Cherry Blossom Prediction Code

Christopher Miranda

2024-02-29

```
library(tidyverse)
```

Groundhog Day Results from 2000 to 2024

```
ghd <- data.frame(year = c(2000:2024),
                  groundhog = c(1,1,1,1,1,1,1,0,1,1,1,0,1,0,1,1,0,1,1,0,0,1,1,1,0))
```

All data wrangling, modelling, and predictions were done separately for each location. Similar processes were done for each location.

7 Variables:

Year, GDD, Average High Temp, Average Low Temp, Number of Birds Observed, Groundhog Day Results, Bloom DOY

Variables for Washington D.C.

Load Daily Temperature Data from NOAA for Washington D.C. From 12/1/1999 to 2/24/2024. Given Highs and Lows were used to calculate Averages and GDDs. Data were grouped and filtered to calculate GDD accumulation, average high, and average low for all winters since 2000.

```
dc_temp <- read.csv('temps/dc_temp.csv') %>%
  mutate(TAVG = ceiling((TMAX+TMIN)/2),
         GDD = if_else(TAVG<45, 0, TAVG-45)) %>%
  group_by(year(DATE), month(DATE, label=T)) %>%
  rename(year = 'year(DATE)',
         month = 'month(DATE, label = T)') %>%
  filter(month == 'Dec' | month == 'Jan' | month == 'Feb') %>%
  summarise(GDD_sum = sum(GDD),
            HI_AVG = ceiling(mean(TMAX)),
            LO_AVG = ceiling(mean(TMIN))) %>%
  group_by(year = ceiling(row_number()/3)) %>%
  summarize(GDD_sum = sum(GDD_sum),
            HI_AVG = ceiling(mean(HI_AVG)),
            LO_AVG = ceiling(mean(LO_AVG))) %>%
  mutate(year = year+1999)
```

Bird sighting data from eBird for Washington D.C. were used to calculate total counts of birds for all winters. Observations recorded as 'X' were changed to equal the mean count of that month.

```
dc_birds <- read.table('eBird/US_DC.txt', sep='\t', header=T, fill=T, quote="") %>%
  select(STATE.CODE, OBSERVATION.DATE, COMMON.NAME, OBSERVATION.COUNT) %>%
  mutate(OBSERVATION.DATE = as_date(OBSERVATION.DATE)) %>%
  mutate(OBSERVATION.COUNT = if_else(OBSERVATION.COUNT=='X', '1.23', OBSERVATION.COUNT),
         OBSERVATION.COUNT = as.integer(OBSERVATION.COUNT)) %>%
  group_by(year(OBSERVATION.DATE), month(OBSERVATION.DATE, label=T)) %>%
  mutate(OBSERVATION.COUNT = if_else(OBSERVATION.COUNT==1.23, mean(OBSERVATION.COUNT), OBSERVATION.COUN
  rename(year = 'year(OBSERVATION.DATE)',
         month = 'month(OBSERVATION.DATE, label = T)') %>%
  summarize(counts = sum(OBSERVATION.COUNT)) %>%
  filter(month == 'Dec' | month == 'Jan' | month == 'Feb') %>%
  group_by(year = ceiling(row_number()/3)) %>%
  summarize(bird_counts = sum(counts)) %>%
  mutate(year = year+1999)
```

Load and filter bloom dates given by GitHub repo

```
dc_blooms <- read.csv('blooms/washingtondc.csv') %>%
  filter(year >= 2000) %>%
  select(year, bloom_doy)
```

Assemble all variables for DC to make predictions later

```
model_dc <- dc_temp %>%
  right_join(dc_birds) %>%
  right_join(ghd) %>%
  right_join(dc_blooms)
```

