Appendix A

Dana Melamed

July 15, 2018

# Prepare input rasters  
  
## 1. Data  
### Study area polygon  
study\_area <- rgdal::readOGR(dsn = file.path("E:","study\_area"), layer = "study\_area")  
  
### DEM   
DEM <- raster::raster("C://nsdnr//dp055v2gr//e055ns20//grid//dem020hy")  
  
### Other input rasters  
  
  
## Crop, reproject and resample the DEM  
DEM <- raster::crop(DEM, study\_area) # crop & reproject  
temp\_raster <- raster(ext = extent(DEM),  
 resolution = res(DEM)) # dummy raster to resample to  
DEM <- raster::resample(DEM, temp\_raster, method = "bilinear")   
par(mfrow = c(2,2))  
  
  
# Create raster brick  
## Set exactly similar properties by resampling to solar raster  
solar\_raster\_df <- make\_temporal\_raster\_df("E://GOES\_200m",  
 ymd('2016-06-01'),  
 ymd('2016-06-08'),  
 date\_chars= c(16,-4),  
 date\_format = "%Y\_%j")  
solar\_raster <- raster(solar\_raster\_df[[1]][[1]])  
for(i in seq\_along(rasters\_list)){  
 rasters\_list[[i]] <- raster::resample(rasters\_list[[i]], solar\_raster)  
}  
  
# Create raster brick  
## Set exactly similar properties by resampling to solar raster  
solar\_raster\_df <- make\_temporal\_raster\_df("E://GOES\_200m",  
 ymd('2016-06-01'),  
 ymd('2016-06-08'),  
 date\_chars= c(16,-4),  
 date\_format = "%Y\_%j")  
solar\_raster <- raster(solar\_raster\_df[[1]][[1]])  
for(i in seq\_along(rasters\_list)){  
 rasters\_list[[i]] <- raster::resample(rasters\_list[[i]], solar\_raster)  
}  
  
  
# Modify the Aspect and PTOC Rasters   
  
##### Recalculate Aspect Raster #####  
asp\_raster\_in <- raster(file.path("Rasters","200","asp.tif"))  
asp\_raster\_out <- abs(asp\_raster - 180)  
writeRaster(asp\_raster\_out, file.path("Rasters","200","asp.tif"),  
 overwrite = TRUE)  
  
##### Limit Proximity to the Coastline Raster #####  
limit <- 30000 # metres  
ptoc\_raster\_in <- raster(file.path("Rasters","200","ptoc.tif"))  
ptoc\_raster\_out <- ptoc\_raster\_in  
ptoc\_raster\_out[ptoc\_raster\_in >= limit] <- limit  
  
gam\_six\_years\_lim <- gam(temp\_mean ~   
 s(dem, k = 9) +  
 s(ptoc, k = 3) +   
 s(sum\_irradiance, k= 9) +   
 s(east,north),  
 data = model\_stations\_df)  
  
  
daily\_temperatures\_df <- read.csv(file.path("E:","WeatherData","daily\_20110101\_20180218"))  
# Add stations easting and northing to data frame  
ns\_stations\_in <- rgdal::readOGR(dsn = file.path("E:","WeatherData"), layer = "NSWeatherStns")  
ns\_stations <- ns\_stations %>% dplyr::select(stationid = StnID, NORTHING = Northing, EASTING = Easting)  
  
swns\_stations\_df <- left\_join(daily\_temperatures\_df,   
 ns\_stations,   
 by = "stationid") %>%  
 extract\_constant\_raster\_values(rasters\_list)  
  
solar\_irradiance\_rasters\_df <- make\_temporal\_raster\_df(  
 in\_folder = "E:\\GOES\_200m",  
 start\_date = ymd('2012-01-01'),  
 end\_date = ymd('2017-12-31'),  
 date\_chars = c(16,-5),  
 date\_format = "%Y\_%j",  
 extension = ".tif")  
  
swns\_stations\_df <- extract\_temporal\_raster\_values(temporal\_rasters\_df = solar\_irradiance\_rasters\_df,  
 temperatures\_df = swns\_stations\_df,  
 col\_name = "sum\_irradiance")  
  
## Generate raster brick of constant rasters  
# List of raster names  
rasters\_names\_list <- list("dem", # elevation  
 "ptoc", # proxmity to coast  
 "east", # easting  
 "north",# northing  
 "asp", # aspect  
 "tpi", # topographic position index  
 "slope") # slope  
# List of raster objects  
rasters\_list <- lapply(FUN = raster,  
 X = paste0("E:\\Packages\\swnsmodelr\\Rasters\\200\\",  
 rasters\_names\_list,   
 ".tif"))  
  
