For Botanist

Stephen Brownsey

03/11/2019

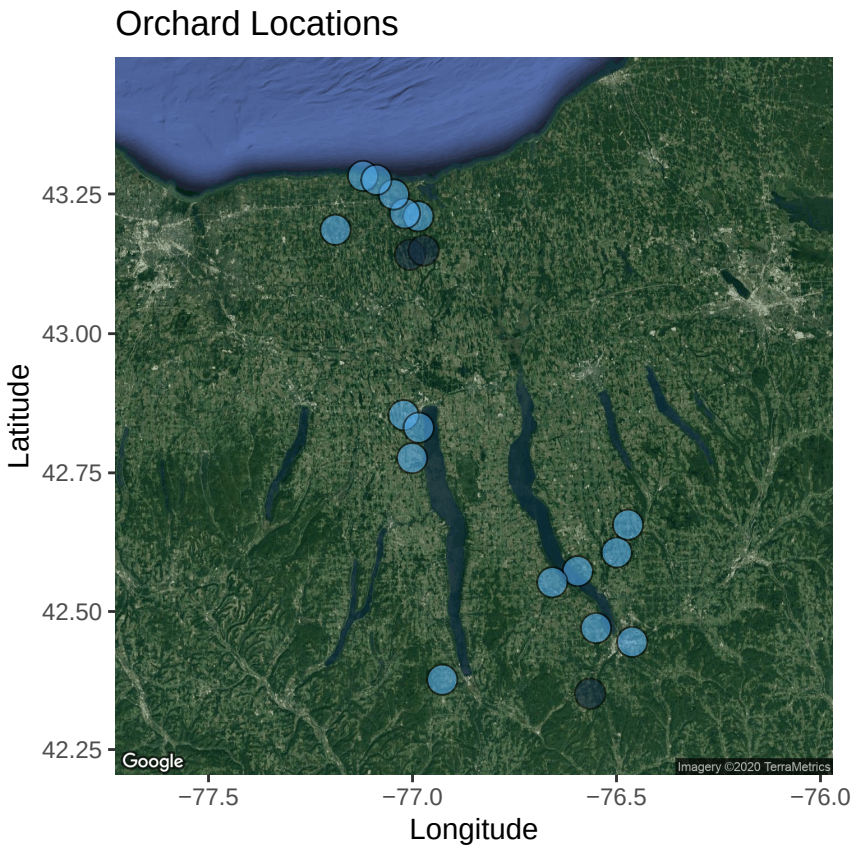
Introduction

The code in this .rmd was used for the discussion with the botanist on the 03/11/2019. It also contains extra EDA on top of the EDA already undertaken as part of the exploratory_data.rmd file. At the latter end the first EDA for clustering was undertaken, the clustering EDA which has utilised in the rest of the project has remained uncommented. Some method were considered but didn't work as desired and have been commented out but the code kept in to demonstrate the thought process - the clustering.rmd the full code for the clustering analysis.

Map Analysis

This is an image of where each of the orchards is located, more for interest but would you think that they would go around taking readings of different orchards based on similar days when they are close to each other? If so, would it make sense to apply some models to just one area?

```
## Source : https://maps.googleapis.com/maps/api/staticmap?center=42.85269,-76.850661&zoom=9&size=640x6
```



The variables in general are very correlated, as you would expect, but this means that for the decision models I would want to be looking at a subset of these but from a botanist's point of view which variables would she consider best? Not that clear in a .pdf but lots are upwards of 0.8/0.9 correlated.

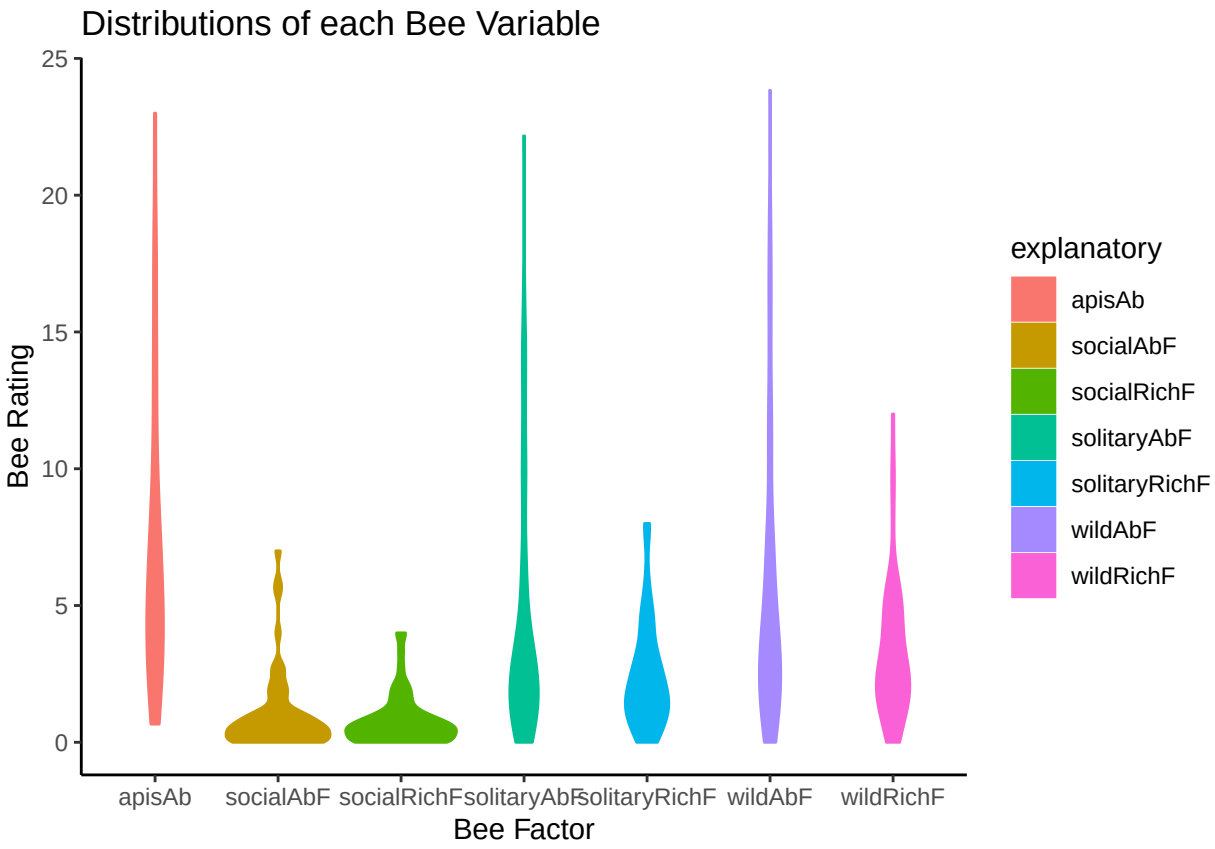
GGTally

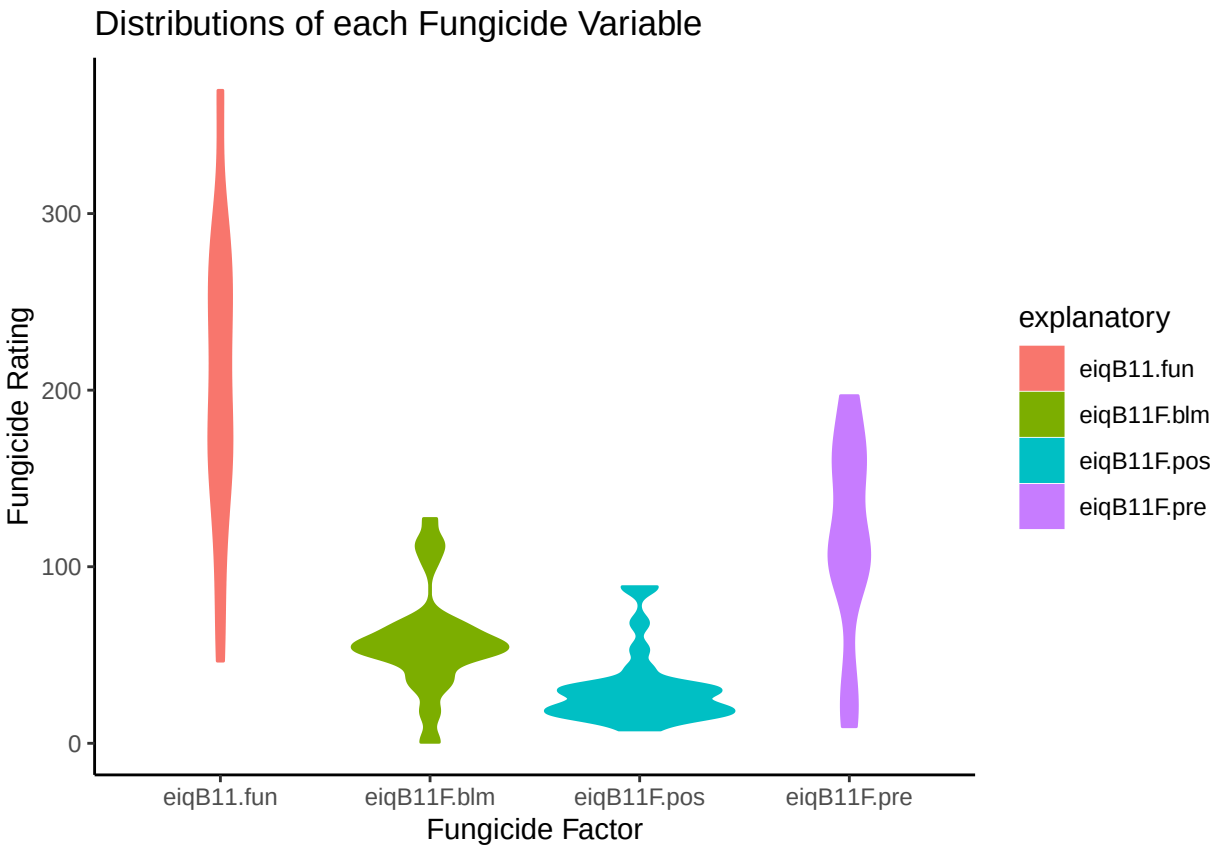
Takes about 5-10 mins to run - produces all the various correlation and associated plots between variables

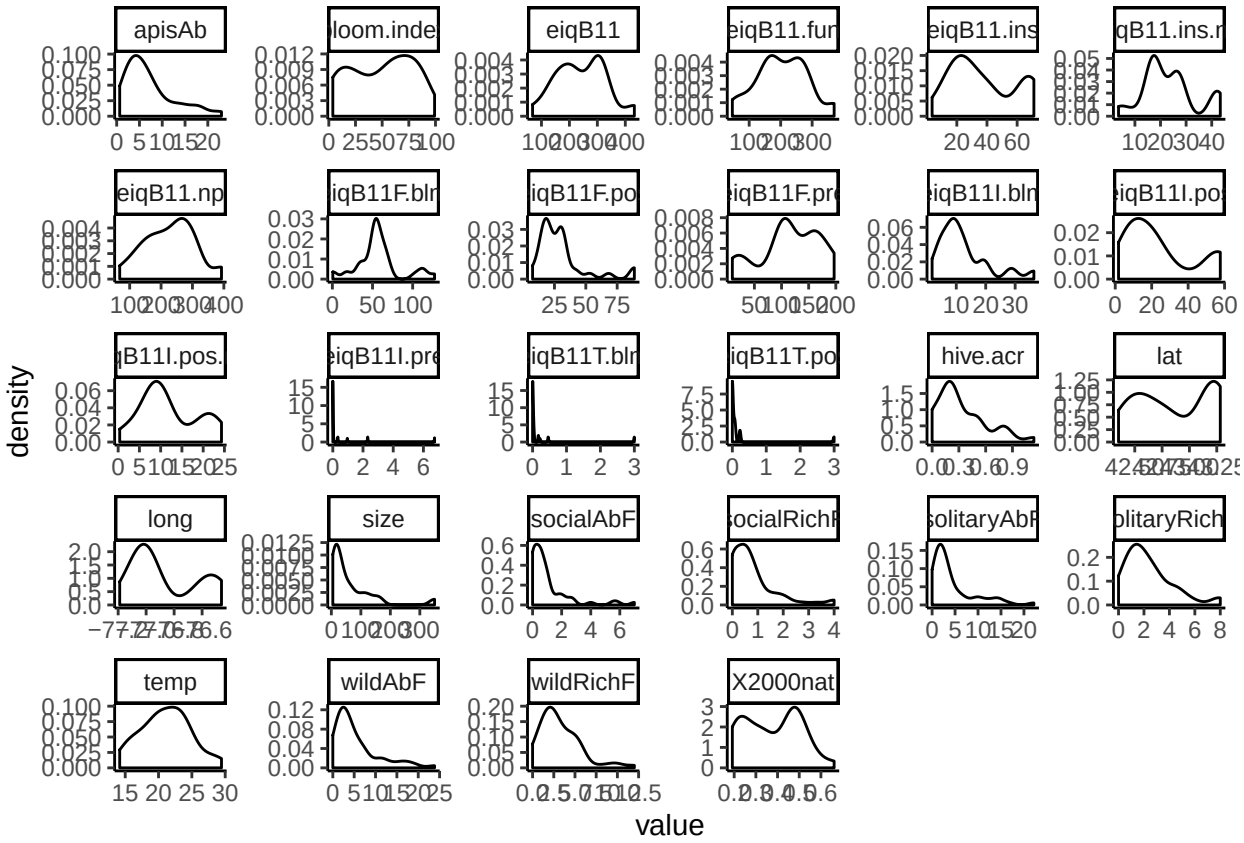
Looking at individual variables

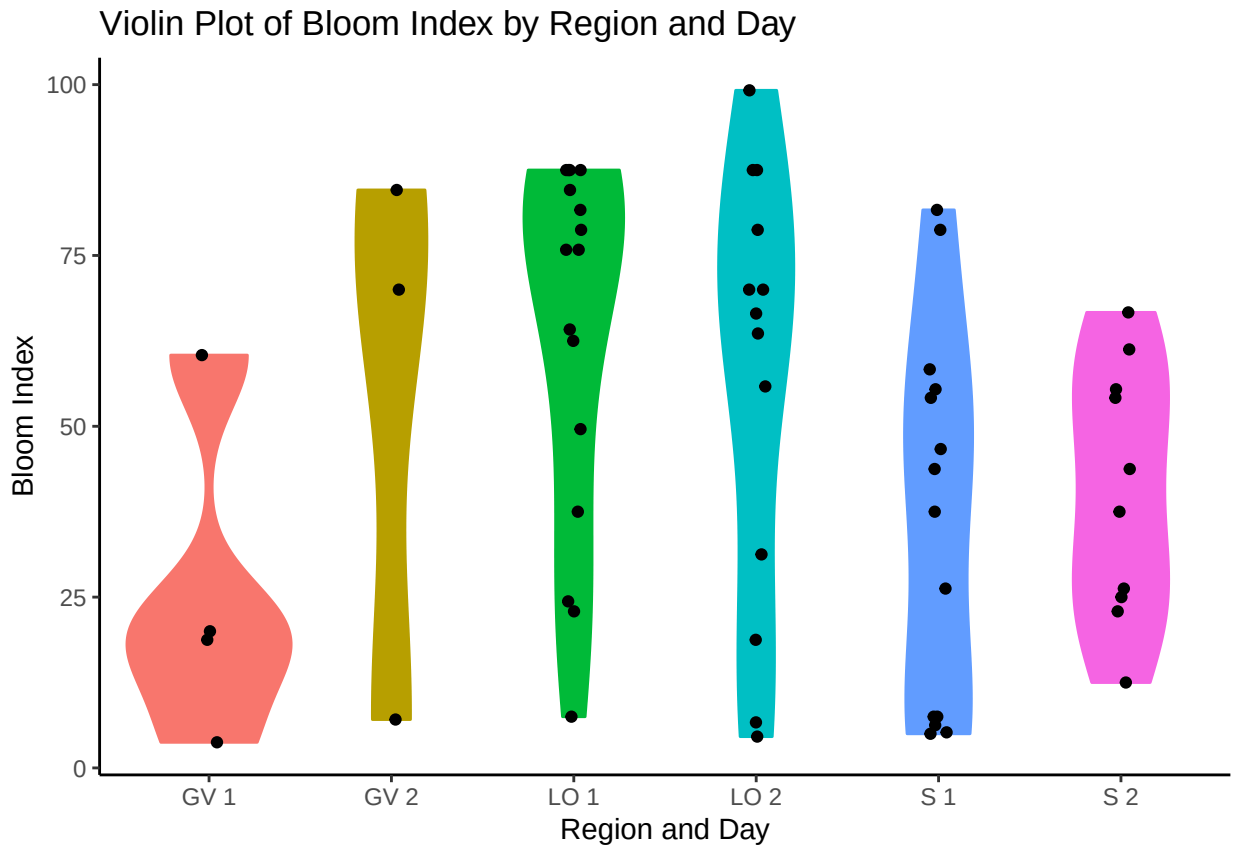
Code post 25/10

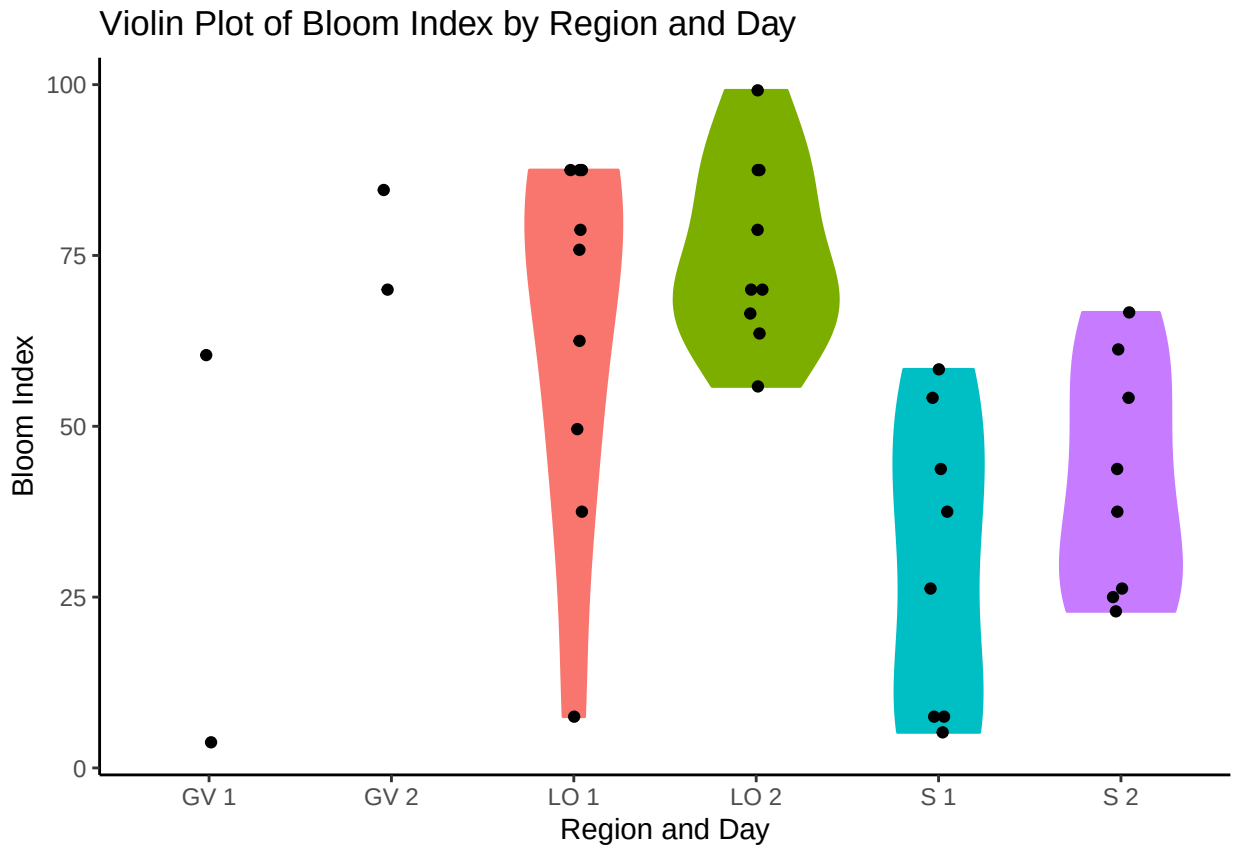
Violin plots of the data - I am a fan of what these show, but in particular note the one for bloom.index (at the bottom), I had thought that day 1 would be $<$ 50% bloom and day 2 post 50% bloom but this clearly shows this is not the case - thoughts? Have also included a distributions plot of the variables in the dataset for interest as it could be helpful when thinking about these things.


