Production Code Draft: ‘0_LiDAR-FAIB-WSVHA-raster-to-raster-production.R’

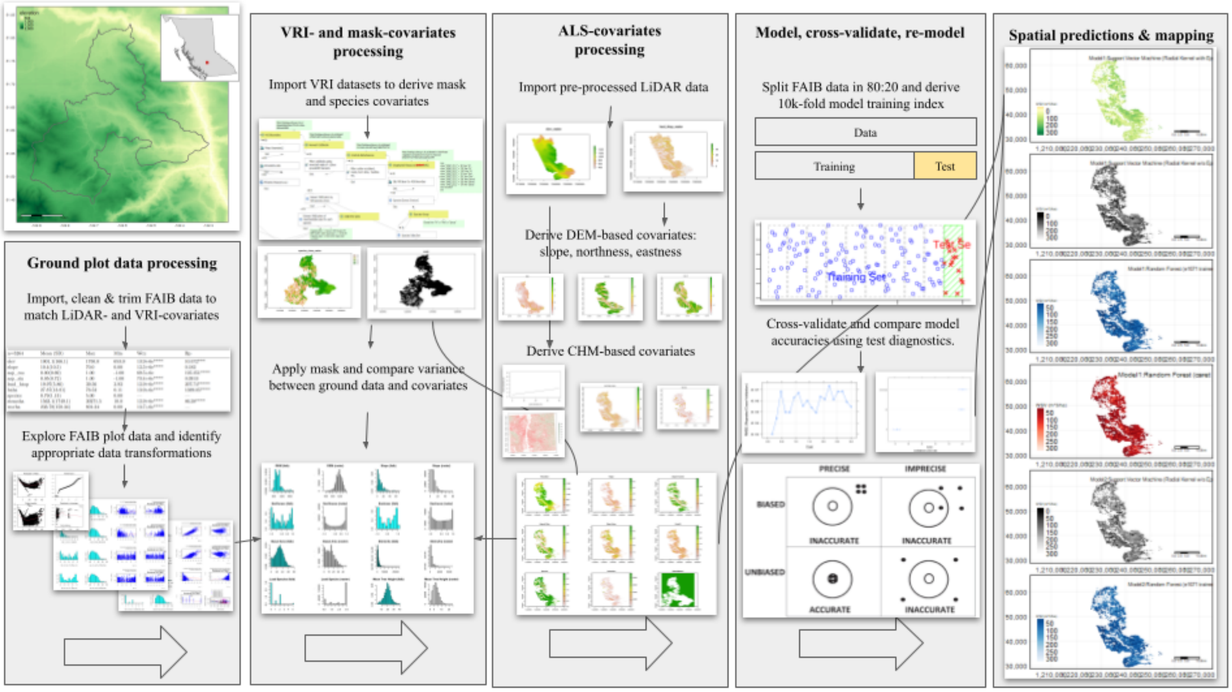
Cabin-GIS

21/05/2022

Contents

Action

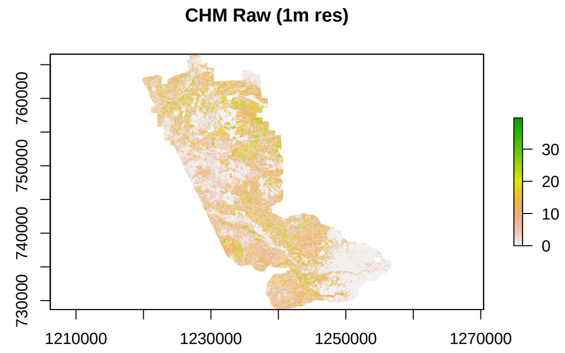
The following markdown report provides a complete run-through and guide of a raster-to-raster workflow to generating Whole Stem Volume (m^3/ha: WSVHA) raster estimates from initial phases of importing liDAR tiles, to deriving stem-detection map and a 95% canopy height model, to generating and masking DEM-based and species covariates, to fitting and training models with faib.csv data, to finally making spatial predictions using raster stack of covariates. The graphical abstract below is offered as reference guide.

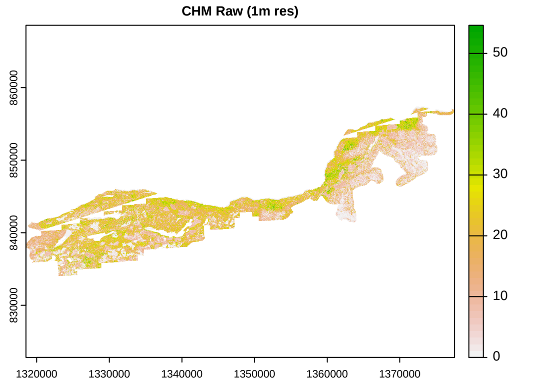

Ground plot data processing
Import, clean & trim FAIB data to match LiDAR- and VRI-covariates
Explore FAIB plot data and identify appropriate data transformations
VRI- and mask-covariates processing
Import VRI datasets to derive mask and species covariates
Apply mask and compare variance between ground data and covariates
ALS-covariates processing
Import pre-processed LiDAR data
Derive DEM-based covariates: slope, northness, eastness
Derive CHM-based covariates
Model, cross-validate, re-model
Split FAIB data in 80:20 and derive 10k-fold model training index
Data
Training
Test
Test Set
Training Set
Cross-validate and compare model accuracies using test diagnostics.
PRECISE
IMPRECISE
BIASED
UNBIASED
INACCURATE
INACCURATE
ACCURATE
INACCURATE
Spatial predictions & mapping

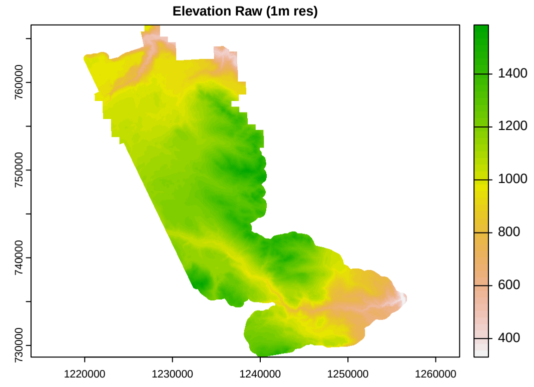
Import LiDAR: Load, list and merge chunks

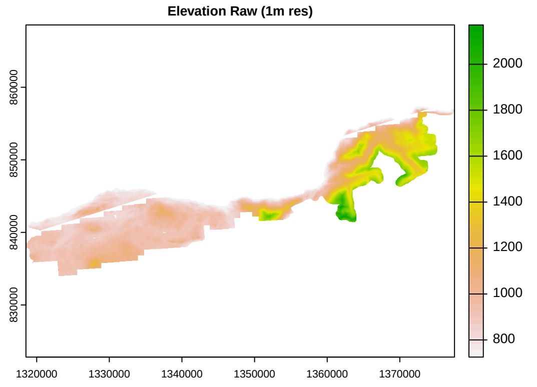
```
zip_file_vh_quesnel = ("/media/seamus/128GB_WORKD/EFI-TCC/LiDAR_Data/quesnel_region/VegHt.zip")
zip_file_be_quesnel = ("/media/seamus/128GB_WORKD/EFI-TCC/LiDAR_Data/quesnel_region/BareEarth.zip")
zip_dir_vh_quesnel = ("/media/seamus/128GB_WORKD/EFI-TCC/LiDAR_Data/quesnel_region")
zip_dir_be_quesnel = ("/media/seamus/128GB_WORKD/EFI-TCC/LiDAR_Data/quesnel_region")
unzip(zip_file_vh_quesnel, exdir = zip_dir_vh_quesnel, overwrite = TRUE)
unzip(zip_file_be_quesnel, exdir=zip_dir_be_quesnel, overwrite = TRUE)
unzip_dir_vh_quesnel <- paste0("/media/seamus/128GB_WORKD/EFI-TCC/LiDAR_Data/quesnel_region/VegHt")
unzip_dir_be_quesnel <- paste0("/media/seamus128GB_WORKD/EFI-TCC/LiDAR_Data/quesnel_region/BareEarth")
zip_file_vh_gaspard = ("/media/seamus/128GB_WORKD/EFI-TCC/LiDAR_Data/gaspard_region/VegHt.zip")
zip_file_be_gaspard = ("/media/seamus/128GB_WORKD/EFI-TCC/LiDAR_Data/gaspard_region/BareEarth.zip")
zip_dir_vh_gaspard = ("/media/seamus/128GB_WORKD/EFI-TCC/LiDAR_Data/gaspard_region")
zip_dir_be_gaspard = ("/media/seamus/128GB_WORKD/EFI-TCC/LiDAR_Data/gaspard_region")
unzip(zip_file_vh_gaspard, exdir=zip_dir_vh_gaspard, overwrite = TRUE)
unzip(zip_file_be_gaspard, exdir=zip_dir_be_gaspard, overwrite = TRUE)
unzip_dir_vh_gaspard <- paste0("/media/seamus/128GB_WORKD/EFI-TCC/LiDAR_Data/gaspard_region/VegHt")
unzip_dir_be_gaspard <- paste0("/media/seamus/128GB_WORKD/EFI-TCC/LiDAR_Data/gaspard_region/BareEarth")
filez_vh_quesnel = list.files(unzip_dir_vh_quesnel, full.names = T, all.files = FALSE, pattern = '.tif$
filez_be_quesnel = list.files(unzip_dir_be_quesnel, full.names = T, all.files = FALSE, pattern = '.tif$
filez_vh_gaspard = list.files(unzip_dir_vh_gaspard, full.names = T, all.files = FALSE, pattern = '.tif$
filez_be_gaspard = list.files(unzip_dir_be_gaspard, full.names = T, all.files = FALSE, pattern = '.tif$
lead_htop_raster_list_quesnel <- lapply(filez_vh_quesnel, raster)
lead_htop_raster_list_gaspard <- lapply(filez_vh_gaspard, raster)
elev_raster_list_quesnel <- lapply(filez_be_quesnel, raster)
elev_raster_list_gaspard <- lapply(filez_be_gaspard, raster)
lead_htop_raster_quesnel = do.call(merge, c(lead_htop_raster_list_quesnel, tolerance = 1))
lead_htop_raster_gaspard = do.call(merge, c(lead_htop_raster_list_gaspard, tolerance = 1))
elev_raster_quesnel = do.call(merge, c(elev_raster_list_quesnel, tolerance = 1))
```

```
elev_raster_gaspard = do.call(merge, c(elev_raster_list_gaspard, tolerance = 1))
writeRaster(lead_htop_raster_quesnel, filename = "/media/seamus/128GB_WORKD/EFI-TCC/LiDAR_Data/quesnel_i
writeRaster(lead_htop_raster_gaspard, filename = "/media/seamus/128GB_WORKD/EFI-TCC/LiDAR_Data/gaspard_i
writeRaster(elev_raster_quesnel, filename = "/media/seamus/128GB_WORKD/EFI-TCC/LiDAR_Data/quesnel_regior
writeRaster(elev_raster_gaspard, filename = "/media/seamus/128GB_WORKD/EFI-TCC/LiDAR_Data/gaspard_regior
```





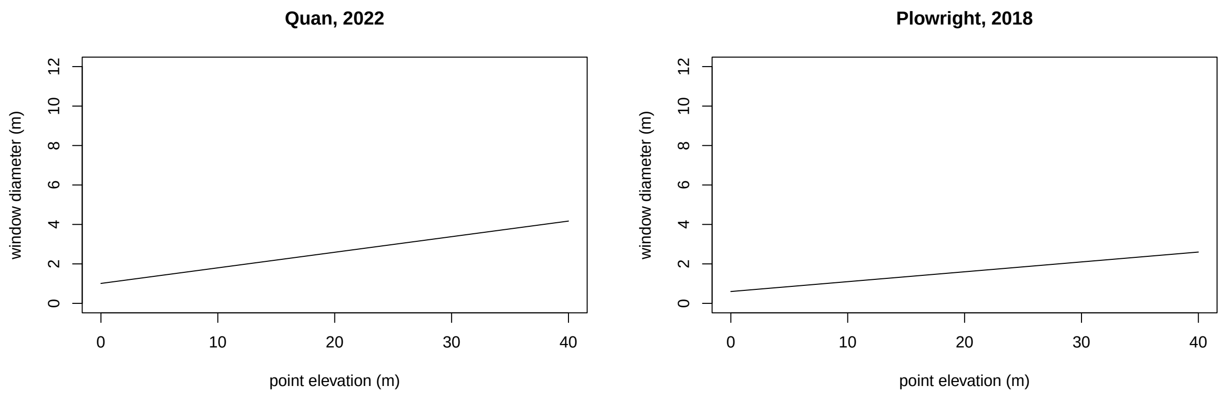




CHM-derived covariates: Variable window function

```
kernel <- matrix(1,3,3)
wf_quan<-function(x){
  a=0.179-0.1
  b=0.51+0.5
  y<-a*x+b
  return(y)}
wf_plowright<-function(x){
  a=0.05
  b=0.6
  y<-a*x+b
  return(y)}
heights <- seq(0,40,0.5)
window_quan <- wf_quan(heights)
window_plowright <- wf_plowright(heights)
```

```
plot(heights, window_quan, type = "l",  ylim = c(0,12), xlab="point elevation (m)", ylab="window diamete
plot(heights, window_plowright, type = "l", ylim = c(0,12), xlab="point elevation (m)", ylab="window dia
```



