Capstone Project - Pollination Date Prediction

Tom Gocken

Monday, February 15, 2016

## Introduction

Research sites that develop new plant varieties must predict when certain seasonal events, such as pollination, will occur. These events drive work timelines and allocation of resources. The timing of these events depends on 1) planting date, 2) variety maturity expressed in growing degree units (GDUs) required to reach a specified growth stage, such as pollination or physiological maturity, 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 predict pollination date and to model planting scenarios.

## Data Sources

* Environmental data for counties of interest: <http://wonder.cdc.gov/EnvironmentalData.html>
  + List variables and provide definitions...
* 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 ERSST 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, Tbase, as follows:

#### GDU = ((Tmax + Tmin) / 2) -- Tbase

where Tmax is equal to the maximum daily temperature but not greater than a defined upper limit and Tmin 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 Steps

Add from previous project...

setwd("C:/Projects/springboard-capstone")  
envdat <- read.delim("../data/envdat.txt") # Previously created tidy dataset  
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)

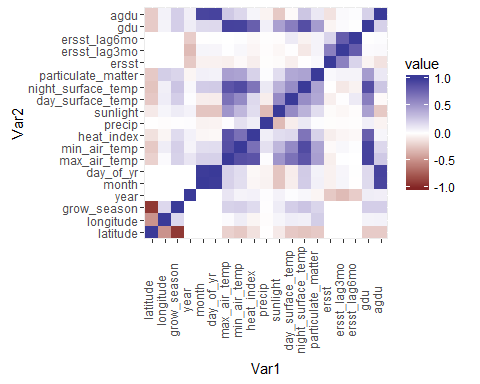
## Correlations

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 ersst, ersst\_lag3mo, and ersst\_lag6mo 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.

library(ggplot2)  
library(reshape2)  
library(dplyr)  
library(tidyr)

# 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)



# 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))

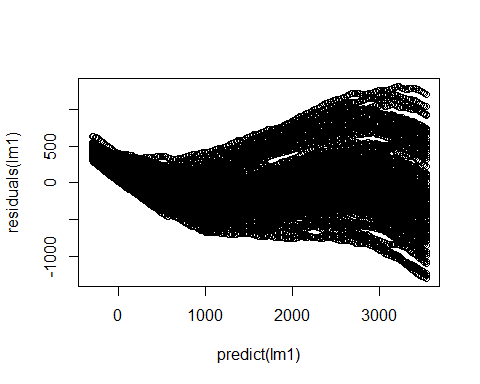
## Var1 Var2 Corr\_Coeff  
## 1 agdu day\_of\_yr 0.9465124  
## 2 agdu month 0.9384011  
## 3 day\_of\_yr agdu 0.9465124  
## 4 day\_of\_yr month 0.9895758  
## 5 ersst ersst\_lag3mo 0.6147758  
## 6 ersst\_lag3mo ersst 0.6147758  
## 7 ersst\_lag3mo ersst\_lag6mo 0.8060249  
## 8 ersst\_lag3mo year -0.3023568  
## 9 ersst\_lag6mo ersst\_lag3mo 0.8060249  
## 10 gdu max\_air\_temp 0.9517451  
## 11 gdu min\_air\_temp 0.9494270  
## 12 gdu sunlight 0.4534379  
## 13 grow\_season latitude -0.9119482  
## 14 latitude grow\_season -0.9119482  
## 15 latitude longitude -0.4849110  
## 16 longitude latitude -0.4849110  
## 17 max\_air\_temp gdu 0.9517451  
## 18 max\_air\_temp min\_air\_temp 0.8814343  
## 19 max\_air\_temp sunlight 0.5095273  
## 20 min\_air\_temp gdu 0.9494270  
## 21 min\_air\_temp max\_air\_temp 0.8814343  
## 22 min\_air\_temp sunlight 0.3383542  
## 23 month agdu 0.9384011  
## 24 month day\_of\_yr 0.9895758  
## 25 precip sunlight -0.3021674  
## 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 ersst\_lag3mo -0.3023568