Distributions of each Bee Variable
Bee Rating
25
20
15
10
5
0
apisAb
socialAbF
socialRichF
solitaryAbF
solitaryRichF
wildAbF
wildRichF
Bee Factor
explanatory
apisAb
socialAbF
socialRichF
solitaryAbF
solitaryRichF
wildAbF
wildRichF


Distributions of each Fungicide Variable
Fungicide Rating
300
200
100
0
eiqB11.fun
eiqB11F.blm
eiqB11F.pos
eiqB11F.pre
Fungicide Factor
explanatory
eiqB11.fun
eiqB11F.blm
eiqB11F.pos
eiqB11F.pre

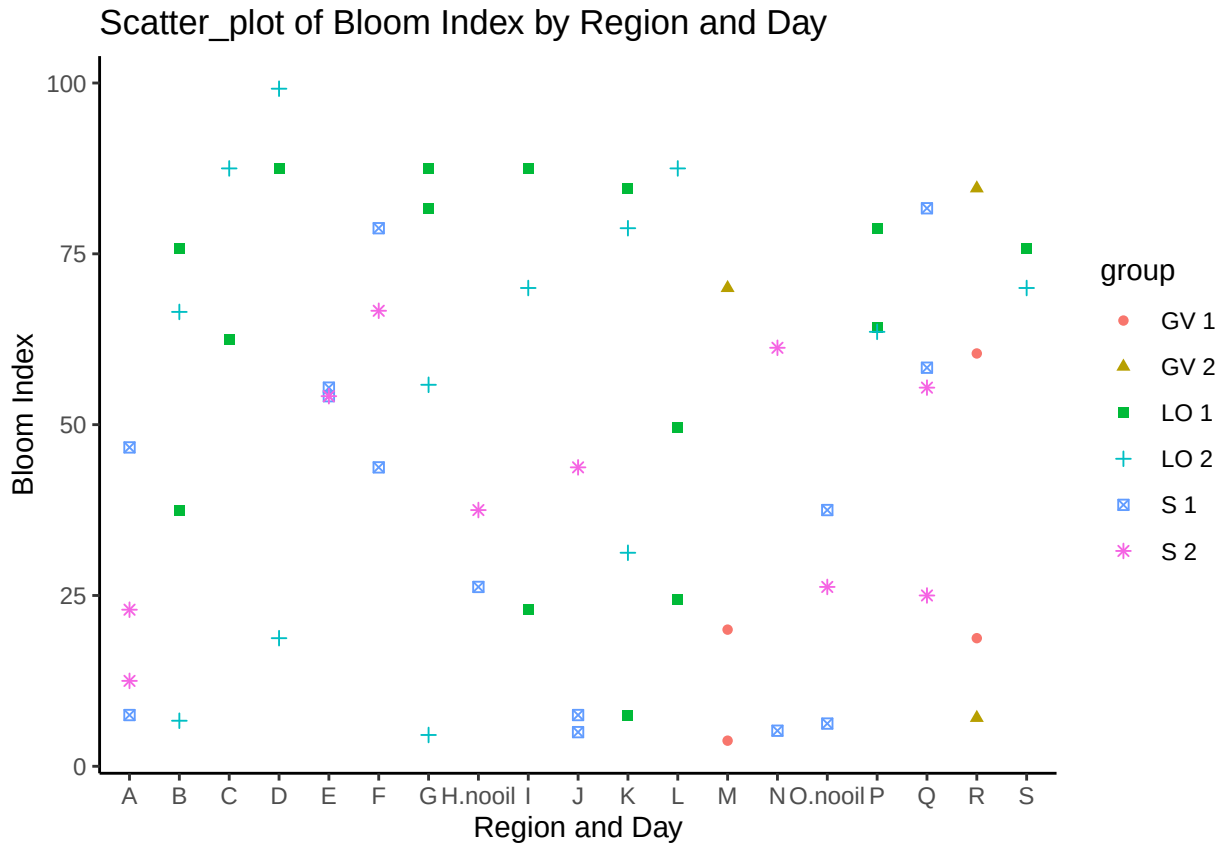

density
value
apisAb
bloom.inde
eiqB11
eiqB11.fun
eiqB11.ins
qB11.ins.r
eiqB11.np
iqB11F.blr
iqB11F.po
iqB11F.pr
iqB11I.bln
iqB11I.po
qB11I.pos.
eiqB11I.pre
iqB11T.bln
iqB11T.po
hive.acr
lat
long
size
socialAbF
ocialRichF
olitaryAbF
olitaryRich
temp
wildAbF
wildRichF
X2000nat

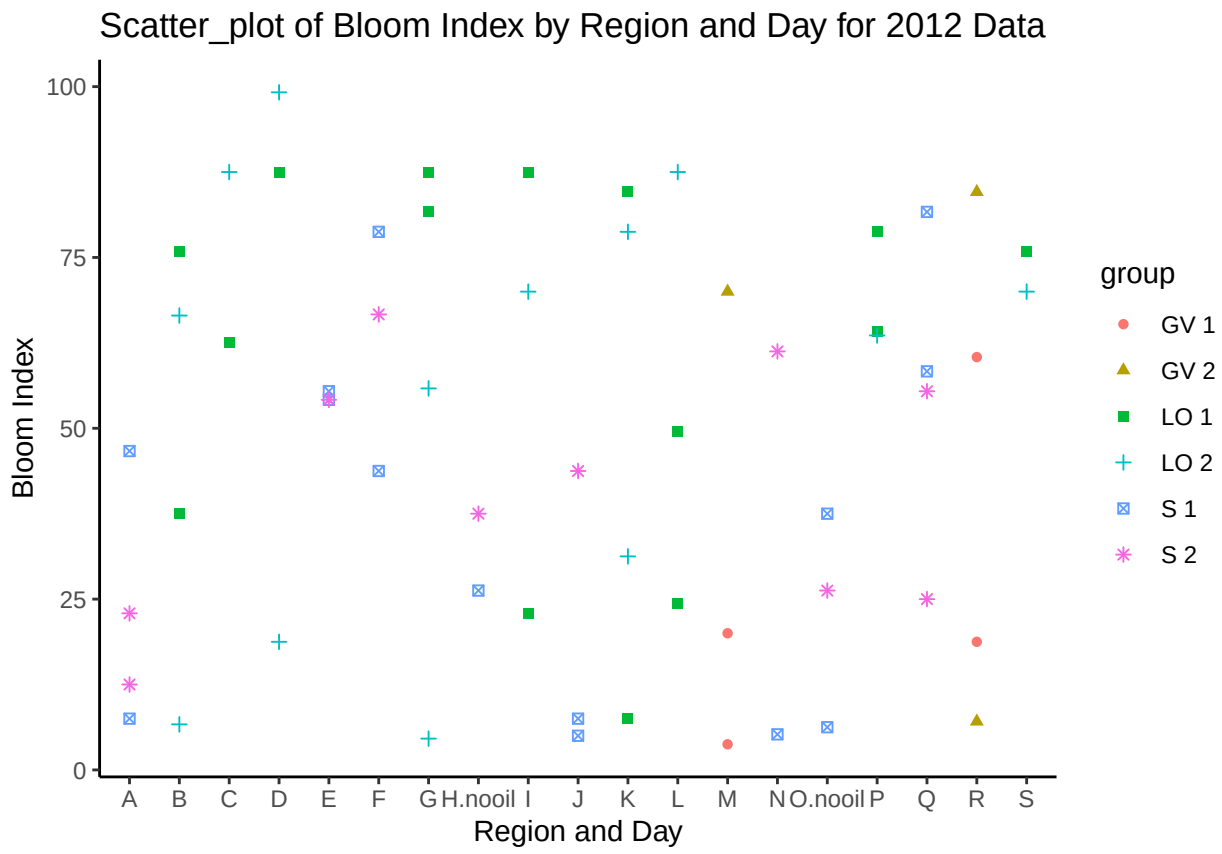

Violin Plot of Bloom Index by Region and Day
Bloom Index
100
75
50
25
0
GV 1
GV 2
LO 1
LO 2
S 1
S 2
Region and Day

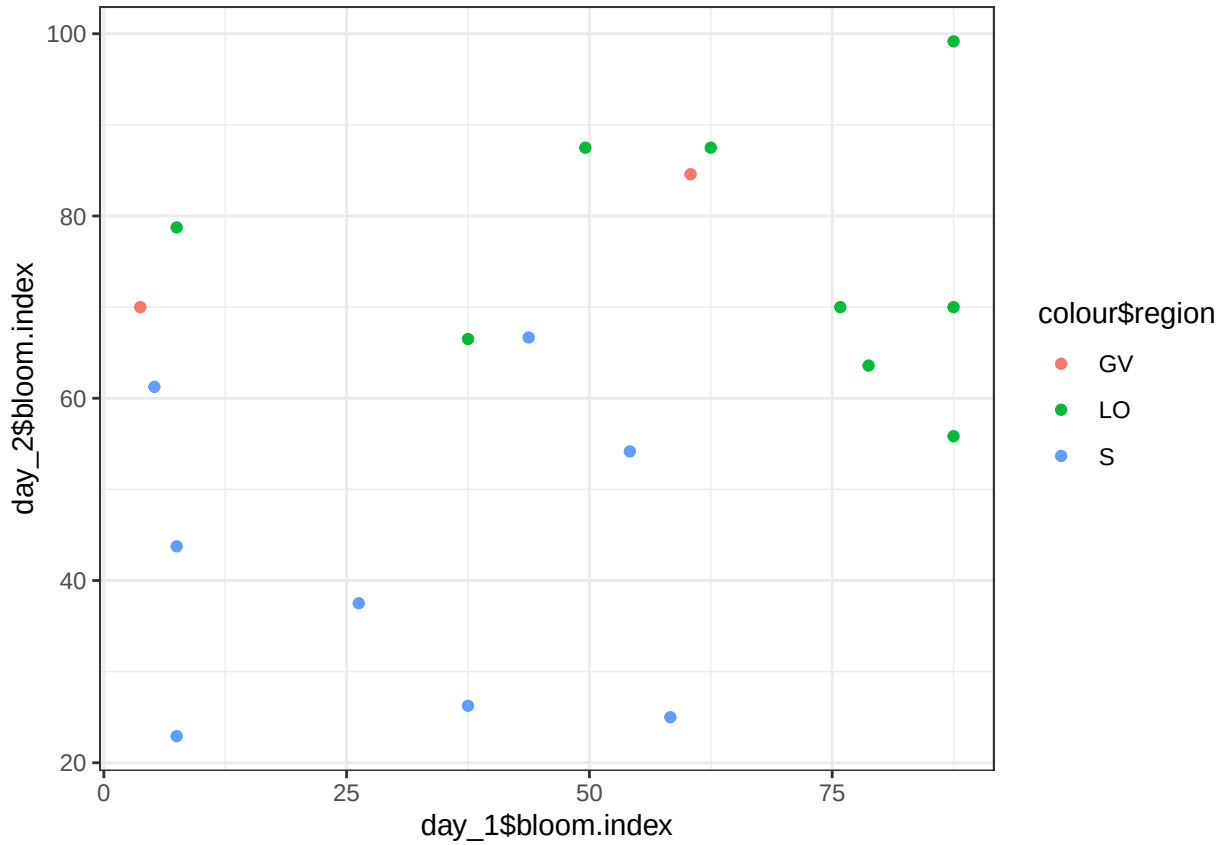

Violin Plot of Bloom Index by Region and Day
Bloom Index
100
75
50
25
0
GV 1
GV 2
LO 1
LO 2
S 1
S 2
Region and Day

Graphics from botanists chat

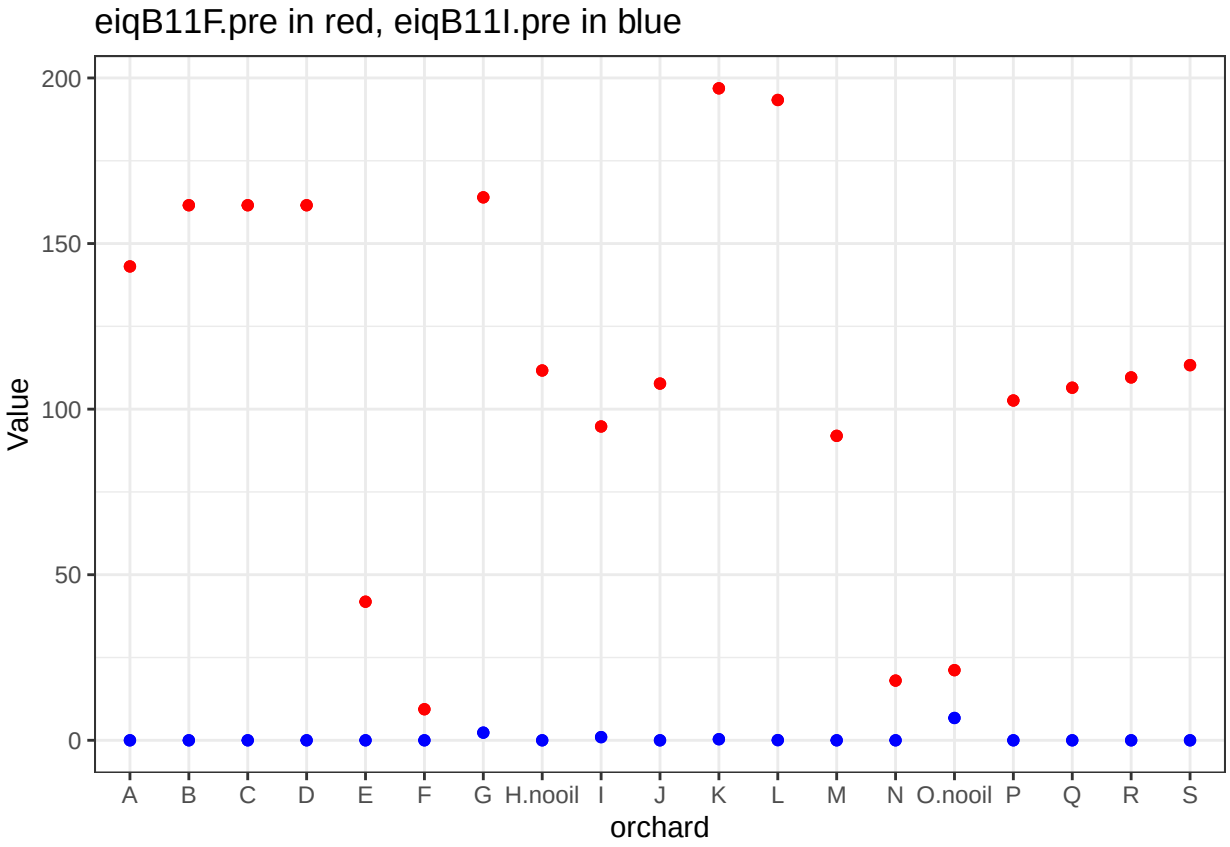
Extra graphics and analysis undertaken based on the chat with the botanist to see relationships.

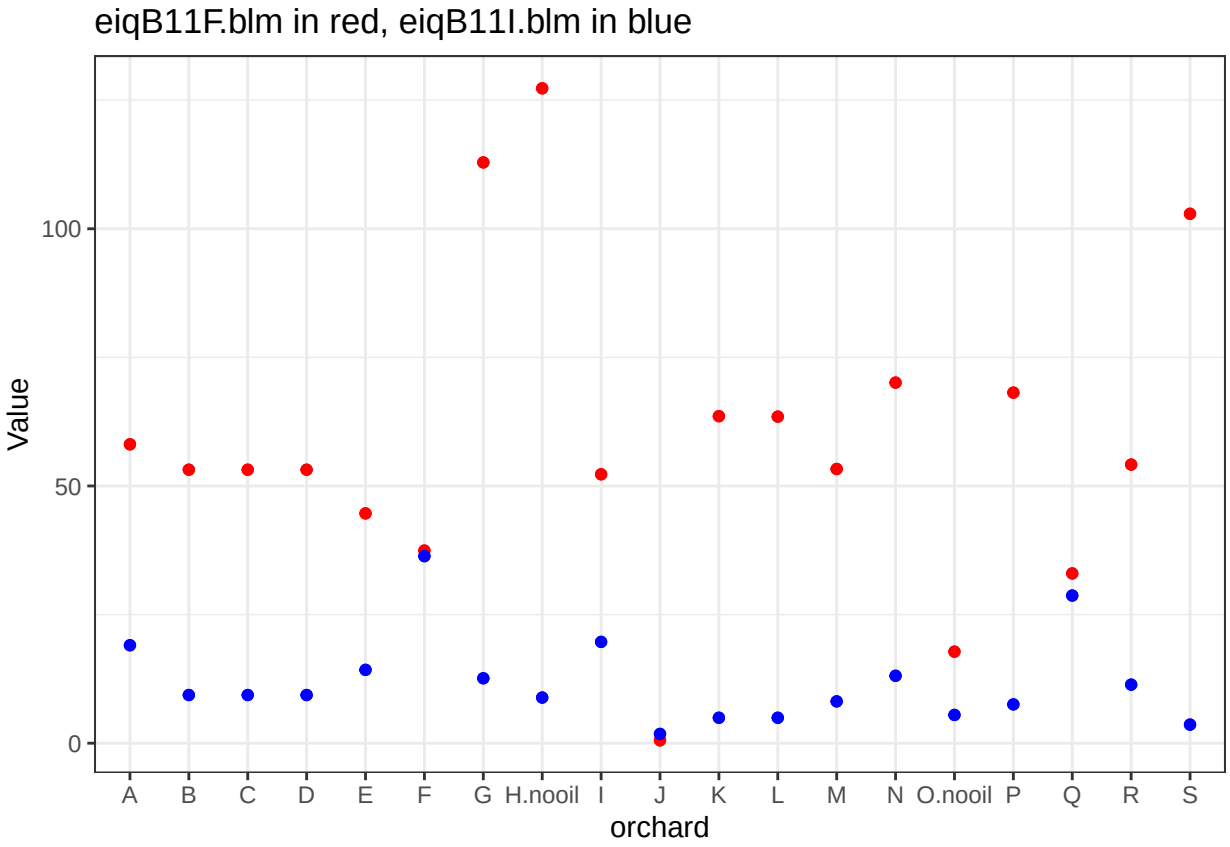


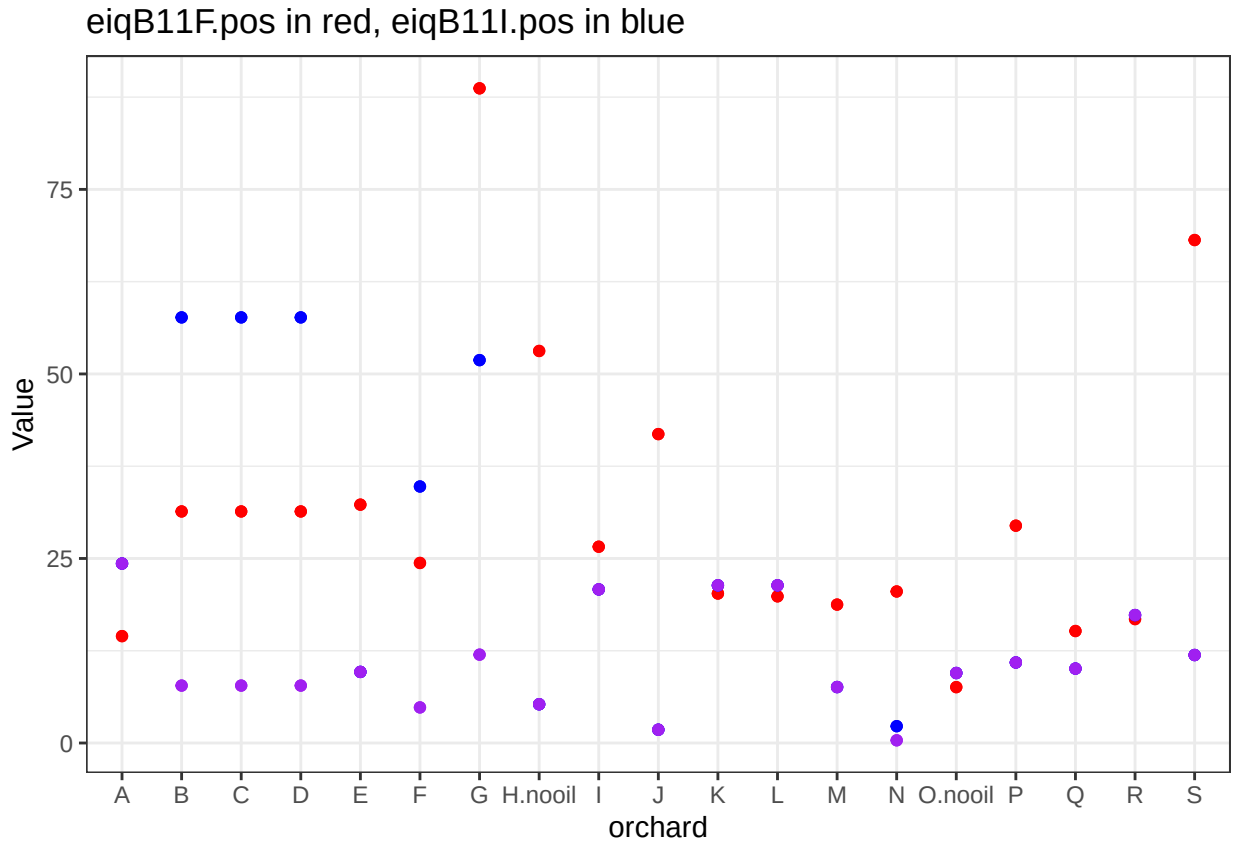

Scatter_plot of Bloom Index by Region and Day for 2012 Data
Bloom Index
100
75
50
25
0
A
B
C
D
E
F
G
H.nooil
I
J
K
L
M
N
O.nooil
P
Q
R
S
Region and Day
group
GV 1
GV 2
LO 1
LO 2
S 1
S 2