CHM-derived covariates: Stem-detection & 95% height

```
lead_htop_raster_1m_smoothed_quesnel = focal(lead_htop_rast_quesnel, w = kernel, fun = median, na.rm = 
lead_htop_raster_1m_smoothed_gaspard = focal(lead_htop_rast_gaspard, w = kernel, fun = median, na.rm = 
ttops_2m_quan_quesnel <- ForestTools::vwf(lead_htop_raster_1m_smoothed_quesnel, wf_quan, 2)
ttops_2m_quan_gaspard = ForestTools::vwf(CHM = lead_htop_raster_1m_smoothed_gaspard, winFun = wf_quan, 
ttops_2m_plowright_quesnel = ForestTools::vwf(CHM = lead_htop_raster_1m_smoothed_quesnel, winFun = wf_pl
ttops_2m_plowright_gaspard = ForestTools::vwf(CHM = lead_htop_raster_1m_smoothed_gaspard, winFun = wf_pl
writeOGR(ttops_2m_quan_quesnel, "/media/seamus/128GB_WORKD/data/vector/stem_maps", "treetops_quan_quesne
writeOGR(ttops_2m_quan_gaspard, "/media/seamus/128GB_WORKD/data/vector/stem_maps", "treetops_quan_gaspa
writeOGR(ttops_2m_plowright_quesnel, "/media/seamus/128GB_WORKD/data/vector/stem_maps", "treetops_plowr
writeOGR(ttops_2m_plowright_gaspard, "/media/seamus/128GB_WORKD/data/vector/stem_maps", "treetops_plowr

quant95 <- function(x, ...)
  quantile(x, c(0.95), na.rm = TRUE)
custFuns <- list(quant95, max)
names(custFuns) <- c("95thQuantile", "Max")

ttops_2m_quan_quesnel <- readOGR(dsn = "/media/seamus/128GB_WORKD/data/vector/stem_maps/treetops_quan_q
ttops_2m_quan_gaspard <- readOGR(dsn = "/media/seamus/128GB_WORKD/data/vector/stem_maps/treetops_quan_g
ttops_2m_Quan_raster_2m1.5m_95th_20cell_quesnel <- ForestTools::sp_summarise(ttops_2m_quan_quesnel, gri
ttops_2m_Quan_raster_2m1.5m_95th_20cell_gaspard <- ForestTools::sp_summarise(ttops_2m_quan_gaspard, gri
ttops_2m_Quan_raster_2m1.5m_95th_100cell_quesnel <- ForestTools::sp_summarise(ttops_2m_quan_quesnel, gr
ttops_2m_Quan_raster_2m1.5m_95th_100cell_gaspard <- ForestTools::sp_summarise(ttops_2m_quan_gaspard, gr
lead_htop_95th_raster_20m_quesnel = ttops_2m_Quan_raster_2m1.5m_95th_20cell_quesnel[["height95thQuantile
lead_htop_95th_raster_20m_gaspard = ttops_2m_Quan_raster_2m1.5m_95th_20cell_gaspard[["height95thQuantile
lead_htop_95th_raster_100m_quesnel = ttops_2m_Quan_raster_2m1.5m_95th_100cell_quesnel[["height95thQuant
lead_htop_95th_raster_100m_gaspard = ttops_2m_Quan_raster_2m1.5m_95th_100cell_gaspard[["height95thQuant
stemsha_L_raster_20m_quesnel = ttops_2m_Quan_raster_2m1.5m_95th_20cell_quesnel[["TreeCount"]]
stemsha_L_raster_20m_gaspard = ttops_2m_Quan_raster_2m1.5m_95th_20cell_gaspard[["TreeCount"]]
stemsha_L_raster_100m_quesnel = ttops_2m_Quan_raster_2m1.5m_95th_100cell_quesnel[["TreeCount"]]
stemsha_L_raster_100m_gaspard = ttops_2m_Quan_raster_2m1.5m_95th_100cell_gaspard[["TreeCount"]]
raster::writeRaster(lead_htop_95th_raster_20m_quesnel, filename = "/media/seamus/128GB_WORKD/data/raste
raster::writeRaster(lead_htop_95th_raster_20m_gaspard, filename = "/media/seamus/128GB_WORKD/data/raste
raster::writeRaster(lead_htop_95th_raster_100m_quesnel, filename = "/media/seamus/128GB_WORKD/data/raste
raster::writeRaster(lead_htop_95th_raster_100m_gaspard, filename = "/media/seamus/128GB_WORKD/data/raste
raster::writeRaster(stemsha_L_raster_20m_quesnel, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/
raster::writeRaster(stemsha_L_raster_20m_gaspard, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/
raster::writeRaster(stemsha_L_raster_100m_quesnel, filename = "/media/seamus/128GB_WORKD/data/raster/tcc
raster::writeRaster(stemsha_L_raster_100m_gaspard, filename = "/media/seamus/128GB_WORKD/data/raster/tcc
```

DEM-derived covariates: Terrain rasters

```
terra::crs(elev_rast_quesnel) = "epsg:3005"
terra::crs(elev_rast_gaspard) = "epsg:3005"
elev_rast_quesnel = terra::aggregate(elev_rast_quesnel, fact = 100, fun = mean)
elev_rast_gaspard = terra::aggregate(elev_rast_gaspard, fact = 100, fun = mean)
slope_rast_quesnel = terra::terrain(elev_rast_quesnel, v="slope", unit="degrees", neighbors=8)
slope_rast_gaspard = terra::terrain(elev_rast_gaspard, v="slope", unit="degrees", neighbors=8)
aspect_rast_quesnel = terra::terrain(elev_rast_quesnel, v="aspect", unit="degrees", neighbors=8)
aspect_rast_gaspard = terra::terrain(elev_rast_gaspard, v="aspect", unit="degrees", neighbors=8)
asp_cos_rast_quesnel = cos((aspect_rast_quesnel*pi)/180)
asp_cos_rast_gaspard = cos((aspect_rast_gaspard*pi)/180)
asp_sin_rast_quesnel = sin((aspect_rast_quesnel*pi)/180)
asp_sin_rast_gaspard = sin((aspect_rast_gaspard*pi)/180)

lead_htop_rast_quesnel = terra::rast(lead_htop_95th_raster_100m_quesnel)
lead_htop_rast_gaspard = terra::rast(lead_htop_95th_raster_100m_gaspard)
stemsha_L_rast_quesnel = terra::rast(stemsha_L_raster_100m_quesnel)
stemsha_L_rast_gaspard = terra::rast(stemsha_L_raster_100m_gaspard)
terra::crs(lead_htop_rast_gaspard) = "epsg:3005"
terra::crs(lead_htop_rast_quesnel) = "epsg:3005"
terra::crs(stemsha_L_rast_gaspard) = "epsg:3005"
terra::crs(stemsha_L_rast_quesnel) = "epsg:3005"

lead_htop_rast_quesnel = terra::resample(lead_htop_rast_quesnel, elev_rast_quesnel)
lead_htop_rast_gaspard = terra::resample(lead_htop_rast_gaspard, elev_rast_gaspard)
stemsha_L_rast_quesnel = terra::resample(stemsha_L_rast_quesnel, elev_rast_quesnel)
stemsha_L_rast_gaspard = terra::resample(stemsha_L_rast_gaspard, elev_rast_gaspard)
elev_rast_quesnel = terra::mask(elev_rast_quesnel, lead_htop_rast_quesnel)
elev_rast_gaspard = terra::mask(elev_rast_gaspard, lead_htop_rast_gaspard)
slope_rast_quesnel = terra::mask(slope_rast_quesnel, lead_htop_rast_quesnel)
slope_rast_gaspard = terra::mask(slope_rast_gaspard, lead_htop_rast_gaspard)
asp_cos_rast_quesnel = terra::mask(asp_cos_rast_quesnel, lead_htop_rast_quesnel)
asp_cos_rast_gaspard = terra::mask(asp_cos_rast_gaspard, lead_htop_rast_gaspard)
asp_sin_rast_quesnel = terra::mask(asp_sin_rast_quesnel, lead_htop_rast_quesnel)
asp_sin_rast_gaspard = terra::mask(asp_sin_rast_gaspard, lead_htop_rast_gaspard)

writeRaster(elev_rast_quesnel, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/unmasked-covariates
writeRaster(elev_rast_gaspard, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/unmasked-covariates
writeRaster(slope_rast_quesnel, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/unmasked-covariate
writeRaster(slope_rast_gaspard, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/unmasked-covariate
writeRaster(asp_cos_rast_quesnel, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/unmasked-covaria
writeRaster(asp_cos_rast_gaspard, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/unmasked-covaria
writeRaster(asp_sin_rast_quesnel, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/unmasked-covaria
writeRaster(asp_sin_rast_gaspard, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/unmasked-covaria
```

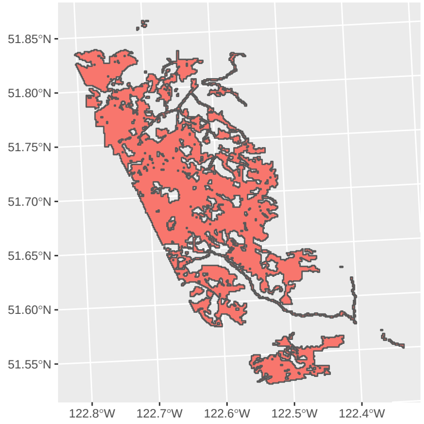
Species covariates: VRI rasterization

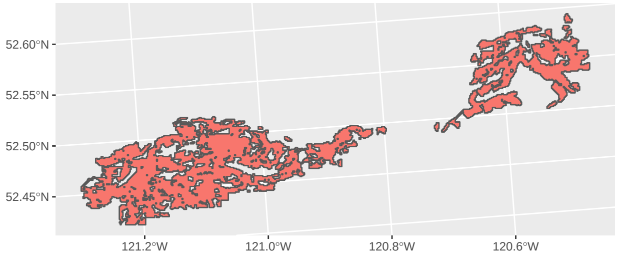
```
lead_htop_sv_quesnel = as.polygons(lead_htop_rast_quesnel)
lead_htop_sv_gaspard = as.polygons(lead_htop_rast_gaspard)
lead_htop_sf_quesnel = sf::st_as_sf(lead_htop_sv_quesnel)
lead_htop_sf_gaspard = sf::st_as_sf(lead_htop_sv_gaspard)
vri_sf = read_sf("/media/seamus/128GB_WORKD/data/vector/vri/vri_bc_2020_rank1.shp")
vri_species = vri_sf[c("SPECIES__1", "SPECIES_CD", "SPECIES_PC")]
vri_species_aoi =  dplyr::filter(vri_species,
    SPECIES__1=='PL' | SPECIES__1=='PLI' | SPECIES__1=='FD' | SPECIES__1=='FDI' |
    SPECIES__1=='SB' | SPECIES__1=='SE' | SPECIES__1=='SW' | SPECIES__1=='SX' |
    SPECIES__1=='CW' | SPECIES__1=='HW' | SPECIES__1=='BL' | SPECIES__1=='LW')
vri_species_aoi_gaspard = vri_species_aoi[!(
  vri_species_aoi$SPECIES__1 == 'FD' & vri_species_aoi$SPECIES_PC >= 50 |
    vri_species_aoi$SPECIES__1 == 'FDI' & vri_species_aoi$SPECIES_PC >=50),]
vri_species_aoi_gaspard$SPECIES__1 = dplyr::recode(vri_species_aoi_gaspard$SPECIES__1,
  PL = 0, PLI = 0, SB = 1, SE = 1, SW = 1, SX = 1, FD = 2, FDI = 2, CW = 3, HW = 4, BL = 5, LW = 6)
vri_species_aoi_gaspard = dplyr::rename(vri_species_aoi_gaspard, species_class = SPECIES__1)
vri_species_aoi_gaspard = vri_species_aoi_gaspard["species_class"]
vri_species_aoi_sf_gaspard = sf::st_as_sf(vri_species_aoi_gaspard)

vri_species_aoi_quesnel = vri_species_aoi
vri_species_aoi_quesnel$SPECIES__1 = dplyr::recode(vri_species_aoi$SPECIES__1, PL = 0, PLI = 0, SB = 1,
vri_species_aoi_quesnel = dplyr::rename(vri_species_aoi_quesnel, species_class = SPECIES__1)
vri_species_aoi_quesnel = vri_species_aoi_quesnel["species_class"]
vri_species_aoi_sf_quesnel = sf::st_as_sf(vri_species_aoi_quesnel)

vri_species_aoi_quesnel = st_intersection(vri_species_aoi_sf_quesnel, st_make_valid(lead_htop_sf_quesnel
vri_species_aoi_gaspard = st_intersection(vri_species_aoi_sf_gaspard, st_make_valid(lead_htop_sf_gaspar
species_class_rast_quesnel = terra::rasterize(vect(vri_species_aoi_quesnel), lead_htop_rast_quesnel, fi
species_class_rast_gaspard = terra::rasterize(vect(vri_species_aoi_gaspard), lead_htop_rast_gaspard, fi
species_class_rast_quesnel = terra::resample(species_class_rast_quesnel, lead_htop_rast_quesnel)
species_class_rast_gaspard = terra::resample(species_class_rast_gaspard, lead_htop_rast_gaspard)
species_class_raster_quesnel = raster::raster(species_class_rast_quesnel)
species_class_raster_gaspard = raster::raster(species_class_rast_gaspard)
raster::writeRaster(species_class_raster_quesnel, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/
raster::writeRaster(species_class_raster_gaspard, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/
```

