

# Cherry Blossom Prediction

## Exploratory Data Analysis (EDA)

Alex Yinan Guo

## Contents

<b>Setup</b>	<b>1</b>
<b>Data Summaries</b>	<b>2</b>
Before Imputation . . . . .	2
Data After Imputation . . . . .	3
<b>Density Plots</b>	<b>4</b>
Overall Density Plots . . . . .	4
Density Plots of Japan . . . . .	5
Density Plots of South Korea . . . . .	6
Density Plots of Switzerland . . . . .	7
Density Plots of United States . . . . .	8
<b>Year v.s. Numeric Variables</b>	<b>9</b>
Year v.s. Numeric Variables of Japan . . . . .	9
Year v.s. Numeric Variables of South Korea . . . . .	10
Year v.s. Numeric Variables of Switzerland . . . . .	11
Year v.s. Numeric Variables of United States . . . . .	12
<b>Visualization of Data Locations</b>	<b>13</b>
<b>Basic Linear Models</b>	<b>15</b>
Fit Linear Regression . . . . .	16
Confidence Intervals . . . . .	22
Prediction Intervals . . . . .	22
Check multicollinearity . . . . .	23
Mix Effects Model . . . . .	26

## Setup

Load packages and read in data sets.

```
library(GGally)
library(rjson)
library(tidyverse)
library(lubridate)
library(cowplot)
library(ggpubr)
library(knitr)
library(mice)
library(mctest)
```

```

library(lme4)
library(quantreg)
library(AER)
library(MASS)
library(dplyr)
library(reshape2)

df <- read.csv(file = '../data/processed/clean_data.csv')
df_non_impute <- read.csv(file = '../data/processed/processed_data.csv')

```

## Data Summaries

### Before Imputation

```
summary(df_non_impute)
```

```

##      id          location        country        city
##  Length:7028    Length:7028    Length:7028    Length:7028
##  Class :character  Class :character  Class :character  Class :character
##  Mode  :character  Mode  :character  Mode  :character  Mode  :character
##
##      lat          long          alt          year
##  Min.   :24.34   Min.   :-77.04   Min.   : 0.00   Min.   : 812
##  1st Qu.:34.69   1st Qu.:131.62   1st Qu.: 6.30   1st Qu.:1964
##  Median :35.44   Median :135.68   Median : 30.00   Median :1983
##  Mean   :36.73   Mean   :125.80   Mean   : 75.46   Mean   :1935
##  3rd Qu.:38.89   3rd Qu.:139.75   3rd Qu.: 44.00   3rd Qu.:1999
##  Max.   :47.57   Max.   :145.59   Max.   :1390.00  Max.   :2021
##
##      bloom_date      bloom_doy       tmax       tmin
##  Length:7028      Min.   : 8.0   Min.   :-1.231  Min.   :-6.892
##  Class :character  1st Qu.:94.0   1st Qu.:15.181  1st Qu.: 6.601
##  Mode  :character  Median :100.0   Median :19.071  Median :10.235
##                  Mean   :101.3   Mean   :17.957  Mean   : 9.665
##                  3rd Qu.:110.0   3rd Qu.:20.593  3rd Qu.:12.037
##                  Max.   :160.0   Max.   :32.300  Max.   :26.500
##                  NA's   :1390    NA's   :1375   NA's   :1375
##
##      prcp      agdd_winter      tmax_winter      tmin_winter
##  Min.   : 0.300   Min.   : 0.0   Min.   :-5.361  Min.   :-19.2316
##  1st Qu.: 3.403   1st Qu.:110.7   1st Qu.: 4.226  1st Qu.: -3.1965
##  Median : 4.649   Median :280.3   Median : 8.774  Median :  0.3411
##  Mean   : 6.210   Mean   :320.8   Mean   : 7.800  Mean   : -0.2942
##  3rd Qu.: 6.927   3rd Qu.:462.9   3rd Qu.:10.871  3rd Qu.:  2.2145
##  Max.   :122.900   Max.   :1713.3   Max.   :24.032  Max.   : 18.4540
##  NA's   :1376    NA's   :1486    NA's   :1494   NA's   :1494
##
##      prcp_winter      co2_percapita      co2_emission
##  Min.   : 0.00   Min.   :0.0003   Min.   :1.099e+04
##  1st Qu.: 16.30  1st Qu.:5.5023   1st Qu.:3.589e+08
##  Median : 26.37  Median :7.9740   Median :9.143e+08
##  Mean   : 36.59  Mean   :7.3841   Mean   :8.547e+08

```

```

## 3rd Qu.: 47.28   3rd Qu.: 9.5871   3rd Qu.:1.223e+09
## Max.    :525.25   Max.    :22.2365   Max.    :6.135e+09
## NA's     :1494      NA's    :749       NA's    :749

```

```
apply(df_non_impute, 2, function(x) length(unique(x)))
```

```

##      id      location      country      city      lat
## 109      105          4        105      112
## long      alt      year bloom_date bloom_doy
## 110      106         837       3359      127
## tmax      tmin      prcp agdd_winter tmax_winter
## 5053     5080       4825       4129      5028
## tmin_winter prcp_winter co2_percapita co2_emission
## 5034     4965       403        402

```

## Data After Imputation

```
head(df)
```

```

##   country      city year bloom_doy      lat      long alt      tmax      tmin
## 1 Japan Ishigakijima 1962      51 24.33667 124.1644 5.7 23.80000 21.05040
## 2 Japan Ishigakijima 1963      46 24.33667 124.1644 5.7 23.80000 21.05040
## 3 Japan Ishigakijima 1967      44 24.33667 124.1644 5.7 23.80000 21.05040
## 4 Japan Ishigakijima 1971      41 24.33667 124.1644 5.7 23.80000 21.05040
## 5 Japan Ishigakijima 1972      51 24.33667 124.1644 5.7 23.80000 21.05040
## 6 Japan Ishigakijima 1973      37 24.33667 124.1644 5.7 26.78116 21.81633
##      prcp agdd_winter tmax_winter tmin_winter prcp_winter co2_percapita
## 1 21.97045      22.35    17.80000    15.01875    98.40000      3.0680
## 2 21.97045      22.35    17.80000    15.01875    98.40000      3.3700
## 3 21.97045      22.35    17.80000    15.01875    98.40000      4.8532
## 4 21.97045      22.35    17.80000    15.01875    98.40000      7.4844
## 5 21.97045      22.35    17.80000    15.01875    98.40000      7.8922
## 6 6.74517      649.50   22.93487    17.36111    48.49867      8.3473

```

```
tail(df)
```

```

##      country      city year bloom_doy      lat      long alt      tmax      tmin
## 6177 Switzerland Liestal 2016      97 47.4814 7.730519 350 17.77065 7.578804
## 6178 Switzerland Liestal 2017      85 47.4814 7.730519 350 16.61808 6.177260
## 6179 Switzerland Liestal 2018      98 47.4814 7.730519 350 17.63562 7.103836
## 6180 Switzerland Liestal 2019      86 47.4814 7.730519 350 17.05753 6.315890
## 6181 Switzerland Liestal 2020      77 47.4814 7.730519 350 17.61284 6.630055
## 6182 Switzerland Liestal 2021      87 47.4814 7.730519 350 15.70822 5.805753
##      prcp agdd_winter tmax_winter tmin_winter prcp_winter co2_percapita
## 6177 1.685870      141.35    6.819355    1.5451613    23.83871      4.677000
## 6178 2.740822      235.55    5.344240   -1.6786098    15.42166      4.515500
## 6179 2.196438      337.85    6.214555    0.5637097    27.30415      4.324300
## 6180 2.327945      374.60    7.906413    0.2789939    20.28763      4.276600
## 6181 2.223224      475.65    9.360289    1.1466073    21.00222      3.731900
## 6182 3.298630      372.05    7.178418    0.4391705    33.85637      2.381433

```

```
summary(df)
```

```

##      country      city      year bloom_doy
## Length:6182 Length:6182 Min.   :1950   Min.   : 8.0
## Class :character Class :character 1st Qu.:1972   1st Qu.: 93.0

```

```

##   Mode :character  Mode :character  Median :1987  Median : 99.0
##                                         Mean   :1987  Mean   :100.9
##                                         3rd Qu.:2002 3rd Qu.:110.0
##                                         Max.   :2021  Max.   :160.0
##      lat          long          alt          tmax
##  Min.   :24.34  Min.   :-77.04  Min.   : 0.0  Min.   :-1.231
##  1st Qu.:34.34  1st Qu.:130.98  1st Qu.: 5.7  1st Qu.:15.445
##  Median :35.54  Median :136.19  Median : 26.5  Median :19.180
##  Mean   :36.84  Mean   :126.61  Mean   : 77.2  Mean   :18.178
##  3rd Qu.:39.70  3rd Qu.:140.10  3rd Qu.: 51.1  3rd Qu.:20.695
##  Max.   :47.57  Max.   :145.59  Max.   :1390.0 Max.   :32.300
##      tmin          prcp          agdd_winter    tmax_winter
##  Min.   :-6.892  Min.   : 0.300  Min.   : 0.0  Min.   :-5.361
##  1st Qu.: 6.732  1st Qu.: 3.472  1st Qu.: 107.2 1st Qu.: 4.469
##  Median :10.368  Median : 4.799  Median : 277.6 Median : 8.940
##  Mean   : 9.929  Mean   : 6.389  Mean   : 320.1 Mean   : 8.089
##  3rd Qu.:12.197  3rd Qu.: 7.179  3rd Qu.: 459.1 3rd Qu.:11.041
##  Max.   :26.500  Max.   :122.900 Max.   :1713.3 Max.   :24.032
##      tmin_winter    prcp_winter    co2_percapita
##  Min.   :-19.2316  Min.   : 0.00  Min.   : 1.238
##  1st Qu.: -3.0480  1st Qu.: 16.64  1st Qu.: 5.963
##  Median :  0.4671  Median : 27.20  Median : 7.980
##  Mean   :  0.0480  Mean   : 37.12  Mean   : 7.504
##  3rd Qu.:  2.3828  3rd Qu.: 47.83  3rd Qu.: 9.736
##  Max.   : 18.4540  Max.   :525.25  Max.   :22.236
apply(df, 2, function(x) length(unique(x)))

##      country        city       year  bloom_doy      lat
## 4           4          105       72       127      112
##      long          alt       tmax      tmin      prcp
## 110         110         106      5417      5429     5177
##  agdd_winter  tmax_winter  tmin_winter  prcp_winter co2_percapita
## 4592        5516        5522      5453      272
unique(df$country)

## [1] "Japan"        "Switzerland"   "South Korea"   "United States"

