Cherry Blossom Predictions

Steven Lio

2/27/2022

Introduction

This document show case the model process for the 2022 Cherry Blossom Prediction Competition from the Department of Statistics at the School of Computing at George Mason University, Virginia, detail can be find [**here**](https://competition.statistics.gmu.edu/competition-rules/). The objective of this model is to predict the Cherry Blossom day in four locations Kyoto(Japan), Liestal-Weideli(Switzerland), Washington DC (USA) and Vancouver(Canada) from 2022 to 2031. The data used in this project is mainly the historical Cherry Blossom date data in each location (except Vancouver) as well as NOAA's weather data (daily max temperature) given by NOAA's API in R (rnoaa) package. For Vancouver, a proxy data is collected from 2004 using Google Trend which indicates the popularity of the search term related to Cherry Blossom in British Columnba or Vancouver.

The original scripts and data used in this document can be found in this GitHub repo [**here**](https://github.com/stevenlio88/peak-bloom-prediction).

Data Preparation and EDA

Load Cleaned NOAA Data for each Location

NOAA were retrieved and processed from the `NOAA Weather Data.Rmd` process and raw csv file is located in `data/NOAA.csv`. Simple temperature (daily max) and other meteorological variables were pulled using `NOAA`'s API. The main variable used is `tmax` - daily max temperature from 1950 and missing data were first imputed using previous/next year values for long period missing data and short term data were imputed (`tmax_impute`) using regression method. Daily max temperature is also predicted using simple regression model (with seasonal terms - `tmax_LM_pred` and `tmax_LM_predbias`) onward up to 2031-12-31 for Cherry Blossom prediction. `tmax_LM_pred` is raw prediction from Linear Regression and `tmax_LM_predbias` added random noise to correct regression bias. Both variables will be evaluate against predicting Cherry Bloom date.

```
weather_data_complete_full <- read.csv("../data/NOAA.csv")
knitr::kable(head(weather_data_complete_full), caption = "An example of NOAA data.")
```

Table 1: An example of NOAA data.

id	location	date	year	qtr	month	day_num	doy	tmin	tavg	tmax	tmax_impute	tmax_LM_pred	tmax_LM_predbias	prcp	snow	snwd
USC00186350	Washington DC	1950-01-01	1950	Q1	1	1	1	NA	NA	11.7	11.7	9.256951	8.473451	0.0	0	0
USC00186350	Washington DC	1950-01-02	1950	Q1	1	2	2	NA	NA	14.4	14.4	9.544130	14.270696	0.0	0	0

id	location	date	year	qtr	month	day_num	doy	tmin	tavg	tmax	tmax_impute	tmax_impute_LM	tmax_pred_LM	prcp	snow	snwd
USC00186350	Washington DC	1950-01-03	1950	Q1	1	3	3	NA	NA	15.0	15.0	9.262354	8.723621	1.8	0	0
USC00186350	Washington DC	1950-01-04	1950	Q1	1	4	4	NA	NA	21.7	21.7	7.502840	7.015077	0.5	0	0
USC00186350	Washington DC	1950-01-05	1950	Q1	1	5	5	NA	NA	21.1	21.1	6.596366	21.222263	0.0	0	0
USC00186350	Washington DC	1950-01-06	1950	Q1	1	6	6	11.1	NA	23.3	23.3	5.840294	19.808059	0.0	0	0

Historical Cherry Blossom data

This model process will be focus on predicting the day of the year for which the Cherry Blossom for each location. Official data is available for Washington DC(US), Liestal(Switzerland) and Kyoto(Japan). For Vancouver(Canada), annual cherry blossom date is not officially available. The Vancouver data is obtained from 2004 using Google Trend of people who searches Cherry Blossom related terms to best estimate the cherry blossom period in hope to confine the Cherry Blossom prediction to a specific period (potential problem would be the definition of Cherry Blossom is not the same as the other locations and also there maybe a delay effect i.e. Cherry Blossom before people searches on Google). The Google Trend data is collected manually and the compilation script can be found in `GoogleTrend_Vancouver.R`. The Cherry Blossom data is then compiled and merge with NOAA's data in this following script:

Table 2: Base table for model

id	location	date	qtr	month	day_num	doy	tmin	tavg	tmax	tmax_impute	tmax_impute_LM	tmax_pred_LM	prcp	snow	snwd	lat	long	alt	bloom_doy	year	bloom_status
USC00186350	Washington DC	1950-01-01	Q1	1	1	1	NA	NA	11.7	11.7	9.256981	8.473450	1.0	0	0	38.88535	-77.03863	0	NA	1950	0
USC00186350	Washington DC	1950-01-02	Q1	1	2	2	NA	NA	14.4	14.4	9.544134	10.270696	0.0	0	0	38.88535	-77.03863	0	NA	1950	0
USC00186350	Washington DC	1950-01-03	Q1	1	3	3	NA	NA	15.0	15.0	9.262354	8.723621	1.8	0	0	38.88535	-77.03863	0	NA	1950	0
USC00186350	Washington DC	1950-01-04	Q1	1	4	4	NA	NA	21.7	21.7	7.502840	7.015077	0.5	0	0	38.88535	-77.03863	0	NA	1950	0
USC00186350	Washington DC	1950-01-05	Q1	1	5	5	NA	NA	21.1	21.1	6.596366	21.222263	0.0	0	0	38.88535	-77.03863	0	NA	1950	0
USC00186350	Washington DC	1950-01-06	Q1	1	6	6	11.1	NA	23.3	23.3	5.840294	19.808059	0.0	0	0	38.88535	-77.03863	0	NA	1950	0

