Capstone Project - Forecasting Pollination Dates

Tom Gocken

Thursday, March 3, 2016

Introduction

Research sites that develop new plant varieties must forecast when certain seasonal events, such as pollination, will occur. These events drive work timelines and allocation of resources. The timing of pollination depends on 1) planting date, 2) variety maturity, expressed in growing degree units (GDUs) needed for pollination, and 3) how rapidly GDUs accumulate during the growing season. While planting date and variety maturity are known values determined by the researcher, the rate of GDU accumulation depends on conditions that vary by growing season and location.

A regression model was developed to predict GDU accumulation during the growing season at five research sites in the U.S. Midwest. Examples are provided demonstrating how predicted accumulated GDUs can be combined with inputs for planting date and variety maturity to forecast pollination date and to model planting scenarios.

Data Sources

- Environmental data for counties of interest: http://wonder.cdc.gov/EnvironmentalData.html
- County centroid coordinates: https://www.census.gov/geo/maps-data/data/gazetteer.html
- Frost-free growing season length: http://davesgarden.com/guides/freeze-frost-dates/ summarized from http://www.ncdc.noaa.gov/
- Monthly sea surface temperature (SST) data measuring El Nino / La Nina effects: http://www.cpc.noaa.gov/products/analysis_monitoring/ensostuff/ensoyears.shtml

GDU Calculation Method

Growing degree units (GDUs), also known as growing degree days, were calculated by taking the average of the daily maximum and minimum temperatures compared to a base temperature, T_{base}, as follows:

$$\mathbf{GDU = ((T_{max} + T_{min}) / 2) - T_{base}}$$

where T_{max} is equal to the maximum daily temperature but not greater than a defined upper limit and T_{min} is equal to the minimum daily temperature but not less than the base temperature. The upper limit and base in this project were set to 50°F and 86°F (10°C and 50°C), respectively, typical values for corn.

References

- http://en.wikipedia.org/wiki/Growing_degree-day
- http://agron-www.agron.iastate.edu/Courses/agron212/Calculations/GDD.htm

Data Wrangling

```r
setwd("D:/Files/RProjects/springboard-capstone")
airtemp <- read.delim("../data/Air Temperature.txt")
precip <- read.delim("../data/Precipitation.txt")
sunlight <- read.delim("../data/Sunlight.txt")
surfacetemp <- read.delim("../data/Surface Temperature.txt")
particulate <- read.delim("../data/Particulate Matter.txt")
coordinates <- read.delim("../data/County Coordinates.txt")

library(dplyr)
library(tidyr)
library(reshape2)
library(ggplot2)
library(gridExtra)
library(stargazer)

# Convert "Missing" strings to NA:
airtemp <- mutate(airtemp, heat_index =
                    type.convert(as.character(Avg.Daily.Max.Heat.Index..F.),
                                 na.strings = "Missing"))

surfacetemp <- mutate(surfacetemp, day_surface_temp =
                        type.convert(as.character(
                          Avg.Day.Land.Surface.Temperature..F.),
                          na.strings = "Missing"),
                      night_surface_temp = type.convert(as.character(
                        Avg.Night.Land.Surface.Temperature..F.),
                        na.strings = "Missing"))

# Load and reshape monthly SST data measuring El Nino / La Nina effects:
el_nino <- read.csv("../data/el_nino.csv")
el_nino <- rename(el_nino, year = Year, "1" = DJF, "2" = JFM, "3" = FMA,
                         "4" = MAM, "5" = AMJ, "6" = MJJ, "7" = JJA, "8" = JAS,
                         "9" = ASO, "10" = SON, "11" = OND, "12" = NDJ)
el_nino_tidy <- tidyr::gather(el_nino, "month", "sst", 2:13)
el_nino_tidy$month <- as.integer(el_nino_tidy$month)

# Add 3 and 6 month lag variables:
el_nino_lag <- mutate(el_nino_tidy,
                             yr_prior3mo = as.integer(ifelse(month < 10, year, year + 1)),
                             mo_prior3mo = as.integer(ifelse(month < 10, month + 3, month - 9)),
                             yr_prior6mo = as.integer(ifelse(month < 7, year, year + 1)),
                             mo_prior6mo = as.integer(ifelse(month < 7, month + 6, month - 6)))
el_nino_prior3mo <- select(el_nino_lag, year = yr_prior3mo, month = mo_prior3mo, sst_prior3mo = sst)
el_nino_prior6mo <- select(el_nino_lag, year = yr_prior6mo, month = mo_prior6mo, sst_prior6mo = sst)
el_nino_all <- left_join(el_nino_tidy, el_nino_prior3mo)
el_nino_all <- left_join(el_nino_all, el_nino_prior6mo)
el_nino_all <- arrange(el_nino_all, year, month)

# Join data into a single tidy dataset:
joindat <- left_join(airtemp, precip)
joindat <- left_join(joindat, sunlight)
joindat <- left_join(joindat, surfacetemp)
joindat <- left_join(joindat, particulate)
```