```

## Density Plots

### Overall Density Plots

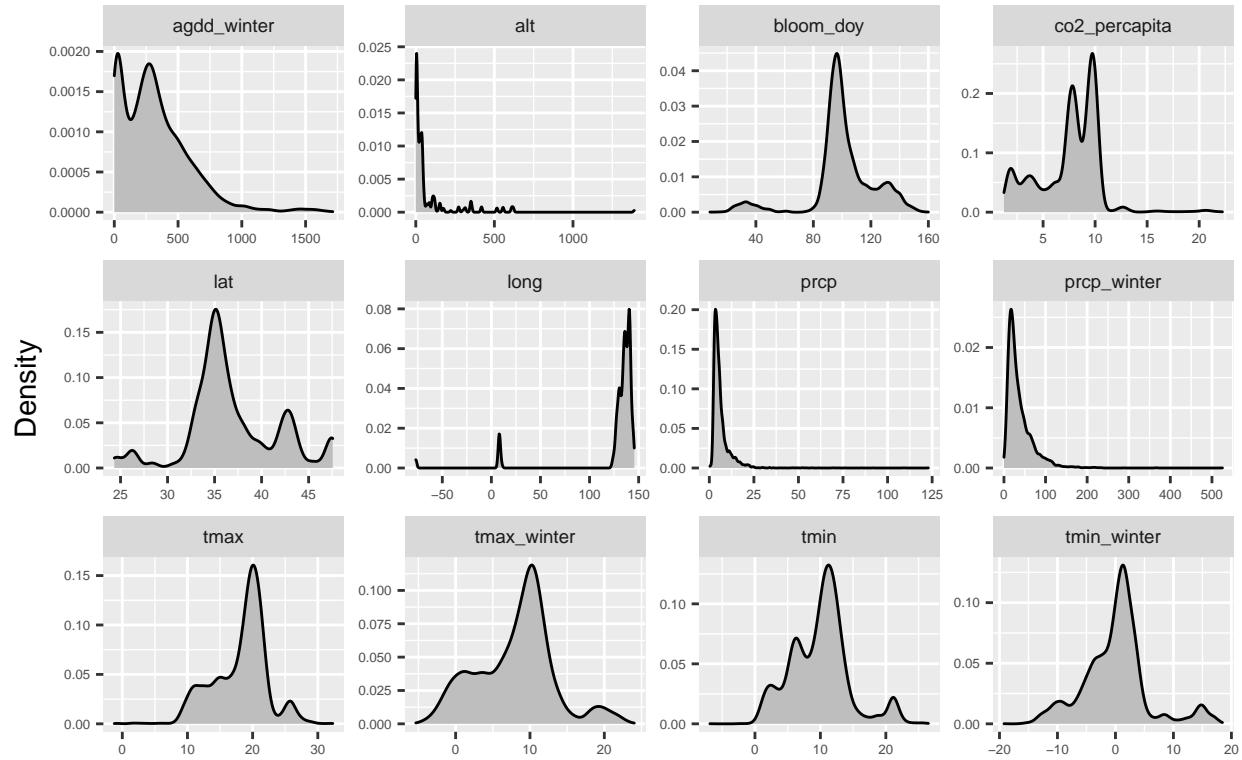
```

df %>%
  select_if(is.numeric) %>%
  dplyr::select(-year) %>%
  pivot_longer(everything()) %>%
  ggplot(aes(x=value)) +
  geom_density(fill='grey') +
  facet_wrap(~name, scales='free') +
  theme(strip.text = element_text(size=7),
        axis.text.x = element_text(size=5),
        axis.text.y = element_text(size=5),
        plot.title = element_text(hjust = 0.5)) +
  labs(x ="",

```

```
y = "Density",
title = 'Overall Density Plots')
```

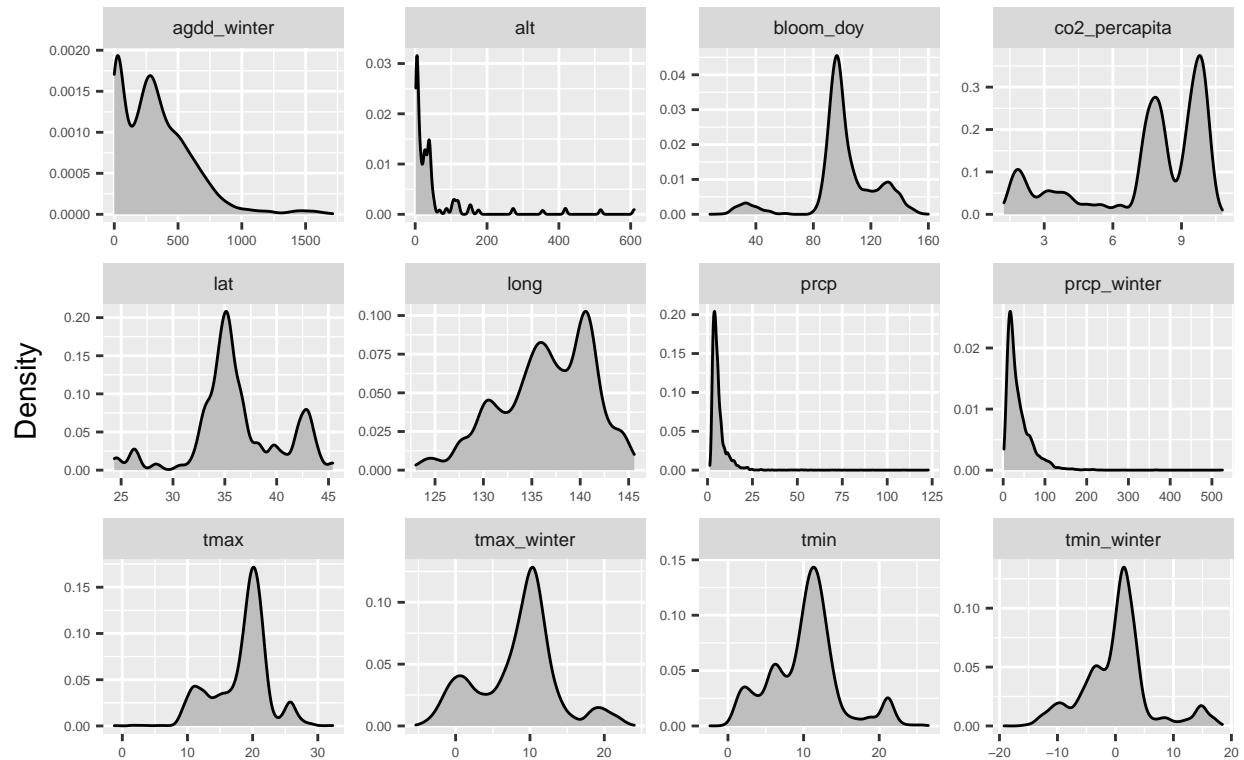
## Overall Density Plots



## Density Plots of Japan

```
df %>%
  filter(country=='Japan') %>%
  select_if(is.numeric) %>%
  dplyr::select(-year) %>%
  pivot_longer(everything()) %>%
  ggplot(aes(x=value)) +
  geom_density(fill='grey') +
  facet_wrap(~name, scales='free') +
  theme(strip.text = element_text(size=7),
        axis.text.x = element_text(size=5),
        axis.text.y = element_text(size=5),
        plot.title = element_text(hjust = 0.5)) +
  labs(x ="",
       y = "Density",
       title = 'Density Plots of Japan')
```

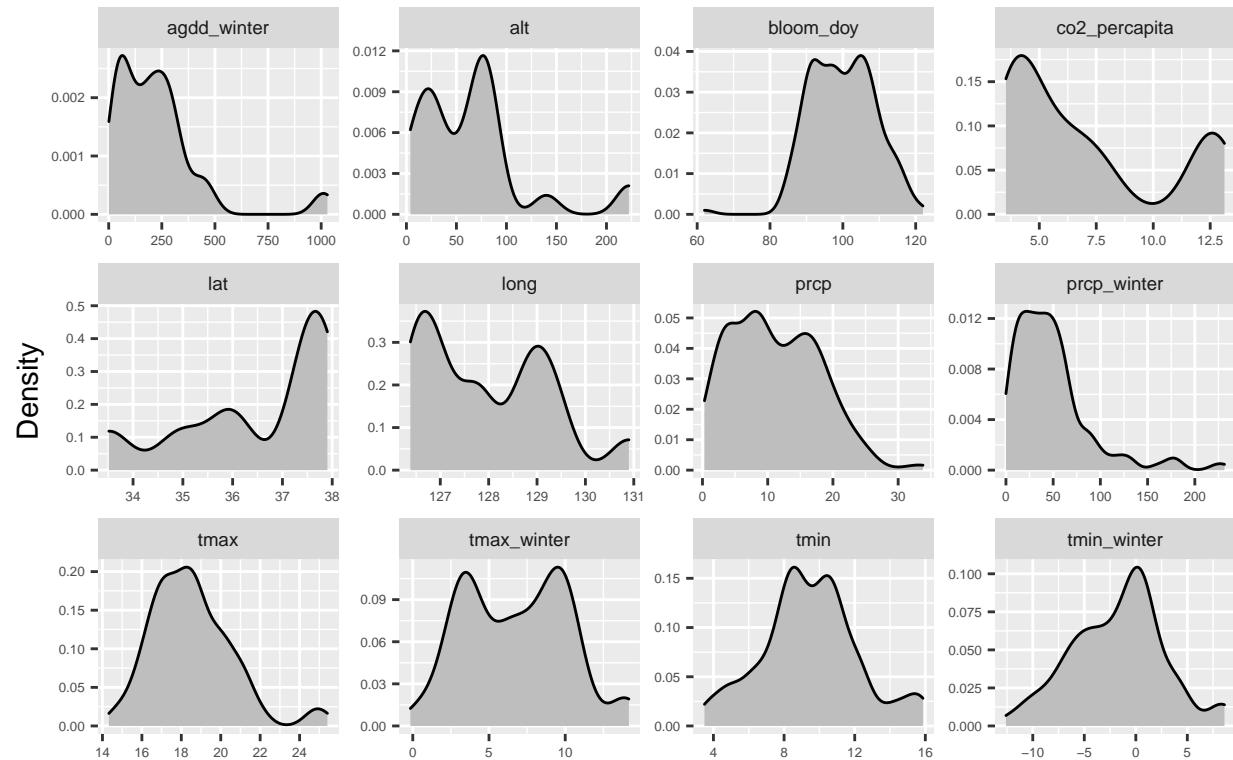
## Density Plots of Japan



## Density Plots of South Korea

```
df %>%
  filter(country=='South Korea') %>%
  select_if(is.numeric) %>%
  dplyr::select(-year) %>%
  pivot_longer(everything()) %>%
  ggplot(aes(x=value)) +
  geom_density(fill='grey') +
  facet_wrap(~name, scales='free') +
  theme(strip.text = element_text(size=7),
        axis.text.x = element_text(size=5),
        axis.text.y = element_text(size=5),
        plot.title = element_text(hjust = 0.5)) +
  labs(x ="",
       y = "Density",
       title = 'Density Plots of South Korea')
```