Variables for Liestal

Load Daily Temperature Data from NOAA for Liestal Given Highs and Lows were used to calculate Averages and GDDs. Data were grouped and filtered to calculate GDD accumulation, average high, and average low for all winters since 2000. NA values replaced with mean values belonging to that month.

```
liestal_temp <- read.csv('temps/liestal_temp.csv') %>%
  group_by(year(DATE), month(DATE, label=T)) %>%
  rename(year = 'year(DATE)',
         month = 'month(DATE, label = T)') %>%
  filter(month == 'Dec' | month == 'Jan' | month == 'Feb') %>%
  mutate(TMAX = if_else(is.na(TMAX), mean(TMAX, na.rm = T), TMAX),
         TMIN = if_else(is.na(TMIN), mean(TMIN, na.rm = T), TMIN),
         TAVG = ceiling((TMAX+TMIN)/2),
         GDD = if_else(TAVG<45, 0, TAVG-45)) %>%
  summarise(GDD_sum = sum(GDD),
            HI_AVG = ceiling(mean(TMAX)),
            LO_AVG = ceiling(mean(TMIN))) %>%
  group_by(year = ceiling(row_number()/3)) %>%
  summarize(GDD_sum = sum(GDD_sum),
            HI_AVG = ceiling(mean(HI_AVG)),
            LO_AVG = ceiling(mean(LO_AVG))) %>%
  mutate(year = year+1999)
```

Bird sighting data from eBird for Liestal were used to calculate total counts of birds for all winters. Observations recorded as 'X' were changed to equal the mean count of that month. Data only available until 2013. Counts for remaining years determined by selecting random observation: filled downwards if still NA.

```
liestal_birds <- read.table('eBird/CH_BL.txt', sep='\t', header=T, fill=T, quote="") %>%
  select(STATE, OBSERVATION.DATE, COMMON.NAME, OBSERVATION.COUNT) %>%
  mutate(OBSERVATION.DATE = as_date(OBSERVATION.DATE)) %>%
  mutate(OBSERVATION.COUNT = if_else(OBSERVATION.COUNT=='X', '1.23', OBSERVATION.COUNT),
         OBSERVATION.COUNT = as.integer(OBSERVATION.COUNT)) %>%
  group_by(year(OBSERVATION.DATE), month(OBSERVATION.DATE, label=T)) %>%
  mutate(OBSERVATION.COUNT = if_else(OBSERVATION.COUNT==1.23, mean(OBSERVATION.COUNT), OBSERVATION.COUN
  rename(year = 'year(OBSERVATION.DATE)',
         month = 'month(OBSERVATION.DATE, label = T)') %>%
  summarize(counts = sum(OBSERVATION.COUNT)) %>%
  filter(month == 'Dec' | month == 'Jan' | month == 'Feb') %>%
  group_by(year = ceiling(row_number()/3)) %>%
  summarize(bird_counts = sum(counts)) %>%
  mutate(year = year+1999) %>%
  add_row(year = 2014:2024) %>%
  mutate(bird_counts = if_else(is.na(bird_counts), sample(bird_counts), bird_counts)) %>%
  fill(bird_counts)
```

Load and filter bloom dates given by GitHub repo

```
liestal_blooms <- read.csv('blooms/liestal.csv') %>%
  filter(year >= 2000) %>%
  select(year, bloom_doy)
```

Variables assembled for Liestal

```r
model_liestal <- liestal_temp %>%
  right_join(liestal_birds) %>%
  right_join(ghd) %>%
  right_join(liestal_blooms)
```

Variables for Kyoto

Load Daily Temperature Data from NOAA for Kyoto. Given Highs and Lows were used to calculate Averages and GDDs. Data were grouped and filtered to calculate GDD accumulation, average high, and average low for all winters since 2000. NA values replaced with mean values belonging to that month. Data missing for 2005, so rows were manually inputted with observations being filled with data from previous year.

```
kyoto_temp <- read.csv('temps/kyoto_temp.csv') %>%
  group_by(year(DATE), month(DATE, label=T)) %>%
  rename(year = 'year(DATE)',
         month = 'month(DATE, label = T)') %>%
  filter(month == 'Dec' | month == 'Jan' | month == 'Feb') %>%
  mutate(TMAX = if_else(is.na(TMAX), mean(TMAX, na.rm = T), TMAX),
         TMIN = if_else(is.na(TMIN), mean(TMIN, na.rm = T), TMIN),
         GDD = if_else(TAVG<45, 0, TAVG-45)) %>%
  summarise(GDD_sum = sum(GDD),
            HI_AVG = ceiling(mean(TMAX)),
            LO_AVG = ceiling(mean(TMIN))) %>%
  ungroup() %>%
  add_row(year=rep(c(2005),3), month=c('Jan','Feb','Dec'), .before=17) %>%
  fill(GDD_sum, HI_AVG, LO_AVG) %>%
  group_by(year = ceiling(row_number()/3)) %>%
  summarize(GDD_sum = sum(GDD_sum),
            HI_AVG = ceiling(mean(HI_AVG)),
            LO_AVG = ceiling(mean(LO_AVG))) %>%
  mutate(year = year+1999)
```