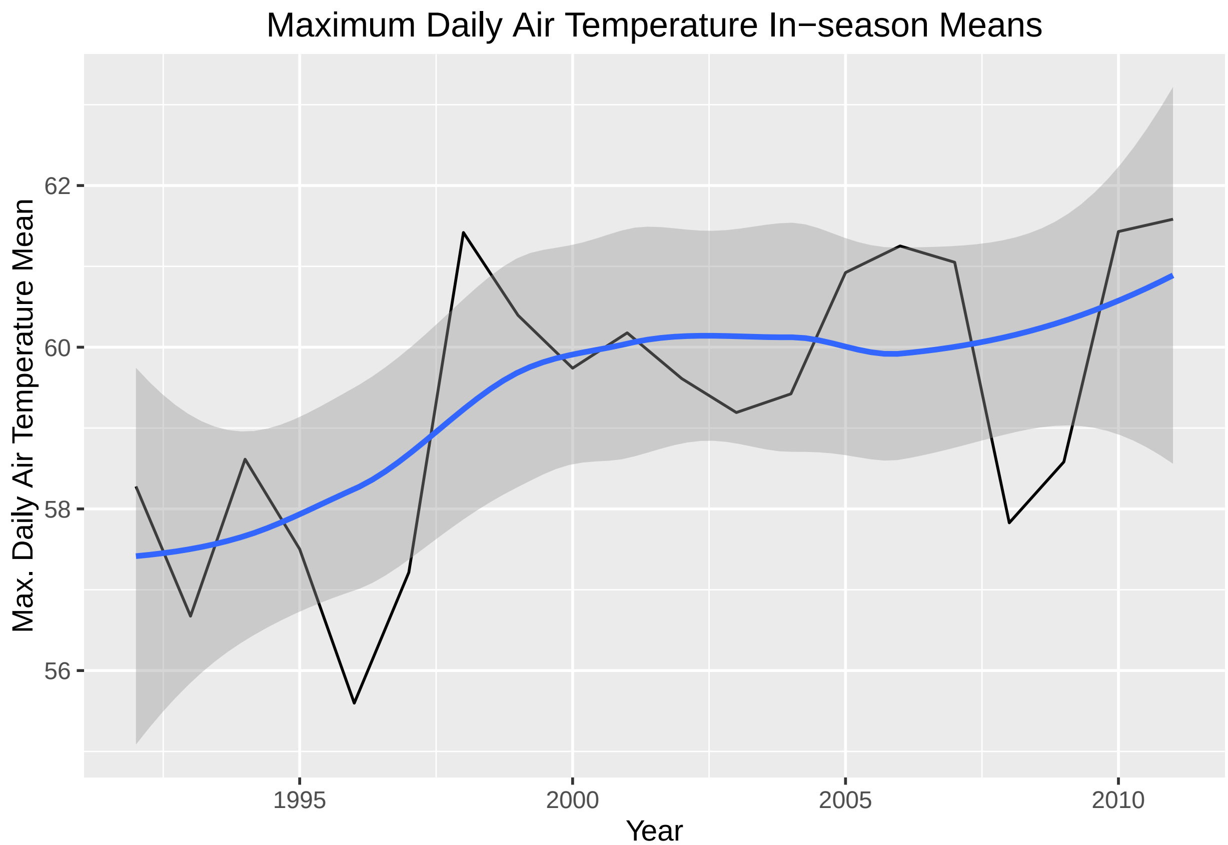
```r
joindat <- left_join(joindat, coordinates)
joindat <- left_join(joindat, el_nino_all, by = c("Year" = "year", "Month.Code" = "month"))

joindat <- mutate(joindat, date = as.Date(paste(joindat$Year.Code,
                                                joindat$Month.Code,
                                                joindat$Day.of.Month.Code,
                                                sep="-")))
envdat <- select(joindat,
                 county = County,
                 latitude = Latitude,
                 longitude = Longitude,
                 grow_season = Frost.Free.Growing.Season,
                 year = Year,
                 month = Month.Code,
                 day_of_yr = Day.of.Year,
                 date,
                 max_air_temp = Avg.Daily.Max.Air.Temperature..F.,
                 min_air_temp = Avg.Daily.Min.Air.Temperature..F.,
                 heat_index,
                 precip = Avg.Daily.Precipitation..mm.,
                 sunlight = Avg.Daily.Sunlight..KJ.m².,
                 day_surface_temp,
                 night_surface_temp,
                 particulate_matter = Avg.Fine.Particulate.Matter..µg.m³.,
                 sst,
                 sst_prior3mo,
                 sst_prior6mo
                 )

# Growing degree unit (GDU) calculation:
envdat <- mutate(envdat, gdu = ifelse(max_air_temp < 50, 0,
                 (((ifelse(max_air_temp > 86, 86, max_air_temp)
                 + ifelse(min_air_temp < 50, 50, min_air_temp)) / 2) - 50)))

envdat <- transform(envdat, agdu = ave(gdu, paste(county, year),
                                       FUN = cumsum))

envdat_inseason <- subset(envdat, day_of_yr >= 90 & day_of_yr < 300)
envdat_train <- subset(envdat_inseason, year < 2010)
envdat_test <- subset(envdat_inseason, year >= 2010)
```