## Model Building

#### Day of Year

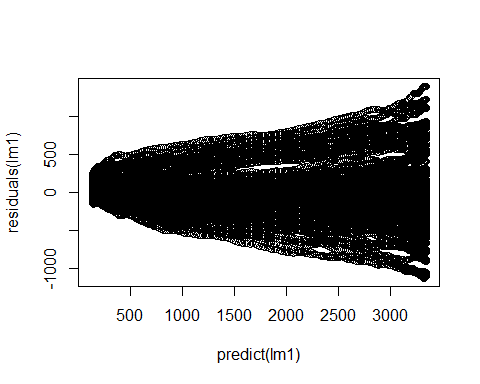
The strongest correlation between response variable agdu and potential predictor variables was with day\_of\_year (0.95). However, this relationship is known to be non-linear since GDUs accumulate more slowly during cool days in the early spring and late fall than they do during hot days in the summer. 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 high heteroscedasticity.

lm1 <- lm(agdu ~ day\_of\_yr, data = envdat\_train)  
plot(predict(lm1), residuals(lm1))



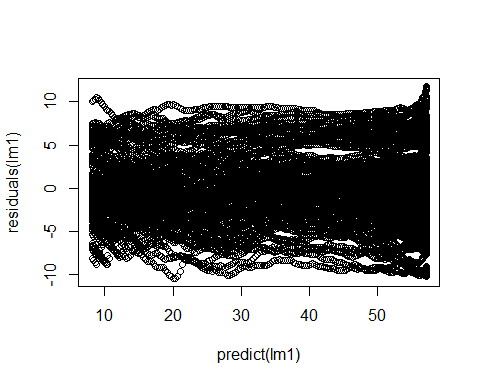
A cubic polynomial model addresses the issue of non-linearity but heteroscedasticity remains.

lm1 <- lm(agdu ~ poly(day\_of\_yr, 3), data = envdat\_train)  
plot(predict(lm1), residuals(lm1))



Heteroscedasticity was addressed with a square root tranformation of response varible agdu, as recommended by James, et al., An Introduction to Statistical Learning (<http://www-bcf.usc.edu/~gareth/ISL/>).

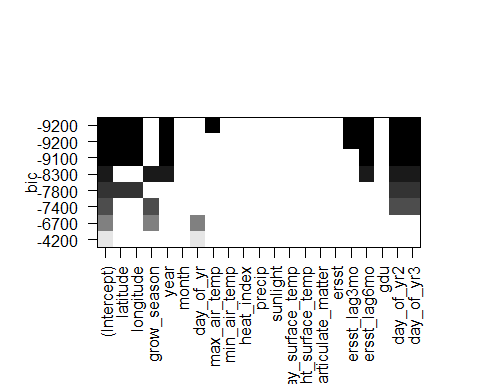
lm1 <- lm(sqrt(agdu) ~ poly(day\_of\_yr, 3), data = envdat\_train)  
plot(predict(lm1), residuals(lm1))



#### 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 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(sqrt(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 (variables day\_of\_yr, doy2, and doy3) 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, grow\_season, year, and ersst\_lag6mo (El Nino effects from 6 months prior. These terms correspond to the best 5 predictor model plus main effects for day of year. Additional terms provide little additional improvement and risk overfitting.

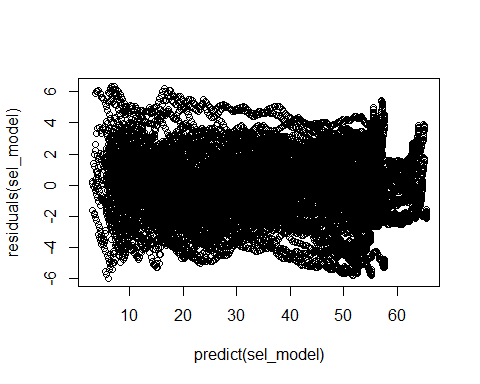
# Training data results:  
sel\_model <- lm(sqrt(agdu) ~ poly(day\_of\_yr, 3) + grow\_season   
 + year + ersst\_lag6mo, data = envdat\_train)  
summary(sel\_model)