Bird sighting data from eBird for Kyoto were used to calculate total counts of birds for all winters. Observations recorded as 'X' were changed to equal the mean count of that month. Data only available until 2017. Counts for remaining years determined by selecting random observation: filled downwards if still NA.

```
kyoto_birds <- read.table('eBird/JP.txt', sep='\t', header=T, fill=T, quote="") %>%
  filter(COUNTY == 'Kyoto') %>%
  select(COUNTY, OBSERVATION.DATE, COMMON.NAME, OBSERVATION.COUNT) %>%
  mutate(OBSERVATION.DATE = as_date(OBSERVATION.DATE)) %>%
  mutate(OBSERVATION.COUNT = if_else(OBSERVATION.COUNT=='X', '1.23', OBSERVATION.COUNT),
         OBSERVATION.COUNT = as.integer(OBSERVATION.COUNT)) %>%
  group_by(year(OBSERVATION.DATE), month(OBSERVATION.DATE, label=T)) %>%
  mutate(OBSERVATION.COUNT = if_else(OBSERVATION.COUNT==1.23, mean(OBSERVATION.COUNT), OBSERVATION.COUN
  rename(year = 'year(OBSERVATION.DATE)',
         month = 'month(OBSERVATION.DATE, label = T)') %>%
  summarize(counts = sum(OBSERVATION.COUNT)) %>%
  filter(month == 'Dec' | month == 'Jan' | month == 'Feb') %>%
  group_by(year = ceiling(row_number()/3)) %>%
  summarize(bird_counts = sum(counts)) %>%
  mutate(year = year+1999) %>%
  add_row(year = 2018:2024) %>%
  mutate(bird_counts = if_else(is.na(bird_counts), sample(bird_counts), bird_counts)) %>%
  fill(bird_counts)
```

Load and filter bloom dates given by GitHub repo

```
kyoto_blooms <- read.csv('blooms/kyoto.csv') %>%
  filter(year >= 2000) %>%
  select(year, bloom_doy)
```

Variables assembled for Kyoto

```
model_kyoto <- kyoto_temp %>%
  right_join(kyoto_birds) %>%
  right_join(ghd) %>%
  right_join(kyoto_blooms)
```

vancouver

Load Daily Temperature Data from NOAA for Vancouver. Given Highs and Lows were used to calculate Averages and GDDs. Data were grouped and filtered to calculate GDD accumulation, average high, and average low for all winters since 2000. NA values replaced with mean values belonging to that month.

```
vancouver_temp <- read.csv('temps/vancouver_temp.csv') %>%
  group_by(year(DATE), month(DATE, label=T)) %>%
  rename(year = 'year(DATE)',
         month = 'month(DATE, label = T)') %>%
  filter(month == 'Dec' | month == 'Jan' | month == 'Feb') %>%
  mutate(TMAX = if_else(is.na(TMAX), mean(TMAX, na.rm = T), TMAX),
         TMIN = if_else(is.na(TMIN), mean(TMIN, na.rm = T), TMIN),
         TAVG = ceiling((TMAX+TMIN)/2),
         GDD = if_else(TAVG<45, 0, TAVG-45)) %>%
  summarise(GDD_sum = sum(GDD),
            HI_AVG = ceiling(mean(TMAX)),
            LO_AVG = ceiling(mean(TMIN))) %>%
  group_by(year = ceiling(row_number()/3)) %>%
  summarize(GDD_sum = sum(GDD_sum),
            HI_AVG = ceiling(mean(HI_AVG)),
            LO_AVG = ceiling(mean(LO_AVG))) %>%
  mutate(year = year+1999)
```