Data Characterization

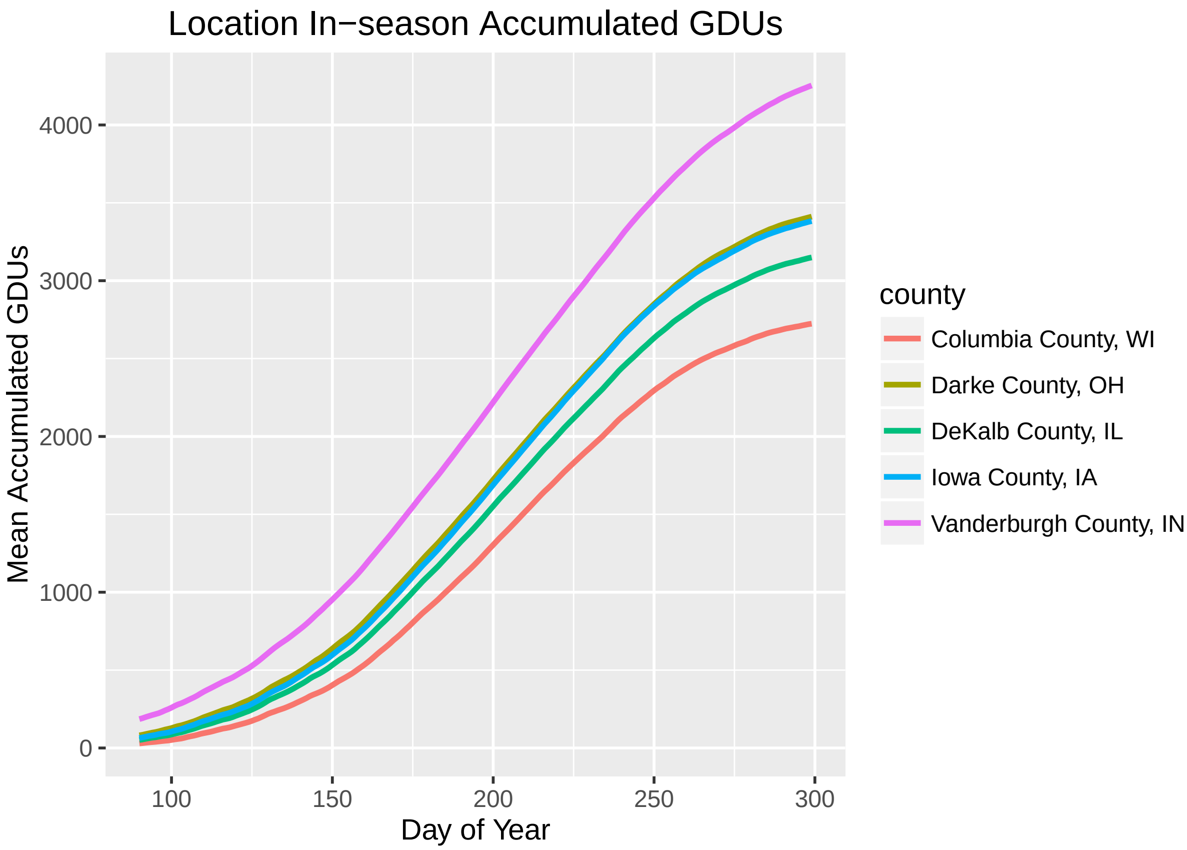
Preliminary exploration of the variables showed a possible upward trend in temperatures over the 20 year period and clear location differences for the rate of GDU accumulation during the growing season.

```r
envdat_by_year <- envdat %>%
  group_by(year) %>%
  summarize(max_air_temp_mean = mean(max_air_temp),
            n = n()) %>%
  arrange(year)
```

```
ggplot(aes(x = year, y = max_air_temp_mean), data = envdat_by_year) +
  geom_line() + geom_smooth() +
  labs(x = "Year", y = "Max. Daily Air Temperature Mean", title = "Maximum Daily Air Temperature In-seas
```




```
county_means <- envdat_inseason %>%
  group_by(county, day_of_yr) %>%
  summarize(agdu_mean = mean(agdu))


ggplot(aes(x = day_of_yr, y = agdu_mean), data = county_means) +
  geom_line(aes(color = county), size = 1.0) +
  labs(x = "Day of Year", y = "Mean Accumulated GDUs", title = "Location In-season Accumulated GDUs")
```

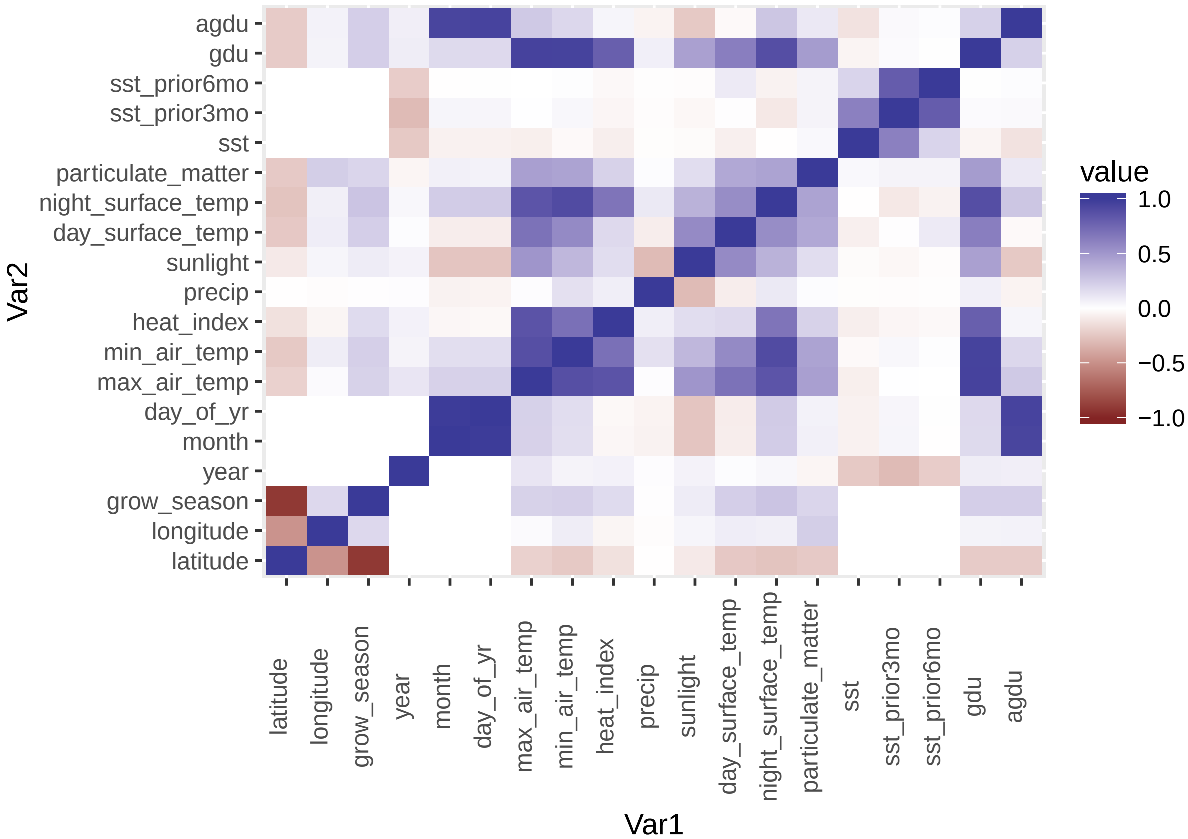


Correlations

Below is a heat map showing correlations among all numeric variables and a table showing correlation coefficients greater than 0.3. As expected, high correlations exist between variables that measure similar things, such as sst, sst_prior3mo, and sst_prior6mo measuring recent El Nino effects; gdu, max_air_temp, and min_air_temp measuring daily temperatures; and day_of_year and month measuring time of year.