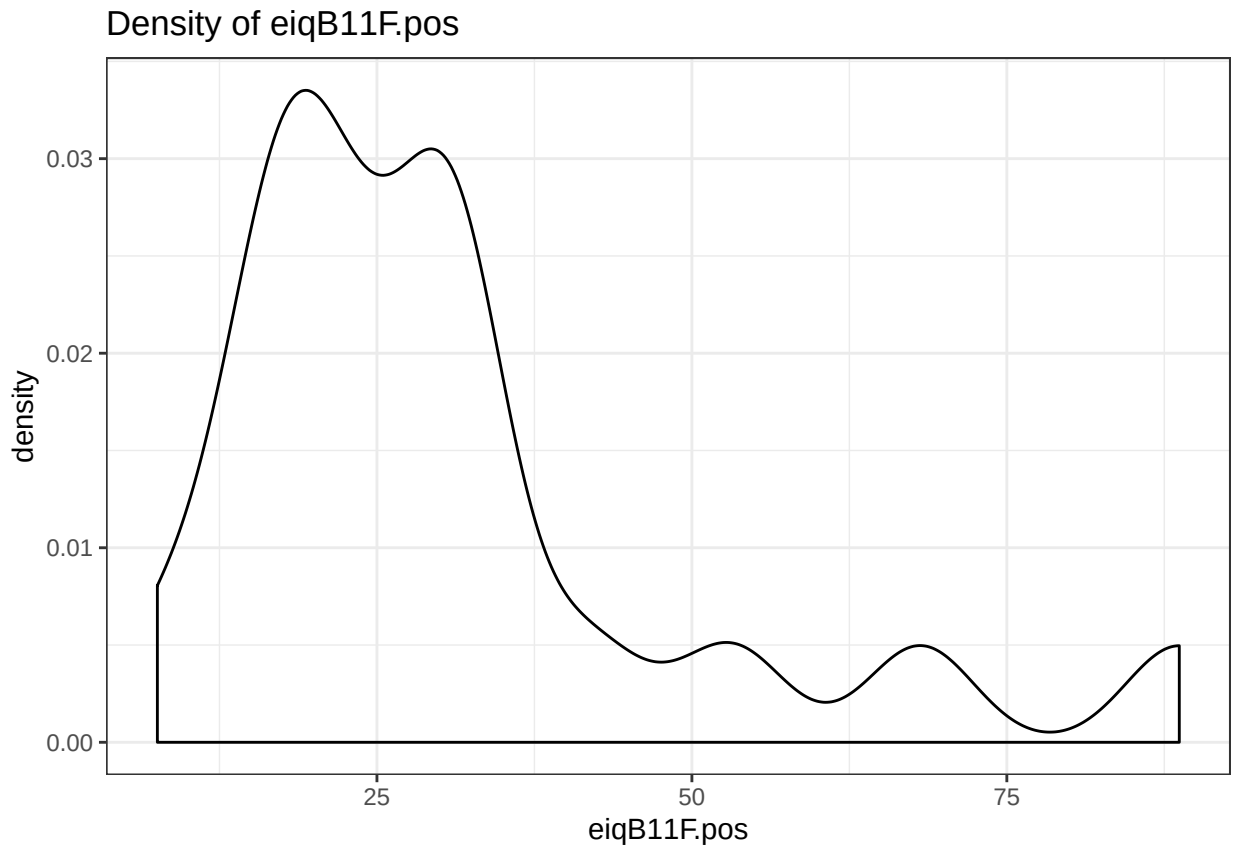
The first step for us is to understand the chemicals applications better, including the combinations. The variables in the data are scores developed by the authors that capture the impact the have, so that is quite ready to use and comparable, need to further investigate this.


eiqB11F.pre in red, eiqB11I.pre in blue
Value
200
150
100
50
0
A
B
C
D
E
F
G
H.nooil
I
J
K
L
M
N
O.nooil
P
Q
R
S
orchard

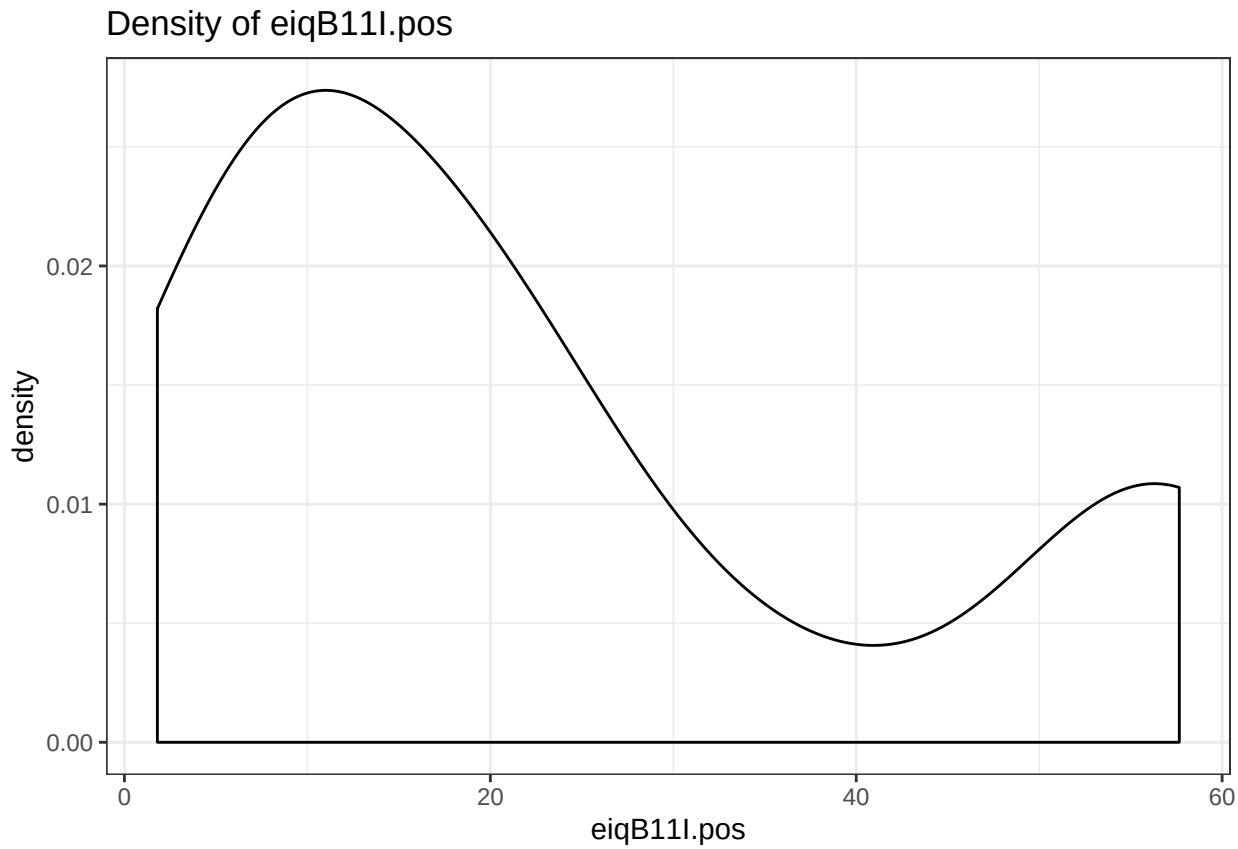

eiqB11F.blm in red, eiqB11I.blm in blue
Value
100
50
0
A
B
C
D
E
F
G
H.nooil
I
J
K
L
M
N
O.nooil
P
Q
R
S
orchard


eiqB11F.pos in red, eiqB11I.pos in blue
Value
75
50
25
0
A
B
C
D
E
F
G
H.nooil
I
J
K
L
M
N
O.nooil
P
Q
R
S
orchard

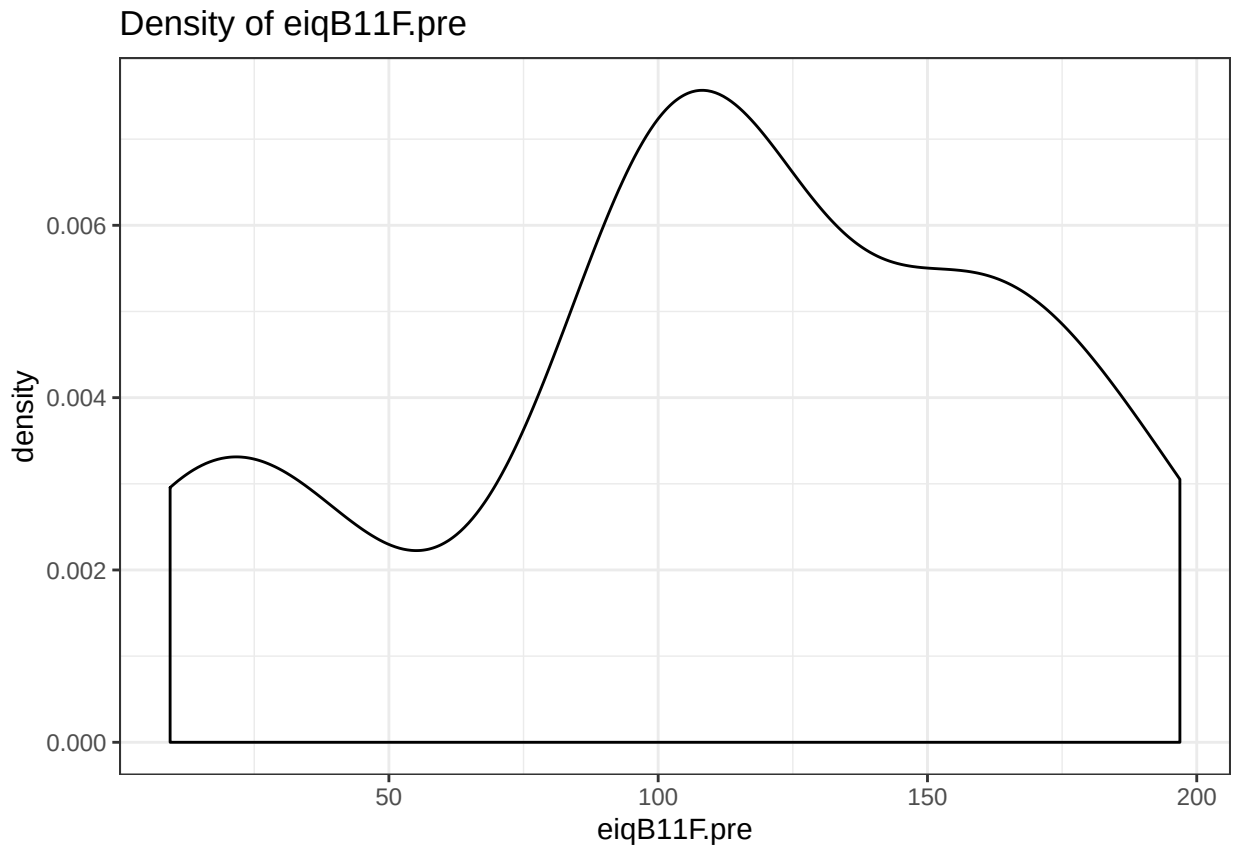
```
## [[1]]
```


Density of eiqB11F.pos
density
0.03
0.02
0.01
0.00
25
50
75
eiqB11F.pos

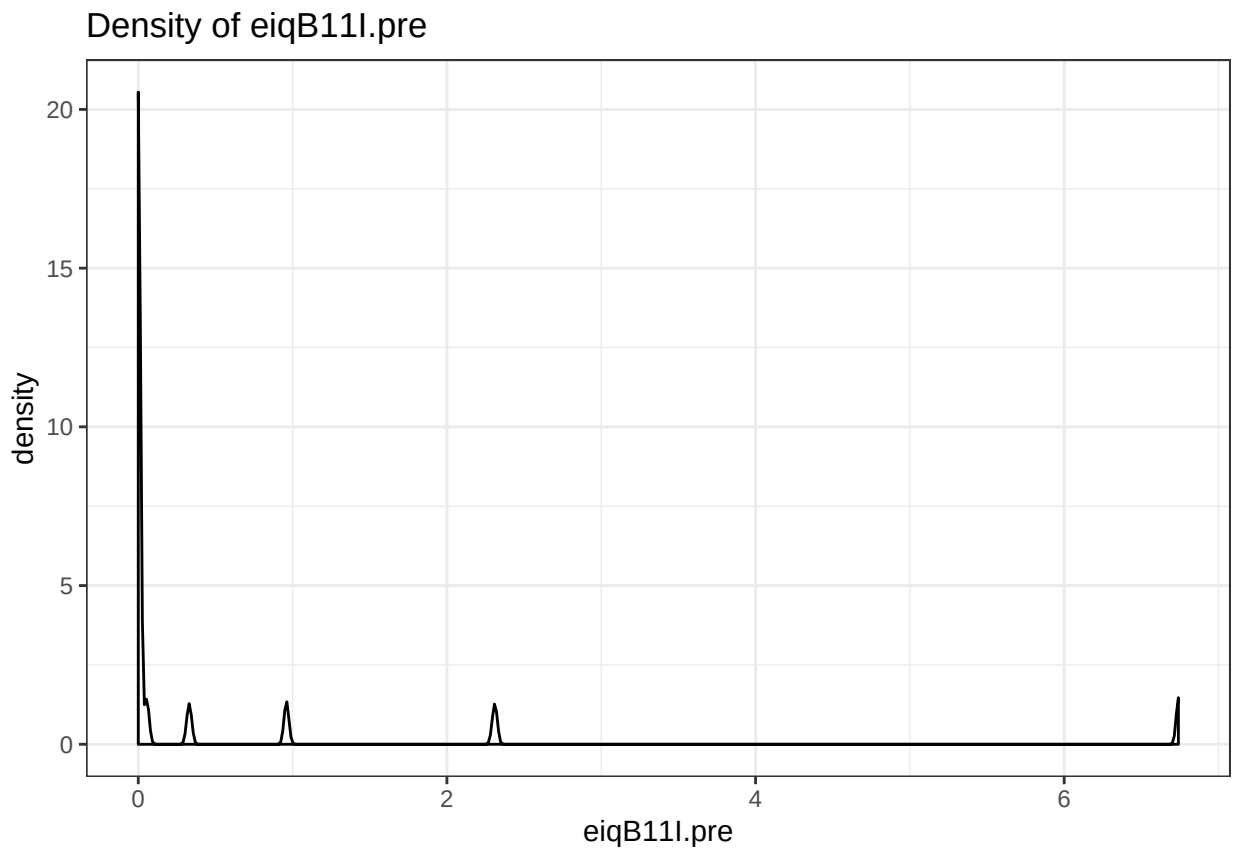
```
##
## [[2]]
```


Density of eiqB11I.pos
density
0.02
0.01
0.00
0
20
40
60
eiqB11I.pos

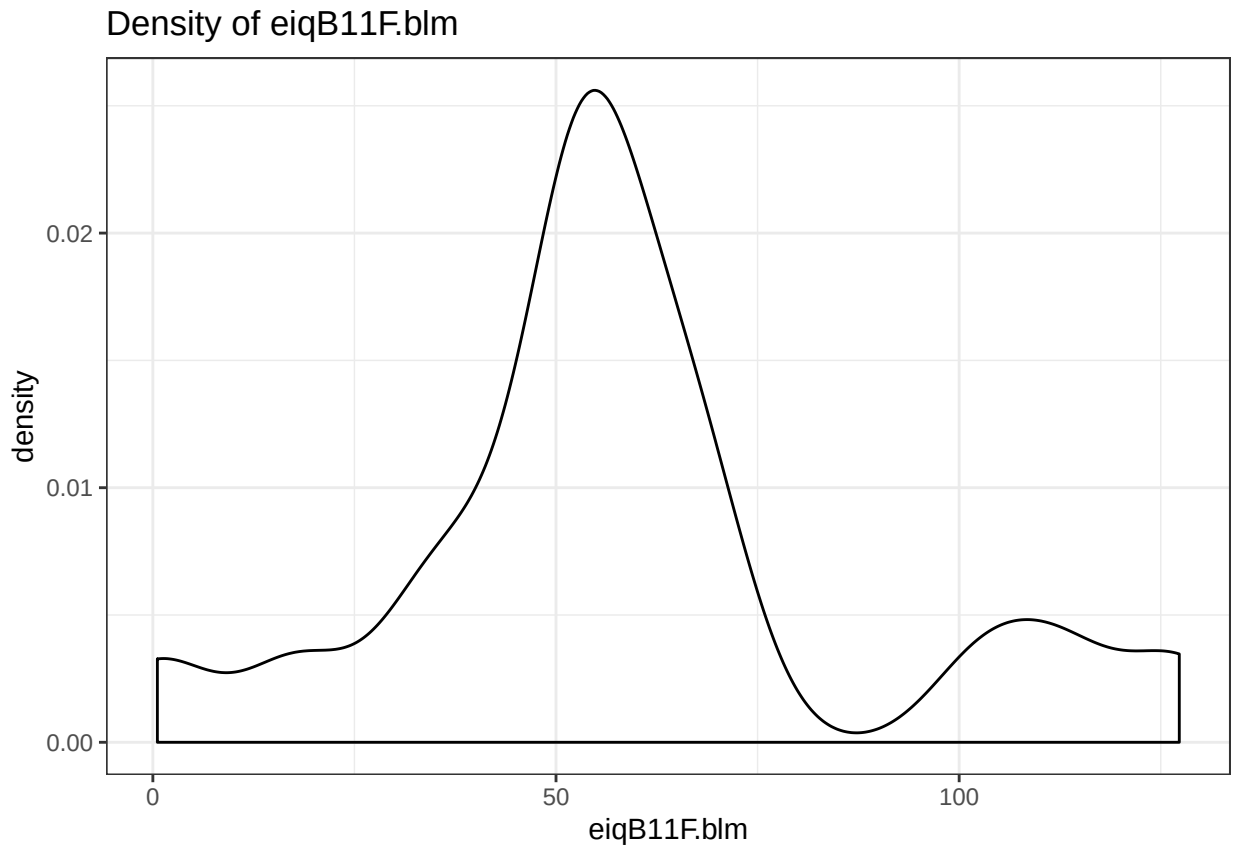
```
##
## [[3]]
```


Density of eiqB11F.pre
density
0.006
0.004
0.002
0.000
50
100
150
200
eiqB11F.pre

