CherryBlssm_Pred

Marie Han

2026-02-27

Loading training data

```r
cherry <- read.csv("data/washingtondc.csv") |>
  bind_rows(read.csv("data/liestal.csv")) |>
  bind_rows(read.csv("data/kyoto.csv")) |>
  bind_rows(read.csv("data/vancouver.csv")) |>
  bind_rows(read.csv("data/nyc.csv"))
```

Function to obtain weather data:

```r
get_ghcn_station= function(station_id, destdir = tempdir()) {
  base_url= "https://www.ncei.noaa.gov/pub/data/ghcn/daily/by_station/"
    file_name= paste0(station_id, ".csv.gz")
    url= paste0(base_url, file_name)
    destfile= file.path(destdir, file_name)
    if (!file.exists(destfile)) {
      download.file(url, destfile, mode = "wb")
    }
    data=read.csv(destfile, header = FALSE)
    data=data[, -c(5:8)]
    data$Date=as.Date(as.character(data$V2), "%Y%m%d")
    data$Year=as.integer(format(data$Date, "%Y"))

    #Pivot to get vars as columns
    data_wide=data %>%
      select(Date, Year, V3, V4) %>%
      pivot_wider(
        names_from = V3,
        values_from = V4)

    # Calculate TAVG if TMAX and TMIN exist
    if (all(c("TMAX", "TMIN") %in% colnames(data_wide))) {
      data_wide= data_wide %>%
        mutate(TAVG = (TMAX + TMIN)/2)
    }

    # Keep only the desired elements
    elements_to_keep= c("TMIN", "TMAX", "TAVG", "PRCP")
    data_wide= data_wide %>%
      select(Date, Year, any_of(elements_to_keep))
    # Remove rows without both TMIN and TMAX
    data_wide= data_wide %>%
      filter(!is.na(TMIN) & !is.na(TMAX))

    return(data_wide)
}
```

Weather by location, putting it into one “daily” dataframe.

```
kyototemp=get_ghcn_station("JA000047759")
washdc=get_ghcn_station("USW00013743")
vancouvertemp=get_ghcn_station("CA001108395")
newyorkcity=get_ghcn_station("USW00014732")
liestaltemp=get_ghcn_station("SZ000001940")

#Adding a location variable for identification
kyototemp$location <- "kyoto"
liestaltemp$location <- "liestal"
newyorkcity$location <- "newyorkcity"
vancouvertemp$location <- "vancouver"
washdc$location <- "washingtondc"
```

```
library("rvest")
```

```
##
## Attaching package: ’rvest’

## The following object is masked from ’package:readr’:
##
##     guess_encoding
```

```
AccuWeather <- read.csv("accuweather_forecast_2026.csv")


get_weather_table <- function(url)
  read_html(url) %>%
  html_nodes("div.monthly-calendar") %>%
  html_text2() %>%
  str_replace("N/A", "N/A N/A") %>%
  str_remove_all("°|Hist. Avg. ") %>%
  str_split(" ", simplify = TRUE) %>%
  parse_number() %>%
  matrix(ncol = 3,
         byrow = TRUE,
         dimnames = list(NULL, c("day", "tmax", "tmin"))) %>%
  as_tibble() %>%
  filter(row_number() >= day) %>%
  filter(!duplicated(day))

vancouver2025 <-
  tibble(
    base_url = "https://web.archive.org/web/20260224/https://www.accuweather.com/en/us/vancouver/98661/
    month = month.name[10:12],
    year = 2025,
    url = str_c(base_url, tolower(month), "-weather/331419?year=", year)) %>%
  mutate(temp = map(url, get_weather_table)) %>%
  pull(temp) %>%
  reduce(bind_rows) %>%
  transmute(date = seq(as.Date("2025-10-01"), as.Date("2025-12-31"), 1),
            year = parse_number(format(date, "%Y")),
            tmax,
            tmin,
            temp = (tmax + tmin) / 2)

vancouver2026<-
  tibble(
    base_url = "https://web.archive.org/web/20260224/https://www.accuweather.com/en/us/vancouver/98661/
    month = month.name[1:2],
    year = 2026,
    url = str_c(base_url, tolower(month), "-weather/331419?year=", year)) %>%
  mutate(temp = map(url, get_weather_table)) %>%
  pull(temp) %>%
  reduce(bind_rows) %>%
  transmute(date = seq(as.Date("2026-01-01"), as.Date("2026-02-28"), 1),
            year = parse_number(format(date, "%Y")),
            tmax,
            tmin,
            temp = (tmax + tmin) / 2)

vancouver_binded <- vancouver2025 %>%
  bind_rows(vancouver2026)   %>%
  mutate(
    Date = as.Date(date),
    year = year(Date),
    TMIN = tmin,
    TMAX = tmax,
    TAVG = temp,
    PRCP = NA,
    location = "vancouver",
    month = month(Date),
    bloom_year = ifelse(month >= 10, year + 1, year)
  ) %>%
  select(-date, -tmin, -tmax, -temp)
```

```
#Now putting them all into one file
Cherry_day <- bind_rows(kyototemp,liestaltemp, newyorkcity, vancouvertemp, washdc) %>%
  rename(year = Year)%>% #to match with the original cherry year label
  mutate(Date = as.Date(Date),
         year = as.numeric(format(Date, "%Y")),
         month = as.numeric(format(Date, "%m")),
         bloom_year = ifelse(month >= 10, year+1, year)
         )

#Adding on the vancouver weather data from this year
Cherry_day <- bind_rows(Cherry_day, vancouver_binded)
```

