

Exploratory Analysis

Stephen Brownsey

31/10/2019

Introduction

The .rmd file for my exploratory data analysis. Loading required Libraries

```
library(tidyverse)
library(ggmap)
library(leaps)
library(GGally)
```

Map Analysis

This analysis covers sensitive PI data and as such will not be used in any final reporting. However, since the data is freely available online, the analysis will be kept in the GitHub page code.

```
#Loading the data in from paper ~ downloaded from their resources as a .xlsx, created csv with useful p
data <- read.csv("data.csv") %>%
  #renaming as column name had a few unrecognised characters in front
  rename(orchard = contains("orchard"))

#Creating an indicator variable to show whether the Orchard appeared in both years or just 1 (0 -> No, 1 -> Yes)
years <- data %>%
  select(orchard, year) %>%
  distinct() %>%
  count(orchard) %>%
  rename(both_years = n) %>%
  mutate(both_years = (both_years - 1)) %>%
  mutate(id = row_number())

data <- merge(data, years)

#Isolating the lat and long
lat_long <- data %>%
  select(lat, long, both_years) %>%
  distinct()

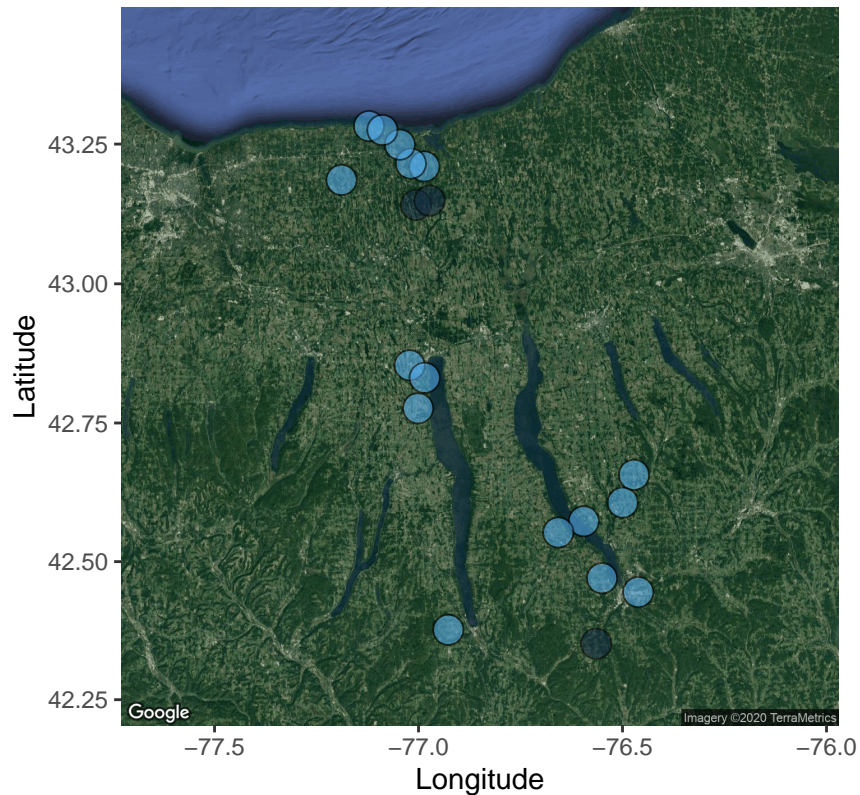
register_google(key = "AIzaSyB9Hpt0vTWrALpm0iUyJxW6C2IuHZy1pC8")
map <- get_map(location = c(lon = mean(lat_long$lon), lat = mean(lat_long$lat)), zoom = 9,
               maptype = "satellite", scale = 2)
```

```
## Source : https://maps.googleapis.com/maps/api/staticmap?center=42.85269,-76.850661&zoom=9&size=640x640
```

```
#~~~~~Plotting the locations of all the orchards:
```

```
plotted_map <- ggmap(map) +  
  geom_point(data = lat_long, aes(x = long, y = lat, fill = both_years , alpha = 0.8), size = 5, shape = )  
  guides(fill = FALSE, alpha = FALSE, size = FALSE) +  
  ggtitle("Orchard Locations") +  
  xlab("Longitude") +  
  ylab("Latitude")  
plotted_map
```

Orchard Locations

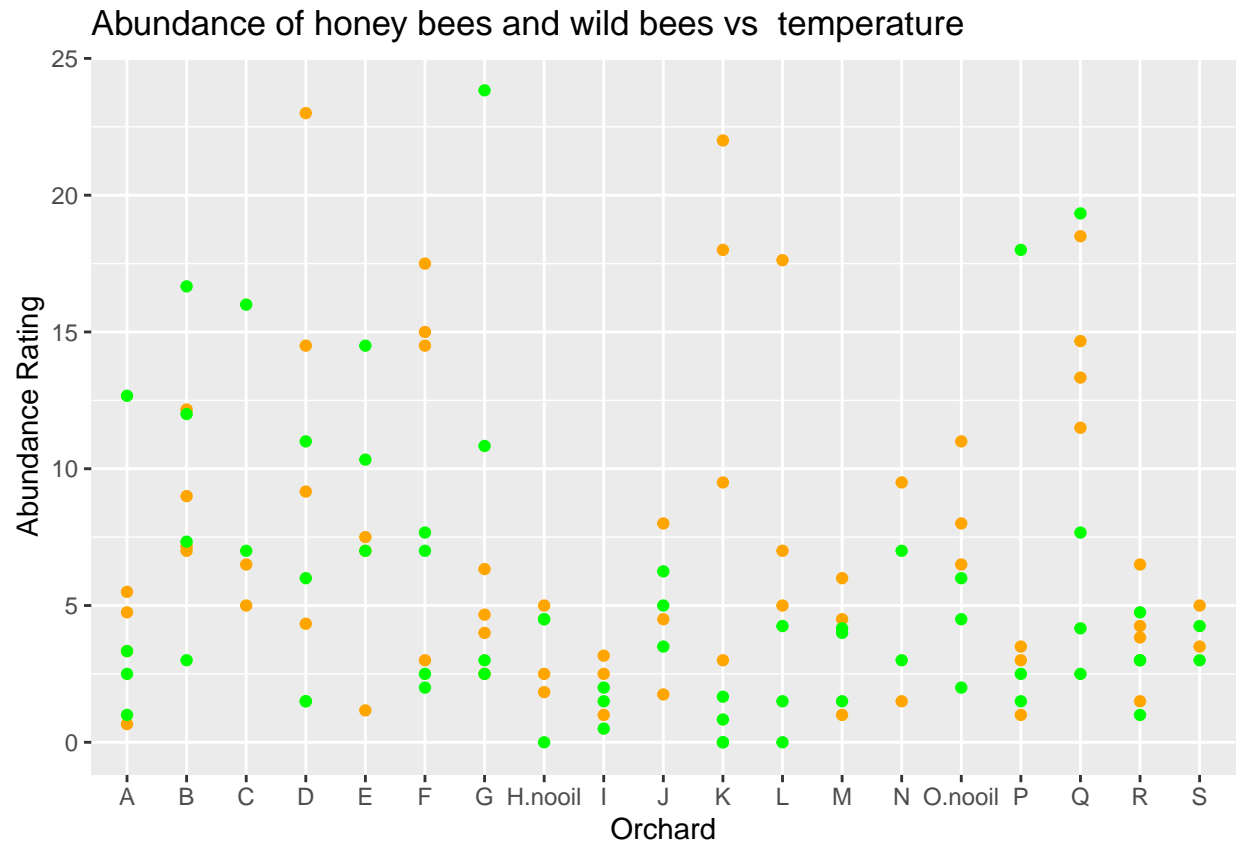


```
##
```

General EDA

```
abundance_plot <- ggplot(data = data) +  
  geom_point(aes(x = orchard, y = apisAb), colour = "orange") +  
  geom_smooth(aes(x = orchard, y = apisAb), colour = "orange") +  
  geom_point(aes(x = orchard, y = wildAbF), colour = "green") +  
  geom_smooth(aes(x = orchard, y = wildAbF), colour = "green") +  
  ggtitle("Abundance of honey bees and wild bees vs temperature") +  
  xlab("Orchard") +  
  ylab("Abundance Rating")  
abundance_plot
```