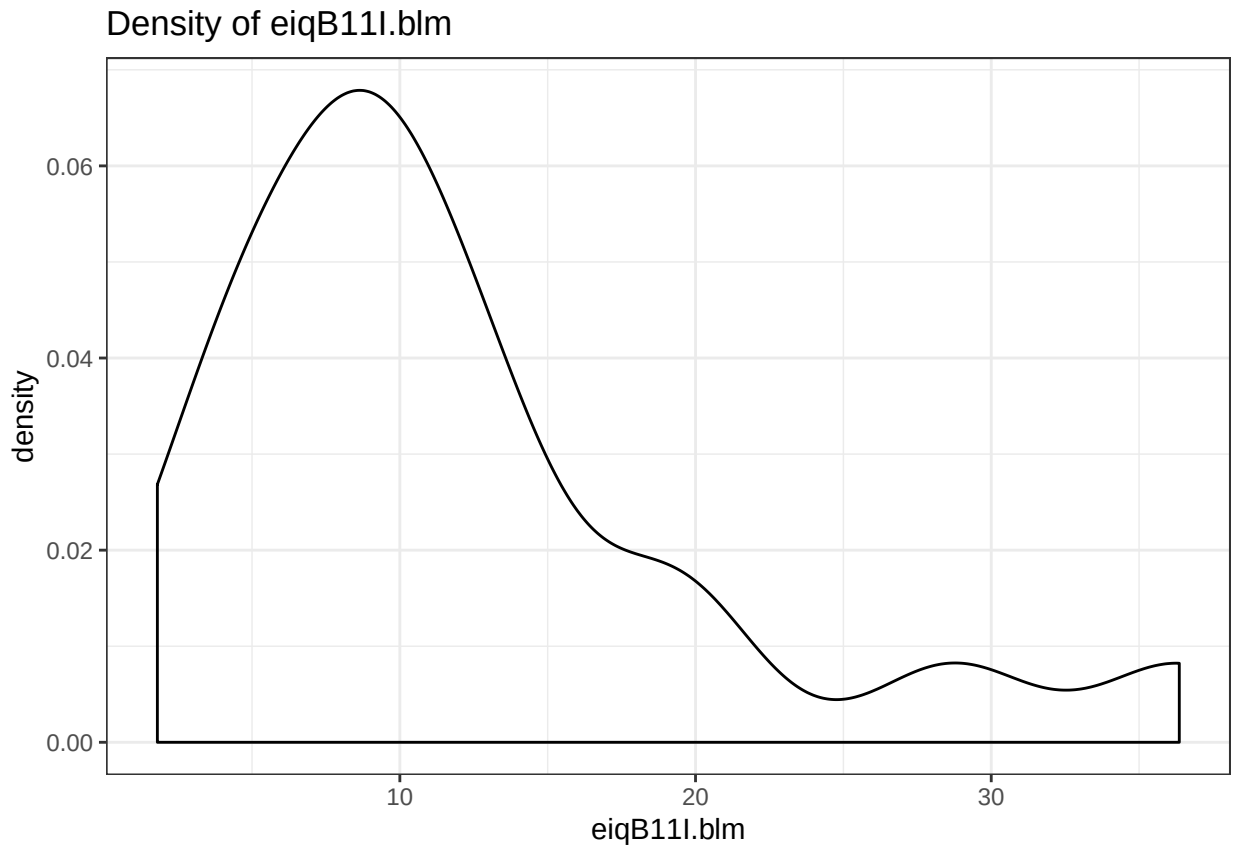
```
##
## [[4]]
```




```
##
## [[5]]
```

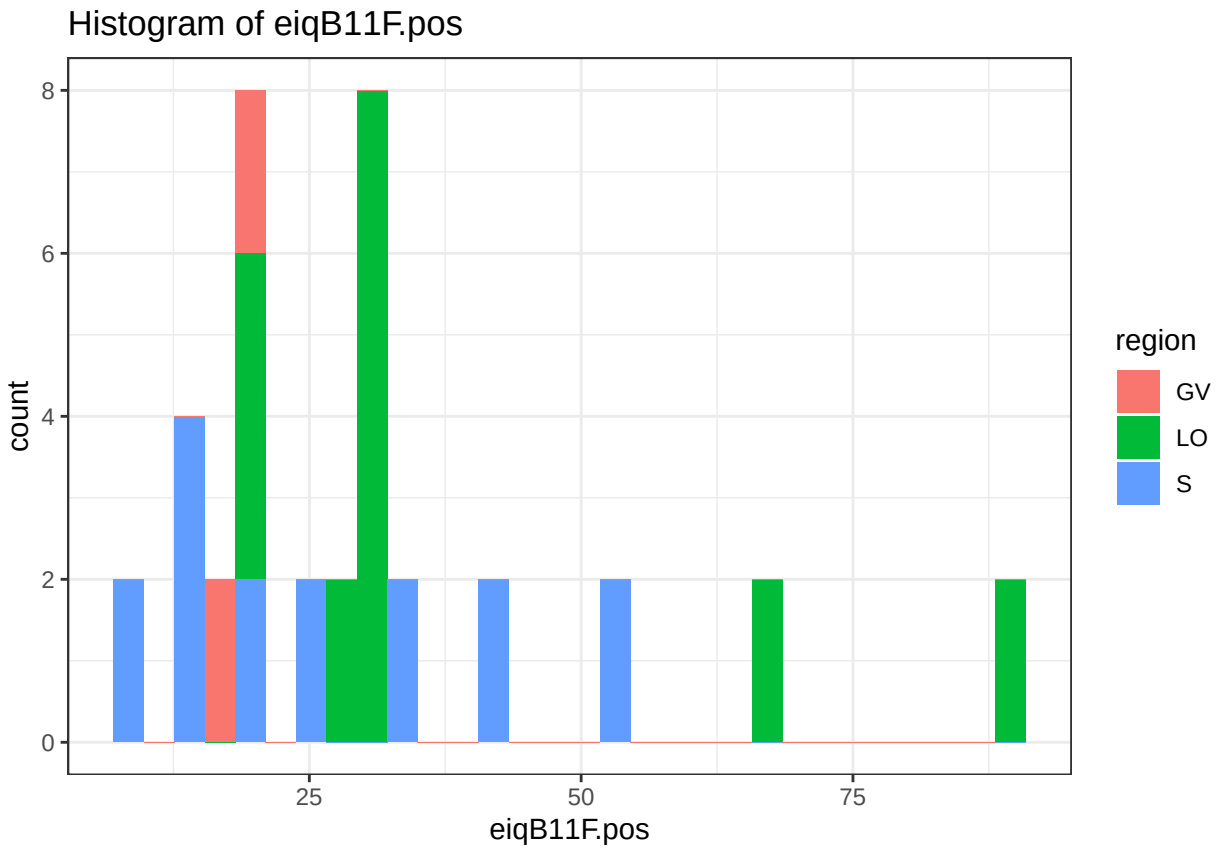

Density of eiqB11F.blm
density
0.02
0.01
0.00
0
50
100
eiqB11F.blm

```
##
## [[6]]
```



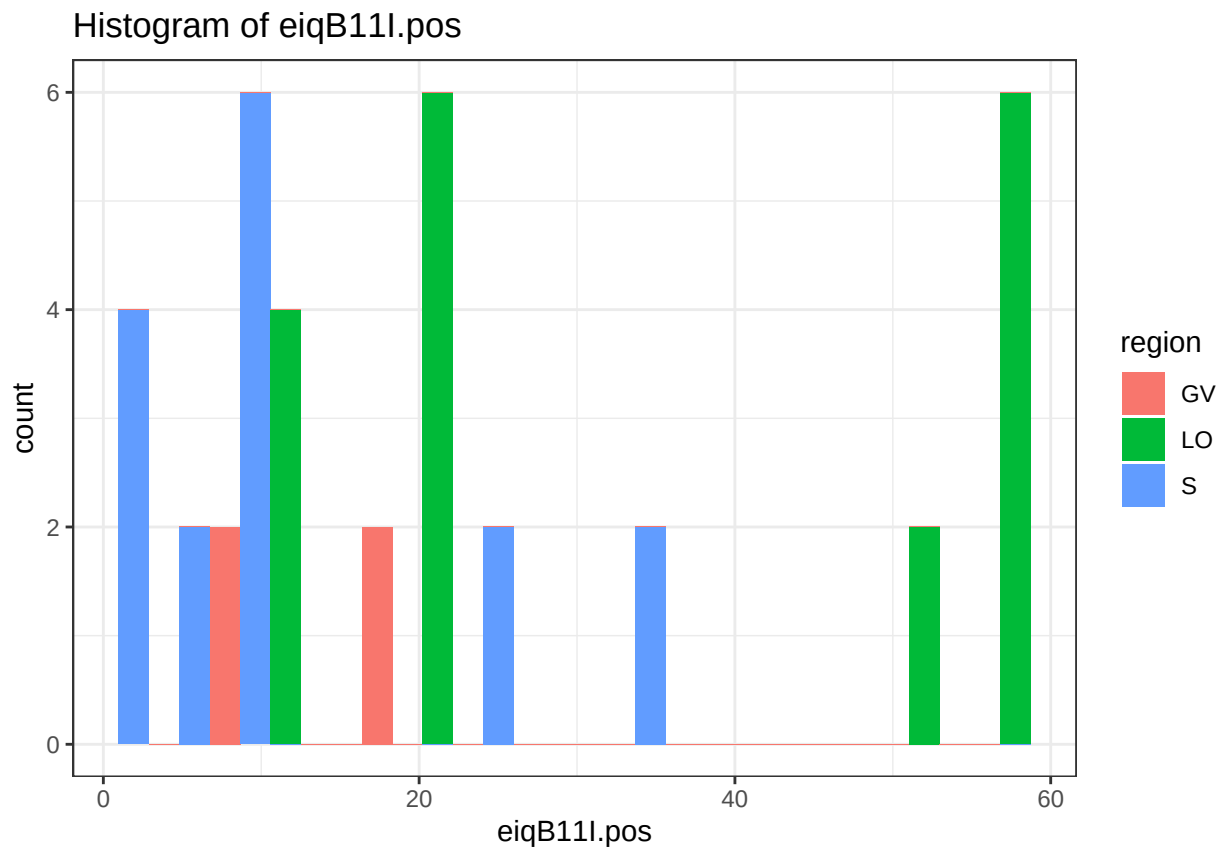

```
## [[1]]
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



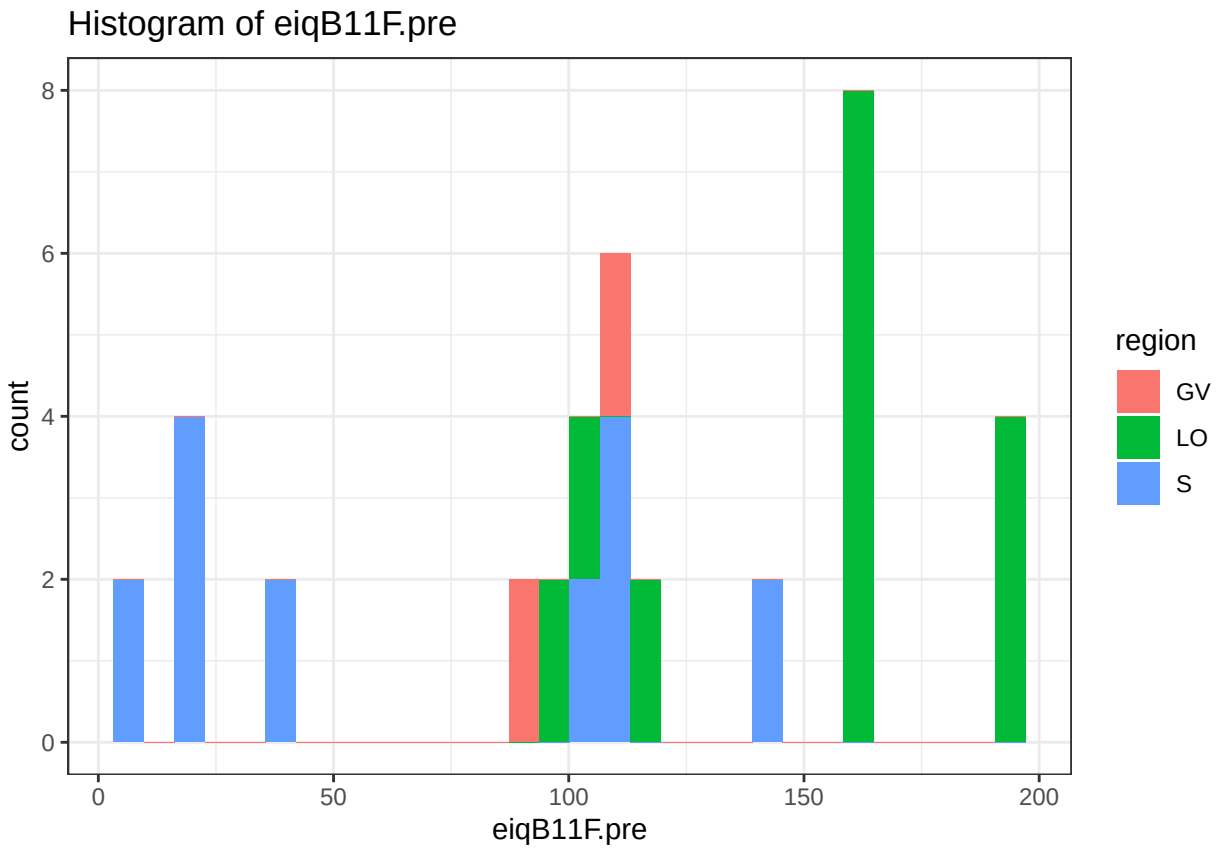

```
##
## [[2]]
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



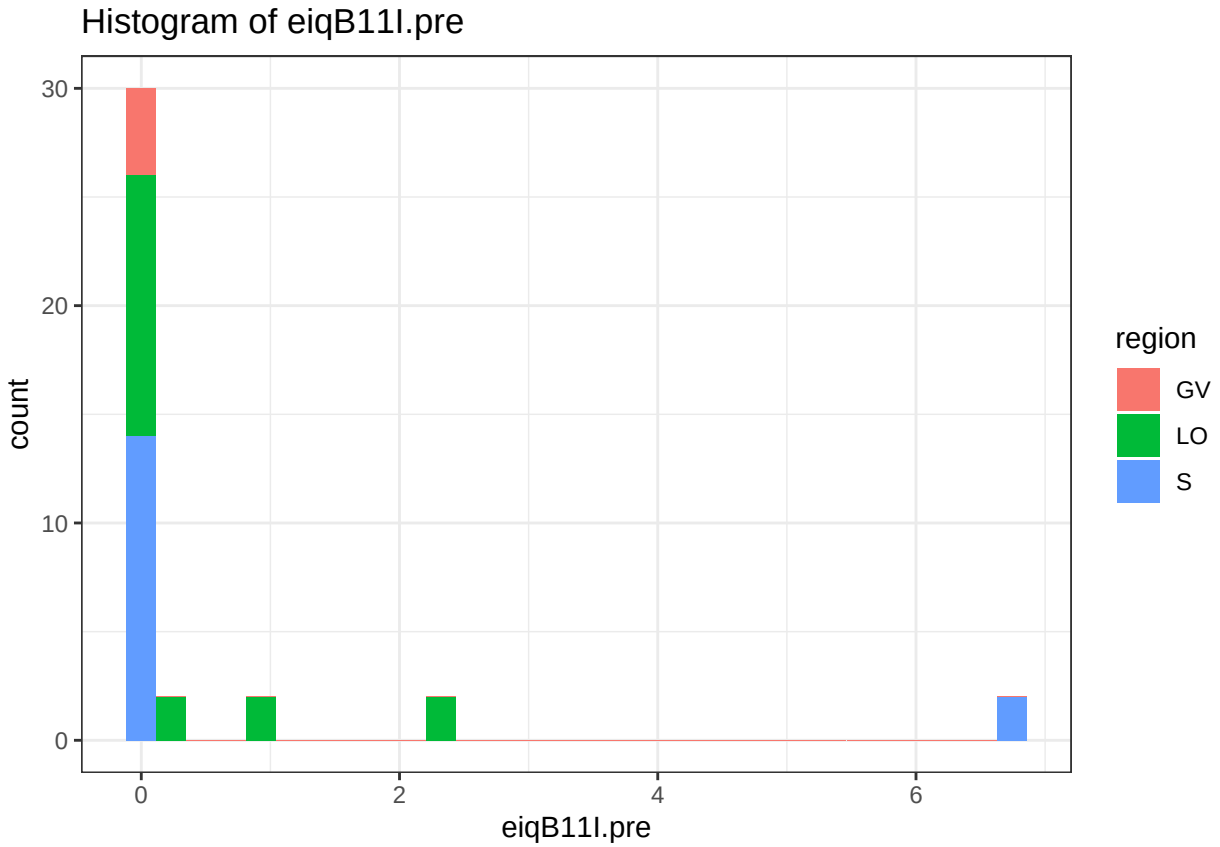

```
##
## [[3]]
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



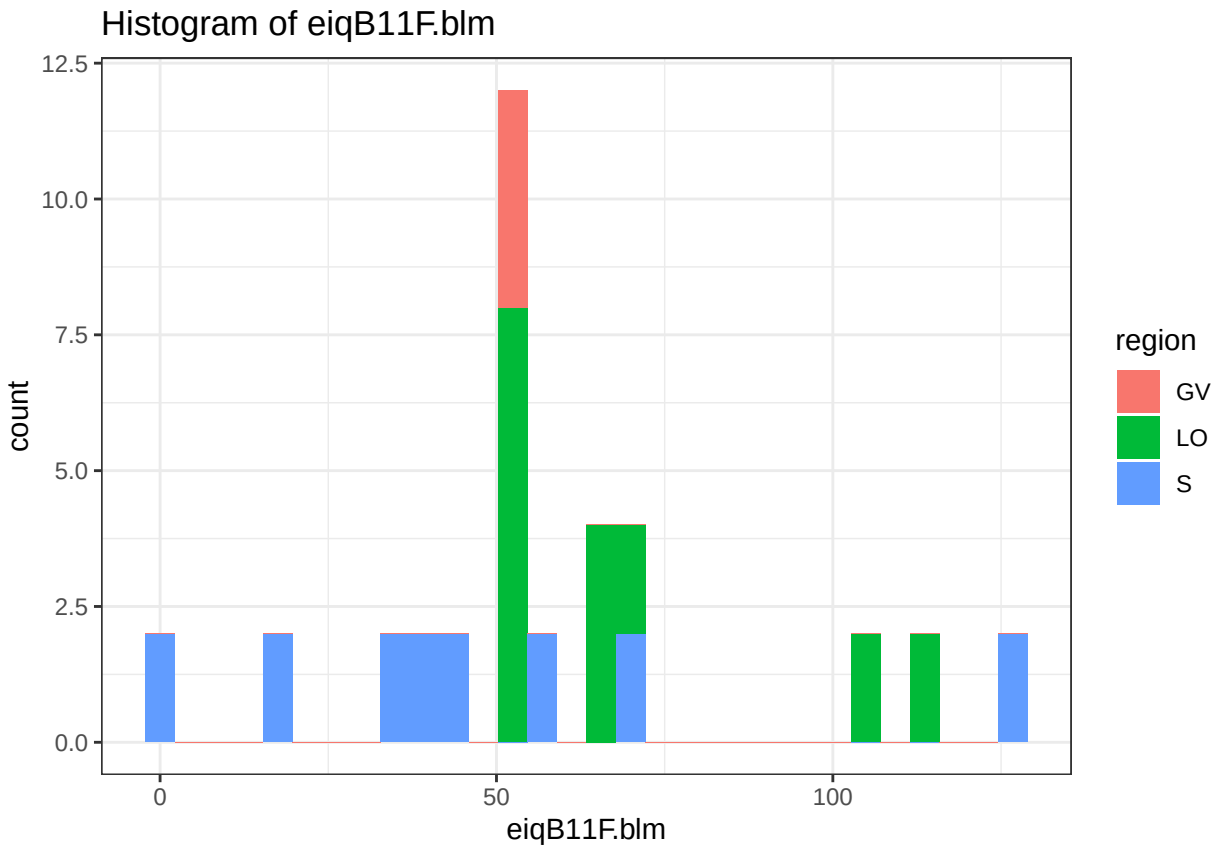

```
##
## [[4]]
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



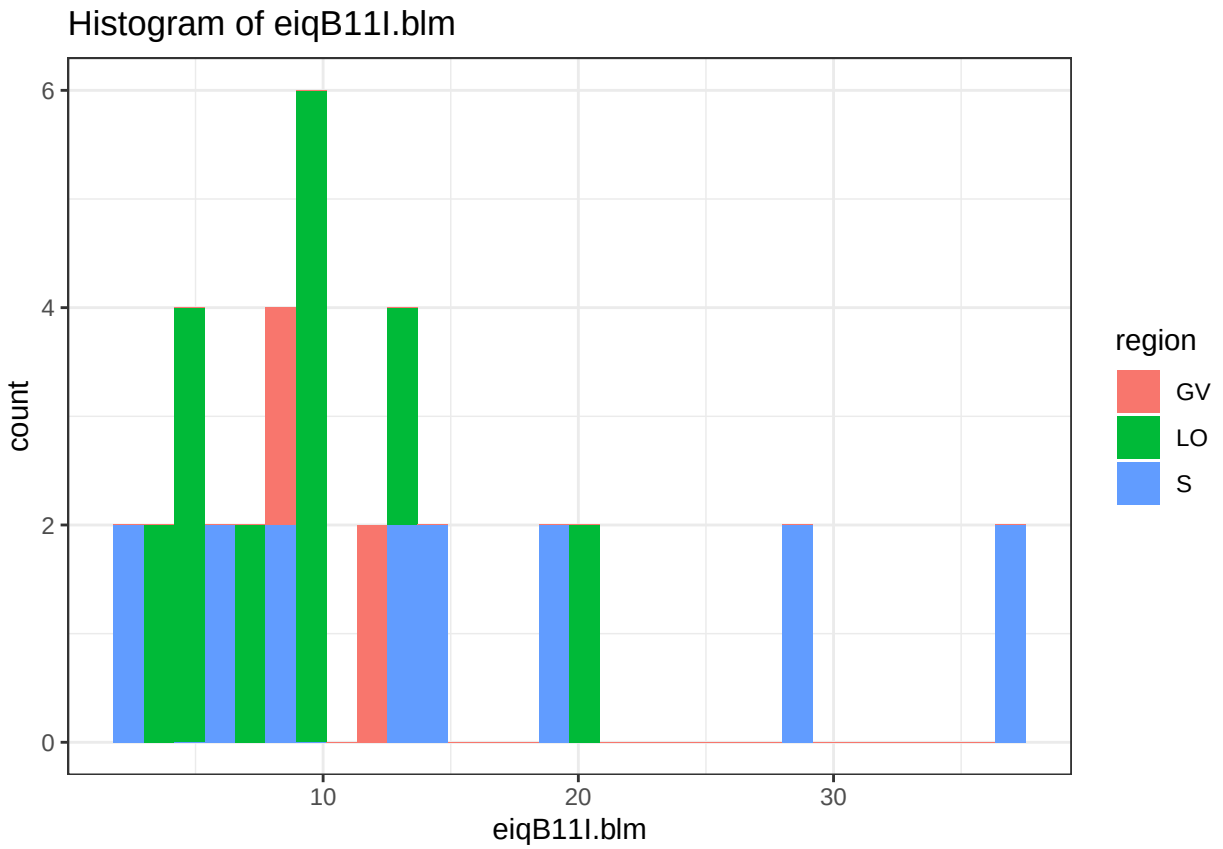

```
##
## [[5]]
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
##
## [[6]]
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Looking at kmeans for clustering of orchards to decide on decision model, using Kmeans for now to generate the outputs quickly as an EDA approach to clustering.