Bird sighting data from eBird for Vancouver were used to calculate total counts of birds for all winters. Separate objects for reading data and filtering due to extremely large dataset. Observations recorded as 'X' were changed to equal the mean count of that month

```
vancouver_birds <- read.table('eBird/CA_BC.txt', sep='\t', header=T, fill=T, quote="")

vancouver_birds <- vancouver_birds %>%
  select(COUNTY, OBSERVATION.DATE, COMMON.NAME, OBSERVATION.COUNT) %>%
  filter(COUNTY == 'Metro Vancouver') %>%
  mutate(OBSERVATION.DATE = as_date(OBSERVATION.DATE)) %>%
  mutate(OBSERVATION.COUNT = if_else(OBSERVATION.COUNT=='X', '1.23', OBSERVATION.COUNT),
         OBSERVATION.COUNT = as.integer(OBSERVATION.COUNT)) %>%
  group_by(year(OBSERVATION.DATE), month(OBSERVATION.DATE, label=T)) %>%
  mutate(OBSERVATION.COUNT = if_else(OBSERVATION.COUNT==1.23, mean(OBSERVATION.COUNT), OBSERVATION.COUN
  rename(year = 'year(OBSERVATION.DATE)',
         month = 'month(OBSERVATION.DATE, label = T)') %>%
  summarize(counts = sum(OBSERVATION.COUNT)) %>%
  filter(month == 'Dec' | month == 'Jan' | month == 'Feb') %>%
  group_by(year = ceiling(row_number()/3)) %>%
  summarize(bird_counts = sum(counts)) %>%
  mutate(year = year+1999)
```

'bloom_avg' created to fill missing bloom data before 2022. Binds blooms dates from previous 3 cities and takes the average. Averages are binded to Vancouver dates given by Github

```
bloom_avg <- dc_blooms %>%
  bind_rows(kyoto_blooms,liestal_blooms) %>%
  filter(year<2022) %>%
  group_by(year) %>%
```

```
  summarize(bloom_doy = ceiling(mean(bloom_doy))) %>%
  arrange(desc(year))

vancouver_blooms <- read.csv('blooms/vancouver.csv') %>%
  select(year, bloom_doy)
vancouver_blooms <- vancouver_blooms %>%
  bind_rows(bloom_avg)
```

Assemble variables for Vancouver

```
model_vancouver <- vancouver_temp %>%
  right_join(vancouver_birds) %>%
  right_join(ghd) %>%
  right_join(vancouver_blooms)
```

newyorkcity ##

Load Daily Temperature Data from NOAA for NYC. Given Highs and Lows were used to calculate Averages and GDDs. Data were grouped and filtered to calculate GDD accumulation, average high, and average low for all winters since 2000.

```
nyc_temp <- read.csv('temps/nyc_temp.csv') %>%
  mutate(TAVG = ceiling((TMAX+TMIN)/2),
         GDD = if_else(TAVG<45, 0, TAVG-45)) %>%
  group_by(year(DATE), month(DATE, label=T)) %>%
  rename(year = 'year(DATE)',
         month = 'month(DATE, label = T)') %>%
  filter(month == 'Dec' | month == 'Jan' | month == 'Feb') %>%
  summarise(GDD_sum = sum(GDD),
            HI_AVG = ceiling(mean(TMAX)),
            LO_AVG = ceiling(mean(TMIN))) %>%
  group_by(year = ceiling(row_number()/3)) %>%
  summarize(GDD_sum = sum(GDD_sum),
            HI_AVG = ceiling(mean(HI_AVG)),
            LO_AVG = ceiling(mean(LO_AVG))) %>%
  mutate(year = year+1999)
```