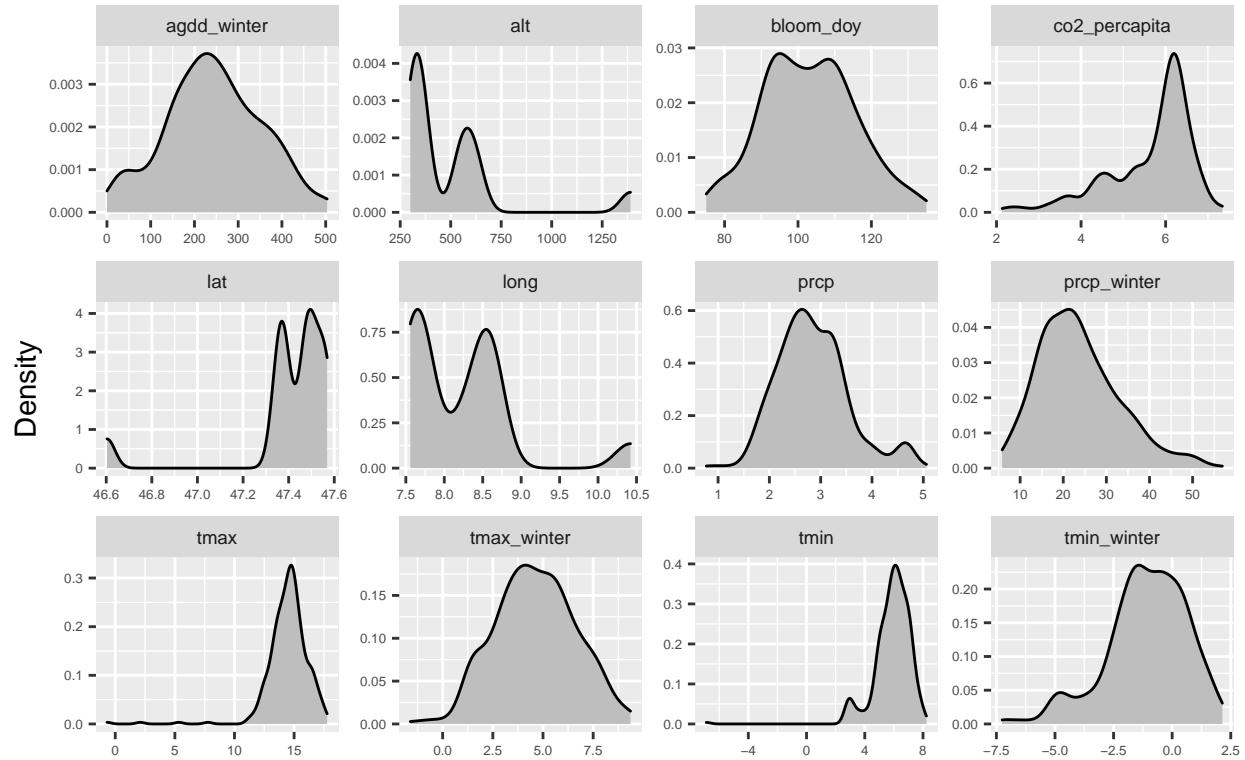
## Density Plots of South Korea



## Density Plots of Switzerland

```
df %>%
  filter(country=='Switzerland') %>%
  select_if(is.numeric) %>%
  dplyr::select(-year) %>%
  pivot_longer(everything()) %>%
  ggplot(aes(x=value)) +
  geom_density(fill='grey') +
  facet_wrap(~name, scales='free') +
  theme(strip.text = element_text(size=7),
        axis.text.x = element_text(size=5),
        axis.text.y = element_text(size=5),
        plot.title = element_text(hjust = 0.5)) +
  labs(x ="",
       y = "Density",
       title = 'Density Plots of Switzerland')
```

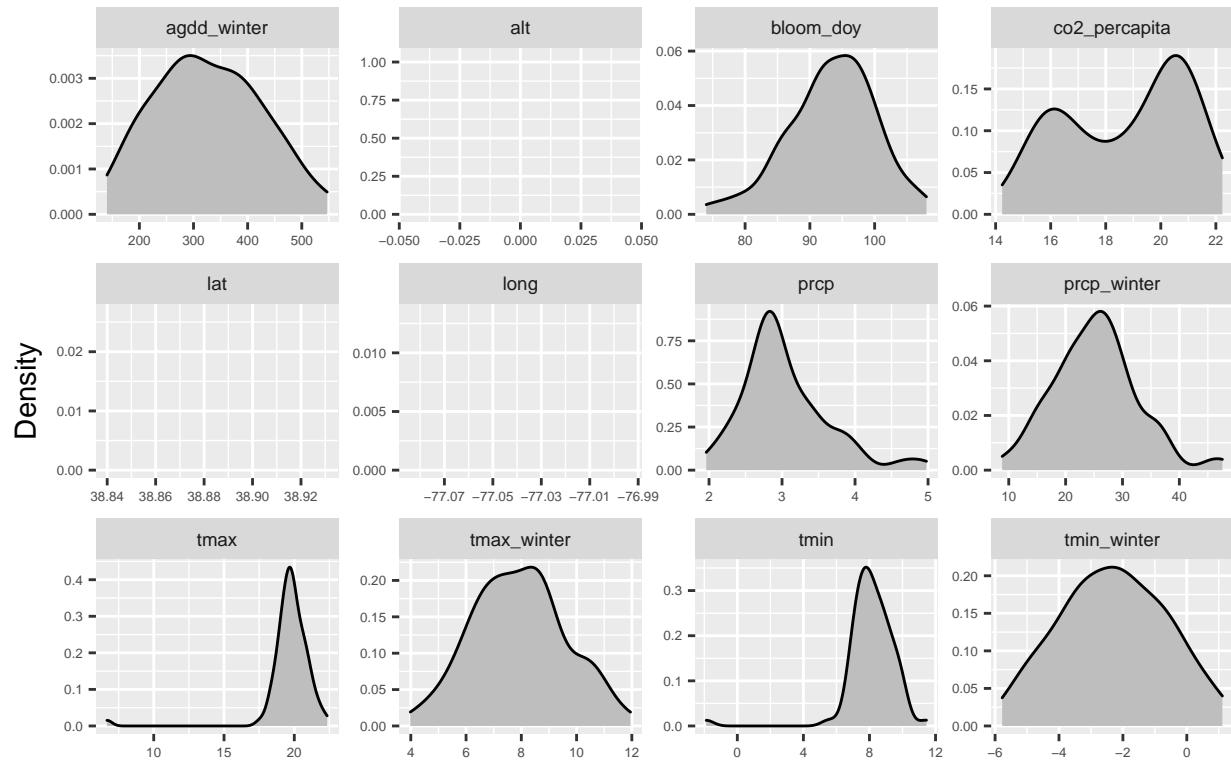
## Density Plots of Switzerland



## Density Plots of United States

```
df %>%
  filter(country=='United States') %>%
  select_if(is.numeric) %>%
  dplyr::select(-year) %>%
  pivot_longer(everything()) %>%
  ggplot(aes(x=value)) +
  geom_density(fill='grey') +
  facet_wrap(~name, scales='free') +
  theme(strip.text = element_text(size=7),
        axis.text.x = element_text(size=5),
        axis.text.y = element_text(size=5),
        plot.title = element_text(hjust = 0.5)) +
  labs(x ="",
       y = "Density",
       title = 'Density Plots of United States')
```

## Density Plots of United States



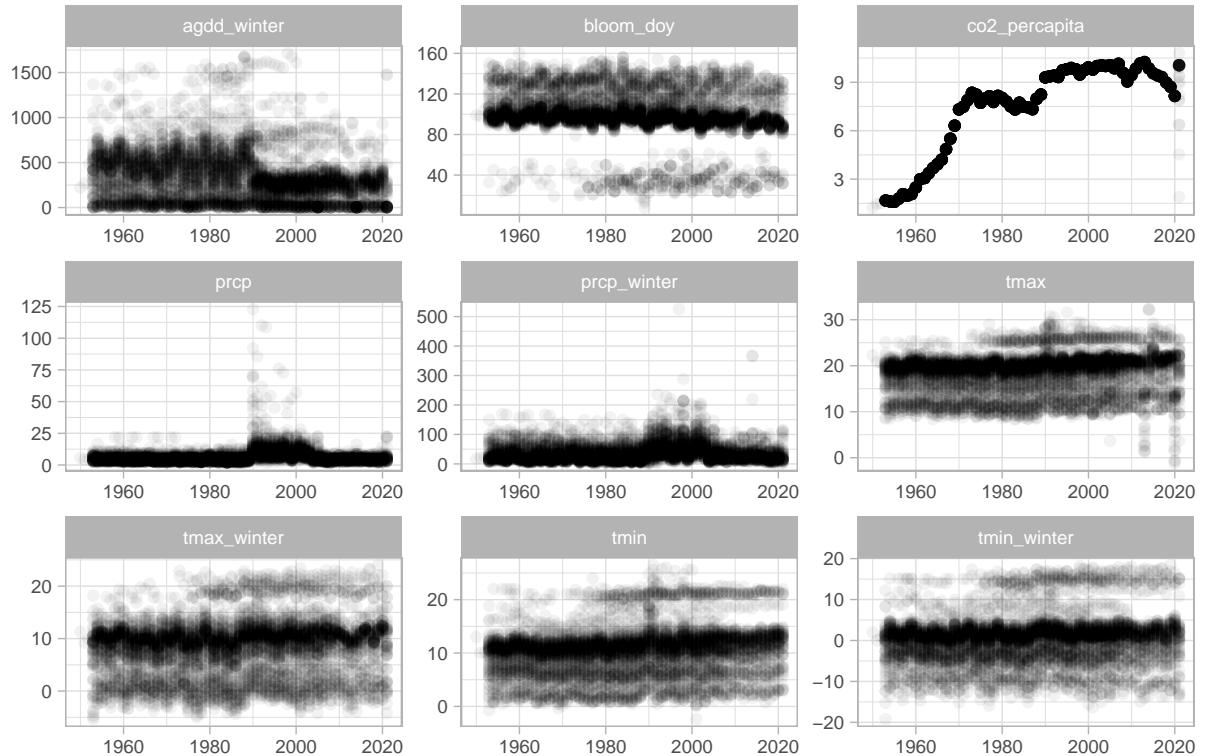
## Year v.s. Numeric Variables

### Year v.s. Numeric Variables of Japan

```
par(mfrow=c(3, 4))

df %>%
  filter(country=='Japan') %>%
  select_if(is.numeric) %>%
  pivot_longer(-c(year, lat, long, alt)) %>%
  ggplot(aes(x=year, y=value)) +
  geom_point(alpha = 0.05) +
  facet_wrap(~name, scales='free') +
  labs(x = "",
       y = "",
       title="Year v.s. Numeric Variables of Japan") +
  theme_light() +
  theme(strip.text = element_text(size=7),
        axis.text.x = element_text(size=7),
        axis.text.y = element_text(size=7),
        plot.title = element_text(hjust = 0.5))
```