Adding precipitation counter, I wanted to capture the rain accumulation over a week, as an inch per week is considered ideal for cherry tree growth and health. This is also where I had to start collapsing each individual day's min/max/average temperature, in order to start looking at things more broadly. Because this is only looking at data starting in October until February, I was not expecting a lot of really hot weather to begin with, but I wasn't counting out the possibility.

```
#To filter things weekly ###
Cherry_week <- Cherry_day %>%
  filter(month >= 10 | month <= 2) %>%
  mutate(week = isoweek(Date)
  ) %>%
  group_by(location, bloom_year, week) %>%
  summarise(
    weekly_PRCP = sum(PRCP,na.rm = TRUE),
    mean_TAVG   = mean(TAVG),
    min_TAVG    = min(TAVG),
    max_TAVG    = max(TAVG),
    .groups = "drop"
  ) %>%
  mutate(
    PRCP_indicator    = ifelse(weekly_PRCP > 254, 1, 0),  # > 254 tenths of mm, as is meausured by data
    TooCold_indicator = ifelse(min_TAVG < -29, 1, 0), #more important if it's been too cold for an enti
    TooHot_indicator  = ifelse(max_TAVG > 29, 1, 0)
  )
```

Now to make our data usable in the model and prediction, collapsing and summarizing it into a year in order to match the original "cherry" format.

```
Cherry_Year <- Cherry_week %>%
  group_by(location,bloom_year) %>%
  summarize(
    precipitation = sum(PRCP_indicator),
    avg_TMAX = mean(max_TAVG),
    avg_TMIN = mean(min_TAVG),
    AVG = mean(mean_TAVG),
    TooCold = sum(TooCold_indicator),
    TooHot = sum(TooHot_indicator),
    .groups = "drop"
) %>%
  #select(-year) %>%
  rename(year = bloom_year ) %>%
  mutate(precipitation = replace_na(precipitation, 0))
```

Model

The too hot indicator was insignificant due to it being fall and winter time during our windowed time frame. It's very unlikely that any of these locations would have a week or more long heatwave, but there is always the possibility.

```
model1<- cherry |>
  left_join(Cherry_Year,
            by = c("location", "year")) |>
  lm(formula = bloom_doy ~ year * location +  AVG + TooCold+TooHot +precipitation )
```

```r
summary(model1)
```

```
##
## Call:
## lm(formula = bloom_doy ~ year * location + AVG + TooCold + TooHot +
##     precipitation, data = left_join(cherry, Cherry_Year, by = c("location",
##     "year")))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -22.9175  -4.1267   0.0169   4.6572  18.2862
##
## Coefficients: (1 not defined because of singularities)
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                2.989e+02  9.522e+01   3.139  0.00188 **
## year                      -9.493e-02  4.814e-02  -1.972  0.04964 *
## locationliestal           -1.789e+01  1.018e+02  -0.176  0.86069
## locationnewyorkcity       -7.748e+00  8.050e+00  -0.963  0.33663
## locationvancouver         -2.383e+03  9.180e+03  -0.260  0.79536
## locationwashingtondc      -1.001e+02  1.188e+02  -0.842  0.40030
## AVG                       -1.250e-01  5.990e-02  -2.087  0.03785 *
## TooCold                    9.543e-01  3.000e-01   3.181  0.00164 **
## TooHot                    -2.713e-02  2.820e-01  -0.096  0.92342
## precipitation             -1.187e-01  2.322e-01  -0.511  0.60955
## year:locationliestal       4.499e-03  5.148e-02   0.087  0.93041
## year:locationnewyorkcity          NA         NA      NA       NA
## year:locationvancouver     1.173e+00  4.537e+00   0.259  0.79618
## year:locationwashingtondc  4.451e-02  5.990e-02   0.743  0.45813
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 7.501 on 274 degrees of freedom
##   (790 observations deleted due to missingness)
## Multiple R-squared:  0.3898, Adjusted R-squared:  0.363
## F-statistic: 14.58 on 12 and 274 DF,  p-value: < 2.2e-16
```