Even though latitude and longitude both measure geographical position, their correlation (-0.48) is an artifact of the data. Since latitude measures North/South direction and longitude measures East/West direction, the two variables are expected to be uncorrelated in a random selection of locations. The high correlation between grow_season (frost-free growing season length) and latitude (-0.91) is expected since growing season length is directly affected by distance from the equator.

```r
# Heat map of correlation matrix
corplot1 <- qplot(x=Var1, y=Var2, data=melt(cor(
  select(envdat_inseason, -county, -date), use="p")), fill=value, geom="tile") +
  scale_fill_gradient2(limits=c(-1, 1)) # create heatmap
corplot2 <- corplot1 +
  theme(axis.text.x=element_text(angle = 90, vjust = 0)) # change label orientation
print(corplot2)
```




```r
# Highest correlations
melt(cor(select(envdat_inseason, -county, -date))) %>% # all numeric variables
  rename(Corr_Coeff = value) %>%
  filter(abs(Corr_Coeff) > 0.3 & Corr_Coeff != 1) %>%
  arrange(as.character(Var1), as.character(Var2))
```

```
##              Var1         Var2 Corr_Coeff
## 1            agdu     day_of_yr  0.9465124
## 2            agdu         month  0.9384011
## 3       day_of_yr          agdu  0.9465124
## 4       day_of_yr         month  0.9895758
## 5             gdu  max_air_temp  0.9517451
## 6             gdu  min_air_temp  0.9494270
## 7             gdu      sunlight  0.4534379
## 8     grow_season      latitude -0.9119482
## 9        latitude   grow_season -0.9119482
## 10       latitude     longitude -0.4849110
## 11      longitude      latitude -0.4849110
## 12 max_air_temp             gdu  0.9517451
## 13 max_air_temp min_air_temp  0.8814343
## 14 max_air_temp        sunlight  0.5095273
## 15 min_air_temp             gdu  0.9494270
## 16 min_air_temp max_air_temp  0.8814343
## 17 min_air_temp        sunlight  0.3383542
## 18          month          agdu  0.9384011
## 19          month     day_of_yr  0.9895758
```

```
## 20       precip     sunlight -0.3021674
## 21          sst sst_prior3mo  0.6147758
## 22 sst_prior3mo          sst  0.6147758
## 23 sst_prior3mo sst_prior6mo  0.8060249
## 24 sst_prior3mo         year -0.3023568
## 25 sst_prior6mo sst_prior3mo  0.8060249
## 26     sunlight          gdu  0.4534379
## 27     sunlight max_air_temp  0.5095273
## 28     sunlight min_air_temp  0.3383542
## 29     sunlight       precip -0.3021674
## 30         year sst_prior3mo -0.3023568
```

