Cherry Blossom Prediction

Exploratory Data Analysis (EDA)

Alex Yinan Guo

Contents

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
##       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           105            72           127           112
##          long           alt          tmax          tmin          prcp
##           110           106          5417          5429          5177
##   agdd_winter   tmax_winter   tmin_winter   prcp_winter co2_percapita
##          4592          5516          5522          5453           272
```

```
unique(df$country)
```

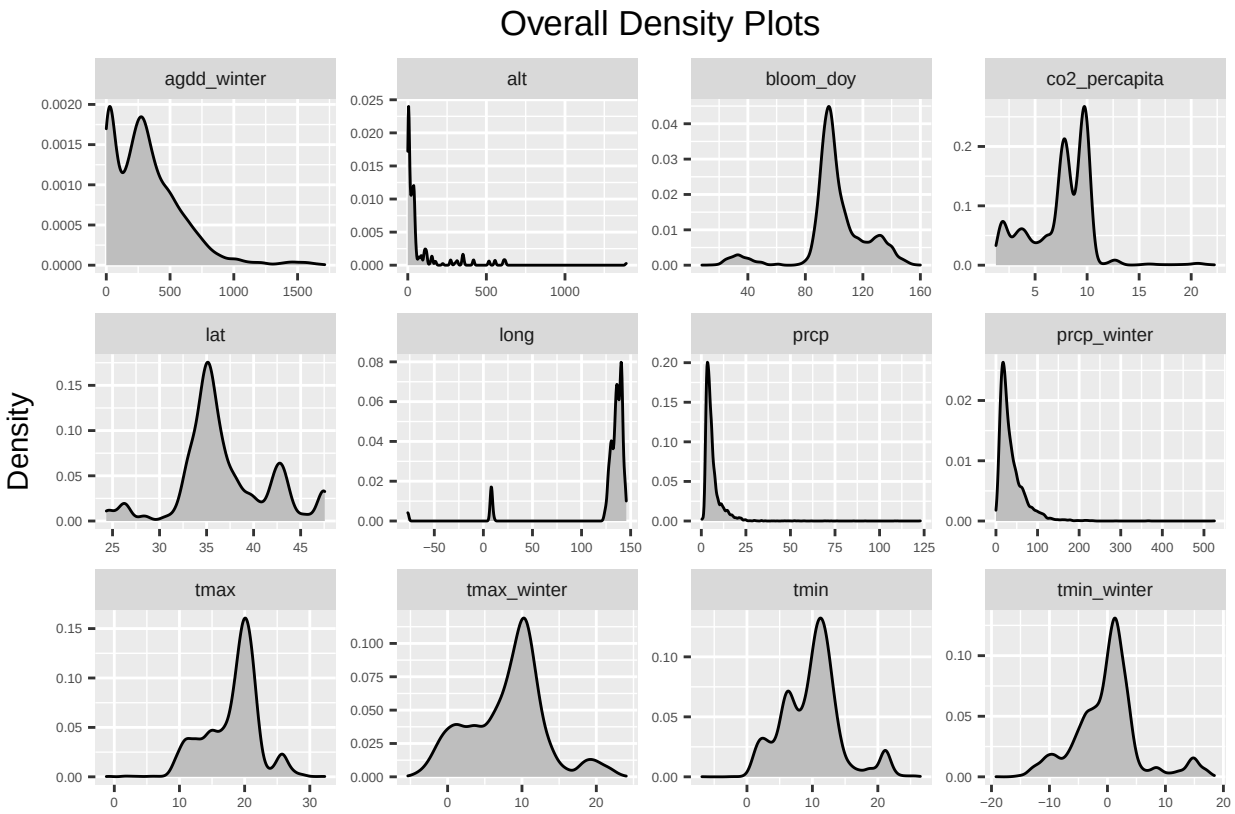
```
## [1] "Japan"         "Switzerland"   "South Korea"   "United States"
```

Density Plots

Overall Density Plots

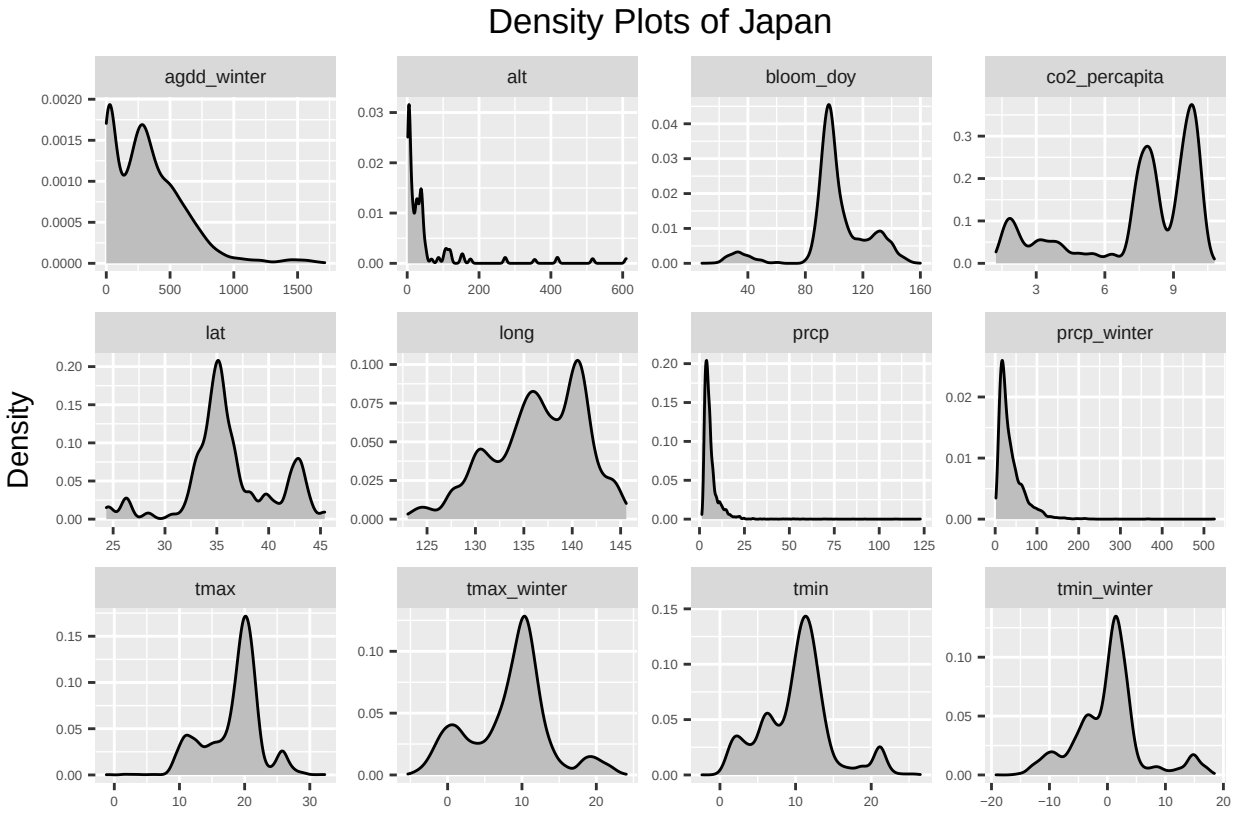
```
df %>%
  select_if(is.numeric) %>%
  dplyr::select(-year) %>%
  pivot_longer(everything()) %>%
  ggplot(aes(x=value)) +
  geom_density(fill='grey') +
  facet_wrap(~name, scales='free') +
  theme(strip.text = element_text(size=7),
        axis.text.x = element_text(size=5),
        axis.text.y = element_text(size=5),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