## Year v.s. Numeric Variables of Japan

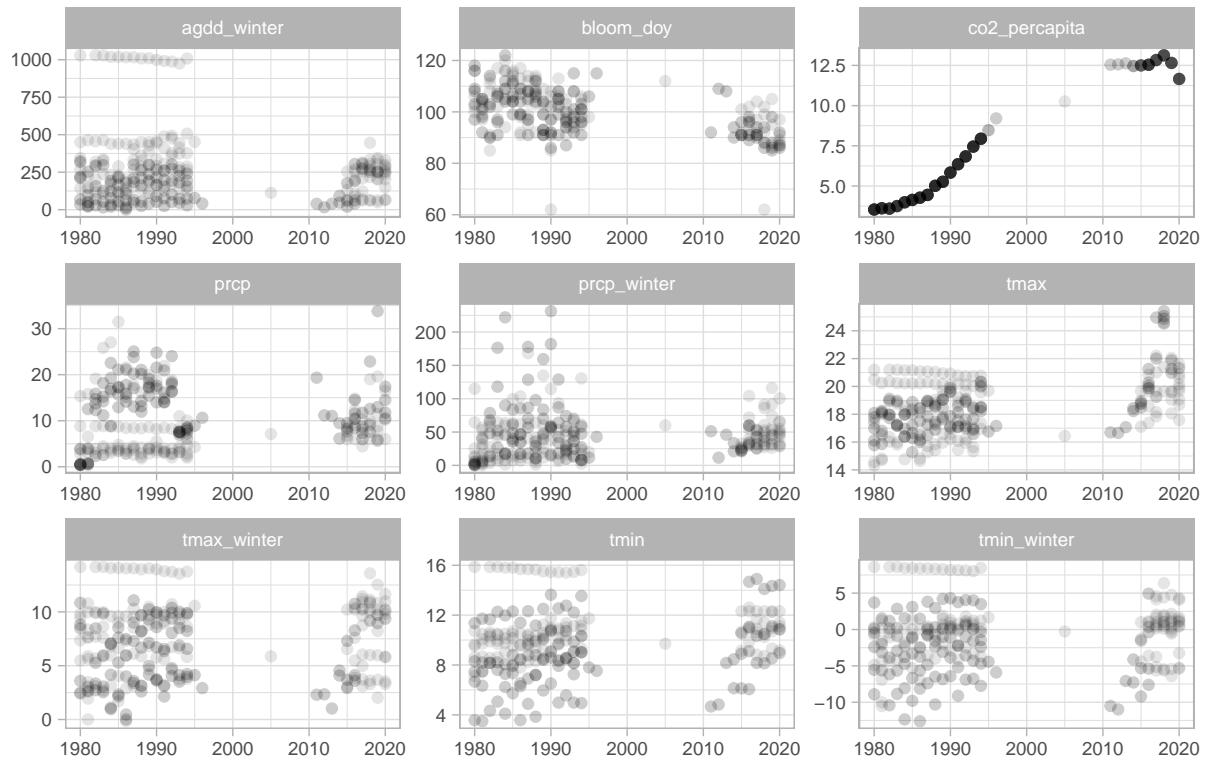


## Year v.s. Numeric Variables of South Korea

```
par(mfrow=c(3, 4))

df %>%
  filter(country=='South Korea') %>%
  select_if(is.numeric) %>%
  pivot_longer(-c(year, lat, long, alt)) %>%
  ggplot(aes(x=year, y=value)) +
  geom_point(alpha = 0.1) +
  facet_wrap(~name, scales='free') +
  labs(x = "",
       y = "",
       title="Year v.s. Numeric Variables of South Korea") +
  theme_light() +
  theme(strip.text = element_text(size=7),
        axis.text.x = element_text(size=7),
        axis.text.y = element_text(size=7),
        plot.title = element_text(hjust = 0.5))
```

## Year v.s. Numeric Variables of South Korea

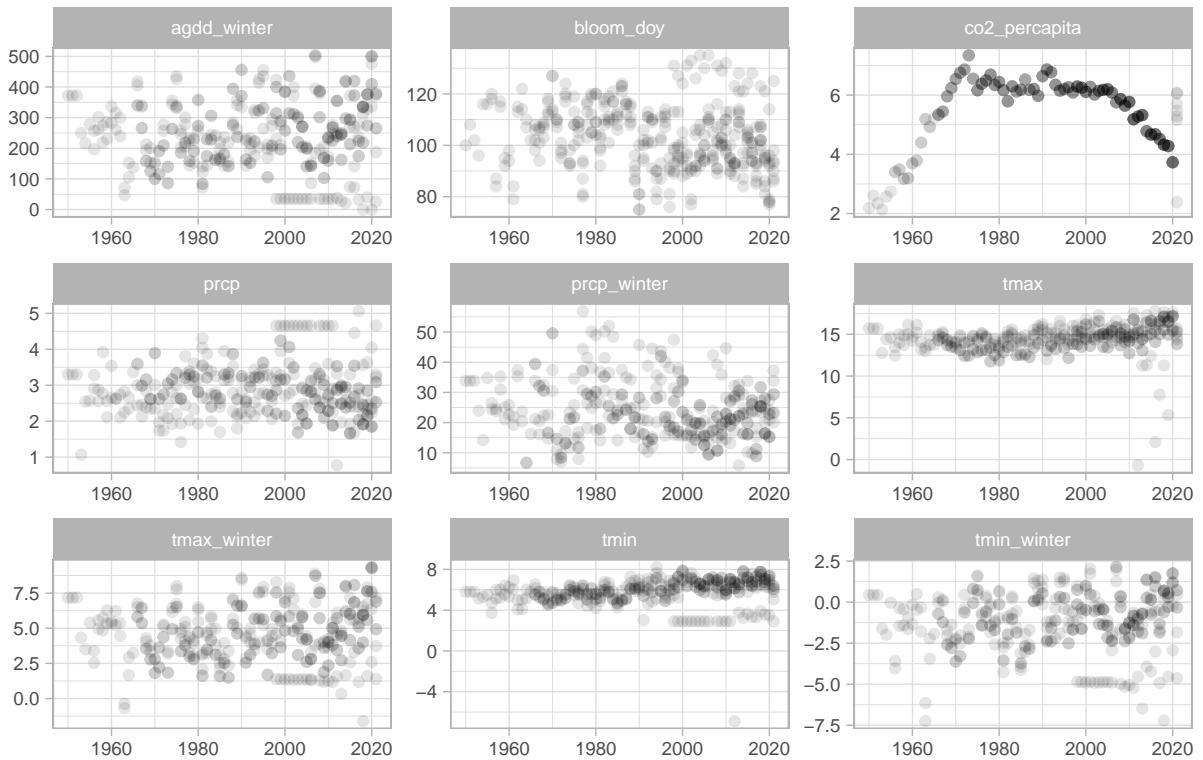


## Year v.s. Numeric Variables of Switzerland

```
par(mfrow=c(3, 4))

df %>%
  filter(country=='Switzerland') %>%
  select_if(is.numeric) %>%
  pivot_longer(-c(year, lat, long, alt)) %>%
  ggplot(aes(x=year, y=value)) +
  geom_point(alpha = 0.1) +
  facet_wrap(~name, scales='free') +
  labs(x = "",
       y = "",
       title="Year v.s. Numeric Variables of Switzerland") +
  theme_light() +
  theme(strip.text = element_text(size=7),
        axis.text.x = element_text(size=7),
        axis.text.y = element_text(size=7),
        plot.title = element_text(hjust = 0.5))
```

## Year v.s. Numeric Variables of Switzerland

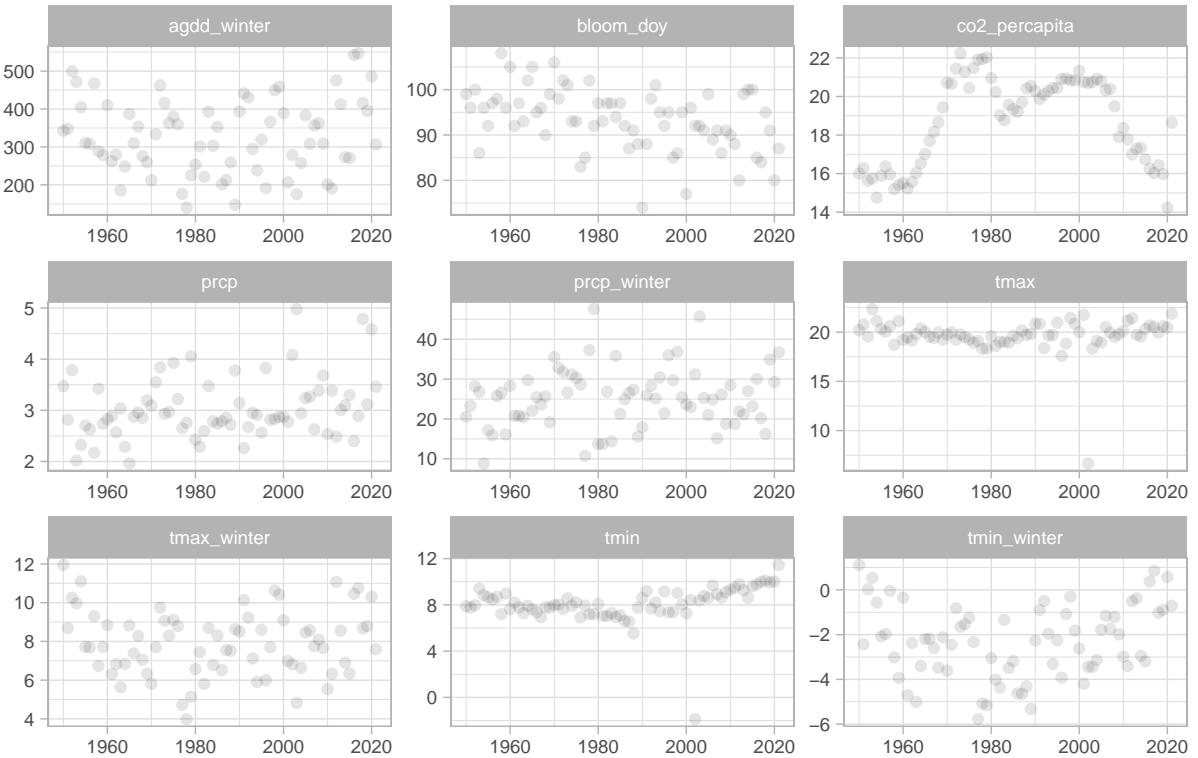


## Year v.s. Numeric Variables of United States

```
par(mfrow=c(3, 4))

df %>%
  filter(country=='United States') %>%
  select_if(is.numeric) %>%
  pivot_longer(-c(year, lat, long, alt)) %>%
  ggplot(aes(x=year, y=value)) +
  geom_point(alpha = 0.1) +
  facet_wrap(~name, scales='free') +
  labs(x = "",
       y = "",
       title="Year v.s. Numeric Variables of United States") +
  theme_light() +
  theme(strip.text = element_text(size=7),
        axis.text.x = element_text(size=7),
        axis.text.y = element_text(size=7),
        plot.title = element_text(hjust = 0.5))
```

## Year v.s. Numeric Variables of United States



## Visualization of Data Locations

```

world <- map_data("world")
all_plot <- ggplot() +
  geom_map(
    data = world, map = world,
    aes(long, lat, map_id = region, fill = region),
    color = "gray", fill = "white") +
  geom_point(
    data = df,
    aes(long, lat, color=country),
    alpha=0.1, size=1) +
  guides(colour = guide_legend(override.aes = list(alpha=0.9,size=4))) +
  labs(title='Data Locations') +
  theme(plot.title = element_text(hjust = 0.5))

jp <- map_data('world', 'Japan')
jp_plot <- ggplot() +
  geom_map(
    data = jp, map = jp,
    aes(long, lat, map_id = region),
    color = "white", fill = "white"
  ) +
  geom_point(