Prediction of Daily Max Temperature from each location

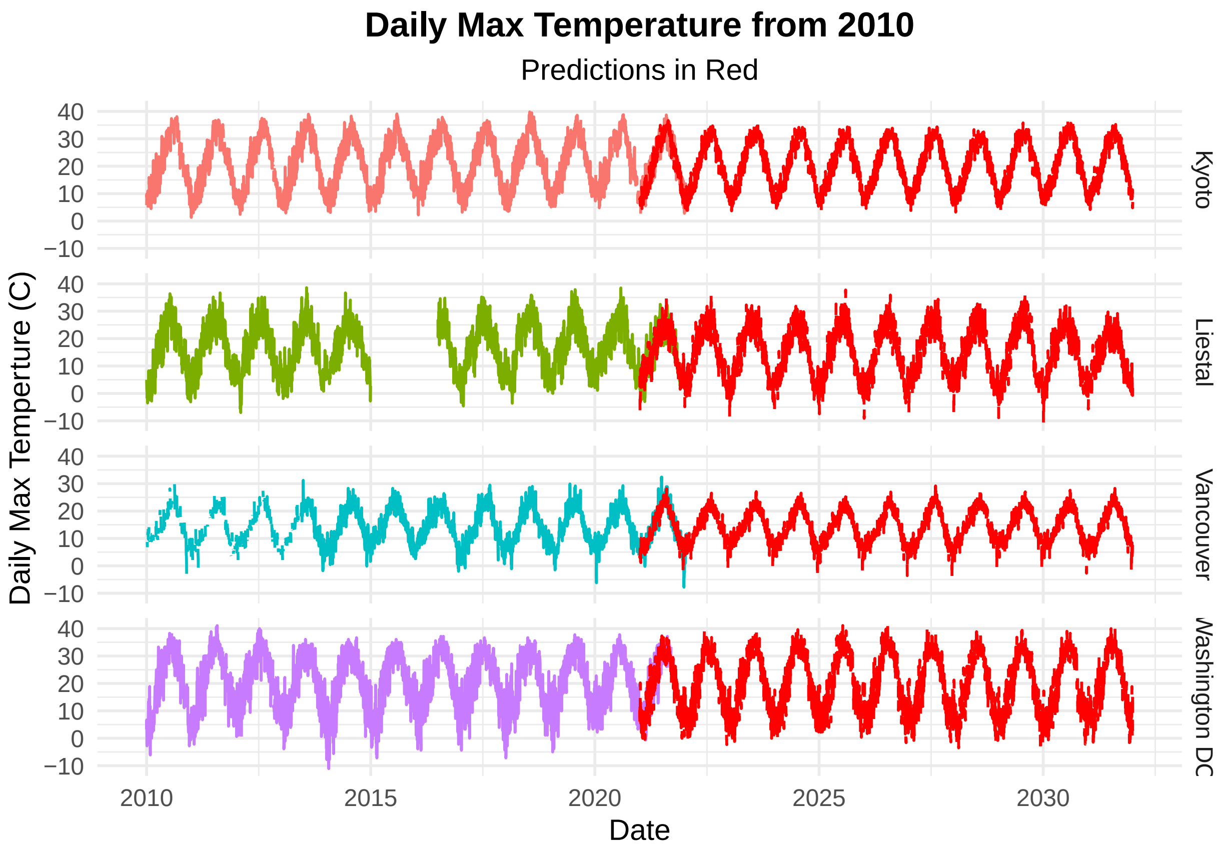

Figure 1: Daily Max Temperature (NOAA) with predictions

Custom defined Moving Average Function:

```r
#' Custom Moving Average Calculation
#'
#' @param x - Time Series sequence data
#' @param window -k-Moving Average Window
#' @param method -Average Calculation method: center-value average around current value, forward-averag
#'
#' @return - Moving Averaged time series data
#' @export
#'
#' @examples
#' x <- 1:10
#' y <- ma(x,window=3,method="center")
#' y
#' > 2.5 3.0 3.5 4.0 5.0 6.0 7.0 7.5 8.0 8.5
ma<-function(x, window=5, method="center"){
  ma_x <- c()

  for (i in 1:length(x)){
    if (method=="center"){
      ma_x[i]<-mean(x[max(1,i-window):min(length(x),i+window)],na.rm=TRUE)
      ma_x[i]<-ifelse(is.na(ma_x[i]),NA,ma_x[i])
    } else if (method=="forward") {
      ma_x[i]<-mean(x[max(1,j-window+1):j],na.rm=TRUE)
      ma_x[i]<-ifelse(is.na(ma_x[i]),NA,ma_x[i])
    } else if (method=="backward") {
      ma_x[i]<-mean(x[length(x):1][max(1,j-window+1):j],na.rm=TRUE)
      ma_x[i]<-ifelse(is.na(ma_x[i]),NA,ma_x[i])
    }
  }
  return(ma_x)
}
```

Custom defined Lag function:

```r
#' Lag function
#'
#' @param x - input sequential data
#' @param lag - k-lag
#'
#' @return lag (shifted) k number of previous values
#' @export
#'
#' @examples
my_lag <- function(x, lag=1){
  return(c(rep(NA,lag),x[1:(length(x)-lag)]))
}
```

Exploratory Data Analysis (EDA)

A quick view on the historical cherry blossom day distribution by each location. We can see that the cherry blossom day is most in-consistence (highest variance) among the four locations over the years.

Summary statistics regarding Cherry Bloom Day in each location:

Table 3: Cherry blossom day statistics

location	Number of observations	Earliest Year	Average Bloom Day	Standard Deviation of Bloom Day
Kyoto	81	1951	97.63	4.55
Liestal	79	1954	98.47	11.64
Vancouver	75	2004	94.58	9.19
Washington DC	82	1950	93.36	6.77

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

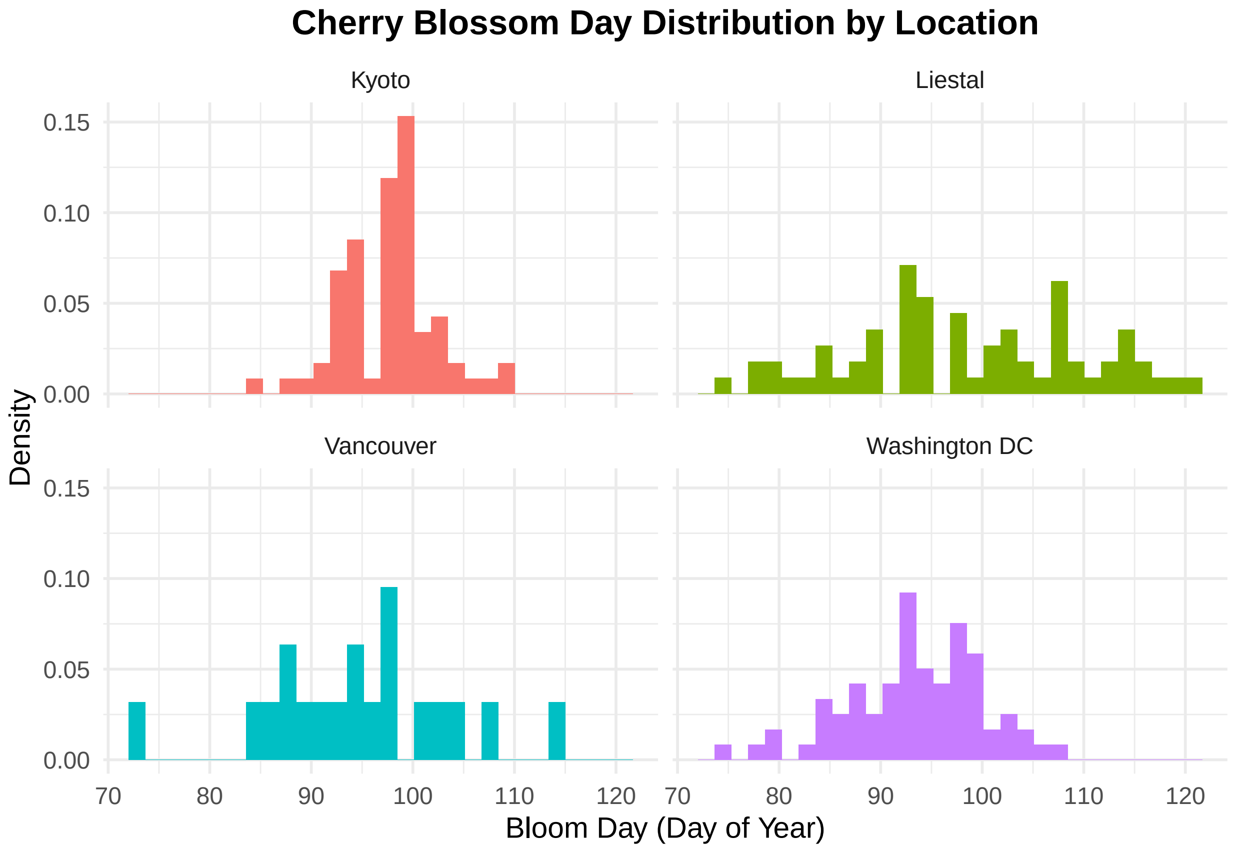

Figure 2: Historical Cherry Blossom Day Density Plot

From below graphs we see the bloom date over the years generally been pushed earlier and coincided with warmer average temperature in recent years except for Vancouver which has seen cooler average temperature in recent year and the estimated bloom date has been pushed back. Kyoto has most consistence bloom date and Liestal has most variance in the bloom date over the years.