# Brick of rasters  
rasters\_brick <- rasters\_list %>% brick()   
  
## Example of how solar radiation rasters are added based on date  
# Make dataframe of solar raster paths and dates  
solar\_rasters\_df <- make\_temporal\_raster\_df(file.path("E:","GOES\_200m"),  
 ymd("2016-01-01"),  
 ymd("2016-12-31"),  
 date\_chars = c(16, -5),  
 date\_format = "%Y\_%j")  
# Filter data frame for solar raster path based on date  
solar\_now\_df <- solar\_rasters\_df %>% filter(date\_time == ymd("2016-04-01"))  
# Generate raster from path  
solar\_raster\_now <- solar\_now\_df[[1]] %>% raster()  
  
## Add solar raster to brick  
rasters\_brick <- addLayer(rasters\_brick, solar\_raster\_now)  
  
## After modelling for date is finished, drop solar raster from brick  
rasters\_brick <- dropLayer(rasters\_brick, solar\_raster\_now)  
  
# Testing models with different daily temperature variables and timeframes  
  
##### All years timeframe #####  
temp\_var <- list("min", "max", "mean")  
all\_years <- list()  
all\_years\_val <- list()  
for(i in seq\_along(temp\_var)){  
 all\_years[[i]] <- gam(formula(paste0("temp\_",temp\_var," ~  
 s(dem,month) +  
 s(ptoc,month, k= 3) +  
 s(sum\_irradiance, month) +  
 s(tpi,month) +  
 s(asp, month) +  
 s(east,north) +  
 s(week) +  
 year")),   
 data = model\_stations\_df)  
 all\_years\_val[[i]] <- add\_residuals(val\_df\_2012, all\_years[[i]])  
  
   
 if(is.na(all\_years\_val[[i]]$abs\_resid)){  
 all\_years[[i]] <- gam(formula(paste0("temp\_",temp\_var[[i]]," ~  
 s(dem,month) +  
 s(ptoc,month, k= 3) +  
 s(tpi,month)+  
 s(asp, month) +  
 s(east,north) +  
 s(week) +  
 year")),   
 data = model\_stations\_df)  
 all\_years\_val[[i]] <- add\_residuals(val\_df\_2012, all\_years[[i]])  
 }  
 all\_years\_val[[i]]$timeframe <- "All years"  
 all\_years\_val[[i]]$knots <- "No limit"  
 all\_years\_val[[i]]$temp\_var <- temp\_var[[i]]  
 all\_years\_val[[i]]$gcv <- all\_years[[i]]$gcv.ubre  
 all\_years\_val[[i]]$rsq <- summary(all\_years[[i]])[[10]]  
 all\_years\_val[[i]]$dev <- summary(all\_years[[i]])[[14]]  
 all\_years\_val[[i]]$abs\_resid <- abs(all\_years\_val[[i]]$resid)  
 for(l in seq\_along(summary(all\_years[[i]])[[7]])){  
 all\_years\_val[[i]]$var\_pval <- summary(all\_years[[i]])[[8]][[l]]  
 names(all\_years\_val[[i]])[names(all\_years\_val[[i]]) == "var\_pval"] <- names(summary(all\_years[[i]])[[7]])[[l]]  
 }  
}  
all\_years\_val$timeframe <- "All years"  
  
##### Annual timeframe ####  
annual <- list()  
annual\_val <- list()  
for(i in seq\_along(temp\_var)){   
 annual[[i]] <- gam(formula(paste0("temp\_",temp\_var[[i]]," ~  
 s(dem,week) +  
 s(ptoc,week, k= 3)+  
 s(sum\_irradiance, week) +  
 s(tpi,week)+  
 s(asp, week) +  
 s(east,week) +  
 s(yday) +  
 month")),   
 data = model\_df\_2012)  
 annual\_val[[i]] <- add\_residuals(val\_df\_2012, annual[[i]])  
  