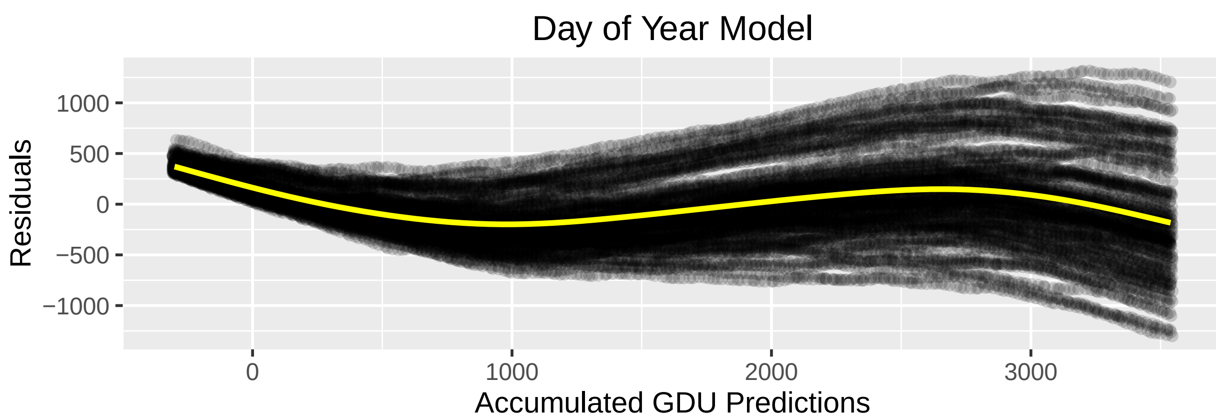
Model Building

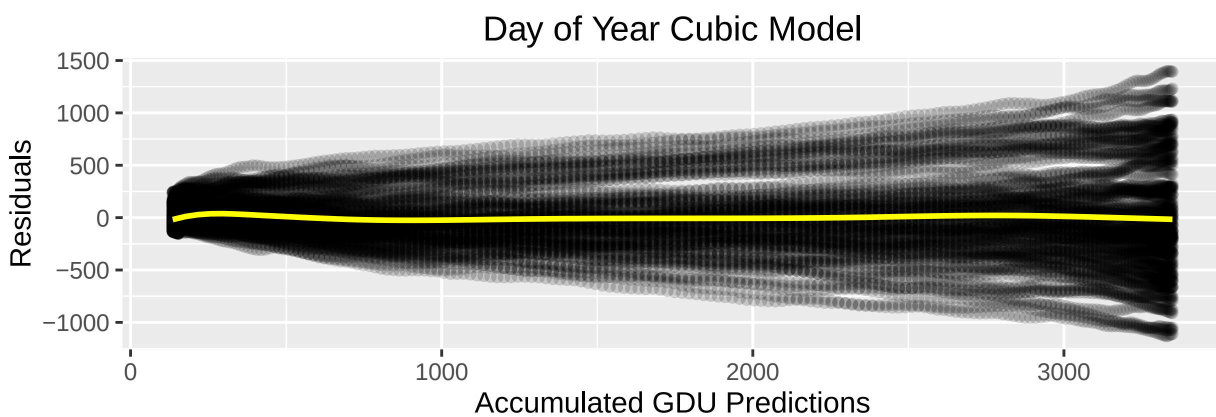
Day of Year

The strongest correlation between response variable agdu and potential predictor variables was with day_of_year (0.95). However, this relationship is known to be non-linear since GDUs accumulate more slowly during cool days in the early spring and late fall than they do during hot days in the summer. To examine the relationship further, a simple regression model was considered for only day_of_yr. The 20 year dataset was split into a training set consisting of data from 1992 through 2009 and a test set with data from 2010 through 2011.

Plotting predicted values versus residuals from the training dataset shows a non-linear distribution with heteroscedasticity or non-constant variances in the errors. A cubic polynomial model addresses the issue of non-linearity but the funnel shaped distribution of residuals shows that heteroscedasticity remains.

```r
lm_doy <- lm(agdu ~ day_of_yr, data = envdat_train)
lm_doy3 <- lm(agdu ~ poly(day_of_yr, 3), data = envdat_train)

plot_doy <- ggplot(aes(x = predict(lm_doy), y = residuals(lm_doy)),
                   data = envdat_train) + geom_jitter(alpha = 1/6) +
                   geom_smooth(color = 'yellow') +
                   labs(x = "Accumulated GDU Predictions", y = "Residuals",
                        title = "Day of Year Model")

plot_doy3 <- ggplot(aes(x = predict(lm_doy3), y = residuals(lm_doy3)),
                    data = envdat_train) + geom_jitter(alpha = 1/6) +
                    geom_smooth(color = 'yellow') +
                    labs(x = "Accumulated GDU Predictions", y = "Residuals",
                         title = "Day of Year Cubic Model")

grid.arrange(plot_doy, plot_doy3, ncol = 1)
```

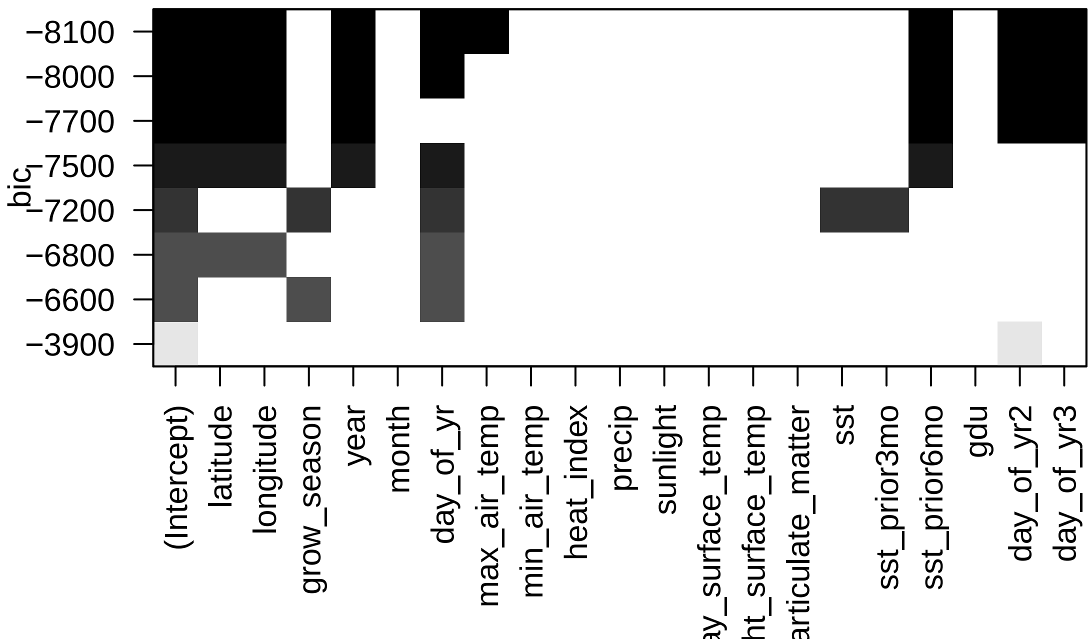




Other Terms

In addition to Day of Year, other variables were considered using the leaps package. All possible combinations were of numeric variables were considered to predict the square root of agdu for the training dataset. The best model for each subset size is plotted below, starting with the best 1 predictor model (excluding the intercept) at the bottom to the best 8 variable model at the top.

```
library (leaps)
envdat_train2 <- select(envdat_train, -county, -date)
envdat_train2$day_of_yr2 <- envdat_train2$day_of_yr^2
envdat_train2$day_of_yr3 <- envdat_train2$day_of_yr^3

models <- regsubsets(agdu ~ . , nbest = 1, data = envdat_train2)
plot(models, scale = "bic") # Bayesian Information Criterion
```



Note that this type of model fitting isn't ideal for the previously described polynomial variables for Day of Year (day_of_yr, day_of_yr2, and day_of_yr3) since they are considered independently but we are interested in their combined effect. Even so, it provides a good indication of the overall combination of variables that will best predict agdu.