```

```

    data = df %>% filter(country=='Japan'),
    aes(long, lat), alpha=0.01) +
  ggtitle("Japan") +
  theme(plot.title = element_text(hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5))

swiss <- map_data('world', 'Switzerland')
swiss_plot <- ggplot() +
  geom_map(
    data = swiss, map = swiss,
    aes(long, lat, map_id = region),
    color = "gray", fill = "white"
  ) +
  geom_point(
    data = df %>% filter(country=='Switzerland'),
    aes(long, lat), alpha=0.01) +
  ggtitle("Switzerland") +
  theme(plot.title = element_text(hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5))

sk <- map_data('world', 'South Korea')
sk_plot <- ggplot() +
  geom_map(
    data = sk, map = sk,
    aes(long, lat, map_id = region),
    color = "white", fill = "white"
  ) +
  geom_point(
    data = df %>% filter(country=='South Korea'),
    aes(long, lat), alpha=0.01) +
  ggtitle("South Korea") +
  theme(plot.title = element_text(hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5))

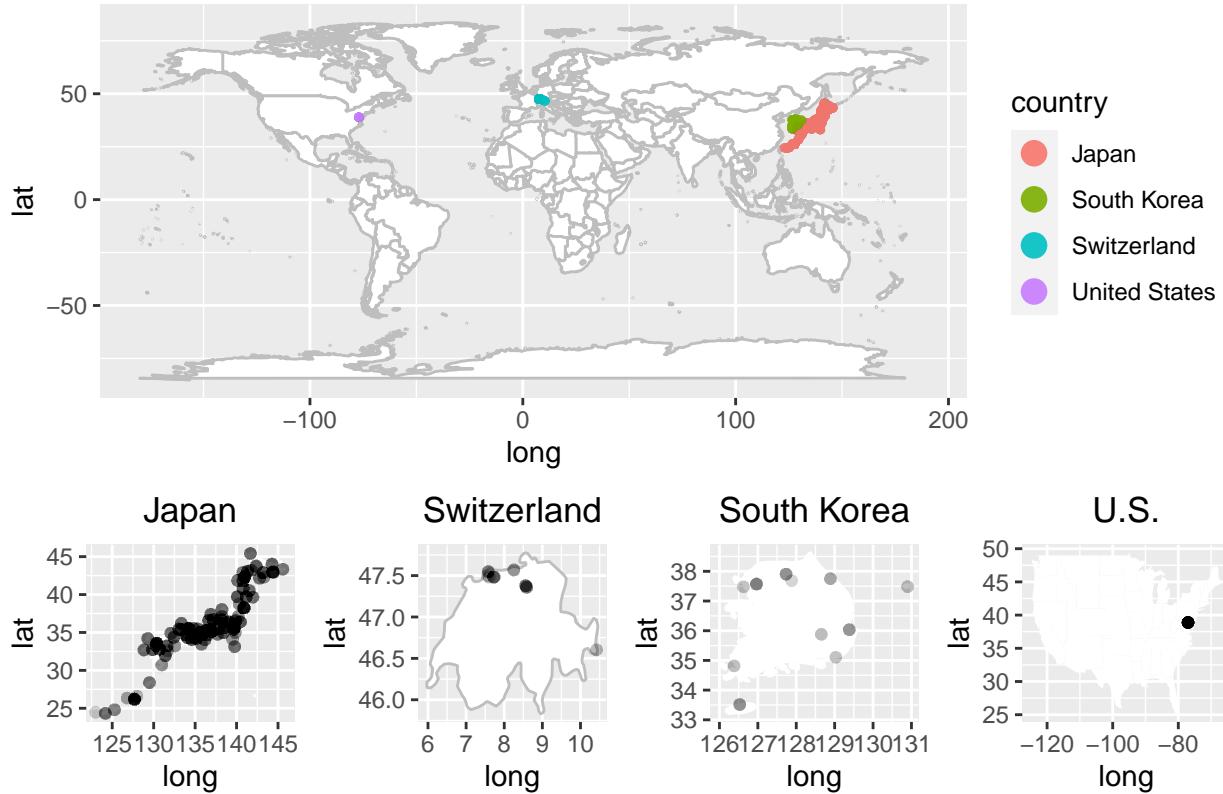
usa <- map_data("state")
usa_plot <- ggplot() +
  geom_map(
    data = usa, map = usa,
    aes(long, lat, map_id = region),
    fill = "white", size = 0.1
  ) +
  geom_point(
    data = df %>% filter(country=='United States'),
    aes(long, lat), alpha=0.1) +
  ggtitle("U.S.") +
  theme(plot.title = element_text(hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5))

individual_plots <- ggarrange(jp_plot, swiss_plot, sk_plot, usa_plot,
                               ncol = 4, nrow = 1)

ggarrange(all_plot, individual_plots,
          ncol = 1, nrow = 2, heights = c(2.5,1.5))

```

## Data Locations



## Basic Linear Models

```
df <- read.csv("../data/processed/clean_data.csv")
summary(df)
```

```
##      country          city        year   bloom_doy
##  Length:6182    Length:6182   Min.   :1950   Min.   : 8.0
##  Class :character  Class :character  1st Qu.:1972   1st Qu.: 93.0
##  Mode  :character  Mode  :character  Median :1987   Median : 99.0
##                                         Mean   :1987   Mean   :100.9
##                                         3rd Qu.:2002   3rd Qu.:110.0
##                                         Max.   :2021   Max.   :160.0
##      lat           long       alt      tmax
##  Min.   :24.34   Min.   :-77.04   Min.   : 0.0   Min.   :-1.231
##  1st Qu.:34.34   1st Qu.:130.98   1st Qu.: 5.7   1st Qu.:15.445
##  Median :35.54   Median :136.19   Median : 26.5   Median :19.180
##  Mean   :36.84   Mean   :126.61   Mean   : 77.2   Mean   :18.178
##  3rd Qu.:39.70   3rd Qu.:140.10   3rd Qu.: 51.1   3rd Qu.:20.695
##  Max.   :47.57   Max.   :145.59   Max.   :1390.0  Max.   :32.300
##      tmin         prcp     agdd_winter   tmax_winter
##  Min.   :-6.892   Min.   : 0.300   Min.   : 0.0   Min.   :-5.361
##  1st Qu.: 6.732   1st Qu.: 3.472   1st Qu.:107.2   1st Qu.: 4.469
##  Median :10.368   Median : 4.799   Median :277.6   Median : 8.940
##  Mean   : 9.929   Mean   : 6.389   Mean   :320.1   Mean   : 8.089
##  3rd Qu.:12.197   3rd Qu.: 7.179   3rd Qu.:459.1   3rd Qu.:11.041
```

```

##   Max.    :26.500   Max.    :122.900   Max.    :1713.3   Max.    :24.032
##   tmin_winter      prcp_winter      co2_percapita
##   Min.   :-19.2316   Min.    :  0.00   Min.    : 1.238
##   1st Qu.: -3.0480   1st Qu.: 16.64   1st Qu.: 5.963
##   Median  :  0.4671   Median  : 27.20   Median  : 7.980
##   Mean    :  0.0480   Mean    : 37.12   Mean    : 7.504
##   3rd Qu.:  2.3828   3rd Qu.: 47.83   3rd Qu.: 9.736
##   Max.    : 18.4540   Max.    :525.25   Max.    :22.236

df <- df %>% na.omit
train <- df[sort(sample(nrow(df), nrow(df)*.7)),]
test <- df[-sort(sample(nrow(df), nrow(df)*.7)),]
```