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



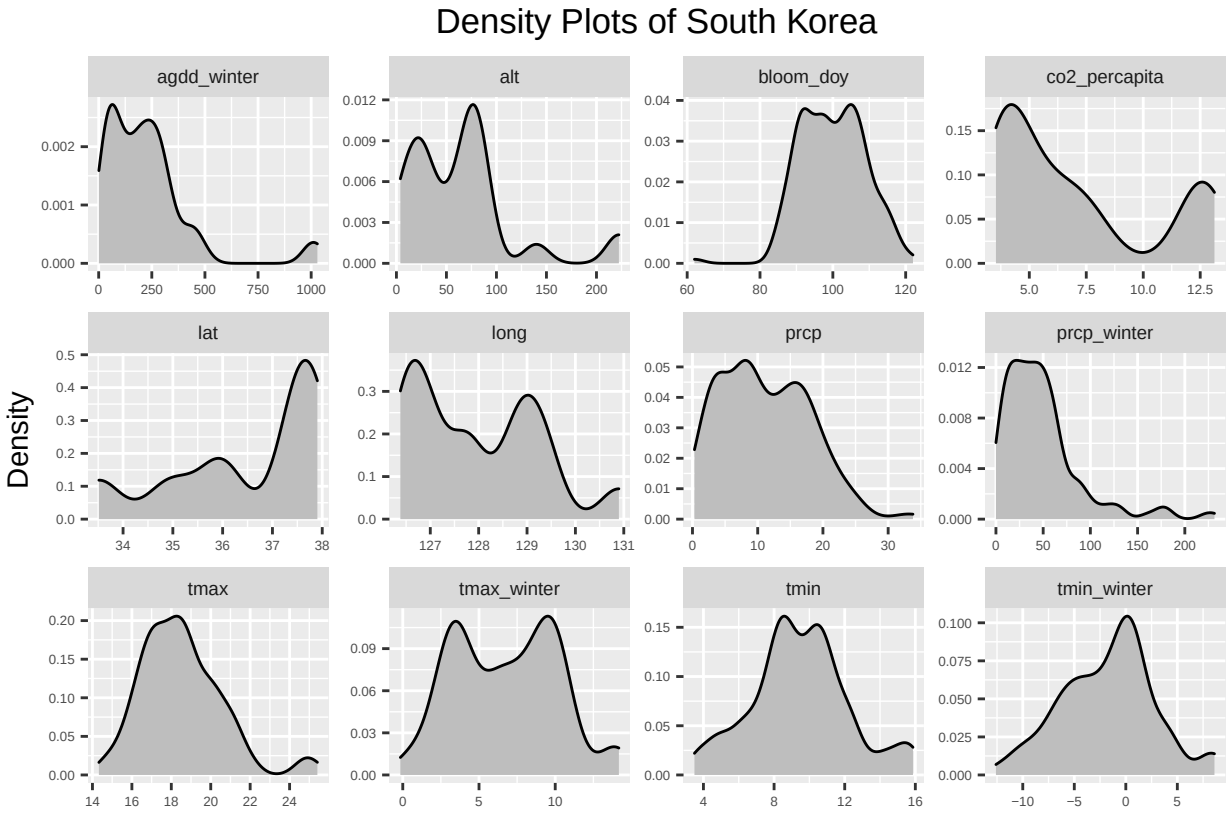
Density Plots of Japan

```
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')
```



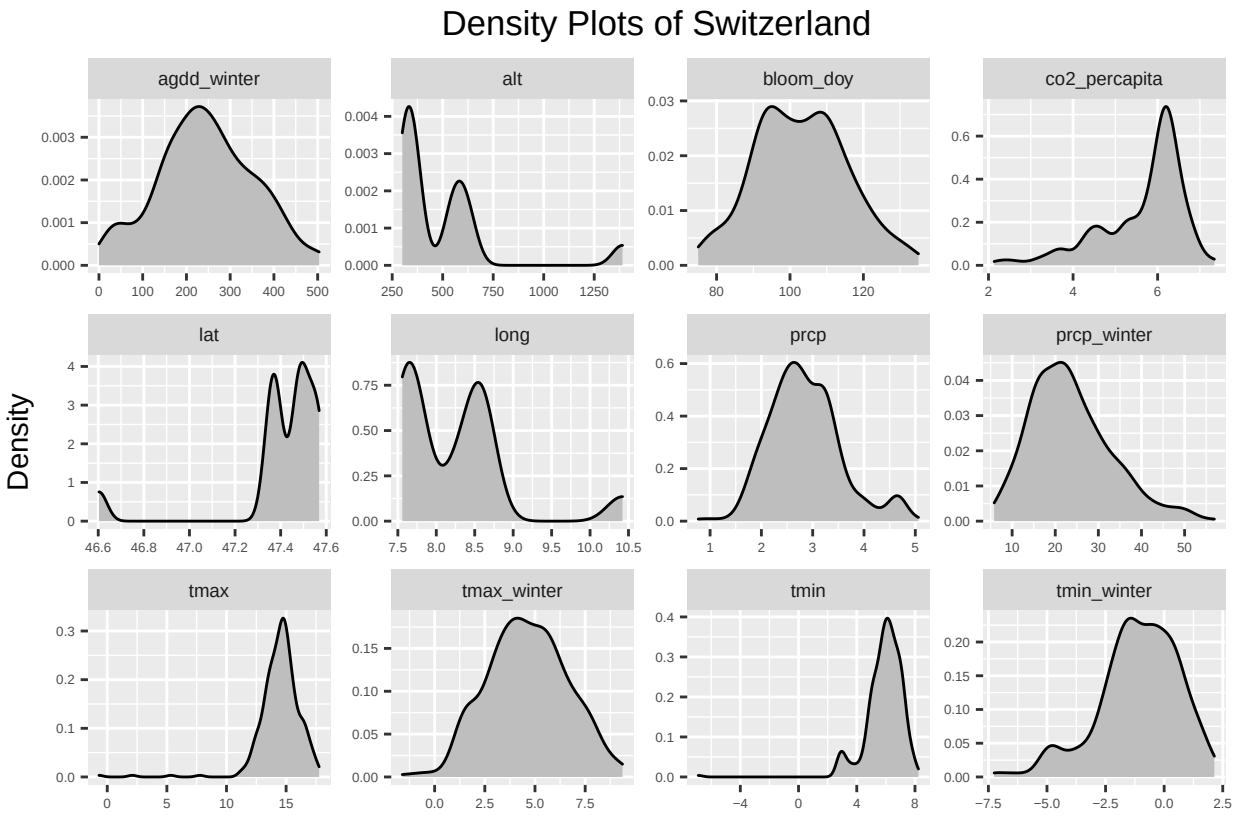
Density Plots of South Korea

```
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')
```