Masking: Generate and merge mask layers

```
mask_burn2017 = sf::read_sf("/media/seamus/128GB_WORKD/data/vector/tcc_mask_layers/TCC_Burn_Severity TC
mask_burn2018 = sf::read_sf("/media/seamus/128GB_WORKD/data/vector/tcc_mask_layers/TCC_Burn_Severity TC
mask_burn2021 = sf::read_sf("/media/seamus/128GB_WORKD/data/vector/tcc_mask_layers/TCC_Burn_Severity TC
mask_burn2017 = mask_burn2017["BurnSev"]
mask_burn2018 = mask_burn2018["BurnSev"]
mask_burn2021 = mask_burn2021["BurnSev"]
mask_burn2017 = dplyr::filter(mask_burn2017, BurnSev == 'High')
```

```
mask_burn2018 = dplyr::filter(mask_burn2018, BurnSev == 'High')
mask_burn2021 = dplyr::filter(mask_burn2021, BurnSev == 'High')
mask_burn2017_quesnel = sf::st_intersection(sf::st_make_valid(mask_burn2017), lead_htop_sf_quesnel)
mask_burn2017_gaspard = sf::st_intersection(sf::st_make_valid(mask_burn2017), lead_htop_sf_gaspard)
mask_burn2018_quesnel = sf::st_intersection(sf::st_make_valid(mask_burn2018), lead_htop_sf_quesnel)
mask_burn2018_gaspard = sf::st_intersection(sf::st_make_valid(mask_burn2018), lead_htop_sf_gaspard)
mask_burn2021_quesnel = sf::st_intersection(sf::st_make_valid(mask_burn2021), lead_htop_sf_quesnel)
mask_burn2021_gaspard = sf::st_intersection(sf::st_make_valid(mask_burn2021), lead_htop_sf_gaspard)
masks_df_quesnel = full_join(as_tibble(mask_burn2017_quesnel), as_tibble(mask_burn2018_quesnel), as_tibl
masks_df_gaspard = full_join(as_tibble(mask_burn2017_gaspard), as_tibble(mask_burn2018_gaspard), as_tibl
masks_sf_quesnel = st_as_sf(masks_df_quesnel) # easier to combine by 'geometry'
masks_sf_gaspard = st_as_sf(masks_df_gaspard) # easier to combine by 'geometry'

mask_clearcut = sf::read_sf("/media/seamus/128GB_WORKD/data/vector/tcc_mask_layers/RSLT_CCRES_CLEAR.shp
mask_clearcut_quesnel = sf::st_intersection(mask_clearcut, st_make_valid(lead_htop_sf_quesnel))
mask_clearcut_gaspard = sf::st_intersection(mask_clearcut, st_make_valid(lead_htop_sf_gaspard))
masks_df_quesnel = full_join(as_tibble(masks_sf_quesnel), as_tibble(mask_clearcut_quesnel), by = 'geome
masks_df_gaspard = full_join(as_tibble(masks_sf_gaspard), as_tibble(mask_clearcut_gaspard), by = 'geome
masks_sf_quesnel = st_as_sf(masks_df_quesnel)
masks_sf_gaspard = st_as_sf(masks_df_gaspard)

mask_blocks = sf::read_sf("/media/seamus/128GB_WORKD/data/vector/tcc_mask_layers/TCC_Blocks_Join.shp")
mask_blocks_quesnel = sf::st_intersection(mask_blocks, st_make_valid(lead_htop_sf_quesnel))
mask_blocks_gaspard = sf::st_intersection(mask_blocks, st_make_valid(lead_htop_sf_gaspard))
masks_df_quesnel = full_join(as_tibble(masks_sf_quesnel), as_tibble(mask_blocks_quesnel), by = 'geometry
masks_df_gaspard = full_join(as_tibble(masks_sf_gaspard), as_tibble(mask_blocks_gaspard), by = 'geometry
masks_sf_quesnel = st_as_sf(masks_df_quesnel)
masks_sf_gaspard = st_as_sf(masks_df_gaspard)

mask_roads_tcc = sf::read_sf("/media/seamus/128GB_WORKD/data/vector/tcc_mask_layers/TCC_Roads.shp")
mask_roads_tcc = sf::st_zm(mask_roads_tcc)
mask_roads_tcc_quesnel = sf::st_intersection(mask_roads_tcc, st_make_valid(lead_htop_sf_quesnel))
mask_roads_tcc_gaspard = sf::st_intersection(mask_roads_tcc, st_make_valid(lead_htop_sf_gaspard))
mask_roads_tcc_quesnel = sf::st_buffer(mask_roads_tcc_quesnel, dist = 15, nQuadSegs = 5, endCapStyle =
mask_roads_tcc_gaspard = sf::st_buffer(mask_roads_tcc_gaspard, dist = 15, nQuadSegs = 5, endCapStyle =

mask_roads_ften = sf::read_sf("/media/seamus/128GB_WORKD/data/vector/tcc_mask_layers/FTEN_Roads_All.shp
mask_roads_ften = sf::st_zm(mask_roads_ften)
mask_roads_ften_quesnel = sf::st_intersection(mask_roads_ften, st_make_valid(lead_htop_sf_quesnel))
mask_roads_ften_gaspard = sf::st_intersection(mask_roads_ften, st_make_valid(lead_htop_sf_gaspard))
mask_roads_ften_quesnel = sf::st_buffer(mask_roads_ften_quesnel, dist = 15, nQuadSegs = 5, endCapStyle =
mask_roads_ften_gaspard = sf::st_buffer(mask_roads_ften_gaspard, dist = 15, nQuadSegs = 5, endCapStyle =
masks_df_quesnel = full_join(as_tibble(masks_sf_quesnel), as_tibble(mask_roads_tcc_quesnel), as_tibble(
masks_df_gaspard = full_join(as_tibble(masks_sf_gaspard), as_tibble(mask_roads_tcc_gaspard), as_tibble(
masks_sf_quesnel = st_as_sf(masks_df_quesnel)
masks_sf_gaspard = st_as_sf(masks_df_gaspard)
masks_rast_quesnel = rasterize(vect(masks_sf_quesnel), lead_htop_rast_quesnel, touches = TRUE)
masks_rast_gaspard = rasterize(vect(masks_sf_gaspard), lead_htop_rast_gaspard, touches = TRUE)
masks_raster_quesnel = raster::raster(masks_rast_quesnel)
masks_raster_gaspard = raster::raster(masks_rast_gaspard)
writeRaster(masks_raster_quesnel, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/mask/mask_raste
writeRaster(masks_raster_gaspard, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/mask/mask_raste
ggplot(masks_sf_gaspard) + geom_sf(aes(fill = 'red'), show.legend = FALSE)
ggplot(masks_sf_quesnel) + geom_sf(aes(fill = 'red'), show.legend = FALSE)
```





Masking: Apply masking

```
elev_rast_quesnel = terra::rast("/media/seamus/128GB_WORKD/data/raster/tcc/unmasked-covariates/elev_ras
elev_rast_gaspard = terra::rast("/media/seamus/128GB_WORKD/data/raster/tcc/unmasked-covariates/elev_ras
slope_rast_quesnel = terra::rast("/media/seamus/128GB_WORKD/data/raster/tcc/unmasked-covariates/slope_ra
slope_rast_gaspard = terra::rast("/media/seamus/128GB_WORKD/data/raster/tcc/unmasked-covariates/slope_ra
asp_cos_rast_quesnel = terra::rast("/media/seamus/128GB_WORKD/data/raster/tcc/unmasked-covariates/asp_c
asp_cos_rast_gaspard = terra::rast("/media/seamus/128GB_WORKD/data/raster/tcc/unmasked-covariates/asp_c
asp_sin_rast_quesnel = terra::rast("/media/seamus/128GB_WORKD/data/raster/tcc/unmasked-covariates/asp_s
asp_sin_rast_gaspard = terra::rast("/media/seamus/128GB_WORKD/data/raster/tcc/unmasked-covariates/asp_s
lead_htop_rast_quesnel = terra::rast("/media/seamus/128GB_WORKD/data/raster/tcc/unmasked-covariates/lea
lead_htop_rast_gaspard = terra::rast("/media/seamus/128GB_WORKD/data/raster/tcc/unmasked-covariates/lea
stemsha_L_rast_quesnel = terra::rast("/media/seamus/128GB_WORKD/data/raster/tcc/unmasked-covariates/ste
stemsha_L_rast_gaspard = terra::rast("/media/seamus/128GB_WORKD/data/raster/tcc/unmasked-covariates/ste
species_class_rast_quesnel = terra::rast("/media/seamus/128GB_WORKD/data/raster/tcc/unmasked-covariates
species_class_rast_gaspard = terra::rast("/media/seamus/128GB_WORKD/data/raster/tcc/unmasked-covariates
masks_rast_quesnel = terra::rast("/media/seamus/128GB_WORKD/data/raster/tcc//mask/mask_raster_100m_ques
masks_rast_gaspard = terra::rast("/media/seamus/128GB_WORKD/data/raster/tcc/mask/mask_raster_100m_gaspa
# masking by stem map
lead_htop_rast_quesnel = terra::resample(lead_htop_rast_quesnel, elev_rast_quesnel)
lead_htop_rast_gaspard = terra::resample(lead_htop_rast_gaspard, elev_rast_gaspard)
stemsha_L_rast_quesnel = terra::resample(stemsha_L_rast_quesnel, elev_rast_quesnel)
stemsha_L_rast_gaspard = terra::resample(stemsha_L_rast_gaspard, elev_rast_gaspard)
elev_rast_quesnel = terra::mask(elev_rast_quesnel, lead_htop_rast_quesnel)
elev_rast_gaspard = terra::mask(elev_rast_gaspard, lead_htop_rast_gaspard)
slope_rast_quesnel = terra::mask(slope_rast_quesnel, lead_htop_rast_quesnel)
slope_rast_gaspard = terra::mask(slope_rast_gaspard, lead_htop_rast_gaspard)
asp_cos_rast_quesnel = terra::mask(asp_cos_rast_quesnel, lead_htop_rast_quesnel)
asp_cos_rast_gaspard = terra::mask(asp_cos_rast_gaspard, lead_htop_rast_gaspard)
asp_sin_rast_quesnel = terra::mask(asp_sin_rast_quesnel, lead_htop_rast_quesnel)
asp_sin_rast_gaspard = terra::mask(asp_sin_rast_gaspard, lead_htop_rast_gaspard)
species_class_rast_quesnel = terra::mask(species_class_rast_quesnel, lead_htop_rast_quesnel)
species_class_rast_gaspard = terra::mask(species_class_rast_gaspard, lead_htop_rast_gaspard)
# masking by mask
masks_rast_quesnel = terra::resample(masks_rast_quesnel, lead_htop_rast_quesnel)
```

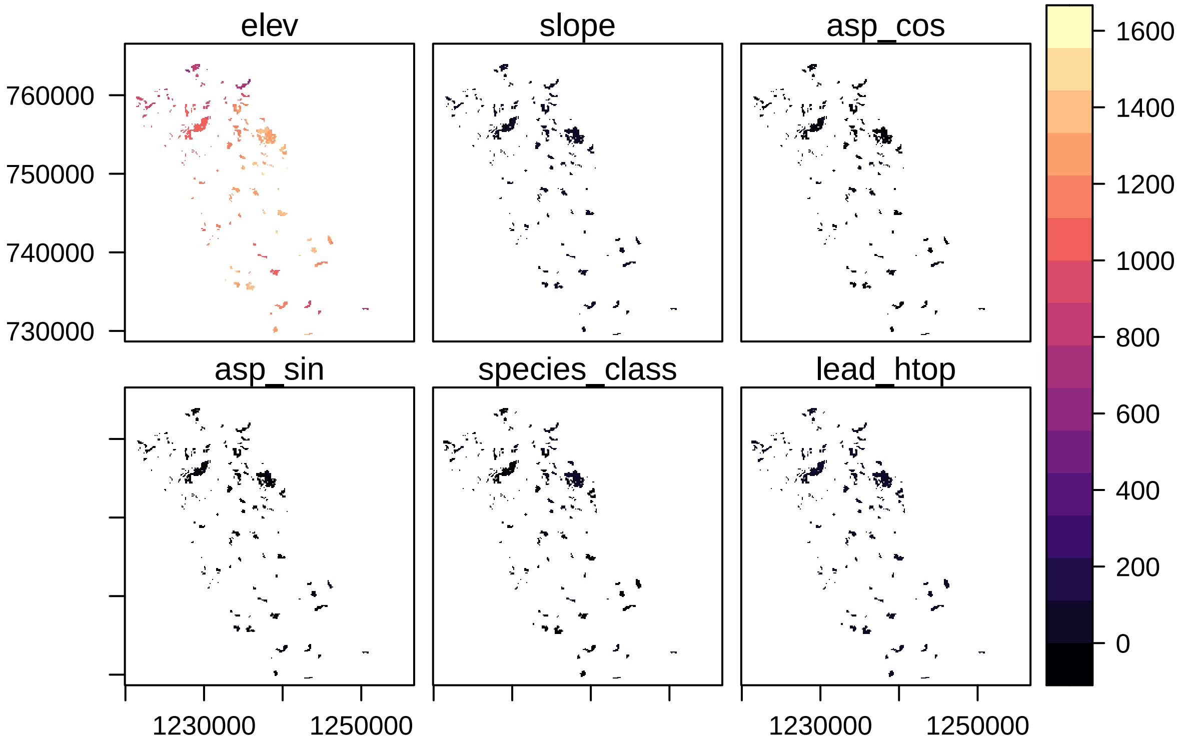
```
masks_rast_gaspard = terra::resample(masks_rast_gaspard, lead_htop_rast_gaspard)
masks_rast_quesnel = terra::resample(masks_rast_quesnel, elev_rast_quesnel)
masks_rast_gaspard = terra::resample(masks_rast_gaspard, elev_rast_gaspard)
lead_htop_rast_quesnel = mask(lead_htop_rast_quesnel, masks_rast_quesnel, inverse=TRUE)
lead_htop_rast_gaspard = mask(lead_htop_rast_gaspard, masks_rast_gaspard, inverse=TRUE)
elev_rast_quesnel = mask(elev_rast_quesnel, masks_rast_quesnel, inverse=TRUE)
elev_rast_gaspard = mask(elev_rast_gaspard, masks_rast_gaspard, inverse=TRUE)
slope_rast_quesnel = mask(slope_rast_quesnel, masks_rast_quesnel, inverse=TRUE)
slope_rast_gaspard = mask(slope_rast_gaspard, masks_rast_gaspard, inverse=TRUE)
asp_cos_rast_quesnel = mask(asp_cos_rast_quesnel, masks_rast_quesnel, inverse=TRUE)
asp_cos_rast_gaspard = mask(asp_cos_rast_gaspard, masks_rast_gaspard, inverse=TRUE)
asp_sin_rast_quesnel = mask(asp_sin_rast_quesnel, masks_rast_quesnel, inverse=TRUE)
asp_sin_rast_gaspard = mask(asp_sin_rast_gaspard, masks_rast_gaspard, inverse=TRUE)
stemsha_L_rast_quesnel = mask(stemsha_L_rast_quesnel, masks_rast_quesnel, inverse=TRUE)
stemsha_L_rast_gaspard = mask(stemsha_L_rast_gaspard, masks_rast_gaspard, inverse=TRUE)
species_class_rast_quesnel = mask(species_class_rast_quesnel, masks_rast_quesnel, inverse=TRUE)
species_class_rast_gaspard = mask(species_class_rast_gaspard, masks_rast_gaspard, inverse=TRUE)
# masking by species
lead_htop_rast_quesnel = mask(lead_htop_rast_quesnel, species_class_rast_quesnel, inverse=FALSE)
lead_htop_rast_gaspard = mask(lead_htop_rast_gaspard, species_class_rast_gaspard, inverse=FALSE)
elev_rast_quesnel = mask(elev_rast_quesnel, species_class_rast_quesnel, inverse=FALSE)
elev_rast_gaspard = mask(elev_rast_gaspard, species_class_rast_gaspard, inverse=FALSE)
slope_rast_quesnel = mask(slope_rast_quesnel, species_class_rast_quesnel, inverse=FALSE)
slope_rast_gaspard = mask(slope_rast_gaspard, species_class_rast_gaspard, inverse=FALSE)
asp_cos_rast_quesnel = mask(asp_cos_rast_quesnel, species_class_rast_quesnel, inverse=FALSE)
asp_cos_rast_gaspard = mask(asp_cos_rast_gaspard, species_class_rast_gaspard, inverse=FALSE)
asp_sin_rast_quesnel = mask(asp_sin_rast_quesnel, species_class_rast_quesnel, inverse=FALSE)
asp_sin_rast_gaspard = mask(asp_sin_rast_gaspard, species_class_rast_gaspard, inverse=FALSE)
stemsha_L_rast_quesnel = mask(stemsha_L_rast_quesnel, species_class_rast_quesnel, inverse=FALSE)
stemsha_L_rast_gaspard = mask(stemsha_L_rast_gaspard, species_class_rast_gaspard, inverse=FALSE)

writeRaster(elev_rast_quesnel, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/masked-covariates/
writeRaster(elev_rast_gaspard, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/masked-covariates/
writeRaster(slope_rast_quesnel, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/masked-covariates
writeRaster(slope_rast_gaspard, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/masked-covariates
writeRaster(asp_cos_rast_quesnel, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/masked-covariat
writeRaster(asp_cos_rast_gaspard, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/masked-covariat
writeRaster(asp_sin_rast_quesnel, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/masked-covariat
writeRaster(asp_sin_rast_gaspard, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/masked-covariat
writeRaster(species_class_rast_quesnel, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/masked-co
writeRaster(species_class_rast_gaspard, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/masked-co
writeRaster(stemsha_L_rast_quesnel, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/masked-covari
writeRaster(stemsha_L_rast_gaspard, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/masked-covari
writeRaster(lead_htop_rast_quesnel, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/masked-covari
writeRaster(lead_htop_rast_gaspard, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/masked-covari
```