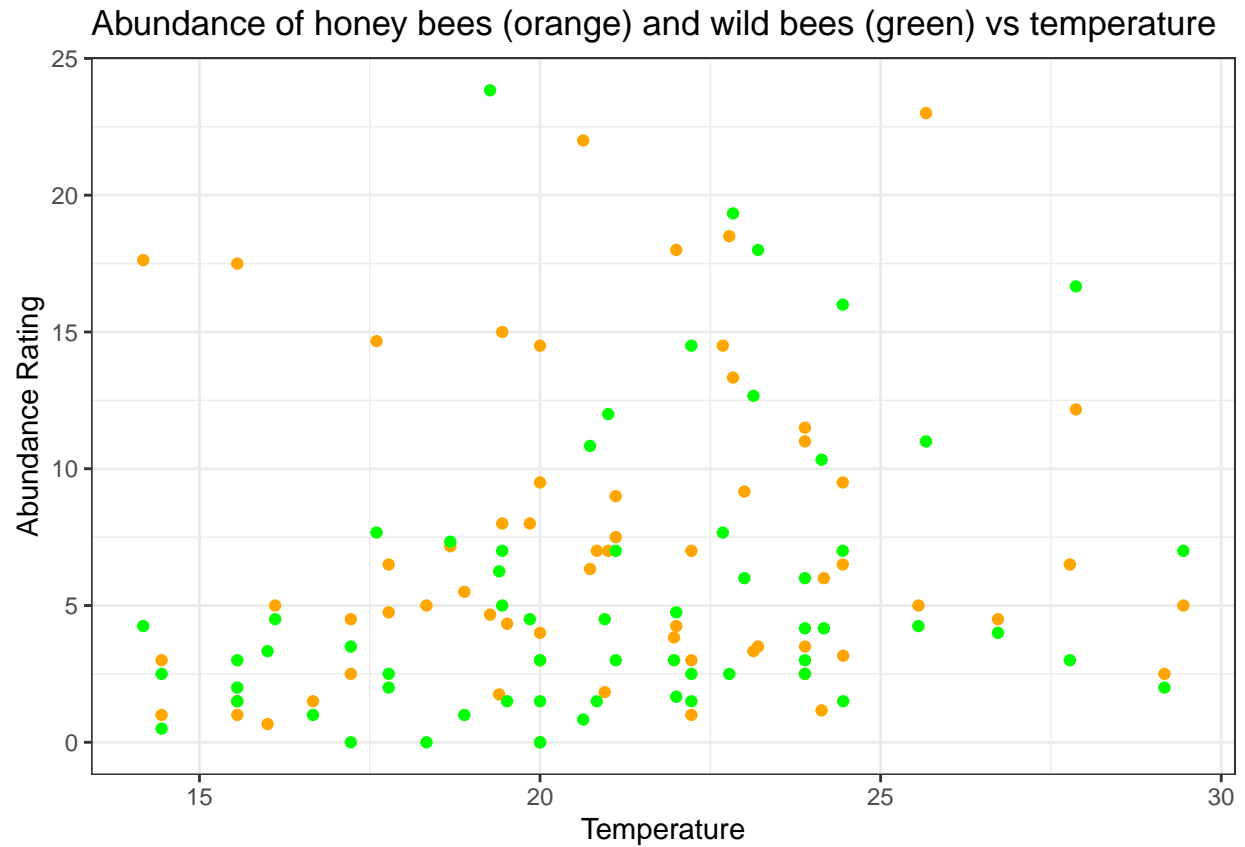
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'  
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



```
abundance_summary <- data %>% summarise(  
  honey_mean = round(mean(apisAb),2),  
  wild_mean = round(mean(wildAbF), 2),  
  honey_sd = round(sd(apisAb), 2),  
  wild_sd = round(sd(wildAbF), 2)  
)  
abundance_summary
```

```
##   honey_mean wild_mean honey_sd wild_sd  
## 1      7.19      5.41      5.51      5.21
```

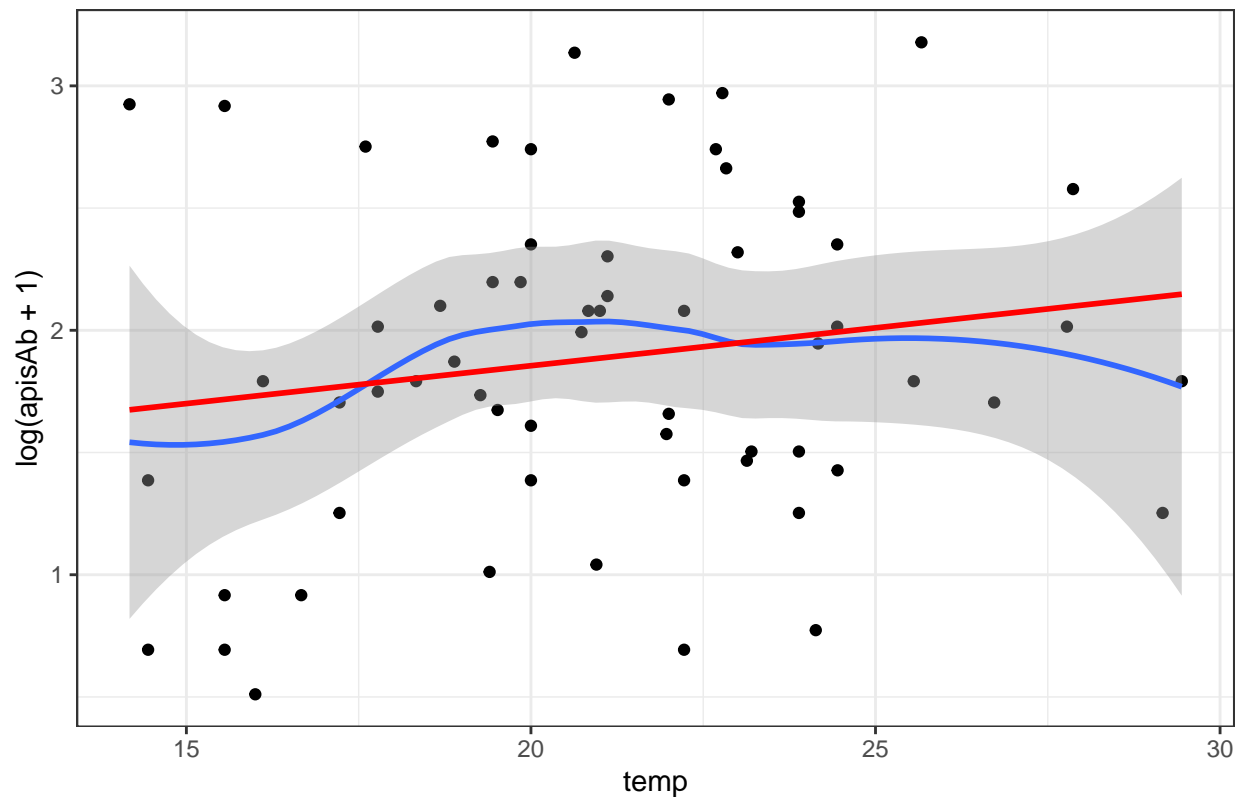
```
#Looking at temperature vs bees  
temperature_plot <- ggplot(data = data) +  
  geom_point(aes(x = temp, y = apisAb), colour = "orange") +  
  geom_point(aes(x = temp, y = wildAbF), colour = "green") +  
  ggtitle("Abundance of honey bees (orange) and wild bees (green) vs temperature") +  
  xlab("Temperature") +  
  ylab("Abundance Rating")+  
  theme_bw()  
temperature_plot
```



```
honey_temp <- ggplot(data = data, aes(x = temp, y = log(apisAb + 1) )) +
  geom_point() +
  ggtitle("Honey Bee Abundance vs Temperature") +
  geom_smooth() +
  geom_smooth(method='lm', formula= y~x, se = FALSE, colour = "red") +
  theme_bw()
honey_temp
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

Honey Bee Abundance vs Temperature



#Looking at the data it seems a quadratic model would be the best fit, shown below

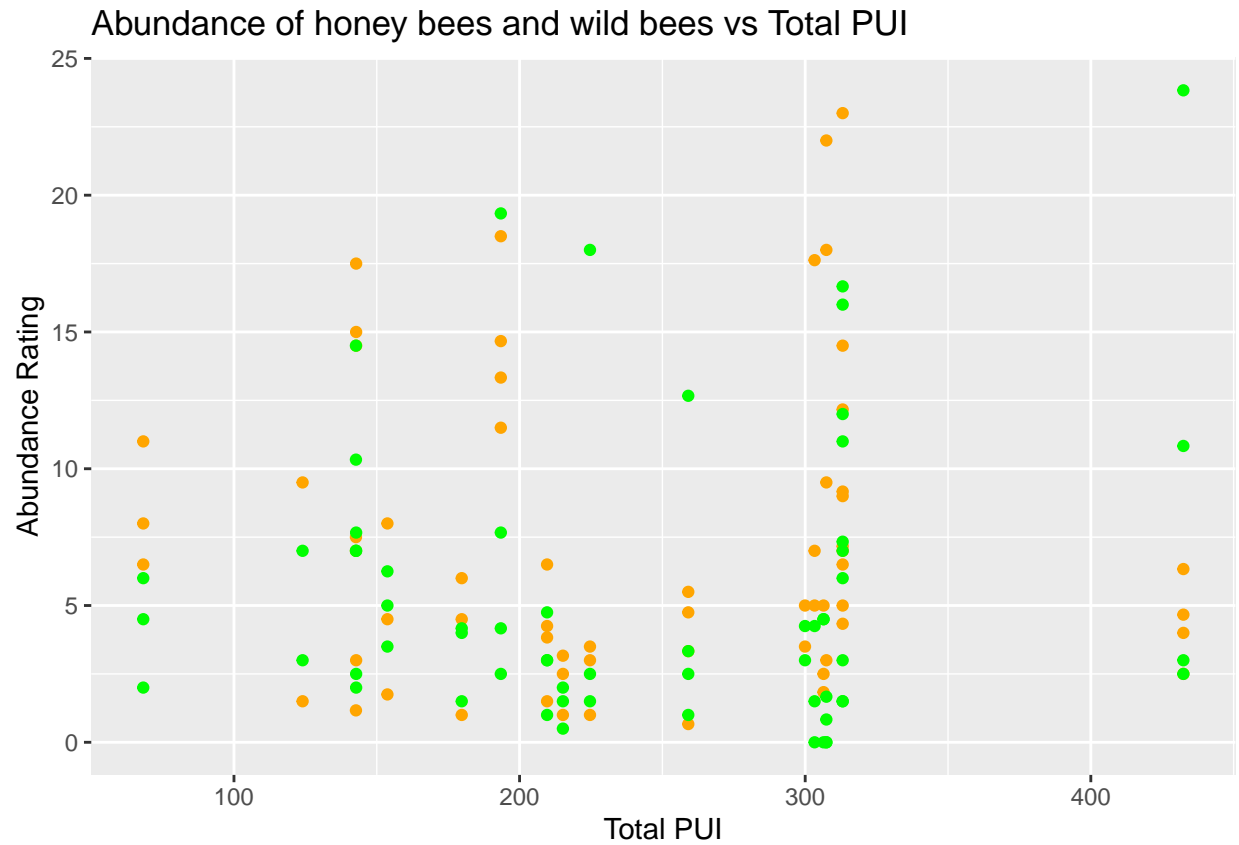
```
temperature_plot <- temperature_plot +
  stat_smooth(aes(x = temp, y = apisAb), method = "lm", formula = y ~ x + I(x^2), colour = "orange") +
  stat_smooth(aes(x = temp, y = wildAbF), method = "lm", formula = y ~ x + I(x^2), colour = "green") +
  theme_bw()
```