Bird sighting data from eBird for NYC were used to calculate total counts of birds for all winters. Observations recorded as 'X' were changed to equal the mean count of that month.

```
nyc_birds <- read.table('eBird/US_NY.txt', sep='\t', header=T, fill=T, quote="") %>%
  select(COUNTY, OBSERVATION.DATE, COMMON.NAME, OBSERVATION.COUNT) %>%
  mutate(OBSERVATION.DATE = as_date(OBSERVATION.DATE)) %>%
  mutate(OBSERVATION.COUNT = if_else(OBSERVATION.COUNT=='X', '1.23', OBSERVATION.COUNT),
         OBSERVATION.COUNT = as.integer(OBSERVATION.COUNT)) %>%
  group_by(year(OBSERVATION.DATE), month(OBSERVATION.DATE, label=T)) %>%
  mutate(OBSERVATION.COUNT = if_else(OBSERVATION.COUNT==1.23, mean(OBSERVATION.COUNT), OBSERVATION.COUN
  rename(year = 'year(OBSERVATION.DATE)',
         month = 'month(OBSERVATION.DATE, label = T)') %>%
  summarize(counts = sum(OBSERVATION.COUNT)) %>%
  filter(month == 'Dec' | month == 'Jan' | month == 'Feb') %>%
  group_by(year = ceiling(row_number()/3)) %>%
  summarize(bird_counts = sum(counts)) %>%
  mutate(year = year+1999)
```

Blooms dates since 2012 obtained from The New York Botanical Garden (object 'nybg'). Filtered by any plant with 'Japanese flowering cherry' in name.

Key to Reproductive Status:

I = buds

F = open flowers

O = old flowers

D = immature fruit

R = mature fruit

Letter combinations indicate concurrent phenotypes

Filtered by open flowers to specify blooms. Convert dates to day_of_year and take earliest bloom for each year.

```
nybg <- readxl::read_xlsx('blooms/ny_garden_bloom.xlsx') %>%
  select(COMMON_NAME, REPRODUCTIVE_STATUS, CHECK_DT) %>%
  filter(grepl('Japanese flowering cherry', COMMON_NAME),
         REPRODUCTIVE_STATUS == 'F') %>%
  group_by(year(CHECK_DT)) %>%
  rename(year = 'year(CHECK_DT)') %>%
  mutate(CHECK_DT = as_date(CHECK_DT),
         CHECK_DT = cropgrowdays::day_of_year(CHECK_DT)) %>%
  summarize(bloom_doy = min(CHECK_DT)) %>%
  arrange(desc(year))
```

Slice 'bloom_avg' and attach to fill appropriate missing data

```
nyc_blooms <- bloom_avg %>%
  slice(11:22) %>%
  bind_rows(nybg) %>%
  arrange(desc(year))
```

Assemble variables for NYC

```
model_nyc <- nyc_temp %>%
  right_join(nyc_birds) %>%
  right_join(ghd) %>%
  right_join(nyc_blooms)
```

Random Forest and Final Predictions

```
library(caTools)
library(randomForest)
```

Use 500 trees with 80% split ratio

washingtondc

```
set.seed(333)
split_dc <- sample.split(model_dc$bloom_doy, SplitRatio = 0.8)
train_dc <- subset(model_dc, split_dc==T)
test_dc <- subset(model_dc, split_dc==F)

class_dc <- randomForest(x=train_dc[-7],
                         y=train_dc$bloom_doy,
                         ntree=500, random_state=0)

pred_dc <- predict(class_dc, newdata = test_dc[-7])

dc_2024 <- dc_temp %>%
  right_join(dc_birds) %>%
  right_join(ghd) %>%
  filter(year==2024)

cherry_dc <- expand_grid(location = 'washingtondc',
                         year = 2024) %>%
  bind_cols(predict(class_dc, newdata = dc_2024)) %>%
  rename(prediction = '...3') %>%
  mutate(prediction = ceiling(prediction))
```