Tidy: Stack covariates

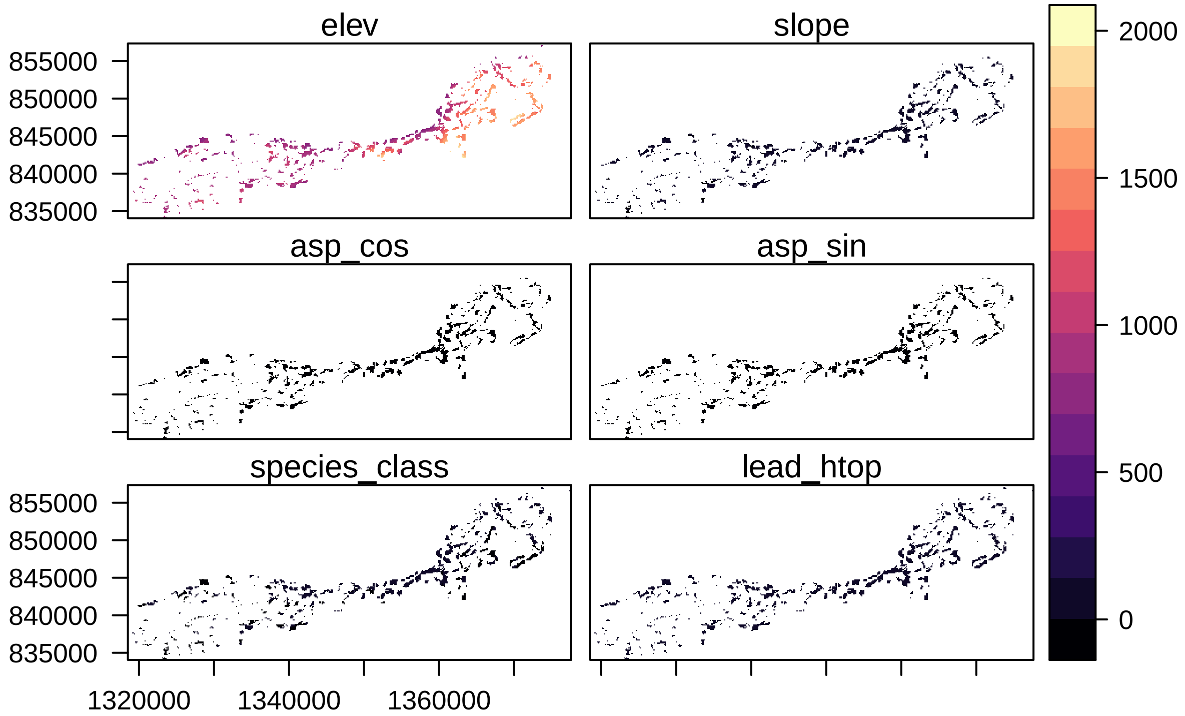
```
#tidy names
names(elev_rast_quesnel) = "elev"
names(slope_rast_quesnel) = "slope"
```

```r
names(asp_cos_rast_quesnel) = "asp_cos"
names(asp_sin_rast_quesnel) = "asp_sin"
names(species_class_rast_quesnel) = "species_class"
names(stemsha_L_rast_quesnel) = "stemsha_L"
names(lead_htop_rast_quesnel) = "lead_htop"

names(elev_rast_gaspard) = "elev"
names(slope_rast_gaspard) = "slope"
names(asp_cos_rast_gaspard) = "asp_cos"
names(asp_sin_rast_gaspard) = "asp_sin"
names(species_class_rast_gaspard) = "species_class"
names(stemsha_L_rast_gaspard) = "stemsha_L"
names(lead_htop_rast_gaspard) = "lead_htop"

elev_raster_quesnel = raster::raster(elev_rast_quesnel)
slope_raster_quesnel = raster::raster(slope_rast_quesnel)
asp_cos_raster_quesnel = raster::raster(asp_cos_rast_quesnel)
asp_sin_raster_quesnel = raster::raster(asp_sin_rast_quesnel)
species_class_raster_quesnel = raster::raster(species_class_rast_quesnel)
stemsha_L_raster_quesnel = raster::raster(stemsha_L_rast_quesnel)
lead_htop_raster_quesnel = raster::raster(lead_htop_rast_quesnel)

elev_raster_gaspard = raster::raster(elev_rast_gaspard)
slope_raster_gaspard = raster::raster(slope_rast_gaspard)
asp_cos_raster_gaspard = raster::raster(asp_cos_rast_gaspard)
asp_sin_raster_gaspard = raster::raster(asp_sin_rast_gaspard)
species_class_raster_gaspard = raster::raster(species_class_rast_gaspard)
stemsha_L_raster_gaspard = raster::raster(stemsha_L_rast_gaspard)
lead_htop_raster_gaspard = raster::raster(lead_htop_rast_gaspard)

elev_raster_list = list(elev_raster_quesnel, elev_raster_gaspard)
slope_raster_list = list(slope_raster_quesnel, slope_raster_gaspard)
asp_cos_raster_list = list(asp_cos_raster_quesnel, asp_cos_raster_gaspard)
asp_sin_raster_list = list(asp_sin_raster_quesnel, asp_sin_raster_gaspard)
species_class_raster_list = list(species_class_raster_quesnel, species_class_raster_gaspard)
stemsha_L_raster_list = list(stemsha_L_raster_quesnel, stemsha_L_raster_gaspard)
lead_htop_raster_list = list(lead_htop_raster_quesnel, lead_htop_raster_gaspard)

elev_raster = do.call(merge, c(elev_raster_list, tolerance = 1))
slope_raster = do.call(merge, c(slope_raster_list, tolerance = 1))
asp_cos_raster = do.call(merge, c(asp_cos_raster_list, tolerance = 1))
asp_sin_raster = do.call(merge, c(asp_sin_raster_list, tolerance = 1))
species_class_raster = do.call(merge, c(species_class_raster_list, tolerance = 1))
stemsha_L_raster = do.call(merge, c(stemsha_L_raster_list, tolerance = 1))
lead_htop_raster = do.call(merge, c(lead_htop_raster_list, tolerance = 1))

names(elev_raster) = "elev"
names(slope_raster) = "slope"
names(asp_cos_raster) = "asp_cos"
names(asp_sin_raster) = "asp_sin"
names(species_class_raster) = "species_class"
names(stemsha_L_raster) = "stemsha_L"
names(lead_htop_raster) = "lead_htop"
```

```
writeRaster(elev_raster, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/masked-covariates/elev_ra
writeRaster(slope_raster, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/masked-covariates/slope_
writeRaster(asp_cos_raster, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/masked-covariates/asp_
writeRaster(asp_sin_raster, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/masked-covariates/asp_
writeRaster(species_class_raster, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/masked-covariate
writeRaster(stemsha_L_raster, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/masked-covariates/st
writeRaster(lead_htop_raster, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/masked-covariates/le

covs_m1_quesnel = raster::stack(
  elev_raster_quesnel,
  slope_raster_quesnel,
  asp_cos_raster_quesnel,
  asp_sin_raster_quesnel,
  species_class_raster_quesnel,
  lead_htop_raster_quesnel)

covs_m2_quesnel = raster::stack(
  elev_raster_quesnel,
  slope_raster_quesnel,
  asp_cos_raster_quesnel,
  asp_sin_raster_quesnel,
  species_class_raster_quesnel,
  lead_htop_raster_quesnel,
  stemsha_L_raster_quesnel)

covs_m1_gaspard = raster::stack(
  elev_raster_gaspard,
  slope_raster_gaspard,
  asp_cos_raster_gaspard,
  asp_sin_raster_gaspard,
  species_class_raster_gaspard,
  lead_htop_raster_gaspard)

covs_m2_gaspard = raster::stack(
  elev_raster_gaspard,
  slope_raster_gaspard,
  asp_cos_raster_gaspard,
  asp_sin_raster_gaspard,
  species_class_raster_gaspard,
  lead_htop_raster_gaspard,
  stemsha_L_raster_gaspard)

covs_m1 = raster::stack(
  elev_raster,
  slope_raster,
  asp_cos_raster,
  asp_sin_raster,
  lead_htop_raster,
  species_class_raster)

covs_m2 = raster::stack(
  elev_raster,
  slope_raster,
  asp_cos_raster,
  asp_sin_raster,
  lead_htop_raster,
  species_class_raster,
  stemsha_L_raster)

rasterVis::levelplot(covs_m1_gaspard)
```



```
rasterVis::levelplot(covs_m1_quesnel)
```

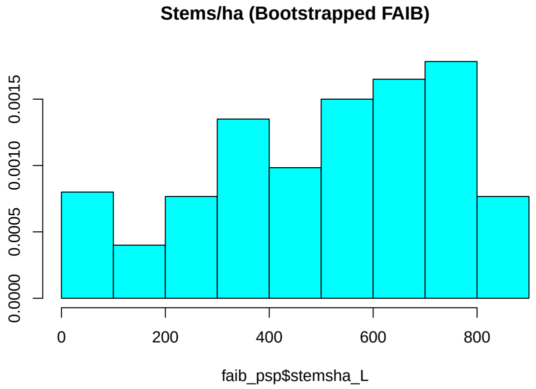


Tidy FAIB data: Bootstrapped sampling

```
faib_psp <- read.csv("/media/seamus/128GB_WORKD/EFI-TCC/0_Caret_Predict_to_writeRasterOutput/Data/FAIB_
faib_psp = subset(faib_psp, util == '12.5')
faib_psp$spc_live1 = as.factor(faib_psp$spc_live1)
faib_psp =  subset(faib_psp,
    spc_live1=='PL' | spc_live1=='PLI' | spc_live1=='FD'| spc_live1=='FDI' |
    spc_live1=='SB' | spc_live1=='SE' | spc_live1=='SW' | spc_live1=='SX' |
    spc_live1=='CW' | spc_live1=='HW' | spc_live1=='BL' | spc_live1=='LW')
faib_psp$species_class = dplyr::recode(faib_psp$spc_live1,
  PL = 1, PLI = 1, SB = 2, SE = 2, SX = 2,
  FD = 3, FDI = 3, CW = 3, HW = 4, BL = 5, LW = 6)
faib_psp$asp_cos = cos((faib_psp$aspect * pi) / 180)
faib_psp$asp_sin = sin((faib_psp$aspect * pi) / 180)
faib_psp = faib_psp[c("elev", "slope", "asp_cos", "asp_sin", "lead_htop", "species_class", "stemsha_L",
faib_psp$elev[faib_psp$elev <= 0] = NA
faib_psp$slope[faib_psp$slope <= 0] = NA
faib_psp$lead_htop[faib_psp$lead_htop < 2] = NA
faib_psp$stemsha_L[faib_psp$stemsha_L <= 0] = NA
faib_psp$wsvha_L[faib_psp$wsvha_L <= 1] = NA
faib_psp = subset(faib_psp, stemsha_L < 864)
faib_psp = na.omit(faib_psp)
psych::describe(faib_psp)
```

```
## # A tibble: 8 x 13
##    vars     n     mean      sd    median  trimmed     mad    min    max  range
##   <int> <dbl>    <dbl>   <dbl>     <dbl>    <dbl>   <dbl>  <dbl>  <dbl>  <dbl>
## 1     1   605 999.     163.     9.74e+ 2 975.     133.    661    1750   1089
## 2     2   605   9.48     8.64   7   e+ 0   7.99     5.93    1      64     63
## 3     3   605  -0.0175   0.690 -1.84e-16  -0.0215   1.05   -1       1      2
## 4     4   605   0.118    0.715  3.09e- 1   0.147    0.896  -1       1      2
## 5     5   605  19.6      6.56   1.91e+ 1  19.3      6.43     5.15   39.6   34.4
## 6     6   605   1.92     1.05   1   e+ 0   1.83     0       1       5      4
## 7     7   605 520.     220.     5.40e+ 2 532.     260.     20     864.    844.
## 8     8   605 179.     156.     1.29e+ 2 156.     119.      1.68  800.    798.
## # ... with 3 more variables: skew <dbl>, kurtosis <dbl>, se <dbl>
```

```
stemsha_L_raster_df = as.data.frame(rasterToPoints(stemsha_L_raster_gaspard))
dist.fun = approxfun(density(stemsha_L_raster_df$stemsha_L))
faib_psp = dplyr::sample_n(faib_psp, 500, weight_by = dist.fun(faib_psp$stemsha_L), replace = TRUE)
faib_psp = dplyr::sample_n(faib_psp, 600, weight_by = dist.fun, replace = TRUE)
truehist(faib_psp$stemsha_L, main="Stems/ha (Bootstrapped FAIB)")
```