```
log_data <- data %>%
  mutate(apisAb = log(apisAb), wildAbF = log(wildAbF))
```

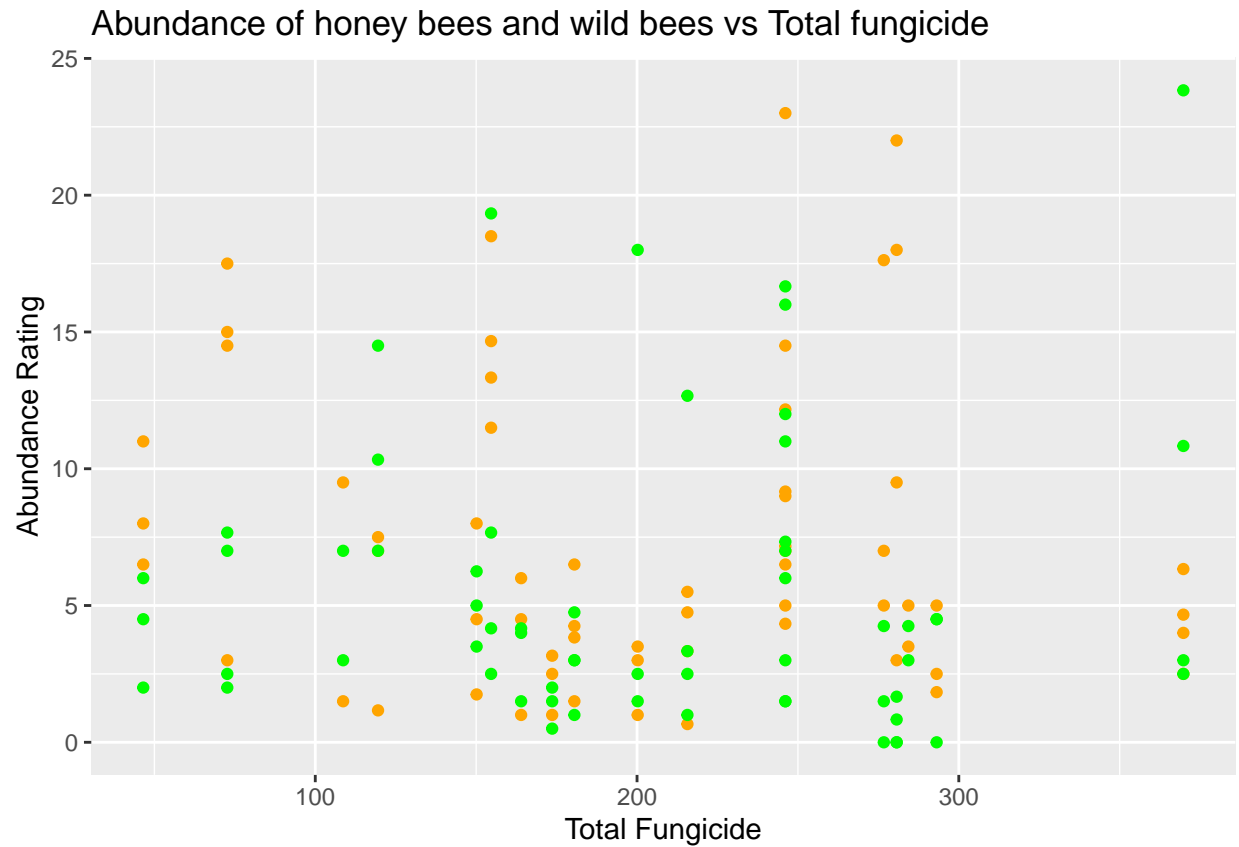
```
temperature_plot_log <- ggplot(data = log_data) +
  geom_point(aes(x = temp, y = apisAb), colour = "orange") +
  geom_point(aes(x = temp, y = wildAbF), colour = "green") +
  ggtitle("Abundance of honey bees and wild bees vs temperature") +
  xlab("Temperature") +
  ylab("Abundance Rating")
```

#Doesn't show too much

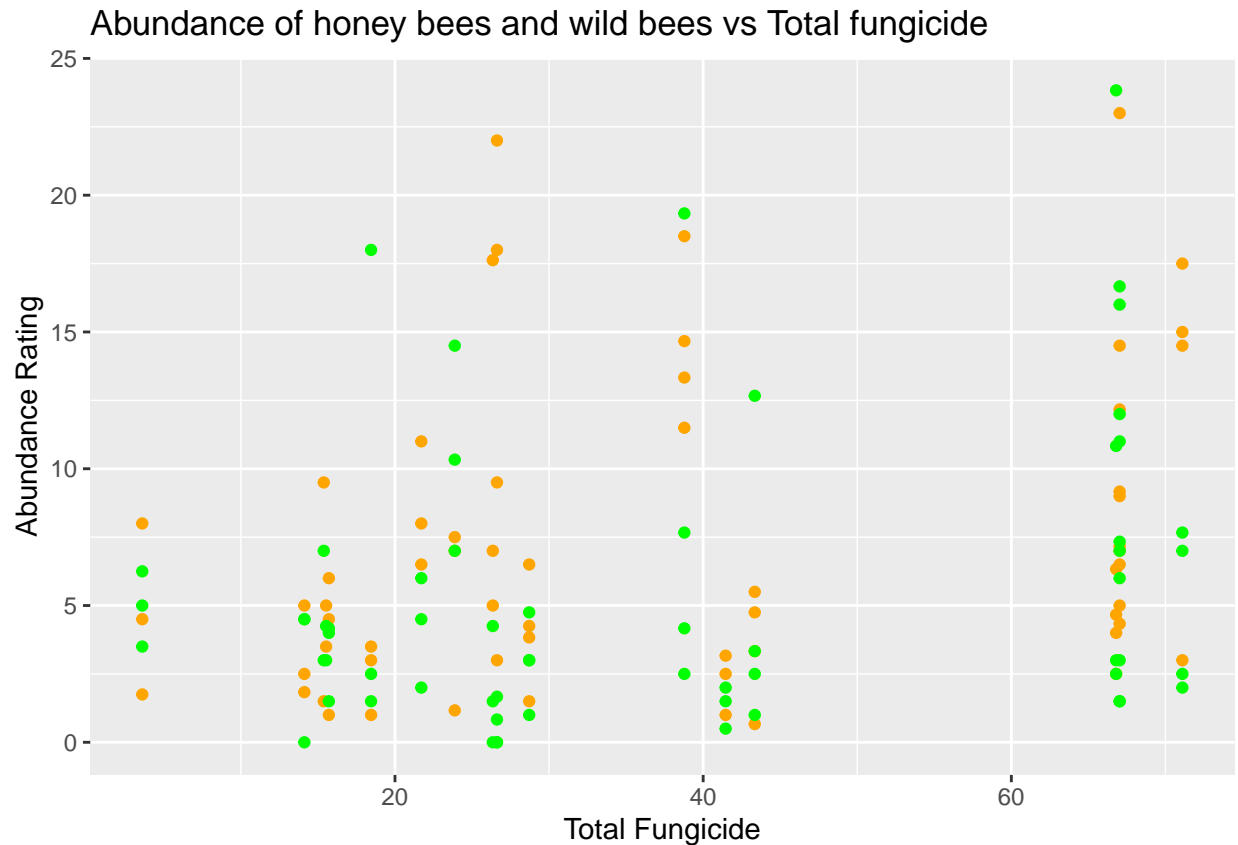
```
pui_plot <- ggplot(data = data) +
  geom_point(aes(x = eqB11, y = apisAb), colour = "orange") +
  geom_point(aes(x = eqB11, y = wildAbF), colour = "green") +
  ggtitle("Abundance of honey bees and wild bees vs Total PUI") +
  xlab("Total PUI") +
  ylab("Abundance Rating")
pui_plot
```



```
fungicide_plot <- ggplot(data = data) +
  geom_point(aes(x = eiQB11.fun, y = apisAb), colour = "orange") +
  geom_point(aes(x = eiQB11.fun, y = wildAbF), colour = "green") +
  ggtitle("Abundance of honey bees and wild bees vs Total fungicide") +
  xlab("Total Fungicide") +
  ylab("Abundance Rating")
fungicide_plot
```



```
Insectcide_plot <- ggplot(data = data) +
  geom_point(aes(x = eiQB11.ins, y = apisAb, colour = "orange")) +
  geom_point(aes(x = eiQB11.ins, y = wildAbF, colour = "green")) +
  ggtitle("Abundance of honey bees and wild bees vs Total fungicide") +
  xlab("Total Fungicide") +
  ylab("Abundance Rating")
Insectcide_plot
```



Leaps

The purpose of the leaps library is to perform best subset selection and see which variables tend to have the largest affect on the response variable.

```
summary(regsubsets( apisAb ~ eiQB11 + eiQB11.np + eiQB11.fun      + eiQB11.ins + eiQB11.ins.np +
                    eiQB11F.pre + eiQB11F.blm + eiQB11F.pos + eiQB11I.pre + eiQB11I.blm +
                    eiQB11I.pos + eiQB11I.pos.np + eiQB11T.blm + eiQB11T.pos + size + hive.a
, data = data, nvmax = 6))
```

```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
## force.in = force.in, : 2 linear dependencies found
```

```
## Reordering variables and trying again:
```

```
## Subset selection object
```

```
## Call: regsubsets.formula(apisAb ~ eiQB11 + eiQB11.np + eiQB11.fun +
##   eiQB11.ins + eiQB11.ins.np + eiQB11F.pre + eiQB11F.blm +
##   eiQB11F.pos + eiQB11I.pre + eiQB11I.blm + eiQB11I.pos + eiQB11I.pos.np +
##   eiQB11T.blm + eiQB11T.pos + size + hive.a + X2000nat, data = data,
##   nvmax = 6)
```