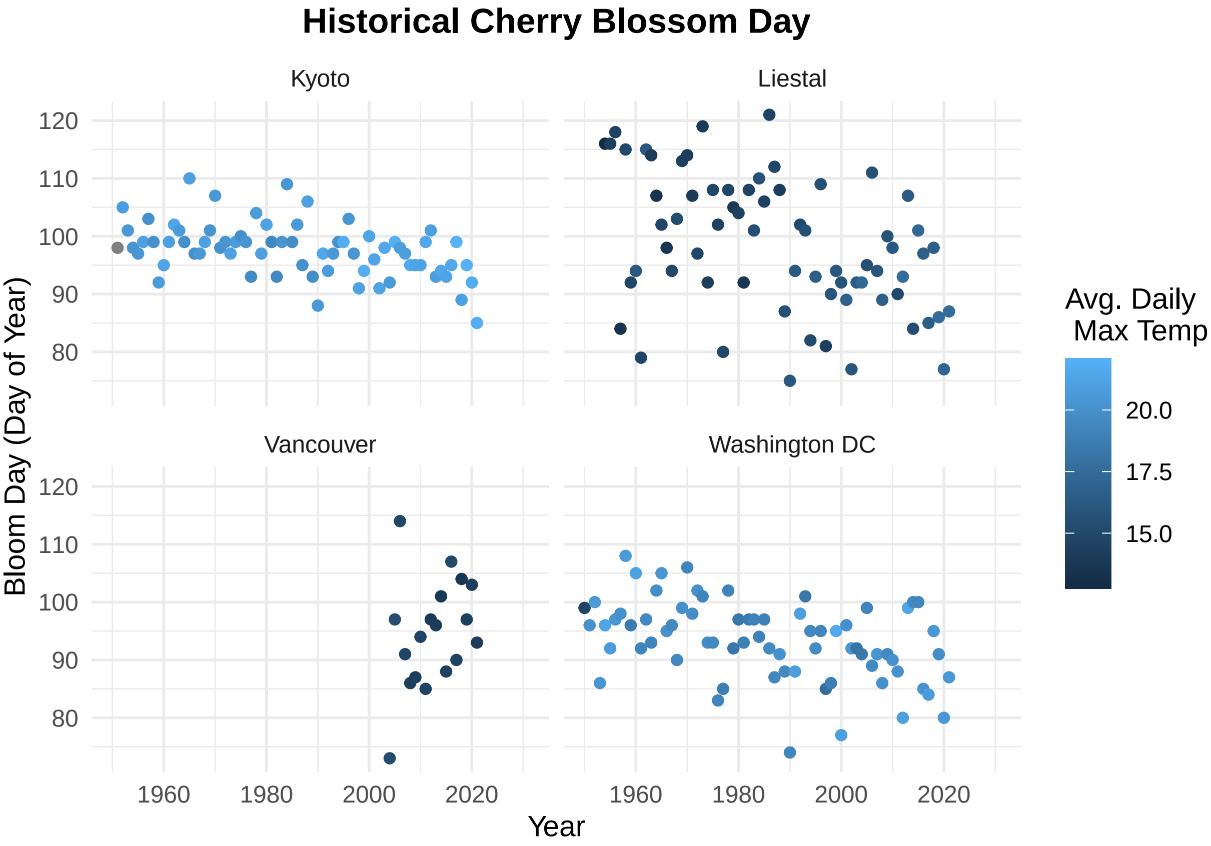

Figure 3: Historical Cherry Blossom Day colored with Previous Year's Average Daily Max Temperature

This following graph shows how the annual average daily max temperature has been changing especially in Liestal and Kyoto where it has been increasing steadily but in Vancouver saw decreases while Washington DC seems to be relatively stable compares to other locations.