## Fit Linear Regression

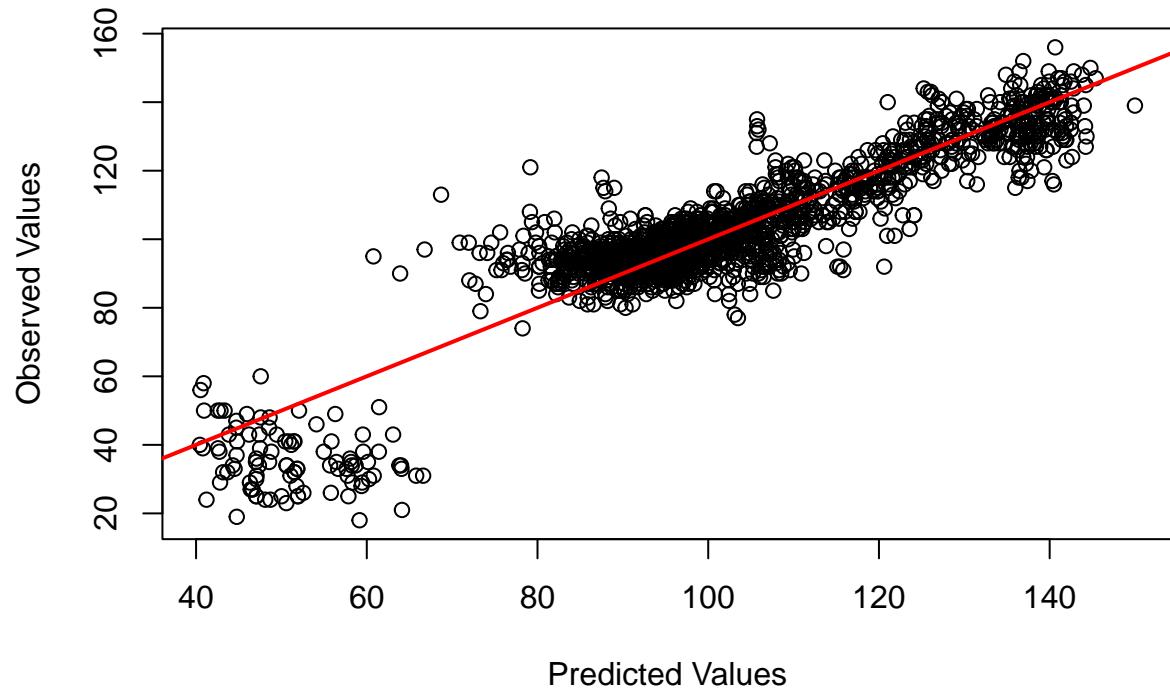
```

lm <- lm(bloom_doy~.-country-city-year, data = train)
summary(lm)

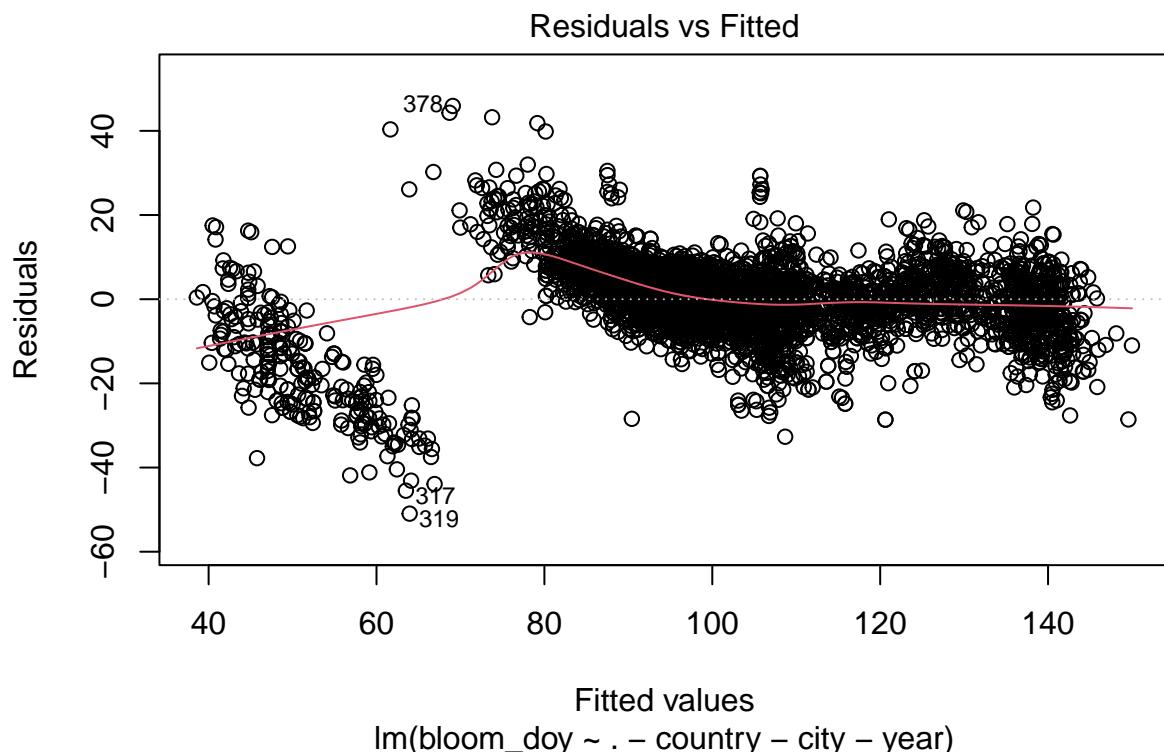
## 
## Call:
## lm(formula = bloom_doy ~ . - country - city - year, data = train)
## 
## Residuals:
##     Min      1Q  Median      3Q      Max 
## -50.949  -3.958   0.735   4.919   45.914 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 43.5056606  5.0484484   8.618 < 2e-16 ***
## lat          1.7488148  0.0899609  19.440 < 2e-16 ***
## long         0.1435628  0.0057362  25.028 < 2e-16 ***
## alt          -0.0090241  0.0011353  -7.949 2.39e-15 ***
## tmax        -0.5548183  0.0983597  -5.641 1.80e-08 ***
## tmin        -1.2007539  0.1625856  -7.385 1.82e-13 *** 
## prcp         0.1028125  0.0293514   3.503 0.000465 *** 
## agdd_winter  0.0063143  0.0008784   7.188 7.70e-13 *** 
## tmax_winter -0.5720664  0.1132419  -5.052 4.56e-07 *** 
## tmin_winter -0.7426349  0.1174456  -6.323 2.82e-10 *** 
## prcp_winter -0.0179078  0.0052243  -3.428 0.000614 *** 
## co2_percapita 0.0190862  0.0533302   0.358 0.720445  
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 9.023 on 4315 degrees of freedom
## Multiple R-squared:  0.8347, Adjusted R-squared:  0.8343 
## F-statistic: 1981 on 11 and 4315 DF,  p-value: < 2.2e-16

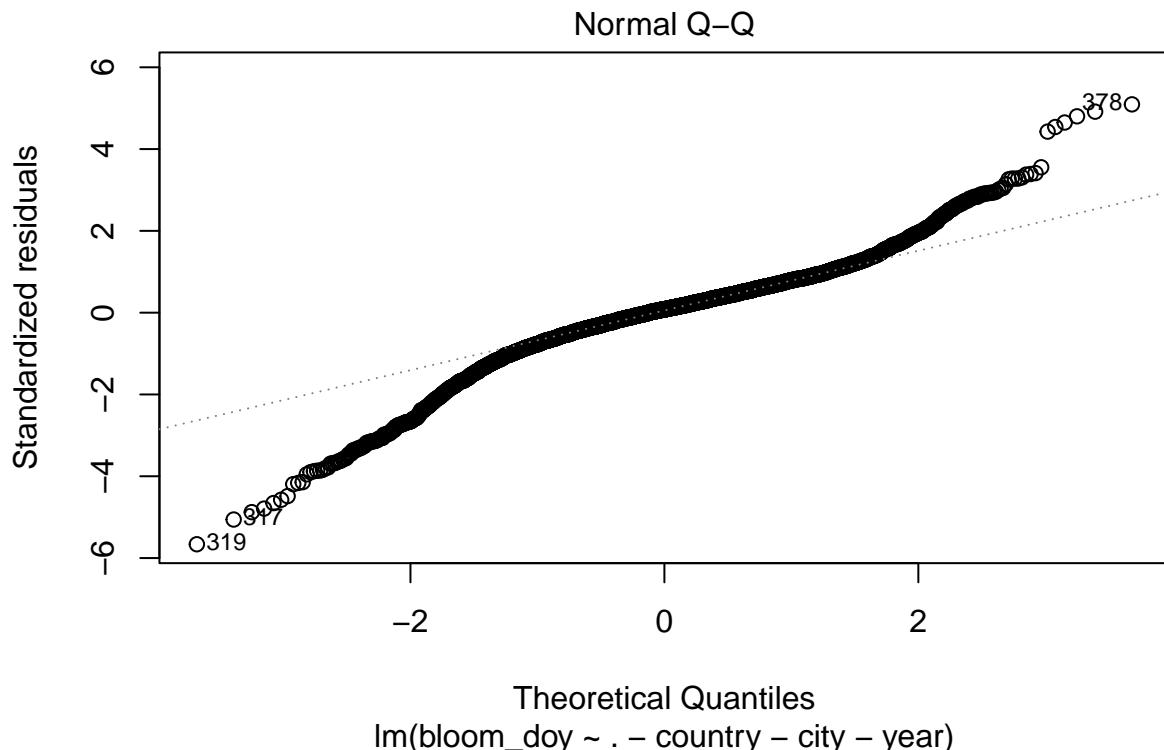
plot(predict(lm, test),
      test$bloom_doy,
      xlab = "Predicted Values",
      ylab = "Observed Values",
      main = 'Predicted Values v.s. Observed Values of Linear Regression')
abline(a=0, b=1, lwd = 2)
```

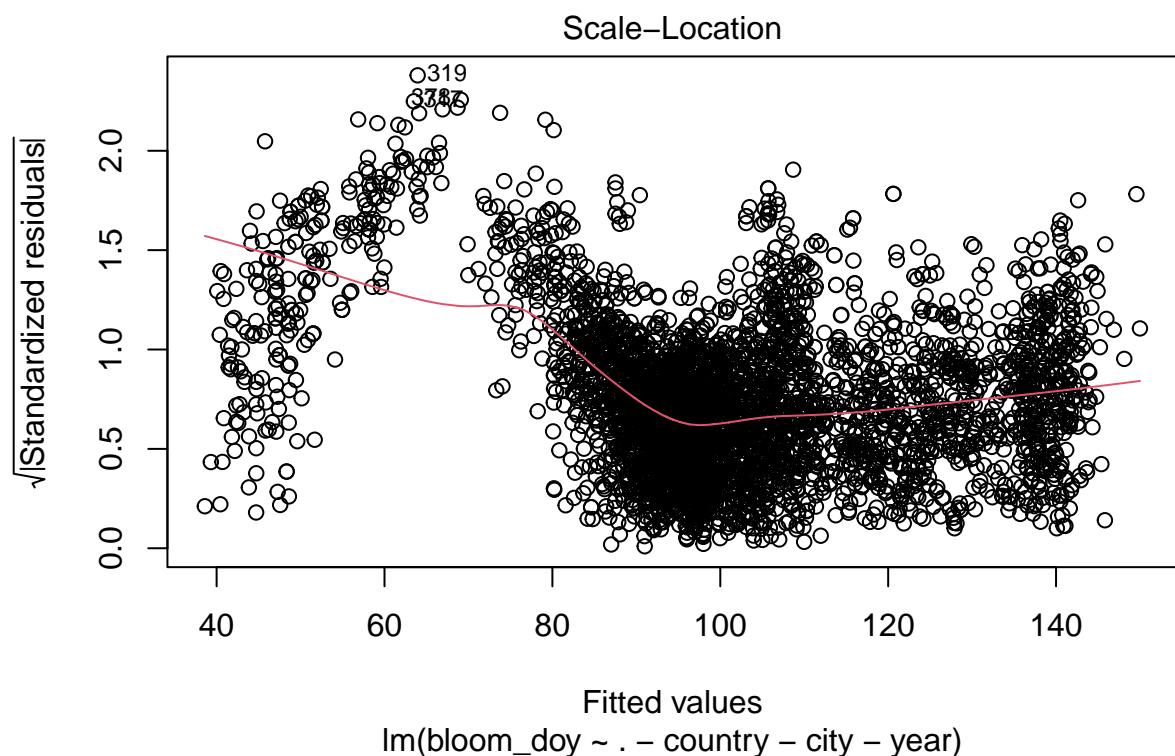
## Predicted Values v.s. Observed Values of Linear Regression

