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"

#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 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 whhere 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),
    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_TEST <- cherry |>
  inner_join(Cherry_Year,
             by = c("location", "year")) |>
  lm(formula = bloom_doy ~ year * location +  AVG + TooCold+TooHot +precipitation )




summary(model1_TEST)
```

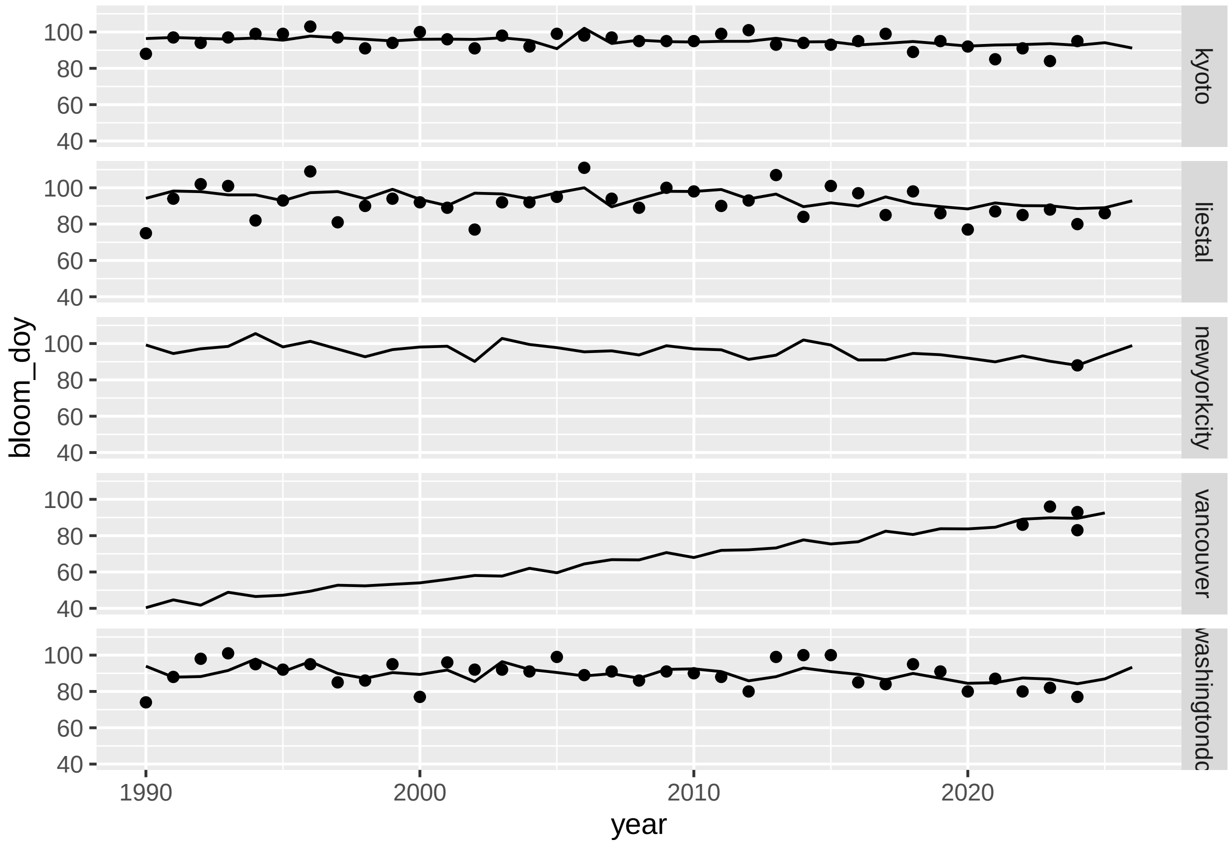
```
##
## Call:
## lm(formula = bloom_doy ~ year * location + AVG + TooCold + TooHot +
##     precipitation, data = inner_join(cherry, Cherry_Year, by = c("location",
##     "year")))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -22.9934  -4.1080   0.0256   4.6970  18.2767
##
## Coefficients: (1 not defined because of singularities)
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                2.933e+02  9.622e+01   3.048  0.00253 **
## year                      -9.206e-02  4.864e-02  -1.893  0.05944 .
## locationliestal           -1.176e+01  1.028e+02  -0.114  0.90902
## locationnewyorkcity       -8.031e+00  8.068e+00  -0.995  0.32041
## locationvancouver         -3.169e+03  9.570e+03  -0.331  0.74080
## locationwashingtondc      -9.340e+01  1.197e+02  -0.780  0.43591
## AVG                       -1.254e-01  5.991e-02  -2.093  0.03728 *
## TooCold                    9.498e-01  3.000e-01   3.166  0.00172 **
## TooHot                    -3.148e-02  2.820e-01  -0.112  0.91119
## precipitation             -8.298e-02  2.316e-01  -0.358  0.72040
## year:locationliestal       1.404e-03  5.199e-02   0.027  0.97848
## year:locationnewyorkcity          NA         NA      NA       NA
## year:locationvancouver     1.561e+00  4.730e+00   0.330  0.74163
## year:locationwashingtondc  4.111e-02  6.038e-02   0.681  0.49648
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 7.503 on 274 degrees of freedom
## Multiple R-squared:  0.3895, Adjusted R-squared:  0.3627
## F-statistic: 14.57 on 12 and 274 DF,  p-value: < 2.2e-16
```

```
###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_TEST, newdata = cherry_gridTEST))
```

```
## Warning: There was 1 warning in ‘mutate()‘.
## i In argument: ‘pred_bloom = predict(model1_TEST, 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 74 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```



Submission

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

```
## # A tibble: 4 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       93.3
## 2 liestal   2026             0     87.1     33.6  58.8       3     16       92.8
## 3 kyoto     2026             0    150.      92.4 120.        0     18       91.2
## 4 newyork~  2026             5     86.6     14.6  51.1       8     19       98.9
## # i 1 more variable: predicted_date <date>
```

Predicted bloom dates

DC: 04/03/2026

Liestal: 04/02/2026

Kyoto: 04/01/2026

NYC: 04/08/2026