```
## # A tibble: 8 x 4
## # Groups:   cluster_4_pre, cluster_2_blm [6]
##   cluster_4_pre cluster_2_blm cluster_2_pos COUNT
##           <int>         <int>         <int> <int>
## 1             1             2             1     8
## 2             2             1             1     2
## 3             2             1             2     2
## 4             2             2             1    12
## 5             3             1             2     2
## 6             3             2             1     2
## 7             3             2             2     6
## 8             4             2             1     4
```

```
## # A tibble: 5 x 4
## # Groups:   cluster_2_pre, cluster_2_blm [3]
##   cluster_2_pre cluster_2_blm cluster_2_pos COUNT
##           <int>         <int>         <int> <int>
## 1             1             1             1     2
## 2             1             1             2     4
## 3             1             2             1    18
## 4             1             2             2     6
## 5             2             2             1     8
```

```
## # A tibble: 10 x 4
## # Groups:   cluster_3_pre, cluster_3_blm [7]
```

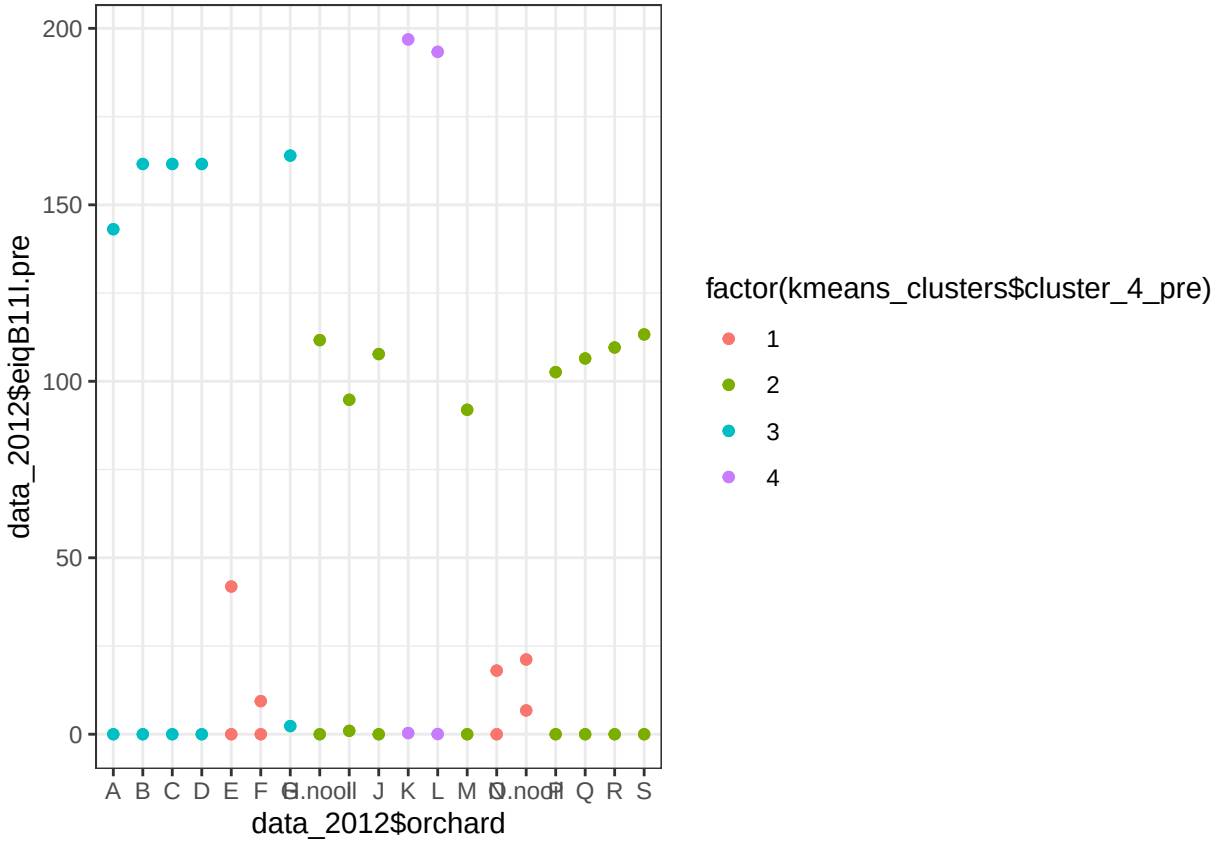
```
##    cluster_3_pre cluster_3_blm cluster_3_pos COUNT
##            <int>         <int>         <int> <int>
##  1             1             2             1     4
##  2             1             2             3     2
##  3             1             3             1     2
##  4             2             1             2     2
##  5             2             2             1     6
##  6             2             2             3     6
##  7             3             1             1     2
##  8             3             1             2     2
##  9             3             2             1    10
## 10             3             3             1     2
```

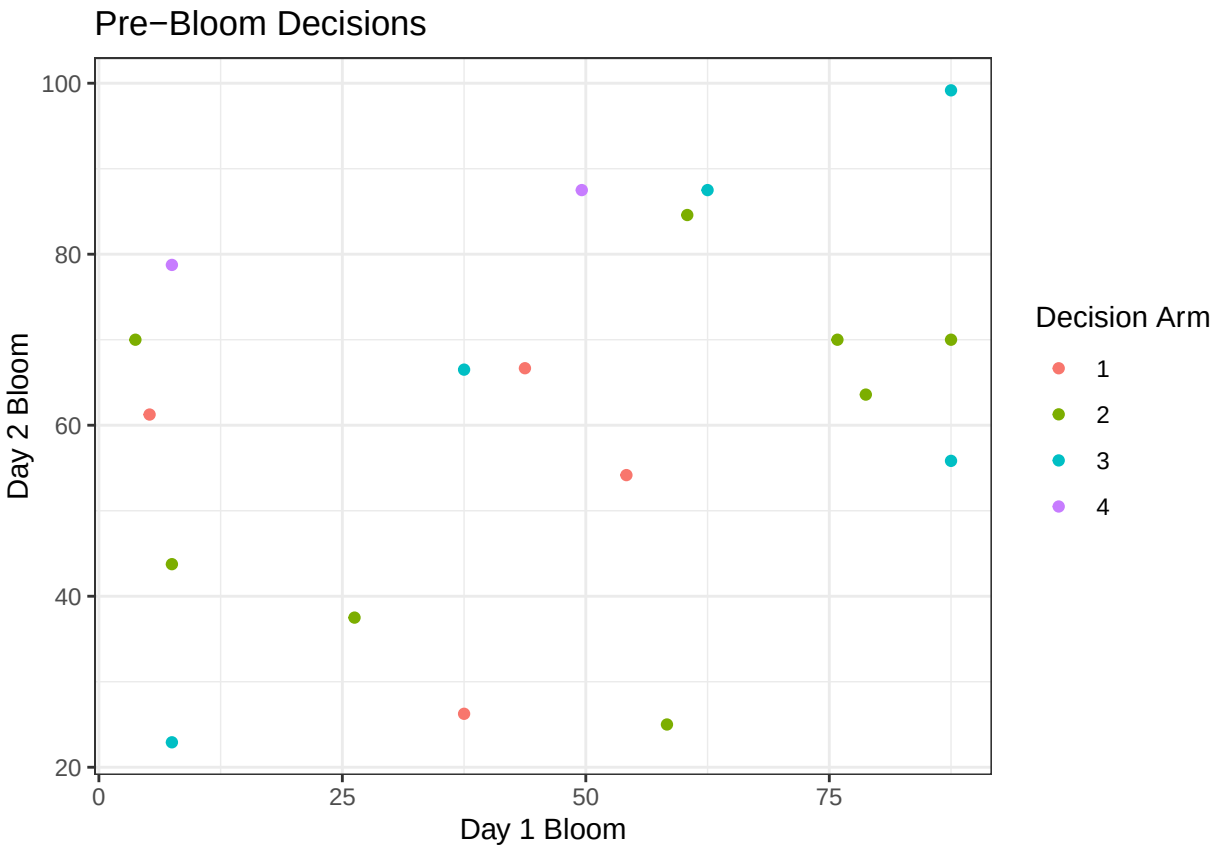
```
## # A tibble: 5 x 9
## # Groups:   cluster_2_pre, cluster_2_blm [3]
##   cluster_2_pre cluster_2_blm cluster_2_pos fung_pre insect_pre fung_blm
##           <int>         <int>         <int>    <dbl>      <dbl>    <dbl>
## 1             1             1             1    112.        0       127.
## 2             1             1             2    139.        1.16    108.
## 3             1             2             1    127.        0.15     49.6
## 4             1             2             2    162.        0        53.2
## 5             2             2             1     22.6       1.68     42.5
## # ... with 3 more variables: insect_blm <dbl>, fung_pos <dbl>,
## #   insect_pos <dbl>
```

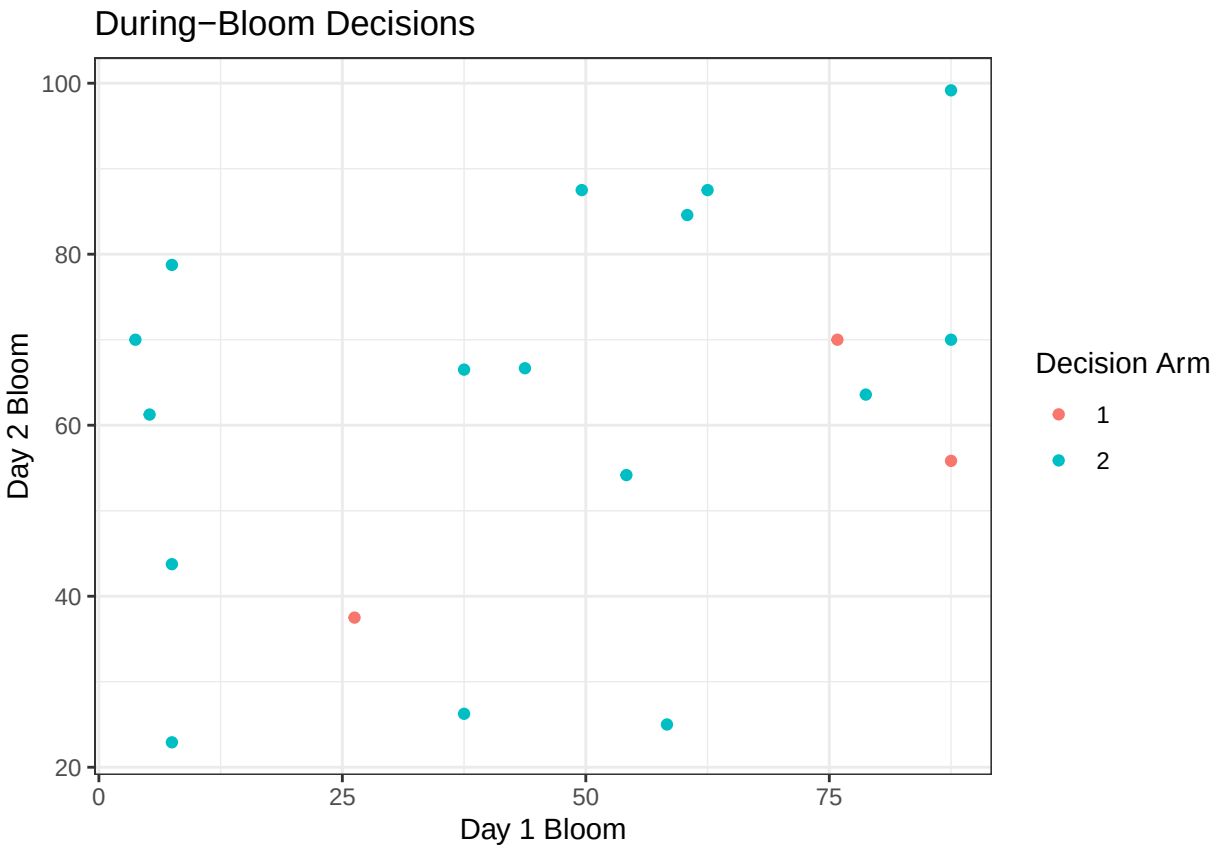
```
## # A tibble: 8 x 9
## # Groups:   cluster_4_pre, cluster_2_blm [6]
##   cluster_4_pre cluster_2_blm cluster_2_pos fung_pre insect_pre fung_blm
##           <int>         <int>         <int>    <dbl>      <dbl>    <dbl>
## 1             1             2             1     22.6       1.68     42.5
## 2             2             1             1    112.        0       127.
## 3             2             1             2    113.        0       103.
## 4             2             2             1    102.        0.16     43.6
## 5             3             1             2    164.        2.31    113.
## 6             3             2             1    143.        0        58.1
## 7             3             2             2    162.        0        53.2
## 8             4             2             1    195.        0.19     63.5
## # ... with 3 more variables: insect_blm <dbl>, fung_pos <dbl>,
## #   insect_pos <dbl>
```

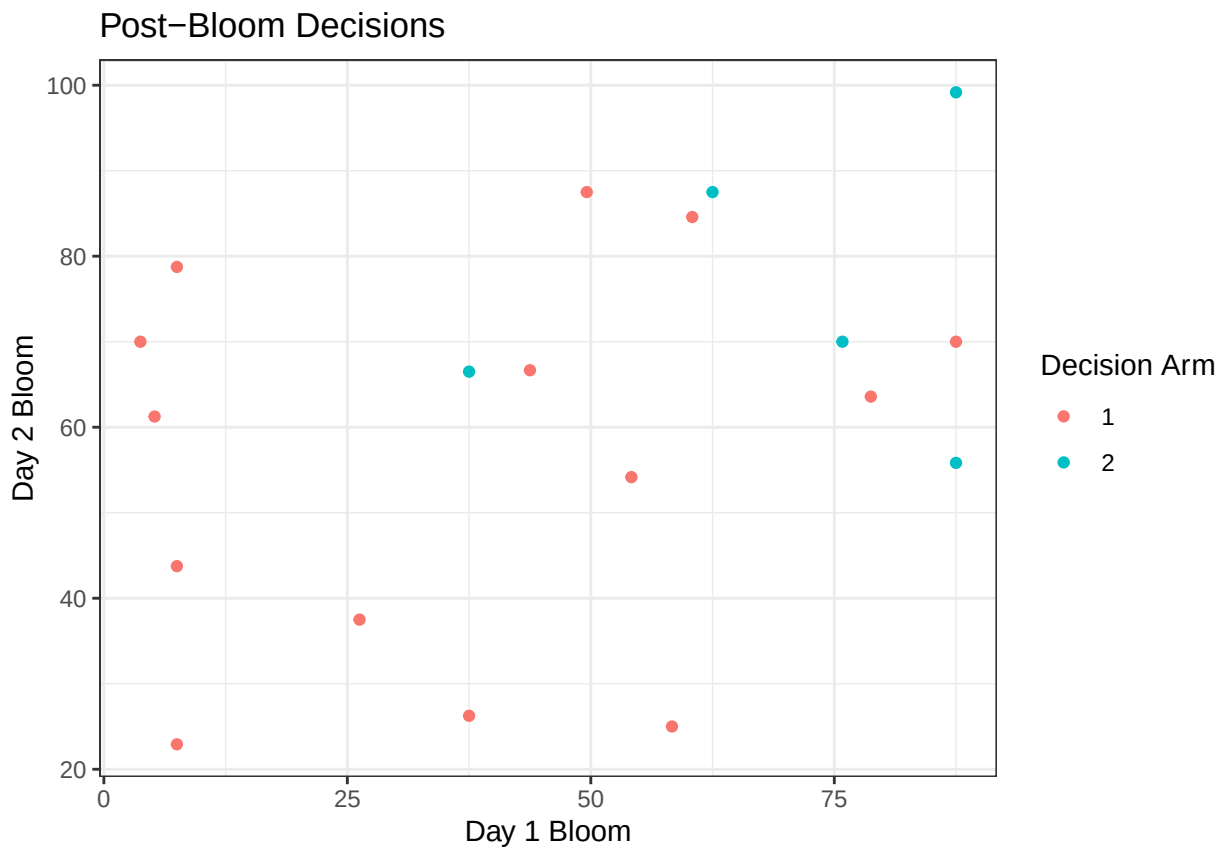
```
## # A tibble: 10 x 9
## # Groups:   cluster_3_pre, cluster_3_blm [7]
##    cluster_3_pre cluster_3_blm cluster_3_pos fung_pre insect_pre fung_blm
##            <int>         <int>         <int>    <dbl>      <dbl>    <dbl>
##  1             1             2             1    29.9        0       57.4
##  2             1             2             3     9.38       0       37.4
##  3             1             3             1    21.2        6.74    17.8
##  4             2             1             2   164.         2.31   113.
##  5             2             2             1   178.         0.13    61.7
##  6             2             2             3   162.         0       53.2
##  7             3             1             1   112.         0      127.
##  8             3             1             2   113.         0      103.
##  9             3             2             1   101.         0.19    52.2
## 10             3             3             1   108.         0        0.56
## # ... with 3 more variables: insect_blm <dbl>, fung_pos <dbl>,
```

```
## #   insect_pos <dbl>
```




Pre−Bloom Decisions
Day 2 Bloom
Day 1 Bloom
100
80
60
40
20
0
25
50
75
Decision Arm
1
2
3
4


During−Bloom Decisions
Day 2 Bloom
Day 1 Bloom
100
80
60
40
20
0
25
50
75
Decision Arm
1
2



Looking at correlations of the bee variables