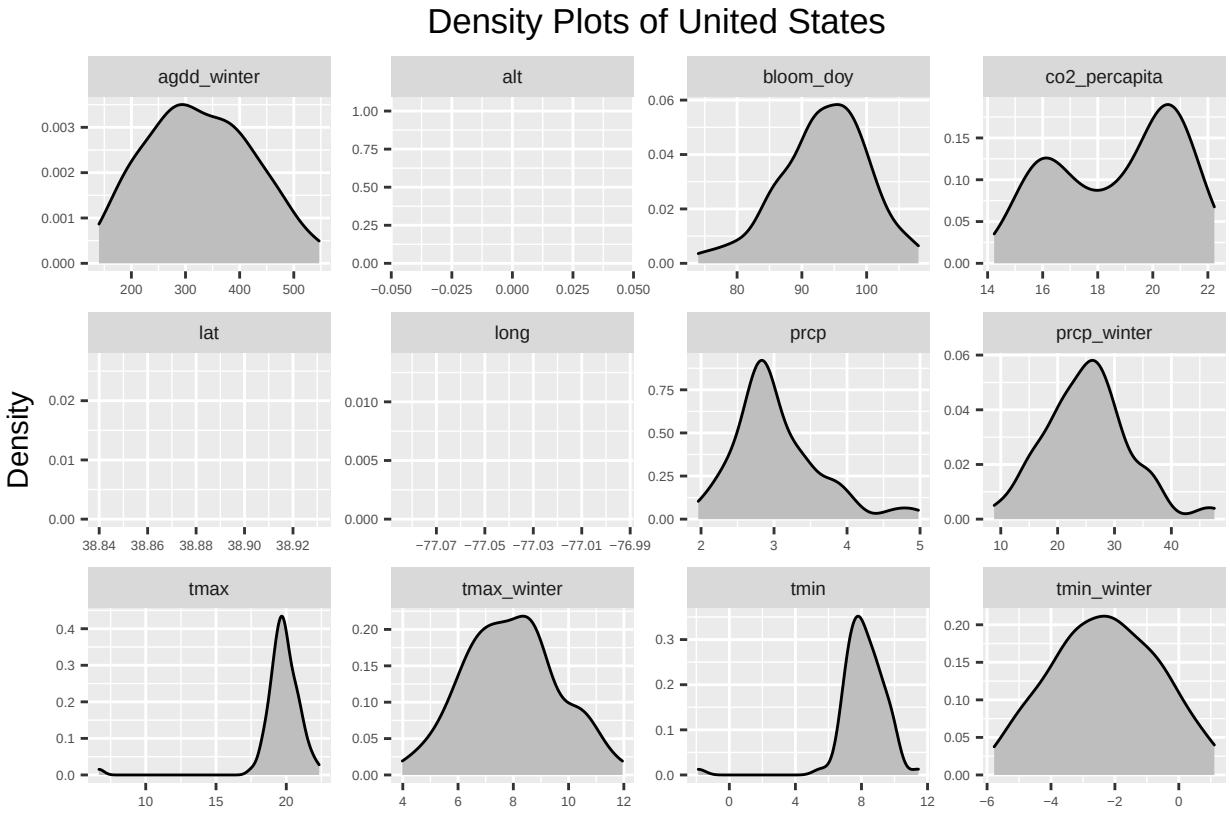
Density Plots of Switzerland

```
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 United States

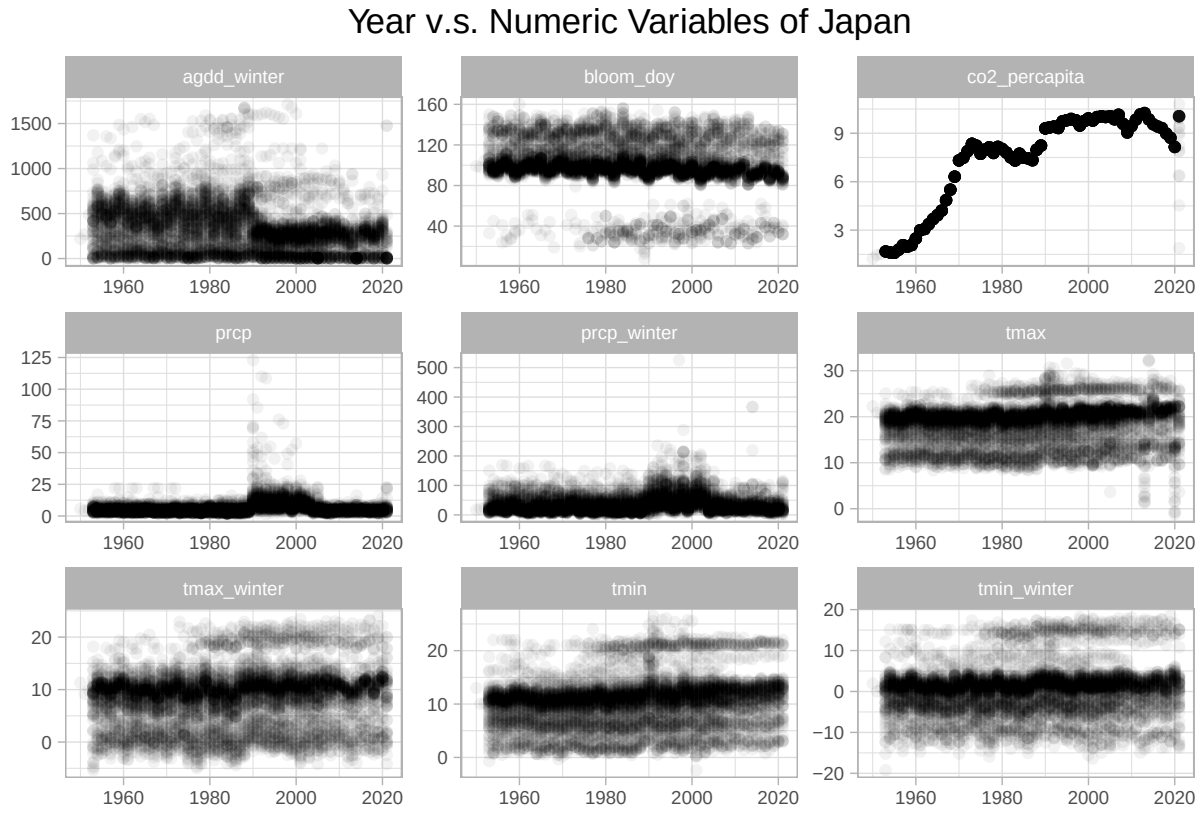
```
df %>%
  filter(country=='United States') %>%
  select_if(is.numeric) %>%
  dplyr::select(-year) %>%
  pivot_longer(everything()) %>%
  ggplot(aes(x=value)) +
  geom_density(fill='grey') +
  facet_wrap(~name, scales='free') +
  theme(strip.text = element_text(size=7),
        axis.text.x = element_text(size=5),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



Year v.s. Numeric Variables

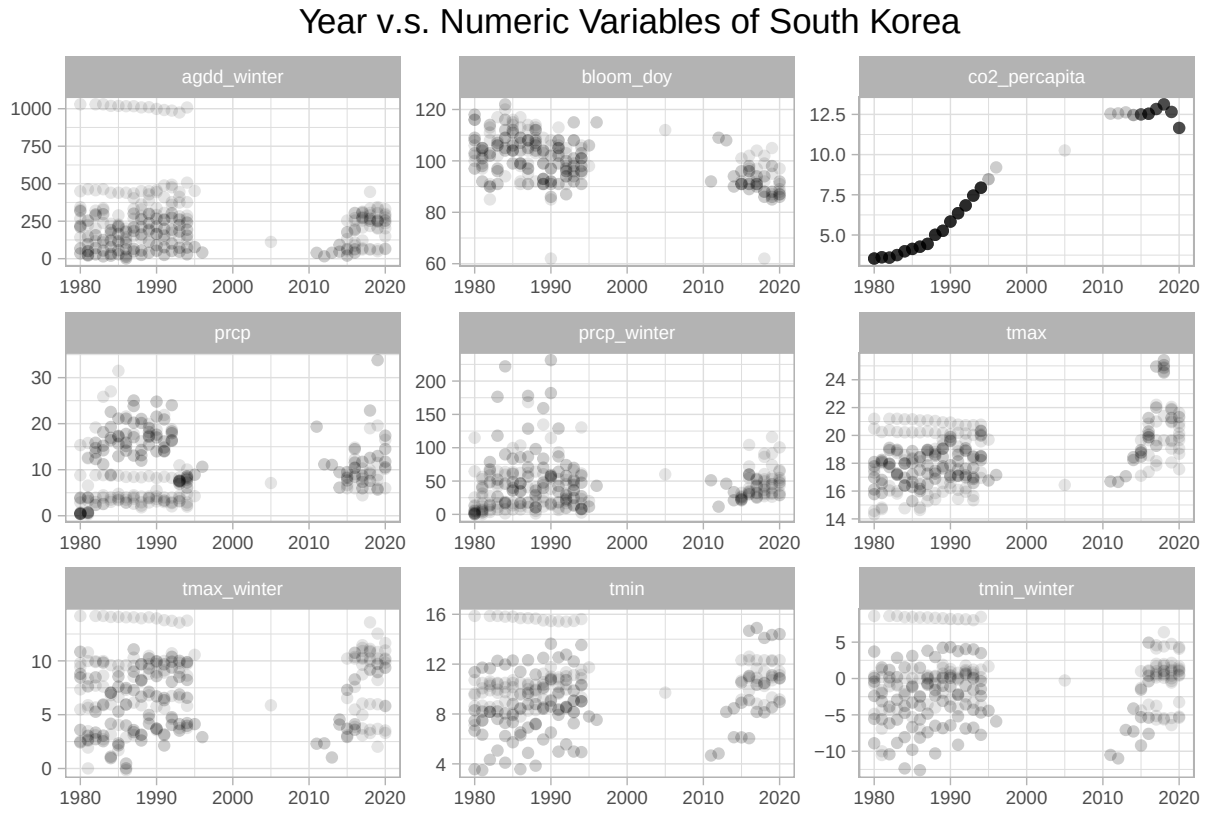
Year v.s. Numeric Variables of Japan

```
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))
```