liestal

```
set.seed(333)
split_liestal <- sample.split(model_liestal$bloom_doy, SplitRatio = 0.8)
train_liestal <- subset(model_liestal, split_liestal==T)
test_liestal <- subset(model_liestal, split_liestal==F)

class_liestal <- randomForest(x=train_liestal[-7],
                              y=train_liestal$bloom_doy,
                              ntree=500, random_state=0)

pred_liestal <- predict(class_liestal, newdata = test_liestal[-7])

liestal_2024 <- liestal_temp %>%
  right_join(liestal_birds) %>%
  right_join(ghd) %>%
  filter(year==2024)

cherry_liestal <- expand_grid(location = 'liestal',
                              year = 2024) %>%
  bind_cols(predict(class_liestal, newdata = liestal_2024)) %>%
  rename(prediction = '...3') %>%
  mutate(prediction = ceiling(prediction))
```

kyoto

```
set.seed(333)
split_kyoto <- sample.split(model_kyoto$bloom_doy, SplitRatio = 0.8)
train_kyoto <- subset(model_kyoto, split_kyoto==T)
test_kyoto <- subset(model_kyoto, split_kyoto==F)

class_kyoto <- randomForest(x=train_kyoto[-7],
                            y=train_kyoto$bloom_doy,
                            ntree=500, random_state=0)

pred_kyoto <- predict(class_kyoto, newdata = test_kyoto[-7])

kyoto_2024 <- kyoto_temp %>%
  right_join(kyoto_birds) %>%
  right_join(ghd) %>%
  filter(year==2024)

cherry_kyoto <- expand_grid(location = 'kyoto',
                            year = 2024) %>%
  bind_cols(predict(class_kyoto, newdata = kyoto_2024)) %>%
  rename(prediction = '...3') %>%
  mutate(prediction = ceiling(prediction))
```

vancouver

```
set.seed(333)
split_vancouver <- sample.split(model_vancouver$bloom_doy, SplitRatio = 0.8)
train_vancouver <- subset(model_vancouver, split_vancouver==T)
test_vancouver <- subset(model_vancouver, split_vancouver==F)

class_vancouver <- randomForest(x=train_vancouver[-7],
                                y=train_vancouver$bloom_doy,
                                ntree=500, random_state=0)

pred_vancouver <- predict(class_vancouver, newdata = test_vancouver[-7])

vancouver_2024 <- vancouver_temp %>%
  right_join(vancouver_birds) %>%
  right_join(ghd) %>%
  filter(year==2024)

cherry_vancouver <- expand_grid(location = 'vancouver',
                                year = 2024) %>%
  bind_cols(predict(class_vancouver, newdata = vancouver_2024)) %>%
  rename(prediction = '...3') %>%
  mutate(prediction = ceiling(prediction))
```

newyorkcity

```
set.seed(333)
split_nyc <- sample.split(model_nyc$bloom_doy, SplitRatio = 0.8)
train_nyc <- subset(model_nyc, split_nyc==T)
```

```
test_nyc <- subset(model_nyc, split_nyc==F)

class_nyc <- randomForest(x=train_nyc[-7],
                          y=train_nyc$bloom_doy,
                          ntree=500, random_state=0)

pred_nyc <- predict(class_nyc, newdata = test_nyc[-7])

nyc_2024 <- nyc_temp %>%
  right_join(nyc_birds) %>%
  right_join(ghd) %>%
  filter(year==2024)

cherry_nyc <- expand_grid(location = 'newyorkcity',
                          year = 2024) %>%
  bind_cols(predict(class_nyc, newdata = nyc_2024)) %>%
  rename(prediction = '...3') %>%
  mutate(prediction = ceiling(prediction))
```

Final Submission (bigcherry)

```
bigcherry <- cherry_dc %>%
  bind_rows(cherry_liestal,
            cherry_kyoto,
            cherry_vancouver,
            cherry_nyc) %>%
  select(-year)

bigcherry
```

```
## # A tibble: 5 x 2
##   location     prediction
##   <chr>             <dbl>
## 1 washingtondc         85
## 2 liestal              87
## 3 kyoto                91
## 4 vancouver            91
## 5 newyorkcity          93
```

```
write.csv(bigcherry, 'bigcherry.csv', row.names = F)
```