```
## 17 Variables (and intercept)
```

```
##           Forced in Forced out
## eiQB11          FALSE      FALSE
## eiQB11.np        FALSE      FALSE
## eiQB11.fun        FALSE      FALSE
## eiQB11.ins        FALSE      FALSE
```

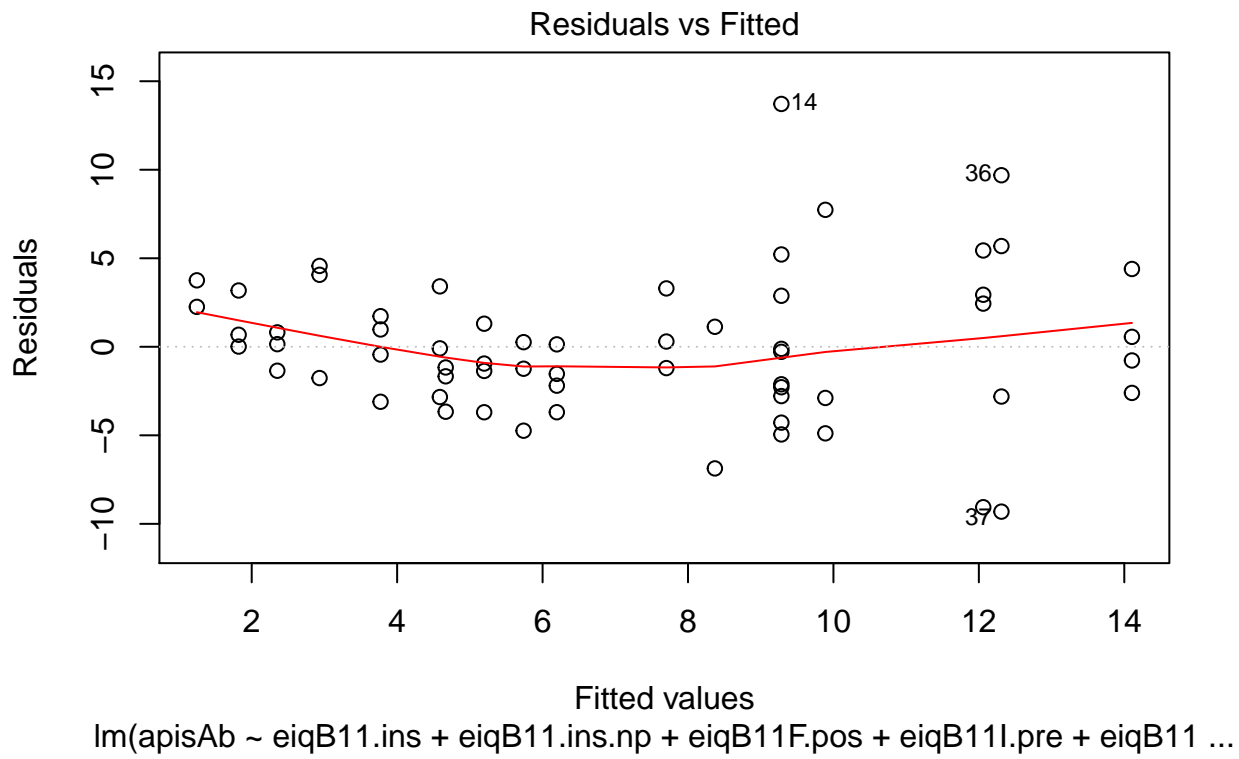


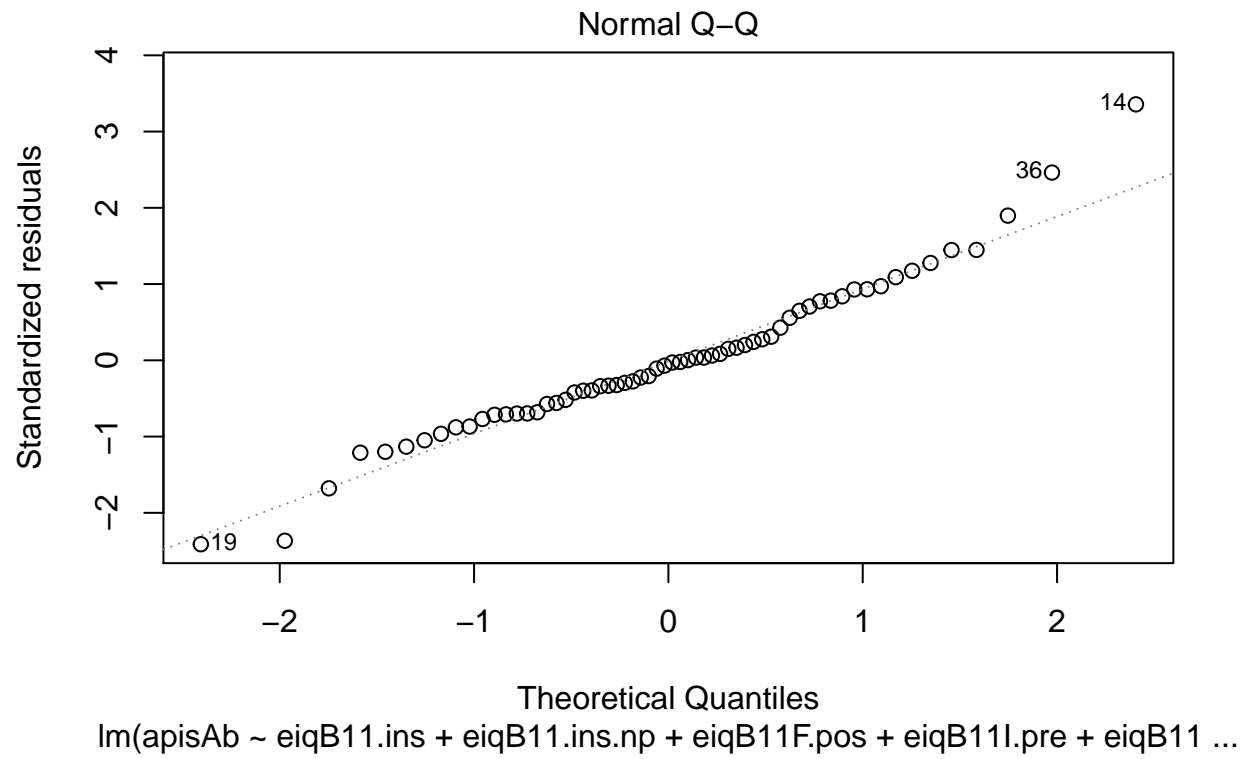
```

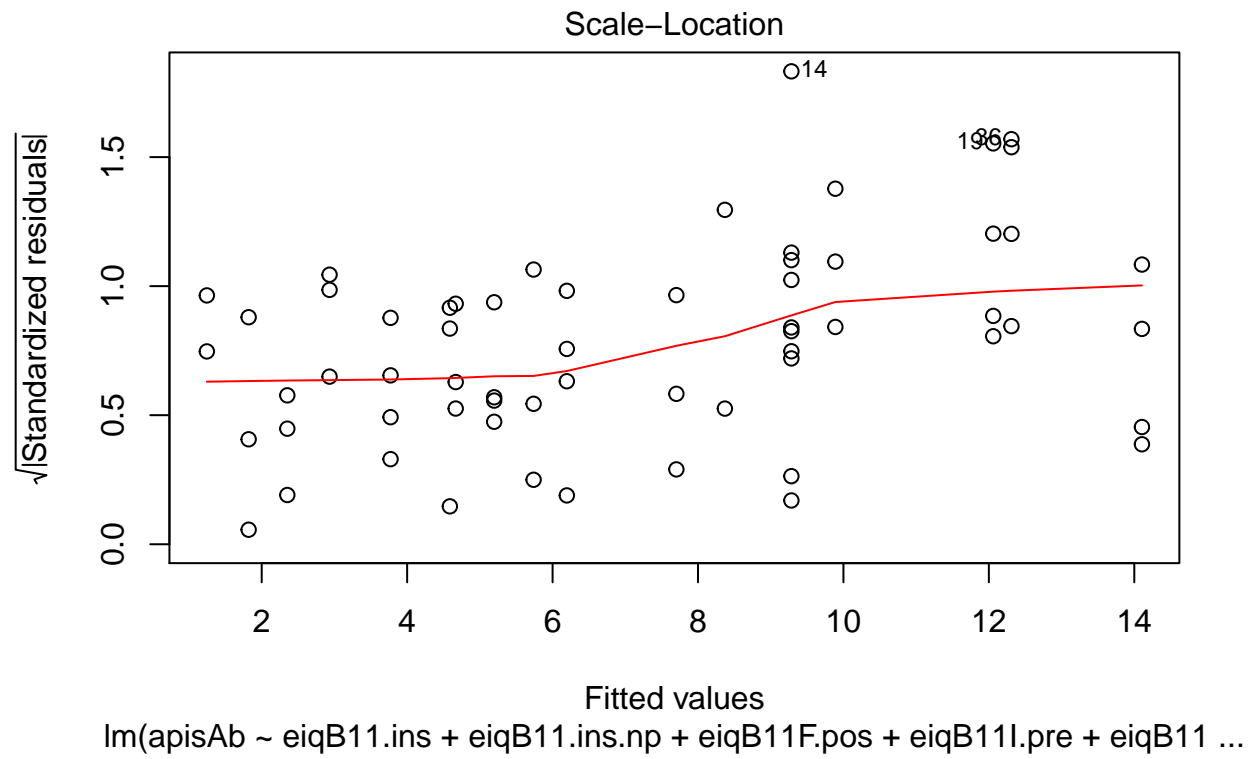
## eqB11F.pre      FALSE      FALSE
## eqB11F.blm      FALSE      FALSE
## eqB11F.pos      FALSE      FALSE
## eqB11I.pre      FALSE      FALSE
## eqB11I.blm      FALSE      FALSE
## eqB11I.pos      FALSE      FALSE
## eqB11I.pos.np   FALSE      FALSE
## eqB11T.blm      FALSE      FALSE
## size            FALSE      FALSE
## hive.acr        FALSE      FALSE
## X2000nat        FALSE      FALSE
## eqB11.ins.np    FALSE      FALSE
## eqB11T.pos      FALSE      FALSE
## 1 subsets of each size up to 7
## Selection Algorithm: exhaustive
##      eqB11 eqB11.np eqB11.fun eqB11.ins eqB11.ins.np eqB11F.pre
## 1 ( 1 ) " " " " " " "*" " " " "
## 2 ( 1 ) " " " " " " "*" " " " "
## 3 ( 1 ) " " "*" " " " " " " "*"
## 4 ( 1 ) "*" "*" " " " " " " "*"
## 5 ( 1 ) " " " " " " " " " " "*"
## 6 ( 1 ) " " " " " " "*" "*" " "
## 7 ( 1 ) " " " " " " "*" "*" " "
##      eqB11F.blm eqB11F.pos eqB11I.pre eqB11I.blm eqB11I.pos
## 1 ( 1 ) " " " " " " " " " "
## 2 ( 1 ) " " "*" " " " " " "
## 3 ( 1 ) " " " " " " "*" " "
## 4 ( 1 ) " " " " " " "*" " "
## 5 ( 1 ) " " "*" "*" "*" " "
## 6 ( 1 ) " " "*" "*" "*" "*"
## 7 ( 1 ) " " "*" "*" "*" "*"
##      eqB11I.pos.np eqB11T.blm eqB11T.pos size hive.acr X2000nat
## 1 ( 1 ) " " " " " " " " " "
## 2 ( 1 ) " " " " " " " " " "
## 3 ( 1 ) " " " " " " " " " "
## 4 ( 1 ) " " " " " " " " " "
## 5 ( 1 ) "*" " " " " " " " "
## 6 ( 1 ) " " " " " " " " " "
## 7 ( 1 ) "*" " " " " " " " "

