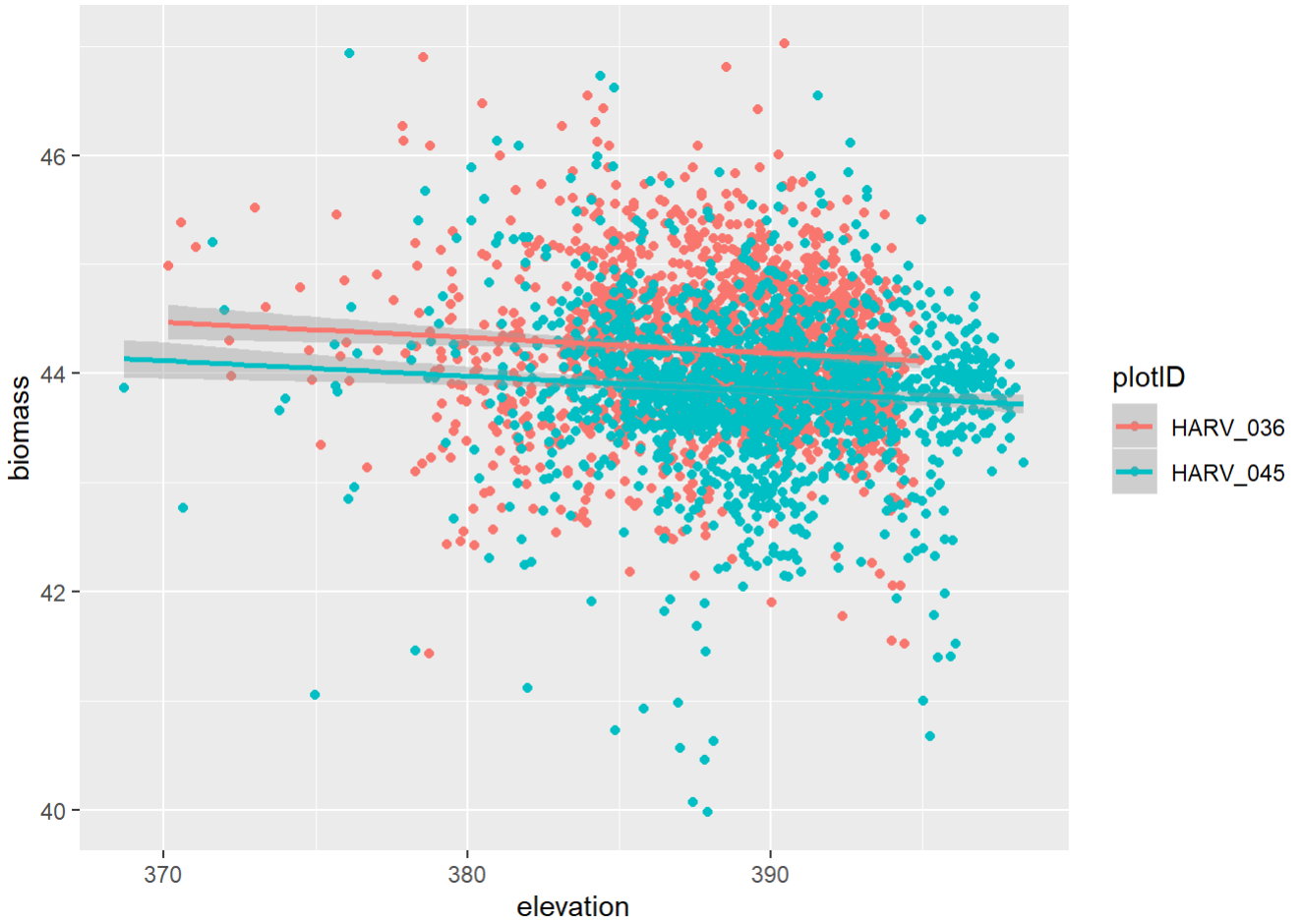
We can examine the relationships when the data frame is ready. As we have two plots at different locations, the site effect may lead to differences in the relationships. So we will group the data of two plots and use two different colors in scatter plots.

```
# relationship between elevation and canopy height
ggplot(df_plots_w,aes(x=DSM, y=CHM,col=plotID))+
  geom_point()+
  geom_smooth(method="lm")+
  xlab("elevation")+ylab("canopy height")
```



- Question: What is the relationship between elevation and canopy height at the two plots of deciduous forest? Any difference in relationship between two plots?

```
# relationship between elevation and biomass
ggplot(df_plots_w,aes(x=DSM, y=Biomass,col=plotID))+
  geom_point()+
  geom_smooth(method="lm")+
  xlab("elevation")+ylab("biomass")
```



- Question: What is the relationship between elevation and biomass at the two plots of deciduous forest? Any difference in relationship between two plots?

To quantify the relationships, you may develop simple linear regression to see the coefficient, R-square value and p-value.

```
lm1=lm(CHM~DSM,data=filter(df_plots_w,plotID=="HARV_036"))
summary(lm1)
```

```
##
## Call:
## lm(formula = CHM ~ DSM, data = filter(df_plots_w, plotID == "HARV_036"))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4626 -0.7489  0.0063  0.7689 12.4206
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.769e+02  3.591e+00  -49.26   <2e-16 ***
## DSM          5.201e-01  9.251e-03   56.22   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.514 on 1598 degrees of freedom
## Multiple R-squared:  0.6642, Adjusted R-squared:  0.664
## F-statistic:  3161 on 1 and 1598 DF,  p-value: < 2.2e-16
```

```
lm2=lm(CHM~DSM,data=filter(df_plots_w,plotID=="HARV_045"))
summary(lm2)
```

```
##
## Call:
## lm(formula = CHM ~ DSM, data = filter(df_plots_w, plotID == "HARV_045"))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.6673 -0.6530  0.0911  0.8071  6.2784
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -82.641420   3.140560  -26.31   <2e-16 ***
## DSM           0.267286   0.008065   33.14   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.361 on 1598 degrees of freedom
## Multiple R-squared:  0.4073, Adjusted R-squared:  0.407
## F-statistic:  1098 on 1 and 1598 DF,  p-value: < 2.2e-16
```

It is your turn to examine the relationship between elevation and NDWI water indices. What do you find?

Write your code below:

If you are curious, you may also examine relationships between any other variables. It would be better to have a hypothesis, so that you can test your hypothesis during examination. Be careful when you make conclusions of what you find, as correlation does not necessarily mean causation.