   
 if(is.na(annual\_val[[i]]$abs\_resid)){  
 annual[[i]] <- gam(formula(paste0("temp\_",temp\_var[[i]]," ~  
 s(dem,week) +  
 s(ptoc,week, k= 3)+  
 s(tpi,week)+  
 s(asp, week) +  
 s(east,week) +  
 s(yday) +  
 month")),   
 data = model\_df\_2012)  
 annual\_val[[i]] <- add\_residuals(val\_df\_2012, annual[[i]])  
 }  
 for(l in seq\_along(summary(annual[[i]])[[7]])){  
 annual\_val[[i]]$var\_pval <- summary(annual[[i]])[[8]][[l]]  
 names(annual\_val[[i]])[names(annual\_val[[i]]) == "var\_pval"] <- names(summary(annual[[i]])[[7]])[[l]]  
 }  
 annual\_val[[i]]$timeframe <- "Annual"  
 annual\_val[[i]]$temp\_var <- temp\_var[[i]]  
 annual\_val[[i]]$gcv <- annual[[i]]$gcv.ubre  
 annual\_val[[i]]$rsq <- summary(annual[[i]])[[10]]  
 annual\_val[[i]]$knots <- "No limit"  
 annual\_val[[i]]$dev <- summary(annual[[i]])[[14]]  
 annual\_val[[i]]$abs\_resid <- abs(annual\_val[[i]]$resid)  
   
}  
  
  
  
##### Monthly Timeframe #####  
# Create list of three dataframes  
monthly\_val\_list <- list()  
# Loop over each option: min, max and mean  
for(i in seq\_along(temp\_var)){  
 monthly\_val\_list[[i]] <- validate\_monthly\_GAMs(model\_stations\_df = model\_stations\_df,  
 val\_stations\_df = val\_stations\_df,  
 years = 2012,  
 months = 1:12,  
 formula =paste0("temp\_",temp\_var[[i]],"~  
 s(east,north) +  
 s(dem, yday) +  
 s(sum\_irradiance, yday) +  
 s(tpi, yday) +   
 s(asp, yday) +  
 s(ptoc, k = 3) +  
 week"),  
 alt\_formula =paste0("temp\_",temp\_var[[i]],"~  
 s(east,north, yday) +  
 s(dem, yday) +  
 s(tpi, yday) +   
 s(asp, yday)  
 s(ptoc, yday, k = 3) +  
 week")  
 )   
 monthly\_val\_list[[i]]$temp\_var <- temp\_var[[i]]  
}  
  
monthly\_val <- dplyr::bind\_rows(monthly\_val\_list)  
monthly\_val$timeframe <- "Monthly"  
  
  
##### Weekly Timeframe #####  
# Create list of three dataframes  
weekly\_val\_list <- list()  
# Loop over each option: min, max and mean  
for(i in seq\_along(temp\_var)){  
 weekly\_val\_list[[i]] <- validate\_weekly\_GAMs(model\_stations\_df = model\_stations\_df,  
 val\_stations\_df = val\_stations\_df,  
 years = 2012,  
 weeks = 1:52,  
 formula =paste0("temp\_",temp\_var[[i]],"~  
 s(east,north) +  
 s(dem, yday) +  
 s(sum\_irradiance, yday) +  
 s(asp, yday) +  
 s(tpi, yday) +   
 s(ptoc,yday)"),  
 alt\_formula =paste0("temp\_",temp\_var[[i]],"~  
 s(east,north) +  
 s(dem, yday) +  
 s(asp, yday) +  
 s(tpi, yday) +   
 s(ptoc,yday)")  
 )   
 weekly\_val\_list[[i]]$temp\_var <- temp\_var[[i]]  
   
}  
  
weekly\_val <- dplyr::bind\_rows(weekly\_val\_list)  
weekly\_val$timeframe <- "weekly"  
  
  
##### Daily Timeframe #####  
# Create list of three dataframes  
daily\_val\_list <- list()  
# Loop over each option: min, max and mean  
for(i in seq\_along(temp\_var)){  
 daily\_val\_list[[i]] <- validate\_daily\_GAMs(model\_stations\_df = model\_stations\_df,  
 val\_stations\_df = val\_stations\_df,  
 years = 2012,  
 days = 1:365,  
 formula =paste0("temp\_",temp\_var[[i]],"~  
 s(east,north) +  
 s(dem) +  
 s(sum\_irradiance) +  
 s(tpi) +   
 s(ptoc, k = 3)"),  
 alt\_formula =paste0("temp\_",temp\_var[[i]],"~  
 s(east,north) +  
 s(dem) +  
 s(tpi) +   
 s(ptoc, k = 3)")  
 )   
 daily\_val\_list[[i]]$temp\_var <- temp\_var[[i]]  
 print(temp\_var[[i]])  
   
}  
daily\_val <- dplyr::bind\_rows(daily\_val\_list)  
daily\_val$timeframe <- "Daily"