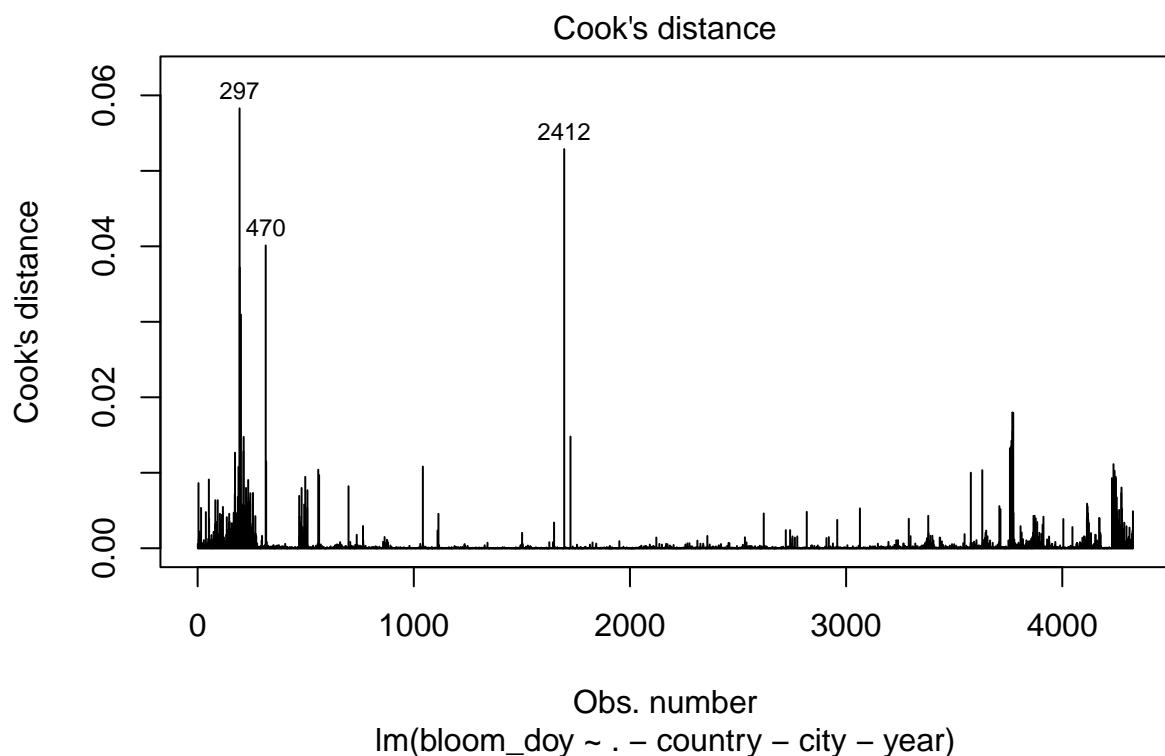


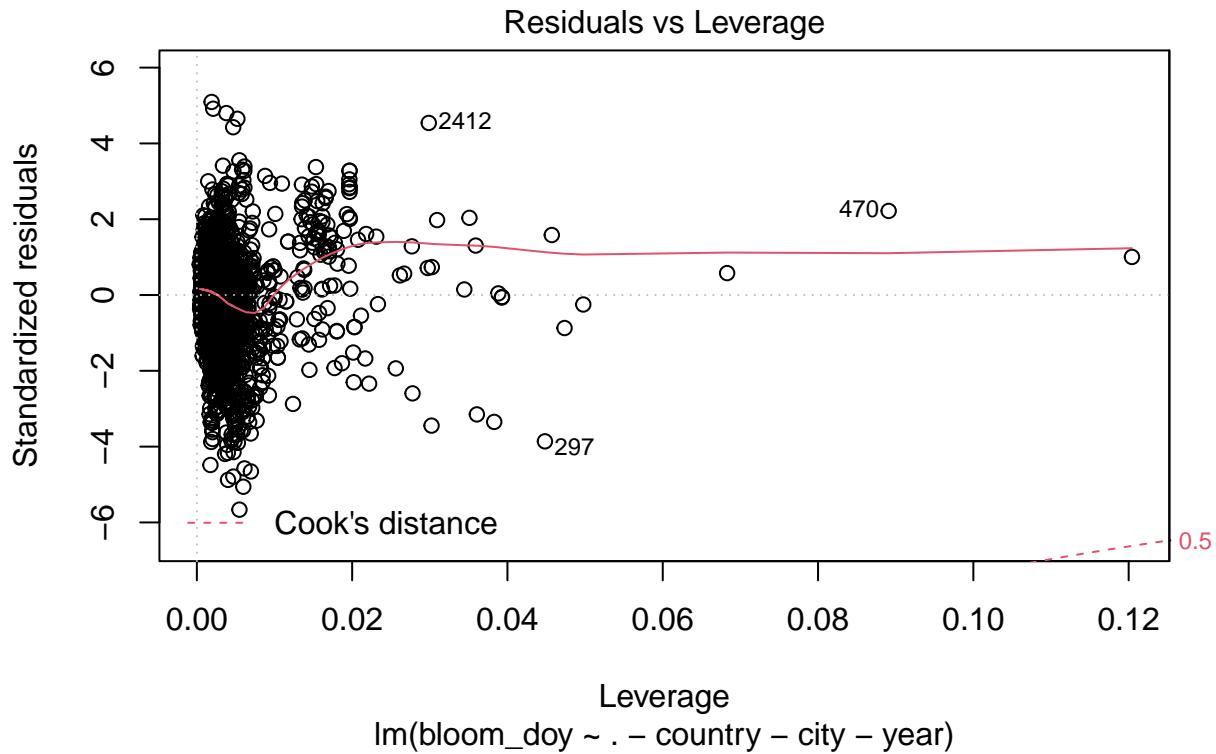
```
plot(lm, which = c(1,2,3,4,5))
```











## Confidence Intervals

The confidence interval reflects the uncertainty around the mean predictions. To display the 95% confidence intervals around the mean the predictions, specify the option `interval = "confidence"`.

For example, the 95% confidence interval associated with the first observation in the test data set is (43.02558, 47.14276). This means that, according to our model, a tree with same parameters with the first observation has, on average, a bloom\_doy between 43 and 47.

```
lm_confidence_intervals <- predict(lm, newdata = test, interval = "confidence")
head(lm_confidence_intervals)
```

```
##          fit      lwr      upr
## 4  44.80336 42.90035 46.70638
## 7  44.76553 42.79727 46.73380
## 9  44.30365 43.33697 45.27032
## 18 44.51024 43.44869 45.57180
## 24 42.58042 41.60738 43.55346
## 27 42.86219 41.86593 43.85844
```

## Prediction Intervals

The prediction interval gives uncertainty around a single value. In the same way, as the confidence intervals, the prediction intervals can be computed as follow.

The 95% prediction intervals associated with the first observation in the test dataset is (27.33479, 62.83354). This means that, according to our model, 95% of trees with same parameters have bloom\_doy between 27 and 63.

```

lm_prediction_intervals <- predict(lm, newdata = test, interval = "prediction")
head(lm_prediction_intervals)

##          fit      lwr      upr
## 4 44.80336 27.01240 62.59433
## 7 44.76553 26.96747 62.56359
## 9 44.30365 26.58836 62.01893
## 18 44.51024 26.78953 62.23096
## 24 42.58042 24.86479 60.29605
## 27 42.86219 25.14526 60.57911

```

## Check multicollinearity

First we check the correlation matrix of independent variables. We found that there are several variables that have high correlation with each other.

```

X <- df %>% dplyr::select(-c(country, city, year, bloom_doy))
cormat <- round(cor(X), 2)

reorder_cormat <- function(cormat){
  # Use correlation between variables as distance
  dd <- as.dist((1-cormat)/2)
  hc <- hclust(dd)
  cormat <- cormat[hc$order, hc$order]
}

# Get lower triangle of the correlation matrix
get_lower_tri <- function(cormat){
  cormat[upper.tri(cormat)] <- NA
  return(cormat)
}

# Get upper triangle of the correlation matrix
get_upper_tri <- function(cormat){
  cormat[lower.tri(cormat)] <- NA
  return(cormat)
}

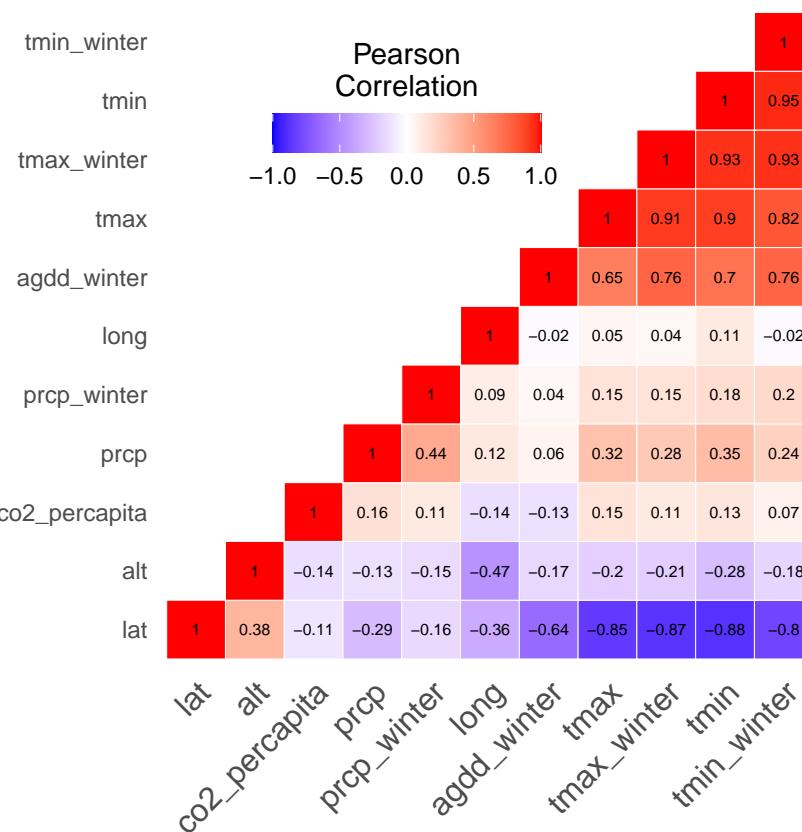
# Reorder the correlation matrix
cormat <- reorder_cormat(cormat)
upper_tri <- get_upper_tri(cormat)
# Melt the correlation matrix
melted_cormat <- melt(upper_tri, na.rm = TRUE)
# Create a ggheatmap
ggheatmap <- ggplot(melted_cormat, aes(Var2, Var1, fill = value))+
  geom_tile(color = "white")+
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
    midpoint = 0, limit = c(-1,1), space = "Lab",
    name="Pearson\nCorrelation") +
  theme_minimal() + # minimal theme
  theme(axis.text.x = element_text(angle = 45, vjust = 1,
    size = 12, hjust = 1))+
  coord_fixed() +
  geom_text(aes(Var2, Var1, label = value), color = "black", size = 2) +

```

```

theme(
  axis.title.x = element_blank(),
  axis.title.y = element_blank(),
  panel.grid.major = element_blank(),
  panel.border = element_blank(),
  panel.background = element_blank(),
  axis.ticks = element_blank(),
  legend.justification = c(1, 0),
  legend.position = c(0.6, 0.7),
  legend.direction = "horizontal") +
guides(fill = guide_colorbar(barwidth = 7, barheight = 1,
                             title.position = "top", title.hjust = 0.5))
# Print the heatmap
print(ggheatmap)

```



The ‘mctest’ package in R provides the Farrar-Glauber test and other relevant tests for multicollinearity. Here we check the overall and individual diagnostic checking for multicollinearity respectively. The calculated value of the Chi-square test statistic is found to be 49102.7, implying the presence of multicollinearity in the model.

Next, we check Farrar–Glauber test (F-test) for the location of the multicollinearity. The VIF, TOL and Wi columns provide the diagnostic output for variance inflation factor, tolerance and Farrar-Glauber F-test respectively. Variables such as `tmin`, `tmin_winter`, and `tmax_winter` show high correlation with other features.

```

options(scipen=999)
omcdiag(lm)

```

```

##
## Call:
## omcdiag(mod = lm)
##
##
## Overall Multicollinearity Diagnostics
##
## MC Results detection
## Determinant |X'X|: 0.0000 1
## Farrar Chi-Square: 48475.7487 1
## Red Indicator: 0.4674 0
## Sum of Lambda Inverse: 99.1698 1
## Theil's Method: -1.2214 0
## Condition Number: 130.0032 1
##
## 1 --> COLLINEARITY is detected by the test
## 0 --> COLLINEARITY is not detected by the test
imcdiag(lm)

##
## Call:
## imcdiag(mod = lm)
##
##
## All Individual Multicollinearity Diagnostics Result
##
##          VIF    TOL      Wi      Fi Leamer    CVIF Klein   IND1
## lat       9.8200 0.1018  3806.7151  4230.6635 0.3191 -0.5230  1 0.0002
## long      2.3408 0.4272   578.6956   643.1441 0.6536 -0.1247  0 0.0010
## alt        1.5366 0.6508   231.5952   257.3877 0.8067 -0.0818  0 0.0015
## tmax      8.8616 0.1128  3393.0560  3770.9357 0.3359 -0.4719  1 0.0003
## tmin      28.0171 0.0357 11660.5675 12959.1881 0.1889 -1.4921  1 0.0001
## prcp       1.6164 0.6187   266.0379   295.6662 0.7865 -0.0861  0 0.0014
## agdd_winter 3.0384 0.3291   879.7898   977.7707 0.5737 -0.1618  0 0.0008
## tmax_winter 18.1967 0.0550  7422.0847  8248.6715 0.2344 -0.9691  1 0.0001
## tmin_winter 23.0721 0.0433  9526.3074 10587.2384 0.2082 -1.2288  1 0.0001
## prcp_winter 1.3849 0.7221   166.1388   184.6415 0.8497 -0.0738  0 0.0017
## co2_percapita 1.2852 0.7781   123.1070   136.8172 0.8821 -0.0684  0 0.0018
##          IND2
## lat       1.3866
## long      0.8843
## alt        0.5391
## tmax      1.3696
## tmin      1.4887
## prcp       0.5887
## agdd_winter 1.0357
## tmax_winter 1.4589
## tmin_winter 1.4769
## prcp_winter 0.4291
## co2_percapita 0.3426
##
## 1 --> COLLINEARITY is detected by the test
## 0 --> COLLINEARITY is not detected by the test
##