##   
## Call:  
## lm(formula = sqrt(agdu) ~ poly(day\_of\_yr, 3) + grow\_season +   
## year + ersst\_lag6mo, data = envdat\_train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -5.9847 -1.3554 0.1647 1.3175 6.3457   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -4.027e+02 5.961e+00 -67.55 <2e-16 \*\*\*  
## poly(day\_of\_yr, 3)1 2.175e+03 2.050e+00 1060.76 <2e-16 \*\*\*  
## poly(day\_of\_yr, 3)2 -2.577e+02 2.051e+00 -125.65 <2e-16 \*\*\*  
## poly(day\_of\_yr, 3)3 -1.506e+02 2.050e+00 -73.45 <2e-16 \*\*\*  
## grow\_season 2.010e-01 8.607e-04 233.47 <2e-16 \*\*\*  
## year 2.031e-01 2.979e-03 68.17 <2e-16 \*\*\*  
## ersst\_lag6mo 6.700e-01 1.720e-02 38.95 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.05 on 18893 degrees of freedom  
## Multiple R-squared: 0.9846, Adjusted R-squared: 0.9846   
## F-statistic: 2.008e+05 on 6 and 18893 DF, p-value: < 2.2e-16

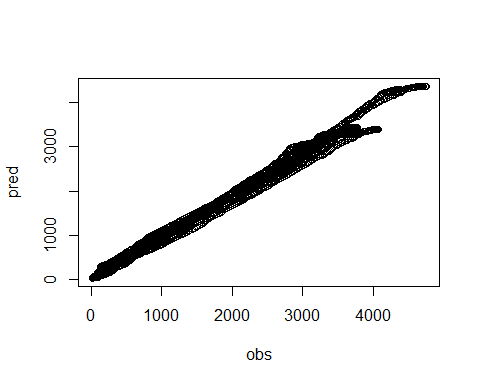
agdu\_predsqrt <- predict(sel\_model, envdat\_test)  
agdu\_compare <- data.frame(obs = envdat\_test$agdu, pred = agdu\_predsqrt^2)  
  
# R sq for test data:  
caret::defaultSummary(agdu\_compare)

## RMSE Rsquared   
## 173.0558143 0.9906839

plot(predict(sel\_model), residuals(sel\_model))



plot(agdu\_compare)



## Prediction Scenarios

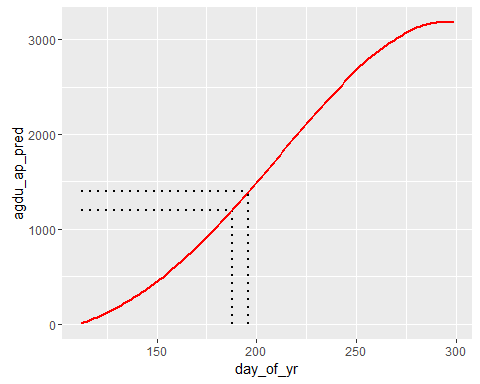
As a researcher, the practical value of pollination date prediction is to model different planting scenarios and make informed resourcing decisions. Below are examples.

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 gdu\_mat# is reached as defined in scenario#.3.

#### 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.

xy <- data.frame(envdat\_test, agdu\_pred = agdu\_predsqrt^2)  
  