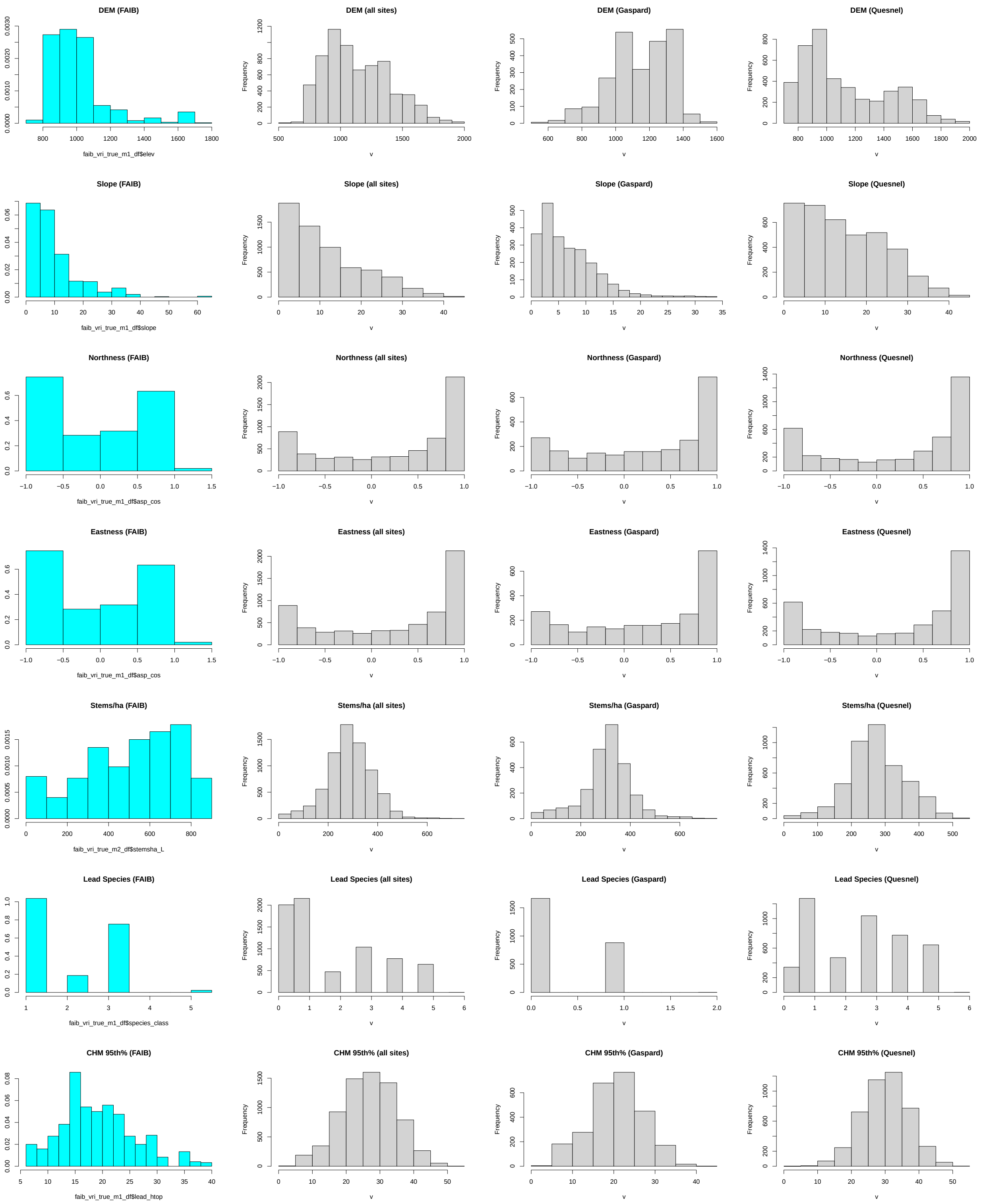
Tidy plot data: Data cleaning & training-test split

```
faib_psp$elev = as.numeric(faib_psp$elev)
faib_psp$slope = as.numeric(faib_psp$slope)
faib_psp$asp_cos = as.numeric(faib_psp$asp_cos)
faib_psp$asp_sin = as.numeric(faib_psp$asp_sin)
faib_psp$lead_htop = as.numeric(faib_psp$lead_htop)
faib_psp$species_class = as.numeric(faib_psp$species_class)
faib_psp$stemsha_L = as.numeric(faib_psp$stemsha_L)
faib_psp$wsvha_L = as.numeric(faib_psp$wsvha_L)
faib_vri_true_m1_df = faib_psp[c("elev", "slope", "asp_cos", "asp_sin", "lead_htop", "species_class", "
faib_vri_true_m2_df = faib_psp[c("elev", "slope", "asp_cos", "asp_sin", "lead_htop", "species_class", "
faib_vri_true_m1_df = na.omit(faib_vri_true_m1_df)
faib_vri_true_m2_df = na.omit(faib_vri_true_m2_df)

n <- nrow(faib_vri_true_m2_df)
frac <- 0.8
ix <- sample(n, frac * n)
```

```
train_m1 = faib_vri_true_m1_df[ix,]
test_m1 = faib_vri_true_m1_df[-ix,]
train_m2 = faib_vri_true_m2_df[ix,]
test_m2 = faib_vri_true_m2_df[-ix,]

X_train_m1=train_m1[,-7]
X_test_m1=test_m1[,-7]
y_train_m1=train_m1[,7]
y_test_m1=test_m1[,7]

X_train_m2=train_m2[,-8]
X_test_m2=test_m2[,-8]
y_train_m2=train_m2[,8]
y_test_m2=test_m2[,8]

X_m1 = faib_vri_true_m1_df[,-7]
y_m1 = faib_vri_true_m1_df[,7]
X_m2 = faib_vri_true_m2_df[,-8]
y_m2 = faib_vri_true_m2_df[,8]
```

Exploratory data analysis: Comparing distributions

```
library(MASS)
truehist(faib_vri_true_m1_df$elev, main="DEM (FAIB)")
hist(elev_raster, main="DEM (all sites)", maxpixels=22000000)
hist(elev_raster_gaspard, main="DEM (Gaspard)", maxpixels=22000000)
hist(elev_raster_quesnel, main="DEM (Quesnel)", maxpixels=22000000)
truehist(faib_vri_true_m1_df$slope, main="Slope (FAIB)")
hist(slope_raster, main="Slope (all sites)", maxpixels=22000000)
hist(slope_raster_gaspard, main="Slope (Gaspard)", maxpixels=22000000)
hist(slope_raster_quesnel, main="Slope (Quesnel)", maxpixels=22000000)
truehist(faib_vri_true_m1_df$asp_cos, main="Northness (FAIB)")
hist(asp_cos_raster, main="Northness (all sites)", maxpixels=22000000)
hist(asp_cos_raster_gaspard, main="Northness (Gaspard)", maxpixels=22000000)
hist(asp_cos_rast_quesnel, main="Northness (Quesnel)", maxpixels=22000000)
truehist(faib_vri_true_m1_df$asp_cos, main="Eastness (FAIB)")
hist(asp_cos_raster, main="Eastness (all sites)", maxpixels=22000000)
hist(asp_cos_raster_gaspard, main="Eastness (Gaspard)", maxpixels=22000000)
hist(asp_cos_rast_quesnel, main="Eastness (Quesnel)", maxpixels=22000000)
truehist(faib_vri_true_m2_df$stemsha_L, main="Stems/ha (FAIB)")
hist(stemsha_L_raster, main="Stems/ha (all sites)", maxpixels=22000000)
hist(stemsha_L_raster_gaspard, main="Stems/ha (Gaspard)", maxpixels=22000000)
hist(stemsha_L_raster_quesnel, main="Stems/ha (Quesnel)", maxpixels=22000000)
faib_vri_true_m1_df$species_class = as.numeric(faib_vri_true_m1_df$species_class)
truehist(faib_vri_true_m1_df$species_class, main="Lead Species (FAIB)")
hist(species_class_raster, main="Lead Species (all sites)", maxpixels=22000000)
hist(species_class_raster_gaspard, main="Lead Species (Gaspard)", maxpixels=22000000)
hist(species_class_raster_quesnel, main="Lead Species (Quesnel)", maxpixels=22000000)
truehist(faib_vri_true_m1_df$lead_htop, main="CHM 95th% (FAIB)")
hist(lead_htop_raster, main="CHM 95th% (all sites)", maxpixels=22000000)
hist(lead_htop_raster_gaspard, main="CHM 95th% (Gaspard)", maxpixels=22000000)
hist(lead_htop_raster_quesnel, main="CHM 95th% (Quesnel)", maxpixels=22000000)
```



## Exploratory data analysis: Visualize trends in variance

```
elev_wsvha_lm = lm(wsvha_L ~ elev, data = faib_vri_true_m1_df)
slope_wsvha_lm = lm(wsvha_L ~ slope, data = faib_vri_true_m1_df)
asp_cos_wsvha_lm = lm(wsvha_L ~ asp_cos, data = faib_vri_true_m1_df)
asp_sin_wsvha_lm = lm(wsvha_L ~ asp_sin, data = faib_vri_true_m1_df)
lead_htop_wsvha_lm = lm(wsvha_L ~ lead_htop, data = faib_vri_true_m1_df)
species_class_wsvha_lm = lm(wsvha_L ~ species_class, data = faib_vri_true_m1_df)
stemsha_L_wsvha_lm = lm(wsvha_L ~ stemsha_L, data = faib_vri_true_m2_df)

summary(elev_wsvha_lm)
```

```
##
## Call:
## lm(formula = wsvha_L ~ elev, data = faib_vri_true_m1_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -194.10 -110.45  -47.43   60.55  558.71
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 282.55941   37.30740   7.574 1.38e-13 ***
## elev         -0.11105    0.03641  -3.050  0.00239 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 156.4 on 598 degrees of freedom
## Multiple R-squared:  0.01532,    Adjusted R-squared:  0.01367
## F-statistic: 9.302 on 1 and 598 DF,  p-value: 0.00239
```

```
truehist(faib_vri_true_m1_df$elev)
truehist(elev_wsvha_lm$residuals)
plot(wsvha_L ~ elev, data = faib_vri_true_m1_df,
     main="Linear function showing negative correlation:\nR^2=0.011,  =-0.0954, p<0.0000",
     col="blue", pch=20, cex=0.5, cex.main=0.6, cex.lab=0.8, cex.axis=0.8, adj=1,
     ylab = "wsvha_L (m3/ha)", xlab = "DEM")
abline(elev_wsvha_lm, col = "red")
plot(elev_wsvha_lm, which=1,
     main="Residuals showing increasing trend\n clustering at larger fitted values",
     col="blue", pch=20, cex=0.5, cex.main=0.6, cex.lab=0.5, cex.axis=0.5, adj=1) # Residuals vs Fitted
summary(slope_wsvha_lm)
```

```
##
## Call:
## lm(formula = wsvha_L ~ slope, data = faib_vri_true_m1_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -181.90 -111.46  -41.51   58.01  572.83
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 192.5757     9.5012  20.268  < 2e-16 ***
## slope        -2.4198     0.7699  -3.143  0.00176 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 156.3 on 598 degrees of freedom
## Multiple R-squared:  0.01625,    Adjusted R-squared:  0.0146
## F-statistic: 9.878 on 1 and 598 DF,  p-value: 0.001756
```

```r
truehist(faib_vri_true_m1_df$slope)
truehist(slope_wsvha_lm$residuals)
plot(wsvha_L ~ slope, data = faib_vri_true_m1_df,
     main="Linear function showing negative correlation:\nR^2=0.0013,  =-0.5171: p=0.0009",
     col="blue", pch=20, cex=0.5, cex.main=0.6, cex.lab=0.8, cex.axis=0.8, adj=1,
     ylab = "wsvha_L (m3/ha)", xlab = "slope")
abline(slope_wsvha_lm, col = "red")
plot(slope_wsvha_lm, which=1,
     main="Residuals showing increasing trend of\nnegative errors near larger fitted values",
     col="blue", pch=20, cex=0.5, cex.main=0.6, cex.lab=0.5, cex.axis=0.5, adj=1)
summary(asp_cos_wsvha_lm)
```

```
##
## Call:
## lm(formula = wsvha_L ~ asp_cos, data = faib_vri_true_m1_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -168.76 -111.46  -53.68   65.80  559.90
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  170.331      6.448  26.416   <2e-16 ***
## asp_cos       -2.646      9.044  -0.293     0.77
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 157.6 on 598 degrees of freedom
## Multiple R-squared:  0.0001431,  Adjusted R-squared:  -0.001529
## F-statistic: 0.08556 on 1 and 598 DF,  p-value: 0.77
```

```r
truehist(faib_vri_true_m1_df$asp_cos)
truehist(asp_cos_wsvha_lm$residuals)
plot(wsvha_L ~ asp_cos, data = faib_vri_true_m1_df,
     main="Linear function showing positive correlation:\nR^2=0.005,  =15.197, p<0.000",
     col="blue", pch=20, cex=0.5, cex.main=0.6, cex.lab=0.8, cex.axis=0.8, adj=1,
     ylab = "wsvha_L (m3/ha)", xlab = "asp_cos")
abline(asp_cos_wsvha_lm, col = "red")
plot(asp_cos_wsvha_lm, which=1,
     main="Residuals showing almost constant variance",
     col="blue", pch=20, cex=0.5, cex.main=0.6, cex.lab=0.5, cex.axis=0.5, adj=1)
summary(asp_sin_wsvha_lm)
```

```
##
## Call:
```

```
## lm(formula = wsvha_L ~ asp_sin, data = faib_vri_true_m1_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -169.29 -112.11  -51.15   67.63  557.88
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 170.3566     6.5032  26.196   <2e-16 ***
## asp_sin       0.9636     9.2744   0.104    0.917
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 157.6 on 598 degrees of freedom
## Multiple R-squared:  1.805e-05,  Adjusted R-squared:  -0.001654
## F-statistic: 0.0108 on 1 and 598 DF,  p-value: 0.9173
```

```
truehist(faib_vri_true_m1_df$asp_sin)
truehist(asp_sin_wsvha_lm$residuals)
plot(wsvha_L ~ asp_sin, data = faib_vri_true_m1_df,
     main="Linear function showing positive correlation:\nR^2=0.005,  =15.197, p<0.000",
     col="blue", pch=20, cex=0.5, cex.main=0.6, cex.lab=0.8, cex.axis=0.8, adj=1,
     ylab = "wsvha_L (m3/ha)", xlab = "asp_sin")
abline(asp_sin_wsvha_lm, col = "red")
plot(asp_sin_wsvha_lm, which=1,
     main="Residuals showing almost constant variance",
     col="blue", pch=20, cex=0.5, cex.main=0.6, cex.lab=0.5, cex.axis=0.5, adj=1)
summary(lead_htop_wsvha_lm)
```

```
##
## Call:
## lm(formula = wsvha_L ~ lead_htop, data = faib_vri_true_m1_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -171.321  -37.857   -9.749   43.918  234.478
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -227.7126     8.5234  -26.72   <2e-16 ***
## lead_htop     20.8526     0.4207   49.56   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 69.74 on 598 degrees of freedom
## Multiple R-squared:  0.8042, Adjusted R-squared:  0.8039
## F-statistic:  2456 on 1 and 598 DF,  p-value: < 2.2e-16
```