```
a <- data.frame(matrix(0, nrow = 0, ncol = 0))

for(i in 1:7){
  for(j in 1:7){
  a[i,j] <- round(cor(data_2012[5+i], data_2012[5 + j]), 2)
  }
}
a
```

```
##   V1.1 V1.2 V1.3 V1.4 V1.5 V1.6 V1.7 V2.1 V2.2 V2.3 V2.4 V2.5 V2.6 V2.7
## 1 1.00 1.00 1.00 1.00 1.00 1.00 1.00 0.29 0.29 0.29 0.29 0.29 0.29 0.29
## 2 0.29 0.29 0.29 0.29 0.29 0.29 0.29 1.00 1.00 1.00 1.00 1.00 1.00 1.00
## 3 0.19 0.19 0.19 0.19 0.19 0.19 0.19 0.91 0.91 0.91 0.91 0.91 0.91 0.91
## 4 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.96 0.96 0.96 0.96 0.96 0.96 0.96
## 5 0.13 0.13 0.13 0.13 0.13 0.13 0.13 0.92 0.92 0.92 0.92 0.92 0.92 0.92
## 6 0.18 0.18 0.18 0.18 0.18 0.18 0.18 0.29 0.29 0.29 0.29 0.29 0.29 0.29
## 7 0.22 0.22 0.22 0.22 0.22 0.22 0.22 0.31 0.31 0.31 0.31 0.31 0.31 0.31
##   V3.1 V3.2 V3.3 V3.4 V3.5 V3.6 V3.7 V4.1 V4.2 V4.3 V4.4 V4.5 V4.6 V4.7
## 1 0.19 0.19 0.19 0.19 0.19 0.19 0.19 0.25 0.25 0.25 0.25 0.25 0.25 0.25
## 2 0.91 0.91 0.91 0.91 0.91 0.91 0.91 0.96 0.96 0.96 0.96 0.96 0.96 0.96
## 3 1.00 1.00 1.00 1.00 1.00 1.00 1.00 0.81 0.81 0.81 0.81 0.81 0.81 0.81
## 4 0.81 0.81 0.81 0.81 0.81 0.81 0.81 1.00 1.00 1.00 1.00 1.00 1.00 1.00
## 5 0.92 0.92 0.92 0.92 0.92 0.92 0.92 0.92 0.92 0.92 0.92 0.92 0.92 0.92
## 6 0.50 0.50 0.50 0.50 0.50 0.50 0.50 0.02 0.02 0.02 0.02 0.02 0.02 0.02
## 7 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.05 0.05 0.05 0.05 0.05 0.05 0.05
```

```
##   V5.1 V5.2 V5.3 V5.4 V5.5 V5.6 V5.7 V6.1 V6.2 V6.3 V6.4 V6.5 V6.6 V6.7
## 1 0.13 0.13 0.13 0.13 0.13 0.13 0.13 0.18 0.18 0.18 0.18 0.18 0.18 0.18
## 2 0.92 0.92 0.92 0.92 0.92 0.92 0.92 0.29 0.29 0.29 0.29 0.29 0.29 0.29
## 3 0.92 0.92 0.92 0.92 0.92 0.92 0.92 0.50 0.50 0.50 0.50 0.50 0.50 0.50
## 4 0.92 0.92 0.92 0.92 0.92 0.92 0.92 0.02 0.02 0.02 0.02 0.02 0.02 0.02
## 5 1.00 1.00 1.00 1.00 1.00 1.00 1.00 0.13 0.13 0.13 0.13 0.13 0.13 0.13
## 6 0.13 0.13 0.13 0.13 0.13 0.13 0.13 1.00 1.00 1.00 1.00 1.00 1.00 1.00
## 7 0.16 0.16 0.16 0.16 0.16 0.16 0.16 0.99 0.99 0.99 0.99 0.99 0.99 0.99
##   V7.1 V7.2 V7.3 V7.4 V7.5 V7.6 V7.7
## 1 0.22 0.22 0.22 0.22 0.22 0.22 0.22
## 2 0.31 0.31 0.31 0.31 0.31 0.31 0.31
## 3 0.53 0.53 0.53 0.53 0.53 0.53 0.53
## 4 0.05 0.05 0.05 0.05 0.05 0.05 0.05
## 5 0.16 0.16 0.16 0.16 0.16 0.16 0.16
## 6 0.99 0.99 0.99 0.99 0.99 0.99 0.99
## 7 1.00 1.00 1.00 1.00 1.00 1.00 1.00
```

```
for(i in 1:7){
 names(a)[i] <- colnames(data_2012[5 +i])
}
```

Objective 1b/c/1d further models and analysis

After speaking with botanist, no predefined appraoch will be taken. All the models will be created using a data driven approach, so going to look at some different clustering methods fitted using the 19 orchards as the data and then compare the different models and clustering methods. From the Kmeans analysis above, the different seed returning differing clusterings is not ideal, so some different approaches will be looked at. It is also clear that more factors need to be taken into account in the clustering process rather than just the insecticide and pesticide levels.

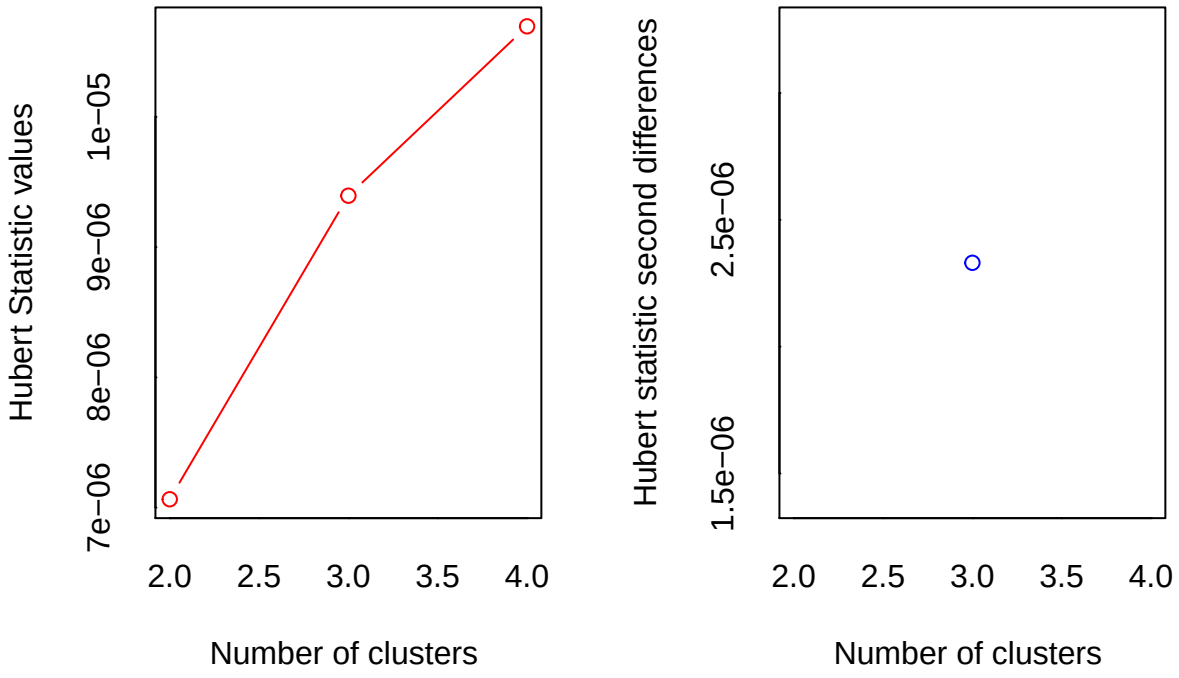
Version 2

Taking the above method a little further to look at different distance metric and add in X2000nat as another variable, this is the clustering options that were reporting for the naive kmeans method. And is the EDA clustering which is reported in the final analysis report. All the commented out clustering code was written to try and determine future methodology but ultimately was not used. It has been kept in as it shows the thought process behind looking and testing other methods but ultimately didn't work in a good way.

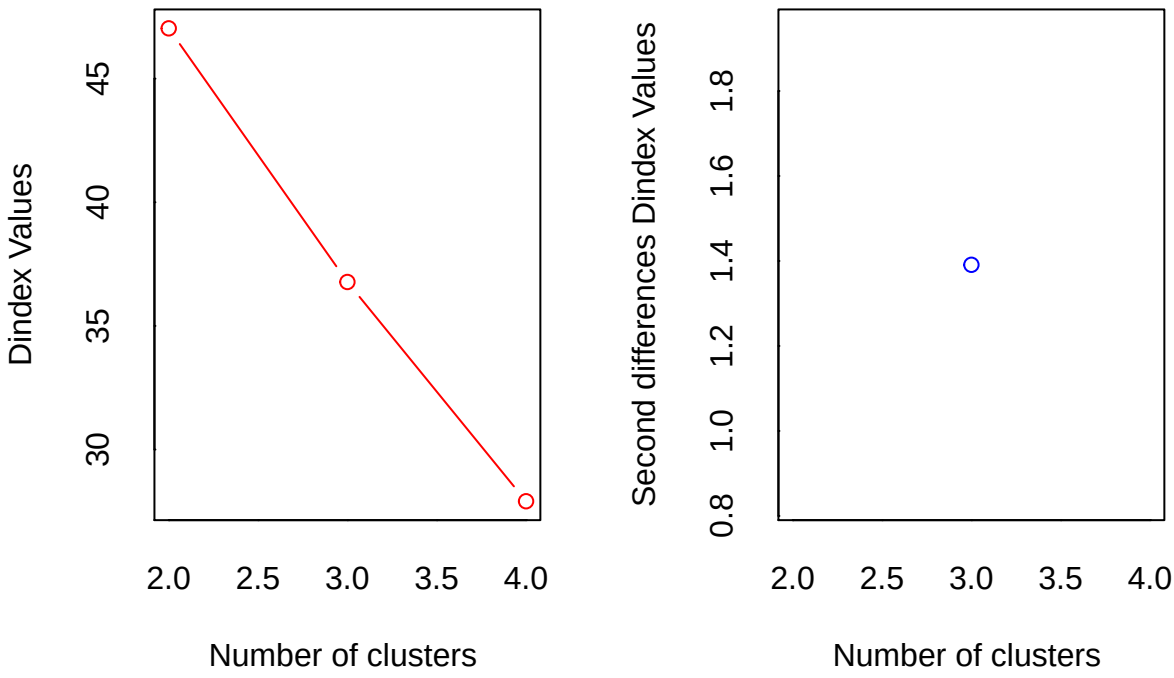
```
cluster_2_data <- data_2012 %>%
  select(ends_with(".pre"),ends_with(".blm"), ends_with(".pos"), X2000nat)
#Shows no difference here although from previous work ward method was best
# matrix of methods to compare
m <- c( "average", "single", "complete", "ward")
names(m) <- c( "average", "single", "complete", "ward")
distances <- c("euclidean", "maximum", "manhattan", "canberra",
               "binary", "minkowski")
names(distances) <- c("euclidean", "maximum", "manhattan",
                      "canberra", "binary", "minkowski")
clust_comps <- matrix(nrow = length(distances), ncol = length(m),
                      dimnames = list(distances,m))
# function to compute coefficient
ac <- function(distance, linkage) {
  dista <- dist(cluster_2_data , method = distance)
  #Agglomerative Nesting form of Hierarchical Clustering
```

```
  agnes(dista, method = linkage)$ac
}
for(i in 1:length(distances)) {
  for(j in 1:length(m)) {
    clust_comps[i,j] <- ac(distances[i], m[j])
  }
}


#Clusters of whole data combined - 7
fviz_nbclust(NbClust(cluster_2_data, distance="euclidean",
                                 min.nc=2, max.nc=4, method="ward.D2", index="all"))
```

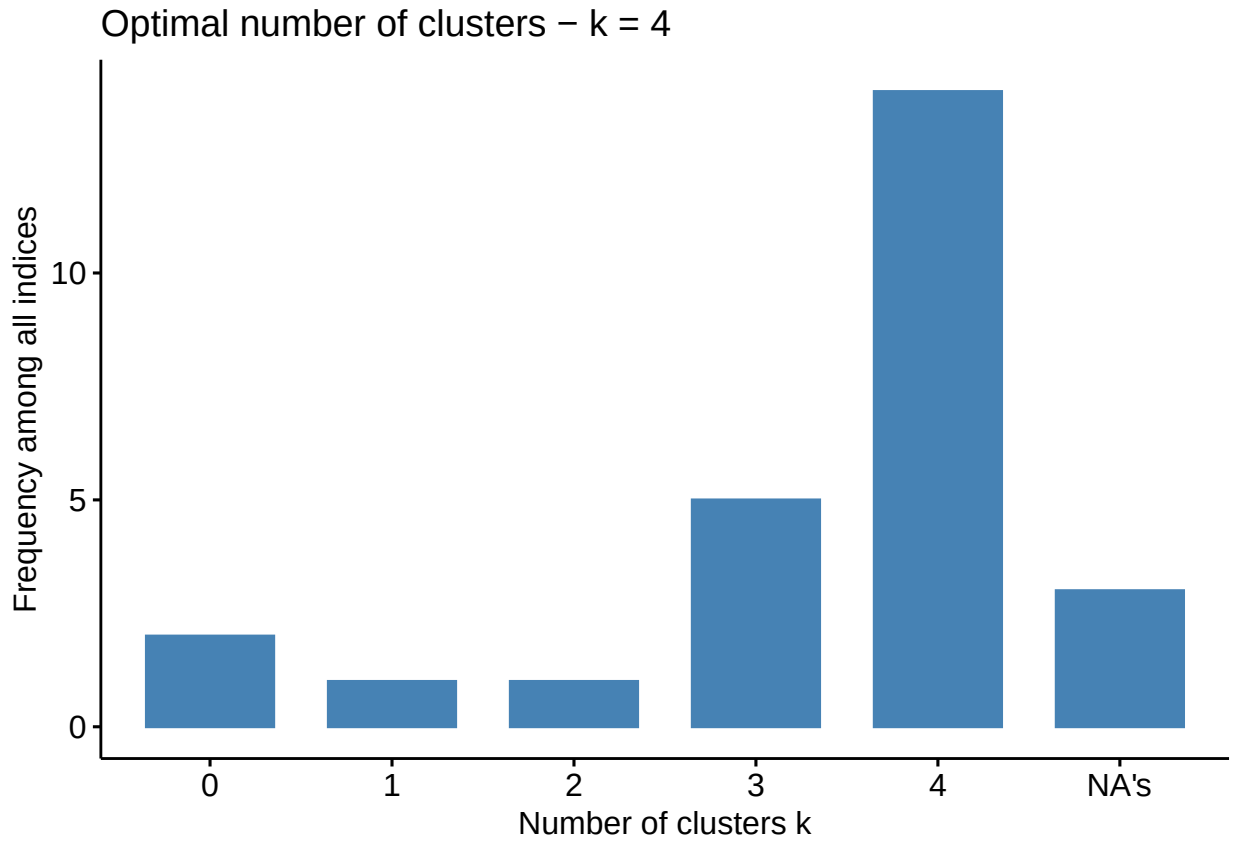



```
## *** : The Hubert index is a graphical method of determining the number of clusters.
##                 In the plot of Hubert index, we seek a significant knee that corresponds to a
##                 significant increase of the value of the measure i.e the significant peak in Hubert
##                 index second differences plot.
##
```

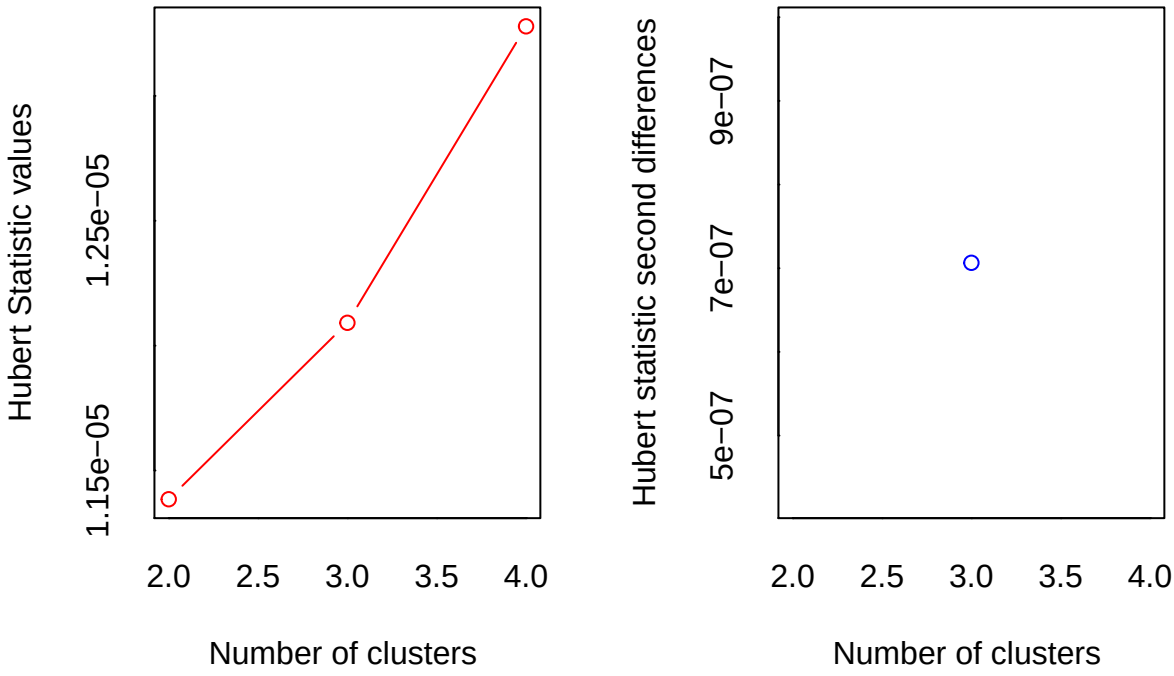



```
## *** : The D index is a graphical method of determining the number of clusters.
##                 In the plot of D index, we seek a significant knee (the significant peak in Dindex
##                 second differences plot) that corresponds to a significant increase of the value of
##                 the measure.
##
## *******************************************************************
## * Among all indices:
## * 1 proposed 2 as the best number of clusters
## * 5 proposed 3 as the best number of clusters
## * 14 proposed 4 as the best number of clusters
##
##                     ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  4
##
##
## *******************************************************************
## Among all indices:
## ==================
## * 2 proposed  0 as the best number of clusters
## * 1 proposed  1 as the best number of clusters
## * 1 proposed  2 as the best number of clusters
## * 5 proposed  3 as the best number of clusters
## * 14 proposed  4 as the best number of clusters
## * 3 proposed  NA's as the best number of clusters
##
```

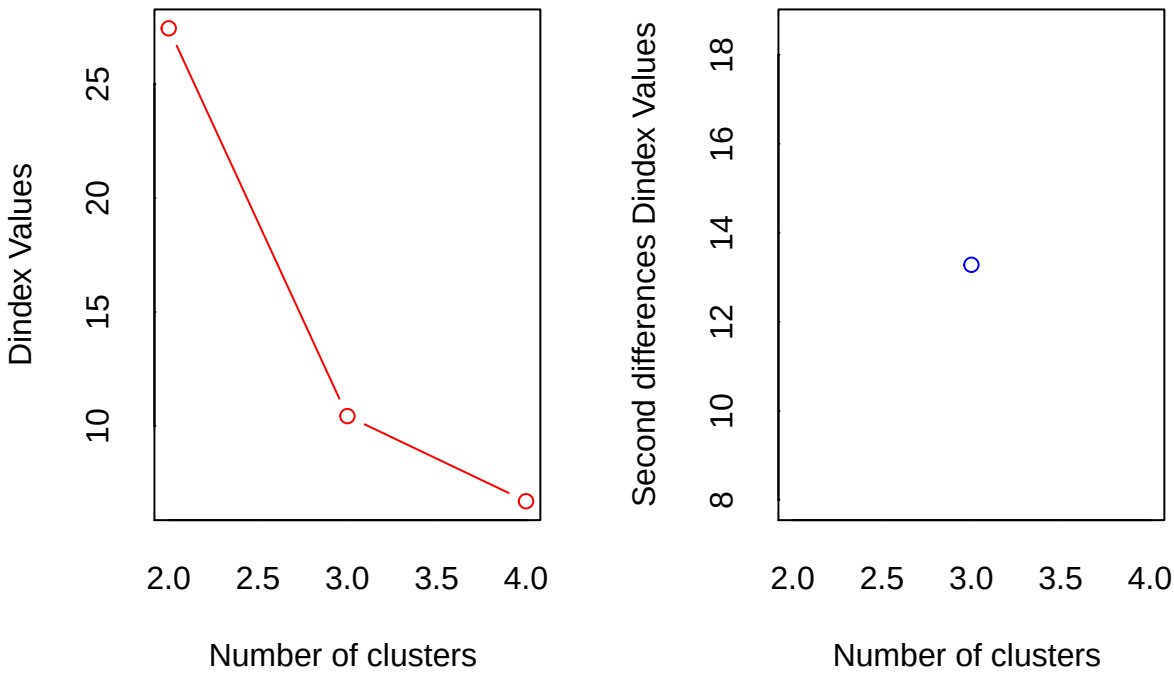
```
## Conclusion
## ========================
## * According to the majority rule, the best number of clusters is  4 .
```