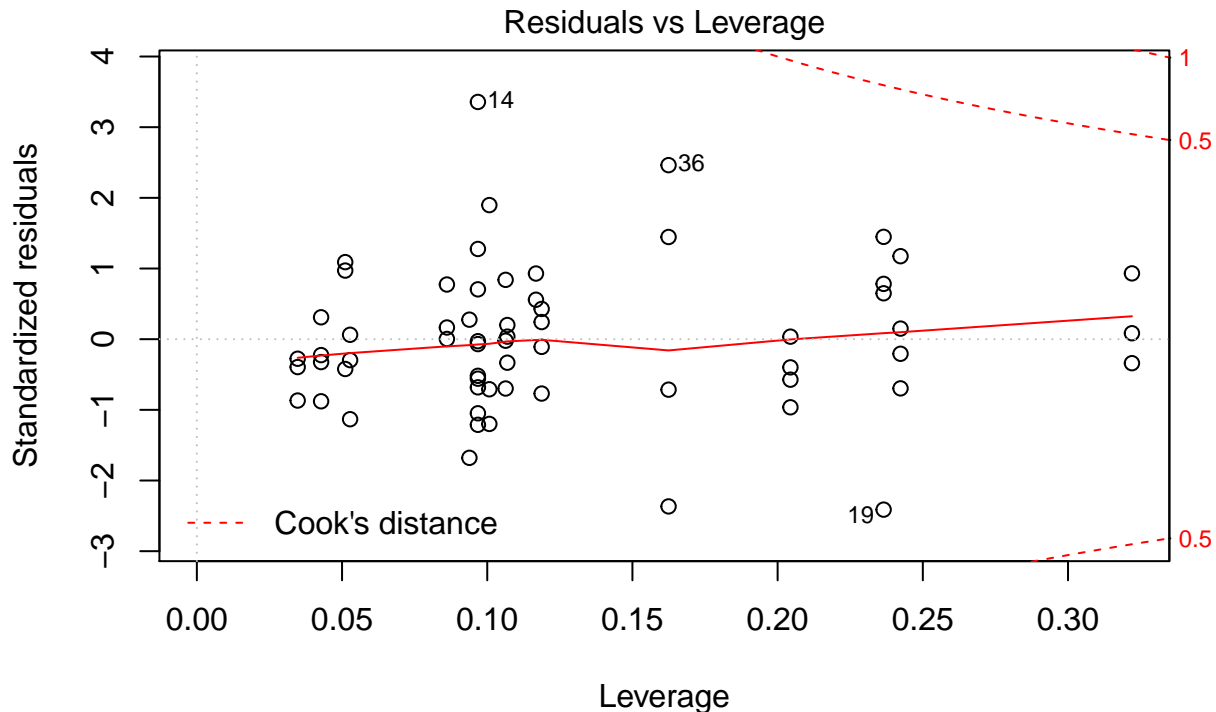
#Let's look at a graph of the one with 7 variables:
#eqB11.ins, eqB11.ins.np eqB11F.pos eqB11I.pre eqB11I.blm eqB11I.pos eqB11I.pos.np
lm_non_vary <- lm(apisAb ~ eqB11.ins + eqB11.ins.np + eqB11F.pos +
                  eqB11I.pre + eqB11I.blm + eqB11I.pos + eqB11I.pos.np, data)
plot(lm_non_vary)

```









lm(apisAb ~ eqB11.ins + eqB11.ins.np + eqB11F.pos + eqB11I.pre + eqB11I.blm + eqB11I.pos + eqB11I.pos.np, data = data)

summary(lm_non_vary)

```
##
## Call:
## lm(formula = apisAb ~ eqB11.ins + eqB11.ins.np + eqB11F.pos +
##     eqB11I.pre + eqB11I.blm + eqB11I.pos + eqB11I.pos.np,
##     data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -9.3103 -2.5251 -0.2008  2.3934 13.7159
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    9.58254    2.00881   4.770 1.44e-05 ***
## eqB11.ins     -613.24900   141.93474  -4.321 6.72e-05 ***
## eqB11.ins.np   -0.62024    0.14172  -4.376 5.56e-05 ***
## eqB11F.pos     -0.10990    0.03161  -3.477 0.00101 **
## eqB11I.pre     613.89664   141.95438   4.325 6.63e-05 ***
## eqB11I.blm     613.65610   141.93831   4.323 6.65e-05 ***
## eqB11I.pos     613.47623   141.96140   4.321 6.70e-05 ***
## eqB11I.pos.np    0.38638    0.16766   2.305 0.02506 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.299 on 54 degrees of freedom
```

```

## Multiple R-squared:  0.4612, Adjusted R-squared:  0.3914
## F-statistic: 6.604 on 7 and 54 DF,  p-value: 1.197e-05

###~~~~ With varying variables ~~~ neither of which appear as relevant at all...
summary(regsubsets( apisAb ~ temp+ bloom.index + eqB11 + eqB11.np + eqB11.fun + eqB11.ins + eqB11F.pre + eqB11F.blm + eqB11F.pos + eqB11I.pre + eqB11I.blm + eqB11I.pos + eqB11I.pos.np + eqB11T.blm + eqB11T.pos + size + hive.acr + X2000nat, data = data, nvmax = 6))

## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
## force.in = force.in, : 2 linear dependencies found

## Reordering variables and trying again:

## Subset selection object
## Call: regsubsets.formula(apisAb ~ temp + bloom.index + eqB11 + eqB11.np +
## eqB11.fun + eqB11.ins + eqB11.ins.np + eqB11F.pre + eqB11F.blm +
## eqB11F.pos + eqB11I.pre + eqB11I.blm + eqB11I.pos + eqB11I.pos.np +
## eqB11T.blm + eqB11T.pos + size + hive.acr + X2000nat, data = data,
## nvmax = 6)
## 19 Variables (and intercept)
##           Forced in Forced out
## temp                FALSE      FALSE
## bloom.index          FALSE      FALSE
## eqB11                FALSE      FALSE
## eqB11.np             FALSE      FALSE
## eqB11.fun            FALSE      FALSE
## eqB11.ins            FALSE      FALSE
## eqB11F.pre           FALSE      FALSE
## eqB11F.blm           FALSE      FALSE
## eqB11F.pos           FALSE      FALSE
## eqB11I.pre           FALSE      FALSE
## eqB11I.blm           FALSE      FALSE
## eqB11I.pos           FALSE      FALSE
## eqB11I.pos.np        FALSE      FALSE
## eqB11T.blm           FALSE      FALSE
## size                 FALSE      FALSE
## hive.acr             FALSE      FALSE
## X2000nat             FALSE      FALSE
## eqB11.ins.np         FALSE      FALSE
## eqB11T.pos           FALSE      FALSE
## 1 subsets of each size up to 7
## Selection Algorithm: exhaustive
##           temp bloom.index eqB11 eqB11.np eqB11.fun eqB11.ins
## 1 ( 1 ) " " " " " " " " " " " "
## 2 ( 1 ) " " " " " " " " " " " "
## 3 ( 1 ) " " " " " " " " " " " "
## 4 ( 1 ) " " " " " " " " " " " "
## 5 ( 1 ) " " " " " " " " " " " "
## 6 ( 1 ) " " " " " " " " " " " "
## 7 ( 1 ) " " " " " " " " " " " "
##           eqB11.ins.np eqB11F.pre eqB11F.blm eqB11F.pos eqB11I.pre
## 1 ( 1 ) " " " " " " " " " " " "
## 2 ( 1 ) " " " " " " " " " " " "
## 3 ( 1 ) " " " " " " " " " " " "
## 4 ( 1 ) " " " " " " " " " " " "

```