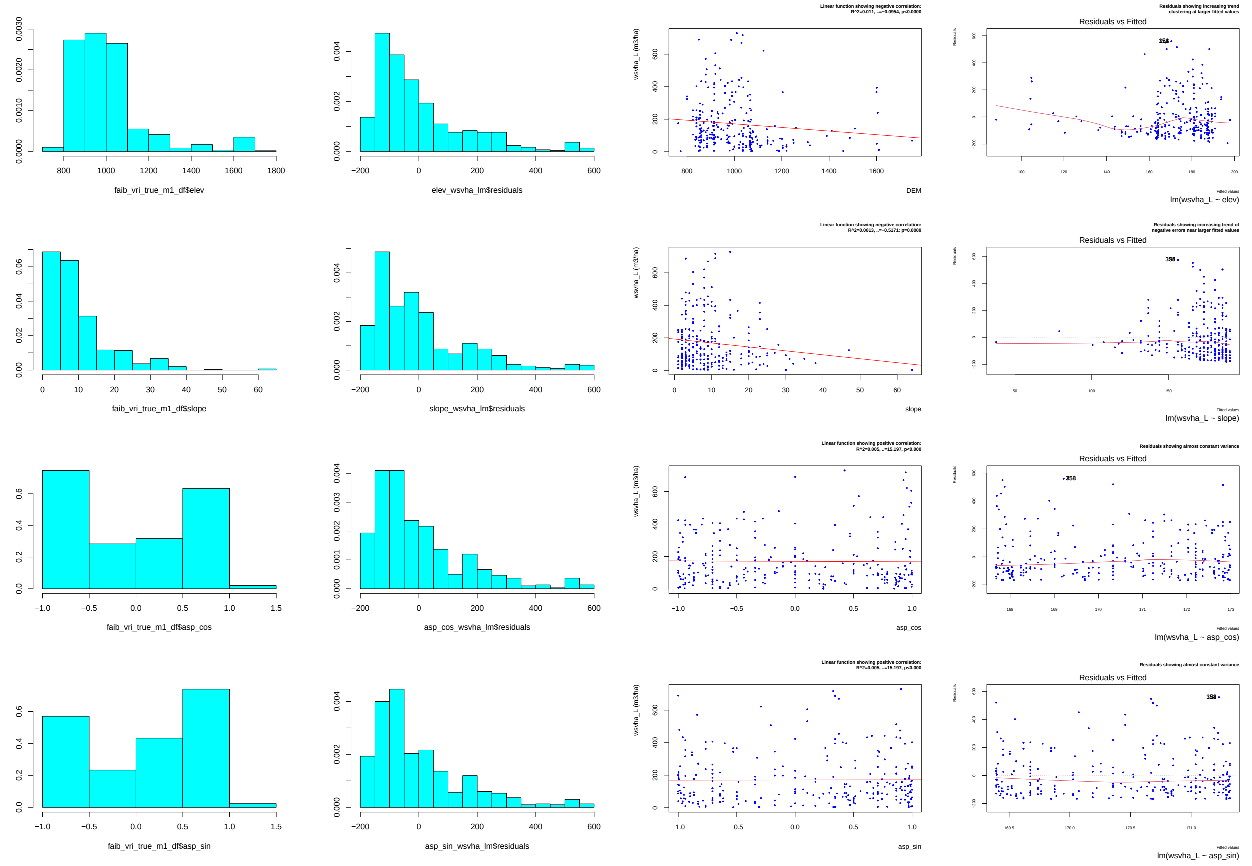
```
truehist(faib_vri_true_m1_df$lead_htop)
truehist(lead_htop_wsvha_lm$residuals)
plot(wsvha_L ~ lead_htop, data = faib_vri_true_m1_df,
     main="Linear function shows positive correlation:\n R^2=0.6508,  =20.6829, p<0.0000",
     col="blue", pch=20, cex=0.5, cex.main=0.6, cex.lab=0.8, cex.axis=0.8, adj=1,
     ylab = "wsvha_L (m3/ha)", xlab = "lead_htop")
abline(lead_htop_wsvha_lm, col = "red")
plot(lead_htop_wsvha_lm, which=1,
     main="Residuals showing non-constant variance with\n increasing trends at smallest and largest fit
     col="blue", pch=20, cex=0.5, cex.main=0.6, cex.lab=0.5, cex.axis=0.5, adj=1)
summary(stemsha_L_wsvha_lm)
```

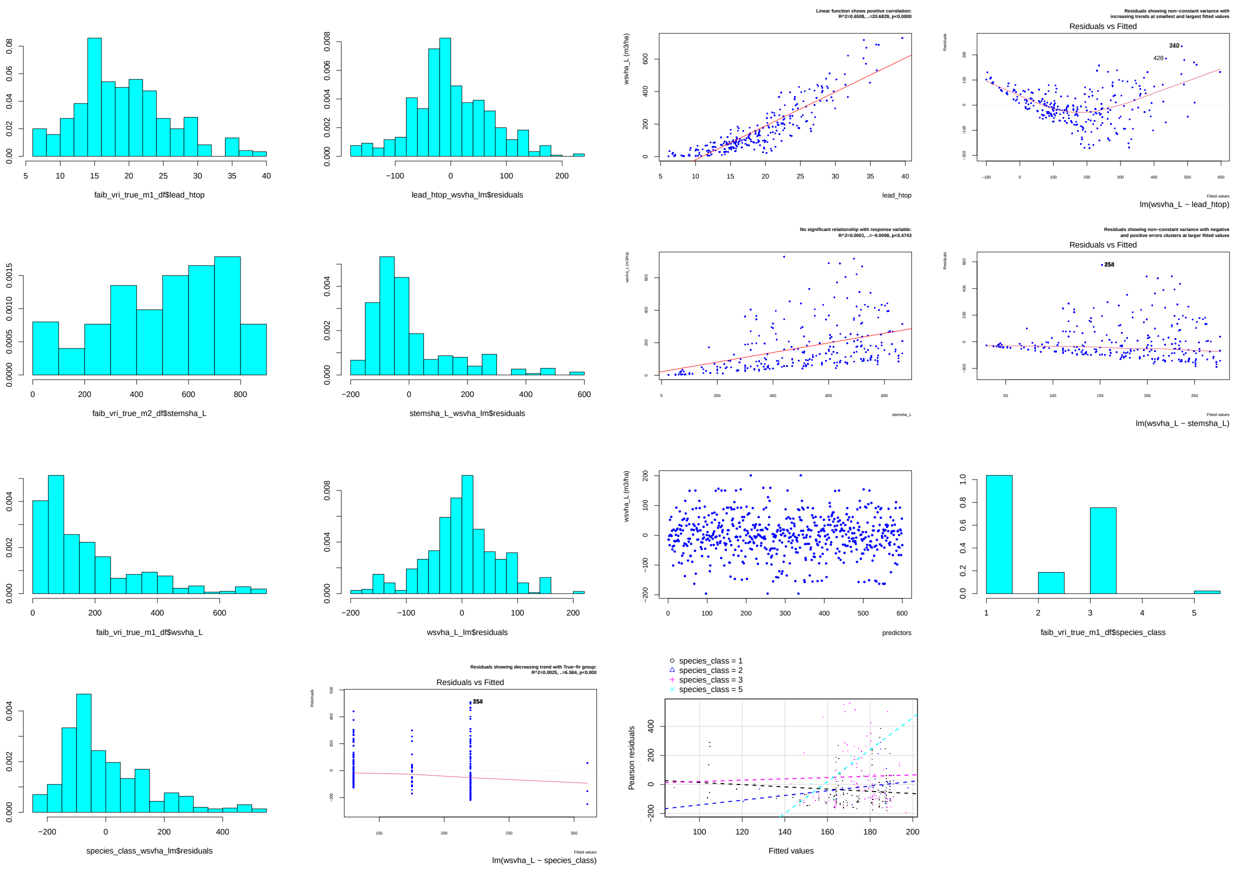
```
##
## Call:
## lm(formula = wsvha_L ~ stemsha_L, data = faib_vri_true_m2_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -187.97  -85.38  -42.28   31.51  577.06
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 22.17525   14.09829   1.573    0.116
## stemsha_L    0.29512    0.02556  11.547   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 142.5 on 598 degrees of freedom
## Multiple R-squared:  0.1823, Adjusted R-squared:  0.1809
## F-statistic: 133.3 on 1 and 598 DF,  p-value: < 2.2e-16
```

```
truehist(faib_vri_true_m2_df$stemsha_L)
truehist(stemsha_L_wsvha_lm$residuals)
plot(wsvha_L ~ stemsha_L, data = faib_vri_true_m2_df,
     main="No significant relationship with response variable:\nR^2=0.0001,  =-0.0008, p<0.4743",
     col="blue", pch=20, cex=0.5, cex.main=0.6, cex.lab=0.5, cex.axis=0.5, adj=1,
     ylab = "wsvha_L (m3/ha)", xlab = "stemsha_L")
abline(stemsha_L_wsvha_lm, col = "red")
plot(stemsha_L_wsvha_lm, which=1,
     main="Residuals showing non-constant variance with negative\nand positive errors clusters at large
     col="blue", pch=20, cex=0.5, cex.main=0.6, cex.lab=0.5, cex.axis=0.5, adj=1)
wsvha_L_lm = lm(wsvha_L ~ ., data = faib_vri_true_m1_df)
truehist(faib_vri_true_m1_df$wsvha_L)
truehist(wsvha_L_lm$residuals)
plot(wsvha_L_lm$residuals,
     main="",
     col="blue", pch=20, cex=0.8, cex.main=0.8, cex.lab=0.8, cex.axis=0.8, adj=1,
     ylab = "wsvha_L (m3/ha)", xlab = "predictors")
abline(wsvha_L_lm, col="red")
summary(species_class_wsvha_lm)
```

```
##
## Call:
## lm(formula = wsvha_L ~ species_class, data = faib_vri_true_m1_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -249.03 -101.57  -45.62   79.13  508.88
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     85.295     13.269   6.428 2.64e-10 ***
## species_class   44.979      6.205   7.249 1.30e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 151.1 on 598 degrees of freedom
## Multiple R-squared:  0.08078,    Adjusted R-squared:  0.07924
## F-statistic: 52.55 on 1 and 598 DF,  p-value: 1.302e-12
```

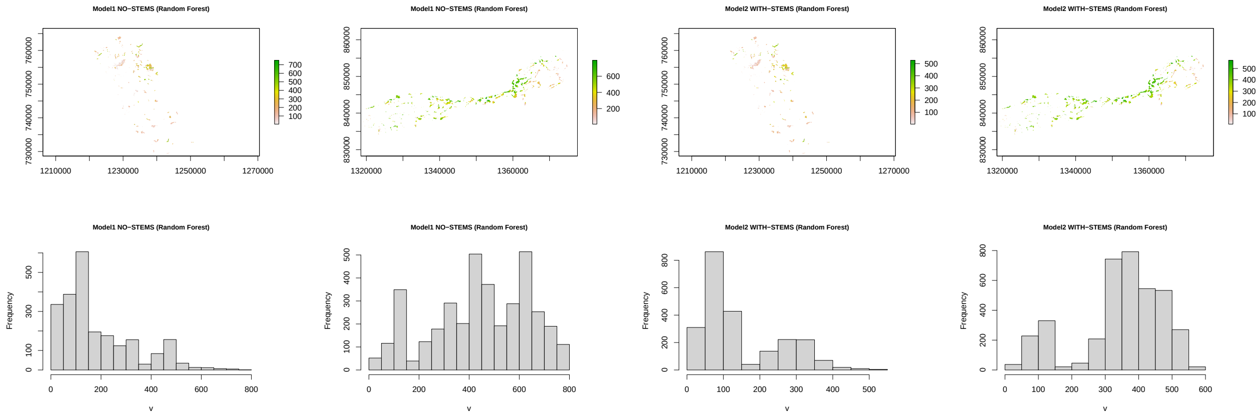
```
faib_vri_true_m1_df$species_class = as.numeric(faib_vri_true_m1_df$species_class)
truehist(faib_vri_true_m1_df$species_class)
truehist(species_class_wsvha_lm$residuals)
plot(species_class_wsvha_lm, which=1,
     main="Residuals showing decreasing trend with True-fir group:\nR^2=0.0025,  =6.584, p<0.000",
     col="blue", pch=20, cex=0.5, cex.main=0.6, cex.lab=0.5, cex.axis=0.5, adj=1)
car::residualPlots(elev_wsvha_lm, terms= ~ 1 | species_class, cex=0.1, pch=19) # plot vs. yhat grouping
```