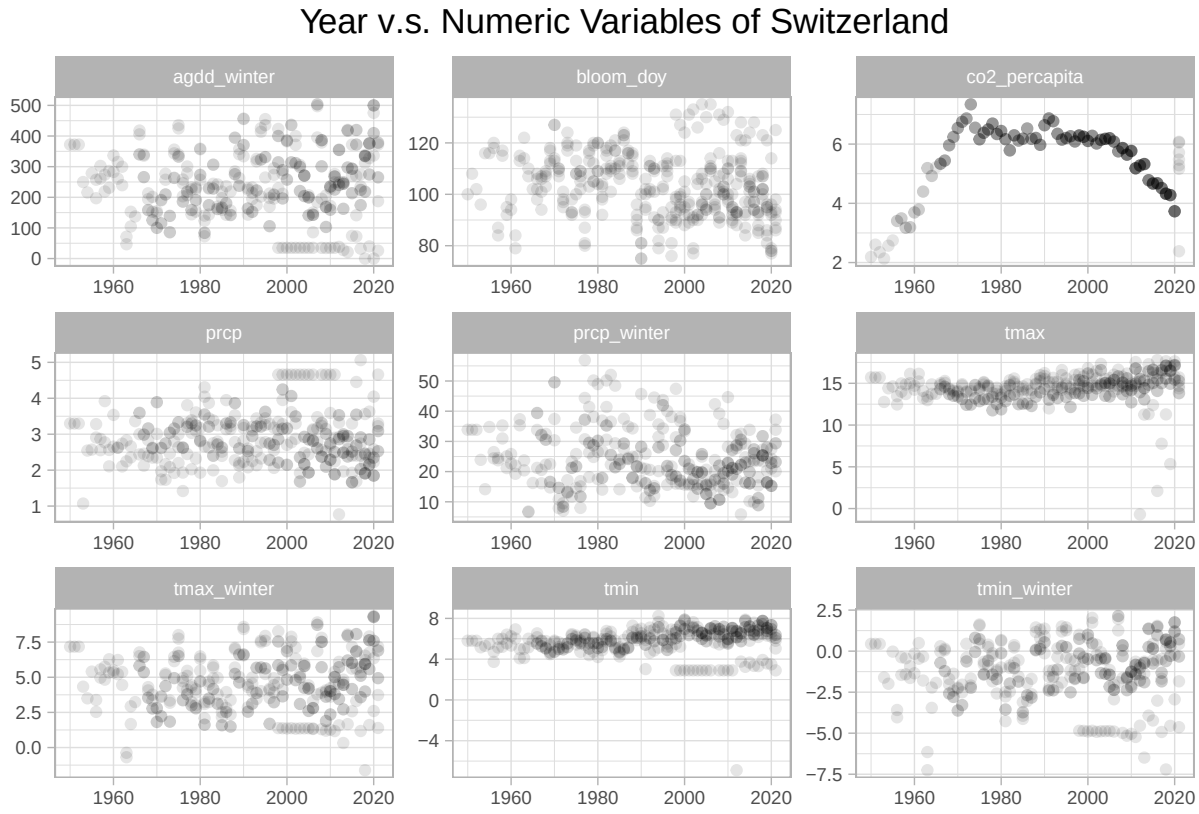
Year v.s. Numeric Variables of South Korea

```
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))
```



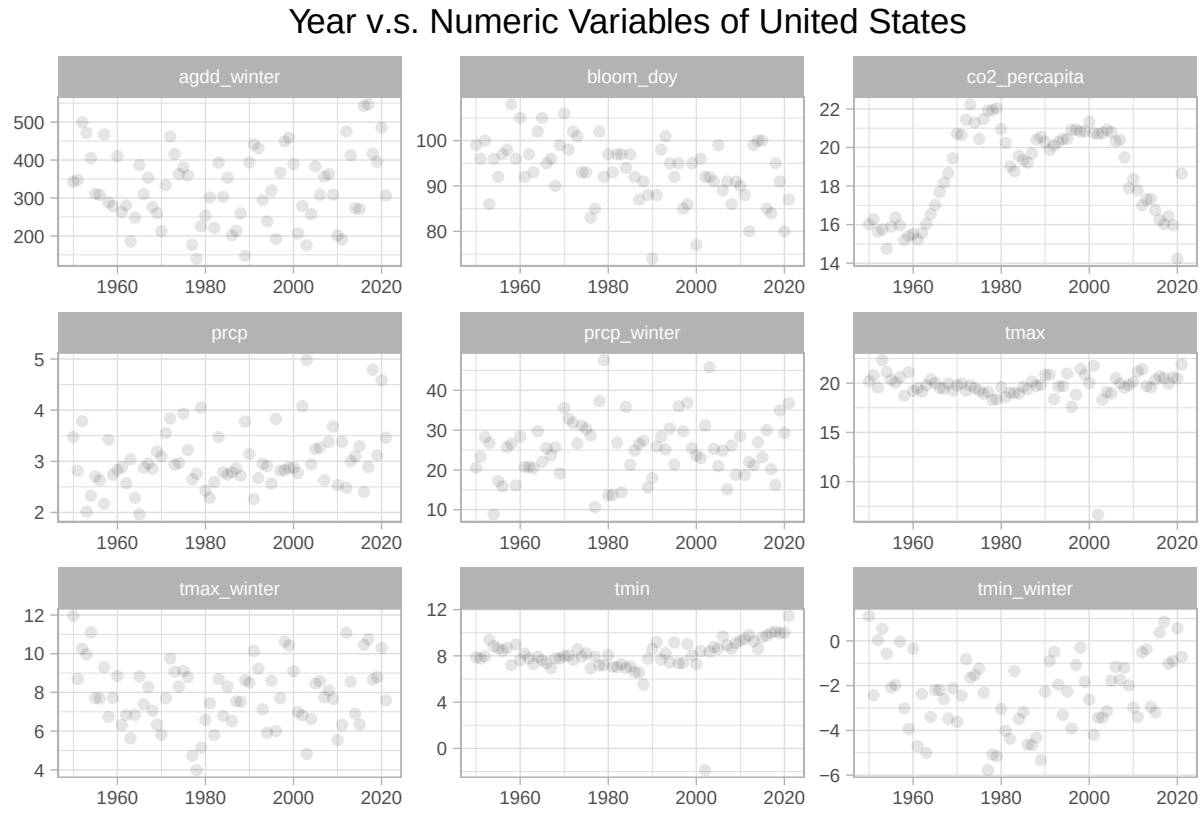
Year v.s. Numeric Variables of Switzerland

```
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 United States

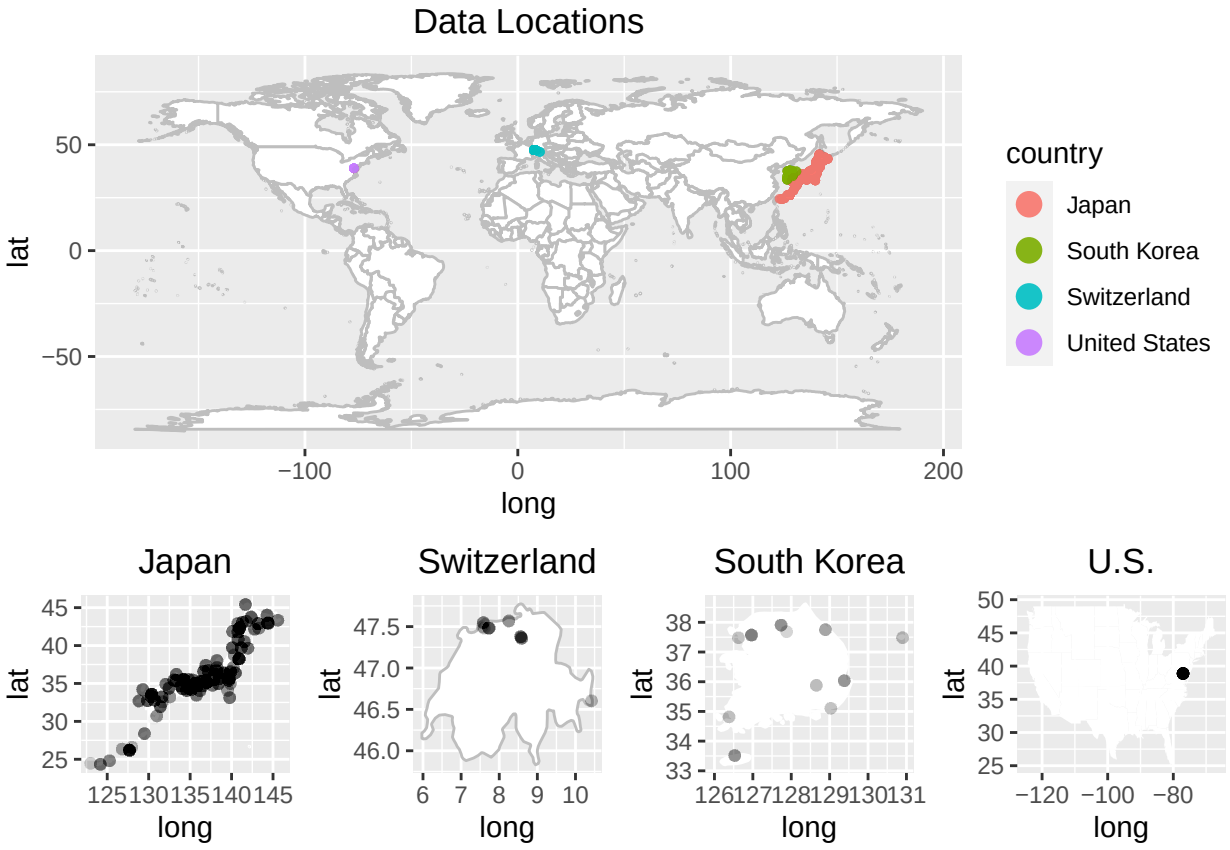
```
par(mfrow=c(3, 4))