```

```

## co2_percapita , coefficient(s) are non-significant may be due to multicollinearity
##
## R-square of y on all x: 0.8347
##
## * use method argument to check which regressors may be the reason of collinearity
## =====

```

## Mix Effects Model

```

suppressWarnings(suppressMessages(
  print(full_mixed_model <- lmer(bloom_doy ~ tmin_winter+
    prcp +
    agdd_winter +
    prcp_winter +
    co2_percapita +
    (tmin_winter +
      prcp +
      agdd_winter +
      prcp_winter +
      co2_percapita | city),
    data = train
  ))))

## Linear mixed model fit by REML ['lmerMod']
## Formula: bloom_doy ~ tmin_winter + prcp + agdd_winter + prcp_winter +
##           co2_percapita + (tmin_winter + prcp + agdd_winter + prcp_winter +
##           co2_percapita | city)
## Data: train
## REML criterion at convergence: 27719.88
## Random effects:
##   Groups   Name        Std.Dev. Corr
##   city     (Intercept) 9.955400
##           tmin_winter  0.096763 -0.09
##           prcp       0.178081 -0.20  0.96
##           agdd_winter  0.007048  0.05  0.90  0.76
##           prcp_winter  0.027074  0.24 -0.24 -0.47 -0.07
##           co2_percapita 0.416789 -0.10  0.05 -0.05  0.20  0.22
##   Residual            5.469872
## Number of obs: 4327, groups: city, 105
## Fixed Effects:
##   (Intercept)  tmin_winter      prcp    agdd_winter    prcp_winter
##   101.308378   -1.584163     -0.108472     0.004587     0.004692
##   co2_percapita
##   -0.228248
## optimizer (nloptwrap) convergence code: 0 (OK) ; 0 optimizer warnings; 2 lme4 warnings
summary(full_mixed_model)

## Linear mixed model fit by REML ['lmerMod']
## Formula: bloom_doy ~ tmin_winter + prcp + agdd_winter + prcp_winter +
##           co2_percapita + (tmin_winter + prcp + agdd_winter + prcp_winter +
##           co2_percapita | city)
## Data: train
##
## REML criterion at convergence: 27719.9

```

```

## 
## Scaled residuals:
##      Min     1Q Median     3Q    Max
## -4.6704 -0.5801  0.0248  0.5941  4.2468
## 
## Random effects:
##   Groups   Name        Variance Std.Dev. Corr
##   city     (Intercept) 99.10999198 9.955400
##          tmin_winter  0.00936315 0.096763 -0.09
##          prcp        0.03171295 0.178081 -0.20  0.96
##          agdd_winter  0.00004967 0.007048  0.05  0.90  0.76
##          prcp_winter  0.00073302 0.027074  0.24 -0.24 -0.47 -0.07
##          co2_percapita 0.17371308 0.416789 -0.10  0.05 -0.05  0.20  0.22
##   Residual           29.91950345 5.469872
## Number of obs: 4327, groups: city, 105
## 
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept) 101.308378  1.133088 89.409
## tmin_winter -1.584163  0.082180 -19.277
## prcp        -0.108472  0.035286 -3.074
## agdd_winter  0.004587  0.001216  3.773
## prcp_winter  0.004692  0.005947  0.789
## co2_percapita -0.228248  0.059252 -3.852
## 
## Correlation of Fixed Effects:
##            (Intr) tmn_wn prcp   agdd_w prcp_w
## tmin_winter  0.189
## prcp        -0.186 -0.032
## agdd_winter -0.298 -0.280  0.386
## prcp_winter -0.010 -0.131 -0.479  0.047
## co2_percept -0.324 -0.173 -0.062  0.297  0.067
## optimizer (nloptwrap) convergence code: 0 (OK)
## unable to evaluate scaled gradient
## Model failed to converge: degenerate Hessian with 2 negative eigenvalues
head(coef(full_mixed_model)$city)

##                (Intercept) tmin_winter       prcp  agdd_winter prcp_winter
## Abashiri        121.58670 -1.551612 -0.1097274 0.009062955 0.026500786
## Aikawa         109.21844 -1.640987 -0.2346136 0.002457516 0.015654096
## Akita          112.13148 -1.634515 -0.2184386 0.003276635 0.007706353
## Aomori         114.97595 -1.628508 -0.2409242 0.004033154 0.030404764
## Asahikawa     116.20559 -1.579076 -0.1267286 0.005492130 0.017847082
## Basel-Binningen 99.50059 -1.740434 -0.3631678 -0.007380650 0.012433915
##                co2_percapita
## Abashiri        -0.2735121
## Aikawa          0.1312248
## Akita          -0.1781142
## Aomori         -0.1932020
## Asahikawa      -0.6748046
## Basel-Binningen -0.1362196

rq <- rq(bloom_doy~tmin_winter+prcp+agdd_winter+prcp_winter+co2_percapita,
          data = train,

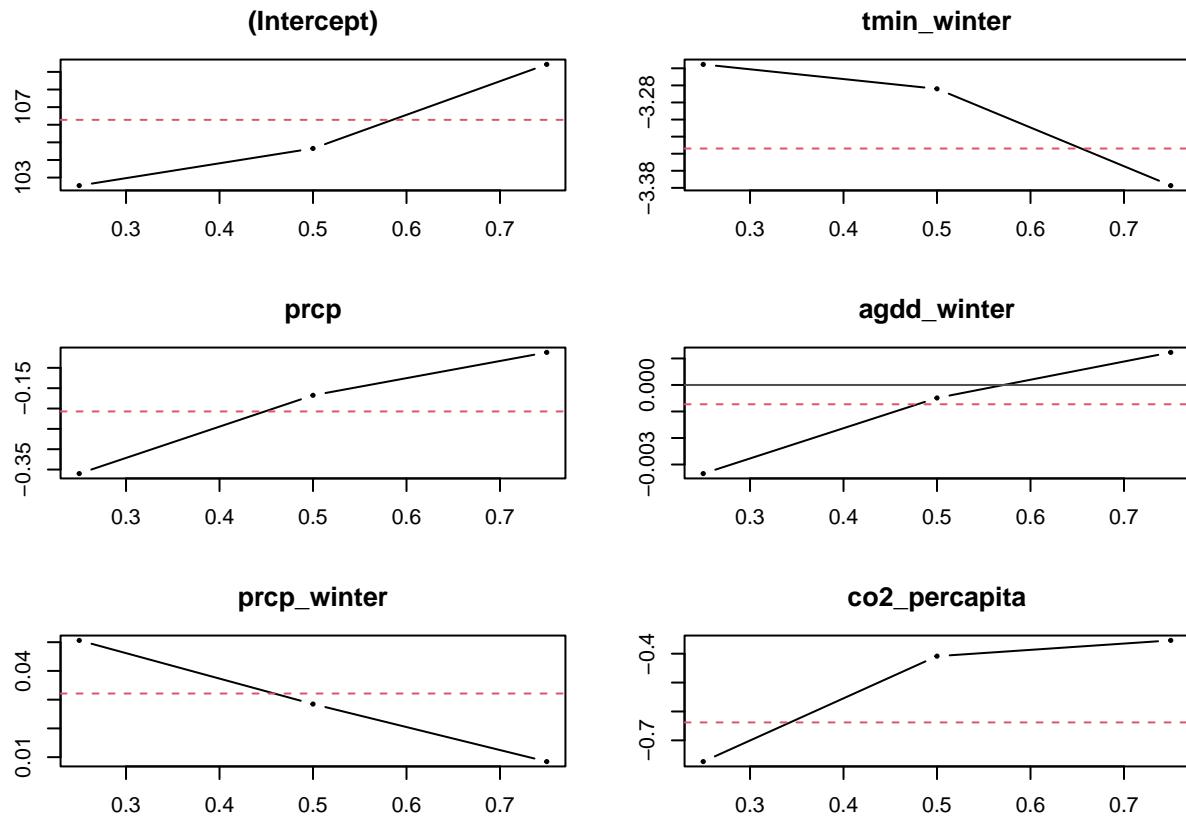
```

```

tau=c(0.25, 0.5, 0.75))
summary(rq)

##
## Call: rq(formula = bloom_doy ~ tmin_winter + prcp + agdd_winter + prcp_winter +
##          co2_per capita, tau = c(0.25, 0.5, 0.75), data = train)
##
## tau: [1] 0.25
##
## Coefficients:
##             Value     Std. Error t value   Pr(>|t|)
## (Intercept) 102.54647  0.79978 128.21904 0.00000
## tmin_winter -3.23544  0.05700 -56.76203 0.00000
## prcp        -0.35990  0.10602 -3.39456 0.00069
## agdd_winter -0.00334  0.00134 -2.49739 0.01255
## prcp_winter  0.05062  0.00918  5.51132 0.00000
## co2_per capita -0.77353  0.07149 -10.81972 0.00000
##
## Call: rq(formula = bloom_doy ~ tmin_winter + prcp + agdd_winter + prcp_winter +
##          co2_per capita, tau = c(0.25, 0.5, 0.75), data = train)
##
## tau: [1] 0.5
##
## Coefficients:
##             Value     Std. Error t value   Pr(>|t|)
## (Intercept) 104.65388  0.62537 167.34778 0.00000
## tmin_winter -3.26401  0.04863 -67.12588 0.00000
## prcp        -0.16750  0.06751 -2.48125 0.01313
## agdd_winter -0.00049  0.00100 -0.48754 0.62590
## prcp_winter  0.02848  0.00617  4.61862 0.00000
## co2_per capita -0.40853  0.05319 -7.68032 0.00000
##
## Call: rq(formula = bloom_doy ~ tmin_winter + prcp + agdd_winter + prcp_winter +
##          co2_per capita, tau = c(0.25, 0.5, 0.75), data = train)
##
## tau: [1] 0.75
##
## Coefficients:
##             Value     Std. Error t value   Pr(>|t|)
## (Intercept) 109.42526  0.70274 155.71160 0.00000
## tmin_winter -3.37732  0.04346 -77.71633 0.00000
## prcp        -0.06202  0.04924 -1.25946 0.20793
## agdd_winter  0.00123  0.00098  1.25482 0.20961
## prcp_winter  0.00846  0.00647  1.30854 0.19076
## co2_per capita -0.35398  0.06219 -5.69220 0.00000
plot(rq)

```



```
head(round(predict(rq, newdata = test), 0))
```

```
##      tau= 0.25 tau= 0.50 tau= 0.75
## 4        45      52      56
## 7        47      52      56
## 9        45      52      56
## 18       48      54      58
## 24       39      46      51
## 27       37      45      50
```