Model: Random Forest Tree Regression

```
tuneResult_rf_m1_full <- tune.randomForest(
  X_m1, y_m1,
  mtry = c(2:10), ntree = 50,
  tunecontrol = tune.control(sampling = "cross", cross = 10),
  preProcess = c("BoxCox","center","scale"))

tuneResult_rf_m2_full <- tune.randomForest(
  X_m2, y_m2,
  mtry = c(2:10), ntree = 50,
  tunecontrol = tune.control(sampling = "cross", cross = 10),
  preProcess = c("BoxCox","center","scale"))

tunedModel_rf_m1_full <- tuneResult_rf_m1_full$best.model
tunedModel_rf_m2_full <- tuneResult_rf_m2_full$best.model
save(tunedModel_rf_m1_full, file = "/media/seamus/128GB_WORKD/data/models/tcc-wsvha/wsvha_model1_random
save(tunedModel_rf_m2_full, file = "/media/seamus/128GB_WORKD/data/models/tcc-wsvha/wsvha_model2_random
tunedModel_rf_m1 = predict(tunedModel_rf_m1_full, X_m1, y_m1, type = "response")
tunedModel_rf_m2 = predict(tunedModel_rf_m2_full, X_m2, y_m2, type = "response")

tuneResult_rf_m1_train <- tune.randomForest(
  X_train_m1, y_train_m1,
  mtry = c(2:10), ntree = 50,
  tunecontrol = tune.control(sampling = "cross", cross = 10),
  preProcess = c("BoxCox","center","scale"))

tuneResult_rf_m2_train <- tune.randomForest(
  X_train_m2, y_train_m2,
  mtry = c(2:10), ntree = 50,
  tunecontrol = tune.control(sampling = "cross", cross = 10),
  preProcess = c("BoxCox","center","scale"))

tunedModel_rf_m1_train <- tuneResult_rf_m1_train$best.model
tunedModel_rf_m2_train <- tuneResult_rf_m2_train$best.model
tunedModel_rf_m1_test = predict(tunedModel_rf_m1_train, X_test_m1, y_test_m1, type="response")
tunedModel_rf_m2_test = predict(tunedModel_rf_m2_train, X_test_m2, y_test_m2, type="response")
tunedModel_rf_m1_full_MAE = MAE(tunedModel_rf_m1, y_m2)
tunedModel_rf_m1_full_RMSE = RMSE(tunedModel_rf_m1, y_m2)
tunedModel_rf_m2_full_MAE = MAE(tunedModel_rf_m2, y_m2)
tunedModel_rf_m2_full_RMSE = RMSE(tunedModel_rf_m2, y_m2)
tunedModel_rf_m1_test_MAE = MAE(tunedModel_rf_m1_test, y_test_m2)
tunedModel_rf_m1_test_RMSE = RMSE(tunedModel_rf_m1_test, y_test_m2)
tunedModel_rf_m2_test_MAE = MAE(tunedModel_rf_m2_test, y_test_m2)
tunedModel_rf_m2_test_RMSE = RMSE(tunedModel_rf_m2_test, y_test_m2)

R2(tunedModel_rf_m1, y_m1)
MAE(tunedModel_rf_m1, y_m1)
RMSE(tunedModel_rf_m1, y_m1)
MAE(tunedModel_rf_m1_test, y_test_m1)
RMSE(tunedModel_rf_m1_test, y_test_m1)
tunedModel_rf_m1_full_RMSE/tunedModel_rf_m1_test_RMSE

R2(tunedModel_rf_m2, y_m2)
MAE(tunedModel_rf_m2, y_m2)
RMSE(tunedModel_rf_m2, y_m2)
MAE(tunedModel_rf_m2_test, y_test_m2)
RMSE(tunedModel_rf_m2_test, y_test_m2)
tunedModel_rf_m2_full_RMSE/tunedModel_rf_m2_test_RMSE

tunedModel_rf_m1_to_raster <- predict(covs_m1, tunedModel_rf_m1_full)
tunedModel_rf_m2_to_raster <- predict(covs_m2, tunedModel_rf_m2_full)
tunedModel_rf_m1_to_raster_gaspard <- predict(covs_m1_gaspard, tunedModel_rf_m1_full)
tunedModel_rf_m2_to_raster_gaspard <- predict(covs_m2_gaspard, tunedModel_rf_m2_full)
tunedModel_rf_m1_to_raster_quesnel <- predict(covs_m1_quesnel, tunedModel_rf_m1_full)
tunedModel_rf_m2_to_raster_quesnel <- predict(covs_m2_quesnel, tunedModel_rf_m2_full)
writeRaster(tunedModel_rf_m2_to_raster, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/wsvha/wsvh
writeRaster(tunedModel_rf_m1_to_raster, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/wsvha/wsvh
writeRaster(tunedModel_rf_m2_to_raster_gaspard, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/ws
writeRaster(tunedModel_rf_m1_to_raster_gaspard, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/ws
writeRaster(tunedModel_rf_m2_to_raster_quesnel, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/ws
writeRaster(tunedModel_rf_m1_to_raster_quesnel, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/ws
```

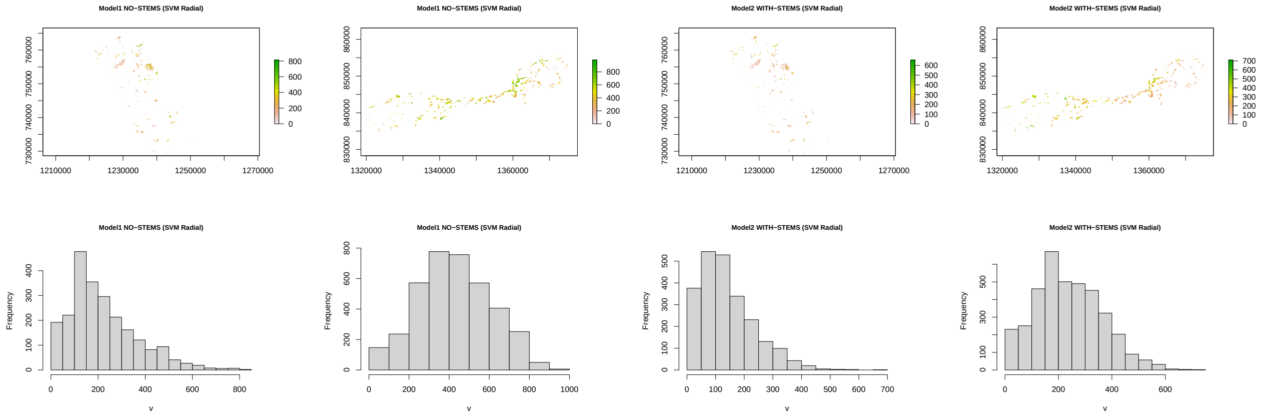


Model: Support Vector Machine Radial Kernel (epsilon-untuned)

```
fitControl_YeoJx1 = caret::trainControl(method="repeatedcv", number=10, repeats=1)
fitControl_YeoJx3 = caret::trainControl(method="repeatedcv", number=10, repeats=3)
fitControl_YeoJx5 = caret::trainControl(method="repeatedcv", number=10, repeats=5)
fitControl_YeoJx10 = caret::trainControl(method="repeatedcv", number=10, repeats=10)

tuneResult_svm_m1_full <- train(wsvha_L~., data=faib_vri_true_m1_df,
  trControl = fitControl_YeoJx10,
  method = 'svmRadial',
  metric = 'RMSE',
  ranges = list(
    cost = c(1,5,7,15,20),
    gamma = 2^(-1:1)),
  tuneLength = 10,
  preProc = c('YeoJohnson', 'scale', 'center', 'corr'),
  verbose=F)

tuneResult_svm_m2_full <- train(wsvha_L~., data=faib_vri_true_m2_df,
  trControl = fitControl_YeoJx10,
  method = 'svmRadial',
  metric = 'RMSE',
  ranges = list(
    cost = c(1,5,7,15,20),
    gamma = 2^(-1:1)),
  tuneLength = 10,
  preProc = c('YeoJohnson', 'scale', 'center', 'corr'),
  verbose=F)

tunedModel_svm_m1_full <- tuneResult_svm_m1_full$finalModel
tunedModel_svm_m2_full <- tuneResult_svm_m2_full$finalModel
save(tunedModel_svm_m1_full, file = "/media/seamus/128GB_WORKD/data/models/tcc-wsvha/wsvha_model1_svmRa
save(tunedModel_svm_m2_full, file = "/media/seamus/128GB_WORKD/data/models/tcc-wsvha/wsvha_model2_svmRa

tuneResult_svm_m1_train <- train(
  X_train_m1, y_train_m1,
  trControl = fitControl_YeoJx10,
```

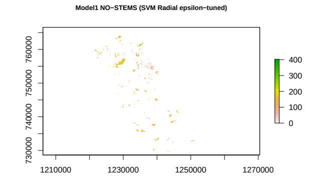
```r
  method = 'svmRadial',
  metric = 'RMSE',
  ranges = list(
    cost = c(1,5,7,15,20),
    gamma = 2^(-1:1)),
  tuneLength = 10,
  preProc = c('YeoJohnson', 'scale', 'center', 'corr'),
  verbose=F)

tuneResult_svm_m2_train <- train(
  X_train_m2, y_train_m2,
  trControl = fitControl_YeoJx10,
  method = 'svmRadial',
  metric = 'RMSE',
  ranges = list(
    cost = c(1,5,7,15,20),
    gamma = 2^(-1:1)),
  tuneLength = 10,
  preProc = c('YeoJohnson', 'scale', 'center', 'corr'),
  verbose=F)

tunedModel_svmRadial_m1_test = predict(tuneResult_svm_m1_train, data = test_m1)
tunedModel_svmRadial_m2_test = predict(tuneResult_svm_m2_train, data = test_m2)
tunedModel_svmRadial_m1_test_MAE = MAE(tunedModel_svmRadial_m1_test, test_m1$wsvha_L)
tunedModel_svmRadial_m2_test_MAE = MAE(tunedModel_svmRadial_m2_test, test_m2$wsvha_L)
tunedModel_svmRadial_m1_test_RMSE = RMSE(tunedModel_svmRadial_m1_test, test_m1$wsvha_L)
tunedModel_svmRadial_m2_test_RMSE = RMSE(tunedModel_svmRadial_m2_test, test_m2$wsvha_L)

tunedModel_svmRadial_m1 = predict(tuneResult_svm_m1_full, data = faib_vri_true_m1_df)
tunedModel_svmRadial_m2 = predict(tuneResult_svm_m2_full, data = faib_vri_true_m2_df)
tunedModel_svmRadial_m1_MAE = MAE(tunedModel_svmRadial_m1, faib_vri_true_m1_df$wsvha_L)
tunedModel_svmRadial_m2_MAE = MAE(tunedModel_svmRadial_m2, faib_vri_true_m2_df$wsvha_L)
tunedModel_svmRadial_m1_RMSE = RMSE(tunedModel_svmRadial_m1, faib_vri_true_m1_df$wsvha_L)
tunedModel_svmRadial_m2_RMSE = RMSE(tunedModel_svmRadial_m2, faib_vri_true_m2_df$wsvha_L)

tunedModel_rf_m1_train <- tuneResult_rf_m1_train$best.model
tunedModel_rf_m2_train <- tuneResult_rf_m2_train$best.model
tunedModel_rf_m1_test = predict(tunedModel_rf_m1_train, X_test_m1, y_test_m1, type="response")
tunedModel_rf_m2_test = predict(tunedModel_rf_m2_train, X_test_m2, y_test_m2, type="response")
tunedModel_rf_m1_full_MAE = MAE(tunedModel_rf_m1, y_m2)
tunedModel_rf_m1_full_RMSE = RMSE(tunedModel_rf_m1, y_m2)
tunedModel_rf_m2_full_MAE = MAE(tunedModel_rf_m2, y_m2)
tunedModel_rf_m2_full_RMSE = RMSE(tunedModel_rf_m2, y_m2)
tunedModel_rf_m1_test_MAE = MAE(tunedModel_rf_m1_test, y_test_m2)
tunedModel_rf_m1_test_RMSE = RMSE(tunedModel_rf_m1_test, y_test_m2)
tunedModel_rf_m2_test_MAE = MAE(tunedModel_rf_m2_test, y_test_m2)
tunedModel_rf_m2_test_RMSE = RMSE(tunedModel_rf_m2_test, y_test_m2)

tunedModel_svmRadial_m2_MAE
tunedModel_svmRadial_m2_RMSE
tunedModel_svmRadial_m2_test_MAE
tunedModel_svmRadial_m2_test_RMSE
tunedModel_svmRadial_m2_RMSE/tunedModel_svmRadial_m2_test_RMSE
```

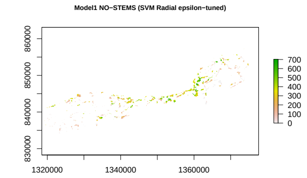
```
tunedModel_svmRadial_m1_MAE
tunedModel_svmRadial_m1_RMSE
tunedModel_svmRadial_m1_test_MAE
tunedModel_svmRadial_m1_test_RMSE
tunedModel_svmRadial_m1_RMSE/tunedModel_svmRadial_m1_test_RMSE


tunedModel_svm_m1_to_raster <- raster::predict(covs_m1, tuneResult_svm_m1_full)
tunedModel_svm_m2_to_raster <- raster::predict(covs_m2, tuneResult_svm_m2_full)
tunedModel_svm_m1_to_raster_gaspard <- raster::predict(covs_m1_gaspard, tuneResult_svm_m1_full)
tunedModel_svm_m2_to_raster_gaspard <- raster::predict(covs_m2_gaspard, tuneResult_svm_m2_full)
tunedModel_svm_m1_to_raster_quesnel <- raster::predict(covs_m1_quesnel, tuneResult_svm_m1_full)
tunedModel_svm_m2_to_raster_quesnel <- raster::predict(covs_m2_quesnel, tuneResult_svm_m2_full)
save(tunedModel_svm_m1_full, file = "/media/seamus/128GB_WORKD/data/models/tcc-wsvha/wsvha_model1_svmRa
save(tunedModel_svm_m2_full, file = "/media/seamus/128GB_WORKD/data/models/tcc-wsvha/wsvha_model2_svmRa
tunedModel_svm_m1_to_raster$layer[tunedModel_svm_m1_to_raster$layer <= 0] = 0
tunedModel_svm_m2_to_raster$layer[tunedModel_svm_m2_to_raster$layer <= 0] = 0
tunedModel_svm_m1_to_raster_gaspard$layer[tunedModel_svm_m1_to_raster_gaspard$layer <= 0] = 0
tunedModel_svm_m2_to_raster_gaspard$layer[tunedModel_svm_m2_to_raster_gaspard$layer <= 0] = 0
tunedModel_svm_m1_to_raster_quesnel$layer[tunedModel_svm_m1_to_raster_quesnel$layer <= 0] = 0
tunedModel_svm_m2_to_raster_quesnel$layer[tunedModel_svm_m2_to_raster_quesnel$layer <= 0] = 0
writeRaster(tunedModel_svm_m1_to_raster, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/wsvha/bo
writeRaster(tunedModel_svm_m2_to_raster, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/wsvha/bo
writeRaster(tunedModel_svm_m1_to_raster_gaspard, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/
writeRaster(tunedModel_svm_m2_to_raster_gaspard, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/
writeRaster(tunedModel_svm_m1_to_raster_quesnel, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/
writeRaster(tunedModel_svm_m2_to_raster_quesnel, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/
```