```
## 5 ( 1 ) " " " " " " " "
## 6 ( 1 ) "*" " " " " " " " "
## 7 ( 1 ) "*" " " " " " " " "
##      eiqB11I.blm eiqB11I.pos eiqB11I.pos.np eiqB11T.blm eiqB11T.pos
## 1 ( 1 ) " " " " " " " "
## 2 ( 1 ) " " " " " " " "
## 3 ( 1 ) "*" " " " " " " " "
## 4 ( 1 ) "*" " " " " " " " "
## 5 ( 1 ) "*" " " "*" " " " "
## 6 ( 1 ) "*" "*" " " " " " "
## 7 ( 1 ) "*" "*" "*" " " " "
##      size hive.acr X2000nat
## 1 ( 1 ) " " " " " "
## 2 ( 1 ) " " " " " "
## 3 ( 1 ) " " " " " "
## 4 ( 1 ) " " " " " "
## 5 ( 1 ) " " " " " "
## 6 ( 1 ) " " " " " "
## 7 ( 1 ) " " " " " "
```

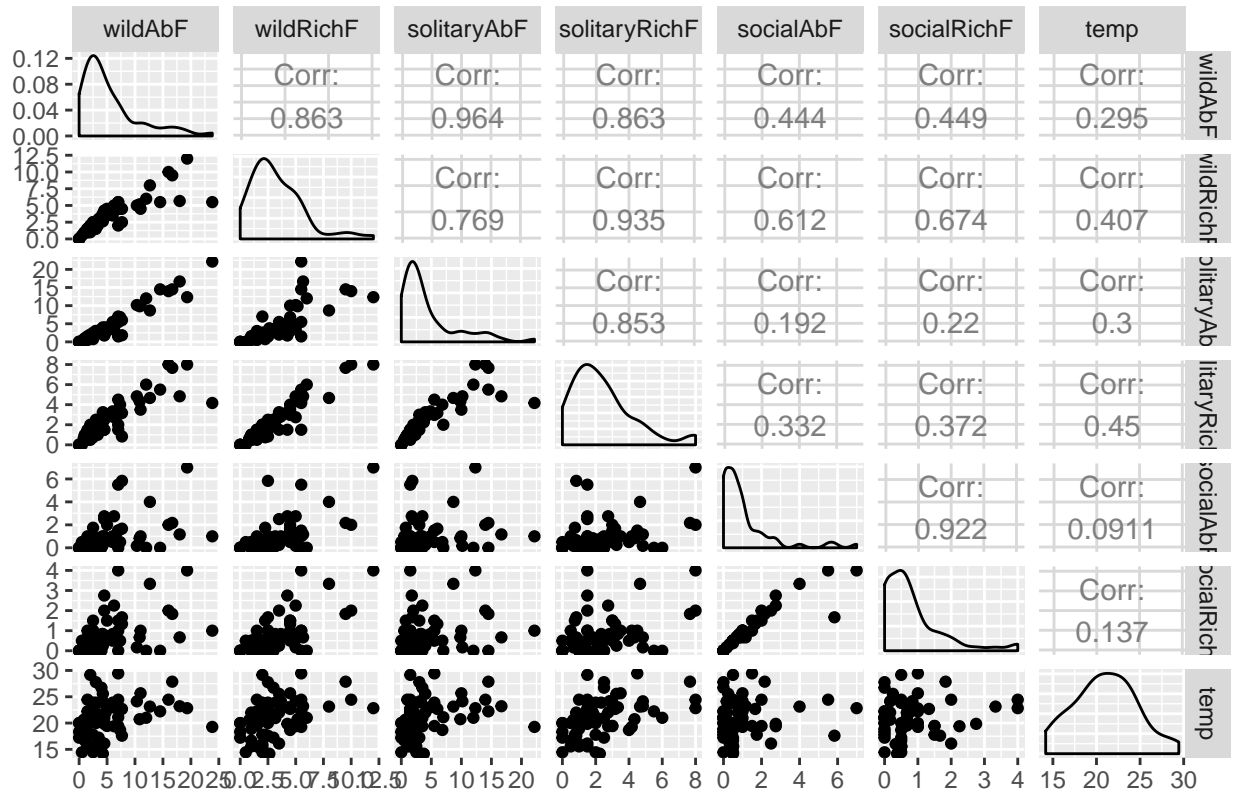
```
##Both stepwise regressions so the same predictors as the best
hist_resid <- ggplot(data=data, aes(lm_non_vary$residuals)) +
  geom_histogram(binwidth = 1, color = "black", fill = "purple4") +
  theme(panel.background = element_rect(fill = "white"),
        axis.line.x=element_line(),
        axis.line.y=element_line()) +
  ggtitle("Histogram for Model Residuals")
```

#Probably need to look into best ways to plot/compare these

GGTally

```
bees_data <- ggpairs(data = data, columns = 7:13, title = "Bees Data")
bees_data
```

Bees Data



```
predictor_data <- ggpairs(data = data, columns = 13:34, title = "Predictor Data")
predictor_data
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removing 1 row that contained a missing value
## Warning: Removed 1 rows containing missing values (geom_point).
## Warning: Removed 1 rows containing non-finite values (stat_density).
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removing 1 row that contained a missing value
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removing 1 row that contained a missing value
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removing 1 row that contained a missing value
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removing 1 row that contained a missing value
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removing 1 row that contained a missing value
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removing 1 row that contained a missing value
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removing 1 row that contained a missing value
```



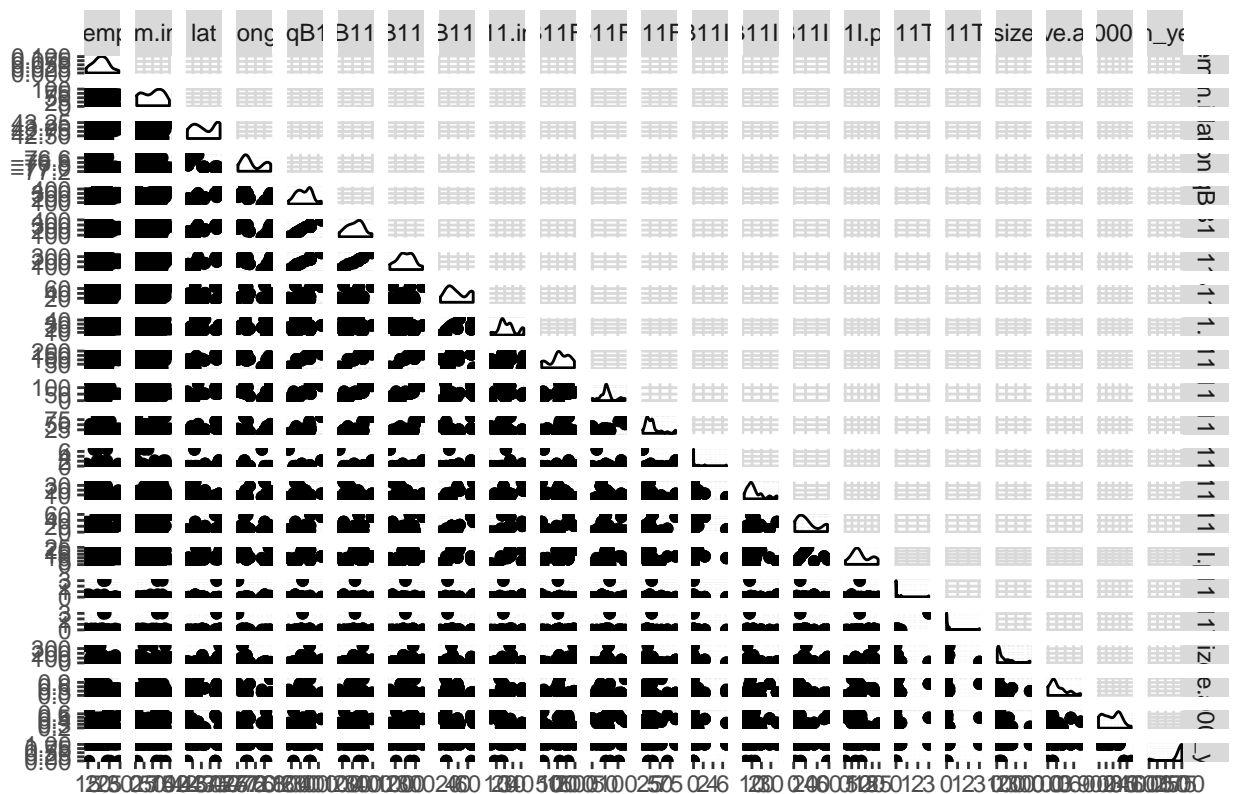
```

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removing 1 row that contained a missing value
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removing 1 row that contained a missing value
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removing 1 row that contained a missing value
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removing 1 row that contained a missing value
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removing 1 row that contained a missing value
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removing 1 row that contained a missing value
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removing 1 row that contained a missing value
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removing 1 row that contained a missing value
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removing 1 row that contained a missing value
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removing 1 row that contained a missing value
## Warning: Removed 1 rows containing missing values (geom_point).
## Warning: Removed 1 rows containing missing values (geom_point).
## Warning: Removed 1 rows containing missing values (geom_point).
## Warning: Removed 1 rows containing missing values (geom_point).
## Warning: Removed 1 rows containing missing values (geom_point).
## Warning: Removed 1 rows containing missing values (geom_point).
## Warning: Removed 1 rows containing missing values (geom_point).
## Warning: Removed 1 rows containing missing values (geom_point).
## Warning: Removed 1 rows containing missing values (geom_point).
## Warning: Removed 1 rows containing missing values (geom_point).
## Warning: Removed 1 rows containing missing values (geom_point).

```

```
## Warning: Removed 1 rows containing missing values (geom_point).
## Warning: Removed 1 rows containing missing values (geom_point).
## Warning: Removed 1 rows containing missing values (geom_point).
## Warning: Removed 1 rows containing missing values (geom_point).
## Warning: Removed 1 rows containing missing values (geom_point).
## Warning: Removed 1 rows containing missing values (geom_point).
## Warning: Removed 1 rows containing missing values (geom_point).
## Warning: Removed 1 rows containing missing values (geom_point).
## Warning: Removed 1 rows containing missing values (geom_point).
```

Predictor Data



Looking at individual variables

```
day_data <- data %>%
  filter(day %in% c("1", "2")) %>%
  mutate(year_day = ifelse(str_detect(day, "1") & str_detect(year, "3"), '11',
    ifelse(str_detect(day, "2") & str_detect(year, "3"), '12',
```