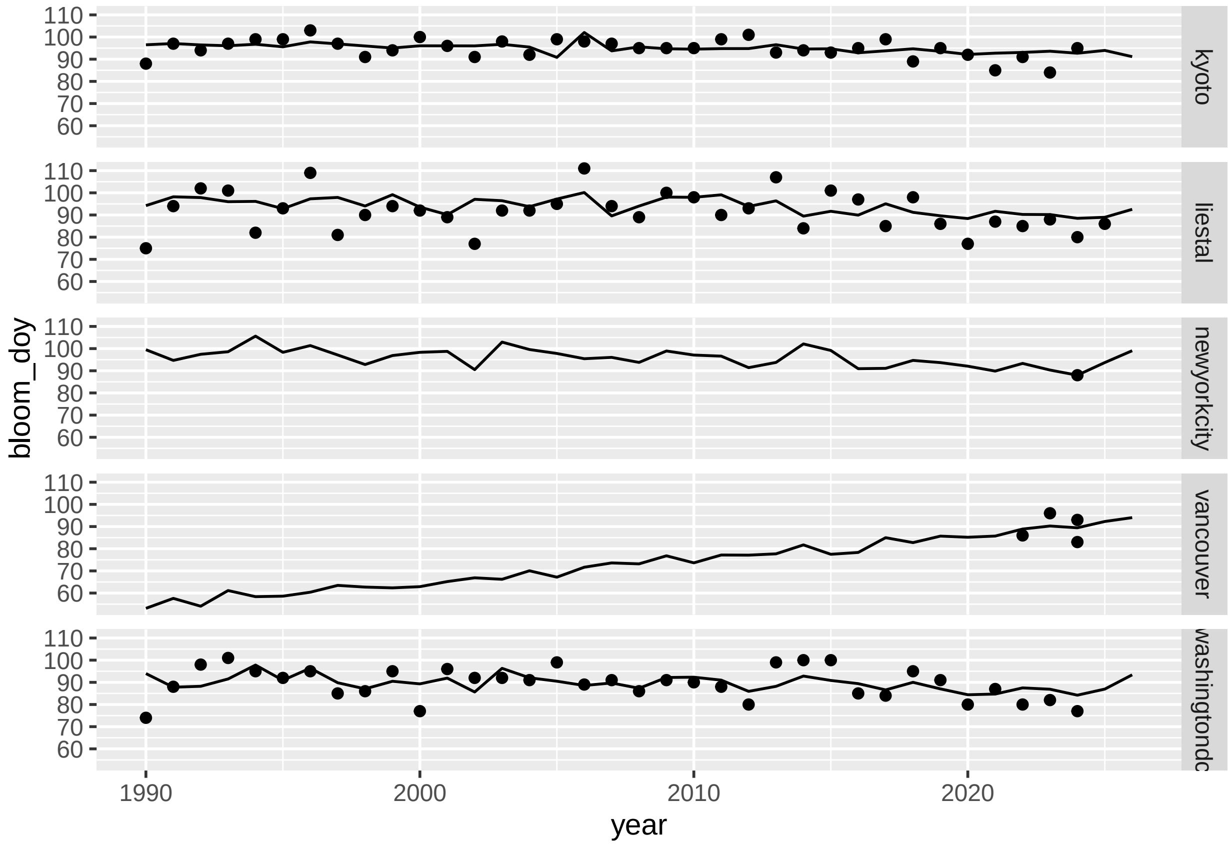
```r
###TEST PREDICTIONS
cherry_gridTEST <- expand_grid(location = unique(cherry$location),
                           year = 1990:2026) |>
  inner_join(Cherry_Year,
             by = c("location", "year"))

predictions2 <- cherry_gridTEST |>
  mutate(pred_bloom = predict(model1, newdata = cherry_gridTEST))
```

```
## Warning: There was 1 warning in ‘mutate()‘.
## i In argument: ‘pred_bloom = predict(model1, newdata = cherry_gridTEST)‘.
## Caused by warning in ‘predict.lm()‘:
## ! prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases
```

```
predictions2 |>
  left_join(cherry,
            by = c("location", "year")) |>
  ggplot(aes(x = year)) +
  geom_point(aes(y = bloom_doy)) +
  geom_line(aes(y = pred_bloom)) +
  facet_grid(rows = vars(location))
```

```
## Warning: Removed 75 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```



Submission

```
predictions2 |>
  filter(year == 2026) |>
  mutate(predicted_date= strptime(paste(year,pred_bloom), "%Y %j") |>
           as_date())
```

```
## # A tibble: 5 x 10
##   location  year precipitation avg_TMAX avg_TMIN   AVG TooCold TooHot pred_bloom
##   <chr>    <dbl>         <dbl>    <dbl>    <dbl> <dbl>   <dbl>  <dbl>      <dbl>
## 1 washing~  2026             4     103.     27.4  64.5       6     19       93.4
## 2 liestal   2026             3     87.1     33.6  58.8       3     16       92.6
## 3 kyoto     2026             0    150.      92.4 120.        0     18       91.1
## 4 vancouv~  2026             0     45.5     41.6  43.5       0     22       94.0
## 5 newyork~  2026             5     86.6     13.7  50.6       8     19       99.0
## # i 1 more variable: predicted_date <date>
```

Predicted bloom dates as of 02/27/2026

DC: 04/03/2026

Liestal: 04/02/2026

Kyoto: 04/01/2026

Vancouver: 04/04/2026

NYC: 04/08/2026