The terms selected for the model are the cubic polynomial predictors for day of year, latitude, longitude, year, and sst_prior6mo (El Nino effects from 6 months prior). These terms correspond to the best 7 predictor model. Additional terms provide little additional improvement and risk overfitting.

Robust Residual Standard Errors

As described earlier, heteroscedasticity was observed when examining the residuals for day of year. To account for the heteroscedasticity, the sandwich package was used to calculate robust residual standard errors (RSEs).

Robust RSEs were slightly higher for some variables than standard RSEs but were much smaller than the model estimates in all cases (< 4%) indicating a good fit to the data.

```
library(sandwich)
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
```

```
##
##     as.Date, as.Date.numeric
```

```
sel_model <- lm(agdu ~ poly(day_of_yr, 3) + latitude + longitude
          + year + sst_prior6mo, data = envdat_train)
R.vcov <- vcovHC(sel_model, type="HC0") # variance covariance heteroscedastic consistent matrix
RSE <- sqrt(diag(R.vcov)) # square root of diagonal = residual standard error
R.test <- coeftest(sel_model, R.vcov)
stargazer(sel_model, R.test, column.labels = c("","Robust RSE"),
          no.space=TRUE, type="text") # comparison of std RSEs to robust RSEs
```

```
##
## ================================================================
##                                  Dependent variable:
##                      -------------------------------------------
##                                  agdu
##                                   OLS              coefficient
##                                                        test
##                                                     Robust RSE
##                                   (1)                 (2)
## ----------------------------------------------------------------
## poly(day_of_yr, 3)1        153,140.600***      153,140.600***
##                               (196.385)           (239.559)
## poly(day_of_yr, 3)2         8,102.320***        8,102.320***
##                               (196.470)           (224.732)
## poly(day_of_yr, 3)3        -17,278.860***      -17,278.860***
##                               (196.384)           (225.433)
## latitude                     -168.649***         -168.649***
##                                (0.886)             (1.112)
## longitude                    -32.912***          -32.912***
##                                (0.675)             (0.516)
## year                          14.689***           14.689***
##                                (0.285)             (0.299)
## sst_prior6mo                  41.467***           41.467***
##                                (1.648)             (1.529)
## Constant                   -23,757.050***      -23,757.050***
##                               (573.244)           (595.597)
## ----------------------------------------------------------------
## Observations                   18,900
## R2                              0.972
## Adjusted R2                     0.972
## Residual Std. Error      196.369 (df = 18892)
## F Statistic          94,096.610*** (df = 7; 18892)
## ================================================================
## Note:                                *p<0.1; **p<0.05; ***p<0.01
```

Prediction Scenarios

As a researcher, the practical value of pollination date prediction is to model different planting scenarios and make informed resourcing decisions. Below are examples.

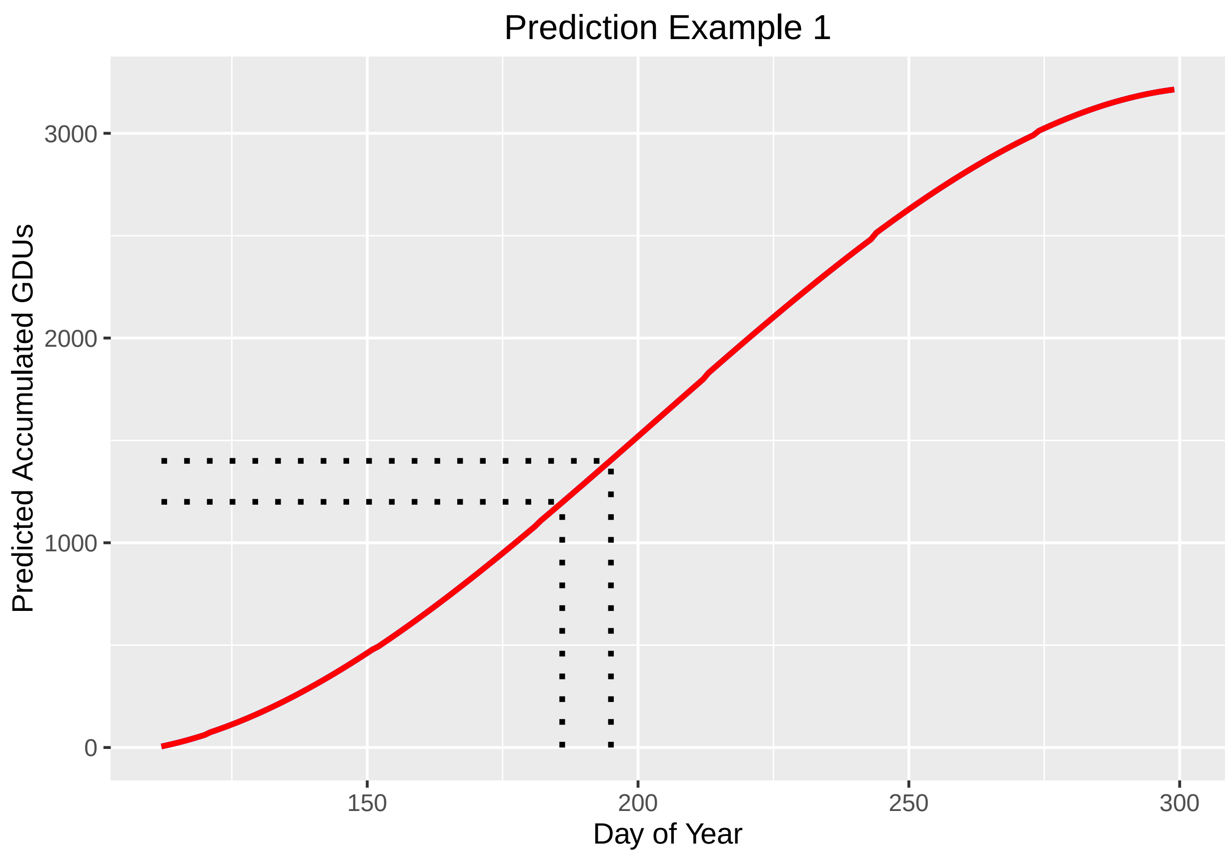
Considerations:

- Planting date and variety maturity are user provided inputs.
- Plant development is only affected by GDUs after planting. GDUs prior to planting are subtracted in variable agdu_ap_pred.
- Predicted pollination date is the date when accumulated GDUs after planting (agdu_ap_pred) reach variety maturity GDUs (gdu_mat#).