# 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)) +  
 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)



scenario1.3$date

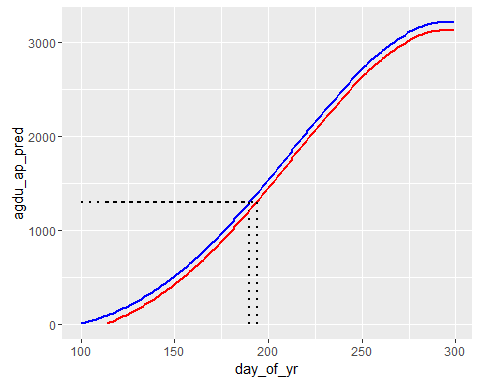
## [1] 2011-07-07  
## 7305 Levels: 1992-01-01 1992-01-02 1992-01-03 1992-01-04 ... 2011-12-31

scenario2.3$date

## [1] 2011-07-15  
## 7305 Levels: 1992-01-01 1992-01-02 1992-01-03 1992-01-04 ... 2011-12-31

#### Example 2: A researcher intended to plant on April 9 (day 100) but was delayed for 14 days due to rain. Predict the number of days that pollination will be delayed.

# Scenario 1 inputs:  
loc1 <- "Darke County, OH"  
plant\_yr1 <- 2011  
plant\_day1 <- 100  
gdu\_mat1 <- 1300  
  
# Scenario 2 inputs:  
loc2 <- "Darke County, OH"  
plant\_yr2 <- 2011  
plant\_day2 <- 114  
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)) +  
 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)



scenario1.3$date

## [1] 2011-07-09  
## 7305 Levels: 1992-01-01 1992-01-02 1992-01-03 1992-01-04 ... 2011-12-31

scenario2.3$date

## [1] 2011-07-13  
## 7305 Levels: 1992-01-01 1992-01-02 1992-01-03 1992-01-04 ... 2011-12-31

Because GDUs accumulate more slowly in the early spring, a 14 day planting delay is predicted to result in only a 4 day delay in pollination.

## Conclusions

...

## Follow-up before submitting...

* Replace above model with robust linear model with no sqrt transformation?
* Get feedback from Henry

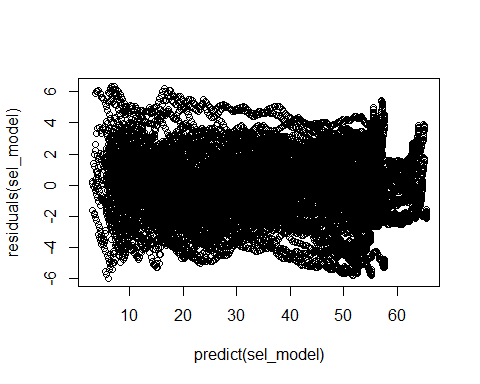
# Training data results:  
sel\_model\_rlm <- MASS::rlm(agdu ~ poly(day\_of\_yr, 3) + grow\_season   
 + year + ersst\_lag6mo, data = envdat\_train)  
summary(sel\_model\_rlm)

##   
## Call: rlm(formula = agdu ~ poly(day\_of\_yr, 3) + grow\_season + year +   
## ersst\_lag6mo, data = envdat\_train)  
## Residuals:  
## Min 1Q Median 3Q Max   
## -807.48 -123.60 13.97 115.77 793.72   
##   
## Coefficients:  
## Value Std. Error t value   
## (Intercept) -28539.3203 564.6188 -50.5462  
## poly(day\_of\_yr, 3)1 153135.6928 194.1987 788.5514  
## poly(day\_of\_yr, 3)2 9316.0361 194.2824 47.9510  
## poly(day\_of\_yr, 3)3 -18109.7407 194.1972 -93.2544  
## grow\_season 15.1692 0.0815 186.0627  
## year 13.8316 0.2822 49.0206  
## ersst\_lag6mo 40.8404 1.6293 25.0660  
##   
## Residual standard error: 176.8 on 18893 degrees of freedom

agdu\_pred\_rlm <- predict(sel\_model\_rlm, envdat\_test)  
agdu\_compare\_rlm <- data.frame(obs = envdat\_test$agdu, pred = agdu\_pred\_rlm)  
  
# R sq for test data:  
caret::defaultSummary(agdu\_compare\_rlm)

## RMSE Rsquared   
## 236.6786619 0.9829922

plot(predict(sel\_model), residuals(sel\_model))



plot(agdu\_compare)

