Cherry blossom predictions 2024

Andrew McDougall

Load required packages

{r} rm(list = ls()) library(tidyverse) library(mgcv) library(gratia) library(dplyr)

Read the historic long term data for visualisation

{r} cherry <- read.csv("washingtondc.csv") %>% bind_rows(read.csv("liestal.csv")) %>% bind_rows(read.csv("kyoto.csv"))

Visualize as time series with linear trend

{r} cherry %>% filter(year >= 1880) %>% ggplot(aes(x = year, y = bloom_doy)) + geom_smooth(method = lm) + scale_x_continuous(breaks = seq(1880, 2020, by = 20)) + facet_grid(cols = vars(str_to_title(location))) + labs(x = "Year", y = "Peak bloom (days since Jan 1st)")+ theme(panel.grid = element_blank())

Now visualize as time series with gam smoothing

{r} cherry %>% filter(year >= 1880) %>% ggplot(aes(x = year, y = bloom_doy)) + geom_smooth(method = "gam", formula = y ~ s(x, bs = "cs")) + scale_x_continuous(breaks = seq(1880, 2020, by = 20)) + facet_grid(cols = vars(str_to_title(location))) + labs(x = "Year", y = "Peak bloom (days since Jan 1st)")+ theme(panel.grid = element_blank())

Read climate data file with Japan, Kyoto, Liestal, and Washington bloom_doy, and all other Japan sites

{r} Wdata <- read.csv("siteclimatedata.csv", header = TRUE, sep = ",")

Fit gam model for climate

{r} gam_fit <- gam(bloom_doy ~ s(Feb_max) + s(Jan_max, k = 3), data = Wdata, subset = year >= 1880, method = "REML")

Model summary

{r} summary(gam_fit)

{r}appraise(gam_fit)

{r} draw(gam_fit)

Read historic Vancouver climate data

{r} Vdata <- read.csv("vancouver.csv", header = TRUE, sep = ",") #predict (hindcast) for the Vancouver climate data {r} vancouver_doy <- predict(gam_fit, newdata = Vdata)

Round estimates and attach estimates to data

{r} vc_doy <- as.integer(vancouver_doy)

{r} vc_doy

{r} Vdata1 <- Vdata %>% mutate(vc_doy) {r} Vdata2 <- Vdata1 %>% rename(bloom_doy = vc_doy) {r} Wdata1 <- Wdata %>% bind_rows(Vdata2)

Read the historic New York climate data

{r} NYdata<- read.csv("new_york.csv", header = TRUE, sep = ",") {r} ny_doy <- predict(gam_fit, newdata = NYdata)

Round estimates and attach estimates to the data

{r} ny_doy <- as.integer(ny_doy)

{r} ny_doy

{r} NYdata1 <- NYdata %>% mutate(ny_doy) {r} NYdata2 <- NYdata1 %>% rename(bloom_doy = ny_doy) {r} Wdata2 <- Wdata1 %>% bind_rows(NYdata2)

Climate data from NOAA, WeatherUnderground for January 2024 and February 2024 last checked 27th February # + 'topped-up' with forecasts, e.g. BBC weather, meteoblue etc.

{r} CL24 <- read.csv("Climate2024predprecip.csv", header = TRUE, sep = ",")

Read worldwide bloom_doy WITH hindcast Vancouver and New York data added on

{r}W1data <- read.csv("siteclimatedatawv.csv", header = TRUE, sep = ",")

Make location a factor

{r}CL24$location <- as.factor(CL24$location)

{r} class(CL24$location)

{r} Wdata2$location <-as.factor(Wdata2$location)

{r} class(Wdata2$location)

fit model for prediction

{r} gam_fit_I <- gam(bloom_doy ~ s(Feb_max, by = location) + s(Jan_max, k = 3), data = Wdata2) {r} summary(gam_fit_I) {r} appraise(gam_fit_I)

Predict fort the 5 sites for 2024

{r} predictions_gam1 <- expand_grid(location = unique(CL24$location), year = 2024) %>% bind_cols(predicted_doy1 = predict(gam_fit_I, newdata = CL24))

{r} predictions_gam1

View 2024 prediction for each site

{r} gampred1 <- predictions_gam1 %>% group_by(year,location) %>% slice_tail(n = 1)

{r} gampred1

{r} print(gampred1 [1:5,])

Round the predictions

{r} submission_predictions <- gampred1 %>% filter(year > 2023) %>% mutate(predicted_doy1 = round(predicted_doy1)) {r} submission_predictions

Ssubtract values from columns to bring estimates to the regionally appropriate phase. Replace estimated Washington D.C. and Vancouver and New York 2024 values (80% flowers open) with regionally used value (70% flowers open) by indexing with bloom_doy - 1. Explanation: the 70% value is estimated at one(1) day prior to the estimate value, i.e. 80% of flowers that are possible to be open concurrently

{r} submission_predictions <- submission_predictions %>% mutate(predicted_doy1 = ifelse(location %in% c("washingtondc", "vancouver", "newyorkcity"), predicted_doy1 - 1, predicted_doy1))

{r} submission_predictions

Replace Liestal 2024 estimate value (80% flowers open) with regionally used value (25% flowers open) by indexing with #bloom_doy - 5. Explanation: the 25% value is estimated at five (5) days prior to the estimate value, i.e. 80% of flowers that are possible to be open concurrently

{r} submission_predictions <- submission_predictions %>% mutate(predicted_doy1 = ifelse(location %in% c("liestal"), predicted_doy1 - 5, predicted_doy1))

View

{r} submission_predictions

Place the estimates in the order required for the competition

{r} submission_predictions <- submission_predictions %>% filter(year > 2023) %>% arrange(case_when(location == "washingtondc" ~ 1, location == "liestal" ~ 2, location == "kyoto" ~ 3, location == "vancouver" ~ 4, location == "newyorkcity" ~ 5))

{r} submission_predictions

Create a data frame with the reordered predictions

{r} output_data <- data.frame(location = c("washingtondc", "liestal", "kyoto", "vancouver", "newyorkcity"), prediction = submission_predictions$predicted_doy1)

Write the data frame to a CSV file

{r} write.csv(output_data, file = "predictions.csv", row.names = FALSE) —