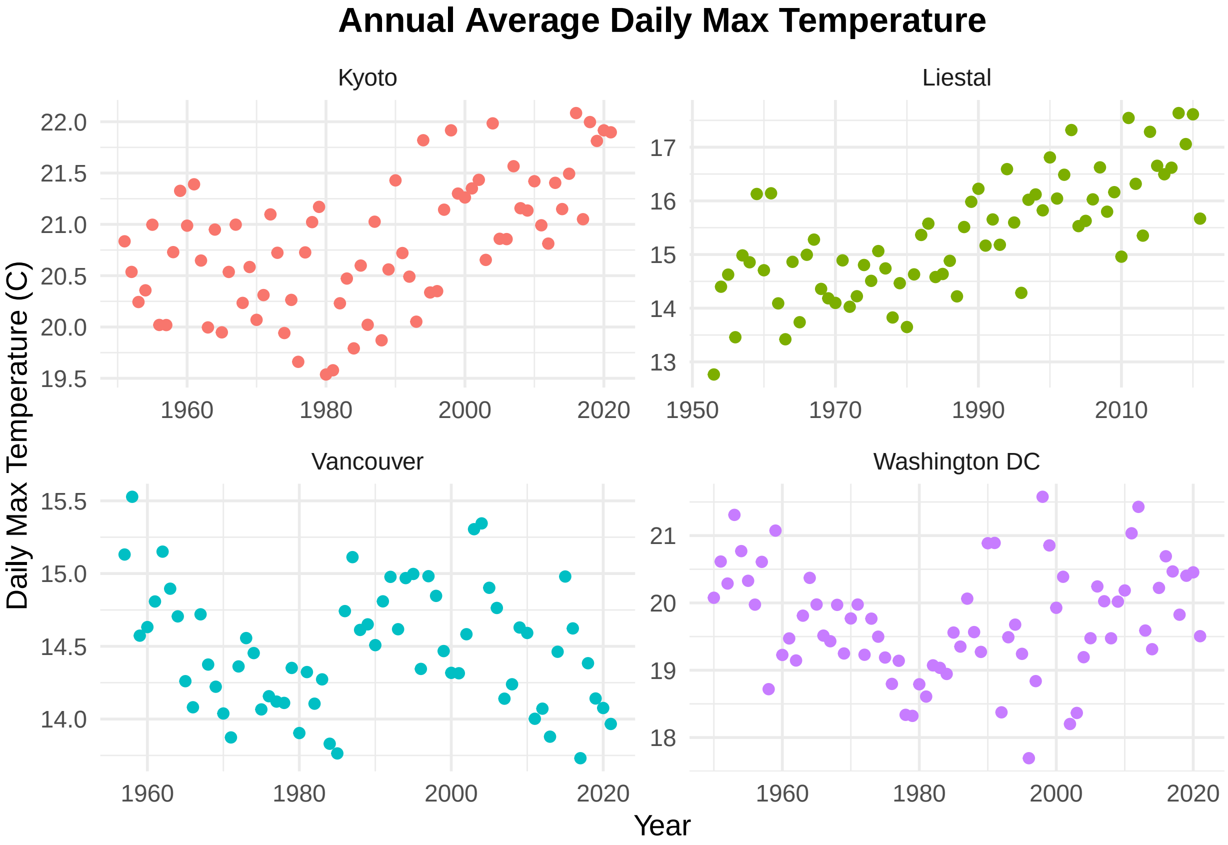

Figure 4: Annual Average Daily Max Temperature Trend

From the graph below we can see that the annual average daily max temperature from previous year has different level of effects on the cherry bloom day in each location, could be problem with the temperature data and also the average temperature is not the best indicator here for all locations.