```
#pre ~ max of 4 clusters at each time point in method, but  in general if you allow
#More as this argument then more returned a better  optimal cluster number
fviz_nbclust(NbClust(cluster_2_data %>% select(ends_with(".pre"), X2000nat), distance="euclidean",
                                 min.nc=2, max.nc=4, method="ward.D2", index="all"))
```

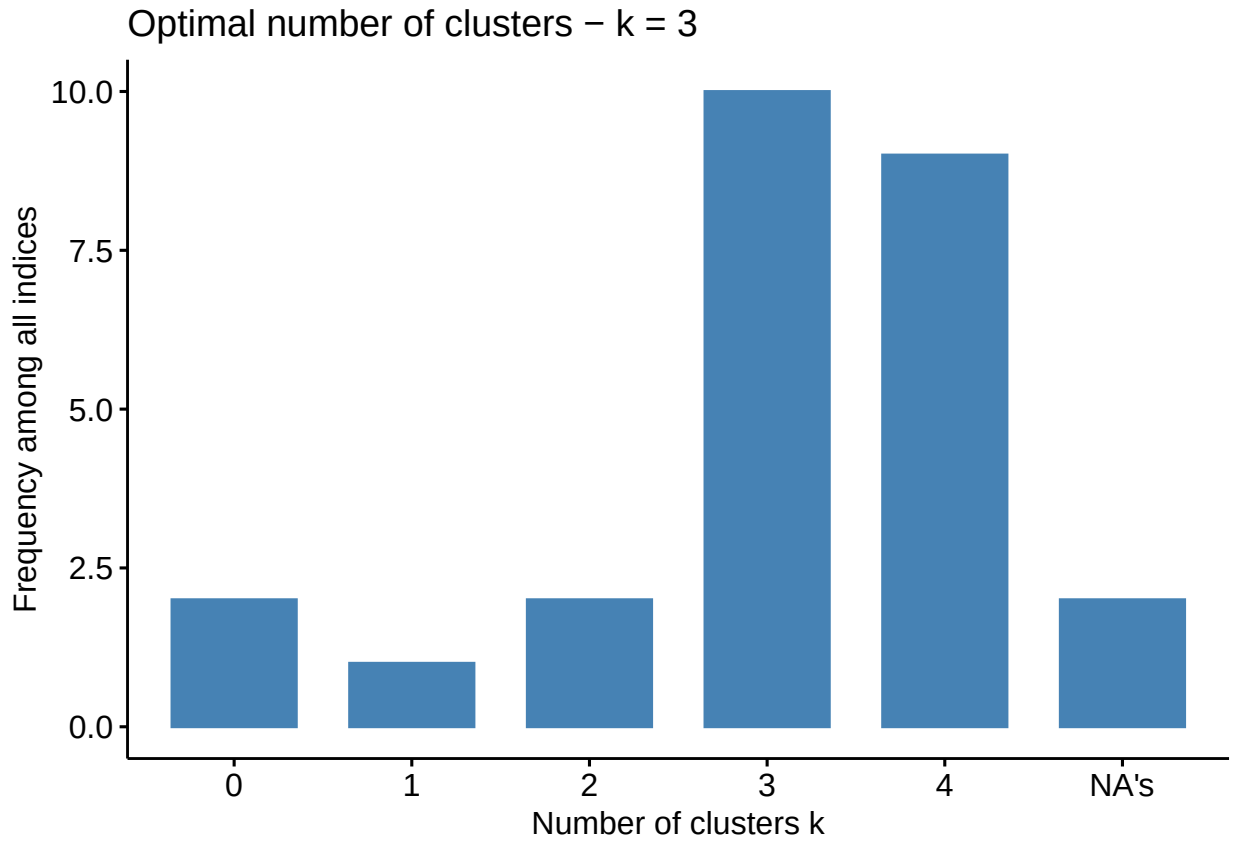



```
## *** : The Hubert index is a graphical method of determining the number of clusters.
##                 In the plot of Hubert index, we seek a significant knee that corresponds to a
##                 significant increase of the value of the measure i.e the significant peak in Hubert
##                 index second differences plot.
##
```

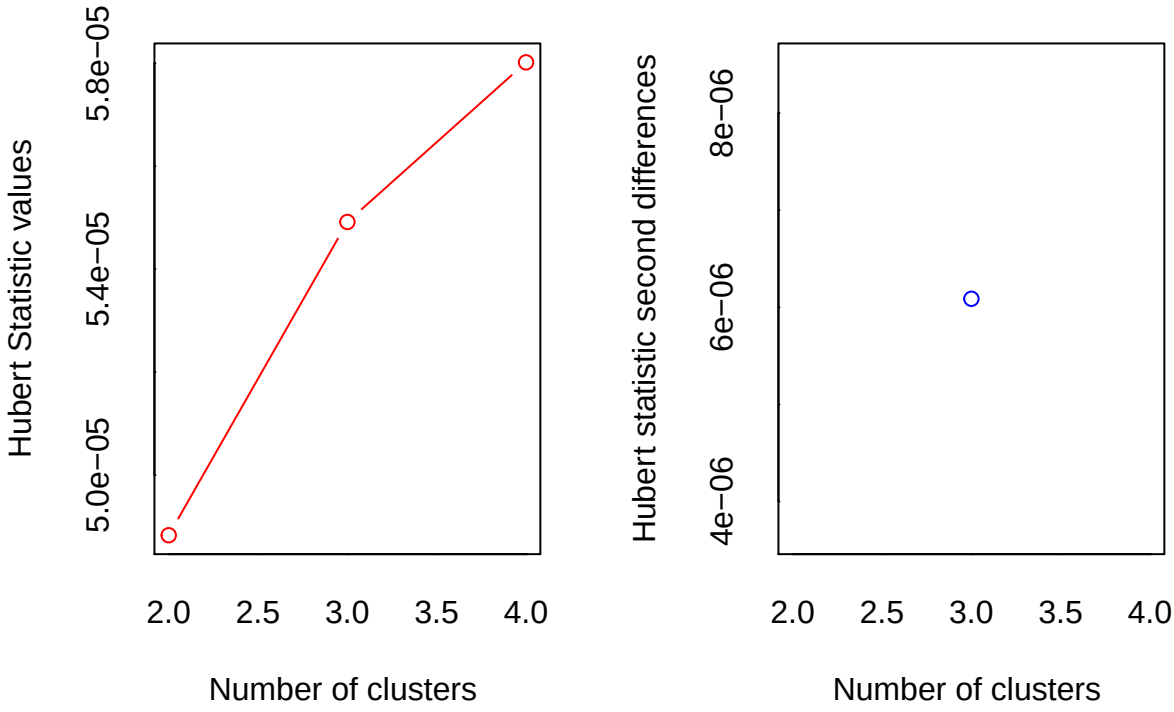



```
## *** : The D index is a graphical method of determining the number of clusters.
##                 In the plot of D index, we seek a significant knee (the significant peak in Dindex
##                 second differences plot) that corresponds to a significant increase of the value of
##                 the measure.
##
## *******************************************************************
## * Among all indices:
## * 2 proposed 2 as the best number of clusters
## * 10 proposed 3 as the best number of clusters
## * 9 proposed 4 as the best number of clusters
##
##                     ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  3
##
##
## *******************************************************************
## Among all indices:
## ==================
## * 2 proposed  0 as the best number of clusters
## * 1 proposed  1 as the best number of clusters
## * 2 proposed  2 as the best number of clusters
## * 10 proposed  3 as the best number of clusters
## * 9 proposed  4 as the best number of clusters
## * 2 proposed  NA's as the best number of clusters
##
```

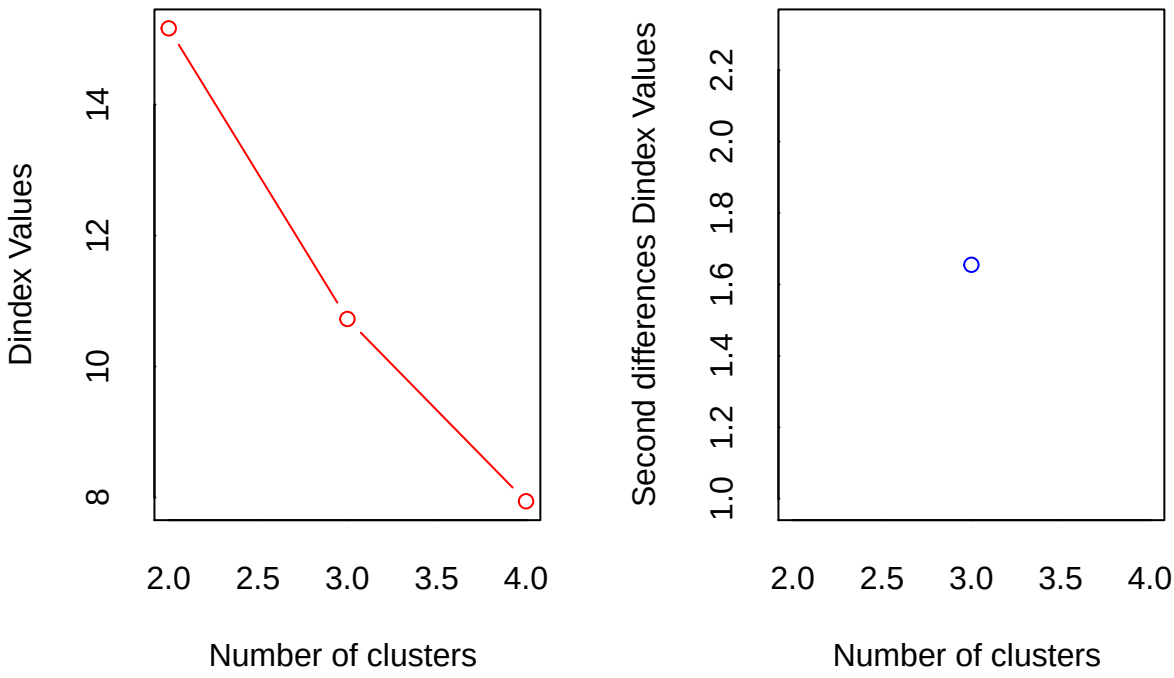
```
## Conclusion
## ========================
## * According to the majority rule, the best number of clusters is  3 .
```




```
#during
fviz_nbclust(NbClust(cluster_2_data %>% select(ends_with(".blm"), X2000nat), distance="euclidean",
                                  min.nc=2, max.nc=4, method="ward.D2", index="all"))
```

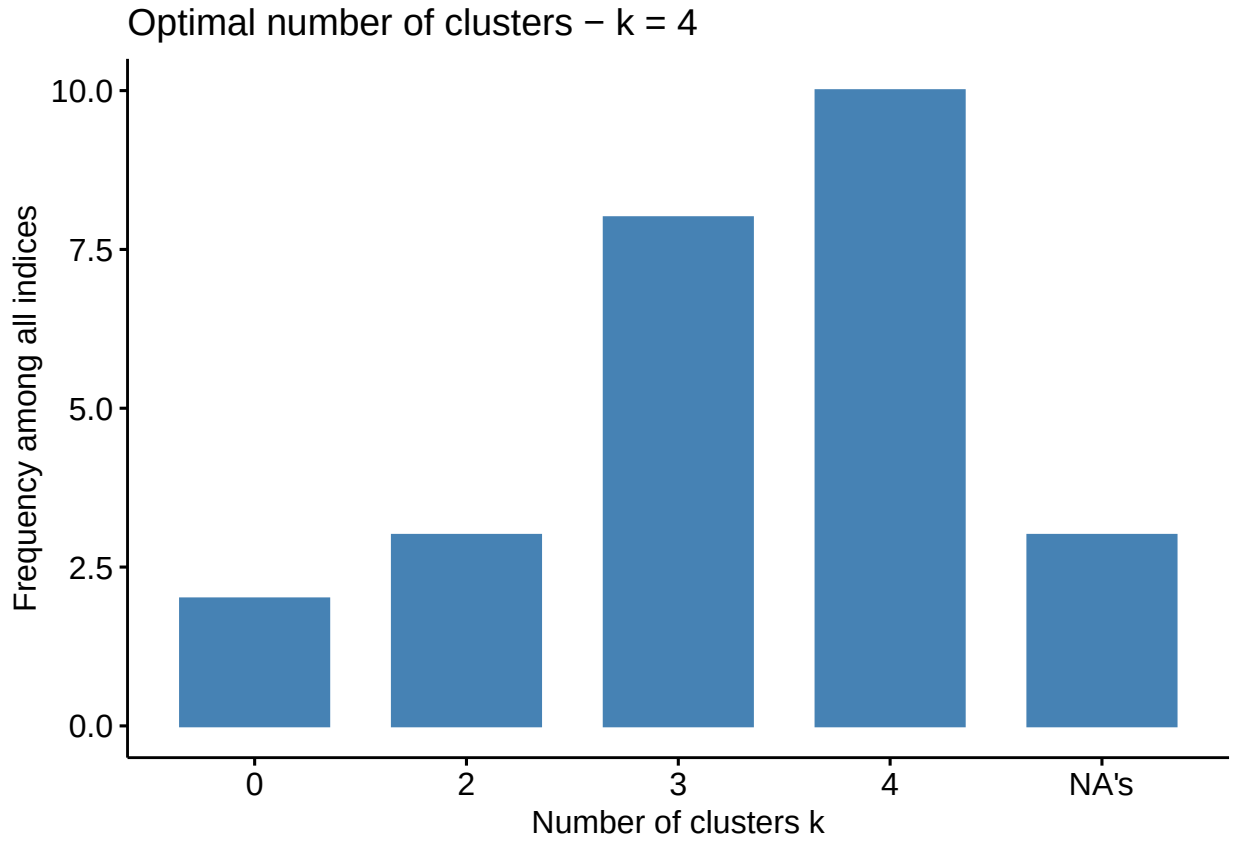



```
## *** : The Hubert index is a graphical method of determining the number of clusters.
##                 In the plot of Hubert index, we seek a significant knee that corresponds to a
##                 significant increase of the value of the measure i.e the significant peak in Hubert
##                 index second differences plot.
##
```

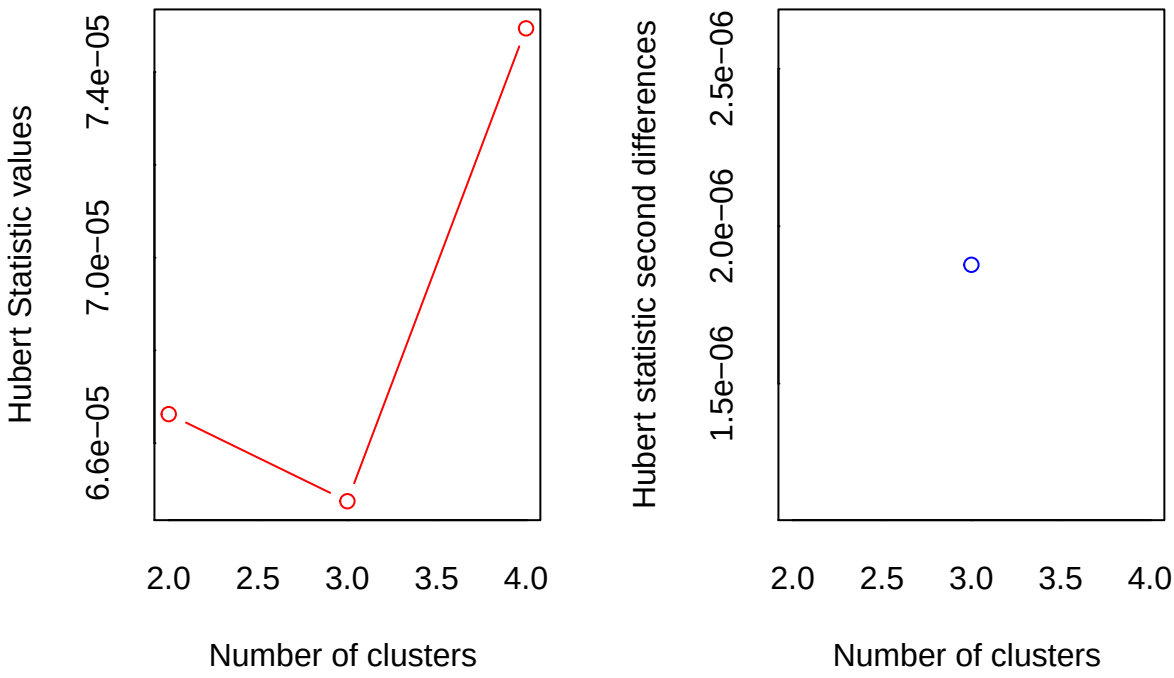



```
## *** : The D index is a graphical method of determining the number of clusters.
##                 In the plot of D index, we seek a significant knee (the significant peak in Dindex
##                 second differences plot) that corresponds to a significant increase of the value of
##                 the measure.
##
## *******************************************************************
## * Among all indices:
## * 3 proposed 2 as the best number of clusters
## * 8 proposed 3 as the best number of clusters
## * 10 proposed 4 as the best number of clusters
##
##                     ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  4
##
##
## *******************************************************************
## Among all indices:
## ==================
## * 2 proposed  0 as the best number of clusters
## * 3 proposed  2 as the best number of clusters
## * 8 proposed  3 as the best number of clusters
## * 10 proposed  4 as the best number of clusters
## * 3 proposed  NA's as the best number of clusters
##
## Conclusion
```

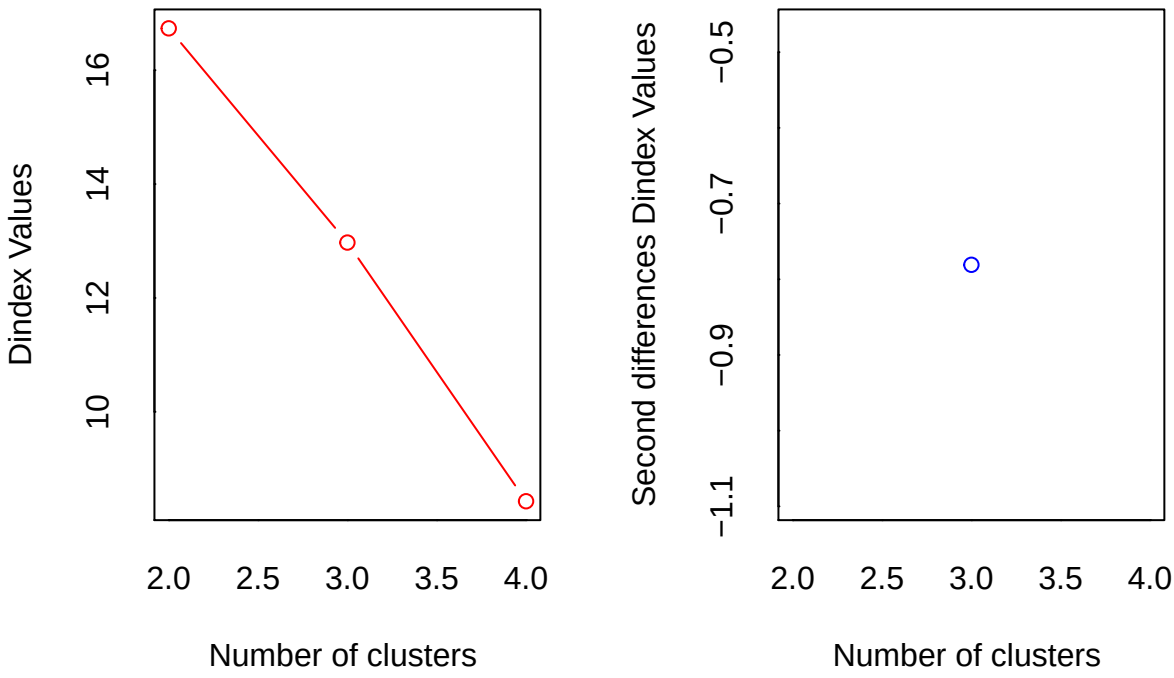
```
## =========================
## * According to the majority rule, the best number of clusters is  4 .
```