Example 1: A researcher plants two varieties on the same date, one that pollinates at 1200 GDUs and one that pollinates at 1400 GDUs. Predict the date each variety will pollinate.

```
agdu_pred <- predict(sel_model, envdat_test) # predicted agdu values for test dataset
xy <- data.frame(envdat_test, agdu_pred)

# Scenario 1 inputs:
loc1 <- "Iowa County, IA"
plant_yr1 <- 2011
plant_day1 <- 112
gdu_mat1 <- 1200

# Scenario 2 inputs:
loc2 <- "Iowa County, IA"
plant_yr2 <- 2011
plant_day2 <- 112
gdu_mat2 <- 1400

scenario1.1 <- subset(xy, county == loc1
                      & year == plant_yr1
                      & day_of_yr == plant_day1 - 1)
scenario1.2 <- mutate(subset(xy, county == loc1
                             & year == plant_yr1
                             & day_of_yr >= plant_day1),
                      agdu_ap_pred = agdu_pred - scenario1.1$agdu_pred)
scenario1.3 <- filter(scenario1.2, abs(agdu_ap_pred - gdu_mat1)
                      == min(abs(agdu_ap_pred - gdu_mat1)))

scenario2.1 <- subset(xy, county == loc2
                      & year == plant_yr2
                      & day_of_yr == plant_day2 - 1)
scenario2.2 <- mutate(subset(xy, county == loc2
                             & year == plant_yr2
                             & day_of_yr >= plant_day2),
                      agdu_ap_pred = agdu_pred - scenario2.1$agdu_pred)
scenario2.3 <- filter(scenario2.2, abs(agdu_ap_pred - gdu_mat2)
                      == min(abs(agdu_ap_pred - gdu_mat2)))

ggplot(mapping = aes(x = day_of_yr, y = agdu_ap_pred)) +
  labs(x = "Day of Year", y = "Predicted Accumulated GDUs", title = "Prediction Example 1") +
  geom_line(data = scenario1.2, color = "blue", size = 1) +
  geom_segment(aes(x = min(scenario1.2$day_of_yr), y = gdu_mat1,
                   xend = scenario1.3$day_of_yr, yend = gdu_mat1),
               size = 1, linetype = 3) +
  geom_segment(aes(x = scenario1.3$day_of_yr, y = 0,
                   xend = scenario1.3$day_of_yr, yend = gdu_mat1),
               size = 1, linetype = 3) +
```

```
  geom_line(data = scenario2.2, color = "red", size = 1) +
  geom_segment(aes(x = min(scenario2.2$day_of_yr), y = gdu_mat2,
                   xend = scenario2.3$day_of_yr, yend = gdu_mat2),
               size = 1, linetype = 3) +
  geom_segment(aes(x = scenario2.3$day_of_yr, y = 0,
                   xend = scenario2.3$day_of_yr, yend = gdu_mat2),
               size = 1, linetype = 3)
```

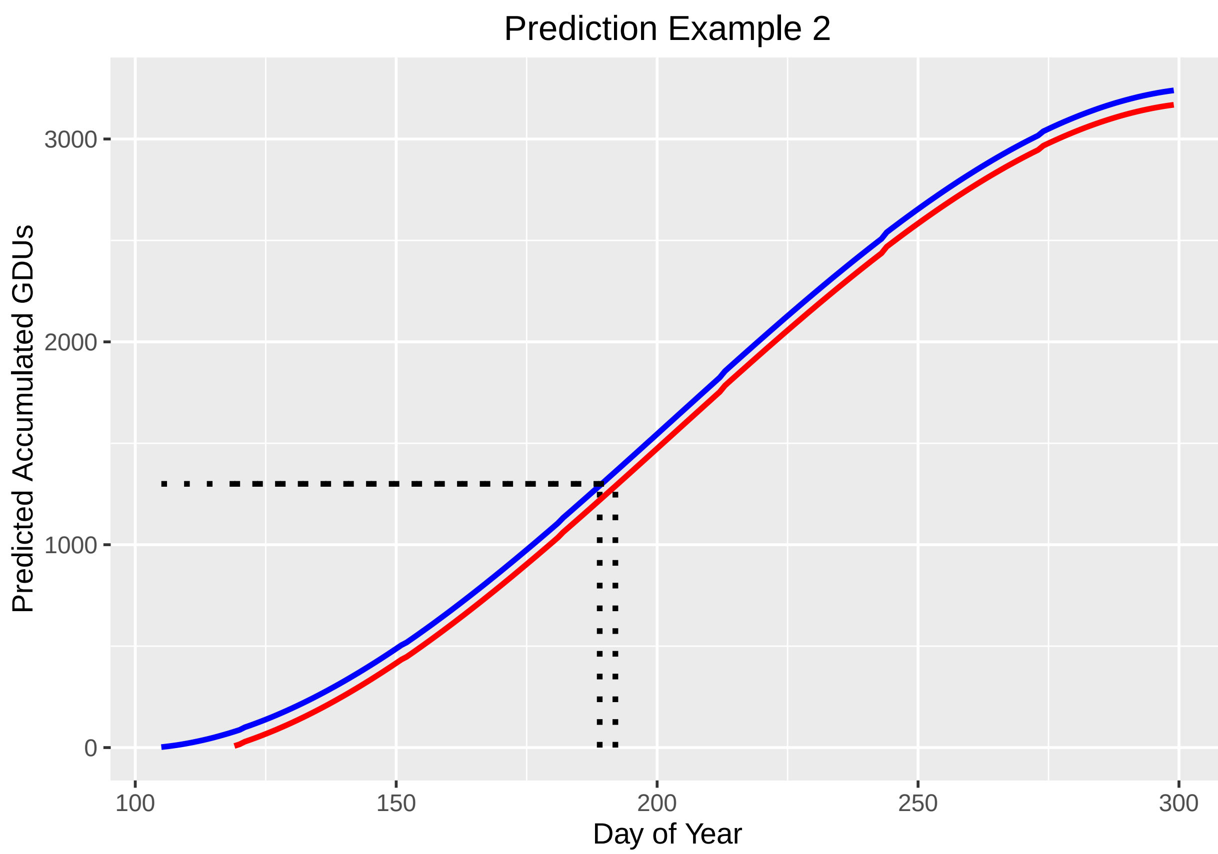



```
s1 <- data.frame(c(Scenario = 1, select(scenario1.3, day_of_yr, date)))
s2 <- data.frame(c(Scenario = 2, select(scenario2.3, day_of_yr, date)))
stargazer(arrange(union(s1, s2), Scenario), summary = FALSE, type="text")
```

```
##
## ===============================
##   Scenario day_of_yr    date
## -------------------------------
## 1    1         186    2011-07-05
## 2    2         195    2011-07-14
## -------------------------------
```