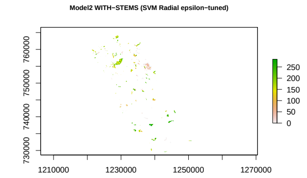


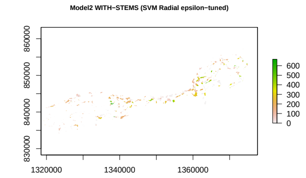
Model: Support Vector Machine Radial Kernel (epsilon-tuned)

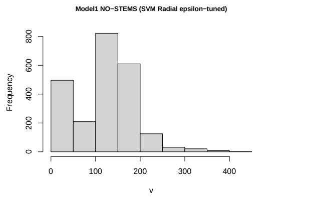
```
tuneResult_svm_m1_full_eps <- train(
  X_m1, y_m1,
  trControl = fitControl_YeoJx10,
  method = 'svmRadial', metric = 'RMSE',
  ranges = list(epsilon = seq(0.02,0.1,0.2), cost = c(1,5,7,15,20), gamma = 2^(-1:1)), tuneLength = 10,
  preProc = c('YeoJohnson', 'scale', 'center', 'corr'),
  verbose=F)

tuneResult_svm_m2_full_eps = train(
  X_m2, y_m2,
  trControl = fitControl_YeoJx10,
  method = 'svmRadial', metric = 'RMSE',
  ranges = list(epsilon = seq(0.02,0.1,0.2), cost = c(1,5,7,15,20), gamma = 2^(-1:1)),tuneLength = 10,
  preProc = c('YeoJohnson', 'scale', 'center', 'corr'),
  verbose=F)

tunedModel_svm_m2_full_eps <- tuneResult_svm_m2_full_eps$finalModel
tunedModel_svm_m1_full_eps <- tuneResult_svm_m1_full_eps$finalModel
save(tunedModel_svm_m1_full_eps, file = "/media/seamus/128GB_WORKD/data/models/tcc-wsvha/wsvha_model1_s
save(tunedModel_svm_m2_full_eps, file = "/media/seamus/128GB_WORKD/data/models/tcc-wsvha/wsvha_model2_s
tunedModel_svm_m1_to_raster_eps <- raster::predict(covs_m1, tuneResult_svm_m1_full_eps)
tunedModel_svm_m2_to_raster_eps <- raster::predict(covs_m2, tuneResult_svm_m2_full_eps)
tunedModel_svm_m1_to_raster_eps_gaspard <- raster::predict(covs_m1_gaspard, tuneResult_svm_m1_full_eps)
tunedModel_svm_m2_to_raster_eps_gaspard <- raster::predict(covs_m2_gaspard, tuneResult_svm_m2_full_eps)
tunedModel_svm_m1_to_raster_eps_quesnel <- raster::predict(covs_m1_quesnel, tuneResult_svm_m1_full_eps)
tunedModel_svm_m2_to_raster_eps_quesnel <- raster::predict(covs_m2_quesnel, tuneResult_svm_m2_full_eps)
tunedModel_svm_m1_to_raster_eps$layer[tunedModel_svm_m1_to_raster_eps$layer <= 0] = 0
tunedModel_svm_m2_to_raster_eps$layer[tunedModel_svm_m2_to_raster_eps$layer <= 0] = 0
tunedModel_svm_m1_to_raster_eps_gaspard$layer[tunedModel_svm_m1_to_raster_eps_gaspard$layer <= 0] = 0
tunedModel_svm_m2_to_raster_eps_gaspard$layer[tunedModel_svm_m2_to_raster_eps_gaspard$layer <= 0] = 0
tunedModel_svm_m1_to_raster_eps_quesnel$layer[tunedModel_svm_m1_to_raster_eps_quesnel$layer <= 0] = 0
tunedModel_svm_m2_to_raster_eps_quesnel$layer[tunedModel_svm_m2_to_raster_eps_quesnel$layer <= 0] = 0
writeRaster(tunedModel_svm_m1_to_raster_eps, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/wsvha
writeRaster(tunedModel_svm_m2_to_raster_eps, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/wsvha
writeRaster(tunedModel_svm_m1_to_raster_eps_gaspard, filename = "/media/seamus/128GB_WORKD/data/raster/t
writeRaster(tunedModel_svm_m2_to_raster_eps_gaspard, filename = "/media/seamus/128GB_WORKD/data/raster/t
writeRaster(tunedModel_svm_m1_to_raster_eps_quesnel, filename = "/media/seamus/128GB_WORKD/data/raster/t
writeRaster(tunedModel_svm_m2_to_raster_eps_quesnel, filename = "/media/seamus/128GB_WORKD/data/raster/t
```

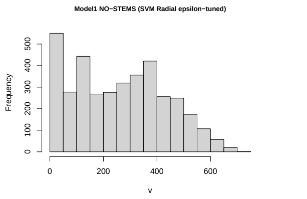


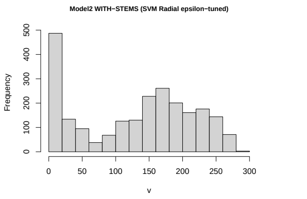


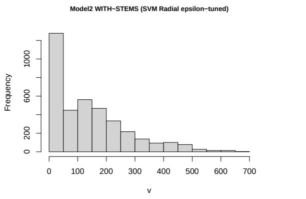










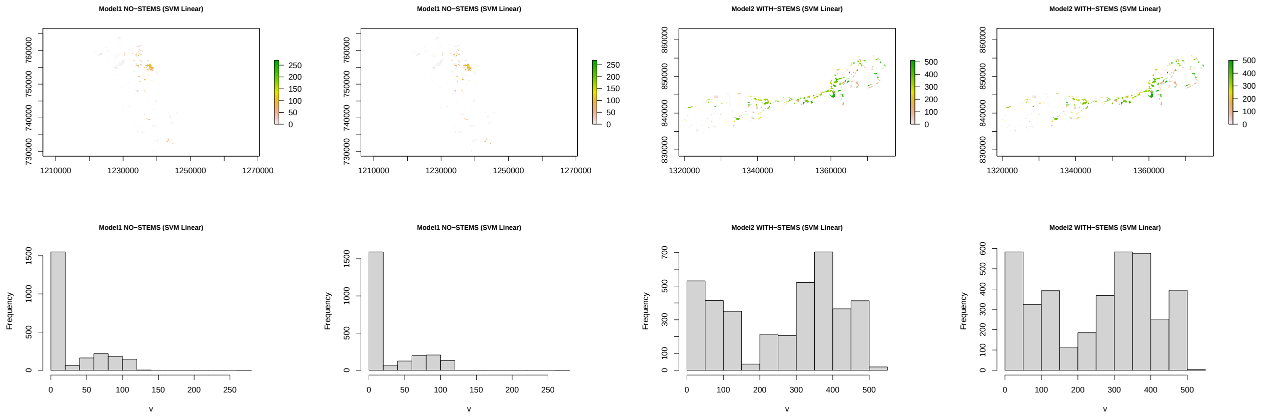


Model: Support Vector Machine Linear Kernel

```
tuneResult_svm_m1_full_linear <- train(
  X_m1, y_m1,
```

```r
  trControl = fitControl_YeoJx10,
  method = 'svmLinear', metric = 'RMSE',
  ranges = list(cost = c(1,5,7,15,20), gamma = 2^(-1:1)),tuneLength = 10,
  preProc = c('YeoJohnson', 'scale', 'center', 'corr'),
  verbose=F)

tuneResult_svm_m2_full_linear <- train(
  X_m2, y_m2,
  trControl = fitControl_YeoJx10,
  method = 'svmLinear',metric = 'RMSE',
  ranges = list(cost = c(1,5,7,15,20), gamma = 2^(-1:1)),tuneLength = 10,
  preProc = c('YeoJohnson', 'scale', 'center', 'corr'),
  verbose=F)

tunedModel_svm_m2_full_linear <- tuneResult_svm_m2_full_linear$finalModel
tunedModel_svm_m1_full_linear <- tuneResult_svm_m1_full_linear$finalModel
save(tunedModel_svm_m1_full_linear, file = "/media/seamus/128GB_WORKD/data/models/tcc-wsvha/wsvha_model1
save(tunedModel_svm_m2_full_linear, file = "/media/seamus/128GB_WORKD/data/models/tcc-wsvha/wsvha_model2

tuneResult_svmLinear_m2_10k_train <- train(
  X_train_m2, y_train_m2,
  trControl = fitControl_YeoJx10,
  method = 'svmLinear', metric = 'RMSE',
  ranges = list(cost = c(1,5,7,15,20), gamma = 2^(-1:1)),tuneLength = 10,
  preProc = c('YeoJohnson', 'scale', 'center', 'corr'),
  verbose=F)

tuneResult_svmLinear_m1_10k_train <- train(
  X_train_m1, y_train_m1,
  trControl = fitControl_YeoJx10,
  method = 'svmLinear',metric = 'RMSE',
  ranges = list(cost = c(1,5,7,15,20), gamma = 2^(-1:1)),tuneLength = 10,
  preProc = c('YeoJohnson', 'scale', 'center', 'corr'),
  verbose=F)

tunedModel_svmLinear_m2_test = predict(tuneResult_svmLinear_m2_10k_train, data = test_m2)
tunedModel_svmLinear_m1_test = predict(tuneResult_svmLinear_m1_10k_train, data = test_m1)
tunedModel_svmLinear_m2_test_MAE = MAE(tunedModel_svmLinear_m2_test, test_m2$wsvha_L)
tunedModel_svmLinear_m1_test_MAE = MAE(tunedModel_svmLinear_m1_test, test_m1$wsvha_L)
tunedModel_svmLinear_m2_test_RMSE = RMSE(tunedModel_svmLinear_m2_test, test_m2$wsvha_L)
tunedModel_svmLinear_m1_test_RMSE = RMSE(tunedModel_svmLinear_m1_test, test_m1$wsvha_L)

tunedModel_svmLinear_m2 = predict(tuneResult_svm_m2_full_linear, data = faib_vri_true_m2_df)
tunedModel_svmLinear_m1 = predict(tuneResult_svm_m1_full_linear, data = faib_vri_true_m1_df)
tunedModel_svmLinear_m2_MAE = MAE(tunedModel_svmLinear_m2, faib_vri_true_m2_df$wsvha_L)
tunedModel_svmLinear_m1_MAE = MAE(tunedModel_svmLinear_m1, faib_vri_true_m1_df$wsvha_L)
tunedModel_svmLinear_m2_RMSE = RMSE(tunedModel_svmLinear_m2, faib_vri_true_m2_df$wsvha_L)
tunedModel_svmLinear_m1_RMSE = RMSE(tunedModel_svmLinear_m1, faib_vri_true_m1_df$wsvha_L)

tunedModel_svm_m2_full_linear
R2(tunedModel_svmLinear_m2, y_m2)
tunedModel_svmLinear_m2_MAE
tunedModel_svmLinear_m2_RMSE
```

```
tunedModel_svmLinear_m2_test_MAE
tunedModel_svmLinear_m2_test_RMSE
tunedModel_svmLinear_m2_RMSE/tunedModel_svmLinear_m2_test_RMSE

tunedModel_svm_m1_full_linear
R2(tunedModel_svmLinear_m1, y_m1)
tunedModel_svmLinear_m1_MAE
tunedModel_svmLinear_m1_RMSE
tunedModel_svmLinear_m1_test_MAE
tunedModel_svmLinear_m1_test_RMSE
tunedModel_svmLinear_m1_RMSE/tunedModel_svmLinear_m1_test_RMSE

tunedModel_svm_m1_to_raster_linear <- raster::predict(covs_m1, tuneResult_svm_m1_full_linear)
tunedModel_svm_m2_to_raster_linear <- raster::predict(covs_m2, tuneResult_svm_m2_full_linear)
tunedModel_svm_m1_to_raster_linear_gaspard <- raster::predict(covs_m1_gaspard, tuneResult_svm_m1_full_l
tunedModel_svm_m2_to_raster_linear_gaspard <- raster::predict(covs_m2_gaspard, tuneResult_svm_m2_full_l
tunedModel_svm_m1_to_raster_linear_quesnel <- raster::predict(covs_m1_quesnel, tuneResult_svm_m1_full_l
tunedModel_svm_m2_to_raster_linear_quesnel <- raster::predict(covs_m2_quesnel, tuneResult_svm_m2_full_l
tunedModel_svm_m1_to_raster_linear$layer[tunedModel_svm_m1_to_raster_linear$layer <= 0] = 0
tunedModel_svm_m2_to_raster_linear$layer[tunedModel_svm_m2_to_raster_linear$layer <= 0] = 0
tunedModel_svm_m1_to_raster_linear_gaspard$layer[tunedModel_svm_m1_to_raster_linear_gaspard$layer <= 0]
tunedModel_svm_m2_to_raster_linear_gaspard$layer[tunedModel_svm_m2_to_raster_linear_gaspard$layer <= 0]
tunedModel_svm_m1_to_raster_linear_quesnel$layer[tunedModel_svm_m1_to_raster_linear_quesnel$layer <= 0]
tunedModel_svm_m2_to_raster_linear_quesnel$layer[tunedModel_svm_m2_to_raster_linear_quesnel$layer <= 0]
writeRaster(tunedModel_svm_m1_to_raster_linear, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/w
writeRaster(tunedModel_svm_m2_to_raster_linear, filename = "/media/seamus/128GB_WORKD/data/raster/tcc/w
writeRaster(tunedModel_svm_m1_to_raster_linear_gaspard, filename = "/media/seamus/128GB_WORKD/data/raste
writeRaster(tunedModel_svm_m2_to_raster_linear_gaspard, filename = "/media/seamus/128GB_WORKD/data/raste
writeRaster(tunedModel_svm_m1_to_raster_linear_quesnel, filename = "/media/seamus/128GB_WORKD/data/raste
writeRaster(tunedModel_svm_m2_to_raster_linear_quesnel, filename = "/media/seamus/128GB_WORKD/data/raste
```


Model1 NO−STEMS (SVM Linear)
Model1 NO−STEMS (SVM Linear)
Model2 WITH−STEMS (SVM Linear)
Model2 WITH−STEMS (SVM Linear)
Frequency
v