```
#post
fviz_nbclust(NbClust(cluster_2_data %>% select(ends_with(".pos"), X2000nat), distance="euclidean",
                                  min.nc=2, max.nc=4, method="ward.D2", index="all"))
```

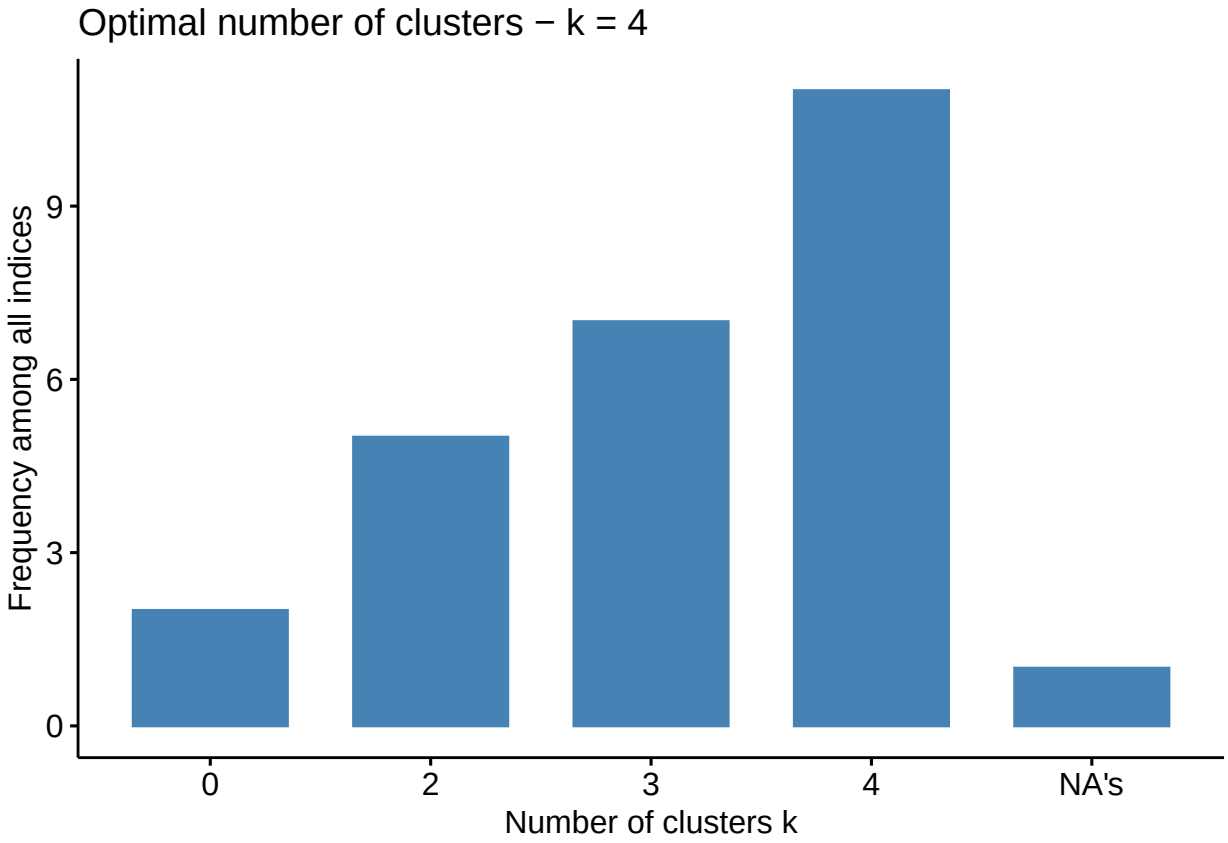



```
## *** : The Hubert index is a graphical method of determining the number of clusters.
##                 In the plot of Hubert index, we seek a significant knee that corresponds to a
##                 significant increase of the value of the measure i.e the significant peak in Hubert
##                 index second differences plot.
##
```




```
## *** : The D index is a graphical method of determining the number of clusters.
##                 In the plot of D index, we seek a significant knee (the significant peak in Dindex
##                 second differences plot) that corresponds to a significant increase of the value of
##                 the measure.
##
## *******************************************************************
## * Among all indices:
## * 5 proposed 2 as the best number of clusters
## * 7 proposed 3 as the best number of clusters
## * 11 proposed 4 as the best number of clusters
##
##                     ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  4
##
##
## *******************************************************************
## Among all indices:
## ==================
## * 2 proposed  0 as the best number of clusters
## * 5 proposed  2 as the best number of clusters
## * 7 proposed  3 as the best number of clusters
## * 11 proposed  4 as the best number of clusters
## * 1 proposed  NA's as the best number of clusters
##
## Conclusion
```

```
## =========================
## * According to the majority rule, the best number of clusters is  4 .
```



Step 1: shows for all the data and pre bloom stage = 3 clusters is best out of our cluster range considered. This analysis was useful for determining the end method but ultimately was not really used, so has been commented out. All the clustering code which was of benefit and used can be seen in the clustering .rmd.

```
#Applying agglomerative clustering with 3 clusters:
#clustered <- agnes(dist(cluster_2_data, method = "euclidian"),
                   # diss=TRUE, method = "ward")
# add cluster labels to the training data
#cluster_2_data$cluster <- cutree(clustered, k=3)
# cl2_bloom <- vector(length = 3)
# #so doesn't work with subclusters 2 and 3
# for(i in 1:3){
#   cl2_bloom[i] <- fviz_nbclust(NbClust(cluster_2_data %>% filter(cluster == i) %>% select(ends_with("
#                                X2000nat), distance="euclidian", min.nc=2, max.nc=4, method="ward.D2",
# }


# fviz_nbclust(NbClust(cluster_2_data %>% filter(cluster == 2) %>% select(ends_with(".blm"), X2000nat),
#                                  min.nc=2, max.nc=4, method="ward.D2", index="all"))
#
#
# hclust(cluster_2_data)
```

Now from though process, only the variables that should be included in the clustering should be the pesticides:

insecticide/pesticide/thinner e.t.c

```
# cluster_2_data <- data_2012 %>%
#   select(ends_with(".pre"),ends_with(".blm"), ends_with(".pos")) %>%
#   unique()
#
#
# #Clusters of whole data combined - 7
# fviz_nbclust(NbClust(cluster_2_data, distance="euclidean",
#                                 min.nc=2, max.nc=4, method="ward.D2", index="all"))
# #pre ~ max of 4 clusters at each time point in method, but  in general if you allow
# #More as this argument then more returned a better  optimal cluster number
# fviz_nbclust(NbClust(cluster_2_data %>% select(ends_with(".pre")), distance="euclidean",
#                                 min.nc=2, max.nc=4, method="ward.D2", index="all"))
# #during
# fviz_nbclust(NbClust(cluster_2_data %>% select(ends_with(".blm")), distance="euclidean",
#                                 min.nc=2, max.nc=4, method="ward.D2", index="all"))
# #post
# fviz_nbclust(NbClust(cluster_2_data %>% select(ends_with(".pos")), distance="euclidean",
#                                 min.nc=2, max.nc=4, method="ward.D2", index="all"))
#
#
# #######Subsetting the data down by cluster results in the matrices being singular and therefore the c
# #run as expected, this could be looked at in detail however, since there are so few data points
# #it is not ideal to have a really sparse cluster set in reality
#
# #Testing
# fviz_nbclust(NbClust(cluster_2_data %>%  filter(cluster == 1) %>% select(ends_with(".pos")), distance
#                                 min.nc=2, max.nc=4, method="ward.D2", index="all"))
#
# cluster_2_data <- cluster_2_data %>%  mutate_all(function(x) ifelse(x == 0, runif(1, 0.00000000001,0.
#
# fviz_nbclust(NbClust(cluster_2_data %>%  filter(cluster == 2) %>% select(ends_with(".pos")), distance
#                                 min.nc=2, max.nc=4, method="ward.D2", index="all"))
#
# test <- agnes(dist(cluster_2_data %>%  filter(cluster == 2) %>% select(ends_with(".blm")), method = "
#                     diss=TRUE, method = "ward")
#
# cluster_2_data$cluster <- cutree(bleh, k=2)
#
# fviz_nbclust(NbClust(cluster_2_data %>%  filter(cluster == 2) %>% select(ends_with(".blm")), distance
#                                 min.nc=1, max.nc=3, method="ward.D2", index="all"))
#
#
# a <- matrix(c(0,1,1,0), nrow=2)
#
# #Checks
# fviz_nbclust(cluster_2_data %>% filter(cluster == 2)%>% select(ends_with(".blm")),pam , method = "sil
#
#
# clustered <- agnes(dist(cluster_2_data, method = "euclidian"),
#                     diss=TRUE, method = "ward")
# # add cluster labels to the training data
# cluster_2_data$cluster <- cutree(clustered, k=3)
```

```
# cl2_bloom <- vector(length = 3)
# #so doesn't work with subclusters 2 and 3
# for(i in 1:3){
#   cl2_bloom[i] <- fviz_nbclust(NbClust(cluster_2_data %>% filter(cluster == i)%>% select(ends_with(".
#                                   min.nc=1, max.nc= min(table(cluster_2_data$cluster))
#                                   , method="ward.D2", index="all"))
# }
#
#
# fviz_nbclust(NbClust(cluster_2_data , distance="euclidean",
#                                   min.nc=2, max.nc=min(table(cluster_2_data$cluster))
#                       , method="ward.D2", index="all"))
#
#
# hclust(cluster_2_data)
```

There is just not enough data to use this clustering technique at each stage of the clustering process in a good format, so will need to ensure that sparsity is taken into consideration as a main downside to clustering methods and these considerations will be properly exaimined in the clustering.rmd. The code here has been been kept as it was important for the future as it demonstrated why some methods wouldn't work adequately. But commented out as not "useful".

Variable justification

Code to decide upon the bee variables to take into consideration for the final response variables.

```
bee_correlations <- data.frame(matrix(0, nrow = 0, ncol = 0))

for(i in 1:7){
  for(j in 1:7){
  bee_correlations[i,j] <- round(cor(data_2012[5+i], data_2012[5 + j]), 2)
  }
}
a
```

```
##   apisAb.1 apisAb.2 apisAb.3 apisAb.4 apisAb.5 apisAb.6 apisAb.7 wildAbF.1
## 1     1.00     1.00     1.00     1.00     1.00     1.00     1.00      0.29
## 2     0.29     0.29     0.29     0.29     0.29     0.29     0.29      1.00
## 3     0.19     0.19     0.19     0.19     0.19     0.19     0.19      0.91
## 4     0.25     0.25     0.25     0.25     0.25     0.25     0.25      0.96
## 5     0.13     0.13     0.13     0.13     0.13     0.13     0.13      0.92
## 6     0.18     0.18     0.18     0.18     0.18     0.18     0.18      0.29
## 7     0.22     0.22     0.22     0.22     0.22     0.22     0.22      0.31
##   wildAbF.2 wildAbF.3 wildAbF.4 wildAbF.5 wildAbF.6 wildAbF.7 wildRichF.1
## 1      0.29      0.29      0.29      0.29      0.29      0.29        0.19
## 2      1.00      1.00      1.00      1.00      1.00      1.00        0.91
## 3      0.91      0.91      0.91      0.91      0.91      0.91        1.00
## 4      0.96      0.96      0.96      0.96      0.96      0.96        0.81
## 5      0.92      0.92      0.92      0.92      0.92      0.92        0.92
## 6      0.29      0.29      0.29      0.29      0.29      0.29        0.50
## 7      0.31      0.31      0.31      0.31      0.31      0.31        0.53
##   wildRichF.2 wildRichF.3 wildRichF.4 wildRichF.5 wildRichF.6 wildRichF.7
## 1        0.19        0.19        0.19        0.19        0.19        0.19
## 2        0.91        0.91        0.91        0.91        0.91        0.91
```

```
## 3        1.00        1.00        1.00        1.00        1.00        1.00
## 4        0.81        0.81        0.81        0.81        0.81        0.81
## 5        0.92        0.92        0.92        0.92        0.92        0.92
## 6        0.50        0.50        0.50        0.50        0.50        0.50
## 7        0.53        0.53        0.53        0.53        0.53        0.53
##   solitaryAbF.1 solitaryAbF.2 solitaryAbF.3 solitaryAbF.4 solitaryAbF.5
## 1          0.25          0.25          0.25          0.25          0.25
## 2          0.96          0.96          0.96          0.96          0.96
## 3          0.81          0.81          0.81          0.81          0.81
## 4          1.00          1.00          1.00          1.00          1.00
## 5          0.92          0.92          0.92          0.92          0.92
## 6          0.02          0.02          0.02          0.02          0.02
## 7          0.05          0.05          0.05          0.05          0.05
##   solitaryAbF.6 solitaryAbF.7 solitaryRichF.1 solitaryRichF.2
## 1          0.25          0.25            0.13            0.13
## 2          0.96          0.96            0.92            0.92
## 3          0.81          0.81            0.92            0.92
## 4          1.00          1.00            0.92            0.92
## 5          0.92          0.92            1.00            1.00
## 6          0.02          0.02            0.13            0.13
## 7          0.05          0.05            0.16            0.16
##   solitaryRichF.3 solitaryRichF.4 solitaryRichF.5 solitaryRichF.6
## 1            0.13            0.13            0.13            0.13
## 2            0.92            0.92            0.92            0.92
## 3            0.92            0.92            0.92            0.92
## 4            0.92            0.92            0.92            0.92
## 5            1.00            1.00            1.00            1.00
## 6            0.13            0.13            0.13            0.13
## 7            0.16            0.16            0.16            0.16
##   solitaryRichF.7 socialAbF.1 socialAbF.2 socialAbF.3 socialAbF.4
## 1            0.13        0.18        0.18        0.18        0.18
## 2            0.92        0.29        0.29        0.29        0.29
## 3            0.92        0.50        0.50        0.50        0.50
## 4            0.92        0.02        0.02        0.02        0.02
## 5            1.00        0.13        0.13        0.13        0.13
## 6            0.13        1.00        1.00        1.00        1.00
## 7            0.16        0.99        0.99        0.99        0.99
##   socialAbF.5 socialAbF.6 socialAbF.7 socialRichF.1 socialRichF.2
## 1        0.18        0.18        0.18          0.22          0.22
## 2        0.29        0.29        0.29          0.31          0.31
## 3        0.50        0.50        0.50          0.53          0.53
## 4        0.02        0.02        0.02          0.05          0.05
## 5        0.13        0.13        0.13          0.16          0.16
## 6        1.00        1.00        1.00          0.99          0.99
## 7        0.99        0.99        0.99          1.00          1.00
##   socialRichF.3 socialRichF.4 socialRichF.5 socialRichF.6 socialRichF.7
## 1          0.22          0.22          0.22          0.22          0.22
## 2          0.31          0.31          0.31          0.31          0.31
## 3          0.53          0.53          0.53          0.53          0.53
## 4          0.05          0.05          0.05          0.05          0.05
## 5          0.16          0.16          0.16          0.16          0.16
## 6          0.99          0.99          0.99          0.99          0.99
## 7          1.00          1.00          1.00          1.00          1.00
```

```r
for(i in 1:7){
 names(bee_correlations)[i] <- colnames(data_2012[5 +i])
}
bee_correlations
```

```
##   apisAb.1 apisAb.2 apisAb.3 apisAb.4 apisAb.5 apisAb.6 apisAb.7 wildAbF.1
## 1     1.00     1.00     1.00     1.00     1.00     1.00     1.00      0.29
## 2     0.29     0.29     0.29     0.29     0.29     0.29     0.29      1.00
## 3     0.19     0.19     0.19     0.19     0.19     0.19     0.19      0.91
## 4     0.25     0.25     0.25     0.25     0.25     0.25     0.25      0.96
## 5     0.13     0.13     0.13     0.13     0.13     0.13     0.13      0.92
## 6     0.18     0.18     0.18     0.18     0.18     0.18     0.18      0.29
## 7     0.22     0.22     0.22     0.22     0.22     0.22     0.22      0.31
##   wildAbF.2 wildAbF.3 wildAbF.4 wildAbF.5 wildAbF.6 wildAbF.7 wildRichF.1
## 1      0.29      0.29      0.29      0.29      0.29      0.29        0.19
## 2      1.00      1.00      1.00      1.00      1.00      1.00        0.91
## 3      0.91      0.91      0.91      0.91      0.91      0.91        1.00
## 4      0.96      0.96      0.96      0.96      0.96      0.96        0.81
## 5      0.92      0.92      0.92      0.92      0.92      0.92        0.92
## 6      0.29      0.29      0.29      0.29      0.29      0.29        0.50
## 7      0.31      0.31      0.31      0.31      0.31      0.31        0.53
##   wildRichF.2 wildRichF.3 wildRichF.4 wildRichF.5 wildRichF.6 wildRichF.7
## 1        0.19        0.19        0.19        0.19        0.19        0.19
## 2        0.91        0.91        0.91        0.91        0.91        0.91
## 3        1.00        1.00        1.00        1.00        1.00        1.00
## 4        0.81        0.81        0.81        0.81        0.81        0.81
## 5        0.92        0.92        0.92        0.92        0.92        0.92
## 6        0.50        0.50        0.50        0.50        0.50        0.50
## 7        0.53        0.53        0.53        0.53        0.53        0.53
##   solitaryAbF.1 solitaryAbF.2 solitaryAbF.3 solitaryAbF.4 solitaryAbF.5
## 1          0.25          0.25          0.25          0.25          0.25
## 2          0.96          0.96          0.96          0.96          0.96
## 3          0.81          0.81          0.81          0.81          0.81
## 4          1.00          1.00          1.00          1.00          1.00
## 5          0.92          0.92          0.92          0.92          0.92
## 6          0.02          0.02          0.02          0.02          0.02
## 7          0.05          0.05          0.05          0.05          0.05
##   solitaryAbF.6 solitaryAbF.7 solitaryRichF.1 solitaryRichF.2
## 1          0.25          0.25            0.13            0.13
## 2          0.96          0.96            0.92            0.92
## 3          0.81          0.81            0.92            0.92
## 4          1.00          1.00            0.92            0.92
## 5          0.92          0.92            1.00            1.00
## 6          0.02          0.02            0.13            0.13
## 7          0.05          0.05            0.16            0.16
##   solitaryRichF.3 solitaryRichF.4 solitaryRichF.5 solitaryRichF.6
## 1            0.13            0.13            0.13            0.13
## 2            0.92            0.92            0.92            0.92
## 3            0.92            0.92            0.92            0.92
## 4            0.92            0.92            0.92            0.92
## 5            1.00            1.00            1.00            1.00
## 6            0.13            0.13            0.13            0.13
## 7            0.16            0.16            0.16            0.16
##   solitaryRichF.7 socialAbF.1 socialAbF.2 socialAbF.3 socialAbF.4
```

```
## 1            0.13        0.18        0.18        0.18        0.18
## 2            0.92        0.29        0.29        0.29        0.29
## 3            0.92        0.50        0.50        0.50        0.50
## 4            0.92        0.02        0.02        0.02        0.02
## 5            1.00        0.13        0.13        0.13        0.13
## 6            0.13        1.00        1.00        1.00        1.00
## 7            0.16        0.99        0.99        0.99        0.99
##   socialAbF.5 socialAbF.6 socialAbF.7 socialRichF.1 socialRichF.2
## 1        0.18        0.18        0.18          0.22          0.22
## 2        0.29        0.29        0.29          0.31          0.31
## 3        0.50        0.50        0.50          0.53          0.53
## 4        0.02        0.02        0.02          0.05          0.05
## 5        0.13        0.13        0.13          0.16          0.16
## 6        1.00        1.00        1.00          0.99          0.99
## 7        0.99        0.99        0.99          1.00          1.00
##   socialRichF.3 socialRichF.4 socialRichF.5 socialRichF.6 socialRichF.7
## 1          0.22          0.22          0.22          0.22          0.22
## 2          0.31          0.31          0.31          0.31          0.31
## 3          0.53          0.53          0.53          0.53          0.53
## 4          0.05          0.05          0.05          0.05          0.05
## 5          0.16          0.16          0.16          0.16          0.16
## 6          0.99          0.99          0.99          0.99          0.99
## 7          1.00          1.00          1.00          1.00          1.00
```