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))
```



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))
```



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
```

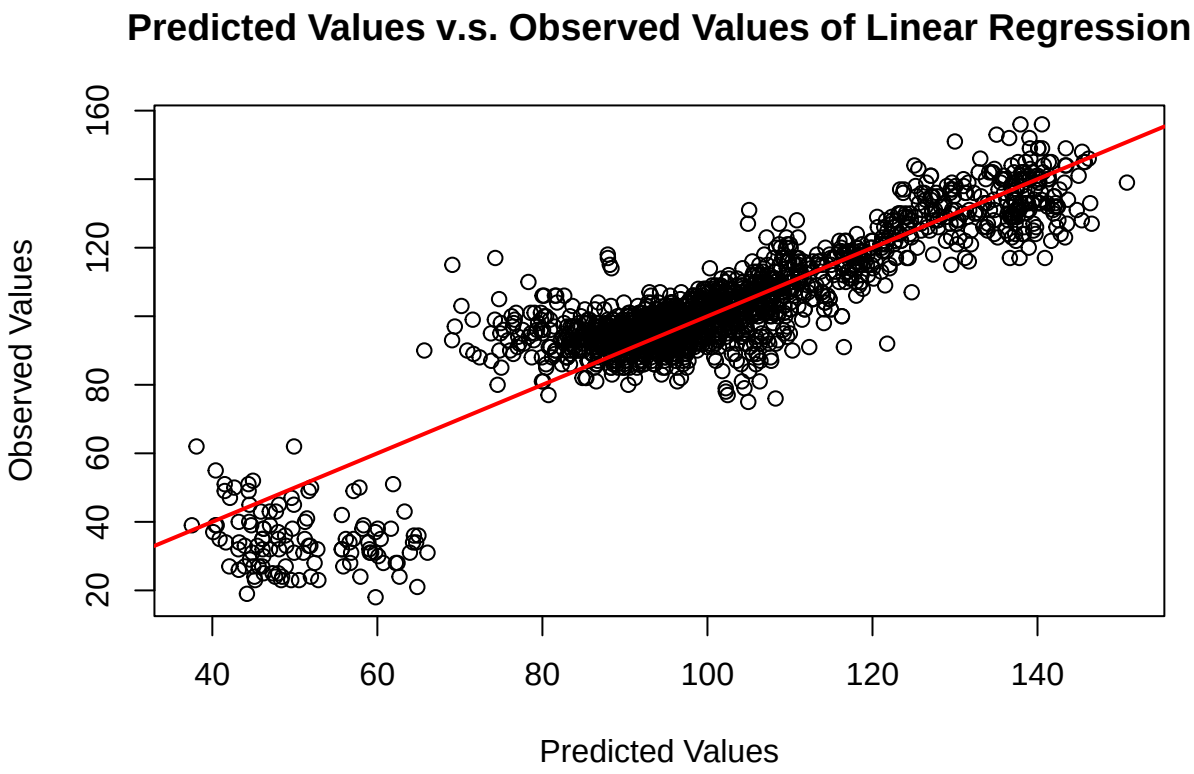
```
df <- df %>% na.omit
train <- df[sort(sample(nrow(df), nrow(df)*.7)),]
test <- df[-sort(sample(nrow(df), nrow(df)*.7)),]
```

Fit Linear Regression

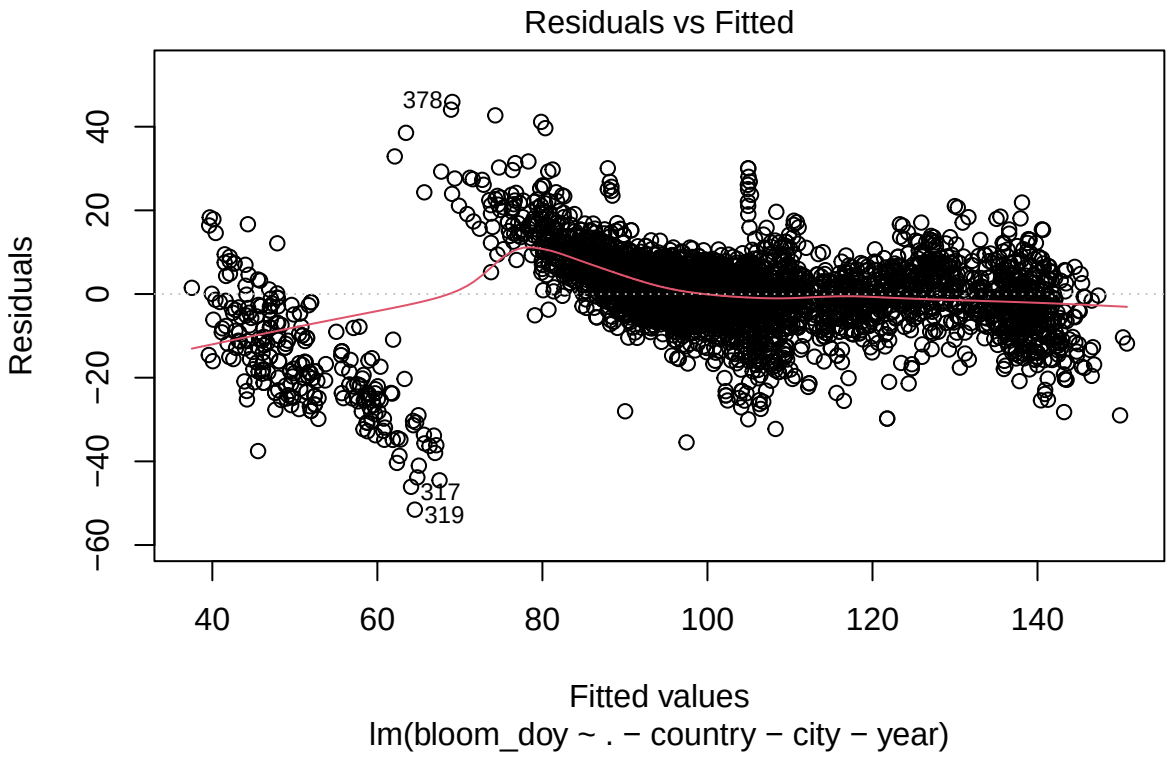
```
lm <- lm(bloom_doy~.-country-city-year, data = train)
summary(lm)
```

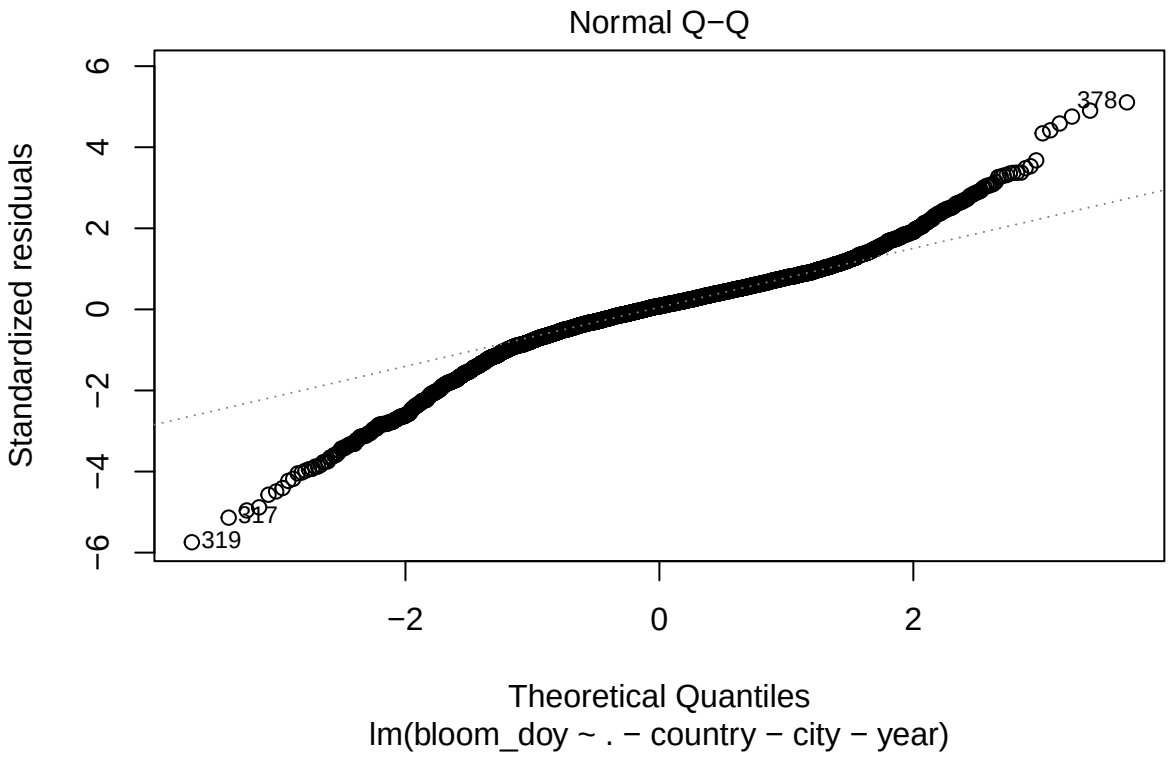
```
##
## Call:
## lm(formula = bloom_doy ~ . - country - city - year, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -51.539  -3.958   0.716   4.867  45.914
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   41.6294759  5.0339233   8.270  < 2e-16 ***
## lat            1.7559479  0.0896389  19.589  < 2e-16 ***
## long           0.1415917  0.0058394  24.247  < 2e-16 ***
## alt           -0.0089106  0.0011216  -7.945 2.46e-15 ***
## tmax          -0.6816203  0.1083703  -6.290 3.50e-10 ***
## tmin          -0.8858588  0.1618398  -5.474 4.66e-08 ***
## prcp           0.1382613  0.0308590   4.480 7.64e-06 ***
## agdd_winter    0.0076526  0.0008727   8.768  < 2e-16 ***
## tmax_winter   -0.5764235  0.1134902  -5.079 3.95e-07 ***
## tmin_winter   -0.9570343  0.1175100  -8.144 4.95e-16 ***
## prcp_winter   -0.0184741  0.0050976  -3.624 0.000293 ***
## co2_percapita  0.0754662  0.0532016   1.418 0.156118
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.998 on 4315 degrees of freedom
## Multiple R-squared:  0.8378, Adjusted R-squared:  0.8374
## F-statistic:  2026 on 11 and 4315 DF,  p-value: < 2.2e-16
```

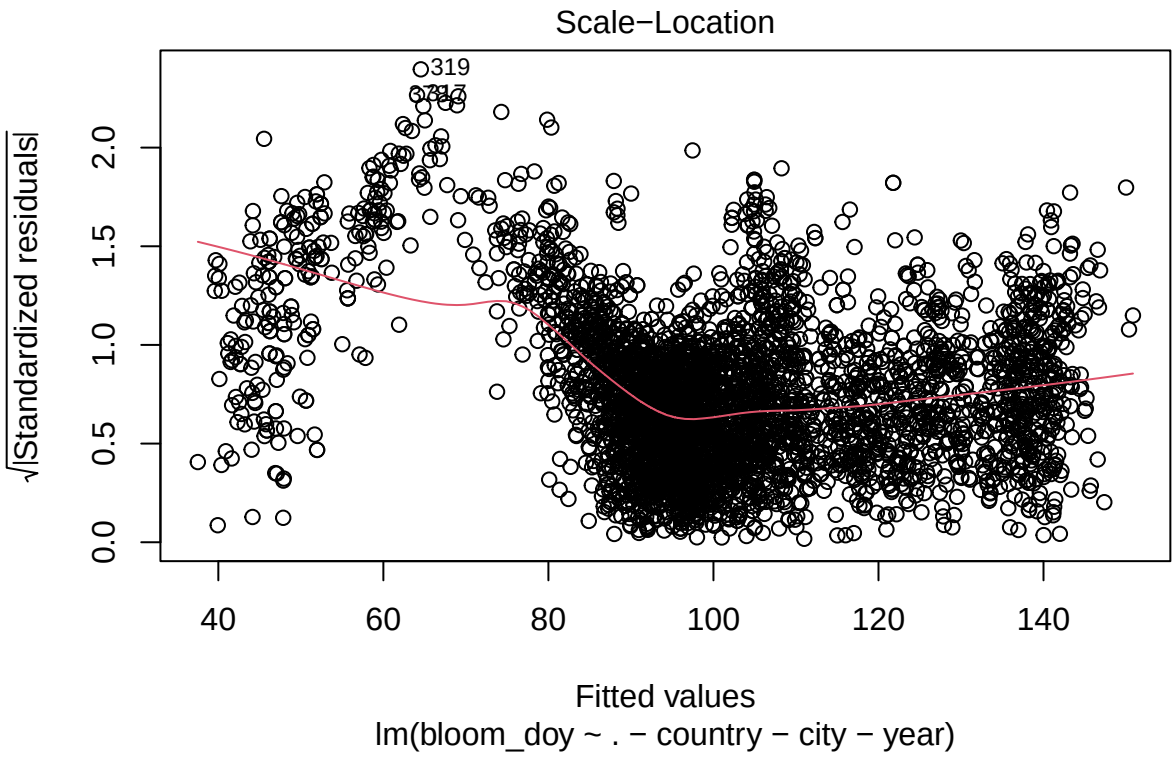
```
plot(predict(lm, test),
     test$bloom_doy,
     xlab = "Predicted Values",
     ylab = "Observed Values",
     main = 'Predicted Values v.s. Observed Values of Linear Regression')
abline(a=0, b=1, lib = 1, col = "red", lwd = 2)
```

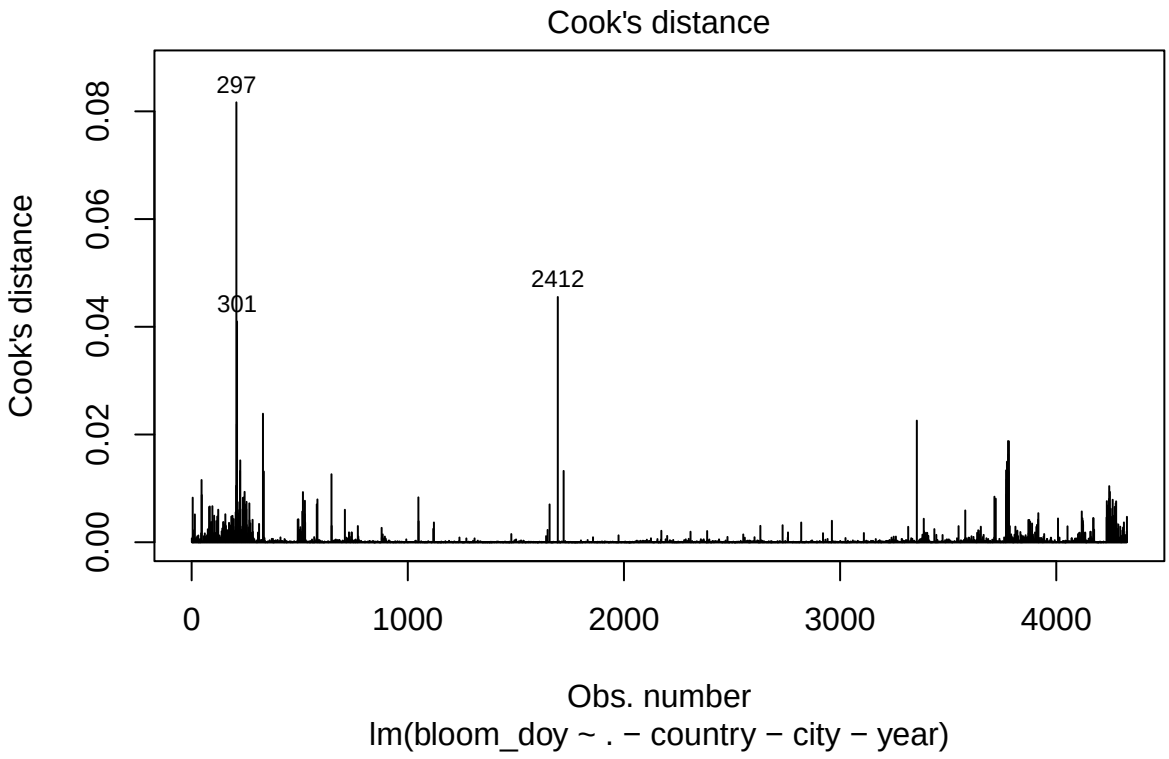


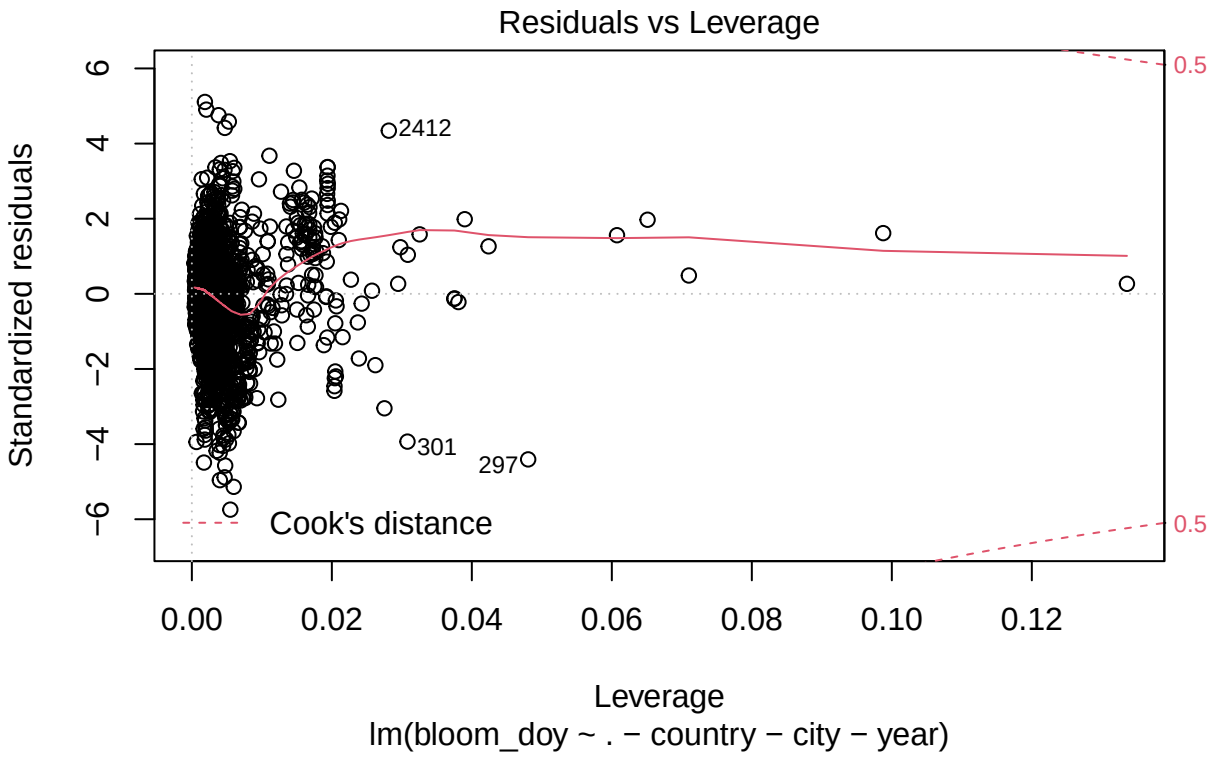