Example 2: A researcher intended to plant on April 14 (day 105) but was delayed for 14 days due to rain. Predict the number of days that pollination will be delayed.

```r
# Scenario 1 inputs:
loc1 <- "Darke County, OH"
plant_yr1 <- 2011
plant_day1 <- 105
gdu_mat1 <- 1300

# Scenario 2 inputs:
loc2 <- "Darke County, OH"
plant_yr2 <- 2011
plant_day2 <- 119
gdu_mat2 <- 1300

scenario1.1 <- subset(xy, county == loc1
                      & year == plant_yr1
                      & day_of_yr == plant_day1 - 1)
scenario1.2 <- mutate(subset(xy, county == loc1
                             & year == plant_yr1
                             & day_of_yr >= plant_day1),
                      agdu_ap_pred = agdu_pred - scenario1.1$agdu_pred)
scenario1.3 <- filter(scenario1.2, abs(agdu_ap_pred - gdu_mat1)
                      == min(abs(agdu_ap_pred - gdu_mat1)))

scenario2.1 <- subset(xy, county == loc2
                      & year == plant_yr2
                      & day_of_yr == plant_day2 - 1)
scenario2.2 <- mutate(subset(xy, county == loc2
                             & year == plant_yr2
                             & day_of_yr >= plant_day2),
                      agdu_ap_pred = agdu_pred - scenario2.1$agdu_pred)
scenario2.3 <- filter(scenario2.2, abs(agdu_ap_pred - gdu_mat2)
                      == min(abs(agdu_ap_pred - gdu_mat2)))

ggplot(mapping = aes(x = day_of_yr, y = agdu_ap_pred)) +
    labs(x = "Day of Year", y = "Predicted Accumulated GDUs", title = "Prediction Example 2") +
  geom_line(data = scenario1.2, color = "blue", size = 1) +
  geom_segment(aes(x = min(scenario1.2$day_of_yr), y = gdu_mat1,
                   xend = scenario1.3$day_of_yr, yend = gdu_mat1),
               size = 1, linetype = 3) +
  geom_segment(aes(x = scenario1.3$day_of_yr, y = 0,
                   xend = scenario1.3$day_of_yr, yend = gdu_mat1),
               size = 1, linetype = 3) +
  geom_line(data = scenario2.2, color = "red", size = 1) +
  geom_segment(aes(x = min(scenario2.2$day_of_yr), y = gdu_mat2,
                   xend = scenario2.3$day_of_yr, yend = gdu_mat2),
               size = 1, linetype = 3) +
  geom_segment(aes(x = scenario2.3$day_of_yr, y = 0,
                   xend = scenario2.3$day_of_yr, yend = gdu_mat2),
               size = 1, linetype = 3)
```




```
s1 <- data.frame(c(Scenario = 1, select(scenario1.3, day_of_yr, date)))
s2 <- data.frame(c(Scenario = 2, select(scenario2.3, day_of_yr, date)))
stargazer(arrange(union(s1, s2), Scenario), summary = FALSE, type="text")
```

```
##
## ================================
##   Scenario day_of_yr    date
## --------------------------------
## 1    1        189    2011-07-08
## 2    2        192    2011-07-11
## --------------------------------
```

Because GDUs accumulate more slowly in the early spring, a 14 day planting delay is predicted to result in only a 3 day delay in pollination for this combination of inputs.

Conclusions

Researchers developing new plant varieties must accurately forecast pollination date to properly allocate resources. This project demonstrated how predictions for accumulated GDUs, when combined with user provided inputs for planting date and variety maturity, can be used to forecast pollination date. Examples were provided comparing different planting scenarios and their impact on pollination date.

A regression model was developed to predict accumulated GDUs based on seasonal, geographical, and environmental variables. While the model was based on specific conditions from five locations in the U.S. Midwest and should not be used to make predictions outside of this region, the code and steps provided here could be readily applied to build a similar model using data from any other region of interest.