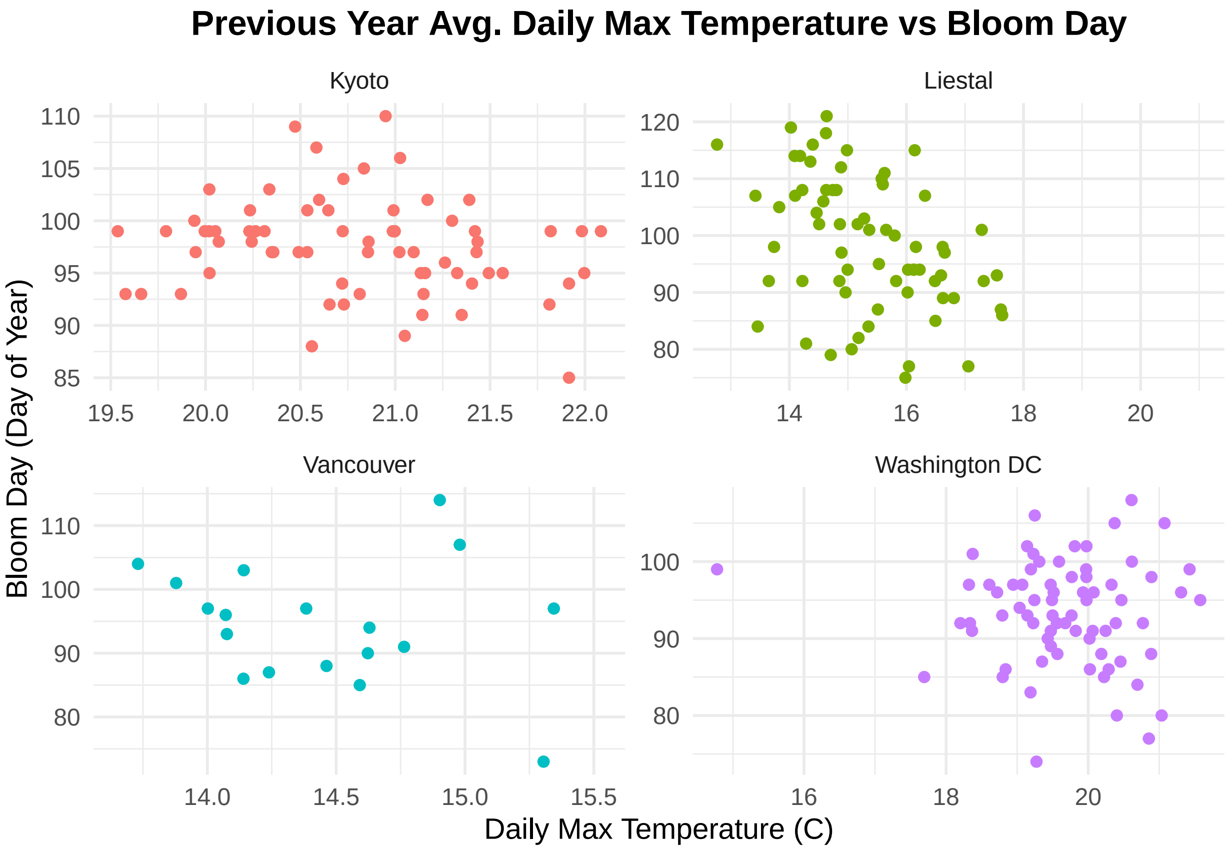

Figure 5: Historical Cherry Blossom Day against previous year's average daily max temperature

Basic Logistic Regression - Predicting probability of Cherry Blossom day

Basic Logistic Regression (greedy method) and constraint to only the first 120 days of the year using `location x year x doy (Day of Year)` interaction terms. The basic model will first assume there is no way Cherry will bloom in the second half of the year in all locations.

Predictions

Model Prediction Accuracy (Average Absolute Bloom Day Difference)