```

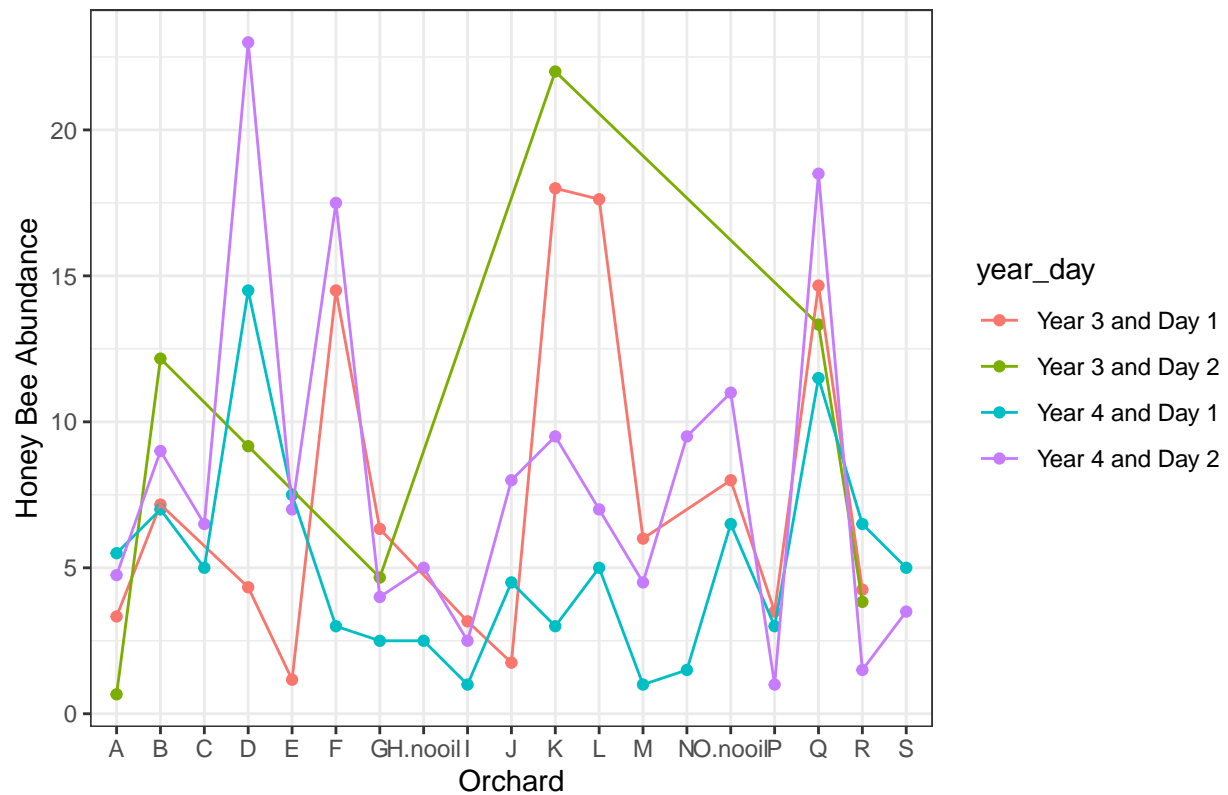
        ifelse(str_detect(day, "1") & str_detect(year, "4"), '21',
               ifelse(str_detect(day, "2") & str_detect(year, "4"), '22', "55"))))) %>%
mutate(year_day = factor(year_day, labels = c("Year 3 and Day 1", "Year 3 and Day 2", "Year 4 and Day 1", "Year 4 and Day 2"),
na.omit())

#checking data as there should be a result for each
tibble(counts = c(count(day_data %>%
  filter(year_day == "Year 3 and Day 1")),
count(day_data %>%
  filter(year_day == "Year 3 and Day 2")),
count(day_data %>%
  filter(year_day == "Year 4 and Day 1")),
count(day_data %>%
  filter(year_day == "Year 4 and Day 2")))) %>%
  view()

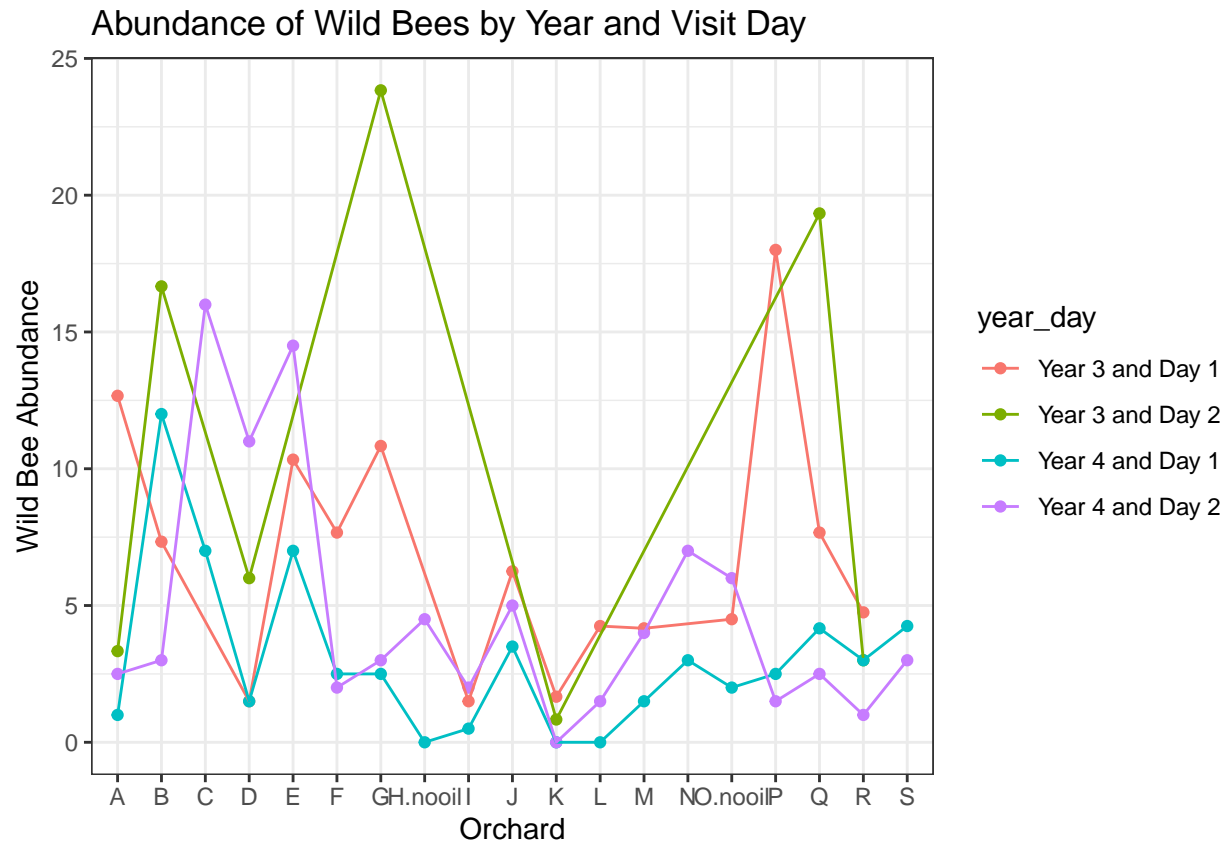
honey_by_year_day <- day_data %>%
  ggplot(aes(x = orchard, y = apisAb, colour = year_day)) +
  geom_point() +
  geom_line(aes(group = year_day)) +
  theme_bw() +
  ggtitle("Abundance of Honey Bees by Year and Visit Day") +
  ylab("Honey Bee Abundance") +
  xlab("Orchard")
honey_by_year_day

```

Abundance of Honey Bees by Year and Visit Day



```
wild_by_year_day <- day_data %>%
  ggplot(aes(x = orchard, y = wildAbF, colour = year_day)) +
  geom_point() +
  geom_line(aes(group = year_day)) +
  theme_bw() +
  ggtitle("Abundance of Wild Bees by Year and Visit Day") +
  ylab("Wild Bee Abundance") +
  xlab("Orchard")
wild_by_year_day
```



```
wild_temp_by_year_day <- day_data %>%
  ggplot(aes(x = temp, y = wildAbF, colour = year_day)) +
  geom_point() +
  geom_line(aes(group = year_day)) +
  theme_classic()

honey_temp_by_year_day <- day_data %>%
  ggplot(aes(x = temp, y = apisAb, colour = year_day)) +
  geom_point() +
  geom_line(aes(group = year_day)) +
  theme_classic()

#think of somehow quantifying bees as the "yield" variable
decision_data <- tibble(
  before = c("Apply Fungicide", "Apply Insecticide", "Apply Both", "Apply Nothing"),
  during = c("a", "b", "c", "d"),
  bee_yield = c(300, 200, 300, 400)
)
```

Code post 25/10

```
data_2012 <- day_data %>%
  filter(year == 4)

#Looking at violin plots of the data
```

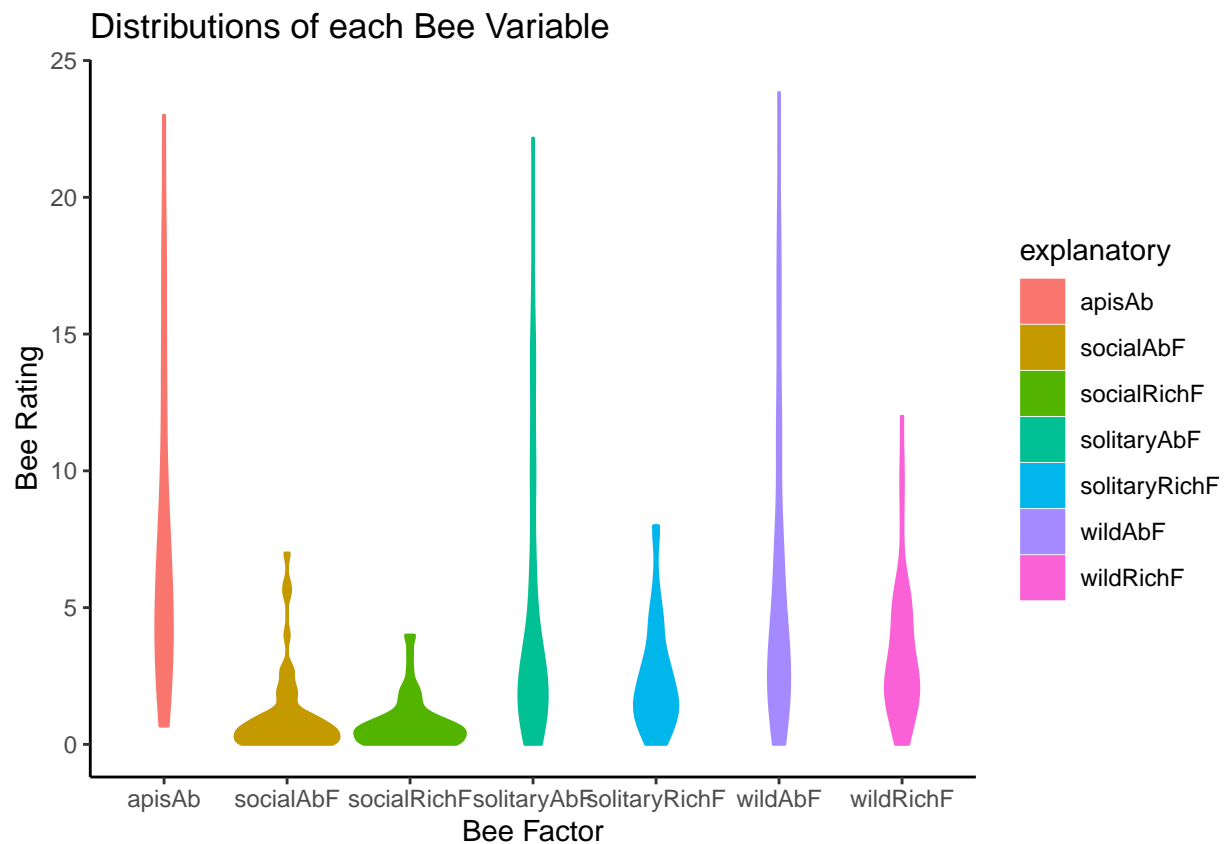
```