```
plot(lm, which = c(1,2,3,4,5))
```


Residuals vs Fitted
Residuals
Fitted values
lm(bloom_doy ~ . − country − city − year)
378
317
319
40
20
0
−20
−40
−60
40
60
80
100
120
140


Normal Q−Q
Standardized residuals
6
4
2
0
−2
−4
−6
378
317
319
−2
0
2
Theoretical Quantiles
lm(bloom_doy ~ . − country − city − year)


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
319
40
60
80
100
120
140
Fitted values
lm(bloom_doy ~ . − country − city − year)


Cook's distance
297
301
2412
Cook's distance
0.00
0.02
0.04
0.06
0.08
0
1000
2000
3000
4000
Obs. number
lm(bloom_doy ~ . − country − city − year)



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
## 5  44.37566 42.49269 46.25862
## 6  40.11403 38.97251 41.25555
## 7  44.19530 42.23455 46.15605
## 12 43.93673 41.81776 46.05570
## 13 45.45028 44.48447 46.41609
## 14 45.10368 44.07760 46.12977
```

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.

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

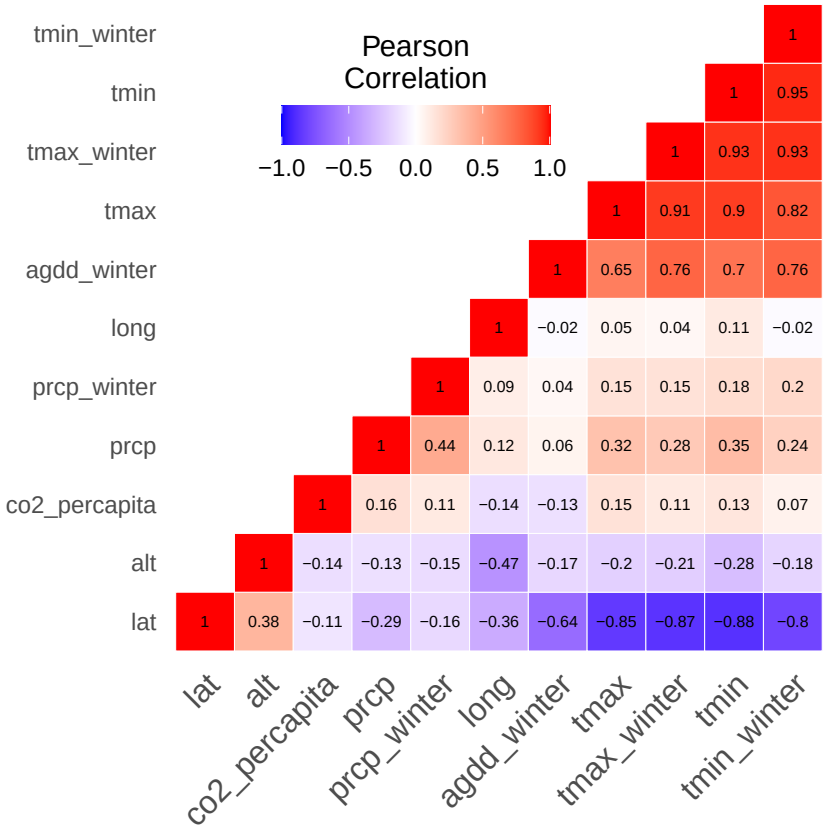
```
##         fit      lwr      upr
## 5  44.37566 26.63548 62.11583
## 6  40.11403 22.43717 57.79089
## 7  44.19530 26.44670 61.94390
## 12 43.93673 26.16996 61.70350
## 13 45.45028 27.78390 63.11666
## 14 45.10368 27.43391 62.77346
```

Check multicollinearity

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

```r
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:     49147.4694         1
## Red Indicator:             0.4686         0
## Sum of Lambda Inverse:   102.3103         1
## Theil's Method:           -1.1834         0
## Condition Number:        129.9377         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             9.8857 0.1012  3835.0896  4262.1980 0.3180 -0.5142     1 0.0002
## long            2.3633 0.4231   588.4170   653.9482 0.6505 -0.1229     0 0.0010
## alt             1.5138 0.6606   221.7738   246.4724 0.8128 -0.0787     0 0.0015
## tmax           10.6871 0.0936  4180.9627  4646.5905 0.3059 -0.5558     1 0.0002
## tmin           28.3946 0.0352 11823.4887 13140.2535 0.1877 -1.4768     1 0.0001
## prcp            1.6972 0.5892   300.9299   334.4440 0.7676 -0.0883     0 0.0014
## agdd_winter     3.0123 0.3320   868.4906   965.2131 0.5762 -0.1567     0 0.0008
## tmax_winter    18.5587 0.0539  7578.3427  8422.3318 0.2321 -0.9652     1 0.0001
## tmin_winter    23.4742 0.0426  9699.8813 10780.1430 0.2064 -1.2209     1 0.0001
## prcp_winter     1.4425 0.6932   190.9820   212.2514 0.8326 -0.0750     0 0.0016
## co2_percapita   1.2807 0.7808   121.1601   134.6535 0.8836 -0.0666     0 0.0018
##                   IND2
## lat             1.3743
## long            0.8820
## alt             0.5190
## tmax            1.3859
## tmin            1.4751
## prcp            0.6281
## agdd_winter     1.0213
## tmax_winter     1.4465
## tmin_winter     1.4638
## prcp_winter     0.4690
## co2_percapita   0.3351
##
## 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: 0.8378
##
## * 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: 27763.09
## Random effects:
##  Groups   Name          Std.Dev.  Corr
##  city     (Intercept)   13.294736
##           tmin_winter    0.970817 -0.16
##           prcp           0.108243 -0.08  0.62
##           agdd_winter    0.006729 -0.03 -0.02  0.72
##           prcp_winter    0.010325 -0.11 -0.53 -0.71 -0.59
##           co2_percapita  0.187241 -0.40 -0.12  0.36  0.77 -0.21
##  Residual                5.489156
## Number of obs: 4327, groups:  city, 105
## Fixed Effects:
##   (Intercept)    tmin_winter           prcp    agdd_winter    prcp_winter
##   100.5226823     -1.5228851     -0.0676715      0.0036726     -0.0007896
## co2_percapita
##    -0.2602755
## optimizer (nloptwrap) convergence code: 0 (OK) ; 0 optimizer warnings; 1 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: 27763.1
```

```
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.7729 -0.5874  0.0258  0.5782  4.1952
##
## Random effects:
##  Groups   Name          Variance     Std.Dev.  Corr
##  city     (Intercept)   176.75000750 13.294736
##           tmin_winter     0.94248531  0.970817 -0.16
##           prcp            0.01171646  0.108243 -0.08  0.62
##           agdd_winter     0.00004527  0.006729 -0.03 -0.02  0.72
##           prcp_winter     0.00010660  0.010325 -0.11 -0.53 -0.71 -0.59
##           co2_percapita   0.03505907  0.187241 -0.40 -0.12  0.36  0.77 -0.21
##  Residual                30.13083625  5.489156
## Number of obs: 4327, groups:  city, 105
##
## Fixed effects:
##                   Estimate  Std. Error t value
## (Intercept)   100.5226823   1.4489979  69.374
## tmin_winter    -1.5228851   0.1322885 -11.512
## prcp           -0.0676715   0.0282823  -2.393
## agdd_winter     0.0036726   0.0011810   3.110
## prcp_winter    -0.0007896   0.0043773  -0.180
## co2_percapita  -0.2602755   0.0433487  -6.004
##
## Correlation of Fixed Effects:
##             (Intr) tmn_wn prcp   agdd_w prcp_w
## tmin_winter -0.015
## prcp        -0.136  0.084
## agdd_winter -0.270 -0.254  0.358
## prcp_winter -0.084 -0.174 -0.452 -0.017
## co2_percapt -0.403 -0.201  0.025  0.488 -0.031
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