Table 4: Model prediction statistics

location	abs_doy_diff
Kyoto	2.9
Liestal	10.9
Vancouver	6.3
Washington DC	4.6

Visualize Model Prediction

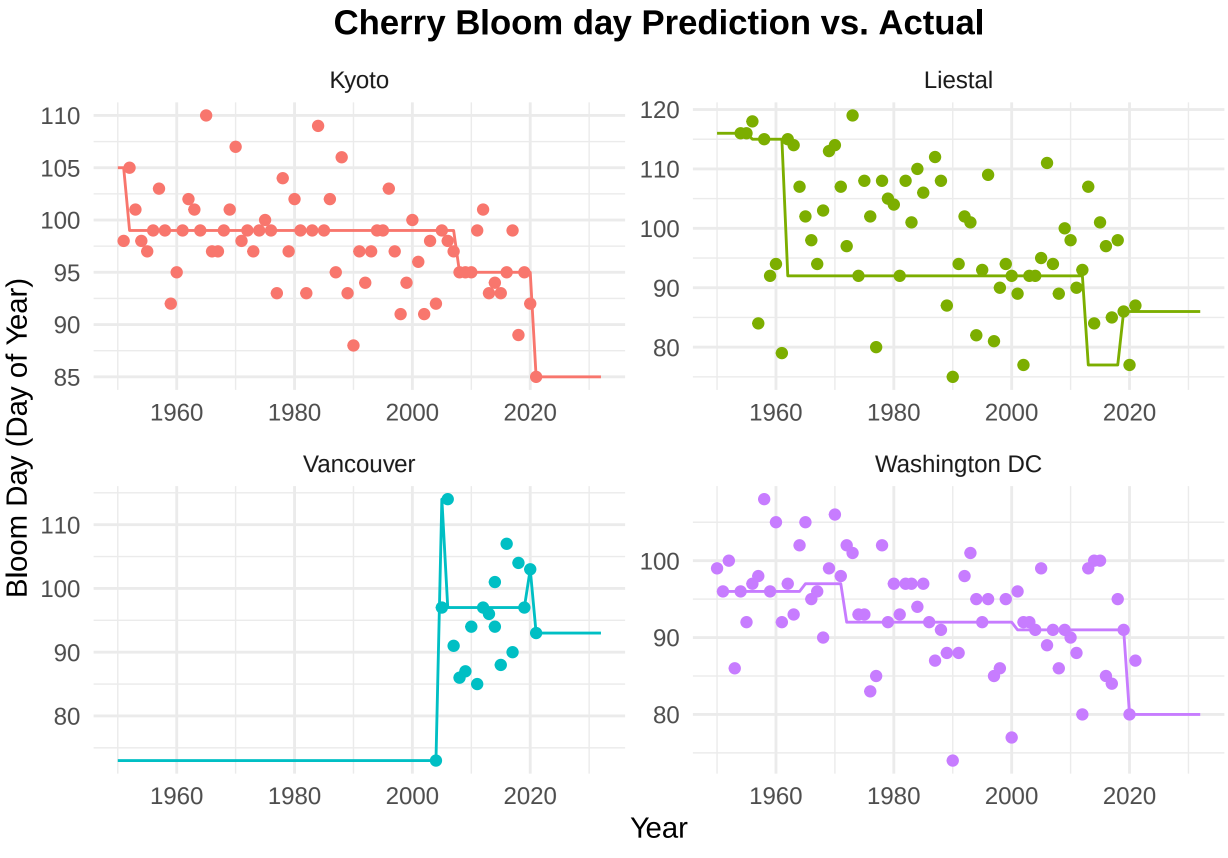

Figure 6: Logistic Regression Model Predictions

Robust logistic regression is probably not the best for this tasks as it fails to capture the different variances of the cherry bloom day in each year and it takes quite a lot of computation resource as expected since it has to calculate many interaction terms.

Seasonal Regression Model (Annual Trend)

The seasonal regression model tries to naively break down the annual cherry blossom date as a time series data and model through sine and cosine terms along with previous average daily max temperature from previous year. Also data used to model in each location will be varied but mostly data from 2000 will be used. As this naive approach work best for short term time series modelling as well as prediction.

Model process is wrapped into a function to make prediction in each locations:

```r
#' Wrapper for Seasonal Regression Model
#'
#' @param data -Input data frame (cherry_full)
#' @param model_year_from -Earliest data year used in model
#' @param loc -Location of the prediction
#' @param max_freq_level -Hyper-parameter to control number of Fourier Frequency terms to use
#'
#' @return
#' @export
#'
#' @examples
Seasonal_model <- function(data, model_year_from=0, loc, max_freq_level=0.5) {

  data_df <- data %>%
    filter(location == loc & year >= model_year_from & !is.na(bloom_doy)) %>%
    select(location,year,bloom_doy,avg_tmax_last1) %>%
    mutate(year_num = 1:n())

  ssp <- spectrum(data_df$bloom_doy, plot=FALSE)
  n_per <- floor(0.5*length(ssp$freq))
  per <- 1/ssp$freq[order(-ssp$spec)][1:n_per]
  sincos <-""
  for (i in 1:length(per)){
    sincos<-paste(sincos, "+sin(2*pi/",per[i],"*year_num)+cos(2*pi/",per[i],"*year_num)",sep="")
  }

  if (loc %in% c("Kyoto","Washington DC")){
    Linear <- lm(as.formula(paste0("bloom_doy~",sincos,"+year_num")),data=data_df)
  } else {
  Linear <- lm(as.formula(paste0("bloom_doy~",sincos,"+year_num+avg_tmax_last1")),data=data_df)
  }

  pred_df <- left_join(tibble(location = loc,
                         year = min(data_df$year):2031,
                         year_num=1:length(year)),
                       data %>%
                         filter(year>=min(data_df$year) & location==loc) %>%
                         select(location,year,avg_tmax_last1) %>% unique(),
                       by=c("location"="location","year"="year")) %>%
            bind_cols(pred_doy = round(predict(Linear, newdata = .),0))

  pred_df <- left_join(pred_df,data_df %>% filter(!is.na(bloom_doy)),by=c("location"="location","year"=
    select(-year_num.x,-avg_tmax_last1.y) %>%
    mutate(year_num=year_num.y,
           avg_tmax_last1=avg_tmax_last1.x)
  return(pred_df)
}
```

Combining the prediction results:

Table 5: Model prediction statistics

location	abs_doy_diff	avg_pred_doy	sd_pred_doy	avg_doy	sd_doy
Kyoto	0.7	93.3	4.5	95.0	3.8
Liestal	2.2	90.2	8.3	93.5	8.2
Vancouver	6.3	97.8	7.5	94.6	6.7
Washington DC	2.1	91.1	6.1	90.1	6.5

As we can see our prediction have the average absolute different from each location that is much less than the standard deviation of the actual blossom day. But this model technique is prone to over fitting as forcing sine and cosine terms in modelling time series using regression can over-fit data which leads to false sense of a good model. But the following graph is hope to shows that this naive model is able to capture the short term (as most of the changes happening in recent years) and made appropriate prediction of the future. The appropriate amount of sine and cosine term (with varies frequency term) were control and selected to act as provide some sort of variance in the data.