violin_data <- day_data %>%
  select(c("region":"X2000nat")) %>%
  gather(key = "explanatory", value = "value", -region, -day) %>%
  na.omit()

violin_plot <- function(x, xlab, ylab, title){
  x %>% ggplot() +
    geom_violin(aes(x = factor(explanatory), y = value, fill = explanatory, colour = explanatory)) +
    ylab(ylab) +
    xlab(xlab) +
    ggtitle(title) +
    theme(legend.position = "none") +
    theme_classic()
}

violin_plot_beets <- violin_data %>%
  subset(explanatory %in% colnames(day_data[4:12])) %>%
  violin_plot("Bee Factor", "Bee Rating", "Distributions of each Bee Variable")
violin_plot_beets

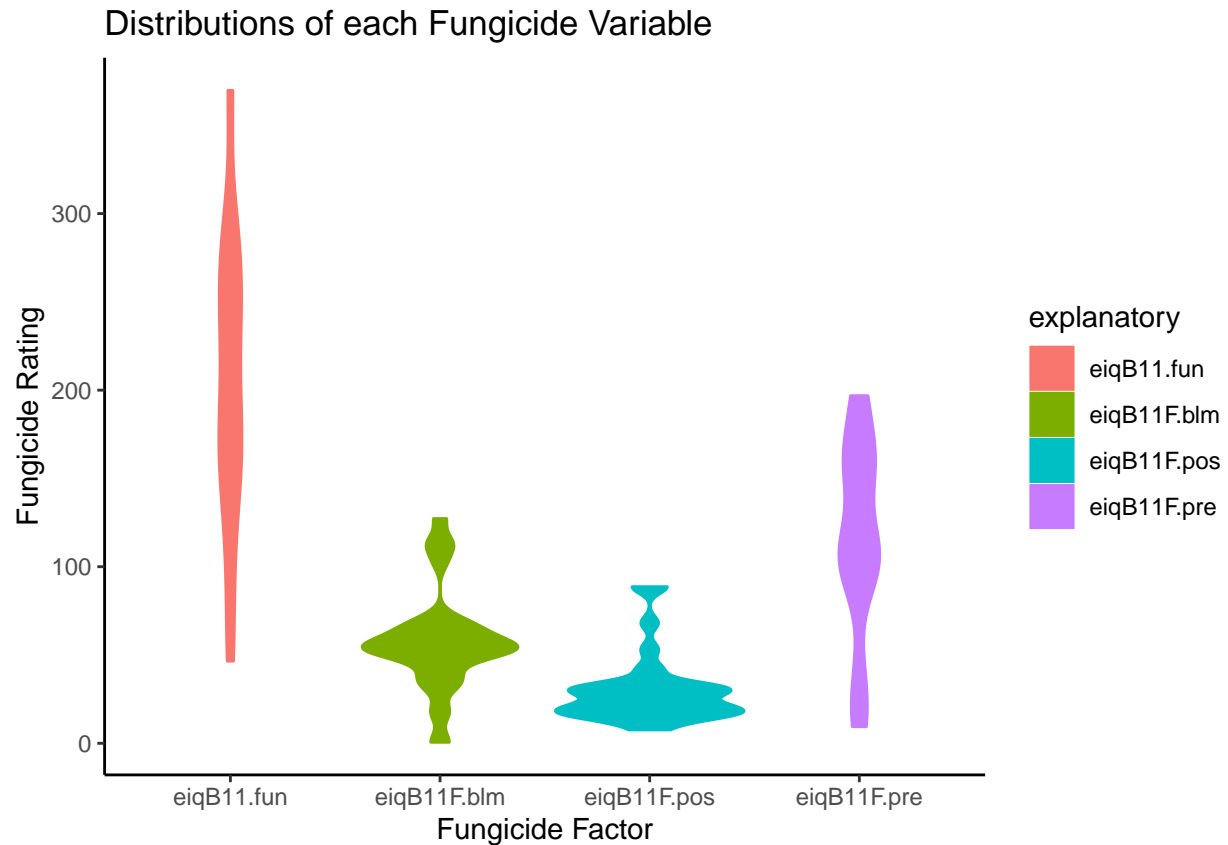
```



```

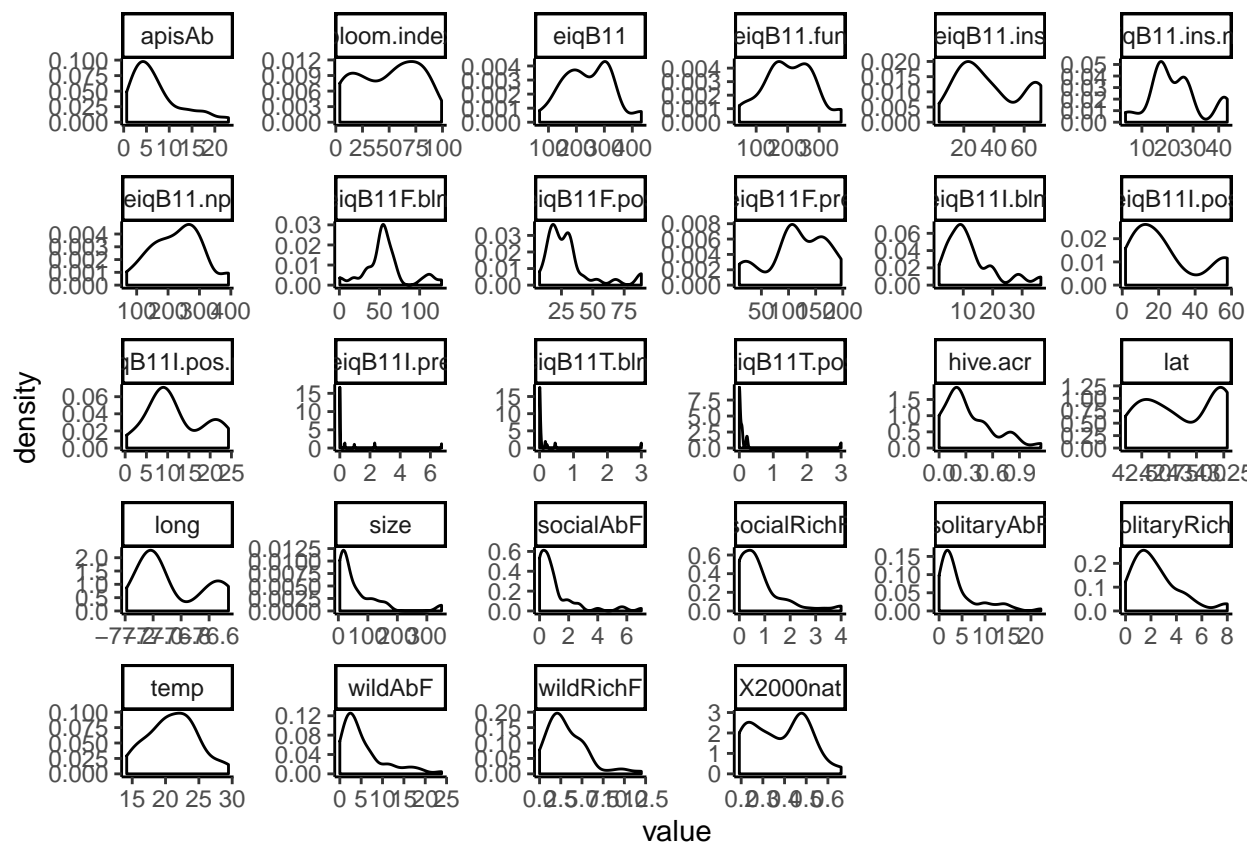
violin_plot_fungicides <- violin_data %>%
  subset(explanatory %in% colnames(day_data[c(19,22,23,24)])) %>%
  violin_plot("Fungicide Factor", "Fungicide Rating", "Distributions of each Fungicide Variable")
violin_plot_fungicides

```



```
violin_plot_insecticide <- violin_data %>%
  subset(explanatory %in% colnames(day_data[c(20,21,25:28)])) %>%
  violin_plot("Insecticide Factor", "Insecticide Rating", "Distributions of each Insecticide Variable")

#Distributions of all variables
distributions <- violin_data %>%
  ggplot() +
  geom_density(aes(value)) +
  facet_wrap(~explanatory, scales = "free") +
  theme_classic()
distributions
```



```
#Bloom values - particularly not ideal as it looks like the day 1 / day 2 actually is just random
#lol not even in relation to bloom levels
bloom_plot <- day_data %>%
  group_by(region, day) %>%
  mutate(group = paste(region, day)) %>%
  ggplot(aes(x= group, y = bloom.index)) +
  geom_violin(aes(fill = group, colour = group)) +
  geom_jitter(height = 0, width = 0.05) +
  theme_classic() +
  theme(legend.position = "none") +
  labs(x = "Region and Day", y = "Bloom Index", title = "Violin Plot of Bloom Index by Region and Day")
bloom_plot
```