```
head(coef(full_mixed_model)$city)
```

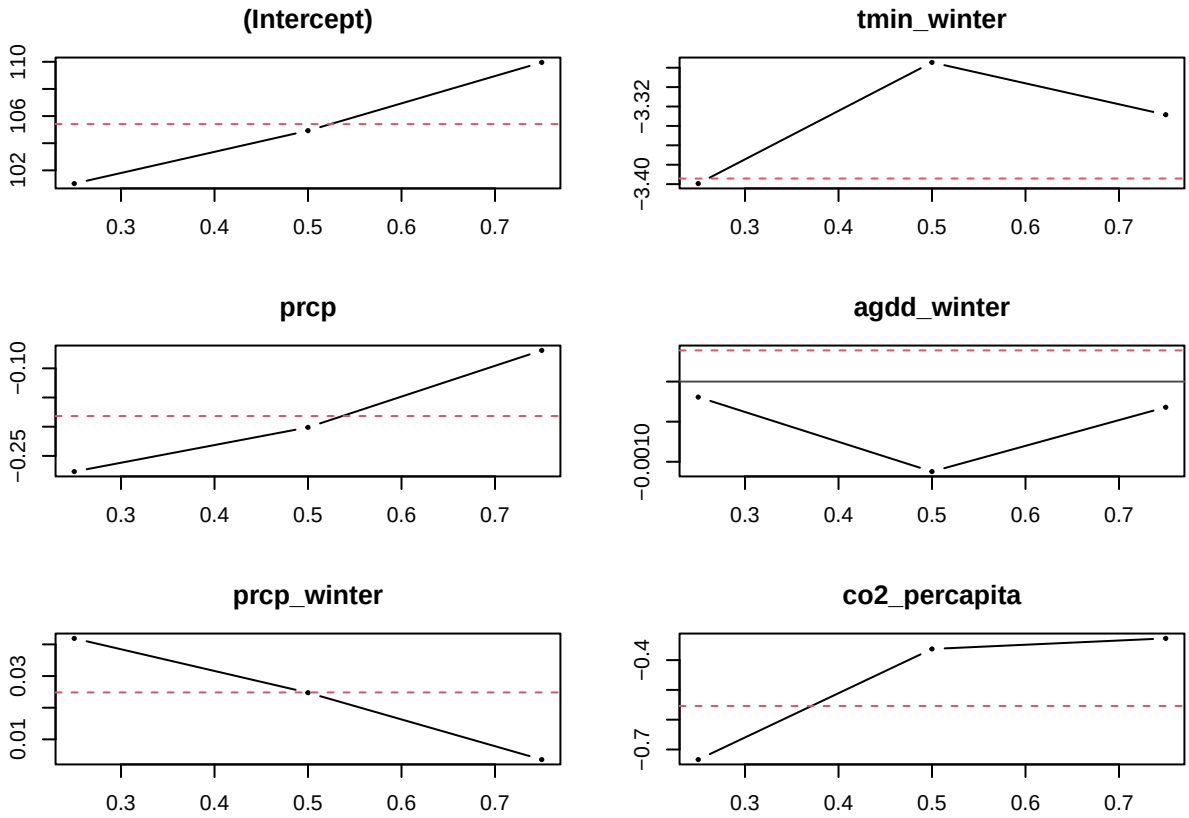
```
##                 (Intercept) tmin_winter        prcp   agdd_winter  prcp_winter
## Abashiri           130.7929  -0.7080442  0.02114797  0.0054337848 -0.011496816
## Aikawa             112.7161  -2.2460968 -0.13577824  0.0031061965  0.003391098
## Akita              111.8598  -1.8788015 -0.09833211  0.0035249784 -0.001283693
## Aomori             111.8213  -2.5727939 -0.13528248  0.0037711756  0.003769242
## Asahikawa          126.9994  -0.6287957 -0.03033859  0.0003235868 -0.007926072
## Basel-Binningen    100.8506  -3.0928900 -0.30015383 -0.0066015042  0.018598767
##                 co2_percapita
## Abashiri           -0.4135017
## Aikawa             -0.2619531
## Akita              -0.3060861
## Aomori             -0.2922221
## Asahikawa          -0.5296348
## Basel-Binningen    -0.4204631
```

```
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_percapita, tau = c(0.25, 0.5, 0.75), data = train)
##
## tau: [1] 0.25
##
## Coefficients:
##               Value     Std. Error t value   Pr(>|t|)
## (Intercept)   101.02559   0.95716  105.54773   0.00000
## tmin_winter    -3.39943   0.07406  -45.90150   0.00000
## prcp           -0.27684   0.06839   -4.04825   0.00005
## agdd_winter    -0.00019   0.00154   -0.12606   0.89969
## prcp_winter     0.04189   0.00755    5.55164   0.00000
## co2_percapita  -0.73384   0.07766   -9.44944   0.00000
##
## Call: rq(formula = bloom_doy ~ tmin_winter + prcp + agdd_winter + prcp_winter +
##     co2_percapita, tau = c(0.25, 0.5, 0.75), data = train)
##
## tau: [1] 0.5
##
## Coefficients:
##               Value     Std. Error t value   Pr(>|t|)
## (Intercept)   104.92458   0.64226  163.36682   0.00000
## tmin_winter    -3.27454   0.04999  -65.50179   0.00000
## prcp           -0.20113   0.06292   -3.19656   0.00140
## agdd_winter    -0.00112   0.00101   -1.11377   0.26544
## prcp_winter     0.02475   0.00677    3.65611   0.00026
## co2_percapita  -0.36258   0.05412   -6.69944   0.00000
##
## Call: rq(formula = bloom_doy ~ tmin_winter + prcp + agdd_winter + prcp_winter +
##     co2_percapita, tau = c(0.25, 0.5, 0.75), data = train)
##
## tau: [1] 0.75
##
## Coefficients:
##               Value     Std. Error t value   Pr(>|t|)
## (Intercept)   109.95463   0.73755  149.08146   0.00000
## tmin_winter    -3.32846   0.04895  -67.99505   0.00000
## prcp           -0.06933   0.05448   -1.27270   0.20319
## agdd_winter    -0.00032   0.00100   -0.31867   0.74999
## prcp_winter     0.00362   0.00632    0.57206   0.56731
## co2_percapita  -0.32717   0.06584   -4.96923   0.00000
```

```
plot(rq)
```




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

```
##    tau= 0.25 tau= 0.50 tau= 0.75
## 5         42        51        56
## 6         36        44        49
## 7         44        52        57
## 12        46        53        58
## 13        42        50        55
## 14        41        49        54
```