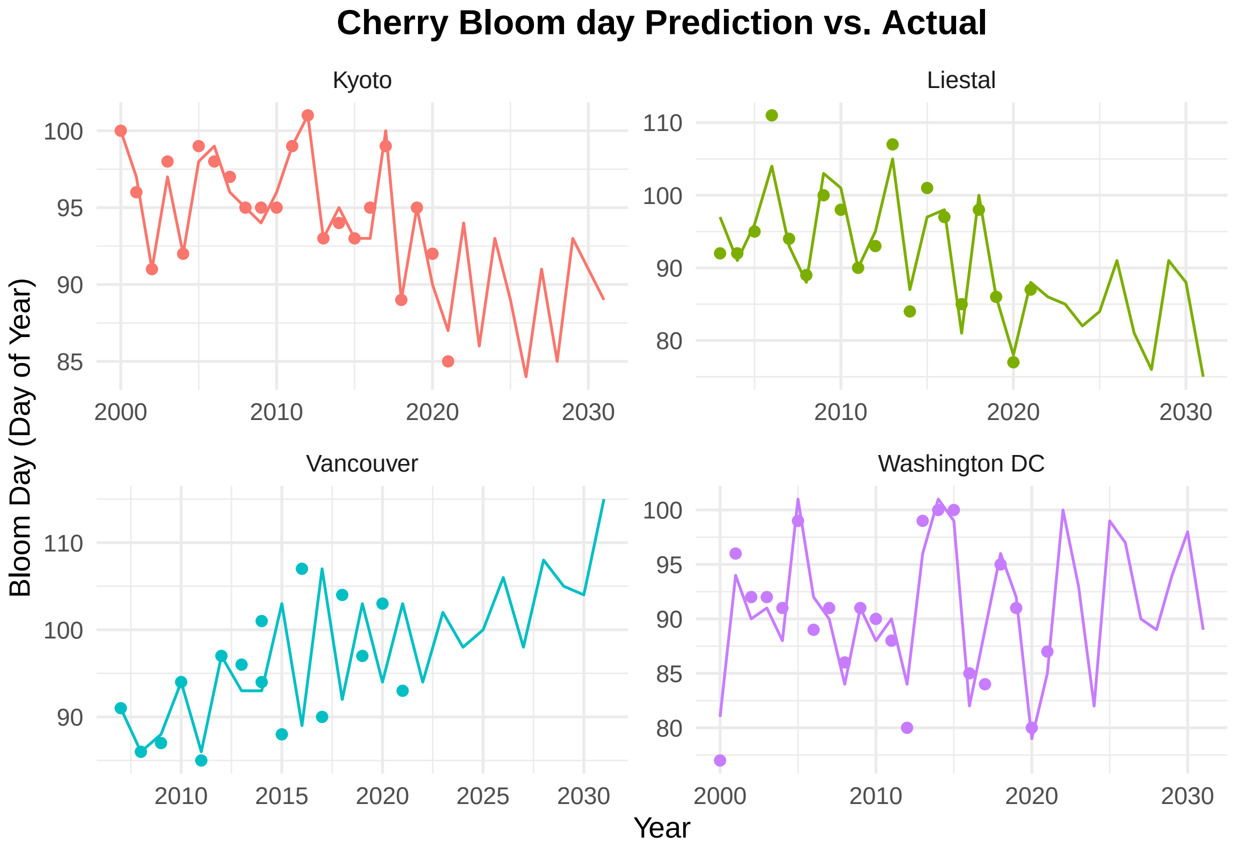

Figure 7: Seasonal Model Prediction

If we were to interpret this model and inference about the future of cherry blossom (highly advice not to), then we are concluding that the Cherry Bloom day in Kyoto and Liestal will see the bloom day to be push forward to earlier in the year while Vancouver will be push back to later in the year (also given the average daily max temperature has been colder in Vancouver) while Washington DC remains the same and slightly pushed back. Given the data used in this model building process and the exploratory data analysis, it is consistence to conclude that temperature does have some correlation with how the date of Cherry bloom is change in different location for different species of Cherry trees.

The amount of data is available for these model is only enough to make a rough prediction on where the future of the Cherry bloom day will be, but a big part of it does relying on how much we predict the future temperature in each location is going to be. Also the "seasonal" effect is artificial and does not conclude that the Cherry blossom day in previous year will have a direct impact in the next year's bloom day. The object of this process is to aim to predict the nearest possible Cherry Blossom day in each location using historical data and predicted annual average daily max temperature.

The final prediction from the year 2022 to 2031 using the seasonal model is given by this following:

Table 6: Cherry Blossom Predictions

year	kyoto	liestal	washingtondc	vancouver
2022	94	86	100	94
2023	86	85	93	102
2024	93	82	82	98
2025	89	84	99	100
2026	84	91	97	106
2027	91	81	90	98
2028	85	76	89	108
2029	93	91	94	105
2030	91	88	98	104
2031	89	75	89	115

Export to csv:

```
write.csv(cherry_export,"../data/Cherry_prediction.csv",row.names = FALSE)
```